REJ09B0212-0100Z

Everywhere you imagine. RENESAS

8

3804 Group (Spec. H)

User's Manual

RENESAS 8-BIT CISC SINGLE-CHIP MICROCOMPUTER

740 FAMILY / 38000 SERIES

Rev. 1.00
Revision date: Jan 14, 2005

RenesasTechnology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of non-flammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

REVISION HISTORY	3804 Group (Spec. H) User's Manual

Rev.	Date	Description	
		Page	Summary
1.00	Jan 14, 2005	–	First edition issued

BEFORE USING THIS USER'S MANUAL

This user's manual consists of the following three chapters. Refer to the chapter appropriate to your conditions, such as hardware design or software development.

1. Organization

● **CHAPTER 1 HARDWARE**

This chapter describes features of the microcomputer and operation of each peripheral function.

● **CHAPTER 2 APPLICATION**

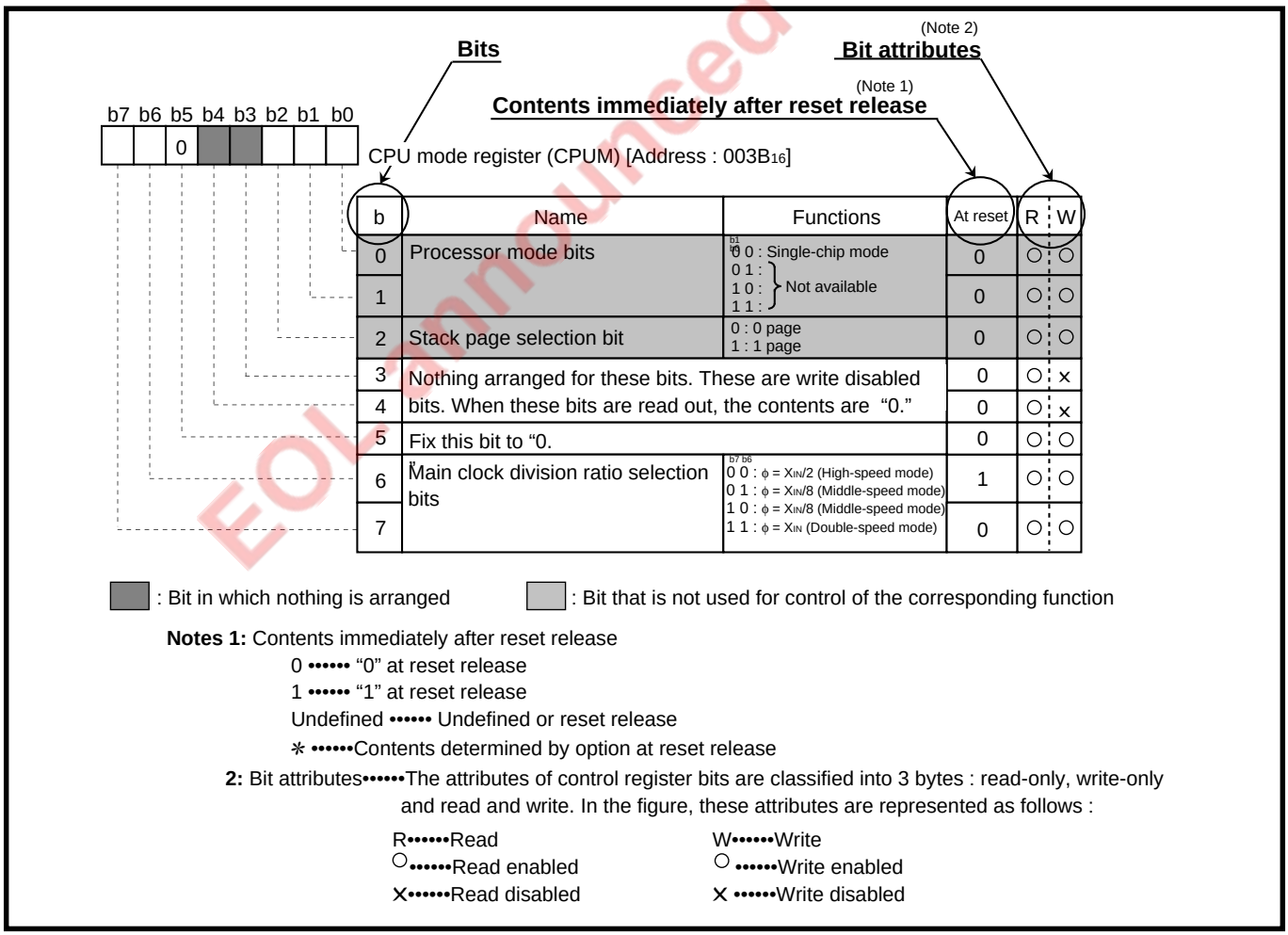
This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of relevant registers.

● **CHAPTER 3 APPENDIX**

This chapter includes necessary information for systems development using the microcomputer, such as the electrical characteristics, the notes, and the list of registers.

2. Structure of Register

The figure of each register structure describes its functions, contents at reset, and attributes as follows:



Notes 1: Contents immediately after reset release

0 •••••• "0" at reset release

1 •••••• "1" at reset release

Undefined •••••• Undefined or reset release

✻ ••••••Contents determined by option at reset release

2: Bit attributes••••••The attributes of control register bits are classified into 3 bytes : read-only, write-only and read and write. In the figure, these attributes are represented as follows :

R••••••Read

○••••••Read enabled

×••••••Read disabled

W••••••Write

○ ••••••Write enabled

× ••••••Write disabled

3. Supplementary

For details of related development tools and documents, refer to web page "3804 Group" on our website (http://www.renesas.com/eng/products/mpumcu/8bit/38000/index.html).

Table of contents

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

List of figures

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

List of tables

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

CHAPTER 1

HARDWARE

DESCRIPTION

The 3804 group (Spec. H) is the 8-bit microcomputer based on the 740 family core technology.
The 3804 group (Spec. H) is designed for household products, office automation equipment, and controlling systems that require analog signal processing, including the A/D converter and D/A converters.

FEATURES

- Basic machine-language instructions 71
- Minimum instruction execution time ..... 0.24 µs
  (at 16.8 MHz oscillation frequency)
- Memory size
  Flash memory ..... 60 K bytes
  RAM ..... 2048 bytes
- Programmable input/output ports 56
- Software pull-up resistors ..... Built-in
- Interrupts
  21 sources, 16 vectors .....
  (external 8, internal 12, software 1)
- Timers ..... 16-bit × 1
  8-bit × 4
  (with 8-bit prescaler)
- Watchdog timer ..... 16-bit × 1
- Serial interface
  Serial I/O1, 3 ..... 8-bit × 2 (UART or Clock-synchronized)
  Serial I/O2 ..... 8-bit × 1 (Clock-synchronized)
- PWM ..... 8-bit × 1 (with 8-bit prescaler)
- Multi-master I^2C-BUS interface ..... 1 channel
- A/D converter ..... 10-bit × 16 channels
  (8-bit reading enabled)
- D/A converter ..... 8-bit × 2 channels
- LED direct drive port 8
- Clock generating circuit ..... Built-in 2 circuits
  (connect to external ceramic resonator or quartz-crystal oscillator)
- Power source voltage
  In high-speed mode
  At 16.8 MHz oscillation frequency ..... 4.5 to 5.5 V
  At 12.5 MHz oscillation frequency ..... 4.0 to 5.5 V
  At 8.4 MHz oscillation frequency) ..... 2.7 to 5.5 V
  In middle-speed mode
  At 16.8 MHz oscillation frequency ..... 4.5 to 5.5 V
  At 12.5 MHz oscillation frequency ..... 2.7 to 5.5 V
  In low-speed mode
  At 32 kHz oscillation frequency ..... 2.7 to 5.5 V
- Power dissipation
  In high-speed mode ..... 27.5 mW (typ.)
  (at 16.8 MHz oscillation frequency, at 5 V power source voltage)
  In low-speed mode ..... 1200 µW (typ.)
  (at 32 kHz oscillation frequency, at 3 V power source voltage)
- Operating temperature range ..... –20 to 85°C
- Packages
  SP ..... 64P4B (64-pin 750 mil SDIP)
  FP ..... 64P6N-A (64-pin 14 × 14 mm QFP)
  HP ..... 64P6Q-A (64-pin 10 × 10 mm LQFP)
  KP ..... 64P6U-A (64-pin 14 × 14 mm LQFP)

<Flash memory mode>

- Power source voltage ..... Vcc = 2.7 to 5.5 V
- Program/Erase voltage ..... Vcc = 2.7 to 5.5 V
- Programming method ..... Programming in unit of byte
- Erasing method ..... Block erasing
- Program/Erase control by software command
- Number of times for programming/erasing 100

■Notes

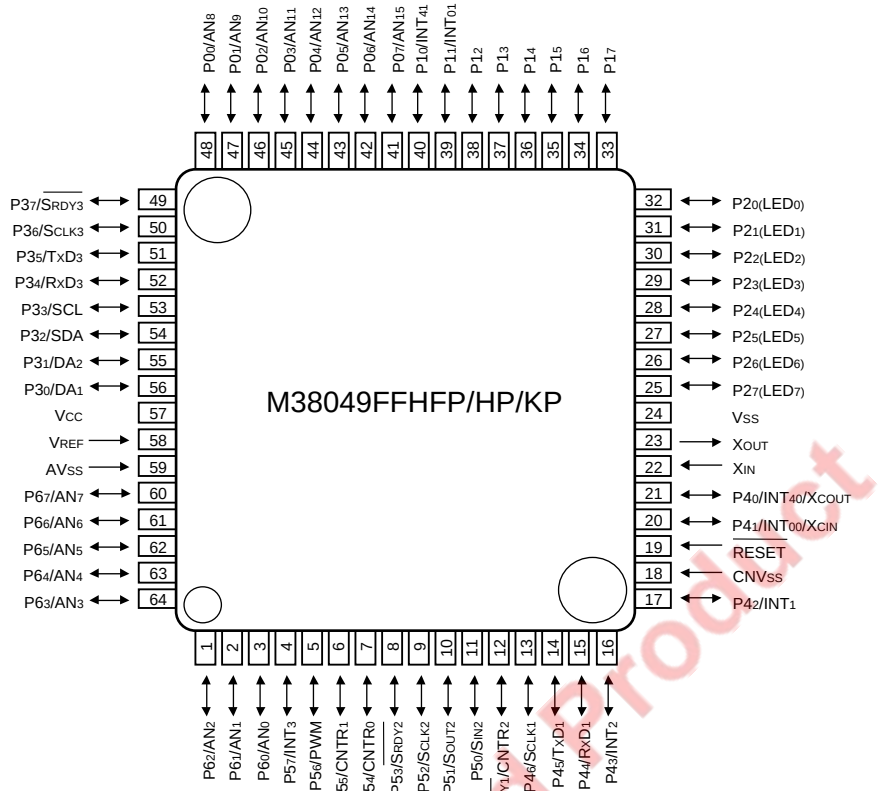
Cannot be used for application embedded in the MCU card.

