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

FEUL62Q2500-04

ML62Q2500 Group
User's Manual

Issue Date: Mar. 26, 2024

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 PRODUCT
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 ML62Q2500 Group.

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

Notation

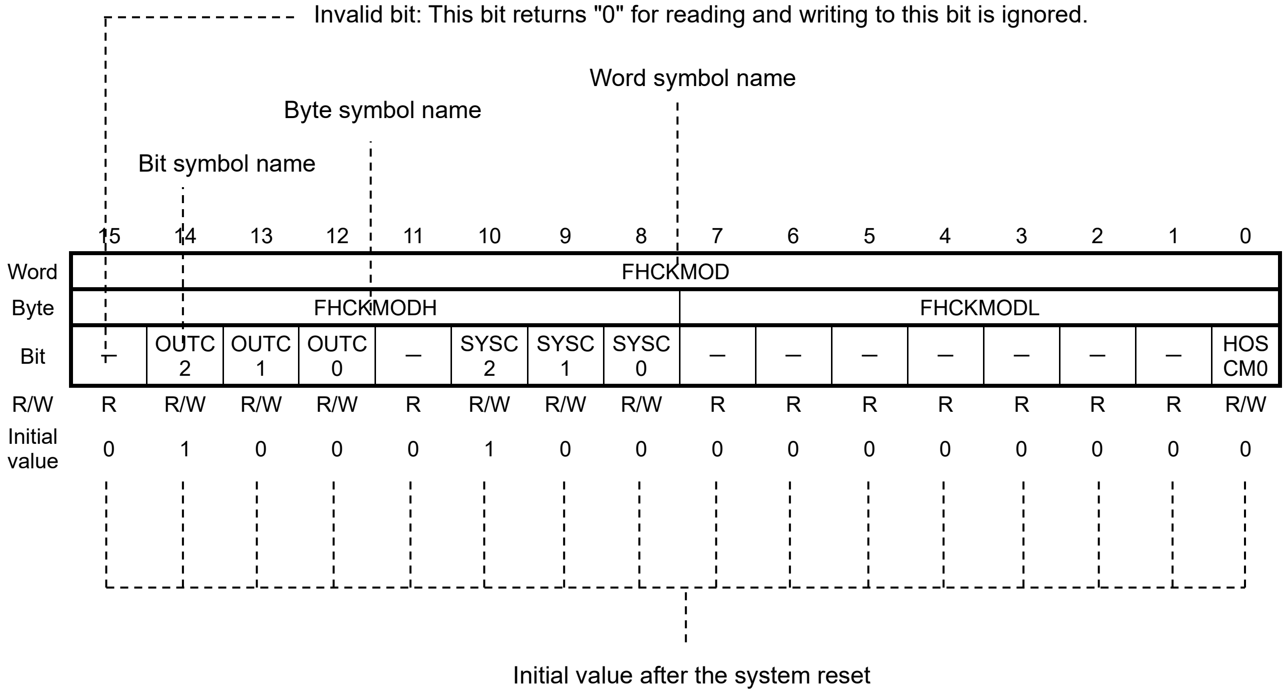
Classification	Notation	Description
Numeric value	XXh, XXH, 0xXX	Indicates a hexadecimal number.
Unit	word, W	1 word = 16 bits
	byte, B	1 byte = 8 bits
	nibble, N	1 nibble = 4 bits
	mega-, M	10^6
	kilo-, K	2^{10} = 1024
	kilo-, k	10^3 = 1000
	milli-, m	10^{-3}
	micro-, μ	10^{-6}
	nano-, n	10^{-9}
	second, 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.
	SFR	Special function register. It is control register for system or peripherals.

- Register description

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

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

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


Invalid bit: This bit returns "0" for reading and writing to this bit is ignored.
Word symbol name
Byte symbol name
Bit symbol name
Initial value after the system reset

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FHCKMOD															
Byte	FHCKMODH								FHCKMODL							
Bit	–	OUTC 2	OUTC 1	OUTC 0	–	SYSC 2	SYSC 1	SYSC 0	–	–	–	–	–	–	–	HOS CM0
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

Chapter 1 Overview

- **Please see the “Notes” and the “Notes for product usage” in this document.**

1. Overview

ML62Q2500 Group is a high performance CMOS 16-bit microcontroller equipped with an 16-bit CPU nX-U16/100 and integrated with program memory(Flash memory), data memory(RAM), data Flash (Erase unit:128byte, Write unit:1byte) and rich peripheral functions such as the multiplier/divider, CRC generator, Clock generator, Timer, General Purpose Ports, UART, Synchronous serial port, I^{2}C bus interface unit(Master, Slave), Voltage Level Supervisor(VLS), Successive approximation type 12bit A/D converter, 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.
The built-in on-chip debug function enables debugging and programming the software. Also, ISP (In-System Programming) function supports the Flash programming in production line.

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

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

- Product List
  The ML62Q2500 Group has products as show in the Table 1-1 with multiple package and memory sizes combinations.

Table 1-1 Product List

Program memory	Data memory (RAM)	Data Flash	32pin TQFP32 WQFN32	40pin WQFN40	48pin TQFP48 WQFN48
128Kbyte	8Kbyte	4Kbyte	ML62Q2504	ML62Q2524	ML62Q2534
64Kbyte			ML62Q2502	ML62Q2522	ML62Q2532

1.1 Features

- CPU
  - 16-bit RISC CPU: nX-U16/100 (A35 core)
  - Instruction system: 16-bit length instructions
  - Instruction set: Transfer, arithmetic operations, comparison, logic operations, multiplication/division, bit manipulations, bit logic operations, jump, conditional jump, call return stack manipulations, arithmetic shift, and so on
  - Built-in On-chip debug function (connect to the LAPIS Technology on-chip-debug emulator)
  - Minimum instruction execution time : 1 count of system clock
    Approximately 30.5μs/62.5ns/41.6ns (at 32.768kHz/16 MHz/24MHz system clock)

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

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

- Operating voltage and temperature
  - Operating voltage: V_{DD} = 1.8 to 5.5 V
  - Operating temperature: -40 °C to +105 °C

- Flash memory

Parameter	Program memory area	Data Flash memory area
Erase/Write count	100 cycles	10,000 cycles
Write unit	32bit(4byte)	8bit(1byte)
Erase unit	16Kbyte/1Kbyte	all area/128byte
Erase/Write temperature	0 °C to +40 °C	-40 °C to +85 °C

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

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

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

- Clock generation circuit
  - Low-speed clock (LSCLK) : Approximately 32.768 kHz
    - Internal low-speed RC oscillation (RC32K)
    - External low-speed clock input (EXT32K)
    - External low-speed crystal oscillation (XT32K)
      4 selectable crystal oscillation mode (Tough, Normal, Low power and Ultra low power mode)
      Tough mode is largest oscillation allowance to make highest resistance against leakage between the pins. Low power mode is smallest oscillation allowance to make lower power consumption.
  - High-speed clock (HSCLK)
    - PLL oscillation: 3 selectable oscillation frequency (24MHz ,16MHz and 1MHz) by code option
  - Watch Dog Timer (WDT): built-in independent clock for WDT (RC1K: Approximately 1.024kHz)
  - High-speed time base clock (HTBCLK)
    Generates a clock with a period of 2 to 8 times that of HSCLK as a peripheral clock.

- Reset
  - System Resets by reset input pin, Power-On Reset, voltage level supervisor (VLS), WDT overflow, WDT invalid clear, RAM parity error, and PC error (unused ROM area access (instruction access))
  - Software reset by BRK instruction (reset CPU only)
  - Reset the peripherals individually/collectively by software

- Power management
  - Optimal power management with various standby modes
    - STOP/STOP-D mode(All clocks are stopped), HALT-D mode(clocks for System and part of the peripheral block are stopped), HALT/HALT-H mode(clocks for System are stopped)
    - HALT-D mode is suitable for long term standby, HALT-H mode is suitable for short term Intermittent operation standby
  - Individual clock input control to the peripheral blocks by software
  - High-speed clock frequency(HSCLK) is configurable (1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64 of PLL clock, Max 7steps)
  - Clock gear: High-speed system clock frequency is changeable dynamically
  - (1/1, 1/2, 1/4, 1/8, 1/16, 1/32 of HSCLK, Max 6steps)

- Interrupt controller
  - Non-maskable interrupt source: 1 (internal sources: WDT)
  - Maskable interrupt sources: 34 (included the external interrupt 8 sources)
  - Four step interrupt levels
  - External interrupt ports: 8 (selectable from max.24 pins) with sampling filter and edge (rise, fall, both) selection

- General-purpose ports (GPIO)
  - I/O port: Max. 40 (Including pins for shared functions)
  - Input port: Max. 3(Including one pin for shared on-chip debug and two pins for shared low speed crystal oscillation)
  - Carrier frequency output function (for IR communication)

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

- Low-speed Time base counter(LTBC) : 2 channels
  - Generate 8 frequency (128, 64, 32, 16, 8, 4, 2, 1[Hz]) internal pulse signals by dividing the Low-speed clock (LSCLK)
  - 4 interrupts are generatable from 8 different frequencies internal pulse signals
  - One of internal pulse signals selected to interrupt can be output from general purpose port (TBCO)

- Functional timer : 2 channels
  - Various modes (Continuous, One shot, capture, PWM with the same period and different duties, and complementary PWM output with the dead time)
  - Event trigger (external pin, 16bit timer, functional timer, LTBC, RC1K)
  - Selectable counter clock from various sources (divided by 1 to 8 of LSCLK, HSCLK, HTBCLK, external clock)

- 16-bit General timers : 6 channels
  - Timer output (toggled by overflow)
  - Selectable counter clock from various sources (divided by 1 to 8 of LSCLK, HSCLK, HTBCLK, LTBC, RC1K, and external clock)
  - Timer X is shared with waiting for the stability of low-speed crystal oscillation

- Synchronous Serial Port : 2 channels (with FIFO: 1channel, without FIFO: 1channel)
  - FIFO: 4steps for each transmitting and receiving
  - Selectable from Master and Slave
  - Selectable from LSB first or MSB first
  - Selectable 8-bit length or 16-bit length

- UART (Full-duplex communication mode): 3 channels
  - Selectable from 5 to 8bit length, parity or no parity, odd parity or even parity, 1 stop bit or 2 stop bits, Positive logic or Negative logic, LSB first or MSB first
  - Sampling filter for receiving data and start bit
  - Built-in baud rate generator（HSCLK(16MHz): 300bps to 2Mbps, LSCLK: up to 4800bps）

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

- Successive approximation type 12bit A/D converter (SA-ADC) : input 14 channels
  - Conversion time: Min. 1.375μs/ch (When the V_{DD} is higher than 2.7V and the conversion clock is 16MHz)
  - Reference voltages are selectable from VDD pin input voltage or External reference voltage (VREF pin)
  - dedicated result register for each channel
  - Continuous conversion, Trigger start, Interrupt determining by upper limit or lower limit threshold of conversion result

- Voltage Level Supervisor (VLS) : 1 channel
  - Threshold voltage: selectable from 15 levels (1.85V to 4.00V)
  - Functional Voltage level detection reset (VLS reset) or Functional Voltage level detection interrupt (VLS0 interrupt) is generable
  - Equipped with single mode / with sampling filter / low consumption operation

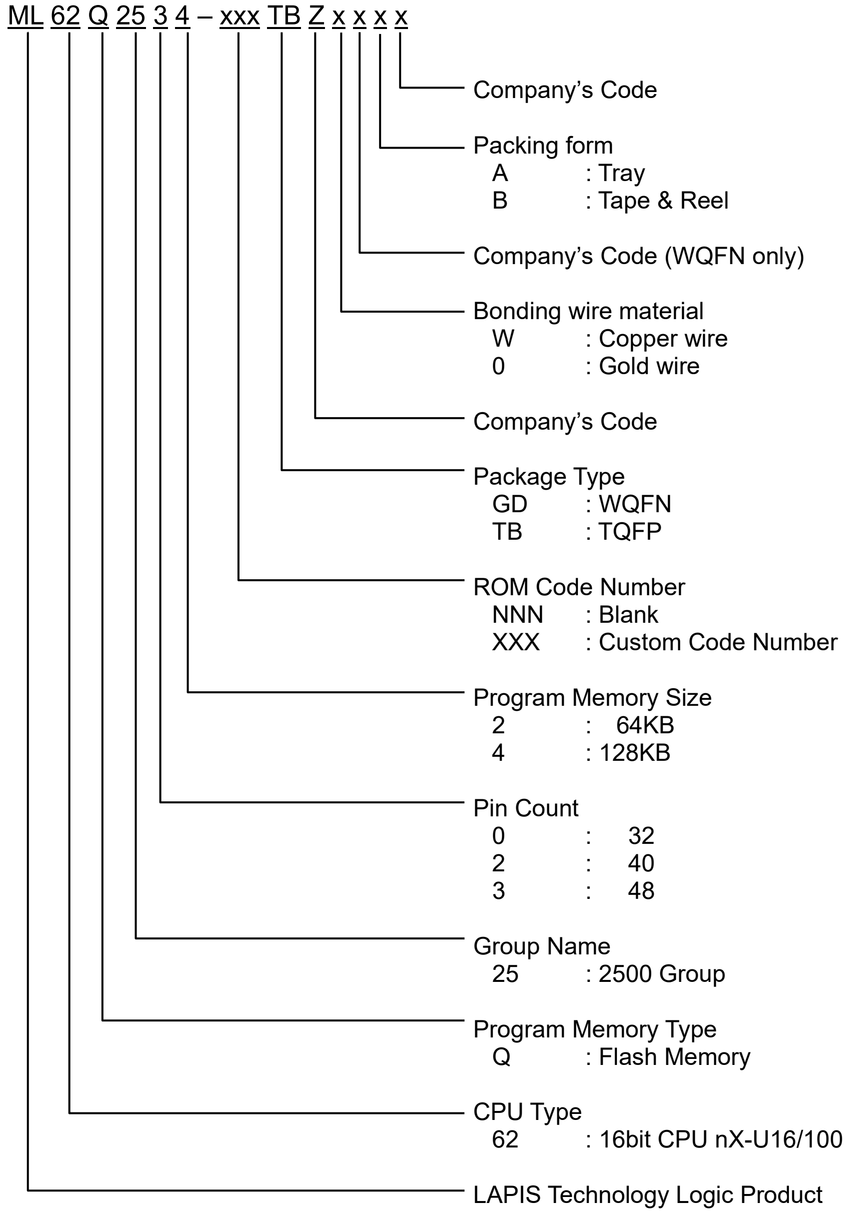
- CRC (Cyclic Redundancy Check) generator
  - Generation equation: $X^{16}+X^{12}+X^5+1$
  - Selectable from LSB first or MSB first
  - Built-in automatic program memory CRC calculation mode in HALT mode

- Safety Function
  - 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
  - Communication loop back test (UART, Synchronous serial port, I^2C bus(master))
  - GPIO test

- Shipping package

Package	Body size (including lead) [mm × mm]	Pin pitch [mm]	Packing form and Product name	
			Tray	Tape & Reel
32 pin plastic TQFP	7.0 × 7.0 (9.0 × 9.0)	0.80	ML62Q2502-xxxTBZWAX ML62Q2504-xxxTBZWAX	ML62Q2502-xxxTBZWBX ML62Q2504-xxxTBZWBX
48 pin plastic TQFP	7.0 × 7.0 (9.0 × 9.0)	0.50	ML62Q2532-xxxTBZWAX ML62Q2534-xxxTBZWAX	ML62Q2532-xxxTBZWBX ML62Q2534-xxxTBZWBX
32 pin plastic WQFN	5.0 × 5.0 (-)	0.50	ML62Q2502-xxxGDZW5AX ML62Q2504-xxxGDZW5AX	ML62Q2502-xxxGDZW5BX ML62Q2504-xxxGDZW5BX
40 pin plastic WQFN	6.0 × 6.0 (-)	0.50	ML62Q2522-xxxGDZW5AX ML62Q2524-xxxGDZW5AX	ML62Q2522-xxxGDZW5BX ML62Q2524-xxxGDZW5BX
48 pin plastic WQFN	7.0 × 7.0 (-)	0.50	ML62Q2532-xxxGDZW5AX ML62Q2534-xxxGDZW5AX	ML62Q2532-xxxGDZW5BX ML62Q2534-xxxGDZW5BX

1.1.1 How To Read The Part Number



ML 62 Q 25 3 4 – xxx TB Z x x x x

Company's Code

Packing form
A : Tray
B : Tape & Reel

Company's Code (WQFN only)

Bonding wire material
W : Copper wire
0 : Gold wire

Company's Code

Package Type
GD : WQFN
TB : TQFP

ROM Code Number
NNN : Blank
XXX : Custom Code Number

Program Memory Size
2 : 64KB
4 : 128KB

Pin Count
0 : 32
2 : 40
3 : 48

Group Name
25 : 2500 Group

Program Memory Type
Q : Flash Memory

CPU Type
62 : 16bit CPU nX-U16/100

LAPIS Technology Logic Product

Figure 1-1 Part Number

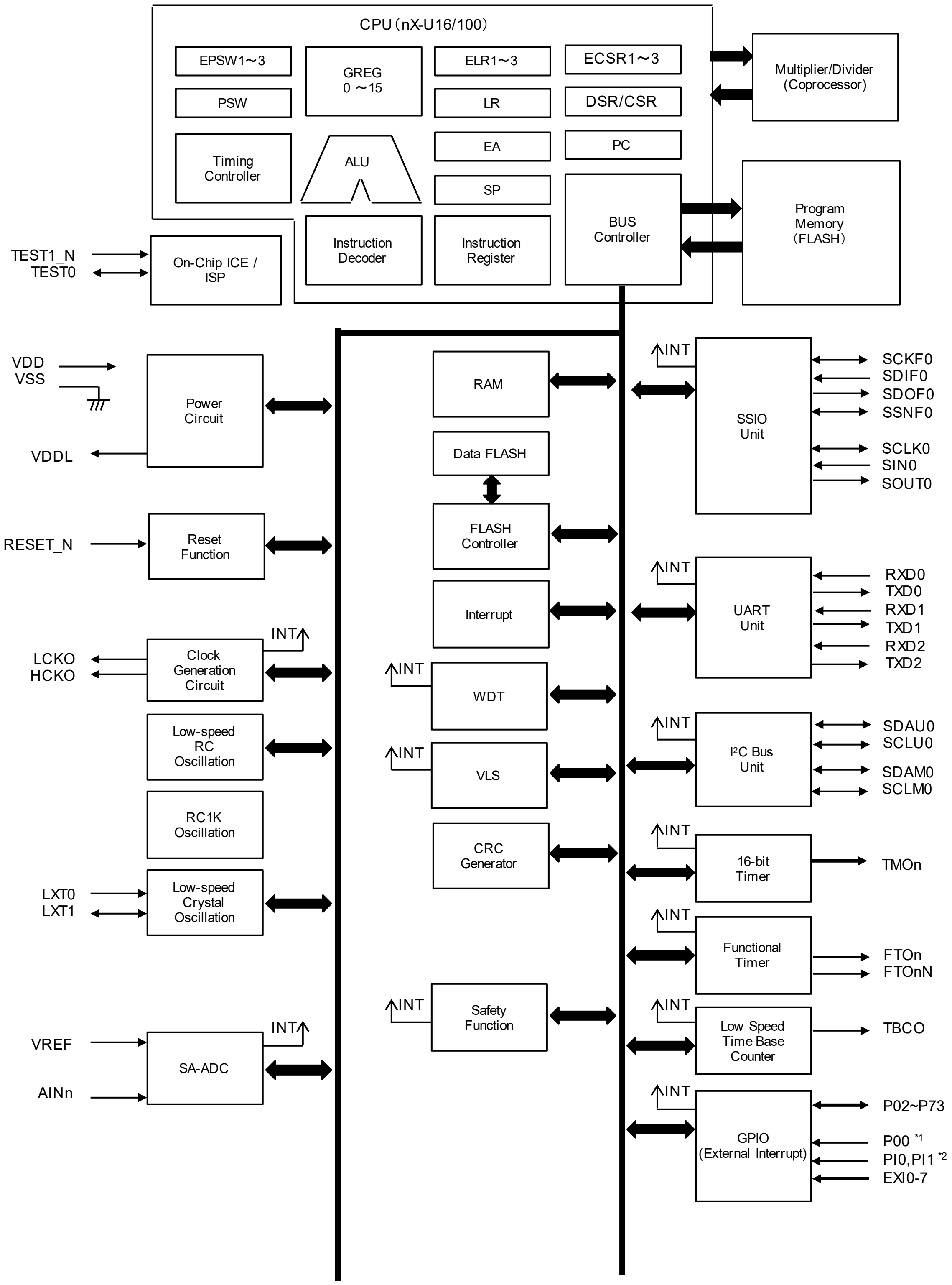
1.1.2 Main Function List

Table 1-2 Main Function List

art number	Pin							Interrupt			Timer					Communication					Analog	
	Total pin	Power pin	Reset Input pin	Debug Input port	General Purpose Input port *1	General Puroise I/O port (LED drive is supported)	External interrupt pin	External interrupt source	Non maskable interrupt source	Internal maskable interrupt source	16bit Timer [ch]	16bit Functional Timer [ch]	16bit Functional Timer [Port]	Watchdog Timer [ch]	Time base counter [ch]	Synchronous serial (without FIFO) [ch]	Synchronous serial (with FIFO) [ch]	Full-duplex UART [ch]	I^2C bus interface (Master only) [ch]	I^2C bus unit (Master/Slave) [ch]	12bit Successive type A/D converter [ch]	Voltage Level Supervisor [ch]
ML62Q2502	32	3	1	1	3	24	16	8	1	26	6	2	4	1	2	1	1	3	1	1	14	1
ML62Q2504																						
ML62Q2522	40					32	19															
ML62Q2524																						
ML62Q2532	48					40	24															
ML62Q2534																						

*1: Shared with pins for crystal oscillation and debug input.

1.2 Block Diagram



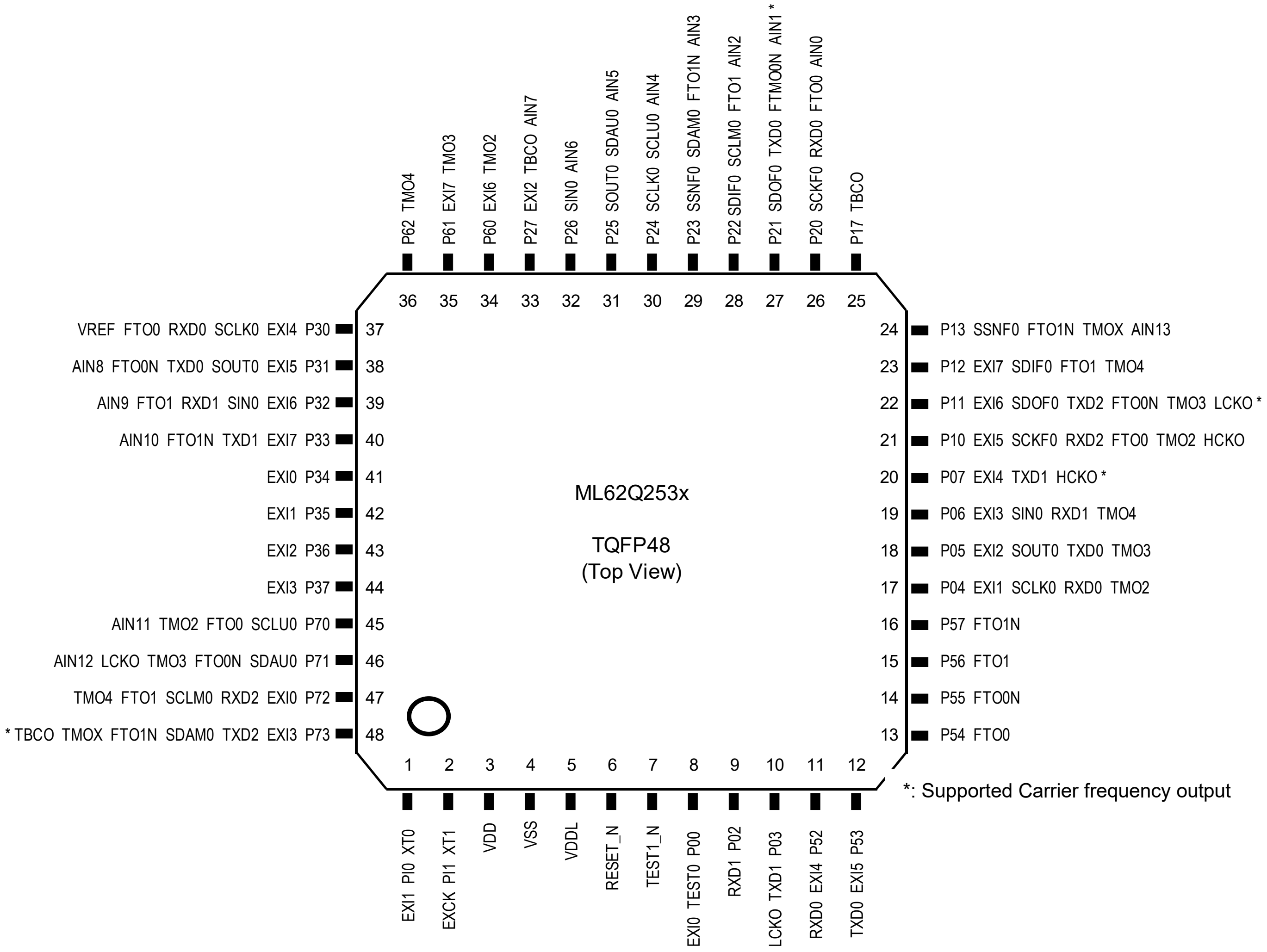
*1 : Not available as the input port when connecting to the on-chip debug emulator.
*2 : Not available as the input port when connecting to the crystal resonator.

Figure 1-2 Block Diagram of ML62Q2500 group

1.3 Pin

1.3.1 Pin Layout

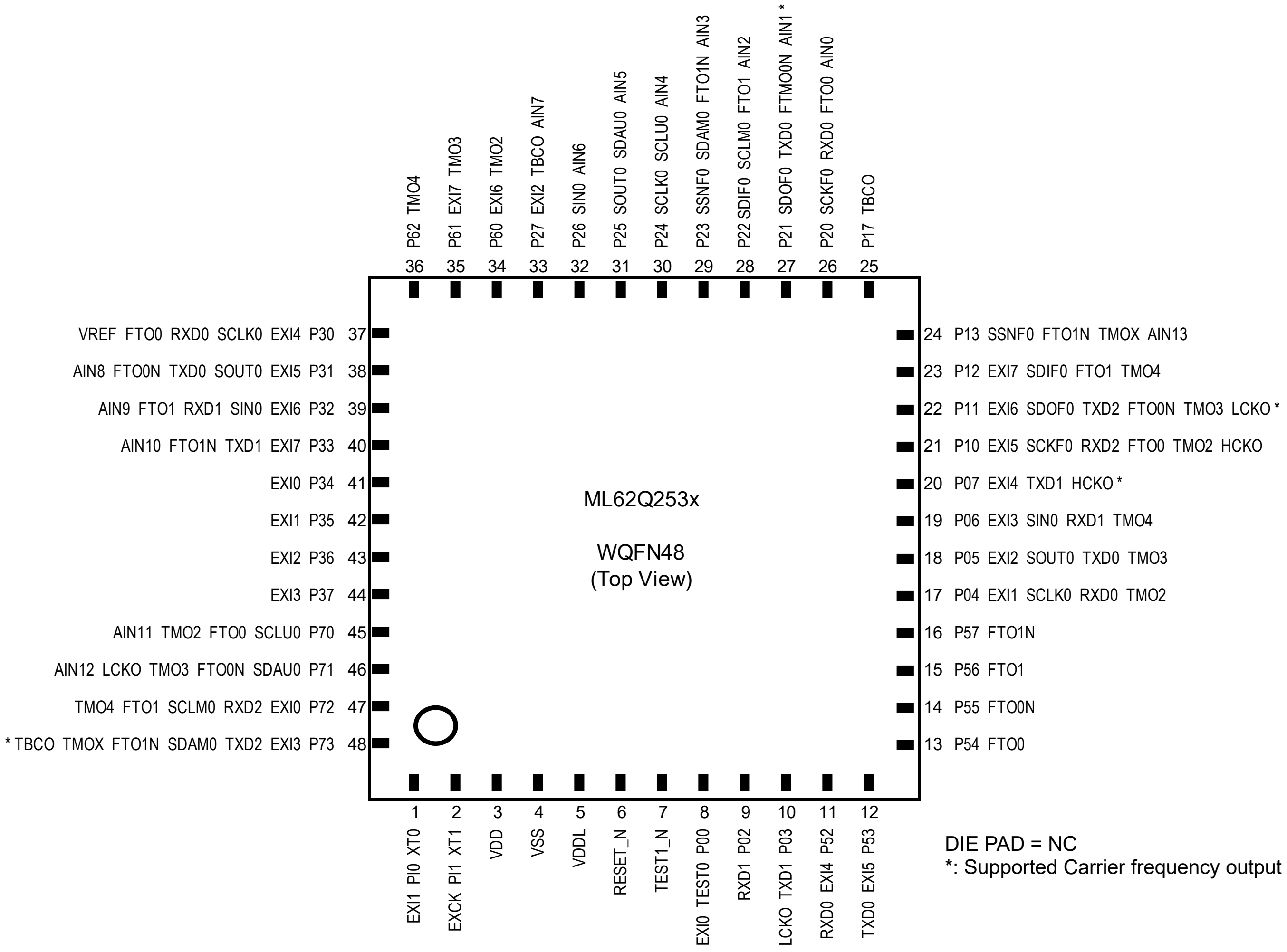
1.3.1.1 ML62Q2534/2532 : 48 pin TQFP


P62 TMO4
P61 EXI7 TMO3
P60 EXI6 TMO2
P27 EXI2 TBCO AIN7
P26 SIN0 AIN6
P25 SOUT0 SDAU0 AIN5
P24 SCLK0 SCLU0 AIN4
P23 SSNF0 SDAM0 FTO1N AIN3
P22 SDIF0 SCLM0 FTO1 AIN2
P21 SDOF0 TXD0 FTMO0N AIN1 *
P20 SCKF0 RXD0 FTO0 AIN0
P17 TBCO
36
35
34
33
32
31
30
29
28
27
26
25
VREF FTO0 RXD0 SCLK0 EXI4 P30 37
AIN8 FTO0N TXD0 SOUT0 EXI5 P31 38
AIN9 FTO1 RXD1 SIN0 EXI6 P32 39
AIN10 FTO1N TXD1 EXI7 P33 40
EXI0 P34 41
EXI1 P35 42
EXI2 P36 43
EXI3 P37 44
AIN11 TMO2 FTO0 SCLU0 P70 45
AIN12 LCKO TMO3 FTO0N SDAU0 P71 46
TMO4 FTO1 SCLM0 RXD2 EXI0 P72 47
* TBCO TMOX FTO1N SDAM0 TXD2 EXI3 P73 48
24 P13 SSNF0 FTO1N TMOX AIN13
23 P12 EXI7 SDIF0 FTO1 TMO4
22 P11 EXI6 SDOF0 TXD2 FTO0N TMO3 LCKO *
21 P10 EXI5 SCKF0 RXD2 FTO0 TMO2 HCKO
20 P07 EXI4 TXD1 HCKO *
19 P06 EXI3 SIN0 RXD1 TMO4
18 P05 EXI2 SOUT0 TXD0 TMO3
17 P04 EXI1 SCLK0 RXD0 TMO2
16 P57 FTO1N
15 P56 FTO1
14 P55 FTO0N
13 P54 FTO0
ML62Q253x
TQFP48
(Top View)
1
2
3
4
5
6
7
8
9
10
11
12
EXI1 PI0 XT0
EXCK PI1 XT1
VDD
VSS
VDDL
RESET_N
TEST1_N
EXI0 TEST0 P00
RXD1 P02
LCKO TXD1 P03
RXD0 EXI4 P52
TXD0 EXI5 P53

*: Supported Carrier frequency output

Figure 1-3-1 48 pin TQFP

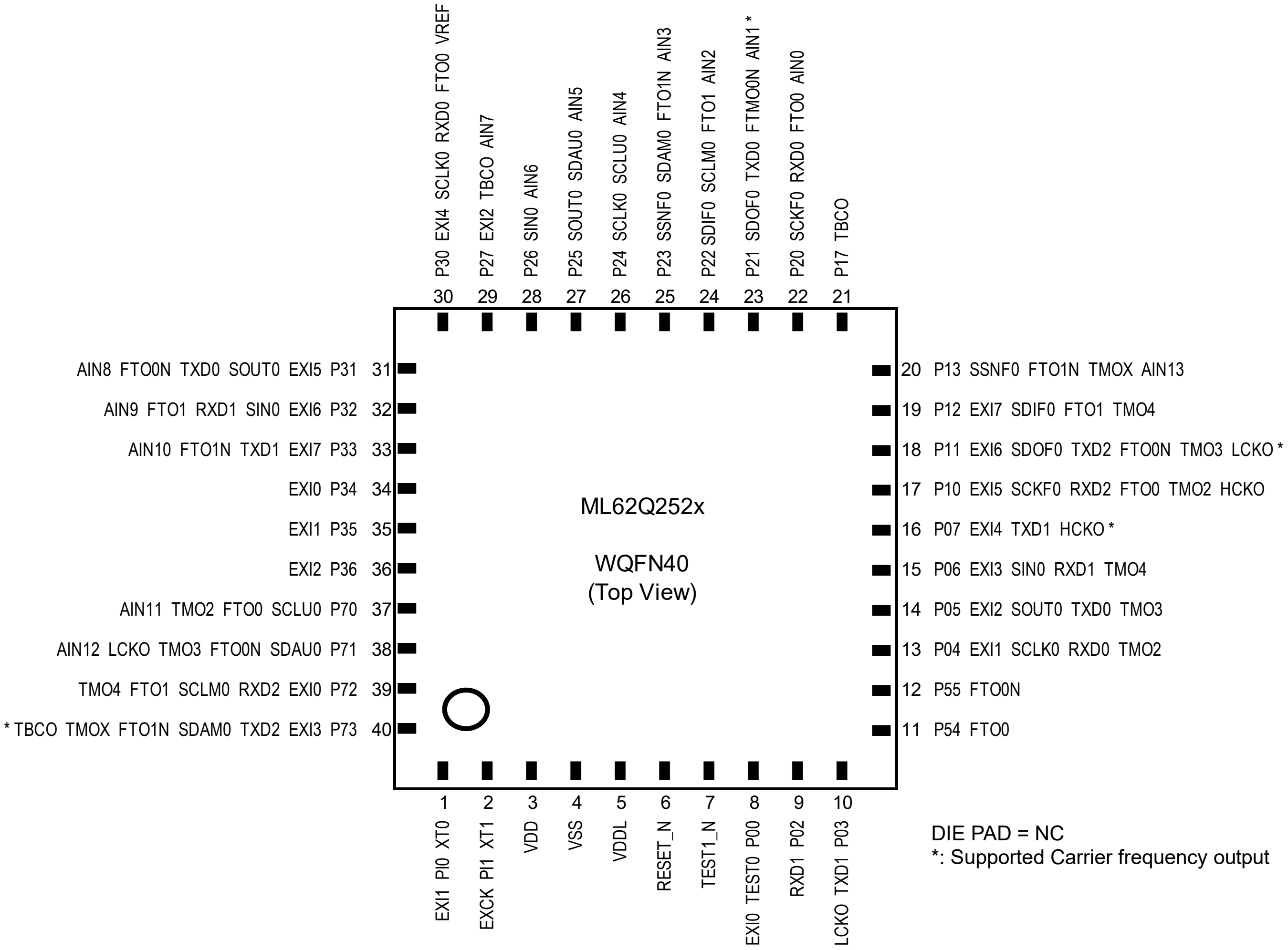
1.3.1.2 ML62Q2534/2532 : 48 pin WQFN


P62 TMO4 36
P61 EXI7 TMO3 35
P60 EXI6 TMO2 34
P27 EXI2 TBCO AIN7 33
P26 SIN0 AIN6 32
P25 SOUT0 SDAU0 AIN5 31
P24 SCLK0 SCLU0 AIN4 30
P23 SSNF0 SDAM0 FTO1N AIN3 29
P22 SDIF0 SCLM0 FTO1 AIN2 28
P21 SDOF0 TXD0 FTMO0N AIN1 * 27
P20 SCKF0 RXD0 FTO0 AIN0 26
P17 TBCO 25
VREF FTO0 RXD0 SCLK0 EXI4 P30 37
AIN8 FTO0N TXD0 SOUT0 EXI5 P31 38
AIN9 FTO1 RXD1 SIN0 EXI6 P32 39
AIN10 FTO1N TXD1 EXI7 P33 40
EXI0 P34 41
EXI1 P35 42
EXI2 P36 43
EXI3 P37 44
AIN11 TMO2 FTO0 SCLU0 P70 45
AIN12 LCKO TMO3 FTO0N SDAU0 P71 46
TMO4 FTO1 SCLM0 RXD2 EXI0 P72 47
* TBCO TMOX FTO1N SDAM0 TXD2 EXI3 P73 48
24 P13 SSNF0 FTO1N TMOX AIN13
23 P12 EXI7 SDIF0 FTO1 TMO4
22 P11 EXI6 SDOF0 TXD2 FTO0N TMO3 LCKO *
21 P10 EXI5 SCKF0 RXD2 FTO0 TMO2 HCKO
20 P07 EXI4 TXD1 HCKO *
19 P06 EXI3 SIN0 RXD1 TMO4
18 P05 EXI2 SOUT0 TXD0 TMO3
17 P04 EXI1 SCLK0 RXD0 TMO2
16 P57 FTO1N
15 P56 FTO1
14 P55 FTO0N
13 P54 FTO0
ML62Q253x
WQFN48
(Top View)
1 EXI1 PI0 XT0
2 EXCK PI1 XT1
3 VDD
4 VSS
5 VDDL
6 RESET_N
7 TEST1_N
8 EXI0 TEST0 P00
9 RXD1 P02
10 LCKO TXD1 P03
11 RXD0 EXI4 P52
12 TXD0 EXI5 P53

DIE PAD = NC
*: Supported Carrier frequency output

Figure 1-3-2 48 pin WQFN

1.3.1.3 ML62Q2524/2522 : 40 pin WQFN


ML62Q252x
WQFN40
(Top View)
30 P30 EXI4 SCLK0 RXD0 FTO0 VREF
29 P27 EXI2 TBCO AIN7
28 P26 SIN0 AIN6
27 P25 SOUT0 SDAU0 AIN5
26 P24 SCLK0 SCLU0 AIN4
25 P23 SSNF0 SDAM0 FTO1N AIN3
24 P22 SDIF0 SCLM0 FTO1 AIN2
23 P21 SDOF0 TXD0 FTMO0N AIN1 *
22 P20 SCKF0 RXD0 FTO0 AIN0
21 P17 TBCO
AIN8 FTO0N TXD0 SOUT0 EXI5 P31 31
AIN9 FTO1 RXD1 SIN0 EXI6 P32 32
AIN10 FTO1N TXD1 EXI7 P33 33
EXI0 P34 34
EXI1 P35 35
EXI2 P36 36
AIN11 TMO2 FTO0 SCLU0 P70 37
AIN12 LCKO TMO3 FTO0N SDAU0 P71 38
TMO4 FTO1 SCLM0 RXD2 EXI0 P72 39
* TBCO TMOX FTO1N SDAM0 TXD2 EXI3 P73 40
20 P13 SSNF0 FTO1N TMOX AIN13
19 P12 EXI7 SDIF0 FTO1 TMO4
18 P11 EXI6 SDOF0 TXD2 FTO0N TMO3 LCKO *
17 P10 EXI5 SCKF0 RXD2 FTO0 TMO2 HCKO
16 P07 EXI4 TXD1 HCKO *
15 P06 EXI3 SIN0 RXD1 TMO4
14 P05 EXI2 SOUT0 TXD0 TMO3
13 P04 EXI1 SCLK0 RXD0 TMO2
12 P55 FTO0N
11 P54 FTO0
1 EXI1 PI0 XT0
2 EXCK PI1 XT1
3 VDD
4 VSS
5 VDDL
6 RESET_N
7 TEST1_N
8 EXI0 TEST0 P00
9 RXD1 P02
10 LCKO TXD1 P03

DIE PAD = NC
*: Supported Carrier frequency output

Figure 1-3-3 40 pin WQFN

1.3.1.4 ML62Q2504/2502 : 32 pin TQFP

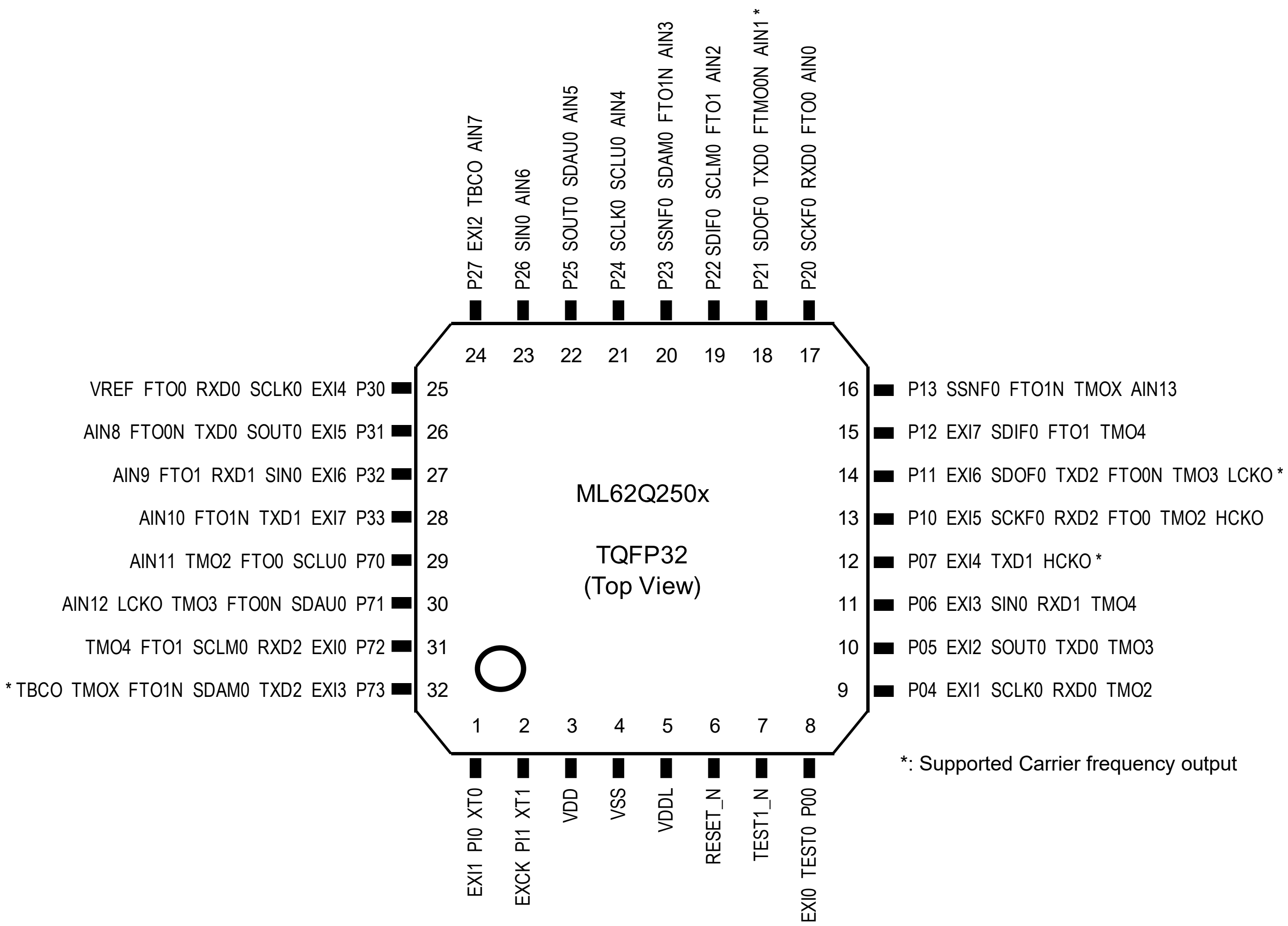
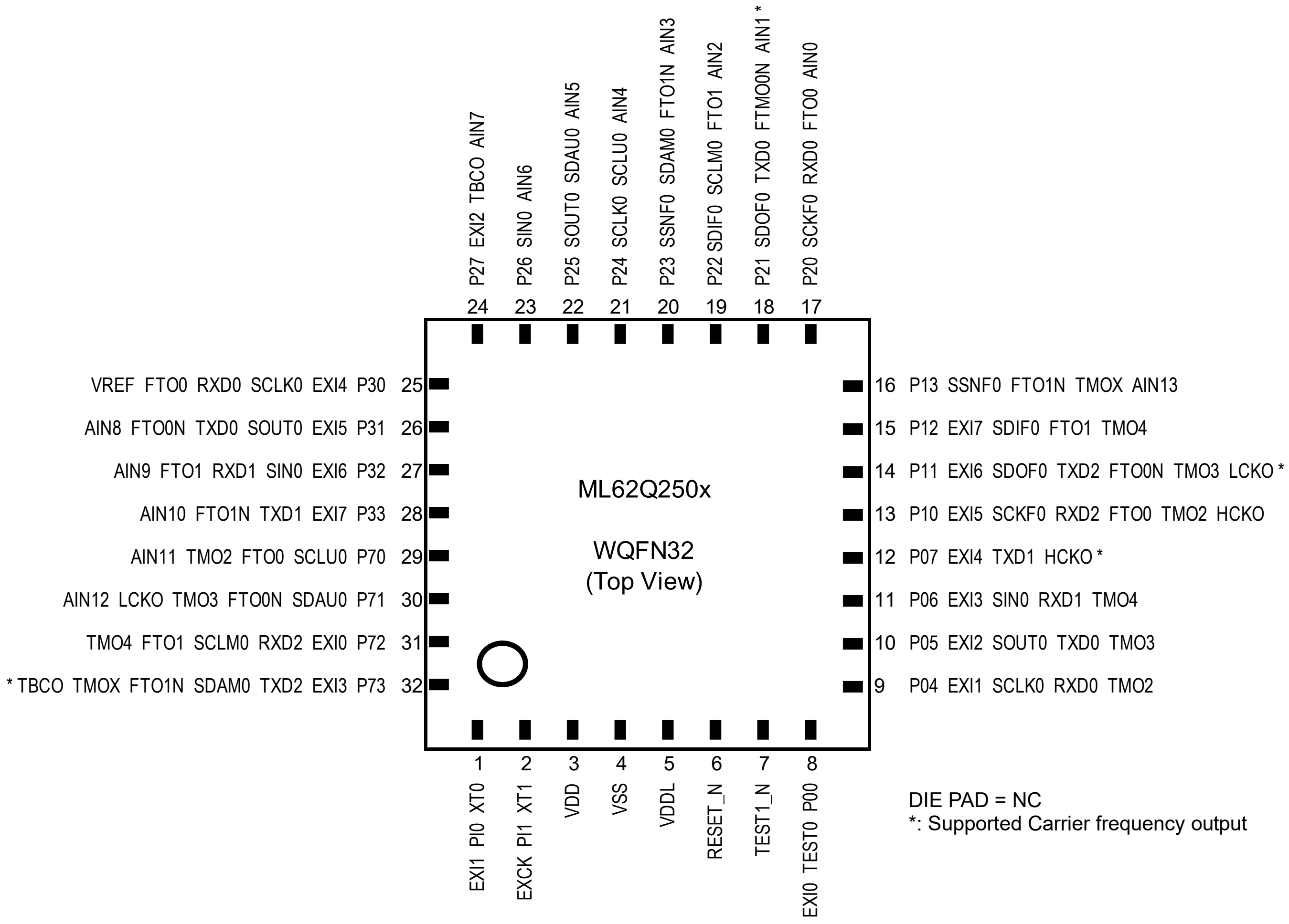

P27 EXI2 TBCO AIN7
P26 SIN0 AIN6
P25 SOUT0 SDAU0 AIN5
P24 SCLK0 SCLU0 AIN4
P23 SSNF0 SDAM0 FTO1N AIN3
P22 SDIF0 SCLM0 FTO1 AIN2
P21 SDOF0 TXD0 FTMO0N AIN1 *
P20 SCKF0 RXD0 FTO0 AIN0
24 23 22 21 20 19 18 17
VREF FTO0 RXD0 SCLK0 EXI4 P30 25
AIN8 FTO0N TXD0 SOUT0 EXI5 P31 26
AIN9 FTO1 RXD1 SIN0 EXI6 P32 27
AIN10 FTO1N TXD1 EXI7 P33 28
AIN11 TMO2 FTO0 SCLU0 P70 29
AIN12 LCKO TMO3 FTO0N SDAU0 P71 30
TMO4 FTO1 SCLM0 RXD2 EXI0 P72 31
* TBCO TMOX FTO1N SDAM0 TXD2 EXI3 P73 32
16 P13 SSNF0 FTO1N TMOX AIN13
15 P12 EXI7 SDIF0 FTO1 TMO4
14 P11 EXI6 SDOF0 TXD2 FTO0N TMO3 LCKO *
13 P10 EXI5 SCKF0 RXD2 FTO0 TMO2 HCKO
12 P07 EXI4 TXD1 HCKO *
11 P06 EXI3 SIN0 RXD1 TMO4
10 P05 EXI2 SOUT0 TXD0 TMO3
9 P04 EXI1 SCLK0 RXD0 TMO2
ML62Q250x
TQFP32
(Top View)
1 2 3 4 5 6 7 8
EXI1 PI0 XT0
EXCK PI1 XT1
VDD
VSS
VDDL
RESET_N
TEST1_N
EXI0 TEST0 P00
*: Supported Carrier frequency output

Figure 1-3-4 32 pin TQFP

1.3.1.5 ML62Q2504/2502 : 32 pin WQFN


P27 EXI2 TBCO AIN7 24
P26 SIN0 AIN6 23
P25 SOUT0 SDAU0 AIN5 22
P24 SCLK0 SCLU0 AIN4 21
P23 SSNF0 SDAM0 FTO1N AIN3 20
P22 SDIF0 SCLM0 FTO1 AIN2 19
P21 SDOF0 TXD0 FTMO0N AIN1 * 18
P20 SCKF0 RXD0 FTO0 AIN0 17
VREF FTO0 RXD0 SCLK0 EXI4 P30 25
AIN8 FTO0N TXD0 SOUT0 EXI5 P31 26
AIN9 FTO1 RXD1 SIN0 EXI6 P32 27
AIN10 FTO1N TXD1 EXI7 P33 28
AIN11 TMO2 FTO0 SCLU0 P70 29
AIN12 LCKO TMO3 FTO0N SDAU0 P71 30
TMO4 FTO1 SCLM0 RXD2 EXI0 P72 31
* TBCO TMOX FTO1N SDAM0 TXD2 EXI3 P73 32
ML62Q250x
WQFN32
(Top View)
16 P13 SSNF0 FTO1N TMOX AIN13
15 P12 EXI7 SDIF0 FTO1 TMO4
14 P11 EXI6 SDOF0 TXD2 FTO0N TMO3 LCKO *
13 P10 EXI5 SCKF0 RXD2 FTO0 TMO2 HCKO
12 P07 EXI4 TXD1 HCKO *
11 P06 EXI3 SIN0 RXD1 TMO4
10 P05 EXI2 SOUT0 TXD0 TMO3
9 P04 EXI1 SCLK0 RXD0 TMO2
1 EXI1 PI0 XT0
2 EXCK PI1 XT1
3 VDD
4 VSS
5 VDDL
6 RESET_N
7 TEST1_N
8 EXI0 TEST0 P00

DIE PAD = NC
*: Supported Carrier frequency output

Figure 1-3-5 32 pin WQFN

1.3.2 Pin List

Table 1-3 shows the pin list of ML62Q2500 group.

Table 1-3 ML62Q2500 Group Pin List

Pin No.			LSI Pin Name (Primary function)	Career frequency output	1st function	1st function	2nd function	3rd function	4th function	5th function	6th function	7th function
ML62Q250x	ML62Q252x	ML62Q253x			GPI/ EXI	ADC	SSIO*/**	UART	I2C*	FTM**	Timer	CLKOUT/ LTBC**
1	1	1	XT0	—	PI0/EXI1	—	—	—	—	—	—	—
2	2	2	XT1	—	PI1	—	—	—	—	—	—	—
3	3	3	VDD	—	—	—	—	—	—	—	—	—
4	4	4	VSS	—	—	—	—	—	—	—	—	—
5	5	5	VDDL	—	—	—	—	—	—	—	—	—
6	6	6	RESET_N	—	—	—	—	—	—	—	—	—
7	7	7	TEST1_N	—	—	—	—	—	—	—	—	—
8	8	8	P00/TEST0	—	EXI0	—	—	—	—	—	—	—
—	9	9	P02	—	—	—	—	RXD1	—	—	—	—
—	10	10	P03	—	—	—	—	TXD1	—	—	—	LCKO
9	13	17	P04	—	EXI1	—	SCLK0-0	RXD0	—	—	TMO2	—
10	14	18	P05	—	EXI2	—	SOUT0-0	TXD0	—	—	TMO3	—
11	15	19	P06	—	EXI3	—	SIN0-0	RXD1	—	—	TMO4	—
12	16	20	P07	●	EXI4	—	—	TXD1	—	—	—	HCKO
13	17	21	P10	—	EXI5	—	SCKF0-0	RXD2	—	FTO0	TMO2	HCKO
14	18	22	P11	●	EXI6	—	SDOF0-0	TXD2	—	FTO0N	TMO3	LCKO
15	19	23	P12	—	EXI7	—	SDIF0-0	—	—	FTO1	TMO4	—
16	20	24	P13	—	—	AIN13	SSNF0-0	—	—	FTO1N	TMOX	—
—	21	25	P17	—	—	—	—	—	—	—	—	TBCO
17	22	26	P20	—	—	AIN0	SCKF0-1	RXD0	—	FTO0	—	—
18	23	27	P21	●	—	AIN1	SDOF0-1	TXD0	—	FTO0N	—	—
19	24	28	P22	—	—	AIN2	SDIF0-1	—	SCLM0-0	FTO1	—	—
20	25	29	P23	—	—	AIN3	SSNF0-1	—	SDAM0-0	FTO1N	—	—
21	26	30	P24	—	—	AIN4	SCLK0-1	—	SCLU0-0	—	—	—
22	27	31	P25	—	—	AIN5	SOUT0-1	—	SDAU0-0	—	—	—
23	28	32	P26	—	—	AIN6	SIN0-1	—	—	—	—	—
24	29	33	P27	—	EXI2	AIN7	—	—	—	—	—	TBCO
25	30	37	P30	—	EXI4	VREF	SCLK0-2	RXD0	—	FTO0	—	—
26	31	38	P31	—	EXI5	AIN8	SOUT0-2	TXD0	—	FTO0N	—	—
27	32	39	P32	—	EXI6	AIN9	SIN0-2	RXD1	—	FTO1	—	—
28	33	40	P33	—	EXI7	AIN10	—	TXD1	—	FTO1N	—	—
—	34	41	P34	—	EXI0	—	—	—	—	—	—	—
—	35	42	P35	—	EXI1	—	—	—	—	—	—	—
—	36	43	P36	—	EXI2	—	—	—	—	—	—	—
—	—	44	P37	—	EXI3	—	—	—	—	—	—	—
—	—	11	P52	—	EXI4	—	—	RXD0	—	—	—	—
—	—	12	P53	—	EXI5	—	—	TXD0	—	—	—	—
—	11	13	P54	—	—	—	—	—	—	FTO0	—	—
—	12	14	P55	—	—	—	—	—	—	FTO0N	—	—
—	—	15	P56	—	—	—	—	—	—	FTO1	—	—
—	—	16	P57	—	—	—	—	—	—	FTO1N	—	—
—	—	34	P60	—	EXI6	—	—	—	—	—	TMO2	—
—	—	35	P61	—	EXI7	—	—	—	—	—	TMO3	—
—	—	36	P62	—	—	—	—	—	—	—	TMO4	—
29	37	45	P70	—	—	AIN11	—	—	SCLU0-1	FTO0	TMO2	—
30	38	46	P71	—	—	AIN12	—	—	SDAU0-1	FTO0N	TMO3	LCKO
31	39	47	P72	—	EXI0	—	—	RXD2	SCLM0-1	FTO1	TMO4	—
32	40	48	P73	●	EXI3	—	—	TXD2	SDAM0-1	FTO1N	TMOX	TBCO
DIE	DIE	DIE	NC	—	—	—	—	—	—	—	—	—

*: The SSIO and I^{2}C use with a combination of the same suffix pins.

**: Assign each function; SCLK0/SCKF0/FTOn/FTOnN/HCKO, to only one LSI pin.

1.3.3 Pin Description

Table 1-4 shows the pin list categorized by the function.
"-" : Power pin, "I": Input pin, "O" Output pin and "I/O" : input/output pin

Table 1-4 Pin Description (1/2)

Function	Functional pin name	LSI pin name	I/O	Description
Power	—	VSS	—	Negative power supply pin (-) Define this terminal potential as V_{SS}
	—	VDD	—	Positive power supply pin (+). Connect a capacitor C_V (more than 1uF) between this pin and VSS. Define this terminal potential as $V_{DD.}$
	—	VDDL	—	Power supply for internal logic (internal regulator's output). Connect a capacitor C_L (1μF) between this pin and VSS.
Debug ISP	TEST0	P00/ TEST0	I/O	Input/output for testing This pin which is shared with P00 is used as on-chip debug interface and ISP function and is initialized as pull-up input mode by the system reset.
	TEST1_N	TEST1_N	I	Input for testing This pin is used as on-chip debug interface and ISP function and is initialized as pull-up input mode by the system reset.
Reset	RESET_N	RESET_N	I	Reset input. Appling this pin "L" level shifts MCU to system reset mode. Appling this pin "H" level shifts MCU to program running mode. No pull-up resistor is installed.
General input port (GPI)	PI0, PI1	XT0, XT1	I	General purpose input. - High-impedance (initial value) - Input without Pull-up
	P00	P00/ TEST0	I	General purpose input. - Input with Pull-up (initial value) - Input without Pull-up Not available as general inputs when using the on-chip debug interface or ISP function.
General port (GPIO)	P02～P07	P02～P07	I/O	General purpose input/output - High-impedance (initial value) - Input with Pull-up - Input without Pull-up - CMOS output - N channel (N-ch) open drain output
	P10～P17	P10～P17		
	P20～P27	P20～P27		
	P30～P37	P30～P37		
	P52～P57	P50～P57		
	P60～P62	P60～P62		
	P70～P73	P70～P73		
Clock Input	XT0	XT0	I	Connect a Low speed(32.768kHz) crystal resonator and connect capacitors between the pin and VSS. When inputting a square wave clock, input from XT1 pin
	XT1	XT1	I/O	
Clock Output (7th function)	HCKO	P07 P10	O	High-speed clock output.
	LCKO	P03 P11 P71		Low-speed clock output.
	TBCO	P17 P27 P73		Low-speed time base counter output.
Career frequency output	—	P07 P11 P21 P73	O	Career frequency output
External Interrupt (1st function)	EXI0	P00 P72 P34		External Maskable Interrupt 0 Input
	EXI1	P04 XT0 P35		External Maskable Interrupt 1 Input
	EXI2	P05 P27 P36		External Maskable Interrupt 2 Input
	EXI3	P06 P73 P37		External Maskable Interrupt 3 Input
	EXI4	P07 P30 P52		External Maskable Interrupt 4 Input
	EXI5	P10 P31 P53		External Maskable Interrupt 5 Input
	EXI6	P11 P32 P60		External Maskable Interrupt 6 Input
	EXI7	P12 P33 P61		External Maskable Interrupt 7 Input

Table 1-4 Pin Description (2/2)

Function	Functional pin name	LSI pin name	I/O	Description
16bit General Timer (6th function)	TMO2	P04 P10 P60 P70	O	16bit General Timer 2 output
	TMO3	P05 P11 P61 P71		16bit General Timer 3 output
	TMO4	P06 P12 P62 P72		16bit General Timer 4 output
	TMOX	P13 P73		16bit General Timer X output
Functional Timer (5th func.)	FTO0	P10 P20 P30 P54 P70	O	Functional Timer0 P output
	FTO0N	P11 P21 P31 P55 P71		Functional Timer0 N output
	FTO1	P12 P22 P32 P56 P72		Functional Timer1 P output
	FTO1N	P13 P23 P33 P57 P73		Functional Timer1 N output
I^{2}C Bus (4th function)	SCLU0	P24 P70	I/O	I^{2}C Unit0 Clock input/output
	SDAU0	P25 P71		I^{2}C Unit0 Data input/output
	SCLM0	P22 P72		I^{2}C Master0 Clock input/output
	SDAM0	P23 P73		I^{2}C Master0 Data input/output
UART (3rd function)	RXD0	P04 P20 P30 P52	I	UART0 received data input
	TXD0	P05 P21 P31 P53	O	UART0 transmission data output
	RXD1	P02 P06 P32	I	UART1 received data input
	TXD1	P03 P07 P33	O	UART1 transmission data output
	RXD2	P10 P72	I	UART2 received data input
	TXD2	P11 P73	O	UART2 transmission data output
Synchronous Serial Port (2nd function)	SCKF0	P10 P20	I/O	Synchronous serial0 (with FIFO) clock input/output
	SDIF0	P12 P22	I	Synchronous serial0 (with FIFO) data input
	SDOF0	P11 P21	O	Synchronous serial0 (with FIFO) data output
	SSNF0	P13 P23	I/O	Synchronous serial0 (with FIFO) slave select input/output
	SCLK0	P04 P24 P30	I/O	Synchronous serial0 clock input/output
	SIN0	P06 P26 P32	I	Synchronous serial0 data input
	SOUT0	P05 P25 P31	O	Synchronous serial0 data output
Successive approximation type A/D converter (SA-ADC) (1st function)	VREF	P30	I	SA-ADC external reference voltage input Define the potential of reference voltage for SA-ADC as V_{REF}
	AIN0~AIN13	P13 P27-P20 P33-P31 P71-P70		SA-ADC channel 0 to 13 analog input

1.3.4 Termination of Unused Pins

Table 1-5 shows the processing of unused pins.

Table 1-5 Termination of unused pins

Pin	pin termination
NC	Open
RESET_N	Connect to VDD
TEST1_N	Connect to VDD
P00/TEST0	Open the pin with the initial condition of pulled-up input mode
XT0, XT1	Open the pins with the initial condition of Hi-impedance (input/output invalid) mode.
P02 ~ P07 P10 ~ P17 P20 ~ P27 P30 ~ P37 P52 ~ P57 P60 ~ P62 P70 ~ P73	

[Note]

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

Chapter 2 CPU and Memory Space

2. CPU and Memory Space

2.1 General Description

ML62Q2500 group 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

Product name	Program memory space	Data memory space	Data flash size	Memory model
	ROM size	RAM size		
ML62Q2502 ML62Q2522 ML62Q2532	64 Kbyte	8 Kbyte	4 Kbyte	SMALL
ML62Q2504 ML62Q2524 ML62Q2534	128 Kbyte			LARGE

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

ML62Q2500 group 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 2-2 shows maximum operating frequency of high-speed clock, peripheral circuit and CPU. See Chapter 30 "Code Option" for details on how to set the Code Option.

Table 2-2 Maximum Operating Frequency

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
1MHz	1MHz	1MHz	1MHz

- 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.
- No-wait mode
  This mode allows the CPU to directly execute instruction codes read from the program memory without involving the buffer. This mode minimizes the number of instruction execution cycles.

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

SFR |= 0x0081;
```

Converted to RB SFR.0 ;

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

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

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

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

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

2.3 Coprocessor

ML62Q2500 group has the built-in multiplier/divider in the coprocessor.
The multiplier/divider is operated using coprocessor data transfer instructions of the CPU. For coprocessor data transfer instructions, see "nX-U16/100 Core Instruction Manual".

2.3.1 Multiplier/Divider

The multiplier/divider has following arithmetic functions:

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

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

2.3.2 List of Coprocessor General-purpose Registers

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

Table 2-3 List of coprocessor general-purpose registers

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

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

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

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

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

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

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

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

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

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

Table 2-4 assignment of an input and results

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

"-" indicates that the previous value is retained.

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

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

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

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

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

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

Table 2-5 Configured CR8

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

Table 2-6 Flag of CR9

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

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

2.3.2.3 Coprocessor ID Register (CR15)

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 bit
Initial value: 0x8100

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

2.3.3 How to Use Multiplier/Divider

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.

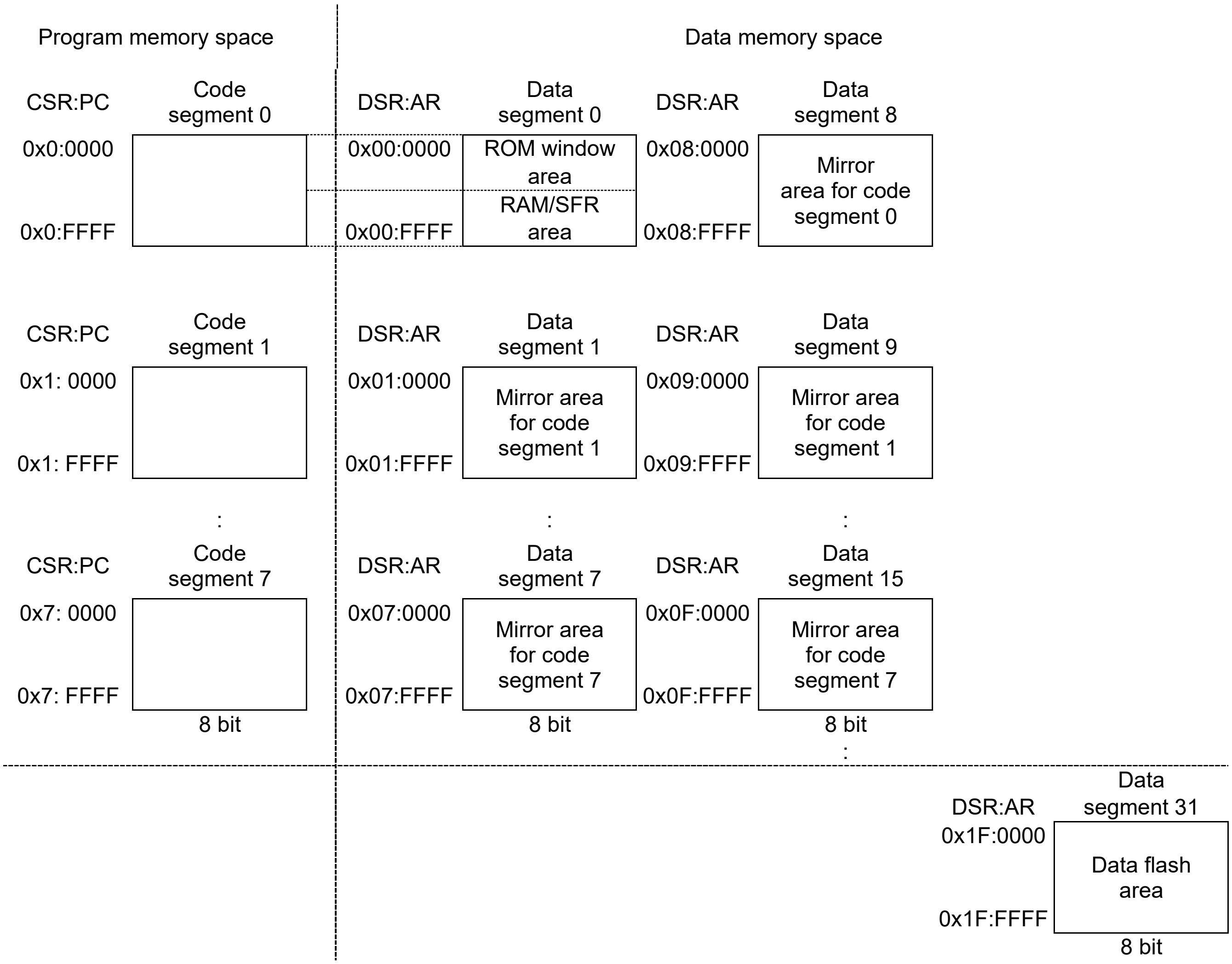


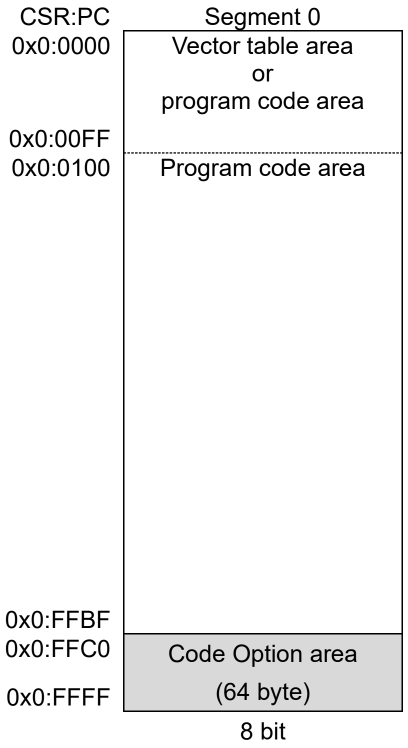
Figure 2-1 General Scheme of Memory Space

2.5 Program Memory Space

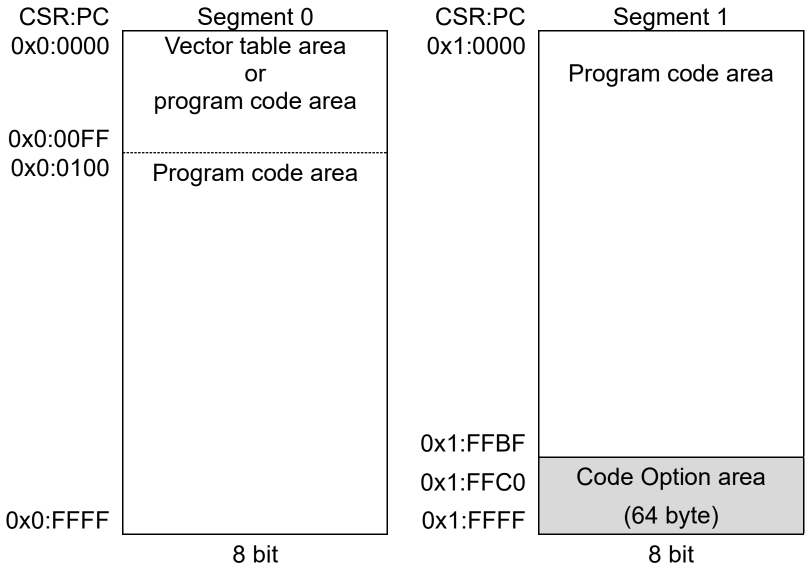
The program memory space is an area to store the program code, vector table, and Code Options.
The program memory space is specified by 20 bits (CSR:PC) consisting of higher 4 bits as code segment register (CSR) and lower 16 bits as program counter (PC).
The vector table area is used as the reset vector, 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 show the program memory space configuration of each product of the ML62Q2500 series.

[Note]

- **CSR[3] is unused on the ML62Q2500 group. 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 30 "Code Option" and make sure the setting data is correct.**
- **It is recommended to fill unused areas with data "0xFFFF" (BRK instruction) in the program memory space to ensure failsafe using the generation tool of the ROM code data. See its manual for details on how to use. See "nX-U16/100 Core Instruction Manual" for details of the BRK instruction.**
- **Do not read or program unused areas to prevent the CPU works incorrectly.**



ML62Q2502 / 2522 / 2532
Size 64 Kbyte
Figure 2-2-1 Configuration of Program Memory Space 1



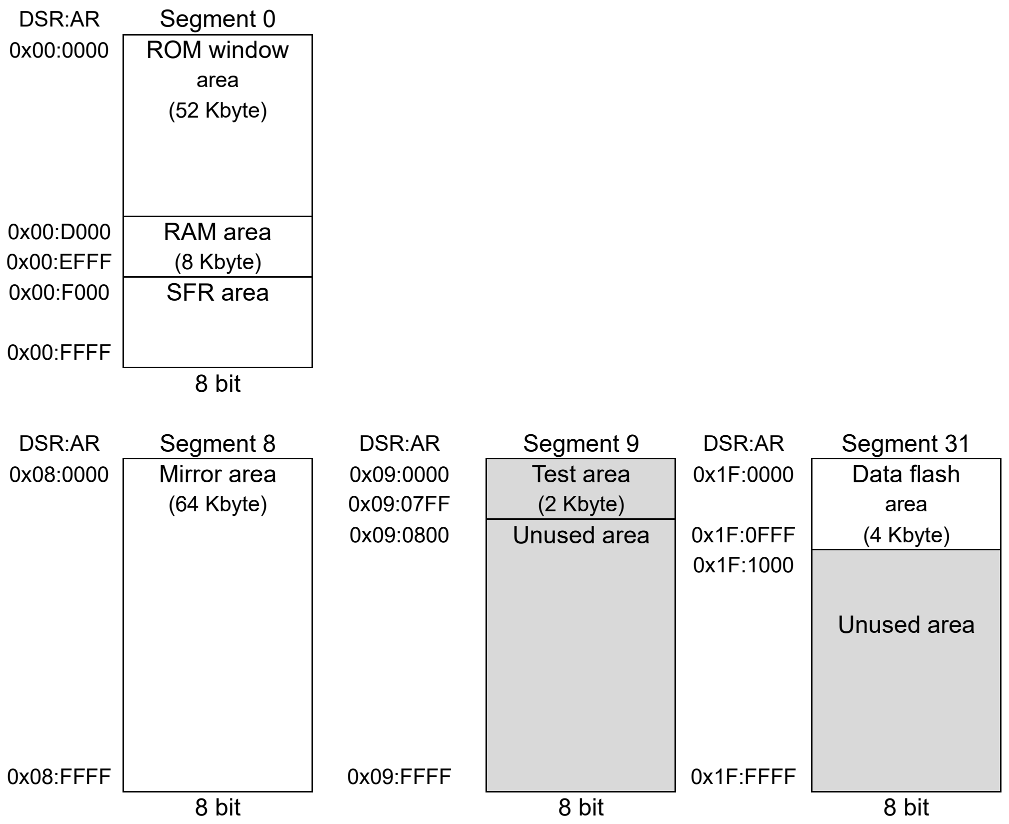
ML62Q2504 / 2524 / 2534
Size 128 Kbyte
Figure 2-2-2 Configuration of Program Memory Space 2

2.6 Data Memory Space

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

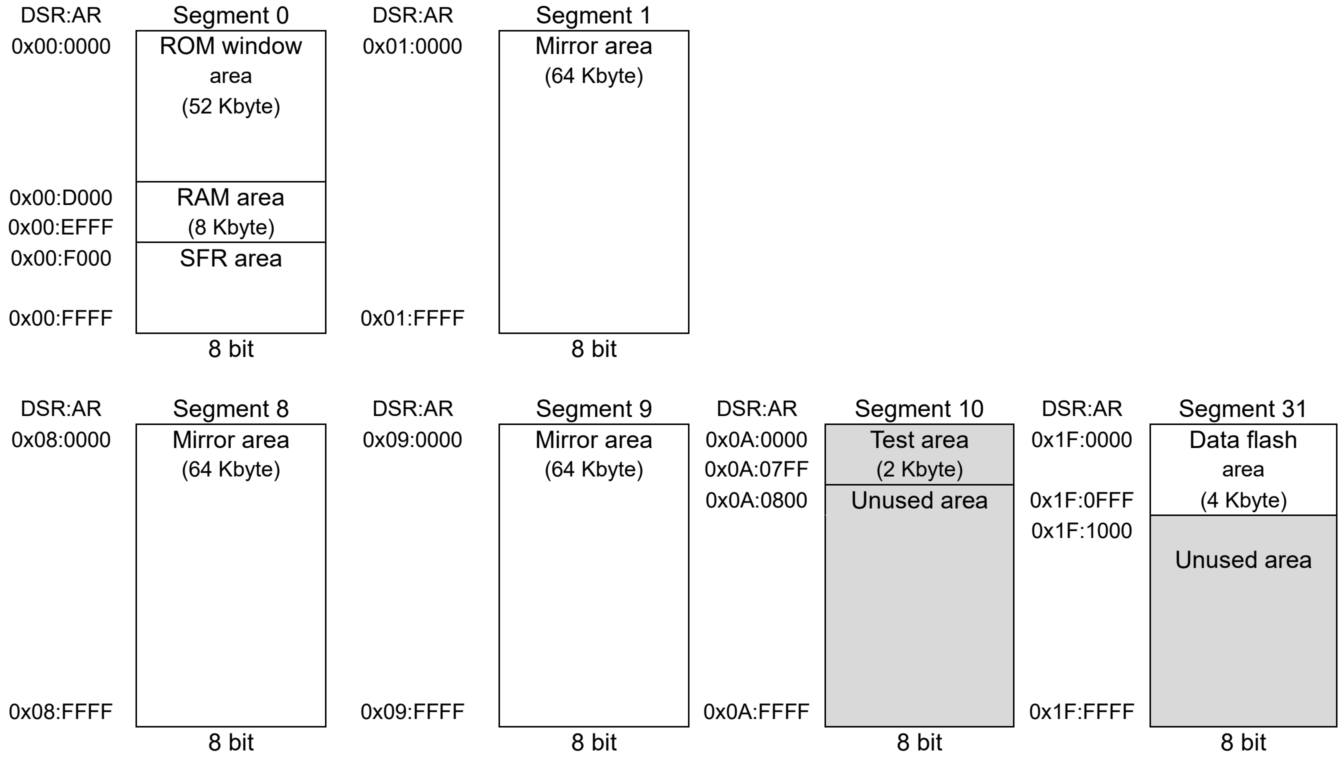
[Note]

- **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.**



ML62Q2502 / 2522 / 2532

Figure 2-3-1 Configuration of Data Memory Space 1



ML62Q2504 / 2524 / 2534

Figure 2-3-2 Configuration of Data Memory Space 2

2.7 Description of Registers

2.7.1 List of Registers

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

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

2.7.2 Data Segment Register (DSR)

DSR is a 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)

2.7.3 Flash Remap Address Register (REMAPADD)

REMAPADD is a 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	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1	*1

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

Bit No.	Bit symbol name	Description
7	-	Reserved bit
6 to 4	RES2 to RES0	Bits to set the code segment of the area to remap. The RES2 and RES1 bits are reserved.
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.

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

2.7.4 Reserved register 1

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

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

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

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

2.7.5 Reserved register 2

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

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

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

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

2.8 Remapping Function

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

Two ways are available to start the remap function.

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

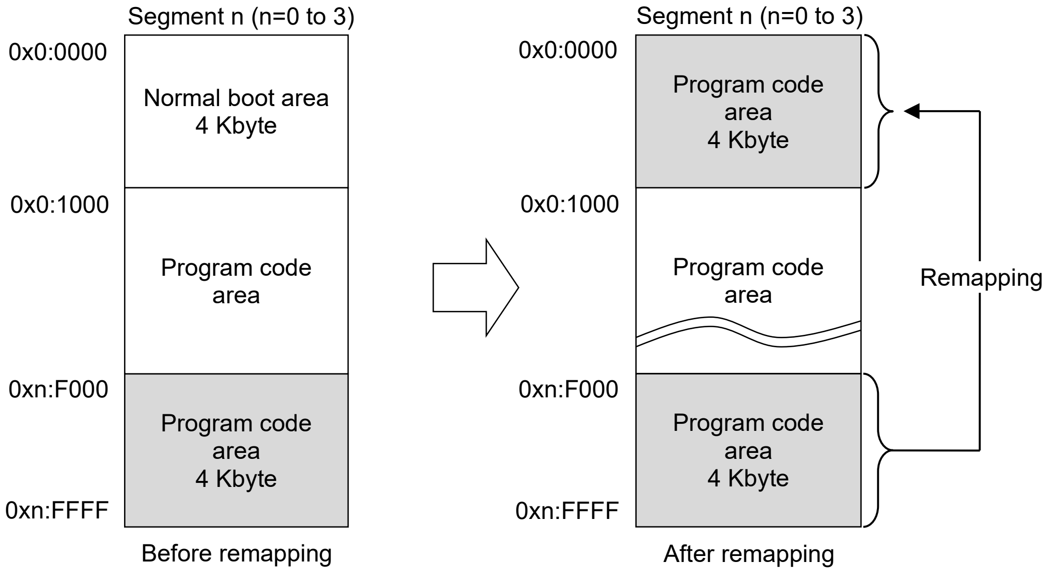


Figure 2-4 Memory Map Before and After Remapping of Program Memory Space

2.8.1 Description of Remapping Function

The remapping function allows the normal boot area of addresses 0x0:0000 to 0x0:0FFF (4 Kbytes) to be replaced (remapped) with the arbitrary 4 Kbyte area set in 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 IAP Sample Program supplied by LAPIS, 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 30 "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 2-5 below shows an example of the program script of software remapping.

<If the beginning address of the area to be remapped is 0x1:F000>

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

Figure 2-5 Program Script Example of Software Remapping

[Note]

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

2.8.3 Code Option Remap

at the system reset on the remap condition.

- If setting both REMAPMD and CREMAPMD to "0", the LSI starts running at the address set in 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.

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

Table 2-7 CPU address at releasing reset

Reset	REMAPMD	CREMAPMD	CPU instruction execution start address
CPU reset (BRK instruction)	1	1	0x0000
	1	0	
	0	1	Address set in 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

ML62Q2500 group has a function to reset the CPU, peripheral circuits and other hardware due to the causes described in Table 3-1.
This chapter describes the system reset mode, reset input pin reset and power-on reset (POR). See reference chapters for other causes of resets. See Table 3-1 for reference for each cause of resets. See Table 3-2 for the availability of resets for each cause.

Table 3-1 Reference for Details of Causes of Resets

Cause	Reference
Reset input pin reset (pin reset)	This chapter
Power-On Reset (POR)	This chapter
Watchdog timer (WDT) overflow reset	Chapter 10 Watchdog Timer
Watchdog timer (WDT) invalid clear reset	Chapter 10 Watchdog Timer
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

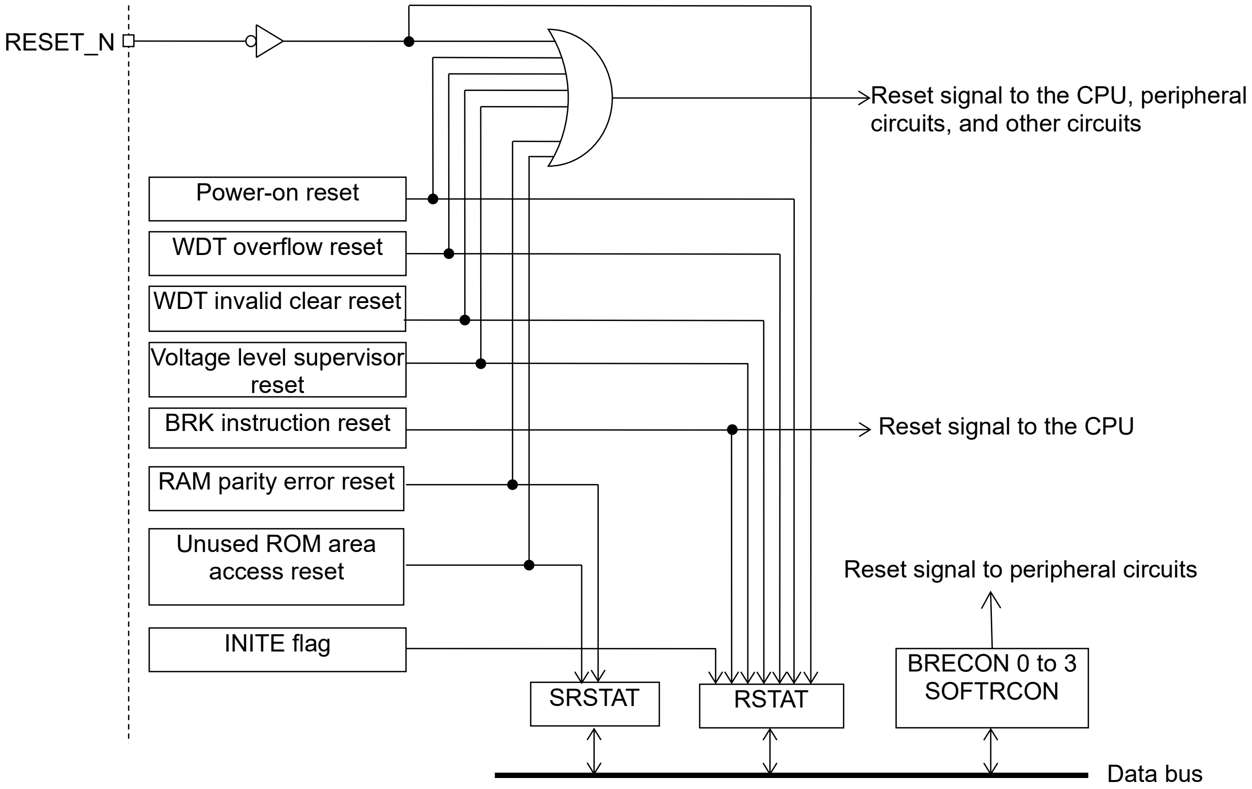
Each reset can uniquely be managed depending on its cause as this function contains following features to identify the cause in an early stage.

- 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)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	RSTAT															
Byte	RSTATH								RSTATL							
Bit	-	-	-	-	-	-	-	BRKR	INITE	RSTR	-	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

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

Bit No.	Bit symbol name	Description (target reset)
15 to 9	-	Reserved bits
8	BRKR	CPU reset by BRK instruction
7	INITE	A read-only bit to indicate that an abnormality occurred in starting LSI. If this bit is set to "1", restart the LSI by causing a reset to occur with the reset input pin reset, WDT invalid reset, WDT overflow reset or power-on. 0: LSI started-up normally 1: Abnormality occurred in start-up of LSI
6	RSTR	Reset input pin reset
5	-	Reserved bit
4	VLS0R	VLS reset
3	WDTWR	WDT invalid clear reset
2	WDTR	WDT overflow reset
1	-	Reserved bit
0	POR	Power-on reset or command reset of the on-chip debug function.

3.2.3 Safety Function Reset Status Register (SRSTAT)

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

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

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

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

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

3.3 Description of Operation

3.3.1 Operation of Reset Function

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

Table 3-2 Availability of Resets for Each Cause

Category	Cause	CPU	RAM	Crystal Oscillation Circuit *1	Voltage Level Supervisor	Other Peripheral Circuit	System Circuit *2
System reset	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	●	-	-	-	●	●
	Command reset in On-chip debug	●	-	-	-	●	●
CPU reset	BRK instruction reset	●	-	-	-	-	-
Peripheral reset	Block reset	-	-	-	-	●	-
	SOFTR reset	-	-	-	-	●	-

●: Reset available -: Reset unavailable

*1: Target SFRs are FLMOD, FBUSTAT register. See Chapter 6 for details.

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

[Note]

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

3.3.2 System Reset Mode

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

1. The fundamental hardware for the LSI operation, such as the power supply circuit and oscillation circuit, is initialized. In addition, functions chosen by the code option are configured. The INITE bit of the reset status register (RSTAT) is set to "1" if an abnormality occurs during the initialization and configuration.
   See the Chapter 30 "Code Option" for details of the code option.
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}).

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.

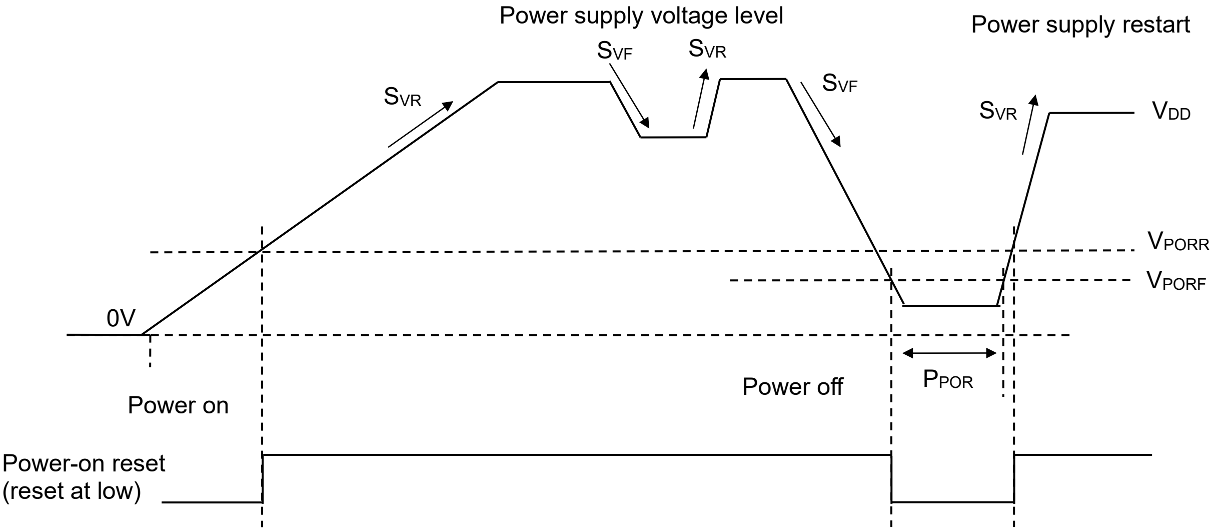


Figure 3-2 Power-on Reset Operation Waveforms

[Note]

- **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

ML62Q2500 group have five power management modes and block control function to save the current consumption. The block control function is to control clock supply and reset with respect to each peripheral.

Figure 4-1 shows the general scheme of the regulator.
The regulator generates a constant internal logic voltage (V_{DDL}) independent of the variation of V_{DD} (1.8 V to 5.5 V) using an amplifier for the low power consumption. The V_{DDL} generated by the regulator is supplied to peripheral circuits such as the internal logic circuit, flash memory, RAM, and oscillation circuit.
In order to stabilize the V_{DDL}, connect the VDDL pin to VSS via a capacitor (1 μF).

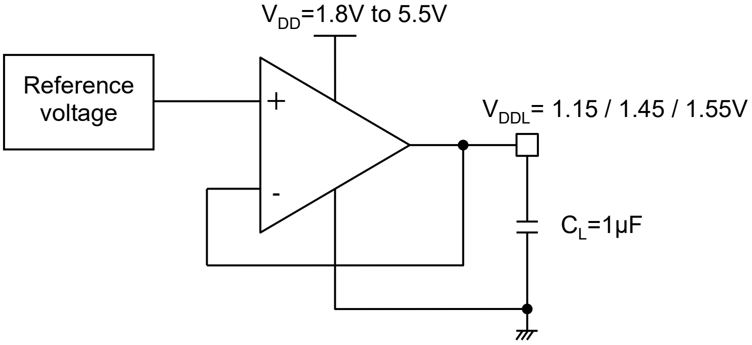


Figure 4-1 General Scheme of Regulator

4.1.1 Features

- 5 standby modes
  - HALT mode : The CPU stops executing instruction, peripheral circuits continue working.
  - HALT-H mode : The CPU stops executing instruction, high-speed clock oscillation stop and peripheral circuits continue working with low-speed clock only. A releasing time from standby mode is min. 60us.
  - HALT-D mode : The CPU stops executing instruction, high-speed clock oscillation stop and peripheral circuits continue working with low-speed clock. The internal logic voltage (V_{DDL}) goes down to reduce the power consumption (RAM data is retained).
  - STOP mode : The CPU stops executing instruction and all internal clocks stop.
  - STOP-D mode : The CPU stops executing instruction and all internal clocks stop. The internal logic voltage (V_{DDL}) goes down to reduce the power consumption (RAM data is retained).
- Stop code acceptor qualifies for entering STOP/STOP-D mode
- Data of RAM and SFR are retained even in all standby modes
- Clock supply is control-able peripheral by peripheral to reduce the current consumption, by block clock control registers
- Reset is control-able peripheral by peripheral by block reset control registers
- Automatic controlling internal voltage by operating mode and code option.

Mode	V_{DDL}
STOP mode	1.55V / 1.45V
HALT mode	1.55V / 1.45V
HALT-H mode	1.55V / 1.45V
Program run mode	1.55V / 1.45V
HALT-D/STOP-D mode (content of RAM and SFR can be retained)	1.15V

4.1.2 Configuration

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

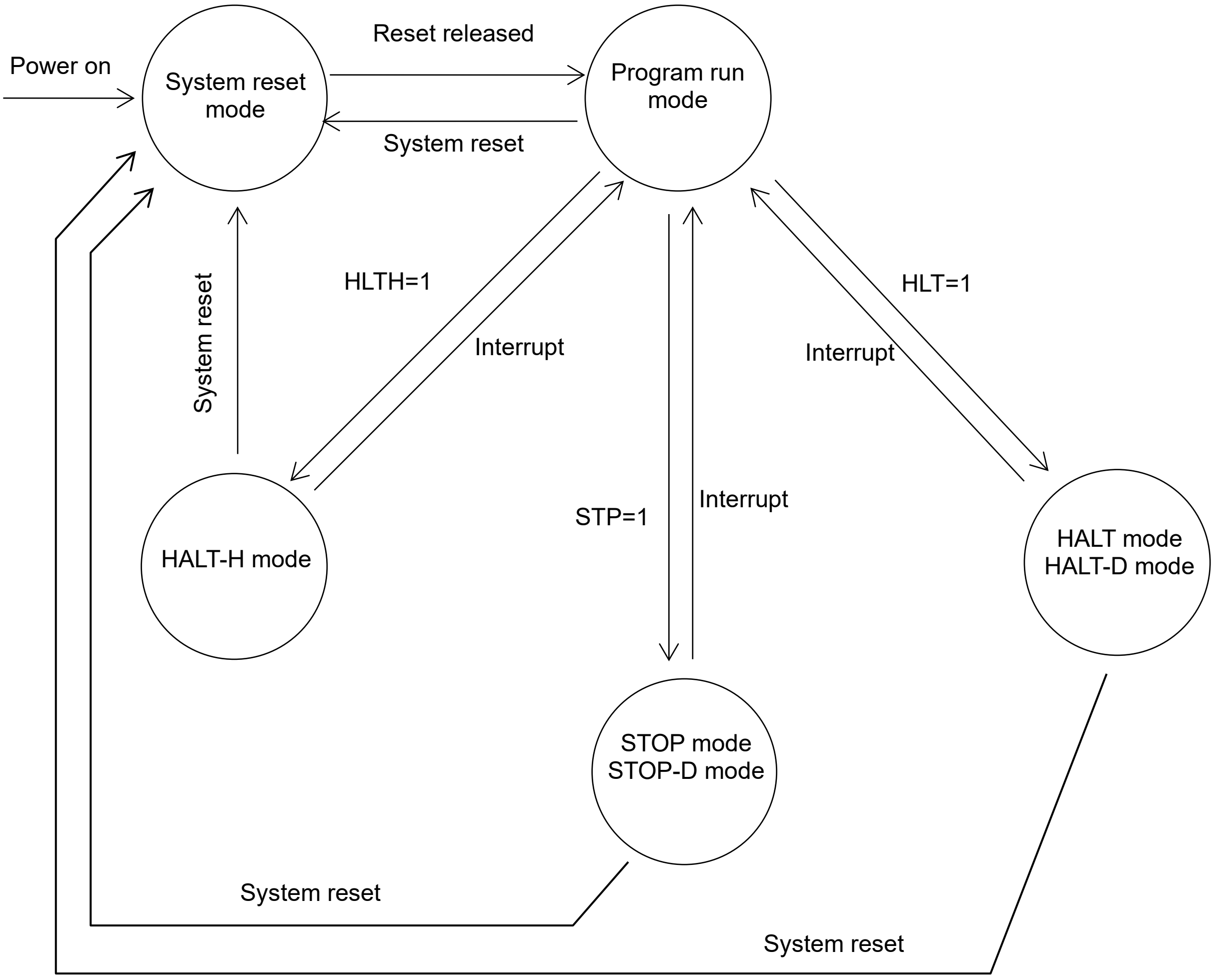


Figure 4-2 Operating State Transition Diagram

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

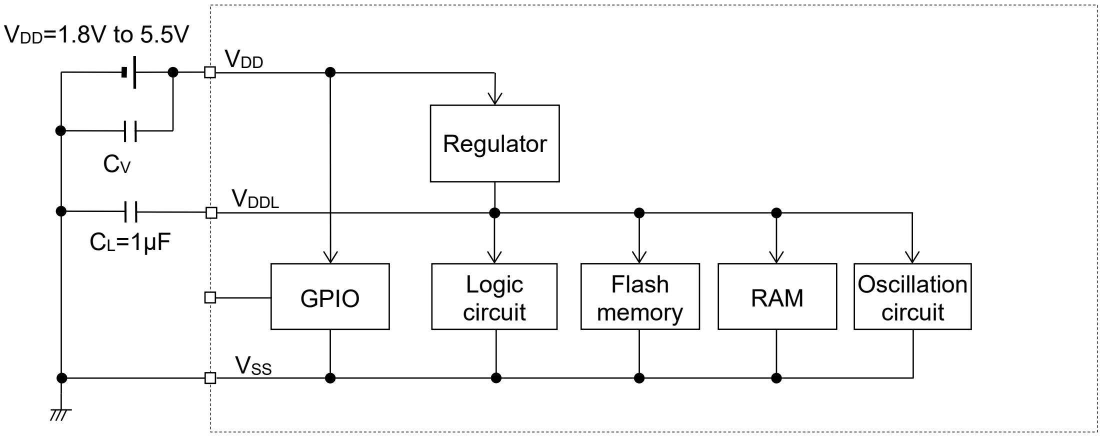


Figure 4-3 Internal Power Supply Configuration

4.1.3 List of Pins

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

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

[Note]

- **In order to improve the noise resistance, place the inter-power supply bypass capacitor (C_V) and the internal logic voltage (V_{DDL}) capacitor (C_L : 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 (VDDL pin output) is unavailable to use for an external device voltage.**

4.2 Description of Registers

4.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF018	Stop code acceptor	STPACP	-	W	8	0x00
0xF019	Reserved	-	-	-	-	-
0xF01A	Standby control register	SBYCONL	SBYCON	W	8/16	0x00
0xF01B		SBYCONH		R/W	8	0x00
0xF01C	Standby prohibition flag register	SBYEFLG	-	R	8	0x00
0xF01D	Reserved	-	-	-	-	-
0xF05C	Software reset acceptor	SOFTRACP	-	W	8	0x00
0xF05D	Reserved	-	-	-	-	-
0xF05E	Software reset control register	SOFTRCON	-	R/W	8	0x00
0xF05F	Reserved	-	-	-	-	-
0xF070	Block clock control register 0	BCKCON0L	BCKCON0	R/W	8/16	0x1F
0xF071		BCKCON0H		R/W	8	0x01
0xF072	Block clock control register 1	BCKCON1L	BCKCON1	R/W	8/16	0x03
0xF073		BCKCON1H		R/W	8	0x11
0xF074	Block clock control register 2	BCKCON2L	BCKCON2	R/W	8/16	0x73
0xF075		BCKCON2H		R/W	8	0x08
0xF076	Block clock control register 3	BCKCON3L	BCKCON3	R/W	8/16	0x01
0xF077		BCKCON3H		R/W	8	0x00
0xF078	Block reset control register 0	BRECON0L	BRECON0	R/W	8/16	0x1F
0xF079		BRECON0H		R/W	8	0x01
0xF07A	Block reset control register 1	BRECON1L	BRECON1	R/W	8/16	0x03
0xF07B		BRECON1H		R/W	8	0x11
0xF07C	Block reset control register 2	BRECON2L	BRECON2	R/W	8/16	0x73
0xF07D		BRECON2H		R/W	8	0x08
0xF07E	Block reset control register 3	BRECON3L	BRECON3	R/W	8/16	0x01
0xF07F		BRECON3H		R/W	8	0x00

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

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

Control register / bit				Available/ Unavailable
Word symbol	Bit symbol	Word symbol	Bit symbol	ML62Q2500 group
BCKCON0	DCKTM0	BRECON0	RSETM0	●
	DCKTM1		RSETM1	●
	DCKTM2		RSETM2	●
	DCKTM3		RSETM3	●
	DCKTM4		RSETM4	●
	DCKTMX		RSETMX	●
BCKCON1	DCKFTM0	BRECON1	RSEFTM0	●
	DCKFTM1		RSEFTM1	●
	DCKI2CM0		RSEI2CM0	●
	DCKI2CU0		RSEI2CU0	●
BCKCON2	DCKSIOF0	BRECON2	RSESIOF0	●
	DCKSIO0		RSESIO0	●
	DCKUA0		RSEUA0	●
	DCKUA1		RSEUA1	●
	DCKUA2		RSEUA2	●
	DCKCRC		RSECRC	●
	DCKACC		RSEACC	●
BCKCON3	DCKSAD	BRECON3	RSESAD	●

4.2.2 Stop Code Acceptor (STPACP)

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

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

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

How to enter the STOP/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/STOP-D mode only once.
2	Set STP bit of SBYCON register to"1".	STP=1 : Enter the STOP/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/STOP-D mode.**

4.2.3 Standby Control Register (SBYCON)

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

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

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

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

Bit No.	Bit symbol name	Description
15 to 10	-	Reserved bits
9	DPM	DPM is a bit used to select deep power-down. Set “1” to this bit if using STOP-D or HALT-D mode. This bit does not influence HLTH bit function. It is available setting to DPM bit and HLT/STP bit at once. 0 : Disabled (Initial value) 1 : Enabled
8 to 3	-	Reserved bits
2	HLTH	HLTH is a bit to stop forcibly the high-speed oscillation and change the operating state into the HALT-H mode. Its wake-up time is shorter than HALT mode. An entering this mode is available if SYSCLK is high speed clock. When using HALT-H mode, set FHRDWN bit in the high-speed clock wake up time setting register too. See "6.3.2.3 HALT-H mode" for wake-up time from the HALT-H mode.
1	STP	STP is a bit to change the operating state into the STOP/STOP-D mode. When "1" is written in the STP bit after entering the STOP/STOP-D mode is allowed by using STPACP, the operating state enters the STOP/STOP-D mode.
0	HLT	HLT is a bit to change the operating state into the HALT mode.

[Note]

- **The operating state does not enter the standby mode under some conditions. See “4.3.2.6 Note of entering to the standby mode” for detail conditions.**
- **When an interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated on the condition of MIE flag of the program status word (PSW) is "0", it cancels the standby mode only and the CPU does not go to the interrupt routine. For more details about MIE flag, see "nX-U16/100 Core Instruction Manual".**
- **Insert two NOP instructions in the next to the instruction of that sets HLT, STP, HLTH and STPD bit to "1". The operation without the two NOP instructions is not guaranteed.**

4.2.4 Standby Prohibition Flag Register (SBYEFLG)

This is a read-only SFR to indicate availability of entering to standby mode.
See “4.3.2.6 Note of entering to the standby mode” for condition that each bit becomes to “1”

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

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

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

Bit No.	Bit symbol name	Description
7 to 5	-	Reserved bits
4	FSTPD	STOP-D mode
3	FHLTD	HALT-D mode
2	FHLTH	HALT-H mode
1	FSTP	STOP mode
0	FHLT	HALT mode

4.2.5 Software Reset Acceptor (SOFTRACP)

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

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

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

How to reset collectively the peripheral circuits:

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

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

4.2.6 Software Reset Control Register (SOFTRCON)

This is a 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 :	8 bit
Initial value :	0x00

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

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	SOFTR	This 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.**

4.2.7 Block Clock Control Register 0 (BCKCON0)

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

Address :	0xF070 (BCKCON0L/BCKCON0), 0xF071 (BCKCON0H)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x011F

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BCKCON0															
Byte	BCKCON0H								BCKCON0L							
Bit	-	-	-	-	-	-	-	DCKT MX	-	-	-	DCKT M4	DCKT M3	DCKT M2	DCKT M1	DCKT M0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1

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

Bit No.	Bit symbol name	Description (target peripheral)
15 to 9	-	Reserved bits
8	DCKTMX	16-bit timer X
7	-	Reserved bit
6	-	Reserved bit
5	-	Reserved bit
4	DCKTM4	16-bit timer 4
3	DCKTM3	16-bit timer 3
2	DCKTM2	16-bit timer 2
1	DCKTM1	16-bit timer 1
0	DCKTM0	16-bit timer 0

4.2.8 Block Clock Control Register 1 (BCKCON1)

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

Address :	0xF072 (BCKCON1L/BCKCON1), 0xF073 (BCKCON1H)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x1103

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

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

Bit No.	Bit symbol name	Description (target peripheral)
15 to 13	-	Reserved bits
12	DCKI2CU0	I^{2}C Bus Unit 0
11,10	-	Reserved bits
9	-	Reserved bit
8	DCKI2CM0	I^{2}C Bus Master 0
7	-	Reserved bit
6	-	Reserved bit
5	-	Reserved bit
4	-	Reserved bit
3	-	Reserved bit
2	-	Reserved bit
1	DCKFTM1	Functional Timer 1
0	DCKFTM0	Functional Timer 0

4.2.9 Block Clock Control Register 2 (BCKCON2)

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

Address :	0xF074 (BCKCON2L/BCKCON2), 0xF075 (BCKCON2H)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0873

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BCKCON2															
Byte	BCKCON2H								BCKCON2L							
Bit	-	-	DCKACC	-	DCKCRC	-	-	-	-	DCKUA2	DCKUA1	DCKUA0	-	-	DCKSIO0	DCKSIOF0
R/W	R	R	R/W	R	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W
Initial value	0	0	0	0	1	0	0	0	0	1	1	1	0	0	1	1

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

Bit No.	Bit symbol name	Description (target peripheral)
15	-	Reserved bit
14	-	Reserved bit
13	DCKACC	Multiplier/Divider An initial value of this bit is ”0”.
12	-	Reserved bit
11	DCKCRC	CRC Calculator
10 to 8	-	Reserved bits
7	-	Reserved bit
6	DCKUA2	UART 2
5	DCKUA1	UART 1
4	DCKUA0	UART 0
3	-	Reserved bit
2	-	Reserved bit
1	DCKSIO0	SSIO 0
0	DCKSIOF0	SSIOF0

[Note]

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

4.2.10 Block Clock Control Register 3 (BCKCON3)

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

Address :	0xF076 (BCKCON3L/BCKCON3), 0xF077 (BCKCON3H)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0001

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BCKCON3															
Byte	BCKCON3H								BCKCON3L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DCKSAD
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	1

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

Bit No.	Bit symbol name	Description (target peripheral)
15 to 8	-	Reserved bits
7	-	Reserved bit
6	-	Reserved bit
5	-	Reserved bit
4	-	Reserved bit
3	-	Reserved bit
2	-	Reserved bit
1	-	Reserved bit
0	DCKSAD	SA-ADC

4.2.11 Block Reset Control Register 0 (BRECON0)

This is a SFR to control resetting the peripheral circuits.

Address :	0xF078(BRECON0L/BRECON0), 0xF079(BRECON0H)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x011F

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BRECON0															
Byte	BRECON0H								BRECON0L							
Bit	-	-	-	-	-	-	-	RSET MX	-	-	-	RSET M4	RSET M3	RSET M2	RSET M1	RSET M0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1

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

Bit No.	Bit symbol name	Description (target peripheral)
15 to 9	-	Reserved bits
8	RSETMX	16-bit timer X
7	-	Reserved bit
6	-	Reserved bit
5	-	Reserved bit
4	RSETM4	16-bit timer 4
3	RSETM3	16-bit timer 3
2	RSETM2	16-bit timer 2
1	RSETM1	16-bit timer 1
0	RSETM0	16-bit timer 0

4.2.12 Block Reset Control Register 1 (BRECON1)

This is a SFR to control resetting the peripheral circuits.

Address :	0xF07A (BRECON1L/BRECON1), 0xF07B (BRECON1H)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x1103

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

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

Bit No.	Bit symbol name	Description (target peripheral)
15 to 13	-	Reserved bits
12	RSEI2CU0	I^{2}C Bus Unit 0
11,10	-	Reserved bits
9	-	Reserved bit
8	RSEI2CM0	I^{2}C Bus Master 0
7	-	Reserved bit
6	-	Reserved bit
5	-	Reserved bit
4	-	Reserved bit
3	-	Reserved bit
2	-	Reserved bit
1	RSEFTM1	Functional Timer 1
0	RSEFTM0	Functional Timer 0

4.2.13 Block Reset Control Register 2 (BRECON2)

This is a SFR to control resetting the peripheral circuits.

Address : 0xF07C(BRECON2L/BRECON2), 0xF07D(BRECON2H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0873

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BRECON2															
Byte	BRECON2H								BRECON2L							
Bit	-	-	RSEACC	-	RSECRC	-	-	-	-	RSEUA2	RSEUA1	RSEUA0	-	-	RSESIO0	RSESIOF0
R/W	R	R	R/W	R	R/W	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W
Initial value	0	0	0	0	1	0	0	0	0	1	1	1	0	0	1	1

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

Bit No.	Bit symbol name	Description (target peripheral)
15	-	Reserved bit
14	-	Reserved bit
13	RSEACC	Multiplier/Divider An initial value of this bit is ”0”.
12	-	Reserved bit
11	RSECRC	CRC Calculator
10 to 8	-	Reserved bits
7	-	Reserved bit
6	RSEUA2	UART 2
5	RSEUA1	UART 1
4	RSEUA0	UART 0
3	-	Reserved bit
2	-	Reserved bit
1	RSESIO0	SSIO 0
0	RSESIOF0	SSIOF0

[Note]

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

4.2.14 Block Reset Control Register 3 (BRECON3)

This is a SFR to control resetting the peripheral circuits.

Address : 0xF07E(BRECON3L/BRECON3), 0xF07F(BRECON3H)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0001

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	BRECON3															
Byte	BRECON3H								BRECON3L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RSESAD
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	1

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

Bit No.	Bit symbol name	Description (target peripheral)
15 to 8	-	Reserved bits
7	-	Reserved bit
6	-	Reserved bit
5	-	Reserved bit
4	-	Reserved bit
3	-	Reserved bit
2	-	Reserved bit
1	-	Reserved bit
0	RSESAD	SA-ADC

4.3 Description of Operation

4.3.1 Program Run Mode

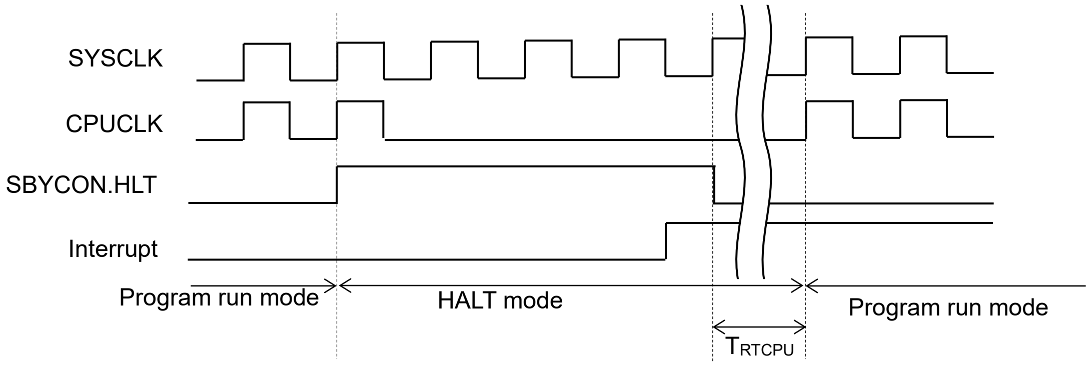
The program run mode is the state where the CPU executes instructions sequentially.
When a reset is released after the reset is generated, the operating state is transferred from the system reset mode to the program run mode.
In addition, if an interrupt request is generated during a standby mode, the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

4.3.2 Standby Mode

4.3.2.1 HALT Mode

The HALT mode is the state where the CPU stops and only the peripheral circuits remain in operation with previous clock condition (LSCLK0 or HSCLK) for the system clock (SYSCLK) chosen before entering the HALT mode. See "4.3.2.8 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 with DRM bit = “0”, the operating state enters the HALT mode. It is avilable to set DPM bit and HLT bit at once.
When a WDT interrupt or an interrupt enabled in registers IE0 to IE7 occurs, the HALT mode is released at the rising edge of the next SYSCLK, then the mode shifts back to the program run mode.
Figure 4-4 shows operation waveforms in the HALT mode.

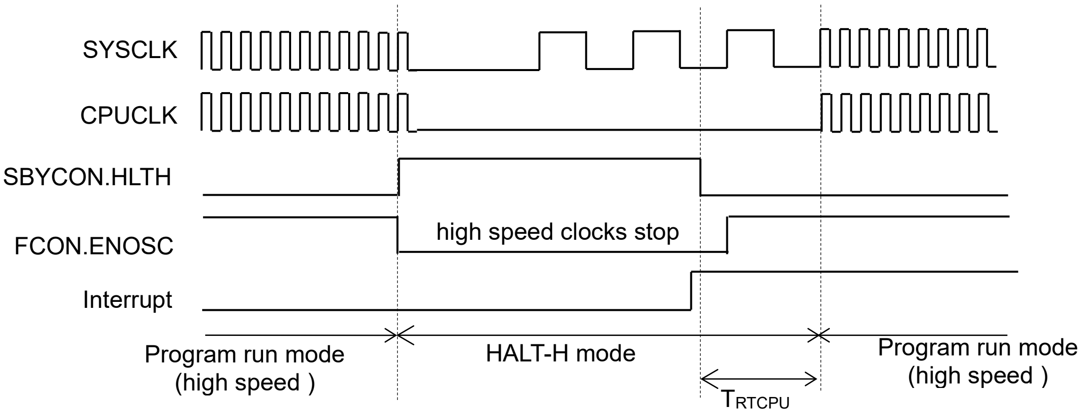


SBYCON : Standby control register
T_{RTCPU} : CPU clock restoring time (see Table 4-6)

Figure 4-4 Operation Waveforms in HALT Mode

4.3.2.2 HALT-H Mode

In the HALT-H mode, high speed clocks (HSCLK/HSOCLK/HCKO) is forcibly stopped, the CPU stops, and only peripheral circuits remain in operation with SYSCLK in the LSCLK0 state. Note that the peripheral circuits in operation with high speed clocks stop operating in the HALT-H mode. It can be used as a short standby for intermittent operation. See "4.3.2.8 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 SYSCLK, HSCLK is forcibly enabled, and the mode shifts back to the program run mode with the SYSCLK in the HSCLK state.
If the low-speed clock (LSCLK) is selected (SELSCLK="0") before entering the HALT-H mode, an entrying HALT-H is ignored.
Figure 4-5 shows operation waveforms in the HALT-H mode.

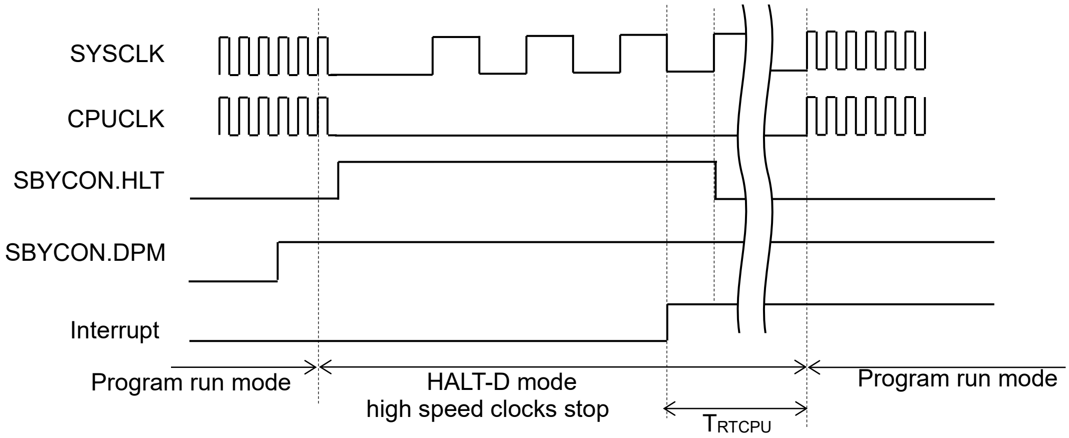


SBYCON : Standby control register
FCON : Frequency control register
T_{RTCPU} : CPU clock restoring time (see Table 4-6)

Figure 4-5 Operation Waveforms in HALT-H Mode

4.3.2.3 HALT-D Mode

In the HALT-D mode, high speed clocks (HSCLK/HSOCLK/HCKO) is forcibly stopped, the CPU stops, and only some peripheral circuits remain in operation with SYSCLK in the LSCLK0 state. Note that the peripheral circuits in operation with high speed clocks stop operating in the HALT-D mode. It can be used as a long standby. See "4.3.2.8 Operation of Each Function in Standby Mode" for operation of each function in the HALT-D mode.
When "1" is written in the HLT bit of the SBYCON register with DRM bit = “1”, the operating state enters the HALT-D mode. It is avilable to set DPM bit and HLT bit at once.
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 SYSCLK, HSCLK is forcibly enabled, and the mode shifts back to the program run mode with the SYSCLK chosen before entering the HALT-D mode.
Figure 4-6 shows operation waveforms in the HALT-D mode.

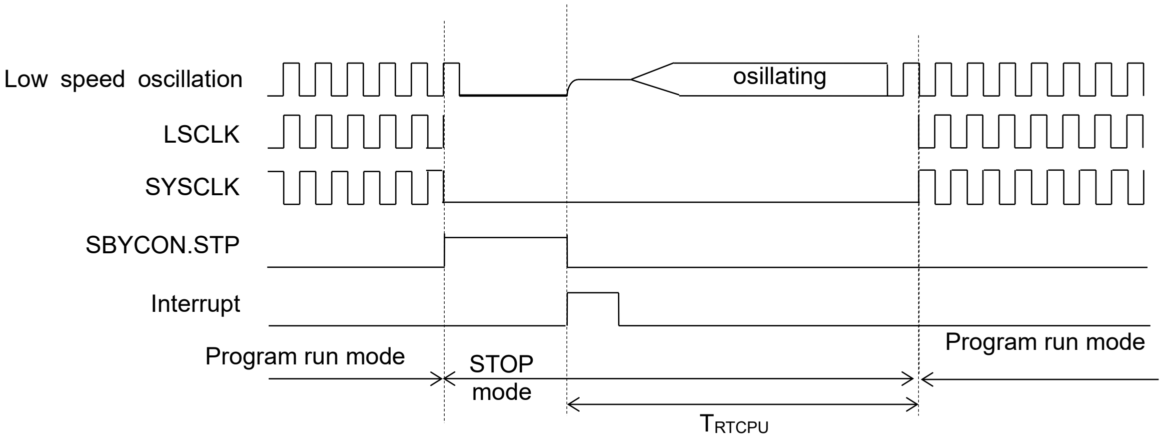


SBYCON : Standby control register
T_{RTCPU} : CPU clock restoring time (see Table 4-6)

Figure 4-6 Operation Waveforms in HALT-D Mode

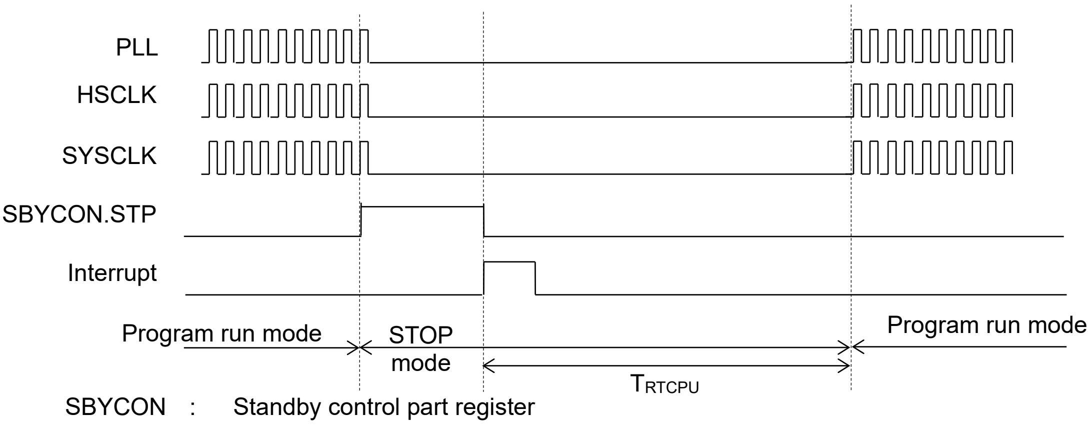
4.3.2.4 STOP Mode

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 "4.3.2.8 Operation of Each Function in Standby Mode" for operation of each function in the STOP mode.
To enter the STOP mode, write "0x5n" and "0xAn" (n = arbitrary) in this order to the STPACP register to enable the transition to the STOP/STOP-D mode, then write "1" to the STP bit of the SBYCON register with DRM bit = “0”. It is avilable to set DPM bit and STP bit at once.
The STOP mode is released by the external interrupts, voltage level supervisor (VLS) or interrupt requests from the I^{2}C bus unit (slave). The operating state returns to the program run mode with the SYSCLK chosen before entering the STOP mode.
Figure 4-7 shows STOP mode operation waveforms.



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

Figure 4-7-1 STOP Mode Operation Waveforms of Low-Speed Oscillation Circuit



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

Figure 4-7-2 STOP Mode Operation Waveforms of PLL Oscillation Circuit

4.3.2.5 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}). See "4.3.2.8 Operation of Each Function in Standby Mode" for operation of each function in the STOP-D mode.
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 STP bit of the SBYCON register with DRM bit = “0”. It is avilable to set DPM bit and STP bit at once.
The STOP-D mode is released by the external interrupts, voltage level supervisor (VLS) or interrupt requests from the I^{2}C bus unit (slave). The operating state returns to the program run mode with the SYSCLK chosen before entering the STOP mode.
The I^{2}C bus unit (slave) operation differs from STOP mode. See Chapter 13 “I^{2}C bus”.
Figure 4-7 shows STOP mode operation waveforms.

4.3.2.6 Note of entering to the standby mode

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

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

Condition	FSTPD	FHLTD	FHLTH	FSTP	FHLT
When setting some bits of SBYCONL register at the same time.	0	0	0	0	0
When System clock is selected LSCLK0.	0	0	1	0	0
When occurring the interrupt request to CPU. Its status is that both interrupt enable register and interrupt request register are asserted.	1	1	1	1	1
When accepter is disabled by SBYACP.	1	0	0	1	0
When A/D conversion of SA-ADC in progress.	1	0	0	1	0
When operating single mode of VLS.	1	0	0	1	0
When waiting for stability time of supervisor mode of VLS.	1	0	0	1	0
When erasing/programming for data flash memory.	1	1	1	1	0

If the high-speed clock stops when some peripherals are operating with high-speed clock (SA-ADC conversion or PWM output by Functional timer), there is a possibility of unintending current flow, depending on the timing of the stop.
This high-speed clock stopping includes HALT-H entry.

4.3.2.7 Note on Return Operation from Standby Mode

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

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

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

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

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

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

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

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

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

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

[Note]

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

4.3.2.8 Operation of Each Function in Standby Mode

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

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

Function blocks	HALT	HALT-H	HALT-D	STOP	STOP-D
Low speed oscillation (Internal RC oscillation)	●	●	●	-	-
Low speed oscillation (crystal or external input)	●	●	●	-	-
High speed oscillation (PLL)	●	-	-	-	-
CPU	-	-	-	-	-
RAM	Retain	Retain	Retain	Retain	Retain
WDT; Watchdog timer	●	●	●	-	-
External interrupt	●	●*2	●*2	●*1	●*1
Low-speed time base counter	●	●	●	-	-
16-bit timer	●	●	●	-*5	-*5
Functional timer	●	●	-*5	-*5	-*5
UART	●	●	-	-	-
SSIO (Master)	●	●*3	-	-	-
SSIO (Slave)	●	●	-*5	-*5	-*5
SSIO with FIFO (Master)	●	●*3	-	-	-
SSIO with FIFO (Slave)	●	●*3	-	-	-
I^{2}C bus unit (Master) / I^{2}C bus master	●	●*3	-	-	-
I^{2}C bus unit (Slave)	●	●	●*4	●*4	●*4
SA-ADC; Successive approximation type A/D converter	●	●	-	-	-
VLS	●	●	●	●*1	●*1
BGO operation (erasing/programming for data flash memory)	●	-	-	-	-
CRC calculator	●	●	-	-	-
Multiplier/Divider	-	-	-	-	-

*1 : If a sampling function is selected, it is forcibly disabled.
*2 : If a sampling function with high speed clock is selected, it is forcibly disabled.
*3 : System clock becomes low speed, so communication speed is influenced.
*4 : It is available to wake up by coincidence of slave address. A system clock supply is needed for communication after wake up.
*5 : Internal clocks is stop. If external clock is selected, the peripheral circuit operates. However its operation is not supported.

4.3.2.9 Wake-up Time from Standby Mode

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

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

Function	Condition	CPU clock restoring time [T_{RTCPU}]	Low-speed clock restoring time (Low-speed RC oscillation) [T_{RTLS}]	High-speed clock restoring time (PLL oscillation) [T_{RTPLL}]	
				FHWUPT=0x01	FHWUPT=0x00
HALT mode	Low-speed CPU clock High-speed clock OFF No CRC calculation	Approx.90µs	Operation continued	Stopped	
	Low-speed CPU clock High-speed clock ON or with CRC calculation	Approx.60µs		Operation continued	
	High-speed CPU clock	-			
HALT-H mode	No CRC calculation	T_{RTPLL}+15us	Operation continued	Approx.45µs	Max.2ms
	With CRC calculation	T_{RTPLL}+15us		Approx.60µs	Max.2ms
HALT-D mode	Low-speed CPU clock	Approx.300µs	Operation continued	Approx.350µs	Max.2.5ms
	High-speed CPU clock	T_{RTPLL}			
STOP/ STOP-D mode	Low-speed CPU clock	T_{RTLS}	Approx.3ms	Approx.3ms	Max.4ms
	High-speed CPU clock	T_{RTPLL}			

[Note]

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

4.3.3 Block Control Function

ML62Q2500 group has the block clock control function, which stops clock supply for each peripheral circuit to reduce current consumption, and the block reset control function to reset each peripheral circuit.
When setting each bit of the BCKCONn registers (n=0 to 3) to "1", the clock supply to the corresponding peripheral circuits stops, and the current consumption is reduced.
When setting each bit of the BRECONn registers (n=0 to 3) to "1", the corresponding peripheral circuits are reset and those SFRs are set with initial values.
Table 4-7 shows the list of peripheral circuits controllable with the block control function and control registers.

Table 4-7 List of Peripheral Circuits and Control Registers

Peripheral circuit	Block clock control function		Block reset control function		Software reset function
	SFR word symbol	SFR bit symbol	SFR word symbol	SFR bit symbol	SFR bit symbol
16-bit timer 0	BCKCON0	DCKTM0	BRECON0	RSETM0	SOFTR*1
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 X		DCKTMX		RSETMX	
Functional timer 0	BCKCON1	DCKFTM0	BRECON1	RSEFTM0	
Functional timer 1		DCKFTM1		RSEFTM1	
I^{2}C bus master 0		DCKI2CM0		RSEI2CM0	
I^{2}C bus unit 0		DCKI2CU0		RSEI2CU0	
SSIOF 0	BCKCON2	DCKSIOF0	BRECON2	RSESIO0	
SSIO 0		DCKSIO0		RSESIOF0	
UART 0		DCKUA0		RSEUA0	
UART 1		DCKUA1		RSEUA1	
UART 2		DCKUA2		RSEUA2	
CRC calculator		DCKCRC		RSECRC	
Multiplier/Divider		DCKACC		RSEACC	
Successive approximation type A/D converter	BCKCON3	DCKSAD	BRECON3	RSESAD	

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

After the system reset is released, operation of each peripheral circuit becomes enabled.
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. However reafing the SFRs of UARTs in the clock stopping state retuns always “0x00”.

[Note]

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

4.3.4 Internal Power Supply Voltage

After power-on, V_{DDL} becomes approximately 1.55 V, then becomes approximately 1.45 V at program run mode.
The management circuit controls V_{DDL} and a current performance of regulator to reduce a power consumption. The code-option VLMD can fix the V_{DDL} to 1.55V except in the STOP-D/HALT-D mode.
Figure 4-8 shows the operation waveforms of the regulator. Table 4-8 shows V_{DDL} for operating mode.

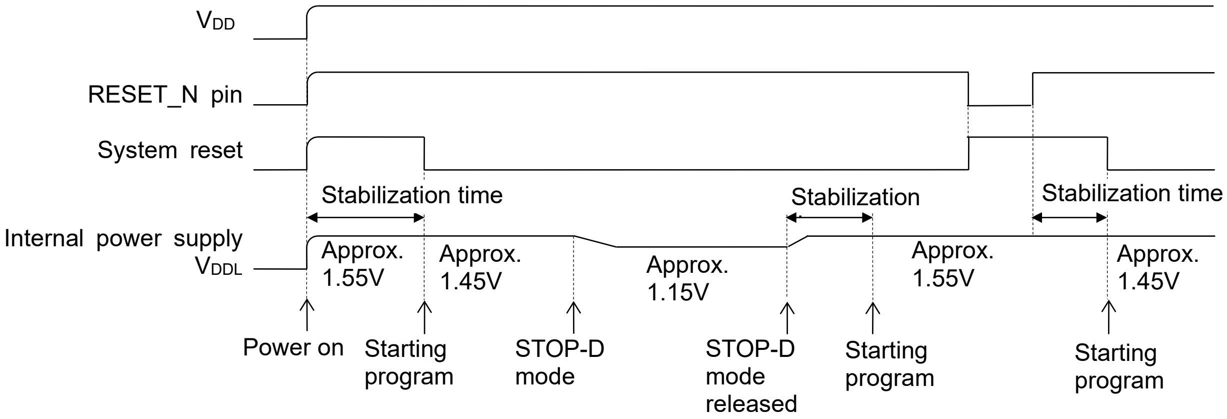


Figure 4-8 Regulator Operation Waveforms

Table 4-8 V_{DDL} for operating mode

Mode	V_{DDL}			
	VLMD=1			VLMD=0
	High speed clock off	High speed clock on with PLL1M mode	High speed clock on with PLL16/24M mode	
STOP mode	1.45V	1.45V	1.55V	1.55V
HALT mode	1.45V	1.45V	1.55V	1.55V
HALT-H mode	1.45V	1.45V	1.45V	1.55V
Program run mode	1.45V	1.45V	1.55V	1.55V
HALT-D/STOP-D mode	1.15V	1.15V	1.15V	1.15V
At erasing/programming FLASH (FLASHSLF.FSELF=1)	-	1.55V	1.55V	1.55V

Chapter 5 Interrupts

5. Interrupt

5.1 General Description

ML62Q2500 group has the non-maskable interrupt, maskable interrupts and the software interrupt (SWI).
For details of each interrupt, see the corresponding Chapters.
See Chapter 29 "Safety Function" for the MCU status interrupt.
See "Table 1-2 Main Function List" to confirm the presence/absence of function in each product.

5.1.1 Features

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

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

5.2.1 List of Registers

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

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

Table 5-1 List of Interrupt Source

● : presence , - : absence

Register assignment			Interrupt source	Interrupt source symbol	ML62Q2500 Group
IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)			
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]	-	-	-
IRQ2[2]	IE2[2]	ILC2[5:4]	MCU Status Interrupt	MCSINT	●
IRQ2[3]	IE2[3]	ILC2[7:6]	UART00 Interrupt	UA00INT	●
IRQ2[4]	IE2[4]	ILC2[9:8]	UART01 Interrupt	UA01INT	●
IRQ2[5]	IE2[5]	ILC2[11:10]	-	-	-
IRQ2[6]	IE2[6]	ILC2[13:12]	Successive Approximation type A-D Converter Interrupt (SA-ADC Interrupt)	SADINT	●
IRQ2[7]	IE2[7]	ILC2[15:14]	SSIOF0 Interrupt	SIOF0INT	●
IRQ3[0]	IE3[0]	ILC3[1:0]	SSIO0 Interrupt	SIO0INT	●
IRQ3[1]	IE3[1]	ILC3[3:2]	-	-	-
IRQ3[2]	IE3[2]	ILC3[5:4]	I^{2}C Bus Master 0 Interrupt	I2CM0INT	●
IRQ3[3]	IE3[3]	ILC3[7:6]	-	-	-
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	●
IRQ4[0]	IE4[0]	ILC4[1:0]	I^{2}C Bus Unit 0 Interrupt	I2CU0INT	●
IRQ4[1]	IE4[1]	ILC4[3:2]	UART10 Interrupt	UA10INT	●
IRQ4[2]	IE4[2]	ILC4[5:4]	UART11 Interrupt	UA11INT	●
IRQ4[3]	IE4[3]	ILC4[7:6]	-	-	-
IRQ4[4]	IE4[4]	ILC4[9:8]	-	-	-
IRQ4[5]	IE4[5]	ILC4[11:10]	-	-	-
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]	UART20 Interrupt	UA20INT	●

Register assignment			Interrupt source	Interrupt source symbol	ML62Q2500 Group
IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)			
IRQ5[1]	IE5[1]	ILC5[3:2]	UART21 Interrupt	UA21INT	●
IRQ5[2]	IE5[2]	ILC5[5:4]	-	-	-
IRQ5[3]	IE5[3]	ILC5[7:6]	-	-	-
IRQ5[4]	IE5[4]	ILC5[9:8]	-	-	-
IRQ5[5]	IE5[5]	ILC5[11:10]	-	-	-
IRQ5[6]	IE5[6]	ILC5[13:12]	16-bit Timer 4 Interrupt	TM4INT	●
IRQ5[7]	IE5[7]	ILC5[15:14]	-	-	-
IRQ6[0]	IE6[0]	ILC6[1:0]	-	-	-
IRQ6[1]	IE6[1]	ILC6[3:2]	-	-	-
IRQ6[2]	IE6[2]	ILC6[5:4]	-	-	-
IRQ6[3]	IE6[3]	ILC6[7:6]	-	-	-
IRQ6[4]	IE6[4]	ILC6[9:8]	-	-	-
IRQ6[5]	IE6[5]	ILC6[11:10]	-	-	-
IRQ6[6]	IE6[6]	ILC6[13:12]	-	-	-
IRQ6[7]	IE6[7]	ILC6[15:14]	-	-	-
IRQ7[0]	IE7[0]	ILC7[1:0]	-	-	-
IRQ7[1]	IE7[1]	ILC7[3:2]	16-bit Timer X Interrupt	TMXINT	●
IRQ7[2]	IE7[2]	ILC7[5:4]	Low-speed Time Base Counter 0 Interrupt	LTB0INT	●
IRQ7[3]	IE7[3]	ILC7[7:6]	Low-speed Time Base Counter 3 Interrupt	LTB3INT	●
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]	-	-	-
IRQ7[7]	IE7[7]	ILC7[15:14]	-	-	-

5.2.2 Interrupt Enable Register 01 (IE01)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IE01															
Byte	IE1								IE0							
Bit	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

Common description of each bits :
It is configured enable/disable a target interrupt

0: Disable a target interrupt (Initial value)
1: Enable a target interrupt

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

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

5.2.3 Interrupt Enable Register 23 (IE23)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IE23															
Byte	IE3								IE2							
Bit	ETM1	ETM0	EFTM1	EFTM0	-	EI2CM0	-	ESIO0	ESIOF0	ESAD	-	EUA01	EUA00	EMCS	-	ECBU
R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R	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

Common description of each bits :
It is configured enable/disable a target interrupt

0: Disable a target interrupt (Initial value)
1: Enable a target interrupt

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

Bit No.	Bit symbol name	Description (target interrupt)
15	ETM1	16-bit Timer 1 interrupt (TM1INT)
14	ETM0	16-bit Timer 0 interrupt (TM0INT)
13	EFTM1	Functional Timer 1 interrupt (FTM1INT)
12	EFTM0	Functional Timer 0 interrupt (FTM0INT)
11	-	Reserved bit
10	EI2CM0	I^{2}C Bus Master 0 interrupt (I2CM0INT)
9	-	Reserved bit
8	ESIO0	SSIO0 interrupt (SIO0INT)
7	ESIOF0	SSIOF0 interrupt (SIOF0INT)
6	ESAD	SA-ADC interrupt (SADINT)
5	-	Reserved bit
4	EUA01	UART01 interrupt (UA01INT)
3	EUA00	UART00 interrupt (UA00INT)
2	EMCS	MCU Status interrupt (MCSINT) See Chapter 29 "Safety Function" for more details.
1	-	Reserved bit
0	ECBU	Clock Backup interrupt (CBUINT) See Chapter 6 "Clock Generation Circuit" for more details.

5.2.4 Interrupt Enable Register 45 (IE45)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IE45															
Byte	IE5								IE4							
Bit	-	ETM4	-	-	-	-	EUA21	EUA20	ETM3	ETM2	-	-	-	EUA11	EUA10	EI2CU0
R/W	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

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

Bit No.	Bit symbol name	Description (target interrupt)
15	-	Reserved bit
14	ETM4	16-bit Timer 4 interrupt (TM4INT)
13	-	Reserved bit
12	-	Reserved bit
11	-	Reserved bit
10	-	Reserved bit
9	EUA21	UART21 interrupt (UA21INT)
8	EUA20	UART20 interrupt (UA20INT)
7	ETM3	16-bit Timer 3 interrupt (TM3INT)
6	ETM2	16-bit Timer 2 interrupt (TM2INT)
5	-	Reserved bit
4	-	Reserved bit
3	-	Reserved bit
2	EUA11	UART11 interrupt (UA11INT)
1	EUA10	UART10 interrupt (UA10INT)
0	EI2CU0	I^{2}C Bus Unit 0 interrupt (I2CU0INT)

5.2.5 Interrupt Enable Register 67 (IE67)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IE67															
Byte	IE7								IE6							
Bit	-	-	ELTBC2	ELTBC1	ELTBC3	ELTBC0	ETMX	-	-	-	-	-	-	-	-	-
R/W	R	R	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

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

Bit No.	Bit symbol name	Description (target interrupt)
15	-	Reserved bit
14	-	Reserved bit
13	ELTBC2	Low speed Time base counter 2 interrupt (LTB2INT)
12	ELTBC1	Low speed Time base counter 1 interrupt (LTB1INT)
11	ELTBC3	Low speed Time base counter 3 interrupt (LTB3INT)
10	ELTBC0	Low speed Time base counter 0 interrupt (LTB0INT)
9	ETMX	16-bit Timer X interrupt (TMXINT)
8	-	Reserved bit
7 to 0	-	Reserved bits

5.2.6 Interrupt Request Register 01 (IRQ01)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IRQ01															
Byte	IRQ1								IRQ0							
Bit	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

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

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

Bit No.	Bit symbol name	Description (target interrupt)
15	QPI7	external interrupt 7 (EXI7INT)
14	QPI6	external interrupt 6 (EXI6INT)
13	QPI5	external interrupt 5 (EXI5INT)
12	QPI4	external interrupt 4 (EXI4INT)
11	QPI3	external interrupt 3 (EXI3INT)
10	QPI2	external interrupt 2 (EXI2INT)
9	QPI1	external interrupt 1 (EXI1INT)
8	QPI0	external interrupt 0 (EXI0INT)
7	-	Reserved bit
6	QVLS0	VLS0 interrupt (VLS0INT)
5 to 1	-	Reserved bits
0	QWDT	external interrupt 7 (EXI7INT)

[Note]

- **There is a risk of clearing other request flags of This IRQ 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.

5.2.7 Interrupt Request Register 23 (IRQ23)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IRQ23															
Byte	IRQ3								IRQ2							
Bit	QTM1	QTM0	QFTM1	QFTM0	-	QI2CM0	-	QSIO0	QSIOF0	QSAD	-	QUA01	QUA00	QMCS	-	QCBU
R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R	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

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

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

Bit No.	Bit symbol name	Description (target interrupt)
15	QTM1	16-bit Timer 1 interrupt (TM1INT)
14	QTM0	16-bit Timer 0 interrupt (TM0INT)
13	QFTM1	Functional Timer 1 interrupt (FTM1INT)
12	QFTM0	Functional Timer 0 interrupt (FTM0INT)
11	-	Reserved bit
10	QI2CM0	I^{2}C Bus Master 0 interrupt (I2CM0INT)
9	-	Reserved bit
8	QSIO0	SSIO0 interrupt (SIO0INT)
7	QSIOF0	SSIOF0interrupt (SIOF0INT)
6	QSAD	SA-ADC interrupt (SADINT)
5	-	Reserved bit
4	QUA01	UART01 interrupt (UA01INT)
3	QUA00	UART00 interrupt (UA00INT)
2	QMCS	MCU Status interrupt (MCSINT) See Chapter 29 "Safety Function" for more details.
1	-	Reserved bit
0	QCBU	Clock Backup interrupt (CBUINT) See Chapter 6 "Clock Generation Circuit" for more details.

[Note]

- **There is a risk of clearing other request flags of This IRQ 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.

5.2.8 Interrupt Request Register 45 (IRQ45)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IRQ45															
Byte	IRQ5								IRQ4							
Bit	-	QTM4	-	-	-	-	QUA21	QUA20	QTM3	QTM2	-	-	-	QUA11	QUA10	QI2CU0
R/W	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

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

Bit No.	Bit symbol name	Description (target interrupt)
15	-	Reserved bit
14	QTM4	16-bit Timer 4 interrupt (TM4INT)
13	-	Reserved bit
12	-	Reserved bit
11	-	Reserved bit
10	-	Reserved bit
9	QUA21	UART21 interrupt (UA21INT)
8	QUA20	UART20 interrupt (UA20INT)
7	QTM3	16-bit Timer 3 interrupt (TM3INT)
6	QTM2	16-bit Timer 2 interrupt (TM2INT)
5	-	Reserved bit
4	-	Reserved bit
3	-	Reserved bit
2	QUA11	UART11 interrupt (UA11INT)
1	QUA10	UART10 interrupt (UA10INT)
0	QI2CU0	I^{2}C Bus Unit 0 interrupt (I2CU0INT)

[Note]

- **There is a risk of clearing other request flags of This IRQ 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.

5.2.9 Interrupt Request Register 67 (IRQ67)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	IRQ67															
Byte	IRQ7								IRQ6							
Bit	-	-	QLTB C2	QLTB C1	QLTB C3	QLTB C0	QTM X	-	-	-	-	-	-	-	-	-
R/W	R	R	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

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

Bit No.	Bit symbol name	Description (target interrupt)
15	-	Reserved bit
14	-	Reserved bit
13	QLTBC2	Low speed Time base counter 2 interrupt (LTB2INT)
12	QLTBC1	Low speed Time base counter 1 interrupt (LTB1INT)
11	QLTBC3	Low speed Time base counter 3 interrupt (LTB3INT)
10	QLTBC0	Low speed Time base counter 0 interrupt (LTB0INT)
9	QTMX	16-bit Timer X interrupt (TMXINT)
8 to 0	-	Reserved bits

[Note]

- **There is a risk of clearing other request flags of This IRQ 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.**

5.2.10 Interrupt Level Control Enable Register (ILEN)

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

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

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

Bit No.	Bit symbol name	Description
15 to 1	-	Reserved bits
0	ILE	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 ILC7) to "0x0000" and confirming the current interrupt request level register (CIL) is "0x00" when the interrupt is disabled (IE01 to IE67 registers are "0x00").**
- **Enable the interrupt level control function by setting the ILE bit to "1" when the interrupt is disabled(IE0 to IE7 registers are "0") or master interrupt enable flag(MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.**

5.2.11 Current Interrupt Level Management Register (CIL)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								CIL							
Bit	-	-	-	-	-	-	-	-	CILN	-	-	-	CILM3	CILM2	CILM1	CILM0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :

0: A target interrupt is not being processed (Initial value)
1: A target interrupt is being processed

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

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

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

5.2.12 Masking Interrupt Level Register (MCIL)

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

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

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

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

Common description of each bits :

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

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

5.2.13 Interrupt Level Control Register 0 (ILC0)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC0															
Byte	ILC01								ILC00							
Bit	-	-	ILVL S0H	ILVL S0L	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured a target interrupt level.

00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description (target interrupt level)
15,14	-	Reserved bits
13,12	ILVLS0H, ILVLS0L	VLS0 interrupt (VLS0INT)
11 to 0	-	Reserved bits

[Note]

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

5.2.14 Interrupt Level Control Register 1 (ILC1)

ILC1 is a 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:	8/16 bit
Initial value:	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC1															
Byte	ILC11								ILC10							
Bit	ILPI 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

Common description of each bits :
It is configured a target interrupt level.

00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description (target interrupt level)
15,14	ILPI7H, ILPI7L	external interrupt 7 (EXI7INT)
13,12	ILPI6H, ILPI6L	external interrupt 6 (EXI6INT)
11,10	ILPI5H, ILPI5L	external interrupt 5 (EXI5INT)
9,8	ILPI4H, ILPI4L	external interrupt 4 (EXI4INT)
7,6	ILPI3H, ILPI3L	external interrupt 3 (EXI3INT)
5,4	ILPI2H, ILPI2L	external interrupt 2 (EXI2INT)
3,2	ILPI1H, ILPI1L	external interrupt 1 (EXI1INT)
1,0	ILPI0H, ILPI0L	external interrupt 0 (EXI0INT)

[Note]

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

5.2.15 Interrupt Level Control Register 2 (ILC2)

ILC2 is a 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 IEL bit of the interrupt level control enable register (ILEN) to "1".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC2															
Byte	ILC21								ILC20							
Bit	ILSIO F0H	ILSIO F0L	ILSA DH	ILSA DL	-	-	ILUA0 1H	ILUA0 1L	ILUA0 0H	ILUA0 0L	ILMC SH	ILMC SL	-	-	ILCB UH	ILCB UL
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured a target interrupt level.

00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description (target interrupt level)
15,14	ILSIOF0H, ILSIOF0L	SSIOF0interrupt (SIOF0INT)
13,12	ILSADH, ILSADL	SA-ADC interrupt (SADINT)
11,10	-	Reserved bit
9,8	ILUA01H, ILUA01L	UART01 interrupt (UA01INT)
7,6	ILUA00H, ILUA00L	UART00 interrupt (UA00INT)
5,4	ILMCSH, ILMCSL	MCU Status interrupt (MCSINT) See Chapter 29 "Safety Function" for more details.
3,2	-	Reserved bit
1,0	ILCBUH, ILCBUL	Clock Backup interrupt (CBUINT) See Chapter 6 "Clock Generation Circuit" for more details.

[Note]

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

5.2.16 Interrupt Level Control Register 3 (ILC3)

ILC3 is a 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:	8/16 bit
Initial value:	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC3															
Byte	ILC31								ILC30							
Bit	ILTM1H	ILTM1L	ILTM0H	ILTM0L	ILFTM1H	ILFTM1L	ILFTM0H	ILFTM0L	-	-	ILI2CM0H	ILI2CM0L	-	-	ILSIO0H	ILSIO0L
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	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

ビット 番号	ビットシンボル名	説明(該当する interrupt)
15,14	ILTM1H, ILTM1L	16-bit Timer 1 interrupt (TM1INT)
13,12	ILTM0H, ILTM0L	16-bit Timer 0 interrupt (TM0INT)
11,10	ILFTM1H, ILFTM1L	Functional timer 1 interrupt (FTM1INT)
9,8	ILFTM0H, ILFTM0L	Functional timer 0 interrupt (FTM0INT)
7,6	-	Reserved bits
5,4	ILI2CM0H, ILI2CM0L	I^{2}C bus master 0 interrupt (I2CM0INT)
3,2	-	Reserved bits
1,0	ILSIO0H, ILSIO0L	SSIO0 interrupt (SIO0INT)

[Note]

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

5.2.17 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 IEL bit of the interrupt level control enable register (ILEN) to "1".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC4															
Byte	ILC41								ILC40							
Bit	ILTM3H	ILTM3L	ILTM2H	ILTM2L	-	-	-	-	-	-	ILUA11H	ILUA11L	ILUA10H	ILUA10L	ILI2CU0H	ILI2CU0L
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit No.	Bit symbol name	Description (target interrupt level)
15,14	ILTM3H, ILTM3L	16-bit Timer 3 interrupt (TM3INT)
13,12	ILTM2H, ILTM2L	16-bit Timer 2 interrupt (TM2INT)
11,10	-	Reserved bit
9,8	-	Reserved bit
7,6	-	Reserved bit
5,4	ILUA11H, ILUA11L	UART11 interrupt (UA11INT)
3,2	ILUA10H, ILUA10L	UART10 interrupt (UA10INT)
1,0	ILI2CU0H, ILI2CU0L	I^{2}C Bus Unit 0 interrupt (I2CU0INT)

[Note]

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

5.2.18 Interrupt Level Control Register 5 (ILC5)

ILC5 is a 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 IEL bit of the interrupt level control enable register (ILEN) to "1".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC5															
Byte	ILC51								ILC50							
Bit	-	-	ILTM4H	ILTM4L	-	-	-	-	-	-	-	-	ILUA21H	ILUA21L	ILUA20H	ILUA20L
R/W	R	R	R/W	R/W	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

Common description of each bits :
It is configured a target interrupt level.

00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description (target interrupt level)
15~12	-	Reserved bit
13,12	ILTM4H, ILTM4L	16-bit Timer 4 interrupt (TM4INT)
11,10	-	Reserved bit
9,8	-	Reserved bit
7,6	-	Reserved bit
5,4	-	Reserved bit
3,2	ILUA21H, ILUA21L	UART21 interrupt (UA21INT)
1,0	ILUA20H, ILUA20L	UART20 interrupt (UA20INT)

[Note]

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

5.2.19 Interrupt Level Control Register 6 (ILC6)

ILC6 is a 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: 8/16 bit
Initial value: 0x0000

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

Bit No.	Bit symbol name	Description (target interrupt level)
15 to 0	-	Reserved bits

[Note]

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

5.2.20 Interrupt Level Control Register 7 (ILC7)

ILC7 is a 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 IEL bit of the interrupt level control enable register (ILEN) to "1".

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	ILC7															
Byte	ILC71								ILC70							
Bit	-	-	-	-	ILLTBC2H	ILLTBC2L	ILLTBC1H	ILLTBC1L	ILLTBC3H	ILLTBC3L	ILLTBC0H	ILLTBC0L	ILTMXH	ILTMXL	-	-
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured a target interrupt level.

00: level 1 ; a priority is lower. (Initial value)
01: level 2
10: level 3
11: level 4 ; a priority is higher.

Bit No.	Bit symbol name	Description (target interrupt level)
15,14	-	Reserved bit
13,12	-	Reserved bit
11,10	ILLTBC2H, ILLTBC2L	Low speed Time base counter 2 interrupt (LTB2INT)
9,8	ILLTBC1H, ILLTBC1L	Low speed Time base counter 1 interrupt (LTB1INT)
7,6	ILLTBC3H, ILLTBC3L	Low speed Time base counter 3 interrupt (LTB3INT)
5,4	ILLTBC0H, ILLTBC0L	Low speed Time base counter 0 interrupt (LTB0INT)
3,2	ILTMXH, ILTMXL	16-bit Timer X interrupt (TMXINT)
1,0	-	Reserved bit

[Note]

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

5.3 Description of Operation

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

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

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

Table 5-2 List of interrupt sources

Interrupt source number (priority)	Register assignment			Interrupt vector address	Mask	External /internal source	Interrupt source	Symbol
	IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)					
1(high)	IRQ0[0]	-	-	0x0008	Disabled	Internal	WDT interrupt	WDTINT
2	-	-	-	0x000A	DIsabled		-	-
3	IRQ0[6]	IE0[6]	ILC0[13:12]	0x000C	Enabled		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		-	-
15	IRQ2[2]	IE2[2]	ILC2[5:4]	0x0024	Enabled		MCU status interrupt*1	MCSINT
16	IRQ2[3]	IE2[3]	ILC2[7:6]	0x0026	Enabled		UART00 interrupt	UA00INT
17	IRQ2[4]	IE2[4]	ILC2[9:8]	0x0028	Enabled		UART01 interrupt	UA01INT
18	IRQ2[5]	IE2[5]	ILC2[11:10]	0x002A	Enabled		-	-
19	IRQ2[6]	IE2[6]	ILC2[13:12]	0x002C	Enabled		SA-ADC interrupt	SADINT
20	IRQ2[7]	IE2[7]	ILC2[15:14]	0x002E	Enabled		SSIOF0 interrupt	SIOF0INT
21	IRQ3[0]	IE3[0]	ILC3[1:0]	0x0030	Enabled	Internal	SSIO0 interrupt	SIO0INT
22	IRQ3[1]	IE3[1]	ILC3[3:2]	0x0032	Enabled		-	-
23	IRQ3[2]	IE3[2]	ILC3[5:4]	0x0034	Enabled		I^{2}C bus master 0 interrupt	I2CM0INT
24	IRQ3[3]	IE3[3]	ILC3[7:6]	0x0036	Enabled		-	-
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^{2}C bus unit interrupt	I2CU0INT
30	IRQ4[1]	IE4[1]	ILC4[3:2]	0x0042	Enabled		UART10 interrupt	UA10INT
31	IRQ4[2]	IE4[2]	ILC4[5:4]	0x0044	Enabled		UART11 interrupt	UA11INT
32	IRQ4[3]	IE4[3]	ILC4[7:6]	0x0046	Enabled		-	-
33	IRQ4[4]	IE4[4]	ILC4[9:8]	0x0048	Enabled		-	-
34	IRQ4[5]	IE4[5]	ILC4[11:10]	0x004A	Enabled		-	-
35	IRQ4[6]	IE4[6]	ILC4[13:12]	0x004C	Enabled		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	Internal	UART20 interrupt	UA20INT
38	IRQ5[1]	IE5[1]	ILC5[3:2]	0x0052	Enabled		UART21 interrupt	UA21INT
39	IRQ5[2]	IE5[2]	ILC5[5:4]	0x0054	Enabled		-	-
40	IRQ5[3]	IE5[3]	ILC5[7:6]	0x0056	Enabled		-	-
41	IRQ5[4]	IE5[4]	ILC5[9:8]	0x0058	Enabled		-	-
42	IRQ5[5]	IE5[5]	ILC5[11:10]	0x005A	Enabled		-	-

Interrupt source number (priority)	Register assignment			Interrupt vector address	Mask	External /internal source	Interrupt source	Symbol
	IRQ (interrupt request)	IE (interrupt enable)	ILC (interrupt level)					
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		-	-
45	IRQ6[0]	IE6[0]	ILC6[1:0]	0x0060	Enabled	Internal	-	-
46	IRQ6[1]	IE6[1]	ILC6[3:2]	0x0062	Enabled		-	-
47	IRQ6[2]	IE6[2]	ILC6[5:4]	0x0064	Enabled		-	-
48	IRQ6[3]	IE6[3]	ILC6[7:6]	0x0066	Enabled		-	-
49	IRQ6[4]	IE6[4]	ILC6[9:8]	0x0068	Enabled		-	-
50	IRQ6[5]	IE6[5]	ILC6[11:10]	0x006A	Enabled		-	-
51	IRQ6[6]	IE6[6]	ILC6[13:12]	0x006C	Enabled		-	-
52	IRQ6[7]	IE6[7]	ILC6[15:14]	0x006E	Enabled		-	-
53	IRQ7[0]	IE7[0]	ILC7[1:0]	0x0070	Enabled	Internal	-	-
54	IRQ7[1]	IE7[1]	ILC7[3:2]	0x0072	Enabled		16-bit Timer X interrupt	TMXINT
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		Low speed timer base counter 3 interrupt	LTB3INT
57	IRQ7[4]	IE7[4]	ILC7[9:8]	0x0078	Enabled		Low speed timer base counter 1 interrupt	LTB1INT
58	IRQ7[5]	IE7[5]	ILC7[11:10]	0x007A	Enabled		Low speed timer base counter 2 interrupt	LTB2INT
59	IRQ7[6]	IE7[6]	ILC7[13:12]	0x007C	Enabled		-	-
60(低)	IRQ7[7]	IE7[7]	ILC7[15:14]	0x007E	Enabled		-	-

*1 The MCU status interrupt occurs when the following request is asserted.

- RAM parity error
- Automatic CRC calculation completion
- Data flash erasing/programming completion

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

[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:

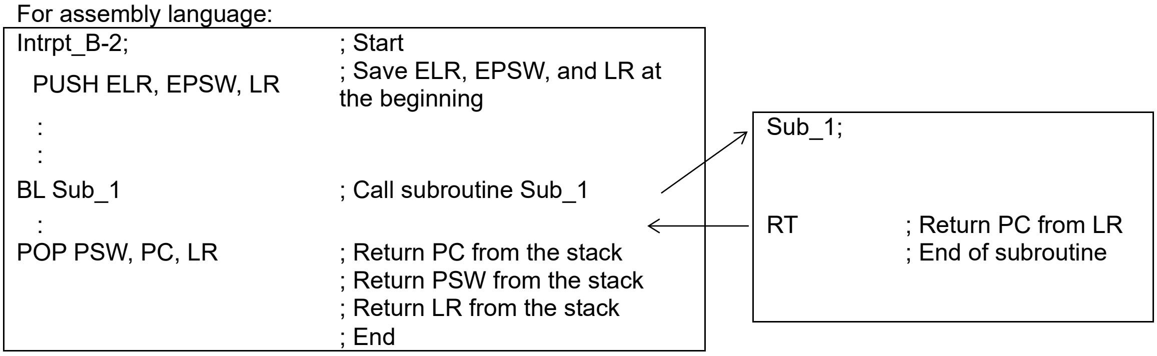
```
static void Intrpt_B_1(void);
#pragma interrupt Intrpt_B_1 0x08 2
static void Intrpt_B_1(void)
{
   :
}                         /* End of interrupt routine */
```

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:



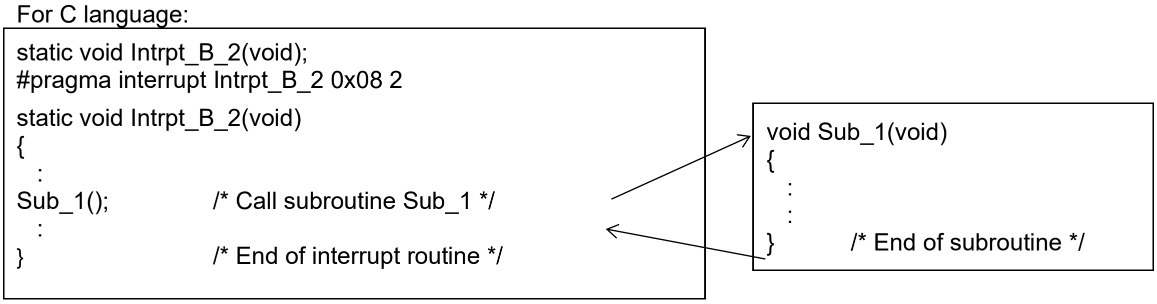
```
Intrpt_B-2;                 ; Start
  PUSH ELR, EPSW, LR        ; Save ELR, EPSW, and LR at
                            the beginning
   :
   :
BL Sub_1                    ; Call subroutine Sub_1
   :
POP PSW, PC, LR             ; Return PC from the stack
                            ; Return PSW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

