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

ROHM GROUP
LAPIS TECHNOLOGY

FEUL62Q1000-10

ML62Q1000 Series
User's Manual
(ML62Q1300/1500/1800/1700 Group)

Issue Date: Dec. 15, 2023

ROHM SEMICONDUCTOR

Notes

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

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

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

4) No license, expressly or implied, is granted hereby under any intellectual property rights or other rights of LAPIS Technology or any third party with respect to LAPIS Technology Products or the information contained in this document (including but not limited to, the Product data, drawings, charts, programs, algorithms, and application examples, etc.). Therefore, LAPIS Technology 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) LAPIS Technology intends our Products to be used in a way indicated in this document. Please be sure to contact a ROHM sales office if you consider the use of our Products in different way from original use indicated in this document. For use of our Products in medical systems, please be sure to contact a LAPIS Technology representative and must obtain written agreement. Do not use our Products in applications which may directly cause injuries to human life, and which require extremely high reliability, such as aerospace equipment, nuclear power control systems, and submarine repeaters, etc. LAPIS Technology disclaims any and all liability for any losses and damages incurred by you or third parties arising by using the Product for purposes not intended by us without our prior written consent.

6) All information contained in this document is subject to change for the purpose of improvement, etc. without any prior notice. Before purchasing or using LAPIS Technology Products, please confirm the latest information with a ROHM sales office. LAPIS Technology has used reasonable care to ensure the accuracy of the information contained in this document, however, LAPIS Technology shall have no responsibility for any damages, expenses or losses arising from inaccuracy or errors of such information.

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

8) 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.

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

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

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

LAPIS Technology Co.,Ltd.
2-4-8 Shinyokohama, Kouhoku-ku, Yokohama 222-8575, Japan
https://www.lapis-tech.com/en/

LTSZ08023・01・002

Notes for product usage

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

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

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

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

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

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

Preface

This manual describes the operation of the hardware of the 16-bit microcontroller ML62Q1000 Series.
This manual is for ML62Q1300 group, ML62Q1500 group, ML62Q1700 group and ML62Q1800 group.
See other manuals for ML62Q1200 group, ML62Q1400 group and ML62Q1600 group.

See the relevant manuals listed in supplementary volume; “MCU Relevant Documents list” as necessary.

Notation

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

♦ Register description

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

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

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

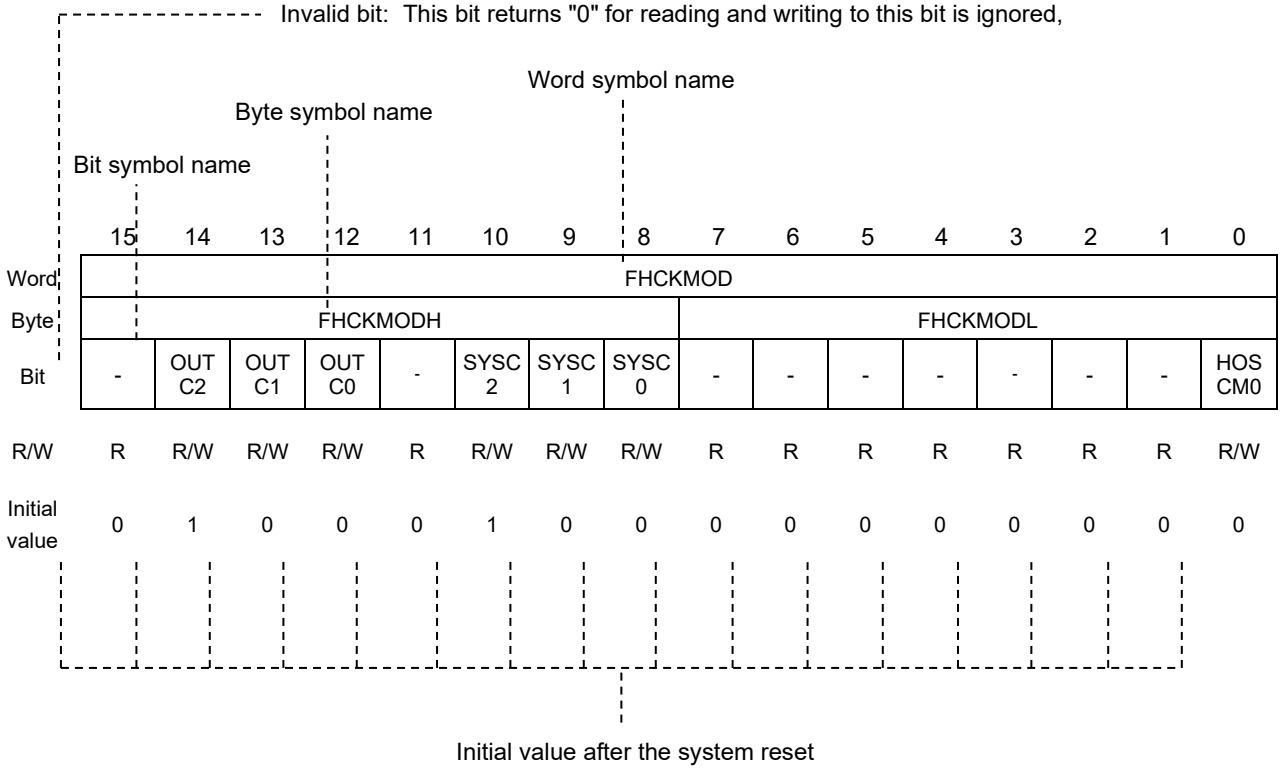


Table of Contents

Chapter 1

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 14

Chapter 15

Chapter 16

Chapter 17

Chapter 18

Chapter 22

Chapter 23

Chapter 24

Chapter 25

Chapter 26

Chapter 27

Chapter 28

Chapter 29

Appendixes

Revision History

Chapter 1 Overview

1. Overview

ML62Q1000 Series 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 generator, DMA controller, Clock generator, Simplified RTC, Timer, General Purpose Ports, UART, Synchronous serial port, I^2C bus interface unit(Master, Slave), Buzzer, Voltage Level Supervisor(VLS), Successive approximation type A/D converter, D/A converter , Analog comparator, LCD driver, Safety function(IEC60730/60335 Class B), and so on.
The CPU nX-U16/100 is capable of efficient instruction execution in 1-instruction 1-clock mode by pipeline architecture parallel processing.

- Aplications

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

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

- Product List

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 product lists are shown below.

Table 1-1 ML62Q1300 Group Product List

Program memory	Data memory	Data Flash	16pin SSOP16 WQFN16	20pin TSSOP20	24pin WQFN24	32pin TQFP32 WQFN32
64Kbyte	4Kbyte	2Kbyte	-	-	ML62Q1347	ML62Q1367
48Kbyte			-	-	ML62Q1346	ML62Q1366
32Kbyte			-	-	ML62Q1345	ML62Q1365
32Kbyte	2Kbyte		ML62Q1325	ML62Q1335	-	-
24Kbyte			ML62Q1324	ML62Q1334	-	-
16Kbyte			ML62Q1323	ML62Q1333	-	-

Table 1-2 ML62Q1500/ML62Q1800 Group Product List

Program memory	Data memory	Data Flash	48pin TQFP48	52pin TQFP52	64pin QFP64 TQFP64	80pin QFP80	100pin QFP100 TQFP100
512Kbyte	32Kbyte	8Kbyte	-	-	ML62Q1859	ML62Q1869	ML62Q1879
384Kbyte			-	-	ML62Q1858	ML62Q1868	ML62Q1878
256Kbyte	16Kbyte	4Kbyte	-	-	ML62Q1557	ML62Q1567	ML62Q1577
192Kbyte			-	-	ML62Q1556	ML62Q1566	ML62Q1576
160Kbyte			-	-	ML62Q1555	ML62Q1565	ML62Q1575
128Kbyte	16Kbyte		-	-	-	ML62Q1564	ML62Q1574
	8Kbyte		ML62Q1534	ML62Q1544	ML62Q1554	-	-
96Kbyte	16Kbyte		-	-	-	ML62Q1563	ML62Q1573
	8Kbyte		ML62Q1533	ML62Q1543	ML62Q1553	-	-
64Kbyte	8Kbyte		ML62Q1532	ML62Q1542	ML62Q1552	-	-
48Kbyte			ML62Q1531	ML62Q1541	ML62Q1551	-	-
32Kbyte			ML62Q1530	ML62Q1540	ML62Q1550	-	-

Table 1-3 ML62Q1700 Group Product List

Program memory	Data memory	Data Flash	48pin TQFP48	52pin TQFP52	64pin QFP64 TQFP64	80pin QFP80	100pin QFP100 TQFP100
512Kbyte	32Kbyte	8Kbyte	-	-	ML62Q1729	ML62Q1739	ML62Q1749
384Kbyte			-	-	ML62Q1728	ML62Q1738	ML62Q1748
256Kbyte	16Kbyte	4Kbyte	-	-	ML62Q1727	ML62Q1737	ML62Q1747
192Kbyte			-	-	ML62Q1726	ML62Q1736	ML62Q1746
160Kbyte			-	-	ML62Q1725	ML62Q1735	ML62Q1745
128Kbyte	16Kbyte		-	-	-	ML62Q1734	ML62Q1744
	8Kbyte		ML62Q1704	ML62Q1714	ML62Q1724	-	-
96Kbyte	16Kbyte		-	-	-	ML62Q1733	ML62Q1743
	8Kbyte		ML62Q1703	ML62Q1713	ML62Q1723	-	-
64Kbyte	8Kbyte		ML62Q1702	ML62Q1712	ML62Q1722	-	-
48Kbyte			ML62Q1701	ML62Q1711	ML62Q1721	-	-
32Kbyte			ML62Q1700	ML62Q1710	ML62Q1720	-	-

Please see the page i “Notes” and the page ii “Notes for product usage” in this document on use with ML62Q1000 Series.

1.1 Features

- CPU
  - 16-bit RISC CPU: nX-U16/100 (A35 core)
  - Instruction system: 16-bit length instructions
  - Instruction set: Transfer, arithmetic operations, comparison, logic operations, multiplication/division, bit manipulations, bit logic operations, jump, conditional jump, call return stack manipulations, arithmetic shift, and so on
  - Built-in On-chip debug function
  - Built-in ISP (In-System Programming) function
  - Minimum instruction execution time
    Approximately 30.5 μs (at 32.768 kHz system clock)
    Approximately 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)
  - Signed or Unsigned is selectable

- Operating voltage and temperature
  - Operating voltage: V_{DD} = 1.6 to 5.5 V (V_{DD} should be 1.8V or over at Power-on)
  - Operating temperature: -40 °C to +105 °C

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

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

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

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

- Clock Generation Circuit
  - Low-speed clock (LSCLK)
    Internal low-speed RC oscillation: Approximately 32.768 kHz
    External low-speed clock input (ML62Q1500/ML62Q1800 and ML62Q1700 group only)
    : Approximately 32.768 kHz
    External low-speed crystal oscillation (ML62Q1500/ML62Q1800 and ML62Q1700 group only)
    : 32.768 kHz crystal resonator is connectable.
    3 selectable crystal oscillation mode (Tough, Normal, and Low current consumption)
    · Tough mode: Largest oscillation allowance to make highest resistance against leakage between the pins
    · Normal mode: Normal oscillation allowance and current consumption
    · Low current consumption mode: Smallest oscillation allowance to make lower current consumption
  - High-speed clock (HSCLK)
    PLL oscillation: 2 selectable oscillation frequency (24MHz and 16MHz) by code option
  - Watch Dog Timer (WDT): built-in independent clock for WDT (RC1K: Approximately 1kHz)

- Reset
  - Reset by reset input pin
  - Reset by Power-On Reset
  - Reset by WDT overflow
  - Reset by WDT invalid clear
  - Reset by RAM parity error
  - Reset by unused ROM area access (instruction access)
  - Reset by voltage level supervisor (VLS)
  - Software reset by BRK instruction (reset CPU only)
  - Reset the peripherals individually
  - Collective reset to the all control pins and peripheral circuits.

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

- Interrupt controller
  - External interrupt ports : max. 12
  - Non-maskable interrupt source: 1 (Internal source: WDT)
  - Maskable interrupt sources: max. 51
  - Four step interrupt levels

- Watchdog timer (WDT)
  - Selectable Operating clock : select RC1K or LSCLK by code option
  - Overflow period: 8selectable (7.8ms, 15.6ms, 31.3ms, 62.5ms, 125ms, 500ms, 2s and 8s)
  - Selectable window function (enable or disable): configurable clear enable period (50% or 75% of overflow period)
  - Selectable WDT operation : select Enable or Disable by code option
  - Readable WDT counter : WDT counter monitor function

- DMA (Direct Memory Access) controller
  - Channel : 2channel
  - Transfer unit: 8bit/16bit
  - Transfer count: 1 to 1024
  - Transfer cycle: 2 cycle transfer
  - Transfer address: Fixed addressing mode, inclement addressing mode , and decrement addressing mode
  - Transfer target: Special Function Register (SFR)/RAM → SFR/RAM (Transfer from/to Flash is not supported)
  - Transfer request: External pins, Serial communication unit, Successive approximation type A/D converter, 16bit timer, and Functional timer

- Low-speed Time base counter
  - Generate 8 frequency (128Hz to1Hz) internal pulse signals by dividing the Low-speed clock (LSCLK)
  - Selectable 3 interrupts from eight frequency internal pulse signals
  - 1Hz or 2Hz output from general purpose port
  - Built-in Frequency adjust function (ML62Q1500/ML62Q1800 and ML62Q1700 group only) : Adjust range: Approximately -488ppm to +488ppm, adjust resolution: Approximately 0.119ppm

- Simplified RTC (ML62Q1500/ML62Q1800 and ML62Q1700 group only)
  - Channel: 1 channel
  - Count by a unit for one second from "00 min. 00 sec" to "59 min. 59 sec"
  - Selectable Periodical interrupt request from four periods (0.5s, 1s, 30s or 60s)
  - Built-in minute and second writing error protraction function

- Functional timer
  - Channel: Max. 8 channel
  - Built-in timer, capture, and PWM function by 16 bit counter
  - One shot mode is available
  - Two types of PWM output with the same period and different duties, and complementary PWM output with the dead time
  - Monitor input signal duty and the period by capture function
  - Generate periodical interrupts, duty interrupts, and interrupts coincided with set value.
  - Counter Start, Stop, Counter clear triggered by an external inputs or Timer
  - Generate Emergency stop and emergency stop interrupt triggered by an external input
  - Same start/stop among different channels of the functional timer
  - Selectable counter clock(external clock or divided by 1 to 128 of LSCLK or HSCLK) for each channels

- 16bit General timers
  - Channel: Max. 8channel
  - 8 bits timer mode and 16-bit timer mode
  - Same start/stop among different channels of 16bit (8bit) timer
  - Timer output (toggled by overflow)
  - Selectable counter clock (external clock or divided by 1 to 128 of LSCLK or HSCLK) for each channels

- Serial communication unit
  - Synchronous Serial Port (SSIO) mode or UART mode is selectable
  - Channel: Max. 6 channel

    < Synchronous Serial Port mode >
    - Selectable from Master and Slave
    - Selectable from LSB first or MSB first
    - Selectable 8-bit length or 16-bit length

    < UART mode>
    - Selectable from Full-duplex communication mode or Half-duplex communication mode
    - 5 to 8 bit length, parity or no parity, odd parity or even parity, 1 stop bit or 2 stop bits
    - Selectable from Positive logic or Negative logic
    - Selectable from LSB first or MSB first
    - Configurable wide range communication speed
      32.768kHz operation clock : 1 bit/s to 4,800 bit/s
      24MHz operation clock : 600 bit/s to 3M bit/s
      16MHz operation clock : 300 bit/s to 2M bit/s
    - Built-in baud rate generator

- I^2C bus unit (Master / Slave)
  - Selectable from Master mode or Slave mode
  - Channel: 1 channel

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

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

- General-purpose ports (GPIO)
  - I/O port: Max. 92 (Including one pin for on-chip debug and pins for other shared functions)
  - Input port: Max. 2 (Including a shared function)
  - External interrupt port: Max. 12
  - LED driver port : Max. 91
  - Carrier frequency output function (for IR communication)

- Successive approximation type A/D converter(SA-ADC)
  - Channel: Max.16channel
  - Resolution: 10bit
  - Conversion time: Min. 2.25μs/channel (When the conversion clock speed is 8MHz)
  - Reference voltages are selectable
    (V_{DD} pin / Internal reference voltage(V_{REFI} = Approximately 1.55V) / External reference voltage (V_{REF} pin))
  - Selected channel repeat conversion
  - Dedicated result register for each channel
  - Interrupt determining by upper limit or lower limit threshold of conversion result

- Voltage Level Supervisor (VLS)
  - Accuracy: ±4%
  - Threshold voltage: 12 selectable (from 1.85V to 4.00V)
  - Functional Voltage level detection reset (VLS reset)
  - Functional Voltage level detection interrupt (VLS0 interrupt)

- Analog comparator
  - Channel: Max. 2channel
  - Selectable interrupt from the comparator output (rising edge or falling edge)
  - Selectable from sampling or without sampling
  - Comparable with external 2 inputs
  - Comparable with external input and internal reference voltage (0.8V)

- D/A converter
  - Channel: Max. 2channel
  - Resolution: 8bit
  - Output impedance: 6k ohm (Typ.)
  - R-2R ladder type

- Buzzer
  - 4 buzzer mode (Continuous sound, Single sound, Intermittent sound 1 and Intermittent sound 2)
  - 8 frequencies (4.096kHz to 293Hz)
  - 15 step duty (1/16 to 15/16)
  - Selectable from positive logic buzzer output or negative logic buzzer output

- CRC(Cyclic Redundancy Check) generator
  - Generation equation: $X^{16}+X^{12}+X^5+1$
  - Selectable from LSB first or MSB first
  - Built-in Automatic program memory CRC calculation mode in HALT mode

- LCD driver (ML62Q1700 group)
  - Max. 480 dots (60seg x 8 com) *1

    ML62Q1700/ML62Q1701/ML62Q1702/ML62Q1703/ML62Q1704:
    24seg×8com (com Max), 29seg×3com (seg Max.)
    ML62Q1710/ML62Q1711/ML62Q1712/ML62Q1713/ML62Q1714:
    27seg×8com (com Max.), 32seg×3com (seg Max.)
    ML62Q1720/ML62Q1721/ML62Q1722/ML62Q1723/ML62Q1724/
    ML62Q1725/ML62Q1726/ML62Q1727/ML62Q1728/ML62Q1729:
    35seg×8com (com Max.), 40seg×3com (seg Max.)
    ML62Q1733/ML62Q1734/ML62Q1735/ML62Q1736/ML62Q1737/ML62Q1738/ML62Q1739:
    45seg×8com（com Max.）, 50seg×3com（seg Max.）
    ML62Q1743/ML62Q1744/ML62Q1745/ML62Q1746/ML62Q1747/ML62Q1748/ML62Q1749:
    60seg×8com（com Max.）, 65seg×3com（seg Max.）

    *1 : Five pins are shared for common or segment, selectable by setting a SFR
  - 1/3 bias (built-in bias generation circuit)
  - Frame frequency (Approximately. 32Hz, 38Hz, 64Hz, 75Hz, 128Hz and 150Hz)
  - Four bias generation modes (Internal voltage boost, External capacitive voltage divide, Internal capacitive voltage divide and External supply voltages)
  - Contrast adjustment (32 steps) is available in the Internal voltage boost mode.

- Safety Function (IEC60730/60335 Class B)
  - Automatic switching to the internal low-speed RC oscillation in case the low-speed crystal oscillation stopped
  - RAM/SFR guard
  - Automatic program memory CRC calculation
  - RAM parity error detection
  - ROM unused area access reset (instruction access)
  - Clock mutual monitoring
  - WDT counter monitoring
  - SA-ADC test
  - UART test
  - Synchronous serial I/O test
  - I^2C bus test
  - GPIO test

- Shipping package

ML62Q1300 group

Package	Body size (including lead) [mm × mm]	Pin pitch [mm]	Packing form and Product name	
			Tray	Tape & Reel
16 pin plastic SSOP	5.0 × 4.4 (5.0 × 6.4)	0.65	ML62Q1323-xxxMBZ0ARL ML62Q1324-xxxMBZ0ARL ML62Q1325-xxxMBZ0ARL	ML62Q1323-xxxMBZ0ATL ML62Q1324-xxxMBZ0ATL ML62Q1325-xxxMBZ0ATL
16 pin plastic WQFN	4.0 × 4.0 (-)	0.50	ML62Q1323-xxxGDZWAX ML62Q1324-xxxGDZWAX ML62Q1325-xxxGDZWAX	ML62Q1323-xxxGDZWBX ML62Q1324-xxxGDZWBX ML62Q1325-xxxGDZWBX
20 pin plastic TSSOP	6.5 × 4.4 (6.5× 6.4)	0.65	ML62Q1333-xxxTDZWARL ML62Q1334-xxxTDZWARL ML62Q1335-xxxTDZWARL	ML62Q1333-xxxTDZWATL ML62Q1334-xxxTDZWATL ML62Q1335-xxxTDZWATL
24 pin plastic WQFN	4.0 × 4.0 (-)	0.50	ML62Q1345-xxxGDZWAX ML62Q1346-xxxGDZWAX ML62Q1347-xxxGDZWAX	ML62Q1345-xxxGDZWBX ML62Q1346-xxxGDZWBX ML62Q1347-xxxGDZWBX
32 pin plastic TQFP	7.0 × 7.0 (9.0 × 9.0)	0.80	ML62Q1365-xxxTBZWAX ML62Q1366-xxxTBZWAX ML62Q1367-xxxTBZWAX	ML62Q1365-xxxTBZWBX ML62Q1366-xxxTBZWBX ML62Q1367-xxxTBZWBX
32 pin plastic WQFN	5.0 × 5.0 (-)	0.5	ML62Q1365-xxxGDZWAX ML62Q1366-xxxGDZWAX ML62Q1367-xxxGDZWAX	ML62Q1365-xxxGDZWBX ML62Q1366-xxxGDZWBX ML62Q1367-xxxGDZWBX

xxx: ROM code number

ML62Q1500/ML62Q1800 group

Package	Body size (including lead) [mm × mm]	Pin pitch [mm]	Packing form and Product name	
			Tray	Tape & Reel
48 pin plastic TQFP	7.0 × 7.0 (9.0 × 9.0)	0.50	ML62Q1530-xxxTBZWAX ML62Q1531-xxxTBZWAX ML62Q1532-xxxTBZWAX ML62Q1533-xxxTBZWAX ML62Q1534-xxxTBZWAX	ML62Q1530-xxxTBZWBX ML62Q1531-xxxTBZWBX ML62Q1532-xxxTBZWBX ML62Q1533-xxxTBZWBX ML62Q1534-xxxTBZWBX
52 pin plastic TQFP	10.0 × 10.0 (12.0 × 12.0)	0.65	ML62Q1540-xxxTBZWAX ML62Q1541-xxxTBZWAX ML62Q1542-xxxTBZWAX ML62Q1543-xxxTBZWAX ML62Q1544-xxxTBZWAX	ML62Q1540-xxxTBZWBX ML62Q1541-xxxTBZWBX ML62Q1542-xxxTBZWBX ML62Q1543-xxxTBZWBX ML62Q1544-xxxTBZWBX
64 pin plastic TQFP	10.0 × 10.0 (12.0 × 12.0)	0.50	ML62Q1550-xxxTBZWAX ML62Q1551-xxxTBZWAX ML62Q1552-xxxTBZWAX ML62Q1553-xxxTBZWAX ML62Q1554-xxxTBZWAX ML62Q1555-xxxTBZWAX ML62Q1556-xxxTBZWAX ML62Q1557-xxxTBZWAX ML62Q1858-xxxTBZWAX ML62Q1859-xxxTBZWAX	ML62Q1550-xxxTBZWBX ML62Q1551-xxxTBZWBX ML62Q1552-xxxTBZWBX ML62Q1553-xxxTBZWBX ML62Q1554-xxxTBZWBX ML62Q1555-xxxTBZWBX ML62Q1556-xxxTBZWBX ML62Q1557-xxxTBZWBX ML62Q1858-xxxTBZWBX ML62Q1859-xxxTBZWBX
64 pin plastic QFP	14.0 × 14.0 (16.0 × 16.0)	0.80	ML62Q1550-xxxGAZWAX ML62Q1551-xxxGAZWAX ML62Q1552-xxxGAZWAX ML62Q1553-xxxGAZWAX ML62Q1554-xxxGAZWAX ML62Q1555-xxxGAZWAX ML62Q1556-xxxGAZWAX ML62Q1557-xxxGAZWAX ML62Q1858-xxxGAZWAX ML62Q1859-xxxGAZWAX	-
80 pin plastic QFP	14.0 × 14.0 (16.0 × 16.0)	0.65	ML62Q1563-xxxGAZWAX ML62Q1564-xxxGAZWAX ML62Q1565-xxxGAZWAX ML62Q1566-xxxGAZWAX ML62Q1567-xxxGAZWAX ML62Q1868-xxxGAZWAX ML62Q1869-xxxGAZWAX	–
100 pin plastic TQFP	14.0 × 14.0 (16.0 × 16.0)	0.5	ML62Q1573-xxxTBZWAX ML62Q1574-xxxTBZWAX ML62Q1575-xxxTBZWAX ML62Q1576-xxxTBZWAX ML62Q1577-xxxTBZWAX ML62Q1878-xxxTBZWAX ML62Q1879-xxxTBZWAX	-
100 pin plastic TQFP	20.0 × 14.0 (25.0 × 19.0)	0.65	ML62Q1573-xxxGAZWAX ML62Q1574-xxxGAZWAX ML62Q1575-xxxGAZWAX ML62Q1576-xxxGAZWAX ML62Q1577-xxxGAZWAX ML62Q1878-xxxTBZWAX ML62Q1879-xxxTBZWAX	–

xxx: ROM code number

ML62Q1700 group

Package	Body size (including lead) [mm × mm]	Pin pitch [mm]	Packing form and Product name	
			Tray	Tape & Reel
48 pin plastic TQFP	7.0 × 7.0 (9.0 × 9.0)	0.50	ML62Q1700-xxxTBZWAX ML62Q1701-xxxTBZWAX ML62Q1702-xxxTBZWAX ML62Q1703-xxxTBZWAX ML62Q1704-xxxTBZWAX	ML62Q1700-xxxTBZWBX ML62Q1701-xxxTBZWBX ML62Q1702-xxxTBZWBX ML62Q1703-xxxTBZWBX ML62Q1704-xxxTBZWBX
52 pin plastic TQFP	10.0 × 10.0 (12.0 × 12.0)	0.65	ML62Q1710-xxxTBZWAX ML62Q1711-xxxTBZWAX ML62Q1712-xxxTBZWAX ML62Q1713-xxxTBZWAX ML62Q1714-xxxTBZWAX	ML62Q1710-xxxTBZWBX ML62Q1711-xxxTBZWBX ML62Q1712-xxxTBZWBX ML62Q1713-xxxTBZWBX ML62Q1714-xxxTBZWBX
64 pin plastic TQFP	10.0 × 10.0 (12.0 × 12.0)	0.50	ML62Q1720-xxxTBZWAX ML62Q1721-xxxTBZWAX ML62Q1722-xxxTBZWAX ML62Q1723-xxxTBZWAX ML62Q1724-xxxTBZWAX ML62Q1725-xxxTBZWAX ML62Q1726-xxxTBZWAX ML62Q1727-xxxTBZWAX ML62Q1728-xxxTBZWAX ML62Q1729-xxxTBZWAX	ML62Q1720-xxxTBZWBX ML62Q1721-xxxTBZWBX ML62Q1722-xxxTBZWBX ML62Q1723-xxxTBZWBX ML62Q1724-xxxTBZWBX ML62Q1725-xxxTBZWBX ML62Q1726-xxxTBZWBX ML62Q1727-xxxTBZWBX ML62Q1728-xxxTBZWBX ML62Q1729-xxxTBZWBX
64 pin plastic QFP	14.0 × 14.0 (16.0 × 16.0)	0.80	ML62Q1720-xxxGAZWAX ML62Q1721-xxxGAZWAX ML62Q1722-xxxGAZWAX ML62Q1723-xxxGAZWAX ML62Q1724-xxxGAZWAX ML62Q1725-xxxGAZWAX ML62Q1726-xxxGAZWAX ML62Q1727-xxxGAZWAX ML62Q1728-xxxGAZWAX ML62Q1729-xxxGAZWAX	-
80 pin plastic QFP	14.0 × 14.0 (16.0 × 16.0)	0.65	ML62Q1733-xxxGAZWAX ML62Q1734-xxxGAZWAX ML62Q1735-xxxGAZWAX ML62Q1736-xxxGAZWAX ML62Q1737-xxxGAZWAX ML62Q1738-xxxGAZWAX ML62Q1739-xxxGAZWAX	-
100 pin plastic TQFP	14.0 × 14.0 (16.0 × 16.0)	0.5	ML62Q1743-xxxTBZWAX ML62Q1744-xxxTBZWAX ML62Q1745-xxxTBZWAX ML62Q1746-xxxTBZWAX ML62Q1747-xxxTBZWAX ML62Q1748-xxxTBZWAX ML62Q1749-xxxTBZWAX	-
100 pin plastic TQFP	20.0 × 14.0 (25.0 × 19.0)	0.65	ML62Q1743-xxxGAZWAX ML62Q1744-xxxGAZWAX ML62Q1745-xxxGAZWAX ML62Q1746-xxxGAZWAX ML62Q1747-xxxGAZWAX ML62Q1748-xxxGAZWAX ML62Q1749-xxxGAZWAX	-

xxx : ROM code number

How To Read The Part Number

ML62Q1300 Group how to read the part number

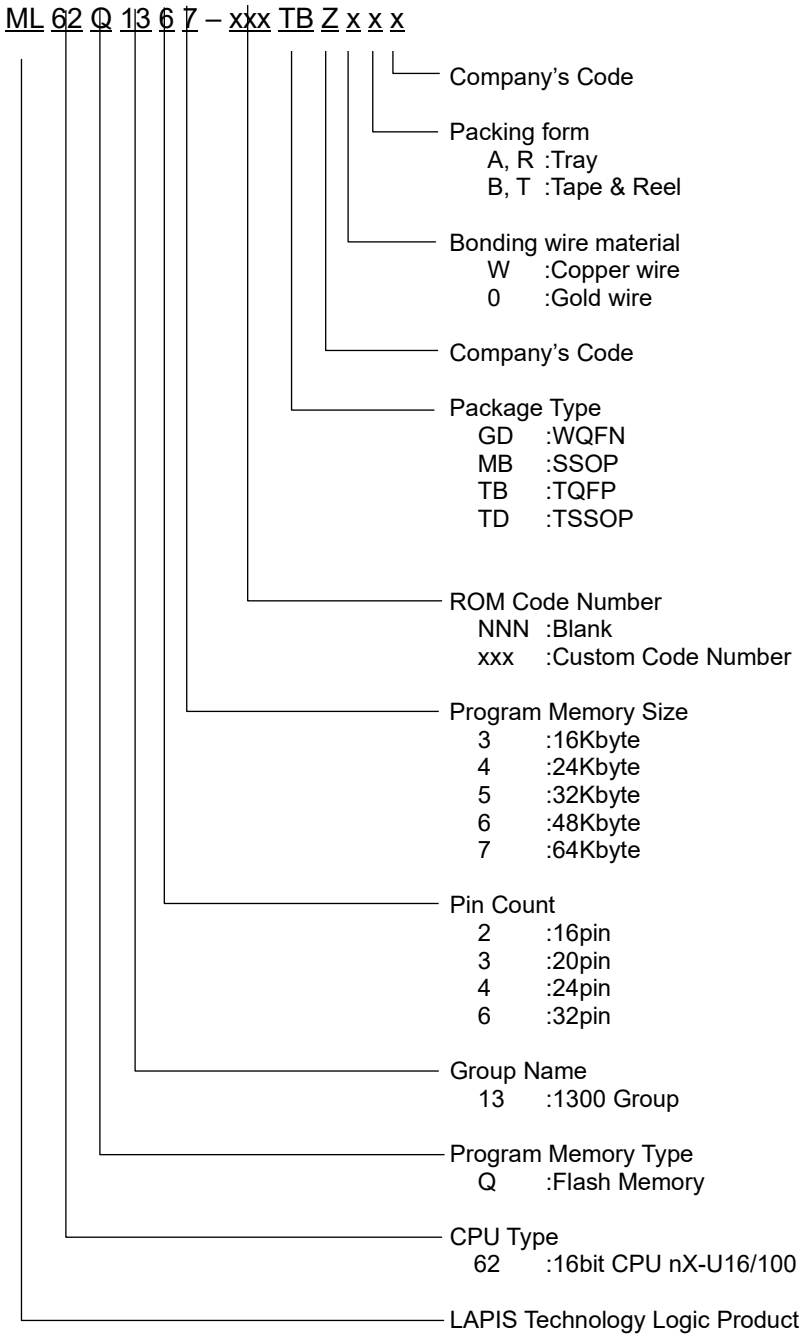


Figure 1-1 ML62Q1300 Group Part Number

1.1.1 How To Read The Part Number

ML62Q1500/ML62Q1800 Group how to read the part number

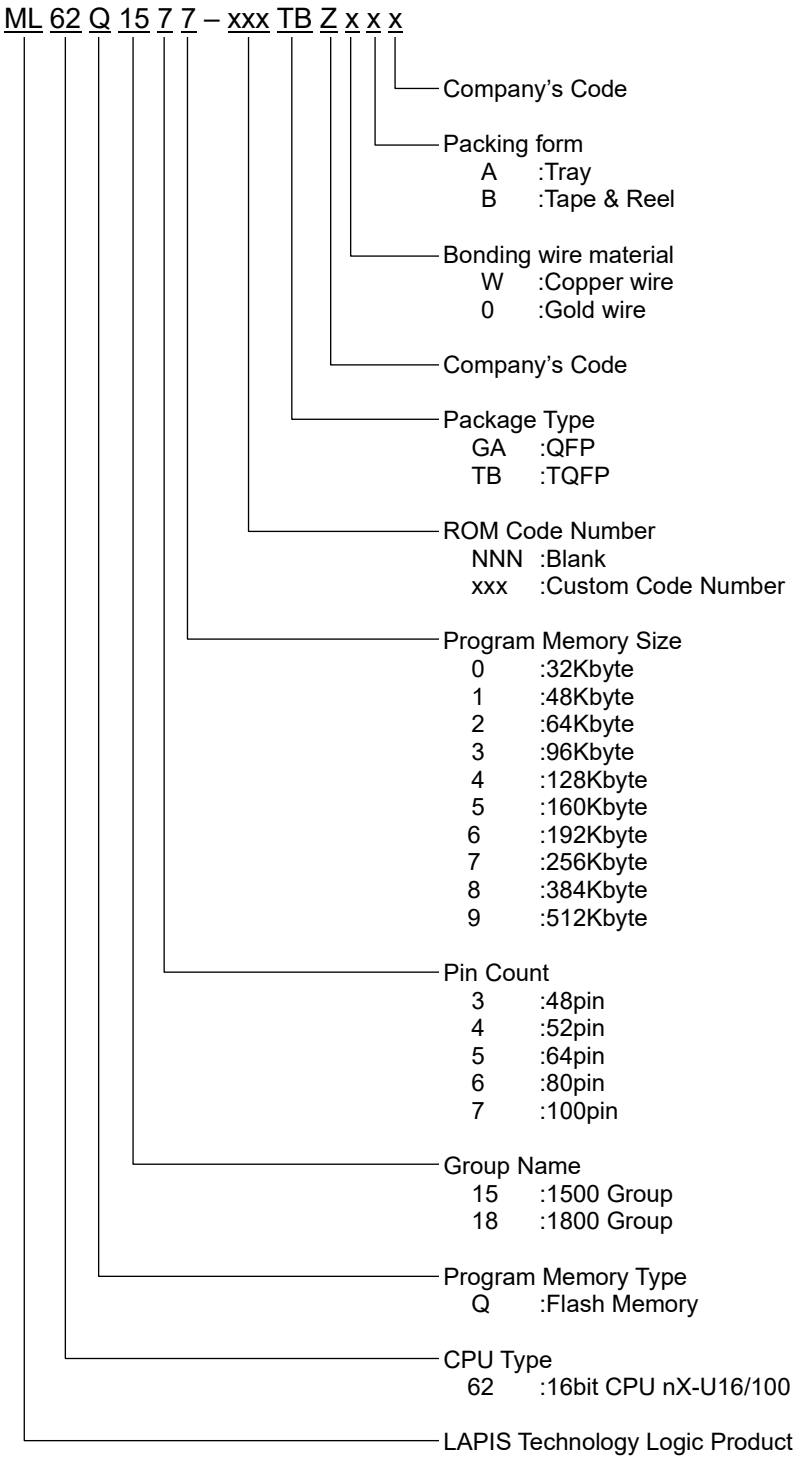


Figure 1-2 ML62Q1500/ML62Q1800 Group Part Number

ML62Q1700 Group how to read the part number

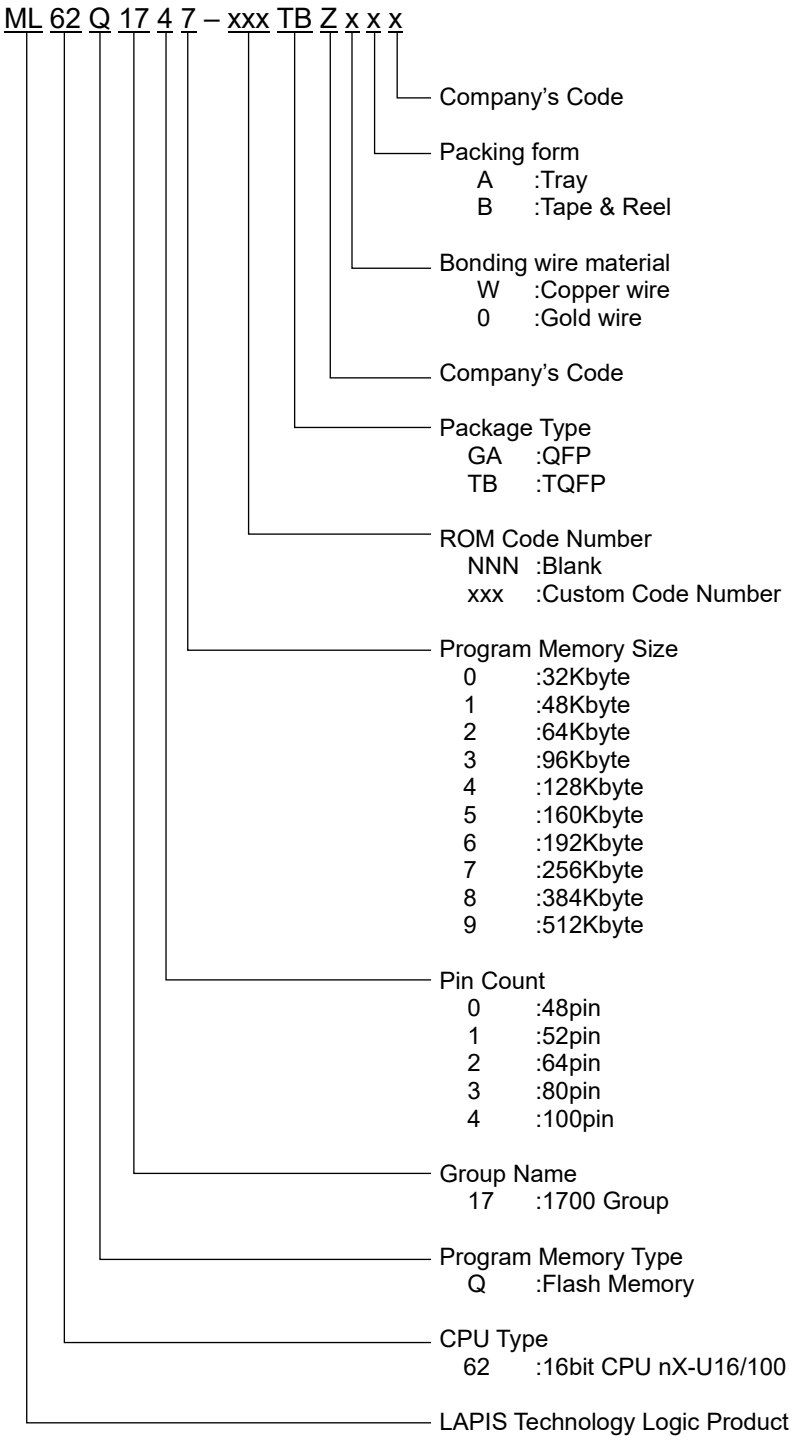


Figure 1-3 ML62Q1700 Group Part Number

1.1.2 Main Function List

ML62Q1300 Group Main Function List

Table 1-4 Main Function List

Part number	Pin					Interrupt		Timer		communication			Analog			
	Total pin-count	Power pin count	Reset Input pin count	I/O pin count	LED drive [port] (shared with the I/O port)	Internal interrupt [source]	External interrupt [port]	Functional Timer [channel]	16bit General I Timer*1 [channel]	Serial communication unit (Full-duplex UART or Synchronous serial) [channel] *2	I²C bus unit (Master/Slave) [channel]	I²C bus interface (Master only) [channel]	Successive type A/D converter [channel]	Analog comparator [channel]	Analog comparator [input pin]	D/A converter [channel]
ML62Q1323	16	3	1	12	11	23	8	4	4	2	1	1	6	1	2	0
ML62Q1324																
ML62Q1325																
ML62Q1333	20			16	15								8			
ML62Q1334																
ML62Q1335																
ML62Q1345	24			20	19	25			6							1
ML62Q1346																
ML62Q1347																
ML62Q1365	32			28	27											
ML62Q1366																
ML62Q1367																

*1 : One 16bit timer is configurable as two 8bit timers
*2 : Synchronous Communication unit includes UART and Synchronous Serial Port. UART and Synchronous Serial Port can not be used at the same time in the same channel.

ML62Q1500/ML62Q1800 Group Main Function List

Table 1-5 Main Function List

Part number	Pin						Interrupt		Timer			communication			Analog			
	Total pin-count	Power pin count	Reset Input pin count	Input pin count*3	I/O pin count	LED drive port (shared with the I/O port)	Internal interrupt [source]	External interrupt [port]	Functional Timer [channel]	16bit General I Timer*1 [channel]	Simplified RTC [channel]	Serial communication unit (Full-duplex UART or Synchronous serial) [channel]*2	I²C bus unit (Master/Slave) [channel]	I²C bus interface (Master only) [channel]	Successive type A/D converter [channel]	Analog comparator [channel]	Analog comparator [input pin]	D/A converter [channel]
ML62Q1530	48	3	1	2	42	41	31	10	6	6	1	2	1	2	12	2	4	1
ML62Q1531																		
ML62Q1532																		
ML62Q1533																		
ML62Q1534																		
ML62Q1540	52				46	45												
ML62Q1541																		
ML62Q1542																		
ML62Q1543																		
ML62Q1544																		
ML62Q1550	64				58	57												
ML62Q1551																		
ML62Q1552																		
ML62Q1553																		
ML62Q1554																		
ML62Q1555																		
ML62Q1556																		
ML62Q1557																		
ML62Q1858																		
ML62Q1859																		
ML62Q1563	80	4			72	71	43	12	8	8		6			16			2
ML62Q1564																		
ML62Q1565																		
ML62Q1566																		
ML62Q1567																		
ML62Q1868																		
ML62Q1869																		
ML62Q1573	100				92	91												
ML62Q1574																		
ML62Q1575																		
ML62Q1576																		
ML62Q1577																		
ML62Q1878																		
ML62Q1879																		

*1 : One 16bit timer is configurable as two 8bit timers
*2 : Synchronous Communication unit includes UART and Synchronous Serial Port. UART and Synchronous Serial Port can not be used at the same time in the same channel.
*3 : Shared with pins for crystal oscillation

ML62Q1700 Group Main Function List

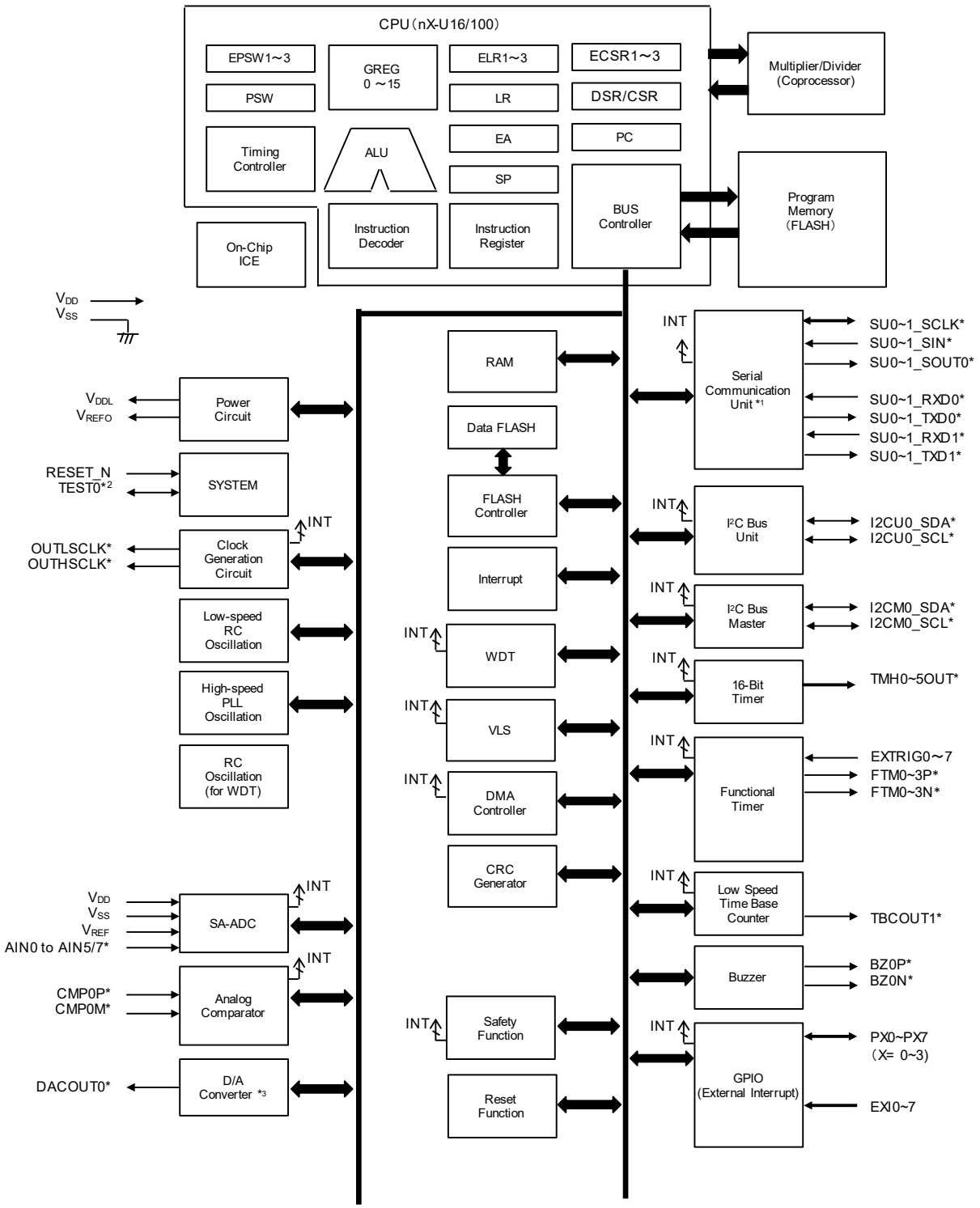
Table 1-6 Main Function List

Part number	Pin										Interrupt		Timer			communication			Analog			
	Total pin-count	Power pin count	Reset Input pin	Input pin count *3	I/O pin count	LED drive port (shared with the I/O port)	LCD common/segment shared pin *4 *5	LCD common pin *5	LCD segment pin *5	LCD bias pin	Internal interrupt [source]	External interrupt [port]	Functional Timer [channel]	16bit General I Timer *1 [channel]	Simplified RTC [channel]	Serial communication unit (Full-duplex UART or Synchronous serial) [channel]*2	I2C bus unit (Master/Slave) [channel]	I2C bus interface (Master only) [channel]	10bit Successive type A/D converter [channel]	Analog comparator [channel]	Analog comparator [input pin]	8bit D/A converter [channel]
ML62Q1700																						
ML62Q1701																						
ML62Q1702	48				37	36			24													
ML62Q1703																						
ML62Q1704																						
ML62Q1710																						
ML62Q1711																						
ML62Q1712	52				41	40			27													
ML62Q1713																						
ML62Q1714		3									31	10	6	6		2			12			1
ML62Q1720																						
ML62Q1721																						
ML62Q1722																						
ML62Q1723																						
ML62Q1724	64				53	52			35													
ML62Q1725																						
ML62Q1726			1	2			5	3		5					1		1	2		2	4	
ML62Q1727																						
ML62Q1728																						
ML62Q1729																						
ML62Q1733																						
ML62Q1734																						
ML62Q1735																						
ML62Q1736	80				67	66			45													
ML62Q1737																						
ML62Q1738																						
ML62Q1739		4									43	12	8	8		6			16			2
ML62Q1743																						
ML62Q1744																						
ML62Q1745																						
ML62Q1746	100				87	86			60													
ML62Q1747																						
ML62Q1748																						
ML62Q1749																						

*1 : One 16bit timer is configurable as two 8bit timers
*2 : Synchronous Communication unit includes UART and Synchronous Serial Port. UART and Synchronous Serial Port can not be used at the same time in the same channel.
*3 : Shared with pins for crystal oscillation
*4 : The LCD common/segment shared pins are shared for common or segment, selectable by setting a SFR
*5 : All LCD drive pins are shared with general purpose I/O ports.

1.2 BLOCK DIAGRAM

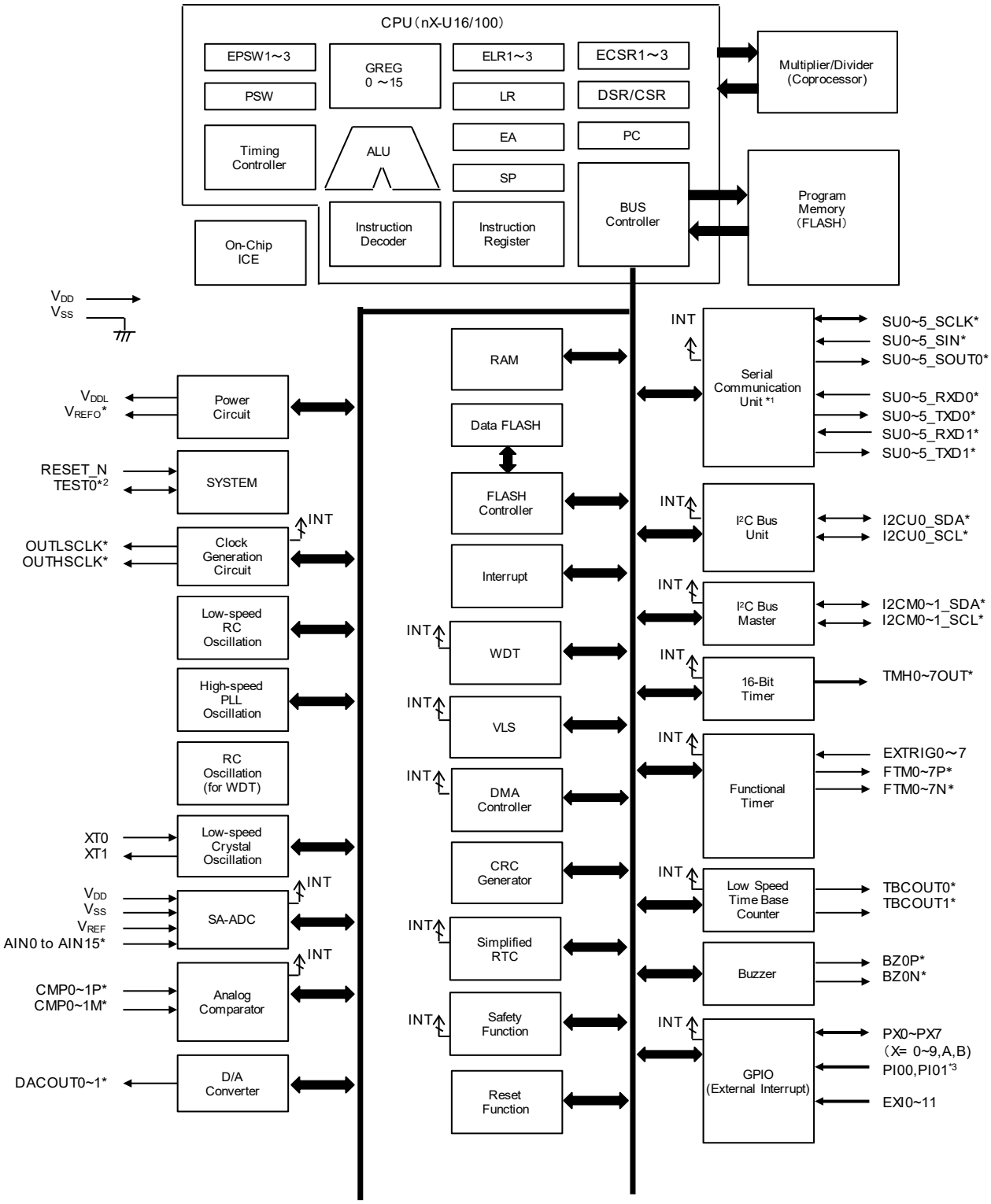
1.2.1 Block Diagram of ML62Q1300 Group



* : Indicates the shared function of general ports.
*1 : Shared UART and Synchronous Serial Port.
*2 : Not available as the input port when connecting to the on-chip debug emulator.
*3 : ML62Q133x and ML62Q132x does not have the peripheral circuits.

Figure 1-4 ML62Q1300 Group Block Diagram

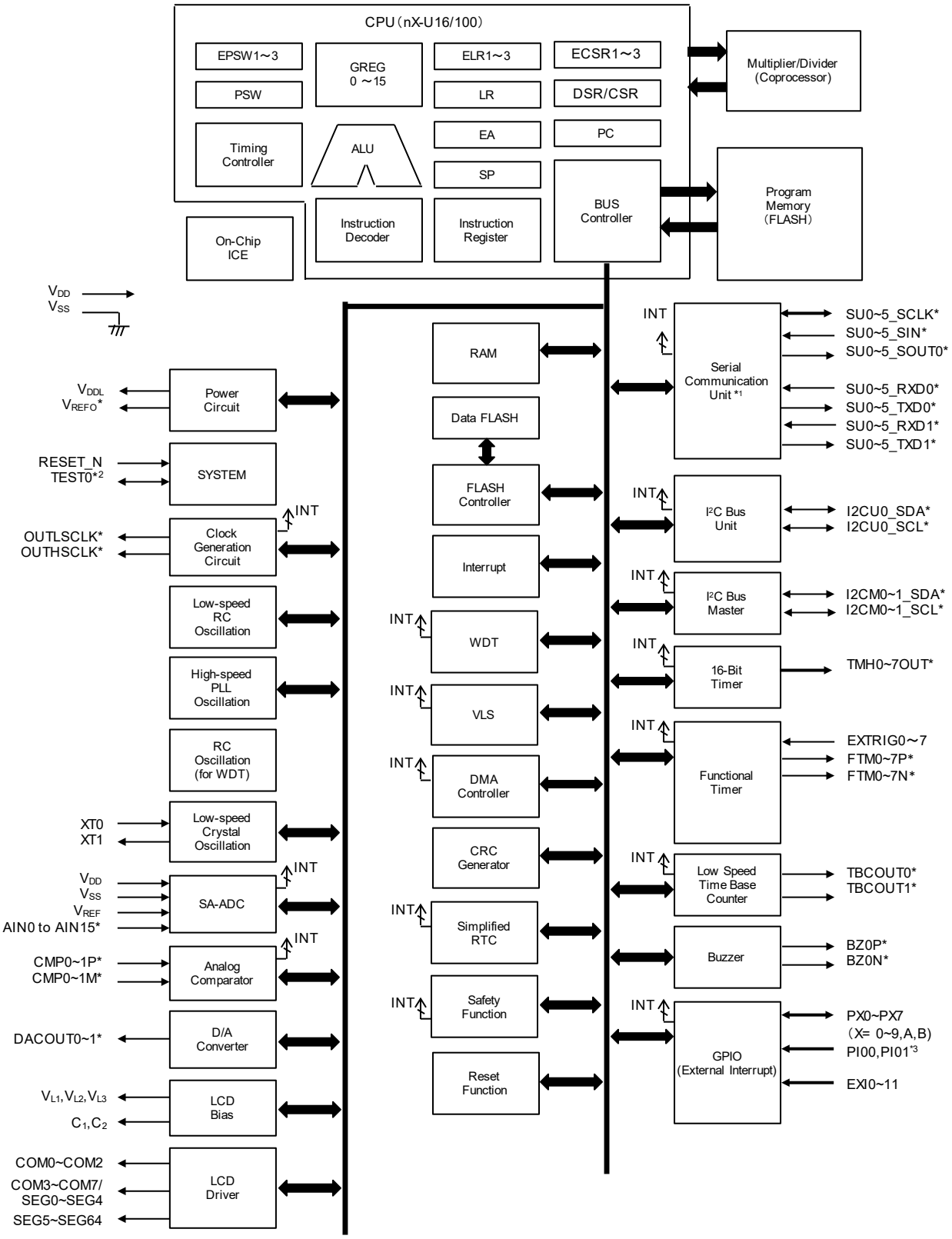
1.2.2 Block Diagram of ML62Q1500/ML62Q1800 Group



* : Indicates the shared function of general ports.
*1 : Shared UART and Synchronous Serial Port.
*2 : Not available as the input port when connecting to the on-chip debug emulator.
*3 : Not available as the input port when connecting to the crystal resonator.

Figure 1-5 ML62Q1500/ML62Q1800 Group Block Diagram

1.2.3 Block Diagram of ML62Q1700 Group



* : Indicates the shared function of general ports.
*1 : Shared UART and Synchronous Serial Port.
*2 : Not available as the input port when connecting to the on-chip debug emulator.
*3 : Not available as the input port when connecting to the crystal resonator.

Figure 1-6 ML62Q1700 Group Block Diagram

1.3 PIN

1.3.1 Pin Layout

1.3.1.1 ML62Q1323/1324/1325 16pin SSOP

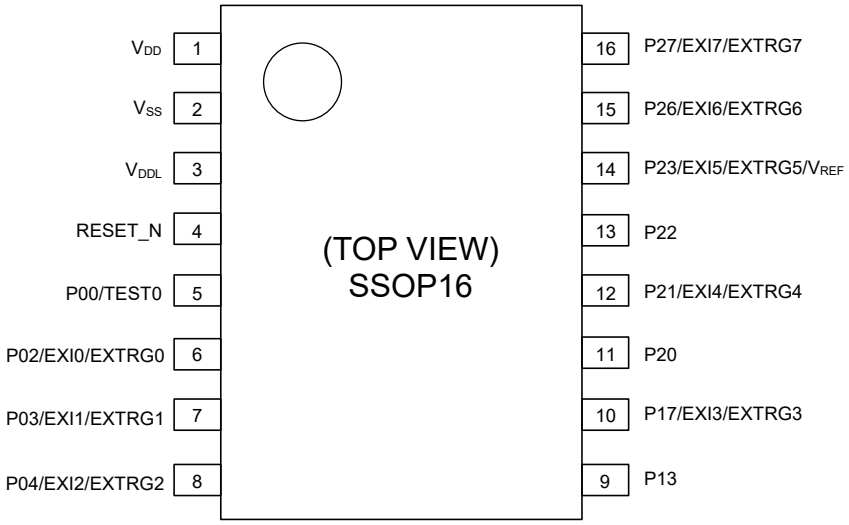


Figure 1-7 Pin Layout of ML62Q1323/1324/1325 16pin SSOP

1.3.1.2 ML62Q1323/1324/1325 16pin WQFN

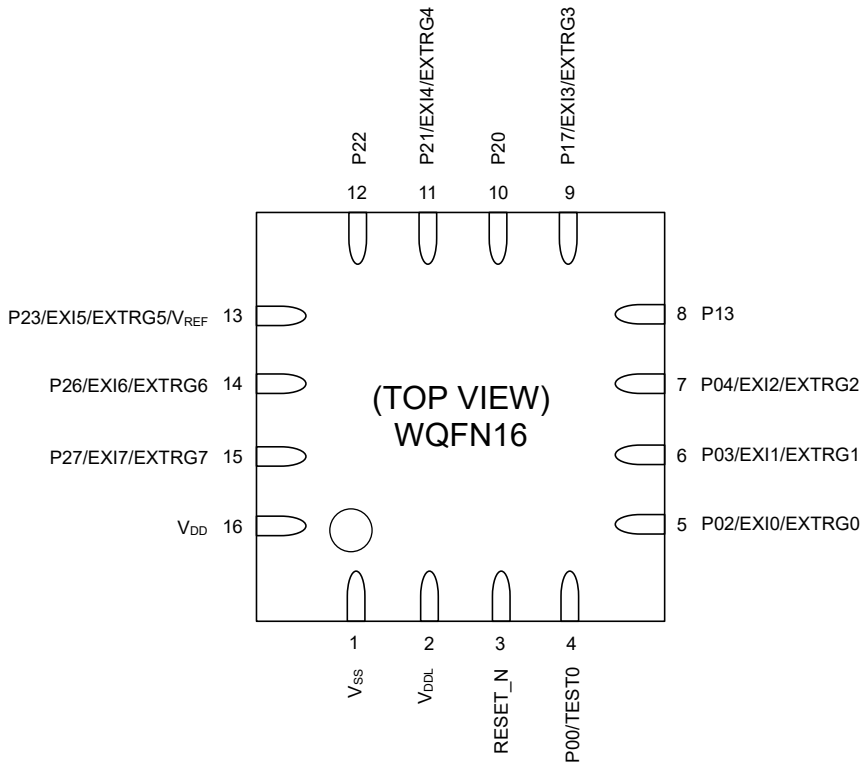


Figure 1-8 Pin Layout of ML62Q1323/1324/1325 16pin WQFN

1.3.1.3 ML62Q1333/1334/1335 20pin TSSOP

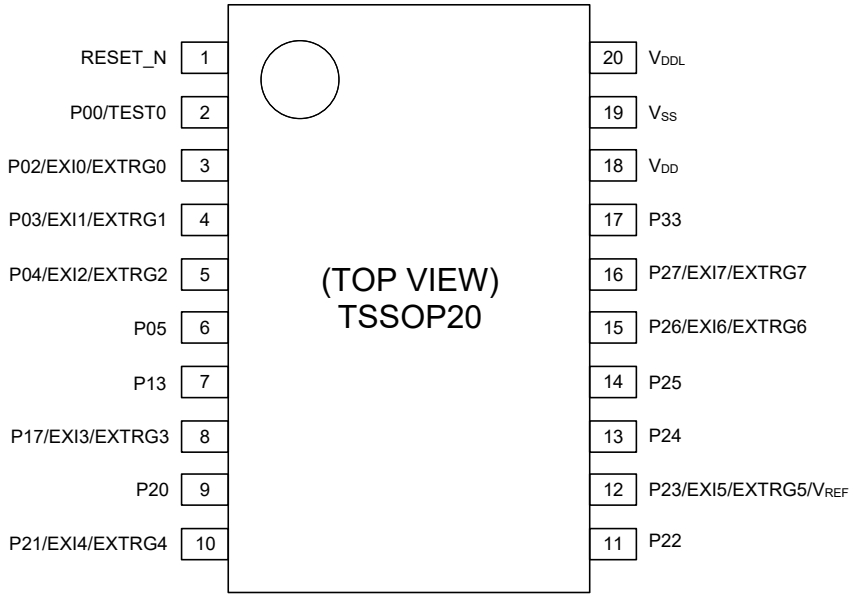


Figure 1-9 Pin Layout of ML62Q1333/1334/1335 20pin TSSOP

1.3.1.4 ML62Q1345/1346/1347 24pin WQFN

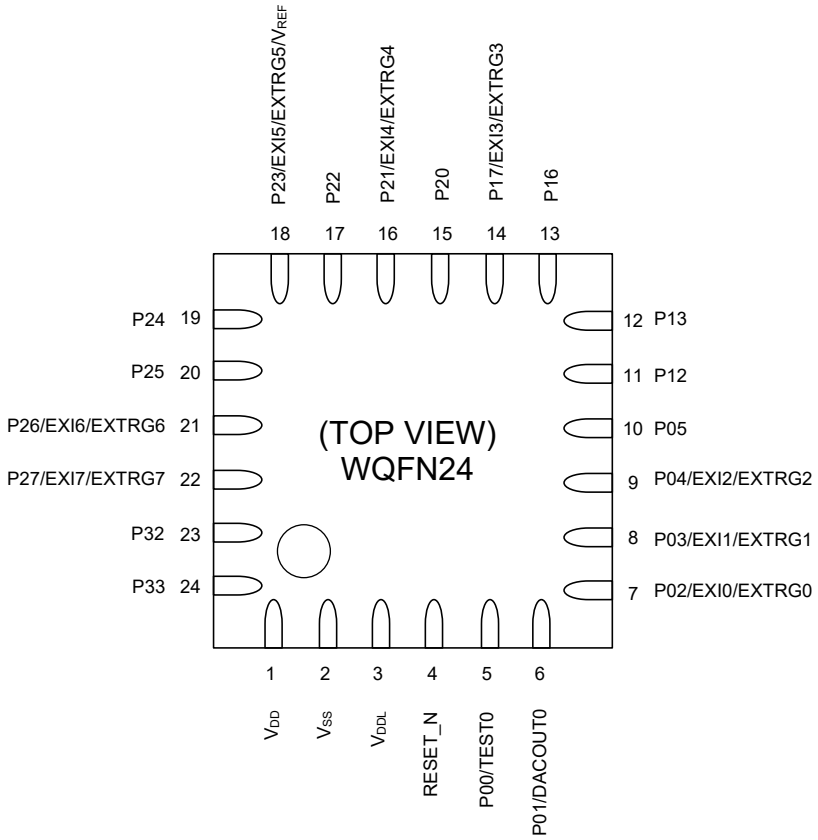


Figure 1-10 Pin Layout of 1345/1346/1347 24pin WQFN

1.3.1.5 ML62Q1365/1366/1367 32pin TQFP

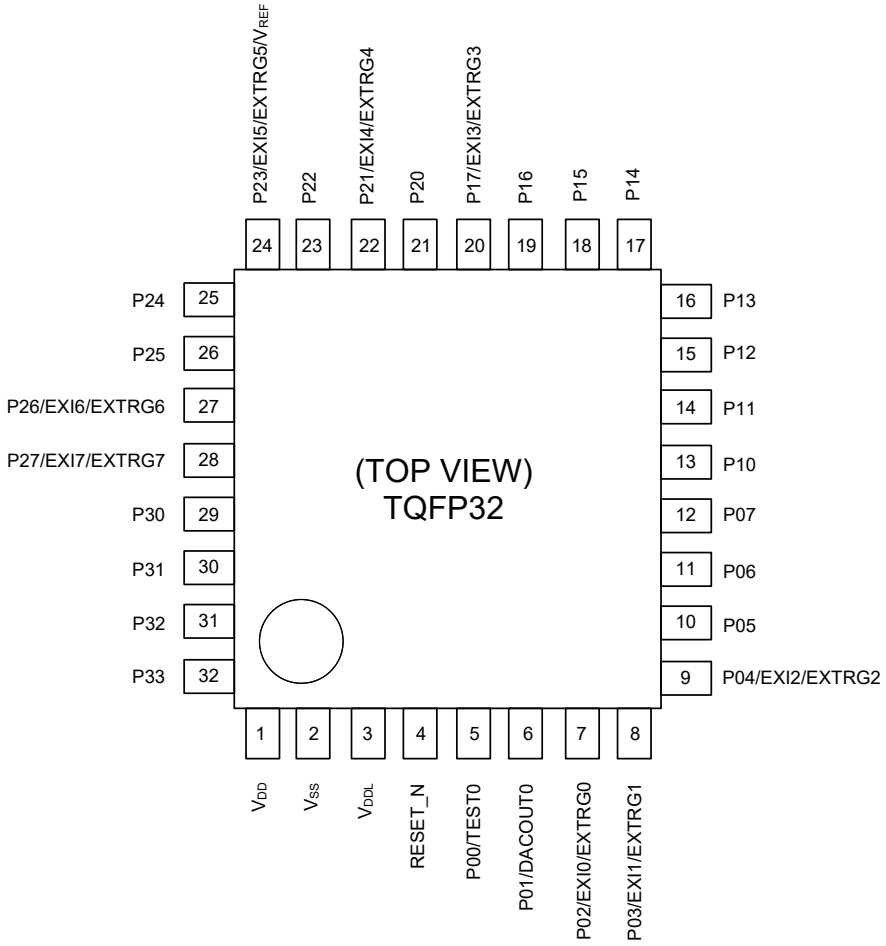


Figure 1-11 Pin Layout of ML62Q1365/1366/1367 32pin TQFP

1.3.1.6 ML62Q1365/1366/1367 32pin WQFN

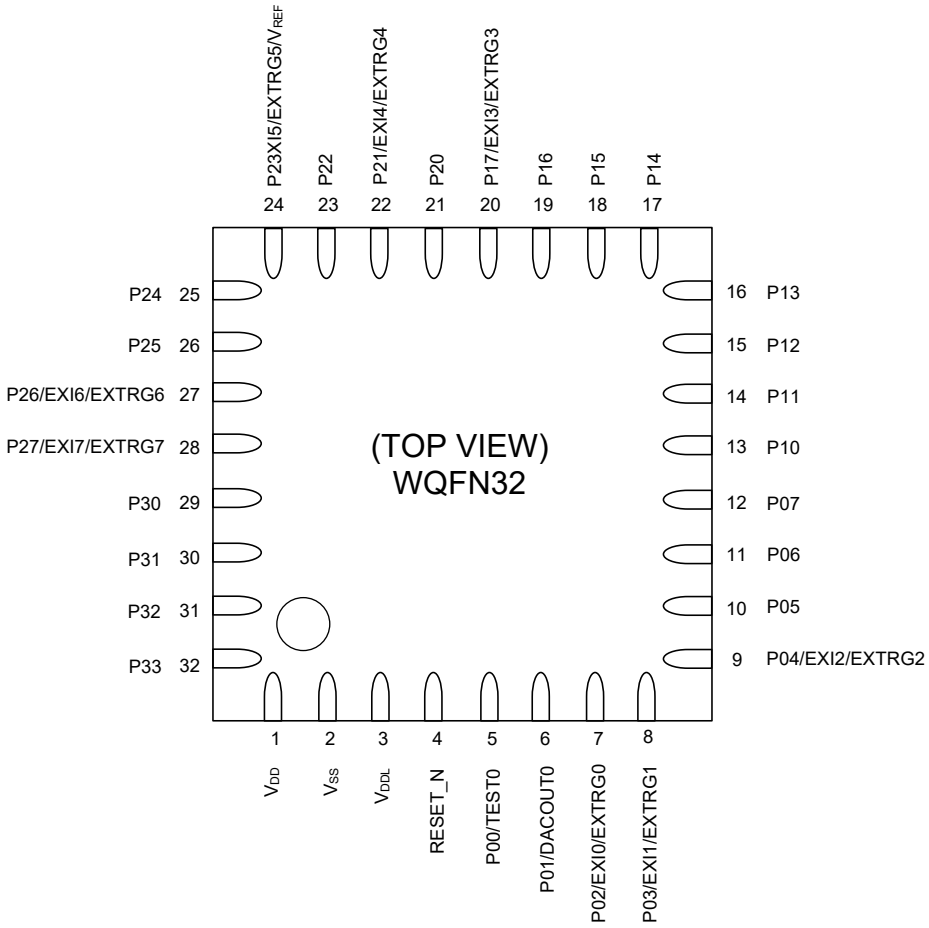


Figure 1-12 Pin Layout of ML62Q1365/1366/1367 32pin WQFN

1.3.1.7 ML62Q1530/1531/1532/1533/1534 48pin TQFP

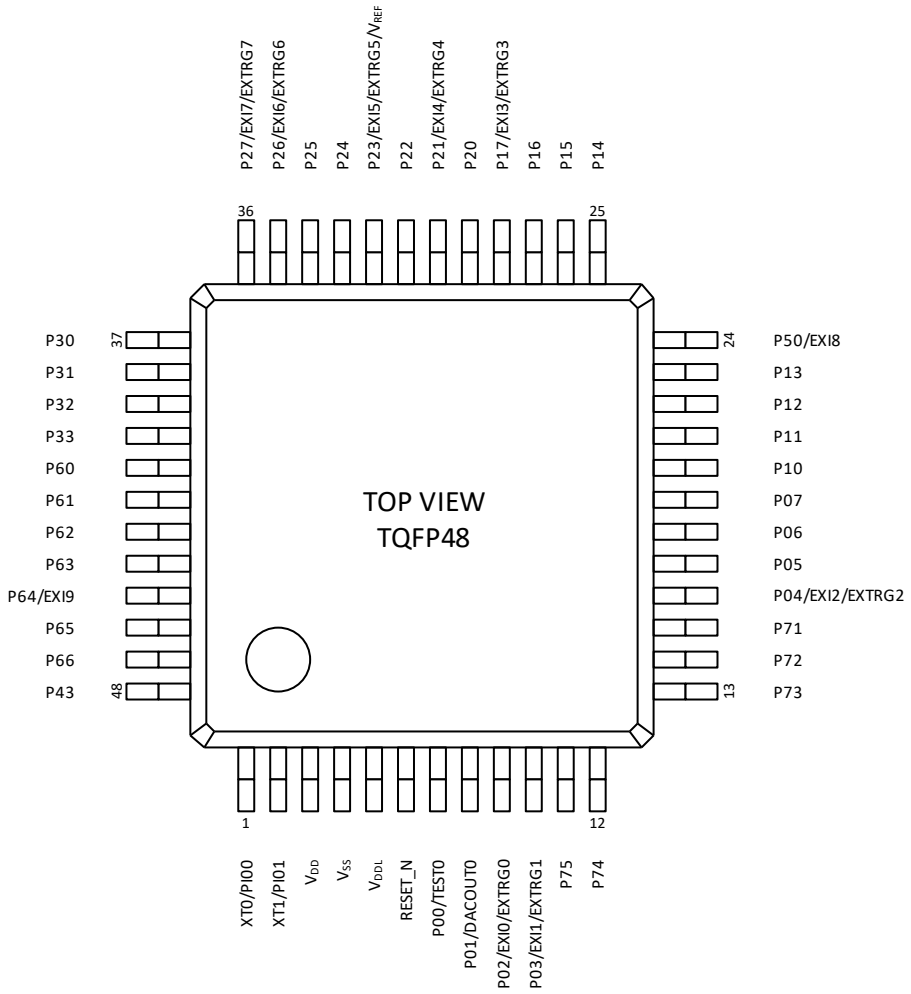


Figure 1-13 ML62Q1530/1531/1532/1533/1534 48pin TQFP

1.3.1.8 ML62Q1540/1541/1542/1543/1544 52pin TQFP

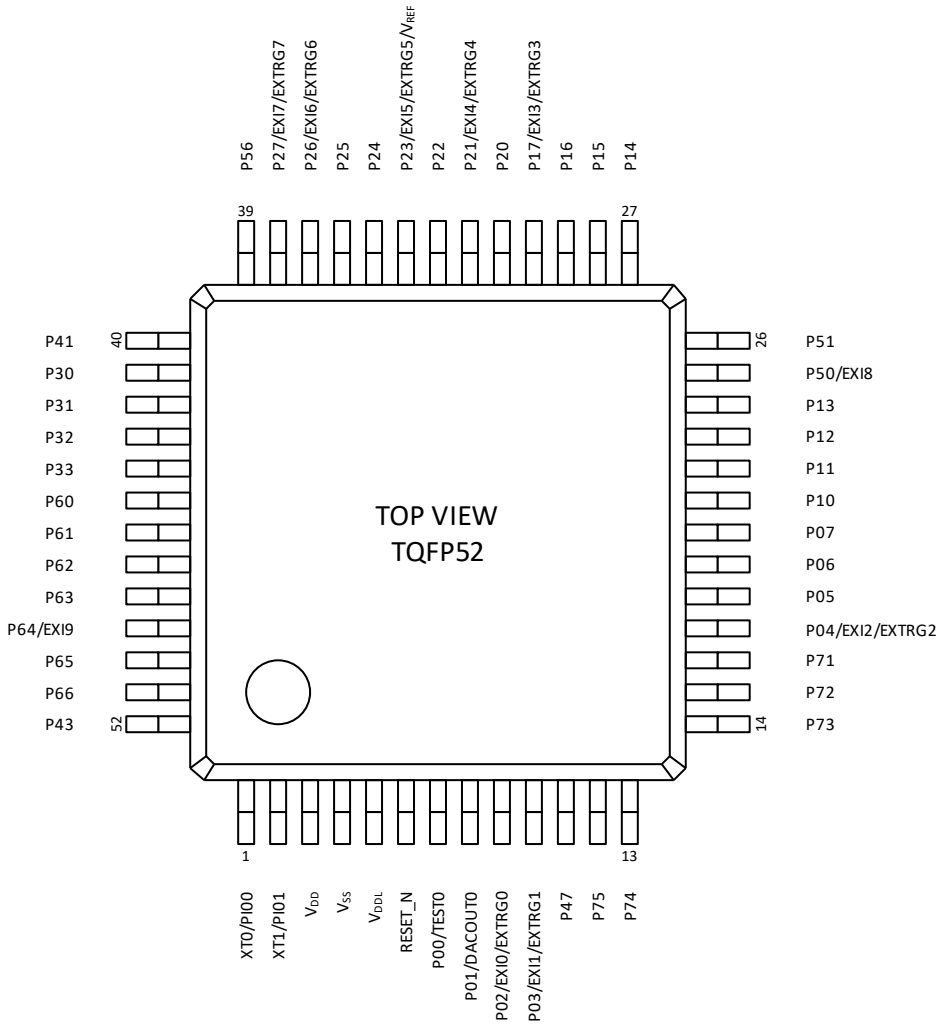


Figure 1-14 ML62Q1540/1541/1542/1543/1544 52pin TQFP

1.3.1.9 ML62Q1550/1551/1552/1553/1554/1555/1556/1557/1858/1859 64pin TQFP/QFP

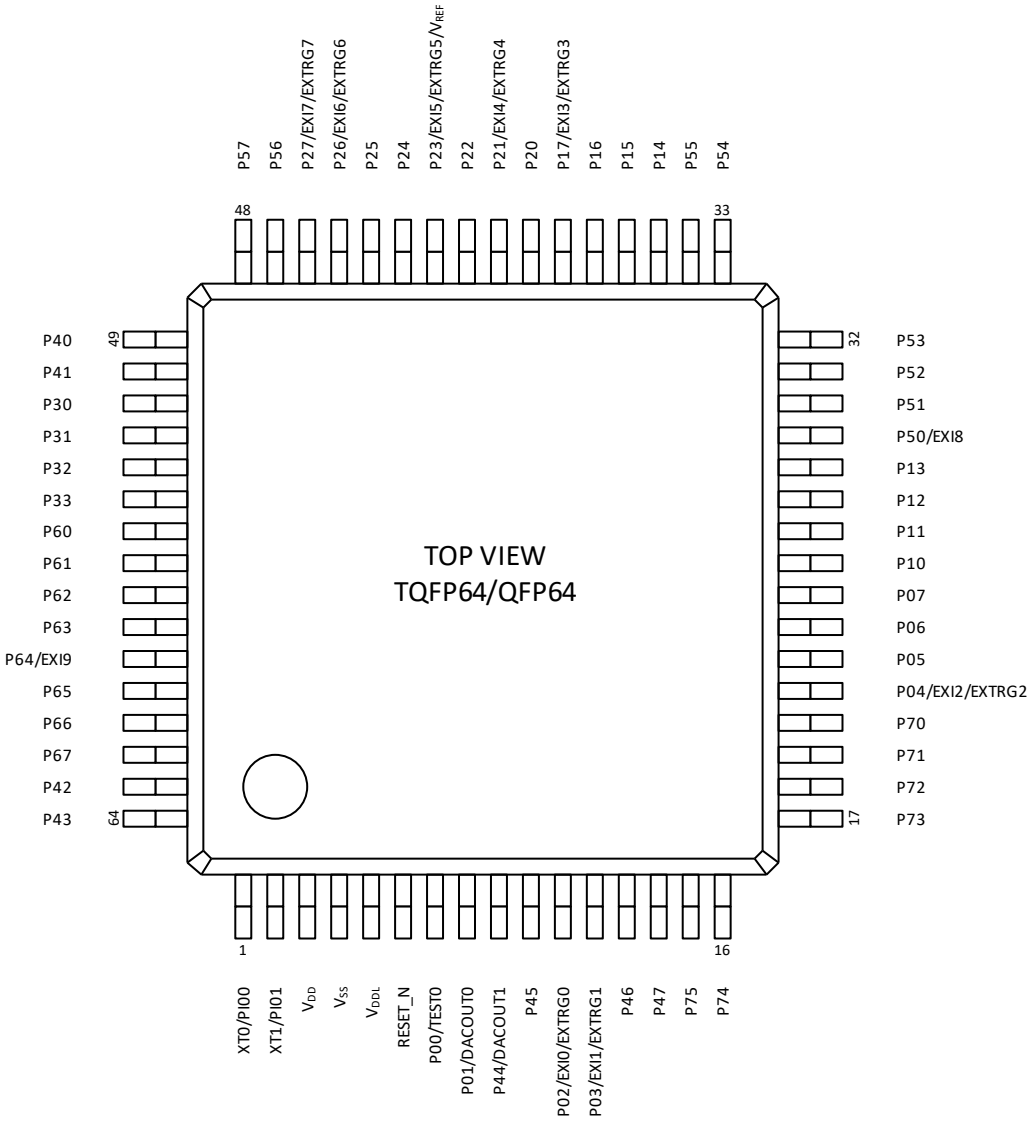


Figure 1-15 ML62Q1550/1551/1552/1553/1554/1555/1556/1557/1858/1859 64pin TQFP/QFP

1.3.1.10 ML62Q1563/1564/1565/1566/1567/1868/1869 80pin QFP

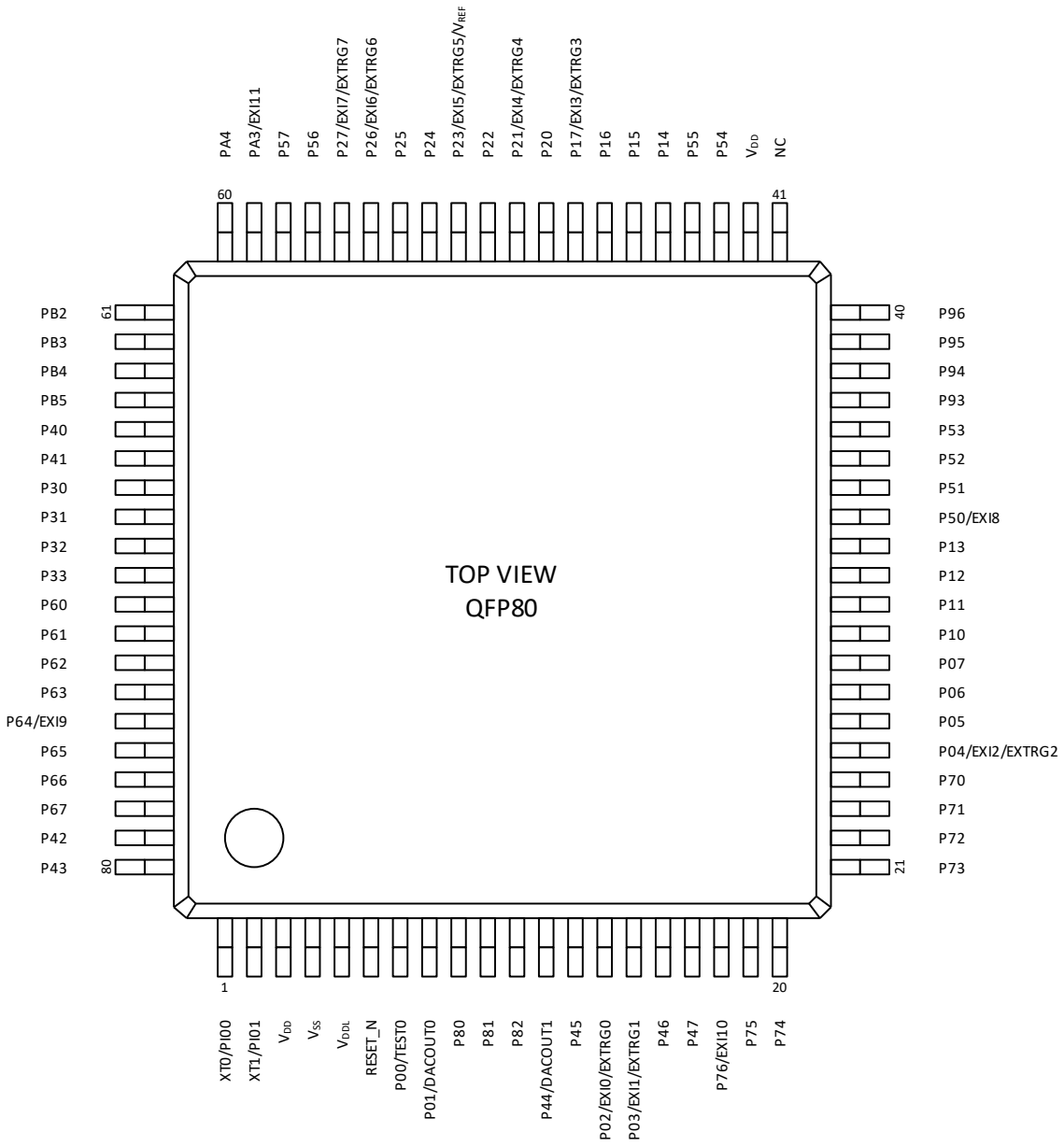


Figure 1-16 ML62Q1563/1564/1565/1566/1567/1868/1869 80pin TQFP/QFP

1.3.1.11 ML62Q1573/1574/1575/1576/1577/1878/1879 100pin TQFP

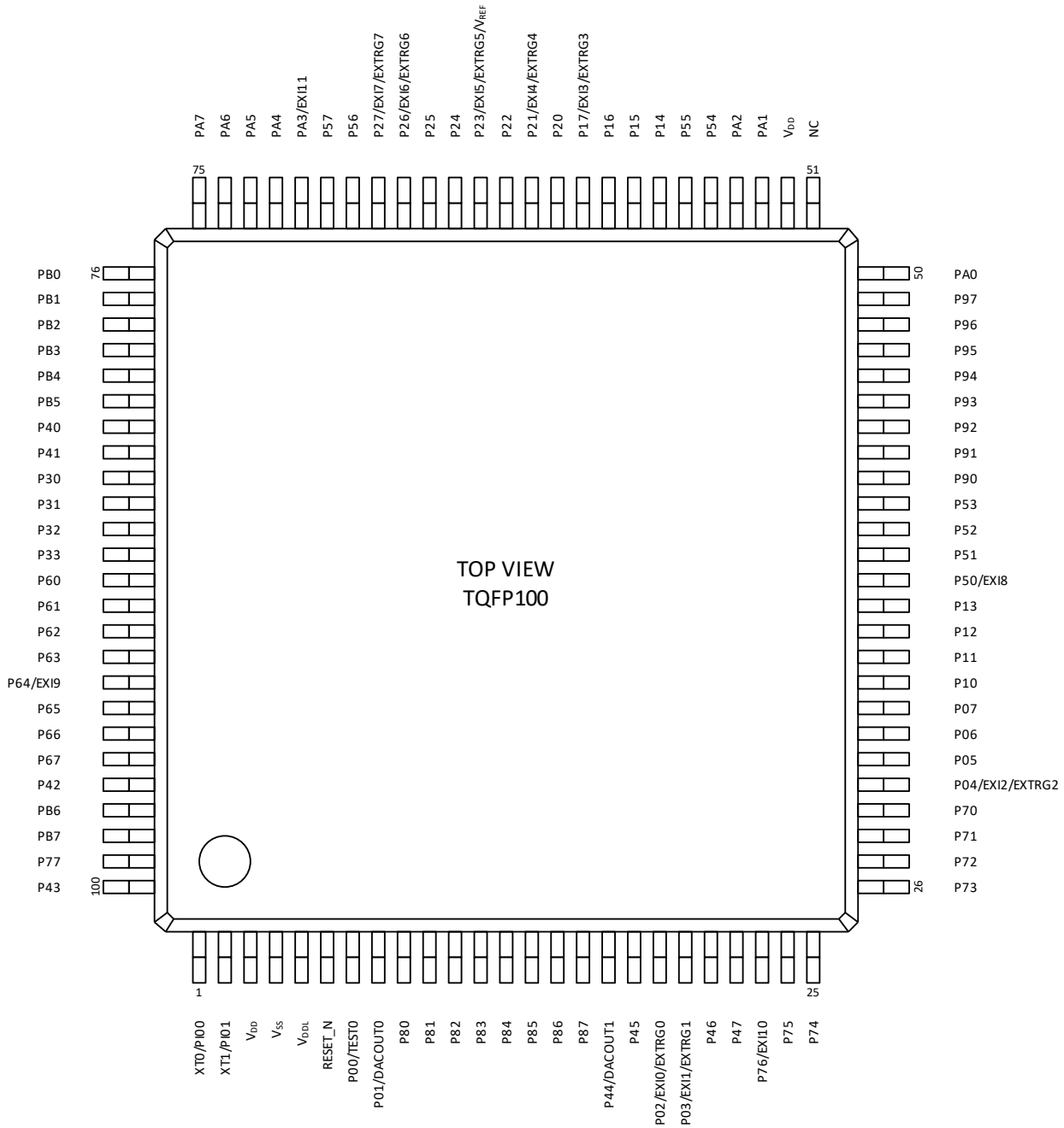


Figure 1-17 ML62Q1573/1574/1575/1576/1577/1878/1879 100pin TQFP

1.3.1.12 ML62Q1573/1574/1575/1576/1577/1878/1879 100pin QFP

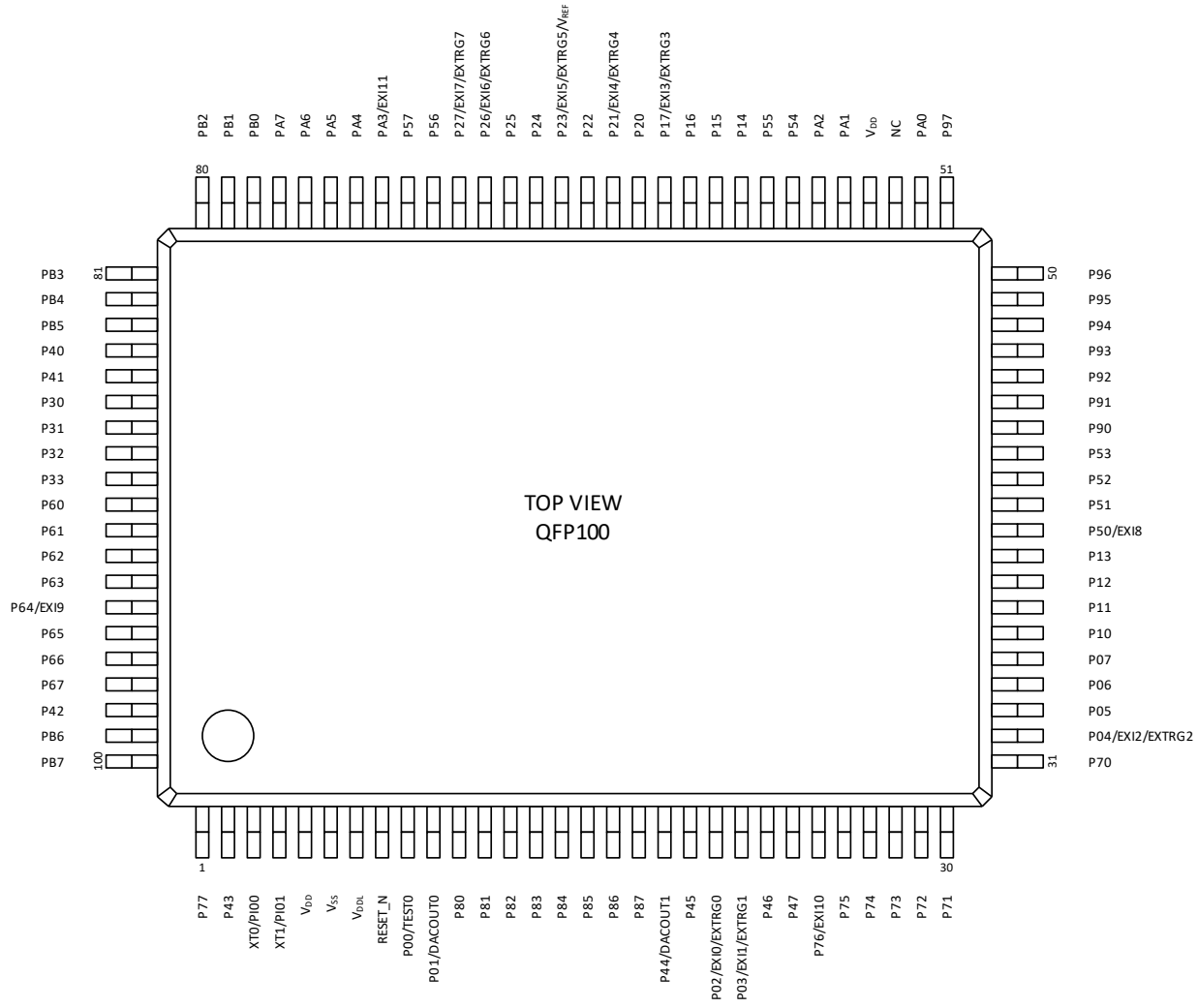


Figure 1-18 ML62Q1573/1574/1575/1576/1577/1878/1879 100pin QFP

1.3.1.13 ML62Q1700/1701/1702/1703/1704 48pin TQFP

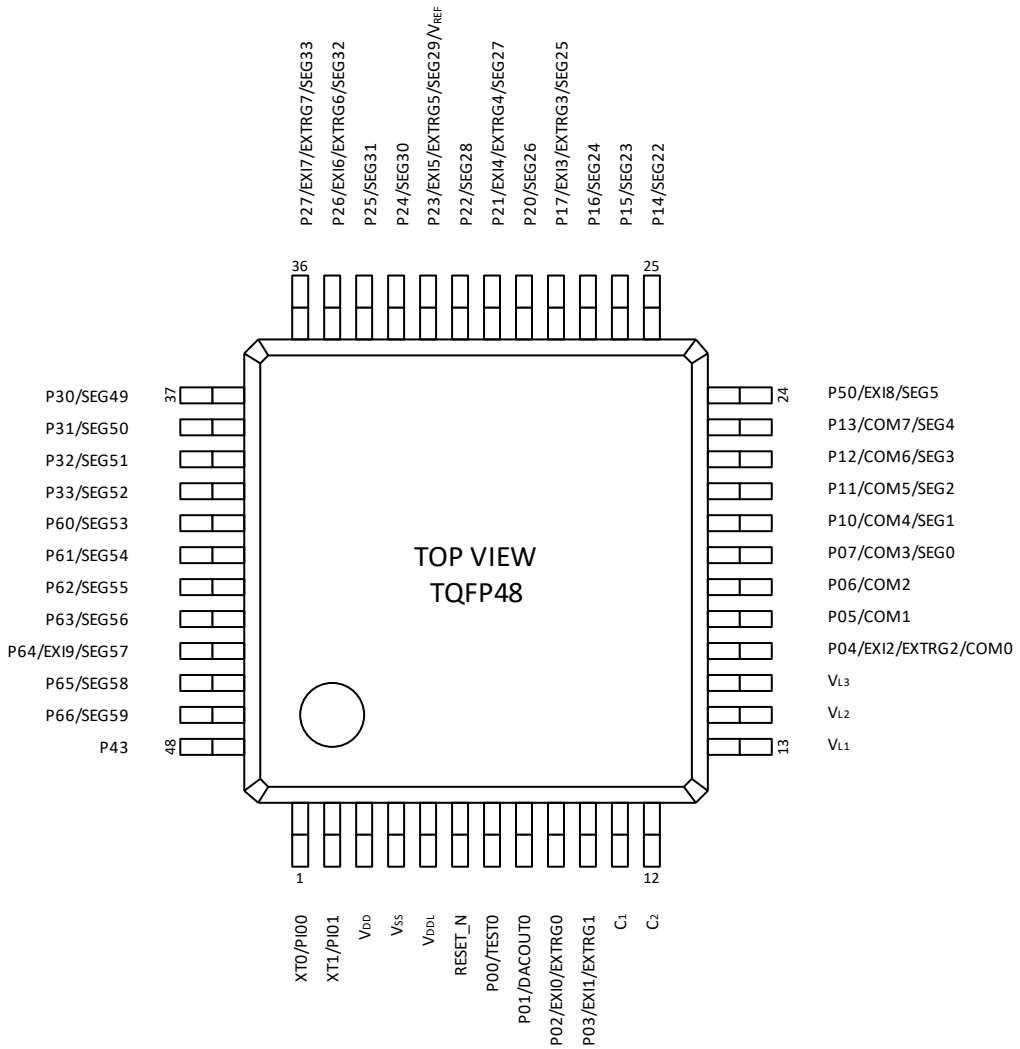


Figure 1-19 ML62Q1700/1701/1702/1703/1704 48pin TQFP

1.3.1.14 ML62Q1710/1711/1712/1713/1714 52pin TQFP

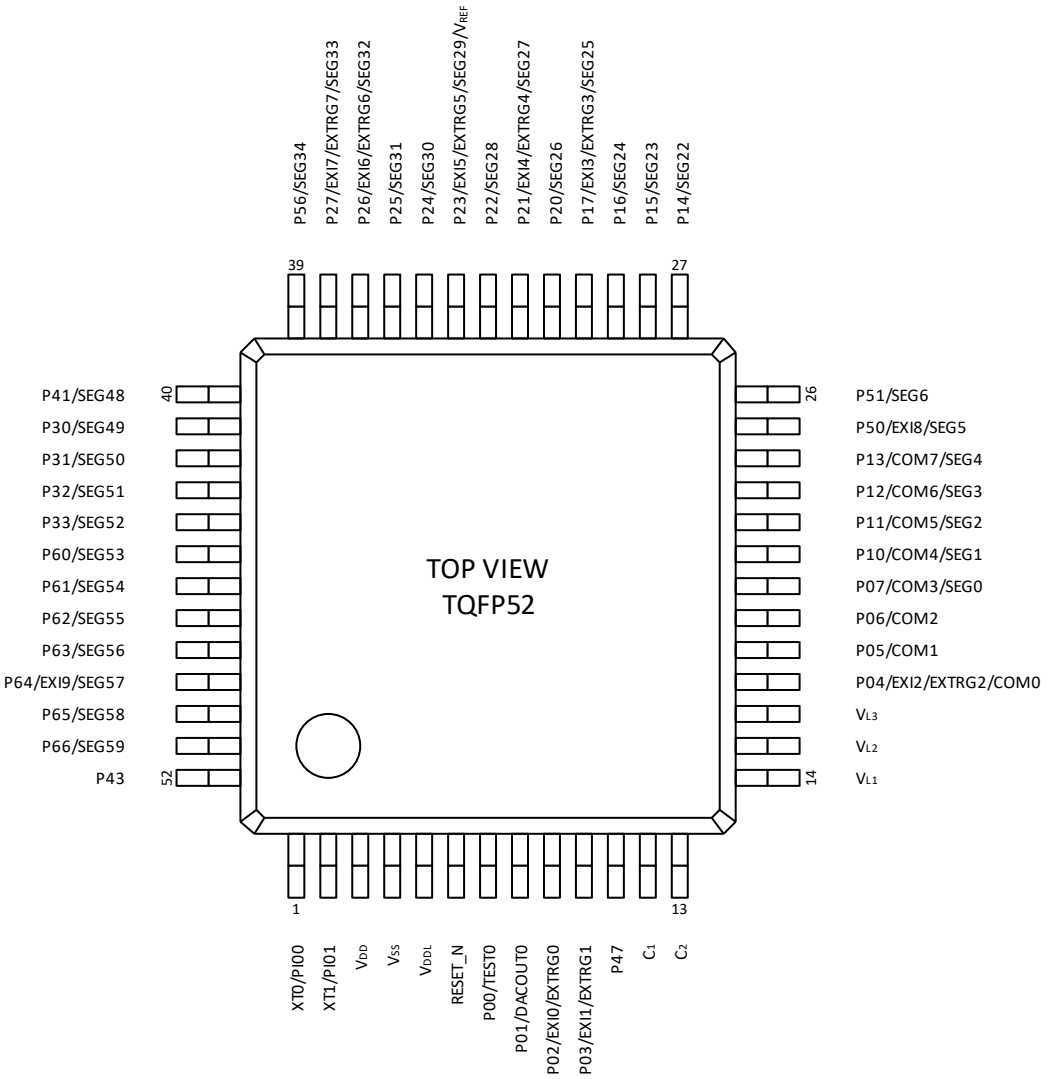


Figure 1-20 ML62Q1710/1711/1712/1713/1714 52pin TQFP

1.3.1.15 ML62Q1720/1721/1722/1723/1724/1725/1726/1727/1728/1729 64pin TQFP/QFP

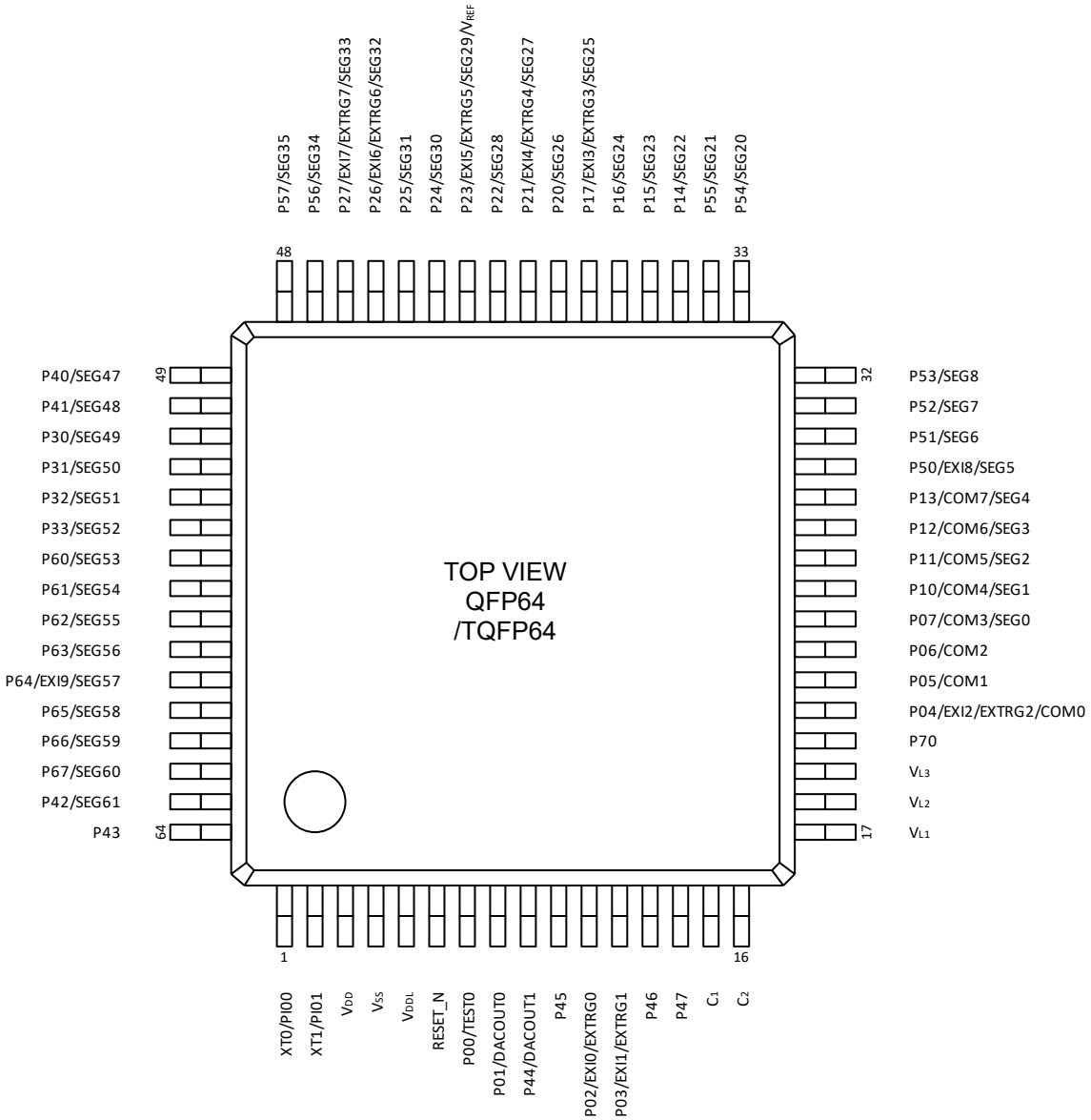


Figure 1-21 ML62Q1720/1721/1722/1723/1724/1725/1726/1727/1728/1729 64pin TQFP/QFP

1.3.1.16 ML62Q1733/1734/1735/1736/1737/1738/1739 80pin QFP

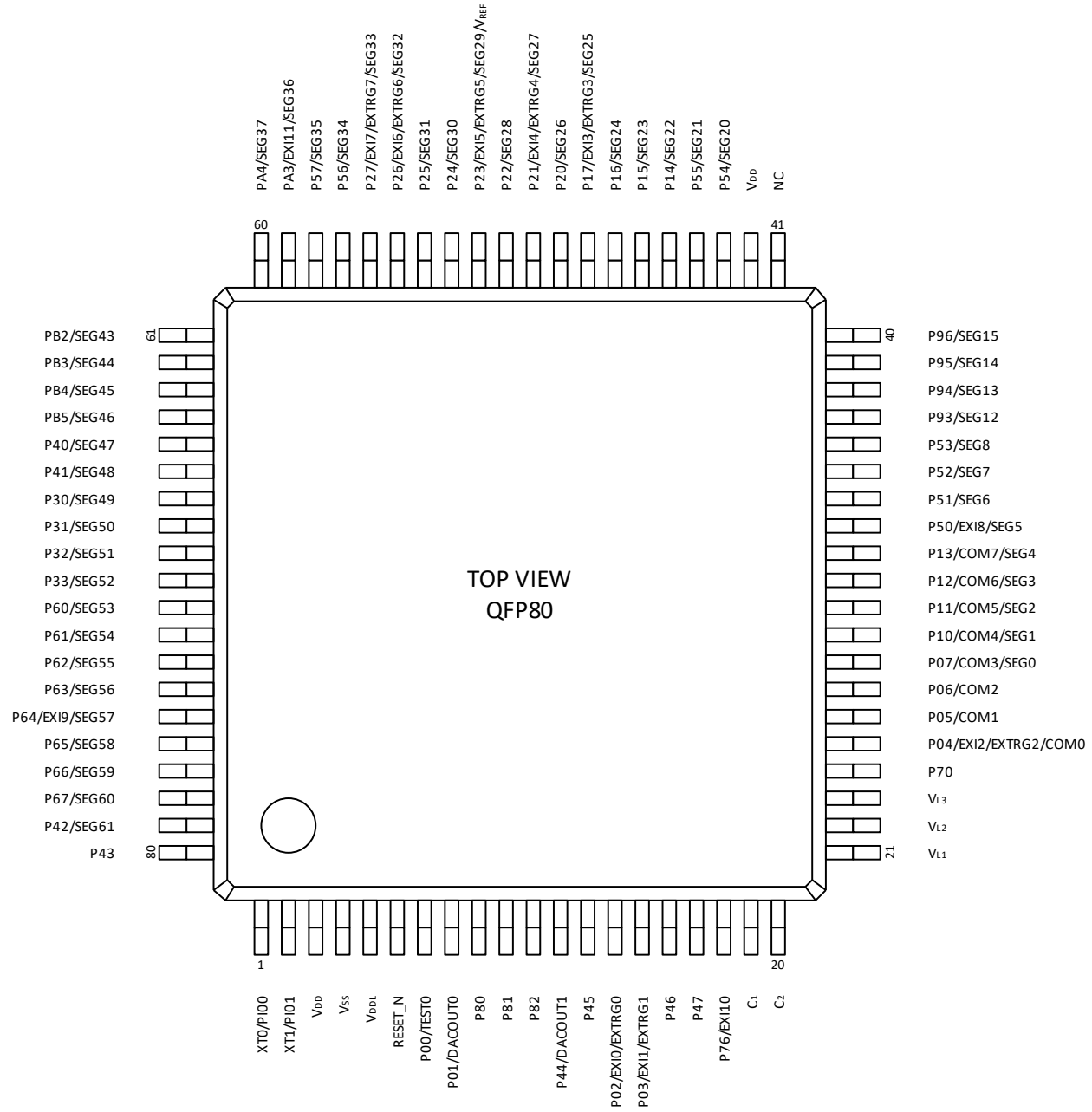


Figure 1-22 ML62Q1733/1734/1735/1736/1737/1738/1739 80pin TQFP/QFP

1.3.1.17 ML62Q1743/1744/1745/1746/1747/1748/1749 100pin TQFP

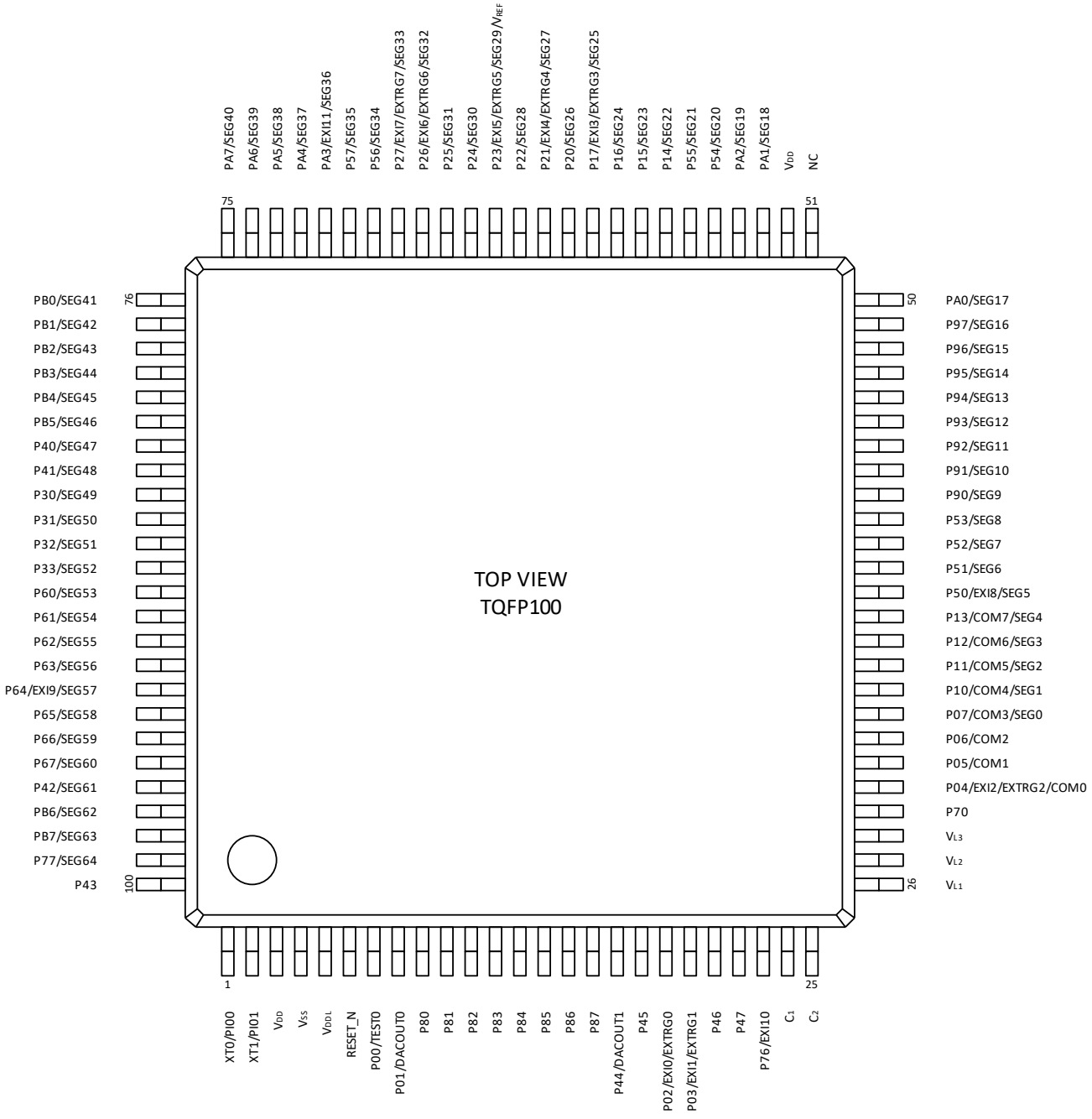


Figure 1-23 ML62Q1743/1744/1745/1746/1747/1748/1749 100pin TQFP

1.3.1.18 ML62Q1743/1744/1745/1746/1747/1748/1749 100pin QFP

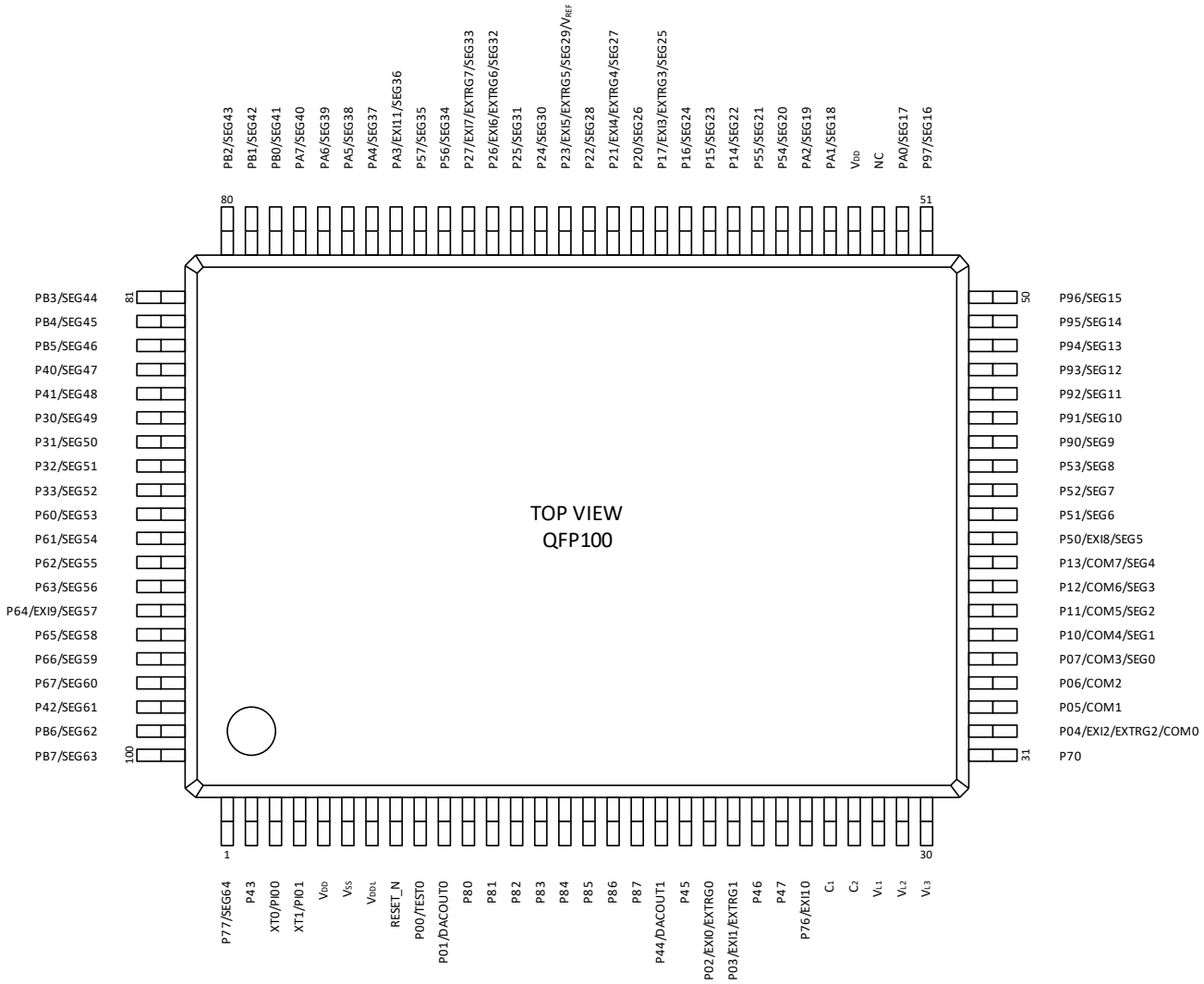


Figure 1-24 ML62Q1743/1744/1745/1746/1747/1748/1749 100pin QFP

1.3.2 PIN LIST

Table 1-7 shows the pin lists of ML62Q1300 group. "(I)" indicates the input pin and "(I/O)" indicates the input/output pin.

Table 1-7 ML62Q1300 Group Pin List

Pin No.					Pin name (Primary function)	Primary function Others	2nd function communications	3rd function communications	4th function communications	5th function Timers	6th function Others	7th function Others	8th function ADC
SSOP16	WQFN16	TSSOP20	WQFN24	TQFP32/WQFN32									
1	16	18	1	1	V_{DD}	-	-	-	-	-	-	-	-
2	1	19	2	2	V_{SS}	-	-	-	-	-	-	-	-
3	2	20	3	3	V_{DDL}	-	-	-	-	-	-	-	-
4	3	1	4	4	RESET_N	-	-	-	-	-	-	-	-
5	4	2	5	5	P00	TEST0	-	-	-	-	-	-	-
-	-	-	6	6	P01	DACOUT0	-	-	-	-	-	-	-
6	5	3	7	7	P02	EXI0 EXTRG0	SU0_RXD0 SU0_SIN	-	-	FTM0P	OUTLSCLK	CMP0M	-
7	6	4	8	8	P03	EXI1 EXTRG1	SU0_TXD0 SU0_SOUT	SU0_TXD1	I2CU0_SDA	FTM0N	OUTHSCLK	CMP0P	-
8	7	5	9	9	P04	EXI2 EXTRG2	SU0_SCLK	-	I2CU0_SCL	TMH0OUT	-	-	-
-	-	6	10	10	P05	-	-	-	-	-	-	-	-
-	-	-	-	11	P06	-	-	-	I2CM0_SDA	-	-	-	-
-	-	-	-	12	P07	-	SU0_RXD1	SU0_RXD0	I2CM0_SCL	-	-	-	-
-	-	-	-	13	P10	-	SU0_TXD1	-	-	-	-	-	-
-	-	-	-	14	P11	-	SU0_SCLK	-	-	-	-	-	-
-	-	-	11	15	P12	-	SU0_RXD0 SU0_SIN	-	-	TMH4OUT	-	-	-
9	8	7	12	16	P13	-	SU0_TXD0 SU0_SOUT	SU0_TXD1	-	TMH1OUT	-	TMH3OUT	-
-	-	-	-	17	P14	-	-	-	-	-	-	-	-
-	-	-	-	18	P15	-	-	-	I2CU0_SDA	-	-	-	-
-	-	-	13	19	P16	-	SU1_SCLK	-	I2CU0_SCL	TMH5OUT	-	-	-
10	9	8	14	20	P17	EXI3 EXTRG3	SU0_RXD1	SU0_RXD0	-	FTM1P	-	BZ0P	AIN0
11	10	9	15	21	P20		SU0_TXD1	-	-	FTM1N	TBCOUT1	BZ0N	AIN1
12	11	10	16	22	P21	EXI4 EXTRG4	SU1_RXD0 SU1_SIN	-	-	FTM2P	OUTLSCLK	-	AIN2
13	12	11	17	23	P22	-	SU1_TXD0 SU1_SOUT	SU1_TXD1	I2CM0_SDA	FTM2N	OUTHSCLK	-	AIN3
14	13	12	18	24	P23	EXI5 EXTRG5 V_{REF}	SU1_SCLK	-	I2CM0_SCL	TMH2OUT	-	-	V_{REFO}
-	-	13	19	25	P24	-	SU1_RXD0 SU1_SIN	-	-	-	-	-	AIN4
-	-	14	20	26	P25	-	SU1_TXD0 SU1_SOUT	SU1_TXD1	-	-	-	-	AIN5
15	14	15	21	27	P26	EXI6 EXTRG6	SU1_RXD1	SU1_RXD0	I2CU0_SDA	FTM3P	-	BZ0P	AIN6
16	15	16	22	28	P27	EXI7 EXTRG7	SU1_TXD1	-	I2CU0_SCL	FTM3N	TBCOUT1	BZ0N	AIN7
-	-	-	-	29	P30	-	-	-	-	-	-	-	-
-	-	-	-	30	P31	-	-	-	-	-	-	-	-
-	-	-	23	31	P32	-	SU1_RXD1	SU1_RXD0	-	-	-	-	-
-	-	17	24	32	P33	-	SU1_TXD1	-	-	TMH3OUT	-	-	-

Table 1-8 shows the pin lists of ML62Q1500/ML62Q1800 group. "(I)" indicates the input pin and "(I/O)" indicates the input/output pin.

Table 1-8 ML62Q1500/ML62Q1800 Group Pin List (1/3)

Pin No.						Pin name (Primary function)	Primary function Others	2nd function communications	3rd function communications	4th function communications	5th function Timers	6th function Others	7th function Others	8th function ADC
48Pin	52Pin	64Pin	80Pin	TQFP100	QFP100									
3	3	3	3	3	5	V_{DD}	-	-	-	-	-	-	-	-
-	-	-	42	52	54	V_{DD}	-	-	-	-	-	-	-	-
4	4	4	4	4	6	V_{SS}	-	-	-	-	-	-	-	-
-	-	-	41	51	53	NC	-	-	-	-	-	-	-	-
5	5	5	5	5	7	V_{DDL}	-	-	-	-	-	-	-	-
1	1	1	1	1	3	XT0	PI00	-	-	-	-	-	-	-
2	2	2	2	2	4	XT1	PI01	-	-	-	-	-	-	-
6	6	6	6	6	8	RESET_N	-	-	-	-	-	-	-	-
7	7	7	7	7	9	P00	TEST0	-	-	-	-	-	-	-
8	8	8	8	8	10	P01	DACOUT0	-	-	-	-	TBCOUT0	TBCOUT1	-
9	9	11	14	19	21	P02	EXI0 EXTRG0	SU0_RXD0 SU0_SIN	-	-	FTM0P	OUTLSCLK	CMP0M	-
10	10	12	15	20	22	P03	EXI1 EXTRG1	SU0_TXD0 SU0_SOUT	SU0_TXD1	I2CU0_SDA	FTM0N	OUTHSCLK	CMP0P	AIN11
16	17	21	25	30	32	P04	EXI2 EXTRG2	SU0_SCLK	-	I2CU0_SCL	TMH0OUT	-	-	-
17	18	22	26	31	33	P05	-	-	-	-	-	-	-	-
18	19	23	27	32	34	P06	-	-	-	I2CM0_SDA	-	-	-	-
19	20	24	28	33	35	P07	-	SU0_RXD1	SU0_RXD0	I2CM0_SCL	-	-	-	-
20	21	25	29	34	36	P10	-	SU0_TXD1	-	-	-	-	-	-
21	22	26	30	35	37	P11	-	SU0_SCLK	-	-	-	-	-	-
22	23	27	31	36	38	P12	-	SU0_RXD0 SU0_SIN	-	-	TMH4OUT	-	-	-
23	24	28	32	37	39	P13	-	SU0_TXD0 SU0_SOUT	SU0_TXD1	-	TMH1OUT	-	TMH3OUT	-
25	27	35	45	57	59	P14	-	-	-	-	-	-	-	-
26	28	36	46	58	60	P15	-	-	-	I2CU0_SDA		-	-	-
27	29	37	47	59	61	P16	-	SU1_SCLK	-	I2CU0_SCL	TMH5OUT	-	-	-
28	30	38	48	60	62	P17	EXI3 EXTRG3	SU0_RXD1	SU0_RXD0	-	FTM1P	TBCOUT0	BZ0P	AIN0
29	31	39	49	61	63	P20	-	SU0_TXD1	-	-	FTM1N	TBCOUT1	BZ0N	AIN1
30	32	40	50	62	64	P21	EXI4 EXTRG4	SU1_RXD0 SU1_SIN	-	-	FTM2P	OUTLSCLK	-	AIN2
31	33	41	51	63	65	P22	-	SU1_TXD0 SU1_SOUT	SU1_TXD1	I2CM0_SDA	FTM2N	OUTHSCLK	-	AIN3
32	34	42	52	64	66	P23	EXI5 EXTRG5 V_{REF}	SU1_SCLK	-	I2CM0_SCL	TMH2OUT	-	-	V_{REFO}
33	35	43	53	65	67	P24	-	SU1_RXD0 SU1_SIN	-	-	-	-	-	AIN4
34	36	44	54	66	68	P25	-	SU1_TXD0 SU1_SOUT	SU1_TXD1	-	-	-	-	AIN5
35	37	45	55	67	69	P26	EXI6 EXTRG6	SU1_RXD1	SU1_RXD0	I2CU0_SDA	FTM3P	TBCOUT0	BZ0P	AIN6
36	38	46	56	68	70	P27	EXI7 EXTRG7	SU1_TXD1	-	I2CU0_SCL	FTM3N	TBCOUT1	BZ0N	AIN7

Table 1-8 ML62Q1500/ML62Q1800 Group Pin List (2/3)

Pin No.						Pin name (Primary function)	Primary function Others	2nd function communications*1	3rd function communications*1	4th function communications	5th function Timers*1	6th function Others	7th function Others	8th function ADC*1
48Pin	52Pin	64Pin	80Pin	TQFP100	QFP100									
37	41	51	67	84	86	P30	-	-	-	-	-	-	-	-
38	42	52	68	85	87	P31	-	-	-	-	-	TBCOUT0	TBCOUT1	-
39	43	53	69	86	88	P32	-	SU1_RXD1	SU1_RXD0	-	-	-	-	-
40	44	54	70	87	89	P33	-	SU1_TXD1	-	-	TMH3OUT	-	-	-
-	-	49	65	82	84	P40	-	SU5_TXD1	-	-	-	-	-	-
-	40	50	66	83	85	P41	-	-	-	-	-	-	-	-
-	-	63	79	96	98	P42	-	SU3_TXD1	-	-	-	-	-	-
48	52	64	80	100	2	P43	-	-	-	-	-	TBCOUT0	TBCOUT1	AIN10
-	-	9	12	17	19	P44	DACOUT1	SU4_RXD1	SU4_RXD0	-	-	-	-	-
-	-	10	13	18	20	P45	-	SU4_TXD1		-	-	-	-	-
-	-	13	16	21	23	P46	-	-	-	-	-	-	-	-
-	11	14	17	22	24	P47	-	-	-	-	-	-	-	-
24	25	29	33	38	40	P50	EXI8	-	-	-	-	-	-	-
-	26	30	34	39	41	P51	-	-	-	-	-	-	-	-
-	-	31	35	40	42	P52	-	SU4_RXD1	SU4_RXD0	-	-	-	-	-
-	-	32	36	41	43	P53	-	SU4_TXD1		-	-	-	-	-
-	-	33	43	55	57	P54	-	SU2_RXD1	SU2_RXD0	-	TMH7OUT	-	-	-
-	-	34	44	56	58	P55	-	SU2_TXD1	-	-	-	-	-	-
-	39	47	57	69	71	P56	-	SU2_RXD0 SU2_SIN	-	-	-	-	-	AIN12
-	-	48	58	70	72	P57	-	SU2_TXD0 SU2_SOUT	SU2_TXD1	-	-	-	-	AIN13
41	45	55	71	88	90	P60	-	-	-	I2CM1_SCL	-	-	-	-
42	46	56	72	89	91	P61	-	-	-	I2CM1_SDA	-	-	-	-
43	47	57	73	90	92	P62	-	-	-	-	FTM4N	-	CMP1P	-
44	48	58	74	91	93	P63	-	-	-	-	FTM4P	-	CMP1M	-
45	49	59	75	92	94	P64	EXI9	SU3_RXD0 SU3_SIN	-	-	FTM5P	-	-	-
46	50	60	76	93	95	P65	-	SU3_TXD0 SU3_SOUT	SU3_TXD1	-	FTM5N	-	-	AIN8
47	51	61	77	94	96	P66	-	SU3_SCLK	-	-	-	-	-	AIN9
-	-	62	78	95	97	P67	-	SU3_RXD1	SU3_RXD0	-	-	-	-	-
-	-	20	24	29	31	P70	-	-	-	-	TMH6OUT	-	-	-
15	16	19	23	28	30	P71	-	-	-	-	-	-	-	-
14	15	18	22	27	29	P72	-	-	-	-	-	-	-	-
13	14	17	21	26	28	P73	-	-	-	-	-	-	-	-
12	13	16	20	25	27	P74	-	-	-	-	-	-	-	-
11	12	15	19	24	26	P75	-	-	-	-	-	-	-	-
-	-	-	18	23	25	P76	EXI10	-	-	-	-	-	-	-
-	-	-	-	99	1	P77	-	-	-	-	-	-	-	-
-	-	-	9	9	11	P80	-	SU4_RXD0 SU4_SIN	-	-	-	-	-	-
-	-	-	10	10	12	P81	-	SU4_TXD0 SU4_SOUT	SU4_TXD1	-	-	-	-	-
-	-	-	11	11	13	P82	-	SU4_SCLK	-	-	-	-	-	-
-	-	-	-	12	14	P83	-	-	-	-	-	-	-	-
-	-	-	-	13	15	P84	-	-	-	-	-	-	-	-
-	-	-	-	14	16	P85	-	-	-	-	-	-	-	-
-	-	-	-	15	17	P86	-	-	-	-	-	-	-	-
-	-	-	-	16	18	P87	-	-	-	-	-	-	-	-

*1: The pins of name with DACOUT1, SU2, SU3, SU4, SU5, TMH6, TMH7, AIN12 or AIN13 are not assigned to products of 48/52/64 PIN-packages.

Table 1-8 ML62Q1500/ML62Q1800 Group Pin List (3/3)

Pin No.						Pin name (Primary function)	Primary function Others	2nd function communications	3rd function communications	4th function communications	5th function Timers	6th function Others	7th function Others	8th function ADC
48Pin	52Pin	64Pin	80Pin	TQFP100	QFP100									
-	-	-	-	42	44	P90	-	-	-	-	-	-	-	-
-	-	-	-	43	45	P91	-	-	-	-	-	-	-	-
-	-	-	-	44	46	P92	-	-	-	-	-	-	-	-
-	-	-	37	45	47	P93	-	SU4_RXD0 SU4_SIN	-	-	FTM6P	-	-	-
-	-	-	38	46	48	P94	-	SU4_TXD0 SU4_SOUT	SU4_TXD1	-	FTM6N	-	-	-
-	-	-	39	47	49	P95	-	SU4_SCLK	-	-	-	-	-	-
-	-	-	40	48	50	P96	-	-	-	-	-	-	-	-
-	-	-	-	49	51	P97	-	-	-	-	-	-	-	-
-	-	-	-	50	52	PA0	-	-	-	-	-	-	-	-
-	-	-	-	53	55	PA1	-	-	-	-	-	-	-	-
-	-	-	-	54	56	PA2	-	-	-	-	-	-	-	-
-	-	-	59	71	73	PA3	EXI11	SU2_SCLK	-	-	FTM7P	-	-	AIN14
-	-	-	60	72	74	PA4	-	-	-	-	FTM7N	-	-	AIN15
-	-	-	-	73	75	PA5	-	-	-	-	-	-	-	-
-	-	-	-	74	76	PA6	-	-	-	-	-	-	-	-
-	-	-	-	75	77	PA7	-	-	-	-	-	-	-	-
-	-	-	-	76	78	PB0	-	-	-	-	-	-	-	-
-	-	-	-	77	79	PB1	-	-	-	-	-	-	-	-
-	-	-	61	78	80	PB2	-	SU5_RXD0 SU5_SIN	-	-	-	-	-	-
-	-	-	62	79	81	PB3	-	SU5_TXD0 SU5_SOUT	SU5_TXD1	-	-	-	-	-
-	-	-	63	80	82	PB4	-	SU5_SCLK	-	-	-	-	-	-
-	-	-	64	81	83	PB5	-	SU5_RXD1	SU5_RXD0	-	-	-	-	-
-	-	-	-	97	99	PB6	-	-	-	-	-	-	-	-
-	-	-	-	98	100	PB7	-	-	-	-	-	-	-	-

Table 1-9 shows the pin lists of ML62Q1700 group. "(I)" indicates the input pin and "(I/O)" indicates the input/output pin.

Table 1-9 ML62Q1700 Group Pin List (1/3)

Pin No.						Pin name (Primary function)	Primary function Others	2nd function communications	3rd function communications	4th function communications	5th function Timers	6th function Others	7th function Others	8th function ADC
48Pin	52Pin	64Pin	80Pin	TQFP100	QFP100									
3	3	3	3	3	5	V_{DD}	-	-	-	-	-	-	-	-
-	-	-	42	52	54	V_{DD}	-	-	-	-	-	-	-	-
4	4	4	4	4	6	V_{SS}	-	-	-	-	-	-	-	-
-	-	-	41	51	53	NC	-	-	-	-	-	-	-	-
5	5	5	5	5	7	V_{DDL}	-	-	-	-	-	-	-	-
1	1	1	1	1	3	XT0	PI00	-	-	-	-	-	-	-
2	2	2	2	2	4	XT1	PI01	-	-	-	-	-	-	-
6	6	6	6	6	8	RESET_N	RESET_N	-	-	-	-	-	-	-
7	7	7	7	7	9	P00	TEST0	-	-	-	-	-	-	-
8	8	8	8	8	10	P01	DACOUT0	-	-	-	FTM3P *1	TBCOUT0	TBCOUT1	-
9	9	11	14	19	21	P02	EXI0 EXTRG0	SU0_RXD0 SU0_SIN	-	I2CU0_SCL *1	FTM0P	OUTLSCLK	CMP0M	-
10	10	12	15	20	22	P03	EXI1 EXTRG1	SU0_TXD0 SU0_SOUT	SU0_TXD1	I2CU0_SDA	FTM0N	OUTHSCLK	CMP0P	AIN11
16	17	21	25	30	32	P04	EXI2 EXTRG2 COM0	SU0_SCLK	-	I2CU0_SCL	TMH0OUT	-	-	-
17	18	22	26	31	33	P05	COM1	-	-	-	-	-	-	-
18	19	23	27	32	34	P06	COM2	-	-	I2CM0_SDA	-	-	-	-
19	20	24	28	33	35	P07	COM3 SEG0	SU0_RXD1	SU0_RXD0	I2CM0_SCL	-	-	-	-
20	21	25	29	34	36	P10	COM4 SEG1	SU0_TXD1	-	-	-	-	-	-
21	22	26	30	35	37	P11	COM5 SEG2	SU0_SCLK	-	-	-	-	-	-
22	23	27	31	36	38	P12	COM6 SEG3	SU0_RXD0 SU0_SIN	-	-	TMH4OUT	-	-	-
23	24	28	32	37	39	P13	COM7 SEG4	SU0_TXD0 SU0_SOUT	SU0_TXD1	-	TMH1OUT	-	TMH3OUT	-
25	27	35	45	57	59	P14	SEG22	-	-	-	-	-	-	-
26	28	36	46	58	60	P15	SEG23	-	-	I2CU0_SDA	-	-	-	-
27	29	37	47	59	61	P16	SEG24	SU1_SCLK	-	I2CU0_SCL	TMH5OUT	-	-	-
28	30	38	48	60	62	P17	EXI3 EXTRG3 SEG25	SU0_RXD1	SU0_RXD0	-	FTM1P	TBCOUT0	BZ0P	AIN0
29	31	39	49	61	63	P20	SEG26	SU0_TXD1	-	-	FTM1N	TBCOUT1	BZ0N	AIN1
30	32	40	50	62	64	P21	EXI4 EXTRG4 SEG27	SU1_RXD0 SU1_SIN	-	-	FTM2P	OUTLSCLK	-	AIN2
31	33	41	51	63	65	P22	SEG28	SU1_TXD0 SU1_SOUT	SU1_TXD1	I2CM0_SDA	FTM2N	OUTHSCLK	-	AIN3
32	34	42	52	64	66	P23	EXI5 EXTRG5 SEG29 V_{REF}	SU1_SCLK	-	I2CM0_SCL	TMH2OUT	-	-	V_{REFO}
33	35	43	53	65	67	P24	SEG30	SU1_RXD0 SU1_SIN	-	-	-	-	-	AIN4
34	36	44	54	66	68	P25	SEG31	SU1_TXD0 SU1_SOUT	SU1_TXD1	-	-	-	-	AIN5
35	37	45	55	67	69	P26	EXI6 EXTRG6 SEG32	SU1_RXD1	SU1_RXD0	I2CU0_SDA	FTM3P	TBCOUT0	BZ0P	AIN6
36	38	46	56	68	70	P27	EXI7 EXTRG7 SEG33	SU1_TXD1	-	I2CU0_SCL	FTM3N	TBCOUT1	BZ0N	AIN7

*1: No assignment to ML62Q1500/ML62Q1800 Group.

Table 1-9 ML62Q1700 Group Pin List (2/3)

Pin No.						Pin name (Primary function)	Primary function Others	2nd function communications*3	3rd function communications*3	4th function communications	5th function Timers*3	6th function Others	7th function Others	8th function ADC*3
48Pin	52Pin	64Pin	80Pin	TQFP100	QFP100									
37	41	51	67	84	86	P30	SEG49	-	-	-	-	-	-	-
38	42	52	68	85	87	P31	SEG50	-	-	-	-	TBCOUT0	TBCOUT1	-
39	43	53	69	86	88	P32	SEG51	SU1_RXD1	SU1_RXD0	-	-	-	-	-
40	44	54	70	87	89	P33	SEG52	SU1_TXD1	-	-	TMH3OUT	-	-	-
-	-	49	65	82	84	P40	SEG47	SU5_TXD1	-	-	-	-	-	-
-	40	50	66	83	85	P41	SEG48	-	-	-	-	-	-	-
-	-	63	79	96	98	P42	SEG61	SU3_TXD1	-	-	-	-	-	-
48	52	64	80	100	2	P43	-	-	-	-	-	TBCOUT0	TBCOUT1	AIN10
-	-	9	12	17	19	P44	DACOUT1	SU4_RXD1	SU4_RXD0	-	FTM3N *1	-	-	-
-	-	10	13	18	20	P45	-	SU4_TXD1	-	-	-	-	-	-
-	-	13	16	21	23	P46	-	-	-	I2CU0_SDA *1	FTM1N *1	-	-	-
-	11	14	17	22	24	P47	-	SU0_SCLK *1	-	I2CU0_SCL *2	FTM1P *1	-	-	-
24	25	29	33	38	40	P50	EXI8 SEG5	-	-	-	-	-	-	-
-	26	30	34	39	41	P51	SEG6	-	-	-	-	-	-	-
-	-	31	35	40	42	P52	SEG7	SU4_RXD1	SU4_RXD0	-	-	-	-	-
-	-	32	36	41	43	P53	SEG8	SU4_TXD1	-	-	-	-	-	-
-	-	33	43	55	57	P54	SEG20	SU2_RXD1	SU2_RXD0	-	TMH7OUT	-	-	-
-	-	34	44	56	58	P55	SEG21	SU2_TXD1	-	-	-	-	-	-
	39	47	57	69	71	P56	SEG34	SU2_RXD0 SU2_SIN	-	-	-	-	-	AIN12
-	-	48	58	70	72	P57	SEG35	SU2_TXD0 SU2_SOUT	SU2_TXD1	-	-	-	-	AIN13
41	45	55	71	88	90	P60	SEG53	-	-	I2CM1_SCL	-	-	-	-
42	46	56	72	89	91	P61	SEG54	-	-	I2CM1_SDA	-	-	-	-
43	47	57	73	90	92	P62	SEG55	-	-	-	FTM4N	-	CMP1P	-
44	48	58	74	91	93	P63	SEG56	-	-	-	FTM4P	-	CMP1M	-
45	49	59	75	92	94	P64	EXI9 SEG57	SU3_RXD0 SU3_SIN	-	-	FTM5P	-	-	-
46	50	60	76	93	95	P65	SEG58	SU3_TXD0 SU3_SOUT	SU3_TXD1	-	FTM5N	-	-	AIN8
47	51	61	77	94	96	P66	SEG59	SU3_SCLK	-	-	-	-	-	AIN9
-	-	62	78	95	97	P67	SEG60	SU3_RXD1	SU3_RXD0	-	-	-	-	-
-	-	20	24	29	31	P70	-	-	-	-	TMH6OUT	-	-	-
15	16	19	23	28	30	V_{L3}	-	-	-	-	-	-	-	-
14	15	18	22	27	29	V_{L2}	-	-	-	-	-	-	-	-
13	14	17	21	26	28	V_{L1}	-	-	-	-	-	-	-	-
12	13	16	20	25	27	C_2	-	-	-	-	-	-	-	-
11	12	15	19	24	26	C_1	-	-	-	-	-	-	-	-
-	-	-	18	23	25	P76	EXI10	-	-	-	-	-	-	-
-	-	-	-	99	1	P77	SEG64	-	-	-	-	-	-	-

*1: No assignment to ML62Q1500/ML62Q1800 Group.
*2: No assignment to ML62Q1500/ML62Q1800 Group and products of 52 PIN-package.
*3: The pins of name with DACOUT1, SU2, SU3, SU4, SU5, TMH6, TMH7, AIN12 or AIN13 are not assigned to products of 48/52/64 PIN-packages.

Table 1-9 ML62Q1700 Group Pin List (3/3)

Pin No.						Pin name (Primary function)	Primary function Others	2nd function communications	3rd function communications	4th function communications	5th function Timers	6th function Others	7th function Others	8th function ADC
48Pin	52Pin	64Pin	80Pin	TQFP100	QFP100									
-	-	-	9	9	11	P80	-	SU4_RXD0 SU4_SIN	-	-	-	-	-	-
-	-	-	10	10	12	P81	-	SU4_TXD0 SU4_SOUT	SU4_TXD1	-	-	-	-	-
-	-	-	11	11	13	P82	-	SU4_SCLK	-	-	-	-	-	-
-	-	-	-	12	14	P83	-	SU5_RXD0 *1	-	-	-	-	-	-
-	-	-	-	13	15	P84	-	SU5_TXD0 *1	SU5_TXD1 *1	-	-	-	-	-
-	-	-	-	14	16	P85	-	-	-	-	-	-	-	-
-	-	-	-	15	17	P86	-	-	-	-	FTM7P *1	-	-	-
-	-	-	-	16	18	P87	-	-	-	-	FTM7N *1	-	-	-
-	-	-	-	42	44	P90	SEG9	-	-	-	-	-	-	-
-	-	-	-	43	45	P91	SEG10	-	-	-	-	-	-	-
-	-	-	-	44	46	P92	SEG11	-	-	-	-	-	-	-
-	-	-	37	45	47	P93	SEG12	SU4_RXD0 SU4_SIN	-	-	FTM6P	-	-	-
-	-	-	38	46	48	P94	SEG13	SU4_TXD0 SU4_SOUT	SU4_TXD1	-	FTM6N	-	-	-
-	-	-	39	47	49	P95	SEG14	SU4_SCLK	-	-	-	-	-	-
-	-	-	40	48	50	P96	SEG15	-	-	-	-	-	-	-
-	-	-	-	49	51	P97	SEG16	-	-	-	-	-	-	-
-	-	-	-	50	52	PA0	SEG17	-	-	-	-	-	-	-
-	-	-	-	53	55	PA1	SEG18	-	-	-	-	-	-	-
-	-	-	-	54	56	PA2	SEG19	-	-	-	-	-	-	-
-	-	-	59	71	73	PA3	EXI11 SEG36	SU2_SCLK	-	-	FTM7P	-	-	AIN14
-	-	-	60	72	74	PA4	SEG37	-	-	-	FTM7N	-	-	AIN15
-	-	-	-	73	75	PA5	SEG38	-	-	-	-	-	-	-
-	-	-	-	74	76	PA6	SEG39	-	-	-	-	-	-	-
-	-	-	-	75	77	PA7	SEG40	-	-	-	-	-	-	-
-	-	-	-	76	78	PB0	SEG41	-	-	-	-	-	-	-
-	-	-	-	77	79	PB1	SEG42	-	-	-	-	-	-	-
-	-	-	61	78	80	PB2	SEG43	SU5_RXD0 SU5_SIN	-	-	-	-	-	-
-	-	-	62	79	81	PB3	SEG44	SU5_TXD0 SU5_SOUT	SU5_TXD1	-	-	-	-	-
-	-	-	63	80	82	PB4	SEG45	SU5_SCLK	-	-	-	-	-	-
-	-	-	64	81	83	PB5	SEG46	SU5_RXD1	SU5_RXD0	-	-	-	-	-
-	-	-	-	97	99	PB6	SEG62	-	-	-	-	-	-	-
-	-	-	-	98	100	PB7	SEG63	-	-	-	-	-	-	-

*1: No assignment to ML62Q1500/ML62Q1800 Group.

1.3.3 PIN DESCRIPTION

Table 1-10 shows the pin list categorized by the function.
"-" : Power pin, "I": Input pin, "O" Output pin and "I/O" : Input/Output pin

Table 1-10 Pin Description

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 between this pin and V_{SS}.	-
	-	V_{DDL}	-	Power supply pin for internal logic (internal regulator's output). Connect a capacitor C_L (1µF) between this pin and V_{SS}.	-
Test	TEST0	P00	I/O	Input for testing, is used as on-chip debug interface and ISP function. P00 is initialized as pull-up input mode by the system reset.	-
Un used	NC	NC	-	Connect to V_{SS}.	-
System	V_{REFO}	P23	-	Reference voltage output	-
	RESET_N	RESET_N	I	Reset input. Applying "L" level shifts the MCU in system reset mode. Applying "H" level shifts the CPU in program running mode. Used for on-chip debug interface and ISP function. No pull-up resistor is installed.	Negative
	XT0	XT0	I	Low speed crystal oscillation pins Connect 32.768kHz crystal resonator and Connect capacitors between the pin and V_{SS}.	-
	XT1	XT1	O		-
	OUTLSCLK	P02	O	Low-speed clock output.	-
		P21			
	OUTHSCLK	P03	O	High-speed clock output.	-
		P22			
General purpose port	PI00,PI01	XT0,XT1	I	General purpose input. Not available as general inputs when using the crystal resonator.	Positive
	P00	P00	I/O	General purpose I/O - High-impedance that both input and output is disabled - Input with Pull-UP (initial value) - Input without Pull-UP - CMOS output - N-channel open drain output Not available to use as I/O pin when using for on-chip debug interface or ISP function.	Positive
	P01 to P07	P01 to P07	I/O	General purpose I/O - High-impedance that both input and output is disabled (initial value) - Input with Pull-UP - Input without Pull-UP - CMOS output - N-channel open drain output	Positive
	P10 to P17	P10 to P17	I/O		Positive
	P20 to P27	P20 to P27	I/O		Positive
	P30 to P33	P30 to P33	I/O		Positive
	P40 to P47	P40 to P47	I/O		Positive
	P50 to P57	P50 to P57	I/O		Positive
	P60 to P67	P60 to P67	I/O		Positive
	P70 to P77	P70 to P77	I/O		Positive
	P80 to P87	P80 to P87	I/O		Positive
	P90 to P97	P90 to P97	I/O		Positive
	PA0 to PA7	PA0 to PA7	I/O		Positive
	PB0 to PB7	PB0 to PB7	I/O		Positive

Function	Signal name	Pin name	I/O	Description	Logic
Serial communication unit (UART mode)	SU0_TXD0	P03	O	Serial communication unit0 UART0 data output	Positive
		P13			
	SU0_RXD0	P02	I	Serial communication unit0 Full-duplex data input Serial communication unit0 UART0 data input	Positive
		P07			
		P12			
		P17			
	SU0_TXD1	P03	O	Serial communication unit0 Full-duplex data output Serial communication unit0 UART1 data output	Positive
		P10			
		P13			
		P20			
	SU0_RXD1	P07	I	Serial communication unit0 UART1 data input	Positive
		P17			
	SU1_TXD0	P22	O	Serial communication unit1 UART0 data output	Positive
		P25			
	SU1_RXD0	P21	I	Serial communication unit1 Full-duplex data input Serial communication unit1 UART0 data input	Positive
		P24			
		P26			
		P32			
	SU1_TXD1	P22	O	Serial communication unit1 Full-duplex data output Serial communication unit1 UART1 data output	Positive
		P25			
		P27			
		P33			
	SU1_RXD1	P26	I	Serial communication unit1 UART1 data input	Positive
		P32			
	SU2_TXD0	P57	O	Serial communication unit2 UART0 data output	Positive
	SU2_RXD0	P54	I	Serial communication unit2 Full-duplex data input Serial communication unit2 UART0 data input	Positive
		P56			
	SU2_TXD1	P55	O	Serial communication unit2 Full-duplex data output Serial communication unit2 UART1 data output	Positive
		P57			
	SU2_RXD1	P54	I	Serial communication unit2 UART1 data input	Positive
	SU3_TXD0	P65	O	Serial communication unit3 UART0 data output	Positive
	SU3_RXD0	P64	I	Serial communication unit3 Full-duplex data input Serial communication unit3 UART0 data input	Positive
		P67			
	SU3_TXD1	P42	O	Serial communication unit3 Full-duplex data output Serial communication unit3 UART1 data output	Positive
		P65			
	SU3_RXD1	P67	I	Serial communication unit3 UART1 data input	Positive
	SU4_TXD0	P81	O	Serial communication unit4 UART0 data output	Positive
		P94			
	SU4_RXD0	P44	I	Serial communication unit4 Full-duplex data input Serial communication unit4 UART0 data input	Positive
		P52			
		P80			
		P93			
	SU4_TXD1	P45	O	Serial communication unit4 Full-duplex data output Serial communication unit4 UART1 data output.	Positive
		P53			
		P81			
		P94			
	SU4_RXD1	P44	I	Serial communication unit4 UART1 data input	Positive
		P52			
	SU5_TXD0	P84	O	Serial communication unit5 UART0 data output	Positive
		PB3			
	SU5_RXD0	P83	I	Serial communication unit5 Full-duplex data input Serial communication unit5 UART0 data input	Positive
		PB2			
		PB5			
	SU5_TXD1	P40	O	Serial communication unit5 Full-duplex data output Serial communication unit5 UART1 data output.	Positive
		P84			
		PB3			

Function	Signal name	Pin name	I/O	Description	Logic
UART mode	SU5_RXD1	PB5	I	Serial communication unit5 UART1 data input	Positive
Synchronous Serial Port	SU0_SIN	P02	I	Serial communication unit0 Synchronous serial data input	Positive
		P12			
	SU0_SCLK	P04	I/O	Serial communication unit0 Synchronous serial clock I/O	Positive
		P11			
		P47			
	SU0_SOUT	P03	O	Serial communication unit0 Synchronous serial data output	Positive
		P13			
	SU1_SIN	P21	I	Serial communication unit1 Synchronous serial data input	Positive
		P24			
	SU1_SCLK	P16	I/O	Serial communication unit1 Synchronous serial clock I/O	Positive
		P23			
	SU1_SOUT	P22	O	Serial communication unit1 Synchronous serial data output	Positive
		P25			
	SU2_SIN	P56	I	Serial communication unit2 Synchronous serial data	Positive
	SU2_SCLK	PA3	I/O	Serial communication unit2 Synchronous serial clock I/O	Positive
	SU2_SOUT	P57	O	Serial communication unit2 Synchronous serial data output	Positive
	SU3_SIN	P64	I	Serial communication unit3 Synchronous serial data input	Positive
	SU3_SCLK	P66	I/O	Serial communication unit3 Synchronous serial clock I/O	Positive
	SU3_SOUT	P65	O	Serial communication unit3 Synchronous serial data output	Positive
	SU4_SIN	P80	I	Serial communication unit4 Synchronous serial data input	Positive
		P93			
	SU4_SCLK	P95	I/O	Serial communication unit4 Synchronous serial clock I/O	Positive
		P82			
	SU4_SOUT	P81	O	Serial communication unit4 Synchronous serial data output	Positive
		P94			
	SU5_SIN	PB2	I	Serial communication unit5 Synchronous serial data input	Positive
	SU5_SCLK	PB4	I/O	Serial communication unit5 Synchronous serial clock I/O	Positive
	SU5_SOUT	PB3	O	Serial communication unit5 Synchronous serial data output	Positive
I^2C Bus	I2CU0_SDA	P03	I/O	I^2C Unit0 (Master and Salve) Data I/O N-channel open drain Connect a pull-up resistor externally	Positive
		P15			
		P26			
		P46			
	I2CU0_SCL	P02	I/O	I^2C Unit0 (Master and Salve) Clock I/O N-channel open drain output Connect a pull-up resistor externally	Positive
		P04			
		P16			
		P27			
		P47			
	I2CM0_SDA	P06	I/O	I^2C Master0 Data I/O pin N-channel open drain output Connect a pull-up resistor externally	Positive
		P22			
	I2CM0_SCL	P07	I/O	I^2C Master0 Clock I/O N-channel open drain output Connect a pull-up resistor externally	Positive
		P23			
	I2CM1_SDA	P61	I/O	I^2C Master1 Data I/O N-channel open drain output Connect a pull-up resistor externally	Positive
	I2CM1_SCL	P60	I/O	I^2C Master1 Clock I/O N-channel open drain output Connect a pull-up resistor externally	Positive

Function	Signal name	Pin name	I/O	Description	Logic
Functional Timer (FTM)	FTM0P	P02	O	Functional Timer0 P output	Positive
	FTM0N	P03	O	Functional Timer0 N output	Negative
	FTM1P	P17	O	Functional Timer1 P output	Positive
		P47			
	FTM1N	P20	O	Functional Timer1 N output	Negative
		P46			
	FTM2P	P21	O	Functional Timer2 P output	Positive
	FTM2N	P22	O	Functional Timer2 N output	Negative
	FTM3P	P01	O	Functional Timer3 P output	Positive
		P26			
	FTM3N	P27	O	Functional Timer3 N output	Negative
		P44			
	FTM4P	P63	O	Functional Timer4 P output	Positive
	FTM4N	P62	O	Functional Timer4 N output	Negative
	FTM5P	P64	O	Functional Timer5 P output	Positive
	FTM5N	P65	O	Functional Timer5 N output	Negative
	FTM6P	P93	O	Functional Timer6 P output	Positive
	FTM6N	P94	O	Functional Timer6 N output	Negative
	FTM7P	P86	O	Functional Timer7 P output	Positive
		PA3			
	FTM7N	P87	O	Functional Timer7 N output	Negative
		PA4			
	EXTRG0	P02	I	Functional Timer event trigger input	-
	EXTRG1	P03	I	Functional Timer event trigger input	-
	EXTRG2	P04	I	Functional Timer event trigger input	-
	EXTRG3	P17	I	Functional Timer event trigger input	-
	EXTRG4	P21	I	Functional Timer event trigger input	-
	EXTRG5	P23	I	Functional Timer event trigger input	-
	EXTRG6	P26	I	Functional Timer event trigger input	-
	EXTRG7	P27	I	Functional Timer event trigger input	-
16 bit Timer	TMH0OUT	P04	O	16bit General Timer 0 output	Positive
	TMH1OUT	P13	O	16bit General Timer 1 output	Positive
	TMH2OUT	P23	O	16bit General Timer 2 output	Positive
	TMH3OUT	P13	O	16bit General Timer 3 output	Positive
		P33			
	TMH4OUT	P12	O	16bit General Timer 4 output	Positive
	TMH5OUT	P16	O	16bit General Timer 5 output	Positive
	TMH6OUT	P70	O	16bit General Timer 6 output	Positive
	TMH7OUT	P54	O	16bit General Timer 7 output	Positive
	EXTRG0	P02	I	16bit Timer trigger input	-
	EXTRG1	P03	I	16bit Timer trigger input	-
Low-speed Time Base Counter	TBCOUT0	P01	O	The virtual frequency adjustment output signal or the low speed time base counter output signal	Positive
		P17			
		P26			
		P31			
		P43			
	TBCOUT1	P01	O	ML62Q1300 group: The low speed time base counter output signal ML62Q1500/ML62Q1800/ML62Q1700 group: 1Hz/2Hz clock for the Simplified RTC	Positive
		P20			
		P27			
		P31			
		P43			
Buzzer	BZ0P	P17	O	Buzzer output (positive phase)	Positive
		P26			
	BZ0N	P20	O	Buzzer output (negative phase)	Negative
		P27			

Function	Signal name	Pin name	I/O	Description	Logic
External Interrupt	EXI0	P02	I	External Interrupt 0 Input	-
	EXI1	P03	I	External Interrupt 1 Input	-
	EXI2	P04	I	External Interrupt 2 Input	-
	EXI3	P17	I	External Interrupt 3 Input	-
	EXI4	P21	I	External Interrupt 4 Input	-
	EXI5	P23	I	External Interrupt 5 Input	-
	EXI6	P26	I	External Interrupt 6 Input	-
	EXI7	P27	I	External Interrupt 7 Input	-
	EXI8	P50	I	External Interrupt 8 Input	-
	EXI9	P64	I	External Interrupt 9 Input	-
	EXI10	P76	I	External Interrupt 10 Input	-
	EXI11	PA3	I	External Interrupt 11 Input	-
Successive approximation type A/D converter (SA-ADC)	V_{REF}	P23	-	SA-ADC external reference voltage input	-
	AIN0	P17	I	SA-ADC channel 0 input	-
	AIN1	P20	I	SA-ADC channel 1 input	-
	AIN2	P21	I	SA-ADC channel 2 input	-
	AIN3	P22	I	SA-ADC channel 3 input	-
	AIN4	P24	I	SA-ADC channel 4 input	-
	AIN5	P25	I	SA-ADC channel 5 input	-
	AIN6	P26	I	SA-ADC channel 6 input	-
	AIN7	P27	I	SA-ADC channel 7 input	-
	AIN8	P65	I	SA-ADC channel 8 input	-
	AIN9	P66	I	SA-ADC channel 9 input	-
	AIN10	P43	I	SA-ADC channel 10 input	-
	AIN11	P03	I	SA-ADC channel 11 input	-
	AIN12	P56	I	SA-ADC channel 12 input	-
	AIN13	P57	I	SA-ADC channel 13 input	-
	AIN14	PA3	I	SA-ADC channel 14 input	-
	AIN15	PA4	I	SA-ADC channel 15 input	-
Analog comparator	CMP0P	P03	I	Comparator input 0 (noninverting input)	-
	CMP0M	P02	I	Comparator input 0 (inverting input)	-
	CMP1P	P62	I	Comparator input 1 (noninverting input)	-
	CMP1M	P63	I	Comparator input 1 (inverting input)	-
D/A converter	DACOUT0	P01	O	D/A converter 0 output	-
	DACOUT1	P44	O	D/A converter 1 output	-

Function	Signal name	Pin name	I/O	Description	Logic
LCD driver	COM0	P04	-	Common output	-
	COM1	P05	-	Common output	-
	COM2	P06	-	Common output	-
	COM3/SEG0	P07	-	Common/Segment output shared	-
	COM4/SEG1	P10	-	Common/Segment output shared	-
	COM5/SEG2	P11	-	Common/Segment output shared	-
	COM6/SEG3	P12	-	Common/Segment output shared	-
	COM7/SEG4	P13	-	Common/Segment output shared	-
	SEG5	P50	-	Segment output	-
	SEG6	P51	-	Segment output	-
	SEG7	P52	-	Segment output	-
	SEG8	P53	-	Segment output	-
	SEG9	P90	-	Segment output	-
	SEG10	P91	-	Segment output	-
	SEG11	P92	-	Segment output	-
	SEG12	P93	-	Segment output	-
	SEG13	P94	-	Segment output	-
	SEG14	P95	-	Segment output	-
	SEG15	P96	-	Segment output	-
	SEG16	P97	-	Segment output	-
	SEG17	PA0	-	Segment output	-
	SEG18	PA1	-	Segment output	-
	SEG19	PA2	-	Segment output	-
	SEG20	P54	-	Segment output	-
	SEG21	P55	-	Segment output	-
	SEG22	P14	-	Segment output	-
	SEG23	P15	-	Segment output	-
	SEG24	P16	-	Segment output	-
	SEG25	P17	-	Segment output	-
	SEG26	P20	-	Segment output	-
	SEG27	P21	-	Segment output	-
	SEG28	P22	-	Segment output	-
	SEG29	P23	-	Segment output	-
	SEG30	P24	-	Segment output	-
	SEG31	P25	-	Segment output	-
	SEG32	P26	-	Segment output	-
	SEG33	P27	-	Segment output	-
	SEG34	P56	-	Segment output	-
	SEG35	P57	-	Segment output	-
	SEG36	PA3	-	Segment output	-
	SEG37	PA4	-	Segment output	-
	SEG38	PA5	-	Segment output	-
	SEG39	PA6	-	Segment output	-
	SEG40	PA7	-	Segment output	-
	SEG41	PB0	-	Segment output	-
	SEG42	PB1	-	Segment output	-
	SEG43	PB2	-	Segment output	-

Function	Signal name	Pin name	I/O	Description	Logic
LCD driver	SEG44	PB3	-	Segment output	-
	SEG45	PB4	-	Segment output	-
	SEG46	PB5	-	Segment output	-
	SEG47	P40	-	Segment output	-
	SEG48	P41	-	Segment output	-
	SEG49	P30	-	Segment output	-
	SEG50	P31	-	Segment output	-
	SEG51	P32	-	Segment output	-
	SEG52	P33	-	Segment output	-
	SEG53	P60	-	Segment output	-
	SEG54	P61	-	Segment output	-
	SEG55	P62	-	Segment output	-
	SEG56	P63	-	Segment output	-
	SEG57	P64	-	Segment output	-
	SEG58	P65	-	Segment output	-
	SEG59	P66	-	Segment output	-
	SEG60	P67	-	Segment output	-
	SEG61	P42	-	Segment output	-
	SEG62	PB6	-	Segment output	-
	SEG63	PB7	-	Segment output	-
	SEG64	P77	-	Segment output	-
	C_1 , C_2	C_1 , C_2	-	LCD bias power source generation capacitor connection	-
	V_{L1} to V_{L3}	V_{L1} to V_{L3}	-	LCD bias power source Connect the capacitors (C_{L1} ,C_{L2} ,C_{L3}) between the pin and Vss.	-

1.3.4 TERMINATION OF UNUSED PINS

Table 1-11 shows how to terminate unused pins.

Table 1-11 Termination of unused pins

Pin	Recommended pin termination
NC	Connect to V_{SS}
RESET_N	Connect to V_{DD}
P00/TEST0	Connect to V_{DD} with initial state (pulled-up input mode)
XT0/PI00, XT1/PI01	Open with initial state(Hi-impedance)
P01 to P07	
P10 to P17	
P20 to P27	
P30 to P33	
P40 to P47	
P50 to P57	
P60 to P67	
P70 to P77	
P80 to P87	
P90 to P97	
PA0 to PA7	
PB0 to PB7	
C_1, C_2	Open
V_{L1}, V_{L2}	Open
V_{L3}	It is recommended to connect to V_{DD} through a resistor (1kΩ or more).

[Note]

Terminate unused input pins according to the table 1-11 in order to avoid unexpected through-current in the pins.

Chapter 2 CPU and Memory Space

2. CPU and Memory Space

2.1 General Description

ML62Q1000 series has LAPIS Technology's original 16-bit CPU nX-U16/100 (A35 core), the multiplier/divider in the coprocessor, flash memory in the program memory space, and RAM and data flash in the data memory space.
In addition, it has the built-in remap function that remaps a 4 Kbyte area in the program memory space.
Table 2-1 to 2-3 show the memory size of the program memory space and the data memory space as well as the CPU memory model. For details of memory model, see "nX-U16/100 Core Instruction Manual".

Table 2-1 Program Memory Space and Data Memory Space of ML62Q1300 Group

Product name	Program memory space	Data memory space	Data flash size	Memory model
	ROM size	RAM size		
ML62Q1323 ML62Q1333	16 Kbyte	2 Kbyte	2 Kbyte	SMALL
ML62Q1324 ML62Q1334	24 Kbyte			
ML62Q1325 ML62Q1335	32 Kbyte			
ML62Q1345 ML62Q1365	32 Kbyte	4 Kbyte		
ML62Q1346 ML62Q1366	48 Kbyte			
ML62Q1347 ML62Q1367	64 Kbyte			

Table 2-2 Program Memory Space and Data Memory Space of ML62Q1500/ML62Q1800 Group

Product name	Program memory space	Data memory space	Data flash size	Memory model
	ROM size	RAM size		
ML62Q1530 ML62Q1540 ML62Q1550	32 Kbyte	8 Kbyte	4 Kbyte	SMALL
ML62Q1531 ML62Q1541 ML62Q1551	48 Kbyte			
ML62Q1532 ML62Q1542 ML62Q1552	64 Kbyte			
ML62Q1533 ML62Q1543 ML62Q1553	96 Kbytes	8 Kbyte		LARGE
ML62Q1563 ML62Q1573		16 Kbyte		
ML62Q1534 ML62Q1544 ML62Q1554	128 Kbyte	8 Kbyte		
ML62Q1564 ML62Q1574		16 Kbyte		
ML62Q1555 ML62Q1565 ML62Q1575	160 Kbyte	16 Kbyte		
ML62Q1556 ML62Q1566 ML62Q1576	192 Kbyte			
ML62Q1557 ML62Q1567 ML62Q1577	256 Kbyte			
ML62Q1858 ML62Q1868 ML62Q1878	384 Kbyte	32 Kbyte	8 Kbyte	
ML62Q1859 ML62Q1869 ML62Q1879	512 Kbyte			

Table 2-3 Program Memory Space and Data Memory Space of ML62Q1700 Group

Product name	Program memory space	Data memory space	Data flash size	Memory model
	ROM size	RAM size		
ML62Q1700 ML62Q1710 ML62Q1720	32 Kbyte	8 Kbyte	4 Kbyte	SMALL
ML62Q1701 ML62Q1711 ML62Q1721	48 Kbyte			
ML62Q1702 ML62Q1712 ML62Q1722	64 Kbyte			
ML62Q1703 ML62Q1713 ML62Q1723	96 Kbytes	8 Kbyte		LARGE
ML62Q1733 ML62Q1743		16 Kbyte		
ML62Q1704 ML62Q1714 ML62Q1724	128 Kbyte	8 Kbyte		
ML62Q1734 ML62Q1744		16 Kbyte		
ML62Q1725 ML62Q1735 ML62Q1745	160 Kbyte	16 Kbyte		
ML62Q1726 ML62Q1736 ML62Q1746	192 Kbyte			
ML62Q1727 ML62Q1737 ML62Q1747	256 Kbyte			
ML62Q1728 ML62Q1738 ML62Q1748	384 Kbyte	32 Kbyte	8 Kbyte	
ML62Q1729 ML62Q1739 ML62Q1749	512 Kbyte			

2.2 CPU nX-U16/100

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

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

2.2.1 Wait Mode and No-wait Mode

ML62Q1000 series has two CPU operation modes: wait mode and no-wait mode.
The mode can be chosen by Code Option. The maximum CPU operating frequency differs between the wait mode and no-wait mode depending on PLL reference frequency chosen by the Code Option. Table 2-4 shows maximum operating frequency of high-speed clock, peripheral circuit and CPU. See Chapter 26 "Code Option" for details on how to set the Code Option.

Table 2-4 Maximum Operating Frequency of High-Speed Clock, Peripheral Circuit and CPU

PLL reference frequency	Maximum operating frequency of peripheral circuit	Maximum operating frequency of CPU	
		Wait mode	No-wait mode
24MHz	24MHz	24MHz	6MHz
16MHz	16MHz	16MHz	8MHz

- Wait mode
  In this mode, instruction codes read from the program memory are stored into the built-in buffer.
  The CPU can work at high speed to read the instructions from the buffer.
  In contiguous address instruction processing, the instructions can be executed without a wait time for storing them in the buffer. In branch instruction processing, the number of execution cycles increases due to a wait time for storing the instructions in the buffer.
  The maximum operating frequency of the CPU in the wait mode is 24 MHz.

- No-wait mode
  This mode allows the CPU to directly execute instruction codes read from the program memory without involving the buffer. This mode minimizes the number of instruction execution cycles.
  The maximum operating frequency of the CPU in the no-wait mode is 8 MHz.

See Appendix C "Instruction execution cycle" for the number of instruction execution cycles in wait and no-wait modes.
The CPU operation mode (wait mode or no-wait mode) can be chosen by the Code Option is applied even when the low-speed clock (LSCLK) is used for the system clock.

2.2.2 Notes When Executing SB/RB Instruction

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

2.2.3 Notes on the Description of Read-modify-write

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

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

```
SFR &= 0xFFFE;   ──► Converted to RB SFR.0 ;
SFR |= 0x0081;   ──► Converted to SB SFR.7 ;
                     SB SFR.0 ;
```

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

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

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

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

2.3 Coprocessor

ML62Q1000 series has the built-in multiplier/divider in the coprocessor.
The multiplier/divider is operated using coprocessor data transfer instructions of the CPU. For coprocessor data transfer instructions, see "nX-U16/100 Core Instruction Manual".

2.3.1 Multiplier/Divider

The multiplier/divider has following arithmetic functions:

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

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

2.3.2 List of Coprocessor General-purpose Registers

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

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

CR0 to CR7 are registers to store the setting of the input values of operations and operation results.
CR8 is a register to set each operation mode (signed, unsigned) and to enable/disable the operation.
CR9 is a register to store the status of each operation result.
CR15 is a register to indicate coprocessor ID.

[Note]

- **Registers CR10 to CR14 have no function. Reading them returns "0x00". These registers are not writable.**

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

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

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

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

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

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

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

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

As soon as the data is written in register CR7, operation is started.
In a saturating multiply-accumulate operation, the result is fixed to 0x7FFF_FFFF for a positive number and 0x8000_0000 for a negative number when it is out of the expressible range.

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

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

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

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

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

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

Bit No.	Bit symbol name	Description
2 to 0	clmod2 to clmod0	Bits to choose the operation mode. 000: Multiplication 16 bit × 16 bit (initial value) 001: Division 32 bit ÷ 16 bit 010: Multiply-accumulate (non-saturating) 16 bit × 16 bit + 32 bit 011: Multiply-accumulate (saturating) 16 bit × 16 bit + 32 bit 100: No operation function 101: Division 32 bit ÷ 32 bit 110: No operation function 111: No operation function

The following table shows values to be set to CR8 register for execution of each operation mode.

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

The following table shows flags changing during each operation.

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

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

2.3.2.3 Coprocessor ID Register (CR15)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CER14															
Byte	CR15								CR14							
Bit	copid 7	copid 6	copid 5	copid 4	copid 3	copid 2	copid 1	copid 0	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

2.3.3 How to Use Multiplier/Divider

For the use of the multiplier/divider, the multiplication/division library is provided. See MULDIVU8LIB manual for details.

2.4 Memory Space

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

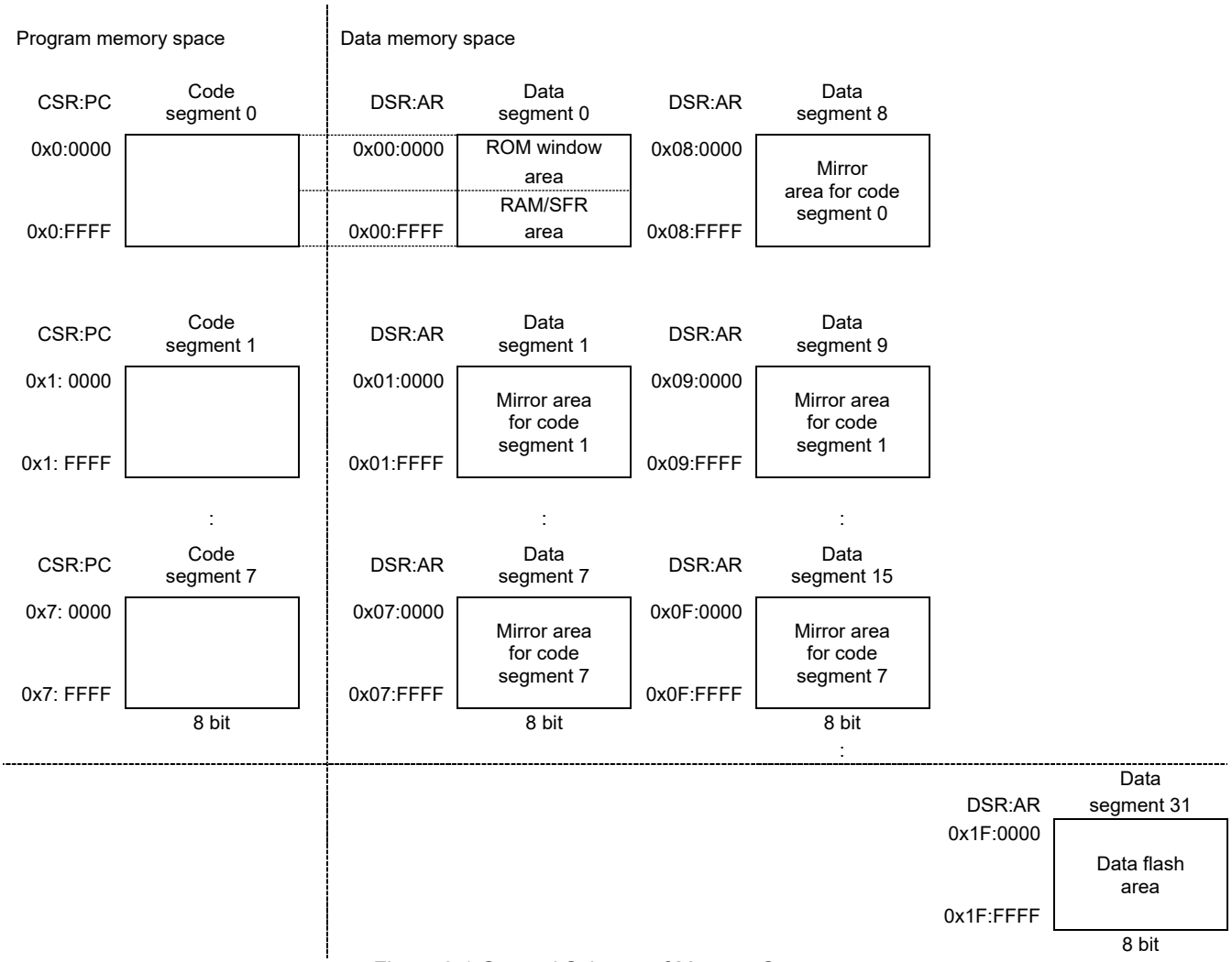


Figure 2-1 General Scheme of Memory Space

2.5 Program Memory Space

The program memory space is an area to store the program code, vector table, and Code Options.
The program memory space is specified by 20 bits (CSR:PC) consisting of higher 4 bits as code segment register (CSR) and lower 16 bits as program counter (PC).
The vector table area is used as the reset vector, hardware interrupt vector, and software interrupt vector. Unused software interrupt vector area is available as a program code area.
The Code Option area can be used to choose the CPU operation mode, PLL reference frequency, watchdog timer (WDT) operation mode, unused ROM area access reset enabled/disabled, and remapping function enabled/disabled.
The program code, vector table, and Code Option areas can be read from the ROM window area or the mirror area of the data memory space by executing the memory access instruction.
Figures 2-2 to 2-18 show the program memory space configuration of each product of the ML62Q1000 series.

[Note]

- **CSR[3] is unused on the ML62Q1000 series. The data of CSR "0x8 to 0xF" are handled as "0x0 to 0x7".**
- **The Code Option area (64 bytes) is not available for the program code area. For details of Code Option settings, see Chapter 26 "Code Option" and make sure the setting data is correct.**
- **It is recommended to fill unused areas with data "0xFFFF" (BRK instruction) in the program memory space to ensure failsafe using the generation tool of the ROM code data. See its manual for details on how to use. See "nX-U16/100 Core Instruction Manual" for details of the BRK instruction.**
- **Do not read or program unused areas to prevent the CPU works incorrectly.**

■ ML62Q1300 group

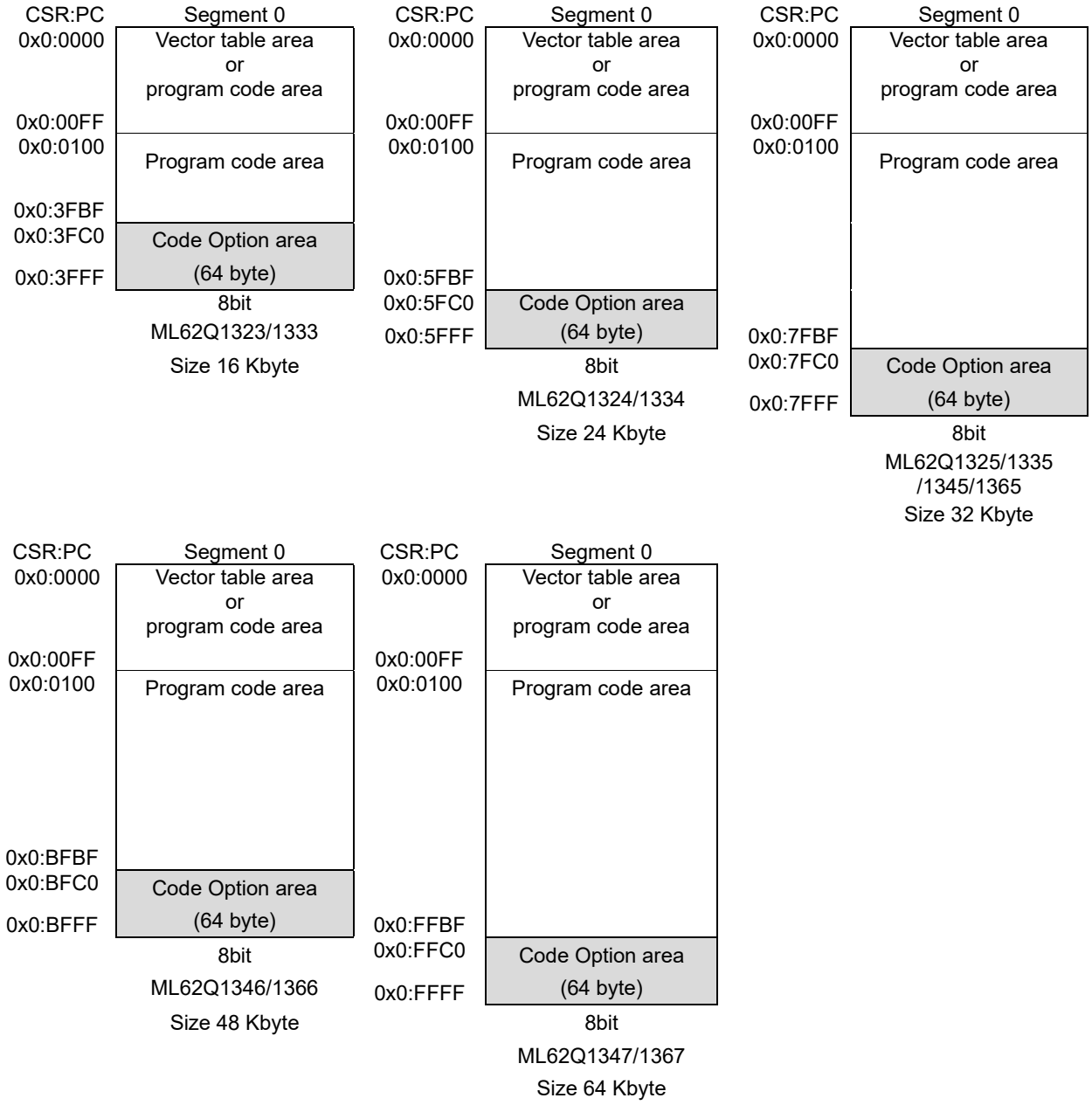


Figure 2-2 ML62Q1300 Group Configuration of Program Memory Space

■ ML62Q1500/ML62Q1800 group

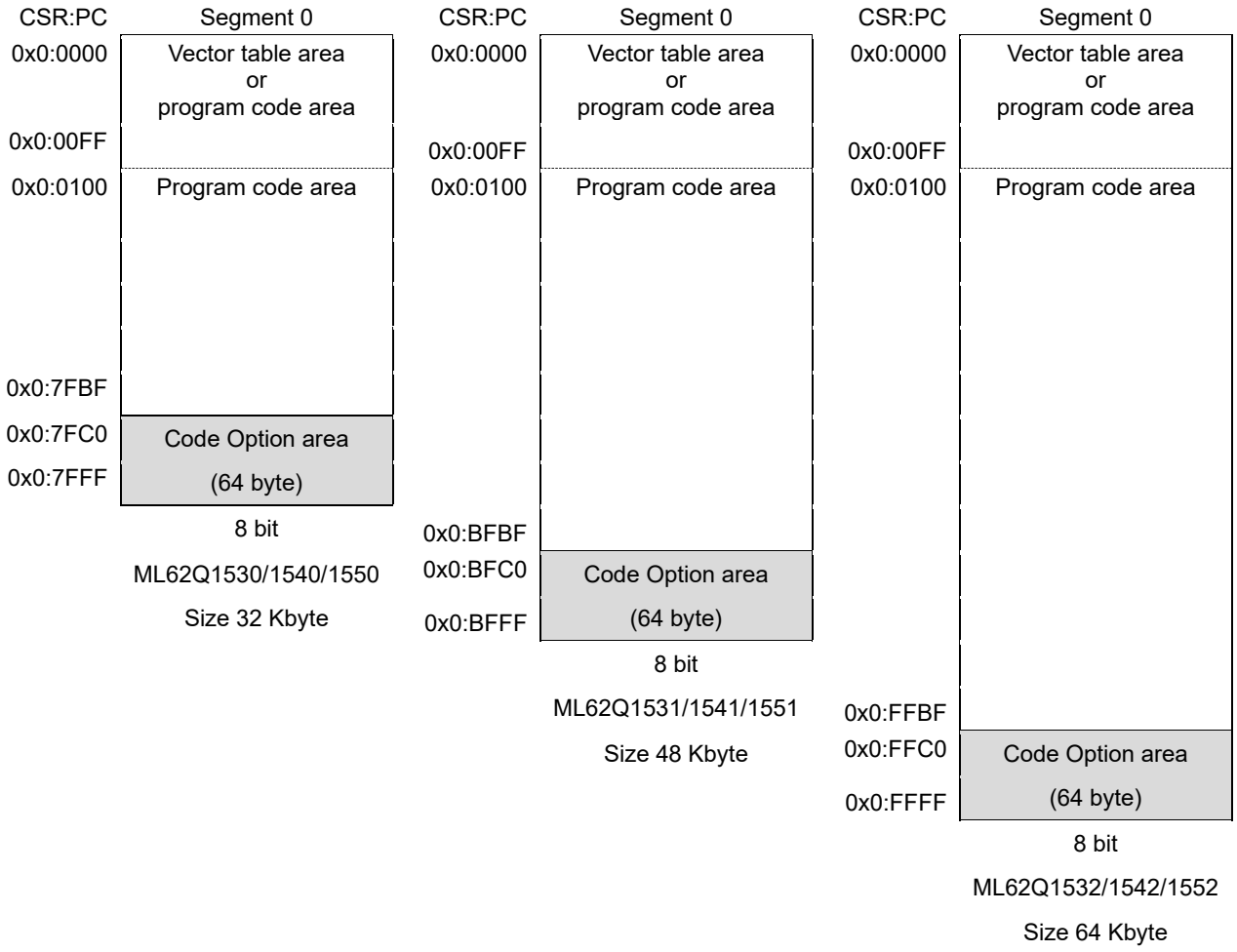
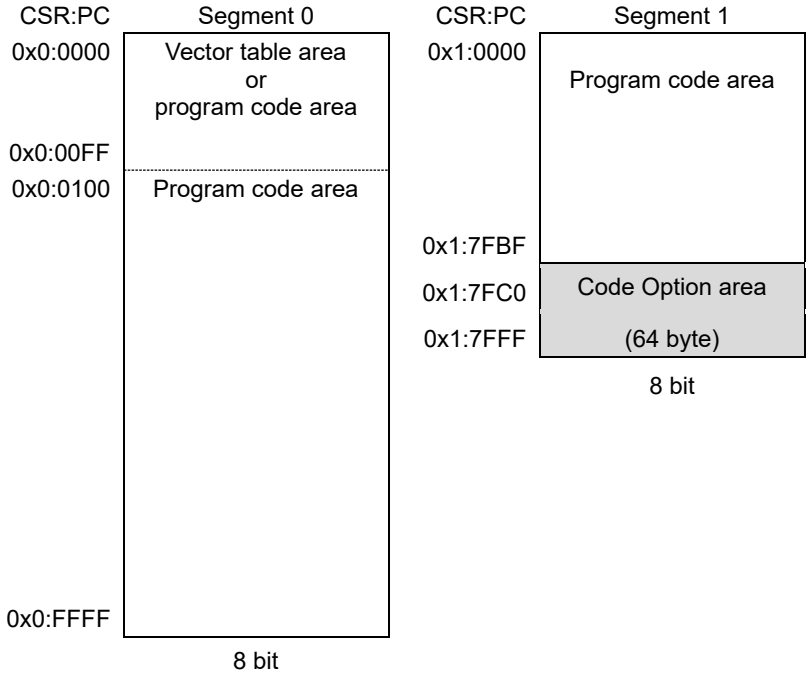


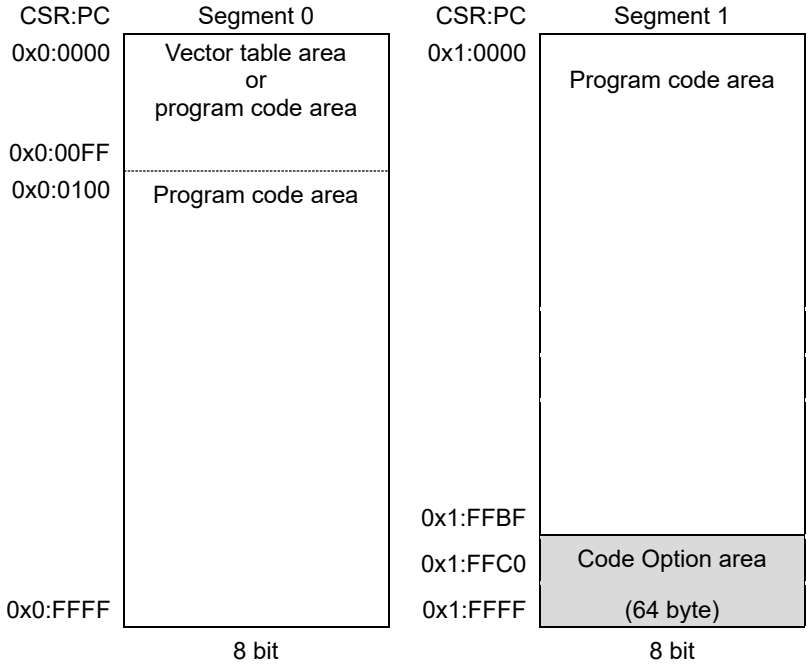
Figure 2-3 ML62Q1500/ML62Q1800 Group Configuration of Program Memory Space 1



ML62Q1533/1543/1553/1563/1573

Size 96 Kbyte

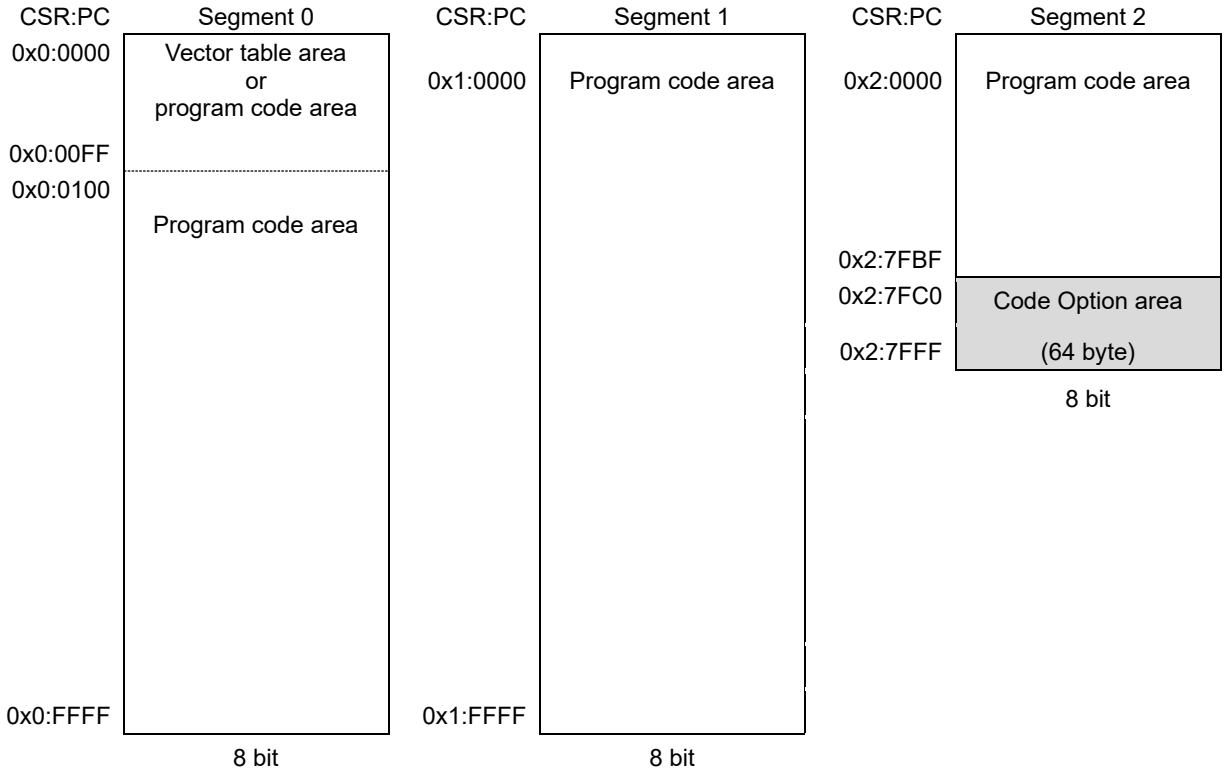
Figure 2-4 ML62Q1500/ML62Q1800 Group Configuration of Program Memory Space 2



ML62Q1534/1544/1554/1564/1574

Size 128 Kbyte

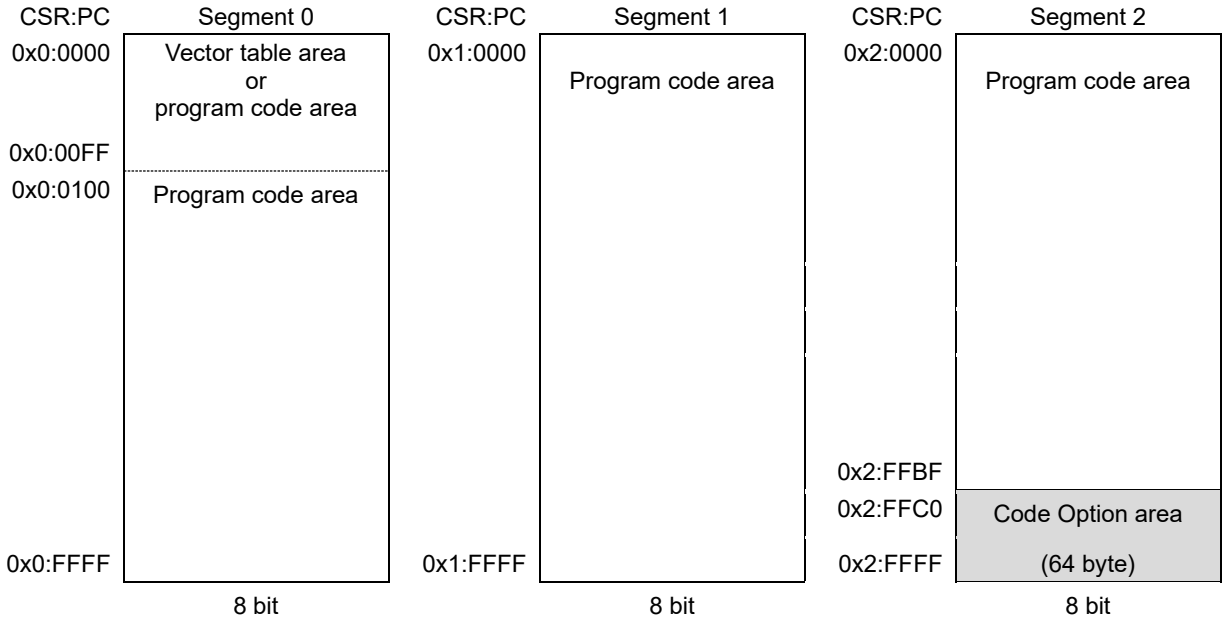
Figure 2-5 ML62Q1500/ML62Q1800 Group Configuration of Program Memory Space 3



ML62Q1555/1565/1575

Size 160 Kbyte

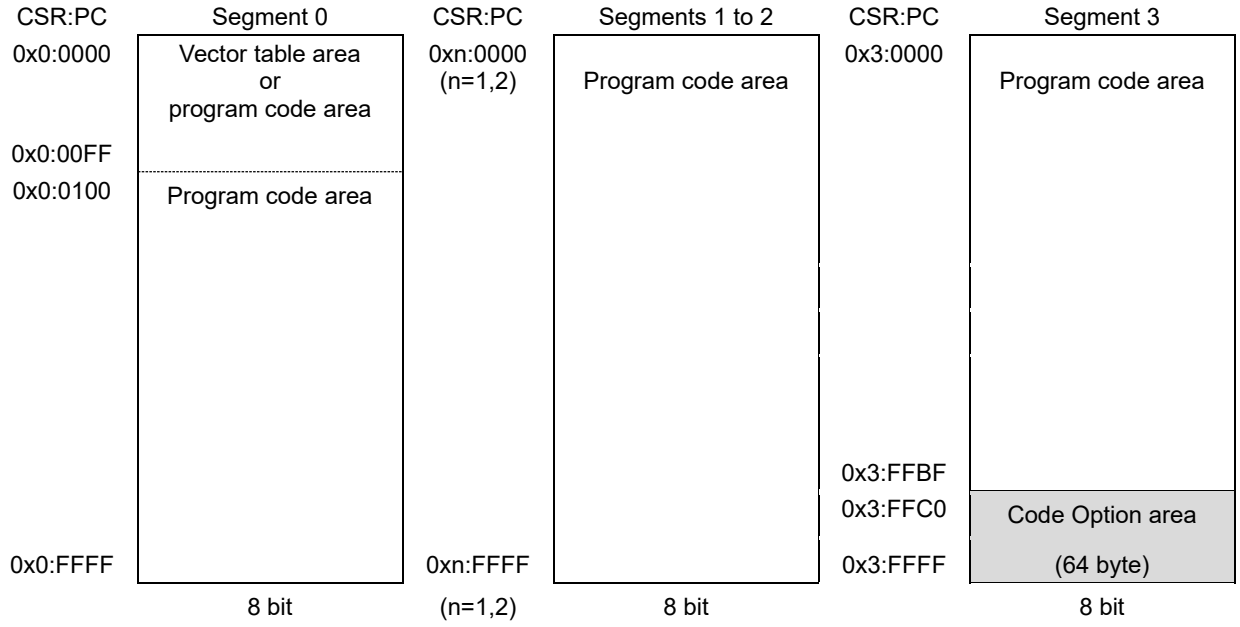
Figure 2-6 ML62Q1500/ML62Q1800 Group Configuration of Program Memory Space 4



ML62Q1556/1566/1576

Size 192 Kbyte

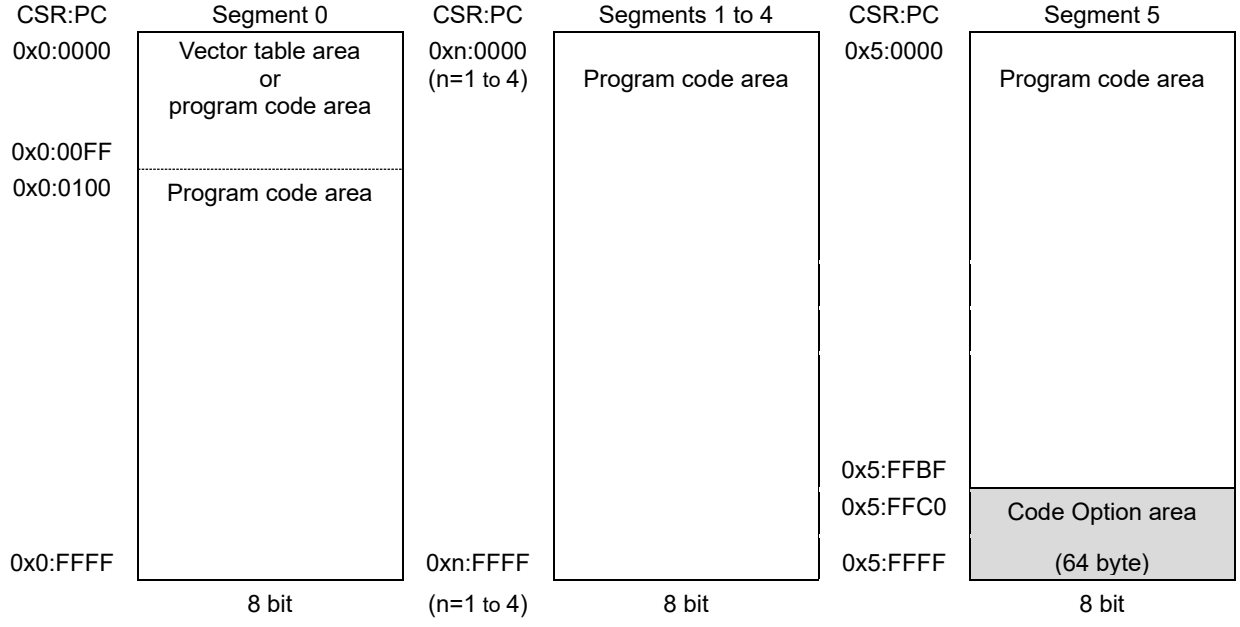
Figure 2-7 ML62Q1500/ML62Q1800 Group Configuration of Program Memory Space 5



ML62Q1557/1567/1577

Size 256 Kbyte

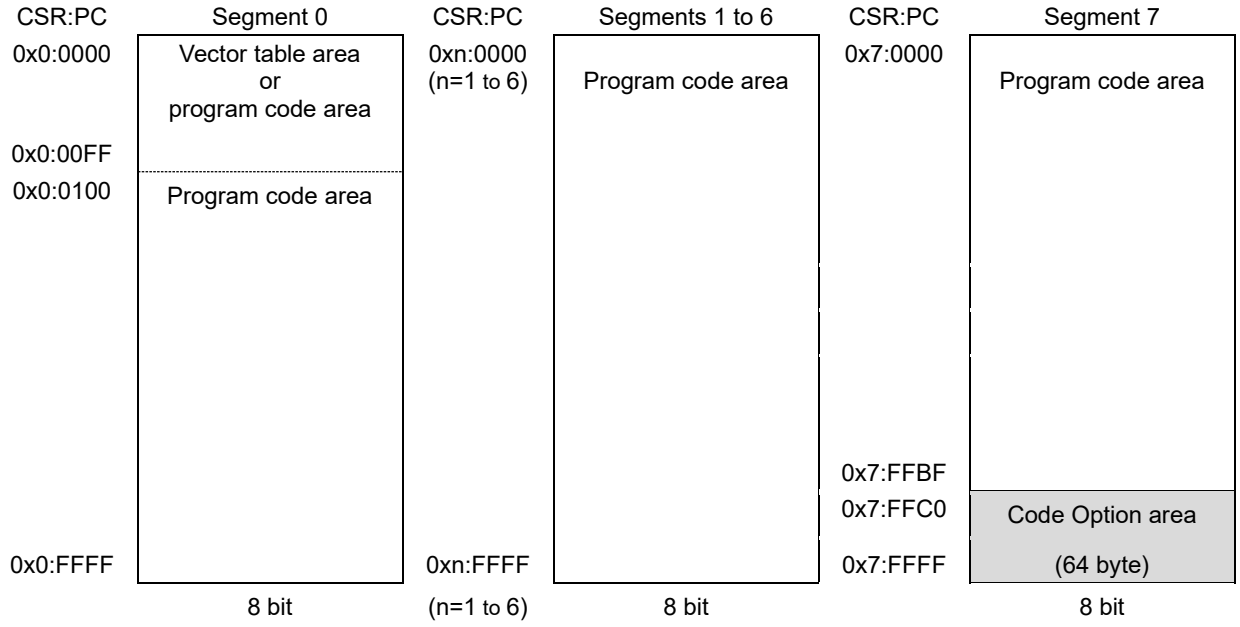
Figure 2-8 ML62Q1500/ML62Q1800 Group Configuration of Program Memory Space 6



ML62Q1858/1868/1878

Size 384 Kbyte

Figure 2-9 ML62Q1500/ML62Q1800 Group Configuration of Program Memory Space 7



ML62Q1859/1869/1879

Size 512 Kbyte

Figure 2-10 ML62Q1500/ML62Q1800 Group Configuration of Program Memory Space 8

■ ML62Q1700 group

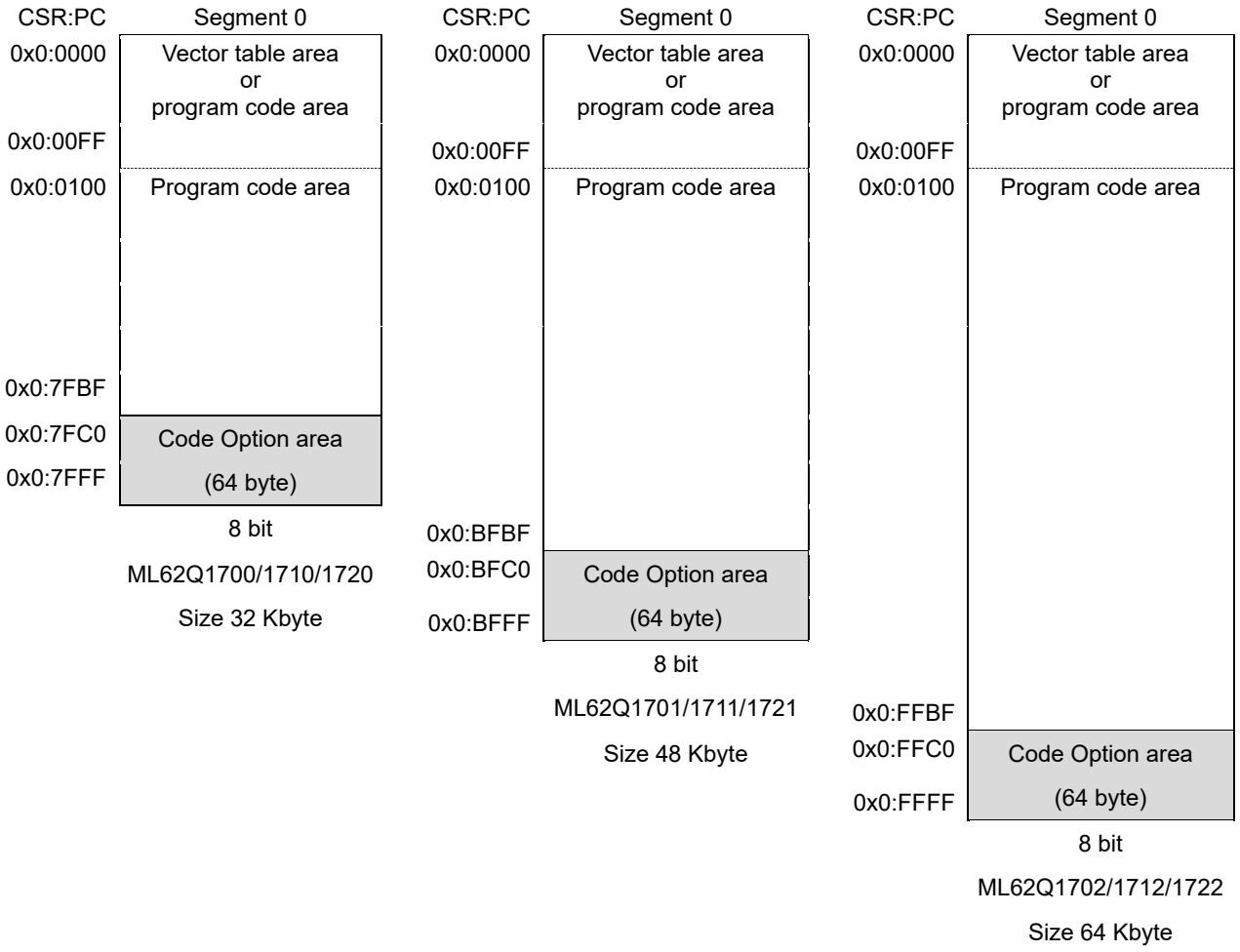
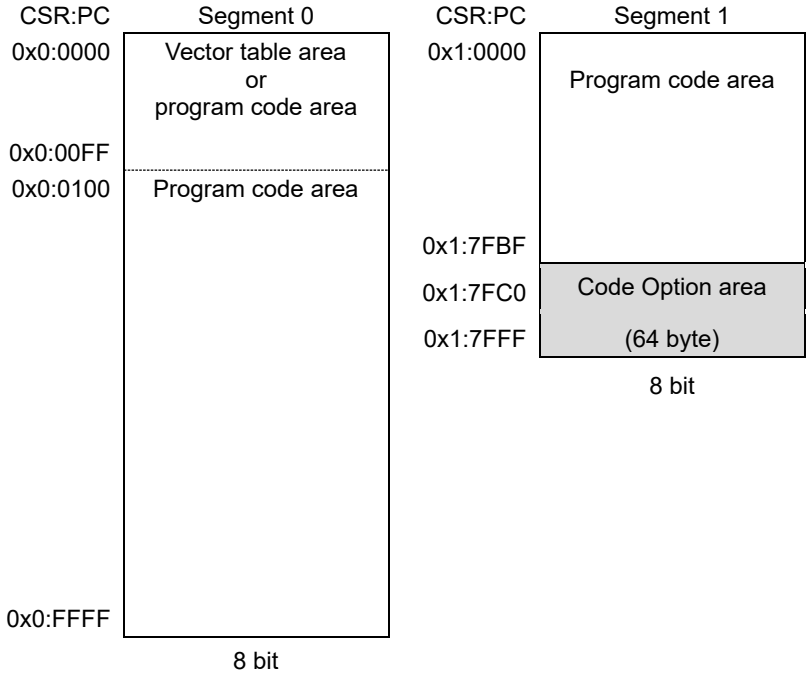


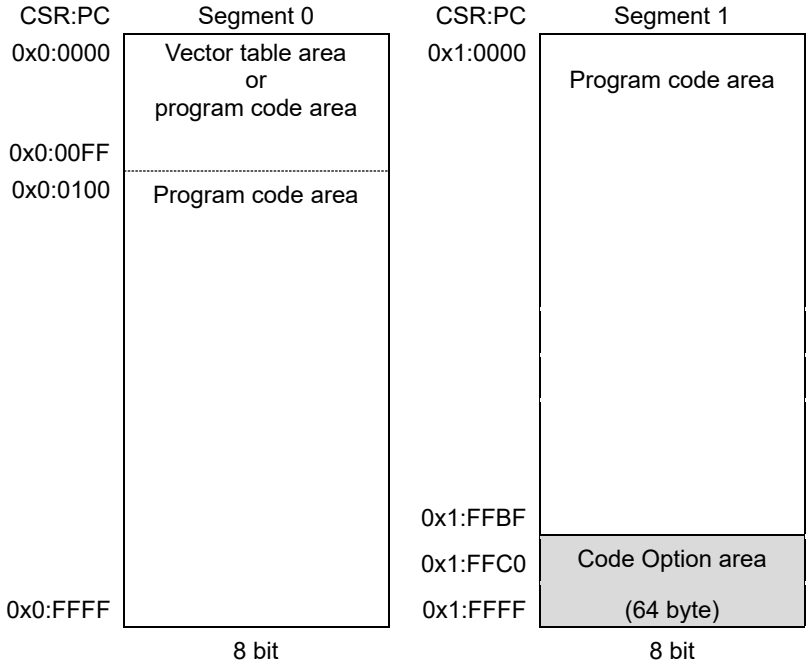
Figure 2-11 ML62Q1700 Group Configuration of Program Memory Space 1



ML62Q1703/1713/1723/1733/1743

Size 96 Kbyte

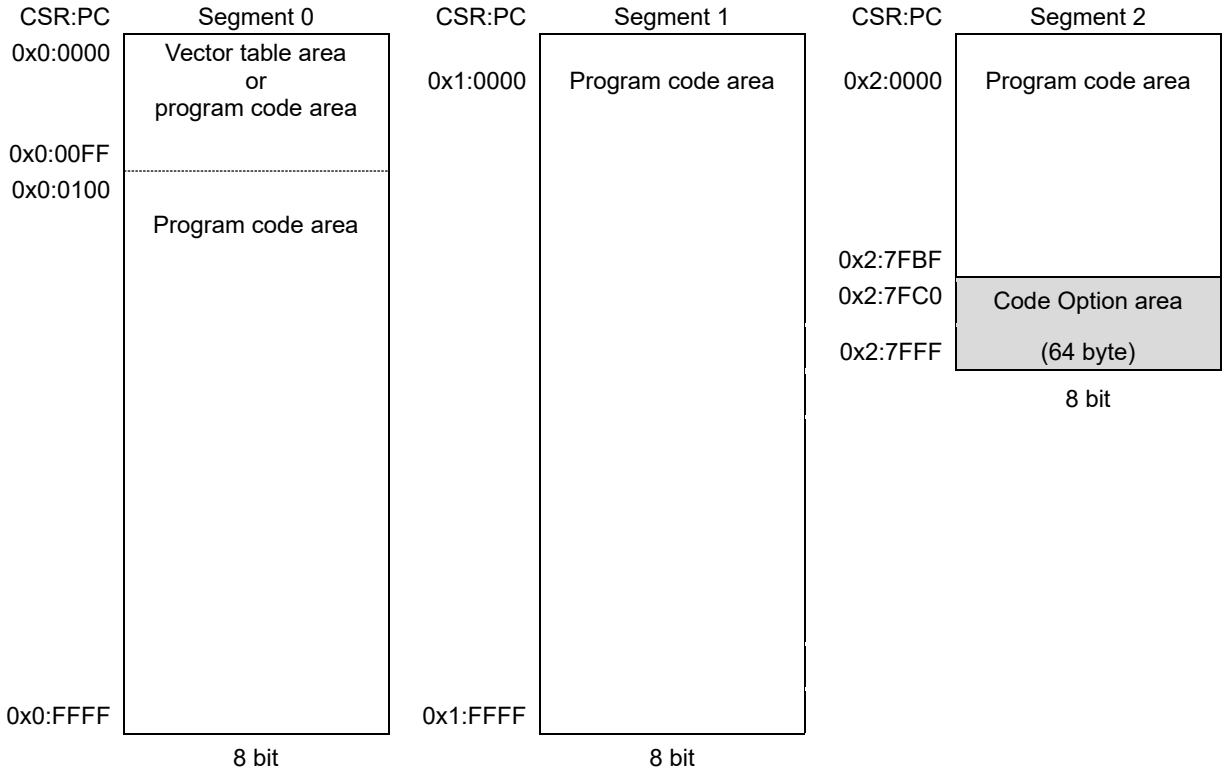
Figure 2-12 ML62Q1700 Group Configuration of Program Memory Space 2



ML62Q1704/1714/1724/1734/1744

Size 128 Kbyte

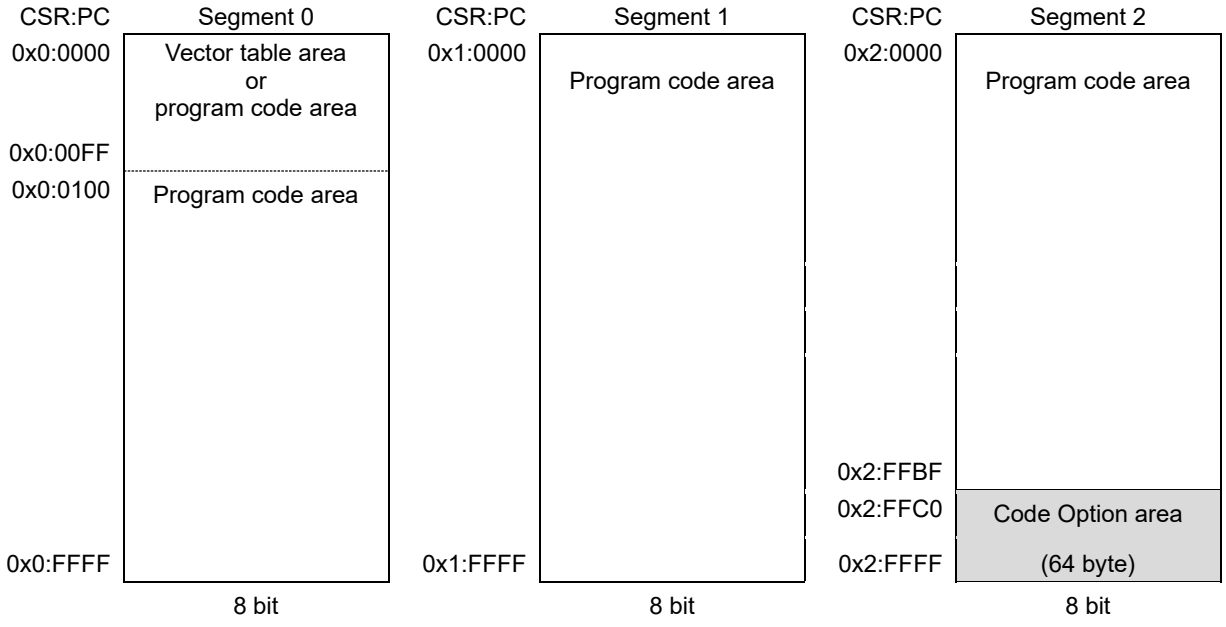
Figure 2-13 ML62Q1700 Group Configuration of Program Memory Space 3



ML62Q1725/1735/1745

Size 160 Kbyte

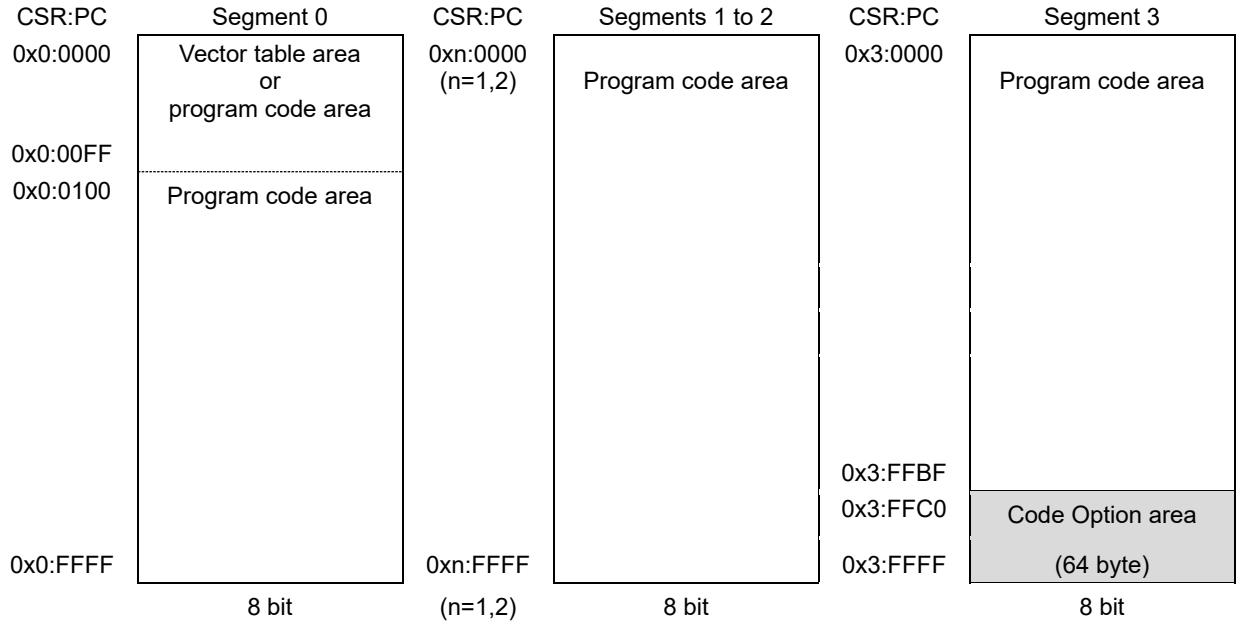
Figure 2-14 ML62Q1700 Group Configuration of Program Memory Space 4



ML62Q1726/1736/1746

Size 192 Kbyte

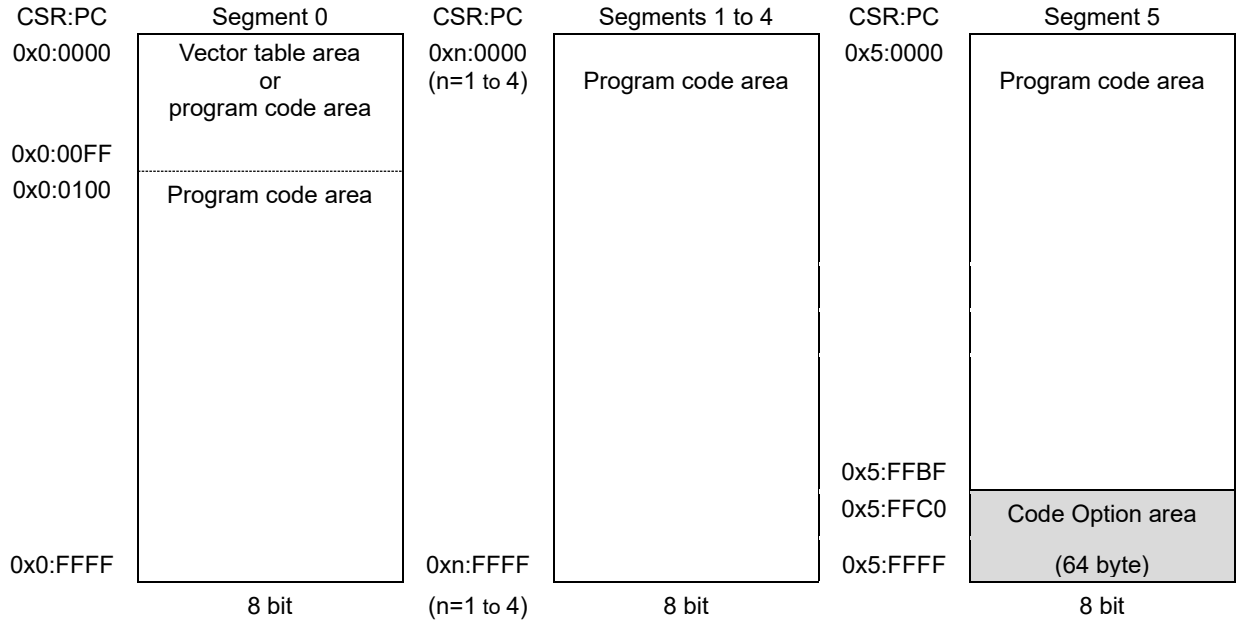
Figure 2-15 ML62Q1700 Group Configuration of Program Memory Space 5



ML62Q1727/1737/1747

Size 256 Kbyte

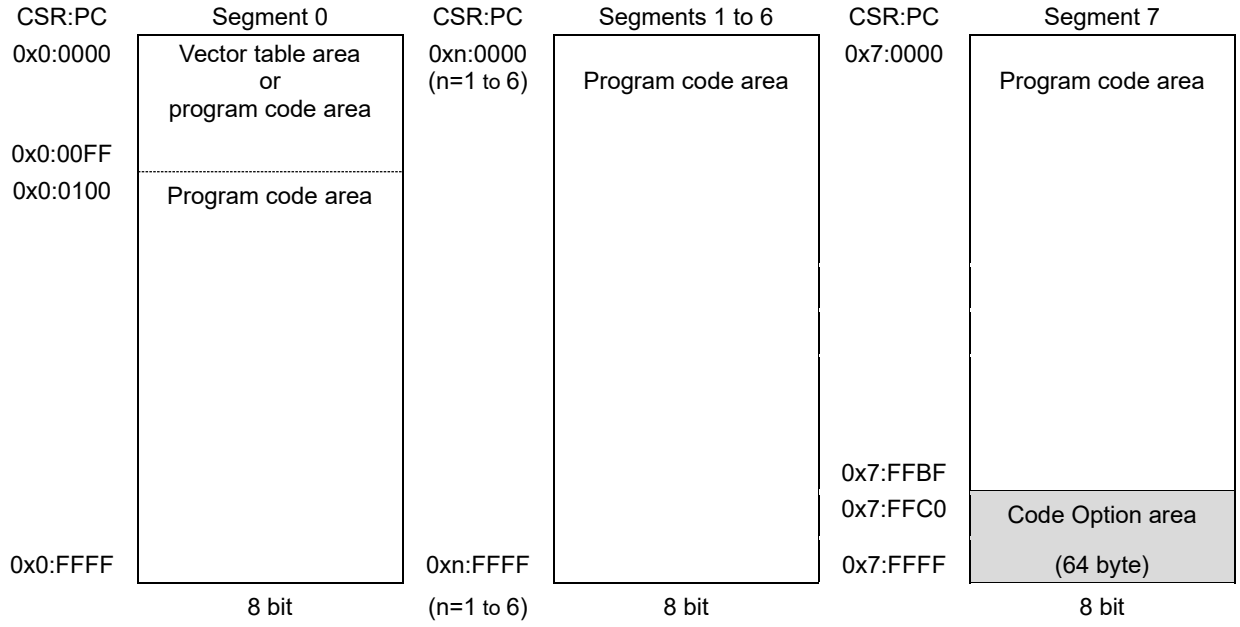
Figure 2-16 ML62Q1700 Group Configuration of Program Memory Space 6



ML62Q1728/1738/1748

Size 384 Kbyte

Figure 2-17 ML62Q1700 Group Configuration of Program Memory Space 7



ML62Q1729/1739/1749

Size 512 Kbyte

Figure 2-18 ML62Q1700 Group Configuration of Program Memory Space 8

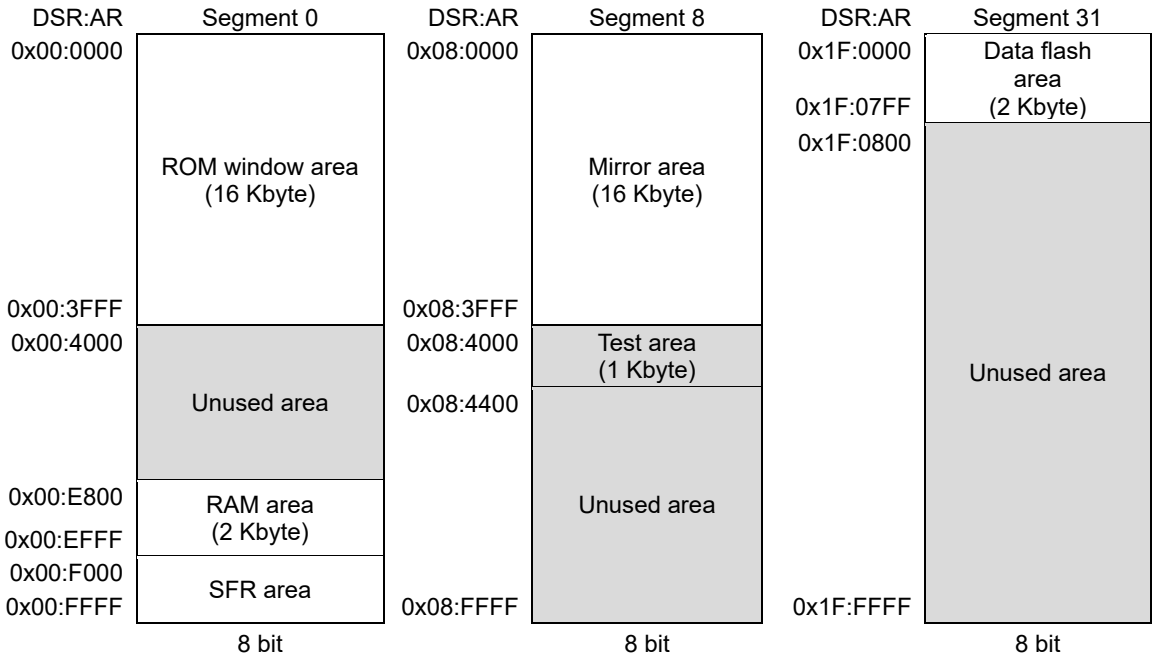
2.6 Data Memory Space

The data memory space consists of the segment 0 for ROM window area, RAM area, SFR area, segments 1 to 15 for mirror area, test area, and segment 31 for the data flash area.
The data memory stores 8-bit data and is specified by 21 bits consisting of higher 5 bits as the data segment register (DSR) and lower 16 bits as data address (address register: AR) specified by each instruction.
The segment 0 of program memory space and the segment of data memory space are in different space, but the segment 0 of program memory space is readable through the ROM window area of the data memory space.
The segment 1 to 7 are mirror area of segment 1 to 7 in the program memory space. The segment 8 to 15 are mirror area of segment 0 to 7 in the program memory space.
The 1K byte of test area includes device-specific data,
Figures 2-19 to 2-48 show the configuration of the data memory space of ML62Q1000 series products. Other segments not shown in the figures are unused areas.