Currently support products are listed below.

Table 1 Support products

Product name	Flash memory size (bytes)	RAM size (bytes)	Package	Remarks
M38049FFHSP	61440	2048	64P4B	Vcc = 2.7 to 5.5 V
M38049FFHFP			64P6N-A	
M38049FFHHP			64P6Q-A	
M38049FFHKP			64P6U-A	

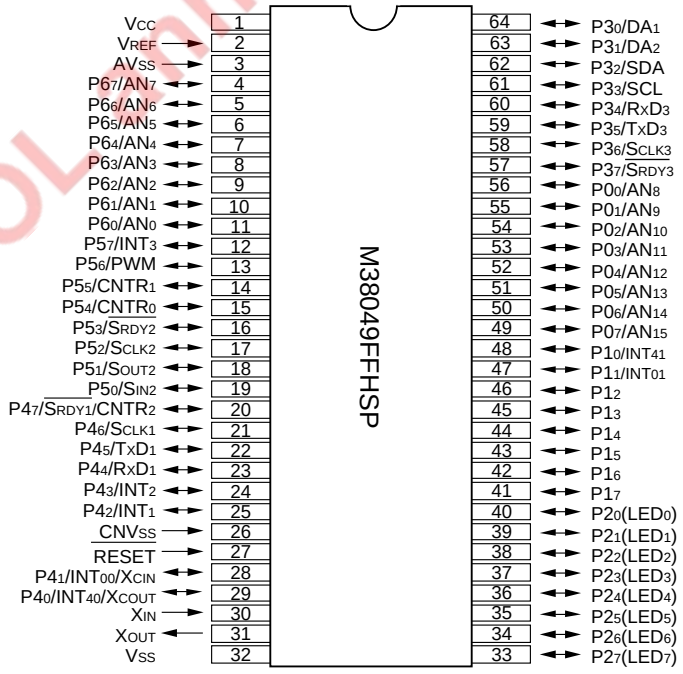
PIN CONFIGURATION (TOP VIEW)



Package type : 64P6N-A/64P6Q-A/64P6U-A

Fig. 1 3804 group (Spec. H) pin configuration

PIN CONFIGURATION (TOP VIEW)



Package type : 64P4B

Fig. 2 3804 group (Spec. H) pin configuration

FUNCTIONAL BLOCK

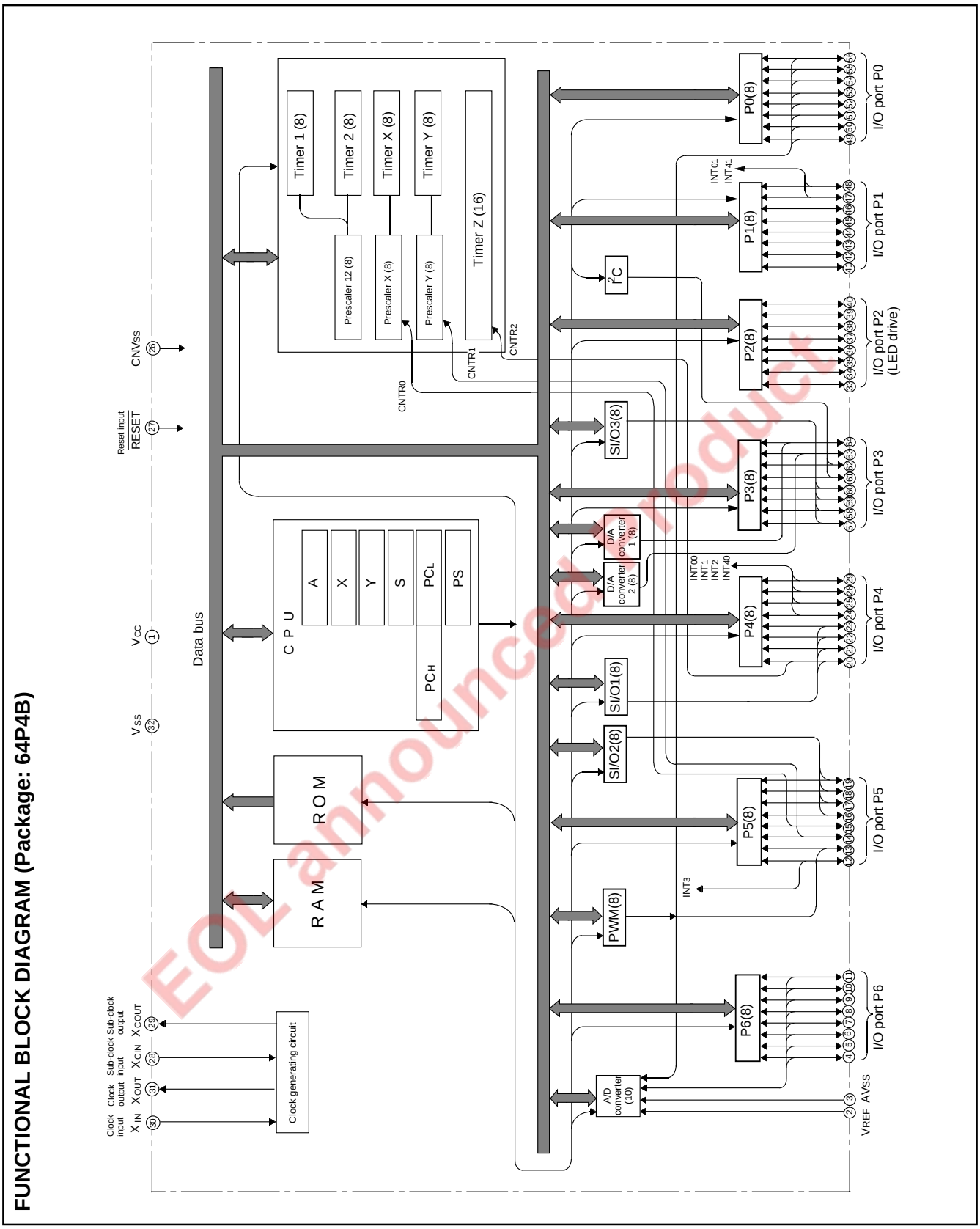


Fig. 3 Functional block diagram

PIN DESCRIPTION

Table 2 Pin description

Pin	Name	Functions	Function except a port function
VCC, VSS	Power source	•Apply voltage of 2.7 V–5.5 V to Vcc, and 0 V to Vss.	
CNVSS	CNVSS input	•This pin controls the operation mode of the chip. •Normally connected to VSS.	
VREF	Reference voltage	•Reference voltage input pin for A/D and D/A converters.	
AVSS	Analog power source	•Analog power source input pin for A/D and D/A converters. •Connect to VSS.	
$\overline{\text{RESET}}$	Reset input	•Reset input pin for active “L”.	
XIN	Clock input	•Input and output pins for the clock generating circuit. •Connect a ceramic resonator or quartz-crystal oscillator between the XIN and XOUT pins to set the oscillation frequency. •When an external clock is used, connect the clock source to the XIN pin and leave the XOUT pin open.	
XOUT	Clock output		
P00/AN8–P07/AN15	I/O port P0	•8-bit CMOS I/O port. •I/O direction register allows each pin to be individually programmed as either input or output. •CMOS compatible input level. •CMOS 3-state output structure. •Pull-up control is enabled in a bit unit. •P20–P27 are enabled to output large current for LED drive.	•A/D converter input pin
P10/INT41 P11/INT01	I/O port P1		•Interrupt input pin
P12–P17			
P20–P27	I/O port P2		
P30/DA1 P31/DA2	I/O port P3	•8-bit CMOS I/O port. •I/O direction register allows each pin to be individually programmed as either input or output. •CMOS compatible input level. •P32 to P33 can be switched between CMOS compatible input level or SMBUS input level in the I2C-BUS interface function. •P30, P31, P34–P37 are CMOS 3-state output structure. •P32, P33 are N-channel open-drain output structure. •Pull-up control of P30, P31, P34–P37 is enabled in a bit unit.	•D/A converter input pin
P32/SDA P33/SCL			•I2C-BUS interface function pins
P34/RxD3 P35/TxD3 P36/SCLK3 P37/$\overline{\text{SRDY3}}$			•Serial I/O3 function pin
P40/INT40/XCOUT P41/INT00/XCIN	I/O port P4	•8-bit CMOS I/O port. •I/O direction register allows each pin to be individually programmed as either input or output. •CMOS compatible input level. •CMOS 3-state output structure. •Pull-up control is enabled in a bit unit.	•Interrupt input pin •Sub-clock generating I/O pin (resonator connected)
P42/INT1 P43/INT2			•Interrupt input pin
P44/RxD1 P45/TxD1 P46/SCLK1			•Serial I/O1 function pin
P47/$\overline{\text{SRDY1}}$ /CNTR2			•Serial I/O1, timer Z function pin
P50/SIN2 P51/SOUT2 P52/SCLK2 P53/$\overline{\text{SRDY2}}$	I/O port P5	•8-bit CMOS I/O port. •I/O direction register allows each pin to be individually programmed as either input or output. •CMOS compatible input level. •CMOS 3-state output structure. •Pull-up control is enabled in a bit unit.	•Serial I/O2 function pin
P54/CNTR0			•Timer X function pin
P55/CNTR1			•Timer Y function pin
P56/PWM			•PWM output pin
P57/INT3			•Interrupt input pin
P60/AN0–P67/AN7	I/O port P6		•A/D converter input pin