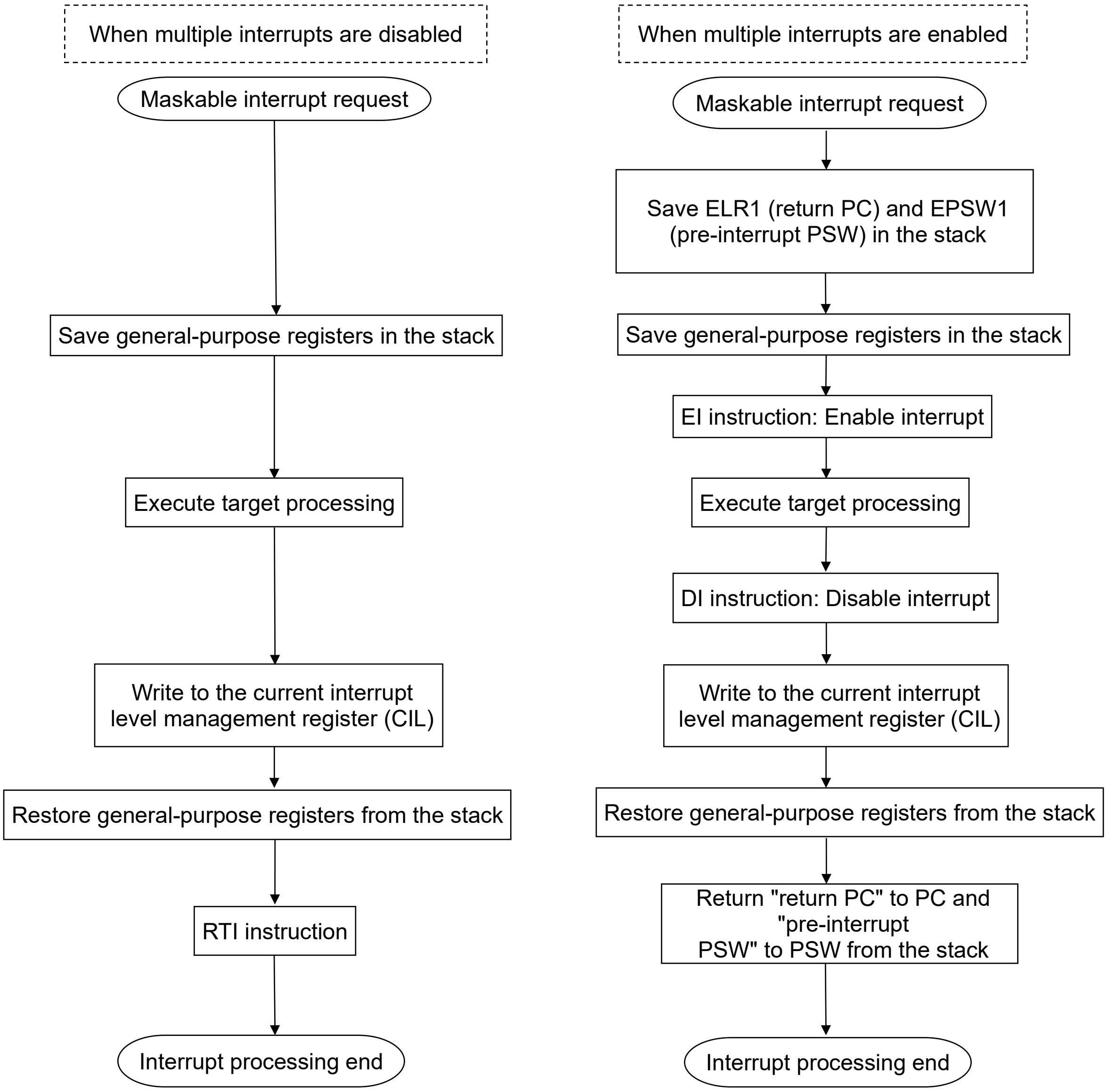

When multiple interrupts are disabled
Maskable interrupt request
Save general-purpose registers in the stack
Execute target processing
Write to the current interrupt level management register (CIL)
Restore general-purpose registers from the stack
RTI instruction
Interrupt processing end
When multiple interrupts are enabled
Maskable interrupt request
Save ELR1 (return PC) and EPSW1 (pre-interrupt PSW) in the stack
Save general-purpose registers in the stack
EI instruction: Enable interrupt
Execute target processing
DI instruction: Disable interrupt
Write to the current interrupt level management register (CIL)
Restore general-purpose registers from the stack
Return "return PC" to PC and "pre-interrupt PSW" to PSW from the stack
Interrupt processing end

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 Source Clocks

Clock Name	Symbol	Frequency	Description
Internal low-speed clock	RC32K	32.768kHz	Internal generated RC oscillating low-speed clock.
Crystal oscillation low-speed clock	XT32K	32.768kHz	low-speed clock with an external crystal unit.
External low-speed clock input	EXT32K	32.768kHz	External low-speed clock input from XT1
Internal 1kHz clock	RC1K	1.024kHz	Internal generated RC oscillating clock that frequency is 1.024 kHz for WDTCLK and clock mutual monitor.
Internal high-speed clock	PLL	-	high-speed clock multiplied LSCLK0. It is selected frequency 1/16/24MHz by the code option.
	PLL24M	24.002560MHz	Multiplied by 732.5
	PLL16M	16.007168MHz	Multiplied by 488.5
	PLL1M	0.999424MHz	Multiplied by 30.5

Table 6-2 Clocks generated by the clock generation circuit

Clock Name	Symbol	Frequency	Description
Low speed clock 0	LSCLK0	32.768kHz	For system and peripheral circuits.
Low speed clock 1	LSCLK1	32.768kHz	For timer and LTBC.
Low speed clock output	LCKO	32.768kHz	port output LSCLK0 orLSCLK1
High speed source clock	HSOCLK	1/16/24MHz	For SA-ADC.
High speed clock	HSCLK	0.125 to 24MHz	Divided HSOCLK. For system and peripheral circuits.
High speed time base clock 0/1	HTBCLK0 HTBCLK1	HSCLK/1 to 8	For peripheral circuits. Divided HSCLK by 1, 2, 3, 4, 5, 6, 7, 8. 2 channels.
High speed clock output	HCKO	0.125 to 6MHz	Port output divided HSOCLK
System clock	SYSCLK	32.768kHz/ 0.125 to 24MHz	A clock. selected from LSCLK0 and a divided HSCLK, for system and communication circuits. It is LSCLK at start-up.
CPU clock	CPUCLK	SYSCLK	A system clock for CPU. It is stop in stand-by mode.
WDT clock	WDTCLK	1.024kHz	For counting watchdog timer.

6.1.1 Features

- Low-speed clock generation circuit
  - Low-speed RC oscillation circuit
  - A crystal resonator is connectable
  - In case the low-speed crystal oscillation stopped, the clock is automatically switched to the low-speed RC oscillation; clock backup function.
  - A low-speed external clock is available to input to XT1 pin.
  - Selectable a clock different from system clock as timer clock.
  - Configurable stability time of crystal oscillation by the timer X.
- High-speed clock generation circuit
  - PLL oscillation mode (1/16/24 MHz is selectable for the PLL frequency by the code option)
  - High-speed clock wake-up time is selectable
  - High-speed time base clock circuit that is divided HSCLK by 1 to 8; it is with 2 channels.
- WDT clock
  - RC 1kHz oscillation circuit that is different from system clock.

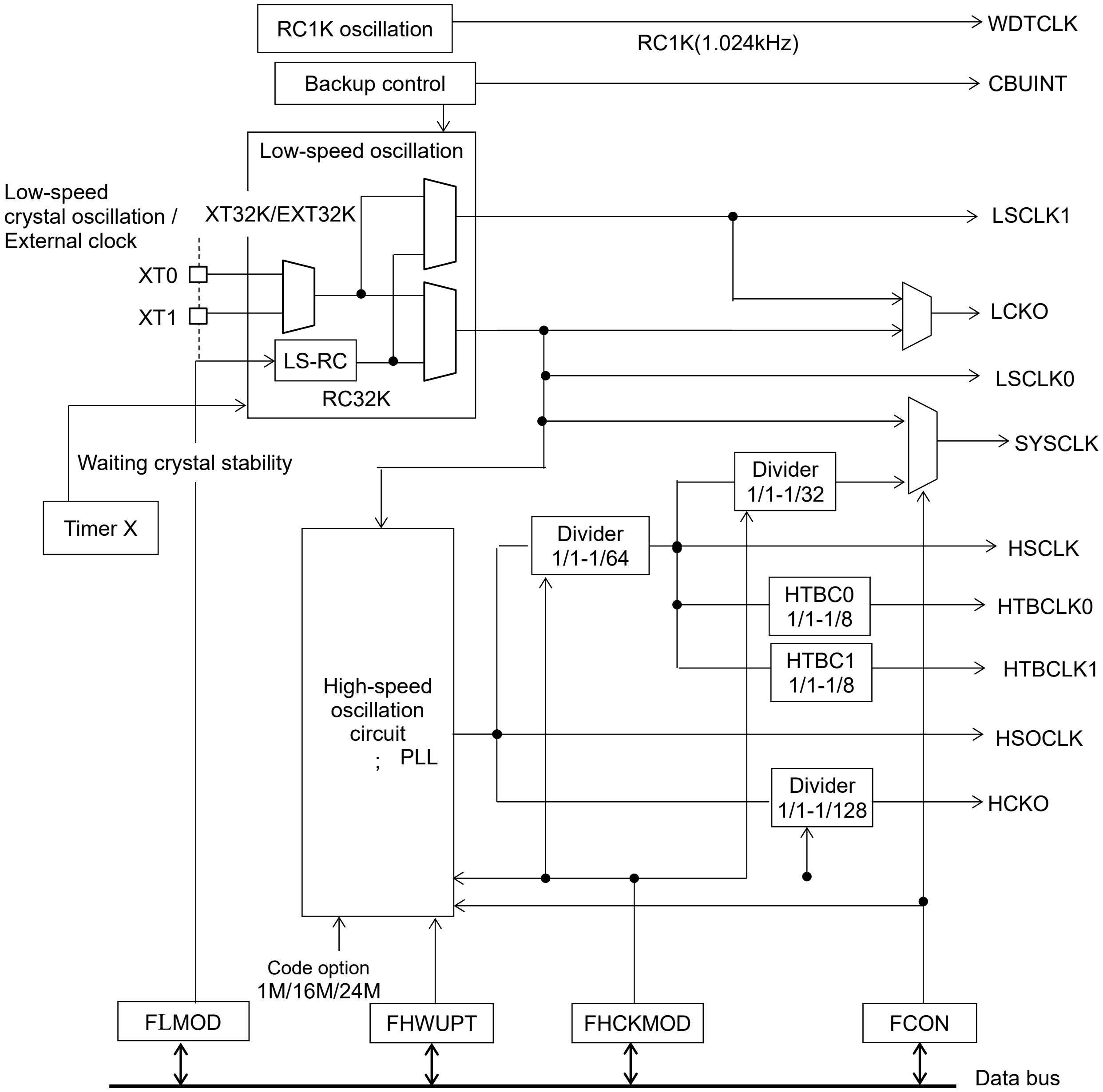
Table 6-3 shows relation of high-speed clocks and PLL frequency.
The CPU operation mode and the PLL oscillation reference frequency are selectable by the code option. See Chapter 30 "Code Option" for more details.

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

PLL	Maximum operating frequency				output frequency (maximum)
	HSOCLK	HSCLK	SYSCLK		HCKO
			Wait mode	No wait mode	
24MHz	24MHz	0.375-24MHz	0.125-24MHz	0.125-6MHz	6MHz
16MHz	16MHz	0.250-16MHz	0.125-16MHz	0.125-8MHz	4MHz
1MHz	1MHz	0.125-1MHz	0.125-1MHz	0.125-1MHz	1MHz

6.1.2 Configuration

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



FLMOD, FHCKMOD, FHWUPT, FCON : SFRs for control.
CBUINT[*1] : Clock backup interrupt register

Figure 6-1 Configuration of Clock Generation Circuit

Table 6-4 Operating clock list in each function

Function	SYSCLK	LSCLK0	LSCLK1	HSCLK	HSOCLK	HTBCLK0	HTBCLK1	WDTCLK
CPU/Data bus/CRC	●	-	-	-	-	-	-	-
RAM	●	-	-	-	-	-	-	-
Watchdog timer	-	-	-	-	-	-	-	●
External interrupt control	-	●*1	-	●*1	-	-	●*1	-
Low-speed time base counter	-	●	●	-	-	-	-	-
16-bit timer 0-4	-	●	●	●	-	●	●	-
16-bit timer X	-	●	●	●	-	●	●	-
Functional timer	-	●	-	●	-	●	●	-
SSIO	●	-	-	-	-	-	-	-
SSIOF	●	-	-	-	-	-	-	-
UART	●	●	-	●	-	-	-	-
I^{2}C bus unit I^{2}C bus master	●	-	-	-	-	-	-	-
SA type A/D converter	-	●	-	-	●	-	-	-
VLS	-	●*1	-	-	-	-	-	-

● : supplied. - : not supplied.

*1 : for controlling to start or sampling.

[Note]

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

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. Table 6-5 shows the list of the output ports and the register setting.

Pin name	I/O	Function
LCKO	O	Low-speed clock output
HCKO	O	High-speed clock output
XT0	I	Low-speed crystal resonator connect pin
XT1	O/I	Low-speed crystal resonator connect pin / Low-speed external clock input pin

Table 6-5 Clock output function port and the register setting

Pin name	Shared port		Setting register	Setting value	ML62Q2500 group
LCKO	P03	7th function	P0MOD3	0110_XXXX*1	●
	P11	7th function	P1MOD1	0110_XXXX*1	●
	P71	7th function	P7MOD1	0110_XXXX*1	●
HCKO	P07	7th function	P0MOD0	0110_XXXX*1	●
	P10	7th function	P2MOD0	0110_XXXX*1	●

*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)

[Note]

- **Assign HCKO function to only one LSI pin.**

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	0x43
0xF004	Low-speed clock mode register	FLMODL	FLMOD	R/W	8/16	0x00
0xF005		FLMODH		R/W	8	0x00
0xF006	Clock control register	FCON	FCONW	R/W	8/16	0x00
0xF007		FCON1	-	R/W	8	0x00
0xF008	High-speed clock wake up time setting register	FHWUPT	-	R/W	8	0x00
0xF009	Reserved	-	-	-	-	-
0xF00A	Reserved	-	-	-	-	-
0xF00B	Reserved	-	-	-	-	-
0xF00C	Backup Clock Status register	FBUSTAT	FBUSTATW	R/W	8/16	0x01
0xF00D		FBUSTATH	-	R	8	0x01
0xF080	Reserved	-	-	-	-	-
0xF086	High speed time base clock setting register	HTBDR	-	R/W	8	0x00
0xF087	Reserved	-	-	-	-	-
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	-	-	-	-	-

6.2.2 High-Speed Clock Mode Register (FHCKMOD)

This is a SFR to choose the oscillation mode of the high-speed clock oscillation circuit and the frequency of high-speed clock.

Address :	0xF002(FHCKMODL/FHCKMOD), 0xF003(FHCKMODH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x4300

	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	-	HSC2	HSC1	HSC0	-	-	-	-
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R
Initial value	0	1	0	0	0	0	1	1	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 (HCKO) output from the general port. This clock frequency is limited 6MHz or lower. 000 : HSOCLK 001 : 1/2 HSOCLK 010 : 1/4 HSOCLK 011 : 1/8 HSOCLK 100 : 1/16 HSOCLK (Initial value) 101 : 1/32 HSOCLK 110 : 1/64 HSOCLK 111 : 1/128 HSOCLK
11	-	Reserved bits
10 to 8	SYSC2 to SYSC0	These bits are used to set frequency of system clock with high-speed. SYSC2 bit is not writable at PLL1M mode. Choose a proper division ratio of the frequency, so that the frequency does not exceed the maximum/minimum frequency of the CPU operating frequency shown in the Table 6-6 "CPU operation mode and PLL oscillation reference frequency". If the setting exceeds the clock frequency range in Table 6-3, it will be corrected internally. However, the SFR read value is the written value.

SYSC	PLL mode	
	24MHz/16MHz	1MHz
000	HSCLK *1	←
001	1/2 HSCLK *1	←
010	1/4 HSCLK	←
011	1/8 HSCLK(Initial value)	←
100	1/16 HSCLK	-
101	1/32 HSCLK	-
110	Do not use (1/32 HSCLK)	-
111	Do not use (1/32 HSCLK)	-

*1 : Actual setting is changed to 1/1, 1/2, 1/4 HSCLK according to HSC setting with no-wait mode

SYSC	HSC	PLL mode	Actual setting
000	000	24MHz	1/4 HSCLK (1/4 HSOCLK)
		16MHz	1/2 HSCLK (1/2 HSOCLK)
	001	24MHz	1/2 HSCLK (1/4 HSOCLK)
001	000	24MHz	1/4 HSCLK (1/4 HSOCLK)

Bit No.	Bit symbol name	Description
7	-	Reserved bits

<table>
<tr><th>Bit No.</th><th>Bit symbol name</th><th>Description</th></tr>
<tr><td>6 to 4</td><td>HSC2 to HSC0</td><td>These bits are used to set frequency of HSCLK. These bits are writable if ENOSC=0.
HSC2 bit is not writable at PLL1M mode.
<table>
<tr><th rowspan="2">HSC</th><th colspan="2">PLL mode</th></tr>
<tr><th>24MHz/16MHz</th><th>1MHz</th></tr>
<tr><td>000</td><td>HSOCLK (Initial value)</td><td>←</td></tr>
<tr><td>001</td><td>1/2 HSOCLK</td><td>←</td></tr>
<tr><td>010</td><td>1/4 HSOCLK</td><td>←</td></tr>
<tr><td>011</td><td>1/8 HSOCLK</td><td>←</td></tr>
<tr><td>100</td><td>1/16 HSOCLK</td><td>-</td></tr>
<tr><td>101</td><td>1/32 HSOCLK</td><td>-</td></tr>
<tr><td>110</td><td>1/64 HSOCLK</td><td>-</td></tr>
<tr><td>111</td><td>Do not use
(1/128 HSOCLK)</td><td>-</td></tr>
</table>
</td></tr>
<tr><td>3 to 0</td><td>-</td><td>Reserved bits</td></tr>
</table>

Table 6-6 HSC/SYSC setting and SYSCLK frequency [MHz]

PLL mode	SYSC/HSC	000	001	010	011	100	101	110
24MHz	000	24(6)	12(6)	6	3	1.5	0.75	0.375
	001	12(6)	6	3	1.5	0.75	0.375	0.187
	010	6	3	1.5	0.75	0.375	0.187	0.187
	011	3	1.5	0.75	0.375	0.187	0.187	0.187
	100	1.5	0.75	0.375	0.187	0.187	0.187	0.187
	101	0.75	0.375	0.187	0.187	0.187	0.187	0.187
16MHz	000	16(8)	8	4	2	1	0.5	0.25
	001	8	4	2	1	0.5	0.25	0.125
	010	4	2	1	0.5	0.25	0.125	0.125
	011	2	1	0.5	0.25	0.125	0.125	0.125
	100	1	0.5	0.25	0.125	0.125	0.125	0.125
	101	0.5	0.25	0.125	0.125	0.125	0.125	0.125
1MHz	000	1	0.5	0.25	0.125	-	-	-
	001	0.5	0.25	0.125	0.125	-	-	-
	010	0.25	0.125	0.125	0.125	-	-	-
	011	0.125	0.125	0.125	0.125	-	-	-

A value in () is frequency when no-wait mode.

6.2.3 Low-speed Clock Mode Register (FLMOD)

This is a SFR to control the low speed clock.
This register(except OUTLS bit) is initialized by only the Power-on-reset and pin reset.

Address :	0xF004(FLMODL/FLMOD), 0xF005(FLMODH)
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	FLMOD															
Byte	FLMODH								FLMODL							
Bit	OUTLS	-	-	-	-	L1CEN	LOSCMD1	LOSCMD0	LMOD1	LMOD0	-	LFLTSEL	-	L1SEL	-	L0SEL
R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R	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
15	OUTLS	This is used to choose clock output as LCKO. This bit is initialized by all system reset. 0 : LSCLK0 (Initial value) 1 : LSCLK1
14 to 11	-	Reserved bits
10	L1CEN	This is used to set enable/stop LSCLK1 oscillation. This is writable only when L1SEL is "0". 0 : LSCLK1 oscillation stop (Initial value) 1 : LSCLK1 oscillation enabled
9,8	LOSCMD1, LOSCMD0	These bits are used to choose crystal oscillation or external input, and to enable crystal oscillation circuit. 00: Disabled the crystal oscillation circuit. (Initial) 01: Enabled the crystal oscillation circuit as crystal oscillation mode; XT32K mode 10: Enabled the crystal oscillation circuit as external clock input mode; EXT32K mode 11: Do not use; invalid. These bits are cleared to "00" if entering STOP/STOP-D mode when these bits are "01"/"10", and then the crystal oscillation circuit turns off. Writing only "00" to these bits is enabled when these bits are "01"/"10".
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; LP mode 10: Tough mode 11: Ultra low power consumption mode; ULP mode The LP mode reduces the current consumption by lowering the oscillation margin than the standard mode. The power consumption in the ULP mode is lower than in the LP 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 changeable when the LOSCMD1-0 bits are "00". 0: Without the noise filter (Initial value) 1: With the noise filter
3	-	Reserved bit

Bit No.	Bit symbol name	Description
2	L1SEL	This bit is configured a source clock of LSLK1. It is unwritable when LOSCMD1-0 is “01 or 10” and “LOSCS=1 or LOSCB=1”. However it is writable when the backup function is disabled, even if LOSCB=1. It is cleared when the LOSCMD1-0 become “00”. 0 : RC32K (Initial value) 1 : XT32K/EXT32K (selected by LOSCMD1-0)
1	-	Reserved bits
0	L0SEL	This bit is configured a source clock of LSLK0. It is unwritable when LOSCMD1-0 is “01 or 10” and “LOSCS=1 or LOSCB=1”. It is cleared when the LOSCMD1-0 become “00”. 0 : RC32K (Initial value) 1 : XT32K/EXT32K (selected by LOSCMD1-0)

6.2.4 Clock Control Register (FCONW)

This is a SFR to control the clock generation circuit.

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FCONW															
Byte	FCON1								FCON							
Bit	ENRC1K	-	-	-	-	-	-	-	LPLL	-	-	-	-	-	ENOSC	SELSCLK
R/W	R/W	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	ENRC1K	It is used to be forcibly turned on RC1K regardless of code-option. The RC1K oscillates in the standby mode, when this bit is “1”. However WDTCLK is supplied accordance with code-option for WDT. This function is of service to mutual monitoring for low-speed clock if the WDT is not in use. 0 : Depend on WDT code-option (Initial value) 1 : Enabled
14 to 8		Reserved bits
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: Disabled/turn off the high-speed clock oscillation (Initial value) 1: Enabled/turn on the high-speed clock oscillation
0	SELSCLK	This bit is used to choose the system clock. This bit is unwritable at ENOSC=0. This bit and ENOSC bit can be set “1” at once. 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: LSCLK0 (Initial value) 1: High-speed clock chosen by the SYSC2 to SYSC0 bit

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

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FHWUPT							
Bit	-	-	-	-	-	-	-	-	-	FHRDWN	-	-	-	-	-	FHUT0
R/W	R	R	R	R	R	R	R	R	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
7	-	Reserved bit
6	FHRDWN	This bit is used to control PLL at wake-up from HALT-H. This function controls PLL to avoid abnormal frequency caused by temperature difference between stand-by enter and wake-up. Set this bit to “1”, if FHUT0=1 and the temperature at the wake-up drops by or more than the following value from stand-by entry. PLL24M mode: 15°C PLL16M mode: 20°C However, this setting is not required in the PLL1M mode. 0 : Disabled (Initial value) 1 : Enabled
5 to 1	-	Reserved bits
0	FHUT0	This bit is used to configured the timing of starting to supply the high-speed clock oscillation when ENOSC=1. 0 : After Locked; the frequency clock is stabilized. (Initial value) A system clock stops until the high-speed clock is locked, if SELSCLK is set to “1”. 1 : Soon after ENOSC=1; approx. 30μs. The frequency is not guaranteed as the specification, however it is useable for the system clock.

6.2.6 High speed Time Base Clock Setting Register (HTBDR)

This is a SFR to set frequency of high speed time base clock.

Address : 0xF086 (HTBDR)
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	-								HTBDR							
Bit	-	-	-	-	-	-	-	-	-	HT1D2	HT1D1	HT1D0	-	HT0D2	HT0D1	HT0D0
R/W	R	R	R	R	R	R	R	R	R	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
7	-	Reserved bits
6 to 4	HT1D2 to HT1D0	These are used to set frequency of HTBCLK1. 000 : HSCLK (Initial value) 001 : HSCLK / 2 010 : HSCLK / 3 011 : HSCLK / 4 100 : HSCLK / 5 101 : HSCLK / 6 110 : HSCLK / 7 111 : HSCLK / 8
3	-	Reserved bits
2 to 0	HT0D2 to HT0D0	These are used to set frequency of HTBCLK0. 000 : HSCLK (Initial value) 001 : HSCLK / 2 010 : HSCLK / 3 011 : HSCLK / 4 100 : HSCLK / 5 101 : HSCLK / 6 110 : HSCLK / 7 111 : HSCLK / 8

6.2.7 Backup Clock Status Register (FBUSTATW)

This is a SFR to indicate status of low speed clock. This is initialized by only Power-on-reset and pin reset.

Address :	0xF00C (FBUSTAT/FBUSTATW), 0xF00D (FBUSTATH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0101

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FBUSTATW															
Byte	FBUSTATH								FBUSTAT							
Bit	-	-	-	-	L1XT	L1RC	L0XT	L0RC	-	-	-	-	-	-	LOSCB	LOSCS
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R
Initial value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit No.	Bit symbol name	Description
15 to 12	-	Reserved bits
11	L1XT	This is flag LSCLK1 is operating with XT32K/EXT32K.
10	L1RC	This is flag LSCLK1 is operating with RC32K.
9	L0XT	This is flag LSCLK0 is operating with XT32K/EXT32K.
8	L0RC	This is flag LSCLK0 is operating with RC32K.
1	LOSCB	This is a flag of the crystal oscillation stop detection. This is cleared to “0” when writing “1” to this bit. 0 : Not stop detection or after clearing flag (Initial value)。 1 : Stop detection. Then LSCLK0 is changed to RC32K by backup function. The backup of LSCLK1 depends on setting to code option. At once, the clock backup interrupt; QCBU bit of IRQ23 register is asserted. This function is enabled at the following conditions: - selected XT32K/EXT32K to LSCLK0; L0SEL=1, - selected XT32K/EXT32K to LSCLK1; L1SEL=1
0	LOSCS	This bit is used to indicate the status of low-speed crystal oscillation clock. When setting XT32K/EXT32K mode with LOSCMD1-0 bits of FLMOD register, this bit is changed to “0” after 16-bit timer X interrupt occurred. This interrupt occurs when the timer X count by XT32K/EXT32K clock coincides with TMHXD value. This bit becomes to “1” on the following conditions. - When the crystal oscillation turn off by the software (LOSCMD1-0=“00”). The interrupt does not occur in this time. - At entry to the STOP/STOP-D mode. The interrupt does not occur in this time. - At detected oscillation stopping. The interrupt; QCBU of IRQ23, occurs in this time. 0 : XT32K/EXT32K is enabled; stable. 1 : XT32K/EXT32K is disabled; not stable/malfunction stopping (Initial value)

6.2.8 Clock Backup Test Mode Acceptor (FBTACP)

This is a write-only SFR to enable writing to Clock Backup Test Mode register (FBTCON).
This is used to prevent erroneous writing to the FBTCON register.

Address :	0xF0C4 (FBTACP)
Access :	W
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FBTACP							
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

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 Register (FBTCON)

This is a SFR to control the clock backup test mode. The clock backup test mode can make purposely the condition that stops the low-speed crystal oscillation (XT32K/EXT32K).

Address :	0xF0C6 (FBTCON)
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	-								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	This is used to select fixed level of low-speed crystal oscillation clock when LOSCT=1. When the LOSCT bit is set to "1", the LOSCL bit determines the fixed level of low-speed crystal oscillation clock. 0: Fixed to "L" level (Initial value) 1: Fixed to "H" level
0	LOSCT	This bit enables the clock backup test mode. Use the clock backup test mode after setting the low-speed crystal oscillation clock mode; XT32K/EXT32K mode. 0: Normal mode (Initial value) 1: Clock backup test mode

6.3 Description of Operation

6.3.1 Low-Speed Clock

The low-speed clock generation circuit supply LSCLK0/LSCLK1 that is selected the following clock source by the L0SEL/L1SEL bit of FLMOD register.

- Low-speed RC oscillation mode (RC32K)
- Low-speed crystal oscillation mode (XT32K) / external clock input mode (EXT32K)

A low-speed clock is output as LCKO. It is selected LSCLK0/LSCLK1 by the OUTLS bit of FLMOD register.
See Table 1-3 pin list for assignment port function.
When a system reset is released, the low-speed clock (LSCLK0, LSCLK1, LCKO) is output and the CPU runs a program after 512 counts of clock chosen for LSCLK0.
The LSCLK0 is RC32K when the power on reset or pin reset is released.

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

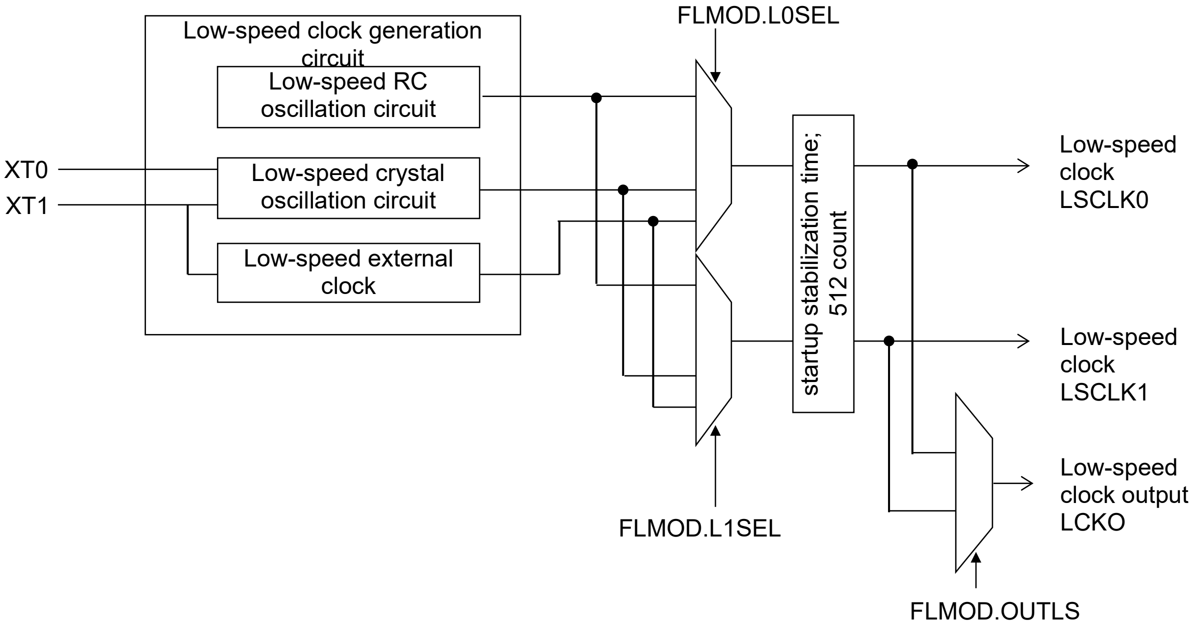

FLMOD.L0SEL
Low-speed clock generation circuit
Low-speed RC oscillation circuit
XT0
XT1
Low-speed crystal oscillation circuit
Low-speed external clock
startup stabilization time; 512 count
Low-speed clock LSCLK0
Low-speed clock LSCLK1
Low-speed clock output LCKO
FLMOD.L1SEL
FLMOD.OUTLS

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

[Note]

- **The LCKO output operation is not guaranteed in the HALT-D mode.**

6.3.1.1 Low-Speed RC Oscillation Circuit

The low-speed RC oscillation clock is chosen for the system clock at the power on.
When a system reset is released, the low-speed clock (LSCLK0, LSCLK1) is output and the CPU runs a program after 512 counts of clock chosen for LSCLK0. When the STOP/STOP-D mode is released, the low-speed clock is output and the CPU runs a program after the low-speed RC oscillation startup time (T_{RCL}) and stability time.
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.

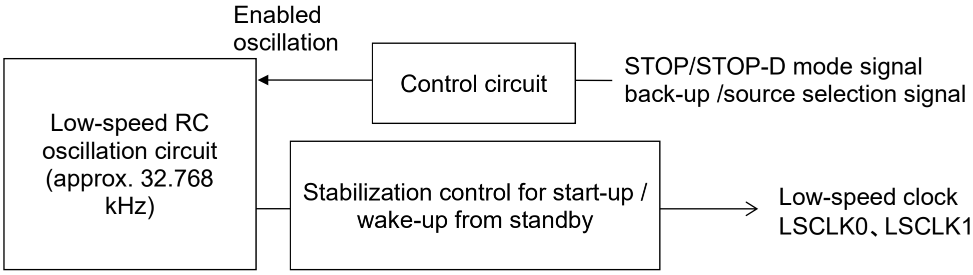

Enabled oscillation
Control circuit
STOP/STOP-D mode signal back-up /source selection signal
Low-speed RC oscillation circuit (approx. 32.768 kHz)
Stabilization control for start-up / wake-up from standby
Low-speed clock LSCLK0、LSCLK1

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

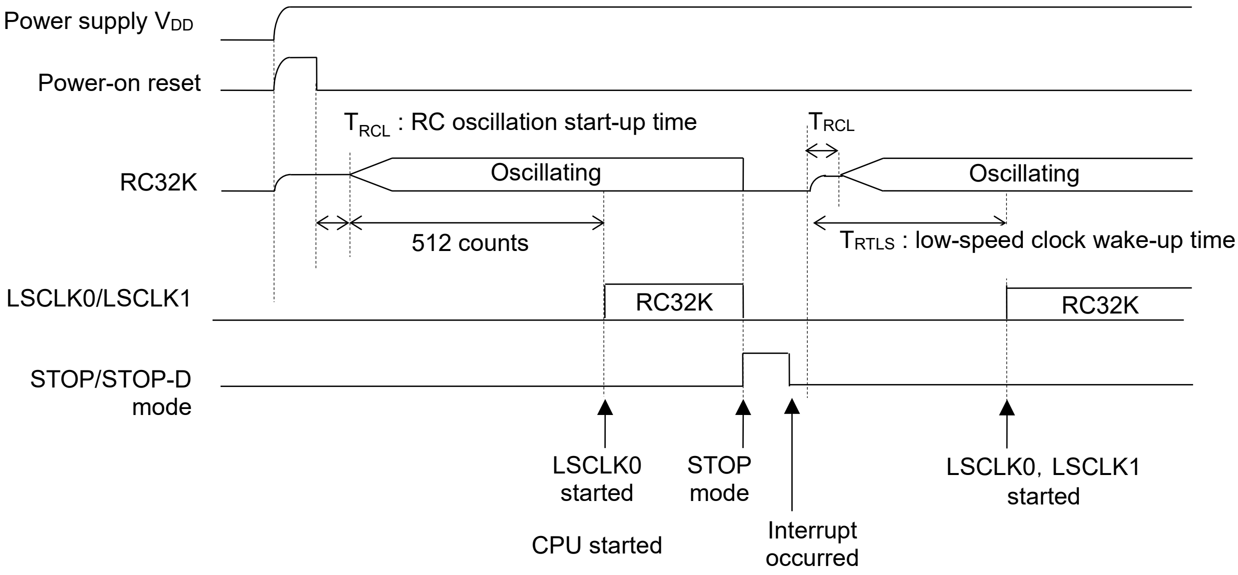


Figure 6-4 Low-speed Clock Operation Waveforms at Start of Low-speed RC Oscillation Circuit and in STOP/STOP-D Mode

6.3.1.2 Low-Speed Crystal Oscillation Circuit

Figure 6-5 shows a configuration of the low-speed crystal oscillation circuit.
The XT32K mode using 32.768 kHz crystal unit or EXT32K mode that clock input from XT1 pin can be chosen in the low-speed clock mode register (FLMOD).
In XT32K mode, the setting of PXTMOD01 register is ignored.
In EXT32K mode, the setting of PXTMOD1 is ignored.

It is necessary to use Timer X when XT32K/EXT32K is used as LSCLK0/1. It can be used as a normal timer when the oscillation stabilization waiting is completed.
The backup function is the function that always monitors the oscillation. If an oscillation stop is detected, it switches the low-speed clock to the low-speed RC oscillation clock.
The switching to RC oscillation for LSCLK1 can be invalid by code option, and it can only indicate the stop detection.
This oscillation circuit stops when STOP/STOP-D mode entry.

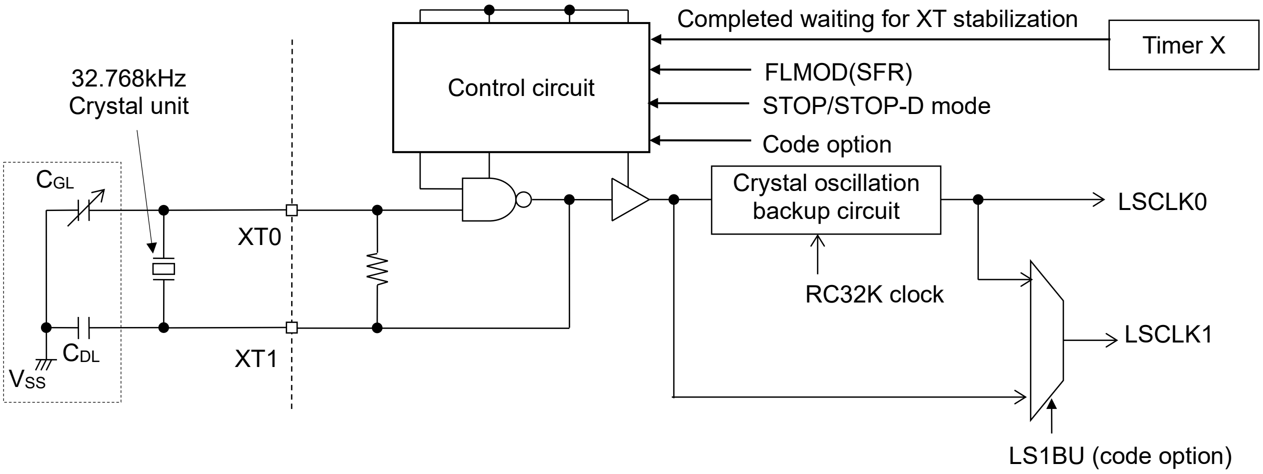


Figure 6-5-1 Circuit Configuration of XT32K mode

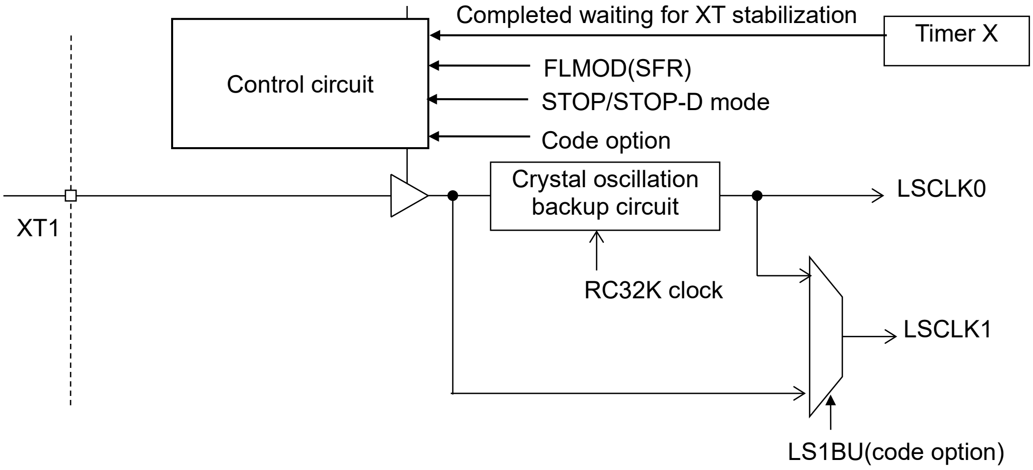


Figure 6-5-2 Circuit Configuration of EXT32K mode

[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 6.3.1.3 "Low-Speed Clock Control".**

Figure 6-6 shows backup mode waveforms at the startup 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 power-on reset and pin reset, the oscillation will continue even if other system reset occurs after choosing the XT32K / EXT32K mode.
The L0SEL/L1SEL should be set "1" after the crystal stable wait with timer X, and the LSCLK0/LSCLK1 is changed XT32K/EXT32K clock. The stable waiting time is configured with TMHXD register.
In addition, the low-speed crystal oscillation circuit turns off when entering the STOP/STOP-D mode. When the STOP/STOP-D mode is released by external interrupts and etc., LSCLK0/1 oscillation is in the RC32K mode.
See Chapter 4 "Power Management" for the STOP/STOP-D mode.
See the data sheet for the low-speed oscillation start time (T_{XTL}).

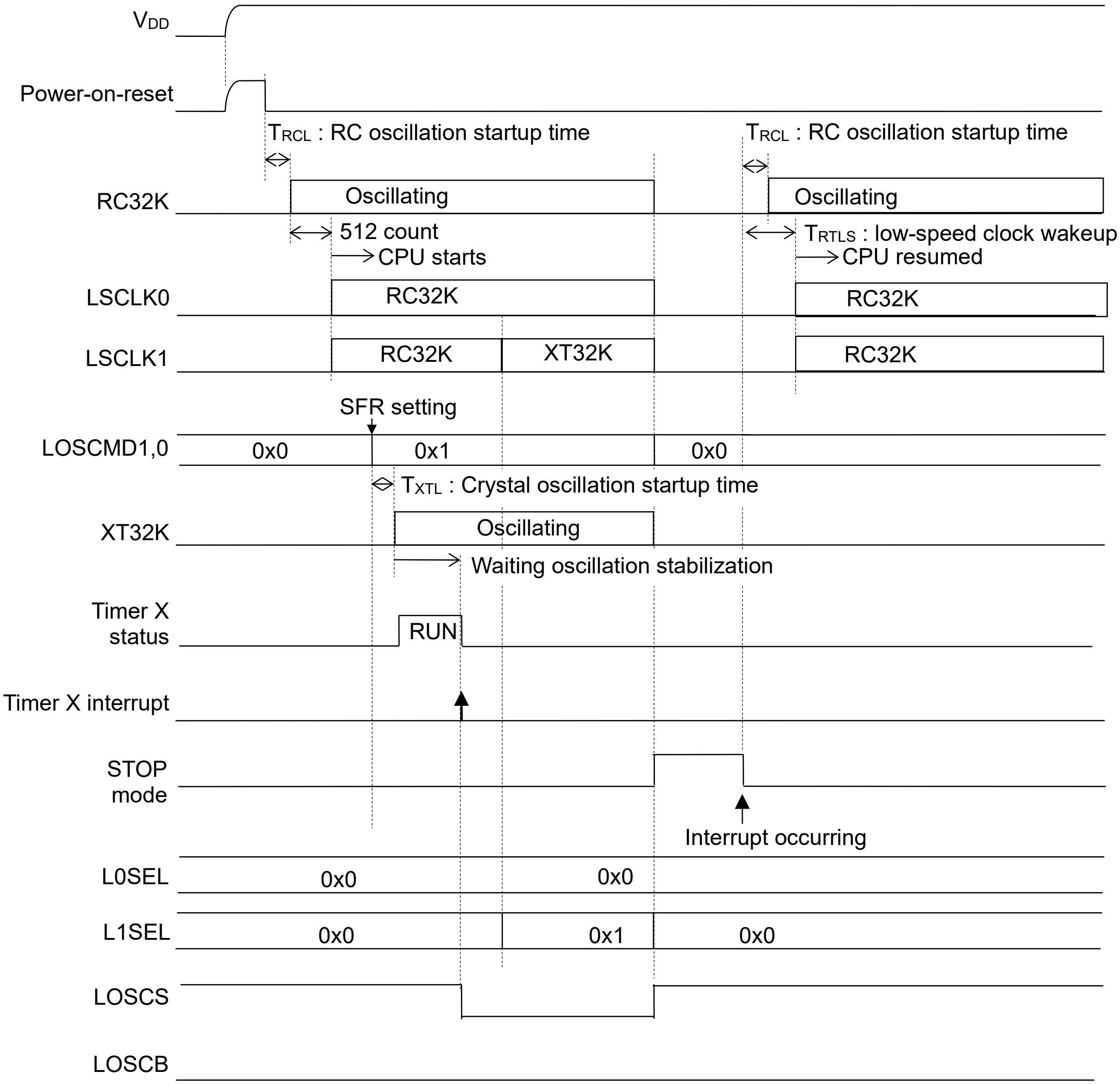


Figure 6-6 Low-speed Crystal Oscillation Circuit Operation (At startup and in the STOP mode)

Figure 6-7 shows operation waveform in the backup mode after the startup 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 2ms(typ.) later, and then the clock backup interrupt (CBUINT) occurs.

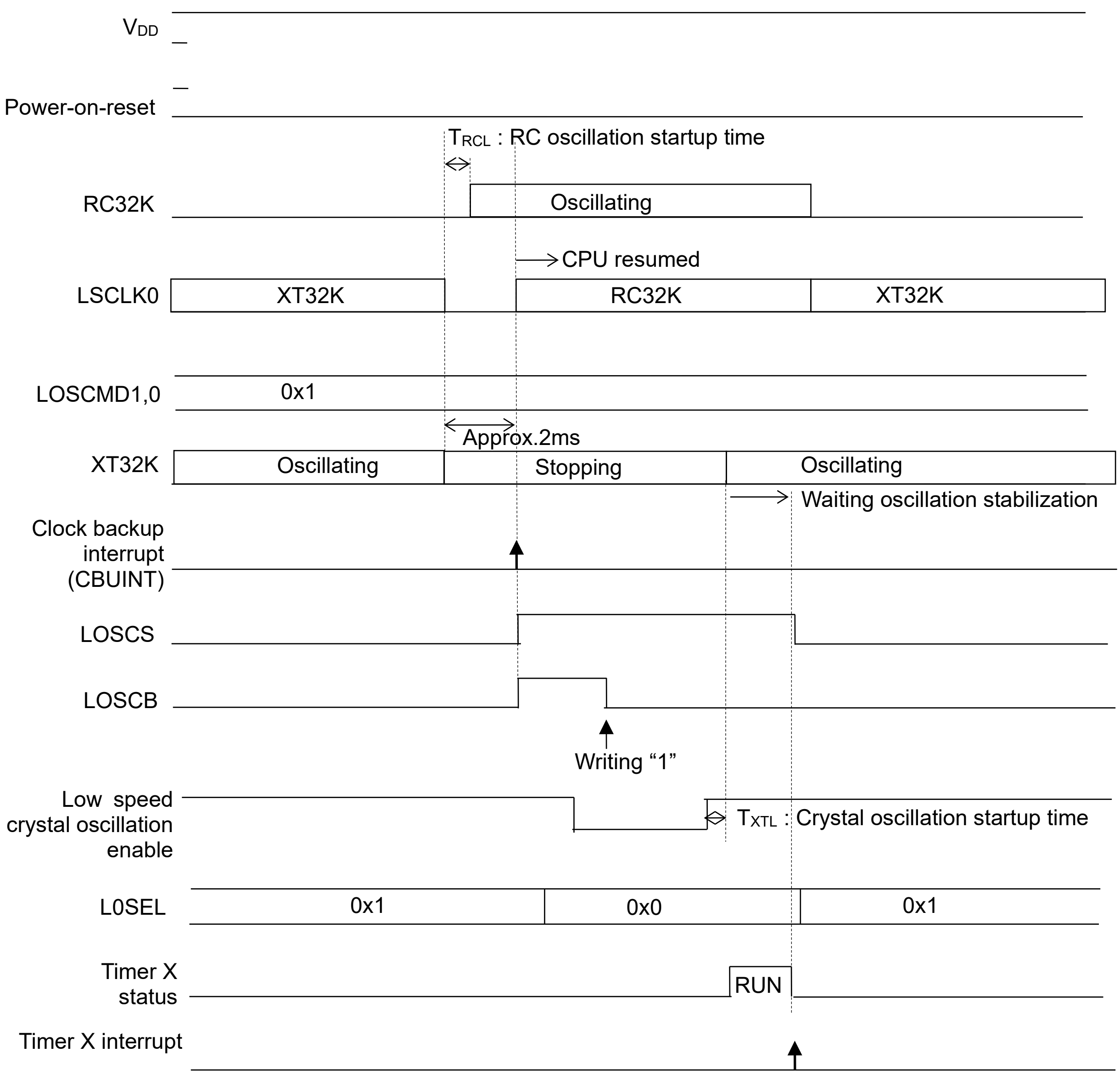


Figure 6-7 Low-speed Oscillation Circuit Operation in the backup mode

Figure 6-8 shows the operation waveforms when a reset occurs in the XT32K/EXT32K mode. The crystal oscillation circuit is not reset by anything other than power-on-reset and pin reset.
See Chapter 3 "Reset Function" for details of resetting.

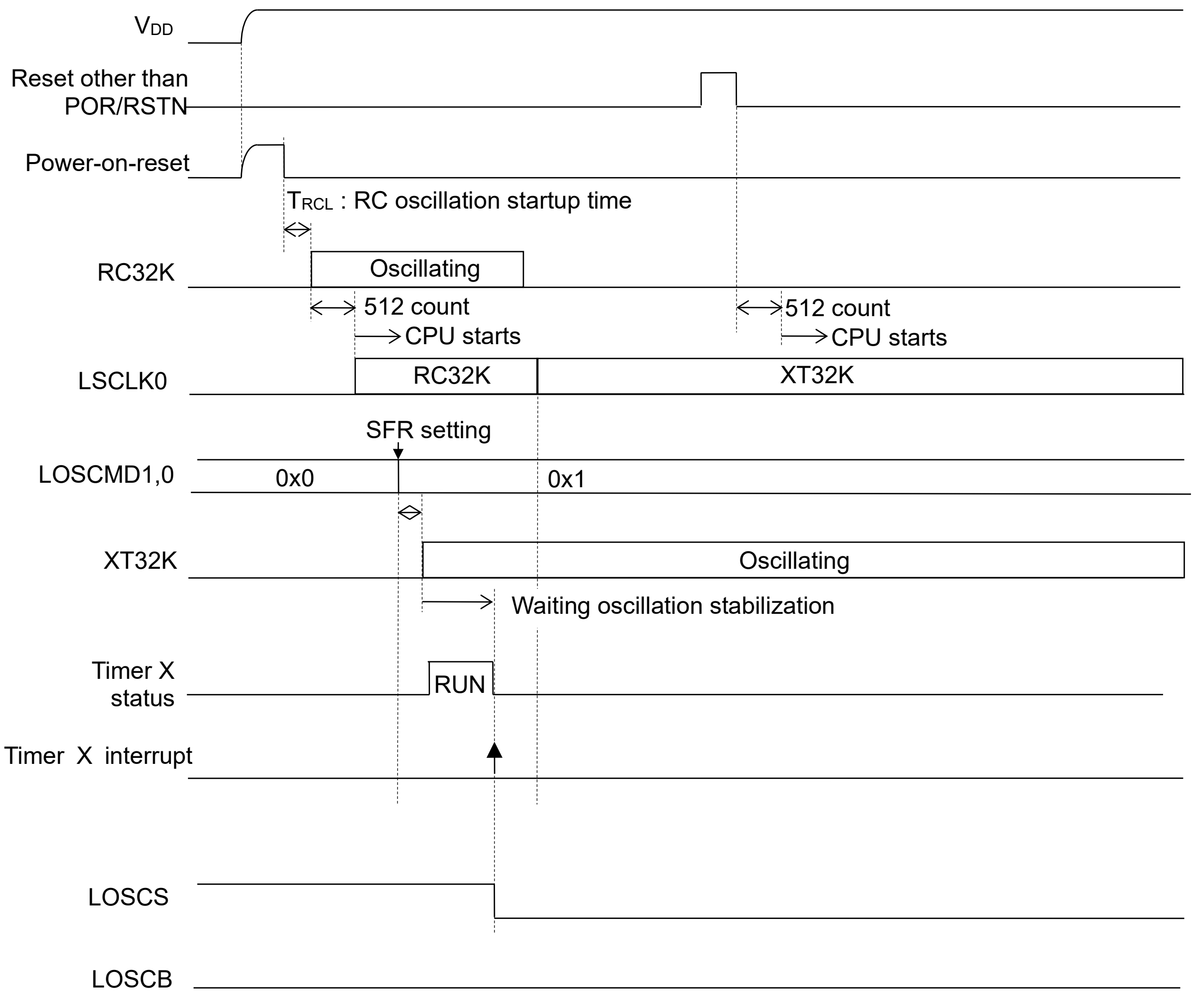
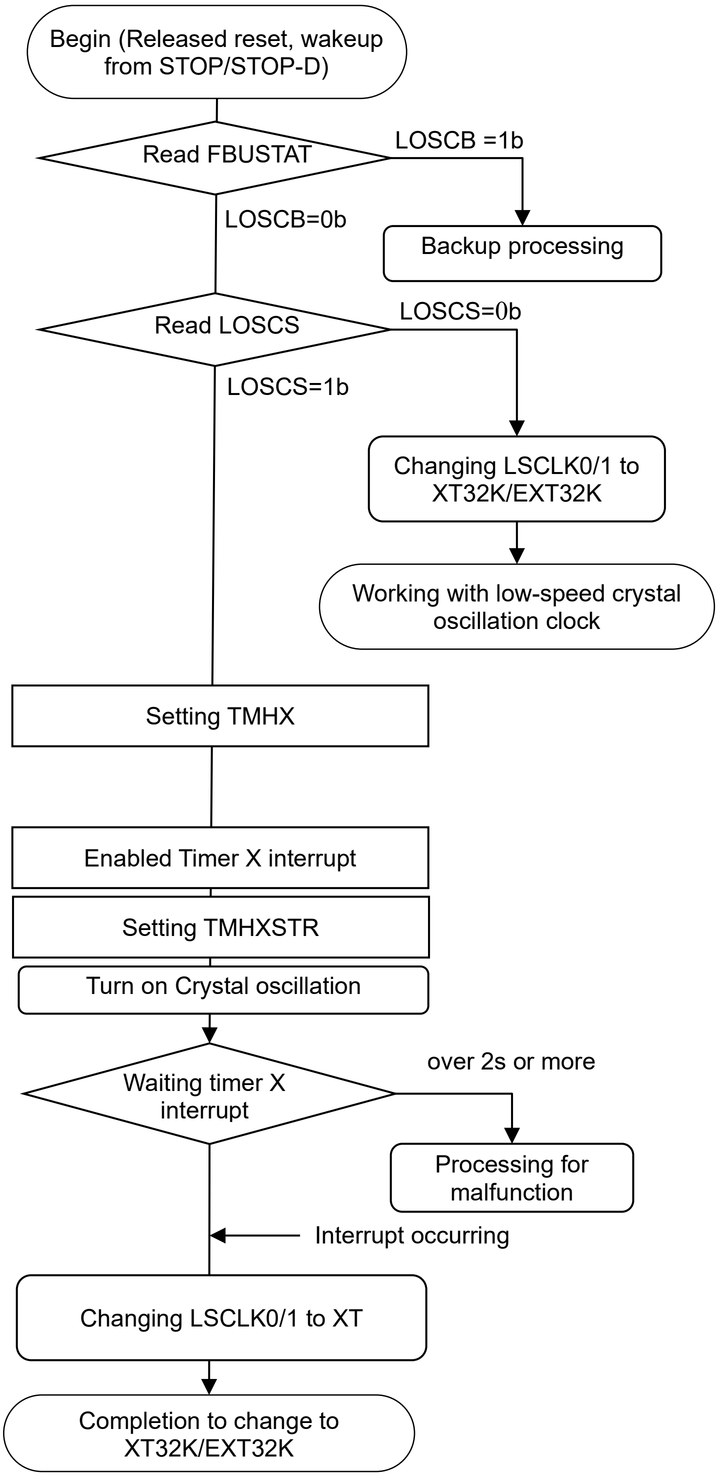


Figure 6-8 Low-speed Oscillation Circuit Operation (at System Reset after the crystal oscillation turned on)

6.3.1.3 Low-Speed Clock Control

Figure 6-9 shows a flow of the low-speed clock setting; from RC32K to XT32K/EXT32K. Use the flow at the wake up from STOP/STOP-D mode.
If using the high-speed clock a the system clock, change the low-speed clock to XT32K/EXT32K first.
See Chapter 8 “16-bit Timer” for timer function.


Begin (Released reset, wakeup from STOP/STOP-D)
Read FBUSTAT
LOSCB =1b
LOSCB=0b
Backup processing
Read LOSCS
LOSCS=0b
LOSCS=1b
Changing LSCLK0/1 to XT32K/EXT32K
Working with low-speed crystal oscillation clock
Setting TMHX
Enabled Timer X interrupt
Setting TMHXSTR
Turn on Crystal oscillation
Waiting timer X interrupt
over 2s or more
Processing for malfunction
Interrupt occurring
Changing LSCLK0/1 to XT
Completion to change to XT32K/EXT32K

Confirm LSCLK status for reading LOSCB bit of FBUSTAT.

LOSCB=1 : It is state the crystal oscillation stop is detected. Do backup processing.

LOSCS=0 : It is state the crystal oscillation stabilization wait is completed. If L0SEL/L1SEL is “0”, set it to “1” and switch to crystal oscillation.

LOSCB=0,LOSCS=1 : It is state the crystal oscillation stabilization wait is not completed.
Do crystal oscillation stabilization with Timer X.

Set stabilization wait cycles for TMHXD in the Timer X. (where, the number of cycles is the number of halves of the XT32K clocks.)
Clear TMHC and set to TMHXCON ;
TLXOST=1 : one-shot mode
THXCK=4 : low-speed crystal oscillation
THXDIV=arbitrary : set when the number of cycles is insufficient.

Enabled Timer-X interrupt.
Clear QTMX bit and Set “1” to ETMX bit.
Set THXRUN=1 to enable timer operation.
Write 1 or 2 to LOSCMD1-0 bits of FLMOD, and the wait oscillation stabilization.

If XT32K stops before the stabilizes , the Timer-X interrupt does not occur, so set timeout and handle appropriately for the application.

Change LSCLK0/LSCLK1 with L0SEL/L1SEL bit of FLMOD, after the timer X interrupt occurs.

Timer X can be in use as general timer, after completion to wait oscillation stabilization.

Figure 6-9 Procedure to change the low-speed clock from RC32K to XT32K/EXT32K

Handle appropriately for the application at the backup processing in Figure 6-9. Figure 6-10 shows the procedure to rechange LSCLK to RC32K.
Clear the LOSCB bit with referring to Figure 6-10, when the L1SEL=1 and backup function is enabled, or when L0SEL=1. Then handle the procedure in the Figure 6-9 if it will use the XT32K/EXT32K mode again.

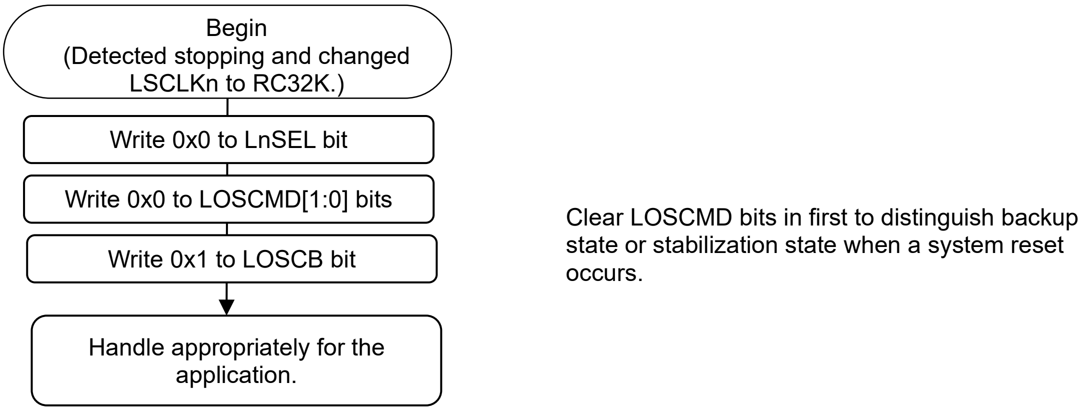
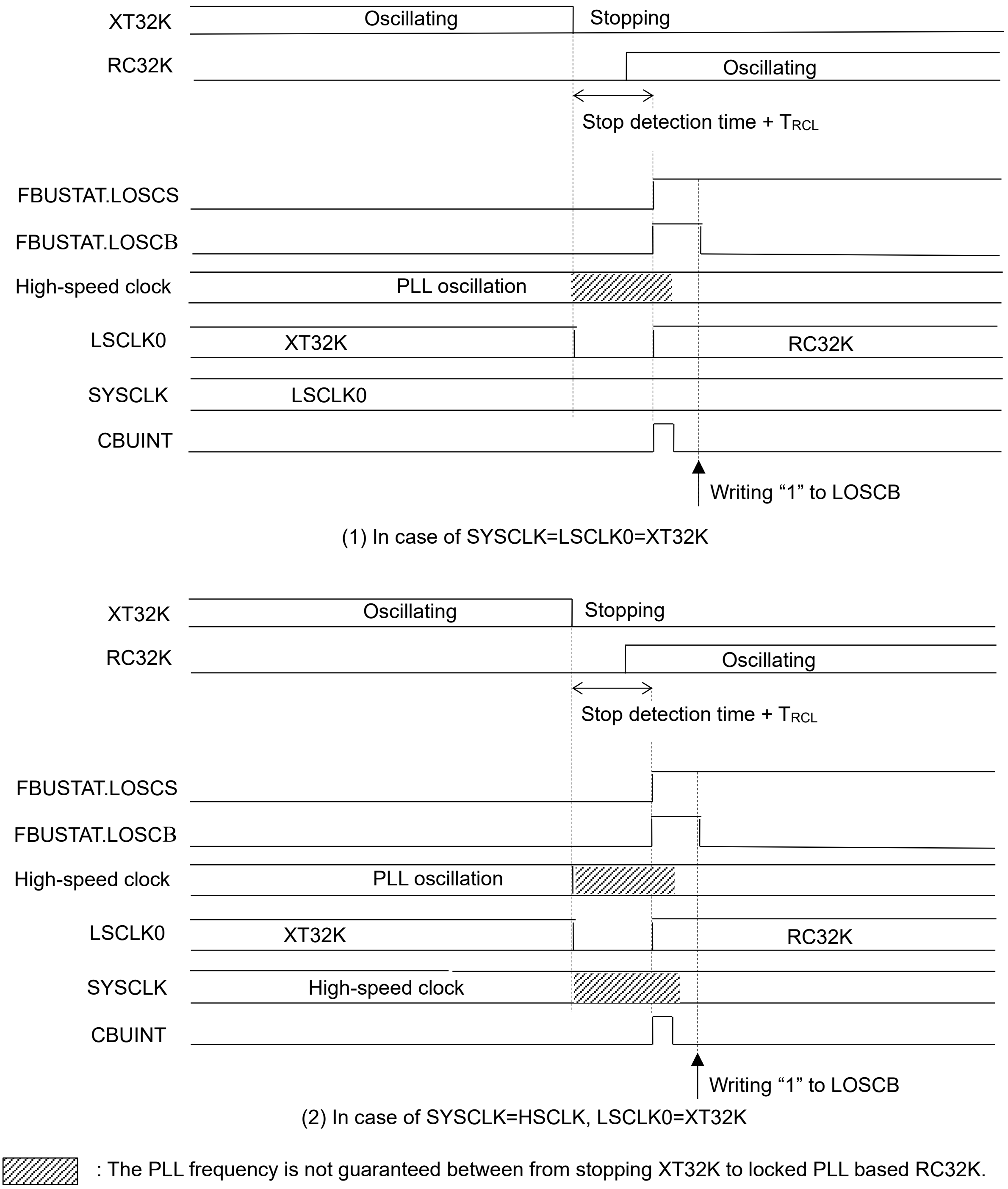

Begin
(Detected stopping and changed LSCLKn to RC32K.)
Write 0x0 to LnSEL bit
Write 0x0 to LOSCMD[1:0] bits
Write 0x1 to LOSCB bit
Handle appropriately for the application.
Clear LOSCMD bits in first to distinguish backup state or stabilization state when a system reset occurs.

Figure 6-10 Backup procedure

The clock backup interrupt occurs at stop detection , even if the LSCLK1 is used without backup function in the XT32K/EXT32K mode. Then newer interrupt does not occur until the LOSCB bit is cleared.

6.3.1.4 Oscillation Stop Detection and Back-up Function

When the XT32K stop detected after LSCLK0/LSCLK1 is set to XT32K, the LSCLK0/LSCLK1 is switched to the RC32K. It is clock backup function.
The stop detection time is approx. 4ms (typ.) when not in the HALT-D mode, and approx. 5ms (typ.) in the HALT-D mode.
Figure 6-11 and 6-12 show the operation of XT32K stop detection under each condition.
In case of LSCLK0=XT32K, LSCLK1=XT32K/OFF :
When the XT32K stops, the LSCLK0 also stops, and the high-speed clock frequency is not guaranteed. When the stop is detected for a certain peeriod of time, the RC32K is turned on and the stop detection process is performed. The LSCLK0 is supplied again with RC32K and recovers the high-speed clock frequency.
If the system clock is low-speed, the SYSCLK stops too; Figure 6-11(1).
If the system clock is high-speed, the SYSCLK does not stop; Figure 6-11(2)



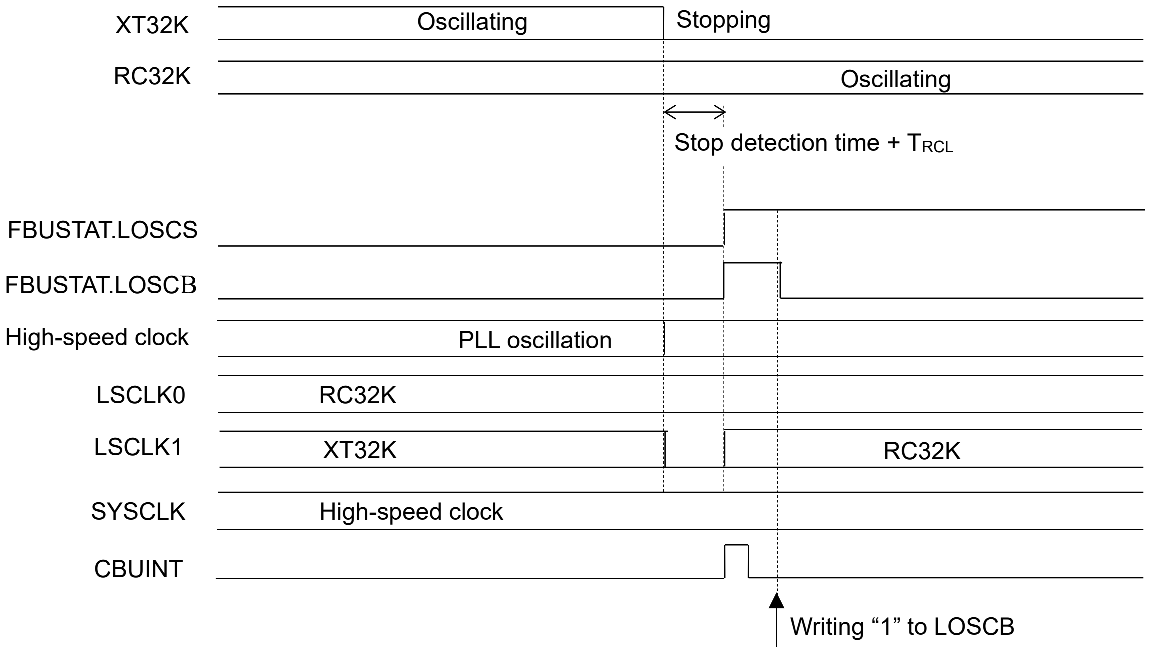
(1) In case of SYSCLK=LSCLK0=XT32K

(2) In case of SYSCLK=HSCLK, LSCLK0=XT32K

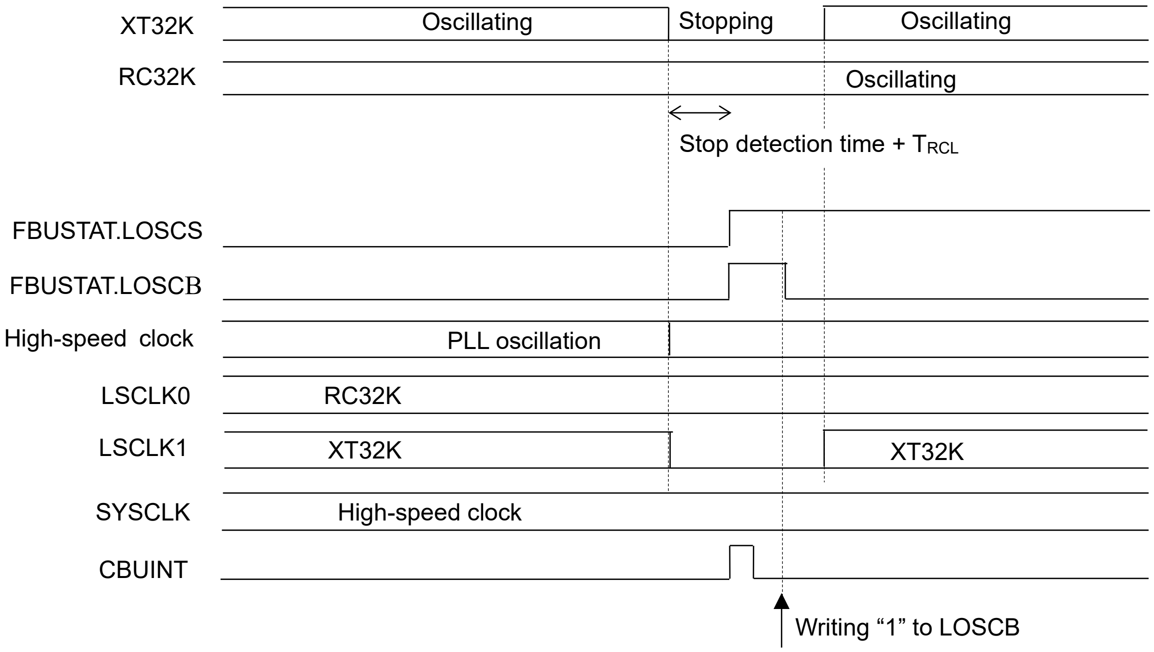
▨ : The PLL frequency is not guaranteed between from stopping XT32K to locked PLL based RC32K.

Figure 6-11 Stop detection for the crystal oscillating: 1 (LSCLK0=XT32K, LSCLK1=XT32K)

In case of LSCLK1=XT32K, LSCLK0=RC32K :
The LSCLK1 stops when XT32K stops. The SYSCLK/High-speed clock do not stop. When the stopping continues in a certain time, the stop detection occurs.
If enabled backup, LSCLK1 becomes to RC32K; Figure 6-12-1(1).
If disabled backup, LSCLK1 is kept XT32K even if stop detection occurs; Figure 6-12-1(2). LSCLK1 stops when XT32K stops, also LSCK1 oscillates when XT32K oscillates again. The clock backup interrupt occurs when the stop detection occurs.

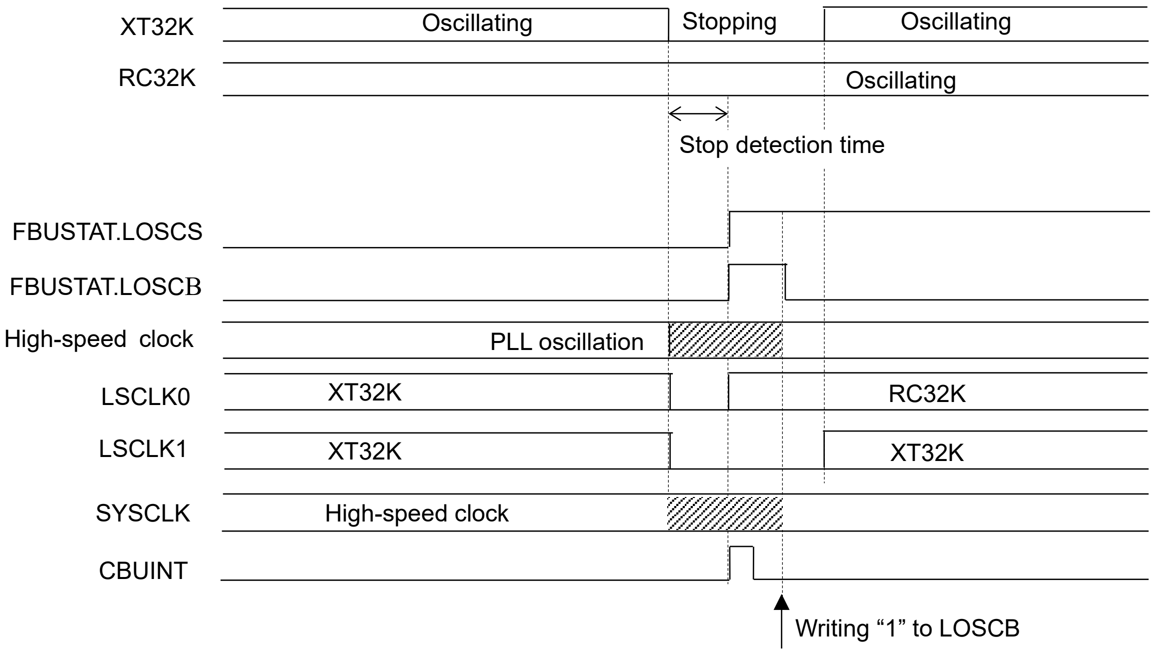


(1) Enabled backup; LS1BU=1



(2) Disabled backup; LS1BU=0

Figure 6-12-1 Stop detection for the crystal oscillating: 2
(SYSCLK=HSCLK, LSCLK0=RC32K, LSCLK1=XT32K)



▨ : The PLL frequency is not guaranteed between from stopping XT32K to locked PLL based RC32K.

Figure 6-12-2 Stop detection for the crystal oscillating: 3
(SYSCLK=HSCLK, LSCLK0=XT32K, LSCLK1=XT32K)

6.3.1.5 Status of XT32K/EXT32K Mode

Table 6-7 shows relationship of each setting and status in the XT32K mode. As for EXT32K mode, change the reading "LOSCMD0" to "LOSCMD1", "XT32K" to "EXT32K".

Table 6-7 Status of LSCLK (XT32K)

LOSCMD0	L1CEN	LS1BU	L1SEL	L0SEL	LOSCB	LOSCS	LSCLK1	LSCLK0
1	1	1	0	0	0	0	RC32K(CW)	RC32K(CW)
1	1	1	0	0	0	1	RC32K(SW)	RC32K(SW)
1	1	1	0	0	1	1	RC32K	RC32K
1	1	1	0	1	0	0	RC32K	XT32K
1	1	1	0	1	1	1	RC32K	RC32K(BU)
1	1	1	1	0	0	0	XT32K	RC32K
1	1	1	1	0	1	1	RC32K(BU)	RC32K
1	1	1	1	1	0	0	XT32K	XT32K
1	1	1	1	1	1	1	RC32K(BU)	RC32K(BU)
1	1	0	0	0	0	0	off	RC32K(CW)
1	1	0	0	0	0	1	off	RC32K(SW)
1	1	0	0	0	1	1	off	RC32K
1	1	0	1	0	0	0	XT32K	RC32K
1	1	0	1	0	1	1	XT32K(SS)	RC32K
1	1	0	1	1	0	0	XT32K	XT32K
1	1	0	1	1	1	1	XT32K(SS)	RC32K(BU)

Character in () means the following:
BU=Backup state, SW=Waiting stability of XT32K, CW=Completion of waiting stability of XT32K, SS=Detected stop
* : LOSCB state is a state before clearing, so it does not show state in the backup procedure.

6.3.2 High-speed Clock

The PLL oscillation circuit generates the high-speed clock; HSOCLK by multiplying the LSCLK0.
The PLL frequency is configured to 24MHz/16MHz/1MHz by the code option.
The high-speed output clock divided the HSOCLK ; HCKO is output from LSI pins. See Table 1-3 for pin assignment.

6.3.2.1 PLL Oscillation Circuit

The PLL oscillation circuit generates the high-speed clock; HSOCLK by multiplying the LSCLK0. The multiplying by 732.5 is 24MHz of PLL frequency, and the multiplying by 488.5 is 16MHz of PLL frequency, and the multiplying by 30.5 is 16MHz of PLL frequency.
After high-speed clock oscillation is enabled , the high-speed clocks; HSCLK/HSCOCLK/HCKO is output by continuing count operation until the PLL oscillation clock is stabilized.

When set “1” to the FHUT0 bit of FHWUPT register, the clock supply is started approximately 30 μs after the high-speed clock oscillation is enabled. The clock frequency reaches to the target approximately 1 ms after the high-speed clock oscillation is enabled. Although the frequency within the 1 ms is not guaranteed, it can be used for the system clock.When set “0” to the FHUT0 bit, the clock supply is started approximately 1ms after it is enabled.

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

Figure 6-13 shows the PLL oscillation circuit configuration.

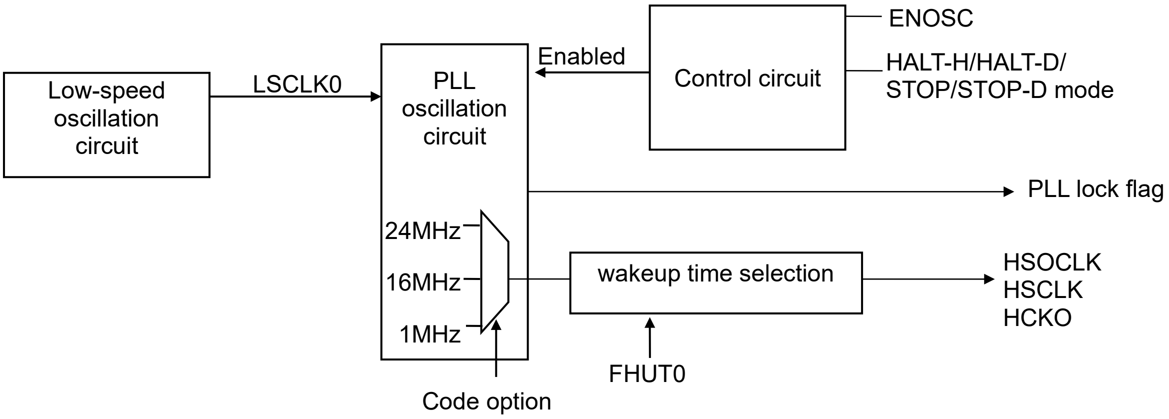


Figure 6-13 PLL Oscillation Circuit Configuration

Figure 6-14 show the high-speed clock operation waveforms at startup PLL circuit, in the standby mode.
See Table 4-6 for the time for wakeup from the HALT-H mode. See Chapter 4 “Power Management” for details of the STOP/STOP-D mode.

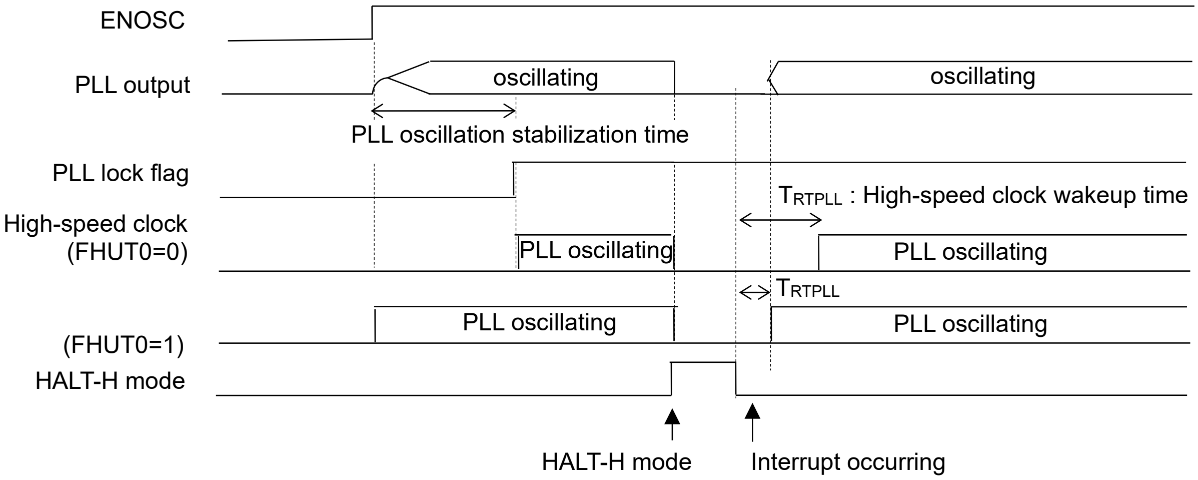


Figure 6-14-1 High-speed clock operation at startup PLL circuit and in HALT-H mode

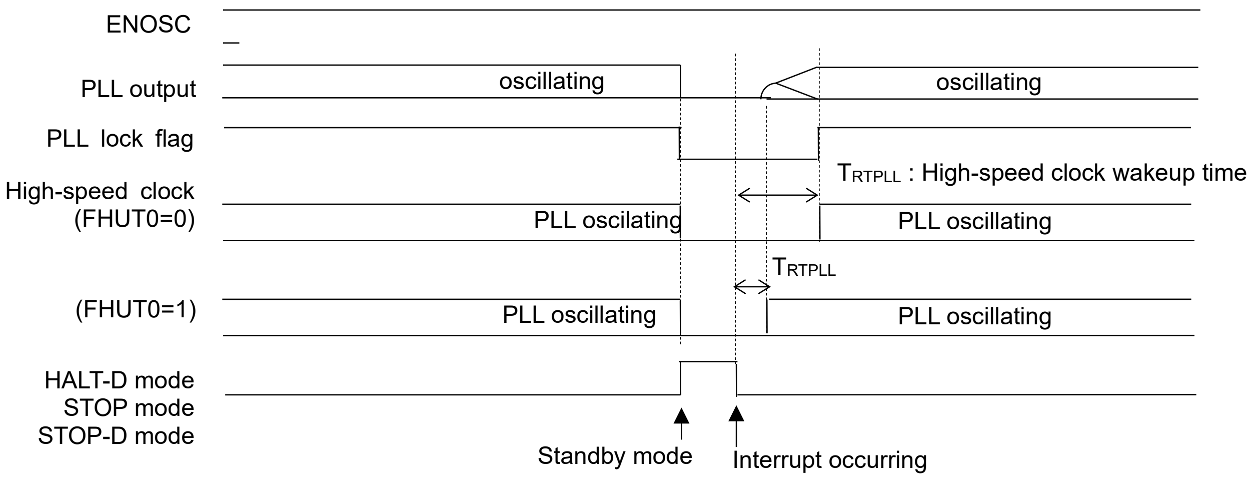


Figure 6-14-2 High-speed clock operation in the HALT-D/STOP/STOP-D mode

6.3.2.2 High-speed Clock Control

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

(1) Set dividing HSCLK by FHCKMODL.
(2) Set with or without waiting stabilization by FHWUPT.
(3) Set enabled PLL oscillation by ENOSC=1. SELSCLK bit can be set to “1” at once.
The system clock stops until the high-speed clock oscillates when SELSCLK=1.

In the case of without waiting stabilization; FHUT0=1: Although the high-speed clock frequency is not guaranteed just after setting ENOSC=1, it can be used for the system clock and peripheral clocks. If specific frequency is required, wait stabilization time; LPLL=1.
In the case of with waiting stabilization; FHUT0=0: The clock is supplied after LPLL bit becomes to “1”.

The control flow when the high-speed clock before stabilization with LSCLK0=RC32K, is used for the initialization processing and the high-speed clock after stabilization with LSCLK0=XT32K is used for the main processing as the system clock is shown below.

(1) Set dividing HSCLK by FHCKMODL.
(2) Set without waiting stabilization by FHWUPT. Because the high-speed clock output pauses when switching of (5) if FHUT0=0.
(3) Set enabled PLL oscillation by ENOSC=1 and changing system clock by SELSCLK=1 at once.
(4) Set enabled XT32K oscillation by FLMOD, and then execute a procedure to wait crystal oscillation stabilization and wait LOSCS=0 .Execute initialization for user program during waiting LOSCS=0.
(5) Set “1” to the L0SEL bit of FLMOD, so that LSCLK0 is changed from RC32K to XT32K.
(6) To wait the high-speed clock frequency stabilization, wait for LPLL=1 after 156μs elapse from (5).

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

[Note]

- **When the XT32K is used for LSCLK0, the high-speed clock may become an unintended frequency due to external factors such as noise, and the MCU may operates abnormally. Please evaluate enough the apparatus/system which implemented this product.**

6.3.2.3 HALT-H mode

The high-speed clock is stopped when entry to HALT-H mode, and the clock is automatically turned on when wake-up.
The PLL frequency stability time is due to temperature difference between the HALT-H mode enter and wake-up.
It takes up to 2ms for the high-speed clock to be supplied if FTUT0=0.

The PLL frequency accuracy is about ±5% in 300μs when the temperature difference is as follows in each mode.

PLL mode	temperature difference range [°C]	Other conditions
24M mode	-22 ~ +16	Code option VLMD=0
16M mode	-17 ~ +13	Code option VLMD=0
1M mode	-13 ~ + 10	-

If no waiting locked PLL; FHUT0=1, the PLL frequency may exceed the operating range depending on the temperature difference. Therefore, control with FHRDWN bit to avoid it. See “6.2.5 High-Speed Clock Wake-up Time Setting Register” for the FHRDWN bit, and “30.2.3 Code Option 1” for code option VLMD.

6.3.3 Internal 1kHz clock (RC1K)

The internal 1kHz clock is oscillation that frequency is 1.024kHz typ. It is supplied to the WDT, 16-bit timer, functional timer.
In the case of the WDT operation is enabled by setting code option; WDTMD=1 :
RC1K is oscillating after system reset is released. It turns off at entry to STOP/STOP-D mode, and then it turns on at wakeup from standby. In the HALT-D mode, it stops only when WDTPWMD1=0 is selected by setting code option. The clock wakeup time is approx. 2ms.
In the case of WDTMD = 0 :
RC1K has stopping after system reset is released. RC1K oscillation is enable by the ENRC1K bit of FCONW register becomes "1", and the clock is supplied after approx. 2.5[ms].

The RC1K oscillation does not stop in the standby mode if ENRC1K=1. However a suppling WDTCLK depends on code options. Also, a suppling RC1K to 16-bit timer/functional timer stops in the STOP/STOP-D mode.
Set ENRC1K=1 before the standby entry if the RC1K oscillating will be required quickly after wakeup from stand-by.
See Chapter 30 "Code Option" for how to set code options.

Figure 6-15 shows configuration of internal RC1K oscillation.
Figure 6-16 shows an operation of RC1K and WDTCLK at startup and wakeup from STOP mode.

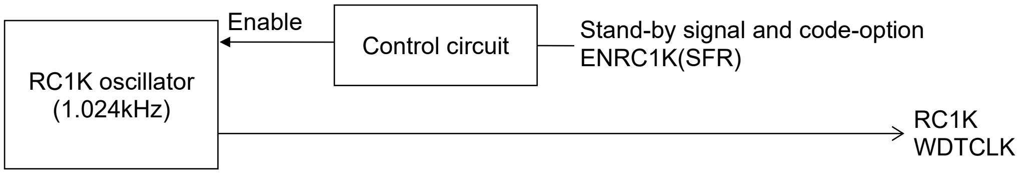


Figure 6-15 Configuration of internal 1kHz oscillator

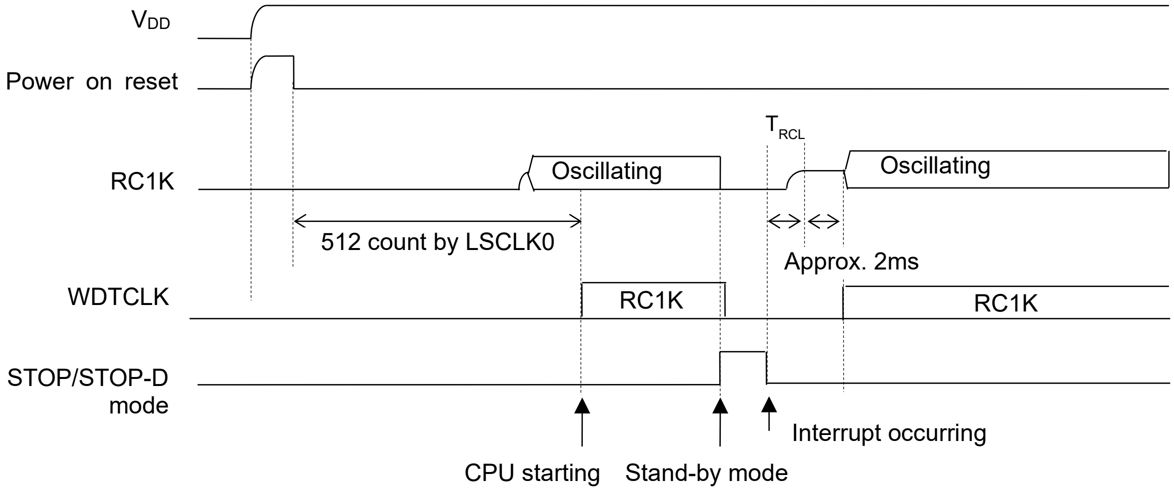


Figure 6-16-1 Startup and Wakeup from STOP mode of RC1K oscillator (ENRC1K=0)

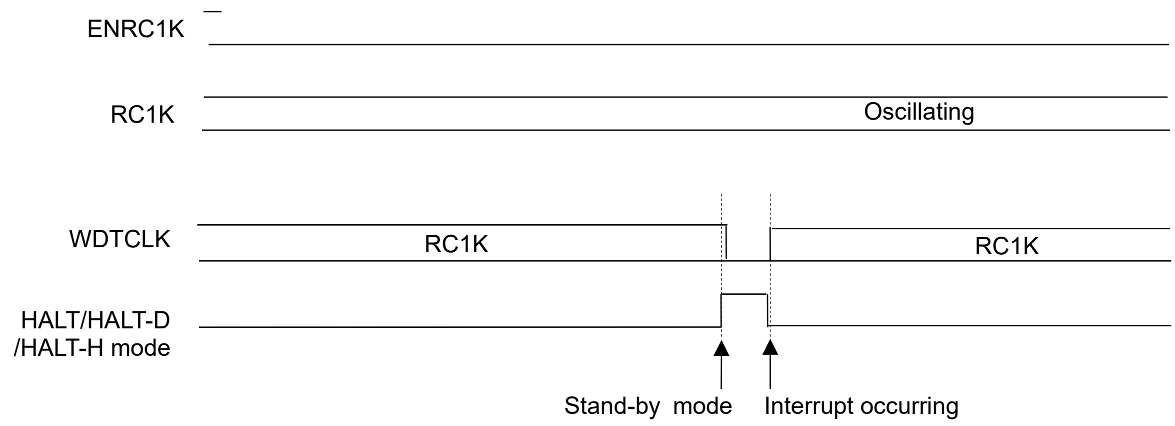


Figure 6-16-2 HALT/HALT-H/HALT-D mode (ENRC1K=0)

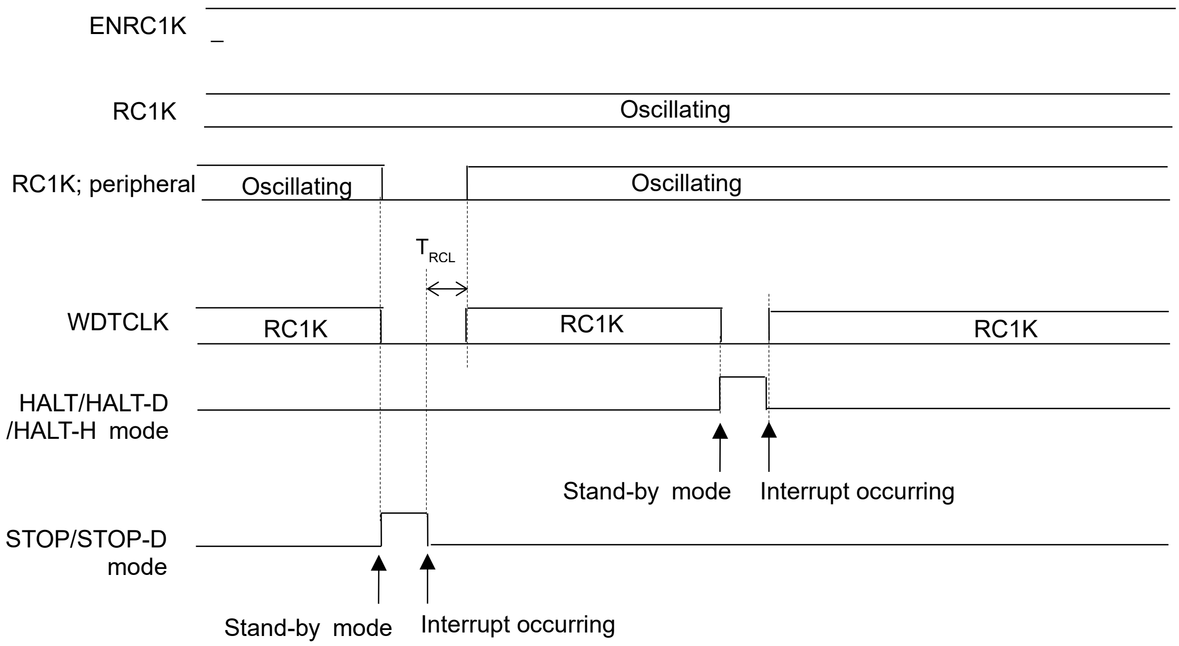


Figure 6-16-3 Stand-by mode (ENRC1K=1)

6.3.4 System Clock

A system clock is LSCLK0 with RC32K after power-on-reset/pin reset. The source clock of LSCLK0 is configured L0SEL after another system reset.
A system clock can be dynamically switched the high-speed system clock or LSCLK0 by SELSCLK bit. The high-speed system clock can be dynamically changed dividing value by SYSC bits.
There are 2 types as system clock; CPUCLK supplied to CPU and SYSCLK supplied to peripherals.
The CPUCLK stops in all stand-by mode. The SYSCLK normally supplies in the HALT/HALT-H mode, however it does not supply to some peripherals in the HALT-D mode. See Chapter 4 “Power Management” for detail.

[Note]

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

6.3.5 Interrupt

The clock back-up control circuit has a interrupt.
The clock buck-up interrupt is generated when the crystal oscillating or external clock input stop in the XT32K/EXT32K mode.
At this interrupt generated, LOSCB becomes to “1”. Write “1” to LOSCB bit to clear this flag.
New interrupt request does not occur until the LOSCB bit is cleared.

To wait crystal oscillation stability, use 16-bit timer X interrupt.

6.3.6 Clock Back-up Test

The clock back-up test function can make purposely the condition that stops the low-speed crystal oscillation. Figure 6-17 shows the test procedure.

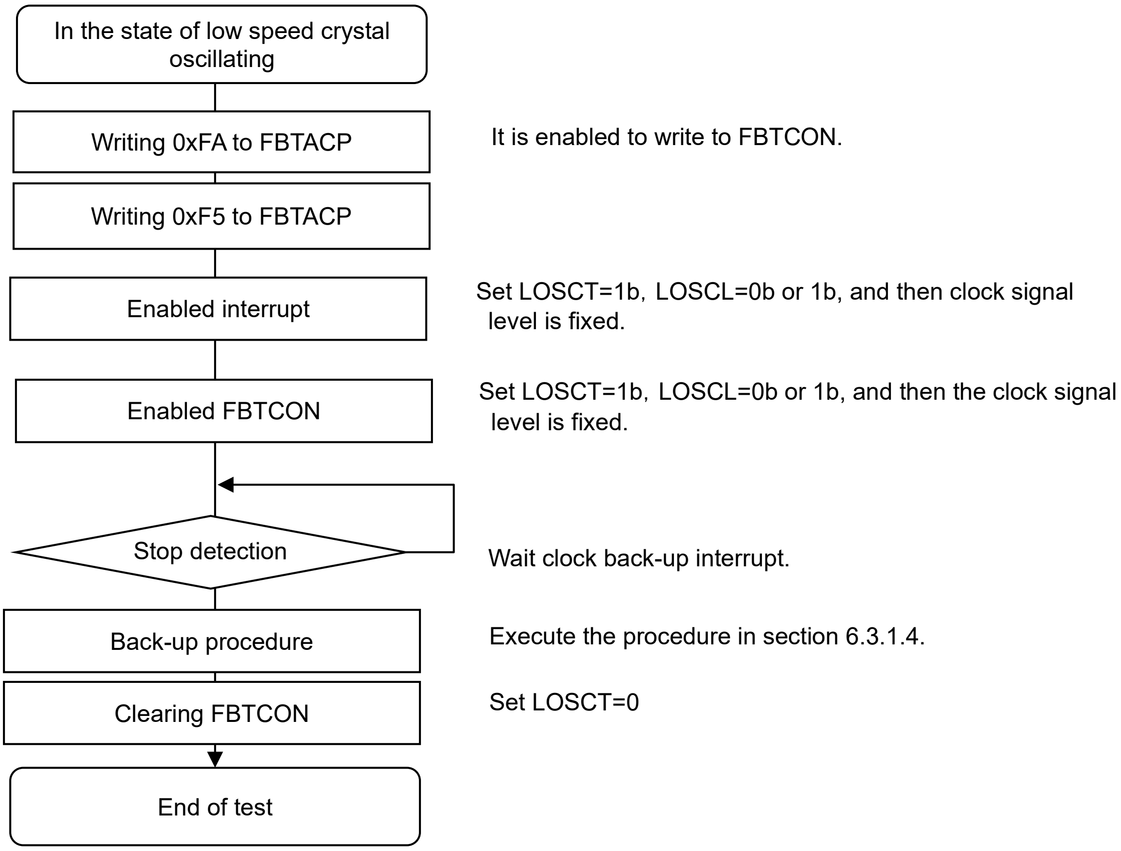

In the state of low speed crystal oscillating
Writing 0xFA to FBTACP
It is enabled to write to FBTCON.
Writing 0xF5 to FBTACP
Enabled interrupt
Set LOSCT=1b, LOSCL=0b or 1b, and then clock signal level is fixed.
Enabled FBTCON
Set LOSCT=1b, LOSCL=0b or 1b, and then the clock signal level is fixed.
Stop detection
Wait clock back-up interrupt.
Back-up procedure
Execute the procedure in section 6.3.1.4.
Clearing FBTCON
Set LOSCT=0
End of test

Figure 6-17 Clock Back-up Test Procedure

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.

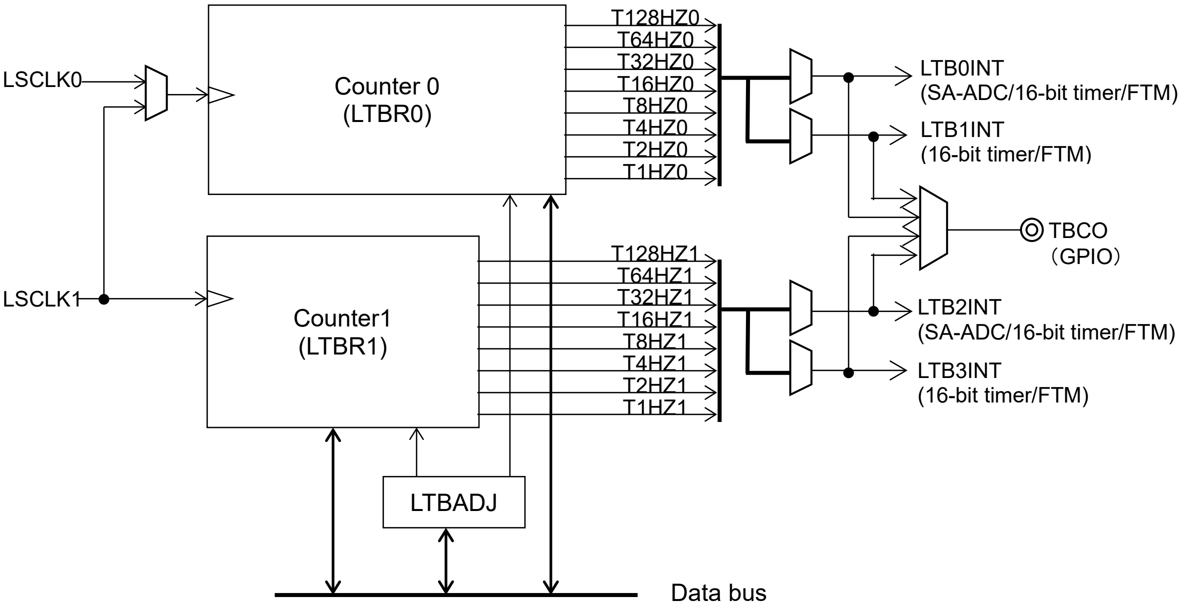
- Generate periodical interrupt requests
- Output periodical pulse signals to the general ports

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 (LSCLK0 or LSCLK1)
- Four interrupt requests can be chosen among eight periodical interrupt requests
- One of a pulse configured as interrupt requests can be output from general ports
- An interrupt request can be used for a trigger event source of the Successive Approximation type A-D Converter, 16-bit timer and functional timer.
- Allows to adjust in a range approx.-488ppm to +488ppm with the resolution approx.0.48ppm.

7.1.2 Configuration

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



LTBADJ	:	Low speed time base counter frequency adjustment register
T1HZ0～T128HZ0	:	Time base counter 0 output signal
T1HZ1～T128HZ1	:	Time base counter 1 output signal
LTB3INT	:	Low speed time base counter 3 interrupt request
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
TBCO	:	GPIO output as LTBC clock

Figure 7-1 Configuration of Low Speed Time Base Counter

7.1.3 List of Pins

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

Signal name	I/O	Function
TBCO	O	The low speed time base counter output signal

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	ML62Q2500 Group		
					32 pin product	40 pin product	48 pin product
TBCO	P17	7th function	P1MOD3	0110_XXXX*1	-	●	●
	P27	7th function	P2MOD7	0110_XXXX*1	●	●	●
	P73	7th function	P7MOD3	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			
0xF3A0	Low-speed Time Base Counter register	LTBR0	LTBR01	R/W	8/16	0x00
0xF3A1		LTBR1		R/W	8	0x00
0xF3A2	Low-speed Time Base Counter Control register	LTBCON0	LTBCON	R/W	8/16	0x03
0xF3A3		LTBCON1		R/W	8	0x02
0xF3A4	Reserved	-	-	-	-	-
0xF3A5	Reserved	-	-	-	-	-
0xF3A6	Low-speed Time Base Counter Frequency Adjustment register	LTBADJL	LTBADJ	R/W	8/16	0x00
0xF3A7		LTBADJH		R/W	8	0x00
0xF3A8	Low-speed Time Base Counter Interrupt selection register	LTBINTL	LTBINT	R/W	8/16	0x60
0xF3A9		LTBINTH		R/W	8	0x71

7.2.2 Low Speed Time Base Counter Register (LTBR01)

This is a SFR to read the value of the low speed time base counter.
Writing any value to the LTBR0, the all bits of T128HZ0 to T1HZ0 are initialized to "0". Writing any value to the LTBR1, the all bits of T128HZ1 to T1HZ1 are initialized to "0". Writing any value to the LTBR01, the both are initialized to “0”.

Address:	0xF3A0(LTBR0/LTBR01), 0xF3A1(LTBR1)
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	LTBR01															
Byte	LTBR1								LTBR0							
Bit	T1HZ1	T2HZ1	T4HZ1	T8HZ1	T16HZ1	T32HZ1	T64HZ1	T128HZ1	T1HZ0	T2HZ0	T4HZ0	T8HZ0	T16HZ0	T32HZ0	T64HZ0	T128HZ0
R/W	R/W	R/W	R/W	R/W	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

T128HZ0 to T1HZ0 / T128HZ1 to T1HZ1 signals have "0" level in the first half cycle and "1" level in the second half cycle. For example, T1HZ0 signal gets reset to "0" by writing any data to LTBR0 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 7-4 "Low speed time base counter interrupt timing and reset timing of reset by writing to LTBR0" for details of the T128HZ0 to T1HZ0 / T128HZ1 to T1HZ1 waveform.

[Note]

- **A time base counter interrupt may occur depending on the timing to write to the LTBR01. See the program example for initializing described in Section 7.3.1 "Operation of the Low-speed Time Base Counter".**
- **Read the LTBR01 register twice to verify the data to prevent reading uncertain data while counting-up.**

7.2.3 Low Speed Time Base Counter Control Register (LTBCON)

This is a SFR to control the function of the time base counter.

Address:	0xF3A2(LTBCON0/LTBCON01), 0xF3A3(LTBCON1)
Access:	R/W
Access size:	8/16 bit
Initial value:	0x0203

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	LTBCON															
Byte	LTBCON1								LTBCON0							
Bit	TBOSEL1	TBOSEL0	-	-	TB1ADJEN	TB0ADJEN	TB1CK	TB0CK	-	-	-	-	-	-	TB1RUN	TB0RUN
R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1

Bit No.	Bit symbol name	Description
15,14	TBOSEL1 to 0	This bit is used to choose a signal as TBCO output. 00: LTB0INT (Initial value) 01: LTB1INT 10: LTB2INT 11: LTB3INT
13 to 12	-	Reserved bits
11	TB1ADJEN	This bit is used to enable or disable adjustment of LTBR1. 0: Disabled (Initial value) 1: Enabled
10	TB0ADJEN	This bit is used to enable or disable adjustment of LTBR0. 0: Disabled (Initial value) 1: Enabled
9	TB1CK	This bit is not configurable. A clock of LTBR1 is LSCLK1. 1: LSCLK1 (Fixed)
8	TB0CK	This bit is used to choose a clock of LTBR0. 0: LSCLK0 (Initial value) 1: LSCLK1
7 to 2	-	Reserved bits
1	TB1RUN	This bit is used to control run/stop counter LTBR1. 0: Stop 1: Run (Initial value)
0	TB0RUN	This bit is used to control run/stop counter LTBR0. 0: Stop 1: Run (Initial value)

[Note]

- **Stop counter LTBR0 (i.e. set 0 to TB0RUN bit), before TB0CK bit is configured.**

7.2.4 Low Speed Time base counter frequency adjustment register (LTBADJ)

This is a SFR to set adjustment value for the frequency of time bask clock.

Address:	0xF3A6(LTBADJL/LTBADJ), 0xF3A7(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	-	-	-	-	-	LADJ10	LADJ9	LADJ8	LADJ7	LADJ6	LADJ5	LADJ4	LADJ3	LADJ2	LADJ1	LADJ0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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 0	LADJ10 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.

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

This is a SFR to specify the low-speed time base clock to be used as an interrupt signal.

Address:	0xF3A8(LTBINTL/LTBINT), 0xF3A9(LTBINTH)
Access:	R/W
Access size:	8/16 bit
Initial value:	0x7160

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	LTBINT															
Byte	LTBINTH								LTBINTL							
Bit	-	LTI3S2	LTI3S1	LTI3S0	-	LTI2S2	LTI2S1	LTI2S0	-	LTI1S2	LTI1S1	LTI1S0	-	LTI0S2	LTI0S1	LTI0S0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Initial value	0	1	1	1	0	0	0	1	0	1	1	0	0	0	0	0

Bit No.	Bit symbol name	Description
15	-	Reserved bit
14 to 12	LTI3S2 to LTI3S0	These bits are used to choose the signal to be assigned to the time base counter interrupt 3 (LTB3INT). 000: T128HZ1 / 100: T8HZ1 001: T64HZ1 / 101: T4HZ1 010: T32HZ1 / 110: T2HZ1 011: T16HZ1 / 111: T1HZ1 (Initial value)
11	-	Reserved bit
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: T128HZ1 / 100: T8HZ1 001: T64HZ1 (Initial value) / 101: T4HZ1 010: T32HZ1 / 110: T2HZ1 011: T16HZ1 / 111: T1HZ1
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: T128HZ0 / 100: T8HZ0 001: T64HZ0 / 101: T4HZ0 010: T32HZ0 / 110: T2HZ0 (Initial value) 011: T16HZ0 / 111: T1HZ0
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: T128HZ0 (Initial value) / 100: T8HZ0 001: T64HZ0 / 101: T4HZ0 010: T32HZ0 / 110: T2HZ0 011: T16HZ0 / 111: T1HZ0

[Note]

- **A time base counter interrupt may occur depending on a write timing to the LTBINT. 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 T128HZ0 to T1HZ0 / T128HZ1 to T1HZ1 signals. Two factors can be chosen from T128HZ0 to T1HZ0 signals, and two factors can be chosen from T128HZ1 to T1HZ1 signals. There are to generate periodical low-speed time base counter interrupt requests.
Values of T128HZ0 to T1HZ0 / T128HZ1 to T1HZ1 signals can be read from the LTBR01 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 (LTBnINT) 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 3)

Figure 7-2 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
ELTBC3 = 0;       // Disable LTBC3 interrupt
LTBINT = 0x0741;// Change assignment of interrupt signal
__asm("NOP");     // Waiting time
QLTBC0 = 0;       // Clear QLTBC0
QLTBC1 = 0;       // Clear QLTBC1
QLTBC2 = 0;       // Clear QLTBC2
QLTBC3 = 0;       // Clear QLTBC3
ELTBC0 = 1;       // Enable LTBC0 interrupt
ELTBC1 = 1;       // Enable LTBC1 interrupt
ELTBC2 = 1;       // Enable LTBC2 interrupt
ELTBC3 = 1;       // Enable LTBC3 interrupt
```

Figure 7-2 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 3) to become "1" after changing the LTBINT register . Therefore, place two NOP instruction after changing the LTBINT register.

When writing arbitrary data to the LTBR0 register, T128HZ0 to T1HZ0 signals of the LTBR01 register are all initialized to "0". When writing arbitrary data to the LTBR1 register, T128HZ1 to T1HZ1 signals of the LTBR01 register are all initialized to "0". When writing arbitrary data to the LTBR01 register, all bits of LTBR01 register are 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 3)

Figure 7-3 shows a sample program for initializing the LTBR0 register.