[Note]

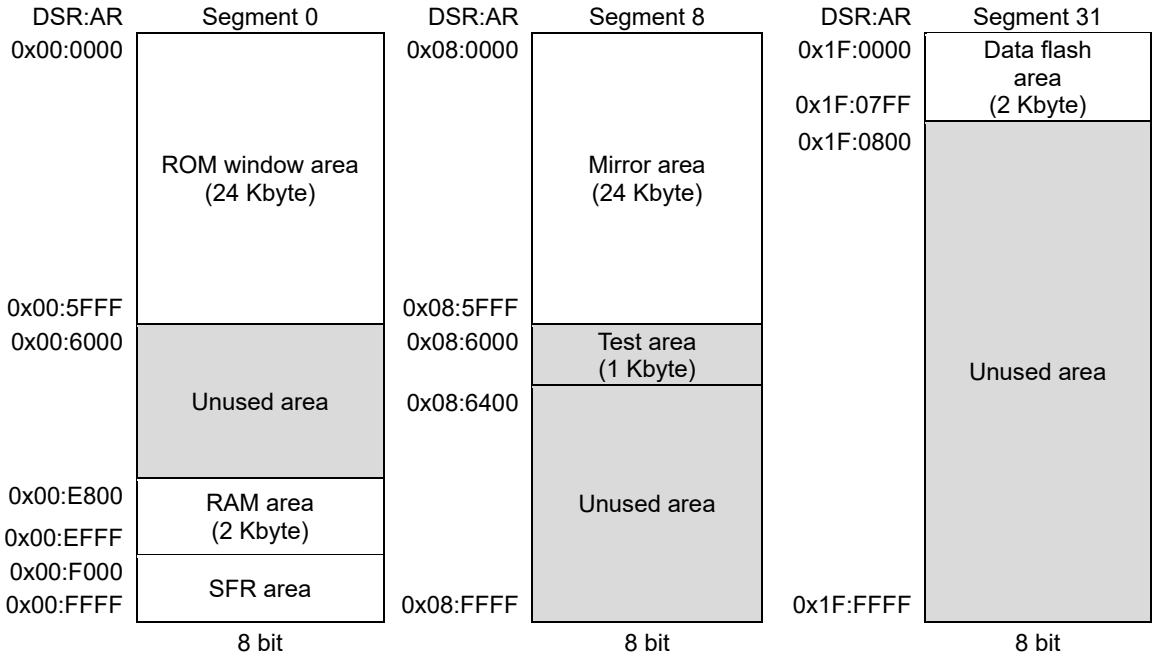
- **The contents of the RAM area are undefined at power-on and system reset. Initialize this area by the software.**
- **Do not read/write unused areas to prevent the CPU works incorrectly.**

■ ML62Q1300 group



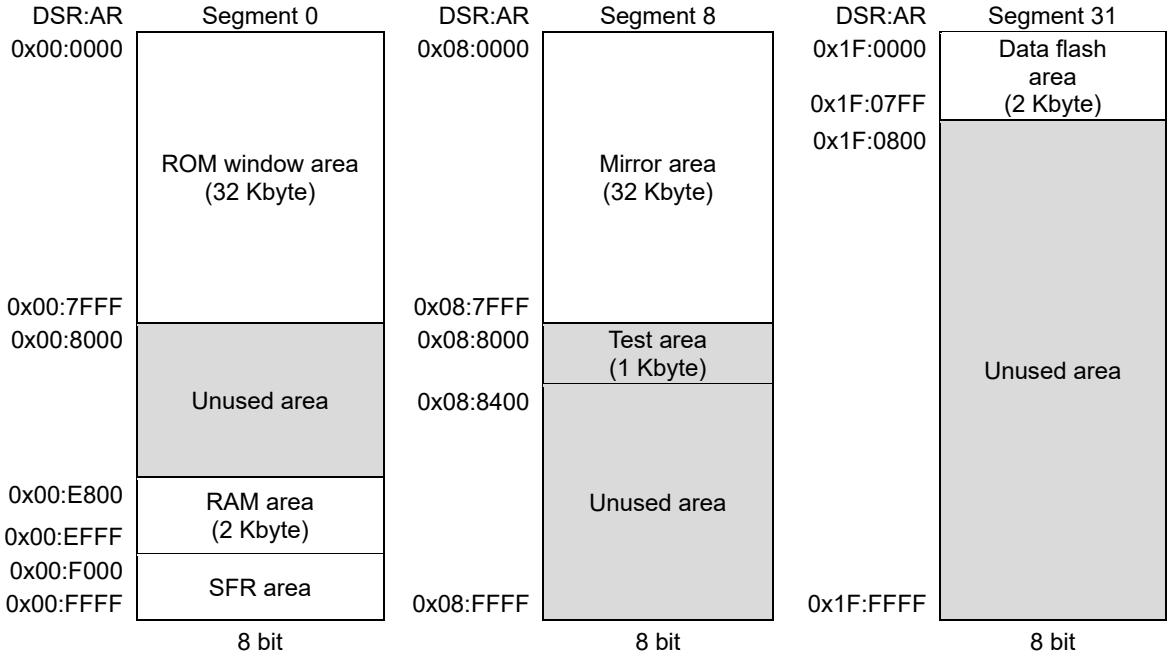
ML62Q1323/1333

Figure 2-19 ML62Q1300 Group Configuration of Data Memory Space 1



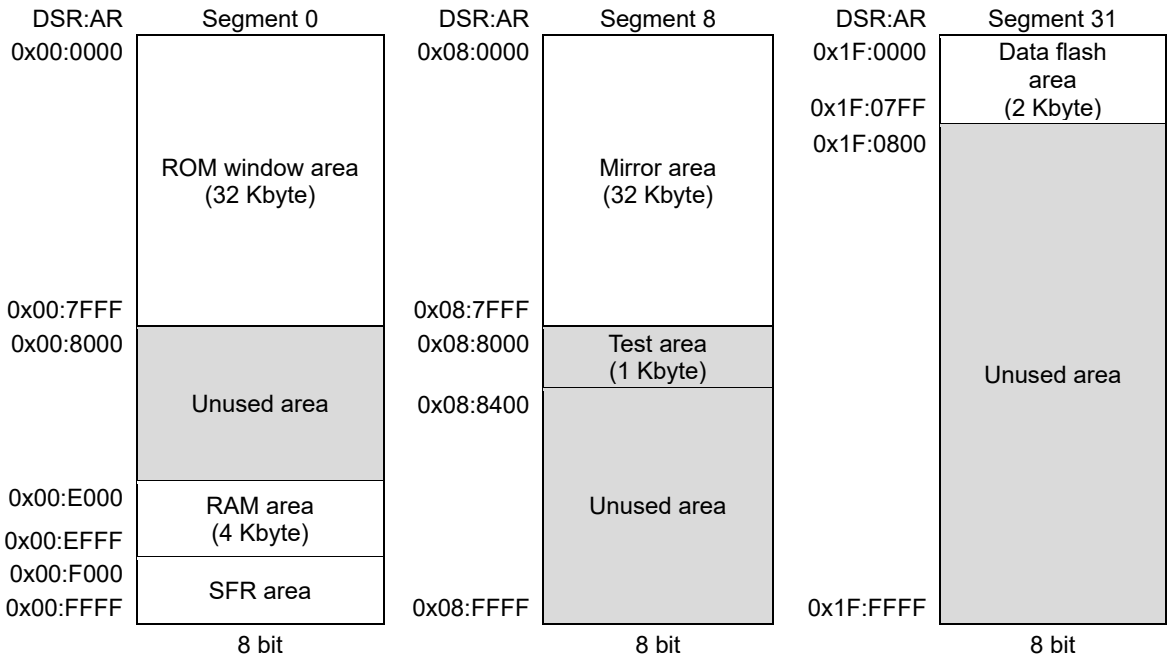
ML62Q1324/1334

Figure 2-20 ML62Q1300 Group Configuration of Data Memory Space 2



ML62Q1325/1335

Figure 2-21 ML62Q1300 Group Configuration of Data Memory Space 3



ML62Q1345/1365

Figure 2-22 ML62Q1300 Group Configuration of Data Memory Space 4

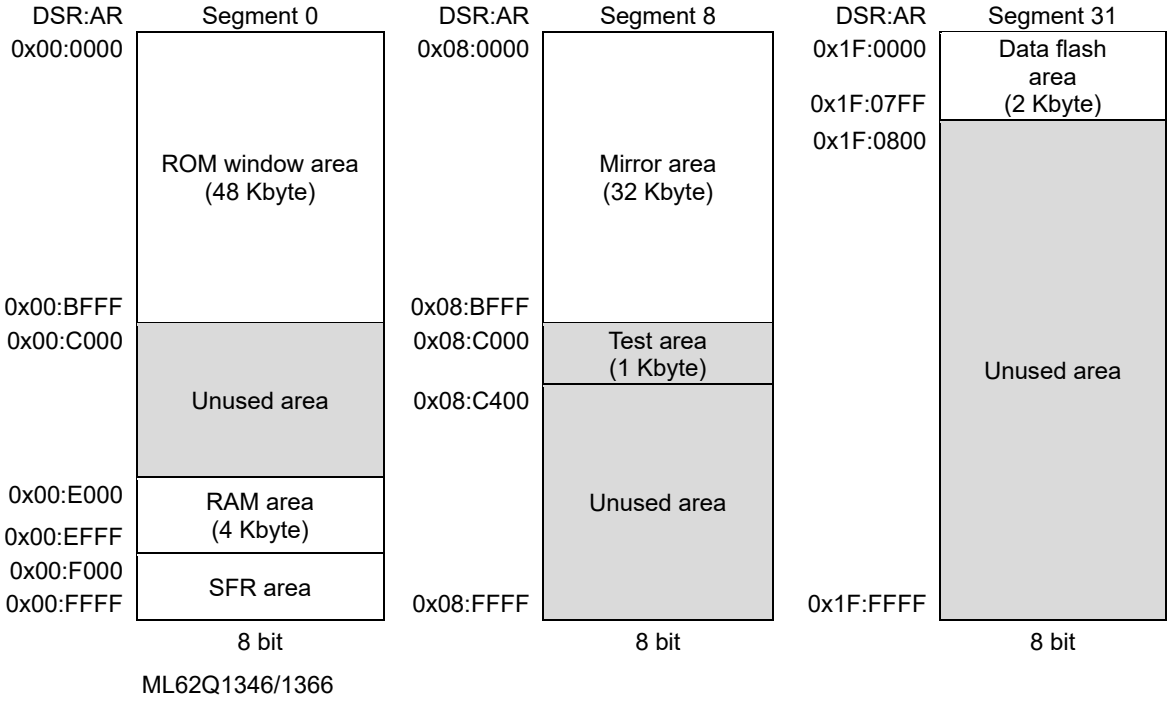


Figure 2-23 ML62Q1300 Group Configuration of Data Memory Space 5

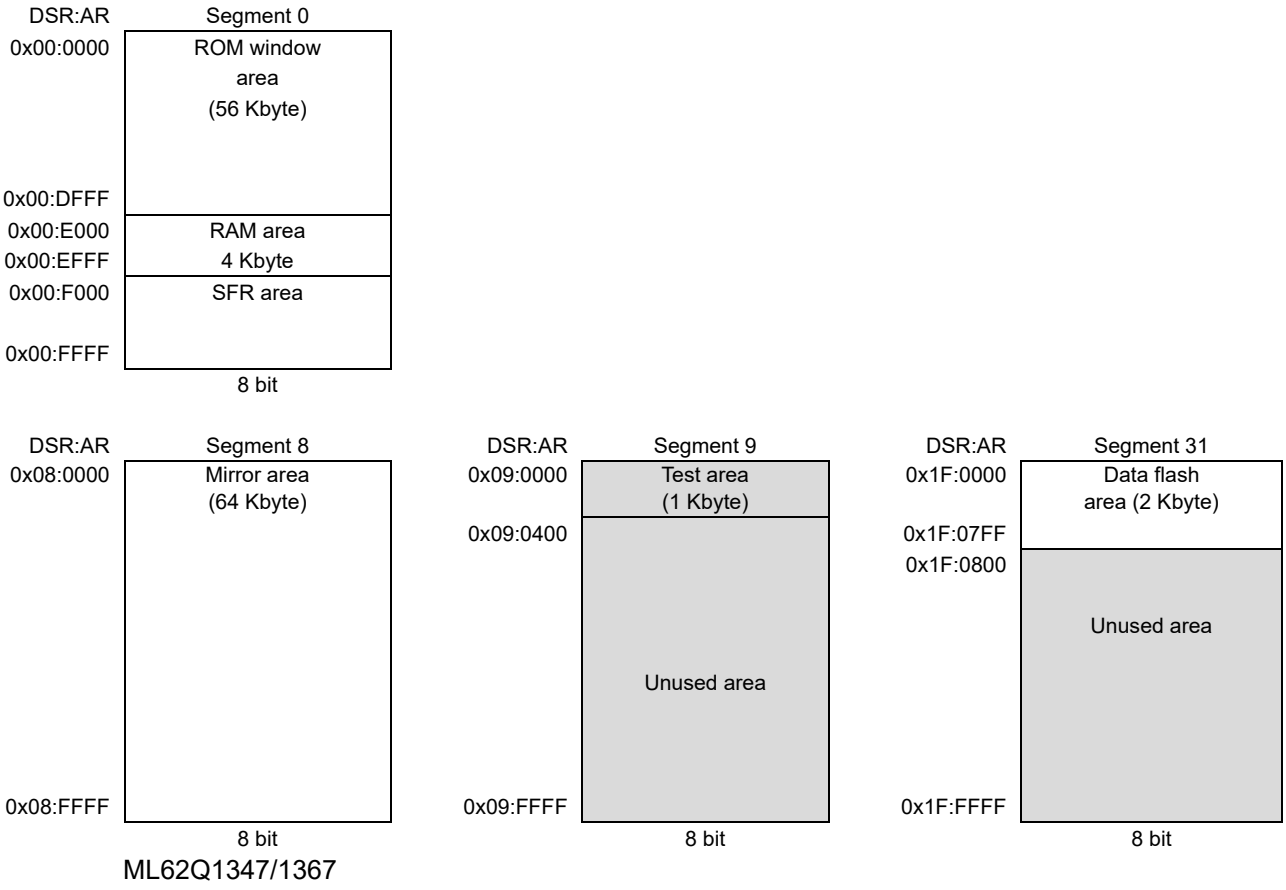
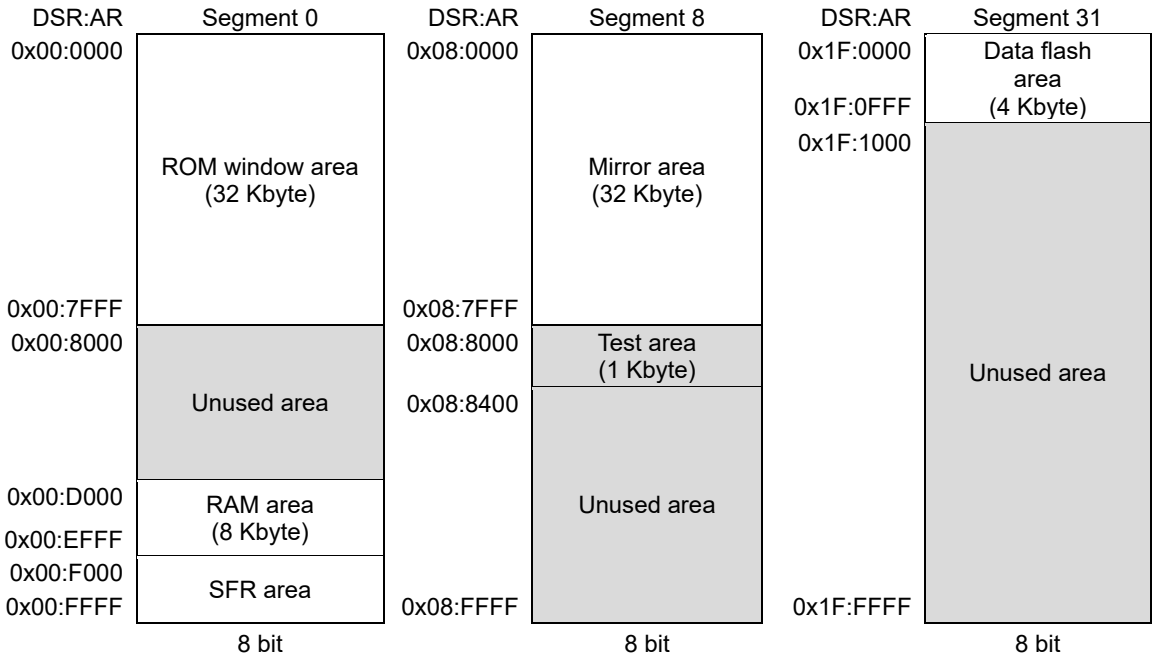


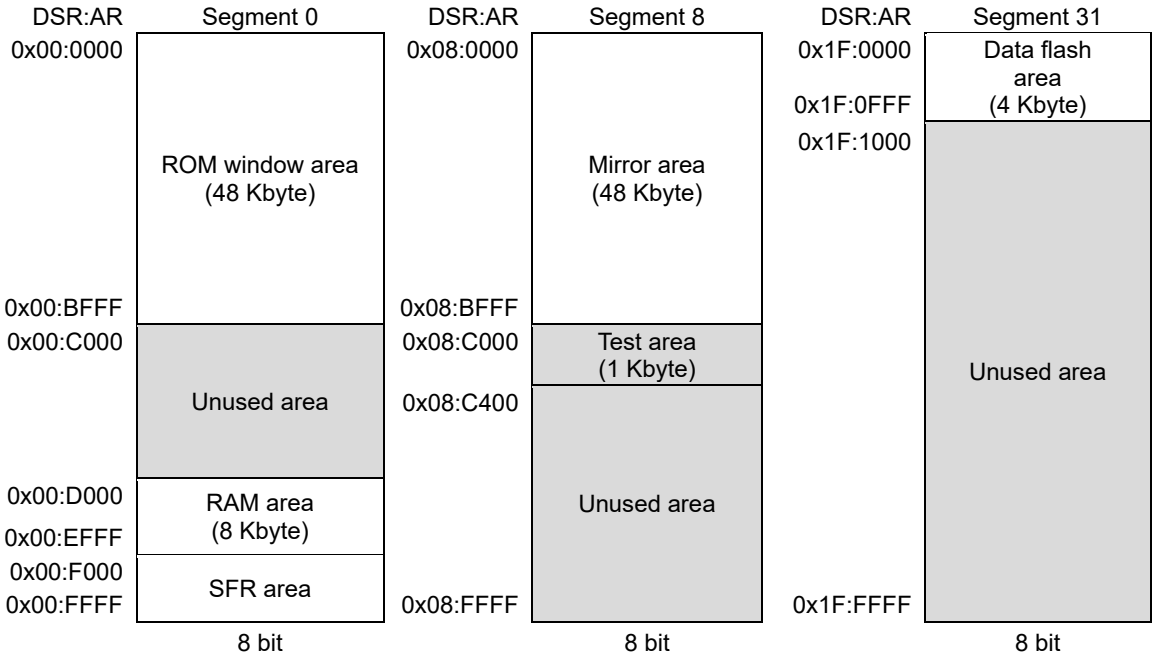
Figure 2-24 ML62Q1300 Group Configuration of Data Memory Space 6

■ ML62Q1500/ML62Q1800 group



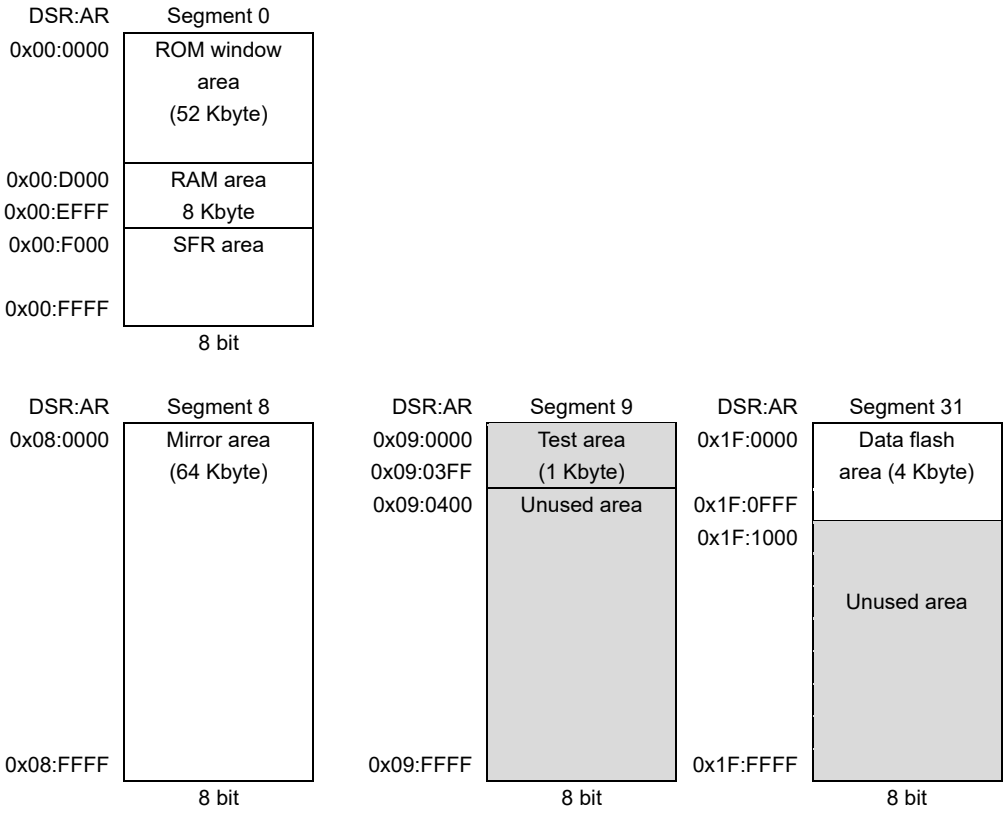
ML62Q1530/1540/1550

Figure 2-25 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 1



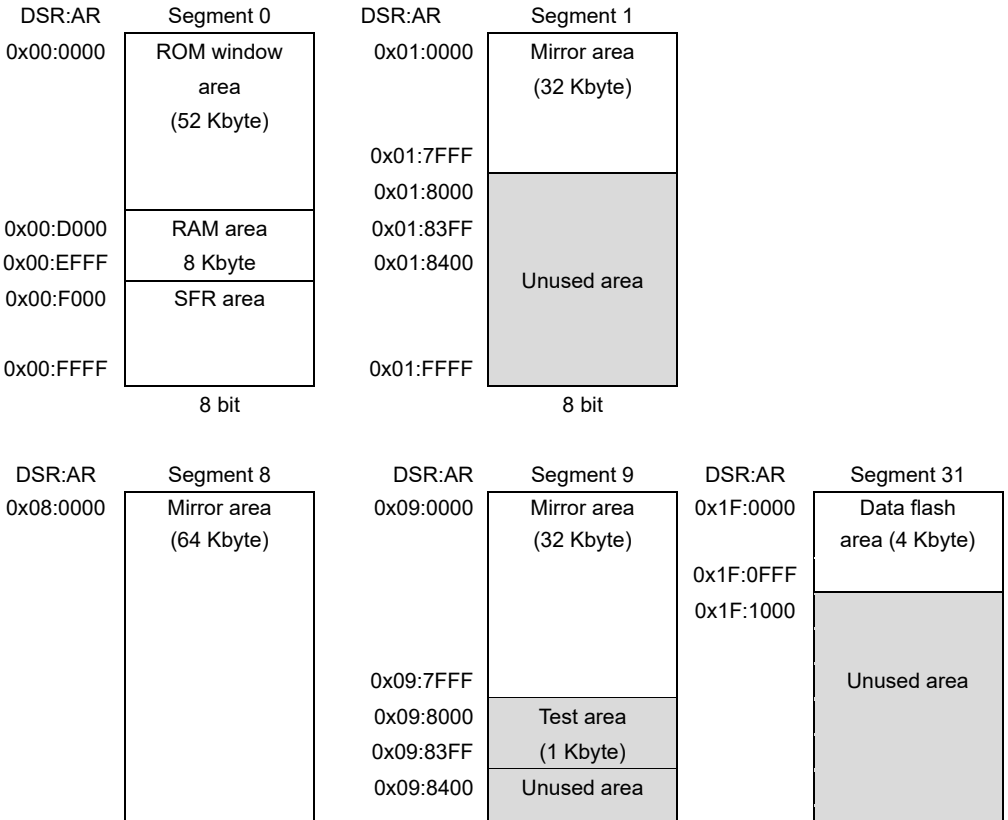
ML62Q1531/1541/1551

Figure 2-26 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 2



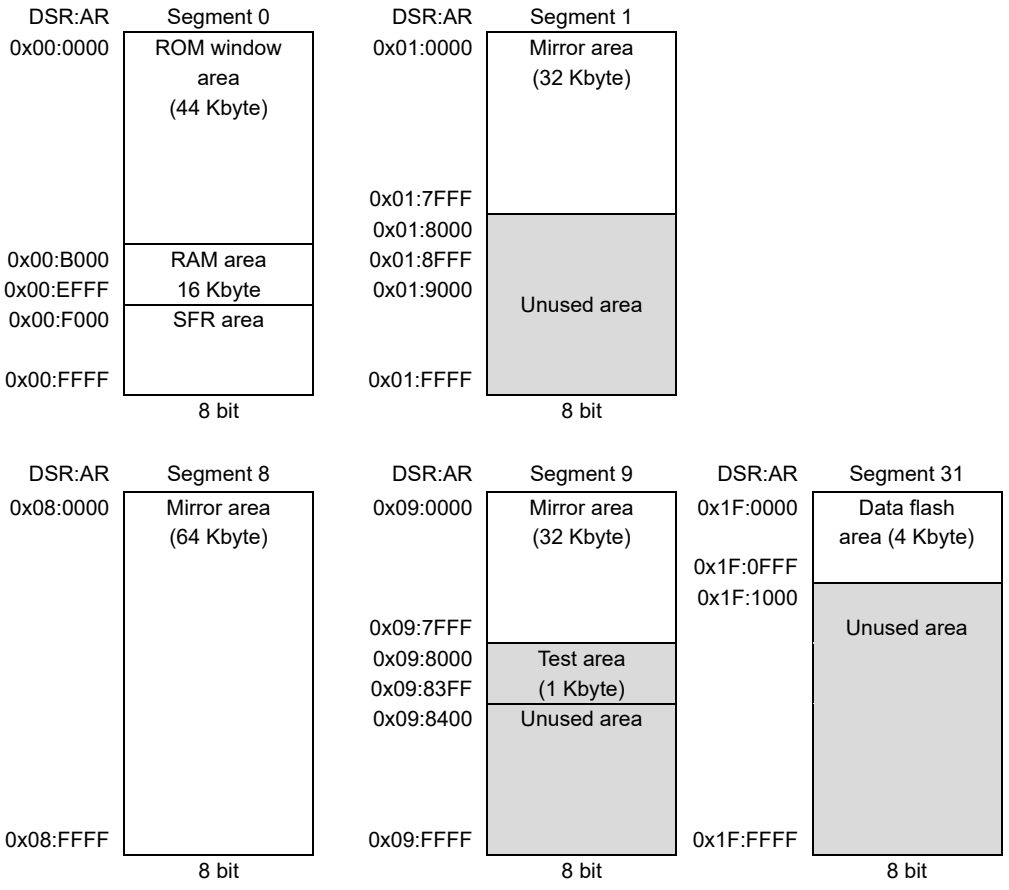
ML62Q1532/1542/1552

Figure 2-27 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 3



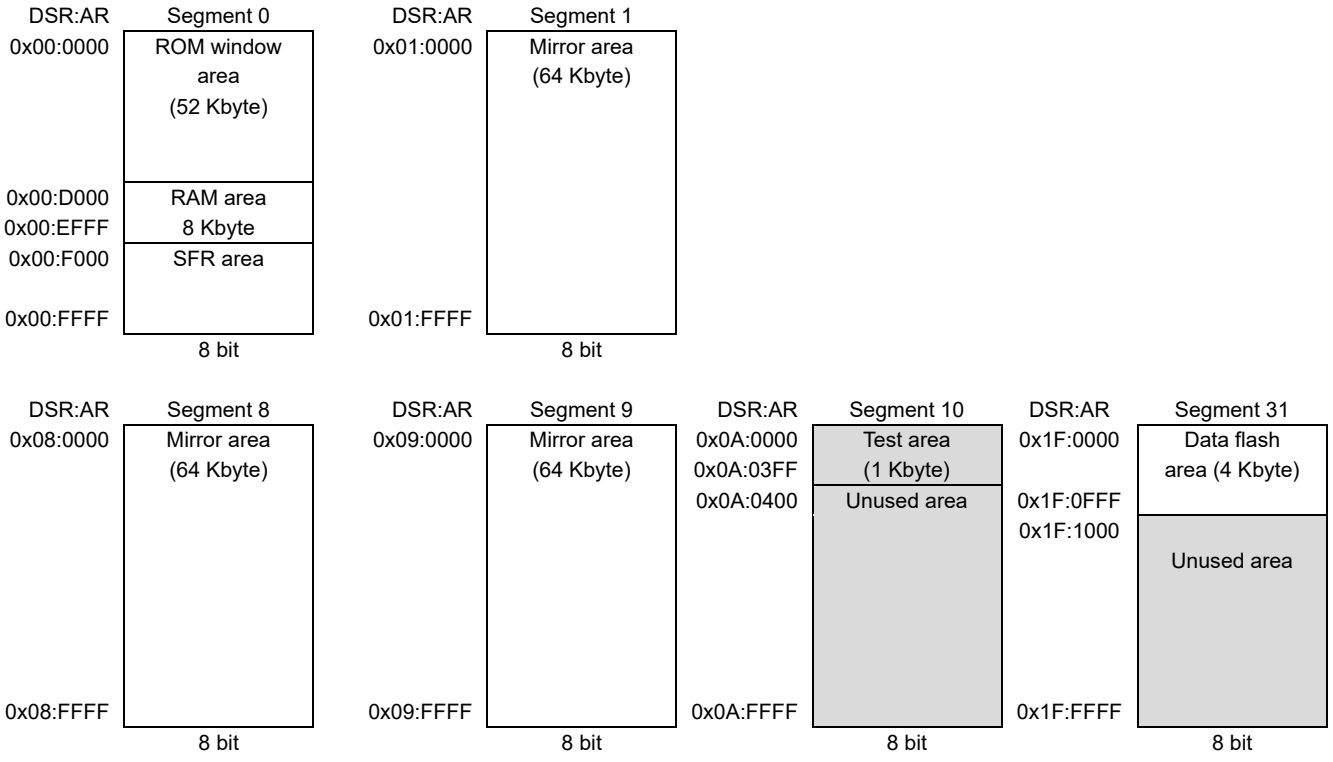
ML62Q1533/1543/1553

Figure 2-28 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 4



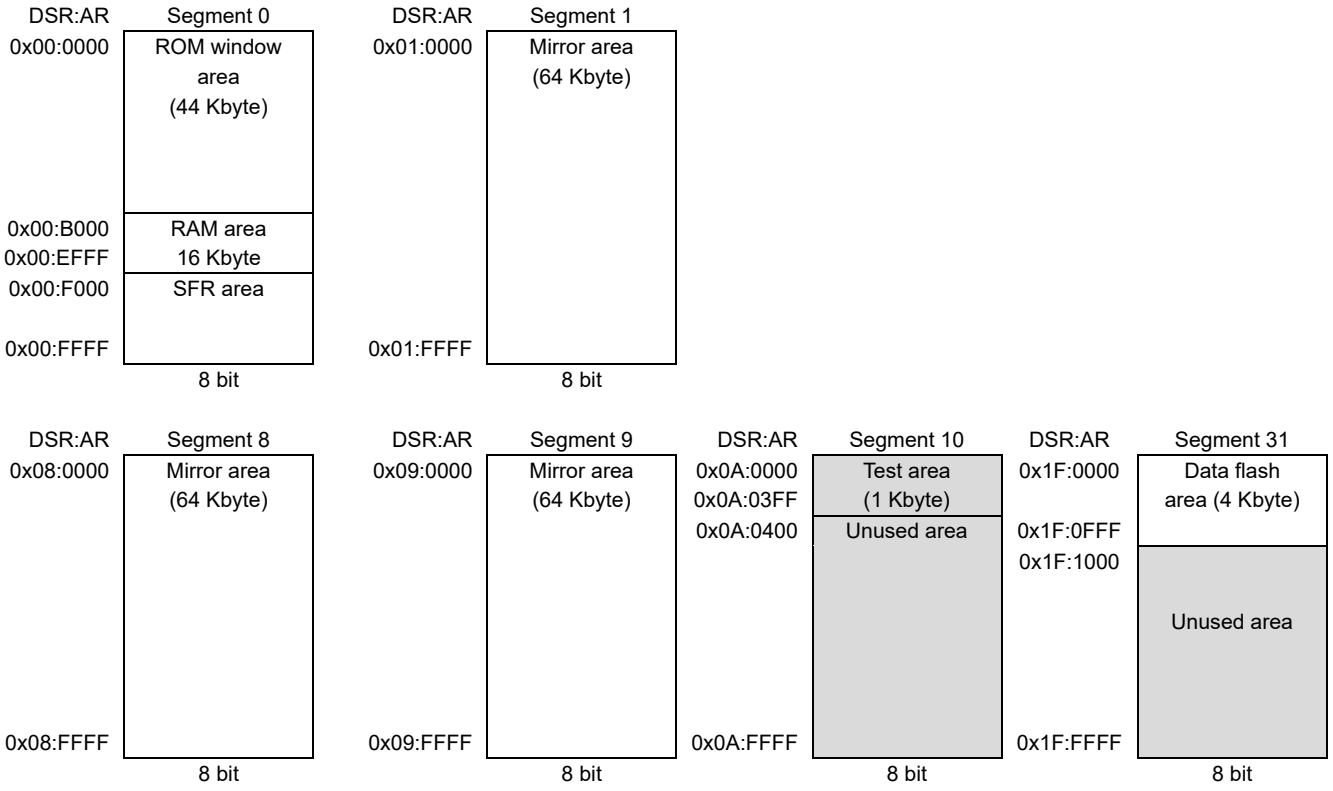
ML62Q1563/1573

Figure 2-29 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 5



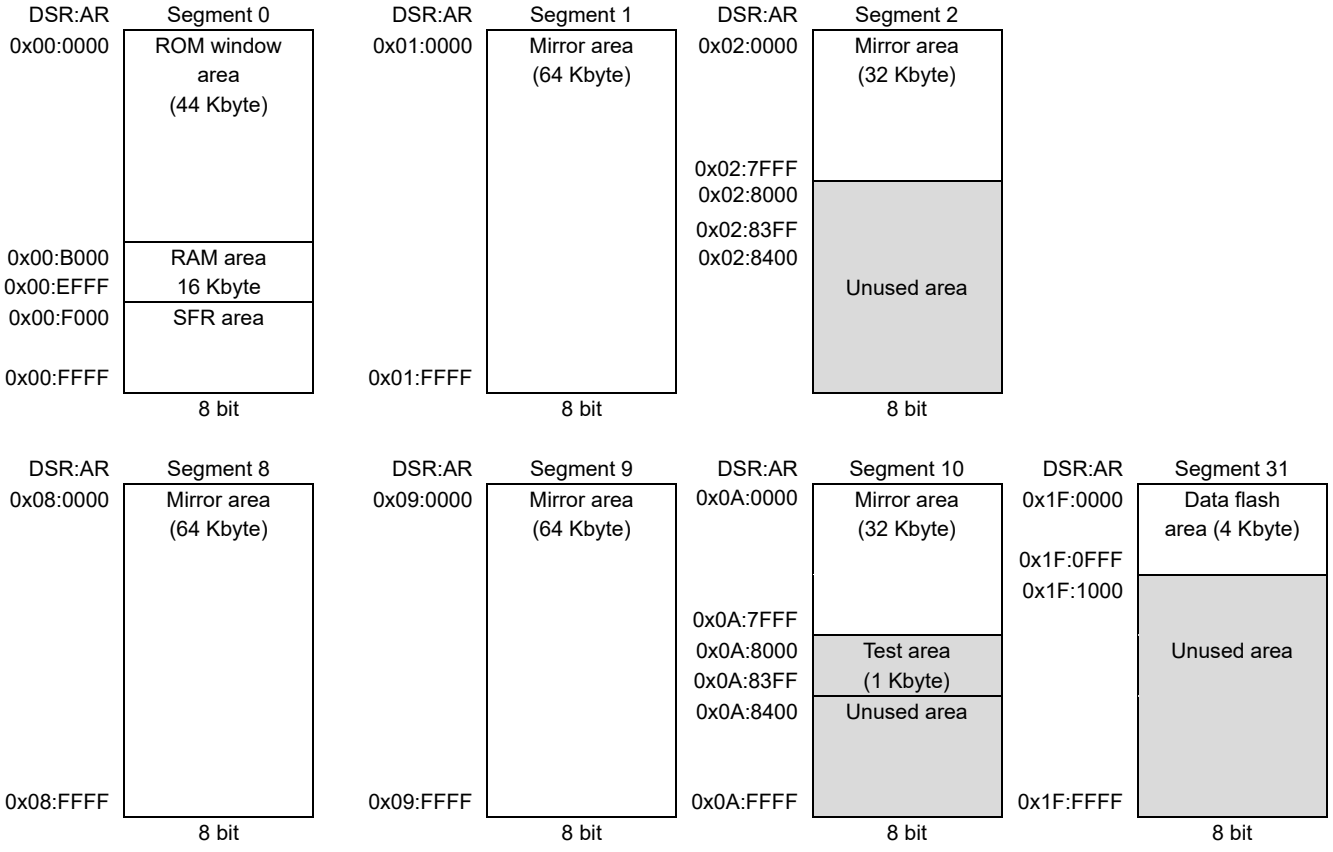
ML62Q1534/1544/1554

Figure 2-30 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 6



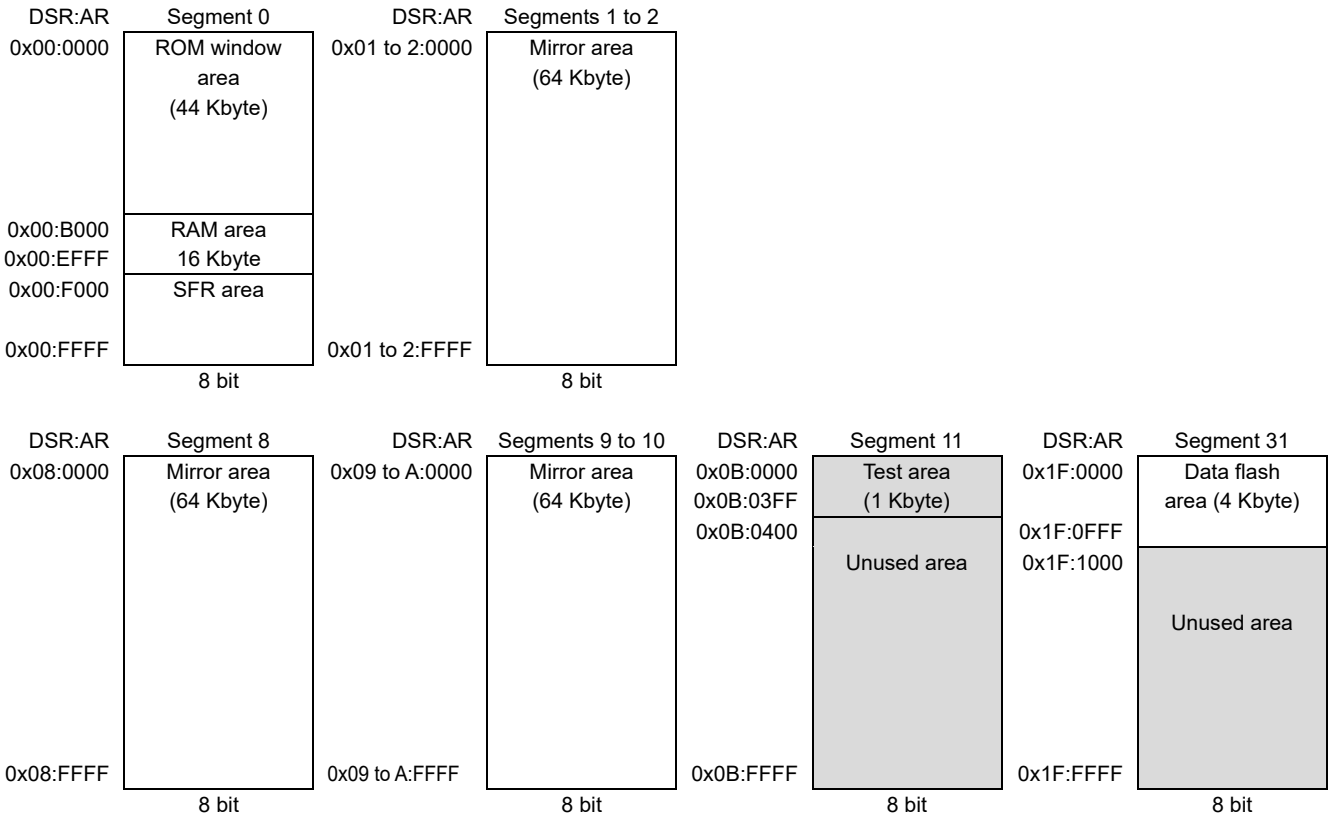
ML62Q1564/1574

Figure 2-31 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 7



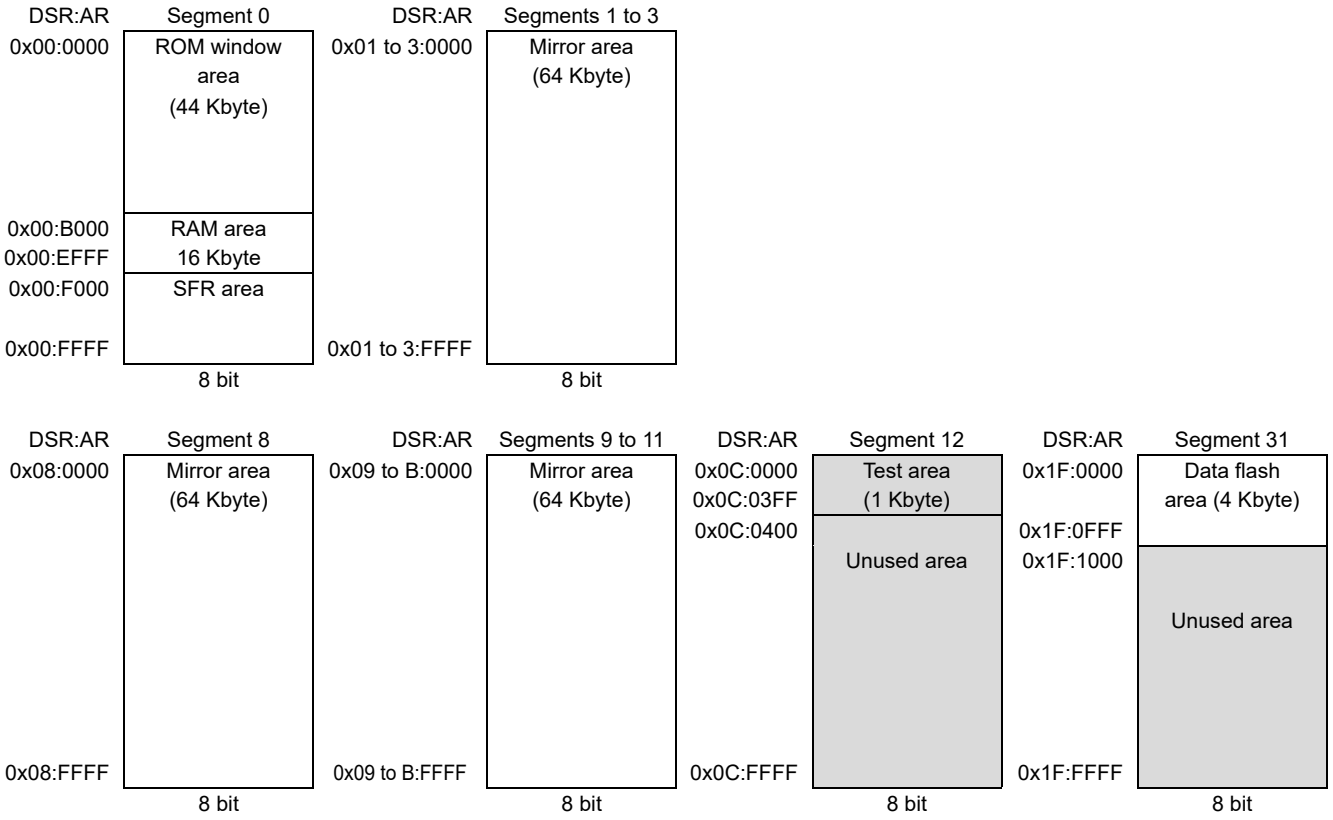
ML62Q1555/1565/1575

Figure 2-32 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 8



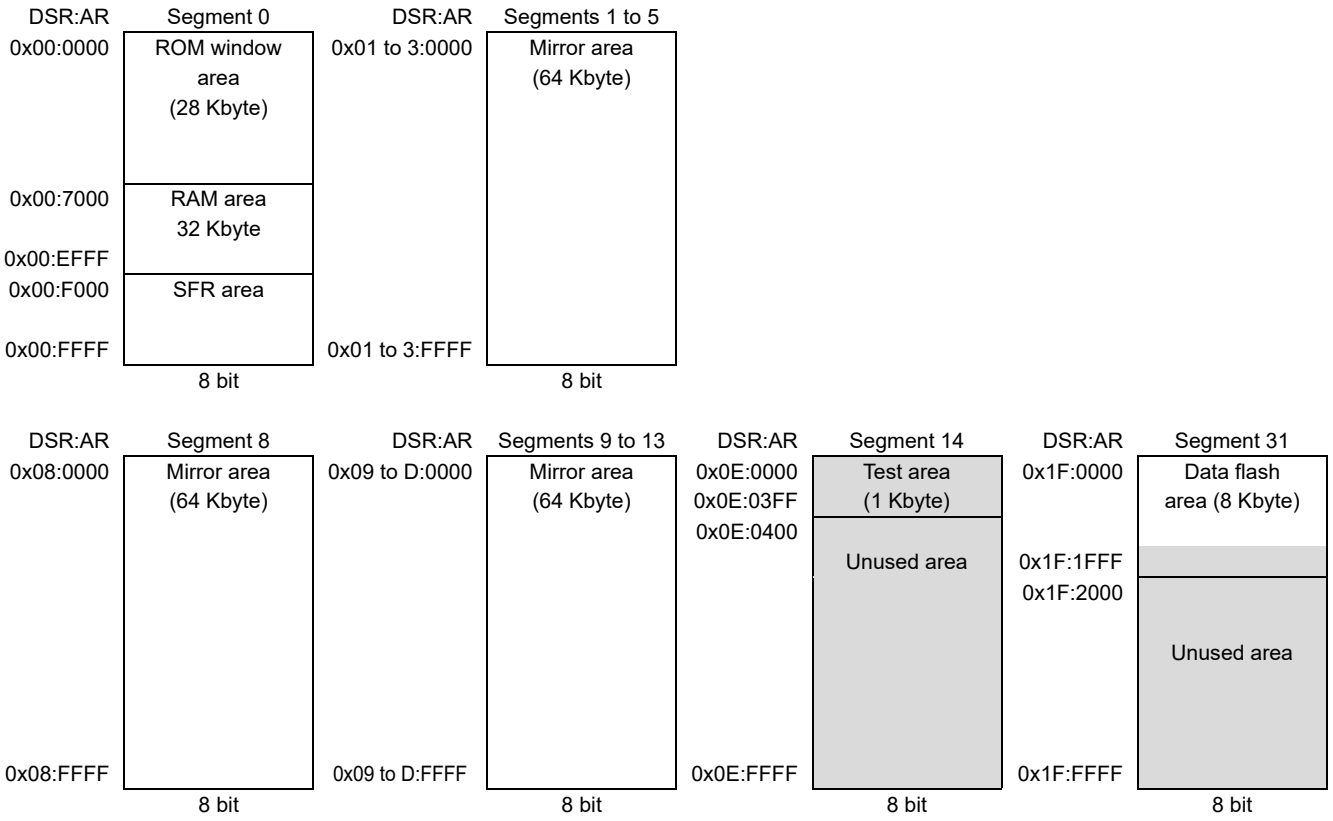
ML62Q1556/1566/1576

Figure 2-33 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 9



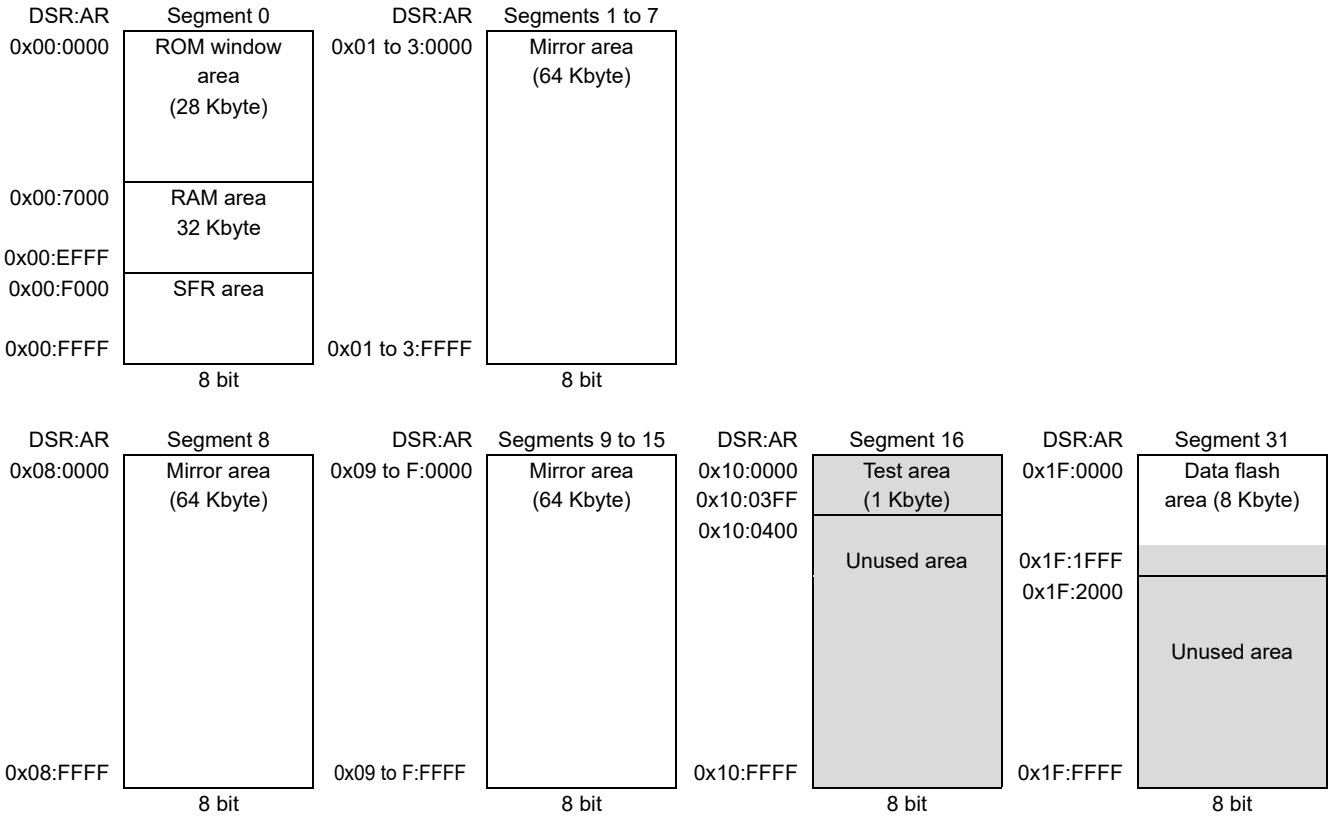
ML62Q1557/1567/1577

Figure 2-34 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 10



ML62Q1858/1868/1878

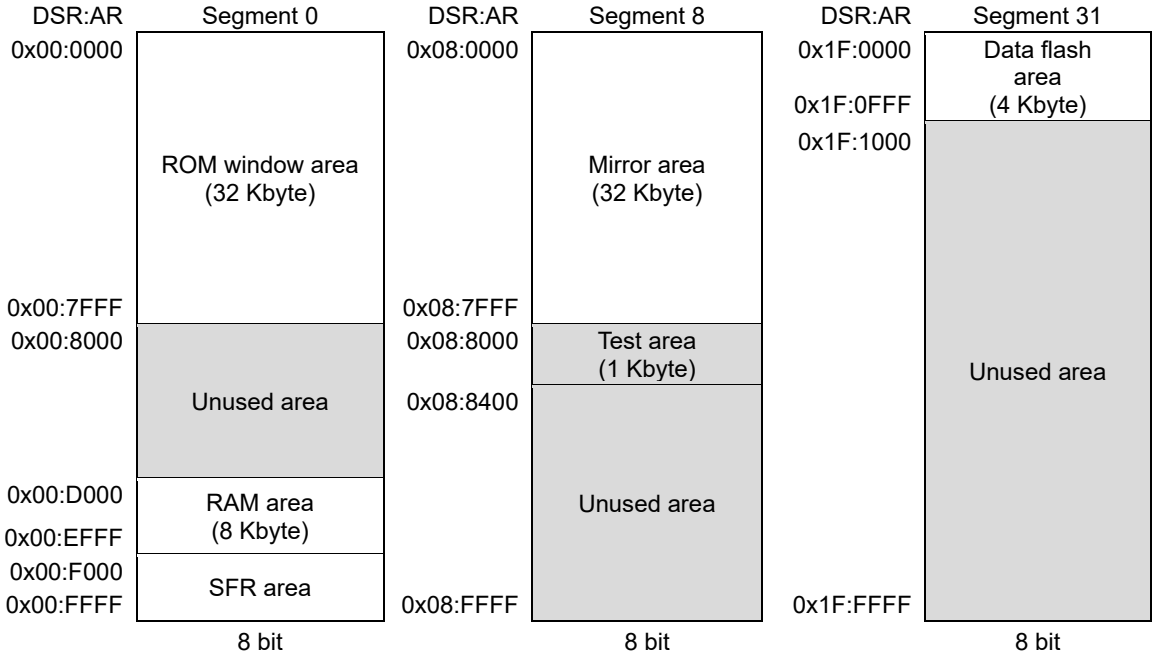
Figure 2-35 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 11



ML62Q1859/1869/1879

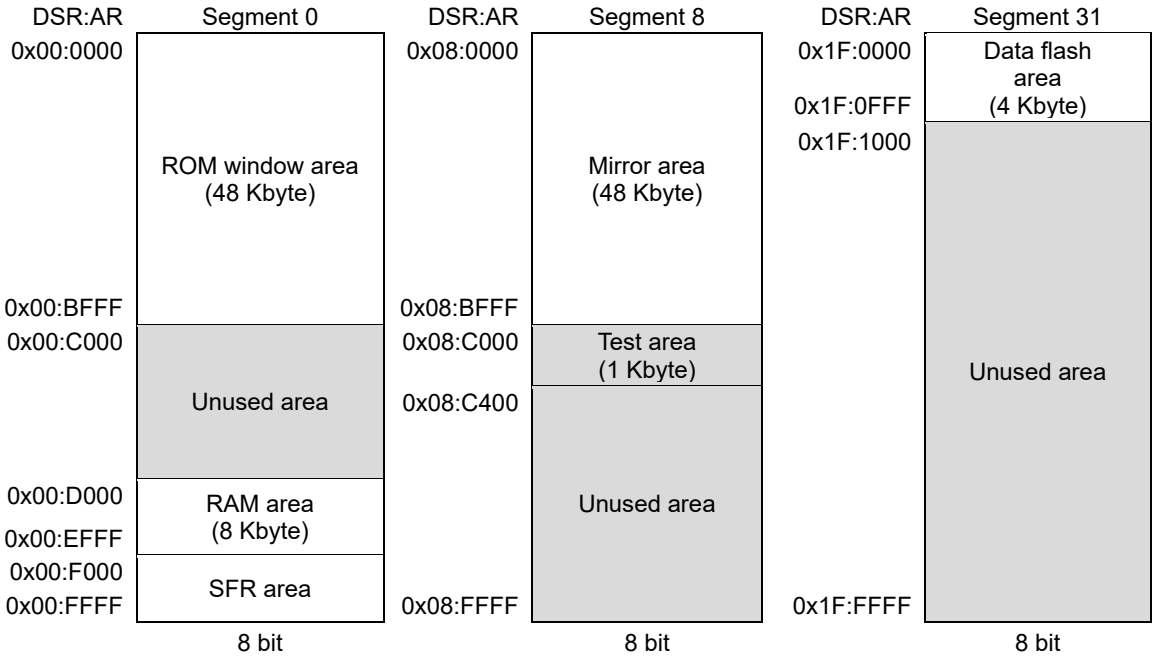
Figure 2-36 ML62Q1500/ML62Q1800 Group Configuration of Data Memory Space 12

■ ML62Q1700 group



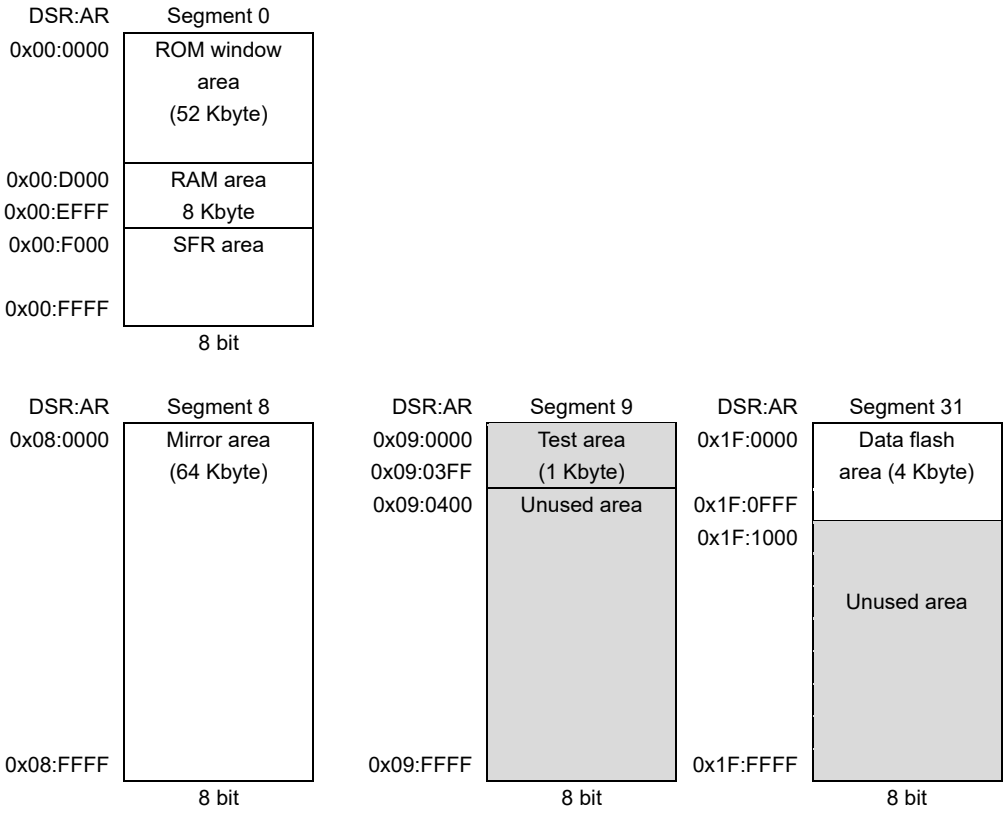
ML62Q1700/1710/1720

Figure 2-37 ML62Q1700 Group Configuration of Data Memory Space 1



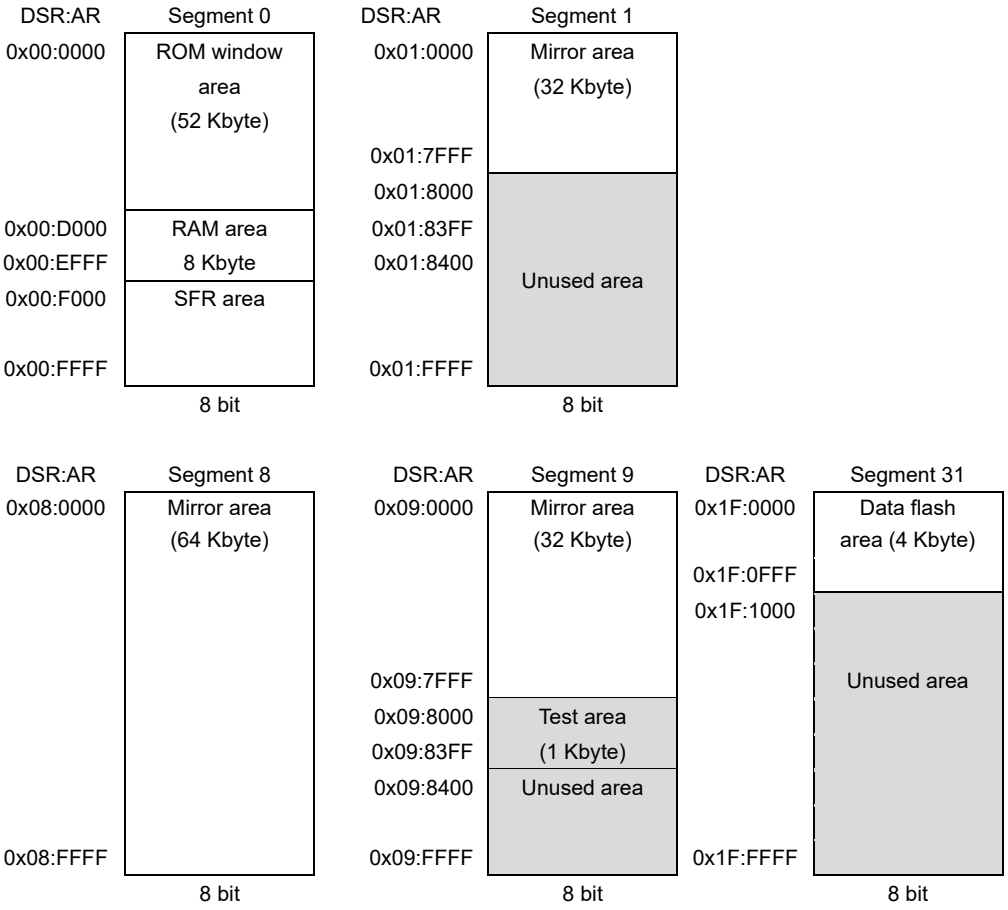
ML62Q1701/1711/1721

Figure 2-38 ML62Q1700 Group Configuration of Data Memory Space 2



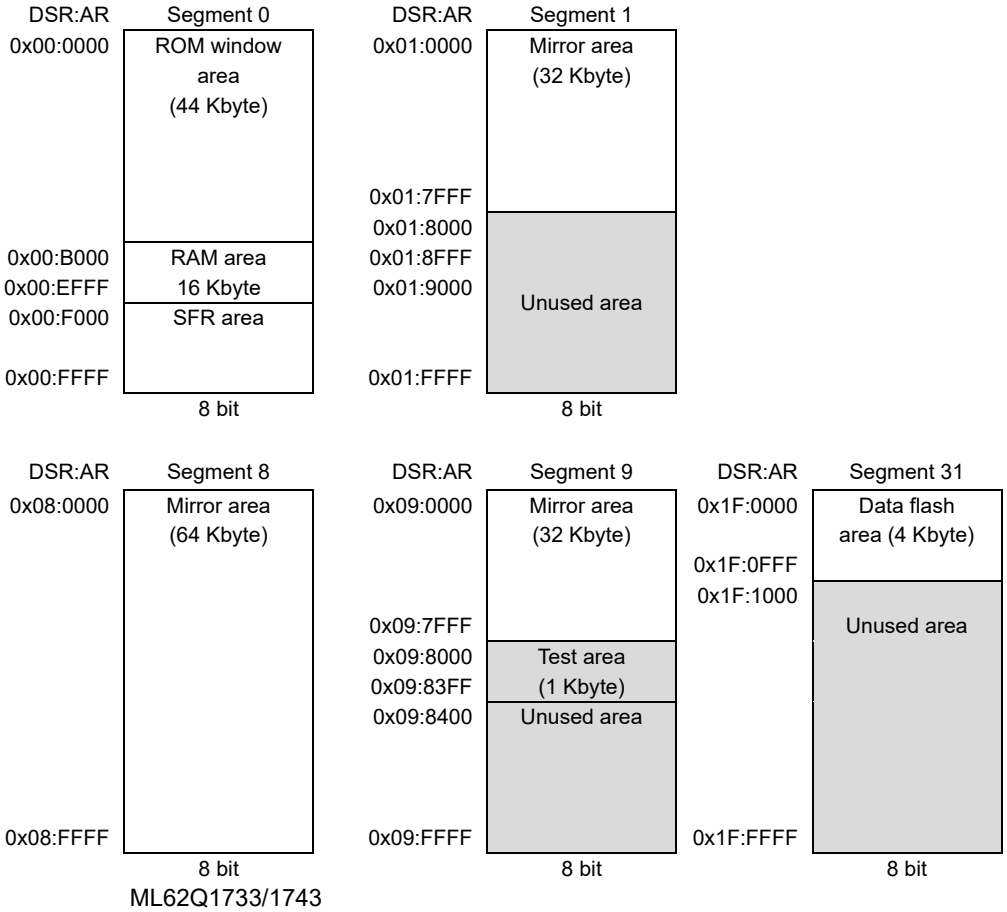
ML62Q1702/1712/1722

Figure 2-39 ML62Q1700 Group Configuration of Data Memory Space 3



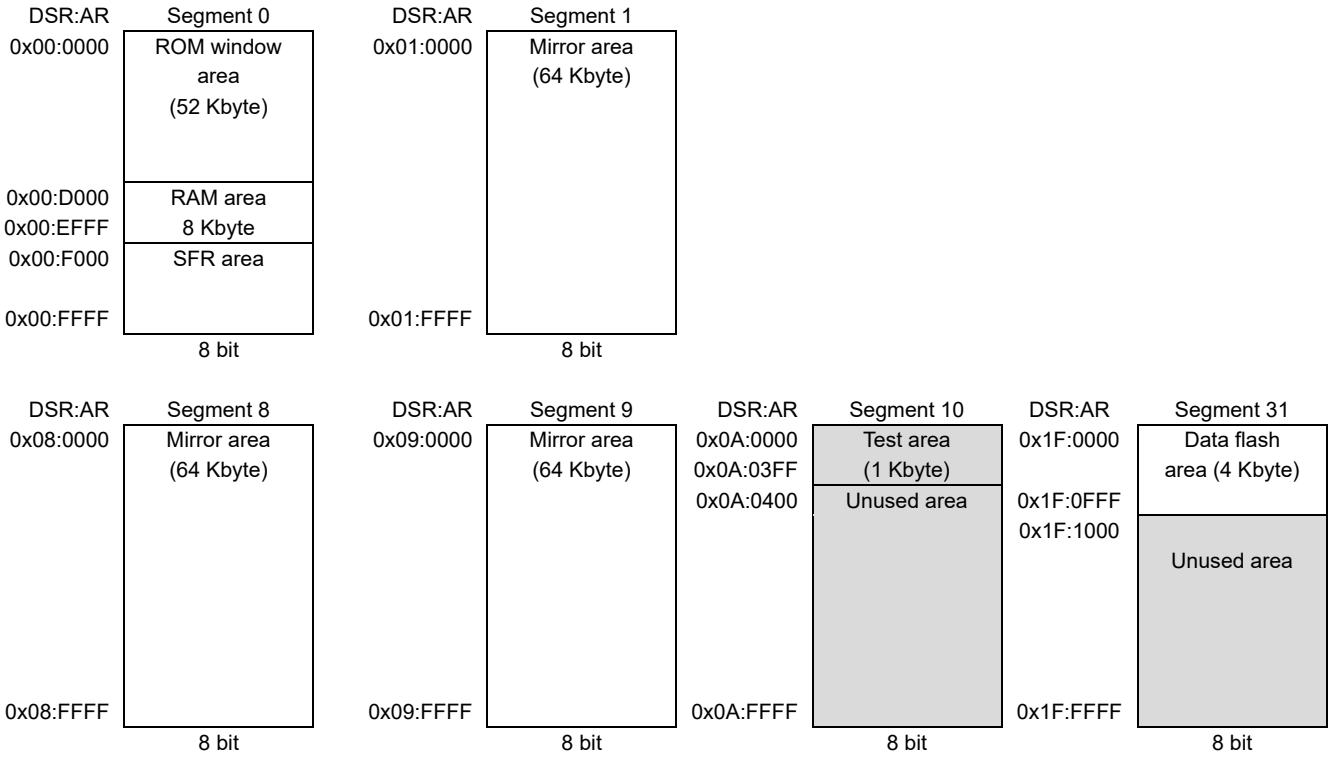
ML62Q1703/1713/1723

Figure 2-40 ML62Q1700 Group Configuration of Data Memory Space 4



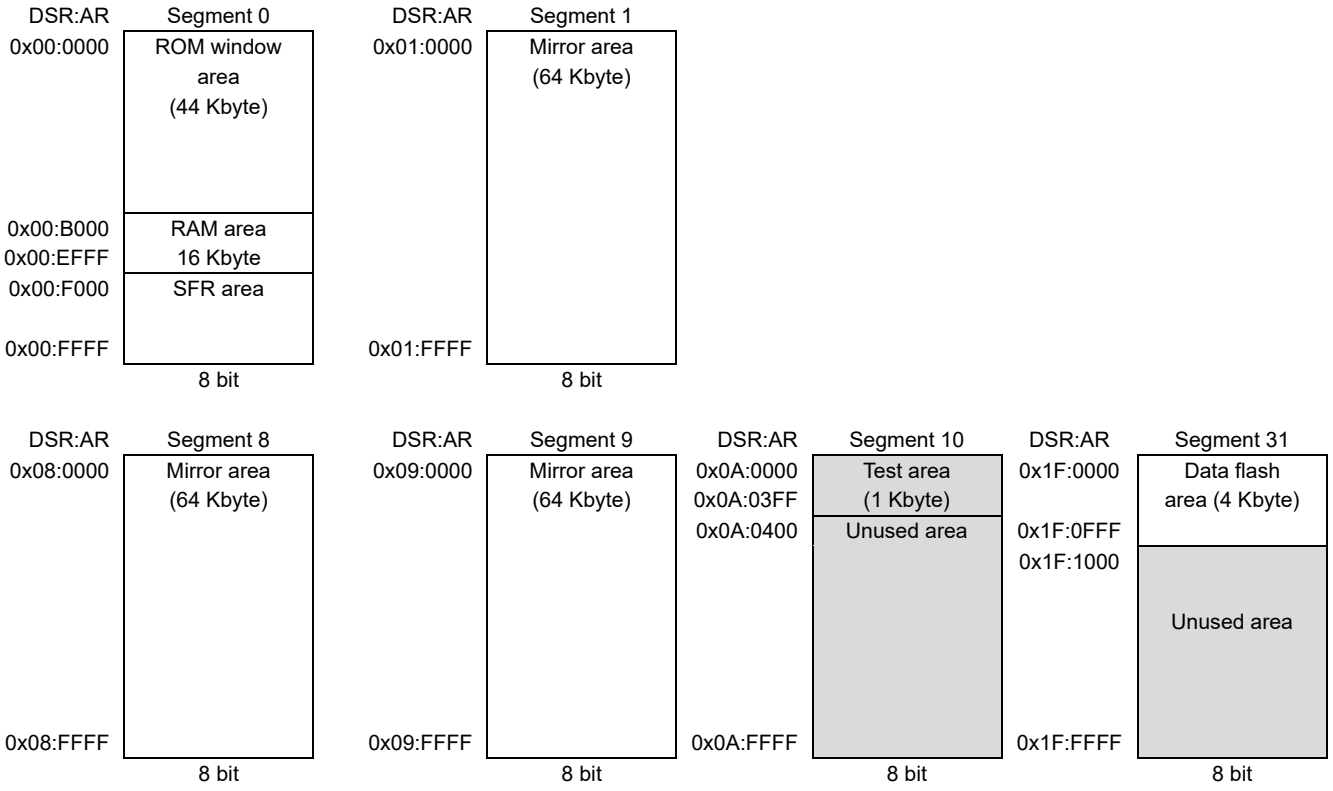
ML62Q1733/1743

Figure 2-41 ML62Q1700 Group Configuration of Data Memory Space 5



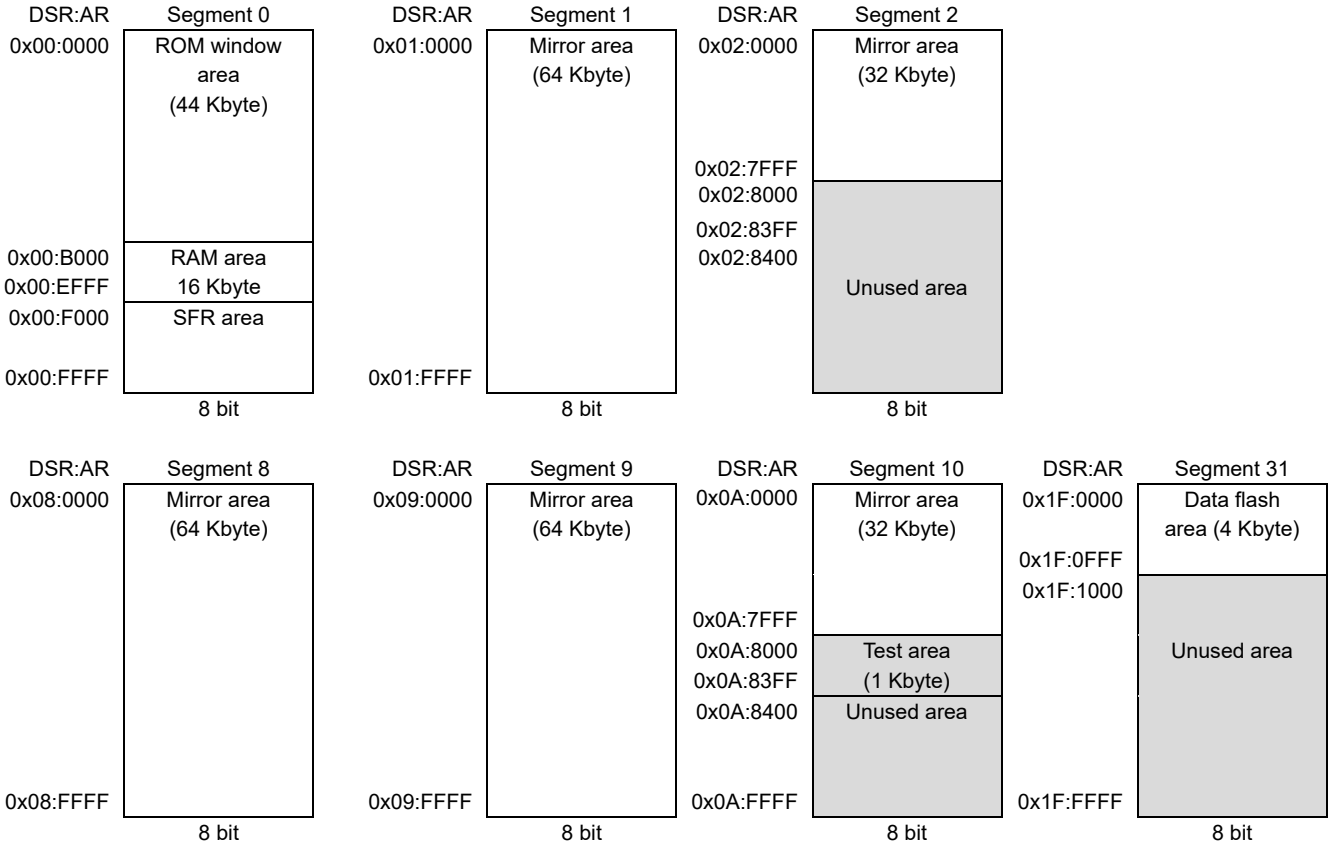
ML62Q1704/1714/1724

Figure 2-42 ML62Q1700 Group Configuration of Data Memory Space 6



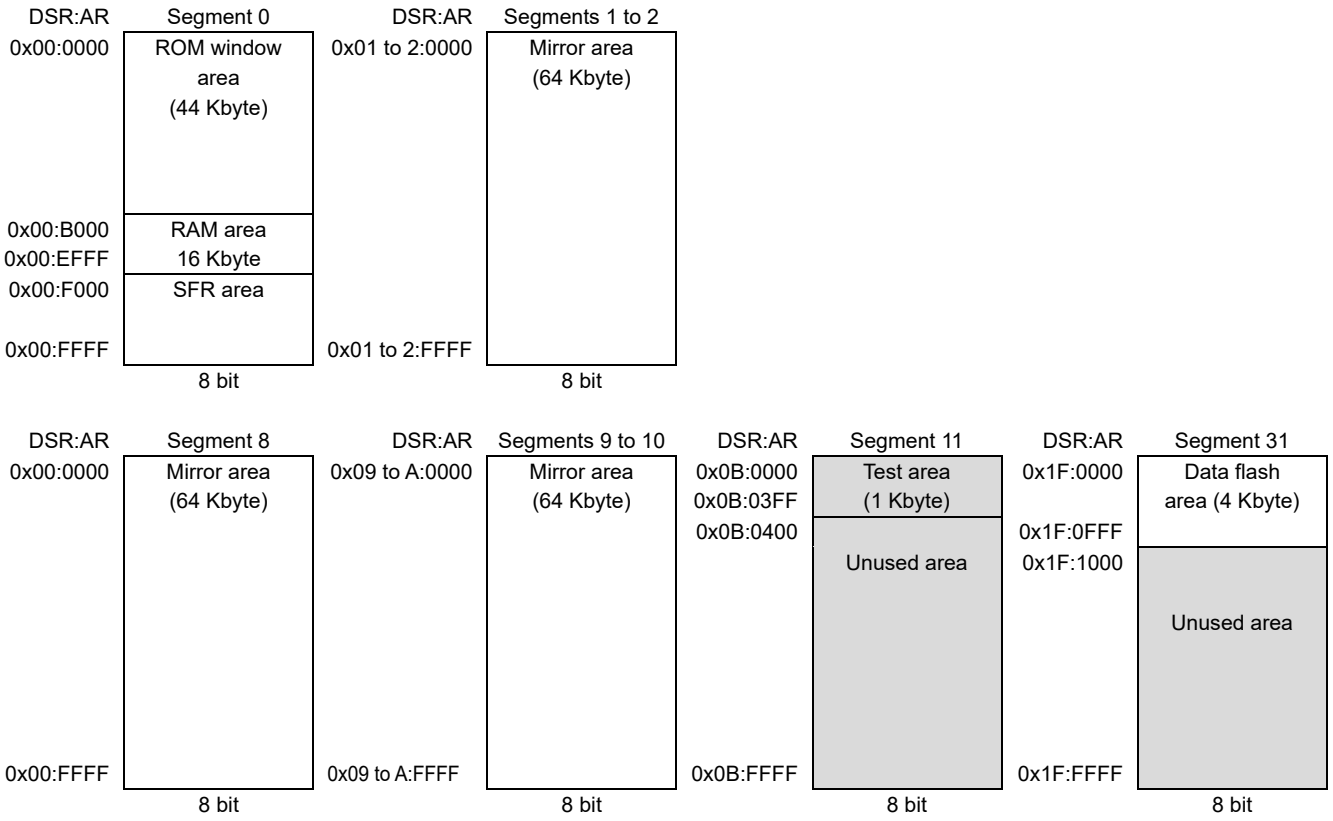
ML62Q1734/1744

Figure 2-43 ML62Q1700 Group Configuration of Data Memory Space 7



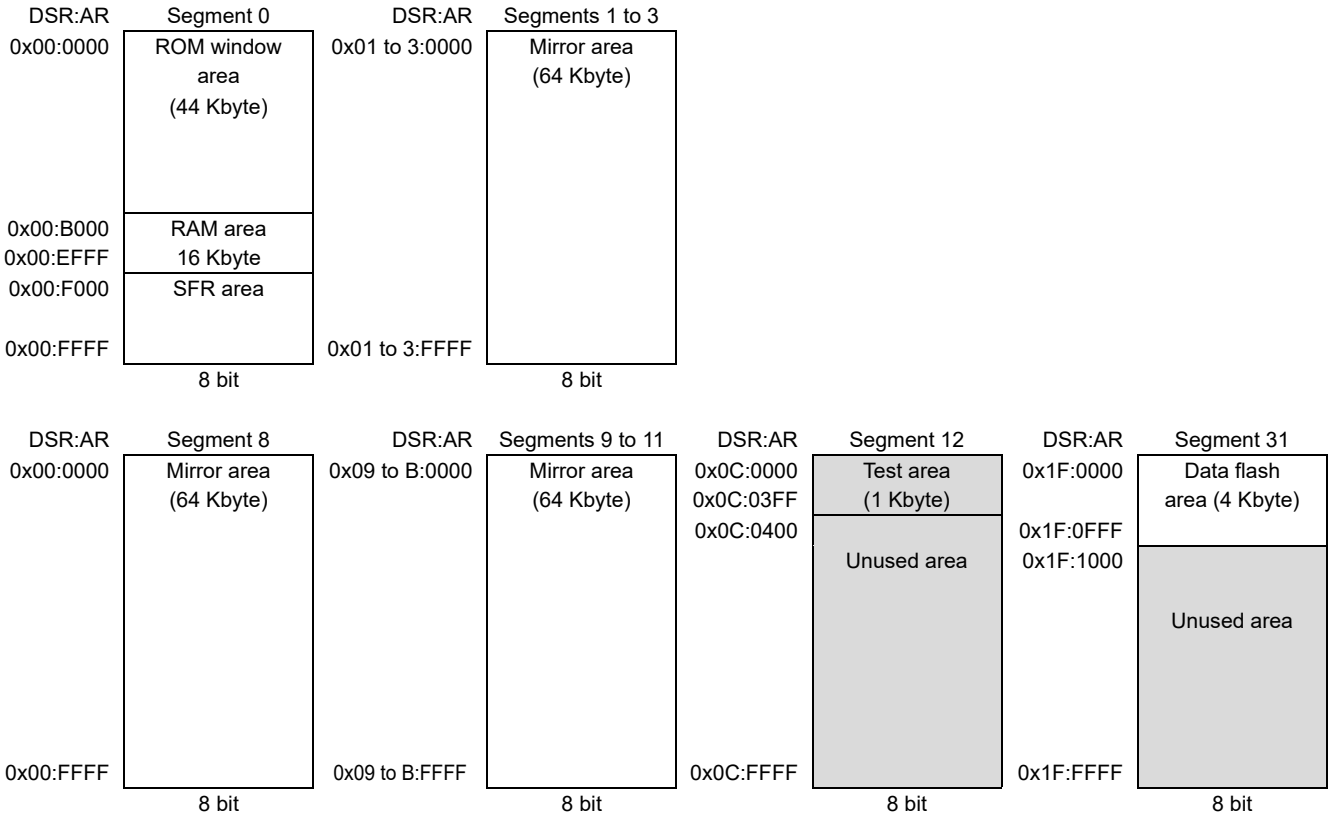
ML62Q1725/1735/1745

Figure 2-44 ML62Q1700 Group Configuration of Data Memory Space 8



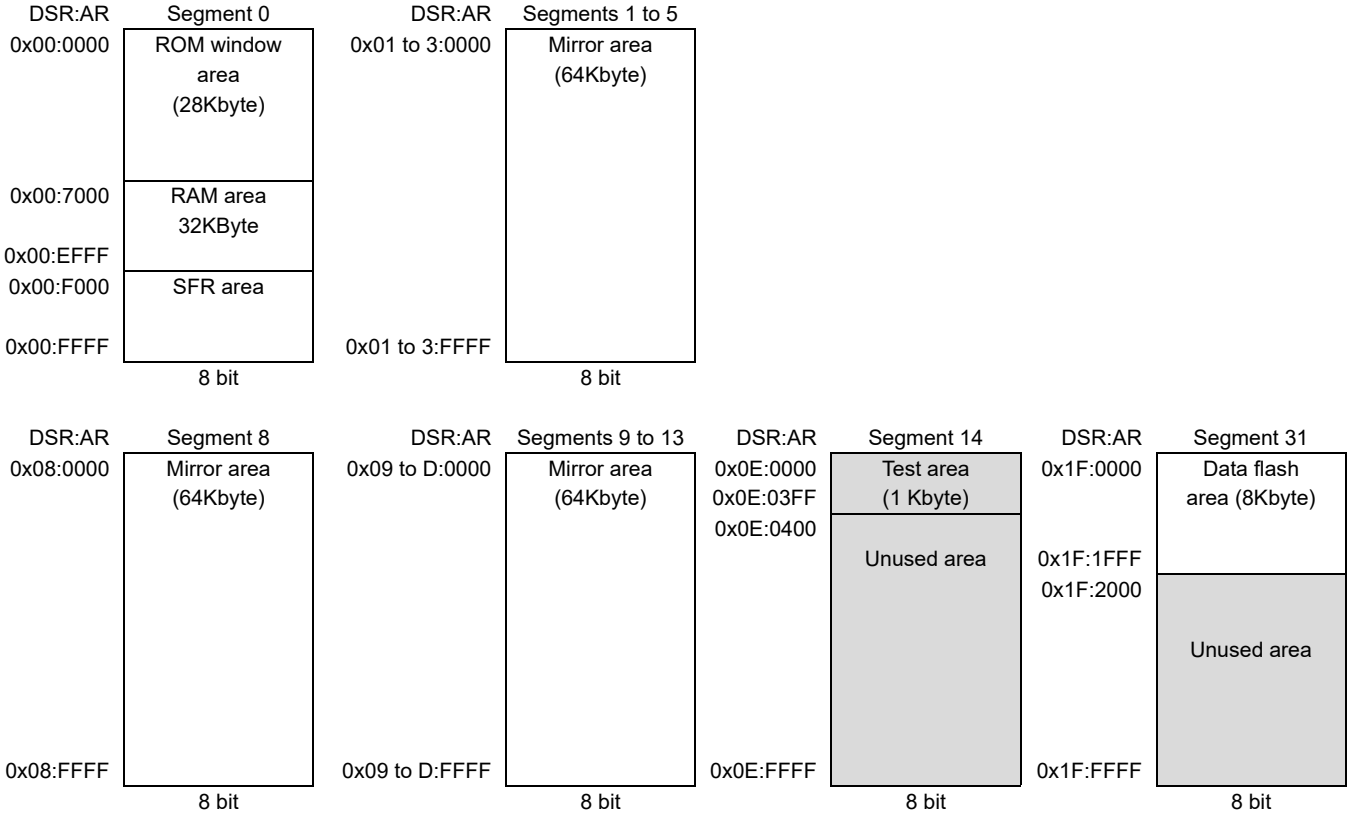
ML62Q1726/1736/1746

Figure 2-45 ML62Q1700 Group Configuration of Data Memory Space 9



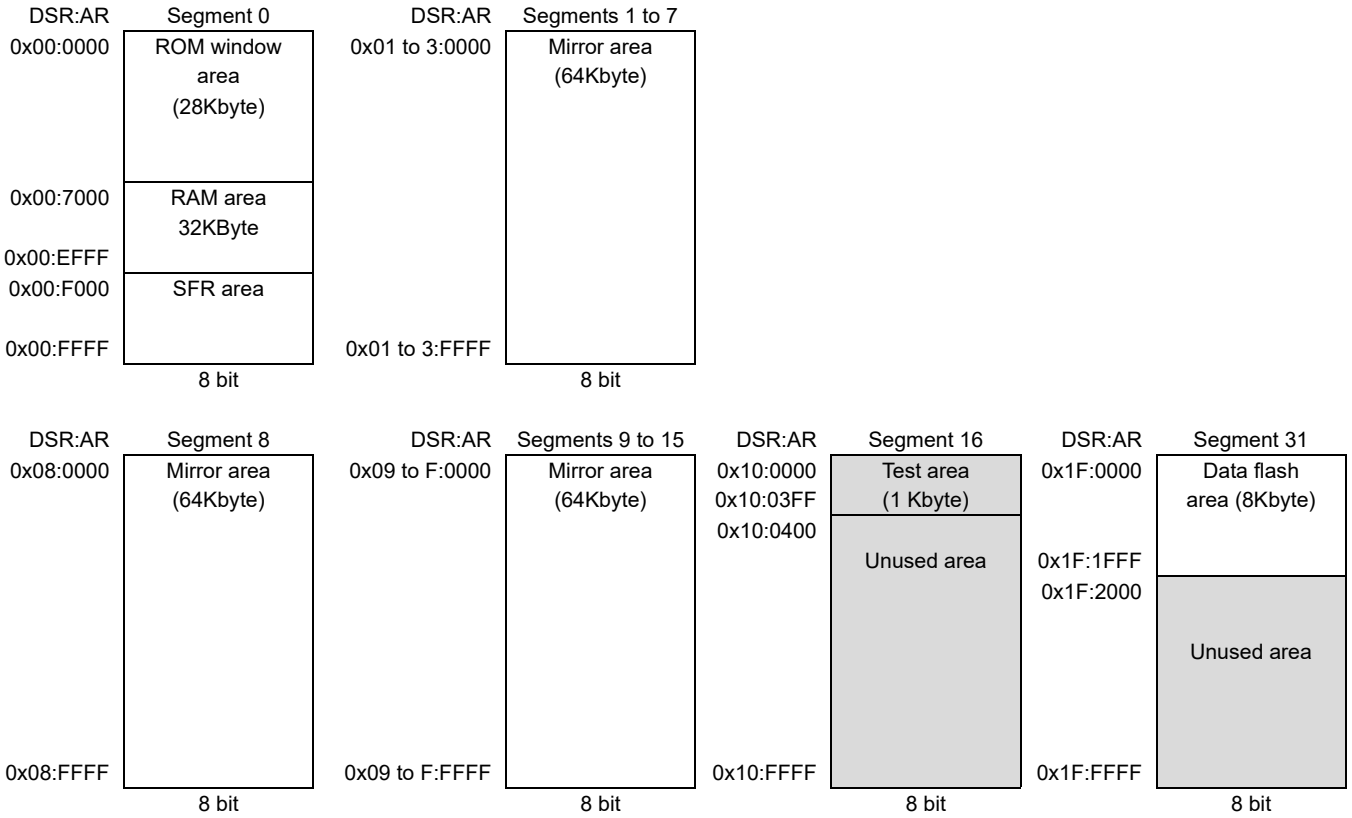
ML62Q1727/1737/1747

Figure 2-46 ML62Q1700 Group Configuration of Data Memory Space 10



ML62Q1728／1738／1748

Figure 2-47 ML62Q1700 Group Configuration of Data Memory Space 11



ML62Q1729／1739／1749

Figure 2-48 ML62Q1700 Group Configuration of Data Memory Space 12

2.7 Description of Registers

2.7.1 List of Registers

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

*1 : The initial value depends on Code Option settings. See “26.2.3 Code Options 2 (CODEOP2)” for details of Code Option settings.

2.7.2 Data Segment Register (DSR)

DSR is a special function register (SFR) used to specify a data segment.
See "nX-U16/100 Core Instruction Manual" for details of DSR.

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

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

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

[Note]
- **Reading the content in unused areas except for data segments 31 returns "0xFF".**
- **Reading the content in unused areas of data segment 31 (data flash) returns "0x00".**

2.7.3 Flash Remap Address Register (REMAPADD)

REMAPADD is a special function register (SFR) used to specify the 4 Kbyte area to be remapped.

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

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

Bit No.	Bit symbol name	Description
7	-	Reserved bit
6 to 4	RES2 to RES0	Bits to set the code segment of the area to remap. For example, when writing "0x1" to RES2-0 and "0xF" to REA15-12, then remapping them, the area of 0xF000-0xFFFF of code segment 1 is remapped with the area of 0x0000-0x0FFF of segment 0. Reset RES0/RES1/RES2 bits to "0" on the products that have 64Kbyte or smaller program memory space (Memory model: SMALL). The RES2 bit is available on the products with 384KB/512KB ROM.
3 to 0	REA15 to REA12	Bits to set the higher 4 bits (bit 15 to 12) of the beginning address of the area to be remapped. For example, when writing "0xF" to REA15-12, then remapping them, the area of 0xF000-0xFFFF is remapped with the area 0x0000-0x0FFF.

*1 : The initial value depends on Code Option settings. See “26.2.3 Code Options 2 (CODEOP2)” for details of Code Option settings.

2.8 Remapping Function

The remapping function replaces the addresses 0x0000 to 0x0FFF (initial boot area) in the program memory space with the specified arbitrary 4 Kbyte area.
Figure 2-49 shows the general scheme of the remapping function.
The program can be started to execute at the area different from the initial boot area using the remapping function, that enables updating(reprograming) the program code area including the initial boot area with the self-programming function.
The remap function and IAP(In-Application Programming) program enable your application to reprogram the firmware.

Two ways are available to start the remap function.

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

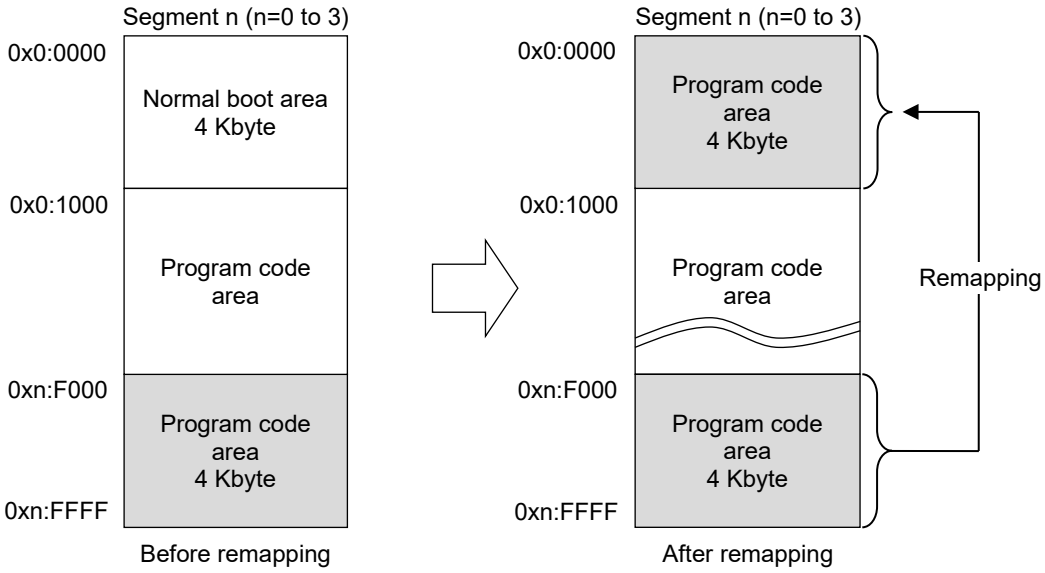


Figure 2-49 Memory Map Before and After Remapping of Program Memory Space

2.8.1 Description of Remapping Function

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

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

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

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

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

Refer to "ML62Q1000 Series Reference Software" for specific details on how to use the remapping function.
Refer to "ML62Q1000 Series IAP Sample Program" for how to re-write the user application program on the flash memory using the remapping function.

2.8.2 Software Remap

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

- Set "0" in advance to the REMAPMD bit of Code Option 0 (see Chapter 26 "Code Option" for details on how to set the Code Option).
- Set the code segment and higher 4 bits of the beginning address of the area to be remapped to the REMAPADD register.
- Set ELEVEL of CPU program status word to "2", then execute the BRK instruction (see "nX-U16/100 Core Instruction Manual" for details of ELEVEL and BRK instruction).
- Only the CPU is initialized and it executes the program from the area specified in the REMAPADD register.

Figure 2-50 below shows an example of the program script of software remapping.

<If the beginning address of the area to be remapped is 0x3:F000>

```
#asm
        mov     r0,     #03fh
        st      r0,     REMAPADD    ; REMAPADD = 0x3F
        mov     psw,    #02h        ; ELEVEL = 2
        nop
        nop
        brk                         ; BRK instruction
#endasm
```

Figure 2-50 Program Script Example of Software Remapping

[Note]

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

2.8.3 Code Option Remap

By setting of CREMAPMD, CRES2-0 and CREA15-CREA12 of the Code Options 2(CODEOP2), the LSI always starts at the system reset on the remap condition.

- If setting both REMAPMD and CREMAPMD to "0", the LSI starts running at the address set in CRES2-0 and CREA15-CREA12.
- After updating the address in the REMAPADD register, the address is not initialized by the CPU reset (BRK instruction) and the remap starts at the updated address. However, the REMAPADD register is initialized by the system reset, the LSI starts running at the address specified by the Code Option.

The following shows the CPU address at releasing reset of each condition.

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

Chapter 3 Reset Function

3. Reset Function

3.1 General Description

ML62Q1000 series has a function to reset the CPU, peripheral circuits and other hardware due to the causes described below.
This chapter describes the system reset mode, reset input pin reset and power-on reset (POR). See reference chapters for other causes of resets. See Table 3-1 for the availability of resets for each cause.

Table 3-1 Reference for Details of Causes of Resets

Cause	Reference
Reset input pin reset (pin reset)	This chapter
Power-On Reset (POR)	This chapter
Watchdog timer (WDT) overflow reset	Chapter 10 Watchdog Timer
Watchdog timer (WDT) invalid clear reset	Chapter 10 Watchdog Timer
Voltage Level Supervisor reset (VLS0 reset)	Chapter 22 Voltage Level Supervisor
RAM parity error reset	Chapter 29 Safety Function
Unused ROM area access reset	Chapter 29 Safety Function
CPU reset by BRK instruction execution (when ELEVEL is 2 or higher)	"nX-U16/100 Core Instruction Manual"
Individual reset to the peripheral circuits(Block reset)	Chapter 4 Power Management
One-time reset to the all peripheral circuits and port controller (SOFTR reset)	Chapter 4 Power Management

3.1.1 Features

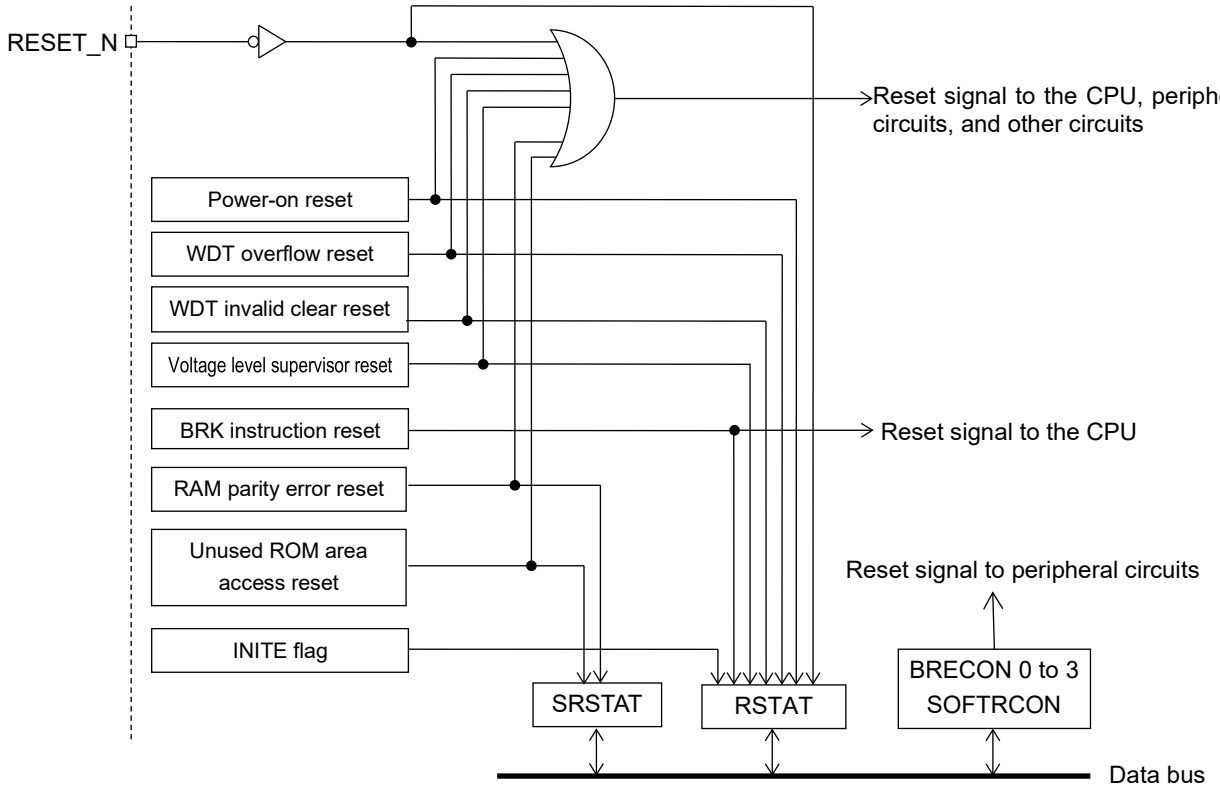
Each reset can uniquely be managed depending on its cause as this function contains following features to identify the cause in an early stage.

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

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

3.1.2 Configuration

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



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

Figure 3-1 Configuration of Reset Generation Circuit

3.1.3 List of Pins

Pin name	I/O	Function
RESET_N	I	Reset input pin

3.2 Description of Registers

3.2.1 List of Registers

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

3.2.2 Reset status register (RSTAT)

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

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

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

Bit No.	Bit symbol name	Description
15 to 9	-	Reserved bits
8	BRKR	A bit to indicate that a CPU reset has occurred through the BRK instruction execution by the CPU. This bit is initialized to "0" when "1" is written. 0: No CPU reset through the BRK instruction occurred 1: CPU reset by BRK instruction occurred
7	INITE	A read-only bit to indicate that an abnormality occurred in starting LSI. If this bit is set to "1", restart the LSI by causing a reset to occur with the reset input pin reset or power-on. 0: LSI started-up normally 1: Abnormality occurred in start-up of LSI
6	RSTR	A bit to indicate that a reset input pin reset has occurred. This bit is initialized to "0" when "1" is written. 0: No reset input pin reset occurred 1: Reset input pin reset occurred
5	-	Reserved bits
4	VLS0R	A bit to indicate that a voltage level supervisor reset has occurred. This bit is initialized to "0" when "1" is written. 0: No voltage level supervisor reset occurred 1: Voltage level supervisor reset occurred
3	WDTWR	A bit to indicate that a WDT invalid clear reset has occurred. This bit is initialized to "0" when "1" is written. 0: No WDT invalid clear reset occurred 1: WDT invalid clear reset occurred
2	WDTR	A bit to indicate that a WDT overflow reset has occurred. This bit is initialized to "0" when "1" is written. 0: No WDT overflow reset occurred 1: WDT overflow reset occurred
1	-	Reserved bits

Bit No.	Bit symbol name	Description
0	POR	A bit to indicate that a power-on reset has occurred. This bit is reset to "0" when "1" is written. 0: No power-on reset occurred 1: Power-on reset occurred

3.2.3 Safety Function Reset Status Register (SRSTAT)

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.

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

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

Bit No.	Bit symbol name	Description
15 to 2	-	Reserved bits
1	RPER	A bit to indicate that a RAM parity error reset has occurred. This bit is initialized to "0" when "1" is written. 0: No RAM parity error reset occurred 1: RAM parity error reset occurred
0	FIAR	A bit to indicate that an unused ROM area access reset has occurred. This bit is initialized to "0" when "1" is written. 0: No unused ROM area access reset occurred 1: Unused ROM area access reset occurred

3.3 Description of Operation

3.3.1 Operation of Reset Function

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

Table 3-2 Availability of Resets for Each Cause

Cause	CPU	RAM	Crystal Oscillation and RTC functions *1	Voltage level supervisor	Other Peripheral circuits	System Circuit *2
Reset input pin reset (pin reset)	●	-	-	●	●	●
Power-on reset (POR)	●	-	●	●	●	●
WDT overflow reset	●	-	-	-	●	●
WDT invalid clear reset	●	-	-	-	●	●
Voltage level supervisor reset	●	-	-	-	●	●
RAM parity error reset	●	-	-	-	●	●
Unused ROM area access reset	●	-	-	-	●	●
BRK instruction reset (CPU reset)	●	-	-	-	-	-
Block reset	-	-	-	-	●*3	-
SOFTR reset	-	-	-	-	●*3	-

●: Reset available -: Reset unavailable

*1: Low-speed crystal oscillation circuit, low-speed external clock input circuit and RTC function. Target SFRs are FLMOD, FBUCON, FBUSTAT, LTBRR, LTBADJ, SRTCMAS registers. Refer to Section 6, 7 and 16 for details.

*2: Power circuit, internal oscillation circuit, start control part, code option control part, etc.

*3: Refer to Section 4.

[Note]

- **The voltage level supervisor function is only initialized at a pin reset or POR.**
- **The BRK instruction reset only initializes the CPU if ELEVEL is 2 or higher. Peripheral circuits and other circuits are not initialized. Use the pin reset or the watchdog timer (WDT) reset to surely initialize the LSI when an abnormality is detected.**
- **In the Block reset and SOFTR reset, only the corresponding peripheral circuits are initialized. The CPU and other circuits are not initialized, and not transferred to the system reset mode.**

3.3.2 System Reset Mode

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

1. The fundamental hardware for the LSI operation, such as the power supply circuit and oscillation circuit, is initialized. In addition, functions chosen by the code option are configured. The INITE bit of the reset status register (RSTAT) is set to "1" if an abnormality occurs during the initialization and configuration.
   See the Chapter 26 "Code Option" for details of the code option.
2. Peripheral circuits, and special function registers (SFRs) with their initial values defined are initialized. See Appendix A "Registers" and chapters for respective functions for the initial values of the SFRs.
3. The CPU is initialized.
   - All the registers in the CPU are initialized.
   - The contents of addresses 0x0000, 0x0001 in segment 0 of the program memory are set to the stack pointer (SP).
   - The contents of addresses 0x0002, 0x0003 in segment 0 of the program memory are set to the program counter (PC).
4. The transition to the program run mode takes place when the reset is released.

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

[Note]

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

3.3.3 Reset Input Pin Reset

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

The reset input pin is also used to debug software using the on-chip debug emulator or to write a program to the flash memory using the ISP function. In these cases, the POR bit is set to "1"and RSTR bit is reset to "0".
See Chapter 28 "On-chip Debug Function" and Chapter 25 "Flash Memory" for how to use the reset input pin.

3.3.4 Power-on Reset

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

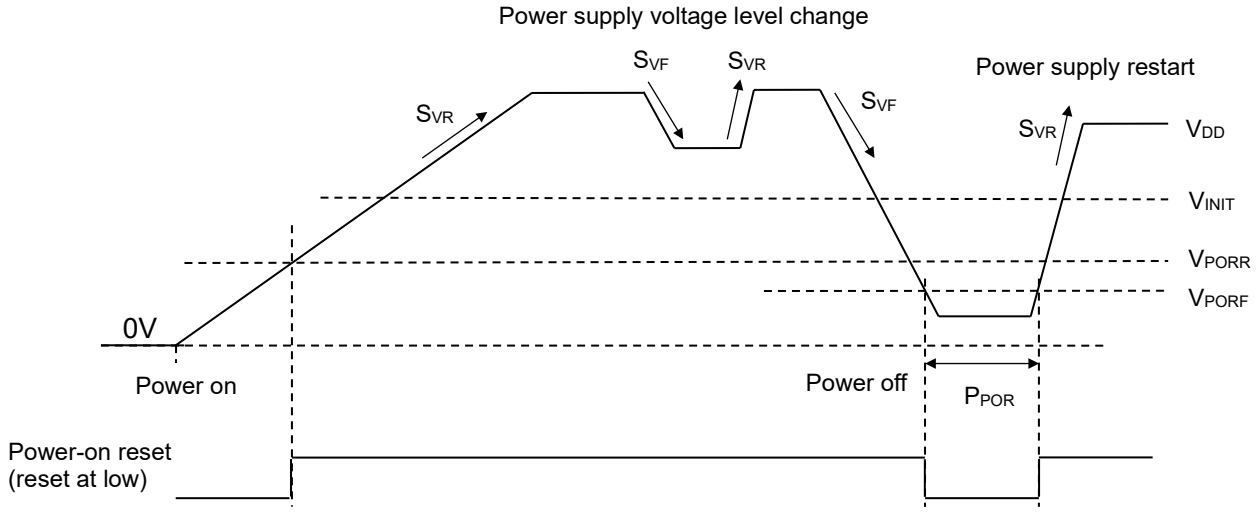


Figure 3-2 Power-on Reset Operation Waveforms

[Note]

- **Rise the V_{DD} to V_{INIT} or higher when powering on.**
- **When using high-speed clock, keep V_{DD} higher than V_{INIT} until the high-speed clock oscillation is enabled.**
- **In case of instantaneous power failure and a pulse shorter than the power-on reset reaction time is asserted to V_{DD}, MCU may not get reset and it may malfunction. In that case, please have preventive measures such as using bypass capacitor to avoid the instantaneous voltage drop or using pin reset to initialize MCU.**

Chapter 4 Power Management

4. Power Management

4.1 General Description

ML62Q1000 series has four power management modes to save the current consumption.

- 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 forcely start the high-speed clock after releasing the mode.
- STOP mode : Stop the CPU, peripheral circuits, low-speed clock and high-speed clock.
- STOP-D mode : Stop the CPU, peripheral circuits, low-speed clock and high-speed clock. V_{DDL} is minimized to lower the current consumption.

4.1.1 Features

- Stop code acceptor 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.
The bit symbols in the figure are assigned to the standby control register (SBYCON).

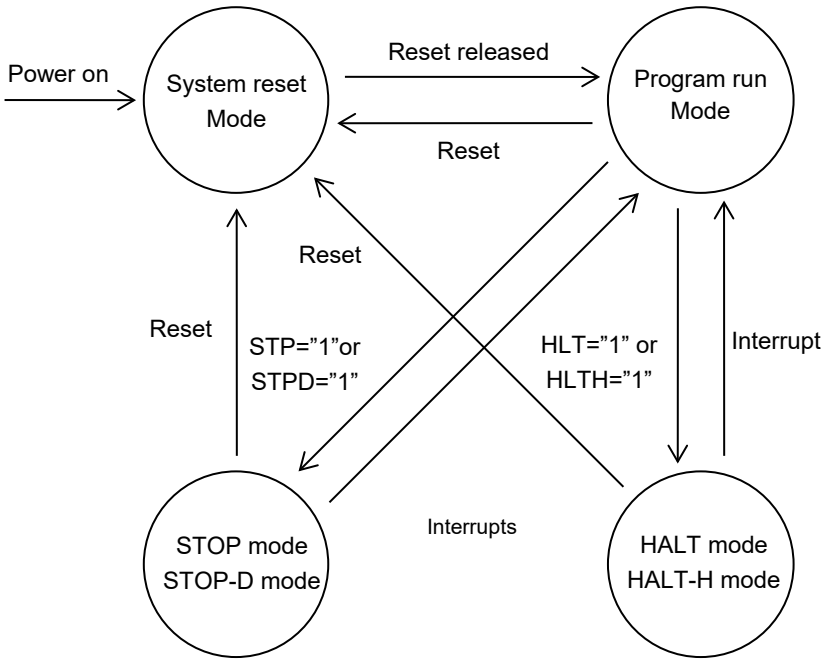


Figure 4-1 Operating State Transition Diagram

4.2 Description of Registers

4.2.1 List of Registers

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

4.2.2 Stop Code Acceptor (STPACP)

STPACP is a write-only special function register (SFR) to be used to change the operating state into the STOP mode and STOP-D mode. The STPACP returns "0x00" for reading.

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

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

How to enter the STOP/STOP-D mode:

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

[Note]

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

4.2.3 Standby Control Register (SBYCON)

SBYCON is a write-only special function register (SFR) to choose a standby mode.
The SBYCON returns "0x0000" for reading.

Address: 0xF01A(SBYCONL/SBYCON), 0xF01B(SBYCONH)
Access: W
Access size: 8/16bit
Initial value: 0x0000

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

Bit No.	Bit symbol name	Description
15 to 4	-	Reserved bit
3	STPD	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 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.
2	HLTH	HLTH is a bit to stop forcibly 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.
1	STP	STP is a bit to change the operating state into the STOP mode. When "1" is written in the STP bit after entering the STOP mode is allowed by using STPACP, the operating state enters the STOP mode. When 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.
0	HLT	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.

[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 condition of MIE flag of the program status word (PSW) is "0", it cancels the standby mode only and the CPU does not go to the interrupt routine. For more details about MIE flag, see "nX-U16/100 Core Instruction Manual".**
- **Insert two NOP instructions in the next to the instruction of that sets HLT, STP, HLTH and STPD bit to "1". The operation without the two NOP instructions is not guaranteed.**
- **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.**
- **When the CPU operation mode is "Wait mode", the PLL reference frequency is 24MHz and the MIE bit is "0", choose 12MHz or slower as the SYSTEMCLK before entering the HALT/HALT-H modes.**
- **When choosing the low-speed crystal oscillation clock or low-speed external clock (LOSCM1-0 of FLMOD register is set to "01" or "11") as the LSCLK, switch the SYSTEMCLK to the low-speed clock before setting STP bit or STPD bit.**
- **Set disable an interrupt of CPU:MIE=0 and choose 16MHz or slower as the SYSTEMCLK, before entering the STOP/STOP-D modes.**

4.2.4 Software Reset Acceptor (SOFTRACP)

SOFTRACP is a write-only special function register (SFR) to enable writing to the SOFTCON register. The SOFTRACP returns "0x00" for reading.

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

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

How to reset collectively the peripheral circuits:

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

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

4.2.5 Software Reset Control Register (SOFTRCON)

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

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

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

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

[Note]

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

4.2.6 Block Clock Control Register 0 (BCKCON0)

BCKCON0 is a special function register (SFR) to control supplying the clock of high-speed and low-speed to the peripheral circuits.
The power consumption can be reduced by stopping the clock supply for unused peripheral circuits.
The bits for the unavailable peripheral circuits are not writeable. They return "0" for reading.
See Table 4-7 "Availability of the SFR bit symbols in BCLCONn register and BRECONn register".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BCKCON0															
Byte	BCKCON0H								BCKCON0L							
Bit	-	-	-	-	-	-	-	-	DCKTM7	DCKTM6	DCKTM5	DCKTM4	DCKTM3	DCKTM2	DCKTM1	DCKTM0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bit
7	DCKTM7	This bit controls the clock supply for the 16-bit Timer 7. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
6	DCKTM6	This bit controls the clock supply for the 16-bit Timer 6. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
5	DCKTM5	This bit controls the clock supply for the 16-bit Timer 5. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
4	DCKTM4	This bit controls the clock supply for the 16-bit Timer 4. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
3	DCKTM3	This bit controls the clock supply for the 16-bit Timer 3. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
2	DCKTM2	This bit controls the clock supply for the 16-bit Timer 2. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
1	DCKTM1	This bit controls the clock supply for the 16-bit Timer 1. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
0	DCKTM0	This bit controls the clock supply for the 16-bit Timer 0. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit

[Note]
- **To restart the operation of the peripheral circuits, reset them at first by the block reset control register (BRECON0) and then cancel the reset after enabling the clock supply by the block clock control register (BCKCON0).**

4.2.7 Block Clock Control Register 1 (BCKCON1)

BCKCON1 is a special function register (SFR) to control supplying the clock of high-speed and low-speed to the peripheral circuits.
The power consumption can be reduced by stopping the clock supply for unused peripheral circuits.
The bits for the unavailable peripheral circuits are not writeable. They return "0" for reading.
See Table 4-7 "Availability of the SFR bit symbols in BCLCONn register and BRECONn register".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BCKCON1															
Byte	BCKCON1H								BCKCON1L							
Bit	-	-	-	DCKI 2CU0	-	-	DCKI 2CM1	DCKI 2CM0	DCKF TM7	DCKF TM6	DCKF TM5	DCKF TM4	DCKF TM3	DCKF TM2	DCKF TM1	DCKF TM0
R/W	R	R	R	R/W	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

Bit No.	Bit symbol name	Description
15 to 13	-	Reserved bits
12	DCKI2CU0	This bit controls the clock supply for the I²C Bus Unit 0. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
11,10	-	Reserved bits
9	DCKI2CM1	This bit controls the clock supply for the I²C Bus Master 1. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
8	DCKI2CM0	This bit controls the clock supply for the I²C Bus Master 0. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
7	DCKFTM7	This bit controls the clock supply for the Functional Timer 7. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
6	DCKFTM6	This bit controls the clock supply for the Functional Timer 6. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
5	DCKFTM5	This bit controls the clock supply for the Functional Timer 5. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
4	DCKFTM4	This bit controls the clock supply for the Functional Timer 4. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
3	DCKFTM3	This bit controls the clock supply for the Functional Timer 3. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit

Bit No.	Bit symbol name	Description
2	DCKFTM2	This bit controls the clock supply for the Functional Timer 3. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
1	DCKFTM1	This bit controls the clock supply for the Functional Timer 2. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
0	DCKFTM0	This bit controls the clock supply for the Functional Timer 0. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit

[Note]

- **To restart the operation of the peripheral circuits, reset them at first by the block reset control register (BRECON1) and then cancel the reset after enabling the clock supply by the block clock control register (BCKCON1).**

4.2.8 Block Clock Control Register 2 (BCKCON2)

BCKCON2 is a special function register (SFR) to control supplying the clock of high-speed and low-speed to the peripheral circuits.
The power consumption can be reduced by stopping the clock supply for unused peripheral circuits.
The bits for the unavailable peripheral circuits are not writeable. They return "0" for reading.
See Table 4-7 "Availability of the SFR bit symbols in BCLCONn register and BRECONn register".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BCKCON2															
Byte	BCKCON2H								BCKCON2L							
Bit	DCKDMA	DCKBUZ	DCKACC	-	DCKCRC	-	-	-	-	-	DCKSU5	DCKSU4	DCKSU3	DCKSU2	DCKSU1	DCKSU0
R/W	R/W	R/W	R/W	R	R/W	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

Bit No.	Bit symbol name	Description
15	DCKDMA	This bit controls the clock supply for the DMA Controller. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
14	DCKBUZ	This bit controls the clock supply for the Buzzer. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
13	DCKACC	This bit controls the clock supply for the Multiplier/Divider. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
12	-	Reserved bit
11	DCKCRC	This bit controls the clock supply for the CRC Calculator. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
10 to 6	-	Reserved bit
5	DCKSU5	This bit controls the clock supply for the Serial Communication Unit 5. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
4	DCKSU4	This bit controls the clock supply for the Serial Communication Unit 4. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
3	DCKSU3	This bit controls the clock supply for the Serial Communication Unit 3. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
2	DCKSU2	This bit controls the clock supply for the Serial Communication Unit 2. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
1	DCKSU1	This bit controls the clock supply for the Serial Communication Unit 1. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit

Bit No.	Bit symbol name	Description
0	DCKSU0	This bit controls the clock supply for the Serial Communication Unit 0. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit

[Note]

- **The DCKACC bit can be set to "1" when the multiplication/division library "muldivu8.lib" is not specified**
- **See manual of the multiplication/division library for specifying "muldivu8.lib".**
- **To restart the operation of the peripheral circuits, reset them at first by the block reset control register (BRECON2) and then cancel the reset after enabling the clock supply by the block clock control register (BCKCON2).**

4.2.9 Block Clock Control Register 3 (BCKCON3)

BCKCON3 is a special function register (SFR) to control supplying the clock of high-speed and low-speed to the peripheral circuits.
The power consumption can be reduced by stopping the clock supply for unused peripheral circuits.
The bits for the unavailable peripheral circuits are not writeable. They return "0" for reading.See Table 4-7 "Availability of the SFR bit symbols in BCLCONn register and BRECONn register".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BCKCON3															
Byte	BCKCON3H								BCKCON3L							
Bit	-	-	-	-	-	-	-	-	-	-	DCK CMP1	DCK CMP0	DCK DAC1	DCKL CD	DCK DAC	DCKS AD
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 5	-	Reserved bit
5	DCKCMP0	This bit controls the clock supply for the Analog Comparator 1. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
4	DCKCMP0	This bit controls the clock supply for the Analog Comparator 0. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
3	DCKDAC1	This bit controls the clock supply for the D/A Converter 1. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
2	DCKLCD	This bit controls the clock supply for the LCD driver. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
1	DCKDAC	This bit controls the clock supply for the D/A Converter 0. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit
0	DCKSAD	This bit controls the clock supply for the Successive approximation type A/D converter. 0: Enable supplying the clock to the peripheral circuit (Initial value) 1: Stop supplying the clock to the peripheral circuit

[Note]
- **To restart the operation of the peripheral circuits, reset them at first by the block reset control register (BRECON3) and then cancel the reset after enabling the clock supply by the block clock control register (BCKCON3).**

4.2.10 Block Reset Control Register 0 (BRECON0)

BRECON0 is a special function register (SFR) to control resetting the peripheral circuits.
The bits for the unavailable peripheral circuits are not writeable. They return "0" for reading.
See Table 4-7 "Availability of the SFR bit symbols in BCLCONn register and BRECONn register".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BRECON0															
Byte	BRECON0H								BRECON0L							
Bit	-	-	-	-	-	-	-	-	RSETM7	RSETM6	RSETM5	RSETM4	RSETM3	RSETM2	RSETM1	RSETM0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bit
7	RSETM7	This bit controls to reset the 16-bit Timer 7. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
6	RSETM6	This bit controls to reset the 16-bit Timer 6. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
5	RSETM5	This bit controls to reset the 16-bit Timer 5. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
4	RSETM4	This bit controls to reset the 16-bit Timer 4. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
3	RSETM3	This bit controls to reset the 16-bit Timer 3. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
2	RSETM2	This bit controls to reset the 16-bit Timer 2. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
1	RSETM1	This bit controls to reset the 16-bit Timer 1. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
0	RSETM0	This bit controls to reset the 16-bit Timer 0. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit

[Note]

- **To restart the operation of the peripheral circuits, reset them at first by the block reset control register (BRECON0) and then cancel the reset after enabling the clock supply by the block clock control register (BCKCON0).**

4.2.11 Block Reset Control Register 1 (BRECON1)

BRECON1 is a special function register (SFR) to control resetting the peripheral circuits.
The bits for the unavailable peripheral circuits are not writeable. They return "0" for reading.See Table 4-7 "Availability of the SFR bit symbols in BCLCONn register and BRECONn register".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BRECON1															
Byte	BRECON1H								BRECON1L							
Bit	-	-	-	RSEI 2CU0	-	-	RSEI 2CM1	RSEI 2CM0	RSEF TM7	RSEF TM6	RSEF TM5	RSEF TM4	RSEF TM3	RSEF TM2	RSEF TM1	RSEF TM0
R/W	R	R	R	R/W	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

Bit No.	Bit symbol name	Description
15 to 13	-	Reserved bit
12	RSEI2CU0	This bit controls to reset the I^{2}C Bus Unit 0. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
11,10	-	Reserved bit
9	RSEI2CM1	This bit controls to reset the I^{2}C Bus Master 1. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
8	RSEI2CM0	This bit controls to reset the I^{2}C Bus Master 0. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
7	RSEFTM7	This bit controls to reset the Functional Timer 7. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
6	RSEFTM6	This bit controls to reset the Functional Timer 6. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
5	RSEFTM5	This bit controls to reset the Functional Timer 5. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
4	RSEFTM4	This bit controls to reset the Functional Timer 4. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
3	RSEFTM3	This bit controls to reset the Functional Timer 3. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
2	RSEFTM2	This bit controls to reset the Functional Timer 2. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit

Bit No.	Bit symbol name	Description
1	RSEFTM1	This bit controls to reset the Functional Timer 1. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
0	RSEFTM0	This bit controls to reset the Functional Timer 0. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit

[Note]

- **To restart the operation of the peripheral circuits, reset them at first by the block reset control register (BRECON1) and then cancel the reset after enabling the clock supply by the block clock control register (BCKCON1).**

4.2.12 Block Reset Control Register 2 (BRECON2)

BRECON2 is a special function register (SFR) to control resetting the peripheral circuits.
The bits for the unavailable peripheral circuits are not writeable. They return "0" for reading.
See Table 4-7 "Availability of the SFR bit symbols in BCLCONn register and BRECONn register".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BRECON2															
Byte	BRECON2H								BRECON2L							
Bit	RSEDMA	RSEBUZ	RSEACC	-	RSECRC	-	-	-	-	-	RSESU5	RSESU4	RSESU3	RSESU2	RSESU1	RSESU0
R/W	R/W	R/W	R/W	R	R/W	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

Bit No.	Bit symbol name	Description
15	RSEDMA	This bit controls to reset the DMA Controller. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
14	RSEBUZ	This bit controls to reset the Buzzer. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
13	RSEACC	This bit controls to reset the Multiplier/Divider. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
12	-	Reserved bit
11	RSECRC	This bit controls to reset the CRC calculator. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
10 to 6	-	Reserved bit
5	RSESU5	This bit controls to reset the Serial Communication Unit 5. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
4	RSESU4	This bit controls to reset the Serial Communication Unit 4. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
3	RSESU3	This bit controls to reset the Serial Communication Unit 3. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
2	RSESU2	This bit controls to reset the Serial Communication Unit 2. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
1	RSESU1	This bit controls to reset the Serial Communication Unit 1. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
0	RSESU0	This bit controls to reset the Serial Communication Unit 0. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit

[Note]

- **The RSEACC bit can be set to "1" when the multiplication/division library "muldivu8.lib" is not specified.**
- **See manual of the multiplication/division library for specifying "muldivu8.lib".**
- **To restart the operation of the peripheral circuits, reset them at first by the block reset control register (BRECON2) and then cancel the reset after enabling the clock supply by the block clock control register (BCKCON2).**

4.2.13 Block Reset Control Register 3 (BRECON3)

BRECON3 is a special function register (SFR) to control resetting the peripheral circuits.
The bits for the unavailable peripheral circuits are not writeable. They return "0" for reading.
See Table 4-7 "Availability of the SFR bit symbols in BCLCONn register and BRECONn register".

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

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

Bit No.	Bit symbol name	Description
15 to 6	-	Reserved bit
5	RSECMP1	This bit controls to reset the Analog Comparator 1. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
4	RSECMP0	This bit controls to reset the Analog Comparator 0. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
3	RSEDAC1	This bit controls to reset the D/A Converter 1. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
2	RSELCD	This bit controls to reset the LCD driver. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
1	RSEDAC	This bit controls to reset the D/A Converter 0. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit
0	RSESAD	This bit controls to reset the Successive approximation type A/D converter. 0: Cancel to reset the peripheral circuit (Initial value) 1: Remain to reset the peripheral circuit

[Note]

- **To restart the operation of the peripheral circuits, reset them at first by the block reset control register (BRECON3) and then cancel the reset after enabling the clock supply by the block clock control register (BCKCON3).**

4.3 Description of Operation

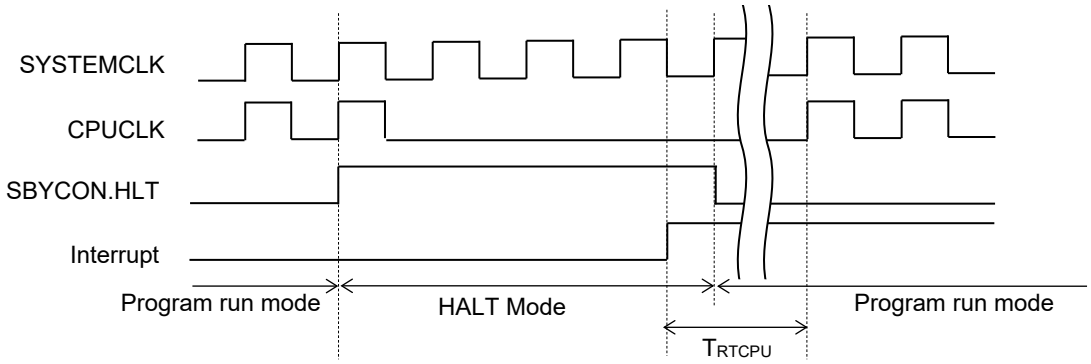
4.3.1 Program Run Mode

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

4.3.2 HALT Mode

The HALT mode is the state where the CPU stops and only the peripheral circuits remain in operation with previous clock condition (LSCLK or HSCLK) for the system clock (SYSTEMCLK) chosen before entering the HALT mode. See "4.3.7 Operation of Each Function in Standby Mode" for the operation of each function in the HALT mode.

When "1" is written to the HLT bit of the SBYCON register, the operating state enters the HALT mode.
When a WDT interrupt or an interrupt enabled in registers IE0 to IE7 occurs, the HALT mode is released at the rising edge of the next SYSTEMCLK, then the mode shifts back to the program run mode with the SYSTEMCLK chosen before entering the HALT mode.
Figure 4-2 shows operation waveforms in the HALT mode.



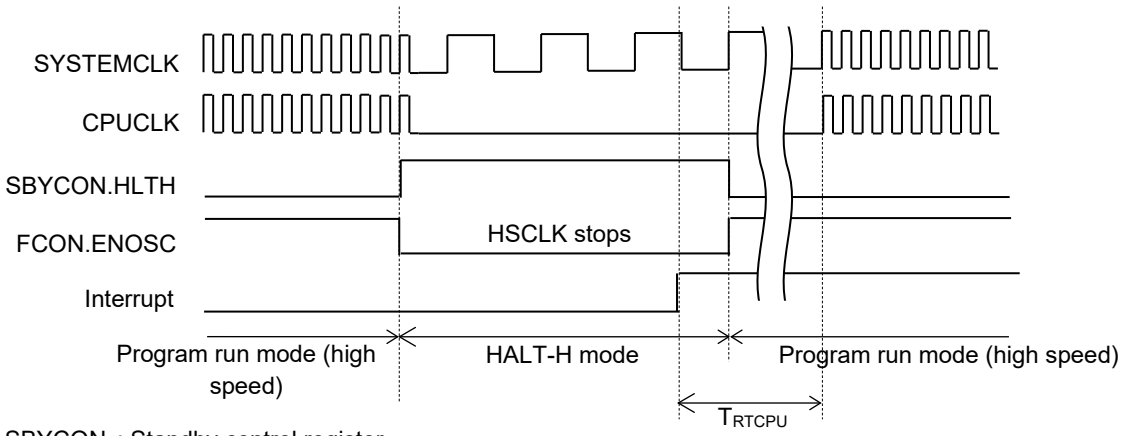
SBYCON : Standby control register
T_{RTCPU} : CPU clock restoring time (see Table 4-5)

Figure 4-2 Operation Waveforms in HALT Mode

4.3.3 HALT-H Mode

In the HALT-H mode, HSCLK is forcibly stopped, the CPU stops, and only peripheral circuits remain in operation. Note that the peripheral circuits in operation with the HSCLK stop operating in the HALT-H mode. See "4.1.7 Operation of Each Function in Standby Mode" for operation of each function in the HALT-H mode.

When "1" is written in the HLTH bit of the SBYCON register, the operating state enters the HALT-H mode.
When a WDT interrupt or an interrupt enabled in registers IE0 to IE7 occurs, the HALT-H mode is released at the rising edge of the next SYSTEMCLK, HSCLK is forcibly enabled, and the mode shifts back to the program run mode with the SYSTEMCLK in the HSCLK state.
Even if the high-speed oscillation is disabled (ENOSC="0") and the low-speed clock (LSCLK) is selected (SELSCLK="0") before entering the HALT-H mode, the high-speed oscillation is forcibly enabled (ENOSC="1") when the HALT-H mode is released and the high-speed clock is chosen for the SELSCLK (SELSCLK="1").
Figure 4-3 shows operation waveforms in the HALT-H mode.



SBYCON : Standby control register
FCON : Frequency control register
T_{RTCPU} : CPU clock restoring time (see Table 4-5)

Figure 4-3 Operation Waveforms in HALT-H Mode

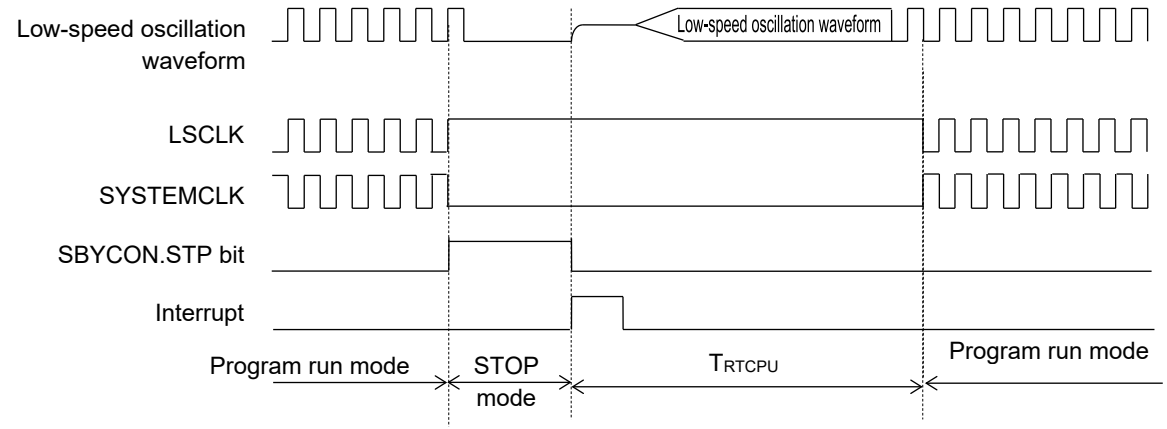
4.3.4 STOP Mode

The STOP mode is the state where all clocks are forcibly stopped, and the CPU and the peripheral circuits which need the clock to operate stop. See "4.3.7 Operation of Each Function in Standby Mode" for operation of each function in the STOP mode.

It needs to disable interrupt (MIE = 0) before entering STOP mode. See “4.3.9 Examples of entering to STOP/STOP-D mode” for details.

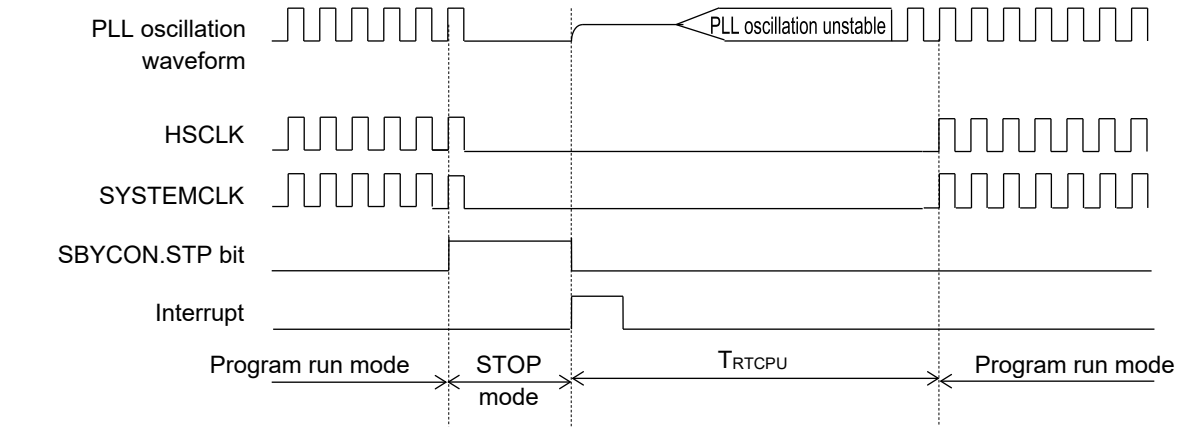
To enter the STOP mode, write "0x5n" and "0xAn" (n = arbitrary) in this order to the STPACP register to enable the transition to the STOP/STOP-D mode, then write "1" to the STP bit of the SBYCON register.
The STOP mode is released by the external interrupts, voltage level supervisor (VLS) or interrupt requests from the analog comparator, I^2C bus unit (slave). The operating state returns to the program run mode with the SYSTEMCLK chosen before entering the STOP mode.

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



SBYCON : Standby control part register
T_{RTCPU} : CPU clock wakeup time (see Table 4-5)

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



SBYCON : Standby control part register
T_{RTCPU} : CPU clock restoring time (see Table 4-5)

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

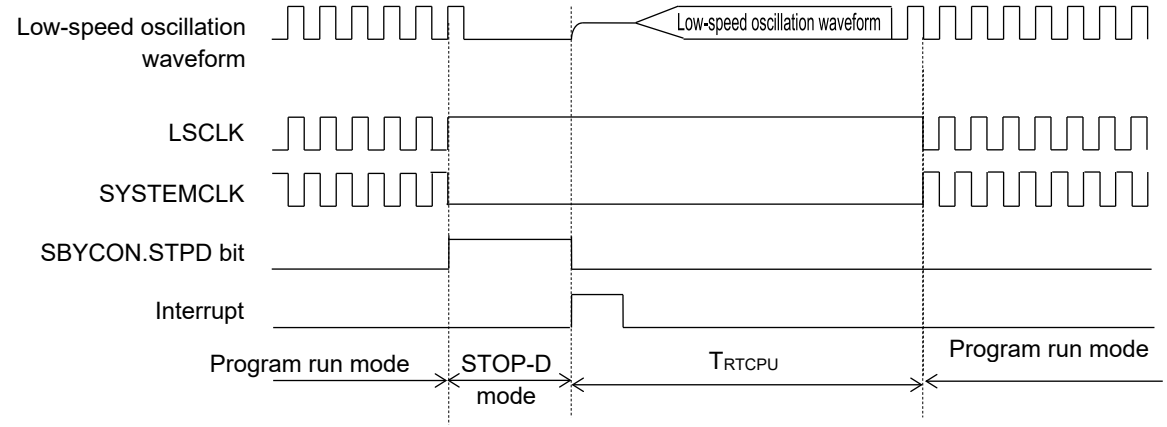
4.3.5 STOP-D Mode

The STOP-D mode has, in addition to the functionality of the STOP mode described in the previous section, an additional control function to decrease the internal logic voltage (V_{DDL}). The current consumption can be further reduced due to lowering of the V_{DDL} voltage. See section "4.3.7 Operation of Each Function in Standby Mode" for the operation of each function in the STOP-D mode.

It needs to disable interrupts (MIE = 0) before entering STOP-D mode. See “4.3.9 Examples of entering to STOP/STOP-D mode” for details

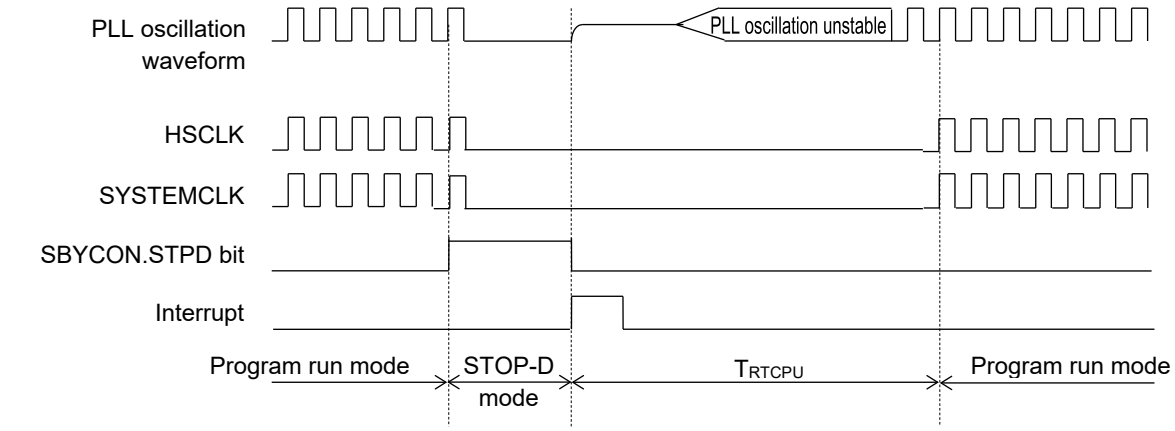
To enter the STOP-D mode, write "0x5n" and "0xAn" (n = arbitrary) in this order to the STPACP register to enable the transition to the STOP/STOP-D mode, then write "1" to the STPD bit of the SBYCON register.
The STOP-D mode is released by the external interrupts, voltage level supervisor (VLS) or interrupt requests from the analog comparator. The operating state returns to the program run mode with the SYSTEMCLK chosen before entering the STOP-D mode.

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



SBYCON : Standby control part register
T_{RTCPU} : CPU clock wakeup time (see Table 4-5)

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



SBYCON : Standby control register
T_{RTCPU} : CPU clock restoring time (see Table 4-5)

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

4.3.6 Note on Return Operation from Standby Mode

The operation of returning the standby mode is caused by the interrupt level (ELEVEL) of the program status word (PSW), master interrupt enable flag (MIE), the contents of the register (IE0 to IE7), non-maskable interrupt, or maskable interrupt. The operation varies depending on the cause. See "nX-U16/100 Core Instruction Manual" for details of PSW and Chapter 5 "Interrupts" for IE and IRQ registers respectively. Tables 4-1 and 4-2 show the return operations from the standby mode for non-maskable interrupt and maskable interrupt respectively.

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

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

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

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

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

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

*1: Do not set to “1” in STOP/STOP-D mode

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

Table 4-3 State of CPU-Processed Interrupt Indicated by ELEVEL

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

[Note]

- **Since up to two instructions are executed during the period between the release of standby mode and a transition to interrupt processing, place two NOP instructions next to the instruction set for the standby mode. When a master interrupt enable (MIE) flag of the program status word (PSW) in the nX-U16/100 CPU core is "1", following the execution of the two NOP instructions, the interrupt transition cycle will be executed and execution of the instruction for interrupt routine begins. If MIE is "0", following the execution of the two NOP instructions, the instruction execution is continued from the one that follows the NOP instruction without transition to the interrupt.**
- **Choose 12MHz or slower as the SYSTEMCLK before entering the HALT/HALT-H mode when the CPU operation mode is “Wait mode”, the PLL reference frequency is 24MHz and the MIE bit is “0”.**
- **Before entering the STOP/STOP-D mode, disable interrupts (MIE = 0) and choose 16MHz or slower as the SYSTEMCLK.**

4.3.7 Operation of Each Function in Standby Mode

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

Table 4-4 State of Each Function in Standby Mode

Function block	HALT	HALT-H*2	STOP	STOP-D
CPU	-	-	-	-
RAM	Retain	Retain	Retain	Retain
Watchdog timer	●	●	-	-
External interrupt	●	●	●*1	●*1
Low-speed time base counter	●	●	-	-
16-bit timer	●	●	-	-
Functional timer	●	●	-	-
Serial communication unit (UART)	●	●	-	-
Serial communication unit (SSIO master)	●	●	-	-
Serial communication unit (SSIO slave)	●	-	-	-
I^2C bus unit	●	-	●	-
I^2C bus master	●	-	-	-
Buzzer	●	●	-	-
Successive approximation type A/D converter	●	●	-	-
D/A converter*3	●	●	●*5	●*5
Analogue comparator	●	●	●*1	●*1
Voltage level monitoring (VLS)	●	●	●*1	●*1
BGO operation*4	●	-	-	-
DMA controller	●	●	-	-
CRC calculator	●*6	●*6	-	-
Multiplier/Divider	●	●	-	-
Crystal oscillation circuit *7	●	●	-	-
Simplified RTC *7	●	●	-	-
LCD driver *8	●	●	-	-

●□: Operable -: Not operable (stop)

*1: If a sampling function is selected, it is forcibly disabled.

*2: In the case of transition to the HALT-H mode, the operation of the peripheral circuit with a high-speed clock will stop. To operate the peripheral circuit even in the HALT-H mode, select the low-speed clock for the operation clock.

*3: The D/A converter in the standby mode retains the state before the transition to the standby mode. Unavailable on 16pin/20pin products of ML62Q1300 group.

*4: The BGO operation means the operation in which erasing or writing the data flash is in progress.

*5: Unavailable to use as a trigger for waking up the STOP/STOP-D mode.

*6: The function is available when the SYSTEMCLK is 16MHz or slower.

*7: Unavailable on ML62Q1300 group.

*8: Available on ML62Q1700 group.

Table 4-5 shows the wake-up time (restoring time) from the standby modes.
See Chapter 6 "Clock Generation Circuit" for details of the FHWUPT register.

Table 4-5 Wake-up Time from Standby Mode

Function	Condition	CPU clock restoring time [T_{RTCPU}]	Low-speed clock restoring time (Low-speed RC oscillation) [T_{RTLS}]	High-speed clock restoring time (PLL oscillation) [T_{RTPLL}]	
				FHWUPT=0x01	FHWUPT=0x00
HALT mode	Low-speed CPU clock High-speed clock OFF No CRC calculation	Approx. 150 µs	Operation continued	Stopped	
	Low-speed CPU clock High-speed clock ON or with CRC calculation	Approx. 60 µs	Operation continued	Operation continued	
	High-speed CPU clock	-	Operation continued	Operation continued	
HALT-H mode	No CRC calculation	T_{RTPLL} + 90 µs	Operation continued	Approx. 60 µs	Approx. 2.5 ms
	With CRC calculation	T_{RTPLL}	Operation continued	Approx. 60 µs	
STOP mode	Low-speed CPU clock	T_{RTLS}	Approx. 320 µs	Approx. 305 µs	
	High-speed CPU clock	T_{RTPLL}	Approx. 320 µs	Approx. 305 µs	
STOP-D mode	Low-speed CPU clock	T_{RTLS}	Approx. 320 µs	Approx. 305 µs	
	High-speed CPU clock	T_{RTPLL}	Approx. 320 µs	Approx. 305 µs	

[Note]

- **If SYSTEMCLK is switched to high-speed after the STOP/STOP-D mode is released and before the high-speed clock wake-up time passes, the CPU must wait to run the program because the clock supply is suspended until the end of the wake-up time.**
- **If peripheral circuits need to work in the HALT-H mode, choose the low-speed clock as the operating clock.**
- **When the FHWUPT register is set to "0x00", the PLL output clock is masked for approx.2.5 ms. HSCLK will be supplied after the elapse of 2.5 ms. If the high-speed clock is selected as SYSTEMCLK, the SYSTEMCLK is stopped for the time period.**
- **When the FHWUPT register is set to "0x01", the frequency of PLL oscillation clock gradually increases from approx. 1 MHz after the elapse of the wake-up time chosen by the FHWUPT register and reaches the target frequency (16 MHz/24 MHz) chosen by the code option before approx. 2 ms elapse. The PLL oscillation clock during this time period can be used for the SYSTEMCLK, however, accuracy of the frequency is not guaranteed.**

4.3.8 Block Control Function

ML62Q1000 series has the block clock control function, which stops clock supply for each peripheral circuit to reduce current consumption, and the block reset control function to reset each peripheral circuit.
When setting each bit of the BCKCONn registers (n=0 to 3) to "1", the clock supply to the corresponding peripheral circuits stops, and the current consumption is reduced.
When setting each bit of the BRECONn registers (n=0 to 3) to "1", the corresponding peripheral circuits are reset and those SFRs are set with initial values.
Table 4-6 shows the list of peripheral circuits controllable with the block control function and control registers.

Table 4-6 List of Peripheral Circuits and Control Registers

Peripheral circuit	Block clock control function		Block reset control function		Software reset function
	SFR word symbol	SFR bit symbol	SFR word symbol	SFR bit symbol	SFR bit symbol
16-bit timer 0	BCKCON0	DCKTM0	BRECON0	RSETM0	SOFTR
16-bit timer 1		DCKTM1		RSETM1	
16-bit timer 2		DCKTM2		RSETM2	
16-bit timer 3		DCKTM3		RSETM3	
16-bit timer 4		DCKTM4		RSETM4	
16-bit timer 5		DCKTM5		RSETM5	
16-bit timer 6		DCKTM6		RSETM6	
16-bit timer 7		DCKTM7		RSETM7	
Functional timer 0	BCKCON1	DCKFTM0	BRECON1	RSEFTM0	
Functional timer 1		DCKFTM1		RSEFTM1	
Functional timer 2		DCKFTM2		RSEFTM2	
Functional timer 3		DCKFTM3		RSEFTM3	
Functional timer 4		DCKFTM4		RSEFTM4	
Functional timer 5		DCKFTM5		RSEFTM5	
Functional timer 6		DCKFTM6		RSEFTM6	
Functional timer 7		DCKFTM7		RSEFTM7	
I^2C bus master 0		DCKI2CM0		RSEI2CM0	
I^2C bus master 1		DCKI2CM1		RSEI2CM1	
I^2C bus unit 0		DCKI2CU0		RSEI2CU0	
Serial communication unit 0	BCKCON2	DCKSU0	BRECON2	RSESU0	
Serial communication unit 1		DCKSU1		RSESU1	
Serial communication unit 2		DCKSU2		RSESU2	
Serial communication unit 3		DCKSU3		RSESU3	
Serial communication unit 4		DCKSU4		RSESU4	
Serial communication unit 5		DCKSU5		RSESU5	
CRC calculator		DCKCRC		RSECRC	
Multiplier/Divider		DCKACC		RSEACC	
Buzzer		DCKBUZ		RSEBUZ	
DMA controller		DCKDMA		RSEDMA	
Successive approximation type A/D converter	BCKCON3	DCKSAD	BRECON3	RSEADC	
D/A converter 0		DCKDAC		RSEDAC	
D/A converter 1		DCKDAC1		RSEDAC1	
LCD driver		DCKLCD		RSELCD	
Analog comparator 0		DCKCMP0		RSECMP0	
Analog comparator 1		DCKCMP1		RSECMP1	
General purpose ports	-	-	-	-	

After the system reset is released, operation of each peripheral circuit becomes enabled.
For unused peripheral circuits, stop the clock supply by setting a bit of the BRECONn register corresponding to each peripheral circuit to "1", then setting the corresponding bit of BCKCONn to "1".
To use the peripheral circuits, release the reset on the peripheral circuit by setting the bit of the BCKCONn register to "0", then setting the bit of the BRECONn register to "0".
Also, setting the bit of the BRECONn register to "1" only causes a reset to occur while retaining clock supply, enabling each peripheral circuit to be initialized.

In the state where clock supply to each peripheral circuit is suspended or in the reset state, writing to SFRs of corresponding peripheral circuits is disabled. The initial values are read for reading the SFRs of peripheral circuits in the reset state.

[Note]

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

Availability of the SFR bit symbols is dependent of the product specification.

Table 4-7 Availability of the SFR bit symbols in BCLCONn register and BRECONn register

Control register / bit				Available / Unavailable								
				ML62Q1300 group				ML62Q1500 / mL62Q1800 / ML62Q1700 group				
Word symbol	Bit symbol	Word symbol	Bit symbol	16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
BCKCON0	DCKTM0	BRECON0	RSETM0	●	●	●	●	●	●	●	●	●
	DCKTM1		RSETM1	●	●	●	●	●	●	●	●	●
	DCKTM2		RSETM2	●	●	●	●	●	●	●	●	●
	DCKTM3		RSETM3	●	●	●	●	●	●	●	●	●
	DCKTM4		RSETM4	-	-	●	●	●	●	●	●	●
	DCKTM5		RSETM5	-	-	●	●	●	●	●	●	●
	DCKTM6		RSETM6	-	-	-	-	-	-	-	●	●
	DCKTM7		RSETM7	-	-	-	-	-	-	-	●	●
BCKCON1	DCKFTM0	BRECON1	RSEFTM0	●	●	●	●	●	●	●	●	●
	DCKFTM1		RSEFTM1	●	●	●	●	●	●	●	●	●
	DCKFTM2		RSEFTM2	●	●	●	●	●	●	●	●	●
	DCKFTM3		RSEFTM3	●	●	●	●	●	●	●	●	●
	DCKFTM4		RSEFTM4	-	-	-	-	●	●	●	●	●
	DCKFTM5		RSEFTM5	-	-	-	-	●	●	●	●	●
	DCKFTM6		RSEFTM6	-	-	-	-	-	-	-	●	●
	DCKFTM7		RSEFTM7	-	-	-	-	-	-	-	●	●
	DCKI2CM0		RSEI2CM0	●	●	●	●	●	●	●	●	●
	DCKI2CM1		RSEI2CM1	-	-	-	-	●	●	●	●	●
	DCKI2CU0		RSEI2CU0	●	●	●	●	●	●	●	●	●
BCKCON2	DCKSU0	BRECON2	RSESU0	●	●	●	●	●	●	●	●	●
	DCKSU1		RSESU1	●	●	●	●	●	●	●	●	●
	DCKSU2		RSESU2	-	-	-	-	-	-	-	●	●
	DCKSU3		RSESU3	-	-	-	-	-	-	-	●	●
	DCKSU4		RSESU4	-	-	-	-	-	-	-	●	●
	DCKSU5		RSESU5	-	-	-	-	-	-	-	●	●
	DCKCRC		RSECRC	●	●	●	●	●	●	●	●	●
	DCKACC		RSEACC	●	●	●	●	●	●	●	●	●
	DCKBUZ		RSEBUZ	●	●	●	●	●	●	●	●	●
	DCKDMA		RSEDMA	●	●	●	●	●	●	●	●	●
BCKCON3	DCKSAD	BRECON3	RSESAD	●	●	●	●	●	●	●	●	●
	DCKDAC		RSEDAC	-	-	●	●	●	●	●	●	●
	DCKLCD		RSELCD	-	-	-	-	●*1	●*1	●*1	●*1	●*1
	DCKDAC1		RSEDAC1	-	-	-	-	-	-	-	●	●
	DCKCMP0		RSECMP0	●	●	●	●	●	●	●	●	●
	DCKCMP1		RSECMP1	-	-	-	-	●	●	●	●	●

●: Available - : Unavailable ●*1: Available on ML62Q1700 group

4.3.9 Examples of entering to STOP/STOP-D mode

When entering to STOP/STOP-D mode, it needs to disable interrupts (MIE = 0). The following two examples are entering to STOP mode, the same applies to entering to the STOP-D mode.

Figure 4-8 shows the example of description that saves the values of MIE first, then disables interrupts and writes back to MIE after returning from the STOP mode. This example uses SB instruction to write to STP bit, so "z" flag of PSW is rewritten to "0". Notice that R0 is broken because of interrupt processing after returning from the STOP mode.

```
#pragma asm
        PUSH  R0              ;save R0
        MOV   R0,   #05ah
        ST    R0,   STPACP    ;5A
        MOV   R0,   #0a5h
        ST    R0,   STPACP    ;A5
        MOV   R0,   PSW       ;save PSW
        DI
        SB    STP             ;STOP mode
        NOP
        NOP
        MOV   PSW,  R0        ;write back to PSW（go to the interrupt routine when MIE=1）
        NOP
        NOP
        POP   R0              ;write back to R0
#pragma endasm
```

Figure 4-8 Example of disabling interrupt with backup of MIE

Figure 4-9 shows the example of description that uses the interrupt disable state during the software interrupt processing.

```
int main （ void ）
{
   :
    __asm （"swi #0"） ;
    __asm （"nop¥n"） ;
    __asm （"nop¥n"） ;
   :
}

#pragma SWI smpl_procSWI0Int 0x0080 1
static void smpl_procSWI0Int （ void ）
{
/* set the CPU mode to 'Stop mode' */
    lp_setStopMode （） ;
}

void lp_setStopMode （ void ）
{
/* set StopCode Accepter */
    write_reg8 （ STPACP, 0x50 ） ;
    write_reg8 （ STPACP, 0xA0 ） ;

/* The CPU mode is changed to the STOP mode. */
    set_bit （ STP ） ;
    __asm （"nop¥n"） ;
    __asm （"nop¥n"） ;
}
```

Figure 4-9 Example of disabling interrupt with software interrupt

[Note]

- **A next instruction of SWI instruction may get executed before interrupt processing as trigger for waking up. Insert two NOP instructions in the next to the SWI instruction.**

4.3.10 Note of entering to the standby mode

The following describes the note of entering to the standby mode.

(1) CPU continues operation without entering to the standby mode in the following cases, when setting each bit of SBYCON register to "1".

- When setting some bits of SBYCON register at the same time.
- When occuring the interrupt request to CPU.
- When accepter is disabled by SBYACP. (Only STP, STPD)
- When A/D conversion of SA-ADC in progress.
- When operating single mode of VLS. (Only STP, STPD)
- When waiting for stability time of supervisor mode of VLS.

(2) Successive Approximation type A/D Converter is stopped by entering to HALT-H mode, because of stopping HSCLK. So, there is a possibility of unintending current flow, depending on the timing of the stop. Please set the Successive Approximation type A/D Converter operated by LSCLK or stop it, when entering to HALT-H mode.

(3) It is unavailable to write and read while writing to or erasing from the Data Flash, because the operation of FLASH is stopped by entering to HALT-H, STOP and STOP-D mode. Don't shift to HALT-H, STOP or STOP-D mode . Please read the FLASHSTA register to check the status of writing to or erasing from the Data Flash.

Chapter 5 Interrupts

5. Interrupt

5.1 General Description

ML62Q1000 series has the non-maskable interrupt, maskable interrupts and the software interrupt (SWI).
For details of each interrupt, see the corresponding Chapters.
See Chapter 29 "Safety Function" for the MCU status interrupt.
See "Table 1-4 Main Function List" for ML62Q1300 group, "Table 1-5 Main Function List" for ML62Q1500/ML62Q1800 group and "Table 1-6 Main Function List" for ML62Q1700 group in the Chapter 1 to confirm the presence/absence of function in each product.

5.1.1 Features

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

5.2 Description of Registers

5.2.1 List of Registers

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF020	Interrupt enable register 01	IE0	IE01	R/W	8/16	0x00
0xF021		IE1		R/W	8	0x00
0xF022	Interrupt enable register 23	IE2	IE23	R/W	8/16	0x00
0xF023		IE3		R/W	8	0x00
0xF024	Interrupt enable register 45	IE4	IE45	R/W	8/16	0x00
0xF025		IE5		R/W	8	0x00
0xF026	Interrupt enable register 67	IE6	IE67	R/W	8/16	0x00
0xF027		IE7		R/W	8	0x00
0xF028	Interrupt request register 01	IRQ0	IRQ01	R/W	8/16	0x00
0xF029		IRQ1		R/W	8	0x00
0xF02A	Interrupt request register 23	IRQ2	IRQ23	R/W	8/16	0x00
0xF02B		IRQ3		R/W	8	0x00
0xF02C	Interrupt request register 45	IRQ4	IRQ45	R/W	8/16	0x00
0xF02D		IRQ5		R/W	8	0x00
0xF02E	Interrupt request register 67	IRQ6	IRQ67	R/W	8/16	0x00
0xF02F		IRQ7		R/W	8	0x00
0xF030	Interrupt level control enable register	ILEN	-	R/W	8	0x00
0xF031	Reserved	-	-	-	-	-
0xF032	Current interrupt level management register	CIL	-	R/W	8	0x00
0xF033	Reserved	-	-	-	-	-
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

The interrupt sources are dependent on the product specification.
Table 5-1 shows presence/absence of interrupt source in each product.

Table 5-1 List of Interrupt Source (1/3)

Register assignment			Interrupt source	Interrupt source symbol	ML62Q1300 Group				ML62Q1500/1800/1700 Group				
IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)			16 pin	20 pin	24 pin	32 pin	48 pin	52 pin	64 pin	80 pin	100 pin
IRQ0[0]	-	-	WDT Interrupt	WDTINT	●	●	●	●	●	●	●	●	●
-	-	-	-	-	-	-	-	-	-	-	-	-	-
IRQ0[6]	IE0[6]	ILC0[13:12]	VLS0 Interrupt	VLS0INT	●	●	●	●	●	●	●	●	●
IRQ0[7]	IE0[7]	ILC0[15:14]	-	-	-	-	-	-	-	-	-	-	-
IRQ1[0]	IE1[0]	ILC1[1:0]	External Interrupt 0	EXI0INT	●	●	●	●	●	●	●	●	●
IRQ1[1]	IE1[1]	ILC1[3:2]	External Interrupt 1	EXI1INT	●	●	●	●	●	●	●	●	●
IRQ1[2]	IE1[2]	ILC1[5:4]	External Interrupt 2	EXI2INT	●	●	●	●	●	●	●	●	●
IRQ1[3]	IE1[3]	ILC1[7:6]	External Interrupt 3	EXI3INT	●	●	●	●	●	●	●	●	●
IRQ1[4]	IE1[4]	ILC1[9:8]	External Interrupt 4	EXI4INT	●	●	●	●	●	●	●	●	●
IRQ1[5]	IE1[5]	ILC1[11:10]	External Interrupt 5	EXI5INT	●	●	●	●	●	●	●	●	●
IRQ1[6]	IE1[6]	ILC1[13:12]	External Interrupt 6	EXI6INT	●	●	●	●	●	●	●	●	●
IRQ1[7]	IE1[7]	ILC1[15:14]	External Interrupt 7	EXI7INT	●	●	●	●	●	●	●	●	●
IRQ2[0]	IE2[0]	ILC2[1:0]	Clock Backup Interrupt	CBUINT	-	-	-	-	●	●	●	●	●
IRQ2[1]	IE2[1]	ILC2[3:2]	DMA Controller Interrupt (DMA Interrupt)	DMACINT	●	●	●	●	●	●	●	●	●
IRQ2[2]	IE2[2]	ILC2[5:4]	MCU Status Interrupt	MCSINT	●	●	●	●	●	●	●	●	●
IRQ2[3]	IE2[3]	ILC2[7:6]	Serial Communication Unit 00 Interrupt	SIU00INT	●	●	●	●	●	●	●	●	●
IRQ2[4]	IE2[4]	ILC2[9:8]	Serial Communication Unit 01 Interrupt	SIU01INT	●	●	●	●	●	●	●	●	●
IRQ2[5]	IE2[5]	ILC2[11:10]	-	-	-	-	-	-	-	-	-	-	-
IRQ2[6]	IE2[6]	ILC2[13:12]	Successive Approximation type A-D Converter Interrupt (SA-ADC Interrupt)	SADINT	●	●	●	●	●	●	●	●	●
IRQ2[7]	IE2[7]	ILC2[15:14]	-	-	-	-	-	-	-	-	-	-	-
IRQ3[0]	IE3[0]	ILC3[1:0]	Expanded External Interrupt	EXTXINT	-	-	-	-	●	●	●	●	●
IRQ3[1]	IE3[1]	ILC3[3:2]	-	-	-	-	-	-	-	-	-	-	-
IRQ3[2]	IE3[2]	ILC3[5:4]	I^2C Bus Master 0 Interrupt	I2CM0INT	●	●	●	●	●	●	●	●	●
IRQ3[3]	IE3[3]	ILC3[7:6]	I^2C Bus Master 1 Interrupt	I2CM1INT	-	-	-	-	●	●	●	●	●
IRQ3[4]	IE3[4]	ILC3[9:8]	Functional Timer 0 Interrupt	FTM0INT	●	●	●	●	●	●	●	●	●
IRQ3[5]	IE3[5]	ILC3[11:10]	Functional Timer 1 Interrupt	FTM1INT	●	●	●	●	●	●	●	●	●
IRQ3[6]	IE3[6]	ILC3[13:12]	16-bit Timer 0 Interrupt	TM0INT	●	●	●	●	●	●	●	●	●
IRQ3[7]	IE3[7]	ILC3[15:14]	16-bit Timer 1 Interrupt	TM1INT	●	●	●	●	●	●	●	●	●

●: Available -: Unavailable

Table 5-1 List of Interrupt Source (2/3)

Register assignment			Interrupt source	Interrupt source symbol	ML62Q1300 Group				ML62Q1500/1800/1700 Group				
IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)			16 pin	20 pin	24 pin	32 pin	48 pin	52 pin	64 pin	80 pin	100 pin
IRQ4[0]	IE4[0]	ILC4[1:0]	I^2C Bus Unit 0 Interrupt	I2CU0INT	●	●	●	●	●	●	●	●	●
IRQ4[1]	IE4[1]	ILC4[3:2]	Serial Communication Unit 10 Interrupt	SIU10INT	●	●	●	●	●	●	●	●	●
IRQ4[2]	IE4[2]	ILC4[5:4]	Serial Communication Unit 11 Interrupt	SIU11INT	●	●	●	●	●	●	●	●	●
IRQ4[3]	IE4[3]	ILC4[7:6]	-	-	-	-	-	-	-	-	-	-	-
IRQ4[4]	IE4[4]	ILC4[9:8]	Functional Timer 2 Interrupt	FTM2INT	●	●	●	●	●	●	●	●	●
IRQ4[5]	IE4[5]	ILC4[11:10]	Functional Timer 3 Interrupt	FTM3INT	●	●	●	●	●	●	●	●	●
IRQ4[6]	IE4[6]	ILC4[13:12]	16-bit Timer 2 Interrupt	TM2INT	●	●	●	●	●	●	●	●	●
IRQ4[7]	IE4[7]	ILC4[15:14]	16-bit Timer 3 Interrupt	TM3INT	●	●	●	●	●	●	●	●	●
IRQ5[0]	IE5[0]	ILC5[1:0]	Serial Communication Unit 20 Interrupt	SIU20INT	-	-	-	-	-	-	-	●	●
IRQ5[1]	IE5[1]	ILC5[3:2]	Serial Communication Unit 21 Interrupt	SIU21INT	-	-	-	-	-	-	-	●	●
IRQ5[2]	IE5[2]	ILC5[5:4]	Analog Comparator 0 Interrupt	CMP0INT	●	●	●	●	●	●	●	●	●
IRQ5[3]	IE5[3]	ILC5[7:6]	Analog Comparator 1 Interrupt	CMP1INT	-	-	-	-	●	●	●	●	●
IRQ5[4]	IE5[4]	ILC5[9:8]	Functional Timer 4 Interrupt	FTM4INT	-	-	-	-	●	●	●	●	●
IRQ5[5]	IE5[5]	ILC5[11:10]	Functional Timer 5 Interrupt	FTM5INT	-	-	-	-	●	●	●	●	●
IRQ5[6]	IE5[6]	ILC5[13:12]	16-bit Timer 4 Interrupt	TM4INT	-	-	●	●	●	●	●	●	●
IRQ5[7]	IE5[7]	ILC5[15:14]	16-bit Timer 5 Interrupt	TM5INT	-	-	●	●	●	●	●	●	●
IRQ6[0]	IE6[0]	ILC6[1:0]	Serial Communication Unit 30 Interrupt	SIU30INT	-	-	-	-	-	-	-	●	●
IRQ6[1]	IE6[1]	ILC6[3:2]	Serial Communication Unit 31 Interrupt	SIU31INT	-	-	-	-	-	-	-	●	●
IRQ6[2]	IE6[2]	ILC6[5:4]	Serial Communication Unit 40 Interrupt	SIU40INT	-	-	-	-	-	-	-	●	●
IRQ6[3]	IE6[3]	ILC6[7:6]	Serial Communication Unit 41 Interrupt	SIU41INT	-	-	-	-	-	-	-	●	●

●: Available -: Unavailable

Table 5-1 List of Interrupt Source (3/3)

Register assignment			Interrupt source	Interrupt source symbol	ML62Q1300 Group				ML62Q1500/1800/1700 Group				
IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)			16 pin	20 pin	24 pin	32 pin	48 pin	52 pin	64 pin	80 pin	100 pin
IRQ6[4]	IE6[4]	ILC6[9:8]	Functional Timer 6 Interrupt	FTM6INT	-	-	-	-	-	-	-	●	●
IRQ6[5]	IE6[5]	ILC6[11:10]	Functional Timer 7 Interrupt	FTM7INT	-	-	-	-	-	-	-	●	●
IRQ6[6]	IE6[6]	ILC6[13:12]	16-bit Timer 6 Interrupt	TM6INT	-	-	-	-	-	-	-	●	●
IRQ6[7]	IE6[7]	ILC6[15:14]	16-bit Timer 7 Interrupt	TM7INT	-	-	-	-	-	-	-	●	●
IRQ7[0]	IE7[0]	ILC7[1:0]	Serial Communication Unit 50 Interrupt	SIU50INT	-	-	-	-	-	-	-	●	●
IRQ7[1]	IE7[1]	ILC7[3:2]	Serial Communication Unit 51 Interrupt	SIU51INT	-	-	-	-	-	-	-	●	●
IRQ7[2]	IE7[2]	ILC7[5:4]	Low-speed Time Base Counter 0 Interrupt	LTB0INT	●	●	●	●	●	●	●	●	●
IRQ7[3]	IE7[3]	ILC7[7:6]	-	-	-	-	-	-	-	-	-	-	-
IRQ7[4]	IE7[4]	ILC7[9:8]	Low-speed Time Base Counter 1 Interrupt	LTB1INT	●	●	●	●	●	●	●	●	●
IRQ7[5]	IE7[5]	ILC7[11:10]	Low-speed Time Base Counter 2 Interrupt	LTB2INT	●	●	●	●	●	●	●	●	●
IRQ7[6]	IE7[6]	ILC7[13:12]	Simplified RTC Interrupt	RTCINT	-	-	-	-	●	●	●	●	●
IRQ7[7]	IE7[7]	ILC7[15:14]	-	-	-	-	-	-	-	-	-	-	-

●: Available -: Unavailable

5.2.2 Interrupt Enable Register 01 (IE01)

IE01 is a special function register (SFR) to enable or disable the interrupt for each interrupt request.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
After the interrupt is accepted, the master interrupt enable flag (MIE) of the CPU is reset to "0", however, the each applicable flag of IE01 is not reset and remains "1".

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

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

Bit No.	Bit symbol name	Description
15	EPI7	This bit controls to enable or disable the external interrupt 7 (EXI7INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
14	EPI6	This bit controls to enable or disable the external interrupt 6 (EXI6INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
13	EPI5	This bit controls to enable or disable the external interrupt 5 (EXI5INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
12	EPI4	This bit controls to enable or disable the external interrupt 4 (EXI4INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
11	EPI3	This bit controls to enable or disable the external interrupt 3 (EXI3INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
10	EPI2	This bit controls to enable or disable the external interrupt 2 (EXI2INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
9	EPI1	This bit controls to enable or disable the external interrupt 1 (EXI1INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
8	EPI0	This bit controls to enable or disable the external interrupt 0 (EXI0INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
7	-	Reserved bit
6	EVLS0	This bit controls to enable or disable the VLS0 interrupt (VLS0INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
5 to 0	-	Reserved bits

5.2.3 Interrupt Enable Register 23 (IE23)

IE23 is a special function register (SFR) to enable or disable the interrupt for each interrupt request.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
After the interrupt is accepted, the master interrupt enable flag (MIE) of the CPU is reset to "0", however, the each applicable flag of IE23 is not reset and remains "1".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IE23															
Byte	IE3								IE2							
Bit	ETM1	ETM0	EFTM1	EFTM0	EI2CM1	EI2CM0	-	EEXTX	-	ESAD	-	ESIU01	ESIU00	EMCS	EDMA	ECBU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	ETM1	This bit controls to enable or disable the 16-bit Timer 1 interrupt (TM1INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
14	ETM0	This bit controls to enable or disable the 16-bit Timer 0 interrupt (TM0INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
13	EFTM1	This bit controls to enable or disable the Functional Timer 1 interrupt (FTM1INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
12	EFTM0	This bit controls to enable or disable the Functional Timer 0 interrupt (FTM0INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
11	EI2CM1	This bit controls to enable or disable the I^2C Bus Master 1 interrupt (I2CM1INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
10	EI2CM0	This bit controls to enable or disable the I^2C Bus Master 0 interrupt (I2CM0INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
9	-	Reserved bit
8	EEXTX	This bit controls to enable or disable the expanded external interrupt (EXTXINT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
7	-	Reserved bit
6	ESAD	This bit controls to enable or disable the SA-ADC interrupt (SADINT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
5	-	Reserved bit

Bit No.	Bit symbol name	Description
4	ESIU01	This bit controls to enable or disable the Serial Communication unit 01 interrupt (SIU01INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
3	ESIU00	This bit controls to enable or disable the Serial Communication unit 00 interrupt (SIU00INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
2	EMCS	This bit controls to enable or disable the MCU Status interrupt (MCSINT) *1. 0: Disable the interrupt (Initial value) 1: Enable the interrupt
1	EDMA	This bit controls to enable or disable the DMA interrupt (DMACINT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
0	ECBU	This bit controls to enable or disable the Clock Backup interrupt (CBUINT) *2. 0: Disable the interrupt (Initial value) 1: Enable the interrupt

*1: See Chapter 29 "Safety Function" for more details.
*2: See Chapter 6 "Clock Generation Circuit" for more details.

5.2.4 Interrupt Enable Register 45 (IE45)

IE45 is a special function register (SFR) to enable or disable the interrupt for each interrupt request.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
After the interrupt is accepted, the master interrupt enable flag (MIE) of the CPU is reset to "0", however, the each applicable flag of IE45 is not reset and remains "1".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IE45															
Byte	IE5								IE4							
Bit	ETM5	ETM4	EFTM5	EFTM4	ECMP1	ECMP0	ESIU21	ESIU20	ETM3	ETM2	EFTM3	EFTM2	-	ESIU11	ESIU10	EI2CU0
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/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	ETM5	This bit controls to enable or disable the 16-bit Timer 5 interrupt (TM5INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
14	ETM4	This bit controls to enable or disable the 16-bit Timer 4 interrupt (TM4INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
13	EFTM5	This bit controls to enable or disable the Functional Timer 5 interrupt (FTM5INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
12	EFTM4	This bit controls to enable or disable the Functional Timer 4 interrupt (FTM4INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
11	ECMP1	This bit controls to enable or disable the Analog Comparator 1 interrupt (CMP1INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
10	ECMP0	This bit controls to enable or disable the Analog Comparator 0 interrupt (CMP0INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
9	ESIU21	This bit controls to enable or disable the Serial Communication unit 21 interrupt (SIU21INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
8	ESIU20	This bit controls to enable or disable the Serial Communication unit 20 interrupt (SIU20INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
7	ETM3	This bit controls to enable or disable the 16-bit Timer 3 interrupt (TM3INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
6	ETM2	This bit controls to enable or disable the 16-bit Timer 2 interrupt (TM2INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt

Bit No.	Bit symbol name	Description
5	EFTM3	This bit controls to enable or disable the Functional Timer 3 interrupt (FTM3INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
4	EFTM2	This bit controls to enable or disable the Functional Timer 2 interrupt (FTM2INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
3	-	Reserved bit
2	ESIU11	This bit controls to enable or disable the Serial Communication unit 11 interrupt (SIU11INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
1	ESIU10	This bit controls to enable or disable the Serial Communication unit 10 interrupt (SIU10INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
0	EI2CU0	This bit controls to enable or disable the I^2C Bus unit 0 interrupt (I2CU0INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt

5.2.5 Interrupt Enable Register 67 (IE67)

IE67 is a special function register (SFR) to enable or disable the interrupt for each interrupt request.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
After the interrupt is accepted, the master interrupt enable flag (MIE) of the CPU is reset to "0", however, the each applicable flag of IE67 is not reset and remains "1".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IE67															
Byte	IE7								IE6							
Bit	-	ERTC	ELTBC2	ELTBC1	-	ELTBC0	ESIU51	ESIU50	ETM7	ETM6	EFTM7	EFTM6	ESIU41	ESIU40	ESIU31	ESIU30
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	-	Reserved bit
14	ERTC	This bit controls to enable or disable the Simplified RTC interrupt (RTCINT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
13	ELTBC2	This bit controls to enable or disable the Low speed Time base counter 2 interrupt (LTB2INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
12	ELTBC1	This bit controls to enable or disable the Low speed Time base counter 1 interrupt (LTB1INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
11	-	Reserved bit
10	ELTBC0	This bit controls to enable or disable the Low speed Time base counter 0 interrupt (LTB0INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
9	ESIU51[1]	This bit controls to enable or disable the Serial Communication unit 51 interrupt (SIU51INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
8	ESIU50	This bit controls to enable or disable the Serial Communication unit 50 interrupt (SIU50INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
7	ETM7 *1	This bit controls to enable or disable the 16-bit Timer 7 interrupt (TM7INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
6	ETM6 *1	This bit controls to enable or disable the 16-bit Timer 6 interrupt (TM6INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
5	EFTM7 *1	This bit controls to enable or disable the Functional Timer 7 interrupt (FTM7INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt

Bit No.	Bit symbol name	Description
4	EFTM6	This bit controls to enable or disable the Functional Timer 6 interrupt (FTM6INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
3	ESIU41	This bit controls to enable or disable the Serial Communication unit 41 interrupt (SIU41INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
2	ESIU40	This bit controls to enable or disable the Serial Communication unit 40 interrupt (SIU40INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
1	ESIU31	This bit controls to enable or disable the Serial Communication unit 31 interrupt (SIU31INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt
0	ESIU30	This bit controls to enable or disable the Serial Communication unit 30 interrupt (SIU30INT). 0: Disable the interrupt (Initial value) 1: Enable the interrupt

5.2.6 Interrupt Request Register 01 (IRQ01)

IRQ01 is a special function register (SFR) to request interrupts.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
QWDT bit of the IRQ01 register becomes "1" when the non-maskable Watch Dog Timer (WDT) interrupt occurs and the CPU goes to the interrupt routine regardless the value of the Master Interrupt Enable flag (MIE bit).
Each request flag of IRQ01 except for the QWDT bit becomes "1" when the interrupt request is generated, regardless of the values of the interrupt enable register 01(IE01) and master interrupt enable flag (MIE). At that time, it requests the interrupt to the CPU if the applicable flag of IE01 is "1" and the CPU accepts the interrupt if the MIE is "1" to goes to the interrupt routine.
Also, an interrupt can be generated by writing "1" to the request flag of IRQ01. In this case, the CPU goes to the interrupt routine immediately after the next one instruction is executed.
The applicable flag of IRQ01 becomes "0" automatically when the interrupt request is accepted by the CPU.

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

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

Bit No.	Bit symbol name	Description
15	QPI7	This bit controls to request the External interrupt 7 (EXI7INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
14	QPI6	This bit controls to request the External interrupt 6 (EXI6INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
13	QPI5	This bit controls to request the External interrupt 5 (EXI5INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
12	QPI4	This bit controls to request the External interrupt 4 (EXI4INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
11	QPI3	This bit controls to request the External interrupt 3 (EXI3INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
10	QPI2	This bit controls to request the External interrupt 2 (EXI2INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
9	QPI1	This bit controls to request the External interrupt 1 (EXI1INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt

Bit No.	Bit symbol name	Description
8	QPI0	This bit controls to request the External interrupt 0 (EXI0INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
7	-	Reserved bit
6	QVLS0	This bit controls to request the VLS0 interrupt (VLS0INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
5 to 1	-	Reserved bits
0	QWDT	This bit controls to request the WDT interrupt (WDTINT). 0: Not request the interrupt (Initial value) 1: Request the interrupt

[Note]

- **There is a risk of clearing other request flags of IRQ01 register, if writing to the specific bit of this register. Use the bit symbol to write to the specific bit.**
  See Section [5.3.8 Writing to IRQ01/IRQ23/IRQ45/IRQ67] for more detail.
- **Re-request interrupt by writing "1" to EEIR bit of EEITNTC register after writing to IRQ01 register, when expanded external interrupt is enabled.**

5.2.7 Interrupt Request Register 23 (IRQ23)

IRQ23 is a special function register (SFR) to request interrupts.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
Each request flag of IRQ23 except for the QWDT bit becomes "1" when the interrupt request is generated, regardless of the values of the interrupt enable register 23(IE23) and master interrupt enable flag (MIE). At that time, it requests the interrupt to the CPU if the applicable flag of IE23 is "1" and the CPU accepts the interrupt if the MIE is "1" to go to the interrupt routine.
Also, an interrupt can be generated by writing "1" to the request flag of IRQ23. In this case, the CPU goes to the interrupt routine immediately after the next one instruction is executed.
The applicable flag of IRQ23 becomes "0" automatically when the interrupt request is accepted by the CPU.

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IRQ23															
Byte	IRQ3								IRQ2							
Bit	QTM1	QTM0	QFTM1	QFTM0	QI2CM1	QI2CM0	-	QEXTX	-	QSAD	-	QSIU01	QSIU00	QMCS	QDMA	QCBU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	QTM1	This bit controls to request the 16-bit Timer 1 interrupt (TM1INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
14	QTM0	This bit controls to request the 16-bit Timer 0 interrupt (TM0INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
13	QFTM1	This bit controls to request the Functional Timer 1 interrupt (FTM1INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
12	QFTM0	This bit controls to request the Functional Timer 0 interrupt (FTM0INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
11	QI2CM1	This bit controls to request the I^2C Bus Master 1 interrupt (I2CM1INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
10	QI2CM0	This bit controls to request the I^2C Bus Master 0 interrupt (I2CM0INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
9	-	Reserved bit
8	QEXTX	This bit controls to request the Expanded External interrupt (EXTXINT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
7	-	Reserved bit

Bit No.	Bit symbol name	Description
6	QSAD	This bit controls to request the Successive approximation type A/D interrupt (SADINT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
5	-	Reserved bit
4	QSIU01	This bit controls to request the Serial Communication unit 01 interrupt (SIU01INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
3	QSIU00	This bit controls to request the Serial Communication unit 00 interrupt (SIU00INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
2	QMCS	This bit controls to request the MCU Status interrupt (MCSINT) *1. 0: Not request the interrupt (Initial value) 1: Request the interrupt
1	QDMA	This bit controls to request the DMA interrupt (DMACINT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
0	QCBU	This bit controls to request the Clock Backup interrupt (CBUINT) *2. 0: Not request the interrupt (Initial value) 1: Request the interrupt

*1: See Chapter 29 "Safety Function" for more details.
*2: See Chapter 6 "Clock Generation Circuit" for more details.

[Note]

- **There is a risk of clearing other request flags of IRQ23 register, if writing to the specific bit of this register. Use the bit symbol to write to the specific bit.**
  See Section [5.3.8 Writing to IRQ01/IRQ23/IRQ45/IRQ67] for more detail.
- **Re-request interrupt by writing "1" to EEIR bit of EEITNTC register after writing to IRQ23 register, when expanded external interrupt is enabled.**

5.2.8 Interrupt Request Register 45 (IRQ45)

IRQ45 is a special function register (SFR) to request interrupts.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
Each request flag of IRQ45 except for the QWDT bit becomes "1" when the interrupt request is generated, regardless of the values of the interrupt enable register 45(IE45) and master interrupt enable flag (MIE). At that time, it requests the interrupt to the CPU if the applicable flag of IE45 is "1" and the CPU accepts the interrupt if the MIE is "1" to goes to the interrupt routine.
Also, an interrupt can be generated by writing "1" to the request flag of IRQ45. In this case, the CPU goes to the interrupt routine immediately after the next one instruction is executed.
The applicable flag of IRQ45 becomes "0" automatically when the interrupt request is accepted by the CPU.

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IRQ45															
Byte	IRQ5								IRQ4							
Bit	QTM5	QTM4	QFTM5	QFTM4	QCMP1	QCMP0	QSIU21	QSIU20	QTM3	QTM2	QFTM3	QFTM2	-	QSIU11	QSIU10	QI2CU0
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/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	QTM5	This bit controls to request the 16-bit Timer 5 interrupt (TM5INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
14	QTM4	This bit controls to request the 16-bit Timer 4 interrupt (TM4INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
13	QFTM5	This bit controls to request the Functional Timer 5 interrupt (FTM5INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
12	QFTM4	This bit controls to request the Functional Timer 4 interrupt (FTM4INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
11	QCMP1	This bit controls to request the Analog Comparator 1 interrupt (CMP1INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
10	QCMP0	This bit controls to request the Analog Comparator 0 interrupt (CMP0INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
9	QSIU21	This bit controls to request the Serial Communication unit 21 interrupt (SIU21INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
8	QSIU20	This bit controls to request the Serial Communication unit 20 interrupt (SIU20INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt

Bit No.	Bit symbol name	Description
7	QTM3	This bit controls to request the 16-bit Timer 3 interrupt (TM3INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
6	QTM2	This bit controls to request the 16-bit Timer 2 interrupt (TM2INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
5	QFTM3	This bit controls to request the Functional Timer 3 interrupt (FTM3INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
4	QFTM2	This bit controls to request the Functional Timer 2 interrupt (FTM2INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
3	-	Reserved bit
2	QSIU11	This bit controls to request the Serial Communication unit 11 interrupt (SIU11INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
1	QSIU10	This bit controls to request the Serial Communication unit 10 interrupt (SIU10INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
0	QI2CU0	This bit controls to request the I^2C Bus unit 0 interrupt (I2CU0INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt

[Note]

- **There is a risk of clearing other request flags of IRQ45 register, if writing to the specific bit of this register. Use the bit symbol to write to the specific bit.**
  See Section [5.3.8 Writing to IRQ01/IRQ23/IRQ45/IRQ67] for more detail.
- **Re-request interrupt by writing "1" to EEIR bit of EEITNTC register after writing to IRQ45 register, when expanded external interrupt is enabled.**

5.2.9 Interrupt Request Register 67 (IRQ67)

IRQ67 is a special function register (SFR) to request interrupts.
The bits of the unavailable peripheral circuits are not writeable. They return "0" for reading.
Each request flag of IRQ67 except for the QWDT bit becomes "1" when the interrupt request is generated, regardless of the values of the interrupt enable register 67 (IE67) and master interrupt enable flag (MIE). At that time, it requests the interrupt to the CPU if the applicable flag of IE67 is "1" and the CPU accepts the interrupt if the MIE is "1" to goes to the interrupt routine.
Also, an interrupt can be generated by writing "1" to the request flag of IRQ67. In this case, the CPU goes to the interrupt routine immediately after the next one instruction is executed.
The applicable flag of IRQ67 becomes "0" automatically when the interrupt request is accepted by the CPU.

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IRQ67															
Byte	IRQ7								IRQ6							
Bit	-	QRTC	QLTBC2	QLTBC1	-	QLTBC0	QSIU51	QSIU50	QTM7	QTM6	QFTM7	QFTM6	QSIU41	QSIU40	QSIU31	QSIU30
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	-	Reserved bit
14	QRTC	This bit controls to request the Simplified RTC interrupt (RTCINT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
13	QLTBC2	This bit controls to request the Low speed Time base counter 2 interrupt (LTB2INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
12	QLTBC1	This bit controls to request the Low speed Time base counter 1 interrupt (LTB1INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
11	-	Reserved bit
10	QLTBC0	This bit controls to request the Low speed Time base counter 0 interrupt (LTB0INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
9	QSIU51	This bit controls to request the Serial Communication unit 51 interrupt (SIU51INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
8	QSIU50	This bit controls to request the Serial Communication unit 50 interrupt (SIU50INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
7	QTM7	This bit controls to request the 16-bit Timer 7 interrupt (TM7INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt

Bit No.	Bit symbol name	Description
6	QTM6	This bit controls to request the 16-bit Timer 6 interrupt (TM6INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
5	QFTM7	This bit controls to request the Functional Timer 7 interrupt (FTM7INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
4	QFTM6	This bit controls to request the Functional Timer 6 interrupt (FTM6INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
3	QSIU41	This bit controls to request the Serial Communication unit 41 interrupt (SIU41INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
2	QSIU40	This bit controls to request the Serial Communication unit 40 interrupt (SIU40INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
1	QSIU31	This bit controls to request the Serial Communication unit 31 interrupt (SIU31INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt
0	QSIU30	This bit controls to request the Serial Communication unit 30 interrupt (SIU30INT). 0: Not request the interrupt (Initial value) 1: Request the interrupt

[Note]

- **There is a risk of clearing other request flags of IRQ67 register, if writing to the specific bit of this register. Use the bit symbol to write to the specific bit.**
  See Section [5.3.8 Writing to IRQ01/IRQ23/IRQ45/IRQ67] for more detail.
- **Re-request interrupt by writing "1" to EEIR bit of EEITNTC register after writing to IRQ67 register, when expanded external interrupt is enabled.**

5.2.10 Interrupt Level Control Enable Register (ILEN)

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

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

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

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

[Note]

- **Disable the interrupt level control function by resetting the ILE bit to "0" after resetting the Interrupt level control register 0 to 7 (ILC0 to ILC1) to "0x0000" and confirming the current interrupt request level register (CIL) is "0x00" when the interrupt is disabled(IE0 to IE7 registers are "0x00").**
- **Enable the interrupt level control function by setting the ILE bit to "1" when the interrupt is disabled(IE0 to IE7 registers are "0") or master interrupt enable flag(MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.11 Current Interrupt Level Management Register (CIL)

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

Each bit of CIL is automatically set to "1", so it 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".

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

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

Bit No.	Bit symbol name	Description
7	CILN	This bit indicates the non-maskable interrupt is being processed or not. 0: The non-maskable interrupt is not being processed (Initial value) 1: The non-maskable interrupt is being processed
6 to 4	-	Reserved bits
3	CILM3	This bit indicates the maskable interrupt with level 4 is being processed or not. 0: The maskable interrupt with level 4 is not being processed (Initial value) 1: The maskable interrupt with level 4 is being processed
2	CILM2	This bit indicates the maskable interrupt with level 3 is being processed or not. 0: The maskable interrupt with level 3 is not being processed (Initial value) 1: The maskable interrupt with level 3 is being processed
1	CILM1	This bit indicates the maskable interrupt with level 2 is being processed or not. 0: The maskable interrupt with level 2 is not being processed (Initial value) 1: The maskable interrupt with level 2 is being processed
0	CILM0	This bit indicates the maskable interrupt with level 1 is being processed or not. 0: The maskable interrupt with level 1 is not being processed (Initial value) 1: The maskable interrupt with level 1 is being processed

5.2.12 Interrupt Level Control Register 0 (ILC0)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC0															
Byte	ILC01								ILC00							
Bit	-	-	ILVLS0H	ILVLS0L	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15,14	-	Reserved bits
13,12	ILVLS0H, ILVLS0L	This bit chooses the priority level of the VSL0 interrupt (VSL0INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
11 to 0	-	Reserved bits

[Note]

- **Write to this register when the interrupt is disabled (IE0 to IE7 registers are "0x00") or the 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)

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

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

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

Bit No.	Bit symbol name	Description
15,14	ILPI7H, ILPI7L	This bit chooses the priority level of the External interrupt 7 (EXI7INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
13,12	ILPI6H, ILPI6L	This bit chooses the priority level of the External interrupt 6 (EXI6INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
11,10	ILPI5H, ILPI5L	This bit chooses the priority level of the External interrupt 5 (EXI5INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
9,8	ILPI4H, ILPI4L	This bit chooses the priority level of the External interrupt 4 (EXI4INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
7,6	ILPI3H, ILPI3L	This bit chooses the priority level of the External interrupt 3 (EXI3INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
5,4	ILPI2H, ILPI2L	This bit chooses the priority level of the External interrupt 2 (EXI2INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

Bit No.	Bit symbol name	Description
3,2	ILPI1H, ILPI1L	This bit chooses the priority level of the External interrupt 1 (EXI1INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
1,0	ILPI0H, ILPI0L	This bit chooses the priority level of the External interrupt 0 (EXI0INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

[Note]

- **Write to this register when the interrupt is disabled (IE0 to IE7 registers are "0x00") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.14 Interrupt Level Control Register 2 (ILC2)

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

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

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

Bit No.	Bit symbol name	Description
15,14	-	Reserved bits
13,12	ILSADH, ILSADL	This bit chooses the priority level of the Successive approximation type A/D interrupt (SADINT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
11,10	-	Reserved bits
9,8	ILSIU01H, ILSIU01L	This bit chooses the priority level of the Serial communication unit 01 interrupt (SIU01INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
7,6	ILSIU00H, ILSIU00L	This bit chooses the priority level of the Serial communication unit 00 interrupt (SIU00INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
5,4	ILMCSH, ILMCSL	This bit chooses the priority level of the MCU status interrupt (MCSINT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
3,2	ILDMAH, ILDMAL	This bit chooses the priority level of the DMA interrupt [*1] (DMACINT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

Bit No.	Bit symbol name	Description
1,0	ILCBUH, ILCBUL	This bit chooses the priority level of the Clock Backup interrupt *2 (CBUINT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

*1: See Chapter 29 "Safety Function" for more details.
*2: See Chapter 6 "Clock Generation Circuit" for more details.

[Note]

- **Write to this register when the interrupt is disabled (IE0 to IE7 registers are "0x00") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.15 Interrupt Level Control Register 3 (ILC3)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC3															
Byte	ILC31								ILC30							
Bit	ILTM 1H	ILTM 1L	ILTM 0H	ILTM 0L	ILFT M1H	ILFT M1L	ILFT M0H	ILFT M0L	ILI2C M1H	ILI2C M1L	ILI2C M0H	ILI2C M0L	-	-	ILEX TXH	ILEX TXL
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	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15,14	ILTM1H, ILTM1L	This bit chooses the priority level of the 16-bit Timer 1 interrupt (TM1INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
13,12	ILTM0H, ILTM0L	This bit chooses the priority level of the 16-bit Timer 0 interrupt (TM0INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
11,10	ILFTM1H, ILFTM1L	This bit chooses the priority level of the Functional Timer 1 interrupt (FTM1INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
9,8	ILFTM0H, ILFTM0L	This bit chooses the priority level of the Functional Timer 0 interrupt (FTM0INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
7,6	ILI2CM1H, ILI2CM1L	This bit chooses the priority level of the I^2C Bus Master 1 interrupt (I2CM1INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
5,4	ILI2CM0H, ILI2CM0L	This bit chooses the priority level of the I^2C Bus Master 0 interrupt (I2CM0INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

Bit No.	Bit symbol name	Description
3,2	-	Reserved bits
1,0	ILEXTXH, ILEXTXL	This bit chooses the priority level of the expanded external interrupt (EXTXINT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

[Note]

- **Write to this register when the interrupt is disabled (IE0 to IE7 registers are "0x00") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.16 Interrupt Level Control Register 4 (ILC4)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC4															
Byte	ILC41								ILC40							
Bit	ILTM 3H	ILTM 3L	ILTM 2H	ILTM 2L	ILFT M3H	ILFT M3L	ILFT M2H	ILFT M2L	-	-	ILSIU 11H	ILSIU 11L	ILSIU 10H	ILSIU 10L	ILI2C U0H	ILI2C U0L
R/W	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

Bit No.	Bit symbol name	Description
15,14	ILTM3H, ILTM3L	This bit chooses the priority level of the 16-bit Timer 3 interrupt (TM3INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
13,12	ILTM2H, ILTM2L	This bit chooses the priority level of the 16-bit Timer 2 interrupt (TM2INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
11,10	ILFTM3H, ILFTM3L	This bit chooses the priority level of the Functional Timer 3 interrupt (FTM3INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
9,8	ILFTM2H, ILFTM2L	This bit chooses the priority level of the Functional Timer 2 interrupt (FTM2INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
7,6	-	Reserved bits
5,4	ILSIU11H, ILSIU11L	This bit chooses the priority level of the Serial Communication unit 11 interrupt (SIU11INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

Bit No.	Bit symbol name	Description
3,2	ILSIU10H, ILSIU10L	This bit chooses the priority level of the Serial Communication unit 10 interrupt (SIU10INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
1,0	ILI2CU0H, ILI2CU0L	This bit chooses the priority level of the I^2C Bus unit 0 interrupt (I2CU0INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

[Note]

- **Write to this register when the interrupt is disabled (IE0 to IE7 registers are "0x00") or the 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)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC5															
Byte	ILC51								ILC50							
Bit	ILTM 5H	ILTM 5L	ILTM 4H	ILTM 4L	ILFT M5H	ILFT M5L	ILFT M4H	ILFT M4L	ILCM P1H	ILCM P1L	ILCM P0H	ILCM P0L	ILSIU 21H	ILSIU 21L	ILSIU 20H	ILSIU 20L
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15,14	ILTM5H, ILTM5L	This bit chooses the priority level of the 16-bit Timer 5 interrupt (TM5INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
13,12	ILTM4H, ILTM4L	This bit chooses the priority level of the 16-bit Timer 4 interrupt (TM4INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
11,10	ILFTM5H, ILFTM5L	This bit chooses the priority level of the Functional Timer 5 interrupt (FTM5INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
9,8	ILFTM4H, ILFTM4L	This bit chooses the priority level of the Functional Timer 4 interrupt (FTM4INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
7,6	ILCMP1H, ILCMP1L	This bit chooses the priority level of the Analog Comparator 1 interrupt (CMP1INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
5,4	ILCMP0H, ILCMP0L	This bit chooses the priority level of the Analog Comparator 0 interrupt (CMP0INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

Bit No.	Bit symbol name	Description
3,2	ILSIU21H, ILSIU21L *1	This bit chooses the priority level of the Serial Communication unit 21 interrupt (SIU21INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
1,0	ILSIU20H, ILSIU20L *1	This bit chooses the priority level of the Serial Communication unit 20 interrupt (SIU20INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

*1: This bit is available to use only on the 80pin or 100pin products.

[Note]

- **Write to this register when the interrupt is disabled (IE0 to IE7 registers are "0x00") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.18 Interrupt Level Control Register 6 (ILC6)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC6															
Byte	ILC61								ILC60							
Bit	ILTM 7H	ILTM 7L	ILTM 6H	ILTM 6L	ILFT M7H	ILFT M7L	ILFT M6H	ILFT M6L	ILSIU 41H	ILSIU 41L	ILSIU 40H	ILSIU 40L	ILSIU 31H	ILSIU 31L	ILSIU 30H	ILSIU 30L
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15,14	ILTM7H, ILTM7L	This bit chooses the priority level of the 16-bit Timer 7 interrupt (TM7INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
13,12	ILTM6H, ILTM6L	This bit chooses the priority level of the 16-bit Timer 6 interrupt (TM6INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
11,10	ILFTM7H, ILFTM7L	This bit chooses the priority level of the Functional Timer 7 interrupt (FTM7INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
9,8	ILFTM6H, ILFTM6L	This bit chooses the priority level of the Functional Timer 6 interrupt (FTM6INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
7,6	ILSIU41H, ILSIU41L	This bit chooses the priority level of the Serial Communication unit 41 interrupt (SIU41INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
5,4	ILSIU40H, ILSIU40L	This bit chooses the priority level of the Serial Communication unit 40 interrupt (SIU40INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

Bit No.	Bit symbol name	Description
3,2	ILSIU31H, ILSIU31L	This bit chooses the priority level of the Serial Communication unit 31 interrupt (SIU31INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
1,0	ILSIU30H, ILSIU30L	This bit chooses the priority level of the Serial Communication unit 30 interrupt (SIU30INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

[Note]

- **Write to this register when the interrupt is disabled (IE0 to IE7 registers are "0x00") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.19 Interrupt Level Control Register 7 (ILC7)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC7															
Byte	ILC71								ILC70							
Bit	-	-	ILRTCH	ILRTCL	ILLTBC2H	ILLTBC2L	ILLTBC1H	ILLTBC1L	-	-	ILLTBC0H	ILLTBC0L	ILSIU51H	ILSIU51L	ILSIU50H	ILSIU50L
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15,14	-	Reserved bits
13,12	ILRTCH, ILRTCL	This bit chooses the priority level of the Simplified RTC interrupt (RTCINT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
11,10	ILLTBC2H, ILLTBC2L	This bit chooses the priority level of the Low speed Time base counter 2 interrupt (LTB2INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
9,8	ILLTBC1H, ILLTBC1L	This bit chooses the priority level of the Low speed Time base counter 1 interrupt (LTB1INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
7,6	-	Reserved bits
5,4	ILLTBC0H, ILLTBC0L	This bit chooses the priority level of the Low speed Time base counter 0 interrupt (LTB0INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)
3,2	ILSIU51H, ILSIU51L	This bit chooses the priority level of the Serial Communication unit 51 interrupt (SIU51INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

Bit No.	Bit symbol name	Description
1,0	ILSIU50H, ILSIU50L	This bit chooses the priority level of the Serial Communication unit 50 interrupt (SIU50INT). 00: Level 1 (Priority is lowest) (Initial) 01: Level 2 10: Level 3 11: Level 4 (Priority is highest)

[Note]

- **Write to this register when the interrupt is disabled (IE0 to IE7 registers are "0x00") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.3 Description of Operation

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

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

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

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

Interrupt source number (priority)	Register assignment			Interrupt vector address	Mask	External/ internal source	Interrupt source	Interrupt source symbol
	IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)					
1 (high)	IRQ0[0]	-	-	0x0008	Disabled	Internal	WDT interrupt	WDTINT
2	-	-	-	0x000A	Disabled		-	-
3	IRQ0[6]	IE0[6]	ILC0[13:12]	0x000C	Enabled		VLS0 interrupt	VLS0INT
4	IRQ0[7]	IE0[7]	ILC0[15:14]	0x000E	Enabled		-	-
5	IRQ1[0]	IE1[0]	ILC1[1:0]	0x0010	Enabled	External	External interrupt 0	EXI0INT
6	IRQ1[1]	IE1[1]	ILC1[3:2]	0x0012	Enabled		External interrupt 1	EXI1INT
7	IRQ1[2]	IE1[2]	ILC1[5:4]	0x0014	Enabled		External interrupt 2	EXI2INT
8	IRQ1[3]	IE1[3]	ILC1[7:6]	0x0016	Enabled		External interrupt 3	EXI3INT
9	IRQ1[4]	IE1[4]	ILC1[9:8]	0x0018	Enabled		External interrupt 4	EXI4INT
10	IRQ1[5]	IE1[5]	ILC1[11:10]	0x001A	Enabled		External interrupt 5	EXI5INT
11	IRQ1[6]	IE1[6]	ILC1[13:12]	0x001C	Enabled		External interrupt 6	EXI6INT
12	IRQ1[7]	IE1[7]	ILC1[15:14]	0x001E	Enabled		External interrupt 7	EXI7INT
13	IRQ2[0]	IE2[0]	ILC2[1:0]	0x0020	Enabled	Internal	Clock backup interrupt	CBUINT
14	IRQ2[1]	IE2[1]	ILC2[3:2]	0x0022	Enabled		DMA controller interrupt	DMACINT
15	IRQ2[2]	IE2[2]	ILC2[5:4]	0x0024	Enabled		MCU status interrupt*1	MCSINT
16	IRQ2[3]	IE2[3]	ILC2[7:6]	0x0026	Enabled		Serial communication unit 00 interrupt	SIU00INT
17	IRQ2[4]	IE2[4]	ILC2[9:8]	0x0028	Enabled		Serial communication unit 01 interrupt	SIU01INT
18	IRQ2[5]	IE2[5]	ILC2[11:10]	0x002A	Enabled		-	-
19	IRQ2[6]	IE2[6]	ILC2[13:12]	0x002C	Enabled		SA-ADC interrupt	SADINT
20	IRQ2[7]	IE2[7]	ILC2[15:14]	0x002E	Enabled		-	-
21	IRQ3[0]	IE3[0]	ILC3[1:0]	0x0030	Enabled	External	Extended external interrupt	EXTXINT
22	IRQ3[1]	IE3[1]	ILC3[3:2]	0x0032	Enabled	Internal	-	-
23	IRQ3[2]	IE3[2]	ILC3[5:4]	0x0034	Enabled		I^2C bus master 0 interrupt	I2CM0INT
24	IRQ3[3]	IE3[3]	ILC3[7:6]	0x0036	Enabled		I^2C bus master 1 interrupt	I2CM1INT
25	IRQ3[4]	IE3[4]	ILC3[9:8]	0x0038	Enabled		Functional timer 0 interrupt	FTM0INT
26	IRQ3[5]	IE3[5]	ILC3[11:10]	0x003A	Enabled		Functional timer 1 interrupt	FTM1INT
27	IRQ3[6]	IE3[6]	ILC3[13:12]	0x003C	Enabled		16-bit timer 0 interrupt	TM0INT
28	IRQ3[7]	IE3[7]	ILC3[15:14]	0x003E	Enabled		16-bit timer 1 interrupt	TM1INT
29	IRQ4[0]	IE4[0]	ILC4[1:0]	0x0040	Enabled	Internal	I^2C bus unit 0 interrupt	I2CU0INT
30	IRQ4[1]	IE4[1]	ILC4[3:2]	0x0042	Enabled		Serial communication unit 10 interrupt	SIU10INT
31	IRQ4[2]	IE4[2]	ILC4[5:4]	0x0044	Enabled		Serial communication unit 11 interrupt	SIU11INT
32	IRQ4[3]	IE4[3]	ILC4[7:6]	0x0046	Enabled		-	-
33	IRQ4[4]	IE4[4]	ILC4[9:8]	0x0048	Enabled		Functional timer 2 interrupt	FTM2INT
34	IRQ4[5]	IE4[5]	ILC4[11:10]	0x004A	Enabled		Functional timer 3 interrupt	FTM3INT

*1 The MCU status interrupt occurs when one of the following interrupt requests is enabled:
- An interrupt request due to occurrence of a RAM parity error
- An interrupt request due to completion of an automatic CRC calculation
- An interrupt request due to completion of erasing/writing to the data flash

Enabling/disabling each interrupt request can be controlled through SFRs. See Chapter 29 "Safety Function" for details.

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

Interrupt source number (priority)	Register assignment			Interrupt vector address	Mask	External/ internal source	Interrupt source	Interrupt source symbol
	IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)					
35	IRQ4[6]	IE4[6]	ILC4[13:12]	0x004C	Enabled	Internal	16-bit timer 2 interrupt	TM2INT
36	IRQ4[7]	IE4[7]	ILC4[15:14]	0x004E	Enabled		16-bit timer 3 interrupt	TM3INT
37	IRQ5[0]	IE5[0]	ILC5[1:0]	0x0050	Enabled		Serial communication unit 20 interrupt	SIU20INT
38	IRQ5[1]	IE5[1]	ILC5[3:2]	0x0052	Enabled		Serial communication unit 21 interrupt	SIU21INT
39	IRQ5[2]	IE5[2]	ILC5[5:4]	0x0054	Enabled		Analogue Comparator 0 interrupt	CMP0INT
40	IRQ5[3]	IE5[3]	ILC5[7:6]	0x0056	Enabled		Analogue Comparator 1 interrupt	CMP1INT
41	IRQ5[4]	IE5[4]	ILC5[9:8]	0x0058	Enabled		Functional timer 4 interrupt	FTM4INT
42	IRQ5[5]	IE5[5]	ILC5[11:10]	0x005A	Enabled		Functional timer 5 interrupt	FTM5INT
43	IRQ5[6]	IE5[6]	ILC5[13:12]	0x005C	Enabled		16-bit timer 4 interrupt	TM4INT
44	IRQ5[7]	IE5[7]	ILC5[15:14]	0x005E	Enabled		16-bit timer 5 interrupt	TM5INT
45	IRQ6[0]	IE6[0]	ILC6[1:0]	0x0060	Enabled	Internal	Serial communication unit 30 interrupt	SIU30INT
46	IRQ6[1]	IE6[1]	ILC6[3:2]	0x0062	Enabled		Serial communication unit 31 interrupt	SIU31INT
47	IRQ6[2]	IE6[2]	ILC6[5:4]	0x0064	Enabled		Serial communication unit 40 interrupt	SIU40INT
48	IRQ6[3]	IE6[3]	ILC6[7:6]	0x0066	Enabled		Serial communication unit 41 interrupt	SIU41INT
49	IRQ6[4]	IE6[4]	ILC6[9:8]	0x0068	Enabled		Functional timer 6 interrupt	FTM6INT
50	IRQ6[5]	IE6[5]	ILC6[11:10]	0x006A	Enabled		Functional timer 7 interrupt	FTM7INT
51	IRQ6[6]	IE6[6]	ILC6[13:12]	0x006C	Enabled		16-bit timer 6 interrupt	TM6INT
52	IRQ6[7]	IE6[7]	ILC6[15:14]	0x006E	Enabled		16-bit timer 7 interrupt	TM7INT
53	IRQ7[0]	IE7[0]	ILC7[1:0]	0x0070	Enabled	Internal	Serial communication unit 50 interrupt	SIU50INT
54	IRQ7[1]	IE7[1]	ILC7[3:2]	0x0072	Enabled		Serial communication unit 51 interrupt	SIU51INT
55	IRQ7[2]	IE7[2]	ILC7[5:4]	0x0074	Enabled		Low-speed time base counter 0 interrupt	LTB0INT
56	IRQ7[3]	IE7[3]	ILC7[7:6]	0x0076	Enabled		-	-
57	IRQ7[4]	IE7[4]	ILC7[9:8]	0x0078	Enabled		Low-speed time base counter 1 interrupt	LTB1INT
58	IRQ7[5]	IE7[5]	ILC7[11:10]	0x007A	Enabled		Low-speed time base counter 2 interrupt	LTB2INT
59	IRQ7[6]	IE7[6]	ILC7[13:12]	0x007C	Enabled		Simplified RTC interrupt	RTCINT
60 (low)	IRQ7[7]	IE7[7]	ILC7[15:14]	0x007E	Enabled		-	-

[Note]

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

5.3.1 Maskable Interrupt Processing

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

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

5.3.2 Non-Maskable Interrupt Processing

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

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

5.3.3 Software Interrupt Processing

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

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

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

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

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

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

State A: Maskable interrupt is being executed

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

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

Example of description: State A-1-1

For assembly language:

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

For C language:

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

A-1-2: When multiple interrupts are enabled

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

Example of description:
State A-1-2

For assembly language:

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

For C language:

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

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

A-2-1: When multiple interrupts are disabled

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

Example of description:
State A-2-1

For assembly language:

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

```
Sub_1;        ;
RT            ; Return PC from LR
              ; End of subroutine
```

For C language:

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

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

[Note]

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

A-2-2: When multiple interrupts are enabled

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

Example of description:
Status A-2-2

For assembly language:

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

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

For C language:

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

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

State B: Non-maskable interrupt is being processed

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

Example of description:
Status B-1

For assembly language:

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

For C language:

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

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

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

Example of description:
Status B-2

For assembly language:

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

```
Sub_1;
                                :
RT                  ; Return PC from LR
                    ; End of subroutine
```

For C language:

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

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

5.3.5 Flow Charts When Interrupt Level Control Is Enabled

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

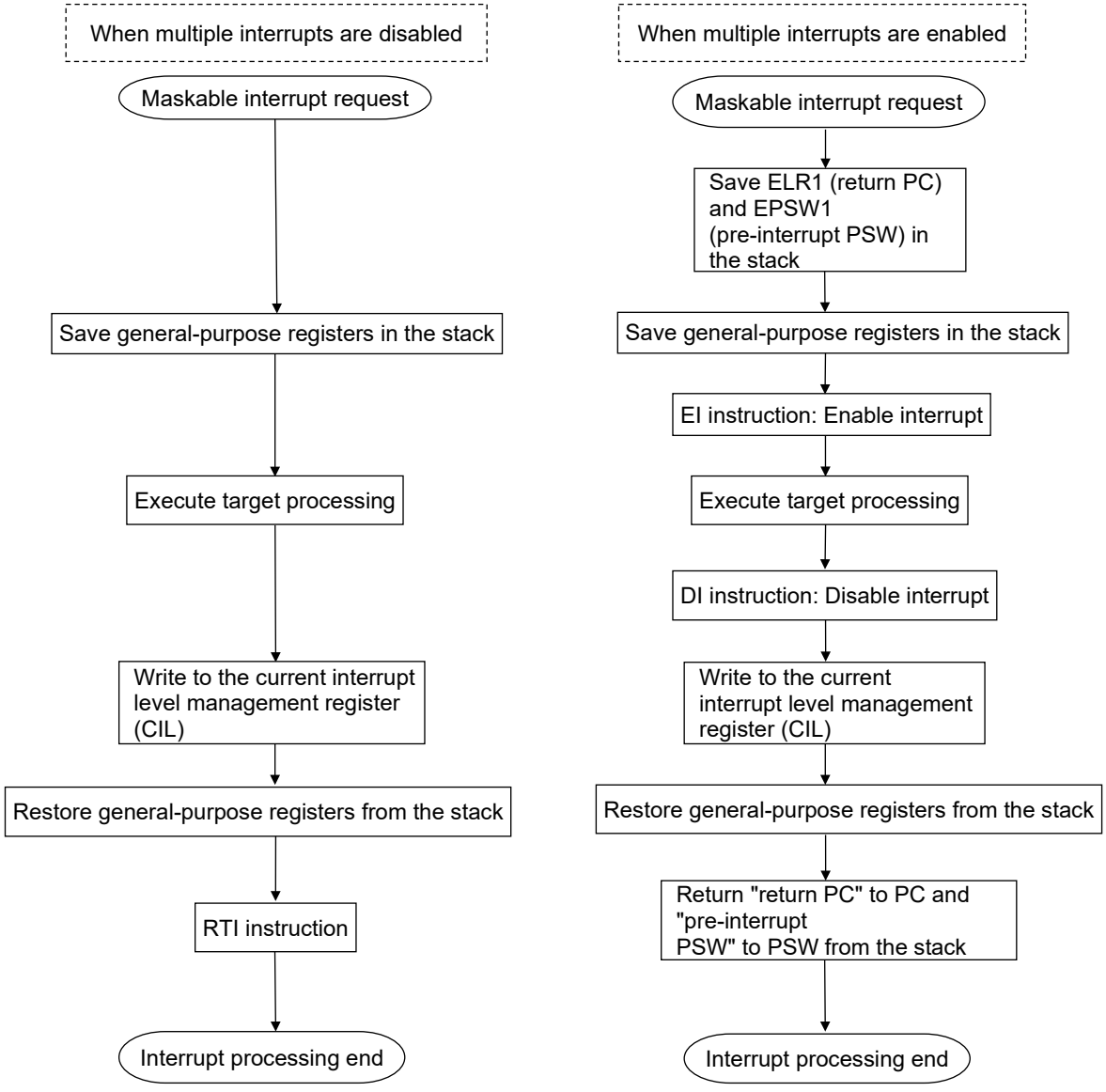


Figure 5-1 Maskable Interrupt Processing Flow

[Note]

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

5.3.6 How To Write Interrupt Processing When Interrupt Level Control Enabled

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

5.3.6.1 Description of Interrupt Function to Disable Multiple Interrupts

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

Example of description

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

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

Example of output

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

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

Example of description

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

Example of output

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

;;  func();
    bl      _func

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

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

[Note]

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

5.3.6.2 Description of Interrupt Function to Enable Multiple Interrupts

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

Example of description

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

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

Example of output

```
_intr_fn_20       :
    push      elr,    epsw
    push      er0

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

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

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

5.3.7 Interrupt Disable State

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

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

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

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

5.3.8 Writing to IRQ01/IRQ23/IRQ45/IRQ67

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

Example of description

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

clear bit (QLTBC0);
```

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

Chapter 6 Clock generation Circuit

6. Clock Generation Circuit

6.1 General Description

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

Table 6-1 Clocks generated by the clock generation circuit

Clock Name	Symbol	Description
Low-speed clock	LSCLK	Low speed clock for peripherals (32.768kHz)
Simplified RTC clock*1	RTCCLK	Low speed clock for the simplified RTC (32.768kHz)
High-speed clock	HSCLK	High speed clock for peripherals (Max. 24MHz) The frequency is selected by code option. 24MHz (multiplying the LSCLK by 732 = 23.986176MHz) 16MHz (multiplying the LSCLK by 488 = 15.990784MHz)
CPU clock	CPUCLK	CPU operating clock (32.768kHz or Max. 24MHz): The maximum frequency depends on the CPU operation mode(See Table 6-2)
System clock	SYSTEMCLK	System control clock: The frequency is the same as CPU clock.
Low-speed output clock	OUTLSCLK	Low speed output from a general port (32.768kHz)
High-speed output clock	OUTHSCLK	High speed output from an general port (Max. 12MHz)
WDT clock	WDTCLK	Clock for the watch dog timer (1.024kHz)

*1: Available except for ML62Q1300 group

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
  - Adjustable to ±1% by using the frequency adjustment function ($V_{DD} \geq 1.8V$)
  - A crystal resonator is connectable*1
  - In case the low-speed crystal oscillation stopped, the clock is automatically switched to the low-speed RC oscillation (clock backup function).*1
  - A low-speed external clock is available to input to XT1 pin*1
  - The crystal oscillation clock and the low-speed external clock each is continuously supplied during the reset input pin reset.*1
- Simplified RTC clock*1
  - Operating by the low-speed clock
- High-speed oscillation circuit
  - PLL oscillation mode (16 MHz or 24 MHz is choosable for the PLL reference frequency by the code option)
  - High-speed clock wake-up time is choosable
- WDT clock
  - RC1K oscillation circuit
  - The RC1K clock or the 1.024 kHz divided from the low-speed RC oscillation clock is choosable by the code option.

*1 Available except for ML62Q1300 group

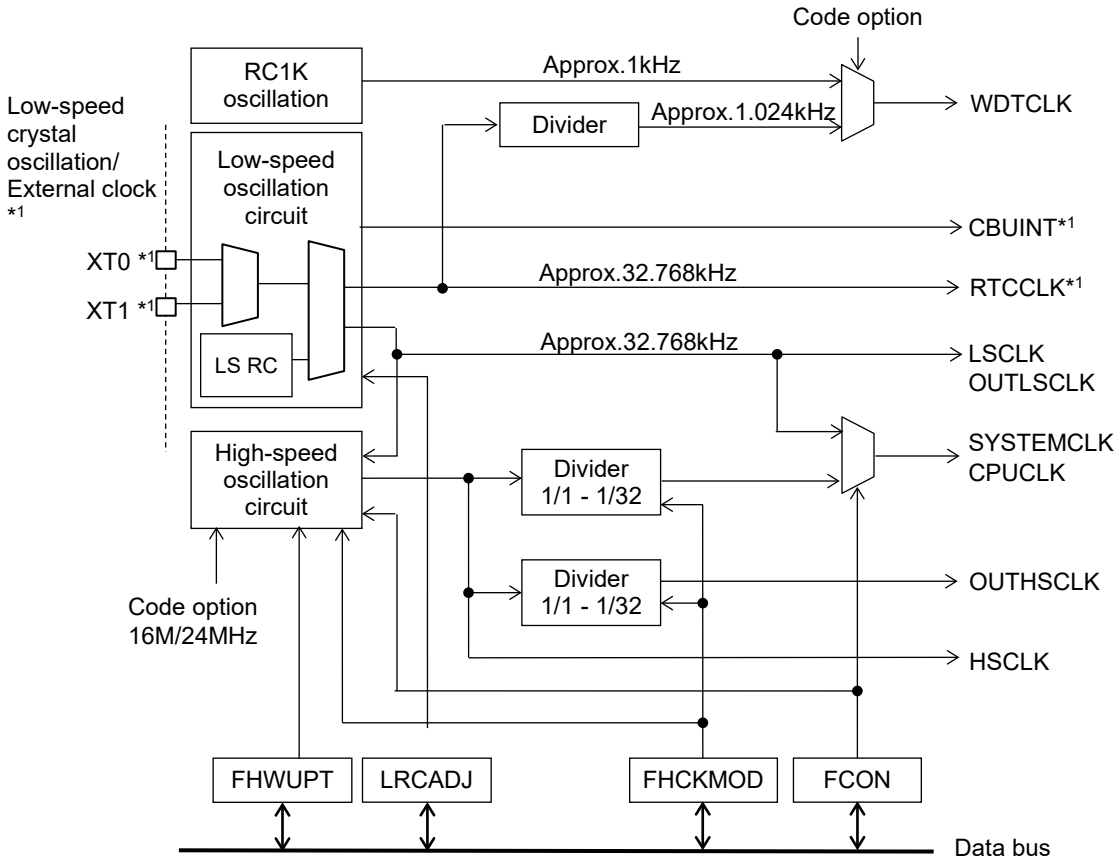
Table 6-2 shows relation of CPU operation mode and PLL oscillation reference frequency.
The CPU operation mode and the PLL oscillation reference frequency are choosable by the code option. See Chapter 26 "Code Option" for more details.

Table 6-2 CPU operation mode and PLL oscillation reference frequency

PLL oscillation reference frequency			Maximum operating frequency		
Selected by code option	The number of multiplying the LSCLK	Frequency	SYSTEMCLK		HSCLK
			Wait mode	No wait mode	
24MHz	732	23.986176 MHz	24MHz	6MHz	24MHz
16MHz	488	15.990784 MHz	16MHz	8MHz	16MHz

6.1.2 Configuration

Figure 6-1shows the configuration of the clock generation circuit.
Table 6-3 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
CBUINT*1	: Clock backup interrupt register

*1 Available except for ML62Q1300 group

Figure 6-1 Configuration of Clock Generation Circuit

[Note]
- **After the power-on or the system reset, LSCLK (32.768 kHz) is initially chosen as SYSTEMCLK.**

Table 6-3 Operating clock list in each function

Function	System clock or CPU clock SYSTEMCLK/ CPUCLK	Low-speed clock LSCLK	Simplified RTC clock RTCCLK*2	High speed clock HSCLK	WDT clock WDTCLK
CPU	●	-	-	-	-
RAM	●	-	-	-	-
Watchdog timer	●	-	-	-	●
External interrupt control	●	●*1	-	●*1	-
Low speed time base counter	●	●	-	-	-
16-bit timer	●	●	-	●	-
Functional timer	●	●	-	●	-
Serial communication unit	●	●	-	●	-
I2C bus unit	●	●	-	●	-
I2C bus master	●	-	-	●	-
Buzzer	●	●	-	-	-
SA type A/D converter	●	●	-	●	-
D/A converter	●	-	-	-	-
Analog comparator	●	●*1	-	●*1	-
Voltage Level Supervisor(VLS)	●	●*1	-	●*1	-
Simplified RTC*2	●	-	●	-	-
LCD driver *3	●	●	-	-	-
DMA controller	●	-	-	-	-
CRC calculator	●	-	-	-	-
Flash (BGO operation)	●	-	-	●	-

●: The clock is supplied -: The clock is Not supplied

*1 : The clock is supplied for start control or sampling

*2 : Available except for ML62Q1300 group

*3 : Available except for ML62Q1300 group and ML62Q1500/ML62Q1800 group

6.1.3 List of Pins

The output pins of the high-speed/low-speed clocks are assigned to the shared function of general purpose ports. For details of pin assignment and the shared function of general purpose ports, see Table 1-7, Table 1-8, Table 1-9 or Table 1-10.

Pin name	I/O	Function
OUTLSCLK	O	Low-speed clock output
OUTHSCLK	O	High-speed clock output
XT0 *1	I	Low-speed crystal resonator connect pin
XT1 *1	O/I	Low-speed crystal resonator connect pin / Low-speed external clock input pin

*1 : ML62Q1300 group is not equipped with

6.2 Description of Registers

6.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF002	High-speed clock mode register	FHCKMODL	FHCKMOD	R/W	8/16	0x00
0xF003		FHCKMODH		R/W	8	0x44
0xF004	Low-speed clock mode register *1	FLMOD	-	R/W	8	0x00
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	0x00
0xF009	Reserved	-	-	-	-	-
0xF00A	Backup Control register *1	FBUCON	-	R/W	8	0x00
0xF00B	Reserved	-	-	-	-	-
0xF00C	Backup Clock Status register *1	FBUSTAT	-	R	8	0x01
0xF00D	Reserved	-	-	-	-	-
0xF080	Low-speed RC oscillation frequency adjustment register	LRCADJ	-	R/W	8	0x00
0xF0C4	Clock Backup Test Mode Acceptor *1	FBTACP	-	W	8	0x00
0xF0C5	Reserved	-	-	-	-	-
0xF0C6	Clock Backup Test Mode register *1	FBTCON	-	R/W	8	0x00
0xF0C7	Reserved	-	-	-	-	-

*1 : Unavailable and the initial value is “0x00” for ML62Q1300 group.

6.2.2 High-Speed Clock Mode Register (FHCKMOD)

FHCKMOD is a special function register (SFR) to choose the oscillation mode of the high-speed clock oscillation circuit (PLL oscillation circuit) and the frequency of high-speed clock.

Address: 0xF002(FHCKMODL/FHCKMOD), 0xF003(FHCKMODH)
Access: R/W
Access size: 8/16bit
Initial value: 0x4400

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

Bit No.	Bit symbol name	Description
15	-	Reserved bit
14 to 12	OUTC2 to OUTC0	These bits are used to choose a division ratio of the frequency of the high-speed output clock (OUTHSCLK) output from the general port. 000: Do not use(HSCLK) 001: 1/2 HSCLK 010: 1/4 HSCLK 011: 1/8 HSCLK 100: 1/16 HSCLK (Initial value) 101: 1/32 HSCLK 110: Do not use (1/32 HSCLK) 111: Do not use (1/32 HSCLK)
11	-	Reserved bit
10 to 8	SYSC2 to SYSC0	These bits are used to choose a division ratio of the frequency of the high-speed of the high-speed clock used for the system clock (SYSTEMCLK). Choose a proper division ratio of the frequency, so that the frequency does not exceed the maximum frequency of the CPU operating frequency shown in the Table 6-2 "CPU operation mode and PLL oscillation reference frequency". 000: HSCLK (in Wait mode) 1/2 HSCLK *1 (in No wait mode) 001: 1/2 HSCLK (in Wait mode) 1/2 HSCLK *1 (in No wait mode) 010: 1/4 HSCLK 011: 1/8 HSCLK 100: 1/16 HSCLK (Initial value) 101: 1/32 HSCLK 110: Do not use (1/32 HSCLK) 111: Do not use (1/32 HSCLK) *1 : Do not use when the PLL reference frequency is 24MHz.
7 to 1	-	Reserved bits
0	HOSCM0	This bit is used to choose the oscillation mode of the high-speed oscillation circuit (PLL oscillation mode) 0: PLL oscillation mode (Initial value) 1: Do not use (PLL oscillation mode)

[Note]

- **When the voltage of V_{DD} is 1.6V ≤ V_{DD} < 1.8V, set the system clock to 4 MHz or lower. If it exceeds 4MHz, 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 ≤ V_{DD} < 1.8V : Choose 4 MHz or lower
  1.8V ≤ V_{DD} ≤ 5.5V : Choose 12 MHz or lower

6.2.3 Low-speed Clock Mode Register (FLMOD)

FLMOD is a special function register (SFR) to control the low-speed clock (LSCLK).
The FLMOD is initialized by only the Power-On-Reset.
This register is unavailable for ML62Q1300 group.

Address: 0xF004 (FLMOD)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FLMOD							
Bit	-	-	-	-	-	-	-	-	LMOD1	LMOD0	-	LFLTSEL	-	-	LOSCM1	LOSCM0
R/W	R	R	R	R	R	R	R	R	R/W	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

Bit No.	Bit symbol name	Description
7,6	LMOD1, LMOD0	These bits are used to choose the mode of low-speed crystal oscillation circuit (except for low-speed external clock). These bits are unchangeable when the LOSCM0 bit is "1". 00: Standard mode (Initial value) 01: Low power consumption mode 10: Do not use 11: Tough mode The low power consumption mode reduces the current consumption by lowering the oscillation margin than the standard mode. The tough mode increases the oscillation margin and heightens the resistance against leakage between the pins, increases the current consumption.
5	-	Reserved bit
4	LFLTSEL	This bit is used to apply a noise filter to the low-speed crystal oscillation clock or the low-speed external clock. This bit is unchangeable when the LOSCM0 bit is "1". 0: Use the noise filter (Initial value) 1: Do not use the noise filter
3,2	-	Reserved bits
1,0	LOSCM1, LOSCM0	These bits are used to choose the clock source of low-speed clock (LSCLK). Low-speed RC oscillation clock (approx.32.768kHz), low-speed crystal oscillation clock (32.768kHz) or low-speed external clock (32.768kHz) can be chosen. 00: Low-speed RC oscillation clock (Initial) 01: Low-speed Crystal oscillation clock 10: Do not use (Low-speed RC oscillation clock) 11: Low-speed External clock input(XT1 pin) The low-speed crystal oscillation clock and the low-speed external clock mode are not available to switch directly each other. When switching to another of these clocks, change the mode to the low-speed RC oscillation clock once.

[Note]

- **Do not change the LOSCM1 bit and LOSCM0 bit when the ENOSC bit in Clock Control Register (FCON) is "1", otherwise the operation is not guaranteed.**

6.2.4 Clock Control Register (FCON)

FCON is a special function register (SFR) to control the clock generation circuit and choose the system clock.

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	-															
Byte	-								FCON							
Bit	-	-	-	-	-	-	-	-	LPLL	-	-	-	-	-	ENOSC	SELSCLK
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	LPLL	This bit indicates that the frequency of the PLL oscillation is within the target error. The LPLL has the read-only attribute. 0: The frequency of PLL oscillation is out of the target error or the PLL oscillation is stopped (Initial value) 1: The frequency of PLL oscillation is within the target error
6 to 2	-	Reserved bits
1	ENOSC	This bit is used to enable/start or disable/stop the oscillation of the high-speed clock oscillation circuit. 0: Disable/Stop the high-speed clock oscillation (Initial value) 1: Enable/Star the high-speed clock oscillation
0	SELSCLK	This bit is used to choose 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 chosen for the system clock. 0: LSCLK (Initial value) 1: High-speed clock chosen by the SYSC2 to SYSC0 bit

[Note]

- **ENOSC bit and SELSCLK bit are forcibly set to "1" after releasing the HALT-H mode.**

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

FHWUPT is a special function register (SFR) to choose the wake-up time of the high-speed clock.
FHWUPT is writable only when the high-speed oscillation is stopped.
See Table 4-5 "Wake-up Time from Standby Mode" in the Chapter 4 for details about the wake-up time from the standby modes.

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

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

Bit No.	Bit symbol name	Description
7 to 3	-	Reserved bits
2,1	rsvd	Reserved bits. Always write "0" to these bits.
0	FHUT0	This bit is used to choose the wake-up time of the high-speed clock. Two options are chosen. One is to supply the frequency-stable clock after enabling the high-speed oscillation. The other one is to supply clock before the frequency gets stable. In the case this bit is set to "1", the clock starts to be supplied approx. 30us after enabling the high-speed oscillation. Then, the frequency of the clock is gradually getting higher and reaches to the target frequency in approx. 2ms. The frequency in the approx.2ms is not guaranteed as the specification, however it is useable for the system clock. 0: The clock starts to be supplied after it's stabilized: approx. 2.5 ms (Initial value) 1: The clock starts to be supplied before it's stabilized: approx. 30 µs

6.2.6 Backup Control Register (FBUCON)

FBUCON is a special function register (SFR) to switch the backup clock.
This register is unavailable for ML62Q1300 group.
The FBUCON is used only when choosing the low-speed crystal oscillation clock or low-speed external clock input as LSCLK.
The FBUCON is initialized by only the Power-On-Reset.

Address: 0xF00A (FBUCON)
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	-															
Byte	-								FBUCON							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LOSCB
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	LOSCB	This bit is used to switch the backup clock. The bit is automatically set to "1" when LOSCM1-0 bits of Low-speed Clock Mode Register (FLMOD) is set to 0x1 or 0x3. Write "1" to this bit to switch the low-speed clock to the low-speed crystal oscillation clock or low-speed external clock. In that case, confirm LOSCS bit of Backup Clock Status Register (FBUSTAT) is "0". 0: Low-speed clock is switched to the low-speed crystal oscillation clock (Initial value) 1: Low-speed clock is switched to the low-speed RC oscillation clock

[Note]
- **Writing "0" to the LOSCB bit is invalid.**
- **Insert two NOP instructions in the next to the instruction of that writes "1" to the LOSCB bit, then ensure to confirm the LOSCB bit is reset to "0" after that.**
- **When switching to the low-speed crystal oscillation clock or low-speed external clock, ensure to use the interrupt referring to the Section 6.3.5 "Switching the Low-speed Clock".**

6.2.7 Backup Clock Status Register (FBUSTAT)

FBUSTAT is a special function register (SFR) to indicate status of low-speed oscillation clock.
This register is unavailable for ML62Q1300 group.
The FBUSTAT is used only when choosing the low-speed crystal oscillation clock or the low-speed external clock input as LSCLK.
The FBUSTAT is initialized by only Power-On-Reset.

Address: 0xF00C (FBUSTAT)
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	-															
Byte	-								FBUSTAT							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LOSCS
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	LOSCS	This bit indicates the status of low-speed crystal oscillation clock. The bit is set to "1" when LOSCM1-0 bits of Low-speed Clock Mode Register (FLMOD) is set to 0x01 or 0x03. The bit automatically changed from "1" to "0" after the low-speed crystal oscillation circuit is stabilized and the clock backup interrupt request bit (CBUINT) of interrupt request register 23 (IRQ23) is set. LOSCS changes from "0" to "1" on the following conditions. 1. When the CPU enters STOP/STOP-D mode When releasing the STOP/STOP-D mode, it automatically changed from "1" to "0" after the low-speed crystal oscillation circuit is stabilized and the clock backup interrupt request bit (CBUINT) of interrupt request register 23 (IRQ23) is set. 2. When the low-speed crystal oscillation stopping is detected In this case, the clock backup interrupt request bit (CBUINT) of interrupt request register 23 (IRQ23) is set. After that, if it came back to the normal operation and then the stabilization time is counted out , the LOSCS changed from "1" to "0" and the clock backup interrupt request bit (CBUINT) of interrupt request register 23 (IRQ23) is set. 0: Low-speed crystal clock is stable (Initial value) 1: Low-speed crystal clock is stopped or the CPU is in the STOP/STOP-D mode

[Note]

- **In case the LOSCS bit gets to "1" after the LOSCB bit is set to "1", immediately set the mode back to "Low-speed RC oscillation clock" by resetting the LOSCM1-0 of FLMOD register to "00" and handle it appropriately for the application.**
- **Refer to the Section 6.3.5 "Switching the Low-speed Clock" to control the LOSCS bit.**

6.2.8 Clock Backup Test Mode Acceptor (FBTACP)

FBTACP is a special function register (SFR) to enable writing to Clock Backup Test Mode register (FBTCON). This register is unavailable for ML62Q1300 group.

Address: 0xF0C4 (FBTACP)
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	-															
Byte	-								FBTACP							
Bit	-	-	-	-	-	-	-	-	OSCBACP7	OSCBACP6	OSCBACP5	OSCBACP4	OSCBACP3	OSCBACP2	OSCBACP1	OSCBACP0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	OSCBACP7 to OSCBACP0	These bits are used to prevent erroneous writing to the FBTCON register. When "0xFA" and "0xF5" are written to the FBTACP register in this order, writing to the FBTCON is allowed only once. It requires writing "0xFA" and "0xF5" in this order every time to enable the continuous writes to the FBTCON. Any other instructions can be executed between the instruction that writes "0xFA" to STPACP and the instruction that writes "0xF5". However, if write data other than "0xF5" after writing "0xFA", the procedure gets invalid, so needs write "0xFA" again.

6.2.9 Clock Backup Test Mode (FBTCON)

FBTCON is a special function register (SFR) to control the clock backup test mode.
This register is unavailable for ML62Q1300 group.
The clock backup test mode can make purposely the condition that stops the low-speed crystal oscillation.

Address: 0xF0C6 (FBTCON)
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	-															
Byte	-								FBTCON							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LOSCL	LOSCT
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1	LOSCL	When the LOSCT bit is set to "1", the LOSCL bit determines the fixed level of low-speed crystal oscillation clock. 0: Low-speed crystal oscillation clock is fixed to "L" level (Initial) 1: Low-speed crystal oscillation clock is fixed to "H" level
0	LOSCT	This bit enables the clock backup test mode. 0: Normal mode (Initial value) 1: Clock backup test mode

[Note]

- **Use the clock backup test mode after setting the low-speed crystal oscillation clock mode.**

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

LRCADJ is a special function register (SFR) to adjust the frequency of the low-speed RC oscillation clock.