PART NUMBERING

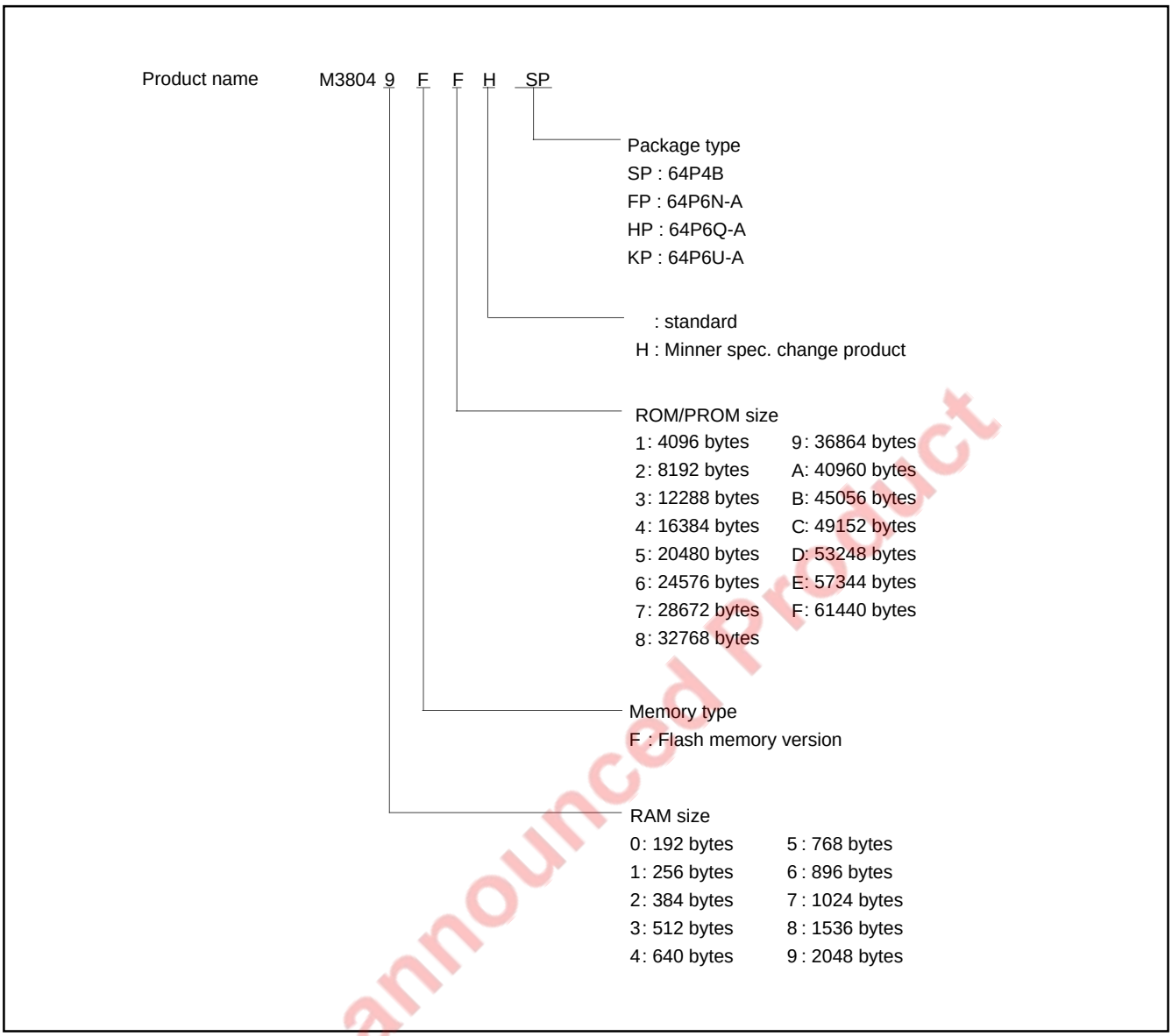


Fig. 4 Part numbering

GROUP EXPANSION

Renesas plans to expand the 3804 group (Spec. H) as follows.

Memory Size

Flash memory size ..... 60 K bytes
RAM size ..... 2048 bytes

Packages

64P4B ..... 64-pin shrink plastic-molded DIP
64P6N-A ..... 0.8 mm-pitch plastic molded QFP
64P6Q-A ..... 0.5 mm-pitch plastic molded LQFP
64P6U-A ..... 0.8 mm-pitch plastic molded LQFP

Memory Expansion Plan

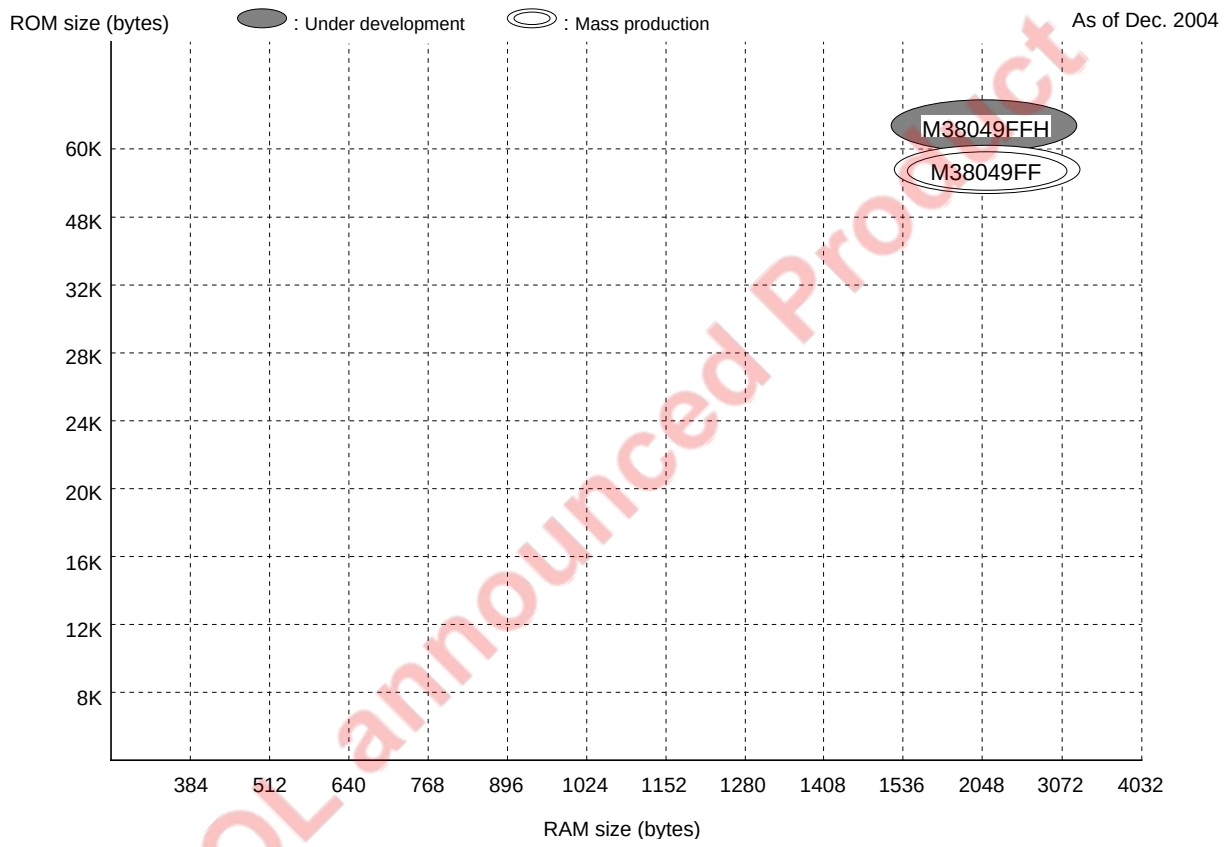


Fig. 5 Memory expansion plan

FUNCTIONAL DESCRIPTION
CENTRAL PROCESSING UNIT (CPU)

The 3804 group (Spec. H) uses the standard 740 Family instruction set. Refer to the table of 740 Family addressing modes and machine instructions or the 740 Family Software Manual for details on the instruction set.

Machine-resident 740 Family instructions are as follows:

The FST and SLW instructions cannot be used.

The STP, WIT, MUL, and DIV instructions can be used.

[Accumulator (A)]

The accumulator is an 8-bit register. Data operations such as data transfer, etc. are executed mainly through the accumulator.

[Index Register X (X)]

The index register X is an 8-bit register. In the index addressing modes, the value of the OPERAND is added to the contents of register X and specifies the real address.

[Index Register Y (Y)]

The index register Y is an 8-bit register. In partial instruction, the value of the OPERAND is added to the contents of register Y and specifies the real address.

[Stack Pointer (S)]

The stack pointer is an 8-bit register used during subroutine calls and interrupts. This register indicates start address of stored area (stack) for storing registers during subroutine calls and interrupts.

The low-order 8 bits of the stack address are determined by the contents of the stack pointer. The high-order 8 bits of the stack address are determined by the stack page selection bit. If the stack page selection bit is “0” , the high-order 8 bits becomes “00_{16}”. If the stack page selection bit is “1”, the high-order 8 bits becomes “01_{16}”.

The operations of pushing register contents onto the stack and popping them from the stack are shown in Figure 7.

Store registers other than those described in Figure 6 with program when the user needs them during interrupts or subroutine calls (see Table 3).

[Program Counter (PC)]

The program counter is a 16-bit counter consisting of two 8-bit registers PC_H and PC_L. It is used to indicate the address of the next instruction to be executed.