```
__DI();           // Disable interrupt (MIE=0)
LTBR0 = 0x00;     // Reset LTBR0
__asm("NOP");     // Waiting time
__asm("NOP");     // Waiting time
QLTBC0 = 0;       // Clear QLTBC0
QLTBC1 = 0;       // Clear QLTBC1
__EI();           // Enable interrupt (MIE=1)
```

Figure 7-3 Sample Program for Initializing LTBR0 Register

It takes one cycle of the system clock for QLTBCn to become "1" from writing to the LTBR register. Therefore, place two NOP instruction after writing to the LTBR01 register.

Figure 7-4 shows the low-speed time base counter interrupt request generation timing when choosing T128HZ0, T16HZ0, and T2HZ0 as interrupt factors in the LTBINT register, and shows the reset timing by writing to LTBR0.

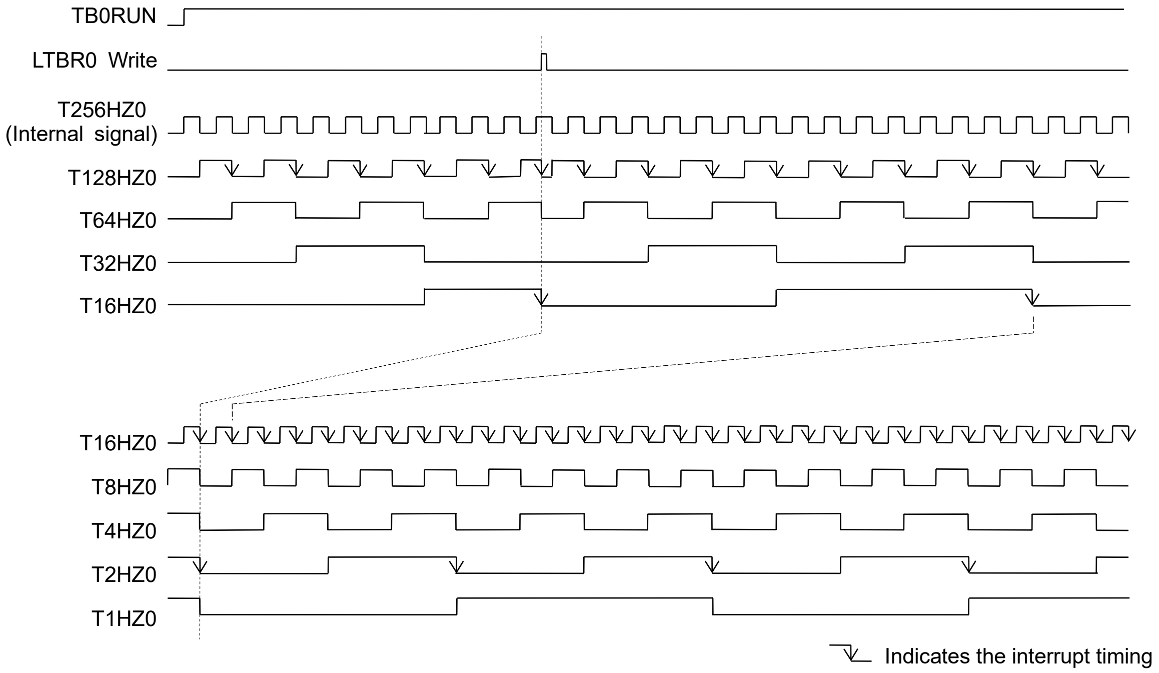


Figure 7-4 Low-speed Time Base Counter Interrupt Timing and Reset Timing by Writing to LTBR0 Register

[Note]

- **After writing to the LTBR01 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.**

7.3.2 Low-speed Time Base Counter Frequency Adjustment Function

For T128HZ0 to T1HZ0 / T128HZ1 and T1HZ1 of the low-speed time base counter, the frequency can be adjusted using the low-speed time base counter frequency adjustment register (LTBADJ). Measure the signal output from the TBCO 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. 0.477 ppm

The following is available to confirm the adjusted frequency:

Frequency adjustment mode	Description
Normal frequency adjustment mode	This is used to confirm that 64 seconds includes exactly 128 cycles (or 64 cycles) of T2HZ0/T2HZ1 (or T1HZ0/T1HZ1),which is output form pin as TBCO under operating with actual adjusted low-speed clock.

Table 7-2 shows the frequency adjustment value set in the LTBADJ and adjustment ratio.

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

LADJ10-0											Hex.	Frequency adjustment ratio (ppm)
0	1	1	1	1	1	1	1	1	1	1	3FFH	+487.80
0	1	1	1	1	1	1	1	1	1	0	3FEH	+487.33
:	:	:	:	:	:	:	:	:	:	:	:	:
0	0	0	0	0	0	0	0	0	1	1	003H	+1.43
0	0	0	0	0	0	0	0	0	1	0	002H	+0.95
0	0	0	0	0	0	0	0	0	0	1	001H	+0.48
0	0	0	0	0	0	0	0	0	0	0	000H	0
1	1	1	1	1	1	1	1	1	1	1	7FFH	-0.48
1	1	1	1	1	1	1	1	1	1	0	7FEH	-0.95
:	:	:	:	:	:	:	:	:	:	:	:	:
1	0	0	0	0	0	0	0	0	0	1	401H	-487.80
1	0	0	0	0	0	0	0	0	0	0	400H	-488.28

The correction values (LADJ10 to LADJ0) set in the LTBADJ register can be calculated using the following formula.

Correction value = Frequency adjustment ratio x 2097152 (decimal)
= Frequency adjustment ratio x 200000h (hexadecimal)

Example 1: When adjusting +15.0 ppm (when the clock loses)
Correction value = +15.0 ppm x 2097152 (decimal)
= +15.0 x 10^{-6} x 2097152
= +31.45728 (decimal)
≈ 1Fh (hexadecimal)

Example 2: When adjusting -25.5 ppm (when the clock gains)
Correction value = -25.5 ppm x 2097152 (decimal)
= -25.5 x 10^{-6} x 2097152
= -53.477376 (decimal)
≈ 7CBh (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.**

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

The timer X is shared function for stability controlling of crystal oscillation. When un-used crystal oscillation, it is used as normal 16-bit timer.
See Chapter 6 "Clock generation circuit" for function for stability controlling of crystal oscillation.

Table 8-1 shows the number of channels.

Table 8-1 Number of 16-bit Timer channels

Channel no.	ML62Q2500 group
0	●
1	●
2	●
3	●
4	●
5	-
6	-
7	-
X	●

8.1.1 Features

Operation mode	Description
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.

- Selectable counter clock from various sources (divided by 1 to 8 of LSCLK0, LSCLK1, HSCLK, HTBCLK0, HTBCLK1, RC1K, external clock, LTBC interrupt, functional timer triggers)
- A timer interrupt request is generated when the value of the timer counter register value coincides with that of the 16-bit timer n data register
- 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 level of the port can be chosen by a register.

8.1.2 Configuration

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

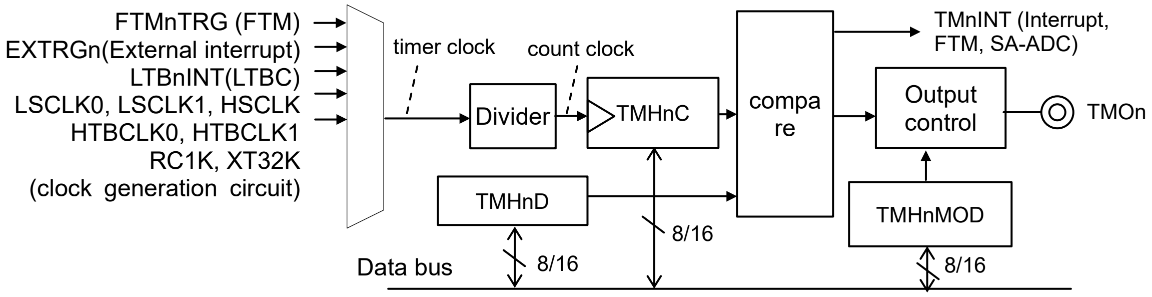


Figure 8-1 Configuration of the timer

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

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
EXIn	I	External clock (used as EXTRGn). The max input frequency is 3MHz.
TMO2	O	16-bit timer channel 2 output
TMO3	O	16-bit timer channel 3 output
TMO4	O	16-bit timer channel 4 output
TMOX	O	16-bit timer channel X output

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	ML62Q2500 group		
					32 pin product	40 pin product	48 pin product
EXI0	Ports assigned for external interrupt. See Chapter 18. External interrupt.				●	●	●
EXI1					●	●	●
EXI2					●	●	●
EXI3					●	●	●
EXI4					●	●	●
EXI5					●	●	●
EXI6					●	●	●
EXI7					●	●	●
TMO2	P04	6th Func.	P0MOD4	0101_XXXX*1	●	●	●
	P10	6th Func.	P1MOD0	0101_XXXX*1	●	●	●
	P60	6th Func.	P6MOD0	0101_XXXX*1	-	-	●
	P70	6th Func.	P7MOD0	0101_XXXX*1	●	●	●
TMO3	P05	6th Func.	P0MOD5	0101_XXXX*1	●	●	●
	P11	6th Func.	P1MOD1	0101_XXXX*1	●	●	●
	P61	6th Func.	P6MOD1	0101_XXXX*1	-	-	●
	P71	6th Func.	P7MOD1	0101_XXXX*1	●	●	●
TMO4	P06	6th Func.	P0MOD6	0101_XXXX*1	●	●	●
	P12	6th Func.	P1MOD2	0101_XXXX*1	●	●	●
	P62	6th Func.	P6MOD2	0101_XXXX*1	-	-	●
	P72	6th Func.	P7MOD2	0101_XXXX*1	●	●	●
TMOX	P13	6th Func.	P1MOD3	0101_XXXX*1	●	●	●
	P73	6th Func.	P7MOD3	0101_XXXX*1	●	●	●

●: Available to use -: Unavailable

*1 : "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

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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF300	16-bit timer 0 data register	TMH0DL	TMH0D	R/W	8/16	0xFF
0xF301		TMH0DH		R/W	8	0xFF
0xF302	16-bit timer 0 counter register	TMH0CL	TMH0C	R/W	8/16	0x00
0xF303		TMH0CH		R/W	8	0x00
0xF304	16-bit timer 0 mode register	TMH0MODL	TMH0MOD	R/W	8/16	0x00
0xF305		TMH0MODH		R/W	8	0x00
0xF306	Reserved	-	-	-	-	-
0xF307						
0xF308	16-bit timer 1 data register	TMH1DL	TMH1D	R/W	8/16	0xFF
0xF309		TMH1DH		R/W	8	0xFF
0xF30A	16-bit timer 1 counter register	TMH1CL	TMH1C	R/W	8/16	0x00
0xF30B		TMH1CH		R/W	8	0x00
0xF30C	16-bit timer 1 mode register	TMH1MODL	TMH1MOD	R/W	8/16	0x00
0xF30D		TMH1MODH		R/W	8	0x00
0xF30E	Reserved	-	-	-	-	-
0xF30F						
0xF310	16-bit timer 2 data register	TMH2DL	TMH2D	R/W	8/16	0xFF
0xF311		TMH2DH		R/W	8	0xFF
0xF312	16-bit timer 2 counter register	TMH2CL	TMH2C	R/W	8/16	0x00
0xF313		TMH2CH		R/W	8	0x00
0xF314	16-bit timer 2 mode register	TMH2MODL	TMH2MOD	R/W	8/16	0x00
0xF315		TMH2MODH		R/W	8	0x00
0xF316	Reserved	-	-	-	-	-
0xF317						
0xF318	16-bit timer 3 data register	TMH3DL	TMH3D	R/W	8/16	0xFF
0xF319		TMH3DH		R/W	8	0xFF
0xF31A	16-bit timer 3 counter register	TMH3CL	TMH3C	R/W	8/16	0x00
0xF31B		TMH3CH		R/W	8	0x00
0xF31C	16-bit timer 3 mode register	TMH3MODL	TMH3MOD	R/W	8/16	0x00
0xF31D		TMH3MODH		R/W	8	0x00
0xF31E	Reserved	-	-	-	-	-
0xF31F						
0xF320	16-bit timer 4 data register	TMH4DL	TMH4D	R/W	8/16	0xFF
0xF321		TMH4DH		R/W	8	0xFF
0xF322	16-bit timer 4 counter register	TMH4CL	TMH4C	R/W	8/16	0x00
0xF323		TMH4CH		R/W	8	0x00
0xF324	16-bit timer 4 mode register	TMH4MODL	TMH4MOD	R/W	8/16	0x00
0xF325		TMH4MODH		R/W	8	0x00
0xF326~ 0xF33F	Reserved	-	-	-	-	-
0xF340	16-bit timer start register	TMHSTRL	TMHSTR	W	8/16	0x00
0xF341		TMHSTRH		W	8	0x00
0xF342	16-bit timer stop register	TMHSTPL	TMHSTP	W	8/16	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF343		TMHSTPH		W	8	0x00
0xF344	16-bit timer status register	TMHSTATL	TMHSTAT	R	8/16	0x00
0xF345		TMHSTATH		R	8	0x00
0xF346	Reserved	-	-	-	-	-
0xF347						
0xF350	16-bit timer X data register	TMHXDL	TMHXD	R/W	8/16	0xFF
0xF351		TMHXDH		R/W	8	0xFF
0xF352	16-bit timer X counter register	TMHXCL	TMHXC	R/W	8/16	0x00
0xF353		TMHXCH		R/W	8	0x00
0xF354	16-bit timer X mode register	TMHXMODL	TMHXMOD	R/W	8/16	0x00
0xF355		TMHXMODH		R/W	8	0x00
0xF356	Reserved	-	-	-	-	-
0xF357						
0xF358	16-bit timer X start register	TMHXSTR	-	W	8	0x00
0xF359	Reserved	-	-	-	-	-
0xF35A	16-bit timer X stop register	TMHXSTP	-	W	8	0x00
0xF35B	Reserved	-	-	-	-	-
0xF35C	16-bit timer X status register	TMHXSTAT	-	R	8	0x00
0xF35D	Reserved	-	-	-	-	-
0xF35E	Reserved	-	-	-	-	-
0xF35F						

8.2.2 16-Bit Timer n Data Register (TMHnD :n=0 to 4, X)

This is a SFR to set the comparison value with the 16-bit timer n counter register (TMHnC).

Address :	0xF300 (TMH0DL/TMH0D), 0xF301 (TMH0DH), 0xF308 (TMH1DL/TMH1D), 0xF309 (TMH1DH) 0xF310 (TMH2DL/TMH2D), 0xF311 (TMH2DH), 0xF318 (TMH3DL/TMH3D), 0xF319 (TMH3DH) 0xF320 (TMH4DL/TMH4D), 0xF321 (TMH4DH), 0xF350 (TMHXDL/TMHXD), 0xF351 (TMHXDH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0xFFFF

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

[Note]

- **Set TMHnD when the 16-bit timer n is stopped (THnSTATL bits of TMHSTAT register are "0").**
- **When "0x0000" is written in TMHnD in the 16-bit timer mode, "0x0001" is set in TMHnD.**
- **Set TMHnD so that the timer output frequency is 1MHz or less, when timer output is used.**
  The count clock frequency [MHz] / (TMHnD value + 1) x 2 ≤ 1 [MHz] , so that
  TMHnD value ≥ (count clock frequency [MHz] / 2) – 1 .

8.2.3 16-Bit Timer n Counter Register (TMHnC :n=0 to 4, X)

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

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

Address :	0xF302 (TMH0C/TMH0CL), 0xF303 (TMH0CH), 0xF30A (TMH1C/TMH1CL), 0xF30B (TMH1CH) 0xF312 (TMH2C/TMH2CL), 0xF313 (TMH2CH), 0xF31A (TMH3C/TMH3CL), 0xF31B (TMH3CH) 0xF322 (TMH4C/TMH4CL), 0xF323 (TMH4CH), 0xF352 (TMHXC/TMHXCL), 0xF353 (TMHXCH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0000

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

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

An available condition:

System clock	Timer clock
LSCLK0	LSCLK0
	LSCLK1 where the source clock is as same as one of LSCLK0.
	LTBnINT where the source clock is as same as one of LSCLK0.
HSCLK or divided HSCLK	HSCLK
	HTBCLK0
	HTBCLK1

[Note]

- **Read the TMHnC register twice to verify the valid data to prevent reading uncertain data while counting-up, if a source of timer clock is as different as one of system clock.**
  In case of SYSCLK frequency = 250kHz, the count clock frequency = 3MHz:
  If first read value is 0x0007, second read value is more than 0x0012. Valid bits are 11 bits of THnC15-5. It depend on reading interval time.

8.2.4 16-Bit Timer n Mode Register (TMHnMOD :n=0 to 4, X)

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

Address :	0xF304 (TMH0MODL/TMH0MOD), 0xF305 (TMH0MODH), 0xF30C (TMH1MODL/TMH1MOD), 0xF30D (TMH1MODH), 0xF314 (TMH2MODL/TMH2MOD), 0xF315 (TMH2MODH), 0xF31C (TMH3MODL/TMH3MOD), 0xF31D (TMH3MODH), 0xF324 (TMH4MODL/TMH4MOD), 0xF325 (TMH4MODH), 0xF354 (TMHXMODL/TMHXMOD), 0xF355 (TMHXMODH),
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	-	-	-	-	-	THnNEG	THnOST	-	-	-	THnDIV1	THnDIV0	THnCK3	THnCK2	THnCK1	THnCK0
R/W	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 11	-	Reserved bits
10	THnNEG	This bit is used to choose the output polarity of timer out (TMOn). 0: Positive logic (initial level is "L") (Initial value) 1: Negative logic (initial level is "H")
9	TLnOST	This bit is used to choose the operation mode. 0: Repeat mode (Initial value) 1: One-shot mode
8 to 6	-	Reserved bits
5 to 4	THnDIV1 to THnDIV0	These bits are used to choose frequency dividing ratio for the count clock 00: No dividing (Initial value) 01: Divided by 2 10: Divided by 4 11: Divided by 8
3 to 0	THnCK3 to THnCK0	These bits are used to choose the timer clock source. See Table 8-3 for detail. THXCK3 of timer X is reserved bit that is fixed 0.

Table 8-3 timer clock list

THnCLK3-0	0	1	2	3	4	X
0000	LSCLK0	LSCLK0	LSCLK0	LSCLK0	LSCLK0	LSCLK0
0001	HSCLK	HSCLK	HSCLK	HSCLK	HSCLK	HSCLK
0010	LSCLK1	LSCLK1	LSCLK1	LSCLK1	LSCLK1	LSCLK1
0011	HTBCLK0	HTBCLK0	HTBCLK0	HTBCLK0	HTBCLK0	HTBCLK0
0100	HTBCLK1	HTBCLK1	HTBCLK1	HTBCLK1	HTBCLK1	XT32K for stability.
0101	LTB1INT	LTB1INT	LTB1INT	LTB1INT	LTB1INT	RC1K
0110	LTB2INT	LTB2INT	LTB2INT	LTB2INT	LTB2INT	rsvd
0111	LTB3INT	LTB3INT	LTB3INT	LTB3INT	LTB3INT	-
1000	EXTRG0	EXTRG0	FTM0TRG	FTM0TRG	FTM0TRG	-
1001	EXTRG1	EXTRG1	FTM1TRG	FTM1TRG	FTM1TRG	-
1010	EXTRG2	EXTRG2	EXTRG2	EXTRG2	EXTRG2	-
1011	EXTRG3	EXTRG3	EXTRG3	EXTRG3	EXTRG3	-
1100	EXTRG4	rsvd	rsvd	EXTRG4	EXTRG4	-
1101	rsvd	EXTRG5	rsvd	EXTRG5	EXTRG5	-
1110	rsvd	rsvd	EXTRG6	EXTRG6	EXTRG6	-
1111	rsvd	rsvd	rsvd	EXTRG7	EXTRG7	-

LTBnINT	: Low speed time base counter interrupt
EXTRGn	: External interrupt trigger output
FTMnTRG	: Functional timer trigger output
RC1K	: RC1K output
XT32K for stability :	It is divided by 2 of XT32K/EXT32K. It is used to count for stability time only.

[Note]

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

8.2.5 16-Bit Timer Start Register (TMHSTR)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHSTR															
Byte	TMHSTRH								TMHSTRL							
Bit	-	-	-	-	-	-	-	-	-	-	-	TH4RUN	TH3RUN	TH2RUN	TH1RUN	TH0RUN
R/W	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is used to start a target timer.

Writing "0": Invalid
Writing "1": Start counting

Bit No.	Bit symbol name	Description (target)
15 to 5	-	Reserved bits
4	TH4RUN	16-bit timer 4
3	TH3RUN	16-bit timer 3
2	TH2RUN	16-bit timer 2
1	TH1RUN	16-bit timer 1
0	TH0RUN	16-bit timer 0

8.2.6 16-Bit Timer Stop Register (TMHSTP)

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

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	TMHSTP															
Byte	TMHSTPH								TMHSTPL							
Bit	-	-	-	-	-	-	-	-	-	-	-	TH4STP	TH3STP	TH2STP	TH1STP	TH0STP
R/W	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is used to stop a target timer.

Writing "0": Invalid
Writing "1": Stop counting

Bit No.	Bit symbol name	Description (target)
15 to 5	-	Reserved bits
4	TH4STP	16-bit timer 4
3	TH3STP	16-bit timer 3
2	TH2STP	16-bit timer 2
1	TH1STP	16-bit timer 1
0	TH0STP	16-bit timer 0

8.2.7 16-Bit Timer Status Register (TMHSTAT)

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

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

<table>
<tr><td></td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Word</td><td colspan="16">TMHSTAT</td></tr>
<tr><td>Byte</td><td colspan="8">TMHSTATH</td><td colspan="8">TMHSTATL</td></tr>
<tr><td>Bit</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>TH4STAT</td><td>TH3STAT</td><td>TH2STAT</td><td>TH1STAT</td><td>TH0STAT</td></tr>
<tr><td>R/W</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Common description of each bits :
It is used to indicate an operating status of a target timer

0: A counting of target timer is stopped (Initial value)
1: A counting of target timer is progress

Bit No.	Bit symbol name	Description (target)
15 to 5	-	Reserved bits
4	TH4STAT	16-bit timer 4
3	TH3STAT	16-bit timer 3
2	TH2STAT	16-bit timer 2
1	TH1STAT	16-bit timer 1
0	TH0STAT	16-bit timer 0

8.2.8 16-Bit Timer X Start Register (TMHXSTR)

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

Address :	0xF358 (TMHXSTR)
Access :	W
Access size :	8 bit
Initial value :	0x00

<table>
<tr><th></th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Word</td><td colspan="16">-</td></tr>
<tr><td>Byte</td><td colspan="8">-</td><td colspan="8">TMHXSTR</td></tr>
<tr><td>Bit</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>THXRUN</td></tr>
<tr><td>R/W</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>W</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	THXRUN	It is used to start timer X. Writing "0":Invalid Writing "1":Start counting

8.2.9 16-Bit Timer X Stop Register (TMHXSTP)

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

Address :	0xF35A (TMHXSTP)
Access :	W
Access size :	8 bit
Initial value :	0x00

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

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	THXSTP	It is used to stop timer X. Writing "0":Invalid Writing "1":Stop counting

8.2.10 16-Bit Timer X Status Register (TMHXSTAT)

This is a SFR to indicate the status of the 16-bit timer X.

Address :	0xF35C (TMHXSTAT)
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	-								TMHXSTAT							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	THXSTAT
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 1	-	Reserved bits
0	THXSTAT	It is used to indicate an operating status of timer X. 0: A counting of target timer is stopped (Initial value) 1: A counting of target timer is progress

8.3 Description of Operation

8.3.1 Operation Mode

Writing "1" to the THnRUN bit causes the 16-bit counter n to start counting up in synchronization with the rising edges of the count clock.
If output of the general-purpose port is enabled by choosing the timer output (TMOn) 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 count clock and the output of the port is reset to the initial value. For the initial value of the port, "H" and "L" levels can be chosen through the THnNEG bit of the TMHnMOD register.

Following two operation modes are available:

- Repeat mode
- One-shot mode

8.3.1.1 Repeat Mode

Figure 8-2 shows the repeat mode operation.

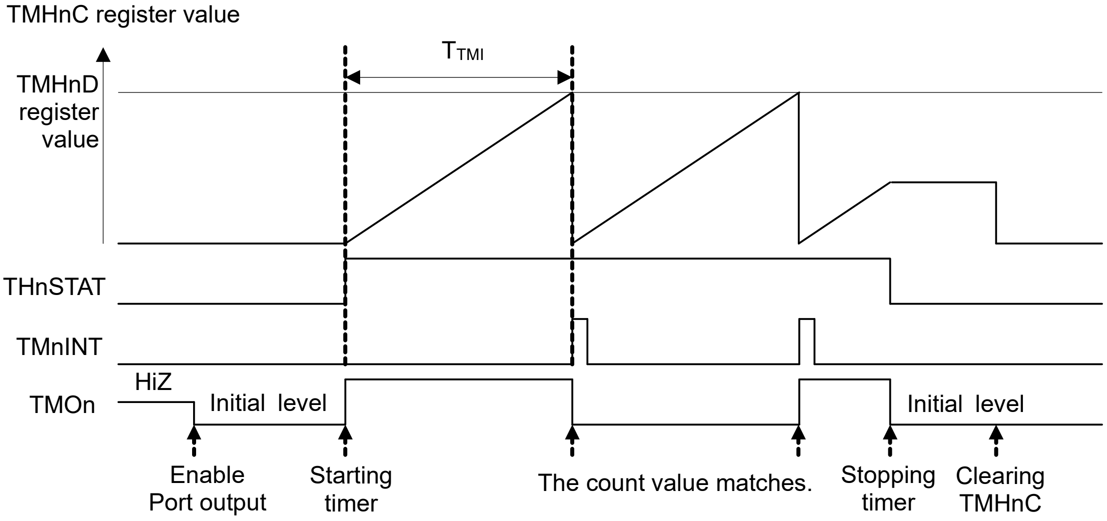


Figure 8-2 Repeat Mode Operation Timing

In the repeat mode, when the timer count value matches with 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:

$$\mathrm{TTMI} = \frac{\mathrm{TMHnD} + 1}{\mathrm{fTHnCK} \quad (\mathrm{Hz})} \qquad (\mathrm{n}=0 \text{ to } 4, \mathrm{X})$$

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

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

8.3.1.2 One-shot Mode

Figure 8-3 shows the one-shot mode operation

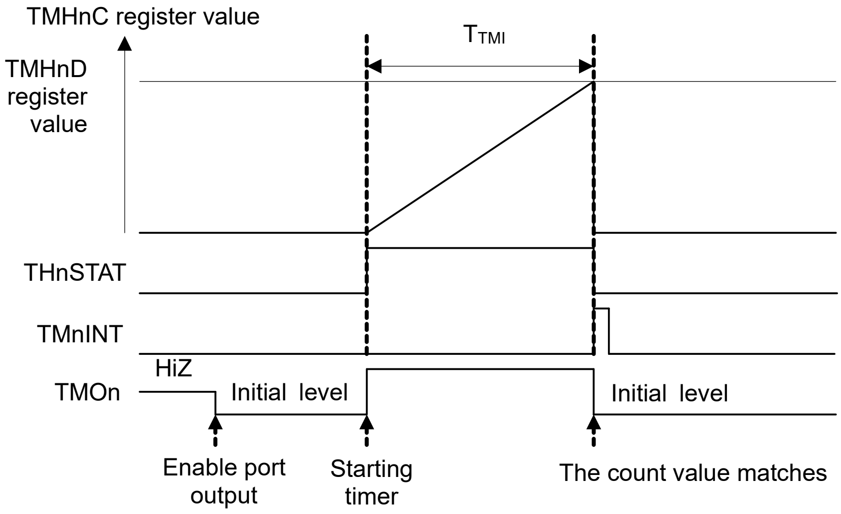


Figure 8-3 One-shot Mode Operation Timing

In the one-shot mode, when the timer count value matches with 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.

8.3.2 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 after the falling edge of the timer clock.
Figure 8-4 shows the timer start timing when the timer clock is LSCLK0 and frequency dividing ratio of the count clock is 1/2 of the timer clock

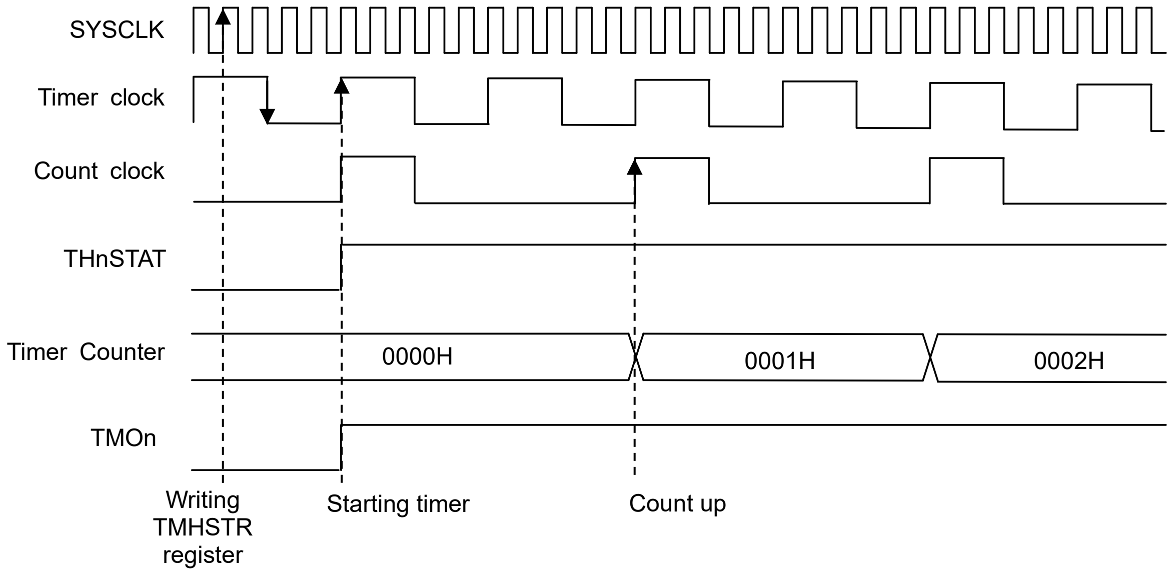


Figure 8-4 Start Timing

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

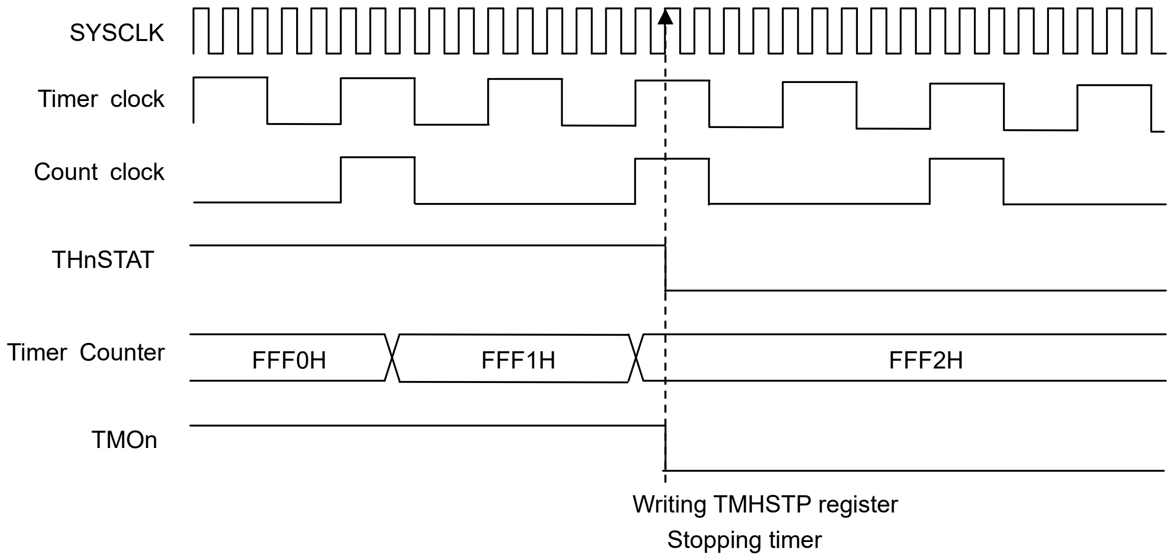


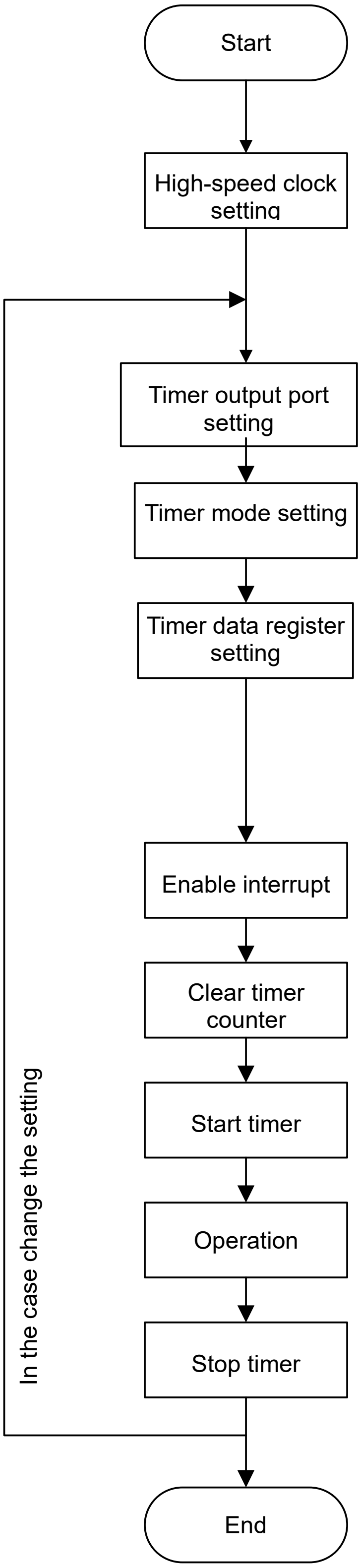
Figure 8-5 Stop Timing

[Note]

- **After 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 Setting Example

Figure 8-6 shows a setting example.



Need when using the high-speed clock (HSCLK or HTBCLK0/1) for the timer clock

Choose TMOn as the shared function and specify enabling the output by the PnMODxx register.

Specify the timer mode, timer operation mode, timer clock, count clock, and the timer output logic in the TMHnMOD register.

Set a comparison value into TMHnD register that is to be compared with TMHnC register.

QTMn=0; Reset corresponding bits of IRQ23 register, IRQ45 register and IRQ67 register to "0".

ETMn=1; Set corresponding bits of IE23 register, IE45 register and IE67 register to "1".

TMHnC=0x0000; Write a data into the TMHnC register

THnRUN=1; Set corresponding bits TMHSTR register to "1"

THnSTP=1; Set corresponding bits TMHSTP register to "1"

Figure 8-6 Setting Example

8.3.4 Using 16-bit Timer X for Crystal Oscillation Stability

It is necessary to use Timer X when using a crystal oscillation/external clock input for the low-speed clock. It can be used as a normal timer when the oscillation stabilization waiting is completed.

16-bit Timer X operates as counter of crystal oscillation stability time with THXCK2-0=4.
In the mode, an interrupt timing is at LOSCS becomes "0".
In other mode, the timing is at coinciding TMHXD value and TMHXC value.
See Chapter 6 "Clock Generation Circuit" for how to use.

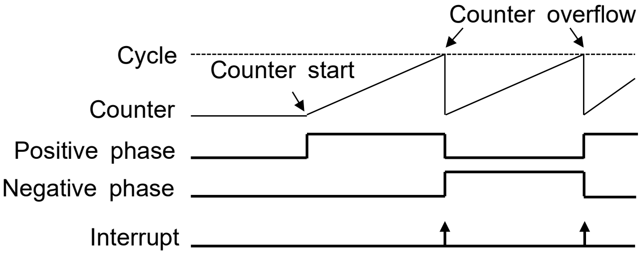
Chapter 9 Functional Timer (FTM)

9. Functional Timer

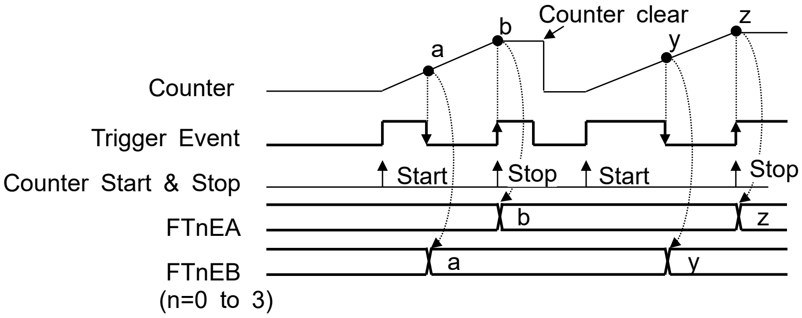
9.1 General Description

The Functional timer enables following functions in four operation modes (TIMER/CAPTURE/PWM1/PWM2).

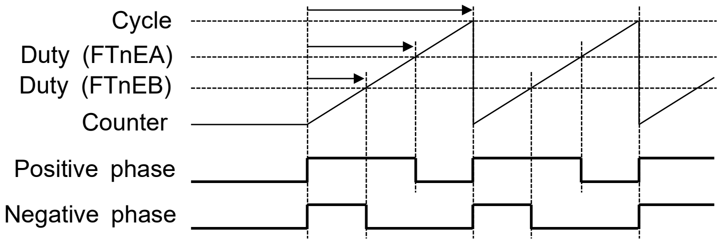
TIMER mode:
In this mode, the Functional Timer generates pulse signals, levels of which are reversed in sync with the counter start and the counter overflow. Also, it generates the interrupt when the counter overflows.



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.

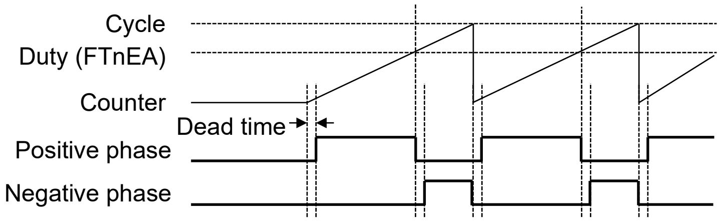


Table 9-1 shows the number of channels.

Table 9-1 Number of Functional Timer channels

Channel no.	ML62Q2500 group
0	●
1	●
2	-
3	-
4	-
5	-
6	-
7	-

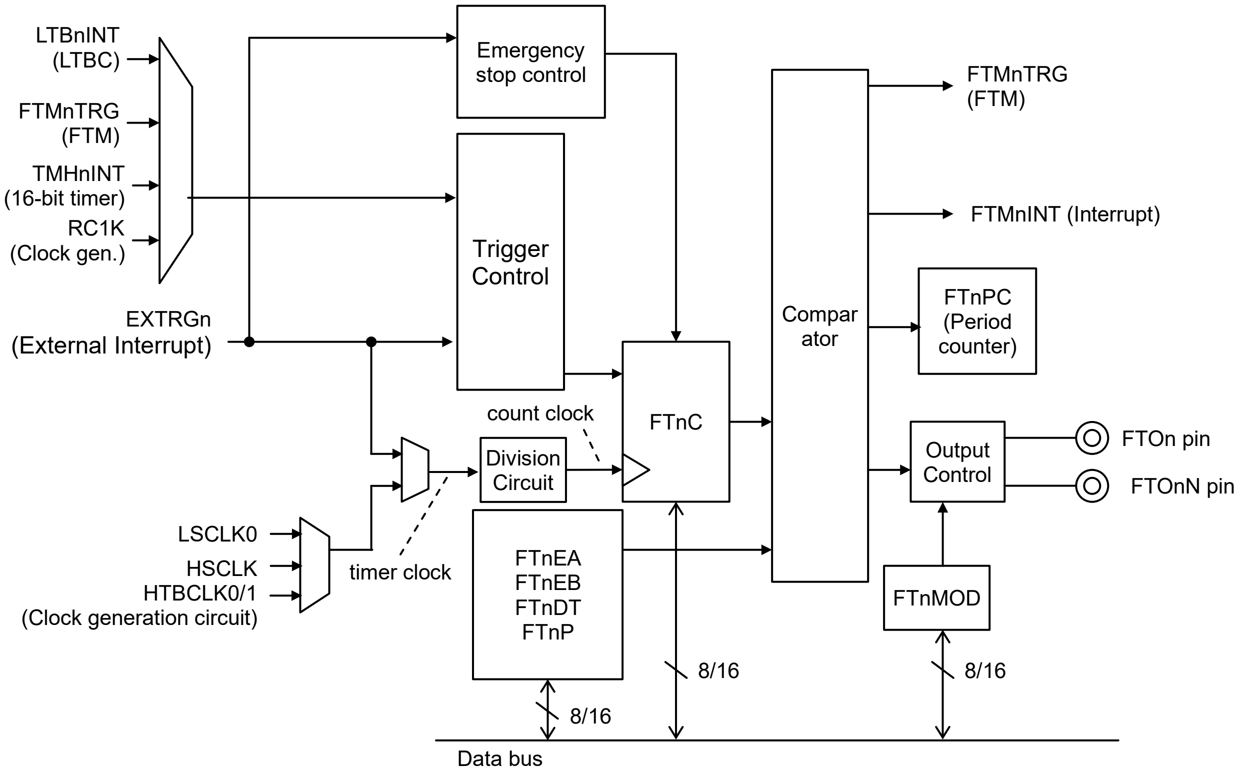
●: Available - : Unavailable

9.1.1 Features

- The Timer/Capture/PWM functions using the 16-bit counter
- Selectable counter clock from various sources (divided by 1 to 128 of LSCLK0, HSCLK, HTBCLK0/1, external clock
- 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

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 (n=0 to 1) : Functional Timer n trigger
EXTRGn : External trigger n / external clock n (n=0 to 7)
LTBnINT : Low speed time base counter interrupt n (n=1 to 3)
TMHnINT : 16-bit Timer n interrupt (n=0 to 4, X)

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
EXIn	I	External trigger/clock (used as EXTRGn). The max input frequency is 3MHz.
FTOn	O	Functional timer channel n output P ; n= 0 to 1
FTOnN	O	Functional timer channel n output N ; n= 0 to 1

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

Pin name	Shared port		Register	Setting value	ML62Q2500 group		
					32pin product	40pin product	48pin product
EXI0	Ports assigned for external interrupt. See Chapter 18. External interrupt.				●	●	●
EXI1					●	●	●
EXI2					●	●	●
EXI3					●	●	●
EXI4					●	●	●
EXI5					●	●	●
EXI6					●	●	●
EXI7					●	●	●
FTO0	P10	5th Func.	P1MOD0	0100_XXXX*1	●	●	●
	P20	5th Func.	P2MOD0	0100_XXXX*1	●	●	●
	P30	5th Func.	P3MOD0	0100_XXXX*1	●	●	●
	P54	5th Func.	P5MOD4	0100_XXXX*1	-	●	●
	P70	5th Func.	P7MOD0	0100_XXXX*1	●	●	●
FTO0N	P11	5th Func.	P1MOD1	0100_XXXX*1	●	●	●
	P21	5th Func.	P2MOD1	0100_XXXX*1	●	●	●
	P31	5th Func.	P3MOD1	0100_XXXX*1	●	●	●
	P55	5th Func.	P5MOD5	0100_XXXX*1	-	●	●
	P71	5th Func.	P7MOD1	0100_XXXX*1	●	●	●
FTO1	P12	5th Func.	P1MOD2	0100_XXXX*1	●	●	●
	P22	5th Func.	P2MOD2	0100_XXXX*1	●	●	●
	P32	5th Func.	P3MOD2	0100_XXXX*1	●	●	●
	P56	5th Func.	P5MOD6	0100_XXXX*1	-	-	●
	P72	5th Func.	P7MOD2	0100_XXXX*1	●	●	●
FTO1N	P13	5th Func.	P1MOD3	0100_XXXX*1	●	●	●
	P23	5th Func.	P2MOD3	0100_XXXX*1	●	●	●
	P33	5th Func.	P3MOD3	0100_XXXX*1	●	●	●
	P57	5th Func.	P5MOD7	0100_XXXX*1	-	-	●
	P73	5th Func.	P7MOD3	0100_XXXX*1	●	●	●

*1 : "XXXX" determines the condition of the port output

XXXX	Condition of the port output
0010	CMOS output
1010	N-ch open drain output (without the pull-up)
1111	N-ch open drain output (with the pull-up)

[Note]

- **Assign FTOn, FTOnN to only one LSI pin each.**

9.2 Description of Registers

9.2.1 List of Registers

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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF380	FTM common update register	FTCUD	-	W	8	0x00
0xF381	Reserved	-	-	-	-	-
0xF382	FTM common control register	FTCCONL	FTCCON	R/W	8/16	0x00
0xF383		FTCCONH		R/W	8	0x00
0xF384	FTM common start register	FTCSTRL	FTCSTR	W	8/16	0x00
0xF385		FTCSTRH		W	8	0x00
0xF386	FTM common stop register	FTCSTPL	FTCSTP	W	8/16	0x00
0xF387		FTCSTPH		W	8	0x00
0xF388	FTM common status register	FTCSTATL	FTCSTAT	R	8/16	0x00
0xF389		FTCSTATH		R	8	0x00
0xF38A	Reserved	-	-	-	-	-
0xF38B		-		-	-	-
0xF38C	Reserved	-	-	-	-	-
0xF38D		-		-	-	-
0xF38E	Reserved	-	-	-	-	-
0xF38F		-		-	-	-
0xF400	FTM0 cycle register	FT0PL	FT0P	R/W	8/16	0xFF
0xF401		FT0PH		R/W	8	0xFF
0xF402	FTM0 event A register	FT0EAL	FT0EA	R/W	8/16	0x00
0xF403		FT0EAH		R/W	8	0x00
0xF404	FTM0 event B register	FT0EBL	FT0EB	R/W	8/16	0x00
0xF405		FT0EBH		R/W	8	0x00
0xF406	FTM0 dead time register	FT0DTL	FT0DT	R/W	8/16	0x00
0xF407		FT0DTH		R/W	8	0x00
0xF408	FTM0 counter register	FT0CL	FT0C	R/W	8/16	0x00
0xF409		FT0CH		R/W	8	0x00
0xF40A	FTM0 status register	FT0STAT	-	R	8	0x30
0xF40B	Reserved	-	-	-	-	-
0xF40C	FTM0 mode register	FT0MODL	FT0MOD	R/W	8/16	0x00
0xF40D		FT0MODH		R/W	8	0x40
0xF40E	FTM0 clock register	FT0CLKL	FT0CLK	R/W	8/16	0x00
0xF40F		FT0CLKH		R/W	8	0x00
0xF410	FTM0 trigger register 0	FT0TRG0L	FT0TRG0	R/W	8/16	0x00
0xF411		FT0TRG0H		R/W	8	0x00
0xF412	FTM0 trigger register 1	FT0TRG1L	FT0TRG1	R/W	8/16	0x00
0xF413		FT0TRG1H		R/W	8	0x00
0xF414	FTM0 interrupt enable register	FT0INTEL	FT0INTE	R/W	8/16	0x00
0xF415		FT0INTEH		R/W	8	0x00
0xF416	FTM0 interrupt status register	FT0INTSL	FT0INTS	R	8/16	0x00
0xF417		FT0INTSH		R	8	0x00
0xF418	FTM0 interrupt clear register	FT0INTCL	-	W	8	0x00
0xF419		FT0INTCH		W	8	0x00
0xF41A	Reserved	-	-	-	-	-
0xF41B		-		-	-	-
0xF41C	Reserved	-	-	-	-	-
0xF41D		-		-	-	-

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF41E	Reserved	-	-	-	-	-
0xF41F		-		-	-	-
0xF420	FTM1 cycle register	FT1PL	FT1P	R/W	8/16	0xFF
0xF421		FT1PH		R/W	8	0xFF
0xF422	FTM1 event A register	FT1EAL	FT1EA	R/W	8/16	0x00
0xF423		FT1EAH		R/W	8	0x00
0xF424	FTM1 event B register	FT1EBL	FT1EB	R/W	8/16	0x00
0xF425		FT1EBH		R/W	8	0x00
0xF426	FTM1 dead time register	FT1DTL	FT1DT	R/W	8/16	0x00
0xF427		FT1DTH		R/W	8	0x00
0xF428	FTM1 counter register	FT1CL	FT1C	R/W	8/16	0x00
0xF429		FT1CH		R/W	8	0x00
0xF42A	FTM1 status register	FT1STAT	-	R	8	0x30
0xF42B	Reserved	-	-	-	-	-
0xF42C	FTM1 mode register	FT1MODL	FT1MOD	R/W	8/16	0x00
0xF42D		FT1MODH		R/W	8	0x40
0xF42E	FTM1 clock register	FT1CLKL	FT1CLK	R/W	8/16	0x00
0xF42F		FT1CLKH		R/W	8	0x00
0xF430	FTM1 trigger register 0	FT1TRG0L	FT1TRG0	R/W	8/16	0x00
0xF431		FT1TRG0H		R/W	8	0x00
0xF432	FTM1 trigger register 1	FT1TRG1L	FT1TRG1	R/W	8/16	0x00
0xF433		FT1TRG1H		R/W	8	0x00
0xF434	FTM1 interrupt enable register	FT1INTEL	FT1INTE	R/W	8/16	0x00
0xF435		FT1INTEH		R/W	8	0x00
0xF436	FTM1 interrupt status register	FT1INTSL	FT1INTS	R	8/16	0x00
0xF437		FT1INTSH		R	8	0x00
0xF438	FTM1 interrupt clear register	FT1INTCL	-	W	8	0x00
0xF439		FT1INTCH		W	8	0x00
0xF43A	Reserved	-	-	-	-	-
0xF43B		-		-	-	-
0xF43C	Reserved	-	-	-	-	-
0xF43D		-		-	-	-
0xF43E	Reserved	-	-	-	-	-
0xF43F		-		-	-	-

9.2.2 FTMn Cycle Register (FTnP :n=0 to 1)

This is a SFR to set the cycle (clock count) of FTMn.
Set this register after setting the operation mode using FTnMD1 to 0 bits of FTnMOD register.

Address :	0xF400 (FT0PL/FT0P), 0xF401 (FT0PH), 0xF420 (FT1PL/FT1P), 0xF421 (FT1PH)
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	FTnP15	FTnP14	FTnP13	FTnP12	FTnP11	FTnP10	FTnP9	FTnP8	FTnP7	FTnP6	FTnP5	FTnP4	FTnP3	FTnP2	FTnP1	FTnP0
R/W	R/W	R/W	R/W	R/W	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.**
- **Set FTnP so that the functional timer output frequency is 3MHz or less, when its output is used.**
  The count clock frequency [MHz] / (FTnP value + 1) ≤ 3 [MHz] , so that
  FTnP value ≥ (count clock frequency [MHz] / 3) – 1 .

9.2.3 FTMn Event A Register (FTnEA :n=0 to 1)

This is a 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 :	0xF402 (FT0EAL/FT0EA), 0xF403 (FT0EAH), 0xF422 (FT1EAL/FT1EA), 0xF423 (FT1EAH)
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	FTnEA 15	FTnEA 14	FTnEA 13	FTnEA 12	FTnEA 11	FTnEA 10	FTnEA 9	FTnEA 8	FTnEA 7	FTnEA 6	FTnEA 5	FTnEA 4	FTnEA 3	FTnEA 2	FTnEA 1	FTnEA 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	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. However, FTnFLGA is cleared only when FTnTGEN=1. See section 9.3.4.2 as for clearing FTnFLGA. -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]. The duty 0% is configurable when FTnEA = 0xFFFF. The duty 100% is configurable when FTnP = FTnEA. The duty becomes 0% when FTnP = 0xFFFF. Do not set value more than one of FTnP except 0xFFFF. -PWM2 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]. The duty 0% is configurable when FTnEA = 0xFFFF. The duty 100% is configurable when FTnP = FTnEA. The duty becomes 0% when FTnP = 0xFFFF. Do not set value more than one of FTnP except 0xFFFF.

[Note]

- **In timer mode , a data set in the FTnEA register must be less than that set in the FTnP register.**
- **In PWM1/2 mode, a data set in the FTnEA register must be 0xFFFF or less than that set in the FTnP register.**

9.2.4 FTMn Event B Register (FTnEB :n=0 to 1)

This is a 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 :	0xF404 (FT0EBL/FT0EB), 0xF405 (FT0EBH), 0xF424 (FT1EBL/FT1EB), 0xF425 (FT1EBH)
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	FTnEB 15	FTnEB 14	FTnEB 13	FTnEB 12	FTnEB 11	FTnEB 10	FTnEB 9	FTnEB 8	FTnEB 7	FTnEB 6	FTnEB 5	FTnEB 4	FTnEB 3	FTnEB 2	FTnEB 1	FTnEB 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	FTnEB15 to FTnEB0	-TIMER mode 0x0000 to 0xFFFF: Set a count value to generate an interrupt. The 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 FTnISA bit of FTMn interrupt status register (FTnINTS) is cleared. In the CAPTURE mode, writing to FTnEB is invalid. However, FTnFLGB is cleared only when FTnTGEN=1. See section 9.3.4.2 as for clearing FTnFLGB. -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]. The duty 0% is configurable when FTnEB = 0xFFFF. The duty 100% is configurable when FTnP = FTnEB. Do not set value more than one of FTnP except 0xFFFF. The duty becomes 0% when FTnP = 0xFFFF. -PWM2 mode Set 0x0000 only.

[Note]

- **In timer mode , a data set in the FTnEB register must be less than that set in the FTnP register.**
- **In PWM1 mode, a data set in the FTnEB register must be 0xFFFF or less than that set in the FTnP register.**

9.2.5 FTMn Dead Time Register (FTnDT :n=0 to 1)

This is a SFR to set the dead time of output signal.
Set this register after setting the operation mode using FTnMD1 to 0 bits of FTnMOD register.

Address :	0xF406 (FT0DTL/FT0DT), 0xF407 (FT0DTH), 0xF426 (FT1DTL/FT1DT), 0xF427 (FT1DTH)
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	FTnDT 15	FTnDT 14	FTnDT 13	FTnDT 12	FTnDT 11	FTnDT 10	FTnDT 9	FTnDT 8	FTnDT 7	FTnDT 6	FTnDT 5	FTnDT 4	FTnDT 3	FTnDT 2	FTnDT 1	FTnDT 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	FTnDT15 to FTnDT0	-TIMER, PWM1/2 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 is invalid.

[Note]

- **In the PWM2 mode, 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.**

9.2.6 FTMn Counter Register (FTnC :n=0 to 1)

This is a SFR to display the counter value of FTMn. When writing to this register, the counter is cleared to "0x0000" in one clock of the count clock.

Address :	0xF408 (FT0CL/FT0C), 0xF409 (FT0CH), 0xF428 (FT1CL/FT1C), 0xF429 (FT1CH)
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	FTnC15	FTnC14	FTnC13	FTnC12	FTnC11	FTnC10	FTnC9	FTnC8	FTnC7	FTnC6	FTnC5	FTnC4	FTnC3	FTnC2	FTnC1	FTnC0
R/W	R/W	R/W	R/W	R/W	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

In case of the following table, Reading value is available. In other case, do reading this register twice and confirm those equivalence.

System clock	Timer clock
LSCLK0	LSCLK0
HSCLK	HSCLK, HTBCLK0, HTBCLK1

[Note]

- **Read FTnC register twice to verify the data to prevent reading uncertain data while counting-up according to need.Read the FTnC register twice to verify the valid data to prevent reading uncertain data while counting-up, if a source of timer clock is as different as one of system clock.**
  In case of SYSCLK frequency = 250kHz, the count clock frequency = 3MHz:
  If first read value is 0x0007, second read value is more than 0x0012. Valid bits are 11 bits of FTnC15-5. It depend on reading interval time.

9.2.7 FTMn Status Register (FTnSTAT :n=0 to 1)

FTnSTAT is a read-only SFR to indicate the state of FTMn.

Address :	0xF40A (FT0STAT), 0xF42A (FT1STAT)
Access :	R
Access size :	8 bit
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 is used to indicate the operation state of FTMn. 0: The counter is stopped (Initial value) 1: The counter is running
6	FTnFLGC	This is used to indicate whether the next event start is enable or disable while a counter chosen by FTnCST bit of FTnTRG0 register is begin stopped. This is cleared by reading FTnC register automatically. 0: Starting by event trigger is enabled. (Initial value) 1: Starting by event trigger is disabled.
5	FTnFLGB	This is used to indicate the state of event timing B. -TIMER/PWM1 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 captured data 1: There is captured data. To be cleared by reading the FTnEB register. However, FTnFLGB is cleared only when FTnTGEN=1. See section 9.3.4.2 as for clearing FTnFLGB. -PWM2 mode Fixed 0.
4	FTnFLGA	This is used to indicate the state of event timing A. -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 captured data 1: There is captured data. To be cleared by reading the FTnEA register. However, FTnFLGA is cleared only when FTnTGEN=1. See section 9.3.4.2 as for clearing FTnFLGA.
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 or 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 1)

This is a SFR to set the FTOn and FTOnN pin output function and the operation mode.

Address :	0xF40C (FT0MODL/FT0MOD), 0xF40D (FT0MODH), 0xF42C (FT1MODL/FT1MOD), 0xF42D (FT1MODH)
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	FTnSTSYN	FTnSTPO	FTnOST	-	FTnDTENN	FTnDTENP	-	-	FTnMD1	FTnMD0
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	R	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	FTnOSL1	This bit is used to choose the phase of signal output at FTOnN pin. 0: Output Negative phase (Initial value) 1: Output Positive phase
14	FTnOSL0	This bit is used to choose the phase of signal output at FTOn pin. 0: Output Negative phase 1: Output Positive phase (Initial value)
13	FTnOSNN	This bit is used to reverse the FTOnN 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, 10	rsvd	Reserved bits. Set "0" to all bits.
9	FTnSTSYN	This bit is used to choose stop timing. 0: Just when writing stop register or receiving a stop trigger. 1: When an end of cycle after writing stop register or receiving a stop trigger. This function is available for stopping from software or trigger. It is not active for the emergency stop trigger If a start event occurs between the time the stop event occurs and the end of the cycle when this bit is “1”, the count operation will continue without stopping at the end of the cycle.
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	-	Reserved bit
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	-	Reserved bits
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

[Note]

- **Set the FTnMOD register when the FTMn is stopped.**
- **Initialize this peripheral with block reset before changing to another mode, if it is in the operation state once.**

9.2.9 FTMn Clock Register (FTnCLK :n=0 to 1)

This is a SFR to set the timer clock and count clock of the FTMn.

Address :	0xF40E (FT0CLKL/FT0CLK), 0xF40F (FT0CLKH), 0xF42E (FT1CLKL/FT1CLK), 0xF42F (FT1CLKH)
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	-	-	-	-	-	-	-	-	-	FTnCKD2	FTnCKD1	FTnCKD0	FTnCK3	FTnCK2	FTnCK1	FTnCK0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
6 to 4	FTnCKD2 to FTnCKD0	These bits are used to configure count clock frequency; divided timer source clock. 000: No dividing (Initial value) 001: Divided the timer clock by 2 010: Divided the timer clock by 4 011: Divided the timer clock by 8 100: Divided the timer clock by 16 101: Divided the timer clock by 32 110: Divided the timer clock by 64 111: Divided the timer clock by 128
3 to 0	FTnCK3 to FTnCK0	This bits are used to choose timer clock source. 0000: LSCLK0 (Initial value) 0001: HSCLK 0010: HTBCLK0 0011: HTBCLK1 0100: Do not use (LSCLK1) 0101: Do not use (HSCLK) 0110: Do not use (HTBCLK0) 0111: Do not use (HTBCLK1) 1000: External clock 0 input (EXTRG0) 1001: External clock 1 input (EXTRG1) 1010: External clock 2 input (EXTRG2) 1011: External clock 3 input (EXTRG3) 1100: External clock 4 input (EXTRG4) 1101: External clock 5 input (EXTRG5) 1110: External clock 6 input (EXTRG6) 1111: External clock 7 input (EXTRG7)

9.2.10 FTMn Trigger Register 0 (FTnTRG0 :n=0 to 1)

This is a SFR to set the trigger function of FTMn.

Address :	0xF410 (FT0TRG0L/FT0TRG0), 0xF411 (FT0TRG0H), 0xF430 (FT1TRG0L/FT1TRG0), 0xF431 (FT1TRG0H)
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	-	FTnEST1	FTnEST0	FTnSTSS	FTnSTS3	FTnSTS2	FTnSTS1	FTnSTS0	FTnDCLD	FTnDCLH	FTnCST	-	FTnSPC	FTnSP	FTnSTC	FTnST
R/W	R	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	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, 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: Reserved 11: Reserved
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) 010XX: Reserved 01100: RC1K 01101: Low speed time base counter interrupt 1 (LTB1INT) 01110: Low speed time base counter interrupt 2 (LTB2INT) 01111: Low speed time base counter interrupt 3 (LTB3INT) X : Don’t care 0 or 1. -TIMER/PWM1/PWM2 mode 10000: 16-bit timer 0 interrupt (TMH0INT) 10001: 16-bit timer 1 interrupt (TMH1INT) 10010: 16-bit timer 2 interrupt (TMH2INT) 10011: 16-bit timer 3 interrupt (TMH3INT) 10100: 16-bit timer 4 interrupt (TMH4INT) 10101: 16-bit timer X interrupt (TMHXINT) 10110: Reserved 10111: Reserved 11000: Functional timer 0 trigger (FTM0TRG) 11001: Functional timer 1 trigger (FTM1TRG) 11010: Reserved 11011: Reserved 11100: Reserved 11101: Reserved 11110: Reserved 11111: Reserved

Bit No.	Bit symbol name	Description
7	FTnDCLD	This bit is used to enable/disable the FTnDCLH bit function at dead-time zone. ・TIMER/PWM1/PWM2 mode 0: Enabled during dead time (Initial value) 1: Disabled during dead time
6	FTnDCLH	This bit is used to disable the counter clear by a trigger event when positive phase output is "H" level. -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	-	Reserved bit
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. 0: Disabled (Initial value) 1: Enabled However, the counter is not cleared/updated regardless of this bit in the following cases. When the emergency stop occurs. When Setting FTnTRM2-0 bits to "000" or "011". When Setting TMHnINT or FTMnTRG as trigger event source.
2	FTnSP	This bit is used to choose whether to enable stopping the counter by a trigger event. 0: Disabled (Initial value) 1: 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: Disabled (Initial value) 1: 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". When Setting TMHnINT 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: Disabled (Initial value) 1: Enabled

[Note]

- **The input pulse width must have two timer clocks or longer if FTnSTSS=0.**
- **The counter forcibly stops and does not run when the emergency stop trigger source is the same as the trigger event source with the FTnETG = 1 and FTnEMGEN = 1.**

9.2.11 FTMn Trigger Register 1 (FTnTRG1 :n=0 to 1)

This is a SFR to set the trigger function of FTMn.

Address :	0xF412 (FT0TRG1L/FT0TRG1), 0xF413 (FT0TRG1H), 0xF432 (FT1TRG1L/FT1TRG1), 0xF433 (FT1TRG1H)
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	-	-	-	-	-	-	-	-	-	-	-	FTnEMGES	-	FTnTRM2	FTnTRM1	FTnTRM0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 5	-	Reserved bits
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
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 FTnSTSS bit is 0. 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]

- **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.**
- **The trigger may occur immediately after setting the FTnTRG1 register in the trigger event enabled.**

9.2.12 FTMn Interrupt Enable Register (FTnINTE :n=0 to 1)

This is a 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.

Address :	0xF414 (FT0INTEL/FT0INTE), 0xF415 (FT0INTEH), 0xF434 (FT1INTEL/FT1INTE), 0xF435 (FT1INTEH)
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	-	-	-	-	-	FTnIOB	FTnIOA	FTnIOP	-	-	-	FTnIETR	FTnIETS	FTnIEB	FTnIEA	FTnIEP
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is used to set enable/disable a target function.

0: Disabled (Initial value)
1: Enabled

Bit No.	Bit symbol name	Description (Target function)
15 to 11	-	Reserved bits
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.
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.
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.
7 to 5	-	Reserved bits
4	FTnIETR	The trigger counter start interrupt
3	FTnIETS	The trigger counter stop interrupt
2	FTnIEB	-TIMER/PWM1/CAPTURE mode The event timing B interrupt -PWM2 mode Write always "0"
1	FTnIEA	The event timing A interrupt
0	FTnIEP	The cyclic interrupt

9.2.13 FTMn Interrupt Status Register (FTnINTS :n=0 to 1)

FTnINTS is a read-only SFR to indicate the interrupt status of FTMn.
The bit 5 to bit 0 is reset to "0" by writing "1" to the same number of bit in the FTnINTC register.

Address :	0xF416 (FT0INTSL/FT0INTS), 0xF417 (FT0INTSH), 0xF436 (FT1INTSL/FT1INTS), 0xF437 (FT1INTSH)
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

Common description of each bits :
It is used to indicate a target interrupt status.

0: Target interrupt has not occurred. (Initial value)
1: Target interrupt has occurred.

Bit No.	Bit symbol name	Description (Target interrupt)
15 to 6	-	Reserved bits
5	FTnISES	The emergency stop interrupt This is cleared when writing 1 to FTnICES bit of FTnINTC register.
4	FTnISTR	The trigger counter start interrupt It is set when counter-starting by trigger event or counter-clear by trigger event for counter-starting occur. This is cleared when writing 1 to FTnICTR bit of FTnINTC register.
3	FTnISTS	The trigger counter stop interrupt It is set when counter-stopping by trigger event or counter-clear by trigger event for counter-stopping occur. This is cleared when writing 1 to FTnICTS bit of FTnINTC register.
2	FTnISB	The event timing B interrupt This is cleared when writing “1” to FTnICB bit of FTnINTC register. It indicates that the captured data is stored to the FTnEB register in the CAPTURE mode. This is cleared when reading the FTnEB register in the CAPTURE mode.
1	FTnISA	The event timing A interrupt This is cleared when writing “1” to FTnICA bit of FTnINTC register. It indicates that the captured data is stored to the FTnEA register in the CAPTURE mode. This is cleared when reading the FTnEA register in the CAPTURE mode.
0	FTnISP	The cyclic interrupt This is cleared when writing "1" to FTnICP bit of FTnINTC register.

[Note]

- **If the FTnINTS register is not zero, a request to interrupt controller is not given when a new interrupt occurs. Clear the FTnINTS register with the FTnINTC register before that time.**

9.2.14 FTMn Interrupt Clear Register L/H (FTnINTCL, FTnINTCH : n=0 to 1)

This is a 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. This register always returns 0x0000 for reading.

Address : 0xF418 (FT0INTCL), 0xF419 (FT0INTCH),
0xF438 (FT1INTCL), 0xF439 (FT1INTCH),
Access : W
Access size : 8 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	FTnINTCH								FTnINTCL							
Bit	FTnIR	-	-	-	-	-	-	-	-	-	FTnICES	FTnICTR	FTnICTS	FTnICB	FTnICA	FTnICP
R/W	W	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits (bit 5-0) :
It is used to clear target interrupt status flag.
Writing "0" : Invalid
Writing "1" : Clear a target interrupt status flag.

Bit No.	Bit symbol name	Description (Target interrupt)
15	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.
14 to 6	-	Reserved bits
5	FTnICES	The emergency stop interrupt
4	FTnICTR	The trigger counter start interrupt
3	FTnICTS	The trigger counter stop interrupt
2	FTnICB	The event timing B interrupt
1	FTnICA	The event timing A interrupt
0	FTnICP	The cyclic interrupt

9.2.15 FTM Common Update Register (FTCUD)

This is a 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 :	0xF380 (FTCUD)
Access :	W
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FTCUD							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FTCUD1	FTCUD0
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 to 0	FTCUD1 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)

This is a SFR to set the function of FTMn. This 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 :	0xF382 (FTCCONL/FTCCON), 0xF383 (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	-	-	-	-	-	-	FT1SDN	FT0SDN	-	-	-	-	-	-	FT1EMGEN	FT0EMGEN
R/W	R	R	R	R	R	R	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 10	-	Reserved bits
9 to 8	FT1SDN to FT0SDN	This bit is used to enable controlling the positive phase/negative phase output. -TIMER/PWM1/PWM2 mode 0: Enabled (Initial value) 1: Disabled (The output is fixed to "L" level) -CAPTURE mode This bit is invalid
7 to 2	-	Reserved bits
1 to 0	FT1EMGEN to FT0EMGEN	This bit is used to enable the emergency stop on the FTMn. 0: Disabled (Initial value) 1: Enabled

9.2.17 FTM Common Start Register (FTCSTR)

This is a SFR to set the function of FTMn. This is an SFR common to each channel. The bit n corresponds to channel n. It is unavailable to write to the bits for unequipped channels.

Address : 0xF384 (FTCSTRL/FTCSTR), 0xF385 (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	-	-	-	-	-	-	FT1ETG	FT0ETG	-	-	-	-	-	-	FT1STR	FT0STR
R/W	R	R	R	R	R	R	W	W	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
15 to 10	-	Reserved bits
9 to 8	FT1ETG to FT0ETG	These bits are used to enable trigger operation; counting stop/start by a trigger event. 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 2	-	Reserved bits
1 to 0	FT1STR 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

9.2.18 FTM Common Stop Register (FTCSTP)

This is a SFR to set the function of FTMn. This 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 :	0xF386 (FTCSTPL/FTCSTP), 0xF387 (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	-	-	-	-	-	-	FT1DTG	FT0DTG	-	-	-	-	-	-	FT1STP	FT0STP
R/W	R	R	R	R	R	R	W	W	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
15 to 10	-	Reserved bits
9 to 8	FT1DTG to FT0DTG	These bits are used to disable trigger operation, counting stop/start by a trigger event. 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 by a trigger event.
7 to 2	-	Reserved bits
1 to 0	FT1STP 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

9.2.19 FTM Common Status Register (FTCSTAT)

This is a SFR to indicate the state of FTMn. This is a common SFR to each channel. The bit n corresponds to channel n.

Address :	0xF388 (FTCSTATL/FTCSTAT), 0xF389 (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	-	-	-	-	-	-	FT1TGEN	FT0TGEN	-	-	-	-	-	-	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 10	-	Reserved bits
9 to 8	FT1TGEN 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) is disabled (initial value) 1: Trigger operation (counting stop/start) is enabled
7 to 2	-	Reserved bits
1 to 0	FT1RUN 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; 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.

Table 9-3 the register settings

	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:

$$T_{period} = \frac{\text{FTnP + 1}}{\text{The count clock frequency [Hz]}} \qquad \text{(FTnP: 0x0001 to 0xFFFF)}$$

6: Choice of the external output signal
FTnOSL1 and FTnOSL0 bits of FTnMOD register are used to choose output driven to the FTOn pin or FTOnN pin.
See “9.3.3. Output Control” for detail.

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 four cycle of the timer clock and the counter operation starts.
When the operation is stopped, the count operation stops and the FTnSTA bit becomes "0". Then the count value is maintained.
If started again, it restarts after four cycle.
If clearing the counter, write an arbitrary value to the FTnC register.

8: Operation process in progress

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/Clearing 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. If the FTnSTSYN bit is “1”, its counting stops at end of cycle.
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, LTB1INT to LTB3INT, TMH0INT to TMH4INT, TMHXINT or FTM0TRG to FTMnTRG.
Depending on the chosen trigger event, an operation (counter start, counter stop, counter start/stop and counter clearing) can be chosen. If the FTnSTSYN bit is “1”, its counting stops at end of cycle.

9.3.2.3 Clearing Counter

A counter can be cleared by the software or trigger-event. A clearing by software is writing any data to FTnC.
A clearing by trigger-event is done when occurred start-trigger with FTnSTC=1 or stop-trigger with FTnSPC=1. See the bit-explanation for detail condition.
Set the clear invaild section by setting the FTnDCLH, FTnDCLD.

- FTnDCLH=1, FTnDCLD=x : no clear invalid section
- FTnDCLH=1, FTnDCLD=0 : clear invalid during positive phase outputs high-level. However it’s valid during the dead-time.
- FTnDCLH=1, FTnDCLD=1 : clear invalid during positive phase outputs high-level.

9.3.3 Output Control

Two types of signals can be output from FTOn/FTOnN pins.
FTnOSL1 and FTnOSL0 bits of FTnMOD register are used to choose the phase of the output signal driven to the FTOn/FTOnN pins. FTnOSNP bit is for reversing the output to the FTMnP pin. FTnOSNN bit is for reversing the output to the FTOnN pin.
If the dead time is enabled, the "L" level output is maintained from the start of counting through the dead time period. Also a setting FTnSDN bit forces the positive/negative phase output to be "L" level. See the explain of each mode for output during operation.

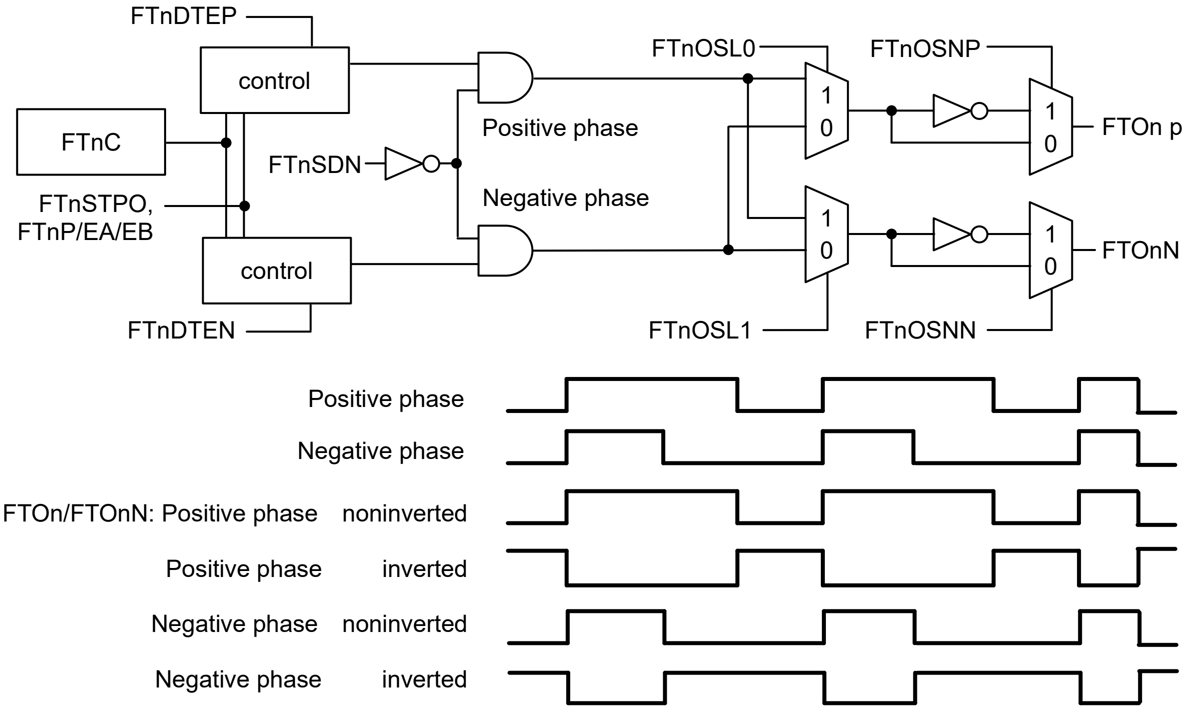
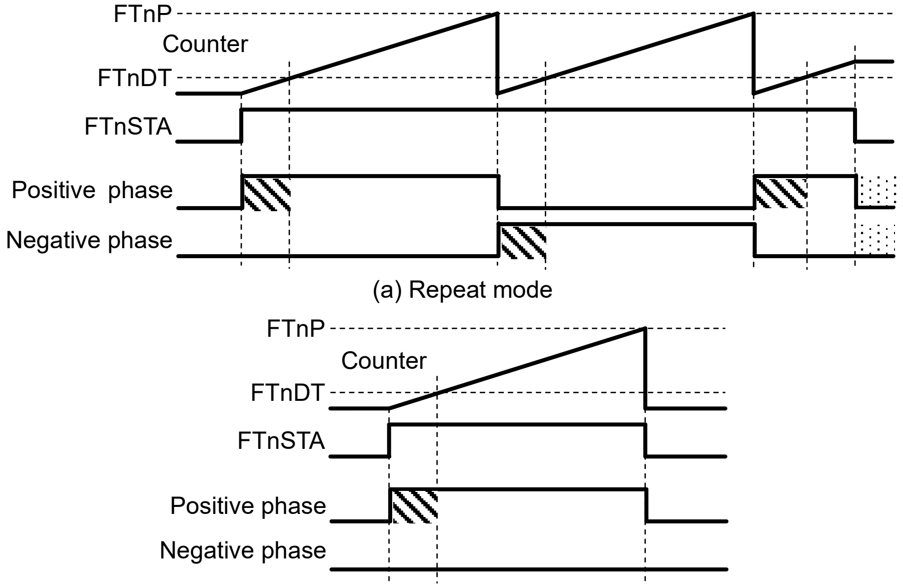


Figure 9-2 Output Control

9.3.3.1 TIMER Mode

The TIMER mode function generates interrupt of counter over-flow and controls output.
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.
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 phase output remains "H" level for one cycle of the timer and then the count stops. The negative phase output is fixed to "L".
Figure 9-3 shows waveforms of the positive phase/negative phase output.



(a) Repeat mode

(b) One-shot mode

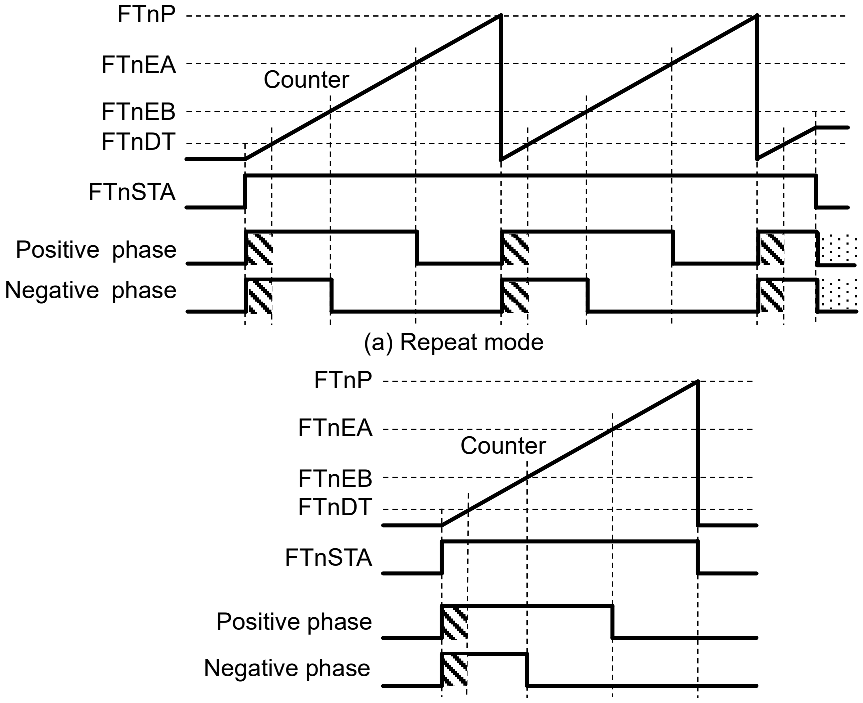
▨ L-level while the dead-time is set by FTnDT if FTnDTENP/FTnDTENN=1.

⬚ L-level if FTnSTPO=0. the level according to the counter value if FTnSTPO=1.

Figure 9-3 Waveforms of the positive phase/negative phase output in the Timer mode

9.3.3.2 PWM1 Mode

The PWM1 mode generates pulse with the cycle configured by FTnP register. The duty of positive phase output is configured by FTnEA register. The duty of negative phase output is configured by FTnEB register.
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 9-4 shows waveforms of the positive phase/negative phase output in the repeat PMW1 mode.



(a) Repeat mode

(b) One-shot mode

▨ L-level while the dead-time is set by FTnDT if FTnDTENP/FTnDTENN=1.

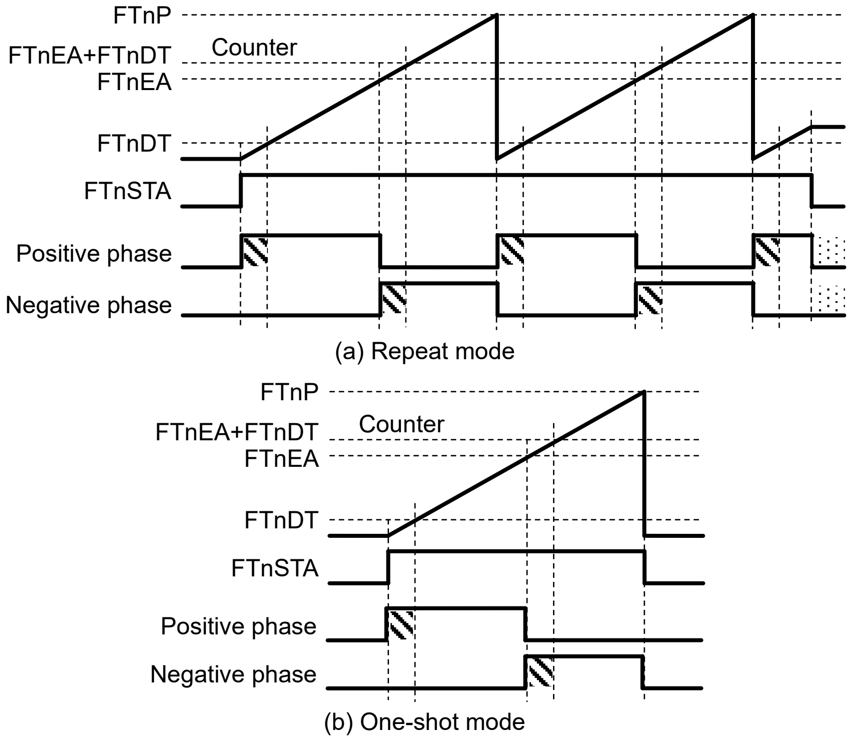
⬚ L-level if FTnSTPO=0. the level according to the counter value if FTnSTPO=1.

Figure 9-4 Waveforms of the positive phase/negative phase output in the PWM1 mode

9.3.3.3 PWM2 Mode

The PWM2 mode generates a complementary output pulse with the cycle configured by FTnP register. The duty of positive/negative phase output is configured by FTnEA register. The FTnEB register is not used.
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 9-5 shows waveforms of the positive phase/negative phase output in the repeat PMW2 mode.



(a) Repeat mode

(b) One-shot mode

▨ L-level while the dead-time is set by FTnDT if FTnDTENP/FTnDTENN=1.

░ L-level if FTnSTPO=0. the level according to the counter value if FTnSTPO=1.

Figure 9-5 Waveforms of the positive phase/negative phase output in the PWM2 mode

9.3.3.4 Output at Counter Stop

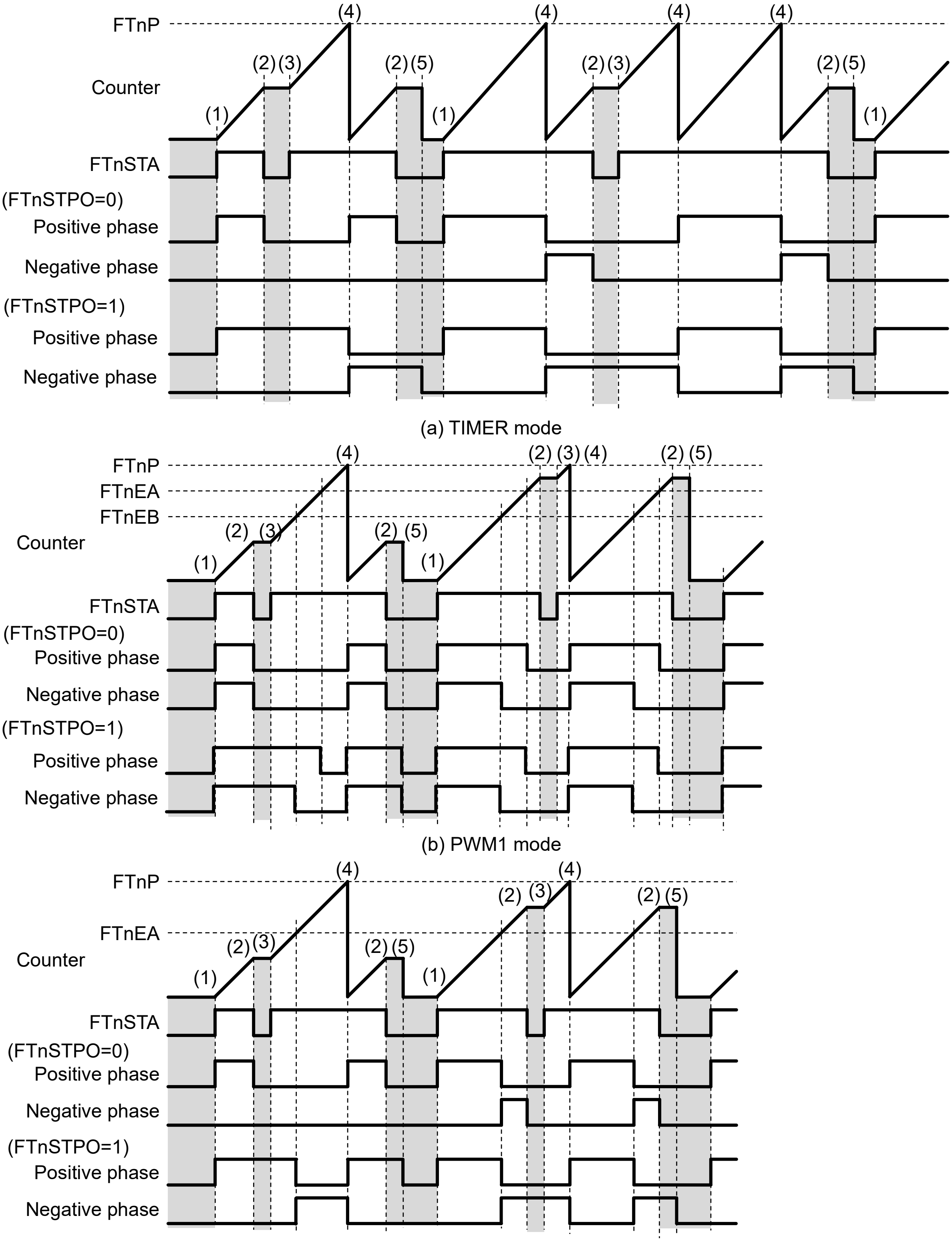
The state of Positive/Negative phase output when the counter is stopped by the software or the event trigger input is determined by the FTnSTPO bit setting of the FTnMOD register.

(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.

FTnSTPO bit of FTnMOD register is used to choose output conditions when the counter stops.
Figure 9-6 shows output waveforms of each mode when the FTnSTPO bit of the FTnMOD register is "0" / "1".



(a) TIMER mode

(b) PWM1 mode

(c) PWM2 mode

Counter operation
(1) Counter start
(2) Counter stop
(3) Counter restart without clear the counter
(4) Counter overflow
(5) Counter clear

▨ Stopping counter

Figure 9-6 Waveforms of the positive phase/negative phase output with counter stop

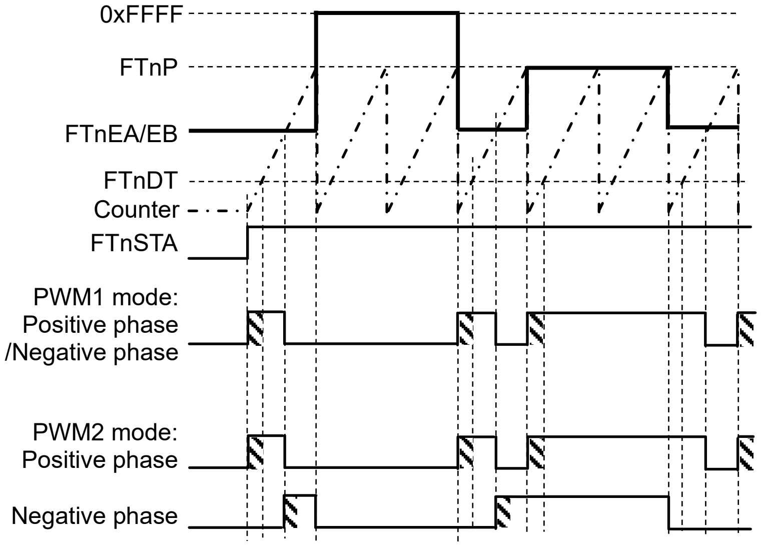
9.3.3.5 Output with DUTY=100%,0%

PWM output waveforms is configured to duty 0% and 100% by the FTnEA/EB registers setting in the PWM1/PWM2 mode.

When FTnEA/FTnEB value is the same as FTnP value, the duty is 100%. When FTnEA/FTnEB value is 0xFFFF, the duty is 0%. When both FTnP and FTnEA/EB is 0xFFFF, the duty is 0%.

When setting to the duty = 100% with deadtime enabled, the dead time apply at first cycle only.

Figure 9-7 shows output waveforms with 0% and 100% in the PWM1 mode.



▨ L-level while the dead-time is set by FTnDT if FTnDTENP/FTnDTENN=1.

Figure 9-7 Waveforms of the positive phase/negative phase output with the duty is 0% and 100%

9.3.4 CAPTURE Mode

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 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 9-8 shows the time chart in this example.

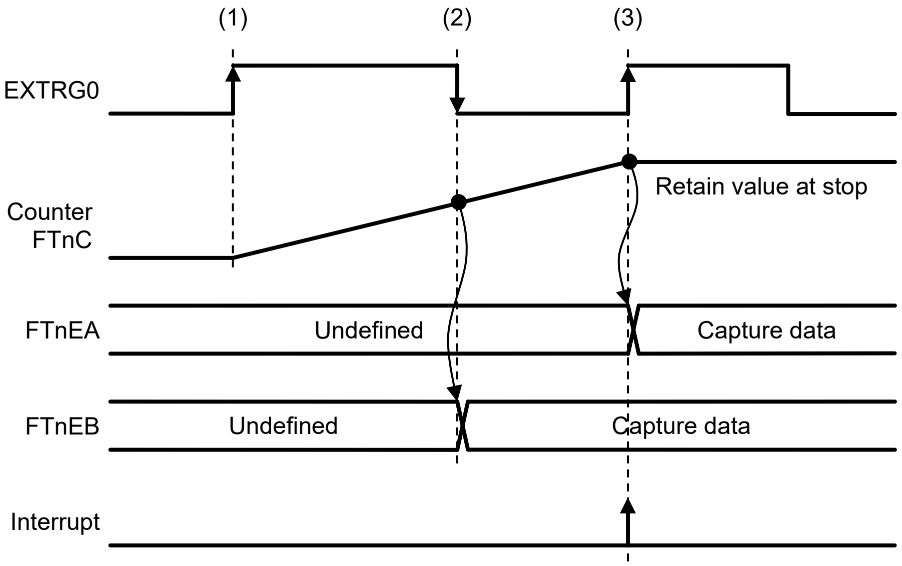


Figure 9-8 Operation Example to measure cycle and duty of PWM signal (one cycle)

(1) The counter starts operating at the rising edge of the signal input from the EXTRG0 pin.
(2) The value of the FTMn counter register FTnC is stored into the FTnEB register at the falling edge of the EXTRG0 pin.
(3) The value of the FTMn counter register FTnC is stored into the FTnEA register at the rising edge of the EXTRG0 pin. The counter stop the operation and the interrupt is generated.
(4) The counter stops and the interrupt occurs.

The value of the FTnEA register corresponds to the cycle of the PWM signal input from the 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 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 FTnSTC bit to "1" to enable the counter clear when the trigger event of counter start 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 9-9 shows the time chart in this example.

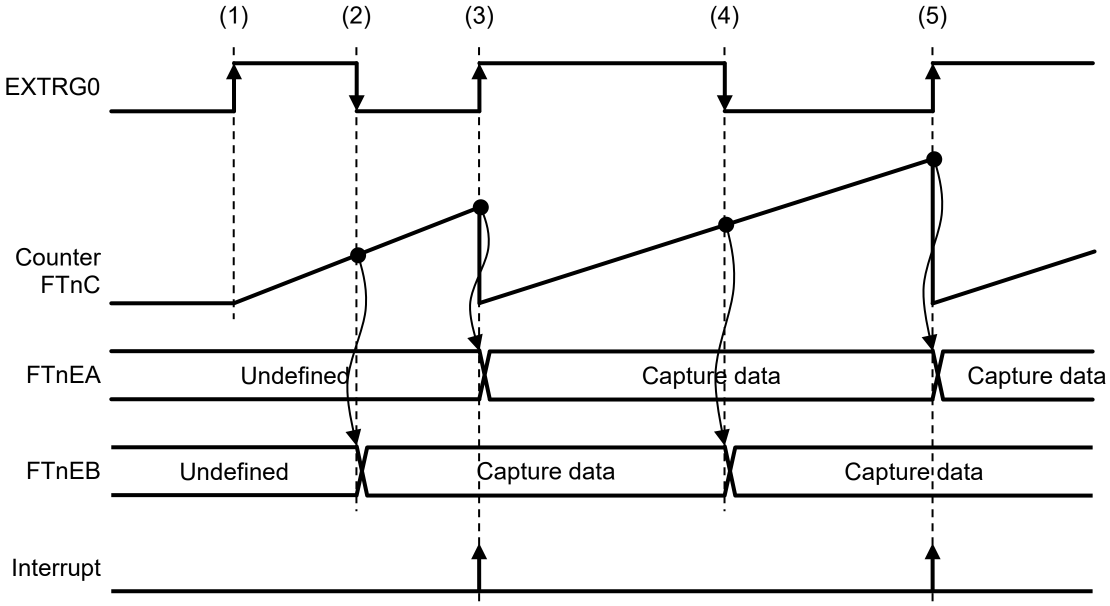


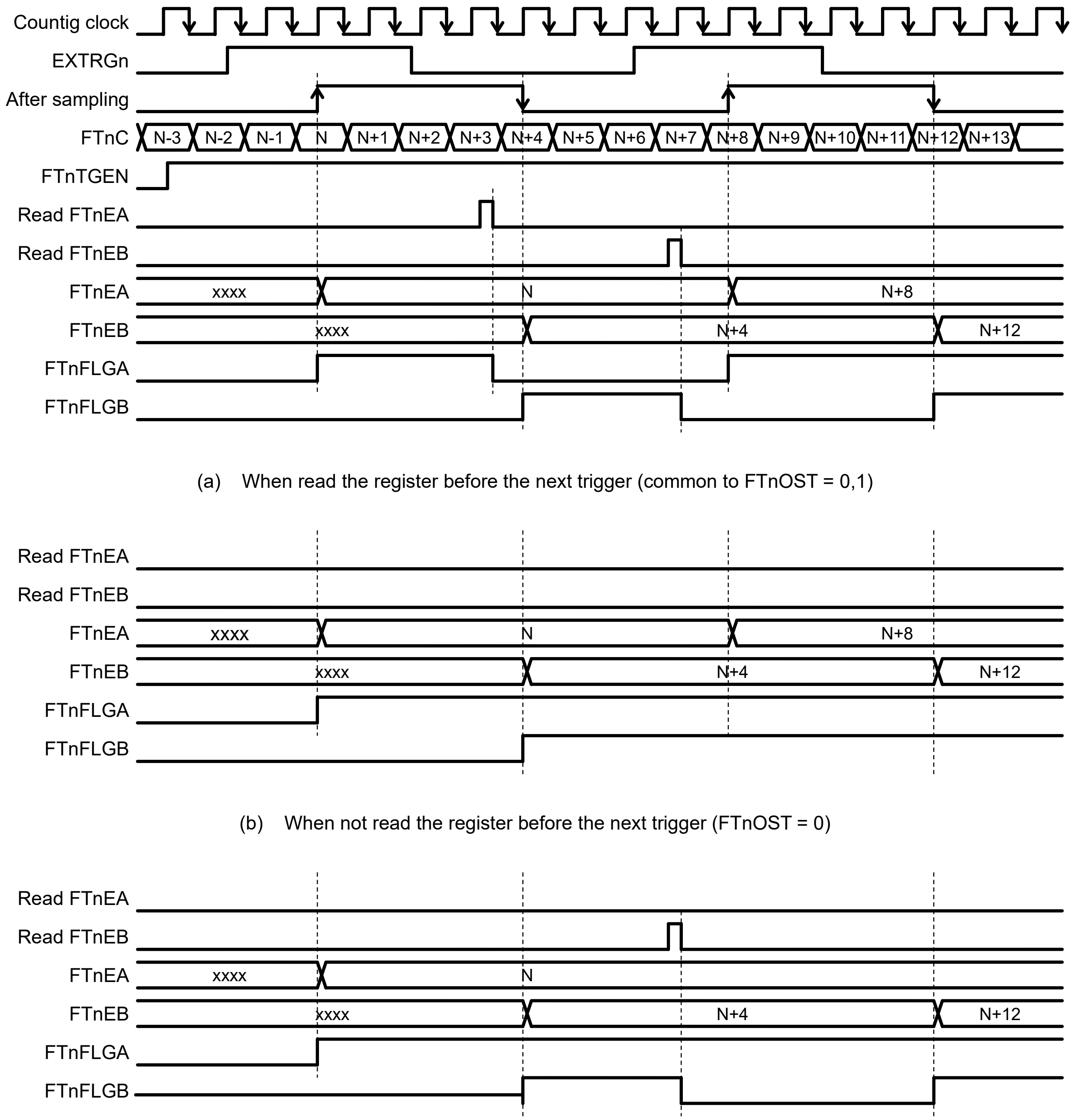
Figure 9-9 Operation Example to measure cycle and duty of PWM signal (repeat cycle)

(1) The counter starts operating at the rising edge of the signal input from the EXTRG0 pin.
(2) The value of the FTMn counter register; FTnC is stored into the FTnEB register at the falling edge of the EXTRG0 pin.
(3) The value of the FTnC is stored into the FTnEA register at the rising edge of the EXTRG0 pin. The counter is cleared and the interrupt is generated. The count operation continues.
(4) The value of the FTnC is stored into the FTnEB register at the falling edge of the EXTRG0 pin.
(5) The value of the FTnC is stored into the FTnEA register at the rising edge of the 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 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 9-10 Operation Timing in CAPTURE Mode

9.3.4.2 Clearing FTnFLGA/FTnFLGB bit

In single mode(FTnOST=1), FTnEA/FTnEB data is not updated if FTnFLGA/FTnFLGB is 1 each.
The FTnFLGA/FTnFLGB is cleared by reading FTnEA/FTnEB, respectively. However it is invalid when FTnTGEN=0.
Even if FTnFLGA/FTnFLGB is cleared and FTnTGEN is set to 0, a trigger may be entered during control and FTnFLGA/FTnFLGB may be set to 1.

To avoid this, set FTnTGEN to 0 after making sure that FTnFLGA/FTnFLGB is 0 with no trigger input. For example, set FTnST to 0 and stop the trigger start. Or initialize this peripheral circuit by block reset after setting FTnTGEN to 0.

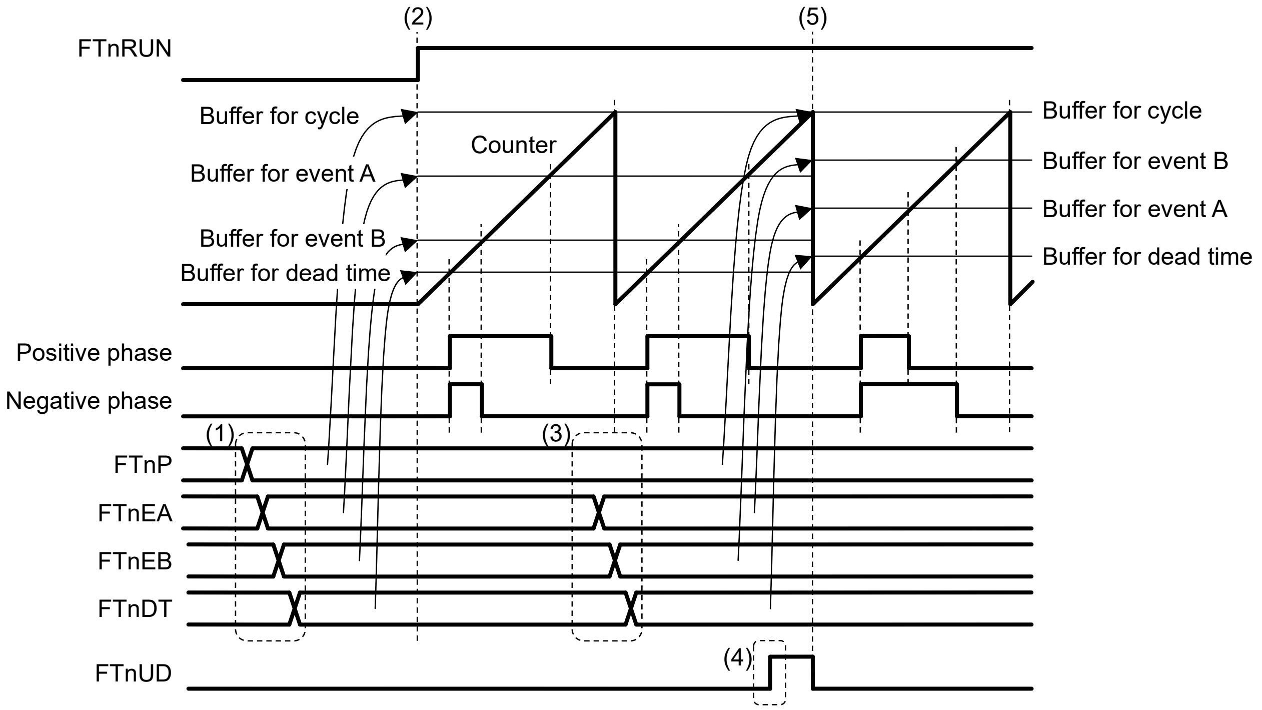
9.3.5 Changing Cycle, Event A/B, and Dead Time during Operation

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.

Table 9-4 Update timing

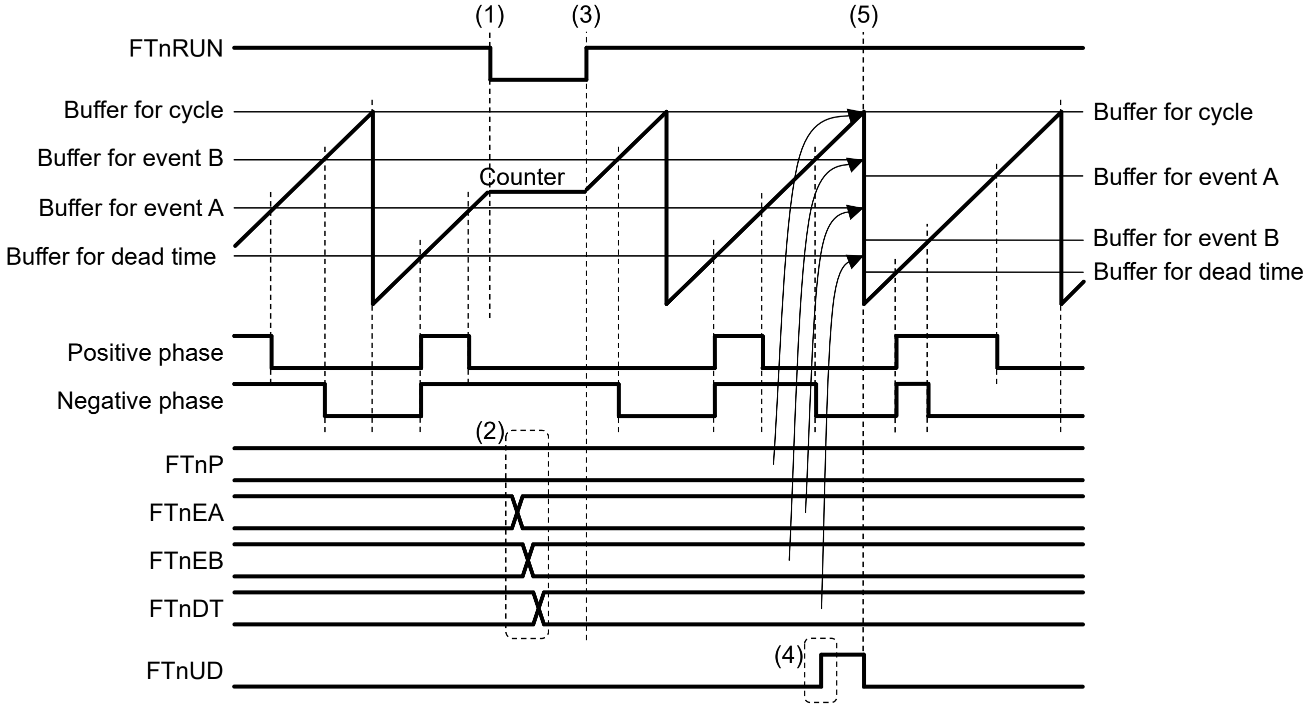
Counter operation status when setting the register	Counter value when setting the register	Update timing
Stop	0x0000	Updated at the counter start
Stop	Other than 0x0000	Updated at the start of cycle while the counter has been restarting and FTCUDn bit is set to "1".
Operating	Any value	Updated at the start of cycle while the counter is operating and FTCUDn bit is set to "1".
Trigger clear	0x0000	Cleared with trigger when the counter is running and FTCUDn =1.

Figure 9-11 shows the operating waveforms when the registers are updated while the counter stops (counter value is 0x0000) or the counter is operating.
Figure 9-12 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 9-11 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 9-12 Update timing while the counter stops(counter value is other than 0x0000)

9.3.6 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 or trigger for capture. The trigger source can be chosen from EXTRG0 to EXTRG7, LTB1INT to LTB3INT, TMH0INT to TMH4INT, TMHXINT, FTMnTRG or RC1K.
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 and EXTRG4.

The EXTRG0 to EXTRG7 are output of sampling controller of the external interrupt function. They are connected to functional timer as event trigger or external clock input.
These input signals are delayed 2 clocks of the timer clock.

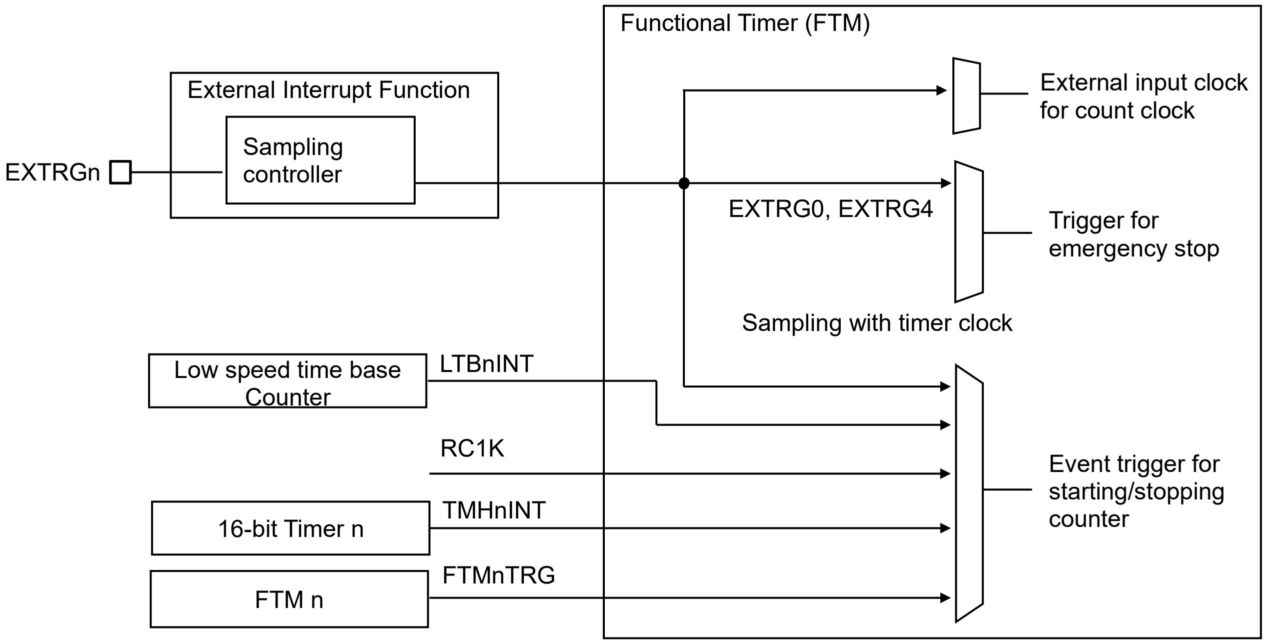

Functional Timer (FTM)
External Interrupt Function
Sampling controller
EXTRGn
External input clock for count clock
EXTRG0, EXTRG4
Trigger for emergency stop
Sampling with timer clock
Low speed time base Counter
LTBnINT
RC1K
16-bit Timer n
TMHnINT
FTM n
FTMnTRG
Event trigger for starting/stopping counter

Figure 9-13 Input Path of Trigger Signal

9.3.6.1 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, LTB1INT to LTB3INT, TMH0INT to TMH4INT, TMHXINT, FTMnTRG)

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 and count clock in the FTnCLK register.

Once the configuration above is completed, control the counter by the FTCSTR register. The procedure is as follows:

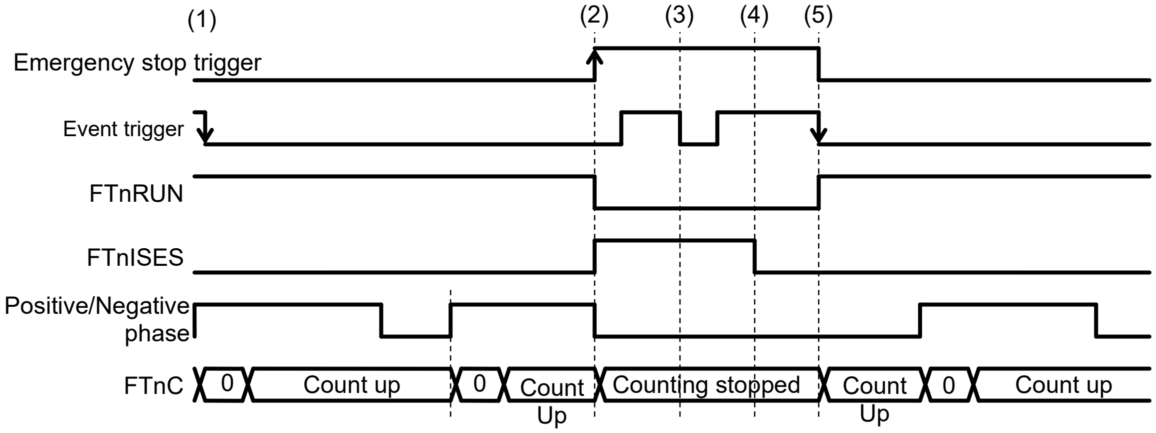
(1) Make the waiting state for an event trigger
Write "1" to the FTnETG bit to make the waiting state for an event trigger. If the level setting is applied for trigger start and the level is applicable, the counter operation is started as soon as the FTnTGEN bit of the FTCSTAT register becomes "1".

(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.

9.3.6.2 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.

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 9-14 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 9-14 Operation Timing Diagram at Emergency Stop

Once the emergency stop occurs, the counter is stopped after two clock of the timer clock, and the FTnISES bit of the FTnINTS register becomes "1" (see (2) in Figure 9-14).
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 9-14), counting up is restarted (see (5) in Figure 9-14).
To restart the counting operation by the software, make sure that the FTnISES bit becomes "0".

9.3.7 Interrupt

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.

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.

Table 9-5 Interrupt status

Name	Status	Mode	How to set	How to clear
Cyclic interrupt	FTnISP bit	All modes	When FTnC = FTnP	Write "1" to FTnICP bit
Event timing A interrupt	FTnISA bit	TIMER/PWM1/ PWM2	When FTnC = FTnEA	Write "1" to FTnICA bit
	FTnISA bit	CAPTURE	When stored capture data into FTnEA	Write "1" to FTnICA bit, or read the FTnEA register
Event timing B interrupt	FTnISB bit	TIMER/PWM1	When FTnC = FTnEB	Write "1" to FTnICB bit
	FTnISB bit	CAPTURE	When stored capture data into FTnEB	Write "1" to FTnICB bit, or read the FTnEB register
Trigger stop interrupt	FTnISTS bit	All modes	Counter stop/clear by trigger-stop event	Write "1" to FTnICTS bit
Trigger start interrupt	FTnISTR bit	All modes	Counter start/clear by trigger-start event	Write "1" to FTnICTR bit
Emergency stop interrupt	FTnISES bit	All modes	Occurring emergency stop	Write "1" to FTnICES bit

The cyclic interrupt/event timing A interrupt/event timing B interrupt can be chosen as the interrupt trigger output.

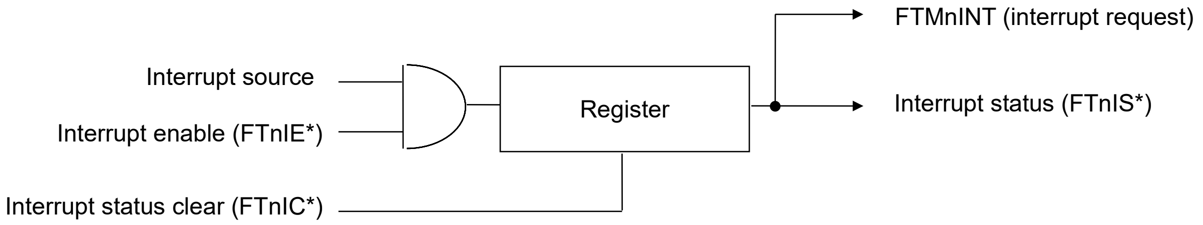

FTMnINT (interrupt request)
Interrupt source
Register
Interrupt status (FTnIS*)
Interrupt enable (FTnIE*)
Interrupt status clear (FTnIC*)

Figure 9-15 Interrupt Control Signal

Chapter 10 Watchdog Timer

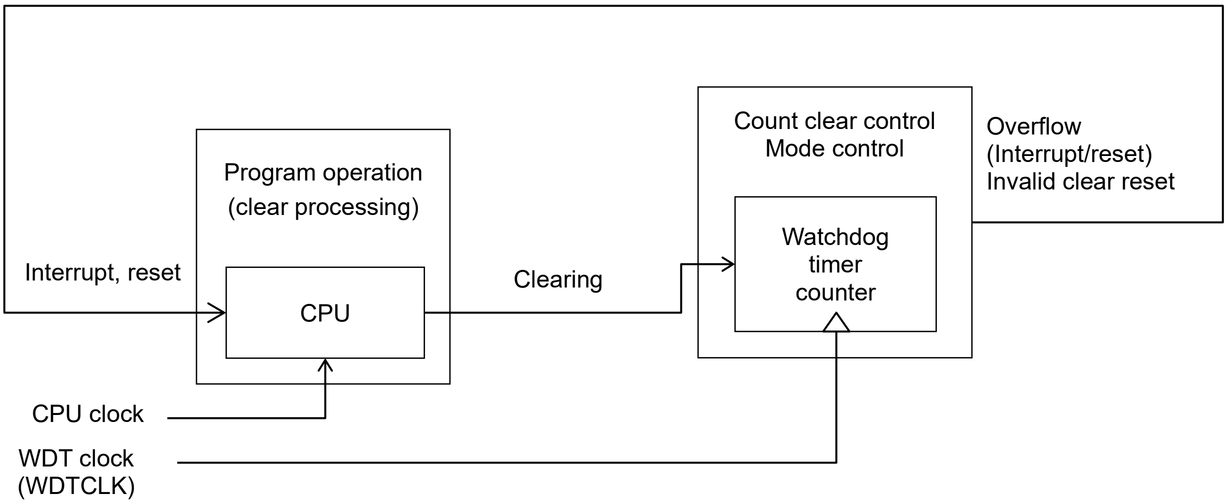
10. Watchdog Timer

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.



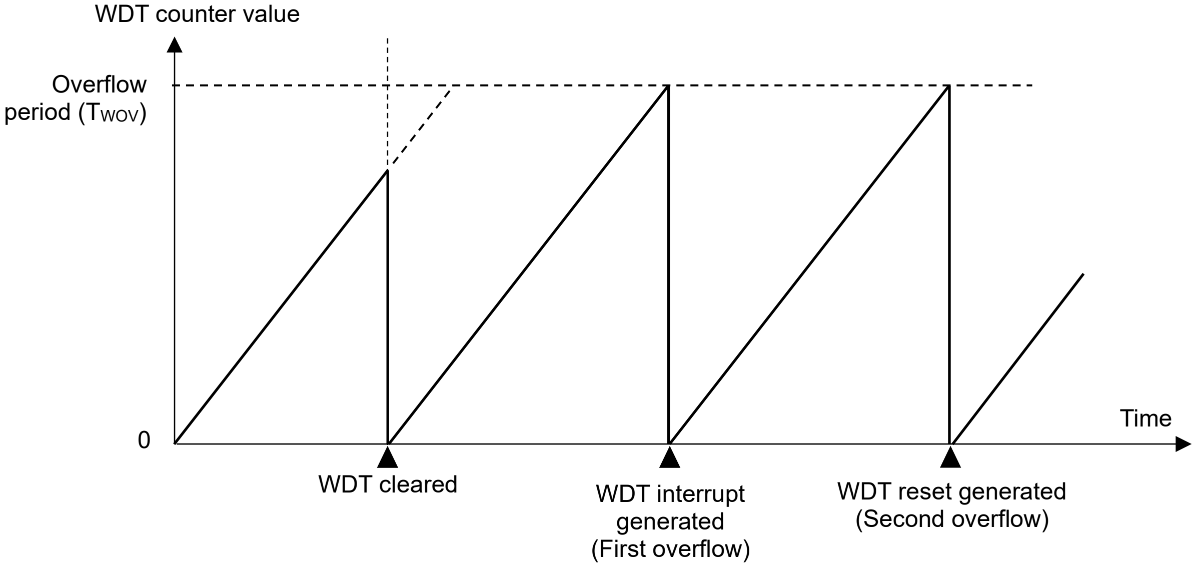


Figure10-1 Watchdog Timer Overview (With the Window Function Disabled)

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

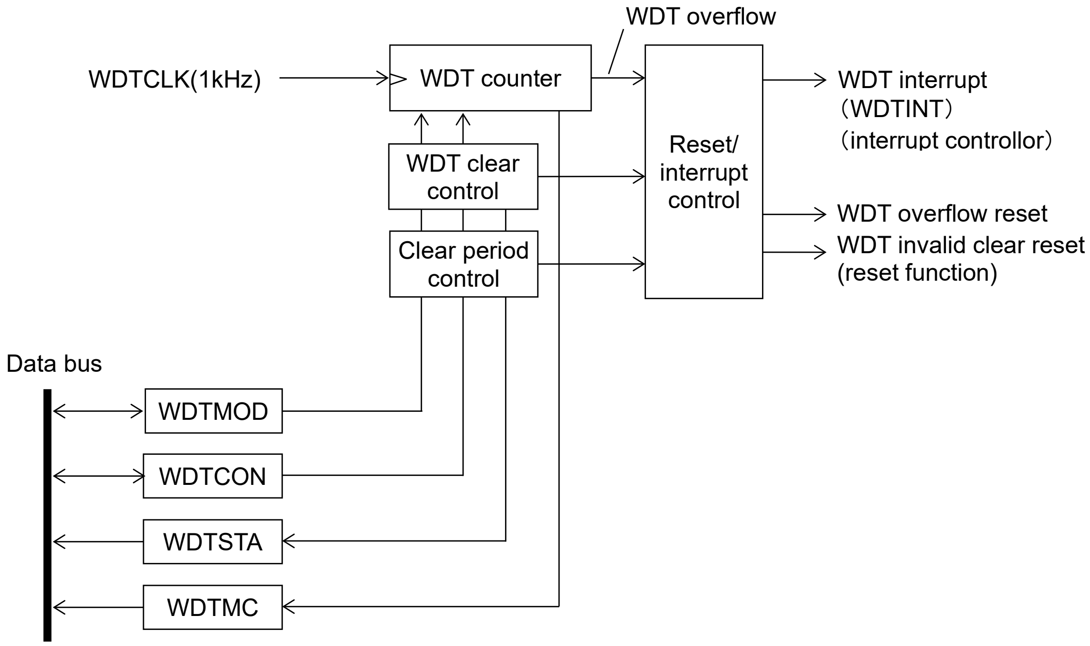
- The following items can be chosen by the code option. See the Chapter 30 "Code Option" for details of the code option.
  - Enabling/disabling the WDT timer operation
  - Operation clock of the WDT counter (32 dividing of low-speed clock LSCLK, WDTCLK RC1K oscillation)
- Using independent internal RC oscillating clock; a frequency accuracy is ± 10%

[Note]

- **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.**

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 (WDTCON)
Access:	R/W
Access size:	8 bit
Initial value:	0x00

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

Bit No.	Bit symbol name	Description
7 to 0	d7 to d0	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 SFR to set the overflow period and the clear enabled period of the WDT counter.

Address:	0xF012 (WDTMOD)
Access:	R/W
Access size:	8 bit
Initial value:	0x06

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								WDTMOD							
Bit	-	-	-	-	-	-	-	-	-	-	WOVF1	WOVF0	-	WDT2	WDT1	WDT0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit No.	Bit symbol name	Description
7, 6, 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 where frequency of WDTCLK is 1.024kHz.

[Note]

- **See the data-sheet for frequency accuracy of RC1K.**

10.2.4 Watchdog Timer Counter Register (WDTMC)

This register is a SFR to read the WDT counter value.

Address: 0xF014 (WDTMCL/WDTMC), 0xF015 (WDTMCH)
Access: R
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	WDTMC															
Byte	WDTMCH								WDTMCL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 0	d15 to d0	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.

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 (WDTSTA)
Access: R
Access size: 8 bit
Initial value: 0x01

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

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	WDTWIN	This bit is used to indicate the status of enabling/disabling WDT counter clearing. It is always “1” when a window function is disabled 0: Clearing is disabled. 1: Clearing is enabled (initial value).

10.3 Description of Operation

The WDT counter starts counting 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 30 "Code Option" for details on how to set the code option.

- Enabling/disabling the WDT timer operation
- Enabling/disabling the WDT timer operation in the HALT/HALT-H mode
- Enabling/disabling the WDT timer operation in the HALT-D mode

10.3.1 How to Clear WDT Counter

The WDT counter can be cleared by writing "0x5A" to the WDTCON register with the WDP bit set to "0", then writing "0xA5" to the WDTCON register with the WDP bit set to "1" while WDT counter clearing is enabled.
The WDP bit is reset to "0" when the system is reset as well as when the WDT counter overflows. It is reversed every time data is written to the WDTCON register.

The following diagram shows the WDT counter clearing timing chart.

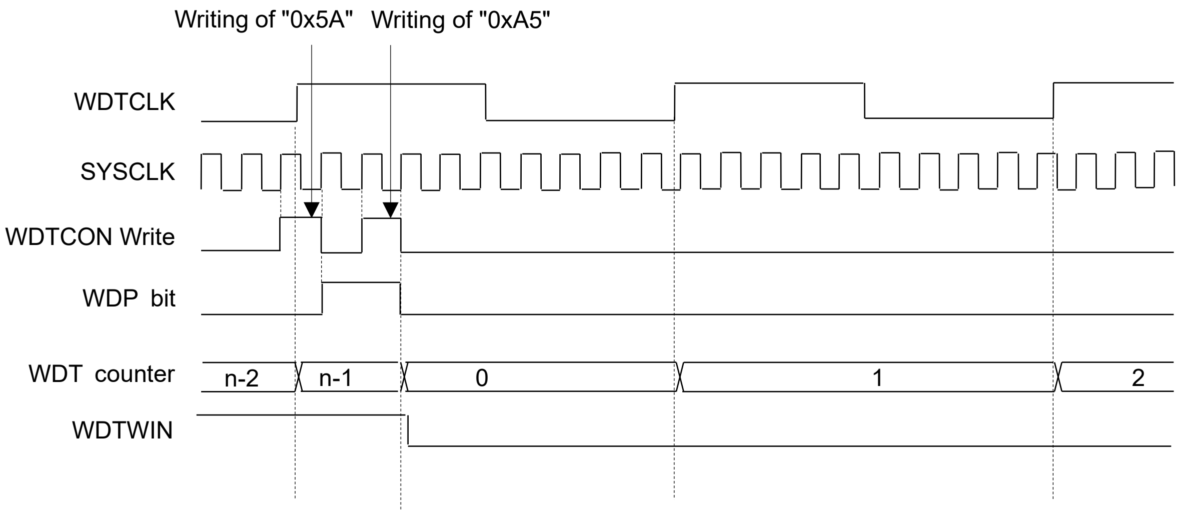


Figure 10-3 WDT Counter Clearing Timing Chart

The following description shows a sample program script of the watchdog timer.