Address: 0xF080 (LRCADJ)
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	-															
Byte	-								LRCADJ							
Bit	-	-	-	-	-	-	-	-	-	-	LRCADJ5	LRCADJ4	LRCADJ3	LRCADJ2	LRCADJ1	LRCADJ0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

As the frequency of the low-speed RC oscillation circuit varies depending on temperature, the frequency is adjusted by using software adjustment[*1]. LRCADJ is set to the correction trimming value which is calculated using following three parameters and primary approximate equation derived between low temperature and ordinary temperature or between ordinary temperature and high temperature.

[Parameter]
1. Temperature sensor A/D conversion value of the current temperature
2. Temperature sensor A/D conversion value of the ordinary temperature (25 °C)
3. Coefficient: proportional to the frequency change with temperature

[*1]: Use the RC oscillation adjustment sample software provided by LAPIS Technology.

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

6.3 Description of Operation

6.3.1 Low-Speed Clock

For the low-speed clock generation circuit, one of the following modes can be chosen through LOSCM1 and LOSCM0 bits in the FLMOD register: *1

- Low-speed RC oscillation mode
- Low-speed crystal oscillation mode
- External clock mode

Figure 6-2 shows the low-speed clock generation circuit configuration.

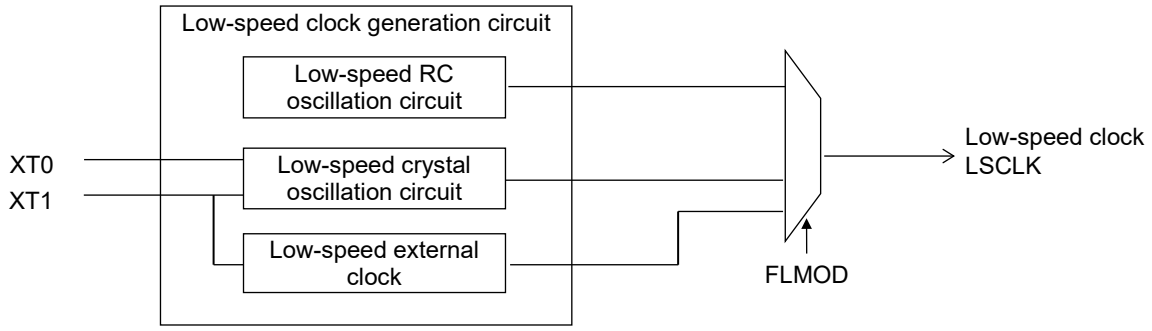


Figure 6-2 Configuration of Low-speed Clock Generation Circuit

*1 : Only Low-speed RC oscillation mode is available for ML62Q1300 group.

6.3.1.1 Configuration of Low-Speed RC Oscillation Circuit

The low-speed RC oscillation clock is chosen for the system clock at the power on.
At the power on, the low-speed clock (LSCLK) is output and the CPU runs a program after 512 counts of the low-speed RC oscillation clock. When the STOP/STOP-D mode is released, the low-speed clock (LSCLK) is output and the CPU runs a program after 10 counts of the low-speed RC oscillation clock.
The frequency of the low-speed RC oscillation circuit can be adjusted with the low-speed RC oscillation frequency adjustment register (LRCADJ). To adjust the frequency, use the sample software "ML62Q1000 Series RC oscillation adjustment sample software" provided by LAPIS Technology.
Figure 6-3 shows the configuration of the low-speed RC oscillation circuit.
Figure 6-4 shows the operation waveforms at the start of the low-speed RC oscillation circuit and in the STOP/STOP-D mode.

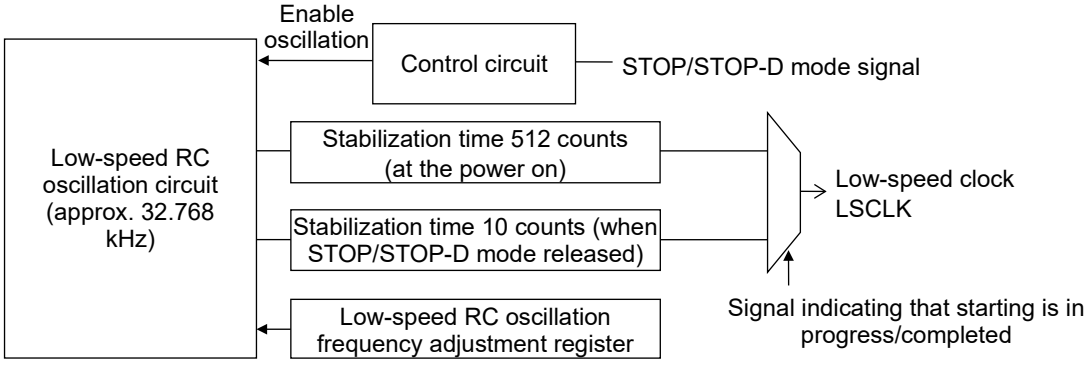


Figure 6-3 Configuration of Low-Speed RC Oscillation Circuit

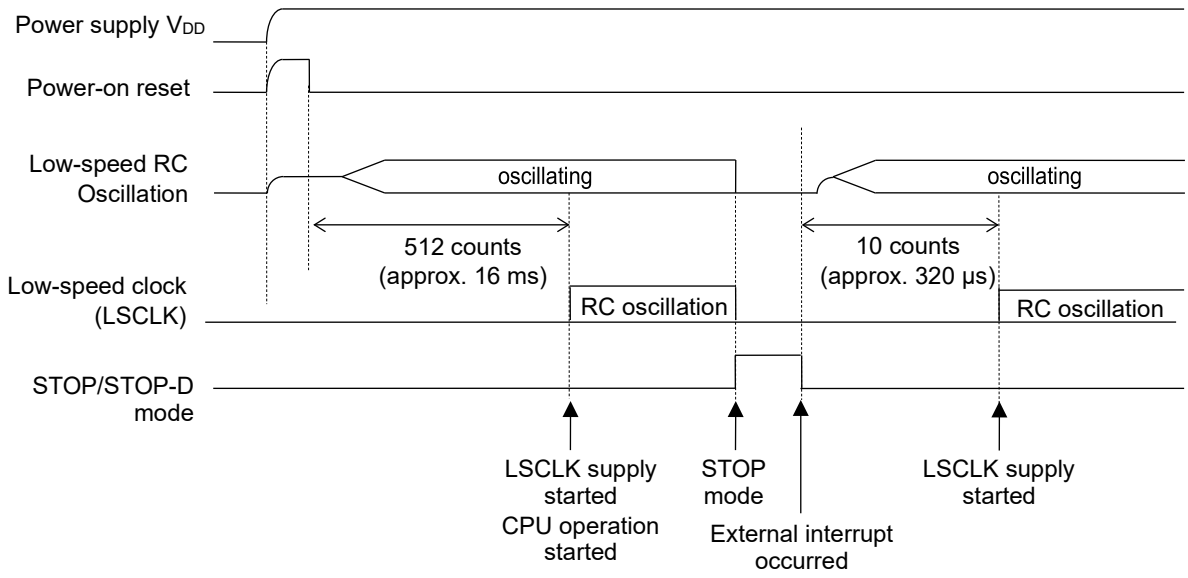


Figure 6-4 Low-speed Clock Operation Waveforms at Start of Low-speed RC Oscillation Circuit and in STOP/STOP-D Mode

6.3.1.2 Configuration of Low-Speed Crystal Oscillation Circuit

Figure 6-5 shows the configuration of the low-speed crystal oscillation circuit.
This circuit is unavailable for ML62Q1300 group.
The 32.768 kHz crystal oscillator can be chosen in the low-speed clock mode register (FLMOD).
When choosing the low-speed crystal oscillation clock for the low-speed clock, the setting of PORTXT mode register 01 (PXTMOD01) is ignored. This is one of safety functions to prevent the low-speed clock supply from stopping.
When choosing a low-speed crystal oscillation circuit for the low-speed clock, the backup function is activated.
The backup function is the function that always monitors the low-speed crystal oscillation. If an oscillation stop is detected, it switches the low-speed clock to the low-speed RC oscillation clock.
The low-speed crystal oscillation circuit stops its operation if entering the STOP/STOP-D mode.
The state on the LSI transfers to the backup mode in the following three cases:
- The low-speed crystal oscillation circuit is chosen for the low-speed clock through the FLMOD register.
- The STOP/STOP-D mode has been released.
- When, in the state that the crystal oscillation clock was chosen by the low-speed clock, the crystal oscillation stops is detected.
  This detection occurs when the crystal oscillation clock is stopped, or when the oscillation frequency is lowering abnormally. When the oscillation clock stop continues, it takes about 8ms(typ.) to detect. During the oscillation clock is stopping or the clock frequency is lowering, the high-speed clock frequency is also lowering.
  When the detection occurs, execute appropriate processing for each application.

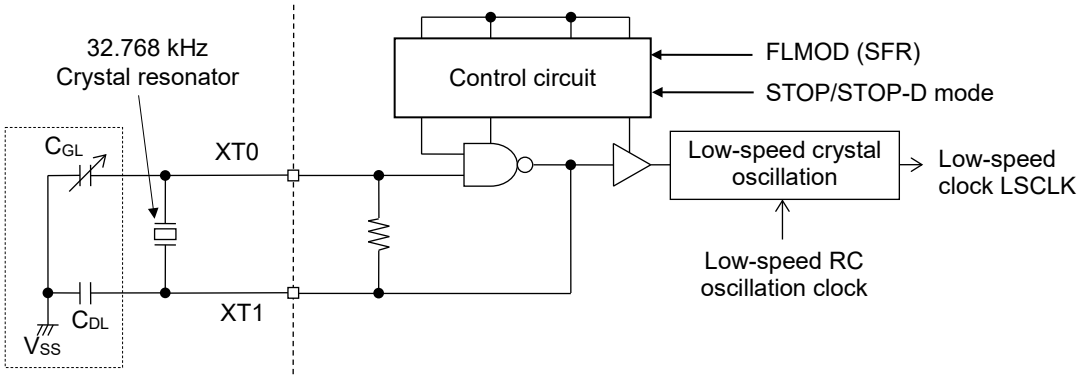


Figure 6-5 Circuit Configuration of Low-speed Crystal Oscillation Mode

[Note]
- **Place the crystal resonator as close to the LSI as possible and make sure that signals causing noise and power supply wiring are not near the crystal and its wiring.**
- **Note that oscillation may stop due to condensation.**
- **When switching to the low speed crystal oscillation clock, ensure to use the interrupt referring to the Section 6.3.5 "Switching the Low-speed Clock".**
- **When using the low-speed crystal oscillation clock and choosing the high-speed clock for the system clock, switch the system clock to the low-speed clock before entering the STOP/STOP-D mode.**

Figure 6-6 describes backup mode waveforms at the start of the low-speed crystal clock and in the STOP/STOP-D mode. The low-speed crystal oscillation circuit operates when choosing it through the low-speed clock mode register (FLMOD) following the start of low-speed RC oscillation circuit operation after the power supply is turned on. Since the FLMOD register is only initialized by the POR (power-on reset), the oscillation will continue even if a reset other than POR occurs after choosing the low-speed crystal.

When choosing the low-speed crystal oscillation circuit through the FLMOD register, The operating state on the LSI transfers to the backup mode and the internal low-speed RC oscillation clock is supplied until the low-speed crystal oscillation gets stable.
After waiting for the low-speed crystal oscillation start time (T_{XTL}) and low-speed crystal oscillation stabilization time (8192 counts), the clock backup interrupt (CBUINT) occurs if the low-speed crystal oscillation circuit properly started up. At this point, release the backup mode. Once the backup mode is released, the crystal oscillation clock is supplied to the low-speed clock (LSCLK).

In addition, the low-speed crystal oscillation circuit stops oscillation when entering the STOP/STOP-D mode. When the STOP/STOP-D mode is released by external interrupts and etc., it resumes the oscillation, and the operating state transfers to the backup mode. Release the backup mode following the aforementioned same procedure. See Chapter 4 "Power Management" for the STOP/STOP-D mode.
See the data sheet for the low-speed oscillation start time (T_{XTL}).

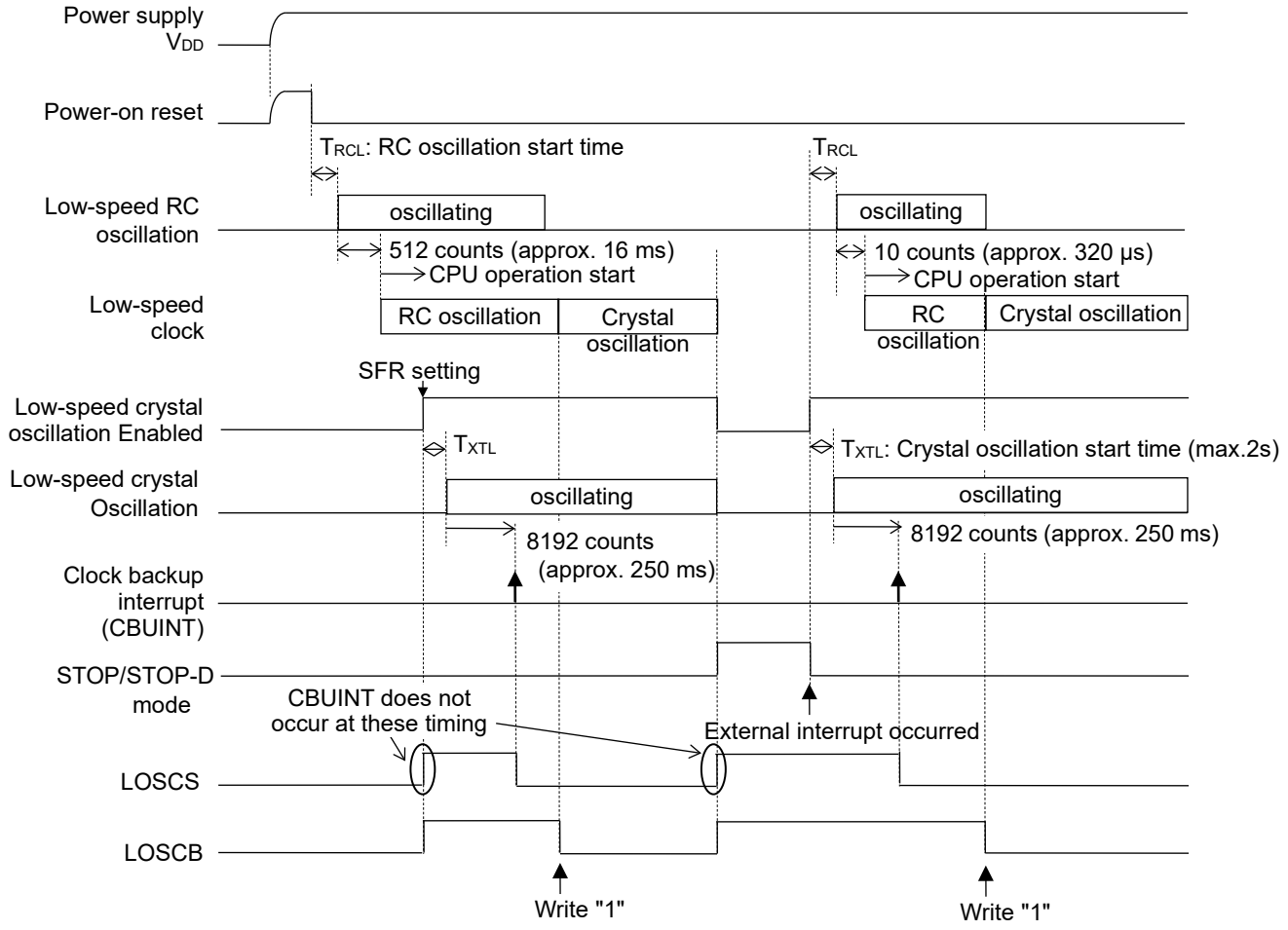


Figure 6-6 Low-speed Crystal Oscillation Circuit Operation
(When Low-speed Crystal Starting Up, in STOP/STOP-D Mode)

Figure 6-7 describes operation waveforms in the backup mode after the start of the low-speed crystal oscillation circuit. When the crystal oscillation clock stops after the low-speed crystal oscillation started, it shifts to the backup mode about 8ms(typ.) later.
The clock backup interrupt (CBUINT) occurs after
(1) shifted to the backup mode
(2) restarted the crystal oscillation

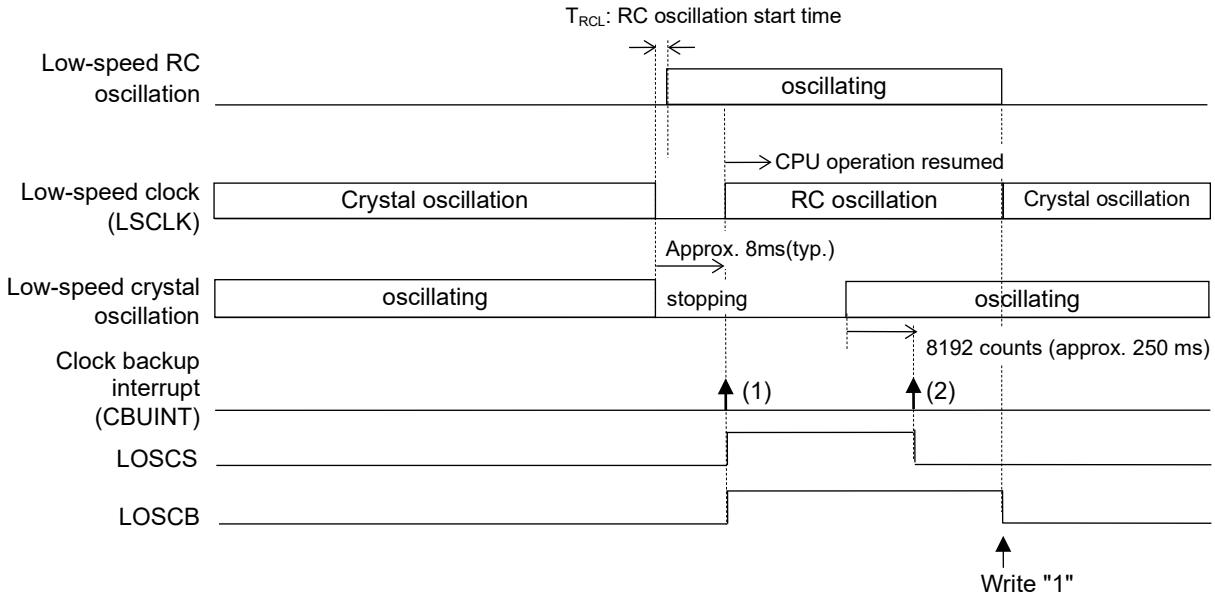


Figure 6-7 Low-speed Crystal Oscillation Circuit Operation (Backup Mode)

Figure 6-8 shows the operation waveforms when a reset occurs after the start of the low-speed crystal oscillation circuit.
The low-speed crystal oscillation circuit is not reset by anything but the power-on reset.
See Chapter 3 "Reset Function" for details of resetting.

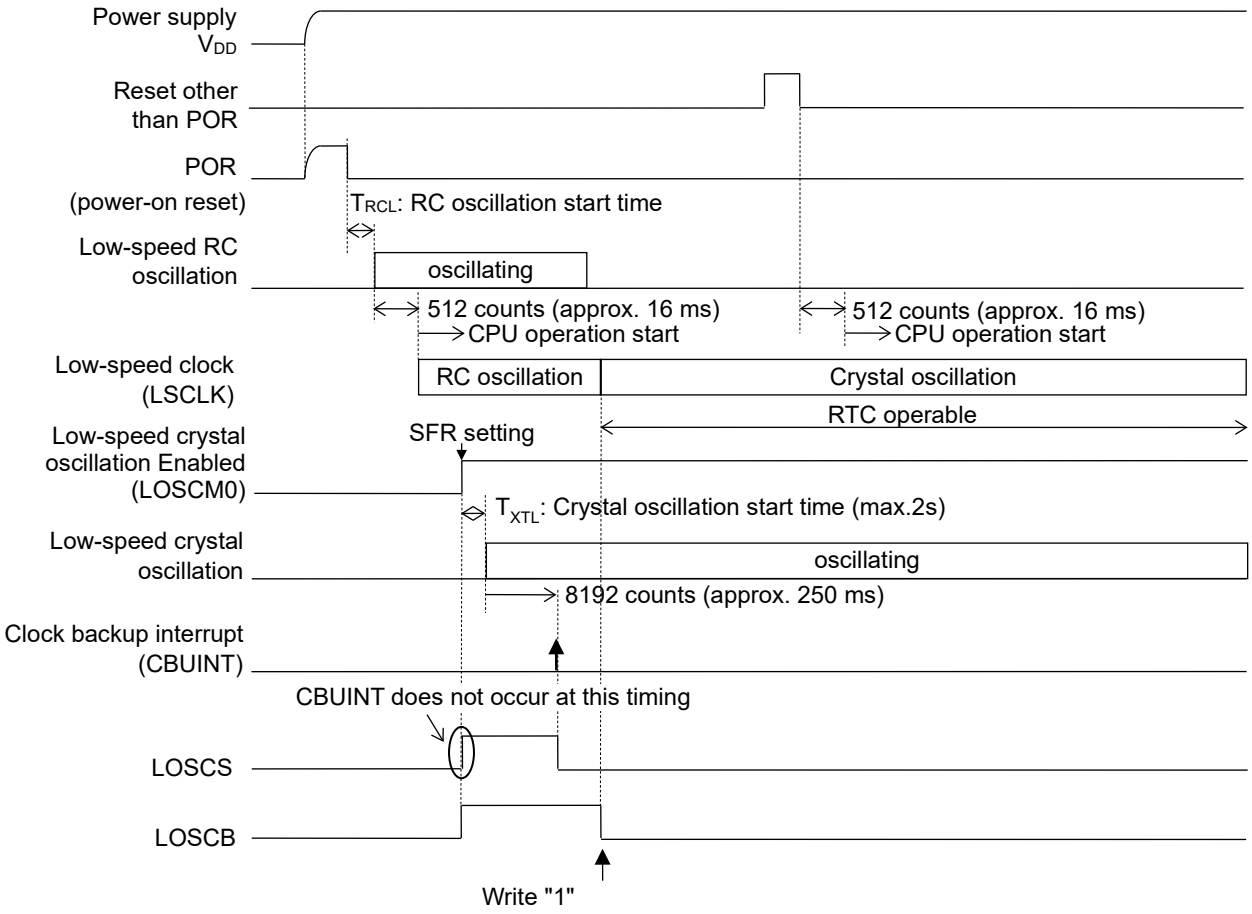


Figure 6-8 Low-speed Crystal Clock Generation Circuit Operation (Reset Operation After Start of Low-speed Crystal)

6.3.1.3 Low-speed External Clock Input

The clock can be input to the XT1 pin for the low-speed crystal oscillation circuit.
This circuit is unavailable for ML62Q1300 group.
A low-speed external clock can be chosen in the low-speed clock mode register (FLMOD).
When choosing the low-speed external clock for a low-speed clock, the setting of PXT1IE bit of PXTMOD01 register is ignored.
The low-speed external clock operates in the same manner as when the low-speed crystal oscillation circuit is chosen.

6.3.2 High-Speed Clock

The high-speed clock is generated by multiplying the low-speed clock (LSCLK) by the high-speed oscillation circuit (PLL oscillation circuit).
The oscillation frequency of PLL can be chosen from 24 MHz or 16 MHz through the code option.
It is also possible to output the high-speed clock (OUTHSCLK) of which frequency is divided from a general-purpose port.

6.3.2.1 Configuration of PLL Oscillation Circuit

The PLL oscillation circuit generates the PLL oscillation clock by multiplying the LSCLK. Multiplying by 488 is 16MHz of PLL basic frequency, and multiplying by 732 is 24MHz of PLL basic frequency.
After high-speed clock oscillation is enabled (through setting ENOSC of the FCON register to "1"), HSCLK (high-speed clock) is output by continuing count operation until the PLL oscillation clock is stabilized.
In addition, the PLL oscillation circuit stops oscillation when entering the HALT-H/STOP/STOP-D mode by the software.

For the PLL oscillation stabilization time, two choices are available by setting the FHWUPT register: whether starting the clock supply after it gets stabilized frequency, or starting the clock supply before it gets the stabilized frequency. When set to start the clock supply before getting the stabilized frequency, the clock supply is started approximately 30 μs after the high-speed clock oscillation is enabled. The clock frequency reaches to the target approximately 2 ms after the high-speed clock oscillation is enabled. Although the frequency within the 2ms is not guaranteed, it can be used for the system clock.

See Table 4-5 for the time for restoration from the STOP/STOP-D mode.
See Chapter 4 "Power Management" for details of the STOP/STOP-D mode.
Figure 6-9 shows the configuration of the PLL oscillation circuit.
Figure 6-10 shows the operation waveforms at the start of the PLL oscillation circuit and in the STOP/STOP-D mode.

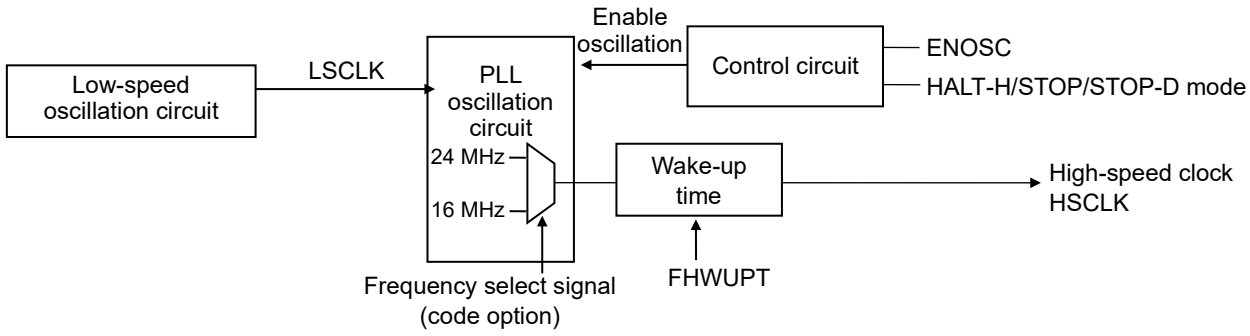
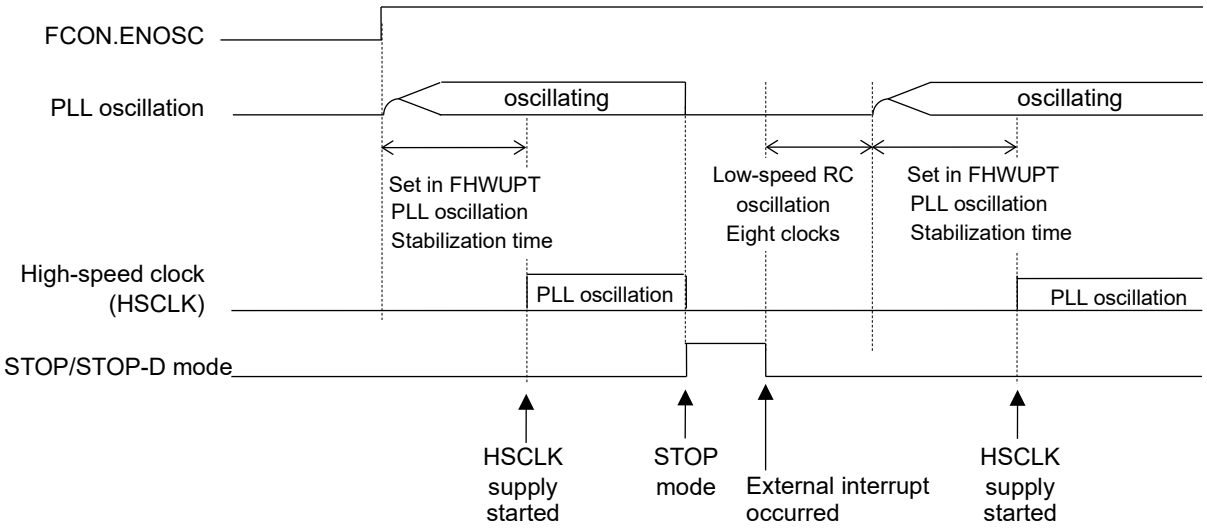


Figure 6-9 PLL Oscillation Circuit Configuration



FCON : Frequency control register
FHWUPT : High-speed clock wake-up time setting register

Figure 6-10 High-speed Clock Operation Waveforms at Start of PLL Oscillation Circuit and in STOP/STOP-D Mode

6.3.3 WDT Clock

For the WDT clock (WDTCLK), the clock with divided frequency of low-speed clock (1.024 kHz) or the WDT dedicated RC1K oscillation clock (approx. 1 kHz) can be chosen by the code option. If the accuracy of the watch dog timer is required, choose the clock with divided frequency of the low-speed clock.
The WDT operation clock stops oscillation in the STOP/STOP-D mode.
After the STOP/STOP-D mode is released, supply of the WDT clock will be started in the stabilization time (one count of the WDT RC1K oscillation circuit clock).
See Chapter 26 "Code Option" for details on how to set the code option.
Figure 6-11 shows the configuration of the WDT RC oscillation circuit.
Figure 6-12 shows WDT clock operation waveforms at the start of the WDT RC oscillation circuit and in the STOP/STOP-D mode.

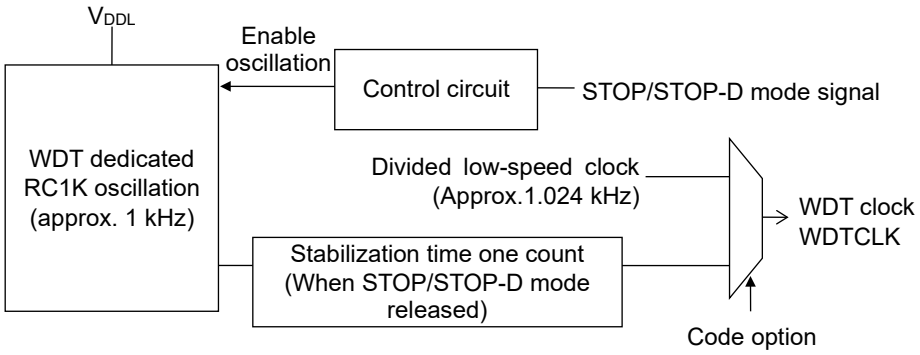


Figure 6-11 WDT RC Oscillation Circuit Configuration

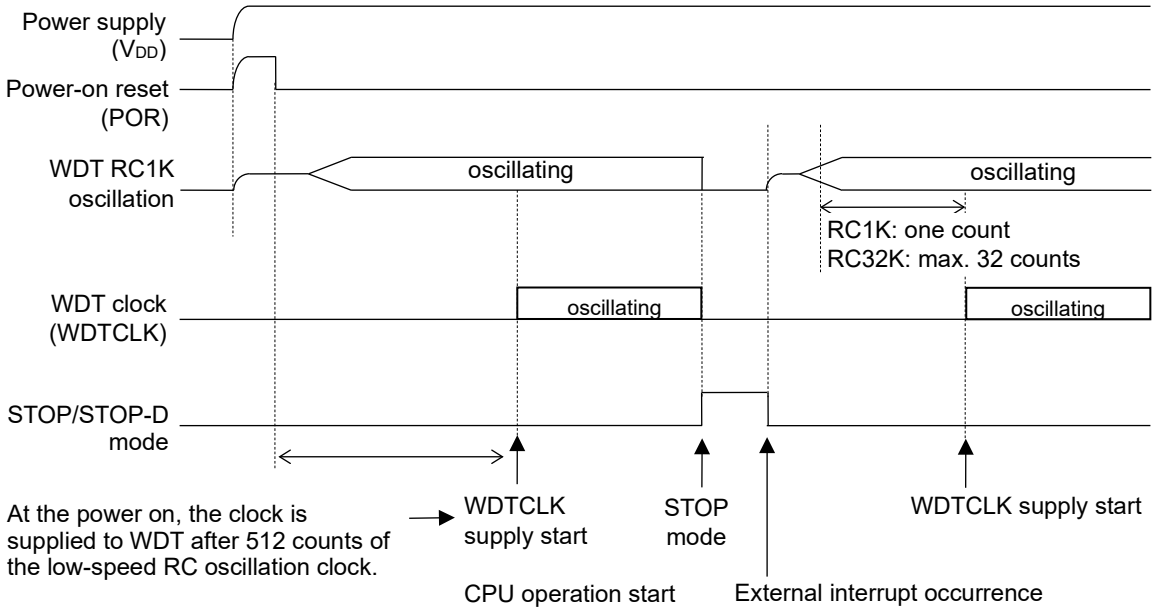


Figure 6-12 Operation Waveforms at Start of WDT RC Oscillation Circuit and in STOP/STOP-D Mode

6.3.4 Switching of System Clock

Figure 6-13 shows the flow chart of the system clock switching (LSCLK → HSCLK).

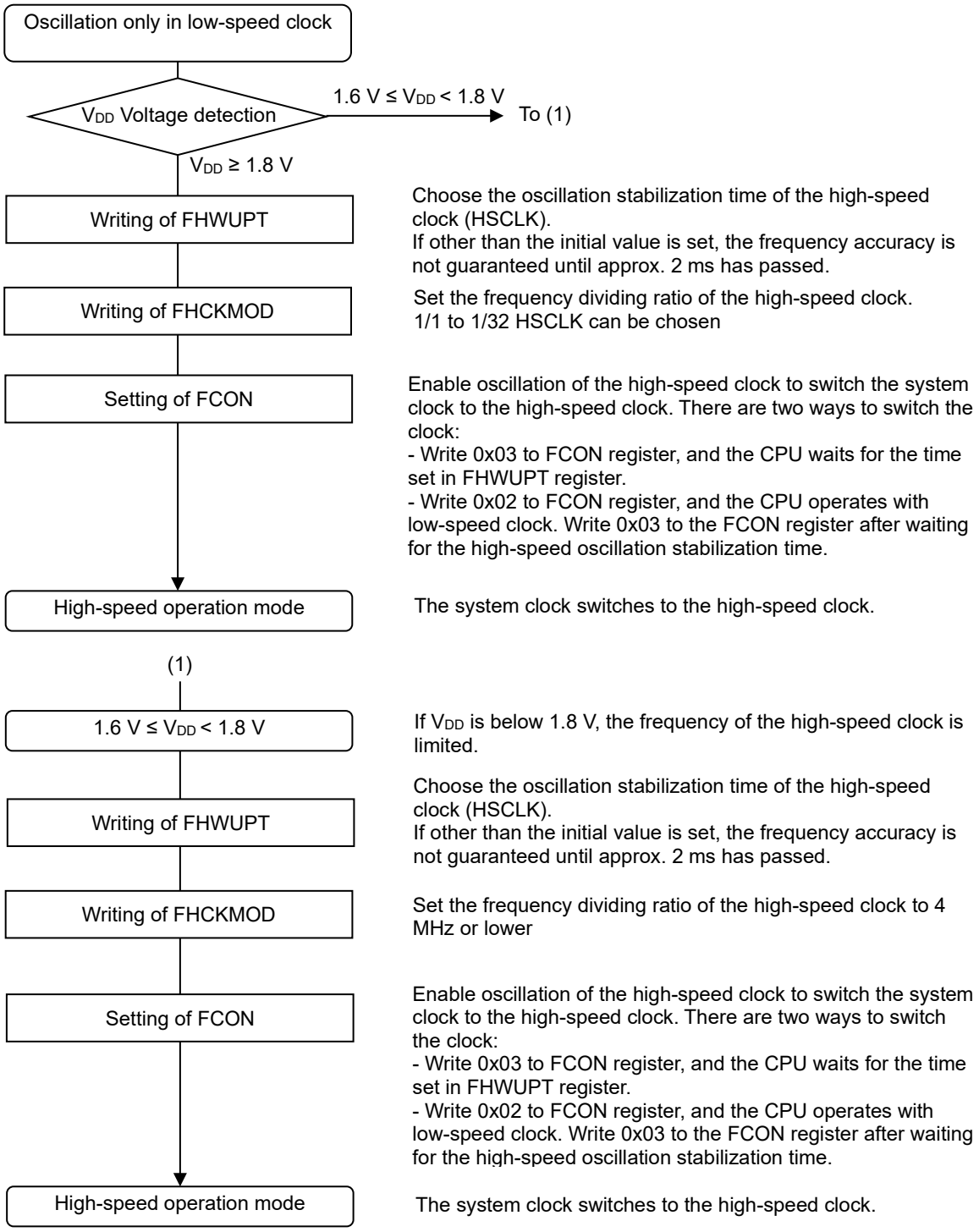


Figure 6-13 Flow Chart of System Clock Switching (LSCLK → HSCLK)

[Note]

- **When the voltage of V_{DD} is 1.6 V ≤ V_{DD} < 1.8 V, set the system clock (SYSTEMCLK) to 4 MHz or below. If it exceeds 4 MHz, the operation is not guaranteed.**

Figure 6-14 shows the flow chart of the system clock switching (HSCLK → LSCLK).

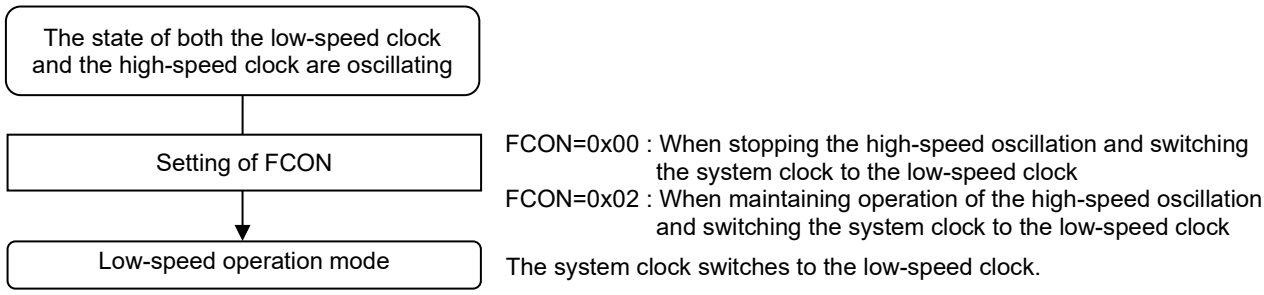


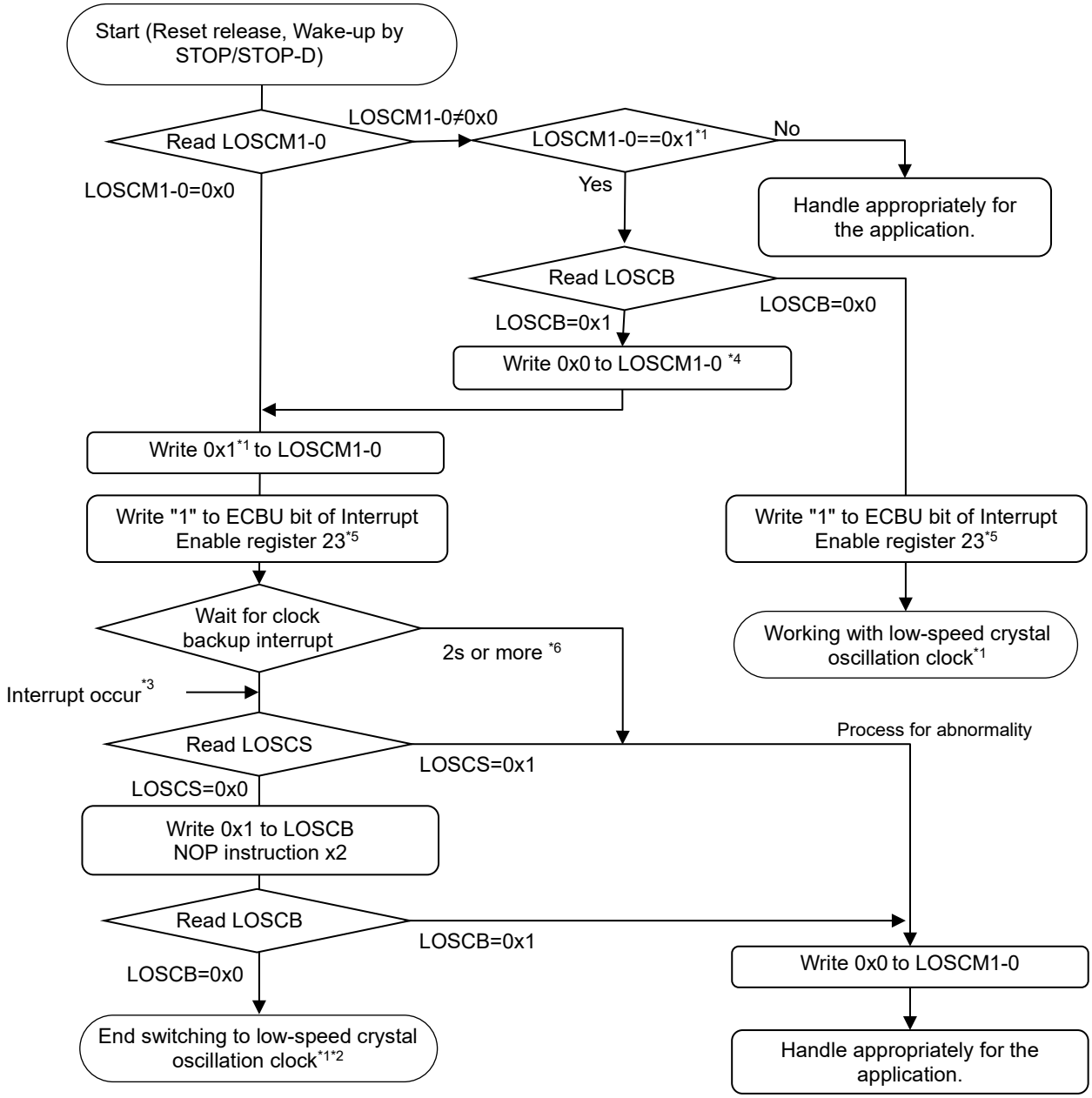
Figure 6-14 Flow Chart of System Clock Switching (HSCLK → LSCLK)

[Note]

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

6.3.5 Switching Low-Speed Clock

Figure 6-15 shows a flow chart of the low-speed clock switching process from the low-speed RC oscillation clock to the low-speed crystal oscillation clock or to the low-speed external clock. Follow the flow chart below to check the state of the low-speed clock after releasing the STOP/STOP-D mode. When switching the system clock to the high-speed clock, first switch the low-speed clock to the low-speed crystal oscillation circuit. This is unavailable for ML62Q1300 group.



*1: Read "0x1" with "0x3" and read "low-speed oscillation clock" with "low-speed external clock" when switching to the "low-speed external clock".
*2: Do not control the power managements (write to SBYCON register) and the clock managements (write to LOSCM1-0 of FLMOD register and write to SELSCLK of FCON register) before the switching to the low-speed crystal oscillation clock.
*3: There are four interrupt sources. After switching to the low-speed crystal oscillation clock, handle them following to the flow of the interrupt.
1) When counting for the clock stabilization is completed after setting LOSCM1-0 to "0x1".
2) When counting for the clock stabilization is completed after releasing STOP/STOP-D mode.
3) If an abnormal condition happened on the low-speed crystal circuit and the oscillation stopping is detected.
4) If an abnormal condition happened on the low-speed crystal circuit and returned to normal oscillating in stable.
*4: Clear the setting once because it's unknown whether the state is in backup clock or the state before setting the crystal oscillation clock.
*5: Enable the interrupt after ending the reset release process because it's unknown whether the state is in backup clock or the state before setting the crystal oscillation clock.
*6: Set the appropriate value according to the specifications and environment of the crystal resonator.

Figure 6-15 Flow Chart for switching the Low-speed clock
(RC oscillation clock to Crystal oscillation/External clock)

Chapter 7 Low Speed Time Base Counter

7. Low Speed Time Base Counter

7.1 General Description

The low speed time base counter enables following functions.

- Generate periodical interrupt requests
- Output periodical pulse signals to the general ports
- Adjust the frequency of simplified RTC clock (Available for ML62Q1500/ML62Q1800 and ML62Q1700 group)

7.1.1 Features

- Generate eight frequency (128Hz, 64Hz, 32Hz, 16Hz, 8Hz, 4Hz, 2Hz and 1Hz) of pulse signals by dividing the low-speed clock (LSCLK)
- Three interrupt requests can be chosen among eight periodical interrupt requests
- The 1Hz or 2Hz signal can be output from general ports
- An interrupt request (LTB0INT) can be used for a trigger event source of the Successive Approximation type A-D Converter.

ML62Q1500/ML62Q1800 and ML62Q1700 group only

- The clock frequency adjust function
  - Allows to adjust in a range approx.-488ppm to +488ppm with the resolution approx.0.119ppm.
  - Two confirmation methods with the low-speed clock or high-speed clock
- The 1Hz or 2Hz signal is used for the simplified RTC clock

7.1.2 Configuration

Figure 7-1 shows the configuration of the low speed time base counter on ML62Q1300 group.

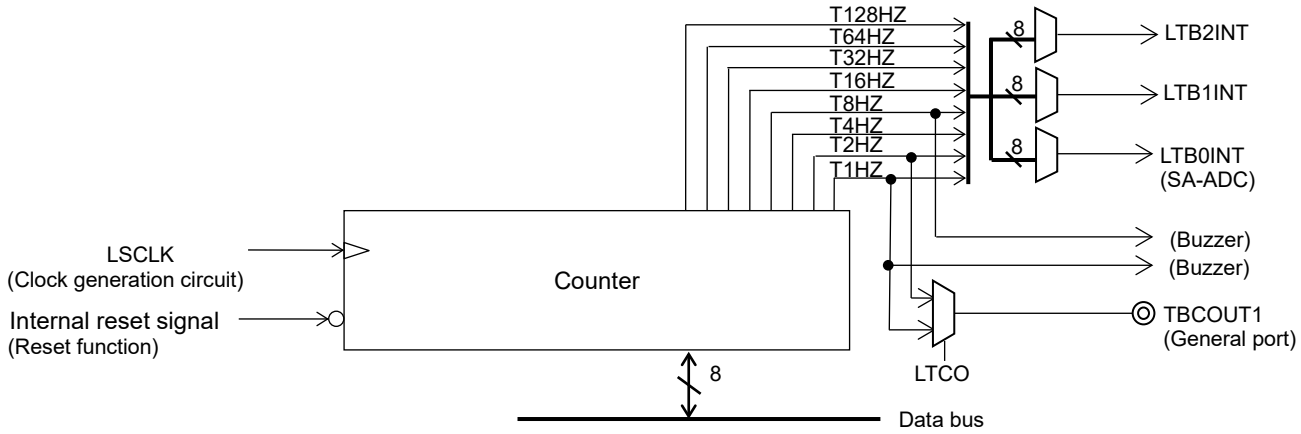
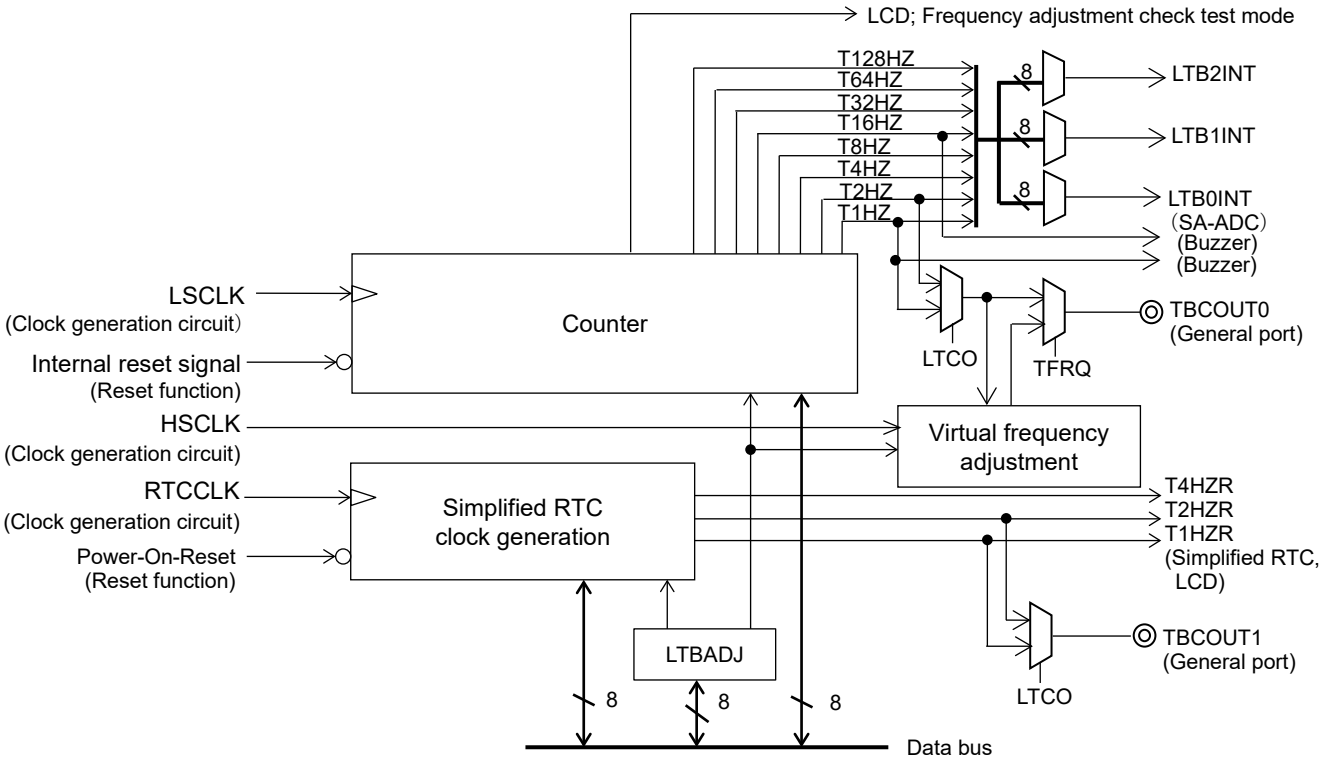


Figure 7-1 Configuration of Low Speed Time Base Counter (ML62Q1300 group)

T1HZ to T128HZ: Time base counter output signal
LTB2INT: Low speed time base counter 2 interrupt request
LTB1INT: Low speed time base counter 1 interrupt request
LTB0INT: Low speed time base counter 0 interrupt request

Figure 7-2 shows the configuration of the low speed time base counter on ML62Q1500/ML62Q1800 and ML62Q1700 group.



LTBADJ:	Low speed time base counter frequency adjustment register
T1HZ to T128HZ:	Time base counter output signal
LTBC2INT:	Low speed time base counter 2 interrupt request
LTBC1INT:	Low speed time base counter 1 interrupt request
LTBC0INT:	Low speed time base counter 0 interrupt request
T2HZR:	Simplified RTC 2Hz Clock
T1HZR to T4HZR:	Simplified RTC Timer Base Counter output signal

Figure 7-2 Configuration of Low Speed Time Base Counter (ML62Q1500/ML62Q1800 and ML62Q1700 group)

7.1.3 List of Pins

The output pins of the low speed time base counter are assigned to the shared function of genral purpose ports.

Signal name	I/O	Function
TBCOUT0	O	The virtual frequency adjustment output signal or the low speed time base counter output signal
TBCOUT1	O	ML62Q1300 group: The low speed time base counter output signal ML62Q1500/ML62Q1800/ML62Q1700 group: 1Hz/2Hz clock for the Simplified RTC

Table 7-1 shows the list of the output ports and the register setting.

Table 7-1 Low speed time base counter function port and the register setting

Pin name	Shared port		Setting register	Setting value	ML62Q1300 group				ML62Q1500/ML62Q1800 ML62Q1700 group				
					16pin product	20 pin product	24 pin product	32 pin product	48 pin product	52 pin product	64 pin product	80 pin product	100 pin product
TBCOUT0	P01	6th function	P0MOD1	0101_XXXX*1	–	–	–	–	●	●	●	●	●
	P17	6th function	P1MOD7	0101_XXXX*1	–	–	–	–	●	●	●	●	●
	P26	6th function	P2MOD6	0101_XXXX*1	–	–	–	–	●	●	●	●	●
	P31	6th function	P3MOD1	0101_XXXX*1	–	–	–	–	●	●	●	●	●
	P43	6th function	P4MOD3	0101_XXXX*1	–	–	–	–	●	●	●	●	●
TBCOUT1	P01	7th function	P0MOD1	0110_XXXX*1	–	–	–	–	●	●	●	●	●
	P20	6th function	P2MOD0	0101_XXXX*1	●	●	●	●	●	●	●	●	●
	P27	6th function	P2MOD7	0101_XXXX*1	●	●	●	●	●	●	●	●	●
	P31	7th function	P3MOD1	0110_XXXX*1	–	–	–	–	●	●	●	●	●
	P43	7th function	P4MOD3	0110_XXXX*1	–	–	–	–	●	●	●	●	●

●: Available - : Unavailable

*1 : XXXX determines the port output condition

XXXX	Port output condition
0010	CMOS output
1010	Nch open drain (without pull-up)
1111	Nch open drain (with pull-up)

7.2 Description of Registers

7.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF060	Low-speed Time Base Counter register	LTBR	-	R/W	8	0x00
0xF061	Reserved	-	-	-	-	-
0xF062	Low-speed Time Base Counter Control register	LTBCCON	-	R/W	8	0x01
0xF063	Reserved	-	-	-	-	-
0xF064	Simplified RTC Time Base Counter register	LTBRR	-	R	8	0x00
0xF065	Reserved	-	-	-	-	-
0xF066	Low-speed Time Base Counter Frequency Adjustment register	LTBADJL	LTBADJ	R/W	8	0x00
0xF067		LTBADJH		R/W	8	0x00
0xF068	Low-speed Time Base Counter Interrupt selection register	LTBINTL	LTBINT	R/W	8/16	0x30
0xF069		LTBINTH		R/W	8	0x06

7.2.2 Low Speed Time Base Counter Register (LTBR)

The low speed time base counter register (LTBR) is a special function register (SFR) to read the value of the low speed time base counter.
Writing any value to the LTBR, the all bits of T128HZ to T1Hz are initialized to "0".

Address: 0xF060 (LTBR)
Access: R/W
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								LTBR							
Bit	-	-	-	-	-	-	-	-	T1HZ	T2HZ	T4HZ	T8HZ	T16HZ	T32HZ	T64HZ	T128HZ
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

- **A time base counter interrupt may occur depending on the timing to write to the LTBR. See the program example for initializing described in Section 7.3.1 "Operation of the Low-speed Time Base Counter".**
- **T128HZ to T1HZ signals have "0" level in the first half cycle and "1" level in the second half cycle. 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 "0" 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 Figure 7-5 "Low speed time base counter interrupt timing and reset timing of reset by writing to LTBR" for details of the T128Hz to T1Hz waveform.**

7.2.3 Low Speed Time Base Register Control Register (LTBCCON)

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

Address: 0xF062 (LTBCCON)
Access: R/W
Access size: 8 bit
Initial value: 0x01

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								LTBCCON							
Bit	-	-	-	-	-	-	-	-	TFRQ	LTCO	-	-	-	-	-	TBRUN
R/W	R	R	R	R	R	R	R	R	R/W	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	1

Bit No.	Bit symbol name	Description
7	TFRQ	This bit is used to choose the frequency adjustment mode (Available for ML62Q1500/ML62Q1800 and ML62Q1700 group). 0: General frequency adjustment mode (Initial value) 1: Virtual frequency adjustment mode
6	LTCO	This bit is used to choose the signal that is output from TBCOUT0 pin and TBCOUT1 pin. 0: T1HZ/T1HZR (Initial value) 1: T2HZ/T2HZR
5 to 1	-	Reserved bit
0	TBRUN	This bit is used to control the start and stop the time base counter. 0: Stop the time base counter 1: Start the time base counter/ The time base counter is in operating (Initial value)

[Note]

- **Enable the high-speed clock (HSCLK) when using the virtual frequency adjustment mode.**
- **It takes max. two cycles of low-speed clock (LSCLK) from writing to the TBRUN bit to when the operation starts or stops.**
- **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. (except ML62Q1300 group)**

7.2.4 Simplified RTC Time Base Counter Register (LTBRR)

The low speed time base counter register for Simplified RTC(LTBRR) is a special function register (SFR) to read the counter value for the Simplified RTC.
The all bits of T128HZ to T1Hz are initialized to "0" by writing to the SRTCMAS register, or by the Power-On-Reset.
This register is available for ML62Q1500/ML62Q1800 and ML62Q1700 group.

Address: 0xF064 (LTBRR)
Access: R
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								LTBRR							
Bit	-	-	-	-	-	-	-	-	T1HZR	T2HZR	T4HZR	T8HZR	T16HZR	T32HZR	T64HZR	T128HZR
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]
- **Do not write to the LTBRR register. It is read only.**
- **Read the LTBRR register twice to verify the data to prevent reading uncertain data while counting-up.**

7.2.5 Low Speed Time Base Counter Frequency Adjustment Register (LTBADJ)

Time base counter frequency adjustment register (LTBADJ) is a special function register (SFR) to set adjustment value for the frequency of time base clock.
The LTBADJ register is initialized only by the Power-On-Reset.
This register is available for ML62Q1500/ML62Q1800 and ML62Q1700 group.

Address: 0xF066 (LTBADJL/LTBADJ), 0xF067 (LTBADJH)
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	LTBADJ															
Byte	LTBADJH								LTBADJL							
Bit	-	-	-	LADJS	LADJ11	LADJ10	LADJ9	LADJ8	LADJ7	LADJ6	LADJ5	LADJ4	LADJ3	LADJ2	LADJ1	LADJ0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 13	-	Reserved bits
12	LADJS	This bit is used to choose the sign of frequency adjustment value. 0: the value in LADJ11-0 is a positive number (Initial value) 1: the value in LADJ11-0 is a negative number
11 to 0	LADJ11 to LADJ0	These bits are used to specify the frequency adjustment ratio. See 7.3.2 "Time Base Counter Frequency Adjustment Function" for the relation of the setting data and the adjustable ppm.

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

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

Address: 0xF068 (LTBINTL/LTBINT), 0xF069 (LTBINTH)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0630

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

Bit No.	Bit symbol name	Description
15 to 11	-	Reserved bits
10 to 8	LTI2S2 to LTI2S0	These bits are used to choose the signal to be assigned to the time base counter interrupt 2 (LTB2INT). 000: T128HZ 001: T64HZ 010: T32HZ 011: T16HZ 100: T8HZ 101: T4HZ 110: T2HZ (Initial value) 111: T1HZ
7	-	Reserved bit
6 to 4	LTI1S2 to LTI1S0	These bits are used to choose the signal to be assigned to the time base counter interrupt 1 (LTB1INT). 000: T128HZ 001: T64HZ 010: T32HZ 011: T16HZ (Initial value) 100: T8HZ 101: T4HZ 110: T2HZ 111: T1HZ
3	-	Reserved bit
2 to 0	LTI0S2 to LTI0S0	These bits are used to choose the signal to be assigned to the time base counter interrupt 0 (LTB0INT). 000: T128HZ (Initial value) 001: T64HZ 010: T32HZ 011: T16HZ 100: T8HZ 101: T4HZ 110: T2HZ 111: 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 Low Speed Time Base Counter Operation

The low-speed time base counter (LTBC) starts counting up from 0x0000 at the falling edge of the low-speed clock after releasing the system reset, then generates T128HZ to T1HZ signals. Three factors can be chosen from T128HZ to T1HZ signals to generate periodical low-speed time base counter interrupt requests.
Values of T128HZ to T1HZ signals can be read from the LTBR register.
The low-speed time base counter interrupt request is generated at the falling edge of a signal chosen in the LTBINT register.

When changing the assignment of interrupt signals in the LTBINT register, low-speed time base counter interrupt requests (LTBCnINT) may be generated depending on write timing to the register. Therefore, change the value in the LTBINT register with the interrupt disabled in the IE67 register before changing the assignment of interrupt signals, and clear the generated low-speed time base counter interrupt request bit (QLTBCn) to "0". (n = 0 to 2)

Figure 7-3 shows a sample program for changing the assignment of low-speed time base counter signals.

```
ELTBC0 = 0;          // Disable LTBC0 interrupt
ELTBC1 = 0;          // Disable LTBC1 interrupt
ELTBC2 = 0;          // Disable LTBC2 interrupt
LTBINT = 0x0741;     // Change assignment of interrupt signal
__asm("NOP");        // Waiting time
QLTBC0 = 0;          // Clear QLTBC0
QLTBC1 = 0;          // Clear QLTBC1
QLTBC2 = 0;          // Clear QLTBC2
ELTBC0 = 1;          // Enable LTBC0 interrupt
ELTBC1 = 1;          // Enable LTBC1 interrupt
ELTBC2 = 1;          // Enable LTBC2 interrupt
```

Figure 7-3 Sample Program for Changing Assignment of Low-speed Time Base Counter Signals

The time equivalent to one clock of the system clock is required for the low-speed time base counter interrupt request bit (QLTBCn bit of IRQ67 register, n=0 to 2) to become "1" after changing the LTBINT register . Therefore, place one NOP instruction after changing the LTBINT register.

When writing arbitrary data to the LTBR register, T128HZ to T1HZ signals of the LTBR register are all initialized to "0". Depending on timing to write to the LTBR register, the signal assigned to the LTBINT register may change from "1" to "0". Also a low-speed time base counter interrupt request may occur. Therefore, with the low-speed time base counter interrupt disabled in the IE67 register, following writing to the LTBR register, clear the generated low-speed time base counter interrupt request bit (the QLTBCn bit of the IRQ67 register) to "0". (n = 0 to 2)

Figure 7-4 shows a sample program for initializing the LTBR register.

```
__DI();              // Disable interrupt (MIE=0)
LTBR = 0x00;         // Reset LTBR
__asm("NOP");        // Waiting time
QLTBC0 = 0;          // Clear QLTBC0
QLTBC1 = 0;          // Clear QLTBC1
QLTBC2 = 0;          // Clear QLTBC2
__EI();              // Enable interrupt (MIE=1)
```

Figure 7-4 Sample Program for Initializing LTBR Register

It takes one cycle of the system clock for QLTBCn to become "1" from writing to the LTBR register. Therefore, place one NOP instruction after writing to the LTBR register.

Figure 7-5 shows the low-speed time base counter interrupt request generation timing when choosing T128HZ, T16HZ, and T2HZ as interrupt factors in the LTBINT register, and shows the reset timing by writing to LTBR.

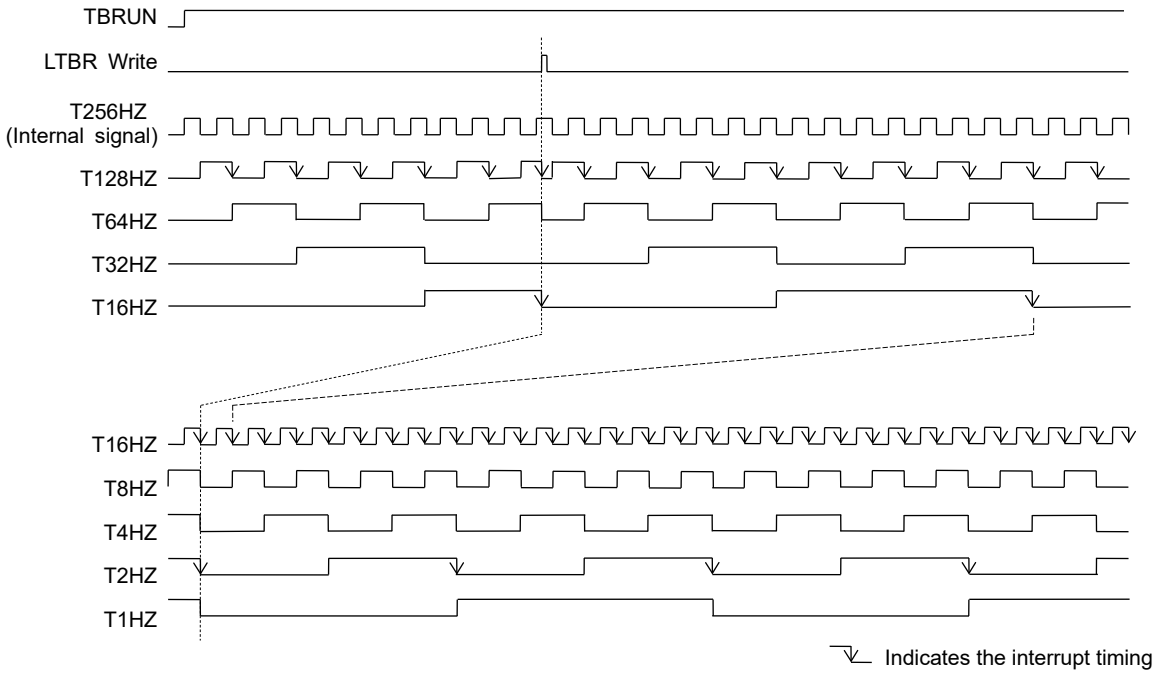


Figure 7-5 Low-speed Time Base Counter Interrupt Timing and Reset Timing by Writing to LTBR Register

[Note]

- **After writing to the LTBR register, the time by which the first low-speed time base counter interrupt request is generated is not guaranteed. If measuring the time using the low-speed time base counter interrupt, do so with reference to the interrupt generation interval.**
- **The time equivalent to max. one cycle of the system clock is required to reset the counter after writing to the LTBR register.**

7.3.2 Low Speed Time Base Counter Frequency Adjustment Function

For T128HZ to T1HZ, T128HZR and T1HZR of the low-speed time base counter, the frequency can be adjusted using the low-speed time base counter frequency adjustment register (LTBADJ). (Available for ML62Q1500/ML62Q1800 and ML62Q1700 group)
Measure the signal output from the TBCOUT0 or TBCOUT1 pin, then adjust the frequency using the LTBADJ register. The adjustment range and resolution are as follows:

- Adjustment range : Approx. -488 ppm to +488 ppm
- Adjustment resolution : Approx. 0.119 ppm

The following two modes are available to confirm the adjusted frequency:

Frequency adjustment mode	Description
Normal frequency adjustment mode	This mode is used to confirm that 256 seconds includes exactly 256 cycles/512 cycles of T1HZR/T2HZR,which is output form pin as TBCOUT1 under operating with actual adjusted low-speed clock.
Virtual frequency adjustment mode	This mode is used to confirm that 1 second/0.5 seconds include exactly 1 cycle of T1HZ/T2HZ,which is output form pin as TBCOUT0 under operating with high-speed clock. There is a frequency error rate of several ppm compared to the normal frequency adjustment mode.

Table 7-2 Frequency adjustment value set in the LTBADJ and Adjustment ratio

LADJS	LADJ11 to 0												Hexadecimal	Frequency adjustment ratio (ppm)
0	1	1	1	1	1	1	1	1	1	1	1	1	0FFFH	+488.162
0	1	1	1	1	1	1	1	1	1	1	1	0	0FFEH	+488.043
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
0	0	0	0	0	0	0	0	0	0	0	1	1	0003H	+0.358
0	0	0	0	0	0	0	0	0	0	0	1	0	0002H	+0.238
0	0	0	0	0	0	0	0	0	0	0	0	1	0001H	+0.119
0	0	0	0	0	0	0	0	0	0	0	0	0	0000H	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1000H	0
1	0	0	0	0	0	0	0	0	0	0	0	1	1001H	–0.119
1	0	0	0	0	0	0	0	0	0	0	1	0	1002H	–0.238
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	1	1	1	1	1	0	1FFEH	–488.043
1	1	1	1	1	1	1	1	1	1	1	1	1	1FFFH	–488.162

The correction values (LADJS, LADJ11 to LADJ0) set in the LTBADJ register can be calculated using the following formula.

Correction value = Frequency adjustment ratio x 8388608 (decimal)
= Frequency adjustment ratio x 800000H (hexadecimal)

Example 1: When adjusting +15.0 ppm (when the clock loses)

Correction value = +15.0 ppm x 8388608 (decimal)
= $+15.0 \times 10^{-6} \times 8388608$
= +125.82912 (decimal)
≈ 7EH (hexadecimal)
LTBADJ = 7EH (hexadecimal)

Example 2: When adjusting -25.5 ppm (when the clock gains)

Correction value = 25.5 ppm x 8388608 (decimal)
= $25.5 \times 10^{-6} \times 8388608$
= 213.909504 (decimal)
≈ D6H (hexadecimal)
When setting LTBADJ, add a sign bit.
LTBADJ = 10D6H (hexadecimal)

[Note]

- The frequency adjustment accuracy does not guarantee the accuracy including the frequency variation of the low-speed oscillation (32.768 kHz) due to temperature variations.

7.3.3 The way of monitoring the frequency on LCD drive outputs

The adjusted frequency by the LTBADJ register can be confirmed on the output waveforms of LCD driver. (Available for ML62Q1700 group)
Set “FRM2 to FRM0” bits of display mode register (DSPMOD) to “110” or “111”, and “LMD1 to LMD2” bits of display
control register (DSPCON) to “11”(all display turns on).

FRM2 to FRM0 bits of display mode register (DSPMOD):

FRM2	FRM1	FRM0	Description
1	1	0	Generates frame frequency 32Hz from the adjusted base clock
1	1	1	Generates frame frequency 64Hz from the adjusted base clock

Chapter 8 16-Bit Timer

8. 16-Bit Timer

8.1 General Description

The 16-bit timer enables following functions.

- Generate periodical interrupts in an arbitrary period
- Generate one shot interrupts in an arbitrary period
- Output pulse signals with an arbitrary frequency to the general ports
- Output one shot pulse signals to the general ports
- Count up the rising edges of the external input signal

The number of 16-bit timer channels is dependent of the product specification.
Table 8-1 shows the number of channels.

Table 8-1 Number of 16-bit Timer channels

Channel no.	ML62Q1300 group				ML62Q1500/ML62Q1800/ML62Q1700 group				
	16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
0	●	●	●	●	●	●	●	●	●
1	●	●	●	●	●	●	●	●	●
2	●	●	●	●	●	●	●	●	●
3	●	●	●	●	●	●	●	●	●
4	−	−	●	●	●	●	●	●	●
5	−	−	●	●	●	●	●	●	●
6	−	−	−	−	−	−	−	●	●
7	−	−	−	−	−	−	−	●	●

●: Available - : Unavailable

8.1.1 Features

- Two timer modes and two operation modes are available

Timer mode	Operation mode	Description
16-bit timer mode	Repeat mode	Count-able to the max. 0xffff Repeat the specified operation until stop by the software.
	One shot mode	Count-able to the max. 0xffff Run the specified operation once and stop it.
8-bit timer mode	Repeat mode	Count-able to the max. 0xff Repeat the specified operation until stop by the software.
	One shot mode	Count-able to the max. 0xff Run the specified operation once and stop it.

- One channel of 16-bit timer is configurable as two channels of 8-bit timer
- LSCLK or HSCLK can be chosen for the timer clock
- A timer clock, a divided time clock or an external input can be chosen for the count clock.
- A timer interrupt request is generated when the value of the timer counter register value coincides with that of the 16-bit timer n data register
- A port output is reversed when the value of the timer counter register value coincides with that of the 16-bit timer n data register
- The initial value of the port can be chosen by a register.

8.1.2 Configuration

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

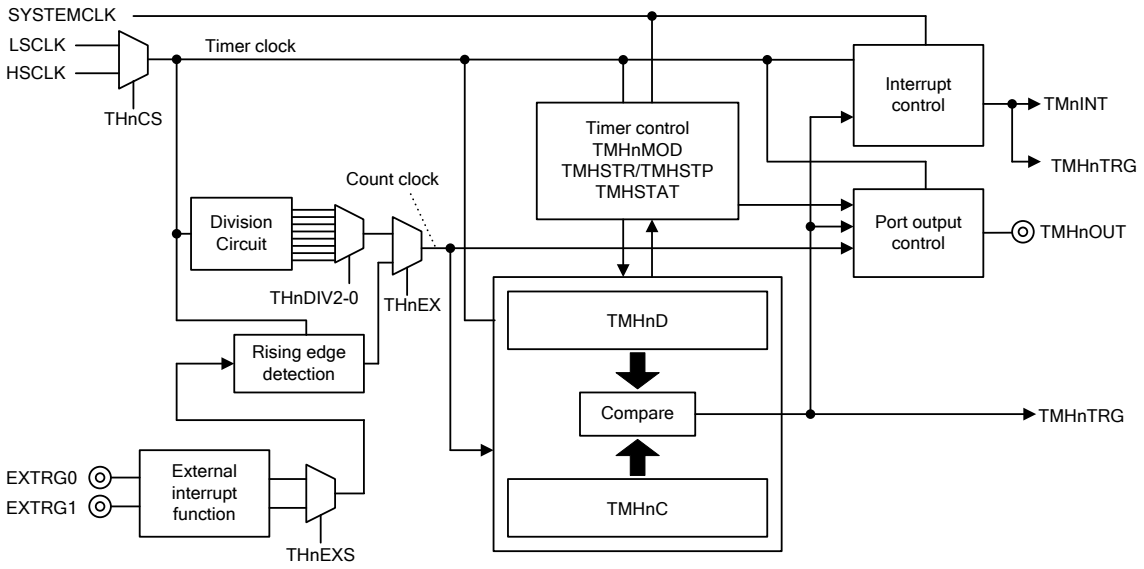


Figure 8-1 Configuration of the timer in 16-bit timer mode

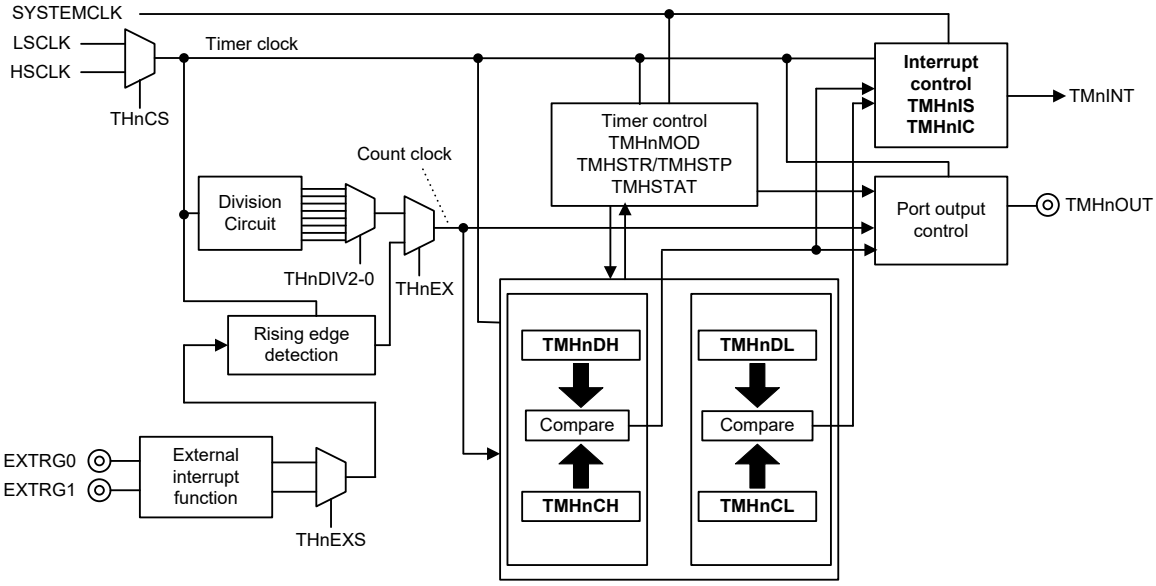


Figure 8-2 Configuration of the timer in 8-bit timer mode

TMnINT	: 16-bit timer n interrupt request	TMHnTRG	: 16-bit timer n trigger
EXTRG0	: EXI0 pin input (come through the noise filter of the external interrupt function)		
EXTRG1	: EXI1 pin input (come through the noise filter of the external interrupt function)		
TMHnD	: 16-bit timer n data register	TMHnC	: 16-bit timer n counter register
TMHnDH	: 16-bit timer n data register upper 8 bit	TMHnCH	: 16-bit timer n counter register upper 8 bit
TMHnDL	: 16-bit timer n data register lower 8 bit	TMHnCL	: 16-bit timer n counter register lower 8 bit

[Note]

- **When the 16-bit timer is used as two channels of 8-bit timer, the same clock settings and interrupts are applied.**
- **In the 8-bit timer mode, the TMHnOUT outputs the comparison result of the upper side ("TMHnDH" and "TMHnCH").**
- **Choose the 16-bit timer mode when using the 16-bit timer DMA request or SA-ADC trigger.**

8.1.3 List of Pin

The I/O pins of the 16-bit timer are assigned to the shared function of the general ports.

Pin name	I/O	Description
EXTRG0	I	External trigger input 0
EXTRG1	I	External trigger input 1
TMHnOUT	O	16-bit timer channel n output When used in an 8-bit timer, only the upper 8-bit timer can output the signal.

Table 8-2 shows the list of the general ports used in the 16-bit timer and the register settings of the ports.

Table 8-2 Ports used in the 16-bit timer and the register settings

Pin name	Shared port		Register	Setting value	ML62Q1300 group				ML62Q1500/ML62Q1800 ML62Q1700 group				
					16pin product	20 pin product	24 pin product	32 pin product	48 pin product	52 pin product	64 pin product	80 pin product	100 pin product
EXTRG0	P02	Primary Func.	P0MOD2	0000_0X01*1	●	●	●	●	●	●	●	●	●
EXTRG1	P03		P0MOD3	0000_0X01*1	●	●	●	●	●	●	●	●	●
TMH0OUT	P04	5th Func.	P0MOD4	0100_XXXX*2	●	●	●	●	●	●	●	●	●
TMH1OUT	P13	5th Func.	P1MOD3	0100_XXXX*2	●	●	●	●	●	●	●	●	●
TMH2OUT	P23	5th Func.	P2MOD3	0100_XXXX*2	●	●	●	●	●	●	●	●	●
TMH3OUT	P13	7th Func.	P1MOD3	0110_XXXX*2	●	●	●	●	●	●	●	●	●
	P33	5th Func.	P3MOD3	0100_XXXX*2	-	●	●	●	●	●	●	●	●
TMH4OUT	P12	5th Func.	P1MOD2	0100_XXXX*2	-	-	●	●	●	●	●	●	●
TMH5OUT	P16	5th Func.	P1MOD6	0100_XXXX*2	-	-	●	●	●	●	●	●	●
TMH6OUT	P70	5th Func.	P7MOD0	0100_XXXX*2	-	-	-	-	-	-	-	●	●
TMH7OUT	P54	5th Func.	P5MOD4	0100_XXXX*2	-	-	-	-	-	-	-	●	●

●: Available to use -: Unavailable

*1 : "X" determines the condition of the port input

X	Condition of the port input
0	Input (without an internal pull-up resistor)
1	Input (with an internal pull-up resistor)

*2 : "XXXX" determines the condition of the port output

XXXX	Condition of the port output
0010	CMOS output
1010	Nch open drain output (without the pull-up)
1111	Nch open drain output (with the pull-up)

8.2 Description of Registers

8.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF300	16-bit timer 0 data register	TMH0DL	TMH0D	R/W	8/16	0xFF
0xF301		TMH0DH		R/W	8	0xFF
0xF302	16-bit timer 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	16-bit timer 6 data register	TMH6DL	TMH6D	R/W	8/16	0xFF
0xF30D		TMH6DH		R/W	8	0xFF
0xF30E	16-bit timer 7 data register	TMH7DL	TMH7D	R/W	8/16	0xFF
0xF30F		TMH7DH		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
0xF31C	16-bit timer 6 counter register	TMH6CL	TMH6C	R/W	8/16	0x00
0xF31D		TMH6CH		R/W	8	0x00
0xF31E	16-bit timer 7 counter register	TMH7CL	TMH7C	R/W	8/16	0x00
0xF31F		TMH7CH		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		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

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF32C	16-bit timer 6 mode register	TMH6MODL	TMH6MOD	R/W	8/16	0x00
0xF32D		TMH6MODH		R/W	8	0x00
0xF32E	16-bit timer 7 mode register	TMH7MODL	TMH7MOD	R/W	8/16	0x00
0xF32F		TMH7MODH		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
0xF33C	16-bit timer 6 interrupt status register	TMH6ISL	TMH6IS	R	8/16	0x00
0xF33D		TMH6ISH		R	8	0x00
0xF33E	16-bit timer 7 interrupt status register	TMH7ISL	TMH7IS	R	8/16	0x00
0xF33F		TMH7ISH		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
0xF34C	16-bit timer 6 interrupt clear register	TMH6ICL	TMH6IC	W	8/16	0x00
0xF34D		TMH6ICH		W	8	0x00
0xF34E	16-bit timer 7 interrupt clear register	TMH7ICL	TMH7IC	W	8/16	0x00
0xF34F		TMH7ICH		W	8	0x00
0xF350	16-bit timer start register	TMHSTRL	TMHSTR	W	8/16	0x00
0xF351		TMHSTRH		W	8	0x00
0xF352	16-bit timer stop register	TMHSTPL	TMHSTP	W	8/16	0x00
0xF353		TMHSTPH		W	8	0x00
0xF354	16-bit timer status register	TMHSTATL	TMHSTAT	R	8/16	0x00
0xF355		TMHSTATH		R	8	0x00

[Note]

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

8.2.2 16-Bit Timer n Data Register (TMHnD: n = 0 to 7)

TMHnD (n = 0 to 7) is a special function register (SFR) to set the comparison value with the 16-bit timer n counter register (TMHnC).
In the 8-bit timer mode, TMHnDL is compared to TMHnCL, and TMHnDH is compared to TMHnCH.

Address: 0xF300(TMH0DL/TMH0D), 0xF301(TMH0DH), 0xF302(TMH1DL/TMH1D), 0xF303(TMH1DH)
0xF304(TMH2DL/TMH2D), 0xF305(TMH2DH), 0xF306(TMH3DL/TMH3D), 0xF307(TMH3DH)
0xF308(TMH4DL/TMH4D), 0xF309(TMH4DH), 0xF30A(TMH5DL/TMH5D), 0xF30B(TMH5DH)
0xF30C(TMH6DL/TMH6D), 0xF30D(TMH6DH), 0xF30E(TMH7DL/TMH7D), 0xF30F(TMH7DH)
Access: R/W
Access size: 8/16 bit
Initial value: 0xFFFF

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

[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 7)

TMHnC is a special function register (SFR) that functions as a 16-bit binary counter.
The data in the TMHnC is counted up synchronizing at the rising edge of the timer clock.
The TMHnC is reset to 0x0000 at the reset function and also when the following event occurred.

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

When an arbitrary value is written in TMHnCH register and TMHnCL register respectively in the 8-bit timer mode, the value of the register is reset to "0x00."
When the value of TMHnDL register coincides with that of the TMHnCL register, the TMHnCL is reset to 0x00.
Also, when the value of TMHnDH register coincides with that of the TMHnCH register, the TMHnCH is reset to 0x00.

Address: 0xF310(TMH0C/TMH0CL), 0xF311(TMH0CH), 0xF312(TMH1C/TMH1CL), 0xF313(TMH1CH)
0xF314(TMH2C/TMH2CL), 0xF315(TMH2CH), 0xF316(TMH3C/TMH3CL), 0xF317(TMH3CH)
0xF318(TMH4C/TMH4CL), 0xF319(TMH4CH), 0xF31A(TMH5C/TMH5CL), 0xF31B(TMH5CH)
0xF31C(TMH6C/TMH6CL), 0xF31D(TMH6CH), 0xF31E(TMH7C/TMH7CL), 0xF31F(TMH7CH)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

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

8.2.4 16-Bit Timer n Mode Register (TMHnMOD: n = 0 to 7)

TMHnMOD is a special function register (SFR) to control the operation mode of 16-bit timer.

Address: 0xF320(TMH0MODL/TMH0MOD), 0xF321(TMH0MODH),
0xF322(TMH1MODL/TMH1MOD), 0xF323(TMH1MODH),
0xF324(TMH2MODL/TMH2MOD), 0xF325(TMH2MODH),
0xF326(TMH3MODL/TMH3MOD), 0xF327(TMH3MODH),
0xF328(TMH4MODL/TMH4MOD), 0xF329(TMH4MODH),
0xF32A(TMH5MODL/TMH5MOD), 0xF32B(TMH5MODH),
0xF32C(TMH6MODL/TMH6MOD), 0xF32D(TMH6MODH),
0xF32E(TMH7MODL/TMH7MOD), 0xF32F(TMH7MODH)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHnMOD															
Byte	TMHnMODH								TMHnMODL							
Bit	rsvd	rsvd	rsvd	rsvd	rsvd	THn NEG	THn OST	THn 8BM	rsvd	THn DIV2	THn DIV1	THn DIV0	THn EXS	THn EX	rsvd	THn CS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 11	rsvd	Reserved bits. Write “0” to these.
10	THnNEG	This bit is used to choose the output polarity of timer out (TMHnOUT). 0: Positive logic (initial level is "L") (Initial value) 1: Negative logic (initial level is "H")
9	THnOST	This bit is used to choose the operation mode of the 16-bit timer n. 0: Repeat mode (Initial value) 1: One-shot mode
8	THn8BM	This bit is used to choose whether the timer works as one 16-bit timer or two channels of 8-bit timer. 0: 16-bit timer mode (Initial value) 1: 8-bit timer mode
7	rsvd	Reserved bit. Write “0” to this.
6 to 4	THnDIV2 to THnDIV0	These bits are used to choose frequency dividing ratio for the count clock in the 16-bit timer n . 000: No dividing (Initial value) 001: 1/2 of the timer clock 010: 1/4 of the timer clock 011: 1/8 of the timer clock 100: 1/16 of the timer clock 101: 1/32 of the timer clock 110: 1/64 of the timer clock 111: 1/128 of the timer clock
3	THnEXS	This bit is used to choose the external trigger supplied as the count clock of the 16-bit timer n. 0: EXTRG0 (Initial value) 1: EXTRG1
2	THnEX	This bit is used to choose the count clock (THnCK) of the 16-bit timer n. 0: The timer is counted by the clock chosen by the THnCS bit and divided by the ratio chosen by the THnDIV2 to 0 bit. (Initial value) 1: The timer is counted by the rising edge of the external trigger signal detected by the clock chosen by the THnCS bit.
1	rsvd	Reserved bit. Write “0” to this.

Bit No.	Bit symbol name	Description
0	THnCS	This bit is used to choose the timer clock of the 16-bit timer n. 0: LSCLK (Initial value) 1: HSCLK

[Note]

- **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 THnOST bit, THnDIV2 to 0 , THnCS0, THnEXS, THnEX are common for both upper side and lower side of the timer.**
- **Configure the external trigger on the condition that the timer clock is set to "No dividing" and THnCS=1 if the system clock is HSCLK or THnCS=0 if the system clock is LSCLK. See figure 8-13 "External Input Count Setting Flow" in section 8.3.3.2.**

8.2.5 16-Bit Timer n Interrupt Status Register (TMHnIS: n = 0 to 7)

TMHnIS is a special function register (SFR) to indicate the status of the interrupt used in the 8-bit timer mode.
In the 8-bit timer mode, it can be checked whether the upper side or lower side of the timer generated the interrupt request.
The TMHnIS is fixed to 0x00 in the 16-bit timer mode.
The THnHIS bit(bit1) or THnLIS bit(bit0) is reset to "0" at the reset function or reset by writing "1" to the same number of bit in the TMHnIC register.

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.

Address: 0xF330(TMH0ISL/TMH0IS), 0xF331(TMH0ISH), 0xF332(TMH1ISL/TMH1IS), 0xF333(TMH1ISH)
0xF334(TMH2ISL/TMH2IS), 0xF335(TMH2ISH), 0xF336(TMH3ISL/TMH3IS), 0xF337(TMH3ISH)
0xF338(TMH4ISL/TMH4IS), 0xF339(TMH4ISH), 0xF33A(TMH5ISL/TMH5IS), 0xF33B(TMH5ISH)
0xF33C(TMH6ISL/TMH6IS), 0xF33D(TMH6ISH), 0xF33E(TMH7ISL/TMH7IS), 0xF33F(TMH7ISH)
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	TMHnIS															
Byte	TMHnISH								TMHnISL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	THnHIS	THnLIS
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 2	-	Reserved bits
1	THnHIS	Interrupt on upper side of 16-bit timer n 0: The interrupt request does not exist (Initial value) 1: The interrupt request exists
0	THnLIS	Interrupt on lower side of 16-bit timer n 0: The interrupt request does not exist (Initial value) 1: The interrupt request exists

[Note]

- **When the THnHIS bit or the THnLIS bit is "1", the interrupt request in the same channel of 8-bit timer does not activate. Clear the THnHIS bit or the THnLIS bit by writing "1" to the same number of bit in the TMHnIC register.**

8.2.6 16-Bit Timer n Interrupt Clear Register (TMHnIC: n = 0 to 7)

TMHnIC is a write-only special function register (SFR) to clear the status of the interrupt used in the 8-bit timer mode. This is not used in the 16-bit mode.
If THnHIC bit(bit1) or THnLIC bit(bit0) of TMHnIC register is set to "1", the interrupt request indicated by the same number of bit in the TMHnIS register is cleared.
Also, if the THnIR bit is set to "1" it generates the interrupt when there is a unhandled interrupt request in the TMHnIS register. TMHnIC register is write-only register and returns always "0x0000" for reading.

Address: 0xF340(TMH0ICL/TMH0IC), 0xF341(TMH0ICH), 0xF342(TMH1ICL/TMH1IC), 0xF343(TMH1ICH)
0xF344(TMH2ICL/TMH2IC), 0xF345(TMH2ICH), 0xF346(TMH3ICL/TMH3IC), 0xF347(TMH3ICH)
0xF348(TMH4ICL/TMH4IC), 0xF349(TMH4ICH), 0xF34A(TMH5ICL/TMH5IC), 0xF34B(TMH5ICH)
0xF34C(TMH6ICL/TMH6IC), 0xF34D(TMH6ICH), 0xF34E(TMH7ICL/TMH7IC), 0xF34F(TMH7ICH)
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	TMHnIC															
Byte	TMHnICH								TMHnICL							
Bit	-	-	-	-	-	-	-	-	THnIR	-	-	-	-	-	THnHIC	THnLIC
R/W	R	R	R	R	R	R	R	R	W	R	R	R	R	R	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bits
7	THnIR	This bit is used to generate a unhandled interrupt request of the upper side 8-bit timer or lower side 8-bit timer (when a not-target interrupt request for clear remains in the TMHnIS register) Writing "0": Invalid Writing "1": Generate the unhandled interrupt request.
6 to 2	-	Reserved bits
1	THnHIC	This bit is used to clear the interrupt request of the upper side 8-bit timer. Writing "0": Invalid Writing "1": Clear the interrupt request status
0	THnLIC	This bit is used to clear the interrupt request of the lower side 8-bit timer. Writing "0": Invalid Writing "1": Clear the interrupt request status

8.2.7 16-Bit Timer Start Register (TMHSTR)

TMHSTR is a special function register (SFR) to control to start counting the 16-bit timer n.
TMHSTRL is used in the 16-bit timer mode.
TMHSTRH is used to start counting the upper side 8bit counter in the 8-bit timer mode.
TMHSTRL is used to start counting the lower side 8bit counter in the 8-bit timer mode.
TMHSTR is a write-only register and returns always "0x0000"for reading.

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHSTR															
Byte	TMHSTRH								TMHSTRL							
Bit	TH7H RUN	TH6H RUN	TH5H RUN	TH4H RUN	TH3H RUN	TH2H RUN	TH1H RUN	TH0H RUN	TH7 RUN	TH6 RUN	TH5 RUN	TH4 RUN	TH3 RUN	TH2 RUN	TH1 RUN	TH0 RUN
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	TH7HRUN	In the 8-bit timer mode, controls the upper side 8bit counter of 16bit timer 7 Writing "0": Invalid Writing "1": Start counting
14	TH6HRUN	In the 8-bit timer mode, controls the upper side 8bit counter of 16bit timer 6 Writing "0": Invalid Writing "1": Start counting
13	TH5HRUN	In the 8-bit timer mode, controls the upper side 8bit counter of 16bit timer 5 Writing "0": Invalid Writing "1": Start counting
12	TH4HRUN	In the 8-bit timer mode, controls the upper side 8bit counter of 16bit timer 4 Writing "0": Invalid Writing "1": Start counting
11	TH3HRUN	In the 8-bit timer mode, controls the upper side 8bit counter of 16bit timer 3 Writing "0": Invalid Writing "1": Start counting
10	TH2HRUN	In the 8-bit timer mode, controls the upper side 8bit counter of 16bit timer 2 Writing "0": Invalid Writing "1": Start counting
9	TH1HRUN	In the 8-bit timer mode, controls the upper side 8bit counter of 16bit timer 1 Writing "0": Invalid Writing "1": Start counting
8	TH0HRUN	In the 8-bit timer mode, controls the upper side 8bit counter of 16bit timer 0 Writing "0": Invalid Writing "1": Start counting

Bit no.	Bit symbol name	Description
7	TH7RUN	In the 16-bit timer mode, controls the counter of 16-bit timer 7 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 7 Writing "0": Invalid Writing "1": Start counting
6	TH6RUN	In the 16-bit timer mode, controls the counter of 16-bit timer 6 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 6 Writing "0": Invalid Writing "1": Start counting
5	TH5RUN	In the 16-bit timer mode, controls the counter of 16-bit timer 5 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 5 Writing "0": Invalid Writing "1": Start counting
4	TH4RUN	In the 16-bit timer mode, controls the counter of 16-bit timer 4 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 4 Writing "0": Invalid Writing "1": Start counting
3	TH3RUN	In the 16-bit timer mode, controls the counter of 16-bit timer 3 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 3 Writing "0": Invalid Writing "1": Start counting
2	TH2RUN	In the 16-bit timer mode, controls the counter of 16-bit timer 2 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 2 Writing "0": Invalid Writing "1": Start counting
1	TH1RUN	In the 16-bit timer mode, controls the counter of 16-bit timer 1 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 1 Writing "0": Invalid Writing "1": Start counting
0	TH0RUN	In the 16-bit timer mode, controls the counter of 16-bit timer 0 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 0 Writing "0": Invalid Writing "1": Start counting

[Note]

- **The bit 15 to 8 of TMHSTR register are not used in the 16-bit timer mode. Writing “1” to those bits is ignored.**
- **Set THnRUN/THnHRUN bits when the timer n is stopped(THnSTAT/THnHSTAT bits of TMHSTAT register are “0”).**

8.2.8 16-Bit Timer Stop Register (TMHSTP)

TMHSTP is a special function register (SFR) to control to stop counting the 16-bit timer n.
TMHSTPL is used in the 16-bit timer mode.
TMHSTPH is used to start counting the upper side 8bit counter in the 8-bit timer mode.
TMHSTPL is used to start counting the lower side 8bit counter in the 8-bit timer mode.
TMHSTP is a write-only register and returns always "0x0000" for reading.

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHSTP															
Byte	TMHSTPH								TMHSTPL							
Bit	TH7HSTP	TH6HSTP	TH5HSTP	TH4HSTP	TH3HSTP	TH2HSTP	TH1HSTP	TH0HSTP	TH7STP	TH6STP	TH5STP	TH4STP	TH3STP	TH2STP	TH1STP	TH0STP
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	TH7HSTP	In the 8-bit timer mode, controls the upper side 8bit counter of 16-bit timer 7 Writing "0": Invalid Writing "1": Stop counting
14	TH6HSTP	In the 8-bit timer mode, controls the upper side 8bit counter of 16-bit timer 6 Writing "0": Invalid Writing "1": Stop counting
13	TH5HSTP	In the 8-bit timer mode, controls the upper side 8bit counter of 16-bit timer 5 Writing "0": Invalid Writing "1": Stop counting
12	TH4HSTP	In the 8-bit timer mode, controls the upper side 8bit counter of 16-bit timer 4 Writing "0": Invalid Writing "1": Stop counting
11	TH3HSTP	In the 8-bit timer mode, controls the upper side 8bit counter of 16-bit timer 3 Writing "0": Invalid Writing "1": Stop counting
10	TH2HSTP	In the 8-bit timer mode, controls the upper side 8bit counter of 16-bit timer 2 Writing "0": Invalid Writing "1": Stop counting
9	TH1HSTP	In the 8-bit timer mode, controls the upper side 8bit counter of 16-bit timer 1 Writing "0": Invalid Writing "1": Stop counting
8	TH0HSTP	In the 8-bit timer mode, controls the upper side 8bit counter of 16-bit timer 0 Writing "0": Invalid Writing "1": Stop counting

Bit no.	Bit symbol name	Description
7	TH7STP	In the 16-bit timer mode, controls the counter of 16-bit timer 7 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 7 Writing "0": Invalid Writing "1": Stop counting
6	TH6STP	In the 16-bit timer mode, controls the counter of 16-bit timer 6 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 6 Writing "0": Invalid Writing "1": Stop counting
5	TH5STP	In the 16-bit timer mode, controls the counter of 16-bit timer 5 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 5 Writing "0": Invalid Writing "1": Stop counting
4	TH4STP	In the 16-bit timer mode, controls the counter of 16-bit timer 4 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 4 Writing "0": Invalid Writing "1": Stop counting
3	TH3STP	In the 16-bit timer mode, controls the counter of 16-bit timer 3 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 3 Writing "0": Invalid Writing "1": Stop counting
2	TH2STP	In the 16-bit timer mode, controls the counter of 16-bit timer 2 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 2 Writing "0": Invalid Writing "1": Stop counting
1	TH1STP	In the 16-bit timer mode, controls the counter of 16-bit timer 1 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 1 Writing "0": Invalid Writing "1": Stop counting
0	TH0STP	In the 16-bit timer mode, controls the counter of 16-bit timer 0 In the 8-bit timer mode, controls the lower side 8bit counter of 16-bit timer 0 Writing "0": Invalid Writing "1": Stop counting

[Note]

- **The bit 15 to 8 of TMHSTP register are not used in the 16-bit timer mode. Writing "1" to those bits is ignored.**
- **Set THnSTP/THnHSTP bits when the timer n is running (THnSTAT/THnHSTAT bits of TMHSTAT register are "1").**
- **To stop counting during one-shot mode with HSCLK is selected for the system clock and LSCLK (THnCS bit = 0 in the TMHnMOD register) is selected for the timer clock, set the THnSTP bit to "1", and then change the timer clock to HSCLK (THnCS bit = 1 in the TMHnMOD register), and set the timer clock to LSCLK again (THnCS bit = 0 in the TMHnMOD register).**

8.2.9 16-Bit Timer Status Register (TMHSTAT)

TMHSTAT is a special function register (SFR) to indicate the status of the 16-bit timer n.
TMHSTATL is used in the 16-bit timer mode. TMHSTATH is fixed to 0 in the 16-bit timer mode.
TMHSTATH is used to start counting the upper side 8bit counter in the 8-bit timer mode.
TMHSTATL is used to start counting the lower side 8bit counter in the 8-bit timer mode.

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHSTAT															
Byte	TMHSTATH								TMHSTATL							
Bit	TH7HSTAT	TH6HSTAT	TH5HSTAT	TH4HSTAT	TH3HSTAT	TH2HSTAT	TH1HSTAT	TH0HSTAT	TH7STAT	TH6STAT	TH5STAT	TH4STAT	TH3STAT	TH2STAT	TH1STAT	TH0STAT
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	TH7HSTAT	In the 8-bit timer mode, indicates the status on the upper side 8bit counter of 16-bit timer 7 0: The counting is stopped (Initial value) 1: The counting is in progress
14	TH6HSTAT	In the 8-bit timer mode, indicates the status on the upper side 8bit counter of 16-bit timer 6 0: The counting is stopped (Initial value) 1: The counting is in progress
13	TH5HSTAT	In the 8-bit timer mode, indicates the status on the upper side 8bit counter of 16-bit timer 5 0: The counting is stopped (Initial value) 1: The counting is in progress
12	TH4HSTAT	In the 8-bit timer mode, indicates the status on the upper side 8bit counter of 16-bit timer 4 0: The counting is stopped (Initial value) 1: The counting is in progress
11	TH3HSTAT	In the 8-bit timer mode, indicates the status on the upper side 8bit counter of 16-bit timer 3 0: The counting is stopped (Initial value) 1: The counting is in progress
10	TH2HSTAT	In the 8-bit timer mode, indicates the status on the upper side 8bit counter of 16-bit timer 2 0: The counting is stopped (Initial value) 1: The counting is in progress
9	TH1HSTAT	In the 8-bit timer mode, indicates the status on the upper side 8bit counter of 16-bit timer 1 0: The counting is stopped (Initial value) 1: The counting is in progress
8	TH0HSTAT	In the 8-bit timer mode, indicates the status on the upper side 8bit counter of 16-bit timer 0 0: The counting is stopped (Initial value) 1: The counting is in progress

Bit no.	Bit symbol name	Description
7	TH7STAT	In the 16-bit timer mode, indicates the status on the 16-bit timer 7 In the 8-bit timer mode, indicates the status on the lower side 8bit counter of 16-bit timer 7 0: The counting is stopped (Initial value) 1: The counting is in progress
6	TH6STAT	In the 16-bit timer mode, indicates the status on the 16-bit timer 6 In the 8-bit timer mode, indicates the status on the lower side 8bit counter of 16-bit timer 6 0: The counting is stopped (Initial value) 1: The counting is in progress
5	TH5STAT	In the 16-bit timer mode, indicates the status on the 16-bit timer 5 In the 8-bit timer mode, indicates the status on the lower side 8bit counter of 16-bit timer 5 0: The counting is stopped (Initial value) 1: The counting is in progress
4	TH4STAT	In the 16-bit timer mode, indicates the status on the 16-bit timer 4 In the 8-bit timer mode, indicates the status on the lower side 8bit counter of 16-bit timer 4 0: The counting is stopped (Initial value) 1: The counting is in progress
3	TH3STAT	In the 16-bit timer mode, indicates the status on the 16-bit timer 3 In the 8-bit timer mode, indicates the status on the lower side 8bit counter of 16-bit timer 3 0: The counting is stopped (Initial value) 1: The counting is in progress
2	TH2STAT	In the 16-bit timer mode, indicates the status on the 16-bit timer 2 In the 8-bit timer mode, indicates the status on the lower side 8bit counter of 16-bit timer 2 0: The counting is stopped (Initial value) 1: The counting is in progress
1	TH1STAT	In the 16-bit timer mode, indicates the status on the 16-bit timer 1 In the 8-bit timer mode, indicates the status on the lower side 8bit counter of 16-bit timer 1 0: The counting is stopped (Initial value) 1: The counting is in progress
0	TH0STAT	In the 16-bit timer mode, indicates the status on the 16-bit timer 0 In the 8-bit timer mode, indicates the status on the lower side 8bit counter of 16-bit timer 0 0: The counting is stopped (Initial value) 1: The counting is in progress

8.3 Description of Operation

Two timer modes are available for the 16-bit timer:
- 16-bit timer mode
- 8-bit timer mode

8.3.1 16-Bit Timer Mode

When the THn8BM bit of the TMHnMOD register is set to "0", the timer operates in the 16-bit timer mode.
In the 16-bit timer mode, writing "1" to the THnRUN bit causes the 16-bit counter to start counting up in synchronization with the rising edges of the timer clock.
If output of the general-purpose port is enabled by choosing the timer output (TMHnOUT) through the shared function setting of the port, the output of the port is reversed when the timer count value matches with TMHnD register value. In addition, writing "1" to the THnSTP bit during counting causes the counting to stop in synchronization with the timer clock and the output of the port is reset to the initial value. For the the initial value of the port, "H" and "L" levels can be chosen through the THnNEG bit of the TMHnMOD register.

In the 16-bit timer mode, following two operation modes are available:
- Repeat mode
- One-shot mode

8.3.1.1 Repeat Mode

Figure 8-3 shows the repeat mode operation in the 16-bit timer mode.

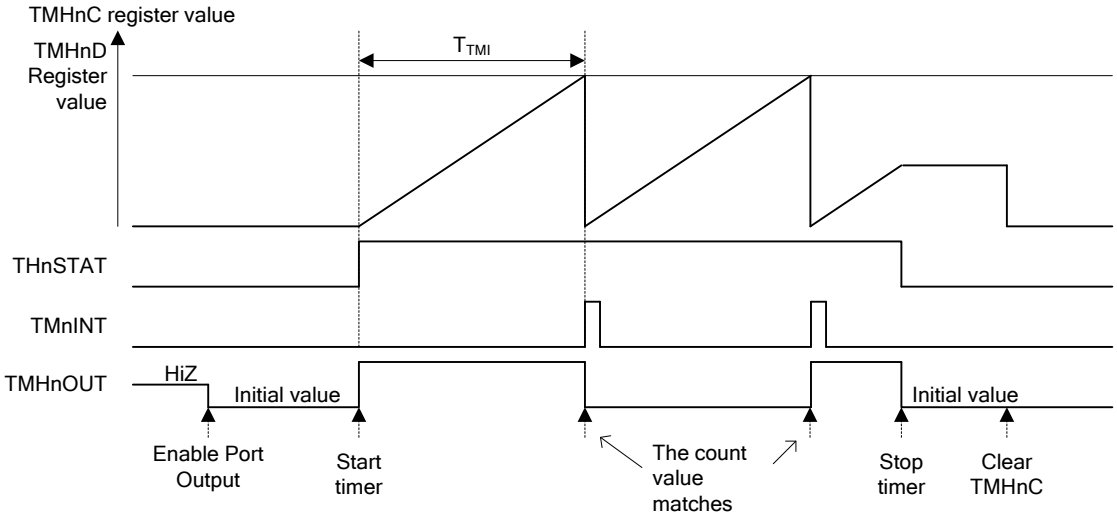


Figure 8-3 Repeat Mode Operation Timing in 16-Bit Timer Mode

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

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

TMHnD : TMHnD register setting value (0001H to 0FFFFH)
fTHnCK : Count clock frequency chosen in the TMHnMODL register

See Section 8.3.3.1 "Start/Stop Timing" for the timing of the timer start/stop and counting up. See Section 8.3.3.2 "External Input Count Timing" for the counting up timing when using the external input.

8.3.1.2 One-shot Mode

Figure 8-4 shows the one-shot mode operation in the 16-bit timer mode.

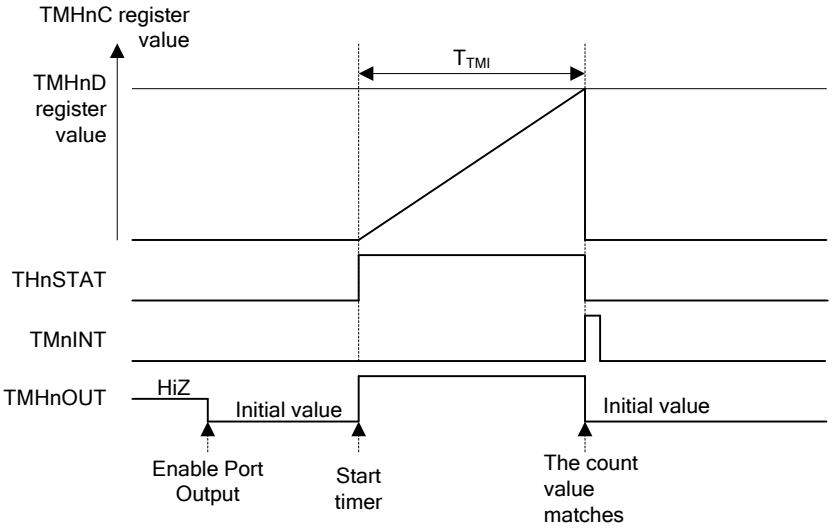


Figure 8-4 One-shot Mode Operation Timing in 16-Bit Timer Mode

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

8.3.1.3 Setting Example

Figure 8-5 shows a setting example when using the 16-bit timer mode.

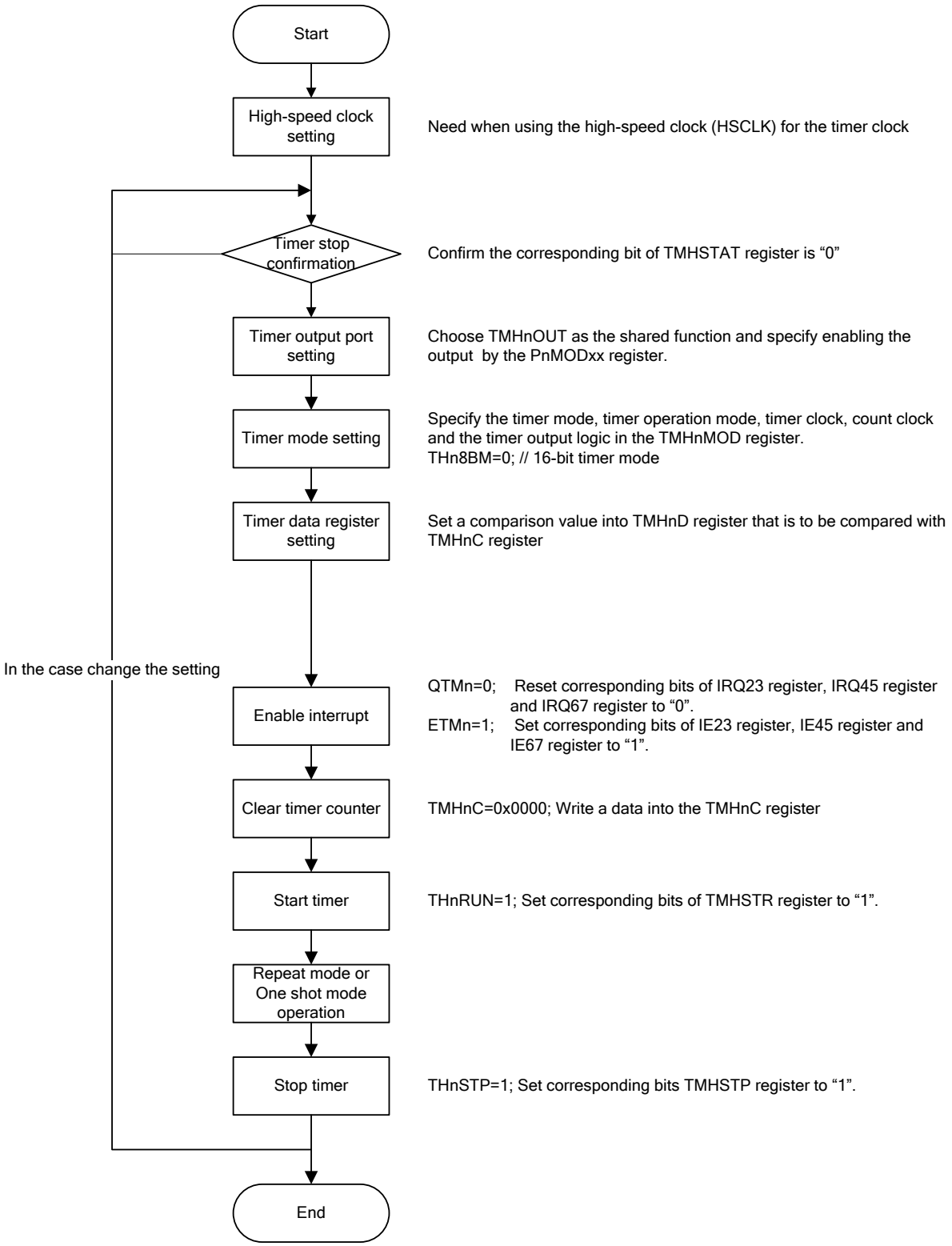


Figure 8-5 Setting Example of 16-Bit Timer Mode

8.3.2 8-Bit Timer Mode

When the THn8BM bit of the TMHnMOD register is set to "1", the timer operates in the 8-bit timer mode.
In the 8-bit timer mode, writing "1" to the THnHRUN bit causes the higher 8 bits (upper side 8-bit timer) of the 16-bit counter to start counting up in synchronization with the rising edges of the timer clock. Writing "1" to the THnRUN bit causes the lower 8 bits (lower side 8-bit timer) of the 16-bit counter to start counting up in synchronization with the rising edges of the timer clock.
The operation mode and count clock of the upper side 8-bit timer and those of the lower side 8-bit timer are identical. Count start/stop can be controlled separately for each of the upper and lower side 8-bit timers.

In the 8-bit timer mode, following two operation modes are available:
- Repeat mode
- One-shot mode

8.3.2.1 Repeat Mode

Figure 8-6 shows the repeat mode operation in the 8-bit timer mode.

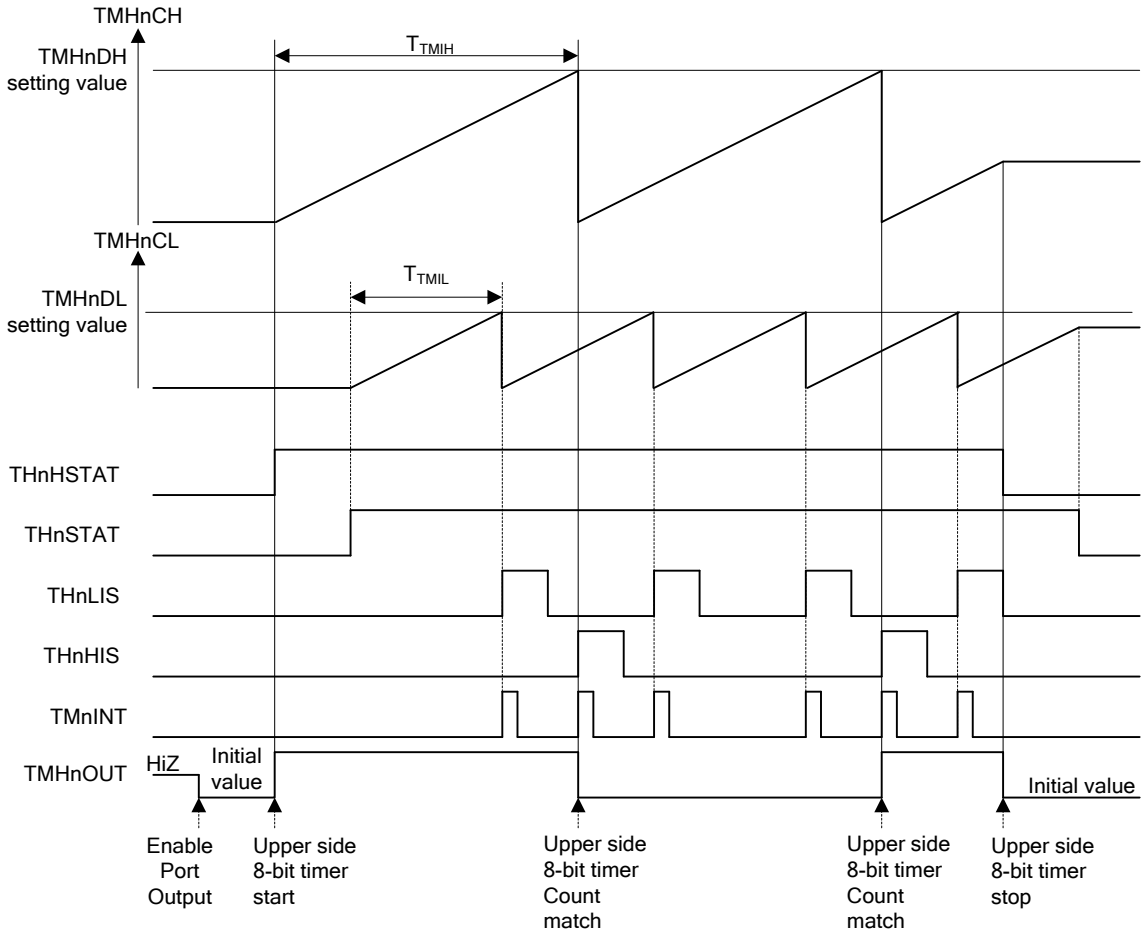


Figure 8-6 Repeat Mode Operation Timing in 8-Bit Timer Mode

In the repeat mode, when the upper side 8-bit timer count value matches with the TMHnDH register value, 16-bit timer n interrupt (TMnINT) is generated and the output of the port is reversed. Then, the timer count value is reset to "0x00" and the counting up operation is automatically resumed. When the lower side 8-bit timer count value matches with the TMHnDL register value, 16-bit timer n interrupt (TMnINT) is generated. Then, the timer count value automatically is reset to "0x00" and the counting up operation is continued.

The interrupt request TMnINT generation cycle and the port output variation cycle (only upper side) can be expressed in the following formula. The TMnINT is shared by the upper side 8-bit timer and the lower side 8-bit timer. See [8.3.2.4 About Interrupt Request in 8-Bit Timer Mode] for more detail.

$$T_{TMIH} = \frac{TMHnDH + 1}{fTHnCK\ (Hz)}$$

$$T_{TMIL} = \frac{TMHnDL + 1}{fTHnCK\ (Hz)}$$

(n=0 to 7)
TMHnDH: TMHnDH register setting value (0x01 to 0xFF)
TMHnDL: TMHnDL register setting value (0x01 to 0xFF)
fTHnCK: Count clock frequency chosen in the TMHnMODL register

See Section 8.3.3.1 "Start/Stop Timing" for the timing of the timer start/stop and counting up. See Section 8.3.3.2 "External Input Count Timing" for the counting up timing when using the external input.

8.3.2.2 One-shot Mode

Figure 8-7 shows the operation waveforms in the one-shot mode.

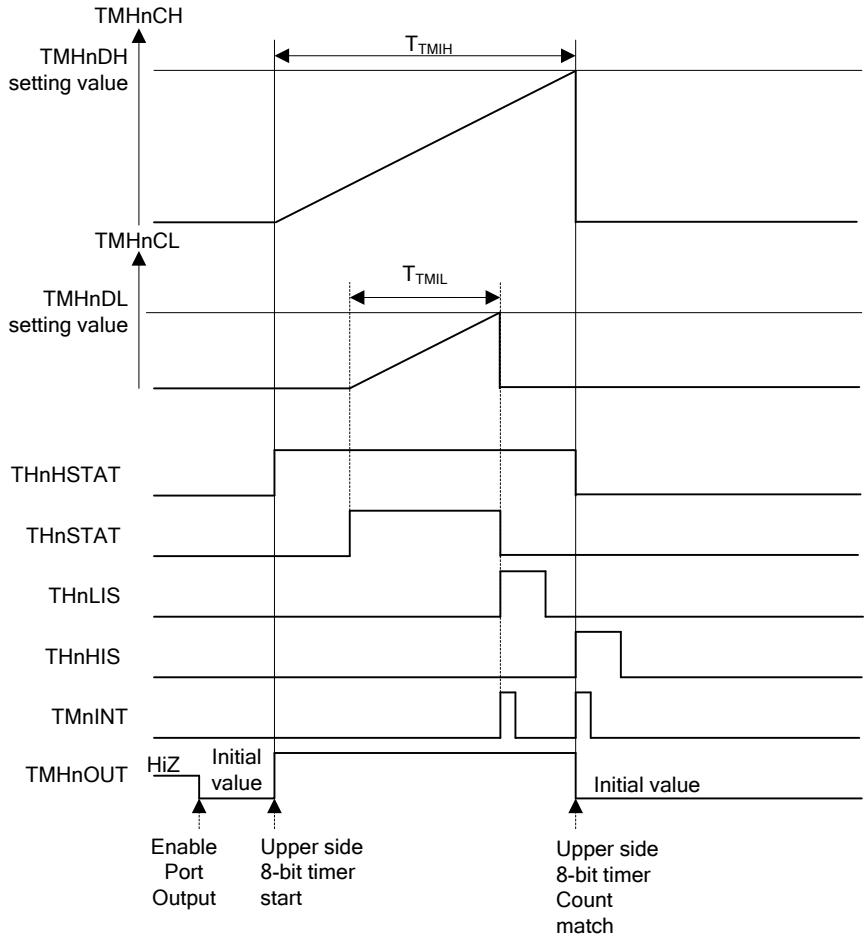


Figure 8-7 One-shot Mode Operation Timing in 8-Bit Timer Mode.

In the one-shot mode, if the upper side 8-bit timer count value matches with the TMHnDH register value, the timer count value is reset to "0x00", then the counting up operation stops. If the lower side 8-bit timer count value matches with the TMHnDL register value, the timer count value is reset to "0x00", then the counting up operation stops.
Other operations are the same as those in the repeat mode.

[Note]

- **In the 8-bit timer mode, the timer output is only available by the upper side 8-bit timer.**

8.3.2.3 Setting Example

Figure 8-8 shows a setting example when using the 8-bit timer mode.

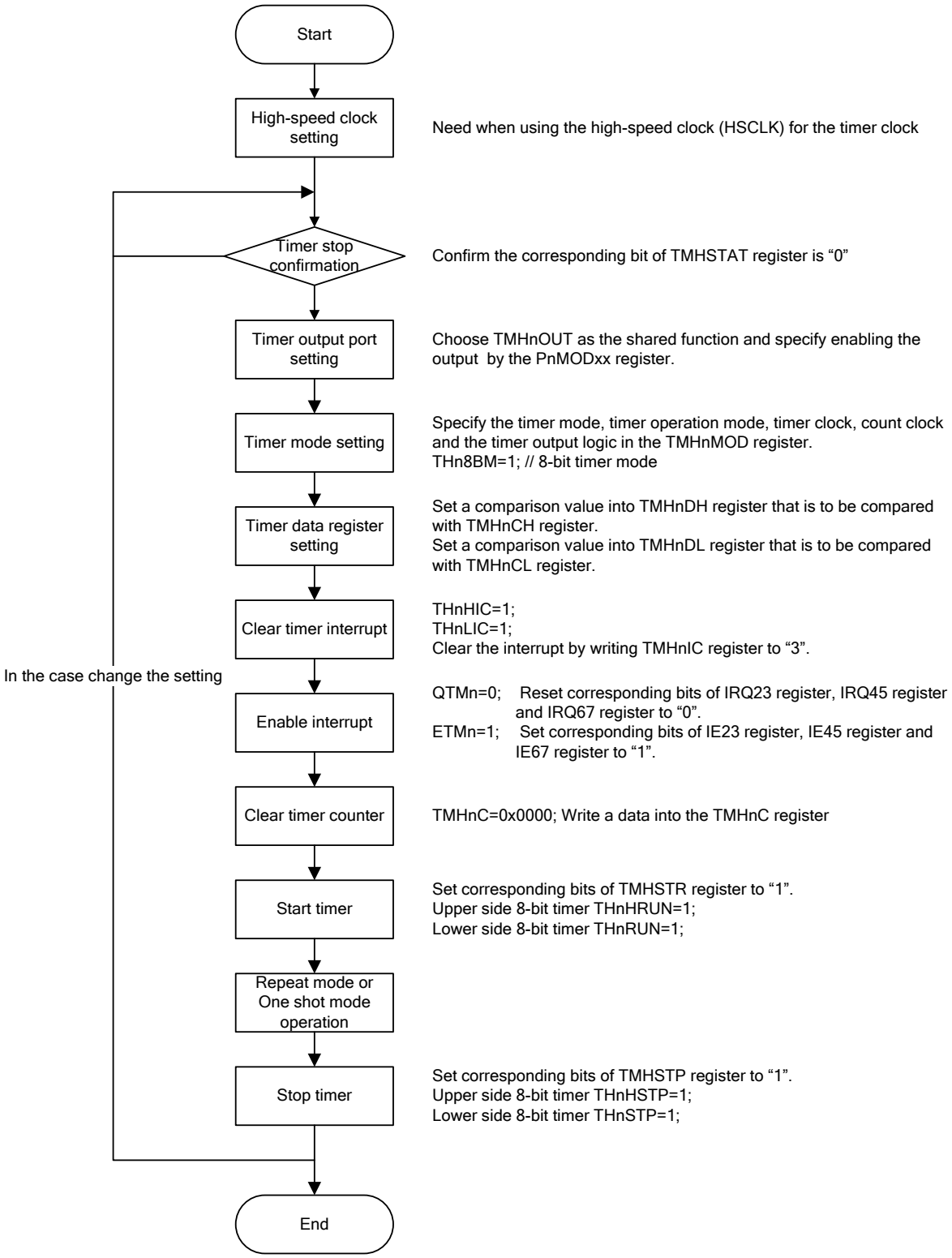


Figure 8-8 Setting Example of 8-Bit Timer Mode

8.3.2.4 About Interrupt Request in 8-Bit Timer Mode

In the 8-bit timer mode, the same interrupt requests (TMnINT) are used for the upper side 8-bit timer and the lower side 8-bit timer.
When the upper side 8-bit timer count value matches the TMHnDH register value, or lower side 8-bit timer count value matches the TMHnDL register value, an interrupt request is generated if any interrupt request has not been generated for neither the upper nor lower side 8-bit timer. Table 8-3 shows interrupt request generation conditions in the 8-bit timer mode.

Table 8-3 Interrupt Request Generation Conditions in 8-Bit Timer Mode.

Condition			Interrupt request (TMnINT)
State of timer counting	State of TMHnIS register		
	THnHIS	THnLIS	
Upper side 8-bit timer count value matches TMHnDH register value	0	0	Generated
	0	1	Not generated
	1	0	
	1	1	
Lower side 8-bit timer count value matches TMHnDL register value	0	0	Generated
	0	1	Not generated
	1	0	
	1	1	

If an 8-bit timer interrupt of the other side is generated while interrupt processing is in progress, the unprocessed interrupt request can be generated again after resuming the main process by writing "1" to the THnIR bit of the TMHnICL register prior to resuming the main process. Figure 8-9 shows an example of the interrupt process in the 8-bit timer mode.

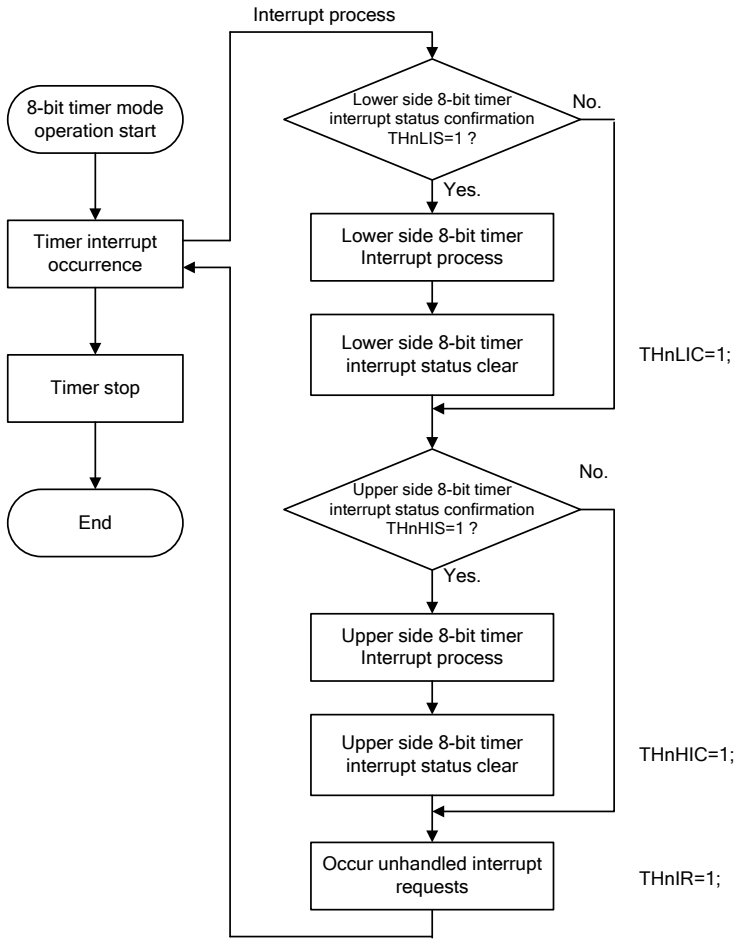


Figure 8-9 Example of Interrupt Process in 8-Bit Timer Mode

8.3.3 Common Operation

8.3.3.1 Start/Stop Timing

Writing "1" to the THnRUN bit of the TMHSTR register causes the counting operation to start at the rising edge of the timer clock that follows.
Figure 8-10 shows the timer start timing when the timer clock is LSCLK and frequency dividing ratio of the count clock is 1/2 of the timer clock.

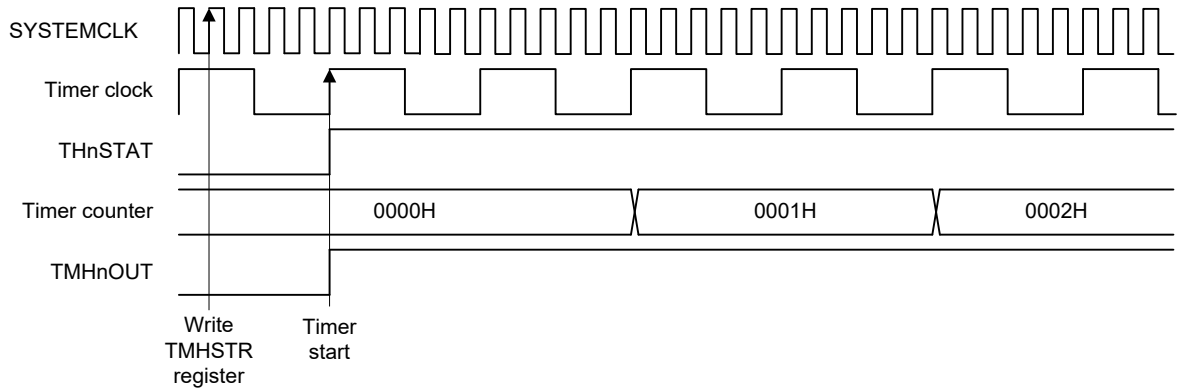


Figure 8-10 Start Timing

Writing "1" to the THnSTP bit of the TMHSTP register causes the counting operation to stop at the rising edge of the timer clock that follows.
Figure 8-11 shows the timer stop timing when the timer clock is LSCLK and frequency dividing ratio of the count clock is 1/2 of the timer clock.

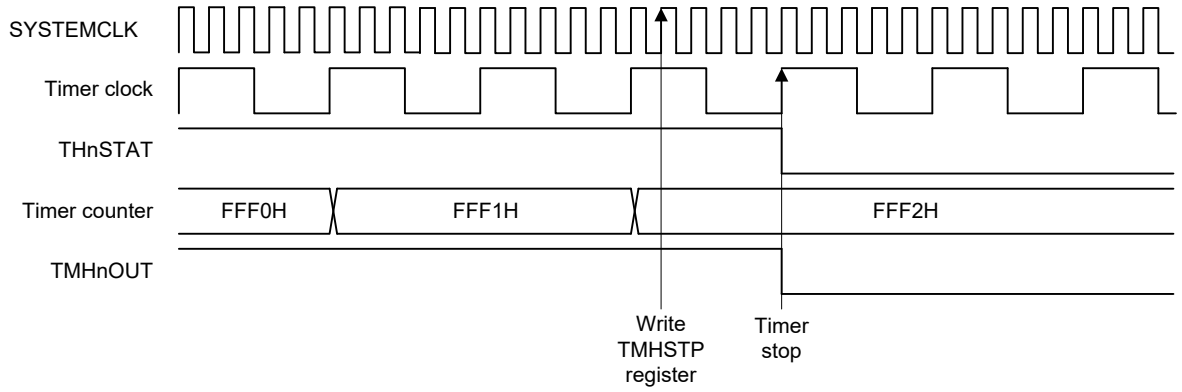


Figure 8-11 Stop Timing

[Note]

- **Since counting operation is not suspended during the period in which THnSTAT bit is "1", restart of the counting is ignored even if THnRUN bit is set to "1" in this period. To restart the counting, make sure that the THnSTAT is set to "0", and then set the THnRUN bit to "1".**
- **After the THnRUN bit is set to "1", the first interrupt has a time error equivalent to maximum of one clock of the timer clock because the counting operation starts in synchronization with the timer clock. The 2nd timer interrupt or later interrupts have constant cycles.**
- **After the THnSTP bit is set to "1", a 16-bit timer n interrupt (TMnINT) may be generated depending on the stop timing because the counting operation stops in synchronization with the timer clock.**

8.3.3.2 External Input Count Timing

If the external input is chosen for a count clock, counting up is executed through sampling the external input with the timer clock to detect a rising edge.
Figure 8-12 shows the count timing when using the external input.

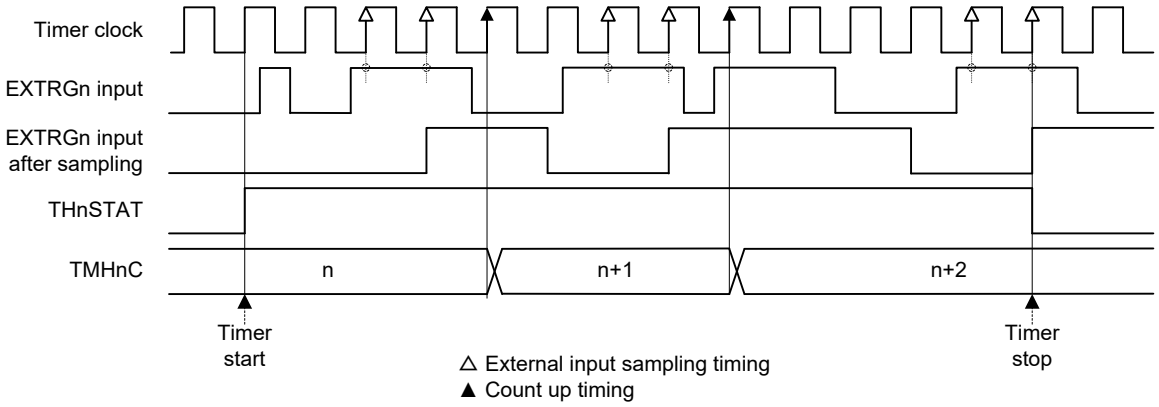


Figure 8-12 External Input Count Timing

Figure 8-13 shows a setting flow when using the external input.

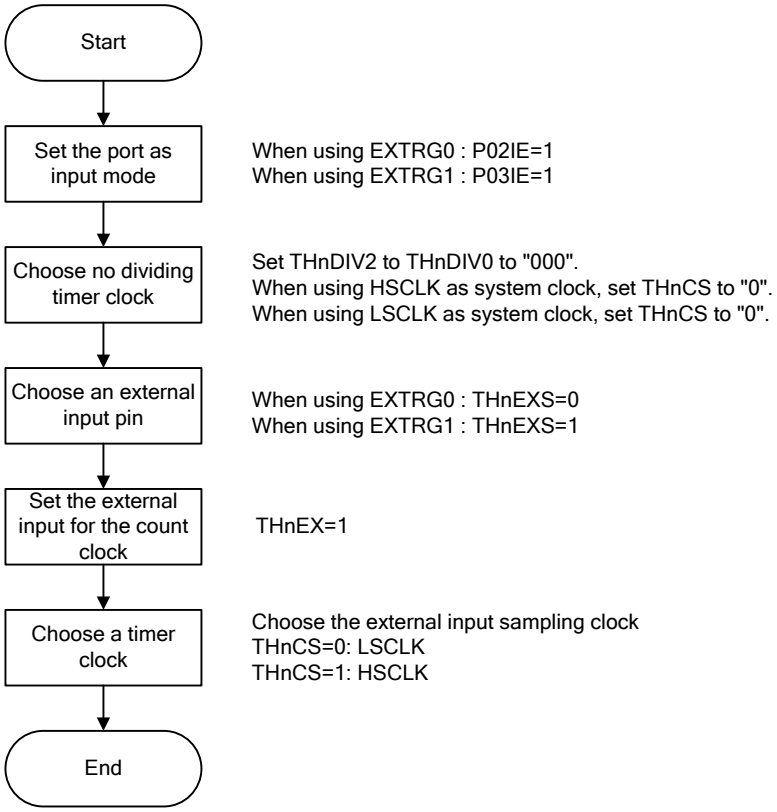


Figure 8-13 External Input Count Setting Flow

[Note]

- **The pulse with the width less than two clocks of the timer clock may be ignored. Always input the external input signal with the width equal to or more than two clocks of the timer clock.**
- **The external input signal (EXTRGn) which is input to the 16-bit timer is the signal that has passed the sampling controller of the external interrupt function. The sampling of the external interrupt function is optional. See Chapter 18 "External Interrupt Function" for details.**

Chapter 9 Functional Timer (FTM)

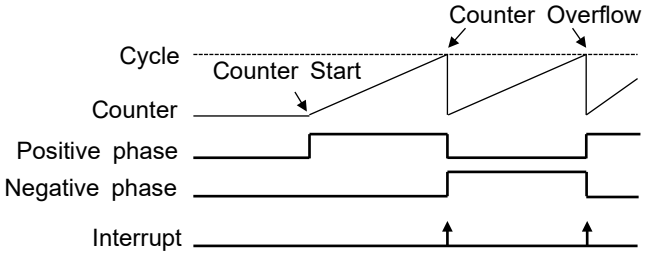
9. Functional Timer

9.1 General Description

The Functional timer enables following functions in four operation modes (TIMER/CAPTURE/PWM1/PWM2).

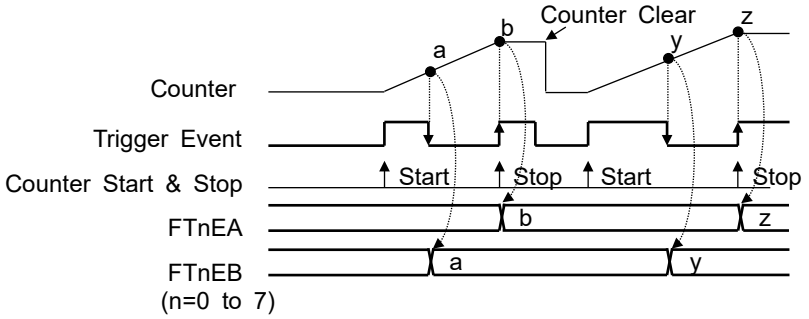
TIMER mode:

In this mode, the Functional Timer generates pulse signals, levels of which are reversed in sync with the counter start and the counter overflow. Also, it generates the interrupt when the counter overflows.



CAPTURE mode:

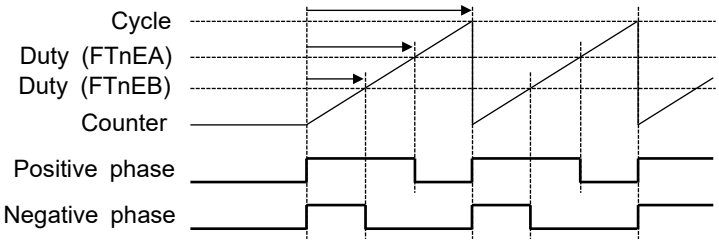
In this mode, the Functional Timer stores the value of counter into FTnEA register at the rising edge of a trigger event, into FTnEB register at the falling edge of a trigger event.



PWM1 mode:

In this mode, the Functional Timer generates two types of PWM waveform that have the same cycle and the start timing.

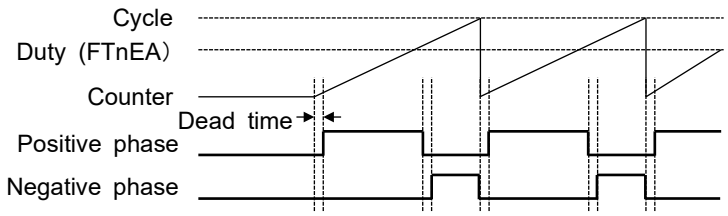
The setting value of FTnEA register makes the duty of the positive phase output and the setting value of FTnEB register makes the duty of the negative phase output.



PWM2 mode:

In this mode, the Functional Timer generates the complimentary PWM waveform of which the positive phase output and the negative phase output works exclusively. The setting of FTnEA register makes the duty of the positive phase output.

Also, a dead time can be configured by setting FTnDT register.



The number of Functional timer channels is dependent of the product specification.
Table 9-1 shows the number of channels.

Table 9-1 Number of Functional Timer channels

Channel no.	ML62Q1300 group				ML62Q1500 / ML62Q1800 / ML62Q1700 group				
	16pin product	20pin product	24pin product	32 pin product	48pin product	52pin product	64pin product	80pin product	100pin product
0	●	●	●	●	●	●	●	●	●
1	●	●	●	●	●	●	●	●	●
2	●	●	●	●	●	●	●	●	●
3	●	●	●	●	●	●	●	●	●
4	-	-	-	-	●	●	●	●	●
5	-	-	-	-	●	●	●	●	●
6	-	-	-	-	-	-	-	●	●
7	-	-	-	-	-	-	-	●	●

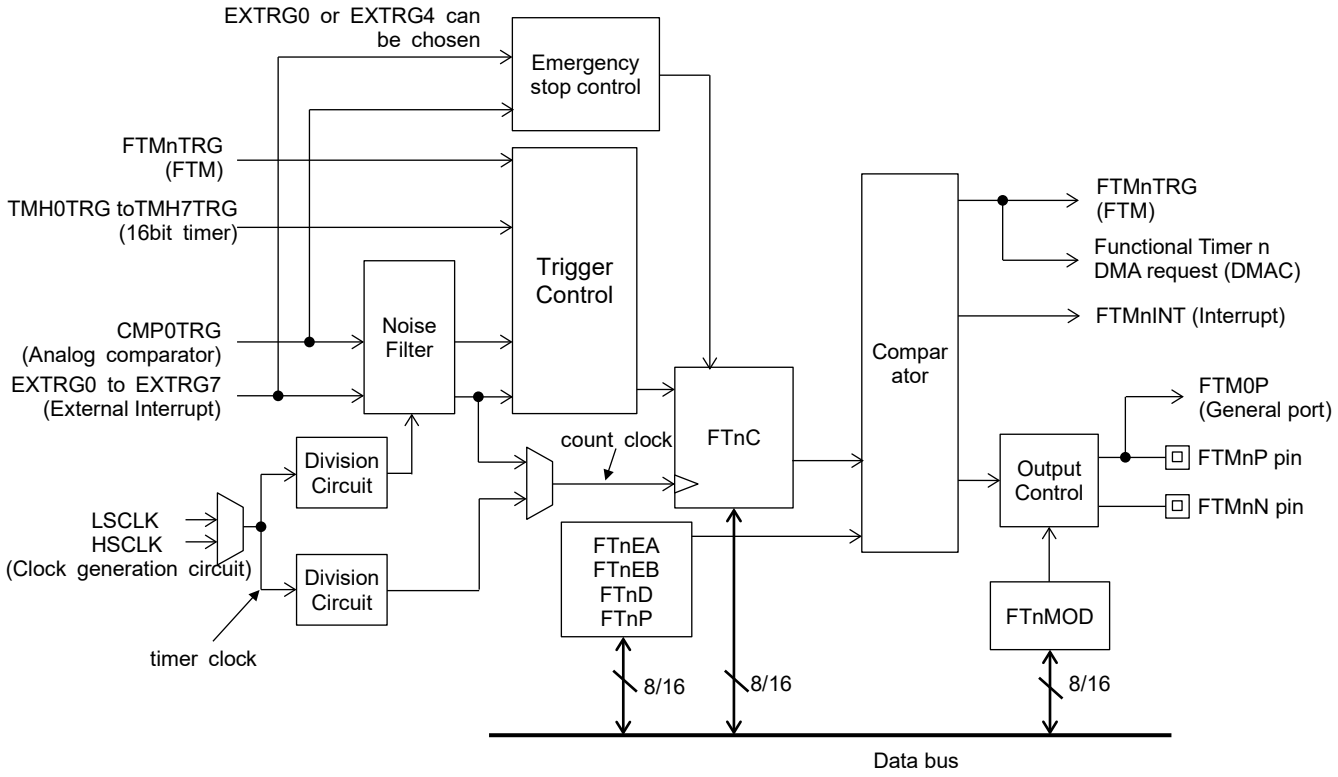
●: Available - : Unavailable

9.1.1 Features

- The Timer/Capture/PWM functions using the 16-bit counter
- The count clock can apply the LSCLK/HSCLK divided by 1 to 128 and the external clock input
- The timer output signal can be switched (Positive logic or Negative logic)
- Generate a cyclic interrupt, a duty interrupt and a coincident interrupt with the setting value
- One-shot mode
- Start/stop/clear the timer by an external trigger input or a timer interrupt request(event triggers)
- Emergency stop and emergency stop interrupt by an external trigger input
- Two types of PWM output with the same cycle and different duties, and complementary PWM output with the dead time
- Input signal duty/cycle measurement by the capture function
- Chosen interrupt source can be notified
- DMA request signal can be used

9.1.2 Configuration

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



FTnEA	: FTMn event A register
FTnEB	: FTMn event B register
FTnDT	: FTMn dead time register
FTnP	: FTMn cycle register
FTnC	: FTMn counter register
FTnMOD	: FTMn mode register
FTMnTRG	: Functional Timer n trigger
EXTRG0 to EXTRG 7	: External trigger n / external clock
CMP0TRG	: Analog comparator 0 trigger
TMH0TRG to TMH7TRG (n=0 to 7)	: 16-bit Timer n trigger

Figure 9-1 Configuration of the Functional Timer

9.1.3 List of Pins

The I/O pins of the Functional timer are assigned to the shared function of the general ports.

Pin name	I/O	Description
EXTRG0 to EXTRG7	I	External trigger 0 to 7 / External clock 0 to 7
FTMnP	O	Functional timer channel n output P
FTMnN	O	Functional timer channel n output N

(n=0 to 7)

Table 9-2 shows the list of the general ports used for the Functional timer and the register settings of the ports.

Table 9-2 Ports used in the Functional timer and the register settings

Channel no.	Pin name	Shared port		Register	Setting value	ML62Q1300 group				ML62Q1500/ML62Q1800/ ML62Q1700 group				
						16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
0 to 7	EXTRG0	P02	Primary Func.	P0MOD2	0000_0X01[*1]	●	●	●	●	●	●	●	●	●
	EXTRG1	P03		P0MOD3	0000_0X01[*1]	●	●	●	●	●	●	●	●	●
	EXTRG2	P04		P0MOD4	0000_0X01[*1]	●	●	●	●	●	●	●	●	●
	EXTRG3	P17		P1MOD7	0000_0X01[*1]	●	●	●	●	●	●	●	●	●
	EXTRG4	P21		P2MOD1	0000_0X01[*1]	●	●	●	●	●	●	●	●	●
	EXTRG5	P23		P2MOD3	0000_0X01[*1]	●	●	●	●	●	●	●	●	●
	EXTRG6	P26		P2MOD6	0000_0X01[*1]	●	●	●	●	●	●	●	●	●
	EXTRG7	P27		P2MOD7	0000_0X01[*1]	●	●	●	●	●	●	●	●	●
0	FTM0P	P02	5th Func.	P0MOD2	0100_XXXX[*2]	●	●	●	●	●	●	●	●	●
	FTM0N	P03	5th Func.	P0MOD3	0100_XXXX[*2]	●	●	●	●	●	●	●	●	●
1	FTM1P	P17	5th Func.	P1MOD7	0100_XXXX[*2]	●	●	●	●	●	●	●	●	●
		P47	5th Func.	P4MOD7	0100_XXXX[*2]	-	-	-	-	-	●[*3]	●[*3]	●[*3]	●[*3]
	FTM1N	P20	5th Func.	P2MOD0	0100_XXXX[*2]	●	●	●	●	●	●	●	●	●
		P46	5th Func.	P4MOD6	0100_XXXX[*2]	-	-	-	-	-	-	●[*3]	●[*3]	●[*3]
2	FTM2P	P21	5th Func.	P2MOD1	0100_XXXX[*2]	●	●	●	●	●	●	●	●	●
	FTM2N	P22	5th Func.	P2MOD2	0100_XXXX[*2]	●	●	●	●	●	●	●	●	●
3	FTM3P	P01	5th Func.	P0MOD1	0100_XXXX[*2]	-	-	-	-	●[*3]	●[*3]	●[*3]	●[*3]	●[*3]
		P26	5th Func.	P2MOD6	0100_XXXX[*2]	●	●	●	●	●	●	●	●	●
	FTM3N	P27	5th Func.	P2MOD7	0100_XXXX[*2]	●	●	●	●	●	●	●	●	●
		P44	5th Func.	P4MOD4	0100_XXXX[*2]	-	-	-	-	-	-	●[*3]	●[*3]	●[*3]
4	FTM4P	P63	5th Func.	P6MOD3	0100_XXXX[*2]	-	-	-	-	●	●	●	●	●
	FTM4N	P62	5th Func.	P6MOD2	0100_XXXX[*2]	-	-	-	-	●	●	●	●	●
5	FTM5P	P64	5th Func.	P6MOD4	0100_XXXX[*2]	-	-	-	-	●	●	●	●	●
	FTM5N	P65	5th Func.	P6MOD5	0100_XXXX[*2]	-	-	-	-	●	●	●	●	●
6	FTM6P	P93	5th Func.	P9MOD3	0100_XXXX[*2]	-	-	-	-	-	-	-	●	●
	FTM6N	P94	5th Func.	P9MOD4	0100_XXXX[*2]	-	-	-	-	-	-	-	●	●
7	FTM7P	P86	5th Func.	P8MOD6	0100_XXXX[*2]	-	-	-	-	-	-	-	-	●[*3]
		PA3	5th Func.	PAMOD3	0100_XXXX[*2]	-	-	-	-	-	-	-	●	●
	FTM7N	P87	5th Func.	P8MOD7	0100_XXXX[*2]	-	-	-	-	-	-	-	-	●[*3]
		PA4	5th Func.	PAMOD4	0100_XXXX[*2]	-	-	-	-	-	-	-	●	●

●: Available -: Unavailable ●[*3]: Available on ML62Q1700 group

[*1] : "X" determines the condition of the port input

X	Condition of the port input
0	Input (without an internal pull-up resistor)
1	Input (with an internal pull-up resistor)

[*2] : "XXXX" determines the condition of the port output

XXXX	Condition of the port output
0010	CMOS output
1010	Nch open drain output (without the pull-up)
1111	Nch open drain output (with the pull-up)

9.2 Description of Registers

9.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
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	FTM4 cycle register	FT4PL	FT4P	R/W	8/16	0xFF
0xF409		FT4PH		R/W	8	0xFF
0xF40A	FTM5 cycle register	FT5PL	FT5P	R/W	8/16	0xFF
0xF40B		FT5PH		R/W	8	0xFF
0xF40C	FTM6 cycle register	FT6PL	FT6P	R/W	8/16	0xFF
0xF40D		FT6PH		R/W	8	0xFF
0xF40E	FTM7 cycle register	FT7PL	FT7P	R/W	8/16	0xFF
0xF40F		FT7PH		R/W	8	0xFF
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	FTM4 event A register	FT4EAL	FT4EA	R/W	8/16	0x00
0xF419		FT4EAH		R/W	8	0x00
0xF41A	FTM5 event A register	FT5EAL	FT5EA	R/W	8/16	0x00
0xF41B		FT5EAH		R/W	8	0x00
0xF41C	FTM6 event A register	FT6EAL	FT6EA	R/W	8/16	0x00
0xF41D		FT6EAH		R/W	8	0x00
0xF41E	FTM7 event A register	FT7EAL	FT7EA	R/W	8/16	0x00
0xF41F		FT7EAH		R/W	8	0x00
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	FTM4 event B register	FT4EBL	FT4EB	R/W	8/16	0x00
0xF429		FT4EBH		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF42A	FTM5 event B register	FT5EBL	FT5EB	R/W	8/16	0x00
0xF42B		FT5EBH		R/W	8	0x00
0xF42C	FTM6 event B register	FT6EBL	FT6EB	R/W	8/16	0x00
0xF42D		FT6EBH		R/W	8	0x00
0xF42E	FTM7 event B register	FT7EBL	FT7EB	R/W	8/16	0x00
0xF42F		FT7EBH		R/W	8	0x00
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
0xF436	FTM3 dead time register	FT3DTL	FT3DT	R/W	8/16	0x00
0xF437		FT3DTH		R/W	8	0x00
0xF438	FTM4 dead time register	FT4DTL	FT4DT	R/W	8/16	0x00
0xF439		FT4DTH		R/W	8	0x00
0xF43A	FTM5 dead time register	FT5DTL	FT5DT	R/W	8/16	0x00
0xF43B		FT5DTH		R/W	8	0x00
0xF43C	FTM6 dead time register	FT6DTL	FT6DT	R/W	8/16	0x00
0xF43D		FT6DTH		R/W	8	0x00
0xF43E	FTM7 dead time register	FT7DTL	FT7DT	R/W	8/16	0x00
0xF43F		FT7DTH		R/W	8	0x00
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	FTM4 counter register	FT4CL	FT4C	R/W	8/16	0x00
0xF449		FT4CH		R/W	8	0x00
0xF44A	FTM5 counter register	FT5CL	FT5C	R/W	8/16	0x00
0xF44B		FT5CH		R/W	8	0x00
0xF44C	FTM6 counter register	FT6CL	FT6C	R/W	8/16	0x00
0xF44D		FT6CH		R/W	8	0x00
0xF44E	FTM7 counter register	FT7CL	FT7C	R/W	8/16	0x00
0xF44F		FT7CH		R/W	8	0x00
0xF450	FTM0 status register	FT0STAT	-	R/W	8	0x30
0xF451	Reserved	-	-	-	-	-
0xF452	FTM1 status register	FT1STAT	-	R/W	8	0x30
0xF453	Reserved	-	-	-	-	-
0xF454	FTM2 status register	FT2STAT	-	R/W	8	0x30
0xF455	Reserved	-	-	-	-	-
0xF456	FTM3 status register	FT3STAT	-	R/W	8	0x30
0xF457	Reserved	-	-	-	-	-

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF458	FTM4 status register	FT4STAT	-	R/W	8	0x30
0xF459	Reserved	-	-	-	-	-
0xF45A	FTM5 status register	FT5STAT	-	R/W	8	0x30
0xF45B	Reserved	-	-	-	-	-
0xF45C	FTM6 status register	FT6STAT	-	R/W	8	0x30
0xF45D	Reserved	-	-	-	-	-
0xF45E	FTM7 status register	FT7STAT	-	R/W	8	0x30
0xF45F	Reserved	-	-	-	-	-
0xF460	FTM0 mode register	FT0MODL	FT0MOD	R/W	8/16	0x00
0xF461		FT0MODH		R/W	8	0x40
0xF462	FTM1 mode register	FT1MODL	FT1MOD	R/W	8/16	0x00
0xF463		FT1MODH		R/W	8	0x40
0xF464	FTM2 mode register	FT2MODL	FT2MOD	R/W	8/16	0x00
0xF465		FT2MODH		R/W	8	0x40
0xF466	FTM3 mode register	FT3MODL	FT3MOD	R/W	8/16	0x00
0xF467		FT3MODH		R/W	8	0x40
0xF468	FTM4 mode register	FT4MODL	FT4MOD	R/W	8/16	0x00
0xF469		FT4MODH		R/W	8	0x40
0xF46A	FTM5 mode register	FT5MODL	FT5MOD	R/W	8/16	0x00
0xF46B		FT5MODH		R/W	8	0x40
0xF46C	FTM6 mode register	FT6MODL	FT6MOD	R/W	8/16	0x00
0xF46D		FT6MODH		R/W	8	0x40
0xF46E	FTM7 mode register	FT7MODL	FT7MOD	R/W	8/16	0x00
0xF46F		FT7MODH		R/W	8	0x40
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	FTM4 clock register	FT4CLKL	FT4CLK	R/W	8/16	0x00
0xF479		FT4CLKH		R/W	8	0x00
0xF47A	FTM5 clock register	FT5CLKL	FT5CLK	R/W	8/16	0x00
0xF47B		FT5CLKH		R/W	8	0x00
0xF47C	FTM6 clock register	FT6CLKL	FT6CLK	R/W	8/16	0x00
0xF47D		FT6CLKH		R/W	8	0x00
0xF47E	FTM7 clock register	FT7CLKL	FT7CLK	R/W	8/16	0x00
0xF47F		FT7CLKH		R/W	8	0x00
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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF486	FTM3 trigger register 0	FT3TRG0L	FT3TRG0	R/W	8/16	0x00
0xF487		FT3TRG0H		R/W	8	0x00
0xF488	FTM4 trigger register 0	FT4TRG0L	FT4TRG0	R/W	8/16	0x00
0xF489		FT4TRG0H		R/W	8	0x00
0xF48A	FTM5 trigger register 0	FT5TRG0L	FT5TRG0	R/W	8/16	0x00
0xF48B		FT5TRG0H		R/W	8	0x00
0xF48C	FTM6 trigger register 0	FT6TRG0L	FT6TRG0	R/W	8/16	0x00
0xF48D		FT6TRG0H		R/W	8	0x00
0xF48E	FTM7 trigger register 0	FT7TRG0L	FT7TRG0	R/W	8/16	0x00
0xF48F		FT7TRG0H		R/W	8	0x00
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	FTM4 trigger register 1	FT4TRG1L	FT4TRG1	R/W	8/16	0x00
0xF499		FT4TRG1H		R/W	8	0x00
0xF49A	FTM5 trigger register 1	FT5TRG1L	FT5TRG1	R/W	8/16	0x00
0xF49B		FT5TRG1H		R/W	8	0x00
0xF49C	FTM6 trigger register 1	FT6TRG1L	FT6TRG1	R/W	8/16	0x00
0xF49D		FT6TRG1H		R/W	8	0x00
0xF49E	FTM7 trigger register 1	FT7TRG1L	FT7TRG1	R/W	8/16	0x00
0xF49F		FT7TRG1H		R/W	8	0x00
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	FTM4 interrupt enable register	FT4INTEL	FT4INTE	R/W	8/16	0x00
0xF4A9		FT4INTEH		R/W	8	0x00
0xF4AA	FTM5 interrupt enable register	FT5INTEL	FT5INTE	R/W	8/16	0x00
0xF4AB		FT5INTEH		R/W	8	0x00
0xF4AC	FTM6 interrupt enable register	FT6INTEL	FT6INTE	R/W	8/16	0x00
0xF4AD		FT6INTEH		R/W	8	0x00
0xF4AE	FTM7 interrupt enable register	FT7INTEL	FT7INTE	R/W	8/16	0x00
0xF4AF		FT7INTEH		R/W	8	0x00
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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
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	FTM4 interrupt status register	FT4INTSL	FT4INTS	R	8/16	0x00
0xF4B9		FT4INTSH		R	8	0x00
0xF4BA	FTM5 interrupt status register	FT5INTSL	FT5INTS	R	8/16	0x00
0xF4BB		FT5INTSH		R	8	0x00
0xF4BC	FTM6 interrupt status register	FT6INTSL	FT6INTS	R	8/16	0x00
0xF4BD		FT6INTSH		R	8	0x00
0xF4BE	FTM7 interrupt status register	FT7INTSL	FT7INTS	R	8/16	0x00
0xF4BF		FT7INTSH		R	8	0x00
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	FTM4 interrupt clear register	FT4INTCL	FT4INTC	W	8/16	0x00
0xF4C9		FT4INTCH		W	8	0x00
0xF4CA	FTM5 interrupt clear register	FT5INTCL	FT5INTC	W	8/16	0x00
0xF4CB		FT5INTCH		W	8	0x00
0xF4CC	FTM6 interrupt clear register	FT6INTCL	FT6INTC	W	8/16	0x00
0xF4CD		FT6INTCH		W	8	0x00
0xF4CE	FTM7 interrupt clear register	FT7INTCL	FT7INTC	W	8/16	0x00
0xF4CF		FT7INTCH		W	8	0x00
0xF4F0	FTM common update register	FTCUD	-	W	8	0x00
0xF4F1	Reserved	-	-	-	-	-
0xF4F2	FTM common control register	FTCCONL	FTCCON	R/W	8/16	0x00
0xF4F3		FTCCONH		R/W	8	0x00
0xF4F4	FTM common start register	FTCSTRL	FTCSTR	W	8/16	0x00
0xF4F5		FTCSTRH		W	8	0x00
0xF4F6	FTM common stop register	FTCSTPL	FTCSTP	W	8/16	0x00
0xF4F7		FTCSTPH		W	8	0x00
0xF4F8	FTM common status register	FTCSTATL	FTCSTAT	R	8/16	0x00
0xF4F9		FTCSTATH		R	8	0x00

[Note]

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

9.2.2 FTMn Cycle Register (FTnP: n = 0 to 7)

FTnP is a special 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 FTnMD1 to 0 bits of FTnMOD register.

Address: 0xF400(FT0PL/FT0P), 0xF401(FT0PH), 0xF402(FT1PL/FT1P), 0xF403(FT1PH),
0xF404(FT2PL/FT2P), 0xF405(FT2PH), 0xF406(FT3PL/FT3P), 0xF407(FT3PH),
0xF408(FT4PL/FT4P), 0xF409(FT4PH), 0xF40A(FT5PL/FT5P), 0xF40B(FT5PH),
0xF40C(FT6PL/FT6P), 0xF40D(FT6PH), 0xF40E(FT7PL/FT7P), 0xF40F(FT7PH)
Access: R/W
Access size: 8/16 bit
Initial value: 0xFFFF

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

Bit No.	Bit symbol name	Description
15 to 0	FTnP15 to FTnP0	0x0001 to 0xFFFF: Set one cycle as the setting value in FTnP register + 1 clocks.

[Note]
- **When 0x0000 is written in this register, 0x0001 is set and the read value is also becomes 0x0001.**

9.2.3 FTMn Event A Register (FTnEA: n = 0 to 7)

FTnEA is a special function register (SFR) to set the event timing of FTMn or display the capture data.
Set this register after setting the operation mode using FTnMD1 to 0 bits of FTnMOD register.
In the CAPTURE mode, the FTnEA is a read-only register and it is invalid to write to this register.

Address: 0xF410(FT0EAL/FT0EA), 0xF411(FT0EAH), 0xF412(FT1EAL/FT1EA), 0xF413(FT1EAH), 0xF414(FT2EAL/FT2EA), 0xF415(FT2EAH), 0xF416(FT3EAL/FT3EA), 0xF417(FT3EAH), 0xF418(FT4EAL/FT4EA), 0xF419(FT4EAH), 0xF41A(FT5EAL/FT5EA), 0xF41B(FT5EAH), 0xF41C(FT6EAL/FT6EA), 0xF41D(FT6EAH), 0xF41E(FT7EAL/FT7EA), 0xF41F(FT7EAH)
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	FTnEA															
Byte	FTnEAH								FTnEAL							
Bit	FTnEA15	FTnEA14	FTnEA13	FTnEA12	FTnEA11	FTnEA10	FTnEA9	FTnEA8	FTnEA7	FTnEA6	FTnEA5	FTnEA4	FTnEA3	FTnEA2	FTnEA1	FTnEA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	FTnEA15 to FTnEA0	• TIMER mode 0x0000 to 0xFFFF: Set a count value to generate an interrupt. (the Interrupt timing is FTnEA setting value + 1) • CAPTURE mode 0x0000 to 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 mode 0x0000 to 0xFFFF: Set the duty of the positive phase output. The duty in the PWM cycle becomes [the value set in this register +1]. Duty 100% is configurable. • PWM2 mode 0x0000 to 0xFFFE: Set the duty of the positive phase output and the negative phase output. The duty in the PWM cycle becomes [the value set in this register +1].

[Note]
- **The data set in the FTnEA register must be less than that set in the FTnP register in the TIMER mode or PWM2 mode.**

9.2.4 FTMn Event B Register (FTnEB: n = 0 to 7)

FTnEB is a special function register (SFR) to set the event timing of FTMn or display the capture data.
Set this register after setting the operation mode using FTnMD1 to 0 bits of FTnMOD register.
In the CAPTURE mode, the FTnEB is a read-only register and it is invalid to write to this register.

Address: 0xF420(FT0EBL/FT0EB), 0xF421(FT0EBH), 0xF422(FT1EBL/FT1EB), 0xF423(FT1EBH), 0xF424(FT2EBL/FT2EB), 0xF425(FT2EBH), 0xF426(FT3EBL/FT3EB), 0xF427(FT3EBH), 0xF428(FT4EBL/FT4EB), 0xF429(FT4EBH), 0xF42A(FT5EBL/FT5EB), 0xF42B(FT5EBH), 0xF42C(FT6EBL/FT6EB), 0xF42D(FT6EBH), 0xF42E(FT7EBL/FT7EB), 0xF42F(FT7EBH)
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	FTnEB															
Byte	FTnEBH								FTnEBL							
Bit	FTnEB15	FTnEB14	FTnEB13	FTnEB12	FTnEB11	FTnEB10	FTnEB9	FTnEB8	FTnEB7	FTnEB6	FTnEB5	FTnEB4	FTnEB3	FTnEB2	FTnEB1	FTnEB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	FTnEB15 to FTnEB0	• TIMER mode 0x0000 to 0xFFFF: Set a count value to generate an interrupt. (Interrupt timing is FTnEB setting value + 1) • CAPTURE mode 0x0000 to 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 mode 0x0000 to 0xFFFF: Set the duty of the negative phase output. The duty in the PWM cycle becomes [the value set in this register +1]. Duty 100% is configurable. • PWM2 mode In this mode, set 0x0000 to FTnEB.

[Note]
- **The data set in the FTnEB register must be less than that set in the FTnP register in the TIMER mode.**

9.2.5 FTMn Dead Time Register (FTnDT: n = 0 to 7)

FTnDT is a special function register (SFR) to set the dead time of timer output.
Set this register after setting the operation mode using FTnMD1 to 0 bits of FTnMOD register.

Address: 0xF430(FT0DTL/FT0DT), 0xF431(FT0DTH), 0xF432(FT1DTL/FT1DT), 0xF433(FT1DTH), 0xF434(FT2DTL/FT2DT), 0xF435(FT2DTH), 0xF436(FT3DTL/FT3DT), 0xF437(FT3DTH), 0xF438(FT4DTL/FT4DT), 0xF439(FT4DTH), 0xF43A(FT5DTL/FT5DT), 0xF43B(FT5DTH), 0xF43C(FT6DTL/FT6DT), 0xF43D(FT6DTH), 0xF43E(FT7DTL/FT7DT), 0xF43F(FT7DTH)
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	FTnDT															
Byte	FTnDTH								FTnDTL							
Bit	FTnDT15	FTnDT14	FTnDT13	FTnDT12	FTnDT11	FTnDT10	FTnDT9	FTnDT8	FTnDT7	FTnDT6	FTnDT5	FTnDT4	FTnDT3	FTnDT2	FTnDT1	FTnDT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	FTnDT15 to FTnDT0	• TIMER, PWM1, PWM2 mode 0x0000 to 0xFFFF: Set the dead time of the positive phase output and negative phase output (Time of FTnDT setting value + 1). When the FTnDTENP bit /FTnDTENN bit of FTMn mode register (FTnMOD) is "1", it is enabled for the positive phase output/negative phase output respectively. • CAPTURE mode This register is disabled in the CAPTURE mode.

[Note]
- **The data set in the FTnDT register must be less than that set in the FTnEA register in the PWM2 mode.**
- **The sum of setting data in the FTnDT register and the FTnEA register must be less than that set in the FTnP register in the PWM2 mode.**

9.2.6 FTMn Counter Register (FTnC: n = 0 to 7)

FTnC is a special function register (SFR) to display the counter value of FTMn.
When writing to this register, the counter is cleared to "0x0000" in one clock of the timer clock.

Address: 0xF440(FT0CL/FT0C), 0xF441(FT0CH), 0xF442(FT1CL/FT1C), 0xF443(FT1CH), 0xF444(FT2CL/FT2C), 0xF445(FT2CH), 0xF446(FT3CL/FT3C), 0xF447(FT3CH), 0xF448(FT4CL/FT4C), 0xF449(FT4CH), 0xF44A(FT5CL/FT5C), 0xF44B(FT5CH), 0xF44C(FT6CL/FT6C), 0xF44D(FT6CH), 0xF44E(FT7CL/FT7C), 0xF44F(FT7CH)
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	FTnC															
Byte	FTnCH								FTnCL							
Bit	FTnC 15	FTnC 14	FTnC 13	FTnC 12	FTnC 11	FTnC 10	FTnC 9	FTnC 8	FTnC 7	FTnC 6	FTnC 5	FTnC 4	FTnC 3	FTnC 2	FTnC 1	FTnC 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.7 FTMn Status Register (FTnSTAT: n = 0 to 7)

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

Address: 0xF450(FT0STAT), 0xF452(FT1STAT),
0xF454(FT2STAT), 0xF456(FT3STAT),
0xF458(FT4STAT), 0xF45A(FT5STAT),
0xF45C(FT6STAT), 0xF45E(FT7STAT)
Access: R
Access size: 8bit
Initial value: 0x30

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FTnSTAT							
Bit	-	-	-	-	-	-	-	-	FTnSTA	FTnFLGC	FTnFLGB	FTnFLGA	-	-	-	FTnUD
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit No.	Bit symbol name	Description
7	FTnSTA	This bit is used to indicate the operation state of FTMn. 0: The counter is stopped (Initial value) 1: The counter is running
6	FTnFLGC	This bit is used to indicate whether the next event start is enable or disable while a counter chosen by FTnCST bit of FTnTRG0 register is being stopped. This bit is cleared by reading FTnC register in one clock of the timer clock. 0: Start by the event trigger is enabled (Initial value) 1: Start by the event trigger is disabled
5	FTnFLGB	This bit is used to indicate the state of event timing B of FTMn. • TIMER, PWM1, PWM2 mode 0: Counter value < Value of FTMn event B register 1: Counter value ≥ Value of FTMn event B register (Initial value) • CAPTURE mode 0: There is no capture data 1: There is a capture data (To be cleared by reading the FTnEB register)
4	FTnFLGA	This bit is used to indicate the state of event timing A of FTMn. • TIMER, PWM1, PWM2 mode 0: Counter value < Value of FTMn event A register 1: Counter value ≥ Value of FTMn event A register (Initial value) • CAPTURE mode 0: There is no capture data 1: There is a capture data (To be cleared by reading the FTnEA register)
3 to 1	-	Reserved bits
0	FTnUD	This bit is used to indicate the state of the completion after generating an update request of the FTnP register or the FTnEA/FTnEB/FTnDT register by writing "1" to FTCUDn bit of FTCUD register. When the transfer is completed, this bit is cleared automatically. 0: The update is completed (Initial value) 1: Requesting the update

9.2.8 FTMn Mode Register (FTnMOD: n = 0 to 7)

FTnMOD is a special function register (SFR) to set the FTMnP and FTMnN pin output function and the operation mode.

Address: 0xF460(FT0MODL/FT0MOD), 0xF461(FT0MODH),
0xF462(FT1MODL/FT1MOD), 0xF463(FT1MODH),
0xF464(FT2MODL/FT2MOD), 0xF465(FT2MODH),
0xF466(FT3MODL/FT3MOD), 0xF467(FT3MODH),
0xF468(FT4MODL/FT4MOD), 0xF469(FT4MODH),
0xF46A(FT5MODL/FT5MOD), 0xF46B(FT5MODH),
0xF46C(FT6MODL/FT6MOD), 0xF46D(FT6MODH),
0xF46E(FT7MODL/FT7MOD), 0xF46F(FT7MODH),
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	FTnMOD															
Byte	FTnMODH								FTnMODL							
Bit	FTnOSL1	FTnOSL0	FTnOSNN	FTnOSNP	rsvd	rsvd	rsvd	FTnSTPO	FTnOST	rsvd	FTnDTENN	FTnDTENP	rsvd	rsvd	FTnMD1	FTnMD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15, 14	FTnOSL1, FTnOSL0	These bits are used to choose the phase of signal output at FTMnN pin and FTMnP pin. FTMnN pin output / FTMnP pin output 00: Output Negative phase / Output Negative phase 01: Output Negative phase (Initial value) / Output Positive phase (Initial value) 10: Output Positive phase / Output Negative phase 11: Output Positive phase / Output Positive phase
13	FTnOSNN	This bit is used to reverse the FTMnN pin output signal. It reverses the output signal chosen by FTnOSL1 bit (bit15). 0: Does not reverse the output. (Initial value) 1: Reverses the output
12	FTnOSNP	This bit is used to reverse the FTMnP pin output signal. It reverses the output signal chosen by FTnOSL0 bit (bit14). 0: Does not reverse the output. (Initial value) 1: Reverses the output
11 to 9	rsvd	Reserved bit. Always write "0" to them.
8	FTnSTPO	This bit is used to set the output state of negative phase signal and the positive phase signal while the FTMn is stopped. • TIMER, PWM1, PWM2 mode 0: The output holds level "L" while the FTMn is stopped. (Initial value) When restarting the FTMn without clearing the counter, the output level is held until the next cycle. 1: The output holds the current level while the FTMn is stopped. When restarting the FTMn without clearing the counter, the output depends on the counter value. The output level becomes "L" when the counter is cleared while FTMn is stopped. • CAPTURE mode This bit is invalid

Bit No.	Bit symbol name	Description
7	FTnOST	This bit is used to set the repeat/one-shot mode of FTMn. • TIMER, PWM1, PWM2 mode 0: Repeat mode (Initial value) 1: One-shot mode • CAPTURE mode 0: Auto mode Even if the capture is performed once, data of the FTnEA and FTnEB register are overwritten (updated) when the next capture is performed. When the counter goes round, it restarts from 0. 1: Single mode Once captured into the FTnEA or FTnEB register, the next capture is not performed until reading the data. When the counter goes round, it stops.
6	rsvd	Reserved bit. Always write "0" to this.
5	FTnDTENN	This bit is used to enable the dead time of negative phase output. • TIMER, PWM1, PWM2 mode 0: Dead time is disabled (Initial value) 1: Dead time is enabled • CAPTURE mode This bit is invalid
4	FTnDTENP	This bit is used to enable the dead time of positive phase output. • TIMER, PWM1, PWM2 mode 0: Dead time is disabled (Initial value) 1: Dead time is enabled • CAPTURE mode This bit is invalid
3, 2	rsvd	Reserved bit. Always write "0" to them.
1, 0	FTnMD1, FTnMD0	These bits are used to choose the mode of FTMn. 00: TIMER mode (Initial value) 01: CAPTURE mode 10: PWM1 mode 11: PWM2 mode

9.2.9 FTMn Clock Register (FTnCLK: n=0 to 7)

FTnCLK is a special function register (SFR) to set the timer clock and count clock of the FTMn.
The timer clock is used for sampling the external trigger input and for detecting the edge of the external clock input.
The count clock is used for the count operation and control of the output waveform.

Address: 0xF470(FT0CLKL/FT0CLK), 0xF471(FT0CLKH),
0xF472(FT1CLKL/FT1CLK), 0xF473(FT1CLKH),
0xF474(FT2CLKL/FT2CLK), 0xF475(FT2CLKH),
0xF476(FT3CLKL/FT3CLK), 0xF477(FT3CLKH),
0xF478(FT4CLKL/FT4CLK), 0xF479(FT4CLKH),
0xF47A(FT5CLKL/FT5CLK), 0xF47B(FT5CLKH),
0xF47C(FT6CLKL/FT6CLK), 0xF47D(FT6CLKH),
0xF47E(FT7CLKL/FT7CLK), 0xF47F(FT7CLKH)
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	FTnCLK															
Byte	FTnCLKH								FTnCLKL							
Bit	rsvd	rsvd	rsvd	rsvd	rsvd	FTnXCK2	FTnXCK1	FTnXCK0	rsvd	FTnCKD2	FTnCKD1	FTnCKD0	FTnEX	rsvd	rsvd	FTnCK0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 11	rsvd	Reserved bit. Always write "0" to them.
10 to 8	FTnXCK2 to FTnXCK0	These bits are used to choose the external clock input used when the FTnEX bit is "1". 000: External clock 0 input (EXTRG0) (Initial value) 001: External clock 1 input (EXTRG1) 010: External clock 2 input (EXTRG2) 011: External clock 3 input (EXTRG3) 100: External clock 4 input (EXTRG4) 101: External clock 5 input (EXTRG5) 110: External clock 6 input (EXTRG6) 111: External clock 7 input (EXTRG7)
7	rsvd	Reserved bit. Always write "0" to this.
6 to 4	FTnCKD2 to FTnCKD0	These bits are used to choose the count clock in the FTMn. When the FTnEX bit is "1", this setting is invalid. 000: the timer clock (Initial value) 001: 1/2 of the timer clock 010: 1/4 of the timer clock 011: 1/8 of the timer clock 100: 1/16 of the timer clock 101: 1/32 of the timer clock 110: 1/64 of the timer clock 111: 1/128 of the timer clock
3	FTnEX	This bit is used to choose whether to use external triggers as the count clock in the FTMn. 0: Count clock chosen by FTnCKD2-0 bits. (Initial value) 1: Rising edge of the external triggers EXTRG0 to EXTRG7 chosen by FTnXCK2-0 bits.

Bit No.	Bit symbol name	Description
2, 1	-	Reserved bit. Always write "0" to them.
0	FTnCK0	This bit is used to choose the timer clock in the FTMn. 0: LSCLK (Initial value) 1: HSCLK

[Note]

- **Configure the external trigger using the timer clock without division on the condition that FTnCK0 “1” and the system clock is HSCLK or on the condition that FTnCK0 is “0” and the system clock is LSCLK. See section 9.3.7.1 and 9.3.7.2.**

9.2.10 FTMn Trigger Register 0 (FTnTRG0: n = 0 to 7)

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

Address: 0xF480(FT0TRG0L/FT0TRG0), 0xF481(FT0TRG0H),
0xF482(FT1TRG0L/FT1TRG0), 0xF483(FT1TRG0H),
0xF484(FT2TRG0L/FT2TRG0), 0xF485(FT2TRG0H),
0xF486(FT3TRG0L/FT3TRG0), 0xF487(FT3TRG0H),
0xF488(FT4TRG0L/FT4TRG0), 0xF489(FT4TRG0H),
0xF48A(FT5TRG0L/FT5TRG0), 0xF48B(FT5TRG0H),
0xF48C(FT6TRG0L/FT6TRG0), 0xF48D(FT6TRG0H),
0xF48E(FT7TRG0L/FT7TRG0), 0xF48F(FT7TRG0H)
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	FTnTRG0															
Byte	FTnTRG0H								FTnTRG0L							
Bit	rsvd	FTnEST1	FTnEST0	FTnSTSS	FTnSTS3	FTnSTS2	FTnSTS1	FTnSTS0	rsvd	FTnDCLH	FTnCST	rsvd	FTnSPC	FTnSP	FTnSTC	FTnST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	rsvd	Reserved bit. Always write "0" to this.
14, 13	FTnEST1, FTnEST0	These bits are used to choose the emergency stop trigger source of FTMn. They are enabled only when FTnEMGEN bit of FTCCON register is "1". 00: External trigger 0 input (EXTRG0) (initial value) 01: External trigger 4 input (EXTRG4) 10: Analog comparator 0 trigger (CMP0TRG) 11: Reserved

Bit No.	Bit symbol name	Description
12 to 8	FTnSTSS, FTnSTS3 to FTnSTS0	These bits are used to choose the trigger event source of FTMn. Choose a source except for the interrupt target (e.g. do not choose the FTM0TRG when using the functional timer FTM0). • TIMER, CAPTURE, PWM1, PWM2 mode 00000: External trigger 0 input (EXTRG0) (Initial value) 00001: External trigger 1 input (EXTRG1) 00010: External trigger 2 input (EXTRG2) 00011: External trigger 3 input (EXTRG3) 00100: External trigger 4 input (EXTRG4) 00101: External trigger 5 input (EXTRG5) 00110: External trigger 6 input (EXTRG6) 00111: External trigger 7 input (EXTRG7) 01000: Analog comparator 0 trigger (CMP0TRG) 010X1: Reserved 01100: For the clock mutual monitoring in the safety function See section "29.3.3 Clock Mutual Monitoring Function" 011X1: Reserved X: Don't care "0" or "1" • TIMER, PWM1, PWM2 mode 10000: 16-bit Timer 0 trigger (TMH0TRG) 10001: 16-bit Timer 1 trigger (TMH1TRG) 10010: 16-bit Timer 2 trigger (TMH2TRG) 10011: 16-bit Timer 3 trigger (TMH3TRG) 10100: 16-bit Timer 4 trigger (TMH4TRG) 10101: 16-bit Timer 5 trigger (TMH5TRG) 10110: 16-bit Timer 6 trigger (TMH6TRG) 10111: 16-bit Timer 7 trigger (TMH7TRG) 11000: Functional Timer 0 trigger (FTM0TRG) 11001: Functional Timer 1 trigger (FTM1TRG) 11010: Functional Timer 2 trigger (FTM2TRG) 11011: Functional Timer 3 trigger (FTM3TRG) 11100: Functional Timer 4 trigger (FTM4TRG) 11101: Functional Timer 5 trigger (FTM5TRG) 11110: Functional Timer 6 trigger (FTM6TRG) 11111: Functional Timer 7 trigger (FTM7TRG)
7	rsvd	Reserved bit. Always write "0" to this.
6	FTnDCLH	This bit is used to disable the counter clear by a trigger event when the output of FTMnP is "H" level before controlling the reverse output by FTnOSNP bit of FTnMOD register. • TIMER, PWM1, PWM2 mode 0: The counter clear is enabled regardless the positive phase output (initial value) 1: The counter clear is disabled when the positive phase output is "H" level. • CAPTURE mode This bit is invalid
5	FTnCST	This bit is used to choose the operation mode for starting the count by a trigger event. 0: A trigger event always can start the counter when the counter stops (except for emergency stop) (initial value) 1: A trigger event does not start the counter until reading FTnC register when the counter stops (except for emergency stop)
4	rsvd	Reserved bit. Always write "0" to this.

Bit No.	Bit symbol name	Description
3	FTnSPC	This bit is used to choose whether to enable clearing the counter when a trigger event for counter-stop occurs (only when the edge is chosen by the FTnTRM2-0 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 FTCUD register is generated when the trigger event occurs, the FTnP, FTnEA, FTnEB and FTnDT register gets updated at the same time as the counter clear. However, the counter is not cleared at emergency stop regardless the data of this bit. 0: The counter clear is disabled (Initial value) 1: The counter clear is enabled
2	FTnSP	This bit is used to choose whether to enable stopping the counter by a trigger event. 0: The counter stop is disabled (Initial value) 1: The counter stop is enabled
1	FTnSTC	This bit is used to choose whether to enable clearing the counter when a trigger event for counter-start occurs (only when the edge is chosen by the FTnTRM2-0 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 FTCUD register is generated when the trigger event occurs, the FTnP, FTnEA, FTnEB and FTnDT register gets updated at the same time as the counter clear. 0: The counter clear is disabled (Initial value) 1: The counter clear is enabled However, the counter is not cleared regardless of this bit in the following cases. ・When the emergency stop occurs. ・When Setting FTnTRM2-0 bits to “000” or “011” with EXTRG0-7 or CMP0TRG chosen as trigger event source. ・When Setting TMHnTRG or FTMnTRG as trigger event source.
0	FTnST	This bit is used to choose whether to enable starting the counter by a trigger event. 0: The counter start is disabled (Initial value) 1: The counter start is enabled

9.2.11 FTMn Trigger Register 1 (FTnTRG1: n = 0 to 7)

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

Address: 0xF490(FT0TRG1L/FT0TRG1), 0xF491(FT0TRG1H),
0xF492(FT1TRG1L/FT1TRG1), 0xF493(FT1TRG1H),
0xF494(FT2TRG1L/FT2TRG1), 0xF495(FT2TRG1H),
0xF496(FT3TRG1L/FT3TRG1), 0xF497(FT3TRG1H),
0xF498(FT4TRG1L/FT4TRG1), 0xF499(FT4TRG1H),
0xF49A(FT5TRG1L/FT5TRG1), 0xF49B(FT5TRG1H),
0xF49C(FT6TRG1L/FT6TRG1), 0xF49D(FT6TRG1H),
0xF49E(FT7TRG1L/FT7TRG1), 0xF49F(FT7TRG1H)
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	FTnTRG1															
Byte	FTnTRG1H								FTnTRG1L							
Bit	rsvd	rsvd	rsvd	rsvd	rsvd	FTnTRF2	FTnTRF1	FTnTRF0	rsvd	rsvd	rsvd	FTnEMGES	rsvd	FTnTRM2	FTnTRM1	FTnTRM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 11	rsvd	Reserved bit. Always write "0" to them.
10 to 8	FTnTRF2 to FTnTRF0	These bits are used to set the noise filter function for the external trigger or the external clock (EXTRG0 to EXTRG7 and CMP0TRG). It is invalid for other event trigger and the emergency stop trigger source. Set “1” to FTnETG bit to enable input to noise filter for trigger. Input signal: EXTRG0 to EXTRG7 or CMP0TRG are sampled by the timer clock. The following shows valid pulse width and invalid pulse width of the input signal. It is uncertain whether the other pulse width can be accepted or removed. Filter function, Filtering clock, Valid pulse width, Invalid pulse width 000: disabled, none, more than 1 cycle, none (Initial value) 001: enabled, 1/2 of timer clock, more than 4 cycles, less than 2 cycles 010: enabled, 1/4 of timer clock, more than 8 cycles, less than 4 cycles 011: enabled, 1/8 of timer clock, more than 16 cycles, less than 8 cycles 100: enabled, 1/16 of timer clock, more than 32 cycles, less than 16 cycles 101: enabled, 1/32 of timer clock, more than 64 cycles, less than 32 cycles 110: enabled, 1/64 of timer clock, more than 128 cycles, less than 64 cycles 111: enabled, 1/128 of timer clock, more than 256 cycles, less than 128 cycles Where ‘Valid pulse’ means signal that counter control (trigger control or FTnC) circuit can certainly accept, ‘Invalid pulse’ means signal that can be certainly removed by the noise filter. Also, in addition to the noise filter function set by these FTnTRF2-0 bits, another noise filter function is available by using the sampling function specified in the external interrupt mode register 0. See Chapter 18 "External Interrupt Function" for details about the external interrupt.
7 to 5	-	Reserved bit. Always write "0" to them.
4	FTnEMGES	This bit is used to choose the edge of the emergency stop trigger of FTMn. 0: Rising edge (initial value) 1: Falling edge
3	-	Reserved bit. Always write "0" to this.

Bit No.	Bit symbol name	Description
2 to 0	FTnTRM2 to FTnTRM0	These bits are used to choose the edge or the level of the trigger event of FTMn. These are enabled only when EXTRG0-7 or CMP0TRG is chosen for the event trigger source. In other cases, it is fixed to the rising edge. Counter start / Counter stop 000: Rising edge / Rising edge (Initial value) 001: Falling edge / Rising edge 010: Rising edge / Falling edge 011: Falling edge / Falling edge 1X0: "H" level / "L" level 1X1: "L" level / "H" level X: Don't care "0" or "1".

[Note]

- **When using the emergency stop trigger, use another filter function in each peripheral module.**
- **If a level setting is chosen for the condition of the counter start and condition is matched, the count operation continues (restart the count-up from 0) even if a stop condition is satisfied in the one-shot mode.**
- **When using the EXTRG0 to EXTRG7, enable the trigger event after setting the noise filter by the FTnTRG1 register. Otherwise, the trigger may occur immediately after setting the FTnTRG1 register.**

9.2.12 FTMn Interrupt Enable Register (FTnINTE: n = 0 to 7)

FTnINTE is a special 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 other channels of FTMn.
The trigger output is available to use as a DMA request signal.

Address: 0xF4A0(FT0INTEL/FT0INTE), 0xF4A1(FT0INTEH),
0xF4A2(FT1INTEL/FT1INTE), 0xF4A3(FT1INTEH),
0xF4A4(FT2INTEL/FT2INTE), 0xF4A5(FT2INTEH),
0xF4A6(FT3INTEL/FT3INTE), 0xF4A7(FT3INTEH),
0xF4A8(FT4INTEL/FT4INTE), 0xF4A9(FT4INTEH),
0xF4AA(FT5INTEL/FT5INTE), 0xF4AB(FT5INTEH),
0xF4AC(FT6INTEL/FT6INTE), 0xF4AD(FT6INTEH),
0xF4AE(FT7INTEL/FT7INTE), 0xF4AF(FT7INTEH)
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	FTnINTE															
Byte	FTnINTEH								FTnINTEL							
Bit	rsvd	rsvd	rsvd	rsvd	rsvd	FTnIOB	FTnIOA	FTnIOP	rsvd	rsvd	rsvd	FTnIETR	FTnIETS	FTnIEB	FTnIEA	FTnIEP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 11	rsvd	Reserved bit. Always write "0" to them.
10	FTnIOB	This bit is used to enable FTMnTRG output in event timing B of FTMn. When it's enabled, the FTMnTRG is output when the data of FTnC register and FTnEB register matched or a data is captured into the FTnEB register. 0: Disable the FTMnTRG output in the event timing B (Initial value) 1: Enable the FTMnTRG output in the event timing B
9	FTnIOA	This bit is used to enable FTMnTRG output in event timing A of FTMn. When it's enabled, the FTMnTRG is output when the data of FTnC register and FTnEA register matched or a data is captured into the FTnEA register. 0: Disable the FTMnTRG output in the event timing A (Initial value) 1: Enable the FTMnTRG output in the event timing A
8	FTnIOP	This bit is used to enable FTMnTRG output related to the FTnP register. When it's enabled, the FTMnTRG is output when the data of FTnC register and FTnP register matched. 0: Disable the FTMnTRG output when the data of FTMn cycle register matched (Initial value) 1: Enable the FTMnTRG output when the data of FTMn cycle register matched
7 to 5	-	Reserved bit. Always write "0" to them.
4	FTnIETR	This bit is used to enable the trigger counter start interrupt of FTMn. 0: Trigger counter start interrupt is disabled (Initial value) 1: Trigger counter start interrupt is enabled
3	FTnIETS	This bit is used to enable the trigger counter stop interrupt of FTMn. 0: Trigger counter stop interrupt is disabled (Initial value) 1: Trigger counter stop interrupt is enabled

Bit No.	Bit symbol name	Description
2	FTnIEB	This bit is used to enable the event timing B interrupt (in TIMER and PWM1 mode) or the capture B interrupt (in CAPTURE mode). • TIMER, PWM1 mode 0: Event timing B interrupt is disabled (Initial value) 1: Event timing B interrupt is enabled • PWM2 mode 0: Write always "0" 1: Do not set • CAPTURE mode 0: Capture B interrupt is disabled 1: Capture B interrupt is enabled
1	FTnIEA	This bit is used to enable the event timing A interrupt (in TIMER, PWM1 and PWM2 mode) or the capture A interrupt (in CAPTURE mode). • TIMER, PWM1, PWM2 mode 0: Event timing A interrupt is disabled (Initial value) 1: Event timing A interrupt is enabled • CAPTURE mode 0: Capture A interrupt is disabled 1: Capture A interrupt is enabled
0	FTnIEP	This bit is used to enable the cyclic interrupt. 0: Cyclic interrupt is disabled (Initial value) 1: Cyclic interrupt is enabled

9.2.13 FTMn Interrupt Status Register (FTnINTS: n = 0 to 7)

FTnINTS is a special function register (SFR) to indicate the interrupt status of FTMn. The FTnINTS is a read-only register.
The bit 5 to bit 0 is reset to "0" by writing "1" to the same number of bit in the MCINTCL register.

Address: 0xF4B0(FT0INTSL/FT0INTS), 0xF4B1(FT0INTSH),
0xF4B2(FT1INTSL/FT1INTS), 0xF4B3(FT1INTSH),
0xF4B4(FT2INTSL/FT2INTS), 0xF4B5(FT2INTSH),
0xF4B6(FT3INTSL/FT3INTS), 0xF4B7(FT3INTSH),
0xF4B8(FT4INTSL/FT4INTS), 0xF4B9(FT4INTSH),
0xF4BA(FT5INTSL/FT5INTS), 0xF4BB(FT5INTSH),
0xF4BC(FT6INTSL/FT6INTS), 0xF4BD(FT6INTSH),
0xF4BE(FT7INTSL/FT7INTS), 0xF4BF(FT7INTSH)
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	FTnINTS															
Byte	FTnINTSH								FTnINTSL							
Bit	-	-	-	-	-	-	-	-	-	-	FTnISES	FTnISTR	FTnISTS	FTnISB	FTnISA	FTnISP
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 6	-	Reserved bits
5	FTnISES	This bit is used to indicate the state of the emergency stop interrupt of FTMn. 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 FTnINTC register.
4	FTnISTR	This bit is used to indicate the state of the trigger counter start interrupt of FTMn. 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 FTnINTC register.
3	FTnISTS	This is a bit to indicate the state of the trigger counter stop interrupt of FTMn. 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 FTnINTC register.
2	FTnISB	This is a bit to indicate the state of the event timing B interrupt of FTMn. In CAPTURE mode, it indicates the status of storing the capture data into the FTnEB register. • TIMER, PWM1, PWM2 mode 0: Event timing B interrupt has not occurred (Initial value) 1: Event timing B interrupt has occurred This bit is cleared by writing "1" to FTnICB bit of FTnINTC register. • CAPTURE mode 0: Capture B interrupt has not occurred 1: Capture B interrupt has occurred Indicates that the captured data is stored to the FTnEB register. This bit is cleared by writing "1" to FTnICB bit of FTnINTC register or by reading the FTnEB register.

Bit No.	Bit symbol name	Description
1	FTnISA	This is a bit to indicate the state of the event timing A interrupt of FTMn. In CAPTURE mode, it indicates the status of storing the capture data into the FTnEA register. • TIMER, PWM1, PWM2 mode 0: Event timing A interrupt has not occurred (initial value) 1: Event timing A interrupt has occurred This bit is cleared by writing "1" to FTnICA bit of FTnINTC register. • CAPTURE mode 0: Capture A interrupt has not occurred 1: Capture A interrupt has occurred This bit is cleared by writing "1" to FTnICA bit of FTnINTC register or by reading the FTnEA register.
0	FTnISP	This is a bit to indicate the state of the cyclic interrupt of FTMn. 0: Cyclic interrupt has not occurred (initial value) 1: Cyclic interrupt has occurred This bit is cleared by writing "1" to FTnICP bit of FTnINTC register.

[Note]

- **No interrupt request is issued if the interrupt status bit is "1" and the same interrupt occurs again. To issue an interrupt request, write "1" to the same bit in the FTnINTC register and clear the status bit to "0".**

9.2.14 FTMn Interrupt Clear Register (FTnINTC: n = 0 to 7)

FTnINTC is a special function register (SFR) to clear the interrupt status of FTMn. If the bit 5 to bit 0 is set to "1", the interrupt request indicated by the same number of bit in the FTnINTS register gets cleared. The FTnINTC always returns 0x0000 for reading.

Address: 0xF4C0(FT0INTCL/FT0INTC), 0xF4C1(FT0INTCH),
0xF4C2(FT1INTCL/FT1INTC), 0xF4C3(FT1INTCH),
0xF4C4(FT2INTCL/FT2INTC), 0xF4C5(FT2INTCH),
0xF4C6(FT3INTCL/FT3INTC), 0xF4C7(FT3INTCH),
0xF4C8(FT4INTCL/FT4INTC), 0xF4C9(FT4INTCH),
0xF4CA(FT5INTCL/FT5INTC), 0xF4CB(FT5INTCH),
0xF4CC(FT6INTCL/FT6INTC), 0xF4CD(FT6INTCH),
0xF4CE(FT7INTCL/FT7INTC), 0xF4CF(FT7INTCH)
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	FTnINTC															
Byte	FTnINTCH								FTnINTCL							
Bit	-	-	-	-	-	-	-	-	FTnIR	-	FTnICES	FTnICTR	FTnICTS	FTnICB	FTnICA	FTnICP
R/W	R	R	R	R	R	R	R	R	W	R	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bits
7	FTnIR	An interrupt request bit of FTMn. Write "1" to this bit at the end of an interrupt routine. Writing "0": Invalid Writing "1": If there is any unhandled interrupt source, the interrupt request is generated again.
6	-	Reserved bit
5	FTnICES	This bit is used to clear the status of the emergency stop interrupt. Writing "0": Invalid Writing "1": Clear the emergency stop interrupt.
4	FTnICTR	This bit is used to clear the status of the trigger counter start interrupt. Writing "0": Invalid Writing "1": Clear the trigger counter start interrupt.
3	FTnICTS	This bit is used to clear the status of the trigger counter stop interrupt. Writing "0": Invalid Writing "1": Clear the trigger counter stop interrupt.
2	FTnICB	This bit is used to clear the status of the event timing B interrupt. Writing "0": Invalid Writing "1": Clear the event timing B interrupt.
1	FTnICA	This bit is used to clear the status of the event timing A interrupt. Writing "0": Invalid Writing "1": Clear the event timing A interrupt.
0	FTnICP	This bit is used to clear the status of the cyclic interrupt. Writing "0": Invalid Writing "1": Clear the status of the cyclic interrupt.

[Note]

- **Write to FTnIR bit by word access. It is invalid to write to the FTnIR bit by byte access or bit access.**
- **It is not able to write to FTnIR bit using the debug tools (ex. The SFR window or watch window function, etc).**
- **Confirm that there is no unhandled interrupt before stopping FTM. The interrupt status is not cleared when you stop FTM while there are some unhandled interrupts.**

9.2.15 FTM Common Update Register (FTCUD)

FTCUD is a special function register (SFR) to update FTnP, FTnEA, FTnEB and FTnDT registers while they are running.
The FTCUD is a common SFR to each channel. The bit n corresponds to channel n.
It is unavailable to write to the bits for unequipped channels.

Address: 0xF4F0(FTCUD)
Access: W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FTCUD							
Bit	-	-	-	-	-	-	-	-	FTCUD7	FTCUD6	FTCUD5	FTCUD4	FTCUD3	FTCUD2	FTCUD1	FTCUD0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	FTCUD7 to FTCUD0	These are write-only bits to update FTnP, FTnEA, FTnEB and FTnDT registers while they are running. After setting the FTnP, FTnEA, FTnEB and FTnDT registers, the setting value is transferred to the internal buffers of FTnP, FTnEA, FTnEB and FTnDT at the end of the cycle by writing "1" to the corresponding bit for the FTMn. Writing "0": Invalid Writing "1": Request the update

9.2.16 FTM Common Control Register (FTCCON)

FTCCON is a special function register (SFR) to set the function of FTMn.
The FTCCON is a common SFR to each channel. The bit n corresponds to channel n.
It is unavailable to write to the bits for unequipped channels.

Address: 0xF4F2(FTCCONL/FTCCON), 0xF4F3(FTCCONH)
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	FTCCON															
Byte	FTCCONH								FTCCONL							
Bit	FT7SDN	FT6SDN	FT5SDN	FT4SDN	FT3SDN	FT2SDN	FT1SDN	FT0SDN	FT7EMGEN	FT6EMGEN	FT5EMGEN	FT4EMGEN	FT3EMGEN	FT2EMGEN	FT1EMGEN	FT0EMGEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	FT7SDN to FT0SDN	This bit is used to enable controlling the positive phase/negative phase output. • TIMER, PWM1, PWM2 mode 0: Enable the output (Initial value) 1: Disable the output (The output is fixed to "L" level) • CAPTURE mode This bit is invalid.
7 to 0	FT7EMGEN to FT0EMGEN	This bit is used to enable the emergency stop on the FTMn. • TIMER, PWM1, PWM2 mode 0: Disable the emergency stop (Initial value) 1: Enable the emergency stop • CAPTURE mode This bit is invalid.

9.2.17 FTM Common Start Register (FTCSTR)

FTCSTR is a special function register (SFR) to set the function of FTMn.
This is an SFR common to each channel. Bit n corresponds to channel n.
The FTCSTR is a common SFR to each channel. The bit n corresponds to channel n.
It is unavailable to write to the bits for unequipped channels.

Address: 0xF4F4(FTCSTRL/FTCSTR), 0xF4F5(FTCSTRH)
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	FTCSTR															
Byte	FTCSTRH								FTCSTRL							
Bit	FT7ETG	FT6ETG	FT5ETG	FT4ETG	FT3ETG	FT2ETG	FT1ETG	FT0ETG	FT7STR	FT6STR	FT5STR	FT4STR	FT3STR	FT2STR	FT1STR	FT0STR
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	FT7ETG to FT0ETG	These bits are used to enable trigger operation; counting stop/start by a trigger event and input to noise filter for trigger. Control by the FTCSTP register to disable it. For clearing the counter by the trigger event, control it by FTnSTC bit and FTnSPC bit of FTnTRG0 register. Trigger operation is disabled in the initial state at the power-on. Writing "0": Invalid Writing "1": Trigger operation is enabled
7 to 0	FT7STR to FT0STR	These bits are used to start counting the FTMn by the software. When "1" is written in these bits, the count starts. In the initial state at the power-on, the counting is stopped. Writing "0": Invalid Writing "1": Counting is started by the software

[Note]
- **Set the FTnSTR when the FTMn stops (FTnSTA bit of FTnSTAT register is "0").**

9.2.18 FTM Common Stop Register (FTCSTP)

FTCSTP is a special function register (SFR) to set the function of FTMn.
The FTCSTP is a common SFR to each channel. The bit n corresponds to channel n.
It is unavailable to write to the bits for unequipped channels.

Address: 0xF4F6(FTCSTPL/FTCSTP), 0xF4F7(FTCSTPH)
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	FTCSTP															
Byte	FTCSTPH								FTCSTPL							
Bit	FT7DTG	FT6DTG	FT5DTG	FT4DTG	FT3DTG	FT2DTG	FT1DTG	FT0DTG	FT7STP	FT6STP	FT5STP	FT4STP	FT3STP	FT2STP	FT1STP	FT0STP
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	FT7DTG to FT0DTG	These bits are used to disable trigger operation, counting stop/start by a trigger event and input to noise filter for trigger. Control by the FTCSTR register to enable it. Trigger operation is enabled in the initial state at the power-on. Writing "0": Invalid Writing "1": Trigger operation is disabled. The counting is stopped when it is operated b by a trigger event
7 to 0	FT7STP to FT0STP	These bits are used to stop counting the FTMn by the software. When "1" is written in these bits, the count stops. In the initial state at the power-on, the counting is stopped. Writing "0": Invalid Writing "1": Counting is stopped by the software

[Note]
- **Set the FTnSTP bits while the FTMn is operating (FTnSTA bit of FTnSTAT register is "1").**
- **To stop counting during one-shot mode with HSCLK is selected for the system clock and LSCLK (FTnCK0 bit = 0 in the FTnCLK register) is selected for the timer clock, set the FTnSTP bit to "1", and then change the timer clock to HSCLK (FTnCK0 bit = 1 in the FTnCLK register), and set the timer clock to LSCLK again (FTnCK0 bit = 0 in the FTnCLK register).**

9.2.19 FTM Common Status Register (FTCSTAT)

FTCSTAT is a special function register (SFR) to indicate the state of FTMn.
The FTCSTAT is a common SFR to each channel. The bit n corresponds to channel n.

Address: 0xF4F8(FTCSTATL/FTCSTAT), 0xF4F9(FTCSTATH)
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	FTCSTAT															
Byte	FTCSTATH								FTCSTATL							
Bit	FT7TGEN	FT6TGEN	FT5TGEN	FT4TGEN	FT3TGEN	FT2TGEN	FT1TGEN	FT0TGEN	FT7RUN	FT6RUN	FT5RUN	FT4RUN	FT3RUN	FT2RUN	FT1RUN	FT0RUN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	FT7TGEN to FT0TGEN	These bits are used to check the setting status of FTMn. The trigger operation is disabled in the initial state at the power-on. 0: Trigger operation (counting stop/start/input to noise filter for trigger.) is disabled (initial value) 1: Trigger operation (counting stop/start/input to noise filter for trigger.) is enabled
7 to 0	FT7RUN to FT0RUN	These bits are used to indicate the counting status of FTMn. This bit indicates the same information as FTnSTA bit. In the initial state at the power-on, counting is stopped. 0: Counting is stopped (initial value) 1: Counting is in progress

9.3 Description of Operation

Four types of operation modes are available for the functional timer:

- TIMER mode
- CAPTURE mode
- PWM1 mode
- PWM2 mode

9.3.1 Common Sequence (Initial setting Common to All Modes)

FTMn starts operating by setting the FTCSTR register after the setting steps from 1 to 6 below.
During operation, the hardware states such as interrupt status can be checked and the cycle/event settings are updateable.

1: Mode setting (FTnMOD register)
Choose the TIMER/CAPTURE/PWM1/PWM2 mode using the FTnMOD register.
In addition, set the repeat mode/one-shot mode.

2: Clock setting (FTnCLK register)
Choose the timer clock and the count clock.
If the internal clock is chosen for the count clock, the frequency dividing ratio can also be set.

3: Trigger setting (FTnTRG0 register, FTnTRG1 register)
Use this setting when starting/stopping the counter by an event trigger.
In the FTnTRG0 register, choose the event trigger source and the action. In the FTnTRG1 register, choose the edge of the event trigger/emergency stop.

4: Interrupt setting (FTnINTE register)
Set the interrupt source.
Choose from cycle/event (counter coincidence, duty, capture) and trigger start/stop interrupt.

5: Cycle/event setting (FTnP register, FTnEA register, FTnEB register, FTnDT register)
Set the cycle, data for counter coincidence, duty, dead time, etc.

	TIMER mode	CAPTURE mode	PWM1 mode	PWM2 mode
FTnP register	Cycle in repeat mode or timeout period in one-shot mode			
FTnEA register	Coincident interrupt setting value	(Capturing data)	Positive phase output duty	Duty
FTnEB register	Coincident interrupt setting value		Negative phase output duty	(Unused)
FTnDT register	Dead time for output	(Unused)	Dead time for output	Dead time for output

The cycle is calculated as follows:

$$T_{cycle} = \frac{FTnP + 1}{\text{The count clock frequency [Hz]}} \quad (FTnP: 0x0001 \text{ to } 0xFFFF)$$

6: Choice of the external output signal

FTnOSL1 and FTnOSL0 bits of FTnMOD register are used to choose the positive output or negative phase output driven to the FTMnP pin or FTMnN pin.
FTnOSNP bit is used to determine if reversing the signal driven to the FTMnP pin.
FTnOSNN bit is used to determine if reversing the signal driven to the FTMnN pin.

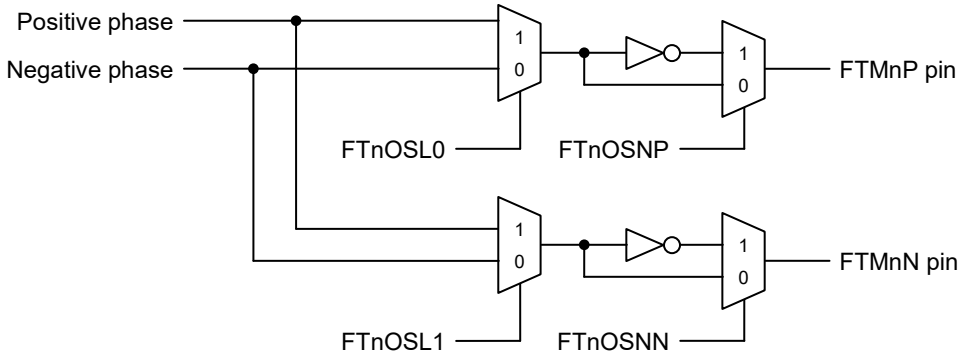


Figure 9-2 Configuration of the external output signal control

7: Control start/stop

Allow the software start, or event trigger reception, emergency stop setting.
The counter operates at the rising edge of the count clock.
Since the software start/stop is synchronized with the count clock, the FTnSTA bit becomes "1" at the start after one cycle of the count clock. After two cycles, the counter operation starts.
When the operation is stopped, the count operation stops after one cycle of the count clock and the FTnSTA bit becomes "0". Then the count value is maintained.
If started again, it restarts after one cycle.
If clearing the counter, write an arbitrary value to the FTnC register.

8: Operation process in progress (FTnSTAT register, FTCSTAT register, FTnINTS register, FTCUD register, FTCCON register)

The state under operation can be checked by the FTnSTAT, FTCSTAT, and FTnINTS registers.
To change the waveform of PWM, etc., set the applicable bit of the FTMUD register after setting the cycle/event. The waveform will be updated in the next cycle.
In addition, setting the FTnSDN bit of the FTCCON register forces the output to be fixed to "L" level.

9.3.2 Counter Operation (Common to All Modes)

The operation of FTM's internal counter is common to each mode.
It counts up to the setting value of the FTnP register.
In the repeat mode (the FTnOST bit of the FTnMOD register is "0"), the counter is cleared at the time of overflow, then continues the counting operation again.
In the one-shot mode (the FTnOST bit of the FTnMOD register is "1"), the counter is cleared at the time of overflow, and then stops the counting operation.
Starting/stopping the counting operation can be executed through the software or a trigger event.

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

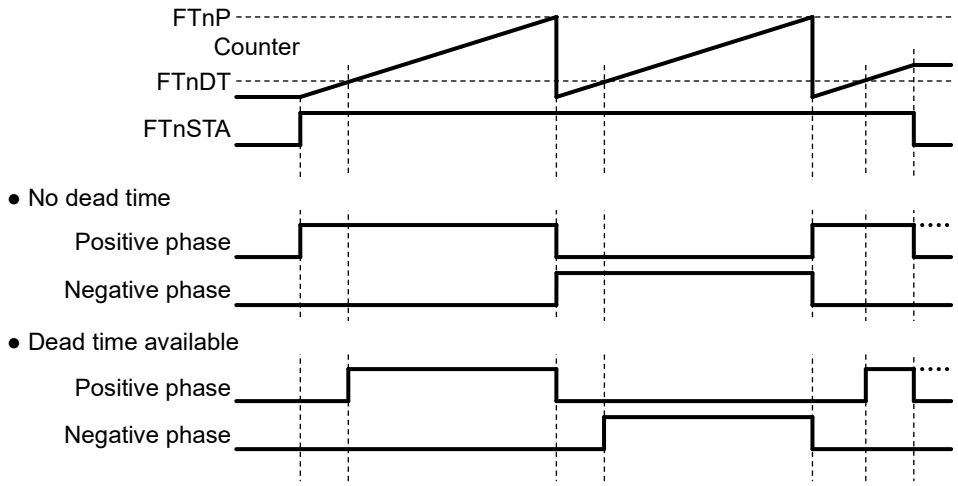
Writing "1" to the FTnETG bit of the FTCSTR register enables the counter operation to be controlled by triggers.
Trigger choice, etc. can be executed through the configuration of FTnTRG0 and FTnTRG1 registers.
The source of a trigger event can be chosen from EXTRG0 to EXTRG7, CMP0TRG, TMH0TRG to TMH7TRG, or FTM0TRG to FTM7TRG.
Depending on the chosen trigger event, an operation (counter start, counter stop, counter start/stop and counter clearing) can be chosen.

9.3.3 TIMER Mode Operation

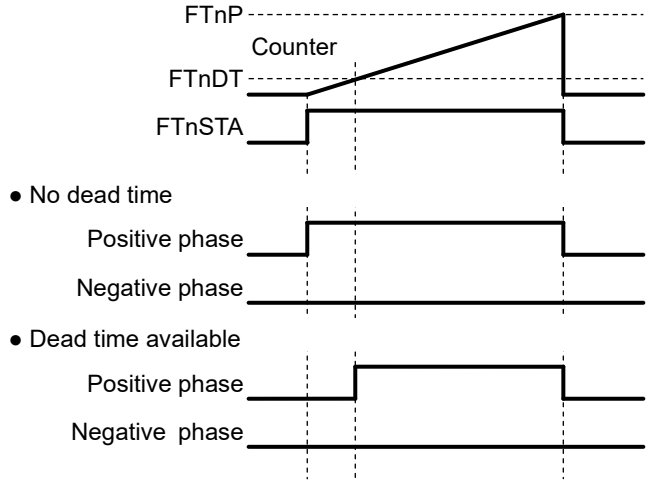
The TIMER mode controls the interrupt generation and output signal using the counter overflow.

9.3.3.1 Output Waveform in TIMER Mode

The timer output has two kinds of phase, the positive phase and the negative phase.
When writing "1" to the FTnSTR bit of the FTCSTR register with the counter set to "0x0000", the positive phase output starts with "H" level and the negative phase output starts with "L" level.
Also, dead time is enabled when writing "1" to the FTnDTENP or FTnDTENN bits of the FTnMOD register, and the output remains "L" level until the count set in the FTnDT register has elapses since start of counting.
In the repeat mode, the output repeats to toggle the signal level synchronizing with the start of count and the overflow.
In the one-shot mode, the positive output remains "H" level for one cycle of the timer and then the count stops. The negative output is fixed to "L".
Figure 9-3 (a) shows waveforms of the positive phase/negative phase output in the repeat mode.
Figure 9-3 (b) shows waveforms of the positive phase/negative phase output in the one-shot mode.



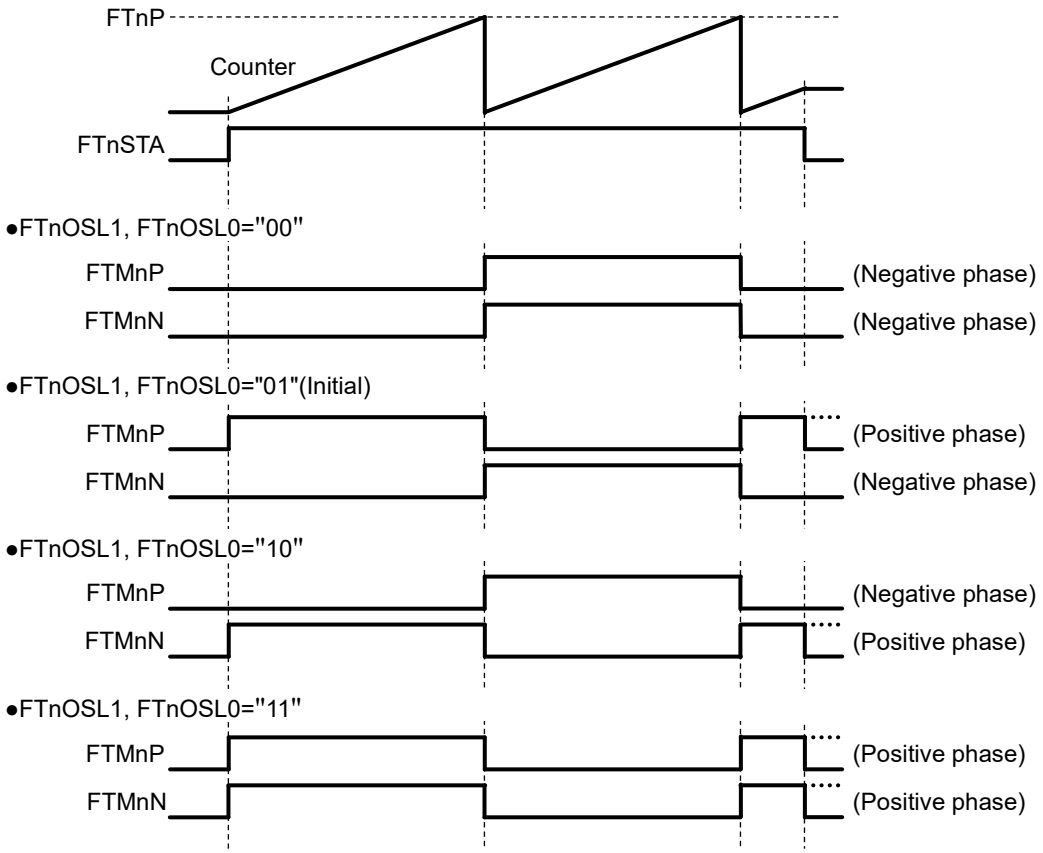
(a) Repeat mode



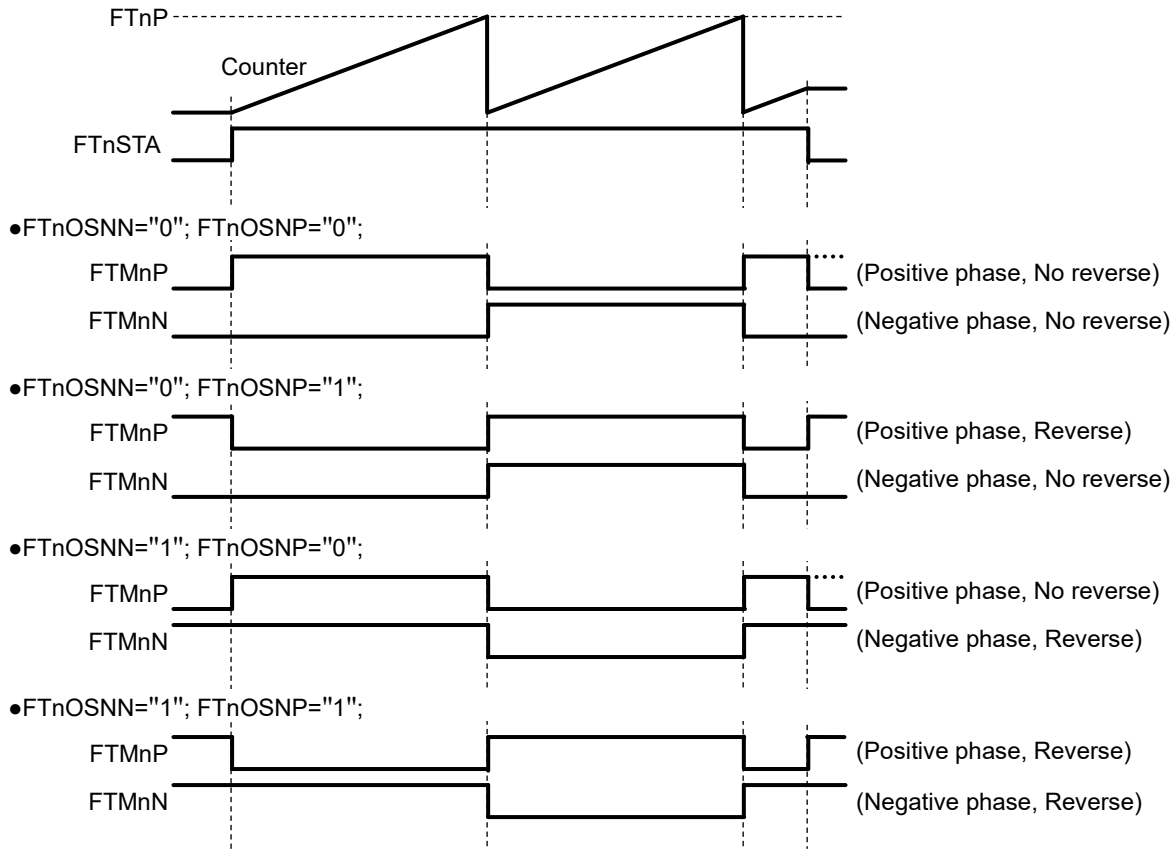
(b) One-shot mode

Figure 9-3 Waveforms of the positive phase/negative phase output in the Timer mode

FTnOSL1 and FTnOSL0 bits of FTnMOD register are used to choose the phase of the output signal driven to the FTMnP/FTMnN pins. FTnOSNP bit is for reversing the output to the FTMnP pin. FTnOSNN bit is for reversing the output to the FTMnN pin.
Figure 9-4(a) shows waveforms on different conditions of the output phase to the FTMnP/FTMnN pins specified by the FTnOSL1 and FTnOSL0 bits of the FTnMOD register.
Figure 9-4(b) shows waveforms when reversing the output to the FTMnP pin by the FTnOSNP bit and reversing the output to the FTMnN pin by the FTnOSNN bit.



(a) Output phases specified by FTnOSL1 bit and FTnOSL0 bit

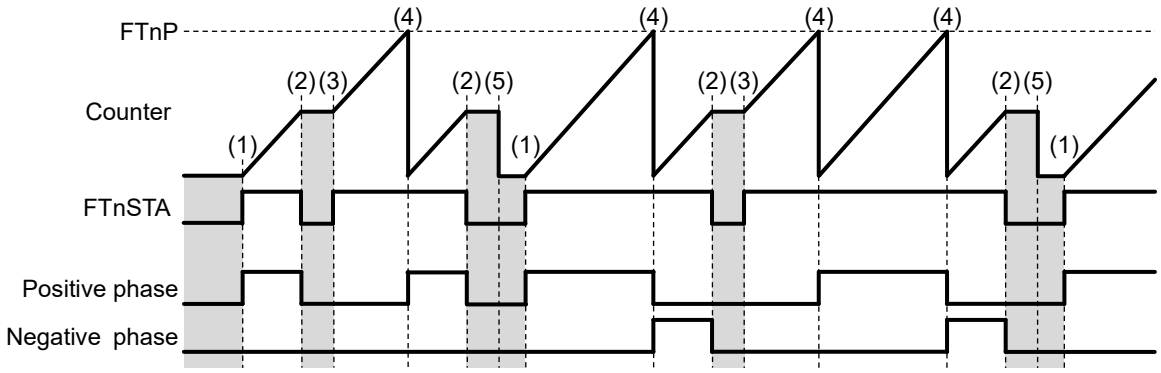


(b) Output reverse specified by FTnOSNN bit and FTnOSNP bit

Figure 9-4 Output waveforms in TIMER mode

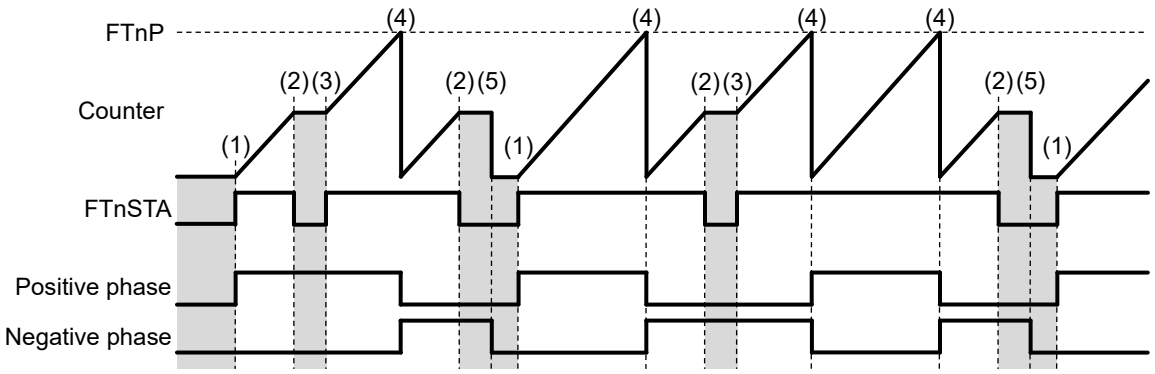
FTnSTPO bit of FTnMOD register is used to choose output conditions when the counter stops.

Figure 9-5(a) shows output waveforms when the FTnSTPO bit of the FTnMOD register is "0".
Figure 9-5(b) shows output waveforms when the FTnSTPO bit of the FTnMOD register is "1".



Counter operation
(1) Counter operation start
(2) Counter operation stop
(3) Counter operation restart without clear the counter
(4) Counter overflow
(5) Counter clear

(a) FTnSTPO=0



Counter operation
(1) Counter operation start
(2) Counter operation stop
(3) Counter operation restart without clear the counter
(4) Counter overflow
Counter clear

(b) FTnSTPO=1

Figure 9-5 Output waveforms in TIMER mode (When counter stop)

9.3.4 CAPTURE Mode Operation

The CAPTURE mode stores the count value, which was obtained when an event trigger source was generated, in the FTnEA or FTnEB register.
The event trigger source for the capture is common to that used for counter start/stop.

Stored data in FTnEA register	Count value at the time when an event trigger rising edge is generated
Stored data in FTnEB register	Count value at the time when an event trigger falling edge is generated

9.3.4.1 Operation Example in CAPTURE Mode

The following example shows the operation of one cycle and duty of the PWM signal input from the P02/EXTRG0 pin in the CAPTURE mode using the counter start/stop through trigger events.
Set each register in the following steps before measuring.

Step 1: Write "01" to the FTnMD1 and FTnMD0 bits of the FTnMOD register to choose the CAPTURE mode.
Step 2: When using an interrupt, write "1" to the FTnIETS bit of the FTnINTE register to enable the trigger counter stop interrupt.
Step 3: Write "0" to the FTnSTSS bit of the FTnTRG0 register, "0000" to FTnSTS3 to FTnSTS0 bits to set the source of the trigger event to "EXTRG0".
Write "1" to the FTnST bit to enable the start function of the counter.
Write "1" to the FTnSP bit to enable the stop function of the counter.
Step 4: Write "000" to FTnTRM2 to FTnTRM0 bits of the FTnTRG1 register to choose the trigger through the rising edge for both of counter start/stop.
Step 5: Write "1" to the FTnETG bit of the FTCSTR register to enable the trigger operation of capturing.

Figure 9-6 shows the time chart in this example.

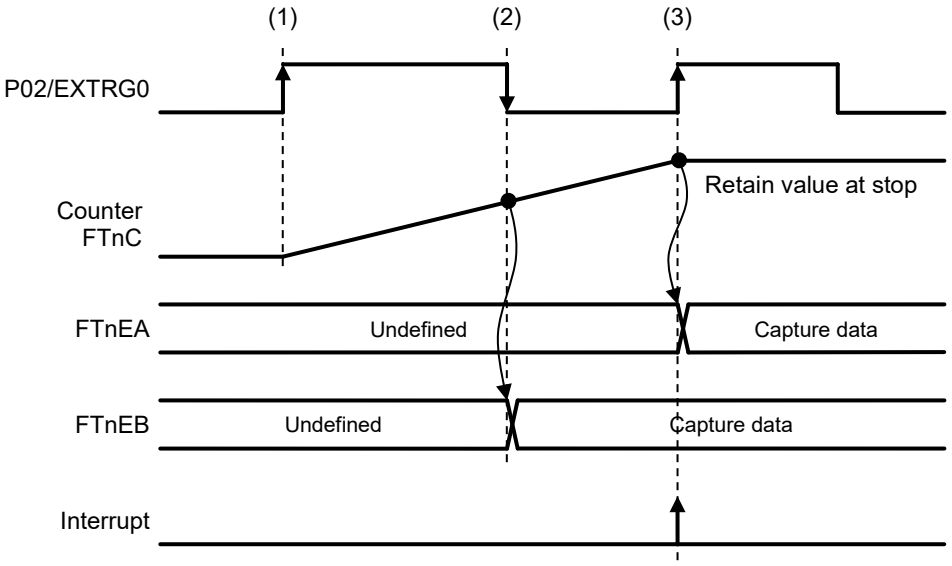


Figure 9-6 Operation Example to measure cycle and duty of PWM signal (one cycle)

(1) The counter starts operating at the rising edge of the signal input from the P02/EXTRG0 pin.
(2) The value of the FTMn counter register FTnC is stored into the FTnEB register at the falling edge of the P02/EXTRG0 pin.
(3) The value of the FTMn counter register FTnC is stored into the FTnEA register at the rising edge of the P02/EXTRG0 pin. The counter stop the operation and the interrupt is generated.

The value of the FTnEA register corresponds to the cycle of the PWM signal input from the P02/EXTRG0 pin, and the value of the FTnEB register corresponds to the duty.

This is an example for measuring the cycle and duty of the PWM signal input from the P02/EXTRG0 pin by the start/stop of a trigger event.
Configure registers as follows before the measurement.

Step 1: Choose the CAPTURE mode by writing "01" to FTnMD1 and FTnMD0 bits of FTnMOD register.
Step 2: When using the interrupt, set FTnIEA bit of FTnINTE register to "1" to enable the event timing A interrupt.
Step 3: Set FTnSTSS bit of FTnTRG0 register to "0"and set FTnSTS3 to FTnSTS0 bits to "0000" to configure the EXTRG0 as the trigger event source.
Set FTnST bit to "1" to enable the start function of the counter.
Set FTnSPC bit to "1" to enable the counter clear when the trigger event of counter stop occurs.
Step 4: Set FTnTRM2 to FTnTRM0 bits of FTnTRG1 register to "000" to choose the rising edge as the trigger for both the counter start and stop.
Step 5: Set FTnETG bit of FTCSTR register to "1" to enable the trigger operation.

Figure 9-7 shows the time chart in this example.

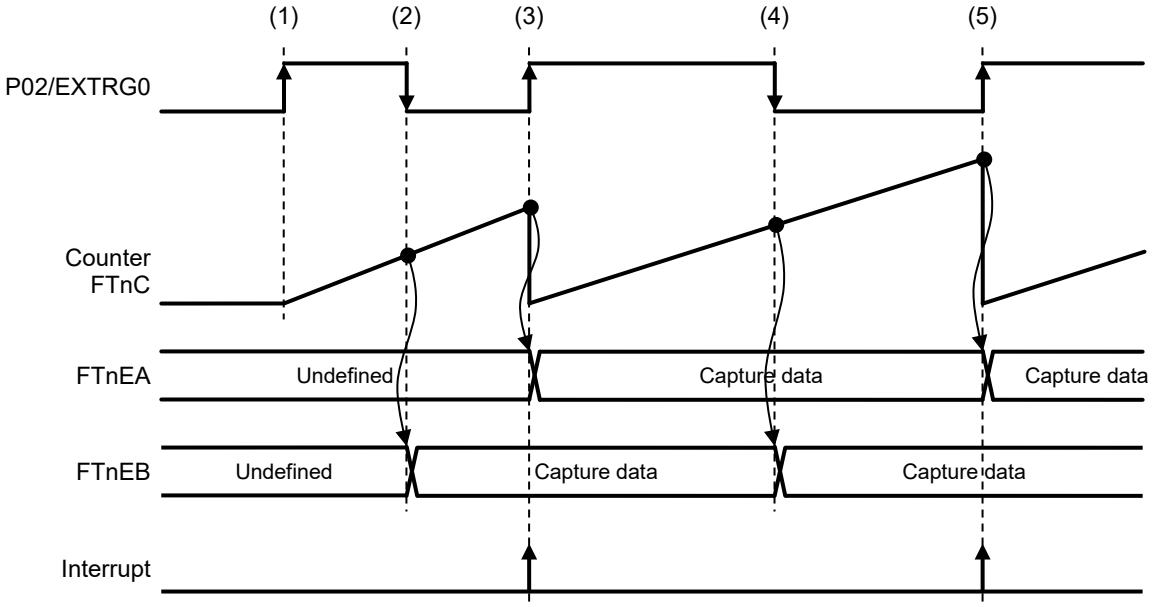


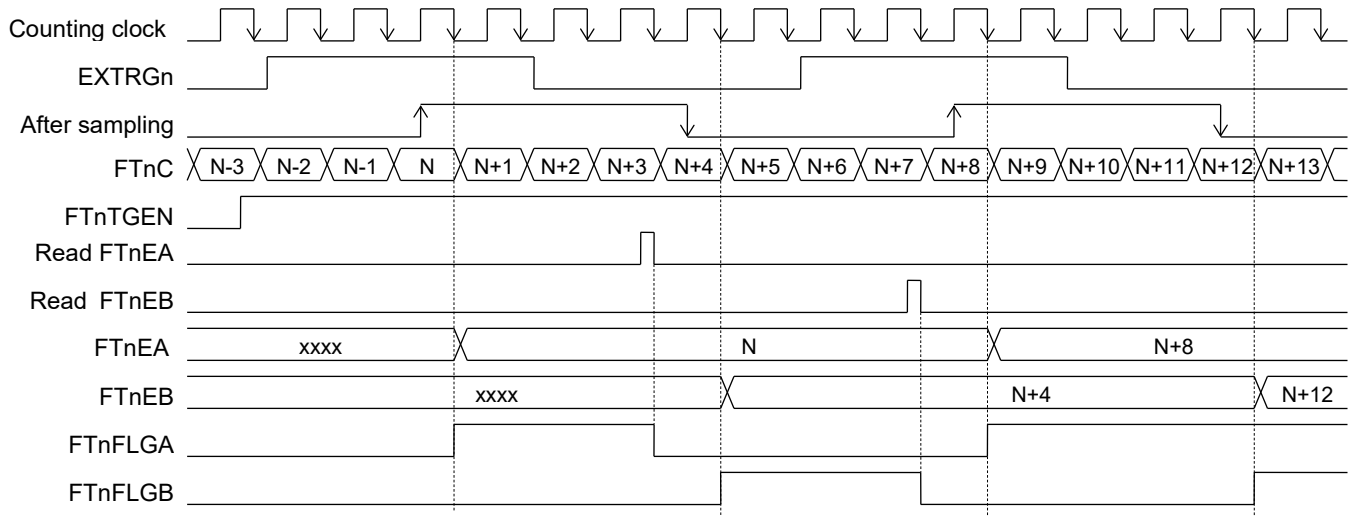
Figure 9-7 Operation Example to measure cycle and duty of PWM signal (repeat cycle)

(1) The counter starts operating at the rising edge of the signal input from the P02/EXTRG0 pin.
(2) The value of the FTMn counter register FTnC is stored into the FTnEB register at the falling edge of the P02/EXTRG0 pin.
(3) The value of the FTMn counter register FTnC is stored into the FTnEA register at the rising edge of the P02/EXTRG0 pin. The counter is cleared and the interrupt is generated. The count operation continues.
(4) The value of the FTMn counter register FTnC is stored into the FTnEB register at the falling edge of the P02/EXTRG0 pin.
(5) The value of the FTMn counter register FTnC is stored into the FTnEA register at the rising edge of the P02/EXTRG0 pin. The counter is cleared and the interrupt is generated. The count operation continues.