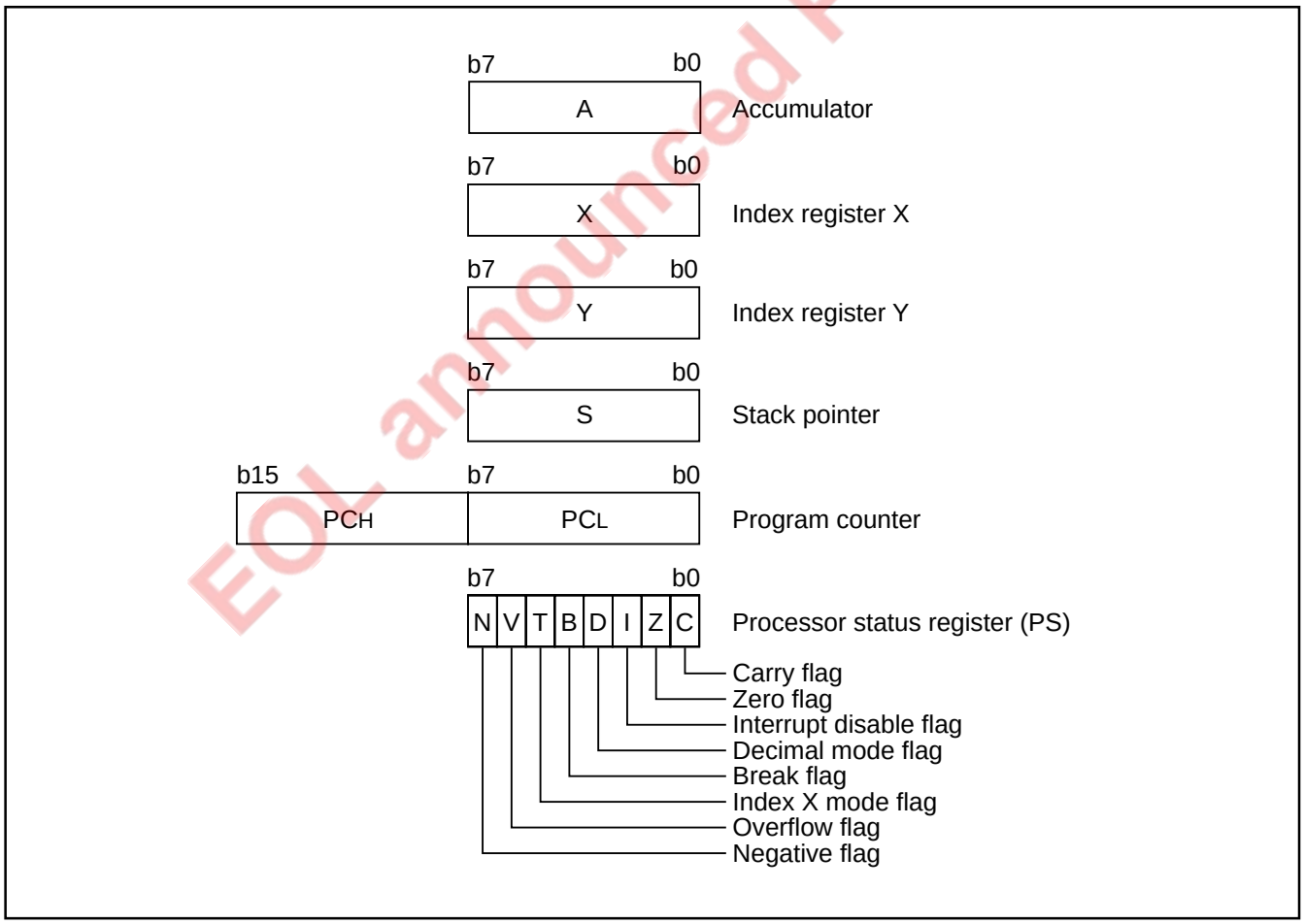


Fig. 6 740 Family CPU register structure

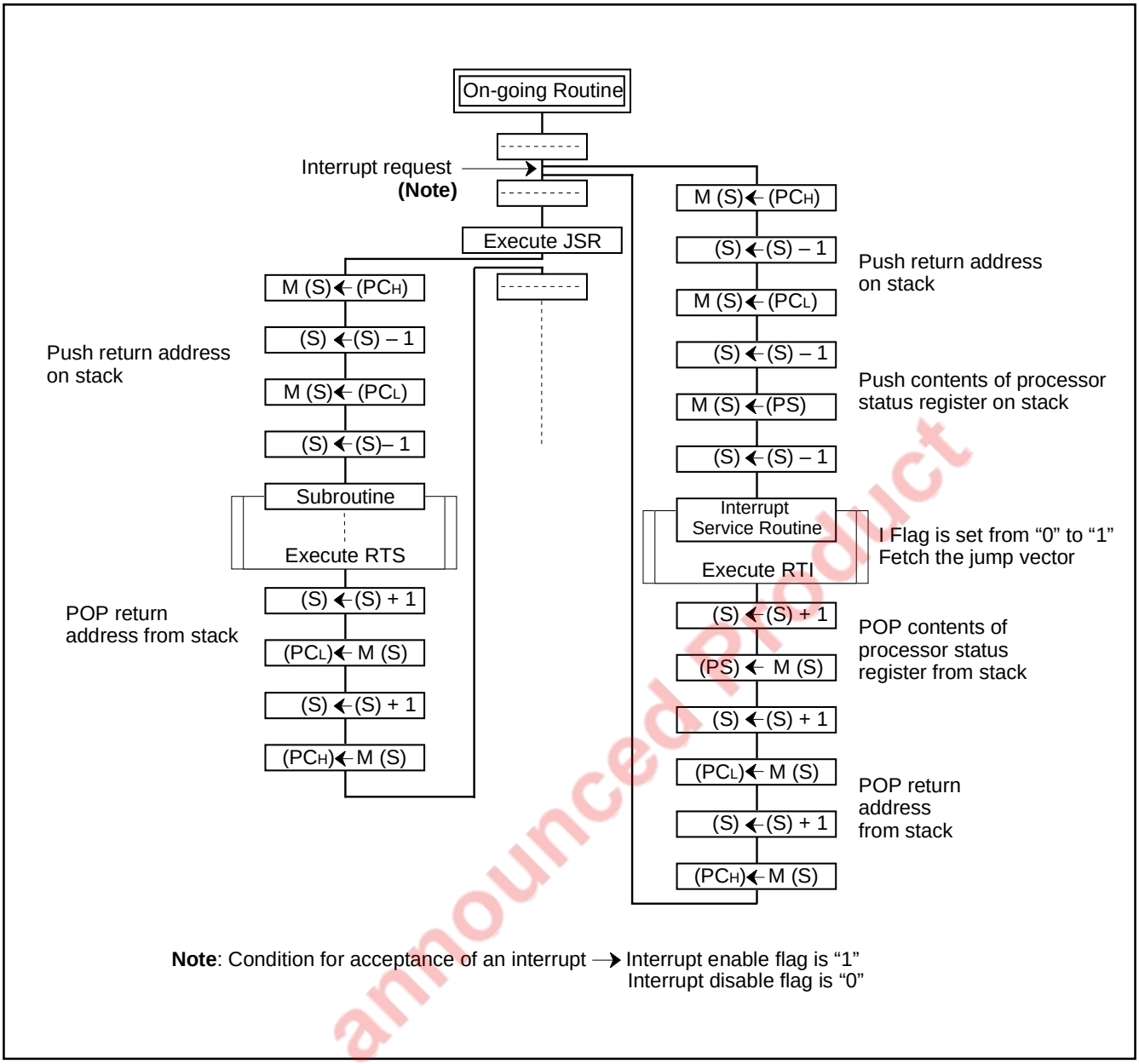


Fig. 7 Register push and pop at interrupt generation and subroutine call

Table 3 Push and pop instructions of accumulator or processor status register

	Push instruction to stack	Pop instruction from stack
Accumulator	PHA	PLA
Processor status register	PHP	PLP

[Processor status register (PS)]

The processor status register is an 8-bit register consisting of 5 flags which indicate the status of the processor after an arithmetic operation and 3 flags which decide MCU operation. Branch operations can be performed by testing the Carry (C) flag , Zero (Z) flag, Overflow (V) flag, or the Negative (N) flag. In decimal mode, the Z, V, N flags are not valid.

•Bit 0: Carry flag (C)
The C flag contains a carry or borrow generated by the arithmetic logic unit (ALU) immediately after an arithmetic operation. It can also be changed by a shift or rotate instruction.

•Bit 1: Zero flag (Z)
The Z flag is set if the result of an immediate arithmetic operation or a data transfer is “0”, and cleared if the result is anything other than “0”.

•Bit 2: Interrupt disable flag (I)
The I flag disables all interrupts except for the interrupt generated by the BRK instruction.
Interrupts are disabled when the I flag is “1”.

•Bit 3: Decimal mode flag (D)
The D flag determines whether additions and subtractions are executed in binary or decimal. Binary arithmetic is executed when this flag is “0”; decimal arithmetic is executed when it is “1”. Decimal correction is automatic in decimal mode. Only the ADC and SBC instructions can execute decimal arithmetic.

•Bit 4: Break flag (B)
The B flag is used to indicate that the current interrupt was generated by the BRK instruction. The BRK flag in the processor status register is always “0”. When the BRK instruction is used to generate an interrupt, the processor status register is pushed onto the stack with the break flag set to “1”.

•Bit 5: Index X mode flag (T)
When the T flag is “0”, arithmetic operations are performed between accumulator and memory. When the T flag is “1”, direct arithmetic operations and direct data transfers are enabled between memory locations.

•Bit 6: Overflow flag (V)
The V flag is used during the addition or subtraction of one byte of signed data. It is set if the result exceeds +127 to -128. When the BIT instruction is executed, bit 6 of the memory location operated on by the BIT instruction is stored in the overflow flag.

•Bit 7: Negative flag (N)
The N flag is set if the result of an arithmetic operation or data transfer is negative. When the BIT instruction is executed, bit 7 of the memory location operated on by the BIT instruction is stored in the negative flag.

Table 4 Set and clear instructions of each bit of processor status register

	C flag	Z flag	I flag	D flag	B flag	T flag	V flag	N flag
Set instruction	SEC	–	SEI	SED	–	SET	–	–
Clear instruction	CLC	–	CLI	CLD	–	CLT	CLV	–

[CPU Mode Register (CPUM)] 003B16

The CPU mode register contains the stack page selection bit, etc.
The CPU mode register is allocated at address 003B16.

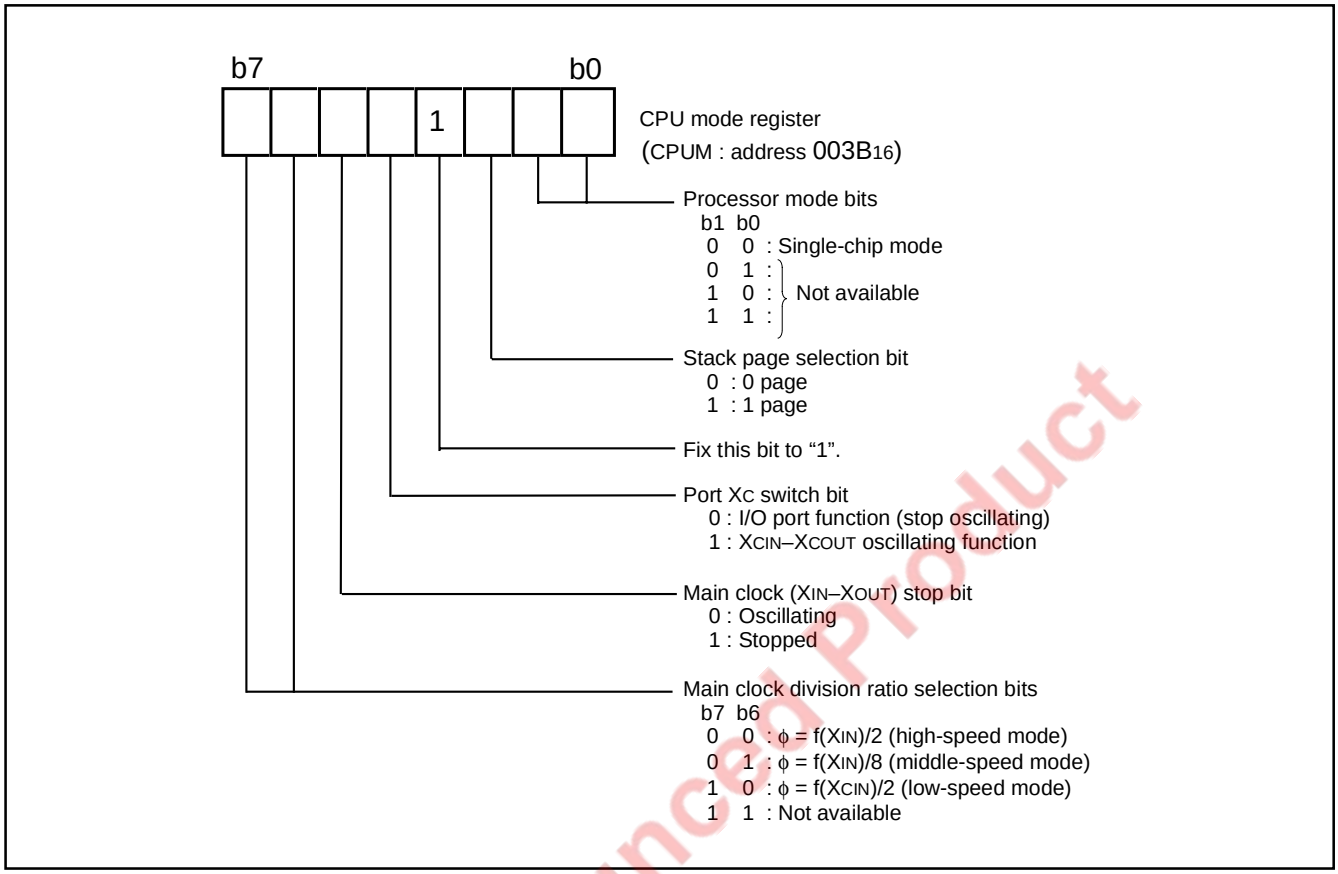


Fig. 8 Structure of CPU mode register

MISRG

(1) Bit 0 of address 0010$_{16}$: Oscillation stabilizing time set after STP instruction released bit

When the MCU stops the clock oscillation by the STP instruction and the STP instruction has been released by an external interrupt source, usually, the fixed values of Timer 1 and Prescaler 12 (Timer 1 = 01$_{16}$, Prescaler 12 = FF$_{16}$) are automatically reloaded in order for the oscillation to stabilize. The user can inhibit the automatic setting by setting "1" to bit 0 of MISRG (address 0010$_{16}$). However, by setting this bit to "1", the previous values, set just before the STP instruction was executed, will remain in Timer 1 and Prescaler 12. Therefore, you will need to set an appropriate value to each register, in accordance with the oscillation stabilizing time, before executing the STP instruction.
Figure 9 shows the structure of MISRG.