```
void wdt_clear( void )
  {
    unsigned char pswval;

    if(WDTCLR1 == 1) {                    // Checking presence of pending clearing process
      return;
    };
    if(WDTCLR2== 1) {                     // Checking whether clearing process is pending or completed
      return;
    };

    pswval = s_drvcommon_getPSW();      // Saving PSW
    _DI();                              // Interrupt disabled (clearing MIE bit)
    do {
         WDTCON = 0x5A;                   // WDT counter clearing
      } while (WDP != 1 );
    WDTCON = 0xA5;
    if ((pswval & 0x08) != 0) {           // Confirming MIE bit
      _EI();                              // Interrupt enabled (setting MIE bit)
    }

    static unsigned char s_drvcommon_getPSW( void ){
      #pragma asm
      mov r0,psw
      rt
      #pragma endasm
    }
```

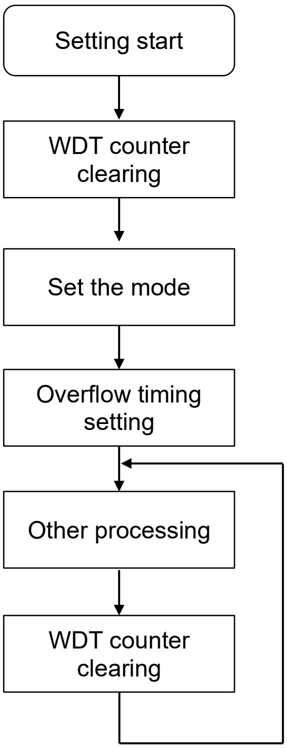
Figure 10-4 Sample Program Script of Watchdog Timer

[Note]

- **In the STOP/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.



Counter clearing before initial mode setting

Set through WOVF1 and WOVF0 bits of WDTMOD register

Set through WDT2 to WDT0 bits of WDTMOD register

Writing "0x5A" with the WDP bit of WDTCON register set to "0", then writing "0xA5" with the WDP bit set to "1" enables WDT counter to be cleared

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.

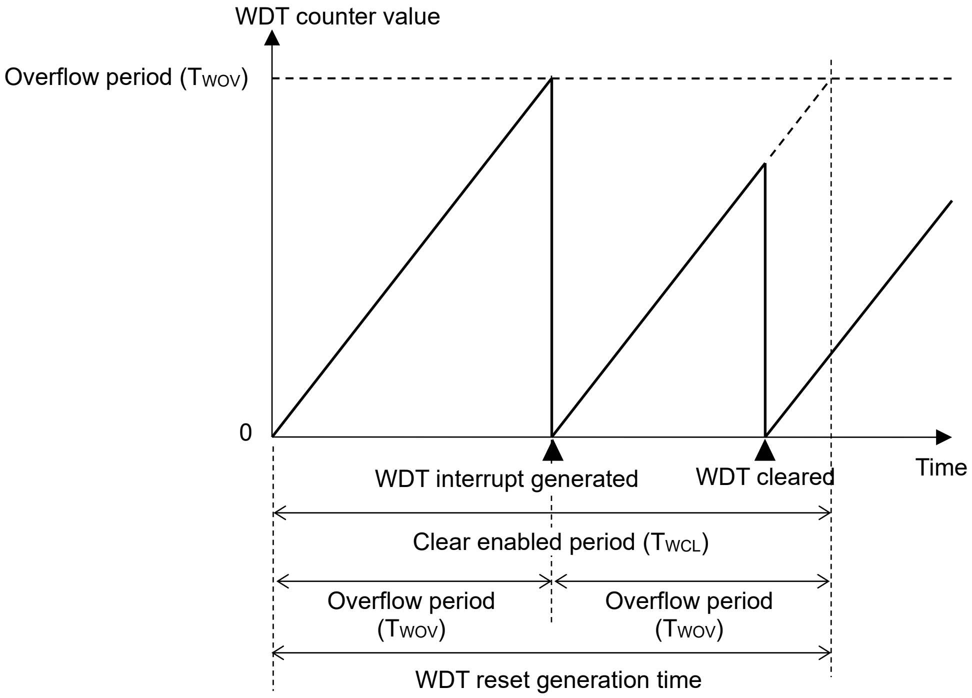

WDT counter value
Overflow period (T_{WOV})
0
Time
WDT interrupt generated
WDT cleared
Clear enabled period (T_{WCL})
Overflow period (T_{WOV})
Overflow period (T_{WOV})
WDT reset generation time

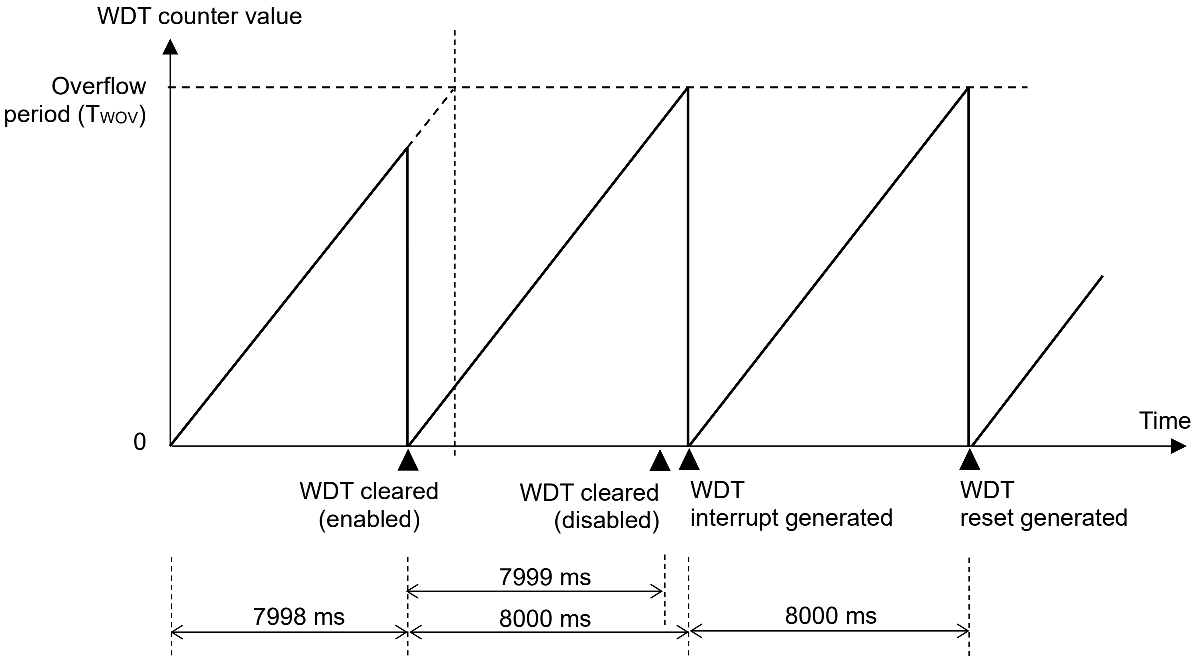
Figure 10-6 Overview of Operation Timing in Window Function Disabled Mode

The following table shows the WDT counter clear enabled period in the window function disabled mode.

Table 10-3 WDT Counter Clear Enabled Period in Window Function Disabled Mode

WDT2	WDT1	WDT0	Overflow period (T_{WOV})*1	WDT reset generation time*1	WDT counter clear enabled period (T_{WCL})*1
0	0	0	7.8 ms	15.6 ms	≈ Overflow period
0	0	1	15.6 ms	31.3 ms	≈ Overflow period
0	1	0	31.3 ms	62.5 ms	≈ Overflow period
0	1	1	62.5 ms	125 ms	≈ Overflow period
1	0	0	125 ms	250 ms	≈ Overflow period
1	0	1	500 ms	1000 ms	≈ Overflow period
1	1	0	2000 ms	4000 ms	≈ Overflow period
1	1	1	8000 ms	16000 ms	≈ Overflow period

*1: where the WDTCLK frequency is 1.024kHz(typ.) that is not included a significant error.



Design the WDT clear timing with time to spare.

Figure 10-7 Example of Operation Timing in Window Function Disabled mode (When Overflow Period=8000 ms)

The following figure shows details of operation timing in the window function disabled mode.

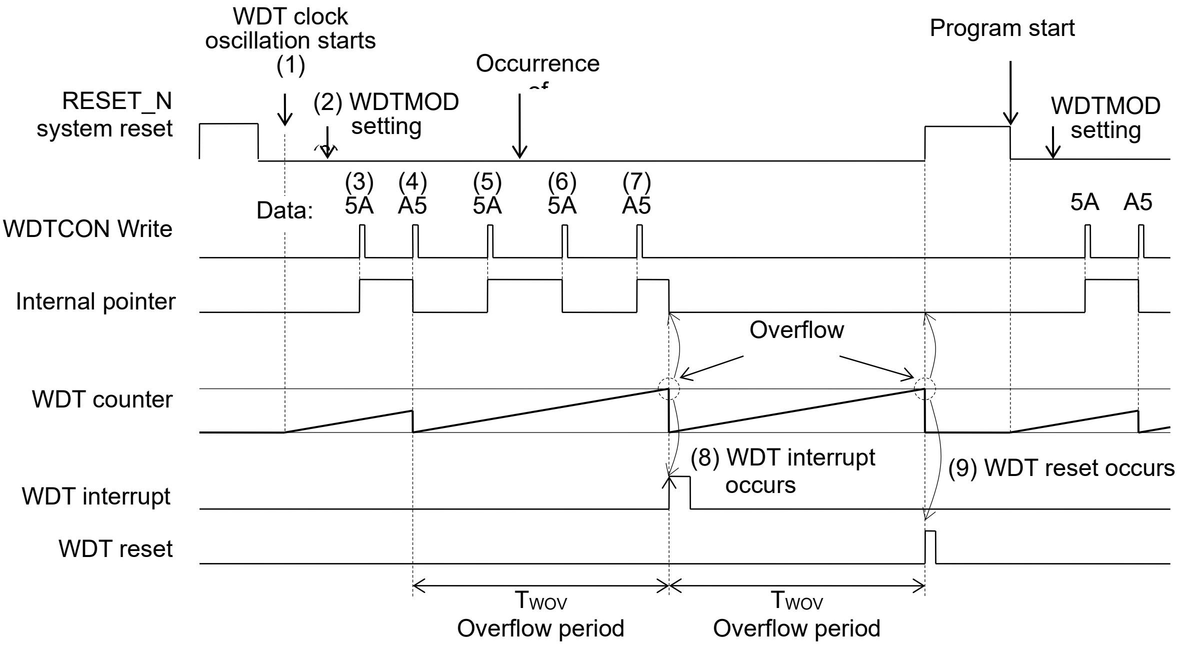


Figure 10-8 Details of Operation Timing in Window Function Disabled Mode

(1) After the system reset is released, the WDT counter starts counting up.
(2) The WDT counter overflow period (TWOV) is set to the WDTMOD register.
(3) "0x5A" is written to the WDTCON register. (Internal pointer WDP: 0 →1)
(4) "0xA5" is written to the WDTCON register to clear the WDT counter. (Internal pointer WDP: 1→0)
(5) "0x5A" is written to the WDTCON register. (Internal pointer WDP: 0 →1)
(6) When "0x5A" is written to the WDTCON register after an abnormality occurred, it is not accepted because the internal pointer WDP is "1". (Internal pointer WDP: 1 →0)
(7) Although "0xA5" is written to the WDTCON register, the WDT counter is not cleared because the internal pointer WDP is "0" and writing of "0x5A" is not accepted in (6). (Internal pointer WDP: 0 →1)
(8) The WDT counter overflows and a WDT interrupt request is generated. (Internal pointer WDP: 1 →0)
Following cleared due to the overflow, the WDT counter continues counting up.
(9) If the WDT counter is not cleared even by the software processing after the WDT interrupt and it overflows again, a WDT reset occurs and the shift to the system reset mode takes place.

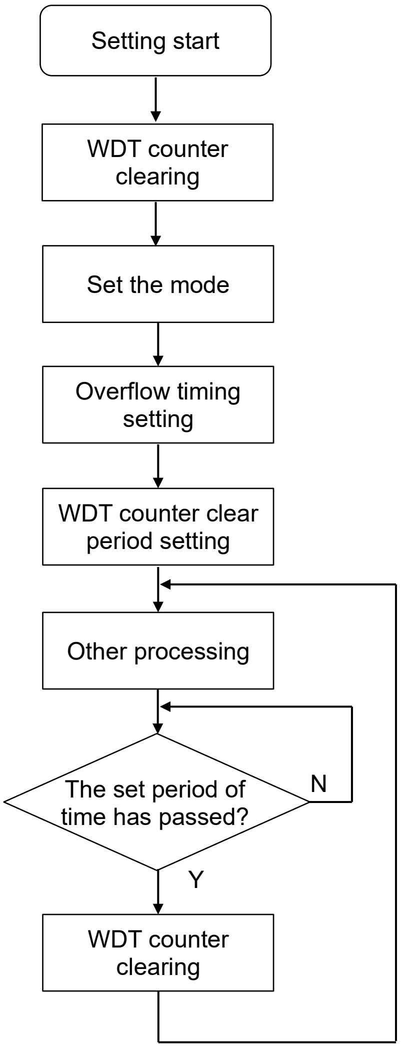
10.3.3 Window Function Enabled Mode

In the window function enabled mode, if the WDT counter is not available to clear within the WDT clear enabled period and the counter overflows first time, the WDT overflow reset is generated.
In addition, if the WDT counter is cleared in the period the counter clear is not enabled, the WDT invalid clear reset is generated.

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)



Counter clearing before initial mode setting

Set through WOVF1 and WOVF0 bits of WDTMOD register

Set through WDT2 to WDT0 bits of WDTMOD register

In order to avoid clearing during the counter clear disabled period, determine the WDT counter clearing period using the timer, etc.

Writing "0x5A" with the WDP bit of WDTCON register set to "0", then writing "0xA5" with the WDP bit set to "1" enables WDT counter to be cleared

Figure 10-9 Procedure to Use WDT (in Window Function Enabled Mode)

Overviews of the operation of each mode are shown below.

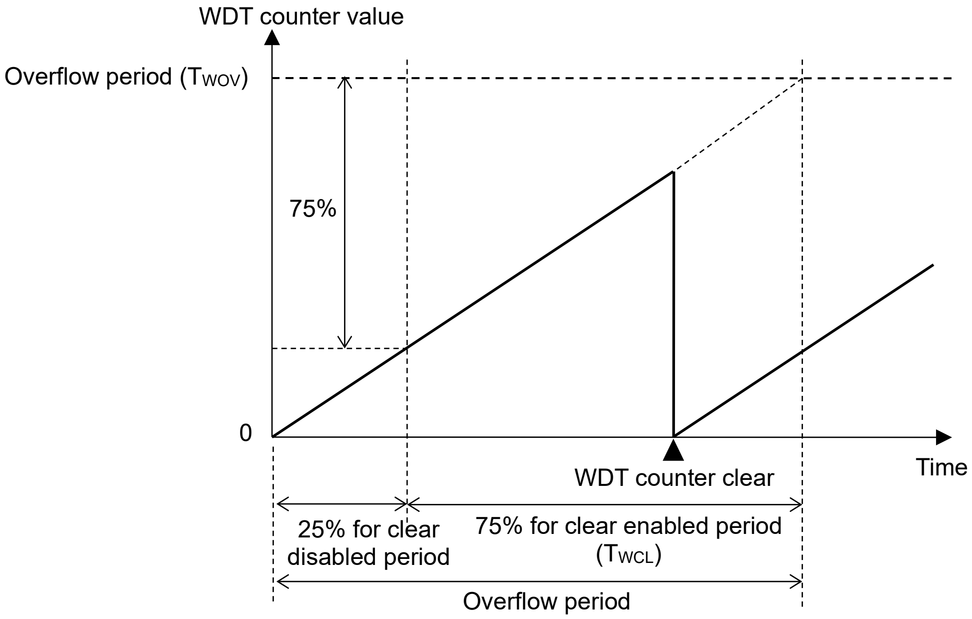


Figure 10-10 Window Function Enabled Mode 1 Operation Overview

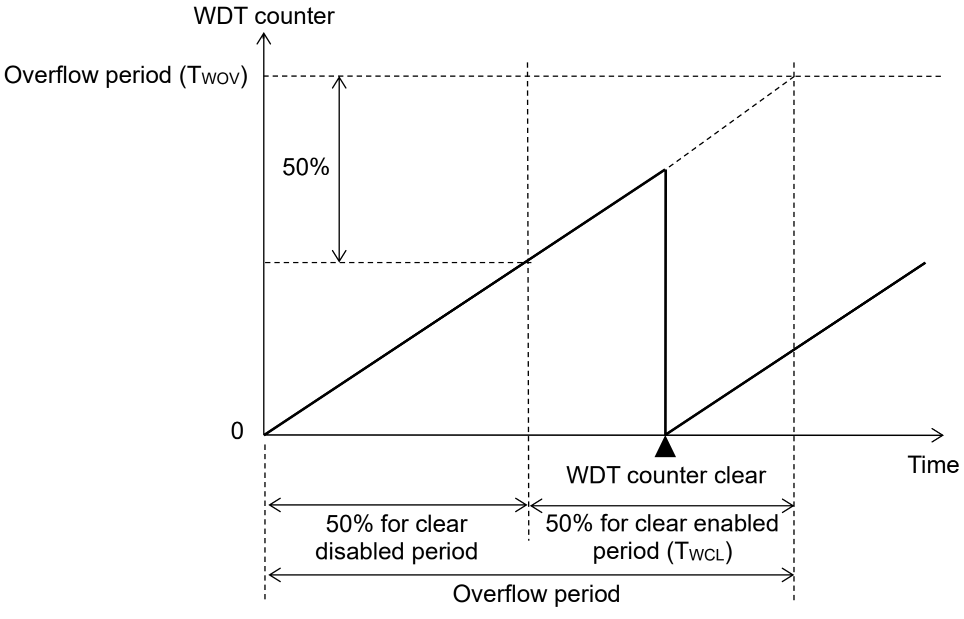


Figure 10-11 Window Function Enabled Mode 2 Operation Overview

The following table shows WDT counter clear enabled periods.
If the overflow period of the WDT counter is set to 62.5 ms or less in WDT2 to 0 bits, the window function is disabled regardless of setting values of WOVF1 and WOVF0 bits.

Table 10-4 WDT Clear Enabled Period in Window Function Enabled Mode 1

WDT2	WDT1	WDT0	Overflow period (T_{WOV})[*1]	WDT reset generation time[*1]	WDT clear enabled period (T_{WCL})[*1]
0	0	0	Approx. 7.8 ms	Approx. 7.8 ms	≈ Overflow period
0	0	1	Approx. 15.6 ms	Approx. 15.6 ms	≈ Overflow period
0	1	0	Approx. 31.3 ms	Approx. 31.3 ms	≈ Overflow period
0	1	1	Approx. 62.5 ms	Approx. 62.5 ms	≈ Overflow period
1	0	0	Approx. 125 ms	Approx. 125 ms	≈ 75% of overflow period
1	0	1	Approx. 500 ms	Approx. 500 ms	≈ 75% of overflow period
1	1	0	Approx. 2000 ms	Approx. 2000 ms	≈ 75% of overflow period
1	1	1	Approx. 8000 ms	Approx. 8000 ms	≈ 75% of overflow period

[*1]: where the WDTCLK frequency is 1.024kHz(typ.) that is not included a significant error.

Table 10-5 WDT Counter Clear Enabled Period in Window Function Enabled Mode 2

WDT2	WDT1	WDT0	Overflow period (T_{WOV})[*1]	WDT reset generation time[*1]	WDT clear enabled period (T_{WCL})[*1*2]
0	0	0	Approx. 7.8 ms	Approx. 7.8 ms	≈ Overflow period
0	0	1	Approx. 15.6 ms	Approx. 15.6 ms	≈ Overflow period
0	1	0	Approx. 31.3 ms	Approx. 31.3 ms	≈ Overflow period
0	1	1	Approx. 62.5 ms	Approx. 62.5 ms	≈ Overflow period
1	0	0	Approx. 125 ms	Approx. 125 ms	≈ 50% of overflow period
1	0	1	Approx. 500 ms	Approx. 500 ms	≈ 50% of overflow period
1	1	0	Approx. 2000 ms	Approx. 2000 ms	≈ 50% of overflow period
1	1	1	Approx. 8000 ms	Approx. 8000 ms	≈ 50% of overflow period

[*1]: where the WDTCLK frequency is 1.024kHz(typ.) that is not included a significant error.

[Note]

- **When using the window function enabled mode, always define a WDT interrupt function even though no WDT interrupt occurs.**

The following figure shows details of operation timing in the window function enabled mode.

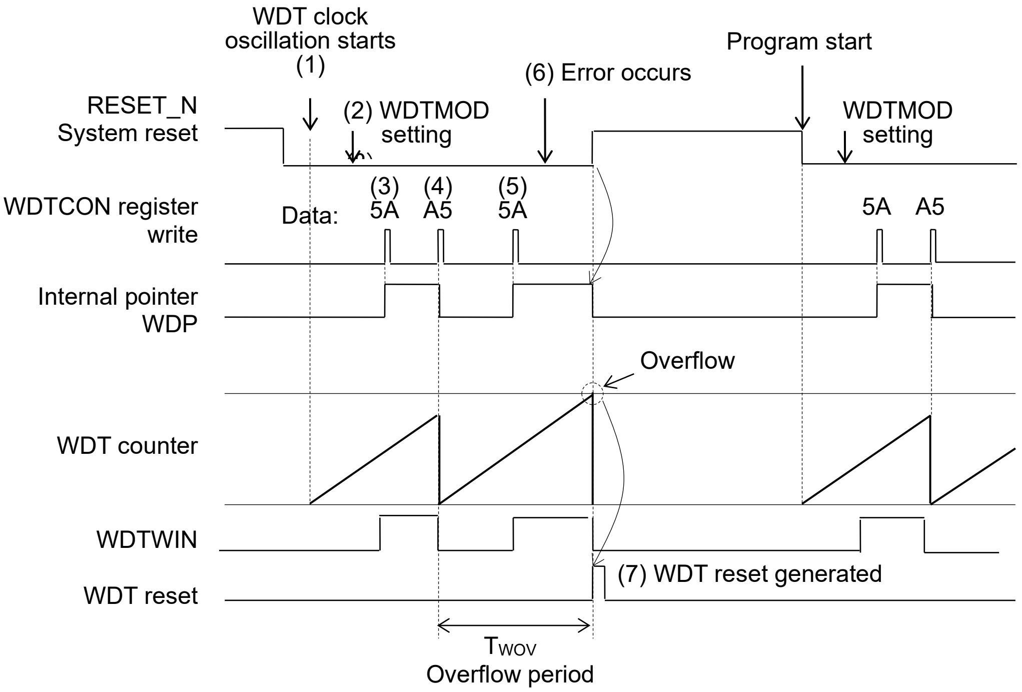


Figure 10-12 Details of Operation Timing in Window Function Enabled Mode

(1) After the system reset is released, the WDT counter starts counting up.
(2) The WDTMOD register is set with the WDT counter overflow period (T_{WOV}) and WDT clear enabled period.
(3) "0x5A" is written to WDTCON during the WDT clear enabled period. (Internal pointer WDP: 0 →1)
(4) "0xA5" is written to the WDTCON register to clear the WDT counter. (Internal pointer WDP: 1 →0)
(5) "0x5A" is written to WDTCON during the WDT clear enabled period. (Internal pointer WDP: 0 →1)
(6) Occurrence of abnormality
(7) The WDT counter overflows and a WDT reset occurs. (Internal pointer WDP: 1 →0)

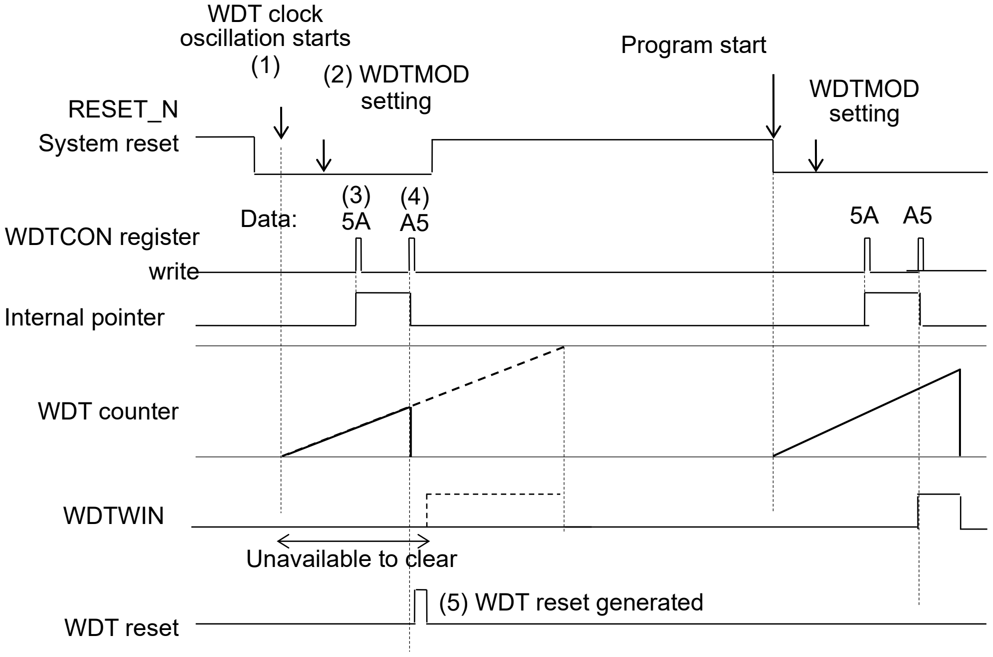


Figure 10-13 WDT invalid clear reset

(1) After the system reset is released, the WDT counter starts counting up.
(2) The WDTMOD register is set with the WDT counter overflow period (T_{WOV}) and WDT clear enabled period.
(3) "0x5A" is written to WDTCON. (Internal pointer WDP: 0 →1)
(4) "0xA5" is written to the WDTCON register to clear the WDT counter. (Internal pointer WDP: 1 →0)
(5) WDT invalid clear reset is occurred by clear processing during the WDT clear disabled period.

[Note]

- **In the watchdog timer (WDT) interrupt function, as the interrupt level (ELEVEL) of the CPU program status word (PSW) becomes "2", the WDT counter is unable to get cleared. Clear the WDT when the ELEVEL is "0" or "1". It is recommended that the WDT counter is cleared at one place in the main loop of the program as a fail-safe.**

Chapter 11 Synchronous Serial Port (SSIO)

11. Synchronous Serial Port

11.1 General Description

ML62Q2500 group has 8-bit/16-bit synchronous serial port (SSIO).
Table 11-1 shows the number of channels.

Table11-1 Number of SSIO channels

Channel no. (n)	ML62Q2500 group
0	●

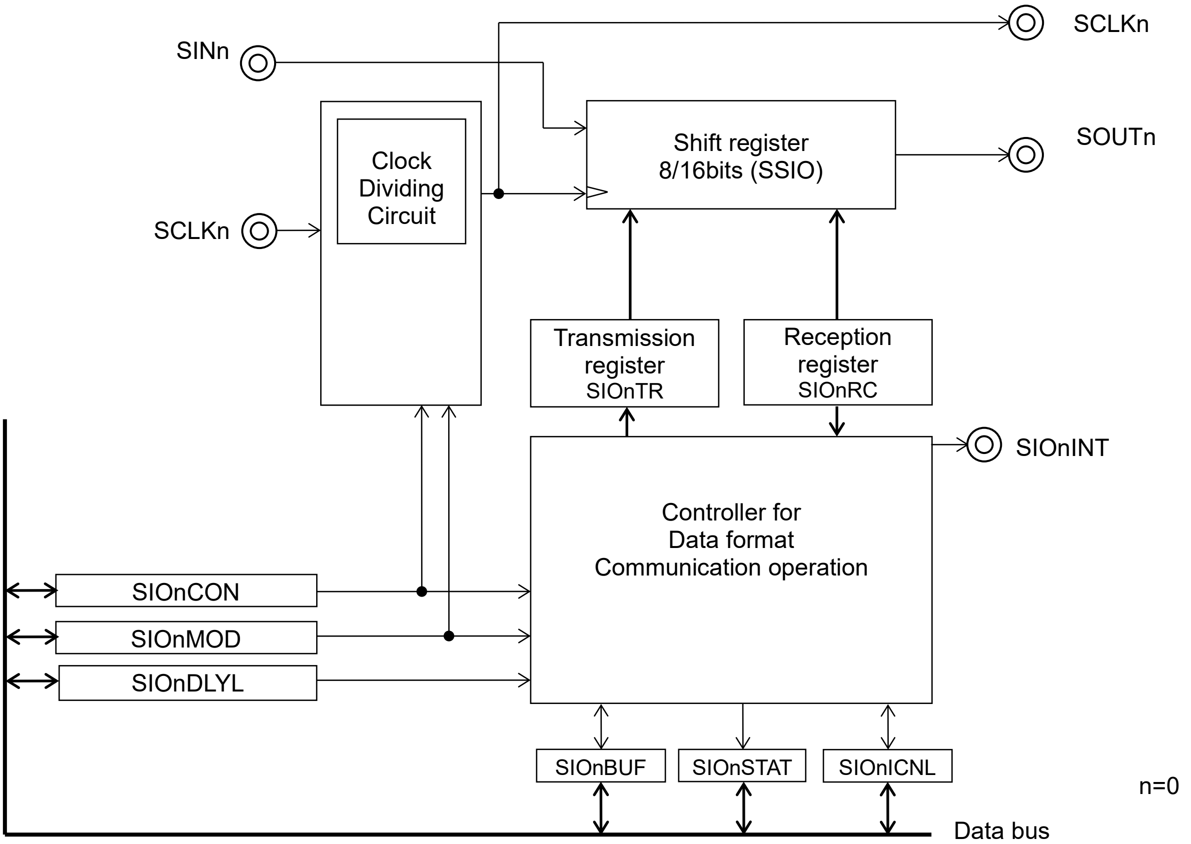
●: Available - : Unavailable

11.1.1 Features

- 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."

11.1.2 Configuration

Figure 11-1 shows configuration of the SSIO.


SCLKn
SINn
SOUTn
Clock Dividing Circuit
Shift register 8/16bits (SSIO)
SCLKn
Transmission register SIOnTR
Reception register SIOnRC
SIOnINT
Controller for Data format Communication operation
SIOnCON
SIOnMOD
SIOnDLYL
SIOnBUF
SIOnSTAT
SIOnICNL
n=0
Data bus

SIOnBUF	:	Serial port n transmission/reception buffer
SIOnMOD	:	Serial port n mode register
SIOnCON	:	Serial port n control register
SIOnDLYL	:	Serial port n interval setting register
SIOnSTAT	:	Serial port n status register
SIOnICNL	:	Serial port n interrupt control register
SIOnINT	:	SSIO n interrupt

Figure 11-1 Configuration of SSIO

11.1.3 List of Pins

The I/O pins of the SSIO are assigned to the shared function of the general ports.

Pin name	I/O	Description
SCLKn	I/O	Synchronous clock input/output of SSIO n
SOUTn	O	Transmission data output of SSIO n
SINn	I	Reception data input of SSIO n

(n=0)

Table 11-2 shows the list of the general ports used for the SSIO and the register settings of the ports.

Table 11-2 Ports used for the SSIO and the register settings (UART)

Channel no.	Pin name	Shared port		Setting register	Setting value	ML62Q2500 group
0	SIN0	P06	2nd Func.	P0MOD6	0001_XXXX*1	●
		P26	2nd Func.	P2MOD6	0001_XXXX*1	●
		P32	2nd Func.	P3MOD2	0001_XXXX*1	●
	SCLK0	P04	2nd Func.	P0MOD4	0001_XXXX*3	●
		P24	2nd Func.	P2MOD4	0001_XXXX*3	●
		P30	2nd Func.	P3MOD0	0001_XXXX*3	●
	SOUT0	P05	2nd Func.	P0MOD5	0001_XXXX*2	●
		P25	2nd Func.	P2MOD5	0001_XXXX*2	●
		P31	2nd Func.	P3MOD1	0001_XXXX*2	●

●: Available to use, -: 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)

*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

SINn, SOUTn and SCLKn are assigned to multiple general ports. Be sure to use the ports in following combinations.

Table 11-3 Combination of the SSIO port

Combination	Channel no.	Port			ML62Q2500 group
		SINn*	SOUTn*	SCLKn*	
1	0	P06	P05	P04	●
2		P26	P25	P24	●
3		P32	P31	P30	●

* :n=channel number. ●: Available to use, -: Unavailable

[Note]

- **Be sure to use the SIN0/SOUT0/SCLK0 ports with combination in the Fig.11-3, and assign each function to only one LSI pin.**

11.2 Description of Registers

11.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF500	Serial port 0 transmission/reception buffer	SIO0BUFL	SIO0BUF	R/W	8/16	0x00
0xF501		SIO0BUFH		R/W	8	0x00
0xF502	Serial port 0 status register	SIO0STATL	SIO0STAT	R	8/16	0x00
0xF503		SIO0STATH		W	8	0x00
0xF504	Serial port 0 control register	SIO0CONL	SIO0CON	R/W	8/16	0x00
0xF505		SIO0CONH		R/W	8	0x00
0xF506	Serial port 0 mode register	SIO0MODL	SIO0MOD	R/W	8/16	0x00
0xF507		SIO0MODH		R/W	8	0x00
0xF508	Serial port 0 interval setting register	SIO0DLYL	-	R/W	8	0x00
0xF509	Reserved	-	-	-	-	-
0xF50A	Serial port 0 interrupt control register	SIO0ICNL	-	R/W	8	0x00
0xF50B	Reserved	-	-	-	-	-
0xF50C	Reserved	-	-	-	-	-
0xF50D		-		-	-	-
0xF50E	Reserved	-	-	-	-	-
0xF50F				-	-	-

11.2.2 Serial Port n Transmission/Reception Buffer (SIOnBUF)

This is a SFR to store transmission/ reception data.

Address :	0xF500 (SIO0BUFL/SIO0BUF), 0xF501 (SIO0BUFH)
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	SIOnBUF															
Byte	SIOnBUFH								SIOnBUFL							
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

If writing data into this register, the data is stored into the transmission register (SIOnTR).
If reading data, the data in the reception data (SIOnRC) is read out.
In an 8-bit mode, a communication is start when writing into SIOnBUFL at the SnEN bit of SIOnCON is "1" and the SnFUL bit of SIOnSTATL is "0".
In an 16-bit mode, a communication is start when writing into SIOnBUFH at the SnEN bit of SIOnCON is "1" and the SnFUL bit of SIOnSTATL is "0".

The transmission data is 0xFF/0xFFFF when transmission data buffer is empty in the slave mode.

Bit No.	Bit symbol name	Description
15 to 8	SnB15 to SnB8	Transmission/Reception data buffer for the upper side 8 bit. If writing data into this register, the data is stored into the transmission register (SIOnTR). If reading data, the data in the reception data (SIOnRC) is read out. These bits are unused and not writable in an 8-bit mode.
7 to 0	SnB7 to SnB0	Transmission/Reception data buffer for the lower side 8 bit. 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.

11.2.3 Serial Port n Status Register (SIOnSTAT)

This is a SFR to indicate the state of the transmission/reception operation.

Address :	0xF502 (SIO0STATL/SIO0STAT),0xF502 (SIO0STATH)
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	SIOnSTAT															
Byte	SIOnSTATH								SIOnSTATL							
Bit	-	-	-	-	-	SnTOC	SnROC	SnTUC	-	SnRFUL	-	SnTXF	SnFUL	SnTOER	SnROER	SnTUER
R/W	R	R	R	R	R	W	W	W	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit no.	Bit symbol name	Description
15 to 11	-	Reserved bits
10	SnTOC	This is used to clear the SnTOER bit of SIOnSTATL. Writing “0”: Invalid Writing “1”: clear a SnTOER bit.
9	SnROC	This is used to clear the SnROER bit of SIOnSTATL. Writing “0”: Invalid Writing “1”: clear a SnROER bit.
8	SnTUC	This is used to clear the SnTUER bit of SIOnSTATL. Writing “0”: Invalid Writing “1”: clear a SnTUER bit.
7	-	Reserved bits
6	SnRFUL	This is used to indicate a state of reception buffer. This is cleared when SIOnBUF is read. 0: Reception buffer is null. (Initial value) 1: Reception buffer is full. The reception overrun error; SnROER occurs when new data is received in the state of SnRFUL = 1. A data in reception buffer is overwritten to newer data at this time.
5	-	Reserved bits
4	SnTXF	This is used to indicate communication state. 0: communication is stop and idle state. (Initial value) 1: communication is active; transmitting / receiving.
3	SnFUL	This is used to indicate state of the transmission buffer (SIOnBUF). This bit is set to “1” by writing a data to SIOnBUFL in the 8-bit mode, or by writing a data to SIOnBUFH in the 16-bit mode, and reset to “0” when starting to transfer the data. When the SnEN bit of the SIOnCON register is set to "1" on the condition of SnFUL is “1” and master mode, the transmission starts. When writing data to SIOnBUF on the condition of SnFUL is “1”, the SIOnBUF register is overwritten. This is reset when writing “1” to SnTFC bit SIOnCON register. 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 SIOnBUF register when the SnFUL bit is “1”, the SnTOER bit is set to “1”. To reset the SnTOER bit, write “1” to SnTOC 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 SIOnBUF register. To reset the SnROER bit, write “1” to SnROC bit. 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” and SnEN bit is “1”, where Transmitting data is 0xFF/0xFFFF. 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 11.3.1.4 “Timing of transmission underrun error (SnTUER). To reset the SnTUER bit, write “1” to SnTUC bit. 0: There was no transmission underrun error (Initial value) 1: There was a transmission underrun error

11.2.4 Serial Port n Control Register (SIOnCON)

This is a SFR to control the SSIO.

Address :	0xF504 (SIO0CONL/SIO0CON), 0xF505 (SIO0CONH)
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	SIOnCON															
Byte	SIOnCONH								SIOnCONL							
Bit	-	-	-	SnTFC	-	-	-	-	-	-	-	-	-	-	-	SnEN
R/W	R	R	R	W	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 13	-	Reserved bits
12	SnTFC	This is used to clear the SnFUL bit of SIOnSTATL and transmittion register; SIOnTR. If clearing SIOnTR, a data in SIOnTR is initialized to 0xFFFF. Writing “0”: Invalid Writing “1”: clear the SnFUL bit and initialize the SIOnTR.
11 to 1	-	Reserved bits
0	SnEN	This bit is used to enable communication. 0: Disabled (Initial value) 1: Enabled

11.2.5 Serial Port n Mode Register (SIOnMOD)

This is a SFR to set a mode of the SSIO n.

Address :	0xF506 (SIO0MODL/SIO0MOD), 0xF507 (SIO0MODH)
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	SIOnMOD															
Byte	SIOnMODH								SIOnMODL							
Bit	-	SnNEG	SnCKT	-	SnCK3	SnCK2	SnCK1	SnCK0	-	-	-	-	SnLG	-	-	SnDIR
R/W	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	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	-	Reserved bit
14	SnNEG	This is used to choose edge of sampling clock. 0: Positive edge (Initial value) 1: Negative edge
13	SnCKT	This is used to choose the phase of transfer clock. Four types of communication are available combining the setting of SnNEG bit. 0: Clock type 0: (Initial value) 1st edge is used to shift a data, 2nd edge is used to sample a data. 以降繰り返し。 1: Clock type 1: 1st edge is used to sample a data, 2nd edge is used to shift a data. 以降繰り返し。
12	-	Reserved bit
11 to 8	SnCK3 to SnCK0	These bits are used to choose the transfer clock. 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 master mode, max frequency of transfer clock is 4MHz; it is specified in the data sheet. In the slave mode, max frequency of transfer clock is 1/4 frequency of the system clock or specified frequency in the data sheet. 0000: 1/1 SYSCLK (Initial value) 0001: 1/2 SYSCLK 0010: 1/4 SYSCLK 0011: 1/8 SYSCLK 0100: 1/16 SYSCLK 0101: 1/32 SYSCLK 0110: 1/64 SYSCLK 0111: 1/128 SYSCLK 1000: External clock (Slave mode) Others: External clock (Slave mode)
7 to 4	-	Reserved bits
3	SnLG	This bit is used choose the bit length of the transmission/reception data. 0: 8-bit length (Initial value) 1: 16-bit length
2 to 1	-	Reserved bits
0	SnDIR	This bit is used to choose the data direction. 0: LSB first (Initial value) 1: MSB first

[Note]

- **Be sure to set the SIOnMOD register while communication is stopped (SnEN=0). If it is rewritten during communication, data may be transmitted or received incorrectly.**

11.2.6 Serial Port n Interval Setting Register (SIOnDLYL)

This is a SFR to set the 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.
This is available in the master mode.

Address :	0xF508 (SIO0DLYL)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

<table>
<tr><td></td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Word</td><td colspan="16">-</td></tr>
<tr><td>Byte</td><td colspan="8">-</td><td colspan="8">SIOnDLYL</td></tr>
<tr><td>Bit</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>SnDL
Y7</td><td>SnDL
Y6</td><td>SnDL
Y5</td><td>SnDL
Y4</td><td>SnDL
Y3</td><td>SnDL
Y2</td><td>SnDL
Y1</td><td>SnDL
Y0</td></tr>
<tr><td>R/W</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Bit no.	Bit symbol name	Description
7 to 0	SIOnDLY7 to SIOnDLY0	These bits are used to set the frame interval. 0x00: no interval (Initial value) other: frame interval = (clock period of SCLKn) × (SIOnDLYL value +1)

11.2.7 Serial Port n Interrupt Control Register (SIOnICNL)

This is a SFR to control interrupt of SSIO.

Address : 0xF50A (SIO0ICNL)
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	-								SIOnICNL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	SnRIE	SnFIE	SnTIE
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

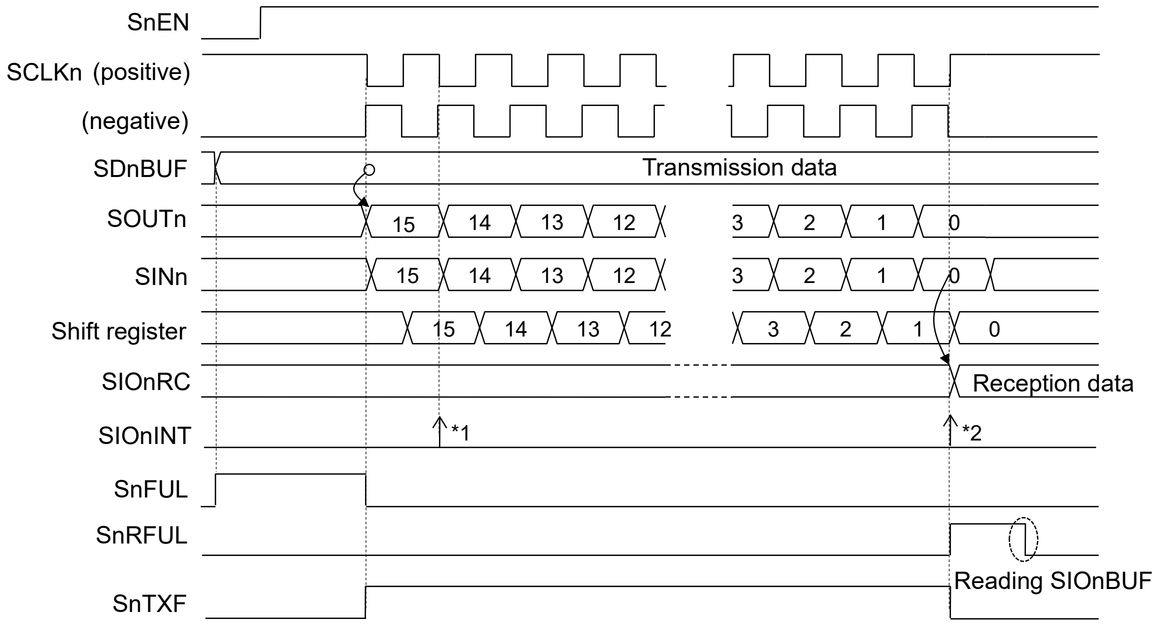
Bit no.	Bit symbol name	Description
15 to 3	-	Reserved bits
2	SnRIE	This is used to enable/disable the end interrupt of one frame transfer 0: Disabled (Initial value) 1: Enabled
1	SnFIE	This is used to enable/disable the transfer completion interrupt. This interrupt occurs when transmission/reception is finished with transmission buffer is empty(SnFUL=”0”). 0: Disabled (Initial value) 1: Enabled
0	SnTIE	This is used to enable/disable the transmission buffer empty interrupt. This interrupt occurs when transmission buffer becomes empty. 0: Disabled (Initial value) 1: Enabled

11.3 Description of Operation

11.3.1 Communication Timing (Master/Slave)

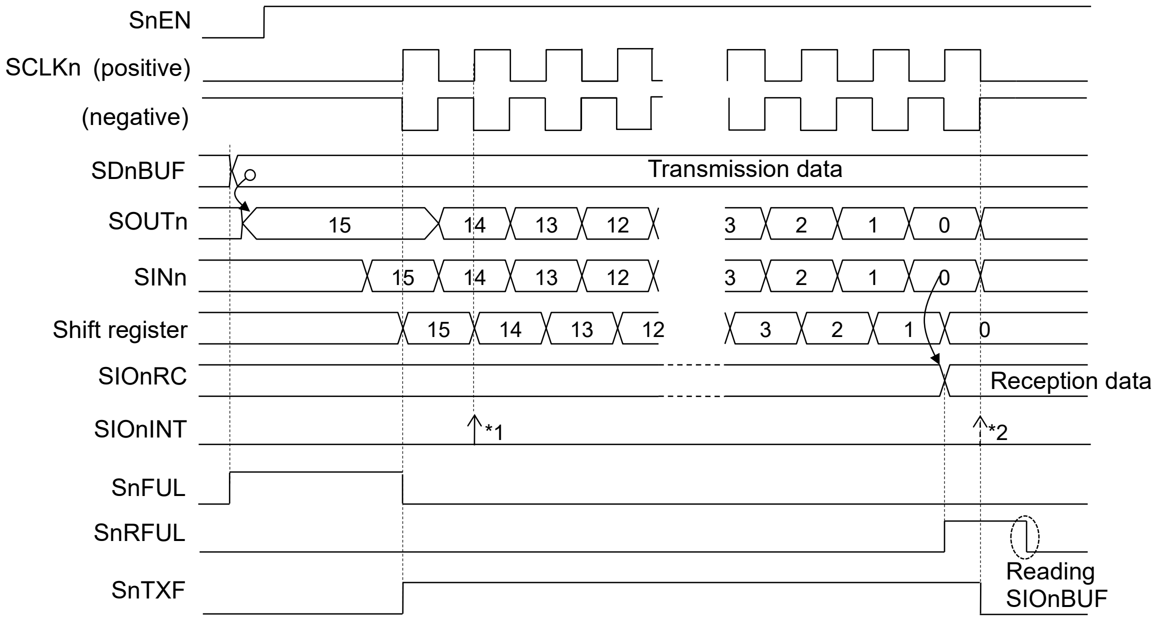
In clock type 0, the SOUTn output level when an end of one frame transfer is kept last bit data.
In clock type 1, the SOUTn output level when an end of one frame transfer becomes first bit data in SIOnBUF. If there is the next data, it is the first bit of the next data. If it is not, the first bit of the latest data.
When SnEN bit is set “0” during transmission/reception, then operation is stopped and SOUTn is kept current level in both clock type0 or 1.

Figure 11-2 shows operation waveforms; with 16-bit length, MSB first.



*1: Transmission buffer empty interrupt, *2: Transfer completion interrupt

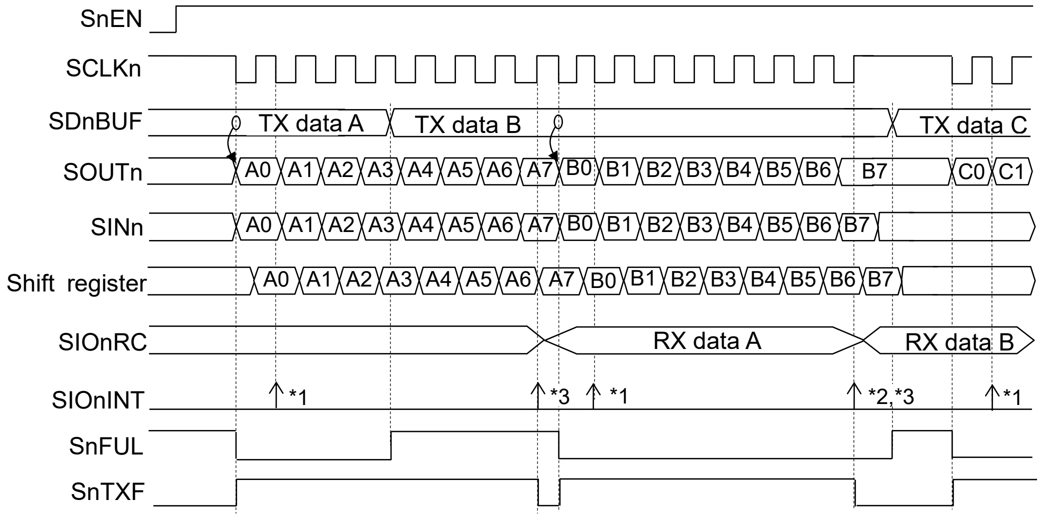
Figure 11-2-1 Waveform for Clock Type 0 with 16-bit length, MSB first



*1: Transmission buffer empty interrupt, *2: Transfer completion interrupt

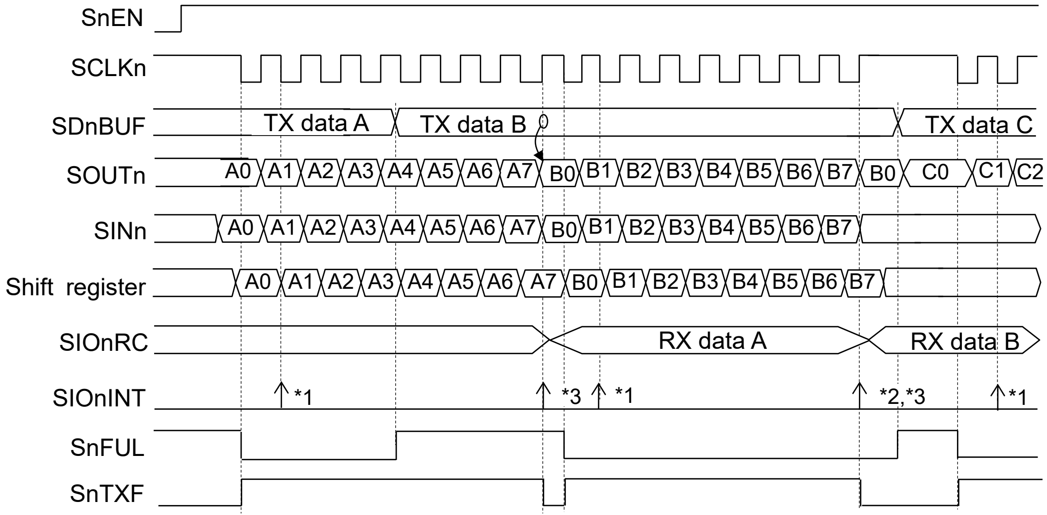
Figure 11-2-2 Waveform for Clock Type 1 with 16-bit length, MSB first

Figure 11-3 shows operation waveforms of multi-frames without interval, where is with SIOnDLY = “0x00” in the master mode, or continuous transfer in the slave mode.


SnEN
SCLKn
SDnBUF
TX data A
TX data B
TX data C
SOUTn
A0 A1 A2 A3 A4 A5 A6 A7 B0 B1 B2 B3 B4 B5 B6 B7 C0 C1
SINn
A0 A1 A2 A3 A4 A5 A6 A7 B0 B1 B2 B3 B4 B5 B6 B7
Shift register
A0 A1 A2 A3 A4 A5 A6 A7 B0 B1 B2 B3 B4 B5 B6 B7
SIOnRC
RX data A
RX data B
SIOnINT
*1
*3
*1
*2,*3
*1
SnFUL
SnTXF

*1: Transmission buffer empty interrupt, *2: Transfer completion interrupt
*3 : End interrupt of one frame transfer.

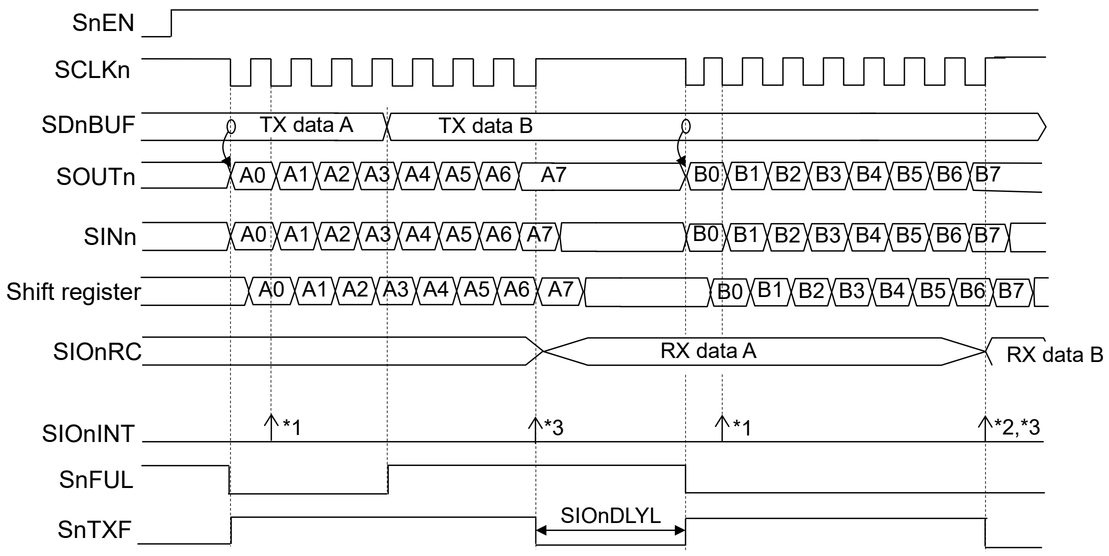
Figure 11-3-1 Waveform of multi-frames without interval in clock type 0 with positive edge


SnEN
SCLKn
SDnBUF
TX data A
TX data B
TX data C
SOUTn
A0 A1 A2 A3 A4 A5 A6 A7 B0 B1 B2 B3 B4 B5 B6 B7 B0 C0 C1 C2
SINn
A0 A1 A2 A3 A4 A5 A6 A7 B0 B1 B2 B3 B4 B5 B6 B7
Shift register
A0 A1 A2 A3 A4 A5 A6 A7 B0 B1 B2 B3 B4 B5 B6 B7
SIOnRC
RX data A
RX data B
SIOnINT
*1
*3
*1
*2,*3
*1
SnFUL
SnTXF

*1: Transmission buffer empty interrupt, *2: Transfer completion interrupt
*3 : End interrupt of one frame transfer.

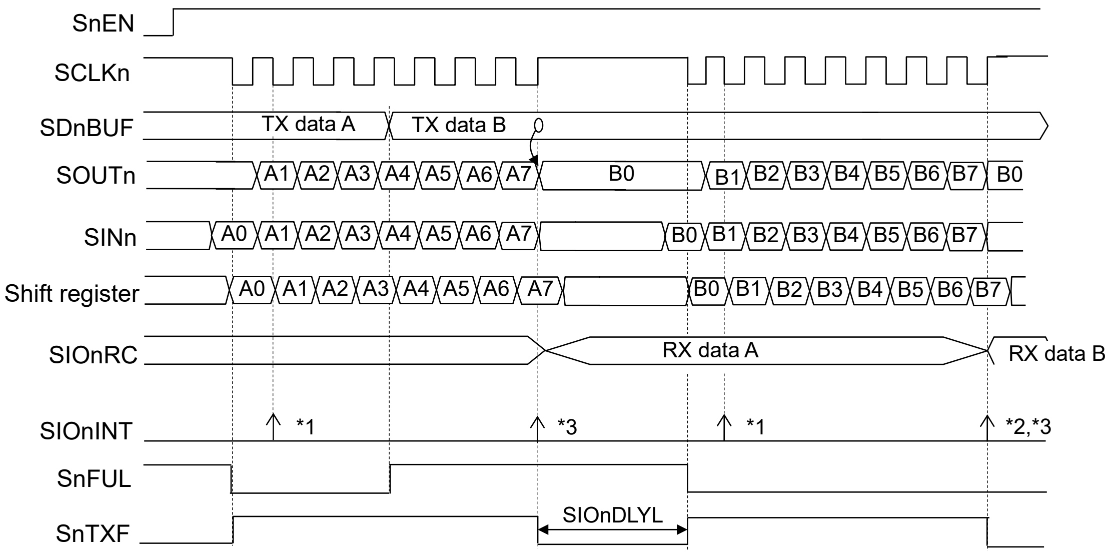
Figure 11-3-2 Waveform of multi-frames without interval in clock type 1 with negative edge

Figure 11-4 shows operation waveforms of multi-frames with frame interval.



*1: Transmission buffer empty interrupt, *2: Transfer completion interrupt
*3 : End interrupt of one frame transfer.

Figure 11-4-1 Waveforms of multi-frames with clock type 0, positive and frame interval



*1: Transmission buffer empty interrupt, *2: Transfer completion interrupt
*3 : End interrupt of one frame transfer.

Figure 11-4-2 Waveforms of multi-frames with clock type 1, negative and frame interval

Figure 11-5 shows operation waveforms when writing “0” to SnEN bit during transmission/reception in the clock type 0.

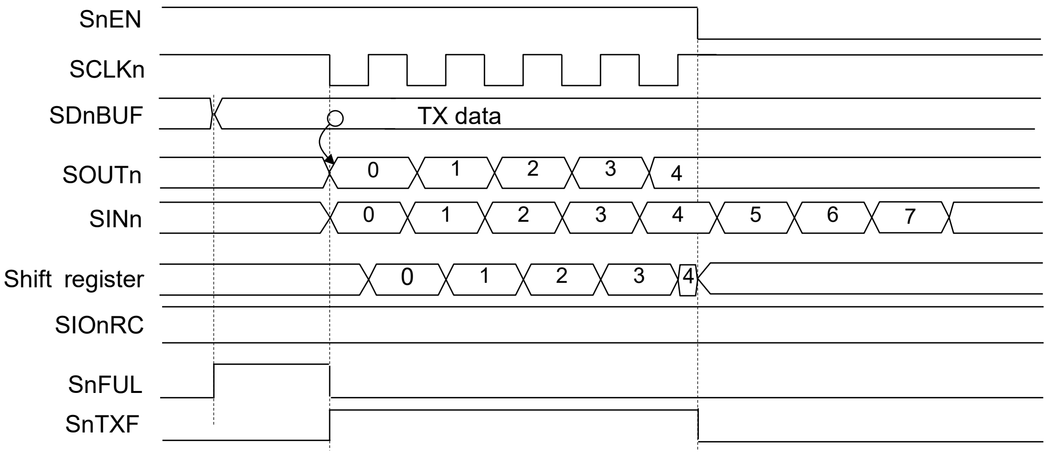


Figure 11-5 Waveforms when writing “0” to SnEN during communication

11.3.2 Interrupt Timing

Table 11-4 shows interrupt timing of SSIO.
The interrupt cause is determined by the SIOnSTAT register.

- Transmission buffer empty : SnTXF=1 & SnFUL=1
- Transfer completion : SnTXF=0 & SnFUL=0

Table 11-4-1 Interrupt Timing in the master mode

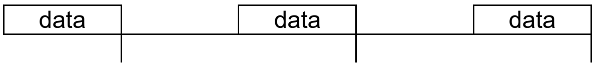
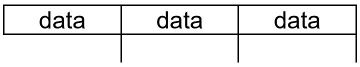
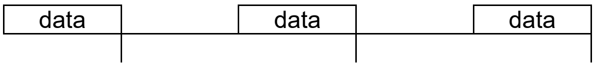
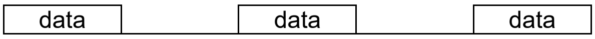
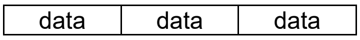
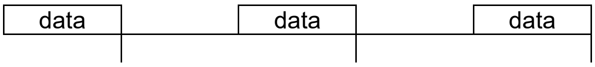
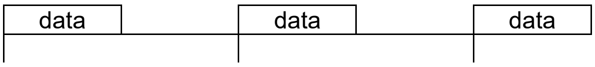
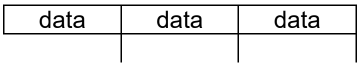
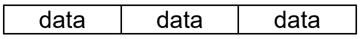
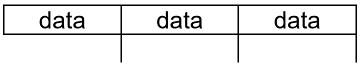
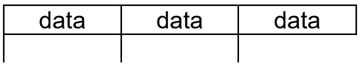
Enabled Interrupt	Frame Interval	SnFUL when end of one frame transfer	Interrupt Timing
End interrupt of one frame transfer	with	1	
	without	1	
	with/without	0	
Transfer completion Interrupt	with	1	
	without	1	
	with/without	0	
Transmission buffer empty Interrupt	with/without	1	

Table 11-4-2 Interrupt Timing in the slave mode

Enabled Interrupt	Frame Interval	SnFUL when end of one frame transfer	Interrupt Timing
End interrupt of one frame transfer	-	-	
Transfer completion Interrupt	-	1	
	-	0	
Transmission buffer empty Interrupt	-	-	

11.3.3 Example of setting

Figure 11-6 shows an example of setting for transmission and reception in the master mode.

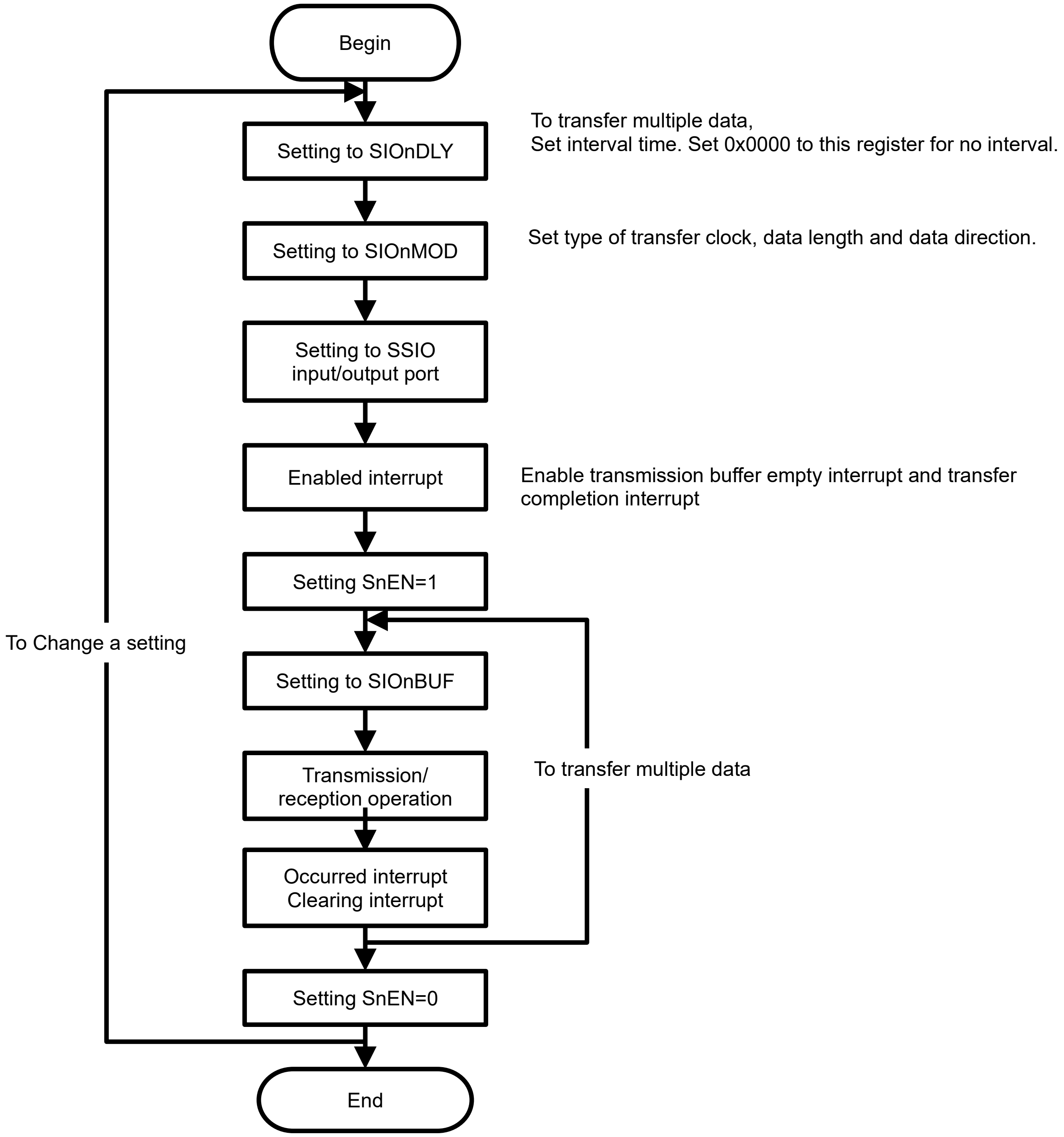

Begin
Setting to SIOnDLY
To transfer multiple data,
Set interval time. Set 0x0000 to this register for no interval.
Setting to SIOnMOD
Set type of transfer clock, data length and data direction.
Setting to SSIO input/output port
Enabled interrupt
Enable transmission buffer empty interrupt and transfer completion interrupt
Setting SnEN=1
To Change a setting
Setting to SIOnBUF
Transmission/ reception operation
To transfer multiple data
Occurred interrupt Clearing interrupt
Setting SnEN=0
End

Figure 11-6 Example of setting for transmission/reception in the master mode

Figure 11-7 shows an example of setting for transmission and reception in the slave mode.

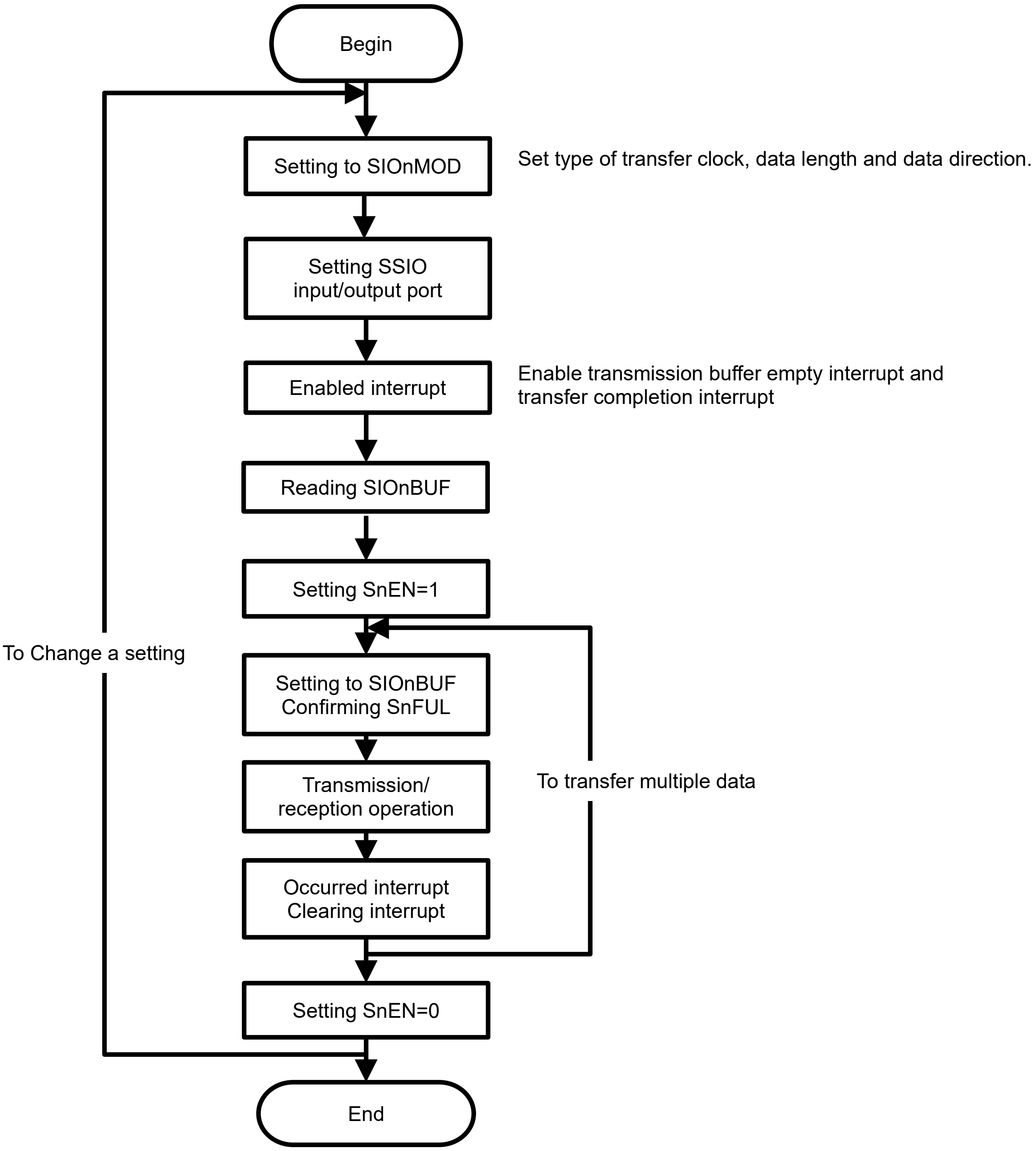
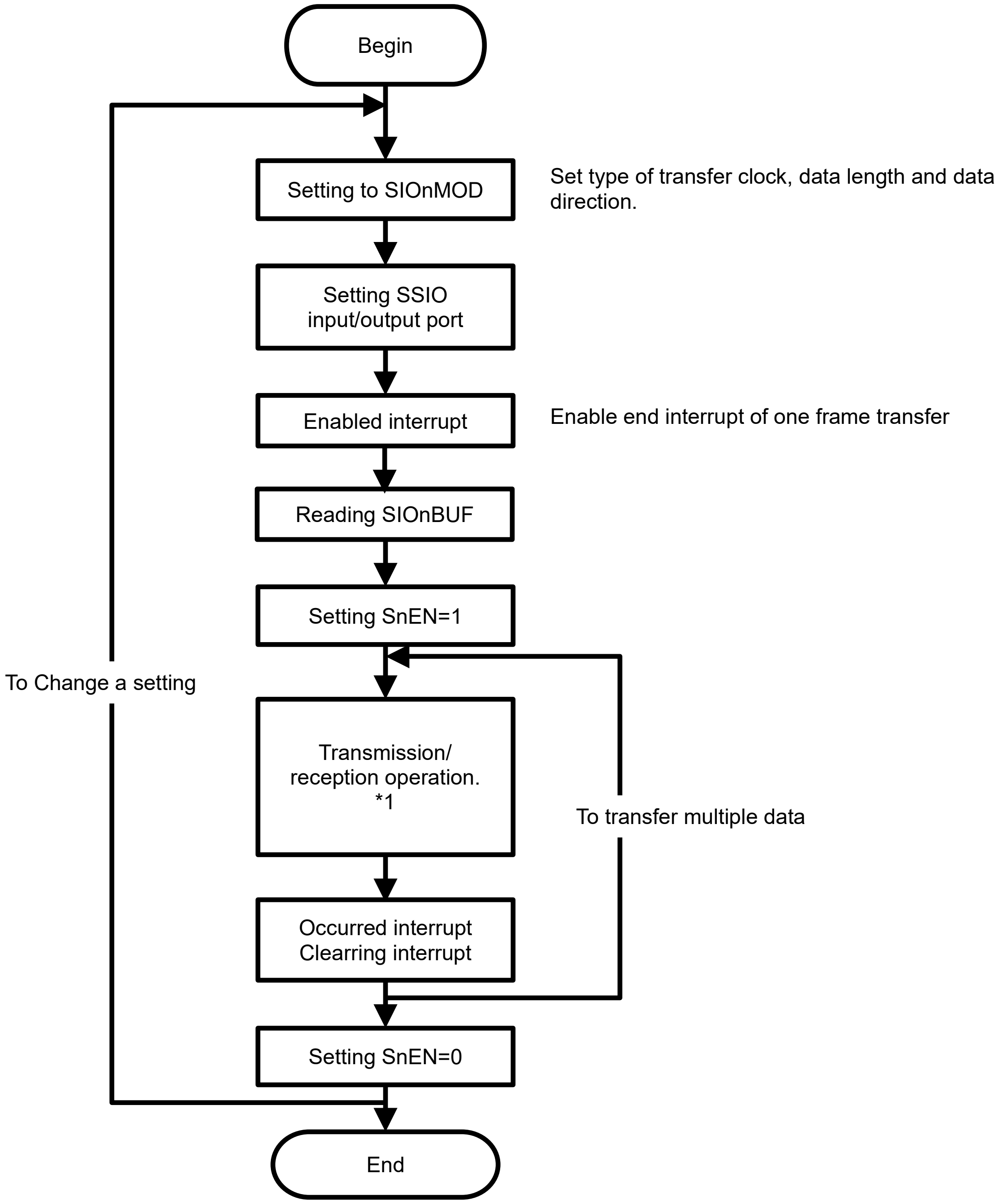

Begin
Setting to SIOnMOD
Set type of transfer clock, data length and data direction.
Setting SSIO input/output port
Enabled interrupt
Enable transmission buffer empty interrupt and transfer completion interrupt
Reading SIOnBUF
Setting SnEN=1
To Change a setting
Setting to SIOnBUF Confirming SnFUL
Transmission/ reception operation
To transfer multiple data
Occurred interrupt Clearing interrupt
Setting SnEN=0
End

Figure 11-7 Example of setting for transmission/reception in the slave mode

[Note]

- **To prevent an overrun error after the first reception, read the SIOnBUF register before setting the SnEN bit to "1".**

Figure 11-8 shows an example of setting for reception only in the slave mode.


Begin
Setting to SIOnMOD
Set type of transfer clock, data length and data direction.
Setting SSIO input/output port
Enabled interrupt
Enable end interrupt of one frame transfer
Reading SIOnBUF
Setting SnEN=1
To Change a setting
Transmission/ reception operation. *1
To transfer multiple data
Occurred interrupt Clearring interrupt
Setting SnEN=0
End

*1 : the transmission data is latest written data. The data is null data (All1) if the SIOnBUF has never been written.

Figure 11-8 Example of setting for reception in the slave mode

[Note]

- **To prevent an overrun error after the first reception, read the SIOnBUF register before setting the SnEN bit to "1".**

11.3.4 Timing of Transmission Underrun Error (SnTUER)

A transmission underrun error (SnTUER) occurs only slave mode. In master mode, SnTUER is not set, because a transmission is executed at SnFUL=”1”.

Figure 11-9 shows a detection timing of transmission underrun error.

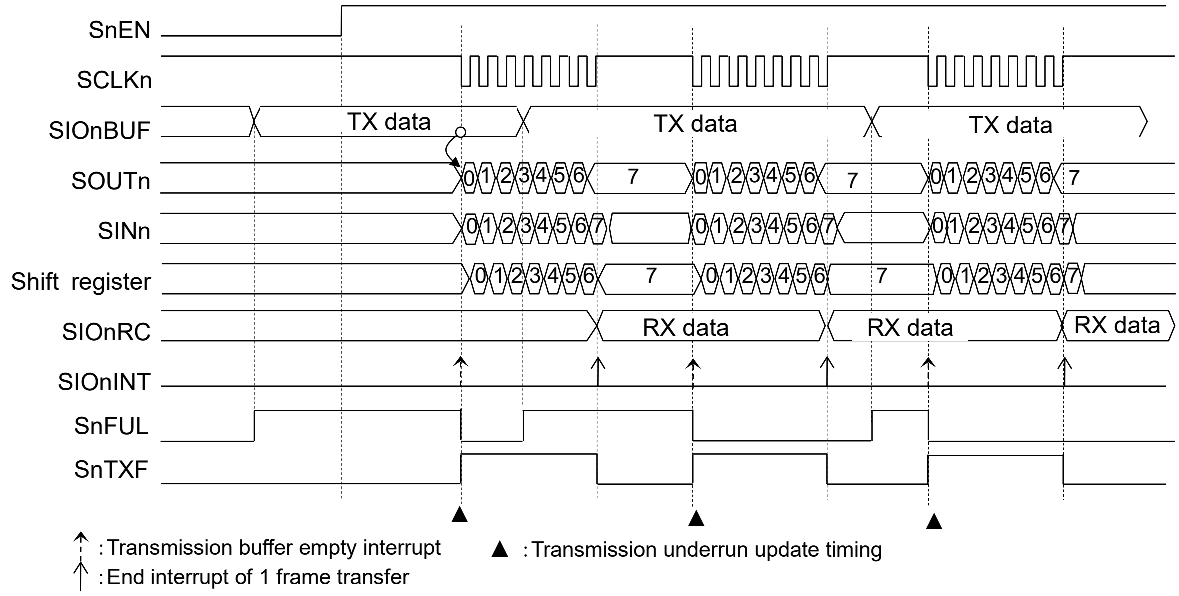


Figure 11-9-1 clock type 0 with positive

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.

It is possible to write data to the transfer buffer after updating underrun before the transfer is actually started (before the external clock is supplied). A latest written data is transmitted.

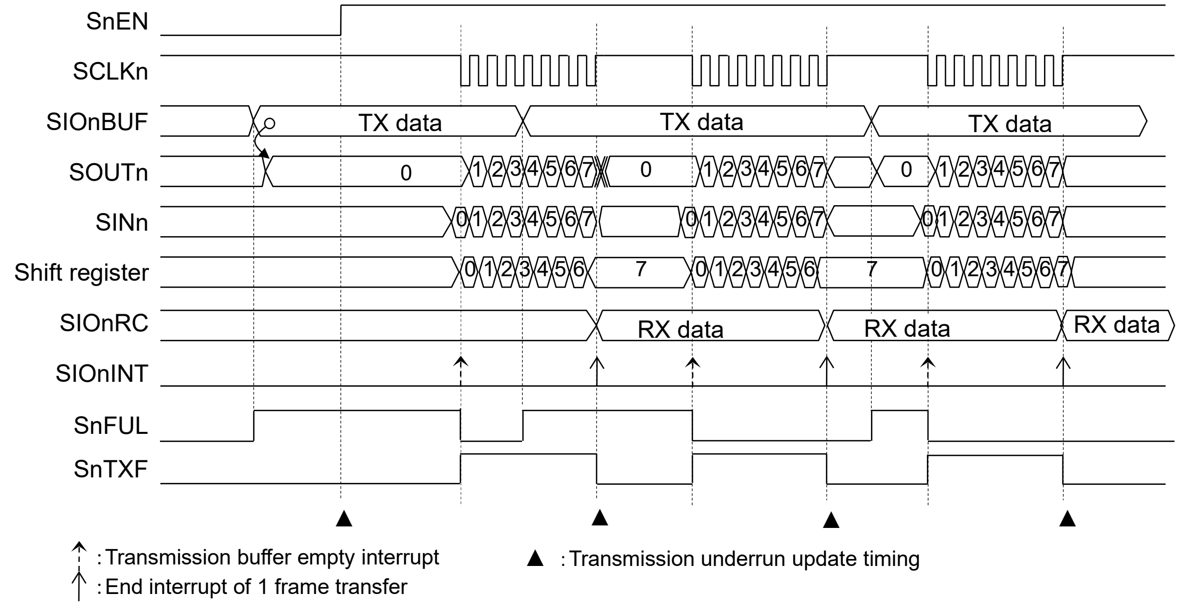


Figure 11-9-2 clock type 1 with negative

[Note]

- **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) in the clock type1 slave mode.**

Chapter 12 Synchronous Serial Port with FIFO (SSIOF)

12. Synchronous Serial Port with FIFO (SSIOF)

12.1 General Description

This SSIOF can communicate with peripherals and other MCUs.
Table 12-1 shows the number of channels.

Table12-1 Number of the SIOF

Channel no.	ML62Q2500 group
0	●

●: Available - : Unavailable

12.1.1 Features

- Full-duplex data transfer
- Master or Slave mode can be selected
- Built-in 4-stage FIFO on each of transmit- and receive-sides
- For the transfer size, 8 bits (byte) or 16 bits (word) can be selected
- The number of received bytes (words) that cause interrupts can be set to 1 to 4.
- The number of untransmitted bytes (words) that cause interrupts can be set to 0 to 3.
- LSB first or MSB first can be selected
- The polarity and phase of the serial clock are selectable
- In Master mode, the OSCLK's 2 to 2046-division clocks can be selected as the sync clock (1023 types)
- In Master mode, the interval before/after transfer can be controlled
- State bit indicating transmission/receive complete and FIFO state
- Detects a mode fault error to avoid multi-master bus contention
- Detects a write overflow error if any further writing is attempted when the transmit FIFO is in the full state
- Self-test function using the master and slave modes. For the self-test functions, see Chapter 29 "Safety Function."

12.1.2 Configuration

Figure 12-1 shows configuration of the SSIOF.

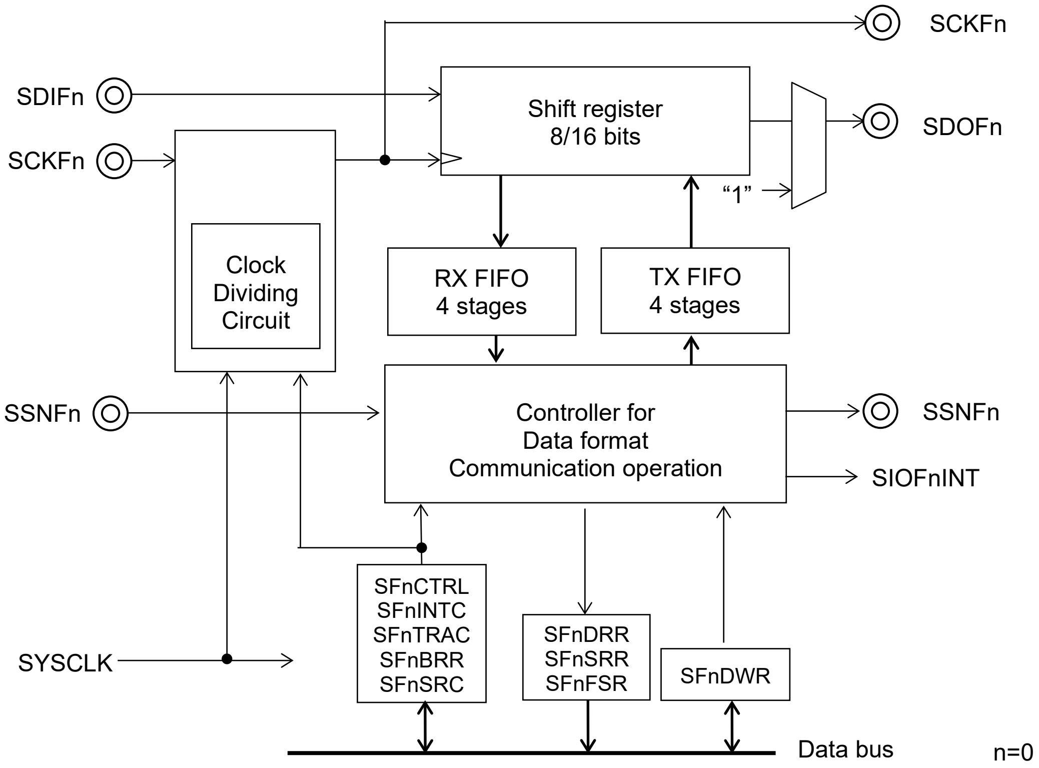


Figure 12-1 Configuration of SSIOF

12.1.3 List of Pins

The I/O pins of the SSIOF are assigned to the shared function of the general ports.

Pin name	I/O	Description
SDOF0	O	SSIOF0 transmission data output
SDIF0	I	SSIOF0 reception data input
SCKF0	I/O	SSIOF0 baud rate synchronous clock
SSNF0	I/O	SSIOF0 slave selection (active level is low)

Table 12-2 shows the list of the general ports used for the SSIOF and the register settings of the ports.

Table 12-2 Ports used for the SSIOF and the register settings

Channel no.	Pin name	Shared port		Setting register	Setting value	ML62Q2500 group
0	SDIF0	P12	2nd Func.	P1MOD2	0001_XXXX*1	●
		P22	2nd Func.	P2MOD2	0001_XXXX*1	●
	SDOF0	P11	2nd Func.	P1MOD1	0001_XXXX*2	●
		P21	2nd Func.	P2MOD1	0001_XXXX*2	●
	SCKF0	P10	2nd Func.	P1MOD0	0001_XXXX*3	●
		P20	2nd Func.	P2MOD0	0001_XXXX*3	●
	SSNF0	P13	2nd Func.	P1MOD3	0001_XXXX*3	●
		P23	2nd Func.	P2MOD3	0001_XXXX*3	●

●: 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.

12.1.4 Combination of SSIOF port

SDOF0, SDIF0, SCKF0, SSNF0 are assigned to multiple general ports.
Be sure to use the ports in following combinations.

Table 12-3 Combination of the SSIOF

Combination	Channel no.	Port				ML62Q2500 group
		SDIFn*	SDOFn*	SCKFn*	SSNFn*	
1	0	P12	P11	P10	P13	●
2		P22	P21	P20	P23	●

* :n=channel number. ●: Available to use, -: Unavailable

[Note]

- **Be sure to use the SDIF0/SDOF0/SCKF0/SSNF0 ports with combination in the Fig.12-3, and assign each function to only one LSI pin.**

12.2 Description of Registers

12.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF580	SIOF0 control register	SF0CTRLL	SF0CTRL	R/W	8/16	0x00
0xF581		SF0CTRLH		R/W	8	0x00
0xF582	SIOF0 interrupt control register	SF0INTCL	SF0INTC	R/W	8/16	0x00
0xF583		SF0INTCH		R/W	8	0x00
0xF584	SIOF0 transfer interval control register	SF0TRACL	SF0TRAC	R/W	8/16	0x02
0xF585		SF0TRACH		R/W	8	0x00
0xF586	SIOF0 baud rate register	SF0BRRL	SF0BRR	R/W	8/16	0x02
0xF587		SF0BRRH		R/W	8	0x50
0xF588	SIOF0 status register	SF0SRRL	SF0SRR	R	8/16	0x00
0xF589		SF0SRRH		R	8	0x14
0xF58A	SIOF0 status clear register (L/H)	SF0SRCL	-	W	8	0x00
0xF58B		SF0SRCH		W	8	0x00
0xF58C	SIOF0 FIFO status register	SF0FSRL	SF0FSR	R	8/16	0x00
0xF58D		SF0FSRH		R	8	0x00
0xF58E	SIOF0 writing data register	SF0DWRL	SF0DWR	R/W	8/16	0x00
0xF58F		SF0DWRH		R/W	8	0x00
0xF590	SIOF0 reading data register	SF0DRRL	SF0DRR	R	8/16	0x00
0xF591		SF0DRRH		R	8	0x00

12.2.2 SIOF0 Control Register (SF0CTRL)

This is as SFR to control SSIOF0 operation.

Address : 0xF580 (SF0CTRLL /SF0CTRL), 0xF581 (SF0CTRLH)
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	SF0CTRL															
Byte	SF0CTRLH								SF0CTRLL							
Bit	-	-	-	-	-	-	-	SF0FICL	-	SF0CPOL	SF0CPHA	SF0LSB	-	SF0SIZ	SF0MST	SF0SPE
R/W	R	R	R	R	R	R	R	R/W	R	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 to 9	-	Reserved bits
8	SF0FICL	This is used to clear FIFO status. 0: No operation (Initial value) 1: Clear frame counts of reception/transmission
7	-	Reserved bits
6	SF0CPOL	This is used to choose edge of transfer clock. 0: 1st edge is positive; the clock level is “L” during the idling. (Initial value) 1: 1st edge is negative; the clock level is “H” during the idling.
5	SF0CPHA	This is used to choose the phase of transfer clock. 0: Clock type 1: (Initial value) 1st edge is used to sample a data, 2nd edge is used to shift a data. After that, it is repeated. 1: Clock type 0: 1st edge is used to shift a data, 2nd edge is used to sample a data. After that, it is repeated.
4	SF0LSB	This is used to choose transfer data direction. 0: LSB first (Initial value) 1: MSB first
3	-	Reserved bits
2	SF0SIZ	This is used to choose transfer size of 1 frame. 0: 8 bit (Initial value) 1: 16 bit
1	SF0MST	This is used to choose master/slave mode. 0: Slave mode (Initial value) 1: Master mode
0	SF0SPE	This is used to choose enable/disable a transfer of the SSIOF. 0: Disabled (Initial value) 1: Enabled

12.2.3 SIOF0 Interrupt Control Register (SF0INTC)

This is a SFR to control interrupt operation for SSIOF0.

Address :	0xF582 (SF0INTCL/SF0INTC), 0xF583 (SF0INTCH)
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	SF0INTC															
Byte	SF0INTCH								SF0INTCL							
Bit	-	-	SF0RFIC1	SF0RFIC0	-	-	SF0TFIC1	SF0TFIC0	-	-	-	-	SF0ORIE	SF0FIE	SF0RFIE	SF0TFIE
R/W	R	R	R/W	R/W	R	R	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 14	-	Reserved bits
13 to 12	SF0RFIC1 to SF0RFIC0	These bits are configured count in reception FIFO for the reception interrupt request occurs. 00: Stacking 1 frame data (Initial value) 01: Stacking 2 frame data 10: Stacking 3 frame data 11: Stacking 4 frame data
11 to 10	-	Reserved bits
9 to 8	SF0TFIC1 to SF0TFIC0	These bits are configured the remaining count in transmission FIFO for the transmission interrupt request occurs. 00: Empty in the FIFO (Initial value) 01: Remaining 1 frame data 10: Remaining 2 frame data 11: Remaining 3 frame data
7 to 4	-	Reserved bits
3	SF0ORIE	This is configured the SSIOF0 overrun error interrupt request. 0: Disabled (Initial value) 1: Enabled
2	SF0FIE	This is configured the SSIOF0 transfer completion interrupt request. 0: Disabled (Initial value) 1: Enabled
1	SF0RFIE	This is configured the SSIOF0 reception interrupt request. 0: Disabled (Initial value) 1: Enabled
0	SF0TFIE	This is configured the SSIOF0 transmission interrupt request. 0: Disabled (Initial value) 1: Enabled

12.2.4 SIOF0 Transfer Interval Control Register (SF0TRAC)

This is a SFR used to set the minimum data transfer interval in Master mode.
See 12.3.6 “Transfer Interval Setting” for details.

Address :	0xF584 (SF0TRACL/SF0TRAC), 0xF585 (SF0TRACH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0002

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SF0TRAC															
Byte	SF0TRACH								SF0TRACL							
Bit	-	-	-	-	-	-	-	SF0D TL8	SF0D TL7	SF0D TL6	SF0D TL5	SF0D TL4	SF0D TL3	SF0D TL2	SF0D TL1	SF0D TL0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

12.2.5 SIOF0 Baud Rate Register (SF0BRR)

This is a SFR used to set the operation mode.
Do not change the setting of this register during transfer.

Address :	0xF586 (SF0BRRL/SF0BRR), 0xF587 (SF0BRRH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x5002

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SF0BRR															
Byte	SF0BRRH								SF0BRRL							
Bit	SF0LAG1	SF0LAG0	SF0LEAD1	SF0LEAD0	-	-	SF0BR9	SF0BR8	SF0BR7	SF0BR6	SF0BR5	SF0BR4	SF0BR3	SF0BR2	SF0BR1	SF0BR0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0

Bit no.	Bit symbol name	Description
15 to 14	SF0LAG1 to SF0LAG0	This is configured interval between SCKF0 and SSNF0 (H). 00: 0.5 × SCKF0 01: 0.5 × SCKF0 (Initial value) 10: 1.0 × SCKF0 11: 1.5 × SCKF0
13 to 12	SF0LEAD1 to SF0LEAD0	This is configured interval between SSNF0 (L) and SCKF0. 00: 0.5 × SCKF0 01: 0.5 × SCKF0 (Initial value) 10: 1.0 × SCKF0 11: 1.5 × SCKF0
11 to 10	-	Reserved bits
9 to 0	SF0BR9 to SF0BR0	This is configured baud rate. It is valid in the master mode. $f_{SCK}=f_{SYSCLK}$ / (2 × SF0BR9-0) f_{SYSCLK} : SYSCLK frequency 0000000000: 2 dividing 0000000001: 2 dividing 0000000010: 4 dividing (Initial value) 0000000011: 6 dividing : 1111111111: 2046 dividing In the master mode, max frequency of transfer clock is 4MHz; it is specified in the data sheet.

12.2.6 SIOF0 Status Register (SF0SRR)

This is a SFR used to indicate the data transfer state and error state of the SSIOF0.

Address :	0xF588 (SF0SRRL/SF0SRR), 0xF589 (SF0SRRH)
Access :	R
Access size :	8/16 bit
Initial value :	0x1400

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SF0SRR															
Byte	SF0SRRH								SF0SRRL							
Bit	-	-	SF0SSF	SF0RFE	SF0RFF	SF0TFE	SF0TFF	SF0WOF	-	-	SF0SPIF	-	SF0ORF	SF0FI	SF0RFI	SF0TFI
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0

Bit no.	Bit symbol name	Description
15 to 14	-	Reserved bits
13	SF0SSF	This is used to monitor slave selection signal; SSNF0. 0: "H" level (Initial value) 1: "L" level; communicating
12	SF0RFE	This is used to indicate reception FIFO empty. 0: Not empty 1: Empty; No interrupt is generated (Initial value)
11	SF0RFF	This is used to indicate reception FIFO full. 0: Not full (Initial value) 1: Full; No interrupt is generated
10	SF0TFE	This is used to indicate transmission FIFO empty. 0: Not empty 1: Empty; No interrupt is generated (Initial value)
9	SF0TFF	This is used to indicate transmission FIFO full. 0: Not full (Initial value) 1: Full; No interrupt is generated
8	SF0WOF	This is used to indicate an overflow of transmission FIFO 0: Not occurred (Initial value) 1: Occurred; No interrupt is generated
7 to 6	-	Reserved bits
5	SF0SPIF	This is used to indicate completion of 1 frame transfer. 0: Not completed (Initial value) 1: Completed
4	-	Reserved bits
3	SF0ORF	This is used to indicate an overrun of reception FIFO. 0: Not occurred (Initial value) 1: Occurred; interrupt is generated
2	SF0FI	This is used to indicate transfer completion interrupt. It occurs when transfer last frame data is completed in condition of that transmission FIFO is empty. 0: No interrupt request (Initial value) 1: Interrupt request
1	SF0RFI	This is used to indicate reception interrupt. It occurs when the number of data in the reception FIFO is equal or more frame count configured with SF0RFIC. 0: No interrupt request (Initial value) 1: Interrupt request
0	SF0TFI	This is used to indicate transmission interrupt. It occurs when the remaining data in the transmission FIFO matches the frame count configured with SF0TFIC. 0: No interrupt request (Initial value) 1: Interrupt request

12.2.7 SIOF0 Status Clear Register L/H (SF0SRCL, SF0SRCH)

This is a SFR used to clear the data transfer state and error state of the SSIOF.

Address :	0xF58A (SF0SRCL), 0xF58B (SF0SRCH)
Access :	W
Access size :	8 bit
Initial value :	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	SF0SRCH								SF0SRCL							
Bit	SF0IRQ	-	-	-	-	-	-	SF0WOFC	-	-	SF0SPIFC	-	SF0ORFC	SF0FC	SF0RFC	SF0TFC
R/W	W	R	R	R	R	R	R	W	R	R	W	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	SF0IRQ	This is the bit to assert interrupt request. When there is any unprocessed interrupt sources, the interrupt request is issued again by writing "1".
14 to 9	-	Reserved bits
8	SF0WOFC	This is used to clear the transmission overflow flag; SF0WOF bit. The flag is cleared by writing “1” to this bit.
7 to 6	-	Reserved bits
5	SF0SPIFC	This is used to clear the one frame transfer completion flag; SF0SPIF bit. The flag is cleared by writing “1” to this bit.
4	-	Reserved bits
3	SF0ORFC	This is used to clear the interrupt request of overrun error flag; SF0ORF bit. The request is cleared by writing “1” to this bit. For the request, check on the SF0ORF bit of the SF0SRR register.
2	SF0FC	This is used to clear the transfer completion interrupt request ;SF0FI bit. The request is cleared by writing “1” to this bit. For the request, check on the SF0FI bit of the SF0SRR register.
1	SF0RFC	This is used to clear the reception interrupt request ;SF0RFI bit. The request is cleared by writing “1” to this bit. For the request, check on the SF0RFI bit of the SF0SRR register.
0	SF0TFC	This is used to clear the transmission interrupt request ;SF0TFI bit. The request is cleared by writing “1” to this bit. For the request, check on the SF0TFI bit of the SF0SRR register.

[Note]

- **Write "1" to SF0IRQ bit while there is any unprocessed interrupt source and processing all the interrupt sources before exiting the interrupt vector will cause re-entry to the interrupt vector with no interrupt source after exiting the interrupt vector. Ensure to write "1" before exiting the interrupt vector.**

12.2.8 SIOF0 FIFO Status Register (SF0FSR)

This is a SFR used to indicate the remaining frame counts in transmission/reception FIFO.

Address :	0xF58C (SF0FSRL/SF0FSR), 0xF58D (SF0FSRH)
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	SF0FSR															
Byte	SF0FSRH								SF0FSRL							
Bit	-	-	-	-	-	SF0R FD2	SF0R FD1	SF0R FD0	-	-	-	-	-	SF0T FD2	SF0T FD1	SF0T FD0
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 番号	Bit シンボル名	Description
15 to 11	-	Reserved bits
10 to 8	SF0RFD2 to SF0RFD0	These bit are used to indicate the remaining frame counts in the reception FIFO. 000: Empty (Initial value) 001: 1 frame 010: 2 frames 011: 3 frames 100: 4 frames (Full)
7 to 3	-	Reserved bits
2 to 0	SF0TFD2 to SF0TFD0	These bit are used to indicate the remaining frame counts in the transmission FIFO. 000: Empty (Initial value) 001: 1 frame 010: 2 frames 011: 3 frames 100: 4 frames (Full)

12.2.9 SIOF0 Writing Data Register (SF0DWR)

This is a SFR used to write transmission data.

Address :	0xF58E (SF0DWRL/SF0DWR), 0xF58F (SF0DWRH)
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	SF0DWR															
Byte	SF0DWRH								SF0DWRL							
Bit	SF0W D15	SF0W D14	SF0W D13	SF0W D12	SF0W D11	SF0W D10	SF0W D9	SF0W D8	SF0W D7	SF0W D6	SF0W D5	SF0W D4	SF0W D3	SF0W D2	SF0W D1	SF0W D0
R/W	R/W	R/W	R/W	R/W	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

Write to this register should be :

8-bit length data write access to the SF0DWRL for 8 bit transmission; SF0SIZ=0.
16-bit length data write access to the SF0DWR for 16 bit transmission; SF0SIZ=1.

12.2.10 SIOF0 Reading Data Register (SF0DRR)

This is a SFR used to read reception data.

Address :	0xF590 (SF0DRRL/SF0DRR), 0xF591 (SF0DRRH)
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	SF0DRR															
Byte	SF0DRRH								SF0DRRL							
Bit	SF0R D15	SF0R D14	SF0R D13	SF0R D12	SF0R D11	SF0R D10	SF0R D9	SF0R D8	SF0R D7	SF0R D6	SF0R D5	SF0R D4	SF0R D3	SF0R D2	SF0R D1	SF0R D0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Read to this register should be:

8-bit length data read access to the SF0DRRL for 8 bit transmission; SF0SIZ=0.
16-bit length data read access to the SF0DRR for 16 bit transmission; SF0SIZ=1.

12.3 Description of Operation

12.3.1 Master / Slave Mode

This unit has 2 modes; the master mode and the slave mode. It is selected by the SF0MST bit of the SIOF0 control register.
SF0BR9-0 , SF0LAG1-0, SF0LEAD1-0 bits of SF0BRR and SF0DTL8-0 bits of SF0TRAC determine SCKF0 and SSNF0 operations. Each bit of SF0CPOL, SF0CPHA, SF0LSB nad SF0SIZ bits need to habe the same value for master and slave.

12.3.2 Serial Clock Baud Rate (Master Mode)

A baud rate is configured by the SF0BR9-0 bits of SF0BRR register. This is only valid in the master mode. The baud rate clock SCKF0 is generated by dividing SYSCLK. The baud rate (fSCK) is calculated as follos:

$f_{SCK}=f_{SYSCLK}/$ (2 × SF0BR9-0)

f_{SCK}	: A frequency of baud rate clock
f_{SYSCLK}	: A frequency of system clock
SF0BR9-0	: Value set in SF0BR9-0 of the SF0BRR (1 to 1023)

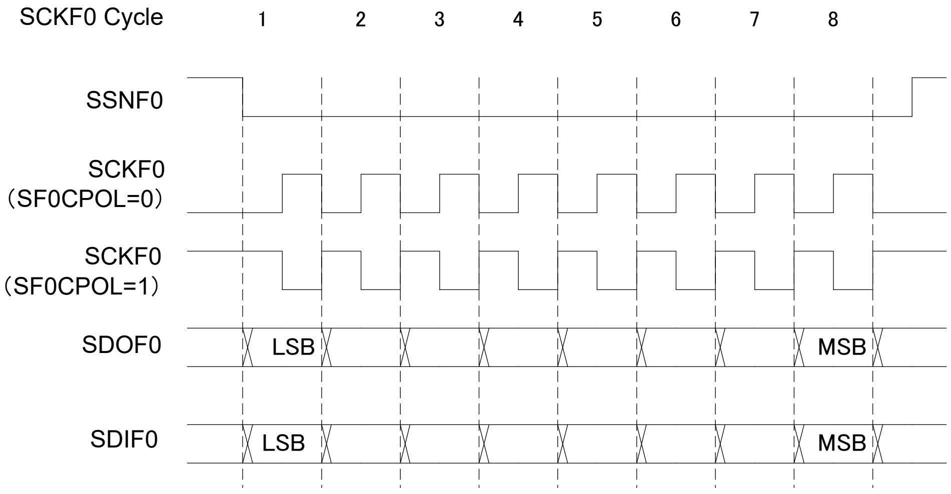
If 0 is set the SF0BR register, it is processed as 1. It can be selected from 1023 dividing types (2 to 2046)

12.3.3 Control of Polarity and Phase of Serial Clock

The SF0CPOL bit of the SF0CTRL register controls the clock polarity. SF0CPHA bit of the SF0CTRL register controls the clock phase and determines the shift timing of transmit data and the sampling timing of received data. The master and slave which communicate with each other must have the same setting values for SF0CPOL and SF0CPHA.

12.3.4 Data Transfer Timing

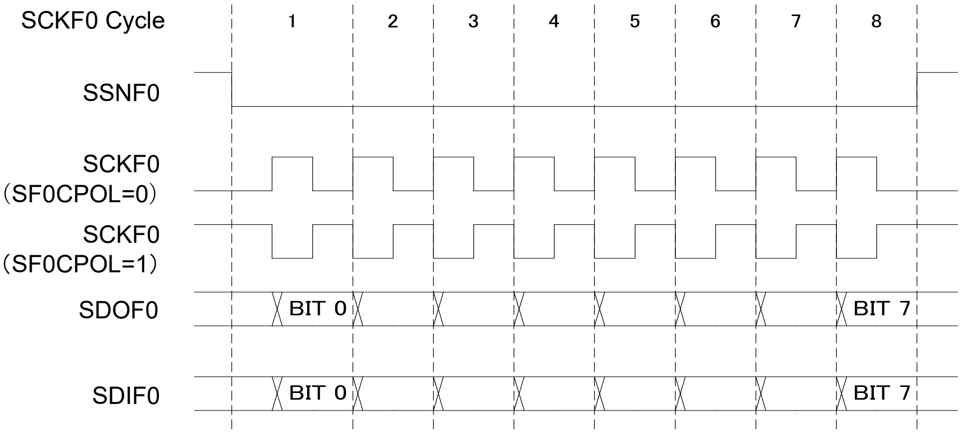
Figure 12-2 shows the data transfer timing when SF0CPHA is "0". SSF0 is the slave selection input in Slave mode. In Master mode, the transfer is started when data is written to the SF0DWR register. In Slave mode, the transfer is started at the SSF0 falling edge. The received data is sampled at the rising-edge of SCKF0 in SF0CPOL is “0” and the falling-edge of SCKF0 in SF0CPOL is “1”. The transmitted data is shifted at the falling-edge of SCKF0 in SF0CPOL is “0” and the rising-edge of SCKF0 in SF0CPOL is “1”.



Example for SF0SIZ=SF0LSB=SF0LEAD=SF0LAG=0

Figure 12-2 8-bit length data transfer at SF0CPHA=0

Figure 12-3 shows the data transfer timing when SF0CPHA is "1". SSF0 is the slave selection input in Slave mode. In Master mode, the transfer is started when data is written to SF0DWR. In Slave mode, the transfer is started at the first edge of SCKF0. The received data is sampled at the falling-edge of SCKF0 in SF0CPOL is “0” and the rising-edge of SCKF0 in SF0CPOL is “1”. The transmitted data is shifted at the rising-edge of SCKF0 in SF0CPOL is “0” and the falling-edge of SCKF0 in SF0CPOL is “1”.



Example for SF0SIZ=SF0LSB=SF0LEAD=SF0LAG=0

Figure 12-3 8-bit length data transfer at SF0CPHA=1

12.3.5 Transfer Size

The transfer size can be selected in 8 bit (byte) or 16 bit (word) as 1frame.
Transfer data read/write must be coincided to the transfer size. As the number of FIFO stages is the same for both byte and word, the number of transfers is the same. The master and slaves which communicate with each other must have the same value for SF0SIZ.

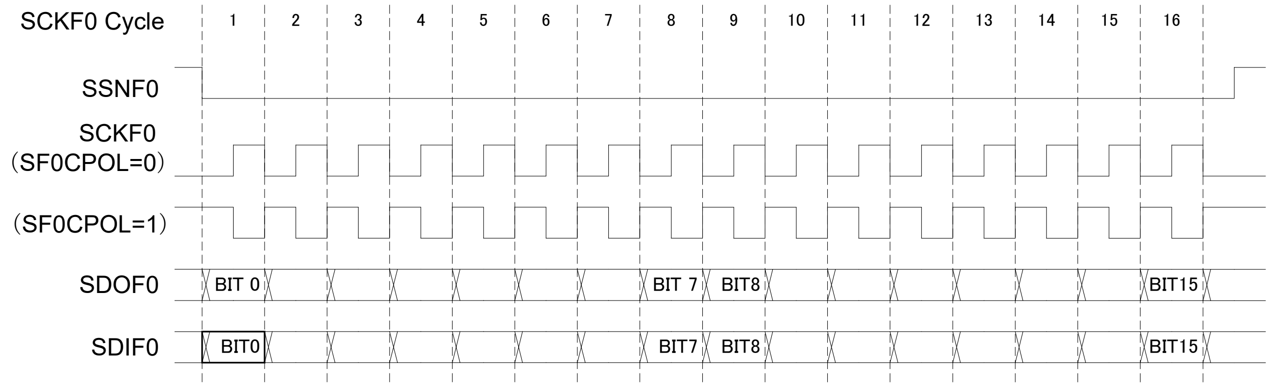


Figure 12-4 SSIOF Bus Waveform with SF0CPHA=0, SF0SIZ=1

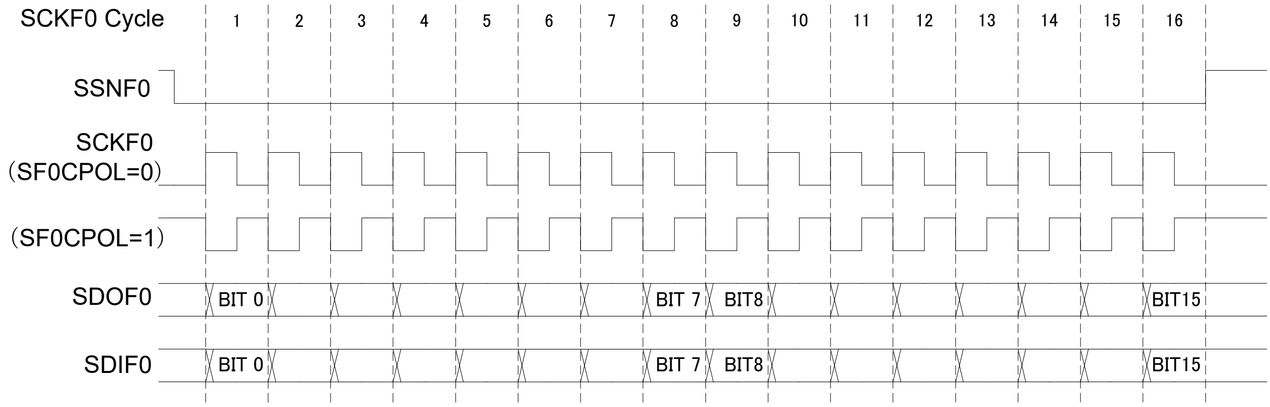


Figure 12-5 SSIOF Bus Waveform with SF0CPHA=1, SF0SIZ=1

12.3.6 Transfer Interval Setting (Master Mode)

The LEAD; SSF0-SCKF0 time, LAG; SCKF0-SSF0(H) time, and TDTL; SSF0(H)-SSF0(H) can be set to adjust the speed to the slave. This setting is only valid in Master mode. It is ignored in Slave mode. Setting during transferring is invalid.

1) LEAD
A value from 0.5 to 1.5 SCKF0 can be set.

2) LAG
A value from 0.5 to 1.5 SCKF0 can be set.

3) TDTL
The minimum transfer interval can be controlled in SCKF0 clocks by setting SF0DTL bit of the SF0TRAC register. If there is any transfer data in FIFO, the time set by this setting (SSF0) changes to "1" during byte/word transfer. If there is no transfer data in FIFO, this is "1" until any transmitted data is written. If SF0DTL bit of the SF0TRAC register is set to 0, the interval after transfer (TDTL) disappears and a continuous transfer is performed. SSF0 is held to 0 and returns to 1 after the transfer is finished.

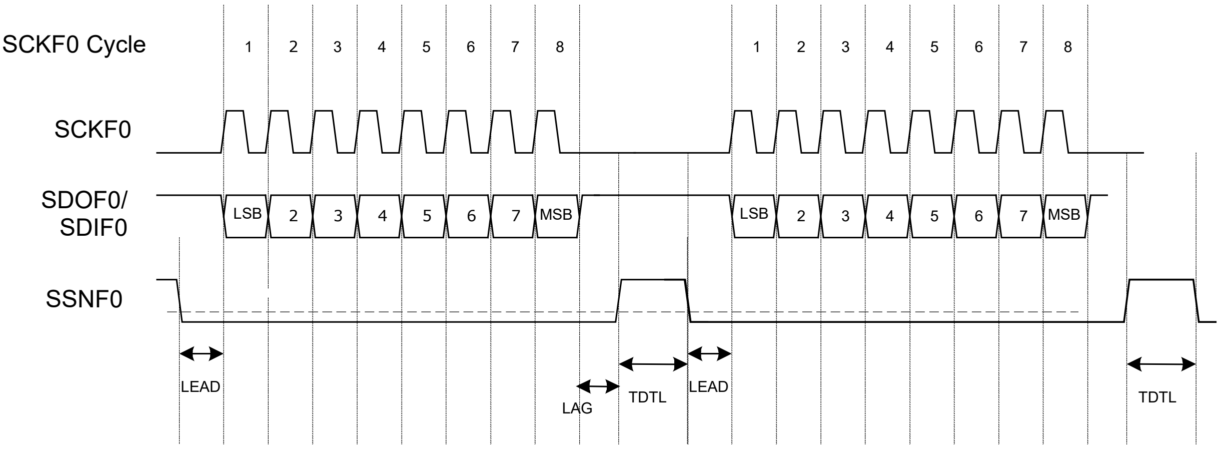


Figure 12-6 Transfer Interval without SF0DTL = "0"

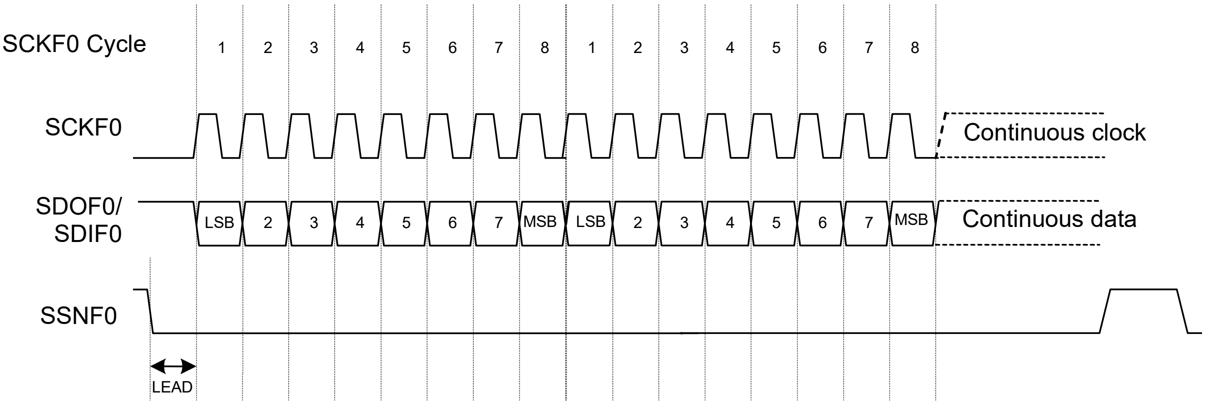


Figure 12-7 Transfer Interval with SF0DTL = "0"

12.3.7 Transmission Operation (Master Mode)

(1) Write the necessary values to SF0CTRL, SF0INTC, SF0BRR, and SF0TRAC, set the SF0MST bit to Master mode, and set the SF0SPE bit to enable the SSIOF transfer.

(2) When the transmitted data is written to SF0DWR, the transmit FIFO Empty flag changes to 0 (SF0TFE = 0). SSIOF starts the automatic transmission and outputs the transmitted data from LSB or MSB on the SOUTF0 pin according to the SF0LSB setting.

(3) The sync clock, which was set by the SF0CPOL, SF0CPHA, and SF0BRR registers, is output from the SCKF0 pin.

(4) Transmitted data can be written to SF0DWR successively. However, if further writing is performed when the transmit FIFO is in Full status (SF0TFF = 1), a write overflow occurs. (SF0WOF = 1, No interrupt is generated.)

(5) The SF0SPIF bit is set each time the transfer of 1 frame is completed. (SF0SPIF=1)

(6) A transmission interrupt occurs if the remaining data in the transmit FIFO matches the frame count selected with SF0TFIC. (SF0TFI=1)

(7) If the transmit FIFO becomes empty and the transfer of the last frame is completed, a transfer completion interrupt is generated. (SF0FI=1)

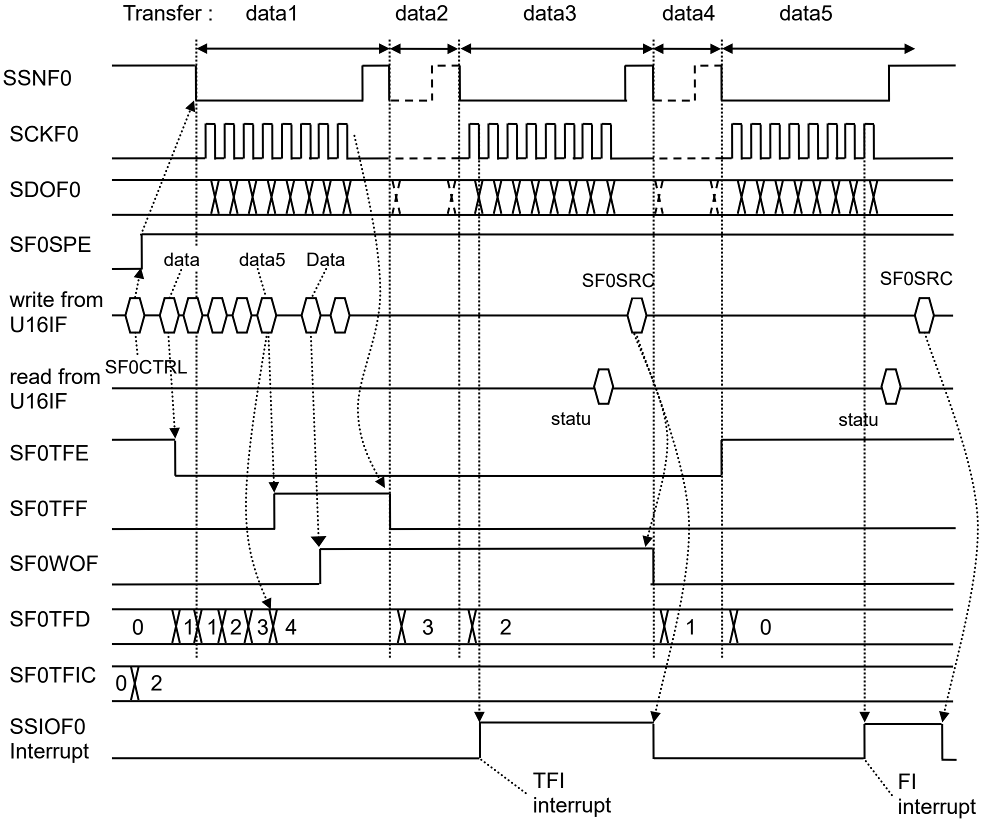


Figure 12-8 Transmission Operation in the Master Mode

12.3.8 Reception Operation (Master Mode)

The master mode of the synchronous serial with FIFO starts by setting data in a transmission buffer. The Data needs to be set into a transmission buffer even master mode reception only.

(1) Write the necessary values to SF0CTRL, SF0INTC, SF0BRR and SF0TRAC, and then set the SF0MST bit to the master mode, and set the SF0SPE bit to enable the SSIOF transfer.
(2) When the data is written to SF0DWR, the SSIOF transfer is started.
(3) The synchronous clock, which was set by the SF0CPOL, SF0CPHA, and SF0BRR0-1 registers, is output from the SCKF0 pin.
(4) On the SINF0 pin, the received data is sampled from LSB or MSB according to the SF0LSB setting and stored in the reception FIFO. The reception FIFO empty flag changes to 0 (RFE = 0).
(5) The SF0SPIF bit is set each time the transfer of 1 frame is completed. (SF0SPIF=1)
(6) If the number of data received in the reception FIFO is equal to or more than matches following the frame count selected with SF0RFIC of SF0CR, SF0RFI of SF0SRR is set to generate a reception interrupt. (SF0RFI=1)
(7) When the reception FIFO becomes full, the subsequent reception is disabled. If the reception is performed in this state, an overrun error interrupt is generated. (SF0ORF=1)
(8) If the temporary data of transmission FIFO becomes empty and the transfer of the last frame is completed, a transfer completion interrupt is generated. (SF0FI=1)

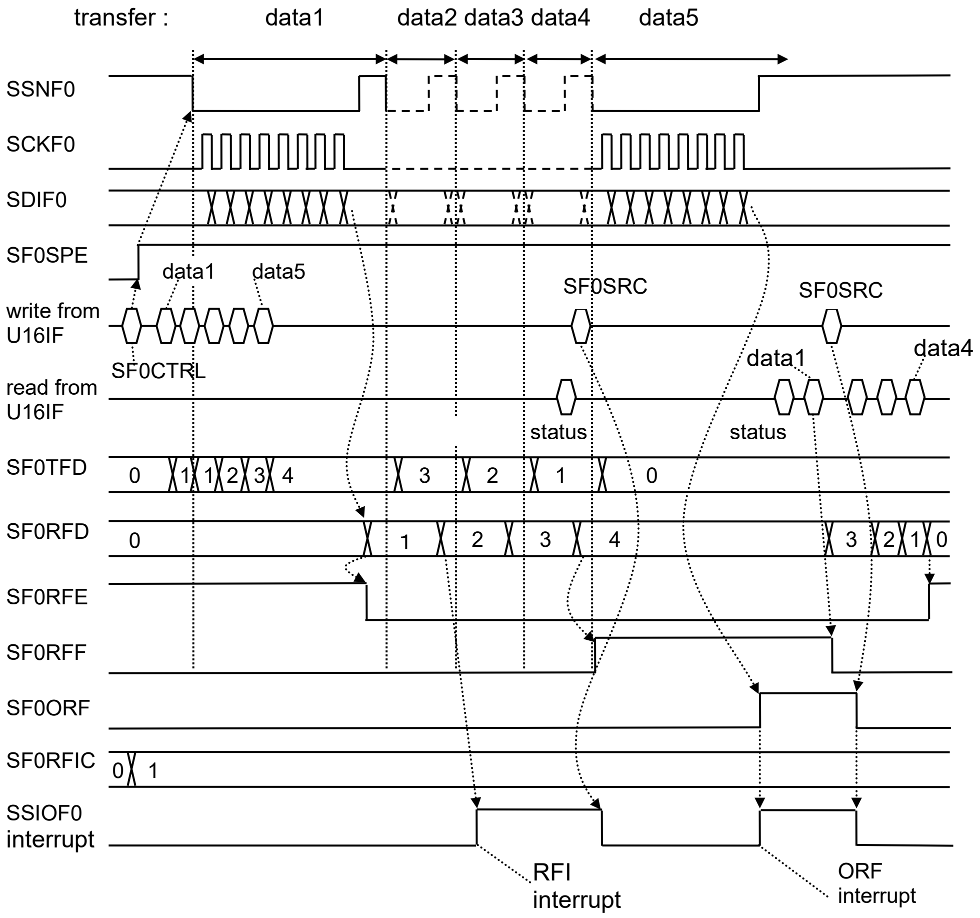


Figure 12-9 Reception Operation in the Master Mode

12.3.9 FIFO Operation

The SSIOF includes the receive FIFO of 16 words and the transmit FIFO of 16 words. The FIFO state is indicated in the SF0TFF, SF0TFE, SF0RFF, and SF0RFE bits of SF0SRR, and the SF0TFD and SF0RFD bit of SF0FSR.
There are three FIFO states, Full (SF0TFF and SF0RFF), Empty (SF0TFE and SF0RFE), and Depth (SF0TFD and SF0RFD).

12.3.10 Writing Overflow for Transmission

If further writing is performed when the transmit FIFO is in Full status (SF0TFF = 1), a write overflow is set. (SF0WOF=1). However, interrupt is not generated even when a write overflow occurs. SF0WOF is cleared when write “1” in SF0WOFC bit of SF0SRCH.

12.3.11 Overrun Error for Reception

If further reception is performed when the reception FIFO is in full status (SF0RFF = 1), an overrun error occurs. (SF0ORF=1)
If an overrun error occurs, the SF0ORF bit of SF0SRR is set, and an overrun error interrupt is generated. The newly received data is not held.
Read the content of the reception FIFO to clear the SF0RFF bit, then write "1" in the SF0ORFC bit to clear the SF0ORFC bit.

12.3.12 FIFO Clearing

The transmission/reception counter control of FIFO can be initialized to the initial setting state (SF0TFF=0, SF0TFE=1, SF0RFF=0, and SF0RFE=1 in the SF0SRR register and SF0TFD2-0=000 and SF0RFD2-0=000 in the SF0FSR register) by setting the SF0FICL bit of the SF0CTRL register to 1.
The SF0FICL bit of the SF0CTRL register needs to be 0, before next transfer operation.
Even if SF0FICL bit of SF0CTRL register is set to 1, the interrupt is not changed for SF0RFIC, SF0TFIC, SF0ORIE, SF0FIE, SF0RFIE, and SF0TFIE of the SF0INTC register, and SF0ORF, SF0FI, SF0RFI, and SF0TFI of the SF0SRR register.
This bit can be used to discard the data of FIFO when the communication is aborted.

12.3.13 Transfer When Slave Has Different Number of FIFO Transfer Frame

1) The master sends data only when the transmission data is already written in FIFO.
2) As the slave's transmission data count is determined by the master, data is transferred as follows if the number of FIFO transfer frame of slave is different from that of the master.
If the transmission data is not written in the slave's FIFO, a 0xFF ((0xFFFF) for word) is sent, including the state after a reset.

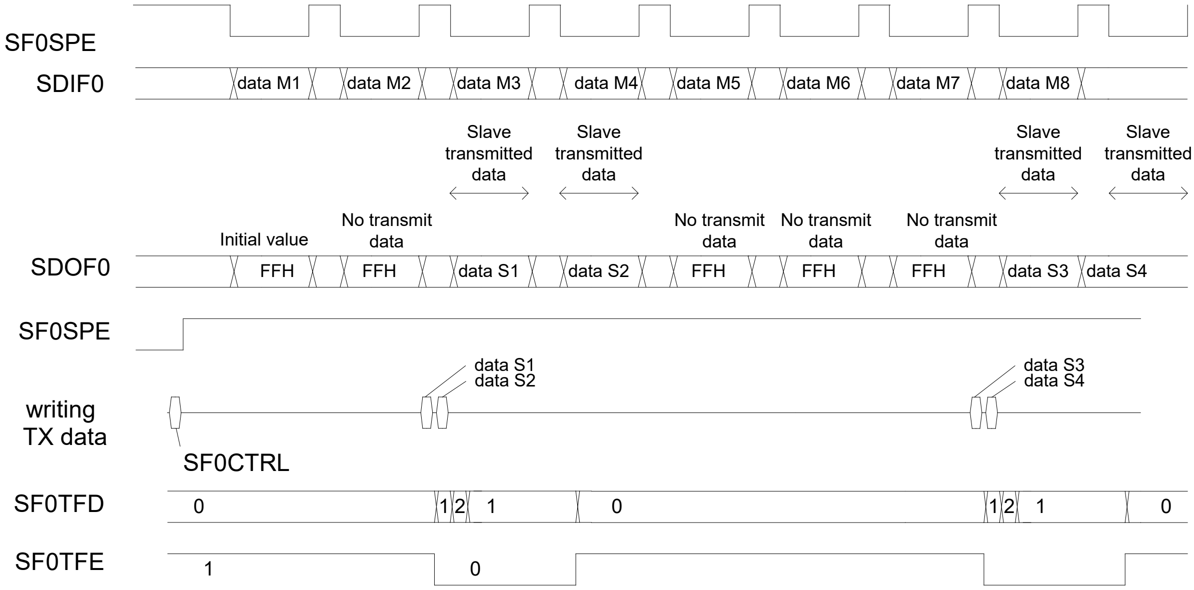


Figure 12-10 Transfer When Slave Has Different Number of FIFO Transfer Frame

12.3.14 Interrupt

12.3.14.1 SSIOF Interrupt Source

There are the following four types.

- Overrun
  If an overrun occurs, SF0ORF of SF0SRR is set, and an overrun error interrupt is generated.
- Transmission FIFO threshold
  If the remaining data of the transmission FIFO matches the frame count selected with SF0TFIC, SF0TFI of SF0SRR is set to generate a transmission interrupt.
- Reception FIFO threshold
  If the number of data received in the reception FIFO is equal to or more than following the frame count selected with SF0RFIC of SF0CR, SF0RFI of SF0SRR is set to generate a reception interrupt.
- Transfer completion
  If the transmission FIFO becomes empty and the transfer of the last frame is completed, SF0FI of SF0SRR is set to generate a transfer completion interrupt.

12.3.14.2 Clearing SSIOF Interrupt

An interrupt request is cleared by writing 1 to each interrupt bit (SF0TFC, SF0RFC, SF0FC, SF0ORFC, SF0MDFC, SF0SPIFC, and SF0WOFC) of the SF0SRR.

12.3.14.3 SSIOF Interrupt Timing

Figure 12-11 shows the interrupt timing.
The transmission interrupt (TFI) generates an interrupt in 3 to 4 SYSCLK after the shift clock of the second bit in the master mode, in 3 to 5 SYSCLK after the shift clock of the second bit in the slave mode.
For reception interrupt (RFI), transfer completion interrupt (FI), and overrun (ORF), an interrupt is generated in 2 to 3 SYSCLK after the sampling clock at the MSB in the master mode, in 2 to 4 SYSCLK after the sampling clock at the MSB in the slave mode.

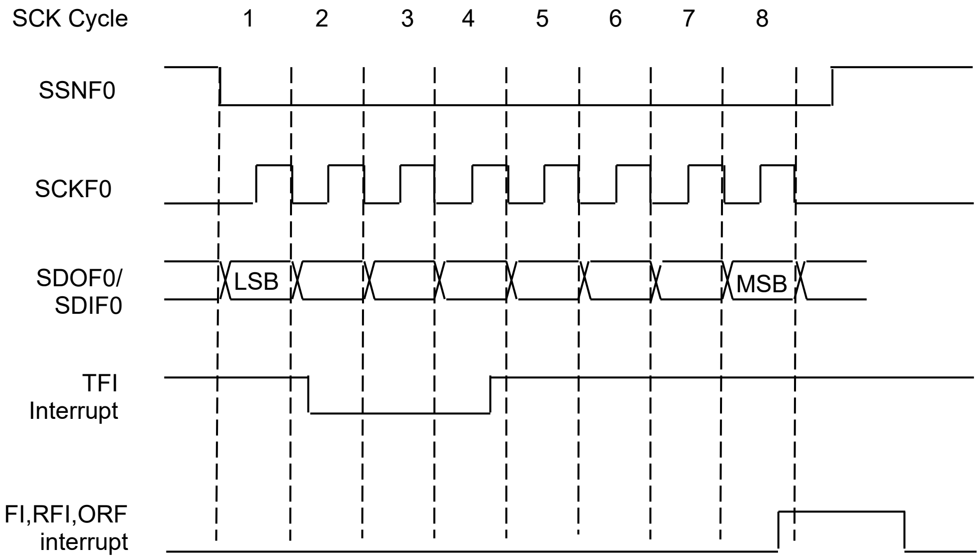


Figure 12-11 Interrupt Timing

12.3.14.4 Interrupt processing flow

Figure 12-12 show the processing flow in the receiving operation of the slave mode.

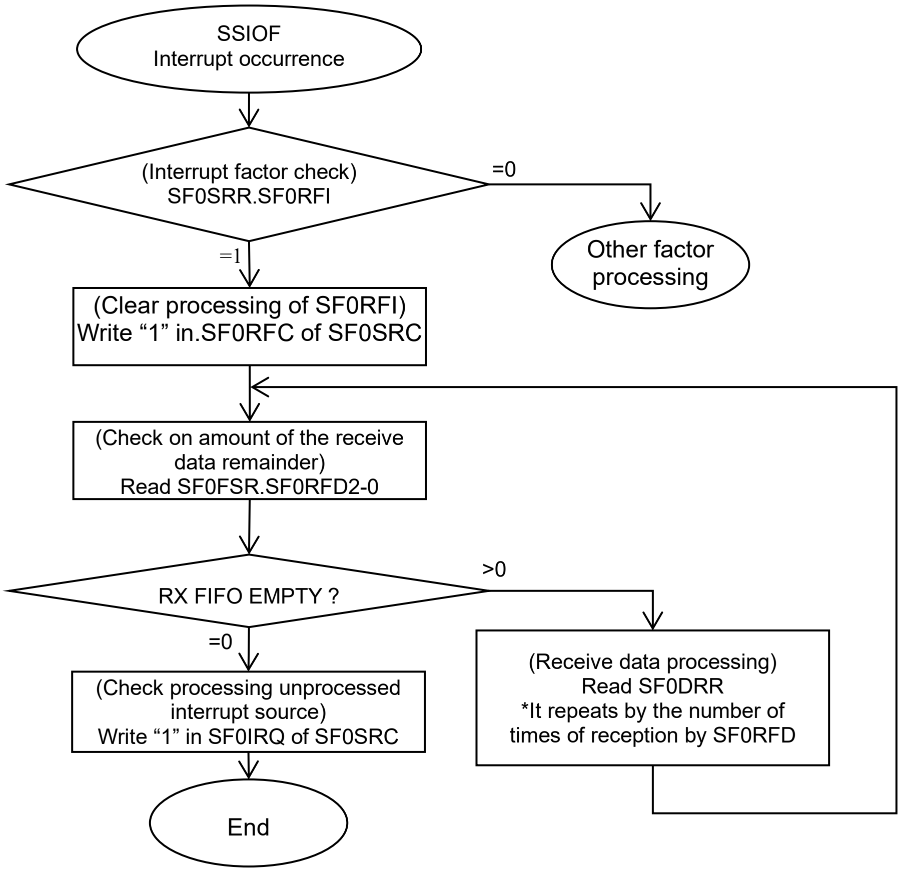

SSIOF
Interrupt occurrence
(Interrupt factor check)
SF0SRR.SF0RFI
=0
=1
Other factor
processing
(Clear processing of SF0RFI)
Write “1” in.SF0RFC of SF0SRC
(Check on amount of the receive
data remainder)
Read SF0FSR.SF0RFD2-0
RX FIFO EMPTY ?
>0
=0
(Receive data processing)
Read SF0DRR
*It repeats by the number of
times of reception by SF0RFD
(Check processing unprocessed
interrupt source)
Write “1” in SF0IRQ of SF0SRC
End

Figure 12-12 Example of the interrupt control flow

Chapter 13 I^{2}C Bus

13. I^{2}C Bus

13.1 General Description

ML62Q2500 group has one channel of I^{2}C bus unit that supports both master and slave function and one channel of I^{2}C bus master.
I^{2}C bus unit is that either of master or slave can be chosen to use and both functions of master and slave are unworkable at the same time.

13.1.1 Features

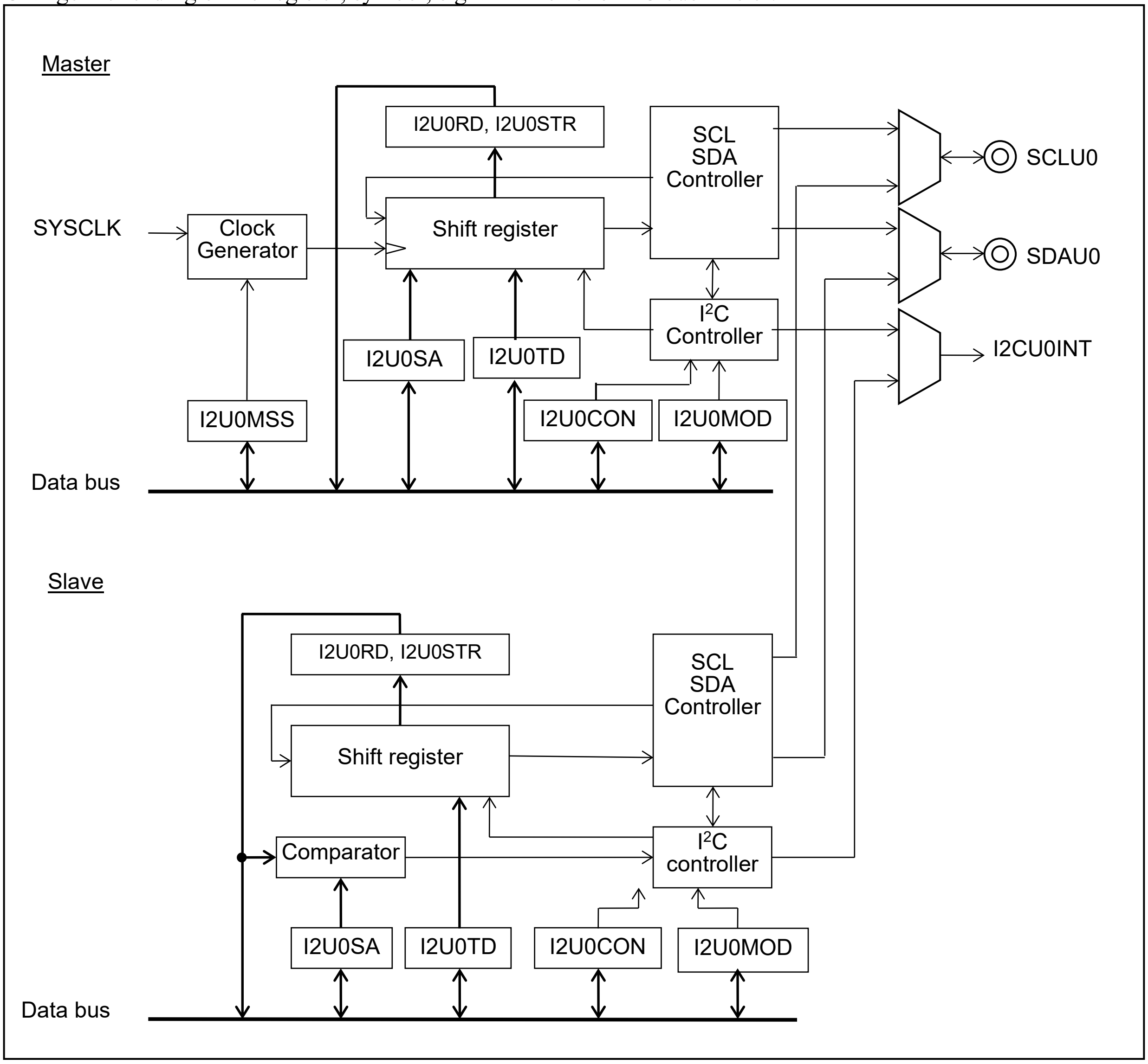
Table 13-1 shows features of I^{2}C bus unit and I^{2}C bus master.

Table 13-1 Features of I^{2}C bus

Function	Operation mode	Features
I^{2}C bus unit	Master function	- Communication speed: Standard mode (100 kbps), fast mode (400 kbps), and original standard 1 Mbps mode (1Mbps) - Support clock stretch function for the Slave - 7 or 10-bit address format - Self-test function by reading transmitted data onto the I^{2}C 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/STOP-D/HALT-D mode by matching slave address
I^{2}C bus master	Master function only	- Communication speed: Standard mode (100 kbps), fast mode (400 kbps), and original standard 1 Mbps mode (1Mbps) - Support clock stretch function for the Slave - 7 or 10-bit address format - Self-test function by reading transmitted data onto the I^{2}C bus (Safety function)

13.1.2 Configuration

Figure 13-1 shows the configuration diagram of the I2C bus unit circuit.
Change the reading of the register, symbol, signal name as for I2C bus master.



SCLU0	:	Serial clock
SDAU0	:	Serial data
I2U0MSS	:	I2C bus unit 0 mode register
I2U0RD	:	I2C bus unit 0 receive register
I2U0SA	:	I2C bus unit 0 slave address register
I2U0TD	:	I2C bus unit 0 transmit data register
I2U0CON	:	I2C bus unit 0 control register
I2U0MOD	:	I2C bus unit 0 mode register
I2U0STR	:	I2C bus unit 0 status register
I2U0SCLR	:	I2C bus unit 0 status clear register

Figure 13-1 Configuration of I2C Bus Unit

13.1.3 List of Pins

The I/O pins of the I2C bus unit are assigned to the shared function of the general ports.

Pin name	I/O	Description
SDAU0	I/O	I2C bus unit 0 data I/O pin
SCLU0	I/O	I2C bus unit 0 clock I/O pin
SDAM0	I/O	I2C bus master 0 data I/O pin
SCLM0	I/O	I2C bus master 0 clock I/O pin

Table 13-2 shows port used in the I2C and the register settings.
In addition to the mode setting of the shared function, choose "Enable Input, Enable Output, N-ch open drain output and without pull-up" by setting following data to the port n mode register m (PnMODm).

Table 13-2 Port used in the I2C and the register settings

Channel no.	Pin name	Shared port		Setting register	Setting value	ML62Q2500 group
0	SDAU0	P25	4th Func.	P2MOD5	0011_1011	●
		P71	4th Func.	P7MOD1	0011_1011	●
	SCLU0	P24	4th Func.	P2MOD4	0011_1011	●
		P70	4th Func.	P7MOD0	0011_1011	●
	SDAM0	P23	4th Func.	P2MOD3	0011_1011	●
		P73	4th Func.	P7MOD3	0011_1011	●
	SCLM0	P22	4th Func.	P2MOD2	0011_1011	●
		P72	4th Func.	P7MOD2	0011_1011	●

●: Available - : Unavailable

13.1.4 Combination of I2C Bus Port

SDAU0/SCLU0/SDAM0/SCLM0 pins are assigned to multiple general ports.
Be sure to use the ports in following combinations.

Combination	Peripheral Type	Channel no.	Port		ML62Q2500 group
			SDAUn/SDAMn	SCLUn/SCLMn	
1	I2C bus unit	0	P25	P24	●
2			P71	P70	●
3	I2C bus master	0	P23	P22	●
4			P73	P72	●

*: n= Channel number. ●: 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 is unsatisfied the I2C bus specification. See the data sheet for each product for the value of internal pull-up resistors.**
- **If powering off this LSI in the slave mode, it disables communications of other devices on the I2C bus. Keep this LSI powered on when it works as a slave mode until the master device is powered off.**
- **Do not connect multiple master devices on the I2C bus when using the master function.**

13.2 Description of Registers

13.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF780	I²C bus unit 0 mode register	I2U0MSS	-	R/W	8	0x00
0xF781	Reserved	-	-	-	-	-
0xF782	I²C bus unit 0 receive register	I2U0RD	-	R	8	0x00
0xF783	Reserved	-	-	-	-	-
0xF784	I²C bus unit 0 slave address register	I2U0SA	-	R/W	8	0x00
0xF785	Reserved	-	-	-	-	-
0xF786	I²C bus unit 0 transmit data register	I2U0TD	-	R/W	8	0x00
0xF787	Reserved	-	-	-	-	-
0xF788	I²C bus unit 0 control register	I2U0CON	-	R/W	8	0x00
0xF789	Reserved	-	-	-	-	-
0xF78A	I²C bus unit 0 mode register	I2U0MODL	I2U0MOD	R/W	8/16	0x00
0xF78B		I2U0MODH		R/W	8	0x02
0xF78C	I²C bus unit 0 status register	I2U0STAT	I2U0STR	R	8/16	0x00
0xF78D		I2U0ISR		R	8	0x00
0xF78E	I²C bus unit 0 status clear register	I2U0SCLRL	I2U0SCLR	W	8/16	0x00
0xF78F		I2U0SCLRH		W	8	0x00

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF7C0	Reserved	-	-	-	-	-
0xF7C1						
0xF7C2	I²C bus master 0 receive register	I2M0RD	-	R	8	0x00
0xF7C3	Reserved	-	-	-	-	-
0xF7C4	I²C bus master 0 slave address register	I2M0SA	-	R/W	8	0x00
0xF7C5	Reserved	-	-	-	-	-
0xF7C6	I²C bus master 0 transmit data register	I2M0TD	-	R/W	8	0x00
0xF7C7	Reserved	-	-	-	-	-
0xF7C8	I²C bus master 0 control register	I2M0CON	-	R/W	8	0x00
0xF7C9	Reserved	-	-	-	-	-
0xF7CA	I²C bus master 0 mode register	I2M0MODL	I2M0MOD	R/W	8/16	0x00
0xF7CB		I2M0MODH		R/W	8	0x02
0xF7CC	I²C bus master 0 status register	I2M0STAT	I2M0STR	R	8/16	0x00
0xF7CD		I2M0ISR		R	8	0x00
0xF7CE	I²C bus master 0 status clear register	I2M0SCLRL	I2M0SCLR	W	8/16	0x00
0xF7CF		I2M0SCLRH		W	8	0x00

13.2.2 I^{2}C Bus Unit 0 Mode Register (I2U0MSS)

This is a SFR used to choose the Master mode or Slave mode of the I2C bus unit 0.

Address : 0xF780 (I2U0MSS)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0MSS							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	I2U0MD
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	I2U0MD	This bit is used to choose the Master mode or Slave mode of the I^{2}C but unit. 0: Master mode (Initial value) 1: Slave mode

[Note]

- **All SFRs are shared in master mode and slave mode. If switching master/slave mode, set “0” I2U0EN bit of I2UMOD register, then Change mode and do reconfiguration each SFRs.**
- **When using the master function, do not connect multiple master devices on the I^{2}C 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.**

13.3 Description of Registers for Master function

This section explains about master mode of I^{2}C bus unit 0 and I^{2}C bus master 0.
In this section, both word symbol, byte symbol and bit symbol are put down with.