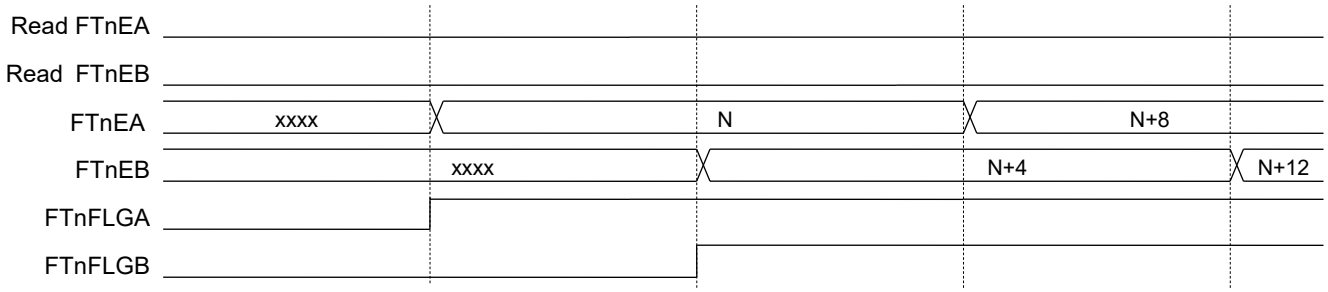
The value of the FTnEA register corresponds to the cycle of the PWM signal input from the P02/EXTRG0 pin, and the value of the FTnEB register corresponds to the duty.

In addition, the operation following the capturing is depending on the setting value in the FTnOST bit of the FTnMOD register.

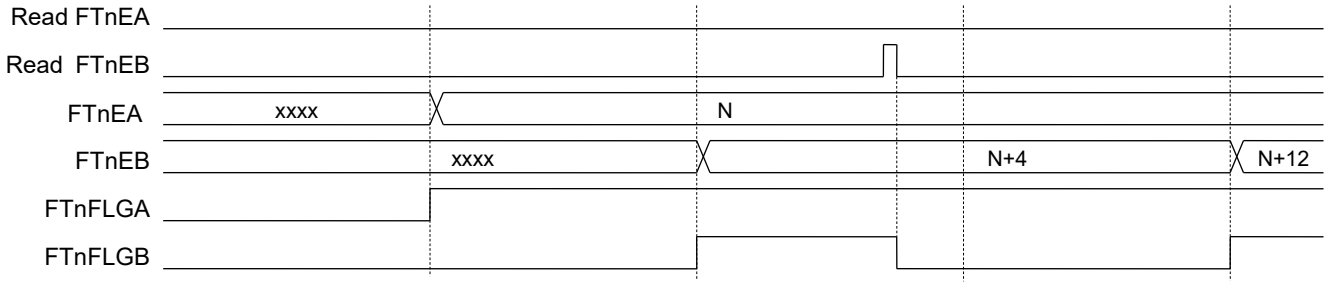
- In the auto mode (FTnOST=0)
  The value of the FTnEA register is updated when the counter is restarted with the signal rising again.
- In the single mode (FTnOST=1)
  The value of the FTnEA register is not updated when the counter is restarted with the signal rising again.



(a) When read the register before the next trigger (common to FTnOST = 0,1)



(b) When not read the register before the next trigger (FTnOST = 0)



(c) When not read the register before the next trigger (FTnOST = 1)

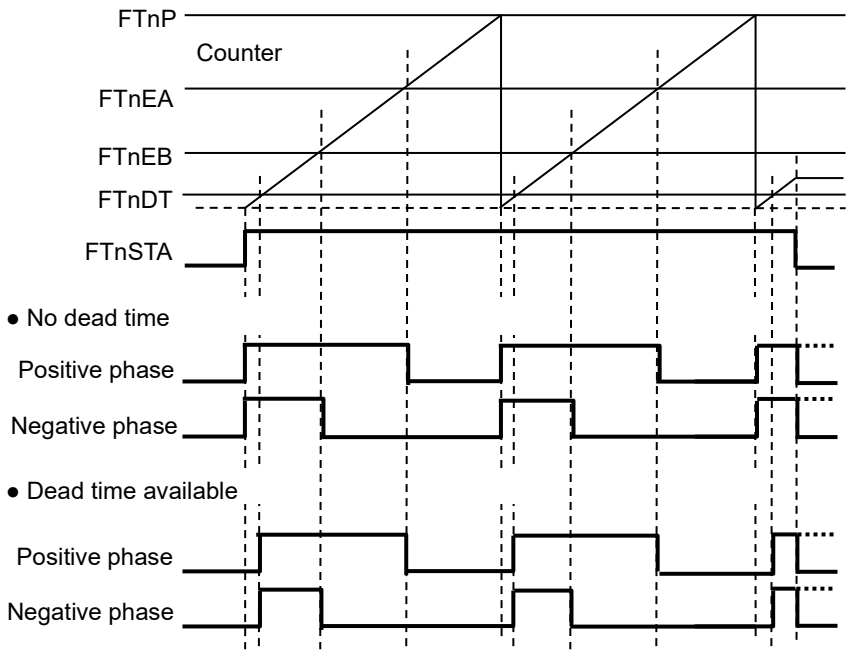
Figure 9-8 Operation Timing in CAPTURE Mode

9.3.5 PWM1 Mode Operation

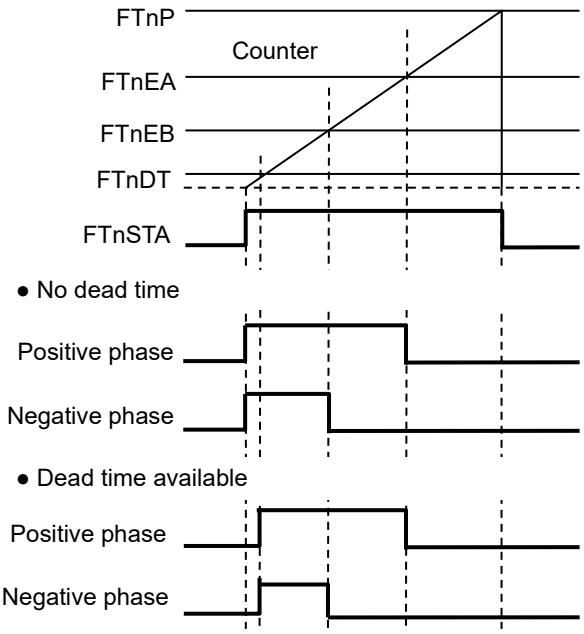
The PWM1 mode generates a pulse with the cycle set in the FTnP register.
The duty of the Positive phase output is set in the FTnEA register and that of the Negative phase output is set in the FTnEB register.

9.3.5.1 Output Waveform in PWM1 Mode

In the repeat mode, the initial values for each of Positive phase/Negative phase outputs are "L" level, and they become "H" level at start. Each of them becomes "L" level depending on the duty value. They resume "H" level in the next cycle. This pattern repeats until the operation is stopped.
In the one-shot mode, they automatically stop after one cycle becoming "L" level. In addition, if the dead time is enabled, the "L" level output is maintained from the start of counting through the dead time period.
Figure 9-9(a) shows waveforms of the positive phase/negative phase output in the repeat PMW1 mode.
Figure 9-9(b) shows waveforms of the positive phase/negative phase output in the one-shot PMW1 mode.



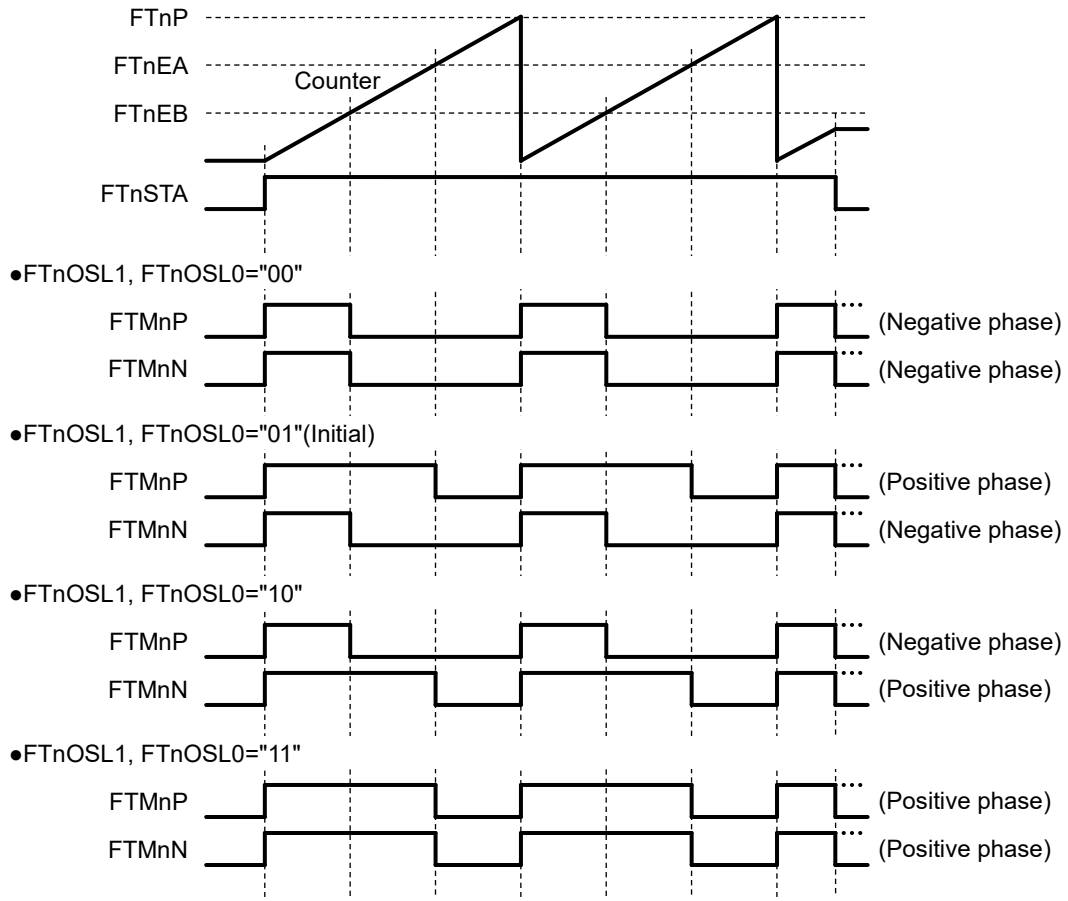
(a) PWM1 Mode Output Waveform (Repeat Mode)



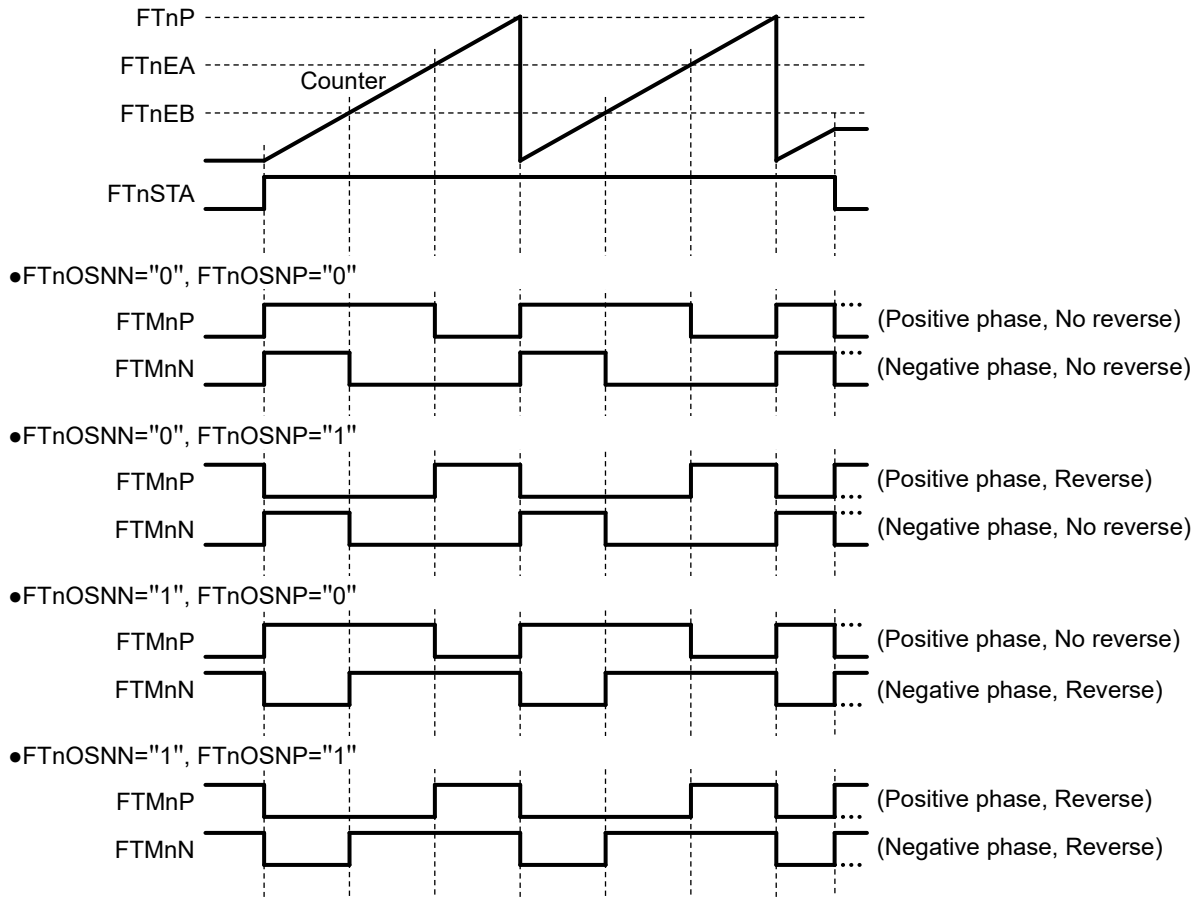
(b) PWM1 Mode Output Waveform (One-shot Mode)

Figure 9-9 PWM1 Mode Output Waveform

FTnOSL1 and FTnOSL0 bits of FTnMOD register are used to choose the phase of the output signal driven to the FTMnP/FTMnN pins. FTnOSNP bit is for reversing the output to the FTMnP pin. FTnOSNN bit is for reversing the output to the FTMnN pin.
Figure 9-10(a) shows waveforms on different conditions of the output phase to the FTMnP/FTMnN pins specified by the FTnOSL1 and FTnOSL0 bits of the FTnMOD register.
Figure 9-10(b) shows waveforms when reversing the output to the FTMnP pin by the FTnOSNP bit and reversing the output to the FTMnN pin by the FTnOSNN bit.



(a) Output phases specified by FTnOSL1 bit and FTnOSL0 bit

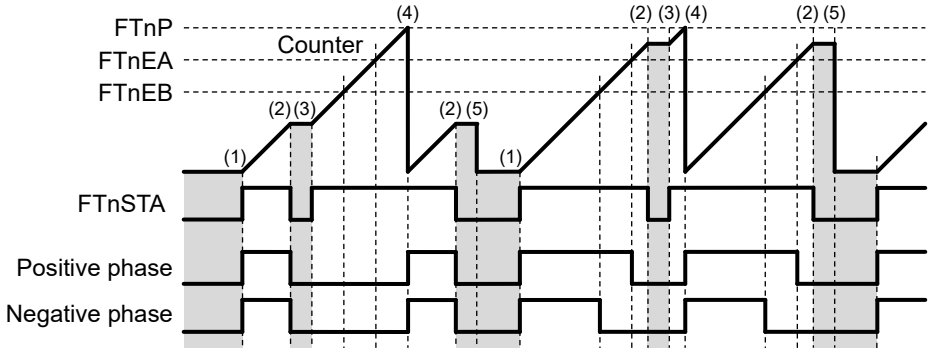


(b) Output reverse specified by FTnOSNN bit and FTnOSNP bit

Figure 9-10 PWM1 Mode Output Waveform (phase/reverse controls)

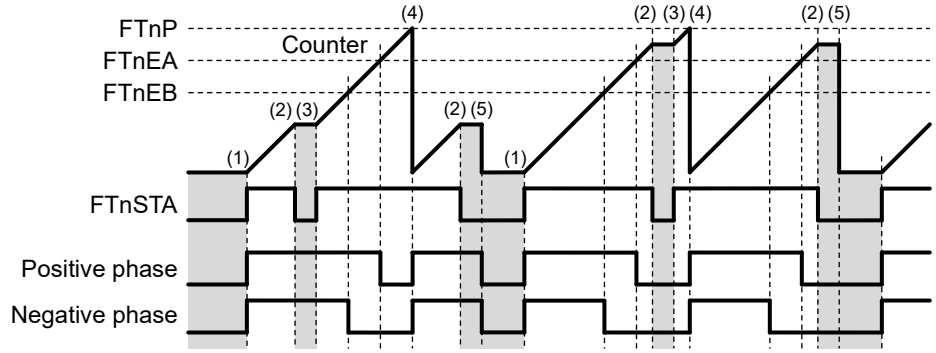
FTnSTPO bit of FTnMOD register is used to choose output conditions when the counter stops.

Figure 9-11(a) shows output waveforms when the FTnSTPO bit of the FTnMOD register is "0".
Figure 9-11(b) shows output waveforms when the FTnSTPO bit of the FTnMOD register is "1".



Counter operation
(1) Counter operation start
(2) Counter operation stop
(3) Counter operation restart without clear the counter
(4) Counter overflow
(5) Counter clear

(a) FTnSTPO=0



Counter operation
(1) Counter operation start
(2) Counter operation stop
(3) Counter operation restart without clear the counter
(4) Counter overflow
(5) Counter clear

(b) FTnSTPO=1

Figure 9-11 PWM1 Mode Output Waveform (when counter stop)

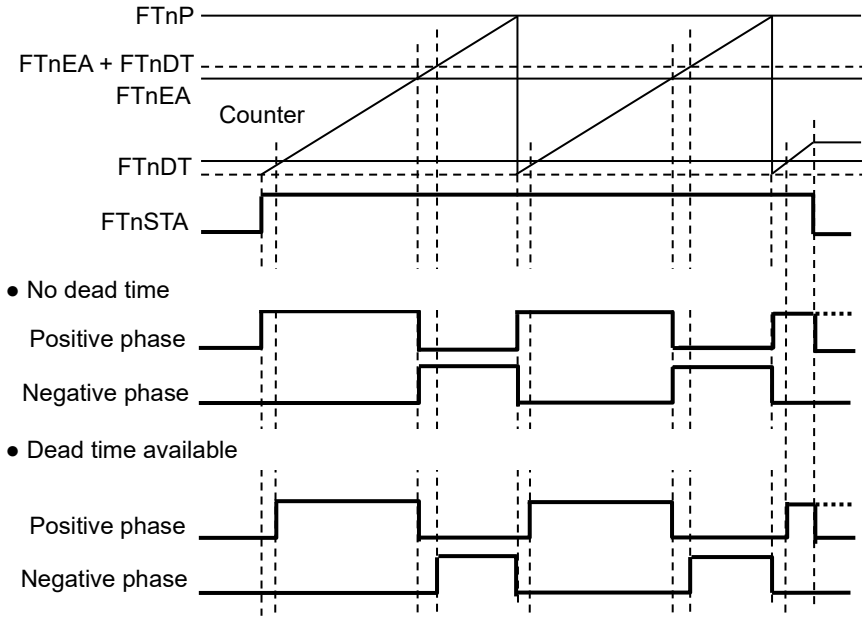
9.3.6 PWM2 Mode Operation

PWM2 mode generates a complementary output pulse with the cycle set in the FTnP register.
The duty of the Positive/Negative phase output is set in the FTnEA register. The FTnEB register is not used.

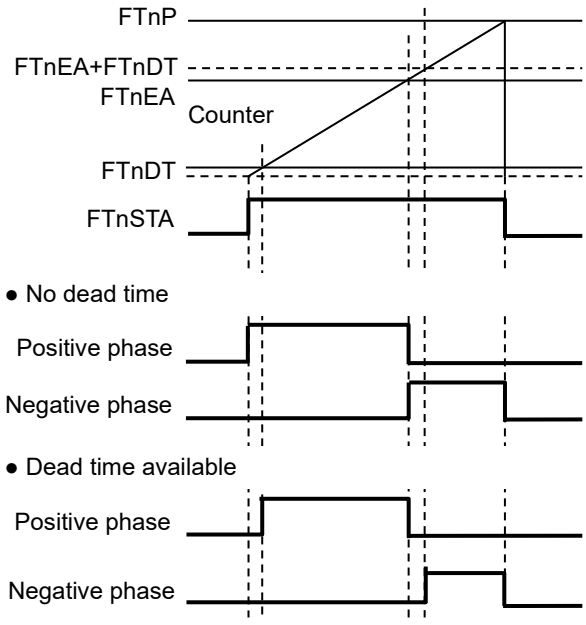
9.3.6.1 Output Waveform in PWM2 Mode

In the repeat mode, "L" level is the initial value for each of Positive phase/Negative phase output, and the positive phase becomes "H" level at start. The positive phase output becomes "L" level and the negative phase output becomes "H" level depending on the duty value. In the next cycle, the positive phase output becomes "H" level and the negative phase output becomes "L" level again. This pattern repeats until the operation is stopped. In the one-shot mode, they automatically stop after one cycle becoming "L" level.
In addition, if the dead time is enabled, the "L" level output is maintained, from the start of counting for the positive phase output and from duty coincidence for the negative phase output, through the dead time period.

Figure 9-12(a) shows waveforms of the positive phase/negative phase output in the repeat PMW2 mode.
Figure 9-12(b) shows waveforms of the positive phase/negative phase output in the one-shot PMW2 mode.



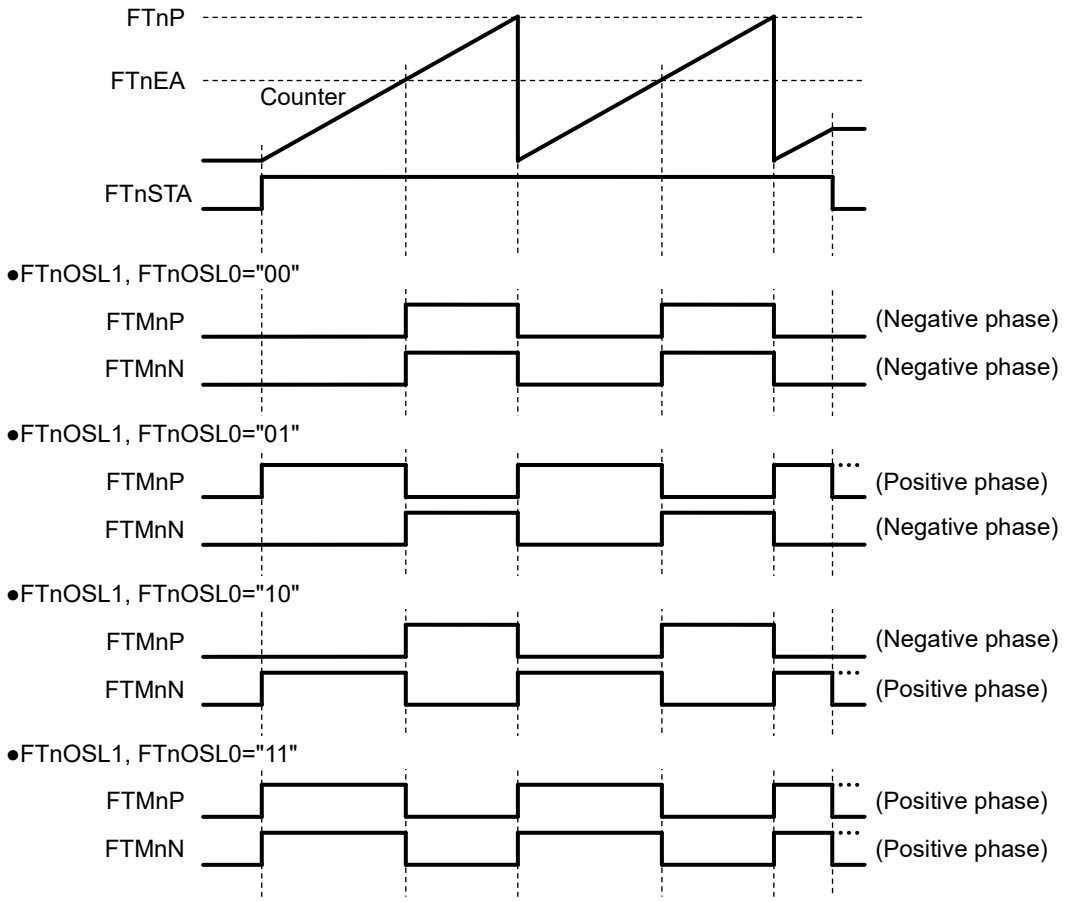
(a) PWM2 Mode Output Waveform (Repeat Mode)



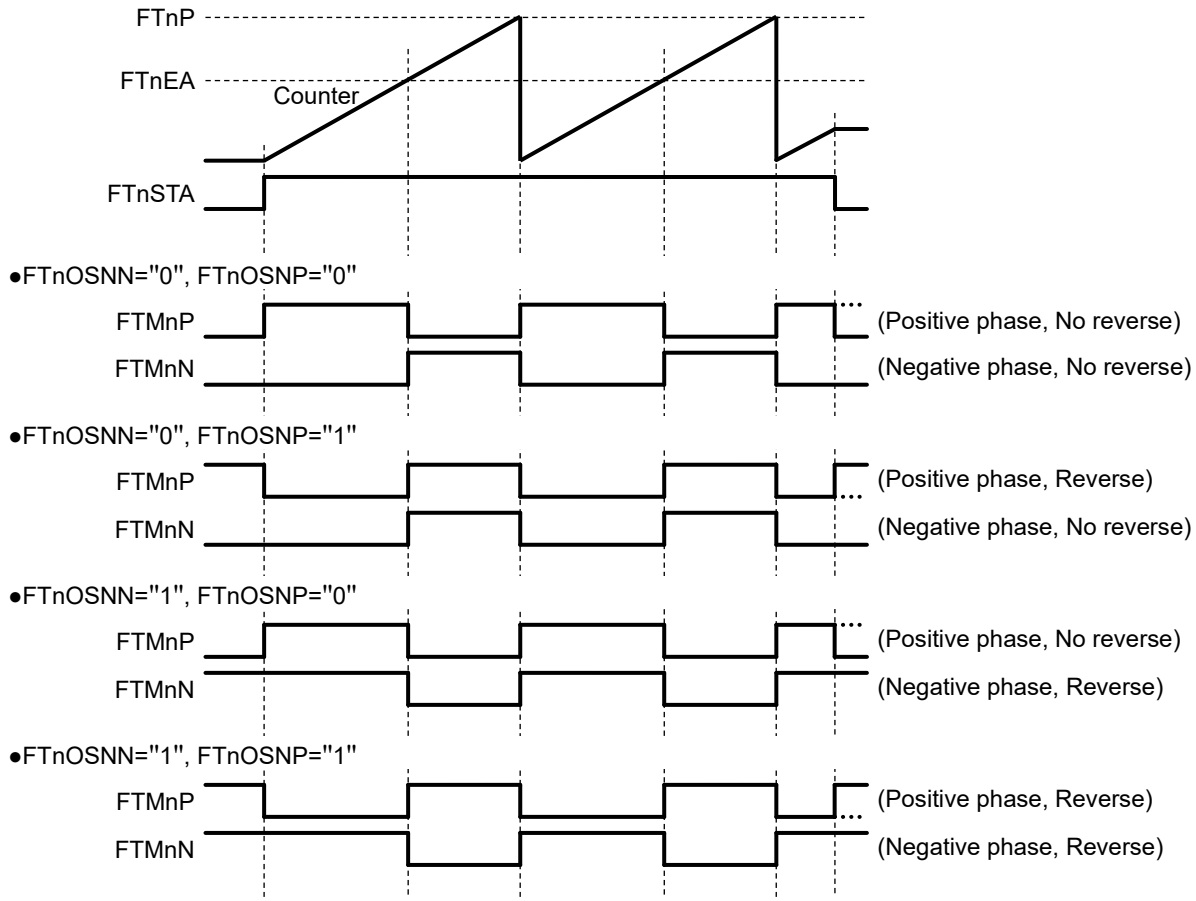
(b) PWM2 Mode Output Waveform (One-shot Mode)

Figure 9-12 PWM2 Mode Output Waveform

FTnOSL1 and FTnOSL0 bits of FTnMOD register are used to choose the phase of the output signal driven to the FTMnP/FTMnN pins. FTnOSNP bit is for reversing the output to the FTMnP pin. FTnOSNN bit is for reversing the output to the FTMnN pin.
Figure 9-13(a) shows waveforms on different conditions of the output phase to the FTMnP/FTMnN pins specified by the FTnOSL1 and FTnOSL0 bits of the FTnMOD register.
Figure 9-13(b) shows waveforms when reversing the output to the FTMnP pin by the FTnOSNP bit and reversing the output to the FTMNN pin by the FTnOSNN bit.



(a) Output phases specified by FTnOSL1 bit and FTnOSL0 bit

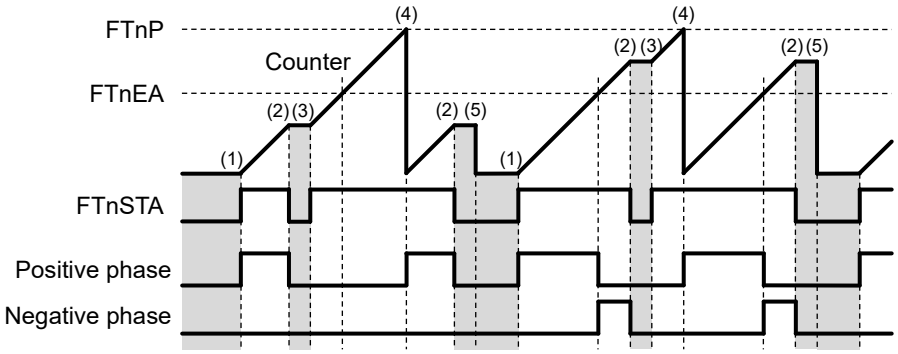


(b) Output reverse specified by FTnOSNN bit and FTnOSNP bit

Figure 9-13 PWM2 Mode Output Waveform (phase/reverse controls)

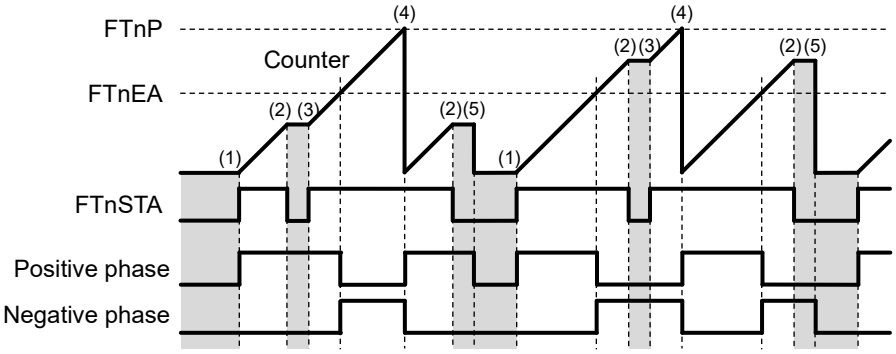
FTnSTPO bit of FTnMOD register is used to choose output conditions when the counter stops.

Figure 9-14(a) shows output waveforms when the FTnSTPO bit of the FTnMOD register is "0".
Figure 9-14(b) shows output waveforms when the FTnSTPO bit of the FTnMOD register is "1".



Counter operation
(1) Counter operation start
(2) Counter operation stop
(3) Counter operation restart without clear the counter
(4) Counter overflow
(5) Counter clear

(a) FTnSTPO=0



Counter operation
(1) Counter operation start
(2) Counter operation stop
(3) Counter operation restart without clear the counter
(4) Counter overflow
(5) Counter clear

(b) FTnSTPO=1

Figure 9-14 PWM2 Mode Output Waveform (when counter stop)

9.3.7 External Clock Input/Event Trigger/Emergency Stop Trigger Control

The functional timer can accept external clock input and two types of trigger signal: event trigger and emergency stop trigger.
The external clock input selected in the EXTRG0 to EXTRG7 is used as the count clock.
The event trigger is used as counter start/stop/clear or trigger for capture. The trigger source can be chosen from EXTRG0 to EXTRG7, CMP0TRG, TMH0TRG to TMH7TRG, or FTM0TRG to FTM7TRG.
See Section "29.3.3 Clock Mutual Monitoring Function" for the clock mutual monitoring in the safety function.

The emergency stop trigger stops the timer operation. It stops the counter and makes the Positive/Negative output "L" level. The trigger source can be chosen from EXTRG0, EXTRG4, and CMP0TRG.

The EXTRG0 to EXTRG7 and CMP0TRG are output of sampling controller of the external interrupt function or the analogue comparator. They are connected to functional timer as event trigger or external clock input.
The input signal is sampled by the timer clock regardless a noise filter function enabled/disabled.
The input signal is delayed 3 clocks of the timer clock if noise filter is disabled. Also, it is delayed 2 clocks of the filtering clock in addition to 3clocks of the timer clocks. See Section "9.3.7.2 Noise Filter Function" for details.

If EXTRG0 to EXTRG7 and CMP0D are chosen as the emergency stop trigger, a noise filter is not available on the functional timer.
Use the sampling controller of the external interrupt function or the analogue comparator.

The interrupt sources of 16-bit timer and functional timer can be set through registers of each timer.

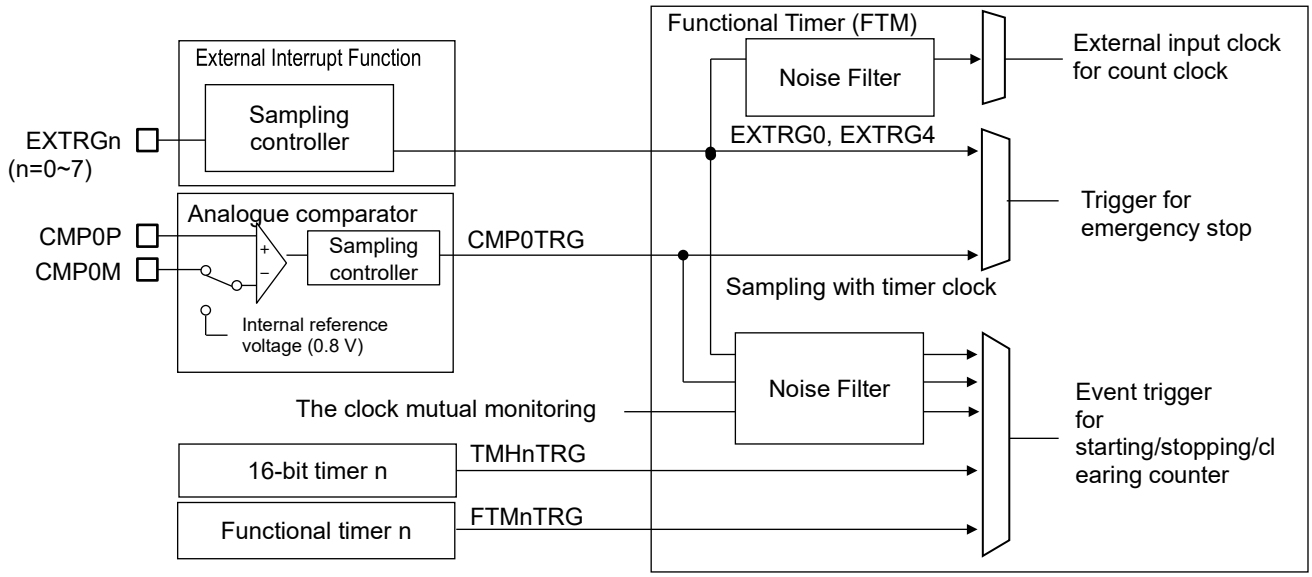


Figure 9-15 Input Path of Trigger Signal

9.3.7.1 External Input Clock

Follow this setting when using the external clock input.
Set FTnXCK2 to FTnXCK0(external clock input source) and FTnEX="1"(to select external clock input) after setting FTnCKD2 to FTnCKD0="000"(no dividing) and setting FTnCK0="1"(if the system clock is high speed clock) or setting FTnCK0="0"(if the system clock is LSCLK). Then, reconfigure the FTnCK0 bit and FTnCKD2 to FTnCKD0 bits for using the timer clock.

9.3.7.2 Noise Filter Function

The noise filter function is available for event triggers of EXTRG0 to EXTRG7 and CMP0TRG or external clock of EXTRG0 to EXTRG7. The enable/disable of the filter function and the width of the filter is configurable by setting FTnTRF2-0 bits of FTNTRG1 resister.
The trigger/external clock generates a signal synchronized with the timer clock and two clocks delayed (sampled signal), regardless the noise filter function is enabled or disabled.
When the noise filter function is disabled (FTnTRF2-0="000"), the counter control (the trigger control or FTnC) circuit accepts the trigger/external clock after one timer clock.
When the noise filter function is enabled (FTnTRF2-0="001" to "111" and FTnETG=1), the counter control circuit accepts a signal generated by detecting same level of the sampled signal for two clocks of the filtering clock (filtered signal), after one timer clock.

Figure 9-16 shows an example of the timing chart when the FTnTRF2-0 is "001". In this setting, the filtering clock is generated by two clocks of the timer clock, it takes four timer clocks to surely accept the pulse and takes two timer clocks to surely filter the pulse.
At the timing of the filtered signal (1), (5) and (6) in the Figure 9-16, the event trigger/external clock are recognized as a valid pulse and the levels are updated because the same level of the event trigger/external clock remains for two filtering clocks.
At the timing of (3), although the level of the event trigger/external clock is same for the two rising edge of the filtering clock, the level changes during the filtering clocks. Therefore, the change is recognized as a noise and it's filtered.
Also, the sampling is implemented by the timer clock. If the event trigger/external clock has short pulses, the sampled signal may be generated as the example of (2) or may not be generated as the example of (4), depending on the timing.

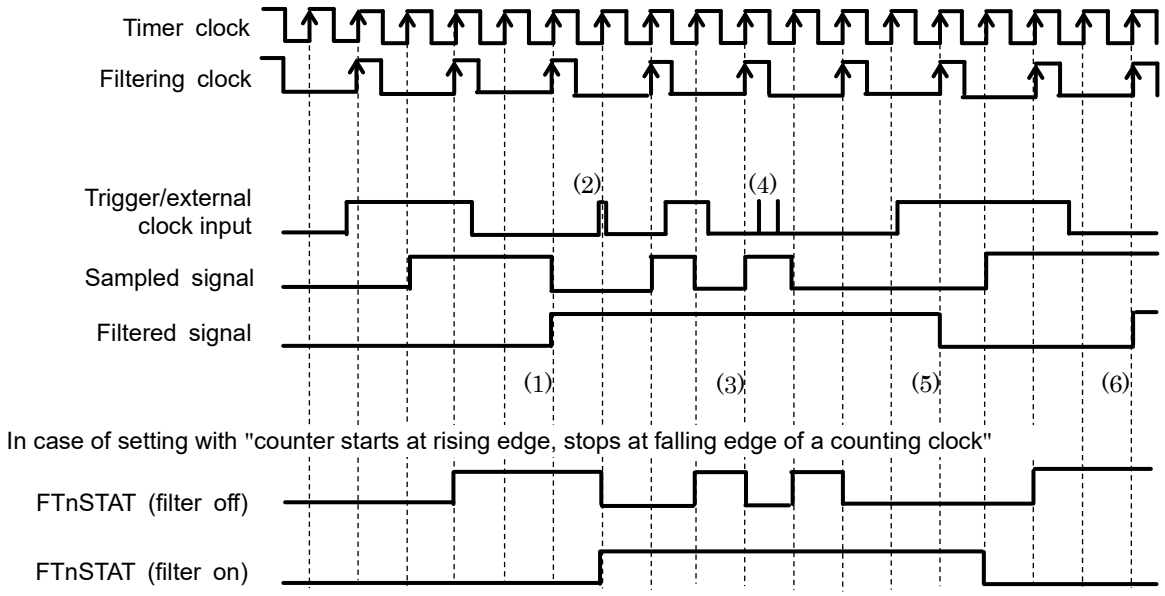


Figure 9-16 Noise filter function

9.3.7.3 Start/Stop Operations by Event Trigger

Here is the setting used to control the counter by event triggers.
First, before controlling the counter, set the following configuration by FTnTRG0 and FTnTRG1 registers.

Choose "no division" as the timer clock.
If using HSCLK as the system clock, write "1" to the FTnCK0 bit of the FTnCLK register, and "000" to FTnCKD2 to FTnCKD0 bits.

Setting the FTnTRG0 register
- Enable/disable counter start/stop with event triggers
- Clear/not clear the counter when starting/stopping with event triggers
- Accept/not accept the next counter start after stopping with event triggers
- Accept/not accept the counter clear if the Positive phase output is "H" level when clearing the counter with event triggers.
- Event trigger source (EXTRG0 to EXTRG7, TMH0TRG to TMH5TRG, FTM0TRG to FTM7TRG, CMP0TRG)

Setting the FTnTRG1 register
The edge/level of the event trigger causing counter start
The edge/level of the event trigger causing counter stop

Setting the timer clock used
Choose the timer clock in the FTnCLK register.
(Even if not changing the setting, choose the timer clock again.)

Once the configuration above is completed, control the counter by the FTCSTR register. The procedure is as follows:

(1) Make the waiting state for an event trigger
Write "1" to the FTnETG bit to make the waiting state for an event trigger (if the level setting is applied for trigger start and the level is applicable, the counter operation is started as soon as the FTnTGEN bit of the FTCSTAT register becomes "1".)

(2) Start the timer counting by the software
If writing "1" to the FTnETG bit, and writing "1" to the FTnSTR bit with the trigger operation enabled, the timer counting is started by the software.
If writing "0" to the FTnSTP bit of the FTCSTP register while counter operation is in progress, the timer counting is stopped by the software.

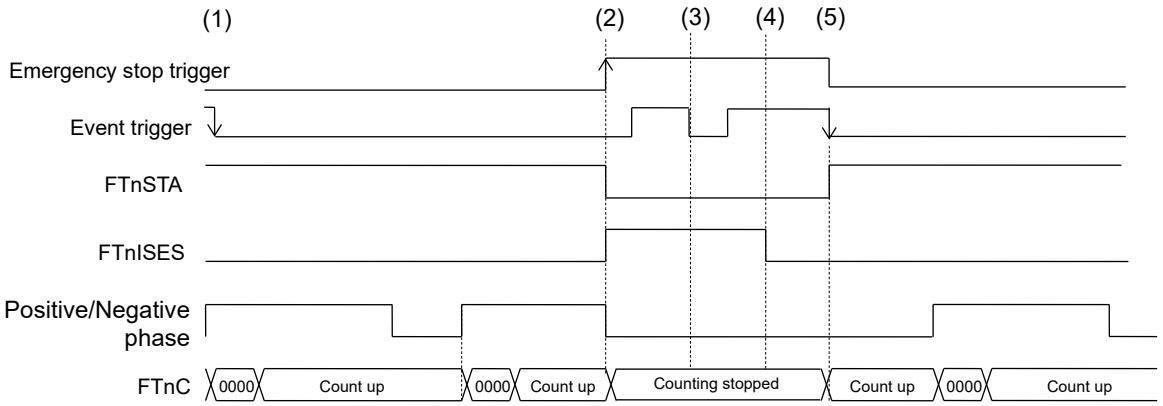
9.3.7.4 Emergency Stop Operation

The emergency stop function is enabled by writing "1" to the FTnEMGEN bit of the FTCCON register. Set the FTnEMGEN bit after the trigger source is chosen in the FTnEST bit of the FTnTRG0 register.

The emergency stop works incorrectly in the following case. It is regardless of timer operating state.
- When FTnEMGEN bit is changed to "1" from "0" under the condition that "H" level is input as trigger and trigger edge setting is "rise".
- When FTnEMGEN bit is changed to "0" from "1" under the condition that "H" level is input as trigger and trigger edge setting is "fall".

The emergency stop interrupt status (FTnISES bit) and the interrupt request bit (QFTMn) becomes "1" after 3 clocks of system clock and 3 clocks of timer clock from changing FTnEMGEN bit of the FTCCON register.

If the emergency stop trigger input (rising edge) is present, the counter is stopped, brings Positive/Negative phase output to "L" level, and generates an emergency stop interrupt.
To restart the counter operation, write "1" to the FTnICES bit of the FTnINTC register to clear the emergency stop interrupt status.
Figure 9-17 shows the operation timing at emergency stop.



(1) The counter operation starts by the event trigger (falling edge).
(2) The counter stops at by the emergency stop trigger (rising edge). The 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 by the event trigger (falling edge).
(The counter is not cleared in this example, so pulse output is restarted after one cycle)

Figure 9-17 Operation Timing Diagram at Emergency Stop

Once the emergency stop occurs, the counter is stopped after one clock of the timer clock, and the FTnISES bit of the FTnINTS register becomes "1" (see (2) in Figure 9-17).
When the FTnISES bit is "1", even if the event trigger of counter start is generated, it is not accepted. If the event trigger for the counter start is generated after the FTnISES bit is cleared (see (4) in Figure 9-17), counting up is restarted (see (5) in Figure 9-17).
To restart the counting operation by the software, make sure that the FTnISES bit becomes "0".

[Note]
- **Change FTnEMGEN bit avoiding the condition that emergency stop function works incorrectly. Also, release the emergency stop to start a timer after the malfunction.**

9.3.8 Output at Counter Stop

The state of Positive/Negative phase output when the counter is stopped by the software or the event trigger input is determined by the FTnSTPO bit setting of the FTnMOD register.

(1) If the FTnSTPO bit is "0":
The Positive/Negative phase outputs become "L" level as soon as the counter is stopped. If the counter is restarted in this state, the Positive/Negative phase output remains at "L" level during the present cycle and changes according to the count value from the next cycle.

(2) If the FTnSTPO bit is "1":
The Positive/Negative phase output remains the state at the time the counter is stopped. When counting is restarted, the state changes according to the count value.

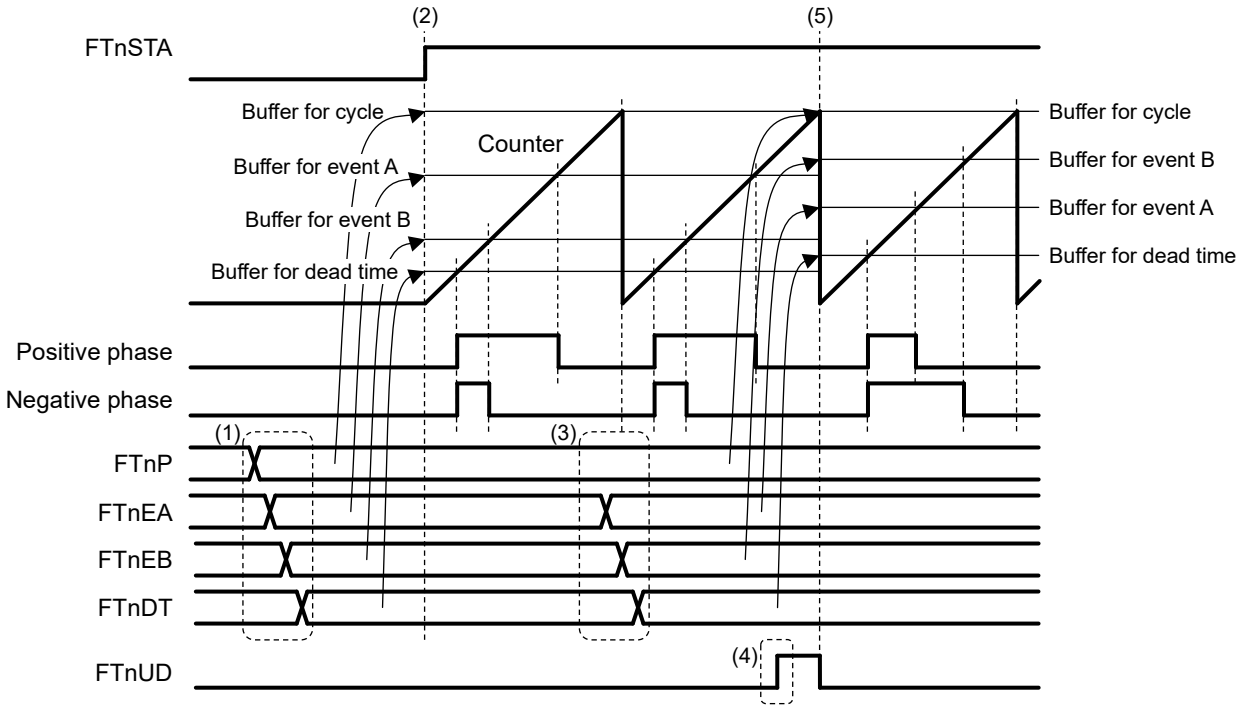
If writing "1" to the FTnSTC bit of the FTnTRG0 register or clearing the counter by the software after the counting operation is stopped, the counter value is counted up from "0x0000", and the output varies depending on the count value.

9.3.9 Changing Cycle, Event A/B, and Dead Time during Operation

The cycle, event A/B, and dead time can be updated by setting FTnP/ FTnEA/ FTnEB/ FTnDT registers.
The update timing is depending on the counter operation status and the counter value when writing data to the registers.

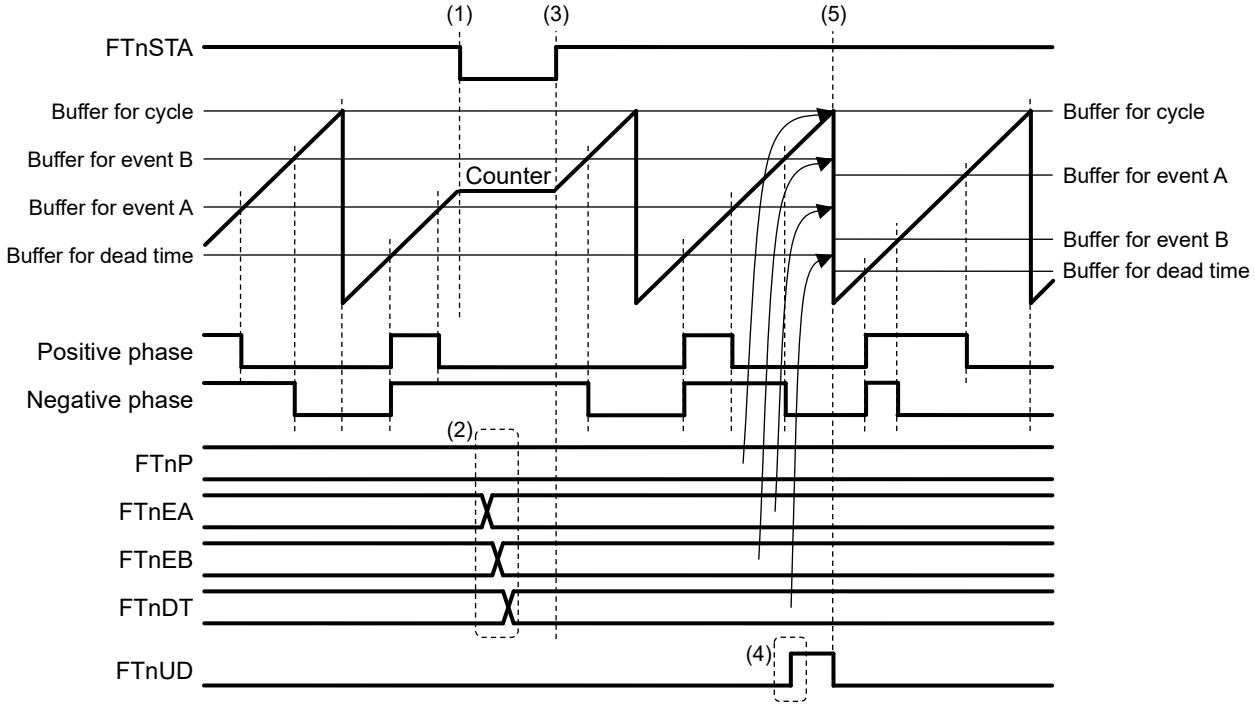
Counter operation status when setting the register	Counter value when setting the register	Update timing
Stop	0x0000	Updated at the counter start
Stop	Other than 0x0000	Updated at the start of cycle while the counter has been restarting and FTCUDn bit is set to "1".
Operating	Any value	Updated at the start of cycle while the counter is operating and FTCUDn bit is set to "1".

Figure 9-18 shows the operating waveforms when the registers are updated while the counter stops (counter value is 0x0000) or the counter is operating.
Figure 9-19 shows the operating waveforms when the registers are updated while the counter stops (counter value is other than 0x0000).



(1) Update the registers while the counter stops (counter value is 0x0000)
(2) Each buffer is update at the start of counter operation
(3) Update the registers while the counter is operating
(4) Set FTnCUDn bit to "1"
(5) Each buffer is updated at the start of cycle and the FTnUD bit gets cleared

Figure 9-18 Update timing while the counter stops (counter value is 0x0000) or the counter is operating



(1) The counter stops
(2) Update the registers while the counter stops (counter value is other than 0x0000)
(3) The counter operation restarts (Each buffer is not updated at this timing)
(4) Set FTCUDn bit to "1"
(5) Each buffer is updated at the start of cycle and the FTnUD bit gets cleared

Figure 9-19 Update timing while the counter stops(counter value is other than 0x0000)

9.3.10 Interrupt Source

This section describes the interrupt source and how to clear it.
Writing "1" to the corresponding bit (FTnIE*) of the FTnINTE register causes each interrupt request to be enabled.
Note that permission of the emergency stop interrupt is not available. If the emergency stop function is enabled, the interrupt are also enabled.
For the source which caused the interrupt status to become "1", write "1" to each interrupt status clear bit (FTnIC*) to clear each interrupt status bit (FTnIS*).
If using an interrupt, clear each interrupt status bit (FTnIS*) at the end of the interrupt routine.

Name	Mode	Status	How to clear
Period coincident interrupt	All modes	FTnISP bit	Write "1" to FTnICP bit
Event A coincident interrupt	TIMER/PWM1/PWM2	FTnISA bit	Write "1" to FTnICA bit
Capture A interrupt	CAPTURE	FTnISA bit	Write "1" to FTnICA bit, or read the FTnEA register
Event B coincident interrupt	TIMER/PWM1	FTnISB bit	Write "1" to FTnICB bit
Capture B interrupt	CAPTURE	FTnISB bit	Write "1" to FTnICB bit, or read the FTnEB register
Trigger stop interrupt	All modes	FTnISTS bit	Write "1" to FTnICTS bit,
Trigger start interrupt	All modes	FTnISTR bit	Write "1" to FTnICTR bit,
Emergency stop interrupt	All modes	FTnISES bit	Write "1" to FTnICES bit,

The cycle coincident interrupt/event A coincident interrupt/event B coincident interrupt can be chosen as the interrupt trigger output.

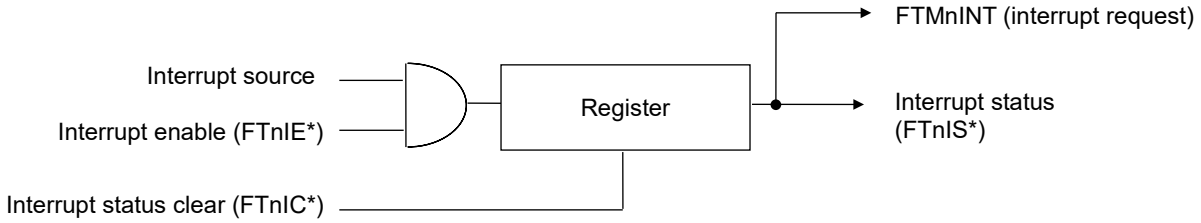


Figure 9-20 Interrupt Control Signal

Chapter 10 Watchdog Timer

10. Watchdog Timer

10.1 General Description

The watchdog timer (WDT) is equipped with the following functions and can detect the runaway state of program or the undefined state of the CPU by generating an interrupt or reset when an abnormality occurs.

- If the counter is not cleared for more than a certain time period in program operation and overflows, the WDT interrupt is generated in the first overflow and the WDT reset in the second overflow (if the window function is disabled).
- If the counter is not cleared for more than a certain time period in program operation and overflow occurs, the WDT reset is generated in the first overflow (if the window function is enabled).
- If the counter is cleared in the unexpected time period, the WDT invalid clear reset is generated (if the window function is enabled).

The window function refers to the function through which "the time period during which WDT counter clear is enabled" = "the time period during which the window is opened" and
"the time period in which WDT counter clear is disabled" = "the time period in which the window is closed" can be set.

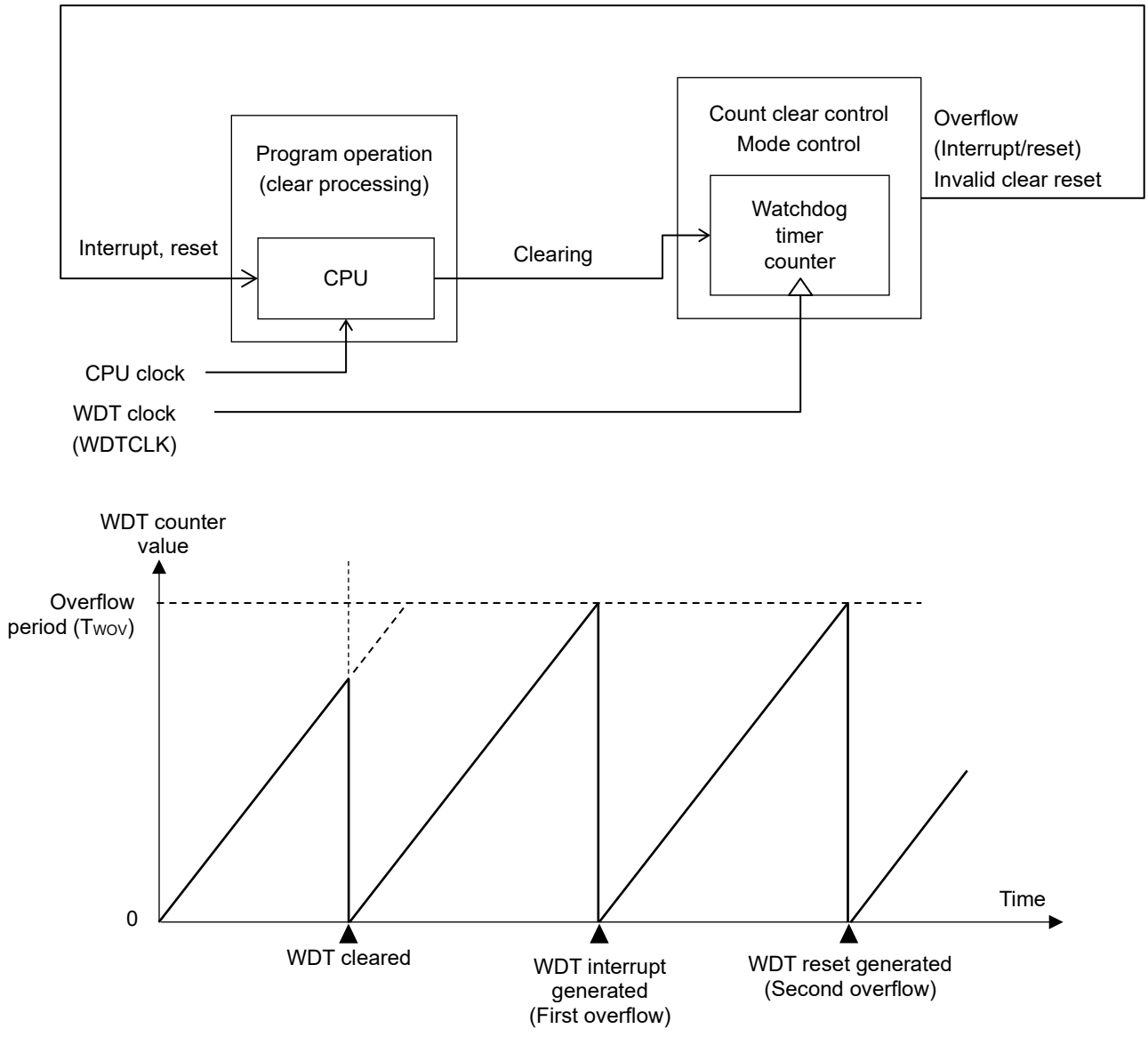


Figure10-1 Watchdog Timer Overview (With the Window Function Disabled)

10.1.1 Features

- Eight types of overflow periods can be chosen (7.8 ms, 15.6 ms, 31.3 ms, 62.5 ms, 125 ms, 500 ms, 2 s, or 8 s)
- Two types of use are available:
  - Window function disabled mode
    The WDT counter can always be cleared. The WDT interrupt is generated when the first counter overflow occurs, and the WDT reset is generated when the second counter overflow occurs.
  - Window function enabled mode
    The periods during which WDT counter clear is enabled and disabled respectively can be set. The WDT reset is generated when the first counter overflow occurs, and the WDT invalid clear reset is generated when the counter is cleared in the period during which WDT counter clear is disabled.

Table 10-1 Watchdog Timer Operation Modes

Mode	overflow		WDT invalid clear reset
	First	Second	
Window function disabled mode	Interrupt	Reset	-
Window function enabled mode	Reset	-	Reset

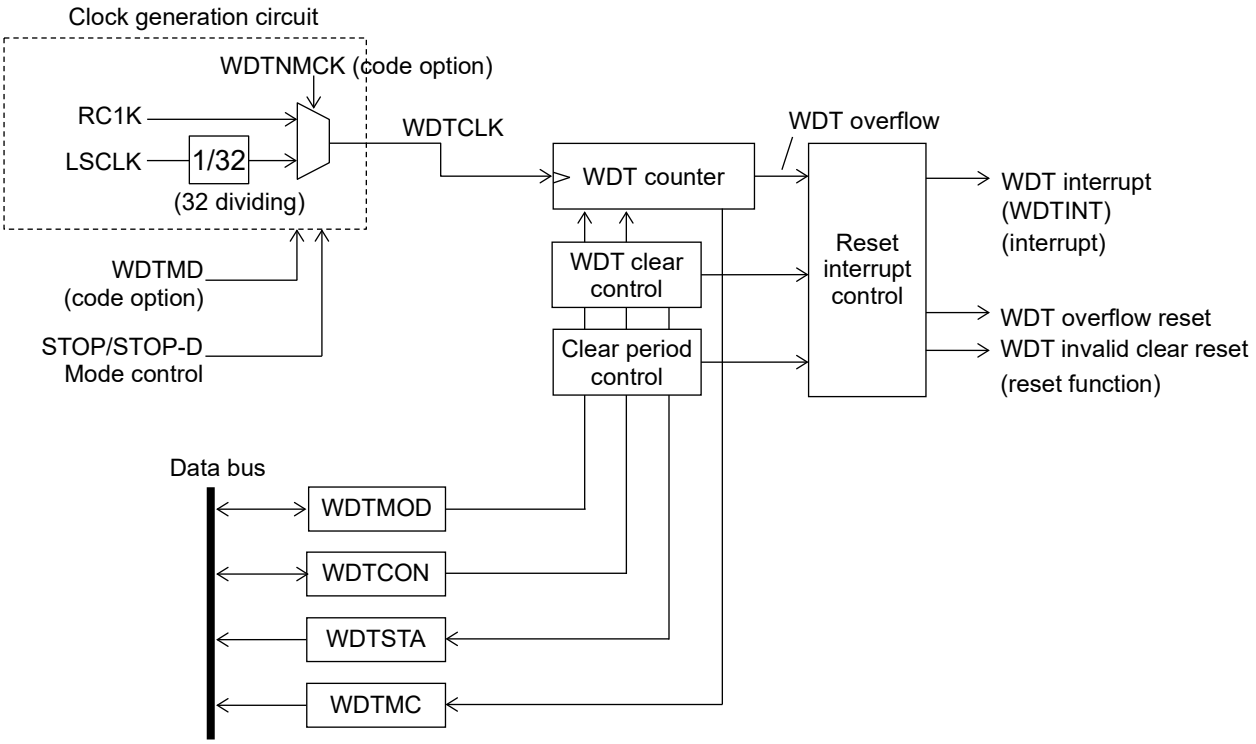
- The following items can be chosen by the code option. See the Chapter 26 "Code Option" for details of the code option.
  - Enabling/disabling the WDT timer operation
  - Operation clock of the WDT counter (32 dividing of low-speed clock LSCLK, WDTCLK RC1K oscillation)

[Note]

- **WDT is the function used to monitor the CPU runaway. Its function as an ordinary timer is not guaranteed.**
- **The watchdog timer is undetectable to all the abnormal operations. Even if the CPU loses control, the watchdog timer is undetectable to the abnormality in the operation state in which the WDT counter is cleared. It is recommended that the WDT counter is cleared at one place in the main loop of the program as a fail-safe.**
- **WDT can be operated based on the clock independent of the system clock by using RC1K oscillation for the WDTCLK, resulting in further improvement of safety. However, it is recommended to choose LSCLK if high accuracy of the frequency is required, since the RC1K oscillation is less accurate than the LSCLK.**

10.1.2 Configuration

The following diagram shows the configuration of the watchdog timer.



WDTCON : Watchdog timer control register
WDTMOD : Watchdog timer mode register
WDTMC : Watchdog timer counter register
WDTSTA : Watchdog timer status register

Figure 10-2 Configuration of Watchdog Timer

10.2 Description of Registers

10.2.1 List of Registers

Address	Name	Symbol name		R/W	Size	Initial value
		byte	Word			
0xF010	Watchdog timer control register	WDTCON	-	R/W	8	0x00
0xF011	Reserved	-	-	-	-	-
0xF012	Watchdog timer mode register	WDTMOD	-	R/W	8	0x06
0xF013	Reserved	-	-	-	-	-
0xF014	Watchdog timer counter register	WDTMCL	WDTMC	R	8/16	0x00
0xF015		WDTMCH		R	8	0x00
0xF016	Watchdog timer status register	WDTSTA	-	R	8	0x01
0xF017	Reserved	-	-	-	-	-

10.2.2 Watchdog Timer Control Register (WDTCON)

This register is a special function register (SFR) to clear the WDT counter.

Address: 0xF010
Access: R/W
Access size: 8 bit
Initial value: 0x00

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

Bit No.	Bit name	Description
7 to 0	d7 to d0	The WDT counter can be cleared by writing "0x5A" with the WDP bit set to "0", then writing "0xA5" with the WDP bit set to "1". In the window mode, WDT invalid clear reset is generated if the WDT counter is cleared in the period during which WDT clear is disabled.
0	WDP	This bit is used to read the value of the WDT bit internal pointer. The WDP bit is reset to "0" when the system is reset as well as when the WDT counter overflows. It is reversed when writing data to the WDTCON register regardless of the data written.

[Note]

- **In the WDT interrupt routine (when the interrupt level (ELEVEL) of the CPU program status word (PSW) is "2"), the WDT counter is unable to get cleared..**

10.2.3 Watchdog Timer Mode Register (WDTMOD)

This register is a special function register (SFR) to set the overflow period and the clear enabled period of the WDT counter.

Address: 0xF012
Access: R/W
Access size: 8 bit
Initial value: 0x06

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								WDTMOD							
Bit	-	-	-	-	-	-	-	-	-	-	WOVF1	WOVF0	-	WDT2	WDT1	WDT0
R/W	R	R	R	R	R	R	R	R	W	W	R/W	R/W	W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit No.	Bit name	Description
7, 6, 3	-	Reserved bits
5, 4	WOVF1, WOVF0	These bits are used to set the mode of WDT. 00: Window function disabled (initial value) 01: Window function enabled mode 1 (the clear enabled period is approximately 75% of the overflow period) 10: Window function enabled mode 2 (the clear enabled period is approximately 50% of the overflow period) 11: Setting disabled (setting of window function enabled mode 2) If the overflow period of the WDT counter is set to 62.5 ms or less in WDT2 to 0 bits, the window function is disabled regardless of setting values of WOVF1 and WOVF0 bits.
2 to 0	WDT2 to WDT0	These bits are used to set the overflow period (T_{WOV}) of the WDT counter. 000: Approx. 7.8 ms 001: Approx. 15.6 ms 010: Approx. 31.3 ms 011: Approx. 62.5 ms 100: Approx. 125 ms 101: Approx. 500 ms 110: Approx. 2 s (initial value) 111: Approx. 8 s

[Note]

- **The overflow period set in WDT2 to WDT0 bits is the time when the WDTCLK is 1.024 kHz. If RC1K oscillation is chosen for the WDTCLK clock, the frequency has a significant error.**
- **If window function enabled mode 1 or window function enabled mode 2 is chosen, no WDT interrupt is generated. A reset is generated in the first overflow.**

10.2.4 Watchdog Timer Counter Register (WDTMC)

This register is a read-only special function register (SFR) to read the WDT counter value.

Address: 0xF014
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	WDTMC															
Byte	WDTMCH								WDTMCL							
Bit	WDTC 15	WDTC 14	WDTC 13	WDTC 12	WDTC 11	WDTC 10	WDTC 9	WDTC 8	WDTC 7	WDTC 6	WDTC 5	WDTC 4	WDTC 3	WDTC 2	WDTC 1	WDTC 0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit name	Description
15 to 0	WDTC15 to WDTC0	These bits are used to read the WDT counter value. The normal counting operation of the WDT counter can be confirmed If values periodically read from the WDT counter vary. Read the value with a cycle slower than 1 kHz, such as low-speed time base counter interrupt, since the WDT counter clock (WDTCLK) is approximately 1 kHz.

[Note]

- **The count value read from the WDT counter are discontinuous due to the hardware structure.**

10.2.5 Watchdog Timer Status Register (WDTSTA)

This register is a read-only special function register (SFR) to indicate the WDT counter clearing state.

Address: 0xF016
Access: R
Access size: 8 bit
Initial value: 0x01

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								WDTSTA							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	WDTCLR2	WDTCLR1	WDTWIN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit No.	Bit name	Description
7 to 3	-	Reserved bits
2	WDTCLR2	This bit is used to read the WDT counter clearing state. It is set to "1" when WDT counter clearing is started and cleared to "0" when clearing is completed. See Figure 10-3 for details of operation waveforms of the WDTCLR2 bit. 0: No WDT counter clearing (initial value) 1: WDT counter clearing in progress
1	WDTCLR1	This bit is used to read the status of acceptance of WDT counter clearing. It is set to "1" when the WDT clear request is accepted in the WDTCON register. It is cleared to "0" when WDT counter clearing is started. While the WDTCLR1 bit is "1", writing to the WDTCON register is invalid. Writing to the WDTCON register should be executed after ensuring that the WDTCLR1 bit is "0". See Figure 10-3 for details of operation waveforms of the WDTCLR1 bit. 0: WDT counter clearing is not pending (initial value) 1: WDT counter clearing is pending
0	WDTWIN	This bit is used to indicate the status of enabling/disabling WDT counter clearing. 0: Clearing is disabled. 1: Clearing is enabled (initial value).

10.3 Description of Operation

The WDT counter starts counting up at the rising edge of the WDT counter operation clock (WDTCLK) chosen by the code option when the system reset is released with operation enabled also by the code option.

The WDT counter can be cleared by writing "0x5A" to the WDTCON register with the WDP bit set to "0", then writing "0xA5" to the WDTCON register with the WDP bit set to "1" while WDT counter clearing is enabled.
The WDP bit is reset to "0" when the system is reset as well as when the WDT counter overflows. It is reversed every time data is written to the WDTCON register.

Two types of use are available: window function disabled mode and window function enabled mode.

- Window function disabled mode
  The WDT counter can always be cleared. The WDT interrupt is generated when the counter overflows for the first time, and the WDT reset is generated when the counter overflows a second time.
- Window function enabled mode
  The periods during which WDT counter clear is enabled and disabled respectively can be set. The WDT reset is generated when the counter overflows for the first time, and the WDT invalid clear reset is generated when the counter is cleared in the period during which WDT counter clear is disabled.

Table 10-2 Watchdog Timer Operation Modes

Mode	Overflow		WDT invalid clear reset
	First	Second	
Window function disabled mode	Interrupt	Reset	-
Window function enabled mode	Reset	-	Reset

The WDT counter overflow period (T_{WOV}) and the WDT counter clear enabled period (T_{WCL}) can be chosen through the WDTMOD register.

The following items can be chosen with the code option. See Chapter 26 "Code Option" for details on how to set the code option.

- Enabling/disabling the WDT timer operation
- Operation clock of the WDT counter (low-speed clock, WDTCLK)

10.3.1 How to Clear WDT Counter

The WDT counter can be cleared by writing "0x5A" to the WDTCON register with the WDP bit set to "0", then writing "0xA5" to the WDTCON register with the WDP bit set to "1" while WDT counter clearing is enabled.
The WDP bit is reset to "0" when the system is reset as well as when the WDT counter overflows. It is reversed every time data is written to the WDTCON register.

The following diagram shows the WDT counter clearing timing chart.

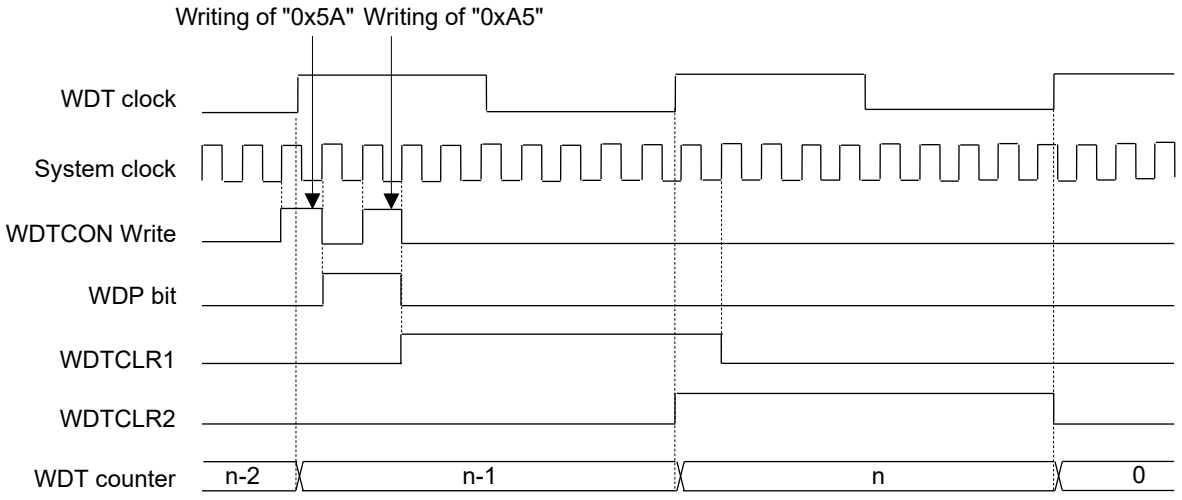


Figure 10-3 WDT Counter Clearing Timing Chart

The following description shows a sample program script of the watchdog timer.

```
void wdt_clear( void )
{
    unsigned char pswval;

    if(WDTCLR1 == 1) {                  // Checking presence of pending clearing process
    return;
    };
    if(WDTCLR2== 1) {                   // Checking whether clearing process is pending or completed
    return;
    };

    pswval = s_drvcommon_getPSW();      // Saving PSW
    _DI();                              // Interrupt disabled (clearing MIE bit)
    do {
        WDTCON = 0x5A;                  // WDT counter clearing
        } while (WDP != 1 );
        WDTCON = 0xA5;

    if ((pswval & 0x08) != 0) {         // Confirming MIE bit
    _EI();                              // Interrupt enabled (setting MIE bit)
    }

    static unsigned char s_drvcommon_getPSW( void ){
    #pragma asm
    mov r0,psw
    rt
    #pragma endasm
    }
```

Figure 10-4 Sample Program Script of Watchdog Timer

[Note]

- **Maximum of two clocks of WDTCLK are required during the period between writing "0x5A", "0xA5" to the WDTCON register and clearing of the WDT counter. To enter the STOP mode or STOP-D mode following WDT clearing, do so after making sure that the WDTCLR1 bit became "0".**
  In addition, if changing the WDTMOD register setting, write to the WDTMOD register after confirming that both of WDTCLR1 and WDTCLR2 bits became "0" as soon as the WDT counter was cleared.
- **In the STOP/STOP-D mode, the WDT timer is stopped.**

10.3.2 Window Function Disabled Mode

In the window function disabled mode, if the WDT counter is not available to clear within the WDT counter overflow period (T_{WOV}) and the counter overflows for the first time, a WDT interrupt is generated. If the WDT counter is not cleared even by the software processing after the WDT interrupt, and overflows again, a WDT reset occurs.
The WDTR bit of the RSTAT register is set to "1" when the WDT reset occurs, and the state on the LSI is transferred to the system reset mode. See Chapter 3 "Reset Function" for details of the RSTAT register.

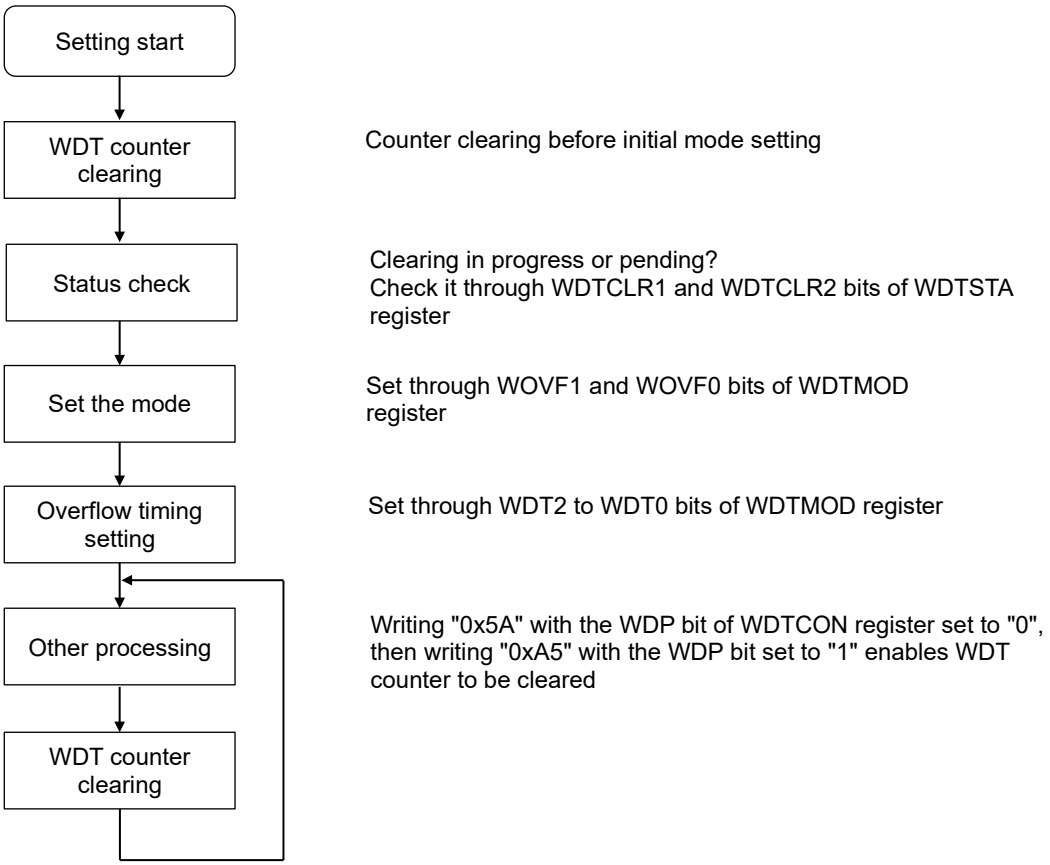


Figure 10-5 Procedure to Use WDT (in Window Function Disabled Mode)

The following figure shows an operation timing overview of the window function disabled mode.

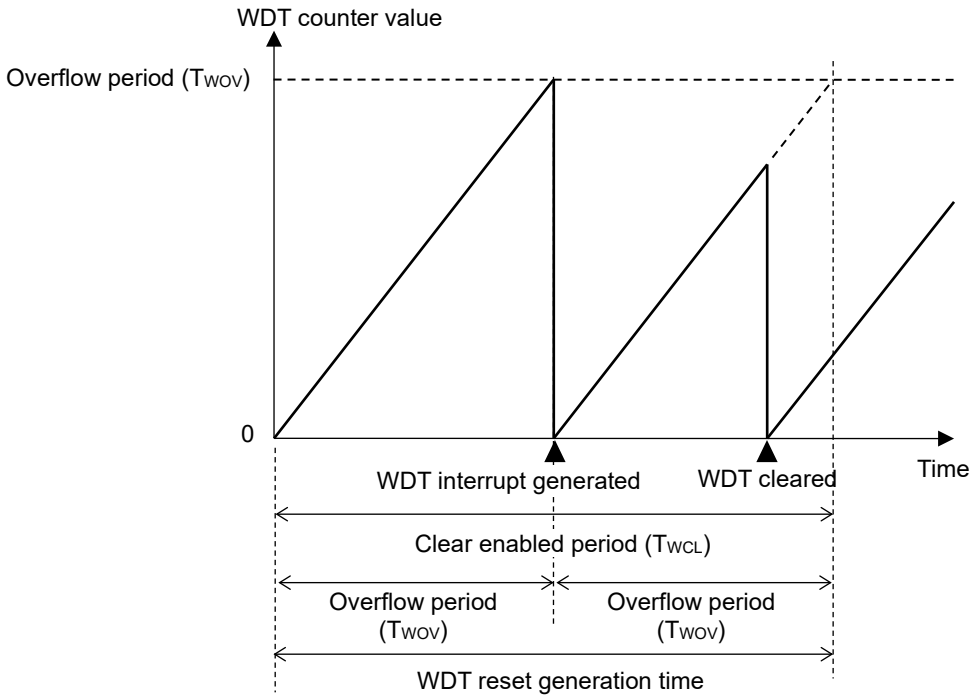


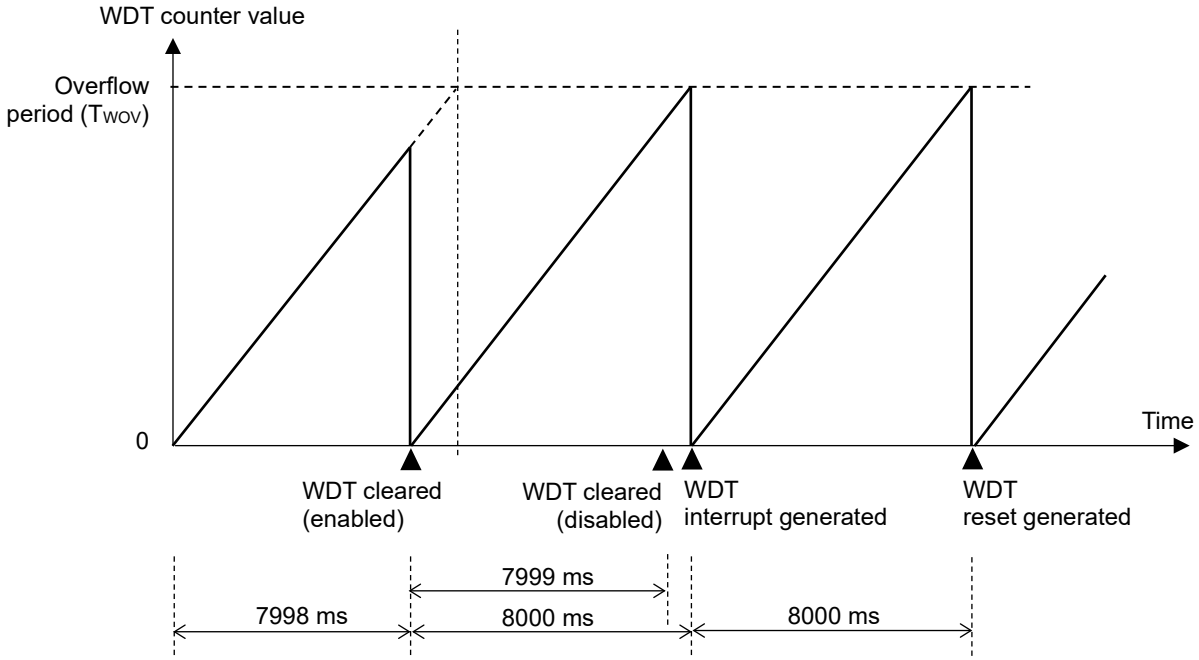
Figure 10-6 Overview of Operation Timing in Window Function Disabled Mode

The following table shows the WDT counter clear enabled period in the window function disabled mode.

Table 10-3 WDT Counter Clear Enabled Period in Window Function Disabled Mode

WDT2	WDT1	WDT0	Overflow period (T_{WOV})*1	WDT reset generation time*1	WDT counter clear enabled period (T_{WCL})*1*2
0	0	0	7.8 ms	15.6 ms	≈ Overflow period
0	0	1	15.6 ms	31.3 ms	≈ Overflow period
0	1	0	31.3 ms	62.5 ms	≈ Overflow period
0	1	1	62.5 ms	125 ms	≈ Overflow period
1	0	0	125 ms	250 ms	≈ Overflow period
1	0	1	500 ms	1000 ms	≈ Overflow period
1	1	0	2000 ms	4000 ms	≈ Overflow period
1	1	1	8000 ms	16000 ms	≈ Overflow period

*1: Time when the WDTCLK is 1 .024kHz. If choosing RC1K oscillation for the WDTCLK, the frequency has a significant error.
*2: The clear processing is enable for two clocks of the WDTCLK (2ms when the WDTCLK is 1.024kHz) before the WDT gets overflowed.



The clear processing is enable for two clocks of the WDTCLK (2ms when the WDTCLK is 1.024kHz) before the WDT gets overflowed. Design the WDT clear timing with time to spare.

Figure 10-7 Example of Operation Timing in Window Function Disabled mode (When Overflow Period=8000 ms)

The following figure shows details of operation timing in the window function disabled mode.

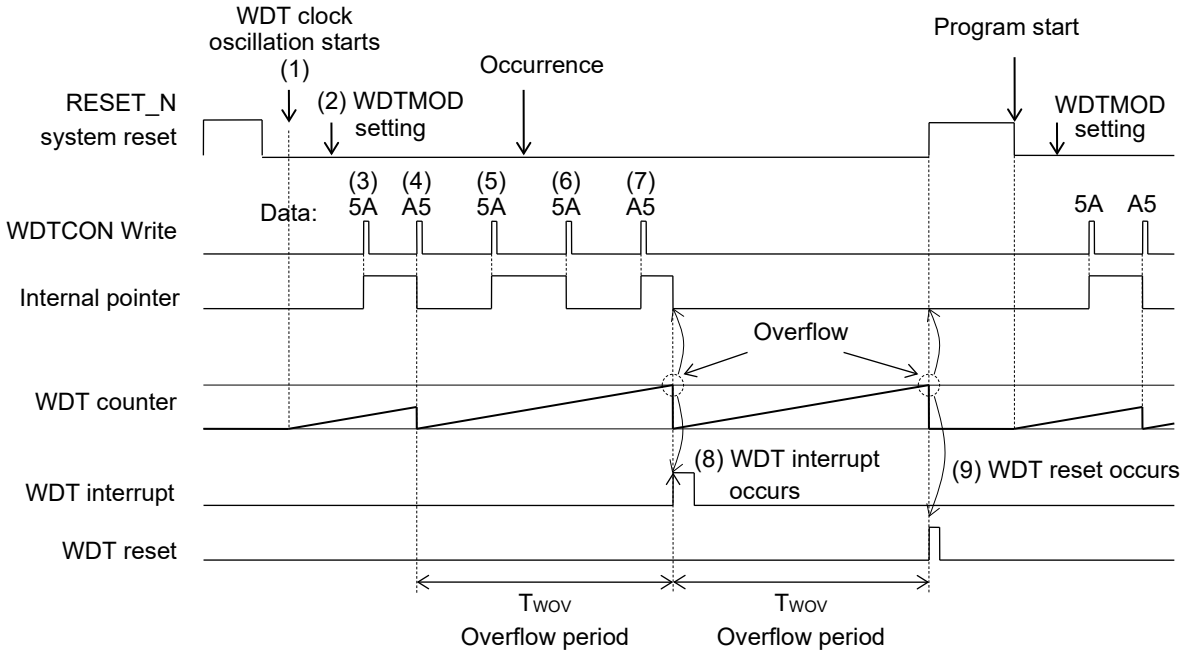


Figure 10-8 Details of Operation Timing in Window Function Disabled Mode

(1) After the system reset is released, the WDT counter starts counting up.

(2) The WDT counter overflow period (T_{WOV}) is set to the WDTMOD register.

(3) "0x5A" is written to the WDTCON register. (Internal pointer WDP: 0 →1)

(4) "0xA5" is written to the WDTCON register to clear the WDT counter. (Internal pointer WDP: 1→0)

(5) "0x5A" is written to the WDTCON register. (Internal pointer WDP: 0 →1)

(6) When "0x5A" is written to the WDTCON register after an abnormality occurred, it is not accepted because the internal pointer WDP is "1". (Internal pointer WDP: 1 →0)

(7) Although "0xA5" is written to the WDTCON register, the WDT counter is not cleared because the internal pointer WDP is "0" and writing of "0x5A" is not accepted in (6). (Internal pointer WDP: 0 →1)

(8) The WDT counter overflows and a WDT interrupt request is generated. (Internal pointer WDP: 1 →0)
Following cleared due to the overflow, the WDT counter continues counting up.

(9) If the WDT counter is not cleared even by the software processing after the WDT interrupt and it overflows again, a WDT reset occurs and the shift to the system reset mode takes place.

10.3.3 Window Function Enabled Mode

In the window function enabled mode, if the WDT counter is not available to clear within the WDT clear enabled period and the counter overflows first time, the WDT overflow reset is generated.
In addition, if the WDT counter is cleared in the period the counter clear is not enabled, the WDT invalid clear reset is generated.

The WDTR bit of the RSTAT register is set to "1" when the WDT overflow reset occurs, and the state on the LSI is transferred to the system reset mode.
The WDTWR bit of the RSTAT register is set to "1" when the WDT invalid clear reset occurs, and the state on the LSI is transferred to the system reset mode. See Chapter 3 "Reset Function" for details of the RSTAT register.

In the window function enabled mode, two types of modes can be chosen through the WDTMOD register:
- Window function enabled mode 1 (the clear enabled period is approximately 75% of the overflow period)
- Window function enabled mode 2 (the clear enabled period is approximately 50% of the overflow period)

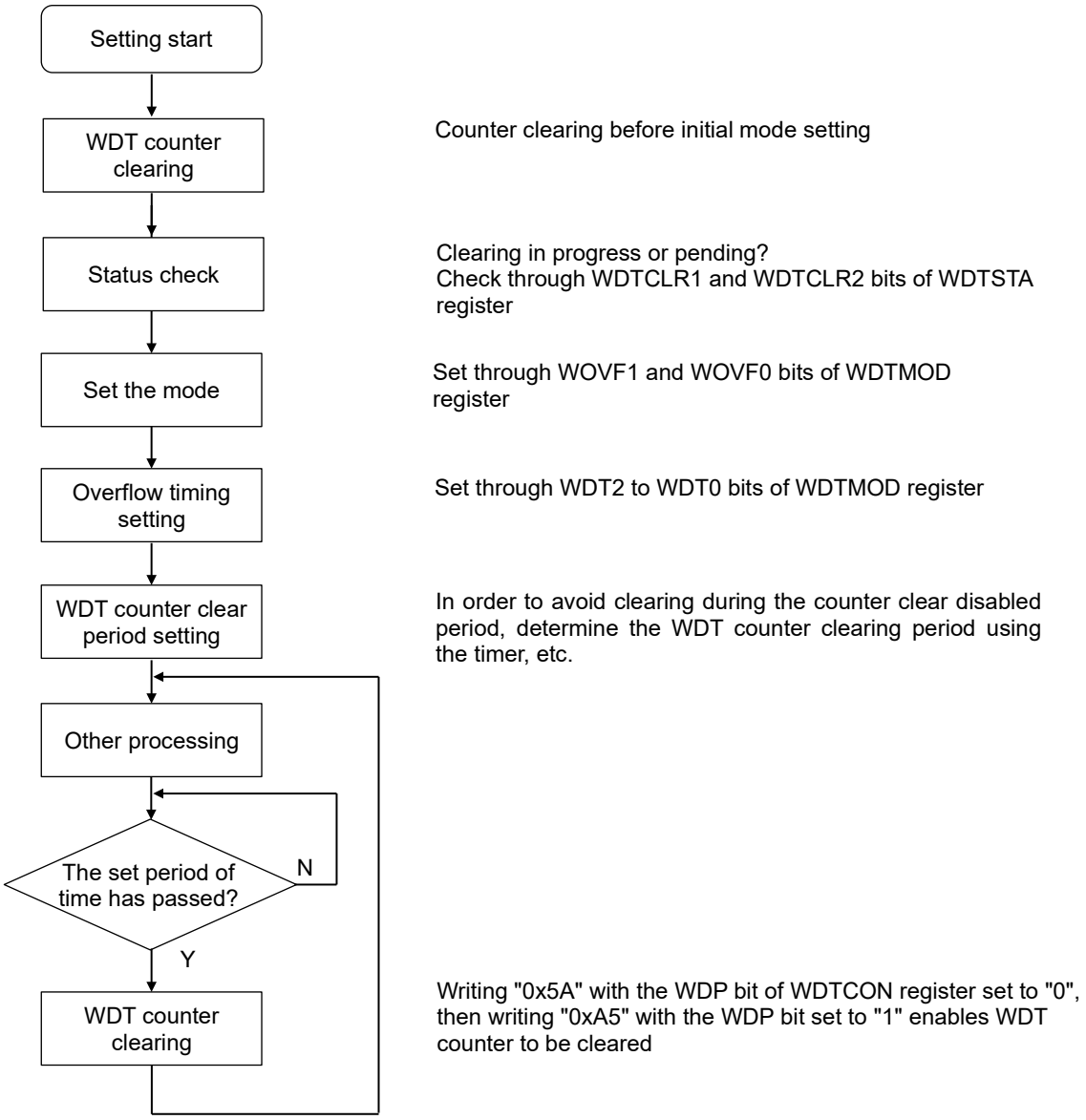


Figure 10-9 Procedure to Use WDT (in Window Function Enabled Mode)

Overviews of the operation of each mode are shown below.

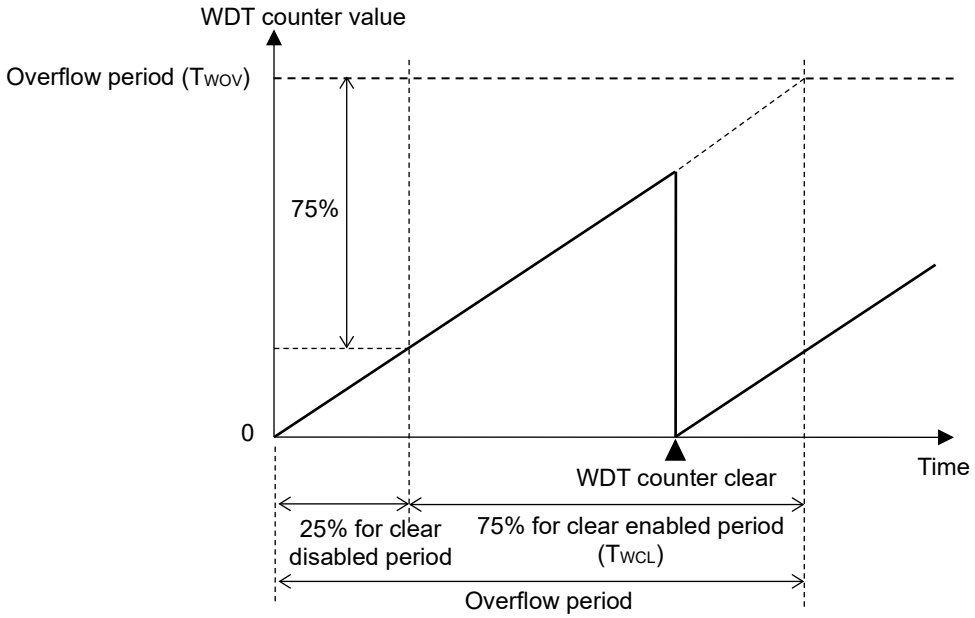


Figure 10-10 Window Function Enabled Mode 1 Operation Overview

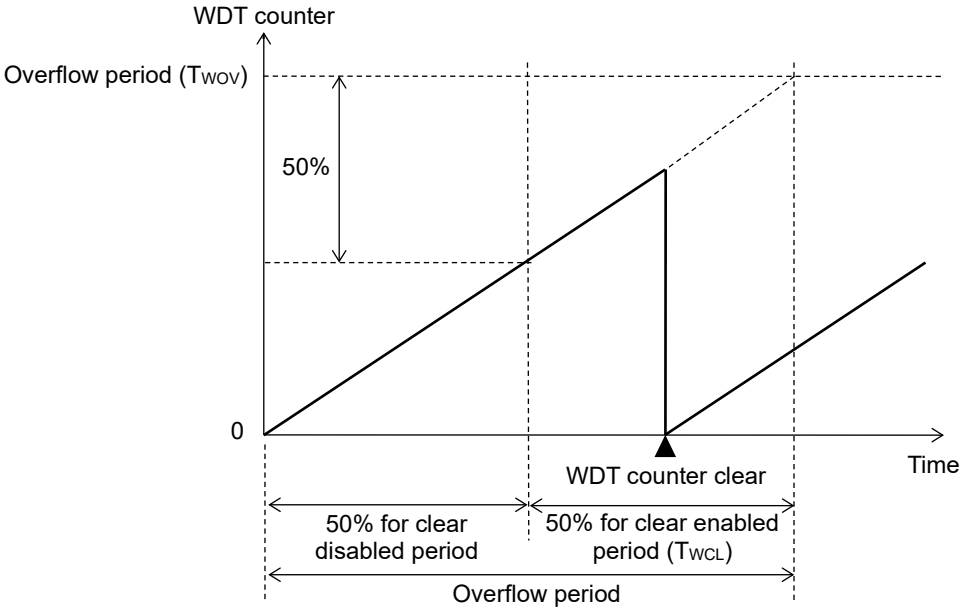


Figure 10-11 Window Function Enabled Mode 2 Operation Overview

The following table shows WDT counter clear enabled periods.
If the overflow period of the WDT counter is set to 62.5 ms or less in WDT2 to 0 bits, the window function is disabled regardless of setting values of WOVF1 and WOVF0 bits.

Table 10-4 WDT Clear Enabled Period in Window Function Enabled Mode 1

WDT2	WDT1	WDT0	Overflow period (T_{WOV})*1	WDT reset generation time*1	WDT clear enabled period (T_{WCL})*1*2
0	0	0	Approx. 7.8 ms	Approx. 7.8 ms	≈ Overflow period
0	0	1	Approx. 15.6 ms	Approx. 15.6 ms	≈ Overflow period
0	1	0	Approx. 31.3 ms	Approx. 31.3 ms	≈ Overflow period
0	1	1	Approx. 62.5 ms	Approx. 62.5 ms	≈ Overflow period
1	0	0	Approx. 125 ms	Approx. 125 ms	≈ 75% of overflow period
1	0	1	Approx. 500 ms	Approx. 500 ms	≈ 75% of overflow period
1	1	0	Approx. 2000 ms	Approx. 2000 ms	≈ 75% of overflow period
1	1	1	Approx. 8000 ms	Approx. 8000 ms	≈ 75% of overflow period

*1: Time when the WDTCLK is 1 .024kHz. For the WDTCLK, select LSCLK through the code option.
*2: The clear processing is enable for two clocks of the WDTCLK (2ms when the WDTCLK is 1.024kHz) before the WDT gets overflowed.

Table 10-5 WDT Counter Clear Enabled Period in Window Function Enabled Mode 2

WDT2	WDT1	WDT0	Overflow period (T_{WOV})*1	WDT reset generation time*1	WDT clear enabled period (T_{WCL})*1*2
0	0	0	Approx. 7.8 ms	Approx. 7.8 ms	≈ Overflow period
0	0	1	Approx. 15.6 ms	Approx. 15.6 ms	≈ Overflow period
0	1	0	Approx. 31.3 ms	Approx. 31.3 ms	≈ Overflow period
0	1	1	Approx. 62.5 ms	Approx. 62.5 ms	≈ Overflow period
1	0	0	Approx. 125 ms	Approx. 125 ms	≈ 50% of overflow period
1	0	1	Approx. 500 ms	Approx. 500 ms	≈ 50% of overflow period
1	1	0	Approx. 2000 ms	Approx. 2000 ms	≈ 50% of overflow period
1	1	1	Approx. 8000 ms	Approx. 8000 ms	≈ 50% of overflow period

*1: Time when the WDTCLK is 1 .024kHz. For the WDTCLK, select LSCLK through the code option.
*2: The clear processing is enable for two clocks of the WDTCLK (2ms when the WDTCLK is 1.024kHz) before the WDT gets overflowed.

[Note]
- **When using the window function enabled mode, always define a WDT interrupt function even though no WDT interrupt occurs.**
- **When using the window function enabled mode, choose "the clock with divided frequency of low-speed oscillation clock (32.768 kHz)" for the WDT count clock with the code option. If "WDT RC1K oscillation clock" is chosen, this function is unusable because the frequency has a significant error.**

The following figure shows details of operation timing in the window function enabled mode.

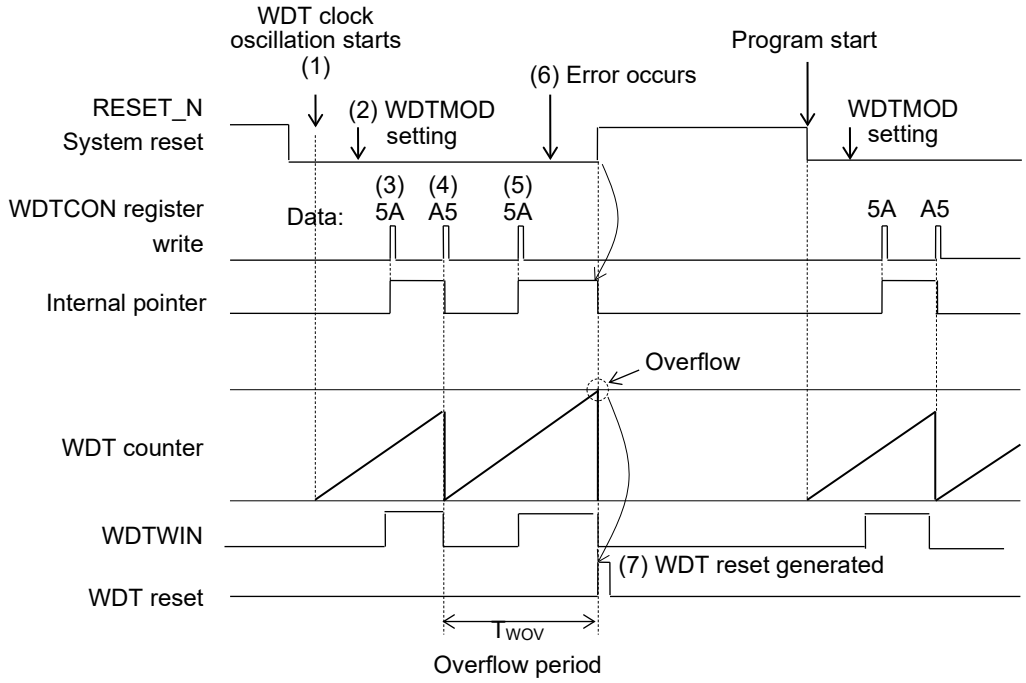


Figure 10-12 Details of Operation Timing in Window Function Enabled Mode

(1) After the system reset is released, the WDT counter starts counting up.
(2) The WDTMOD register is set with the WDT counter overflow period (TWOV) and WDT clear enabled period.
(3) "0x5A" is written to WDTCON during the WDT clear enabled period. (Internal pointer WDP: 0 →1)
(4) "0xA5" is written to the WDTCON register to clear the WDT counter. (Internal pointer WDP: 1 →0)
(5) "0x5A" is written to WDTCON during the WDT clear enabled period. (Internal pointer WDP: 0 →1)
(6) Occurrence of abnormality
(7) The WDT counter overflows and a WDT reset occurs. (Internal pointer WDP: 1 →0)

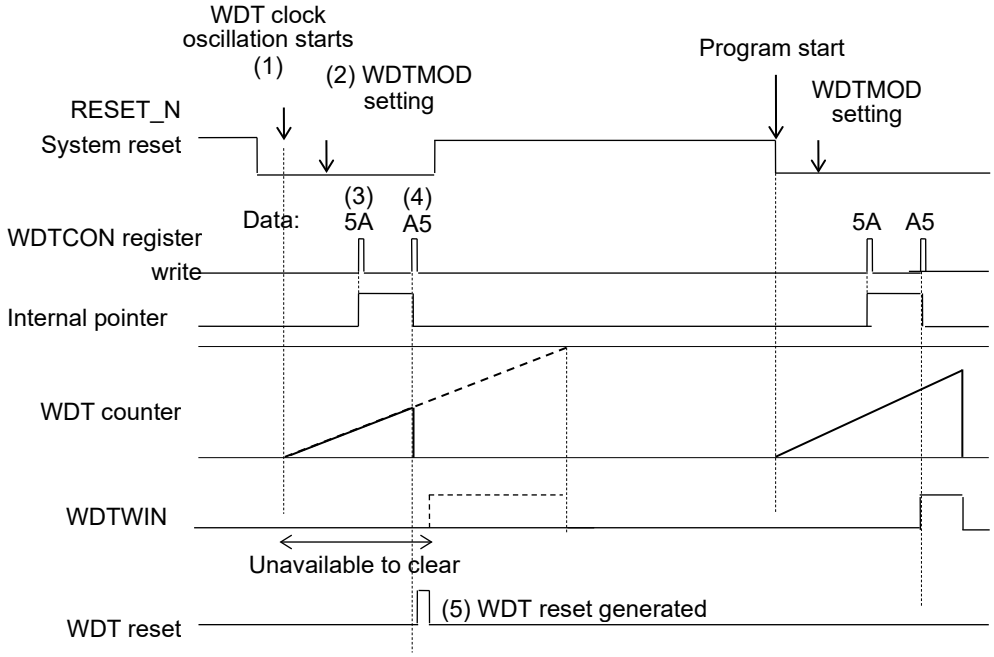


Figure 10-13 WDT invalid clear reset

(1) After the system reset is released, the WDT counter starts counting up.
(2) The WDTMOD register is set with the WDT counter overflow period (TWOV) and WDT clear enabled period.
(3) "0x5A" is written to WDTCON. (Internal pointer WDP: 0 →1)
(4) "0xA5" is written to the WDTCON register to clear the WDT counter. (Internal pointer WDP: 1 →0)
(5) WDT invalid clear reset is occurred by clear processing during the WDT clear disabled period.

[Note]

- **In the watchdog timer (WDT) interrupt function, as the interrupt level (ELEVEL) of the CPU program status word (PSW) becomes "2", the WDT counter is unable to get cleared. Clear the WDT when the ELEVEL is "0" or "1". It is recommended that the WDT counter is cleared at one place in the main loop of the program as a fail-safe.**

Chapter 11 Serial Communication Unit

11. Serial Communication Unit

11.1 General Description

ML62Q1000 Series has two types of the serial communication function.

- 8-bit/16-bit synchronous serial port (SSIO)
- Asynchronous serial interface UART (Universal Asynchronous Receiver Transmitter)

The number of serial communication unit channels is dependent of the product specification.
Table 11-1 shows the number of channels.

Table11-1 Number of Serial Communication Unit channels

Channel no.	ML62Q1300 group				ML62Q1500/ML62Q1800/ML62Q1700 group				
	16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80 pin product	100pin product
0	●	●	●	●	●	●	●	●	●
1	●	●	●	●	●	●	●	●	●
2	−	−	−	−	−	−	−	●	●
3	−	−	−	−	−	−	−	●	●
4	−	−	−	−	−	−	−	●	●
5	−	−	−	−	−	−	−	●	●

●: Available - : Unavailable

11.1.1 Features

Two serial communication modes are available. Table 11-2 shows features of the serial communication.

Table 11-2 Features of the Serial Communication

Serial Communication mode	Operation mode	Features
Synchronous Serial I/O Port (SSIO)	8-bit mode	・ Max. 6ch (Both SSIO and UART are unavailable to use in the same channel) ・ Master mode / Slave mode ・ MSB first / LSB first ・ 8bit / 16bit data length ・ Self-test function using the master and slave modes. For the self-test functions, see Chapter 29 "Safety Function."
	16-bit mode	
UART mode	Half-duplex communication mode	・ 5-bit/6-bit/7-bit/8-bit data length ・ Odd parity/even parity/0 parity/1 parity/and no parity ・ One stop bit/Two stop bits ・ Positive logic/negative logic for communication logic ・ MSB first / LSB first ・ Wide range of communication speed - 1bps to 4,800bps (Clock frequency is 32.768kHz) - 600bps to 3Mbps (Clock frequency is 24MHz) - 300bps to 2Mbps(Clock frequency is 16MHz) ・ Built-in baud rate generator for each channel ・ Parity error flag, overrun error flag, framing error flag, transmission buffer status flag, reception buffer status flag ・ Self-test function using transmission and reception For the self-test functions, see Chapter 29 "Safety Function."
	Full-duplex communication mode	

11.1.2 Configuration

Figure 11-1 shows configuration of the serial communication unit.

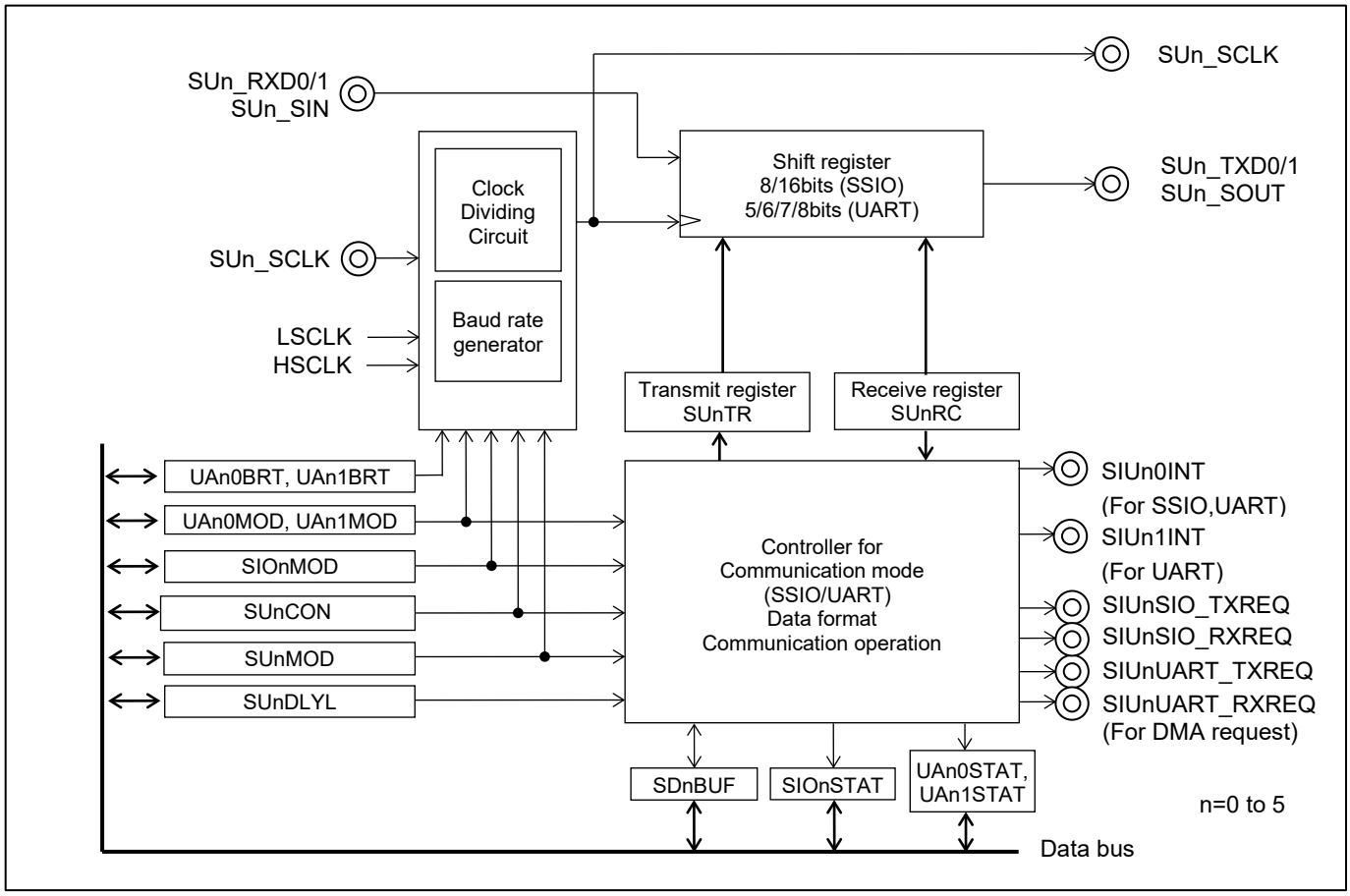


Figure 11-1 Configuration of the Serial Communication Unit

SDnBUF	: Serial communication unit n transmission/reception buffer
SUnMOD	: Serial communication unit n mode register
SUnCON	: Serial communication unit n control register
SUnDLYL	: Serial communication unit n transmission interval setting register
SIOnMOD	: Synchronous serial port n mode register
SIOnSTAT	: Synchronous serial port n status register
UAn0MOD, UAn1MOD	: UARTn0 mode register, UARTn1 mode register
UAn0BRT, UAn1BRT	: UARTn0 baud rate register, UARTn1 baud rate register
UAn0STAT, UAn1STAT	: UARTn0 status register, UARTn1 status register
SIUn0INT	: Serial communication unit n0 Interrupt
SIUn1INT	: Serial communication unit n1 Interrupt
SIUnSIO_TXREQ	: Serial communication unit n SSIO transmission DMA request
SIUnSIO_RXREQ	: Serial communication unit n SSIO reception DMA request
SIUnUART_TXREQ	: Serial communication unit n UART transmission DMA request
SIUnUART_RXREQ	: Serial communication unit n UART reception DMA request

11.1.3 List of Pins

The I/O pins of the serial communication unit are assigned to the shared function of the general ports.

Pin name	I/O	Description
SUn_RXD0	I	Full-duplex data input / half-duplex UART0 data input of serial communication unit n
SUn_RXD1	I	UART1 data input of serial communication unit n
SUn_TXD0	O	UART0 data output of serial communication unit n
SUn_TXD1	O	Full-duplex data output / half-duplex UART1 data output of serial communication unit n
SUn_SCLK	I/O	SSIO synchronous clock input/output of serial communication unit n
SUn_SOUT	O	SSIO transmission data output of serial communication unit n
SUn_SIN	I	SSIO reception data input of serial communication unit n

(n=0 to 5)

Table 11-3 (1) and (2) show the list of the general ports used for the serial communication unit and the register settings of the ports.

Table 11-3(1) Ports used for the serial communication unit and the register settings (UART)

Channel no.	Pin name	Shared port		Setting register	Setting value	ML62Q1300 group				ML62Q1500/ML62Q1800 ML62Q1700 group				
						16pin product	20 pin product	24 pin product	32 pin product	48 pin product	52 pin product	64 pin product	80 pin product	100 pin product
0	SU0_TXD0	P03	2nd Func.	P0MOD3	0001_XXXX*2	●	●	●	●	●	●	●	●	●
		P13	2nd Func.	P1MOD3	0001_XXXX*2	●	●	●	●	●	●	●	●	●
	SU0_RXD0	P02	2nd Func.	P0MOD2	0001_XXXX*1	●	●	●	●	●	●	●	●	●
		P07	3rd Func.	P0MOD7	0010_XXXX*1	-	-	-	●	●	●	●	●	●
		P12	2nd Func.	P1MOD2	0001_XXXX*1	-	-	●	●	●	●	●	●	●
		P17	3rd Func.	P1MOD7	0010_XXXX*1	●	●	●	●	●	●	●	●	●
	SU0_TXD1	P03	3rd Func.	P0MOD3	0010_XXXX*2	●	●	●	●	●	●	●	●	●
		P10	2nd Func.	P1MOD0	0001_XXXX*2	-	-	-	●	●	●	●	●	●
		P13	3rd Func.	P1MOD3	0010_XXXX*2	●	●	●	●	●	●	●	●	●
		P20	2nd Func.	P2MOD0	0001_XXXX*2	●	●	●	●	●	●	●	●	●
	SU0_RXD1	P07	2nd Func.	P0MOD7	0001_XXXX*1	-	-	-	●	●	●	●	●	●
		P17	2nd Func.	P1MOD7	0001_XXXX*1	●	●	●	●	●	●	●	●	●
1	SU1_TXD0	P22	2nd Func.	P2MOD2	0001_XXXX*2	●	●	●	●	●	●	●	●	●
		P25	2nd Func.	P2MOD5	0001_XXXX*2	-	●	●	●	●	●	●	●	●
	SU1_RXD0	P21	2nd Func.	P2MOD1	0001_XXXX*1	●	●	●	●	●	●	●	●	●
		P24	2nd Func.	P2MOD4	0001_XXXX*1	-	●	●	●	●	●	●	●	●
		P26	3rd Func.	P2MOD6	0010_XXXX*1	●	●	●	●	●	●	●	●	●
		P32	3rd Func.	P3MOD2	0010_XXXX*1	-	-	●	●	●	●	●	●	●
	SU1_TXD1	P22	3rd Func.	P2MOD2	0010_XXXX*2	●	●	●	●	●	●	●	●	●
		P25	3rd Func.	P2MOD5	0010_XXXX*2	-	●	●	●	●	●	●	●	●
		P27	2nd Func.	P2MOD7	0001_XXXX*2	●	●	●	●	●	●	●	●	●
		P33	2nd Func.	P3MOD3	0001_XXXX*2	-	●	●	●	●	●	●	●	●
	SU1_RXD1	P26	2nd Func.	P2MOD6	0001_XXXX*1	●	●	●	●	●	●	●	●	●
		P32	2nd Func.	P3MOD2	0001_XXXX*1	-	-	●	●	●	●	●	●	●
2	SU2_TXD0	P57	2nd Func.	P5MOD7	0001_XXXX*2	-	-	-	-	-	-	-	●	●
	SU2_RXD0	P54	3rd Func.	P5MOD4	0010_XXXX*1	-	-	-	-	-	-	-	●	●
		P56	2nd Func.	P5MOD6	0001_XXXX*1	-	-	-	-	-	-	-	●	●
	SU2_TXD1	P55	2nd Func.	P5MOD5	0001_XXXX*2	-	-	-	-	-	-	-	●	●
		P57	3rd Func.	P5MOD7	0010_XXXX*2	-	-	-	-	-	-	-	●	●
	SU2_RXD1	P54	2nd Func.	P5MOD4	0001_XXXX*1	-	-	-	-	-	-	-	●	●
3	SU3_TXD0	P65	2nd Func.	P6MOD5	0001_XXXX*2	-	-	-	-	-	-	-	●	●
	SU3_RXD0	P64	2nd Func.	P6MOD4	0001_XXXX*1	-	-	-	-	-	-	-	●	●
		P67	3rd Func.	P6MOD7	0010_XXXX*1	-	-	-	-	-	-	-	●	●
	SU3_TXD1	P42	2nd Func.	P4MOD2	0001_XXXX*2	-	-	-	-	-	-	-	●	●
		P65	3rd Func.	P6MOD5	0010_XXXX*2	-	-	-	-	-	-	-	●	●
	SU3_RXD1	P67	2nd Func.	P6MOD7	0001_XXXX*1	-	-	-	-	-	-	-	●	●

Channel no.	Pin name	Shared port		Setting register	Setting value	ML62Q1300 group				ML62Q1500/ML62Q1800 ML62Q1700 group				
						16pin product	20 pin product	24 pin product	32 pin product	48 pin product	52 pin product	64 pin product	80 pin product	100 pin product
4	SU4_TXD0	P81	2nd Func.	P8MOD1	0001_XXXX*2	-	-	-	-	-	-	-	●	●
		P94	2nd Func.	P9MOD4	0001_XXXX*2	-	-	-	-	-	-	-	●	●
4	SU4_RXD0	P44	3rd Func.	P4MOD4	0010_XXXX*1	-	-	-	-	-	-	-	●	●
		P52	3rd Func.	P5MOD2	0010_XXXX*1	-	-	-	-	-	-	-	●	●
		P80	2nd Func.	P8MOD0	0001_XXXX*1	-	-	-	-	-	-	-	●	●
		P93	2nd Func.	P9MOD3	0001_XXXX*1	-	-	-	-	-	-	-	●	●
	SU4_TXD1	P45	2nd Func.	P4MOD5	0001_XXXX*2	-	-	-	-	-	-	-	●	●
		P53	2nd Func.	P5MOD3	0001_XXXX*2	-	-	-	-	-	-	-	●	●
		P81	3rd Func.	P8MOD1	0010_XXXX*2	-	-	-	-	-	-	-	●	●
		P94	3rd Func.	P9MOD4	0010_XXXX*2	-	-	-	-	-	-	-	●	●
	SU4_RXD1	P44	2nd Func.	P4MOD4	0001_XXXX*1	-	-	-	-	-	-	-	●	●
		P52	2nd Func.	P5MOD2	0001_XXXX*1	-	-	-	-	-	-	-	●	●
5	SU5_TXD0	P84	2nd Func.	P8MOD4	0001_XXXX*2	-	-	-	-	-	-	-	-	●*3
		PB3	2nd Func.	PBMOD3	0001_XXXX*2	-	-	-	-	-	-	-	●	●
5	SU5_RXD0	P83	2nd Func.	P8MOD3	0001_XXXX*1	-	-	-	-	-	-	-	-	●*3
		PB2	2nd Func.	PBMOD2	0001_XXXX*1	-	-	-	-	-	-	-	●	●
		PB5	3rd Func.	PBMOD5	0010_XXXX*1	-	-	-	-	-	-	-	●	●
	SU5_TXD1	P40	2nd Func.	P4MOD0	0001_XXXX*2	-	-	-	-	-	-	-	●	●
		P84	3rd Func.	P8MOD4	0010_XXXX*2	-	-	-	-	-	-	-	-	●*3
		PB3	3rd Func.	PBMOD3	0010_XXXX*2	-	-	-	-	-	-	-	●	●
	SU5_RXD1	PB5	2nd Func.	PBMOD5	0001_XXXX*1	-	-	-	-	-	-	-	●	●

●: Available to use ●*3: Unavailable on ML62Q1500/ML62Q1800 group and Available on ML62Q1700 group -: Unavailable

*1 : "XXXX" determines the condition of the port input

XXXX	Condition of the port input
0001	Input (without an internal pull-up resistor)
0101	Input (with an internal pull-up resistor)

*2 : "XXXX" determines the condition of the port output

XXXX	Condition of the port output
0010	CMOS output
1010	Nch open drain output (without the pull-up)
1111	Nch open drain output (with the pull-up)

Table 11-3(2) Ports used in the serial communication unit and the register settings (SSIO)

Channel no.	Pin name	Shared port		Setting register	Setting value	ML62Q1300 group				ML62Q1500/ML62Q1800 ML62Q1700 group				
						16pin product	20 pin product	24 pin product	32 pin product	48 pin product	52 pin product	64 pin product	80 pin product	100 pin product
0	SU0_SIN	P02	2nd Func.	P0MOD2	0001_XXXX*1	●	●	●	●	●	●	●	●	●
		P12	2nd Func.	P1MOD2	0001_XXXX*1	-	-	●	●	●	●	●	●	●
	SU0_SCLK	P04	2nd Func.	P0MOD4	0001_XXXX*3	●	●	●	●	●	●	●	●	●
		P11	2nd Func.	P1MOD1	0001_XXXX*3	-	-	-	●	●	●	●	●	●
		P47	2nd Func.	P4MOD7	0001_XXXX*3	-	-	-	-	-	●	●	●	●
	SU0_SOUT	P03	2nd Func.	P0MOD3	0001_XXXX*2	●	●	●	●	●	●	●	●	●
		P13	2nd Func.	P1MOD3	0001_XXXX*2	●	●	●	●	●	●	●	●	●
1	SU1_SIN	P21	2nd Func.	P2MOD1	0001_XXXX*1	●	●	●	●	●	●	●	●	●
		P24	2nd Func.	P2MOD4	0001_XXXX*1	-	●	●	●	●	●	●	●	●
	SU1_SCLK	P16	2nd Func.	P1MOD6	0001_XXXX*3	-	-	●	●	●	●	●	●	●
		P23	2nd Func.	P2MOD3	0001_XXXX*3	●	●	●	●	●	●	●	●	●
	SU1_SOUT	P22	2nd Func.	P2MOD2	0001_XXXX*2	●	●	●	●	●	●	●	●	●
		P25	2nd Func.	P2MOD5	0001_XXXX*2	-	●	●	●	●	●	●	●	●
2	SU2_SIN	P56	2nd Func.	P5MOD6	0001_XXXX*1	-	-	-	-	-	-	-	●	●
	SU2_SCLK	PA3	2nd Func.	PAMOD3	0001_XXXX*3	-	-	-	-	-	-	-	●	●
	SU2_SOUT	P57	2nd Func.	P5MOD7	0001_XXXX*2	-	-	-	-	-	-	-	●	●
3	SU3_SIN	P64	2nd Func.	P6MOD4	0001_XXXX*1	-	-	-	-	-	-	-	●	●
	SU3_SCLK	P66	2nd Func.	P6MOD6	0001_XXXX*3	-	-	-	-	-	-	-	●	●
	SU3_SOUT	P65	2nd Func.	P6MOD5	0001_XXXX*2	-	-	-	-	-	-	-	●	●
4	SU4_SIN	P80	2nd Func.	P8MOD0	0001_XXXX*1	-	-	-	-	-	-	-	●	●
		P93	2nd Func.	P9MOD3	0001_XXXX*1	-	-	-	-	-	-	-	●	●
	SU4_SCLK	P95	2nd Func.	P9MOD5	0001_XXXX*3	-	-	-	-	-	-	-	●	●
		P82	2nd Func.	P8MOD2	0001_XXXX*3	-	-	-	-	-	-	-	●	●
	SU4_SOUT	P81	2nd Func.	P8MOD1	0001_XXXX*2	-	-	-	-	-	-	-	●	●
		P94	2nd Func.	P9MOD4	0001_XXXX*2	-	-	-	-	-	-	-	●	●
5	SU5_SIN	PB2	2nd Func.	PBMOD2	0001_XXXX*1	-	-	-	-	-	-	-	●	●
	SU5_SCLK	PB4	2nd Func.	PBMOD4	0001_XXXX*3	-	-	-	-	-	-	-	●	●
	SU5_SOUT	PB3	2nd Func.	PBMOD3	0001_XXXX*2	-	-	-	-	-	-	-	●	●

●: Available to use -: Unavailable

*1 : "XXXX" determines the condition of the port input

XXXX	Condition of the port
0001	Input (without an internal pull-up resistor)
0101	Input (with an internal pull-up resistor)

*2 : "XXXX" determines the condition of the port output

XXXX	Condition of the port
0010	CMOS output
1010	Nch open drain output (without the pull-up)
1111	Nch open drain output (with the pull-up)

*3 : "XXXX" determines the condition of the port input / output
In the master mode, see to *2 for use as output.
In the slave mode, see to *1 for use as input.

11.1.4 Combination of SSIO port

SUn_SIN, SUn_SOUT and SUn_SCLK are assigned to multiple general ports.
Be sure to use the ports in following combinations.

Combination	Channel	Port			ML62Q1300 group				ML62Q1500/ ML62Q1800 ML62Q1700 group				
		SUn_SIN*	SUn_SOUT*	SUn_SCLK*	16 pin product	20 pin product	24 pin product	32 pin product	48 pin product	52 pin product	64 pin product	80 pin product	100 pin product
1	0	P02	P03	P04	●	●	●	●	●	●	●	●	●
2		P02	P03	P47	-	-	-	-	-	●*1	●*1	●*1	●*1
3		P12	P13	P11	-	-	-	●	●	●	●	●	●
4	1	P21	P22	P16	-	-	●	●	●	●	●	●	●
5		P24	P25	P23	-	●	●	●	●	●	●	●	●
6		P21	P22	P23	●	●	●	●	●	●	●	●	●
7	2	P56	P57	PA3	-	-	-	-	-	-	-	●	●
8	3	P64	P65	P66	-	-	-	-	-	-	-	●	●
9	4	P80	P81	P82	-	-	-	-	-	-	-	●	●
10		P93	P94	P95	-	-	-	-	-	-	-	●	●
11	5	PB2	PB3	PB4	-	-	-	-	-	-	-	●	●

*: n = channel number
●: Available to use, ●*1: Unavailable on ML62Q1500/ML62Q1800 group and Available on ML62Q1700 group, -: Unavailable

11.1.5 Combination of UART port

The pin assignment depends on the communication mode. See Table 11-4 in section 11.2.1 for details of the pin assignment.

11.2 Description of Registers

11.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF600	Serial communication unit 0 transmission/reception 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	-	-	-	-	-
0xF604	Serial communication unit 0 transmission interval setting register	SU0DLYL	-	R/W	8	0x00
0xF605	Reserved	-	-	-	-	-
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	-	-	-	-	-
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	8/16	0x00
0xF61B	Reserved	-	-	-	-	-

Address [H]	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF620	Serial communication unit 1 transmission/reception 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	-	R/W	8	0x00
0xF625	Reserved	-	-	-	-	-
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
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	-	-	-	-	-

Address [H]	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF640	Serial communication unit 2 transmission/reception buffer	SD2BUFL	SD2BUF	R/W	8/16	0x00
0xF641		SD2BUFH		R/W	8	0x00
0xF642	Serial communication unit 2 mode register	SU2MOD	-	R/W	8	0x00
0xF643	Reserved	-	-	-	-	-
0xF644	Serial communication unit 2 transmission interval setting register	SU2DLYL	-	R/W	8	0x00
0xF645	Reserved	-	-	-	-	-
0xF646	Serial communication unit 2 control register	SU2CONL	SU2CON	R/W	8/16	0x00
0xF647		SU2CONH		R/W	8	0x00
0xF648	Synchronous serial port 2 mode register	SIO2MODL	SIO2MOD	R/W	8/16	0x00
0xF649		SIO2MODH		R/W	8	0x00
0xF64A	Synchronous serial port 2 status register	SIO2STAT	-	R/W	8	0x00
0xF64B	Reserved	-	-	-	-	-
0xF64C	UART20 mode register	UA20MODL	UA20MOD	R/W	8/16	0x00
0xF64D		UA20MODH		R/W	8	0x00
0xF64E	UART20 baud rate register	UA20BRTL	UA20BRT	R/W	8/16	0xFF
0xF64F		UA20BRTH		R/W	8	0xFF
0xF650	UART20 baud rate adjustment register	UA20BRC	-	R/W	8	0x00
0xF651	Reserved	-	-	-	-	-
0xF652	UART20 status register	UA20STAT	-	R/W	8	0x00
0xF653	Reserved	-	-	-	-	-
0xF654	UART21 mode register	UA21MODL	UA21MOD	R/W	8/16	0x00
0xF655		UA21MODH		R/W	8	0x00
0xF656	UART21 baud rate register	UA21BRTL	UA21BRT	R/W	8/16	0xFF
0xF657		UA21BRTH		R/W	8	0xFF
0xF658	UART21 baud rate adjustment register	UA21BRC	-	R/W	8	0x00
0xF659	Reserved	-	-	-	-	-
0xF65A	UART21 status register	UA21STAT	-	R/W	8	0x00
0xF65B	Reserved	-	-	-	-	-

Address [H]	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF660	Serial communication unit 3 transmission/reception buffer	SD3BUFL	SD3BUF	R/W	8/16	0x00
0xF661		SD3BUFH		R/W	8	0x00
0xF662	Serial communication unit 3 mode register	SU3MOD	-	R/W	8	0x00
0xF663	Reserved	-	-	-	-	-
0xF664	Serial communication unit 3 transmission interval setting register	SU3DLYL	-	R/W	8	0x00
0xF665	Reserved	-	-	-	-	-
0xF666	Serial communication unit 3 control register	SU3CONL	SU3CON	R/W	8/16	0x00
0xF667		SU3CONH		R/W	8	0x00
0xF668	Synchronous serial port 3 mode register	SIO3MODL	SIO3MOD	R/W	8/16	0x00
0xF669		SIO3MODH		R/W	8	0x00
0xF66A	Synchronous serial port 3 status register	SIO3STAT	-	R/W	8	0x00
0xF66B	Reserved	-	-	-	-	-
0xF66C	UART30 mode register	UA30MODL	UA30MOD	R/W	8/16	0x00
0xF66D		UA30MODH		R/W	8	0x00
0xF66E	UART30 baud rate register	UA30BRTL	UA30BRT	R/W	8/16	0xFF
0xF66F		UA30BRTH		R/W	8	0xFF
0xF670	UART30 baud rate adjustment register	UA30BRC	-	R/W	8	0x00
0xF671	Reserved	-	-	-	-	-
0xF672	UART30 status register	UA30STAT	-	R/W	8	0x00
0xF673	Reserved	-	-	-	-	-
0xF674	UART31 mode register	UA31MODL	UA31MOD	R/W	8/16	0x00
0xF675		UA31MODH		R/W	8	0x00
0xF676	UART31 baud rate register	UA31BRTL	UA31BRT	R/W	8/16	0xFF
0xF677		UA31BRTH		R/W	8	0xFF
0xF678	UART31 baud rate adjustment register	UA31BRC	-	R/W	8	0x00
0xF679	Reserved	-	-	-	-	-
0xF67A	UART31 status register	UA31STAT	-	R/W	8	0x00
0xF67B	Reserved	-	-	-	-	-

Address [H]	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF680	Serial communication unit 4 transmission/reception buffer	SD4BUFL	SD4BUF	R/W	8/16	0x00
0xF681		SD4BUFH		R/W	8	0x00
0xF682	Serial communication unit 4 mode register	SU4MOD	-	R/W	8	0x00
0xF683	Reserved	-	-	-	-	-
0xF684	Serial communication unit 4 transmission interval setting register	SU4DLYL	-	R/W	8	0x00
0xF685	Reserved	-	-	-	-	-
0xF686	Serial communication unit 4 control register	SU4CONL	SU4CON	R/W	8/16	0x00
0xF687		SU4CONH		R/W	8	0x00
0xF688	Synchronous serial port 4 mode register	SIO4MODL	SIO4MOD	R/W	8/16	0x00
0xF689		SIO4MODH		R/W	8	0x00
0xF68A	Synchronous serial port 4 status register	SIO4STAT	-	R/W	8	0x00
0xF68B	Reserved	-	-	-	-	-
0xF68C	UART40 mode register	UA40MODL	UA40MOD	R/W	8/16	0x00
0xF68D		UA40MODH		R/W	8	0x00
0xF68E	UART40 baud rate register	UA40BRTL	UA40BRT	R/W	8/16	0xFF
0xF68F		UA40BRTH		R/W	8	0xFF
0xF690	UART40 baud rate adjustment register	UA40BRC	-	R/W	8	0x00
0xF691	Reserved	-	-	-	-	-
0xF692	UART40 status register	UA40STAT	-	R/W	8	0x00
0xF693	Reserved	-	-	-	-	-
0xF694	UART41 mode register	UA41MODL	UA41MOD	R/W	8/16	0x00
0xF695		UA41MODH		R/W	8	0x00
0xF696	UART41 baud rate register	UA41BRTL	UA41BRT	R/W	8/16	0xFF
0xF697		UA41BRTH		R/W	8	0xFF
0xF698	UART41 baud rate adjustment register	UA41BRC	-	R/W	8	0x00
0xF699	Reserved	-	-	-	-	-
0xF69A	UART41 status register	UA41STAT	-	R/W	8	0x00
0xF69B	Reserved	-	-	-	-	-

Address [H]	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF6A0	Serial communication unit 5 transmission/reception buffer	SD5BUFL	SD5BUF	R/W	8/16	0x00
0xF6A1		SD5BUFH		R/W	8	0x00
0xF6A2	Serial communication unit 5 mode register	SU5MOD	-	R/W	8	0x00
0xF6A3	Reserved	-	-	-	-	-
0xF6A4	Serial communication unit 5 transmission interval setting register	SU5DLYL	-	R/W	8	0x00
0xF6A5	Reserved	-	-	-	-	-
0xF6A6	Serial communication unit 5 control register	SU5CONL	SU5CON	R/W	8/16	0x00
0xF6A7		SU5CONH		R/W	8	0x00
0xF6A8	Synchronous serial port 5 mode register	SIO5MODL	SIO5MOD	R/W	8/16	0x00
0xF6A9		SIO5MODH		R/W	8	0x00
0xF6AA	Synchronous serial port 5 status register	SIO5STAT	-	R/W	8	0x00
0xF6AB	Reserved	-	-	-	-	-
0xF6AC	UART50 mode register	UA50MODL	UA50MOD	R/W	8/16	0x00
0xF6AD		UA50MODH		R/W	8	0x00
0xF6AE	UART50 baud rate register	UA50BRTL	UA50BRT	R/W	8/16	0xFF
0xF6AF		UA50BRTH		R/W	8	0xFF
0xF6B0	UART50 baud rate adjustment register	UA50BRC	-	R/W	8	0x00
0xF6B1	Reserved	-	-	-	-	-
0xF6B2	UART50 status register	UA50STAT	-	R/W	8	0x00
0xF6B3	Reserved	-	-	-	-	-
0xF6B4	UART51 mode register	UA51MODL	UA51MOD	R/W	8/16	0x00
0xF6B5		UA51MODH		R/W	8	0x00
0xF6B6	UART51 baud rate register	UA51BRTL	UA51BRT	R/W	8/16	0xFF
0xF6B7		UA51BRTH		R/W	8	0xFF
0xF6B8	UART51 baud rate adjustment register	UA51BRC	-	R/W	8	0x00
0xF6B9	Reserved	-	-	-	-	-
0xF6BA	UART51 status register	UA51STAT	-	R/W	8	0x00
0xF6BB	Reserved	-	-	-	-	-

Table 11-4 shows lists of the special function register (SFR), the communication pin and the interrupt used in each mode. The communication mode is chosen by the SUnMD1 bit and SUnMD0 bit of SUnMOD register.

Table 11-4 Lists of the special function register (SFR), the communication pin and the interrupt

Item	Register name	Symbol name (Byte)	SSIO port	UART Full-duplex mode	UART Half-duplex mode	
					UARTn1	UARTn0
SFR	Serial Communication Unit n Transmission/Reception Buffer	SDnBUFL	●	● Used as Rx buffer	-	●
		SDnBUFH	● Used in 16-bit mode	● Used as Tx buffer	●	-
	Serial Communication Unit n Mode Register	SUnMOD	●	●	●	●
	Serial Communication Unit n Transmission Interval Setting Register	SUnDLYL	●	● Transmit only	●	●
	Serial Communication Unit n Control Register	SUnCONL	● Use SnEN only	● Use Un0EN only	-	● Use Un0EN only
		SUnCONH	-	-	● Use Un1EN only	-
	Synchronous Serial Port n Mode Register	SIOnMODL SIOnMODH	●	-	-	-
	Synchronous Serial Port n Status Register	SIOnSTAT	●	-	-	-
	UARTn0 Mode Register	UAn0MODL UAn0MODH	-	●	-	●
	UART0 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 I/O pin	-	SUn_SCLK	-	-	-
UART pin	RXD pin	-	-	SUn_RXD0	SUn_RXD1	SUn_RXD0
	TXD pin	-	-	SUn_TXD1	SUn_TXD1	SUn_TXD0
Interrupt	Reception Interrupt	-	SIUn0INT	SIUn0INT	SIUn1INT	SIUn0INT
	Transmission Interrupt	-		SIUn1INT		

●: Used for the communication - : Not used

11.2.2 Serial Communication Unit n Transmission/Reception Buffer (SDnBUF)

SDnBUF is a special function register (SFR) to store transmission/ reception data of the serial communication unit.

Address: 0xF600(SD0BUFL/SD0BUF), 0xF601(SD0BUFH), 0xF620(SD1BUFL/SD1BUF), 0xF621(SD1BUFH), 0xF640(SD2BUFL/SD2BUF), 0xF641(SD2BUFH), 0xF660(SD3BUFL/SD3BUF), 0xF661(SD3BUFH), 0xF680(SD4BUFL/SD4BUF), 0xF681(SD4BUFH), 0xF6A0(SD5BUFL/SD5BUF), 0xF6A1(SD5BUFH)
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	SDnBUF															
Byte	SDnBUFH								SDnBUFL							
Bit	SnB15	SnB14	SnB13	SnB12	SnB11	SnB10	SnB9	SnB8	SnB7	SnB6	SnB5	SnB4	SnB3	SnB2	SnB1	SnB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description	
15 to 8	SnB15 to SnB8	SSIO communication mode	
		8-bit mode	Unused
		16-bit mode	Transmit/Receive data buffer for the upper side 8bit. If writing data into this register, the data is stored into the transmission register (SUnTR). If reading data, the data in the reception data (SUnRC) is read out.
		UART communication mode	
		Full-duplex mode	Transmit data buffer for UARTn. Set the transmission data.
		Half-duplex mode	Transmit/Receive data buffer for UARTn1. Write transmission data in the transmission mode. Received data is stored in the reception mode.
7 to 0	SnB7 to SnB0	SSIO communication mode	
		8-bit mode	Transmit/Receive data buffer. If writing data into this register, the data is stored into the transmission register (SUnTR). If reading data, the data in the reception data (SUnRC) is read out.
		16-bit mode	Transmit/Receive data buffer for the lower side 8bit. If writing data into this register, the data is stored into the transmission register (SUnTR). If reading data, the data in the reception data (SUnRC) is read out.
		UART communication mode	
		Full-duplex mode	Receive data buffer for UARTn. Received data is stored.
		Half-duplex mode	Transmit/Receive data buffer for UARTn0. Write transmission data in the transmission mode. Received data is stored in the reception mode.

- Synchronous Serial I/O (SSIO) port mode
  If writing a data to the SDnBUF register, it also written to the transmission register (SUnTR).
  If reading the SDnBUF resister, data in the reception register (SUnRC) is read out.
  In 8-bit mode, the SDnBUFH is not available to use. The operation of transmission/reception/transmission & reception starts by writing data to the SDnBUFL. In 16-bit mode, The operation of transmission/reception/transmission & reception starts by writing data to the SDnBUFH.

- UART Full-duplex mode
  The SDnBUFL works as the reception buffer and the SDnBUFH works as the transmission buffer.

  Data at the end of reception communication is overwritten into the SDnBUFL, so read out the SDnBUFL by using the serial communication n0 interrupt generated at the end of reception communication.
  Writing to the SDnBUFL is invalid in the Full-duplex communication mode.
  When choosing the 5 to 7 bit length, unused bits return “0” for reading.

  Write transmission data to the SDnBUFH. For continuous transmitting, write the next transmission data to the SDnBUFH after checking Un0FUL bit of UARTn0 status register (UAn0STAT) is “0”. The written data in the SDnBUFH can be read out.
  When choosing the 5 to 7 bit length, written data in unused bits are invalid.

- UART Half-duplex mode
  The SDnBUFL and the SDnBUFH work as the reception buffer or transmission buffer.

  Data received at the end of reception communication is overwritten into the SDnBUFL or SDnBUFH, so read out the registers by using the serial communication n0 interrupt or the serial communication n1 interrupt generated at the end of reception communication.
  Writing to the SDnBUFL or SDnBUFH is invalid in the half-duplex reception mode.
  When choosing the 5 to 7 bit length, unused bits return “0” for reading.

  Write transmission data to the SDnBUFL or SDnBUFH. For continuous transmitting, write the next transmission data to the registers after checking Un0FUL bit of UARTn0 status register (UAn0STAT) or Un1FUL bit of UARTn1 status register (UAn1STAT) is “0”. The written data in the SDnBUFH can be read out .
  When choosing the 5 to 7 bit length, written data in unused bits are invalid.

[Note]

- **In the half-duplex communication mode of UART, be sure to choose the transmission mode by setting Un0IO and Un1IO bit of the UARTn mode register (UAn0MOD, UAn1MOD) before writing the transmission data to SDnBUFL and SDnBUFH.**
- **Do not perform write-operation to the SDnBUF in the SSIO slave reception mode.**

11.2.3 Serial Communication Unit n Mode Register (SUnMOD)

SUnMOD is a special function register (SFR) to choose the communication mode of the serial communication unit.

Address: 0xF602(SU0MOD), 0xF622(SU1MOD), 0xF642(SU2MOD), 0xF662(SU3MOD), 0xF682(SU4MOD), 0xF6A2(SU5MOD)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SUnMOD							
Bit	-	-	-	-	-	-	-	-	SUnINTS	SUnTIMD	SUnRIMD	-	-	-	SUnMD1	SUnMD0
R/W	R	R	R	R	R	R	R	R	R/W	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

Bit No.	Bit symbol name	Description
7	SUnINTS	This bit is used to choose the operation mode of the transmission/reception interrupt. Refer to the description about SUnRIMD bit and SUnTIMD bit.
6	SUnTIMD	This bit is used to choose the timing of transmission interrupt occurrence. When the SUnINTS bit is “0”: 0: The interrupt occurs at the end of data transmission (Initial) 1: The Interrupt occurs at the start and end of data transmission When the SUnINTS bit is “1”: 0: The interrupt occurs at the end of data transmission 1: The Interrupt occurs at the start of data transmission
5	SUnRIMD	This bit is used to choose the timing of reception interrupt occurrence. When the SUnINTS bit is “0”: 0: The interrupt occurs at the end of data reception (Initial) 1: The Interrupt occurs at the start and end of data reception When the SUnINTS bit is “1”: 0: The interrupt occurs at the end of data reception 1: The Interrupt occurs at the start of data reception
4 to 2	-	Reserved bits
1,0	SUnMD1 to SUnMD0	This bit is used to choose the communication mode of the serial communication unit. 00: SSIO mode (Initial value) 01: SSIO mode 10: UART Full-duplex communication mode 11: UART Half-duplex communication mode

[Note]

- **Be sure to set the SUnMOD register while communication is stopped (SUnCON register = 0x00) and do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.**
- **See section 11.3.2.11 “Note on usage of Half-duplex UART” if using “UART Half-duplex communication mode”.**

11.2.4 Serial Communication Unit n Transmission Interval Setting Register (SUnDLYL)

SUnDLYL is a special function register (SFR) to set the transmission frame interval of serial communication. It is used for the slave device to wait for a data reception process when continuously transmitting the serial data.

Address: 0xF604(SU0DLYL), 0xF624(SU1DLYL), 0xF644(SU2DLYL), 0xF664(SU3DLYL), 0xF684(SU4DLYL), 0xF6A4(SU5DLYL)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SUnDLYL							
Bit	-	-	-	-	-	-	-	-	SUnDLY7	SUnDLY6	SUnDLY5	SUnDLY4	SUnDLY3	SUnDLY2	SUnDLY1	SUnDLY0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	SUnDLY7 to SUnDLY0	These bits are used to set the transmission frame interval in the SSIO mode or UART mode.

When SUnDLYL is 0x00, the transmission frame interval is “0”.

When SUnDLYL is not 0x00, the transmission frame interval of serial communication is calculated by the following formula.

In the SSIO mode:
Transmit frame interval =
Transfer clock*1 cycle x ROUNDUP*2 ((Setting value in SUnDLYL register + 2) / dividing ratio of transfer clock)

In the UART mode:
Transmit frame interval =
Base clock*3 cycle x (UAn0BRT+1) x ROUNDUP*2 ((Setting value in SUnDLYL register + 2) / (UAn0BRT+1))

*1 : See section 11.2.6 “Synchronous Serial Port n Mode Register” for details of the transfer clock.
*2 : ROUNDUP = Round the answer up to the nearest whole number
*3 : See section 11.2.8 “UARTn0 Mode Register”, or section 11.2.9 “UARTn1 Mode Register” fot details of the base clock.

[Note]
- **Set “0x00” to the SUnDLYL register in the SSIO slave mode.**
- **The SUnDLYL register is invalid in the SSIO master reception mode.**

11.2.5 Serial Communication Unit n Control Register (SUnCON)

SUnCON is a special function register (SFR) to control the serial communication unit.

Address: 0xF606(SU0CONL/SU0CON), 0xF607(SU0CONH), 0xF626(SU1CONL/SU1CON), 0xF627(SU1CONH), 0xF646(SU2CONL/SU2CON), 0xF647(SU2CONH), 0xF666(SU3CONL/SU3CON), 0xF667(SU3CONH), 0xF686(SU4CONL/SU4CON), 0xF687(SU4CONH), 0xF6A6(SU5CONL/SU5CON), 0xF6A7(SU5CONH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SUnCON															
Byte	SUnCONH								SUnCONL							
Bit	-	-	-	-	-	-	Un1EN	-	-	-	-	-	-	-	Un0EN	SnEN
R/W	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

Bit No.	Bit symbol name	Description
15 to 10	-	Reserved bits
9	Un1EN	This bit is used to enable the UARTn1 communication in the half-duplex mode. ・ UART Half-duplex mode 0: Stop the UARTn1 communication (Initial value) 1: Start the UARTn1 communication ・ UART Full-duplex mode or SSIO mode It is invalid to write this bit.
8 to 2	-	Reserved bits
1	Un0EN	This bit is used to enable the UARTn communication in the UART mode. ・ UART Full-duplex mode 0: Stop the UARTn communication (Initial value) 1: Start the UARTn communication ・ UART Half-duplex mode 0: Stop the UARTn0 communication (Initial value) 1: Start the UARTn0 communication ・ SSIO mode It is invalid to write this bit.
0	SnEN	This bit is used to enable the synchronous serial communication in the SSIO mode. ・ SSIO mode 0: Stop the communication (Initial value) 1: Start the communication ・ UART mode It is invalid to write this bit.

11.2.6 Synchronous Serial Port n Mode Register (SIOnMOD)

SIOnMOD is a special function register (SFR) to set the communication mode of the SSIO port.

Address: 0xF608(SIO0MODL/SIO0MOD), 0xF609(SIO0MODH),
0xF628(SIO1MODL/SIO1MOD), 0xF629(SIO1MODH),
0xF648(SIO2MODL/SIO2MOD), 0xF649(SIO2MODH),
0xF668(SIO3MODL/SIO3MOD), 0xF669(SIO3MODH),
0xF688(SIO4MODL/SIO4MOD), 0xF689(SIO4MODH),
0xF6A8(SIO5MODL/SIO5MOD), 0xF6A9(SIO5MODH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SIOnMOD															
Byte	SIOnMODH								SIOnMODL							
Bit	-	SnNEG	SnCKT	SnCK4	SnCK3	SnCK2	SnCK1	SnCK0	-	-	-	-	SnLG	SnMD1	SnMD0	SnDIR
R/W	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

Bit No.	Bit symbol name	Description
15	-	Reserved bit
14	SnNEG	This bit is used to choose logic of the transfer clock in the SSIO mode. 0: Positive logic (Initial value) 1: Negative logic
13	SnCKT	This bit is used to choose the phase of transfer clock output in the SSIO mode. Four types of communication are available combining the setting of SnNEG bit. 0: Clock type 0: Output with Initial value = "H" level (Initial value) 1: Clock type 1: Output with Initial value = "L" level
12 to 8	SnCK4 to SnCK0	These bits are used to choose the transfer clock of SSIO. When an internal clock is chosen for the transfer clock, the SSIO performs the master mode. When an external clock is chosen, it performs the slave mode. In the slave mode, input the external clock with 1/8 frequency of the system clock or lower. 00000: LSCLK (Initial) 00001: 1/2 LSCLK 10000: 1/1 HSCLK 10001: 1/2 HSCLK 10010: 1/4 HSCLK 10011: 1/8 HSCLK 10100: 1/16 HSCLK 10101: 1/32 HSCLK 10110: 1/64 HSCLK 10111: 1/128 HSCLK 11000: External clock (Slave mode) Others: Do not use (LSCLK)
7 to 4	-	Reserved bits
3	SnLG	This bit is used choose the bit length of the transmission/reception data in the SSIO mode. 0: 8-bit length (Initial value) 1: 16-bit length

Bit No.	Bit symbol name	Description
2 to 1	SnMD1 to SnMD0	These bits are used to choose the transmission/reception mode in the SSIO mode. 00: Transmit/Receive is stopped (Initial value) 01: Reception mode 10: Transmission mode 11: Transmission/Reception mode
0	SnDIR	This bit is used to choose the communication direction in the SSIO mode. 0: LSB first (Initial value) 1: MSB first

[Note]

- **Be sure to set the SIOnMOD register while communication is stopped (SnEN=0) and 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 choosing 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.**

11.2.7 Synchronous Serial Port n Status Register (SIOnSTAT)

SIOnSTAT is s special function register (SFR) to indicate the state of the transmission/reception operation in the SSIO mode.

Address: 0xF60A(SIO0STAT), 0xF62A(SIO1STAT), 0xF64A(SIO2STAT), 0xF66A(SIO3STAT), 0xF68A(SIO4STAT), 0xF6AA(SIO5STAT)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SIOnSTAT							
Bit	-	-	-	-	-	-	-	-	-	SnRFUL	SnRXF	SnTXF	SnFUL	SnTOER	SnROER	SnTUER
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	-	Reserved bit
6	SnRFUL	Do not use
5	SnRXF	This bit is used to indicate receiving data in the SSIO mode. 0: Data reception is stopped (Initial value) 1: Data reception is in progress
4	SnTXF	This bit is used to indicate transmitting data in the SSIO mode. 0: Data transmission is stopped (Initial value) 1: Data transmission is in progress
3	SnFUL	This bit is used to indicate state of the transmission buffer (SDnBUF) in the SSIO transmission mode. This 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. When the SnEN bit is set to "1" on the condition of SnFUL is “0”, the transmission does not start until a data is written to the SDnBUF register. When writing data to SDnBUF on the condition of SnFUL is “1”, the SDnBUF register is overwritten. 0: Transmission buffer has no data (initial value) 1: Transmission buffer has data
2	SnTOER	This bit is used to indicate a transmission overrun error. If writing a data to SDnBUF register when the SnFUL bit is “1”, the SnTOER bit is set to “1”. To reset the SnTOER bit, write “1” to this bit. 0: There was no transmission overrun error (Initial value) 1: There was a transmission overrun error
1	SnROER	This bit is used to indicate a reception overrun error. If receiving the next data before reading the data in the SDnBUFL register. To reset the SnROER bit, write “1” to this bit. The SnTOER bit is fixed to “0” in the transmission mode. 0: There was no reception overrun error (Initial value) 1: There was a reception overrun error

Bit No.	Bit symbol name	Description
0	SnTUER	This bit is used to indicate a transmission underrun error. This bit is set to “1” by transmitting start when the SnFUL bit is “0”. In the clock type 1 slave mode, this bit is set to “1”, when the SnEN bit is set to “1” or each transmission completed while the SnFUL bit is “0”. See section 11.3.1.6 “Timing in Clock Type 1 Slave Mode” for details of update timing of the SnTUER bit. To reset the SnTUER bit, write “1” to this bit. The SnTUER bit is fixed to “0” in the reception mode. 0: There was no transmission underrun error (Initial value) 1: There was a transmission underrun error

[Note]
- **Update the SnTUER, SnROER and SnTOER bits by using a byte access.**
- **Do not write the SnTUER, SnROER and SnTOER bits while the transmission/reception in progress (SnTXF=1 / SnRXF=1).**

11.2.8 UARTn0 Mode Register (UAn0MOD)

UAn0MOD is a special function register (SFR) to set the mode in UARTn0 full-duplex communication mode and half-duplex communication mode.

Address: 0xF60C(UA00MODL/UA00MOD), 0xF60D(UA00MODH),
0xF62C(UA10MODL/UA10MOD), 0xF62D(UA10MODH),
0xF64C(UA20MODL/UA20MOD), 0xF64D(UA20MODH),
0xF66C(UA30MODL/UA30MOD), 0xF66D(UA30MODH),
0xF68C(UA40MODL/UA40MOD), 0xF68D(UA40MODH),
0xF6AC(UA50MODL/UA50MOD), 0xF6AD(UA50MODH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	UAn0MOD															
Byte	UAn0MODH								UAn0MODL							
Bit	Un0DIR	Un0NEG	Un0STP	Un0PT2	Un0PT1	Un0PT0	Un0LG1	Un0LG0	Un0RSS	-	-	-	-	Un0CK1	Un0CK0	Un0IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	Un0DIR	This bit is used to choose the communication direction in UARTn0 full-duplex and half-duplex mode. 0: LSB first (Initial value) 1: MSB first
14	Un0NEG	This bit is used to choose logic of the data input / output in UARTn0 full-duplex and half-duplex mode. 0: Positive logic (Initial value) 1: Negative logic
13	Un0STP	This bit is used to choose the stop bit length in UARTn0 full-duplex and half-duplex mode. 0: 1 stop bit (Initial value) 1: 2 stop bit
12 to 10	Un0PT2 to Un0PT0	These bits are used to choose the parity bit in UARTn0 full-duplex and half-duplex mode. 000: No parity bit (Initial value) 001: Odd parity 010: No parity bit 011: Even parity 100: No parity bit 101: Parity bit is fixed to "1" 110: No parity bit 111: Parity bit is fixed to "0"
9,8	Un0LG1 to Un0LG0	These bits are used to choose the communication data length in UARTn0 full-duplex and half-duplex mode. 00: 8-bit length (Initial value) 01: 7-bit length 10: 6-bit length 11: 5-bit length
7	Un0RSS	This bit is used to choose sampling timing of the reception data in UARTn0 full-duplex and half-duplex mode. 0: (Values set to UAn0BRTH and UAn0BRTL registers)/2 (Initial value) 1: {(Values set to UAn0BRTH and UAn0BRTL registers)/2} -1

Bit No.	Bit symbol name	Description
6 to 3	-	Reserved bits
2,1	Un0CK1 to Un0CK0	These bits are used to choose base clock of baud rate generator in UARTn0 full-duplex and half-duplex mode. 00: LSCLK (initial value) 01: Do not use (LSCLK) 10: HSCLK 11: Do not use (HSCLK)
0	Un0IO	This bit is used to choose the transmission mode or reception mode in UARTn0 full-duplex and half-duplex mode. When the full-duplex communication mode is chosen, this bit is fixed to “1”, and writing to this bit is ignored. 0: Transmission mode (Initial value) 1: Reception mode

[Note]

- **Be sure to set the UAn0MOD register while communication is stopped (Un0EN=0). 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)

UAn1MOD is a special function register (SFR) to set the transfer mode in UARTn1 half-duplex communication mode. When the full-duplex communication is chosen, no need to specify UAn1MOD register.

Address: 0xF614(UA01MODL/UA01MOD), 0xF615(UA01MODH),
0xF634(UA11MODL/UA11MOD), 0xF635(UA11MODH),
0xF654(UA21MODL/UA21MOD), 0xF655(UA21MODH),
0xF674(UA31MODL/UA31MOD), 0xF675(UA31MODH),
0xF694(UA41MODL/UA41MOD), 0xF695(UA41MODH),
0xF6B4(UA51MODL/UA51MOD), 0xF6B5(UA51MODH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	UAn1MOD															
Byte	UAn1MODH								UAn1MODL							
Bit	Un1DIR	Un1NEG	Un1STP	Un1PT2	Un1PT1	Un1PT0	Un1LG1	Un1LG0	Un1RSS	-	-	-	-	Un1CK1	Un1CK0	Un1IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	Un1DIR	This bit is used to choose the communication direction in UARTn1 half-duplex mode. 0: LSB first (Initial value) 1: MSB first
14	Un1NEG	This bit is used to choose logic of the data input / output data in UARTn1 half-duplex mode. 0: Positive logic (Initial value) 1: Negative logic
13	Un1STP	This bit is used to choose the stop bit length in UARTn1 half-duplex mode. 0: 1 stop bit (Initial value) 1: 2 stop bit
12 to 10	Un1PT2 to Un1PT0	These bits are used to choose the parity bit in UARTn1 half-duplex mode. 000: No parity bit (Initial value) 001: Odd parity 010: No parity bit 011: Even parity 100: No parity bit 101: Parity bit is fixed to "1" 110: No parity bit 111: Parity bit is fixed to "0"
9,8	Un1LG1 to Un1LG0	These bits are used to choose the communication data length in UARTn1 half-duplex mode. 00: 8-bit length (Initial value) 01: 7-bit length 10: 6-bit length 11: 5-bit length
7	Un1RSS	This bit is used to choose the sampling timing of the reception data in UARTn1 half-duplex mode. 0: (Values set to UAn1BRTH and UAn1BRTL registers)/2 (Initial value) 1: {(Values set to UAn1BRTH and UAn1BRTL registers)/2} -1
6 to 3	-	Reserved bits

Bit No.	Bit symbol name	Description
2,1	Un1CK1 to Un1CK0	These bits are used to choose the base clock of baud rate generator in UARTn1 half-duplex mode. 00: LSCLK (Initial value) 01: Do not use (LSCLK) 10: HSCLK 11: Do not use (HSCLK)
0	Un1IO	This bit is used to choose the transmission mode or reception mode in UARTn1 full-duplex mode. When the full-duplex communication mode is chosen, this bit is fixed to “0” and writing to this bit is ignored. 0: Transmission mode (Initial value) 1: Reception mode

[Note]

- **Be sure to set the UAn1MOD register while communication is stopped (Un1EN=0). Do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.**

11.2.10 UARTn0 Baud Rate Register (UAn0BRT)

UAn0BRT is a special function register (SFR) to set the count value of the baud rate generator in UARTn0 full-duplex communication mode and half-duplex communication mode.
For details of relation between the count value of the baud rate generator and the baud rate, see Section "11.3.2.2 “Baud Rate”.

Address: 0xF60E(UA00BRTL/UA00BRT), 0xF60F(UA00BRTH),
0xF62E(UA10BRTL/UA10BRT), 0xF62F(UA10BRTH),
0xF64E(UA20BRTL/UA20BRT), 0xF64F(UA20BRTH),
0xF66E(UA30BRTL/UA30BRT), 0xF66F(UA30BRTH),
0xF68E(UA40BRTL/UA40BRT), 0xF68F(UA40BRTH),
0xF6AE(UA50BRTL/UA50BRT), 0xF6AF(UA50BRTH)
Access: R/W
Access size: 8/16bit
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	UAn0BRT															
Byte	UAn0BRTH								UAn0BRTL							
Bit	Un0BR15	Un0BR14	Un0BR13	Un0BR12	Un0BR11	Un0BR10	Un0BR9	Un0BR8	Un0BR7	Un0BR6	Un0BR5	Un0BR4	Un0BR3	Un0BR2	Un0BR1	Un0BR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.2.11 UARTn1 Baud Rate Register (UAn1BRT)

UAn1BRT is a special function register (SFR) to set the count value of the baud rate generator in UARTn1 half-duplex communication mode. No need to specify UAn1BRT when using the full-duplex communication mode.
For details of relation between the count value of the baud rate generator and the baud rate, see Section 11.3.2.2 “Baud Rate”.

Address: 0xF60E(UA01BRT/UA01BRTL), 0xF60F(UA01BRTH),
0xF62E(UA11BRT/UA11BRTL), 0xF62F(UA11BRTH),
0xF64E(UA21BRT/UA21BRTL), 0xF64F(UA21BRTH),
0xF66E(UA31BRT/UA31BRTL), 0xF66F(UA31BRTH),
0xF68E(UA41BRT/UA41BRTL), 0xF68F(UA41BRTH),
0xF6AE(UA51BRT/UA51BRTL), 0xF6AF(UA51BRTH)
Access: R/W
Access size: 8/16bit
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	UAn1BRT															
Byte	UAn1BRTH								UAn1BRTL							
Bit	Un1BR15	Un1BR14	Un1BR13	Un1BR12	Un1BR11	Un1BR10	Un1BR9	Un1BR8	Un1BR7	Un1BR6	Un1BR5	Un1BR4	Un1BR3	Un1BR2	Un1BR1	Un1BR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

[Note]

- **Be sure to set the UAn0BRT and UAn1BRT register while communication is stopped (Un0EN=0, Un1EN=0). Do not rewrite it during communication.**

11.2.12 UARTn0 Baud Rate Adjustment Register (UAn0BRC)

UAn0BRC is a special function register (SFR) to adjust the count value of the baud rate generator in UARTn0 full-duplex communication mode and half-duplex communication mode.
For details of relation between the value of UAn0BRC and the correction value, see Section 11.3.2.2 “Baud Rate”.

Address: 0xF610(UA00BRC), 0xF630(UA10BRC), 0xF650(UA20BRC), 0xF670(UA30BRC), 0xF690(UA40BRC), 0xF6B0(UA50BRC)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAn0BRC							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	Un0BRC2	Un0BRC1	Un0BRC0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.2.13 UARTn1 Baud Rate Adjustment Register (UAn1BRC)

UAn1BRC is a special function register (SFR) to adjust the count value of the baud rate generator in UARTn1 half-duplex communication mode.
No need to specify UAn1MOD when using the full-duplex communication mode.
For details of relation between the value of UAn1BRC and the correction value, see Section 11.3.2.2 “Baud Rate."

Address: 0xF618(UA01BRC), 0xF638(UA11BRC), 0xF658(UA21BRC), 0xF678(UA31BRC), 0xF698(UA41BRC), 0xF6B8(UA51BRC)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAn1BRC							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	Un1BRC2	Un1BRC1	Un1BRC0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

- **Be sure to set the UAn0BRC and UAn1BRC register while communication is stopped (Un0EN=0, Un1EN=0). Do not rewrite it during communication.**

11.2.14 UARTn0 Status Register (UAn0STAT)

UAn0STAT is a special function register (SFR) to indicate the state in the transmit/receive operation in UARTn0 full-duplex communication mode and half-duplex communication mode.

Address: 0xF612(UA00STAT), 0xF632(UA10STAT), 0xF652(UA20STAT), 0xF672(UA30STAT), 0xF692(UA40STAT), 0xF6B2(UA50STAT)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAn0STAT							
Bit	-	-	-	-	-	-	-	-	-	-	Un0RXF	Un0TXF	Un0FUL	Un0PER	Un0OER	Un0FER
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7, 6	-	Reserved bits
5	Un0RXF	This bit is used to indicate the UART is receiving data in UARTn0 full-duplex communication mode and half-duplex communication mode. 0: Data reception is stopped (Initial value) 1: Data reception is in progress
4	Un0TXF	This bit is used to indicate the UART is transmitting data in UARTn0 full-duplex communication mode and half-duplex communication mode. 0: Data transmission is stopped (Initial value) 1: Data transmission is in progress
3	Un0FUL	This bit is used to indicate the state of the transmission/reception buffer in UARTn0 full-duplex communication mode and half-duplex communication mode. When the full-duplex communication mode is chosen, this bit becomes "1" when transmission data is written to the SD0BUFH and becomes "0" when the transmission data is transferred to the shift register. To transmit data successively, check that the Un0FUL bit is "0" before writing write the next transmit data to the SDn0BUFH. When the half-duplex mode is chosen, this bit becomes "1" when transmission data is written to the SDn0BUFL register in the transmission mode and becomes "0" when the transmit data is transferred to the shift register. To transmit data successively, check that the Un0FUL bit is "0" before writing the next transmission data to the SD0BUFL. The Un0FUL bit is forcibly reset to “0” by writing “1” to this bit. The Un0FUL bit is fixed to "0" in the reception mode. ・ Full-duplex communication mode 0: No data in the SDnBUFH (Initial value) 1: There is data in the SDnBUFH ・ Half-duplex communication mode 0: No data in the SDnBUFL (Initial value) 1: There is data in the SDnBUFL
2	Un0PER	This bit is used to indicate a parity error in UARTn0 full-duplex communication mode and 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, this bit becomes "1". The Un0PER bit is forcibly reset to "0" by writing “1” to this bit. The Un0PER is fixed to "0" in the transmission mode. 0: The parity error has not occurred (Initial value) 1: The parity error has occurred

Bit No.	Bit symbol name	Description
1	Un0OER	This bit is used to indicate an overrun error in UARTn0 full-duplex communication mode and half-duplex communication mode. This bit becomes "1" if the next data is received before reading the previous receive data in the Serial Communication Unit n Transmit/Receive Buffer L (SDnBUFL). The Un0OER bit is forcibly reset to "0" by writing “1” to this bit. Write “1” to Un0OER when Un0OER bit becomes “1” or after reading reception buffer for detecting overrun error correctly. The Un0OER is fixed to "0" in the transmission mode. 0: The overrun error has not occurred (Initial value) 1: The overrun error has occurred
0	Un0FER	This bit is used to indicate a framing error in UARTn0 full-duplex communication mode and half-duplex communication mode. This bit becomes "1" when an error occurs in the stop bit. The Un0FER bit is forcibly reset to "0" by writing “1” to this bit. The Un0FER is fixed to "0" in the transmission mode. 0: The framing error has not occurred (Initial value) 1: The framing error has occurred

[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 be sure to read the SDnBUFL even if the data is unnecessary when the reception is completed.**
- **When an error occurs in the start bit, the state returns to the reception waiting state.**
- **Write the Un0FER bit, Un0OER bit, Un0PER bit and Un0FUL bit by a byte-access.**

11.2.15 UARTn1 Status Register (UAn1STAT)

UAn1STAT is a special 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 chosen, the contents of the UAn1STAT register is invalid.

Address: 0xF61A(UA01STAT), 0xF63A(UA11STAT), 0xF65A(UA21STAT), 0xF67A(UA31STAT), 0xF69A(UA41STAT), 0xF6BA(UA51STAT)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAn1STAT							
Bit	-	-	-	-	-	-	-	-	-	-	Un1RXF	Un1TXF	Un1FUL	Un1PER	Un1OER	Un1FER
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7, 6	-	Reserved bits
5	Un1RXF	This bit is used to indicate the UART1 is receiving data in UARTn1 half-duplex communication mode. 0: Data reception is stopped (Initial value) 1: Data reception is in progress
4	Un1TXF	This bit is used to indicate the UART1 is transmitting data in UARTn0 half-duplex communication mode. 0: Data transmission is stopped (Initial value) 1: Data transmission is in progress
3	Un1FUL	This bit is used to indicate the state of the transmission/reception buffer in UARTn1 half-duplex communication mode. When the half-duplex communication mode is chosen, this bit becomes "1" when transmission data is written to the SD0BUFH and becomes "0" when the transmission data is transferred to the shift register. To transmit data successively, check that the Un1FUL bit is "0" before writing write the next transmit data to the SDn0BUFH. The Un1FUL bit is forcibly reset to “0” by writing “1” to this bit. The Un1FUL bit is fixed to "0" in the reception mode. 0: No data in the SD0BUFH (Initial value) 1: There is data in the SD0BUFH
2	Un1PER	This bit is used 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, this bit becomes "1". The Un1PER bit is forcibly reset to "0" by writing “1” to this bit. The Un1PER is fixed to "0" in the transmission mode. 0: The parity error has not occurred (Initial value) 1: The parity error has occurred

Bit No.	Bit symbol name	Description
1	Un1OER	This bit is used to indicate an overrun error in UARTn1 half-duplex communication mode. This bit becomes "1" if the next data is received before reading the previous receive data in the Serial Communication Unit n Transmit/Receive Buffer H (SDnBUFH). The Un1OER bit is forcibly reset to "0" by writing “1” to this bit. Write “1” to Un1OER when Un1OER bit becomes “1” or after reading reception buffer for detecting overrun error correctly. The Un1OER is fixed to "0" in the transmission mode. 0: The overrun error has not occurred (Initial value) 1: The overrun error has occurred
0	Un1FER	This bit is used to indicate a framing error in UARTn1 half-duplex communication mode. This bit becomes "1" when an error occurs in the stop bit. The Un1FER bit is forcibly reset to "0" by writing “1” to this bit. The Un1FER is fixed to "0" in the transmission mode. 0: The framing error has not occurred (Initial value) 1: The framing error has occurred

[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 be sure to read the SDnBUFH even if the data is not necessary when the reception is completed.**
- **When an error occurs in the start bit, the state returns to the reception waiting state.**
- **Write the Un1FER bit, Un1OER bit, Un1PER bit and Un1FUL bit by a 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 transmission 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).
The SnFUL bit is "0" at the start of transmission, and then the SnTXF bit is "1" one cycle after the system clock.
In the clock type1 slave mode, a timing that SnTXF becomes to "1" is different from other mode/clock type. See section 11.3.1.6 “Timing in Clock Type 1 Slave Mode” for details of its timing.
Stop communication (SnEN="0") need to be done after confirming that transmission is complete. Where is to confirm the completion of transmission, confirm sequetialy that SnFUL is "0" and then that SnTXF is "0".
After the completion of transmission, the value of the SUn_SOUT will be "1" with 16-bit length, will be the bit 8 of SDnBUF register with 8-bit length, LSB first, or will be "0" with 8-bit length, MSB first.

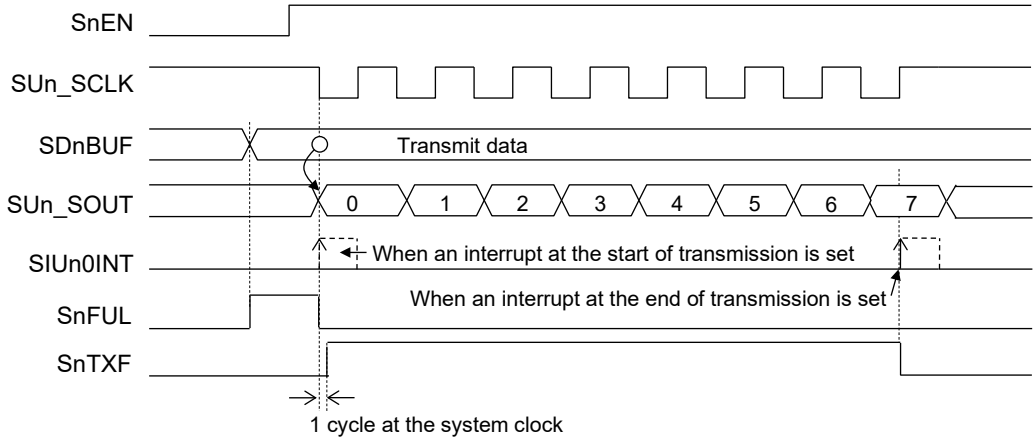


Figure 11-2 Transmit Operation Waveforms of Synchronous Serial Port for Clock Type 0 (Positive Logic)
(8-Bit Length, LSB First)

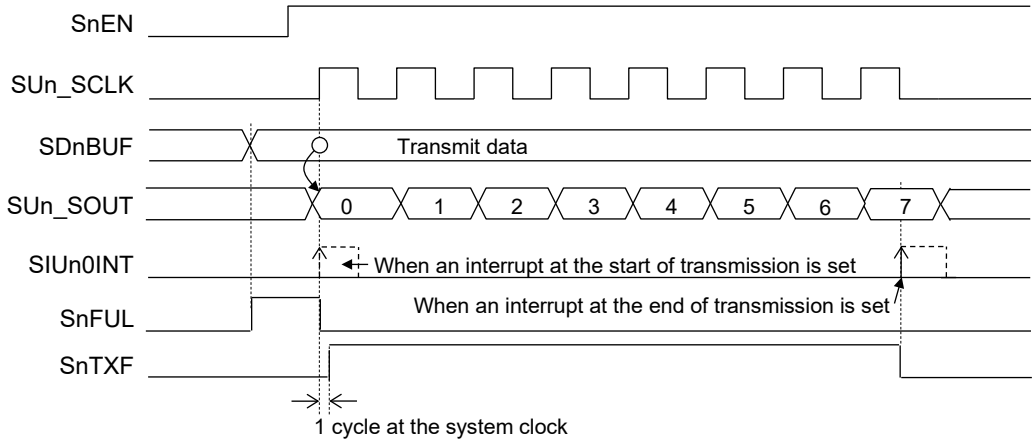


Figure 11-3 Transmit Operation Waveforms of Synchronous Serial Port for Clock Type 0 (Negative Logic)
(8-Bit Length, LSB First)

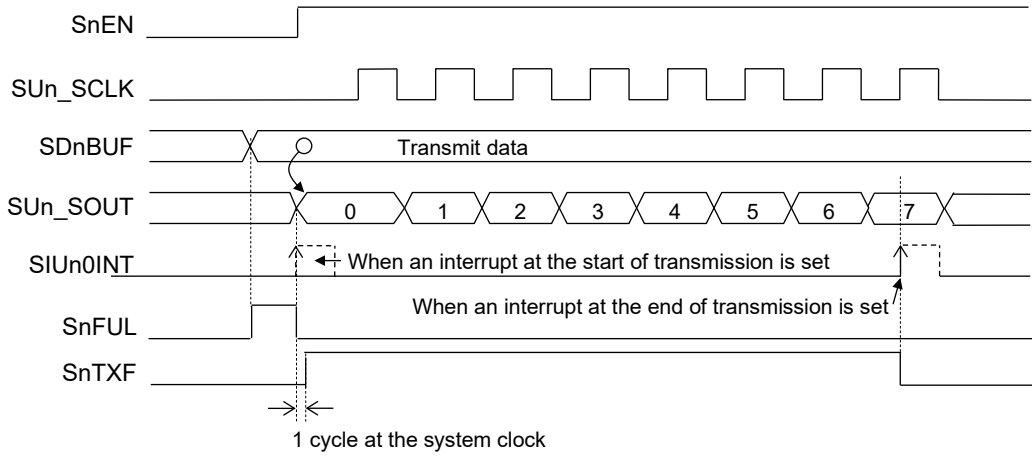


Figure 11-4 Transmit Operation Waveforms of Synchronous Serial Port for Clock Type 1 (Positive Logic)
(8-Bit Length, LSB First)

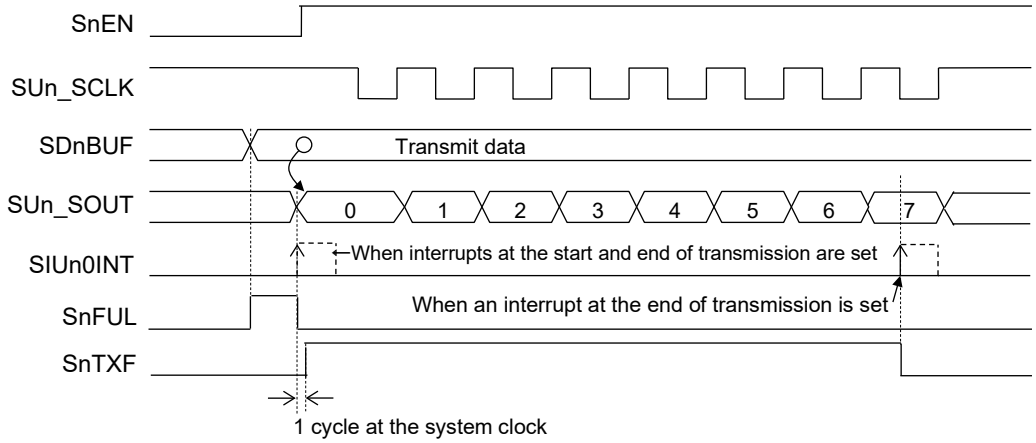


Figure 11-5 Transmit Operation Waveforms of Synchronous Serial Port for Clock Type 1 (Negative Logic)
(8-Bit Length, LSB First)

11.3.1.2 Receive Operation Timing

Figure 11-6 shows the reception 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).
In the clock type 1 slave mode, a timing that SnRXF becomes to "1" is different from other mode/clock type. See section 11.3.1.6 “Timing in Clock Type 1 Slave Mode” for details of its timing.
SUn_SCLK is output by writing dummy transmission data to SDnBUF. Therefore, writing the dummy is required for reception.
If it is preferable to set the frame interval at the time of reception, use the transmission/reception mode.

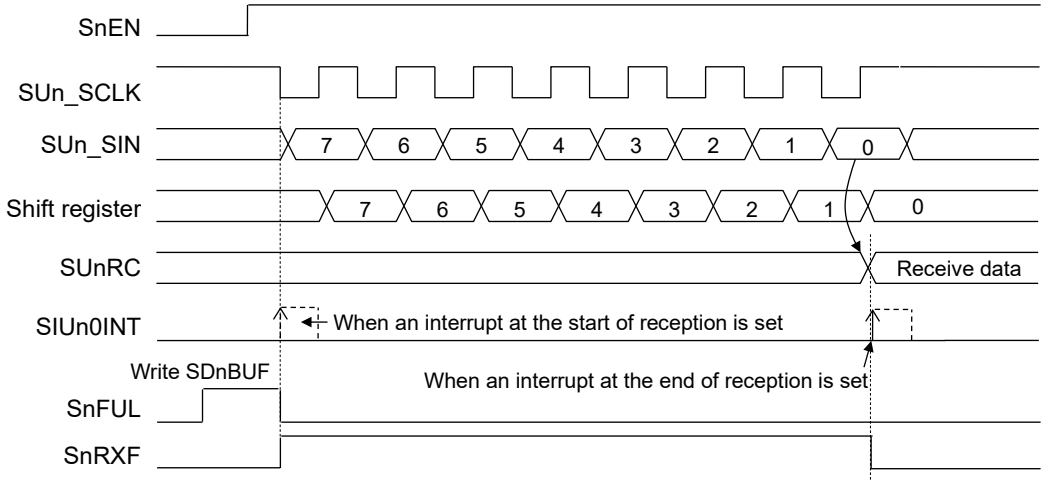


Figure 11-6 Receive Operation Waveforms of Synchronous Serial Port for Clock Type 0 (Positive Logic)
(8-Bit Length, MSB First)

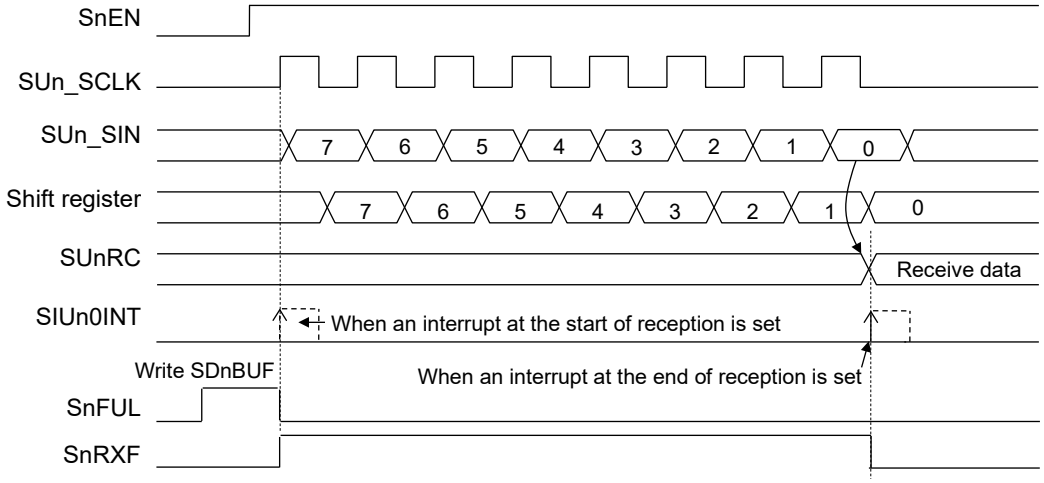


Figure 11-7 Receive Operation Waveforms of Synchronous Serial Port for Clock Type 0 (Negative Logic)
(8-Bit Length, MSB First)

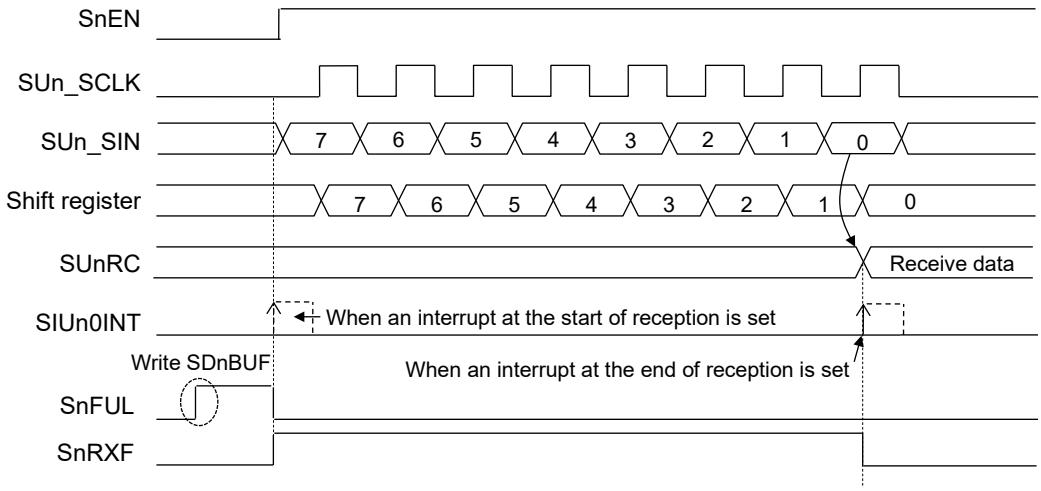


Figure 11-8 Receive Operation Waveforms of Synchronous Serial Port for Clock Type 1 (Positive Logic)
(8-Bit Length, MSB First)

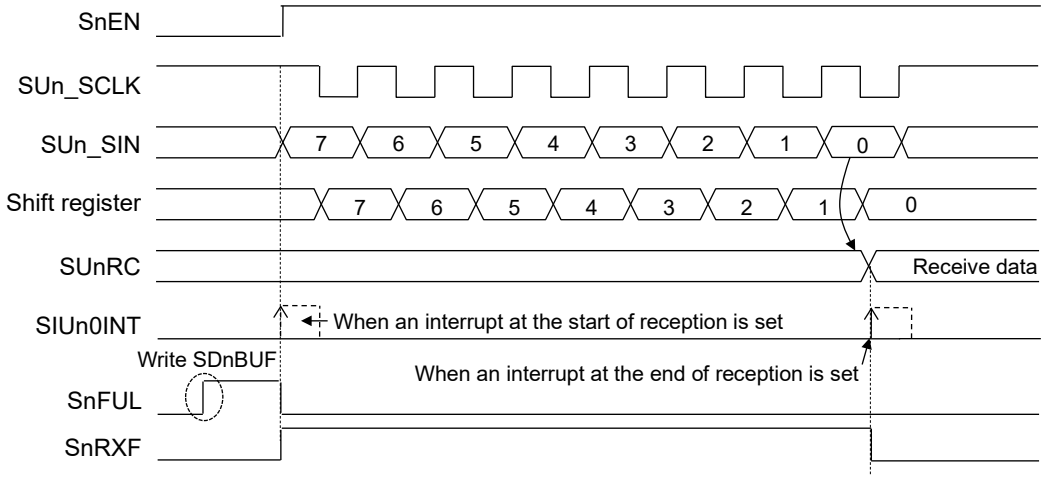


Figure 11-9 Receive Operation Waveforms of Synchronous Serial Port for Clock Type 1 (Negative Logic)
(8-Bit Length, MSB First)

11.3.1.3 Transmit/Receive Operation Timing

Figure 11-10 shows the transmission/reception operation waveform (with 16-bit length, MSB first, clock type 0) of the synchronous serial port.
After the completion of transmission or reception, the value of the SUn_SOUT will be "1" with 16-bit length, will be the bit 8 of SDnBUF register with 8-bit length, LSB first, or will be "0" with 8-bit length, MSB first.

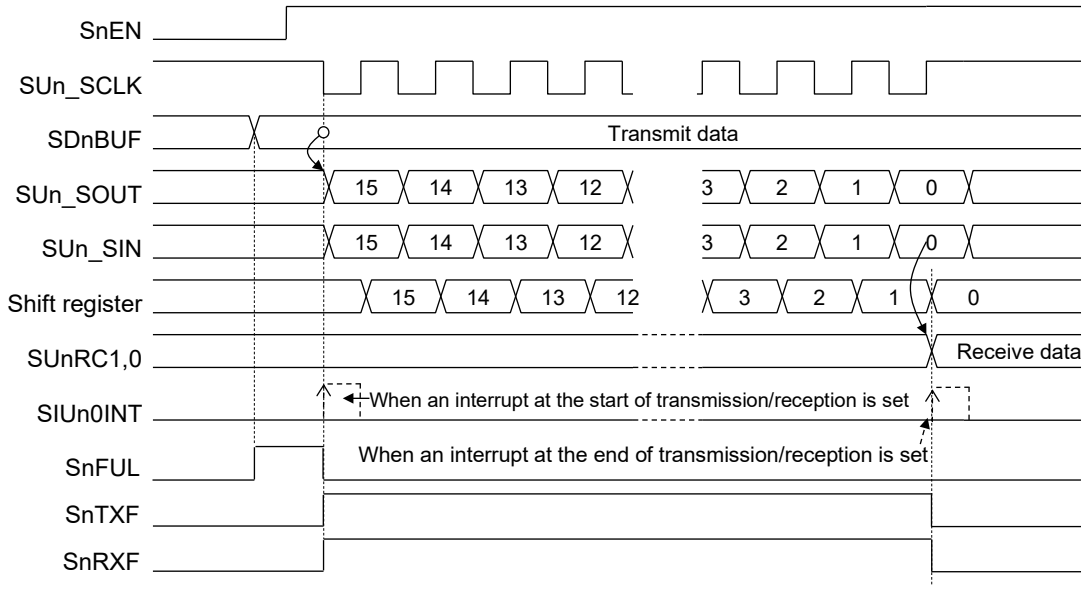


Figure 11-10 Receive Operation Waveforms of Synchronous Serial Port for Clock Type 0 (Positive Logic)
(16-Bit Length, MSB First)

11.3.1.4 Interrupt Generation Timing

Table 11-5 shows the interrupt generation timing in the synchronous serial port mode.

Table 11-5 Interrupt Generation Timing in Synchronous Serial Port Mode (1/2)

■ Master mode

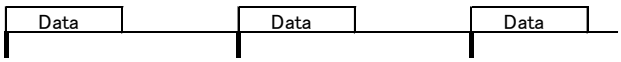
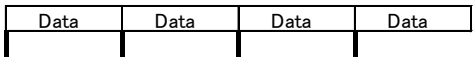
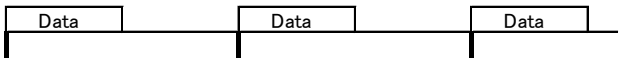
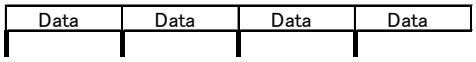
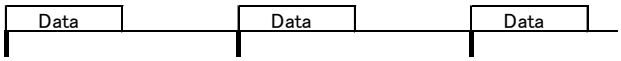
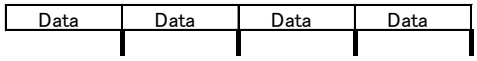
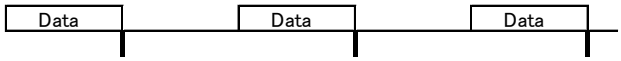
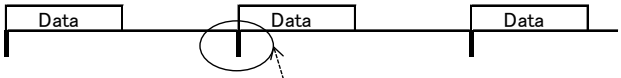
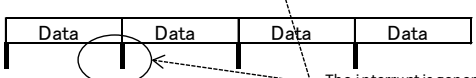
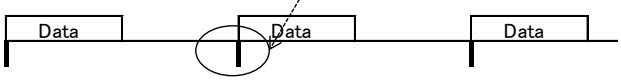
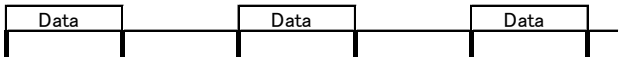
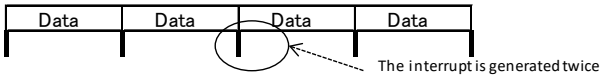
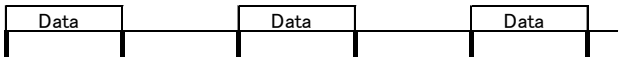
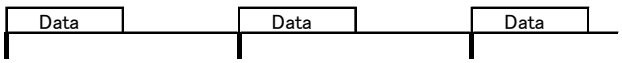
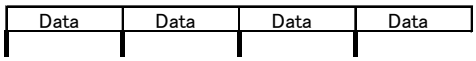
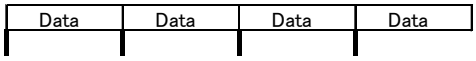
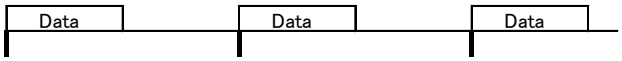
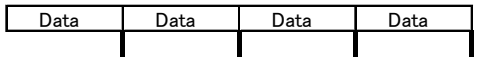
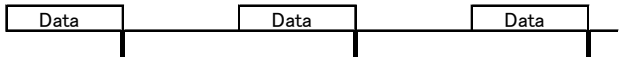
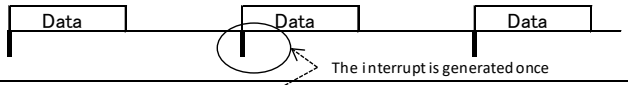
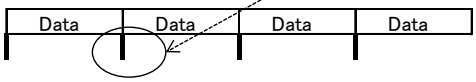
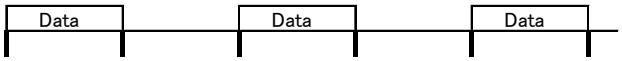
Operation mode	Setting of interrupt timing	Setting of transmission interval time	SnFUL	Timing of the interrupt generation ("|" indicates the interrupt)
Transmission	At the start of transmission	The interval time is set	The buffer has data	
		The interval time is not set	The buffer has data	
		Both case	The buffer has no data	
Reception	At the start of reception	–	The buffer has data	
			The buffer has no data	
	At the end of reception	–	The buffer has data	
			The buffer has no data	
Transmission / Reception	at the start of transmission and reception	The interval time is set	The buffer has data	
		The interval time is not set	The buffer has data	
		Both case	The buffer has no data	
	At the start of transmission and at the end of reception	The interval time is set	The buffer has data	
		The interval time is not set	The buffer has data	
		Both case	The buffer has no data	

Table 11-5 Interrupt Generation Timing in Synchronous Serial Port Mode (2/2)

■ Slave mode

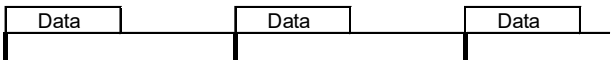
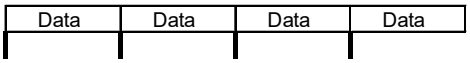
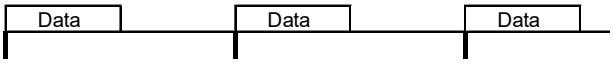
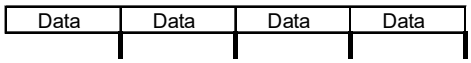
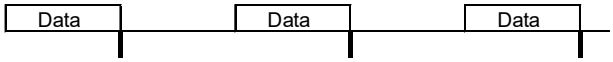
Operation mode	Setting of interrupt timing	Setting of transmission interval time	SnFUL	Timing of the interrupt generation ("|" indicates the interrupt)
Transmission	Th the start of transmission	–	The buffer has data	
		–	The buffer has no data	
Reception	At the start of reception	–	The buffer has no data	
			The buffer has data	
	At the end of reception	–	The buffer has no data	
			The buffer has data	
Transmission / Reception	At the start of transmission and reception	–	The buffer has data	
		–	The buffer has no data	
	At the start of transmission and at the end of reception	–	The buffer has data	
		–	The buffer has no data	

11.3.1.5 DMA Request Timing

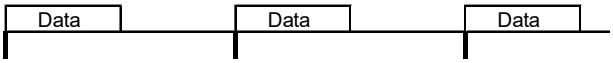
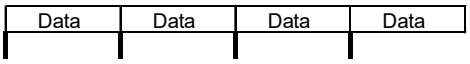
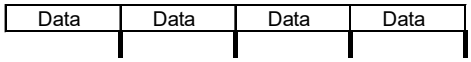
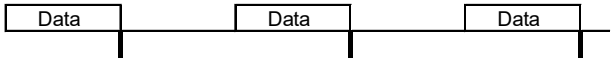
Table 11-6 shows the DMA request timing in the synchronous serial port mode.

Table 11-6 DMA Request Timing in Synchronous Serial Port Mode

■ Master mode

Operation mode	Setting of DMA request	Setting of transmission interval time	SnFUL	Timing of the interrupt generation ("|" indicates the interrupt)
Transmission	At the start of transmission	The interval time is set	The buffer has data	
		The interval time is not set	The buffer has data	
		Both case	The buffer has no data	
Reception	At the end of reception	-	The buffer has data	
		-	The buffer has no data	

■ Slave mode

Operation mode	Setting of DMA request	Setting of transmission interval time	SnFUL	Timing of the interrupt generation ("|" indicates the interrupt)
Transmission	At the start of transmission	-	The buffer has data	
		-	The buffer has no data	
Reception	At the end of reception	-	The buffer has no data	
		-	The buffer has data	

11.3.1.6 Timing in Clock Type 1 Slave Mode

The operation is fundamentally performed in the same timing in both the master and slave modes. In the clock type 1 slave mode, in order for it to be able to perform any time when the clock is supplied from the master, preparation for data transfer that follows is started as soon as the preceding data transfer is completed. Therefore, the underrun error status is updated, when the SnEN bit is set to "1" or each transmission completed.

Figure 11-11 shows the transmission/reception operation waveform (with 8-bit length, clock type 1) of the synchronous serial port.

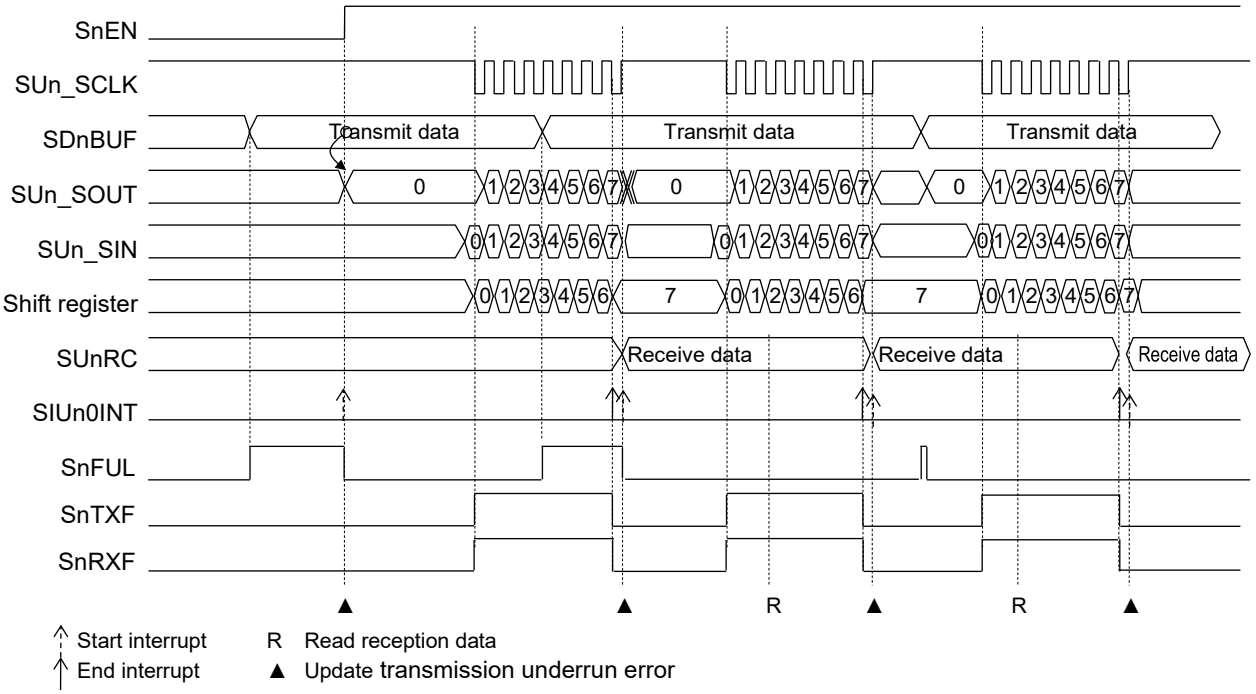


Figure 11-11 Transmit/Receive Operation Waveforms of Synchronous Serial Port in Clock Type 1 (Positive Logic) Slave Mode

[Note]

- **When the timing of transmission interrupt is chosen "at the end of data transmission" (SUnTIMD=0), it is possible to write data to the transfer buffer before the transfer is actually started (before the external clock is supplied). In that case, the data written just before the start of the transfer is transferred. To ensure that data is successfully transmitted, it is recommended that data is written when SnEN is "0" or while the transfer of previous data is in progress (SnTXF=1).**

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 choosable: 5 to 8 bits for the data bit, even/odd/ fixed to "1", or fixed to "0" for the parity bit, 1 stop bit or 2 stop bit for the stop bit, LSB first or MSB first for the transfer direction, and positive logic or negative logic for the logic of the serial input/output.
All of these are set in the UARTn0 mode register (UAn0MOD) or UARTn1 mode register (UAn1MOD).
Figure 11-12 and Figure 11-13 show the positive logic input/output format and negative logic input/output format, respectively.

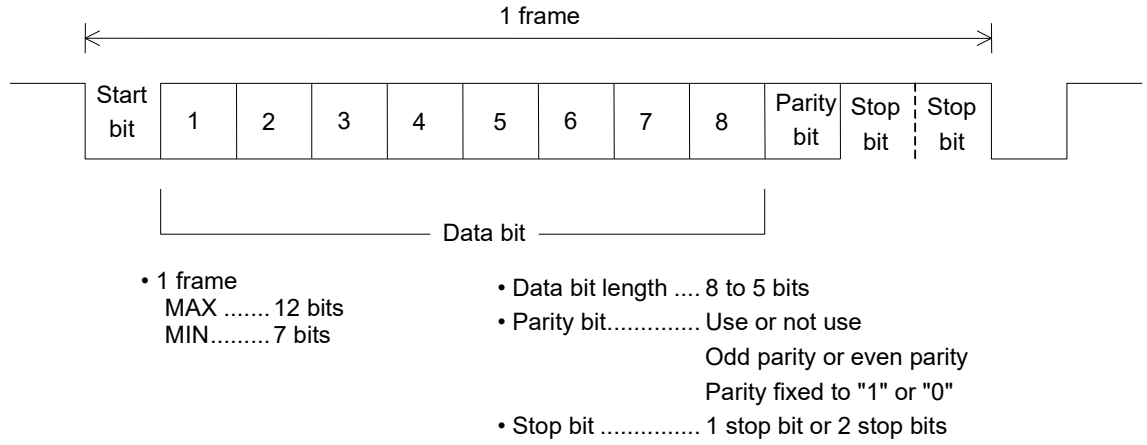


Figure 11-12 Format of Positive Logic Input/Output (LSB First)

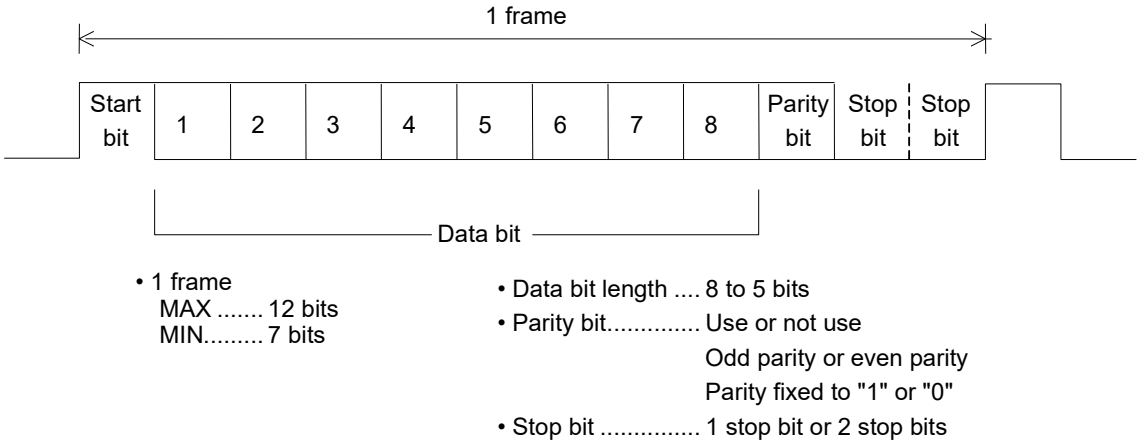


Figure 11-13 Format of Negative Logic Input/Output (LSB First)

11.3.2.2 Baud Rate

The baud rate generator generates a baud rate using the base clock chosen in the UARTn0 mode register (UAn0MOD) and the UARTn1 mode register (UAn1MOD). The setting values for the UARTn0 baud rate register (UAn0BRT), the UARTn1 baud rate register (UAn1BRT), the UARTn0 baud rate adjustment register (UAn0BRC) and the UARTn1 baud rate adjustment register (UAn1BRC) can be calculated by the following formulae.

UAn0BRT, UAn1BRT = ROUNDDOWN (Base clock frequency (Hz) / Baud rate (bps)) - 1
UAn0BRC, UAn1BRC = ROUND ((Base clock frequency (Hz) % Baud rate (bps)) x 8 / Baud rate (bps))

ROUNDDOWN: Rounded down, ROUND: Rounded to the nearest whole number, %:Surplus
Setting range of UAn0BRC, UAn1BRC is 0 to 7. If the calculated value of UAn0BRC, UAn1BRC is 8, add 1 to UAn0BRT, UAn1BRT and set 0 to UAn1BRC, UAn1BRC.

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.

Example: Base clock frequency: Approx.24 MHz (23.986176 MHz), Baud rate: 115,200 bps
UAn0BRT, UAn1BRT = 23.986176 MHz / 115,200 bps - 1
= 208.21333… - 1 = 207 (rounding down to the nearest integer)
= 0x00CF

UAn0BRC, UAn1BRC= (23.986176 MHz % 115,200 bps) x 8 / Baud rate (bps)
= (24576 x 8) / 115,200
= 196608 / 115,200
= 1.70666… = 2 (rounding to the nearest integer)
= 0x02

The actual baud rate calculated from the setting value for the baud rate can be expressed by the following formula:

Actual baud rate (bps) = [Base clock frequency] / {(UAn0BRT + 1) + (UAn0BRC / 8)}
Actual baud rate (bps) = [Base clock frequency] / {(UAn1BRT + 1) + (UAn1BRC / 8)}

Example: Base clock frequency: Approx.24 MHz (23.986176 MHz), Baud rate ideal value: 1200 bps
Actual baud rate (bps) = 23.986176 MHz / {(0x4E13 + 1)} + (0x04 / 8)}
≈ 1199.99

Table 11-7 lists the count values for typical baud rates.

Table 11-7 Count Values for Typical Baud Rates (1/2)

Base clock	Baud rate	UAn0BRT UAn1BRT	UAn0BRC UAn1BRC	Actual baud rate
Approx. 24 MHz (approx. 23.986176 MHz)	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

Table 11-7 Count Values for Typical Baud Rates (2/2)

Base clock	Baud rate	UAn0BRT UAn1BRT	UAn0BRC UAn1BRC	Actual baud rate
Approx. 16 MHz (approx. 15.990784 MHz)	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. 32.768 kHz	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 Direction of Transmit/Receive Data

Figure 11-14 shows the relationship between the transmission/reception buffer and transmission/reception data.

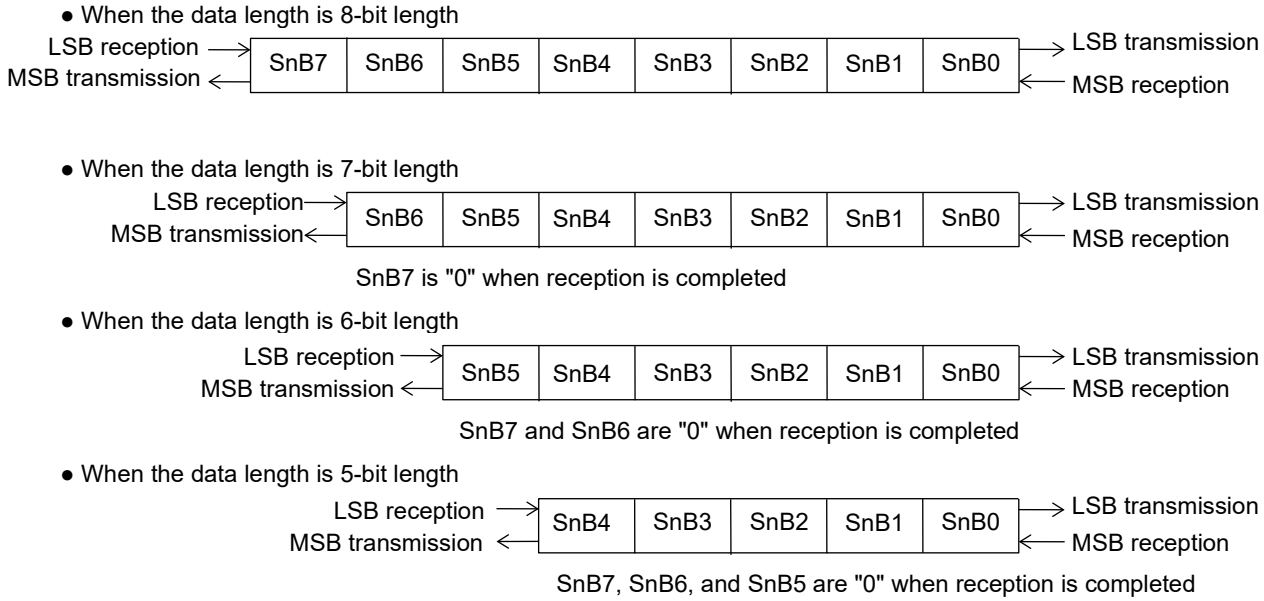


Figure 11-14 Relationship between Transmission/Reception Buffer and Transmission/Reception Data

[Note]

- **When the SUn_TXDn pin is set to the shared function in the reception mode, "H" level is output from the SUn_TXDn pin.**

11.3.2.4 Transmit Operation

The following shows the transmission procedure in the UART full-duplex communication mode. Figure 11-15 shows the operation timing for transmission.

- To prepare the communication (settings common to transmission/reception for full-duplex communication):
  - Choose the full-duplex communication mode in the serial communication unit n mode register (SUnMOD).
  - If using the transmission interval function, set the serial communication unit n transmission interval setting register (SUnDLYL).
  - Choose the communication mode with the UARTn0 mode register (UAn0MOD).
  - Set the baud rate with the UARTn0 baud rate register (UAn0BRT) and the UARTn0 baud rate adjustment register (UAn0BRC).
  - Set the shared function for the general-purpose port to use for UART communication, then choose the pin mode.
  - Clear the request bit for 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 SDnBUFL to prevent a false detection of overrun errors. The read data can be discarded.
  - Write "0xFF" to the UARTn0 status register (UAn0STAT) to clear each flags.
- To start transmission:
  - Enable transmission/reception by setting Un0EN bit of the serial communication unit n control register (SUnCON) to "1". (1)
  - Writing transmission data to SDnBUFH (2) causes Un0FUL of UAn0STAT to be set to "1" and the baud rate generator to generate the internal transfer clock. Then baud rate generator starts generating the internal transfer clock after one cycle of the system clock.
  - Once the the transmission is started, the start bit is output to the SUn_TXD1 pin at the falling edge of the internal transfer clock. Un0FUL simultaneously becomes "0". (3)
  - Then, one cycle after the internal transfer clock, the start bit is output to the SU_TXD1 pin, and the Un0FUL bit becomes "0" after on cycle of the base clock.
  - Then, the serial communication unit n1 interrupt (SIUn1INT) is generated if generation of interrupts at the start and end of the data transmission is chosen in the serial communication unit n mode register (SUnMOD).
- Continuous transmission
  - Write the next transmission data into the SDnBUFH after checking the Un0FUL bit gets "0" by the previous transmission and the serial communication unit n1 interrupt (SIUn1INT) has occurred (4). This writing causes the Un0FUL bit to be set to "1". Writing subsequently transmission data within the transmission/reception buffer write enable period (5) makes it possible to transmission data continuously.
  - Once the transmission of the stop bit of the first transmitted data is completed (6), the transmission is continued if the Un0FUL bit is set to "1" and the serial communication unit n1 interrupt (SIUn1INT) is simultaneously generated.
- Transmission end
  - When the transmission is completed without writing the subsequently transmitted data (7), the transmission is stopped and the serial communication unit n1 interrupt (SIUn1INT) is simultaneously generated.
  - If continuing the transmission, write the transmission data to SDnBUFH. To stop all the UART transmission/reception, reset the Un0EN bit of the SUnCONL to "0".

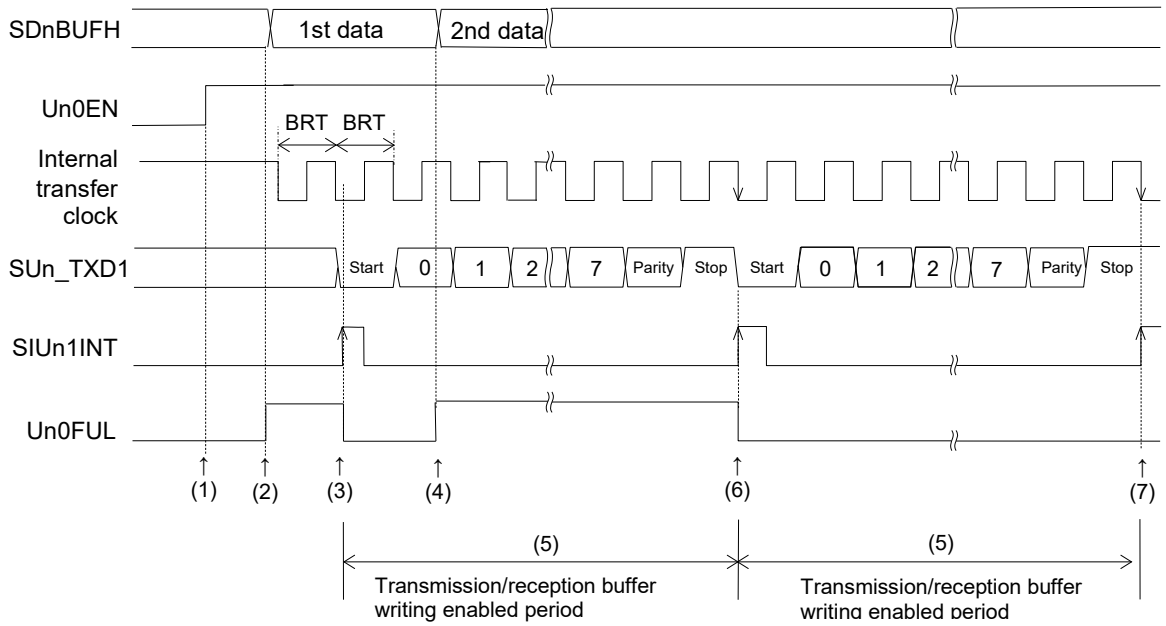


Figure 11-15 Operation Timing in Transmission

11.3.2.5 Receive Operation

The following shows the reception procedure in the UART full-duplex communication mode. Figure 11-16 shows the operation timing for reception.

- To prepare the communication (settings common to transmission/reception for full-duplex communication):
  - Choose the full-duplex communication mode in the serial communication unit n mode register (SUnMOD).
  - If using the transmission interval function, set the serial communication unit n transmission interval setting register (SUnDLYL).
  - Choose the communication mode with the UARTn0 mode register (UAn0MOD).
  - Set the baud rate with the UARTn0 baud rate register (UAn0BRT) and the UARTn0 baud rate adjustment register (UAn0BRC).
  - Set the shared function for the general-purpose port to use for UART communication, then choose the pin mode.
  - Clear the request bit for 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 SDnBUFL to prevent a false detection of overrun errors. The read data can be discarded.
  - Write "0xFF" to the UARTn0 status register (UAn0STAT) to clear each flags.
- To start reception:
  - Enable transmission/reception by setting Un0EN bit of the serial communication unit n control register (SUnCON) to "1". (1)
  - Start detecting the start bit input to the SUn_RXD0 pin.
  - When "L" level of the SUn_RXD0 pin is detected (2), the baud rate generator starts generating the transfer clock. When "H" level is received in the middle of the start bit, it is recognized as an unintended operation and the detection of the start bit is resumed.
  - When "L" level is received in the middle of the start bit, the reception operation is started. Data input to SUn_RXD0 at the rising edge of the internal transfer clock is loaded to the shift register.
  - Once loading received data and parity bits is completed, the loaded data is transferred to the SDnBUFL.
  - In the middle of the stop bit (4), the serial communication unit n0 interrupt (SIUn0INT) is generated, and a framing error (stop bit error) and parity bit error are determined. If an error is detected, applicable bits (Un0FER and Un0PER) of the UARTn0 status register (UAn0STAT) are set to "1". Also, the operation simultaneously shifts to the detection of the subsequent start bit.
  - When continuously receiving data, if SDnBUFL is overwritten with the subsequently received data before the CPU reads the received data (SDnBUF), the overrun error (Un0OER) bit of the UAnSTAT register becomes "1".
- Reception end
  - If terminating the UART reception, set the Un0EN bit of the SUnCON register to "0". If resetting the Un0EN bit to "0" in the middle of the reception, the received data may be destroyed.

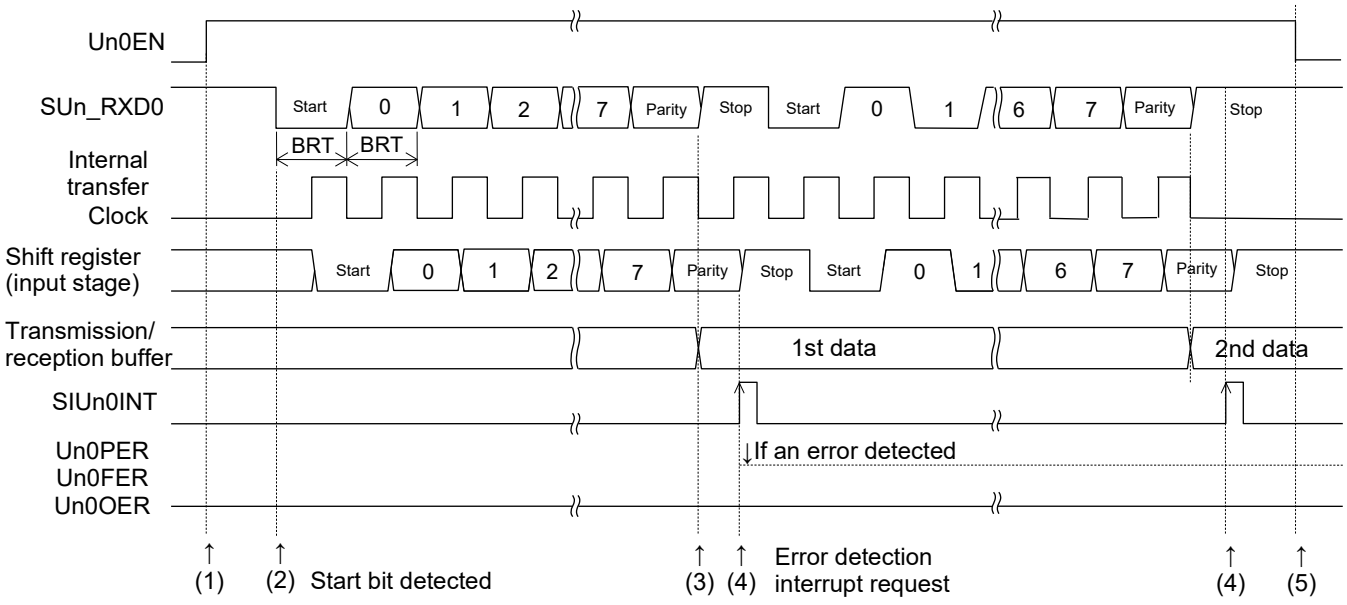
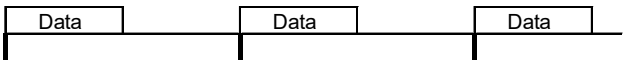
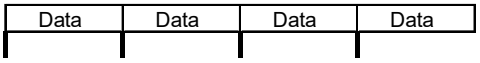
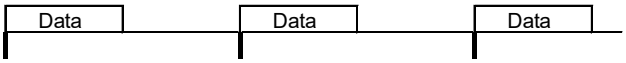


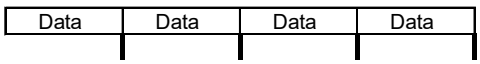
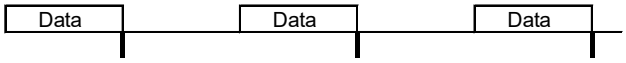
Figure 11-16 Operation Timing in Reception

11.3.2.6 Interrupt Generation Timing

Table 11-8 shows the interrupt generation timing in the UART mode.

Table 11-8 Interrupt Generation Timing in UART Mode

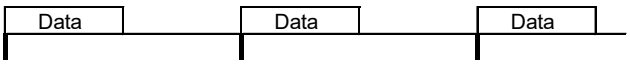
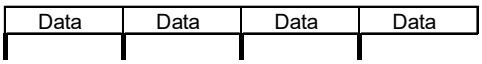
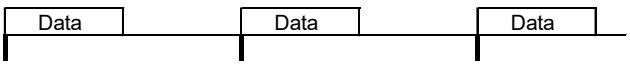
Operation mode	Setting of Interrupt timing	Setting of transmission interval time	SnFUL	Timing of the interrupt generation ("|" indicates the interrupt)
Transmission	At the start of transmission	The interval time is set	The buffer has data	
		The interval time is not set	The buffer has data	
		Both case	The buffer has no data	

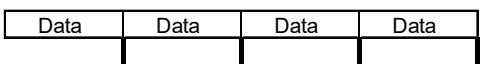
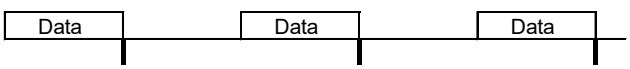
Operation mode	Setting of Interrupt timing	Setting of transmission interval time	SnFUL	Timing of the interrupt generation ("|" indicates the interrupt)
Reception	At the end of reception	-	The buffer has no data	
		-	The buffer has data	

11.3.2.7 DMA Request Timing

Table 11-9 shows the interrupt generation timing in the UART mode.

Table 11-9 DMA Request Timing in UART Mode

Operation mode	Setting of DMA request	Setting of transmission interval time	SnFUL	Timing of the interrupt generation ("|" indicates the interrupt)
Transmission	At the start of transmission	The interval time is set	The buffer has data	
		The interval time is not set	The buffer has data	
		Both case	The buffer no has data	

Operation mode	Setting of DMA request	Setting of transmission interval time	SnFUL	Timing of the interrupt generation ("|" indicates the interrupt)
Reception	At the end of reception	-	The buffer no has data	
		-	The buffer has data	

11.3.2.8 Detection of Start Bit

The start bit is sampled with the baud rate generator base clock. Therefore, the start bit detection may be delayed for one cycle of the baud rate generator clock at the maximum.
Figure 11-17 shows the start bit detection timing.

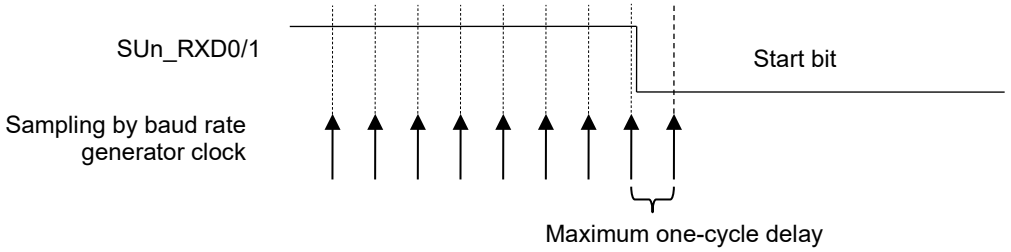
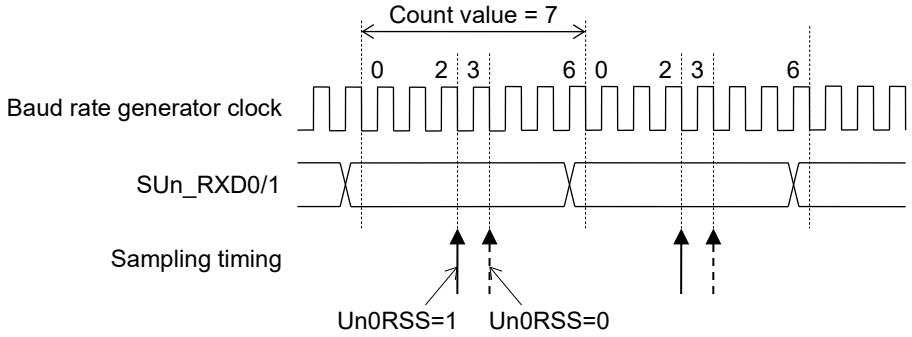


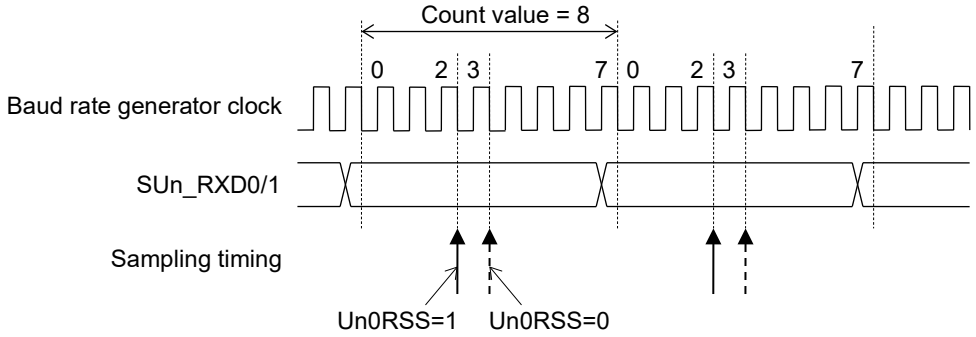
Figure 11-17 Start Bit Detection Timing (with Positive Logic)

11.3.2.9 Sampling Timing

When the start bit is detected, the received data that was input to SUn_RXD0/1 is sampled almost at the center of the baud rate, and then loaded to the shift register.
This sampling timing the shift register uses to load data can be adjusted for one clock of the baud rate generator clock in the Un0RSS bit of the UARTn0 mode register (UAn0MOD) or the Un1RSS bit of the UARTn1 mode register (UAn1MOD).
Figure 11-18 shows the relationship between the Un0RSS bit of the UARTn0 mode register and the sampling timing.



(1) When the baud rate generator count value is "7" (odd)



(2) When the baud rate generator count value is "8" (even)

Figure 11-18 Relationship between Un0RSS Bit and Sampling Timing

11.3.2.10 Receive Margin

If there is an error between the sender baud rate and the receiver baud rate generated by the baud rate generator, the error accumulates until the last stop bit loading in one frame, decreasing the reception margin.
Figure 11-19 shows the baud rate errors and reception margin waveforms.

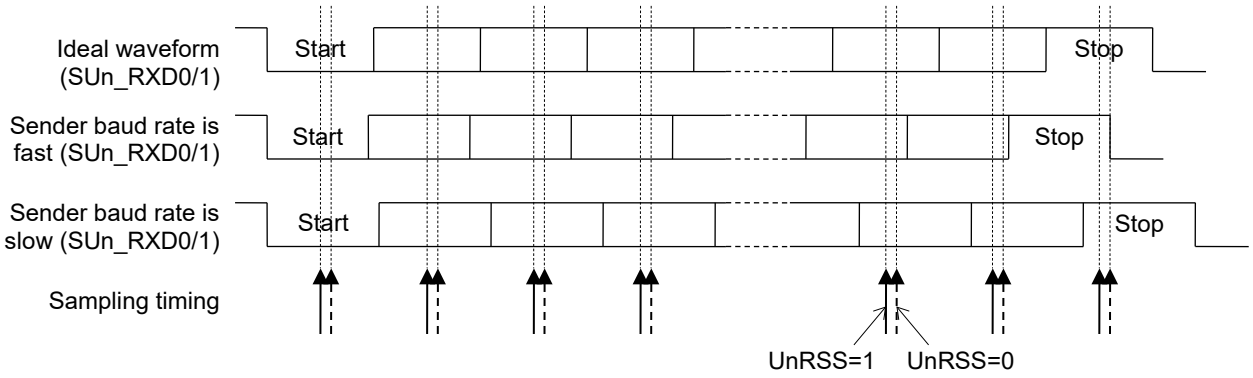


Figure 11-19 Baud Rate Errors and Reception Margin

[Note]

- **When designing the system, consider the difference of the baud rate between the transmission side and reception side and delay of the start bit detection and adjust the baud rate in the UAn0BRT, UAn1BRT, UAn0BRC, and UAn1BRC registers.**

11.3.2.11 Note on usage of Half-duplex UART

When using the half-duplex UART and changing the transmission mode to reception mode, you must initialize and reconfigure the channel as the reception mode after reset the channel of serial communication unit by Block Reset Control Register 2 (BRECON2). Notice that the reset by the BRECON2 register also resets other channel of the half-duplex UART.
Figure 11-20 shows the flow chart.