(2) Bits 1, 2, 3 of address 0010$_{16}$: Middle-speed Mode Automatic Switch Function

In order to switch the clock mode of an MCU which has a sub-clock, the following procedure is necessary:
set CPU mode register (003B$_{16}$) --> start main clock oscillation --> wait for oscillation stabilization --> switch to middle-speed mode (or high-speed mode).
However, the 3804 group (Spec. H) has the built-in function which automatically switches from low to middle-speed mode either by the SCL/SDA interrupt or by program.

●Middle-speed mode automatic switch by SCL/SDA Interrupt

The SCL/SDA interrupt source enables an automatic switch when the middle-speed mode automatic switch set bit (bit 1) of MISRG (address 0010$_{16}$) is set to "1". The conditions for an automatic switch execution depend on the settings of bits 5 and 6 of the I^2C START/STOP condition control register (address 0016$_{16}$). Bit 5 is the SCL/SDA interrupt pin polarity selection bit and bit 6 is the SCL/SDA interrupt pin selection bit. The main clock oscillation stabilizing time can also be selected by middle-speed mode automatic switch wait time set bit (bit 2) of the MISRG.

●Middle-speed mode automatic switch by program

The middle-speed mode can also be automatically switched by program while operating in low-speed mode. By setting the middle-speed automatic switch start bit (bit 3) of MISRG (address 0010$_{16}$) to "1" in the condition that the middle-speed mode automatic switch set bit is "1" while operating in low-speed mode, the MCU will automatically switch to middle-speed mode. In this case, the oscillation stabilizing time of the main clock can be selected by the middle-speed automatic switch wait time set bit (bit 2) of MISRG (address 0010$_{16}$).

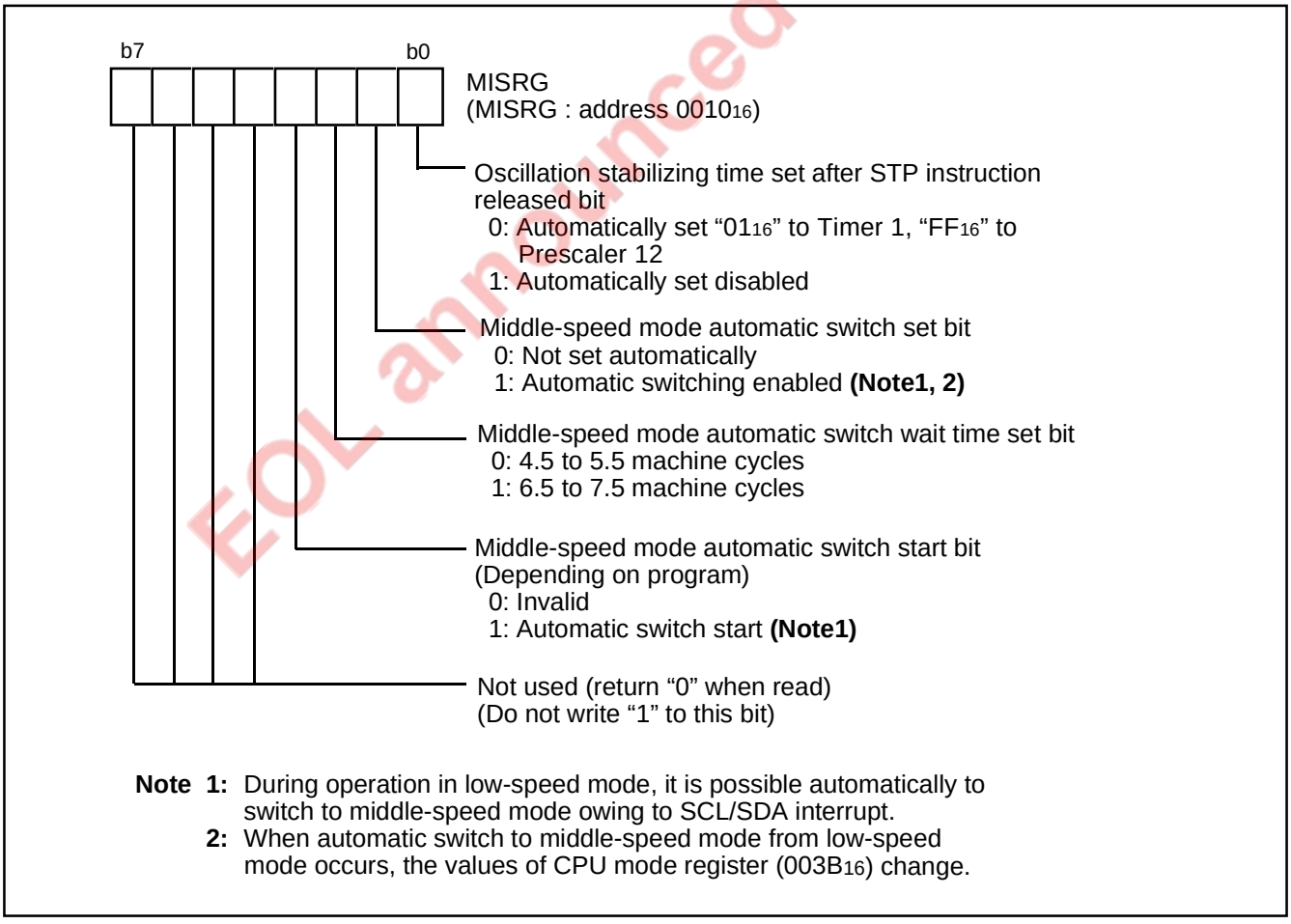


Fig. 9 Structure of MISRG

MEMORY

Special Function Register (SFR) Area

The Special Function Register area in the zero page contains control registers such as I/O ports and timers.

RAM

The RAM is used for data storage and for stack area of subroutine calls and interrupts.

ROM

The ROM area can program/erase.

Interrupt Vector Area

The interrupt vector area contains reset and interrupt vectors.

Zero Page

Access to this area with only 2 bytes is possible in the zero page addressing mode.

Special Page

Access to this area with only 2 bytes is possible in the special page addressing mode.

RAM area

RAM size (bytes)	Address XXXX16
192	00FF16
256	013F16
384	01BF16
512	023F16
640	02BF16
768	033F16
896	03BF16
1024	043F16
1536	063F16
2048	083F16

ROM area

ROM size (bytes)	Address YYYY16
4096	F00016
8192	E00016
12288	D00016
16384	C00016
20480	B00016
24576	A00016
28672	900016
32768	800016
36864	700016
40960	600016
45056	500016
49152	400016
53248	300016
57344	200016
61440	100016

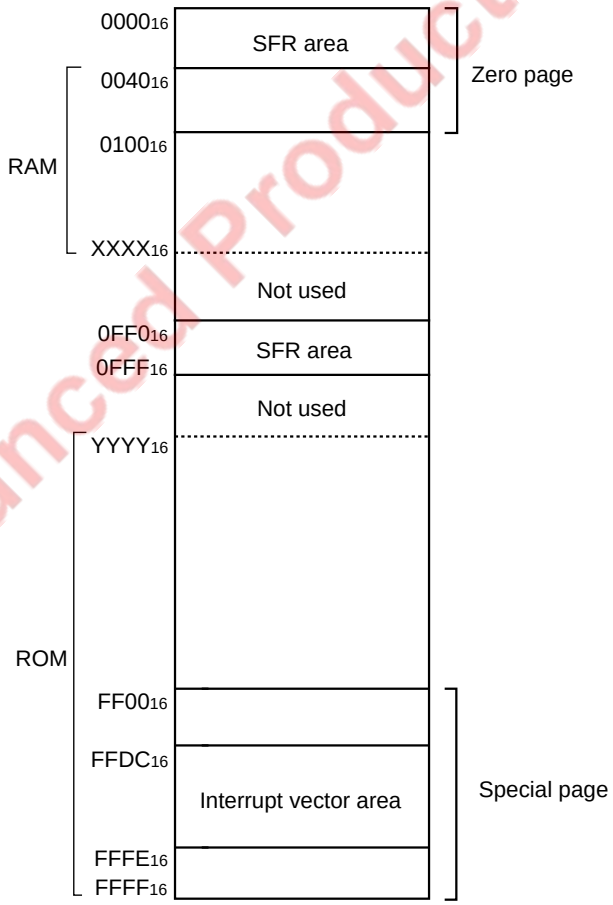


Fig. 10 Memory map diagram

0000₁₆	Port P0 (P0)
0001₁₆	Port P0 direction register (P0D)
0002₁₆	Port P1 (P1)
0003₁₆	Port P1 direction register (P1D)
0004₁₆	Port P2 (P2)
0005₁₆	Port P2 direction register (P2D)
0006₁₆	Port P3 (P3)
0007₁₆	Port P3 direction register (P3D)
0008₁₆	Port P4 (P4)
0009₁₆	Port P4 direction register (P4D)
000A₁₆	Port P5 (P5)
000B₁₆	Port P5 direction register (P5D)
000C₁₆	Port P6 (P6)
000D₁₆	Port P6 direction register (P6D)
000E₁₆	Timer 12, X count source selection register (T12XCSS)
000F₁₆	Timer Y, Z count source selection register (TYZCSS)
0010₁₆	MISRG
0011₁₆	I^2C data shift register (S0)
0012₁₆	I^2C special mode status register (S3)
0013₁₆	I^2C status register (S1)
0014₁₆	I^2C control register (S1D)
0015₁₆	I^2C clock control register (S2)
0016₁₆	I^2C START/STOP condition control register (S2D)
0017₁₆	I^2C special mode control register (S3D)
0018₁₆	Transmit/Receive buffer register 1 (TB1/RB1)
0019₁₆	Serial I/O1 status register (SIO1STS)
001A₁₆	Serial I/O1 control register (SIO1CON)
001B₁₆	UART1 control register (UART1CON)
001C₁₆	Baud rate generator 1 (BRG1)
001D₁₆	Serial I/O2 control register (SIO2CON)
001E₁₆	Watchdog timer control register (WDTCON)
001F₁₆	Serial I/O2 register (SIO2)