A prefix of symbol for I^{2}C bus unit 0 is I2U. A prefix of symbol for I^{2}C bus master 0 is I2M.
All SFRs are shared in master mode and slave mode. Set master mode on I2U0MSS, at the begining.

13.3.1 I^{2}C Bus Unit 0 Receive Register (I2U0RD), I^{2}C Bus Master 0 Receive Register (I2M0RD)

This is a read-only SFR used to store the received data.
This is initialized, in addition to reset function, by writing “0” to I2U0EN/I2M0EN bit in I2U0MOD/I2M0MOD register.

Address :	0xF782(I2U0RD), 0xF7C2(I2M0RD)
Access :	R
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0RD							
Bit	-	-	-	-	-	-	-	-	I2U0R7	I2U0R6	I2U0R5	I2U0R4	I2U0R3	I2U0R2	I2U0R1	I2U0R0
Word	-															
Byte	-								I2M0RD							
Bit	-	-	-	-	-	-	-	-	I2M0R7	I2M0R6	I2M0R5	I2M0R4	I2M0R3	I2M0R2	I2M0R1	I2M0R0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is updated after completion of each reception.

Bit No.	Bit symbol name	Description
7 to 0	I2U0R7 to I2U0R0/ I2M0R7 to I2M0R0	These bits are used to store the received data. This data is updated at coinciding slave-address and data reception. Reading this register enables following confirmation. - Reading when receiving data: Can confirm the received data. - Reading slave address or Reading when transmitting data: Can confirm the transmission data is surely transmitted.

13.3.2 I^{2}C Bus Unit 0 Slave Address Register (I2U0SA), I^{2}C Bus Master 0 Slave Address Register (I2M0SA)

This is a SFR to set the address and transmission/reception mode of the slave device.
This is initialized, in addition to reset function, by writing “0” to I2U0EN/I2M0EN bit in I2U0MOD/I2M0MOD register.

Address :	0xF784(I2U0SA), 0xF7C4(I2M0SA)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0SA							
Bit	-	-	-	-	-	-	-	-	I2U0A6	I2U0A5	I2U0A4	I2U0A3	I2U0A2	I2U0A1	I2U0A0	I2U0RW
Word	-															
Byte	-								I2M0SA							
Bit	-	-	-	-	-	-	-	-	I2M0A6	I2M0A5	I2M0A4	I2M0A3	I2M0A2	I2M0A1	I2M0A0	I2M0RW
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	I2U0A6 to I2U0A0/ I2M0A6 to I2M0A0	These bits are used to set the address of the communication partner.
0	I2U0RW/ I2M0RW	This bit is used to choose direction of the data communication. 0: Data transmission mode (Initial value) 1: Data reception mode

13.3.3 I^{2}C Bus Unit 0 Transmit Data Register (I2U0TD), I^{2}C Bus Master 0 Transmit Data Register (I2M0TD)

This is a SFR used to set the transmission data.
This is initialized, in addition to reset function, by writing “0” to I2U0EN/I2M0EN bit in I2U0MOD/I2M0MOD register.

Address :	0xF786(I2U0TD), 0xF7C6(I2M0TD)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0TD							
Bit	-	-	-	-	-	-	-	-	I2U0T7	I2U0T6	I2U0T5	I2U0T4	I2U0T3	I2U0T2	I2U0T1	I2U0T0
Word	-															
Byte	-								I2M0TD							
Bit	-	-	-	-	-	-	-	-	I2M0T7	I2M0T6	I2M0T5	I2M0T4	I2M0T3	I2M0T2	I2M0T1	I2M0T0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	I2U0T7 to I2U0T0/ I2M0T7 to I2M0T0	These bits are used to set the transmission data.

13.3.4 I^{2}C Bus Unit 0 Control Register (I2U0CON), I^{2}C Bus Master 0 Control Register (I2M0CON)

This is a SFR used to control transmission and reception operations.
This is initialized, in addition to reset function, by writing “0” to I2U0EN/I2M0EN bit in I2U0MOD/I2M0MOD register.

Address :	0xF788(I2U0CON), 0xF7C8(I2M0CON)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0CON							
Bit	-	-	-	-	-	-	-	-	I2U0ACT	-	-	-	-	I2U0RS	I2U0SP	I2U0ST
Word	-															
Byte	-								I2M0CON							
Bit	-	-	-	-	-	-	-	-	I2M0ACT	-	-	-	-	I2M0RS	I2M0SP	I2M0ST
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	W	W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	I2U0ACT/ I2M0ACT	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	I2U0RS/ I2M0RS	This bit is a write-only and used to request a restart. When "1" is written to this bit during data communication, this module shifts to the restart condition and the communication restarts from the slave address. "1" can be written to this bit only while communication is active (I2U0ST /I2M0ST = "1"). This bit always returns "0" for reading. 0: No restart request (Initial value) 1: Restart request
1	I2U0SP/ I2M0SP	This bit is a write-only and used to request a stop condition. When "1" is written to this bit, the module shifts to the stop condition and the communication stops. This bit always returns "0" for reading. 0: No stop condition request (Initial value) 1: Stop condition request
0	I2U0ST/ I2M0ST	This bit is used to control the communication operation of this module. When "1" is written to this bit during I2U0ST/I2M0ST bit is “0”, the communication starts. When "1" is overwritten to this bit in a next data transmission/reception wait state after transmission/reception of acknowledgment, the data transmission/reception restarts. When "0" is written to this bit, the communication is stopped forcibly. When "1" is written to this bit, the I2U0ST/I2M0ST bit is reset to “0”. 0: Stops communication (Initial value) 1: Starts communication

[Note]

- **Update it without a bit access instructions in the control register setting wait state.**
- **When the I2U0ST/I2M0ST bit is "1", write other bits of I2U0CON/I2M0CON register in the control register setting wait state.**

13.3.5 I^{2}C Bus Unit 0 Mode Register (I2U0MOD), I^{2}C Bus Master 0 Mode Register (I2M0MOD)

This is a SFR used to set the operation mode.
See “13.5.4 Operation Waveforms” for detail of communication speed.

Address :	0xF78A(I2U0MODL/I2U0MOD), 0xF78B(I2U0MODH), 0xF7CA(I2M0MODL/I2M0MOD), 0xF7CB(I2M0MODH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0200

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0MOD															
Byte	I2U0MODH								I2U0MODL							
Bit	I2U0TI2	I2U0TI1	I2U0TI0	-	-	-	I2U0CD1	I2U0CD0	-	-	I2U0MD4	I2U0MD3	I2U0MD2	I2U0MD1	I2U0MD0	I2U0EN
Word	I2M0MOD															
Byte	I2M0MODH								I2M0MODL							
Bit	I2M0TI2	I2M0TI1	I2M0TI0	-	-	-	I2M0CD1	I2M0CD0	-	-	I2M0MD4	I2M0MD3	I2M0MD2	I2M0MD1	I2M0MD0	I2M0EN
R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 13	I2U0TI2 to I2U0TI0/ I2M0TI2 to I2M0TI0	These bits are used to choose the timing parameter for an I^{2}C communication rate. See Table 13-3 of “13.5.4 Operation Waveforms” for detail. 000: a parameter for SYSCLK = 24MHz (Initial value) 001: a parameter for SYSCLK = 16MHz 010: a parameter for SYSCLK = 12MHz 011: a parameter for SYSCLK = 8MHz 100: a parameter for SYSCLK = 1MHz 101: a parameter for SYSCLK = LSCLK0 110: Do not use (for SYSCLK = 24MHz) 111: Do not use (for SYSCLK = 24MHz)
12 to 10	-	Reserved bits
9 to 8	I2U0CD1 to I2U0CD0/ I2M0CD1 to I2M0CD0	These bits are used to choose the I^{2}C operating clock. 00: SYSCLK 01: 1/2 SYSCLK 10: 1/4 SYSCLK (Initial value) 11: 1/8 SYSCLK See “13.5.4 Operation Waveforms” about communication speed.
7, 6	-	Reserved bits
5	I2U0MD4/ I2M0MD4	This bit is used to choose whether to or not to use the clock stretch (handshake) function. This function monitors the I^{2}C bus, therefore the communication speed gets lower depending on the load of I^{2}C bus. 0: Not use the clock stretch function (Initial value) 1: Use the clock stretch function
4, 3	I2U0MD3, I2U0MD2/ I2M0MD3, I2M0MD2	These bits are used to set the communication speed reduction rate. Specify these bits not so that the communication speed exceeds 100 kbps/400kbps/1 Mbps. When parameter for LSCLK0 or 1MHz is chosen by the I2U0TI2-0/I2M0TI2-0 bits, "No communication speed reduction" is chosen regardless this setting. See Table 13-3 of “13.5.4 Operation Waveforms” for detail. 00: No communication speed reduction (Initial value) 01: Approximately 10% communication speed reduction 10: Approximately 17% communication speed reduction 11: Approximately 24% communication speed reduction

Bit No.	Bit symbol name	Description
2, 1	I2U0MD1, I2U0MD0/ I2M0MD1, I2M0MD0/	These bits are used to set the communication speed mode. 00: Standard mode (Initial value) (100 kbps*) 01: Fast mode (400 kbps*) 10: 1Mbps mode (1Mbps*) 11: 1Mbps mode (1Mbps*) * : When SYSCLK=24 or 16 MHz and I2U0CD1-0/I2M0CD1-0 = “00” and I2U0MD4/I2M0MD4 = “0” and I2U0TI2-0/I2M0TI2-0 = “000/001” and I2U0MD3-2/I2M0MD3-2 = “00”.
0	I2U0EN/ I2M0EN	This bit is used to enable the master operation. When "1" is written to this bit, the I2U0ST/I2M0ST bit can be set. When "0" is written to this bit, the I²C master stops operation and the I2U-RD/I2M0RD, I2U0SA/I2M0SA, I2U0TD/I2M0TD, I2U0CON/I2M0CON and I2U0STR/I2M0STR are initialized. If this bit is written “0” during a communication, do initialization and re-setting. 0: Stop the I²C master operation (Initial value) 1: Enable the I²C master operation

13.3.6 I^{2}C Bus Unit 0 Status Register (I2U0STR), I^{2}C Bus Master 0 Status Register (I2M0STR)

This is a SFR to indicate the state of the I^{2}C bus unit / master.
This is initialized, in addition to reset function, by writing “0” to I2U0EN/I2M0EN bit in I2U0MOD/I2M0MOD register.
Each bit is initialized by writing “1” to a corresponding bit of I2U0SCLR/I2M0SCLR register.

Address :	0xF78C(I2U0STAT/I2U0STR), 0xF78D(I2U0ISR)
	0xF7CC(I2M0STAT/I2M0STR), 0xF7CD(I2M0ISR)
Access :	R
Access size :	8/16 bit
Initial value :	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0STR															
Byte	I2U0ISR								I2U0STAT							
Bit	-	-	-	-	-	I2U0SPS	I2U0DS	I2U0AS	rsvd	-	-	-	-	I2U0ER	I2U0ACR	-
Word	I2M0STR															
Byte	I2M0ISR								I2M0STAT							
Bit	-	-	-	-	-	I2M0SPS	I2M0DS	I2M0AS	rsvd	-	-	-	-	I2M0ER	I2M0ACR	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 11	-	Reserved bits
10	I2U0SPS/ I2M0SPS	This bit is used to indicate the usage state of the I^{2}C bus. This bit is set to "1" when transmitting the stop condition has been completed on the I^{2}C bus. To reset this bit, write "1" to I2U0CSPS/ I2M0CSP bit of I2U0SCLR/I2M0SCLR register. 0: The stop condition has not been transmitted (Initial value) 1: The stop condition has been transmitted
9	I2U0DS/ I2M0DS	This bit is used to indicate the usage state of the I^{2}C bus. This bit is set to "1" when transmitting data or receiving data has been completed on the I^{2}C bus. To reset this bit, write "1" toI2U0CDS/ I2M0CDS bit of I2U0SCLR/I2M0SCLR register. 0: The transmission/reception has not been completed (Initial value) 1: The transmission/reception has been completed
8	I2U0AS/ I2M0AS	This bit is used to indicate the usage state of the I^{2}C bus. This bit is set to "1" when transmitting the start condition and 7 bit slave address have been completed on the I^{2}C bus. To reset this bit, write "1" to I2U0CAS/ I2M0CAS bit of I2U0SCLR/I2M0SCLR register.. 0: The start condition and the slave address have not been transmitted (Initial value) 1: The start condition and the slave address have been transmitted
7	rsvd	Reserved bit
6 to 3	-	Reserved bits
2	I2U0ER/ I2M0ER	This bit is used to indicate a transmission error. When a bit of transmission data and the value on the SDAU0/SDAM0 pin do not coincide, "1" is set to this bit. To reset this bit, write "1" toI2U0CER/ I2M0CER bit of I2U0SCLR/I2M0SCLR register. When this bit is set to "1" and the clock stretch function is used (I2U0MD4/ I2M0MD4 = "1"), the SDAU0/SDAM0 pin output is disabled until the subsequent byte data communication terminates. Even if this bit is set to "1", the SDAU0/SDAM0 pin output continues until the subsequent byte data communication terminates when the clock stretch function is not used (I2U0MD4/ I2M0MD4 = "0"). 0: There was no transmission error (Initial value) 1: There was a transmission error

Bit No.	Bit symbol name	Description
1	I2U0ACR/ I2M0ACR	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 I2U0CACR/ I2M0CACR bit of I2U0SCLR/I2M0SCLR register.. 0: Received acknowledgment "0" (Initial value) 1: Received acknowledgment "1"
0	-	Reserved bit

13.3.7 I^{2}C Bus Unit 0 Status Clear Register (I2U0SCLR), I^{2}C Bus Master 0 Status Clear Register (I2M0SCLR)

This is a SFR to clear the state of the I^{2}C bus unit / master.
When Each bit is written “1”, a corresponding bit of I2U0STR/I2M0STR register is initialized to “0”.

Address :	0xF78E(I2U0SCLRL/I2U0SCLR), 0xF78F(I2U0SCLRH)
	0xF7CE(I2M0SCLRL/I2M0SCLR), 0xF7CF(I2M0SCLRH)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0SCLR															
Byte	I2U0SCLRH								I2U0SCLRL							
Bit	-	-	-	-	-	I2U0C SPS	I2U0C DS	I2U0C AS	-	-	-	-	-	I2U0C ER	I2U0C ACR	-
Word	I2M0SCLR															
Byte	I2M0SCLRH								I2M0SCLRL							
Bit	-	-	-	-	-	I2M0C SPS	I2M0C DS	I2M0C AS	-	-	-	-	-	I2M0C ER	I2M0C ACR	-
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is used to clear a target status.
Writing “0”: Invalid
Writing “1”: clear a target interrupt

Bit No.	Bit symbol name	Description (target)
15 to 11	-	Reserved bits
10	I2U0CSPS/ I2M0CSPS	I2U0SPS bit of I2U0STR register/ I2M0SPS bit of I2M0STR register
9	I2U0CDS/ I2M0CDS	I2U0DS bit of I2U0STR register/ I2M0DS bit of I2M0STR register
8	I2U0CAS/ I2M0CAS	I2U0AS bit of I2U0STR register/ I2M0AS bit of I2M0STR register
7 to 3	-	Reserved bit
2	I2U0CER/ I2M0CER	I2U0ER bit of I2U0STR register/ I2M0ER bit of I2M0STR register
1	I2U0CACR/ I2M0CACR	I2U0ACR bit of I2U0STR register/ I2M0ACR bit of I2M0STR register
0	-	Reserved bit

13.4 Description of Registers for Slave function

This section explains about slave mode of I^{2}C bus unit 0.
A prefix of symbol for I^{2}C bus unit 0 is I2U. A prefix of symbol for I^{2}C bus master 0 is I2M.
Set master mode on I2U0MSS, at the begining I^{2}C bus unit 0. I^{2}C bus master does not have this function.

13.4.1 I^{2}C Bus Unit 0 Receive Register (I2U0RD)

This is a read-only SFR used to store the received data.

Address :	0xF782(I2U0RD)
Access :	R
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0RD							
Bit	-	-	-	-	-	-	-	-	I2U0R7	I2U0R6	I2U0R5	I2U0R4	I2U0R3	I2U0R2	I2U0R1	I2U0R0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7to0	I2U0R7toI2U0R0	These bits are used to store the received data. This data is updated at coinciding slave-address and data reception. Reading this register enables following confirmation. - Reading when receiving data: Can confirm the received data. - Reading when transmitting data: Can confirm the transmission data is surely transmitted.

13.4.2 I²C Bus Unit 0 Slave Address Register (I2U0SA)

This is a SFR used to set the slave address.

Address : 0xF784(I2U0SA)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0SA							
Bit	-	-	-	-	-	-	-	-	I2U0A6	I2U0A5	I2U0A4	I2U0A3	I2U0A2	I2U0A1	I2U0A0	-
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 1	I2U0A6 to I2U0A0	These bits are used to set the slave address.
0	-	Reserved bit

13.4.3 I²C Bus Unit 0 Transmit Data Register (I2U0TD)

This is a SFR used to set the transmission data.

Address : 0x786(I2U0TD)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0TD							
Bit	-	-	-	-	-	-	-	-	I2U0T7	I2U0T6	I2U0T5	I2U0T4	I2U0T3	I2U0T2	I2U0T1	I2U0T0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 0	I2U0T7 to I2U0T0	These bits are used to set the transmission data.

13.4.4 I^{2}C Bus Unit 0 Control Register (I2U0CON)

This is a SFR used to control transmission and reception operations.

Address :	0xF788(I2U0CON)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								I2U0CON							
Bit	-	-	-	-	-	-	-	-	I2U0ACT	-	I2U0WT	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R/W	R	W	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7	I2U0ACT	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	I2U0WT	This bit is used to release the communication wait state ("L" level output on the SCLU0 pin) in the slave mode. Writing "1" to this bit during the communication wait state releases the state ("L" level output of the SCLU0 pin is released). This bit is a write-only bit and always returns "0" for reading. 0: Not release the communication wait state (Initial value) 1: Release the communication wait state
4 to 0	-	Reserved bits

[Note]

- **If system clock is extremely slower than the communication speed, the data transmission/reception can be failed.**
- **Before releasing the communication wait state, change the system clock enough speed for the communication.**

13.4.5 I^{2}C Bus Unit 0 Mode Register (I2U0MOD)

This is a SFR used to set the operation mode.

Address :	0xF78A(I2U0MODL/I2U0MOD), 0xF78B(I2U0MODH),
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0MOD															
Byte	I2U0MODH								I2U0MODL							
Bit	-	-	-	-	-	-	-	-	-	I2U0MD5	I2U0MD4	I2U0MD3	I2U0MD2	I2U0MD1	-	I2U0EN
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W
Initial value	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	I2U0MD5	This bit is used to enable or disable the start condition interrupt in the slave mode. 0: Disabled (Initial value) 1: Enabled
5	I2U0MD4	This bit is used to enable or disable the stop condition interrupt in the slave mode. 0: Disabled (Initial value) 1: Enabled
4	I2U0MD3	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: Disabled (Initial value) 1: Enabled
3	I2U0MD2	This bit is used to enable or disable the communication wait function of the I^{2}C bus unit (output "L" level on the SCLU0 pin) when transmitting to the master and receiving the acknowledge data "1" from the master. Set this bit to "1" to use the communication wait function. 0: Disabled (Initial value) 1: Enabled
2	I2U0MD1	This bit is used to select mode when the stop condition interrupt is enabled (I2U0MD4=1). This function performs detecting the status of I2US0SAA bit. Do not clear the I2US0SAA bit by the software when enabling the interrupt. 0: The interrupt occur while the master is communicating with self-slave or other slaves (Initial value) 1: The interrupt occur while the master is communicating with only self-slave
1	-	Reserved bit
0	I2U0EN	This bit is used to enable the slave operation of the I^{2}C bus unit. When "1" is written to this bit, the operation of the I^{2}C bus unit 0 is enabled. When "0" is written to this bit, all the bits of the I^{2}C bus status register (I2US0STR) are initialized to "0", and the operation of the I^{2}C bus unit 0 is stopped. 0: Stop the I^{2}C slave operation (Initial value) 1: Enable the I^{2}C slave operation

[Note]

- **To be disable the wake-up from standby mode by matching the slave address, Stop the operation by resetting I2U0EN bit to "0" before entering STOP/STOP-D/HALT-D mode.**

13.4.6 I^{2}C Bus Unit 0 Status Register (I2U0STR)

This is a SFR to indicate the state of the I^{2}C bus unit.
Each bit is initialized, in addition to reset function, by writing “1” to a corresponding bit of I2U0SCLR/I2M0SCLR register.

Address : 0xF78C(I2U0STAT/I2U0STR), 0xF78D(I2U0ISR)
Access : R
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0STR															
Byte	I2U0ISR								I2U0STAT							
Bit	-	-	I2U0ASNA	I2U0RAS	I2U0STS	I2U0SPS	I2U0DS	I2U0AS	-	-	-	I2U0TR	I2U0SAA	I2U0ER	I2U0ACR	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 14	-	Reserved bits
13	I2U0ASNA	This bit is used to indicate status of reception. It is set when a coinciding slave-address and transmitting “H” as acknowledge occurs in the STOP-D/HALT-D mode. To reset this bit, write "1" to I2U0CASNA bit of I2U0SCLR register. 0: It has not occur. (Initial value) 1: It occurred
12	I2U0RAS	This bit is used to indicate status of the interrupt when enabling the start condition interrupt (I2U0MD3 bit = 1). To reset this bit, write "1" to I2U0CRAS bit of I2U0SCLR register. 0: Unmatched the slave address is detected after the start condition (Initial value) 1: Unmatched the slave address is detected after the start condition
11	I2U0STS	This bit is used to indicate status of transmission and reception. This bit is set to "1" when receiving the start condition. This bit is available when I2U0MD5 bit is “1”. To reset this bit, write "1" to I2U0CSTS bit of I2U0SCLR register. 0: The start condition has not been received (Initial value) 1: The start condition has been received
10	I2U0SPS	This bit is used to indicate status of transmission and receive. This bit is set to "1" when receiving the stop condition. This bit is available when I2U0MD4 bit is “1”. To reset this bit, write "1" to I2U0CSPS bit of I2U0SCLR register. 0: The stop condition has not been received (Initial value) 1: The stop condition has been received
9	I2U0DS	This bit is used to indicate status of transmission and reception. This bit is set to "1" when transmitting or receiving data on the condition of that slave address is matched. However, this bit does not become to “1” when it is happened wake-up from STOP-D/HALT-D mode by the address matching. To reset this bit, write "1" to I2U0CDS bit of I2U0SCLR register. 0: The data has not been transmitted or received (Initial value) 1: The data has been transmitted or received
8	I2U0AS	This bit is used to indicate status of transmission and reception. This bit is set to "1" when receiving the slave address data and it is matched. To reset this bit, write "1" to I2U0CAS bit of I2U0SCLR register. 0: The slave address has not been received or it is not matched (Initial value) 1: The slave address has been received and it is matched
7 to 5	-	Reserved bits
4	I2U0TR	This bit is used to indicate the transmitting or receiving state. This bit is set to "1" when detecting the data reception mode. This bit is reset to "0" when detecting a stop condition or detecting the data transmission mode. To reset this bit, write "1" to I2U0CTR bit of I2U0SCLR register. 0: Receiving state (Initial value) 1: Transmitting state

Bit No.	Bit symbol name	Description
3	I2U0SAA	This bit is used to indicate that this device is specified as a slave address. This bit is set to "1" when the content of the slave address output by the master device coincides with the contents of I2US0SA register. However it is not set to "1" when wake-up from STOP-D/HALT-D mode even if the slave address is matching. This bit is reset to "0" when a stop condition is received or when "1" is written to I2U0CSAA bit of I2U0SCLR register. 0: Not coincide with the slave address (Initial value) 1: Coincides with the slave address
2	I2U0ER	This bit is used to indicate a transmission error. When the value of the bit transmitted and the value of the SDAU0 pin do not coincide, this bit is set to "1". When this bit is set to "1", the SDAU0 pin output is disabled until the subsequent byte data communication terminates. To reset this bit, write "1" toI2U0CER bit of I2U0SCLR register. 0: There was no transmission error (Initial value) 1: There was a transmission error
1	I2U0ACR	This bit is used to store an acknowledgment signal received. The acknowledgment signals are received each time the slave address is received and data transmission or reception is completed. To reset this bit, write "1" to I2U0CTR bit of I2U0SCLR register. 0: Received the acknowledgment "0" (Initial value) 1: Received the acknowledgment "1"
0	-	Reserved bit

13.4.7 I^{2}C Bus Unit 0 Status Clear Register (I2U0SCLR)

This is a SFR to clear the state of the I^{2}C bus unit.
When Each bit is written “1”, a corresponding bit of I2U0STR/I2M0STR register is initialized to “0”.

Address : 0xF78E(I2U0SCLRL/I2U0SCLR), 0xF78F(I2U0SCLRH)
Access : W
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	I2U0SCLR															
Byte	I2U0SCLRH								I2U0SCLRL							
Bit	-	-	I2U0C ASNA	I2U0C RAS	I2U0C STS	I2U0C SPS	I2U0C DS	I2U0C AS	-	-	-	I2U0C TR	I2U0C SAA	I2U0C ER	I2U0C ACR	-
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is used to clear a target status.
Writing “0”: Invalid
Writing “1”: clear a target interrupt

Bit No.	Bit symbol name	Description (Target status)
15 to 14	-	Reserved bits
13	I2U0CASNA	I2U0ASNA bit of I2U0STR register
12	I2U0CRAS	I2U0RAS bit of I2U0STR register
11	I2U0CSTS	I2U0STS bit of I2U0STR register
10	I2U0CSPS	I2U0SPS bit of I2U0STR register
9	I2U0CDS	I2U0DS bit of I2U0STR register
8	I2U0CAS	I2U0AS bit of I2U0STR register
7 to 5	-	Reserved bits
4	I2U0CTR	I2U0TR bit of I2U0STR register
3	I2U0CSAA	I2U0SAA bit of I2U0STR register
2	I2U0CER	I2U0ER bit of I2U0STR register
1	I2U0CACR	I2U0ACR bit of I2U0STR register
0	-	Reserved bit

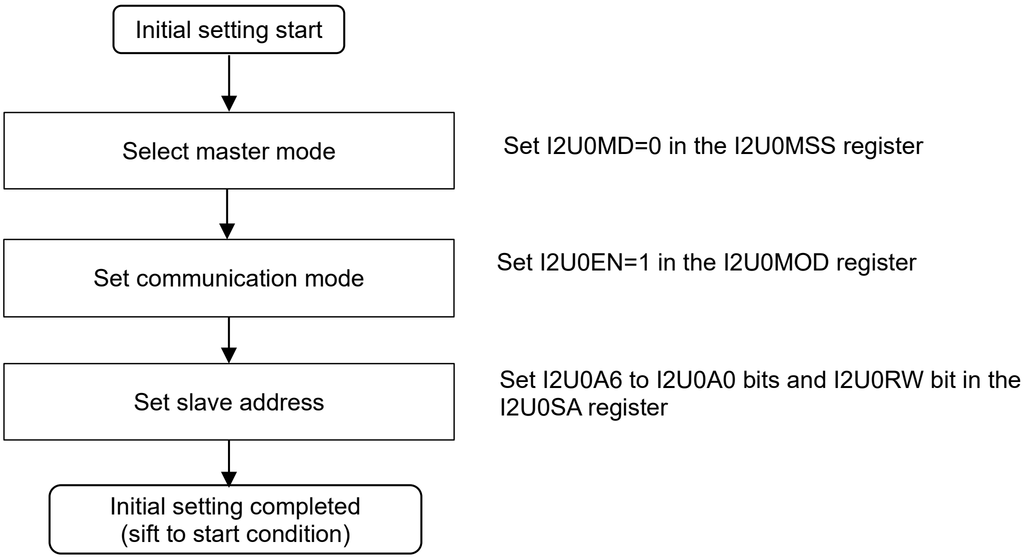
13.5 Description of Operation for Master function

The following explains for I^{2}C bus unit 0. Change the reading of the register, symbol, signal name as for I^{2}C bus master.

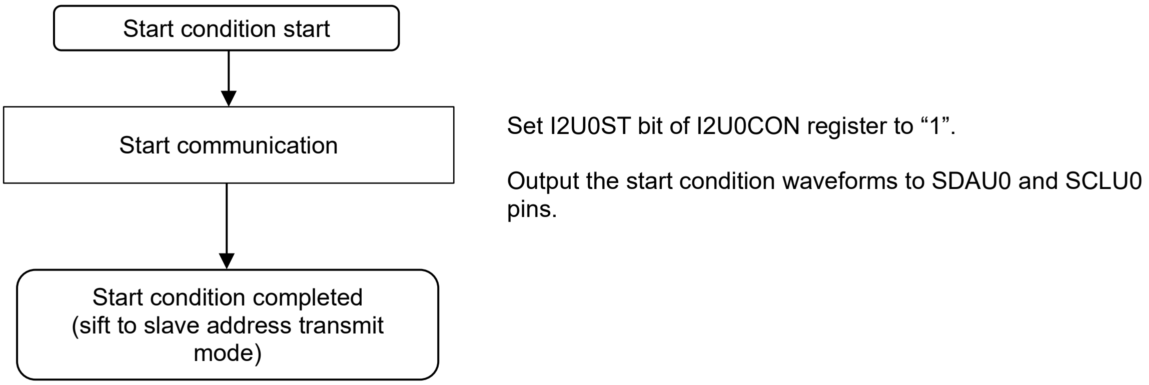
13.5.1 Control Procedures

The following flow charts describe procedures of each operation in the master mode.

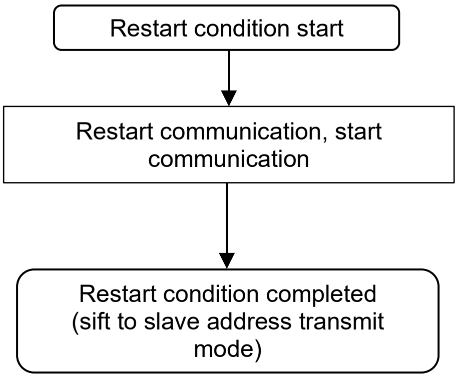
13.5.1.1 Initial Setting of Communication Operation


Initial setting start
Select master mode
Set I2U0MD=0 in the I2U0MSS register
Set communication mode
Set I2U0EN=1 in the I2U0MOD register
Set slave address
Set I2U0A6 to I2U0A0 bits and I2U0RW bit in the I2U0SA register
Initial setting completed (sift to start condition)

13.5.1.2 Start Condition


Start condition start
Start communication
Set I2U0ST bit of I2U0CON register to “1”.
Output the start condition waveforms to SDAU0 and SCLU0 pins.
Start condition completed (sift to slave address transmit mode)

13.5.1.3 Restart Condition

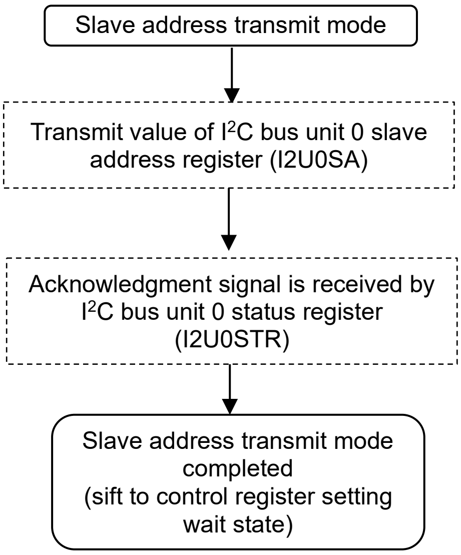

Restart condition start
Restart communication, start communication
Restart condition completed (sift to slave address transmit mode)

Communication in progress (I2U0ST=1)

Set I2U0RS=1 and I2U0ST=1 in the I2U0CON register

Output restart condition waveforms to SDAU0 and SCLU0 pins.

13.5.1.4 Slave Address Transmission Mode

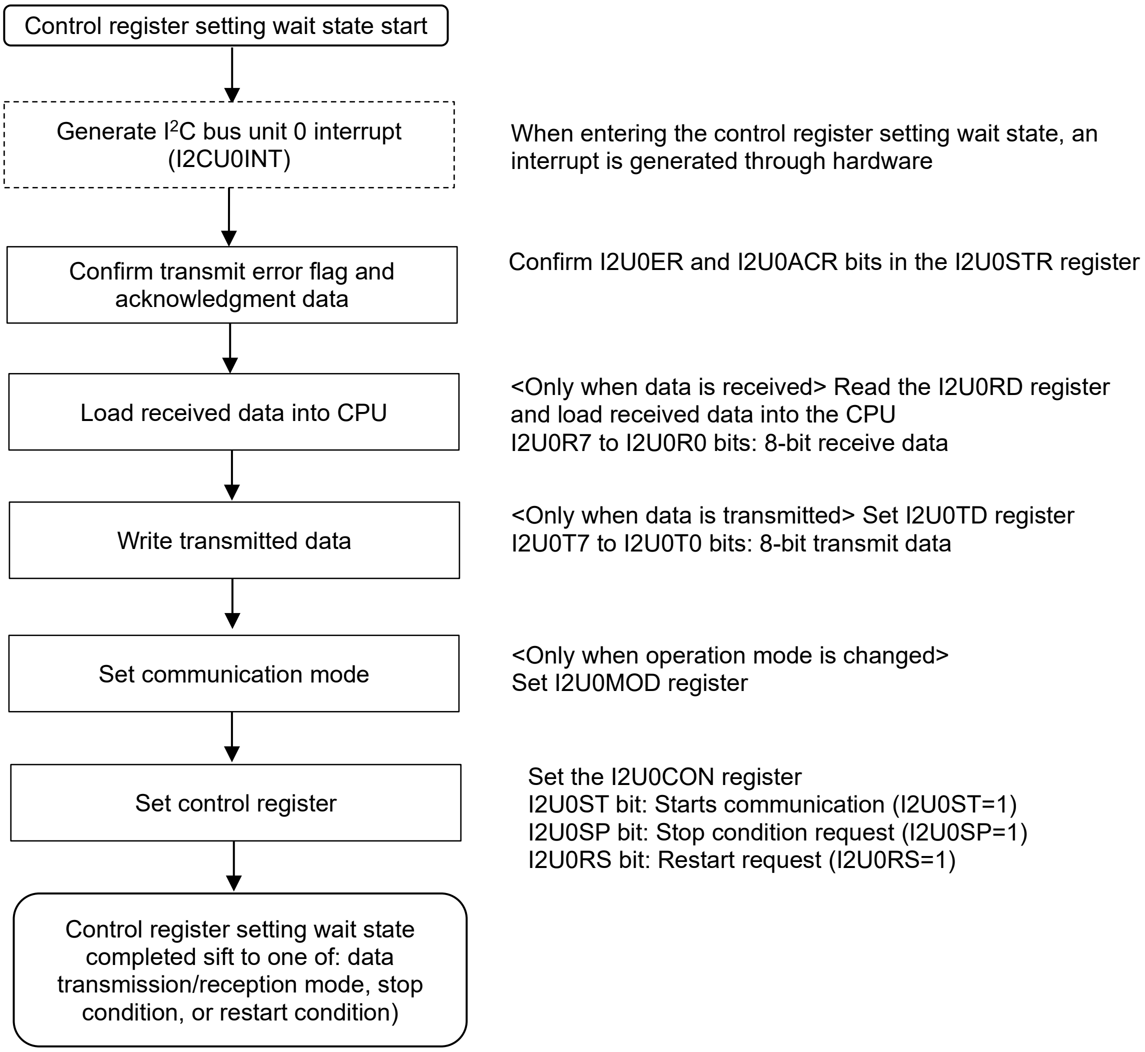

Slave address transmit mode
Transmit value of I²C bus unit 0 slave address register (I2U0SA)
Acknowledgment signal is received by I²C bus unit 0 status register (I2U0STR)
Slave address transmit mode completed (sift to control register setting wait state)

The value is transmitted from SDAU0 pin in MSB first through hardware following the start condition
  I2U0A6 to I2U0A0 bits: Slave address
  I2U0RW: Data direction (transmission/reception)
Value transmitted from the SDAU0 pin is stored in the I2U0RD register

Acknowledgment signal is received through hardware
I2UM0ACR bit: Acknowledgment data

I²C bus 0 control register (I2UM0CON) setting wait state

13.5.1.5 Control Register Setting Wait State


Control register setting wait state start
Generate I2C bus unit 0 interrupt (I2CU0INT)
Confirm transmit error flag and acknowledgment data
Load received data into CPU
Write transmitted data
Set communication mode
Set control register
Control register setting wait state completed sift to one of: data transmission/reception mode, stop condition, or restart condition)

When entering the control register setting wait state, an interrupt is generated through hardware

Confirm I2U0ER and I2U0ACR bits in the I2U0STR register

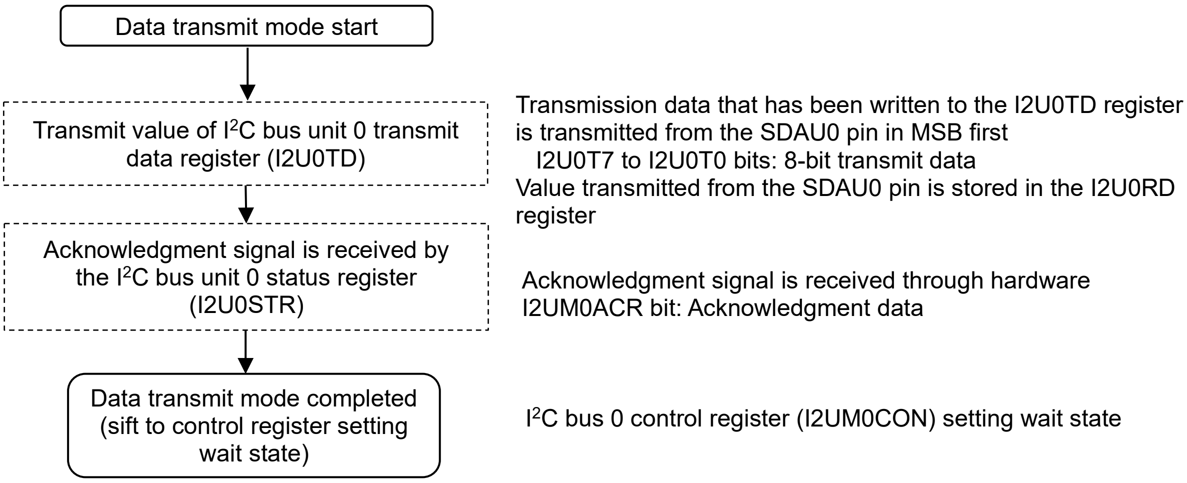
<Only when data is received> Read the I2U0RD register and load received data into the CPU
I2U0R7 to I2U0R0 bits: 8-bit receive data

<Only when data is transmitted> Set I2U0TD register
I2U0T7 to I2U0T0 bits: 8-bit transmit data

<Only when operation mode is changed>
Set I2U0MOD register

Set the I2U0CON register
I2U0ST bit: Starts communication (I2U0ST=1)
I2U0SP bit: Stop condition request (I2U0SP=1)
I2U0RS bit: Restart request (I2U0RS=1)

13.5.1.6 Data Transmission Mode

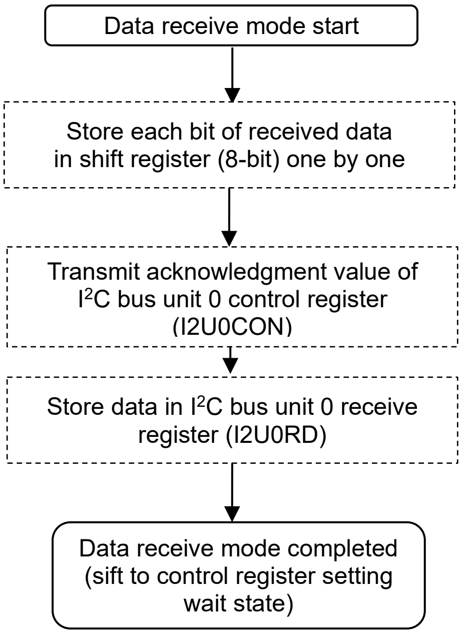

Data transmit mode start
Transmit value of I2C bus unit 0 transmit data register (I2U0TD)
Acknowledgment signal is received by the I2C bus unit 0 status register (I2U0STR)
Data transmit mode completed (sift to control register setting wait state)

Transmission data that has been written to the I2U0TD register is transmitted from the SDAU0 pin in MSB first
I2U0T7 to I2U0T0 bits: 8-bit transmit data
Value transmitted from the SDAU0 pin is stored in the I2U0RD register

Acknowledgment signal is received through hardware
I2UM0ACR bit: Acknowledgment data

I2C bus 0 control register (I2UM0CON) setting wait state

13.5.1.7 Data Reception Mode


Data receive mode start
Store each bit of received data in shift register (8-bit) one by one
Transmit acknowledgment value of I2C bus unit 0 control register (I2U0CON)
Store data in I2C bus unit 0 receive register (I2U0RD)
Data receive mode completed (sift to control register setting wait state)

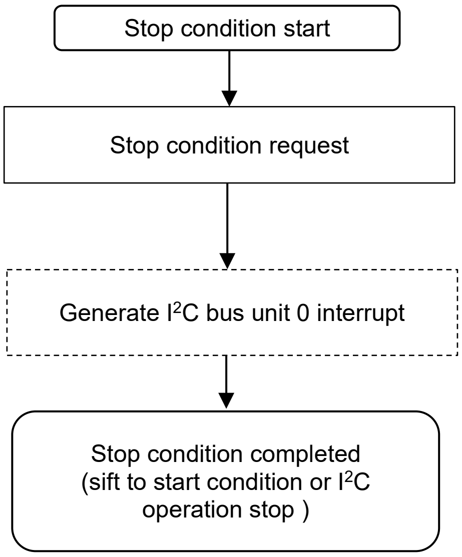
Value (received data) input to SDAU0 pin is stored in synchronization with rising edge of transfer clock input to SCLU0 pin in MSB first

Acknowledgment signal is transmitted through hardware
I2U0ACT bit: Acknowledgment value
Transmitted acknowledgment value is stored in the I2U0ACR bit of the I2U0STA register

Received data is stored from the shift register after acknowledgment signal is transmitted
I2U0R7 to I2U0R0 bits: 8-bit receive data

I2C bus unit 0 control register (I2U0CON) setting wait state

13.5.1.8 Stop Condition


Stop condition start
Stop condition request
Generate I2C bus unit 0 interrupt
Stop condition completed (sift to start condition or I2C operation stop)

Set I2U0SP bit of I2U0CON register to “1”.

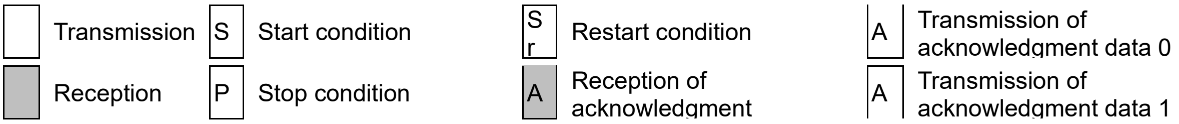
Output stop condition waveforms to SDAU0 and SCLU0 pins.

After the stop condition waveform is output, an interrupt is generated through hardware

Sift to start condition or I2C operation stop (I2U0EN = 0)

13.5.2 Communication Operation Timing

Figures 13-2 to 13-4 show the operation timing and control method for each communication mode during the master operation.


Transmission
Reception
S Start condition
P Stop condition
Sr Restart condition
A Reception of acknowledgment
A Transmission of acknowledgment data 0
A Transmission of acknowledgment data 1

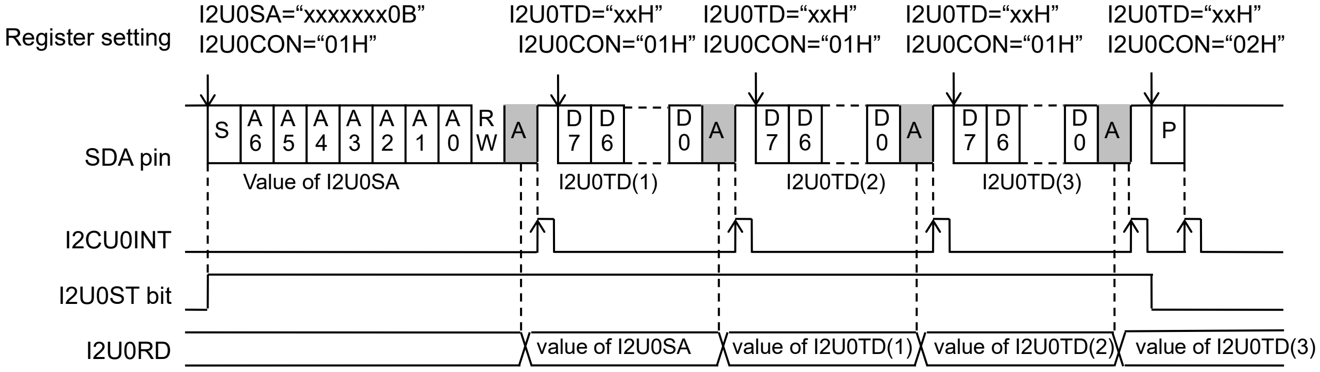


Figure 13-2 Operation timing during data transmission in the master mode

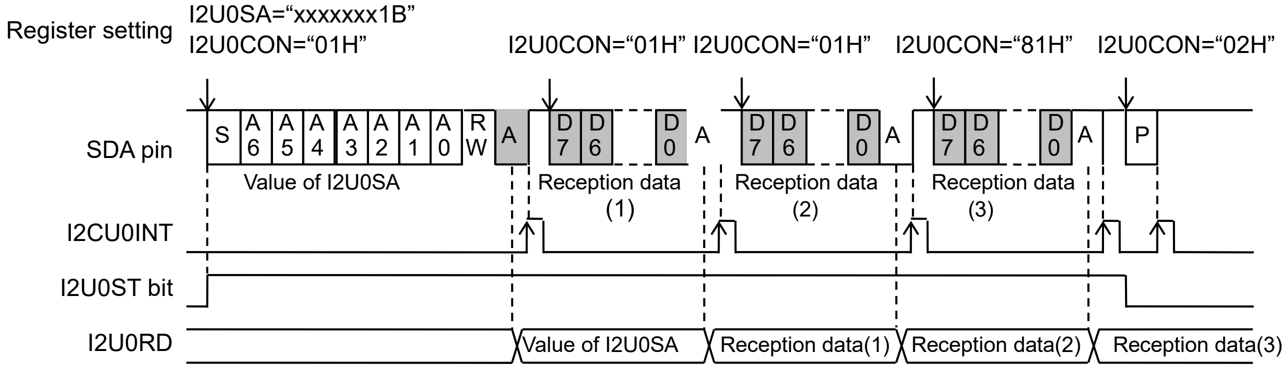


Figure 13-3 Operation timing during data reception in the master mode

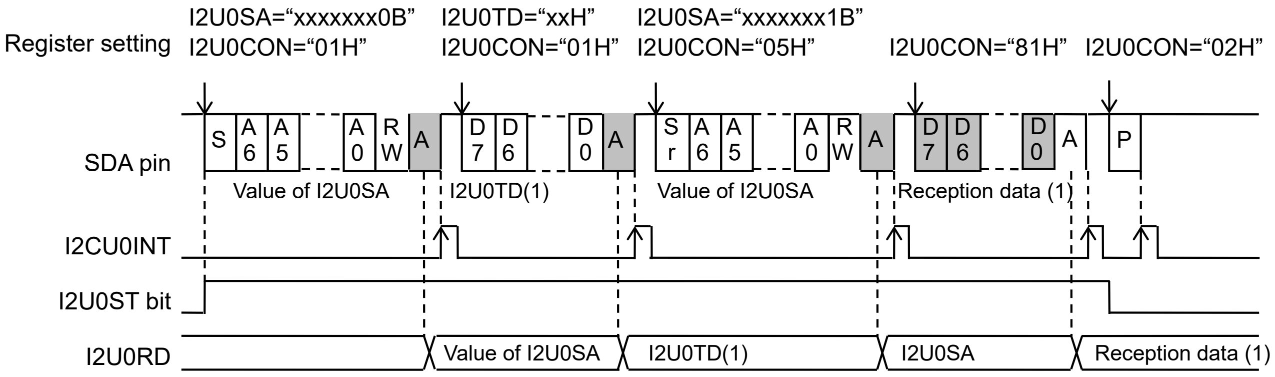


Figure 13-4 Operation timing during data transmission/ reception in the master mode

Figure 13-5 shows the operation timing and control method when an acknowledgment error occurs.

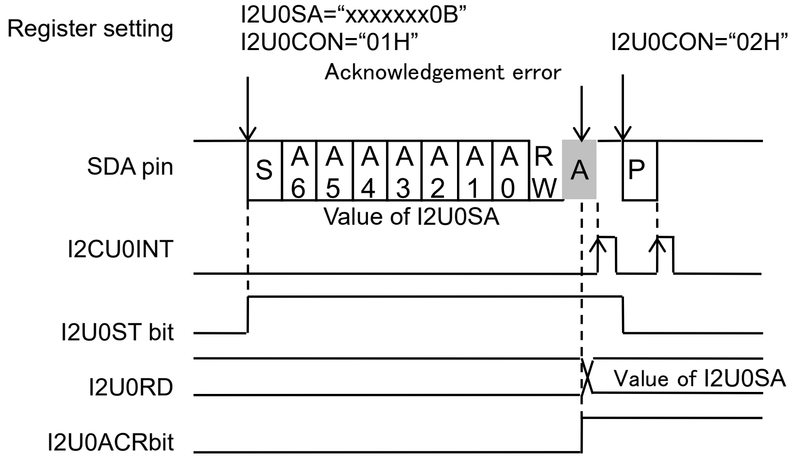


Figure 13-5 Operation suspend timing at occurrence of acknowledgment error in the master mode

When the values of the transmitted bit and the SDAU0 pin do not coincide, the I2U0ER bit of the I^{2}C bus unit 0 status register (I2U0STR) is set to "1" and the SDAU0 pin output is disabled until termination of the subsequent byte data communication.

Figure 13-6 shows the operation timing and control method when transmission fails.

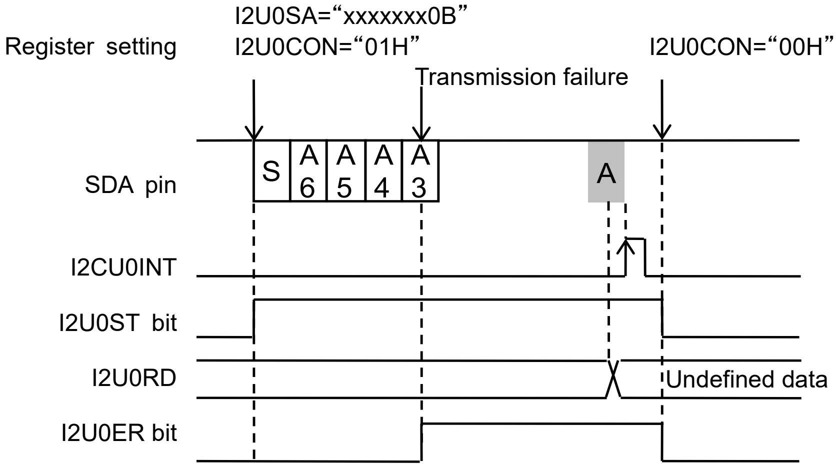


Figure 13-6 Operation timing when transmission fails in the master mode

13.5.3 Interrupt

The following is interrupt causes in master operation.

Interrupt causes	Timing that the interrupt is occurred
Transmission of a slave address	At entry to control register setting wait state after end of slave address transmission mode
Data transmission	At entry to control register setting wait state after end of data transmission mode
Data reception	At entry to control register setting wait state after end of data reception mode
Output stop condition	After output stop condition waveform and passing tBUF.

13.5.4 Operation Waveforms

Figure 13-7 shows the operation waveforms of SDAU0 and SCLU0 pins. Table 13-3 shows the relationship between communication speeds and I²C operating clock counts.

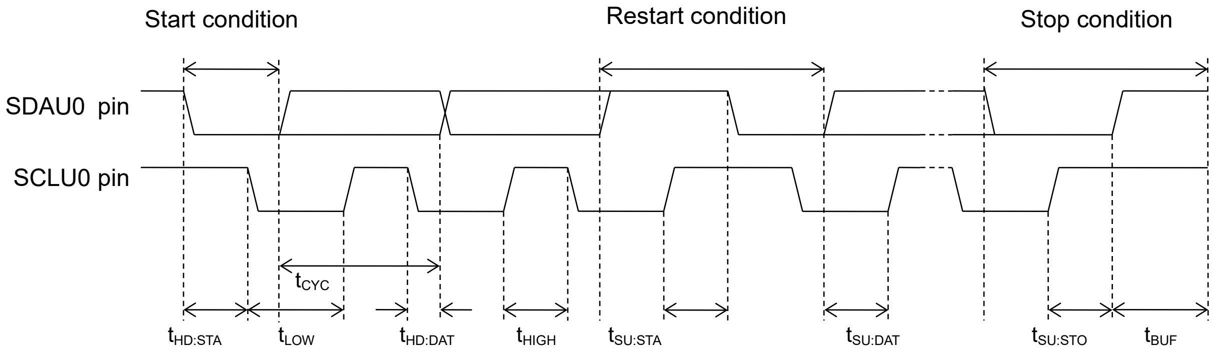


Figure 13-7 Operation Waveforms of SDAU0 and SCLU0 Pins

Table 13-3 relationship between communication speeds and I²C operating clock counts.

I2U0MOD register			AC timing [I²C operating clock counts]									I²C operating frequency and communication speed [kHz]			
I2U0TI2-0	I2U0MD1-0	I2U0MD3-2	t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}	24MHz	16MHz	1MHz	LSCLK
0	0	0	240	108	132	24	108	132	108	108	132	100.0	66.7	4.2	0.1
0	0	1	264	120	144	24	120	144	120	120	144	90.9	60.6	3.8	0.1
0	0	2	288	132	156	24	132	156	132	132	156	83.3	55.6	3.5	0.1
0	0	3	312	144	168	24	144	168	144	144	168	76.9	51.3	3.2	0.1
0	1	0	60	24	36	12	24	36	24	24	36	400.0	266.7	16.7	0.5
0	1	1	66	27	39	12	27	39	27	27	39	363.6	242.4	15.2	0.5
0	1	2	72	30	42	12	30	42	30	30	42	333.3	222.2	13.9	0.5
0	1	3	78	33	45	12	33	45	33	33	45	307.7	205.1	12.8	0.4
0	2	0	24	10	14	4	10	14	10	10	14	1000.0	666.7	41.7	1.4
0	2	1	26	11	15	4	11	15	11	11	15	923.1	615.4	38.5	1.3
0	2	2	29	13	16	4	13	16	12	13	16	827.6	551.7	34.5	1.1
0	2	3	31	14	17	4	14	17	13	14	17	774.2	516.1	32.3	1.1
1	0	0	160	72	88	16	72	88	72	72	88	150.0	100.0	6.3	0.2
1	0	1	176	80	96	16	80	96	80	80	96	136.4	90.9	5.7	0.2
1	0	2	192	88	104	16	88	104	88	88	104	125.0	83.3	5.2	0.2
1	0	3	208	96	112	16	96	112	96	96	112	115.4	76.9	4.8	0.2
1	1	0	40	14	26	12	14	26	14	14	26	600.0	400.0	25.0	0.8
1	1	1	44	16	28	12	16	28	16	16	28	545.5	363.6	22.7	0.7
1	1	2	48	18	30	12	18	30	18	18	30	500.0	333.3	20.8	0.7
1	1	3	52	20	32	12	20	32	20	20	32	461.5	307.7	19.2	0.6
1	2	0	16	6	10	4	6	10	6	6	10	1500.0*	1000.0	62.5	2.0
1	2	1	18	7	11	4	7	11	7	7	11	1333.3*	888.9	55.6	1.8
1	2	2	19	8	11	4	8	11	7	8	11	1263.2*	842.1	52.6	1.7
1	2	3	21	9	12	4	9	12	8	9	12	1142.9*	761.9	47.6	1.6
2	0	0	120	54	66	12	54	66	54	54	66	200.0	133.3	8.3	0.3
2	0	1	132	60	72	12	60	72	60	60	72	181.8	121.2	7.6	0.2
2	0	2	144	66	78	12	66	78	66	66	78	166.7	111.1	6.9	0.2
2	0	3	156	72	84	12	72	84	72	72	84	153.8	102.6	6.4	0.2
2	1	0	30	12	18	6	12	18	12	12	18	800.0	533.3	33.3	1.1
2	1	1	33	14	19	6	14	19	13	14	19	750.0	500.0	31.3	1.0
2	1	2	36	15	21	6	15	21	15	15	21	666.7	444.4	27.8	0.9

I2U0MOD register			AC timing [I^{2}C operating clock counts]									I^{2}C operating frequency and communication speed [kHz]			
I2U0TI2-0	I2U0MD1-0	I2U0MD3-2	t_{CYC}	$t_{HD:STA}$	t_{LOW}	$t_{HD:DAT}$	t_{HIGH}	$t_{SU:STA}$	$t_{SU:DAT}$	$t_{SU:STO}$	t_{BUF}	24MHz	16MHz	1MHz	LSCLK
2	1	3	39	17	22	6	17	22	16	17	22	631.6	421.1	26.3	0.9
2	2	0	12	5	7	2	5	7	5	5	7	2000.0*	1333.3*	83.3	2.7
2	2	1	13	6	7	2	6	7	5	6	7	2000.0*	1333.3*	83.3	2.7
2	2	2	14	6	8	2	6	8	6	6	8	1714.3*	1142.9*	71.4	2.3
2	2	3	15	7	8	2	7	8	6	7	8	1600.0*	1066.7*	66.7	2.2
3	0	0	80	36	44	8	36	44	36	36	44	300.0	200.0	12.5	0.4
3	0	1	88	40	48	8	40	48	40	40	48	272.2	181.8	11.4	0.4
3	0	2	96	44	52	8	44	52	44	44	52	250.0	166.7	10.4	0.3
3	0	3	104	48	56	8	48	56	48	48	56	230.8	153.8	9.6	0.3
3	1	0	20	7	13	6	7	13	7	7	13	1200.0*	800.0	50.0	1.6
3	1	1	22	8	14	6	8	14	8	8	14	1090.9*	727.3	45.5	1.5
3	1	2	24	9	15	6	9	15	9	9	15	1000.0	666.7	41.7	1.4
3	1	3	26	10	16	6	10	16	10	10	16	923.1	615.4	38.5	1.3
3	2	0	8	3	5	2	3	5	3	3	5	3000.0*	2000.0*	125.0	4.1
3	2	1	9	4	5	2	4	5	3	4	5	3000.0*	2000.0*	125.0	4.1
3	2	2	10	4	6	2	4	6	4	4	6	2666.7*	1777.8*	111.1	3.6
3	2	3	11	5	6	2	5	6	4	5	6	2400.0*	1600.0*	100.0	3.3
4	0	**	10	5	5	1	5	5	4	5	5	2400.0*	1600.0*	100.0	3.3
4	1	**	4	2	2	1	2	2	1	2	2	6000.0*	4000.0*	250.0	8.2
4	2	**	4	2	2	1	2	2	1	2	2	6000.0*	4000.0*	250.0	8.2
5	0	**	16	8	8	1	8	8	7	8	8	1500.0*	1000.0*	62.5	2.0
5	1	**	8	4	4	1	4	4	3	4	4	3000.0*	2000.0*	125.0	4.1
5	2	**	4	2	2	1	2	2	1	2	2	6000.0*	4000.0*	250.0	8.2

*: The operation is not guaranteed when over 1MHz speed.
**: The setting is invalid.

These clock counts is in case of I2U0CD1-0/I2M0CD1-0 bits = “00”. If it is not “00”, its counts increase depending on dividing rate.

[Note]

- **When the slave device uses the clock stretch function which holds the SCLU0 pin at "L" level, the time t_{CYC} and time t_{LOW} are extended.**

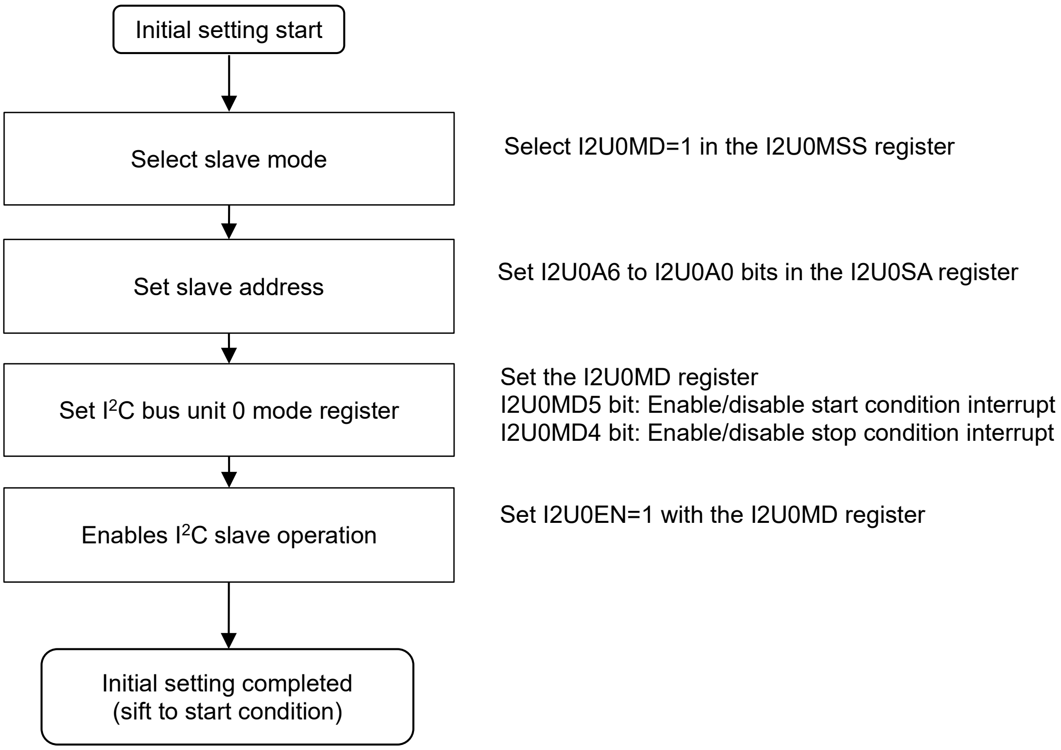
13.6 Description of Operation for Slave function

I^{2}C bus unit 0 only have slave function.

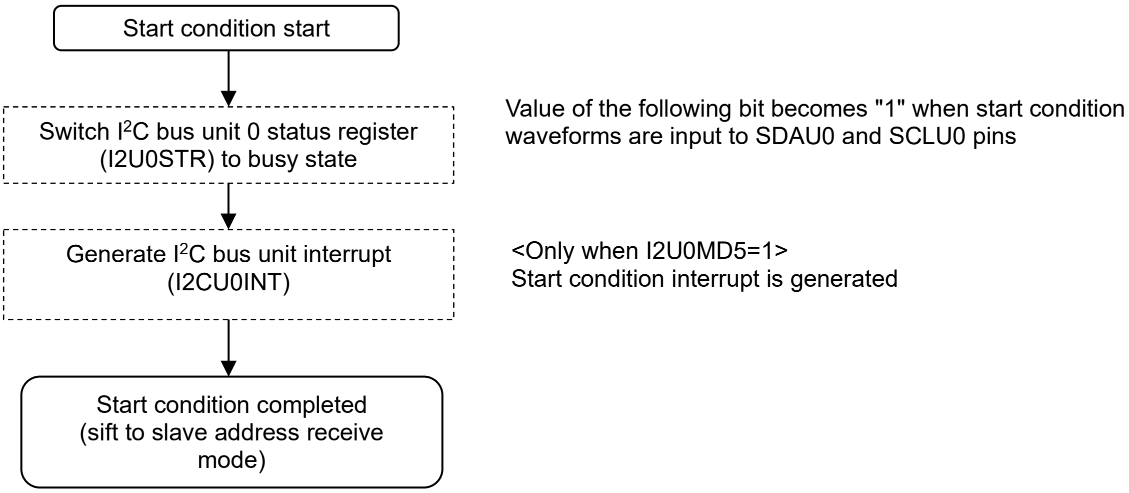
13.6.1 Procedures

The following flow charts describe procedures of each operation in the slave mode.

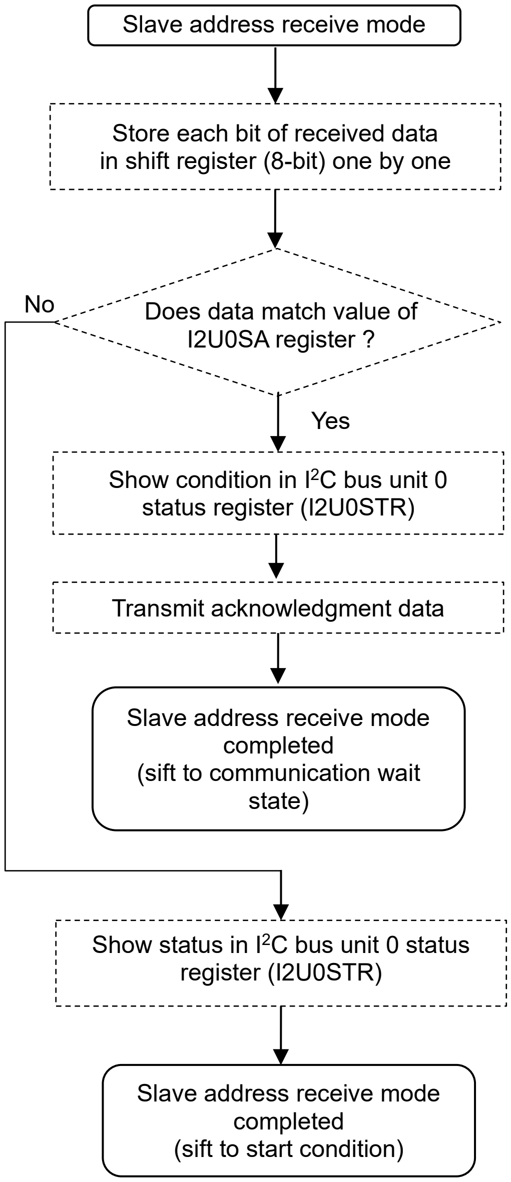
13.6.1.1 Initial Setting of Communication Operation


Initial setting start
Select slave mode
Select I2U0MD=1 in the I2U0MSS register
Set slave address
Set I2U0A6 to I2U0A0 bits in the I2U0SA register
Set I^{2}C bus unit 0 mode register
Set the I2U0MD register
I2U0MD5 bit: Enable/disable start condition interrupt
I2U0MD4 bit: Enable/disable stop condition interrupt
Enables I^{2}C slave operation
Set I2U0EN=1 with the I2U0MD register
Initial setting completed
(sift to start condition)

13.6.1.2 Start Condition


Start condition start
Switch I^{2}C bus unit 0 status register
(I2U0STR) to busy state
Value of the following bit becomes "1" when start condition
waveforms are input to SDAU0 and SCLU0 pins
Generate I^{2}C bus unit interrupt
(I2CU0INT)
<Only when I2U0MD5=1>
Start condition interrupt is generated
Start condition completed
(sift to slave address receive
mode)

13.6.1.3 Slave Address Reception Mode


Slave address receive mode
Store each bit of received data in shift register (8-bit) one by one
Does data match value of I2U0SA register ?
No
Yes
Show condition in I²C bus unit 0 status register (I2U0STR)
Transmit acknowledgment data
Slave address receive mode completed (sift to communication wait state)
Show status in I²C bus unit 0 status register (I2U0STR)
Slave address receive mode completed (sift to start condition)

Value (slave address, data direction bit) input to SDAU0 pin is stored in synchronization with rising edge of transfer clock input to SCLU0 pin in MSB first

Compare slave address stored in shift register with slave address of I2U0SA

I2U0SAA bit: Data matches slave address (I2U0SAA=1)
I2U0TR bit: Transmit state/receive state

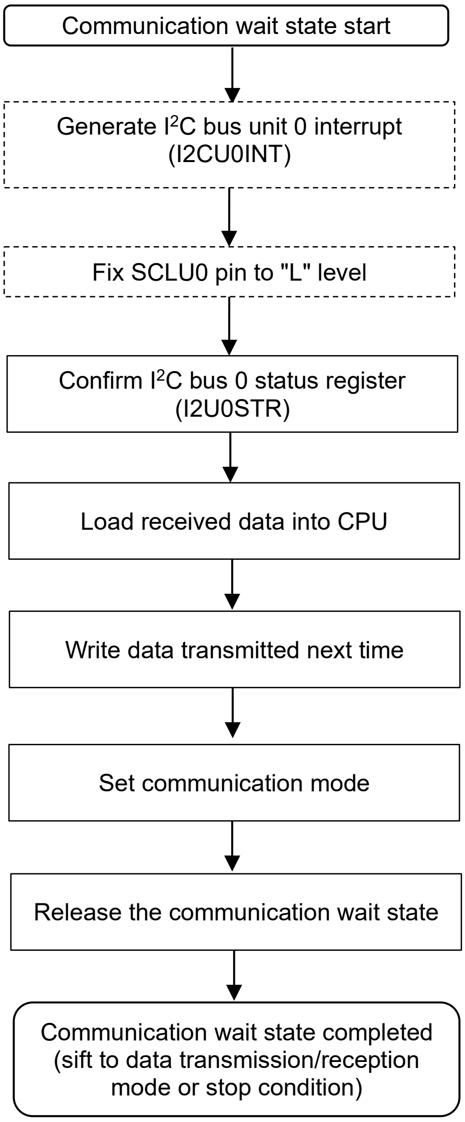
Acknowledgment data ("L" level) is output

Shift to the communication wait state after detection of falling edge of transfer clock which was input to SCLU0 pin transmitting acknowledgment data

I2U0SAA bit: Data does not match slave address (I2U0SAA=0)

If not match, neither acknowledgment nor interrupt is generated

13.6.1.4 Communication Wait State


Communication wait state start
Generate I²C bus unit 0 interrupt (I2CU0INT)
Fix SCLU0 pin to "L" level
Confirm I²C bus 0 status register (I2U0STR)
Load received data into CPU
Write data transmitted next time
Set communication mode
Release the communication wait state
Communication wait state completed (sift to data transmission/reception mode or stop condition)

When entering communication wait state,
an interrupt is generated through hardware

Confirm the I2U0STR register
I2U0ER bit: transmit error flag
I2U0ACR bit: Acknowledgment data

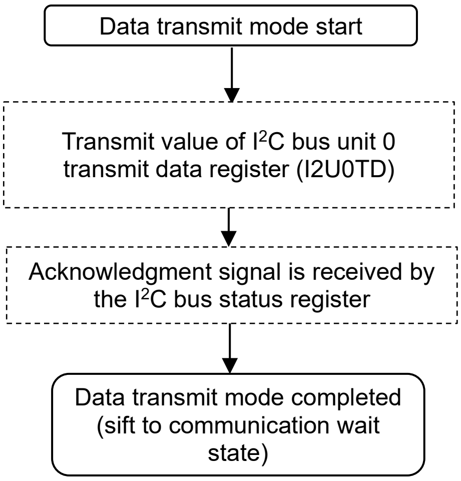
<Only when data is received> Read I2U0RD
I2U0R7 to I2U0R0 bits: 8-bit receive data

<Only when data is transmitted> Set the I2U0TD register
I2U0T7 to I2U0T0 bits: 8-bit transmit data

<Only when interrupt source is changed>
Set the I2U0MD register

Set I2U0WT=1 with I2U0CON
Switch to the high-speed clock after setting the I2U0WT bit.

13.6.1.5 Data Transmission Mode

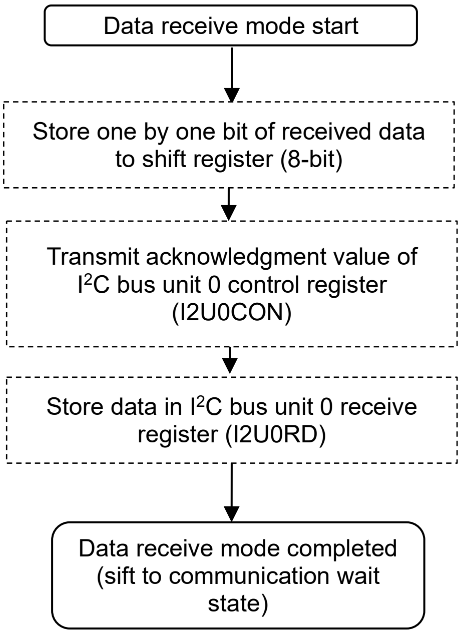

Data transmit mode start
Transmit value of I²C bus unit 0 transmit data register (I2U0TD)
Acknowledgment signal is received by the I²C bus status register
Data transmit mode completed (sift to communication wait state)

The transmit data that has been written to the I2U0TD register is transmitted from SDAU0 pin in MSB first
I2U0T7 to I2U0T0 bits: 8-bit transmit data
Value transmitted from the SDAU0 pin is stored in the I2U0RD register

Acknowledgment signal is received through hardware
I2U0ACR bit: Acknowledgment data

Shift to communication wait state after detection of falling edge of transfer clock which was input to SCLU0 pin receiving acknowledgment data

13.6.1.6 Data Reception Mode


Data receive mode start
Store one by one bit of received data to shift register (8-bit)
Transmit acknowledgment value of I^2C bus unit 0 control register (I2U0CON)
Store data in I^2C bus unit 0 receive register (I2U0RD)
Data receive mode completed (sift to communication wait state)

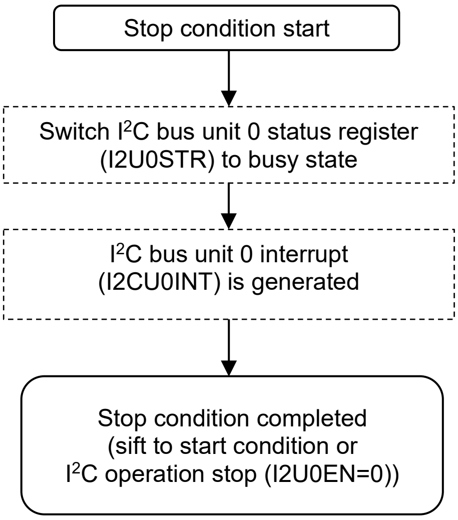
Value (received data) input to SDAU0 pin is stored in synchronization with rising edge of transfer clock input to SCLU0 pin in MSB first

Acknowledgment signal is transmitted through hardware
I2U0ACT bit: Acknowledgment value
Transmitted acknowledgment value is stored in the I2U0ACR bit of the I2U0STR register

Received data is stored from the shift register after acknowledgment signal is transmitted
I2U0R7 to I2U0R0 bits: 8-bit receive data

Shift to communication wait state after detection of falling edge of transfer clock which was input to SCLU0 pin transmitting acknowledgment data

13.6.1.7 Stop Condition


Stop condition start
Switch I^2C bus unit 0 status register (I2U0STR) to busy state
I^2C bus unit 0 interrupt (I2CU0INT) is generated
Stop condition completed (sift to start condition or I^2C operation stop (I2U0EN=0))

Value of the following bit becomes "0" when stop condition waveforms are input to SDAU0 and SCLU0 pins.

<Only when I2U0MD4=1>
Stop condition interrupt is generated

13.6.2 Communication Operation Timing

Figures 13-8 to 13-10 show the operation timing and control method for each communication mode.

□	Transmission	S	Start condition	Sr	Restart condition	A	Transmission of acknowledgment data 0
▨	Reception	P	Stop condition	A	Reception of acknowledgment	A	Transmission of acknowledgment data 1

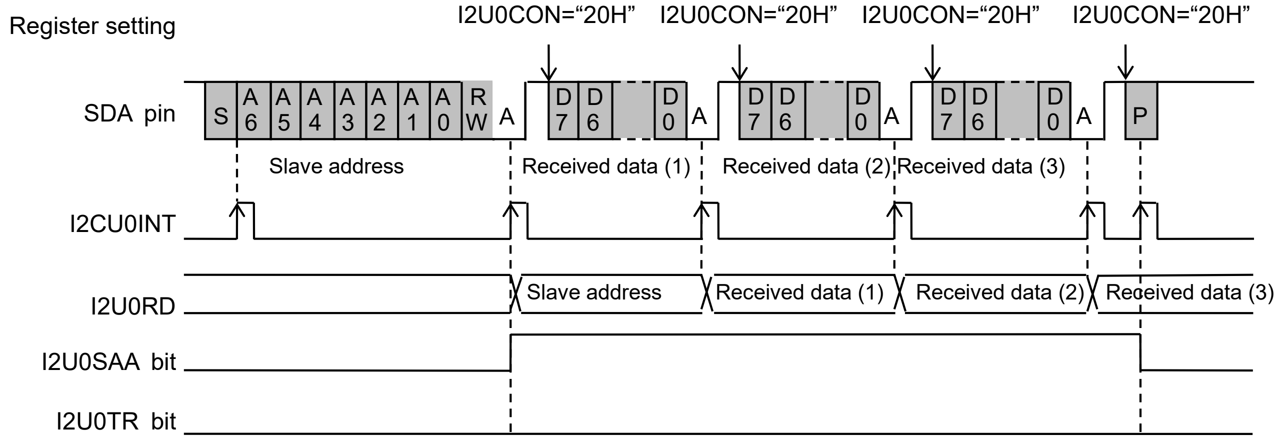


Figure 13-8 Operation Timing in Data Reception Mode When Slave Mode is Chosen

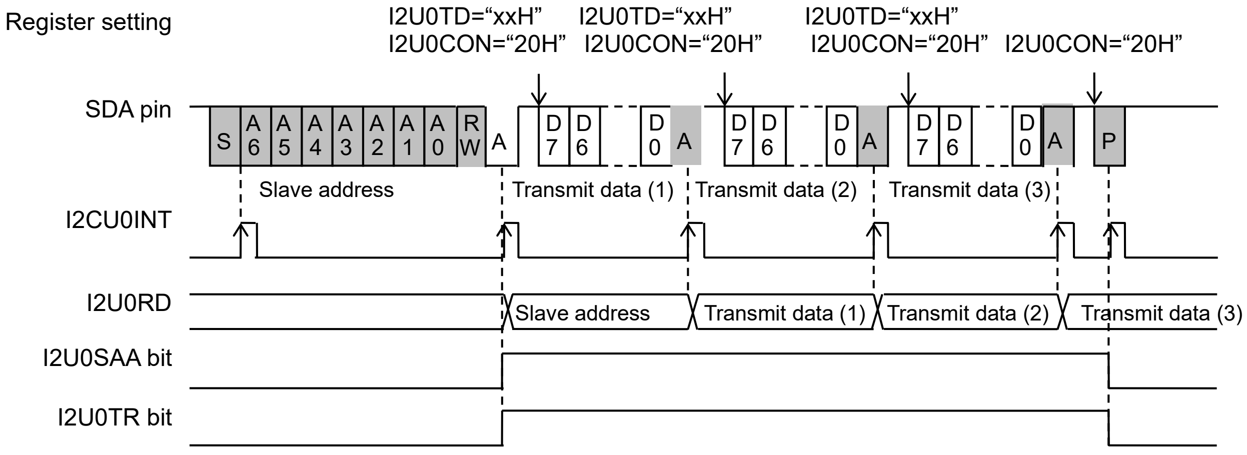


Figure 13-9 Operation Timing in Data Transmission Mode When Slave Mode is Chosen

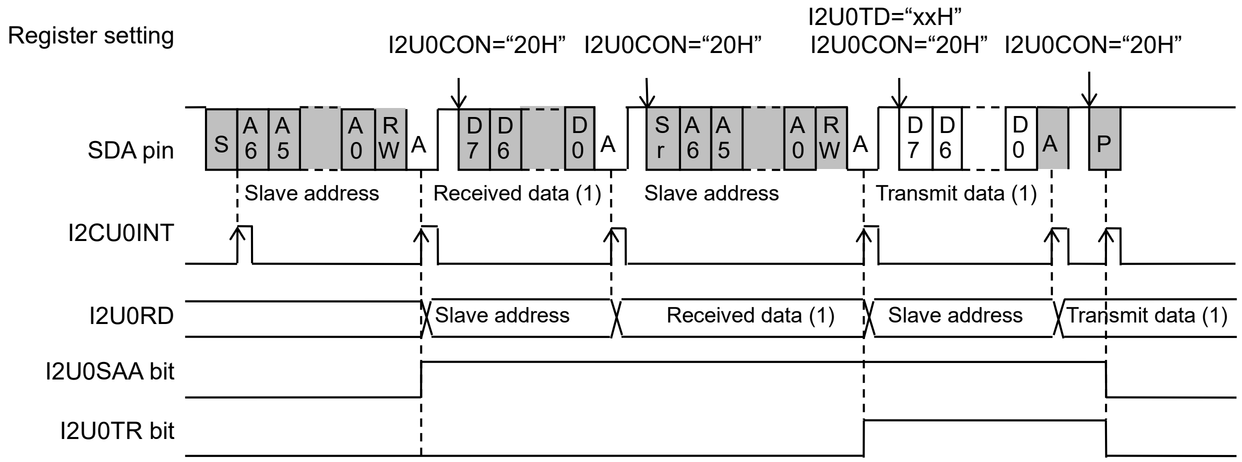


Figure 13-10 Operation Timing at Data Transmission/Reception Mode Switching When Slave Mode is Chosen

When the values of the transmitted bit and the SDAU0 pin do not coincide, the I2U0ER bit of the I^{2}C bus unit 0 status register (I2U0STR) is set to "1" and the SDAU0 pin output is disabled until termination of the subsequent byte data communication.
Figure 13-11 shows the operation timing and control method when transmission fails.

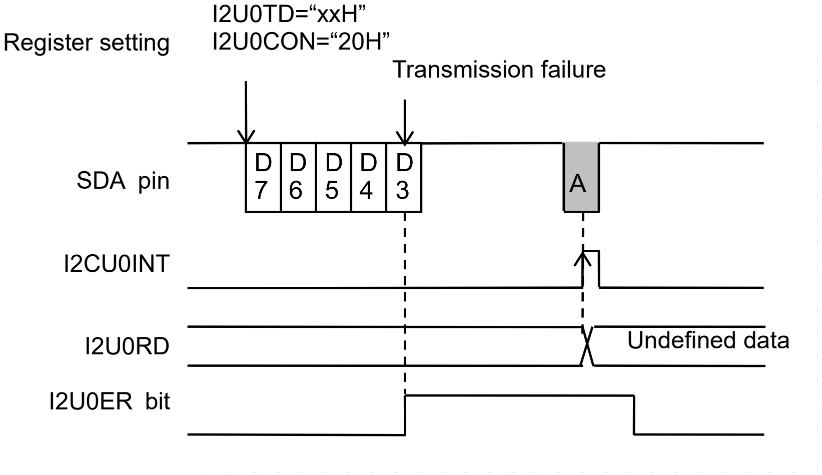


Figure 13-11 Operation Timing When Transmission Fails When Slave Mode is Chosen

[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).**

13.6.3 Interrupt

Table 13-4 shows interrupt causes in slave operation.

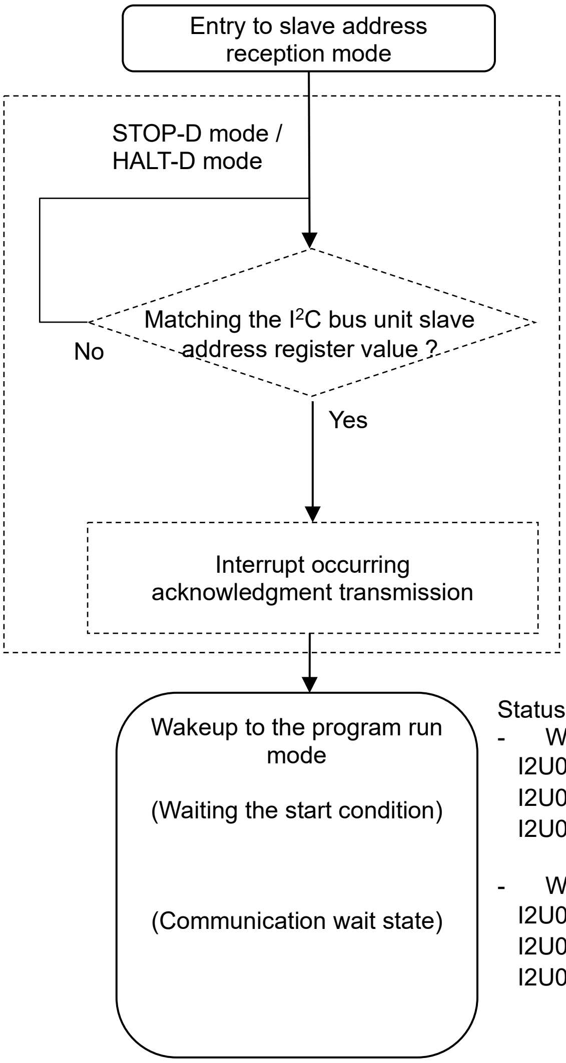
Table 13-4 List of slave interrupt

Interrupt causes	Setting to enable	Status flag in the I2U0STR register	Timing that the interrupt is occurred
Start condition	I2U0MD5=1	I2U0STS=1	After output start condition waveform.
Coinciding Slave Address	-	I2U0AS=1	At entry to control register setting wait state with coinciding slave-address after end of slave address reception mode
Coinciding Slave Address (in the HALT-D／STOP-D mode)	-	I2U0ASNA=1	At output "H" as acknowledge with coinciding slave-address after end of slave address reception mode.
Data transmission	-	I2U0DS=1	At entry to control register setting wait state after end of data transmission mode
Data reception	-	I2U0DS=1	At entry to control register setting wait state after end of data reception mode
Stop condition	I2U0MD4=1	I2U0SPS=1	At detecting stop-condition waveform.
Re-start condition, and then the master selects another slave.	I2U0MD3=1	I2U0RAS=1	At end of slave-address reception mode without coinciding slave-address after detected re-start condition with I2USAA=1.

Detect stop-condition for another slave.	I2U0MD4=1,
	I2U0MD1=0
		I2U0SPS=1, I2U0AS=1	At detecting stop-condition waveform with I2U0SAA=1.

13.6.4 Wake-up from STOP-D/HALT-D mode by the Slave Address Coinciding

The LSI can return from stand-by mode to program run mode by matching slave address.
In the HALT/HALT-H/STOP mode, the acknowledge data “L” level is output. In the STOP-D/HALT-D, the acknowledge data “H” level is output.
When acknowledge data “H” level is output, the communication can be performed by retransmitting from the start condition.
Depending on the timing of an entry to the STOP-D/HALT-D mode and slave address reception, it may wakeup with acknowledge data “L” level output. The I2U0STR register indicates these state. See section “13.6.3 Interrupt” for detail.


Entry to slave address reception mode
STOP-D mode / HALT-D mode
Matching the I^{2}C bus unit slave address register value ?
No
Yes
Interrupt occurring acknowledgment transmission
Wakeup to the program run mode
(Waiting the start condition)
(Communication wait state)

Enter to STOP-D/HALT-D mode after initial setup.

It compares the slave address stored in the shift register with the I2U0SA data.

If the results do not match, The stand by mode is kept. and the interrupt does not occur..

The interrupt occurs and the LSI wakeups from STOP-D/HALT-D mode.
“H” level data is sent as the acknowledgement.
“L” level data may be sent if the entry to standby and slave address reception timing are close.

Status in the I2U0STR register after wakeup by slave address matching.

- When Acknowledgement “H” level is sent :
  I2U0ASNA＝1
  I2U0SAA＝0
  I2U0AS＝0

- When Acknowledgement “H” level is sent :
  I2U0ASNA＝0
  I2U0SAA＝1
  I2U0AS＝1

Figure 13-12 Wakeup from STOP-D/HALT-D mode by a coinciding slave address

[Note]

- **The master device should Wait for the SYSCLK to be supplied in order to transmit the start condition after wakeup from the STOP-D/HALT-D mode by slave address matching.**
- **It is supported the Standard/Fast mode (to max. 400 kbps) in the STOP-D／HALT-D mode.**

Chapter 14 UART

14. UART

14.1 General Description

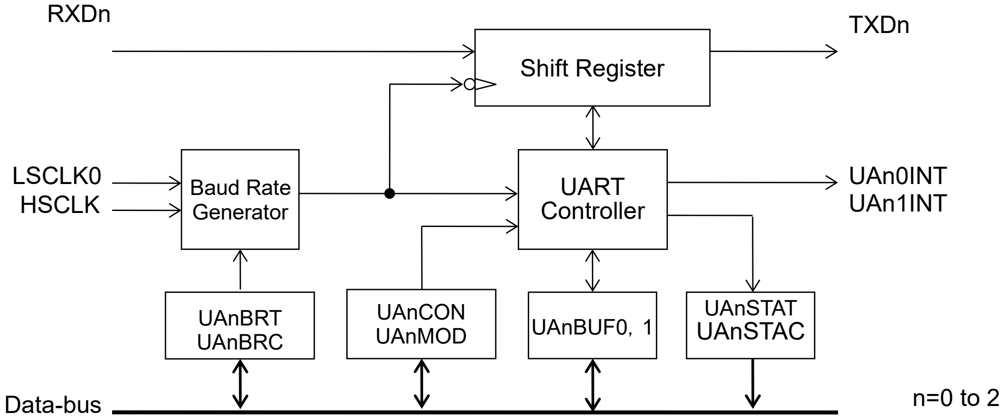
ML62Q2500 group has full-duplexed universal asynchronous receiver transmitter; UART.

14.1.1 Features

- Data length : 5/6/7/8 bit
- Data parity : odd, even, fixed 0, fixed 1, none
- Stop bit : 1bit or 2bit
- Status flags: parity error, overrun error, framing error, transmission buffer
- Signal level : positive, negative
- Data direction : LSB first or MSB first
- Wide range of communication speed
  - 1bps~4,800bps (Clock frequency is 32.768kHz)
  - 300bps~2Mbps (Clock frequency is 16MHz)
  - 600bps~3Mbps (Clock frequency is 24MHz)
- Built-in baud rate generator for each channel
- Self-test function using transmission and reception. See Chapter 29 "Safety Function." for the self-test functions.

14.1.2 Configuration

Figure 14-1 shows configuration of the UART.



UAnBUF0,1	:	UARTn transmission/reception buffer
UAnBRT	:	UARTn baud rate register
UAnBRC	:	UARTn baud rate adjustment register
UAnCON	:	UARTn control register
UAnMOD	:	UARTn mode register
UAnSTAT, UAnSTAC	:	UARTn status register, UARTn status clear register

Figure 14-1 Configuration of UART

14.1.3 List of Pins

The I/O pins of the UART are assigned to the shared function of the general ports.

Pin name	I/O	Description
RXDn	I	Reception data input of UART n
TXDn	O	Transmission data output of UART n

Table 14-1 shows the list of the general ports used for the UART and the register settings of the ports.

Table 14-1 Ports used for the UART and the register settings

Channel no.	Pin name	Shared port		Setting register	Setting value	ML62Q2500 group		
						32 pin product	40 pin product	48 pin product
0	RXD0	P04	3rd Func.	P0MOD4	0010_XXXX*1	●	●	●
		P20	3rd Func.	P2MOD0	0010_XXXX*1	●	●	●
		P30	3rd Func.	P3MOD0	0010_XXXX*1	●	●	●
		P52	3rd Func.	P5MOD2	0010_XXXX*1	-	-	●
	TXD0	P05	3rd Func.	P0MOD5	0010_XXXX*2	●	●	●
		P21	3rd Func.	P2MOD1	0010_XXXX*2	●	●	●
		P31	3rd Func.	P3MOD1	0010_XXXX*2	●	●	●
		P53	3rd Func.	P5MOD3	0010_XXXX*2	-	-	●
1	RXD1	P02	3rd Func.	P0MOD2	0010_XXXX*1	-	-	●
		P06	3rd Func.	P0MOD6	0010_XXXX*1	●	●	●
		P32	3rd Func.	P3MOD2	0010_XXXX*1	●	●	●
	TXD1	P03	3rd Func.	P0MOD3	0010_XXXX*2	-	-	●
		P07	3rd Func.	P0MOD7	0010_XXXX*2	●	●	●
		P33	3rd Func.	P3MOD3	0010_XXXX*2	●	●	●
2	RXD2	P10	3rd Func.	P1MOD0	0010_XXXX*1	●	●	●
		P72	3rd Func.	P7MOD2	0010_XXXX*1	●	●	●
	TXD2	P11	3rd Func.	P1MOD1	0010_XXXX*2	●	●	●
		P73	3rd Func.	P7MOD3	0010_XXXX*2	●	●	●

●: Available to use, -: 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	N-ch open drain output (without the pull-up)
1111	N-ch open drain output (with the pull-up)

14.2 Description of Registers

14.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF600	UART0 reception buffer	UA0BUF0	-	R	8	0x00
0xF601	UART0 transmission buffer	UA0BUF1	-	R/W	8	0x00
0xF602	UART0 status register	UA0STAT	-	R	8	0x00
0xF603	UART0 status clear register	UA0STAC	-	W	8	0x00
0xF604	UART0 control register	UA0CON	-	R/W	8	0x00
0xF605	Reserved	-	-	-	-	-
0xF606	UART0 mode register	UA0MODL	UA0MOD	R/W	8/16	0x00
0xF607		UA0MODH		R/W	8	0x00
0xF608	UART0 interrupt enable register	UA0INTE	-	R/W	8	0x00
0xF609	Reserved	-	-	-	-	-
0xF60A	UART0 baud rate register	UA0BRTL	UA0BRT	R/W	8/16	0xFF
0xF60B		UA0BRTH		R/W	8	0x0F
0xF60C	UART0 baud rate adjustment register	UA0BRC	-	R/W	8	0x00
0xF60D~ 0xF60F	Reserved	-	-	-	-	-
0xF610	UART1 reception buffer	UA1BUF0	-	R	8	0x00
0xF611	UART1 transmission buffer	UA1BUF1	-	R/W	8	0x00
0xF612	UART1 status register	UA1STAT	-	R	8	0x00
0xF613	UART1 status clear register	UA1STAC	-	W	8	0x00
0xF614	UART1 control register	UA1CON	-	R/W	8	0x00
0xF615	Reserved	-	-	-	-	-
0xF616	UART1 mode register	UA1MODL	UA1MOD	R/W	8/16	0x00
0xF617		UA1MODH		R/W	8	0x00
0xF618	UART1 interrupt enable register	UA1INTE	-	R/W	8	0x00
0xF619	Reserved	-	-	-	-	-
0xF61A	UART1 baud rate register	UA1BRTL	UA1BRT	R/W	8/16	0xFF
0xF61B		UA1BRTH		R/W	8	0x0F
0xF61C	UART1 baud rate adjustment register	UA1BRC	-	R/W	8	0x00
0xF61D~ 0xF61F	Reserved	-	-	-	-	-
0xF620	UART2 reception buffer	UA2BUF0	-	R	8	0x00
0xF621	UART2 transmission buffer	UA2BUF1	-	R/W	8	0x00
0xF622	UART2 status register	UA2STAT	-	R	8	0x00
0xF623	UART2 status clear register	UA2STAC	-	W	8	0x00
0xF624	UART2 control register	UA2CON	-	R/W	8	0x00
0xF625	Reserved	-	-	-	-	-
0xF626	UART2 mode register	UA2MODL	UA2MOD	R/W	8/16	0x00
0xF627		UA2MODH		R/W	8	0x00
0xF628	UART2 interrupt enable register	UA2INTE	-	R/W	8	0x00
0xF629	Reserved	-	-	-	-	-
0xF62A	UART2 baud rate register	UA2BRTL	UA2BRT	R/W	8/16	0xFF
0xF62B		UA2BRTH		R/W	8	0x0F
0xF62C	UART2 baud rate adjustment register	UA2BRC	-	R/W	8	0x00

[Note]

- **When the DCKUAn of Block control register 2; BCKCON2, A reading value of relevant SFR is 0x00/0x0000. However values written to the SFR is kept. A reading value of relevant SFR is setting value after the DCKUAn bit is returned “0”. See Chapter 4 “Power Management” as for block control register.**

14.2.2 UARTn Reception Buffer (UAnBUF0)

This is a SFR to store reception data.

Address :	0xF600(UA0BUF0),0xF610(UA1BUF0),0xF620(UA2BUF0),
Access :	R
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAnBUF0							
Bit	-	-	-	-	-	-	-	-	UnRD 7	UnRD 6	UnRD 5	UnRD 4	UnRD 3	UnRD 2	UnRD 1	UnRD 0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit no.	Bit symbol name	Description
7 to 0	UnRD7 to UnRD0	This works as the reception buffer. Data at the end of reception communication is overwritten into this register, so read out this register by using the UARTn0 interrupt generated at the end of reception communication. Writing to this register is invalid. When choosing the 5 to 7 bit length, unused bits return “0” for reading.

14.2.3 UARTn Transmission Buffer (UAnBUF1)

This is a SFR to store transmission data.

Address :	0xF601(UA0BUF1),0xF611(UA1BUF1),0xF621(UA2BUF1),
Access :	R/W
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAnBUF1							
Bit	-	-	-	-	-	-	-	-	UnTD 7	UnTD 6	UnTD 5	UnTD 4	UnTD 3	UnTD 2	UnTD 1	UnTD 0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit no.	Bit symbol name	Description
7 to 0	UnTD7 to UnTD0	This works as the transmission buffer. Write transmission data to this. For continuous transmitting, write the next transmission data to this register after checking UnFUL bit of UAnSTAT is “0”. The written data in this register can be read out. When choosing the 5 to 7 bit length, written data in unused bits are invalid.

14.2.4 UARTn Status Register (UAnSTAT)

This is a SFR to indicate states of the transmission/reception operation.

Address :	0xF602(UA0STAT),0xF612(UA1STAT),0xF622(UA2STAT)
Access :	R
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAnSTAT							
Bit	-	-	-	-	-	-	-	-	-	-	UnRXF	UnTXF	UnFUL	UnPER	UnOER	UnFER
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit no.	Bit symbol name	Description
7 to 6	-	Reserved bits
5	UnRXF	This is used to indicate reception state. 0: Data reception is stopped (Initial value) 1: Data reception is in progress
4	UnTXF	This is used to indicate transmission state. 0: Data transmission is stopped (Initial value) 1: Data transmission is in progress
3	UnFUL	This is used to indicate transmission buffer state. This bit is set to “1” by writing a data to UAnBUF1 and reset to “0” when starting to transfer the data. To transmit data successively, check that the UnFUL bit is "0" before writing the next transmission data to the UAnBUF1.This bit is forcibly reset to "0" by writing “1” to the UnFULC bit of UAnSTAC register. 0: Transmission buffer has no data (Initial value) 1: Transmission buffer has data
2	UnPER	This is used to indicate a parity error. The parity of the received data and the parity bit added to the data are compared and if they do not match, this bit becomes "1".This bit is forcibly reset to "0" by writing “1” to the UnPERC bit of UAnSTAC register. 0: The parity error has not occurred. (Initial value) 1: The parity error has occurred.
1	UnOER	This is used to indicate a reception overrun error. This bit becomes "1" if the next data is received before reading the previous receive data in reception buffer (SDnBUFL). Even if reception is stopped by the UnEN bit and then reception is re-started, this bit is set to “1” unless the previously received data is not read. Therefore, make sure that data is always read from the reception buffer even if the data is not required. This is forcibly reset to "0" by writing “1” to the UnOERC bit of UAnSTAC register. 0: The overrun error has not occurred (Initial value) 1: The overrun error has occurred
0	UnFER	This is used to indicate a framing error. This becomes "1" when an error occurs in the start/stop bit. The Un0FER bit is forcibly reset to "0" by writing “1” to the UnFERC bit of UAnSTARC register. 0: The framing error has not occurred (Initial value) 1: The framing error has occurred

14.2.5 UARTn Status Clear Register (UAnSTAC)

This is a write-only SFR to clear states of the transmission/reception operation.

Address :	0xF603(UA0STAC),0xF613(UA1STAC),0xF623(UA2STAC)
Access :	W
Access size :	8 bit
Initial value :	0x00

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

Bit no.	Bit symbol name	Description
7 to 3	-	Reserved bits
3	UnFULC	This is used to clear UnFUL bit ; transmission buffer state flag. Writing “0” : Invalid Writing “1” : Clear the UnFUL bit
2	UnPERC	This is used to clear UnPER bit ; parity error flag. Writing “0” : Invalid Writing “1” : Clear the UnPER bit.
1	UnOERC	This is used to clear UnOER bit ; overrun error flag. Writing “0” : Invalid Writing “1” : Clear the UnOER bit.
0	UnFERC	This is used to clear UnFER bit ; framing error flag. Writing “0” : Invalid Writing “1” : Clear the UnFER bit.

14.2.6 UARTn Control Register (UAnCON)

This is a SFR to control enable/disable communication.

Address :	0xF604(UA0CON),0xF614(UA1CON),0xF624(UA2CON)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

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

Bit no.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	UnEN	This is used to enable the UART n communication. See section “14.3.4 Transmission Operation” for details. A both of transmission and reception is enabled when UnEN = 1. Set “0” to this bit if communication is stopped. 0: Disabled (Initial value) 1: Enabled

[Note]

- **Do setting for used ports and the mode/baud rate before setting “1” to UnEN bit.**

14.2.7 UARTn Mode Register (UAnMOD)

This is a SFR to set the transfer mode.

Address :	0xF606(UA0MODL/ UA0MOD),0xF607(UA0MODH), 0xF616(UA1MODL/UA1MOD),0xF617(UA1MODH), 0xF626(UA2MODL/UA2MOD),0xF627(UA2MODH)
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	UAnMOD															
Byte	UAnMODH								UAnMODL							
Bit	UnDIR	UnNEG	UnSTP	UnPT2	UnPT1	UnPT0	UnLG1	UnLG0	-	UnRSS	-	-	-	-	UnCK0	-
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/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
15	UnDIR	This is used to choose the data direction. 0: LSB first (Initial value) 1: MSB first
14	UnNEG	This is used to choose a logic of the data input / output. 0: Positive logic (Initial value) 1: Negative logic
13	UnSTP	This is used to choose a stop bit length. 0: 1 stop bit (Initial value) 1: 2 stop bit
12 to 10	UnPT2 to UnPT0	This is used to choose a parity bit. 00 0: No parity bit (Initial value) 001: Odd parity 010: No parity bit 011: Even parity 10 0: No parity bit 101: Parity bit is fixed to “1” 110: No parity bit 111: Parity bit is fixed to “0”
9 to 8	UnLG1 to UnLG0	These are used to choose a data length. 00: 8-bit length (Initial value) 01: 7-bit length 10: 6-bit length 11: 5-bit length
7	-	Reserved bits
6	UnRSS	This is used to choose sampling timing of the reception data. 0: (Values set to UAn0BRTH and UAn0BRTL registers)/2 (Initial value) 1: {(Values set to UAn0BRTH and UAn0BRTL registers)/2} -1
5~2	-	Reserved bits
1	UnCK0	This is used to choose base clock baud rate generator. 0: LSCLK0 (Initial value) 1: HSCLK
0	-	Reserved bits

[Note]

- **Be sure to set the UAn0MOD register while communication is stopped (Un0EN=0).**

14.2.8 UARTn Interrupt Enable Register (UAnINTE)

This is a SFR to enable interrupt requests.

Address :	0xF608 (UA0INTE),0xF618 (UA1INTE),0xF628 (UA2INTE)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

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

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

Bit no.	Bit symbol name	Description
7 to 3	-	Reserved bits
2	UnFIE	Transmission completion interrupt This has occurred when transmission data is transmitted in the condition of transmission buffer empty.
1	UnTIE	Transmission buffer empty interrupt. This has occurred when transmission buffer becomes empty.
0	UnRIE	Reception interrupt. This has occurred at receiving a data.

14.2.9 UARTn Baud Rate Register (UAnBRT)

This is a SFR to set the count value of the baud rate generator in UARTn.
For details of relation between the count value of the baud rate generator and the baud rate, see Section 14.3.3 “Baud Rate”.

Address : 0xF60A (UA0BRTL/UA0BRT), 0xF60B (UA0BRTH),
0xF61A (UA1BRTL/UA1BRT), 0xF61B (UA1BRTH),
0xF62A (UA2BRTL/UA2BRT), 0xF62B (UA2BRTH)
Access : R/W
Access size : 8/16 bit
Initial value : 0x0FFF

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

14.2.10 UARTn Baud Rate Adjustment Register (UAnBRC)

This is a SFR to adjust the count value of the baud rate generator in UARTn.
For details of relation between the value of UAnBRC and the correction value, see Section 14.3.3 “Baud Rate”.

Address : 0xF60C(UA0BRC),0xF61C(UA1BRC),0xF62C(UA2BRC)
Access : R/W
Access size : 8 bit
Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								UAnBRC							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	UnBR C2	UnBR C1	UnBR C0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[Note]

- **Be sure to set the UAnBRT and UAnBRC register while communication is stopped (UnEN=0). Do not rewrite it during communication.**

14.3 Description of Operation

14.3.1 Frame Format

In the transfer data format, one frame contains a start bit, a data bit, a parity bit, and a stop bit. In this format, the following are choosable: 5 to 8 bits for the data bit, even/odd/ fixed to "1", or fixed to "0" for the parity bit, 1 stop bit or 2 stop bit for the stop bit, LSB first or MSB first for the transfer direction, and positive logic or negative logic for the logic of the serial input/output.
All of these are set in the UARTn mode register (UAnMOD).
Figure 14-2 shows the input/output format.

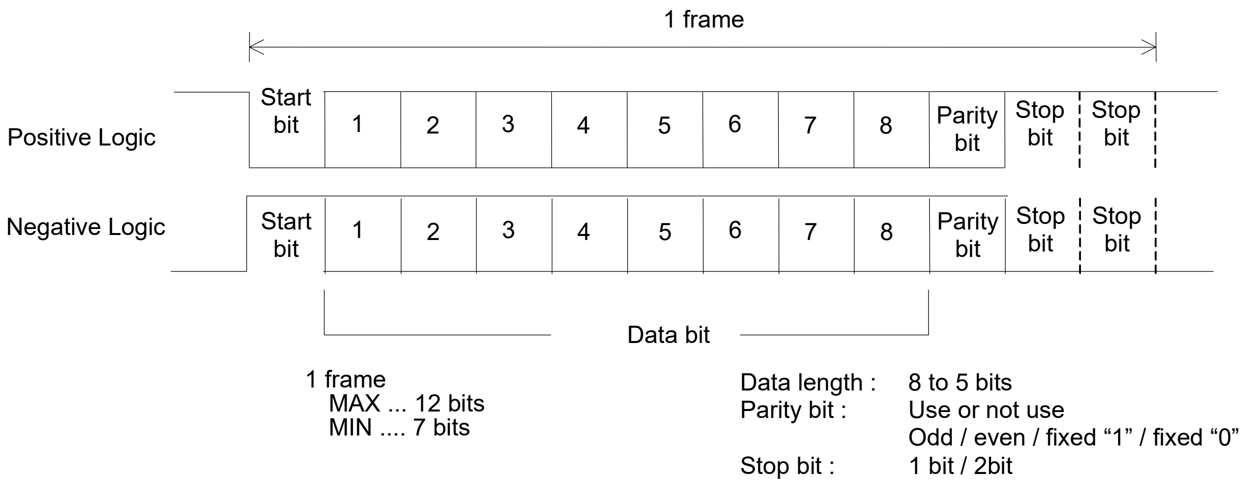
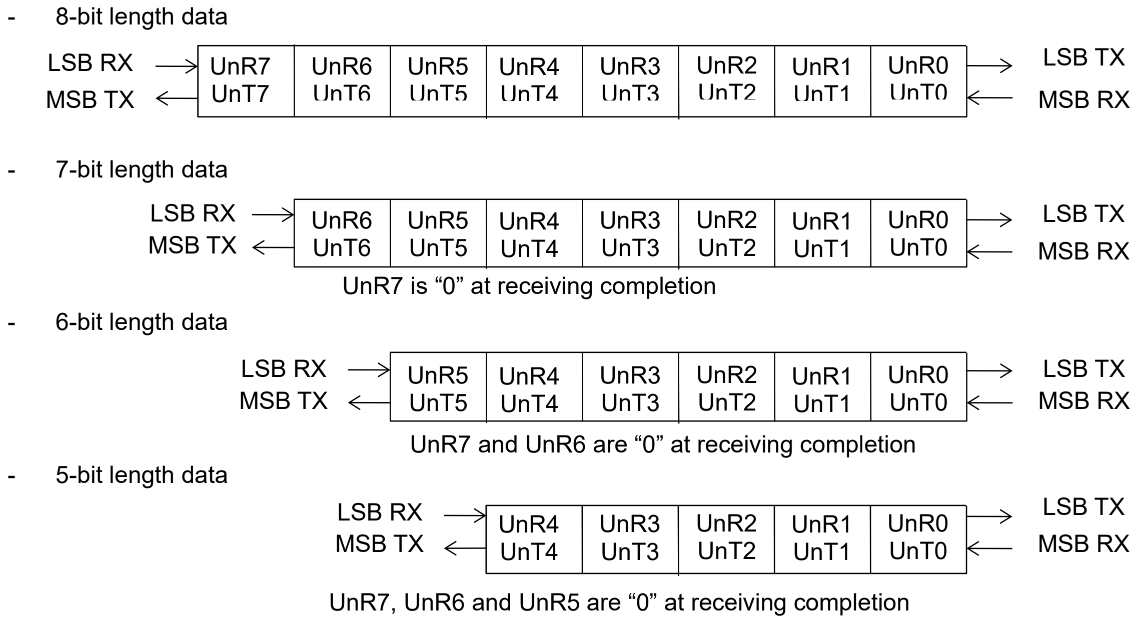


Figure 14-2 Format of Input/Output (LSB first)

14.3.2 Data Direction

Figure 14-3 shows a relationship between the transmission/reception buffer and data.



- 8-bit length data

LSB RX → | UnR7
UnT7 | UnR6
UnT6 | UnR5
UnT5 | UnR4
UnT4 | UnR3
UnT3 | UnR2
UnT2 | UnR1
UnT1 | UnR0
UnT0 | → LSB TX
MSB TX ← ... ← MSB RX

LSB RX → MSB TX ←	UnR7 UnT7	UnR6 UnT6	UnR5 UnT5	UnR4 UnT4	UnR3 UnT3	UnR2 UnT2	UnR1 UnT1	UnR0 UnT0	→ LSB TX ← MSB RX

- 7-bit length data

LSB RX → MSB TX ←	UnR6 UnT6	UnR5 UnT5	UnR4 UnT4	UnR3 UnT3	UnR2 UnT2	UnR1 UnT1	UnR0 UnT0	→ LSB TX ← MSB RX

UnR7 is “0” at receiving completion

- 6-bit length data

LSB RX → MSB TX ←	UnR5 UnT5	UnR4 UnT4	UnR3 UnT3	UnR2 UnT2	UnR1 UnT1	UnR0 UnT0	→ LSB TX ← MSB RX

UnR7 and UnR6 are “0” at receiving completion

- 5-bit length data

LSB RX → MSB TX ←	UnR4 UnT4	UnR3 UnT3	UnR2 UnT2	UnR1 UnT1	UnR0 UnT0	→ LSB TX ← MSB RX

UnR7, UnR6 and UnR5 are “0” at receiving completion

Figure 14-3 Relationship between the Transmission/Reception Buffer and Data

14.3.3 Baud Rate

The baud rate generator generates a baud rate using the base clock chosen in the UARTn mode register (UAnMOD). The setting values for the UARTn baud rate register (UAnBRT) and the UARTn baud rate adjustment register (UAnBRC) can be calculated by the following formulae.

UAnBRT = ROUNDDOWN (Base clock frequency (Hz) / Baud rate (bps)) – 1 + Carryover of UAnBRC
UAnBRC = ROUND ((Base clock frequency (Hz) % Baud rate (bps)) × 8 / Baud rate (bps))

where is ROUNDDOWN: Rounded down, ROUND: Rounded to the nearest whole number, %:Surplus.
Setting range of UAnBRC is 0 to 7. If the calculated value of UAnBRC is 8, add 1 to UAnBRT and set 0 to UAnBRC.

Example(1) : Base clock frequency: Approx.24 MHz (24.002560MHz), Baud rate ideal value: 115,200 bps