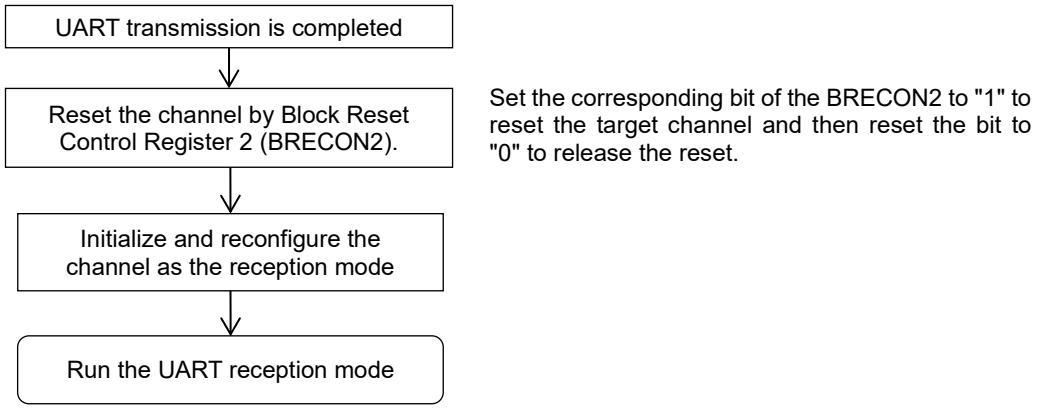


Figure 11-20 Flow chart for switching the transmission mode to the reception mode in Half-duplex UART mode

This process is not necessary when using the half-duplex UART and changing the reception mode to transmission mode.

Chapter 12 I^2C Bus Unit

12. I²C Bus Unit

12.1 General Description

ML62Q1000 series has one channel of I²C bus unit that supports both master and slave function.
Either of master or slave can be chosen to use and both functions of master and slave are unworkable at the same time.

Table 12-1 shows the number of I²C Bus Unit channels.

Table12-1 Number of I²C Bus Unit channels

Channel no.	ML62Q1300 group				ML62Q1500 / ML62Q1800 / ML62Q1700 group				
	16pin product	20pin product	24pin product	32pin product	48pin Product	52pin product	64pin product	80pin product	100pin product
0	●	●	●	●	●	●	●	●	●

●: Available

12.1.1 Features

Table 12-2 shows the features of I²C bus unit.

Table 12-2 Features of I²C bus unit

Function	Operation mode	Features
I²C bus unit	Master function	・ Communication speed: Standard mode (100 kbps), fast mode (400 kbps), and original standard 1 Mbps mode (1Mbps) ・ Support clock stretch function for the Slave ・ 7-bit address format (only the master function supports 10-bit address format) ・ Self-test function by reading transmitted data onto the I²C bus (Safety function)
	Slave function	・ Communication speed: Standard mode (100 kbps), fast mode (400 kbps), and original standard 1 Mbps mode (1Mbps) ・ Clock stretch function ・ 7-bit address format ・ Wake-up from STOP mode by matching slave address

12.1.2 Configuration

Figure 12-1 shows the configuration diagram of the I²C bus unit circuit.

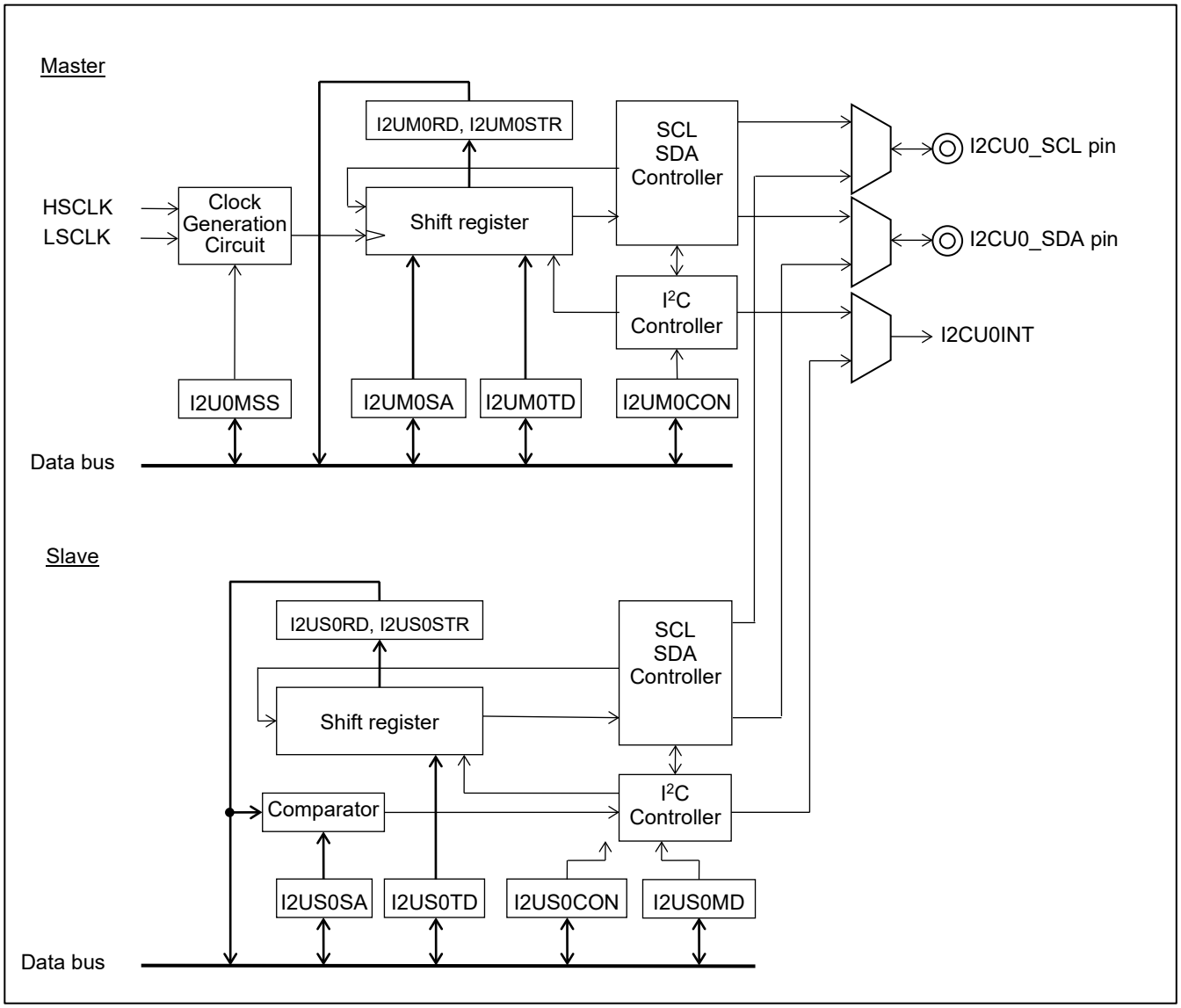


Figure 12-1 Configuration of I²C Bus Unit

I2CU0_SCL: Serial Clock
I2CU0_SDA: Serial Data
I2U0MSS: I²C bus unit n mode register
I2UM0RD: I²C bus 0 receive register (master)
I2UM0SA: I²C bus 0 slave address register (master)
I2UM0TD: I²C bus 0 transmit data register (master)
I2UM0CON: I²C bus 0 control register (master)
I2UM0STR: I²C bus 0 status register (master)
I2US0RD: I²C bus 0 receive register (slave)
I2US0SA: I²C bus 0 slave address register (slave)
I2US0TD: I²C bus 0 transmit data register (slave)
I2US0CON: I²C bus 0 control register (slave)
I2US0MD: I²C bus 0 mode register (slave)
I2US0STR: I²C bus 0 status register (slave)

12.1.3 List of Pins

The I/O pins of the I²C bus unit are assigned to the shared function of the general ports.

Pin name	I/O	Description
I2CU0_SDA	I/O	I²C bus unit 0 data I/O pin
I2CU0_SCL	I/O	I²C bus unit 0 clock I/O pin

12.1.4 Pin Setting

I2CU0_SDA pin and I2CU0_SCL pin are assigned to multiple general ports.
Be sure to use the ports in following combinations.

Pin name	Combination 1	Combination 2	Combination 3	Combination 4 *1	Combination 5 *1
I2CU0_SDA	P03	P15	P26	P03	P46
I2CU0_SCL	P04	P16	P27	P02	P47

*1 : Available for ML62Q1700 group only

In addition to the mode setting of the shared function, choose "Enable Input, Enable Output, Nch open drain output and without pull-up" by setting following data to the port n mode register m (PnMODm).

Table 12-3 I²C bus unit general port combinations

Port name	PnMODm	Combination	Setting data	ML62Q1300 group				ML62Q1500/ ML62Q1800 group					ML62Q1700 group				
				16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product	48pin product	52pin product	64pin product	80pin product	100pin product
P03	P0MOD3	1	0x3B	●	●	●	●	●	●	●	●	●	●	●	●	●	●
P04	P0MOD4	1	0x3B	●	●	●	●	●	●	●	●	●	●	●	●	●	●
P15	P1MOD5	2	0x3B	-	-	-	●	●	●	●	●	●	●	●	●	●	●
P16	P1MOD6	2	0x3B	-	-	-	●	●	●	●	●	●	●	●	●	●	●
P26	P2MOD6	3	0x3B	●	●	●	●	●	●	●	●	●	●	●	●	●	●
P27	P2MOD7	3	0x3B	●	●	●	●	●	●	●	●	●	●	●	●	●	●
P03	P0MOD3	4	0x3B	-	-	-	-	-	-	-	-	-	●	●	●	●	●
P02	P0MOD2	4	0x3B	-	-	-	-	-	-	-	-	-	●	●	●	●	●
P46	P4MOD6	5	0x3B	-	-	-	-	-	-	-	-	-	-	-	●	●	●
P47	P4MOD7	5	0x3B	-	-	-	-	-	-	-	-	-	-	-	●	●	●

n : General port number (0 to 3) m : Bit number (0 to 7) ● : Available - : Unavailable

[Note]

- **Use external pull-up resistors for SDA pin and SCL pin referring to the I²C bus specification. The internal pull-up resistors unsatisfy the I²C bus specification. See the data sheet for each product for the value of internal pull-up resistors.**
- **If powering off this LSI in the slave mode, it disables communications of other devices on the I²C bus. Keep this LSI powered on when it works as a slave mode until the master device is powered off.**
- **When using the master function, do not connect multiple master devices on the I²C bus.**

12.2 Description of Registers

12.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF6C0	I²C bus unit 0 mode register	I2U0MSS	-	R/W	8	0x00
0xF6C1	Reserved	-	-	-	-	-
0xF6C2	I²C bus 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	-	-	-	-	-
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	I2UM0STR	R/W	8/16	0x00
0xF6CD		I2UM0ISR		R/W	8	0x00
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	I2US0STR	R/W	8/16	0x00
0xF6D9		I2US0ISR		R/W	8	0x00

12.2.2 I2C Bus Unit 0 Mode Register (I2U0MSS)

I2U0MD is a special function register (SFR) used to choose the Master mode or Slave mode of the I2C bus unit.

Address: 0xF6C0 (I2U0MSS)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0MSS							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	I2U0MD
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	I2U0MD	This bit is used to choose the Master mode or Slave mode of the I2C bus unit. 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.**
- **When using the master function, do not connect multiple master devices on the I2C bus.**
- **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 salve function, switch the system clock to the high-speed clock if releasing the communication wait status.**
- **When using the salve function with multi-slaves connected to the I2C bus, conform to the following conditions while enabling the I2C bus function (I2U0MD=1 and I2US0EN=1) regardless communicating or not.**
  - **Specify SYSTEMCLK as four time or higher than the I2C bus communication speed.**
    SYSTEMCLK needs to be 500kHz or higher when the I2C bus communication speed is 100kbps.
    SYSTEMCLK needs to be 2MHz or higher when the I2C bus communication speed is 400kbps.
    SYSTEMCLK needs to be 4MHz or higher when the I2C bus communication speed is 1Mbps.
  - **Do not use LSCLK as the SYSTEMCLK.**
  - **Do not enter HALT-H mode while enabling the I2C bus function.**

12.2.3 I²C Bus 0 Receive Register (Master) (I2UM0RD)

I2UM0RD is a read-only special function register (SFR) used to store the received data in the master mode.
The I2UM0RD is updated after completion of each reception.
This register is initialized, in addition to reset function, by writing "0" to I2UM0EN bit in I2UM0MOD register.

Address: 0xF6C2 (I2UM0RD)
Access: R
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2UM0RD							
Bit	-	-	-	-	-	-	-	-	I2UM0R7	I2UM0R6	I2UM0R5	I2UM0R4	I2UM0R3	I2UM0R2	I2UM0R1	I2UM0R0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	I2UM0R7 to I2UM0R0	These bits are used to store the received data in the master mode. The signal input to the I2CU0_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 I2CU0_SCL pin. Reading this register enables following confirmation. ・ Reading when receiving data: Can confirm the received data. ・ Reading slave address or Reading when transmitting data: Can confirm the transmission data is surely transmitted.

12.2.4 I²C Bus 0 Slave Address Register (Master) (I2UM0SA)

I2UM0SA is a special function register (SFR) to set the address and transmission/reception mode of the slave device in the master mode.
This register is initialized, in addition to reset function, by writing "0" to I2UM0EN bit in I2UM0MOD register.

Address: 0xF6C4 (I2UM0SA)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2UM0SA							
Bit	-	-	-	-	-	-	-	-	I2UM0A6	I2UM0A5	I2UM0A4	I2UM0A3	I2UM0A2	I2UM0A1	I2UM0A0	I2UM0RW
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	I2UM0A6 to I2UM0A0	These bits are used to set the address of the communication partner in the Master mode.
0	I2UM0RW	This bit is used to choose direction of the data communication in the master mode. 0: Data transmission mode (Initial value) 1: Data reception mode

12.2.5 I²C Bus 0 Transmit Data Register (Master) (I2UM0TD)

I2UM0TD is a special function register (SFR) used to set the transmission data in the master mode.
This register is initialized, in addition to reset function, by writing "0" to I2UM0EN bit in I2UM0MOD register.

Address: 0xF6C6 (I2UM0TD)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2UM0TD							
Bit	-	-	-	-	-	-	-	-	I2UM0T7	I2UM0T6	I2UM0T5	I2UM0T4	I2UM0T3	I2UM0T2	I2UM0T1	I2UM0T0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	I2UM0T7 to I2UM0T0	These bits are used to set the transmission data in the master mode.

12.2.6 I²C Bus 0 Control Register (Master) (I2UM0CON)

I2UM0CON is a special function register (SFR) used to control transmission and reception operations in the master mode. This register is initialized, in addition to reset function, by writing "0" to I2UM0EN bit in I2UM0MOD register.

Address: 0xF6C8 (I2UM0CON)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2UM0CON							
Bit	-	-	-	-	-	-	-	-	I2UM0ACT	-	-	-	-	I2UM0RS	I2UM0SP	I2UM0ST
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	W	W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	I2UM0ACT	This bit is used to set the acknowledgment data to be output at completion of reception in the master mode. 0: Acknowledgment data "0" (Initial value) 1: Acknowledgment data "1"
6 to 3	-	Reserved bits
2	I2UM0RS	This bit is a write-only and used to request a restart in the master mode. When "1" is written to this bit during data communication, the LSI shifts to the restart condition and the communication restarts from the slave address. "1" can be written to the I2UM0RS bit only while communication is active (I2UM0ST = "1"). The I2UM0RS bit always returns "0" for reading. 0: No restart request (Initial value) 1: Restart request
1	I2UM0SP	This bit is a write-only and used to request a stop condition in the master mode. When "1" is written to this bit, the LSI shifts to the stop condition and the communication stops. The I2UM0SP bit always returns "0" for reading. 0: No stop condition request (Initial value) 1: Stop condition request
0	I2UM0ST	This bit is used to control the communication operation in the master mode of the I2C bus unit. When "1" is written to this bit, the communication starts. When "1" is overwritten to this bit in a next data transmission/reception wait state after transmission/reception of acknowledgment, the data transmission/reception restarts. When "0" is written to this bit, the communication is stopped forcibly. When "1" is written to I2UM0SP bit, I2UM0ST bit is reset to "0". 0: Stops communication (Initial value) 1: Starts communication

[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 other bits of I2UM0CON register in the control register setting wait state.**

12.2.7 I²C Bus 0 Mode Register (Master) (I2UM0MOD)

I2UM0MOD is a special function register (SFR) used to set the operation mode in the master mode.

Address: 0xF6CA (I2UM0MDL/I2UM0MOD), 0xF6CB (I2UM0MDH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0200

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2UM0MOD															
Byte	I2UM0MDH								I2UM0MDL							
Bit	-	-	-	-	-	I2UM0CD2	I2UM0CD1	I2UM0CD0	-	-	I2UM0SYN	I2UM0DW1	I2UM0DW0	I2UM0MD1	I2UM0MD0	I2UM0EN
R/W	R	R	R	R	R	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	1	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 11	-	Reserved bits
10 to 8	I2UM0CD2 to I2UM0CD0	These bits are used to choose the operating frequency of I²C communication in the master mode. 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 (*) (*) When choosing LSCLK or 1/8LSCLK, "No communication speed reduction" is chosen regardless the setting of I2UM0DW1 to 0 bits. See Section 12.3.5 "Operation Waveforms" for details of the communication speed and clock counts.
7, 6	-	Reserved bits
5	I2UM0SYN	This bit is used to choose whether to or not to use the clock stretch (handshake) function in the master mode. Set this bit to "1" when using the clock stretch function. Monitor the I²C bus by setting this bit to "1", therefore the communication speed gets lower depending on the load of I²C bus. 0: Not use the clock stretch function (Initial value) 1: Use the clock stretch function
4, 3	I2UM0DW1, I2UM0DW0	These bits are used to set the communication speed reduction rate of the I²C bus unit in the master mode. Specify this bit not so that the communication speed exceeds 100 kbps/400kbps/1 Mbps. When LSCLK or 1/8 LSCLK is chosen by the I2UM0CD2 to 0 bits, "No communication speed reduction" is chosen regardless the setting of I2UM0DW1 to 0 bits. 0 0: No communication speed reduction (Initial value) 0 1: Approximately 9.1% communication speed reduction 1 0: Approximately 16.7% communication speed reduction 1 1: Approximately 23.1% communication speed reduction
2, 1	I2UM0MD1, I2UM0MD0	These bits are used to set the communication speed of the I²C bus unit in the master mode. 0 0: Standard mode (Initial value) (100 kbps*) 0 1: Fast mode (400 kbps*) 1 0: 1Mbps mode (1Mbps*) 1 1: 1Mbps mode (1Mbps*) * : When I2UM0CD2 to 0 bits are "000" and I2UM0SYN bit is "0".

Bit No.	Bit symbol name	Description
0	I2UM0EN	This bit is used to enable the master operation. When "1" is written to this bit, the I2UM0ST bit can be set and the I2UM0BB bit starts operation. When "0" is written to this bit, the I2C master stops operation and the I2UM0RD, I2UM0SA, I2UM0TD, I2UM0CON and I2UM0STR registers are initialized. In the case "0" is written to this bit during the communication, initialize the I²C bus unit and reconfigure it. 0: Stop the I²C master operation (Initial value) 1: Enable the I²C master operation

[Note]

- **When using the high-speed clock for the I²C operation, specify the following I²C operating clock frequency depending on the mode and the reference frequency of the PLL oscillation.**
  When HSCLK = 24MHz
  Standard mode: HSCLK to 1/4HSCLK
  Fast mode: HSCLK to 1/2HSCLK
  1Mbps mode: HSCLK to 1/2HSCLK

  When HSCLK = 16MHz
  Standard mode: HSCLK to 1/2HSCLK
  Fast mode: HSCLK
  1Mbps mode: HSCLK

12.2.8 I²C Bus 0 Status Register (Master) (I2UM0STR)

I2UM0STR is a special function register (SFR) to indicate the state of the I²C bus unit in the master mode.
This register is initialized, in addition to reset function, by writing "0" to I2UM0EN bit in I2UM0MOD register.

Address: 0xF6CC (I2UM0STA/I2UM0STR), 0xF6CD (I2UM0ISR)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2UM0STR															
Byte	I2UM0ISR								I2UM0STA							
Bit	-	-	-	-	-	I2UM0SPS	I2UM0DS	I2UM0AS	I2UM0BO	-	-	-	-	I2UM0ER	I2UM0ACR	I2UM0BB
R/W	R	R	R	R	R	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

Bit No.	Bit symbol name	Description
15 to 11	-	Reserved bits
10	I2UM0SPS	This bit is used to indicate the usage state of the I²C bus in the master mode. This bit is set to "1" when transmitting the stop condition has been completed on the I²C bus. To reset this bit, write "1" to this bit. 0: The stop condition has not been transmitted (Initial value) 1: The stop condition has been transmitted
9	I2UM0DS	This bit is used to indicate the usage state of the I²C bus in the master mode. This bit is set to "1" when transmitting data or receiving data has been completed on the I²C bus. To reset this bit, write "1" to this bit. 0: The transmission/reception has not been completed (Initial value) 1: The transmission/reception has been completed
8	I2UM0AS	This bit is used to indicate the usage state of the I²C bus in the master mode. This bit is set to "1" when transmitting the start condition and 7 bit slave address have been completed on the I²C bus. To reset this bit, write "1" to this bit. 0: The start condition and the slave address have not been transmitted (Initial value) 1: The start condition and the slave address have been transmitted
7	I2UM0BO	This bit is used to indicate the usage state of the I²C bus in the master mode. This bit is set to "1" when transmitting the start condition has been completed and is reset to "0" when the time (t_{BUF}) has passed after transmitting the stop condition or there happened a data communication error on the I2CU0_SDA pin. To reset this bit, write "1" to this bit. 0: The use right of the I²C bus has not been acquired (Initial value) 1: The use right of the I²C bus has been acquired
6 to 3	-	Reserved bits

Bit No.	Bit symbol name	Description
2	I2UM0ER	This bit is used to indicate a transmission error in the master mode. When a bit of transmission data and the value on the I2CU0_SDA pin do not coincide, "1" is set to this bit. To reset this bit, write "1" to this bit. 0: There was no transmission error (Initial value) 1: There was a transmission error When this bit is set to "1" and the clock stretch function is used (I2UM0SYN = "1"), the I2CU0_SDA pin output is disabled until the subsequent byte data communication terminates. Even if this bit is set to "1", the I2CU0_SDA pin output continues until the subsequent byte data communication terminates when the clock stretch function is not used (I2UM0SYN = "0").
1	I2UM0ACR	This bit is used to store the acknowledgment signal received in the master mode. Acknowledgment signals are received when the slave address is transmitted and the data transmission or reception is completed. To reset this bit, write "1" to this bit. 0: Received acknowledgment "0" (Initial value) 1: Received acknowledgment "1"
0	I2UM0BB	This bit is used to indicate the usage state of the I²C bus in the master mode. When the start condition is generated on the I²C bus this bit is set to "1", and when the stop condition is generated this bit is reset to "0". To reset this bit, write "1" to this bit. 0: The status of I²C bus is free (Initial value) 1: The status of I²C bus is busy

[Note]

- **Do not update each bit of the I2UM0STR 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.**
- **I2UM0BB bit and I2UM0BO bit are reset in one I²C operating clock after writing "1" to the bits.**

12.2.9 I²C Bus 0 Receive Register (Slave) (I2US0RD)

I2US0RD is a read-only special function register (SFR) used to store the received data in the slave mode.
The I2US0RD is updated after completion of each reception.

Address: 0xF6CE (I2US0RD)
Access: R
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2US0RD							
Bit	-	-	-	-	-	-	-	-	I2US0R7	I2US0R6	I2US0R5	I2US0R4	I2US0R3	I2US0R2	I2US0R1	I2US0R0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	I2US0R7 to I2US0R0	These bits are used to store the received data in the slave mode. The signal input to the I2CU0_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 I2CU0_SCL pin. Reading this register enables following confirmation. ・ Reading when receiving data: Can confirm the received data. ・ Reading when transmitting data: Can confirm the transmission data is surely transmitted.

12.2.10 I²C Bus 0 Slave Address Register (Slave) (I2US0SA)

I2US0SA is a special function register (SFR) used to set the slave address in the slave mode.

Address: 0xF6D0 (I2US0SA)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2US0SA							
Bit	-	-	-	-	-	-	-	-	I2US0A6	I2US0A5	I2US0A4	I2US0A3	I2US0A2	I2US0A1	I2US0A0	-
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	I2US0A6 to I2US0A0	These bits are used to set the slave address in the slave mode.
0	-	Reserved bit

12.2.11 I²C Bus 0 Transmit Data Register (Slave) (I2US0TD)

I2US0TD is a special function register (SFR) used to set the transmission data in the slave mode.

Address: 0xF6D2 (I2US0TD)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2US0TD							
Bit	-	-	-	-	-	-	-	-	I2US0T7	I2US0T6	I2US0T5	I2US0T4	I2US0T3	I2US0T2	I2US0T1	I2US0T0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	I2US0T7 to I2US0T0	These bits are used to set the transmission data in the slave mode.

12.2.12 I²C Bus 0 Control Register (Slave) (I2US0CON)

I2US0CON is a special function register (SFR) used to control transmission and reception operations in the slave mode.

Address: 0xF6D4 (I2US0CON)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2US0CON							
-Bit	-	-	-	-	-	-	-	-	I2US0ACT	-	I2US0WT	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R/W	R	W	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	I2US0ACT	This bit is used to set the acknowledgment data to be output at completion of reception in the slave mode. 0: Acknowledgment data "0" (Initial value) 1: Acknowledgment data "1"
6	-	Reserved bit
5	I2US0WT	This bit is used to release the communication wait state ("L" level output on the I2CU0_SCL pin) in the slave mode. Writing "1" to this bit during the communication wait state releases the state ("L" level output of the I2CU0_SCL pin is released). The I2US0WT bit is a write-only bit and always returns "0" for reading. 0: Not release the communication wait state (Initial value) 1: Release the communication wait state
4 to 0	-	Reserved bits

[Note]

- **Switch the system clock to the high-speed clock when releasing the communication wait status.**

12.2.13 I²C Bus 0 Mode Register (Slave) (I2US0MD)

I2US0MD is a special function register (SFR) used to set the operation mode in the slave mode.

Address: 0xF6D6 (I2US0MD)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2US0MD							
Bit	-	-	-	-	-	-	-	-	-	I2US0SIE	I2US0PIE	I2US0RIE	I2US0NAL	I2US0SPE	-	I2US0EN
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	-	Reserved bit
6	I2US0SIE	This bit is used to enable or disable the start condition interrupt in the slave mode. 0: Disables the start condition interrupt (Initial value) 1: Enables the start condition interrupt
5	I2US0PIE	This bit is used to enable or disable the stop condition interrupt in the slave mode. 0: Disables the stop condition interrupt (Initial value) 1: Enables the stop condition interrupt
4	I2US0RIE	This bit controls to enable or disable the slave address unmatched interrupt while communicating to the master, receiving re-start condition and an another slave is chosen. This function performs detecting the status of I2US0SAA bit. Do not clear the I2US0SAA bit by the software when enabling the interrupt. 0: Disable the slave address unmatched interrupt after the restart condition (Initial value) 1: Enable the slave address unmatched interrupt after the restart condition
3	I2US0NAL	This bit is used to enable or disable the communication wait function of the I²C bus unit (output "L" level on the I2CU0_SCL pin) when transmitting to the master and receiving the acknowledge data "1" from the master. Set this bit to "1" to use the communication wait function. 0: Disable the communication wait function when receiving the acknowledge data "1" from the master (Initial value) 1: Enable the communication wait function when receiving the acknowledge data "1" from the master
2	I2US0SPE	This bit is used to enable or disable the stop condition interrupt, which is detected by the stop condition that the master outputs while communicating with other slaves. This function performs detecting the status of I2US0SAA bit. Do not clear the I2US0SAA bit by the software when enabling the interrupt. This bit selects the mode when the stop condition interrupt is enabled (PIE=1). 0: The interrupt occur while the master is communicating with self-slave or other slaves (Initial value) 1: The interrupt occur while the master is communicating with only self-slave
1	-	Reserved bit
0	I2US0EN	This bit is used to enable the slave operation of the I²C bus unit. When "1" is written to this bit, the operation of the I²C bus unit 0 is enabled. When "0" is written to this bit, all the bits of the I²C bus status register (I2US0STR) are initialized to "0", and the operation of the I²C bus unit 0 is stopped. 0: Stop the I²C slave operation (Initial value) 1: Enable the I²C slave operation

[Note]

- **Stop the operation by resetting I2US0EN bit to "0" before entering STOP-D mode. Have the same handling if disable the wake-up from STOP mode by matching the slave address.**

12.2.14 I²C Bus 0 Status Register (Slave) (I2US0STR)

I2US0STR is a special function register (SFR) to indicate the state of the I²C bus unit in the slave mode.
This register is initialized, in addition to reset function, by writing "0" to I2US0EN bit in I2US0MD register.

Address: 0xF6D8 (I2US0STA/I2US0STR), 0xF6D9 (I2US0ISR)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2US0STR															
Byte	I2US0ISR								I2US0STA							
Bit	-	-	-	I2U0RAS	I2US0STS	I2U0SPS	I2US0DS	I2US0AS	-	-	-	I2US0TR	I2US0SAA	I2US0ER	I2US0ACR	I2US0BB
R/W	R	R	R	R/W	R/W	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

Bit No.	Bit symbol name	Description
15 to 13	-	Reserved bits
12	I2US0RAS	This bit is used to indicate status of the interrupt when enabling the start condition interrupt (I2US0RIE bit = 1) in the slave mode. To reset the I2US0RAS bit, write "1"to this bit. 0: Unmatched the slave address is detected after the start condition (Initial value) 1: Unmatched the slave address is detected after the start condition
11	I2US0STS	This bit is used to indicate status of transmission and reception in the slave mode. This bit is set to "1" when receiving the start condition. To reset the I2US0STS bit, write "1" to this bit. 0: The start condition has not been received (Initial value) 1: The start condition has been received
10	I2US0SPS	This bit is used to indicate status of transmission and receive in the slave mode. This bit is set to "1" when receiving the stop condition. To reset the I2US0SPS bit, write "1" to this bit. 0: The stop condition has not been received (Initial value) 1: The stop condition has been received
9	I2US0DS	This bit is used to indicate status of transmission and reception in the slave mode. This bit is set to "1" when transmitting or receiving data on the condition of that slave address is matched. To reset the I2US0DS bit, write "1" to this bit. 0: The data has not been transmitted or received (Initial value) 1: The data has been transmitted or received
8	I2US0AS	This bit is used to indicate status of transmission and reception in the slave mode. This bit is set to "1" when receiving the slave address data and it is matched. To reset the I2US0AS bit, write "1" to this bit. 0: The slave address has not been received or it is not matched (Initial value) 1: The slave address has been received and it is matched
7 to 5	-	Reserved bits

Bit No.	Bit symbol name	Description
4	I2US0TR	This bit is used to indicate the transmitting or receiving state in the slave mode. This bit is set to "1" when detecting the I2UM0RW bit of I2UM0SA register is "1" (data received mode). This bit is reset to "0" when detecting a stop condition, detecting the I2UM0RW bit is "0" (data transmission mode), or writing "1" to this bit. 0: Receiving state (Initial value) 1: Transmitting state
3	I2US0SAA	This bit is used to indicate that this device is specified as a slave address in the slave mode. This bit is set to "1" when the content of the slave address output by the master device coincides with the contents of I2US0SA register. This bit is reset to "0" when a stop condition is received or when "1" is written to this bit. 0: Not coincide with the slave address (Initial value) 1: Coincides with the slave address
2	I2US0ER	This bit is used to indicate a transmission error in the slave mode. When the value of the bit transmitted and the value of the I2CU0_SDA pin do not coincide, this bit is set to "1". When this bit is set to "1", the I2CU0_SDA pin output is disabled until the subsequent byte data communication terminates. To reset the I2US0ER bit, write "1" to this bit. 0: There was no transmission error (Initial value) 1: There was a transmission error
1	I2US0ACR	This bit is used to store an acknowledgment signal received in the slave mode. The acknowledgment signals are received each time the slave address is received and data transmission or reception is completed. To reset the I2US0ACR bit, write "1" to this bit. 0: Received the acknowledgment "0" (Initial value) 1: Received the acknowledgment "1"
0	I2US0BB	This bit is used to indicate the state of use of the I²C bus in the slave mode. When the start condition is generated on the I²C bus, this bit is set to "1" and when the stop condition is generated, this bit is reset to "0". To reset the I2US0BB bit, write "1" to this bit. 0: I²C bus-free state (Initial value) 1: I²C bus-busy state

[Note]

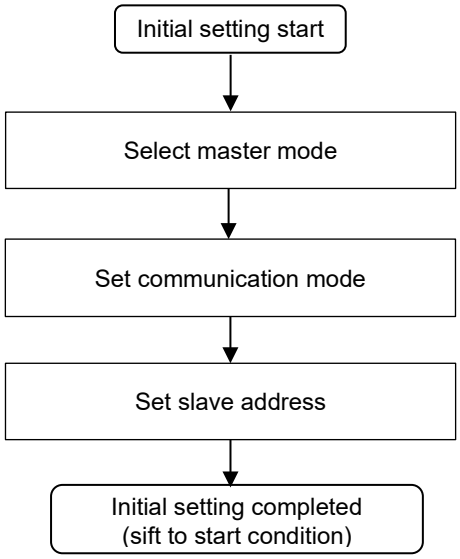
- **Do not update each bit of the I2UM0STA register by using the bit symbol. Update it by using a byte access or word access, not so that unintented bits are changed by the bit access instructions.**

12.3 Description of Operation

12.3.1 Master Operation

The following flow charts describe procedures of each operation in the master mode.

12.3.1.1 Initial Setting of Communication Operation

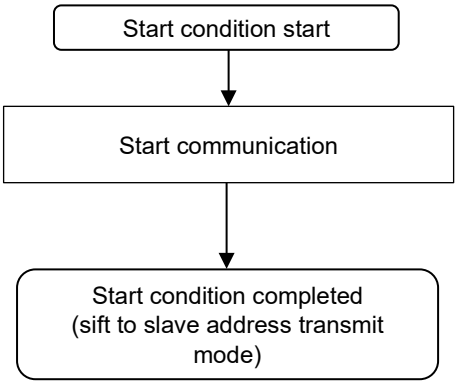


Set I2UMD=0 in the I2U0MSS register

Set I2UM0EN=1 in the I2UM0MOD register

Set I2UM0A6 to I2UM0A0 bits and I2UM0RW bit in the I2UM0SA register

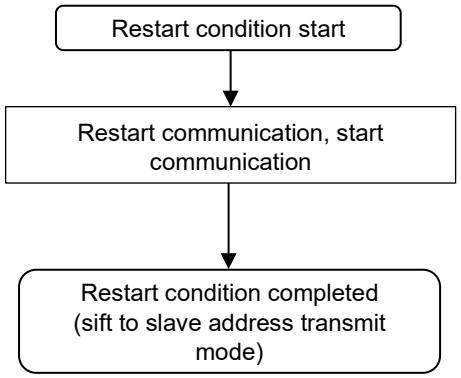
12.3.1.2 Start Condition



Set I2UM0ST bit of I2UM0CON register to “1”.

Output the start condition waveforms to I2CU0_SDA and I2CU0_SCL pins.

12.3.1.3 Restart Condition

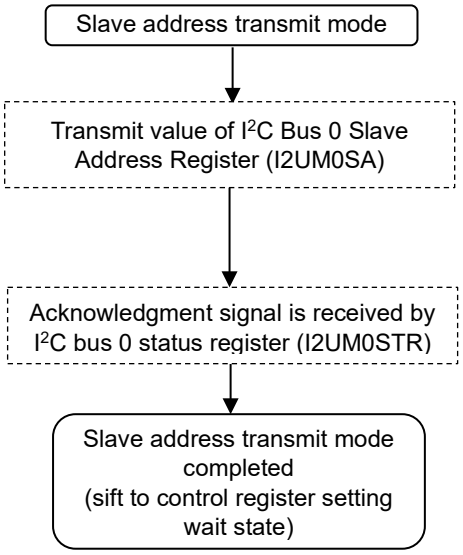


Communication in progress (I2UM0ST=1)

Set I2UM0RS=1 and I2UM0ST=1 in the I2UM0CON register

Output restart condition waveforms to I2CU0_SDA and I2CU0_SCL pins.

12.3.1.4 Slave Address Transmission Mode

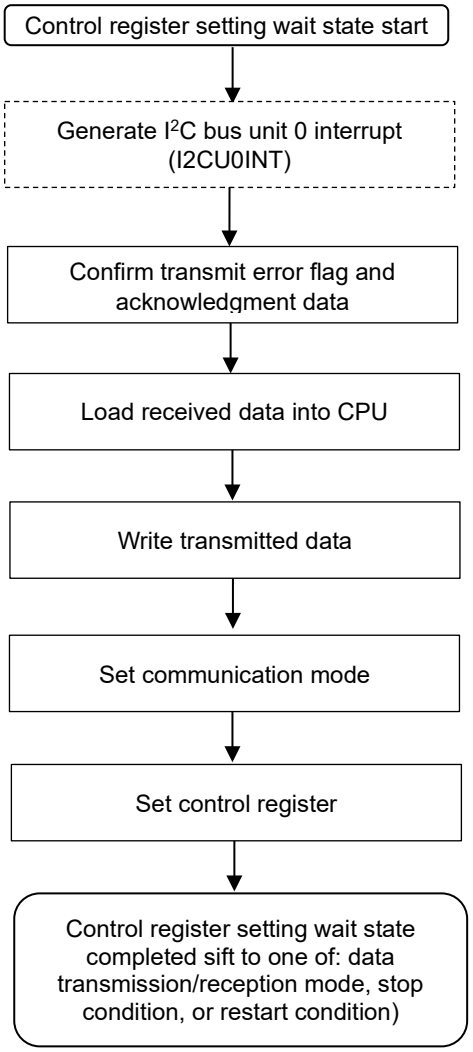


The value is transmitted from I2CU0_SDA pin in MSB first through hardware following the start condition
I2UM0A6 to I2UM0A0 bits: Slave address
I2UM0RW: Data direction (transmission/reception)
Value transmitted from the I2CU0_SDA pin is stored in the I2UM0RD register

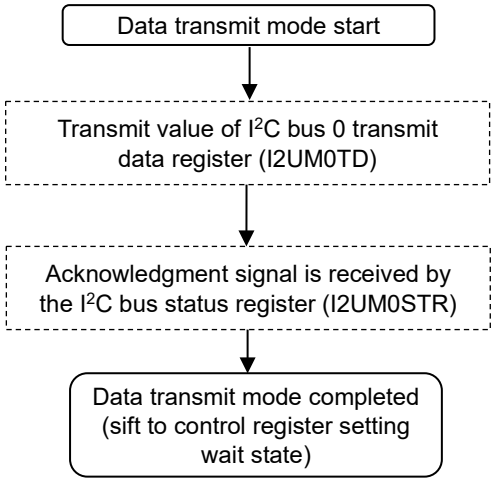
Acknowledgment signal is received through hardware
I2UM0ACR bit: Acknowledgment data

I²C bus 0 control register (I2UM0CON) setting wait state

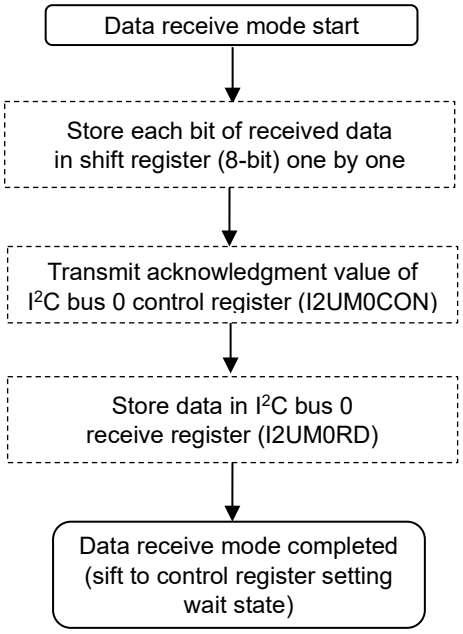
12.3.1.5 Control Register Setting Wait State



12.3.1.6 Data Transmission Mode



12.3.1.7 Data Reception Mode



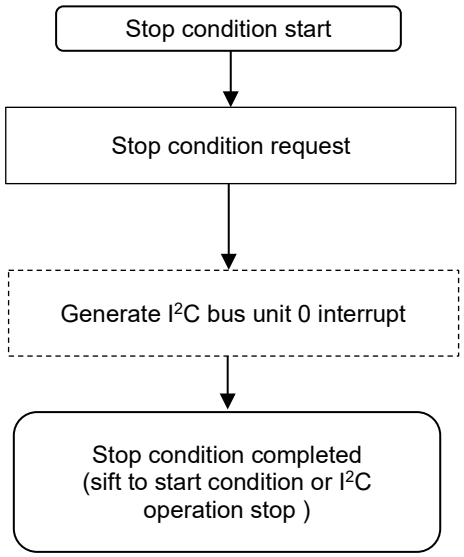
Value (received data) input to I2CU0_SDA pin is stored in synchronization with rising edge of transfer clock input to I2CU0_SCL pin in MSB first

Acknowledgment signal is transmitted through hardware
I2UM0ACT bit: Acknowledgment value
Transmitted acknowledgment value is stored in the I2UM0ACR bit of the I2UM0STA register

Received data is stored from the shift register after acknowledgment signal is transmitted
I2UM0R7 to I2UM0R0 bits: 8-bit receive data

I²C bus 0 control register (I2UM0CON) setting wait state

12.3.1.8 Stop Condition



Set I2UM0SP bit of I2UM0CON register to “1”.

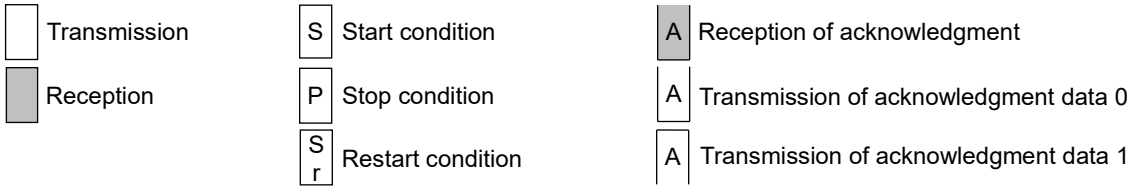
Output stop condition waveforms to I2CU0_SDA and I2CU0_SCL pins.

After the stop condition waveform is output, an interrupt is generated through hardware

Sift to start condition or I²C operation stop (I2UM0EN = 0)

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.



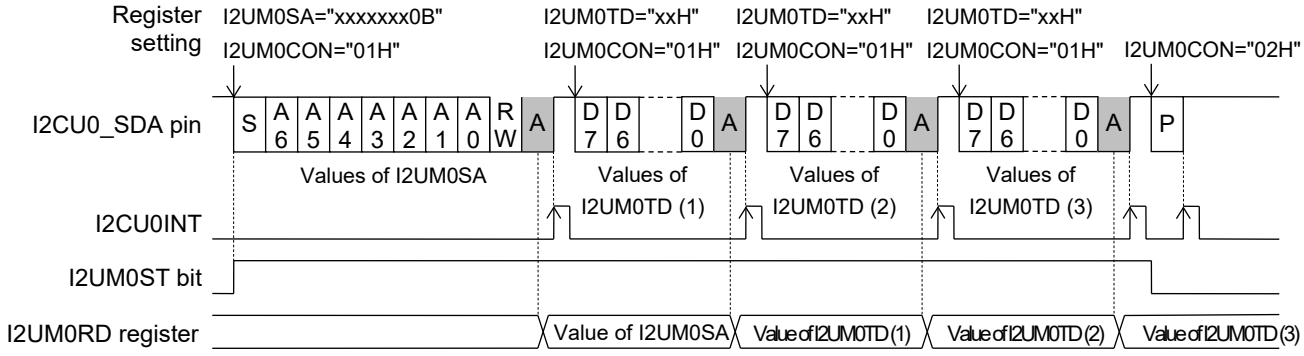


Figure 12-2 Operation timing during data transmission in the master mode

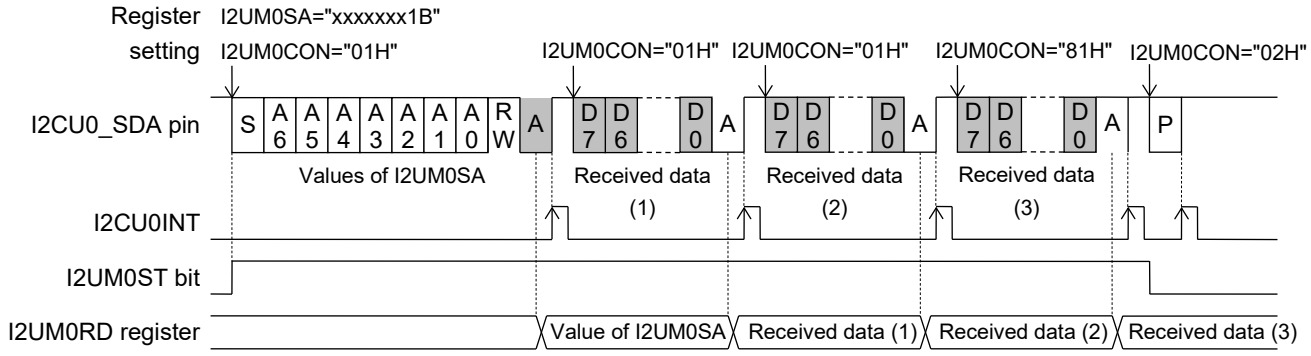


Figure 12-3 Operation timing during data reception in the master mode

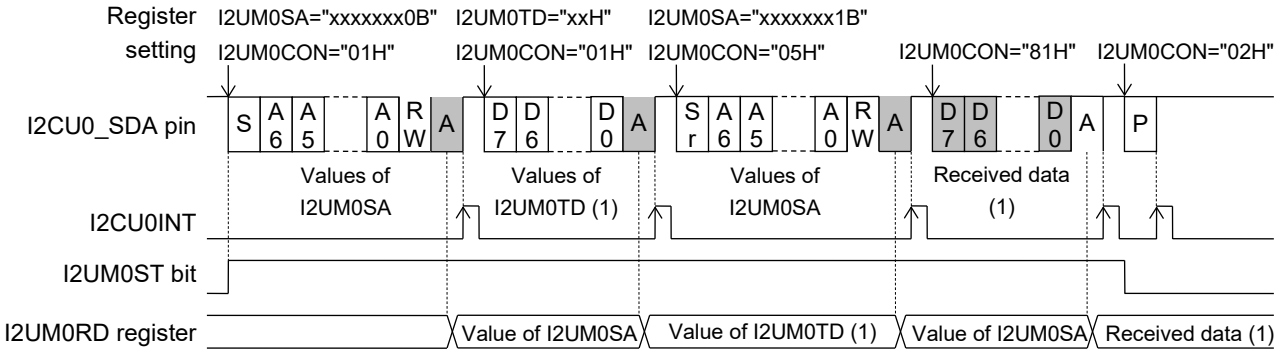


Figure 12-4 Operation timing during data transmission/ reception in the master mode

Figure 12-5 shows the operation timing and control method when an acknowledgment error occurs.

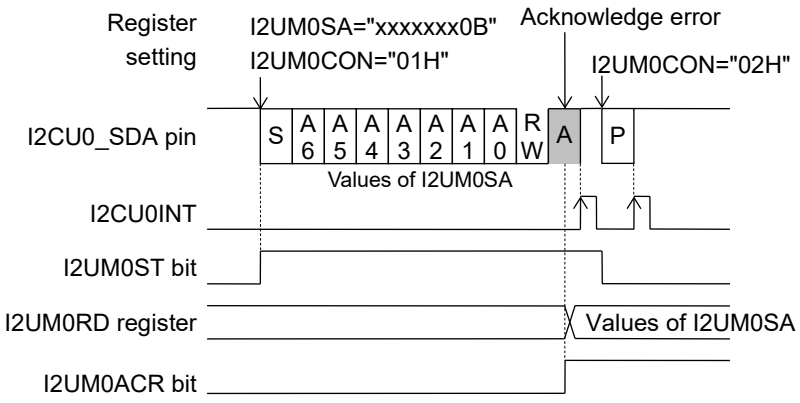


Figure 12-5 Operation suspend timing at occurrence of acknowledgment error in the master mode

When the values of the transmitted bit and the I2CU0_SDA pin do not coincide, the I2UM0ER bit of the I²C bus 0 status register (I2UM0STA) is set to "1" and the I2CU0_SDA pin output is disabled until termination of the subsequent byte data communication.
Figure 12-6 shows the operation timing and control method when transmission fails.

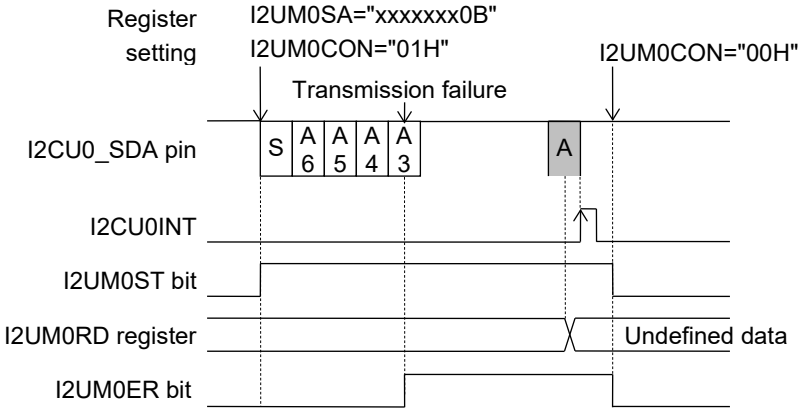
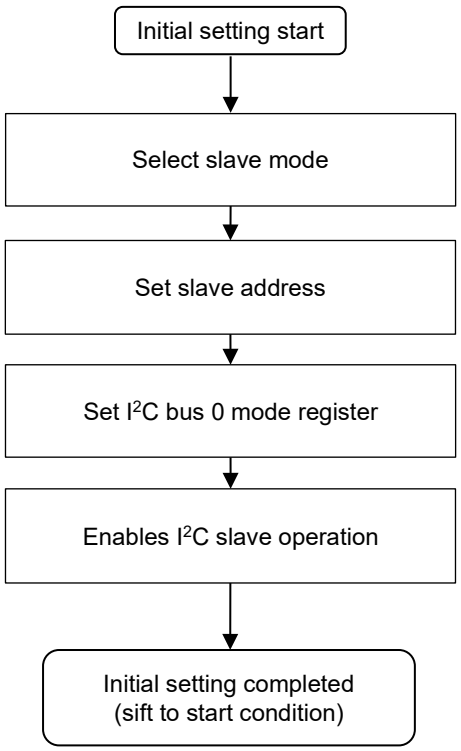


Figure 12-6 Operation timing when transmission fails in the master mode

12.3.3 Slave Operation

The following flow charts describe procedures of each operation in the slave mode.

12.3.3.1 Initial Setting of Communication Operation



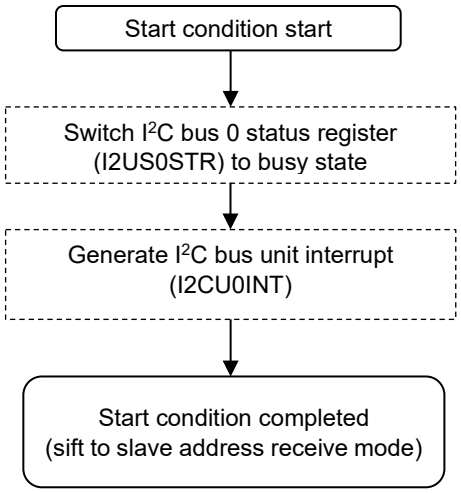
Select I2U0MD=1 in the I2U0MSS register

Set I2US0A6 to I2US0A0 bits in the I2US0SA register

Set the I2US0MD register
I2US0SIE bit: Enable/disable start condition interrupt
I2US0PIE bit: Enable/disable stop condition interrupt

Set I2US0EN=1 with the I2US0MD register

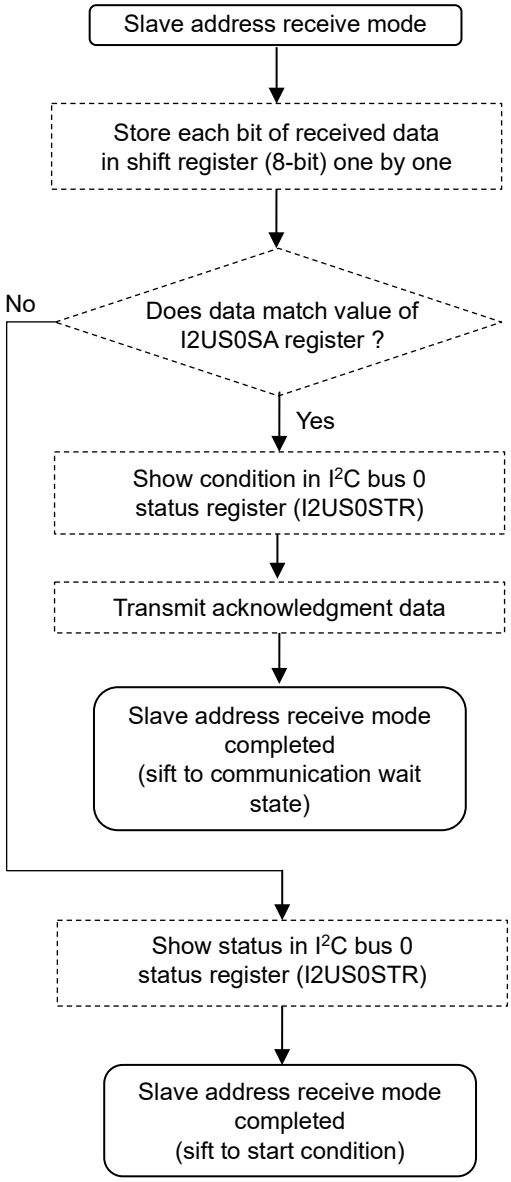
12.3.3.2 Start Condition



Value of the following bit becomes "1" when start condition waveforms are input to I2CU0_SDA and I2CU0_SCL pins:
I2US0BB bit: I²C bus busy state (I2US0BB=1)

<Only when I2US0SIE=1>
Start condition interrupt is generated

12.3.3.3 Slave Address Reception Mode



Value (slave address, data direction bit) input to I2CU0_SDA pin is stored in synchronization with rising edge of transfer clock input to I2CU0_SCL pin in MSB first

Compare slave address stored in shift register with slave address of I2US0SA

I2US0SAA bit: Data matches slave address (I2US0SAA=1)
I2US0TR bit: Transmit state/receive state

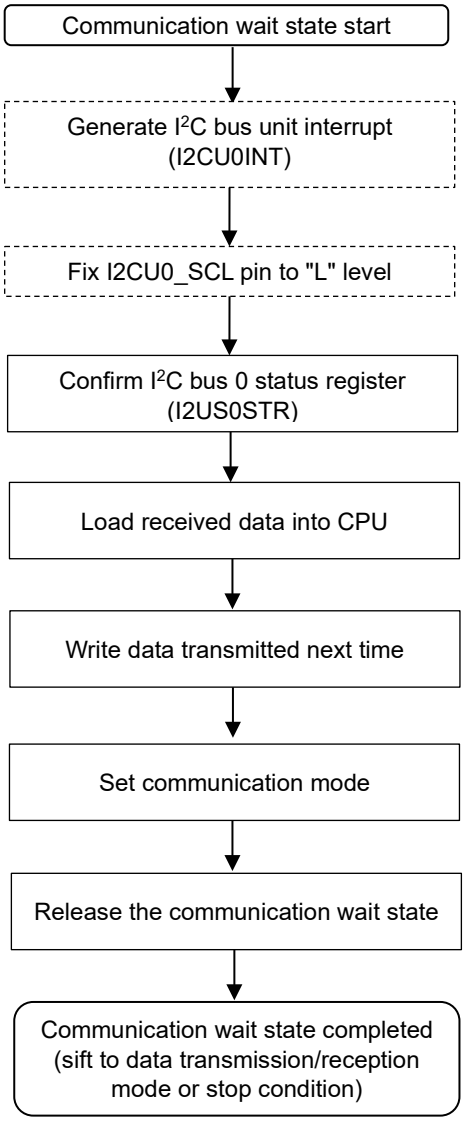
Acknowledgment data ("L" level) is output

Shift to the communication wait state after detection of falling edge of transfer clock which was input to I2CU0_SCL pin transmitting acknowledgment data

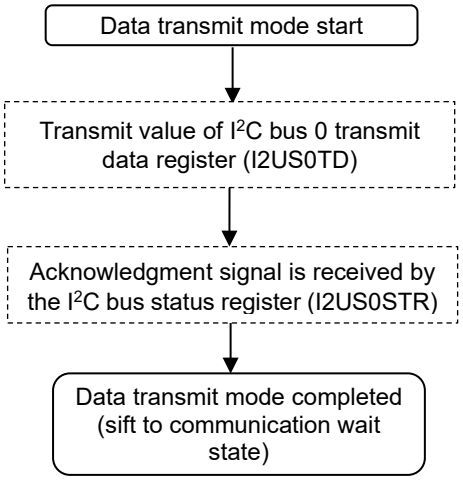
I2US0SAA bit: Data does not match slave address (I2US0SAA=0)

If not match, neither acknowledgment nor interrupt is generated

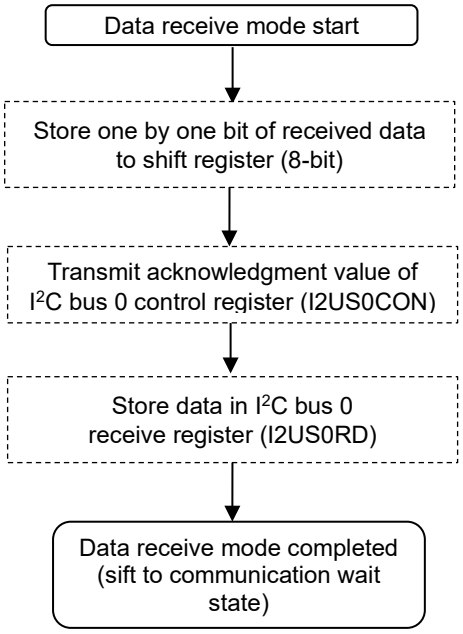
12.3.3.4 Communication Wait State



12.3.3.5 Data Transmission Mode



12.3.3.6 Data Reception Mode



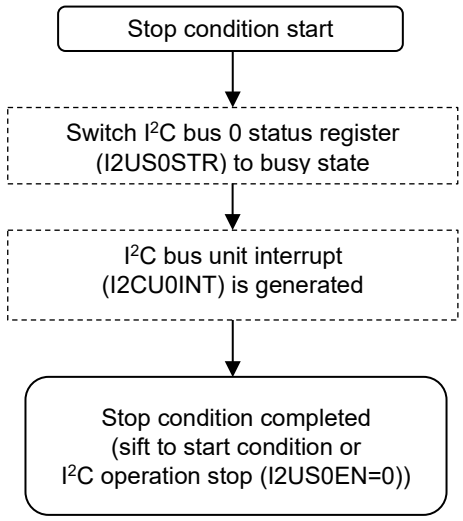
Value (received data) input to I2CU0_SDA pin is stored in synchronization with rising edge of transfer clock input to I2CU0_SCL pin in MSB first

Acknowledgment signal is transmitted through hardware
I2US0ACT bit: Acknowledgment value
Transmitted acknowledgment value is stored in the I2US0ACR bit of the I2US0STR register

Received data is stored from the shift register after acknowledgment signal is transmitted
I2US0R7 to I2US0R0 bits: 8-bit receive data

Shift to communication wait state after detection of falling edge of transfer clock which was input to I2CU0_SCL pin transmitting acknowledgment data

12.3.3.7 Stop Condition

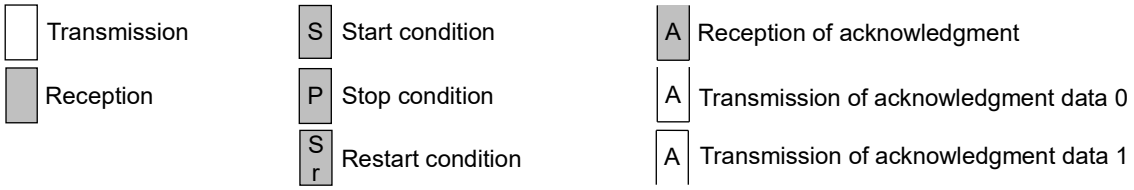


Value of the following bit becomes "0" when stop condition waveforms are input to I2CU0_SDA and I2CU0_SCL pins:
I2US0BB bit: I²C bus free state (I2US0BB=0)

<Only when I2US0PIE=1>
Stop condition interrupt is generated

12.3.4 Slave Mode Communication Operation Timing

Figures 12-7 to 12-9 show the operation timing and control method for each communication mode.



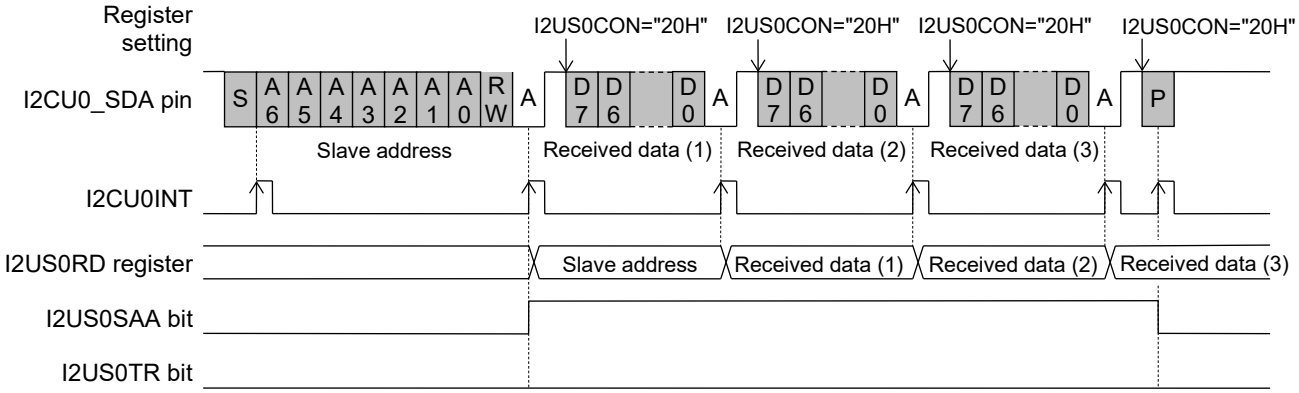


Figure 12-7 Operation Timing in Data Reception Mode When Slave Mode is Chosen

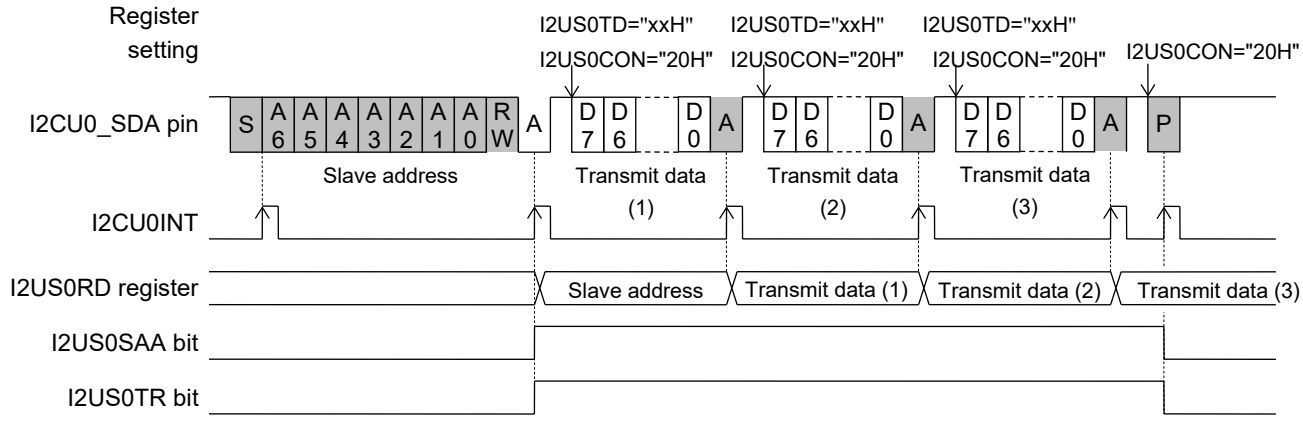


Figure 12-8 Operation Timing in Data Transmission Mode When Slave Mode is Chosen

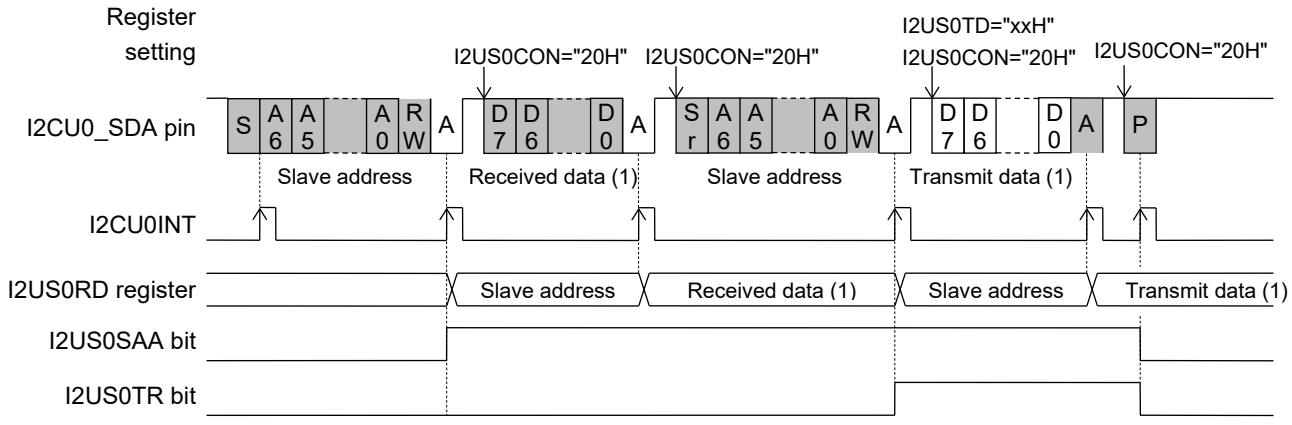


Figure 12-9 Operation Timing at Data Transmission/Reception Mode Switching When Slave Mode is Chosen

When the values of the transmitted bit and the I2CU0_SDA pin do not coincide, the I2US0ER bit of the I²C bus 0 status register (I2US0STR) is set to "1" and the I2CU0_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.

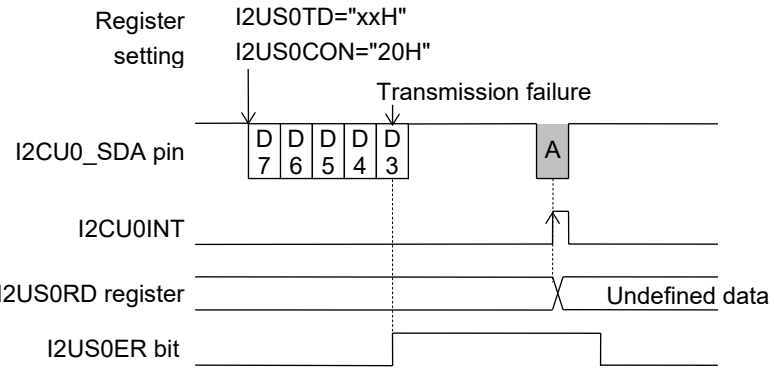


Figure 12-10 Operation Timing When Transmission Fails When Slave Mode is Chosen

[Note]

- **If entering to the STOP/STOP-D mode while the slave mode is enabled, first make sure that communication is not in progress (from coincidence of address to reception of stop condition).**

12.3.5 Operation Waveforms

Figure 12-11 shows the operation waveforms of I2CU0_SDA and I2CU0_SCL pins and the I2UM0BB flag of the I2UM0STR register. Table 12-4 and 12-5 show the relationship between communication speeds and HSCLK clock counts. Table 12-6 shows relationship between communication speeds and LSCLK clock counts.

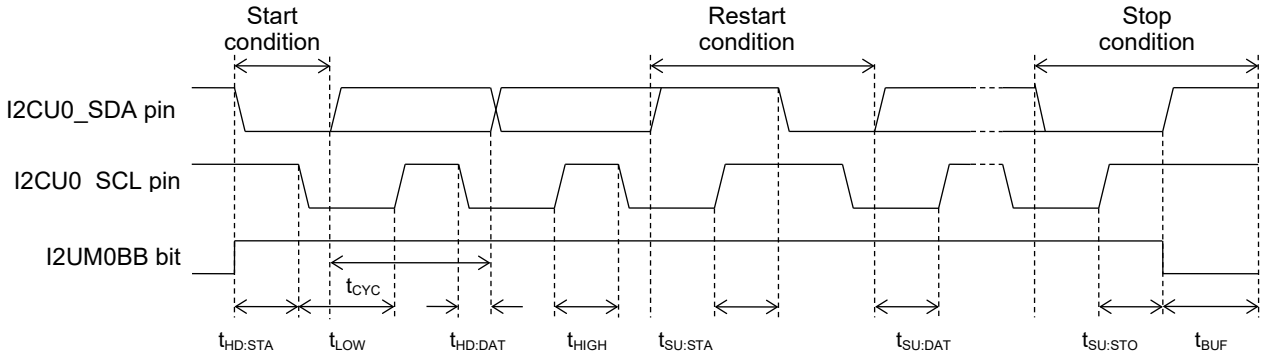


Figure 12-11 Operation Waveforms of I2CU0_SDA and I2CU0_SCL Pins and I2UM0BB Flag

Table 12-4 Relationship between Communication Speeds and HSCLK Clock Counts (at HSCLK=24 MHz)

I2UM0MOD register		t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}
Communication speed (I2UM0MD1, I2UM0MD0 bits)	Speed reduction (I2UM0DW1, I2UM0DW0 bits)									
00 (Standard mode : 100 kbps)	00 (no reduction)	240 φ	108 φ	132 φ	24 φ	108 φ	132 φ	108 φ	108 φ	132 φ
	01 (10% reduction)	264 φ	120 φ	144 φ	24 φ	120 φ	144 φ	120 φ	120 φ	144 φ
	10 (20% reduction)	288 φ	132 φ	156 φ	24 φ	132 φ	156 φ	132 φ	132 φ	156 φ
	11 (30% reduction)	312 φ	144 φ	168 φ	24 φ	144 φ	168 φ	144 φ	144 φ	168 φ
01 (Fast mode : 400 kbps)	00 (no reduction)	60 φ	24 φ	36 φ	12 φ	24 φ	36 φ	24 φ	24 φ	36 φ
	01 (10% reduction)	66 φ	27 φ	39 φ	12 φ	27 φ	39 φ	27 φ	27 φ	39 φ
	10 (20% reduction)	72 φ	30 φ	42 φ	12 φ	30 φ	42 φ	30 φ	30 φ	42 φ
	11 (30% reduction)	78 φ	33 φ	45 φ	12 φ	33 φ	45 φ	33 φ	33 φ	45 φ
10 or 11 (1 Mbps mode : 1 Mbps)	00 (no reduction)	24 φ	10 φ	14 φ	4 φ	10 φ	14 φ	10 φ	10 φ	14 φ
	01 (10% reduction)	26 φ	11 φ	15 φ	4 φ	11 φ	15 φ	11 φ	11 φ	15 φ
	10 (20% reduction)	29 φ	13 φ	16 φ	4 φ	13 φ	16 φ	12 φ	13 φ	16 φ
	11 (30% reduction)	31 φ	14 φ	17 φ	4 φ	14 φ	17 φ	13 φ	14 φ	17 φ

The above clock counts are values when HSCLK is chosen for the operating frequency (I2UM0CD2 to 0 bits of the I2UM0MOD register = "000"). When 1/2 or 1/4HSCLK is chosen, the counts increase in proportion to the dividing ratio.

When using the high-speed clock for the I²C operation, specify the following I²C operating clock frequency depending on the mode and the reference frequency of the PLL oscillation.

Standard mode: HSCLK to 1/4HSCLK
Fast mode: HSCLK to 1/2HSCLK
1Mbps mode: HSCLK to 1/2HSCLK

φ: Clock cycle of 1/m HSCLK

1/m HSCLK: Set in I2UM0CD2 to I2UM0CD0 bits of the I2UM0MOD register.
(m=1, 2, 4)

Example)

I²C operating clock frequency = 24 MHz : φ≈41.67 ns
= 12 MHz : φ≈83.33 ns
= 6 MHz : φ≈166.67 ns

Table 12-5 Relationship between Communication Speeds and HSCLK Clock Counts (at HSCLK=16MHz)

I2UM0MOD register		t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}
Communication speed (I2UM0MD1, I2UM0MD0 bits)	Speed reduction (I2UM0DW1, I2UM0DW0 bits)									
00 (Standard mode : 100 kbps)	00 (no reduction)	160 φ	72 φ	88 φ	16 φ	72 φ	88 φ	72 φ	72 φ	88 φ
	01 (10% reduction)	176 φ	80 φ	96 φ	16 φ	80 φ	96 φ	80 φ	80 φ	96 φ
	10 (20% reduction)	192 φ	88 φ	104 φ	16 φ	88 φ	104 φ	88 φ	88 φ	104 φ
	11 (30% reduction)	208 φ	96 φ	112 φ	16 φ	96 φ	112 φ	96 φ	96 φ	112 φ
01 (Fast mode : 400 kbps)	00 (no reduction)	40 φ	14 φ	26 φ	12 φ	14 φ	26 φ	14 φ	14 φ	26 φ
	01 (10% reduction)	44 φ	16 φ	28 φ	12 φ	16 φ	28 φ	16 φ	16 φ	28 φ
	10 (20% reduction)	48 φ	18 φ	30 φ	12 φ	18 φ	30 φ	18 φ	18 φ	30 φ
	11 (30% reduction)	52 φ	20 φ	32 φ	12 φ	20 φ	32 φ	20 φ	20 φ	32 φ
10 or 11 (1 Mbps mode : 1 Mbps)	00 (no reduction)	16 φ	6 φ	10 φ	4 φ	6 φ	10 φ	6 φ	6 φ	10 φ
	01 (10% reduction)	18 φ	7 φ	11 φ	4 φ	7 φ	11 φ	7 φ	7 φ	11 φ
	10 (20% reduction)	19 φ	8 φ	11 φ	4 φ	8 φ	11 φ	7 φ	8 φ	11 φ
	11 (30% reduction)	21 φ	9 φ	12 φ	4 φ	9 φ	12 φ	8 φ	9 φ	12 φ

The above clock counts are values when HSCLK is chosen for the operating frequency (I2UM0CD2 to 0 bits of the I2UM0MOD register = "000"). When 1/2HSCLK is chosen, the counts increase in proportion to the dividing ratio.
When using the high-speed clock for the I²C operation, specify the following I²C operating clock frequency depending on the mode and the reference frequency of the PLL oscillation.

Standard mode: HSCLK or 1/2HSCLK
Fast mode: HSCLK
1Mbps mode: HSCLK

φ: Clock cycle of 1/m HSCLK
1/m HSCLK: Set in I2UM0CD2 to I2UM0CD0 bits of the I2UM0MOD register.
(m=1, 2)
Example)
I²C operating clock frequency = 16 MHz : φ≈62.50 ns
= 8 MHz : φ≈125.00 ns

Table 12-6 Relationship between Communication Speeds and LSCLK Clock Counts

I2UM0MD1	I2UM0MD0	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	16 φ	8 φ	8 φ	1 φ	8 φ	8 φ	7 φ	8 φ	8 φ
0	1	4.096kbps	8 φ	4 φ	4 φ	1 φ	4 φ	4 φ	3 φ	4 φ	4 φ
1	*	8.192kbps	4 φ	2 φ	2 φ	1 φ	2 φ	2 φ	1 φ	2 φ	2 φ

The above clock counts are values when LSCLK is chosen for the operating frequency (I2UM0CD2 to 0 bits of the I2UM0MOD register = "100"). When 1/2 to 1/8LSCLK is chosen, the counts increase in proportion to the dividing ratio.
φ: Clock cycle of 1/m LSCLK
1/m LSCLK: Set in I2UM0CD2 to 0 bits of the I2UM0MOD register.
When LSCLK is used as the clock, the setting values of I2UM0DW1 and I2UM0DW0 bits of the I2UM0MOD register are ignored.
*: 1 or 0

[Note]

- **When the slave device uses the clock stretch function which holds the I2CU0_SCL pin at "L" level, the time t_{CYC} and time t_{LOW} are extended.**

Chapter 13 I^2C Bus Master

13. I²C Bus Master

13.1 General Description

ML62Q1000 series has two channels of I²C bus master that support only the master function on the I²C bus specification. This unit has the function of the I²C bus unit, described in Chapter 12, not containing the slave function and the low-speed clock (LSCLK) operation.
Table 13-1 shows the number of I²C Bus Master channels.

Table13-1 Number of I²C Bus Unit channels

Channel no.	ML62Q1300 group				ML62Q1500 / ML62Q1800 / ML62Q1700 group				
	16pin product	20pin product	24pin product	32pin product	48pin Product	52pin product	64pin product	80pin product	100pin product
0	●	●	●	●	●	●	●	●	●
1	-	-	-	-	●	●	●	●	●

●: Available -: Unavailable

13.1.1 Features

Table 13-2 shows the features of I²C bus master.

Table 13-2 Features of I2C bus master

Function	Operation mode	Features
I²C bus master	Master mode	・ Communication speed: Standard mode (100 kbps), fast mode (400 kbps), and original standard 1 Mbps mode (1Mbps) ・ Support clock stretch function for the Slave ・ 7-bit address format (only the master function supports 10-bit address format) ・ Self-test function by reading transmitted data onto the I2C bus (Safety function)

13.1.2 Configuration

Figure 13-1 shows the configuration diagram of the I²C bus master circuit.

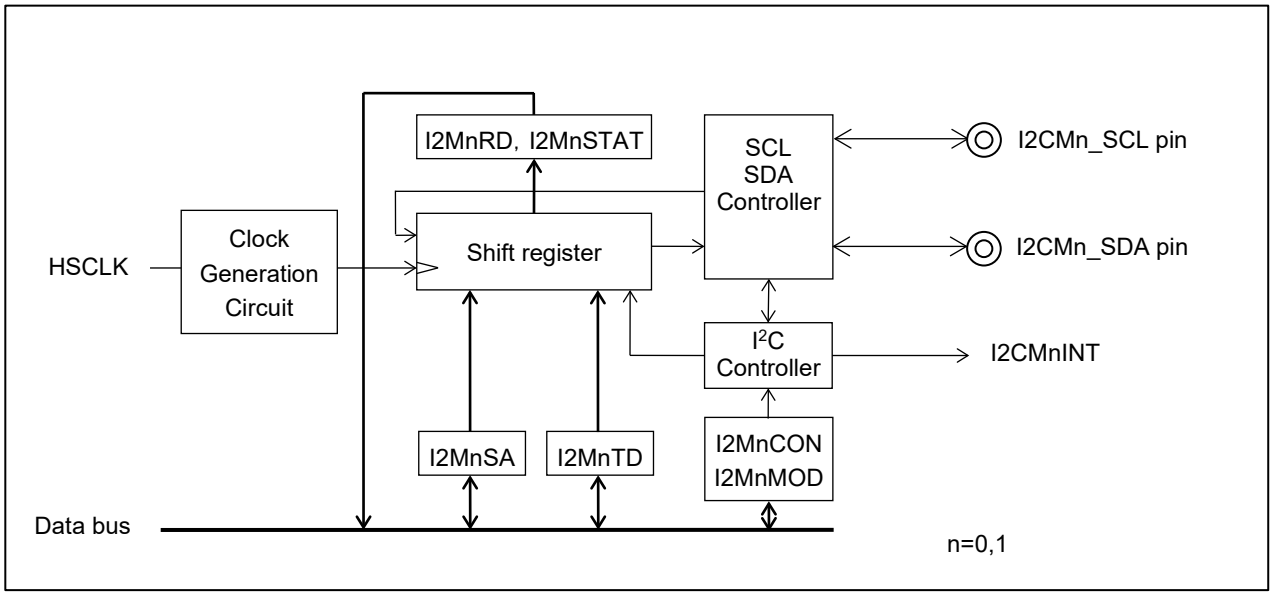


Figure 13-1 Configuration of I²C Bus Master

I2CMn_SCL: Serial Clock
I2CMn_SDA: Serial Data
I2MnRD: I²C master n receive register
I2MnSA: I²C master n slave address register
I2MnTD: I²C master n transmit data register
I2MnCON: I²C master n control register
I2MnMOD: I²C master n mode register
I2MnSTAT: I²C master n status register

13.1.3 List of Pins

The I/O pins of the I²C bus master are assigned to the shared function of the general ports.

Pin name	I/O	Function
I2CMn_SDA	I/O	I²C bus master n data I/O pin
I2CMn_SCL	I/O	I²C bus master n clock I/O pin

13.1.4 Pin Setting

I2CMn_SDA pin and I2CMn_SCL pin are assigned to multiple general ports.
Be sure to use the ports in following combinations.

Pin name	Combination 1	Combination 2
I2CM0_SDA	P06	P22
I2CM0_SCL	P07	P23
Pin name	Combination 3	
I2CM1_SDA	P61	
I2CM1_SCL	P60	

In addition to the mode setting of the shared function, choose "Enable Input, Enable Output, Nch open drain output and without pull-up" by setting following data to the port n mode register m (PnMODm).

Table 13-3 I²C bus master general port combinations

Port name	PnMODm	Combination	Setting data	ML62Q1300 group				ML62Q1500/1800/1700 group				
				16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
P06	P0MOD6	1	0x3B	-	-	-	●	●	●	●	●	●
P07	P0MOD7	1	0x3B	-	-	-	●	●	●	●	●	●
P22	P2MOD2	2	0x3B	●	●	●	●	●	●	●	●	●
P23	P2MOD3	2	0x3B	●	●	●	●	●	●	●	●	●
P60	P6MOD0	3	0x3B	-	-	-	-	●	●	●	●	●
P61	P6MOD1	3	0x3B	-	-	-	-	●	●	●	●	●

n : General port number (0 to 3) m : Bit number (0 to 7) ● : Available - : Unavailable

[Note]

- **Use external pull-up resistors for SDA pin and SCL pin referring to the I²C bus specification. The internal pull-up resistors unsatisfy the I²C bus specification. See the data sheet for each product for the value of internal pull-up resistors.**
- **Do not connect multiple master devices on the I²C bus.**

13.2 Description of Registers

13.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF6E0	Reserved	-	-	-	-	-
0xF6E1						
0xF6E2	I²C master 0 receive register	I2M0RD	-	R	8	0x00
0xF6E3	Reserved	-	-	-	-	-
0xF6E4	I²C master 0 slave address register	I2M0SA	-	R/W	8	0x00
0xF6E5	Reserved	-	-	-	-	-
0xF6E6	I²C master 0 transmit data register	I2M0TD	-	R/W	8	0x00
0xF6E7	Reserved	-	-	-	-	-
0xF6E8	I²C master 0 control register	I2M0CON	-	R/W	8	0x00
0xF6E9	Reserved	-	-	-	-	-
0xF6EA	I²C master 0 mode register	I2M0MODL	I2M0MOD	R/W	8/16	0x00
0xF6EB		I2M0MODH		R/W	8	0x02
0xF6EC	I²C master 0 status register	I2M0STAT	I2M0STR	R/W	8/16	0x00
0xF6ED		I2M0ISR		R/W	8	0x00
0xF6F0	Reserved	-	-	-	-	-
0xF6F1						
0xF6F2	I²C master 1 receive register	I2M1RD	-	R	8	0x00
0xF6F3	Reserved	-	-	-	-	-
0xF6F4	I²C master 1 slave address register	I2M1SA	-	R/W	8	0x00
0xF6F5	Reserved	-	-	-	-	-
0xF6F6	I²C master 1 transmit data register	I2M1TD	-	R/W	8	0x00
0xF6F7	Reserved	-	-	-	-	-
0xF6F8	I²C master 1 control register	I2M1CON	-	R/W	8	0x00
0xF6F9	Reserved	-	-	-	-	-
0xF6FA	I²C master 1 mode register	I2M1MODL	I2M1MOD	R/W	8/16	0x00
0xF6FB		I2M1MODH		R/W	8	0x02
0xF6FC	I²C master 1 status register	I2M1STAT	I2M1STR	R/W	8/16	0x00
0xF6FD		I2M1ISR		R/W	8	0x00

13.2.2 I²C Master n Receive Register (I2MnRD: n=0,1)

I2MnRD is a read-only special function register (SFR) used to store the received data.
The I2MnRD is updated after completion of each reception.
This register is initialized, in addition to reset function, by writing "0" to I2MnEN bit in I2MnMOD register.

Address: 0xF6E2(I2M0RD), 0xF6F2(I2M1RD)
Access: R
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2MnRD							
Bit	-	-	-	-	-	-	-	-	I2MnR7	I2MnR6	I2MnR5	I2MnR4	I2MnR3	I2MnR2	I2MnR1	I2MnR0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	I2MnR7 to I2MnR0	These bits are used to store the received data in the master mode. The signal input to the I2CM0_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 I2CM0_SCL pin. Reading this register enables following confirmation. ・ Reading when receiving data: Can confirm the received data. ・ Reading slave address or Reading when transmitting data: Can confirm the transmission data is surely transmitted.

13.2.3 I²C Master n Slave Address Register (I2MnSA: n=0,1)

I2MnSA is a special function register (SFR) to set the address and transmission/reception mode of the slave device. This register is initialized, in addition to reset function, by writing "0" to I2MnEN bit in I2MnMOD register.

Address: 0xF6E4(I2M0SA), 0xF6F4(I2M1SA)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2MnSA							
Bit	-	-	-	-	-	-	-	-	I2MnA6	I2MnA5	I2MnA4	I2MnA3	I2MnA2	I2MnA1	I2MnA0	I2MnRW
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	I2MnA6 to I2MnA0	These bits are used to set the address of the communication partner.
0	I2MnRW	This bit is used to choose direction of the data communication. 0: Data transmission mode (initial value) 1: Data Reception Mode

13.2.4 I²C Master n Transmit Data Register (I2MnTD:n=0,1)

I2MnTD is a special function register (SFR) used to set the transmission data.
This register is initialized, in addition to reset function, by writing "0" to I2MnEN bit in I2MnMOD register.

Address: 0xF6E6(I2M0TD), 0xF6F6(I2M1TD)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2MnTD							
Bit	-	-	-	-	-	-	-	-	I2MnT7	I2MnT6	I2MnT5	I2MnT4	I2MnT3	I2MnT2	I2MnT1	I2MnT0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	I2MnT7 to I2MnT0	These bits are used to set the transmission data.

13.2.5 I²C Master n Control Register (I2MnCON:n=0,1)

I2UnCON is a special function register (SFR) used to control transmission and reception operations.
This register is initialized, in addition to reset function, by writing "0" to I2MnEN bit in I2MnMOD register.

Address: 0xF6E8(I2M0CON), 0xF6F8(I2M1CON)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2MnCON							
Bit	-	-	-	-	-	-	-	-	I2MnACT	-	-	-	-	I2MnRS	I2MnSP	I2MnST
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	W	W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	I2MnACT	This bit is used to set the acknowledgment data to be output at completion of reception. 0: Acknowledgment data "0" (Initial value) 1: Acknowledgment data "1"
6 to 3	-	Reserved bits
2	I2MnRS	This bit is a write-only and used to request a restart. When "1" is written to this bit during data communication, the LSI shifts to the restart condition and the communication restarts from the slave address. "1" can be written to the I2MnRS bit only while communication is active (I2MnST = "1"). The I2MnRS bit always returns "0" for reading. 0: No restart request (Initial value) 1: Restart request
1	I2MnSP	This bit is a write-only and used to request a stop condition. When "1" is written to this bit, the LSI shifts to the stop condition and the communication stops. The I2MnSP bit always returns "0" for reading. When "1" is written to the I2MnSP bit, the I2MnSP bit is reset to "0". 0: No stop condition request (Initial value) 1: Stop condition request
0	I2MnST	This bit is used to control the communication operation of the I2C bus master. When "1" is written to this bit, the communication starts. When "1" is overwritten to this bit in a next data transmission/reception wait state after transmission/reception of acknowledgment, the data transmission/reception restarts. When "0" is written to this bit, the communication is stopped forcibly. When "1" is written to the I2MnSP bit, the I2MnST bit is reset to "0". 0: Stops communication (Initial value) 1: Starts communication

[Note]
- **Do not update the I2MnACT bit by using the bit symbol. Update it by using a byte access, not so that unintended bits are changed by the bit access instructions.**
- **When the I2MnST bit is "1", write other bits of I2MnCON register in the control register setting wait state.**

13.2.6 I²C Master n Mode Register (I2MnMOD: n=0,1)

I2UM0MOD is a special function register (SFR) used to set the operation mode.

Address: 0xF6EA(I2M0MODL/I2M0MOD), 0xF6EB(I2M0MODH),
0xF6FA(I2M1MODL/I2M1MOD), 0xF6FB(I2M1MODH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0200

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2MnMOD															
Byte	I2MnMODH								I2MnMODL							
Bit	-	-	-	-	-	-	I2MnCD1	I2MnCD0	-	-	I2MnSYN	I2MnDW1	I2MnDW0	I2MnMD1	I2MnMD0	I2MnEN
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 10	-	Reserved bits
9, 8	I2MnCD1, I2MnCD0	These bits are used to choose the I²C operating clock. 0 0: HSCLK 0 1: 1/2HSCLK 1 0: 1/4HSCLK (Initial value) 1 1: Do not use
7, 6	-	Reserved bit
5	I2MnSYN	This bit is used to choose whether to or not to use the clock stretch (handshake) function. Set this bit to "1" when using the clock stretch function. Setting this bit to "1" monitors the I²C bus, therefore the communication speed gets lower depending on the load of I²C bus. 0: Not use the clock stretch function (Initial value) 1: Use the clock stretch function
4, 3	I2MnDW1, I2MnDW0	These bits are used to set the communication speed reduction rate of the I²C bus master. Specify this bit not so that the communication speed exceeds 100 kbps/400kbps/1 Mbps. When LSCLK or 1/8 LSCLK is chosen by the I2UM0CD2 to 0 bits, "No communication speed reduction" is chosen regardless the setting of I2UM0DW1 to 0 bits. 0 0: No communication speed reduction (Initial value) 0 1: Approximately 9.1% communication speed reduction 1 0: Approximately 16.7% communication speed reduction 1 1: Approximately 23.1% communication speed reduction
2, 1	I2MnMD1, I2MnMD0	These bits are used to set the communication speed of the I²C bus unit in the master mode. 0 0: Standard mode (Initial value) (100 kbps*) 0 1: Fast mode (400 kbps*) 1 0: 1Mbps mode (1Mbps*) 1 1: 1Mbps mode (1Mbps*) * : When I2MnCD1 to 0 bits are "00" and I2MnSYN bit is "0".
0	I2MnEN	This bit is used to enable the master operation. When "1" is written to this bit, the I2MnST bit can be set and the I2MnBB bit starts operation. When "0" is written to this bit, the I²C master stops operation and the I2MnRD, I2MnSA, I2MnTD, I2MnCON and I2MnSTR are initialized. 0: Stop the I²C master operation (Initial value) 1: Enable the I²C master operation

[Note]

- **When using the high-speed clock for the I²C operation, specify the following I²C operating clock frequency depending on the mode and the reference frequency of the PLL oscillation.**
  When HSCLK = 24MHz
  Standard mode: HSCLK to 1/4HSCLK
  Fast mode: HSCLK to 1/2HSCLK
  1Mbps mode: HSCLK to 1/2HSCLK

  When HSCLK = 16MHz
  Standard mode: HSCLK to 1/2HSCLK
  Fast mode: HSCLK
  1Mbps mode: HSCLK

13.2.7 I²C Master n Status Register (I2MnSTR: n=0,1)

I2MnSTR is a special function register (SFR) to indicate the state of the I²C bus unit in the master mode.
This register is initialized, in addition to reset function, by writing "0" to I2MnEN bit in I2MnMOD register.

Address: 0xF6EC(I2M0STAT/I2M0STR), 0xF6ED(I2M0ISR)
0xF6FC(I2M1STAT/I2M1STR), 0xF6FD(I2M1ISR)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2MnSTR															
Byte	I2MnISR								I2MnSTAT							
Bit	-	-	-	-	-	I2MnSPS	I2MnDS	I2MnAS	I2MnBO	-	-	-	-	I2MnER	I2MnACR	I2MnBB
R/W	R	R	R	R	R	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

Bit No.	Bit symbol name	Description
15 to 11	-	Reserved bits
10	I2MnSPS	This bit is used to indicate the usage state of the I²C bus. This bit is set to "1" when transmitting the stop condition has been completed on the I²C bus. To reset this bit, write "1" to this bit. 0: The stop condition has not been transmitted (Initial value) 1: The stop condition has been transmitted
9	I2MnDS	This bit is used to indicate the usage state of the I²C bus. This bit is set to "1" when transmitting data or receiving data has been completed on the I²C bus. To reset this bit, write "1" to this bit. 0: The transmission/reception has not been completed (Initial value) 1: The transmission/reception has been completed
8	I2MnAS	This bit is used to indicate the usage state of the I²C bus. This bit is set to "1" when transmitting the start condition and 7 bit slave address have been completed on the I²C bus. To reset this bit, write "1" to this bit. 0: The start condition and the slave address have not been transmitted (Initial value) 1: The start condition and the slave address have been transmitted
7	I2MnBO	This bit is used to indicate the usage state of the I²C bus. This bit is set to "1" when transmitting the start condition has been completed and is reset to "0" when the time (t_{BUF}) has passed after transmitting the stop condition or there happened a data communication error on the I2CMn_SDA pin. When this bit is "1", it means the master has acquired use right of the I²C bus. To reset this bit, write "1" to this bit. 0: The use right of the I²C bus has not been acquired (Initial value) 1: The use right of the I²C bus has been acquired
6 to 3	-	Reserved bits

Bit No.	Bit symbol name	Description
2	I2MnER	This bit is used to indicate a transmission error. When a bit of transmission data and the value on the I2CMn_SDA pin do not coincide, "1" is set to this bit. To reset this bit, write "1" to this bit. When this bit is set to "1" and the clock stretch function is used (I2MnSYN = "1"), the I2CMn_SDA pin output is disabled until the subsequent byte data communication terminates. Even if this bit is set to "1", the I2CMn_SDA pin output continues until the subsequent byte data communication terminates when the clock stretch function is not used (I2MnSYN = "0"). 0: There was no transmission error (Initial value) 1: There was a transmission error
1	I2MnACR	This bit is used to store the acknowledgment signal received. Acknowledgment signals are received when the slave address is transmitted and the data transmission/reception is completed. To reset this bit, write "1" to this bit. 0: Received acknowledgment "0" (Initial value) 1: Received acknowledgment "1"
0	I2MnBB	This bit is used to indicate the usage state 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 this bit is reset to "0". To reset this bit, write "1" to this bit. 0: The status of I²C bus is free (Initial value) 1: The status of I²C bus is busy

[Note]

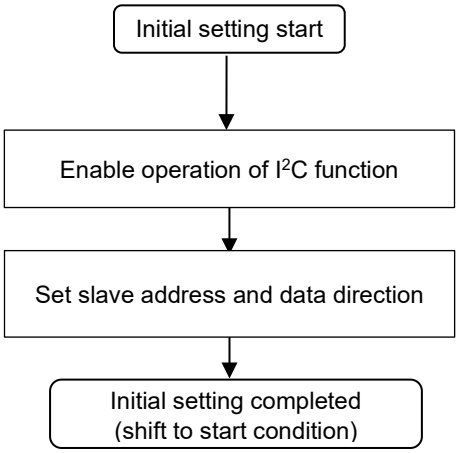
- **Do not update each bit of the I2MnSTR register by using the bit symbol. Update it by using a byte access or word access, not so that unintended bits are changed by the bit access instructions.**
- **I2MnBB bit and I2MnBO bit are reset in one I²C operating clock after writing "1" to the bits.**

13.3 Description of Operation

13.3.1 Master Operation

The following flow charts describe each operation procedure of the master.

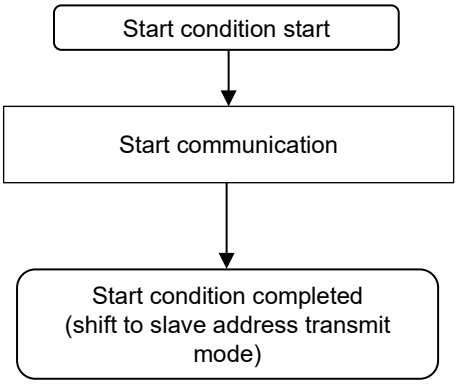
13.3.1.1 Initial Setting of Communication Operation



Set I2MnEN=1 in the I2MnMOD register

Set the I2MnSA register
I2MnA6 to I2MnA0 bits: Set slave address
I2MnRW: Set data direction (transmission/reception)

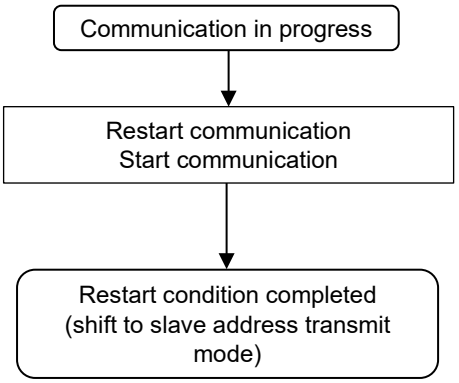
13.3.1.2 Start Condition



Set I2MnST bit of I2MnCON register to "1".

Output the start condition waveform to 2CMn_SDA and I2CMn_SCL pins.

13.3.1.3 Restart Condition

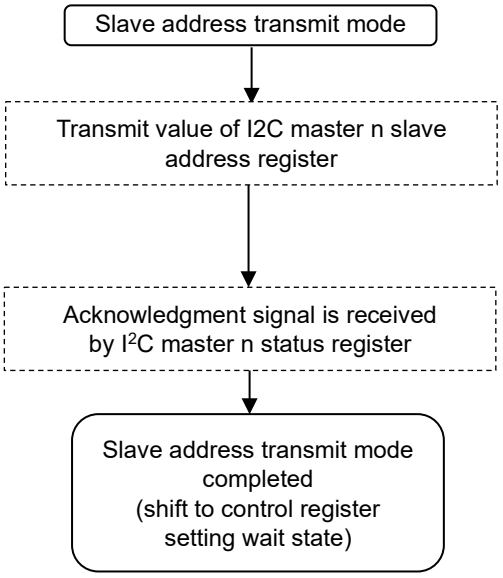


Start the restart condition
(I2MnST=1)

Set the I2MnCON register
I2MnRS=1
I2MnST=1

Output restart condition waveform to I2CMn_SDA and
I2CMn_SCL pins.

13.3.1.4 Slave Address Transmission Mode

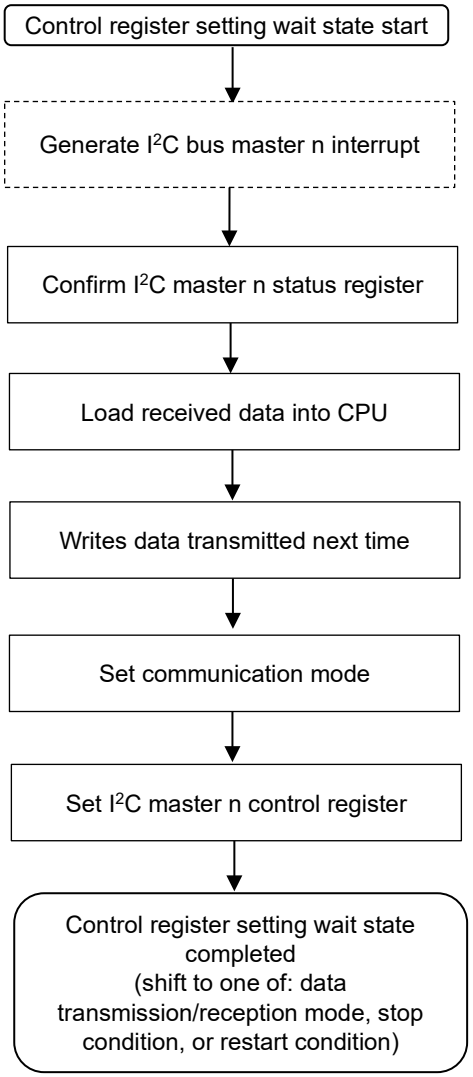


The value is transmitted from I2CMn_SDA pin in MSB first
by hardware after start condition
I2MnA6 to I2MnA0 bits: Slave address
I2MnRW: Data direction (transmission/reception)
Value transmitted from the I2CMn_SDA pin
is stored in the I2MnRD register

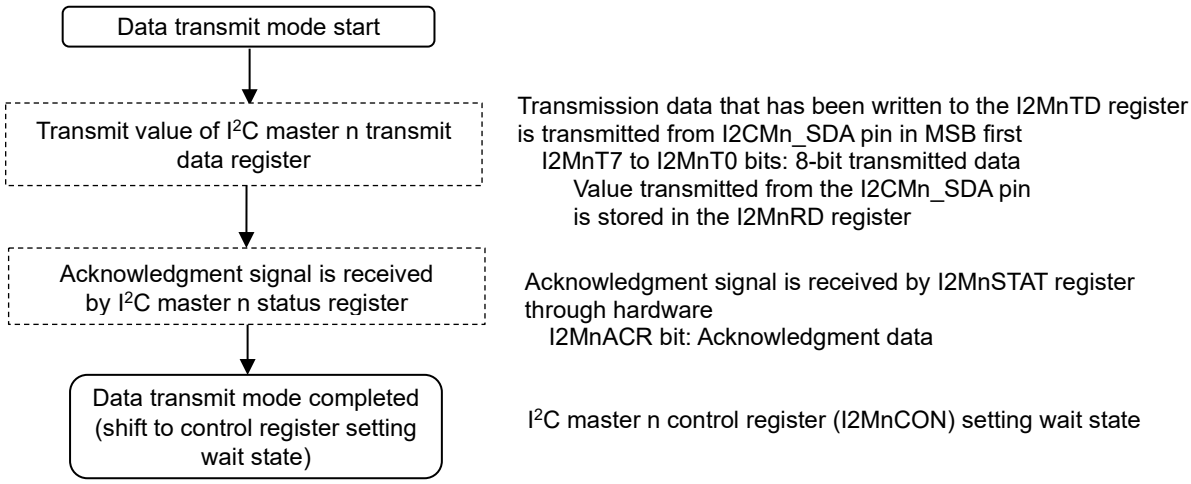
Acknowledgment signal is received by hardware
I2MnACR bit: Acknowledgment data

I²C master 0 control register (I2MnCON) setting wait state

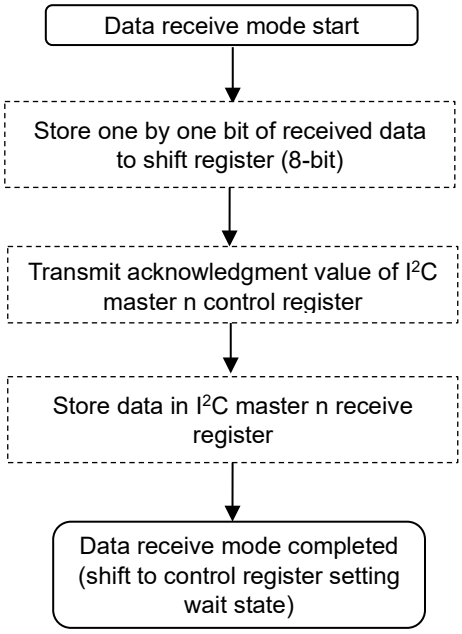
13.3.1.5 Control Register Setting Wait State



13.3.1.6 Data Transmission Mode



13.3.1.7 Data Reception Mode



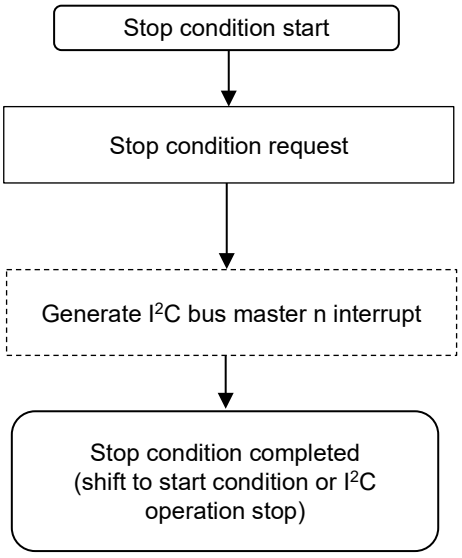
Value (receive data) input to I2CMn_SDA pin is stored in synchronization with rising edge of transfer clock input to I2CMn_SCL pin in MSB first

Transmit acknowledgment signal of I2MnCON register through hardware
I2MnACT bit: Acknowledgment value
Transmitted acknowledgment value is stored in I2MnACR bit of I2MnSTAT register

Received data is stored in the I2MnRD register from the shift register after acknowledgment signal is transmitted
I2MnR7 to I2MnR0 bits: 8-bit receive data

I²C master 0 control register (I2MnCON) setting wait state

13.3.1.8 Stop Condition



Set I2MnSP bit of I2MnCON register to “1”.

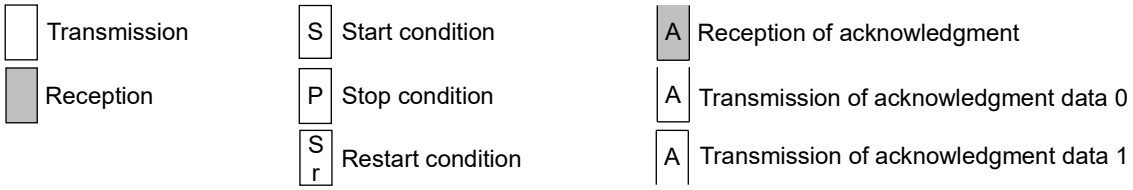
Output stop condition waveforms to 2CMn_SDA and I2CMn_SCL pins.

I2CMnIN is generated
After the stop condition waveform is output, an interrupt is generated through hardware

Move to start condition or I²C operation stop (I2MnEN = 0)

13.3.2 Communication Operation Timing

Figures 13-2 to 13-4 show the operation timing and control method for each communication mode.



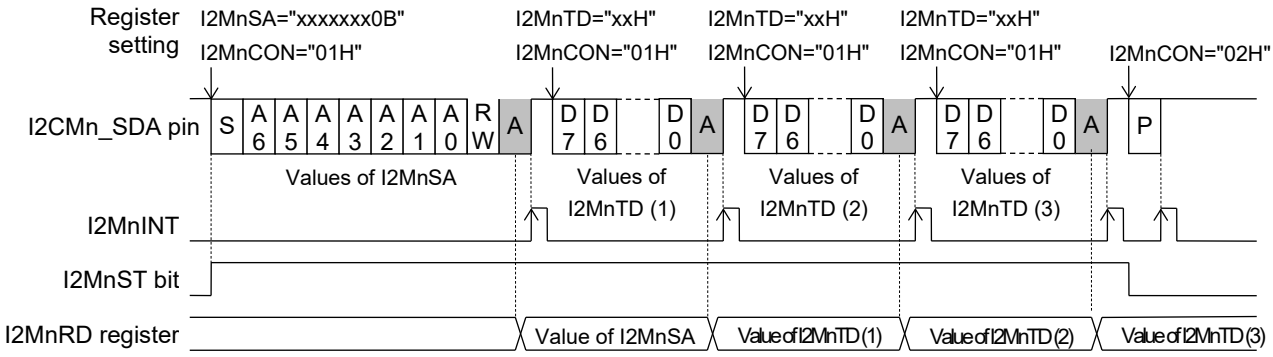


Figure 13-2 Operation timing during data transmission

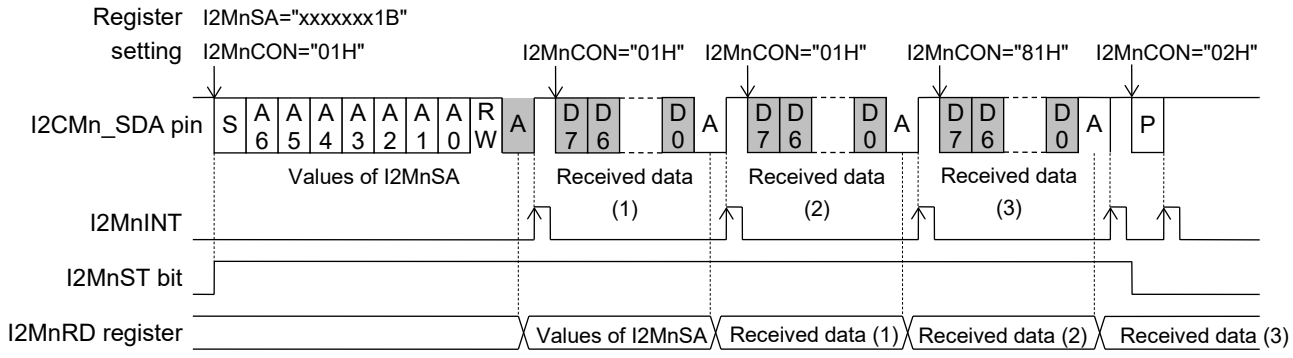


Figure 13-3 Operation timing during data reception

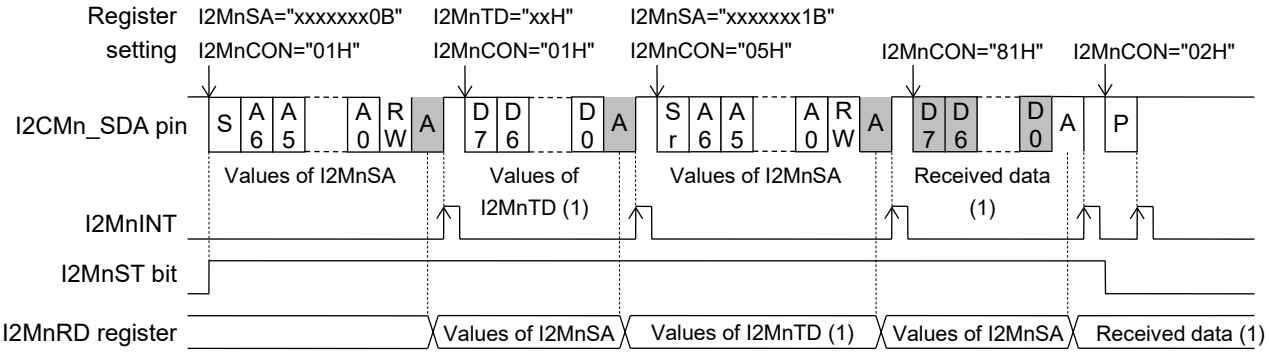


Figure 13-4 Operation timing during data transmission/ reception

Figure 13-5 shows the operation timing and control method when an acknowledgment error occurs.

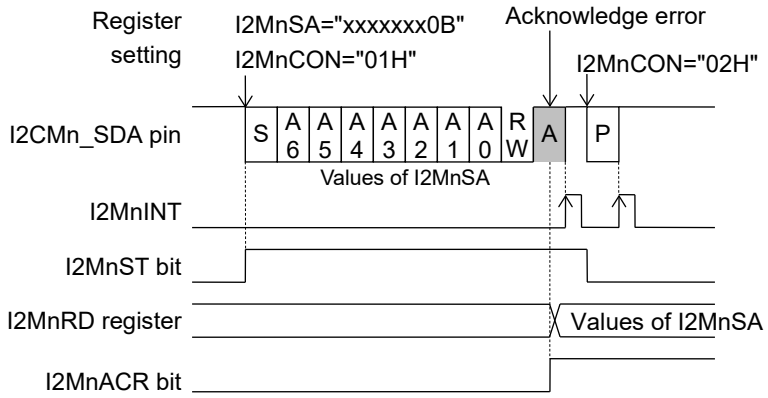


Figure 13-5 Operation suspend timing at occurrence of acknowledgment error

When the values of the transmitted bit and the I2CMn_SDA pin do not coincide, the I2MnER bit of the I²C master n status register (I2MnSTR) is set to "1" and the I2CMn_SDA pin output is disabled until termination of the subsequent byte data communication.
Figure 13-6 shows the operation timing and control method when transmission fails.

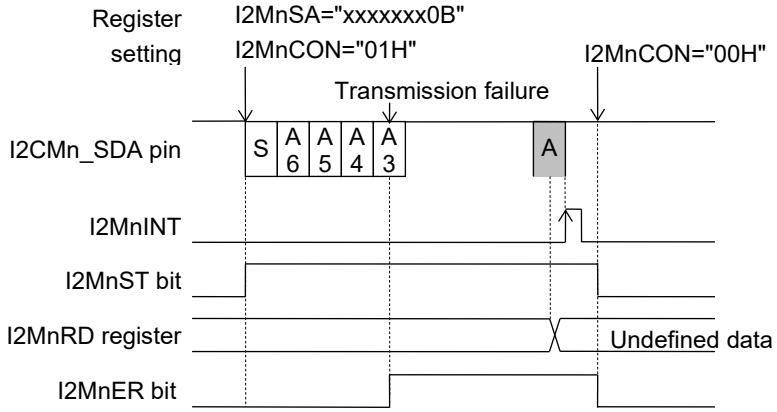


Figure 13-6 Operation timing when transmission fails

13.3.3 Operation Waveforms

Figure 13-7 shows the operation waveforms of the I2CMn_SDA and I2CMn_SCL pins and the I2MnBB flag. Tables 13-4 and 13-5 show the relationship between communication speeds and HSCLK clock counts.

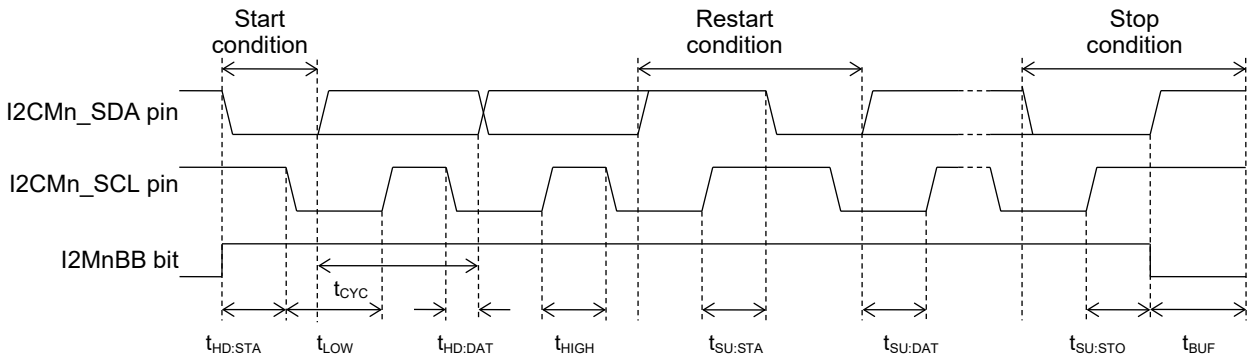


Figure 13-7 Operation Waveforms of I2CMn_SDA and I2CMn_SCL Pins and I2MnBB Flag

Table 13-4 Relationship between Communication Speeds and HSCLK Clock Counts (at HSCLK=24 MHz)

I2MnMOD register		t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}
Communication speed (I2MnMD1 and I2MnMD0 bits)	Speed reduction (I2MnDW1 and I2MnDW0 bits)									
00 (Standard mode : 100 kbps)	00 (no reduction)	240 φ	108 φ	132 φ	24 φ	108 φ	132 φ	108 φ	108 φ	132 φ
	01 (10% reduction)	264 φ	120 φ	144 φ	24 φ	120 φ	144 φ	120 φ	120 φ	144 φ
	10 (20% reduction)	288 φ	132 φ	156 φ	24 φ	132 φ	156 φ	132 φ	132 φ	156 φ
	11 (30% reduction)	312 φ	144 φ	168 φ	24 φ	144 φ	168 φ	144 φ	144 φ	168 φ
01 (Fast mode : 400 kbps)	00 (no reduction)	60 φ	24 φ	36 φ	12 φ	24 φ	36 φ	24 φ	24 φ	36 φ
	01 (10% reduction)	66 φ	27 φ	39 φ	12 φ	27 φ	39 φ	27 φ	27 φ	39 φ
	10 (20% reduction)	72 φ	30 φ	42 φ	12 φ	30 φ	42 φ	30 φ	30 φ	42 φ
	11 (30% reduction)	78 φ	33 φ	45 φ	12 φ	33 φ	45 φ	33 φ	33 φ	45 φ
10 or 11 (1 Mbps mode : 1 Mbps)	00 (no reduction)	24 φ	10 φ	14 φ	4 φ	10 φ	14 φ	10 φ	10 φ	14 φ
	01 (10% reduction)	26 φ	11 φ	15 φ	4 φ	11 φ	15 φ	11 φ	11 φ	15 φ
	10 (20% reduction)	29 φ	13 φ	16 φ	4 φ	13 φ	16 φ	12 φ	13 φ	16 φ
	11 (30% reduction)	31 φ	14 φ	17 φ	4 φ	14 φ	17 φ	13 φ	14 φ	17 φ

The above clock counts are values when HSCLK is chosen for the operating frequency (I2MnCD1 and I2MnCD0 bits of the I2MnMOD register = "00"). When 1/2 or 1/4HSCLK is chosen, the counts increase in proportion to the dividing ratio.

When using the high-speed clock for the I²C operation, specify the following I²C operating clock frequency depending on the mode and the reference frequency of the PLL oscillation.

Standard mode: HSCLK to 1/4HSCLK
Fast mode: HSCLK to 1/2HSCLK
1Mbps mode: HSCLK to 1/2HSCLK

φ: Clock cycle of 1/mHSCLK

1/mHSCLK: Depends on settings for I2MnCD1 and I2MnCD0 bits of I2MnMOD register
(m=1, 2, 4)

Example)

I²C operating clock frequency = 24 MHz : φ≈41.67 ns
= 12 MHz : φ≈83.33 ns
= 6 MHz : φ≈166.67 ns

Table 13-5 Relationship between Communication Speeds and HSCLK Clock Counts (at HSCLK=16MHz)

I2MnMOD register		t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}
Communication speed (I2MnMD1 and I2MnMD0 bits)	Speed reduction (I2MnDW1 and I2MnDW0 bits)									
00 (Standard mode : 100 kbps)	00 (no reduction)	160 φ	72 φ	88 φ	16 φ	72 φ	88 φ	72 φ	72 φ	88 φ
	01 (10% reduction)	176 φ	80 φ	96 φ	16 φ	80 φ	96 φ	80 φ	80 φ	96 φ
	10 (20% reduction)	192 φ	88 φ	104 φ	16 φ	88 φ	104 φ	88 φ	88 φ	104 φ
	11 (30% reduction)	208 φ	96 φ	112 φ	16 φ	96 φ	112 φ	96 φ	96 φ	112 φ
01 (Fast mode : 400 kbps)	00 (no reduction)	40 φ	14 φ	26 φ	12 φ	14 φ	26 φ	14 φ	14 φ	26 φ
	01 (10% reduction)	44 φ	16 φ	28 φ	12 φ	16 φ	28 φ	16 φ	16 φ	28 φ
	10 (20% reduction)	48 φ	18 φ	30 φ	12 φ	18 φ	30 φ	18 φ	18 φ	30 φ
	11 (30% reduction)	52 φ	20 φ	32 φ	12 φ	20 φ	32 φ	20 φ	20 φ	32 φ
10 or 11 (1 Mbps mode : 1 Mbps)	00 (no reduction)	16 φ	6 φ	10 φ	4 φ	6 φ	10 φ	6 φ	6 φ	10 φ
	01 (10% reduction)	18 φ	7 φ	11 φ	4 φ	7 φ	11 φ	7 φ	7 φ	11 φ
	10 (20% reduction)	19 φ	8 φ	11 φ	4 φ	8 φ	11 φ	7 φ	8 φ	11 φ
	11 (30% reduction)	21 φ	9 φ	12 φ	4 φ	9 φ	12 φ	8 φ	9 φ	12 φ

The above clock counts are values when HSCLK is chosen for the operating frequency (I2MnCD1 and I2MnCD0 bits of the I2MnMOD register = "00"). When 1/2HSCLK is chosen, the counts increase in proportion to the dividing ratio.

When using the high-speed clock for the I²C operation, specify the following I²C operating clock frequency depending on the mode and the reference frequency of the PLL oscillation.

Standard mode: HSCLK or 1/2HSCLK
Fast mode: HSCLK
1Mbps mode: HSCLK

φ: Clock cycle of 1/mHSCLK
1/mHSCLK: Depends on settings for I2MnCD1 and I2MnCD0 bits of I2MnMOD register
(m=1, 2)
Example)
I²C operating clock frequency = 16 MHz : φ≈62.50 ns
= 8 MHz : φ≈125.00 ns

[Note]

- **When the slave device uses the clock stretch function which holds the I2CMn_SCL pin at "L" level, the time t_{CYC} and time t_{LOW} are extended.**

Chapter 14 DMA Controller

14. DMA Controller

14.1 General Description

ML62Q1000 series has two channels of the Direct Memory Access Controller (DMAC), which enables to transfer data between SFR of peripheral circuits and the built-in RAM without the CPU operation.
Table 14-1 in the section 14.3.6 "DMA Transfer Target Block" shows available peripheral blocks to use as the DMA transfer source or destination.

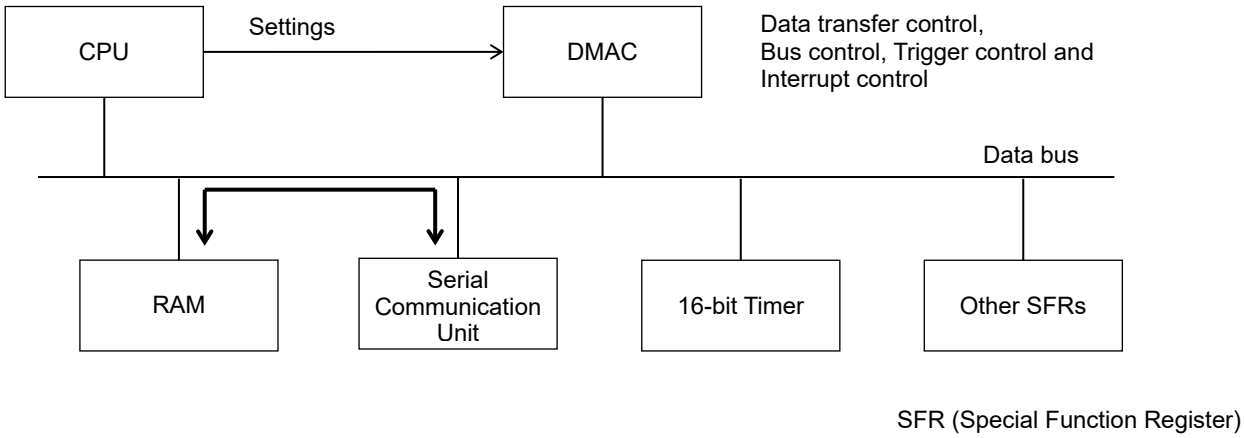


Figure 14-1 DMA Controller Overview

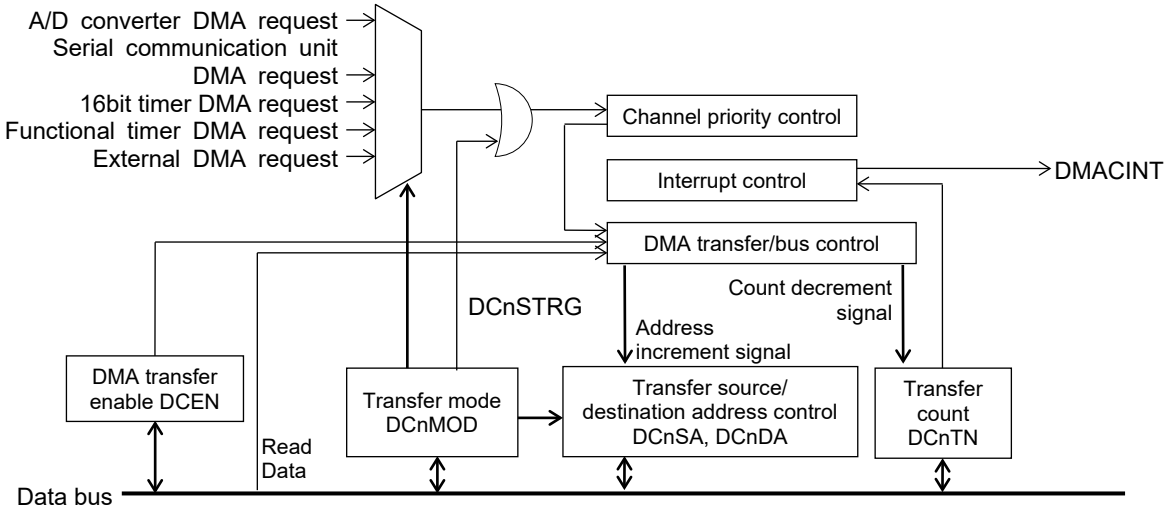
[Note]
- **Do not use the DMA controller and the Coprocessor (Hardware multiplier/divider) simultaneously.**

14.1.1 Features

- Transfer unit : 8bit/16bit
- Transfer count : 1 to 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 request : Serial communication DMA request, SA-ADC DMA request, 16bit timer DMA request, Functional timer DMA request, External DMA request and the software DMA request
- Transfer priority : Channel 0 > Channel 1 (Channel 0 has higher priority)
- Interrupt : The DMA Controller interrupt occurs when the all transfers are completed.

14.1.2 Configuration

Figure 14-2 shows the configuration of the DMA Control circuit.



DCnMOD	: DMA channel n transfer mode register
DCnTN	: DMA channel n transfer count register
DCnSA	: DMA channel n transfer source address register
DCnDA	: DMA channel n transfer destination address register
DCEN	: DMA transfer enable register
DCnSTRG	: DMA channel n software request (n = 0,1)

Figure 14-2 Configuration of DMA Controller Circuit

14.2 Description of Registers

14.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
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 address register	DC1SAL	DC1SA	R/W	8/16	0x00
0xF70D		DC1SAH		R/W	8	0x00
0xF70E	DMA channel 1 transfer destination 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	-	-	-	-	-
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	-	-	-	-	-

14.2.2 DMA Channel n Transfer Mode Register (DCnMOD: n = 0, 1)

DCnMOD is a special function register (SFR) used to set the transfer request, transfer unit and addressing mode of the transfer source and transfer destination.

Address: 0xF700(DC0MODL/DC0MOD), 0xF701(DC0MODH),
0xF708(DC1MODL/DC1MOD), 0xF709(DC1MODH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	DCnMOD															
Byte	DCnMODH								DCnMODL							
-Bit	DCnSTRG	-	-	DCnTRG4	DCnTRG3	DCnTRG2	DCnTRG1	DCnTRG0	-	-	-	DCnDS	DCnDAMD1	DCnDAMD0	DCnSAMD1	DCnSAMD0
R/W	W	-	-	R/W	R/W	R/W	R/W	R/W	-	-	-	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

Bit No.	Bit symbol name	Description
15	DCnSTRG	This bit is a write-only bit used to generate the software request of channel n. Only one transfer is performed by setting the DCnTRG bit to "1". This bit always returns "0" for reading. Writing "0": Invalid (Initial) Writing "1": Generate the software request
14, 13	-	Reserved bits
12 to 8	DCnTRG4 to DCnTRG0	These bits are used to choose the DMA transfer trigger of channel n. 00000: No DMA request (Initial value) 00001: Successive approximation type A/D (SA-ADC) DMA request 00010: Serial communication unit 0 UART Reception DMA request 00011: Serial communication unit 0 UART Transmission DMA request 00100: Serial communication unit 1 UART Reception DMA request 00101: Serial communication unit 1 UART Transmission DMA request 00110: Serial communication unit 0 SSIO Reception DMA request 00111: Serial communication unit 0 SSIO Transmission DMA request 01000: Serial communication unit 1 SSIO Reception DMA request 01001: Serial communication unit 1 SSIO Transmission DMA request 01010 to 01111: Do not use 10000: 16-bit timer 0 DMA request 10001: 16-bit timer 1 DMA request 10010: 16-bit timer 2 DMA request 10011: 16-bit timer 3 DMA request 10100: Functional timer 0 DMA request 10101: Functional timer 1 DMA request 10110: Functional timer 2 DMA request 10111: Functional timer 3 DMA request 11000: External 0 DMA request 11001: External 1 DMA request 11010: External 2 DMA request 11011: External 3 DMA request 11100: External 4 DMA request 11101: External 5 DMA request 11110: External 6 DMA request 11111: External 7 DMA request

Bit No.	Bit symbol name	Description
7 to 5	-	Reserved bits
4	DCnDS	This bit is used to set the transfer data unit of channel n. 0: 8bit (Initial) 1: 16bit
3, 2	DCnDAMD1, DCnDAMD0	These bits are used to set the addressing mode of the transfer destination of channel n. 00: Fixed addressing mode (Initial value) The transfer source address or transfer destination address is fixed. 01: Increment addressing mode DCnDA register is incremented by one in 8-bit transfer mode (DCnDS bit = 0) and incremented by two in 16-bit transfer mode (DCnDS bit = 1). 10: Decrement addressing mode DCnDA register is decremented by one in 8-bit transfer mode (DCnDS bit = 0) and decremented by two in 16-bit transfer mode (DCnDS bit = 1). 11: Do not use (Decrement addressing mode)
1, 0	DCnSAMD1, DCnSAMD0	These bits are used to set the addressing mode of the transfer source of channel n. 00: Fixed addressing mode (Initial value) The transfer source address or transfer destination address is fixed. 01: Increment addressing mode DCnSA register is incremented by one in 8-bit transfer mode (DCnDS bit = 0) and incremented by two in 16-bit transfer mode (DCnDS bit = 1). 10: Decrement addressing mode DCnSA register is decremented by one in 8-bit transfer mode (DCnDS bit = 0) and decremented by two in 16-bit transfer mode (DCnDS bit = 1). 11: Do not use (Decrement addressing mode)

[Note]

- **Set the bits except for DCnSTRG bit when the transfer is disabled (DCnEN bit of DCEN register = 0).**
- **When performing the software request 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 16-bit timer DMA request, choose the 16-bit timer mode by setting THn8BM bit of 16-bit timer n mode register (TMHnMOD) to "0".**

14.2.3 DMA Channel n Transfer Count Register (DCnTN: n = 0, 1)

DCnTN is a special function register (SFR) used to set the transfer count for channel n.

Address: 0xF702(DC0TNL/DC0TN), 0xF703(DC0TNH),
0xF70A(DC1TNL/DC1TN), 0xF70B(DC1TNH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	DCnTN															
Byte	DCnTNH								DCnTNL							
Bit	-	-	-	-	-	-	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
R/W	-	-	-	-	-	-	R/W	R/W	R/W	R/W	R/W	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

Bit No.	Bit symbol name	Description
15 to 10	-	Reserved bits
9 to 0	d9 to d0	This register is used to set the transfer count for channel n. The transfer count can be set to between 1 and 1024. As the transfer count is decremented by one in every DMA transfer, the rest of the transfer count can be checked by reading this register. The DMA interrupt (DMACINT) request is generated after the DMA transfer is completed when the transfer count is changed from "0x0001" to "0x0000". This register becomes 0x0000 at the DMA transfer completion. 0000000000: 1024 (Initial) 0000000001: 1 0000000010: 2 0000000011: 3 …… …… …… 1111111101: 1021 1111111110: 1022 1111111111: 1023

[Note]

- **Set the DCnTN register 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)

DCnSA 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 addressing mode, an address is incremented or decremented every transfer. The address is incremented or decremented by one in the 8-bit data transfer and is incremented or decremented by two in the 16-bit data transfer. The lowest bit is ignored (d0=0) and the even addresses are targeted in the 16-bit data transfer.

Address: 0xF704(DC0SAL/DC0SA), 0xF705(DC0SAH),
0xF70C(DC1SAL/DC1SA), 0xF70D(DC1SAH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	DCnSA															
Byte	DCnSAH								DCnSAL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
R/W	R/W	R/W	R/W	RW	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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

[Note]

- **Set the DCnSA register when the transfer is disabled (DCnEN bit = 0). The DCnTN is not writable if the transfer is enabled (DCnEN bit = 1).**
- **If the transfer is stopped (DCnEN bit = 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)

DCnSA 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 addressing mode, an address is incremented or decremented every transfer. The address is incremented or decremented once (±1) in the 8-bit data transfer and is incremented or decremented twice (±2) in the 16-bit data transfer. The lowest bit is ignored (d0=0) and even-numbered addresses become the targets in the 16-bit data transfer.

Address: 0xF706(DC0DAL/DC0DA), 0xF707(DC0DAH),
0xF70E(DC1DAL/DC1DA), 0xF70F(DC1DAH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	DCnDA															
Byte	DCnDAH								DCnDAL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
R/W	R/W	R/W	R/W	RW	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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

[Note]

- **Set the DCnDA register when the transfer is disabled (DCnEN bit = 0). The DCnTN is not writable if the transfer is enabled (DCnEN bit = 1).**
- **If the transfer is stopped (DCnEN bit = 0) before finishing the specified transfer count, the value of the DCnSA is not guaranteed. Reconfigure the DCnDA when restarting the transfer.**

14.2.6 DMA Transfer Enable Register (DCEN)

DCEN is a special function register (SFR) used to enable the DMA transfer and to set the behavior in the case two channels work competing.

Address: 0xF720
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								DCEN							
Bit	-	-	-	-	-	-	-	-	DCF	-	-	-	-	-	DC1EN	DC0EN
R/W	-	-	-	-	-	-	-	-	R/W	-	-	-	-	-	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	DCF	This bit is used to choose the behavior of DMA, whether to fix the channel until all the transfer, the count of which is set in the DMA channel n transfer count register (DCnTN), is completed. When this bit is set to "1", a transfer request of another channel is ignored until the current transfer is completed. When DCF bit is 0 and both DC0EN bit and DC1EN bit are set to “1”, the channel which generates a transfer request first starts to transfer. If both channels generate request at the same time, channel 0 starts to transfer. However, see “14.3.3 DMA Transfer Request” for the request generated during DMA process. 0: DMA transfer channel is free (Initial value) 1: DMA transfer channel is fixed
6 to 2	-	Reserved bits
1, 0	DC1EN, DC0EN	DCnEN (n = 0, 1) bit is used to enable the DMA transfer channel n. On the condition of this bit set to "1", the DMA transfer starts when a transfer request is generated. When the transfer count set in the DMA channel n transfer count register (DCnTN) is completed, the DCnEN bits are automatically reset to "0". Reset the DCnEN bit to "0" to stop the DMA transfer. 0: Stop the DMA channel n transfer (Initial value) 1: Enable the DMA channel n transfer

[Note]

- **Set DCF bit when the transfer is disabled (DCnEN bit = 0). It is invalid to write DCF bit when the transfer is enabled (DCnEN bit = 1).**
- **When the specified transfer count of channel n is completed, the DCnISTA bit of the DMA status register (DSTAT) is set to "1". Be sure to clear the DCnISTA bit by using the DMA interrupt status clear register (DICLR) before enabling the next DMA transfer. When the DCnISTA bit is "1", the DMA transfer can’t be enabled. Clear the status bit (DCnISTA) regardless using or not using the DMA interrupt.**

14.2.7 DMA Status Register (DSTAT)

DSTAT is a special function register (SFR) used to indicate the status of the DMA transfer channels.

Address: 0xF722(DSTATL/DSTAT), 0xF723(DSTATH)
Access: R
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	DSTAT															
Byte	DSTATH								DSTATL							
Bit	-	-	-	-	-	-	DC1STA	DC0STA	-	-	-	-	-	-	DC1ISTA	DC0ISTA
R/W	-	-	-	-	-	-	R	R	-	-	-	-	-	-	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 10	-	Reserved bits
9, 8	DC1STA, DC0STA	These bits indicate that channel n is transferring, in the transfer channel fixed (DCF bit of DCEN register is "1"). These bits are fixed to "0" in the transfer channel free (DCF bit = "0"). 0: DMA channel n is stopped (Initial value) 1: DMA channel n is transferring
7 to 2	-	Reserved bits
1, 0	DC1ISTA, DC0ISTA	These bits indicate that DMA transfer of channel n is completed. Check the transfer is finished after the interrupt occurred by reading the DCnISTA bit. The DCnISTA bits are cleared by writing "1" to corresponding bit of DMA interrupt status clear register (DICLR). 0: DMA channel n is running or stopped (Initial value) 1: Operation of DMA channel n is completed

14.2.8 DMA Interrupt Status Clear Register (DICLR)

DICLR is a special function register (SFR) used to clear the interrupt status of the DMA transfer channel.

Address: 0xF724
Access: W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								DICLR							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DICLR1	DICLR0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1	DICLR1	DC1ISTA bit of DSTAT register is cleared to "0" by writing "1" to this DICLR1 bit. The DICLR1 bit always return "0" for reading. Writing "0": Invalid Writing "1": Clears the DMA channel 1 interrupt status
0	DICLR0	DC0ISTA bit of DSTAT register is cleared to "0" by writing "1" to this DICLR0 bit. The DICLR0 bit always return "0" for reading. Writing "0": Invalid Writing "1": Clears the DMA channel 0 interrupt status

14.3 Description of Operation

The DMA controller can be used to transfer data between special function registers (SFRs) of peripheral circuits and data memory (RAM) without the CPU intervention.
After selecting the transfer unit, transfer count, transfer addressing, and enabling the DMA transfer, once the specified transfer count is completed, the DMA controller interrupt request is generated.

14.3.1 Procedure to Use DMA Controller

The following chart shows the DMA controller setting and termination procedure.

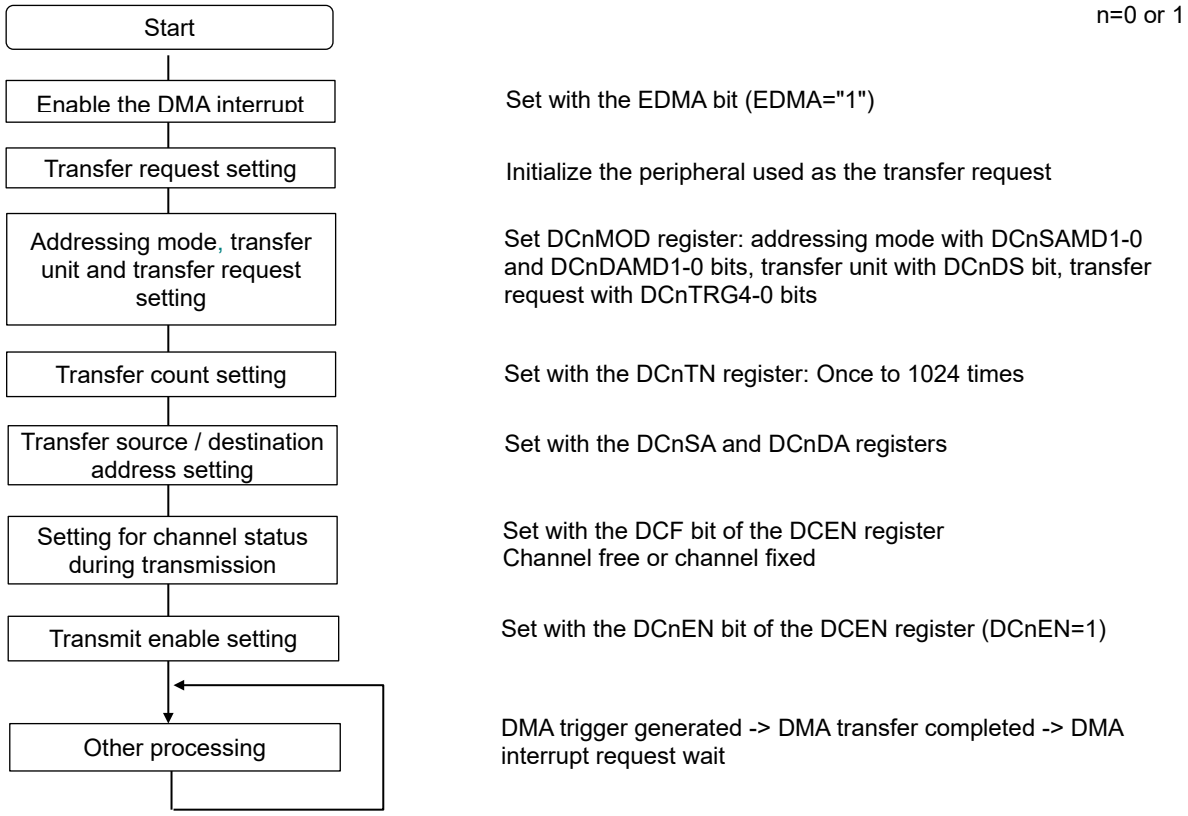


Figure 14-3 DMA Setting Procedure

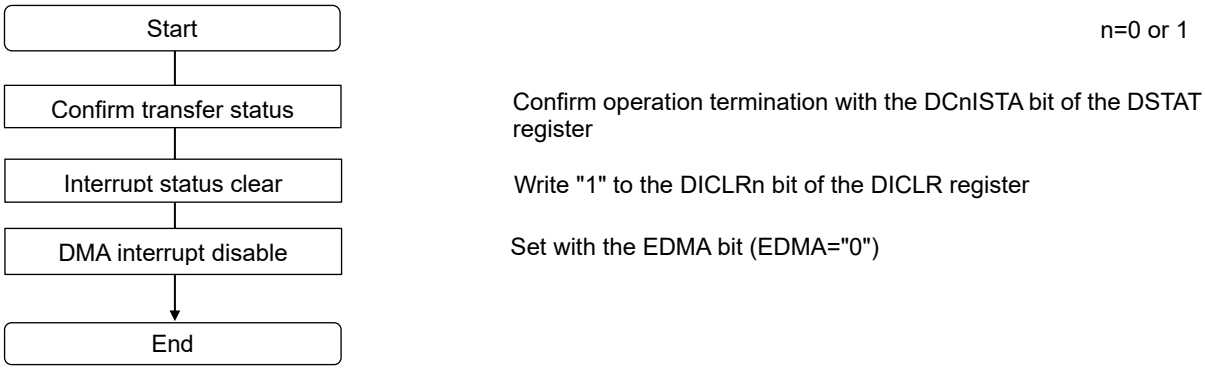


Figure 14-4 DMA Termination Procedure

[Note]

- **If a non-existing address is set, operation is not guaranteed.**

14.3.2 DMA transfer Operation Timing Diagram

The MCU has 2 channels DMA controllers.
In the DMA transfer, when DCnEN is set to “1”, the software request or the request from a peripheral chosen in DCnMOD register is generated, then a data is read from the address set in DCnSA register and it is written to the address set in DCnDA register. Once a transfer is completed, the value in DCnTN register is decreased by one, and the source/destination address set in DCnSA/DCnDA register is updated. When the value of DCnTN register becomes “0”, DCnEN becomes “0”.
The following chart shows an operation timing diagram with the transfer source set to RAM/increment addressing, transfer destination to SFR/fixed address, transfer unit to 8-bit, and transfer count to twice.

1. Set DCnEN of the DMA transfer enable register (DCEN) to "1" to enable transfer. When a transfer request is generated, the transfer automatically starts.
2. Once the DMA transfer has been performed up to the number of times set in the DMA channel n transfer count register (DCnTN), the DMA interrupt (DMACINT) request is generated.
3. Read the DCnISTA bit of the DMA status register (DSTAT) to find the channel in which the transfer is completed and write "1" to the DICLRn bit to clear the DCnISTA bit.

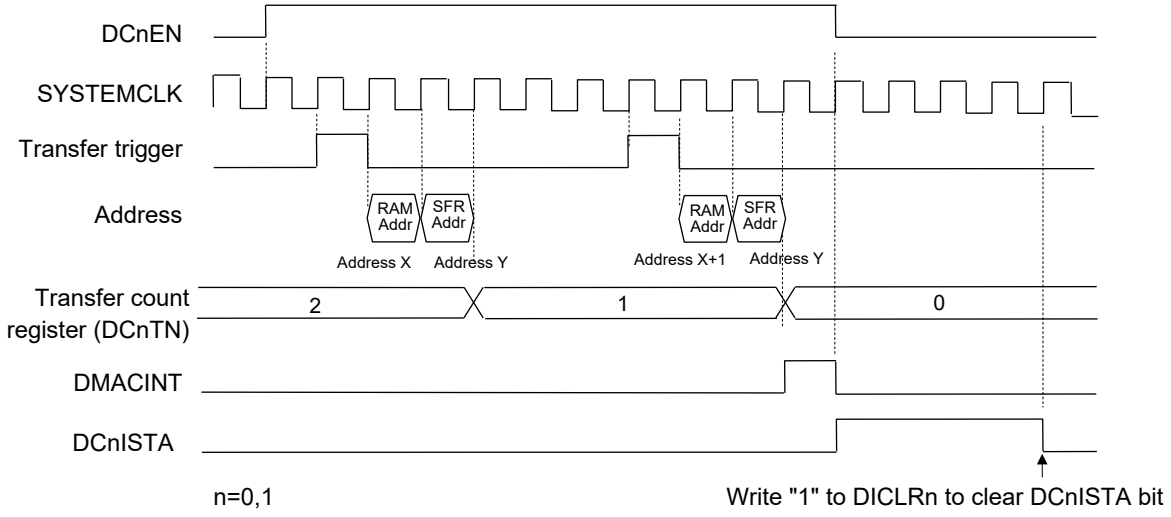
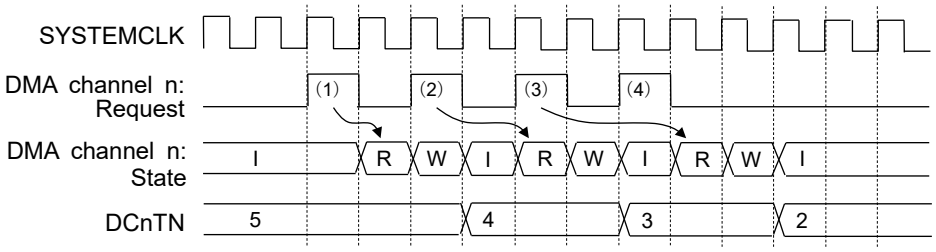


Figure 14-5 Operation Timing Diagram
(transfer source=RAM/increment addressing, transfer destination=SFR/fixed addressing, transfer unit=8-bit, transfer count=twice)

14.3.3 DMA Transfer Request

The DMA transfer shares a data bus and it has priority order (CPU > DMA channel 0 > DMA channel 1). The DMA transfer is put on hold while the CPU is using data bus. The DMA controller starts the process when the data bus is free. The DMA process has state that changes from "Idle", through "Read from transfer source" and "Write to transfer destination", to "Idle". The DMA controller ignores the request (including a request by the software) generated during the DMA process until the state changes to "Read from transfer source". When using two channels of DMA controller, one channel DMA process is deferred to start until another channel DMA process is completed and stopped. Minimum interval time of the transfer request is 2 clocks of the system clock on the condition of that CPU does not occupy the data bus, it also takes 3 clocks to return to the idle state.
It takes the following interval time for the periodical transfers.
When one channel of DMAC (Channel 0 or Channel 1) is use : 3 clocks of system clock
Channel 0 when two channels of DMAC are used : 3 clocks of system clock
Channel 1 when two channels of DMAC are used : 6 clocks of system clock

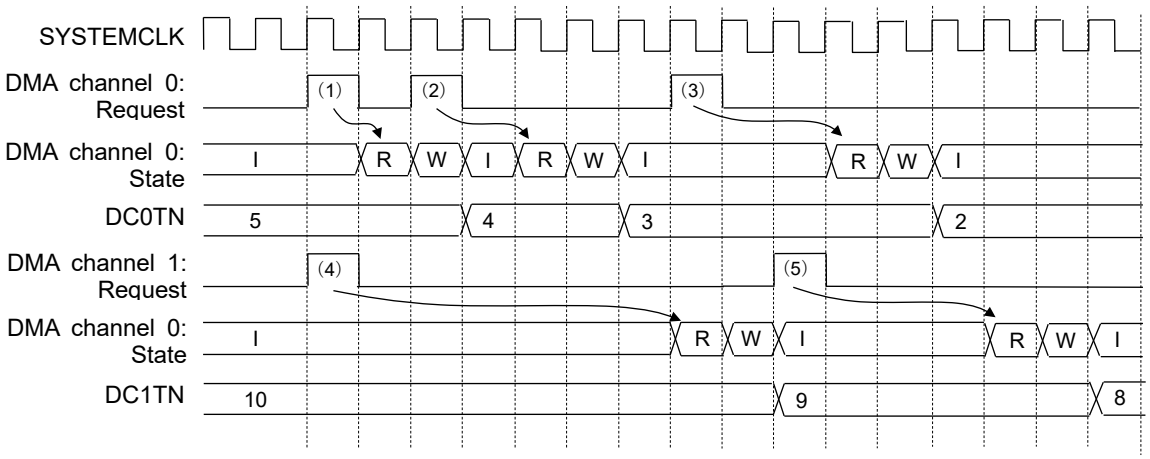
Figure 14-6 shows an example when using one channel of the DMA controller.
The request (1) "Read from transfer source" "Write to transfer destination" start at the timing of following clock. The DMA transfer for the request (2) and (3) start after the read and write processes for the previous request has been completed and it returns to the Idle state. The request (4) is ignored as the request occurred before the process of "Read from transfer source" for the previous request (3).



I: Idle, R: Read from transfer source, W: Write to transfer destination

Figure 14-6 Transfer Request when using one channel

Figure 14-7 shows an example when using two channels of the DMA controller.
According to the request (1), the process of the DMA starts in the same way as that aforementioned in Figure14-6. The request (2) takes priority over the pre-occurred request (4) as the DMA transfer in the channel 1 for the request (4) has not started. The DMA transfer for the request (3) in the channel 0 starts after the process in the channel 1 has been completed.
The request (4) in the channel 1 has lower priority than the request (1) and the next request (2) in the channel 0. The process for the request (5) also starts after the process for the request (3).



I: Idle, R: Read from transfer source, W: Write to transfer destination

Figure 14-7 Transfer Request when using two channels

[Note]

- **The DMA transfer has priority orders (CPU > DMA channel 0 > DMA channel 1) and limitation of the interval time for the periodical transfers.**

14.3.4 UART Continuous Transmission Using DMA Transfer

The following flow chart describes an example of UART continuous transmission using DMA transfer.
See Chapter 11 "Serial Communication Unit" for details of UART.
[Operation specifications]

- UART transmission of 15 consecutive bytes using DMA channel 1
- Use the serial communication unit 0 UART transmission DMA request for a transfer request
- Transfer data in address 0xEFED to 0xEFFB of RAM to SD0BUFH of serial communication unit 0.
- Transfer format: full-duplex communication mode, 115200 bps, 8-bit length, no parity, 1 stop bit

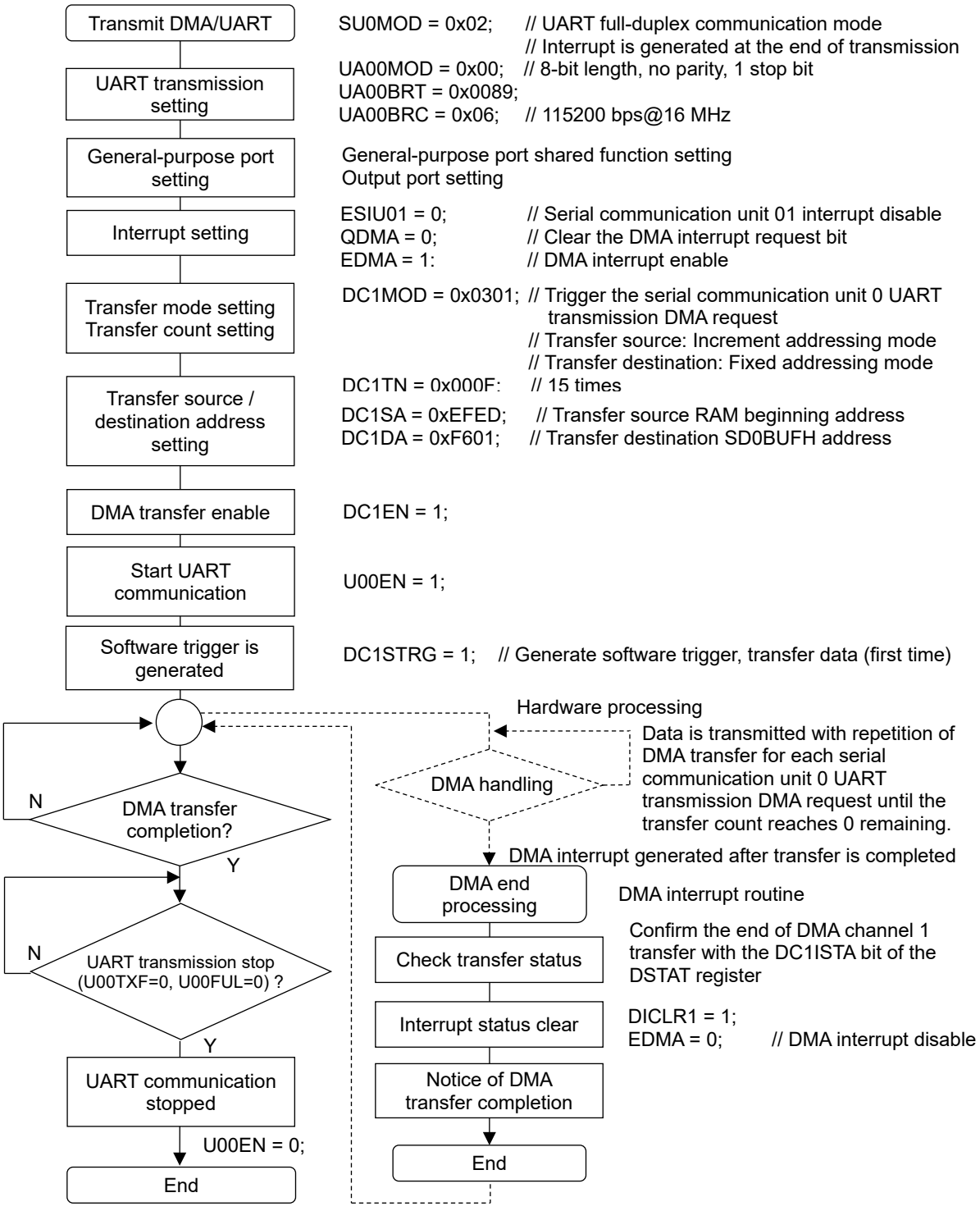


Figure 14-8 UART Continuous Transmission Flow Using DMA Transfer

14.3.5 UART Continuous Reception Using DMA Transfer

The following flow chart describes an example of UART continuous reception using DMA transfer.
See Chapter 11 "Serial Communication Unit" for details of UART.
[Operation specifications]

- UART reception of 15 consecutive bytes using DMA channel 0
- Use the serial communication unit 0 reception DMA request for a transfer request
- Transfer data from SD0BUFL of serial communication unit 0 to RAM address 0xEFED through 0xEFFB.
- Transfer format: full-duplex communication mode, 115200 bps, 8-bit length, no parity, 1 stop bit

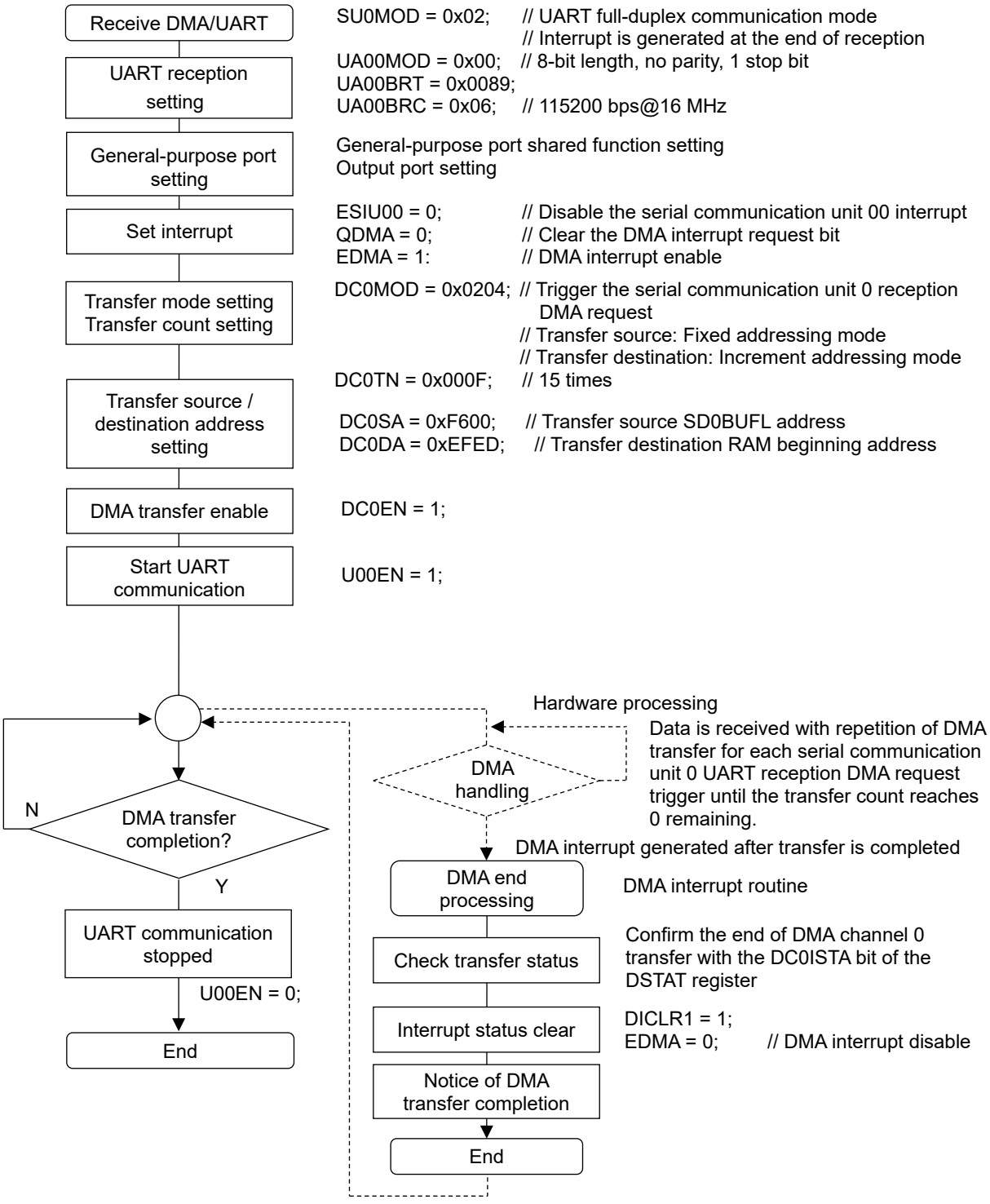


Figure 14-8 UART Continuous Reception Flow Using DMA Transfer

14.3.6 DMA Transfer Target Block

The following table shows function blocks available for the DMA transfer source/destination.

Table 14-1 DMA Transfer Target Block

	Function block																										
	Memory				SFR																						
	RAM	Program code area	Code option area	Data flash area	Multiplier/Divider	Reset Functions	Power management	Interrupt	Clock Generation Circuit	Low-speed time base counter	16-bit timer	Functional timer	Watchdog timer	Serial communication unit	I²C bus unit	I²C bus master	DMA controller	Buzzer	General-purpose port	External interrupt	CRC calculator	Analogue comparator	D/A Converter	Voltage level monitoring (VLS)	Successive approximation type A/D converter	LCD driver	Safety function	Simplified RTC
Transfer source	●	-	-	-	-	-	-	-	-	-	●	●	-	●	-	-	-	-	●	-	-	-	●	-	●	-	-	-
Transfer destination	●	-	-	-	-	-	-	-	-	-	●	●	-	●	-	-	-	-	●	-	●	-	●	-	●	●	-	-

●: Available
-: Unavailable

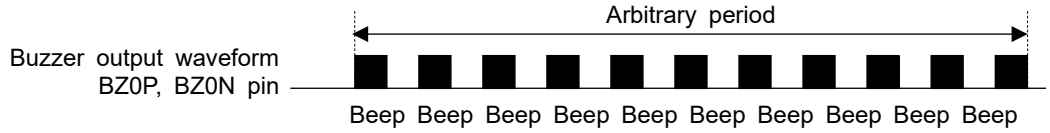
Chapter 15 Buzzer

15. Buzzer

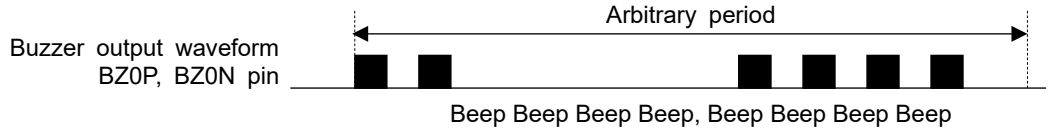
15.1 General Description

The buzzer circuit generates a base signal in combination of 8 frequencies and 15 duties and outputs the signal in four modes.

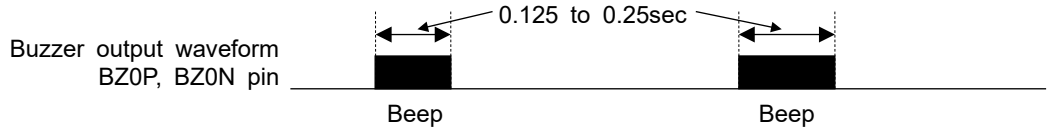
Intermittent sound 1 mode:



Intermittent sound 2 mode:



Single sound mode:



Continuous sound:

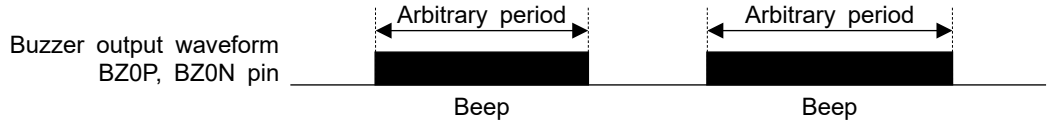


Figure 15-1 Buzzer output image

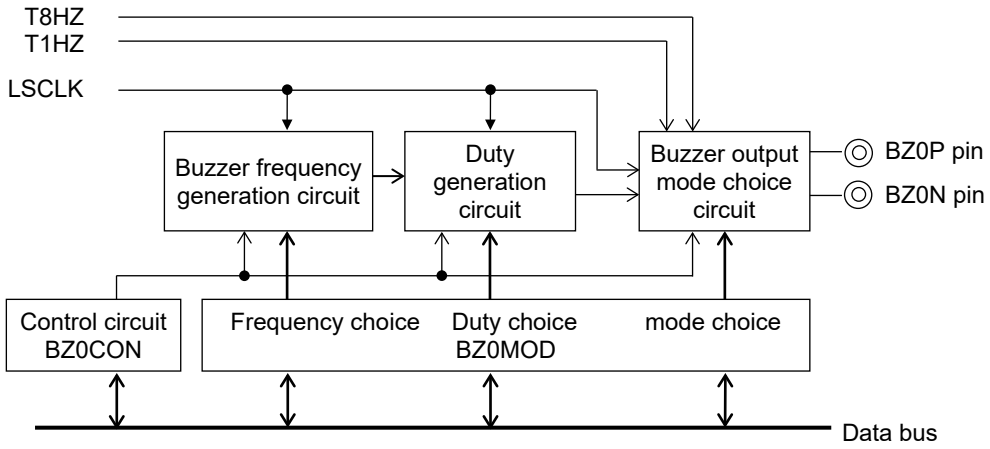
The buzzer circuit outputs a positive phase pulse (BZ0P) and negative phase pulse (BZ0N).
Also, for details of the clock used in the buzzer block (T8HZ, T1HZ), see Chapter 7 "Low Speed Time Base Counter".

15.1.1 Features

- Four types of buzzer mode (Intermittent sound 1, Intermittent sound 2, Single sound and Continuous sound)
- Eight types of frequency (4.096 kHz to 293 Hz)
- 15 duties (1/16 to 15/16 = 6.25% to 93.75%)
  Only seven duties (1/8 to 7/8) are available when the buzzer frequency is 4.096 kHz.
- The initial level (positive logic or negative logic) of the buzzer output pins can be chosen

15.1.2 Configuration

Figure 15-2 shows the configuration of the buzzer circuit.



BZ0CON : Buzzer 0 control register
BZ0MOD : Buzzer 0 mode register

Figure 15-2 Configuration of Buzzer

15.1.3 List of Pins

The output pins of the buzzer signal are assigned to the shared function of the general port.

Pin name	I/O	Function
BZ0P	O	Buzzer output (positive phase)
BZ0N	O	Buzzer output (negative phase)

Table 15-1 shows the list of the general ports used for the buzzer output and the register settings of the ports.

Table 15-1 Ports used for the buzzer and the register settings

Pin name	Shared pin		Setting register	Setting value
BZ0P	P17	7th Function	P1MOD7	0110_XXXX
	P26	7th Function	P2MOD6	0110_XXXX
BZ0N	P20	7th Function	P2MOD0	0110_XXXX
	P27	7th Function	P2MOD7	0110_XXXX

"XXXX" determines the condition of the port output

XXXX	Condition of the port output
0010	CMOS output
1010	Nch open drain output (without the pull-up)
1111	Nch open drain output (with the pull-up)

BZ0P pin and BZ0N pin are assigned to two general ports.
Be sure to use the ports in following combinations.

Output pin	Combination 1	Combination 2
BZ0P	P26	P17
BZ0N	P27	P20

15.2 Description of Registers

15.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF0C0	Buzzer 0 control register	BZ0CON	-	R/W	8	0x00
0xF0C1	Reserved	-	-	-	-	-
0xF0C2	Buzzer 0 mode register	BZ0MODL	BZ0MOD	R/W	8/16	0x00
0xF0C3		BZ0MODH		R/W	8	0x00

15.2.2 Buzzer 0 Control Register (BZ0CON)

BZ0CON is a special function register (SFR) used to control the buzzer.

Address: 0xF0C0 (BZ0CON)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								BZ0CON							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BZ0RUN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	BZ0RUN	This bit used to start or stop the buzzer output. When the single sound output mode is chosen by BZ0MD to BZ0MD0 bits of BZ0MOD register, the buzzer output automatically stops and the BZ0RUN bit is automatically reset to "0". 0: Stop buzzer output (initial value) 1: Start buzzer output

15.2.3 Buzzer 0 Mode Register (BZ0MOD)

BZ0MOD is a special function register (SFR) used to set the buzzer output waveform.

Address: 0xF0C2 (BZ0MOD/ BZ0MODL), 0xF0C3(BZ0MODH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BZ0MOD															
Byte	BZ0MODH								BZ0MODL							
Bit	-	-	-	BZ0INI	BZ0D3	BZ0D2	BZ0D1	BZ0D0	-	BZ0F2	BZ0F1	BZ0F0	-	-	BZ0MD1	BZ0MD0
R/W	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

Bit No.	Bit symbol name	Description
15 to 13	-	Reserved bits
12	BZ0INI	This bit is used to choose the initial logic of the buzzer output pins. 0: The output of buzzer pin (BZ0P, BZ0N) has positive logic (Initial) 1: The output of buzzer pin (BZ0P, BZ0N) has negative logic
11 to 8	BZ0D3 to BZ0D0	These bits are used to choose duty of the buzzer output. The volume of the buzzer sound changes by changing the duty. 0000: Buzzer Frequency 4.096kHz: Duty is 1/8 (12.5%) (Initial); Buzzer Frequency other than 4.096kHz: Duty is 1/16 (6.25%) (Initial) 0001: Duty is 1/8 (12.5%); Duty is 1/16(6.25%) 0010: Duty is 1/8 (12.5%); Duty is 2/16(12.5%) 0011: Duty is 1/8 (12.5%); Duty is 3/16(18.75%) 0100: Duty is 2/8 (25%); Duty is 4/16(25%) 0101: Duty is 2/8 (25%); Duty is 5/16 (31.25%) 0110: Duty is 3/8 (37.5%); Duty is 6/16 (37.5%) 0111: Duty is 3/8 (37.5%); Duty is 7/16 (43.75%) 1000: Duty is 4/8 (50%); Duty is 8/16 (50%) 1001: Duty is 4/8 (50%); Duty is 9/16 (56.25%) 1010: Duty is 5/8 (62.5%); Duty is 10/16 (62.5%) 1011: Duty is 5/8 (62.5%); Duty is 11/16 (68.75%) 1100: Duty is 6/8 (75%); Duty is 12/16 (75%) 1101: Duty is 6/8 (75%); Duty is 13/16 (81.25%) 1110: Duty is 7/8 (87.5%); Duty is 14/16 (87.5%) 1111: Duty is 7/8 (87.5%); Duty is 15/16 (93.75%) small ▲ Volume ▼ large
7	-	Reserved bit
6 to 4	BZ0F2 to BZ0F0	These bits are used to choose the frequency of the buzzer output. The pitch of the buzzer sound changes by changing the frequency. 000: 4.096kHz (Initial) 001: 2.048kHz 010: 1.024kHz 011: 683Hz 100: 512Hz 101: 410Hz 110: 341Hz 111: 293Hz High ▲ Pitch ▼ Low
3, 2	-	Reserved bits

Bit No.	Bit symbol name	Description
1, 0	BZ0MD1, BZ0MD0	These bits are used to choose the buzzer output mode. 00: Intermittent sound 1 output mode (initial value) Sound like: "Beep Beep Beep Beep Beep Beep Beep Beep" 01: Intermittent sound 2 output mode Sound like: "Beep Beep Beep Beep" 10: Single sound mode Sound like: "Beep" (0.125 to 0.25sec) 11: Continuous sound mode Sound like: "Beep" (arbitrary period) For details of the waveforms of the buzzer output, see Section 15.3 "Description of Operation".

15.3 Description of Operation

15.3.1 Intermittent Sound 1 Mode

15.3.1.1 Operation of Intermittent Sound 1 Mode

Figure 15-3 shows the buzzer output waveform of the intermittent sound 1 mode.
Each of black areas indicates a period of time during which the buzzer signal pulse is output.
See Section 15.3.5.1 "Buzzer Output Start and Stop Timing" for details of pulse signal timing.

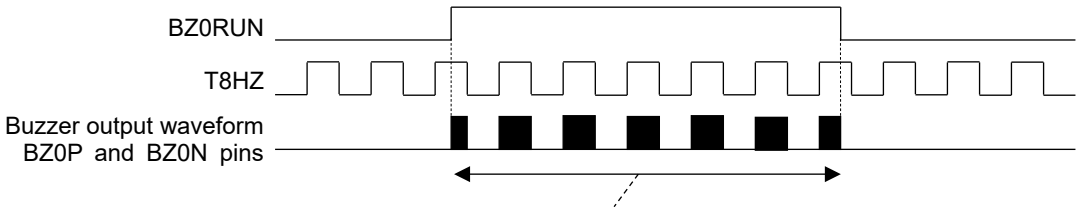


Figure 15-3 Output Waveforms in Intermittent Sound 1 Mode
(Sound like: "Beep Beep Beep Beep Beep Beep Beep Beep" (continues for an arbitrary period of time))

15.3.1.2 Example of Intermittent Sound 1 Mode Setting Procedure

Figure 15-4 shows an example of the intermittent sound 1 mode setting procedure.

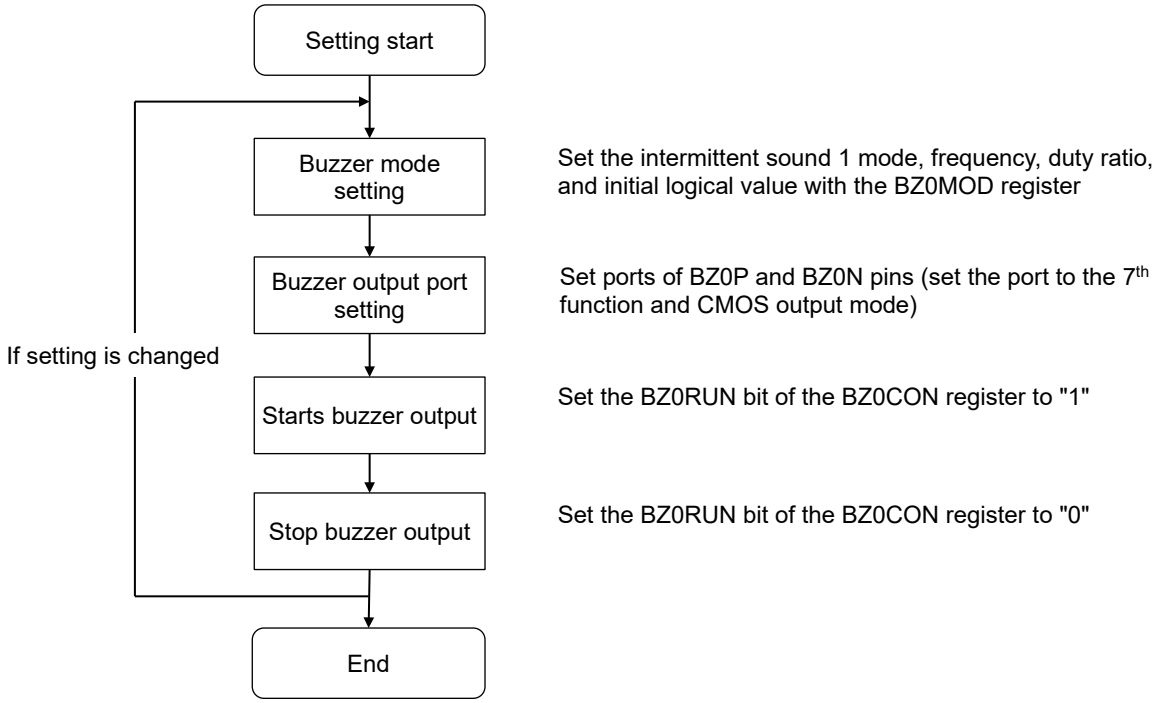


Figure 15-4 Example of Intermittent Sound 1 Mode Setting Procedure

[Note]

- **The buzzer output may be started or stopped in the middle of the buzzer waveform depending on timing of setting the BZ0RUN bit of the BZ0CON register. If it causes a problem, take one of the following measure A or measure B:**
  - **Measure A: Use the low-speed time base counter interrupt (choose T8HZ or T1HZ for signal assignment).**
  - **Measure B: Use the LTBR register to synchronize the falling edge of the T8HZ or T1HZ signal with the timing BZ0RUN is set.**

15.3.2 Intermittent Sound 2 Mode

15.3.2.1 Operation of Intermittent Sound 2 Mode

Figure 15-5 shows the buzzer output waveform of the intermittent sound 2 mode.
Each of black areas indicates a period of time during which the buzzer signal pulse is output.
See Section 15.3.5.1 "Buzzer Output Start and Stop Timing" for details of pulse signal timing.

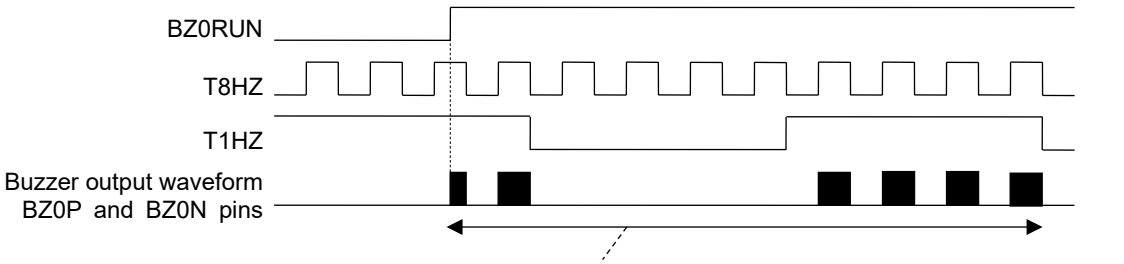


Figure 15-5 Output Waveforms in Intermittent Sound 2 Mode
(Sound like: "Beep Beep Beep Beep, Beep Beep Beep Beep" (continues for an arbitrary period of time))

15.3.2.2 Example of Intermittent Sound 2 Mode Setting Procedure

Figure 15-6 shows an example of the intermittent sound 2 mode setting procedure.

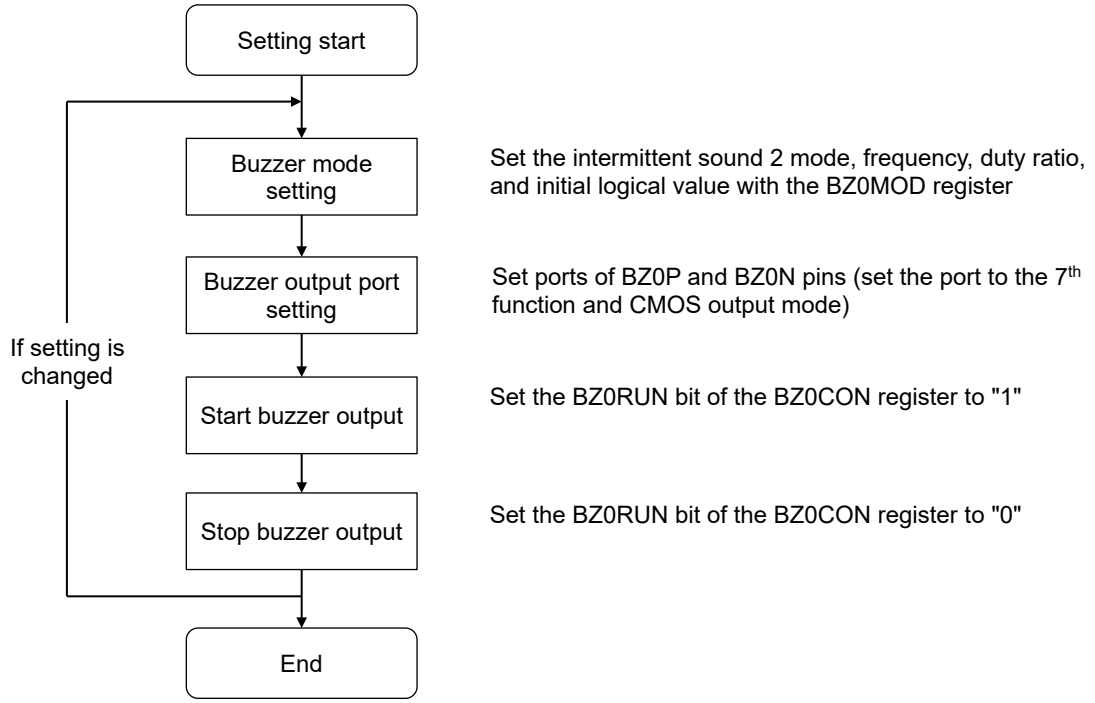


Figure 15-6 Example of Intermittent Sound 2 Mode Setting Procedure

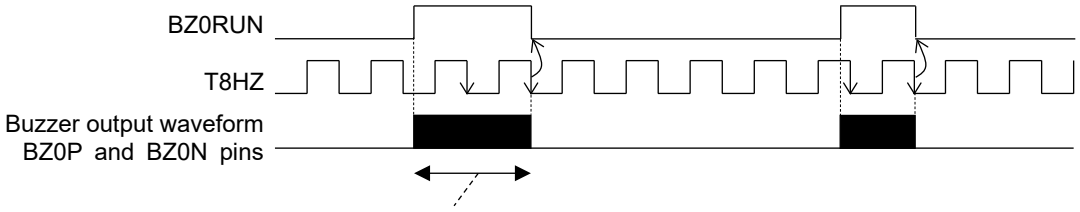
[Note]

- **The buzzer output may be started or stopped in the middle of the buzzer waveform depending on timing of setting the BZ0RUN bit of the BZ0CON register. If it causes a problem, take one of the following measure A or measure B:**
  - **Measure A: Use the low-speed time base counter interrupt (choose T8HZ or T1HZ for signal assignment).**
  - **Measure B: Use the LTBR register to synchronize the falling edge of the T8HZ or T1HZ signal with the timing BZ0RUN is set.**

15.3.3 Single Sound Mode

15.3.3.1 Single Sound Mode Operation

Figure 15-7 shows the buzzer output waveform of the single sound mode.
Each of black areas indicates a period of time during which the buzzer signal pulse is output.
See Section 15.3.5.1 "Buzzer Output Start and Stop Timing" for details of pulse signal timing.



A buzzer sound is output until the falling edge of T8HZ clock is detected twice after the BZ0RUN bit of the BZ0CON register is set to "1".

Figure 15-7 Output Waveforms in Single Sound Mode (Sound like: "Beep" (for 0.125 to 0.25 seconds))

15.3.3.2 Example of Single Sound Mode Setting Procedure

Figure 15-8 shows an example of the single sound mode setting procedure.

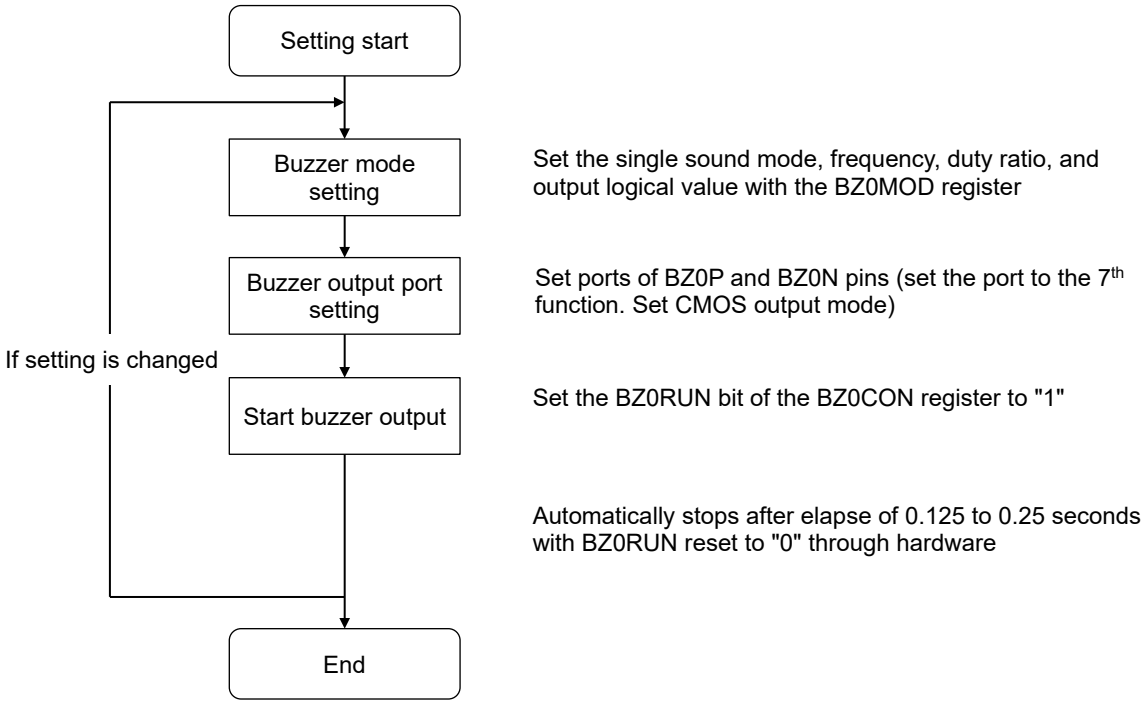


Figure 15-8 Example of Single Sound Mode Setting Procedure

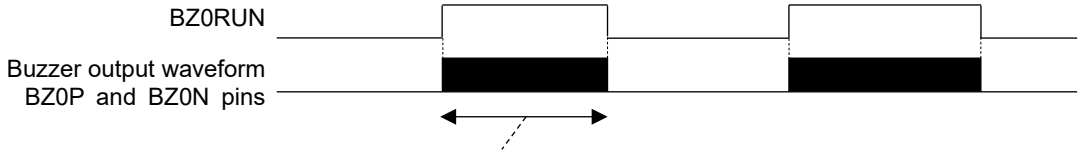
[Note]

- **The buzzer output may be started in the middle of the buzzer waveform depending on timing the BZ0RUN bit of the BZ0CON register is set to "1". If it causes a problem, take one of the following measures: A or B described below.**
  - **Measure A: Use the low-speed time base counter interrupt (choose T8HZ for signal assignment).**
  - **Measure B: Use the LTBR register to synchronize the falling edge of the T8HZ signal with the timing at which BZ0RUN is set to "1".**

15.3.4 Continuous Sound Mode

15.3.4.1 Continuous Sound Mode Operation

Figure 15-9 shows the buzzer output waveform of the continuous sound mode.
Each of black areas indicates a period of time during which the buzzer signal pulse is output.
See Section 15.3.5.1 "Buzzer Output Start and Stop Timing" for details of pulse signal timing.



A buzzer sound is output until the BZ0RUN bit of the BZ0CON register is cleared to "0" after the BZ0RUN bit is set to "1".

Figure 15-9 Output Waveforms in Continuous Sound Mode
(Sound like: "Beep" (continues for an arbitrary period of time))

15.3.4.2 Example of Continuous Sound Mode Setting Procedure

Figure 15-10 shows an example of the continuous sound mode setting procedure.

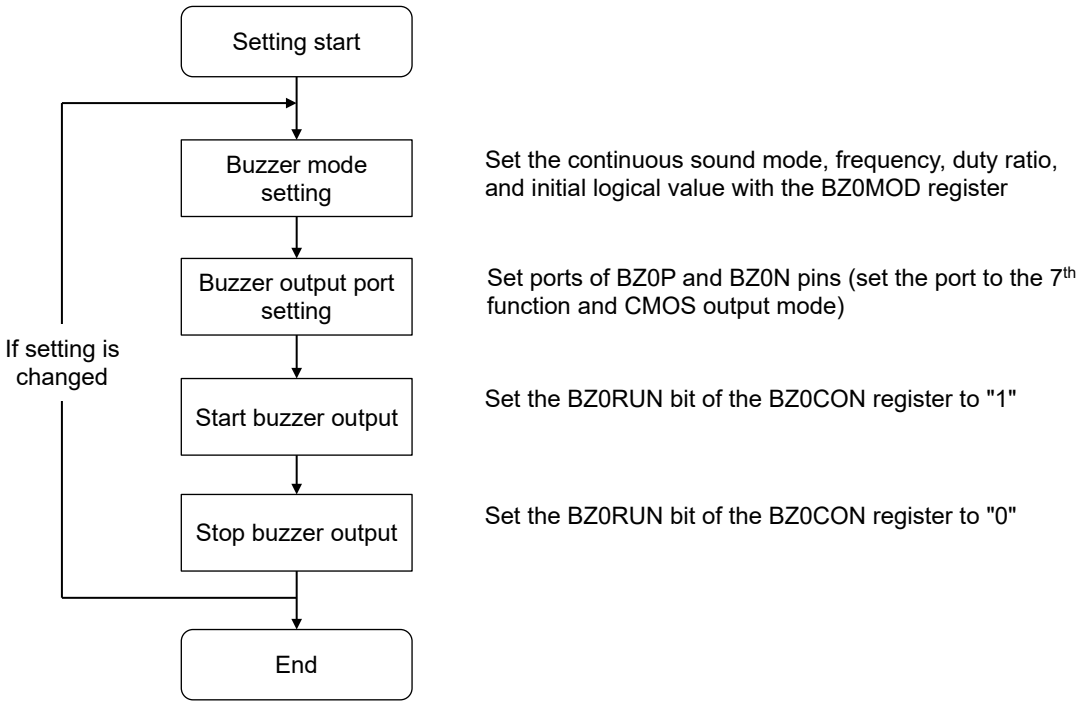
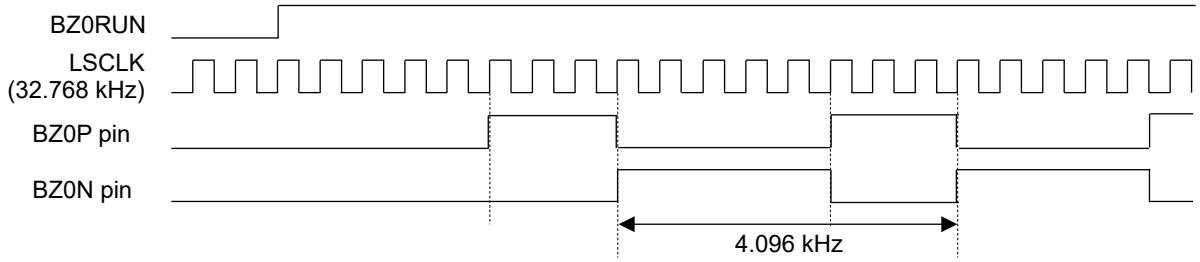


Figure 15-10 Example of Continuous Sound Mode Setting Procedure

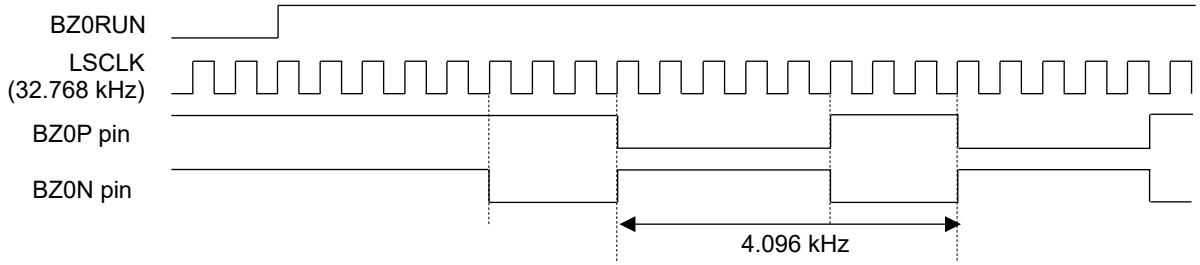
15.3.5 Common Operation

15.3.5.1 Buzzer Output Start and Stop Timing

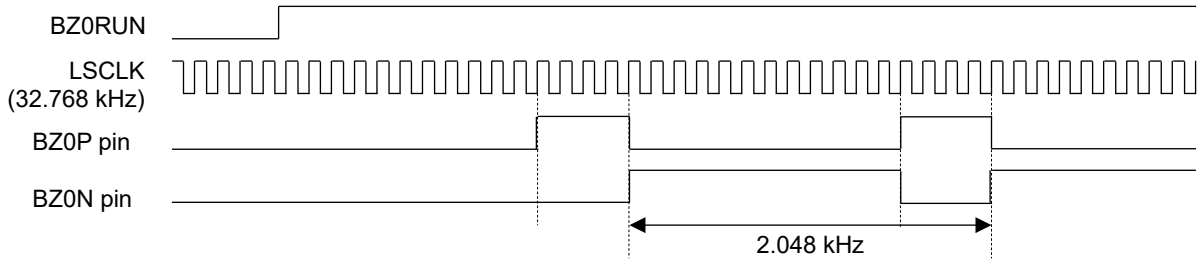
Figures 15-11 to 15-15 show buzzer signal output start and stop timing waveforms.



(1) When positive logic is applied (BZ0INI bit of BZ0MOD register=0)
Setting example: buzzer frequency=4.096 kHz, duty=3/8 (37.5%)



(2) When negative logic is applied (BZ0INI bit of BZ0MOD register=1)
Setting example: buzzer frequency=4.096 kHz, duty=3/8 (37.5%)

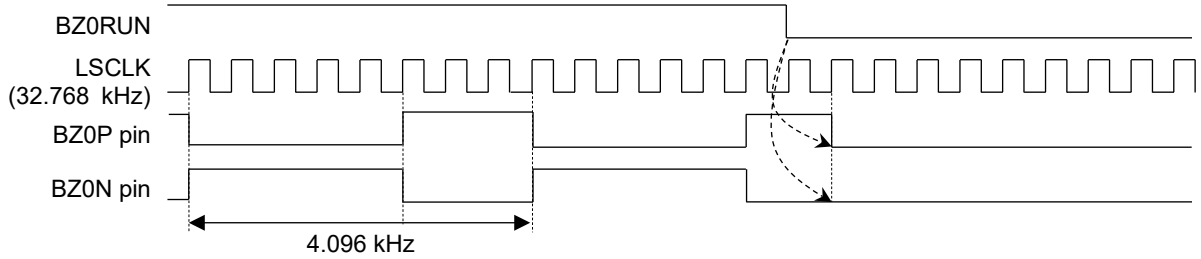


(3) When positive logic is applied (BZ0INI bit of BZ0MOD register=0)
Setting example: buzzer frequency=2.048 kHz, duty=4/16 (25%)

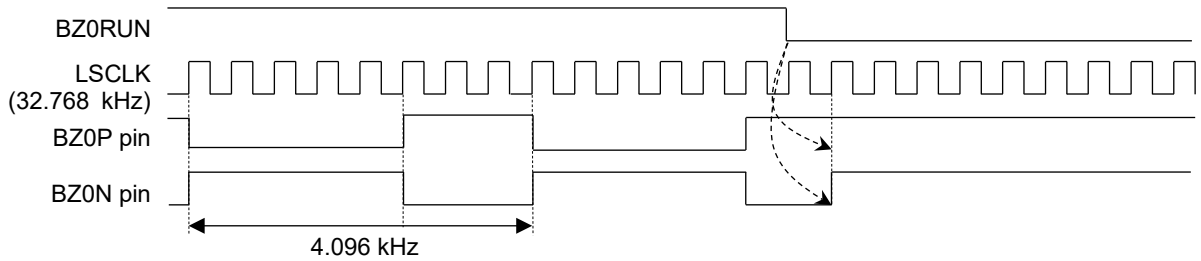
Figure 15-11 Buzzer Output Start Timing Controlled by BZ0RUN Bit of BZ0CON Register

[Note]

- **An error to a maximum of one clock of the low-speed clock (LSCLK) occurs by the time the buzzer output is started after writing "1" to the BZ0RUN bit of the BZ0CON register.**



(1) When positive logic is applied (BZ0INI bit of BZ0MOD register=0)
Setting example: buzzer frequency=4.096 kHz, duty=3/8 (37.5%)

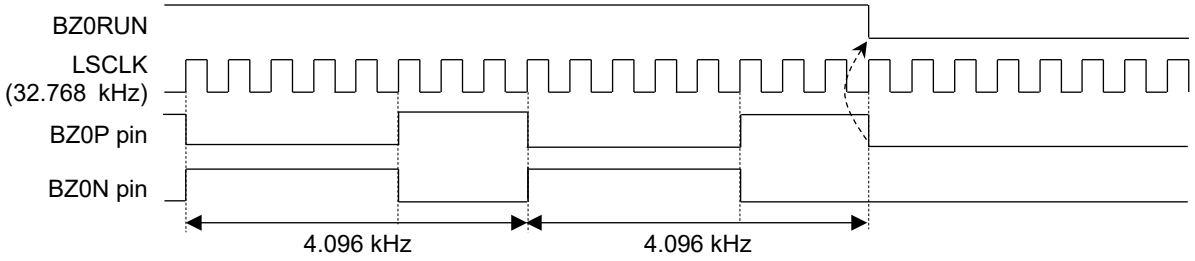


(2) When negative logic is applied (BZ0INI bit of BZ0MOD register=1)
Setting example: buzzer frequency=4.096 kHz, duty=3/8 (37.5%)

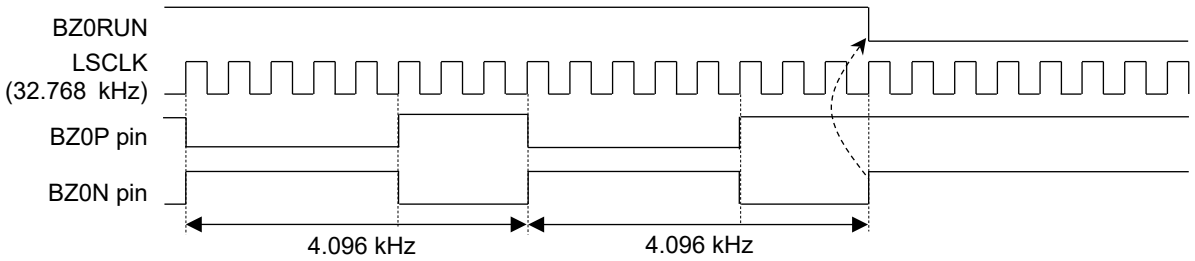
Figure 15-12 Buzzer Output Stop Timing Controlled by BZ0RUN of BZ0CON Register

[Note]

- **An error to a maximum of one clock of the low-speed clock (LSCLK) occurs by the time the buzzer output is stopped after writing "0" to the BZ0RUN bit of the BZ0CON register.**



(1) When positive logic is applied (BZ0INI bit of BZ0MOD register=0)
Setting example: buzzer frequency=4.096 kHz, duty=3/8 (37.5%)

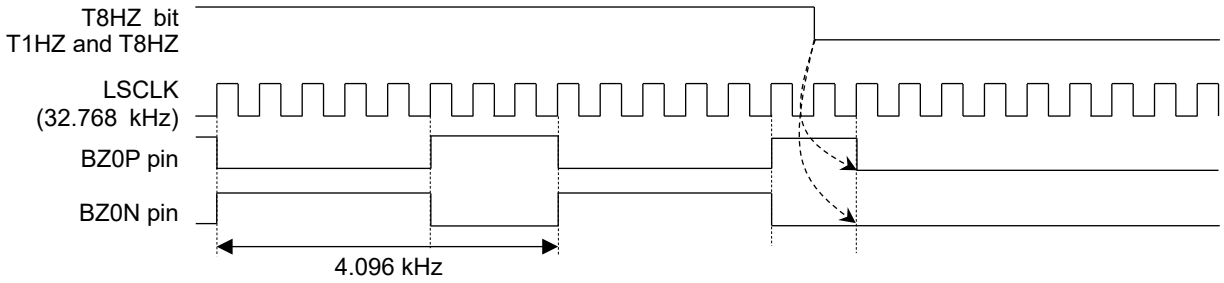


(2) When negative logic is applied (BZ0INI bit of BZ0MOD register=1)
Setting example: buzzer frequency=4.096 kHz, duty=3/8 (37.5%)

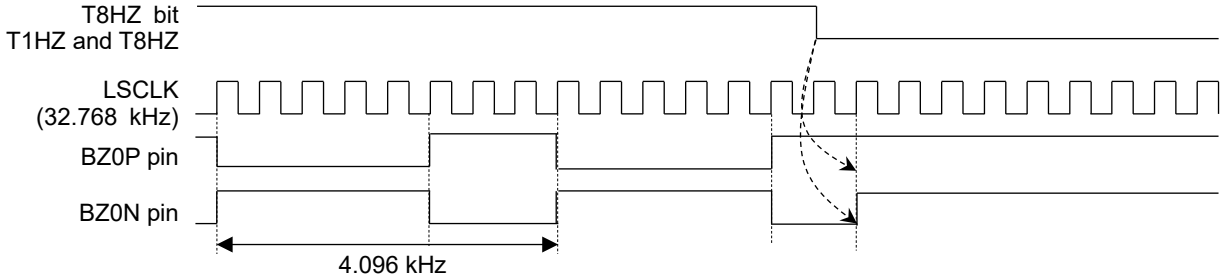
Figure 15-13 Buzzer Output Automatic Stop Timing in Single Sound Mode

[Note]

- **In the single sound mode, the BZ0RUN bit of the BZ0CON register is cleared to "0" when the single sound buzzer output is ended.**

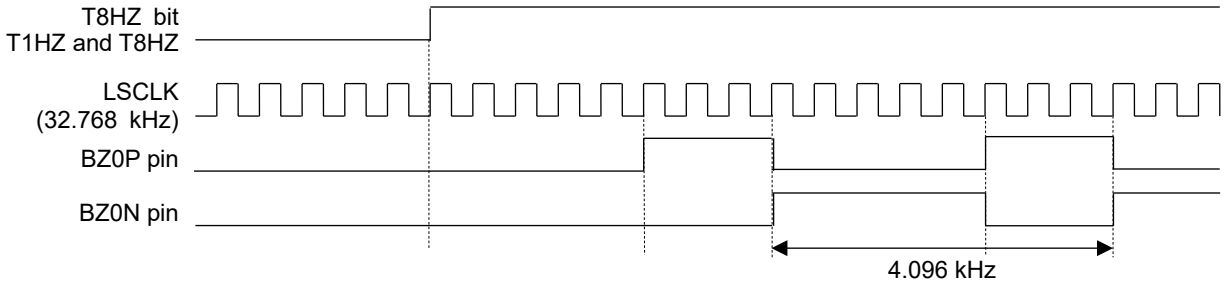


(1) When positive logic is applied (BZ0INI bit of BZ0MOD register=0)
Setting example: buzzer frequency=4.096 kHz, duty=3/8 (37.5%)

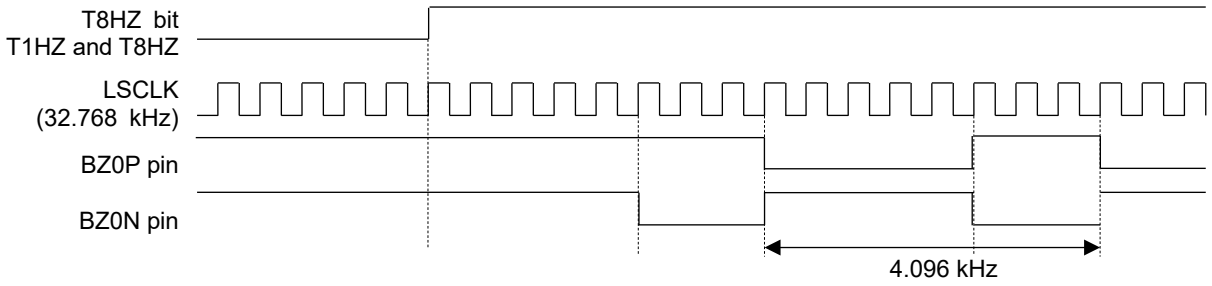


(2) When negative logic is applied (BZ0INI bit of BZ0MOD register=1)
Setting example: buzzer frequency=4.096 kHz, duty=3/8 (37.5%)

Figure 15-14 Buzzer Output Pause Timing in Intermittent Sound 1 Mode and Intermittent Sound 2 Mode



(1) When positive logic is applied (BZ0INI bit of BZ0MOD register=0)
Setting example: buzzer frequency=4.096 kHz, duty=3/8 (37.5%)



(2) When negative logic is applied (BZ0INI bit of BZ0MOD register=1)
Setting example: buzzer frequency=4.096 kHz, duty=3/8 (37.5%)

Figure 15-15 Restart Timing from Buzzer Output Pause in Intermittent Sound 1 Mode or Intermittent Sound 2 Mode

[Note]

- **In the intermittent sound 1 or 2 mode, an error to a maximum of one clock of the low-speed clock (LSCLK) occurs by the time the buzzer output is started after the T8HZ signal became "1".**

Chapter 16 Simplified RTC

16. Simplified RTC

16.1 General Description

ML62Q1000 series has the simplified RTC (RTC: Real Time Clock).
The simplified RTC counts up from 00 minutes 00 seconds to 59 minutes 59 seconds in the unit of one second and also generates an interrupt request periodically.
For the interrupt enable/request flags, etc. described in this chapter, refer to Chapter 5 "Interrupts".

Table 16-1 shows the availability of the Simplified RTC.

Table16-1 Availability of RTC in ML62Q1000 series

Item	ML62Q1300 group	ML62Q1500/ ML62Q1800 group	ML62Q1700 group
Availability of the Simplified RTC	-	●	●

●: Available - : Unavailable

16.1.1 Features

- A desired periodical interrupt request can be chosen from among four types (0.5, 1, 30 and 60 seconds).
- A function to prevent erroneous writing to the simplified RTC minute/second counter included.
- The simplified RTC minute/second counter continues counting operation even when a reset (other than power-on reset) is generated.
- Counting operation is continued all the time, except when operating in the STOP/STOP-D mode.

16.1.2 Configuration

Figure 16-1 shows the configuration of simplified RTC.

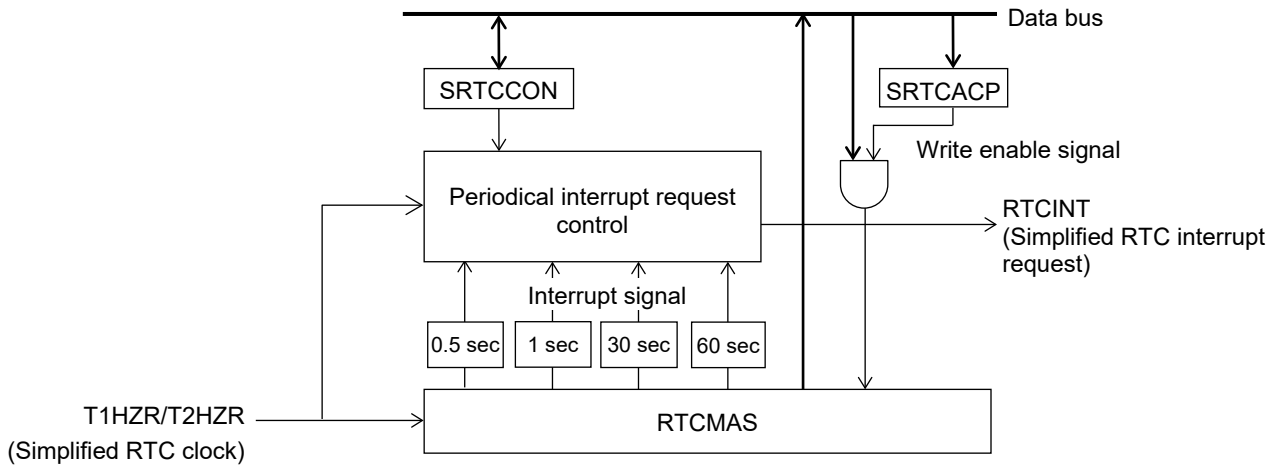


Figure 16-1 Simplified RTC Configuration

SRTCACP : Simplified RTC acceptor
SRTCMAS : Simplified RTC minute/second counter
SRTCMIN (minute counter), SRTCSEC (second counter)
SRTCCON : Simplified RTC control register

16.2 Description of Registers

16.2.1 List of Registers

Address	Name	Symbol name		R/W	Size	Initial value
		Byte	Word			
0xF0C8	Simplified RTC acceptor	SRTCACP	-	W	8	0x00
0xF0C9	Reserved	-	-	-	-	-
0xF0CA	Simplified RTC minute/second counter	SRTCSEC	SRTCMAS	R/W	8/16	0x00
0xF0CB		SRTCMIN		R/W	8	0x80
0xF0CC	Simplified RTC control register	SRTCCON	-	R/W	8	0x00
0xF0CD	Reserved	-	-	-	-	-

[Note]
- **SRTCMAS is reset only by the power-on reset.**

16.2.2 Simplified RTC Acceptor (SRTCACP)

This register is a write-only special function register (SFR) used to enable writing to the simplified RTC minute/second counter (SRTCMAS).
"0x00" is read for reading the SRTCACP register.
When "0x3C" and "0xC3" are written to the SRTCACP register in this order, writing to the SRTCMAS register is enabled only once.

Address: 0xF0C8 (SRTCACP)
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	-															
Byte	-								SRTCACP							
Bit	-	-	-	-	-	-	-	-	d7	d6	d5	d4	d3	d2	d1	d0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]
- **After writing "0x3C" to SRTCACP, if data other than "0x3C" or "0xC3" is written to SRTCACP, writing of "0x3C" becomes invalid.**
- **When writing "0x3C" and "0xC3" to SRTCACP in this order, and writing a value other than "0xC3" to SRTCACP with writing to SRTCMAS enabled, writing to SRTCMAS becomes invalid.**

16.2.3 Simplified RTC Minute/Second Counter (SRTCMAS)

This register is a special function register (SFR) used to show the minute/second data.
After enabling to write to the SRTCMAS register using the SRTCACP register, data can be written to the SRTCMAS register.
The SRTCMAS register is initialized only by the power-on reset.

Address: 0xF0CA (SRTCSEC/SRTCMAS), 0xF0CB (SRTCMIN)
Access: R/W
Access size: 8/16 bits
Initial value: 0x8000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SRTCMAS															
Byte	SRTCMIN								SRTCSEC							
Bit	POR STAT	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1	-	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	PORSTAT	This bit is used to indicate the simplified RTC operating status. The initial value of PORSTAT bit is "1". To write data to the SRTCMAS register, write "0" to the PORSTAT bit. When "1" is written to the PORSTAT bit, the initial value is written to the SRTCMAS register. 0: Normal operation 1: After power-on reset, no data has been written to the SRTCMAS register, or data outside the configurable range has been written to the SRTCMAS register
14 to 8	MIN40 to MIN1	These bits are used to store the minute data. Count up as soon as the second counter (SRTCSEC) changes from 59 to 00. The SRTCMIN register indicates 0 to 59 minutes as a decimal number. MIN40 to MIN10 bits indicate a tens place, whereas MIN8 to MIN1 bits indicate an ones place. MIN40 to MIN10 bits can be written within the range of 0 to 5, whereas MIN8 to MIN1 bits can be written within the range of 0 to 9.
7	-	Reserved bit
6 to 0	SEC40 to SEC1	These bits are used to store the second data. Count up at the falling edge of the T1HZR signal. The SRTCSEC register indicates 0 to 59 minutes as a decimal number. SEC40 to SEC10 bits indicate a tens place, whereas SEC8 to SEC1 bits indicate an ones place. SEC40 to SEC10 bits can be written within the range of 0 to 5, whereas SEC8 to SEC1 bits can be written within the range of 0 to 9.

[Note]

- **When reading the SRTCMAS register, read it twice and check that the two values coincide with each other to prevent reading of undefined data during counting up.**
- **If the data outside the range from 00 minutes 00 seconds to 59 minutes 59 seconds is written to the SRTCMAS register, the register will be set to the initial value.**
- **An interrupt request may be generated immediately after writing depending on the timing of writing data to the SRTCMAS register. To prevent an interrupt request from being generated while writing time data, disable RTCINT using the simplified RTC control register (SRTCCON) before writing to the SRTCMAS register.**
- **It is recommended that data is written to the SRTCMAS register with word access.**
- **After enabling the write operation using the SRTCACP register, data can be written to the SRTCMAS register only once regardless of using byte or word access. If writing twice using 8-bit access after the write operation is enabled, the second writing is ignored.**
- **If 0 second (0x00) is written to the second counter (SRTCSEC) when it is 59 second (0x59), the minute counter (SRTCMIN) counts up. However, if the minute counter is also written at the same time using 16-bit access, then it does not count up and the written value becomes valid.**

16.2.4 Simplified RTC Control Register (SRTCCON)

This register is a special function register (SFR) used to set a periodical interrupt request.

Address: 0xF0CC (SRTCCON)
Access: R/W
Access size: 8 bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SRTCCON							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	RINT2	RINT1	RINT0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 3	-	Reserved bits
2 to 0	RINT2 to RINT0	Bits to set a periodical interrupt request. 000: Periodical interrupt request disabled (initial value) 001: 0.5-second interrupt request 010: 1-second interrupt request 011: 30-second interrupt request Generate an interrupt request when SRTCSEC reaches 30 seconds and 60 seconds (changes from 29 to 30 seconds, and 59 to 00 seconds). 100: 60-second interrupt request Generate an interrupt request when SRTCSEC reaches 60 seconds (changes from 59 to 00 seconds). Others: 60-second interrupt request

16.3 Description of Operation

The simplified RTC starts operation after a power-on reset is released. Since the value of the PORSTAT bit of the SRTCMAS register is "1" after the power-on reset is released, data needs to be written to the SRTCMAS register to set the minute and second. See Figure 16-3 "Simplified RTC Setting Example for Writing Time Data" for the procedure of writing to the SRTCMAS register.
The SRTCMAS register is initialized only by a power-on reset. The counting operation is continued when any other system resets occur.
Table 16-2 shows a list of count values of each counter.

Table 16-2 Count Value of Each Counter

Counter name		Count value
Second counter	SRTCSEC40,20,10	0x0 to 0x5
	SRTCSEC4,2,1	0x0 to 0x9
Minute counter	SRTCMIN40,20,10	0x0 to 0x5
	SRTCMIN4,2,1	0x0 to 0x9

16.3.1 Simplified RTC Time Data Writing Operation

Figure 16-2 shows the timing chart when data is written to the SRTCMAS register.
Writing data to the write-enabled SRTCMAS register causes T1HZR/T2HZR of the time base counter to be initialized.

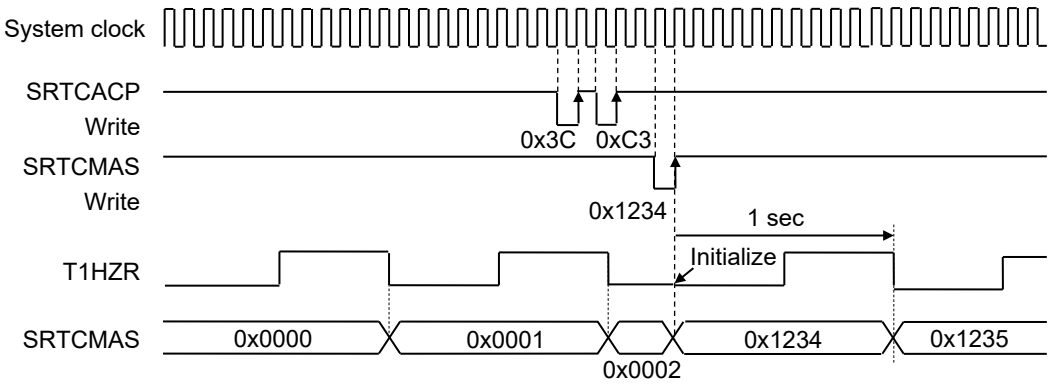


Figure 16-2 Timing Chart When Data is Written to SRTCMAS Register

16.3.2 Simplified RTC Setting Example for Writing Time Data

Figure 16-3 shows an example of setting the simplified RTC to write 29 minutes 39 seconds to the SRTCMAS register.

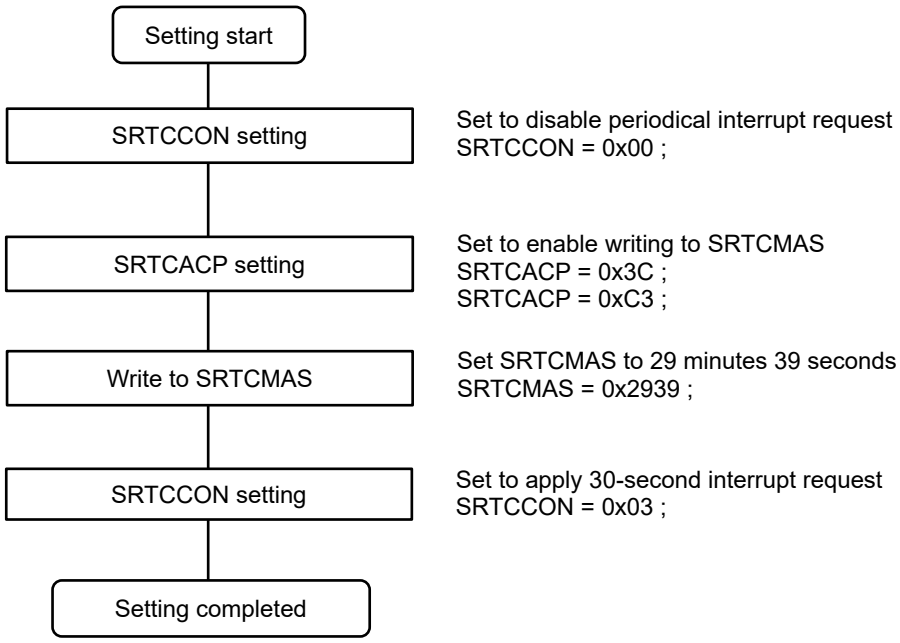


Figure 16-3 Simplified RTC Setting Example for Writing Time Data

Chapter 17 GPIO

17. General Purpose Port

17.1 General Description

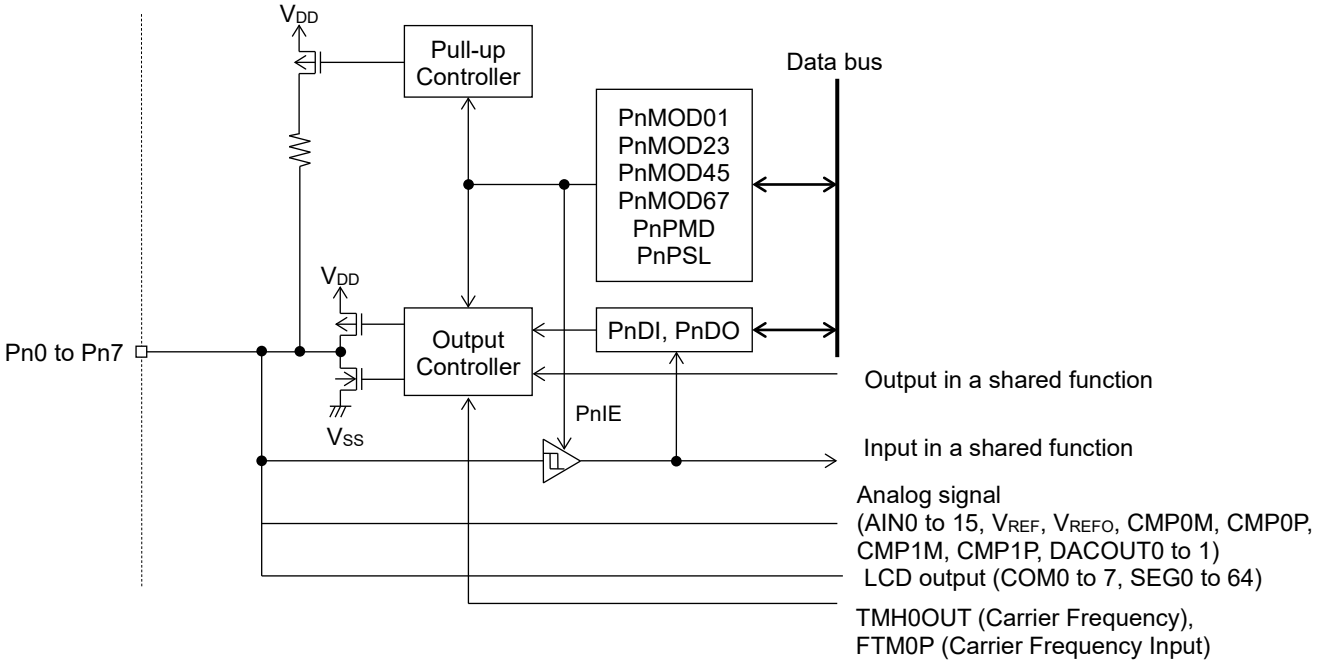
The general purpose port is used as an input port or an output port.
The input and output is switchable on each pin. Max. 8 pins are available to read or to change the level of output in the same time. A general input port or output port shares a numbers of functions. See "1.3.2 List of Pins" or "1.3.3 Description of Pins"for more detail.
The general input/output ports can be used for external interrupts and external inputs for the functional timers. Also, used as an input or an output pins in shared functions by setting the port n mode register.
Two general input ports are shared with the crystal resonator connection pins.
The number of general port is dependent of each product. See Table 17-1 "List of Pins".

17.1.1 Features

- Input or output can be chosen in each pin
- Pull-up resistor can be chosen in each pin
- CMOS output or N-channel open drain output is can be chosen in each pin
- Direct driving LEDs is supported when the N-channel open drain output is chosen
- Carrier frequency output function
- Port output level test function

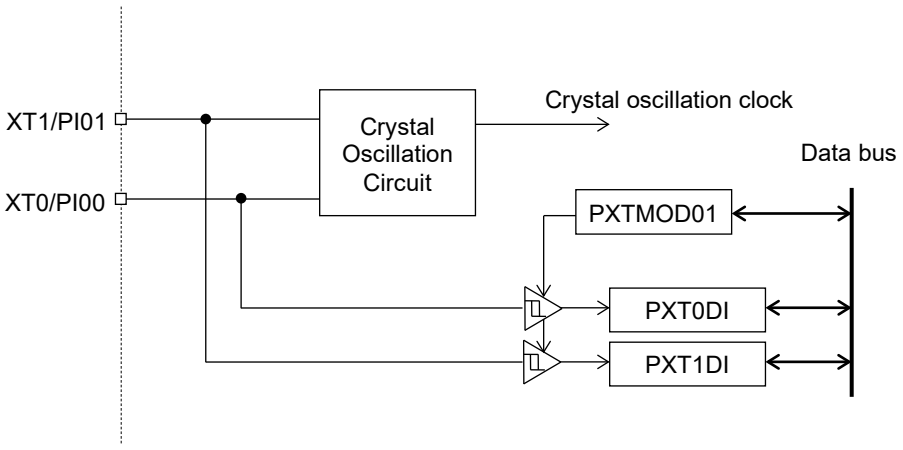
17.1.2 Configuration

Figure 17-1 shows the configuration of the general purpose port. See "17.2.1 List of registers" for available pins and registers.