0020₁₆	Prescaler 12 (PRE12)
0021₁₆	Timer 1 (T1)
0022₁₆	Timer 2 (T2)
0023₁₆	Timer XY mode register (TM)
0024₁₆	Prescaler X (PREX)
0025₁₆	Timer X (TX)
0026₁₆	Prescaler Y (PREY)
0027₁₆	Timer Y (TY)
0028₁₆	Timer Z low-order (TZL)
0029₁₆	Timer Z high-order (TZH)
002A₁₆	Timer Z mode register (TZM)
002B₁₆	PWM control register (PWMCON)
002C₁₆	PWM prescaler (PREPWM)
002D₁₆	PWM register (PWM)
002E₁₆	
002F₁₆	Baud rate generator 3 (BRG3)
0030₁₆	Transmit/Receive buffer register 3 (TB3/RB3)
0031₁₆	Serial I/O3 status register (SIO3STS)
0032₁₆	Serial I/O3 control register (SIO3CON)
0033₁₆	UART3 control register (UART3CON)
0034₁₆	AD/DA control register (ADCON)
0035₁₆	AD conversion register 1 (AD1)
0036₁₆	DA1 conversion register (DA1)
0037₁₆	DA2 conversion register (DA2)
0038₁₆	AD conversion register 2 (AD2)
0039₁₆	Interrupt source selection register (INTSEL)
003A₁₆	Interrupt edge selection register (INTEDGE)
003B₁₆	CPU mode register (CPUM)
003C₁₆	Interrupt request register 1 (IREQ1)
003D₁₆	Interrupt request register 2 (IREQ2)
003E₁₆	Interrupt control register 1 (ICON1)
003F₁₆	Interrupt control register 2 (ICON2)

0FE0₁₆	Flash memory control register 0 (FMCR0)
0FE1₁₆	Flash memory control register 1 (FMCR1)
0FE2₁₆	Flash memory control register 2 (FMCR2)
0FE3₁₆	Reserved ✻
0FE4₁₆	Reserved ✻
0FE5₁₆	Reserved ✻
0FE6₁₆	Reserved ✻
0FE7₁₆	Reserved ✻
0FE8₁₆	Reserved ✻
0FE9₁₆	Reserved ✻
0FEA₁₆	Reserved ✻
0FEB₁₆	Reserved ✻
0FEC₁₆	Reserved ✻
0FED₁₆	Reserved ✻
0FEE₁₆	Reserved ✻
0FEF₁₆	Reserved ✻

0FF0₁₆	Port P0 pull-up control register (PULL0)
0FF1₁₆	Port P1 pull-up control register (PULL1)
0FF2₁₆	Port P2 pull-up control register (PULL2)
0FF3₁₆	Port P3 pull-up control register (PULL3)
0FF4₁₆	Port P4 pull-up control register (PULL4)
0FF5₁₆	Port P5 pull-up control register (PULL5)
0FF6₁₆	Port P6 pull-up control register (PULL6)
0FF7₁₆	I^2C slave address register 0 (S0D0)
0FF8₁₆	I^2C slave address register 1 (S0D1)
0FF9₁₆	I^2C slave address register 2 (S0D2)

✻ Reserved area: Do not write any data to these addresses, because these areas are reserved.

Fig. 11 Memory map of special function register (SFR)

I/O PORTS

The I/O ports have direction registers which determine the input/output direction of each individual pin. Each bit in a direction register corresponds to one pin, and each pin can be set to be input port or output port.

When "0" is written to the bit corresponding to a pin, that pin becomes an input pin. When "1" is written to that bit, that pin becomes an output pin.

If data is read from a pin which is set to output, the value of the port output latch is read, not the value of the pin itself. Pins set to input are floating. If a pin set to input is written to, only the port output latch is written to and the pin remains floating.

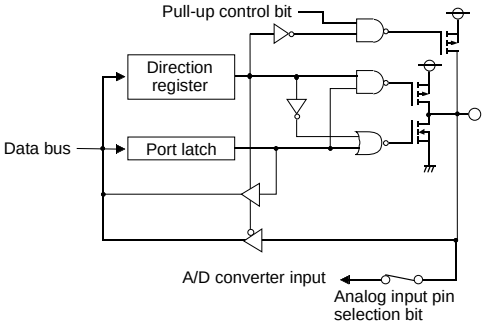
Table 5 I/O port function

Pin	Name	I/O Structure	Non-Port Function	Related SFRs	Ref.No.
P0$_0$/AN$_8$–P0$_7$/AN$_{15}$	Port P0	CMOS compatible input level CMOS 3-state output	A/D converter input	AD/DA control register	(1)
P1$_0$/INT$_{41}$ P1$_1$/INT$_{01}$	Port P1		External interrupt input	Interrupt edge selection register	(2)
P1$_2$–P1$_7$					(3)
P2$_0$/LED$_0$– P2$_7$/LED$_7$	Port P2				
P3$_0$/DA$_1$ P3$_1$/DA$_2$	Port P3	CMOS compatible input level CMOS 3-state output	D/A converter output	AD/DA control register	(4)
P3$_2$/SDA P3$_3$/SCL		CMOS compatible input level N-channel open-drain output CMOS/SMBUS input level (when selecting I^{2}C-BUS interface function)	I^{2}C-BUS interface function I/O	I^{2}C control register	(5)
P3$_4$/RxD$_3$ P3$_5$/TxD$_3$ P3$_6$/SCLK$_3$ P3$_7$/$\overline{\text{SRDY}_3}$		CMOS compatible input level CMOS 3-state output	Serial I/O3 function I/O	Serial I/O3 control register UART3 control register	(6) (7) (8) (9)
P4$_0$/INT$_{40}$/XCIN P4$_1$/INT$_{00}$/XCOUT	Port P4	CMOS compatible input level CMOS 3-state output	External interrupt input Sub-clock generating circuit	Interrupt edge selection register CPU mode register	(10) (11)
P4$_2$/INT$_1$ P4$_3$/INT$_2$			External interrupt input	Interrupt edge selection register	(2)
P4$_4$/RxD$_1$ P4$_5$/TxD$_1$ P4$_6$/SCLK$_1$			Serial I/O1 function I/O	Serial I/O1 control register UART1 control register	(6) (7) (8)
P4$_7$/$\overline{\text{SRDY}_1}$/CNTR$_2$			Serial I/O1 function I/O Timer Z function I/O	Serial I/O1 control register Timer Z mode register	(12)
P5$_0$/SIN$_2$ P5$_1$/SOUT$_2$ P5$_2$/SCLK$_2$ P5$_3$/$\overline{\text{SRDY}_2}$	Port P5	CMOS compatible input level CMOS 3-state output	Serial I/O2 function I/O	Serial I/O2 control register	(13) (14) (15) (16)
P5$_4$/CNTR$_0$ P5$_5$/CNTR$_1$			Timer X, Y function I/O	Timer XY mode register	(17)
P5$_6$/PWM			PWM output	PWM control register	(18)
P5$_7$/INT$_3$			External interrupt input	Interrupt edge selection register	(2)
P6$_0$/AN$_0$–P6$_7$/AN$_7$	Port P6	CMOS compatible input level CMOS 3-state output	A/D converter input	AD/DA control register	(1)

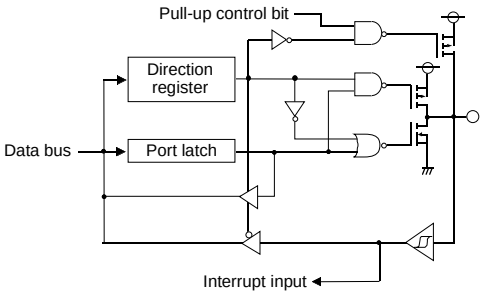
Notes 1: Refer to the applicable sections how to use double-function ports as function I/O ports.

2: Make sure that the input level at each pin is either 0 V or V_{CC} during execution of the STP instruction.
When an input level is at an intermediate potential, a current will flow from V_{CC} to V_{SS} through the input-stage gate.