```
24.002560MHz ／ 115,200bps - 1 = 208.35555… - 1   = 207 (rounding down to the nearest integer)
                                                 = 0x00CF
(24.002560MHz % 115,200bps) × 8 ／ 115,200bps    = 40960 × 8 ／ 115,200
                                                 = 2.84444… = 3   (rounding to the nearest integer)
                                                 = 0x03
UAnBRT = 0x00CF,   UAnBRC = 0x03
```

Example(2) : Base clock frequency: Approx.16MHz (16.007168MHz), Baud rate ideal value:115,200bps

```
16.007168MHz ／ 115,200bps - 1 = 137.95111… - 1   = 137 (rounding down to the nearest integer)
                                                 = 0x0089
(16.007168MHz % 115,200bps) × 8 ／ 115,200bps    = 109568 × 8 ／ 115,200
                                                 = 7.60888… = 8   (rounding to the nearest integer)
                                                 = 0x08            UAnBRC carrier over occurred
UAnBRT = 0x0089+1 = 0x008A,   UAnBRC = 0x08 = 0x00
```

The actual baud rate calculated from the setting value for the baud rate can be expressed by the following formula:

Actual baud rate (bps) = [Base clock frequency] / {(UAnBRT + 1) + (UAnBRC / 8)}

Example: Base clock frequency: Approx.24 MHz (23.986176 MHz), Baud rate ideal value: 1200 bps
Actual baud rate (bps) = 24.002560MHz / {(0x4E21 + 1)} + (0x01 / 8)} ≈ 1199.99

Table 14-2 lists the count values for typical baud rates.

Table 14-2 Count Values for Typical Baud Rates

Base clock	Baud rate	UAnBRT	UAnBRC	Actual baud rate
PLL 24MHz (Approx. 24.0025600MHz)	1,200bps	0x4E21	0x01	1200.00bps
	2,400bps	0x2710	0x01	2399.99bps
	4,800bps	0x1387	0x04	4800.03bps
	9,600bps	0x09C3	0x02	9600.06bps
	19,200bps	0x04E1	0x01	19200.13bps
	38,400bps	0x0270	0x01	38396.42bps
	57,600bps	0x019F	0x06	57594.63bps
	115,200bps	0x00CF	0x03	115189.25bps
PLL 16MHz (Approx. 16.007168MHz)	300bps	0xD06C	0x02	300.00bps
	1,200bps	0x341A	0x02	1200.01bps
	2,400bps	0x1A0C	0x05	2400.01bps
	4,800bps	0x0D05	0x07	4799.93bps
	9,600bps	0x0682	0x03	9600.22bps
	19,200bps	0x0340	0x06	19199.00bps
	38,400bps	0x019F	0x07	38398.00bps
	57,600bps	0x0114	0x07	57605.64bps
	115,200bps	0x008A	0x00	115159.48bps
PLL 1MHz (Approx. 0.999424MHz)	300bps	0x0D02	0x03	300.00bps
	1,200bps	0x033F	0x07	1199.97bps
	2,400bps	0x019F	0x03	2400.30bps
	4,800bps	0x00CF	0x02	4799.16bps
	9,600bps	0x0067	0x01	9598.31bps
	19,200bps	0x0033	0x00	19219.69bps
	38,400bps	0x0019	0x00	38439.39bps
	57,600bps	0x0010	0x03	57520.81bps
	115,200bps	0x0007	0x05	115875.25bps
32.768kHz	200bps	0x00A2	0x07	199.95bps
	300bps	0x006C	0x02	299.93bps
	1,200bps	0x001A	0x02	1202.49bps
	2,400bps	0x000C	0x05	2404.99bps
	4,800bps	0x0005	0x07	4766.25bps

14.3.4 Transmission Operation

Transmission is started by setting the UnEN bit of the UART n control register (UAnCON) to "1" and set transfer data to UAnBUF1. The order of UAnEN setting and UAnBUF1 setting does not matter.
Figure 14-5 shows the operation timing for transmission.

When the UnEN bit is set to "1", the transmission status; UnTXF is set to “1” after 2 cycles of the system clock. (at (1)) An internal transfer clock of baud rate supplies after 2 cycles of the base clock(LSCLK0/HSCLK), and then the start bit is output the TXD pin. (at (2)) Subsequently, the transmitted data, a parity bit, and a stop bit are output.
When the start bit is output, the transmission buffer status flag; UnFUL is return to “0” and the transmission interrupt is requested on the rising edge of the internal transfer clock. (at (3)) In the UARTn transmission interrupt routine, the next data to be transmitted is written to the transmission buffer (UAnBUF1). Then the UnFUL is set to "1". It is same as (2) after transmission of the stop bit (at(4)). At this time if the UART transmission interrupt routine is terminated without writing the next data to the transmit buffer; it means the stop-bit is sent when UnFUL is not set to “1”, transmission is stop. Then the UnTXF bit is reset to "0", and the UART transmission interrupt is requested. (at (5))
The valid period for the next transmission data to be written to the transmission buffer is from when the UnFUL bit becomes “0” after the interrupt occurs to the termination of stop bit transmission. (at (6))

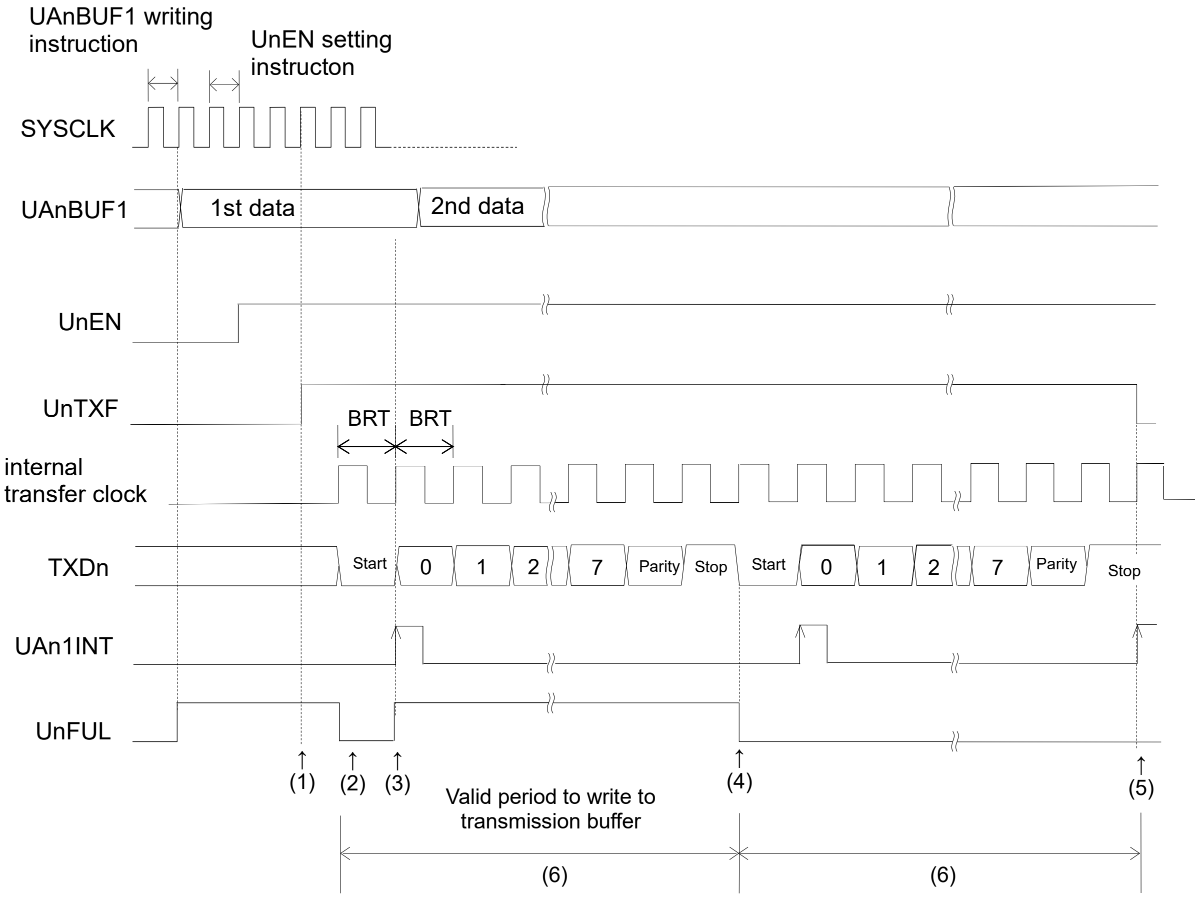


Figure 14-5 Transmission Timing

[Note]

- **The transmission is start when setting “1” to UnEN bit of UAnCON with the UnFUL bit =1. Write “1” to the UnFULC bit in the UAnSTAC register to reset the UnFUL bit, and then set “1” to the UnEN to allow transmission/reception, if the transmission data is not ready and the reception is permitted first.**

14.3.5 Reception Operation

A reception is started by setting the UnEN bit of the UART n control register (UAnCON) to "1".
Figure 14-6 shows the operation timing for reception.

When a reception starts, this module checks the data sent to the input pin RXD and waits for the arrival of a start bit. When detecting a start bit ((2) in Figure 14-6), It generates the internal transfer clock of the baud rate set with the start bit detect point as a reference and performs reception operation.
The shift register shifts in the data input to RXD on the rising edge of the internal transfer clock. The data and parity bit are shifted into the shift register and 5- to 8- bit received data is transferred to the reception buffer (UAnBUF0) concurrently with the falling edge of the internal transfer clock of (3) in Figure 14-6.
This module requests a UART reception interrupt on the rising edge of the internal transfer clock subsequent to the internal transfer clock by which the received data was fetched ((4) in Figure 14-6) and checks for a stop bit error and a parity bit error. When an error is detected, this module sets the corresponding bit of the UART n status register (UAnSTAT) to "1".

Parity error : UnPER ="1"
Overrun error : UnOER ="1"
Framing error : UnFER ="1"

As shown in Figure 14-6, the rise of the internal transfer clock is set so that it may fall into the middle of the bit interval of the received data.
A reception continues until the UnEN bit is reset to "0" by the program. When the UnEN bit is reset to "0" during reception, the received data may be destroyed. When the UnEN bit is reset to "0" during the "UnEN reset enable period" in Figure 14-6, the received data is protected.

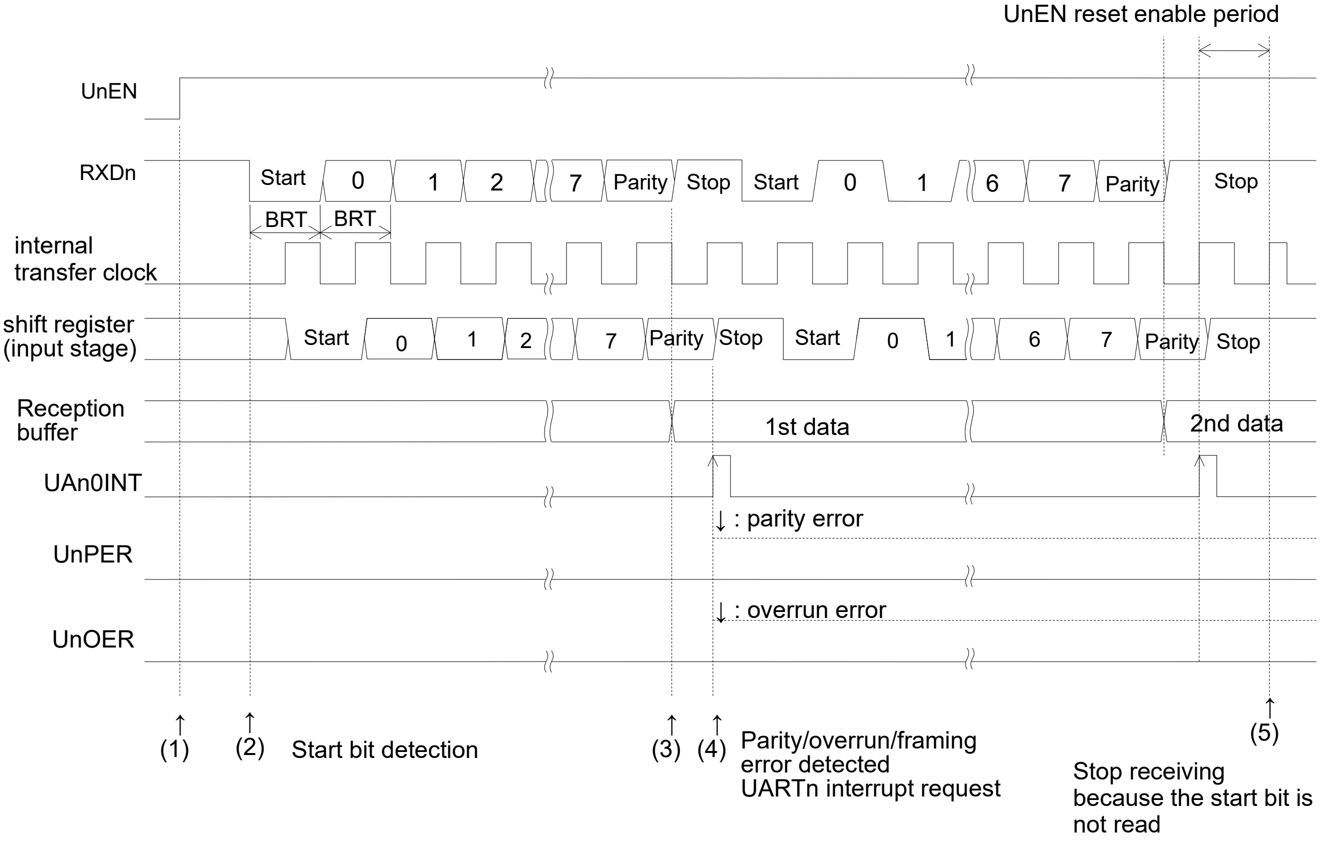


Figure 14-6 Reception Timing

14.3.5.1 Detection of Start Bit

The start bit is sampled with the baud rate generator clock selected by the UnCK0 bit of the UAnMOD register.
Therefore, the start bit detection may be delayed for one cycle of the baud rate generator clock at the maximum.
Figure 14-7 shows the start bit detection timing.

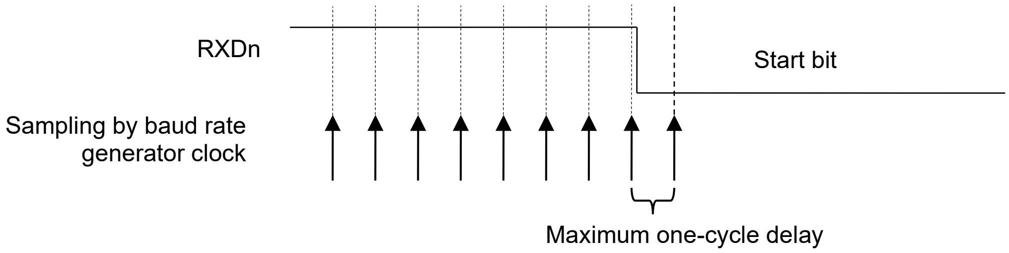
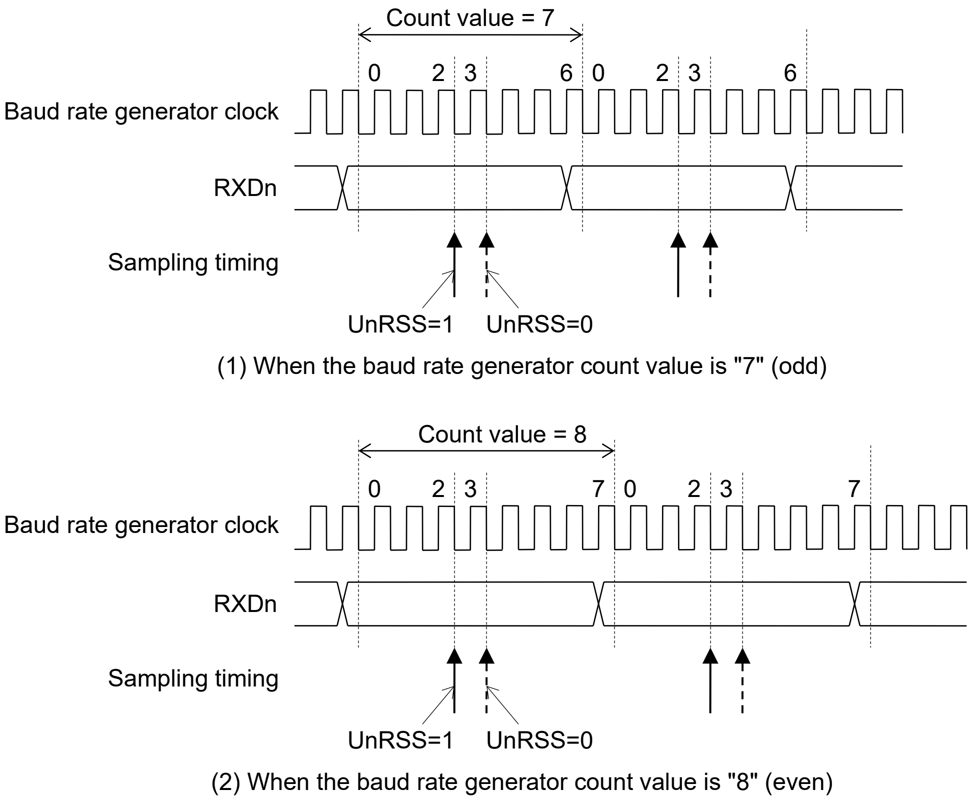


Figure 14-7 Start Bit Detection Timing (with Positive Logic)

14.3.5.2 Sampling Timing

When the start bit is detected, the received data that was input to RXDn is sampled almost at the center of the baud rate, and then loaded to the shift register.
This sampling timing the shift register uses to load data can be adjusted for one clock of the baud rate generator clock in the UnRSS bit of the UART n mode register (UAnMOD).
Figure 14-8 shows the relationship between the UnRSS bit and the sampling timing.



(1) When the baud rate generator count value is "7" (odd)

(2) When the baud rate generator count value is "8" (even)

Figure 14-8 Relationship between UnRSS Bit and Sampling Timing

14.3.5.3 Receiving Margin

If there is an error between the sender baud rate and the receiver baud rate generated by the baud rate generator, the error accumulates until the last stop bit loading in one frame, decreasing the reception margin.
Figure 14-9 shows the baud rate errors and reception margin waveforms.

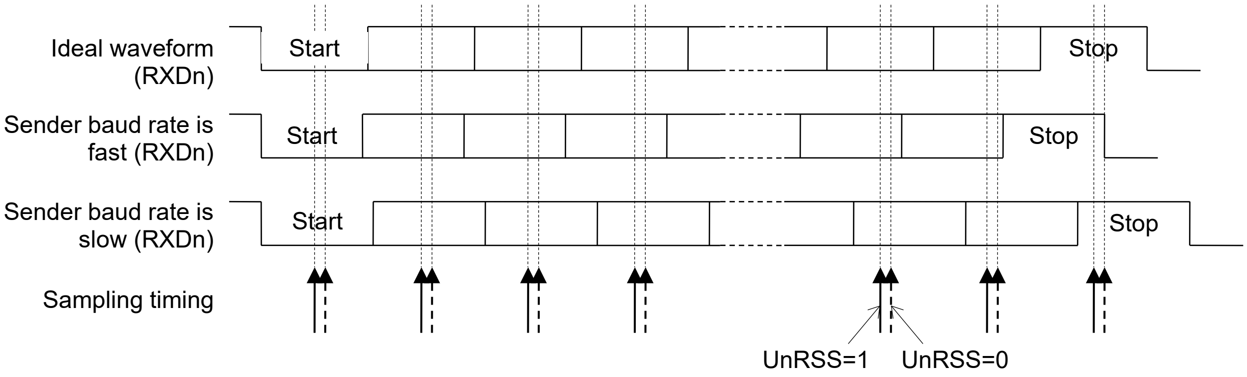


Figure 14-9 Baud Rate Errors and Reception Margin

[Note]

- **When designing the system, consider the difference of the baud rate between the transmission side and reception side, a delay of the start bit detection, signal degradation and noise influence, then adjust the baud rate and reception timing to ensure sufficient receiving margin.**

14.3.5.4 Reception Filter

This unit has reception data filter for a noise reduction. Figure 14-10 shows the RXD0 waveform before/after noise reduction.

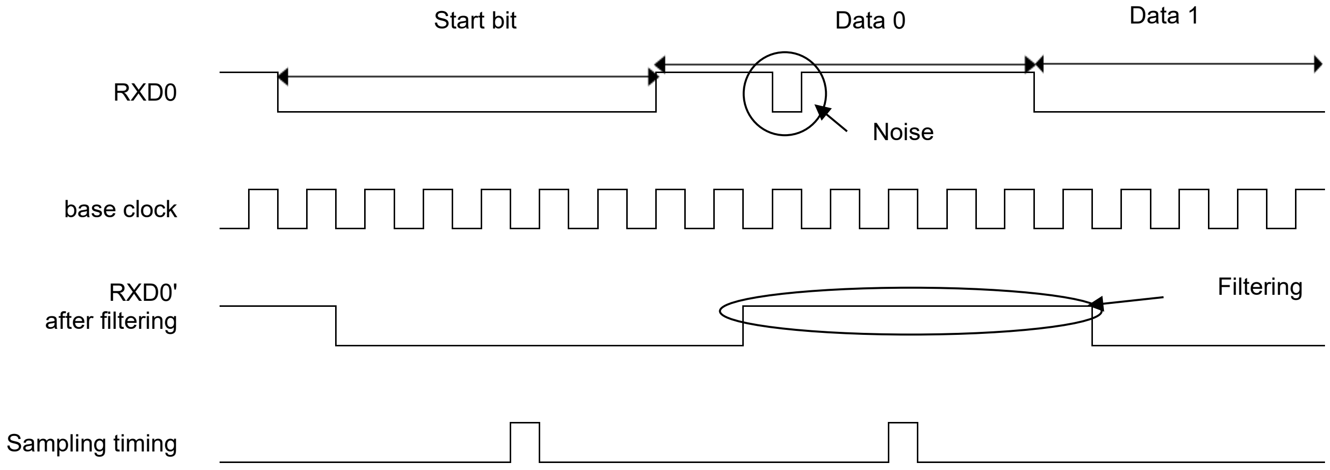


Figure 14-10 Noise reduction

14.3.6 Interrupt

Figure 14-11 shows the interrupt timing.
The transmission empty interrupt is generated as UAn1INT at the end of start-bit after a transmission buffer becomes empty. At this time, UnFUL=0 and UnTXF=1.
The transmission completion interrupt is generated as UAn1INT when a transmission is completed in condition of that the transmission buffer is empty. At this time, UnFUL=0, UnTXF=0.
The reception interrupt is generated as UAn0INT when reception data is stacked the buffer.

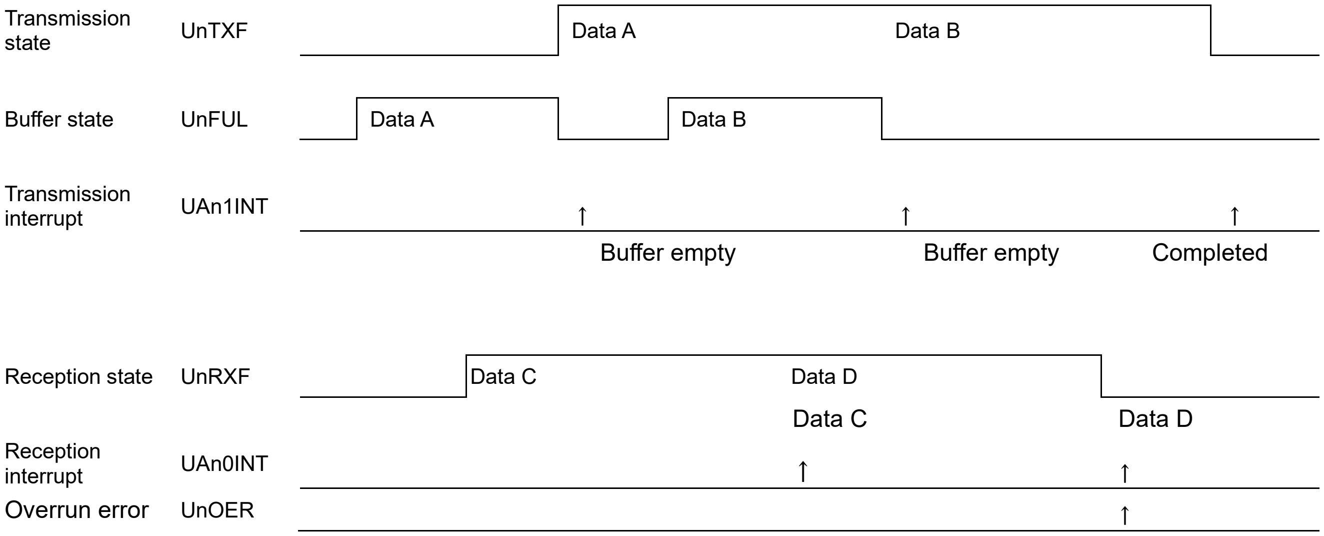


Figure 14-11 Interrupt Timing

Chapter 17 GPIO

17. GPIO

17.1 General Description

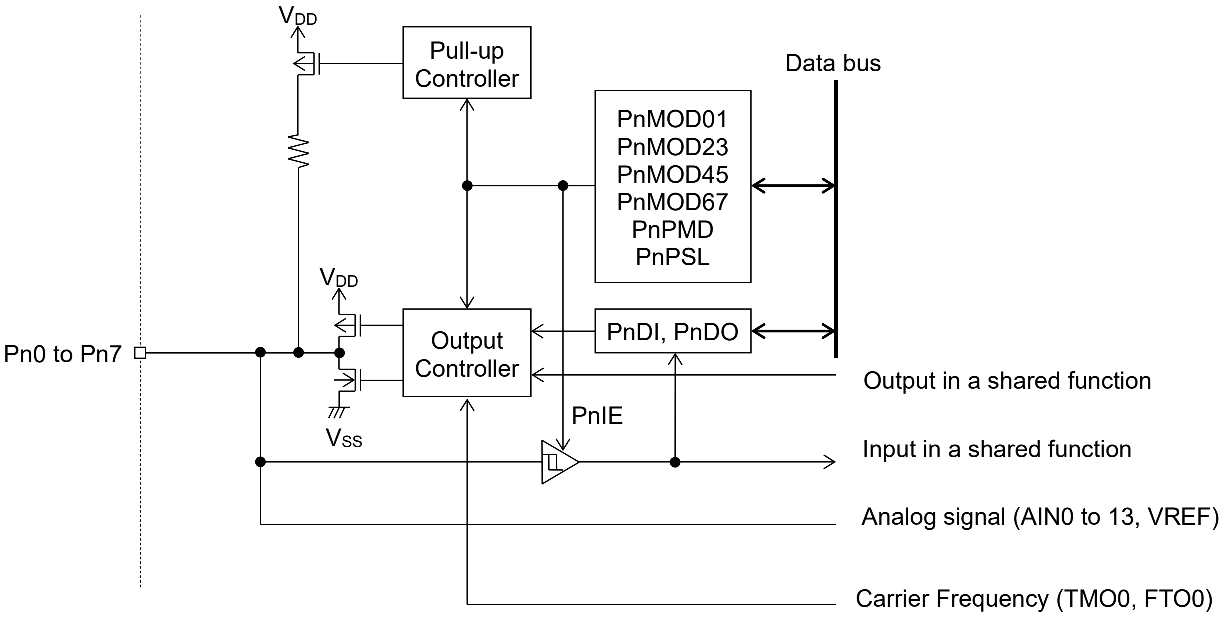
The general purpose port (GPIO) is 2 types of an input only port (GPI) and input/output port (GPI/O).
The input and output of a GPI/O is switchable on each pin. Max. 8 pins are available to read or to change the level of output in the same time. A general input port or output port shares a numbers of functions. See "1.3.2 List of Pins" or "1.3.3 Description of Pins" for more detail.
The GPIs are input is shared with the crystal resonator connection pins or debug/ISP interface 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

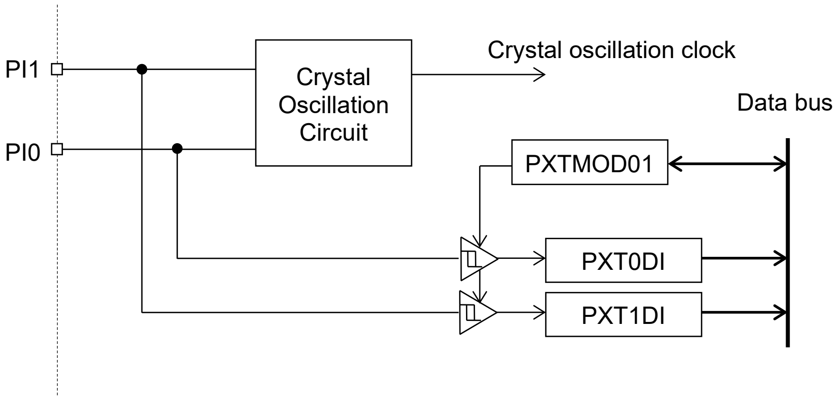
Figure 17-1 shows the configuration of the general purpose port. See "17.2.1 List of registers" for available pins and registers.


VDD
Pull-up Controller
Data bus
PnMOD01
PnMOD23
PnMOD45
PnMOD67
PnPMD
PnPSL
VDD
Pn0 to Pn7
Output Controller
PnDI, PnDO
Output in a shared function
PnIE
VSS
Input in a shared function
Analog signal (AIN0 to 13, VREF)
Carrier Frequency (TMO0, FTO0)

PnDI	: Port n data register (bit 7 to 0)
PnDO	: Port n data register (bit 15 to 8)
PnMOD01	: Port n mode register 01
PnMOD23	: Port n mode register 23
PnMOD45	: Port n mode register 45
PnMOD67	: Port n mode register 67
PnPMD	: Port n pulse mode register
PnPSL	: Port n pulse selection register

Figure 17-1 Configuration of GPI/O port n

Figure 17-2 shows the configuration of the GPI; PI0/PI1.



PXT0DI	: PORTXT data input register (bit 0)
PXT1DI	: PORTXT data input register (bit 1)
PXTMOD01	: PORTXT mode register01

Figure 17-2 Configuration of GPI; PI0/PI1

17.1.3 List of Pins

Table 17-1 List of Pins (● is available)

Pin Name	Primary Function	ML62Q2500 group		
		32 pin product	40 pin product	48 pin product
XT0	GPI(PI0) / Crystal resonator connection / EXI1	●	●	●
XT1	GPI(PI1) / Crystal resonator connection / External clock input	●	●	●
P00/TEST0	GPI / EXI0	●	●	●
P02	GPI/O	-	●	●
P03	GPI/O	-	●	●
P04	GPI/O / EXI1	●	●	●
P05	GPI/O / EXI2	●	●	●
P06	GPI/O / EXI3	●	●	●
P07	GPI/O / EXI4	●	●	●
P10	GPI/O / EXI5	●	●	●
P11	GPI/O / EXI6	●	●	●
P12	GPI/O / EXI7	●	●	●
P13	GPI/O / EXI3	●	●	●
P17	GPI/O	-	●	●
P20	GPI/O	●	●	●
P21	GPI/O	●	●	●
P22	GPI/O	●	●	●
P23	GPI/O	●	●	●
P24	GPI/O	●	●	●
P25	GPI/O	●	●	●
P26	GPI/O	●	●	●
P27	GPI/O / EXI2	●	●	●
P30	GPI/O / EXI4	●	●	●
P31	GPI/O / EXI5	●	●	●
P32	GPI/O / EXI6	●	●	●
P33	GPI/O / EXI7	●	●	●
P34	GPI/O / EXI0	-	●	●
P35	GPI/O / EXI1	-	●	●
P36	GPI/O / EXI2	-	●	●
P37	GPI/O / EXI3	-	-	●
P52	GPI/O / EXI4	-	-	●
P53	GPI/O / EXI5	-	-	●
P54	GPI/O	-	●	●
P55	GPI/O	-	●	●
P56	GPI/O	-	-	●
P57	GPI/O	-	-	●
P60	GPI/O / EXI6	-	-	●
P61	GPI/O / EXI7	-	-	●
P62	GPI/O	-	-	●
P70	GPI/O	●	●	●
P71	GPI/O	●	●	●
P72	GPI/O / EXI0	●	●	●
P73	GPI/O / EXI3	●	●	●

17.2 Description of Registers

17.2.1 List of Registers

Writing to SFRs of unequipped port is not available. PnDI return 0xFF for reading. Other SFRs return 0x0000/0x00 for reading.

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF200	Port 0 data register	P0DI	P0D	R	8/16	0xFF
0xF201		P0DO		R/W	8	0x00
0xF202	Port 0 mode register 0	P0MOD0	-	R/W	8/16	0x05
0xF203	Reserved	-	-	-	-	-
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	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	Reserved	-	-	-	-	-
0xF217		-		-	-	-
0xF218	Reserved	-	-	-	-	-
0xF219	Port 1 mode register 7	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	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

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
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	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	Port 3 mode register 45	P3MOD4	P3MOD45	R/W	8/16	0x00
0xF237		P3MOD5		R/W	8	0x00
0xF238	Port 3 mode register 67	P3MOD6	P3MOD67	R/W	8/16	0x00
0xF239		P3MOD7		R/W	8	0x00
0xF23A ~0xF23F	Reserved	-	-	-	-	-
0xF240 ~0xF24F	Reserved	-	-	-	-	-
0xF250	Port 5 data register	P5DI	P5D	R	8/16	0xFF
0xF251		P5DO		R/W	8	0x00
0xF252	Reserved	-	-	-	-	-
0xF253						
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 ~0xF25F	Reserved	-	-	-	-	-
0xF260	Port 6 data register	P6DI	P6D	R	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 2	P6MOD2	-	R/W	8/16	0x00
0xF265	Reserved	-	-	-	-	-
0xF266 ~0xF26F	Reserved	-	-	-	-	-

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF270	Port 7 data register	P7DI	P7D	R	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 ~0xF279	Reserved	-	-	-	-	-
0xF27A	Port 7 pulse mode register	P7PMDL	P7PMD	R/W	8/16	0x00
0xF27B		P7PMDH		R/W	8	0x00
0xF27C	Port 7 pulse selection register	P7PSLL	P7PSL	R/W	8/16	0x00
0xF27D		P7PSLH		R/W	8	0x00
0xF27E ~0xF2EF	Reserved	-	-	-	-	-
0xF2F0	PORTXT data input register	PXTDI	-	R	8	0x03
0xF2F1	Reserved	-		-	-	-
0xF2F2	PORTXT mode register 01	PXTMOD0	PXTMOD01	R/W	8/16	0x00
0xF2F3		PXTMOD1		R/W	8	0x00

Table 17-2 List of Registers / Bits

Port Name	Pin Name	Control register / bit						Available		
								ML62Q2500 group		
		Port n data register (PnD)		Port n mode register m (PnMODm)	Port n pulse mode register (PnPMD)		Port n pulse selection register (PnPSL)	32 pin product	40 pin product	48 pin product
Port XT	PI00	-	PXT0DI	PXTMOD0	-	-	-	●	●	●
	PI01	-	PXT1DI	PXTMOD1	-	-	-	●	●	●
Port 0	P00	-	P00DI	P0MOD0	-	-	-	●	●	●
	P02	P02DO	P02DI	P0MOD2	-	-	-	-	●	●
	P03	P03DO	P03DI	P0MOD3	-	-	-	-	●	●
	P04	P04DO	P04DI	P0MOD4	-	-	-	●	●	●
	P05	P05DO	P05DI	P0MOD5	-	-	-	●	●	●
	P06	P06DO	P06DI	P0MOD6	-	-	-	●	●	●
	P07	P07DO	P07DI	P0MOD7	P07PLVL	P07PEN	P07PSL	●	●	●
Port 1	P10	P10DO	P10DI	P1MOD0	-	-	-	●	●	●
	P11	P11DO	P11DI	P1MOD1	P11PLVL	P11PEN	P11PSL	●	●	●
	P12	P12DO	P12DI	P1MOD2	-	-	-	●	●	●
	P13	P13DO	P13DI	P1MOD3	-	-	-	●	●	●
	P17	P17DO	P17DI	P1MOD7	-	-	-	-	●	●
Port 2	P20	P20DO	P20DI	P2MOD0	-	-	-	●	●	●
	P21	P21DO	P21DI	P2MOD1	P21PLVL	P21PEN	P21PSL	●	●	●
	P22	P22DO	P22DI	P2MOD2	-	-	-	●	●	●
	P23	P23DO	P23DI	P2MOD3	-	-	-	●	●	●
	P24	P24DO	P24DI	P2MOD4	-	-	-	●	●	●
	P25	P25DO	P25DI	P2MOD5	-	-	-	●	●	●
	P26	P26DO	P26DI	P2MOD6	-	-	-	●	●	●
	P27	P27DO	P27DI	P2MOD7	-	-	-	●	●	●
Port 3	P30	P30DO	P30DI	P3MOD0	-	-	-	●	●	●
	P31	P31DO	P31DI	P3MOD1	-	-	-	●	●	●
	P32	P32DO	P32DI	P3MOD2	-	-	-	●	●	●
	P33	P33DO	P33DI	P3MOD3	-	-	-	●	●	●
	P34	P34DO	P34DI	P3MOD4*1	-	-	-	-	●	●
	P35	P35DO	P35DI	P3MOD5*1	-	-	-	-	●	●
	P36	P36DO	P36DI	P3MOD6*1	-	-	-	-	●	●
	P37	P37DO	P37DI	P3MOD7*1	-	-	-	-	-	●
Port 5	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	-	-	-	-	-	●

Port Name	Pin Name	Control register / bit						Available		
								ML62Q2500 group		
		Port n data register (PnD)		Port n mode register m (PnMODm)	Port n pulse mode register (PnPMD)		Port n pulse selection register (PnPSL)	32 pin product	40 pin product	48 pin product
Port 7	P70	P70DO	P70DI	P7MOD0	-	-	-	●	●	●
	P71	P71DO	P71DI	P7MOD1	-	-	-	●	●	●
	P72	P72DO	P72DI	P7MOD2	-	-	-	●	●	●
	P73	P73DO	P73DI	P7MOD3	P73PLVL	P73PEN	P73PSL	●	●	●

*1 P34,P35,P36,P37 pins have GPI/O function only. So Writing to bit 4-7 of their PnPMODm is not available.

17.2.2 Port 0 Data Register (P0D)

P0D is a SFR used to read the level of the port n pin and write output data.
The input level of the port 0 pins can be read by reading P0DI in the input mode. Data written to P0DO in the output mode are output to the port 0 pins. The data written to P0DO is readable. The bit can be set when output is enabled or disabled. Enable or disable the input or output by using the port 0 mode registers. See Table 17-2 "List of Registers / Bits" to check available pins and bits. Write "0" to the bits of P0DO that have no corresponding pin.

Address:	0xF200(P0DI/P0D), 0xF201(P0DO)
Access:	R/W
Access size:	8/16 bit
Initial value:	0x00FF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	P0D															
Byte	P0DO								P0DI							
Bit	P07DO	P06DO	P05DO	P04DO	P03DO	P02DO	-	-	P07DI	P06DI	P05DI	P04DI	P03DI	P02DI	-	P00DI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	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 10	P07DO to P02DO	These bits are used to set the output level of port 0 pins. 0: Output "L" (Initial value) 1: Output "H"
9 to 8, 1	-	Reserved bits
7 to 2, 0	P07DI to P00DI	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 Data Register (PnD:n＝1 to 3, 5 to 7)

PnD is a SFR used to read the level of the port n pin and write output data.
The input level of the port n pins can be read by reading PnDI in the input mode. Data written to PnDO in the output mode are output to the port n pins. The data written to PnDO is readable. The bit can be set when output is enabled or disabled. Enable or disable the input or output by using the port n mode register. See Table 17-2 "List of Registers / Bits" to check available pins and bits. Write "0" to the bits of PnDO that have no corresponding pin.

Address:	0xF210(P1DI/P1D), 0xF211(P1DO), 0xF220(P2DI/P2D), 0xF221(P2DO), 0xF230(P3DI/P3D), 0xF231(P3DO), 0xF250(P5DI/P5D), 0xF251(P5DO), 0xF260(P6DI/P6D), 0xF261(P6DO), 0xF270(P7DI/P7D), 0xF271(P7DO)
Access:	R/W
Access size:	8/16 bit
Initial value:	0x00FF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnD															
Byte	PnDO								PnDI							
Bit	Pn7DO	Pn6DO	Pn5DO	Pn4DO	Pn3DO	Pn2DO	Pn1DO	Pn0DO	Pn7DI	Pn6DI	Pn5DI	Pn4DI	Pn3DI	Pn2DI	Pn1DI	Pn0DI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 8	Pn7DO to Pn0DO	These bits are used to set the output level of port n pins. 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 pins. 0: The input level is "L" 1: The input level is "H" (Initial value)

17.2.4 Port 0 Mode Register 0 (P0MOD0)

P0MOD0 is SFR to set P00 pin.

Address:	0xF202(P0MOD0)
Access:	R/W
Access size:	8/16 bit
Initial value:	0x0005

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								P0MOD0							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	P00PU	P00OE	P00IE
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit No.	Bit symbol name	Description
15 to 3	-	Reserved bits
2	P00PU	This bit is used to enable the internal pull-up resistor of P00 pin. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor
1	P00OE	This bit is used to enable the output of P00 pin 0: Disable the output (Initial value) 1: Enable the output
0	P00IE	This bit is used to enable the input of P00 pin 0: Disable the input (Initial value) 1: Enable the input

[Note]

- **The P00 pin is initially configured as the input with pull-up resistor. If input "L" level at an initial setting, the input current flows.**

17.2.5 Port n Mode Register 01 (PnMOD01:n＝1 to 3, 5 to 7)

PnMOD01 is a 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 pins.

Address: 0xF212(P1MOD0/P1MOD01), 0xF213(P1MOD1),
0xF222(P2MOD0/P2MOD01), 0xF223(P2MOD1), 0xF232(P3MOD0/P3MOD01), 0xF233(P3MOD1),
0xF262(P6MOD0/P6MOD01), 0xF263(P6MOD1), 0xF272(P7MOD0/P7MOD01), 0xF273(P7MOD1)
Access: R/W
Access size: 8/16 bit
Initial value: 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD01															
Byte	PnMOD1								PnMOD0							
Bit	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	0	0

Bit No.	Bit symbol name	Description
15 to 12	Pn1MD3 to Pn1MD0	These bits are used to choose the shared function of Pn1 pins. For the details of the shared function, see Table 1-3 "ML62Q2500 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 pins. 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
10	Pn1PU	This bit is used to enable the internal pull-up resistor of Pn1 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor
9	Pn1OE	This bit is used to enable the output of Pn1 pins. 0: Disable the output (Initial value) 1: Enable the output
8	Pn1IE	This bit is used to enable the input of Pn1 pins. 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 pins. For the details of the shared function, see Table 1-3 "ML62Q2500 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)

Bit No.	Bit symbol name	Description
3	Pn0OD	These bits are used choose the output type of Pn0 pins. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is 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 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor
1	Pn0OE	This bit is used to enable the output of Pn0 pins. 0: Disable the output (Initial value) 1: Enable the output
0	Pn0IE	This bit is used to enable the input of Pn0 pins. 0: Disable the input (Initial value) 1: Enable the input

[Note]

- **Be sure to set the PnMOD01 registers before setting EICON0, EIMOD0 and IE1 registers. If setting the PnMOD01 register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**
- **Don't set un-assigned shared functions on the PnmMD3-0 bits.**

17.2.6 Port n Mode Register 23 (PnMOD23:n＝0 to 3, 5 to 7)

PnMOD23 is a 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 pins.

Address:	0xF204(P0MOD2/P0MOD23), 0xF205(P0MOD3), 0xF214(P1MOD2/P1MOD23), 0xF215(P1MOD3), 0xF224(P2MOD2/P2MOD23), 0xF225(P2MOD3), 0xF234(P3MOD2/P3MOD23), 0xF235(P3MOD3), 0xF254(P5MOD2/P5MOD23), 0xF255(P5MOD3), 0xF264(P6MOD2), 0xF274(P7MOD2/P7MOD23), 0xF275(P7MOD3)
Access:	R/W
Access size:	8/16 bit
Initial value:	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD23															
Byte	PnMOD3								PnMOD2							
Bit	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 pins. For the details of the shared function, see Table 1-3 "ML62Q2500 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	These bits are used choose the output type of Pn3 pins. 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
10	Pn3PU	This bit is used to enable the internal pull-up resistor of Pn3 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor
9	Pn3OE	This bit is used to enable the output of Pn3 pins. 0: Disable the output (Initial value) 1: Enable the output
8	Pn3IE	This bit is used to enable the input of Pn3 pins. 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 pins. For the details of the shared function, see Table 1-3 "ML62Q2500 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)

Bit No.	Bit symbol name	Description
3	Pn2OD	These bits are used choose the output type of Pn2 pins. 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 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor
1	Pn2OE	This bit is used to enable the output of Pn2 pins. 0: Disable the output (Initial value) 1: Enable the output
0	Pn2IE	This bit is used to enable the input of Pn2 pins. 0: Disable the input (Initial value) 1: Enable the input

[Note]

- **Be sure to set the PnMOD23 registers before setting EICON0, EIMOD0 and IE1 registers. If setting the PnMOD23 register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**
- **Don't set un-assigned shared functions on the PnmMD3-0 bits.**

17.2.7 Port n Mode Register 45 (PnMOD45:n＝0 to 3, 5 to 7)

PnMOD45 is a 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 pins.

Address:	0xF206（P0MOD4/P0MOD45）, 0xF207（P0MOD5）,0xF226（P2MOD4/P2MOD45）, 0xF227（P2MOD5）, 0xF236（P3MOD4/P3MOD45）, 0xF237（P3MOD5）,0xF256（P5MOD4/P5MOD45）, 0xF257（P5MOD5）
Access:	R/W
Access size:	8/16 bit
Initial value:	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD45															
Byte	PnMOD5								PnMOD4							
Bit	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 pins. For the details of the shared function, see Table 1-3 "ML62Q2500 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) * P35 pin does not have shared function. P35MD3-0 are not writable. The reading value is "0".
11	Pn5OD	These bits are used choose the output type of Pn5 pins. 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
10	Pn5PU	This bit is used to enable the internal pull-up resistor of Pn5 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor
9	Pn5OE	This bit is used to enable the output of Pn5 pins. 0: Disable the output (Initial value) 1: Enable the output
8	Pn5IE	This bit is used to enable the input of Pn5 pins. 0: Disable the input (Initial value) 1: Enable the input

Bit No.	Bit symbol name	Description
7 to 4	Pn4MD3 to Pn4MD0	These bits are used to choose the shared function of Pn4 pins. For the details of the shared function, see Table 1-3 "ML62Q2500 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) * P34 pin does not have shared function. P35MD3-0 are not writable. The reading value is "0".
3	Pn4OD	These bits are used choose the output type of Pn4 pins. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is 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 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor
1	Pn4OE	This bit is used to enable the output of Pn4 pins. 0: Disable the output (Initial value) 1: Enable the output
0	Pn4IE	This bit is used to enable the input of Pn4 pins. 0: Disable the input (Initial value) 1: Enable the input

[Note]

- **Be sure to set the PnMOD45 registers before setting EICON0, EIMOD0 and IE1 registers. If setting the PnMOD45 register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**
- **Don't set un-assigned shared functions on the PnmMD3-0 bits.**

17.2.8 Port n Mode Register 67 (PnMOD67:n＝0 to 3, 5)

PnMOD67 is a 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 pins.

Address:	0xF208（P0MOD6/P0MOD67）, 0xF209（P0MOD7）, 0xF219（P1MOD7）, 0xF228（P2MOD6/P2MOD67）, 0xF229（P2MOD7）, 0xF238（P3MOD6/P3MOD67）, 0xF239（P3MOD7）, 0xF258（P5MOD6/P5MOD67）, 0xF259（P5MOD7）
Access:	R/W
Access size:	8/16 bit
Initial value:	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD67															
Byte	PnMOD7								PnMOD6							
Bit	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 pins. For the details of the shared function, see Table 1-3 "ML62Q2500 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) * P37 pin does not have shared function. P37MD3-0 are not writable. The reading value is "0".
11	Pn7OD	These bits are used choose the output type of Pn7 pins. 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
10	Pn7PU	This bit is used to enable the internal pull-up resistor of Pn7 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor
9	Pn7OE	This bit is used to enable the output of Pn7 pins. 0: Disable the output (Initial value) 1: Enable the output
8	Pn7IE	This bit is used to enable the input of Pn7 pins. 0: Disable the input (Initial value) 1: Enable the input

Bit No.	Bit symbol name	Description
7 to 4	Pn6MD3 to Pn6MD0	These bits are used to choose the shared function of Pn6 pins. For the details of the shared function, see Table 1-3 "ML62Q2500 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) * P36 pin does not have shared function. P36MD3-0 are not writable. The reading value is "0".
3	Pn6OD	These bits are used choose the output type of Pn6 pins. 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 pins. 0: Without a pull-up resistor (Initial value) 1: With a pull-up resistor
1	Pn6OE	This bit is used to enable the output of Pn6 pins. 0: Disable the output (Initial value) 1: Enable the output
0	Pn6IE	This bit is used to enable the input of Pn6 pins. 0: Disable the input (Initial value) 1: Enable the input

[Note]

- **Be sure to set the PnMOD67 registers before setting EICON0, EIMOD0 and IE1 registers. If setting the PnMOD67 register when the interrupt is enabled, unexpected interrupts may happen.**
- **It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.**
- **Don't set un-assigned shared functions on the PnmMD3-0 bits.**

17.2.9 Port n Pulse Mode Register (PnPMD:n＝0 to 2, 7)

PnPMD is a SFR used when outputting a carrier frequency (pulse output) to the port n.
See Table 17-2 "List of Registers / Bits" to check available pins and bits. Write "0" to the bits of PnPMD register that have no corresponding pin.

Address:	0xF20A（P0PMDL/P0PMD）, 0xF20B（P0PMDH）, 0xF21A（P1PMDL/P1PMD）, 0xF21B（P1PMDH）, 0xF22A（P2PMDL/P2PMD）, 0xF22B（P2PMDH）, 0xF27A（P7PMDL/P7PMD）, 0xF27B（P7PMDH）
Access:	R/W
Access size:	8/16 bit
Initial value:	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnPMD															
Byte	PnPMDH								PnPMDL							
Bit	Pn7PLVL	Pn6PLVL	Pn5PLVL	Pn4PLVL	Pn3PLVL	Pn2PLVL	Pn1PLVL	Pn0PLVL	Pn7PEN	Pn6PEN	Pn5PEN	Pn4PEN	Pn3PEN	Pn2PEN	Pn1PEN	Pn0PEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	Pn7PLVL to Pn0PLVL	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

17.2.10 Port n Pulse Selection Register (PnPSL:n＝0 to 2, 7)

PnPSL is a 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）, 0xF27C（P7PSLL/P7PSL）, 0xF27D（P7PSLH）
Accress:	R/W
Access size:	8/16 bit
Initial value:	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnPSL															
Byte	PnPSLH								PnPSLL							
Bit	-	-	-	-	-	-	-	-	Pn7PSL	Pn6PSL	Pn5PSL	Pn4PSL	Pn3PSL	Pn2PSL	Pn1PSL	Pn0PSL
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 8	-	Reserved bits
7 to 0	PnmPSL	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", where m = 0 to 7). 0: 16-bit timer 0 output (TMH0OUT) (Initial value) 1: Functional timer 0 output (FTM0P)

17.2.11 PORTXT Data Input Register (PXTDI)

PXTDI is a SFR used for reading the level of XT0/XT1 pin.
The level of XT0/PI0 and XT1/PI1 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:	8 bit
Initial value:	0x03

	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	1	1

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1	PXT1DI	This bit is used for reading the level of XT1/PI1. 0: The input level of XT1/PI1 pin is “L” 1: The input level of XT1/PI1 pin is “H”
0	PXT0DI	This bit is used for reading the level of XT0/PI0. 0: The input level of XT0/PI0 pin is “L” 1: The input level of XT0/PI0 pin is “H”

[Note]

- **PI0 and PI1 are unavailable to use as input ports when using the crystal resonator for the oscillation clock. Also, PI1 is unavailable to use as an input port when using the XT1 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.**

17.2.12 PORTXT Mode Register 01 (PXTMOD01)

PXTMOD01 is a SFR used to choose the input mode of the XT0/PI0 pin and XT1/PI1 pin.
The port is unavailable to use when connecting the crystal resonator.

Address:	0xF2F2(PXTMOD0/PXTMOD01), 0xF2F3(PXTMOD1)
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	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/PI1 pin. 0: High impedance (Initial value) 1: Input mode
7 to 1	-	Reserved bits
0	PXT0IE	This bit is used to choose the input mode of the XT0/PI0 pin. 0: High impedance (Initial value) 1: Input mode

17.3 Description of Operation

The following shows description of port functions, where "n" is port number 0 to 3 and 5 to 7, and "m" is bit number 0 to 7.

17.3.1 Input

Each pin of port n except for the P00 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 of pins except for the P00. As one of the P00, input, input enabled and pull-up are selected.

17.3.2 Output

Each pin of port n sets 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 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 EXI7), analog input for SA-ADC or crystal/external clock input can be used as the primary function other than the input/output function.
When using EXI0 to EXI7 as external interrupt input and the clock inputs of the 16-bit timer or trigger/clock input of the functional timer, set the PnMODm register of the applicable port to input enabled (PnmIE bit="1").
When using as analog input for SA-ADC; AIN0 to AIN13 and VREF, set the PnMODm register of the applicable port to input disabled (PnmIE bit="0" and PnmOE bit="0").
When using as crystal/external clock input, set by the FLMOD register; refer to Chaptuer 6. If it have been set, it is ignored to set by PORTXT mode register.

See Chapter 18 "External Interrupt Control" for external interrupts, Chapter 8 "16-Bit Timer" for clock input of the 16-bit timer, and Chapter 9 "Functional Timer" for external trigger/clock input of the functional timer.

17.3.4 Shared Function

Each pin of port n can use 2nd to 7th functions as the shared function.
Set PnmMD3 to PnmMD0 bits of the PnMODm register to choose each of the 1st to 8th functions.

17.3.5 Carrier Frequency Output

17.3.5.1 Carrier Frequency Output Operation

A carrier frequency signal can be output from port n by setting the PnPMD Register. See Table 17-2 "List of Registers/Bits" for pins supporting the carrier frequency output function. For the carrier frequency output, either of 16-bit timer 0 output (TMO0) or functional timer 0 output(FTO0) 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.

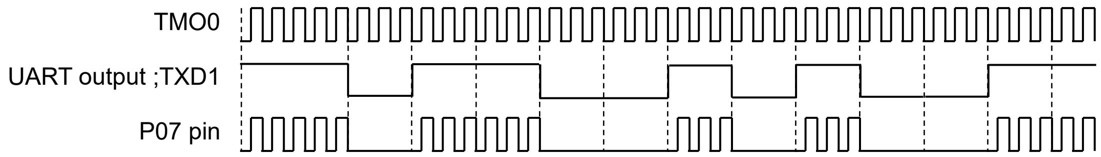


Figure 17-3 Example of Carrier Frequency Output When P07 Pin is Assigned to UART Output Pin (P07PEN bit ="1", P07PLVL bit= "0" of P0PMD register)

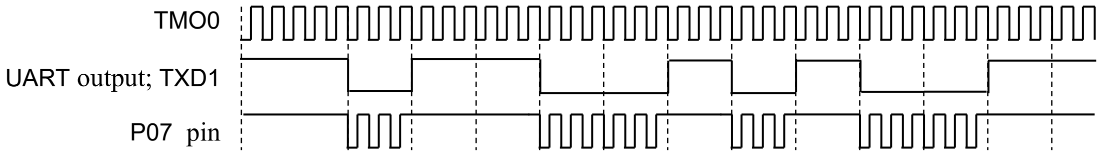
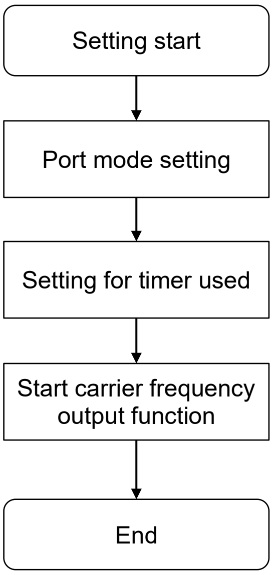


Figure 17-4 Example of Carrier Frequency Output When P07 Pin is Assigned to UART Output Pin (P07PEN bit ="1", P07PLVL bit= "1" of P0PMD register)

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 P07 pin used, TXD1 shared function, functional timer 0 output (FTO0) used as a timer, carrier frequency output at "L" level).



Set output enabled, output type, 3rd function with the mode register (P0MOD7 register)
P0MOD7 = 0x22

Set timer used for outputting carrier frequency using the P07PSL bit of the P0PSL register
P07PSL = 1

Set output function enabled using the P07PEN bit of the P0PMD register and output determination level using the P07PLVL bit
P07PLVL = 1
P07PEN = 1

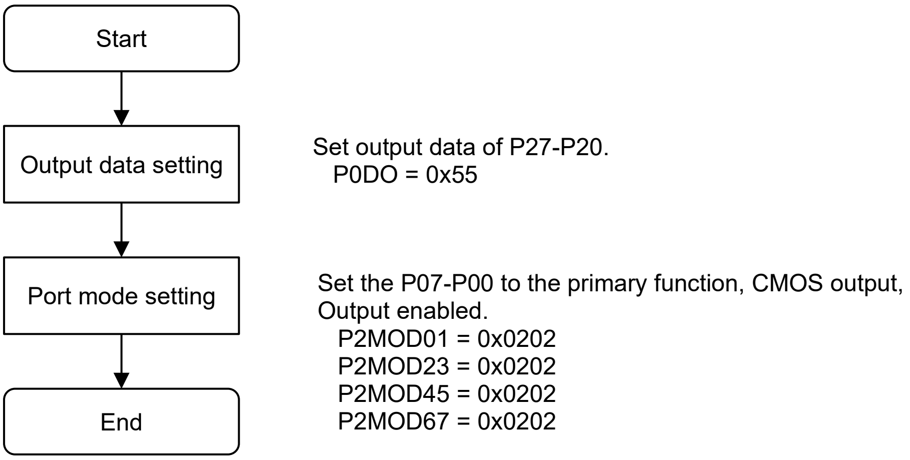
Figure 17-5 Example of Carrier Frequency Output Function Setting Procedure

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.

17.3.7 Port Setting Example

Figure 17-6 shows an example for setting port registers to output 0x55 to a port 2. It is also available to set output level before outputting.



Set output data of P27-P20.
P0DO = 0x55

Set the P07-P00 to the primary function, CMOS output,
Output enabled.
P2MOD01 = 0x0202
P2MOD23 = 0x0202
P2MOD45 = 0x0202
P2MOD67 = 0x0202

Figure 17-6 Setting example to output data to port 2

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.
When using the on-chip debug function or ISP function, P00/TEST0 is unavailable to use as the general purpose port.
When using the general port, P00/TEST0 is unavailable to use for the on-chip debug function or ISP function.

Chapter 18 External Interrupt Function

18. External Interrupt Function

18.1 General Description

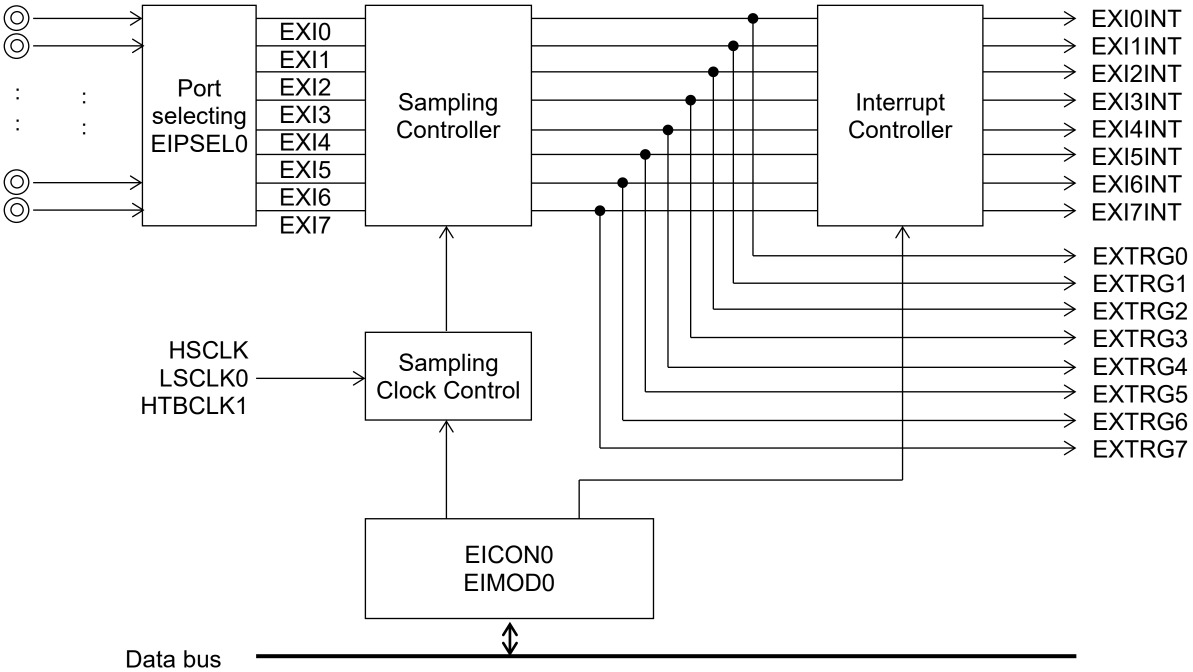
The external interrupt function generates interrupts by signals input to the general ports.
The interrupt channel has each dedicated interrupt vector. See Chapter 5 "Interrupt" for details of the interrupt vector.

18.1.1 Features

- Maskable 8 interrupts
- Each interrupt is assigned from max. 3 pins
- 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 LSCLK0, HSCLK or HTBCLK1)

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
EIPSEL0: External interrupt port select register 0

Figure 18-1 Configuration of External Interrupt Function

18.1.3 List of Pins

The external interrupt is assigned to the primary function of the general port.

Pin name	I/O	Function
EXI0	I	External Interrupt Input 0
EXI1	I	External Interrupt Input 1
EXI2	I	External Interrupt Input 2
EXI3	I	External Interrupt Input 3
EXI4	I	External Interrupt Input 4
EXI5	I	External Interrupt Input 5
EXI6	I	External Interrupt Input 6
EXI7	I	External Interrupt Input 7

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	ML62Q2500 group		
					32 pin product	40 pin product	48 pin product
EXI0	P00	Primary function	P0MOD0	0000_0X01*1	●	●	●
	P72		P7MOD2	0000_0X01*1	●	●	●
	P34		P3MOD4	0000_0X01*1	-	●	●
EXI1	P04		P0MOD4	0000_0X01*1	●	●	●
	XT0(PI0)		PXTMOD0	0000_0001*1	●	●	●
	P35		P3MOD5	0000_0X01*1	-	●	●
EXI2	P05		P0MOD5	0000_0X01*1	●	●	●
	P27		P2MOD7	0000_0X01*1	●	●	●
	P36		P3MOD6	0000_0X01*1	-	●	●
EXI3	P06		P0MOD6	0000_0X01*1	●	●	●
	P73		P7MOD3	0000_0X01*1	●	●	●
	P37		P3MOD7	0000_0X01*1	-	-	●
EXI4	P07		P0MOD7	0000_0X01*1	●	●	●
	P30		P3MOD0	0000_0X01*1	●	●	●
	P52		P5MOD2	0000_0X01*1	-	-	●
EXI5	P10		P1MOD0	0000_0X01*1	●	●	●
	P31		P3MOD1	0000_0X01*1	●	●	●
	P53		P5MOD3	0000_0X01*1	-	-	●
EXI6	P11		P1MOD1	0000_0X01*1	●	●	●
	P32		P3MOD2	0000_0X01*1	●	●	●
	P60		P6MOD0	0000_0X01*1	-	●	●
EXI7	P12		P1MOD2	0000_0X01*1	●	●	●
	P33		P3MOD3	0000_0X01*1	-	●	●
	P61		P6MOD1	0000_0X01*1	-	-	●

*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	Reserved	-	-	-	-	-
0xF048	External interrupt mode register 0	EIMOD0L	EIMOD0	R/W	8/16	0x00
0xF049		EIMOD0H		R/W	8	0x00
0xF04A	Reserved	-	-	-	-	-
0xF04B	Reserved	-	-	-	-	-
0xF04C	External interrupt port selection register 0	EIPSEL0L	EIPSEL0	R/W	8/16	0x00
0xF04D		EIPSEL0H		R/W	8	0x00

18.2.2 External Interrupt Control Register 0 (EICON0)

This is a 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 :	8/16 bit
Initial value :	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EICON0															
Byte	EICON0H								EICON0L							
Bit	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). 00 : Interrupt disabled (Initial value) 01 : Falling-edge interrupt 10 : Rising-edge interrupt 11 : Both-edge interrupt The relation of the bit number and the target external interrupt: Bit 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
7 to 0	PI7E0 to PI0E0	

18.2.3 External Interrupt Mode Register 0 (EIMOD0)

This is a 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 :	8/16 bit
Initial value :	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EIMOD0															
Byte	EIMOD0H								EIMOD0L							
Bit	-	-	PG0DIV1	PG0DIV0	PG0CS1	PG0CS0	-	-	PI7SM	PI6SM	PI5SM	PI4SM	PI3SM	PI2SM	PI1SM	PI0SM
R/W	R	R	R/W	R/W	R/W	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 to 14	-	Reserved bits
13 to 12	PG0DIV1 to PG0DIV0	These bits are used to choose frequency dividing ratio for the sampling clock in the EXI0 to EXI7. 00: No dividing (Initial value) 01: 1/2 of the sampling clock source 10: 1/4 of the sampling clock source 11: 1/8 of the sampling clock source
11 to 10	PG0CS1 to PG0CS0	These bits are used to choose the sampling clock source in the EXI0 to EXI7. 00: LSCLK0 (Initial value) 01: HSCLK 10: HTBCLK1 11: rsvd
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 except in STOP/STOP-D mode 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]

- **If chosen high-speed clock as sampling clock source, it works without sampling when the high-speed clock does not supply; it include stop by entry to standby mode. Set to LSCLK0 as sampling clock if needed.**
- **In the STOP/STOP-D mode, it works without sampling.**

18.2.4 External Interrupt Port Selection Register 0 (EIPSEL0)

This is a SFR used to select a port assigned to EXI0 to EXI7.

Address :	0xF04C(EIPSEL0L/EIPSEL0), 0xF04D(EIPSEL0H),
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	EIPSEL0															
Byte	EIPSEL0H								EIPSEL0L							
Bit	EI7PS1	EI7PS0	EI6PS1	EI6PS0	EI5PS1	EI5PS0	EI4PS1	EI4PS0	EI3PS1	EI3PS0	EI2PS1	EI2PS0	EI1PS1	EI1PS0	EI0PS1	EI0PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
15-0	EInPS1, EInPS0	This bit is used to select a port assigned to EXIn. See Table 18-2 for detail. 00: Selection 0 (Initial value) 01: Selection 1 10: Selection 2 11: Selection 3; it is reserved.

Table 18-2 assignment port to each EXI

EInPS1,EInPS0	EXI7	EXI6	EXI5	EXI4	EXI3	EXI2	EXI1	EXI0
00	P12	P11	P10	P07	P06	P05	P04	P00
01	P33	P32	P31	P30	P73	P27	XT0(PI0)	P72
10	P61*2	P60*2	P53*2	P52*2	P37*2	P36*1	P35*1	P34*1
11	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd

*1: 40/48 pin product only. *2: 48 pin product only. Unavailable selection is as equal as Selection 0.
**rsvd is as equal as Selection 2.

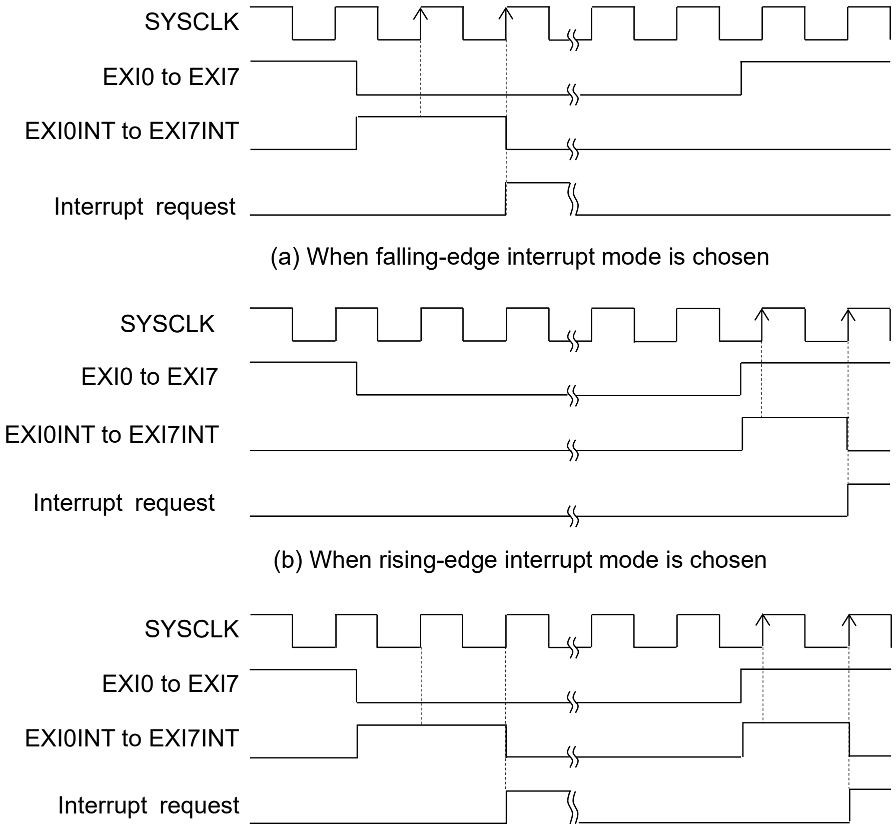
18.3 Description of Operation

18.3.1 Interrupt Request Timing

Figure 18-2 shows the interrupt generation timing without sampling (when the rising-edge/falling-edge/both-edge interrupt mode is chosen). Figure 18-3 shows the interrupt generation timing with sampling (when the rising-edge interrupt mode is chosen).
Table 18-3 shows the difference between the external interrupt generation timings with or without sampling after detection of the edge.

Table 18-3 EXI0INT to EXI11INT Generation After Detection of Edge of EXI0 to EXI7

Sampling	Generation timing
No	Generated in synchronization with SYSCLK
Yes	Generated in synchronization with SYSCLK, when no transition for three periods with sampling clock after detecting edge.



(a) When falling-edge interrupt mode is chosen

(b) When rising-edge interrupt mode is chosen

(c) When both-edge interrupt mode is chosen

Figure 18-2 External Interrupt Generation Timing (without Sampling)

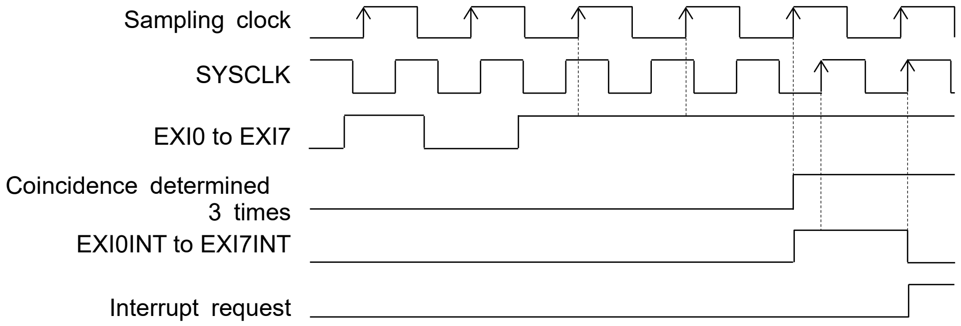
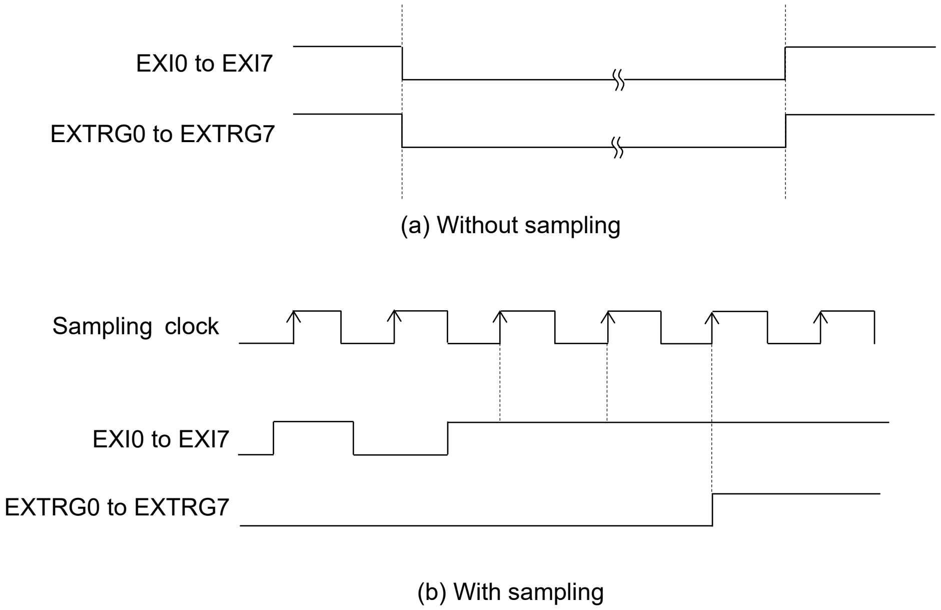


Figure 18-3 External Interrupt Generation Timing (with Sampling, with Rising-edge Interrupt Mode Chosen)

18.3.2 External Trigger Signal

Pins assigned with external interrupt can be used as external trigger signals (EXTRG0 to EXTRG7) for the 16-bit timer and function timer.
In addition, the sampling function contained in the external interrupt function can be used.

Figure 18-4 shows the external trigger signal timing.



(a) Without sampling

(b) With sampling

Figure 18-4 Functional Timer Trigger Signal

18.3.3 External Interrupt Setting Flow

Figure 18-5 shows the external interrupt setting flow.

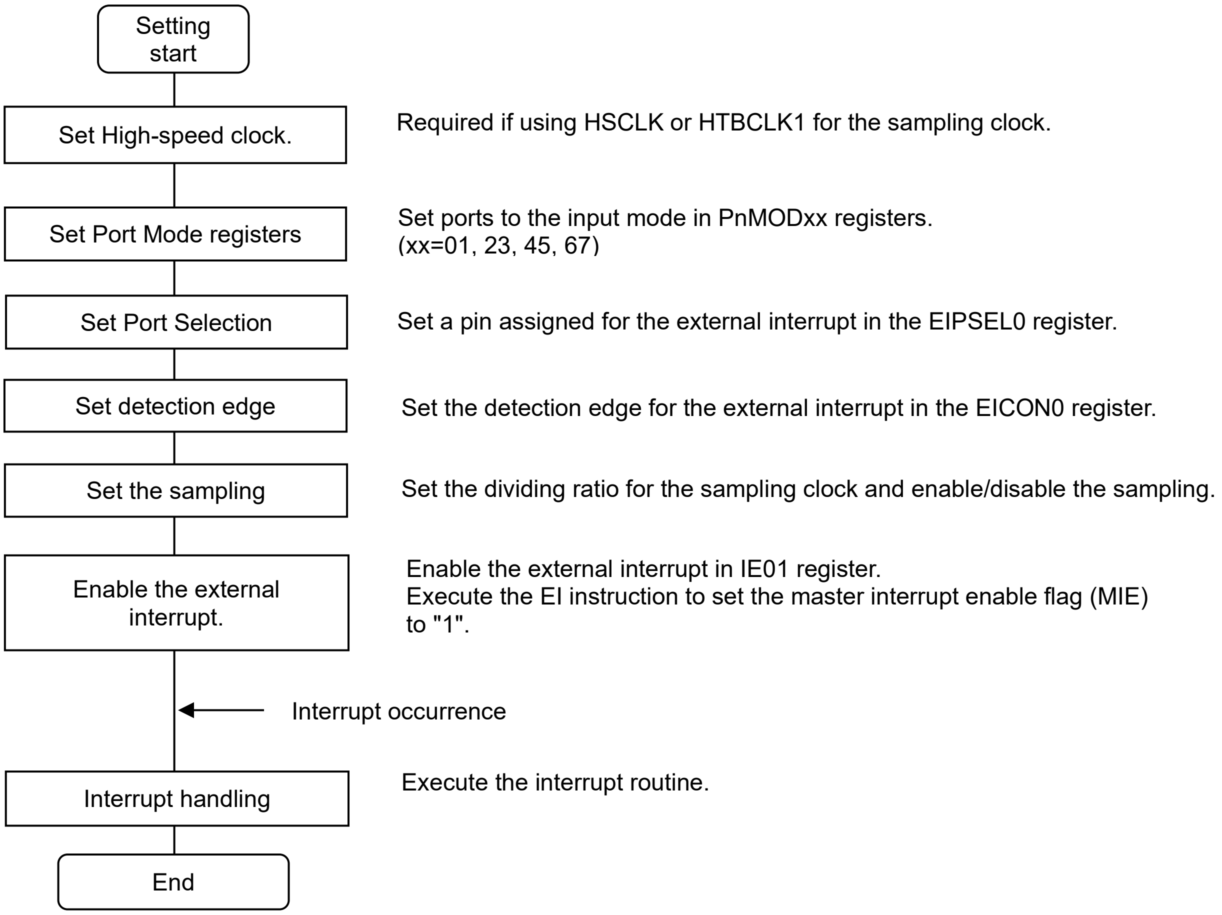

Setting start
Set High-speed clock.
Required if using HSCLK or HTBCLK1 for the sampling clock.
Set Port Mode registers
Set ports to the input mode in PnMODxx registers. (xx=01, 23, 45, 67)
Set Port Selection
Set a pin assigned for the external interrupt in the EIPSEL0 register.
Set detection edge
Set the detection edge for the external interrupt in the EICON0 register.
Set the sampling
Set the dividing ratio for the sampling clock and enable/disable the sampling.
Enable the external interrupt.
Enable the external interrupt in IE01 register. Execute the EI instruction to set the master interrupt enable flag (MIE) to "1".
Interrupt occurrence
Interrupt handling
Execute the interrupt routine.
End

Figure 18-5 External Interrupt Setting Flow

Chapter 19 CRC Calculator

19. CRC Calculator

19.1 General Description

ML62Q2500 groups have the CRC (Cyclic Redundancy Check) calculator that performs CRC calculation and generates the CRC data used for error detection in serial communications.
Also, It has automatic CRC calculation mode to check data in program memory, available in HALT mode or HALT-H mode.

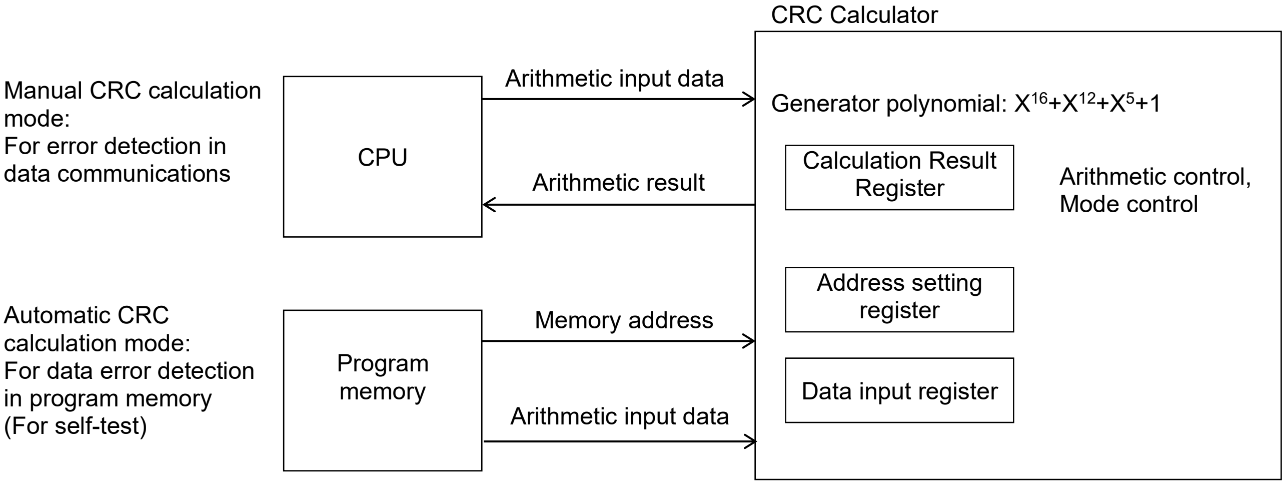

CRC Calculator
Manual CRC calculation mode:
For error detection in data communications
CPU
Arithmetic input data
Arithmetic result
Generator polynomial: $X^{16}+X^{12}+X^5+1$
Calculation Result Register
Arithmetic control, Mode control
Address setting register
Data input register
Automatic CRC calculation mode:
For data error detection in program memory
(For self-test)
Program memory
Memory address
Arithmetic input data

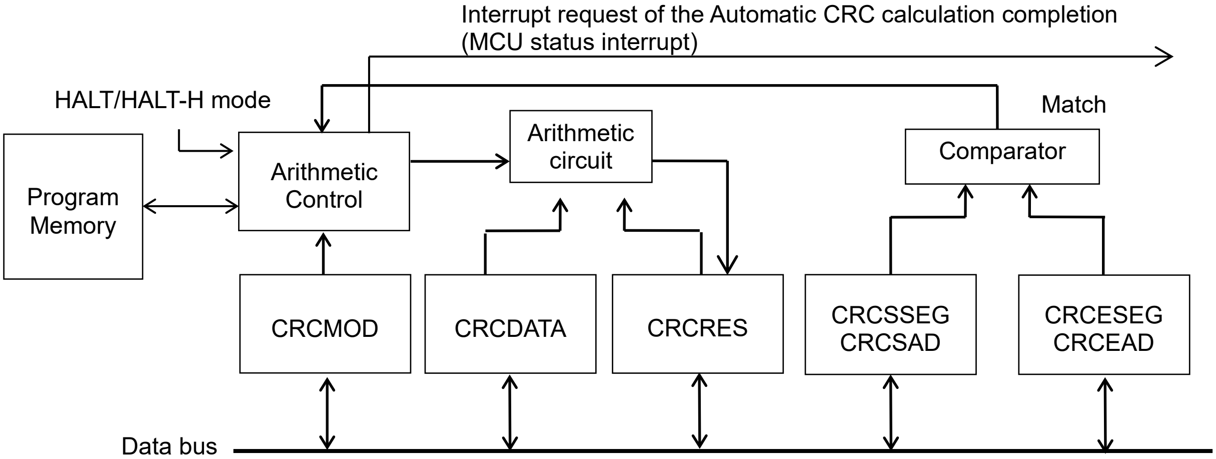
Figure 19-1 CRC calculator overview

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 calculator.



CRCMOD	: CRC Calculation Mode Register	CRCSSEG	: Automatic CRC Calculation Start Segment Setting Register
CRCDATA	: CRC Calculation Data Register	CRCESEG	: Automatic CRC Calculation End Segment Setting Register
CRCRES	: CRC Calculation Result 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 SFR used to set the start address of automatic CRC calculation.
This register is incremented during the automatic CRC calculation mode. This register is writable if the CRCAEN bit of the CRCMOD register is “0”.

Address:	0xF0D0 (CRCSADL/CRCSADL), 0xF0D1 (CRCSADH)
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	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

19.2.3 Automatic CRC Calculation End Address Setting Register (CRCEAD)

CRCEAD is a SFR used to set the end address of automatic CRC calculation.
To write this register is available if the CRCAEN bit of the CRCMOD register is “0” only.

Address:	0xF0D2 (CRCEADL/CRCEAD), 0xF0D3 (CRCEADH)
Access:	R/W
Access size:	8/16 bit
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]

- **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.**
- **If an address set to CRCEAD and CRCESEG is smaller than one of CRCSAD and CRCSSEG, the calculation does not execute. 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 SFR used to set the start segment of automatic CRC calculation.
This register is incremented during the automatic CRC calculation mode. This register is writable if the CRCAEN bit of the CRCMOD register is “0”.

Address:	0xF0D4 (CRCSSEG)
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	-								CRCSSEG							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	CRCS SEG3	CRCS SEG2	CRCS SEG1	CRCS SEG0
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

19.2.5 Automatic CRC Calculation End Segment Setting Register (CRCESEG)

CRCESEG is a SFR used to set the end segment of automatic CRC calculation.
To write this register is available if the CRCAEN bit of the CRCMOD register is “0” only.

Address:	0xF0D6 (CRCESEG)
Access:	R/W
Access size:	8 bit
Initial value:	0xFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								CRCESEG							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	CRCE SEG3	CRCE SEG2	CRCE SEG1	CRCE SEG0
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]

- **If an address set to CRCEAD and CRCESEG is smaller than one of CRCSAD and CRCSSEG, the calculation does not execute. 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 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). This register is writable if the CRCAEN bit of the CRCMOD register is "0".

Address:	0xF0D8 (CRCDATA)
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	-								CRCDATA							
Bit	-	-	-	-	-	-	-	-	CRCD ATA7	CRCD ATA6	CRCD ATA5	CRCD ATA4	CRCD ATA3	CRCD ATA2	CRCD ATA1	CRCD ATA0
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

19.2.7 CRC Calculation Result Register (CRCRES)

CRCRES is a SFR. The CRC calculation result is stored by the hardware.
Set data to the CRCRES as an initial data for the CRC calculation. To write this register is available if the CRCAEN bit of the CRCMOD register is "0" only.

Address:	0xF0DA (CRCRESL/CRCRES), 0xF0DB (CRCRESH)
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	CRCRES															
Byte	CRCRESH								CRCRESL							
Bit	CRCR ES15	CRCR ES14	CRCR ES13	CRCR ES12	CRCR ES11	CRCR ES10	CRCR ES9	CRCR ES8	CRCR ES7	CRCR ES6	CRCR ES5	CRCR ES4	CRCR ES3	CRCR ES2	CRCR ES1	CRCR ES0
R/W	R/W	R/W	R/W	R/W	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

19.2.8 CRC Calculation Mode Register (CRCMOD)

CRCMOD is SFR used to control the CRC calculation mode.

Address:	0xF0DC (CRCMOD)
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	-								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: Disable (Initial value) 1: Enable

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.
Transmission and CRC calculation data: 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x81, 0x7F

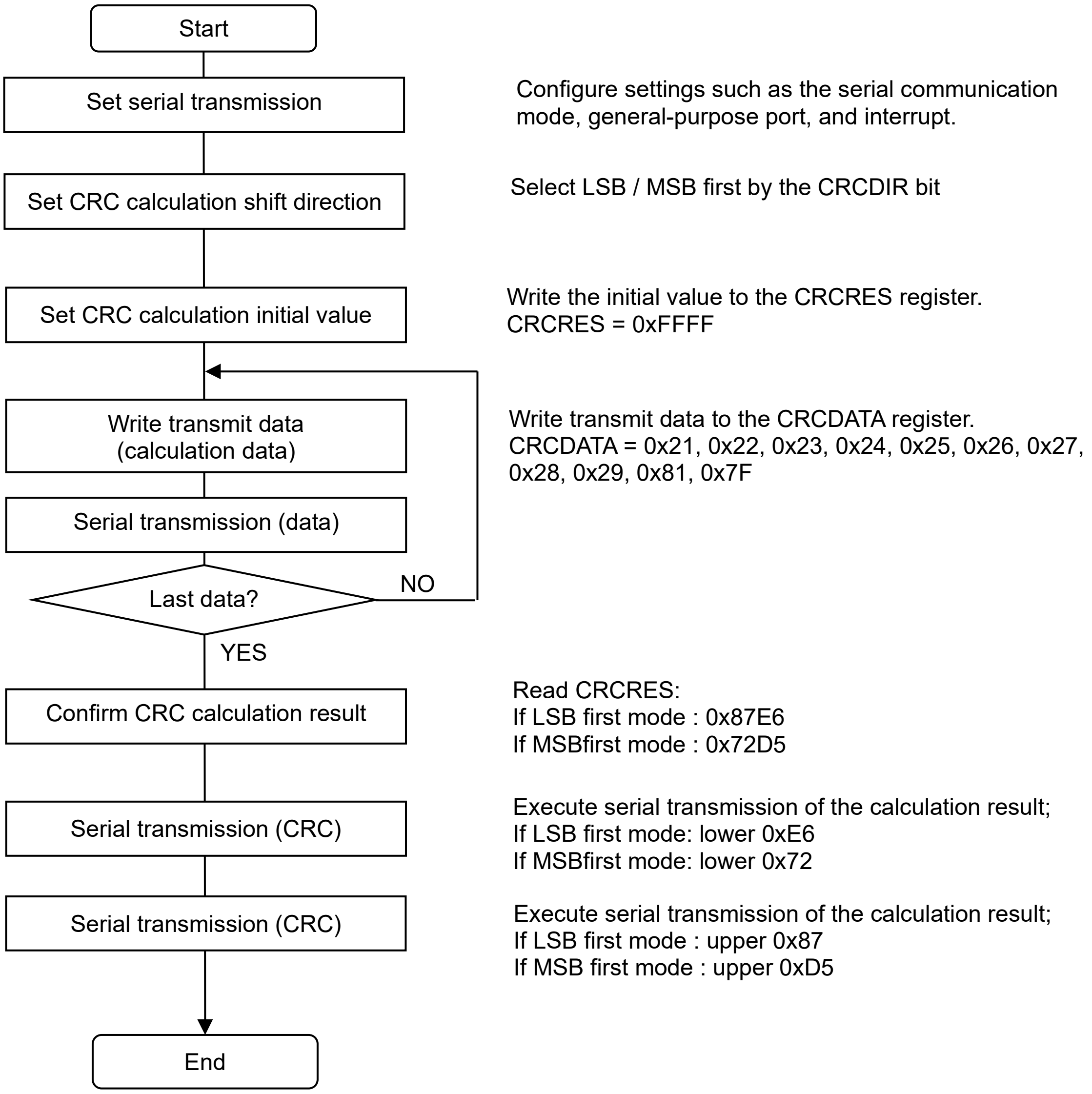

Start
Set serial transmission
Configure settings such as the serial communication mode, general-purpose port, and interrupt.
Set CRC calculation shift direction
Select LSB / MSB first by the CRCDIR bit
Set CRC calculation initial value
Write the initial value to the CRCRES register.
CRCRES = 0xFFFF
Write transmit data (calculation data)
Write transmit data to the CRCDATA register.
CRCDATA = 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x81, 0x7F
Serial transmission (data)
Last data?
NO
YES
Confirm CRC calculation result
Read CRCRES:
If LSB first mode : 0x87E6
If MSBfirst mode : 0x72D5
Serial transmission (CRC)
Execute serial transmission of the calculation result;
If LSB first mode: lower 0xE6
If MSBfirst mode: lower 0x72
Serial transmission (CRC)
Execute serial transmission of the calculation result;
If LSB first mode : upper 0x87
If MSB first mode : upper 0xD5
End

Figure 19-3 CRC Calculation Processing Flow 1 (Serial Transmission)

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 received data with 0x21 in the beginning is used as calculation data. The first 11 bytes of the CRC calculation result is added to the last two bytes.

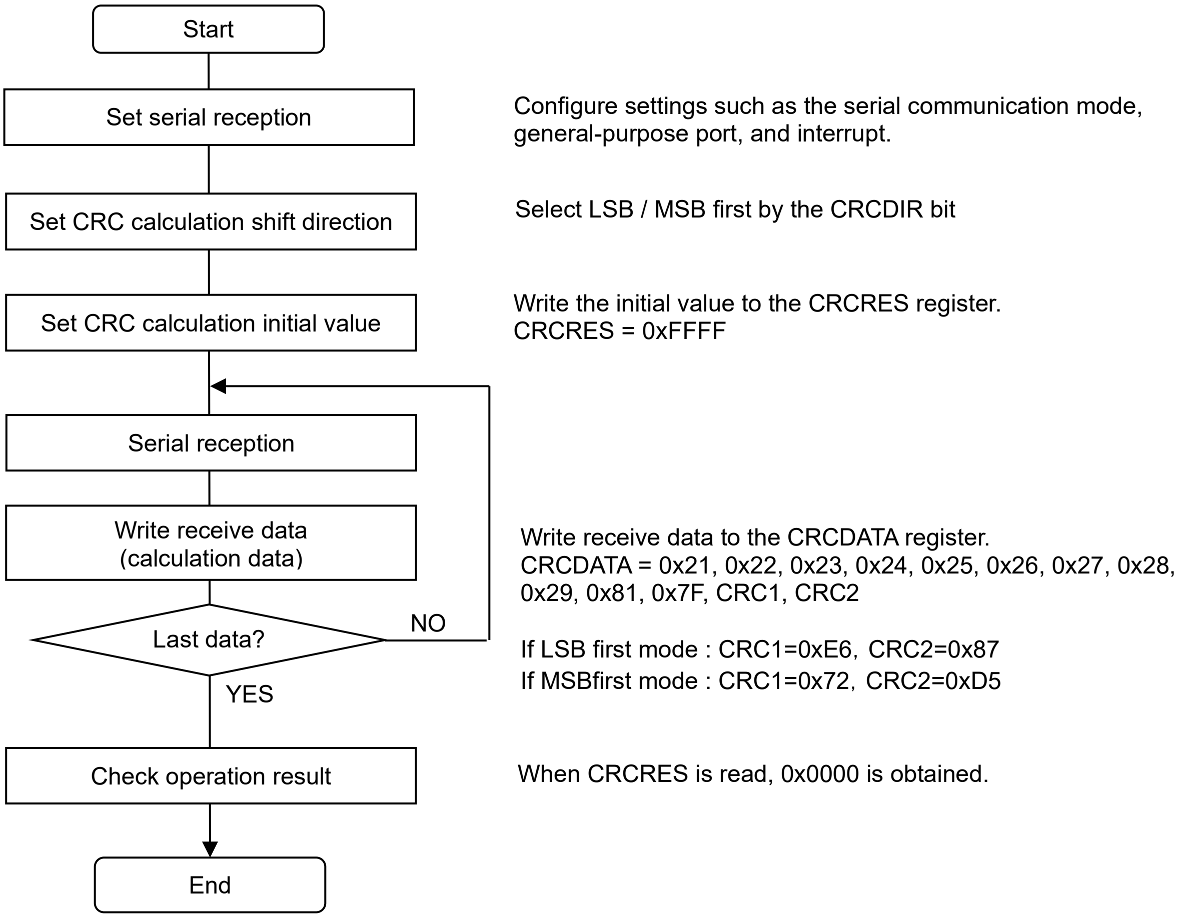
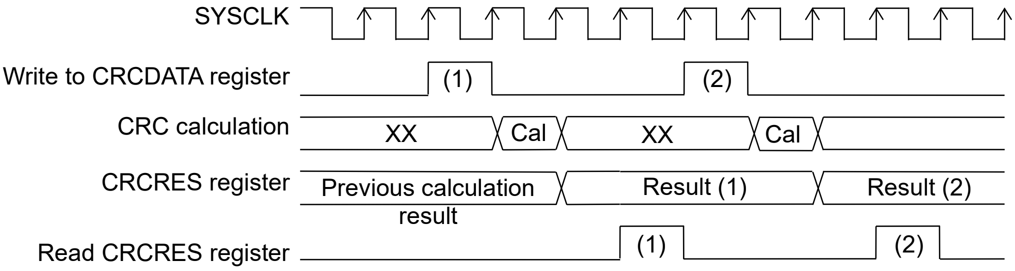

Start
Set serial reception
Configure settings such as the serial communication mode, general-purpose port, and interrupt.
Set CRC calculation shift direction
Select LSB / MSB first by the CRCDIR bit
Set CRC calculation initial value
Write the initial value to the CRCRES register.
CRCRES = 0xFFFF
Serial reception
Write receive data (calculation data)
Write receive data to the CRCDATA register.
CRCDATA = 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x81, 0x7F, CRC1, CRC2
Last data?
NO
YES
If LSB first mode : CRC1=0xE6, CRC2=0x87
If MSBfirst mode : CRC1=0x72, CRC2=0xD5
Check operation result
When CRCRES is read, 0x0000 is obtained.
End

Figure 19-4 CRC Calculation Processing Flow 2 (Serial Reception/LSB 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 19-5 shows the operation timing chart of CRC calculation.



"Cal" means "Calculation state"

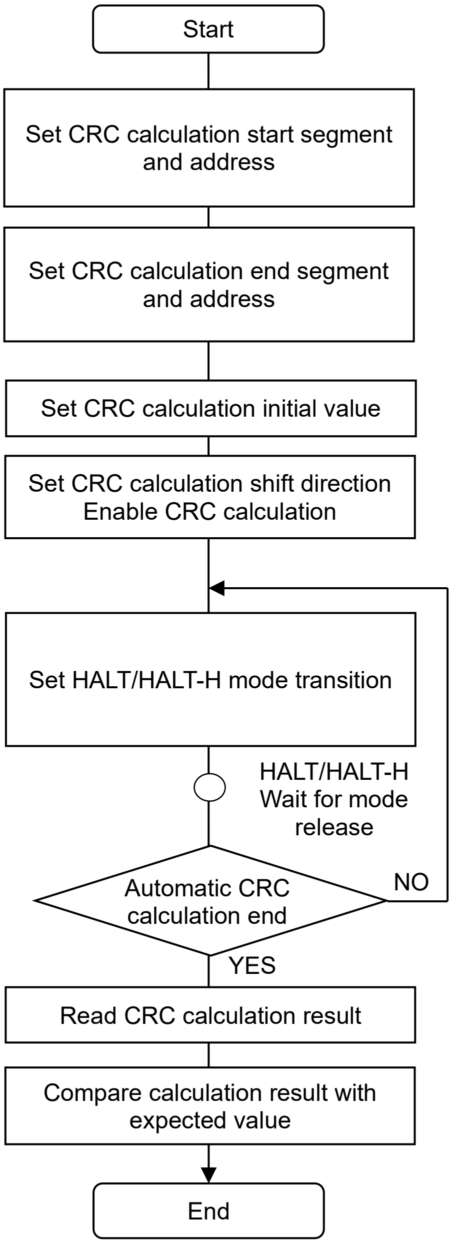
Figure 19-5 Timing Chart of CRC Calculation.

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). The calculation is four-byte length. Generate an expected value by four bytes.
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.

19.3.2.1 Example of Use of Automatic CRC Calculation Mode

The following chart shows the automatic CRC calculation process flow.


Start
Set CRC calculation start segment and address
Set CRC calculation end segment and address
Set CRC calculation initial value
Set CRC calculation shift direction
Enable CRC calculation
Set HALT/HALT-H mode transition
HALT/HALT-H
Wait for mode release
Automatic CRC calculation end
NO
YES
Read CRC calculation result
Compare calculation result with expected value
End

Set the values of address and segment from which the CRC calculation is started within the program code area to the CRCSAD and CRCSSEG registers.

Set the values of address and segment in which the CRC calculation is ended to the CRCEAD and CRCESEG registers.

Write the initial value to the CRCRES register.
CRCRES = 0xFFFF

Write "0x01" to the CRCMOD register to set the CRC calculation shift direction to LSB first and enable the automatic CRC calculation mode.

Shift the operation state from the program run mode to the HALT/HALT-H mode.

When a WDT interrupt or other interrupts enabled in the interrupt enable registers (IE0 to IE7) is generated, the operation returns from the HALT/HALT-H mode to the program run mode.

After the HALT/HALT-H mode is released, read CRCMOD or MCI1S bit of the MCISTATL register to confirm that the automatic CRC calculation is ended.

Read the CRC calculation result from the CRCRES register.

Compare the CRC calculation result and expected value.

Figure 19-6 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 "ML62Q2000 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.**

Chapter 22 Voltage Level Supervisor

22. Voltage Level Supervisor

22.1 General Description

ML62Q2500 group 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

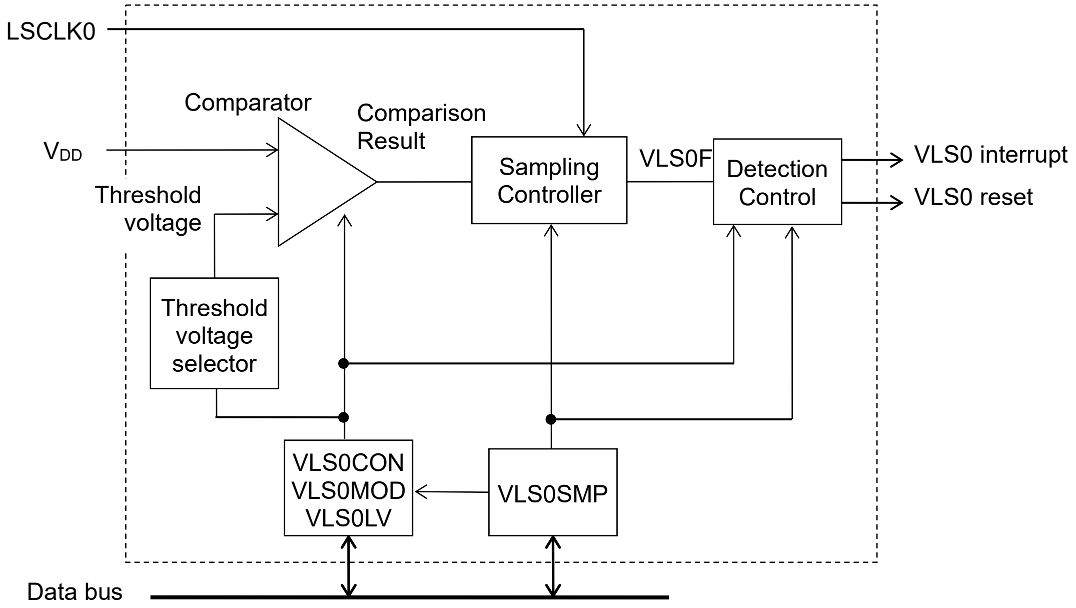
- Accuracy: ±4 %
- Threshold voltage: Selectable from 15 values (1.85 to 4.00 V)
- Operation mode: Supervisor mode (continuous detection) or single mode (one detection)

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

Figure 22-1 shows the configuration of the VLS.



VLS0CON : Voltage level supervisor 0 control register
VLS0MOD : Voltage level supervisor 0 mode register
VLS0LV : Voltage level supervisor 0 level register
VLS0SMP : Voltage level supervisor 0 sampling register

Figure 22-1 Configuration of Voltage Level Supervisor

22.2 Description of Registers

22.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF890	Voltage level supervisor 0 control register	VLS0CON	-	R/W	8	0x00
0xF891	Reserved	-	-	-	-	-
0xF892	Voltage level supervisor 0 mode register	VLS0MOD	-	R/W	8	0x00
0xF893	Reserved	-	-	-	-	-
0xF894	Voltage level supervisor 0 level register	VLS0LV	-	R/W	8	0x0E
0xF895	Reserved	-	-	-	-	-
0xF896	Voltage level supervisor 0 sampling register	VLS0SMP	-	R/W	8	0x00
0xF897	Reserved	-	-	-	-	-

22.2.2 Voltage Level Supervisor 0 Control Register (VLS0CON)

This is a SFR used to control the VLS0 (Voltage Level Supervisor).
This is unresetable by anything other than the Power On Reset(POR) and RESET_N pin reset.

Address:	0xF890 (VLS0CON)
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	-								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 VLS 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 VLS operation. In the single mode, this bit is automatically reset to "0" after detecting the voltage level and the VLS stops operating. 0: Disable operating the VLS (Initial value) 1: Enable operating the VLS

[Note]

- **There is a limitation in each mode for entering the STOP/STOP-D mode while the VLS is running.**

Operation	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)

This is a SFR used to control the operation mode of the VLS (Voltage Level Supervisor).
Set this register only when the VLS is stopped (VLS0EN bit of VLS0CON register is "0").
This register is unresetable by anything other than the Power On Reset (POR) and RESET_N pin reset.

Address:	0xF892 (VLS0MOD)
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	-								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 VLS 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.**

Operation	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)

This is a 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 RESET_N pin reset.

Address:	0xF894 (VLS0LV)
Access:	R/W
Access size:	8 bit
Initial value:	0x0E

	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	1	1	1	0

Bit No.	Bit symbol name	Description
7 to 4	-	Reserved bits
3 to 0	VLS0LV3 to VLS0LV0	These bits are used to choose the threshold voltage (V_{VLSF} / V_{VLSR}) of VLS detected while the V_{DD} is falling or rising. The VLS 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: 3.99V ± 4% 0001: 3.68V ± 4% 0010: 3.05V ± 4% 0011: 2.96V ± 4% 0100: 2.84V ± 4% 0101: 2.76V ± 4% 0110: 2.66V ± 4% 0111: 2.54V ± 4% 1000: 2.45V ± 4% 1001: 2.35V ± 4% 1010: 2.24V ± 4% 1011: 2.16V ± 4% 1100: 2.05V ± 4% 1101: 1.95V ± 4% 1110: 1.85V ± 4% (Initial value) 1111: Do not use (1.85V±4%) where these value is V_{VLSF}.

22.2.5 Voltage Level Supervisor 0 Sampling Register (VLS0SMP)

This is a 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 RESET_N pin reset.

Address:	0xF896 (VLS0SMP)
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	-								VLS0SMP							
Bit	-	-	-	-	-	-	-	-	-	rsvd	VLS0DIV1	VLS0DIV0	VLS0SM1	rsvd	-	-
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	rsvd	Reserved bit
5 to 4	VLS0DIV1 to VLS0DIV0	These bits are used to choose frequency dividing ratio for the sampling clock. 00: No dividing (Initial value) 01:divided 2 10: divided 4 11: divided 8
3	VLS0SM1	This is used to choose the sampling clock source for detecting the voltage level. 0: No sampling (Initial value) 1: LSCLK0
2	rsvd	Reserved bit. Set "0" to this bit.
1, 0	-	Reserved bits

[Note]

- **In the STOP/STOP-D mode, the VLS works without sampling regardless the setting in VLS0SM1 bit.**

22.3 Description of Operation

VLS 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. VLS 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 VLS:

– Supervisor mode:

Operation	“1" is written to VLS0EN to enable operation of VLS, 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.

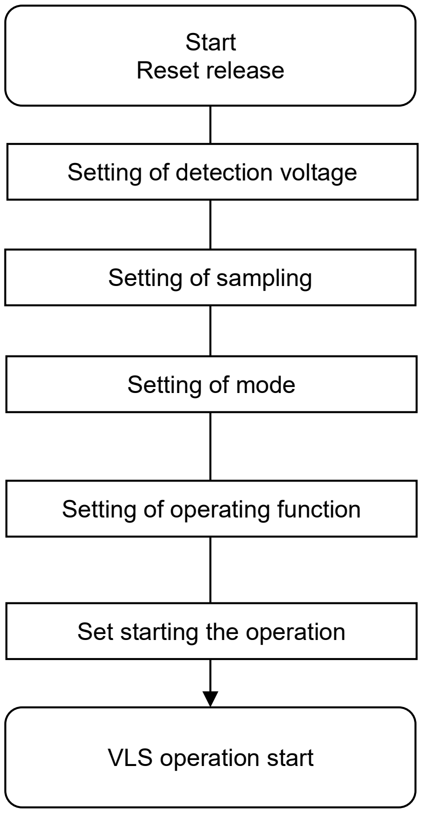
– Single mode:

Operation	"1" is written to VLS0EN to enable operation of VLS, 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.


Start
Reset release
Setting of detection voltage
Setting of sampling
Setting of mode
Setting of operating function
Set starting the operation
VLS operation start

Specify VLS0LV3 to VLS0LV0 bits of VLS0LV register.

Specify VLS0SM1 and VLS0DIV1,0 bits of VLS0SMP register.

Write "0x2" or "0x3" to VLS0AMD1 and VLS0AMD0 bits of VLS0MOD register to choose the supervisor mode.

Specify VLS0SEL1 and VLS0SEL0 bits of VLS0MOD register.

Set VLS0EN bit of VLS0CON register to "1".

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.

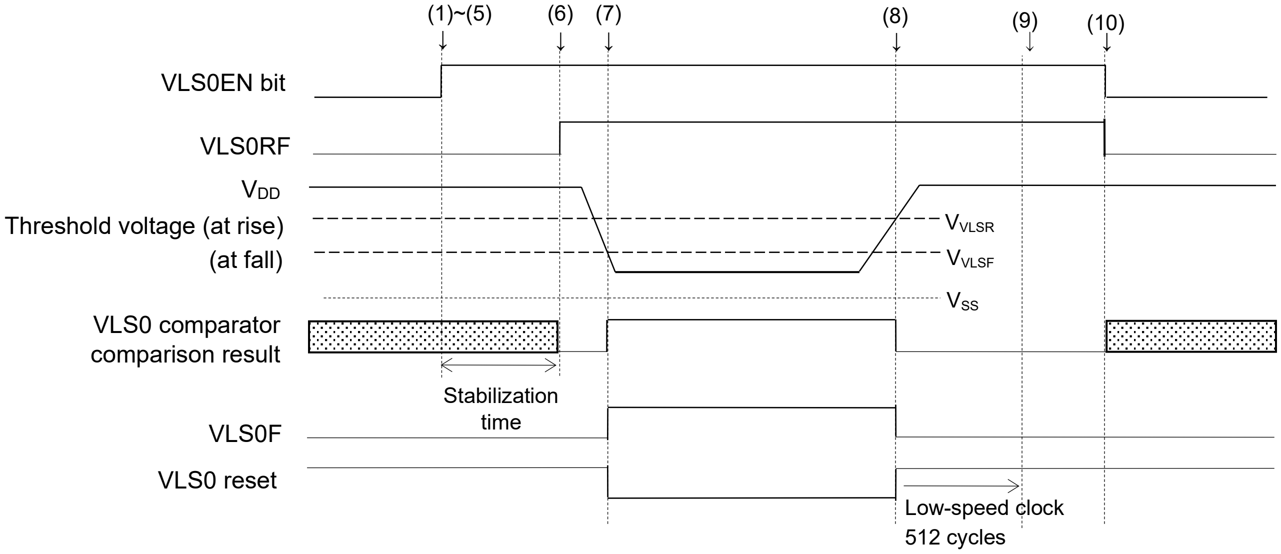


Figure 22-3 Operation Timing Chart When the VLS0 Reset Output without Sampling is specified

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 VLS0SM1bit 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.

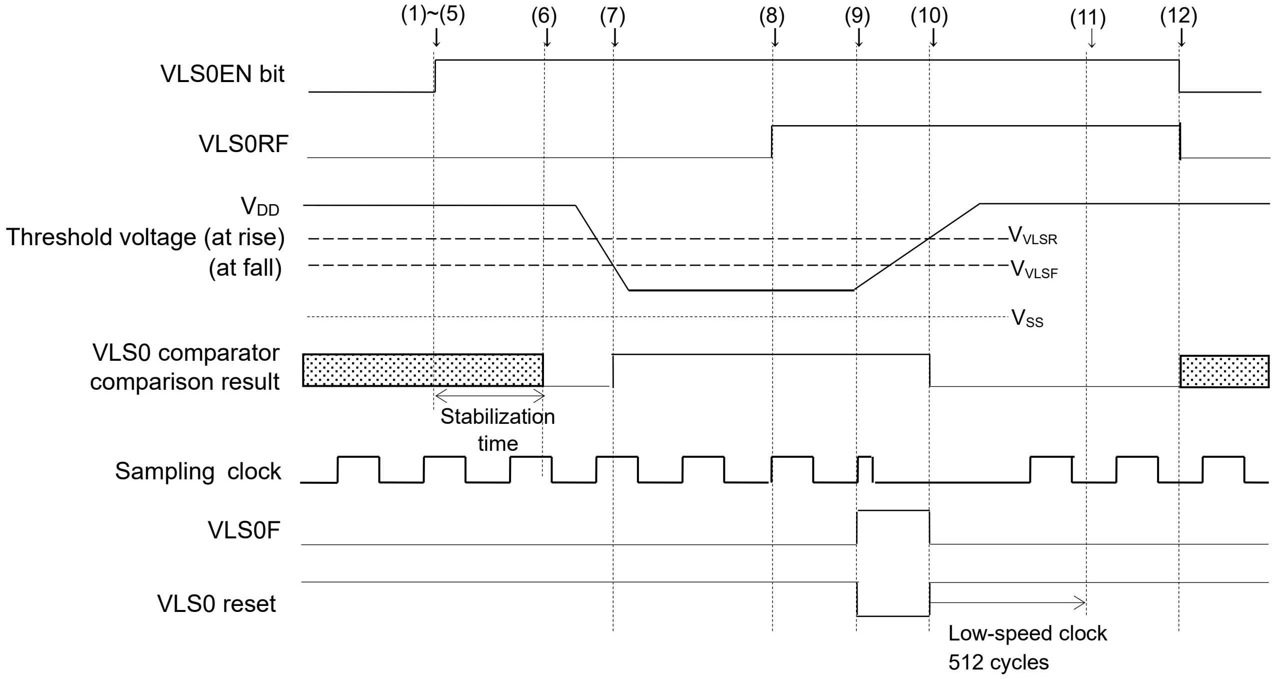


Figure 22-4 Operation Timing Chart When the VLS0 Reset Output with Sampling is specified

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 LSCLK0" by the VLS0SM1 bit of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV1 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.
Write “0x1” to VLS0SEL1 and VLS0SEL0 bits of the VLS0MOD register in order to enable the VLS0 reset.
(4) Write "1" to the VLS0EN bit to enable VLS operation.
(5) Wait until the comparison result of the VLS comparator is stabilized (approx. 300 μs).
(6) V_{DD} becomes below the threshold voltage (V_{VLSF}).
(7) Once the comparison result of the VLS comparator is stabilized, the VLS0RF bit is set to "1" after three cycles of the sampling clock.
(8) If the comparison result of the VLS 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.
(9) If the comparison result of the VLS comparator becomes equal to or above the threshold voltage (V_{VLSR}), the VLS0F bit is cleared to "0" to release the VLS0 reset.
(10) The CPU starts after 512 cycles of low-speed clock. The VLS does not operate while the sampling clock is stops.
(11) Write "0" to the VLS0EN bit to disable VLS operation.

[Note]

- **Entering the STOP/STOP-D mode is not allowed during the VLS 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 VLS detection voltage is 1.85V, so the MCU becomes in reset mode when the V_{DD} is 1.85V 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 VLS 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.

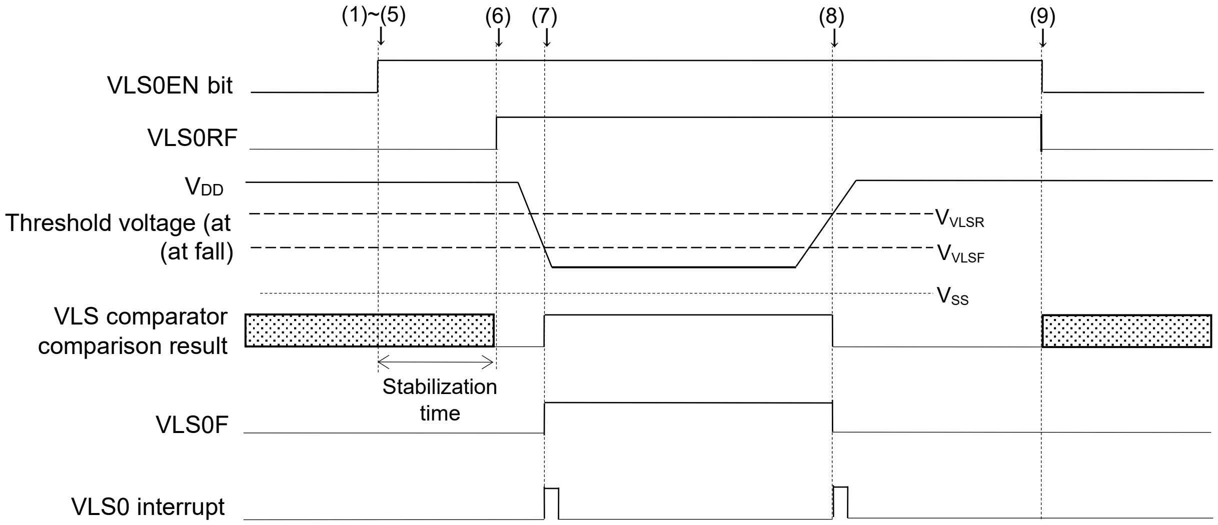


Figure 22-5 Operation Timing Chart When the VLS0 Interrupt Output without Sampling is specified

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 bit 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 VLS operation.
(6) When the comparison result of the VLS 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 VLS operation.

Figure 22-6 shows an example of the operation timing chart when the VLS0 interrupt output with sampling is specified.

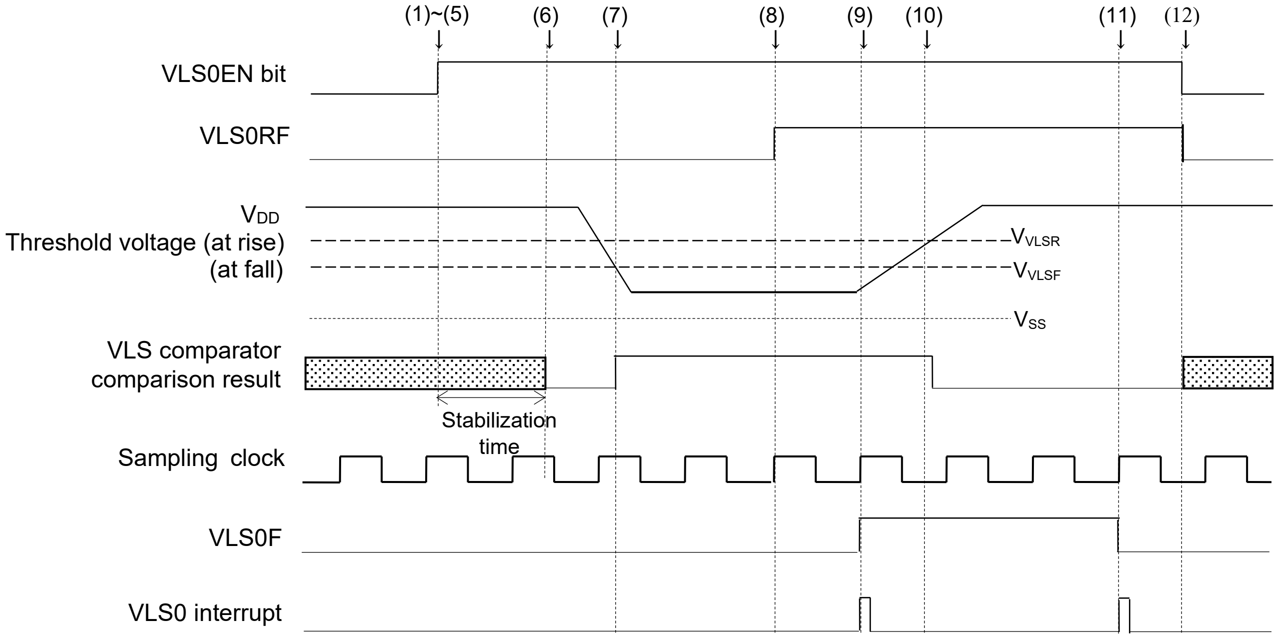


Figure 22-6 Operation Timing Chart When the VLS0 Interrupt Output with Sampling is specified

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 LSCLK0" by the VLS0SM1 bit of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV1 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 VLS operation.
(6) Wait until the comparison result of the VLS comparator is stabilized (approx. 300 µs).
(7) V_{DD} becomes below the threshold voltage (V_{VLSF}).
(8) Once the comparison result of the VLS comparator is stabilized, the VLS0RF bit is set to "1" after three cycles of the sampling clock.
(9) If the comparison result of the VLS 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 VLS comparator becomes equal to or above the threshold voltage (V_{VLSR}).
(11) If the comparison result of the VLS 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 VLS operation.

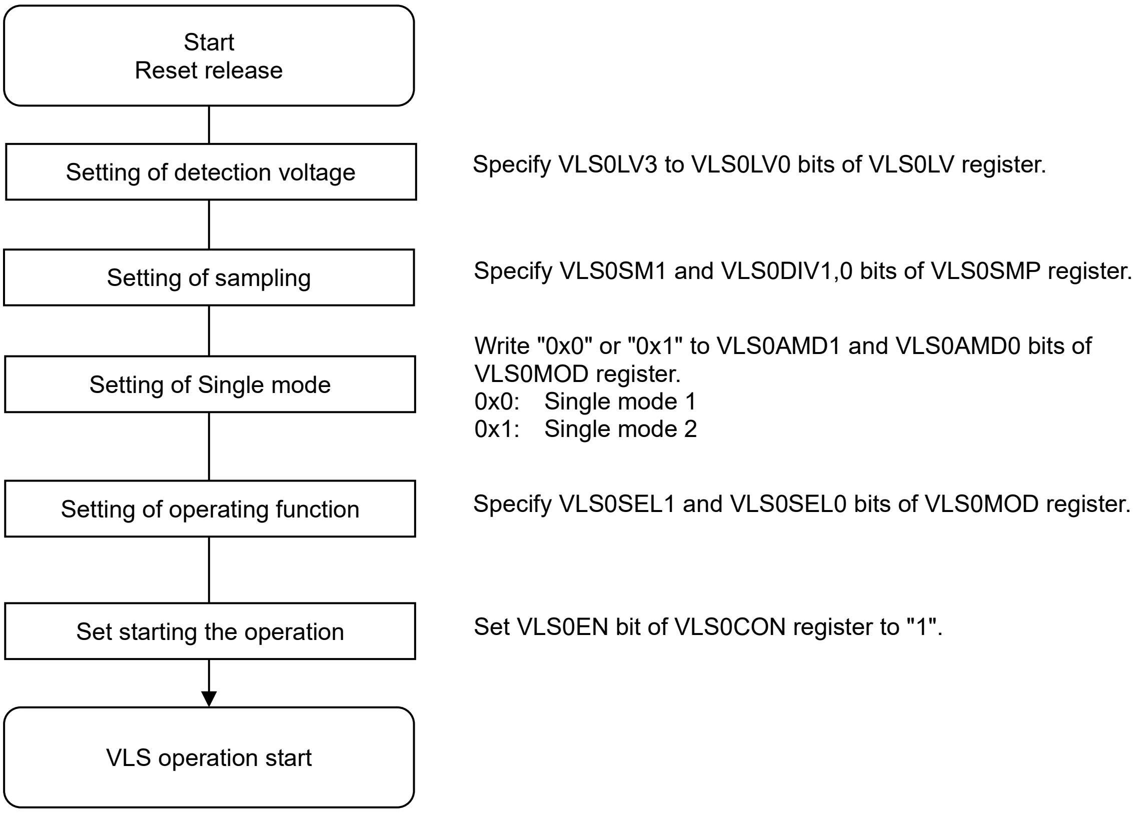
[Note]

- **Entering the STOP/STOP-D mode is not allowed during the VLS 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.


Start
Reset release
Setting of detection voltage
Setting of sampling
Setting of Single mode
Setting of operating function
Set starting the operation
VLS operation start

Specify VLS0LV3 to VLS0LV0 bits of VLS0LV register.

Specify VLS0SM1 and VLS0DIV1,0 bits of VLS0SMP register.

Write "0x0" or "0x1" to VLS0AMD1 and VLS0AMD0 bits of VLS0MOD register.
0x0: Single mode 1
0x1: Single mode 2

Specify VLS0SEL1 and VLS0SEL0 bits of VLS0MOD register.

Set VLS0EN bit of VLS0CON register to "1".

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.

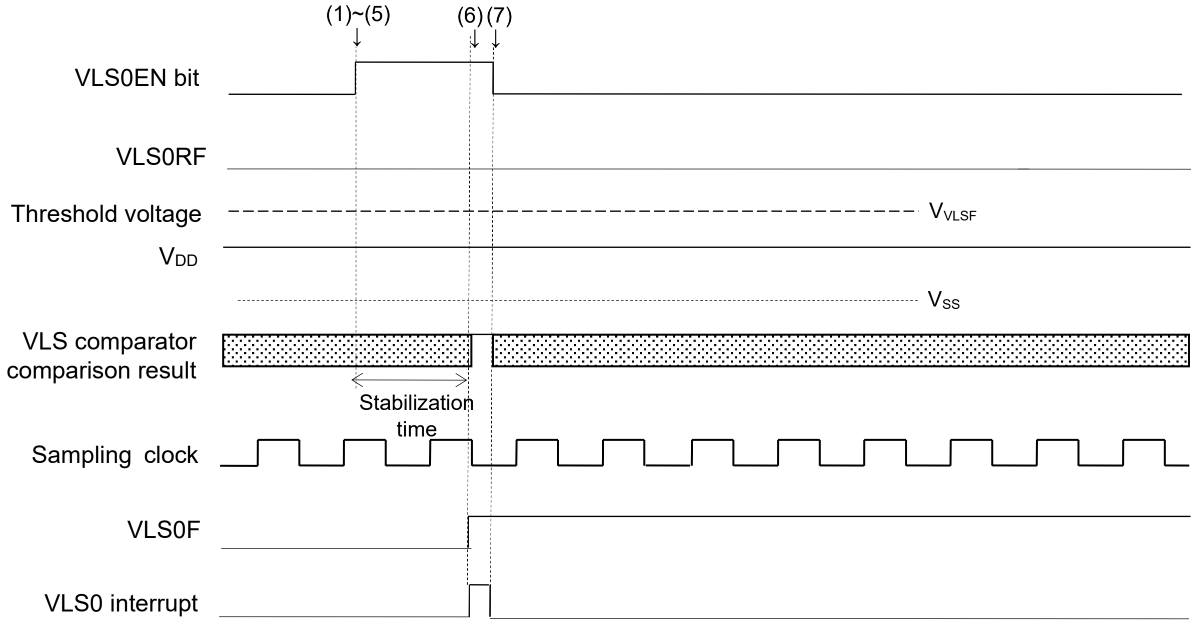


Figure 22-8 Operation Timing Chart without Sampling (Single Mode 1)

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 bit 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 VLS operation.
(6) If V_{DD} is below the threshold voltage (V_{VLSF}) when the comparison result of the VLS 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 VLS 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.

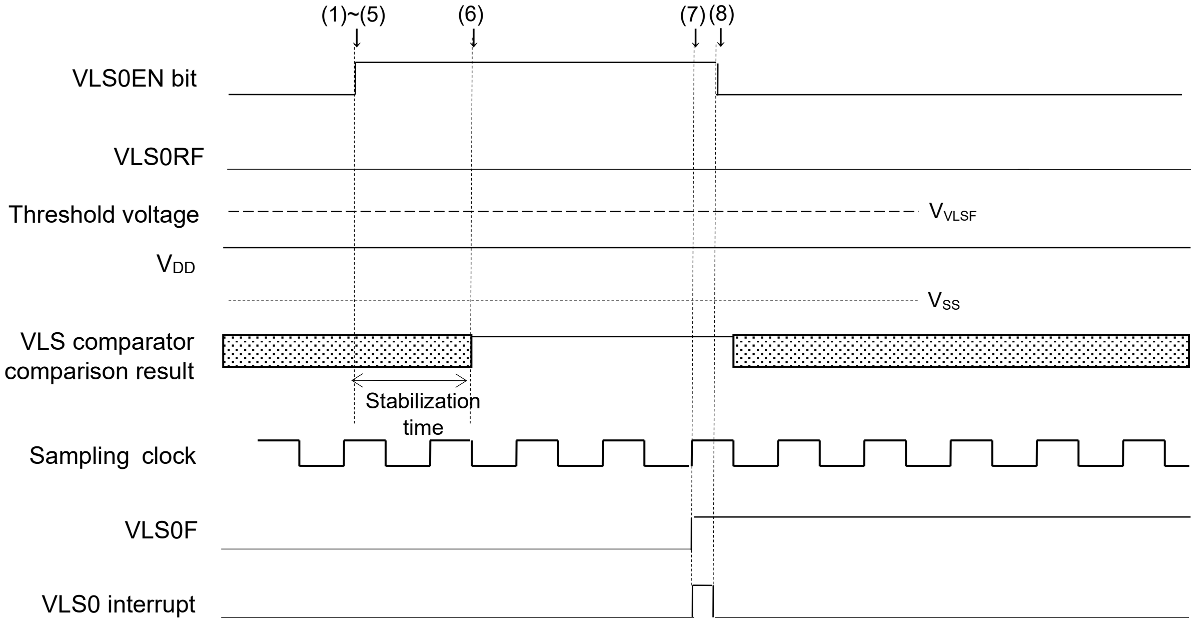


Figure 22-9 Operation Timing Chart with Sampling (Single Mode 1)

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 LSCLK0" by the VLS0SM1 bit of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV1 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 VLS operation.
(6) Wait until the comparison result of the VLS 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 VLS 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.

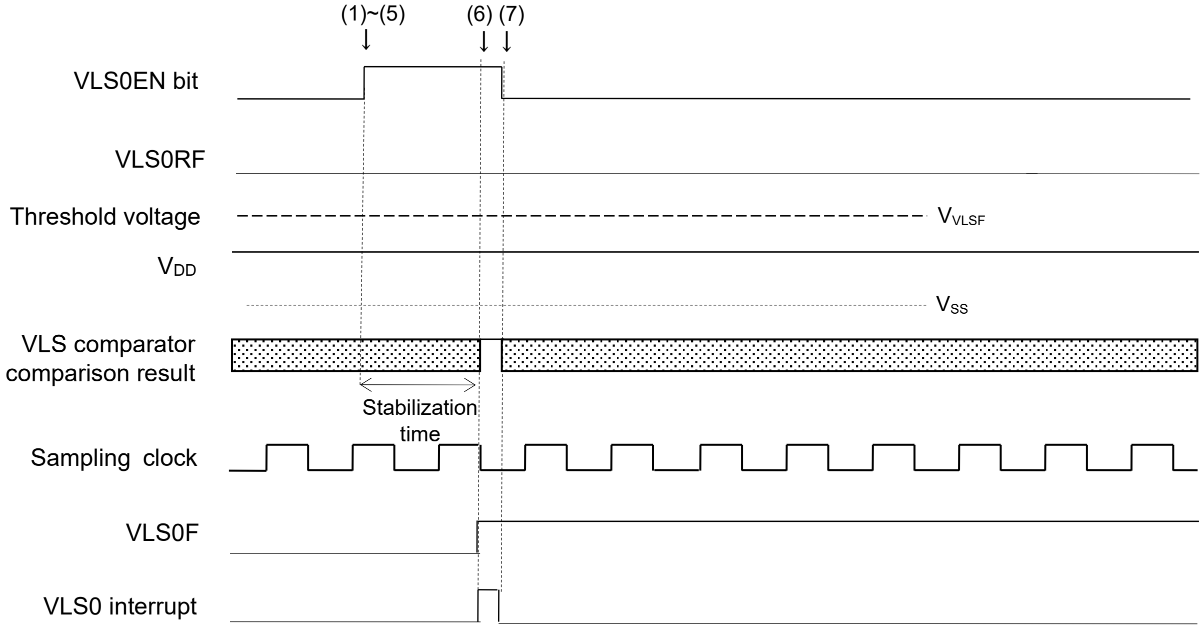


Figure 22-10 Operation Timing Chart without Sampling (Single Mode 2)

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 bit 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 VLS.
(6) If V_{DD} is below the specified threshold voltage (V_{VLSF}) when the comparison result of the VLS 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 VLS 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.

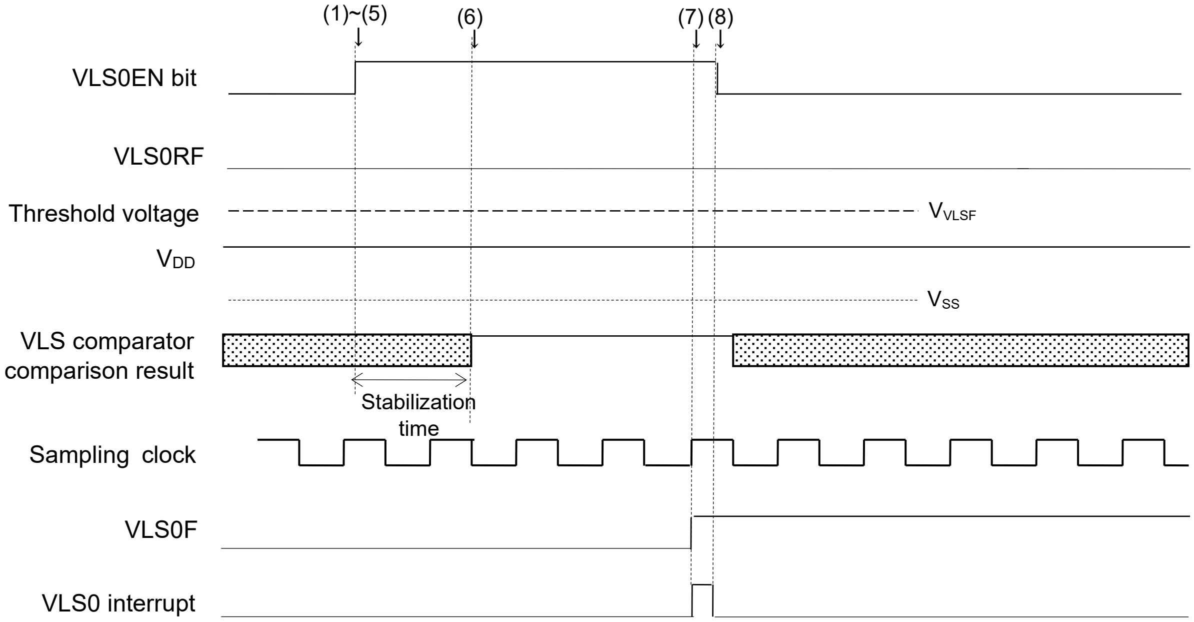


Figure 22-11 Operation Timing Chart with Sampling (Single Mode 2)

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 LSCLK" by the VLS0SM1 bit of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV1 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 VLS operation.
(6) Wait until the comparison result of the VLS 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 VLS 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

ML62Q2500 group 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.	ML62Q2500 group
0	●
1	●
2	●
3	●
4	●
5	●
6	●
7	●
8	●
9	●
10	●
11	●
12	●
13	●

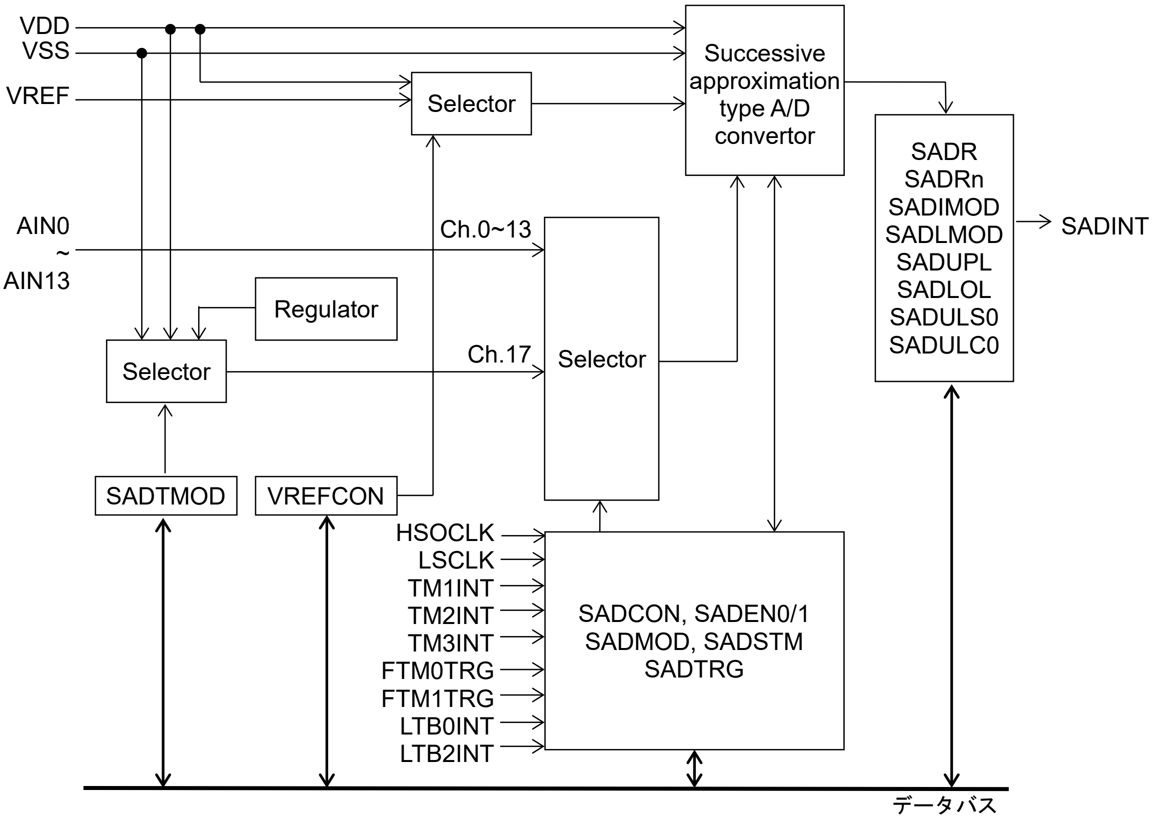
●: Available - : Unavailable

23.1.1 Features

- Resolution : 12bit
- Conversion time : Min. 1.375μs/channel (conversion clock is 16MHz)
- Number of input channel : Max. 14ch
- Reference voltage: Voltage input from the VDD pin or External reference voltage(VREF 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 interrupt request (TM1INT, TM2INT, TM3INT)
  – Functional Timer trigger (FTM0TRG, FTM1TRG)
  – Low-speed Time Base Counter interrupt (LTB0INT, LTB2INT)

23.1.2 Configuration

Figure 23-1 shows the configuration of SA-ADC.



SADCON	: SA-ADC control register
SADEN0/1	: SA-ADC enable register 0, 1
SADMOD	: SA-ADC mode register
SADSTM	: SA-ADC conversion interval register
SADR	: SA-ADC result register
SADRn	: SA-ADC result register n (n=0 to 13)
SADIMOD	: SA-ADC interrupt mode register
SADLMOD	: SA-ADC upper/lower limit mode register
SADUPL	: SA-ADC upper limit setting register
SADLOL	: SA-ADC lower limit setting register
SADULS0	: SA-ADC upper/lower limit status register 0
SADULC0	: SA-ADC upper/lower limit status clear register 0
VREFCON	: Reference voltage control register
SADTRG	: SA-ADC trigger register
SADTMOD	: SA-ADC test mode register
SADINT	: SA-ADC interrupt request
TM1INT,TM2INT,TM3INT	: 16-bit timer 1,2,3 interrupt
FTM0TRG,FTM1TRG	: Functional timer 0, 1 trigger
LTB0INT	: Low speed time base counter 0 interrupt
LTB2INT	: Low speed time base counter 2 interrupt

Figure 23-1 Configuration of successive approximation type A/D Converter

23.1.3 List of Pins

The I/O pins of the Successive Approximation type A/D converter are assigned to the shared function of the general ports.

Pin name	I/O	Description
VDD	-	Positive power supply for SA-ADC
VSS	-	Negative power supply for SA-ADC
VREF	-	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

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	ML62Q2500 group
0	AIN0	P20	Primary Func.	P2MOD0	0000_0000	●
1	AIN1	P21	Primary Func.	P2MOD1	0000_0000	●
2	AIN2	P22	Primary Func.	P2MOD2	0000_0000	●
3	AIN3	P23	Primary Func.	P2MOD3	0000_0000	●
4	AIN4	P24	Primary Func.	P2MOD4	0000_0000	●
5	AIN5	P25	Primary Func.	P2MOD5	0000_0000	●
6	AIN6	P26	Primary Func.	P2MOD6	0000_0000	●
7	AIN7	P27	Primary Func.	P2MOD7	0000_0000	●
8	AIN8	P31	Primary Func.	P3MOD1	0000_0000	●
9	AIN9	P32	Primary Func.	P3MOD2	0000_0000	●
10	AIN10	P33	Primary Func.	P3MOD3	0000_0000	●
11	AIN11	P70	Primary Func.	P7MOD0	0000_0000	●
12	AIN12	P71	Primary Func.	P7MOD1	0000_0000	●
13	AIN13	P13	Primary Func.	P1MOD3	0000_0000	●
-	VREF	P30	Primary Func.	P3MOD0	0000_0000	●

[Note]

- **When using the SA-ADC, set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 1, 2, 3, 7, 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

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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF800	SA-ADC mode register	SADMODL	SADMOD	R/W	8/16	0x00
0xF801		SADMODH		R/W	8	0x00
0xF802	SA-ADC control register	SADCONL	SADCON	R/W	8/16	0x00
0xF803		SADCONH		R/W	8	0x00
0xF804	SA-ADC conversion interval register	SADSTML	SADSTM	R/W	8/16	0x00
0xF805		SADSTMH		R/W	8	0x00
0xF806	Reference voltage control register	VREFCON	-	R/W	8	0x00
0xF807	Reserved	-	-	-	-	-
0xF808	SA-ADC interrupt mode register	SADIMOD	-	R/W	8	0x00
0xF809	Reserved	-	-	-	-	-
0xF80A	SA-ADC trigger register	SADTRG	-	R/W	8	0x00
0xF80B	Reserved	-	-	-	-	-
0xF80C	SA-ADC enable register 0	SADEN0L	SADEN0	R/W	8/16	0x00
0xF80D		SADEN0H		R/W	8	0x00
0xF80E	SA-ADC enable register 1	SADEN1L	SADEN1	R/W	8/16	0x00
0xF80F		SADEN1H		R/W	8	0x00
0xF810 to 0xF81F	Reserved	-	-	-	-	-
0xF820	SA-ADC upper/lower limit mode register	SADLMODL	SADLMOD	R/W	8/16	0x00
0xF821		SADLMODH		R/W	8	0x00
0xF822	SA-ADC upper limit setting register	SADUPLL	SADUPL	R/W	8/16	0xF0
0xF823		SADUPLH		R/W	8	0xFF
0xF824	SA-ADC lower limit setting register	SADLOLL	SADLOL	R/W	8/16	0x00
0xF825		SADLOLH		R/W	8	0x00
0xF826	SA-ADC upper/lower limit status register 0	SADULS0L	SADULS0	R	8/16	0x00
0xF827		SADULS0H		R	8	0x00
0xF828	Reserved	-	-	-	-	-
0xF829	Reserved	-	-	-	-	-
0xF82A	SA-ADC upper/lower limit status clear register 0	SADULC0L	SADULC0	W	8/16	0x00
0xF82B		SADULC0H		W	8	0x00
0xF82C	Reserved	-	-	-	-	-
0xF82D	Reserved	-	-	-	-	-
0xF82E to 0xF82F	Reserved	-	-	-	-	-
0xF830	SA-ADC test mode register	SADTMOD	-	R/W	8	0x00
0xF831 to 0xF83D	Reserved	-	-	-	-	-

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF83E	SA-ADC result register	SADRL	SADR	R	8/16	0x00
0xF83F		SADRH		R	8	0x00
0xF840	SA-ADC result register 0	SADR0L	SADR0	R	8/16	0x00
0xF841		SADR0H		R	8	0x00
0xF842	SA-ADC result register 1	SADR1L	SADR1	R	8/16	0x00
0xF843		SADR1H		R	8	0x00
0xF844	SA-ADC result register 2	SADR2L	SADR2	R	8/16	0x00
0xF845		SADR2H		R	8	0x00
0xF846	SA-ADC result register 3	SADR3L	SADR3	R	8/16	0x00
0xF847		SADR3H		R	8	0x00
0xF848	SA-ADC result register 4	SADR4L	SADR4	R	8/16	0x00
0xF849		SADR4H		R	8	0x00
0xF84A	SA-ADC result register 5	SADR5L	SADR5	R	8/16	0x00
0xF84B		SADR5H		R	8	0x00
0xF84C	SA-ADC result register 6	SADR6L	SADR6	R	8/16	0x00
0xF84D		SADR6H		R	8	0x00
0xF84E	SA-ADC result register 7	SADR7L	SADR7	R	8/16	0x00
0xF84F		SADR7H		R	8	0x00
0xF850	SA-ADC result register 8	SADR8L	SADR8	R	8/16	0x00
0xF851		SADR8H		R	8	0x00
0xF852	SA-ADC result register 9	SADR9L	SADR9	R	8/16	0x00
0xF853		SADR9H		R	8	0x00
0xF854	SA-ADC result register 10	SADR10L	SADR10	R	8/16	0x00
0xF855		SADR10H		R	8	0x00
0xF856	SA-ADC result register 11	SADR11L	SADR11	R	8/16	0x00
0xF857		SADR11H		R	8	0x00
0xF858	SA-ADC result register 12	SADR12L	SADR12	R	8/16	0x00
0xF859		SADR12H		R	8	0x00
0xF85A	SA-ADC result register 13	SADR13L	SADR13	R	8/16	0x00
0xF85B		SADR13H		R	8	0x00

23.2.2 SA-ADC Mode Register (SADMOD)

This is a SFR to set an operation mode and operating clock frequency for SA-ADC.

Address :	0xF800 (SADMODL/SADMOD), 0xF801 (SADMODH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADMOD															
Byte	SADMODH								SADMODL							
Bit	SAINIT T3	SAINIT T2	SAINIT T1	SAINIT T0	SAINIT	-	-	SASHT 4	SASHT 3	SASHT 2	SASHT 1	SASHT 0	SACK2	SACK1	SACK0	SALP
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	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	SAINITT3 to SAINITT0	These bits are used to be configured amplifier stability time at conversion starting. wait time for stability [s] = setting value / SAD_CLK frequency This time should be equal or more than 0.5[μs] When SAINIT=”1”, it is included discharge time for sample hold capacitor. In this case, this time should be equal or more than 0.65[μs]. Make decision on the value with external impedance of input pin. Table 23-3 shows example for typical setting.
11	SAINIT	This 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: Without discharging (Initial value) 1: With discharging
10 to 9	-	Reserved bits
8 to 4	SASHT4 to SASHT0	These bits are used to set the sampling time. sampling time [s] = (setting value + 1) / SAD_CLK frequency This setting value should be equal or more than 3. This time should be equal or more than 0.5[μs] at V_{REF}≥2.7V, 4[μs] at V_{REF}≥2.1V Make decision on the value with external impedance of input pin. Table 23-4 shows example for typical setting.
3 to 1	SACK2 to SACK0	These bits are used to choose the frequency of the A/D conversion operating clock (SAD_CLK). The SAD_CLK frequency should be equal or lower 16 MHz. 000: 1/1 x HSOCLK (Initial value) 001: 1/2 x HSOCLK 010: 1/4 x HSOCLK 011: 1/8 x HSOCLK 100: 1/16 x HSOCLK 101: Do not use 110: Do not use 111: 1/1 x LSCLK0 The following formula is calculated A/D conversion time without discharge/amp. stability time. A/D conversion time [s] = (SASHT4 to SASHT0 value + 15) / SAD_CLK frequency
0	SALP	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

Table 23-3 Example for SAINITT3 to 0 setting

SAD_CLK	SAINITT3 to 0 SAINIT=1 (Discharge time/Amp. stability) > 0.65us	SAINITT3 to 0 SAINIT=0 (Amp. stability) > 0.5us
16 MHz	1010	1000
12 MHz	1000	0110
8 MHz	0101	0100
6 MHz	0100	0011
4 MHz	0011	0010
< 4 MHz	0010	0010

Table 23-4 Example for SASHT4 to 0 setting

SAD_CLK	SASHT4 to 0 (Sampling time)		
	V_{REF} ≥ 2.1V	V_{REF} ≥ 2.4V	V_{REF} ≥ 2.7V
~ 16 MHz	-	-	00111
~ 8 MHz	-	00011	00011
1 MHz	00011	00011	00011
32.768 kHz	00011	00011	00011

23.2.3 SA-ADC Control Register (SADCON)

SADCON is a SFR used to control the operation of the A/D converter.

Address :	0xF802 (SADCONL/SADCON), 0xF803 (SADCONH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0000

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

Bit No.	Bit symbol name	Description
15 to 2	-	Reserved bits
1	SATGEN	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 (LSCLK0) to start or stop the A/D conversion after setting or resetting the SARUN bit.**

23.2.4 SA-ADC Conversion Interval Register (SADSTM)

This is a SFR used to set the interval time in the consecutive scan A/D conversion mode.

Address :	0xF804 (SADSTML/SADSTM), 0xF805 (SADSTMH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADSTM															
Byte	SADSTMH								SADSTML							
Bit	SADSTM15	SADSTM14	SADSTM13	SADSTM12	SADSTM11	SADSTM10	SADSTM9	SADSTM8	SADSTM7	SADSTM6	SADSTM5	SADSTM4	SADSTM3	SADSTM2	SADSTM1	SADSTM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The interval time is determined by the following formula.

A/D conversion interval time = HSCLK cycle x SADSTM setting value

For an example, supposing to A/D convert channel 2 and channel 5, the A/D conversion interval time means the time after the channel 2 and channel 5 are A/D converted consecutively and before the A/D conversion of channel 2 is started. The next A/D conversion starts at the timing that the value set in this register has been counted with SAD_CLK.

23.2.5 Reference Voltage Control Register (VREFCON)

This is a SFR used to choose reference voltage for the SA-ADC.

Address :	0xF806 (VREFCON)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								VREFCON							
Bit	-	-	-	-	-	-	-	-	-	-	-	VREFP0	-	-	-	rsvd
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 5	-	Reserved bits