PnDI : Port n data register(bit 7 to 0)
PnDO : Port n data register(bit 15 to 8)
PnMOD01 : Port n mode register 01
PnMOD23 : Port n mode register 23
PnMOD45 : Port n mode register 45
PnMOD67 : Port n mode register 67
PnPMD : Port n pulse mode register
PnPSL : Port n pulse selection register

Figure 17-1 Configuration of General Purpose I/O port n



PXT0DI : PORTXT data input register(bit 0)
PXT1DI : PORTXT data input register(bit 1)
PXTMOD01 : PORTXT mode register01

Figure 17-2 Configuration of General Purpose Input Port

17.1.3 List of Pins

Table 17-1 List of Pins

Pin Name	Primary Function	Available / Unavailable								
		ML62Q1300 group				ML62Q1500/1800/1700 group				
		16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
PI00	Crystal resonator connection pin	-	-	-	-	●	●	●	●	●
PI01	Crystal resonator connection pin	-	-	-	-	●	●	●	●	●
P00	I/O port	●	●	●	●	●	●	●	●	●
P01	I/O port / DACOUT0	-	-	●	●	●	●	●	●	●
P02	I/O port / EXI0	●	●	●	●	●	●	●	●	●
P03	I/O port / EXI1	●	●	●	●	●	●	●	●	●
P04	I/O port / EXI2	●	●	●	●	●	●	●	●	●
P05	I/O port	-	●	●	●	●	●	●	●	●
P06	I/O port	-	-	-	●	●	●	●	●	●
P07	I/O port	-	-	-	●	●	●	●	●	●
P10	I/O port	-	-	-	●	●	●	●	●	●
P11	I/O port	-	-	-	●	●	●	●	●	●
P12	I/O port	-	-	●	●	●	●	●	●	●
P13	I/O port	●	●	●	●	●	●	●	●	●
P14	I/O port	-	-	-	●	●	●	●	●	●
P15	I/O port	-	-	-	●	●	●	●	●	●
P16	I/O port	-	-	●	●	●	●	●	●	●
P17	I/O port / EXI3	●	●	●	●	●	●	●	●	●
P20	I/O port	●	●	●	●	●	●	●	●	●
P21	I/O port / EXI4	●	●	●	●	●	●	●	●	●
P22	I/O port	●	●	●	●	●	●	●	●	●
P23	I/O port / EXI5	●	●	●	●	●	●	●	●	●
P24	I/O port	-	●	●	●	●	●	●	●	●
P25	I/O port	-	●	●	●	●	●	●	●	●
P26	I/O port / EXI6	●	●	●	●	●	●	●	●	●
P27	I/O port / EXI7	●	●	●	●	●	●	●	●	●
P30	I/O port	-	-	-	●	●	●	●	●	●
P31	I/O port	-	-	-	●	●	●	●	●	●
P32	I/O port	-	-	●	●	●	●	●	●	●
P33	I/O port	-	●	●	●	●	●	●	●	●

Pin Name	Primary Function	Available / Unavailable								
		ML62Q1300 group				ML62Q1500/1800/1700 group				
		16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
P40	I/O port	-	-	-	-	-	-	●	●	●
P41	I/O port	-	-	-	-	-	●	●	●	●
P42	I/O port	-	-	-	-	-	-	●	●	●
P43	I/O port	-	-	-	-	●	●	●	●	●
P44	I/O port / DACOUT1	-	-	-	-	-	-	●	●	●
P45	I/O port	-	-	-	-	-	-	●	●	●
P46	I/O port	-	-	-	-	-	-	●	●	●
P47	I/O port	-	-	-	-	-	●	●	●	●
P50	I/O port / EXI8	-	-	-	-	●	●	●	●	●
P51	I/O port	-	-	-	-	-	●	●	●	●
P52	I/O port	-	-	-	-	-	-	●	●	●
P53	I/O port	-	-	-	-	-	-	●	●	●
P54	I/O port	-	-	-	-	-	-	●	●	●
P55	I/O port	-	-	-	-	-	-	●	●	●
P56	I/O port	-	-	-	-	-	●	●	●	●
P57	I/O port	-	-	-	-	-	-	●	●	●
P60	I/O port	-	-	-	-	●	●	●	●	●
P61	I/O port	-	-	-	-	●	●	●	●	●
P62	I/O port	-	-	-	-	●	●	●	●	●
P63	I/O port	-	-	-	-	●	●	●	●	●
P64	I/O port / EXI9	-	-	-	-	●	●	●	●	●
P65	I/O port	-	-	-	-	●	●	●	●	●
P66	I/O port	-	-	-	-	●	●	●	●	●
P67	I/O port	-	-	-	-	-	-	●	●	●
P70	I/O port	-	-	-	-	-	-	●	●	●
P71	I/O port	-	-	-	-	●*1	●*1	●*1	●*1	●*1
P72	I/O port	-	-	-	-	●*1	●*1	●*1	●*1	●*1
P73	I/O port	-	-	-	-	●*1	●*1	●*1	●*1	●*1
P74	I/O port	-	-	-	-	●*1	●*1	●*1	●*1	●*1
P75	I/O port	-	-	-	-	●*1	●*1	●*1	●*1	●*1
P76	I/O port / EXI10	-	-	-	-	-	-	-	●	●
P77	I/O port	-	-	-	-	-	-	-	-	●
P80	I/O port	-	-	-	-	-	-	-	●	●
P81	I/O port	-	-	-	-	-	-	-	●	●
P82	I/O port	-	-	-	-	-	-	-	●	●
P83	I/O port	-	-	-	-	-	-	-	-	●
P84	I/O port	-	-	-	-	-	-	-	-	●

Pin Name	Primary Function	Available / Unavailable								
		ML62Q1300 group				ML62Q1500/1800/1700 group				
		16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
P85	I/O port	-	-	-	-	-	-	-	-	●
P86	I/O port	-	-	-	-	-	-	-	-	●
P87	I/O port	-	-	-	-	-	-	-	-	●
P90	I/O port	-	-	-	-	-	-	-	-	●
P91	I/O port	-	-	-	-	-	-	-	-	●
P92	I/O port	-	-	-	-	-	-	-	-	●
P93	I/O port	-	-	-	-	-	-	-	●	●
P94	I/O port	-	-	-	-	-	-	-	●	●
P95	I/O port	-	-	-	-	-	-	-	●	●
P96	I/O port	-	-	-	-	-	-	-	●	●
P97	I/O port	-	-	-	-	-	-	-	-	●
PA0	I/O port	-	-	-	-	-	-	-	-	●
PA1	I/O port	-	-	-	-	-	-	-	-	●
PA2	I/O port	-	-	-	-	-	-	-	-	●
PA3	I/O port / EXI11	-	-	-	-	-	-	-	●	●
PA4	I/O port	-	-	-	-	-	-	-	●	●
PA5	I/O port	-	-	-	-	-	-	-	-	●
PA6	I/O port	-	-	-	-	-	-	-	-	●
PA7	I/O port	-	-	-	-	-	-	-	-	●
PB0	I/O port	-	-	-	-	-	-	-	-	●
PB1	I/O port	-	-	-	-	-	-	-	-	●
PB2	I/O port	-	-	-	-	-	-	-	●	●
PB3	I/O port	-	-	-	-	-	-	-	●	●
PB4	I/O port	-	-	-	-	-	-	-	●	●
PB5	I/O port	-	-	-	-	-	-	-	●	●
PB6	I/O port	-	-	-	-	-	-	-	-	●
PB7	I/O port	-	-	-	-	-	-	-	-	●

●: Available to use -: Unavailable ●*1 : Available on the ML62Q1500/ML62Q1800 group only

17.2 Description of Registers

17.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
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	0x05
0xF203		P0MOD1		R/W	8	0x00
0xF204	Port 0 mode register 23	P0MOD2	P0MOD23	R/W	8/16	0x00
0xF205		P0MOD3		R/W	8	0x00
0xF206	Port 0 mode register 45	P0MOD4	P0MOD45	R/W	8/16	0x00
0xF207		P0MOD5		R/W	8	0x00
0xF208	Port 0 mode register 67	P0MOD6	P0MOD67	R/W	8/16	0x00
0xF209		P0MOD7		R/W	8	0x00
0xF20A	Port 0 pulse mode register	P0PMDL	P0PMD	R/W	8/16	0x00
0xF20B		P0PMDH		R/W	8	0x00
0xF20C	Port 0 pulse selection register	P0PSLL	P0PSL	R/W	8/16	0x00
0xF20D		P0PSLH		R/W	8	0x00
0xF20E	Reserved	-	-	-	-	-
0xF20F						
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	Port 1 pulse selection register	P1PSLL	P1PSL	R/W	8/16	0x00
0xF21D		P1PSLH		R/W	8	0x00
0xF21E	Reserved	-	-	-	-	-
0xF21F						
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

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF22A	Port 2 pulse mode register	P2PMDL	P2PMD	R/W	8/16	0x00
0xF22B		P2PMDH		R/W	8	0x00
0xF22C	Port 2 pulse selection register	P2PSLL	P2PSL	R/W	8/16	0x00
0xF22D		P2PSLH		R/W	8	0x00
0xF22E	Reserved	-	-	-	-	-
0xF22F						
0xF230	Port 3 data register	P3DI	P3D	R/W	8/16	0xFF
0xF231		P3DO		R/W	8	0x00
0xF232	Port 3 mode register 01	P3MOD0	P3MOD01	R/W	8/16	0x00
0xF233		P3MOD1		R/W	8	0x00
0xF234	Port 3 mode register 23	P3MOD2	P3MOD23	R/W	8/16	0x00
0xF235		P3MOD3		R/W	8	0x00
0xF236 to 0xF239	Reserved	-	-	-	-	-
0xF23A	Port 3 pulse mode register	P3PMDL	P3PMD	R/W	8/16	0x00
0xF23B		P3PMDH		R/W	8	0x00
0xF23C	Port 3 pulse selection register	P3PSLL	P3PSL	R/W	8/16	0x00
0xF23D		P3PSLH		R/W	8	0x00
0xF23E	Reserved	-	-	-	-	-
0xF23F						
0xF240	Port 4 data register	P4DI	P4D	R/W	8/16	0xFF
0xF241		P4DO		R/W	8	0x00
0xF242	Port 4 mode register 01	P4MOD0	P4MOD01	R/W	8/16	0x00
0xF243		P4MOD1		R/W	8	0x00
0xF244	Port 4 mode register 23	P4MOD2	P4MOD23	R/W	8/16	0x00
0xF245		P4MOD3		R/W	8	0x00
0xF246	Port 4 mode register 45	P4MOD4	P4MOD45	R/W	8/16	0x00
0xF247		P4MOD5		R/W	8	0x00
0xF248	Port 4 mode register 67	P4MOD6	P4MOD67	R/W	8/16	0x00
0xF249		P4MOD7		R/W	8	0x00
0xF24A to 0xF24F	Reserved	-	-	-	-	-
0xF250	Port 5 data register	P5DI	P5D	R/W	8/16	0xFF
0xF251		P5DO		R/W	8	0x00
0xF252	Port 5 mode register 01	P5MOD0	P5MOD01	R/W	8/16	0x00
0xF253		P5MOD1		R/W	8	0x00
0xF254	Port 5 mode register 23	P5MOD2	P5MOD23	R/W	8/16	0x00
0xF255		P5MOD3		R/W	8	0x00
0xF256	Port 5 mode register 45	P5MOD4	P5MOD45	R/W	8/16	0x00
0xF257		P5MOD5		R/W	8	0x00
0xF258	Port 5 mode register 67	P5MOD6	P5MOD67	R/W	8/16	0x00
0xF259		P5MOD7		R/W	8	0x00
0xF25A to 0xF25F	Reserved	-	-	-	-	-

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF260	Port 6 data register	P6DI	P6D	R/W	8/16	0xFF
0xF261		P6DO		R/W	8	0x00
0xF262	Port 6 mode register 01	P6MOD0	P6MOD01	R/W	8/16	0x00
0xF263		P6MOD1		R/W	8	0x00
0xF264	Port 6 mode register 23	P6MOD2	P6MOD23	R/W	8/16	0x00
0xF265		P6MOD3		R/W	8	0x00
0xF266	Port 6 mode register 45	P6MOD4	P6MOD45	R/W	8/16	0x00
0xF267		P6MOD5		R/W	8	0x00
0xF268	Port 6 mode register 67	P6MOD6	P6MOD67	R/W	8/16	0x00
0xF269		P6MOD7		R/W	8	0x00
0xF26A to 0xF26F	Reserved	-	-	-	-	-
0xF270	Port 7 data register	P7DI	P7D	R/W	8/16	0xFF
0xF271		P7DO		R/W	8	0x00
0xF272	Port 7 mode register 01	P7MOD0	P7MOD01	R/W	8/16	0x00
0xF273		P7MOD1		R/W	8	0x00
0xF274	Port 7 mode register 23	P7MOD2	P7MOD23	R/W	8/16	0x00
0xF275		P7MOD3		R/W	8	0x00
0xF276	Port 7 mode register 45	P7MOD4	P7MOD45	R/W	8/16	0x00
0xF277		P7MOD5		R/W	8	0x00
0xF278	Port 7 mode register 67	P7MOD6	P7MOD67	R/W	8/16	0x00
0xF279		P7MOD7		R/W	8	0x00
0xF27A to 0xF27F	Reserved	-	-	-	-	-
0xF280	Port 8 data register	P8DI	P8D	R/W	8/16	0xFF
0xF281		P8DO		R/W	8	0x00
0xF282	Port 8 mode register 01	P8MOD0	P8MOD01	R/W	8/16	0x00
0xF283		P8MOD1		R/W	8	0x00
0xF284	Port 8 mode register 23	P8MOD2	P8MOD23	R/W	8/16	0x00
0xF285		P8MOD3		R/W	8	0x00
0xF286	Port 8 mode register 45	P8MOD4	P8MOD45	R/W	8/16	0x00
0xF287		P8MOD5		R/W	8	0x00
0xF288	Port 8 mode register 67	P8MOD6	P8MOD67	R/W	8/16	0x00
0xF289		P8MOD7		R/W	8	0x00
0xF28A to 0xF28F	Reserved	-	-	-	-	-
0xF290	Port 9 data register	P9DI	P9D	R/W	8/16	0xFF
0xF291		P9DO		R/W	8	0x00
0xF292	Port 9 mode register 01	P9MOD0	P9MOD01	R/W	8/16	0x00
0xF293		P9MOD1		R/W	8	0x00
0xF294	Port 9 mode register 23	P9MOD2	P9MOD23	R/W	8/16	0x00
0xF295		P9MOD3		R/W	8	0x00
0xF296	Port 9 mode register 45	P9MOD4	P9MOD45	R/W	8/16	0x00
0xF297		P9MOD5		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF298	Port 9 mode register 67	P9MOD6	P9MOD67	R/W	8/16	0x00
0xF299		P9MOD7		R/W	8	0x00
0xF29A to 0xF29F	Reserved	-	-	-	-	-
0xF2A0	Port A data register	PADI	PAD	R/W	8/16	0xFF
0xF2A1		PADO		R/W	8	0x00
0xF2A2	Port A mode register 01	PAMOD0	PAMOD01	R/W	8/16	0x00
0xF2A3		PAMOD1		R/W	8	0x00
0xF2A4	Port A mode register 23	PAMOD2	PAMOD23	R/W	8/16	0x00
0xF2A5		PAMOD3		R/W	8	0x00
0xF2A6	Port A mode register 45	PAMOD4	PAMOD45	R/W	8/16	0x00
0xF2A7		PAMOD5		R/W	8	0x00
0xF2A8	Port A mode register 67	PAMOD6	PAMOD67	R/W	8/16	0x00
0xF2A9		PAMOD7		R/W	8	0x00
0xF2AA to 0xF2AF	Reserved	-	-	-	-	-
0xF2B0	Port B data register	PBDI	PBD	R/W	8/16	0xFF
0xF2B1		PBDO		R/W	8	0x00
0xF2B2	Port B mode register 01	PBMOD0	PBMOD01	R/W	8/16	0x00
0xF2B3		PBMOD1		R/W	8	0x00
0xF2B4	Port B mode register 23	PBMOD2	PBMOD23	R/W	8/16	0x00
0xF2B5		PBMOD3		R/W	8	0x00
0xF2B6	Port B mode register 45	PBMOD4	PBMOD45	R/W	8/16	0x00
0xF2B7		PBMOD5		R/W	8	0x00
0xF2B8	Port B mode register 67	PBMOD6	PBMOD67	R/W	8/16	0x00
0xF2B9		PBMOD7		R/W	8	0x00
0xF2BA to 0xF2EF	Reserved	-	-	-	-	-
0xF2F0	PORTXT data input register	PXTDI	-	R	8	Undefined
0xF2F1	Reserved	-	-	-	-	-
0xF2F2	PORTXT mode register 01	PXTMOD0	PXTMOD01	R/W	8/16	0x00
0xF2F3		PXTMOD1		R/W	8	0x00

[Note]

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

Table 17-2 List of Registers / Bits

Port Name	Pin Name	Control register / bit *1						Available / Unavailable *2								
								ML62Q1300 group				ML62Q1500/1800 ML62Q1700 group				
		Port n data register (PnD)		Port n mode register m (PnMODm)	Port n pulse mode register (PnPMD)		Port n pulse selection register (PnPSL)	16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
Port XT	PI00	-	PXT0DI	PXTMOD0	-	-	-	-	-	-	-	●	●	●	●	●
	PI01	-	PXT1DI	PXTMOD1	-	-	-	-	-	-	-	●	●	●	●	●
Port 0	P00	P00DO	P00DI	P0MOD0	-	-	-	●	●	●	●	●	●	●	●	●
	P01	P01DO	P01DI	P0MOD1	-	-	-	-	-	●	●	●	●	●	●	●
	P02	P02DO	P02DI	P0MOD2	-	-	-	●	●	●	●	●	●	●	●	●
	P03	P03DO	P03DI	P0MOD3	P03PLVL	P03PEN	P03PSL	●	●	●	●	●	●	●	●	●
	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	P11PSL	-	-	-	●	●	●	●	●	●
	P12	P12DO	P12DI	P1MOD2	-	-	-	-	-	●	●	●	●	●	●	●
	P13	P13DO	P13DI	P1MOD3	P13PLVL	P13PEN	P13PSL	●	●	●	●	●	●	●	●	●
	P14	P14DO	P14DI	P1MOD4	-	-	-	-	-	-	●	●	●	●	●	●
	P15	P15DO	P15DI	P1MOD5	-	-	-	-	-	-	●	●	●	●	●	●
	P16	P16DO	P16DI	P1MOD6	-	-	-	-	-	●	●	●	●	●	●	●
	P17	P17DO	P17DI	P1MOD7	-	-	-	●	●	●	●	●	●	●	●	●
Port 2	P20	P20DO	P20DI	P2MOD0	P20PLVL	P20PEN	P20PSL	●	●	●	●	●	●	●	●	●
	P21	P21DO	P21DI	P2MOD1	-	-	-	●	●	●	●	●	●	●	●	●
	P22	P22DO	P22DI	P2MOD2	P22PLVL	P22PEN	P22PSL	●	●	●	●	●	●	●	●	●
	P23	P23DO	P23DI	P2MOD3	-	-	-	●	●	●	●	●	●	●	●	●
	P24	P24DO	P24DI	P2MOD4	-	-	-	-	●	●	●	●	●	●	●	●
	P25	P25DO	P25DI	P2MOD5	P25PLVL	P25PEN	P25PSL	-	●	●	●	●	●	●	●	●
	P26	P26DO	P26DI	P2MOD6	-	-	-	●	●	●	●	●	●	●	●	●
	P27	P27DO	P27DI	P2MOD7	P27PLVL	P27PEN	P27PSL	●	●	●	●	●	●	●	●	●
Port 3	P30	P30DO	P30DI	P3MOD0	-	-	-	-	-	-	●	●	●	●	●	●
	P31	P31DO	P31DI	P3MOD1	-	-	-	-	-	-	●	●	●	●	●	●
	P32	P32DO	P32DI	P3MOD2	-	-	-	-	-	●	●	●	●	●	●	●
	P33	P33DO	P33DI	P3MOD3	P33PLVL	P33PEN	P33PSL	-	●	●	●	●	●	●	●	●

Port Name	Pin Name	Control register / bit *1						Available / Unavailable *2								
								ML62Q1300 group				ML62Q1500/1800 ML62Q1700 group				
		Port n data register (PnD)		Port n mode register m (PnMODm)	Port n pulse mode register (PnPMD)		Port n pulse selection register (PnPSL)	16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
Port 4	P40	P40DO	P40DI	P4MOD0	-	-	-	-	-	-	-	-	-	●	●	●
	P41	P41DO	P41DI	P4MOD1	-	-	-	-	-	-	-	-	●	●	●	●
	P42	P42DO	P42DI	P4MOD2	-	-	-	-	-	-	-	-	-	●	●	●
	P43	P43DO	P43DI	P4MOD3	-	-	-	-	-	-	-	●	●	●	●	●
	P44	P44DO	P44DI	P4MOD4	-	-	-	-	-	-	-	-	-	●	●	●
	P45	P45DO	P45DI	P4MOD5	-	-	-	-	-	-	-	-	-	●	●	●
	P46	P46DO	P46DI	P4MOD6	-	-	-	-	-	-	-	-	-	●	●	●
	P47	P47DO	P47DI	P4MOD7	-	-	-	-	-	-	-	-	●	●	●	●
Port 5	P50	P50DO	P50DI	P5MOD0	-	-	-	-	-	-	-	●	●	●	●	●
	P51	P51DO	P51DI	P5MOD1	-	-	-	-	-	-	-	-	●	●	●	●
	P52	P52DO	P52DI	P5MOD2	-	-	-	-	-	-	-	-	-	●	●	●
	P53	P53DO	P53DI	P5MOD3	-	-	-	-	-	-	-	-	-	●	●	●
	P54	P54DO	P54DI	P5MOD4	-	-	-	-	-	-	-	-	-	●	●	●
	P55	P55DO	P55DI	P5MOD5	-	-	-	-	-	-	-	-	-	●	●	●
	P56	P56DO	P56DI	P5MOD6	-	-	-	-	-	-	-	-	●	●	●	●
	P57	P57DO	P57DI	P5MOD7	-	-	-	-	-	-	-	-	-	●	●	●
Port 6	P60	P60DO	P60DI	P6MOD0	-	-	-	-	-	-	-	●	●	●	●	●
	P61	P61DO	P61DI	P6MOD1	-	-	-	-	-	-	-	●	●	●	●	●
	P62	P62DO	P62DI	P6MOD2	-	-	-	-	-	-	-	●	●	●	●	●
	P63	P63DO	P63DI	P6MOD3	-	-	-	-	-	-	-	●	●	●	●	●
	P64	P64DO	P64DI	P6MOD4	-	-	-	-	-	-	-	●	●	●	●	●
	P65	P65DO	P65DI	P6MOD5	-	-	-	-	-	-	-	●	●	●	●	●
	P66	P66DO	P66DI	P6MOD6	-	-	-	-	-	-	-	●	●	●	●	●
	P67	P67DO	P67DI	P6MOD7	-	-	-	-	-	-	-	-	-	●	●	●
Port 7	P70	P70DO	P70DI	P7MOD0	-	-	-	-	-	-	-	-	-	●	●	●
	P71	P71DO	P71DI	P7MOD1	-	-	-	-	-	-	-	●*1	●*1	●*1	●*1	●*1
	P72	P72DO	P72DI	P7MOD2	-	-	-	-	-	-	-	●*1	●*1	●*1	●*1	●*1
	P73	P73DO	P73DI	P7MOD3	-	-	-	-	-	-	-	●*1	●*1	●*1	●*1	●*1
	P74	P74DO	P74DI	P7MOD4	-	-	-	-	-	-	-	●*1	●*1	●*1	●*1	●*1
	P75	P75DO	P75DI	P7MOD5	-	-	-	-	-	-	-	●*1	●*1	●*1	●*1	●*1
	P76	P76DO	P76DI	P7MOD6	-	-	-	-	-	-	-	-	-	-	●	●
	P77	P77DO	P77DI	P7MOD7	-	-	-	-	-	-	-	-	-	-	-	●

Port Name	Pin Name	Control register / bit *1						Available / Unavailable *2								
								ML62Q1300 group				ML62Q1500/1800 ML62Q1700 group				
		Port n data register (PnD)		Port n mode register m (PnMODm)	Port n pulse mode register (PnPMD)		Port n pulse selection register (PnPSL)	16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
Port 8	P80	P80DO	P80DI	P8MOD0	-	-	-	-	-	-	-	-	-	-	●	●
	P81	P81DO	P81DI	P8MOD1	-	-	-	-	-	-	-	-	-	-	●	●
	P82	P82DO	P82DI	P8MOD2	-	-	-	-	-	-	-	-	-	-	●	●
	P83	P83DO	P83DI	P8MOD3	-	-	-	-	-	-	-	-	-	-	-	●
	P84	P84DO	P84DI	P8MOD4	-	-	-	-	-	-	-	-	-	-	-	●
	P85	P85DO	P85DI	P8MOD5	-	-	-	-	-	-	-	-	-	-	-	●
	P86	P86DO	P86DI	P8MOD6	-	-	-	-	-	-	-	-	-	-	-	●
	P87	P87DO	P87DI	P8MOD7	-	-	-	-	-	-	-	-	-	-	-	●
Port 9	P90	P90DO	P90DI	P9MOD0	-	-	-	-	-	-	-	-	-	-	-	●
	P91	P91DO	P91DI	P9MOD1	-	-	-	-	-	-	-	-	-	-	-	●
	P92	P92DO	P92DI	P9MOD2	-	-	-	-	-	-	-	-	-	-	-	●
	P93	P93DO	P93DI	P9MOD3	-	-	-	-	-	-	-	-	-	-	●	●
	P94	P94DO	P94DI	P9MOD4	-	-	-	-	-	-	-	-	-	-	●	●
	P95	P95DO	P95DI	P9MOD5	-	-	-	-	-	-	-	-	-	-	●	●
	P96	P96DO	P96DI	P9MOD6	-	-	-	-	-	-	-	-	-	-	●	●
	P97	P97DO	P97DI	P9MOD7	-	-	-	-	-	-	-	-	-	-	-	●
Port A	PA0	PA0DO	PA0DI	PAMOD0	-	-	-	-	-	-	-	-	-	-	-	●
	PA1	PA1DO	PA1DI	PAMOD1	-	-	-	-	-	-	-	-	-	-	-	●
	PA2	PA2DO	PA2DI	PAMOD2	-	-	-	-	-	-	-	-	-	-	-	●
	PA3	PA3DO	PA3DI	PAMOD3	-	-	-	-	-	-	-	-	-	-	●	●
	PA4	PA4DO	PA4DI	PAMOD4	-	-	-	-	-	-	-	-	-	-	●	●
	PA5	PA5DO	PA5DI	PAMOD5	-	-	-	-	-	-	-	-	-	-	-	●
	PA6	PA6DO	PA6DI	PAMOD6	-	-	-	-	-	-	-	-	-	-	-	●
	PA7	PA7DO	PA7DI	PAMOD7	-	-	-	-	-	-	-	-	-	-	-	●
Port B	PB0	PB0DO	PB0DI	PBMOD0	-	-	-	-	-	-	-	-	-	-	-	●
	PB1	PB1DO	PB1DI	PBMOD1	-	-	-	-	-	-	-	-	-	-	-	●
	PB2	PB2DO	PB2DI	PBMOD2	-	-	-	-	-	-	-	-	-	-	●	●
	PB3	PB3DO	PB3DI	PBMOD3	-	-	-	-	-	-	-	-	-	-	●	●
	PB4	PB4DO	PB4DI	PBMOD4	-	-	-	-	-	-	-	-	-	-	●	●
	PB5	PB5DO	PB5DI	PBMOD5	-	-	-	-	-	-	-	-	-	-	●	●
	PB6	PB6DO	PB6DI	PBMOD6	-	-	-	-	-	-	-	-	-	-	-	●
	PB7	PB7DO	PB7DI	PBMOD7	-	-	-	-	-	-	-	-	-	-	-	●

*1 : Corresponding bits and registers for each pin.
*2 : "●" Available "-" Unavailable "●*1" Available on the ML62Q1500/ML62Q1800 group only. See descriptions in each register for writing/reading the unavailable registers and bits.

17.2.2 Port n Data Register (PnD:n=0 to 9, A, B)

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 data written to PnDO is readable. The bit can be set when output is enabled or disabled.
Enable or disable the input or output by using the port n mode register.
See Table 17-2 “List of Registers / Bits” to check available pins and bits.
Write “0” to the bits of PnDO that have no corresponding pin.

Address: 0xF200(P0DI/P0D), 0xF201(P0DO), 0xF210(P1DI/P1D), 0xF211(P1DO),
0xF220(P2DI/P2D), 0xF221(P2DO), 0xF230(P3DI/P3D), 0xF231(P3DO),
0xF240(P4DI/P4D), 0xF241(P4DO), 0xF250(P5DI/P5D), 0xF251(P5DO),
0xF260(P6DI/P6D), 0xF261(P6DO), 0xF270(P7DI/P7D), 0xF271(P7DO),
0xF280(P8DI/P8D), 0xF281(P8DO), 0xF290(P9DI/P9D), 0xF291(P9DO),
0xF2A0(PADI/PAD), 0xF2A1(PADO), 0xF2B0(PBDI/PBD), 0xF2B1(PBDO)
Access: R/W
Access size: 8/16bit
Initial value: 0x00FF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnD															
Byte	PnDO								PnDI							
Bit	Pn7DO	Pn6DO	Pn5DO	Pn4DO	Pn3DO	Pn2DO	Pn1DO	Pn0DO	Pn7DI	Pn6DI	Pn5DI	Pn4DI	Pn3DI	Pn2DI	Pn1DI	Pn0DI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 8	Pn7DO to Pn0DO	These bits are used to set the output level of port n pin. 0: Output "L" (Initial value) 1: Output "H"
7 to 0	Pn7DI to Pn0DI	These bits are used to set the input level of port n pin. 0: The input level is "L" 1: The input level is "H" (Initial value)

17.2.3 Port n Mode Register 01 (PnMOD01:n=0 to 9, A, B)

PnMOD01 is a special function register (SFR) to choose the input/output mode, input/output status, and shared function of Pn0 pin and Pn1 pin.
See Table 17-2 “List of Registers / Bits” to check available pins and bits.
Write “0” to the bits of PnMOD01 register that have no corresponding pin.

Address: 0xF202(P0MOD0/P0MOD01), 0xF203(P0MOD1),
0xF212(P1MOD0/P1MOD01), 0xF213(P1MOD1),
0xF222(P2MOD0/P2MOD01), 0xF223(P2MOD1),
0xF232(P3MOD0/P3MOD01), 0xF233(P3MOD1),
0xF242(P4MOD0/P4MOD01), 0xF243(P4MOD1),
0xF252(P5MOD0/P5MOD01), 0xF253(P5MOD1),
0xF262(P6MOD0/P6MOD01), 0xF263(P6MOD1),
0xF272(P7MOD0/P7MOD01), 0xF273(P7MOD1),
0xF282(P8MOD0/P8MOD01), 0xF283(P8MOD1),
0xF292(P9MOD0/P9MOD01), 0xF293(P9MOD1),
0xF2A2(PAMOD0/PAMOD01), 0xF2A3(PAMOD1),
0xF2B2(PBMOD0/PBMOD01), 0xF2B3(PBMOD1)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000 (0x0005 at P0MOD01)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD01															
Byte	PnMOD1								PnMOD0							
Bit	Pn1MD3	Pn1MD2	Pn1MD1	Pn1MD0	Pn1OD	Pn1PU	Pn1OE	Pn1IE	Pn0MD3	Pn0MD2	Pn0MD1	Pn0MD0	Pn0OD	Pn0PU	Pn0OE	Pn0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	*	0	*

*:The initial value of P00IE and P00PU for the Port0 is "1" and other bits are "0".

Bit No.	Bit symbol name	Description
15 to 12	Pn1MD3 to Pn1MD0	These bits are used to choose the shared function of Pn1 pin. For the details of the shared function, see Table 1-7 "ML62Q1300 Group Pin List", Table 1-8 "ML62Q1500/ML62Q1800 Group Pin List" and Table 1-9 "ML62Q1700 Group Pin List". 0000: Primary function (Initial value) 0001: 2nd function 0010: 3rd function 0011: 4th function 0100: 5th function 0101: 6th function 0110: 7th function 0111: 8th function 1XXX: Do not use (Primary function) X: 0 or 1 (don't care)
11	Pn1OD	These bits are used choose the output type of Pn1 pin. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is chosen. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output

Bit No.	Bit symbol name	Description
10	Pn1PU	This bit is used to enable the internal pull-up resistor of Pn1 pin. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The internal pull-up resistor can be enabled on following conditions of the port. It is disabled in other setting. · The input is enabled and the output is disabled on the port (Pn1IE=1, Pn1OE=0) · The input is enabled and the N-channel open drain output is chosen on the port (Pn1IE=1, Pn1OE=1, Pn1OD=1)
9	Pn1OE	This bit is used to enable the output of Pn1 pin 0: Disable the output (Initial value) 1: Enable the output
8	Pn1IE	This bit is used to enable the input of Pn1 pin 0: Disable the input (Initial value) 1: Enable the input
7 to 4	Pn0MD3 to Pn0MD0	These bits are used to choose the shared function of Pn0 pin. For the details of the shared function, see Table 1-7 "ML62Q1300 Group Pin List", Table 1-8 "ML62Q1500/ML62Q1800 Group Pin List" and Table 1-9 "ML62Q1700 Group Pin List". 0000: Primary function (Initial value) 0001: 2nd function 0010: 3rd function 0011: 4th function 0100: 5th function 0101: 6th function 0110: 7th function 0111: 8th function 1XXX: Do not use (Primary function) X: 0 or 1 (don't care)
3	Pn0OD	This bit is used choose the output type of Pn0 pin. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is chosen. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output
2	Pn0PU	This bit is used to enable the internal pull-up resistor of Pn0 pin. 0: Without a pull-up resistor (Initial value*) 1: With a pull-up resistor *: The initial value is "1" at P0MOD0. The internal pull-up resistor can be enabled on following conditions of the port. It is disabled in other setting. · The input is enabled and the output is disabled on the port (Pn0IE=1, Pn0OE=0) · The input is enabled and the N-channel open drain output is chosen on the port (Pn0IE=1, Pn0OE=1, Pn0OD=1)
1	Pn0OE	This bit is used to enable the output of Pn0 pin 0: Disable the output (Initial value) 1: Enable the output
0	Pn0IE	This bit is used to enable the input of Pn0 pin 0: Disable the input (initial value) 1: Enable the input *: The initial value is "1" at P0MOD0.

[Note]

- **Be sure to set the PnMODm(n=0 to B, m=0 to 7) registers before setting EICON0, EIMOD0 and IE1 registers. If setting the PnMODm register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**

17.2.4 Port n Mode Register 23 (PnMOD23:n=0 to 9, A, B)

PnMOD23 is a special function register (SFR) to choose the input/output mode, input/output status, and shared function of Pn2 pin and Pn3 pin.
See Table 17-2 "List of Registers / Bits" to check available pins and bits.
Write "0" to the bits of PnMOD23 register that have no corresponding pin.

Address: 0xF204(P0MOD2/P0MOD23), 0xF205(P0MOD3),
0xF214(P1MOD2/P1MOD23), 0xF215(P1MOD3),
0xF224(P2MOD2/P2MOD23), 0xF225(P2MOD3),
0xF234(P3MOD2/P3MOD23), 0xF235(P3MOD3),
0xF244(P4MOD2/P4MOD23), 0xF245(P4MOD3),
0xF254(P5MOD2/P5MOD23), 0xF255(P5MOD3),
0xF264(P6MOD2/P6MOD23), 0xF265(P6MOD3),
0xF274(P7MOD2/P7MOD23), 0xF275(P7MOD3),
0xF284(P8MOD2/P8MOD23), 0xF285(P8MOD3),
0xF294(P9MOD2/P9MOD23), 0xF295(P9MOD3),
0xF2A4(PAMOD2/PAMOD23), 0xF2A5(PAMOD3),
0xF2B4(PBMOD2/PBMOD23), 0xF2B5(PBMOD3)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD23															
Byte	PnMOD3								PnMOD2							
Bit	Pn3MD3	Pn3MD2	Pn3MD1	Pn3MD0	Pn3OD	Pn3PU	Pn3OE	Pn3IE	Pn2MD3	Pn2MD2	Pn2MD1	Pn2MD0	Pn2OD	Pn2PU	Pn2OE	Pn2IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 12	Pn3MD3 to Pn3MD0	These bits are used to choose the shared function of Pn3 pin. For the details of the shared function, see Table 1-7 "ML62Q1300 Group Pin List", Table 1-8 "ML62Q1500/ML62Q1800 Group Pin List" and Table 1-9 "ML62Q1700 Group Pin List". 0000: Primary function (Initial value) 0001: 2nd function 0010: 3rd function 0011: 4th function 0100: 5th function 0101: 6th function 0110: 7th function 0111: 8th function 1XXX: Do not use (Primary function) X: 0 or 1 (don't care)
11	Pn3OD	This bit is used choose the output type of Pn3 pin. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is chosen. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output

Bit No.	Bit symbol name	Description
10	Pn3PU	This bit is used to enable the internal pull-up resistor of Pn3 pin. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The internal pull-up resistor can be enabled on following conditions of the port. It is disabled in other setting. · The input is enabled and the output is disabled on the port (Pn3IE=1, Pn3OE=0) · The input is enabled and the N-channel open drain output is chosen on the port (Pn3IE=1, Pn3OE=1, Pn3OD=1)
9	Pn3OE	This bit is used to enable the output of Pn3 pin 0: Disable the output (Initial value) 1: Enable the output
8	Pn3IE	This bit is used to enable the input of Pn3 pin 0: Disable the input (Initial value) 1: Enable the input
7 to 4	Pn2MD3 to Pn2MD0	These bits are used to choose the shared function of Pn2 pin. For the details of the shared function, see Table 1-7 "ML62Q1300 Group Pin List", Table 1-8 "ML62Q1500/ML62Q1800 Group Pin List" and Table 1-9 "ML62Q1700 Group Pin List". 0000: Primary function (Initial value) 0001: 2nd function 0010: 3rd function 0011: 4th function 0100: 5th function 0101: 6th function 0110: 7th function 0111: 8th function 1XXX: Do not use (Primary function) X: 0 or 1 (don't care)
3	Pn2OD	This bit is used choose the output type of Pn2 pin. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is chosen. See the data sheet for details about the current drive ability. 0: CMOS output (Initial value) 1: N-channel open drain output
2	Pn2PU	This bit is used to enable the internal pull-up resistor of Pn2 pin. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor The internal pull-up resistor can be enabled on following conditions of the port. It is disabled in other setting. · The input is enabled and the output is disabled on the port (Pn2IE=1, Pn2OE=0) · The input is enabled and the N-channel open drain output is chosen on the port (Pn2IE=1, Pn2OE=1, Pn2OD=1)
1	Pn2OE	This bit is used to enable the output of Pn2 pin 0: Disable the output (initial value) 1: Enable the output

Bit No.	Bit symbol name	Description
0	Pn2IE	This bit is used to enable the input of Pn2 pin 0: Disable the input (initial value) 1: Enable the input

[Note]

- **Be sure to set the PnMODm(n=0 to B, m=0 to 7) registers before setting EICON0, EIMOD0 and IE1 registers. If setting the PnMODm register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**

17.2.5 Port n Mode Register 45 (PnMOD45:n=0 to 2, 4 to 9, A, B)

PnMOD45 is a special function register (SFR) to choose the input/output mode, input/output status, and shared function of Pn4 pin and Pn5 pin.
See Table 17-2 "List of Registers / Bits" to check available pins and bits.
Write "0" to the bits of PnMOD45 register that have no corresponding pin.

Address: 0xF206(P0MOD4/P0MOD45), 0xF207(P0MOD5),
0xF216(P1MOD4/P1MOD45), 0xF217(P1MOD5),
0xF226(P2MOD4/P2MOD45), 0xF227(P2MOD5),
0xF246(P4MOD4/P4MOD45), 0xF247(P4MOD5),
0xF256(P5MOD4/P5MOD45), 0xF257(P5MOD5),
0xF266(P6MOD4/P6MOD45), 0xF267(P6MOD5),
0xF276(P7MOD4/P7MOD45), 0xF277(P7MOD5),
0xF286(P8MOD4/P8MOD45), 0xF287(P8MOD5),
0xF296(P9MOD4/P9MOD45), 0xF297(P9MOD5),
0xF2A6(PAMOD4/PAMOD45), 0xF2A7(PAMOD5),
0xF2B6(PBMOD4/PBMOD45), 0xF2B7(PBMOD5)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD45															
Byte	PnMOD5								PnMOD4							
Bit	Pn5MD3	Pn5MD2	Pn5MD1	Pn5MD0	Pn5OD	Pn5PU	Pn5OE	Pn5IE	Pn4MD3	Pn4MD2	Pn4MD1	Pn4MD0	Pn4OD	Pn4PU	Pn4OE	Pn4IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 12	Pn5MD3 to Pn5MD0	These bits are used to choose the shared function of Pn5 pin. For the details of the shared function, see Table 1-7 "ML62Q1300 Group Pin List", Table 1-8 "ML62Q1500/ML62Q1800 Group Pin List" and Table 1-9 "ML62Q1700 Group Pin List". 0000: Primary function (initial value) 0001: 2nd function 0010: 3rd function 0011: 4th function 0100: 5th function 0101: 6th function 0110: 7th function 0111: 8th function 1XXX: Do not use (Primary function) X: 0 or 1 (don't care)
11	Pn5OD	This bit is used choose the output type of Pn5 pin. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is chosen. See the data sheet for details about the current drive ability. 0: CMOS output (initial value) 1: N-channel open drain output

Bit No.	Bit symbol name	Description
10	Pn5PU	This bit is used to enable the internal pull-up resistor of Pn5 pin. 0: Without a pull-up resistor (initial value) 1: With a pull-up resistor The internal pull-up resistor can be enabled on following conditions of the port. It is disabled in other setting. · The input is enabled and the output is disabled on the port (Pn5IE=1, Pn5OE=0) · The input is enabled and the N-channel open drain output is chosen on the port (Pn5IE=1, Pn5OE=1, Pn5OD=1)
9	Pn5OE	This bit is used to enable the output of Pn5 pin 0: Disable the output (initial value) 1: Enable the output
8	Pn5IE	This bit is used to enable the input of Pn5 pin 0: Disable the input (initial value) 1: Enable the input
7 to 4	Pn4MD3 to Pn4MD0	These bits are used to choose the shared function of Pn4 pin. For the details of the shared function, see Table 1-7 "ML62Q1300 Group Pin List", Table 1-8 "ML62Q1500/ML62Q1800 Group Pin List" and Table 1-9 "ML62Q1700 Group Pin List". 0000: Primary function (initial value) 0001: 2nd function 0010: 3rd function 0011: 4th function 0100: 5th function 0101: 6th function 0110: 7th function 0111: 8th function 1XXX: Do not use (Primary function) X: 0 or 1 (don't care)
3	Pn4OD	This bit is used choose the output type of Pn4 pin. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is chosen. See the data sheet for details about the current drive ability. 0: CMOS output (initial value) 1: N-channel open drain output
2	Pn4PU	This bit is used to enable the internal pull-up resistor of Pn4 pin. 0: Without a pull-up resistor (initial value) 1: With a pull-up resistor The internal pull-up resistor can be enabled on following conditions of the port. It is disabled in other setting. · The input is enabled and the output is disabled on the port (Pn4IE=1, Pn4OE=0) · The input is enabled and the N-channel open drain output is chosen on the port (Pn4IE=1, Pn4OE=1, Pn4OD=1)
1	Pn4OE	This bit is used to enable the output of Pn4 pin 0: Disable the output (initial value) 1: Enable the output

Bit No.	Bit symbol name	Description
0	Pn4IE	This bit is used to enable the input of Pn4 pin 0: Disable the input (initial value) 1: Enable the input

[Note]

- **Be sure to set the PnMODm(n=0 to B, m=0 to 7) registers before setting EICON0, EIMOD0 and IE1 registers. If setting the PnMODm register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**

17.2.6 Port n Mode Register 67 (PnMOD67:n=0 to 2, 4 to 9, A, B)

PnMOD67 is a special function register (SFR) to choose the input/output mode, input/output status, and shared function of Pn6 pin and Pn7 pin.
See Table 17-2 "List of Registers / Bits" to check available pins and bits.
Write "0" to the bits of PnMOD67 register that have no corresponding pin.

Address: 0xF208(P0MOD6/P0MOD67), 0xF209(P0MOD7),
0xF218(P1MOD6/P1MOD67), 0xF219(P1MOD7),
0xF228(P2MOD6/P2MOD67), 0xF229(P2MOD7),
0xF248(P4MOD6/P4MOD67), 0xF249(P4MOD7),
0xF258(P5MOD6/P5MOD67), 0xF259(P5MOD7),
0xF268(P6MOD6/P6MOD67), 0xF269(P6MOD7),
0xF278(P7MOD6/P7MOD67), 0xF279(P7MOD7),
0xF288(P8MOD6/P8MOD67), 0xF289(P8MOD7),
0xF298(P9MOD6/P9MOD67), 0xF299(P9MOD7),
0xF2A8(PAMOD6/PAMOD67), 0xF2A9(PAMOD7),
0xF2B8(PBMOD6/PBMOD67), 0xF2B9(PBMOD7)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD67															
Byte	PnMOD7								PnMOD6							
Bit	Pn7MD3	Pn7MD2	Pn7MD1	Pn7MD0	Pn7OD	Pn7PU	Pn7OE	Pn7IE	Pn6MD3	Pn6MD2	Pn6MD1	Pn6MD0	Pn6OD	Pn6PU	Pn6OE	Pn6IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 12	Pn7MD3 to Pn7MD0	These bits are used to choose the shared function of Pn7 pin. For the details of the shared function, see Table 1-7 "ML62Q1300 Group Pin List", Table 1-8 "ML62Q1500/ML62Q1800 Group Pin List" and Table 1-9 "ML62Q1700 Group Pin List". 0000: Primary function (initial value) 0001: 2nd function 0010: 3rd function 0011: 4th function 0100: 5th function 0101: 6th function 0110: 7th function 0111: 8th function 1XXX: Do not use (Primary function) X: 0 or 1 (don't care)
11	Pn7OD	This bit is used choose the output type of Pn7 pin. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is chosen. See the data sheet for details about the current drive ability. 0: CMOS output (initial value) 1: N-channel open drain output

Bit No.	Bit symbol name	Description
10	Pn7PU	This bit is used to enable the internal pull-up resistor of Pn7 pin. 0: Without a pull-up resistor (initial value) 1: With a pull-up resistor The internal pull-up resistor can be enabled on following conditions of the port. It is disabled in other setting. · The input is enabled and the output is disabled on the port (Pn7IE=1, Pn7OE=0) · The input is enabled and the N-channel open drain output is chosen on the port (Pn7IE=1, Pn7OE=1, Pn7OD=1)
9	Pn7OE	This bit is used to enable the output of Pn7 pin 0: Disable the output (initial value) 1: Enable the output
8	Pn7IE	This bit is used to enable the input of Pn7 pin 0: Disable the input (initial value) 1: Enable the input
7 to 4	Pn6MD3 to Pn6MD0	These bits are used to choose the shared function of Pn6 pin. For the details of the shared function, see Table 1-7 "ML62Q1300 Group Pin List", Table 1-8 "ML62Q1500/ML62Q1800 Group Pin List" and Table 1-9 "ML62Q1700 Group Pin List". 0000: Primary function (initial value) 0001: 2nd function 0010: 3rd function 0011: 4th function 0100: 5th function 0101: 6th function 0110: 7th function 0111: 8th function 1XXX: Do not use (Primary function) X: 0 or 1 (don't care)
3	Pn6OD	This bit is used choose the output type of Pn6 pin. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is chosen. See the data sheet for details about the current drive ability. 0: CMOS output (initial value) 1: N-channel open drain output
2	Pn6PU	This bit is used to enable the internal pull-up resistor of Pn6 pin. 0: Without a pull-up resistor (initial value) 1: With a pull-up resistor The internal pull-up resistor can be enabled on following conditions of the port. It is disabled in other setting. · The input is enabled and the output is disabled on the port (Pn6IE=1, Pn6OE=0) · The input is enabled and the N-channel open drain output is chosen on the port (Pn6IE=1, Pn6OE=1, Pn6OD=1)
1	Pn6OE	This bit is used to enable the output of Pn6 pin 0: Disable the output (initial value) 1: Enable the output

Bit No.	Bit symbol name	Description
0	Pn6IE	This bit is used to enable the input of Pn6 pin 0: Disable the input (initial value) 1: Enable the input

[Note]

- **Be sure to set the PnMODm(n=0 to B, m=0 to 7) registers before setting EICON0, EIMOD0 and IE1 registers. If setting the PnMODm register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**

17.2.7 Port n Pulse Mode Register (PnPMD:n=0 to 3)

PnPMD is a special function register (SFR) used when outputting a carrier frequency (pulse output) to the port n.
See Table 17-2 "List of Registers / Bits" to check available pins and bits.
Write "0" to the bits of PnPMD register that have no corresponding pin.

Address: 0xF20A(P0PMDL/P0PMD), 0xF20B(P0PMDH), 0xF21A(P1PMDL/P1PMD), 0xF21B(P1PMDH), 0xF22A(P2PMDL/P2PMD), 0xF22B(P2PMDH), 0xF23A(P3PMDL/P3PMD), 0xF23B(P3PMDH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnPMD															
Byte	PnPMDH								PnPMDL							
Bit	Pn7PLVL	Pn6PLVL	Pn5PLVL	Pn4PLVL	Pn3PLVL	Pn2PLVL	Pn1PLVL	Pn0PLVL	Pn7PEN	Pn6PEN	Pn5PEN	Pn4PEN	Pn3PEN	Pn2PEN	Pn1PEN	Pn0PEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	Pn7PLVL to Pn0PLVL	These bits are used to choose the condition of synchronizing level for outputting the carrier frequency to the pins. 0: Output the carrier frequency to the pins when the output level is "H" (initial value) 1: Output the carrier frequency to the pins when the output level is "L"
7 to 0	Pn7PEN to Pn0PEN	These bits are used to enable or disable the pulse output of Pn7 to Pn0. These bits are valid when the Pn7 to Pn0 pins are configured as the output is enabled (Pn7OE to Pn0OE are "0"). 0: Disable the pulse output (initial value) 1: Enable the pulse output

17.2.8 Port n Pulse Selection Register (PnPSL:n=0 to 3)

PnPSL is a special function register (SFR) used to choose the timer for generating the carrier frequency to the port n.
See Table 17-2 "List of Registers / Bits" to check available pins and bits.
Write "0" to the bits of PnPSL register that have no corresponding pin.

Address: 0xF20C(P0PSLL/P0PSL), 0xF20D(P0PSLH), 0xF21C(P1PSLL/P1PSL), 0xF21D(P1PSLH), 0xF22C(P2PSLL/P2PSL), 0xF22D(P2PSLH), 0xF23C(P3PSLL/P3PSL), 0xF23D(P3PSLH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnPSL															
Byte	PnPSLH								PnPSLL							
Bit	-	-	-	-	-	-	-	-	Pn7PSL	Pn6PSL	Pn5PSL	Pn4PSL	Pn3PSL	Pn2PSL	Pn1PSL	Pn0PSL
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bits
7 to 0	PnmPSL	These bits are used to choose the timer for generating the carrier frequency to the port nm. These bits are valid when the Pnm pins are configured as the output is enabled (PnmOE bits are "0"). n: 0 to 9, A, B m: 0 to 7 0: 16-bit timer 0 output (TMH0OUT) (Initial value) 1: Functional timer 0 output (FTM0P)

17.2.9 PORTXT data input register (PXTDI)

PXTDI is a special function register (SFR) used for reading the level of XT0/XT1 pin.
The level of XT0/PI00 and XT1/PI01 is readable in the input mode.
Set PXT0IE bit and PXT1IE bit of PXTMOD01 register for switching the port to the input mode.
The port is unavailable to use when connecting the crystal resonator.

Address: 0xF2F0(PXTDI)
Access: R
Access size: 8bit
Initial value: Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								PXTDI							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PXT1DI	PXT0DI
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	undefined	undefined

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1	PXT1DI	This bit is used for reading the level of XT1/PI01. 0: The input level of XT1/PI01 pin is “L” 1: The input level of XT1/PI01 pin is “H”
0	PXT0DI	This bit is used for reading the level of XT0/PI00. 0: The input level of XT0/PI00 pin is “L” 1: The input level of XT0/PI00 pin is “H”

[Note]

- **PI00 and PI01 are unavailable to use as input ports when using the crystal resonator for the oscillation clock.**
  Also, PI01 is unavailable to use as an input port when using the PI01 for the external clock input.
  See Chapter 6 “Clock Generation Circuit” for more details on how to use the crystal oscillation or external clock input.

17.2.10 PORTXT mode register 01 (PXTMOD01)

PXTMOD01 is a special function register (SFR) used to choose the input mode of the XT0/PI00 pin and XT1/PI01 pin. The port is unavailable to use when connecting the crystal resonator.

Address: 0xF2F2(PXTMOD0/PXTMOD01), 0xF2F3(PXTMOD1)
Access: R/W
Access size: 8bit/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PXTMOD01															
Byte	PXTMOD1								PXTMOD0							
Bit	-	-	-	-	-	-	-	PXT1IE	-	-	-	-	-	-	-	PXT0IE
R/W	R	R	R	R	R	R	R	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

Bit No.	Bit symbol name	Description
15 to 9	-	Reserved bits
8	PXT1IE	This bit is used to choose the input mode of the XT1/PI01 pin. 0: High impedance that input is disabled (Initial value) 1: Input mode
7 to 1	-	Reserved bits
0	PXT0IE	This bit is used to choose the input mode of the XT0/PI00 pin. 0: High impedance that input is disabled (Initial value) 1: Input mode

17.3 Description of Operation

17.3.1 Input

Each pin of port n sets the PnmIE bit of the PnMODm register to enter the state where input is enabled.
In the state with input enabled, the pin level can be read using the PnDI.
In addition, pull-up can be enabled by setting the PnmPU bit of the PnMODm register.
At a system reset, input disabled and no pull-up are selected as the initial status.
n: Port number 0 to 9, A, B
m: Bit number 0 to 7

17.3.2 Output

Each pin of port n sets the PnmOD bit of the PnMODm register to choose either CMOS output or N-channel open drain output as an output type and sets PnmOE bit of the PnMODm register to enter the state where output is enabled.
In the state with output enabled, "L" or "H" level is output to each pin of the general-purpose port (GPIOn) according to the value set in the PnDO.
At a system reset, output disabled and CMOS output are selected as the initial status.
n: Port number 0 to 9, A, B
m: Bit number 0 to 7

17.3.3 Primary Functions Other than Input/Output Function

External input (EXI0 to EXI11) can be used as the primary function other than the input/output function.
When using EXI0 to EXI11 as external interrupt input and the external clock inputs of the 16-bit timer or external trigger/external clock input of the functional timer, set the PnMODm register of the applicable port to input enabled (PnmIE bit="1").
See Chapter 18 "External Interrupt Control" for external interrupts, Chapter 8 "16-Bit Timer" for external clock input of the 16-bit timer, and Chapter 9 "Functional Timer" for external trigger/external clock input of the functional timer.
n: Port number 0 to 9, A, B
m: Bit number 0 to 7

17.3.4 Shared Function

Each pin of port n can use secondary to octonary functions as the shared function.
Set PnmMD3 to PnmMD0 bits of the PnMODm register to choose each of the secondary to octonary functions.
For the usable shared function, see Table 1-7 "ML62Q1300 Group Pin List", Table 1-8 "ML62Q1500/ML62Q1800 Group Pin List" and Table 1-9 "ML62Q1700 Group Pin List".
n: Port number 0 to 9, A, B
m: Bit number 0 to 7

17.3.5 Carrier Frequency Output

17.3.5.1 Carrier Frequency Output Operation

A carrier frequency signal can be output from port n by setting the PnPMD Register.
See Table 17-2 "List of Registers/Bits" for pins supporting the carrier frequency output function.
The carrier frequency output can be applied to all output mode of supported pins.
For the carrier frequency output, either of 16-bit timer 0 output (TMH0OUT) or functional timer 0 output can be used through setting the PnPSL register.
See Chapter 8 "16-Bit Timer" for details of 16-bit timer 0, and Chapter 9 "Functional Timer" for functional timer 0.
Figures 17-3 and 17-4 show an example of use of the carrier frequency output function.

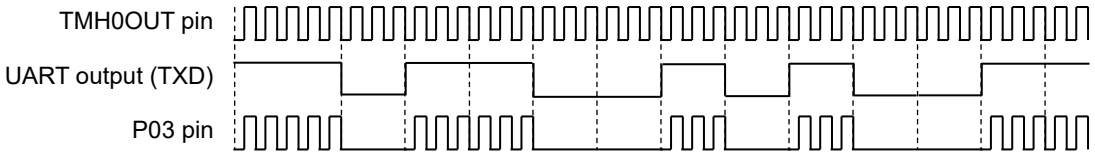


Figure 17-3 Example of Carrier Frequency Output When P03 Pin is Assigned to UART Output Pin
(P03PEN bit ="1", P03PLVL bit= "0" of P0PMD register)

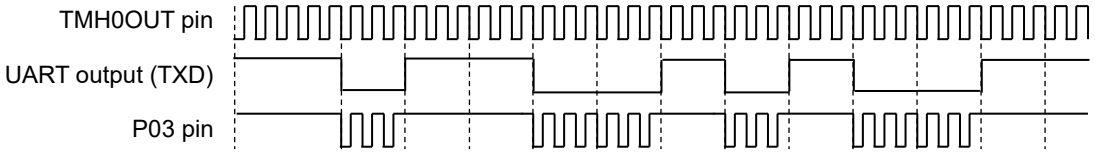


Figure 17-4 Example of Carrier Frequency Output When P03 Pin is Assigned to UART Output Pin
(P03PEN bit ="1", P03PLVL bit= "1" of P0PMD register)

17.3.5.2 Carrier Frequency Output Function Setting Procedure

Figure 17-5 shows an example of the carrier frequency output function setting procedure (with P03 pin used, SU0_TXD0 shared function, functional timer 0 output (FTM0P) used as a timer, carrier frequency output at "L" level).

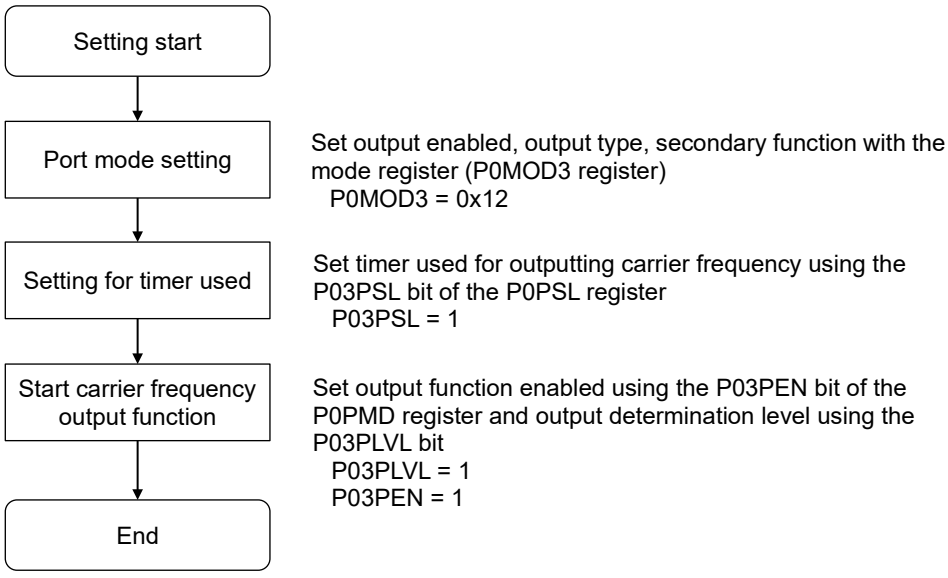


Figure 17-5 Example of Carrier Frequency Output Function Setting Procedure

17.3.6 Port Output Level Test

The level specified in the PnDO can be read from the PnDI by setting the PnmOE bit of the PnMODm register to "1" and the PnmIE bit to "1".
Use of this function allows confirmation that the level set in the PnDO is being normally output to the port.
n: Port number 0 to 9, A, B
m: Bit number 0 to 7

17.3.7 Port Setting Example

Figure 17-6 shows an example for setting port registers to output 0x55 to a port.
It is also available to set output level before outputting.

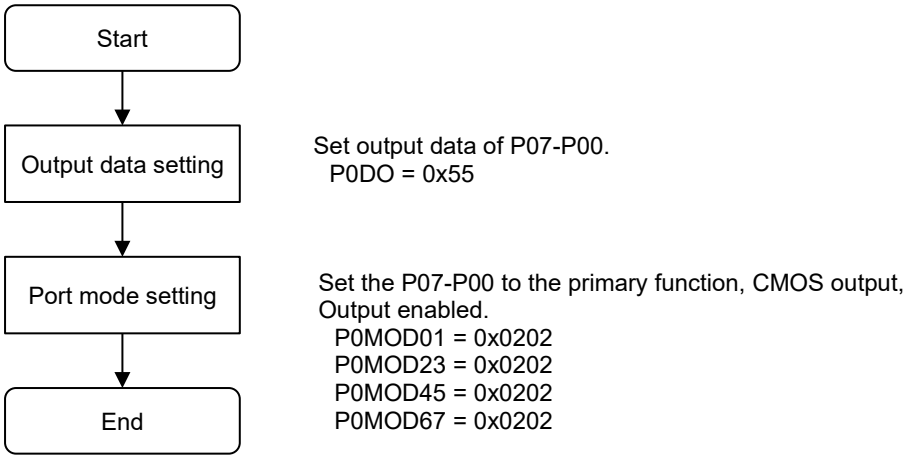


Figure 17-6 Setting example to output data to port 0

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 function.
Confirm following notes in each usage.

17.3.8.1 When using as the general purpose port

When using the general purpose port, P00/TEST0 is unavailable to use as on-chip debug function or ISP function.
Make sure following notes when using the reset function by the RESET_N pin.
- Hold the P00/TEST0 pin to "H" level input or "pull-up" 1ms before and after the RESET_N pin gets to "H" level.

See Chapter 3 "Reset Function" for details about the RESET_N pin reset.
Figure 17-7 shows the sequence.

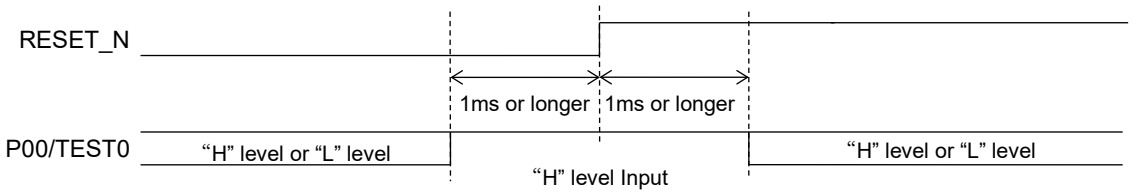


Figure 17-7 P00/TEST0 pin setting when releasing the RESET_N pin to “H” level

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 is unavailable to use as the general purpose port.
- Do not program the software that makes the P00/TEST0 pin output mode.
- Make P00/TEST0 pin able to be connected to V_{DD} with a jumper or something when not using the on-chip debug function and ISP function.

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.

[Note]

- **The P00/TEST0 is initially configured as the input with pull-up resistor. If input "L" level at the initial setting, the input current flows.**

Chapter 18 External Interrupt Function

18. External Interrupt Function

18.1 General Description

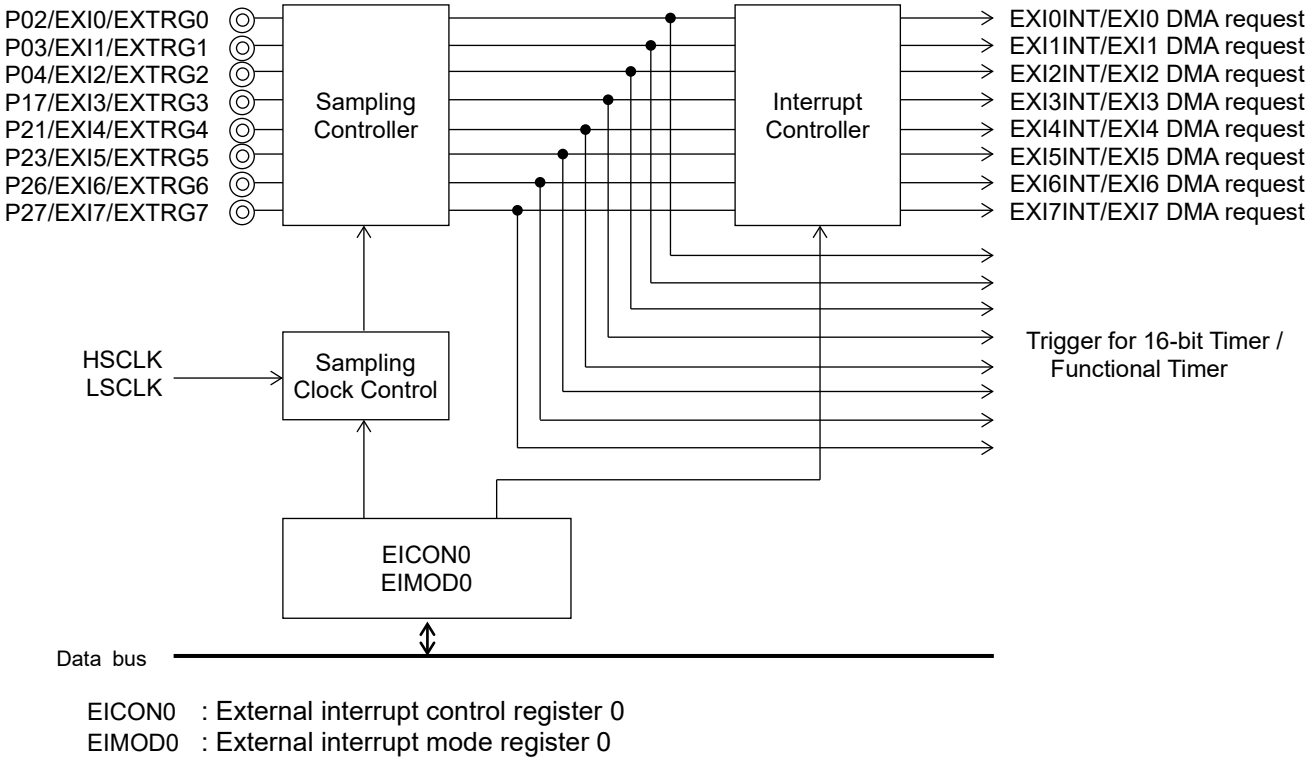
The external interrupt function generates interrupts by signals input to the general ports.
The interrupt channel has each dedicated interrupt vector.
See Chapter 5 "Interrupt" for details of the interrupt vector.
The number of general port with the external interrupt function is dependent of each product. See Table 18-1 " Ports used for the external interrupt and the register settings".

18.1.1 Features

- Maskable nine interrupts (one vector is shared for four external input pins: Expanded external interrupt)
- Available to choose the interrupt mode: interrupt disabled mode, falling-edge interrupt mode, rising-edge interrupt mode or both-edge interrupt mode
- Available to choose "with sampling" or "without sampling" for the input signal (the sampling clock is LSCLK or HSCLK)

18.1.2 Configuration

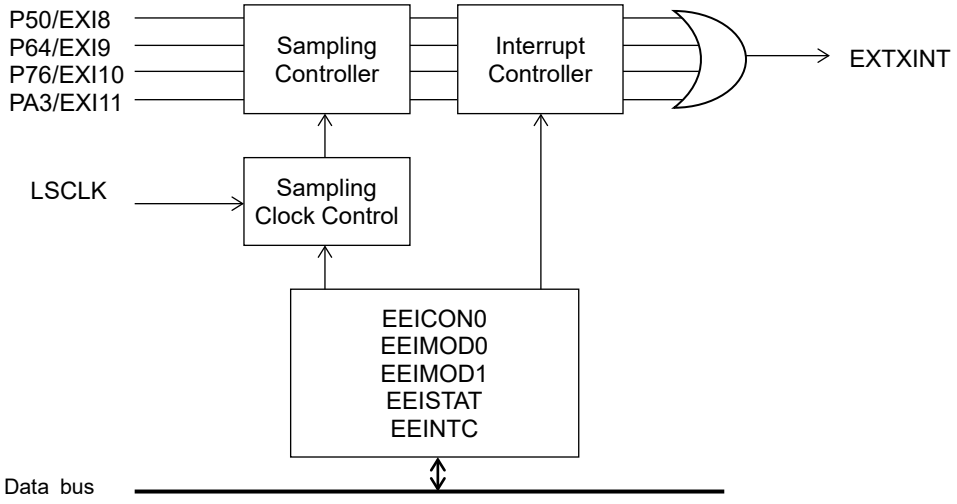
Figure 18-1 shows the configuration of the external interrupt function (EXI0 to EXI7)



EICON0 : External interrupt control register 0
EIMOD0 : External interrupt mode register 0

Figure 18-1 Configuration of External Interrupt Function (EXI0 to EXI7)

Figure 18-2 shows the configuration of the external interrupt function (EXI8 to EXI11)



EEICON0 : Expanded External Interrupt Control Register 0
EEIMOD0 : Expanded External Interrupt Mode Register 0
EEIMOD1 : Expanded External Interrupt Mode Register 1
EEISTAT : Expanded External Interrupt Status Register
EEINTC : Expanded External Interrupt Clear Register

Figure 18-2 Configuration of Expanded External Interrupt Function (EXI8 to EXI11)

18.1.3 List of Pins

The external interrupt is assigned to the primary function of the general port.

Pin name	I/O	Function
EXI0	I	External Interrupt Input 0
EXI1	I	External Interrupt Input 1
EXI2	I	External Interrupt Input 2
EXI3	I	External Interrupt Input 3
EXI4	I	External Interrupt Input 4
EXI5	I	External Interrupt Input 5
EXI6	I	External Interrupt Input 6
EXI7	I	External Interrupt Input 7
EXI8	I	External Interrupt Input 8 (Expanded external interrupt)
EXI9	I	External Interrupt Input 9 (Expanded external interrupt)
EXI10	I	External Interrupt Input 10 (Expanded external interrupt)
EXI11	I	External Interrupt Input 11 (Expanded external interrupt)

Table 18-1 shows the list of the general ports used for the external interrupt and the register settings of the ports.

Table 18-1 Ports used for the external interrupt and the register settings

Pin name	Shared port		Setting register	Setting value	ML62Q1300 group				ML62Q1500/ML62Q1800/ ML62Q1700 group				
					16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
EXI0	P02	Primary Function	P0MOD2	0000_0X01*1	●	●	●	●	●	●	●	●	●
EXI1	P03		P0MOD3	0000_0X01*1	●	●	●	●	●	●	●	●	●
EXI2	P04		P0MOD4	0000_0X01*1	●	●	●	●	●	●	●	●	●
EXI3	P17		P1MOD7	0000_0X01*1	●	●	●	●	●	●	●	●	●
EXI4	P21		P2MOD1	0000_0X01*1	●	●	●	●	●	●	●	●	●
EXI5	P23		P2MOD3	0000_0X01*1	●	●	●	●	●	●	●	●	●
EXI6	P26		P2MOD6	0000_0X01*1	●	●	●	●	●	●	●	●	●
EXI7	P27		P2MOD7	0000_0X01*1	●	●	●	●	●	●	●	●	●
EXI8	P50		P5MOD0	0000_0X01*1	-	-	-	-	●	●	●	●	●
EXI9	P64		P6MOD4	0000_0X01*1	-	-	-	-	●	●	●	●	●
EXI10	P76		P7MOD6	0000_0X01*1	-	-	-	-	-	-	-	●	●
EXI11	PA3		PAMOD3	0000_0X01*1	-	-	-	-	-	-	-	●	●

●: Available -: Unavailable

*1 : "X" determines the condition of the port input

X	Condition of the port input
0	Input (without an internal pull-up resistor)
1	Input (with an internal pull-up resistor)

18.2 Description of Registers

18.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF044	External Interrupt Control register 0	EICON0L	EICON0	R/W	8/16	0x00
0xF045		EICON0H		R/W	8	0x00
0xF046	Reserved	-	-	-	-	-
0xF047						
0xF048	External Interrupt Mode register 0	EIMOD0L	EIMOD0	R/W	8/16	0x00
0xF049		EIMOD0H		R/W	8	0x00
0xF04A 0xF04B 0xF0E0 0xF0E1 0xF0E2 0xF0E3	Reserved	-	-	-	-	-
0xF0E4	Expanded external interrupt control register 0 *1	EEICON0L	EEICON0	R/W	8/16	0x00
0xF0E5		EEICON0H		R/W	8	0x00
0xF0E6	Reserved	-	-	-	-	-
0xF0E7						
0xF0E8	Expanded external interrupt mode register 0 *1	EEIMOD0L	EEIMOD0	R/W	8/16	0x00
0xF0E9		EEIMOD0H		R	8	0x00
0xF0EA	Expanded external interrupt mode register 1 *1	EEIMOD1L	EEIMOD1	R/W	8/16	0x00
0xF0EB		EEIMOD1H		R	8	0x00
0xF0EC	Expanded external interrupt status register *1	EEISTATL	EEISTAT	R	8/16	0x00
0xF0ED		EEISTATH		R	8	0x00
0xF0EE	Expanded external interrupt clear register *1	EEINTCL	EEINTC	W	8/16	0x00
0xF0EF		EEINTCH		W	8	0x00

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

18.2.2 External Interrupt Control Register 0 (EICON0)

EICON0 is a special function register (SFR) used to choose the detection edge of the external interrupt input (EXI0 to EXI7).
Detecting the edge can generate the external interrupt (EXI0INT to EXI7INT).

Address: 0xF044(EICON0L/EICON0), 0xF045(EICON0H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EICON0															
Byte	EICON0H								EICON0L							
Bit	PI7E1	PI6E1	PI5E1	PI4E1	PI3E1	PI2E1	PI1E1	PI0E1	PI7E0	PI6E0	PI5E0	PI4E0	PI3E0	PI2E0	PI1E0	PI0E0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	PI7E1 to PI0E1	These bits are used to choose the detection edge of the external interrupt (EXI0 to EXI7).
7 to 0	PI7E0 to PI0E0	PInE1 PInE0 Description 0 0 Interrupt disabled (Initial value) 0 1 Falling-edge interrupt 1 0 Rising-edge interrupt 1 1 Both-edge interrupt The relation of the bit number and the target external interrupt: Bit 15, 7 (PI7E1, PI7E0) : EXI7INT Interrupt Bit 14, 6 (PI6E1, PI6E0) : EXI6INT Interrupt Bit 13, 5 (PI5E1, PI5E0) : EXI5INT Interrupt Bit 12, 4 (PI4E1, PI4E0) : EXI4INT Interrupt Bit 11, 3 (PI3E1, PI3E0) : EXI3INT Interrupt Bit 10, 2 (PI2E1, PI2E0) : EXI2INT Interrupt Bit 9, 1 (PI1E1, PI1E0) : EXI1INT Interrupt Bit 8, 0 (PI0E1, PI0E0) : EXI0INT Interrupt

18.2.3 External Interrupt Mode Register 0 (EIMOD0)

EIMOD0 is a special function register (SFR) to choose the sampling clock and with/without sampling for the external interrupt (EXI0 to EXI7). Only one sampling clock can be chosen and it is shared for all the interrupt EXI0 to EXI7.

Address: 0xF048(EIMOD0L/EIMOD0), 0xF049(EIMOD0H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EIMOD0															
Byte	EIMOD0H								EIMOD0L							
Bit	-	PG0DIV2	PG0DIV1	PG0DIV0	-	PG0CS0	-	-	PI7SM	PI6SM	PI5SM	PI4SM	PI3SM	PI2SM	PI1SM	PI0SM
R/W	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

Bit No.	Bit symbol name	Description
15	-	Reserved bit
14 to 12	PG0DIV2 to PG0DIV0	These bits are used to choose frequency dividing ratio for the sampling clock in the EXI0 to EXI7. 000: No dividing (Initial value) 001: 1/2 of the sampling clock source 010: 1/4 of the sampling clock source 011: 1/8 of the sampling clock source 100: 1/16 of the sampling clock source 101: 1/32 of the sampling clock source 110: 1/64 of the sampling clock source 111: No dividing
11	-	Reserved bit
10	PG0CS0	This bit is used to choose the sampling clock source in the EXI0 to EXI7. 0: LSCLK (Initial value) 1: HSCLK
9, 8	-	Reserved bits
7 to 0	PI7SM to PI0SM	These bits are used to choose whether the input signals of EXI0 to EXI7 are detected with the sampling clock. 0: Detected without the sampling clock (Initial value) 1: Detected with the sampling clock The relation of the bit number and the target external interrupt: Bit 7 (PI7SM) : EXI7INT Interrupt Bit 6 (PI6SM) : EXI6INT Interrupt Bit 5 (PI5SM) : EXI5INT Interrupt Bit 4 (PI4SM) : EXI4INT Interrupt Bit 3 (PI3SM) : EXI3INT Interrupt Bit 2 (PI2SM) : EXI2INT Interrupt Bit 1 (PI1SM) : EXI1INT Interrupt Bit 0 (PI0SM) : EXI0INT Interrupt

[Note]

- **In the STOP/STOP-D/HALT-H(*1) mode, the sampling clock stops and the sampling function does not work regardless the setting in PI7SM to PI0SM bits of EIMOD0 register. When choosing "with sampling" and entering those mode, there is a time period (*2) in which interrupts get disabled. An unintended interrupt may occur, when returning the program run mode from the HALT-H mode. When entering to those modes, specify the external interrupt as "without sampling".**

  ***1 HALT-H in the case the high-speed clock is chosen**
  ***2 When entering the STOP/STOP-D/HALT-H(*1) mode: Max.30µs. When returning from those modes, the interrupt is disabled until the sampling clock starts to be supplied. The start-up time for supplying clock is dependent of the clock or register settings. For details about it, see Table 4-5 "Wake-up Time from Standby Mode" in the Chapter 4 "Power Management".**

- **An unintended interrupt may occur when switching ENOSC bit of the FCON register, if the HSCLK is chosen for the sampling clock. Choose "without sampling" or disable the external interrupt to prevent an unintended interrupt.**

18.2.4 Expanded External Interrupt Control Register 0 (EEICON0)

EEICON0 is a special function register (SFR) to choose the detection edge of EXI8 to EXI11.

Address: 0xF0E4(EEICON0L/EEICON0), 0xF0E5(EEICON0H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EEICON0															
Byte	EEICON0H								EEICON0L							
Bit	-	-	-	-	EPI3E1	EPI2E1	EPI1E1	EPI0E1	-	-	-	-	EPI3E0	EPI2E0	EPI1E0	EPI0E0
R/W	R	R	R	R	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

Bit No	Bit symbol name	Description
15 to 12, 7 to 4	-	Reserved bits
11 to 8	EPI3E1 to EPI0E1	These bits are used to choose the detection edge of the external interrupt (EXI8 to EXI11).
3 to 0	EPI3E0 to EPI0E0	EPInE1 EPInE0 Description 0 0 Interrupt disabled (Initial value) 0 1 Falling-edge interrupt 1 0 Rising-edge interrupt 1 1 Both-edge interrupt The relation of the bit number and the target external interrupt: Bit 11, 3 (EPI3E1, EPI3E0) : EXI11INT Interrupt Bit 10, 2 (EPI2E1, EPI2E0) : EXI10INT Interrupt Bit 9, 1 (EPI1E1, EPI1E0) : EXI9INT Interrupt Bit 8, 0 (EPI0E1, EPI0E0) : EXI8INT Interrupt

[Note]

- **Re-request interrupt by writing "1" to EEIR bit of EEITNTC register when interrupt request registers (IRQ01, IRQ23, IRQ45, IRQ67) are written by CPU while expanded external interrupt is enabled.**

18.2.5 Expanded External Interrupt Mode Register 0 (EEIMOD0)

EEIMOD0 is a special function register (SFR) to choose the detection with/without sampling for the external interrupt (EXI8 to EXI11).

Address: 0xF0E8(EEIMOD0L/EEIMOD0), 0xF0E9(EEIMOD0H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EEIMOD0															
Byte	EEIMOD0H								EEIMOD0L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	EPI3SM	EPI2SM	EPI1SM	EPI0SM
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 4	-	Reserved bits
3 to 0	EPI3SM to EPI0SM	These bits are used to specify the input signals of EXI8 to EXI11 are detected with/without sampling. Set always this bit to "1". 0: Detected without the sampling clock (Initial value) 1: Detected with the sampling clock The relation of the bit number and the target external interrupt: Bit 3 (EPI3SM) : EXI11INT Interrupt Bit 2 (EPI2SM) : EXI10INT Interrupt Bit 1 (EPI1SM) : EXI9INT Interrupt Bit 0 (EPI0SM) : EXI8INT Interrupt

[Note]

- **In the STOP/STOP-D mode, the sampling clock stops and the sampling function does not work regardless the setting in EPI3SM to EPI0SM bits of EEIMOD0 register. There is a time period[*1] in which interrupts get disabled.**
  [*1] When entering the STOP/STOP-D mode: Max.30µs. When returning from the STOP/STOP-D mode, the interrupt is disable until the sampling clock starts to be supplied. The start-up time for supplying clock is dependent of the clock or register settings. For details about it, see Table 4-5 "Wake-up Time from Standby Mode" in the Chapter 4 "Power Management".

18.2.6 Expanded External Interrupt Mode Register 1(EEIMOD1)

EEIMOD1 is a special function register (SFR) to choose dividing ratio of the sampling clock for the external interrupt (EXI8 to EXI11).

Address: 0xF0EA(EEIMOD1L/EEIMOD1), 0xF0EB(EEIMOD1H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EEIMOD1															
Byte	EEIMOD1H								EEIMOD1L							
Bit	-	-	-	-	-	-	-	-	-	EPG0DIV2	EPG0DIV1	EPG0DIV0	-	rsvd	-	-
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 7	-	Reserved bits
6 to 4	EPG0DIV2 to EPG0DIV0	These bits are used to choose frequency dividing ratio for the sampling clock in the EXI8 to EXI11. 000: No dividing (Initial value) 001: 1/2 of the sampling clock source 010: 1/4 of the sampling clock source 011: 1/8 of the sampling clock source 100: 1/16 of the sampling clock source 101: 1/32 of the sampling clock source 110: 1/64 of the sampling clock source 111: Do not use (No dividing)
3	-	Reserved bit
2	rsvd	Reserved bit. Write "0" to this bit.
1, 0	-	Reserved bits

18.2.7 Expanded External Interrupt Status Register (EEISTAT)

EEISTAT is a special function register (SFR) used to indicate the expanded external interrupt status.
The EEISTAT is the read-only register. The EEI3S bit to EEI0S bit is reset to "0" by writing "1" to the same number of bit in the EEINTC register.

Address: 0xF0EC(EEISTATL/EEISTAT), 0xF0ED(EEISTATH)
Access: R
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EEISTAT															
Byte	EEISTATH								EEISTATL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	EEI3S	EEI2S	EEI1S	EEI0S
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 4	-	Reserved bits
3 to 0	EEI3S to EEI0S	These bits are used to indicate the expanded external interrupt status. 0: No interrupt occurred (Initial value) 1: Interrupt occurred The relation of the bit number and the target external interrupt: Bit 3 (EEI3S) : EXI11INT Interrupt Bit 2 (EEI2S) : EXI10INT Interrupt Bit 1 (EEI1S) : EXI9INT Interrupt Bit 0 (EEI0S) : EXI8INT Interrupt

[Note]

- **When debugging the program on the debugger, remain enabling "External Interrupt" check box on "Peripheral Circuit" tab in the Operation setting menu. If uncheck the option, these status bits might get cleared.**
- **No interrupt request is issued if the interrupt status bit is "1" and the same interrupt occurs again. To issue an interrupt request, write "1" to the same bit in the EEINTC register and clear the status bit to "0".**

18.2.8 Expanded External Interrupt Clear Register (EEINTC)

EEINTC is a special function register (SFR) used to clear the expanded external interrupt status.
The EEI3C bit to EEI0C bit is set to "1", the interrupt request indicated by the same number of bit in the EEISTAT register gets cleared.
The EEINTC always returns "0x0000" for reading.

Address: 0xF0EE(EEINTCL/EEINTC), 0xF0EF(EEINTCH)
Access: W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EEINTC															
Byte	EEINTCH								EEINTCL							
Bit	EEIR	-	-	-	-	-	-	-	-	-	-	-	EEI3C	EEI2C	EEI1C	EEI0C
R/W	W	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	EEIR	This bit indicates the request of the expanded external interrupt. Write "1" to this bit before returning from the interrupt routine and after writing IRQ01/IRQ23/IRQ45/IRQ67 register. Writing "0": Invalid Writing "1": If there is any unhandled interrupt source, the interrupt request is generated again.
14 to 4	-	Reserved bits
3 to 0	EEI3C to EEI0C	If writing "1" to these bits, it clears the target bits of interrupt status. Writing "0" to these bits does not clear the interrupt status. The relation of the bit number and the target external interrupt: Bit 3 (EEI3C) : EXI11INT Interrupt Bit 2 (EEI2C) : EXI10INT Interrupt Bit 1 (EEI1C) : EXI9INT Interrupt Bit 0 (EEI0C) : EXI8INT Interrupt

[Note]
- **Do not set EEIR bit and EEI3C to EEI0C bits simultaneously.**

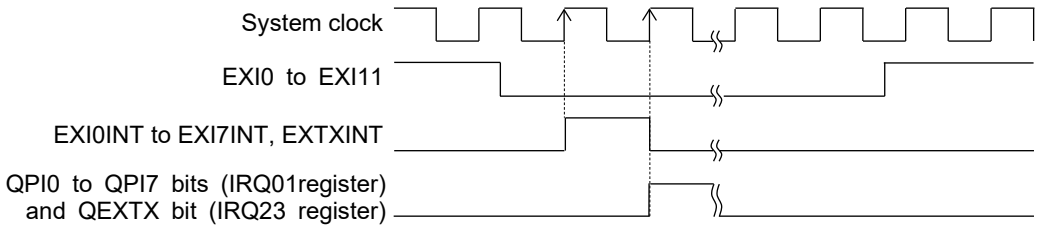
18.3 Description of Operation

18.3.1 Interrupt Request Timing

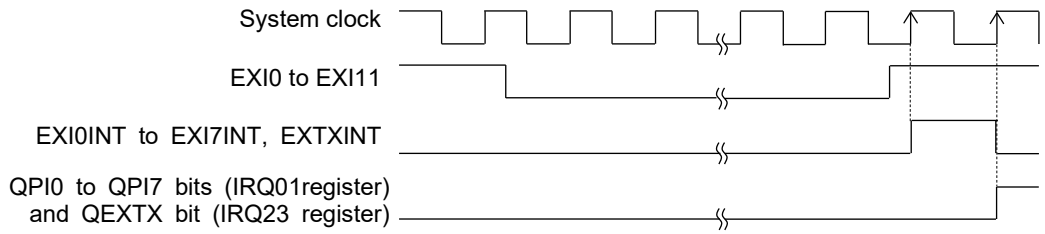
Figure 18-3 shows the interrupt generation timing without sampling (when the rising-edge/falling-edge/both-edge interrupt mode is chosen). Figure 18-4 shows the interrupt generation timing with sampling (when the rising-edge interrupt mode is chosen).
Table 18-2 shows the difference between the external interrupt generation timings with or without sampling after detection of the edge.

Table 18-2 EXI0INT to EXI11INT Generation After Detection of Edge of EXI0 to EXI11

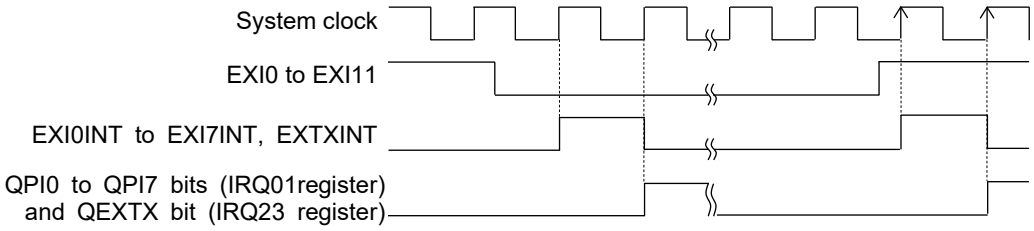
Sampling	Generation timing
No	Generated in synchronization with system clock
Yes	Generated in synchronization with system clock, when no transition for three periods with sampling clock after detecting edge.



(a) When falling-edge interrupt mode is chosen



(b) When rising-edge interrupt mode is chosen



(c) When both-edge interrupt mode is chosen

Figure 18-3 External Interrupt Generation Timing (without Sampling)

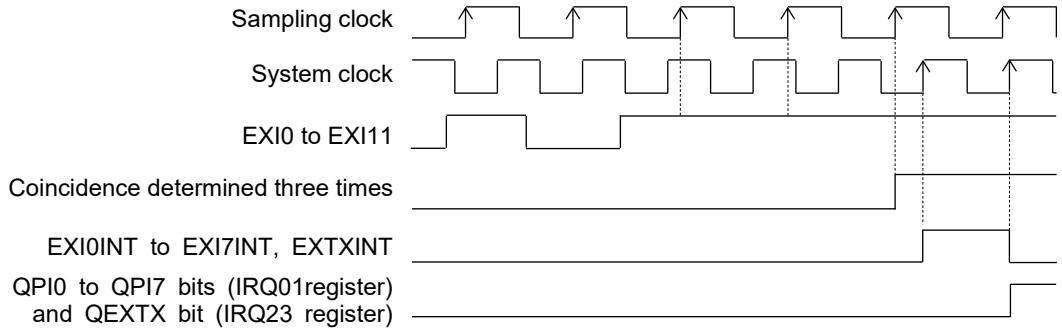
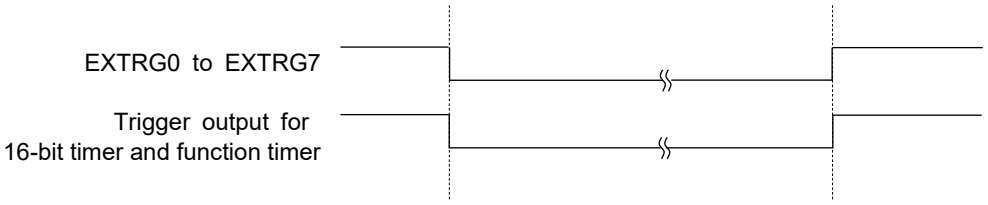


Figure 18-4 External Interrupt Generation Timing (with Sampling, with Rising-edge Interrupt Mode Chosen)

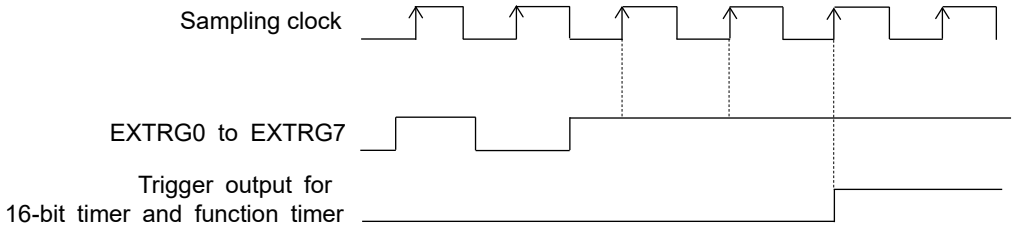
18.3.2 External Trigger Signal

Pins assigned with external interrupt can be used as external trigger signals (EXTRG0 to EXTRG7) for the 16-bit timer and function timer.
In addition, the sampling function contained in the external interrupt function can be used.

Figure 18-5 shows the external trigger signal timing.



(a) Without sampling



(b) With sampling

Figure 18-5 16-Bit Timer and Functional Timer Trigger Signal

18.3.3 External Interrupt Setting Flow

Figure 18-6 shows the external interrupt setting flow.

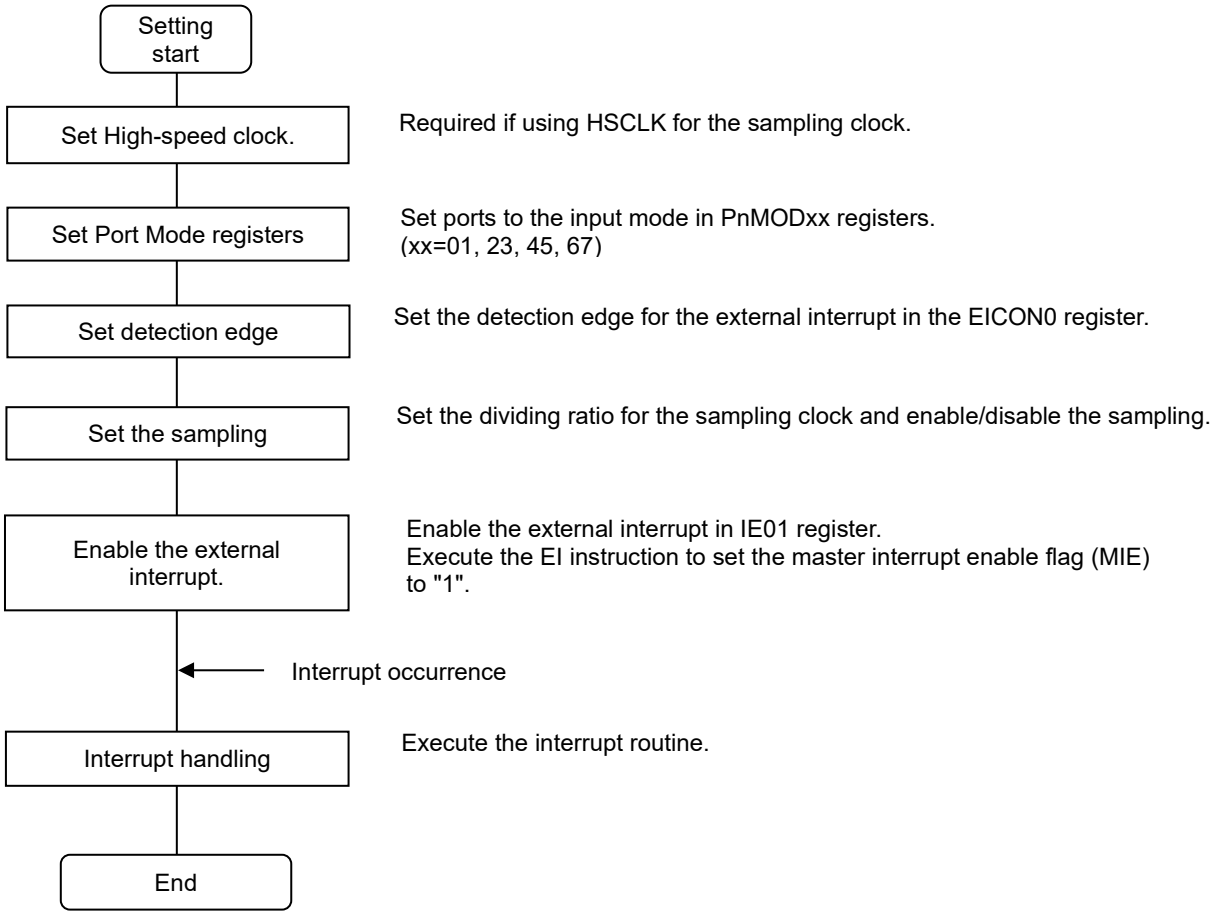
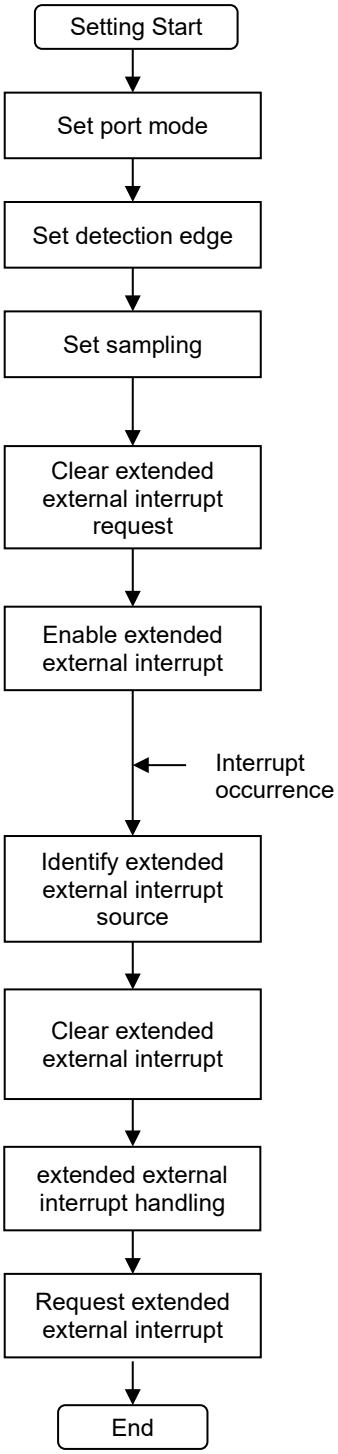


Figure 18-6 External Interrupt Setting Flow

18.3.4 Expanded External Interrupt Setting Flow

Figure 18-7 shows the expanded external interrupt setting flow.



Set ports to the input mode in PnMODxx registers.
(xx=01, 23, 45, 67)

Set the detection edge for the extended external register in EEICON0 register.

Choose "with sampling" in EEIMOD0L register.
EEIMOD0L=0x0F;

Set the dividing ratio of the sampling clock in the EEIMOD1L register.

Clear QEXTX bit .
QEXTX=0;

Enable the extended external interrupt by EEXTX bit.
EEXTX=1;

Execute the EI instruction to set the master interrupt enable flag (MIE) to "1".

Read EEISTAT register to identify the extended external interrupt source.

Write “1” to the bits in EEINTCL register to clear the extended external interrupt source.

Execute the interrupt routine.

Write 0x80 to EEINTCH register to request the extended external interrupt source (if other expanded external interrupts are remained, they would be accepted to occur the interrupt again）.
EEINTCH=0x80;

*1:Re-request interrupt by writing "1" to EEIR bit of EEITNTC register when interrupt request registers (IRQ01, IRQ23, IRQ45, IRQ67) are written by CPU while expanded external interrupt is enabled.

Figure 18-7 External Interrupt Setting Flow

Chapter 19 CRC Calculator

19. CRC Calculator

19.1 General Description

ML62Q1000 series has the CRC (Cyclic 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, available in HALT mode or HALT-H mode.

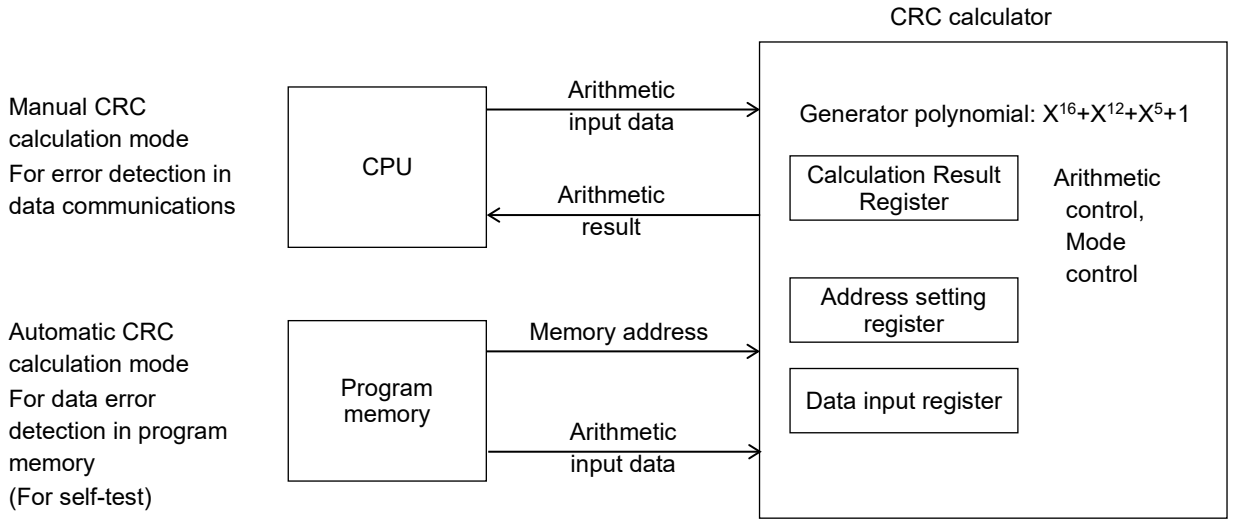


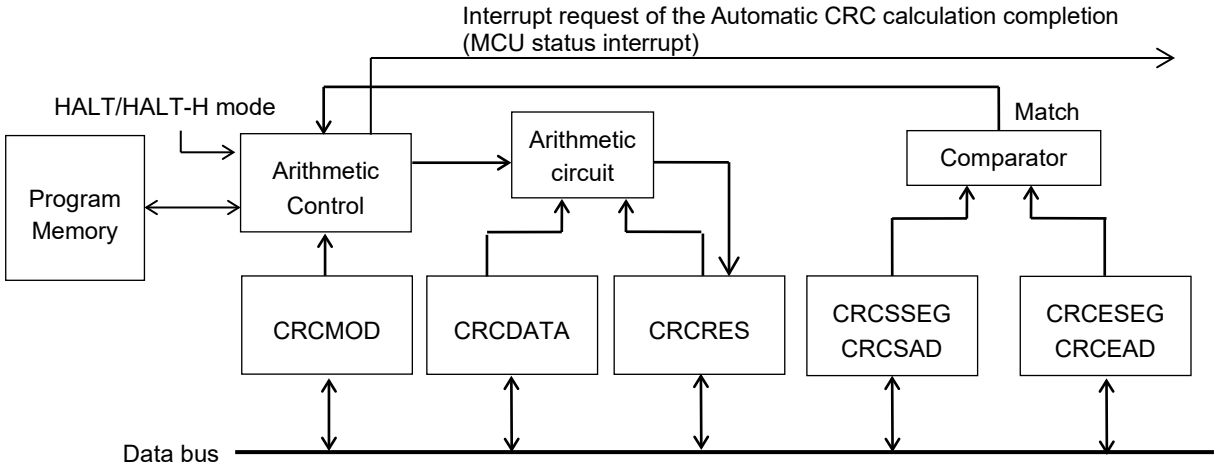
Figure 19-1 CRC generator overview

19.1.1 Features

- Manual CRC calculation mode
  Generates CRC data from data set in CRC calculation register by the software
  Calculation unit is 8bit
- Automatic CRC calculation mode
  Automatic CRC calculation by the hardware to check data in program memory in HALT or HALT-H mode and generates CRC data
  Calculation unit is 32bit. The interrupt occurs when the arithmetic operation is completed
- Generator polynomial: $X^{16}+X^{12}+X^{5}+1$
- MSB first or LSB first selectable

19.1.2 Configuration

Figure 19-2 shows the configuration of the CRC generator.



CRCMOD : CRC Calculation Mode Register
CRCDATA : CRC Calculation Data Register
CRCRES : CRC Calculation Result Register
CRCSSEG : Automatic CRC Calculation Start Segment Setting Register
CRCESEG : Automatic CRC Calculation End Segment Setting Register
CRCSAD : Automatic CRC Calculation Start Address Setting Register
CRCEAD : Automatic CRC Calculation End Address Setting Register

Figure 19-2 Configuration of CRC Generator

19.2 Description of Registers

19.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
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 Calculation 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	CRC Calculation Mode Register	CRCMOD	-	R/W	8	0x00
0xF0DD	Reserved	-	-	-	-	-

19.2.2 Automatic CRC Calculation Start Address Setting Register (CRCSAD)

CRCSAD is a special function register (SFR) used to set the start address of automatic CRC calculation. This register is incremented during the automatic CRC calculation mode.

Address: 0xF0D0
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CRCSAD															
Byte	CRCSADH								CRCSADL							
Bit	CRCSAD15	CRCSAD14	CRCSAD13	CRCSAD12	CRCSAD11	CRCSAD10	CRCSAD9	CRCSAD8	CRCSAD7	CRCSAD6	CRCSAD5	CRCSAD4	CRCSAD3	CRCSAD2	-	-
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	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

- **Write to the CRCSAD register when CRCAEN bit of the CRCMOD register is "0". Any writing is ignored when the CRCAEN bit is "1".**
- **Automatic CRC calculation is four-byte length. Generate an expected value by four bytes. Writing to the bit1 and bit0 are ignored; they are fixed to "0" internally during the calculation.**
- **Do not specify segment or address out of program code area. See section 2.5 "Program Memory Space" for details of the program code area.**

19.2.3 Automatic CRC Calculation End Address Setting Register (CRCEAD)

CRCEAD is a special function register (SFR) used to set the end address of automatic CRC calculation.

Address: 0xF0D2
Access: R/W
Access size: 8/16bit
Initial value: 0xFFFC

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CRCEAD															
Byte	CRCEADH								CRCEADL							
Bit	CRCEAD15	CRCEAD14	CRCEAD13	CRCEAD12	CRCEAD11	CRCEAD10	CRCEAD9	CRCEAD8	CRCEAD7	CRCEAD6	CRCEAD5	CRCEAD4	CRCEAD3	CRCEAD2	-	-
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	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

[Note]

- **Write to the CRCEAD register when CRCAEN bit of the CRCMOD register is "0". Any writing is ignored 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; they are fixed to "1" internally during the calculation.**
- **Do not specify segment or address out of program code area. See section 2.5 "Program Memory Space" for details of the program code area.**

19.2.4 Automatic CRC Calculation Start Segment Setting Register (CRCSSEG)

CRCSSEG is a special function register (SFR) used to set the start segment of automatic CRC calculation. This register is incremented during the automatic CRC calculation mode.

Address: 0xF0D4
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								CRCSSEG							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	CRCSSEG3	CRCSSEG2	CRCSSEG1	CRCSSEG0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

- **Write to the CRCSSEG register when CRCAEN bit of the CRCMOD register is "0". Any writing is ignored when the CRCAEN bit is "1".**
- **Do not specify segment or address out of program code area. See section 2.5 "Program Memory Space" for details of the program code area.**

19.2.5 Automatic CRC Calculation End Segment Setting Register (CRCESEG)

CRCESEG is a special function register (SFR) used to set the end segment of automatic CRC calculation.

Address: 0xF0D6
Access: R/W
Access size: 8bit
Initial value: 0xFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								CRCESEG							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	CRCESEG3	CRCESEG2	CRCESEG1	CRCESEG0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

[Note]

- **Write to the CRCESEG register when CRCAEN bit of the CRCMOD register is "0". Any writing is ignored when the CRCAEN bit is "1".**
- **Do not specify segment or address out of program code area. See section 2.5 “Program Memory Space” for details of the program code area.**

19.2.6 CRC Calculation Data Register (CRCDATA)

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).

Address: 0xF0D8
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								CRCDATA							
Bit	-	-	-	-	-	-	-	-	CRCDATA7	CRCDATA6	CRCDATA5	CRCDATA4	CRCDATA3	CRCDATA2	CRCDATA1	CRCDATA0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

- **Write to the CRCDATA register when CRCAEN bit of the CRCMOD register is "0". Any writing is ignored when the CRCAEN bit is "1".**

19.2.7 CRC Calculation Result Register (CRCRES)

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.

Address: 0xF0DA
Access: R/W
Access size: 8/16bit
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CRCRES															
Byte	CRCRESH								CRCRESL							
Bit	CRCRES15	CRCRES14	CRCRES13	CRCRES12	CRCRES11	CRCRES10	CRCRES9	CRCRES8	CRCRES7	CRCRES6	CRCRES5	CRCRES4	CRCRES3	CRCRES2	CRCRES1	CRCRES0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

[Note]

- **Write to the CRCRES register when CRCAEN bit of the CRCMOD register is "0". Any writing is ignored when the CRCAEN bit is "1".**

19.2.8 CRC Calculation Mode Register (CRCMOD)

CRCMOD is a special function register (SFR) used to control the CRC calculation mode.

Address: 0xF0DC
Access: R/W
Access size: 8bit
Initial value: 0x00

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

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bit
1	CRCDIR	This bit is used to choose the shift direction of the CRC calculation. 0: LSB first (Initial value) 1: MSB first
0	CRCAEN	This bit is used to enable the automatic CRC calculation mode. If entering the HALT/HALT-H mode when the CRCAEN bit is "1", the CRC calculation starts for the program code area in the range specified by the CRCSSEG and CRCESEG register and CRCSAD and CRCEAD register. When CRC calculation is completed, the CRCAEN is reset to "0.", also the CRC calculation completion interrupt is generated. See Chapter 29 “Safety Function” for details of the automatic CRC calculation completion interrupt. 0: Disables the automatic CRC calculation mode (Initial value) 1: Enables the automatic CRC calculation mode

[Note]

- **When the CPU operation mode is "Wait mode" and the PLL reference frequency is 24MHz, choose 12MHz or slower as the SYSTEMCLK before entering the HALT/HALT-H mode.**

19.3 Description of Operation

Two modes are available for the CRC calculator: manual CRC calculation mode and automatic CRC calculation mode.

- Manual CRC Calculation Mode
  CRC calculation is executed by hardware as needed through writing data to the CRC calculation register by software.
  Calculation unit: 8-bit.
- Automatic CRC Calculation Mode
  In the HALT/HALT-H mode, data in the program memory area is automatically CRC-calculated by hardware.
  Calculation unit: 32-bits with the interrupt generated when the automatic CRC calculation is completed.

19.3.1 Manual CRC Calculation Mode

In the manual CRC calculation mode, the calculation result is outputted to the CRC calculation result register (CRCRES) by writing the initial value to the 16-bit CRC calculation result register (CRCRES) then writing data to 8-bit CRC calculation data register (CRCDATA). For data error detection in serial communication, etc., presence of errors can be detected by transferring data with the calculation result attached when transmission and performing the same CRC calculation in the reception side.

19.3.1.1 Example of Use of Manual CRC Calculation Mode

The following chart shows the process flow of serial transmission with the CRC calculation result attached to data.

In this example, 11-byte data with 0x21 in the beginning is used as transmit data, and calculation result is obtained using the calculation shift direction in LSB first mode.
Transmission and CRC calculation data: 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x81, 0x7F

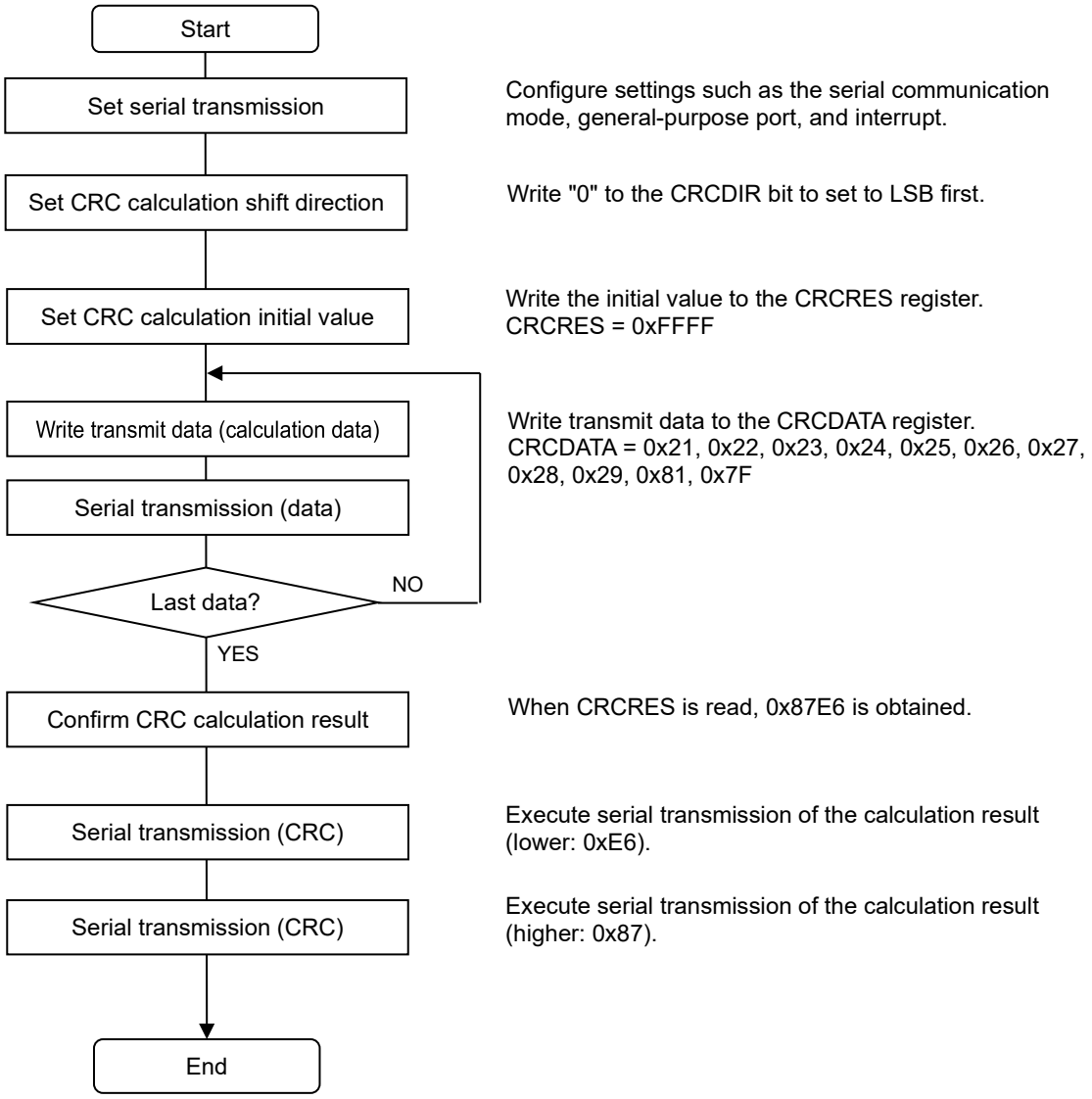


Figure 19-3 CRC Calculation Processing Flow 1 (Serial Transmission/LSB First)

The following chart shows the CRC calculation process flow with the CRC calculation result attached to the serial receive data.
In this example, 13-byte data with 0x21 in the beginning is used as calculation data. From the calculation data, calculation result is obtained using the calculation shift direction in LSB first mode. The first 11 bytes of the CRC calculation result is added to the last two bytes.

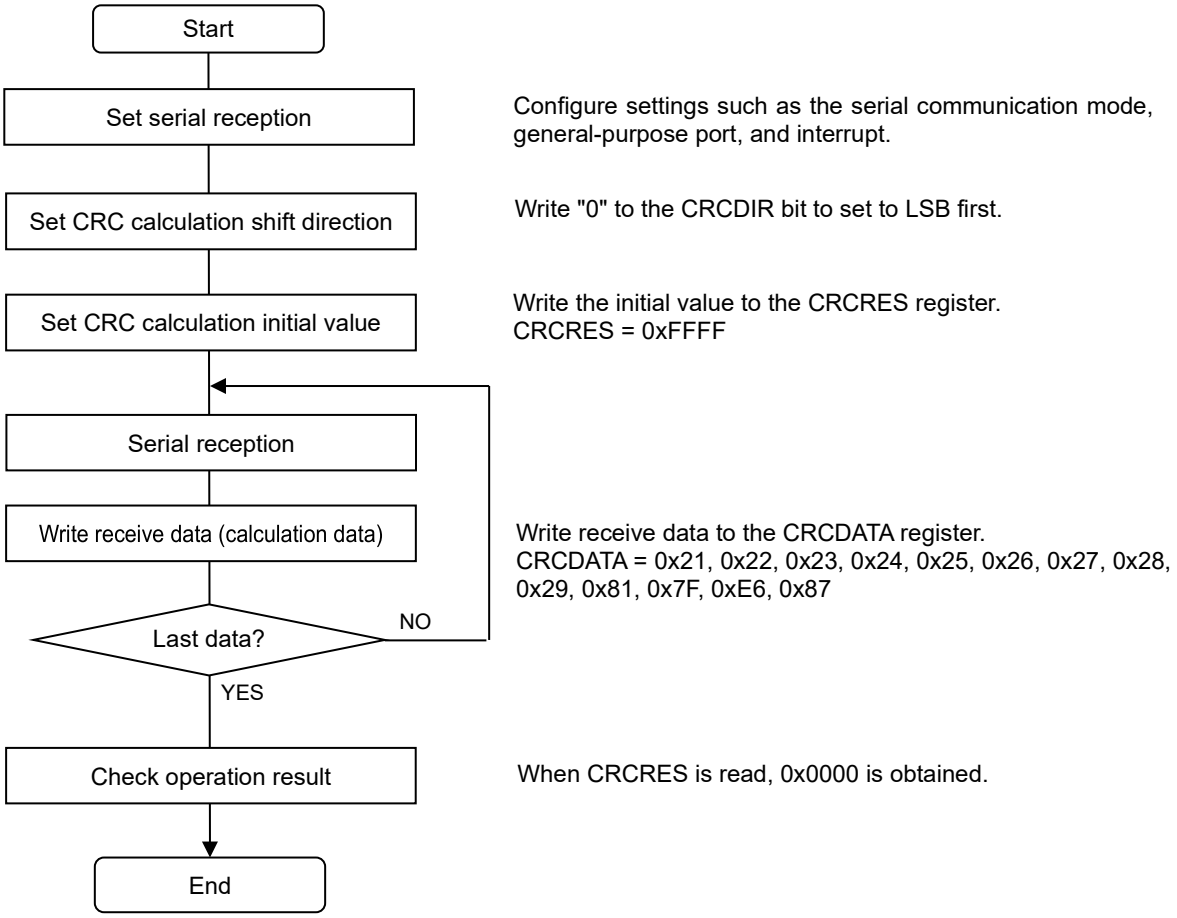


Figure 19-4 CRC Calculation Processing Flow 2 (Serial Reception/LSB First)

The following chart shows the process flow of serial transmission with the CRC calculation result attached to data.

In this example, 11-byte data with 0x21 in the beginning is used as transmit data, and calculation result is obtained using the calculation shift direction in MSB first mode.
Transmission and CRC calculation data: 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x81, 0x7F

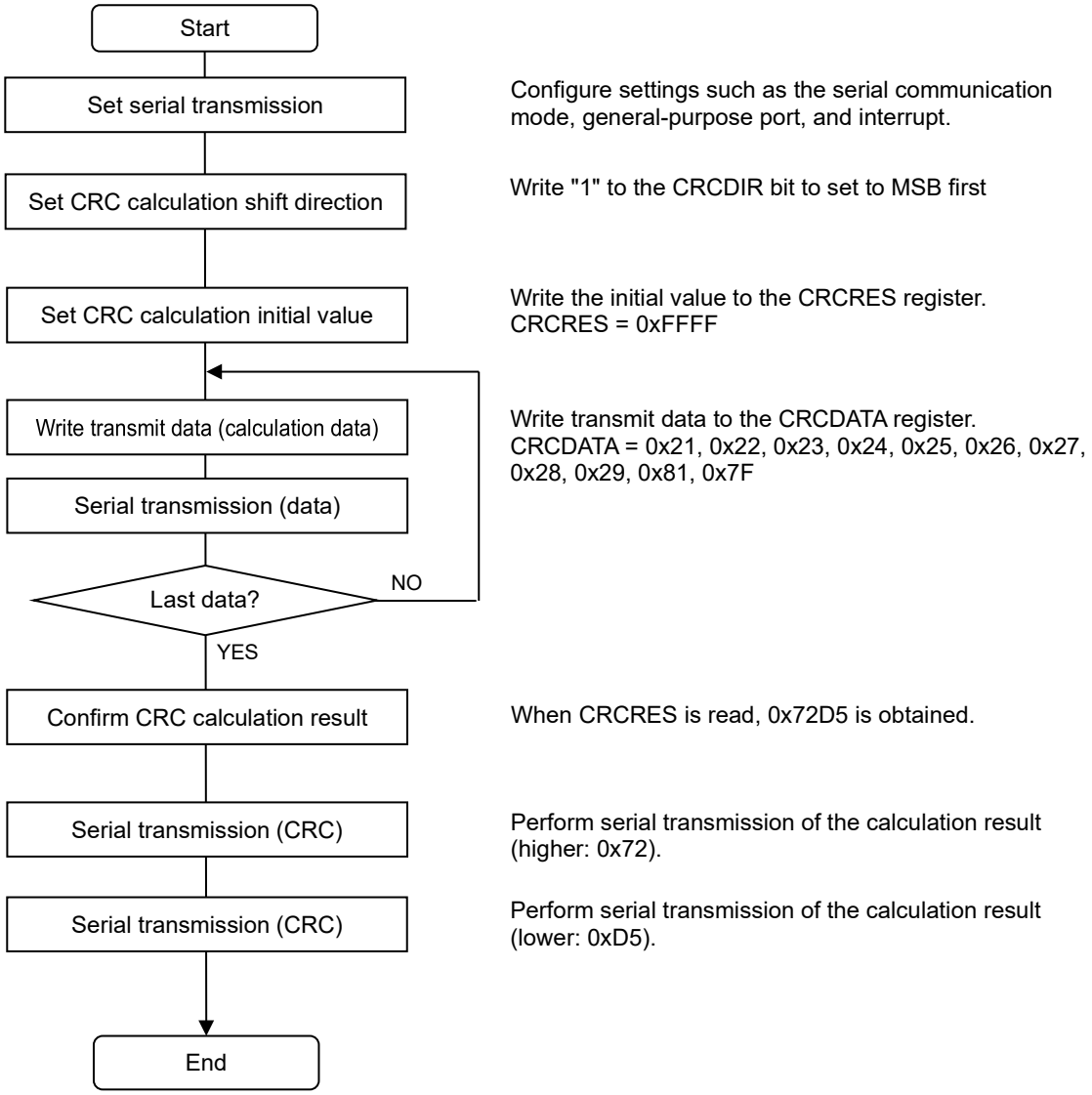


Figure 19-5 CRC Calculation Processing Flow 3 (Serial Transmission/MSB First)

The following chart shows the CRC calculation process flow with the CRC calculation result attached to the serial receive data.

In this example, 13-byte data with 0x21 in the beginning is used as calculation data. From the calculation data, calculation result is obtained using the calculation shift direction in MSB first mode. The first 11 bytes of the CRC calculation result is added to the last two bytes.

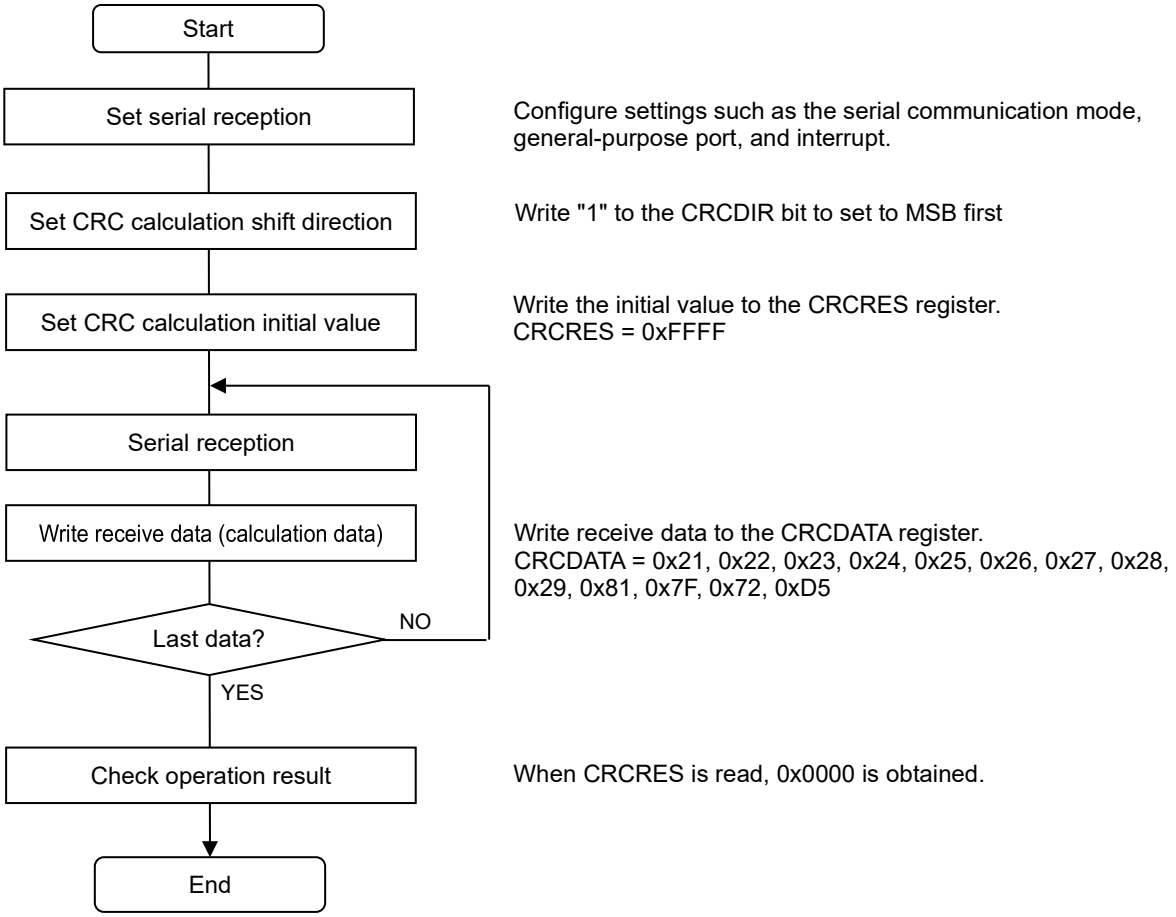


Figure 19-6 CRC Calculation Processing Flow 4 (Serial Reception/MSB First)

19.3.1.2 Operation Timing Chart in Manual CRC Calculation Mode

Set the initial value of CRC calculation in the CRCRES register. When 8-bit data is written to the CRCDATA register, the calculation result is stored in the CRCRES register on the next clock rising-edge. The CRC calculation result can be checked anytime by reading the CRCRES register.
Figure 19-7 shows the operation timing chart of CRC calculation.

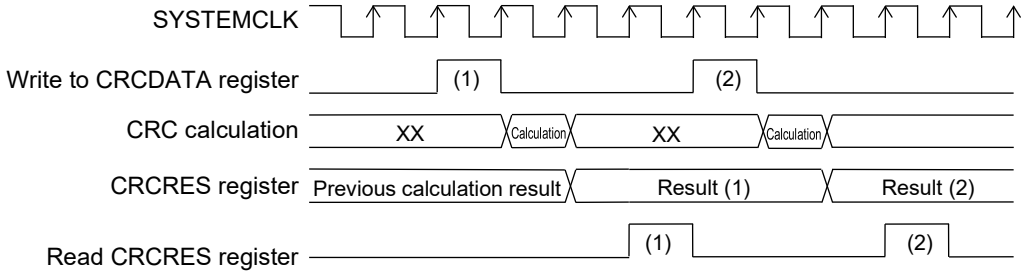


Figure 19-7 Timing Chart of CRC Calculation.

19.3.2 Automatic CRC Calculation Mode

In the automatic CRC calculation mode, an arbitrary program memory area is automatically CRC-calculated in the HALT/HALT-H mode and the result is output to the CRC calculation result register (CRCRES).
For data error detection in program memory (for self-test), using software, the result of the automatic calculation can be compared with the expected value written to Flash memory in advance.
The expected value is created in the generation tool of the ROM code data from LAPIS Technology.

19.3.2.1 Example of Use of Automatic CRC Calculation Mode

The following chart shows the automatic CRC calculation process flow.

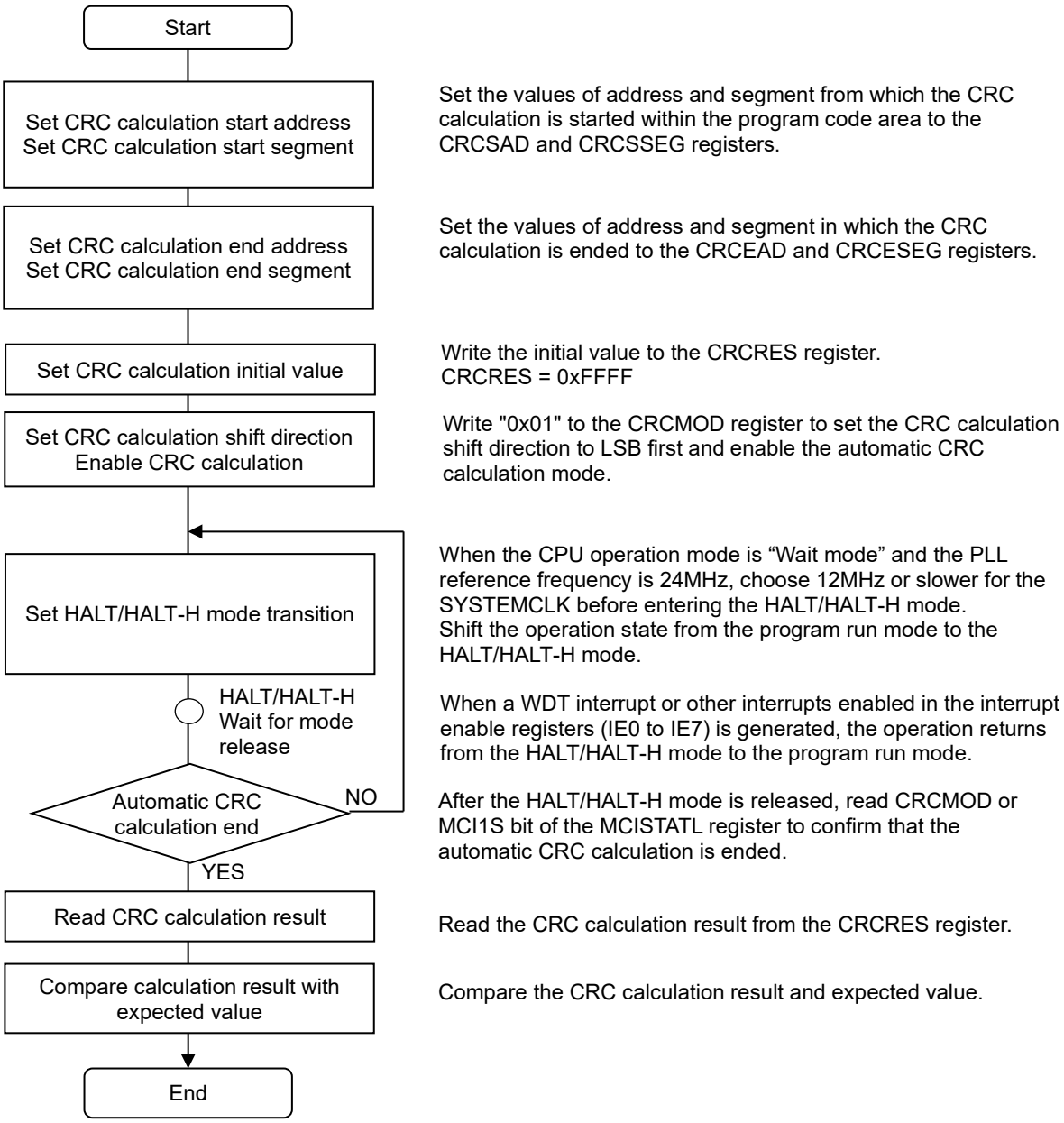


Figure 19-8 Automatic CRC Calculation Process Flow

The CRC calculation of data in the program code area configured in the CRCSSEG, CRCSAD, CRCESEG, and CRCEAD registers is started when entering to the HALT/HALT-H mode, if the CRCAEN bit of CRCMOD register is "1".

When the HALT/HALT-H mode released while the calculation is in progress, the calculation is aborted. If shifting to the HALT/HALT-H mode again, the calculation resumes at the address it was aborted. The CRCSSEG and CRCAD registers are incremented each time data is read from the program code area.

If the calculation start segment and address (values of CRCSSEG and CRCSAD registers) match the calculation end segment and address (values of CRCESEG and CRCEAD registers), the CRC calculation is ended, the CRCAEN bit becomes "0", and the automatic CRC calculation completion interrupt request is generated. If the automatic CRC calculation completion interrupt is enabled, then the HALT/HALT-H mode is released and the MCU status interrupt is generated.

To enable/disable the automatic CRC calculation completion interrupt is set by the MCU status interrupt enable register (MCINTEL). See Chapter 29 "Safety Function" for details of the MCINTEL register.

See "ML62Q1000 Series Self-test Sample Software AP Notes" and a manual of the generation tool of the ROM code data for details of self-test program using the automatic CRC calculation mode or how to generate expected values.

[Note]

- **To perform CRC calculation in the manual mode when automatic CRC calculation is not completed, save the value in the CRCRES register before calculation. Once the CRC calculation in the manual mode is completed, move the saved value back to the CRCRES register and set the CRCAEN bit to "1". If entering the HALT/HALT-H mode, then the automatic CRC calculation can be restarted.**
  The final addresses at the end of the previous operation are stored in the CRCSAD and CRCSSEG registers. If values in the CRCSAD and CRCSSEG registers are overwritten with the CRCAEN bit set to "0", the calculation works incorrectly.
- **When the CPU operation mode is "Wait mode" and the PLL reference frequency is 24MHz, choose 12MHz or slower for the SYSTEMCLK before entering the HALT/HALT-H mode.**

Chapter 20 Analog Comparator

20. Analog Comparator

20.1 General Description

The Analog Comparator enables to use following functions.

- Compare voltages input to the two pins
- Compare a voltage input to the one pin with the internal reference voltage (Approx. 0.8V)

Table 20-1 shows the number of channels.

Table 20-1 Number of Analog Comparator channels

Channel no.	ML62Q1300 group				ML62Q1500/ML62Q1800/ML62Q1700 group				
	16pin product	20pin product	24 pin product	32 pin product	48pin product	52pin product	64pin product	80pin product	100pin product
0	●	●	●	●	●	●	●	●	●
1	−	−	−	−	●	●	●	●	●

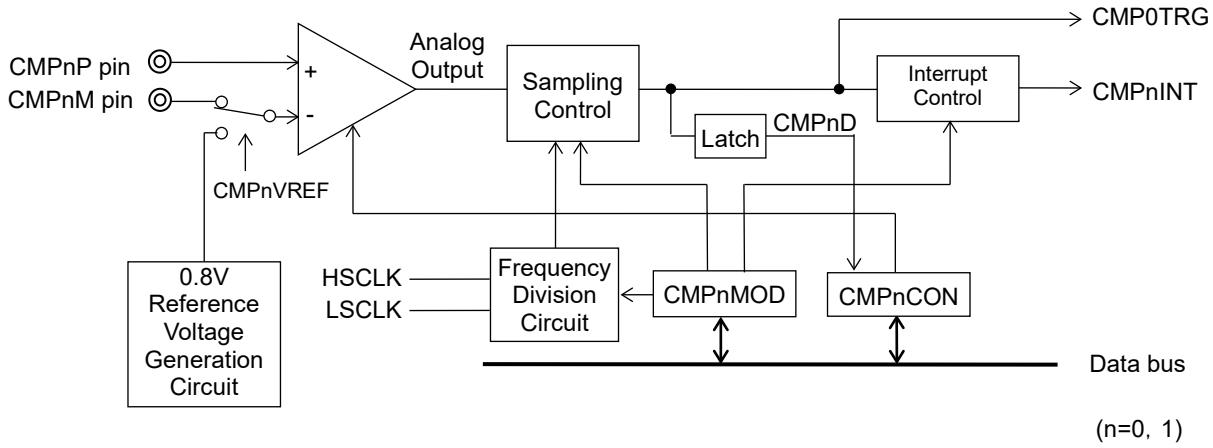
●: Available - : Unavailable

20.1.1 Features

- Comparable with external 2 voltage inputs.
- Comparable with external voltage input and internal reference voltage (approx. 0.8V).
- Three types of interrupt timing generated by the voltage comparison are available.
  - Rising edge of the comparison result
  - Falling edge of the comparison result
  - Rising edge and Falling edge of the comparison result
- The sampling with a clock is optional for the comparison result
  - HSCLK
  - LSCLK
  - 1/2 HSCLK to 1/64 HSCLK
  - 1/2 LSCLK to 1/64 HSCLK
- Last comparison result CMPnD(n=0,1) is retained when the analog comparator is stopped
- The analog comparator result output can be used as a trigger event source for the Functional Timer.

20.1.2 Configuration

Figure 20-1 shows the configuration of the analog comparator.



CMPnCON : Comparator n control register
CMPnMOD : Comparator n mode register
CMPnD : Analog Comparator n result
CMPnINT : Analog Comparator n Interrupt
CMPnVREF : Reference voltage select setting
CMP0TRG : Analog Comparator 0 output. Trigger event source for the Functional Timer.

Figure 20-1 Configuration of Analog Comparator

20.1.3 List of Pins

The I/O pins of the Analog Comparator are assigned to the shared function of the general ports.
For details of pin assignment, see Chapter 17 "GPIO".

Pin name	I/O	Function
CMPnP	I	Analog comparator n non-inverting input
CMPnM	I	Analog comparator n inverting input

(n=0 to 1)

Table 20-2 shows the list of the general ports used for the Analog Comparator and the register settings of the ports.

Table 20-2 Ports used in the Analog Comparator and the register settings

Pin name	Shared port		Setting Register	Setting value	ML62Q1300 group				ML62Q1500/ML62Q1800 ML62Q1700 group				
					16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
CMP0P	P03	7th Func.	P0MOD3	0110_0000	●	●	●	●	●	●	●	●	●
CMP0M	P02	7th Func.	P0MOD2	0110_0000	●	●	●	●	●	●	●	●	●
CMP1P	P62	7th Func.	P6MOD2	0110_0000	–	–	–	–	●	●	●	●	●
CMP1M	P63	7th Func.	P6MOD3	0110_0000	–	–	–	–	●	●	●	●	●

●: Available -: Unavailable

[Note]

- **When using the analog comparator, write "0" to the target PnmIE bit and PnmOE bit of port n mode registers (n=0 to 9, A, B, m=0 to 7) to set the general port to Hi-impedance, otherwise a shoot-through current may flow.**
- **An influence of the noise is reducible by preventing the switching of neighboring pins when the comparator enables.**

20.2 Description of Registers

20.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
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
0xF844 to 0xF847	Reserved	-	-	-	-	-
0xF848	Comparator 1 control register	CMP1CON	-	R/W	8	0x00
0xF849	Reserved	-	-	-	-	-
0xF84A	Comparator 1 mode register	CMP1MODL	CMP1MOD	R/W	8/16	0x00
0xF84B		CMP1MODH		R/W	8	0x00

20.2.2 Comparator n Control Register (CMPnCON: n=0,1)

CMPnCON is a special function register (SFR) used to control the analog comparator.

Address: 0xF840(CMP0CON), 0xF848(CMP1CON)
Access: R/W
Access size: 8bit
Initial value: 0x00

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

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1	CMPnD	This bit is used to indicate the comparison result of the analog comparator. The last comparison result is retained when the comparator is stopped. 0: CMPnP < CMPnM or 0.8V internal reference (Initial) 1: CMPnP > CMPnM or 0.8V internal reference
0	CMPnEN	This bit is used to control enable or disable the operation of the comparator. 0: Disables operating the comparator (initial value) 1: Enables operating the comparator

[Note]

- **When using the analog comparator, write "0" to the target PnmIE bit and PnmOE bit of port n mode registers (n=0 to 9, A, B, m=0 to 7) to set the general port to Hi-impedance, otherwise a shoot-through current may flow.**
- **An influence of the noise is reducible by preventing the switching of neighboring pins when the comparator enables.**

20.2.3 Comparator n Mode Register (CMPnMOD: n=0,1)

CMPnMOD is a special function register (SFR) used to set the operation mode of the analog comparator.

Address: 0xF842(CMP0MODL/CMP0MOD), 0xF843(CMP0MODH),
0xF84A(CMP1MODL/CMP1MOD), 0xF84B(CMP1MODH),
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CMPnMOD															
Byte	CMPnMODH								CMPnMODL							
Bit	-	-	-	-	-	-	-	CMPn VREF	-	CMPn DIV2	CMPn DIV1	CMPn DIV0	CMPn CS1	CMPn CS0	CMPn E1	CMPn E0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 9	-	Reserved bits
8	CMPnVREF	This bit is used to choose the reference voltage of the analog comparator. 0: Use a reference voltage input from the CMPnM pin (initial value) 1: Use the internal 0.8V reference voltage
7	-	Reserved bit
6 to 4	CMPnDIV2 to CMPnDIV0	These bits are used to choose frequency dividing ratio for the sampling clock in the analog comparator. 000: No dividing (Initial value) 001: 1/2 of the sampling clock source 010: 1/4 of the sampling clock source 011: 1/8 of the sampling clock source 100: 1/16 of the sampling clock source 101: 1/32 of the sampling clock source 110: 1/64 of the sampling clock source 111: Do not use (No dividing)
3, 2	CMPnCS1, CMPnCS0	These bits are used to choose the sampling clock source in the analog comparator. 00: No sampling (Initial value) 01: Sampling with HSCLK 10: Sampling with LSCLK 11: Do not use (No sampling)
1, 0	CMPnE1, CMPnE0	These bits are used to choose the timing of interrupt request generation. 00: Disable the interrupt request generation (Initial value) 01: Falling-edge 10: Rising-edge 11: Both-edge

[Note]

- **In the STOP/STOP-D mode, the sampling clock stops and the VLS works without sampling regardless the setting in CMPnCS1 bit and CMPnCS0 bit. When choosing "with sampling" and entering those mode, there is a time period(*1) in which interrupts gets disabled.**

  ***1 Time period to entering the STOP/STOP-D mode: Max.30µs. When returning from those modes, the interrupts are disabled until the sampling clock starts being supplied. The delay time depends on the configuration of clock and registers. See Table 4-5 "Wake-up Time from Standby Mode" in the Chapter 4 "Power Management".**

- **When the HSCLK is chosen for the sampling clock and the high-speed clock is not oscillating, the sampling circuit does not work. When using analog comparator in this case, choose "No sampling" or "Sampling with LSCLK" for sampling condition. For how to enable the high-speed clock oscillation, see Chapter 6 "Clock Generation Circuit".**

- **Write CMPnMOD register when the comparator stops (CMPnEN bit of CMPnCON register is "0"), otherwise the comparison result is unguaranteed.**

- **The internal reference voltage controlled by CMPnVREF bit is for the comparator. See the chapter of "Successive Approximation type A/D Converter" for the reference voltage used in the A/D converter.**

20.3 Description of Operation

20.3.1 Analog Comparator Operation

Figure 20-2 shows an analog comparator operation overview.

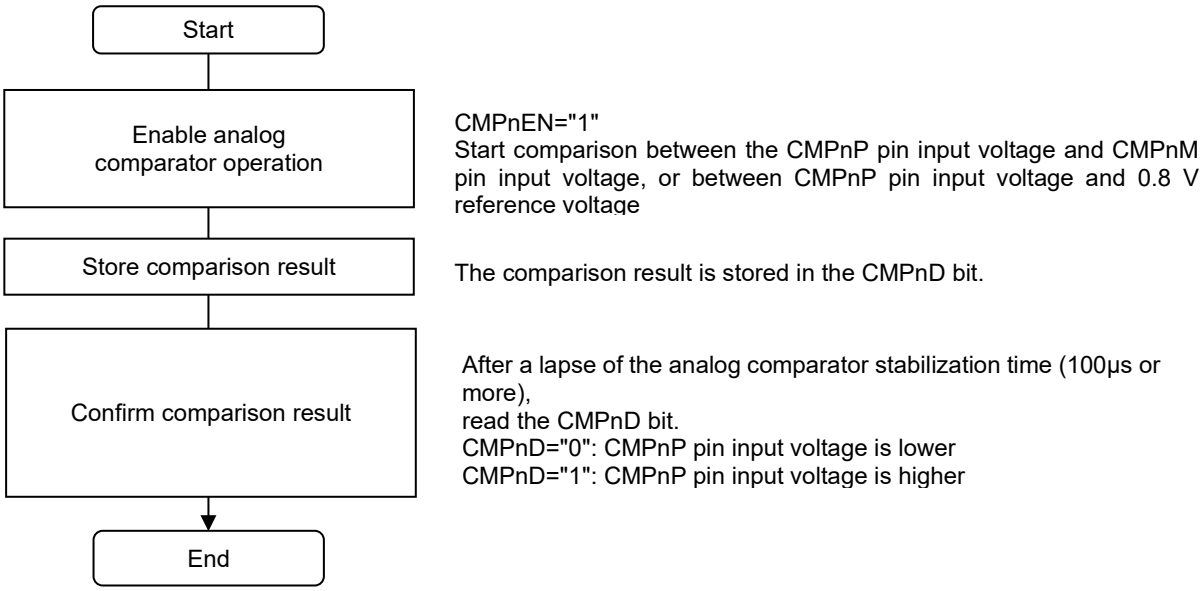


Figure 20-2 Analog Comparator Operation Overview

Figure 20-3 shows an example of the analog comparator operation timing.

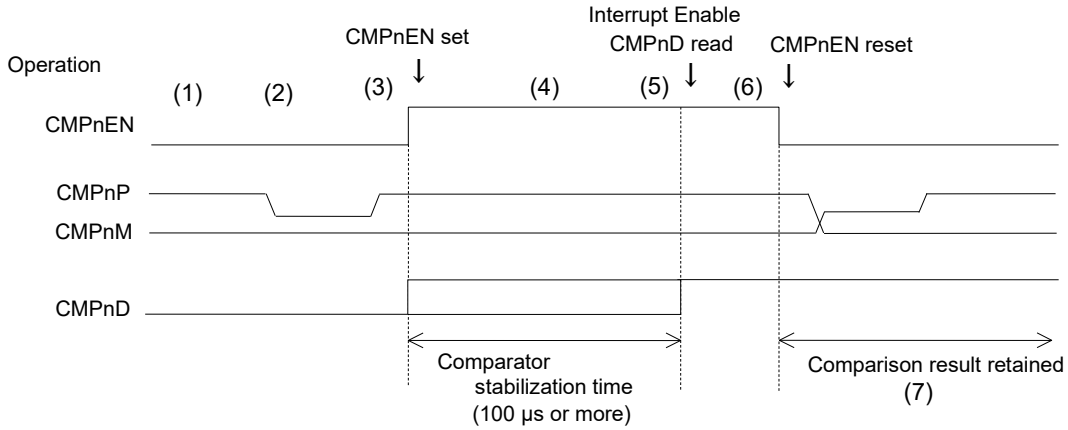


Figure 20-3 Example of Analog Comparator Operation Timing

Operation shown in Figure 20-3 above is described below.

(1) To operate the analog comparator, first set the following configuration:
   - Set the general-purpose port used for the analog comparator to high-impedance that both input and output is disabled, by writing "0" to PnmIE bit and PnmOE bit (m: bit number 0 to 7).
   - If using the high-speed clock for the sampling clock, write "1" to the ENOSC bit of the frequency control register (FCON).
   - Enable clock supply using the block clock control register 3 (BCKCON3).
   - Release the analog comparator reset using the block reset control register 3 (BRECON3).

(2) Choose the interrupt mode and sampling conditions using the CMPnMOD register.

(3) Write "1" to the CMPnEN bit to enable the analog comparator operation.

(4) Wait for the stabilization time (100 µs or more) of the analog comparator.

(5) Read the comparison result from the CMPnD bit.

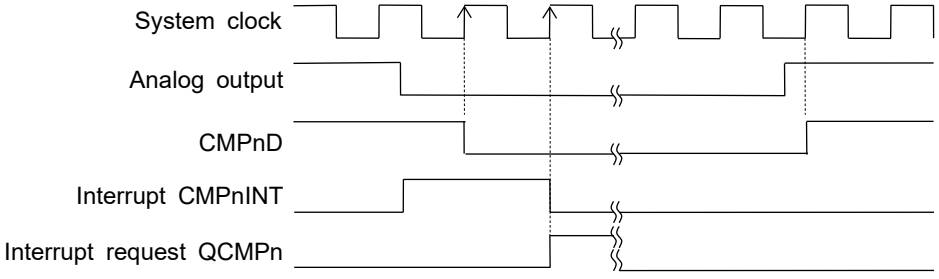
If using interrupt, write "1" to the ECMPn bit of the interrupt enable register 45 (IE45), after clearing QCMPn bit of the interrupt request register 45 (IRQ45).

(6) Write "0" to the CMPnEN bit to disable the analog comparator operation.

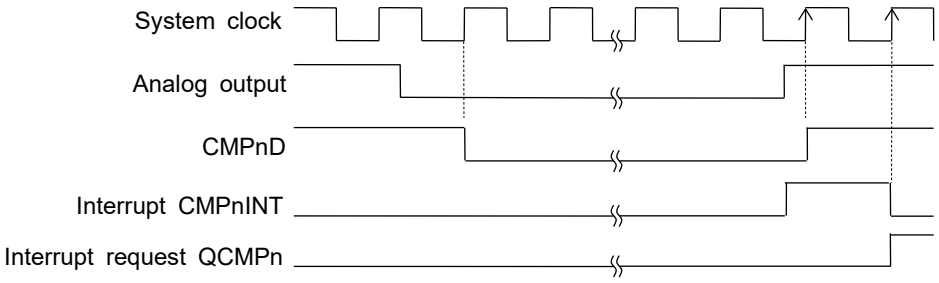
(7) The CMPnD bit may be read after "0" is written to the CMPnEN bit because the CMPnD bit holds the comparison result at the time when "0" is written to the CMPnEN bit.

20.3.2 Interrupt Request

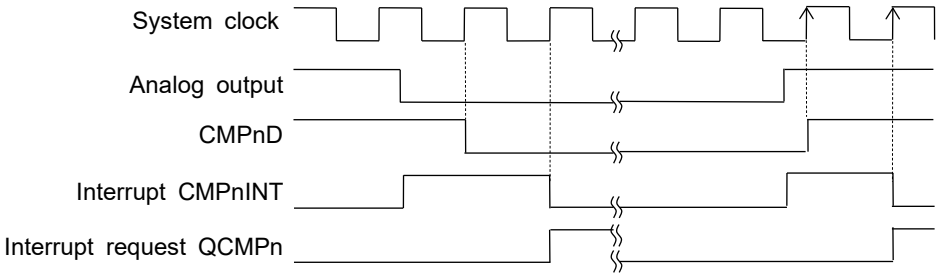
If the interrupt edge chosen in the CMPnE0 and CMPnE1 bits of the CMPnMOD register is detected, the analog comparator n interrupt (CMPnINT) is generated.
Figure 20-4 shows the interrupt generation timing without sampling (when the falling-edge/rising-edge/both-edge interrupt mode is chosen).
Figure 20-5 shows the interrupt generation timing with sampling (when the rising-edge interrupt mode is chosen).
Figure 20-6 shows the interrupt generation timing in the STOP/STOP-D mode.



(a) When falling-edge interrupt mode is chosen



(b) When rising-edge interrupt mode is chosen



(c) When both-edge interrupt mode is chosen

Figure 20-4 Analog Comparator Interrupt Generation Timing (without Sampling)

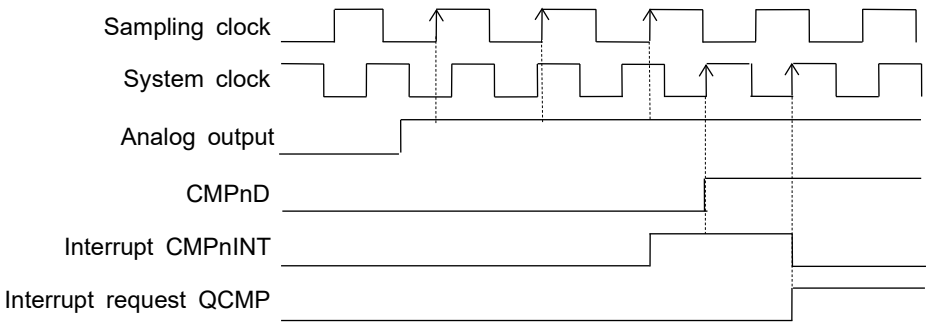


Figure 20-5 Analog Comparator Interrupt Generation Timing
(with Sampling, When Rising-edge Interrupt Mode is chosen)

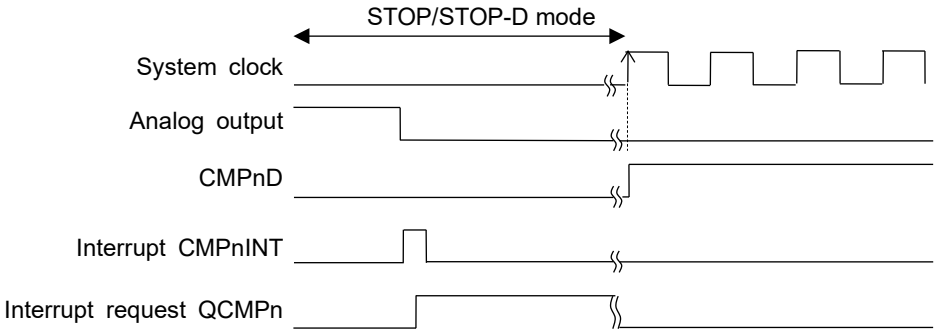


Figure 20-6 Analog Comparator Interrupt Generation Timing in STOP/STOP-D Mode
(When Falling-edge Interrupt Mode is chosen)

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.
The number of D/A converter channels is dependent of the product specification.
Table 21-1 shows the number of channels.

Table 21-1 Number of D/A converter channels

Channel no.	ML62Q1300 group				ML62Q1500/ML62Q1800/ML62Q1700 group				
	16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
0	−	−	●	●	●	●	●	●	●
1	−	−	−	−	−	−	−	●	●

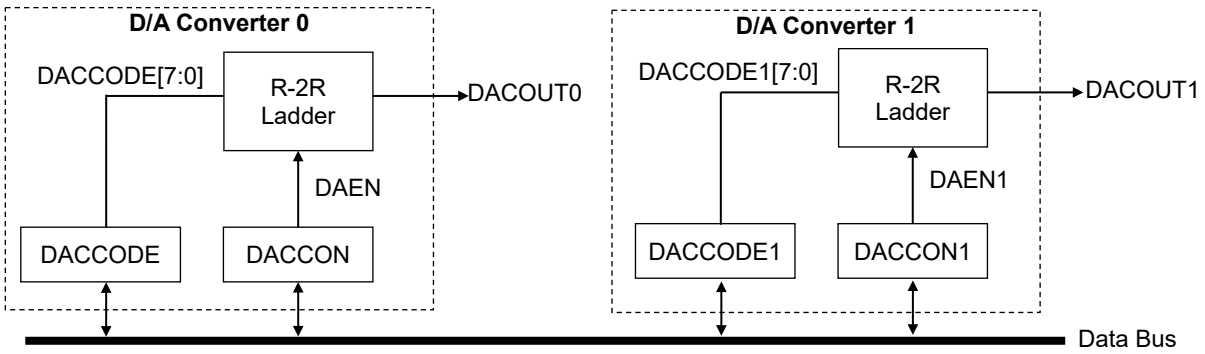
●: Available - : Unavailable

21.1.1 Features

- 8-bit resolution
- R-2R ladder method
- Analog output voltage (DACOUT0/DACOUT1)
  - Output voltage : V_{DD} x (Setting value in the SFR) / 256
  - Output impedance : 6kΩ (Typ.)

21.1.2 Configuration

Figure 21-1 shows the configuration of the D/A converter.



DACCON : D/A converter 0 control register
DACCODE : D/A converter 0 code register
DACCON1 : D/A converter 1 control register
DACCODE1 : D/A converter 1 code register

Figure 21-1 Configuration of D/A Converter Circuit

21.1.3 List of Pins

The I/O pins of the D/A converter are assigned to the shared function of the general ports.

Pin name	I/O	Function
DACOUT0	O	D/A converter 0 output
DACOUT1	O	D/A converter 1 output

Table 21-2 shows the list of the general ports used for the D/A converter and the register settings of the ports.

Table 21-2 Ports used in the D/A converter and the register settings

Channel no.	Pin name	Shared port		Setting Register	Setting value	ML62Q1300 group				ML62Q1500/ML62Q1800 ML62Q1700 group				
						16pin product	20 pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
0	DACOUT0	P01	Primary Func.	P0MOD1	0000_0000	-	-	●	●	●	●	●	●	●
1	DACOUT1	P44	Primary Func.	P4MOD4	0000_0000	-	-	-	-	-	-	-	●	●

●: Available - : Unavailable

[Note]

- **When using the D/A converter, write "0" to the target PnmIE bit and PnmOE bit of port n mode registers (n=0 to 9, A, B, m=0 to 7) to set the general port to Hi-impedance (input and output disabled), 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 conveter is operating.**

21.2 Description of Registers

21.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF860	D/A converter 0 control register	DACCON	-	R/W	8	0x00
0xF861	Reserved	-	-	-	-	-
0xF862	D/A converter 0 code register	DACCODE	-	R/W	8	0x00
0xF863 to 0xF867	Reserved	-	-	-	-	-
0xF868	D/A converter 1 control register	DACCON1	-	R/W	8	0x00
0xF869	Reserved	-	-	-	-	-
0xF86A	D/A converter 1 code register	DACCODE1	-	R/W	8	0x00
0xF86B	Reserved	-	-	-	-	-

21.2.2 D/A Converter 0 Control Register (DACCON)

DACCON is a special function register (SFR) used to control the D/A converter 0.

Address: 0xF860 (DACCON)
Access: R/W
Access size: 8bit
Initial value: 0x00

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

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	DAEN	This bit is used to enable or disable the operation of the D/A converter 0. 0: Disable/Stop operating the D/A converter 0 (Initial value) 1: Enable operating the D/A converter 0

[Note]

- **When using the D/A converter, write "0" to the target PnmIE bit and PnmOE bit of port n mode registers (n=0 to 9, A, B, m=0 to 7) to set the general port to Hi-impedance (input and output disabled), othewise a shoot-through current may flow.**

21.2.3 D/A Converter 0 Code Register (DACCODE)

DACCODE is a special function register (SFR) used to set the conversion value of the D/A converter 0.

Address: 0xF862 (DACCODE)
Access: R/W
Access size: 8bit
Initial value: 0x00

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

Bit No.	Bit symbol name	Description
7 to 0	d7 to d0	These bits are used to set the output voltage of the D/A converter 0. A voltage of V_{DD} x DACCODE / 256 is output. 0x00: Output V_{SS} (Initial value) 0x01 to 0xFF: Output a voltage = V_{DD} x (decimal code of DACCODE / 256)

21.2.4 D/A Converter 1 Control Register (DACCON1)

DACCON1 is a special function register (SFR) used to control the D/A converter 1.

Address: 0xF868 (DACCON1)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								DACCON1							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DAEN1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bit
0	DAEN1	This bit is used to enable or disable the operation of the D/A converter 1. 0: Disable/Stop operating the D/A converter 1 (Initial value) 1: Enable operating the D/A converter 1

[Note]

- **When using the D/A converter, write "0" to the target PnmIE bit and PnmOE bit of port n mode registers (n=0 to 9, A, B, m=0 to 7) to set the general port to Hi-impedance (input and output disabled), othewise a shoot-through current may flow.**

21.2.5 D/A Converter 1 Code Register (DACCODE1)

DACCODE1 is a special function register (SFR) used to the conversion value of the D/A converter 1.

Address: 0xF86A (DACCODE1)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								DACCODE1							
Bit	-	-	-	-	-	-	-	-	d7	d6	d5	d4	d3	d2	d1	d0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	d7 to d0	These bits are used to set the conversion value of the D/A converter 1. A voltage of V_{DD} x DACCODE / 256 is output. 0x00: Output V_{SS} (Initial value) 0x01 to 0xFF: Output a voltage = V_{DD} x (decimal code of DACCODE1 / 256)

21.3 Description of Operation

21.3.1 D/A Converter Operation

Figure 21-2 shows a process flow chart to control the D/A converter n.
Figure 21-3 shows the operation timing chart.

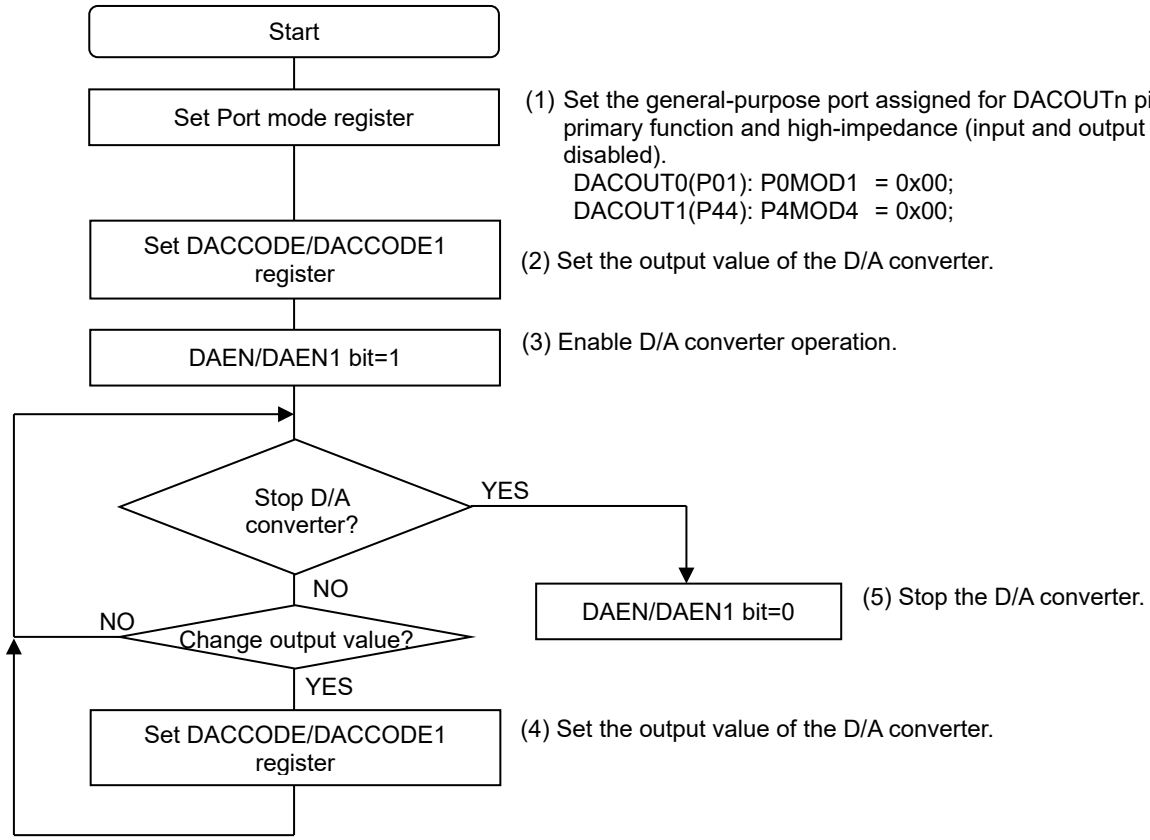
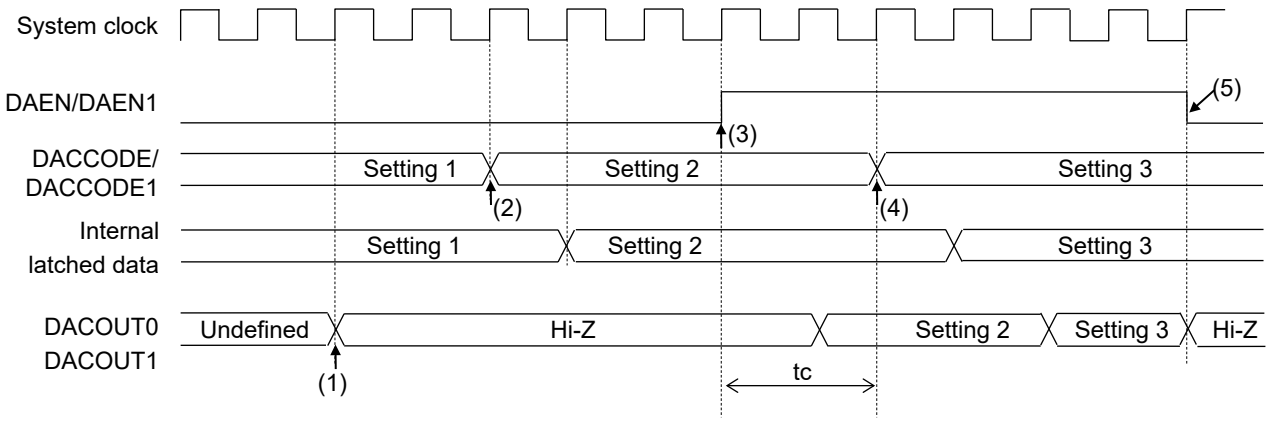


Figure 21-2 D/A Converter Process Flow Chart



tc : Conversion cycle of D/A converter

Figure 21-3 D/A Conversion Operation Timing Chart

Chapter 22 Voltage Level Supervisor

22. Voltage Level Supervisor

22.1 General Description

ML62Q1000 series has the Voltage Level Supervisor (VLS0) that detects whether the voltage level of V_{DD} is lower or higher than the specified threshold voltage.

22.1.1 Features

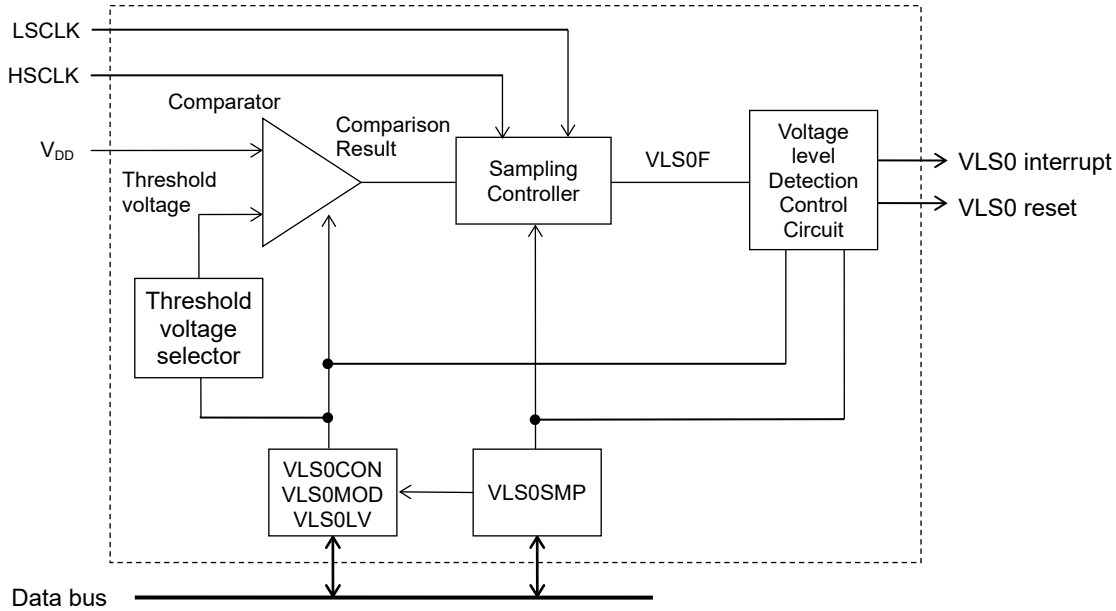
- Accuracy: ±4 %
- Threshold voltage: Selectable from 12 values (1.85 to 4.00 V)
- Operation mode: Supervisor mode (continuous detection) or single mode (one detection)

Mode	Description
Single mode 1	Detect the voltage level of V_{DD} only once. The interrupt occurs after detecting the voltage of V_{DD}, indicates the detection has been completed.
Single mode 2	Detect the voltage level of V_{DD} only once. The interrupt occurs after detecting the voltage of V_{DD} is lower than the threshold voltage, indicates the MCU is in the low voltage condition.
Supervisor mode	Detect continuously the voltage level of V_{DD}, suitable for always detecting the low voltage level of V_{DD} and generating the interrupt or reset. The interrupt or reset occurs according to the setting in the VLS0MOD register. The VLS0 reset function is available by choosing the supervisor mode.

- Voltage level supervisor reset (VLS0 reset)
- Voltage level supervisor interrupt (VLS0 interrupt)
- Initialized by the power-on reset (POR) or pin reset

22.1.2 Configuration

The voltage level supervisor (VLS0) consists of a comparator, a sampling control circuit, and a voltage level detection control circuit. Figure 22-1 shows the configuration of the VLS0.



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		R/W	Size	Initial Value
		Byte	Word			
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	-	-	-	-	-

22.2.2 Voltage Level Supervisor 0 Control Register (VLS0CON)

VLS0CON is a special function register (SFR) used to control the VLS0 (Voltage Level Supervisor).
This register is unresetable by anything other than the Power On Reset(POR) and RESETN pin reset.

Address: 0xF850 (VLS0CON)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								VLS0CON							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	VLS0RF	VLS0F	VLS0EN
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 3	-	Reserved bit
2	VLS0RF	This bit is used to indicate whether the voltage level detection result is valid or not. This bit is valid only in the supervisor mode and fixed to "0" in the single mode. 0: The VLS0 circuit is stopped or VLS0 is being stabilized (initial value) 1: The VLS0 detection result is valid (readable)
1	VLS0F	This bit for monitoring the voltage level retains the last detection result. This bit is cleared to "0" by writing "1" to this bit, but not cleared by writing "0". Also, this bit is cleared to "0" when the VL0 starts operating. 0: The power voltage(V_{DD}) is higher than the threshold voltage (initial value) 1: The power voltage(V_{DD}) is lower than the threshold voltage
0	VLS0EN	This bit is used to control the VLS0 operation. In the single mode, this bit is automatically reset to "0" after detecting the voltage level and the VLS0 stops operating. 0: Disable operating the VLS0 (Initial value) 1: Enable operating the VLS0

[Note]

- **There is a limitation in each mode for entering the STOP/STOP-D mode while the VLS0 is running.**

Mode	Description
Running in the supervisor mode	The MCU can enter the STOP/STOP-D mode only when the VLS0RF bit is "1".
Running in the single mode	The MCU is unable to enter the STOP/STOP-D mode. Enter the STOP/STOP-D mode when the VLS0 is not running (when the VLS0EN bit is "0").

- **Even if resets other than the POR and RESET_N pin reset occurred, the VLS0 remains running.**

22.2.3 Voltage Level Supervisor 0 Mode Register (VLS0MOD)

VLS0MOD is a special function register (SFR) used to control the operation mode of the VLS0 (Voltage Level Supervisor).
Set this register only when the VLS0 is stopped (VLS0EN bit of VLS0CON register is "0").
This register is unresetable by anything other than the Power On Reset (POR) and RESETN pin reset.

Address: 0xF852 (VLS0MOD)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								VLS0MOD							
Bit	-	-	-	-	-	-	-	-	-	-	VLS0AMD1	VLS0AMD0	-	-	VLS0SEL1	VLS0SEL0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7, 6	-	Reserved bits
5, 4	VLS0AMD1, VLS0AMD0	These bits are used to choose the VLS0 operating mode 00: Single mode 1 (Initial value) It detects the voltage level of V_{DD} only once. When VLS0SEL1 and VLS0SEL0 bits are "0x2", the interrupt occurs when detecting the voltage level of V_{DD}. The result can be checked by reading VLS0F bit of VLS0CON register. 01: Single mode 2 It detects the voltage level of V_{DD} only once. When VLS0SEL1 and VLS0SEL0 bits are "0x2", the interrupt occurs when detecting the voltage level of V_{DD} is lower than the threshold voltage (when the VSL0F of VLS0CON is "1"). 1X: Supervisor mode It always detects the voltage level of V_{DD}. The interrupt or reset occurs depending on the conditions of VLS0SEL1 and VLS0SEL0 bits.
3, 2	-	Reserved bits
1, 0	VLS0SEL1, VLS0SEL0	These bits are used to control enable/disable of the VLS0 reset / VLS0 interrupt request. See section 22.3 “Description of Operation” for details of the occurrence condition of VLS0 reset / VLS0 interrupt request. 00: Reset function is disable and Interrupt function is disable (Initial value) 01: Reset function is enable and Interrupt function is disable 10: Reset function is disable and Interrupt function is enable 11: Reset function is enable and Interrupt function is disable

[Note]

- **There is a limitation in each mode for entering the STOP/STOP-D mode while the VLS0 is running.**

Mode	Description
Running in the supervisor mode	The MCU can enter the STOP/STOP-D mode only when the VLS0RF bit is "1".
Running in the single mode	The MCU is unable to enter the STOP/STOP-D mode. Enter the STOP/STOP-D mode when the VLS0 is not running (when the VLS0EN bit is "0").

22.2.4 Voltage Level Supervisor 0 Level Register (VLS0LV)

VLS0LV is a special function register (SFR) used to set the detection voltage.
Set this register only when the VLS0 is stopped (VLS0EN bit of VLS0CON register is "0").
This register is unresetable by anything other than the Power On Reset (POR) and RESETN pin reset.

Address: 0xF854 (VLS0LV)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								VLS0LV							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	VLS0LV3	VLS0LV2	VLS0LV1	VLS0LV0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 4	-	Reserved bit
3 to 0	VLS0LV3 to VLS0LV0	These bits are used to choose the threshold voltage (V_{VLSF} / V_{VLSR}) of VLS0 detected while the V_{DD} is falling or rising. The VLS0 has hysteresis characteristics. For the characteristics of threshold voltage detected while the power is falling or rising, see the data sheet of each product. 0000: 4.00V±4% (Initial value) 0001: 3.70V±4% 0010: 3.05V±4% 0011: 2.95V±4% 0100: 2.85V±4% 0101: 2.75V±4% 0110: 2.65V±4% 0111: 2.55V±4% 1000: 2.45V±4% 1001: 2.05V±4% 1010: 1.95V±4% 1011: 1.85V±4% 1100: Do not use (1.85V±4%) 1101: Do not use (1.85V±4%) 1110: Do not use (1.85V±4%) 1111: Do not use (1.85V±4%)

22.2.5 Voltage Level Supervisor 0 Sampling Register (VLS0SMP)

VLS0SMP is a special function register (SFR) used to control sampling the voltage level detection.
Set this register only when the VLS0 is stopped (VLS0EN bit of VLS0CON register is "0").
This register is unresetable by anything other than the Power On Reset (POR) and RESETN pin reset.

Address: 0xF856 (VLS0SMP)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								VLS0SMP							
Bit	-	-	-	-	-	-	-	-	-	VLS0DIV2	VLS0DIV1	VLS0DIV0	VLS0SM1	VLS0SM0	-	-
R/W	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

Bit No	Bit symbol name	Description
7	-	Reserved bit
6 to 4	VLS0DIV2 to VLS0DIV0	These bits are used to choose frequency dividing ratio for the sampling clock. 000: No dividing (Initial value) 001: 1/2 of the sampling clock source 010: 1/4 of the sampling clock source 011: 1/8 of the sampling clock source 100: 1/16 of the sampling clock source 101: 1/32 of the sampling clock source 110: 1/64 of the sampling clock source 111: No dividing
3, 2	VLS0SM1, VLS0SM0	These bits are used to choose the sampling clock source for detecting the voltage level. 00: No sampling (Initial value) 01: Sampling with HSCLK 10: Sampling with LSCLK 11: No sampling
1, 0	-	Reserved bit

[Note]

- **In the STOP/STOP-D mode, the sampling clock stops and the VLS works without sampling regardless the setting in VLS0SM1 and VLS0SM0 bit. When choosing "with sampling" and entering those mode, there is a time period(*1) in which interrupts gets disabled.**

  ***1 Time period to entering the STOP/STOP-D mode: Max.30µs. When returning from those modes, the interrupts are disabled until the sampling clock starts being supplied. The delay time depends on the configuration of clock and registers. See Table 4-5 "Wake-up Time from Standby Mode" in the Chapter 4 "Power Management".**

- **When the HSCLK is chosen for the sampling clock and the high-speed clock is not oscillating, the sampling circuit does not work and it does not monitor voltage. When using VLS in this case, choose "No sampling" or "Sampling with LSCLK" for sampling condition. For how to enable the high-speed clock oscillation, see Chapter 6 "Clock Generation Circuit".**

22.3 Description of Operation

VLS0 can be used to verify if V_{DD} is lower or higher than the specified threshold voltage. In addition, it generates VLS0 interrupt or VLS0 reset.VLS0 has hysteresis characteristics. See the data sheet of each product for characteristics of the threshold voltage at power voltage fall / rise.

The following two operation modes are available for VLS0:

Supervisor mode:

Operation	“1" is written to VLS0EN to enable operation of VLS0, and then detecting the voltage is executed. The result is notified of through the VLS0RF flag as at the time the detection result becomes valid. The detection still continues.	
Function	Interrupt of detecting voltage variations	The interrupt is generated when the power voltage becomes lower or higher than the threshold voltage.
	Reset of detecting low voltage	The reset can be generated when the power voltage becomes lower than the threshold voltage.

Single mode:

Operation	"1" is written to VLS0EN to enable operation of VLS0, and then detecting the voltage is executed. "0" is automatically written to VLS0EN to end the detection when the detection result becomes valid.	
Function	Single mode 1: Interrupt that indicates the detecting voltage has been completed	The interrupt is generated at the time of completion of the voltage detection.
	Single mode 2: Interrupt of detecting low voltage	The interrupt is generated when the power voltage becomes lower than the threshold voltage.

22.3.1 Supervisor Mode

In the supervisor mode, the voltage level of V_{DD} can be constantly detected. This mode is suitable for using the reset when the low voltage is detected, or the interrupt when the voltage variations is detected.

Figure 22-2 shows the flow chart for starting the VLS in the supervisor mode.

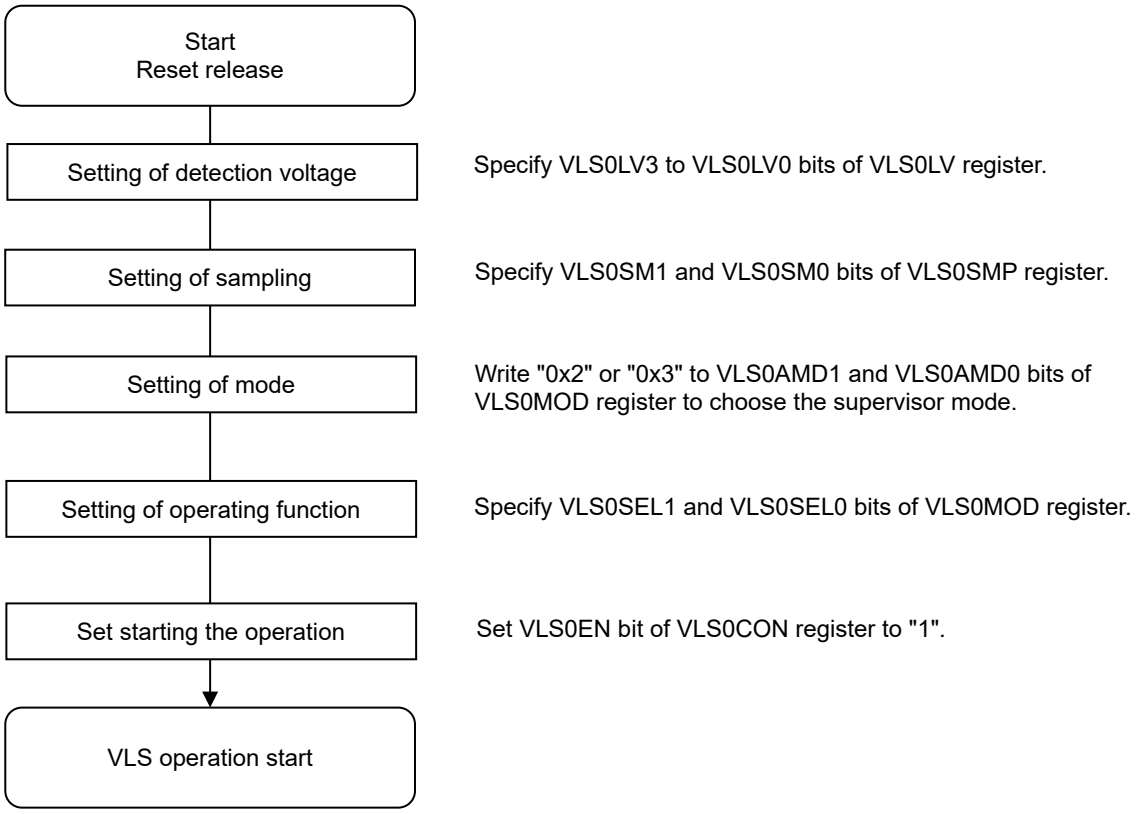


Figure 22-2 Flow chart for starting the VLS in the supervisor mode

22.3.1.1 Reset Output

Figure 22-3 shows the operation timing chart when the VLS0 reset output without sampling is specified.

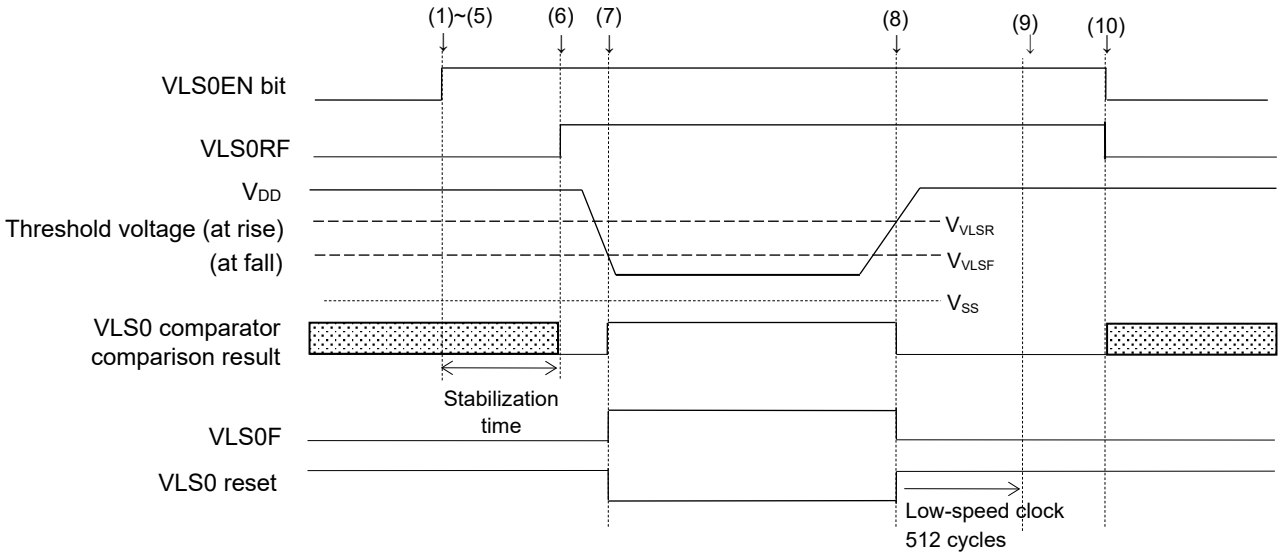


Figure 22-3 Operation Timing Chart When the VLS0 Reset Output without Sampling is specified

The operation shown in Figure 22-3 is described below:

(1) Choose a detection voltage by the VLS0LV3 to VLS0LV0 bits of the VLS0LV register.
(2) Choose "No sampling" by the VLS0SM1 and VLS0SM0 bits of the VLS0SMP register.
(3) Write "0x2" or "0x3" to VLS0AMD1 and VLS0AMD0 bits of VLS0MOD register in order to choose the supervisor mode.
(4) Write “0x1” to VLS0SEL1 and VLS0SEL0 bits of the VLS0MOD register in order to enable the VLS0 reset.
(5) Set the VLS0EN bit of the VLS0CON register to "1" (VLS0 starts operation in the supervisor mode).
(6) After approximately 300 µs passed, the detection result of VLS0 becomes stabilized and the VLS0RF bit of the VLSCON register is set to "1" (value of the voltage level supervisor bit (VLS0F) is read in software) (*[1]).
(7) When the power voltage (V_{DD}) becomes below the threshold voltage V_{VLSF}, the VLS0F bit is set to "1" to generate the VLS0 reset.
(8) If V_{DD} becomes equal to or above the threshold voltage (V_{VLSR}), the VLS0F bit is cleared to "0" to release the VLS0 reset.
(9) The CPU starts after 512 cycles of low-speed clock.
(10) Write "0" to the VLS0EN bit to disable VLS0 operation.

*[1]: VLS0F bit/interrupt/reset is masked until the VLS0RF bit becomes "1".

Figure 22-4 shows the operation timing chart when the VLS0 reset output with sampling is specified.

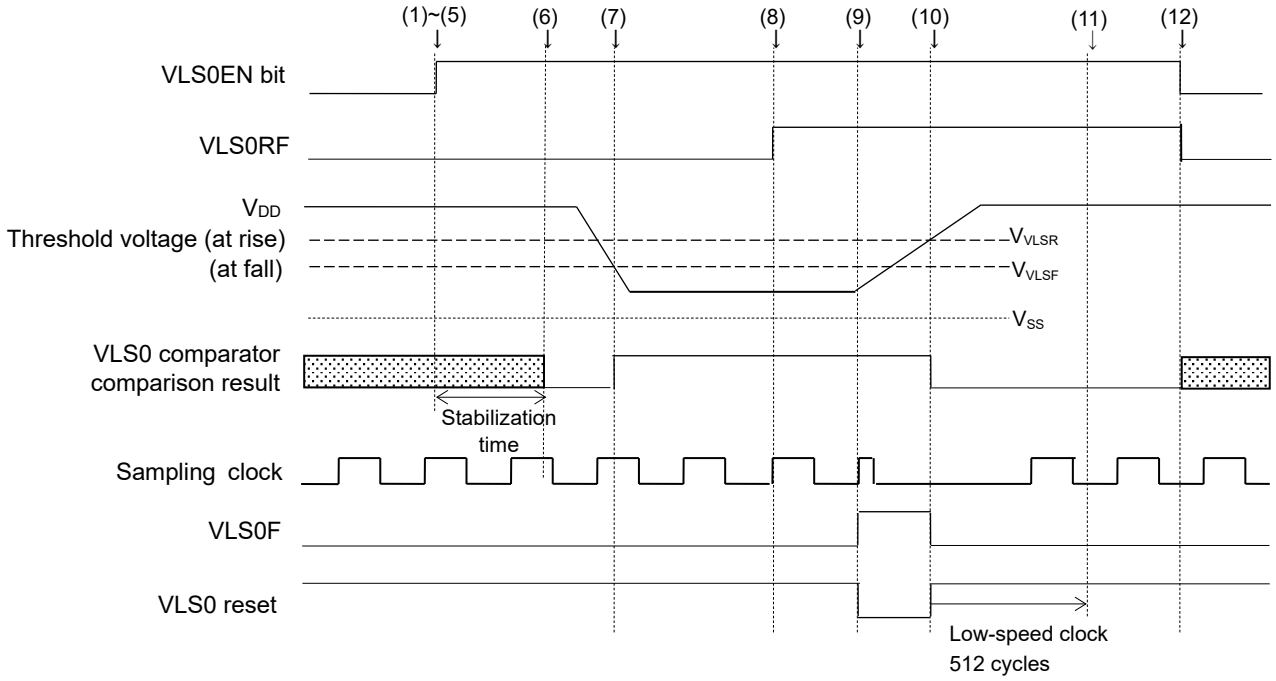


Figure 22-4 Operation Timing Chart When the VLS0 Reset Output with Sampling is specified

The operation shown in Figure 22-4 is described below:

(1) Choose a detection voltage by the VLS0LV3 to VLS0LV0 bits of the VLS0LV register.
(2) Choose "Sampling with HSCLK" or "Sampling with LSCLK" by the VLS0SM1 and VLS0SM0 bits of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV2 to VLS0DIV0 bits of the VLS0SMP register.
(3) Write "0x2" or "0x3" to VLS0AMD1 and VLS0AMD0 bits of the VLS0MOD register in order to choose the supervisor mode.
(4) Write “0x1” to VLS0SEL1 and VLS0SEL0 bits of the VLS0MOD register in order to enable the VLS0 reset.
(5) Write "1" to the VLS0EN bit to enable VLS0 operation.
(6) Wait until the comparison result of the VLS0 comparator is stabilized (approx. 300 μs).
(7) V_{DD} becomes below the threshold voltage (V_{VLSF}).
(8) Once the comparison result of the VLS0 comparator is stabilized, the VLS0RF bit is set to "1" after three cycles of the sampling clock.
(9) If the comparison result of the VLS0 comparator is below the threshold voltage (V_{VLSF}) and this condition continues for the duration of three cycles or more of the sampling clock, the VLS0F bit is set to "1" and the VLS0 reset is generated.
(10) If the comparison result of the VLS0 comparator becomes equal to or above the threshold voltage (V_{VLSR}), the VLS0F bit is cleared to "0" to release the VLS0 reset.
(11) The CPU starts after 512 cycles of low-speed clock. The VLS does not operate while the sampling clock is stops.
(12) Write "0" to the VLS0EN bit to disable VLS0 operation.

[Note]

- **Entering the STOP/STOP-D mode is not allowed during the VLS0 stabilization time. If entering the STOP/STOP-D mode after the supervisor mode is enabled, make sure that the VLS0RF bit is set to "1", and then enter the STOP/STOP-D mode.**
- **The initial value of the VLS0 detection voltage is 4V, so the MCU becomes in reset mode when the V_{DD} is 4V or lower and VLS0 is specified as supervisor mode with the reset function. Therefore, set the detection voltage before enabling the VLS0 operation.**
- **If you want to use the VLS0 reset function like a reset IC, start the VLS0 when the CPU initially runs at the low-speed clock after the power up.**

22.3.1.2 Interrupt Output

Figure 22-5 shows an example of the operation timing chart when the VLS0 interrupt output without sampling is specified.

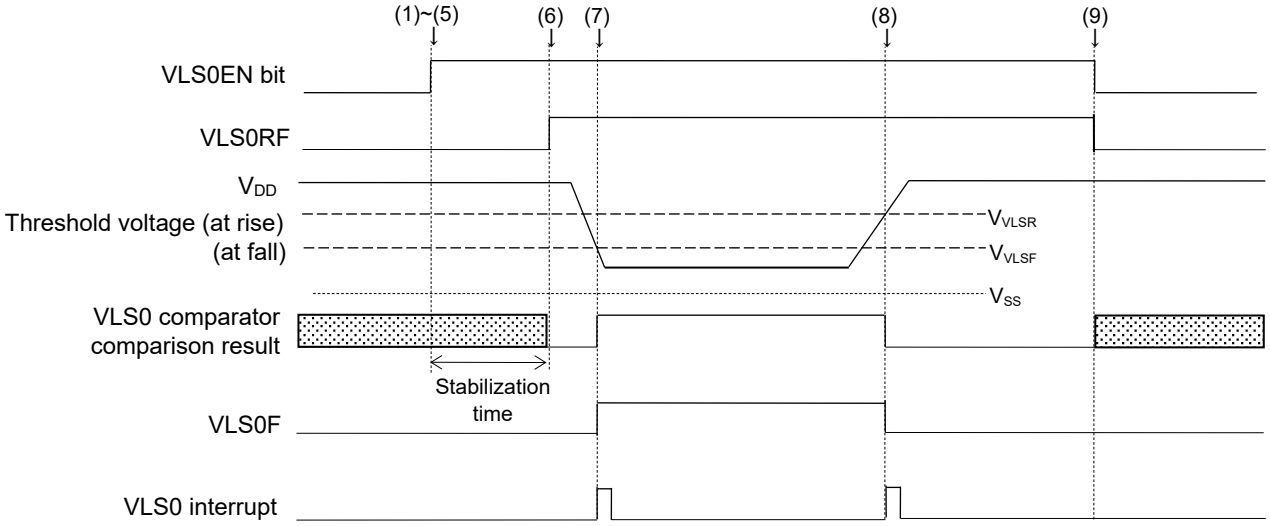


Figure 22-5 Operation Timing Chart When the VLS0 Interrupt Output without Sampling is specified

The operation shown in Figure 22-5 is described below:

(1) Choose a detection voltage by the VLS0LV3 to VLS0LV0 bits of the VLS0LV register.
(2) Choose "No sampling" by the VLS0SM1 and VLS0SM0 bits of the VLS0SMP register.
(3) Write "0x2" or "0x3" to VLS0AMD1 and VLS0AMD0 bits of VLS0MOD register in order to choose the supervisor mode.
(4) Write “0x2” to VLS0SEL1 and VLS0SEL0 bits of the VLS0MOD register in order to enable the VLS0 interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS0 operation.
(6) When the comparison result of the VLS0 comparator is stabilized, the VLS0RF bit is set to "1".
(7) When V_{DD} becomes below the threshold voltage (V_{VLSF}), the VLS0F bit is set to "1" to generate the VLS0 interrupt.
(8) If V_{DD} becomes equal to or above the threshold voltage (V_{VLSR}), the VLS0F bit is cleared to "0" to generate the VLS0 interrupt.
(9) Write "0" to the VLS0EN bit to disable VLS0 operation.

Figure 22-6 shows an example of the operation timing chart when the VLS0 interrupt output with sampling is specified.

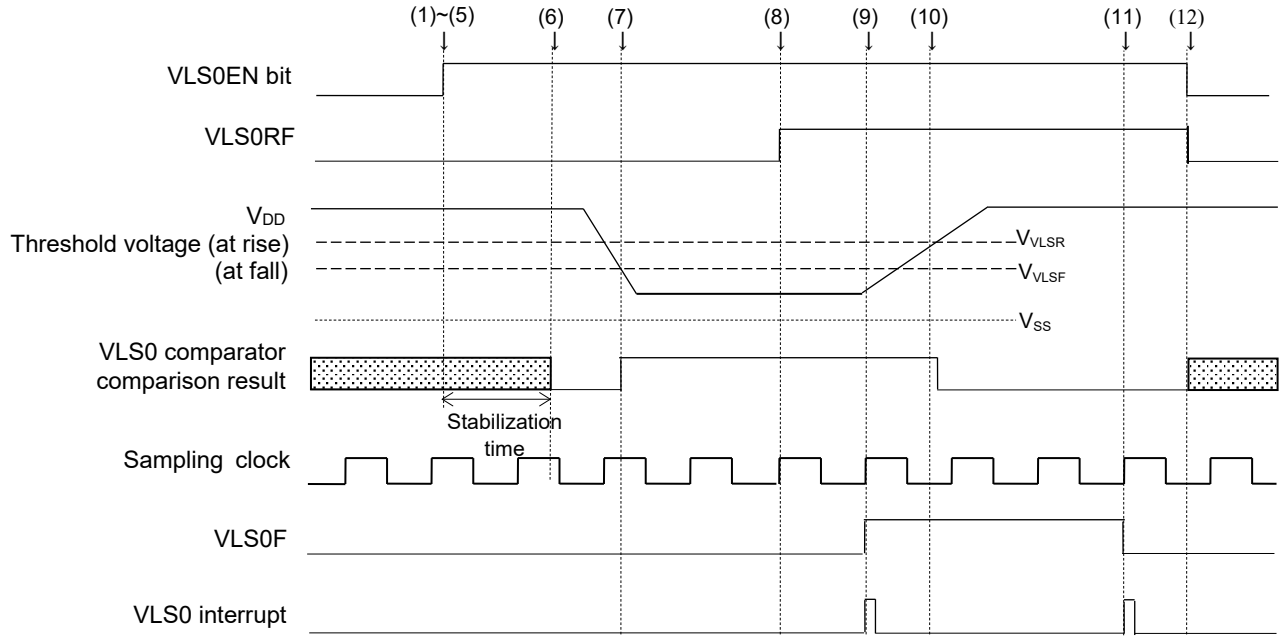


Figure 22-6 Operation Timing Chart When the VLS0 Interrupt Output with Sampling is specified

The operation shown in Figure 22-6 is described below:

(1) Choose a detection voltage by the VLS0LV3 to VLS0LV0 bits of the VLS0LV register.
(2) Choose "Sampling with HSCLK" or "Sampling with LSCLK" by the VLS0SM1 and VLS0SM0 bits of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV2 to VLS0DIV0 bits of the VLS0SMP register.
(3) Write "0x2" or "0x3" to VLS0AMD1 and VLS0AMD0 bits of VLS0MOD register in order to choose the supervisor mode.
(4) Write “0x2” to VLS0SEL1 and VLS0SEL0 bits of the VLS0MOD register in order to enable the VLS0 interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS0 operation.
(6) Wait until the comparison result of the VLS0 comparator is stabilized (approx. 300 μs).
(7) V_{DD} becomes below the threshold voltage (V_{VLSF}).
(8) Once the comparison result of the VLS0 comparator is stabilized, the VLS0RF bit is set to "1" after three cycles of the sampling clock.
(9) If the comparison result of the VLS0 comparator is below the threshold voltage (V_{VLSF}) and this condition continues for the duration of three cycles or more of the sampling clock, the VLS0F bit is set to "1" and the VLS0 interrupt is generated.
(10) The comparison result of the VLS0 comparator becomes equal to or above the threshold voltage (V_{VLSR}).
(11) If the comparison result of the VLS0 comparator is equal to or above the threshold voltage (V_{VLSR}) and this condition continues for the duration of three cycles or more of the sampling clock, the VLS0F bit is cleared to "0" and the VLS0 interrupt is generated.
(12) Write "0" to the VLS0EN bit to disable VLS0 operation.

[Note]

- **Entering the STOP/STOP-D mode is not allowed during the VLS0 stabilization time. If entering the STOP/STOP-D mode after the supervisor mode is enabled, make sure that the VLS0RF bit is set to "1", and then enter the STOP/STOP-D mode.**
- **When VLS0 is stopped (VLS0EN bit="0") while the V_{DD} is lower than the specified threshold voltage (VLS0F bit="1"), the VLS0 interrupt is generated.**

22.3.2 Single Mode

In the single mode, the software waits for the VLS0 interrupt to detect the voltage. It is useful for intermittently checking V_{DD}.

Figure 22-7 shows the flow chart for starting the VLS in the single mode.

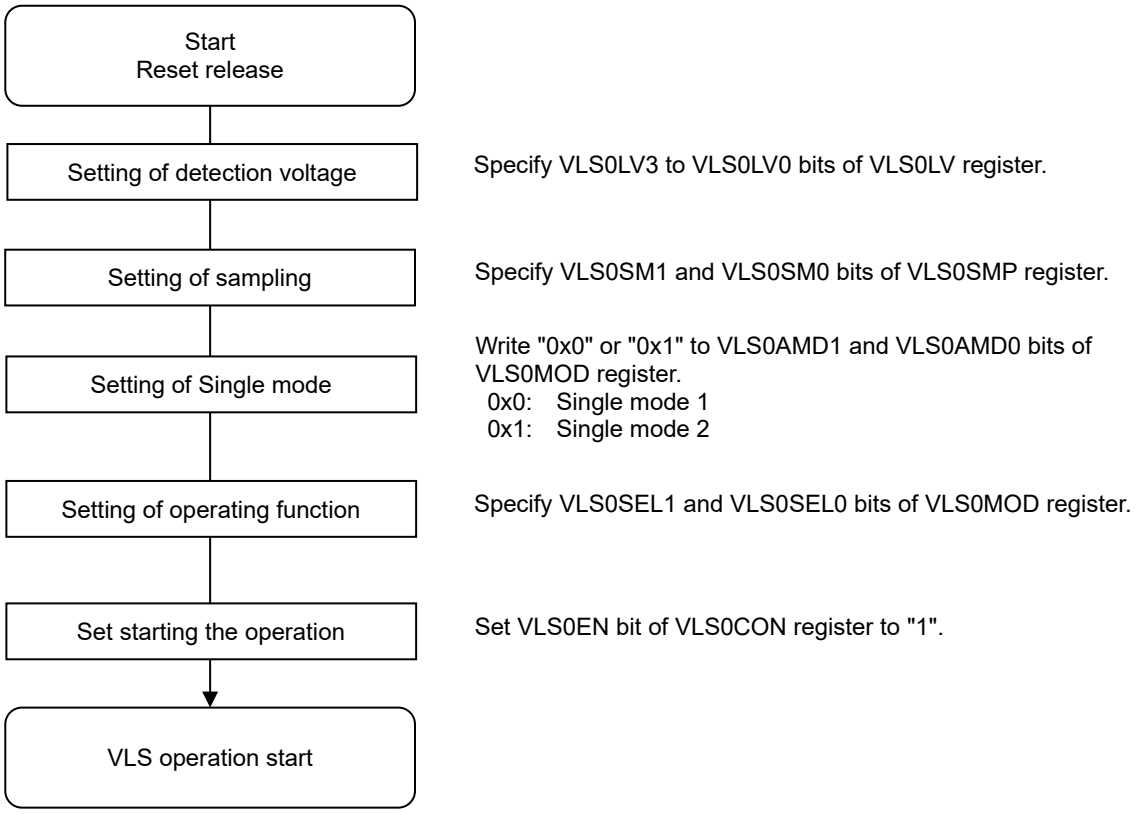


Figure 22-7 Flow chart for starting the VLS in the single mode

22.3.2.1 Single mode 1

The single mode 1 always generates the interrupt at completing the detection.
Figure 22-8 shows an example of the operation timing chart without sampling in single mode 1.

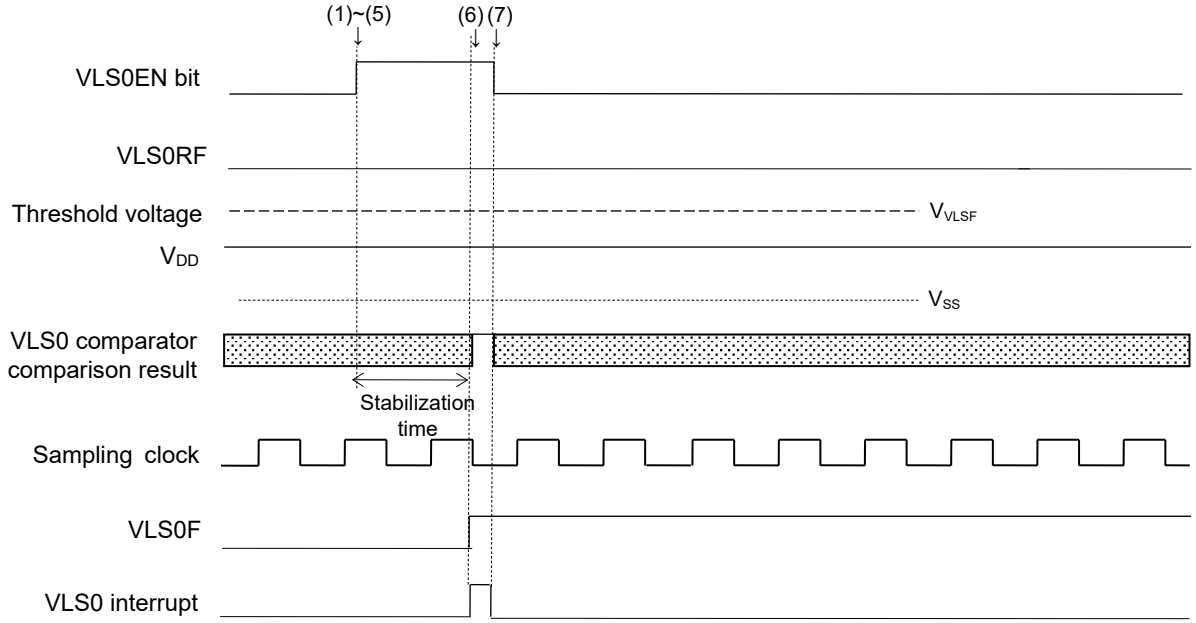


Figure 22-8 Operation Timing Chart without Sampling (Single Mode 1)

The operation shown in Figure 22-8 is described below:

(1) Choose a detection voltage by the VLS0LV3 to VLS0LV0 bits of the VLS0LV register.
(2) Choose "No sampling" by the VLS0SM1 and VLS0SM0 bits of the VLS0SMP register.
(3) Write "0x0" to VLS0AMD1 and VLS0AMD0 bits of VLS0MOD register in order to choose the single mode 1.
(4) Write “0x2” to VLS0SEL1 and VLS0SEL0 bits of the VLS0MOD register in order to enable the VLS0 interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS0 operation.
(6) If V_{DD} is below the threshold voltage (V_{VLSF}) when the comparison result of the VLS0 comparator is stabilized (*1), the VLS0F bit is set to "1" and the VLS0 interrupt (detection complete) is generated. The VLS0 interrupt (detection complete) is generated regardless of the detection result of V_{DD}.
(7) After the interrupt is generated, the VLS0EN bit is cleared to "0" and VLS0 operation is disabled.
(8) Read the VLS0F bit to confirm the detection result.

*1: Stabilization time: Approximately 300 μs (approx. 300 μs + sampling clock cycle x 3 when sampling is enabled)

Figure 22-9 shows an example of the operation timing chart with sampling in single mode 1.

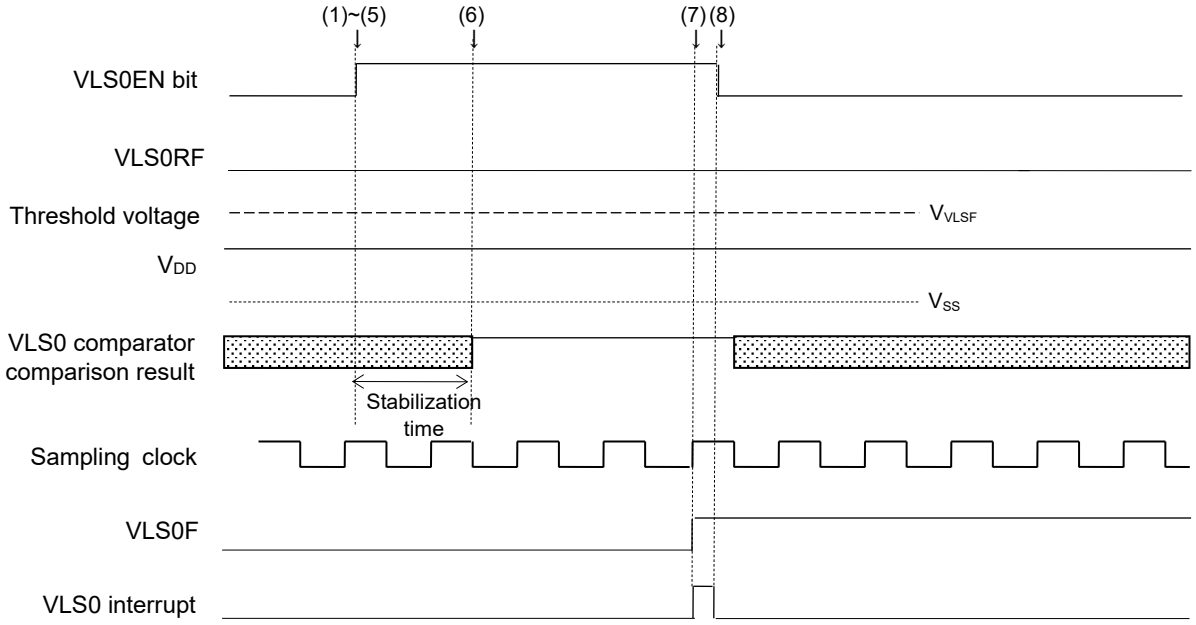


Figure 22-9 Operation Timing Chart with Sampling (Single Mode 1)

The operation shown in Figure 22-9 is described below:

(1) Choose a detection voltage by the VLS0LV3 to VLS0LV0 bits of the VLS0LV register.
(2) Choose "Sampling with HSCLK" or "Sampling with LSCLK" by the VLS0SM1 and VLS0SM0 bits of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV2 to VLS0DIV0 bits of the VLS0SMP register.
(3) Write "0x0" to VLS0AMD1 and VLS0AMD0 bits of VLS0MOD register in order to choose the single mode 1.
(4) Write “0x2” to VLS0SEL1 and VLS0SEL0 bits of the VLS0MOD register in order to enable the VLS0 interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS0 operation.
(6) Wait until the comparison result of the VLS0 comparator is stabilized (approx. 300 μs).
(7) If V_{DD} is below the threshold voltage (V_{VLSF}) after three cycles of the sampling clock, the VLS0F bit is set to "1" and the VLS0 interrupt (detection complete) is generated. The VLS0 interrupt (detection complete) is generated regardless of the detection result of V_{DD}.
(8) After the interrupt is generated, the VLS0EN bit is cleared to "0" and VLS0 operation is disabled.
(9) Read the VLS0F bit to confirm the detection result.

[Note]

- **Entering the STOP/STOP-D mode is not allowed while the single mode operation is in progress. Enter the STOP/STOP-D mode after the single mode operation is completed (VLS0EN bit="0").**

22.3.2.2 Single mode 2

The single mode 2 generates the interrupt when the V_{DD} is lower than the threshold voltage.
Figure 22-10 shows an example of the operation timing chart without sampling in single mode 2.

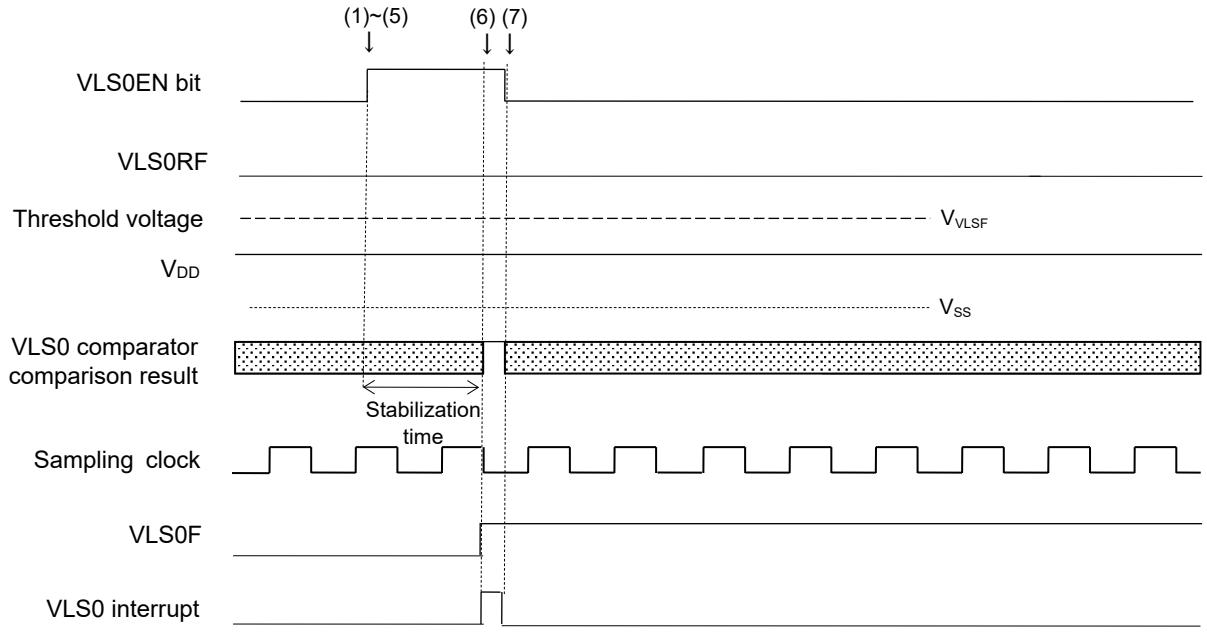


Figure 22-10 Operation Timing Chart without Sampling (Single Mode 2)

The operation shown in Figure 22-10 is described below:

(1) Choose a detection voltage by the VLS0LV3 to VLS0LV0 bits of the VLS0LV register.
(2) Choose "No sampling" by the VLS0SM1 and VLS0SM0 bits of the VLS0SMP register.
(3) Write "0x1" to VLS0AMD1 and VLS0AMD0 bits of VLS0MOD register in order to choose the single mode 2.
(4) Write “0x2” to VLS0SEL1 and VLS0SEL0 bits of the VLS0MOD register in order to enable the VLS0 interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS0.
(6) If V_{DD} is below the specified threshold voltage (V_{VLSF}) when the comparison result of the VLS0 comparator is stabilized, voltage level supervisor flag (VLS0F) is set to "1" and the VLS0 interrupt (low voltage) is generated. If V_{DD} is higher than the specified threshold voltage (V_{VLSF}), the VLS0F bit is cleared to "0" and the VLS0 interrupt (low voltage) is not generated.
(7) The VLS0EN bit is set to "0" and VLS0 is disabled regardless of whether the VLS0 interrupt occurs or not.

Figure 22-11 shows an example of the operation timing chart with sampling in single mode 2.

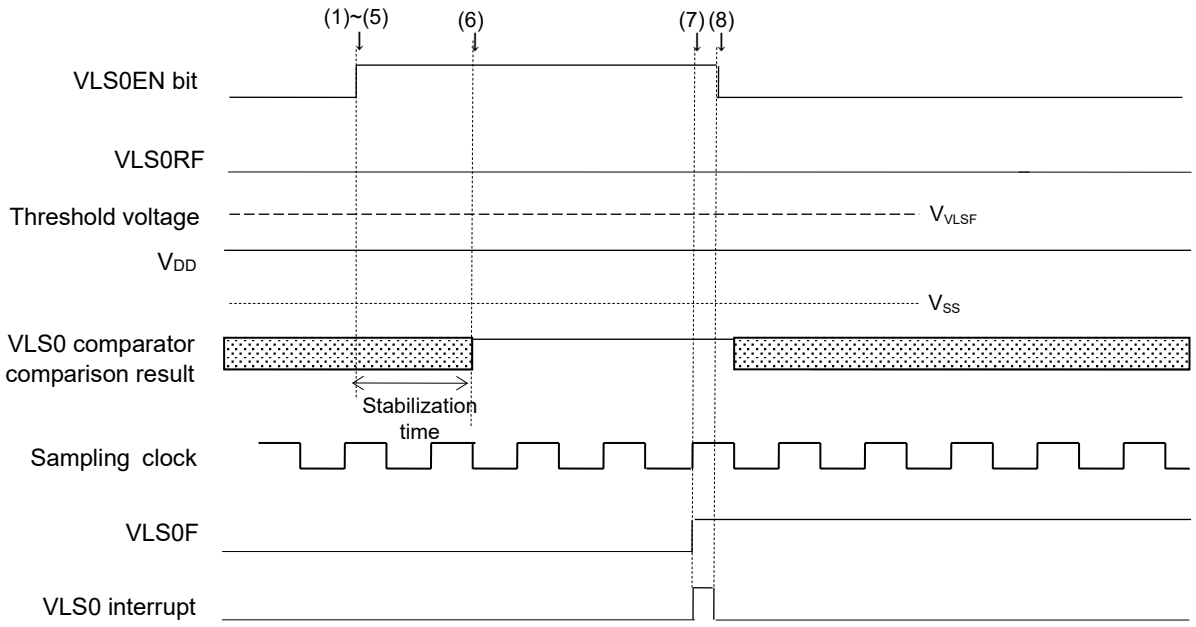


Figure 22-11 Operation Timing Chart with Sampling (Single Mode 2)

The operation shown in Figure 22-11 is described below:

(1) Choose a detection voltage by the VLS0LV3 to VLS0LV0 bits of the VLS0LV register.
(2) Choose "Sampling with HSCLK" or "Sampling with LSCLK" by the VLS0SM1 and VLS0SM0 bits of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV2 to VLS0DIV0 bits of the VLS0SMP register.
(3) Write "0x1" to VLS0AMD1 and VLS0AMD0 bits of VLS0MOD register in order to choose the single mode 2.
(4) Write “0x2” to VLS0SEL1 and VLS0SEL0 bits of the VLS0MOD register in order to enable the VLS0 interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS0 operation.
(6) Wait until the comparison result of the VLS0 comparator is stabilized (approx. 300 μs).
(7) If V_{DD} is below the threshold voltage (V_{VLSF}) after three cycles of the sampling clock, the VLS0F bit is set to "1" and the VLS0 interrupt (low voltage) is generated. If V_{DD} is equal to or above the threshold voltage (V_{VLSF}), the VLS0F bit is cleared to "0" and the VLS0 interrupt (low voltage) is not generated.
(8) The VLS0EN bit is set to "0" and VLS0 is disabled regardless of whether the VLS0 interrupt occurs or not.

[Note]
- **Entering the STOP/STOP-D mode is not allowed while the single mode operation is in progress. Enter the STOP/STOP-D mode after the single mode operation is completed (VLS0EN bit="0").**
- **If V_{DD} is higher than the specified threshold voltage, the VLS0 interrupt is not generated.**

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), converts an analog input level to a digital value.
The number of A/D Converter channels is dependent of the product specification.
Table 23-1 shows the number of channels.

Table 23-1 Number of A/D Converter channels

Channel no.	ML62Q1300 group				ML62Q1500 / ML62Q1800 / ML62Q1700 group				
	16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
0	●	●	●	●	●	●	●	●	●
1	●	●	●	●	●	●	●	●	●
2	●	●	●	●	●	●	●	●	●
3	●	●	●	●	●	●	●	●	●
4	−	●	●	●	●	●	●	●	●
5	−	●	●	●	●	●	●	●	●
6	●	●	●	●	●	●	●	●	●
7	●	●	●	●	●	●	●	●	●
8	−	−	−	−	●	●	●	●	●
9	−	−	−	−	●	●	●	●	●
10	−	−	−	−	●	●	●	●	●
11	−	−	−	−	●	●	●	●	●
12	−	−	−	−	−	−	−	●	●
13	−	−	−	−	−	−	−	●	●
14	−	−	−	−	−	−	−	●	●
15	−	−	−	−	−	−	−	●	●

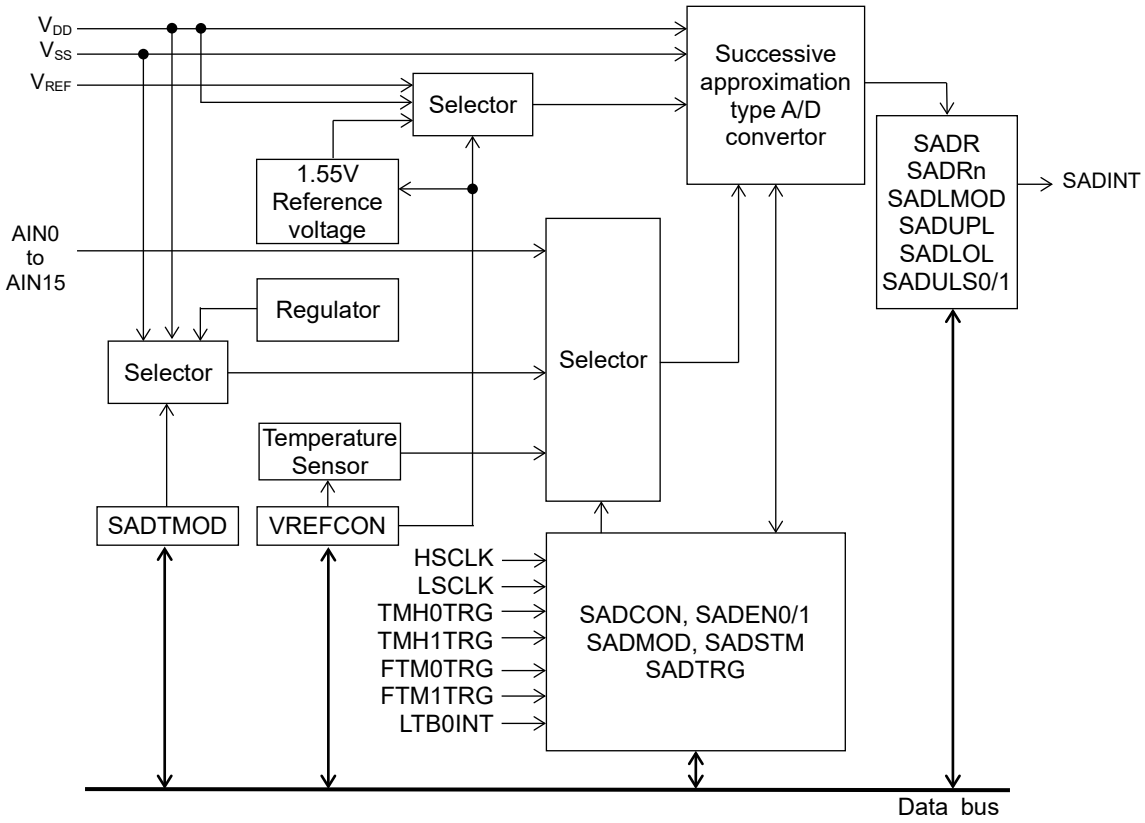
●: Available - : Unavailable

23.1.1 Features

- Resolution : 10bit
- Conversion time : Min. 2.25µs/channel (conversion clock is 8MHz)
- Number of input channel : Max. 16ch
- Reference voltage: Voltage input from the V_{DD} pin, Internal reference voltage(approx.1.55V) or External reference voltage(V_{REF} pin)
- Sampling time can be chosen
- Consecutive scan conversion function for target channels
- Consecutive scan conversion with a specific interval time
- One conversion result register for each channel
- Upper /Lower limit is configurable for the conversion result, generates an interrupt
- A built-in temperature sensor usable for the low-speed RC oscillation adjustment
- A/D converter self test function (full scale, zero scale, internal reference voltage)
- Following triggers is available to start the A/D conversion
  - 16-bit Timer 0 trigger (TMH0TRG)
  - 16-bit Timer 1 trigger (TMH1TRG)
  - Functional Timer 0 trigger (FTM0TRG)
  - Functional Timer 1 trigger (FTM1TRG)
  - Low-speed Time Base Counter interrupt (LTB0INT)

23.1.2 Configuration

Figure 23-1 shows the configuration of SA-ADC.



SADCON : SA-ADC control register
SADEN0/1 : SA-ADC enable register 0, 1
SADMOD : SA-ADC mode register
SADSTM : SA-ADC scan conversion interval setting register
SADR : SA-ADC result register
SADRn : SA-ADC result register n (n = 0 to 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=0,1)
VREFCON : Reference voltage control register
SADTRG : SA-ADC trigger register
SADTMOD : SA-ADC test mode register
SADINT : SA-ADC interrupt request, SA-ADC DMA request
TMH0TRG, TMH1TRG : 16-bit Timer 0, 1 interrupt
FTM0TRG, FTM1TRG : Functional Timer 0, 1 interrupt
LTB0INT : Low speed time base counter 0 interrupt request

Figure 23-1 Configuration of successive approximation type A/D Converter

23.1.3 List of Pins

The I/O pins of the Successive Approximation type A/D converter are assigned to the shared function of the general ports.

Pin name	I/O	Description
V_{DD}	-	Positive power supply for SA-ADC
V_{SS}	-	Negative power supply for SA-ADC
V_{REF}	-	Reference power supply for SA-ADC
AIN0	I	SA-ADC channel 0 analog input
AIN1	I	SA-ADC channel 1 analog input
AIN2	I	SA-ADC channel 2 analog input
AIN3	I	SA-ADC channel 3 analog input
AIN4	I	SA-ADC channel 4 analog input
AIN5	I	SA-ADC channel 5 analog input
AIN6	I	SA-ADC channel 6 analog input
AIN7	I	SA-ADC channel 7 analog input
AIN8	I	SA-ADC channel 8 analog input
AIN9	I	SA-ADC channel 9 analog input
AIN10	I	SA-ADC channel 10 analog input
AIN11	I	SA-ADC channel 11 analog input
AIN12	I	SA-ADC channel 12 analog input
AIN13	I	SA-ADC channel 13 analog input
AIN14	I	SA-ADC channel 14 analog input
AIN15	I	SA-ADC channel 15 analog input

Table 23-2 shows the list of the general ports used for the A/D Converter and the register settings of the ports.

Table 23-2 Ports used in the A/D Converter and the register settings

Channel no.	Pin name	Shared port		Setting Register	Setting value	ML62Q1300 group				ML62Q1500/ML62Q1800 ML62Q1700 group				
						16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product
0	AIN0	P17	8th Func.	P1MOD7	0111_0000	●	●	●	●	●	●	●	●	●
1	AIN1	P20	8th Func.	P2MOD0	0111_0000	●	●	●	●	●	●	●	●	●
2	AIN2	P21	8th Func.	P2MOD1	0111_0000	●	●	●	●	●	●	●	●	●
3	AIN3	P22	8th Func.	P2MOD2	0111_0000	●	●	●	●	●	●	●	●	●
4	AIN4	P24	8th Func.	P2MOD4	0111_0000	-	●	●	●	●	●	●	●	●
5	AIN5	P25	8th Func.	P2MOD5	0111_0000	-	●	●	●	●	●	●	●	●
6	AIN6	P26	8th Func.	P2MOD6	0111_0000	●	●	●	●	●	●	●	●	●
7	AIN7	P27	8th Func.	P2MOD7	0111_0000	●	●	●	●	●	●	●	●	●
8	AIN8	P65	8th Func.	P6MOD5	0111_0000	-	-	-	-	●	●	●	●	●
9	AIN9	P66	8th Func.	P6MOD6	0111_0000	-	-	-	-	●	●	●	●	●
10	AIN10	P43	8th Func.	P4MOD3	0111_0000	-	-	-	-	●	●	●	●	●
11	AIN11	P03	8th Func.	P0MOD3	0111_0000	-	-	-	-	●	●	●	●	●
12	AIN12	P56	8th Func.	P5MOD6	0111_0000	-	-	-	-	-	-	-	●	●
13	AIN13	P57	8th Func.	P5MOD7	0111_0000	-	-	-	-	-	-	-	●	●
14	AIN14	PA3	8th Func.	PAMOD3	0111_0000	-	-	-	-	-	-	-	●	●
15	AIN15	PA4	8th Func.	PAMOD4	0111_0000	-	-	-	-	-	-	-	●	●

●: Available - : Unavailable

[Note]

- **When using the SA-ADC, set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 0 to 9, A, B, m: bit number 0 to 7) to "0" as "Disable input" and "Disable output", otherwise a shoot-through current may flow.**
- **While the A/D converter is operating, an influence of the noise is reducible by preventing the switching of neighboring pins or A/D converting in the HALT mode.**

23.2 Description of Registers

23.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
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	SA-ADC result register 8	SADR8L	SADR8	R	8/16	0x00
0xF811		SADR8H		R	8	0x00
0xF812	SA-ADC result register 9	SADR9L	SADR9	R	8/16	0x00
0xF813		SADR9H		R	8	0x00
0xF814	SA-ADC result register 10	SADR10L	SADR10	R	8/16	0x00
0xF815		SADR10H		R	8	0x00
0xF816	SA-ADC result register 11	SADR11L	SADR11	R	8/16	0x00
0xF817		SADR11H		R	8	0x00
0xF818	SA-ADC result register 12	SADR12L	SADR12	R	8/16	0x00
0xF819		SADR12H		R	8	0x00
0xF81A	SA-ADC result register 13	SADR13L	SADR13	R	8/16	0x00
0xF81B		SADR13H		R	8	0x00
0xF81C	SA-ADC result register 14	SADR14L	SADR14	R	8/16	0x00
0xF81D		SADR14H		R	8	0x00
0xF81E	SA-ADC result register 15	SADR15L	SADR15	R	8/16	0x00
0xF81F		SADR15H		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

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
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
0xF832	SA-ADC conversion interval setting 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	-	-	-	-	-
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	-	-	-	-	-
0xF0BA	SA-ADC test mode	SADTMOD	-	R/W	8	0x00
0xF0BB	Reserved	-	-	-	-	-

[Note]

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

23.2.2 SA-ADC Result Register n (SADRn : n=0 to 15, 16)

SADRn is a special function register (SFR) used to store the SA-ADC conversion results on channels 0 to 15 and channel 16 (temperature sensor).
The A/D conversion result of each channel can be read from SADRn.

Symbol name	Channel
SADR0	The conversion result of channel 0 (AIN0)
SADR1	The conversion result of channel 1 (AIN1)
SADR2	The conversion result of channel 2 (AIN2)
SADR3	The conversion result of channel 3 (AIN3)
SADR4	The conversion result of channel 4 (AIN4)
SADR5	The conversion result of channel 5 (AIN5)
SADR6	The conversion result of channel 6 (AIN6)
SADR7	The conversion result of channel 7 (AIN7)
SADR8	The conversion result of channel 8 (AIN8)
SADR9	The conversion result of channel 9 (AIN9)
SADR10	The conversion result of channel 10 (AIN10)
SADR11	The conversion result of channel 11 (AIN11)
SADR12	The conversion result of channel 12 (AIN12)
SADR13	The conversion result of channel 13 (AIN13)
SADR14	The conversion result of channel 14 (AIN14)
SADR15	The conversion result of channel 15 (AIN15)
SADR16	The conversion result of channel 16 (temperature sensor)

Address: 0xF800(SADR0L/SADR0), 0xF801(SADR0H),
0xF802(SADR1L/SADR1), 0xF803(SADR1H),
0xF804(SADR2L/SADR2), 0xF805(SADR2H),
0xF806(SADR3L/SADR3), 0xF807(SADR3H),
0xF808(SADR4L/SADR4), 0xF809(SADR4H),
0xF80A(SADR5L/SADR5), 0xF80B(SADR5H),
0xF80C(SADR6L/SADR6), 0xF80D(SADR6H),
0xF80E(SADR7L/SADR7), 0xF80F(SADR7H),
0xF810(SADR8L/SADR8), 0xF811(SADR8H),
0xF812(SADR9L/SADR9), 0xF813(SADR9H),
0xF814(SADR10L/SADR10), 0xF815(SADR10H),
0xF816(SADR11L/SADR11), 0xF817(SADR11H),
0xF818(SADR12L/SADR12), 0xF819(SADR12H),
0xF81A(SADR13L/SADR13), 0xF81B(SADR13H),
0xF81C(SADR14L/SADR14), 0xF81D(SADR14H),
0xF81E(SADR15L/SADR15), 0xF81F(SADR15H),
0xF820(SADR16L/SADR16), 0xF821(SADR16H)
Access: R
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADRn															
Byte	SADRnH								SADRnL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.2.3 SA-ADC Result Register (SADR)

SADR is a read-only special function register (SFR) used to store the A/D conversion results on channels 0 to 15, 16 (temperature sensor) and 17 (A/D converter test function).
The A/D conversion results of all channels are stored to this register. The result of each channel is overwritten.
Use this register when transferring conversion results on multiple channels to RAM using the DMA controller.
The A/D conversion test result on channel 17 is stored to this register only.

Symbol name	Channel
SADR	Conversion results of channels 0 to 15, 16 and 17

Address: 0xF822(SADRL/SADR), 0xF823(SADRH)
Access: R
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADR															
Byte	SADRH								SADRL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.2.4 SA-ADC Upper/Lower Limit Status Register 0 (SADULS0)

SAULS0 is a special function register (SFR) used to indicate whether the A/D conversion result matches to the condition of upper/lower limit.

Address: 0xF824(SADULS0L/SADULS0), 0xF825(SADULS0H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADULS0															
Byte	SADULS0H								SADULS0L							
Bit	SAULS 15	SAULS 14	SAULS 13	SAULS 12	SAULS 11	SAULS 10	SAULS 09	SAULS 08	SAULS 07	SAULS 06	SAULS 05	SAULS 04	SAULS 03	SAULS 02	SAULS 01	SAULS 00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	SAULS15 to SAULS00	These bits are used to indicate whether the A/D conversion results of channels 0 to 15 matches to the condition of upper/lower limit. 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 any bit of SADULS0/1 is “1” when all chosen channels A/D conversion are completed. Refer to Figure 23-7 for the timing of the interrupt and updates of detection result. Each bit is forcibly cleared to "0" by writing 1 to each bit. The writing "0" does not clear the bit. SAULS00: Detection result for the upper/lower limit on channel 0 (AIN0) SAULS01: Detection result for the upper/lower limit on channel 1 (AIN1) SAULS02: Detection result for the upper/lower limit on channel 2 (AIN2) SAULS03: Detection result for the upper/lower limit on channel 3 (AIN3) SAULS04: Detection result for the upper/lower limit on channel 4 (AIN4) SAULS05: Detection result for the upper/lower limit on channel 5 (AIN5) SAULS06: Detection result for the upper/lower limit on channel 6 (AIN6) SAULS07: Detection result for the upper/lower limit on channel 7 (AIN7) SAULS08: Detection result for the upper/lower limit on channel 8 (AIN8) SAULS09: Detection result for the upper/lower limit on channel 9 (AIN9) SAULS10: Detection result for the upper/lower limit on channel 10 (AIN10) SAULS11: Detection result for the upper/lower limit on channel 11 (AIN11) SAULS12: Detection result for the upper/lower limit on channel 12 (AIN12) SAULS13: Detection result for the upper/lower limit on channel 13 (AIN13) SAULS14: Detection result for the upper/lower limit on channel 14 (AIN14) SAULS15: Detection result for the upper/lower limit on channel 15 (AIN15) 0: The A/D conversion result unmatched to the condition of upper/lower limit (SALMD1 to 0) (Initial value) 1: The A/D conversion result matched to the condition of upper/lower limit (SALMD1 to 0)

[Note]

- **Do not use bit access instructions and use word or byte access instructions for writing this register.**
- **When using the A/D conversion result upper/lower limit detection function (SALEN bit =1), the interrupt can be cleared by clearing the corresponding bit of SAULS15 to SAULS00 or by resetting the LSI.**
- **When performing the A/D conversion only one time (SALP bit =0), confirm the bit of SAULS15 to SAULS00 is "0" before setting SARUN bit to "1".**
- **When performing the consecutive scan A/D conversion (SALP bit =1), confirm the bit of SAULS15 to SAULS00 is "0", before the next A/D conversion ends.**

23.2.5 SA-ADC Upper/Lower Limit Status Register 1 (SADULS1)

SAULS1 is a special function register (SFR) used to indicate whether the A/D conversion result matches to the condition of upper/lower limit on channel 16.

Address: 0xF826(SADULS1L/SADULS1), 0xF827(SADULS1H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADULS1															
Byte	SADULS1H								SADULS1L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SAULS16
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 1	-	Reserved bits
0	SAULS16	This bit is used to indicate whether the A/D conversion results of channels 16 (temperature sensor) matches to the condition of upper/lower limit. 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 any bit of SADULS0/1 is “1” when all chosen channels A/D conversion are completed. Refer to Figure 23-7 for the timing of the interrupt and updates of detection result. The SAULS16 bit is forcibly cleared to "0" by writing 1 to this bit. The writing "0" does not clear the bit. 0: The A/D conversion result unmatched to the condition of upper/lower limit (SALMD1 to 0) (Initial value) 1: The A/D conversion result matched to the condition of upper/lower limit (SALMD1 to 0)

[Note]
- **Do not use bit access instructions and use word or byte access instructions for writing this register.**
- **When using the A/D conversion result upper/lower limit detect function (SALEN bit =1), the interrupt can be cleared by clearing the corresponding bit of SAULS16 or by resetting the LSI.**
- **When performing the A/D conversion only one time (SALP bit =0), confirm the bit of SAULS16 is "0" before setting SARUN bit to "1".**
- **When performing the consecutive scan A/D conversion (SALP bit =1), confirm the bit of SAULS16 is "0", before the next A/D conversion ends.**

23.2.6 SA-ADC Mode Register (SADMOD)

SADMOD is a special function register (SFR) used to set the operation mode and operating clock frequency of the A/D converter.

Address: 0xF828(SADMODL/SADMOD), 0xF829(SADMODH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADMOD															
Byte	SADMODH								SADMODL							
Bit	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	SAINIT	SASHT3	SASHT2	SASHT1	SASHT0	SACK2	SACK1	SACK0	SALP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 9	rsvd	Reserved bits. Write "0" to these bits.
8	SAINIT	This 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 the next SA-ADC conversion. 0: Start the A/D conversion without discharging the electrical charge accumulated in the sample hold capacitor (Initial value) 1: Start the A/D conversion after discharging the electrical charge accumulated in the sample hold capacitor
7 to 4	SASHT3 to SASHT0	These bits are used to set the sampling time. See Chapter 23.3.3 "A/D Conversion Time Setting" for details.
3 to 1	SACK2 to SACK0	These bits are used to choose the frequency of the A/D conversion operating clock (SAD_CLK). See Chapter 23.3.2 "A/D Conversion Time Setting" for the operating clock, A/D conversion time and sample time. 000: 8MHz (Initial value) 001: 4MHz 010: 2MHz 011: 1MHz 100: 0.5MHz 101: Do not use 110: Do not use 111: 32kHz
0	SALP	This bit is used to choose whether the A/D conversion is performed once only for each channel or consecutively. The conversion interval time in the consecutive scan A/D conversion mode is specified in the SADSTM register. 0: Single A/D conversion (Initial value) 1: Consecutive scan A/D conversion

[Note]

- **Write "0" to the SADMODH[7:1] bits. The operation when "1" is written to the bits is unguaranteed.**

23.2.7 SA-ADC Control Register (SADCON)

SADCON is a special function register (SFR) used to control the operation of the A/D converter.

Address: 0xF82A(SADCONL/SADCON), 0xF82B(SADCONH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

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

Bit No	Bit symbol name	Description
15 to 2	-	Reserved bits
1	SATGEN	This bit is used to enable starting the A/D conversion by the trigger events. 0: Disable the trigger operation (Initial value) 1: Enable the trigger operation
0	SARUN	This bit is used to start or stop the A/D conversion. Write "1" to this bit to start the A/D conversion, and "0" to stop it. When "0" is written to SALP bit and the A/D conversion on the largest number of channel is ended, this SARUN bit is automatically reset to "0". When "1" is written to SALP, the A/D conversion repeats until the SARUN bit is reset to "0" by the software. 0: Stop the A/D conversion (Initial value) 1: Start the A/D conversion

[Note]

- **Start the A/D conversion with one or more channels chosen by the SA-ADC enable registers (SADEN0 and SADEN1). If no channel is chosen, the operation does not start.**
- **Enter STOP/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 to 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)

SADEN0 is a special function register (SFR) used to choose channels of the A/D converter and enable/disable the conversion.

Address: 0xF82C(SADEN0L/SADEN0), 0xF82D(SADEN0H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADEN0															
Byte	SADEN0H								SADEN0L							
Bit	SACH 15	SACH 14	SACH 13	SACH 12	SACH 11	SACH 10	SACH 09	SACH 08	SACH 07	SACH 06	SACH 05	SACH 04	SACH 03	SACH 02	SACH 01	SACH 00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	SACH15 to SACH00	These bits are used to choose channel n (n=0 to 15) of the A/D converter and enable/disable the conversion. 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 SACH08: Enable or Disable the A/D conversion on channel 8 SACH09: Enable or Disable the A/D conversion on channel 9 SACH10: Enable or Disable the A/D conversion on channel 10 SACH11: Enable or Disable the A/D conversion on channel 11 SACH12: Enable or Disable the A/D conversion on channel 12 SACH13: Enable or Disable the A/D conversion on channel 13 SACH14: Enable or Disable the A/D conversion on channel 14 SACH15: Enable or Disable the A/D conversion on channel 15 0: Disable the conversion on channel n (initial value) 1: Enable the conversion on channel n

[Note]

- **When multiple bits of SACHn (n=00 to 17) 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 bits of SACHn (n=00 to 17) are "0". In that case SARUN bit of SADCON register does not get to "1".**

23.2.9 SA-ADC Enable Register 1 (SADEN1)

SADEN1 is a special function register (SFR) used to choose channels of the A/D converter and enable/disable the conversion.

Address: 0xF82E(SADEN1L/SADEN1), 0xF82F(SADEN1H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADEN1															
Byte	SADEN1H								SADEN1L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SACH 17	SACH 16
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 2	-	Reserved bits
1, 0	SACH17, SACH16	These bits are used to choose channel n (n=16, 17) of the A/D converter and enable/disable the conversion. 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) 0: Disable the conversion on channel n (initial value) 1: Enable the conversion on channel n

[Note]

- **When multiple bits of SACHn (n=00 to 17) 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 bits of SACHn (n=00 to 17) are "0". In that case the SARUN bit does not get to "1".**
- **Channel 16 (SACH16) is used for adjusting frequency of the low-speed RC oscillation clock. When using the channel 16 (SACH16), enable the internal reference voltage/temperature sensor and choose the internal reference voltage by setting VREFCON register.**

23.2.10 SA-ADC Conversion Interval Setting Register (SADSTM)

SADSTM is a special function register (SFR) used to set the interval time in the consecutive scan A/D conversion mode. The interval time is determined by the following formula.

A/D conversion interval time = HSCLK cycle x SADSTM setting value

For an example, supposing to A/D convert channel 2 and channel 5, the A/D conversion interval time means the time after the channel 2 and channel 5 are A/D converted consecutively and before the A/D conversion of channel 2 is started. The next A/D conversion starts at the timing that the value set in this register has been counted with HSCLK.

Address: 0xF832(SADSTML/SADSTM), 0xF833(SADSTMH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADSTM															
Byte	SADSTMH								SADSTML							
Bit	SADSTM15	SADSTM14	SADSTM13	SADSTM12	SADSTM11	SADSTM10	SADSTM9	SADSTM8	SADSTM7	SADSTM6	SADSTM5	SADSTM4	SADSTM3	SADSTM2	SADSTM1	SADSTM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

- **When SACK2-0 bits of SADMOD register is "111", the interval time is always minimum (0ns).**

23.2.11 SA-ADC Upper/Lower Limit Mode Register (SADLMOD)

SADLMOD is a special function register (SFR) used to set modes in the A/D conversion result upper/lower limit detection function.