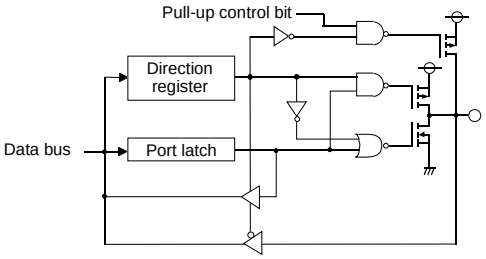
(1) Ports P0, P6



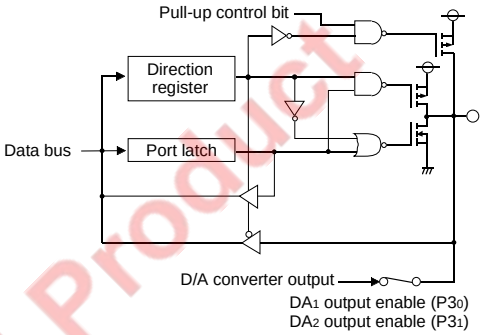
(2) Ports P10, P11, P42, P43, P57



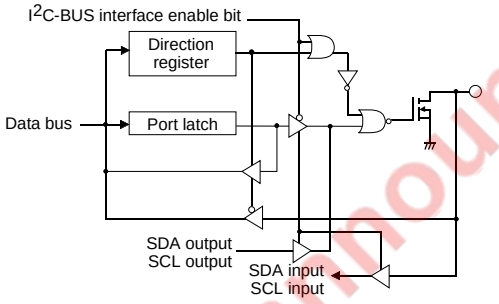
(3) Ports P12 to P17, P2



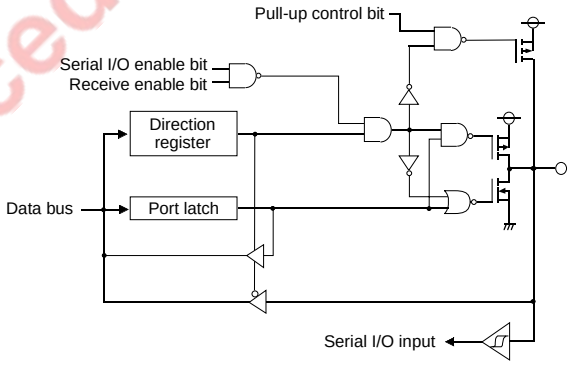
(4) Ports P30, P31



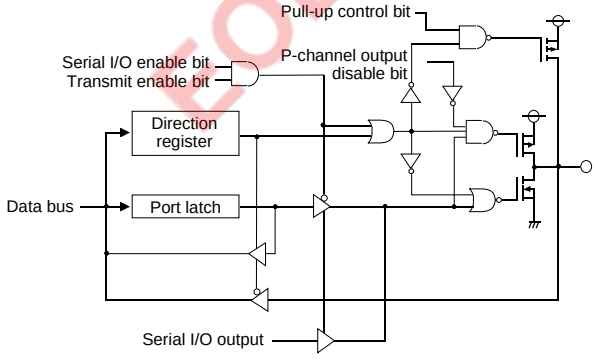
(5) Ports P32, P33



(6) Ports P34, P44



(7) Ports P35, P45



(8) Ports P36, P46

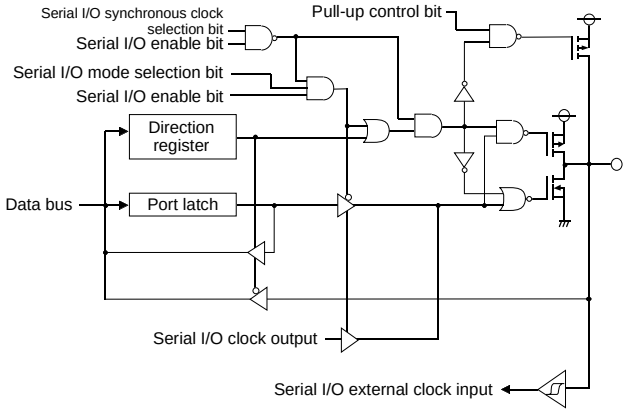
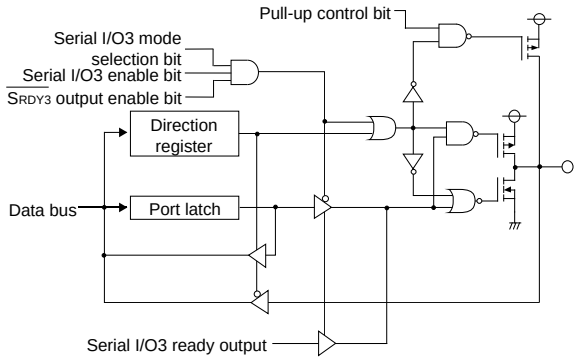
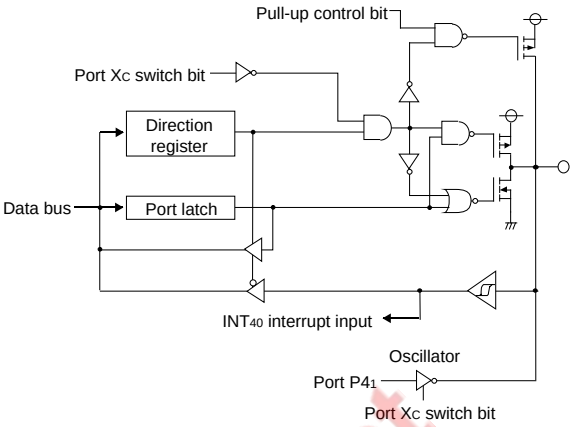


Fig. 12 Port block diagram (1)

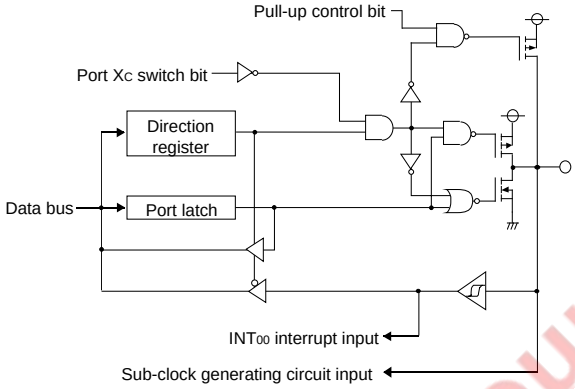
(9) Port P3$_7$



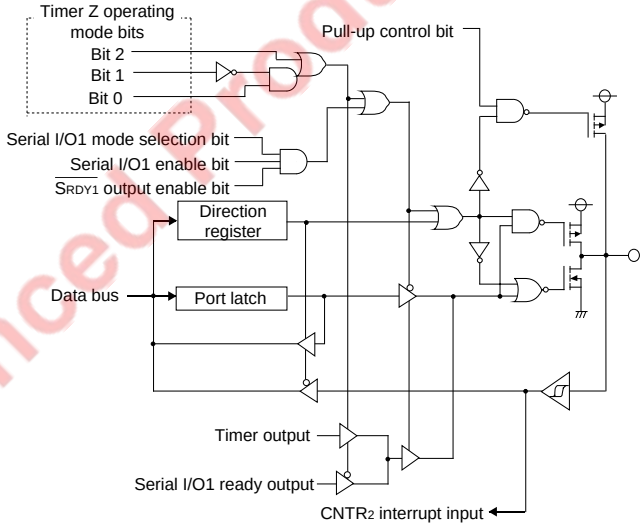
(10) Port P4$_0$



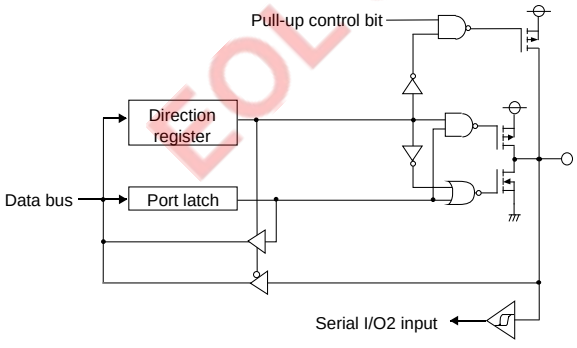
(11) Port P4$_1$



(12) Port P4$_7$



(13) Port P5$_0$



(14) Port P5$_1$

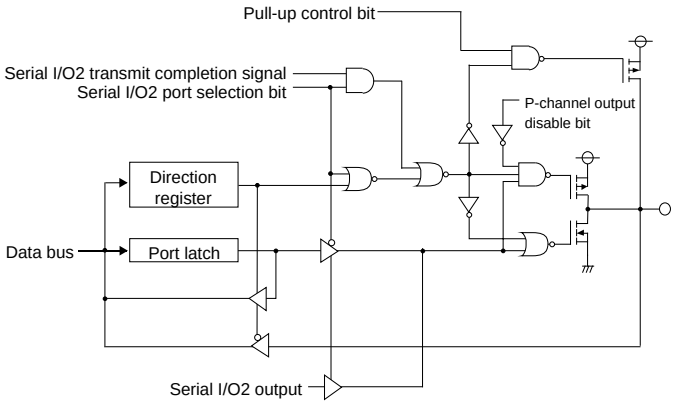
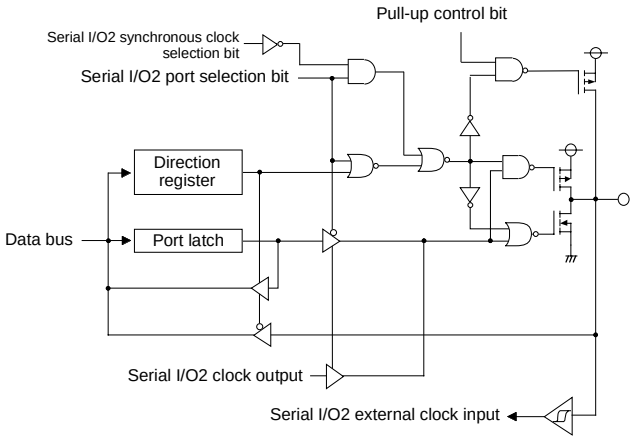
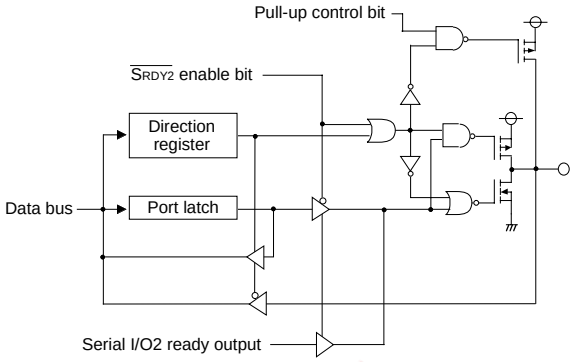


Fig. 13 Port block diagram (2)

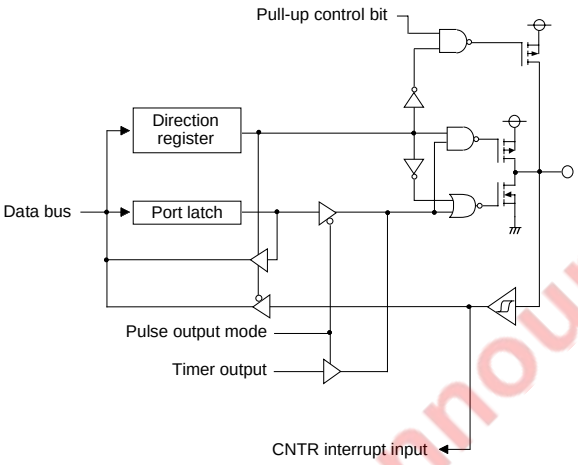
(15) Port P5$_2$



(16) Port P5$_3$



(17) Ports P5$_4$, P5$_5$



(18) Port P5$_6$

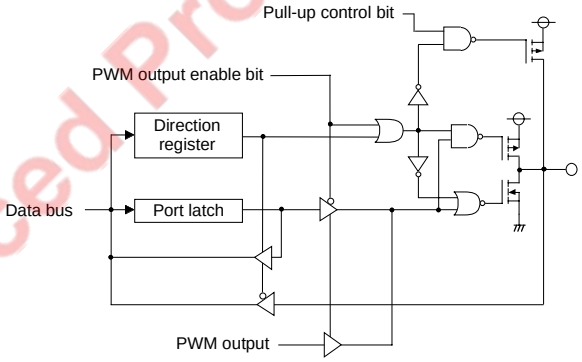


Fig. 14 Port block diagram (3)