4	VREFP0	This is used to choose the reference voltage for the A/D conversion. 0: VDD pin (Initial value) 1: VREF pin
3 to 1	-	Reserved bits
0	rsvd	Reserved bit. Set “0” to this bit.

23.2.6 SA-ADC Interrupt Mode Register (SADIMOD)

This is a SFR used to choose the interrupt mode of the SA-ADC.

Address :	0xF808 (SADIMOD)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SADIMOD							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SADIMD1	SADIMD0
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	SADIMD1	This bit is used to choose the occurrence timing of SA-ADC interrupt request with upper/lower limit detection function. 0: Make the interrupt request at a timing corresponding to SADIMD0 setting, only when the detection function result coincides. (Initial value) 1: Make the interrupt request at a timing corresponding to SADIMD0 setting without the detection function result.
0	SADIMD0	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

23.2.7 SA-ADCTrigger Register (SADTRG)

This is a SFR used to control the trigger event for the SA-ADC.

Address :	0xF80A (SADTRG)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

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

Bit No.	Bit symbol name	Description
7 to 4	-	Reserved bits
3 to 0	SASTS3 to SASTS0	These bits are used to choose the source of the trigger event to start SA-ADC. 0000: 16-bit Timer 1 interrupt (TM1INT) (Initial value) 0001: 16-bit Timer 2 interrupt (TM2INT) 0010: 16-bit Timer 3 interrupt (TM3INT) 0011: Do not use (Reserved) 0100: Functional Timer 0 trigger (FTM0TRG) 0101: Functional Timer 1 trigger (FTM1TRG) 0110: Do not use (Reserved) 0111: Do not use (Reserved) 1000: Time Base Counter 0 interrupt (LTB0INT) 1001: Time Base Counter 2 interrupt (LTB2INT) 1110 to 1111: Do not use (Reserved)

23.2.8 SA-ADC Enable Register 0 (SADEN0)

This is a SFR used to choose channels of the A/D converter and enable/disable the conversion.

Address :	0xF80C (SADEN0L/SADEN0), 0xF80D (SADEN0H)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADEN0															
Byte	SADEN0H								SADEN0L							
Bit	-	-	SACH 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	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	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
This bit is used to control enable/disable the conversion on a target channel.

0: Disabled (Initial value)
1: Enabled

When multiple bits of SACHn (n=00 to 17) are set to "1", the A/D conversion starts in the order of smaller channel number.

Bit No.	Bit symbol name	Description (target channel)
15	-	Reserved bits
14	-	Reserved bits
13	SACH13	Channel 13
12	SACH12	Channel 12
11	SACH11	Channel 11
10	SACH10	Channel 10
9	SACH09	Channel 9
8	SACH08	Channel 8
7	SACH07	Channel 7
6	SACH06	Channel 6
5	SACH05	Channel 5
4	SACH04	Channel 4
3	SACH03	Channel 3
2	SACH02	Channel 2
1	SACH01	Channel 1
0	SACH00	Channel 0

[Note]

- **Do not start the A/D conversion when the all bits of SACHn (n=00 to 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)

This is a SFR used to choose channels of the A/D converter and enable/disable the conversion.

Address :	0xF80E (SADEN1L/SADEN1), 0xF80F (SADEN1H)
Access :	R/W
Access size :	8/16 bit
Initial value :	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADEN1															
Byte	SADEN1H								SADEN1L							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SACH 17	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	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 (target channel)
15 to 2	-	Reserved bits
1	SACH17	This bits is used to control enable/disable the conversion on channel 17; A/D converter test. 0: Disable the conversion on channel 17 (initial value) 1: Enable the conversion on channel 17
0	-	Reserved bit

[Note]

- **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".**

23.2.10 SA-ADC Upper/Lower Limit Mode Register (SADLMOD)

This is a SFR used to set modes in the A/D conversion result upper/lower limit detection function.

Address : 0xF820 (SADLMODL/SADLMOD), 0xF821 (SADLMODH)
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	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 (SADULS0) 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. If the interrupt occurred by satisfying the upper/lower limit detection condition, check the SA-ADC upper/lower status registers 0(SADULS0) to see which channel of A/D conversion result matched to the condition. SA-ADC Upper/Lower Limit Status Register 0 (SADULS0) are not updated when this bit is "0". 0: Disabled (Initial value) 1: Enabled

23.2.11 SA-ADC Upper Limit Setting Register (SADUPL)

This is a SFR used to set the upper limit of A/D conversion result.

Address :	0xF822 (SADUPLL/SADUPL), 0xF823 (SADUPLH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0xFFF0

	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	d5	d4	-	-	-	-
R/W	R/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	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

23.2.12 SA-ADC Lower Limit Setting Register (SADLOL)

This is a SFR used to set the lower limit of A/D conversion result.

Address :	0xF824 (SADLOLL/SADLOL), 0xF825 (SADLOLH)
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	SADLOL															
Byte	SADLOLH								SADLOLL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	-	-	-	-
R/W	R/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	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.2.13 SA-ADC Upper/Lower Limit Status Register 0 (SADULS0)

This is a read-only SFR used to indicate whether the A/D conversion result matches to the condition of upper/lower limit.

Address :	0xF826 (SADULS0L/SADULS0), 0xF827 (SADULS0H)
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	SADULS0															
Byte	SADULS0H								SADULS0L							
Bit	-	-	SAULS13	SAULS12	SAULS11	SAULS10	SAULS09	SAULS08	SAULS07	SAULS06	SAULS05	SAULS04	SAULS03	SAULS02	SAULS01	SAULS00
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is used to indicate whether the A/D conversion results of target channel matches to the condition of upper/lower limit.
0: Unmatched to the condition of upper/lower limit. (Initial value)
1: Matched to the condition of upper/lower limit.
Each bit is forcibly cleared to "0" by writing 1 corresponding bit in the SADULC0 register.
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 is “1” and a timing configured on the SAIMOD register.
Refer to Figure 23-9 to 23-11 for the timing of the interrupt and updates of detection result.

Bit No.	Bit symbol name	Description (target channel)
15	-	Reserved bit
14	-	Reserved bit
13	SAULS13	Channel 13 (AIN13)
12	SAULS12	Channel 12 (AIN12)
11	SAULS11	Channel 11 (AIN11)
10	SAULS10	Channel 10 (AIN10)
9	SAULS09	Channel 9 (AIN9)
8	SAULS08	Channel 8 (AIN8)
7	SAULS07	Channel 7 (AIN7)
6	SAULS06	Channel 6 (AIN6)
5	SAULS05	Channel 5 (AIN5)
4	SAULS04	Channel 4 (AIN4)
3	SAULS03	Channel 3 (AIN3)
2	SAULS02	Channel 2 (AIN2)
1	SAULS01	Channel 1 (AIN1)
0	SAULS00	Channel 0 (AIN0)

[Note]

- **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 SAULS13 to SAULS00 or by resetting the LSI.**
- **When performing the A/D conversion only one time (SALP bit =0), confirm the bit of SAULS13 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 SAULS13 to SAULS00 is "0", before the next A/D conversion ends.**

23.2.14 SA-ADC Upper/Lower Limit Status Clear Register 0 (SADULC0)

This is a write-only SFR used to clear the A/D conversion result matches to the condition of upper/lower limit.

Address :	0xF82A (SADULC0L/SADULC0), 0xF82B (SADULC0H)
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	SADULC0															
Byte	SADULC0H								SADULC0L							
Bit	-	-	SAULC13	SAULC12	SAULC11	SAULC10	SAULC09	SAULC08	SAULC07	SAULC06	SAULC05	SAULC04	SAULC03	SAULC02	SAULC01	SAULC00
R/W	R	R	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is used to clear a target result of upper/lower limit detection.
Writing “0”: Invalid
Writing “1”: clear a target result

Bit No.	Bit symbol name	Description (target channel)
15	-	Reserved bit
14	-	Reserved bit
13	SAULC13	Channel 13 (AIN13)
12	SAULC12	Channel 12 (AIN12)
11	SAULC11	Channel 11 (AIN11)
10	SAULC10	Channel 10 (AIN10)
9	SAULC09	Channel 9 (AIN9)
8	SAULC08	Channel 8 (AIN8)
7	SAULC07	Channel 7 (AIN7)
6	SAULC06	Channel 6 (AIN6)
5	SAULC05	Channel 5 (AIN5)
4	SAULC04	Channel 4 (AIN4)
3	SAULC03	Channel 3 (AIN3)
2	SAULC02	Channel 2 (AIN2)
1	SAULC01	Channel 1 (AIN1)
0	SAULC00	Channel 0 (AIN0)

23.2.15 SA-ADC Result Register (SADR)

This is a read-only SFR used to store the A/D conversion results on channels 0 to 13 and 17 (A/D converter test function).

Address :	0xF83E (SADRL/SADR), 0xF83F (SADRH)
Access :	R
Access size :	8/16 bit
Initial value :	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADR															
Byte	SADRH								SADRL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The A/D conversion results of all channels are stored to this register. The result of each channel is overwritten. The A/D conversion test result on channel 17 is stored to this register only.

Symbol name	Channel
SADR	Latest conversion result of channels 0 to 13 and 17

23.2.16 SA-ADC Result Register n (SADRn : n=0 to 13)

This is a SFR used to store the SA-ADC conversion results on channels 0 to 13.

Address : 0xF840 (SADR0L/SADR0), 0xF841 (SADR0H),
0xF842 (SADR1L/SADR1), 0xF843 (SADR1H),
0xF844 (SADR2L/SADR2), 0xF845 (SADR2H),
0xF846 (SADR3L/SADR3), 0xF847 (SADR3H),
0xF848 (SADR4L/SADR4), 0xF849 (SADR4H),
0xF84A (SADR5L/SADR5), 0xF84B (SADR5H),
0xF84C (SADR6L/SADR6), 0xF84D (SADR6H),
0xF84E (SADR7L/SADR7), 0xF84F (SADR7H),
0xF850 (SADR8L/SADR8), 0xF851 (SADR8H),
0xF852 (SADR9L/SADR9), 0xF853 (SADR9H),
0xF854 (SADR10L/SADR10), 0xF855 (SADR10H),
0xF856 (SADR11L/SADR11), 0xF857 (SADR11H),
0xF858 (SADR12L/SADR12), 0xF859 (SADR12H),
0xF85A (SADR13L/SADR13), 0xF85B (SADR13H),
Access : R
Access size : 8/16 bit
Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SADRn															
Byte	SADRnH								SADRnL							
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The A/D conversion result of each channel can be read from SADRn.

Symbol name	Channel
SADR0	Channel 0 (AIN0)
SADR1	Channel 1 (AIN1)
SADR2	Channel 2 (AIN2)
SADR3	Channel 3 (AIN3)
SADR4	Channel 4 (AIN4)
SADR5	Channel 5 (AIN5)
SADR6	Channel 6 (AIN6)
SADR7	Channel 7 (AIN7)
SADR8	Channel 8 (AIN8)
SADR9	Channel 9 (AIN9)
SADR10	Channel 10 (AIN10)
SADR11	Channel 11 (AIN11)
SADR12	Channel 12 (AIN12)
SADR13	Channel 13 (AIN13)

23.2.17 SA-ADC Test Mode Register (SADTMOD)

This is a SFR used to control the SA-ADC test function.

Address :	0xF830 (SADTMOD)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								SADTMOD							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SADT M1	SADT M0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This function enables to check if the successive approximation type A/D converter and the analog switch work correctly, by performing the A/D conversion for the full scale, zero scale and the internal reference voltage (approx. 1.0 V).
The A/D conversion result is stored in the SA-ADC result register (SADR).

Also, the AIN0-13 input level is measured by using a measurement value of internal reference voltage.
For example:
1: Convert at V_{REF} = V_{DD}, SACH17=1, SADTM1-0=3, where the result is “a”.
2: Convert at V_{REF} = V_{DD}, SACHn=1, where the result is “b”.
An input level from AINn is b/a [V] .

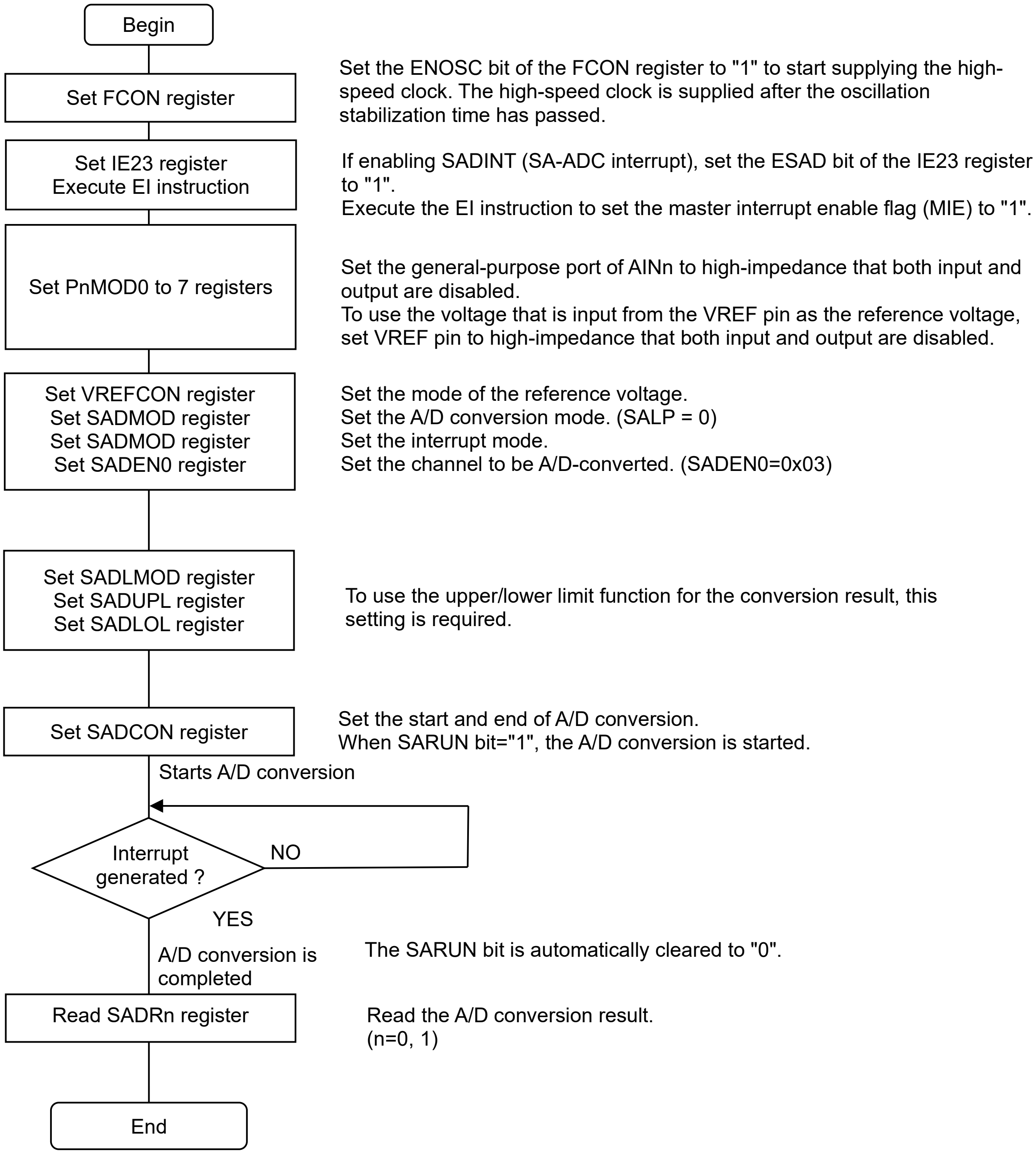
The seriality measurement can be by setting the SACHn and SACH17 at once. Read the results from SADRn and SADR.

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1 to 0	SADTM1, SADTM0	These bits are used to set the successive approximation type A/D converter test function. This is selected input to channel 17. 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.1.0V) 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.



Set the ENOSC bit of the FCON register to "1" to start supplying the high-speed clock. The high-speed clock is supplied after the oscillation stabilization time has passed.

If enabling SADINT (SA-ADC interrupt), set the ESAD bit of the IE23 register to "1".
Execute the EI instruction to set the master interrupt enable flag (MIE) to "1".

Set the general-purpose port of AINn to high-impedance that both input and output are disabled.
To use the voltage that is input from the VREF pin as the reference voltage, set VREF pin to high-impedance that both input and output are disabled.

Set the mode of the reference voltage.
Set the A/D conversion mode. (SALP = 0)
Set the interrupt mode.
Set the channel to be A/D-converted. (SADEN0=0x03)

To use the upper/lower limit function for the conversion result, this setting is required.

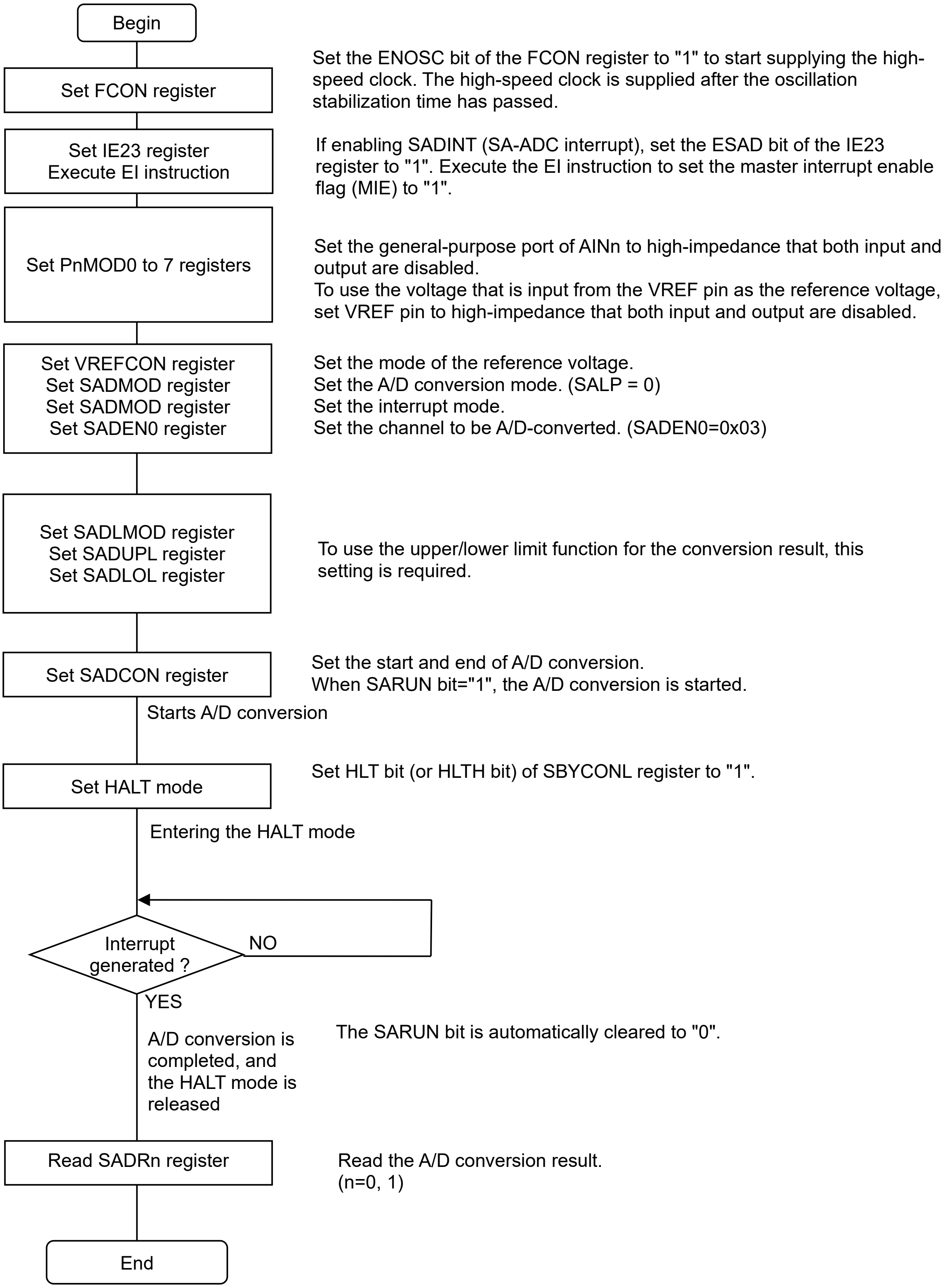
Set the start and end of A/D conversion.
When SARUN bit="1", the A/D conversion is started.

The SARUN bit is automatically cleared to "0".

Read the A/D conversion result.
(n=0, 1)

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.



Set the ENOSC bit of the FCON register to "1" to start supplying the high-speed clock. The high-speed clock is supplied after the oscillation stabilization time has passed.

If enabling SADINT (SA-ADC interrupt), set the ESAD bit of the IE23 register to "1". Execute the EI instruction to set the master interrupt enable flag (MIE) to "1".

Set the general-purpose port of AINn to high-impedance that both input and output are disabled.
To use the voltage that is input from the VREF pin as the reference voltage, set VREF pin to high-impedance that both input and output are disabled.

Set the mode of the reference voltage.
Set the A/D conversion mode. (SALP = 0)
Set the interrupt mode.
Set the channel to be A/D-converted. (SADEN0=0x03)

To use the upper/lower limit function for the conversion result, this setting is required.

Set the start and end of A/D conversion.
When SARUN bit="1", the A/D conversion is started.

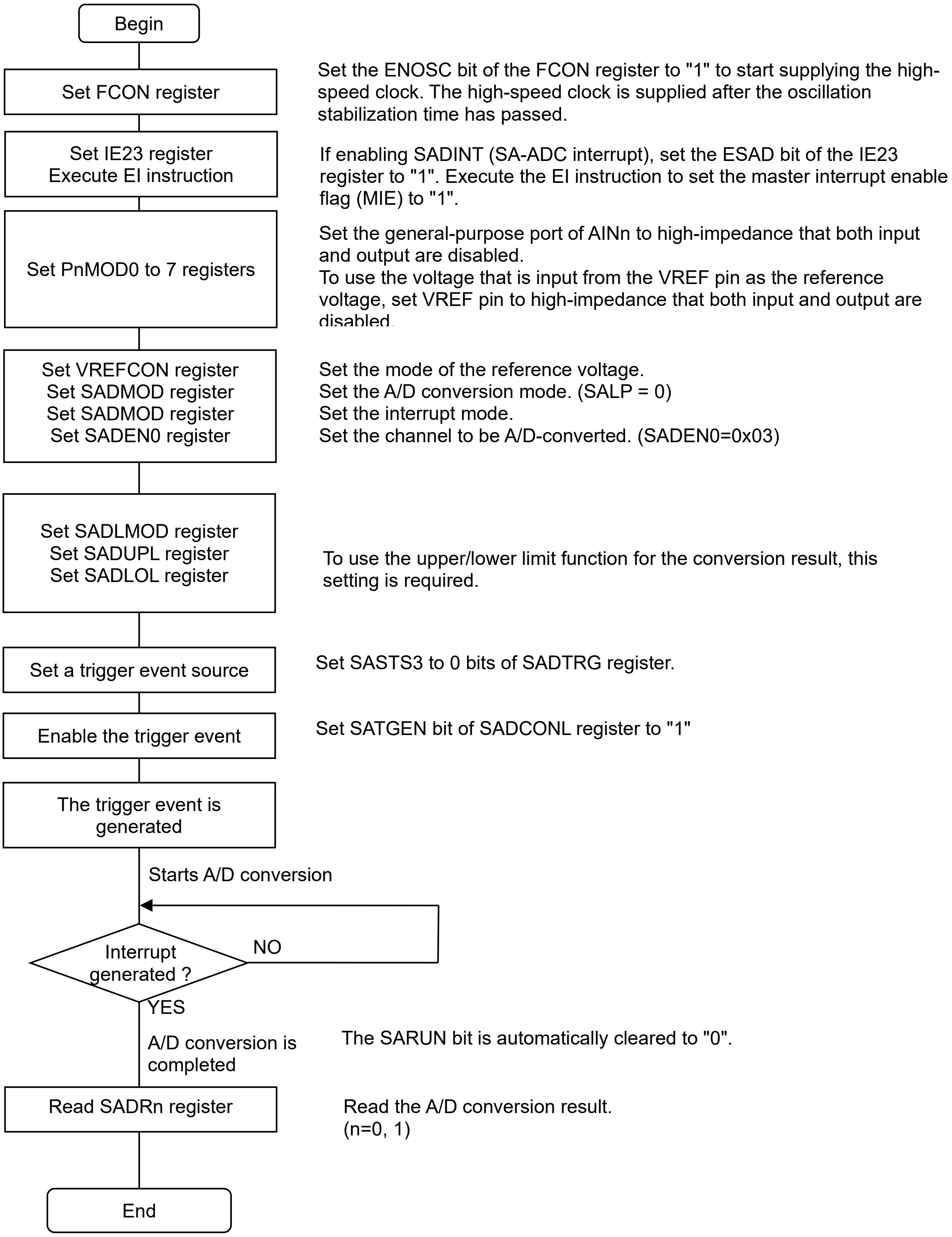
Set HLT bit (or HLTH bit) of SBYCONL register to "1".

The SARUN bit is automatically cleared to "0".

Read the A/D conversion result.
(n=0, 1)

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.



Set the ENOSC bit of the FCON register to "1" to start supplying the high-speed clock. The high-speed clock is supplied after the oscillation stabilization time has passed.

If enabling SADINT (SA-ADC interrupt), set the ESAD bit of the IE23 register to "1". Execute the EI instruction to set the master interrupt enable flag (MIE) to "1".

Set the general-purpose port of AINn to high-impedance that both input and output are disabled.
To use the voltage that is input from the VREF pin as the reference voltage, set VREF pin to high-impedance that both input and output are disabled.

Set the mode of the reference voltage.
Set the A/D conversion mode. (SALP = 0)
Set the interrupt mode.
Set the channel to be A/D-converted. (SADEN0=0x03)

To use the upper/lower limit function for the conversion result, this setting is required.

Set SASTS3 to 0 bits of SADTRG register.

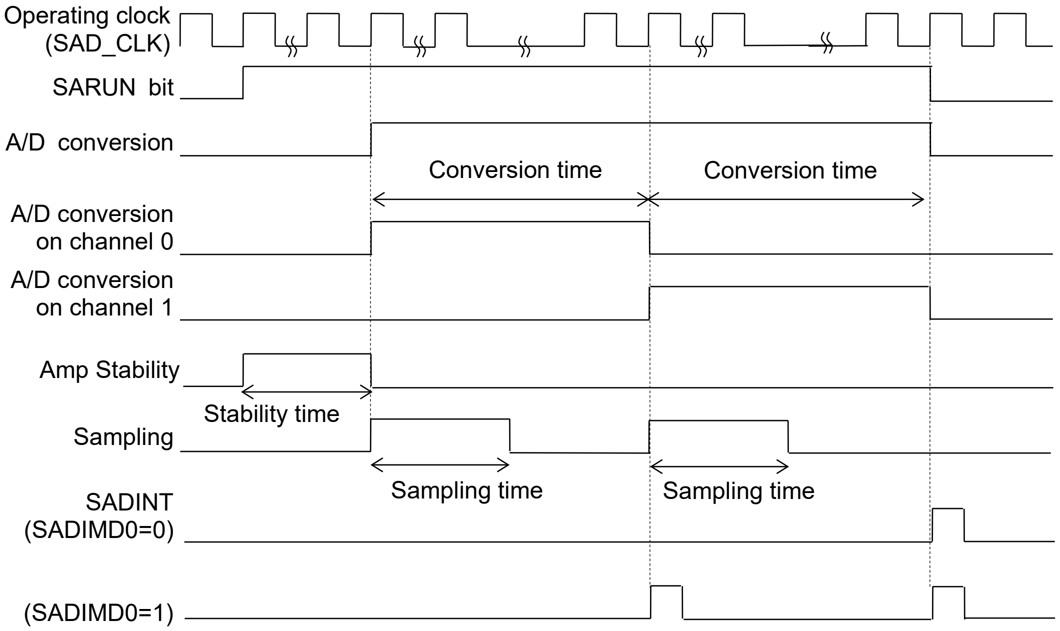
Set SATGEN bit of SADCONL register to "1"

The SARUN bit is automatically cleared to "0".

Read the A/D conversion result.
(n=0, 1)

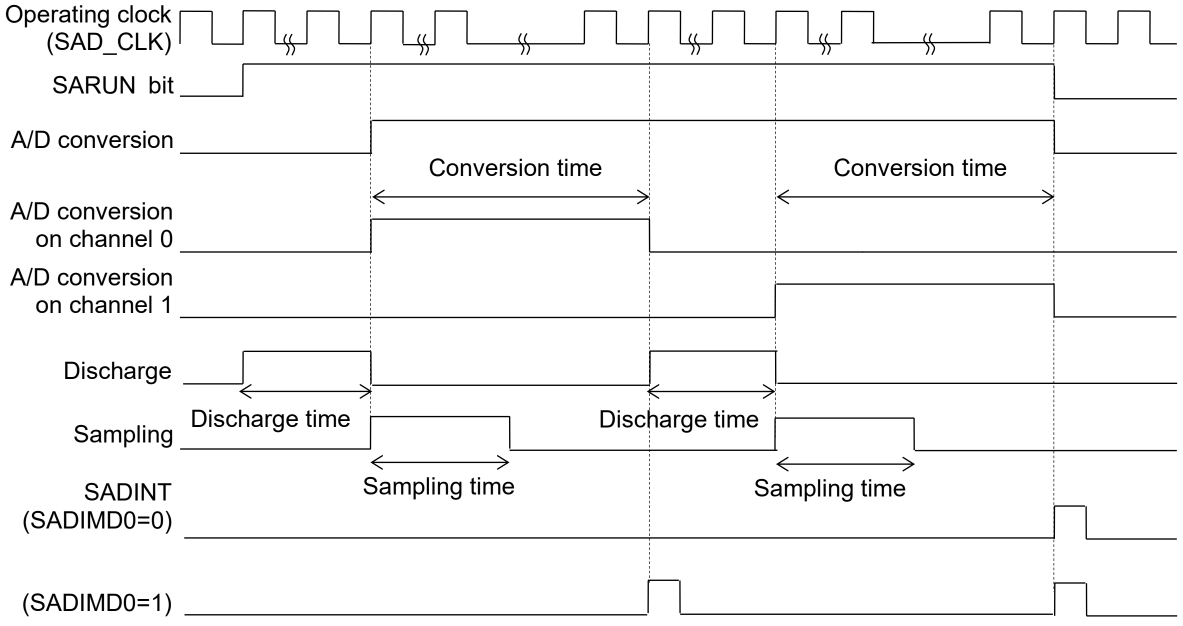
Figure 23-4 Example of A/D Conversion Setting (Start converting by a trigger event)

Figure 23-5 and 23-6 show operation waveforms when one-time A/D conversion is performed using channel 1 and 0.



Operating clock (SAD_CLK) is configured by SACK2-0 bits of SADMOD register.
Amp stability time is configured by SAINITT3-0 bits of SADMOD register.
Sampling time is configured by SASHT4-0 bits of SADMOD register.
Interrupt mode is configured by SADIMOD register.

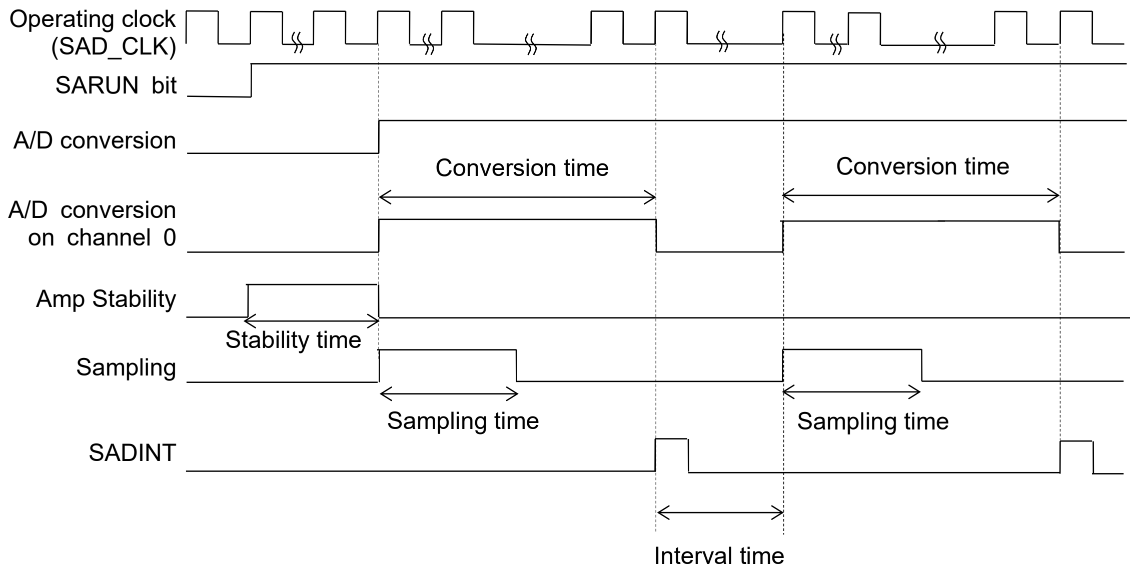
Figure 23-5 Operation Waveforms of A/D Conversion (One-time Conversion, Without Discharge)



Operating clock (SAD_CLK) is configured by SACK2-0 bits of SADMOD register.
Discharge enabling and time are configured by SAINIT bit and SAINITT3-0 bits of SADMOD register.
Sampling time is configured by SASHT4-0 bits of SADMOD register.
Interrupt mode is configured by SADIMOD register.

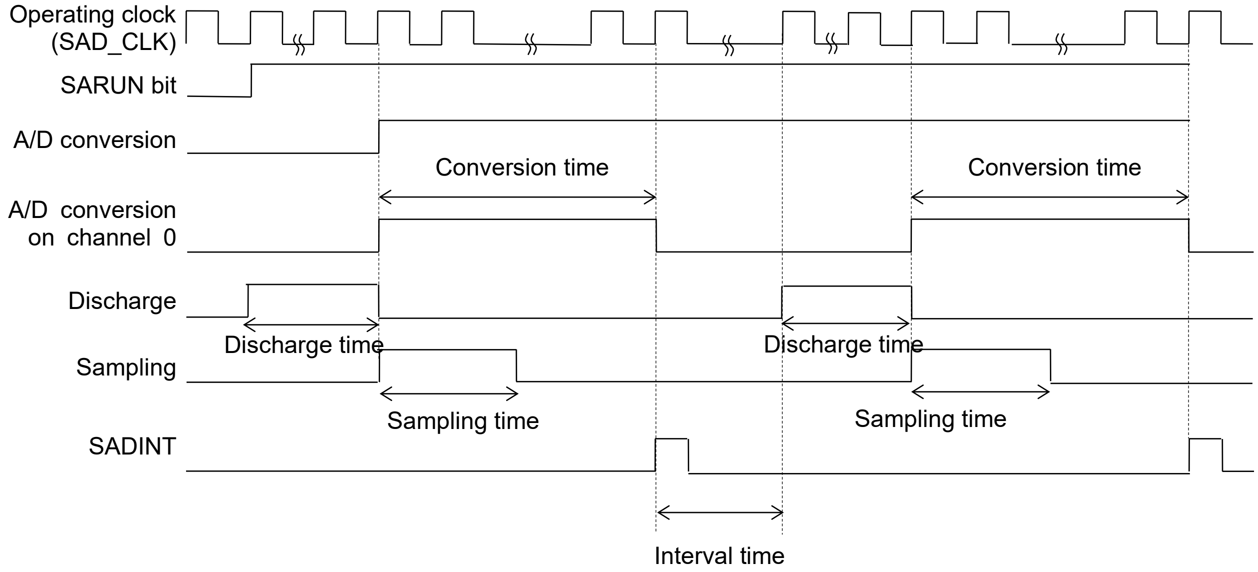
Figure 23-6 Operation Waveforms of A/D Conversion (One-time Conversion, With Discharge)

Figure 23-7 and 23-8 show the operation waveforms when the continuous A/D conversion is performed using channel 0.



Operating clock (SAD_CLK) is configured by SACK2-0 bits of SADMOD register.
Amp stability time is configured by SAINITT3-0 bits of SADMOD register.
Sampling time is configured by SASHT4-0 bits of SADMOD register.
Interrupt mode is configured by SADIMOD register.

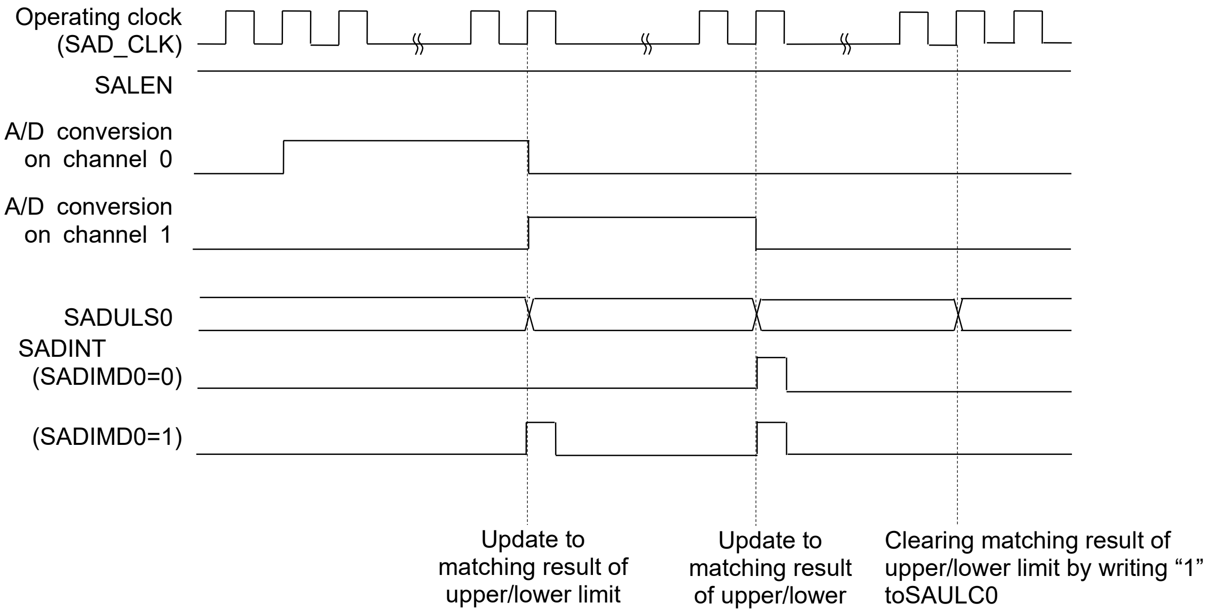
Figure 23-7 Operation Waveforms of A/D Conversion (Continuous Conversion, Without Discharge)



Operating clock (SAD_CLK) is configured by SACK2-0 bits of SADMOD register.
Discharge enabling and time are configured by SAINIT bit and SAINITT3-0 bits of SADMOD register.
Sampling time is configured by SASHT4-0 bits of SADMOD register.
Interrupt mode is configured by SADIMOD register.

Figure 23-8 Operation Waveforms of A/D Conversion (Continuous Conversion, With Discharge)

Figure 23-9 to 23-11 show the operation waveforms when an A/D conversion is performed with upper/lower limit function using channel 1 and 0.


Operating clock
(SAD_CLK)
SALEN
A/D conversion on channel 0
A/D conversion on channel 1
SADULS0
SADINT
(SADIMD0=0)
(SADIMD0=1)
Update to matching result of upper/lower limit
Update to matching result of upper/lower
Clearing matching result of upper/lower limit by writing "1" toSAULC0

A reflection of writing "1" to SAULC0 bit to clear matching result of upper/lower limit, is delayed maximum 1clock of the SAD_CLK.

Figure 23-9 Operation Waveforms of A/D Conversion with Upper/Lower Limit
(when set SADIMD1 bit to 0, in the case of matched to the limit ranges (SALMD1 to 0))

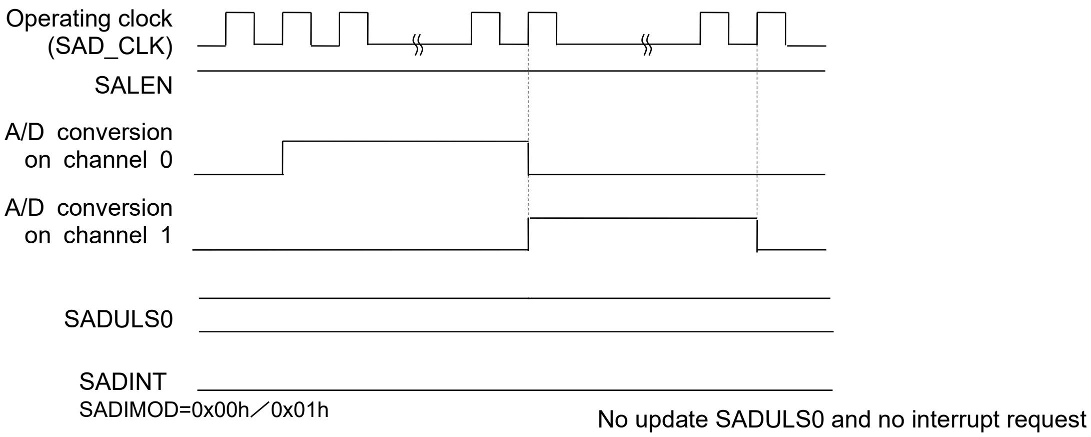
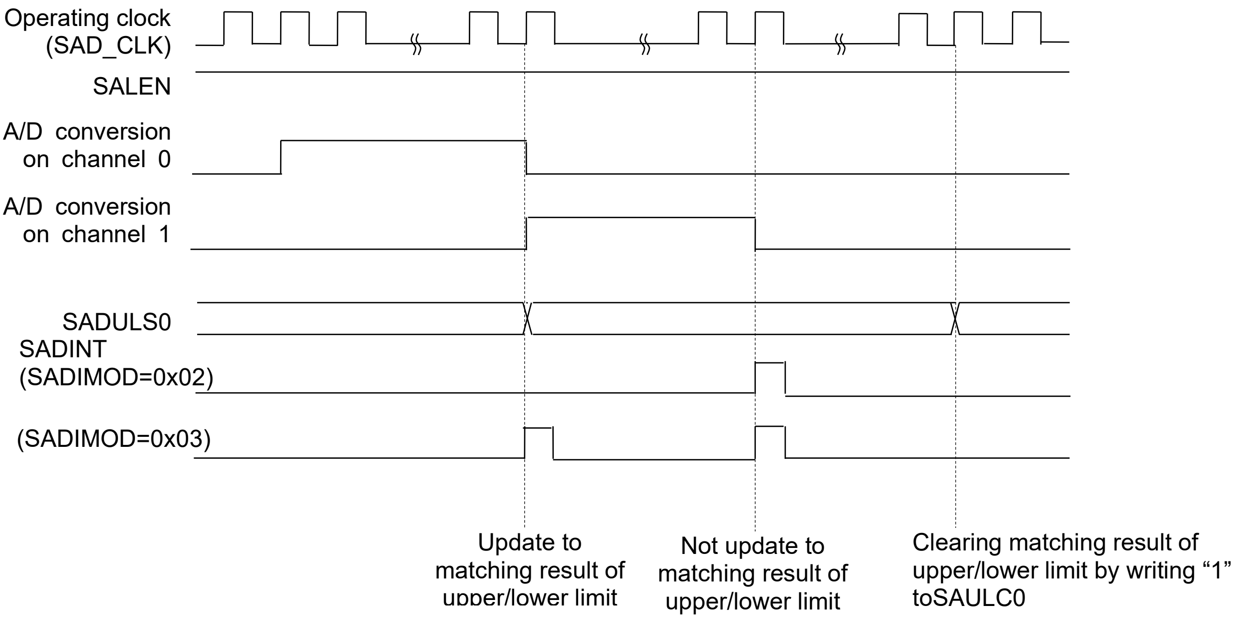

Operating clock
(SAD_CLK)
SALEN
A/D conversion on channel 0
A/D conversion on channel 1
SADULS0
SADINT
SADIMOD=0x00h／0x01h
No update SADULS0 and no interrupt request

Figure 23-10 Operation Waveforms of A/D Conversion with Upper/Lower Limit
(when set SADIMD1 bit to 0, in the case of not matched to the limit ranges (SALMD1 to 0))



The interrupt occurs regardless of matching result of upper/lower limit.

Figure 23-11 Operation Waveforms of A/D Conversion with Upper/Lower Limit (SADIMD1 = 1)

23.3.2 How to test the Successive Approximation Type A/D Converter

The self test can be performed by A/D-converting the full scale, zero scale and internal reference voltage.
Follow this procedure to check if the successive approximation type A/D converter works correctly. (n=0 to 13)

(1) A/D convert AINn pin. (conversion result 1)
(2) A/D convert AIN=full scale by setting the SADTMOD register (SADTMOD=0x01).
(3) A/D convert the AINn pin. (conversion result 2)
(4) A/D convert AIN=zero scale by setting the SADTMOD register (SADTMOD=0x02).
(5) A/D convert the AINn pin. (conversion result 3)
(6) A/D convert AIN=internal reference voltage(approx.1.0V) by setting the SADTMOD register (SADTMOD=0x03).
(7) A/D convert the AINn pin. (conversion result 4)
(8) Confirm conversion result 1 = conversion result 2 = conversion result 3 = conversion result 4.
Use the same AINn pin for the A/D conversion in (1), (3), (4) and (7).
(9) Confirm the conversion result in (2), (4) and (6) is different each other and also different from the result in (1), (3), (5) and (7).

23.4 Notes on SA-ADC

23.4.1 Sampling Time Setting

Sampling time of the SA-ADC should satisfy the following formula:

$$\text{Sampling time} > 9(C_{SAMPLE} + C_{PARA})(R_1 + R_2)$$

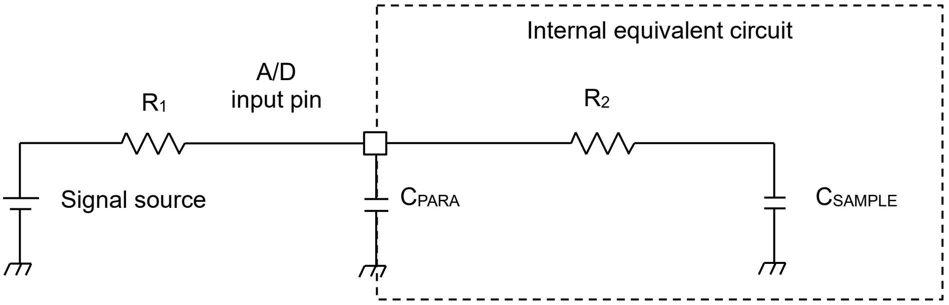
To calculate sampling time more precisely, use the following formula:

$$\text{Sampling time} = \left\{log_e(2^n) + log_e\left(\frac{C_{SAMPLE}}{C_{SAMPLE} + C_{PARA}}\right)\right\}(C_{SAMPLE} + C_{PARA})(R_1 + R_2)$$

C_{PARA} varies depending on board-layout and connected parts. Please check the accuracy of SA-ADC with the actual board.

R_1 : Input impedance of external resistor
R_2 : Internal resistor value which is the sum of the internal resistor and the ON register of the switch
C_{SAMPLE} : Sample hold capacitor
C_{PARA} : Parasitic capacitance of the A/D input line.(Measure the capacitance between the A/D input line and $V_{SS.}$)
n : Resolution of SA-ADC

The following diagram shows the equivalent circuit in this case:



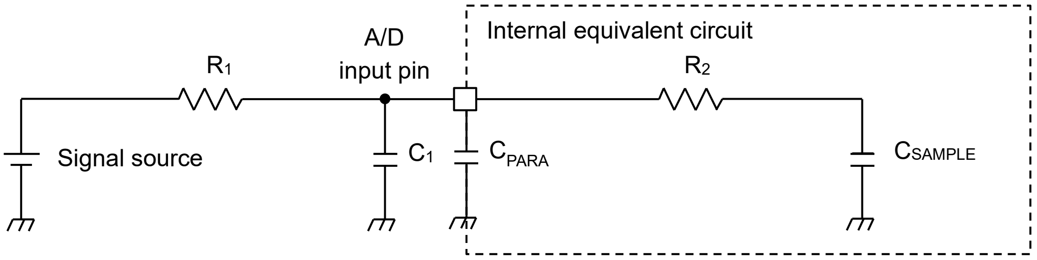
V_{REF}	R_2[kΩ]	C_{SAMPLE}[pF]
2.1V≤V_{REF}≤2.4V	170k	5pF
2.4V≤V_{REF}≤2.7V	20k	5pF
2.7V≤V_{REF}≤5.5V	10k	5pF

The values above are reference values.
Set the sampling time for V_{REF} condition that includes the lowest voltage of the usage range of V_{REF}.
If the sampling time above is unsatisfied, connect the external capacitor near by A/D input pin to satisfy the following formula.

$$(C_1 + C_{PARA}) > 2^n C_{SAMPLE}$$
$$\text{Sampling time} > 9C_{SAMPLE}R_2$$

C_1 : External capacitor

The equivalent circuit when the external capacitor C_1 is connected is as follows:



Note that the voltage at the A/D input pin transitionally changes due to the external capacitor C_1 and the external resistor R_1. Therefore, when sampling data, wait until the voltage is stabilized. If the stabilization timing is unknown, perform A/D conversion once, then wait for time constant τ (=R_1C_1) to 4τ or so and perform A/D conversion again. Confirm that the difference between values is small, and then sample data.

23.4.2 Noise Suppression

In order to prevent deterioration in accuracy of A/D conversion, operate the A/D converter in the environment with little noise.
The following processes are recommended for noise reduction:

- Perform A/D conversion in the HALT mode.
- Do not have clock input/output to and from a pin located in the vicinity of the pin in which A/D conversion is in progress.
- Do not have clock input/output to and from the pin in which A/D conversion is in progress and other A/D conversion pins.

In addition, the capacitor for noise suppression should be connected between VREF and VSS, as well as between VDD and VSS. When connecting, place the capacitor in the immediate vicinity of LSI using short wiring.

Chapter 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
    - Stack overflow/underflow break
    - Unused ROM area access break
    - RAM parity error break
  - Real time watch
  - CPU resource display/change
    - Program memory reference/disassembly
    - RAM and SFR display/change
    - Register display/change in the CPU
  - Program download
    - Program download/read/erase to/from flash memory
    - Data write/read/erase to/from data flash
  - Peripheral circuit operation continue/stop control during break
    Target peripheral circuits : External interrupt, Low-speed time base counter, 16-bit timer, Functional timer, UART, Analog module (Successive approximation type A/D converter, 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}=1.8 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.

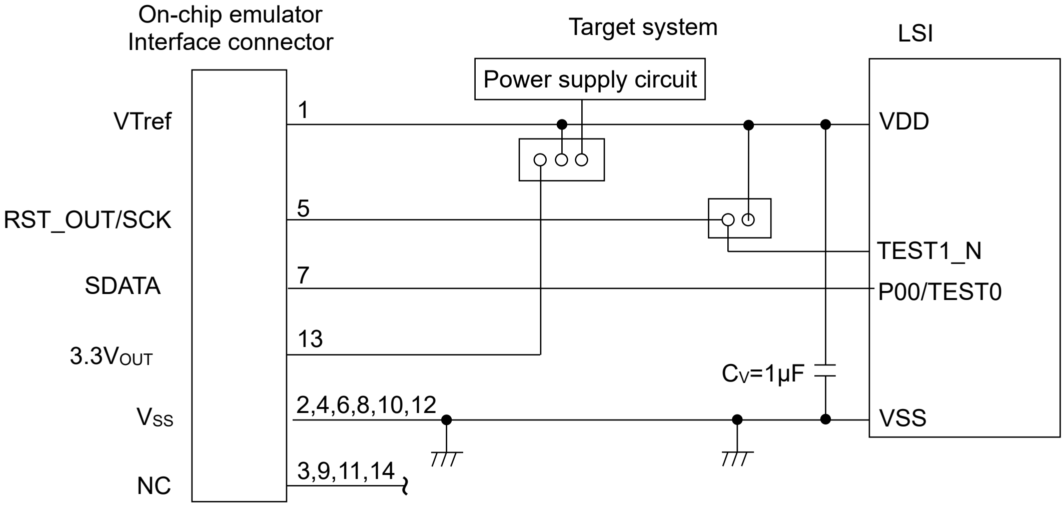


Figure 28-1 Connection Example When Using On-chip Emulator 3.3 VOUT Power Supply

28.1.2.2 Using Power Supply of Target System (V_{DD}=1.8 V to 5.5 V)

Figure 28-2 shows a connection example when using the power supply (V_{DD}=1.8 V to 5.5 V) of the target system.

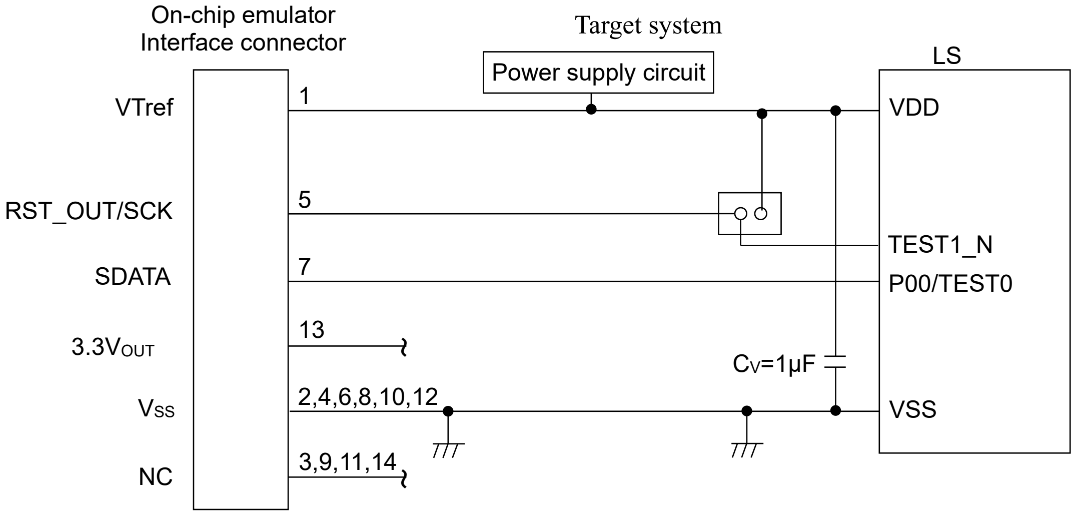


Figure 28-2 Connection Example When Using Target System Power Supply

28.1.3 List of Pins

The following pins are used for the on-chip debug function.

Signal name	I/O	Function
TEST1_N	I	On-chip debug function signal 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 TEST1_N pin able to be connected to VDD with a jumper or something when not using the on-chip debug function.
- 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 VDD 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.
- If the contents of the data memory are displayed in the debugger in a state where a RAM parity error may occur (including when the RAM is not initialized), a RAM parity error may occur even if the RAM area is not displayed.
- The all interrupts and watchdog timer operation always stop while the debugger is in the break state.
- 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 TEST1_N pin, prepare a jumper function on the board so that the capacitor gets dis-connectable 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.
LTBR1	LTBR1 of Low-speed Time Base	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.
LTBR0	LTBR0 of Low-speed Time Base	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.
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	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.
UART	UART	If the item is checked on, operation of the peripheral operation continues during the break. If the item checked off, operation of the peripheral stops during the break.
Analog Module （ADC/VLS）	SA-ADC and 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.

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 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

ML62Q2500 group 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^{2}C self-test function	Make the I^{2}C 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 to 0xFBB	Reserved	-	-	-	-	-
0xF0BC	RAM Parity Setting Register	RASFMOD	-	R/W	8	0x00
0xF0BD	Reserved	-	-	-	-	-
0xF0BE	Communication Test Setting Register 0	COMFT0L	COMFT0	R/W	8/16	0x00
0xF0BF		COMFT0H		R/W	8	0x00
0xF050	MCU Status Interrupt Enable Register	MCINTEL	-	R/W	8	0x00
0xF051	Reserved	-	-	-	-	-
0xF052	MCU Status Interrupt Register	MCISTATL	-	R	8	0x00
0xF053	Reserved	-	-	-	-	-
0xF054	MCU Status Interrupt Clear Register (L/H)	MCINTCLL	-	W	8	0x00
0xF055		MCINTCLH		W	8	0x00

29.2.2 RAM Guard Setting Register (RAMGD)

RAMGD is a SFR used to disable writing the RAM.
Data in the specified RAM area is protectable.

Address:	0xF0B0 (RAMGD)
Access:	R/W
Access size:	8 bit
Initial value:	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								RAMGD							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	RGD2	RGD1	RGD0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No.	Bit symbol name	Description
7 to 3	-	Reserved bits
2 to 0	RGD2 to RGD0	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 SFR used to disable writing certain SFRs.
Data in the specified SFR area is protectable.

Address:	0xF0B4 (SFRGD0L/SFRGD0), 0xF0B5(SFRGD0H)
Access:	R/W
Access size:	8/16 bit
Initial value:	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SFRGD0															
Byte	SFRGD0H								SFRGD0L							
Bit	-	-	-	-	-	-	-	-	-	-	SGD05	SGD04	SGD03	SGD02	SGD01	SGD00
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured writable/unwritable target SFRs
0: Target SFRs are writable and readable (Initial value)
1: Target SFRs are unwritable and readable

Bit No.	Bit symbol name	Description (target SFRs)
15 to 6	-	Reserved bits
5	SGD05	WDTMOD register; see Chapter “10 Watchdog timer”
4	SGD04	BCKCONn and BRECONn registers (n=0 to 3); see Chapter “4 Power management”
3	SGD03	RASFMOD register; see this chapter.
2	SGD02	SFRs described in chapter 22. VLS.
1	SGD01	SFRs described in chapter 6. Clock Generation Circuit.
0	SGD00	SFRs described in chapter 5 Interrupt

29.2.4 SFR Guard Setting Register 1 (SFRGD1)

SFRGD1 is a SFR used to disable writing certain SFRs.
Data in the specified SFR area is protectable.

Address:	0xF0B6(SFRGD1L/SFRGD1), 0xF0B7(SFRGD1H)
Access:	R/W
Access size:	8/16 bit
Initial value:	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	SFRGD1															
Byte	SFRGD1H								SFRGD1L							
Bit	SGD1F	-	-	-	-	-	-	-	SGD17	SGD16	SGD15	-	SGD13	SGD12	SGD11	SGD10
R/W	R/W	R	R	R	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	0	0	0	0	0	0	0

Common description of each bits :
It is configured writable/unwritable target SFRs in chapter 17. GPIO.
0: Target SFRs are writable and readable (Initial value)
1: Target SFRs are unwritable and readable

Bit No.	Bit symbol name	Description (target SFRs)
15	SGD1F	SFRs related to the port XT
14 to 8	-	Reserved bits
7	SGD17	SFRs related to the port 7
6	SGD16	SFRs related to the port 6
5	SGD15	SFRs related to the port 5
4	-	Reserved bit
3	SGD13	SFRs related to the port 3
2	SGD12	SFRs related to the port 2
1	SGD11	SFRs related to the port 1
0	SGD10	SFRs related to the port 0

29.2.5 RAM Parity Setting Register (RASFMOD)

RASFMOD is a special function register (SFR) used to control the RAM parity error reset function.
The RAM parity error is detectable and the RAM parity error reset is generatable.
The reset flag by a RAM parity error can be checked by the reset status register (SRSTAT).
See Chapter 3 "Reset Function"for details about the reset flag.

Address:	0xF0BC(RASFMOD)
Access:	R/W
Access size:	8 bit
Initial value:	0x00

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

Bit No.	Bit symbol name	Description
7	PERF	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: Disabled (Initial value) 1: Enabled
6 to 1	-	Reserved bits
0	PEREN	This bit is used to enable/disable the RAM parity error reset function. 0: Disabled (Initial value) 1: Enabled

29.2.6 Communication Test Setting Register (COMFT0)

COMFT0 is a SFR used to control the communication test function, which enables the loop back test with transmit data in the serial communication units. See Section 29.3.1 "Communication Function Self Test" for more details.
As the I^{2}C bus unit and the I^{2}C master are equipped with the function to read the transmit data, the function can be used for testing. For details, see Chapter 13 "I^{2}C Bus".

Address:	0xF0BE(COMFT0/COMFT0L), 0xF0BF(COMFT0H)
Access:	R/W
Access size:	8/16 bit
Initial value:	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ワード	COMFT0															
バイト	COMFT0H								COMFT0L							
ビット	-	-	-	CMFT12	-	-	-	CMFT08	-	-	-	-	-	CMFT02	CMFT01	CMFT00
R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W
初期値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Common description of each bits :
It is configured enable/disable the self-test for target communication function.
0: Target function is disabled. (Initial value)
1: Target function is enabled.

Bit No.	Bit symbol name	Description (target)
15 to 13	-	Reserved bits
12	CMFT12	SSIOF0
11 to 9	-	Reserved bits
8	CMFT08	SSIO0
7 to 3	-	Reserved bits
2	CMFT02	UART2
1	CMFT01	UART1
0	CMFT00	UART0

29.2.7 MCU Status Interrupt Enable Register (MCINTEL)

MCINTEL is a SFR used to control enabling/disabling three types of interrupt status on the microcontroller.

Address:	0xF050 (MCINTEL)
Access:	R/W
Access size:	8 bit
Initial value:	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								MCINTEL							
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	MCI2E	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

Common description of each bits :
It is configured enable/disable target interrupt.
0: Target interrupt is disabled. (Initial value)
1: Target interrupt is enabled.

Bit No.	Bit symbol name	Description (target interrupt)
7 to 3	-	Reserved bits
2	MCI2E	The interrupt at the completion of data flash erasing/programming.
1	MCI1E	The interrupt at the completion of automatic CRC calculation.
0	MCI0E	The interrupt at the occurrence of RAM parity error.

29.2.8 MCU Status Interrupt Register (MCISTATL)

MCISTATL is a read-only SFR used to indicate status of the three types of interrupts.
The MCI2S bit to MCI0S bit is initialized, in addition to reset function, by writing "1" to the same number of bit in the MCINTCL register.

Address: 0xF052 (MCISTATL)
Access: R
Access size: 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

Common description of each bits :
It is to indicate status of target interrupt.
0: Target interrupt has not been generated. (Initial value)
1: Target interrupt has been generated.

Bit No.	Bit symbol name	Description (target interrupt)
7 to 3	-	Reserved bits
2	MCI2S	The interrupt at the completion of data flash erasing/programming.
1	MCI1S	The interrupt at the completion of automatic CRC calculation.
0	MCI0S	The interrupt at the occurrence of RAM parity error.

[Note]

- **If the MCISTATL register is not zero, a request to interrupt controller is not given when a new interrupt occurs. Clear the MCISTATL register with the MCINTCL register before that time.**

29.2.9 MCU Status Interrupt Clear Register L/H (MCINTCLL, MCINTCLH)

MCINTCL is a write-only special function register (SFR) used to clear the MCU status interrupts.
If the MCI2C bit to MCI0C bit is set to "1", the interrupt request indicated by the same number of bit in the MCISTATL register gets cleared.
This register always returns "0x0000" for reading.

Address:	0xF054(MCINTCLL), 0xF055(MCINTCLH)
Access:	W
Access size:	8/16 bit
Initial value:	0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	MCINTCLH								MCINTCLL							
Bit	MCIR	-	-	-	-	-	-	-	-	-	-	-	-	MCI2C	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

Common description of each bits; bit 2 to 0 :
It is to indicate status of target interrupt.
Writing “0” : Invalid
Writeing “1”: Target interrupt status gets cleared.

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	The interrupt at the completion of data flash erasing/programming.
1	MCI1C	The interrupt at the completion of automatic CRC calculation.
0	MCI0C	The interrupt at the occurrence of RAM parity error.

29.3 Description of Operation

29.3.1 Communication Function Self-Test

This self test is enabled by the COMFT0 register setting.
The communication function can be tested through the self test by internally connecting transmit and receive data of UART 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 (2nd to 8th function) of the general-purpose port.
Figure 29-1 shows a concept diagram of the communication test. Figure 29-2 shows a flow chart of the communication test.

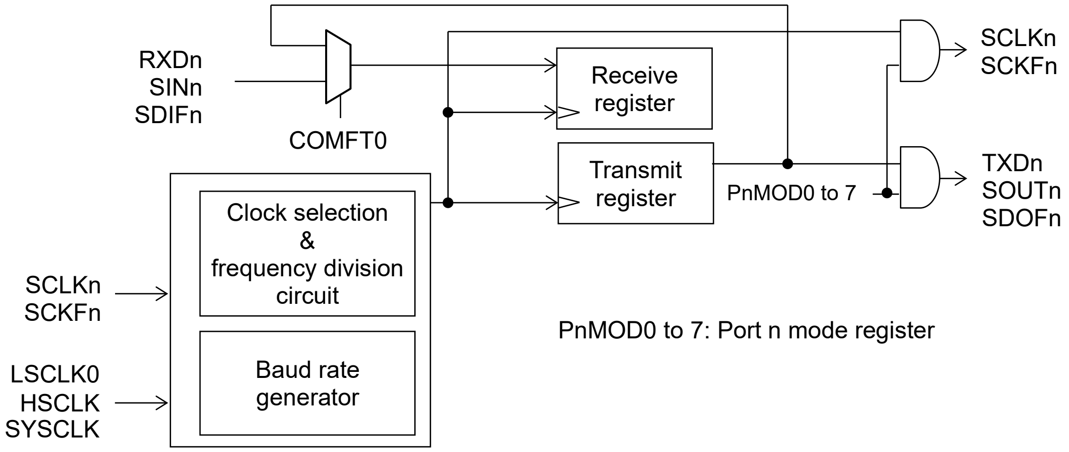


Figure 29-1 Communication Test Concept Diagram

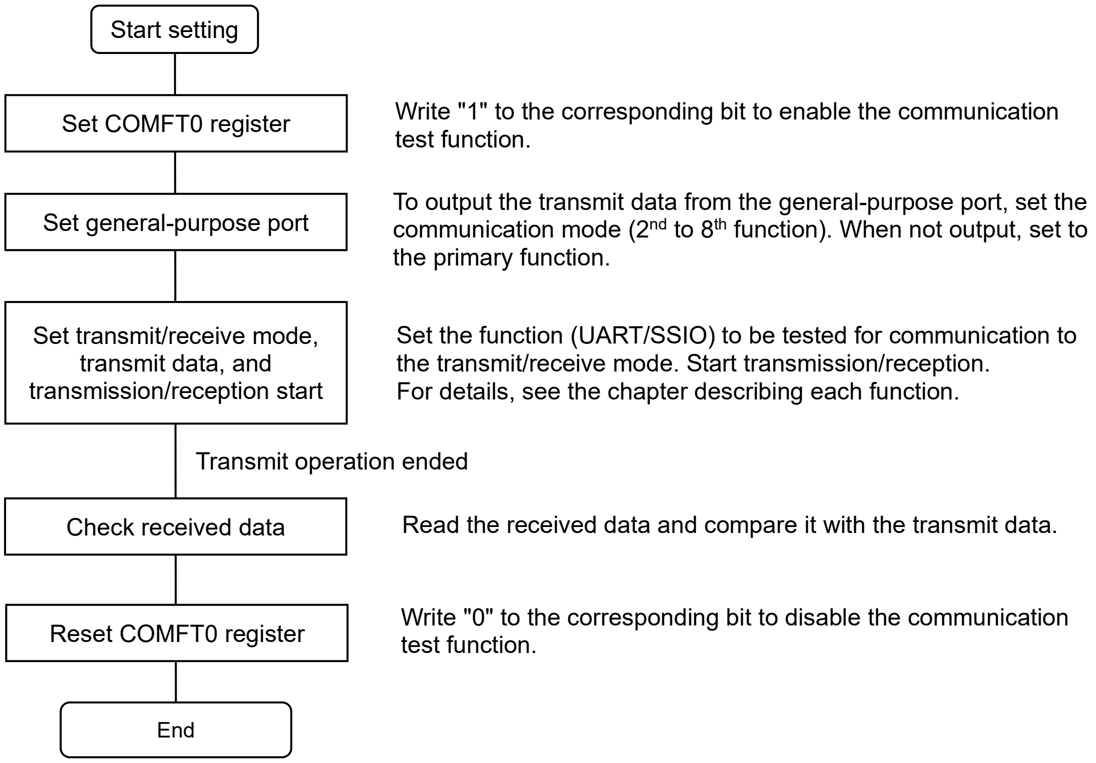

Start setting
Set COMFT0 register
Write "1" to the corresponding bit to enable the communication test function.
Set general-purpose port
To output the transmit data from the general-purpose port, set the communication mode (2nd to 8th function). When not output, set to the primary function.
Set transmit/receive mode, transmit data, and transmission/reception start
Set the function (UART/SSIO) to be tested for communication to the transmit/receive mode. Start transmission/reception.
For details, see the chapter describing each function.
Transmit operation ended
Check received data
Read the received data and compare it with the transmit data.
Reset COMFT0 register
Write "0" to the corresponding bit to disable the communication test function.
End

Figure 29-2 Communication Test Flow Chart

29.3.2 Unused ROM Area Access Reset Function

This function constantly monitors the program counter (PC) of the CPU. It generates the LSI reset when it detects that the program counter (PC) executes a program located outside of the area. This function can be enabled/disabled by the code option. The reset flag due to unused ROM area access can be confirmed with the SRSTAT register.
See Chapter 3 "Reset Function" for details of the reset flag.

<ROM unused area>

Program memory size	: CSR:PC
128KB	: 0x1:0FFC0 to 0x7:0FFFF
64KB	: 0x0:0FFC0 to 0x7:0FFFF

[Note]

- **CSR[3] is unused on the ML62Q2500 group. 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.

LSCLK0 is countable by a trigger of the RC1K, enables to monitor mutually the two oscillation clocks.
See the application note for more details.

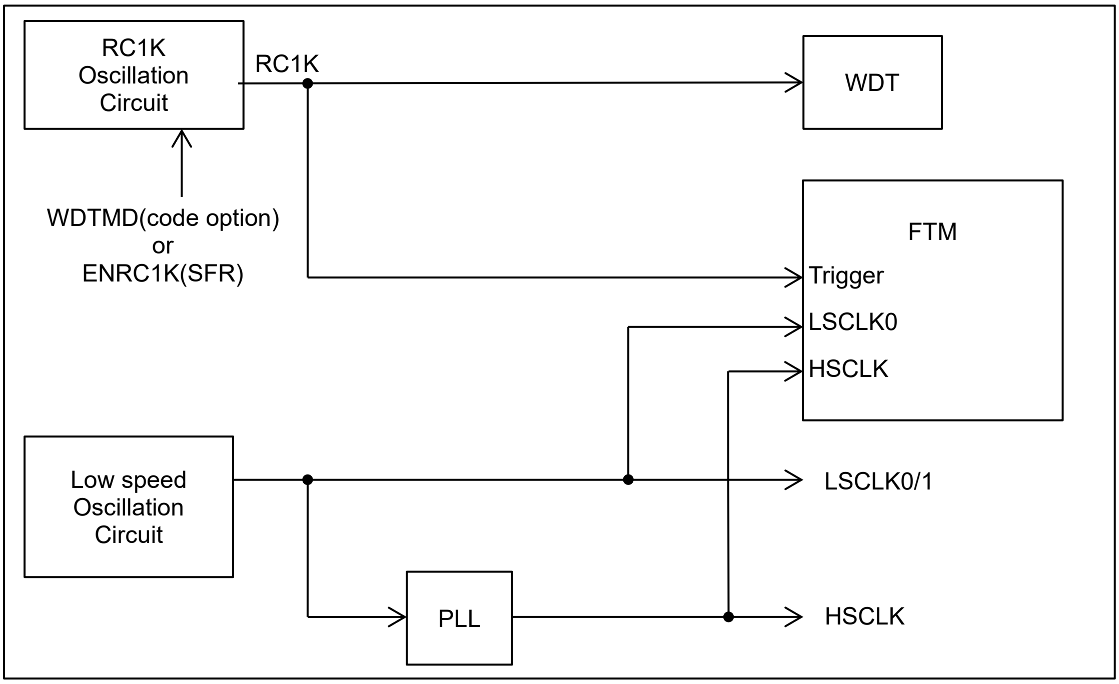
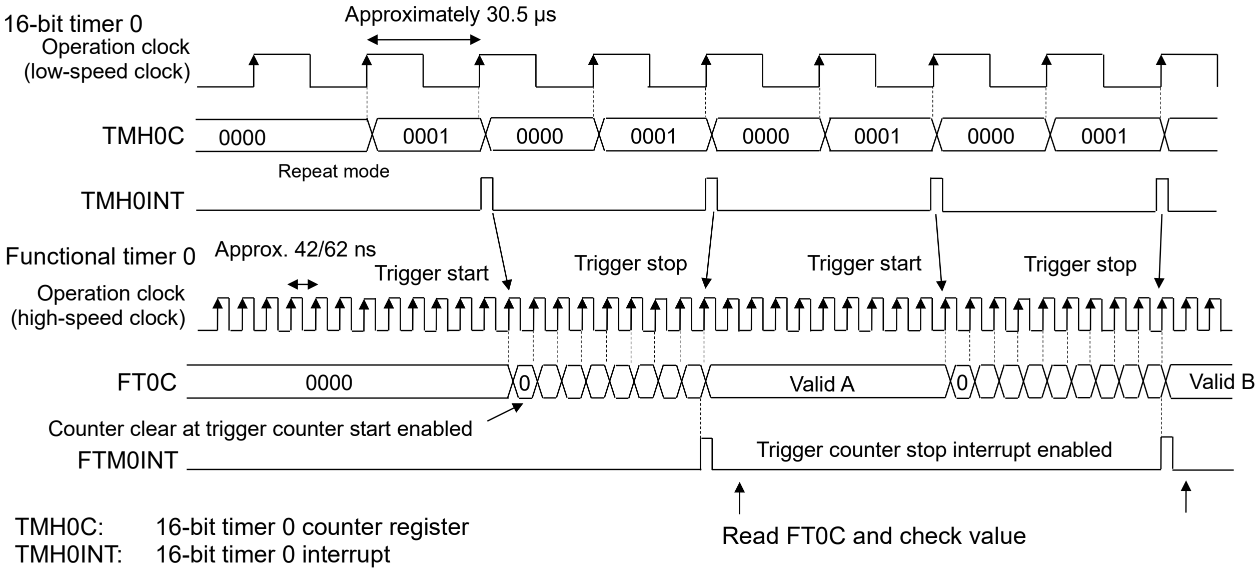


Figure 29-3 Clock Mutual Monitoring Function Block Diagram

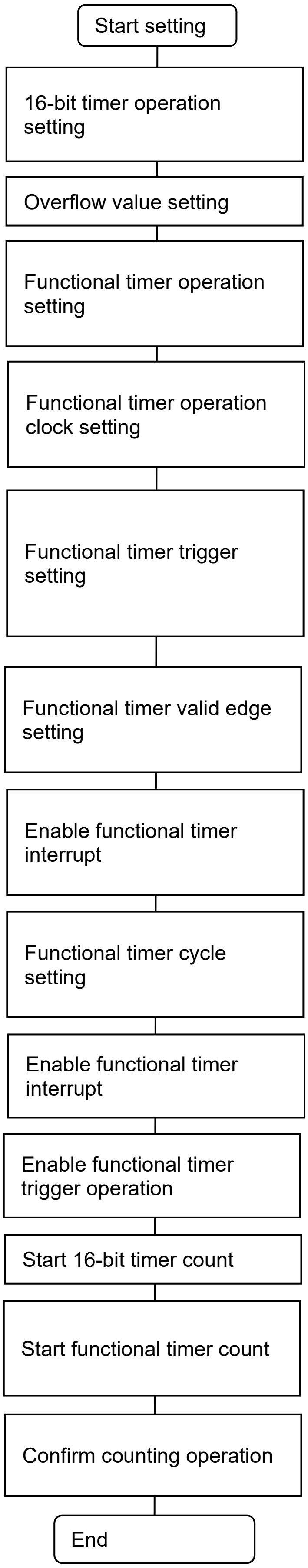
Figure 29-4 shows an example of the monitoring operation, using 16-bit timer 0 and Functional timer 0, for the high-speed clock (PLL oscillation circuit) oscillation.



TMH0C: 16-bit timer 0 counter register
TMH0INT: 16-bit timer 0 interrupt
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.


Start setting
16-bit timer operation setting
Overflow value setting
Functional timer operation setting
Functional timer operation clock setting
Functional timer trigger setting
Functional timer valid edge setting
Enable functional timer interrupt
Functional timer cycle setting
Enable functional timer interrupt
Enable functional timer trigger operation
Start 16-bit timer count
Start functional timer count
Confirm counting operation
End

Set the following items through the TMH0MOD register of the 16-bit timer 0.
- TH0OST bit=0 (initial value): Repeat mode
- TH08BM bit=0 (initial value): 16-bit timer mode
- TH0CS bit=0 (initial value): LSCLK0 is chosen for timer clock

Set the overflow value through the TMH0D register of the 16-bit timer 0.

Set the following items through the FT0MOD register of the functional timer 0.
- FT0MD1-0 bit=0 (initial value): TIMER mode
- FT0DTENP, FT0DTENN bits=00 (initial value): Dead time disabled
- FT0OST bit=0 (initial value): Repeat mode

Set the following items through the FT0CLK register of the functional timer 0.
- FT0CK0 bit=0 (initial value): HSCLK is chosen for operation clock
- Other bits=0

Set the following items through the FT0TRG0 register of the functional timer 0.
- FT0ST bit=1: Counter start through trigger event: Enabled
- FT0STC bit=1: Counter clear through trigger event for count start: Enabled
- FT0SP bit=1: Counter stop through trigger event: Enabled
- FT0STSS, FT0STS3 to 0 bits: TMH0INT is chosen for trigger event source
- Other bits="0"

Set the following items through the FT0TRG1 register of the functional timer 0.
- FT0TRM2 to 0 bits=000 (initial value): rising edge counter start, rising edge counter stop
- Other bits=0

Set the following items through the FT0INTE register of the functional timer 0.
- FT0IETS bit=1: Trigger counter stop interrupt: Enabled
- Other bits=0

Write "0FFFFH" to the FT0P register of the functional timer 0.
Through this writing the use of the counter can be maximized in the trigger start/stop mode.

Enable the functional timer 0 interrupt (FTM0INT) with the IE23 register.
(EFTM0 bit=1)

Set the following item through the FTCSTR register of the functional timer 0.
- Functional timer 0 trigger operation: Enabled (FT0ETG bit=1)

Start 16-bit timer 0 count through the TMHSTR register of the 16-bit timer 0.

The functional timer starts/stops counting using HSCLK (high-speed clock) through the overflow interrupt of the 16-bit timer which is counting using LSCLK (low-speed clock).

An interrupt is generated when the functional timer stops. Then read the value of the FT0C register to confirm if the high-speed clock is normally oscillating.

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.**

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 “23.3.2 Test function of 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.

Chapter 26 Flash Memory

26. Flash Memory

26.1 General Description

ML62Q2500 group has the flash memory in the program memory space and data flash area. For details of the program memory space and data flash area, see Chapter 2 "CPU and Memory Space".
The flash memory is programmable by following three ways.

- The ways of programming the flash memory

Programming method	Tool/Register/Communication	Reference Chapter
Programming by the on-chip debug function	On-chip debug emulator or other flash programmers	Chapter 28 "On chip Debug function"
Self-Programming by using the special function register(SFR)	Special Function Registers(SFRs) for programming the flash memory	Section 26.3 "Self-programming"
Programming by the ISP (In-System Programming) function	UART communication with an external device 3rd Party Flash programmers (*1)	Section 26.4 "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
ML62Q2500	ML62Q2502/2522/2532	64KByte	0x0:0000 to 0x0:FFFF	4KByte (128Byte x 32sector)	0x1F:0000 to 0x1F:0FFF
	ML62Q2504/2524/2534	128KByte	0x0:0000 to 0x1:FFFF		

- Program memory space and Data flash area Overview (Functions and Characteristics)

Item		Program memory space	Data flash area
Erasing and programming unit	Chip erasing (ISP only)	All area	All area
	Block erasing	16K byte	all area
	Sector erasing	1K byte	128 byte
	Programming	4 byte (32bit)	1 byte (8bit)
Erasing and programming time	Chip erasing (ISP only)	Max. 50ms	Max. 50ms
	Block erasing		
	Sector erasing		
	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

26.1.1 List of Pins

Programming by the ISP function uses the following pins.

Pin name	I/O	Function
TEST1_N	I	clock input for ISP
TEST0	I/O	data input/output for ISP

26.2 Register Description

26.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial Value
		Byte	Word			
0xF090	Flash address register	FLASHAL	FLASHA	R/W	8/16	0xFF
0xF091		FLASHAH		R/W	8	0xFF
0xF092	Flash data register 0	FLASHD0L	FLASHD0	R/W	8/16	0xFF
0xF093		FLASHD0H		R/W	8	0xFF
0xF094	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	-	-	-	-	-

26.2.2 Flash Address Register (FLASHA)

This is a SFR used to set the erasing and programming address.

Address :	0xF090 (FLASHAL/FLASHA), 0xF091 (FLASHAH)
Access :	R/W
Access size :	8/16 bit
Initial value :	0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FLASHA															
Byte	FLASHAH								FLASHAL							
Bit	FA15	FA14	FA13	FA12	FA11	FA10	FA9	FA8	FA7	FA6	FA5	FA4	FA3	FA2	FA1	FA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 0	FA15 to FA0	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.**

26.2.3 Flash Segment Register (FLASHSEG)

This is a SFR used to set the segment for erasing and programming the flash memory.

Address :	0xF09A (FLASHSEG)
Access :	R/W
Access size :	8 bit
Initial value :	0x10

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FLASHSEG							
Bit	-	-	-	-	-	-	-	-	-	-	-	FSEG4	FSEG3	FSEG2	FSEG1	FSEG0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 5	-	Reserved bits
4 to 0	FSEG4 to FSEG0	These bits are used to set the flash memory segment address.

Table 26-1 shows the address setting value for block erasing and Table 26-2 shows the address setting value for sector erasing.

Table 26-1 Address Setting Values for Block Erasing

Segment	Block	Address	Size	FLASHSEG register	FLASHA register
Segment 0	Block 0	0x0000 to 0x3FFF	16KByte	0x00	0x0000
	Block 1	0x4000 to 0x7FFF	16KByte		0x4000
	Block 2	0x8000 to 0xBFFF	16KByte		0x8000
	Block 3	0xC000 to 0xFFFF	16KByte		0xC000
Segment 1	Block 4	0x0000 to 0x3FFF	16KByte	0x01	0x0000
	Block 5	0x4000 to 0x7FFF	16KByte		0x4000
	Block 6	0x8000 to 0xBFFF	16KByte		0x8000
	Block 7	0xC000 to 0xFFFF	16KByte		0xC000
Segment 2	Block 8	0x0000 to 0x3FFF	16KByte	0x02	0x0000
	Block 9	0x4000 to 0x7FFF	16KByte		0x4000
	Block 10	0x8000 to 0xBFFF	16KByte		0x8000
	Block 11	0xC000 to 0xFFFF	16KByte		0xC000
Segment 3	Block 12	0x0000 to 0x3FFF	16KByte	0x03	0x0000
	Block 13	0x4000 to 0x7FFF	16KByte		0x4000
	Block 14	0x8000 to 0xBFFF	16KByte		0x8000
	Block 15	0xC000 to 0xFFFF	16KByte		0xC000
Segment 4	Block 16	0x0000 to 0x3FFF	16KByte	0x04	0x0000
	Block 17	0x4000 to 0x7FFF	16KByte		0x4000
	Block 18	0x8000 to 0xBFFF	16KByte		0x8000
	Block 19	0xC000 to 0xFFFF	16KByte		0xC000
Segment 5	Block 20	0x0000 to 0x3FFF	16KByte	0x05	0x0000
	Block 21	0x4000 to 0x7FFF	16KByte		0x4000
	Block 22	0x8000 to 0xBFFF	16KByte		0x8000
	Block 23	0xC000 to 0xFFFF	16KByte		0xC000
Segment 6	Block 24	0x0000 to 0x3FFF	16KByte	0x06	0x0000
	Block 25	0x4000 to 0x7FFF	16KByte		0x4000
	Block 26	0x8000 to 0xBFFF	16KByte		0x8000
	Block 27	0xC000 to 0xFFFF	16KByte		0xC000
Segment 7	Block 28	0x0000 to 0x3FFF	16KByte	0x07	0x0000
	Block 29	0x4000 to 0x7FFF	16KByte		0x4000
	Block 30	0x8000 to 0xBFFF	16KByte		0x8000
	Block 31	0xC000 to 0xFFFF	16KByte		0xC000
Segment 31	Block 0	0x0000 to 0x0FFF	4KByte	0x1F	0x0000

Table 26-2-1 Address Setting Values for Sector Erasing

Segment	Block	Address	Size	FLASHSEG register	FLASHA register
Segment 0	Sector 0	0x0000 to 0x03FF	1KByte	0x00	0x0000
	Sector 1	0x0400 to 0x07FF	1KByte		0x0400
	:	:	:		:
	Sector 62	0xF800 to 0xFBFF	1KByte		0xF800
	Sector 63	0xFC00 to 0xFFFF	1KByte		0xFC00
Segment 1	Sector 64	0x0000 to 0x03FF	1KByte	0x01	0x0000
	Sector 65	0x0400 to 0x07FF	1KByte		0x0400
	:	:	:		:
	Sector 126	0xF800 to 0xFBFF	1KByte		0xF800
	Sector 127	0xFC00 to 0xFFFF	1KByte		0xFC00
Segment 2	Sector 128	0x0000 to 0x03FF	1KByte	0x02	0x0000
	Sector 129	0x0400 to 0x07FF	1KByte		0x0400
	:	:	:		:
	Sector 190	0xF800 to 0xFBFF	1KByte		0xF800
	Sector 191	0xFC00 to 0xFFFF	1KByte		0xFC00
Segment 3	Sector 192	0x0000 to 0x03FF	1KByte	0x03	0x0000
	Sector 193	0x0400 to 0x07FF	1KByte		0x0400
	:	:	:		:
	Sector 254	0xF800 to 0xFBFF	1KByte		0xF800
	Sector 255	0xFC00 to 0xFFFF	1KByte		0xFC00
Segment 4	Sector 256	0x0000 to 0x03FF	1KByte	0x04	0x0000
	Sector 257	0x0400 to 0x07FF	1KByte		0x0400
	:	:	:		:
	Sector 318	0xF800 to 0xFBFF	1KByte		0xF800
	Sector 319	0xFC00 to 0xFFFF	1KByte		0xFC00
Segment 5	Sector 320	0x0000 to 0x03FF	1KByte	0x05	0x0000
	Sector 321	0x0400 to 0x07FF	1KByte		0x0400
	:	:	:		:
	Sector 382	0xF800 to 0xFBFF	1KByte		0xF800
	Sector 383	0xFC00 to 0xFFFF	1KByte		0xFC00
Segment 6	Sector 384	0x0000 to 0x03FF	1KByte	0x06	0x0000
	Sector 385	0x0400 to 0x07FF	1KByte		0x0400
	:	:	:		:
	Sector 446	0xF800 to 0xFBFF	1KByte		0xF800
	Sector 447	0xFC00 to 0xFFFF	1KByte		0xFC00
Segment 7	Sector 448	0x0000 to 0x03FF	1KByte	0x07	0x0000
	Sector 449	0x0400 to 0x07FF	1KByte		0x0400
	:	:	:		:
	Sector 510	0xF800 to 0xFBFF	1KByte		0xF800
	Sector 511	0xFC00 to 0xFFFF	1KByte		0xFC00

Table 26-2-2 Address Setting Values for Sector Erasing

Segment	Block	Address	Size	FLASHSEG register	FLASHA register
Segment 31	Sector 0	0x0000 to 0x007F	128Byte	0x1F	0x0000
	Sector 1	0x0080 to 0x00FF	128Byte		0x0080
	Sector 2	0x0100 to 0x017F	128Byte		0x0100
	Sector 3	0x0180 to 0x01FF	128Byte		0x0180
	:	:	:		:
	Sector 12	0x0600 to 0x067F	128Byte		0x0600
	Sector 13	0x0680 to 0x06FF	128Byte		0x0680
	Sector 14	0x0700 to 0x077F	128Byte		0x0700
	Sector 15	0x0780 to 0x07FF	128Byte		0x0780
	:	:	:		:
	Sector 28	0x0E00 to 0x0E7F	128Byte		0x0E00
	Sector 29	0x0E80 to 0x0EFF	128Byte		0x0E80
	Sector 30	0x0F00 to 0x0F7F	128Byte		0x0F00
	Sector 31	0x0F80 to 0x0FFF	128Byte		0x0F80

26.2.4 Flash Data Register 0 (FLASHD0)

This is a SFR used to set programming data.

Address : 0xF092 (FLASHD0L/FLASHD0), 0xF093 (FLASHD0H)
Access : R/W
Access size : 8/16 bit
Initial value : 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	FLASHD0															
Byte	FLASHD0H								FLASHD0L							
Bit	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 8	FD15 to FD8	These bits are used to set the 2nd byte data.
7 to 0	FD7 to FD0	These bits are used to set the 1st byte data.

26.2.5 Flash Data Register 1 (FLASHD1)

This is a SFR used to set programming data.

Address : 0xF094 (FLASHD1L/FLASHD1), 0xF095 (FLASHD1H)
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	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	How to start programming to flash	Description
Program memory space	Four bytes specified in FLASHD0 register(FLASHD0H, FLASHD0L) and FLASHD1 register(FLASHD1H, FLASHD1L)	Writing data into FLASHD1/FLASHD1H	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.	Writing data into FLASHD0/FLASHD0L	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.**

- **During programming data-flash, a CPU can execute instruction by the back ground operation function; BGO. Confirm FDPRSTA bit of FLASHSTA register for complition of programming.**
- **Erase data in the addresses in advance. Programmed data without erase is unguaranteed.**
- **Do not read or program unused areas to prevent the CPU works incorrectly.**

26.2.6 Flash Control Register (FLASHCON)

This is a write-only SFR used to control the block erasing and sector erasing for the flash memory. This register always returns 0x00 for reading.

Address :	0xF096 (FLASHCON)
Access :	W
Access size :	8 bit
Initial value :	0x00

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

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1, 0	FSERS, FERS	These bits are used to start the sector erasing or block erasing. Setting the FSERS bit to "1" starts erasing the sector and setting the FERS bit to "1" starts erasing the block specified by the FLASHSEG and FLASHAH register. 00: No function (Initial value) 01: Start block erasing 10: Start sector erasing 11: Do not use (No function)

26.2.7 Flash Acceptor (FLASHACP)

This is a write-only SFR used to accept for erasing/programming the flash memory.

Address :	0xF098 (FLASHACP)
Access :	W
Access size :	8 bit
Initial value :	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	-															
Byte	-								FLASHACP							
Bit	-	-	-	-	-	-	-	-	fac7	fac6	fac5	fac4	fac3	fac2	fac1	fac0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These bits are used to accept for erasing/programming the flash memory in order to prevent an unintended erasing/programming operation.
When "0xFA" and "0xF5" are written to the FLASHACP in this order, the erasing or programming function is enabled only once. For subsequent erasing or programming, "0xFA" and "0xF5" must be written to FLASHACP each time.

Even if other instructions are executed between the instruction that writes "0xFA" and "0xF5" to the FLASHACP, the erasing or programming function is still valid.
If data other than "0xF5" is written to the FLASHACP after "0xFA" is written, "0xFA" becomes invalid. In this case, it needs to write "0xFA" again.

[Note]

- **A flash memory data in processing to program is not guaranteed, if this register is written any data when FLASHSTA is not 0x0.**

26.2.8 Flash Self Register (FLASHSLF)

This is a SFR used to enable erasing and programming the flash memory. When system clock is the low-speed clock, it is not writable.

Address :	0xF09C (FLASHSLF)
Access :	R/W
Access size :	8 bit
Initial value :	0x00

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

Bit No.	Bit symbol name	Description
7 to 1	-	Reserved bits
0	FSELF	This bit is used to enable erasing and programming the flash memory. This bit is kept after completed erasing/programming. 0: Disabled (Initial value) 1: Enabled

[Note]

- **A flash memory data in processing to program is not guaranteed, if this register is written any data when FLASHSTA is not 0x0.**

26.2.9 Flash Status Register (FLASHSTA)

This is a read-only SFR used to check status of the flash memory.

Address :	0xF09E (FLASHSTA)
Access :	R
Access size :	8 bit
Initial value :	0x00

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

Bit No.	Bit symbol name	Description
7 to 2	-	Reserved bits
1	FDPRSTA	This bit is used to indicate whether the data flash area is in the state of programming. 0: Not in the state of programming (Initial value) 1: In the state of programming
0	FDERSTA	This bit is used to indicate whether the data flash area is in the state of erasing. 0: Not in the state of erasing (Initial value) 1: In the state of erasing

This register is used when erasing or programming the data flash memory.

Erasing/Programming target	Availability to read this register while erasing/programming	Description
Program memory space	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". Do not perform the erasing or programming when either the FDERSTA bit or the FDPRSTA bit is "1".**

26.3 Self-programming

The self-programming is the function to program (erase and program) the program memory space and data flash area using SFRs.
Table 26-3 shows the self-programming specifications for each of the program memory space and data flash area.

Table 26-3 Self-programming of Program Memory Space and Data Flash Area

		Program memory space (Segment 0 to 7)	Data flash area (segment 31)
Programming unit	Erasing block	16 Kbyte	all area
	Erasing sector	1 Kbyte	128 byte
	Programming	4 byte	1 byte
CPU operation during block/sector erase or program		Stop program processing (after completion of erasing/programming, resume program processing)	Continue program processing through the background operation (BGO) function
Confirmation of end of block/sector erasing or programming		Confirmation not required (as program run is stopped during erasing/programming)	Confirmation can be made through FLASHSTA register
Target area where block/sector erasing has been applied		Every bit becomes "1" (the bit written with "0" by writing becomes "0" from "1")	
Note on data programming		Erase the area to be reprogrammed (data programmed without erasing is unguaranteed)	
Function to prevent unintended erasing/programming		Flash self-register (FLASHSLF) and flash acceptor (FLASHACP) incorporated (*1)	
Flash memory erasing/programming		Supported only when system clock is the high-speed clock (*2)	
Note on user program programming		Before programming the user program, prepare a program for self-programming in the program code area which is not erased/reprogrammed	-
Remapping function		User program update, etc. can be performed by simultaneously using remapping function	-

*1: After the programming is enabled by the FLASHSLF register, if "0xFA" and "0xF5" are written to the flash acceptor (FLASHACP), block/sector erase or reprogram is enabled only once.
*2: See Chapter 6 "Clock Generation Circuit" for enabling oscillation of the high-speed oscillation circuit and switching the system clock.

26.3.1 Notes on Debugging Self-programming Code

When debugging the area within the scope of program for self-programming (from setting the flash acceptor to writing the flash data register 0, 1) using U16 development environment (debugger), use the debugger according to the precautions described in Table 26-4.

Table 26-4 Notes on Debugging Self-programming

Limited function	Notes
Breakpoint setting	Do not perform the real time execution with break points set in the scope of program for self-programming (from setting the flash acceptor to setting the flash data register0, 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.

26.3.2 Programming Program Memory Space

In the program memory space (flash memory), block erase in units of 16 Kbytes, sector erase in units of 1 Kbyte, and reprogram in units of 4 bytes can be executed.
Figure 26-1 shows the flow diagram for erasing the program memory space.

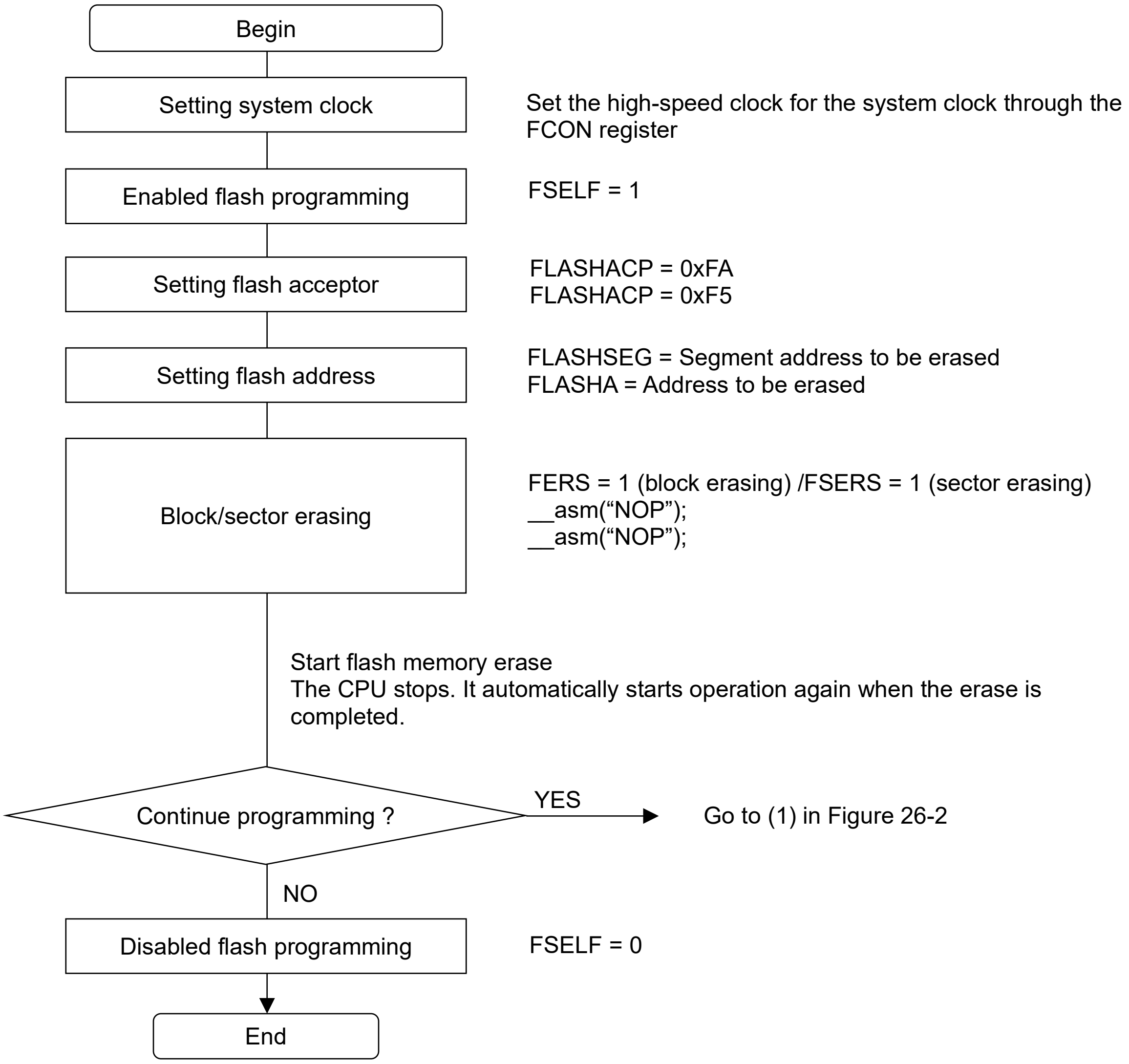

Begin
Setting system clock
Set the high-speed clock for the system clock through the FCON register
Enabled flash programming
FSELF = 1
Setting flash acceptor
FLASHACP = 0xFA
FLASHACP = 0xF5
Setting flash address
FLASHSEG = Segment address to be erased
FLASHA = Address to be erased
Block/sector erasing
FERS = 1 (block erasing) /FSERS = 1 (sector erasing)
__asm("NOP");
__asm("NOP");
Start flash memory erase
The CPU stops. It automatically starts operation again when the erase is completed.
Continue programming ?
YES
Go to (1) in Figure 26-2
NO
Disabled flash programming
FSELF = 0
End

Figure 26-1 Flow Diagram for Erasing Program Memory Space

[Note]
- **Only erase areas irrelevant to program processing. If erasing the area where program processing is in progress, the LSI works incorrectly.**
- **During block/sector erasing, the CPU stops the operation for maximum 50 ms whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.**
- **For block/sector erasing, place two NOP instructions following the instruction used to set FERS/FSERS bits of the FLASHCON register to "1".**

Figure 26-2 shows the flow diagram for programming the program memory space.

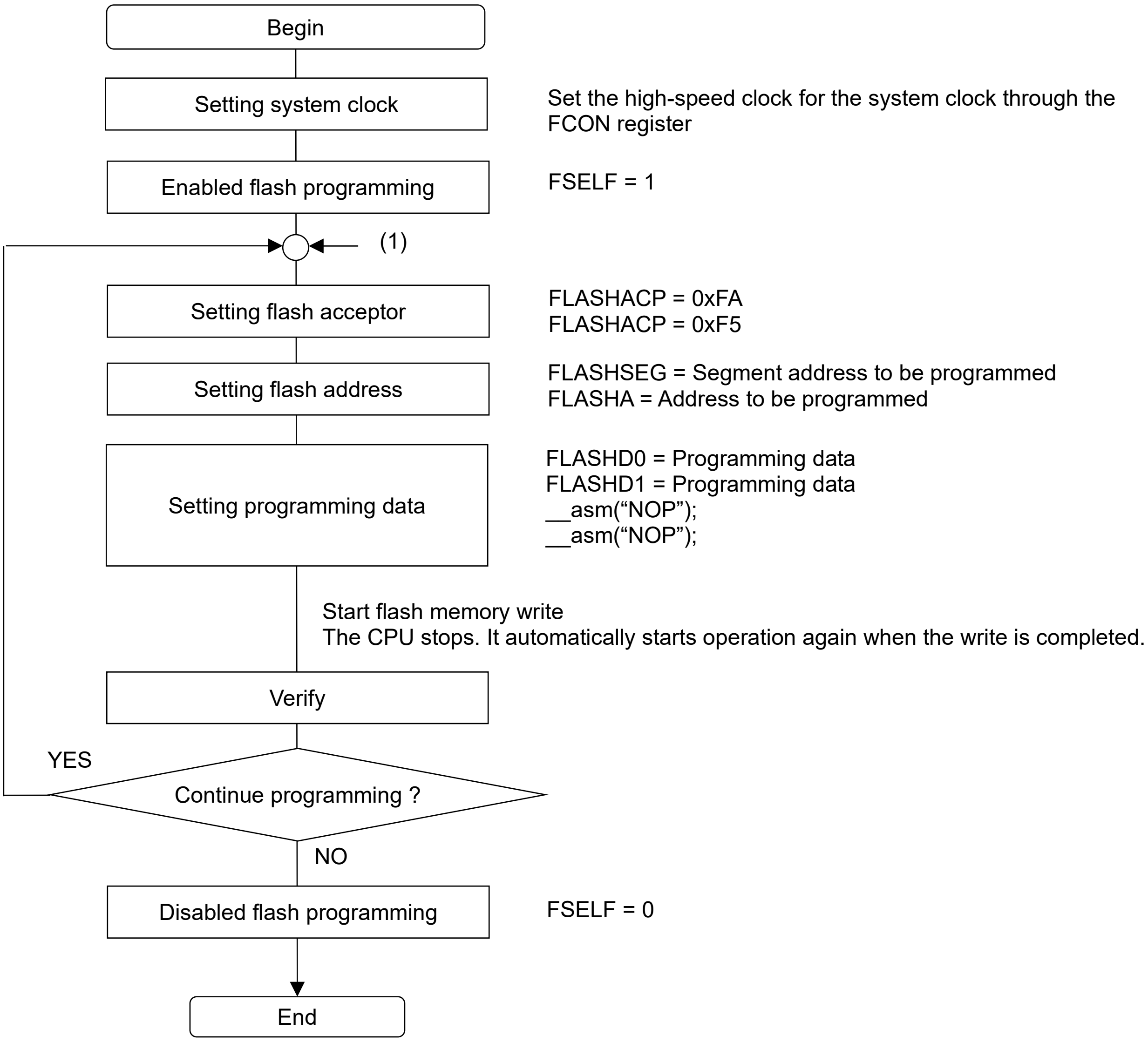

Begin
Setting system clock
Set the high-speed clock for the system clock through the FCON register
Enabled flash programming
FSELF = 1
(1)
Setting flash acceptor
FLASHACP = 0xFA
FLASHACP = 0xF5
Setting flash address
FLASHSEG = Segment address to be programmed
FLASHA = Address to be programmed
Setting programming data
FLASHD0 = Programming data
FLASHD1 = Programming data
__asm("NOP");
__asm("NOP");
Start flash memory write
The CPU stops. It automatically starts operation again when the write is completed.
Verify
YES
Continue programming ?
NO
Disabled flash programming
FSELF = 0
End

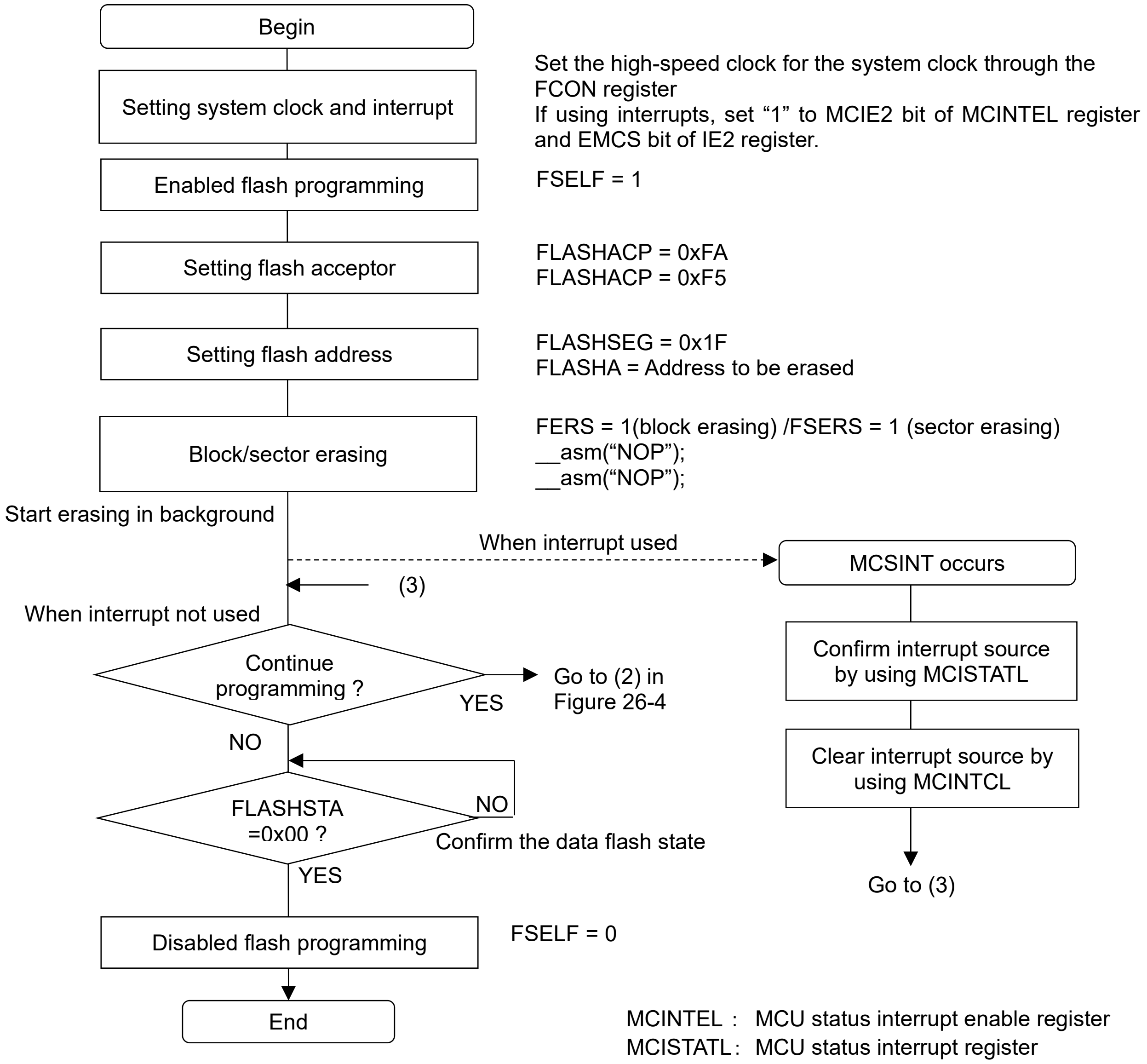
Figure 26-2 Flow Diagram for Programming Program Memory Space

[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.**

26.3.3 Programming Data Flash Area

In the data flash area (flash memory), block erase in all area, sector erase in units of 128 bytes, and programming in units of 1 byte can be executed. During block/sector erase or program in the data flash area, the CPU continues program processing using the background operation (BGO) function.
Figure 26-3 shows the flow diagram for erasing the data flash area.


Begin
Setting system clock and interrupt
Set the high-speed clock for the system clock through the FCON register
If using interrupts, set “1” to MCIE2 bit of MCINTEL register and EMCS bit of IE2 register.
Enabled flash programming
FSELF = 1
Setting flash acceptor
FLASHACP = 0xFA
FLASHACP = 0xF5
Setting flash address
FLASHSEG = 0x1F
FLASHA = Address to be erased
Block/sector erasing
FERS = 1(block erasing) /FSERS = 1 (sector erasing)
__asm(“NOP”);
__asm(“NOP”);
Start erasing in background
When interrupt used
MCSINT occurs
(3)
When interrupt not used
Continue programming ?
YES
Go to (2) in Figure 26-4
NO
Confirm interrupt source by using MCISTATL
Clear interrupt source by using MCINTCL
FLASHSTA =0x00 ?
NO
Confirm the data flash state
YES
Go to (3)
Disabled flash programming
FSELF = 0
End

MCINTEL : MCU status interrupt enable register
MCISTATL: MCU status interrupt register
MCINTCL : MCU status interrupt clear register
IE2 : Interrupt enable register 2

See Chapter 5 "Interrupts" and Sections 29.2.7 to 29.2.9 in Chapter 29 "Safety Function" for using interrupts.

Figure 26-3 Flow Diagram for Erasing Data Flash Area

[Note]

- **The CPU continues program processing even while data flash erasing is in progress. An entering to the STOP/STOP-D/HALT-D/HALT-H mode is not available during the erasing. In addition, set the FSELF bit of the FLASHSLF register to "0" after the erasing is completed.**
- **The data flash area is unreadable during erasing.**
- **For block/sector erasing, place two NOP instructions following the instruction used to set FERS/FSERS bits of the FLASHCON register to "1".**

Figure 26-4 shows the flow diagram for programming the data flash area.

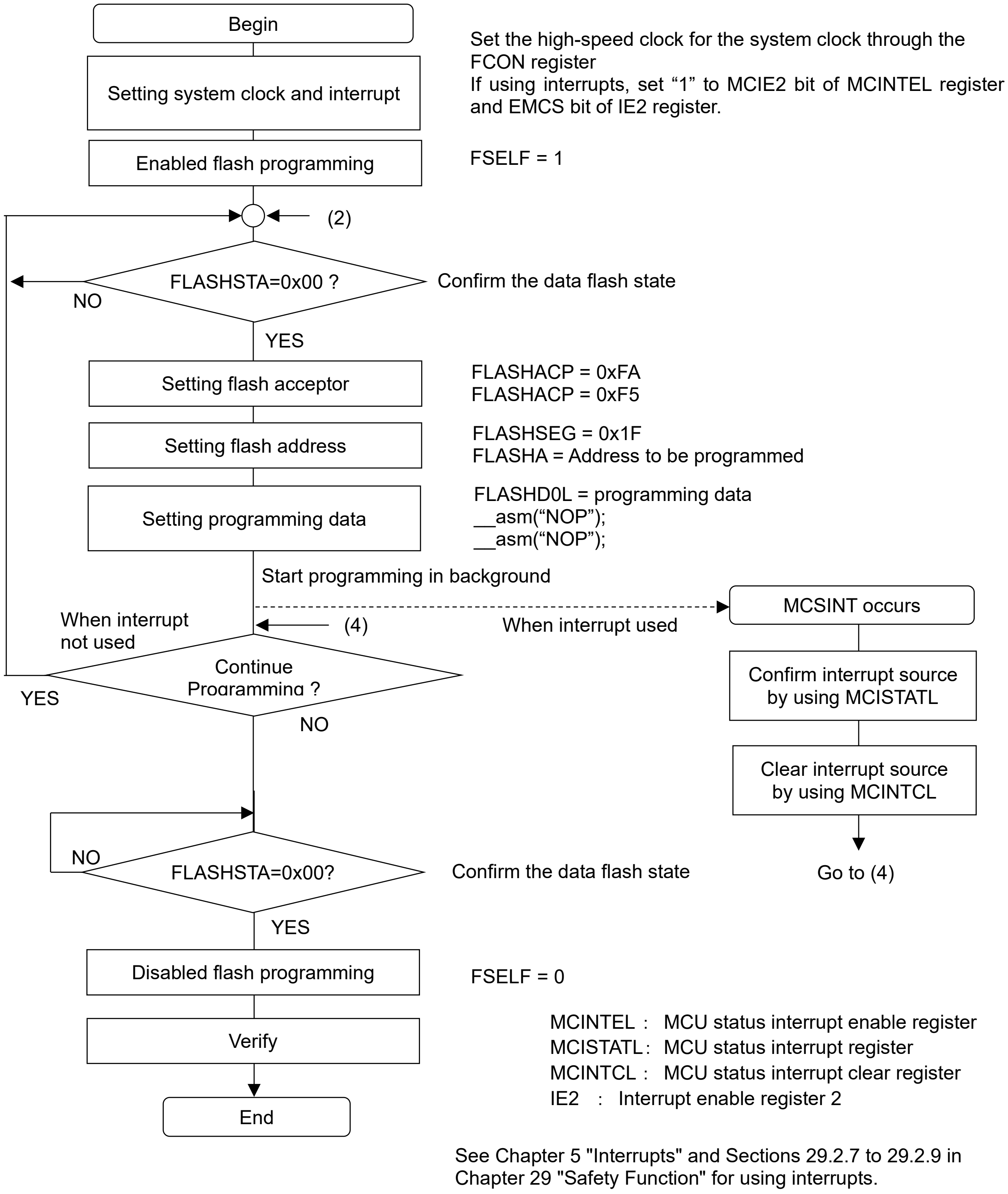

Begin
Setting system clock and interrupt
Set the high-speed clock for the system clock through the FCON register
If using interrupts, set “1” to MCIE2 bit of MCINTEL register and EMCS bit of IE2 register.
Enabled flash programming
FSELF = 1
(2)
FLASHSTA=0x00 ?
Confirm the data flash state
NO
YES
Setting flash acceptor
FLASHACP = 0xFA
FLASHACP = 0xF5
Setting flash address
FLASHSEG = 0x1F
FLASHA = Address to be programmed
Setting programming data
FLASHD0L = programming data
__asm(“NOP”);
__asm(“NOP”);
Start programming in background
MCSINT occurs
When interrupt not used
(4)
When interrupt used
Continue Programming ?
YES
NO
Confirm interrupt source by using MCISTATL
Clear interrupt source by using MCINTCL
NO
FLASHSTA=0x00?
Confirm the data flash state
Go to (4)
YES
Disabled flash programming
FSELF = 0
Verify
End
MCINTEL : MCU status interrupt enable register
MCISTATL : MCU status interrupt register
MCINTCL : MCU status interrupt clear register
IE2 : Interrupt enable register 2
See Chapter 5 "Interrupts" and Sections 29.2.7 to 29.2.9 in Chapter 29 "Safety Function" for using interrupts.

Figure 26-4 Flow Diagram for Programming Data Flash Area

[Note]

- **The CPU continues program processing even while data flash programming is in progress. An entering to the STOP/STOP-D/HALT-D/HALT-H mode is not available 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.**

26.3.4 Notes on use of self-programming

Table 26-5 shows the notes on the use of self-programming (block erasing/sector erasing/programming).

Table 26-5 Notes on Use of Self-programming

Item	Notes
System clock during use of self-programming	Set to high-speed clock. See Chapter "6 Clock Generation Circuit" for enabling the high-speed clock oscillation and switching the system clock.
If power outage or forced termination due to a reset occurs	Data in flash memory is not guaranteed. Perform block/sector erase again then program data.
If LSI does not start up due to occurrence of power outage or forced termination during programming (*1)	Program the program again using on-chip debug emulator or ISP function.
Access to SFRs related flash control.	Do not perform to write to the FLASHACP/FLASHSLF register during self-programming; when FLASHSTA is not 0x0.

*1: While programming the block or sector including address 0:0000 of the program area is in progress.

26.4 In-System Programming Function

The In-System Programming (ISP) function is used to program a program memory space or data flash area through 2-wired synchronous serial port communication with an external device.

26.4.1 Programming Procedure

Figure 26-5 shows the flow diagram for programming the flash memory using the ISP function.

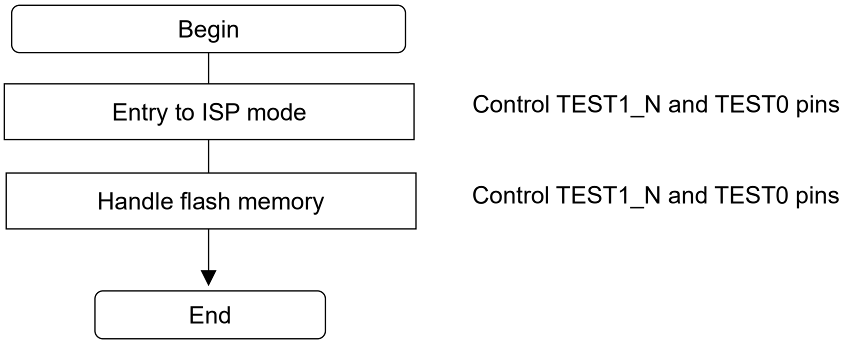

Begin
Entry to ISP mode
Control TEST1_N and TEST0 pins
Handle flash memory
Control TEST1_N and TEST0 pins
End

Figure 26-5 Flow Diagram for Programming Flash Memory Using ISP Function

26.4.2 Communication Method

Table 26-6 describes the communication method of the ISP function.

Table 26-6 ISP Function Communication Method

Item	Description
Communication Method	2-wired synchronous serial port
Data format	8-bit length, MSB first
Baud rate	1.5 to 2.5MHz

26.4.3 Control Command

3-byte commands are used to make the communication in the ISP function.
Table 26-7 shows the ISP mode commands.

Table 26-7 ISP Mode Command List

Command		1st byte	2nd byte	3rd byte
Entry	Mode entry command (1)	0xAB	0x63	0x59
	Mode entry command (2)	0xA4	0x55	0x0D
	Entry confirmation command *1	0x01	(Read) 0x06	(Read) 8’b111x_xxxx
Initial setting	Initial setting command (1)	0x1A	0x00	0x08
	Initial setting command (2)	0x1A	0x00	0x00
	Initial setting command (3)	0xC0	0x00	0x01
	Initial setting command (4)	0xC0	0x00	0x05
	Initial setting command (5)	0xC0	0x00	0x03
	Initial setting command (6)	0xCE	0x00	0x01
	Initial setting command (7)	0xCE	0x00	0x00
	Initial setting command (8)	0x96	0xFF	0xFF
	Initial setting command (9)	0x98	0xFF	0xFF
	Initial setting command (10)	0x9A	0xFF	0xFF
	Initial setting command (11)	0x9C	0xFF	0xFF
	Initial setting command (12)	0x9E	0xFF	0xFF
	Initial confirmation command (1) *1	0x01	(Read) 0x06	(Read) 0xC0
	Initial confirmation command (2) *1	0x91	(Read) 0x00	(Read) 0x00
Common setting	Segment setting command	0xC6	0x00	(segment value) 0x00-0x1F
	Address setting command	0xC8	Higher 8 bits	Lower 8 bits
	BUSY confirmation command *1	0xC5	(Read) 0x00: IDLE other :not IDLE	(Read) 0x1F
Block erasing	Block erasing command	0xC2	0x00	0x05
Chip erasing	Chip erasing command	0xC2	0x00	0x06
For programing data	Data setting command H ; in program code area (higher 2bytes)	0xD2	Higher Byte	Lower Byte
	Data setting command L ; in program code area (lower 2bytes)	0xCA	Higher Byte	Lower Byte
	Data setting command D ; in data flash area	0xCA	0xFF	1Byte data
	Programming command	0xC2	0x00	0x04
For verify	Expected data setting command H ; in program code area (higher 2bytes)	0xE4	Higher Byte	Lower Byte
	Expected data setting command L ; in program code area (lower 2bytes)	0xE2	Higher Byte	Lower Byte
	Expected data setting command D ; in data flash area	0xE4	0xFF	1Byte data
	Verify command	0xC2	0x00	0x02
	Verify confirmation command *1 ; collation result of expected data	0xE7	(Read) 0x00	(Read) 0x03 : OK 0x01 : OK at current cycle, but has been NG in the past cycles. 0x00 or 0x02: NG
	Stack clear command 1	0xD2	0x00	0x00
	Stack clear command 2	0xCA	0x00	0x00
	Stack clear command 3	0xE6	0x00	0x03

[Note]

- **Accessing to the program code area is performed in units of four bytes. Set four byte boundaries (0H/4H/8H/CH) for lower four bits of the address. Accessing to the data flash area is performed in units of one byte.**
- **All commands except some confirmation commands (*1) are reflected when a next command is sent.**

26.4.3.1 Command Timing

See data sheet for AC specifications.
Transmit one command within 80 us interval. Retry communication after 200 us as an interval time if communication is However, it have timeout function. The timeout is to quit the ISP mode if a specific command is not received for a certain period of time. when in the ISP mode.
Transmit one of following command within 800 ms interval in the ISP mode.

- Initial setting command (1) + any commands
- Expected data setting command H + any commands
- Expected data setting command D + any commands
- BUSY confirmation command

The execution timing of the command except entry and confirming commands (mode entry command (1)(2), entry confirmation command, initial confirmation command (1)(2), BUSY confirmation command, verify confirm command), is at first positive edge of TEST1_N of next command.

26.4.4 How to Entry ISP Mode

Figure 26-6 shows flow diagram and timing diagram to entry ISP mode.

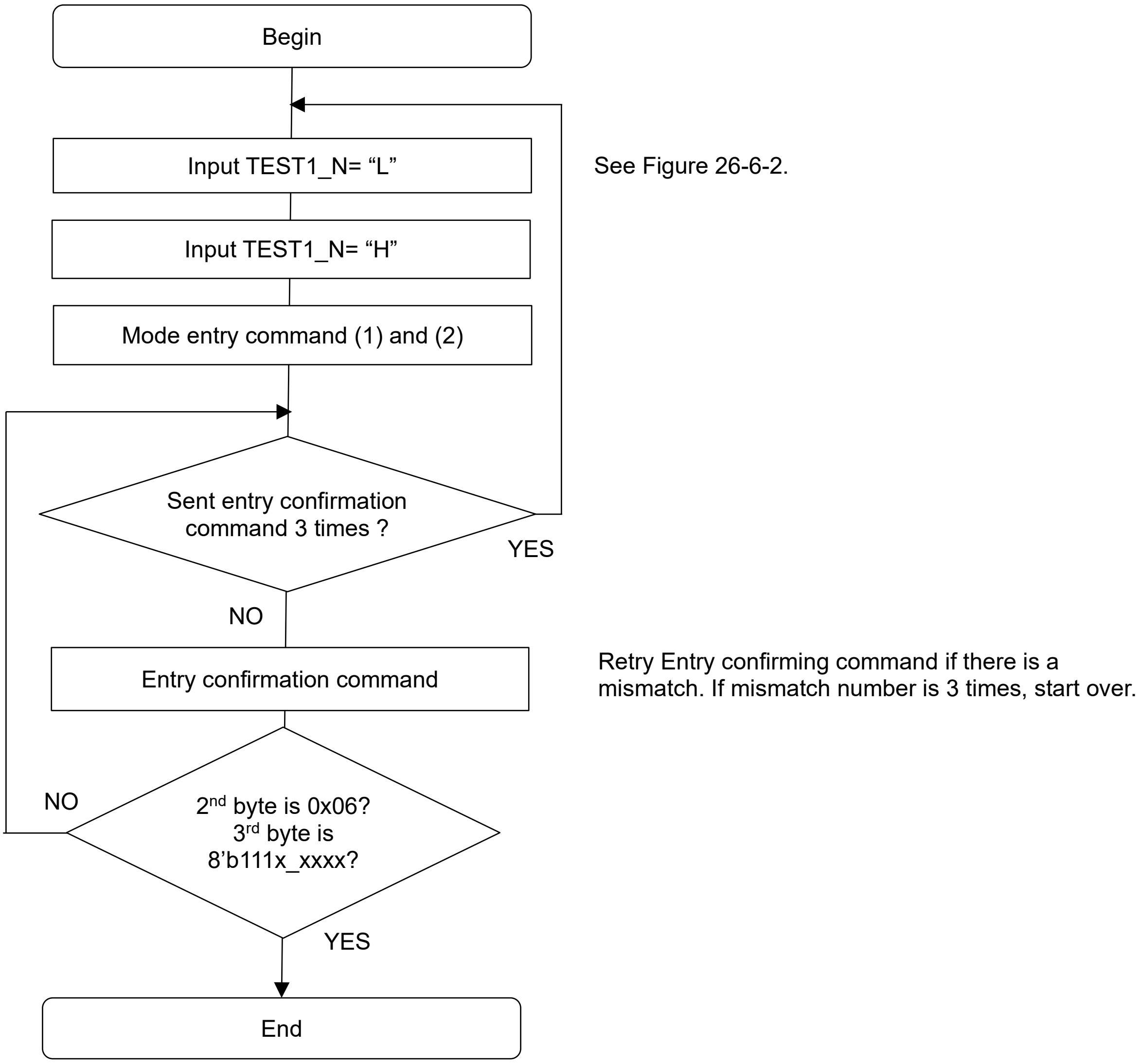

Begin
Input TEST1_N= "L"
See Figure 26-6-2.
Input TEST1_N= "H"
Mode entry command (1) and (2)
Sent entry confirmation command 3 times ?
YES
NO
Entry confirmation command
Retry Entry confirming command if there is a mismatch. If mismatch number is 3 times, start over.
NO
2nd byte is 0x06?
3rd byte is
8'b111x_xxxx?
YES
End

Figure 26-6-1 Flow diagram to entry ISP mode

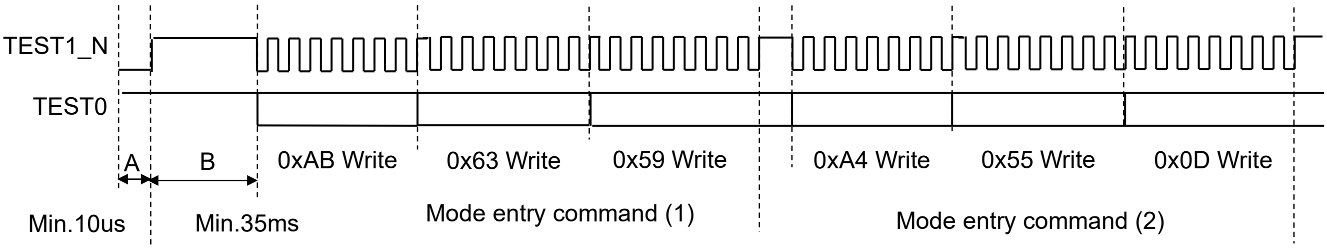


Figure 26-6-2 Timing diagram to entry ISP mode

26.4.5 Handling the Flash Memory

Figure 26-7 shows the flow diagram for erasing/programming the flash memory after transition to the ISP mode.

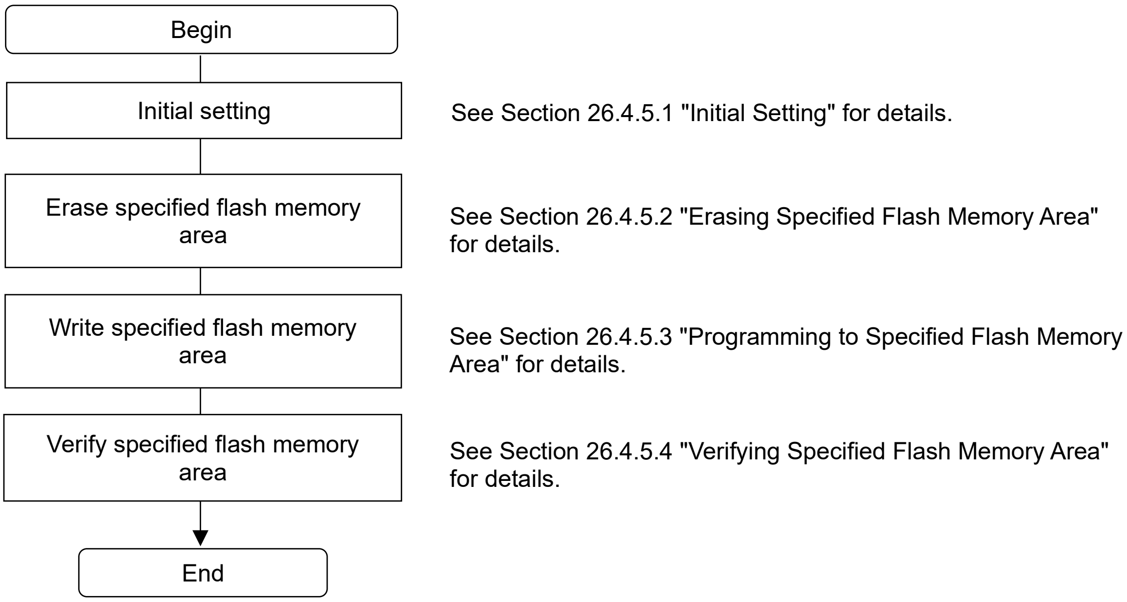

Begin
Initial setting
See Section 26.4.5.1 "Initial Setting" for details.
Erase specified flash memory area
See Section 26.4.5.2 "Erasing Specified Flash Memory Area" for details.
Write specified flash memory area
See Section 26.4.5.3 "Programming to Specified Flash Memory Area" for details.
Verify specified flash memory area
See Section 26.4.5.4 "Verifying Specified Flash Memory Area" for details.
End

Figure 26-7 Flow Diagram for Erasing/Programming Flash Memory (Overview)

26.4.5.1 Initial Setting

Figure 26-8 shows the initial setting flow.

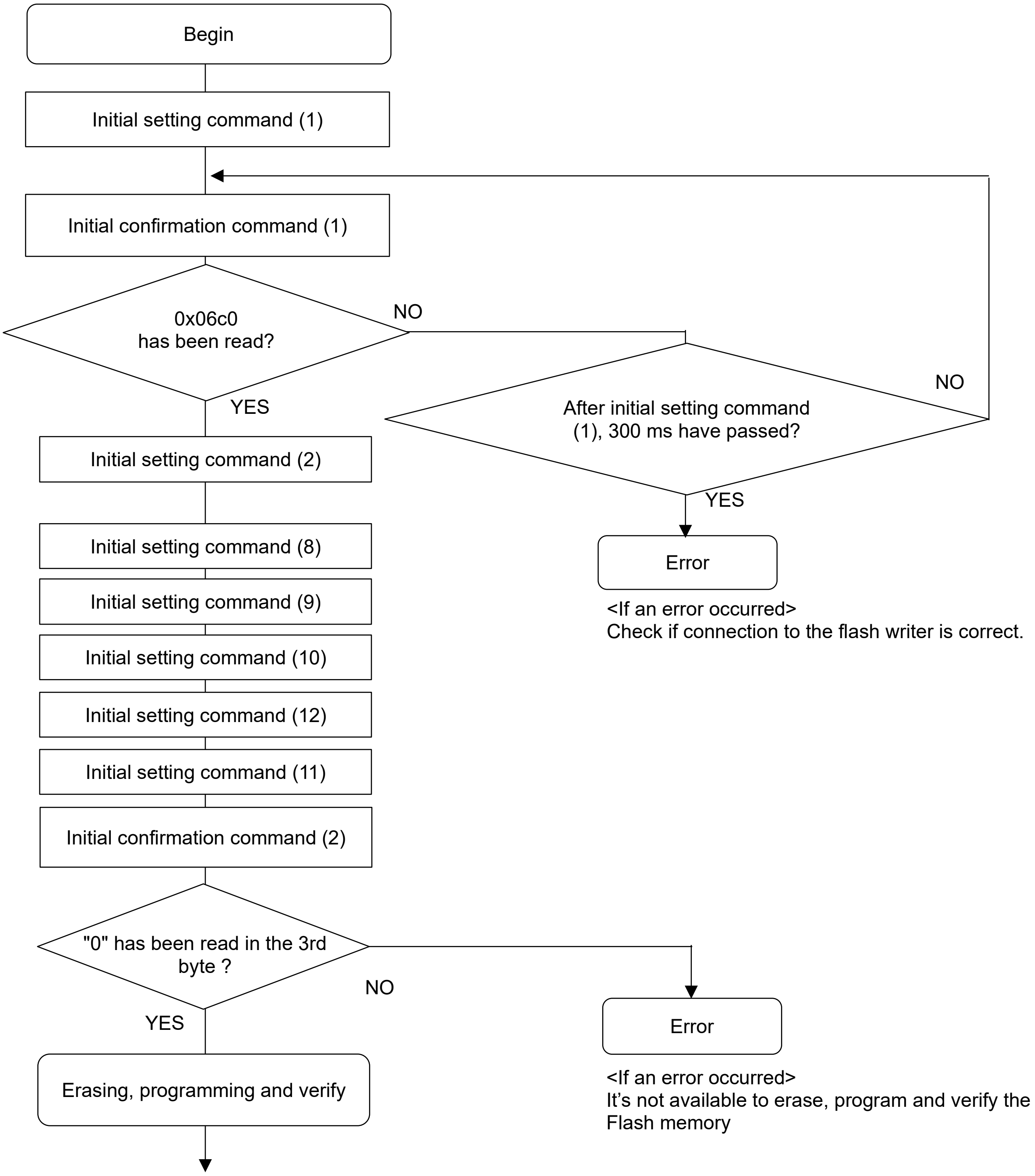

Begin
Initial setting command (1)
Initial confirmation command (1)
0x06c0
has been read?
NO
YES
After initial setting command
(1), 300 ms have passed?
NO
YES
Error
<If an error occurred>
Check if connection to the flash writer is correct.
Initial setting command (2)
Initial setting command (8)
Initial setting command (9)
Initial setting command (10)
Initial setting command (12)
Initial setting command (11)
Initial confirmation command (2)
"0" has been read in the 3rd
byte ?
NO
YES
Error
<If an error occurred>
It's not available to erase, program and verify the
Flash memory
Erasing, programming and verify

Figure 26-8 Initial Setting Flow Diagram

[Note]

- **Transmit command to avoid a timeout. See Section “26.4.3.1 Command Timing”.**

26.4.5.2 Erasing Data in Specified Flash Memory Area

Figure 26-9 shows the flow diagram for erasing data in specified flash memory area.

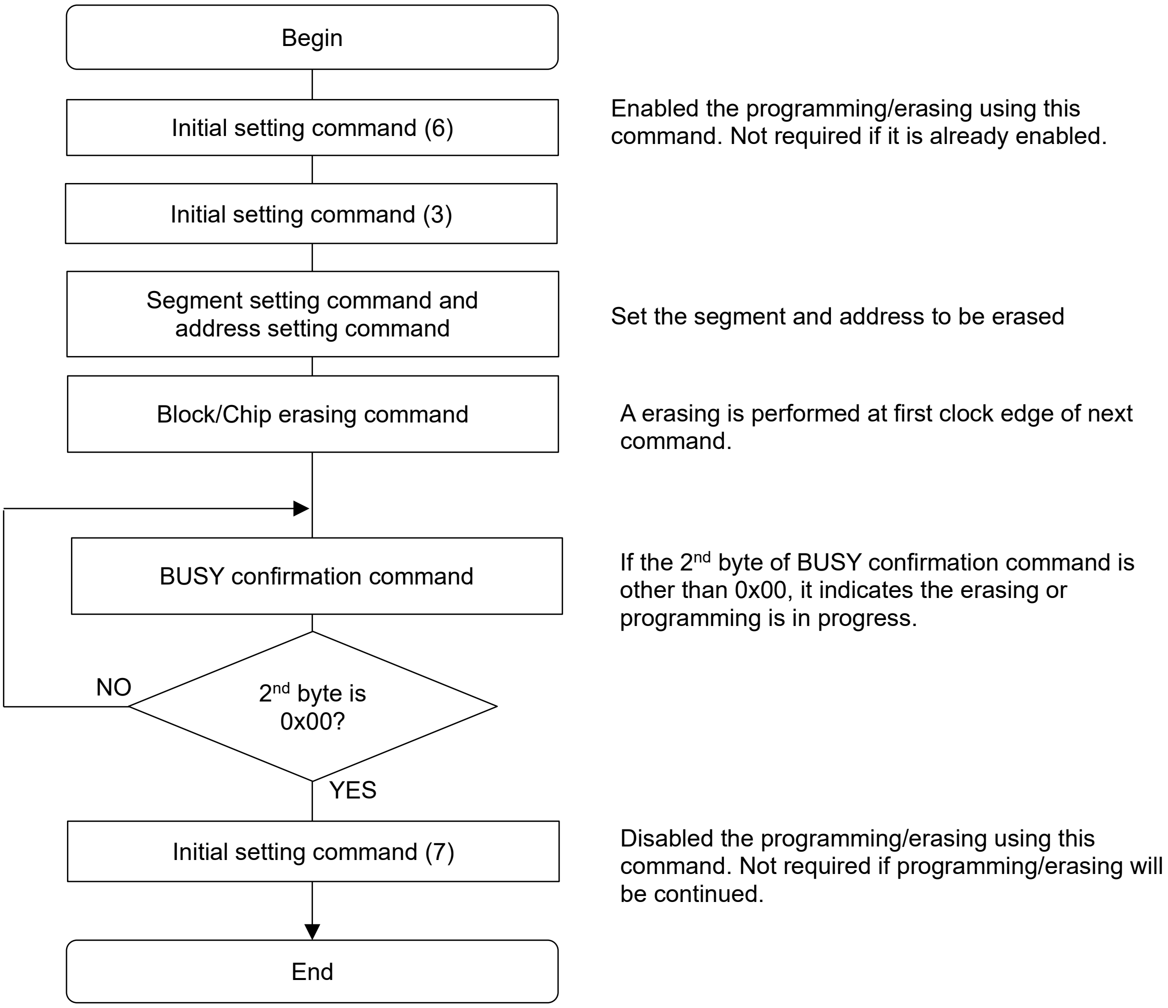

Begin
Initial setting command (6)
Enabled the programming/erasing using this command. Not required if it is already enabled.
Initial setting command (3)
Segment setting command and address setting command
Set the segment and address to be erased
Block/Chip erasing command
A erasing is performed at first clock edge of next command.
BUSY confirmation command
If the 2nd byte of BUSY confirmation command is other than 0x00, it indicates the erasing or programming is in progress.
NO
2nd byte is 0x00?
YES
Initial setting command (7)
Disabled the programming/erasing using this command. Not required if programming/erasing will be continued.
End

Figure 26-9 Flow Diagram for Erasing in Specified Flash Memory Area

[Note]

- **Transmit command to avoid a timeout. See Section “26.4.3.1 Command Timing”.**
- **Transmit any command after ‘initial setting command (7)’ if other command will not be transmit.**

26.4.5.3 Programming to Specified Flash Memory Area

Figure 26-10 shows the flow diagram for programming to the specified flash memory area.

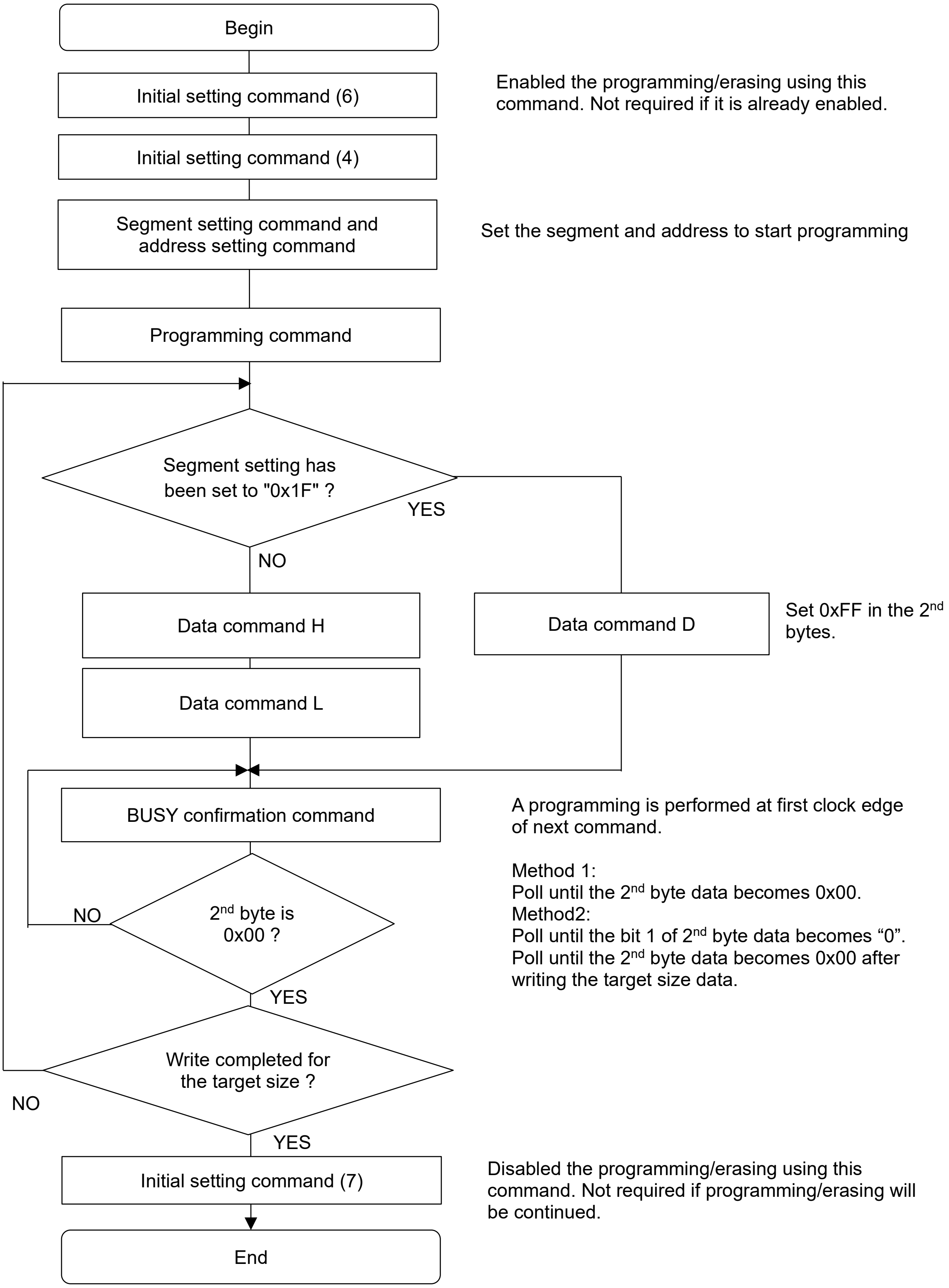

Begin
Initial setting command (6)
Enabled the programming/erasing using this command. Not required if it is already enabled.
Initial setting command (4)
Segment setting command and address setting command
Set the segment and address to start programming
Programming command
Segment setting has been set to "0x1F" ?
YES
NO
Data command H
Data command D
Set 0xFF in the 2nd bytes.
Data command L
BUSY confirmation command
A programming is performed at first clock edge of next command.
Method 1:
Poll until the 2nd byte data becomes 0x00.
Method2:
Poll until the bit 1 of 2nd byte data becomes “0”.
Poll until the 2nd byte data becomes 0x00 after writing the target size data.
NO
2nd byte is 0x00 ?
YES
Write completed for the target size ?
NO
YES
Initial setting command (7)
Disabled the programming/erasing using this command. Not required if programming/erasing will be continued.
End

Figure 26-10 Flow Diagram for Programming to Specified Flash Memory Area

[Note]

- **Transmit command to avoid a timeout. See Section “26.4.3.1 Command Timing”.**
- **Transmit any command after ‘initial setting command (7)’ if other command will not be transmit.**

26.4.5.4 Verifying data in Specified Flash Memory Area

Figure 26-11 shows the flow diagram for verifying data in the specified flash memory area.

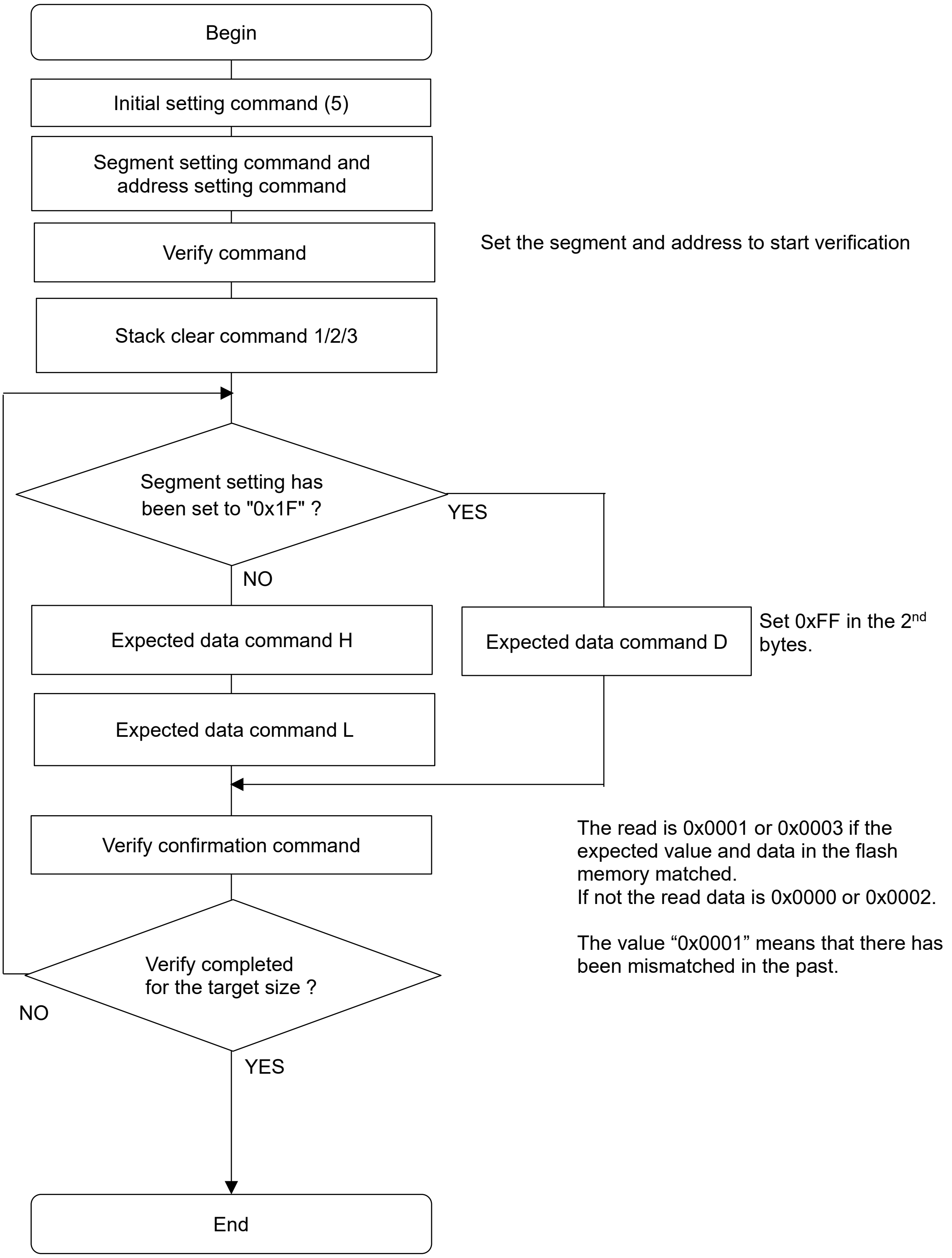

Begin
Initial setting command (5)
Segment setting command and address setting command
Verify command
Set the segment and address to start verification
Stack clear command 1/2/3
Segment setting has been set to "0x1F" ?
YES
NO
Expected data command H
Expected data command D
Set 0xFF in the 2nd bytes.
Expected data command L
Verify confirmation command
The read is 0x0001 or 0x0003 if the expected value and data in the flash memory matched.
If not the read data is 0x0000 or 0x0002.
The value "0x0001" means that there has been mismatched in the past.
Verify completed for the target size ?
NO
YES
End

Figure 26-11 Flow Diagram for Verifying Specified Flash Memory Area

[Note]

- **Transmit command to avoid a timeout. See Section "26.4.3.1 Command Timing".**

26.4.6 Advanced Control of Flash Memory Erasing/Programming

This section describes how to implement shorter Flash memory erasing/programing/verify time.

The programming flow shown in section 26.4.5.3 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}).

The mismatch results are stacked, so the confirming each sending expect data is not needed. The stacked result is cleared by clear command.

26.4.6.1 Timing to transmit command of Advanced Control

The LSI executes erasing/programming instructions to the Flash memory when it receives 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} = (24[bit] / transfer rate[bps])
Wait time : t_{wait} = Busy time : t_{busy} – (t_{cmd} × number of commands)

Figure 26-12 shows an example of command transmit for programming with using BUSY confirmation command.
When the transfer rate is 2Mbps and programming data in program code area:
Send the commands so that the interval between the first clocks of the two “BUSY confirmation command” is t_{busy} or more.

t_{cmd} = 24 bit / 2Mbps = 12 [µs]
t_{wait} = t_{busy} – (t_{cmd} × 2) = 80 – 12 × 2 = 56 [µs]

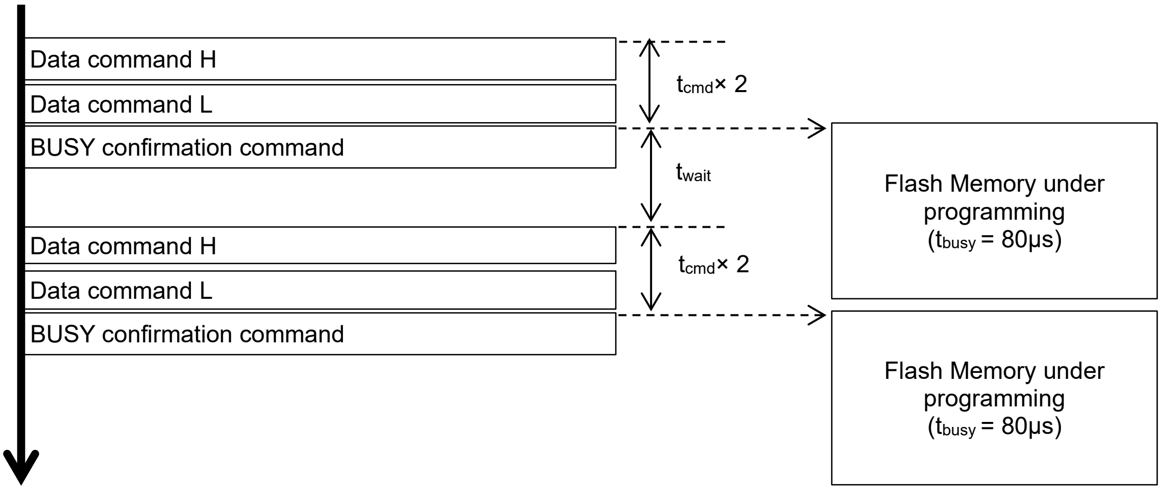

Data command H
Data command L
BUSY confirmation command
Data command H
Data command L
BUSY confirmation command
t_{cmd}× 2
t_{wait}
t_{cmd}× 2
Flash Memory under programming (t_{busy} = 80µs)
Flash Memory under programming (t_{busy} = 80µs)

Figure 26-12 Advanced control #1 of programming the program code area

Figure 26-13 shows an example of using “data command H” instead of “BUSY confirmation command”.

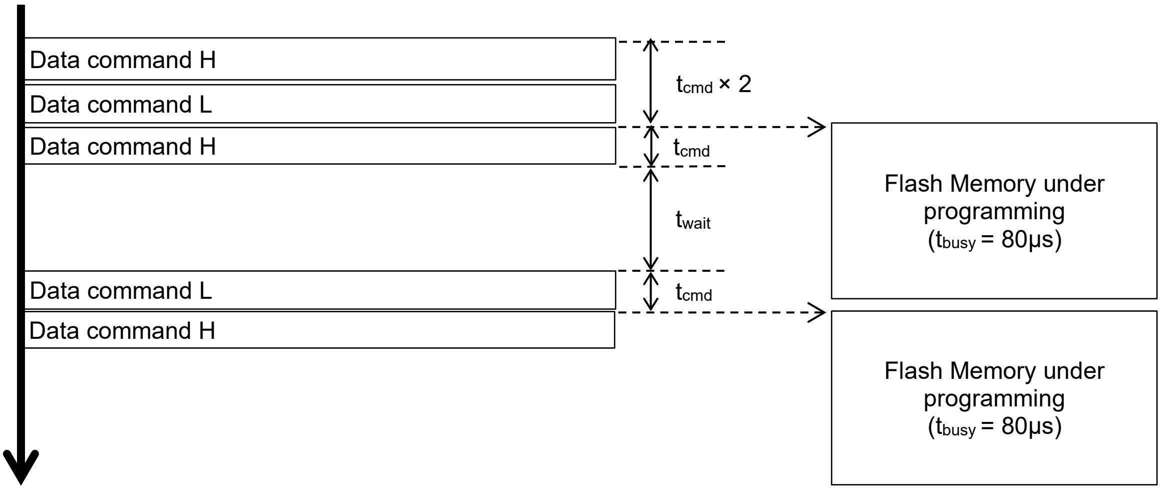


Figure 26-13 Advanced control #2 of programming the program code area

Figure 26-14 shows an example for programming to data flash area.
When the transfer rate is 2Mbps and programming data in data flash area:
Send the command “Data command D” so that the command acceptance interval is t_{busy} or more.

t_{cmd} = 24 bit / 2Mbps = 12 [μs]
t_{wait} = t_{busy} – (t_{cmd} × 1) = 40 – 12 × 1 = 28 [μs]

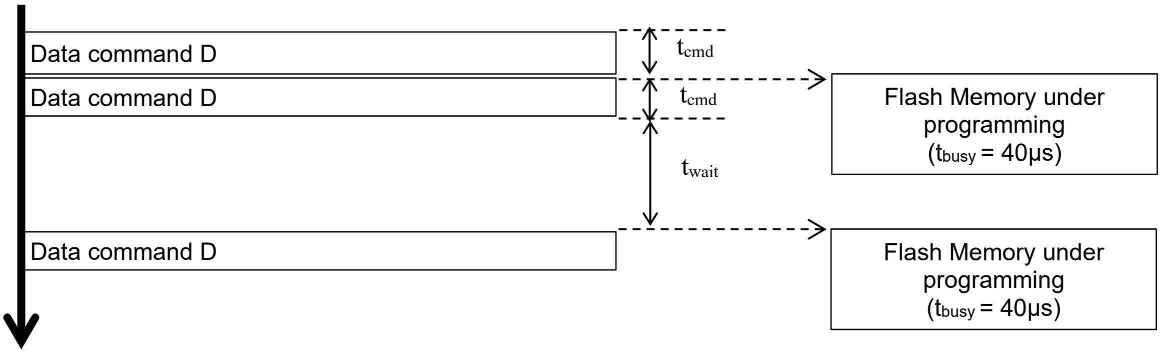

Data command D
Data command D
Data command D
tcmd
tcmd
twait
Flash Memory under programming (tbusy = 40μs)
Flash Memory under programming (tbusy = 40μs)

Figure 26-14 Advanced control of programming to the data flash