Address: 0xF834(SADLMODL/SADLMOD), 0xF835(SADLMODH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADLMOD															
Byte	SADLMODH								SADLMODL							
Bit	-	-	-	-	-	-	SALMD1	SALMD0	-	-	-	-	-	-	-	SALEN
R/W	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

Bit No.	Bit symbol name	Description
15 to 10	-	Reserved bits
9, 8	SALMD1, SALMD0	These 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 interrupt request. 00: SADLOL value ≤ A/D conversion value ≤ SADUPL value (Initial value) 01: A/D conversion value > SADUPL value 10: A/D conversion value < SADLOL value 11: A/D conversion value > SADUPL or A/D conversion value < SADLOL value
7 to 1	-	Reserved bits
0	SALEN	This 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". 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

[Note]

- **The upper/lower limit detection function is available to make the interrupt request for the A/D conversion result on all chosen 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)

SADUPL is a special function register (SFR) used to set the upper limit of A/D conversion result.

Address: 0xF836(SADUPLL/SADUPL), 0xF836(SADUPLH)
Access: R/W
Access size: 8/16bit
Initial value: 0xFFC0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADUPL															
Byte	SADUPLH								SADUPLL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

23.2.13 SA-ADC Lower Limit Setting Register (SADLOL)

SADLOL is a special function register (SFR) used to set the lower limit of A/D conversion result.

Address: 0xF838(SADLOLL/SADLOL), 0xF838(SADLOLH)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADLOL															
Byte	SADLOLH								SADLOLL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.2.14 Reference Voltage Control Register (VREFCON)

VREFCON is a special function register (SFR) used to choose the internal reference voltage operation and control the operation of the temperature sensor,.

Address: 0xF83A(VREFCON)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								VREFCON							
Bit	-	-	-	-	-	-	-	-	-	-	VREFP1	VREFP0	-	-	-	VREFEN
R/W	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

Bit No.	Bit symbol name	Description
7, 6	-	Reserved bits
5, 4	VREFP1, VREFP0	These bits are used to choose the reference voltage for the A/D conversion. 00: Voltage input from the V_{DD} pin (Initial value) 01: Voltage input from the V_{REF} pin 10: Voltage generated by the internal reference voltage circuit (approx.1.55V) 11: Do not use (Voltage input from the V_{DD} pin)
3 to 1	-	Reserved bits
0	VREFEN	This 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". 0: Disable the operation of internal reference voltage and temperature sensor (Initial value) 1: Enable the operation of internal reference voltage and temperature sensor

[Note]

- **It takes 200μs(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 possible 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.**
- **The internal reference voltage controlled by the VREFEN bit is for the A/D converter. See chapter 20 "Analog Comparator" for the reference voltage used in the analog comparator.**

23.2.15 SA-ADC Interrupt Mode Register (SADIMOD)

SADIMOD is a special function register (SFR) used to choose the interrupt mode of the SA-ADC.

Address: 0xF83C(SADIMOD)
Access: R/W
Access size: 8bit
Initial value: 0x00

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

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	SADIMD	This bit is used to choose the occurrence timing of SA-ADC interrupt request. 0: Make the interrupt request after the A/D conversion is completed on all channels (Initial value) 1: Make the interrupt request whenever the A/D conversion is completed on each channel

[Note]

- **If SALEN bit of the SA-ADC upper/lower limit mode register (SADLMOD) is set to "1", the interrupt by the upper/lower limit detection function gets enabled and the setting for the SADIMD bit of SA-ADC Interrupt Mode Register (SADIMOD) gets invalid.**

23.2.16 SA-ADC Trigger Register (SADTRG)

SADTRG is a special function register (SFR) used to control the trigger event for the SA-ADC.

Address: 0xF83E(SADTRG)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SADTRG							
Bit	-	-	-	-	-	-	-	-	-	-	-	SASTS4	SASTS3	SASTS2	SASTS1	SASTS0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 5	-	Reserved bits
4 to 0	SASTS4 to SASTS0	These bits are used to choose the source of the trigger event for SA-ADC. 00000: 16-bit Timer 0 trigger (TMH0TRG) (Initial value) 00001: 16-bit Timer 1 trigger (TMH1TRG) 00010: Do not use (Reserved) 00011: Do not use (Reserved) 00100: Functional Timer 0 trigger (FTM0TRG) 00101: Functional Timer 1 trigger (FTM1TRG) 00110: Do not use (Reserved) 00111: Do not use (Reserved) 01000: Time Base Counter 0 interrupt (LTB0INT) 01001 to 11111: Do not use (Reserved)

[Note]

- **When choosing the 16-bit Timer trigger (TMH0TRG and TMH1TRG), set the THn8BM bit of the 16bit Timer n Mode Register (TMHnMOD) to "0" to choose the 16bit timer mode.**

23.2.17 SA-ADC test mode register (SADTMOD)

SADTMOD is a register (SFR) used to control the SA-ADC 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 (approx. 1.55 V).
The A/D conversion result is stored in the SA-ADC result register (SADR).

Address: 0xF0BA(SADTMOD)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SADTMOD							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SADTM1	SADTM0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1 to 0	SADTM1, SADTM0	These bits are used to set the successive approximation type A/D converter test function. 00: Do not use the A/D converter test function (Initial value) 01: Full scale A/D conversion 10: Zero scale A/D conversion 11: Internal reference voltage (approx.1.55V) A/D conversion

23.3 Description of Operation

23.3.1 Operation of Successive Approximation Type A/D Converter

Figure 23-2 shows a setting example when one-time A/D conversion is performed using channel 1 and 0.

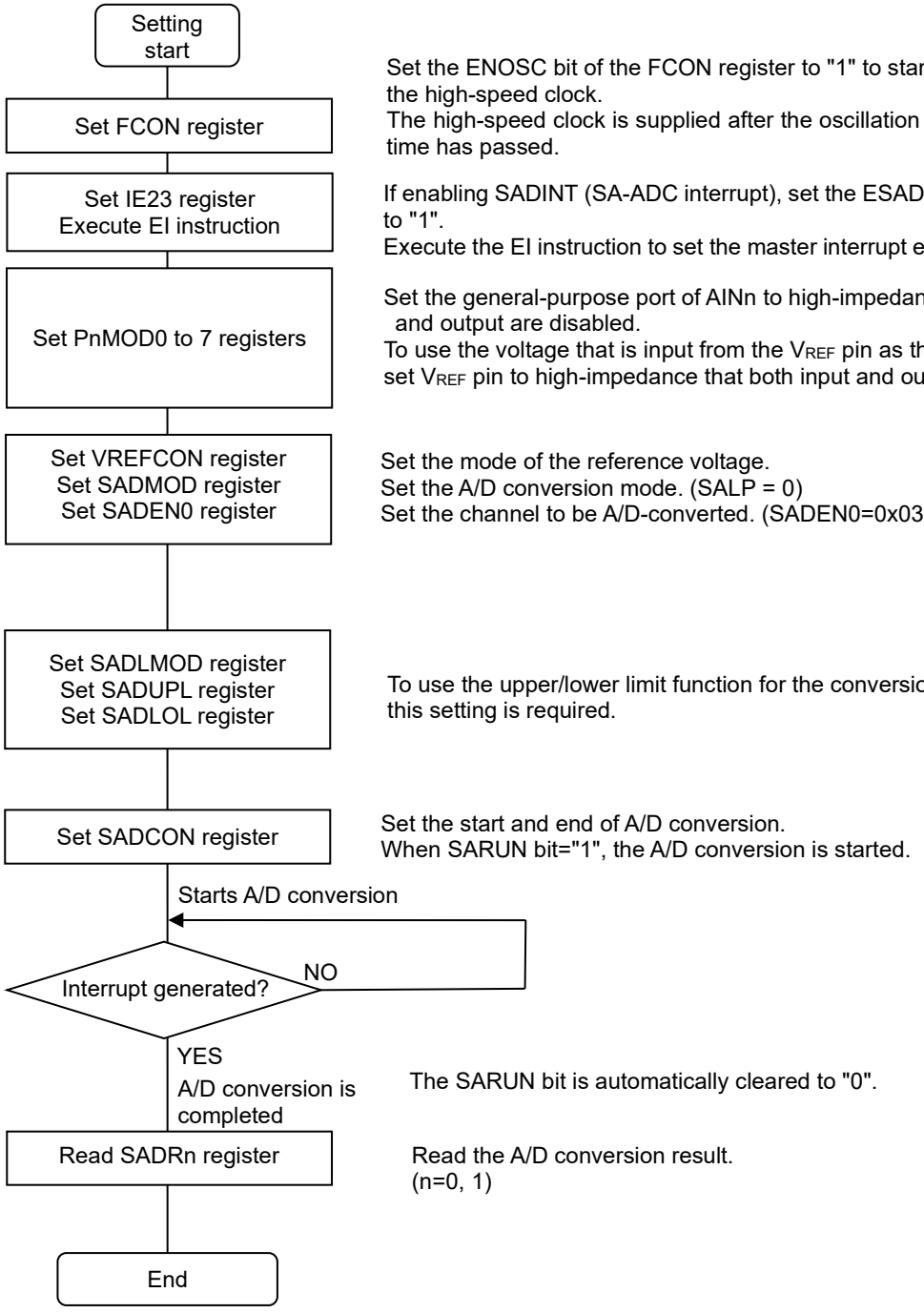


Figure 23-2 Example of A/D Conversion Setting

Figure 23-3 shows a setting example when one-time A/D conversion is performed in HALT mode using channel 1 and 0.

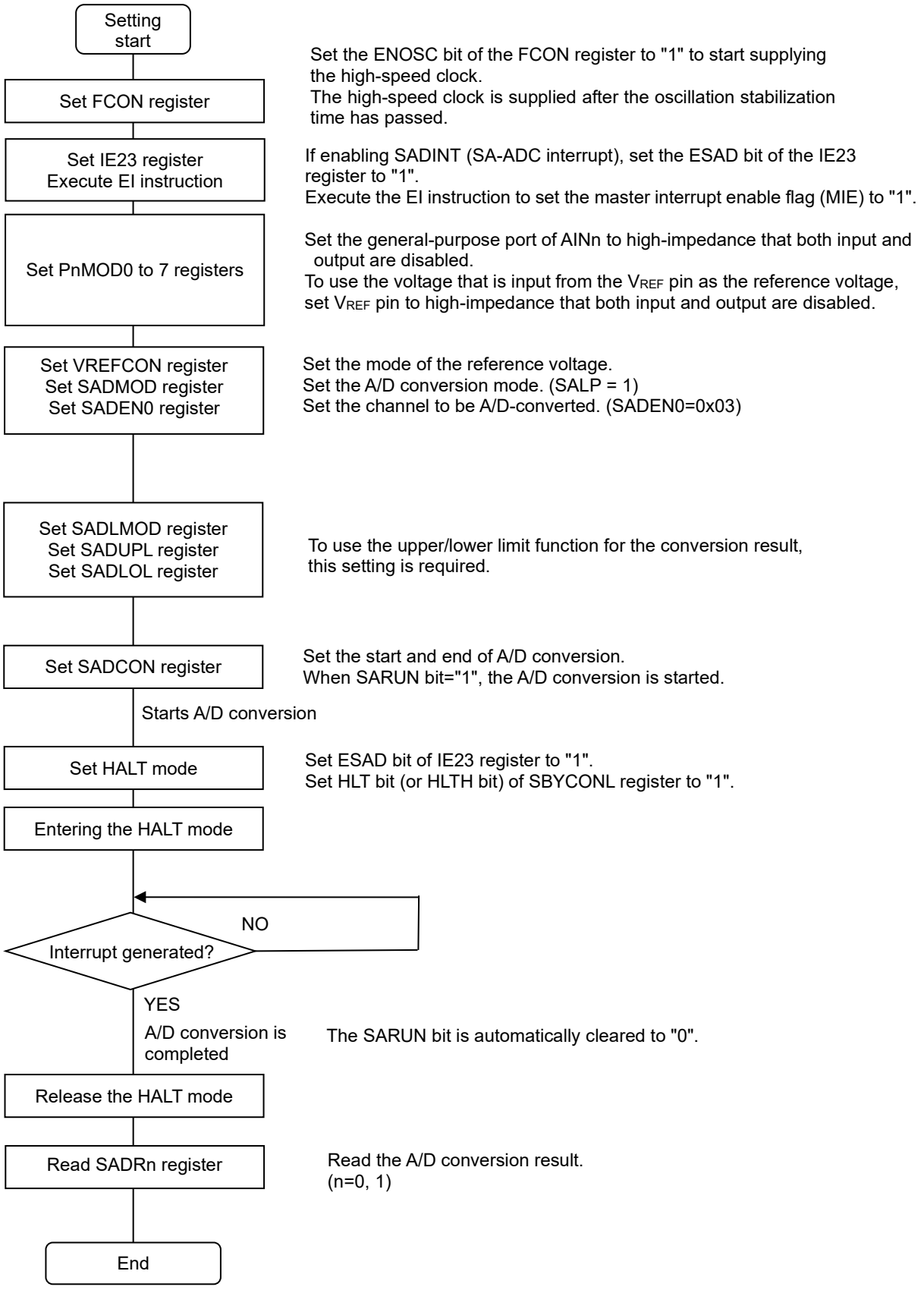


Figure 23-3 Example of A/D Conversion Setting (Converting in HALT mode)

Figure 23-4 shows a setting example when one-time A/D conversion is performed using channel 1 and 0 starting by a trigger event.

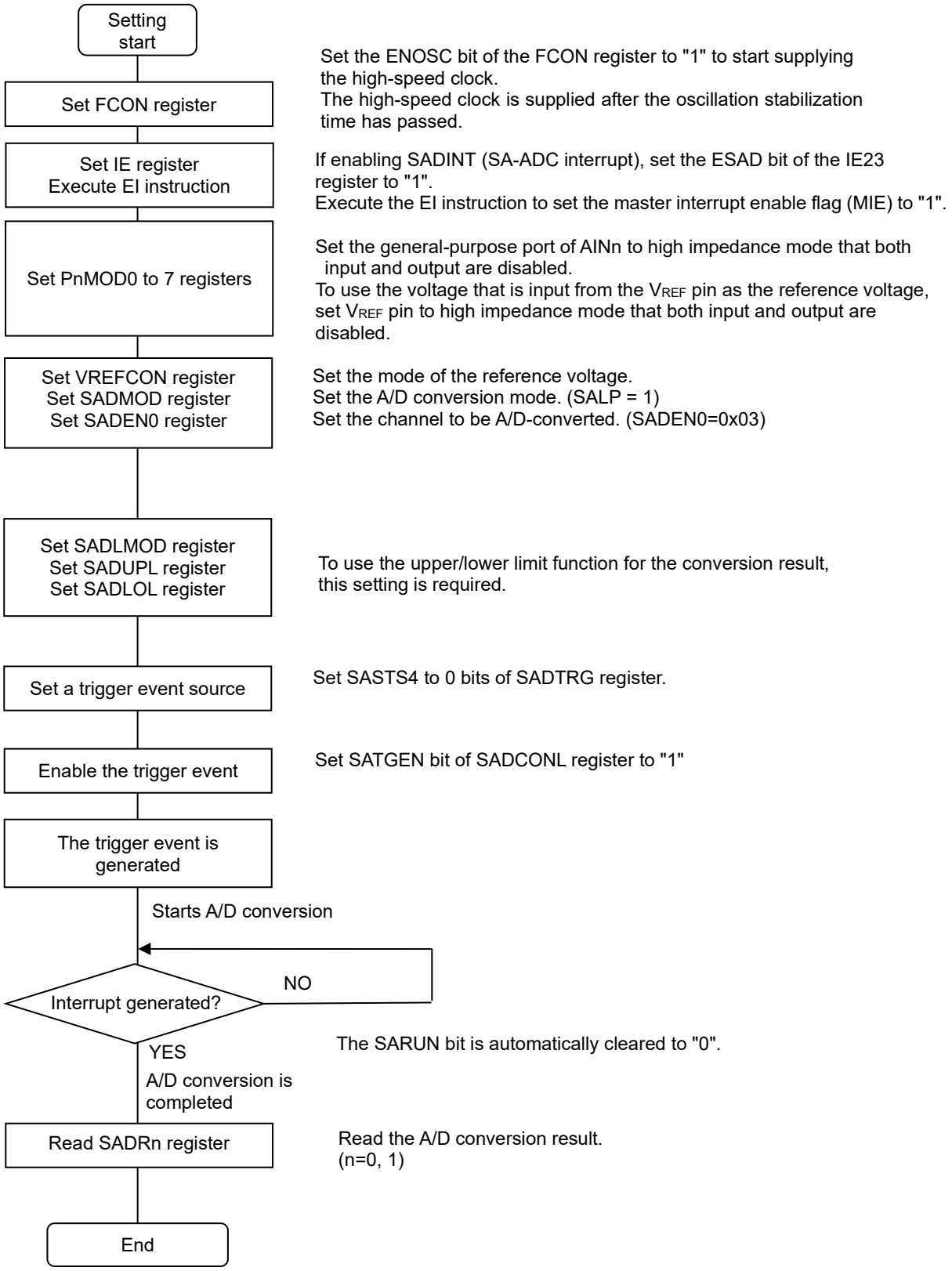
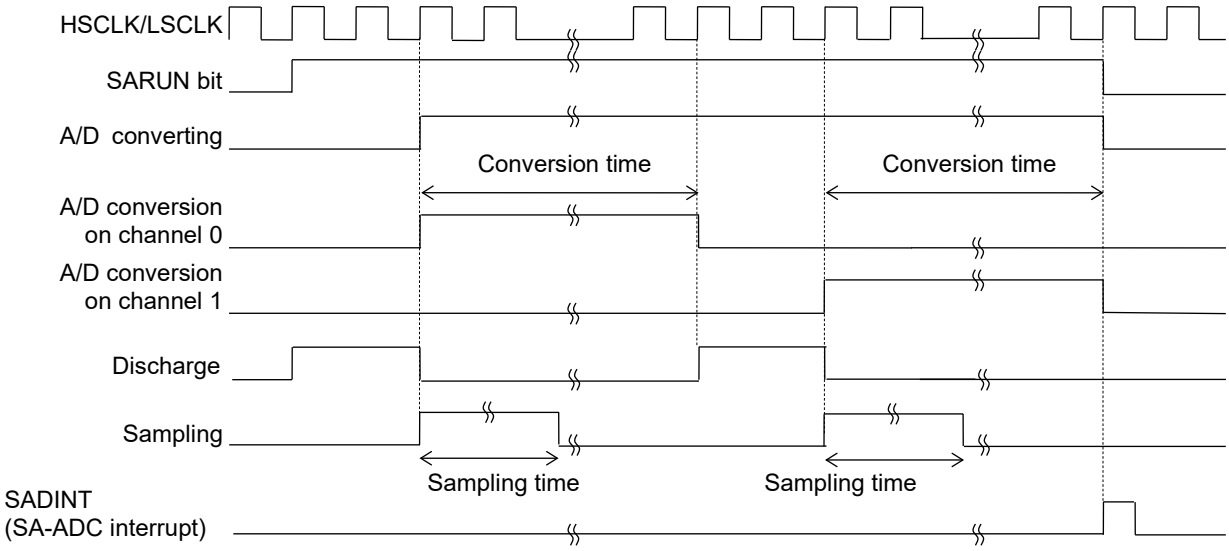


Figure 23-4 Example of A/D Conversion Setting (Start converting by a trigger event)

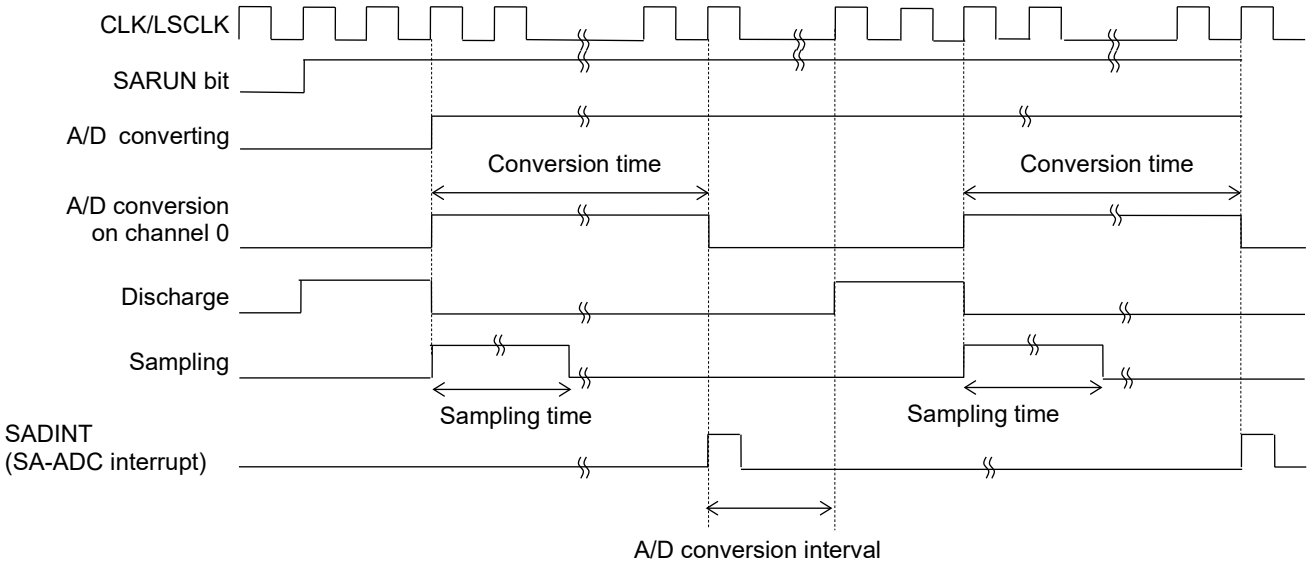
Figure 23-5 shows operation waveforms when one-time A/D conversion is performed using channel 1 and 0.



Presence/absence of discharge can be chosen by SAINIT bit of SADMOD register.
It takes two clocks of SAD_CLK for discharging.
See Section 23.3.3 “A/D Conversion Time Setting” for details of the sampling time and conversion time.

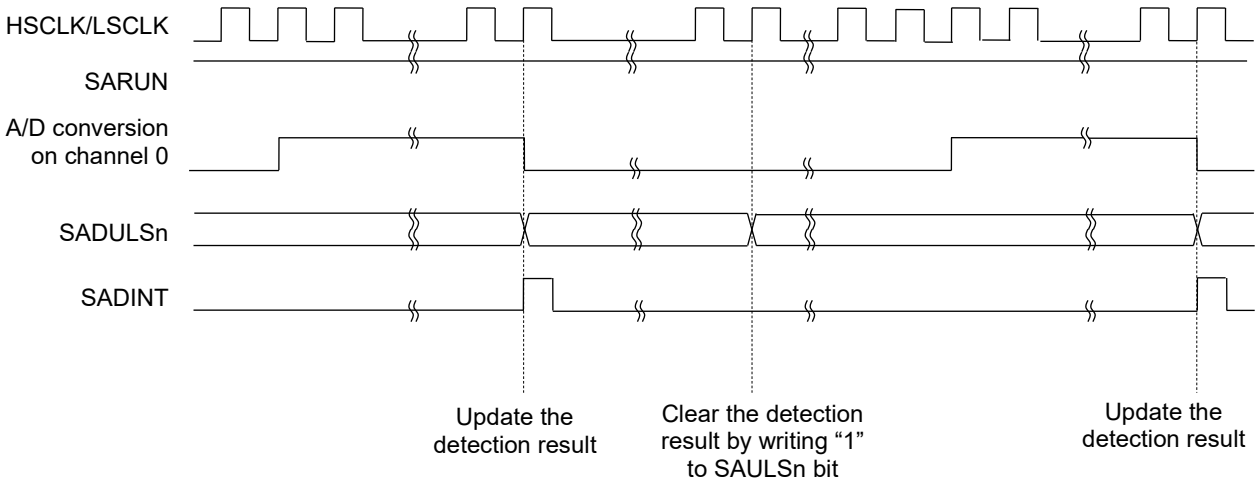
Figure 23-5 Operation Waveforms of A/D Conversion (One-time Conversion)

Figure 23-6 shows the operation waveforms when the continuous A/D conversion is performed using channel 0.



Presence/absence of discharge can be chosen by SAINIT bit of SADMOD register.
It takes two clocks of SAD_CLK for discharging.
See Section 23.3.3 “A/D Conversion Time Setting” for details of the sampling time and conversion time.

Figure 23-6 Operation Waveforms of A/D Conversion (Continuous Conversion)



The figure shows the waveforms when the condition set in SALMD1 to 0 bits satisfied.
The clear to the SAULSn bit takes max. one clock of SAD_CLK after writing "1" to the SAULSn bit .

Figure 23-7 Operation Waveforms of SADULSn register when using the upper/lower limit detection function

23.3.2 How to test the Successive Approximation Type A/D Converter

Follow this procedure to check if the successive approximation type A/D converter works correctly. (n=0 to 15)

(1) Enable the internal reference voltage and choose V_{DD} for the reference voltage (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(approx.1.55V) 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).

23.3.3 A/D Conversion Time Setting

For the A/D conversion time and sampling time, the conversion time and configurable range vary depending on the following settings:

- Reference voltage set using the reference voltage control register (VREFCON)
- A/D conversion operation clock (SAD_CLK) set using the SA-ADC mode register (SADMOD)

Table 23-3 and 23-4 show SADMOD register setting vs the A/D conversion time. Table 23-5 and 23-6 show SADMOD register setting (SASHT3 to SASHT0 bits, SACK2 to SACK0 bits) vs the sampling time.
See the following description for each table vs the reference voltage.

Table 23-3 A/D Conversion time when using V_{DD} or V_{REF} pin as reference voltage

SADMOD				Conversion clock count	Conversion time*1					
					SAD_CLK					
SASHT3 to SASHT0					32kHz	0.5MHz	1MHz	2MHz	4MHz	8MHz
0	0	0	0	14	427 µs	28 µs	Prohibited	Prohibited	Prohibited	Prohibited
0	0	0	1	15	Prohibited	30 µs	15 µs			
0	0	1	0	16		32 µs	16 µs	8 µs		
0	0	1	1	17		34 µs	17 µs	8.5 µs	4.25 µs	
0	1	0	0	18		36 µs	18 µs	9 µs	4.5 µs	2.25 µs
0	1	0	1	19		38 µs	19 µs	9.5 µs	4.75 µs	2.375 µs
0	1	1	0	20		40 µs	20 µs	10 µs	5 µs	2.5 µs
0	1	1	1	21		42 µs	21 µs	10.5 µs	5.25 µs	2.625 µs
1	0	0	0	29		58 µs	29 µs	14.5 µs	7.25 µs	3.625 µs
1	0	0	1	45		90 µs	45 µs	22.5 µs	11.25 µs	5.625 µs
1	0	1	0	61		122 µs	61 µs	30.5 µs	15.25 µs	7.625 µs
1	0	1	1	77		154 µs	77 µs	38.5 µs	19.25 µs	9.625 µs
1	1	0	0	93		186 µs	93 µs	46.5 µs	23.25 µs	11.625 µs
1	1	0	1	109		218 µs	109 µs	54.5 µs	27.25 µs	13.625 µs
1	1	1	0	125		250 µs	125 µs	62.5 µs	31.25 µs	15.625 µs
1	1	1	1	141		282 µs	141 µs	70.5 µs	35.25 µs	17.625 µs

*1: The A/D conversion time does not include discharging time (two clocks of the SAD_CLK) and the clock frequency error.

Table 23-4 A/D conversion time when using the internal reference voltage as reference voltage

SADMOD				Conversion clock count	Conversion time*1					
					SAD_CLK					
SASHT3 to SASHT0					32kHz	0.5MHz	1MHz	2MHz	4MHz	8MHz
0	0	0	0	14	427 µs	Prohibited	Prohibited	Prohibited	Prohibited	Prohibited
0	0	0	1	15	Prohibited					
0	0	1	0	16						
0	0	1	1	17						
0	1	0	0	18						
0	1	0	1	19						
0	1	0	0	20						
0	1	1	1	21						
1	0	0	0	29		58 µs				
1	0	0	1	45		90 µs	45 µs			
1	0	1	0	61		122 µs	61 µs			
1	0	1	1	77		154 µs	77 µs	38.5 µs		
1	1	0	0	93		186 µs	93 µs	46.5 µs		
1	1	0	1	109		218 µs	109 µs	54.5 µs		
1	1	1	0	125		250 µs	125 µs	62.5 µs		
1	1	1	1	141		282 µs	141 µs	70.5 µs	35.25 µs	

*1: The A/D conversion time does not include discharging time (two clocks of the SAD_CLK) and the clock frequency error.

Table 23-5 Sampling time when using V_{DD} or V_{REF} pin as reference voltage

SADMOD				Sample/ hold clock count	Sampling time					
					SAD_CLK					
SASHT3 to SASHT0					32kHz	0.5MHz	1MHz	2MHz	4MHz	8MHz
0	0	0	0	1	30 µs	2 µs	Prohibited	Prohibited	Prohibited	Prohibited
0	0	0	1	2	Prohibited	4 µs	2 µs			
0	0	1	0	3		6 µs	3 µs	1.5 µs		
0	0	1	1	4		8 µs	4 µs	2 µs	1 µs	
0	1	0	0	5		10 µs	5 µs	2.5 µs	1.25 µs	0.625 µs
0	1	0	1	6		12 µs	6 µs	3 µs	1.5 µs	0.75 µs
0	1	1	0	7		14 µs	7 µs	3.5 µs	1.75 µs	0.875 µs
0	1	1	1	8		16 µs	8 µs	4 µs	2 µs	1 µs
1	0	0	0	16		32 µs	16 µs	8 µs	4 µs	2 µs
1	0	0	1	32		64 µs	32 µs	16 µs	8 µs	4 µs
1	0	1	0	48		96 µs	48 µs	24 µs	12 µs	6 µs
1	0	1	1	64		128 µs	64 µs	32 µs	16 µs	8 µs
1	1	0	0	80		160 µs	80 µs	40 µs	20 µs	10 µs
1	1	0	1	96		192 µs	96 µs	48 µs	24 µs	12 µs
1	1	1	0	112		224 µs	112 µs	56 µs	28 µs	14 µs
1	1	1	1	128		256 µs	128 µs	64 µs	32 µs	16 µs

Table 23-6 Sampling time when using the internal reference voltage as reference voltage

SADMOD				Sample/ hold clock count	Sampling time					
					SAD_CLK					
SASHT3 to SASHT0					32kHz	0.5MHz	1MHz	2MHz	4MHz	8MHz
0	0	0	0	1	30 µs	Prohibited	Prohibited	Prohibited	Prohibited	Prohibited
0	0	0	1	2	Prohibited					
0	0	1	0	3						
0	0	1	1	4						
0	1	0	0	5						
0	1	0	1	6						
0	1	1	0	7						
0	1	1	1	8						
1	0	0	0	16		32 µs				
1	0	0	1	32		64 µs	32 µs			
1	0	1	0	48		96 µs	48 µs			
1	0	1	1	64		128 µs	64 µs	32 µs		
1	1	0	0	80		160 µs	80 µs	40 µs		
1	1	0	1	96		192 µs	96 µs	48 µs		
1	1	1	0	112		224 µs	112 µs	56 µs		
1	1	1	1	128		256 µs	128 µs	64 µs	32 µs	

23.4 Notes on SA-ADC

23.4.1 Sampling Time Setting

Sampling time of the SA-ADC should satisfy the following formula:

$$\text{Sampling time} > 8(C_{SAMPLE} + C_{PARA})(R_1 + R_2)$$

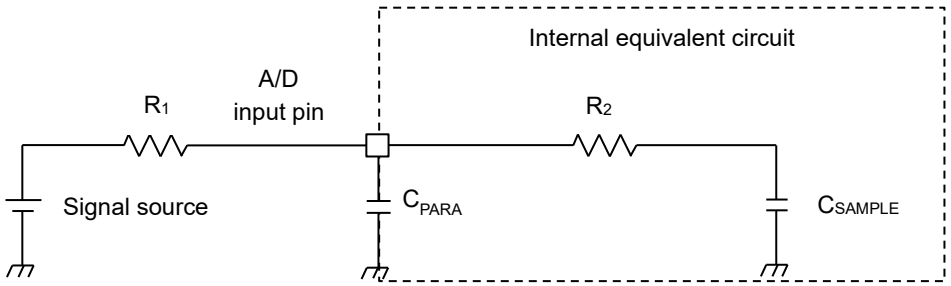
To calculate sampling time more precisely, use the following formula:

$$\text{Sampling time} = \left\{log_e(2^n) + log_e\left(\frac{C_{SAMPLE}}{C_{SAMPLE} + C_{PARA}}\right)\right\}(C_{SAMPLE} + C_{PARA})(R_1 + R_2)$$

C_{PARA} varies depending on board-layout and connected parts. Please check the accuracy of SA-ADC with the actual board.

R_1 : Input impedance of external resistor
R_2 : Internal resistor value which is the sum of the internal resistor and the ON register of the switch
C_{SAMPLE} : Sample hold capacitor
C_{PARA} : Parasitic capacitance of the A/D input line.(Measure the capacitance between the A/D input line and V_{SS}.)
n : Resolution of SA-ADC

The following diagram shows the equivalent circuit in this case:



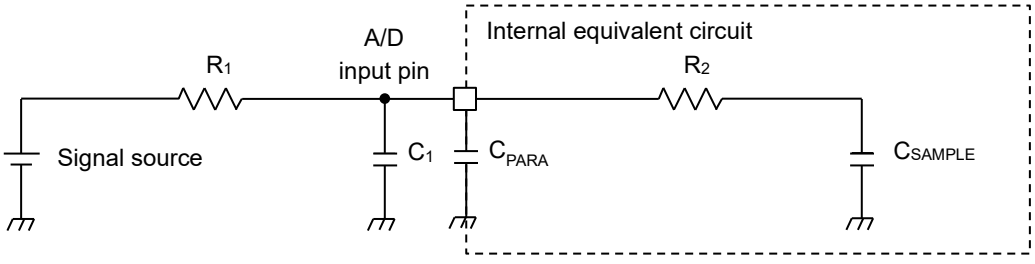
V_{REF}	R_2[kΩ]	C_{SAMPLE}[pF]
1.8 V≤V_{REF}≤2.2V	500	5
2.2 V≤V_{REF}≤2.7V	100	5
2.7 V≤V_{REF}≤4.5V	8	5
4.5 V≤V_{REF}≤5.5V	7	5

The values above are reference values.
If the sampling time above is unsatisfied, connect the external capacitor near by A/D input pin to satisfy the following formula.

$$(C_1 + C_{PARA}) > 2^n C_{SAMPLE}$$
$$\text{Sampling time} > 8C_{SAMPLE}R_2$$

C_1 : External capacitor

The equivalent circuit when the external capacitor C_1 is connected is as follows:



Note that the voltage at the A/D input pin transitionally changes due to the external capacitor C_1 and the external resistor R_1. Therefore, when sampling data, wait until the voltage is stabilized. If the stabilization timing is unknown, perform A/D conversion once, then wait for time constant τ ($=R_1C_1$) to 4τ or so and perform A/D conversion again. Confirm that the difference between values is small, and then sample data.

23.4.2 Noise Suppression

In order to prevent deterioration in accuracy of A/D conversion, operate the A/D converter in the environment with little noise.
The following processes are recommended for noise reduction:
- Perform A/D conversion in the HALT mode.
- Do not have clock input/output to and from a pin located in the vicinity of the pin in which A/D conversion is in progress.
- Do not have clock input/output to and from the pin in which A/D conversion is in progress and other A/D conversion pins.

In addition, the capacitor for noise suppression should be connected between V_{REF} and V_{SS}, as well as between V_{DD} and V_{SS}. When connecting, place the capacitor in the immediate vicinity of LSI using short wiring.

Chapter 24 Regulator

24. Regulator

24.1 General Description

ML62Q1000 series incorporates the regulator.
Figure 24-1 shows the general scheme of the regulator.
The regulator generates a constant internal logic voltage (V_{DDL}) independent of the variation of V_{DD} (1.6 V to 5.5 V) using an amplifier for the low power consumption. The V_{DDL} generated by the regulator is supplied to peripheral circuits such as the internal logic circuit, flash memory, RAM, and oscillation circuit.
In order to stabilize the V_{DDL}, connect the V_{DDL} pin to V_{SS} via a capacitor (1 µF).

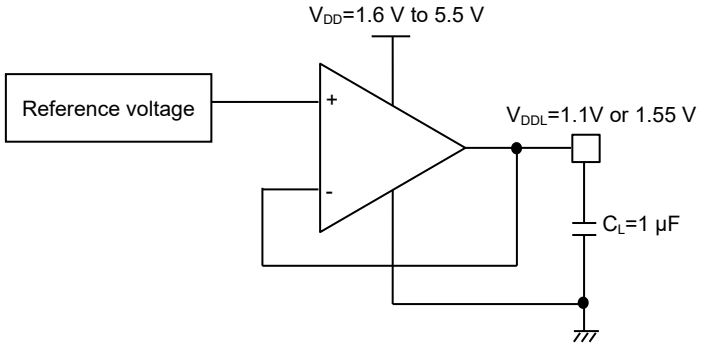


Figure 24-1 General Scheme of Regulator

24.1.1 Features

Mode	VDDL voltage
STOP mode	1.55 V
HALT mode	1.55 V
HALT-H mode	1.55 V
Program run mode	1.55 V
STOP-D mode (content of RAM and SFR can be retained)	1.1 V

24.1.2 Configuration

Figure 24-2 shows the configuration of the internal power supply.

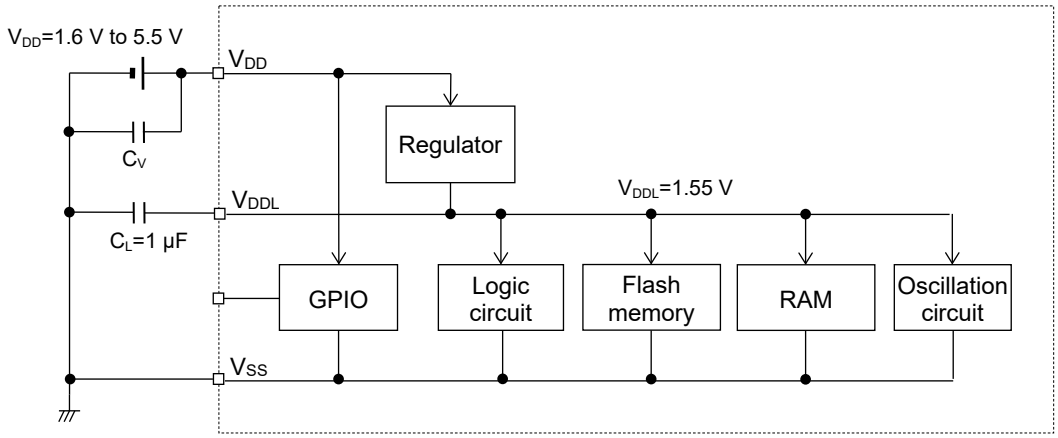


Figure 24-2 Internal Power Supply Configuration

24.1.3 List of Pins

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

Pin name	I/O	Function
V_{DDL}	-	Positive power supply for the internal logic circuits
V_{REFO}	-	Reference voltage output

[Note]

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

24.2 Description of Operation

After power-on, V_{DDL} becomes approximately 1.55 V.
In the STOP-D mode, V_{DDL} is lowered to approximately 1.1 V to reduce the standby current. In the STOP-D mode, the content of RAM and SFR is retained.
In the HALT, HALT-H, or STOP mode, V_{DDL} is approximately 1.55 V.
Figure 24-3 shows the operation waveforms of the regulator.
See Chapter 4 "Power Management" and Chapter 6 "Clock Generation Circuit" for the STOP-D mode and stabilization time.

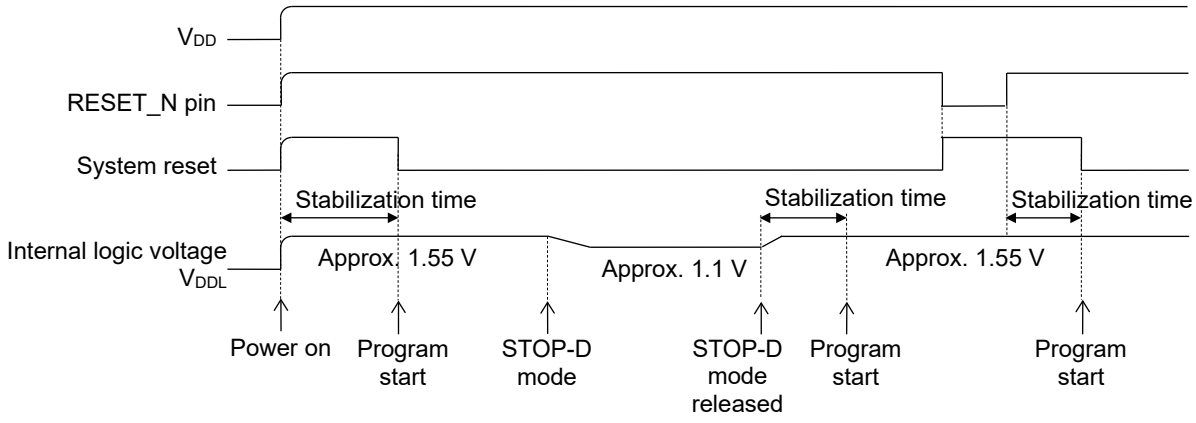


Figure 24-3 Regulator Operation Waveforms

24.2.1 Reference Voltage Output

See Section 23.2.14 "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.

- The ways of programming the flash memory

Programming method	Tool/Register/Communication	Reference Chapter
Programming by the on-chip debug function	On-chip debug emulator or other flash programmers	Chapter 28 "On chip Debug function"
Self-Programming by using the special function register(SFR)	Special Function Registers(SFRs) for programming the flash memory	Section 25.3 "Self-programming"
Programming by the ISP (In-System Programming) function	UART communication with an external device 3rd Party Flash programmers (*1)	Section 25.4 "ISP function"

*1: Contact the 3rd party makers for details about the Flash programmer.

The specification of the program memory space and data flash are is dependent of the product.

- Program memory space and Data flash area Overview (Size and Address)

Part name		Program memory space		Data flash area	
		Size	Address	Size	Address
ML62Q1300 group	ML62Q1323/1333	16K byte	0x0:0000 to 0x0:3FFF	2K byte (128 byte x 16 sector)	0x1F:0000 to 0x1F:07FF
	ML62Q1324/1334	24K byte	0x0:0000 to 0x0:5FFF		
	ML62Q1325/1335/1345/1365	32K byte	0x0:0000 to 0x0:7FFF		
	ML62Q1346/1366	48K byte	0x0:0000 to 0x0:BFFF		
	ML62Q1347/1367	64K byte	0x0:0000 to 0x0:FFFF		
ML62Q1500/ ML62Q1800 group	ML62Q1530/1540/1550	32K byte	0x0:0000 to 0x0:7FFF	4K byte (128 byte x 32 sector)	0x1F:0000 to 0x1F:0FFF
	ML62Q1531/1541/1551	48K byte	0x0:0000 to 0x0:BFFF		
	ML62Q1532/1542/1552	64K byte	0x0:0000 to 0x0:FFFF		
	ML62Q1533/1543/1553/1563/1573	96K byte	0x0:0000 to 0x1:7FFF		
	ML62Q1534/1544/1554/1564/1574	128K byte	0x0:0000 to 0x1:FFFF		
	ML62Q1555/1565/1575	160K byte	0x0:0000 to 0x2:0:7FFF		
	ML62Q1556/1566/1576	192K byte	0x0:0000 to 0x2:FFFF		
	ML62Q1557/1567/1577	256K byte	0x0:0000 to 0x3:FFFF		
	ML62Q1858/1868/1878	384K byte	0x0:0000 to 0x5:FFFF	8K byte (128 byte x 64 sector)	0x1F:0000 to 0x1F:1FFF
	ML62Q1859/1869/1879	512K byte	0x0:0000 to 0x7:FFFF		
ML62Q1700 group	ML62Q1700/1710/1720	32K byte	0x0:0000 to 0x0:7FFF	4K byte (128 byte x 32 sector)	0x1F:0000 to 0x1F:0FFF
	ML62Q1701/1711/1721	48K byte	0x0:0000 to 0x0:BFFF		
	ML62Q1702/1712/1722	64K byte	0x0:0000 to 0x0:FFFF		
	ML62Q1703/1713/1723/1733/1743	96K byte	0x0:0000 to 0x1:7FFF		
	ML62Q1704/1714/1724/1734/1744	128K byte	0x0:0000 to 0x1:FFFF		
	ML62Q1725/1735/1745	160K byte	0x0:0000 to 0x2:7FFF		
	ML62Q1726/1736/1746	192K byte	0x0:0000 to 0x2:FFFF		
	ML62Q1727/1737/1747	256K byte	0x0:0000 to 0x3:FFFF		
	ML62Q1728/1738/1748	384K byte	0x0:0000 to 0x5:FFFF	8K byte (128 byte x 64 sector)	0x1F:0000 to 0x1F:1FFF
	ML62Q1729/1739/1749	512K byte	0x0:0000 to 0x7: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	all area
	Sector erase	1K byte	128 byte
	Programming	4 byte (32bit)	1 byte (8bit)
Erasing and programming time	Chip erase(ISP only)	Max. 50ms	Max. 50ms
	Block erase		
	Sector erase		
	Programming	Max. 80μs	Max. 40μs
Programming cycle		100 times	10,000 times
Erasing and programming temperature		0°C to +40°C	-40°C to +85°C
Background operation(BGO) function		-	Yes
Erasing and programming completion Interrupt		No	Yes

25.1.1 List of Pins

Programming by the ISP function uses the following pins.

Signal name	I/O	Function
RESET_N	I	Input signal for entering the ISP mode
TEST0	I/O	Input signal for entering the ISP mode and data input/output data in the single wired UART communication

25.2 Description of Registers

25.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF090	Flash address register	FLASHAL	FLASHA	R/W	8/16	0xFF
0xF091		FLASHAH		R/W	8	0xFF
0xF092	Flash data register 0	FLASHD0L	FLASHD0	R/W	8/16	0xFF
0xF093		FLASHD0H		R/W	8	0xFF
0xF094	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	-	-	-	-	-

25.2.2 Flash Address Register (FLASHA)

FLASHA is a special function register (SFR) used to set the erasing and programming address.

Address: 0xF090(FLASHAL/FLASHA), 0xF091(FLASHAH)
Access: R/W
Access size: 8/16bit
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FLASHA															
Byte	FLASHAH								FLASHAL							
Bit	FA15	FA14	FA13	FA12	FA11	FA10	FA9	FA8	FA7	FA6	FA5	FA4	FA3	FA2	FA1	FA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 0	FA15 to FA0	These bits are used to set the erasing or programming address.

[Note]

- **Note that programming for the program memory space is performed by the unit of 4 bytes. Because of this, the setting values in the FA1 bit and FA0 bit are ignored.**

25.2.3 Flash Segment Register (FLASHSEG)

FLASHSEG is a special function register (SFR) used to set the segment for erasing and programming the flash memory.

Address: 0xF09A(FLASHSEG)
Access: R/W
Access size: 8bit
Initial value: 0x10

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FLASHSEG							
Bit	-	-	-	-	-	-	-	-	-	-	-	FSEG4	FSEG3	FSEG2	FSEG1	FSEG0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 5	-	Reserved bits
4 to 0	FSEG4 to FSEG0	These bits are used to set the flash memory segment address.

Table 25-1 shows the address setting value for block erase and Table 25-2 shows the address setting value for sector erase.

Table 25-1 Address Setting Values for Block Erase

Segment	Block	Address	Size	FLASHSEG register	FLASHA register
Segment 0	Block 0	0x0000 to 0x3FFF	16K byte	0x00	0x0000
	Block 1	0x4000 to 0x7FFF	16K byte		0x4000
	Block 2	0x8000 to 0xBFFF	16K byte		0x8000
	Block 3	0xC000 to 0xFFFF	16K byte		0xC000
Segment 1	Block 4	0x0000 to 0x3FFF	16K byte	0x01	0x0000
	Block 5	0x4000 to 0x7FFF	16K byte		0x4000
	Block 6	0x8000 to 0xBFFF	16K byte		0x8000
	Block 7	0xC000 to 0xFFFF	16K byte		0xC000
Segment 2	Block 8	0x0000 to 0x3FFF	16K byte	0x02	0x0000
	Block 9	0x4000 to 0x7FFF	16K byte		0x4000
	Block 10	0x8000 to 0xBFFF	16K byte		0x8000
	Block 11	0xC000 to 0xFFFF	16K byte		0xC000
Segment 3	Block 12	0x0000 to 0x3FFF	16K byte	0x03	0x0000
	Block 13	0x4000 to 0x7FFF	16K byte		0x4000
	Block 14	0x8000 to 0xBFFF	16K byte		0x8000
	Block 15	0xC000 to 0xFFFF	16K byte		0xC000
Segment 4	Block 16	0x0000 to 0x3FFF	16K byte	0x04	0x0000
	Block 17	0x4000 to 0x7FFF	16K byte		0x4000
	Block 18	0x8000 to 0xBFFF	16K byte		0x8000
	Block 19	0xC000 to 0xFFFF	16K byte		0xC000
Segment 5	Block 20	0x0000 to 0x3FFF	16K byte	0x05	0x0000
	Block 21	0x4000 to 0x7FFF	16K byte		0x4000
	Block 22	0x8000 to 0xBFFF	16K byte		0x8000
	Block 23	0xC000 to 0xFFFF	16K byte		0xC000
Segment 6	Block 24	0x0000 to 0x3FFF	16K byte	0x06	0x0000
	Block 25	0x4000 to 0x7FFF	16K byte		0x4000
	Block 26	0x8000 to 0xBFFF	16K byte		0x8000
	Block 27	0xC000 to 0xFFFF	16K byte		0xC000
Segment 7	Block 28	0x0000 to 0x3FFF	16K byte	0x07	0x0000
	Block 29	0x4000 to 0x7FFF	16K byte		0x4000
	Block 30	0x8000 to 0xBFFF	16K byte		0x8000
	Block 31	0xC000 to 0xFFFF	16K byte		0xC000
Segment 31	Block 0	0x0000 to 0x07FF 0x0000 to 0x0FFF 0x0000 to 0x1FFF	2K byte 4K byte 8K byte	0x1F	0x0000

Table 25-2 Address Setting Values for Sector Erase(1/2)

Segment	Block	Address	Size	FLASHSEG register	FLASHA register
Segment 0	Sector 0	0x0000 to 0x03FF	1K byte	0x00	0x0000
	Sector 1	0x0400 to 0x07FF	1K byte		0x0400
	:	:	:		:
	Sector 62	0xF800 to 0xFBFF	1K byte		0xF800
	Sector 63	0xFC00 to 0xFFFF	1K byte		0xFC00
Segment 1	Sector 64	0x0000 to 0x03FF	1K byte	0x01	0x0000
	Sector 65	0x0400 to 0x07FF	1K byte		0x0400
	:	:	:		:
	Sector 126	0xF800 to 0xFBFF	1K byte		0xF800
	Sector 127	0xFC00 to 0xFFFF	1K byte		0xFC00
Segment 2	Sector 128	0x0000 to 0x03FF	1K byte	0x02	0x0000
	Sector 129	0x0400 to 0x07FF	1K byte		0x0400
	:	:	:		:
	Sector 190	0xF800 to 0xFBFF	1K byte		0xF800
	Sector 191	0xFC00 to 0xFFFF	1K byte		0xFC00
Segment 3	Sector 192	0x0000 to 0x03FF	1K byte	0x03	0x0000
	Sector 193	0x0400 to 0x07FF	1K byte		0x0400
	:	:	:		:
	Sector 254	0xF800 to 0xFBFF	1K byte		0xF800
	Sector 255	0xFC00 to 0xFFFF	1K byte		0xFC00
Segment 4	Sector 256	0x0000 to 0x03FF	1K byte	0x04	0x0000
	Sector 257	0x0400 to 0x07FF	1K byte		0x0400
	:	:	:		:
	Sector 318	0xF800 to 0xFBFF	1K byte		0xF800
	Sector 319	0xFC00 to 0xFFFF	1K byte		0xFC00
Segment 5	Sector 320	0x0000 to 0x03FF	1K byte	0x05	0x0000
	Sector 321	0x0400 to 0x07FF	1K byte		0x0400
	:	:	:		:
	Sector 382	0xF800 to 0xFBFF	1K byte		0xF800
	Sector 383	0xFC00 to 0xFFFF	1K byte		0xFC00
Segment 6	Sector 384	0x0000 to 0x03FF	1K byte	0x06	0x0000
	Sector 385	0x0400 to 0x07FF	1K byte		0x0400
	:	:	:		:
	Sector 446	0xF800 to 0xFBFF	1K byte		0xF800
	Sector 447	0xFC00 to 0xFFFF	1K byte		0xFC00
Segment 7	Sector 448	0x0000 to 0x03FF	1K byte	0x07	0x0000
	Sector 449	0x0400 to 0x07FF	1K byte		0x0400
	:	:	:		:
	Sector 510	0xF800 to 0xFBFF	1K byte		0xF800
	Sector 511	0xFC00 to 0xFFFF	1K byte		0xFC00

Table 25-2 Address Setting Values for Sector Erase(2/2)

Segment	Block	Address	Size	FLASHSEG register	FLASHA register
Segment 31	Sector 0	0x0000 to 0x007F	128 byte	0x1F	0x0000
	Sector 1	0x0080 to 0x00FF	128 byte		0x0080
	Sector 2	0x0100 to 0x017F	128 byte		0x0100
	Sector 3	0x0180 to 0x01FF	128 byte		0x0180
	:	:	:		:
	Sector 12	0x0600 to 0x067F	128 byte		0x0600
	Sector 13	0x0680 to 0x06FF	128 byte		0x0680
	Sector 14	0x0700 to 0x077F	128 byte		0x0700
	Sector 15	0x0780 to 0x07FF	128 byte		0x0780
	:	:	:		:
	Sector 28	0x0E00 to 0x0E7F	128 byte		0x0E00
	Sector 29	0x0E80 to 0x0EFF	128 byte		0x0E80
	Sector 30	0x0F00 to 0x0F7F	128 byte		0x0F00
	Sector 31	0x0F80 to 0x0FFF	128 byte		0x0F80
	:	:	:		:
	Sector 60	0x1E00 to 0x1E7F	128 byte		0x1E00
	Sector 61	0x1E80 to 0x1EFF	128 byte		0x1E80
	Sector 62	0x1F00 to 0x1F7F	128 byte		0x1F00
	Sector 63	0x1F80 to 0x1FFF	128 byte		0x1F80

25.2.4 Flash Data Register 0 (FLASHD0)

FLASHD0 is a special function register (SFR) used to set programming data.

Address: 0xF092(FLASHD0L/FLASHD0), 0xF093(FLASHD0H)
Access: R/W
Access size: 8/16bit
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FLASHD0															
Byte	FLASHD0H								FLASHD0L							
Bit	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 8	FD15 to FD8	These bits are used to set the second byte data.
7 to 0	FD7 to FD0	These bits are used to set the first byte data.

There are some differences for programming the program memory space and the data flash area.

Programming target	Register	Description	Note
Program memory space	Four bytes specified in FLASHD0 register(FLASHD0H, FLASHD0L) and FLASHD1 register(FLASHD1H, FLASHD1L)	The programming starts by writing data into FLASHD1H register.	Write data into FLASHD0 register at first and FLASHD1 register the second.
Data flash area	FLASHD0L register only (one byte) in FLASHD0 register.	The programming starts by writing data into FLASHD0L register.	Data written into FLASHD0H register and FLASHD1 register are invalid.

[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 is unguaranteed.**
- **Do not read or program unused areas to prevent the CPU works incorrectly.**

25.2.5 Flash Data Register 1 (FLASHD1)

FLASHD1 is a special function register (SFR) used to set programming data.

Address: 0xF094(FLASHD1L/FLASHD1), 0xF095(FLASHD1H)
Access: R/W
Access size: 8/16bit
Initial value: 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FLASHD1															
Byte	FLASHD1H								FLASHD1L							
Bit	FD31	FD30	FD29	FD28	FD27	FD26	FD25	FD24	FD23	FD22	FD21	FD20	FD19	FD18	FD17	FD16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 8	FD31 to FD24	These bits are used to set the 4th byte data.
7 to 0	FD23 to FD16	These bits are used to set the 3rd byte data.

There are some differences for programming the program memory space and the data flash area.

Programming target	Register	Description	Note
Program memory space	Four bytes specified in FLASHD0 register(FLASHD0H, FLASHD0L) and FLASHD1 register(FLASHD1H, FLASHD1L)	The programming starts by writing data into FLASHD1H register.	Write data into FLASHD0 register at first and FLASHD1 register the second, in LSB first.
Data flash area	FLASHD0L register only (one byte) in FLASHD0 register.	The programming starts by writing data into FLASHD0L register.	Data written into FLASHD0H register and FLASHD1 register are invalid.

[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 is unguaranteed.**
- **Do not read or program unused areas to prevent the CPU works incorrectly.**

25.2.6 Flash Control Register (FLASHCON)

FLASHCON is a write-only special function register (SFR) used to control the block erase and sector erase for the flash memory. This register always returns 0x00 for reading.

Address: 0xF096(FLASHCON)
Access: W
Access size: 8bit
Initial value: 0x00

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

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1, 0	FSERS, FERS	These bits are used to start the sector erase or block erase. Setting the FSERS bit to "1" starts erasing the sector and setting the FERS bit to "1" starts erasing the block specified by the FLASHSEG and FLASHAH register. 00: No function (Initial value) 01: Start block erasing 10: Start sector erasing 11: Do not use (No function)

25.2.7 Flash Acceptor (FLASHACP)

FLASHACP is a write-only special function register (SFR) used to accept for erasing/programming the flash memory.

Address: 0xF098(FLASHACP)
Access: W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FLASHACP							
Bit	-	-	-	-	-	-	-	-	fac7	fac6	fac5	fac4	fac3	fac2	fac1	fac0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	fac7 to fac0	These bits are used to accept for erasing/programming the flash memory in order to prevent an unintended erasing/programming operation. When "0xFA" and "0xF5" are written to the FLASHACP in this order, the erasing or programming function is enabled only once. For subsequent erasing or programming, "0xFA" and "0xF5" must be written to FLASHACP each time.

[Note]

- **Even if other instructions are executed between the instruction that writes "0xFA" and "0xF5" to the FLASHACP, the erasing or programming function is still valid.**
- **If data other than "0xF5" is written to the FLASHACP after "0xFA" is written, "0xFA" becomes invalid. In this case, it needs to write "0xFA" again.**

25.2.8 Flash Self Register (FLASHSLF)

FLASHSLF is a special function register (SFR) used to enable erasing and programming the flash memory. When system clock is the low-speed clock, it is disabled to write.

Address: 0xF09C(FLASHSLF)
Access: R/W
Access size: 8bit
Initial value: 0x00

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

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	FSELF	This bit is used to enable erasing and programming the flash memory. 0: Erasing and programming the flash memory is disabled (Initial value) 1: Erasing and programming the flash memory is enabled

25.2.9 Flash Status Register (FLASHSTA)

FLASHSTA is a read-only special function register (SFR) used to check status of the flash memory.

Address: 0xF09E(FLASHSTA)
Access: R
Access size: 8bit
Initial value: 0x00

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

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1	FDPRSTA	This bit is used to indicate whether the data flash area is in the state of programming. 0: Data flash memory is not in the state of programming (Initial value) 1: Data flash memory is in the state of programming
0	FDERSTA	This bit is used to indicate whether the data flash area is in the state of erasing. 0: Data flash memory is not in the state of erasing (Initial value) 1: Data flash memory is in the state of erasing

FLASHSTA is used when erasing or programming the data flash memory.

Erasing/Programming target	Availability to read this register while erasing/programming	Description
Program memory space	Unavailable	Do not use the FLASHSTA register.
Data flash area	Available	Start erasing/programming the flash checking the bit is "0".

The CPU stops running the program codes while erasing or programming the program flash memory, therefore FLASHSTA is not readable in that case.

As the Back Ground Operation(BGO) function allows the CPU continue running the program codes, make a process for the next erasing and programming by checking the FDERSTA bit or FDPRSTA bit to see if the erasing or programming is completed.

[Note]

- **Perform the erasing or programming after checking the FDERSTA bit or FDPRSTA bit are "0". The erasing or programming becomes invalid when either the FDERSTA bit or the FDPRSTA bit is "1".**

25.3 Self-programming

The self-programming is the function to program (erase and program) the program memory space and data flash area using special function registers (SFRs).
Table 25-3 shows the self-programming specifications for each of the program memory space and data flash area.

Table 25-3 Self-programming of Program Memory Space and Data Flash Area

		Program memory space (Segment 0 to 7)	Data flash area (segment 31)
Programming unit	Erase block	16 Kbyte	all area
	Erase sector	1 Kbyte	128 byte
	Program	4 byte	1 byte
CPU operation during block/sector erase or program		Stop program processing (after completion of erase/program, resume program processing)	Continue program processing through the background operation (BGO) function
Confirmation of end of block/sector erase or program		Confirmation not required (as program run is stopped during erase/program)	Confirmation can be made through FLASHSTA register
Target area where block/sector erase has been applied		Every bit becomes "1" (the bit written with "0" by writing becomes "0" from "1")	
Note on data programming		Erase the area to be reprogrammed (data programmed without erasing is unguaranteed)	
Function to prevent unintended erase/program		Flash self-register (FLASHSLF) and flash acceptor (FLASHACP) incorporated (*1)	
Flash memory erase/program		Supported only when system clock is the high-speed clock (*2)	
Note on user program programming		Before programming the user program, prepare a program for self-programming in the program code area which is not erased/reprogrammed	-
Remapping function		User program update, etc. can be performed by simultaneously using remapping function	-

*1: After the programming is enabled by the FLASHSLF register, if "0xFA" and "0xF5" are written to the flash acceptor (FLASHACP), block/sector erase or reprogram is enabled only once.
*2: See Chapter 6 "Clock Generation Circuit" for enabling oscillation of the high-speed oscillation circuit and switching the system clock.

25.3.1 Notes on Debugging Self-programming Code

When debugging the area within the scope of program for self-programming (from setting the flash acceptor to writing the flash data register 0, 1) using U16 development environment (debugger), use the debugger according to the precautions described in Table 25-4.

Table 25-4 Notes on Debugging Self-programming

Limited function	Notes
Breakpoint setting	Do not perform the real time execution with break points set in the scope of program for self-programming (from setting the flash acceptor to setting the flash data register0, 1). Otherwise, the flash memory may not be reprogrammed if break points occur within the scope of program for self-programming.
Step execution	Do not perform the step execution within the scope of program for self-programming. Otherwise, the flash memory may not be reprogrammed if the step execution is performed within the scope of program for self-programming.

25.3.2 Programming Program Memory Space

In the program memory space (flash memory), block erase in units of 16 Kbytes, sector erase in units of 1 Kbyte, and reprogram in units of 4 bytes can be executed.
Figure 25-1 shows the flow diagram for erasing the program memory space.

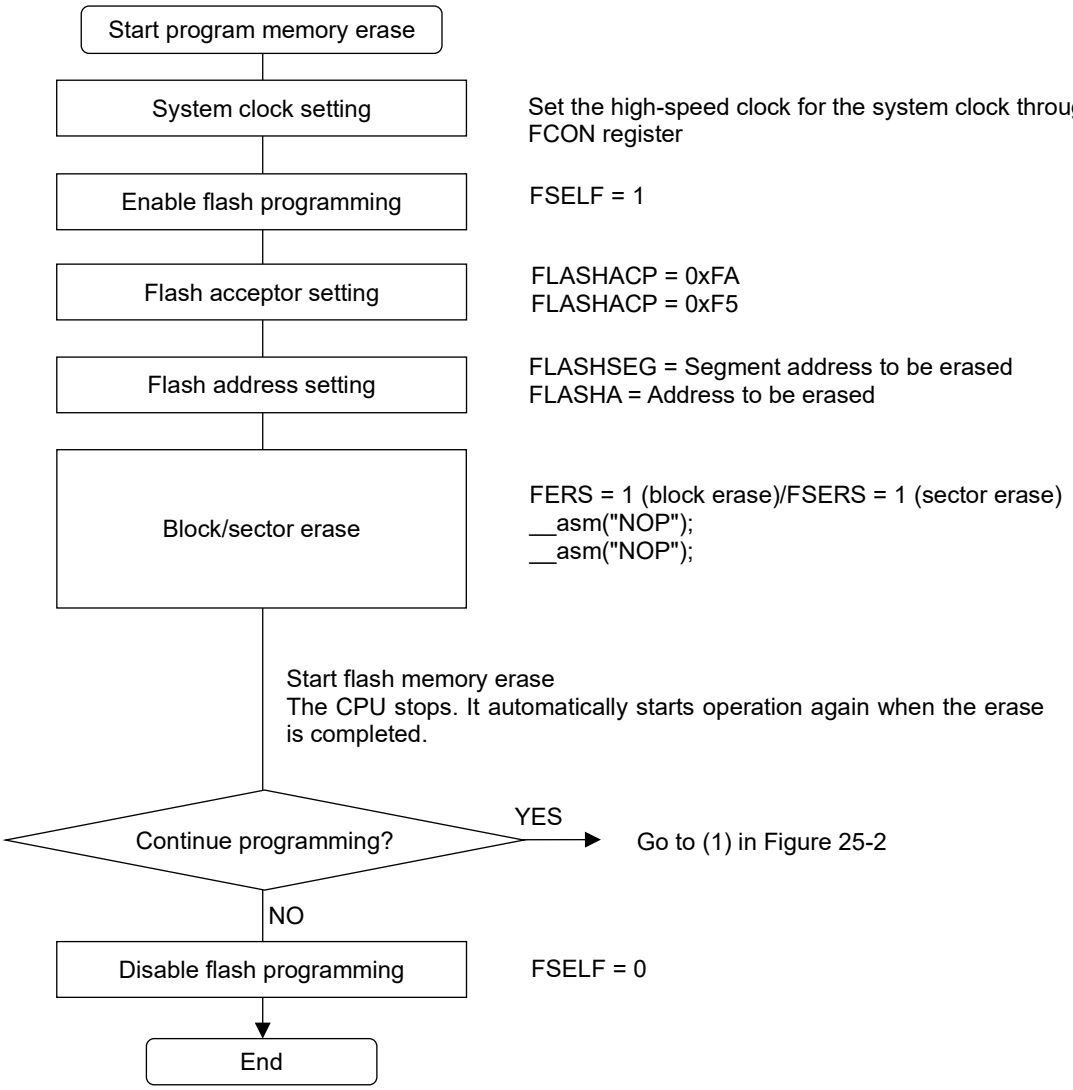


Figure 25-1 Flow Diagram for Erasing Program Memory Space

[Note]

- **Only erase areas irrelevant to program processing. If erasing the area where program processing is in progress, the LSI works incorrectly.**
- **During block/sector erase, the CPU stops the operation for maximum 50 ms whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.**
- **For block/sector erase, place two NOP instructions following the instruction used to set FERS/FSERS bits of the FLASHCON register to "1".**

Figure 25-2 shows the flow diagram for programming the program memory space.

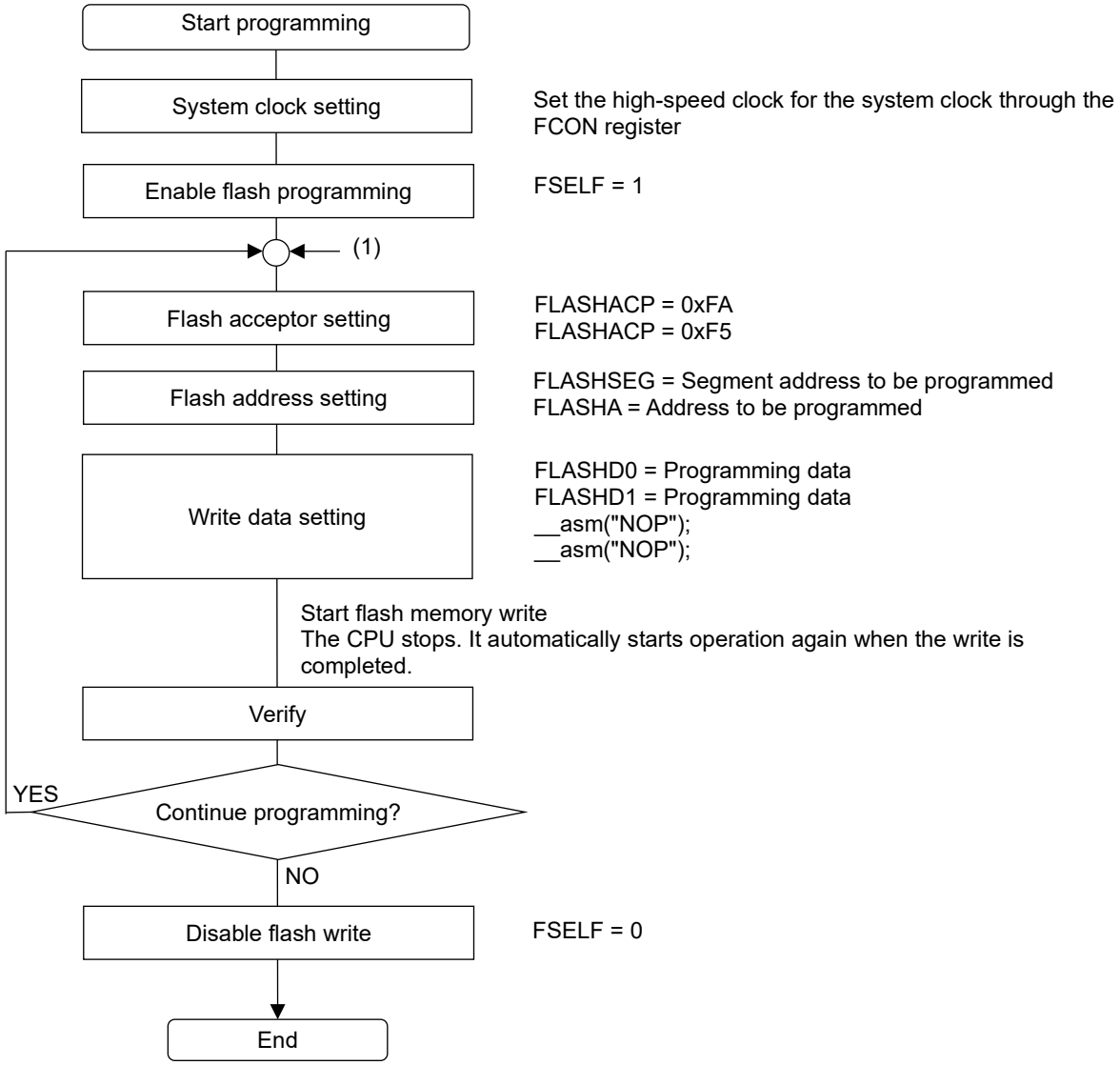


Figure 25-2 Flow Diagram for Programming Program Memory Space

[Note]

- **Only erase areas irrelevant to program processing. If erasing the area where program processing is in progress, the LSI works incorrectly.**
- **During the programming, the CPU stops the operation for maximum 80 µs whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.**
- **For data programming setting, place two NOP instructions following the instruction used to set the programming data in the FLASHD1 register.**

25.3.3 Programming Data Flash Area

In the data flash area (flash memory), block erase in all area, sector erase in units of 128 bytes, and programming in units of 1 byte can be executed. During block/sector erase or program in the data flash area, the CPU continues program processing using the background operation (BGO) function.
Figure 25-3 shows the flow diagram for erasing the data flash area.

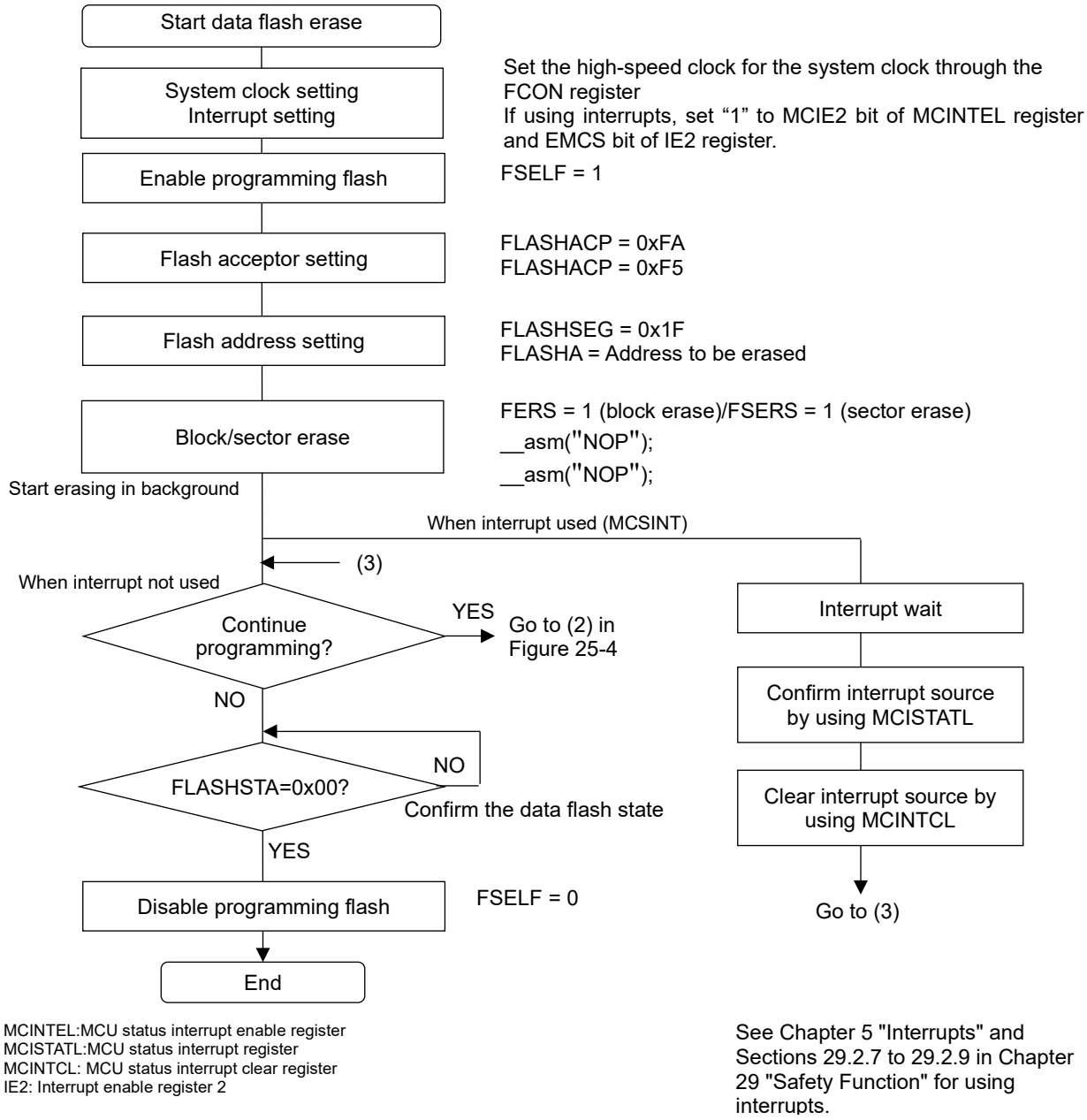


Figure 25-3 Flow Diagram for Erasing Data Flash Area

[Note]

- **The CPU continues program processing even while data flash erase is in progress. Do not enter the STOP mode, STOP-D mode or HALT-H mode during the erase. In addition, set the FSELF bit of the FLASHSLF register to "0" after the erase is completed.**
- **The data flash area is unreadable during erasing.**
- **For block/sector erase, place two NOP instructions following the instruction used to set FERS/FSERS bits of the FLASHCON register to "1".**

Figure 25-4 shows the flow diagram for programming the data flash area.

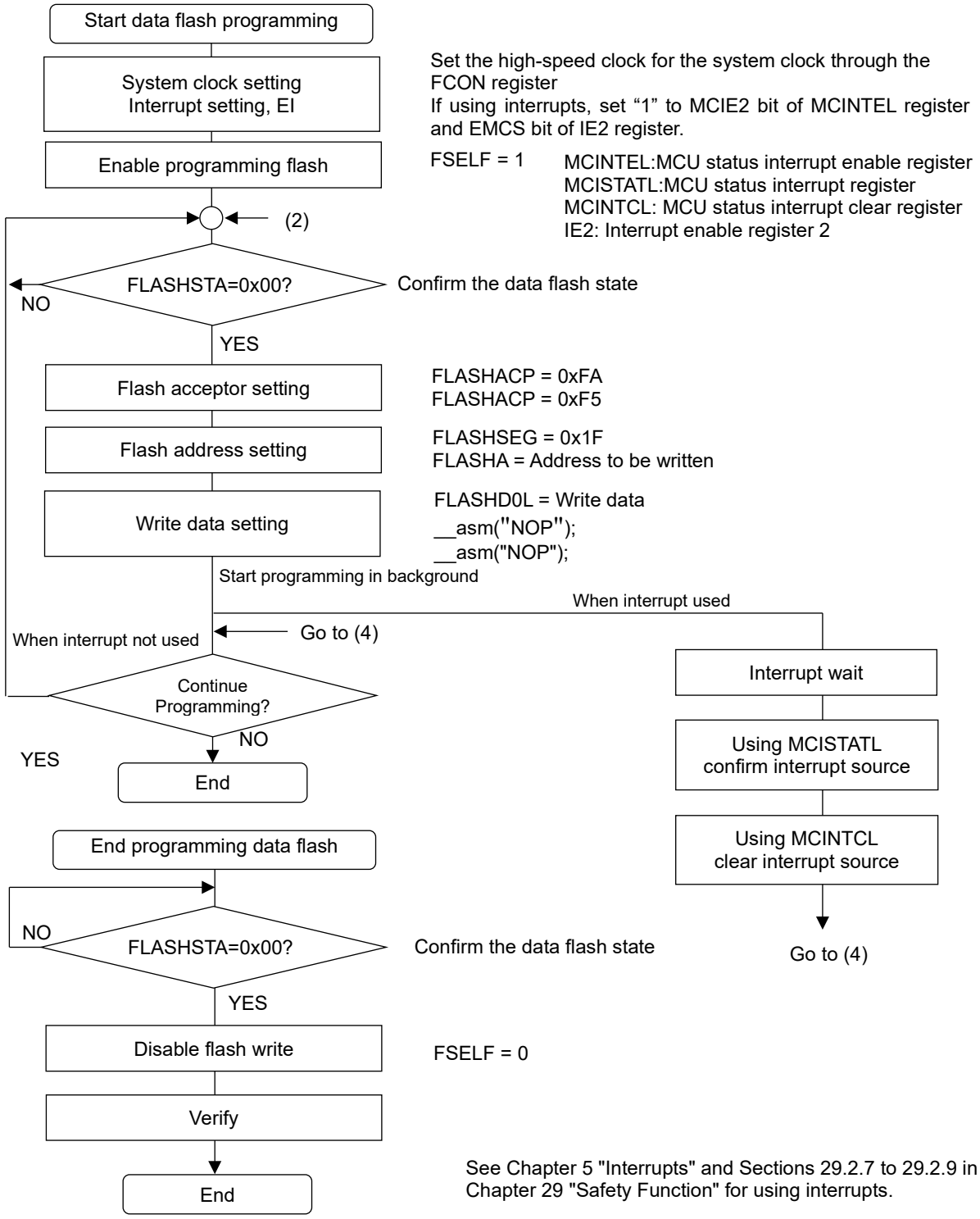


Figure 25-4 Flow Diagram for Programming Data Flash Area

[Note]

- **The CPU continues program processing even while data flash programming is in progress. Do not enter the STOP mode, STOP-D mode or HALT-H mode during the programming. In addition, set the FSELF bit of the FLASHSLF register to "0" (erase/program disabled) after the programming ended.**
- **The data flash area is unreadable during programming.**
- **For data programming setting, place two NOP instructions following the instruction used to set the programming data in the FLASHD0L register.**

25.3.4 Notes on use of self-programming

Table 25-5 shows the notes on the use of self-programming (block erase/sector erase/program).

Table 25-5 Notes on Use of Self-programming

Item	Notes
System clock during use of self-programming	Set to high-speed clock. See "Chapter 6 Clock Generation Circuit" for enabling the high-speed clock oscillation and switching the system clock.
If power outage or forced termination due to a reset occurs	Data in flash memory is not guaranteed. Perform block/sector erase again then program data.
If LSI does not start up due to occurrence of power outage or forced termination during programming (*1)	Program the program again using on-chip debug emulator or ISP function.

*1: While programming the block or sector including address 0:0000 of the program area is in progress.

25.4 In-System Programming Function

The In-System Programming (ISP) function is used to program a program memory space or data flash area through UART communication with an external device.

25.4.1 Programming Procedure

Figure 25-5 shows the flow diagram for programming the flash memory using the ISP function.

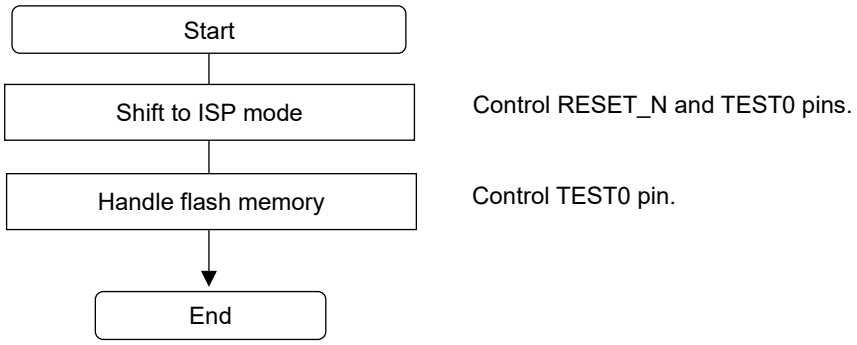


Figure 25-5 Flow Diagram for Programming Flash Memory Using ISP Function

25.4.2 Communication Method

Table 25-6 describes the communication method of the ISP function.

Table 25-6 ISP Function Communication Method

Item	Description
Pins used for ISP function	RESET_N and TEST0 pins
Pins used when entering ISP mode	
Communication Method	Half-duplex UART
Pins used for UART communication	TEST0 pin
UART communication format	8-bit length, LSB first, 1 stop bit, no parity bit
Baud rate	Automatic detection between 4800 bps to 2.0 Mbps

[Note]

- **The UART communication for the ISP function might be affected due to an error of the baud rate and slow slope of the signals. Be sure to evaluate the operation.**

25.4.3 Communication Command

3-byte commands are used to make the communication in the ISP function.
Table 25-7 shows the ISP mode commands.

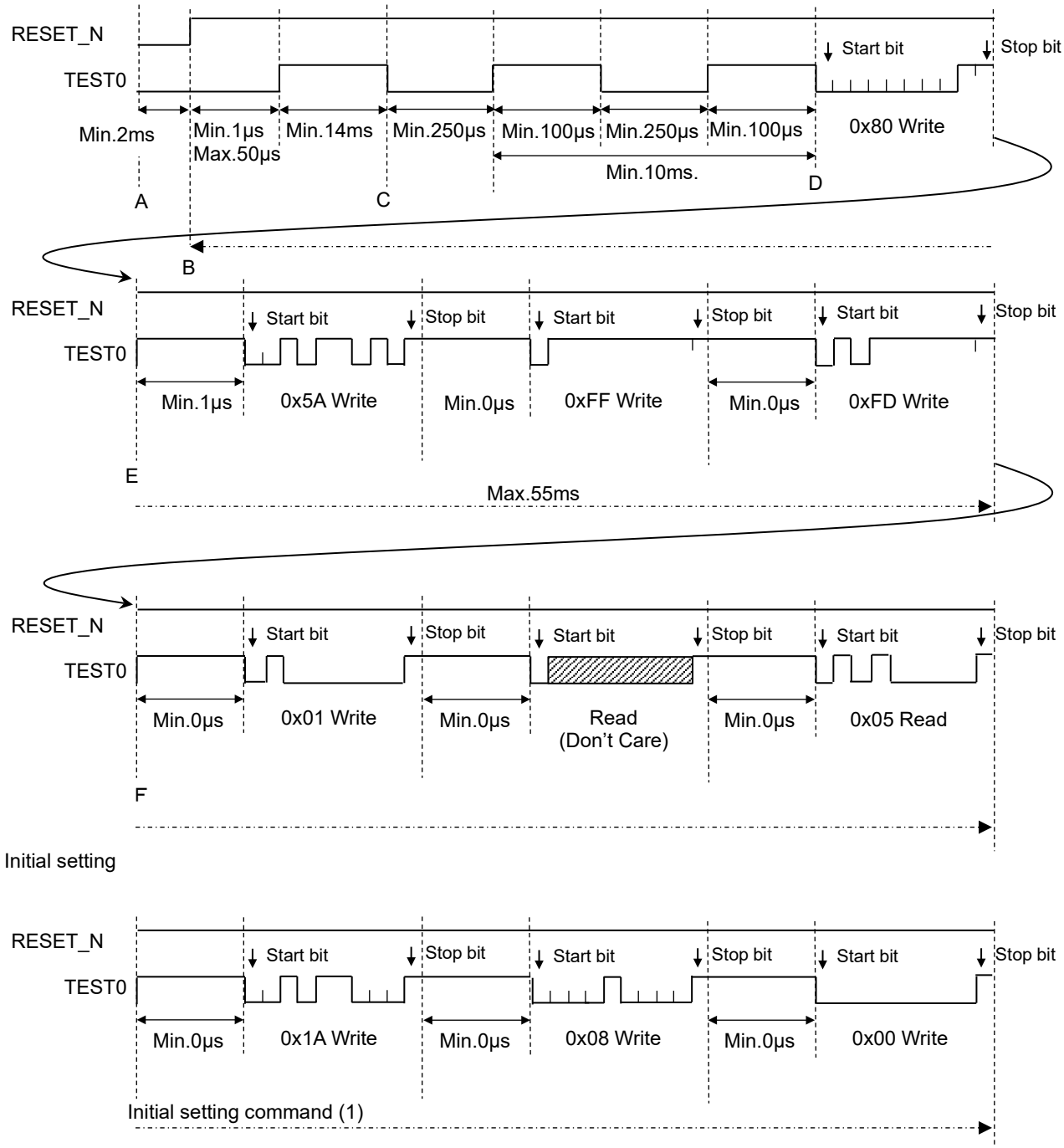
Table 25-7 ISP Mode Command List

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	0x00:1F (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 Transition Command to ISP Mode

Figure 25-6 shows the timing diagram for transition to the ISP mode.



A. System reset. Assert RESET_N="L" and TEST0="L".
B. Raise the RESET_N from "L" to "H" during the TEST0 is "L".
C. Transmit "L" level for min.250 µs twice on the TEST0 pin.
D. Transmit data 0x80 using the UART communication baud rate configured in the host side.
E. Transmit data using the UART communication baud rate configured in the host side in the following order:
0x5A → 0xFF → 0xFD
F. Transmit 0x01 using the UART communication baud rate configured in the host side.
Repeat the steps A to F until the 3rd byte becomes 0x05.

Figure 25-6 Timing Diagram for Transition to ISP Mode

[Note]

- **The transition process from point B to the end of initial setting command (1) shown in Figure 25-6 needs to be completed within 55ms.**
- **Except ISP mode, don't set TEST0 to "L" when RESET_N is raising to "H".**

25.4.5 Flash Memory Handling

Figure 25-7 shows the flow diagram for erasing/programming the flash memory after transition to the ISP mode.

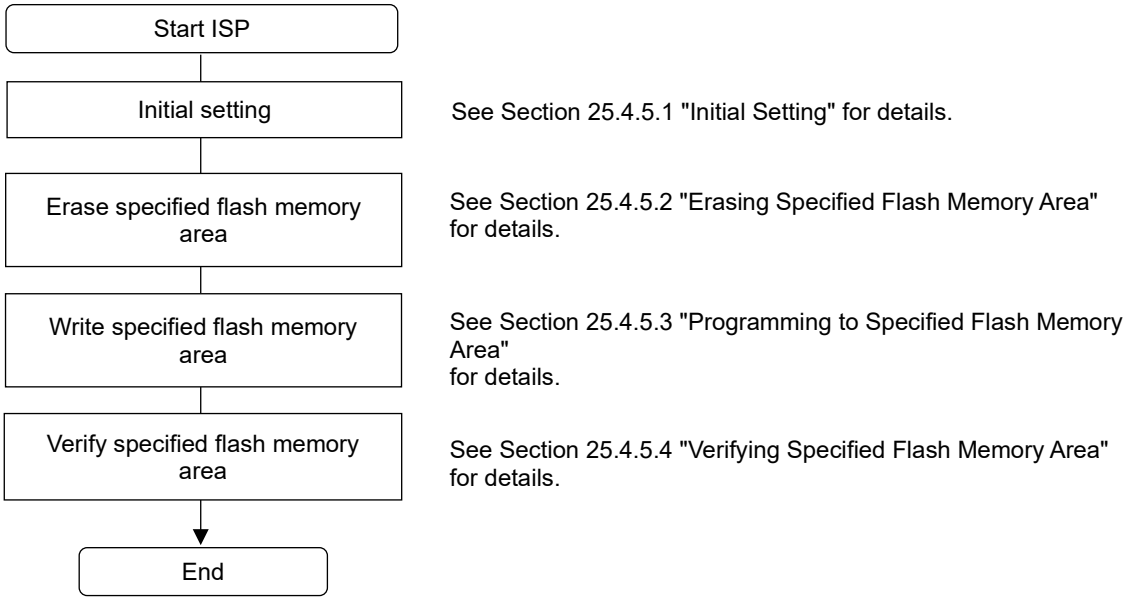


Figure 25-7 Flow Diagram for Erasing/Programming Flash Memory (Overview)

25.4.5.1 Initial Setting

Figure 25-8 shows the initial setting flow.
The flash memory erase/program protection is released at the time of initial setting.

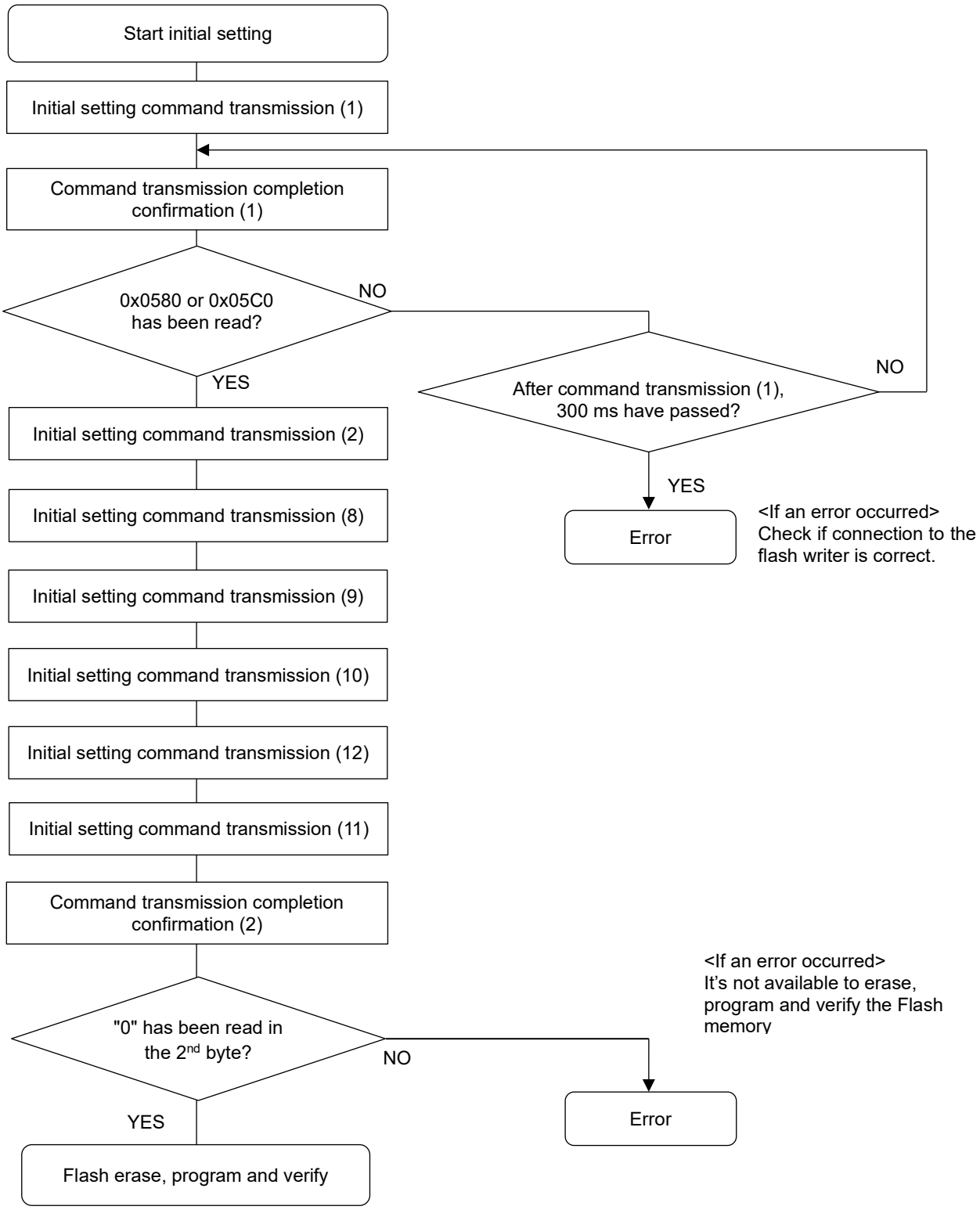


Figure 25-8 Initial Setting Flow Diagram

[Note]

- **The initialization process needs to be completed within 1000ms.**

25.4.5.2 Erasing Specified Flash Memory Area

Figure 25-9 shows the flow diagram for erasing the specified flash memory area.

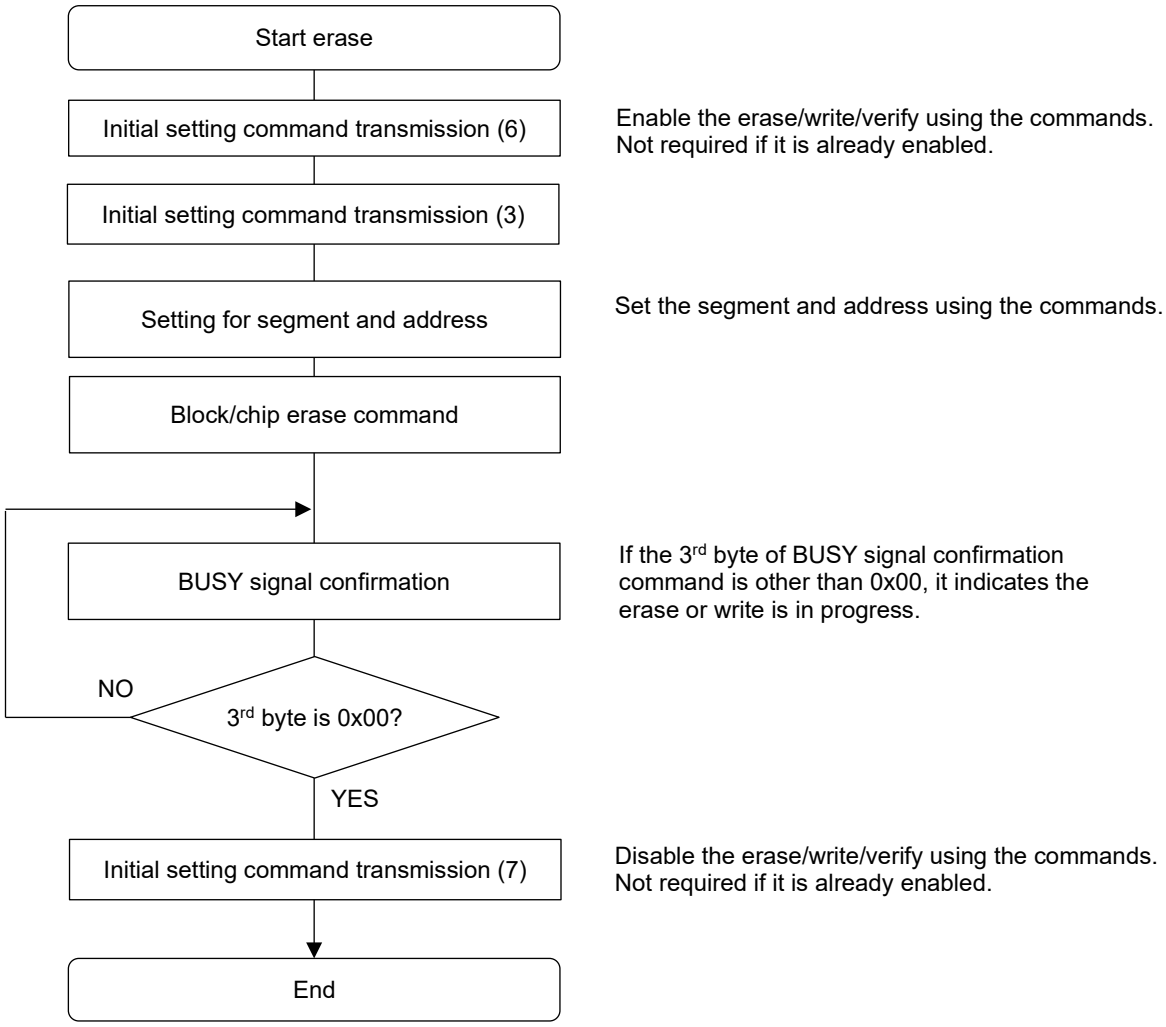


Figure 25-9 Flow Diagram for Erasing Specified Flash Memory Area

[Note]

- **The erase process needs to be completed within 500ms.**

25.4.5.3 Programming to Specified Flash Memory Area

Figure 25-10 shows the flow diagram for programming to the specified flash memory area.

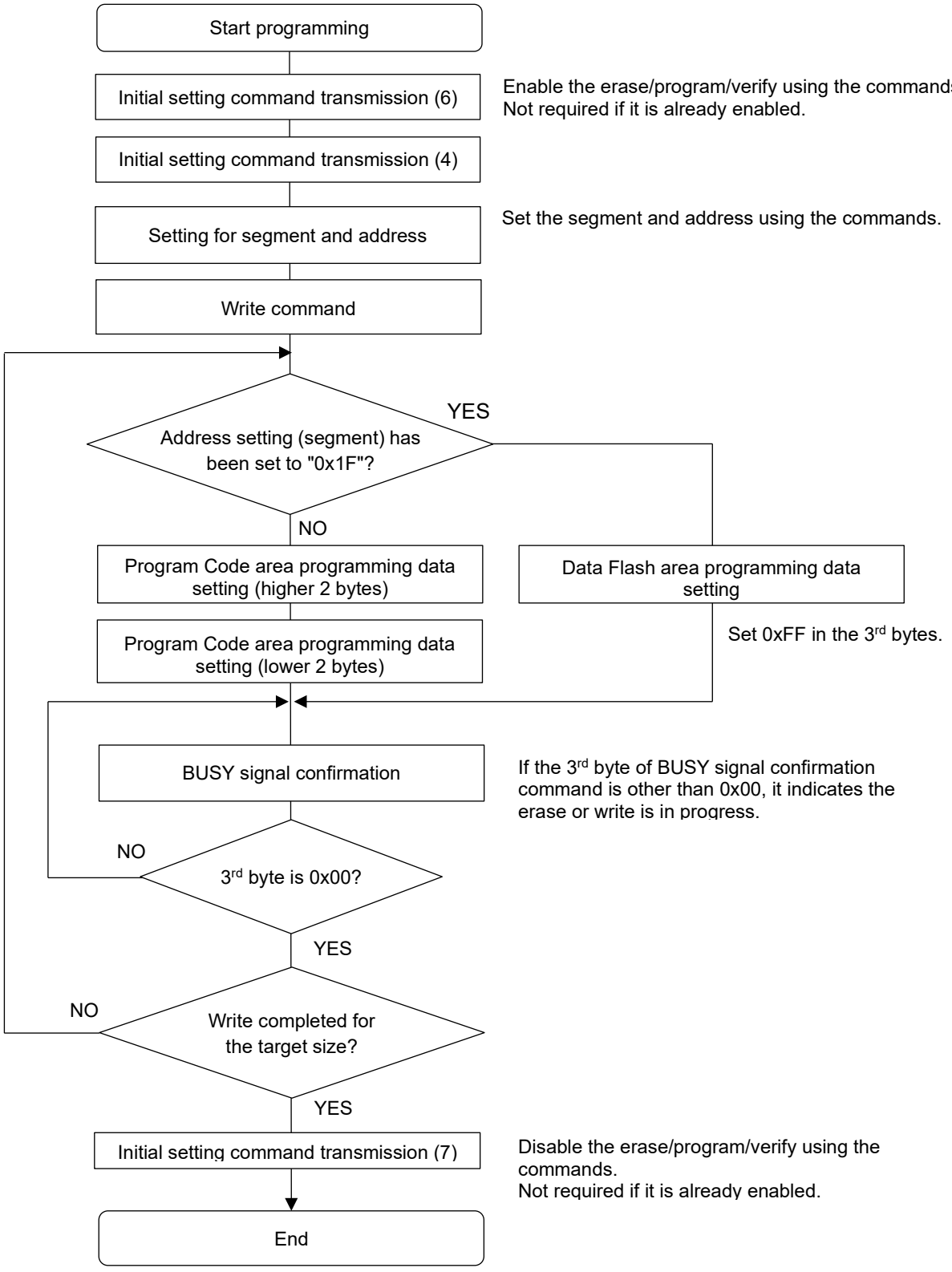


Figure 25-10 Flow Diagram for Programming to Specified Flash Memory Area

[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 Verifying Specified Flash Memory Area

Figure 25-11 shows the flow diagram for verifying the specified flash memory area.

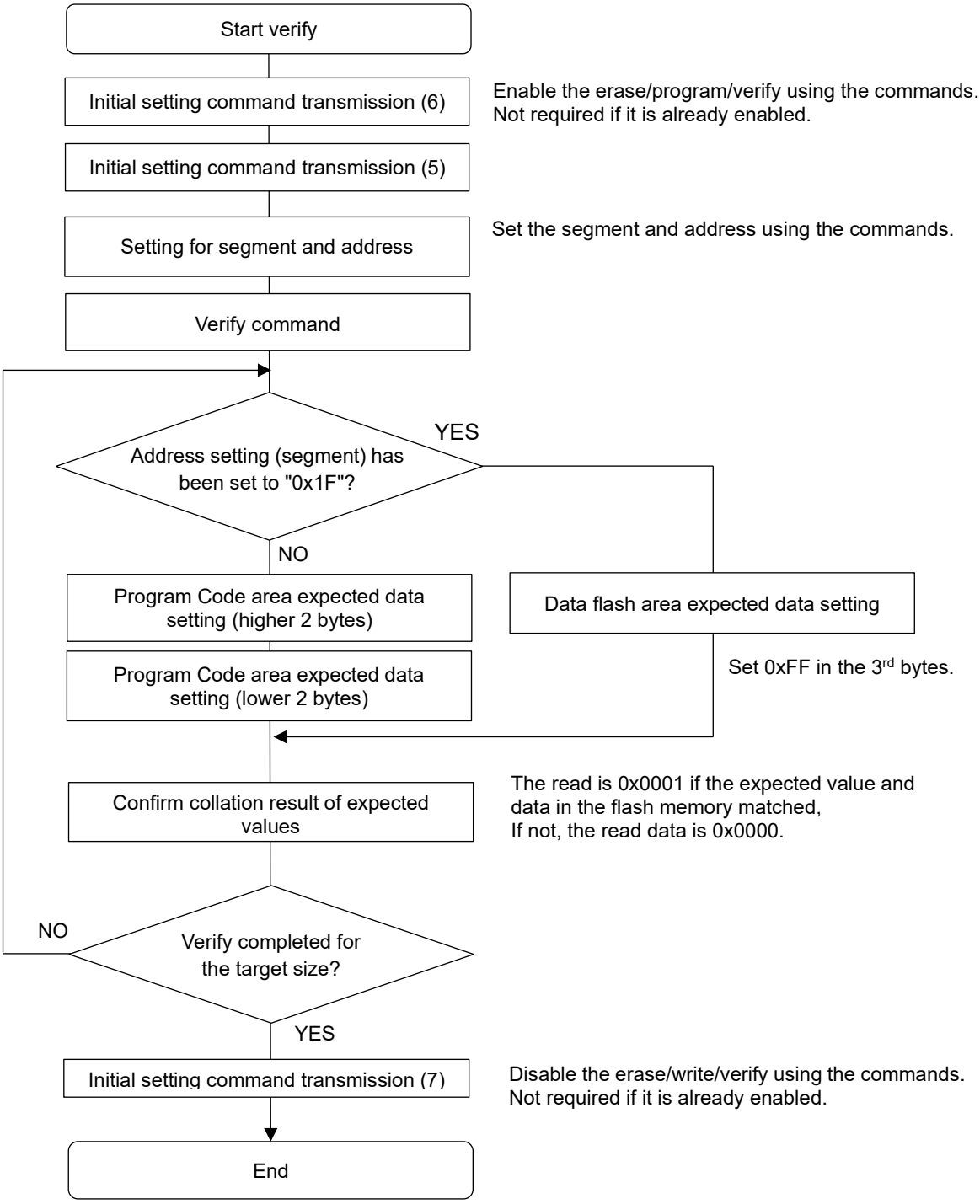


Figure 25-11 Flow Diagram for Verifying Specified Flash Memory Area

[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.**

25.4.6 Advanced Control of Flash Memory Erasing/Programming

This section describes how to implement shorter Flash memory erasing/programing time.

The erasing/programming flow shown in section 25.4.5.2 and 25.4.5.3 is to confirm the busy signal after transmitting the erase command or the programming data before transmitting a next command or data. The processing time is reducible by transmitting the commands during the time the BUSY signal is released (t_{busy}).

25.4.6.1 Timing to transmit command of Advanced Control

The LSI executes erasing/programming instructions to the Flash memory when it receives communication commands for the erasing/programming. It requires the Busy time (t_{busy}) as an interval time to accept the next communication command. Therefore, transmit the communication commands for erasing/programming the Flash memory with an interval of longer than t_{busy}.

The timing of command transmit is calculated as follows.

Command transfer time : t_{cmd} = (10[bit] / transfer rate[bps])
Wait time : t_{wait} = Busy time : t_{busy} – (t_{cmd} × number of commands)

Example of command transmit for programming:

(1) When the transfer rate is 1Mbps and transmitting "Program code area programming data (lower 2 byte)" command, transmit the command at intervals of longer than the Busy time (t_{busy}).

t_{cmd} = 10 bit / 1Mbps = 10 [µs]
t_{wait} = t_{busy} – (t_{cmd} × 2) = 80 - 10 × 2 = 60 [µs]

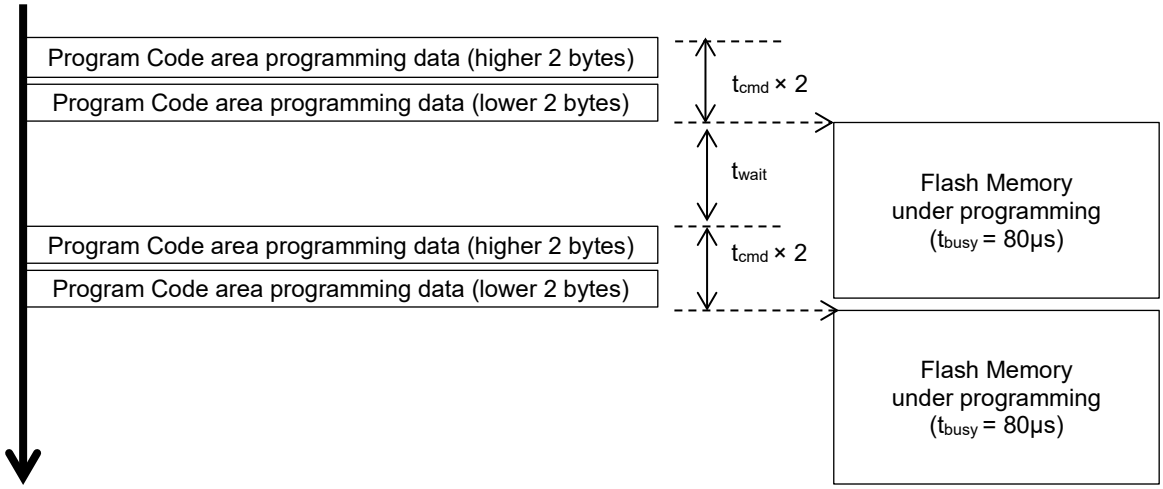


Figure 25-12 Advanced control #1 of programming the program code area

Figure 25-13 shows that the timing of transmitting the "program code area programming data (higher 2 bytes)" command is flexible, it just has to be put before transmitting the "program code area programming data (lower 2 bytes)" command.

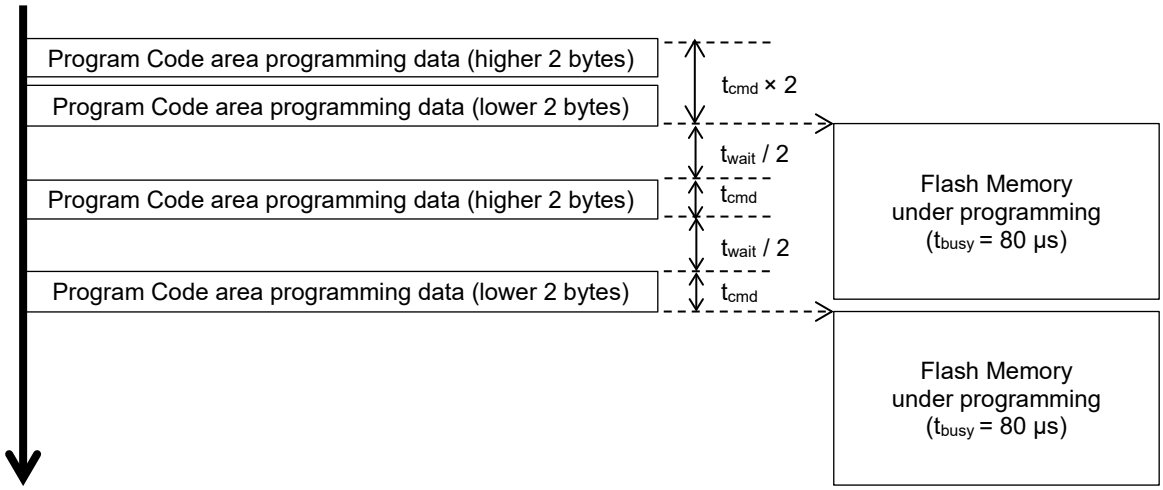


Figure 25-13 Advanced control #2 of programming the program code area

(2) When the transfer rate is 1Mbps and transmitting "Data Flash area programming data" command, transmit the command at intervals of longer than the Busy time (t_{busy}).

t_{cmd} = 10 bit / 1Mbps = 10 [µs]
t_{wait} = $t_{busy} - (t_{cmd} \times 2)$ = $40 - 10 \times 1$ = 30 [µs]

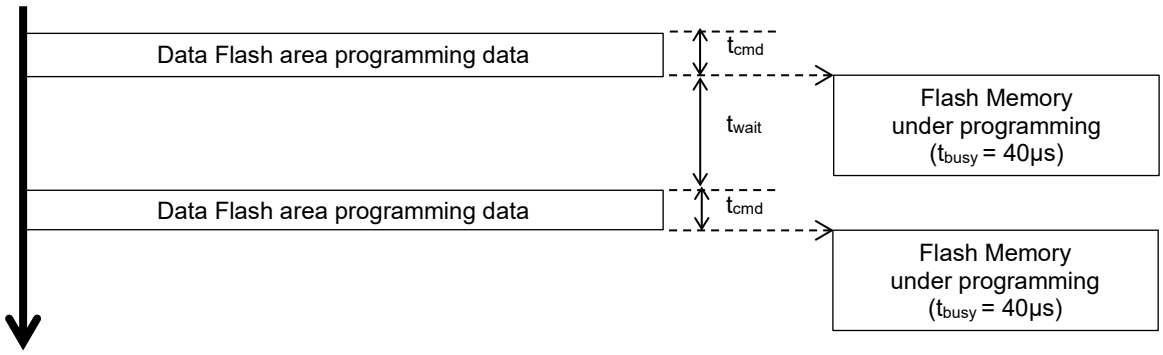


Figure 25-14 Advanced control of programming the data flash area

25.4.6.2 Time-out in Advanced Control

The ISP has a time-out function that the LSI exits from the SIP mode if a certain time has passed. Please transmit one of the following commands at intervals of shorter than 2800ms.

- "Command transmission completion confirmation (1)" command
- "Busy signal confirmation" command
- "Confirm collation result of expected values" command

25.4.6.3 Erasing Specified Flash Memory Area (Advanced control)

Figure 25-15 shows the flow diagram for erasing the specified flash memory area by the advanced control.

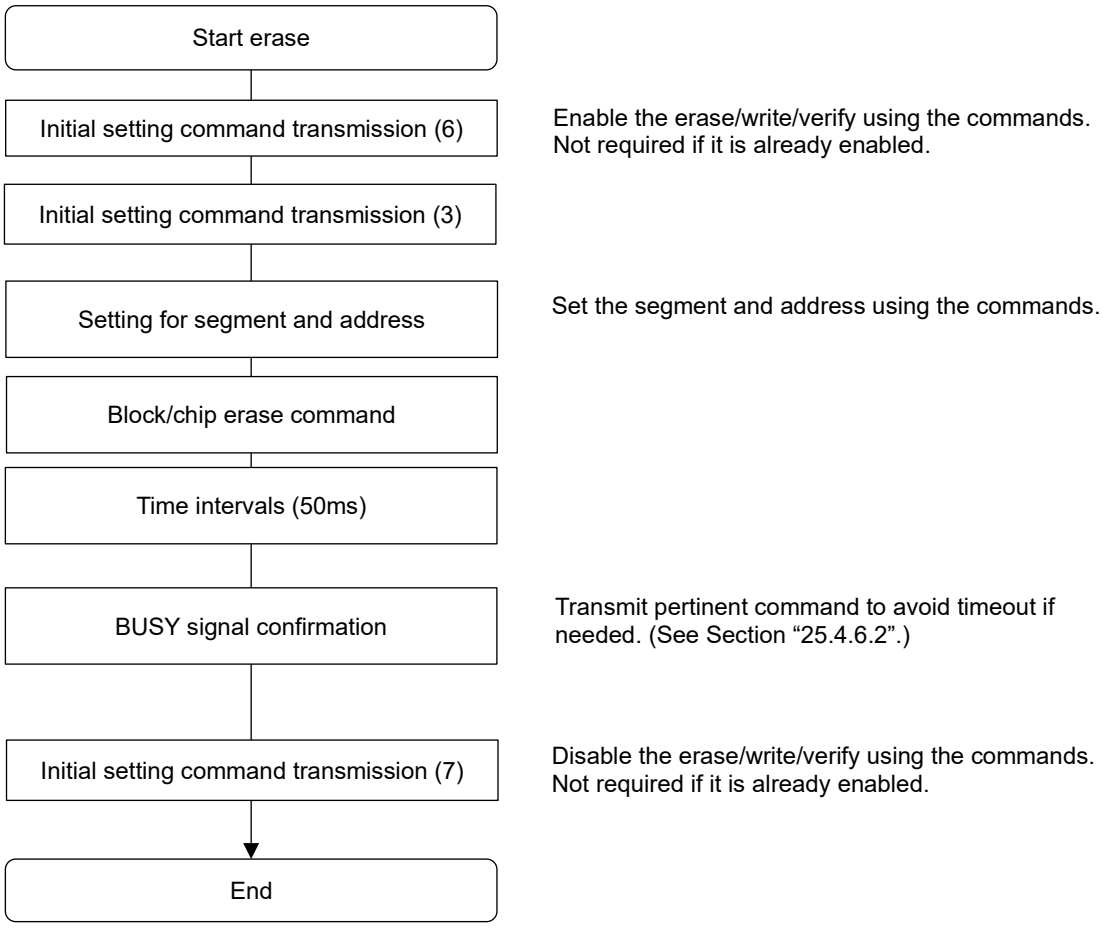


Figure 25-15 Flow Diagram for Erasing Specified Flash Memory Area (Advanced Control)

25.4.6.4 Programming Specified Flash Memory Area (Advanced control)

Figure 25-16 shows the flow diagram for programming to the specified flash memory area by the advanced control.

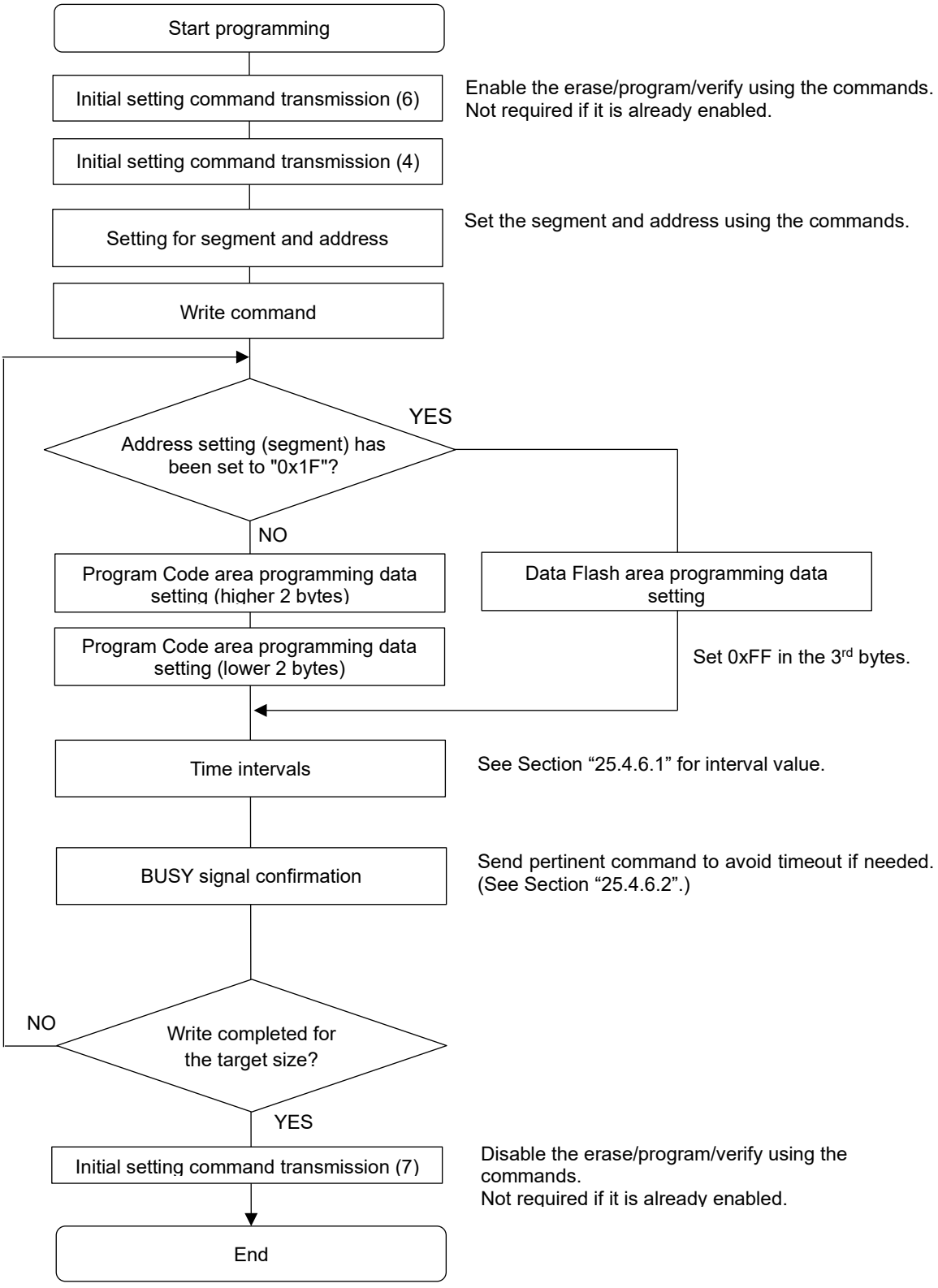


Figure 25-16 Flow Diagram for Programming Specified Flash Memory Area (Advanced Control)

Chapter 26 Code Option

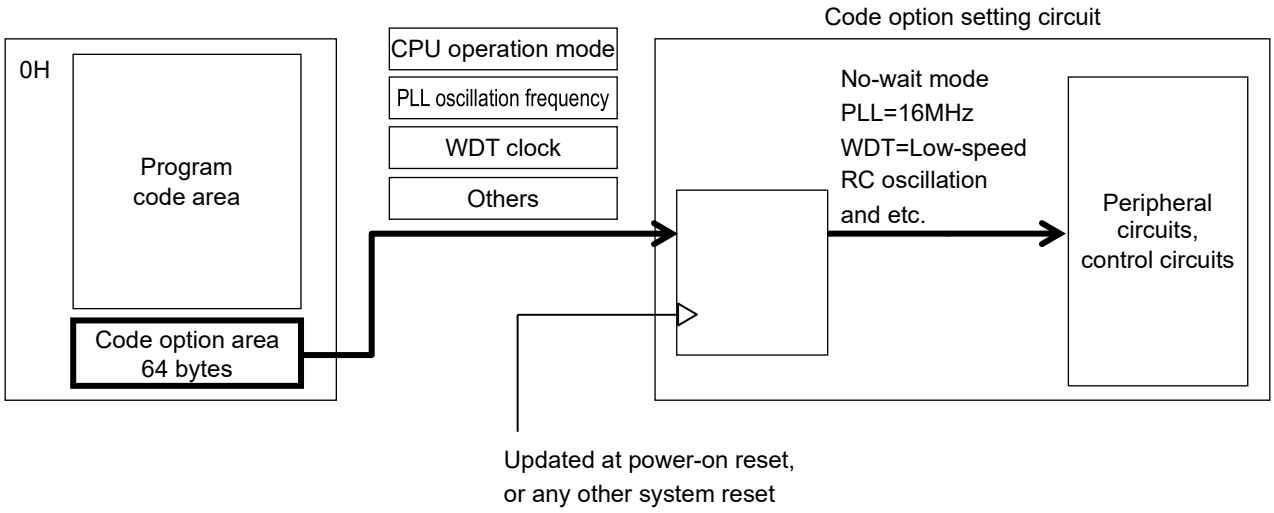
26. Code Option

26.1 General Description

The code option is used to choose the CPU operating mode, PLL reference frequency, watchdog timer operation clock, etc. depending on values written in the code option area of the program memory space.

The hardware automatically refers to data in the code option area when the microcontroller starts up due to one of system resets described below to set each function.

- Power-on reset
- Voltage Level Supervisor reset
- RESET_N pin reset
- Watchdog timer (WDT) overflow reset
- Watchdog timer (WDT) invalid clear reset
- RAM parity error reset
- Unused ROM area access reset



The code option area can be erased or programmed through the on-chip debug function, self-rewrite function of flash memory, or ISP function.

Figure 26-1 Code Option Overview

26.1.1 Function List

The following choices are available by the code option.

- Enabling or disabling the unused ROM area access reset
- Enabling or disabling the remapping function
- Watchdog timer operation clock (low-speed oscillation LSCLK/WDT oscillation)
- Enabling or disabling the watchdog timer operation
- PLL reference frequency (16 MHz or 24 MHz)
- CPU operation mode (wait mode or no-wait mode)
- The software remap or hardware remap is selectable for the remap function

26.2 Description of Code Option

26.2.1 Code Options 0 (CODEOP0)

This is the symbol assigned to address in the code option area of the program memory space (different from the special function registers (SFR)).

Address: (See Table 26-1)
Initial value: 0xFFFF (Erased or factory default setting for products with blank flash memory)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CODEOP0															
Bit	-	-	-	PCERMD	-	-	-	REMAPMD	-	-	-	-	-	WDTNMCK	WDTSPMD	WDTMD
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit name	Description
15 to 13	-	Reserved bits
12	PCERMD	This bit is used to choose to enable/disable the unused ROM area access reset. See Chapter 29.3.2 "Unused ROM Area Access Reset Function" for the unused ROM area access reset. 0: Unused ROM area access reset disabled 1: Unused ROM area access reset enabled (Initial value)
11 to 9	-	Reserved bits
8	REMAPMD	This bit is used to choose to enable/disable the remapping function (software remap or hardware remap) operation. See Chapter 2.8 "Remapping Function" for details of the remapping function. 0: Remapping function operation enabled 1: Remapping function operation disabled (Initial value)
7 to 3	-	Reserved bits
2	WDTNMCK	This bit is used to choose the watchdog timer (WDT) operation clock. 0: Clock with divided frequency (1.024 kHz) of oscillation clock (32.768 kHz) 1: Watchdog timer RC1K oscillation clock (Initial value) See Chapter 10 "Watchdog Timer" for details of the watchdog timer.
1	WDTSPMD	Set this bit to "0".
0	WDTMD	This bit is used to choose to enable/disable the watchdog timer (WDT) operation. 0: WDT operation disabled 1: WDT operation enabled (Initial value)

[Note]
- **Set the WDTSPMD bit to "0".**

26.2.2 Code Options 1 (CODEOP1)

This is the symbol assigned to address in the code option area of the program memory space (different from the special function registers (SFR)).

Address: (See Table 26-1)
Initial value: 0xFFFF (Erased or factory default setting for products with blank flash memory)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CODEOP1															
Bit	-	-	-	-	-	-	-	-	-	-	-	-	PLLMD1	PLLMD0	CPUMD1	CPUMD0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit name	Description
15 to 4	-	Reserved bits
3, 2	PLLMD1, PLLMD0	These bits are used to choose the PLL reference frequency. 00: Do not use 01: Do not use 10: PLL reference frequency=24 MHz 11: PLL reference frequency=16 MHz (Initial value) The following table shows the relation between the PLL reference frequency and the maximum operating frequency of CPU and peripheral circuits.

PLL reference frequency	Maximum operating frequency		
	Peripheral circuit	CPU (wait mode)	CPU (no-wait mode)
24MHz	24MHz	24MHz	6MHz
16MHz	16MHz	16MHz	8MHz

		See Chapter 2 "CPU and Memory Space" and Appendix C "Instruction Execution Cycle" for the CPU operation modes (wait mode and no-wait mode).
1, 0	CPUMD1, CPUMD0	These bits are used to choose the CPU operation mode. 00: Prohibited to use (wait mode) 01: Wait mode 10: Prohibited to use (no-wait mode) 11: No-wait mode (Initial value)

26.2.3 Code Options 2 (CODEOP2)

This is the symbol assigned to address in the code option area of the program memory space (different from the special function registers (SFR)).

Address: (See Table 26-1)
Initial value: 0xFFFF (factory default setting for products with blank flash memory)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CODEOP2															
Bit	CREMAPMD	CRES2	CRES1	CRES0	CREA15	CREA14	CREA13	CREA12	-	-	-	-	-	-	-	-
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit name	Description
15	CREMAPMD	This bit is used to control the initial value of Flash Remap Address Register (REMAPADD) at the system reset. 0: The initial value of the REMAPADD consists of CREA15 to 12 bits and CRES2 to 0 bits 1: The initial value of the REMAPADD is 0x00 If setting this bit to "0", The initial value of the REMAPADD consists of CREA15 to 12 bits and CRES2 to 0 bits. For details on REMAPADD, see Section 2.7.3 "Flash Remap Address Register (REMAPADD)". The MCU remaps to the address specified with the CREA15 to 12 bits and the CRES2 to 0 bits every time at the system reset. See also Section 2.8.3 "Code Option Remap". The remap function is enabled by setting REMAPMD bit of the Code Options 0.
14 to 12	CRES2 to CRES0	These bits are used to set the initial values of RES2 to RES0 bits of the Flash Remap Address Register (REMAPADD). The CRES2 and RES2 bit are available on the products with 384KB/512KB ROM
11 to 8	CREA15 to CREA12	These bits are used to set the initial values of REA15 to REA12 bits of the Flash Remap Address Register (REMAPADD).
7 to 0	-	Reserved bits

CPU instruction execution start address after releasing the reset

Reset	REMAPMD	CREMAPMD	Remap function	CPU instruction execution start address
CPU reset (BRK instruction)	1	1	Disable	0x0000
	1	0		
	0	1	Enable (Software Remap)	Address set in the REMAPADD register
	0	0		
System reset (See “3.3.2. System Reset Mode” for detail)	1	1	Disable	0x0000
	1	0		
	0	1		
	0	0	Enable (Code Option Remap)	Initial data of the REMAPADD register (data set by the Code Options 2)

See Section 2.7.3 "Flash Remap Address Register (REMAPADD)" and Section 2.8.3 "Code Option Remap".

26.3 Code Option Data Setting

The address of code option area is dependent of the size of the program memory space (flash memory).
Table 26-1 shows addresses of code option areas for each product.

Table 26-1 List of Addresses of Code Option Areas for Each Product

ML62Q1300 group

Product name	Program memory space size	Code Option area	Address		
			CODEOP2	CODEOP1	CODEOP0
ML62Q1323/1333	16K byte	0x0:3FC0 to 0x0:3FFF	0x0:3FD4	0x0:3FD2	0x0:3FD0
ML62Q1324/1334	24K byte	0x0:5FC0 to 0x0:5FFF	0x0:5FD4	0x0:5FD2	0x0:5FD0
ML62Q1325/1335/1345 /1365	32K byte	0x0:7FC0 to 0x0:7FFF	0x0:7FD4	0x0:7FD2	0x0:7FD0
ML62Q1346/1366	48K byte	0x0:BFC0 to 0x0:BFFF	0x0:BFD4	0x0:BFD2	0x0:BFD0
ML62Q1347/1367	64K byte	0x0:FFC0 to 0x0:FFFF	0x0:FFD4	0x0:FFD2	0x0:FFD0

ML62Q1500/ML62Q1800 group

Product name	Program memory space size	Code Option area	Address		
			CODEOP2	CODEOP1	CODEOP0
ML62Q1530/1540/1550	32K byte	0x0:7FC0 to 0x0:7FFF	0x0:7FD4	0x0:7FD2	0x0:7FD0
ML62Q1531/1541/1551	48K byte	0x0:BFC0 to 0x0:BFFF	0x0:BFD4	0x0:BFD2	0x0:BFD0
ML62Q1532/1542/1552	64K byte	0x0:FFC0 to 0x0:FFFF	0x0:FFD4	0x0:FFD2	0x0:FFD0
ML62Q1533/1543/1553 /1563/1573	96K byte	0x1:7FC0 to 0x1:7FFF	0x1:7FD4	0x1:7FD2	0x1:7FD0
ML62Q1534/1544/1554 /1564/1574	128K byte	0x1:FFC0 to 0x1:FFFF	0x1:FFD4	0x1:FFD2	0x1:FFD0
ML62Q1555/1565/1575	160K byte	0x2:7FC0 to 0x2:7FFF	0x2:7FD4	0x2:7FD2	0x2:7FD0
ML62Q1556/1566/1576	192K byte	0x2:FFC0 to 0x2:FFFF	0x2:FFD4	0x2:FFD2	0x2:FFD0
ML62Q1557/1567/1577	256K byte	0x3:FFC0 to 0x3:FFFF	0x3:FFD4	0x3:FFD2	0x3:FFD0
ML62Q1858/1868/1878	384K byte	0x5:FFC0 to 0x5:FFFF	0x5:FFD4	0x5:FFD2	0x5:FFD0
ML62Q1859/1869/1879	512K byte	0x7:FFC0 to 0x7:FFFF	0x7:FFD4	0x7:FFD2	0x7:FFD0

ML62Q1700 group

Product name	Program memory space size	Code Option area	Address		
			CODEOP2	CODEOP1	CODEOP0
ML62Q1700/1710/1720	32K byte	0x0:7FC0 to 0x0:7FFF	0x0:7FD4	0x0:7FD2	0x0:7FD0
ML62Q1701/1711/1721	48K byte	0x0:BFC0 to 0x0:BFFF	0x0:BFD4	0x0:BFD2	0x0:BFD0
ML62Q1702/1712/1722	64K byte	0x0:FFC0 to 0x0:FFFF	0x0:FFD4	0x0:FFD2	0x0:FFD0
ML62Q1703/1713/1723 /1733/1743	96K byte	0x1:7FC0 to 0x1:7FFF	0x1:7FD4	0x1:7FD2	0x1:7FD0
ML62Q1704/1714/1724 /1734/1744	128K byte	0x1:FFC0 to 0x1:FFFF	0x1:FFD4	0x1:FFD2	0x1:FFD0
ML62Q1725/1735/1745	160K byte	0x2:7FC0 to 0x2:7FFF	0x2:7FD4	0x2:7FD2	0x2:7FD0
ML62Q1726/1736/1746	192K byte	0x2:FFC0 to 0x2:FFFF	0x2:FFD4	0x2:FFD2	0x2:FFD0
ML62Q1727/1737/1747	256K byte	0x3:FFC0 to 0x3:FFFF	0x3:FFD4	0x3:FFD2	0x3:FFD0
ML62Q1728/1738/1748	384K byte	0x5:FFC0 to 0x5:FFFF	0x5:FFD4	0x5:FFD2	0x5:FFD0
ML62Q1729/1739/1749	512K byte	0x7:FFC0 to 0x7:FFFF	0x7:FFD4	0x7:FFD2	0x7:FFD0

Figure 26-2 shows an example of a code option setting program (for products with the program memory space=64 Kbytes). The setting is described in the start-up file (ML621xxx.ASM) of each product.
Set every unused bit of the code option data area to "1".
For products with blank flash memory, every bit has been set to "1" as the factory default setting.

```
;-------------------------------------------------------------------------
;           Filling the Test area at code memory
;-------------------------------------------------------------------------
   cseg #0 at       0ffc0h   ; address
            dw       0ffffh   ; 0ffc0h
            dw       0ffffh   ; 0ffc2h
            dw       0ffffh   ; 0ffc4h
            dw       0ffffh   ; 0ffc6h
            dw       0ffffh   ; 0ffc8h
            dw       0ffffh   ; 0ffcah
            dw       0ffffh   ; 0ffcch
            dw       0ffffh   ; 0ffceh
            dw       0eef8h   ; 0ffd0h(CODEOP0)
                              ; Unused ROM area access reset disabled, remapping operation enabled, WDT operation disabled
            dw       0fff9h   ; 0ffd2h(CODEOP1)
                              ; PLL frequency 24 MHz, CPU wait mode
            dw       04dffh   ; 0ffd4h(CODEOP2)
                              ; Remapping address(0:d000h)
            dw       0ffffh   ; 0ffd6h
            dw       0ffffh   ; 0ffd8h
            dw       0ffffh   ; 0ffdah
            dw       0ffffh   ; 0ffdch
            dw       0ffffh   ; 0ffdeh

   cseg #0 at 0ffe0h         ; address
            dw       0ffffh   ; 0ffe0h
            ……
            ……
```

Figure 26-2 Example of Code Option Data Program (for Products with the Program Memory Space = 64 Kbytes).

[Note]

- **For the code option data definition, always use the dw directive instruction to configure the data in the unit of word.**

Chapter 27 LCD Driver

27. LCD driver

27.1 General Description

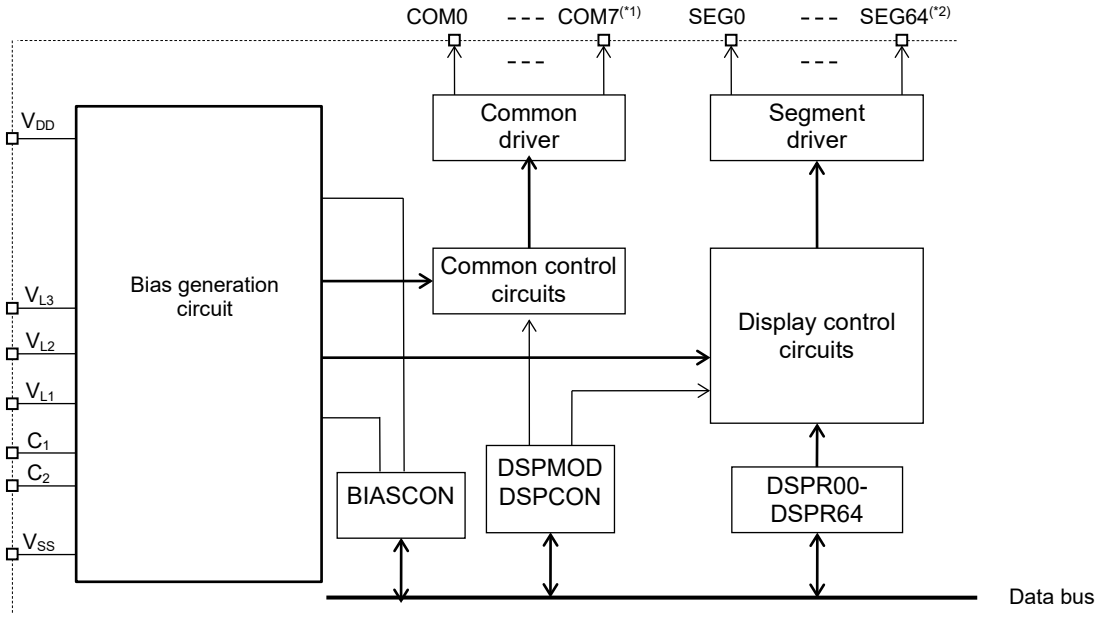
The ML62Q1000 series has the LCD driver that displays the contents of display register 00 to 64 onto a LCD panel.

27.1.1 Features

- Max.480 dots
  ML62Q1700/ML62Q1701/ML62Q1702/ML62Q1703/ML62Q1704:
  24seg×8com (com Max.), 29seg×3com (seg Max.)
  ML62Q1710/ML62Q1711/ML62Q1712/ML62Q1713/ML62Q1714:
  27seg×8com (com Max.), 32seg×3com (seg Max.)
  ML62Q1720/ML62Q1721/ML62Q1722/ML62Q1723/ML62Q1724
  ML62Q1725/ML62Q1726/ML62Q1727/ML62Q1728/ML62Q1729:
  35seg×8com (com Max.), 40seg×3com (seg Max.)
  ML62Q1733/ML62Q1734/ML62Q1735/ML62Q1736/ML62Q1737/ML62Q1738/ML62Q1739:
  45seg×8com (com Max.), 50seg×3com (seg Max.)
  ML62Q1743/ML62Q1744/ML62Q1745/ML62Q1746/ML62Q1747/ML62Q1748/ML62Q1749:
  60seg×8com (com Max.), 65seg×3com (seg Max.)
- Unused segment/common output pins can be used as general-purpose input/output pins.
- 1/1 duty to 1/8 duty
- 1/3 bias (bias generation circuit embedded)
- LCD drive waveform selectable (waveform A or B)
- Frame frequency selectable (approx. 32Hz, 38 Hz, 64 Hz, 75 Hz, 128 Hz or 150 Hz)
- LCD drive voltage generation method selectable (internal boosting/external supply capacitive dividing/internal application dividing/external supply)
- LCD stop, LCD display, all on and all off modes selectable
- LCD reverse available
- LED display control available (external MOS transistors are required for driving)
- Contrast adjustment: Adjusted in 32 levels (only in the internal boosting mode)
- LCD blinking available (the upper 4 bit or lower 4 bit of display register switch automatically or by software in 1/1 to 1/4 duty mode only)

27.1.2 Configuration of LCD Display Function

Figure 27-1 shows the configuration of the LCD display function circuit.



*1: COM0 pin to COM2 pin are shared with general-purpose input/output pins.
COM3 pin to COM7 pin are shared with segment output pins or general-purpose input/output pins.

*2: SEG0 pin to SEG4 pin are shared with common output pins or general-purpose input/output pins.
SEG5 pin to SEG64 pin are shared with general-purpose input/output pins.

BIASCON : Bias control register
DSPMOD : Display mode register
DSPCON : Display control register
DSPR0 0to DSPR64 : Display register 00 to 64

Figure 27-1 Configuration of LCD Driver

27.1.3 Configuration of Bias Generation Circuit

The bias generation circuit operation is selectable from the following four types.

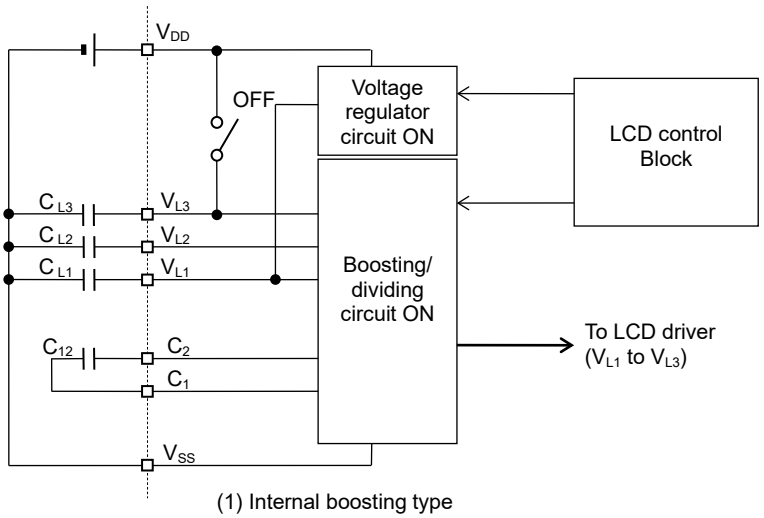
(1) Internal boosting type:
Boost the voltage (V_{L1}) generated from the internal voltage regulator circuit using the capacitor (C_{12}) to output the LCD driving voltages (V_{L1} to V_{L3}).
The display contrast can be adjusted in 32 levels using LCN4 to LCN0 bits in bias control register (BIASCON)..

(2) Internal capacitive dividing type:
Connect V_{L3} to V_{DD} internally and generate V_{L2} and V_{L1} by the means of capacitance voltage division using the capacitor (C_{12}).

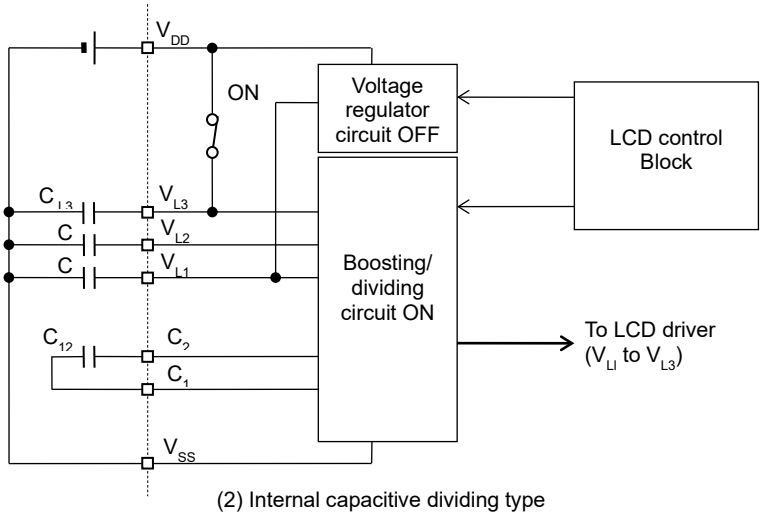
(3) External supply capacitive dividing type:
Supply a voltage to V_{L3} externally and generate V_{L2} and V_{L1} by the means of capacitance voltage division using the capacitor (C_{12}).

(4) External supply type:
Supply voltages to V_{L1} to V_{L3} externally.

Setting the BSON bit of the bias control register (BIASCON) to "1" starts the bias generation circuit operation. For the internal boosting type, the display contrast can be adjusted in 32 levels using LCN4 to LCN0 bits.
Figure 27-2 shows an external configuration example for each type of the bias generation circuit.

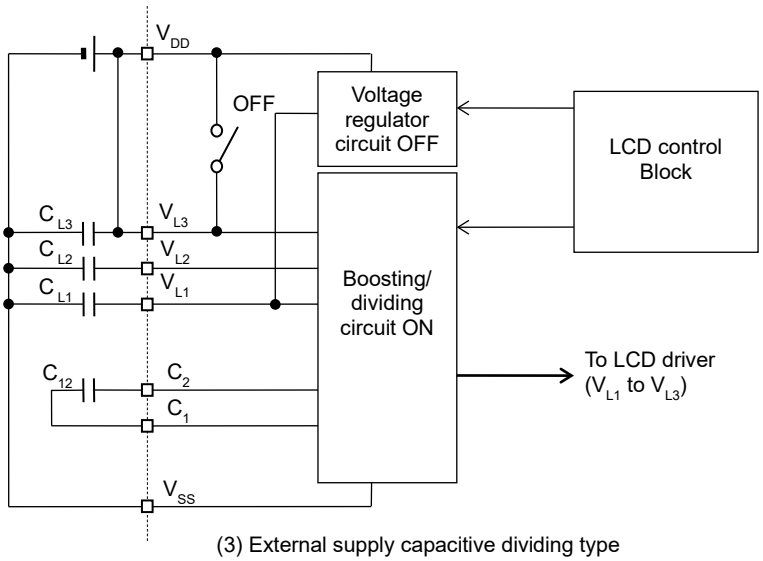


(1) Internal boosting type

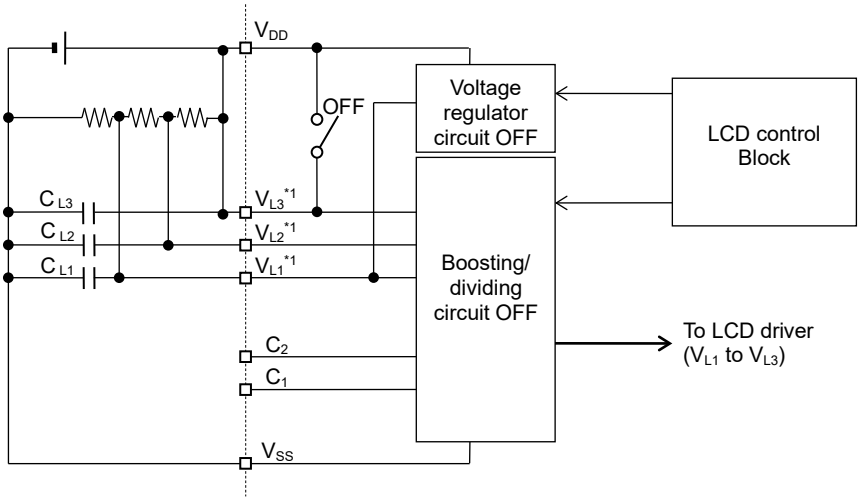


(2) Internal capacitive dividing type

Figure 27-2 External Configuration Example of Bias Generation Circuit (1)



(3) External supply capacitive dividing type



(*1) See “27.3.7 Configuration Example to drive LED” if using the LED drive mode.

(4) External supply type

Figure 27-2 External Configuration Example of Bias Generation Circuit (2)

27.1.4 List of Pins

●: Available -: Unavailable

Pin name	48pin product ML62Q1700/ ML62Q1701/ ML62Q1702/ ML62Q1703/ ML62Q1704	52pin product ML62Q1710/ ML62Q1711/ ML62Q1712/ ML62Q1713/ ML62Q1714	64pin product ML62Q1720/ ML62Q1721/ ML62Q1722/ ML62Q1723/ ML62Q1724/ ML62Q1725/ ML62Q1726/ ML62Q1727/ ML62Q1728/ ML62Q1729	80pin product ML62Q1733/ ML62Q1734/ ML62Q1735/ ML62Q1736/ ML62Q1737/ ML62Q1738/ ML62Q1739	100pin product ML62Q1743/ ML62Q1744/ ML62Q1745/ ML62Q1746/ ML62Q1747/ ML62Q1748/ ML62Q1749	Function
C_1	●	●	●	●	●	Capacitor connection 1 for LCD bias power generation
C_2	●	●	●	●	●	Capacitor connection 2 for LCD bias power generation
V_{L1}	●	●	●	●	●	LCD bias power supply 1
V_{L2}	●	●	●	●	●	LCD bias power supply 2
V_{L3}	●	●	●	●	●	LCD bias power supply 3
P04/COM0	●	●	●	●	●	General-purpose input/output/ Common output
P05/COM1	●	●	●	●	●	General-purpose input/output / Common output
P06/COM2	●	●	●	●	●	General-purpose input/output/ Common output
P07/ COM3/SEG0	●	●	●	●	●	General-purpose input/output / Common output / Segment output
P10/ COM4/SEG1	●	●	●	●	●	General-purpose input/output / Common output / Segment output
P11/ COM5/SEG2	●	●	●	●	●	General-purpose input/output / Common output / Segment output
P12/ COM6/SEG3	●	●	●	●	●	General-purpose input/output/ Common output / Segment output
P13/ COM7/SEG4	●	●	●	●	●	General-purpose input/output / Common output / Segment output
P50/SEG5	●	●	●	●	●	General-purpose input/output/ Segment output
P51/SEG6	-	●	●	●	●	General-purpose input/output / Segment output
P52/SEG7	-	-	●	●	●	General-purpose input/output / Segment output
P53/SEG8	-	-	●	●	●	General-purpose input/output/ Segment output
P90/SEG9	-	-	-	-	●	General-purpose input/output / Segment output
P91/SEG10	-	-	-	-	●	General-purpose input/output / Segment output
P92/SEG11	-	-	-	-	●	General-purpose input/output/ Segment output
P93/SEG12	-	-	-	●	●	General-purpose input/output/ Segment output
P94/SEG13	-	-	-	●	●	General-purpose input/output/ Segment output
P95/SEG14	-	-	-	●	●	General-purpose input/output/ Segment output
P96/SEG15	-	-	-	●	●	General-purpose input/output/ Segment output
P97/SEG16	-	-	-	-	●	General-purpose input/output/ Segment output
PA0/SEG17	-	-	-	-	●	General-purpose input/output/ Segment output
PA1/SEG18	-	-	-	-	●	General-purpose input/output/ Segment output

Pin name	48pin product ML62Q1700/ ML62Q1701/ ML62Q1702/ ML62Q1703/ ML62Q1704	52pin product ML62Q1710/ ML62Q1711/ ML62Q1712/ ML62Q1713/ ML62Q1714	64pin product ML62Q1720/ ML62Q1721/ ML62Q1722/ ML62Q1723/ ML62Q1724/ ML62Q1725/ ML62Q1726/ ML62Q1727/ ML62Q1728/ ML62Q1729	80pin product ML62Q1733/ ML62Q1734/ ML62Q1735/ ML62Q1736/ ML62Q1737/ ML62Q1738/ ML62Q1739	100pin product ML62Q1743/ ML62Q1744/ ML62Q1745/ ML62Q1746/ ML62Q1747/ ML62Q1748/ ML62Q1749	Function
PA2/SEG19	-	-	-	-	●	General-purpose input/output/ Segment output
P54/SEG20	-	-	●	●	●	General-purpose input/output/ Segment output
P55/SEG21	-	-	●	●	●	General-purpose input/output/ Segment output
P14/SEG22	●	●	●	●	●	General-purpose input/output/ Segment output
P15/SEG23	●	●	●	●	●	General-purpose input/output/ Segment output
P16/SEG24	●	●	●	●	●	General-purpose input/output/ Segment output
P17/SEG25	●	●	●	●	●	General-purpose input/output/ Segment output
P20/SEG26	●	●	●	●	●	General-purpose input/output/ Segment output
P21/SEG27	●	●	●	●	●	General-purpose input/output/ Segment output pin
P22/SEG28	●	●	●	●	●	General-purpose input/output/ Segment output
P23/SEG29	●	●	●	●	●	General-purpose input/output/ Segment output
P24/SEG30	●	●	●	●	●	General-purpose input/output/ Segment output
P25/SEG31	●	●	●	●	●	General-purpose input/output/ Segment output
P26/SEG32	●	●	●	●	●	General-purpose input/output/ Segment output
P27/SEG33	●	●	●	●	●	General-purpose input/output/ Segment output
P56/SEG34	-	●	●	●	●	General-purpose input/output/ Segment output
P57/SEG35	-	-	●	●	●	General-purpose input/output/ Segment output
PA3/SEG36	-	-	-	●	●	General-purpose input/output/ Segment output
PA4/SEG37	-	-	-	●	●	General-purpose input/output/ Segment output
PA5/SEG38	-	-	-	-	●	General-purpose input/output/ Segment output
PA6/SEG39	-	-	-	-	●	General-purpose input/output/ Segment output
PA7/SEG40	-	-	-	-	●	General-purpose input/output/ Segment output
PB0/SEG41	-	-	-	-	●	General-purpose input/output/ Segment output
PB1/SEG42	-	-	-	-	●	General-purpose input/output/ Segment output
PB2/SEG43	-	-	-	●	●	General-purpose input/output/ Segment output
PB3/SEG44	-	-	-	●	●	General-purpose input/output/ Segment output
PB4/SEG45	-	-	-	●	●	General-purpose input/output/ Segment output
PB5/SEG46	-	-	-	●	●	General-purpose input/output/ Segment output

Pin name	48pin product ML62Q1700/ ML62Q1701/ ML62Q1702/ ML62Q1703/ ML62Q1704	52pin product ML62Q1710/ ML62Q1711/ ML62Q1712/ ML62Q1713/ ML62Q1714	64pin product ML62Q1720/ ML62Q1721/ ML62Q1722/ ML62Q1723/ ML62Q1724/ ML62Q1725/ ML62Q1726/ ML62Q1727/ ML62Q1728/ ML62Q1729	80pin product ML62Q1733/ ML62Q1734/ ML62Q1735/ ML62Q1736/ ML62Q1737/ ML62Q1738/ ML62Q1739	100pin product ML62Q1743/ ML62Q1744/ ML62Q1745/ ML62Q1746/ ML62Q1747/ ML62Q1748/ ML62Q1749	Function
P40/SEG47	-	-	●	●	●	General-purpose input/output/ Segment output pin
P41/SEG48	-	●	●	●	●	General-purpose input/output/ Segment output
P30/SEG49	●	●	●	●	●	General-purpose input/output/ Segment output
P31/SEG50	●	●	●	●	●	General-purpose input/output/ Segment output
P32/SEG51	●	●	●	●	●	General-purpose input/output/ Segment output
P33/SEG52	●	●	●	●	●	General-purpose input/output/ Segment output
P60/SEG53	●	●	●	●	●	General-purpose input/output/ Segment output
P61/SEG54	●	●	●	●	●	General-purpose input/output/ Segment output
P62/SEG55	●	●	●	●	●	General-purpose input/output/ Segment output
P63/SEG56	●	●	●	●	●	General-purpose input/output/ Segment output
P64/SEG57	●	●	●	●	●	General-purpose input/output/ Segment output
P65/SEG58	●	●	●	●	●	General-purpose input/output/ Segment output
P66/SEG59	●	●	●	●	●	General-purpose input/output/ Segment output
P67/SEG60	-	-	●	●	●	General-purpose input/output/ Segment output
P42/SEG61	-	-	●	●	●	General-purpose input/output/ Segment output
PB6/SEG62	-	-	-	-	●	General-purpose input/output/ Segment output
PB7/SEG63	-	-	-	-	●	General-purpose input/output/ Segment output
P77/SEG64	-	-	-	-	●	General-purpose input/output/ Segment output

27.2 Description of Registers

27.2.1 List of Registers

Address	Name	Symbol(Byte)	Symbol(Word)	R/W	Size	Initial Value
0xF0F0	Bias control register	BIASCONL	BIASCON	R/W	8/16	0x08
0xF0F1		BIASCONH		R/W	8	0x00
0xF0F2	Display mode register	DSPMODL	DSPMOD	R/W	8/16	0x40
0xF0F3		DSPMODH		R/W	8	0x00
0xF0F4	Display control register	DSPCONL	DSPCON	R/W	8/16	0x00
0xF0F5		DSPCONH		R/W	8	0x00
0xF0F6	Segment mode register 0	SEGMOD0L	SEGMOD0	R/W	8/16	0x00
0xF0F7		SEGMOD0H		R/W	8	0x00
0xF0F8	Segment mode register 1	SEGMOD1L	SEGMOD1	R/W	8/16	0x00
0xF0F9		SEGMOD1H		R/W	8	0x00
0xF0FA	Segment mode register 2	SEGMOD2L	SEGMOD2	R/W	8/16	0x00
0xF0FB		SEGMOD2H		R/W	8	0x00
0xF0FC	Segment mode register 3	SEGMOD3L	SEGMOD3	R/W	8/16	0x00
0xF0FD		SEGMOD3H		R/W	8	0x00
0xF0FE	Segment mode register 4	SEGMOD4L	SEGMOD4	R/W	8/16	0x00
0xF0FF		-		R	8	0x00
0xF100 to 0xF140	Display register 00 to Display register 64	DSPR00 to DSPR64	DSPRWn (n:an even number in decimalization)	R/W	8/16	Undefined
				R/W	8	Undefined

27.2.2 Bias Control Register(BIASCON)

BIASCON is a special function register (SFR) used to control the bias generation circuit.
Write the BIASCONL when the display turns off (when the LMD1 bit and LMD0 bit of DSPCON register are "0").
Writing to the BIASCONL is disabled in other modes.

Address: 0xF0F0(BIASCONL/BIASCON), 0xF0F1(BIASCONH)
Access: R/W
Access size: 8/16bits
Initial value: 0x0008

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BIASCON															
Byte	BIASCONH								BIASCONL							
Bit	-	-	-	LCN4	LCN3	LCN2	LCN1	LCN0	BTSEL1	BTSEL0	DSMD1	DSMD0	BSN2	BSN1	BSN0	BSON
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit No.	Bit symbol name	Description
15 to 13	-	Reserved bits
12 to 8	LCN4 to LCN0	These bits are used to adjust the display contrast in 32 voltage levels. It is adjustable by controlling the VL1 voltage. The display contrast depends on the set value. The display contrast turns thicker when the higher value is set, and turns thinner when lower value is set. See the electrical characteristics in the data sheet for more details about the voltage level. Contrast (voltage) 0x00: 0.950V (Initial value) 0x10: 1.350V 0x01: 0.975V 0x11: 1.375V 0x02: 1.000V 0x12: 1.400V 0x03: 1.025V 0x13: 1.425V 0x04: 1.050V 0x14: 1.450V 0x05: 1.075V 0x15: 1.475V 0x06: 1.100V 0x16: 1.500V 0x07: 1.125V 0x17: 1.525V 0x08: 1.150V 0x18: 1.550V 0x09: 1.175V 0x19: 1.575V 0x0A: 1.200V 0x1A: 1.600V 0x0B: 1.225V 0x1B: 1.625V 0x0C: 1.250V 0x1C: 1.650V 0x0D: 1.275V 0x1D: 1.675V 0x0E: 1.300V 0x1E: 1.700V 0x0F: 1.325V 0x1F: 1.725V
7, 6	BTSEL1, BTSEL0	These bits are used to select the means of bias generation circuit operation. Select the external supply type when driving the LED. 00: External supply type (Initial value) 01: Internal boosting type 10: External supply capacitive dividing type 11: Internal capacitive dividing type
5	DSMD1	This bit is used to select the display device. 0: LCD (Initial value) 1: LED

Bit No.	Bit symbol name	Description
4	DSMD0	This bit is used to select the white & black reverse display mode. 0: Normal display (Initial value) 1: White & black reverse display
3 to 1	BSN2 to BSN0	These bits are used to select the clock for boosting the voltage in the bias generation circuit. 000: LSCLK(32.768kHz) 001: 1/2 LSCLK(16.384kHz) 010: 1/4 LSCLK(8.192kHz) 011: 1/8 LSCLK(4.096kHz) 100: 1/16 LSCLK(2.048kHz) (Initial value) 101: 1/32 LSCLK(1.024kHz) 110: 1/64 LSCLK(512Hz) 111: 1/128 LSCLK(256Hz)
0	BSON	This bit is used to control the bias generation circuit operation. Setting the BSON bit to "1" generates the LCD driving voltages (VL1 to VL3). 0: Bias generation circuit turns off (Initial value) 1: Bias generation circuit turns on

27.2.3 Display Mode Register (DSPMOD)

DSPMODH is a special function register (SFR) to control the LCD driver waveform type, frame frequency and duty . Write the DSPMODL when the display turns off (when the LMD1 bit and LMD0 bit of DPSCON register are "0"). Writing to the DSPMODL is disabled in other modes.

Address: 0xF0F2(DSPMODL/DSPMOD), 0xF0F3(DSPMODH)
Access: R/W
Access size: 8/16 bits
Initial value: 0x0040

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	DSPMOD															
Byte	DSPMODH								DSPMODL							
Bit	-	-	-	WTYPE	-	-	-	-	FRM2	FRM1	FRM0	-	DUTY3	DUTY2	DUTY1	DUTY0
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 13	-	Reserved bits
12	WTYPE	This bit is used to select the LCD drive waveform. Select the waveform A when driving the LED. 0: Waveform A:reversed line (Initial value) 1: Waveform B:reversed frame
11 to 8	-	Reserved bits
7 to 5	FRM2 to FRM0	These bits are used to select the frame frequency of the LCD driver. 000: Frame frequency 32 Hz 001: Frame frequency 38 Hz 010: Frame frequency 64 Hz (Initial value) 011: Frame frequency 75 Hz 100: Frame frequency 128 Hz 101: Frame frequency 150 Hz 110: Frequency adjustment check test mode *1 Generates a Frame frequency of 32 Hz by using the adjusted frequency. 111: Frequency adjustment check test mode *1 Generates a Frame frequency of 64 Hz by using the adjusted frequency. *1: The Frequency adjustment check test mode is for monitoring the adjusted frequency of low-speed time base counter. For more details, see Section 7.3.3 “The way of monitoring the frequency on LCD drive outputs“.
4	-	Reserved bits
3 to 0	DUTY3 to DUTY0	These bits are used to set the display duty and common pin. For the COM7 to COM3 pin functions, the settings of these bits are given priority over the segment mode register. 0000: COM pin not used (Initial value). 0001: 1/1 duty. COM0 pin used. 0010: 1/2 duty. COM0 to COM1 pins used. 0011: 1/3 duty. COM0 to COM2 pins used. 0100: 1/4 duty. COM0 to COM3 pins used. 0101: 1/5 duty. COM0 to COM4 pins used. 0110: 1/6 duty. COM0 to COM5 pins used. 0111: 1/7 duty. COM0 to COM6 pins used. 1000: 1/8 duty. COM0 to COM7 pins used. Others: Setting prohibited (COM pin not used).

27.2.4 Display Control Register (DSPCON)

DSPCON is a special function register (SFR) to control the display mode.

Address: 0xF0F4
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	DSPCON															
Byte	DSPCONH								DSPCONL							
Bit	-	-	-	-	-	MMOD2	MMOD1	MMOD0	-	-	-	-	-	-	LMD1	LMD0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 11	-	Reserved bits
10 to 8	MMOD2 to MMOD0	These bits are used to select the ways of display in 1/1 duty to 1/4 duty. When a duty of 1/5 to 1/8 are selected, this function is disabled. 000: Display 1: Display the lower 4-bit data of the display register00 to 64 (DSPR00 to 64) (Initial value). 001: Display 2: Display the higher 4-bit data of the display register00 to 64 (DSPR00 to 64). 010: Switching display 1: Display the data area in the display 1 and display 2 alternately every one second; negedge of T1HZR. 011: Switching display 2: Display the data area in the display 1 and display 2 alternately every 1/2 second; negedge of T2HZR. 100: Switching display 4: Display the data area in the display 1 and display 2 alternately every 1/4 second; negedge of T4HZR. Others: Setting prohibited (Switching display 4)
7 to 2	-	Reserved bits
1, 0	LMD1, LMD0	These bits are used to control the LCD display. Display status — Common/segment pin status 00: Display stop (Initial). — V_{SS} level 01: All off — Normal display mode[*1] : Off waveform Reverse display mode[*1]: On waveform 10: Display — Normal display mode[*1]: Data “1“ of display register 00 to 64 drives On waveform Reverse display mode[*1]: Data “0“ of display register00 to 64 drives Off waveform 11: All on — Normal display mode[*1]: On waveform Reverse display mode[*1]: Off waveform [*1]: Bias control register (BIASCON) DSMD0 bit setting

27.2.5 Segment Mode Register 0 (SEGMOD0)

SEGMOD0 is a special function register (SFR) to select the SEG15 to SEG0 functions.

Address: 0xF0F6
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	SEGMOD0															
Byte	SEGMOD0H								SEGMOD0L							
Bit	S15MD	S14MD	S13MD	S12MD	S11MD	S10MD	S9MD	S8MD	S7MD	S6MD	S5MD	S4MD	S3MD	S2MD	S1MD	S0MD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	S15MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P96 (Initial value) 1: SEG15
14	S14MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P95 (Initial value) 1: SEG14
13	S13MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P94 (Initial value) 1: SEG13
12	S12MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P93 (Initial value) 1: SEG12
11	S11MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P92 (Initial value) 1: SEG11
10	S10MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P91 (Initial value) 1: SEG10
9	S9MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P90 (Initial value) 1: SEG9
8	S8MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P53 (Initial value) 1: SEG8
7	S7MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P52 (Initial value) 1: SEG7
6	S6MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P51. (Initial value) 1: SEG6.
5	S5MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P50 (Initial value) 1: SEG5

Bit No.	Bit symbol name	Description
4	S4MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P13 (Initial value) 1: SEG4
3	S3MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P12 (Initial value) 1: SEG3
2	S2MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P11 (Initial value) 1: SEG2
1	S1MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P10 (Initial value) 1: SEG1
0	S0MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P07 (Initial value) 1: SEG0

[Note]

- **Write the segment mode register in the display stop mode (LMD1 bit and LMD0 bit of DSPCON register is "0") to prevent erroneous displays or damage of panel.**

27.2.6 Segment Mode Register 1(SEGMOD1)

SEGMOD1 is a special function register (SFR) to select the SEG31 to SEG16 functions.

Address: 0xF0F8
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	SEGMOD1															
Byte	SEGMOD1H								SEGMOD1L							
Bit	S31MD	S30MD	S29MD	S28MD	S27MD	S26MD	S25MD	S24MD	S23MD	S22MD	S21MD	S20MD	S19MD	S18MD	S17MD	S16MD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	S31MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P25 (Initial value) 1: SEG31
14	S30MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P24 (Initial value) 1: SEG30
13	S29MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P23 (Initial value) 1: SEG29
12	S28MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P22 (Initial value) 1: SEG28
11	S27MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P21 (Initial value) 1: SEG27
10	S26MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P20 (Initial value) 1: SEG26
9	S25MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P17 (Initial value) 1: SEG25
8	S24MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P16 (Initial value) 1: SEG24
7	S23MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P15 (Initial value) 1: SEG23
6	S22MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P14 (Initial value) 1: SEG22
5	S21MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P55 (Initial value) 1: SEG21

Bit No.	Bit symbol name	Description
4	S20MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P54 (Initial value) 1: SEG20
3	S19MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PA2 (Initial value) 1: SEG19
2	S18MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PA1 (Initial value) 1: SEG18
1	S17MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PA0 (Initial value) 1: SEG17
0	S16MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P97 (Initial value) 1: SEG16

[Note]

- **Write the segment mode register in the display stop mode (LMD1 bit and LMD0 bit of DSPCON register is "0") to prevent erroneous displays or damage of panel.**

27.2.7 Segment Mode Register 2(SEGMOD2)

SEGMOD2 is a special function register (SFR) to select the SEG47 to SEG32 functions.

Address: 0xF0FA
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	SEGMOD2															
Byte	SEGMOD2H								SEGMOD2L							
Bit	S47MD	S46MD	S45MD	S44MD	S43MD	S42MD	S41MD	S40MD	S39MD	S38MD	S37MD	S36MD	S35MD	S34MD	S33MD	S32MD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	S47MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P40 (Initial value) 1: SEG47
14	S46MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PB5 (Initial value) 1: SEG46
13	S45MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PB4 (Initial value) 1: SEG45
12	S44MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PB3 (Initial value) 1: SEG44
11	S43MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PB2 (Initial value) 1: SEG43
10	S42MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PB1 (Initial value) 1: SEG42
9	S41MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PB0 (Initial value) 1: SEG41
8	S40MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PA7 (Initial value) 1: SEG40
7	S39MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PA6 (Initial value) 1: SEG39
6	S38MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PA5 (Initial value) 1: SEG38
5	S37MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PA4 (Initial value) 1: SEG37

Bit No.	Bit symbol name	Description
4	S36MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PA3 (Initial value) 1: SEG36
3	S35MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P57 (Initial value) 1: SEG35
2	S34MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P56 (Initial value) 1: SEG34
1	S33MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P27 (Initial value) 1: SEG33
0	S32MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P26 (Initial value) 1: SEG32

[Note]

- **Write the segment mode register in the display stop mode (LMD1 bit and LMD0 bit of DSPCON register is "0") to prevent erroneous displays or damage of panel.**

27.2.8 Segment Mode Register 3(SEGMOD3)

SEGMOD3 is a special function register (SFR) to select the SEG63 to SEG48 functions.

Address: 0xF0FC
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	SEGMOD3															
Byte	SEGMOD3H								SEGMOD3L							
Bit	S63MD	S62MD	S61MD	S60MD	S59MD	S58MD	S57MD	S56MD	S55MD	S54MD	S53MD	S52MD	S51MD	S50MD	S49MD	S48MD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	S63MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PB7 (Initial value) 1: SEG63
14	S62MD	This bit is used to select the general-purpose input/output and segment output functions. 0: PB6 (Initial value) 1: SEG62
13	S61MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P42 (Initial value) 1: SEG61
12	S60MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P67 (Initial value) 1: SEG60
11	S59MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P66 (Initial value) 1: SEG59
10	S58MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P65 (Initial value) 1: SEG58
9	S57MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P64 (Initial value) 1: SEG57
8	S56MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P63 (Initial value) 1: SEG56
7	S55MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P62 (Initial value) 1: SEG55
6	S54MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P61 (Initial value) 1: SEG54
5	S53MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P60 (Initial value) 1: SEG53

Bit No.	Bit symbol name	Description
4	S52MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P33 (Initial value) 1: SEG52
3	S51MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P32 (Initial value) 1: SEG51
2	S50MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P31 (Initial value) 1: SEG50
1	S49MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P30 (Initial value) 1: SEG49
0	S48MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P41 (Initial value) 1: SEG48

[Note]

- **Write the segment mode register in the display stop mode (LMD1 bit and LMD0 bit of DSPCON register is "0") to prevent erroneous displays or damage of panel.**

27.2.9 Segment Mode Register 4 (SEGMOD4)

SEGMOD4 is a special function register (SFR) to select the SEG64 functions.

Address: 0xF0FE
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	SEGMOD4															
Byte	-								SEGMOD4L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	S64MD
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 1	-	Reserved bits
0	S64MD	This bit is used to select the general-purpose input/output and segment output functions. 0: P77 (Initial value) 1: SEG64

[Note]

- **Write the segment mode register in the display stop mode (LMD1 bit and LMD0 bit of DSPCON register is "0") to prevent erroneous displays or damage of panel.**

27.2.10 Display Register00 to 64 (DSPR00 to DSPR64)

DSPR00 to DSPR64 are special function registers (SFR) to store display data.
As the initial value of these registers are undefined, set data to determine the contents of display.

Address: 0xF100(DSPRW00/DSPR00), 0xF101(DSPR01), 0xF102(DSPRW02/DSPR02), 0xF103(DSPR03), 0xF104(DSPRW04/DSPR04), 0xF105(DSPR05), 0xF106(DSPRW06/DSPR06), 0xF107(DSPR07), 0xF108(DSPRW08/DSPR08), 0xF109(DSPR09), 0xF10A(DSPRW10/DSPR10), 0xF10B(DSPR11), 0xF10C(DSPRW12/DSPR12), 0xF10D(DSPR13), 0xF10E(DSPRW14/DSPR14), 0xF10F(DSPR15), 0xF110(DSPRW16/DSPR16), 0xF111(DSPR17), 0xF112(DSPRW18/DSPR18), 0xF113(DSPR19), 0xF114(DSPRW20/DSPR20), 0xF115(DSPR21), 0xF116(DSPRW22/DSPR22), 0xF117(DSPR23), 0xF118(DSPRW24/DSPR24), 0xF119(DSPR25), 0xF11A(DSPRW26/DSPR26), 0xF11B(DSPR27), 0xF11C(DSPRW28/DSPR28), 0xF11D(DSPR29), 0xF11E(DSPRW30/DSPR30), 0xF11F(DSPR31), 0xF120(DSPRW32/DSPR32), 0xF121(DSPR33), 0xF122(DSPRW34/DSPR34), 0xF123(DSPR35), 0xF124(DSPRW36/DSPR36), 0xF125(DSPR37), 0xF126(DSPRW38/DSPR38), 0xF127(DSPR39), 0xF128(DSPRW40/DSPR40), 0xF129(DSPR41), 0xF12A(DSPRW42/DSPR42), 0xF12B(DSPR43), 0xF12C(DSPRW44/DSPR44), 0xF12D(DSPR45), 0xF12E(DSPRW46/DSPR46), 0xF12F(DSPR47), 0xF130(DSPRW48/DSPR48), 0xF131(DSPR49), 0xF132(DSPRW50/DSPR50), 0xF133(DSPR51), 0xF134(DSPRW52/DSPR52), 0xF135(DSPR53), 0xF136(DSPRW54/DSPR54), 0xF137(DSPR55), 0xF138(DSPRW56/DSPR56), 0xF139(DSPR57), 0xF13A(DSPRW58/DSPR58), 0xF13B(DSPR59), 0xF13C(DSPRW60/DSPR60), 0xF13D(DSPR61), 0xF13E(DSPRW62/DSPR62), 0xF13F(DSPR63), 0xF140(DSPRW64/DSPR64),
Initial value: R/W
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	DSPRWn															
Byte	DSPRm								DSPRn							
Bit	C7	C6	C5	C4	C3	C2	C1	C0	C7	C6	C5	C4	C3	C2	C1	C0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

m: odd number (ex. 01, 03, 05, ・・・63)
n: even number (ex. 00, 02, 04, ・・・64)

Bit No.	Bit symbol name	Description
15 to 8	C7 to C0	These bits are used to select drive off waveform or drive on waveform。 0: Drive Off waveform 1: Drive On waveform
7 to 0	C7 to C0	These bits are used to select drive off waveform or drive on waveform。 0: Drive Off waveform 1: Drive On waveform

Table 27-1 shows a list of registers available for each product.
"-" means the register does not exist and writing to register is invalid and "0" is read.

Table 27-1 List of Display Registers

●: Available -: Unavailable

Register name	Address	Corresponding segment	48pin product	52pin product	64pin product	80pin product	100pin product
			ML62Q1700/ ML62Q1701/ ML62Q1702/ ML62Q1703/ ML62Q1704	ML62Q1710/ ML62Q1711/ ML62Q1712/ ML62Q1713/ ML62Q1714	ML62Q1720/ ML62Q1721/ ML62Q1722/ ML62Q1723/ ML62Q1724/ ML62Q1725/ ML62Q1726/ ML62Q1727/ ML62Q1728/ ML62Q1729	ML62Q1733/ ML62Q1734/ ML62Q1735/ ML62Q1736/ ML62Q1737/ ML62Q1738/ ML62Q1739	ML62Q1743/ ML62Q1744/ ML62Q1745/ ML62Q1746/ ML62Q1747/ ML62Q1748/ ML62Q1749
DSPR00	0xF100	SEG0	●	●	●	●	●
DSPR01	0xF101	SEG1	●	●	●	●	●
DSPR02	0xF102	SEG2	●	●	●	●	●
DSPR03	0xF103	SEG3	●	●	●	●	●
DSPR04	0xF104	SEG4	●	●	●	●	●
DSPR05	0xF105	SEG5	●	●	●	●	●
DSPR06	0xF106	SEG6	-	●	●	●	●
DSPR07	0xF107	SEG7	-	-	●	●	●
DSPR08	0xF108	SEG8	-	-	●	●	●
DSPR09	0xF109	SEG9	-	-	-	-	●
DSPR10	0xF10A	SEG10	-	-	-	-	●
DSPR11	0xF10B	SEG11	-	-	-	-	●
DSPR12	0xF10C	SEG12	-	-	-	●	●
DSPR13	0xF10D	SEG13	-	-	-	●	●
DSPR14	0xF10E	SEG14	-	-	-	●	●
DSPR15	0xF10F	SEG15	-	-	-	●	●
DSPR16	0xF110	SEG16	-	-	-	-	●
DSPR17	0xF111	SEG17	-	-	-	-	●
DSPR18	0xF112	SEG18	-	-	-	-	●
DSPR19	0xF113	SEG19	-	-	-	-	●
DSPR20	0xF114	SEG20	-	-	●	●	●
DSPR21	0xF115	SEG21	-	-	●	●	●
DSPR22	0xF116	SEG22	●	●	●	●	●
DSPR23	0xF117	SEG23	●	●	●	●	●
DSPR24	0xF118	SEG24	●	●	●	●	●
DSPR25	0xF119	SEG25	●	●	●	●	●
DSPR26	0xF11A	SEG26	●	●	●	●	●
DSPR27	0xF11B	SEG27	●	●	●	●	●
DSPR28	0xF11C	SEG28	●	●	●	●	●
DSPR29	0xF11D	SEG29	●	●	●	●	●
DSPR30	0xF11E	SEG30	●	●	●	●	●
DSPR31	0xF11F	SEG31	●	●	●	●	●
DSPR32	0xF120	SEG32	●	●	●	●	●
DSPR33	0xF121	SEG33	●	●	●	●	●
DSPR34	0xF122	SEG34	-	●	●	●	●
DSPR35	0xF123	SEG35	-	-	●	●	●
DSPR36	0xF124	SEG36	-	-	-	●	●
DSPR37	0xF125	SEG37	-	-	-	●	●
DSPR38	0xF126	SEG38	-	-	-	-	●

Register name	Address	Corresponding segment	48pin product	52pin product	64pin product	80pin product	100pin product
			ML62Q1700/ ML62Q1701/ ML62Q1702/ ML62Q1703/ ML62Q1704	ML62Q1710/ ML62Q1711/ ML62Q1712/ ML62Q1713/ ML62Q1714	ML62Q1720/ ML62Q1721/ ML62Q1722/ ML62Q1723/ ML62Q1724/ ML62Q1725/ ML62Q1726/ ML62Q1727/ ML62Q1728/ ML62Q1729	ML62Q1733/ ML62Q1734/ ML62Q1735/ ML62Q1736/ ML62Q1737/ ML62Q1738/ ML62Q1739	ML62Q1743/ ML62Q1744/ ML62Q1745/ ML62Q1746/ ML62Q1747/ ML62Q1748/ ML62Q1749
DSPR39	0xF127	SEG39	-	-	-	-	●
DSPR40	0xF128	SEG40	-	-	-	-	●
DSPR41	0xF129	SEG41	-	-	-	-	●
DSPR42	0xF12A	SEG42	-	-	-	-	●
DSPR43	0xF12B	SEG43	-	-	-	●	●
DSPR44	0xF12C	SEG44	-	-	-	●	●
DSPR45	0xF12D	SEG45	-	-	-	●	●
DSPR46	0xF12E	SEG46	-	-	-	●	●
DSPR47	0xF12F	SEG47	-	-	●	●	●
DSPR48	0xF130	SEG48	-	●	●	●	●
DSPR49	0xF131	SEG49	●	●	●	●	●
DSPR50	0xF132	SEG50	●	●	●	●	●
DSPR51	0xF133	SEG51	●	●	●	●	●
DSPR52	0xF134	SEG52	●	●	●	●	●
DSPR53	0xF135	SEG53	●	●	●	●	●
DSPR54	0xF136	SEG54	●	●	●	●	●
DSPR55	0xF137	SEG55	●	●	●	●	●
DSPR56	0xF138	SEG56	●	●	●	●	●
DSPR57	0xF139	SEG57	●	●	●	●	●
DSPR58	0xF13A	SEG58	●	●	●	●	●
DSPR59	0xF13B	SEG59	●	●	●	●	●
DSPR60	0xF13C	SEG60	-	-	●	●	●
DSPR61	0xF13D	SEG61	-	-	●	●	●
DSPR62	0xF13E	SEG62	-	-	-	-	●
DSPR63	0xF13F	SEG63	-	-	-	-	●
DSPR64	0xF140	SEG64	-	-	-	-	●

27.3 Description of Operation

27.3.1 Operation of LCD Driver Circuit

Figure 27-3 shows the LCD driver circuit operation.

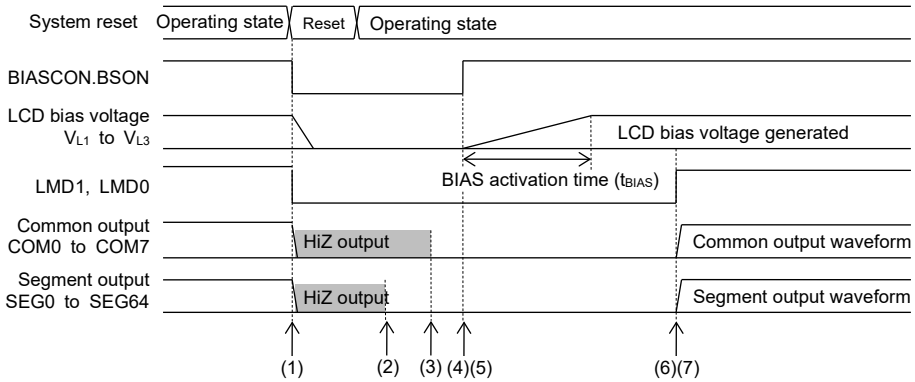


Figure 27-3 LCD Driver Circuit Operation

① A system reset stops the operation of bias generation circuit and LCD driver and the common output and segment output pins get high impedance that both input and output is disabled state.
② Select segments by setting the segment mode register0 to 4. The output pins of selected segment make a Vss level output.
③ Set the frame frequency and duty by the display mode register (DSPMOD). The common output pins corresponding to the selected duties make a Vss level output.
④ Set the bias generation circuit operation mode by the bias control register (BIASCON).
⑤ To use a display type other than external supply mode, turn on the bias generation circuit by setting BSON bit of the bias control register (BIASCON) to "1".
⑥ Set display data in the display registers00 to 64 (DSPR00 to DSPR64).
⑦ Wait for the bias activation time (t_{BIAS}) or longer. Then set to the display mode using the LMD1 and LMD0 bits of the display control register (DSPCON). (The display waveform is output to common output and segment output pins.)
For the bias activation time (t_{BIAS}), refer to the electrical characteristics in the data sheet.

27.3.2 Display Register Segment Map

Figure 27-4 shows the segment map configuration of the display registers00 to 64 (DSPR00 to DSPR64).

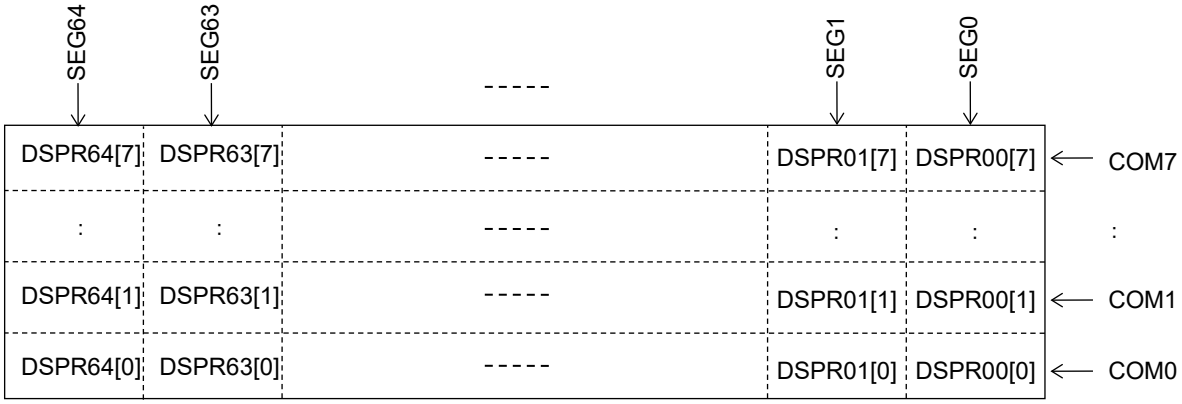


Figure 27-4 Display Register Segment Map Configuration

27.3.3 Common Output Waveform

Figure 27-5 shows the waveform A, the output from the common pins in 1/3 duty mode.

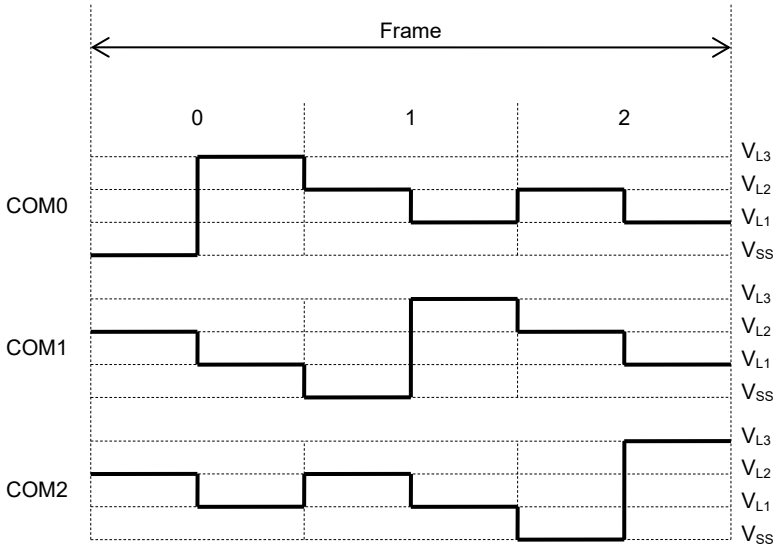


Figure 27-5 Waveform A, the output from Common Pins in 1/3 Duty

Figure 27-6 shows the waveform A, the output from the common pins in 1/4 duty mode.

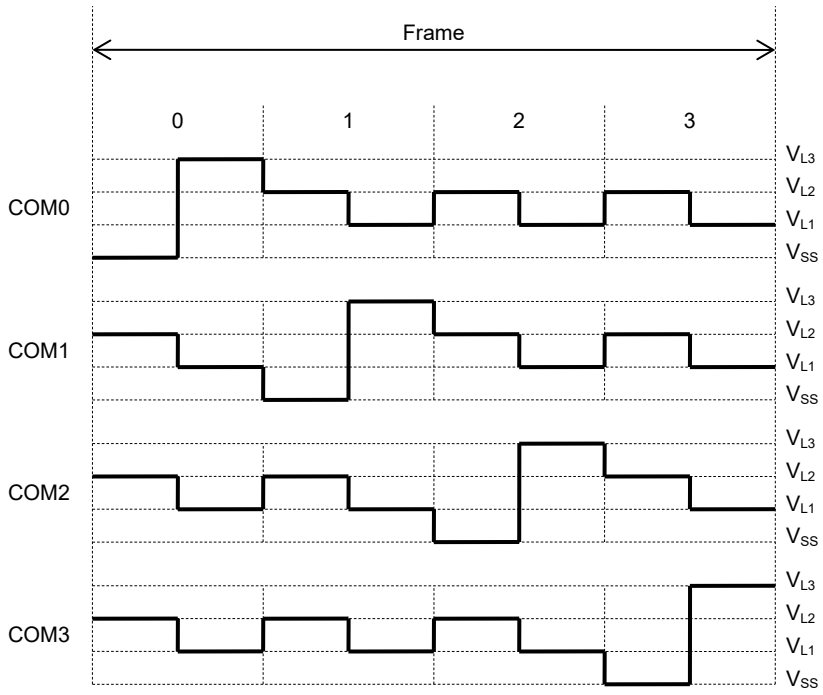


Figure 27-6 Waveform A, the output from Common Pins in 1/4 Duty

Figure 27-7 shows the waveform B, the output from the common pins in 1/3 duty mode.

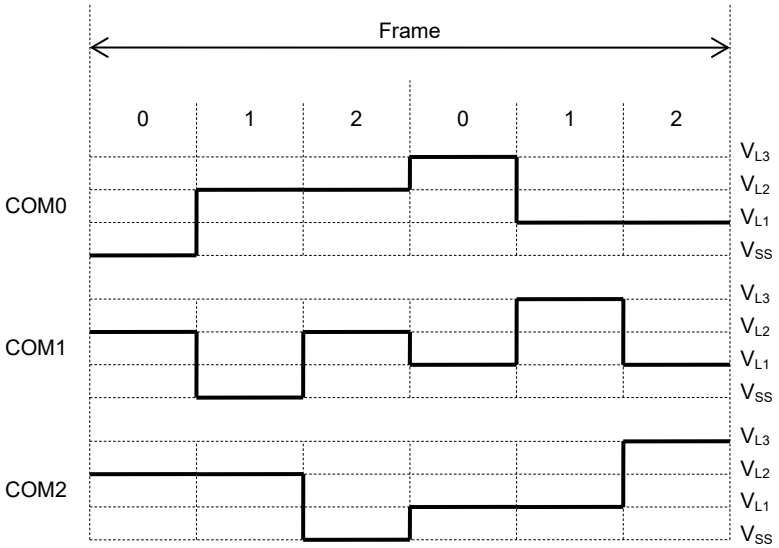


Figure 27-7 Waveform B, the output from Common Pins in 1/3 Duty

Figure 27-8 shows the waveform B, the output from the common pins in 1/4 duty mode.

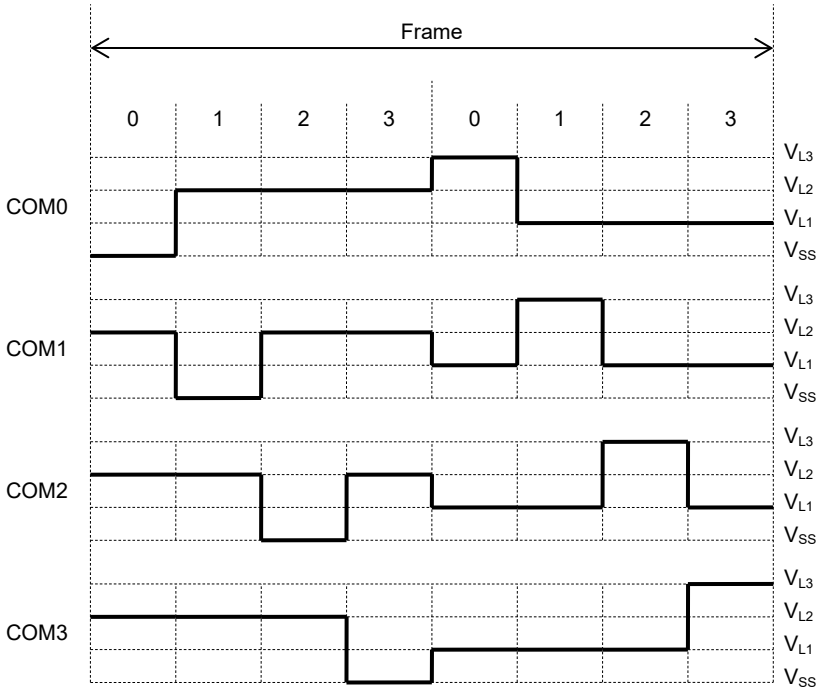


Figure 27-8 Waveform B, the output from Common Pins in 1/4 Duty

27.3.4 Segment Output Waveform

Figure 27-9 shows the waveform A, the output from the segment pins in 1/3 duty mode.

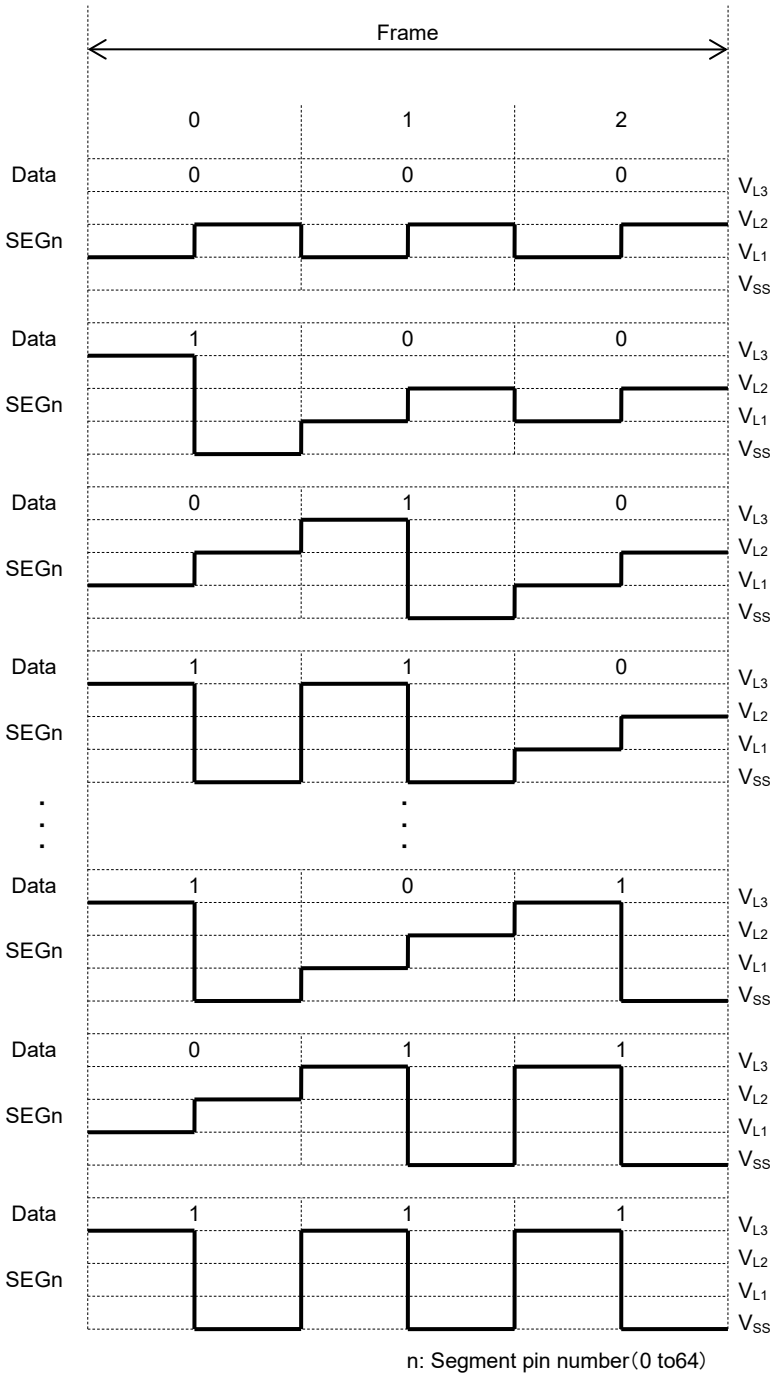


Figure 27-9 Waveform A, output from Segment Pins in 1/3 Duty

Figure 27-10 shows the waveform A, the output from the segment pins in 1/4 duty mode.

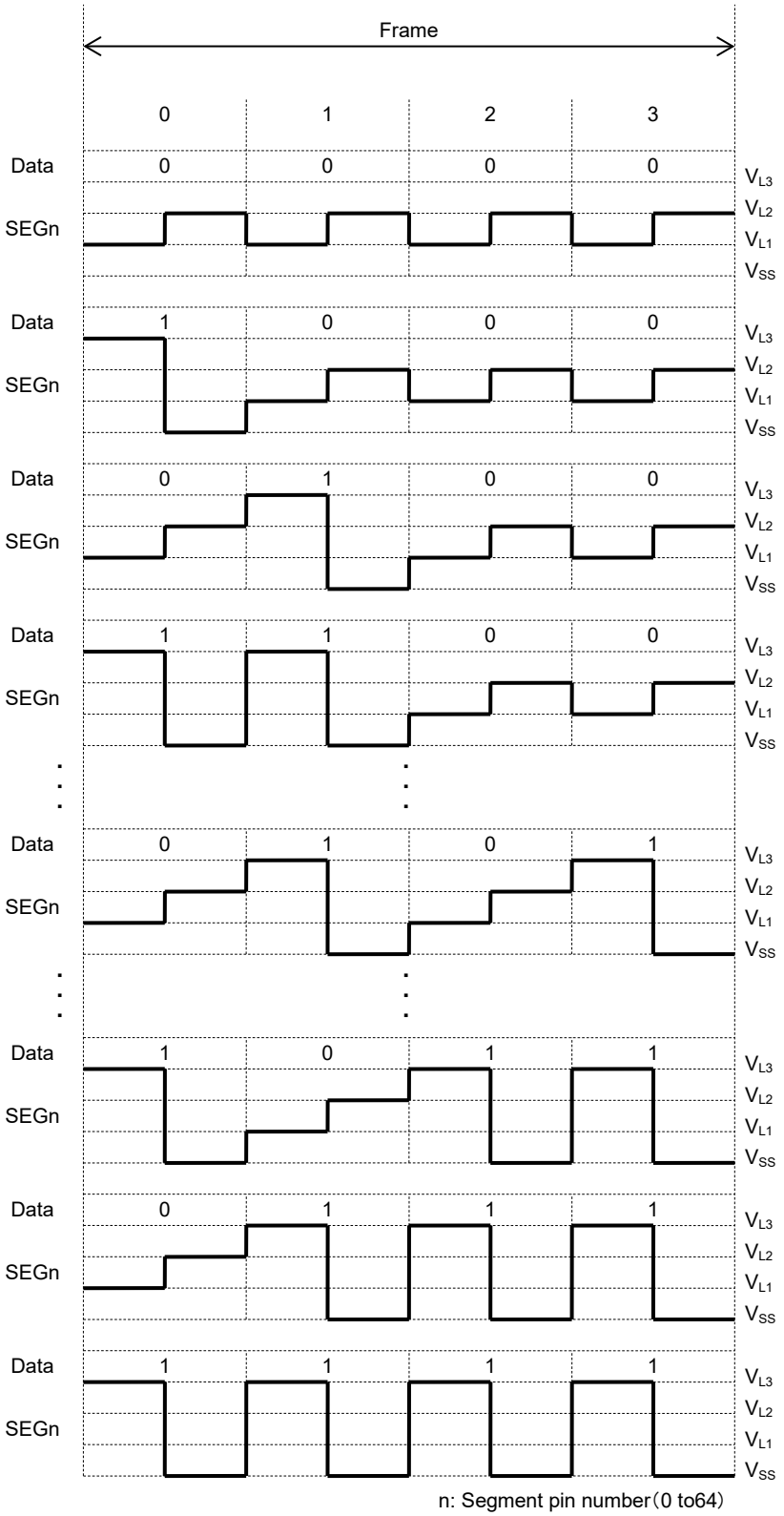


Figure 27-10 Waveform A, the output from Segment Pins in 1/4 Duty

Figure 27-11 shows the waveform B, the output from the segment pins in 1/3 duty mode.

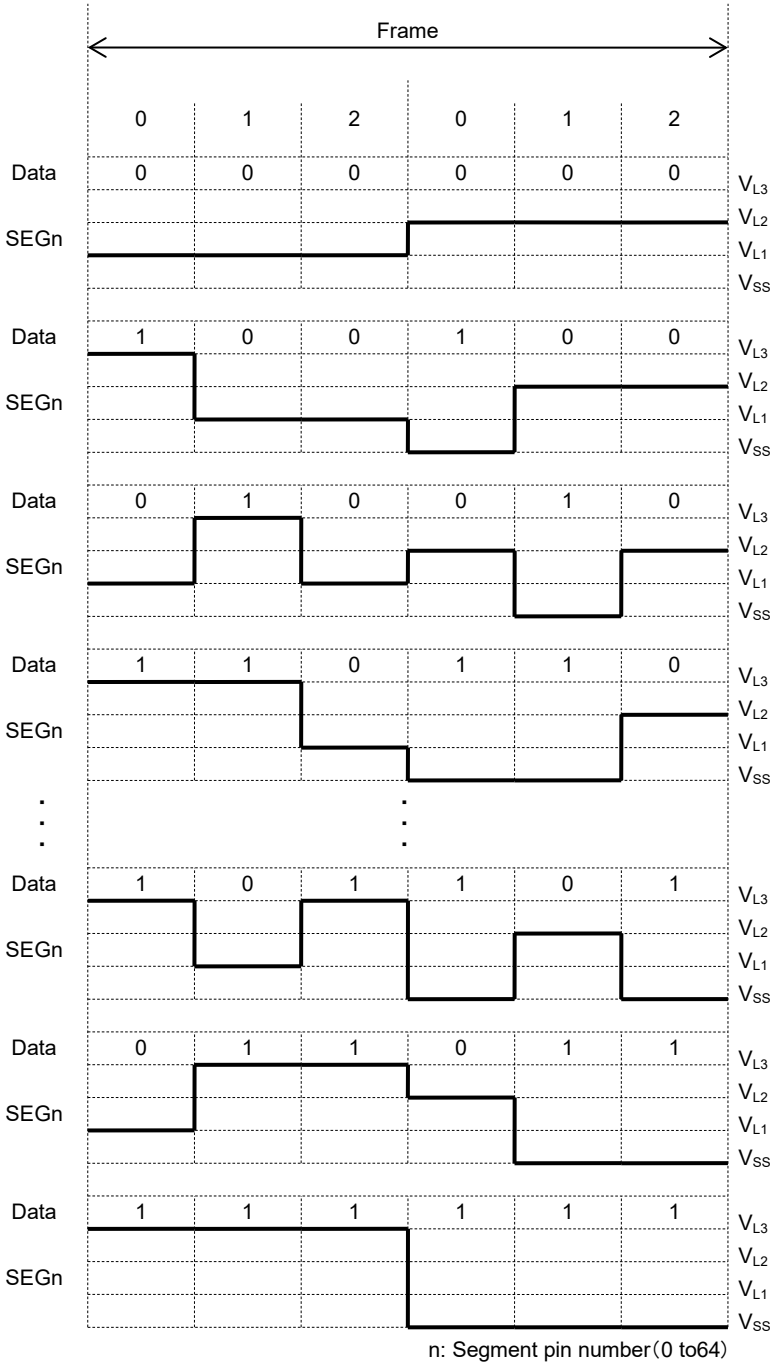


Figure 27-11 Waveform B, the output from Segment Pins in 1/3 Duty

Figure 27-12 shows the waveform B, the output from the segment pins in 1/4 duty mode.

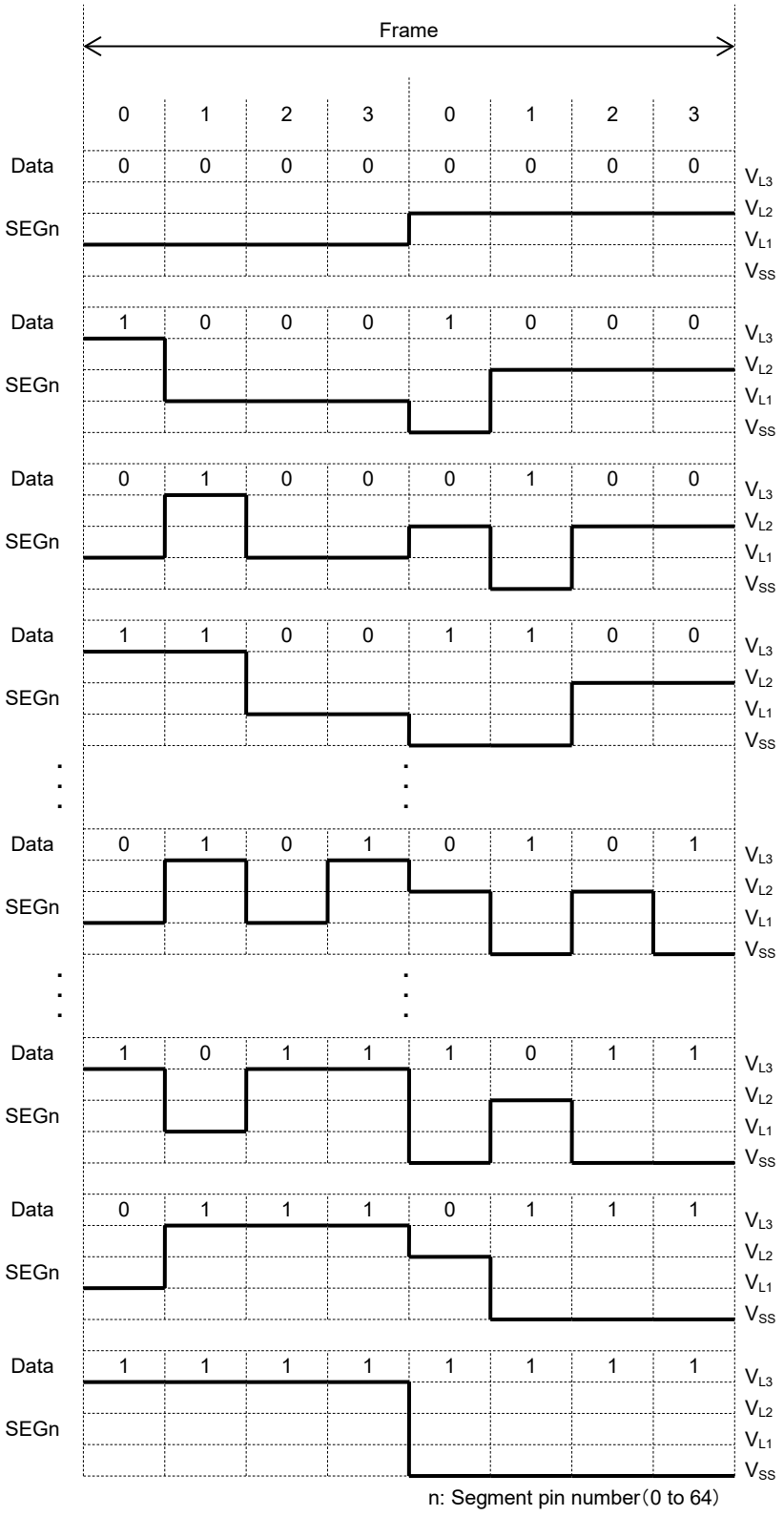


Figure 27-12 Waveform B, the output from Segment Pins in 1/4 Duty

27.3.5 Common Output Waveform for LED drive

Figure 27-13 shows the waveform A, the output from the common pins in 1/3 duty and LED drive mode.

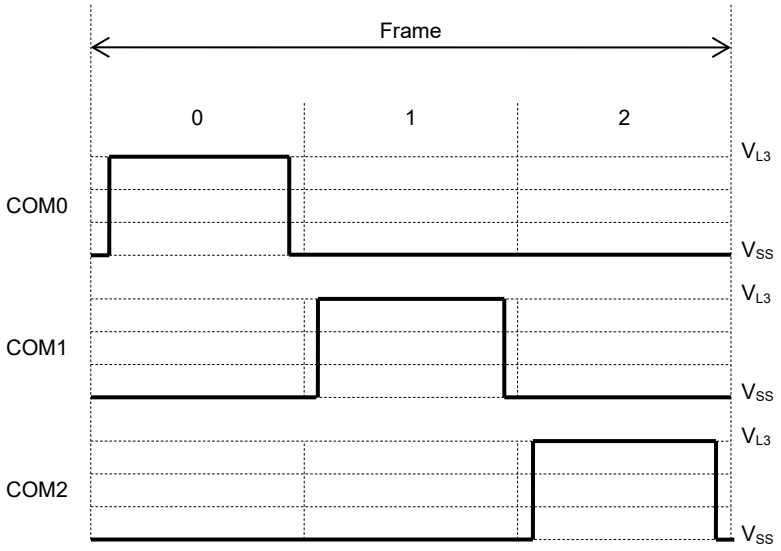


Figure 27-13 Waveform A, output from Common Pins in 1/3 Duty and LED drive mode

Figure 27-14 shows the waveform A, the output from the common pins in 1/4 duty and LED drive mode.

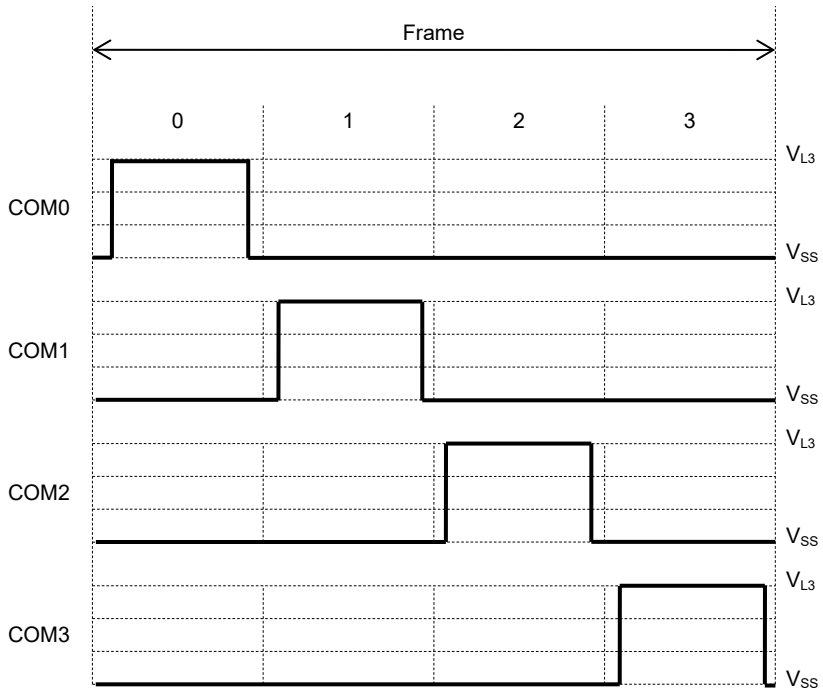


Figure 27-14 Waveform A, the output from Common Pins in 1/4 Duty and LED drive mode

27.3.6 Segment Output Waveform for LED drive

Figure 27-15 shows the waveform A, the output from the segment pins in 1/3 duty and LED drive mode.

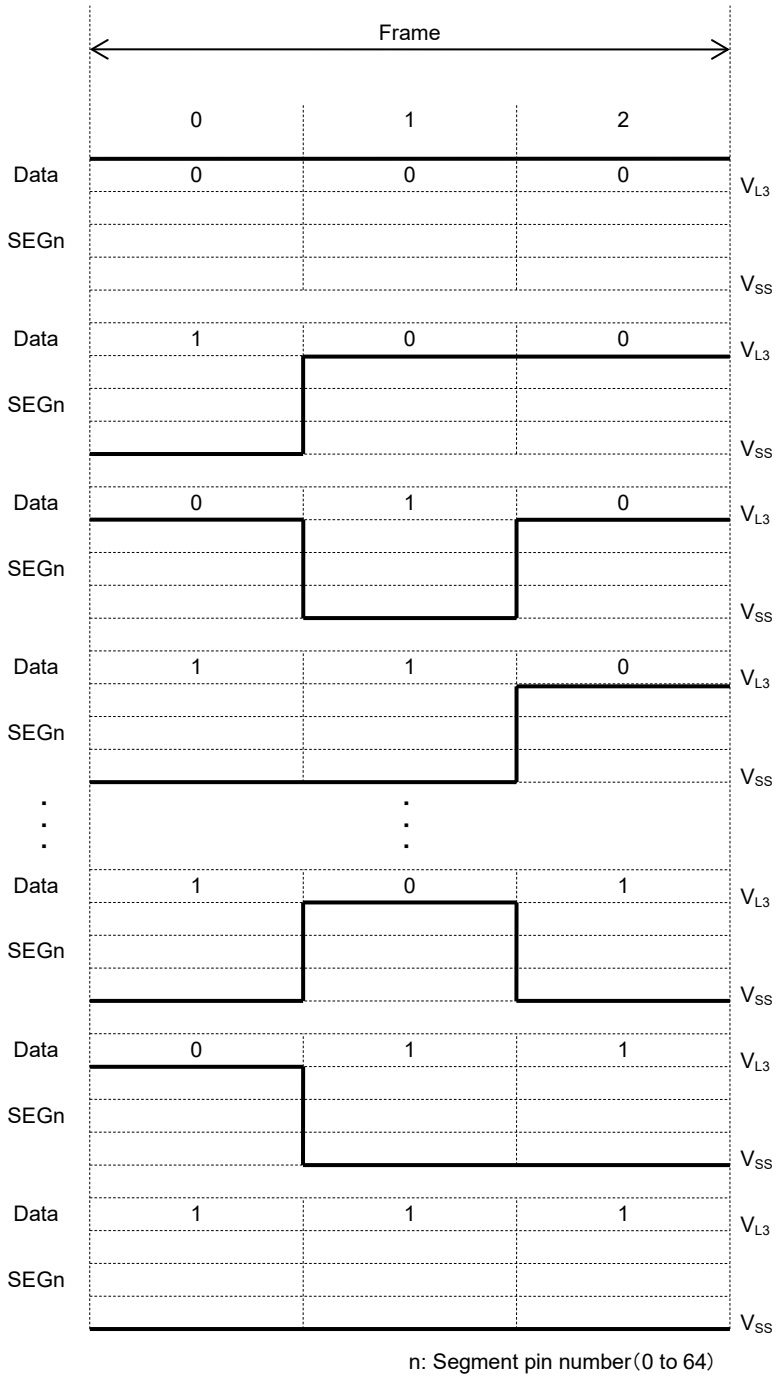


Figure 27-15 Waveform A, the output from Segment Pins in 1/3 Duty and LED drive mode

Figure 27-16 shows the waveform A, the output from the segment pins in 1/4 duty and LED drive mode.

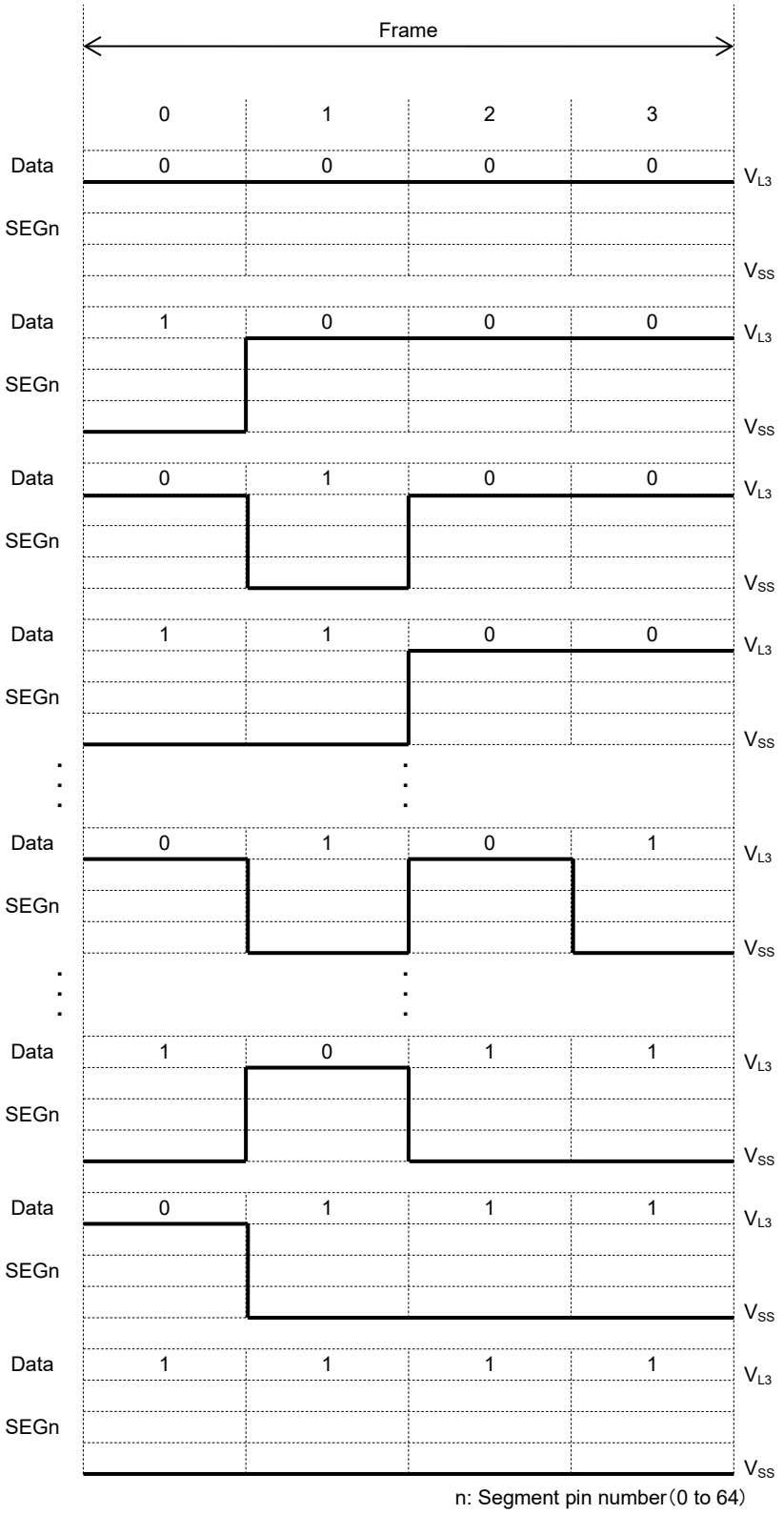


Figure 27-16 Waveform A, output from Segment Pins in 1/4 Duty and LED drive mode

27.3.7 Configuration Example to drive LEDs

It is necessary MOS transistors for driving LEDs as external component. Figure 27-17 shows the configuration example in LED drive mode.

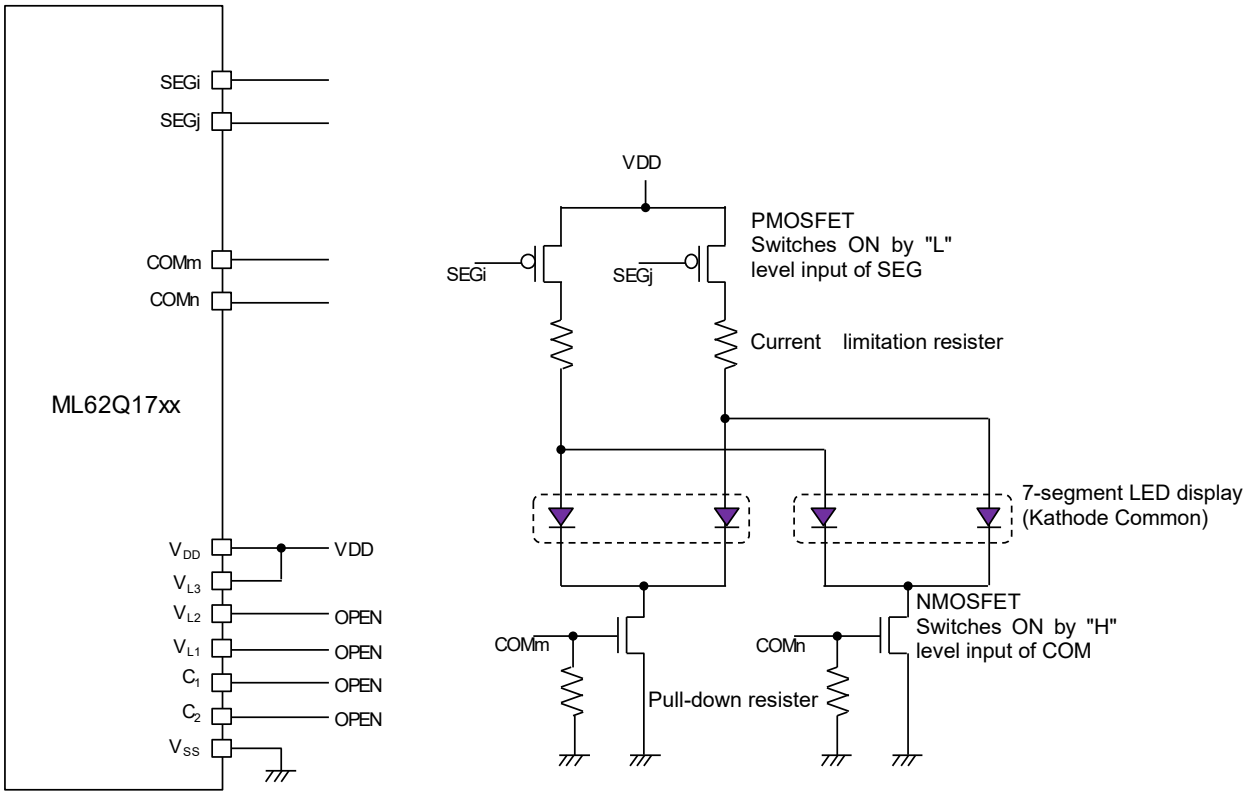


Figure 27-17 Configuration example in LED drive mode

Chapter 28 On-Chip Debug Function

28. On-Chip Debug Function

28.1 General Description

This function is used by connecting the host PC and LSI through the on-chip debug emulator (hereafter referred to as "On-chip emulator").
On-board debugging or programming is available by using the program development environment software (debugger) installed on the host PC.

28.1.1 Features

- The following debug functions are provided using the debugger by connecting LSI and On-chip emulator
  - Emulation
    - Real time emulation
    - Single step emulation
  - Break
    - Hardware break point break (four points)
    - RAM data matching break
    - Sequential break
    - Trace overflow break
    - Stack overflow/underflow break
    - Unused ROM area access break
    - RAM parity error break
  - Trace
    - Branch trace
  - Real time watch
  - CPU resource display/change
    - Program memory reference/disassembly
    - RAM and SFR display/change
    - Register display/change in the CPU
  - Program download
    - Program download/read/erase to/from flash memory
    - Data write/read/erase to/from data flash
  - Peripheral circuit operation continue/stop control during break

    Target peripheral circuits
    - External interrupt
    - Low-speed time base counter
    - 16-bit timer
    - Functional timer
    - Serial communication unit (synchronous serial port/UART)
    - I^2C bus master
    - I^2C bus unit (master/slave)
    - DMA controller
    - Buzzer
    - Analog module

      (Analog comparator, successive approximation type A/D converter, voltage level supervisor (VLS))

- The following program download function is provided using the flash multi-writer by connecting LSI and On-chip emulator.
  - Program download
    - Erasing/Programming the program memory space
    - Erasing/Programming the data flash memory area

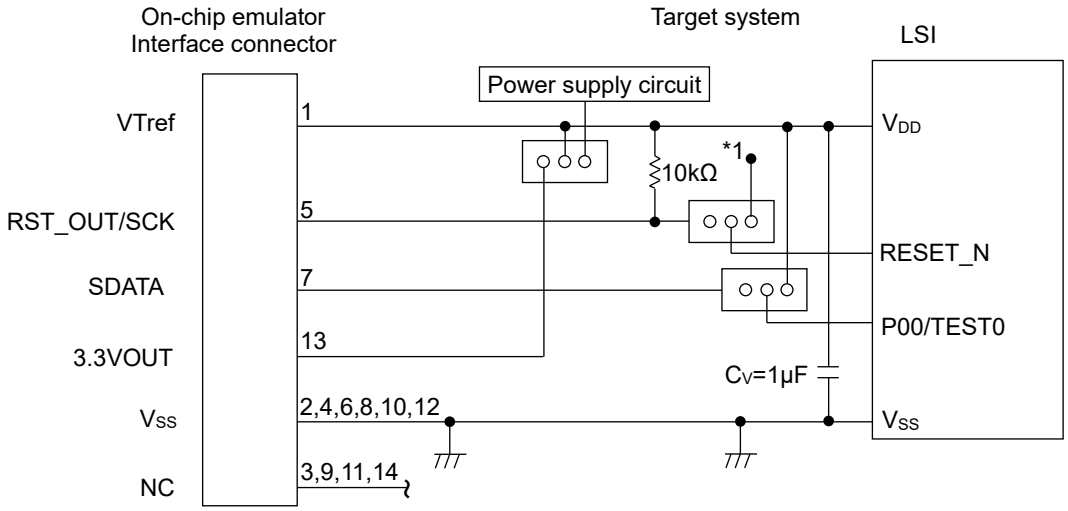
28.1.2 Configuration

When using the on-chip debug function, two methods are available for power supply to LSI as described below:

- Use the 3.3 VOUT power supply (+3.3 V/100 mA) of On-chip emulator
- Use the power supply of the target system (V_{DD}=1.6 V to 5.5 V)

28.1.2.1 Using 3.3 VOUT Power Supply (+3.3 V/100 mA) of On-chip Emulator

Figure 28-1 shows a connection example when using the 3.3 VOUT power supply (+3.3 V/100 mA) of On-chip emulator.

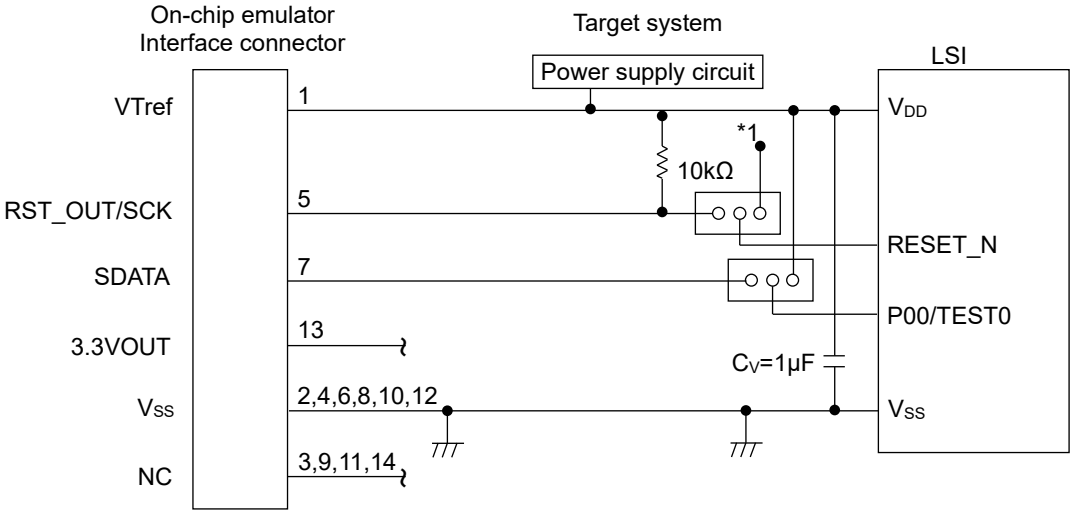


*1) Normal operation (reset IC, V_{DD}, etc.)

Figure 28-1 Connection Example When Using On-chip Emulator 3.3 VOUT Power Supply

28.1.2.2 Using Power Supply of Target System (V_{DD}=1.6 V to 5.5 V)

Figure 28-2 shows a connection example when using the power supply (V_{DD}=1.6 V to 5.5 V) of the target system.



*1) Normal operation (reset IC, V_{DD}, etc.)

Figure 28-2 Connection Example When Using Target System Power Supply

28.1.3 List of Pins

The following pins are used for the on-chip debug function.

Signal name	I/O	Function
RESET_N	I	Reset input
P00/TEST0	I/O	On-chip debug function signal input/output

28.2 How to Use On-chip Debug Function

See manual of the debugger for how to use the on-chip debug function using On-chip emulator and the debugger.
See manual of the flash multi-writer for how to download a program using On-chip emulator and flash multi-writer.

28.3 Precautions

[Note] on usage of the on-chip debug function.

- **Make RESET_N pin able to be connected to V_{DD} with a jumper or something when not using the on-chip debug function.**
- **Make P00/TEST0 pin able to be connected to V_{DD} with a jumper or something when not using the on-chip debug function.**
- **Do not program instruction codes into the LSI that set the P00/TEST0 pin to the output mode. If P00/TEST0 is set to the output mode before On-chip emulator performs read/write to/from the target chip, communication with On-chip emulator after that will be disabled. Also note that the input/output mode of P00/TEST0 is uninitialized by On-chip emulator.**
- **Validate the ROM code on user production board without the On-chip emulator.**
- **Disconnect On-chip emulator when measuring the current consumption of the target system. If On-chip emulator remains connected, the current consumption increases as the on-chip debug circuit inside the LSI works for the communication.**
- **When using the 3.3 VOUT power supply of On-chip emulator, do not apply power of the target system to the V_{DD} pin of LSI. If both power supplies are connected, On-chip emulator may be damaged, or an electric shock or fire may occur.**
- **LSI used to debug a program is not covered by the product warranty. Do not use the LSI for mass-production.**
- **A reset due to unused ROM area access does not occur in the on-chip debug mode regardless of code option settings.**
- **A RAM parity error reset does not occur in the on-chip debug mode and the break operation occurs instead.**
- **RAM parity error may occur even if the RAM area is not displayed in case the contents of the data memory are displayed in the debugger in a state where a RAM parity error may occur (including uninitialized RAM).**
- **The all interrupts and watchdog timer operation always stop while the debugger is in the break state.**
- **On-chip emulator might be affected by the external environments such as the host PC, USB cable, On-chip emulator interface cable and the target system. Please confirm proper environments before using on-chip emulator.**
- **If adding an external capacitor to the RESET_N pin, prepare a jumper function on the board so that the capacitor gets disconnectable when using the debugger or Flash multi-writer.**

28.4 Operation of Peripheral Circuits during breaks in the on-chip debug mode

The debugger allows users to choose whether to continue or stop operating the peripheral circuits during the break state on the debugger.
Table 28-1 shows the optional items, the target peripherals and how the operation is controlled.
Each optional item is displayed with a check box on the debugger. See manual of the debugger for more details on how to use the function.

Table 28-1 Peripheral controls during the break on the debugger

Optional item	Peripheral Circuit	Description
External Interrupt	External Interrupt	If the item is checked on, the target LSI accepts the external input during the break. If the item checked off, the target LSI does not accept the external input during the break. 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 (SSIO/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	

28.5 Reset in the On-Chip Debug Tool

By executing reset from the debug tool, RSTAT register POR bit is set to "1". However, low-speed crystal oscillation, RTC, and VLS functions are not reset. If it is necessary to start them from initial state, set these pertinent SFRs to initial value. Then execute reset from debug tools.

Chapter 29 Safety Function

29. Safety Function

29.1 General Description

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 Class B.

29.1.1 Features

- Safety Functions on the LSI

Function Name	Description	Control by SFR
RAM guard	Protect from the miss-writing to the specified RAM area	Available
SFR guard	Protect from the miss-writing to the specified SFR	Available
Successive approximation type A/D converter test	Successive approximation type AD converter test function	Available
RAM parity error detection	RAM parity error check and generates a reset on error (enable/disable reset by SFR, with reset status flag and parity error flag)	Available
ROM unused area access reset	Make a reset in case the CPU executes an instruction in the unused area (enable/disable reset by the code option, with reset status flag)	-
Clock mutual monitoring	Monitor to check whether the oscillation of the high-speed and low-speed clocks are normal	Available
CRC calculation	Detect data error in the flash memory or data error in communications	Available
UART self-test function	Make the UART self-test	Available
SSIO self-test function	Make the SSIO self-test	Available
I^2C self-test function	Make the I^2C self-test function	Available
WDT counter read	WDT counter read function	Available
Port output level self-test function	General port self-test function	Available
Clock backup function and the self-test	Switch automatically to the low-speed RC oscillation in case the low-speed crystal oscillation stopped	Available
MCU status interrupt	Control interrupts generated by RAM parity error, automatic CRC calculation completion, and data flash erase/program completion.	Available

29.2 Description of Registers

29.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF0B0	RAM Guard Setting Register 0	RAMGD	-	R/W	8	0x00
0xF0B1 to 0xF0B3	Reserved	-	-	-	-	-
0xF0B4	SFR Guard Setting Register 0	SFRGD0L	SFRGD0	R/W	8/16	0x00
0xF0B5		SFRGD0H		R/W	8	0x00
0xF0B6	SFR Guard Setting Register 1	SFRGD1L	SFRGD1	R/W	8/16	0x00
0xF0B7		SFRGD1H		R/W	8	0x00
0xF0B8 0xF0B9 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
0xF050	MCU Status Interrupt Enable Register	MCINTEL	-	R/W	8	0x00
0xF051	Reserved	-	-	-	-	-
0xF052	MCU Status Interrupt Register	MCISTATL	-	R	8	0x00
0xF053	Reserved	-	-	-	-	-
0xF054	MCU Status Interrupt Clear Register	MCINTCLL	MCINTCL	W	8/16	0x00
0xF055		MCINTCLH		W	8	0x00

29.2.2 RAM Guard Setting Register (RAMGD)

RAMGD is a special function register (SFR) used to disable writing the RAM by the CPU and the DMA Controller. Data in the specified RAM area is protectable.

Address: 0xF0B0 (RAMGD)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								RAMGD							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	rsvd	RGD2	RGD1	RGD0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No,	Bit symbol name	Description
7 to 4	-	Reserved bits
3	rsvd	Reserved bit: write “0” to this.
2 to 0	RGD2 to RGD0	These bits are used to choose a protect area for writing on the RAM. 000: All RAM area writable and readable (Initial value) 001: 0x0:0EFC0 to 0x0:0EFFF (64 byte) is unwritable and readable 010: 0x0:0EF80 to 0x0:0EFFF (128 byte) is unwritable and readable 011: 0x0:0EF00 to 0x0:0EFFF (256 byte) is unwritable and readable 100: 0x0:0EE00 to 0x0:0EFFF (512 byte) is unwritable and readable 101: Do not use (0x0:0EE00 to 0x0:0EFFF (512 byte) is unwritable and readable) 110: Do not use (0x0:0EE00 to 0x0:0EFFF (512 byte) is unwritable and readable) 111: Do not use (0x0:0EE00 to 0x0:0EFFF (512 byte) is unwritable and readable)

29.2.3 SFR Guard Setting Register 0 (SFRGD0)

SFRGD0 is a special function register (SFR) used to disable writing the SFR by the CPU and the DMA Controller. Data in the specified SFR area is protectable.

Address: 0xF0B4 (SFRGD0L/SFRGD0), 0xF0B5(SFRGD0H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SFRGD0															
Byte	SFRGD0H								SFRGD0L							
Bit	-	-	-	-	-	-	-	-	rsvd	rsvd	SGD05	SGD04	SGD03	SGD02	SGD01	SGD00
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bits
7 to 6	rsvd	Reserved bits: write “0” to them.
5	SGD05	This bit is used to disable WDTMOD register. See Chapter 10 "Watch Dog Timer" for details of the register. 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
4	SGD04	This bit is used to disable BCKCONn register and BRECONn register (n=0 to 3). See Chapter 4 "Power management" for details of the registers. 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
3	SGD03	This bit is used to disable RASFMOD register. See Section 29.2.5 "RAM Parity Setting Register (RASFMOD) "for details of the register. 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
2	SGD02	This bit is used to disable SFRs described in Chapter 22 "Voltage Level Supervisor (VLS)". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
1	SGD01	This bit is used to disable SFRs described in Chapter 6 "Clock Generation Circuit". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
0	SGD00	This bit is used to disable SFRs described in Chapter 5 "Interrupt". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable

29.2.4 SFR Guard Setting Register 1 (SFRGD1)

SFRGD1 is a special function register (SFR) used to disable writing the SFR by the CPU and the DMA Controller. Data in the specified SFR area is protectable.

Address: 0xF0B6(SFRGD1L/SFRGD1), 0xF0B7(SFRGD1H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SFRGD1															
Byte	SFRGD1H								SFRGD1L							
Bit	SGD1F	-	-	-	SGD1B	SGD1A	SGD19	SGD18	SGD17	SGD16	SGD15	SGD14	SGD13	SGD12	SGD11	SGD10
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	SGD1F	This bit is used to disable SFRs related to the port XT described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
14 to 12	-	Reserved bits
11	SGD1B	This bit is used to disable SFRs related to the port B described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
10	SGD1A	This bit is used to disable SFRs related to the port A described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
9	SGD19	This bit is used to disable SFRs related to the port 9 described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
8	SGD18	This bit is used to disable SFRs related to the port 8 described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
7	SGD17	This bit is used to disable SFRs related to the port 7 described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
6	SGD16	This bit is used to disable SFRs related to the port 6 described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
5	SGD15	This bit is used to disable SFRs related to the port 5 described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable

Bit No.	Bit symbol name	Description
4	SGD14	This bit is used to disable SFRs related to the port 4 described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
3	SGD13	This bit is used to disable SFRs related to the port 3 described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
2	SGD12	This bit is used to disable SFRs related to the port 2 described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
1	SGD11	This bit is used to disable SFRs related to the port 1 described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable
0	SGD10	This bit is used to disable SFRs related to the port 0 described in Chapter 17 "General Purpose Port". 0: The SFR is writable and readable (Initial value) 1: The SFR is unwritable and readable

29.2.5 RAM Parity Setting Register (RASFMOD)

RASFMOD is a special function register (SFR) used to control the RAM parity error reset function.
The RAM parity error is detectable and the RAM parity error reset is generatable.
The reset flag by a RAM parity error can be checked by the reset status register (SRSTAT).
See Chapter 3 "Reset Function"for details about the reset flag.

Address: 0xF0BC(RASFMOD)
Access: R/W
Access size: 8bit
Initial value: 0x00

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

Bit No.	Bit symbol name	Description
7	PERF	This bit is used to check whether the RAM parity error occurs. Write "1" to this bit to clear. When PEREN is set to "1" to enable the parity error reset function, the reset status register (SRSTAT) can be used to check. 0: RAM parity error reset function is disabled (Initial value) 1: RAM parity error reset function is enabled
6 to 1	-	Reserved bits
0	PEREN	This bit is used to enable/disable the RAM parity error reset function. 0: RAM parity error reset function is disabled (Initial value) 1: RAM parity error reset function is enabled

[Note]

- **If reading the RAM data without initializing the RAM, a parity error may occur. When using parity errors (checking parity error flags or enabling parity error reset), initialize all areas of the RAM before start using.**

29.2.6 Communication Test Setting Register (COMFT0)

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. See Section 29.3.1 "Communication Function Self Test" for more details.
As the I^2C bus unit and the I^2C master are equipped with the function to read the transmit data, the function can be used for testing. For details, see Chapter 12 "I2C Buss Unit" and Chapter 13" I2C Master".

Address: 0xF0BE(COMFT0/COMFT0L), 0xF0BF(COMFT0H)
Access: R/W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	COMFT0															
Byte	COMFT0H								COMFT0L							
Bit	-	-	-	rsvd	-	-	-	rsvd	-	-	CMFT05	CMFT04	CMFT03	CMFT02	CMFT01	CMFT00
R/W	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 13	-	Reserved bits
12		Reserved bit: write “0” to this.
11 to 9		Reserved bits
8		Reserved bit: write “0” to this.
7 to 6		Reserved bits
5	CMFT05	This bit is used to enable/disable the self-test for the serial communication unit 5. 0: The self-test for the communication function is disabled (Initial value) 1: The self-test for the communication function is enabled
4	CMFT04	This bit is used to enable/disable the self-test for the serial communication unit 4. 0: The self-test for the communication function is disabled (Initial value) 1: The self-test for the communication function is enabled
3	CMFT03	This bit is used to enable/disable the self-test for the serial communication unit 3. 0: The self-test for the communication function is disabled (Initial value) 1: The self-test for the communication function is enabled
2	CMFT02	This bit is used to enable/disable the self-test for the serial communication unit 2. 0: The self-test for the communication function is disabled (Initial value) 1: The self-test for the communication function is enabled
1	CMFT01	This bit is used to enable/disable the self-test for the serial communication unit 1. 0: The self-test for the communication function is disabled (Initial value) 1: The self-test for the communication function is enabled
0	CMFT00	This bit is used to enable/disable the self-test for the serial communication unit 0. 0: The self-test for the communication function is disabled (Initial value) 1: The self-test for the communication function is enabled

29.2.7 MCU Status Interrupt Enable Register (MCINTEL)

MCINTEL is a special function register (SFR) used to control enabling/disabling three types of interrupt status on the microcontroller.

Address: 0xF050 (MCINTEL)
Access: R/W
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								MCINTEL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	MCI2E	MCI1E	MCI0E
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 3	-	Reserved bits
2	MCI2E	This bit is used to enable/disable the interrupt at the completion of data flash erasing/programming. 0: The data flash erasing/programming completion interrupt is disabled (Initial value) 1: The data flash erasing/programming completion interrupt is enabled
1	MCI1E	This bit is used to enable/disable the interrupt at the completion of automatic CRC calculation. 0: The automatic CRC calculation completion interrupt is disabled (Initial value) 1: The automatic CRC calculation completion interrupt is enabled
0	MCI0E	This bit is used to enable/disable the interrupt at the occurrence of RAM parity error. 0: The RAM parity error occurrence interrupt is disabled (Initial value) 1: The RAM parity error occurrence interrupt is enabled

29.2.8 MCU Status Interrupt Register (MCISTATL)

MCISTATL is a read-only special function register (SFR) used to indicate status of the three types of interrupts.
The MCI2S bit to MCI0S bit is initialized, in addition to reset function, by writing "1" to the same number of bit in the MCINTCL register.

Address: 0xF052 (MCISTATL)
Access: R
Access size: 8bit
Initial value: 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								MCISTATL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	MCI2S	MCI1S	MCI0S
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 3	-	Reserved bits
2	MCI2S	This bit is used to indicate status of the data flash erasing/programming completion interrupt. 0: The data flash erasing/programming completion interrupt has not been generated (Initial value) 1: The data flash erasing/programming completion interrupt has been generated
1	MCI1S	This bit is used to indicate status of the automatic CRC calculation completion interrupt. 0: The automatic CRC calculation completion interrupt has not been generated (Initial value) 1: The automatic CRC calculation completion interrupt has been generated
0	MCI0S	This bit is used to indicate status of the RAM parity error occurrence interrupt. 0: The RAM parity error occurrence interrupt has not been generated (Initial value) 1: The RAM parity error occurrence interrupt has been generated

[Note]

- **No interrupt request is issued if the interrupt status bit is "1" and the same interrupt occurs again. To issue an interrupt request, write "1" to the same bit in the MCINTCL register and clear the status bit to "0".**

29.2.9 MCU Status Interrupt Clear Register (MCINTCL)

MCINTCL is a write-only special function register (SFR) used to clear the MCU status interrupts.
If the MCI2C bit to MCI0C bit is set to "1", the interrupt request indicated by the same number of bit in the MCISTATL register gets cleared.
This register always returns "0x0000" for reading.

Address: 0xF054(MCINTCLL/MCINTCL), 0xF055(MCINTCLH)
Access: W
Access size: 8/16bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	MCINTCL															
Byte	MCINTCLH								MCINTCLL							
Bit	MCIR	-	-	-	-	-	-	-	-	-	-	-	-	MCI2C	MCI1C	MCI0C
R/W	W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	MCIR	This bit is a request bit for the MCU status interrupt. Write "1" to this bit before returning from the interrupt routine. Writing "0": Invalid Writing "1": If an unhandled interrupt exists, it generates the interrupt request again.
14 to 3	-	Reserved bits
2	MCI2C	This bit is used to clear status of the data flash erasing/programming completion interrupt. Writing "0": Invalid Writing "1": MCI2S bit of MCISTATL register gets cleared.
1	MCI1C	This bit is used to clear status of the automatic CRC calculation completion interrupt. Writing "0": Invalid Writing "1": MCI1S bit of MCISTATL register gets cleared.
0	MCI0C	This bit is used to clear status of the RAM parity error occurrence interrupt. Writing "0": Invalid Writing "1": MCI0S bit of MCISTATL register gets cleared.

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 (secondary to octonary function) of the general-purpose port.
For receive side data, it is not required to set the mode (2^{nd} to 8^{th} function) of the general-purpose port.
Figure 29-1 shows a concept diagram of the communication test. Figure 29-2 shows a flow chart of the communication test.

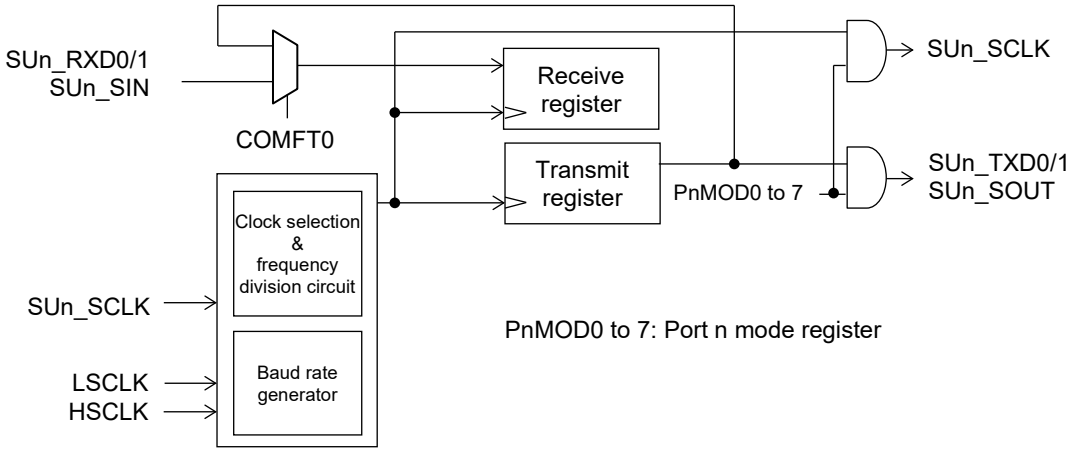


Figure 29-1 Communication Test Concept Diagram

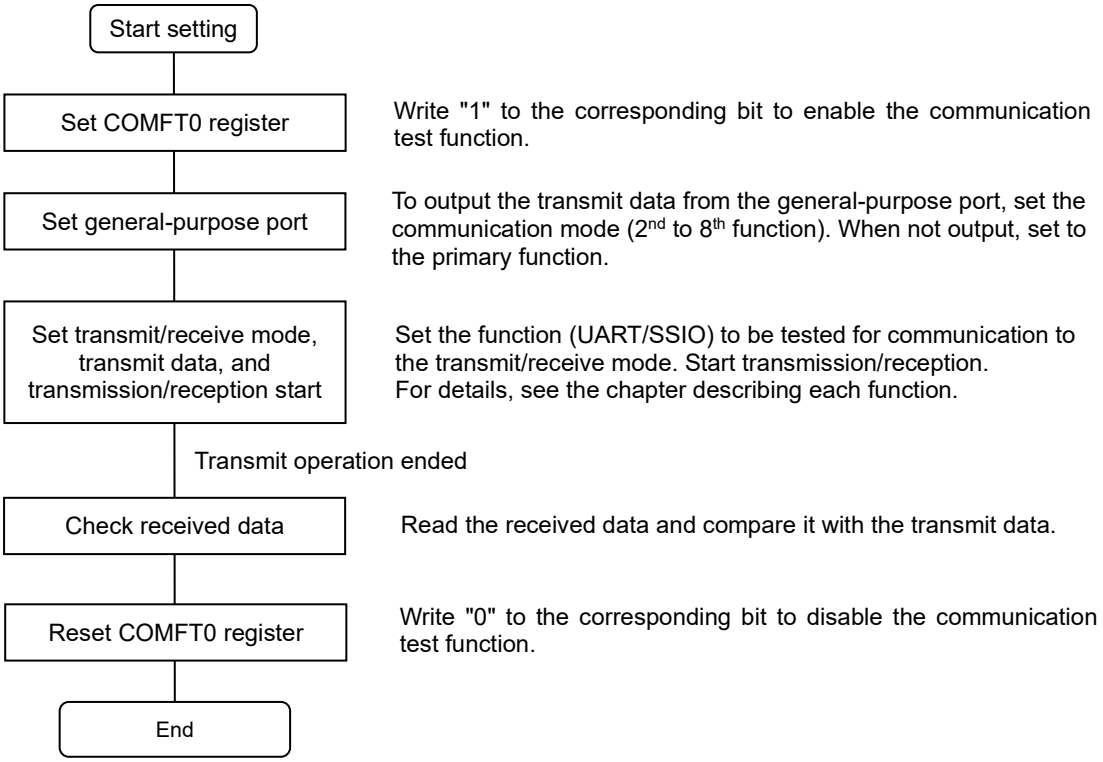


Figure 29-2 Communication Test Flow Chart

29.3.2 Unused ROM Area Access Reset Function

This function constantly monitors the program counter (PC) of the CPU.
It generates the LSI reset when it detects that the program counter (PC) executes a program located outside of the area.
This function can be enabled/disabled by the code option.
The reset flag due to unused ROM area access can be confirmed with the SRSTAT register.
See Chapter 3 "Reset Function" for details of the reset flag.

<ROM unused area>

■ ML62Q1300 group

		CSR:PC
ML62Q1367/ML62Q1347	:	0x0:0FFC0 to 0x7:0FFFF
ML62Q1366/ML62Q1346	:	0x0:0BFC0 to 0x7:0FFFF
ML62Q1365/ML62Q1345/ML62Q1335/ML62Q1325	:	0x0:07FC0 to 0x7:0FFFF
ML62Q1334/ML62Q1324	:	0x0:05FC0 to 0x7:0FFFF
ML62Q1333/ML62Q1323	:	0x0:03FC0 to 0x7:0FFFF

■ ML62Q1500/ML62Q1800 group

		CSR:PC
ML62Q1879/ML62Q1869/ML62Q1859	:	0x7:0FFC0 to 0x7:0FFFF
ML62Q1878/ML62Q1868/ML62Q1858	:	0x5:0FFC0 to 0x7:0FFFF
ML62Q1577/ML62Q1567/ML62Q1557	:	0x3:0FFC0 to 0x7:0FFFF
ML62Q1576/ML62Q1566/ML62Q1556	:	0x2:0FFC0 to 0x7:0FFFF
ML62Q1575/ML62Q1565/ML62Q1555	:	0x2:07FC0 to 0x7:0FFFF
ML62Q1574/ML62Q1564/ML62Q1554/ML62Q1544/ML62Q1534	:	0x1:0FFC0 to 0x7:0FFFF
ML62Q1573/ML62Q1563/ML62Q1553/ML62Q1543/ML62Q1533	:	0x1:07FC0 to 0x7:0FFFF
ML62Q1552/ML62Q1542/ML62Q1532	:	0x0:0FFC0 to 0x7:0FFFF
ML62Q1551/ML62Q1541/ML62Q1531	:	0x0:0BFC0 to 0x7:0FFFF
ML62Q1550/ML62Q1540/ML62Q1530	:	0x0:07FC0 to 0x7:0FFFF

■ ML62Q1700 group

		CSR:PC
ML62Q1749/ML62Q1739/ML62Q1729	:	0x7:0FFC0 to 0x7:0FFFF
ML62Q1748/ML62Q1738/ML62Q1728	:	0x5:0FFC0 to 0x7:0FFFF
ML62Q1747/ML62Q1737/ML62Q1727	:	0x3:0FFC0 to 0x7:0FFFF
ML62Q1746/ML62Q1736/ML62Q1726	:	0x2:0FFC0 to 0x7:0FFFF
ML62Q1745/ML62Q1735/ML62Q1725	:	0x2:07FC0 to 0x7:0FFFF
ML62Q1744/ML62Q1734/ML62Q1724/ML62Q1714/ML62Q1704	:	0x1:0FFC0 to 0x7:0FFFF
ML62Q1743/ML62Q1733/ML62Q1723/ML62Q1713/ML62Q1703	:	0x1:07FC0 to 0x7:0FFFF
ML62Q1722/ML62Q1712/ML62Q1702	:	0x0:0FFC0 to 0x7:0FFFF
ML62Q1721/ML62Q1711/ML62Q1701	:	0x0:0BFC0 to 0x7:0FFFF
ML62Q1720/ML62Q1710/ML62Q1700	:	0x0:07FC0 to 0x7:0FFFF

[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.

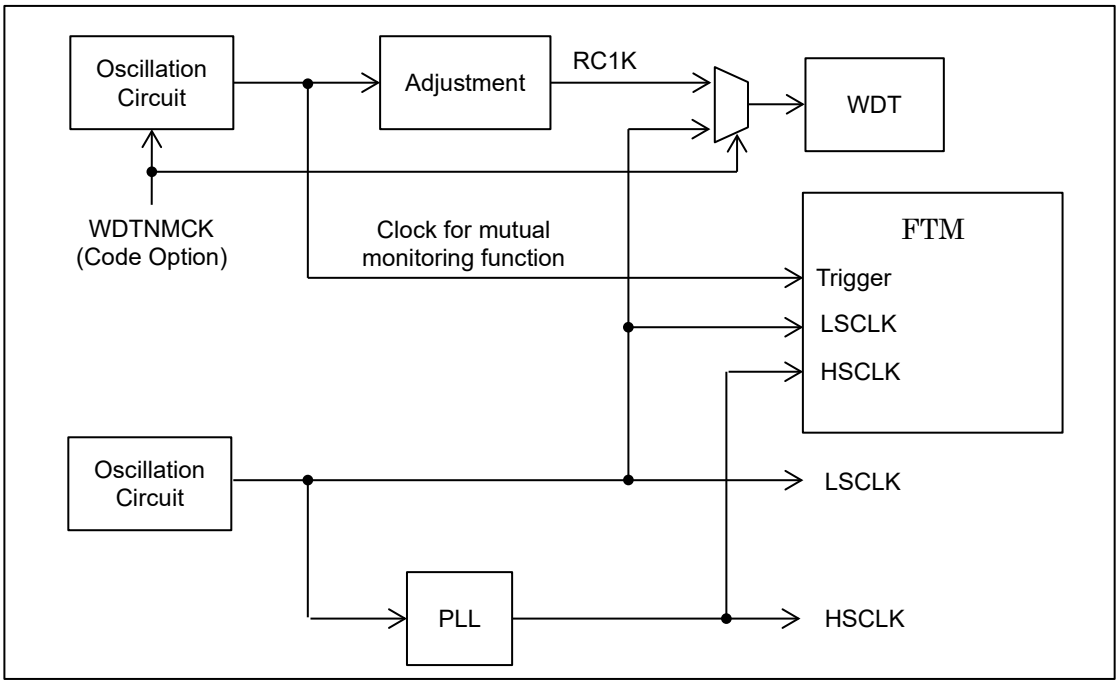
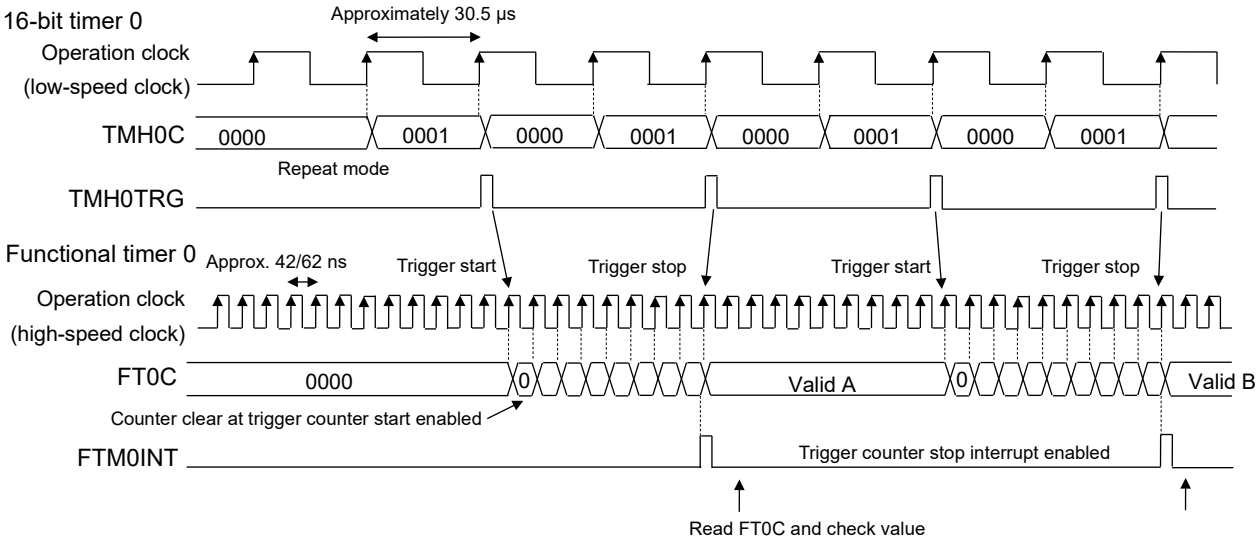


Figure 29-3 Clock Mutual Monitoring Function Block Diagram

Figure 29-4 shows an example of the monitoring operation, using 16-bit timer 0 and Functional timer 0, for the high-speed clock (PLL oscillation circuit) oscillation.



TMH0C: 16-bit timer 0 counter register
TMH0TRG: 16-bit timer 0 trigger
FT0C: FTM0 counter register
FTM0INT: FTM0 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.

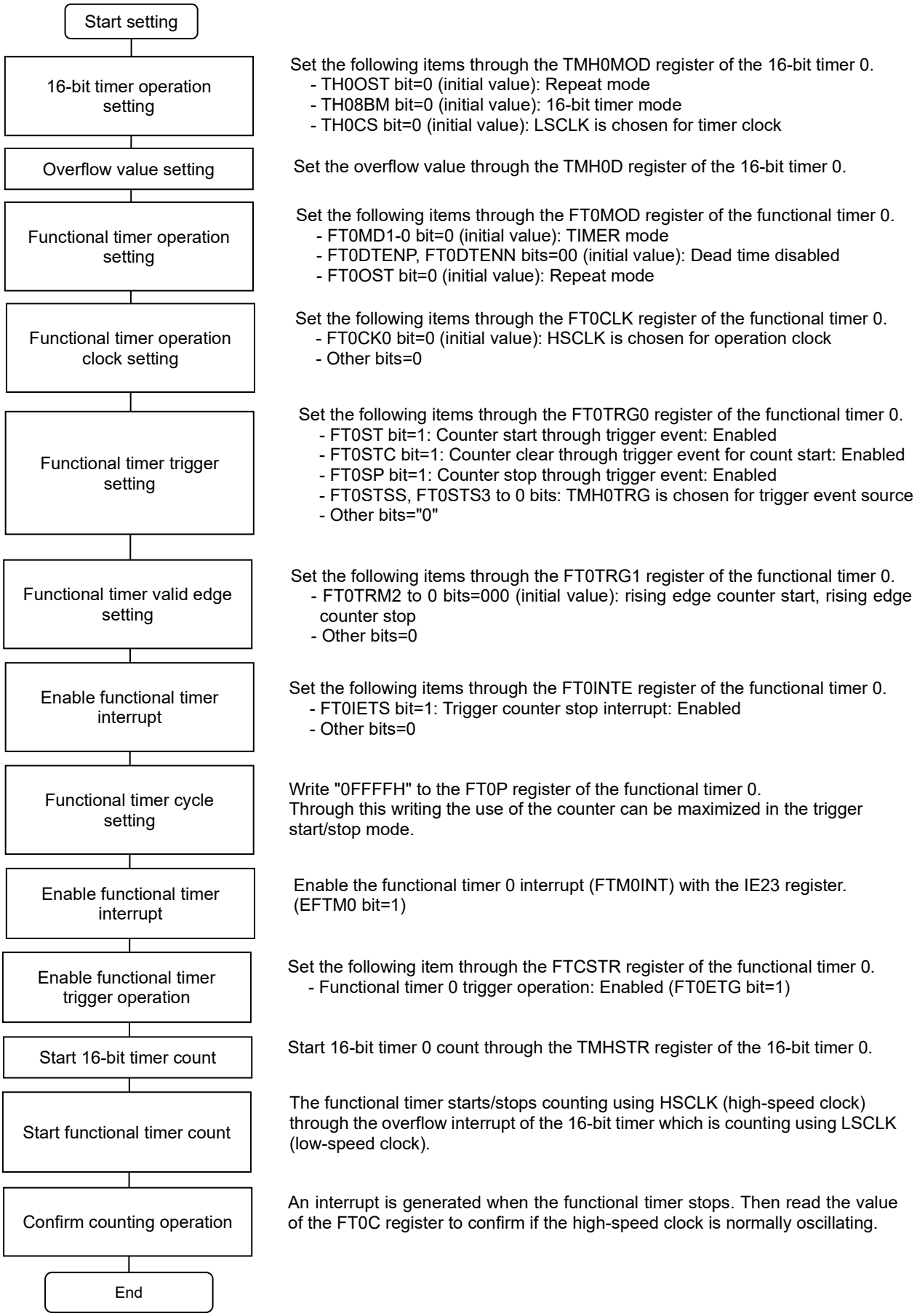


Figure 29-5 Setting of High-Speed Clock (PLL Oscillation Circuit) Oscillation Monitoring Example

[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 DMA controller 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.

29.3.8 Clock Backup Function and Its Test

The built-in test function automatically switches the low-speed crystal oscillation to the low-speed RC oscillation, when the oscillation is stopped.
See Chapter 6 "Clock Generation Circuit" for details.

Appendix A

Appendix A Register List

The register lists are shown below.Access "Reserved" register is not guaranteed. Please do not access.
Initial value with *1 depend on code option that is set. See [26.2.3 code option 2 (CODEOP2)] for more details.

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF000	Data segment register	DSR	-	R/W	8	0x00
0xF001	Reserved	-	-	-	-	-
0xF002	High-speed clock mode register	FHCKMODL	FHCKMOD	R/W	8/16	0x00
0xF003		FHCKMODH		R/W	8	0x44
0xF004	Low-speed clock mode register	FLMOD	-	R/W	8	0x00
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	0x00
0xF009	Reserved	-	-	-	-	-
0xF00A	Backup Control register	FBUCON	-	R/W	8	0x00
0xF00B	Reserved	-	-	-	-	-
0xF00C	Backup Clock Status register	FBUSTAT	-	R	8	0x01
0xF00D	Reserved	-	-	-	-	-
0xF00E	Reserved	-	-	-	-	-
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	8/16	0x00
0xF01B	Standby control register H	SBYCONH		R	8	0x00
0xF01C to 0xF01F	Reserved	-	-	-	-	-
0xF020	Interrupt enable register 01	IE0	IE01	R/W	8/16	0x00
0xF021		IE1		R/W	8	0x00
0xF022	Interrupt enable register 23	IE2	IE23	R/W	8/16	0x00
0xF023		IE3		R/W	8	0x00
0xF024	Interrupt enable register 45	IE4	IE45	R/W	8/16	0x00
0xF025		IE5		R/W	8	0x00
0xF026	Interrupt enable register 67	IE6	IE67	R/W	8/16	0x00
0xF027		IE7		R/W	8	0x00
0xF028	Interrupt request register 01	IRQ0	IRQ01	R/W	8/16	0x00
0xF029		IRQ1		R/W	8	0x00
0xF02A	Interrupt request register 23	IRQ2	IRQ23	R/W	8/16	0x00
0xF02B		IRQ3		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF02C	Interrupt request register 45	IRQ4	IRQ45	R/W	8/16	0x00
0xF02D		IRQ5		R/W	8	0x00
0xF02E	Interrupt request register 67	IRQ6	IRQ67	R/W	8/16	0x00
0xF02F		IRQ7		R/W	8	0x00
0xF030	Interrupt level control enable register	ILEN	-	R/W	8	0x00
0xF031	Reserved	-	-	-	-	-
0xF032	Current interrupt level management register	CIL	-	R/W	8	0x00
0xF033	Reserved	-	-	-	-	-
0xF034	Interrupt level control register 0	ILC00	ILC0	R/W	8/16	0x00
0xF035		ILC01		R/W	8	0x00
0xF036	Interrupt level control register 1	ILC10	ILC1	R/W	8/16	0x00
0xF037		ILC11		R/W	8	0x00
0xF038	Interrupt level control register 2	ILC20	ILC2	R/W	8/16	0x00
0xF039		ILC21		R/W	8	0x00
0xF03A	Interrupt level control register 3	ILC30	ILC3	R/W	8/16	0x00
0xF03B		ILC31		R/W	8	0x00
0xF03C	Interrupt level control register 4	ILC40	ILC4	R/W	8/16	0x00
0xF03D		ILC41		R/W	8	0x00
0xF03E	Interrupt level control register 5	ILC50	ILC5	R/W	8/16	0x00
0xF03F		ILC51		R/W	8	0x00
0xF040	Interrupt level control register 6	ILC60	ILC6	R/W	8/16	0x00
0xF041		ILC61		R/W	8	0x00
0xF042	Interrupt level control register 7	ILC70	ILC7	R/W	8/16	0x00
0xF043		ILC71		R/W	8	0x00
0xF044	External Interrupt Control register 0	EICON0L	EICON0	R/W	8/16	0x00
0xF045		EICON0H		R/W	8	0x00
0xF046	Reserved	-	-	-	-	-
0xF047	Reserved	-	-	-	-	-
0xF048	External Interrupt Mode register 0	EIMOD0L	EIMOD0	R/W	8/16	0x00
0xF049		EIMOD0H		R/W	8	0x00
0xF04A to 0xF04F	Reserved	-	-	-	-	-
0xF050	MCU Status Interrupt Enable register	MCINTEL	-	R/W	8	0x00
0xF051	Reserved	-	-	-	-	-
0xF052	MCU Status Interrupt register	MCISTATL	-	R	8	0x00
0xF053	Reserved	-	-	-	-	-
0xF054	MCU Status Interrupt Clear register	MCINTCLL	MCINTCL	W	8/16	0x00
0xF055		MCINTCLH		W	8	0x00
0xF056	Reserved	-	-	-	-	-
0xF057	Reserved	-	-	-	-	-
0xF058	Reset Status register	RSTATL	RSTAT	R/W	8/16	Undefined
0xF059		RSTATH		R/W	8	Undefined
0xF05A	Safety Function Reset Status register	SRSTAT	-	R/W	8	Undefined
0xF05B	Reserved	-	-	-	-	-
0xF05C	Software Reset Acceptor	SOFTRACP	-	W	8	0x00
0xF05D	Reserved	-	-	-	-	-
0xF05E	Software Reset Control register	SOFTRCON	-	R/W	8	0x00
0xF05F	Reserved	-	-	-	-	-
0xF060	Low-speed Time Base Counter register	LTBR	-	R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF061	Reserved	-	-	-	-	-
0xF062	Low-speed Time Base Counter Control register	LTBCCON	-	R/W	8	0x01
0xF063	Reserved	-	-	-	-	-
0xF064	Simplified RTC Time Base Counter register	LTBRR	-	R	8	0x00
0xF065	Reserved	-	-	-	-	-
0xF066	Low-speed Time Base Counter Frequency Adjustment register	LTBADJL	LTBADJ	R/W	8	0x00
0xF067		LTBADJH		R/W	8	0x00
0xF068	Low-speed Time Base Counter Interrupt selection register	LTBINTL	LTBINT	R/W	8/16	0x30
0xF069		LTBINTH		R/W	8	0x06
0xF06A to 0xF06F	Reserved	-	-	-	-	-
0xF070	Block clock control register 0	BCKCON0L	BCKCON0	R/W	8/16	0x00
0xF071		BCKCON0H		R/W	8	0x00
0xF072	Block clock control register 1	BCKCON1L	BCKCON1	R/W	8/16	0x00
0xF073		BCKCON1H		R/W	8	0x00
0xF074	Block clock control register 2	BCKCON2L	BCKCON2	R/W	8/16	0x00
0xF075		BCKCON2H		R/W	8	0x00
0xF076	Block clock control register 3	BCKCON3L	BCKCON3	R/W	8/16	0x00
0xF077		BCKCON3H		R/W	8	0x00
0xF078	Block reset control register 0	BRECON0L	BRECON0	R/W	8/16	0x00
0xF079		BRECON0H		R/W	8	0x00
0xF07A	Block reset control register 1	BRECON1L	BRECON1	R/W	8/16	0x00
0xF07B		BRECON1H		R/W	8	0x00
0xF07C	Block reset control register 2	BRECON2L	BRECON2	R/W	8/16	0x00
0xF07D		BRECON2H		R/W	8	0x00
0xF07E	Block reset control register 3	BRECON3L	BRECON3	R/W	8/16	0x00
0xF07F		BRECON3H		R/W	8	0x00
0xF080	Low-speed RC oscillation frequency adjustment register	LRCADJ	-	R/W	8	0x00
0xF081 to 0xF08F	Reserved	-	-	-	-	-
0xF090	Flash address register	FLASHAL	FLASHA	R/W	8/16	0xFF
0xF091		FLASHAH		R/W	8	0xFF
0xF092	Flash data register 0	FLASHD0L	FLASHD0	R/W	8/16	0xFF
0xF093		FLASHD0H		R/W	8	0xFF
0xF094	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	*1

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF0A1 to 0xF0AF	Reserved	-	-	-	-	-
0xF0B0	RAM guard setting register 0	RAMGD	-	R/W	8	0x00
0xF0B1	Reserved	-	-	-	-	-
0xF0B2	Reserved	-	-	-	-	-
0xF0B3	Reserved	-	-	-	-	-
0xF0B4	SFR guard setting register 0	SFRGD0L	SFRGD0	R/W	8/16	0x00
0xF0B5		SFRGD0H		R/W	8	0x00
0xF0B6	SFR guard setting register 1	SFRGD1L	SFRGD1	R/W	8/16	0x00
0xF0B7		SFRGD1H		R/W	8	0x00
0xF0B8	Reserved	-	-	-	-	-
0xF0B9	Reserved	-	-	-	-	-
0xF0BA	SA-ADC test mode	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
0xF0C3		BZ0MODH		R/W	8	0x00
0xF0C4	Clock Backup Test Mode Acceptor	FBTACP	-	W	8	0x00
0xF0C5	Reserved	-	-	-	-	-
0xF0C6	Clock Backup Test Mode register	FBTCON	-	R/W	8	0x00
0xF0C7	Reserved	-	-	-	-	-
0xF0C8	Simplified RTC Acceptor	SRTCACP	-	W	8	0x00
0xF0C9	Reserved	-	-	-	-	-
0xF0CA	Simplified RTC Minute/Second Counter	SRTCSEC	SRTCMAS	R/W	8/16	0x00
0xF0CB		SRTCMIN		R/W	8	0x80
0xF0CC	Simplified RTC Control Register	SRTCCON	-	R/W	8	0x00
0xF0CD to 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 Calculation 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	CRC Calculation Mode Register	CRCMOD	-	R/W	8	0x00
0xF0DD	Reserved	-	-	-	-	-

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF0E2	Reserved	-	-	-	-	-
0xF0E3	Reserved	-	-	-	-	-
0xF0E4	Expanded external interrupt control register 0	EEICON0L	EEICON0	R/W	8/16	0x00
0xF0E5		EEICON0H		R/W	8	0x00
0xF0E6	Reserved	-	-	-	-	-
0xF0E7	Reserved	-	-	-	-	-
0xF0E8	Expanded external interrupt mode register 0	EEIMOD0L	EEIMOD0	R/W	8/16	0x00
0xF0E9		EEIMOD0H		R	8	0x00
0xF0EA	Expanded external interrupt mode register 1	EEIMOD1L	EEIMOD1	R/W	8/16	0x00
0xF0EB		EEIMOD1H		R	8	0x00
0xF0EC	Expanded external interrupt status register	EEISTATL	EEISTAT	R	8/16	0x00
0xF0ED		EEISTATH		R	8	0x00
0xF0EE	Expanded external interrupt clear register	EEINTCL	EEINTC	W	8/16	0x00
0xF0EF		EEINTCH		W	8	0x00
0xF0F0	Bias control register	BIASCONL	BIASCON	R/W	8/16	0x08
0xF0F1		BIASCONH		R/W	8	0x00
0xF0F2	Display mode register	DSPMODL	DSPMOD	R/W	8/16	0x40
0xF0F3		DSPMODH		R/W	8	0x00
0xF0F4	Display control register	DSPCONL	DSPCON	R/W	8/16	0x00
0xF0F5		DSPCONH		R/W	8	0x00
0xF0F6	Segment mode register 0	SEGMOD0L	SEGMOD0	R/W	8/16	0x00
0xF0F7		SEGMOD0H		R/W	8	0x00
0xF0F8	Segment mode register 1	SEGMOD1L	SEGMOD1	R/W	8/16	0x00
0xF0F9		SEGMOD1H		R/W	8	0x00
0xF0FA	Segment mode register 2	SEGMOD2L	SEGMOD2	R/W	8/16	0x00
0xF0FB		SEGMOD2H		R/W	8	0x00
0xF0FC	Segment mode register 3	SEGMOD3L	SEGMOD3	R/W	8/16	0x00
0xF0FD		SEGMOD3H		R/W	8	0x00
0xF0FE	Segment mode register 4	SEGMOD4L	SEGMOD4	R/W	8/16	0x00
0xF0FF		-		R	8	0x00
0xF100	Display register 0	DSPR00	DSPRW00	R/W	8/16	Undefined
0xF101	Display register 1	DSPR01		R/W	8	Undefined
0xF102	Display register 2	DSPR02	DSPRW02	R/W	8/16	Undefined
0xF103	Display register 3	DSPR03		R/W	8	Undefined
0xF104	Display register 4	DSPR04	DSPRW04	R/W	8/16	Undefined
0xF105	Display register 5	DSPR05		R/W	8	Undefined
0xF106	Display register 6	DSPR06	DSPRW06	R/W	8/16	Undefined
0xF107	Display register 7	DSPR07		R/W	8	Undefined
0xF108	Display register 8	DSPR08	DSPRW08	R/W	8/16	Undefined
0xF109	Display register 9	DSPR09		R/W	8	Undefined
0xF10A	Display register 10	DSPR10	DSPRW10	R/W	8/16	Undefined
0xF10B	Display register 11	DSPR11		R/W	8	Undefined
0xF10C	Display register 12	DSPR12	DSPRW12	R/W	8/16	Undefined
0xF10D	Display register 13	DSPR13		R/W	8	Undefined
0xF10E	Display register 14	DSPR14	DSPRW14	R/W	8/16	Undefined
0xF10F	Display register 15	DSPR15		R/W	8	Undefined
0xF110	Display register 16	DSPR16	DSPRW16	R/W	8/16	Undefined
0xF111	Display register 17	DSPR17		R/W	8	Undefined
0xF112	Display register 18	DSPR18	DSPRW18	R/W	8/16	Undefined
0xF113	Display register 19	DSPR19		R/W	8	Undefined

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF114	Display register 20	DSPR20	DSPRW20	R/W	8/16	Undefined
0xF115	Display register 21	DSPR21		R/W	8	Undefined
0xF116	Display register 22	DSPR22	DSPRW22	R/W	8/16	Undefined
0xF117	Display register 23	DSPR23		R/W	8	Undefined
0xF118	Display register 24	DSPR24	DSPRW24	R/W	8/16	Undefined
0xF119	Display register 25	DSPR25		R/W	8	Undefined
0xF11A	Display register 26	DSPR26	DSPRW26	R/W	8/16	Undefined
0xF11B	Display register 27	DSPR27		R/W	8	Undefined
0xF11C	Display register 28	DSPR28	DSPRW28	R/W	8/16	Undefined
0xF11D	Display register 29	DSPR29		R/W	8	Undefined
0xF11E	Display register 30	DSPR30	DSPRW30	R/W	8/16	Undefined
0xF11F	Display register 31	DSPR31		R/W	8	Undefined
0xF120	Display register 32	DSPR32	DSPRW32	R/W	8/16	Undefined
0xF121	Display register 33	DSPR33		R/W	8	Undefined
0xF122	Display register 34	DSPR34	DSPRW34	R/W	8/16	Undefined
0xF123	Display register 35	DSPR35		R/W	8	Undefined
0xF124	Display register 36	DSPR36	DSPRW36	R/W	8/16	Undefined
0xF125	Display register 37	DSPR37		R/W	8	Undefined
0xF126	Display register 38	DSPR38	DSPRW38	R/W	8/16	Undefined
0xF127	Display register 39	DSPR39		R/W	8	Undefined
0xF128	Display register 40	DSPR40	DSPRW40	R/W	8/16	Undefined
0xF129	Display register 41	DSPR41		R/W	8	Undefined
0xF12A	Display register 42	DSPR42	DSPRW42	R/W	8/16	Undefined
0xF12B	Display register 43	DSPR43		R/W	8	Undefined
0xF12C	Display register 44	DSPR44	DSPRW44	R/W	8/16	Undefined
0xF12D	Display register 45	DSPR45		R/W	8	Undefined
0xF12E	Display register 46	DSPR46	DSPRW46	R/W	8/16	Undefined
0xF12F	Display register 47	DSPR47		R/W	8	Undefined
0xF130	Display register 48	DSPR48	DSPRW48	R/W	8/16	Undefined
0xF131	Display register 49	DSPR49		R/W	8	Undefined
0xF132	Display register 50	DSPR50	DSPRW50	R/W	8/16	Undefined
0xF133	Display register 51	DSPR51		R/W	8	Undefined
0xF134	Display register 52	DSPR52	DSPRW52	R/W	8/16	Undefined
0xF135	Display register 53	DSPR53		R/W	8	Undefined
0xF136	Display register 54	DSPR54	DSPRW54	R/W	8/16	Undefined
0xF137	Display register 55	DSPR55		R/W	8	Undefined
0xF138	Display register 56	DSPR56	DSPRW56	R/W	8/16	Undefined
0xF139	Display register 57	DSPR57		R/W	8	Undefined
0xF13A	Display register 58	DSPR58	DSPRW58	R/W	8/16	Undefined
0xF13B	Display register 59	DSPR59		R/W	8	Undefined
0xF13C	Display register 60	DSPR60	DSPRW60	R/W	8/16	Undefined
0xF13D	Display register 61	DSPR61		R/W	8	Undefined
0xF13E	Display register 62	DSPR62	DSPRW62	R/W	8/16	Undefined
0xF13F	Display register 63	DSPR63		R/W	8	Undefined
0xF140	Display register 64	DSPR64	DSPRW64	R/W	8/16	Undefined
0xF141 to 0xF1FF	Reserved	-	-	-	-	-
0xF200	Port 0 data register	P0DI	P0D	R/W	8/16	0xFF
0xF201		P0DO		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF202	Port 0 mode register 01	P0MOD0	P0MOD01	R/W	8/16	0x05
0xF203		P0MOD1		R/W	8	0x00
0xF204	Port 0 mode register 23	P0MOD2	P0MOD23	R/W	8/16	0x00
0xF205		P0MOD3		R/W	8	0x00
0xF206	Port 0 mode register 45	P0MOD4	P0MOD45	R/W	8/16	0x00
0xF207		P0MOD5		R/W	8	0x00
0xF208	Port 0 mode register 67	P0MOD6	P0MOD67	R/W	8/16	0x00
0xF209		P0MOD7		R/W	8	0x00
0xF20A	Port 0 pulse mode register	P0PMDL	P0PMD	R/W	8/16	0x00
0xF20B		P0PMDH		R/W	8	0x00
0xF20C	Port 0 pulse selection register	P0PSLL	P0PSL	R/W	8/16	0x00
0xF20D		P0PSLH		R/W	8	0x00
0xF20E	Reserved	-	-	-	-	-
0xF20F		-	-	-	-	-
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	Port 1 pulse selection register	P1PSLL	P1PSL	R/W	8/16	0x00
0xF21D		P1PSLH		R/W	8	0x00
0xF21E	Reserved	-	-	-	-	-
0xF21F		-	-	-	-	-
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
0xF22C	Port 2 pulse selection register	P2PSLL	P2PSL	R/W	8/16	0x00
0xF22D		P2PSLH		R/W	8	0x00
0xF22E	Reserved	-	-	-	-	-
0xF22F		-	-	-	-	-
0xF230	Port 3 data register	P3DI	P3D	R/W	8/16	0xFF
0xF231		P3DO		R/W	8	0x00
0xF232	Port 3 mode register 01	P3MOD0	P3MOD01	R/W	8/16	0x00
0xF233		P3MOD1		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF234	Port 3 mode register 23	P3MOD2	P3MOD23	R/W	8/16	0x00
0xF235		P3MOD3		R/W	8	0x00
0xF236 to 0xF239	Reserved	-	-	-	-	-
0xF23A	Port 3 pulse mode register	P3PMDL	P3PMD	R/W	8/16	0x00
0xF23B		P3PMDH		R/W	8	0x00
0xF23C	Port 3 pulse selection register	P3PSLL	P3PSL	R/W	8/16	0x00
0xF23D		P3PSLH		R/W	8	0x00
0xF23E	Reserved	-	-	-	-	-
0xF23F		-	-	-	-	-
0xF240	Port 4 data register	P4DI	P4D	R/W	8/16	0xFF
0xF241		P4DO		R/W	8	0x00
0xF242	Port 4 mode register 01	P4MOD0	P4MOD01	R/W	8/16	0x00
0xF243		P4MOD1		R/W	8	0x00
0xF244	Port 4 mode register 23	P4MOD2	P4MOD23	R/W	8/16	0x00
0xF245		P4MOD3		R/W	8	0x00
0xF246	Port 4 mode register 45	P4MOD4	P4MOD45	R/W	8/16	0x00
0xF247		P4MOD5		R/W	8	0x00
0xF248	Port 4 mode register 67	P4MOD6	P4MOD67	R/W	8/16	0x00
0xF249		P4MOD7		R/W	8	0x00
0xF24A to 0xF24F	Reserved	-	-	-	-	-
0xF250	Port 5 data register	P5DI	P5D	R/W	8/16	0xFF
0xF251		P5DO		R/W	8	0x00
0xF252	Port 5 mode register 01	P5MOD0	P5MOD01	R/W	8/16	0x00
0xF253		P5MOD1		R/W	8	0x00
0xF254	Port 5 mode register 23	P5MOD2	P5MOD23	R/W	8/16	0x00
0xF255		P5MOD3		R/W	8	0x00
0xF256	Port 5 mode register 45	P5MOD4	P5MOD45	R/W	8/16	0x00
0xF257		P5MOD5		R/W	8	0x00
0xF258	Port 5 mode register 67	P5MOD6	P5MOD67	R/W	8/16	0x00
0xF259		P5MOD7		R/W	8	0x00
0xF25A to 0xF25F	Reserved	-	-	-	-	-
0xF260	Port 6 data register	P6DI	P6D	R/W	8/16	0xFF
0xF261		P6DO		R/W	8	0x00
0xF262	Port 6 mode register 01	P6MOD0	P6MOD01	R/W	8/16	0x00
0xF263		P6MOD1		R/W	8	0x00
0xF264	Port 6 mode register 23	P6MOD2	P6MOD23	R/W	8/16	0x00
0xF265		P6MOD3		R/W	8	0x00
0xF266	Port 6 mode register 45	P6MOD4	P6MOD45	R/W	8/16	0x00
0xF267		P6MOD5		R/W	8	0x00
0xF268	Port 6 mode register 67	P6MOD6	P6MOD67	R/W	8/16	0x00
0xF269		P6MOD7		R/W	8	0x00
0xF26A to 0xF26F	Reserved	-	-	-	-	-

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF270	Port 7 data register	P7DI	P7D	R/W	8/16	0xFF
0xF271		P7DO		R/W	8	0x00
0xF272	Port 7 mode register 01	P7MOD0	P7MOD01	R/W	8/16	0x00
0xF273		P7MOD1		R/W	8	0x00
0xF274	Port 7 mode register 23	P7MOD2	P7MOD23	R/W	8/16	0x00
0xF275		P7MOD3		R/W	8	0x00
0xF276	Port 7 mode register 45	P7MOD4	P7MOD45	R/W	8/16	0x00
0xF277		P7MOD5		R/W	8	0x00
0xF278	Port 7 mode register 67	P7MOD6	P7MOD67	R/W	8/16	0x00
0xF279		P7MOD7		R/W	8	0x00
0xF27A to 0xF27F	Reserved	-	-	-	-	-
0xF280	Port 8 data register	P8DI	P8D	R/W	8/16	0xFF
0xF281		P8DO		R/W	8	0x00
0xF282	Port 8 mode register 01	P8MOD0	P8MOD01	R/W	8/16	0x00
0xF283		P8MOD1		R/W	8	0x00
0xF284	Port 8 mode register 23	P8MOD2	P8MOD23	R/W	8/16	0x00
0xF285		P8MOD3		R/W	8	0x00
0xF286	Port 8 mode register 45	P8MOD4	P8MOD45	R/W	8/16	0x00
0xF287		P8MOD5		R/W	8	0x00
0xF288	Port 8 mode register 67	P8MOD6	P8MOD67	R/W	8/16	0x00
0xF289		P8MOD7		R/W	8	0x00
0xF28A to 0xF28F	Reserved	-	-	-	-	-
0xF290	Port 9 data register	P9DI	P9D	R/W	8/16	0xFF
0xF291		P9DO		R/W	8	0x00
0xF292	Port 9 mode register 01	P9MOD0	P9MOD01	R/W	8/16	0x00
0xF293		P9MOD1		R/W	8	0x00
0xF294	Port 9 mode register 23	P9MOD2	P9MOD23	R/W	8/16	0x00
0xF295		P9MOD3		R/W	8	0x00
0xF296	Port 9 mode register 45	P9MOD4	P9MOD45	R/W	8/16	0x00
0xF297		P9MOD5		R/W	8	0x00
0xF298	Port 9 mode register 67	P9MOD6	P9MOD67	R/W	8/16	0x00
0xF299		P9MOD7		R/W	8	0x00
0xF29A to 0xF29F	Reserved	-	-	-	-	-
0xF2A0	Port A data register	PADI	PAD	R/W	8/16	0xFF
0xF2A1		PADO		R/W	8	0x00
0xF2A2	Port A mode register 01	PAMOD0	PAMOD01	R/W	8/16	0x00
0xF2A3		PAMOD1		R/W	8	0x00
0xF2A4	Port A mode register 23	PAMOD2	PAMOD23	R/W	8/16	0x00
0xF2A5		PAMOD3		R/W	8	0x00
0xF2A6	Port A mode register 45	PAMOD4	PAMOD45	R/W	8/16	0x00
0xF2A7		PAMOD5		R/W	8	0x00
0xF2A8	Port A mode register 67	PAMOD6	PAMOD67	R/W	8/16	0x00
0xF2A9		PAMOD7		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF2AA to 0xF2AF	Reserved	-	-	-	-	-
0xF2B0	Port B data register	PBDI	PBD	R/W	8/16	0xFF
0xF2B1		PBDO		R/W	8	0x00
0xF2B2	Port B mode register 01	PBMOD0	PBMOD01	R/W	8/16	0x00
0xF2B3		PBMOD1		R/W	8	0x00
0xF2B4	Port B mode register 23	PBMOD2	PBMOD23	R/W	8/16	0x00
0xF2B5		PBMOD3		R/W	8	0x00
0xF2B6	Port B mode register 45	PBMOD4	PBMOD45	R/W	8/16	0x00
0xF2B7		PBMOD5		R/W	8	0x00
0xF2B8	Port B mode register 67	PBMOD6	PBMOD67	R/W	8/16	0x00
0xF2B9		PBMOD7		R/W	8	0x00
0xF2BA to 0xF2EF	Reserved	-	-	-	-	-
0xF2F0	PORTXT data input register	PXTDI	-	R	8	Undefined
0xF2F1	Reserved	-	-	-	-	-
0xF2F2	PORTXT mode register 01	PXTMOD0	PXTMOD01	R/W	8/16	0x00
0xF2F3		PXTMOD1		R/W	8	0x00
0xF2F4 to 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	16-bit timer 6 data register	TMH6DL	TMH6D	R/W	8/16	0xFF
0xF30D		TMH6DH		R/W	8	0xFF
0xF30E	16-bit timer 7 data register	TMH7DL	TMH7D	R/W	8/16	0xFF
0xF30F		TMH7DH		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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF31A	16-bit timer 5 counter register	TMH5CL	TMH5C	R/W	8/16	0x00
0xF31B		TMH5CH		R/W	8	0x00
0xF31C	16-bit timer 6 counter register	TMH6CL	TMH6C	R/W	8/16	0x00
0xF31D		TMH6CH		R/W	8	0x00
0xF31E	16-bit timer 7 counter register	TMH7CL	TMH7C	R/W	8/16	0x00
0xF31F		TMH7CH		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		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	16-bit timer 6 mode register	TMH6MODL	TMH6MOD	R/W	8/16	0x00
0xF32D		TMH6MODH		R/W	8	0x00
0xF32E	16-bit timer 7 mode register	TMH7MODL	TMH7MOD	R/W	8/16	0x00
0xF32F		TMH7MODH		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
0xF33C	16-bit timer 6 interrupt status register	TMH6ISL	TMH6IS	R	8/16	0x00
0xF33D		TMH6ISH		R	8	0x00
0xF33E	16-bit timer 7 interrupt status register	TMH7ISL	TMH7IS	R	8/16	0x00
0xF33F		TMH7ISH		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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF34C	16-bit timer 6 interrupt clear register	TMH6ICL	TMH6IC	W	8/16	0x00
0xF34D		TMH6ICH		W	8	0x00
0xF34E	16-bit timer 7 interrupt clear register	TMH7ICL	TMH7IC	W	8/16	0x00
0xF34F		TMH7ICH		W	8	0x00
0xF350	16-bit timer start register	TMHSTRL	TMHSTR	W	8/16	0x00
0xF351		TMHSTRH		W	8	0x00
0xF352	16-bit timer stop register	TMHSTPL	TMHSTP	W	8/16	0x00
0xF353		TMHSTPH		W	8	0x00
0xF354	16-bit timer status register	TMHSTATL	TMHSTAT	R	8/16	0x00
0xF355		TMHSTATH		R	8	0x00
0xF356 to 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	FTM4 cycle register	FT4PL	FT4P	R/W	8/16	0xFF
0xF409		FT4PH		R/W	8	0xFF
0xF40A	FTM5 cycle register	FT5PL	FT5P	R/W	8/16	0xFF
0xF40B		FT5PH		R/W	8	0xFF
0xF40C	FTM6 cycle register	FT6PL	FT6P	R/W	8/16	0xFF
0xF40D		FT6PH		R/W	8	0xFF
0xF40E	FTM7 cycle register	FT7PL	FT7P	R/W	8/16	0xFF
0xF40F		FT7PH		R/W	8	0xFF
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	FTM4 event A register	FT4EAL	FT4EA	R/W	8/16	0x00
0xF419		FT4EAH		R/W	8	0x00
0xF41A	FTM5 event A register	FT5EAL	FT5EA	R/W	8/16	0x00
0xF41B		FT5EAH		R/W	8	0x00
0xF41C	FTM6 event A register	FT6EAL	FT6EA	R/W	8/16	0x00
0xF41D		FT6EAH		R/W	8	0x00
0xF41E	FTM7 event A register	FT7EAL	FT7EA	R/W	8/16	0x00
0xF41F		FT7EAH		R/W	8	0x00
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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
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	FTM4 event B register	FT4EBL	FT4EB	R/W	8/16	0x00
0xF429		FT4EBH		R/W	8	0x00
0xF42A	FTM5 event B register	FT5EBL	FT5EB	R/W	8/16	0x00
0xF42B		FT5EBH		R/W	8	0x00
0xF42C	FTM6 event B register	FT6EBL	FT6EB	R/W	8/16	0x00
0xF42D		FT6EBH		R/W	8	0x00
0xF42E	FTM7 event B register	FT7EBL	FT7EB	R/W	8/16	0x00
0xF42F		FT7EBH		R/W	8	0x00
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
0xF436	FTM3 dead time register	FT3DTL	FT3DT	R/W	8/16	0x00
0xF437		FT3DTH		R/W	8	0x00
0xF438	FTM4 dead time register	FT4DTL	FT4DT	R/W	8/16	0x00
0xF439		FT4DTH		R/W	8	0x00
0xF43A	FTM5 dead time register	FT5DTL	FT5DT	R/W	8/16	0x00
0xF43B		FT5DTH		R/W	8	0x00
0xF43C	FTM6 dead time register	FT6DTL	FT6DT	R/W	8/16	0x00
0xF43D		FT6DTH		R/W	8	0x00
0xF43E	FTM7 dead time register	FT7DTL	FT7DT	R/W	8/16	0x00
0xF43F		FT7DTH		R/W	8	0x00
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	FTM4 counter register	FT4CL	FT4C	R/W	8/16	0x00
0xF449		FT4CH		R/W	8	0x00
0xF44A	FTM5 counter register	FT5CL	FT5C	R/W	8/16	0x00
0xF44B		FT5CH		R/W	8	0x00
0xF44C	FTM6 counter register	FT6CL	FT6C	R/W	8/16	0x00
0xF44D		FT6CH		R/W	8	0x00
0xF44E	FTM7 counter register	FT7CL	FT7C	R/W	8/16	0x00
0xF44F		FT7CH		R/W	8	0x00
0xF450	FTM0 status register	FT0STAT	-	R/W	8	0x30
0xF451	Reserved	-	-	-	-	-
0xF452	FTM1 status register	FT1STAT	-	R/W	8	0x30
0xF453	Reserved	-	-	-	-	-
0xF454	FTM2 status register	FT2STAT	-	R/W	8	0x30
0xF455	Reserved	-	-	-	-	-

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF456	FTM3 status register	FT3STAT	-	R/W	8	0x30
0xF457	Reserved	-	-	-	-	-
0xF458	FTM4 status register	FT4STAT	-	R/W	8	0x30
0xF459	Reserved	-	-	-	-	-
0xF45A	FTM5 status register	FT5STAT	-	R/W	8	0x30
0xF45B	Reserved	-	-	-	-	-
0xF45C	FTM6 status register	FT6STAT	-	R/W	8	0x30
0xF45D	Reserved	-	-	-	-	-
0xF45E	FTM7 status register	FT7STAT	-	R/W	8	0x30
0xF45F	Reserved	-	-	-	-	-
0xF460	FTM0 mode register	FT0MODL	FT0MOD	R/W	8/16	0x00
0xF461		FT0MODH		R/W	8	0x40
0xF462	FTM1 mode register	FT1MODL	FT1MOD	R/W	8/16	0x00
0xF463		FT1MODH		R/W	8	0x40
0xF464	FTM2 mode register	FT2MODL	FT2MOD	R/W	8/16	0x00
0xF465		FT2MODH		R/W	8	0x40
0xF466	FTM3 mode register	FT3MODL	FT3MOD	R/W	8/16	0x00
0xF467		FT3MODH		R/W	8	0x40
0xF468	FTM4 mode register	FT4MODL	FT4MOD	R/W	8/16	0x00
0xF469		FT4MODH		R/W	8	0x40
0xF46A	FTM5 mode register	FT5MODL	FT5MOD	R/W	8/16	0x00
0xF46B		FT5MODH		R/W	8	0x40
0xF46C	FTM6 mode register	FT6MODL	FT6MOD	R/W	8/16	0x00
0xF46D		FT6MODH		R/W	8	0x40
0xF46E	FTM7 mode register	FT7MODL	FT7MOD	R/W	8/16	0x00
0xF46F		FT7MODH		R/W	8	0x40
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	FTM4 clock register	FT4CLKL	FT4CLK	R/W	8/16	0x00
0xF479		FT4CLKH		R/W	8	0x00
0xF47A	FTM5 clock register	FT5CLKL	FT5CLK	R/W	8/16	0x00
0xF47B		FT5CLKH		R/W	8	0x00
0xF47C	FTM6 clock register	FT6CLKL	FT6CLK	R/W	8/16	0x00
0xF47D		FT6CLKH		R/W	8	0x00
0xF47E	FTM7 clock register	FT7CLKL	FT7CLK	R/W	8/16	0x00
0xF47F		FT7CLKH		R/W	8	0x00
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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF488	FTM4 trigger register 0	FT4TRG0L	FT4TRG0	R/W	8/16	0x00
0xF489		FT4TRG0H		R/W	8	0x00
0xF48A	FTM5 trigger register 0	FT5TRG0L	FT5TRG0	R/W	8/16	0x00
0xF48B		FT5TRG0H		R/W	8	0x00
0xF48C	FTM6 trigger register 0	FT6TRG0L	FT6TRG0	R/W	8/16	0x00
0xF48D		FT6TRG0H		R/W	8	0x00
0xF48E	FTM7 trigger register 0	FT7TRG0L	FT7TRG0	R/W	8/16	0x00
0xF48F		FT7TRG0H		R/W	8	0x00
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	FTM4 trigger register 1	FT4TRG1L	FT4TRG1	R/W	8/16	0x00
0xF499		FT4TRG1H		R/W	8	0x00
0xF49A	FTM5 trigger register 1	FT5TRG1L	FT5TRG1	R/W	8/16	0x00
0xF49B		FT5TRG1H		R/W	8	0x00
0xF49C	FTM6 trigger register 1	FT6TRG1L	FT6TRG1	R/W	8/16	0x00
0xF49D		FT6TRG1H		R/W	8	0x00
0xF49E	FTM7 trigger register 1	FT7TRG1L	FT7TRG1	R/W	8/16	0x00
0xF49F		FT7TRG1H		R/W	8	0x00
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	FTM4 interrupt enable register	FT4INTEL	FT4INTE	R/W	8/16	0x00
0xF4A9		FT4INTEH		R/W	8	0x00
0xF4AA	FTM5 interrupt enable register	FT5INTEL	FT5INTE	R/W	8/16	0x00
0xF4AB		FT5INTEH		R/W	8	0x00
0xF4AC	FTM6 interrupt enable register	FT6INTEL	FT6INTE	R/W	8/16	0x00
0xF4AD		FT6INTEH		R/W	8	0x00
0xF4AE	FTM7 interrupt enable register	FT7INTEL	FT7INTE	R/W	8/16	0x00
0xF4AF		FT7INTEH		R/W	8	0x00
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	FTM4 interrupt status register	FT4INTSL	FT4INTS	R	8/16	0x00
0xF4B9		FT4INTSH		R	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF4BA	FTM5 interrupt status register	FT5INTSL	FT5INTS	R	8/16	0x00
0xF4BB		FT5INTSH		R	8	0x00
0xF4BC	FTM6 interrupt status register	FT6INTSL	FT6INTS	R	8/16	0x00
0xF4BD		FT6INTSH		R	8	0x00
0xF4BE	FTM7 interrupt status register	FT7INTSL	FT7INTS	R	8/16	0x00
0xF4BF		FT7INTSH		R	8	0x00
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	FTM4 interrupt clear register	FT4INTCL	FT4INTC	W	8/16	0x00
0xF4C9		FT4INTCH		W	8	0x00
0xF4CA	FTM5 interrupt clear register	FT5INTCL	FT5INTC	W	8/16	0x00
0xF4CB		FT5INTCH		W	8	0x00
0xF4CC	FTM6 interrupt clear register	FT6INTCL	FT6INTC	W	8/16	0x00
0xF4CD		FT6INTCH		W	8	0x00
0xF4CE	FTM7 interrupt clear register	FT7INTCL	FT7INTC	W	8/16	0x00
0xF4CF		FT7INTCH		W	8	0x00
0xF4D0 to 0xF4EF	Reserved	-	-	-	-	-
0xF4F0	FTM common update register	FTCUD	-	W	8	0x00
0xF4F1	Reserved	-	-	-	-	-
0xF4F2	FTM common control register	FTCCONL	FTCCON	R/W	8/16	0x00
0xF4F3		FTCCONH		R/W	8	0x00
0xF4F4	FTM common start register	FTCSTRL	FTCSTR	W	8/16	0x00
0xF4F5		FTCSTRH		W	8	0x00
0xF4F6	FTM common stop register	FTCSTPL	FTCSTP	W	8/16	0x00
0xF4F7		FTCSTPH		W	8	0x00
0xF4F8	FTM common status register	FTCSTATL	FTCSTAT	R	8/16	0x00
0xF4F9		FTCSTATH		R	8	0x00
0xF4FA to 0xF5FF	Reserved	-	-	-	-	-
0xF600	Serial communication unit 0 transmission/reception 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	-	-	-	-	-
0xF604	Serial communication unit 0 transmission interval setting register	SU0DLYL	-	R/W	8	0x00
0xF605	Reserved	-	-	-	-	-
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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
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	8/16	0x00
0xF61B to 0xF61F	Reserved	-	-	-	-	-
0xF620	Serial communication unit 1 transmission/reception 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	-	R/W	8	0x00
0xF625	Reserved	-	-	-	-	-
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
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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF63B to 0xF63F	Reserved	-	-	-	-	-
0xF640	Serial communication unit 2 transmission/reception buffer	SD2BUFL	SD2BUF	R/W	8/16	0x00
0xF641		SD2BUFH		R/W	8	0x00
0xF642	Serial communication unit 2 mode register	SU2MOD	-	R/W	8	0x00
0xF643	Reserved	-	-	-	-	-
0xF644	Serial communication unit 2 transmission interval setting register	SU2DLYL	-	R/W	8	0x00
0xF645	Reserved	-	-	-	-	-
0xF646	Serial communication unit 2 control register	SU2CONL	SU2CON	R/W	8/16	0x00
0xF647		SU2CONH		R/W	8	0x00
0xF648	Synchronous serial port 2 mode register	SIO2MODL	SIO2MOD	R/W	8/16	0x00
0xF649		SIO2MODH		R/W	8	0x00
0xF64A	Synchronous serial port 2 status register	SIO2STAT	-	R/W	8	0x00
0xF64B	Reserved	-	-	-	-	-
0xF64C	UART20 mode register	UA20MODL	UA20MOD	R/W	8/16	0x00
0xF64D		UA20MODH		R/W	8	0x00
0xF64E	UART20 baud rate register	UA20BRTL	UA20BRT	R/W	8/16	0xFF
0xF64F		UA20BRTH		R/W	8	0xFF
0xF650	UART20 baud rate adjustment register	UA20BRC	-	R/W	8	0x00
0xF651	Reserved	-	-	-	-	-
0xF652	UART20 status register	UA20STAT	-	R/W	8	0x00
0xF653	Reserved	-	-	-	-	-
0xF654	UART21 mode register	UA21MODL	UA21MOD	R/W	8/16	0x00
0xF655		UA21MODH		R/W	8	0x00
0xF656	UART21 baud rate register	UA21BRTL	UA21BRT	R/W	8/16	0xFF
0xF657		UA21BRTH		R/W	8	0xFF
0xF658	UART21 baud rate adjustment register	UA21BRC	-	R/W	8	0x00
0xF659	Reserved	-	-	-	-	-
0xF65A	UART21 status register	UA21STAT	-	R/W	8	0x00
0xF65B to 0xF65F	Reserved	-	-	-	-	-
0xF660	Serial communication unit 3 transmission/reception buffer	SD3BUFL	SD3BUF	R/W	8/16	0x00
0xF661		SD3BUFH		R/W	8	0x00
0xF662	Serial communication unit 3 mode register	SU3MOD	-	R/W	8	0x00
0xF663	Reserved	-	-	-	-	-
0xF664	Serial communication unit 3 transmission interval setting register	SU3DLYL	-	R/W	8	0x00
0xF665	Reserved	-	-	-	-	-
0xF666	Serial communication unit 3 control register	SU3CONL	SU3CON	R/W	8/16	0x00
0xF667		SU3CONH		R/W	8	0x00
0xF668	Synchronous serial port 3 mode register	SIO3MODL	SIO3MOD	R/W	8/16	0x00
0xF669		SIO3MODH		R/W	8	0x00
0xF66A	Synchronous serial port 3 status register	SIO3STAT	-	R/W	8	0x00
0xF66B	Reserved	-	-	-	-	-
0xF66C	UART30 mode register	UA30MODL	UA30MOD	R/W	8/16	0x00
0xF66D		UA30MODH		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF66E	UART30 baud rate register	UA30BRTL	UA30BRT	R/W	8/16	0xFF
0xF66F		UA30BRTH		R/W	8	0xFF
0xF670	UART30 baud rate adjustment register	UA30BRC	-	R/W	8	0x00
0xF671	Reserved	-	-	-	-	-
0xF672	UART30 status register	UA30STAT	-	R/W	8	0x00
0xF673	Reserved	-	-	-	-	-
0xF674	UART31 mode register	UA31MODL	UA31MOD	R/W	8/16	0x00
0xF675		UA31MODH		R/W	8	0x00
0xF676	UART31 baud rate register	UA31BRTL	UA31BRT	R/W	8/16	0xFF
0xF677		UA31BRTH		R/W	8	0xFF
0xF678	UART31 baud rate adjustment register	UA31BRC	-	R/W	8	0x00
0xF679	Reserved	-	-	-	-	-
0xF67A	UART31 status register	UA31STAT	-	R/W	8	0x00
0xF67B to 0xF67F	Reserved	-	-	-	-	-
0xF680	Serial communication unit 4 transmission/reception buffer	SD4BUFL	SD4BUF	R/W	8/16	0x00
0xF681		SD4BUFH		R/W	8	0x00
0xF682	Serial communication unit 4 mode register	SU4MOD	-	R/W	8	0x00
0xF683	Reserved	-	-	-	-	-
0xF684	Serial communication unit 4 transmission interval setting register	SU4DLYL	-	R/W	8	0x00
0xF685	Reserved	-	-	-	-	-
0xF686	Serial communication unit 4 control register	SU4CONL	SU4CON	R/W	8/16	0x00
0xF687		SU4CONH		R/W	8	0x00
0xF688	Synchronous serial port 4 mode register	SIO4MODL	SIO4MOD	R/W	8/16	0x00
0xF689		SIO4MODH		R/W	8	0x00
0xF68A	Synchronous serial port 4 status register	SIO4STAT	-	R/W	8	0x00
0xF68B	Reserved	-	-	-	-	-
0xF68C	UART40 mode register	UA40MODL	UA40MOD	R/W	8/16	0x00
0xF68D		UA40MODH		R/W	8	0x00
0xF68E	UART40 baud rate register	UA40BRTL	UA40BRT	R/W	8/16	0xFF
0xF68F		UA40BRTH		R/W	8	0xFF
0xF690	UART40 baud rate adjustment register	UA40BRC	-	R/W	8	0x00
0xF691	Reserved	-	-	-	-	-
0xF692	UART40 status register	UA40STAT	-	R/W	8	0x00
0xF693	Reserved	-	-	-	-	-
0xF694	UART41 mode register	UA41MODL	UA41MOD	R/W	8/16	0x00
0xF695		UA41MODH		R/W	8	0x00
0xF696	UART41 baud rate register	UA41BRTL	UA41BRT	R/W	8/16	0xFF
0xF697		UA41BRTH		R/W	8	0xFF
0xF698	UART41 baud rate adjustment register	UA41BRC	-	R/W	8	0x00
0xF699	Reserved	-	-	-	-	-
0xF69A	UART41 status register	UA41STAT	-	R/W	8	0x00
0xF69B to 0xF69F	Reserved	-	-	-	-	-
0xF6A0	Serial communication unit 5 transmission/reception buffer	SD5BUFL	SD5BUF	R/W	8/16	0x00
0xF6A1		SD5BUFH		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF6A2	Serial communication unit 5 mode register	SU5MOD	-	R/W	8	0x00
0xF6A3	Reserved	-	-	-	-	-
0xF6A4	Serial communication unit 5 transmission interval setting register	SU5DLYL	-	R/W	8	0x00
0xF6A5	Reserved	-	-	-	-	-
0xF6A6	Serial communication unit 5 control register	SU5CONL	SU5CON	R/W	8/16	0x00
0xF6A7		SU5CONH		R/W	8	0x00
0xF6A8	Synchronous serial port 5 mode register	SIO5MODL	SIO5MOD	R/W	8/16	0x00
0xF6A9		SIO5MODH		R/W	8	0x00
0xF6AA	Synchronous serial port 5 status register	SIO5STAT	-	R/W	8	0x00
0xF6AB	Reserved	-	-	-	-	-
0xF6AC	UART50 mode register	UA50MODL	UA50MOD	R/W	8/16	0x00
0xF6AD		UA50MODH		R/W	8	0x00
0xF6AE	UART50 baud rate register	UA50BRTL	UA50BRT	R/W	8/16	0xFF
0xF6AF		UA50BRTH		R/W	8	0xFF
0xF6B0	UART50 baud rate adjustment register	UA50BRC	-	R/W	8	0x00
0xF6B1	Reserved	-	-	-	-	-
0xF6B2	UART50 status register	UA50STAT	-	R/W	8	0x00
0xF6B3	Reserved	-	-	-	-	-
0xF6B4	UART51 mode register	UA51MODL	UA51MOD	R/W	8/16	0x00
0xF6B5		UA51MODH		R/W	8	0x00
0xF6B6	UART51 baud rate register	UA51BRTL	UA51BRT	R/W	8/16	0xFF
0xF6B7		UA51BRTH		R/W	8	0xFF
0xF6B8	UART51 baud rate adjustment register	UA51BRC	-	R/W	8	0x00
0xF6B9	Reserved	-	-	-	-	-
0xF6BA	UART51 status register	UA51STAT	-	R/W	8	0x00
0xF6BB to 0xF6BF	Reserved	-	-	-	-	-
0xF6C0	I^2C bus unit 0 mode register	I2U0MSS	-	R/W	8	0x00
0xF6C1	Reserved	-	-	-	-	-
0xF6C2	I^2C bus 0 receive register (master)	I2UM0RD	-	R	8	0x00
0xF6C3	Reserved	-	-	-	-	-
0xF6C4	I^2C bus 0 slave address register (master)	I2UM0SA	-	R/W	8	0x00
0xF6C5	Reserved	-	-	-	-	-
0xF6C6	I^2C bus 0 transmit data register (master)	I2UM0TD	-	R/W	8	0x00
0xF6C7	Reserved	-	-	-	-	-
0xF6C8	I^2C bus 0 control register (master)	I2UM0CON	-	R/W	8	0x00
0xF6C9	Reserved	-	-	-	-	-
0xF6CA	I^2C bus 0 mode register (master)	I2UM0MDL	I2UM0MOD	R/W	8/16	0x00
0xF6CB		I2UM0MDH		R/W	8	0x02
0xF6CC	I^2C bus 0 status register (master)	I2UM0STA	I2UM0STR	R/W	8/16	0x00
0xF6CD		I2UM0ISR		R/W	8	0x00
0xF6CE	I^2C bus 0 receive register (slave)	I2US0RD	-	R	8	0x00
0xF6CF	Reserved	-	-	-	-	-
0xF6D0	I^2C bus 0 slave address register (slave)	I2US0SA	-	R/W	8	0x00
0xF6D1	Reserved	-	-	-	-	-
0xF6D2	I^2C bus 0 transmit data register (slave)	I2US0TD	-	R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
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	I2US0STR	R/W	8/16	0x00
0xF6D9		I2US0ISR		R/W	8	0x00
0xF6DA to 0xF6E1	Reserved	-	-	-	-	-
0xF6E2	I²C master 0 receive register	I2M0RD	-	R	8	0x00
0xF6E3	Reserved	-	-	-	-	-
0xF6E4	I²C master 0 slave address register	I2M0SA	-	R/W	8	0x00
0xF6E5	Reserved	-	-	-	-	-
0xF6E6	I²C master 0 transmit data register	I2M0TD	-	R/W	8	0x00
0xF6E7	Reserved	-	-	-	-	-
0xF6E8	I²C master 0 control register	I2M0CON	-	R/W	8	0x00
0xF6E9	Reserved	-	-	-	-	-
0xF6EA	I²C master 0 mode register	I2M0MODL	I2M0MOD	R/W	8/16	0x00
0xF6EB		I2M0MODH		R/W	8	0x02
0xF6EC	I²C master 0 status register	I2M0STAT	I2M0STR	R/W	8/16	0x00
0xF6ED		I2M0ISR		R/W	8	0x00
0xF6EE	Reserved	-	-	-	-	-
0xF6EF	Reserved	-	-	-	-	-
0xF6F0	Reserved	-	-	-	-	-
0xF6F1	Reserved	-	-	-	-	-
0xF6F2	I²C master 1 receive register	I2M1RD	-	R	8	0x00
0xF6F3	Reserved	-	-	-	-	-
0xF6F4	I²C master 1 slave address register	I2M1SA	-	R/W	8	0x00
0xF6F5	Reserved	-	-	-	-	-
0xF6F6	I²C master 1 transmit data register	I2M1TD	-	R/W	8	0x00
0xF6F7	Reserved	-	-	-	-	-
0xF6F8	I²C master 1 control register	I2M1CON	-	R/W	8	0x00
0xF6F9	Reserved	-	-	-	-	-
0xF6FA	I²C master 1 mode register	I2M1MODL	I2M1MOD	R/W	8/16	0x00
0xF6FB		I2M1MODH		R/W	8	0x02
0xF6FC	I²C master 1 status register	I2M1STAT	I2M1STR	R/W	8/16	0x00
0xF6FD		I2M1ISR		R/W	8	0x00
0xF6FE	Reserved	-	-	-	-	-
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

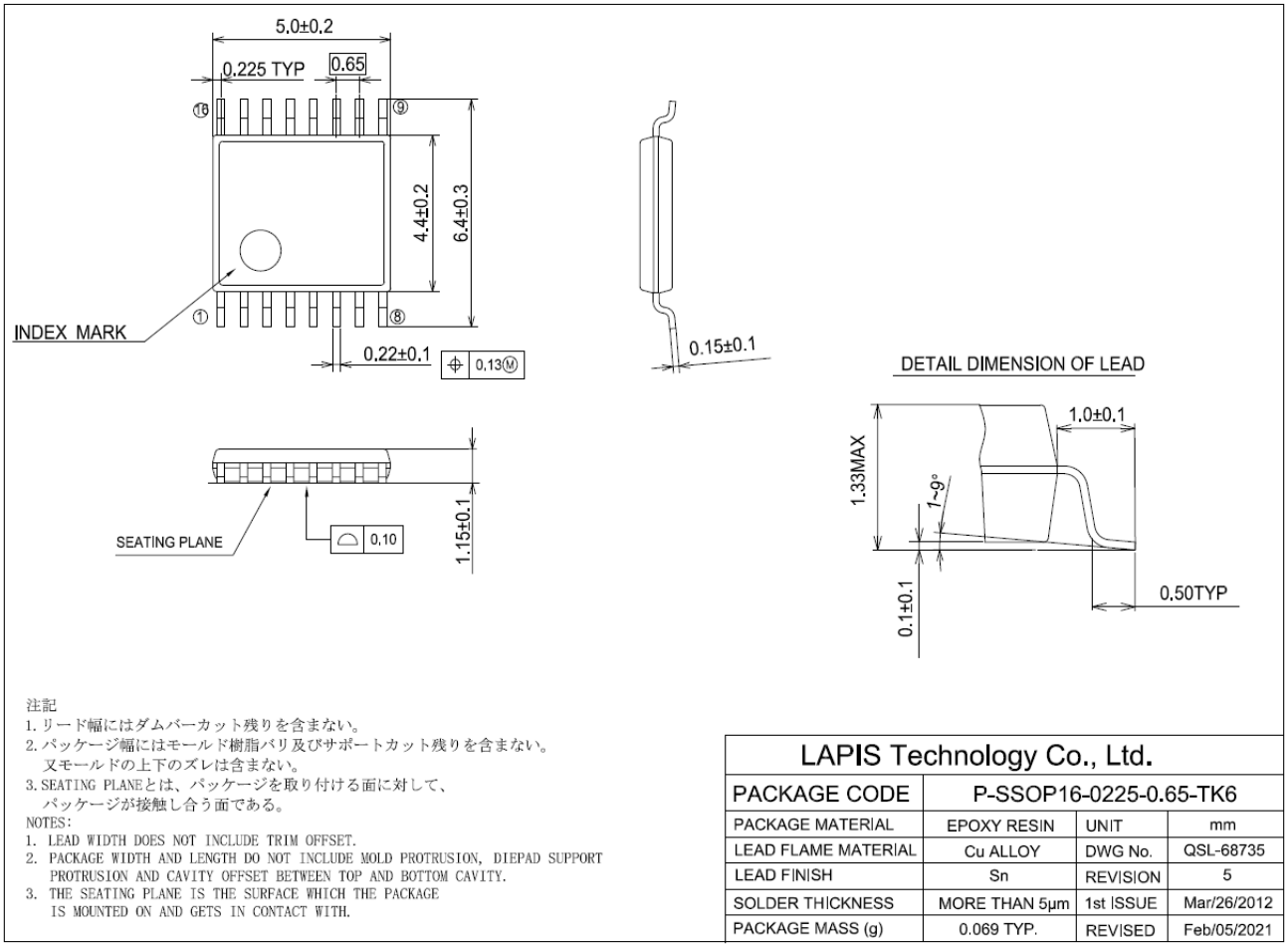
Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF70A	DMA channel 1 transfer count register	DC1TNL	DC1TN	R/W	8/16	0x00
0xF70B		DC1TNH		R/W	8	0x00
0xF70C	DMA channel 1 transfer source address register	DC1SAL	DC1SA	R/W	8/16	0x00
0xF70D		DC1SAH		R/W	8	0x00
0xF70E	DMA channel 1 transfer destination address register	DC1DAL	DC1DA	R/W	8/16	0x00
0xF70F		DC1DAH		R/W	8	0x00
0xF710 to 0xF71F	Reserved	-	-	-	-	-
0xF720	DMA transfer enable register	DCEN	-	R/W	8	0x00
0xF721	Reserved	-	-	-	-	-
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 to 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
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	SA-ADC result register 8	SADR8L	SADR8	R	8/16	0x00
0xF811		SADR8H		R	8	0x00
0xF812	SA-ADC result register 9	SADR9L	SADR9	R	8/16	0x00
0xF813		SADR9H		R	8	0x00
0xF814	SA-ADC result register 10	SADR10L	SADR10	R	8/16	0x00
0xF815		SADR10H		R	8	0x00
0xF816	SA-ADC result register 11	SADR11L	SADR11	R	8/16	0x00
0xF817		SADR11H		R	8	0x00
0xF818	SA-ADC result register 12	SADR12L	SADR12	R	8/16	0x00
0xF819		SADR12H		R	8	0x00
0xF81A	SA-ADC result register 13	SADR13L	SADR13	R	8/16	0x00
0xF81B		SADR13H		R	8	0x00
0xF81C	SA-ADC result register 14	SADR14L	SADR14	R	8/16	0x00
0xF81D		SADR14H		R	8	0x00
0xF81E	SA-ADC result register 15	SADR15L	SADR15	R	8/16	0x00
0xF81F		SADR15H		R	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
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	-	-	-	-	-
0xF831	Reserved	-	-	-	-	-
0xF832	SA-ADC conversion interval setting 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	-	-	-	-	-
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
0xF844 to 0xF847	Reserved	-	-	-	-	-
0xF848	Comparator 1 control register	CMP1CON	-	R/W	8	0x00
0xF849	Reserved	-	-	-	-	-
0xF84A	Comparator 1 mode register	CMP1MODL	CMP1MOD	R/W	8/16	0x00
0xF84B		CMP1MODH		R/W	8	0x00
0xF84C to 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	-	-	-	-	-

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
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 to 0xF85F	Reserved	-	-	-	-	-
0xF860	D/A converter 0 control register	DACCON	-	R/W	8	0x00
0xF861	Reserved	-	-	-	-	-
0xF862	D/A converter 0 code register	DACCODE	-	R/W	8	0x00
0xF863	Reserved	-	-	-	-	-
0xF864		-	-	-	-	-
0xF865		-	-	-	-	-
0xF866		-	-	-	-	-
0xF867		-	-	-	-	-
0xF868	D/A converter 1 control register	DACCON1	-	R/W	8	0x00
0xF869	Reserved	-	-	-	-	-
0xF86A	D/A converter 1 code register	DACCODE1	-	R/W	8	0x00
0xF86B to 0xFFFF	Reserved	-	-	-	-	-

Appendix B Package Dimensions

16pin SSOP



LAPIS Technology 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	5
SOLDER THICKNESS	MORE THAN 5μm	1st ISSUE	Mar/26/2012
PACKAGE MASS (g)	0.069 TYP.	REVISED	Feb/05/2021

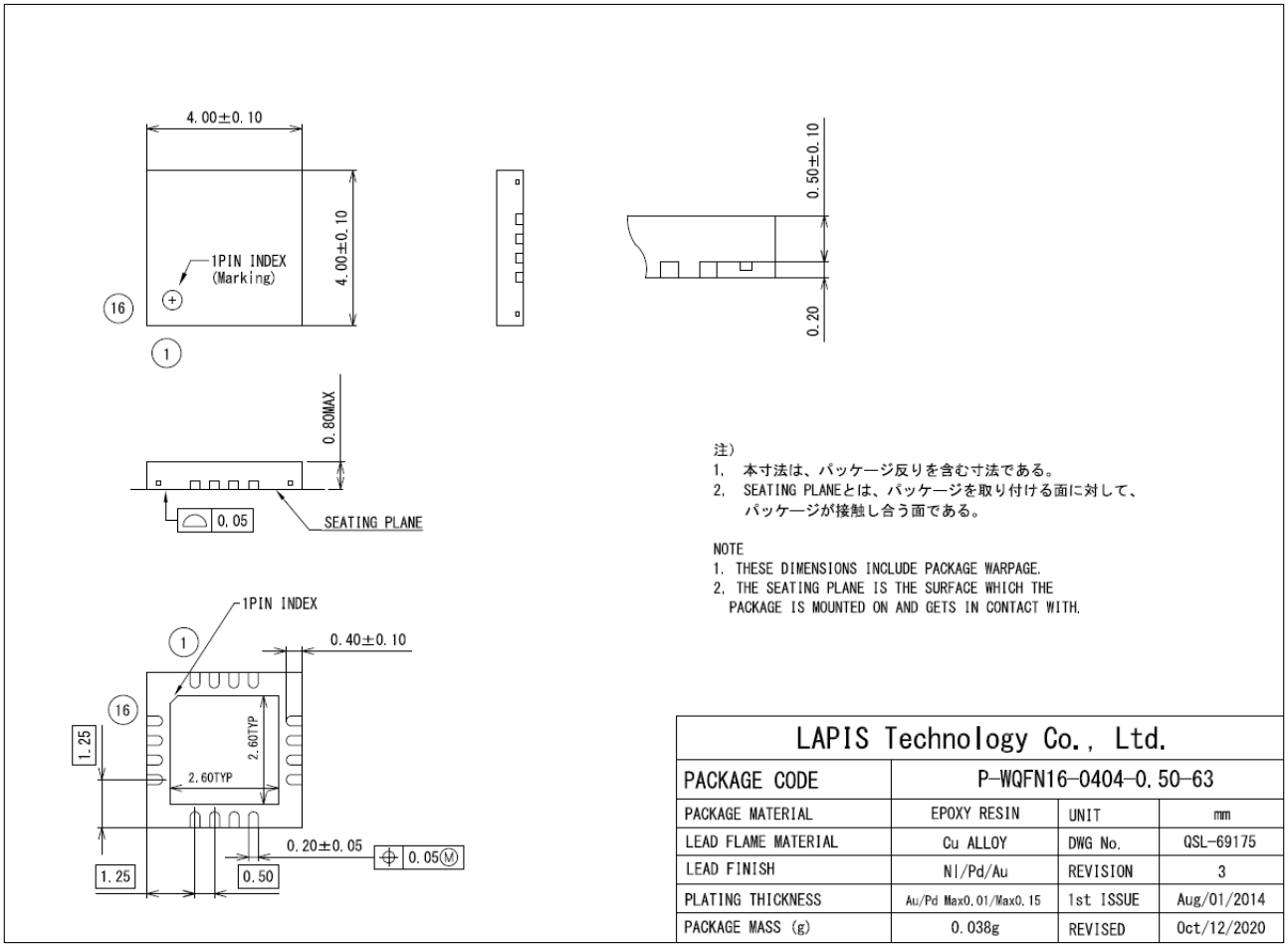
(Unit: mm)

Figure B-1 SSOP16 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

16pin WQFN



LAPIS Technology 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	3
PLATING THICKNESS	Au/Pd Max0.01/Max0.15	1st ISSUE	Aug/01/2014
PACKAGE MASS (g)	0.038g	REVISED	Oct/12/2020

(Unit: mm)

Figure B-2 WQFN16 Package Dimension

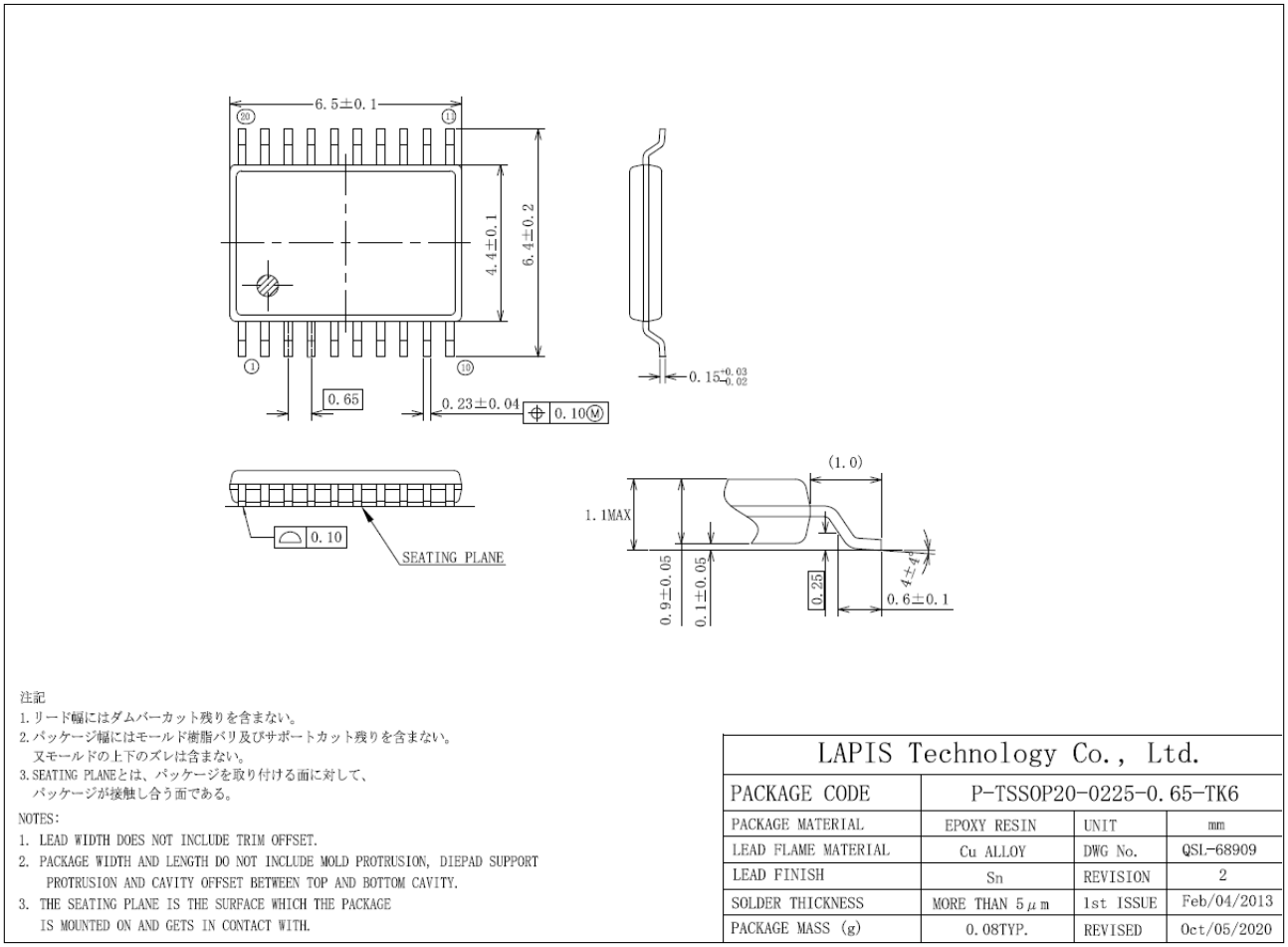
[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

[Note] Notes for the package with exposed die pad

The die pad is exposed on the bottom of WQFN package. Make the die pad electrically open when soldering onto the PCB.

20pin TSSOP



LAPIS Technology 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	2
SOLDER THICKNESS	MORE THAN 5μm	1st ISSUE	Feb/04/2013
PACKAGE MASS (g)	0.08TYP.	REVISED	Oct/05/2020

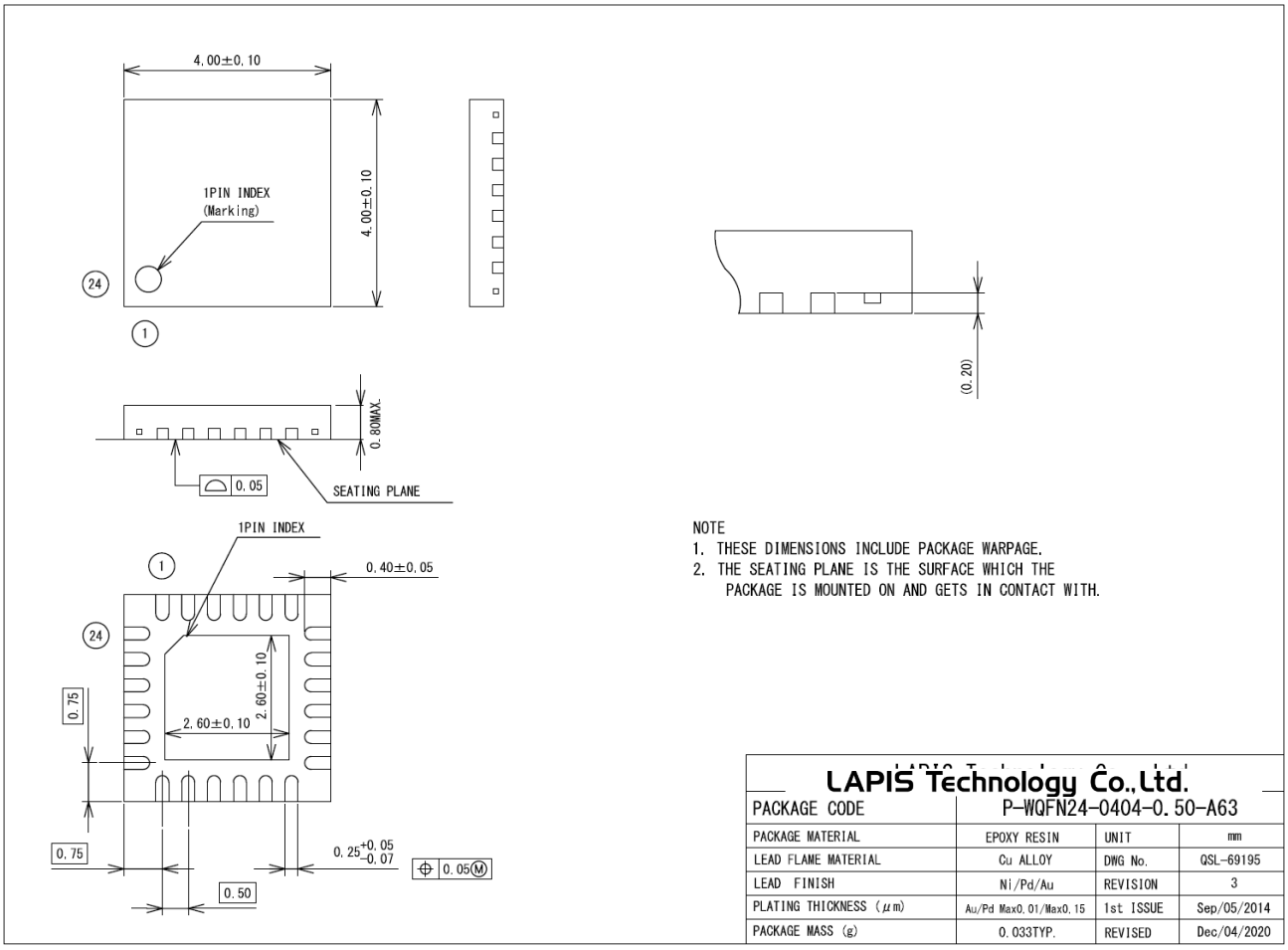
(Unit: mm)

Figure B-3 TSSOP20 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

24pin WQFN



(Unit: mm)

Figure B-4 WQFN24 Package Dimension

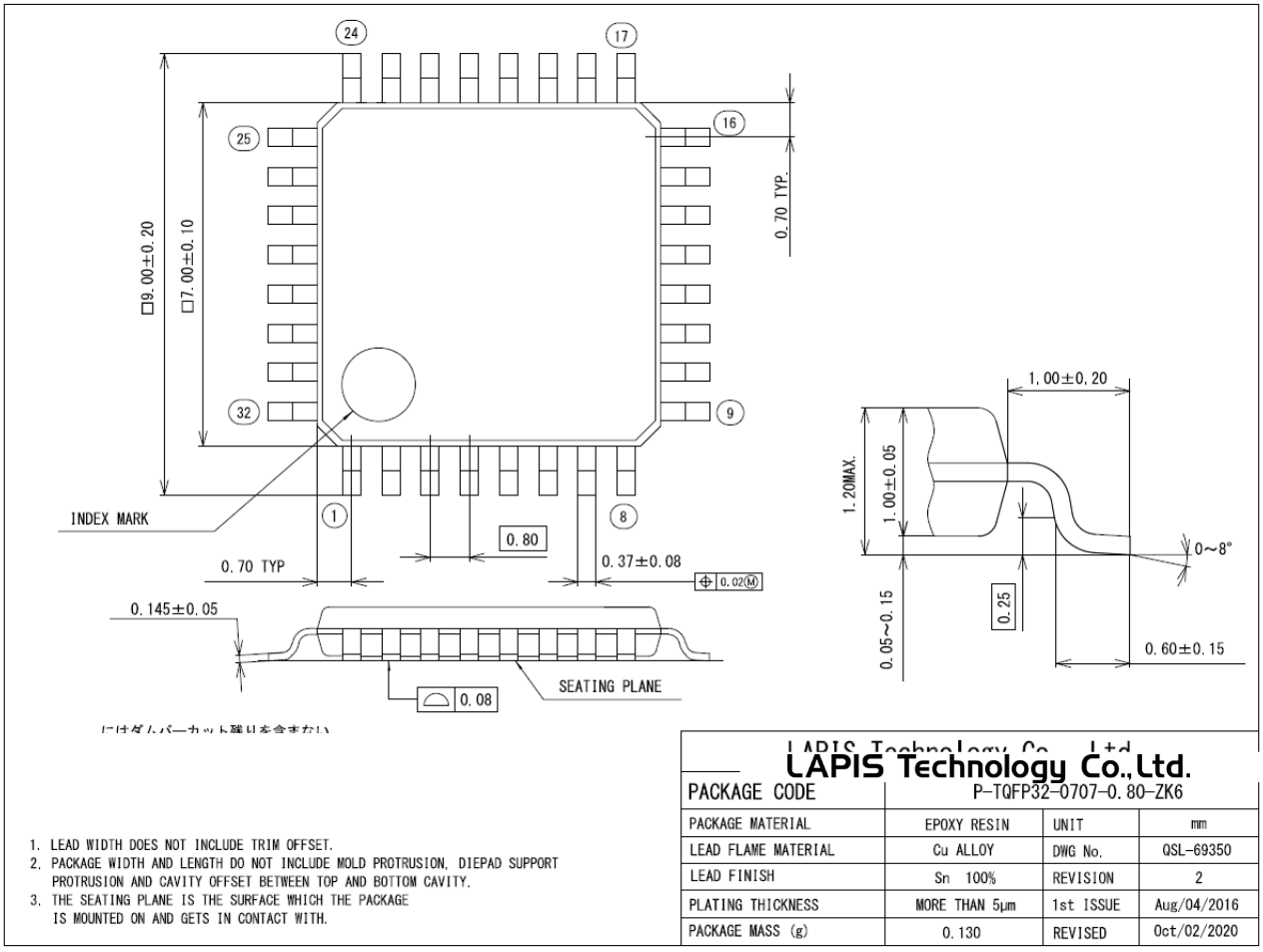
[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

[Note] Notes for the package with exposed die pad

The die pad is exposed on the bottom of WQFN package. Make the die pad electrically open when soldering onto the PCB.

32pin TQFP



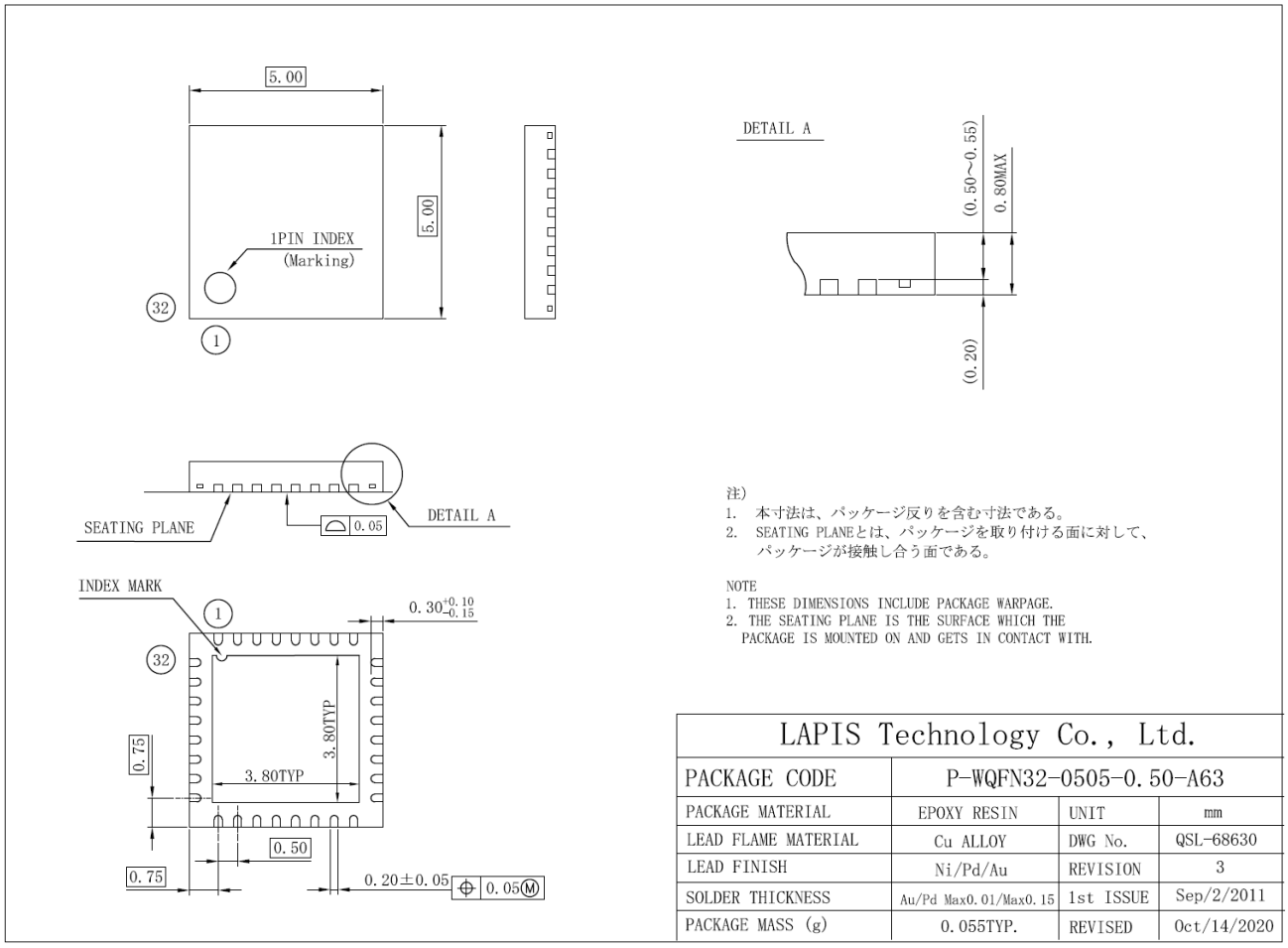
(Unit: mm)

Figure B-5 TQFP32 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

32pin WQFN



LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-WQFN32-0505-0.50-A63		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-68630
LEAD FINISH	Ni/Pd/Au	REVISION	3
SOLDER THICKNESS	Au/Pd Max0.01/Max0.15	1st ISSUE	Sep/2/2011
PACKAGE MASS (g)	0.055TYP.	REVISED	Oct/14/2020

(Unit: mm)

Figure B-6 WQFN32 Package Dimension

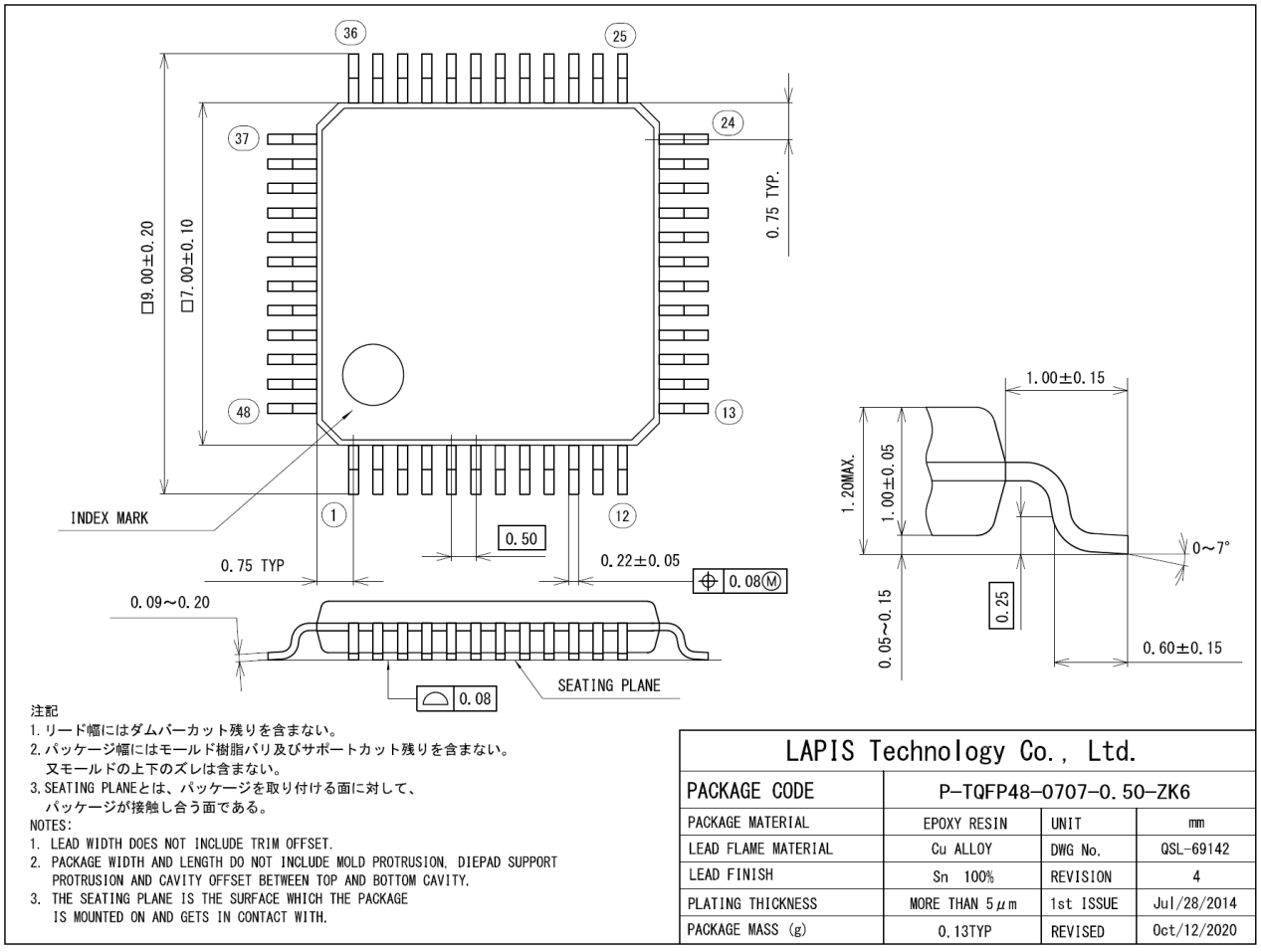
[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

[Note] Notes for the package with exposed die pad

The die pad is exposed on the bottom of WQFN package. Make the die pad electrically open when soldering onto the PCB.

48pin TQFP



LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-TQFP48-0707-0.50-ZK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-69142
LEAD FINISH	Sn 100%	REVISION	4
PLATING THICKNESS	MORE THAN 5μm	1st ISSUE	Jul/28/2014
PACKAGE MASS (g)	0.13TYP	REVISED	Oct/12/2020

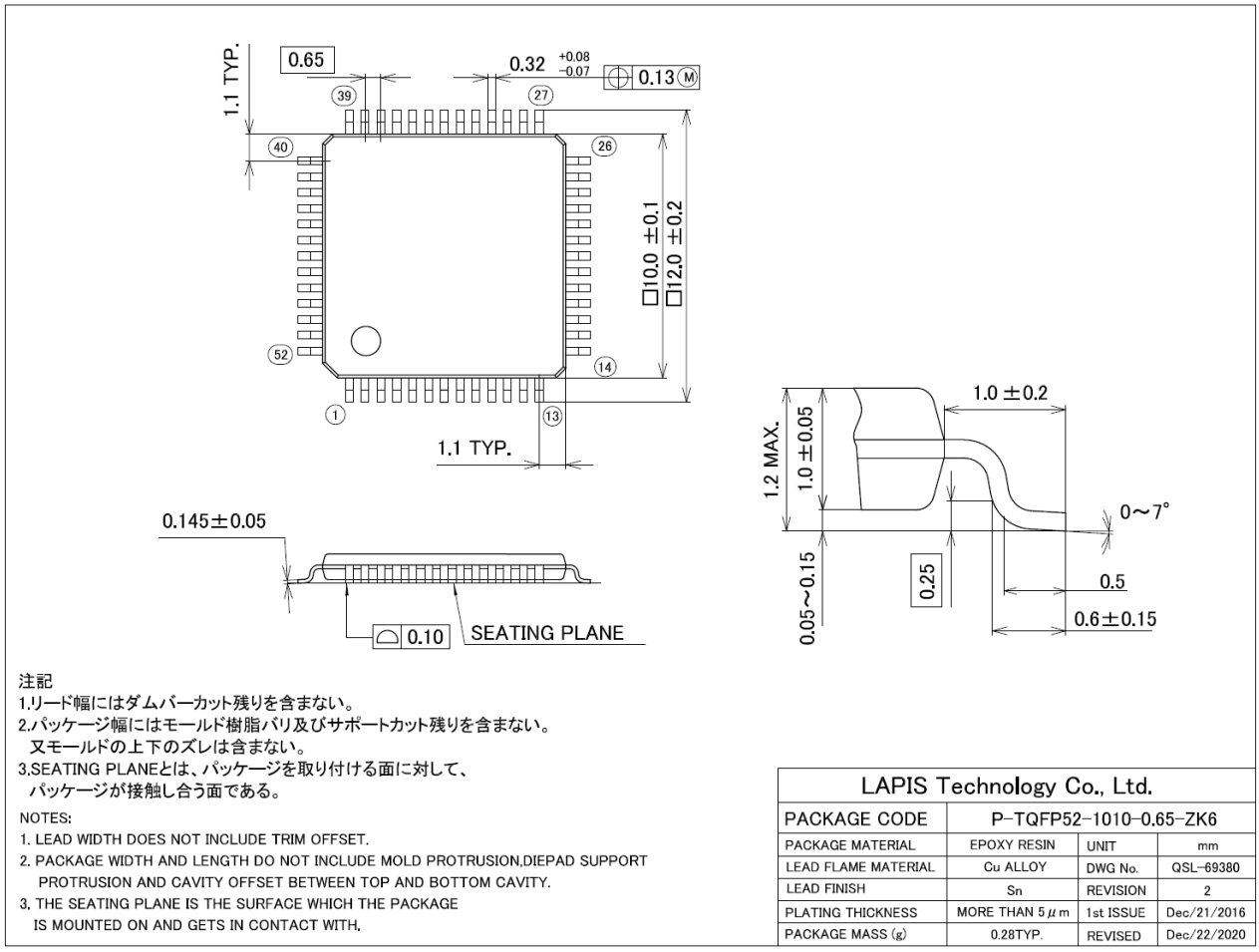
(Unit: mm)

Figure B-7 TQFP48 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

52pin TQFP



注記
1.リード幅にはダムバーカット残りを含まない。
2.パッケージ幅にはモールド樹脂バリ及びサポートカット残りを含まない。
又モールドの上下のズレは含まない。
3.SEATING PLANEとは、パッケージを取り付ける面に対して、
パッケージが接触し合う面である。

NOTES:
1. LEAD WIDTH DOES NOT INCLUDE TRIM OFFSET.
2. PACKAGE WIDTH AND LENGTH DO NOT INCLUDE MOLD PROTRUSION,DIEPAD SUPPORT PROTRUSION AND CAVITY OFFSET BETWEEN TOP AND BOTTOM CAVITY.
3. THE SEATING PLANE IS THE SURFACE WHICH THE PACKAGE IS MOUNTED ON AND GETS IN CONTACT WITH.

LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-TQFP52-1010-0.65-ZK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-69380
LEAD FINISH	Sn	REVISION	2
PLATING THICKNESS	MORE THAN 5 μm	1st ISSUE	Dec/21/2016
PACKAGE MASS (g)	0.28TYP.	REVISED	Dec/22/2020

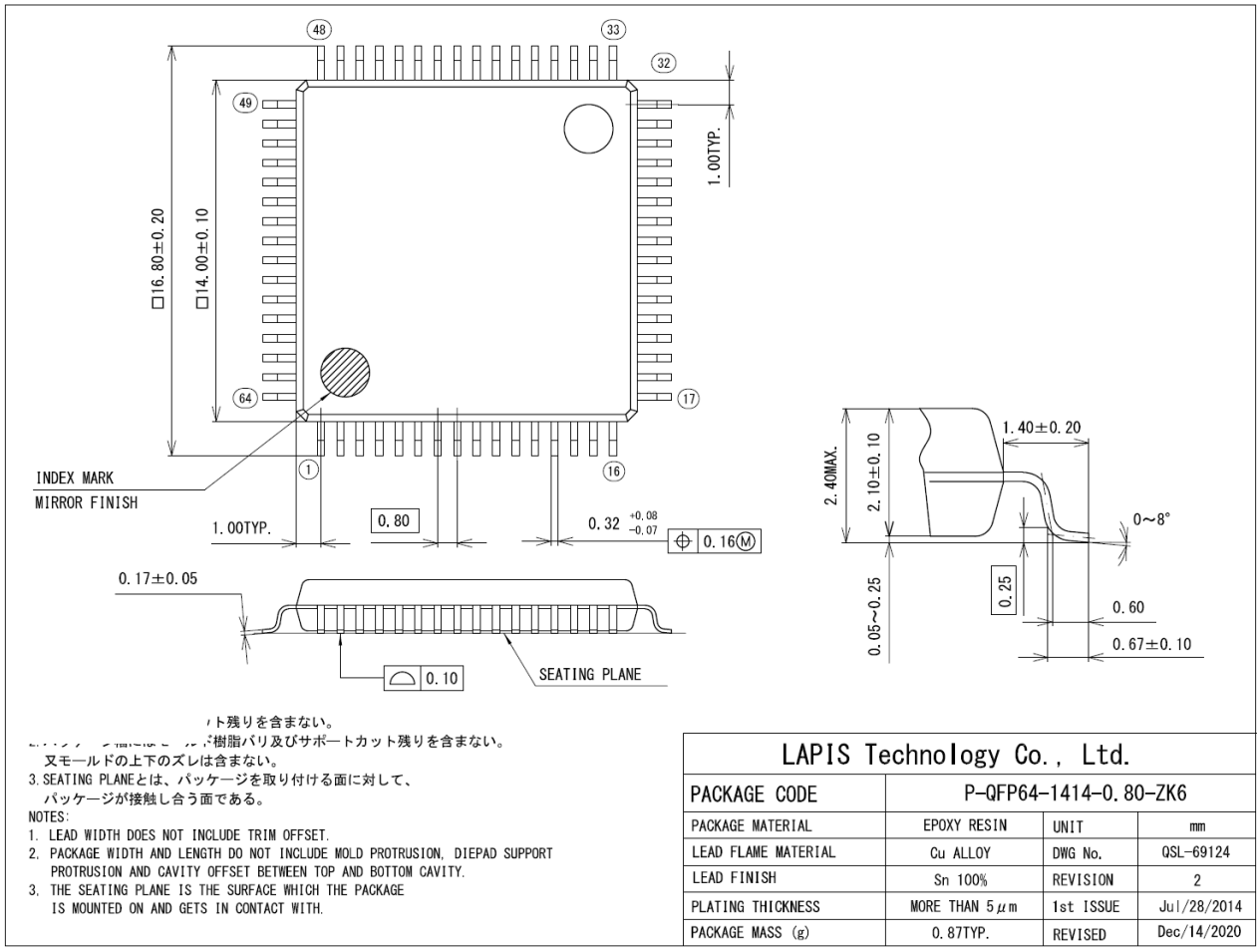
(Unit: mm)

Figure B-8 TQFP52 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

64pin QFP



ト残りを含まない。
ゲート樹脂バリ及びサポートカット残りを含まない。
又モールドの上下のズレは含まない。
3. SEATING PLANEとは、パッケージを取り付ける面に対して、
パッケージが接触し合う面である。
NOTES:
1. LEAD WIDTH DOES NOT INCLUDE TRIM OFFSET.
2. PACKAGE WIDTH AND LENGTH DO NOT INCLUDE MOLD PROTRUSION, DIEPAD SUPPORT PROTRUSION AND CAVITY OFFSET BETWEEN TOP AND BOTTOM CAVITY.
3. THE SEATING PLANE IS THE SURFACE WHICH THE PACKAGE IS MOUNTED ON AND GETS IN CONTACT WITH.

LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-QFP64-1414-0.80-ZK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-69124
LEAD FINISH	Sn 100%	REVISION	2
PLATING THICKNESS	MORE THAN 5μm	1st ISSUE	Jul/28/2014
PACKAGE MASS (g)	0.87TYP.	REVISED	Dec/14/2020

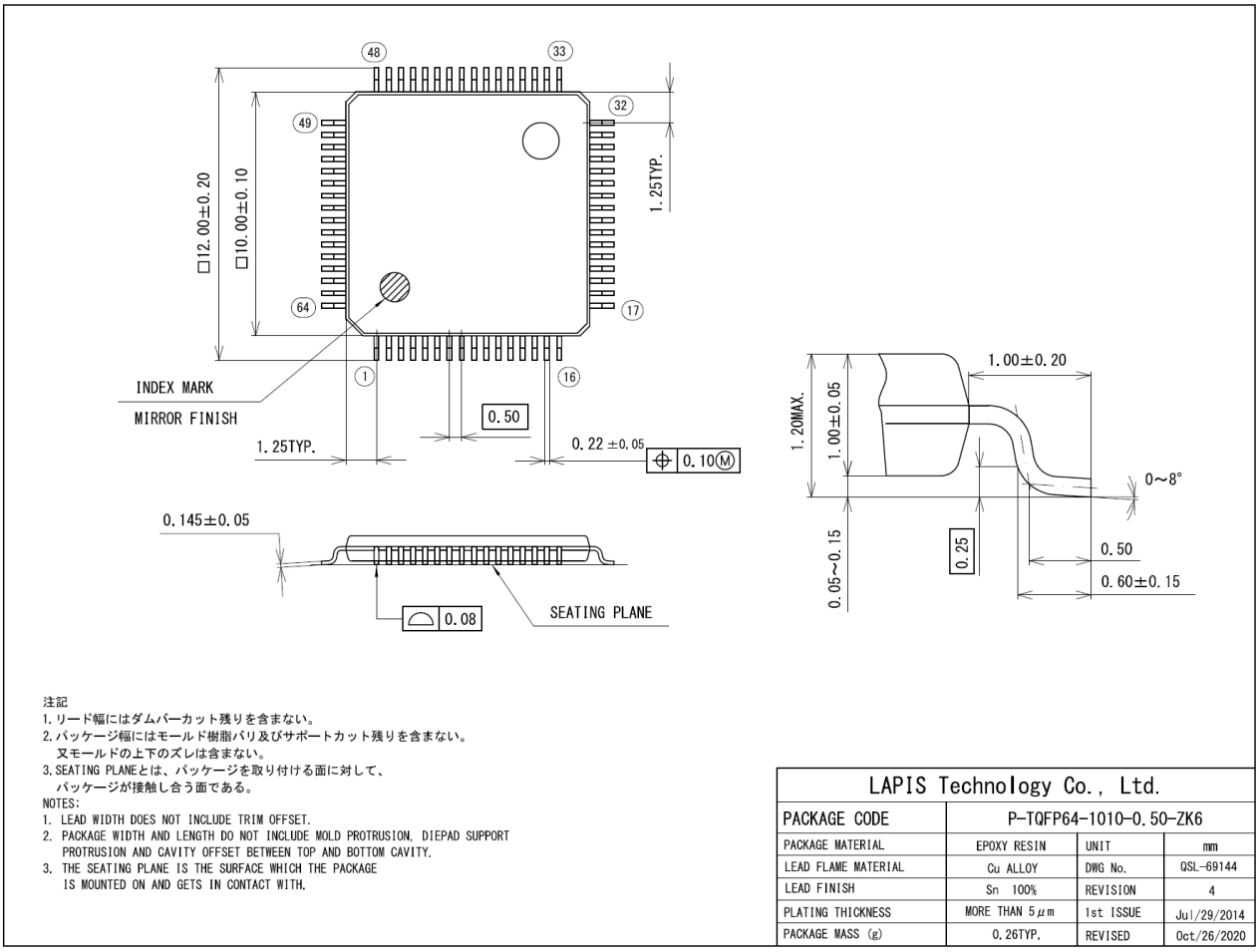
(Unit: mm)

Figure B-9 QFP64 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

64pin TQFP



LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-TQFP64-1010-0.50-ZK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-69144
LEAD FINISH	Sn 100%	REVISION	4
PLATING THICKNESS	MORE THAN 5μm	1st ISSUE	Jul/29/2014
PACKAGE MASS (g)	0.26TYP.	REVISED	Oct/26/2020

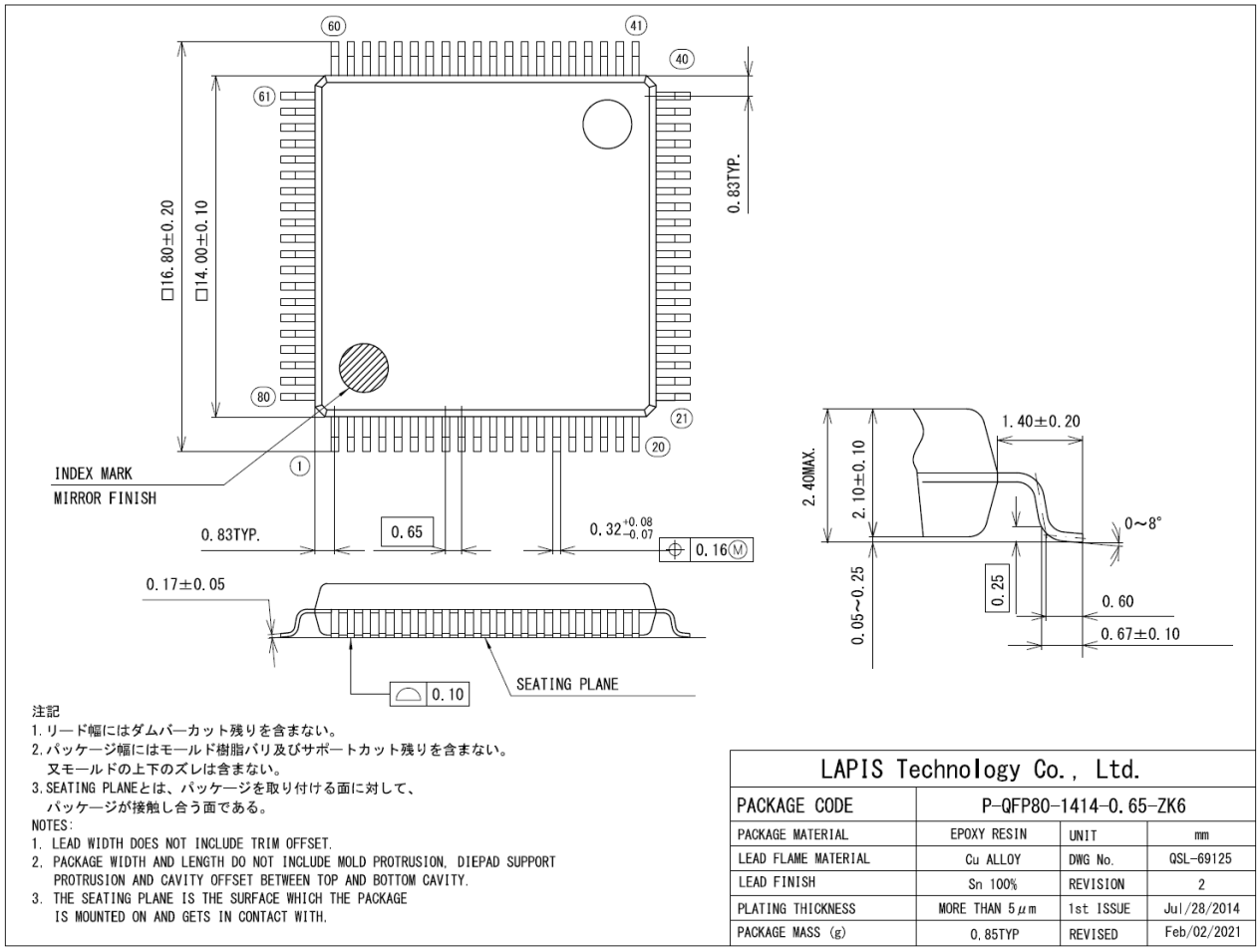
(Unit: mm)

Figure B-10 TQFP64 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

80pin QFP



LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-QFP80-1414-0.65-ZK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-69125
LEAD FINISH	Sn 100%	REVISION	2
PLATING THICKNESS	MORE THAN 5μm	1st ISSUE	Jul/28/2014
PACKAGE MASS (g)	0.85TYP	REVISED	Feb/02/2021

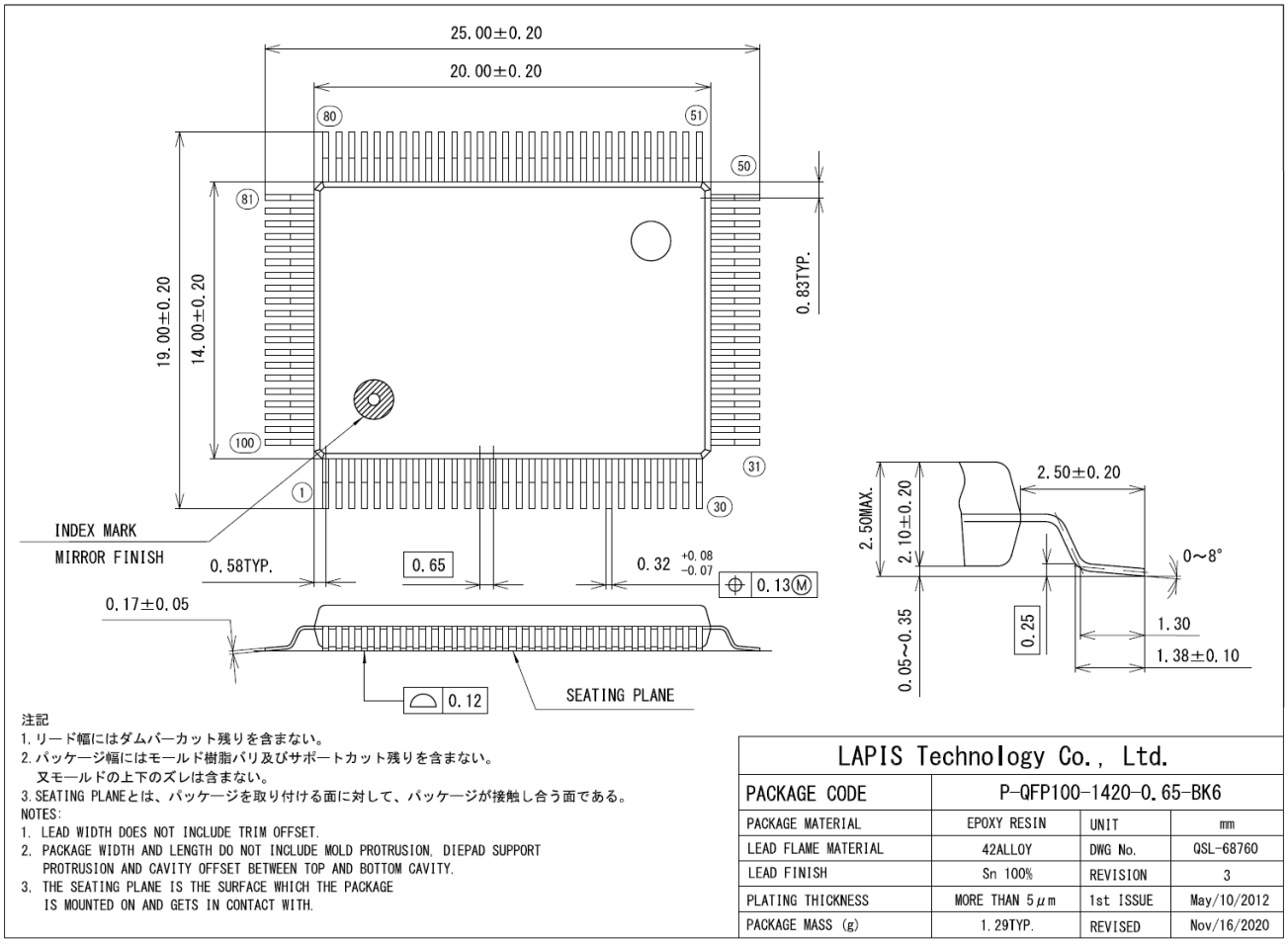
(Unit: mm)

Figure B-11 QFP80 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

100pin QFP



注記
1. リード幅にはダムバーカット残りを含まない。
2. パッケージ幅にはモールド樹脂バリ及びサポートカット残りを含まない。
又モールドの上下のズレは含まない。
3. SEATING PLANEとは、パッケージを取り付ける面に対して、パッケージが接触し合う面である。

NOTES:
1. LEAD WIDTH DOES NOT INCLUDE TRIM OFFSET.
2. PACKAGE WIDTH AND LENGTH DO NOT INCLUDE MOLD PROTRUSION, DIEPAD SUPPORT PROTRUSION AND CAVITY OFFSET BETWEEN TOP AND BOTTOM CAVITY.
3. THE SEATING PLANE IS THE SURFACE WHICH THE PACKAGE IS MOUNTED ON AND GETS IN CONTACT WITH.

LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-QFP100-1420-0.65-BK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	42ALLOY	DWG No.	QSL-68760
LEAD FINISH	Sn 100%	REVISION	3
PLATING THICKNESS	MORE THAN 5μm	1st ISSUE	May/10/2012
PACKAGE MASS (g)	1.29TYP.	REVISED	Nov/16/2020

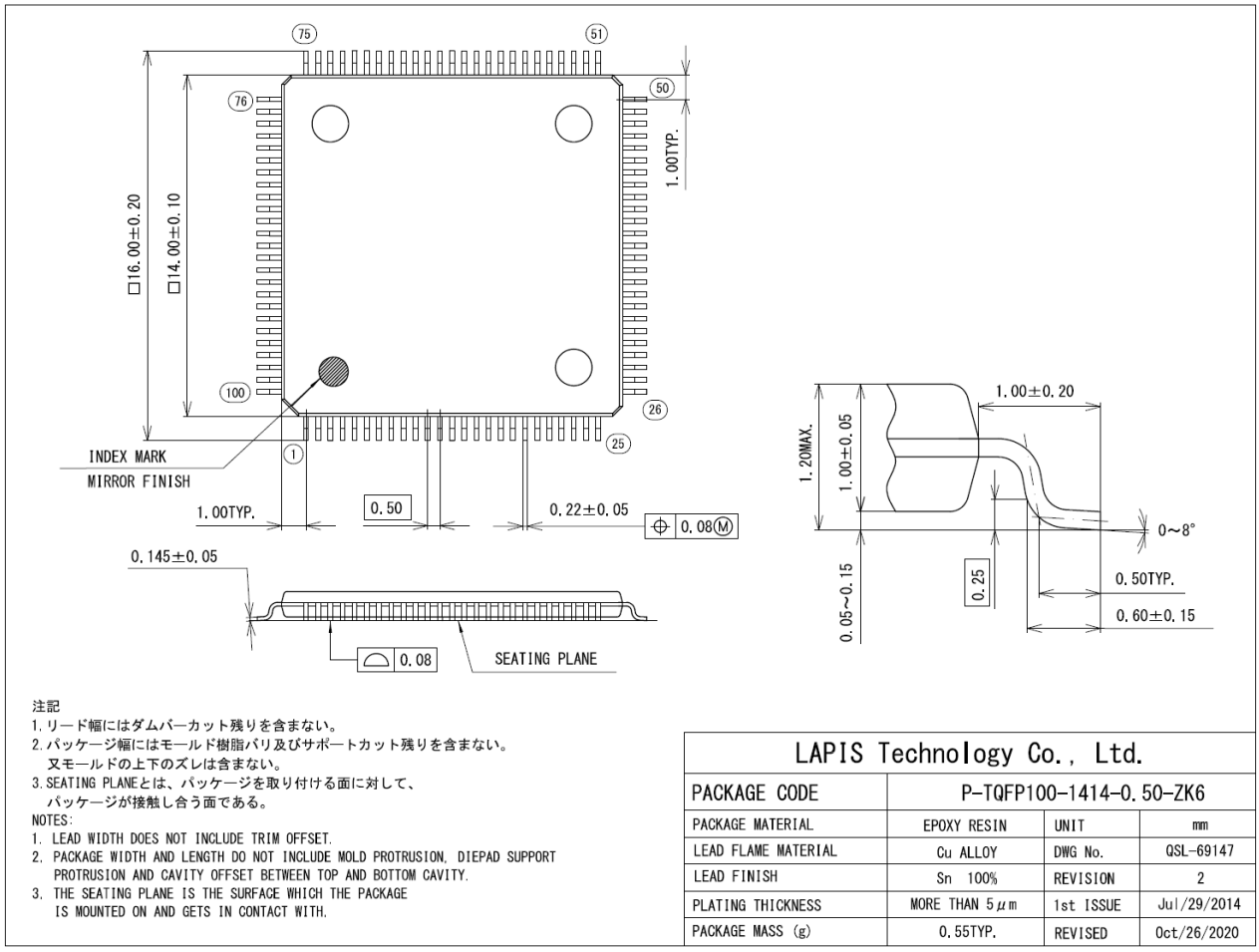
(Unit: mm)

Figure B-12 QFP100 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

100pin TQFP



注記
1. リード幅にはダムバーカット残りを含まない。
2. パッケージ幅にはモールド樹脂バリ及びサポートカット残りを含まない。
又モールドの上下のズレは含まない。
3. SEATING PLANEとは、パッケージを取り付ける面に対して、
パッケージが接触し合う面である。

NOTES:
1. LEAD WIDTH DOES NOT INCLUDE TRIM OFFSET.
2. PACKAGE WIDTH AND LENGTH DO NOT INCLUDE MOLD PROTRUSION, DIEPAD SUPPORT PROTRUSION AND CAVITY OFFSET BETWEEN TOP AND BOTTOM CAVITY.
3. THE SEATING PLANE IS THE SURFACE WHICH THE PACKAGE IS MOUNTED ON AND GETS IN CONTACT WITH.

LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-TQFP100-1414-0.50-ZK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-69147
LEAD FINISH	Sn 100%	REVISION	2
PLATING THICKNESS	MORE THAN 5μm	1st ISSUE	Jul/29/2014
PACKAGE MASS (g)	0.55TYP.	REVISED	Oct/26/2020

(Unit: mm)

Figure B-13 TQFP100 Package Dimension

[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

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 instruction execution cycles are different each other.

CPU Operation Mode	Description
No wait mode	There is no increase of the instruction execution cycle, as there is no wait cycle for reading the program memory during the instruction execution.
Wait mode	There are some increases of the instruction execution cycle, as there are some wait cycles for reading the program memory during the instruction 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 there is no memory access during the instruction execution. See "Example of Instruction execution cycle" for details on how to read the table.

Example of Instruction execution cycle

(1)			(2)-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 execution cycle of each instruction.
The values in column (2)-1 are execution cycles in no wait mode.
The values in column (2)-2 are execution cycles in wait mode.

(3) Additional execution cycle when the instruction refers to ROM.
The values in column (3)-1 are minimum cycles for reading when the instruction refers to ROM.
The values in column (3)-2 are execution cycles that added waiting cycle into the values in (3)-1.

(4) Additional execution cycle when the instruction reads the address allocated in segment 1 or larger.
One cycle is added in spite of the CPU operating mode.
For more details, see the section 1.3.4 "DSR Prefix Instructions" in the nX-U16/100 core instruction manual.

(5) Additional execution effected by the instruction with the [EA+] addressing.
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 / 2 (*1)	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 / 2 (*1)	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 / 2 (*1)	-	-	-	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 / 2 (*1)	-	-	-	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	-	-	-	-

(*1) When the immediately preceding instruction is for reading the data memory or not (not the instruction for reading the data memory / the instruction for reading the data memory)

Control Register Access Instructions

Instruction			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)	4 / 5 (*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) When the 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 / 7 (*1)	-	-	-	1
BL	*Cadr*	2	6	-	-	-	1
	ER*n*	2	6 / 7 (*1)	-	-	-	1

(*1) When the immediately preceding instruction is for reading the data memory or not (not the instruction for reading the data memory / the instruction for reading the data memory)

Conditional Relative Branch Instructions

Instruction		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 / 2(*1)	1 / 7(*1)	-	-	-	1
BLT	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BGT	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BLE	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BGES	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BLTS	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BGTS	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BLES	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BNE	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BEQ	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BNV	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BOV	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BPS	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BNS	*Radr*	1 / 2(*1)	1 / 7(*1)	-	-	-	1
BAL	*Radr*	2	7	-	-	-	1

(*1) When the branch condition is matched or not (Not matched / Matched)

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	-	-	-	-

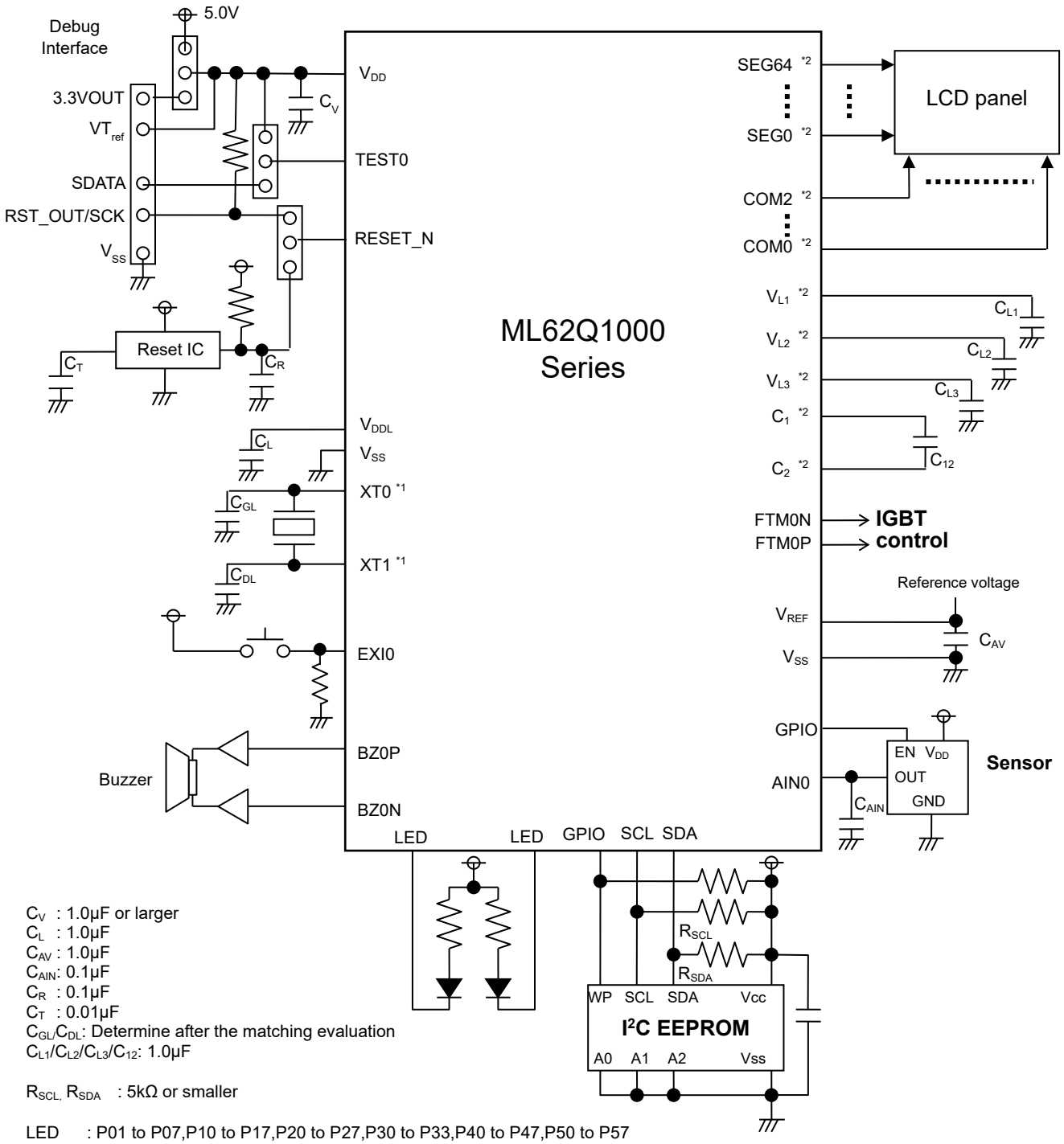
Interrupts

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



C_V : 1.0µF or larger
C_L : 1.0µF
C_{AV} : 1.0µF
C_{AIN}: 0.1µF
C_R : 0.1µF
C_T : 0.01µF
C_{GL}/C_{DL}: Determine after the matching evaluation
$C_{L1}/C_{L2}/C_{L3}/C_{12}$: 1.0µF

R_{SCL}, R_{SDA} : 5kΩ or smaller

LED : P01 to P07,P10 to P17,P20 to P27,P30 to P33,P40 to P47,P50 to P57
P60 to P67,P70 to P77,P80 to P87,P90 to P97,PA0 to PA7,PB0 to PB7

SCL : I2CU0_SCL/I2CM0_SCL
SDA : I2CU0_SDA/I2CM0_SDA

Reset IC : BU4217 (ROHM, Nch open drain output)

*1 : Available on ML62Q1500/ML62Q1800 group and ML62Q1700
*2 : Available on ML62Q1700 group

[Note]

- **Place the capacitor for V_{DDL} pin as close to the LSI power pins as possible.**

Appendix E. List of Notes

This Check List has important notes to prevent commonly-made programming mistakes and frequently overlooked or misunderstood hardware specifications of the LSI. Check each note listed in chapter by chapter when coding or evaluating the program.

Common to all Chapters

[ ] Word access is available for registers with the word symbol. Specify an even address for the word access. See "List of Registers" in each chapter.

[ ] Registers for unequipped channels are not available to use. They return 0x0000 for reading. See "List of Registers" in each chapter.

Chapter 1 Overview

See Section 1.3.4 "TERMINATION OF UNUSED PINS".

[ ] Terminate unused input pins according to the table 1-11 in order to avoid unexpected through-current in the pins.

Chapter 2 CPU and Memory Space

See Section 2.3.2 "List of Coprocessor General-purpose Registers".

[ ] Registers CR10 to CR14 have no function. Reading them returns "0x00". These registers are not writable. See Section 2.3.2.1 "A, B, C, D Registers (CR0 to CR7)".

[ ] "-" indicates that the previous value is retained.

[ ] In a signed operation, each of the most significant bits of input and output is a sign.

See Section 2.5 "Program Memory Space".

[ ] CSR[3] is unused on the ML62Q1000 series. The data of CSR "0x8 to 0xF" are handled as "0x0 to 0x7".

[ ] The Code Option area (64 bytes) is not available for the program code area. For details of Code Option Settings, see Chapter 26 "Code Option" and make sure the setting data is correct.

[ ] It is recommended to fill unused areas with data "0xFFFF" (BRK instruction) in the program memory space to ensure failsafe using Program Development Support Software. See Program Development Support Software User's Manual for details on how to use. See "nX-U16/100 Core Instruction Manual" for details of the BRK instruction.

[ ] Do not read or program unused areas to prevent the CPU works incorrectly.

See Section 2.6 "Data Memory Space".

[ ] The contents of the RAM area are undefined at power-on and system reset. Initialize this area by the software.

[ ] Do not read/write unused areas to prevent the CPU works incorrectly.

See Section 2.7.2 "Data Segment Register (DSR)".

[ ] Reading the content in unused areas except for data segments 31 returns "0xFF".

[ ] Reading the content in unused areas of data segment 31 (data flash) returns "0x00".

See Section 2.8.2 "Software Remap".

[ ] If the entire LSI is reset through RESET_N terminal reset, etc., the remapping function is disabled as the REMAPADD register is restored with the initial value.

Chapter 3 Reset Function

See Section 3.3.1 "Operation of Reset Function".

[ ] The voltage level supervisor function is only initialized at pin reset or POR.

[ ] The BRK instruction reset only initializes the CPU if ELEVEL is 2 or higher. Peripheral circuits and other circuits are not initialized. Use the pin reset or the watchdog timer (WDT) reset to surely initialize the LSI when an abnormality is detected.

[ ] In the Blockreset and SOFTR reset, only the corresponding peripheral circuits are initialized. The CPU and other circuits are not initialized, and not transferred to the system reset mode.

See Section 3.3.2 "System Reset Mode".

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

See Section 3.3.4 "Power-on Reset".

[ ] Rise the V_{DD} to V_{INIT} or higher when powering on.

[ ] When using high-speed clock, keep V_{DD} higher than V_{INIT} until the high-speed clock oscillation is enabled.

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

Chapter 4 Power Management

See 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.

See 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 condition of MIE flag of the program status word (PSW) is "0", it cancels the standby mode only and the CPU does not go to the interrupt routine. For more details about MIE flag, see "nX-U16/100 Core Instruction Manual".

[ ] Insert two NOP instructions in the next to the instruction of that sets HLT, STP, HLTH and STPD bit to "1". The operation without the two NOP instructions is not guaranteed.

[ ] 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 run mode.

[ ] When choosing the low-speed crystal oscillation clock or low-speed external clock (LOSCM[1:0] of FLMOD register is set to "01" or "11") as the LSCLK, switch the SYSTEMCLK to the low-speed clock before setting STP bit or STPD bit.

See Section 4.2.3 "Standby Control Register (SBYCON)" and Section 4.3.6 "Note on Return Operation from Standby Mode".

[ ] When the CPU operation mode is "Wait mode", the PLL reference frequency is 24MHz and the MIE bit is "0", choose 12MHz or slower as the SYSTEMCLK before entering the HALT/HALT-H modes.

[ ] Set disable an interrupt of CPU:MIE=0 and choose 16MHz or slower as the SYSTEMCLK, before entering the STOP/STOP-D modes.

See Section 4.2.5 "Software Reset Control Register (SOFTRCON)".

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

See Section 4.2.6 "Block Clock Control Register 0 (BCKCON0)" to Section 4.2.13 "Block Reset Control Register 3 (BRECON3)".

[ ] To restart the operation of the peripheral circuits, reset them at first by the block reset control register (BRECONn) and then cancel the reset after enabling the clock supply by the block clock control register (BCKCONn).

See Section 4.2.8 "Block Clock Control Register 2 (BCKCON2)".
See Section 4.2.12 "Block Reset Control Register 2 (BRECON2)".

[ ] The DCKACC/RSEACC bit can be set to "1" when the multiplication/division library "muldivu8.lib" is not specified.

[ ] The RSEACC bit can be set to "1" when the multiplication/division library "muldivu8.lib" is not specified.

See Section 4.3.6 "Note on Return Operation from Standby Mode".

[ ] The operation of returning the standby mode is caused by the interrupt level (ELEVEL) of the program status word (PSW), master interrupt enable flag (MIE), the contents of the register (IE0 to IE7), non-maskable interrupt, or maskable interrupt.

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

See Section 4.3.7 "Operation of Each Function in Standby Mode".

[ ] If SYSTEMCLK is switched to high-speed after the STOP/STOP-D mode is released and before the high-speed clock wake-up time passes, the CPU must wait to run the program because the clock supply is suspended until the end of the wake-up time.

[ ] If peripheral circuits need to work in the HALT-H mode, choose the low-speed clock as the operating clock.

[ ] When the FHWUPT register is set to "0x00", the PLL output clock is masked for approx.2.5 ms. HSCLK will be supplied after the elapse of 2.5 ms. If the high-speed clock is selected as SYSTEMCLK, the SYSTEMCLK is stopped for the time period.
[ ] When the FHWUPT register is set to "0x01", the frequency of PLL oscillation clock gradually increases from approx. 1 MHz after the elapse of the wake-up time chosen by the FHWUPT register and reaches the target frequency (16 MHz/24 MHz) chosen by the code option before approx. 2 ms elapse. The PLL oscillation clock during this time period can be used for the SYSTEMCLK, however, accuracy of the frequency is not guaranteed.

See Section 4.3.8 "Block Control Function".
[ ] If the clock supply is only stopped without resetting each peripheral circuit using the block control function, it may cause the output levels of the timer, communication and buzzer pins to be fixed, causing the excess current to flow. Also, in the successive approximation type A/D converter, D/A converter and analog comparator, the circuits may stop their function with the current kept flowing. Stop clock supply in the appropriate state of each circuit. It is recommended to stop the clock with the reset established using the BRECONn register.

See Section 4.3.9 "Examples of entering to STOP/STOP-D mode".
[ ] A next instruction of SWI instruction may get executed before interrupt processing as trigger for waking up. Insert two NOP instructions in the next to the SWI instruction.

Chapter 5 Interrupts
See Section 5.2.6 "Interrupt Request Register 01(IRQ01)".
See Section 5.2.7 "Interrupt Request Register 23(IRQ23)".
See Section 5.2.8 "Interrupt Request Register 45(IRQ45)".
See Section 5.2.9 "Interrupt Request Register 67(IRQ67)".
[ ] There is a risk of clearing other request flags of IRQ01/IRQ23/IRQ45/IRQ67 register, if writing to the specific bit of this register. Use the bit symbol to write to the specific bit. (See Section [5.3.8 Writing to IRQ01/IRQ23/IRQ45/IRQ67] for more detail.)
[ ] Re-request interrupt by writing "1" to EEIR bit of EEITNTC register after writing to IRQ01/IRQ23/IRQ45/IRQ67 register, when expanded external interrupt is enabled.

See Section 5.2.10 "Interrupt Level Control Enable Register (ILEN)".
[ ] Disable the interrupt level control function by resetting the ILE bit to "0" after resetting the Interrupt level control register 0 to 7 (ILC0 to ILC1) to "0x0000" and confirming the current interrupt request level register (CIL) is "0x00" when the interrupt is disabled(IE0 to IE7 registers are "0x00").
[ ] Enable the interrupt level control function by setting the ILE bit to "1" when the interrupt is disabled(IE0 to IE7 registers are "0") or master interrupt enable flag(MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.

See Section 5.2.12 "Interrupt Level Control Register 0 (ILC0)" to Section 5.2.19 "Interrupt Level Control Register 7 (ILC7)".
[ ] Write to this register when the interrupt is disabled (IE0 to IE7 registers are "0x00") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.

See Section 5.3 "Description of Operation".
[ ] The WDT interrupt (WDTINT) is a non-maskable interrupt. If the non-maskable interrupt occurs while an interrupt processing is in progress, abort the interrupt processing and proceed with processing the non-maskable interrupt preferentially regardless of multiple interrupts enabled/disabled.
[ ] For failsafe, define unused all interrupt vectors. If an unused interrupt occurs, it may indicate the possibility that the CPU went out of control. It is recommended to cause the watch dog timer (WDT) reset to occur using the infinite loop to initialize the LSI.

See Section 5.3.4 "Notes on Interrupt Routine (with Interrupt Level Control Disabled)".
[ ] Writing "0" to the ILE bit of the interrupt level control enable register (ILEN) causes the interrupt level control to be disabled.
[ ] Do not enable interrupts in a subroutine called from an interrupt routine for which multiple interrupts are disabled. Otherwise, the program may run out of control when multiple interrupts occur.

See Section 5.3.5 "Flow Charts When Interrupt Level Control Is Enabled".
[ ] For processing of non-maskable interrupt, follow the flow chart "When multiple interrupts are enabled". Registers that should be saved in the stack are ELR2 and EPSW2.

[ ] When programming in C, it is not required to write program codes for saving/restoring registers because they are generated in the C compiler. However, program codes for enabling/disabling interrupts through EI and DI instructions and for writing to the current interrupt level management register (CIL) must be written. See Section 5.3.6 "How To Write Interrupt Processing When Interrupt Level Control Enabled" for the specific program description.

See Section 5.3.6.1 "Description of Interrupt Function to Disable Multiple Interrupts".

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

Chapter 6 Clock Generation Circuit

See Section 6.1.2 "Configuration".

[ ] After the power-on or the system reset, LSCLK (32.768 kHz) is initially chosen as SYSTEMCLK.

See 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 4MHz, 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}.

$1.6V \leqq V_{DD} < 1.8V$: Choose 4 MHz or lower

$1.8V \leqq V_{DD} \leqq 5.5V$: Choose 12 MHz or lower

See Section 6.2.3 "Low-speed Clock Mode Register (FLMOD)".

[ ] Do not change the LOSCM1 bit and LOSCM0 bit when the ENOSC bit in Clock Control Register (FCON) is "1", otherwise the operation is not guaranteed.

See Section 6.2.4 "Clock Control Register (FCON)".

[ ] ENOSC bit and SELSCLK bit are forcibly set to "1" after releasing the HALT-H mode.

See Section 6.2.6 "See Section 6.2.4 "Clock Control Register (FCON)".

[ ] Writing "0" to the LOSCB bit is invalid.

[ ] Insert two NOP instructions in the next to the instruction of that writes "1" to the LOSCB bit, then ensure to confirm the LOSCB bit is reset to "0" after that.

[ ] When switching to the low-speed crystal oscillation clock or low-speed external clock, ensure to use the interrupt referring to the Section 6.3.5 "Switching the Low-speed Clock".

See Section 6.2.7 "Backup Clock Status Register (FBUSTAT)".

[ ] In case the LOSCS bit gets to "1" after the LOSCB bit is set to "1", immediately set the mode back to "Low-speed RC oscillation clock" by resetting the LOSCM1-0 of FLMOD register to "00" and handle it appropriately for the application.

[ ] Refer to the Section 6.3.5 "Switching the Low-speed Clock" to control the LOSCS bit.

See Section 6.2.9 "Clock Backup Test Mode (FBTCON)".

[ ] Use the clock backup test mode after setting the low-speed crystal oscillation clock mode.

See Section 6.2.10 "Low-Speed RC Oscillation Frequency Adjustment Register (LRCADJ)".

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

See Section 6.3.1.2 "Configuration of Low-Speed Crystal Oscillation Circuit".

[ ] Place the crystal resonator as close to the LSI as possible and make sure that signals causing noise and power supply wiring are not near the crystal and its wiring.

[ ] Note that oscillation may stop due to condensation.

[ ] When switching to the low speed crystal oscillation clock, ensure to use the interrupt referring to the Section 6.3.5 "Switching the Low-speed Clock".

[ ] When using the low-speed crystal oscillation clock and choosing the high-speed clock for the system clock, switch the system clock to the low-speed clock before entering the STOP/STOP-D mode.

See Section 6.3.4 "Switching of System Clock".

[ ] When the voltage of V_{DD} is $1.6\ V \leq V_{DD} < 1.8\ V$, set the system clock (SYSTEMCLK) to 4 MHz or below. If it exceeds 4 MHz, the operation is not guaranteed.

[ ] While the CPU is running with the low-speed clock, if running the peripheral circuits with the high-speed clock which can frequently generate interrupts, the operation may fail to function properly due to the CPU becoming

incapable of processing interrupts in time. If interrupts frequently occur for reasons such as short interrupt cycles of peripheral circuits, take into account the operating frequency of the CPU so that it can process interrupts in time.

Chapter 7 Low-speed Time Base Counter

See Section 7.2.2 "Low-speed Time Base Counter Operation".

[ ] A time base counter interrupt may occur depending on the timing to write to the LTBR. See the program example for initializing described in Section 7.3.1 "Operation of the Low-speed Time Base Counter".

[ ] T128HZ to T1HZ signals have "0" level in the first half cycle and "1" level in the second half cycle. 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 "0" 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 Figure 7-5 "Low speed time base counter interrupt timing and reset timing of reset by writing to LTBR" for details of the T128Hz to T1Hz waveform.

See Section 7.2.3 "Low Speed Time Base Register Control Register (LTBCCON)".

[ ] Enable the high-speed clock (HSCLK) when using the virtual frequency adjustment mode.

[ ] It takes max. two cycles of low-speed clock (LSCLK) from writing to the TBRUN bit to when the operation starts or stops.

[ ] 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. (except ML62Q1300 group)

See Section 7.2.4 "Simplified RTC Time Base Counter register (LTBRR)".

[ ] Do not write to the LTBRR register. It is read only.

[ ] Read the LTBRR register twice to verify the data to prevent reading uncertain data while counting-up.

See Section 7.2.6 "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".

See Section 7.3.1 "Low-speed Time Base Counter Operation".

[ ] After writing to the LTBR register, the time by which the first low-speed time base counter interrupt request is generated is not guaranteed. If measuring the time using the low-speed time base counter interrupt, do so with reference to the interrupt generation interval.

[ ] The time equivalent to max. one cycle of the system clock is required to reset the counter after writing to the LTBR register.

See Section 7.3.2 "Low-speed Time Base Counter Frequency Adjustment Function".

[ ] The frequency adjustment accuracy does not guarantee the accuracy including the frequency variation of the low-speed oscillation (32.768 kHz) due to temperature variations.

Chapter 8 16bit Timer

See Section 8.1.2 "Configuration".

[ ] When the 16-bit timer is used as two channels of 8-bit timer, the same clock settings and interrupts are applied.

[ ] In the 8-bit timer mode, the TMHnOUT outputs the comparison result of the upper side ("TMHnDH" and "TMHnCH").

[ ] Choose the 16-bit timer mode when using the 16-bit timer DMA request or SA-ADC trigger.

See Section 8.2.2 "16-Bit Timer n Data Register (TMHnD: n = 0 to 7)".

[ ] 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.

See Section 8.2.4 "16-Bit Timer n Mode Register (TMHnMOD: n = 0 to 7)".

[ ] 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 THnOST bit, THnDIV2 to 0 , THnCS0, THnEXS, THnEX are common for both upper side and lower side of the timer.

[ ] Configure the external trigger on the condition that the timer clock is set to "No dividing" and THnCS=1 if the system clock is HSCLK or THnCS=0 if the system clock is LSCLK. See figure 8-13 "External Input Count Setting Flow" in section 8.3.3.2.

See Section 8.2.5 "16-Bit Timer n Interrupt Status Register (TMHnIS: n = 0 to 7)".

[ ] When the THnHIS bit or the THnLIS bit is "1", the interrupt request in the same channel of 8bit timer does not activate. Clear the THnHIS bit or the THnLIS bit by writing "1" to the same number of bit in the TMHnIC register.

See Section 8.2.7 "16-Bit Timer Start Register (TMHSTR)".

[ ] The bit 15 to 8 of TMHSTR register are not used in the 16-bit timer mode. Writing "1" to those bits is ignored.

[ ] Set THnRUN/THnHRUN bits when the timer n is stopped(THnSTAT/THnHSTAT bits of TMHSTAT register are "0").

See Section 8.2.8 "16-Bit Timer Stop Register (TMHSTP)".

[ ] The bit 15 to 8 of TMHSTP register are not used in the 16-bit timer mode. Writing "1" to those bits is ignored.

[ ] Set THnSTP/THnHSTP bits when the timer n is running(THnSTAT/THnHSTAT bits of TMHSTAT register are "1").

[ ] To stop counting during one-shot mode with HSCLK is selected for the system clock and LSCLK (THnCS bit = 0 in the TMHnMOD register) is selected for the timer clock, set the THnSTP bit to "1", and then change the timer clock to HSCLK (THnCS bit = 1 in the TMHnMOD register), and set the timer clock to LSCLK again (THnCS bit = 0 in the TMHnMOD register).

See Section 8.3.2.2 "One-shot Mode".

[ ] In the 8-bit timer mode, the timer output is only available by the upper side 8-bit timer.

See Section 8.3.3.1 "Start/Stop Timing".

[ ] Since counting operation is not suspended during the period in which THnSTAT bit is "1", restart of the counting is ignored even if THnRUN bit is set to "1" in this period. To restart the counting, make sure that the THnSTAT is set to "0", and then set the THnRUN bit to "1".

[ ] After the THnRUN bit is set to "1", the first interrupt has a time error equivalent to maximum of one clock of the timer clock because the counting operation starts in synchronization with the timer clock. The 2nd timer interrupt or later interrupts have constant cycles.

[ ] After the THnSTP bit is set to "1", a 16-bit timer n interrupt (TMnINT) may be generated depending on the stop timing because the counting operation stops in synchronization with the timer clock.

See Section 8.3.3.2 "External Input Count Timing".

[ ] The pulse with the width less than two clocks of the timer clock may be ignored. Always input the external input signal with the width equal to or more than two clocks of the timer clock.

[ ] The external input signal (EXTRGn) which is input to the 16-bit timer is the signal that has passed the sampling controller of the external interrupt function. The sampling of the external interrupt function is optional. See Chapter 18 "External Interrupt Function" for details.

Chapter 9 Functional Timer

See Section 9.2.2 "FTMn Cycle Register (FTnP: n = 0 to 7)".

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

See Section 9.2.3 "FTMn Event Register A (FTnEA: n = 0 to 7)".

[ ] The data set in the FTnEA register must be less than that set in the FTnP register in the TIMER mode or PWM2 mode.

See Section 9.2.4 "FTMn Event Register B (FTnEB: n = 0 to 7)".

[ ] The data set in the FTnEB register must be less than that set in the FTnP register in the TIMER mode.

See Section 9.2.5 " FTMn Dead Time Register (FTnDT: n = 0 to 7)".

[ ] The data set in the FTnDT register must be less than that set in the FTnEA register in the PWM2 mode.

[ ] The sum of setting data in the FTnDT register and the FTnEA register must be less than that set in the FTnP register in the PWM2 mode.

See Section 9.2.9 "FTMn Clock Register (FTnCLK: n=0 to 7)".

[ ] Configure the external trigger using the timer clock without division on the condition that FTnCK0 "1" and the system clock is HSCLK or on the condition that FTnCK0 is "0" and the system clock is LSCLK. See section 9.3.7.1 and 9.3.7.2.

See Section 9.2.11 "FTMn Trigger Register 1 (FTnTRG1: n = 0 to 7)".

[ ] When using the the emergency stop trigger, use another filter function in each peripheral module.

[ ] If a level setting is chosen for the condition of the counter start and condition is matched, the count operation continues (restart the count-up from 0) even if a stop condition is satisfied in the one-shot mode.

[ ] When using the EXTRG0 to EXTRG7, enable the trigger event after setting the noise filter by the FTnTRG1 register. Otherwise, the trigger may occur immediately after setting the FTnTRG1 register.

See Section 9.2.13 "FTMn Interrupt Status Register (FTnINTS: n = 0 to 7)".

[ ] No interrupt request is issued if the interrupt status bit is "1" and the same interrupt occurs again. To issue an interrupt request, write "1" to the same bit in the FTnINTC register and clear the status bit to "0".

See Section 9.2.14 "FTMn Interrupt Clear Register (FTnINTC: n = 0 to 7)".

[ ] Write to FTnIR bit by word access. It is invalid to write to the FTnIR bit by byte access or bit access.

[ ] It is not able to write to FTnIR bit using the debug tools (ex. The SFR window or watch window function, etc).

See Section 9.2.17 "FTM Common Start Register (FTCSTR)".

[ ] Set the FTnSTR when the FTMn stops (FTnSTA bit of FTnSTAT register is "0").

See Section 9.2.18 "FTM Common Stop Register (FTCSTP)".

[ ] Set the FTnSTP bits while the FTMn is operating (FTnSTA bit of FTnSTAT register is "1").

[ ] To stop counting during one-shot mode with HSCLK is selected for the system clock and LSCLK (FTnCK0 bit = 0 in the FTnCLK register) is selected for the timer clock, set the FTnSTP bit to "1", and then change the timer clock to HSCLK (FTnCK0 bit = 1 in the FTnCLK register), and set the timer clock to LSCLK again (FTnCK0 bit = 0 in the FTnCLK register).

See Section 9.3.7.4 " Emergency Stop Operation ".

[ ] Change FTnEMGEN bit avoiding the condition that emergency stop function works incorrectly. Also, release the emergency stop to start a timer after the malfunction.

[ ] Confirm that there is no unhandled interrupt before stopping FTM. The interrupt status is not cleared when you stop FTM while there are some unhandled interrupts.

Chapter 10 Watchdog Timer

See Section 10.1.1 "Features".

[ ] WDT is the function used to monitor the CPU runaway. Its function as an ordinary timer is not guaranteed.

[ ] The watchdog timer is unable to detect all the abnormal operations. Even if the CPU loses control, the watchdog timer is unable to detect the abnormality in the operation state in which the WDT counter is cleared. It is recommended that the WDT counter is cleared at one place in the main loop of the program as a fail-safe.

[ ] WDT can be operated based on the clock independent of the system clock by using RC1K oscillation for the WDTCLK, resulting in further improvement of safety. However, it is recommended to choose LSCLK if high accuracy of the frequency is required, since the RC1K oscillation is less accurate than the LSCLK.

See Section 10.2.2 "Watchdog timer control register (WDTCON)".

[ ] In the WDT interrupt routine (when the interrupt level (ELEVEL) of the CPU program status word (PSW) is "2"), the WDT counter is unable to get cleared.

See Section 10.2.3 "Watchdog Timer Mode Register (WDTMOD)".

[ ] The overflow period set in WDT2 to WDT0 bits is the time when the WDTCLK is 1.024 kHz. If RC1K oscillation is chosen for the WDTCLK clock, the frequency has a significant error.

[ ] If window function enabled mode 1 or window function enabled mode 2 is chosen, no WDT interrupt is generated. A reset is generated in the first overflow.

See Section 10.2.4 "Watchdog Timer Counter Register (WDTMC)".

[ ] The count value read from the WDT counter are discontinuous due to the hardware structure.

See Section 10.3.1 "How to Clear WDT Counter ".

[ ] Maximum of two clocks of WDTCLK are required during the period between writing "0x5A", "0xA5" to the WDTCON register and clearing of the WDT counter. To enter the STOP mode or STOP-D mode following WDT clearing, do so after making sure that the WDTCLR1 bit became "0". In addition, if changing the WDTMOD register setting, write to the WDTMOD register after confirming that both of WDTCLR1 and WDTCLR2 bits became "0" as soon as the WDT counter was cleared.

[ ] In the STOP/STOP-D mode, the WDT timer is stopped.

See Section 10.3.3 "Window Function Enabled Mode".

[ ] When using the window function enabled mode, always define a WDT interrupt function even though no WDT interrupt occurs.

[ ] When using the window function enabled mode, choose "the clock with divided frequency of low-speed oscillation clock (32.768 kHz)" for the WDT count clock with the code option. If "WDT RC1K oscillation clock" is chosen, this function is unavailable to use because the frequency has a significant error.
[ ] In the watchdog timer (WDT) interrupt function, as the interrupt level (ELEVEL) of the CPU program status word (PSW) becomes "2", the WDT counter is unable to get cleared. Clear the WDT when the ELEVEL is "0" or "1". It is recommended that the WDT counter is cleared at one place in the main loop of the program as a fail-safe.

Chapter 11 Serial Communication Unit

See Section 11.2.2 "Serial Communication Unit n Transmit/Receive Buffer (SDnBUF)".
[ ] In the half-duplex communication mode of UART, be sure to choose the transmission mode by setting Un0IO and Un1IO bit of the UARTn mode register (UAn0MOD, UAn1MOD) before writing the transmission data to SDnBUFL and SDnBUFH.
[ ] Do not perform write-operation to the SDnBUF in the SSIO slave reception mode.

See Section 11.2.3 "Serial Communication Unit n Mode Register (SUnMOD)".
[ ] Be sure to set the SUnMOD register while communication is stopped (SUnCON register = 0x00) and do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.
[ ] See section 11.3.2.11 “Note on usage of Half-duplex UART” if using “UART Half-duplex communication mode”.

See 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 invalid in the SSIO master reception mode.

See Section 11.2.6 "Synchronous Serial Port n Mode Register (SIOnMOD)".
[ ] Be sure to set the SIOnMOD register while communication is stopped(SnEN=0) and 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 choosing 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.

See Section 11.2.7 "Synchronous Serial Port n Status Register (SIOnSTAT)".
[ ] Update the SnTUER, SnROER and SnTOER bits by using a byte access.
[ ] Do not write the SnTUER, SnROER and SnTOER bits during while the transmission/reception in progress (SnTXF=1 / SnRXF=1).

See Section 11.2.8 "UARTn0 Mode Register (UAn0MOD)".
[ ] Be sure to set the UAn0MOD register while communication is stopped (Un0EN=0). Do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.

See Section 11.2.9 "UARTn1 Mode Register (UAn1MOD)".
[ ] Be sure to set the UAn1MOD register while communication is stopped (Un1EN=0). Do not rewrite it during communication. If it is rewritten during communication, data may be transmitted or received incorrectly.

See Section 11.2.11 "UARTn1 Baud Rate Register (UAn1BRT)".
[ ] Be sure to set the UAn0BRT and UAn1BRT register while communication is stopped (Un0EN=0, Un1EN=0). Do not rewrite it during communication.

See Section 11.2.13 "UARTn1 Baud Rate Adjustment Register (UAn1BRC)".
[ ] Be sure to set the UAn0BRC and UAn1BRC register while communication is stopped (Un0EN=0, Un1EN=0). Do not rewrite it during communication.

See 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 be sure to read the SDnBUFL even if the data is unnecessary when the reception is completed
[ ] When an error occurs in the start bit, the state returns to the reception waiting state.
[ ] Write the Un0FER bit, Un0OER bit, Un0PER bit and Un0FUL bit by a byte-access.

See 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 be sure to read the SDnBUFH even if the data is not necessary when the reception is completed.
[ ] When an error occurs in the start bit, the state returns to the reception waiting state.
[ ] Write the Un1FER bit, Un1OER bit, Un1PER bit and Un1FUL bit by a byte-access.

See Section 11.3.1.6 "Timing in Clock Type 1 Slave Mode".
[ ] When the timing of transmission interrupt is chosen "at the end of data transmission" (SUnTIMD=0), it is possible to write data to the transfer buffer before the transfer is actually started (before the external clock is supplied). In that case, the data written just before the start of the transfer is transferred. To ensure that data is successfully transmitted, it is recommended that data is written when SnEN is "0" or while the transfer of previous data is in progress (SnTXF=1).

See Section 11.3.2.3 "Direction of Transmit/Receive Data".
[ ] When the SUn_TXDn pin is set to the shared function in the reception mode, "H" level is output from the SUn_TXDn pin.

See Section 11.3.2.10 "Receive Margin".
[ ] When designing the system, consider the difference of the baud rate between the transmission side and reception side and delay of the start bit detection and adjust the baud rate in the UAn0BRT, UAn1BRT, UAn0BRC, and UAn1BRC registers.

See Section 11.3.2.11 "Note on usage of Half-duplex UART".
[ ] When switching the transmission mode to reception mode in the half-duplex UART mode, switch to the reception mode after resetting the channel of SIU(Serial Communication Unit) by Block Reset Control Register 2 (BRECON2). Note that the reset by the BRECON2 also resets an another half-duplex UART in the same channel of SIU.

Chapter 12 I²C Bus Unit

See Section 12.1.4 "Pin Setting".
[ ] Use external pull-up resistors for SDA pin and SCL pin referring to the I²C bus specification. The internal pull-up resistors unsatisfy the I2C bus specification. See the data sheet for each product for the value of internal pull-up resistors.
[ ] If powering off this LSI in the slave mode, it disables communications of other devices on the I²C bus. Keep this LSI powered on when it works as a slave mode until the master device is powered off.
[ ] When using the master function, do not connect multiple master devices on the I²C bus.

See Section 12.2.2 " I²C 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.
[ ] 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 salve function, switch the system clock to the high-speed clock if releasing the communication wait status.
[ ] When using the salve function with multi-slaves connected to the I²C bus, conform to the following conditions while enabling the I²C bus function (I2U0MD=1 and I2US0EN=1) regardless communicating or not.
- Specify SYSTEMCLK as four time or higher than the I²C bus communication speed.
SYSTEMCLK needs to be 500kHz or higher when the I²C bus communication speed is 100kbps.
SYSTEMCLK needs to be 2MHz or higher when the I²C bus communication speed is 400kbps.
SYSTEMCLK needs to be 4MHz or higher when the I²C bus communication speed is 1Mbps.
- Do not use LSCLK as the SYSTEMCLK.
- Do not enter HALT-H mode while enabling the I²C bus function.

See Section 12.2.6 "I²C 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.

See Section 12.2.7 "I²C Bus 0 Mode Register (Master) (I2UM0MOD)".
[ ] When using the high-speed clock for the I²C operation, specify the following I²C operating clock frequency depending on the mode and the reference frequency of the PLL oscillation.
When HSCLK = 24MHz

Standard mode: HSCLK to 1/4HSCLK
Fast mode: HSCLK to 1/2HSCLK
1Mbps mode: HSCLK to 1/2HSCLK
When HSCLK = 16MHz
Standard mode: HSCLK to 1/2HSCLK
Fast mode: HSCLK
1Mbps mode: HSCLK

See Section 12.2.8 "I²C Bus 0 Status Register (Master) (I2UM0STR)".
[ ] Do not update each bit of the I2UM0STR 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.
[ ] I2UM0BB bit and I2UM0BO bit are reset in one I²C operating clock after writing "1" to the bits.

See Section 12.2.12 "I²C Bus 0 Control Register (Slave) (I2US0CON)".
[ ] Switch the system clock to the high-speed clock when releasing the communication wait status.

See Section 12.2.13 "I²C Bus 0 Mode Register (Slave) (I2US0MD)".
[ ] Stop the operation by resetting I2US0EN bit to "0" before entering STOP-D mode. Have the same handling if disable the wake-up from STOP mode by matching the slave address.

See 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 **or** word access, not so that unintented bits are changed by the bit access instructions.
See Section 12.3.4 "Slave Mode Communication Operation Timing".
[ ] If entering to the STOP/STOP-D mode while the slave mode is enabled, first make sure that communication is not in progress (from coincidence of address to reception of stop condition).

See Section 12.3.5 "Operation Waveforms".
[ ] When the slave device uses the clock stretch function which holds the I2CU0_SCL pin at "L" level, the time t_{CYC} and time t_{LOW} are extended.

Chapter 13 I²C Bus Master
See Section 13.1.4 "Pin Setting".
[ ] Use external pull-up resistors for SDA pin and SCL pin referring to the I²C bus specification. The internal pull-up resistors unsatisfy the I²C bus specification. See the data sheet for each product for the value of internal pull-up resistors.
[ ] Do not connect multiple master devices on the I²C bus.

See Section 13.2.5 "I²C Master n Control Register (I2MnCON:n=0,1)".
[ ] 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 other bits of I2MnCON register in the control register setting wait state.

See Section 13.2.6 "I²C Master n Mode Register (I2MnMOD:n=0,1)".
[ ] When using the high-speed clock for the I²C operation, specify the following I²C operating clock frequency depending on the mode and the reference frequency of the PLL oscillation.
When HSCLK = 24MHz
Standard mode: HSCLK to 1/4HSCLK
Fast mode: HSCLK to 1/2HSCLK
1Mbps mode: HSCLK to 1/2HSCLK
When HSCLK = 16MHz
Standard mode: HSCLK to 1/2HSCLK
Fast mode: HSCLK
1Mbps mode: HSCLK

See Section 13.2.7 "I²C Master n Status Register (I2MnSTR: n=0,1)".
[ ] Do not update each bit of the I2MnSTR register by using the bit symbol. Update it by using a byte access or word access, not so that unintented bits are changed by the bit access instructions.
[ ] I2MnBB bit and I2MnBO bit are reset in one I²C operating clock after writing "1" to the bits.

See Section 13.3.3 "Operation Waveforms".
[ ] When the slave device uses the clock stretch function which holds the I2CMn_SCL pin at "L" level, the time t_{CYC} and time t_{LOW} are extended.

Chapter 14 DMA Controller

See Section 14.1 "General Description".

[ ] Do not use the DMA controller and the Coprocessor (Hardware multiplier/divider) simultaneously.

See Section 14.2.2 "DMA Channel n Transfer Mode Register (DCnMOD: n = 0, 1)".

[ ] Set the bits except for DCnSTRG bit when the transfer is disabled (DCnEN bit of DCEN register = 0).

[ ] When performing the software request 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, choose the 16bit timer mode by setting THn8BM bit of 16bit timer n mode register (TMHnMOD) to "0".

See Section 14.2.3 "DMA Channel n Transfer Count Register (DCnTN: n = 0, 1)".
See Section 14.2.4 "DMA Channel n Transfer Source Address Register (DCnSA: n = 0, 1)".
See Section 14.2.5 "DMA Channel n Transfer Destination Address Register (DCnDA: n = 0, 1)".

[ ] Set the DCnTN/DCnSA/DCnDA registers when the transfer is disabled (DCnEN = 0). These registers are not writable if the transfer is enabled (DCnEN = 1).

[ ] If the transfer is stopped (DCnEN = 0) before finishing the specified transfer count, the values of the DCnTN/DCnSA/DCnDA is not guaranteed. Reconfigure the DCnTN/DCnSA/DCnDA when restarting the transfer.

See Section 14.2.6 "DMA Transfer Enable Register (DCEN)".

[ ] Set DCF bit when the transfer is disabled (DCnEN bit = 0). It is invalid to write DCF bit when the transfer is enabled (DCnEN bit = 1).

[ ] When the specified transfer count of channel n is completed, the DCnISTA bit of the DMA status register (DSTAT) is set to "1". Be sure to clear the DCnISTA bit by using the DMA interrupt status clear register (DICLR) before enabling the next DMA transfer. When the DCnISTA bit is "1", the DMA transfer can't be enabled. Clear the status bit (DCnISTA) regardless using or not using the DMA interrupt.

See Section 14.3.1 " Procedure to Use DMA Controller".

[ ] If a non-existing address is set, operation is not guaranteed.

See Section 14.3.3 "DMA transfer request".

[ ] The DMA transfer has priority orders (CPU > DMA channel 0 > DMA channel 1) and limitation of the interval time for the periodical transfers.

Chapter 15 Buzzer

See Section 15.3.1.2 "Example of Intermittent Sound 1 Mode Setting Procedure".
See Section 15.3.2.2 "Example of Intermittent Sound 2 Mode Setting Procedure".

[ ] The buzzer output may be started or stopped in the middle of the buzzer waveform depending on timing of setting the BZ0RUN bit of the BZ0CON register. If it causes a problem, take one of the following measure A or measure B:
Measure A: Use the low-speed time base counter interrupt (choose T8HZ or T1HZ for signal assignment).
Measure B: Use the LTBR register to synchronize the falling edge of the T8HZ or T1HZ signal with the timing BZ0RUN is set.

See Section 15.3.5.1 "Buzzer Output Start and Stop Timing".

[ ] An error to a maximum of one clock of the low-speed clock (LSCLK) occurs by the time the buzzer output is started after writing "1" to the BZ0RUN bit of the BZ0CON register.

[ ] An error to a maximum of one clock of the low-speed clock (LSCLK) occurs by the time the buzzer output is stopped after writing "0" to the BZ0RUN bit of the BZ0CON register.

[ ] In the single sound mode, the BZ0RUN bit of the BZ0CON register is cleared to "0" when the single sound buzzer output is ended.

[ ] In the intermittent sound 1 or 2 mode, an error to a maximum of one clock of the low-speed clock (LSCLK) occurs by the time the buzzer output is started after the T8HZ signal became "1".

Chapter 16 Simplified RTC

See Section 16.2.1 "List of Registers".

[ ] SRTCMAS is reset only by the power-on reset.

See Section 16.2.2 "Simplified RTC Acceptor (SRTCACP)".

[ ] After writing "0x3C" to SRTCACP, if data other than "0x3C" or "0xC3" is written to SRTCACP, writing of "0x3C" becomes invalid.
[ ] When writing "0x3C" and "0xC3" to SRTCACP in this order, and writing a value other than "0xC3" to SRTCACP with writing to SRTCMAS enabled, writing to SRTCMAS becomes invalid.

See Section 16.2.3 "Simplified RTC Minute/Second Counter (SRTCMAS)".
[ ] When reading the SRTCMAS register, read it twice and check that the two values coincide with each other to prevent reading of undefined data during counting up.
[ ] If the data outside the range from 00 minutes 00 seconds to 59 minutes 59 seconds is written to the SRTCMAS register, the register will be set to the initial value.
[ ] An interrupt request may be generated immediately after writing depending on the timing of writing data to the SRTCMAS register. To prevent an interrupt request from being generated while writing time data, disable RTCINT using the simplified RTC control register (SRTCCON) before writing to the SRTCMAS register.
[ ] It is recommended that data is written to the SRTCMAS register with word access.
[ ] After enabling the write operation using the SRTCACP register, data can be written to the SRTCMAS register only once regardless of using byte or word access. If writing twice using 8-bit access after the write operation is enabled, the second writing is ignored.
[ ] If 0 second (0x00) is written to the second counter (SRTCSEC) when it is 59 second (0x59), the minute counter (SRTCMIN) counts up. However, if the minute counter is also written at the same time using 16-bit access, then it does not count up and the written value becomes valid.

Chapter 17 General Purpose Port
See Section 17.2.3 "Port n Mode Register 01 (PnMOD01:n＝0 to 9, A, B)" to Section 17.2.6 "Port n Mode Register 67 (PnMOD67:n＝0 to 9, A, B)".
[ ] Be sure to set the PnMODm(n＝0 to B, m=0 to 7) registers before setting EICON0, EIMOD0 and IE1 registers. If setting the PnMODm register when the interrupt is enabled, unexpected interrupts may happen.
[ ] It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.

See Section 17.2.9 "PORTXT data input register (PXTDI)".
[ ] PI00 and PI01 are unavailable to use as input ports when using the crystal resonator for the oscillation clock. Also, PI01 is unavailable to use as an input port when using the PI01 for the external clock input. See Chapter 6 “Clock Generation Circuit” for more details on how to use the crystal oscillation or external clock input.

See Section 17.3.8 "Notes for using the P00/TEST0 pin".
[ ] 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.

See Section 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 is unavailable to use 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.

Chapter 18 External Interrupt Function
See Section 18.2.3 "External Interrupt Mode Register 0 (EIMOD0)".
[ ] In the STOP/STOP-D/HALT-H (*1) mode, the sampling clock stops and the sampling function does not work regardless the setting in PI7SM to PI0SM bits of EIMOD0 register. When choosing "with sampling" and entering those mode, there is a time period (*2) in which interrupts get disabled. An unintended interrupt may occur, when returning the program run mode from the HALT-H mode. When entering to those modes, specify the external interrupt as "without sampling".
*1 HALT-H in the case the high-speed clock is chosen
*2 When entering the STOP/STOP-D/HALT-H(*1) mode: Max.30µs. When returning from those modes, the interrupt is disabled until the sampling clock starts to be supplied. The start-up time for supplying clock is dependent of the clock or register settings. For details about it, see Table 4-5 "Wake-up Time from Standby Mode" in the Chapter 4 "Power Management".
[ ] An unintended interrupt may occur when switching ENOSC bit of the FCON register, if the HSCLK is chosen for the sampling clock. Choose “without sampling” or disable the external interrupt to prevent an unintended interrupt.

See Section 18.2.4 "Expanded External Interrpt Control Register 0 (EEICON0)".

[ ] Re-request interrupt by writing "1" to EEIR bit of EEITNTC register when interrupt request registers (IRQ01, IRQ23, IRQ45, IRQ67) are written by CPU while expanded external interrupt is enabled.

See Section 18.2.5 "Expanded External Interrupt Mode Register 0(EEIMOD0)".

[ ] In the STOP/STOP-D mode, the sampling clock stops and the sampling function does not work regardless the setting in EPI3SM to EPI0SM bits of EEIMOD0 register. There is a time period*1 in which interrupts get disabled.
(*1) When entering STOP/STOP-D mode: Max.30µs. When returning from STOP/STOP-D mode, the interrupt is disable until the sampling clock (LSCLK) starts to be supplied. The start-up time for supplying clock is dependent of the clock or register settings. For details about it, see Table 4-5 "Wake-up Time from Standby Mode" in the Chapter 4 "Power Management".

See Section 18.2.7 "Expanded External Interrupt Status Register (EEISTAT)".

[ ] When debugging the program on the debugger, remain enabling "External Interrupt" check box on "Peripheral Circuit" tab in the Operation setting menu. If uncheck the option, these status bits might get cleared.
[ ] No interrupt request is issued if the interrupt status bit is "1" and the same interrupt occurs again.
To issue an interrupt request, write "1" to the same bit in the EEINTC register and clear the status bit to "0".

See Section 18.2.8 "Expanded External Interrupt Clear Register (EEINTC)".

[ ] Do not set EEIR bit and EEI3C to EEI0C bits simultaneously.

Chapter 19 CRC Generator

See Section 19.2.2 "Automatic CRC Calculation Start Address Setting Register (CRCSAD)".
See Section 19.2.3 "Automatic CRC Calculation End Address Setting Register (CRCEAD)".
See Section 19.2.4 "Automatic CRC Calculation Start Segment Setting Register (CRCSSEG)".
See Section 19.2.5 "Automatic CRC Calculation End Segment Setting Register (CRCESEG)".

[ ] Write to the registers when CRCAEN bit of the CRCMOD register is "0". Any writing is ignored when the CRCAEN bit is "1".
[ ] Automatic CRC calculation is four-byte length. Generate an expected value by four bytes. Writing to the bit 1 and bit0 are ignored, they are fixed to "0" internally during the calculation.
[ ] Do not specify segment or address out of program code area. See section 2.5 ""Program Memory Space"" for details of the program code area.

See Section 19.2.6 "CRC Calculation Data Register (CRCDATA)".
See Section 19.2.7 "CRC Calculation Result Register (CRCRES)".

[ ] Write to the registers when CRCAEN bit of the CRCMOD register is "0". Any writing is ignored when the CRCAEN bit is "1".

See Section 19.2.8 "CRC Calculation Mode Register (CRCMOD)".

[ ] When the CPU operation mode is "Wait mode" and the PLL reference frequency is 24MHz, choose 12MHz or slower as the SYSTEMCLK before entering the HALT/HALT-H mode.

See Section 19.3.2.1 "Example of Use of Automatic CRC Calculation Mode".

[ ] To perform CRC calculation in the manual mode when automatic CRC calculation is not completed, save the value in the CRCRES register before calculation. Once the CRC calculation in the manual mode is completed, move the saved value back to the CRCRES register and set the CRCAEN bit to "1". If entering the HALT/HALT-H mode then, the automatic CRC calculation can be restarted. The final addresses at the end of the previous operation are stored in the CRCSAD and CRCSSEG registers. If values in the CRCSAD and CRCSSEG registers are overwritten with the CRCAEN bit set to "0", the calculation works correctly.
[ ] When the CPU operation mode is "Wait mode" and the PLL reference frequency is 24MHz, choose 12MHz or slower for the SYSTEMCLK before entering the HALT/HALT-H mode.

Chapter 20 Analog Comparator

See Section 20.1.3 "List of Pins".
See Section 20.2.2 "Comparator n Control Register (CMPnCON: n=0,1)".

[ ] When using the analog comparator, write "0" to the target PnmIE bit and PnmOE bit of port n mode registers (n=0 to 9, A, B, m=0 to 7) to set the general port to Hi-impedance that both input and output is disabled, otherwise a shoot-through current may flow.
[ ] An influence of the noise is reducible by preventing the switching of neighboring pins when the comparator enables.

See Section 20.2.3 "Comparator n Mode Register (CMPnMOD: n=0,1)".

[ ] In the STOP/STOP-D mode, the sampling clock stops and the VLS works without sampling regardless the setting in CMPnCS1 bit and CMPnCS0 bit. When choosing "with sampling" and entering those mode, there is a time period (*1) in which interrupts gets disabled.
*1 Time period to entering the STOP/STOP-D mode: Max.30µs. When returning from those modes, the interrupts are disabled until the sampling clock starts being supplied. The delay time depends on the configuration of clock and registers. See Table 4-5 "Wake-up Time from Standby Mode" in the Chapter 4 "Power Management".

[ ] When the HSCLK is chosen for the sampling clock and the high-speed clock is not oscillating, the sampling circuit does not work. When using analog comparator in this case, choose “No sampling” or “Sampling with LSCLK” for sampling condition. For how to enable the high-speed clock oscillation, see Chapter 6 "Clock Generation Circuit".

[ ] Write CMPnMOD register when the comparator stops (CMPnEN bit of CMPnCON register is "0”), otherwise the comparison result is unguaranteed.

[ ] The internal reference voltage controlled by CMPnVREF bit is for the comparator. See the chapter of "Successive Approximation type A/D Converter" for the reference voltage used in the A/D converter.

Chapter 21 D/A Converter

See Section 21.1.3 "List of Pins".

See Section 21.2.2 "D/A Converter 0 Control Register (DACCON)".

See Section 21.2.4 "D/A Converter 1 Control Register (DACCON1)".

[ ] When using the D/A converter, write "0" to the target PnmIE bit and PnmOE bit of port n mode registers (n=0 to 9, A, B, m=0 to 7) to set the general port to Hi-impedance (input and output disabled), otherwise a shoot-through current may flow.

[ ] An influence of the noise is reducible by preventing the switching of neighboring pins while the D/A converter is operating.

Chapter 22 Voltage Level Supervisor (VLS)

See Section 22.2.2 "Voltage Level Supervisor 0 Control Register (VLS0CON)".

See Section 22.2.3 "Voltage Level Supervisor 0 Mode Register (VLS0MOD)".

[ ] There is a limitation in each mode for entering the STOP/STOP-D mode while the VLS0 is running.

[ ] Even if resets other than the POR and RESET_N pin reset occurred, the VLS0 remains running.

See Section 22.2.5 "Voltage Level Supervisor 0 Sampling Register (VLS0SMP)".

[ ] In the STOP/STOP-D mode, the sampling clock stops and the VLS works without sampling regardless the setting in VLS0SM1 and VLS0SM0 bit. When choosing "with sampling" and entering those mode, there is a time period (*1) in which interrupts gets disabled.
*1 Time period to entering the STOP/STOP-D mode: Max.30µs. When returning from those modes, the interrupts are disabled until the sampling clock starts being supplied. The delay time depends on the configuration of clock and registers. See Table 4-5 "Wake-up Time from Standby Mode" in the Chapter 4 "Power Management".

[ ] When the HSCLK is chosen for the sampling clock and the high-speed clock is not oscillating, the sampling circuit does not work and it does not monitor voltage. When using VLS in this case, choose “No sampling” or “Sampling with LSCLK” for sampling condition. For how to enable the high-speed clock oscillation, see Chapter 6 "Clock Generation Circuit".

See Section 22.3.1.1 "Reset Output".

[ ] Entering the STOP/STOP-D mode is not allowed during the VLS0 stabilization time. If entering the STOP/STOP-D mode after the supervisor mode is enabled, make sure that the VLS0RF bit is set to "1", and then enter the STOP/STOP-D mode.

[ ] The initial value of the VLS0 detection voltage is 4V, so the MCU becomes in reset mode when the V_{DD} is 4V or lower and VLS0 is specified as supervisor mode with the reset function. Therefore, set the detection voltage before enabling the VLS0 operation.

[ ] If you want to use the VLS0 reset function like a reset IC, start the VLS0 when the CPU initially runs at the low-speed clock after the power up.

See Section 22.3.1.2 "Interrupt Output".

[ ] Entering the STOP/STOP-D mode is not allowed during the VLS0 stabilization time. If entering the STOP/STOP-D mode after the supervisor mode is enabled, make sure that the VLS0RF bit is set to "1", and then enter the STOP/STOP-D mode.

[ ] When VLS0 is stopped (VLS0EN bit="0") while the V_{DD} is lower than the specified threshold voltage (VLS0F bit="1"), the VLS0 interrupt is generated.

See Section 22.3.2.2 "Single mode 1" and Section 22.3.2.2 "Single mode 2".

[ ] Entering the STOP/STOP-D mode is not allowed while the single mode operation is in progress. Enter the STOP/STOP-D mode after the single mode operation is completed (VLS0EN bit="0").

See Section 22.3.2.2 "Single mode 2".
[ ] If V_{DD} is higher than the specified threshold voltage, the VLS0 interrupt is not generated.

Chapter 23 Successive Approximation type A/D Converter
See Section 23.1.3 "List of Pins".
[ ] When using the SA-ADC, set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 0 to 9, A, B, m: bit number 0 to 7) to "0" as "Disable input" and "Disable output", otherwise a shoot-through current may flow.
[ ] While the A/D converter is operating, an influence of the noise is reducible by preventing the switching of neighboring pins or A/D converting in the HALT mode.

See 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 writing this register.
[ ] When using the A/D conversion result upper/lower limit detection function (SALEN bit =1), the interrupt can be cleared by clearing the corresponding bit of SAULS15 to SAULS00 or by resetting the LSI.
[ ] When performing the A/D conversion only one time (SALP bit =0), confirm the bit of SAULS15 to SAULS00 is "0" before setting SARUN bit to "1".
[ ] When performing the consecutive scan A/D conversion (SALP bit =1), confirm the bit of SAULS15 to SAULS00 is "0", before the next A/D conversion ends.

See 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 writing this register.
[ ] When using the A/D conversion result upper/lower limit detect function (SALEN bit =1), the interrupt can be cleared by clearing the corresponding bit of SAULS16 or by resetting the LSI.
[ ] When performing the A/D conversion only one time (SALP bit =0), confirm the bit of SAULS16 is "0" before setting SARUN bit to "1".
[ ] When performing the consecutive scan A/D conversion (SALP bit =1), confirm the bit of SAULS16 is "0", before the next A/D conversion ends.

See Section 23.2.6 "SA-ADC Mode Register (SADMOD)".
[ ] Write "0" to the SADMODH[7:1] bits. The operation when "1" is written to the bits is unguaranteed.

See Section 23.2.7 "SA-ADC Control Register (SADCON)".
[ ] Start the A/D conversion with one or more channels chosen by the SA-ADC enable registers (SADEN0 and SADEN1). If no channel is chosen, the operation does not start.
[ ] Enter STOP/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 to 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.

See Section 23.2.8 "SA-ADC Enable Register 0 (SADEN0)".
See Section 23.2.9 "SA-ADC Enable Register 1 (SADEN1)".
[ ] When multiple bits of SACHn (n=00 to 17) 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 bits of SACHn (n=00 to 17) are "0". In that case SARUN bit of SADCON register does not get to "1".
[ ] Channel 16 (SACH16) is used for adjusting frequency of the low-speed RC oscillation clock. When using the channel 16 (SACH16), enable the internal reference voltage/temperature sensor and choose the internal reference voltage by setting VREFCON register.

See Section 23.2.10 "SA-ADC Conversion Interval Setting Register (SADSTM) ".
[ ] When SACK2-0 bits of SADMOD register is "111", the interval time is always minimum (0ns).

See 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 chosen 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.

See Section 23.2.14 "SA-ADC Reference Voltage Control Register (VREFCON)".

[ ] It takes 200 us(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 possible to get incorrect A/D conversion results as affected by external factors.

[ ] When using the external reference voltage input from VREF pin(P23), set VREFP1 bit to "0" and VREFP0 bit to "1" and P2MOD3 register to 0x00.

[ ] The internal reference voltage controlled by the VREFEN bit is for the A/D converter. See chapter 20 "Analog Comparator" for the reference voltage used in the analog comparator.

See 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 by the upper/lower limit detection function gets enabled and the setting for the SADIMD bit of SA-ADC Interrupt Mode Register (SADIMOD) gets invalid.

See Section 23.2.16 "SA-ADC Trigger Register (SADTRG)".

[ ] When choosing the 16-bit Timer trigger (TMH0TRG and TMH1TRG), set the THn8BM bit of the 16bit Timer n Mode Register (TMHnMOD) to "0" to choose the 16bit timer mode.

See Section 23.4 "Notes on SA-ADC".

[ ] Check the notes for using the Successive Approximation Type A/D Converter.

Chapter 24 Regulator

See Section 24.1.3 "List of Pins".

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

[ ] The internal logic voltage (V_{DDL}) is unavailable to use for an external device voltage.

Chapter 25 Flash Memory

See Section 25.2.2 "Flash Address Register (FLASHA)".

[ ] Note that programming for the program memory space is performed by the unit of 4 bytes.Because of this, the setting values in the FA1 bit and FA0 bit are ignored.

See Section 25.2.4 "Flash Data Register 0 (FLASHD0)".
See Section 25.2.5 "Flash Data Register 1 (FLASHD1)".

[ ] Write data into FLASHD0 register at first and FLASHD1 register the second.

[ ] Data written into FLASHD0H register and FLASHD1 register are invalid.

[ ] 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 is unguaranteed.

[ ] Do not read or program unused areas to prevent the CPU works incorrectly.

See Section 25.2.7 "Flash Acceptor (FLASHACP)".

[ ] Even if other instructions are executed between the instruction that writes "0xFA" and "0xF5H to the FLASHACP, the erasing or programming function is still valid.

[ ] If data other than "0xF5" is written to the FLASHACP after "0xFA" is written, "0xFA" becomes invalid. In this case, it needs to write "0xFA" again.

See Section 25.2.9 "Flash Status Register (FLASHSTA)".

[ ] Perform the erasing or programming after checking the FDERSTA bit or FDPRSTA bit are "0". The erasing or programming becomes invalid when either the FDERSTA bit or the FDPRSTA bit is "1".

See Section 25.3 "Self-programming".

[ ] Erase the area to be reprogrammed (data programmed without erasing is unguaranteed).

[ ] Before programming the user program, prepare a program for self-programming in the program code area which is not erased/reprogrammed.

See Section 25.3.1 "Notes on Debugging Self-programming Code".

[ ] Do not perform the real time execution with break points set in the scope of program for self-programming (from setting the flash acceptor to setting the flash data register). Otherwise, the flash memory may not be reprogrammed if break points occur within the scope of program for self-programming.
[ ] Do not perform the step execution within the scope of program for self-programming. Otherwise, the flash memory may not be reprogrammed if the step execution is performed within the scope of program for self-programming.

See Section 25.3.2 "Programming Program Memory Space".
[ ] Only erase areas irrelevant to program processing. If erasing the area where program processing is in progress, the LSI works incorrectly.
[ ] During block/sector erase, the CPU stops the operation for maximum 50 ms whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.
[ ] For block/sector erase, place two NOP instructions following the instruction used to set FERS/FSERS bits of the FLASHCON register to "1".
[ ] Only erase areas irrelevant to program processing. If erasing the area where program processing is in progress, the LSI works incorrectly.
[ ] During the programming, the CPU stops the operation for maximum 80 µs whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.
[ ] For data programming setting, place two NOP instructions following the instruction used to set the programming data in the FLASHD1 register.

See Section 25.3.3 "Programming Data Flash Area".
[ ] The CPU continues program processing even while data flash erase is in progress. Do not enter the STOP mode, STOP-D mode or HALT-H mode during the erase. In addition, set the FSELF bit of the FLASHSLF register to "0" after the erase is completed.
[ ] The data flash area is unreadable during erasing.
[ ] For block/sector erase, place two NOP instructions following the instruction used to set FERS/FSERS bits of the FLASHCON register to "1".
[ ] The CPU continues program processing even while data flash programming is in progress. Do not enter the STOP mode, STOP-D mode or HALT-H mode during the programming. In addition, set the FSELF bit of the FLASHSLF register to "0" (erase/program disabled) after the programming ended.
[ ] The data flash area is unreadable during programming.
[ ] For data programming setting, place two NOP instructions following the instruction used to set the programming data in the FLASHD0L register.

See Section 25.3.4 "Notes on use of self-programming".
[ ] Set to high-speed clock when using the self-programming function. See "Chapter 6 Clock Generation Circuit" for enabling the high-speed clock oscillation and switching the system clock.
[ ] Data in flash memory is not guaranteed if power outage or forced termination due to a reset occurs. Perform block/sector erase again then program data.
[ ] Program the program again using on-chip debug emulator or ISP function, in case the LSI does not start up due to occurrence of power outage or forced termination during programming.

See Section 25.4.2 "Communication Method".
[ ] The UART communication for the ISP function might be affected due to an error of the baud rate and slow slope of the signals. Be sure to evaluate the operation.

See Section 25.4.3 "Communication Command".
[ ] Programming the program 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 area is performed in units of one byte.

See Section 25.4.4 "Transition Command to ISP Mode".
[ ] The transition process from point B to the end of initial setting command (1) shown in the figure 25-6 needs to be completed within 55ms.
[ ] Except ISP mode, don't set TEST0 to "L" when RESET_N is raising to "H".
See Section 25.4.5.1 "Initial Setting".
[ ] The initialization process needs to be completed within 1000ms.

See Section 25.4.5.2 "Erasing Specified Flash Memory Area".
[ ] The erase process needs to be completed within 500ms.

See Section 25.4.5.3 "Programming to Specified Flash Memory Area".

See Section 25.4.5.4 "Verifying Specified Flash Memory Area".

[ ] The programming/verify process needs to be completed within 500ms. In the case of programming to/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

See Section 26.2.1 "Code Options 0 (CODEOP0)".

[ ] Set the WDTSPMD bit to "0".

See Section 26.3 "Code Options 2 (CODEOP2)".

[ ] For the code option data definition, always use the dw directive instruction to configure the data in the unit of word.

Chapter 27 LCD Driver

See Section 27.2.5 "Segment Mode Register 0 (SEGMOD0)" to Section 27.2.9 "Segment Mode Register 4 (SEGMOD4)".

[ ] Write the segment mode register when the display stop mode (LMD1 bit and LMD0 bit of DSPCON register is "0"), prevent erroneous displays or damage of panel.

Chapter 28 On-Chip Debug Function

See Section 28.3 "Precautions".

[ ] Make RESET_N pin able to be connected to V_{DD} with a jumper or something when not using the on-chip debug function.

[ ] Make P00/TEST0 pin able to be connected to V_{DD} with a jumper or something when not using the on-chip debug function.

[ ] Do not program instruction codes into the LSI that set the P00/TEST0 pin to the output mode. If P00/TEST0 is set to the output mode before On-chip emulator performs read/write to/from the target chip, communication with On-chip emulator after that will be disabled. Also note that the input/output mode of P00/TEST0 is uninitialized by the On-chip emulator.

[ ] Validate the ROM code on user production board without the On-chip emulator.

[ ] Disconnect On-chip emulator when measuring the current consumption of the target system. If On-chip emulator remains connected, the current consumption increases as the on-chip debug circuit inside the LSI works for the communication.

[ ] When using the 3.3 VOUT power supply of On-chip emulator, do not apply power of the target system to the V_{DD} pin of LSI. If both power supplies are connected, On-chip emulator may be damaged, or an electric shock or fire may occur.

[ ] LSI used to debug a program is not covered by the product warranty. Do not use the LSI for mass-production.

[ ] A reset due to unused ROM area access does not occur in the on-chip debug mode regardless of code option settings.

[ ] A RAM parity error reset does not occur in the on-chip debug mode and the break operation occurs instead.

[ ] RAM parity error may occur even if the RAM area is not displayed in case the contents of the data memory are displayed in the debugger in a state where a RAM parity error may occur (including uninitialized RAM).

[ ] The all interrupts and watchdog timer operation always stop while the debugger is in the break state.

[ ] On-chip emulator might be affected by the external environments such as the host PC, USB cable, On-chip emulator interface cable and the target system. Please confirm proper environments before using the On-chip emulator.

[ ] If adding an external capacitor to the RESET_N pin, prepare a jumper function on the board so that the capacitor gets disconnectable when using the debugger or Flash multi-writer.

Chapter 29 Safety Function

See Section 29.2.5 "RAM Parity Setting Register (RASFMOD)".

[ ] □ If reading the RAM data without initializing the RAM, a parity error may occur. When using parity errors (checking parity error flags or enabling parity error reset), initialize all areas of the RAM before start using.

See Section 29.2.8 "MCU Status Interrupt Register (MCISTATL)".

[ ] No interrupt request is issued if the interrupt status bit is "1" and the same interrupt occurs again.
To issue an interrupt request, write "1" to the same bit in the MCINTCL register and clear the status bit to "0".

See Section 29.3.2 "Unused ROM Area Access Reset Function".

[ ] CSR[3] is unused on the ML62Q1000 series. The data of CSR "0x8 to 0xF" are handled as "0x0 to 0x7".

See Section 29.3.3 "Clock Mutual Monitoring Function".

[ ] 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.

Appendix A Register List (SFR List)

[ ] Note there are some SFRs that has undefined initial value.

Appendix B Package Dimensions

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).

Notes for the package with exposed die pad

[ ] The die pad is exposed on the bottom of WQFN package. Make the die pad electrically open when soldering onto the PCB.

Appendix D Application Circuit Example

[ ] Place the capacitor for V_{DDL} pin as close to the LSI power pins as possible.

Data Sheet Electrical Characteristics

External capacitor for power pins

[ ] C_L = 1.0uF (for V_{DDL} pin), [ ] C_V (for V_{DD})

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 +85°C (Programming the Flash)

Revision History

REVISION HISTORY

Document No,	Date	Page		Description
		Previous Edition	Current Edition	
FEUL62Q1000-01	2018.12.11	-	-	ML62Q1300/1500/1700 Integrated First Edition
FEUL62Q1000-02	2019.02.08	1-1	1-1	[1.1] Added "WQFN32" in Table 1-1.
		1-1~2, 1-7~8, 1-10~11, 1-13~14, 1-27~30, 1-33~36	1-1~2, 1-7~8, 1-10~11, 1-13~14, 1-26~29, 1-32~35	Added the Products with 384KB/512KB ROM in the following: [1.1] Table1-2,1-3, 1-5 and 1-6 [1.1] LCD driver section and shipping package section [1.1] Figure 1-2 and 1-3 [1.3.1.9]~[1.3.1.12] and [1.3.1.15]~[1.3.1.18]
		1-3	1-3	[1.1] Changed the description of erase unit of data-flash.
		1-4	1-4	[1.1] Added one-time reset in the reset function section.
		1-4, 1-12~14	1-4, 1-12~14	[1.1] Corrected numbers of interrupt.
		1-6	1-6	[1.1] Corrected an address format of I^2C Slave.
		1-16~18	1-15~17	[1.1] Added V_{REFO} in Figure 1-4, 1-5 and 1-6
		1-37~86	1-36~42	[1.3.2] Changed table layout and corrected pin list in Figure 1-7~9.
		1-87, 1-90	1-43, 1-46	[1.3.3] Corrected description of V_{DD}, TEST0, low-speed time base counter and 16-bit timers.
		1-94	1-50	[1.3.4] Added NC and P71-75 pins in Figure 1-10 and 1-11
		2-1~2, - - -	2-2~3, 2-19~20, 2-24~25, 2-35, 2-41	Added the Products with 384KB/512KB ROM in the following: [2.1] Table 2-2 and 2-3 [2.5] Figure 2-9, 2-10, 2-17 and 2-18 [2.6] Figure 2-35, 2-36, 2-47 and 2-48
		2-13	2-14	[2.5] Moved and added the note.
		2-22	2-26	[2.6] Moved the note.
		2-37	2-43	[2.7.2] Corrected the note.
		2-38, 2-41	2-44, 2-47	[2.7.3][2.8.3] Added RES2 and CRES2 bit.
		3-10	3-10	[3.3.3] Deleted figure 3-2.
		3-11	3-11	[3.3.4] Corrected symbol of power-on reset reaction time.
		4-6	4-6	[4.2.3] Deleted comment *1.
		4-14, 4-20	4-14, 4-20	[4.2.9][4.2.13] Corrected access type R/W of bit 2.
		6-1	6-1	[6.1.1] Added the condition of adjustable to ±1%
		6-2	6-2	[6.1.2] Corrected the column in table 6-3
		6-3, 6-21	6-3, 6-21	[6.1.3][6.3.2.1] Corrected table number in the body text.
		7-15, 7-16	7-15, 7-16	[7.3.2][7.3.3] Updated the body text.
		8-10	8-10	[8.2.4] Added the note.
		8-28	8-28	[8.3.3.2] Corrected the flow in figure 8-13.
		9-15, 9-21	9-15, 9-21	[9.2.5][9.2.9] Added notes.
		9-37	9-37	[9.3.1] Corrected symbol in the formula.
		9-39	9-39	[9.3.2.1] Added an exposition of update timing of the configuration.
		9-55	9-55	[9.3.7][9.3.7.1] Changed title. [9.3.7.1] Added an exposition of external clock input
		11-8	11-8	[11.1.4] Updated the combination list.
		15-4	15-4	[15.1.3] Deleted "ML62Q1700 group" and corrected "Pin name" column in table 15-1.
		16-7	16-7	[16.3] Corrected table number in the body text.
		18-8	18-8	[18.2.3] Updated the description of 1st note.
		18-11	18-10	[18.2.5] Moved the note from [18.2.6].
		20-8	20-8	[20.2.3] Updated the description of 1st note.
		23-12	23-12	[23.2.6] Updated the exposition of the SAINT bit.
		23-28, 23-29	23-28, 23-29	[23.3.3] Changed the notes to the comment in figure 23-3 and 23-4.
		23-31	23-31	[23.4.1] Updated the exposition of sampling time.
		24-1	24-1	[24.1] Corrected the body text.
		24-4	24-4	[24.1.3] Added V_{REFO} pin in the pin list.
		-	24-4	[24.2.1] Added new section.
		25-2	25-2	[25.1] Added the Products with 384KB/512KB ROM
		25-3, 25-15, 25-18	25-3, 25-17, 25-20	[25.1][25.3][25.3.3] Changed the description of erase unit of data-flash.

Document No,	Date	Page		Description
		Previous Edition	Current Edition	
(FEUL62Q1000-02)	(2019.02.08)	25-7~8	25-8~10	[25.2.3] Added an address of the products with 384KB/512KB ROM
		25-28	25-30	[25.4.5.4] Corrected command name in figure 25-11.
		-	25-31~34	[25.4.6] Added new section.
		26-4	26-4	[26.2.3] Added the CRES2 bit.
		26-4	26-4	[26.2.3] Corrected the CPU instruction execution start address in the table.
		26-5	26-5	[26.3] Added the Products with 384KB/512KB ROM
		27-1, 27-5~7, 27-23~24	27-1, 27-5~7, 27-23~24	[27.1.1][27.1.4][27.2.10]Added the Products with 384KB/512KB ROM
		29-14	29-14	[29.3.2] Added the Products with 384KB/512KB ROM
		29-15	29-15	[29.3.3] Updated the exposition.
		A-2, A-5~7	A-2, A-5~7	Modified Japanese words to English.
		A-5	A-5	Corrected byte symbol of DSPR0-9.
		A-14	A-14	Corrected R/W type of reserved bits.
		Contents-*	Contents-*	Updated the page numbers according to revised pages in each chapter.
		E-*	E-*	Updated the description all according to revised pages in each chapter.
FEUL62Q1000-03	2020.01.24		3	Updated the relevant documents
		1-11	1-11	[1.1] Corrected pin count number in Fig. 1-3
		1-50	1-50	[1.3.4] Updated about RESET_N and TEST0 pin in table 1-11
		3-11	3-11	[3.3.4] Updated the exposition and note.
		4-6, 4-23~25	4-6, 4-23~25	[4.2.3][4.3.4]~[4.3.6] Added note.
		4-8, 4-9	4-8, 4-9	[4.2.5][4.2.6] Corrected access type.
		4-20	4-20	[4.2.13] Corrected access type
		4-29	4-29	[4.3.8] Updated description of note.
		-	4-30	[4.3.9] Added new section
		6-4	6-4	[6.2.1] Corrected access type (FBTACP)
		6-11	6-11	[6.2.7] Corrected initial value
		7-5	7-5	[7.1.3] Corrected exposition
		7-8	7-8	[7.2.3] Corrected exposition of bit 6 and note.
		7-9	7-9	[7.2.4] Corrected exposition and note.
		7-15	7-15	[7.3.2] Updated exposition
		8-9, 8-10	8-9, 8-10	[8.2.4] Added "rsvd" bit
		8-23	8-23	[8.3.2.1] Updated exposition
		9-6	9-6	[9.1.3] Corrected register symbol of FTM0P in table 9-2
		9-13	9-13	[9.2.3] Corrected bits exposition
		9-20~27	9-20~27	[9.2.9]~[9.2.12] Added "rsvd" bit
		9-21, 9-24, 9-26, 9-56	9-21, 9-24, 9-26, 9-57	[9.2.9]~[9.2.11][9.3.7.3] Deleted exposition/note about trigger width
		9-25	9-25	[9.2.11] Updated exposition of bit 10 to 8
		9-31	9-31	[9.2.14] Corrected FTnIR bit, and added note.
		9-32	9-32	[9.2.15] Corrected bit exposition
		9-33	9-33	[9.2.16] Corrected symbol of bit 5 and 4
		9-55	9-55	[9.3.7] Updated exposition and Fig. 9-15
		-	9-56	[9.3.7.2] Added new section
		9-57	9-58	[9.3.7.4] Added exposition and note
		11-5~7	11-5~7	[11.1.3] Corrected comment in table 11-3.
		11-8	11-8	[11.1.5] Updated exposition
		11-9~14	11-9~14	[11.2.1] Corrected access type and size of UA00STAT, SU1CON, SU2CON, SU3CON, SU4CON and SU5CON
		11-23, 11-25, 11-27, 11-29~31	11-22, 11-24, 11-26, 11-28~30	[11.2.6]~[11.2.13] Added description into note
		11-24, 11-38~40, 11-44	11-23, 11-37~39, 11-43	[11.2.7] Prohibited use of SnRFUL bit [11.3.1.2][11.3.1.3][11.3.1.6] Deleted SnRFUL in Fig. 11-6 to 11-11
		11-25, 11-44	11-24, 11-43	[11.2.7] Added exposition of bit 0. [11.3.1.6] Added exposition and corrected Fig 11-11
		11-33, 11-35	11-32, 11-34	[11.2.14][11.2.15] Added exposition of bit 1

Document No,	Date	Page		Description
		Previous Edition	Current Edition	
(FEUL62Q1000-03)	(2020.01.24)	11-36, 11-38	11-35, 11-37	[11.3.1.1][11.3.1.2] Added exposition
		11-44	11-43	[11.3.1.6] Corrected note
		11-46	11-45	[11.3.2.2] Updated expression
		12-1, 12-3	12-1, 12-3	[12.1.1][12.1.4] Corrected description in table 12-2 and 12-3.
		12-4	12-4	[12.2.1] Corrected access type and size of I2US0STR
		12-6~9, 12-11, 12-12	12-6~9, 12-11, 12-12	[12.2.3]~[12.2.8] Added/Corrected exposition about I2UM0EN
		12-10	12-10	[12.2.7] Corrected comment of bit 10 to 8
		12-24, 12-30	12-24, 12-30	[12.3.1.8][12.3.3.7] Corrected value of I2UM0EN/I2US0EN
		13-5~9, 13-11	13-5~9, 13-11	[13.2.2]~[13.2.7] Added/Corrected exposition about I2MnEN
		13-16	13-16	[13.3.1.8] Corrected value of I2MnEN
		14-13~15	14-13~15	[14.3.1]~[14.3.3] Updated exposition/note and added new section
		14-16, 14-17	14-16, 14-17	[14.3.4][14.3.5] Corrected flow
		15-7	15-7	[15.2.3] Corrected description of bit 12
		15-9, 15-10	15-9, 15-10	[15.3.1.2][15.3.2.2] Updated description of note.
		16-4, 16-7	16-4, 16-7	[16.2.3] Corrected symbol T1HZ to T1HZR in the explanation of bit 0 [16.3.1] Corrected symbol T1HZ to T1HZR in Fig. 16-2
		17-3	17-3	[17.1.2] Updated Fig. 17-1
		17-17, 17-20, 17-23, 17-26	17-17, 17-20, 17-23, 17-26	[17.2.3]~[17.2.6] Updated note
		17-32	17-32	[17.3.5.1] Corrected symbol P3PMD to P0PMD
		17-33	17-33	[17.3.7] Updated exposition
		17-34	17-34	[17.3.8.1][17.3.8.2] Updated exposition
		18-8	18-8	[18.2.3] Updated note
		18-15	18-15	[18.3.2] Corrected Fig. 18-5
		19-5~8	19-5~8	[19.2.2]~[19.2.5] Added note
		20-3	20-3	[20.1.2] Corrected Fig 20-1
		20-4, 20-6	20-4, 20-6	[20.1.3][20.2.2] Corrected description of note
		20-8	20-8	[20.2.3] Corrected note
		20-9, 20-10	20-9, 20-10	[20.3.1] Corrected exposition
		20-10, 20-11	20-10, 20-11	[20.3.2] Corrected Fig 20-4 to 20-6
		22-8	22-7	[22.2.4] Corrected exposition of bits
		22-9	22-8	[22.2.5] Added note
		22-10, 22-11	22-9, 22-10	[22.3][22.3.1] Updated explanation
		23-10, 23-11	23-10, 23-11	[23.2.4][23.2.5] Corrected bits explanation
		23-16	23-16	[23.2.10] Corrected expression and exposition
		23-24	23-24	[23.3.1] Corrected value of IE23 in Fig. 23-3
		23-31	23-31	[23.4.1] Corrected expression
		24-3	24-3	[24.1.2] Updated Fig. 24-2
		25-25	25-25	[25.4.4] Added note
		26-4	26-4	[26.2.3] Corrected table of starting address
		27-1	27-1	[27.1.1] Corrected description
		27-9	27-10	[27.2.2] Corrected exposition of bit 4
		27-21	27-21	[27.2.9] Corrected exposition of bit 0
		28-3	28-3	[28.1.2.1][28.1.2.2] Updated Fig. 28-1, 28-2
		28-5	28-5	[28-3] Updated notes
		29-4, 29-5, 29-9	29-4, 29-5, 29-9	[29.2.2][29.2.3][29.2.6] Added "rsvd" bit
		29-15	29-15	[29.3.3] Corrected signal name TMnINT to TMHnTRG in Fig. 29-4
		A-1	A-1	Added description
		A-*	A-*	Corrected register name, type and so on.
		E-*	E-*	Updated the description all according to revised pages in each chapter.
		-	*-*	Corrected typo, unified wording, revised descriptions
FEUL62Q1000-04	2020.03.19	*-*	*-*	Corrected typo
FEUL62Q1000-05	2020.07.15	1-47	1-47	[1.3.3] Corrected description of SA-ADC
		3-9	3-9	[3.3.1] Corrected Table.3-2 and Updated comment.

Document No,	Date	Page		Description
		Previous Edition	Current Edition	
(FEUL62Q1000-05)	(2020.07.15)	6-18~20	6-18~20	[6.3.1.2] Updated Fig.6-6 to 6-8
		6-11, 6-17, 6-25	6-11, 6-17, 6-25	[6.3.1.2] Added and updated exposition of backup mode [6.2.7][6.3.5] Updated exposition of backup mode
		6-20	6-20	[6.3.1.3] Corrected exposition
		7-4	7-4	[7.1.2] Corrected Fig.7-2
		7-7	7-7	[7.2.2] Deleted note
		7-6, 7-9, A-3	7-6, 7-9, A-3	[7.2.1][7.2.4] Changed LTBRR register prohibition to write.
		1-5, 1-12~17, 11-2, 11-18	1-5, 1-12~17, 11-2, 11-18	[1.1][1.2][11.1.1][11.2.3] Changed exposition of UART.
		17-15, 17-16	17-15, 17-16	[17.2.3] Added an exposition; initial value of P00
		22-6	22-6	[22.2.3] Deleted note
		22-11, 22-12	22-11, 22-12	[22.3.1.1] Corrected Fig.22-3 and 22-4, added exposition.
		27-12	27-12	[27.2.4] Added exposition to bit 10-8
		-	27-35	[27.3.7] Added section for configuration example to drive LEDs
		-	28-6	[28.5] Added section for reset
		D-1	D-1	Corrected connection of debug-interface
		-	*-*	Corrected LTBC0INT, LTBC1INT, LTBC2INT to LTB0INT, LTB1INT, LTB2INT
		E-*	E-*	Updated the description all according to revised pages in each chapter.
		-	*-*	Corrected typo, unified wording, revised descriptions
FEUL62Q1000-06	2021.06.01	ii	ii	Added note of characteristics difference between the product
		1-2	1-2	[1.1] Added exposition of seeing the notes for product usage
		1-46	1-46	[1.3.3] Added exposition of low-speed time base counter TBCOUT0
		1-46	1-46	[1.3.3] Added exposition of low-speed time base counter TBCOUT1
		1-50	1-50	[1.3.4] Corrected exposition of termination of V_{L3}
		-	2-5	[2.2.2] Added note of executing SB/RB instruction
		-	2-5	[2.2.3] Added note on the description of read-modify-write
		3-6	3-6	[3.2.2] Added and supplemented exposition of RSTAT
		3-9	3-9	[3.3.1] Deleted FBTCON from *1 of Table 3-2
		-	4-31	[4.3.10] Added note of entering to the standby mode
		5-14,5-16,5-18, 5-20	5-14,5-16,5-18, 5-20	[5.2.6], [5.2.7] [5.2.8], [5.2.9] Added note on interrupt request register
		5-22	5-22	[5.2.11] Corrected bit exposition of CIL
		-	5-52	[5.3.8] Added note on writing to IRQ01/IRQ23/IRQ45/IRQ67
		6-1	6-1	[Table 6-1], [Table 6-2] Added calculation value on the frequency of HSCLK
		9-23,9-24	9-23,9-24	[9.2.10] Corrected bit exposition of FTnDCLH(Bit6) and FTnSTC(Bit1)
		9-32	-	[9.2.14] Added note on FTnINTC
		9-45	9-46	[9.3.4.1] Corrected exposition of operation example in CAPTURE mode
		11-19	11-19	[11.2.4] Corrected the name of clock on calculation formula
		11-23	11-23	[11.2.7] Corrected bit exposition of SnTOER(Bit2)
		11-35, 11-39	11-35, 11-39	[11.3.1.1], [11.3.1.3] Added exposition of SOUT value after the completion if transmission
		11-45	11-45	[11.3.2.2] Added exposition of treatment when BRC value on calculation formula of baud rate become to 8
		12-9	12-9	[12.2.6] Corrected note of the I2UM0CON
		12-10	12-10	[12.2.7] Corrected the speed reduction rate of I2UMnDW1,0[Bit4,3]
		12-18	12-18	[12.2.13] Corrected bit exposition of I2US0SPE(Bit2)
		13-8	13-8	[13.2.5] Corrected note of the I2MnCON
		13-9	13-9	[13.2.6] Corrected the speed reduction rate of I2MnDW1,0[Bit4,3]
		17-34	17-34	[17.3.8.1] Added exposition of using on-chip debug function
		18-9	18-9	[18.2.4] Added note of EEICON0
		18-13	18-13	[18.2.8] Corrected bit exposition of EEIR(Bit15)
		18-17	18-17	[18.3.4] Added annotation on setting flow
		E-*	E-*	Updated the description all according to revised pages in each chapter.
		-	*-*	Corrected typo, unified wording, revised descriptions

Document No,	Date	Page		Description
		Previous Edition	Current Edition	
FEUL62Q1000-07	2022.2.21	ii	ii	Updated note of characteristics difference between the product
		-	*-*	Corrected typo
FEUL62Q1000-08	2023.1.10	*-*	*-*	Corrected typo
		-	*-*	Unified wording as for hi-impedance state of port.
		3-9	3-9	[3.3.1] Corrected comment *1 in the Table 3-2
		6-1	6-1	[6-1]Table6-1, [6.1.1]Table6-2: Corrected detail value of 24MHz
		6-25	6-25	[6.3.5]Table6-15: Added comment *6
		7-10	7-10	[7.2.5] Added description
		9-25, 9-35, 9-57	9-25, 9-35, 9-57	[9.2.11] Added explanation of FTnTRF2-0 bits [9.2.17] Added explanation of FTnETG bit [9.3.7.2] Added description.
		11-8	11-8	[11.1.3] Updated comment *3
		11-35, 11-36	11-35, 11-36	[11.3.1.1] Corrected operation description and updated Fig11-2 to 11-5
		11-48	11-48	[11.3.2.4] Corrected operation description and Fig11-15
		18-14, 18-15	18-14, 18-15	[18.3.1] Updated waveform signal name in the Fig18-3 and 18-4
		18-17	18-17	[18.3.4] Corrected a flow in the Fig18-7
		B-*	B-*	Revised Package Dimension (not updated dimension value)
FEUL62Q1000-09	2023.6.12	8-16	8-16	[8.2.8] Added a note about one-shot mode.
		9-30	9-30	[9.2.13] Added a note about interrupt clearing.
		9-36	9-36	[9.2.18] Added a note about one-shot mode.
		10-18	1-18	[10.3.3] Correct the note.
		18-12	18-12	[18.2.7] Added a note about interrupt clearing.
		23-31	23-31	[23.4.1] Typo correction. (V_{DD}->V_{REF})
		28-5	28-5	[28.3] Added a note about RAM parity.
		29-8	29-8	[29.2.5] Added a note about RAM parity.
		-	*-*	Corrected typo
FEUL62Q1000-10	2023.12.15	i	i	Rivised the Note.
		-	1-1	[1] Added aplication information.
		1-8	1-8 ~1-10	[1.1] Changed the format of the Shipping package information.
		1-9~1-11	1-11~1-13	[1.1] Updated "how to read the part numbter".
		1-42	1-42	[1.3.2] Table1-9 (3/3) Corrected typo
		28-5	28-5	[28.3] Correct the note.
		B-4	-	Delete SSOP20 information
		E-18	E-18	Correct the note.