26.4.6.2 Erasing Data in Specified Flash Memory Area (Advanced control)

Figure 26-15 shows the flow diagram for erasing the specified flash memory area by the advanced control.

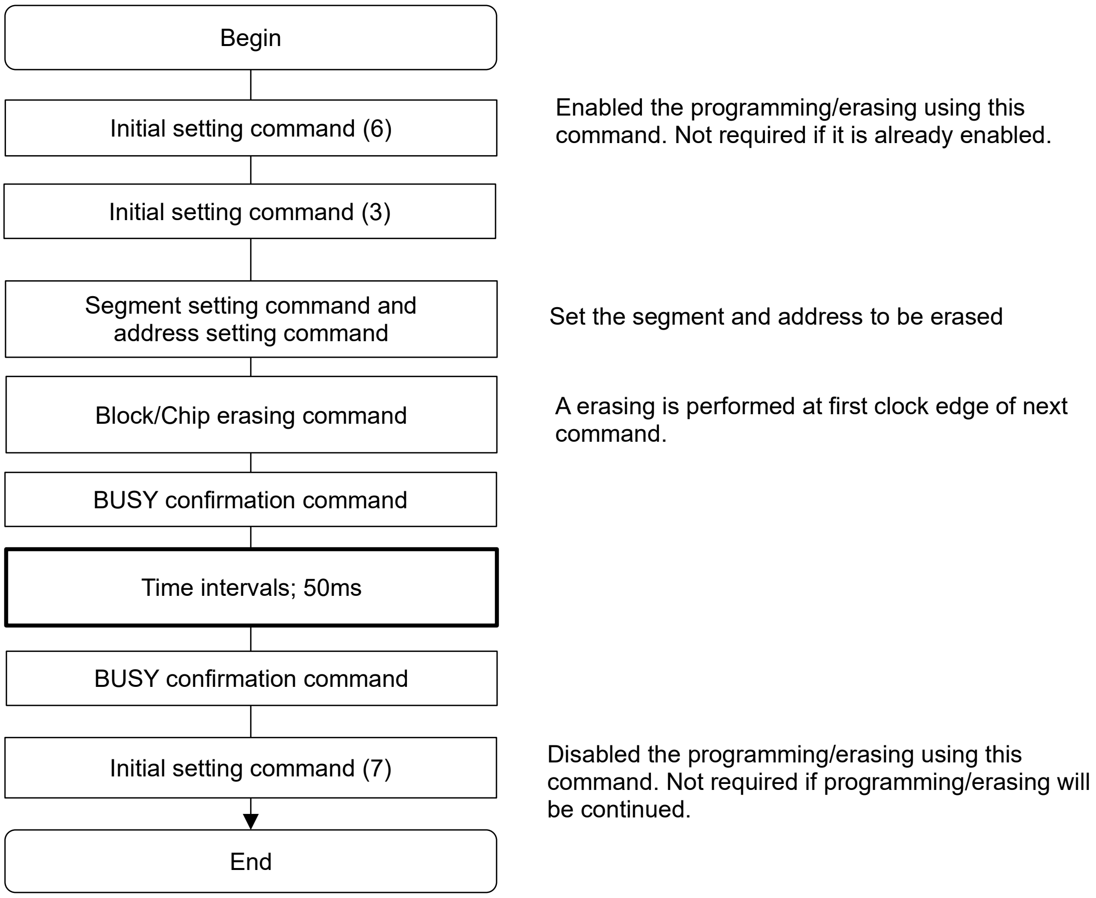

Begin
Initial setting command (6)
Enabled the programming/erasing using this command. Not required if it is already enabled.
Initial setting command (3)
Segment setting command and address setting command
Set the segment and address to be erased
Block/Chip erasing command
A erasing is performed at first clock edge of next command.
BUSY confirmation command
Time intervals; 50ms
BUSY confirmation command
Initial setting command (7)
Disabled the programming/erasing using this command. Not required if programming/erasing will be continued.
End

Figure 26-15 Flow Diagram for Erasing Specified Flash Memory Area (Advanced Control)

[Note]

- **Transmit command to avoid a timeout. See Section “26.4.3.1 Command Timing”.**
- **Transmit any command after ‘initial setting command (7)’ if other command will not be transmit.**

26.4.6.3 Programming to Specified Flash Memory Area (Advanced control)

Figure 26-16 shows the flow diagram for programming to the specified flash memory area by the advanced control.

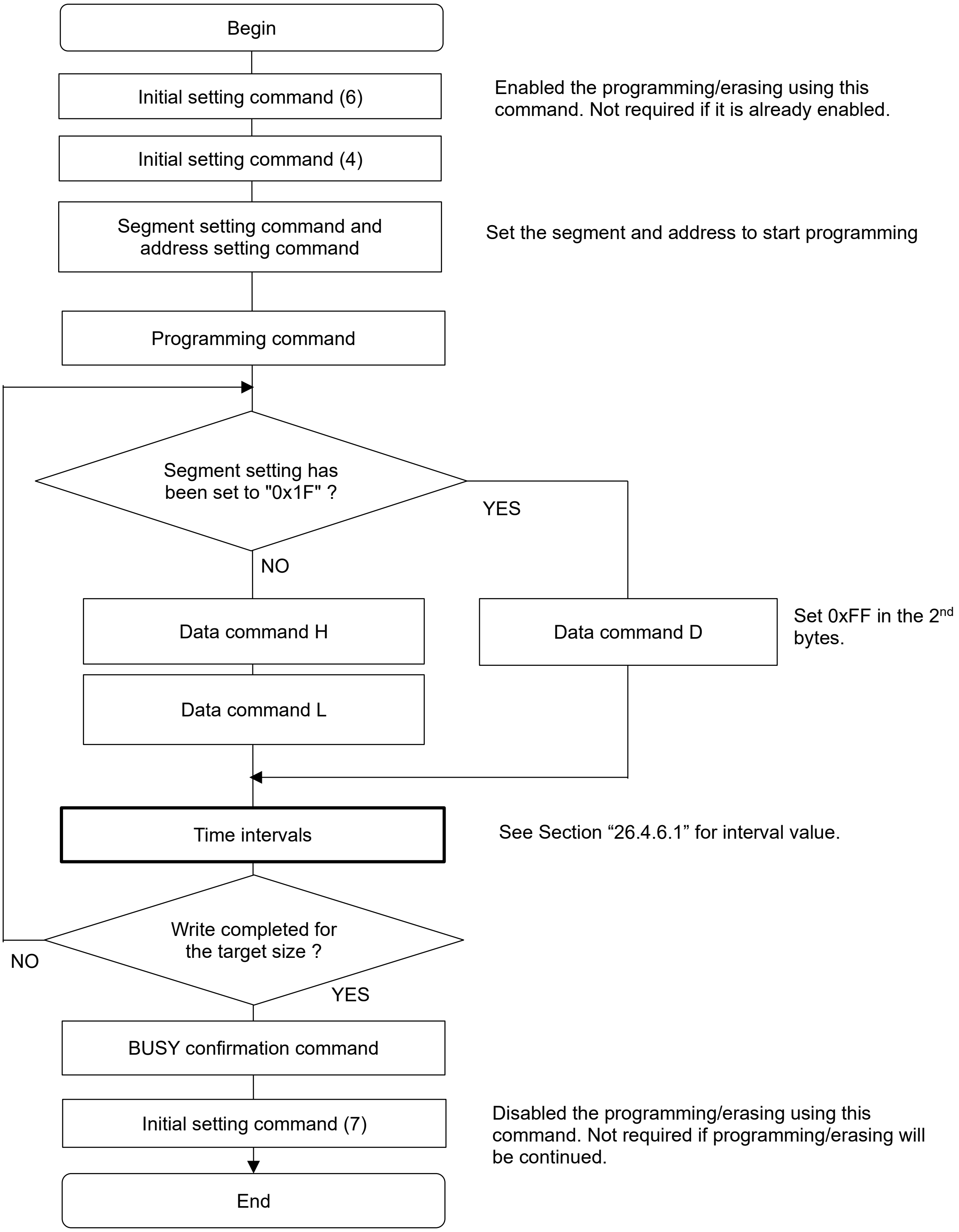

Begin
Initial setting command (6)
Enabled the programming/erasing using this command. Not required if it is already enabled.
Initial setting command (4)
Segment setting command and address setting command
Set the segment and address to start programming
Programming command
Segment setting has been set to "0x1F" ?
YES
NO
Data command H
Data command D
Set 0xFF in the 2nd bytes.
Data command L
Time intervals
See Section "26.4.6.1" for interval value.
Write completed for the target size ?
NO
YES
BUSY confirmation command
Initial setting command (7)
Disabled the programming/erasing using this command. Not required if programming/erasing will be continued.
End

Figure 26-16 Flow Diagram for Programming Specified Flash Memory Area (Advanced Control)

[Note]

- **Transmit command to avoid a timeout. See Section "26.4.3.1 Command Timing".**
- **Transmit any command after 'initial setting command (7)' if other command will not be transmit.**

26.4.6.4 Verifying Data in Specified Flash Memory Area (Advanced control)

Figure 26-17 shows the flow diagram for verifying data in the specified flash memory area by the advanced control.

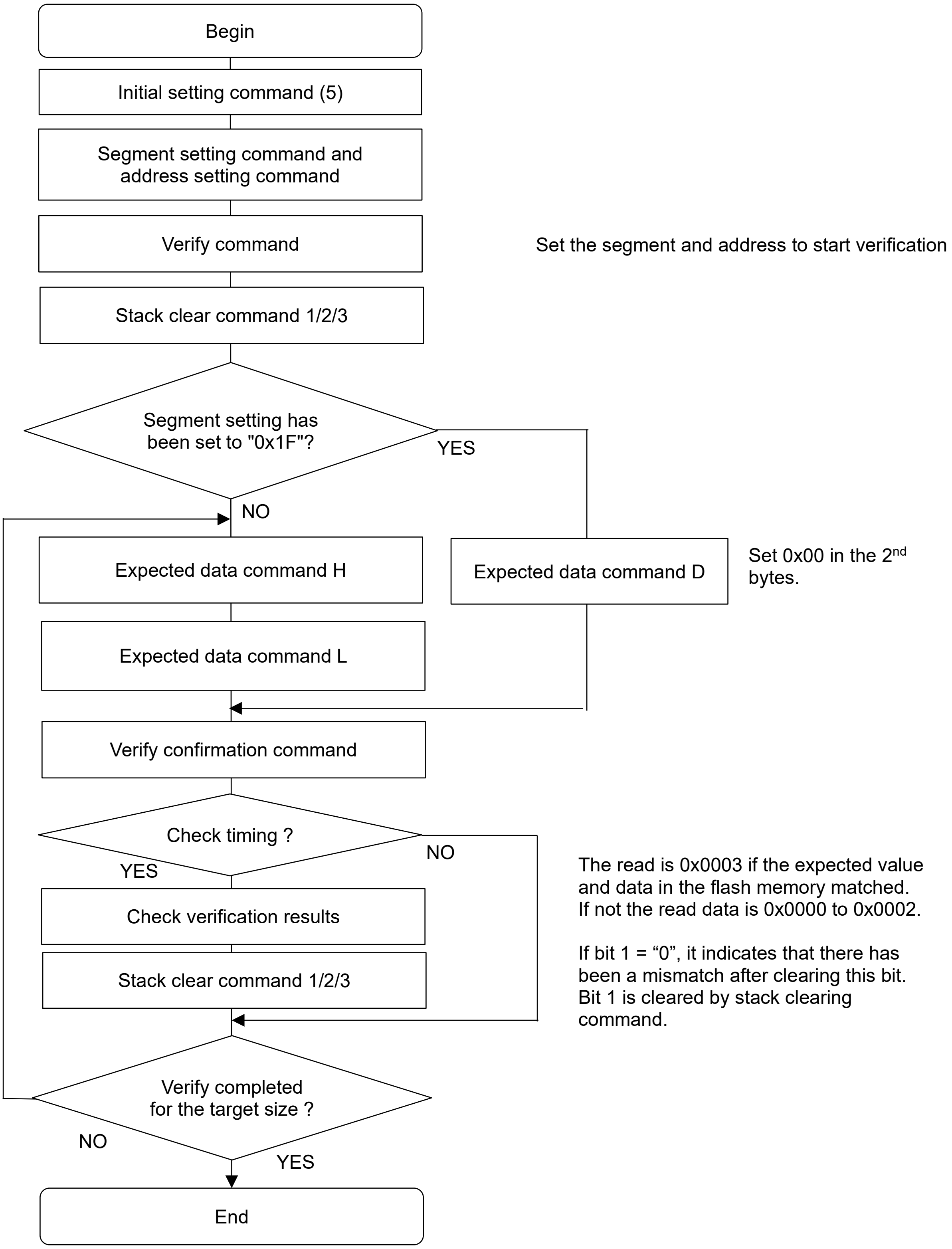

Begin
Initial setting command (5)
Segment setting command and address setting command
Verify command
Set the segment and address to start verification
Stack clear command 1/2/3
Segment setting has been set to "0x1F"?
YES
NO
Expected data command H
Expected data command D
Set 0x00 in the 2nd bytes.
Expected data command L
Verify confirmation command
Check timing ?
NO
YES
Check verification results
Stack clear command 1/2/3
The read is 0x0003 if the expected value and data in the flash memory matched. If not the read data is 0x0000 to 0x0002.
If bit 1 = "0", it indicates that there has been a mismatch after clearing this bit. Bit 1 is cleared by stack clearing command.
Verify completed for the target size ?
NO
YES
End

Figure 26-17 Flow Diagram for Verifying Specified Flash Memory Area (advanced control)

[Note]

- **Transmit command to avoid a timeout. See Section “26.4.3.1 Command Timing”.**
- **Transmit any command after ‘stack clear command 3’ if other command will not be transmit.**

Chapter 30 Code Option

30. Code Option

30.1 General Description

The code option is used to choose a CPU operating mode, PLL reference frequency, watchdog timer operation etc. depending on values written in the code option area of the program memory area.

The hardware automatically refers to data in the code option area when the microcontroller starts up due to one of system resets described below to set each function.
The code option area can be erased or programmed through the on-chip debug function, self-rewrite function of flash memory, or ISP function.

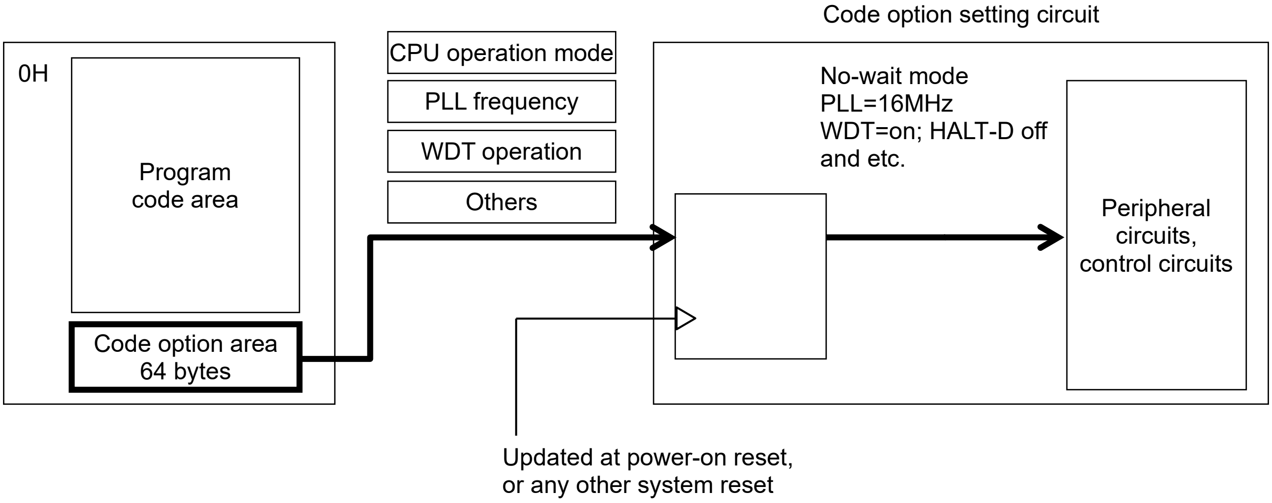

Code option setting circuit
0H
Program code area
Code option area 64 bytes
CPU operation mode
PLL frequency
WDT operation
Others
No-wait mode
PLL=16MHz
WDT=on; HALT-D off
and etc.
Peripheral circuits, control circuits
Updated at power-on reset, or any other system reset

Figure 30-1 Code Option Overview

30.1.1 Function List

- Readable configured code options from SFRs
- Enabling or disabling the unused ROM area access reset
- Enabling or disabling the remapping function
- The software remap or hardware remap is selectable for the remap function
- Enabling or disabling the watchdog timer operation
- Enabling or disabling the watchdog timer operation at HALT/HALT-H, HALT-D
- PLL reference frequency (1MHz / 16MHz / 24MHz)
- CPU operation mode (wait mode or no-wait mode)
- Enabling or disabling a clock back-up function of LSCLK1
- Configured V_{DDL} voltage

30.2 Description of Code Option

30.2.1 Reading from SFRs

The address of code option area is dependent of the size of the program memory space (flash memory).
The address of SFRs for reading is fixed.

SFR address	Register Name	Symbol name		R/W	Size
		Byte	Word		
0xF920	Code Option 0	CODEOP0L	CODEOP0	R	8/16
0xF921		CODEOP0H		R	8
0xF922	Code Option 1	CODEOP1L	CODEOP1	R	8/16
0xF923		CODEOP1H		R	8
0xF924	Code Option 2	CODEOP2L	CODEOP2	R	8/16
0xF925		CODEOP2H		R	8

[Note]

- **There are available to read the code option values from SFRs, if INITE flag bit of Reset Status Register (RSTAT) is “0”.**

30.2.2 Code Option 0 (CODEOP0）

Address:	(See Table 30-1)
SFR address for reading:	0xF920 (CODEOP0L/CODEOP0), 0xF921(CODEOP0H)
Access to SFR :	R
Access size to SFR:	8/16 bit
Initial value:	0xFFFF (Erased or factory default setting for products with blank flash memory)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CODEOP0															
Byte	CODEOP0H								CODEOP0L							
Bit	-	-	-	PCER MD	-	-	-	REMA PMD	LS1BU	-	-	-	WDTP WMD1	WDTP WMD0	-	WDTM D
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 13	-	Reserved bits. Set "1" to all bits.
12	PCERMD	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: Disabled 1: Enabled (Initial value)
11 to 9	-	Reserved bits. Set "1" to all 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: Enabled 1: Disabled (Initial value)
7	LS1BU	This bit is used to enable /disable a back-up function of LSCLK1. 0: Disabled 1: Enabled (Initial value)
6 to 4	-	Reserved bits. Set "1" to all bits.
3	WDTPWMD1	This bit is used to choose to enable/disable the watchdog timer (WDT) operation in HALT-D mode, if WDTMD = “1”. 0: Disabled 1: Enabled (Initial value)
2	WDTPWMD0	This bit is used to choose to enable/disable the watchdog timer (WDT) operation in HALT/HALT-H mode, if WDTMD = “1”. 0: Disabled 1: Enabled (Initial value)
1	-	Reserved bit. Set "1" to this bit.
0	WDTMD	This bit is used to choose to enable/disable the watchdog timer (WDT) operation. 0: Disabled 1: Enabled (Initial value)

30.2.3 Code Option 1 (CODEOP1)

Address:	(See Table 30-1)
SFR address for reading:	0xF922 (CODEOP1L/CODEOP1), 0xF923(CODEOP1H)
Access to SFR :	R
Access size to SFR:	8/16 bit
Initial value:	0xFFFF (Erased or factory default setting for products with blank flash memory)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CODEOP1															
Byte	CODEOP1H								CODEOP1L							
Bit	-	-	-	-	-	-	-	-	-	VLMD	-	-	PLLMD1	PLLMD0	CPUMD1	CPUMD0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

<table>
<tr><th>Bit No.</th><th>Bit symbol name</th><th>Description</th></tr>
<tr><td>15 to 7</td><td>-</td><td>Reserved bits. Set "1" to all bits.</td></tr>
<tr><td>6</td><td>VLMD</td><td>Set 0 to this bit if VDDL is not down in the condition except the STOP-D/HALT-D.
If it is 1, a wake-up time from the HALT-H for PLL16/24M is longer than when it is 0.</td></tr>
<tr><td>5, 4</td><td>-</td><td>Reserved bits. Set "1" to all bits.</td></tr>
<tr><td>3, 2</td><td>PLLMD1,
PLLMD0</td><td>These bits are used to choose the PLL frequency.
00: 1MHz
01: Do not use
10: 24 MHz
11: 16 MHz (Initial value)
The following table shows the relation between the PLL frequency and the maximum operating frequency of CPU and peripheral circuits.
<table>
<tr><th rowspan="2">PLL frequency</th><th colspan="3">Maximum operating frequency</th></tr>
<tr><th>Peripheral circuit</th><th>CPU (wait mode)</th><th>CPU (no-wait mode)</th></tr>
<tr><td>24MHz</td><td>24MHz</td><td>24MHz</td><td>6MHz</td></tr>
<tr><td>16MHz</td><td>16MHz</td><td>16MHz</td><td>8MHz</td></tr>
<tr><td>1MHz</td><td>1MHz</td><td>1MHz</td><td>1MHz</td></tr>
</table>
See Chapter 2 "CPU and Memory Space" and Appendix C "Instruction Execution Cycle" for the CPU operation modes (wait mode and no-wait mode).</td></tr>
<tr><td>1, 0</td><td>CPUMD1,
CPUMD0</td><td>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)</td></tr>
</table>

30.2.4 Code Option 2 (CODEOP2)

Address: (See Table 30-1)
SFR address for reading: 0xF924 (CODEOP2L/CODEOP2), 0xF925 (CODEOP2H)
Access to SFR : R
Access size to SFR: 8/16 bit
Initial value: 0xFFFF (Erased or factory default setting for products with blank flash memory)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CODEOP2															
Byte	CODEOP2H								CODEOP2L							
Bit	CREMAPMD	CRES2	CRES1	CRES0	CREA15	CREA14	CREA13	CREA12	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15	CREMAPMD	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 CRES1 to 0 bits 1: The initial value of the REMAPADD is 0x00 If setting this bit to "0", The initial value of the REMAPADD consists of CREA15 to 12 bits and CRES1 to 0 bits. For details on REMAPADD, see Section 2.7.3 "Flash Remap Address Register (REMAPADD)". The MCU remaps to the address specified with the CREA15 to 12 bits and the CRES1 to 0 bits every time at the system reset. See also Section 2.8.3 "Code Option Remap". The remap function is enabled by setting REMAPMD bit of the Code Options 0.
14 to 12	CRES2 to CRES0	These bits are used to set the initial values of RES2 to RES0 bits of the Flash Remap Address Register (REMAPADD). RES2 and RES1 are reserved bits.
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. Set "1" to all bits.

CPU instruction execution start address after releasing the reset

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

See Section 2.7.3 "Flash Remap Address Register (REMAPADD)" and Section 2.8.3 "Code Option Remap".

30.3 Code Option Data Setting

The address of code option area is dependent of the size of the program memory space (flash memory).
Table 30-1 shows addresses of code option areas for each product.

Table 30-1 List of Addresses of Code Option Areas for Each Product

Product name	Program memory space size	Code Option area	Address		
			CODEOP2	CODEOP1	CODEOP0
ML62Q2502/2522/2532	64K byte	0x0:FFC0～0x0:FFFF	0x0:FFD4	0x0:FFD2	0x0:FFD0
ML62Q2504/2524/2534	128K byte	0x1:FFC0～0x1:FFFF	0x1:FFD4	0x1:FFD2	0x1:FFD0

Figure 30-2 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 (ML622xxx.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 30-2 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 31 Auxiliary Function

31. Auxiliary Function

31.1 General Description

- Indication of Product ID
- Indication of unique chip ID (32bit); reading from FLASH
- Bit swap for 32bit data; converting MSB/LSB.
- Byte swap for 32bit data; converting Big/Little endian.

31.2 Description of Registers

31.2.1 List of Registers

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF930	Product ID register 0	PID0L	PID0	R	8/16	*1
0xF931		PID0H		R	8	*1
0xF932	Product ID register 1	PID1L	PID1	R	8/16	0x22
0xF933		PID1H		R	8	0x06
0xF940	Converting Base Data register L	CNVBD0	CNVBDL	R/W	8/16	Undefined
0xF941		CNVBD1		R/W	8	Undefined
0xF942	Converting Base Data register H	CNVBD2	CNVBDH	R/W	8/16	Undefined
0xF943		CNVBD3		R/W	8	Undefined
0xF944	Bit Swap Result register L	CNVAD0	CNVADL	R	8/16	Undefined
0xF945		CNVAD1		R	8	Undefined
0xF946	Bit Swap Result register H	CNVAD2	CNVADH	R	8/16	Undefined
0xF947		CNVAD3		R	8	Undefined
0xF948	Byte Swap Result register L	CNVED0	CNVEDL	R	8/16	Undefined
0xF949		CNVED1		R	8	Undefined
0xF94A	Byte Swap Result register H	CNVED2	CNVEDH	R	8/16	Undefined
0xF94B		CNVED3		R	8	Undefined

*1: It depends on product.

31.2.2 Product ID Register 0,1 (PID0, PID1)

This is a SFR to indicate product ID.

Address :	0xF930 (PID0L/PID0), 0xF931(PID0H), 0xF932 (PID1L/PID1), 0xF933(PID1H),
Access :	R
Access size :	8/16 bit
Initial value :	PID1: 0622, PID0:5xx0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PID1															
Byte	PID1H								PID1L							
Bit	-	-	-	-	d53	d52	d51	d50	d43	d42	d41	d40	d33	d32	d31	d30
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PID0															
Byte	PID0H								PID0L							
Bit	d23	d22	d21	d20	d13	d12	d11	d10	d03	d02	d01	d00	ex3	ex2	ex1	ex0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	1	0	1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0	0	0	0

Bit No.	Bit symbol name	Description
15 to 4	-	It indicates 6-digit number of product name. The upper 20-bit number is fixed at “0x6225”. The lower 8-bit number depends on the product name.
3 to 0	-	It indicates 1 character from A to F as extended identifier of product. If the product name has “P” or “T”, it is ignored. If the product name has no alphabet, this indicates “0” in principle.

Ex) ML62Q2534 : “0x0622_5340”

31.2.3 Converting Base Data Register L/H (CNVBDL, CNVBDH)

This is a SFR to set a conversion source data.

Address :	0xF940 (CNVBD0/CNVBDL), 0xF941 (CNVBD1), 0xF942 (CNVBD2/CNVBDH), 0xF943 (CNVBD3)
Access :	R/W
Access size :	8/16 bit
Initial value :	Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CNVBDH															
Byte	CNVBD3								CNVBD2							
Bit	d31	d30	d29	d28	d27	d26	d25	d24	d23	d22	d21	d20	d19	d18	d17	d16
R/W	R/W	R/W	R/W	R/W	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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CNVBDL															
Byte	CNVBD1								CNVBD0							
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	R/W	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

31.2.4 Bit Swap Result Register L/H (CNVADL, CNVADH)

This is a SFR to read a bit-swap converted data. The bit-swap is a conversion that reverses bit by bit.

Address :	0xF944 (CNVAD0/CNVADL), 0xF945 (CNVAD1), 0xF946 (CNVAD2/CNVADH), 0xF947 (CNVAD3)
Access :	R
Access size :	8/16 bit
Initial value :	Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CNVADH															
Byte	CNVAD3								CNVAD2							
Bit	d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CNVADL															
Byte	CNVAD1								CNVAD0							
Bit	d16	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26	d27	d28	d29	d30	d31
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
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

31.2.5 Byte Swap Result Register L/H (CNVEDL, CNVEDH)

This is a SFR to read a byte-swap converted data. The bit-swap is a conversion that reverses byte by byte.

Address :	0xF948 (CNVED0/CNVEDL), 0xF949 (CNVED1), 0xF94A (CNVED2/CNVEDH), 0xF94B (CNVED3)
Access :	R
Access size :	8/16 bit
Initial value :	Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CNVEDH															
Byte	CNVED3								CNVED2							
Bit	d7	d6	d5	d4	d3	d2	d1	d0	d15	d14	d13	d12	d11	d10	d9	d8
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	CNVEDL															
Byte	CNVED1								CNVED0							
Bit	d23	d22	d21	d20	d19	d18	d17	d16	d31	d30	d29	d28	d27	d26	d25	d24
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
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

31.3 Description of Operation

31.3.1 How to Confirm Unique ID

A LSI chip of ML62Q2500 group has unique ID with 32-bit. It is to read from test area in data memory space.
Its address is different for each product.

Table 31-1 Address to read for each product

Product name	Program Memory size	Data Memory Space Address to read.
ML62Q25x2	64KByte	0x1:03F8~B
ML62Q25x4	128KByte	0x2:03F8~B

31.3.2 Data Swap Function

Set data to CNVBDL and CNVBDH registers, then Read a bit-swap result from CNVADL/CNVADH registers and a byte-swap result from CNVEDL/CNVEDH registers.

Ex)
In the case that CNVBDH = 0x1234, CNVBDL = 0x5678,

Result :
CNVADH = 0x1E6A, CNVADL = 0x2C48
CNVEDH = 0x7856, CNVADL = 0x3412

Appendix

Appendix A SFR List

The SFR list is show below. Access "Reserved" register is not guaranteed. Please do not access them.
Initial value with *1 depend on code option that is set. See Chapter "30 Code option".
Initial value with *2 depend on product. See Chapter "31 Auxiliary Function".

Address	Name	Symbol		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	0x43
0xF004	Low-speed clock mode register	FLMODL	FLMOD	R/W	8/16	0x00
0xF005		FLMODH		R/W	8	0x00
0xF006	Clock control register	FCON	FCONW	R/W	8/16	0x00
0xF007		FCON1		R/W	8	0x00
0xF008	High-speed clock wake up time setting register	FHWUPT	-	R/W	8	0x00
0xF009 ~0xF00B	Reserved	-	-	-	-	-
0xF00C	Backup Clock Status register	FBUSTAT	FBUSTATW	R/W	8/16	0x01
0xF00D		FBUSTATH		R	8	0x01
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	SBYCONL	SBYCON	W	8/16	0x00
0xF01B		SBYCONH		R/W	8	0x00
0xF01C	Standby prohibition flag register	SBYEFLG	-	R	8	0x00
0xF01D ~0xF01F	Reserved	-	-	-	-	-
0xF020	Interrupt enable register 01	IE0	IE01	R/W	8/16	0x00
0xF021		IE1		R/W	8	0x00
0xF022	Interrupt enable register 23	IE2	IE23	R/W	8/16	0x00
0xF023		IE3		R/W	8	0x00
0xF024	Interrupt enable register 45	IE4	IE45	R/W	8/16	0x00
0xF025		IE5		R/W	8	0x00
0xF026	Interrupt enable register 67	IE6	IE67	R/W	8/16	0x00
0xF027		IE7		R/W	8	0x00
0xF028	Interrupt request register 01	IRQ0	IRQ01	R/W	8/16	0x00
0xF029		IRQ1		R/W	8	0x00
0xF02A	Interrupt request register 23	IRQ2	IRQ23	R/W	8/16	0x00
0xF02B		IRQ3		R/W	8	0x00
0xF02C	Interrupt request register 45	IRQ4	IRQ45	R/W	8/16	0x00
0xF02D		IRQ5		R/W	8	0x00
0xF02E	Interrupt request register 67	IRQ6	IRQ67	R/W	8/16	0x00
0xF02F		IRQ7		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF030	Interrupt level control enable register	ILEN	-	R/W	8	0x00
0xF031	Reserved	-	-	-	-	-
0xF032	Current interrupt level management register	CIL	-	R/W	8	0x00
0xF033	Interrupt level mask register	MCIL	-	R/W	8	0x00
0xF034	Interrupt level control register 0	ILC00	ILC0	R/W	8/16	0x00
0xF035		ILC01		R/W	8	0x00
0xF036	Interrupt level control register 1	ILC10	ILC1	R/W	8/16	0x00
0xF037		ILC11		R/W	8	0x00
0xF038	Interrupt level control register 2	ILC20	ILC2	R/W	8/16	0x00
0xF039		ILC21		R/W	8	0x00
0xF03A	Interrupt level control register 3	ILC30	ILC3	R/W	8/16	0x00
0xF03B		ILC31		R/W	8	0x00
0xF03C	Interrupt level control register 4	ILC40	ILC4	R/W	8/16	0x00
0xF03D		ILC41		R/W	8	0x00
0xF03E	Interrupt level control register 5	ILC50	ILC5	R/W	8/16	0x00
0xF03F		ILC51		R/W	8	0x00
0xF040	Interrupt level control register 6	ILC60	ILC6	R/W	8/16	0x00
0xF041		ILC61		R/W	8	0x00
0xF042	Interrupt level control register 7	ILC70	ILC7	R/W	8/16	0x00
0xF043		ILC71		R/W	8	0x00
0xF044	External interrupt control register 0	EICON0L	EICON0	R/W	8/16	0x00
0xF045		EICON0H		R/W	8	0x00
0xF046	Reserved	-	-	-	-	-
0xF047	Reserved	-	-	-	-	-
0xF048	External interrupt mode register 0	EIMOD0L	EIMOD0	R/W	8/16	0x00
0xF049		EIMOD0H		R/W	8	0x00
0xF04A	Reserved	-	-	R	8	0x00
0xF04B	Reserved	-	-	R	8	0x00
0xF04C	External interrupt port selection register 0	EIPSEL0L	EIPSEL0	R/W	8/16	0x00
0xF04D		EIPSEL0H		R/W	8	0x00
0xF04E	Reserved	-	-	-	-	-
0xF04F	Reserved	-	-	-	-	-
0xF050	MCU Status Interrupt Enable Register	MCINTEL	-	R/W	8	0x00
0xF051	Reserved	-	-	-	-	-
0xF052	MCU Status Interrupt Register	MCISTATL	-	R	8	0x00
0xF053	Reserved	-	-	-	-	-
0xF054	MCU Status Interrupt Clear Register (L/H)	MCINTCLL	-	W	8	0x00
0xF055		MCINTCLH		W	8	0x00
0xF056	Reserved	-	-	-	-	-
0xF057	Reserved	-	-	-	-	-
0xF058	Reset status register	RSTATL	RSTAT	R/W	8/16	Undefined
0xF059		RSTATH		R/W	8	Undefined
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 ~0xF06F	Reserved	-	-	-	-	-
0xF070	Block clock control register 0	BCKCON0L	BCKCON0	R/W	8/16	0x1F

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF071		BCKCON0H		R/W	8	0x01
0xF072	Block clock control register 1	BCKCON1L	BCKCON1	R/W	8/16	0x03
0xF073		BCKCON1H		R/W	8	0x11
0xF074	Block clock control register 2	BCKCON2L	BCKCON2	R/W	8/16	0x73
0xF075		BCKCON2H		R/W	8	0x08
0xF076	Block clock control register 3	BCKCON3L	BCKCON3	R/W	8/16	0x01
0xF077		BCKCON3H		R/W	8	0x00
0xF078	Block reset control register 0	BRECON0L	BRECON0	R/W	8/16	0x1F
0xF079		BRECON0H		R/W	8	0x01
0xF07A	Block reset control register 1	BRECON1L	BRECON1	R/W	8/16	0x03
0xF07B		BRECON1H		R/W	8	0x11
0xF07C	Block reset control register 2	BRECON2L	BRECON2	R/W	8/16	0x73
0xF07D		BRECON2H		R/W	8	0x08
0xF07E	Block reset control register 3	BRECON3L	BRECON3	R/W	8/16	0x01
0xF07F		BRECON3H		R/W	8	0x00
0xF080	Reserved	-	-	-	-	-
0xF081 ~0xF085	Reserved	-	-	-	-	-
0xF086	High speed time base clock setting register	HTBDR	-	R/W	8	0x00
0xF087 ~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
0xF0A1 ~0xF0A3	Reserved	-	-	-	-	-
0xF0A4	Reserved	-	-	-	-	-
0xF0A5	Reserved	-	-	-	-	-
0xF0A6	Reserved	-	-	-	-	-
0xF0A7 ~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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF0B6	SFR Guard Setting Register 1	SFRGD1L	SFRGD1	R/W	8/16	0x00
0xF0B7		SFRGD1H		R/W	8	0x00
0xF0B8 ~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 ~0xF0C3	Reserved	-	-	-	-	-
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 ~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 ~0xF1FF	Reserved	-	-	-	-	-
0xF200	Port 0 data register	P0DI	P0D	R	8/16	0xFF
0xF201		P0DO		R/W	8	0x00
0xF202	Port 0 mode register 0	P0MOD0	-	R/W	8/16	0x05
0xF203	Reserved	-	-	-	-	-
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	8/16	0xFF
0xF211		P1DO		R/W	8	0x00
0xF212	Port 1 mode register 01	P1MOD0	P1MOD01	R/W	8/16	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
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	Reserved	-	-	-	-	-
0xF217						
0xF218	Reserved	-	-	-	-	-
0xF219	Port 1 mode register 7	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	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	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	Port 3 mode register 45	P3MOD4	P3MOD45	R/W	8/16	0x00
0xF237		P3MOD5		R/W	8	0x00
0xF238	Port 3 mode register 67	P3MOD6	P3MOD67	R/W	8/16	0x00
0xF239		P3MOD7		R/W	8	0x00
0xF23A ~0xF24F	Reserved	-	-	-	-	-
0xF250	Port 5 data register	P5DI	P5D	R	8/16	0xFF
0xF251		P5DO		R/W	8	0x00
0xF252	Reserved	-	-	-	-	-
0xF253				-	-	-
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

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF25A ~0xF25F	Reserved	-	-	-	-	-
0xF260	Port 6 data register	P6DI	P6D	R	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 2	P6MOD2	-	R/W	8/16	0x00
0xF265	Reserved	-	-	-	-	-
0xF266 ~0xF26F	Reserved	-	-	-	-	-
0xF270	Port 7 data register	P7DI	P7D	R	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 ~0xF279	Reserved	-	-	-	-	-
0xF27A	Port 7 pulse mode register	P7PMDL	P7PMD	R/W	8/16	0x00
0xF27B		P7PMDH		R/W	8	0x00
0xF27C	Port 7 pulse selection register	P7PSLL	P7PSL	R/W	8/16	0x00
0xF27D		P7PSLH		R/W	8	0x00
0xF27E ~0xF2EF	Reserved	-	-	-	-	-
0xF2F0	PORTXT data input register	PXTDI	-	R	8	0x03
0xF2F1	Reserved	-	-	-	-	-
0xF2F2	PORTXT mode register 01	PXTMOD0	PXTMOD01	R/W	8/16	0x00
0xF2F3		PXTMOD1		R/W	8	0x00
0xF2F4 ~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 0 counter register	TMH0CL	TMH0C	R/W	8/16	0x00
0xF303		TMH0CH		R/W	8	0x00
0xF304	16-bit timer 0 mode register	TMH0MODL	TMH0MOD	R/W	8/16	0x00
0xF305		TMH0MODH		R/W	8	0x00
0xF306	Reserved	-	-	-	-	-
0xF307	Reserved	-	-	-	-	-
0xF308	16-bit timer 1 data register	TMH1DL	TMH1D	R/W	8/16	0xFF
0xF309		TMH1DH		R/W	8	0xFF
0xF30A	16-bit timer 1 counter register	TMH1CL	TMH1C	R/W	8/16	0x00
0xF30B		TMH1CH		R/W	8	0x00
0xF30C	16-bit timer 1 mode register	TMH1MODL	TMH1MOD	R/W	8/16	0x00
0xF30D		TMH1MODH		R/W	8	0x00
0xF30E	Reserved	-	-	-	-	-
0xF30F	Reserved	-	-	-	-	-
0xF310	16-bit timer 2 data register	TMH2DL	TMH2D	R/W	8/16	0xFF
0xF311		TMH2DH		R/W	8	0xFF
0xF312	16-bit timer 2 counter register	TMH2CL	TMH2C	R/W	8/16	0x00
0xF313		TMH2CH		R/W	8	0x00
0xF314	16-bit timer 2 mode register	TMH2MODL	TMH2MOD	R/W	8/16	0x00
0xF315		TMH2MODH		R/W	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF316	Reserved	-	-	-	-	-
0xF317	Reserved	-	-	-	-	-
0xF318	16-bit timer 3 data register	TMH3DL	TMH3D	R/W	8/16	0xFF
0xF319		TMH3DH		R/W	8	0xFF
0xF31A	16-bit timer 3 counter register	TMH3CL	TMH3C	R/W	8/16	0x00
0xF31B		TMH3CH		R/W	8	0x00
0xF31C	16-bit timer 3 mode register	TMH3MODL	TMH3MOD	R/W	8/16	0x00
0xF31D		TMH3MODH		R/W	8	0x00
0xF31E	Reserved	-	-	-	-	-
0xF31F	Reserved	-	-	-	-	-
0xF320	16-bit timer 4 data register	TMH4DL	TMH4D	R/W	8/16	0xFF
0xF321		TMH4DH		R/W	8	0xFF
0xF322	16-bit timer 4 counter register	TMH4CL	TMH4C	R/W	8/16	0x00
0xF323		TMH4CH		R/W	8	0x00
0xF324	16-bit timer 4 mode register	TMH4MODL	TMH4MOD	R/W	8/16	0x00
0xF325		TMH4MODH		R/W	8	0x00
0xF326 ~0xF33F	Reserved	-	-	-	-	-
0xF340	16-bit timer start register	TMHSTRL	TMHSTR	W	8/16	0x00
0xF341		TMHSTRH		W	8	0x00
0xF342	16-bit timer stop register	TMHSTPL	TMHSTP	W	8/16	0x00
0xF343		TMHSTPH		W	8	0x00
0xF344	16-bit timer status register	TMHSTATL	TMHSTAT	R	8/16	0x00
0xF345		TMHSTATH		R	8	0x00
0xF346	Reserved	-	-	-	-	-
0xF347	Reserved	-	-	-	-	-
0xF350	16-bit timer X data register	TMHXDL	TMHXD	R/W	8/16	0xFF
0xF351		TMHXDH		R/W	8	0xFF
0xF352	16-bit timer X counter register	TMHXCL	TMHXC	R/W	8/16	0x00
0xF353		TMHXCH		R/W	8	0x00
0xF354	16-bit timer X mode register	TMHXMODL	TMHXMOD	R/W	8/16	0x00
0xF355		TMHXMODH		R/W	8	0x00
0xF356	Reserved	-	-	-	-	-
0xF357	Reserved	-	-	-	-	-
0xF358	16-bit timer X start register	TMHXSTR	-	W	8	0x00
0xF359	Reserved	-	-	-	-	-
0xF35A	16-bit timer X stop register	TMHXSTP	-	W	8	0x00
0xF35B	Reserved	-	-	-	-	-
0xF35C	16-bit timer X status register	TMHXSTAT	-	R	8	0x00
0xF35D ~0xF37F	Reserved	-		-	-	-
0xF380	FTM common update register	FTCUD	-	W	8	0x00
0xF381	Reserved	-	-	-	-	-
0xF382	FTM common control register	FTCCONL	FTCCON	R/W	8/16	0x00
0xF383		FTCCONH		R/W	8	0x00
0xF384	FTM common start register	FTCSTRL	FTCSTR	W	8/16	0x00
0xF385		FTCSTRH		W	8	0x00
0xF386	FTM common stop register	FTCSTPL	FTCSTP	W	8/16	0x00
0xF387		FTCSTPH		W	8	0x00
0xF388	FTM common status register	FTCSTATL	FTCSTAT	R	8/16	0x00
0xF389		FTCSTATH		R	8	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF38A ~0xF39F	Reserved	-		-	-	-
0xF3A0	Low-speed Time Base Counter register	LTBR0	LTBR01	R/W	8/16	0x00
0xF3A1		LTBR1		R/W	8	0x00
0xF3A2	Low-speed Time Base Counter Control register	LTBCON0	LTBCON	R/W	8/16	0x03
0xF3A3		LTBCON1		R/W	8	0x02
0xF3A4	Reserved	-	-	-	-	-
0xF3A5	Reserved	-	-	-	-	-
0xF3A6	Low-speed Time Base Counter Frequency Adjustment register	LTBADJL	LTBADJ	R/W	8/16	0x00
0xF3A7		LTBADJH		R/W	8	0x00
0xF3A8	Low-speed Time Base Counter Interrupt selection register	LTBINTL	LTBINT	R/W	8/16	0x60
0xF3A9		LTBINTH		R/W	8	0x71
0xF3AA ~0xF3FF	Reserved	-		-	-	-
0xF400	FTM0 cycle register	FT0PL	FT0P	R/W	8/16	0xFF
0xF401		FT0PH		R/W	8	0xFF
0xF402	FTM0 event A register	FT0EAL	FT0EA	R/W	8/16	0x00
0xF403		FT0EAH		R/W	8	0x00
0xF404	FTM0 event B register	FT0EBL	FT0EB	R/W	8/16	0x00
0xF405		FT0EBH		R/W	8	0x00
0xF406	FTM0 dead time register	FT0DTL	FT0DT	R/W	8/16	0x00
0xF407		FT0DTH		R/W	8	0x00
0xF408	FTM0 counter register	FT0CL	FT0C	R/W	8/16	0x00
0xF409		FT0CH		R/W	8	0x00
0xF40A	FTM0 status register	FT0STAT	-	R	8	0x30
0xF40B	Reserved	-	-	-	-	-
0xF40C	FTM0 mode register	FT0MODL	FT0MOD	R/W	8/16	0x00
0xF40D		FT0MODH		R/W	8	0x40
0xF40E	FTM0 clock register	FT0CLKL	FT0CLK	R/W	8/16	0x00
0xF40F		FT0CLKH		R/W	8	0x00
0xF410	FTM0 trigger register 0	FT0TRG0L	FT0TRG0	R/W	8/16	0x00
0xF411		FT0TRG0H		R/W	8	0x00
0xF412	FTM0 trigger register 1	FT0TRG1L	FT0TRG1	R/W	8/16	0x00
0xF413		FT0TRG1H		R/W	8	0x00
0xF414	FTM0 interrupt enable register	FT0INTEL	FT0INTE	R/W	8/16	0x00
0xF415		FT0INTEH		R/W	8	0x00
0xF416	FTM0 interrupt status register	FT0INTSL	FT0INTS	R	8/16	0x00
0xF417		FT0INTSH		R	8	0x00
0xF418	FTM0 interrupt clear register	FT0INTCL	-	W	8	0x00
0xF419		FT0INTCH		W	8	0x00
0xF41A ~0xF41F	Reserved	-	-	-	-	-
0xF420	FTM1 cycle register	FT1PL	FT1P	R/W	8/16	0xFF
0xF421		FT1PH		R/W	8	0xFF
0xF422	FTM1 event A register	FT1EAL	FT1EA	R/W	8/16	0x00
0xF423		FT1EAH		R/W	8	0x00
0xF424	FTM1 event B register	FT1EBL	FT1EB	R/W	8/16	0x00
0xF425		FT1EBH		R/W	8	0x00
0xF426	FTM1 dead time register	FT1DTL	FT1DT	R/W	8/16	0x00
0xF427		FT1DTH		R/W	8	0x00
0xF428	FTM1 counter register	FT1CL	FT1C	R/W	8/16	0x00

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF429		FT1CH		R/W	8	0x00
0xF42A	FTM1 status register	FT1STAT	-	R	8	0x30
0xF42B	Reserved	-	-	-	-	-
0xF42C	FTM1 mode register	FT1MODL	FT1MOD	R/W	8/16	0x00
0xF42D		FT1MODH		R/W	8	0x40
0xF42E	FTM1 clock register	FT1CLKL	FT1CLK	R/W	8/16	0x00
0xF42F		FT1CLKH		R/W	8	0x00
0xF430	FTM1 trigger register 0	FT1TRG0L	FT1TRG0	R/W	8/16	0x00
0xF431		FT1TRG0H		R/W	8	0x00
0xF432	FTM1 trigger register 1	FT1TRG1L	FT1TRG1	R/W	8/16	0x00
0xF433		FT1TRG1H		R/W	8	0x00
0xF434	FTM1 interrupt enable register	FT1INTEL	FT1INTE	R/W	8/16	0x00
0xF435		FT1INTEH		R/W	8	0x00
0xF436	FTM1 interrupt status register	FT1INTSL	FT1INTS	R	8/16	0x00
0xF437		FT1INTSH		R	8	0x00
0xF438	FTM1 interrupt clear register	FT1INTCL	FT1INTC	W	8/16	0x00
0xF439		FT1INTCH		W	8	0x00
0xF43A ~0xF4FF	Reserved	-	-	-	-	-
0xF500	Serial port 0 transmission/reception buffer	SIO0BUFL	SIO0BUF	R/W	8/16	0x00
0xF501		SIO0BUFH		R/W	8	0x00
0xF502	Serial port 0 status register	SIO0STATL	SIO0STAT	R	8/16	0x00
0xF503		SIO0STATH		W	8	0x00
0xF504	Serial port 0 control register	SIO0CONL	SIO0CON	R/W	8/16	0x00
0xF505		SIO0CONH		R/W	8	0x00
0xF506	Serial port 0 mode register	SIO0MODL	SIO0MOD	R/W	8/16	0x00
0xF507		SIO0MODH		R/W	8	0x00
0xF508	Serial port 0 interval setting register	SIO0DLYL	-	R/W	8	0x00
0xF509	Reserved	-	-	-	-	-
0xF50A	Serial port 0 interrupt control register	SIO0ICNL	-	R/W	8	0x00
0xF50B ~0xF57F	Reserved	-	-	-	-	-
0xF580	SIOF0 control register	SF0CTRLL	SF0CTRL	R/W	8/16	0x00
0xF581		SF0CTRLH		R/W	8	0x00
0xF582	SIOF0 interrupt control register	SF0INTCL	SF0INTC	R/W	8/16	0x00
0xF583		SF0INTCH		R/W	8	0x00
0xF584	SIOF0 transfer interval control register	-	SF0TRAC	R/W	16	0x0002
0xF585		-				
0xF586	SIOF0 baud rate register	-	SF0BRR	R/W	16	0x5002
0xF587		-				
0xF588	SIOF0 status register	SF0SRRL	SF0SRR	R	8/16	0x00
0xF589		SF0SRRH		R	8	0x14
0xF58A	SIOF0 status clear register (L/H)	SF0SRCL	-	W	8	0x00
0xF58B		SF0SRCH	-	W	8	0x00
0xF58C	SIOF0 FIFO status register	SF0FSRL	SF0FSR	R	8/16	0x00
0xF58D		SF0FSRH		R	8	0x00
0xF58E	SIOF0 writing data register	SF0DWRL	SF0DWR	R/W	8/16	0x00
0xF58F		SF0DWRH		R/W	8	0x00
0xF590	SIOF0 reading data register	SF0DRRL	SF0DRR	R	8/16	0x00
0xF591		SF0DRRH		R	8	0x00
0xF592	Reserved	-	-	-	-	-

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
~0xF5FF						
0F600	UART0 reception buffer	UA0BUF0	-	R	8	0x00
0F601	UART0 transmission buffer	UA0BUF1		R/W	8	0x00
0F602	UART0 status register	UA0STAT	-	R	8	0x00
0F603	UART0 status clear register	UA0STAC		W	8	0x00
0F604	UART0 control register	UA0CON	-	R/W	8	0x00
0F605	Reserved	-	-	-	-	-
0F606	UART0 mode register	UA0MODL	UA0MOD	R/W	8/16	0x00
0F607		UA0MODH		R/W	8	0x00
0F608	UART0 interrupt enable register	UA0INTE	-	R/W	8	0x00
0F609	Reserved	-	-	-	-	-
0F60A	UART0 baud rate register	UA0BRTL	UA0BRT	R/W	8/16	0xFF
0F60B		UA0BRTH		R/W	8	0x0F
0F60C	UART0 baud rate adjustment register	UA0BRC	-	R/W	8	0x00
0F60D ~0F60F	Reserved	-	-	-	-	-
0F610	UART1 reception buffer	UA1BUF0	-	R	8	0x00
0F611	UART1 transmission buffer	UA1BUF1	-	R/W	8	0x00
0F612	UART1 status register	UA1STAT	-	R	8	0x00
0F613	UART1 status clear register	UA1STAC	-	W	8	0x00
0F614	UART1 control register	UA1CON	-	R/W	8	0x00
0F615	Reserved	-	-	-	-	-
0F616	UART1 mode register	UA1MODL	UA1MOD	R/W	8/16	0x00
0F617		UA1MODH		R/W	8	0x00
0F618	UART1 interrupt enable register	UA1INTE	-	R/W	8	0x00
0F619	UART1 reception buffer	-	-	-	-	-
0F61A	UART1 transmission buffer UART1 status register	UA1BRTL	UA1BRT	R/W	8/16	0xFF
0F61B		UA1BRTH		R/W	8	0x0F
0F61C	UART1 status clear register	UA1BRC	-	R/W	8	0x00
0F61D ~0F61F	Reserved	-	-	-	-	-
0F620	UART2 reception buffer	UA2BUF0	-	R	8	0x00
0F621	UART2 transmission buffer	UA2BUF1	-	R/W	8	0x00
0F622	UART2 status register	UA2STAT	-	R	8	0x00
0F623	UART2 status clear register	UA2STAC	-	W	8	0x00
0F624	UART2 control register	UA2CON	-	R/W	8	0x00
0F625	Reserved	-	-	-	-	-
0F626	UART2 mode register	UA2MODL	UA2MOD	R/W	8/16	0x00
0F627		UA2MODH		R/W	8	0x00
0F628	UART2 interrupt enable register	UA2INTE	-	R/W	8	0x00
0F629	Reserved	-	-	-	-	-
0F62A	UART2 baud rate register	UA2BRTL	UA2BRT	R/W	8/16	0xFF
0F62B		UA2BRTH		R/W	8	0x0F
0F62C	UART2 baud rate adjustment register	UA2BRC	-	R/W	8	0x00
0F62D ~0F77F	Reserved	-	-	-	-	-
0xF780	I^{2}C bus unit 0 mode register	I2U0MSS	-	R/W	8	0x00
0xF781	Reserved	-	-	-	-	-
0xF782	I^{2}C bus unit 0 receive register	I2U0RD	-	R	8	0x00
0xF783	Reserved	-	-	-	-	-
0xF784	I^{2}C bus unit 0 slave address register	I2U0SA	-	R/W	8	0x00

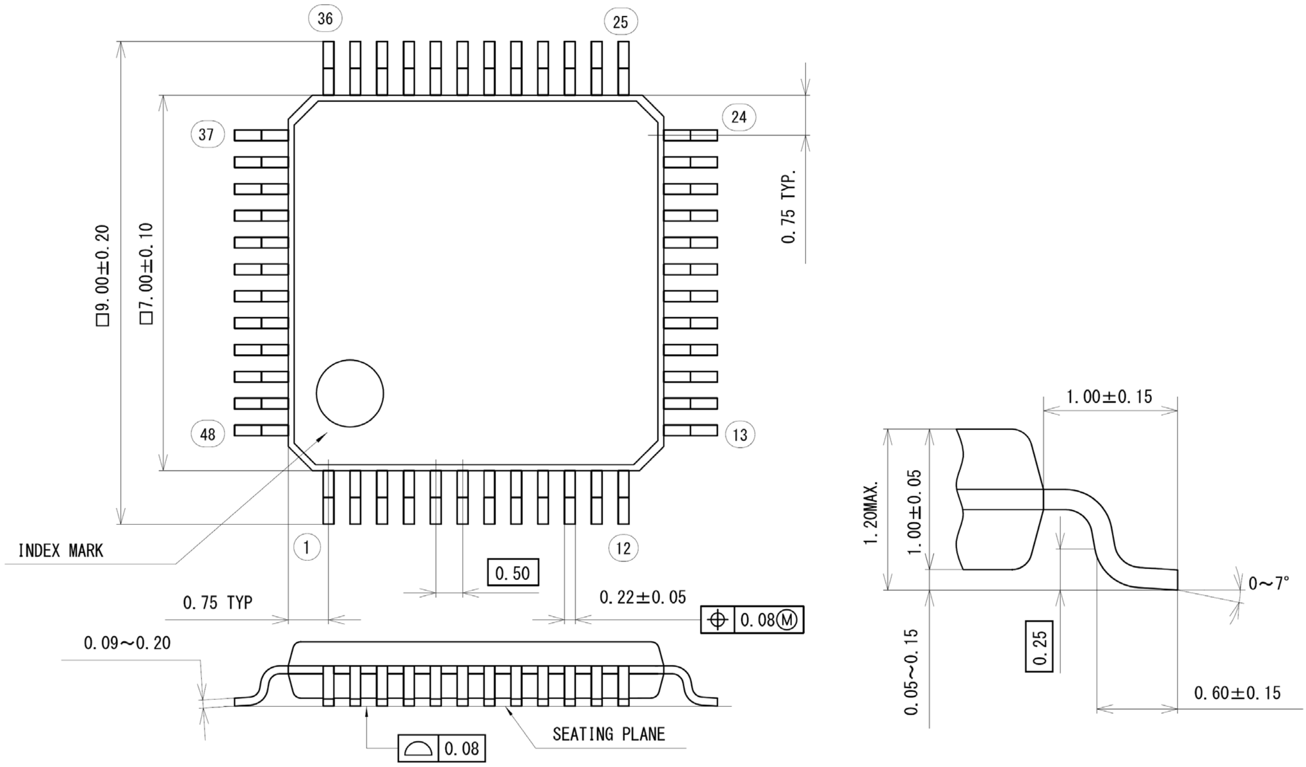
Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF785	Reserved	-	-	-	-	-
0xF786	I²C bus unit 0 transmit data register	I2U0TD	-	R/W	8	0x00
0xF787	Reserved	-	-	-	-	-
0xF788	I²C bus unit 0 control register	I2U0CON	-	R/W	8	0x00
0xF789	Reserved	-	-	-	-	-
0xF78A	I²C bus unit 0 mode register	I2U0MODL	I2U0MOD	R/W	8/16	0x00
0xF78B		I2U0MODH		R/W	8	0x02
0xF78C	I²C bus unit 0 status register	I2U0STAT	I2U0STR	R	8/16	0x00
0xF78D		I2U0ISR		R	8	0x00
0xF78E	I²C bus unit 0 status clear register	I2U0SCLRL	I2U0SCLR	W	8/16	0x00
0xF78F		I2U0SCLRH		W	8	0x00
0F790 ~0F7C1	Reserved	-	-	-	-	-
0xF7C2	I²C bus master 0 receive register	I2M0RD	-	R	8	0x00
0xF7C3	Reserved	-	-	-	-	-
0xF7C4	I²C bus master 0 slave address register	I2M0SA	-	R/W	8	0x00
0xF7C5	Reserved	-	-	-	-	-
0xF7C6	I²C bus master 0 transmit data register	I2M0TD	-	R/W	8	0x00
0xF7C7	Reserved	-	-	-	-	-
0xF7C8	I²C bus master 0 control register	I2M0CON	-	R/W	8	0x00
0xF7C9	Reserved	-	-	-	-	-
0xF7CA	I²C bus master 0 mode register	I2M0MODL	I2M0MOD	R/W	8/16	0x00
0xF7CB		I2M0MODH		R/W	8	0x02
0xF7CC	I²C bus master 0 status register	I2M0STAT	I2M0STR	R	8/16	0x00
0xF7CD		I2M0ISR		R	8	0x00
0xF7CE	I²C bus master 0 status clear register	I2M0SCLRL	I2M0SCLR	W	8/16	0x00
0xF7CF		I2M0SCLRH		W	8	0x00
0xF7D0 ~0xF7FF	Reserved	-	-	-	-	-
0xF800	SA-ADC mode register	SADMODL	SADMOD	R/W	8/16	0x00
0xF801		SADMODH		R/W	8	0x00
0xF802	SA-ADC control register	SADCONL	SADCON	R/W	8/16	0x00
0xF803		SADCONH		R/W	8	0x00
0xF804	SA-ADC conversion interval register	SADSTML	SADSTM	R/W	8/16	0x00
0xF805		SADSTMH		R/W	8	0x00
0xF806	Reference voltage control register	VREFCON	-	R/W	8	0x00
0xF807	Reserved	-	-	-	-	-
0xF808	SA-ADC interrupt mode register	SADIMOD	-	R/W	8	0x00
0xF809	Reserved	-	-	-	-	-
0xF80A	SA-ADC trigger register	SADTRG	-	R/W	8	0x00
0xF80B	Reserved	-	-	-	-	-
0xF80C	SA-ADC enable register 0	SADEN0L	SADEN0	R/W	8/16	0x00
0xF80D		SADEN0H		R/W	8	0x00
0xF80E	SA-ADC enable register 1	SADEN1L	SADEN1	R/W	8/16	0x00
0xF80F		SADEN1H		R/W	8	0x00
0xF810 ~0xF81F	Reserved	-	-	-	-	-
0xF820	SA-ADC upper/lower limit mode register	SADLMODL	SADLMOD	R/W	8/16	0x00
0xF821		SADLMODH		R/W	8	0x00
0xF822	SA-ADC upper limit setting register	SADUPLL	SADUPL	R/W	8/16	0xF0
0xF823		SADUPLH		R/W	8	0xFF

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF824	SA-ADC lower limit setting register	SADLOLL	SADLOL	R/W	8/16	0x00
0xF825		SADLOLH		R/W	8	0x00
0xF826	SA-ADC upper/lower limit status register 0	SADULS0L	SADULS0	R	8/16	0x00
0xF827		SADULS0H		R	8	0x00
0xF828	Reserved	-	-	-	-	-
0xF829	Reserved	-	-	-	-	-
0xF82A	SA-ADC upper/lower limit status clear register 0	SADULC0L	SADULC0	W	8/16	0x00
0xF82B		SADULC0H		W	8	0x00
0xF82C ~0xF82F	Reserved	-	-	-	-	-
0xF830	SA-ADC test mode register	SADTMOD	-	R/W	8	0x00
0xF831 ~0xF83D	Reserved	-	-	-	-	-
0xF83E	SA-ADC result register	SADRL	SADR	R	8/16	0x00
0xF83F		SADRH		R	8	0x00
0xF840	SA-ADC result register 0	SADR0L	SADR0	R	8/16	0x00
0xF841		SADR0H		R	8	0x00
0xF842	SA-ADC result register 1	SADR1L	SADR1	R	8/16	0x00
0xF843		SADR1H		R	8	0x00
0xF844	SA-ADC result register 2	SADR2L	SADR2	R	8/16	0x00
0xF845		SADR2H		R	8	0x00
0xF846	SA-ADC result register 3	SADR3L	SADR3	R	8/16	0x00
0xF847		SADR3H		R	8	0x00
0xF848	SA-ADC result register 4	SADR4L	SADR4	R	8/16	0x00
0xF849		SADR4H		R	8	0x00
0xF84A	SA-ADC result register 5	SADR5L	SADR5	R	8/16	0x00
0xF84B		SADR5H		R	8	0x00
0xF84C	SA-ADC result register 6	SADR6L	SADR6	R	8/16	0x00
0xF84D		SADR6H		R	8	0x00
0xF84E	SA-ADC result register 7	SADR7L	SADR7	R	8/16	0x00
0xF84F		SADR7H		R	8	0x00
0xF850	SA-ADC result register 8	SADR8L	SADR8	R	8/16	0x00
0xF851		SADR8H		R	8	0x00
0xF852	SA-ADC result register 9	SADR9L	SADR9	R	8/16	0x00
0xF853		SADR9H		R	8	0x00
0xF854	SA-ADC result register 10	SADR10L	SADR10	R	8/16	0x00
0xF855		SADR10H		R	8	0x00
0xF856	SA-ADC result register 11	SADR11L	SADR11	R	8/16	0x00
0xF857		SADR11H		R	8	0x00
0xF858	SA-ADC result register 12	SADR12L	SADR12	R	8/16	0x00
0xF859		SADR12H		R	8	0x00
0xF85A	SA-ADC result register 13	SADR13L	SADR13	R	8/16	0x00
0xF85B		SADR13H		R	8	0x00
0xF85C ~0xF88F	Reserved	-	-	-	-	-
0xF890	Voltage level supervisor 0 control register	VLS0CON	-	R/W	8	0x00
0xF891	Reserved	-	-	-	-	-
0xF892	Voltage level supervisor 0 mode register	VLS0MOD	-	R/W	8	0x00
0xF893	Reserved	-	-	-	-	-
0xF894	Voltage level supervisor 0 level register	VLS0LV	-	R/W	8	0x0E
0xF895	Reserved	-	-	-	-	-

Address	Name	Symbol		R/W	Size	Initial value
		Byte	Word			
0xF896	Voltage level supervisor 0 sampling register	VLS0SMP	-	R/W	8	0x00
0xF897 ~0xF91F	Reserved	-	-	-	-	-
0xF920	Code Option 0	CODEOP0L	CODEOP0	R	8/16	*1
0xF921		CODEOP0H		R	8	*1
0xF922	Code Option 1	CODEOP1L	CODEOP1	R	8/16	*1
0xF923		CODEOP1H		R	8	*1
0xF924	Code Option 2	CODEOP2L	CODEOP2	R	8/16	*1
0xF925		CODEOP2H		R	8	*1
0xF926 ~0xF92F	Reserved	-	-	-	-	-
0xF930	Product ID register 0	PID0L	PID0	R	8/16	*2
0xF931		PID0H		R	8	*2
0xF932	Product ID register 1	PID1L	PID1	R	8/16	0x22
0xF933		PID1H		R	8	0x06
0xF834 ~0xF93F	Reserved	-	-	-	-	-
0xF940	Converting Base Data register L	CNVBD0	CNVBDL	R/W	8/16	Undefined
0xF941		CNVBD1		R/W	8	Undefined
0xF942	Converting Base Data register H	CNVBD2	CNVBDH	R/W	8/16	Undefined
0xF943		CNVBD3		R/W	8	Undefined
0xF944	Bit Swap Result register L	CNVAD0	CNVADL	R	8/16	Undefined
0xF945		CNVAD1		R	8	Undefined
0xF946	Bit Swap Result register H	CNVAD2	CNVADH	R	8/16	Undefined
0xF947		CNVAD3		R	8	Undefined
0xF948	Byte Swap Result register L	CNVED0	CNVEDL	R	8/16	Undefined
0xF949		CNVED1		R	8	Undefined
0xF94A	Byte Swap Result register H	CNVED2	CNVEDH	R	8/16	Undefined
0xF94B		CNVED3		R	8	Undefined
0xF94C ~0xFFFF	Reserved	-	-	-	-	-

Appendix B Package Dimensions

48pin TQFP



NOTES:
1. LEAD WIDTH DOES NOT INCLUDE TRIM OFFSET.
2. PACKAGE WIDTH AND LENGTH DO NOT INCLUDE MOLD PROTRUSION, DIEPAD SUPPORT PROTRUSION AND CAVITY OFFSET BETWEEN TOP AND BOTTOM CAVITY.
3. THE SEATING PLANE IS THE SURFACE WHICH THE PACKAGE IS MOUNTED ON AND GETS IN CONTACT WITH.

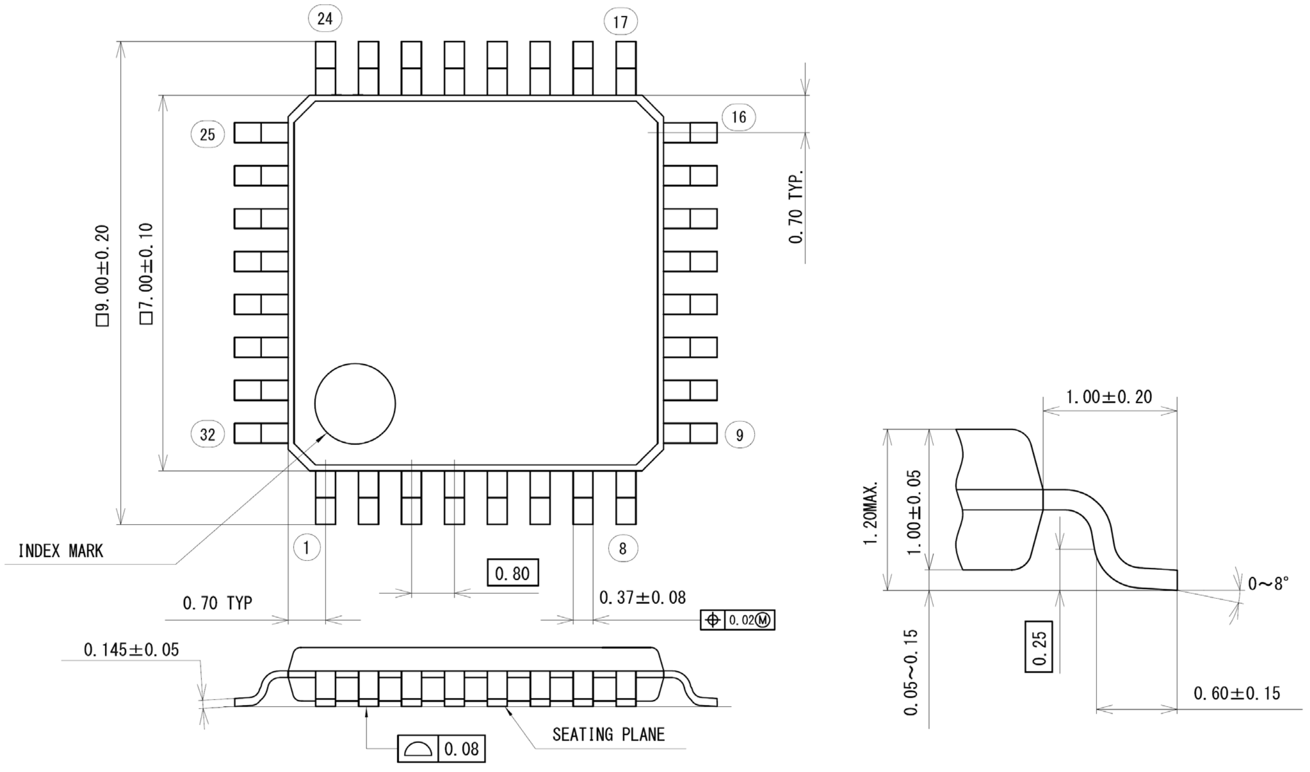
LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-TQFP48-0707-0.50-ZK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-69142
LEAD FINISH	Sn 100%	REVISION	4
PLATING THICKNESS	MORE THAN 5μm	1st ISSUE	Jul/28/2014
PACKAGE MASS (g)	0.13TYP	REVISED	Oct/12/2020

(Unit : mm)

[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 TQFP



NOTES:
1. LEAD WIDTH DOES NOT INCLUDE TRIM OFFSET.
2. PACKAGE WIDTH AND LENGTH DO NOT INCLUDE MOLD PROTRUSION, DIEPAD SUPPORT PROTRUSION AND CAVITY OFFSET BETWEEN TOP AND BOTTOM CAVITY.
3. THE SEATING PLANE IS THE SURFACE WHICH THE PACKAGE IS MOUNTED ON AND GETS IN CONTACT WITH.

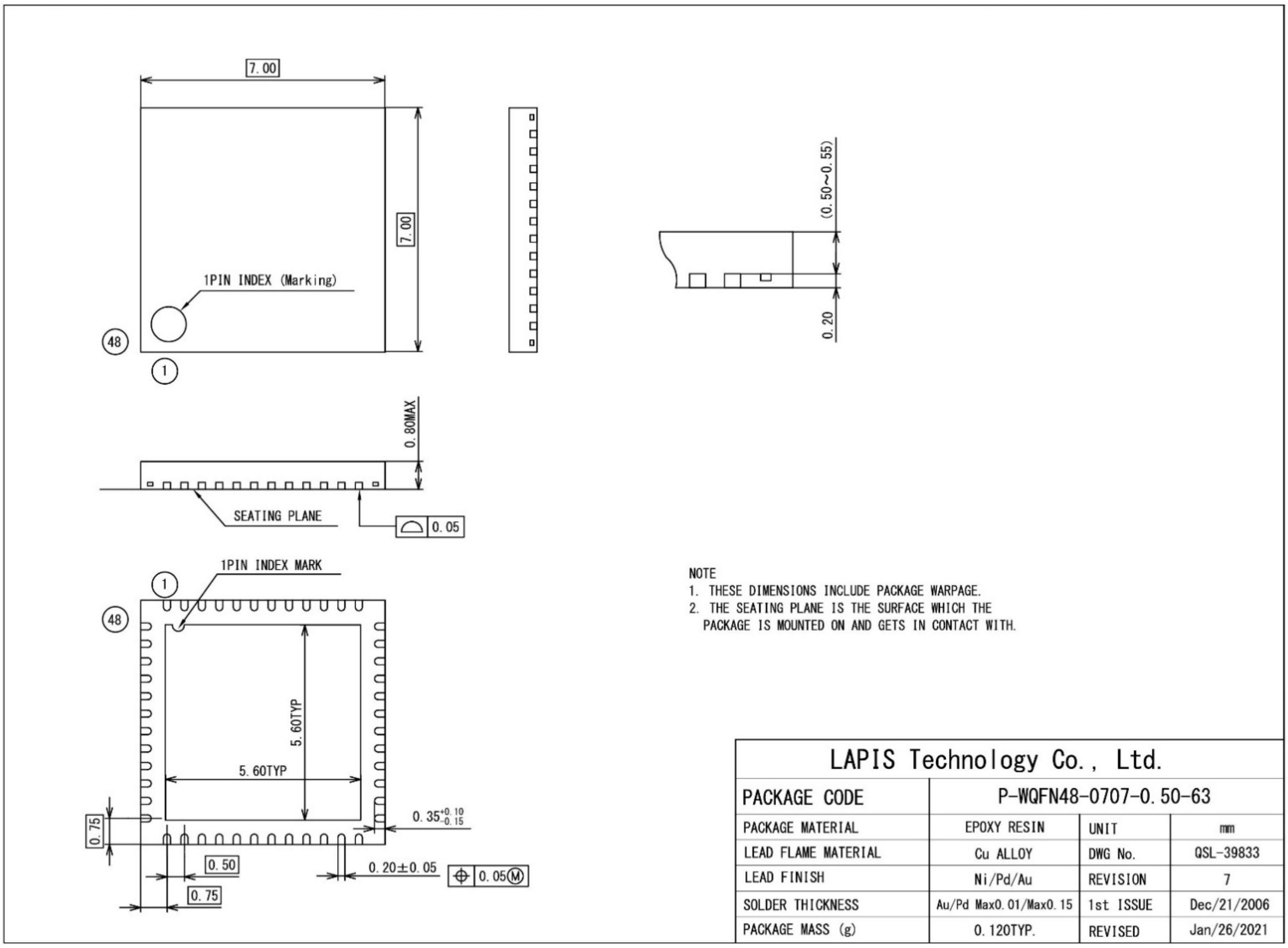
LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-TQFP32-0707-0.80-ZK6		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-69350
LEAD FINISH	Sn 100%	REVISION	2
PLATING THICKNESS	MORE THAN 5μm	1st ISSUE	Aug/04/2016
PACKAGE MASS (g)	0.130	REVISED	Oct/02/2020

(Unit : mm)

[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).

48pin WQFN



NOTE
1. THESE DIMENSIONS INCLUDE PACKAGE WARPAGE.
2. THE SEATING PLANE IS THE SURFACE WHICH THE PACKAGE IS MOUNTED ON AND GETS IN CONTACT WITH.

LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-WQFN48-0707-0.50-63		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-39833
LEAD FINISH	Ni/Pd/Au	REVISION	7
SOLDER THICKNESS	Au/Pd Max0.01/Max0.15	1st ISSUE	Dec/21/2006
PACKAGE MASS (g)	0.120TYP.	REVISED	Jan/26/2021

(Unit : mm)

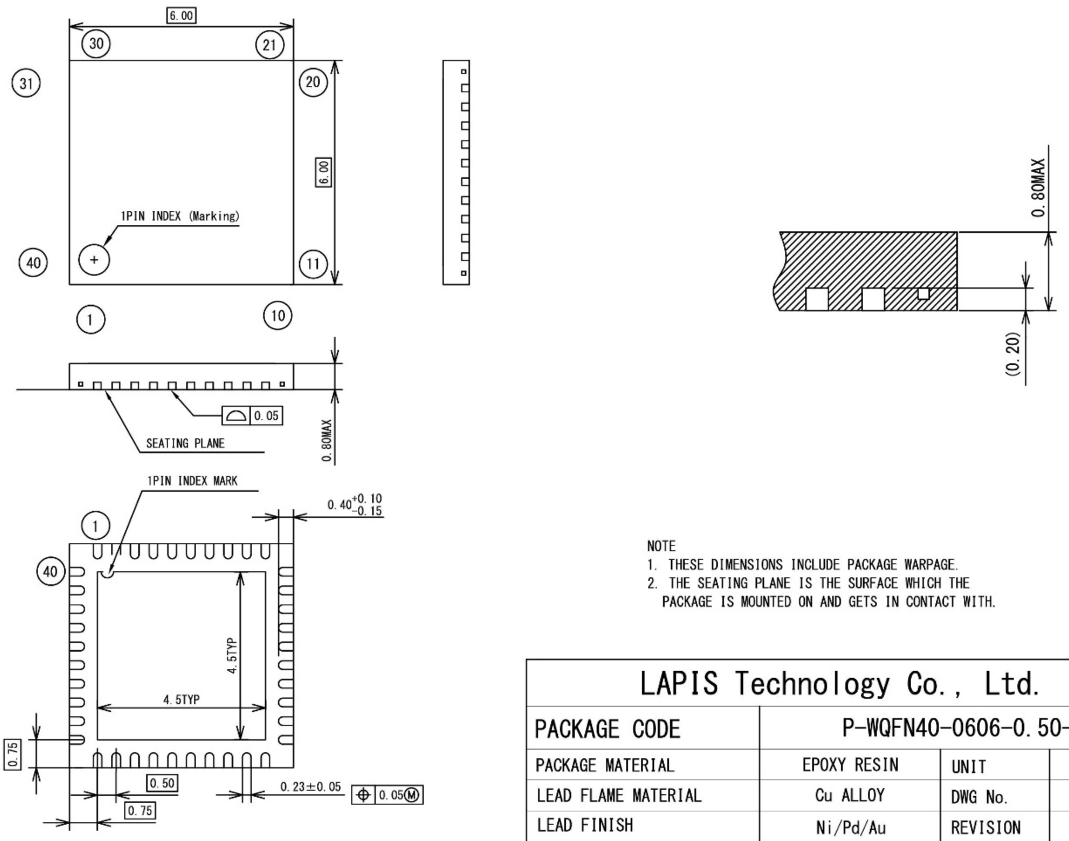
[Note] Notes for Mounting the Surface Mount Type Package

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.

40pin WQFN



NOTE
1. THESE DIMENSIONS INCLUDE PACKAGE WARPAGE.
2. THE SEATING PLANE IS THE SURFACE WHICH THE PACKAGE IS MOUNTED ON AND GETS IN CONTACT WITH.

LAPIS Technology Co., Ltd.			
PACKAGE CODE	P-WQFN40-0606-0.50-63		
PACKAGE MATERIAL	EPOXY RESIN	UNIT	mm
LEAD FLAME MATERIAL	Cu ALLOY	DWG No.	QSL-39752
LEAD FINISH	Ni/Pd/Au	REVISION	7
SOLDER THICKNESS	Au/Pd Max0.01/Max0.15	1st ISSUE	Jun/28/2006
PACKAGE MASS (g)	0.087TYP.	REVISED	Oct/23/2020

(Unit : mm)

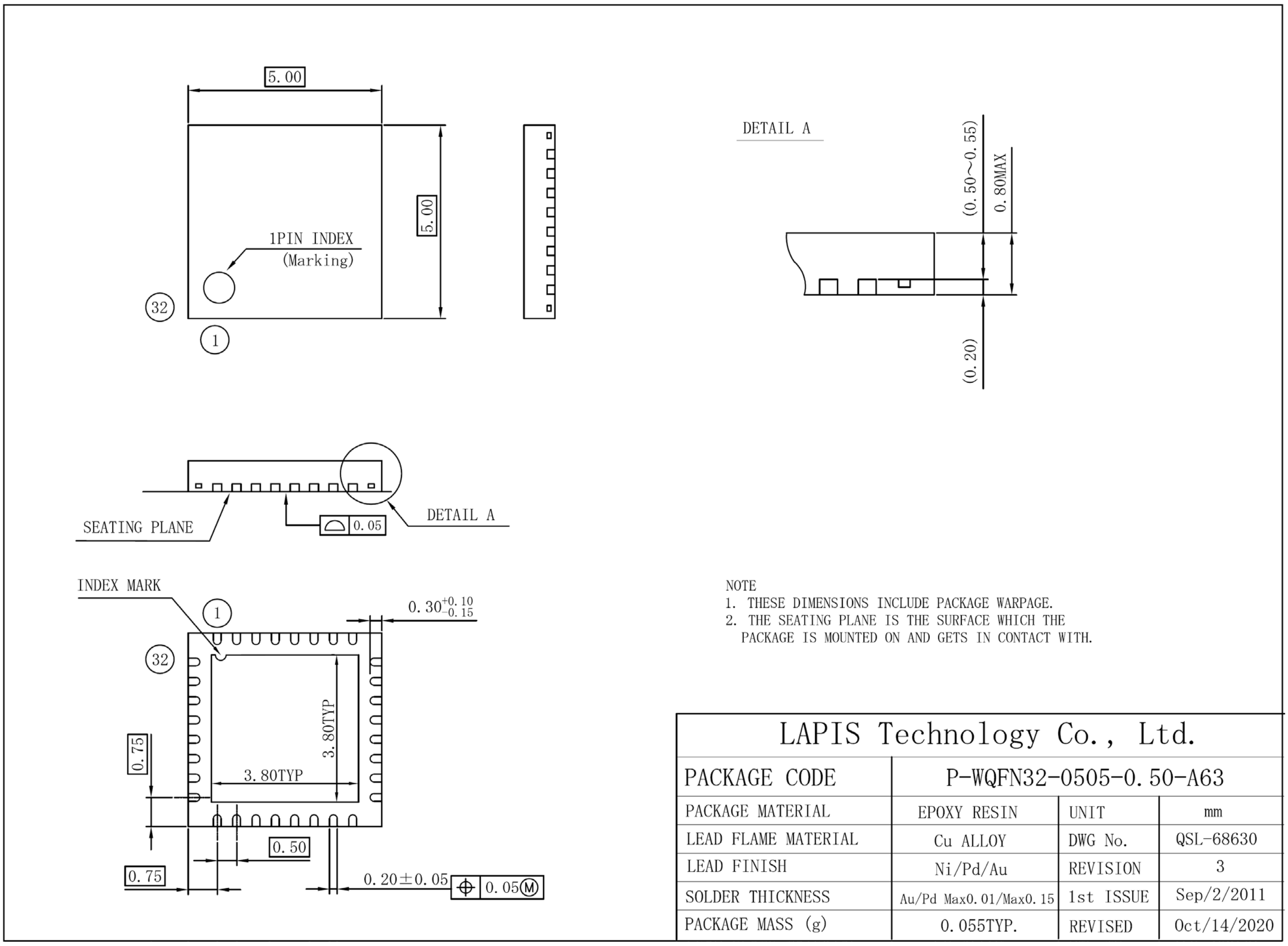
[Note] Notes for Mounting the Surface Mount Type Package

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 WQFN



NOTE
1. THESE DIMENSIONS INCLUDE PACKAGE WARPAGE.
2. THE SEATING PLANE IS THE SURFACE WHICH THE PACKAGE IS MOUNTED ON AND GETS IN CONTACT WITH.

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)

[Note] Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

[Note] Notes for the package with exposed die pad

The die pad is exposed on the bottom of WQFN package. Make the die pad electrically open when soldering onto the PCB.

Appendix C Instruction Execution Cycle

ML62Q2500 group 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	-
		[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	ER*n*	ER*m*	1	1	-	-	-	-
		#imm7	1	1	-	-	-	-
MOV	R*n*	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	-
		[ERm]	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	D*bitadr*	2	3	-	-	1	-
	R*n.bit_offset*	1	1	-	-	-	-
RB	D*bitadr*	2	3	-	-	1	-
	R*n.bit_offset*	1	1	-	-	-	-
TB	D*bitadr*	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	-	-	-	-

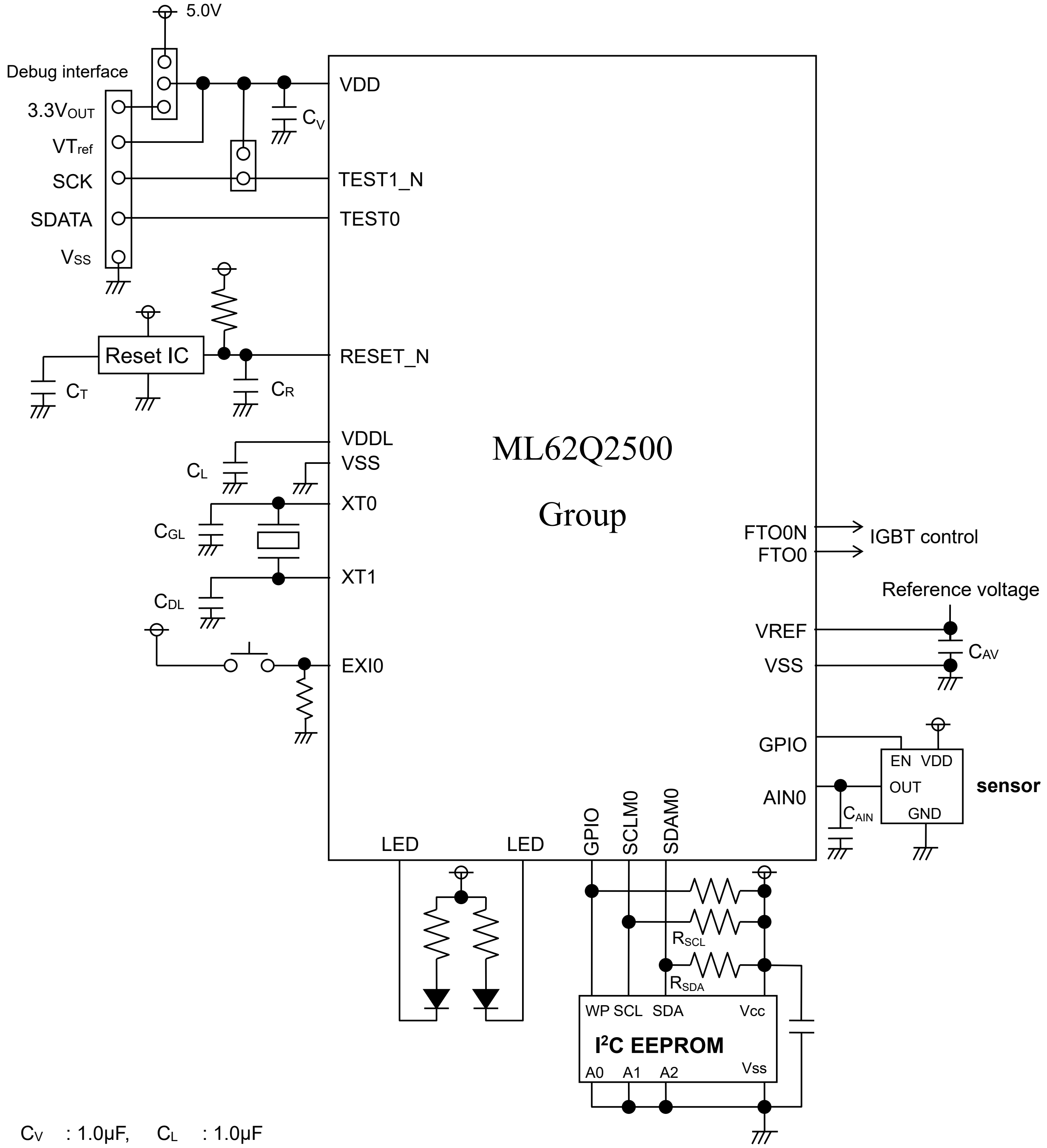
Interrupts

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,	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.

R_{SCL}, R_{SDA} : 5kΩ or smaller

LED : P02~P07, P10~P17, P20~P27, P30~P37, P52~P57, P60~P62, P70~P73

Reset IC : BU4217 (ROHM, Nch open drain output)

[Note]

- **Place the capacitor for VDDL pin as close to the LSI power pins as possible.**

Appendix E List of Notes

This Check List has important notes to prevent commonly-made programming mistakes and frequently overlooked or misunderstood hardware specifications of the LSI. Check each note listed in chapter by chapter when coding or evaluating the program.
Number in a mark [ ] shows section number to refer.

Common to all Chapters

□ Please see the “Notes” and the “Notes for product usage” in this document front pages.
□ Word access is available for registers with the word symbol. Specify an even address for the word access. See "List of Registers" in each chapter.
□ Registers for unequipped channels are not available to use. They return 0x0000 for reading. See "List of Registers" in each chapter.

1. Overview

□ [1.3.4] Terminate unused input pins according to the Table 1-5 in order to avoid unexpected through-current in the pins.

2. CPU and Memory Space

□ [2.5] CSR[3] is unused on the ML62Q2500 group. The data of CSR "0x8 to 0xF" are handled as "0x0 to 0x7".
□ [2.5] The Code Option area (64 bytes) is not available for the program code area. For details of Code Option settings, see Chapter 30 "Code Option" and make sure the setting data is correct.
□ [2.5] It is recommended to fill unused areas with data "0xFFFF" (BRK instruction) in the program memory space to ensure failsafe using the generation tool of the ROM code data. See its manual for details on how to use. See "nX-U16/100 Core Instruction Manual" for details of the BRK instruction.
□ [2.5, 2.6] Do not read or program unused areas to prevent the CPU works incorrectly.
□ [2.6] The contents of the RAM area are undefined at power-on and system reset. Initialize this area by the software.
□ [2.8.2] If the entire LSI is reset through a system reset, the remapping function is disabled as the REMAPADD register is restored with the initial value.

3. Reset Function

□ [3.3.1] The BRK instruction reset only initializes the CPU if ELEVEL is 2 or higher. Peripheral circuits and other circuits are not initialized. Use the pin reset or the watchdog timer (WDT) reset to surely initialize the LSI when an abnormality is detected.
□ [3.3.1] Command reset in on-chip debug does not reset to crystal oscillation circuit and VLS parts. Do initialization of these functions by writing SFRs on debug, if needed. See Chapter 28 for details.
□ [3.3.2] In system reset mode, the contents of data memory (RAM) and SFRs that have an undefined initial value are not initialized. Initialize them by the software.
□ [3.3.4] In case of instantaneous power failure and a pulse shorter than the power-on reset reaction time is asserted to VDD, MCU may not get reset and it may malfunction. In that case, please have preventive measures such as using bypass capacitor to avoid the instantaneous voltage drop or using pin reset to initialize MCU.

4. Power Management

□ [4.1.3] In order to improve the noise resistance, place the inter-power supply bypass capacitor (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.
□ [4.1.3] The internal logic voltage (VDDL pin output) is unavailable to use for an external device voltage.
□ [4.2.2] Writing to the stop code acceptor is invalid on the condition both interrupts enable bits and interrupt request bits are "1", it will not get enabled for entering to the STOP/STOP-D mode.
□ [4.2.3] The operating state does not enter the standby mode under some conditions. See “4.3.2.6 Note of entering to the standby mode” for detail conditions.
□ [4.2.3] When an interrupt enabled in the interrupt enable registers (IE0 to IE7) is generated on the condition of MIE flag of the program status word (PSW) is "0", it cancels the standby mode only and the CPU does not go to the interrupt routine. For more details about MIE flag, see "nX-U16/100 Core Instruction Manual".
□ [4.2.3] Insert two NOP instructions in the next to the instruction of that sets HLT, STP, HLTH and STPD bit to "1". The operation without the two NOP instructions is not guaranteed.
□ [4.2.6] Do not enter the standby mode when the SOFTR bit is "1". Ensure the SOFTR bit is "0" before entering the standby mode.
□ [4.2.9, 4.2.13] The DCKACC/RSEACC bit can be set to "0" when the multiplication/division library "muldivu8.lib" is specified. See a manual of the multiplication/division library for how to use.
□ [4.3.2.6] Note of entering to the standby mode
□ [4.3.2.7] Note on Return Operation from Standby Mode

- □ [4.3.2.7] Since up to two instructions are executed during the period between the release of standby mode and a transition to interrupt processing, place two NOP instructions next to the instruction set for the standby mode. When a master interrupt enable (MIE) flag of the program status word (PSW) in the nX-U16/100 CPU core is "1", following the execution of the two NOP instructions, the interrupt transition cycle will be executed and execution of the instruction for interrupt routine begins. If MIE is "0", following the execution of the two NOP instructions, the instruction execution is continued from the one that follows the NOP instruction without transition to the interrupt.
- □ [4.3.2.9] When the FHWUPT register is set to "0x01", the frequency of PLL oscillation clock gradually increases and reaches the target frequency chosen by the code option before approx. 2 ms elapse. The PLL oscillation clock during this time period can be used for the SYSCLK, however, accuracy of the frequency is not guaranteed.
- □ [4.3.3] If the clock supply is only stopped without resetting each peripheral circuit using the block control function, it may cause the output levels of the timer and communication pins to be fixed, causing the excess current to flow. Also, in the successive approximation type A/D converter, the circuits may stop their function with the current kept flowing.

5. Interrupts

- □ [5.2.6 ~ 5.2.9] There is a risk of clearing other request flags of This IRQ 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.
- □ [5.2.10] 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 (IE01 to IE67 registers are "0x00").
- □ [5.2.10] Enable the interrupt level control function by setting the ILE bit to "1" when the interrupt is disabled(IE0 to IE7 registers are "0") or master interrupt enable flag(MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.
- □ [5.2.13 ~ 5.2.20] Write to this register when the interrupt is disabled (IE01 to IE67 registers are "0x0000") or the master interrupt enable flag (MIE) is "0", otherwise, an interrupt may occur with an unexpected interrupt level.
- □ [5.3] The WDT interrupt (WDTINT) is a non-maskable interrupt. If the non-maskable interrupt occurs while an interrupt processing is in progress, abort the interrupt processing and proceed with processing the non-maskable interrupt preferentially regardless of multiple interrupts enabled/disabled.
- □ [5.3] For failsafe, define unused all interrupt vectors. If an unused interrupt occurs, it may indicate the possibility that the CPU went out of control. It is recommended to cause the WDT overflow reset to occur using the infinite loop to initialize the LSI.
- □ [5.3.4] Notes on Interrupt Routine (with Interrupt Level Control Disabled)
- □ [5.3.4] Do not enable interrupts in a subroutine called from an interrupt routine for which multiple interrupts are disabled. Otherwise, the program may run out of control when multiple interrupts occur.
- □ [5.3.5] For processing of non-maskable interrupt, follow the flow chart "When multiple interrupts are enabled". Registers that should be saved in the stack are ELR2 and EPSW2.
- □ [5.3.5] When programming in C, it is not required to write program codes for saving/restoring registers because they are generated in the C compiler. However, program codes for enabling/disabling interrupts through EI and DI instructions and for writing to the current interrupt level management register (CIL) must be written. See Section 5.3.6 "How To Write Interrupt Processing When Interrupt Level Control Enabled" for the specific program description.
- □ [5.3.6.1] Do not enable interrupts in a function called from a function for which multiple interrupts are disabled. Otherwise, the program may run out of control when the multiple interrupts occur.

6. Clock Generation Circuit

- □ [6.1.2] After the power-on or the system reset, LSCLK0 (32.768 kHz) is initially chosen as SYSCLK.
- □ [6.1.3] Assign HCKO function to only one LSI pin.
- □ [6.3.1] The LCKO output operation is not guaranteed in the HALT-D mode.
- □ [6.3.1.2] 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.
- □ [6.3.1.2] Note that oscillation may stop due to condensation.
- □ [6.3.1.2] When switching to the low speed crystal oscillation clock, ensure to use the interrupt referring to the Section 6.3.1.3 "Low-Speed Clock Control".
- □ [6.3.2.2] When the XT32K is used for LSCLK0, the high-speed clock may become an unintended frequency due to external factors such as noise, and the MCU may operates abnormally. Please evaluate enough the apparatus/system which implemented this product.
- □ [6.3.4] While the CPU is running with the low-speed clock, if running the peripheral circuits with the high-speed clock which can frequently generate interrupts, the operation may fail to function properly due to the CPU becoming incapable of processing interrupts in time. If interrupts frequently occur for reasons such as short interrupt cycles of peripheral circuits, take into account the operating frequency of the CPU so that it can process interrupts in time.

7. Low Speed Time Base Counter

- □ [7.2.2] A time base counter interrupt may occur depending on the timing to write to the LTBR01. See the program example for initializing described in Section 7.3.1 "Operation of the Low-speed Time Base Counter".

□ [7.2.2] Read the LTBR01 register twice to verify the data to prevent reading uncertain data while counting-up.
□ [7.2.3] Stop counter LTBR0 (i.e. set 0 to TB0RUN bit), before TB0CK bit is configured.
□ [7.2.5] A time base counter interrupt may occur depending on a write timing to the LTBINT. See the program example for initializing described in "7.3.1 Operation of the Low-speed Time Base Counter".
□ [7.3.1] After writing to the LTBR01 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.
□ [7.3.2] 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.

8. 16-bit Timer

□ [8.2.2] Set TMHnD when the 16-bit timer n is stopped (THnSTATL bits of TMHSTAT register are "0").
□ [8.2.2] When "0x0000" is written in TMHnD in the 16-bit timer mode, "0x0001" is set in TMHnD.
□ [8.2.2] Set TMHnD so that the timer output frequency is 1MHz or less, when timer output is used.
□ [8.2.3] Read the TMHnC register twice to verify the valid data to prevent reading uncertain data while counting-up, if a source of timer clock is as different as one of system clock.
□ [8.2.4] Set TMHnMOD when the timer n is stopped (THnSTAT bits of TMHSTAT/TMHXSTAT register are "0"). If it is changed while it is operating, the operation is not guaranteed.
□ [8.3.2] After 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.
□ [8.3.2] 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.

9. Functional Timer

□ [9.1.3] Assign FTOn, FTOnN to only one LSI pin each.
□ [9.2.2] When 0x0000 is written in this register, 0x0001 is set and the read value is also becomes 0x0001.
□ [9.2.2] Set FTnP so that the functional timer output frequency is 3MHz or less, when its output is used.
□ [9.2.3, 9.2.4] In timer mode , a data set in the FTnEA/FTnEB register must be less than that set in the FTnP register.
□ [9.2.3] In PWM1/2 mode, a data set in the FTnEA register must be 0xFFFF or less than that set in the FTnP register.
□ [9.2.4] In PWM1 mode, a data set in the FTnEB register must be 0xFFFF or less than that set in the FTnP register.
□ [9.2.5] In the PWM2 mode, the data set in the FTnDT register must be less than that set in the FTnEA register.
□ [9.2.5] 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.
□ [9.2.6] Read FTnC register twice to verify the data to prevent reading uncertain data while counting-up according to need.
□ [9.2.8] Set the FTnMOD register when the FTMn is stopped.
□ [9.2.8] Initialize this peripheral with block reset before changing to another mode, if it is in the operation state once.
□ [9.2.10] The input pulse width must have two timer clocks or longer if FTnSTSS=0.
□ [9.2.10] The counter forcibly stops and does not run when the emergency stop trigger source is the same as the trigger event source with the FTnETG = 1 and FTnEMGEN = 1.
□ [9.2.11] 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.
□ [9.2.11] The trigger may occur immediately after setting the FTnTRG1 register in the trigger event enabled.
□ [9.2.13] If the FTnINTS register is not zero, a request to interrupt controller is not given when a new interrupt occurs. Clear the FTnINTS register with the FTnINTC register before that time.

10. Watchdog Timer

□ [10.1.1] 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.
□ [10.2.2] 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] See the data-sheet for frequency accuracy of RC1K.
□ [10.3.1] In the STOP/STOP-D mode, the WDT timer is stopped.
□ [10.3.3] When using the window function enabled mode, always define a WDT interrupt function even though no WDT interrupt occurs.
□ [10.3.3] In the watchdog timer (WDT) interrupt function, as the interrupt level (ELEVEL) of the CPU program status word (PSW) becomes "2", the WDT counter is unable to get cleared. Clear the WDT when the ELEVEL is "0" or "1". It is recommended that the WDT counter is cleared at one place in the main loop of the program as a fail-safe.

11. Synchronous Serial Port

□ [11.1.4] Be sure to use the SIN0/SOUT0/SCLK0 ports with combination in the Fig.11-3, and assign each function to only one LSI pin.
□ [11.2.5] Be sure to set the SIOnMOD register while communication is stopped (SnEN=0). If it is rewritten during communication, data may be transmitted or received incorrectly.
□ [11.3.3] To prevent an overrun error after the first reception, read the SIOnBUF register before setting the SnEN bit to "1".
□ [11.3.4] 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) in the clock type1 slave mode.

12. Synchronous Serial Port with FIFO

□ [12.1.4] Be sure to use the SDIF0/SDOF0/SCKF0/SSNF0 ports with combination in the Fig.12-3, and assign each function to only one LSI pin.
□ [12.2.7] Write "1" to SF0IRQ bit while there is any unprocessed interrupt source and processing all the interrupt sources before exiting the interrupt vector will cause re-entry to the interrupt vector with no interrupt source after exiting the interrupt vector. Ensure to write "1" before exiting the interrupt vector.

13. I^{2}C Bus

□ [13.1.4] Use external pull-up resistors for SDA pin and SCL pin referring to the I^{2}C bus specification. The internal pull-up resistors is unsatisfied the I^{2}C bus specification. See the data sheet for each product for the value of internal pull-up resistors.
□ [13.1.4, 13.2.2] If powering off this LSI in the slave mode, it disables communications of other devices on the I^{2}C bus. Keep this LSI powered on when it works as a slave mode until the master device is powered off.
□ [13.1.4, 13.2.2] Do not connect multiple master devices on the I2C bus when using the master function.
□ [13.2.2] All SFRs are shared in master mode and slave mode. If switching master/slave mode, set “0” I2U0EN bit of I2UMOD register, then Change mode and do reconfiguration each SFRs.
□ [13.3.4] Update it without a bit access instructions in the control register setting wait state.
□ [13.3.4] When the I2U0ST/I2M0ST bit is "1", write other bits of I2U0CON/I2M0CON register in the control register setting wait state.
□ [13.4.4] If system clock is extremely slower than the communication speed, the data transmission/reception can be failed.
□ [13.4.4] Before releasing the communication wait state, change the system clock enough speed for the communication.
□ [13.4.5] To be disable the wake-up from standby mode by matching the slave address, Stop the operation by resetting I2U0EN bit to "0" before entering STOP/STOP-D/HALT-D mode.
□ [13.5.4] When the slave device uses the clock stretch function which holds the SCLU0 pin at "L" level, the time t_{CYC} and time t_{LOW} are extended.
□ [13.6.2] 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).
□ [13.6.4] The master device should Wait for the SYSCLK to be supplied in order to transmit the start condition after wakeup from the STOP-D/HALT-D mode by slave address matching.
□ [13.6.4] It is supported the Standard/Fast mode (to max. 400 kbps) in the STOP-D／HALT-D mode.

14.UART

□ [14.2.1] When the DCKUAn of Block control register 2; BCKCON2, A reading value of relevant SFR is 0x00/0x0000. However values written to the SFR is kept. A reading value of relevant SFR is setting value after the DCKUAn bit is returned “0”. See Chapter 4 “Power Management” as for block control register.
□ [14.2.6] Do setting for used ports and the mode/baud rate before setting “1” to UnEN bit.
□ [14.2.7] Be sure to set the UAn0MOD register while communication is stopped (Un0EN=0).
□ [14.2.9, 14.2.10] Be sure to set the UAnBRT and UAnBRC register while communication is stopped (UnEN=0). Do not rewrite it during communication.
□ [14.3.4] The transmission is start when setting “1” to UnEN bit of UAnCON with the UnFUL bit =1. Write “1” to the UnFULC bit in the UAnSTAC register to reset the UnFUL bit, and then set “1” to the UnEN to allow transmission/reception, if the transmission data is not ready and the reception is permitted first.
□ [14.3.5.3] When designing the system, consider the difference of the baud rate between the transmission side and reception side, a delay of the start bit detection, signal degradation and noise influence, then adjust the baud rate and reception timing to ensure sufficient receiving margin.

17. GPIO

□ [17.2.4] The P00 pin is initially configured as the input with pull-up resistor. If input "L" level at an initial setting, the input current flows.
□ [17.2.5 ~ 17.2.8] Be sure to set the PnMODm (m=0 to 7) registers before setting EICON0, EIMOD0 and IE1 registers. If setting the PnMOD01 register when the interrupt is enabled, unexpected interrupts may happen.
□ [17.2.5 ~ 17.2.8] It is recommended to enable the output after setting a peripheral and shared function to prevent the unexpected output.

□ [17.2.5 ~ 17.2.8] Don't set un-assigned shared functions on the PnmMD3-0 bits.

□ [17.2.11] PI0 and PI1 are unavailable to use as input ports when using the crystal resonator for the oscillation clock. Also, PI1 is unavailable to use as an input port when using the XT1 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.

□ [17.3.8] Notes for using the P00/TEST0 pin

18. External Interrupt Function

□ [18.2.3] If chosen high-speed clock as sampling clock source, it works without sampling when the high-speed clock does not supply; it include stop by entry to standby mode. Set to LSCLK0 as sampling clock if needed.

□ [18.2.3] In the STOP/STOP-D mode, it works without sampling.

19. CRC Calculator

□ [19.2.2 ~ 19.2.5] 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.

□ [19.2.3, 19.2.5] If an address set to CRCEAD and CRCESEG is smaller than one of CRCSAD and CRCSSEG, the calculation does not execute. 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.

□ [19.3.2] 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.

□ [19.3.2] 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.

22. Voltage Level Supervisor

□ [22.2.2] Even if resets other than the POR and RESET_N pin reset occurred, the VLS0 remains running.

□ [22.2.3] There is a limitation in each mode for entering the STOP/STOP-D mode while the VLS0 is running.

□ [22.2.5] In the STOP/STOP-D mode, the VLS works without sampling regardless the setting in VLS0SM1 and VLS0SM0 bit.

□ [22.3.1.1, 22.3.1.2] Entering the STOP/STOP-D mode is not allowed during the VLS 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.

□ [22.3.1.1] The initial value of the VLS detection voltage is 1.85V, so the MCU becomes in reset mode when the VDD is 1.85V or lower and VLS0 is specified as supervisor mode with the reset function. Therefore, set the detection voltage before enabling the VLS0 operation.

□ [22.3.1.1] If you want to use the VLS0 reset function like a reset IC, start the VLS when the CPU initially runs at the low-speed clock after the power up.

□ [22.3.1.2] When VLS0 is stopped (VLS0EN bit="0") while the VDD is lower than the specified threshold voltage (VLS0F bit="1"), the VLS0 interrupt is generated.

□ [22.3.2.1, 22.3.2.2] 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").

□ [22.3.2.2] If V_{DD} is higher than the specified threshold voltage, the VLS0 interrupt is not generated.

23. Successive Approximation Type A/D Converter

□ [23.1.3] When using the SA-ADC, set PnmIE bit and PnmOE bit of port n mode register 01/23/45/67 (n: port number 1, 2, 3, 7, m: bit number 0 to 7) to "0" as "Disable input" and "Disable output", otherwise a shoot-through current may flow.

□ [23.1.3] While the A/D converter is operating, an influence of the noise is reducible by preventing the switching of neighboring pins or A/D converting in the HALT mode.

□ [23.2.3] Start the A/D conversion with one or more channels chosen by the SA-ADC enable registers (SADEN0 and SADEN1). If no channel is chosen, the operation does not start.

□ [23.2.3] Enter STOP/STOP-D mode after checking SARUN bit is "0". It does not enter the STOP/STOP-D mode when the SARUN bit is "1".

□ [23.2.3] When SACK2 to 0 bits are set to 0x7, it takes max. 3 clocks of the low-speed clock (LSCLK0) to start or stop the A/D conversion after setting or resetting the SARUN bit.

□ [23.2.8, 23.2.9] 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".

□ [23.2.13] 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 SAULS13 to SAULS00 or by resetting the LSI.

□ [23.2.13] When performing the A/D conversion only one time (SALP bit =0), confirm the bit of SAULS13 to SAULS00 is "0" before setting SARUN bit to "1".

□ [23.2.13] When performing the consecutive scan A/D conversion (SALP bit =1), confirm the bit of SAULS13 to SAULS00 is "0", before the next A/D conversion ends.
□ [23.4] Notes on SA-ADC

26. Flash Memory

□ [26.2.2] 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.
□ [26.2.4, 26.2.5] Specify a segment address to the FLASHSEG at first, because it determines whether the programming is for program memory space or data flash memory.
□ [26.2.4, 26.2.5] During programming data-flash, a CPU can execute instruction by the back ground operation function; BGO. Confirm FDPRSTA bit of FLASHSTA register for complition of programming.
□ [26.2.4, 26.2.5] Erase data in the addresses in advance. Programmed data without erase is unguaranteed.
□ [26.2.4, 26.2.5] Do not read or program unused areas to prevent the CPU works incorrectly.
□ [26.2.7, 26.2.8] A flash memory data in processing to program is not guaranteed, if this register is written any data when FLASHSTA is not 0x0.
□ [26.2.9] Perform the erasing or programming after checking the FDERSTA bit or FDPRSTA bit are "0". Do not perform the erasing or programming when either the FDERSTA bit or the FDPRSTA bit is "1".
□ [26.3.1] Notes on Debugging Self-programming Code
□ [26.3.2] Only erase areas irrelevant to program processing. If erasing the area where program processing is in progress, the LSI works incorrectly.
□ [26.3.2] During block/sector erasing, the CPU stops the operation for maximum 50 ms whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.
□ [26.3.2, 26.3.3] For block/sector erasing, place two NOP instructions following the instruction used to set FERS/FSERS bits of the FLASHCON register to "1".
□ [26.3.2] Only erase areas irrelevant to program processing. If erasing the area where program processing is in progress, the LSI works incorrectly.
□ [26.3.2] During the programming, the CPU stops the operation for maximum 80 μs whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.
□ [26.3.2] For data programming setting, place two NOP instructions following the instruction used to set the programming data in the FLASHD1 register.
□ [26.3.3] The CPU continues program processing even while data flash erasing is in progress. An entering to the STOP/STOP-D/HALT-D/HALT-H mode is not available during the erasing. In addition, set the FSELF bit of the FLASHSLF register to "0" after the erasing is completed.
□ [26.3.3] The data flash area is unreadable during erasing.
□ [26.3.3] The CPU continues program processing even while data flash programming is in progress. An entering to the STOP/STOP-D/HALT-D/HALT-H mode is not available during the programming. In addition, set the FSELF bit of the FLASHSLF register to "0" (erase/program disabled) after the programming ended.
□ [26.3.3] For data programming setting, place two NOP instructions following the instruction used to set the programming data in the FLASHD0L register.
□ [26.3.4] Notes on use of self-programming
□ [26.4.3] Accessing to the program code area is performed in units of four bytes. Set four byte boundaries (0H/4H/8H/CH) for lower four bits of the address. Accessing to the data flash area is performed in units of one byte.
□ [26.4.3] All commands except some confirmation commands (*1) are reflected when a next command is sent.
□ [26.4.5.1 ~ 26.4.5.4, 26.4.6.2 ~ 26.4.6.4] Transmit command to avoid a timeout. See Section "26.4.3.1 Command Timing".
□ [26.4.5.2 ~ 26.4.5.4, 26.4.6.2 ~ 26.4.6.4] Transmit any command after 'initial setting command (7)'/'stack clear command 3' if other command will not be transmit.

28. On-Chip Debug Function

□ [28.3] Make TEST1_N pin able to be connected to VDD with a jumper or something when not using the on-chip debug function.
□ [28.3] Validate the ROM code on user production board without the On-chip emulator.
□ [28.3] Disconnect On-chip emulator when measuring the current consumption of the target system. If On-chip emulator remains connected, the current consumption increases as the on-chip debug circuit inside the LSI works for the communication.
□ [28.3] When using the 3.3 VOUT power supply of On-chip emulator, do not apply power of the target system to the VDD pin of LSI. If both power supplies are connected, On-chip emulator may be damaged, or an electric shock or fire may occur.
□ [28.3] LSI used to debug a program is not covered by the product warranty. Do not use the LSI for mass-production.
□ [28.3] A reset due to unused ROM area access does not occur in the on-chip debug mode regardless of code option settings.
□ [28.3] A RAM parity error reset does not occur in the on-chip debug mode and the break operation occurs instead.
□ [28.3] If the contents of the data memory are displayed in the debugger in a state where a RAM parity error may occur (including when the RAM is not initialized), a RAM parity error may occur even if the RAM area is not displayed.
□ [28.3] The all interrupts and watchdog timer operation always stop while the debugger is in the break state.

□ [28.3] 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.

□ [28.3] If adding an external capacitor to the TEST1_N pin, prepare a jumper function on the board so that the capacitor gets dis-connectable when using the debugger or Flash multi-writer.

29. Safety Function

□ [29.2.8] If the MCISTATL register is not zero, a request to interrupt controller is not given when a new interrupt occurs. Clear the MCISTATL register with the MCINTCL register before that time.

□ [29.3.2] CSR[3] is unused on the ML62Q2500 group. The data of CSR "0x8 to 0xF" are handled as "0x0 to 0x7".

□ [29.3.3] 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.

30. Code Option

□ [30.2.1] There are available to read the code option values from SFRs, if INITE flag bit of Reset Status Register (RSTAT) is "0".

□ [30.3] For the code option data definition, always use the dw directive instruction to configure the data in the unit of word.

31. Auxiliary Function

A. SFR List

□ Access "Reserved" register is not guaranteed. Please do not access them.

B. Package Dimensions

□ The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact a ROHM sales office for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

□ The die pad is exposed on the bottom of WQFN package. Make the die pad electrically open when soldering onto the PCB.

C. Instruction Execution Cycle

D. Application Circuit Example

□ Place the capacitor for VDDL pin as close to the LSI power pins as possible.

Revision History

REVISION HISTORY

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL62Q2500-01	2022.10.21	—	—	1st edition.
FEUL62Q2500-02	2022.11.10	1-3	1-3	[1.1] Updated description in the power management section, and corrected description about HSCLK and clock gear
		9-8, 9-9, 9-12	9-8, 9-9, 9-12	[9.2.3][9.2.4][9.2.7] Corrected description of clearing FTnFLGA/FTnFLGB
		9-14	9-14	[9.2.8] Added note
		-	9-38	[9.3.4.2] Added section; clearing FTnFLGA/FTnFLGB bit
		E-*	E-*	As each chapter is updated
FEUL62Q2500-03	2023.04.17	6-28	6-28	[6.3.2.2] Corrected description in a flow.
		9-11	9-11	[9.2.6] Updated note as for the counter reading
		9-20	9-20	[9.2.13] Added note as for clearing interrupt status
		23-8	23-8	[23.2.2] Added a condition of 4MHz in the Fig 23-3
		23-31	23-31	[23.4.1] Added reference value of R_2 at V_{DD}≥2.7V and corrected V_{DD} to V_{REF}
		28-1	28-1	[28.1.1] Deleted trace function
		28-3	28-3	[28.3] Added note as for a RAM parity error
		29-9	29-9	[29.2.8] Added note as for clearing interrupt status
		E-*	E-*	As each chapter is updated
FEUL62Q2500-04	2024.03.26	1	1	Updated note
		1-1	1-1	[1.1] Added use application
		1-4, 1-5	1-4, 1-5	[1.1][1.1.1] Updated description for product name
		1-5, 1-6	1-5, 1-6	[1.1.1][1.1.2] Separated section
		4-21, 4-23	4-21, 4-23	[4.3.2.2][4.3.2.6] Corrected description as for HALT-H
		4-27	4-27	[4.3.4] Corrected Table 4-8
		6-3	6-3	[6.1.2] Corrected Table 6-4
		9-6	9-6	[9.2.1] Corrected access type of the FT1INTCL/H
		9-41	9-41	[9.3.6.1] Corrected setting flow of the event trigger.
		10-16	10-16	[10.3.3] Corrected note.
		11-17, 11-18	11-17, 11-18	[11.3.3] Corrected Figure11-6,11-7 and added note.
		26-5	26-5	[26.2.3] Corrected Table 26-1
		28-3	28-3	[28.3] Correct the note.
		E-6	E-6	Correct the note.