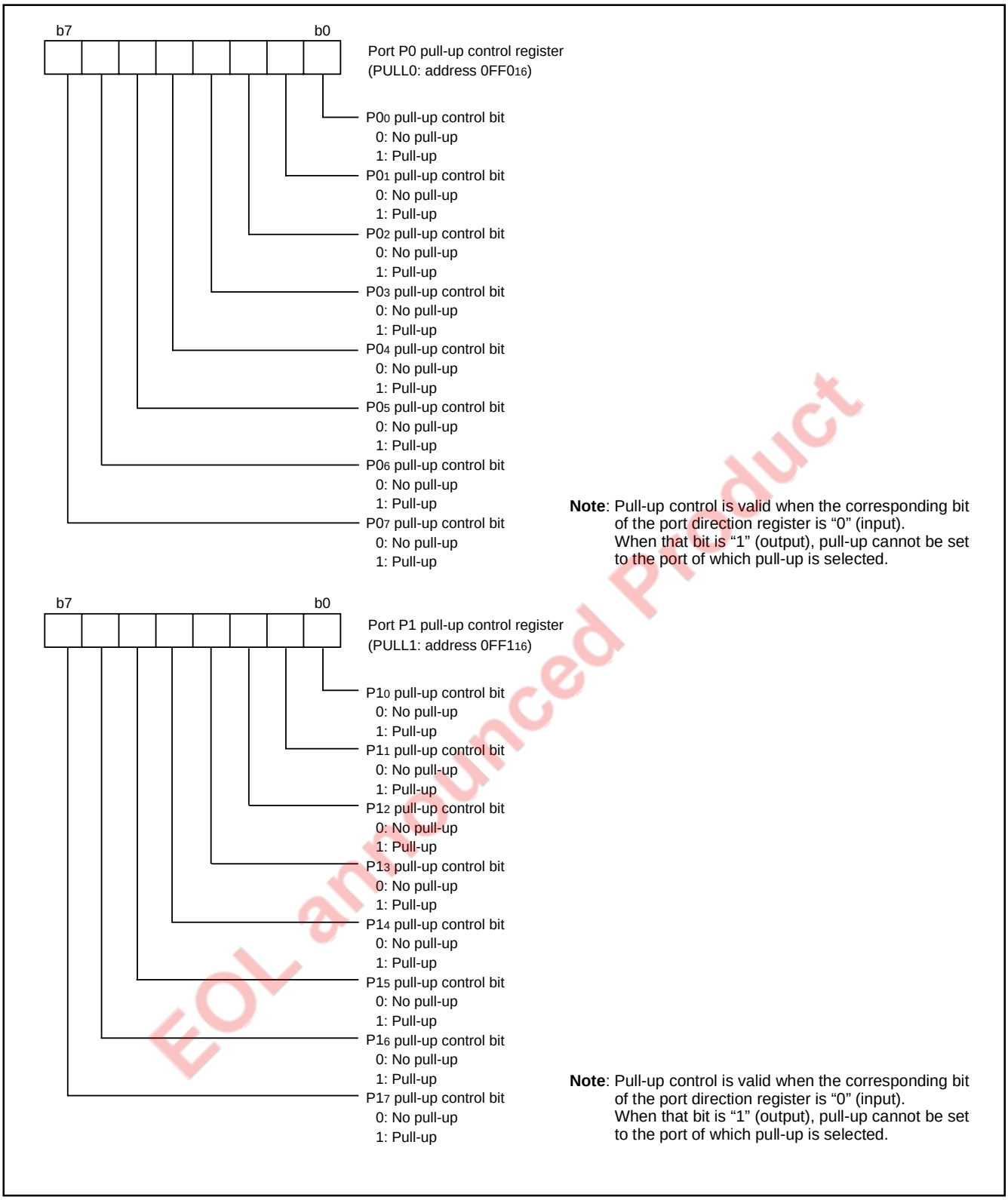
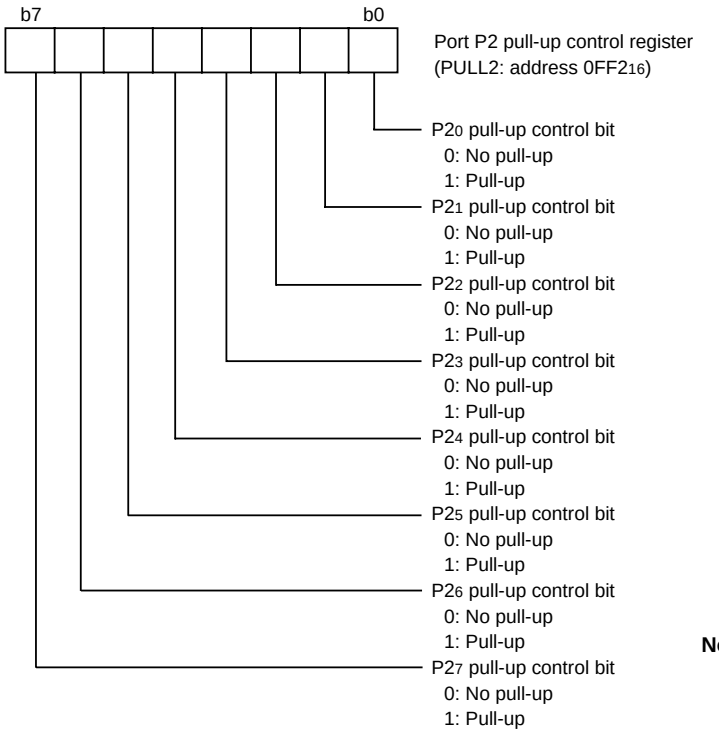
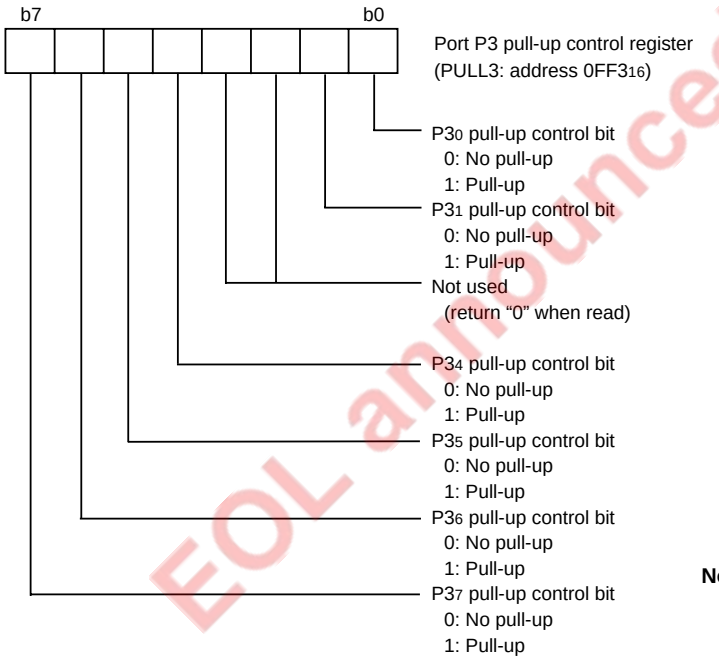


Fig. 15 Structure of port pull-up control register (1)



Note: Pull-up control is valid when the corresponding bit of the port direction register is "0" (input). When that bit is "1" (output), pull-up cannot be set to the port of which pull-up is selected.



Note: Pull-up control is valid when the corresponding bit of the port direction register is "0" (input). When that bit is "1" (output), pull-up cannot be set to the port of which pull-up is selected.

Fig. 16 Structure of port pull-up control register (2)

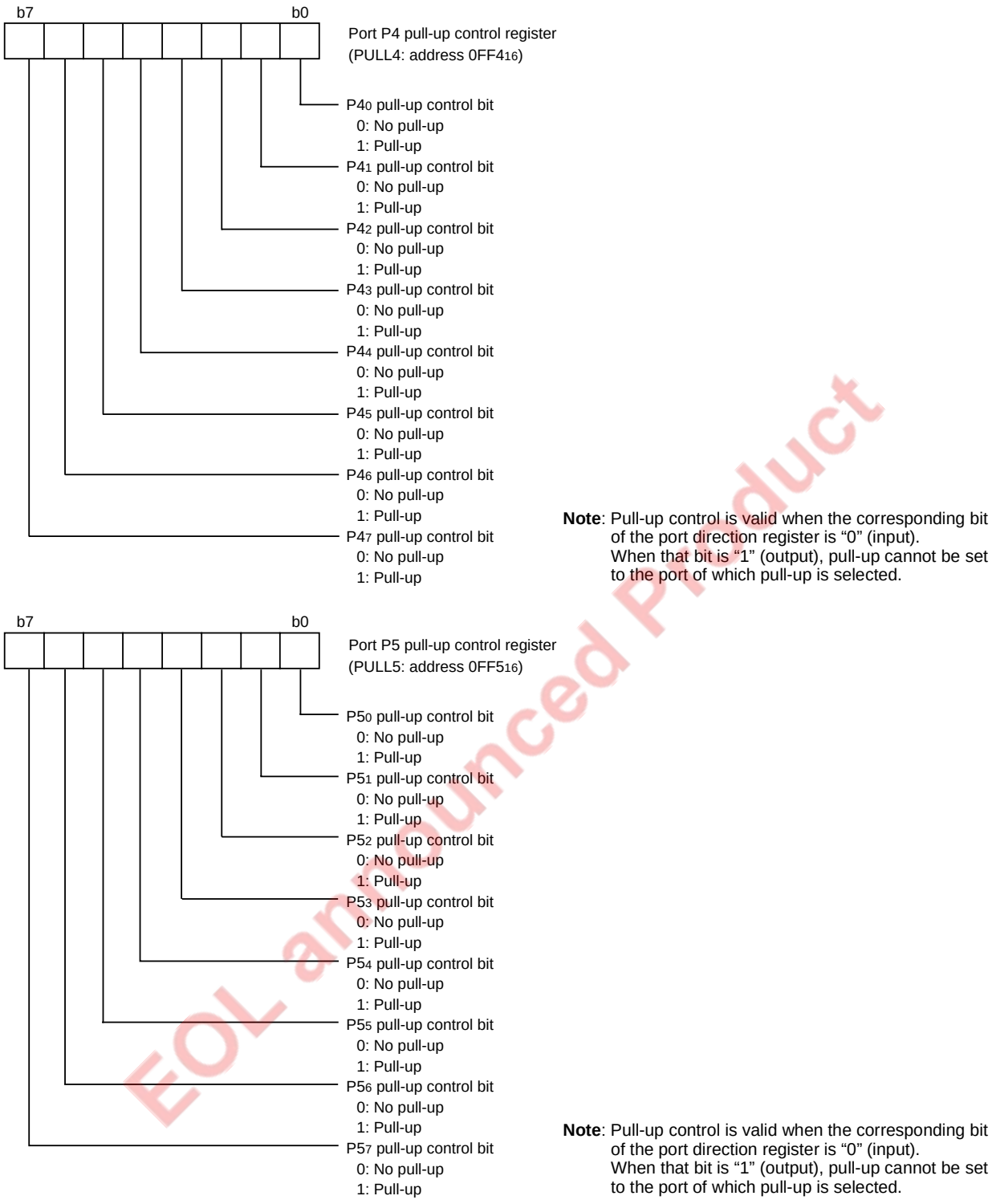


Fig. 17 Structure of port pull-up control register (3)

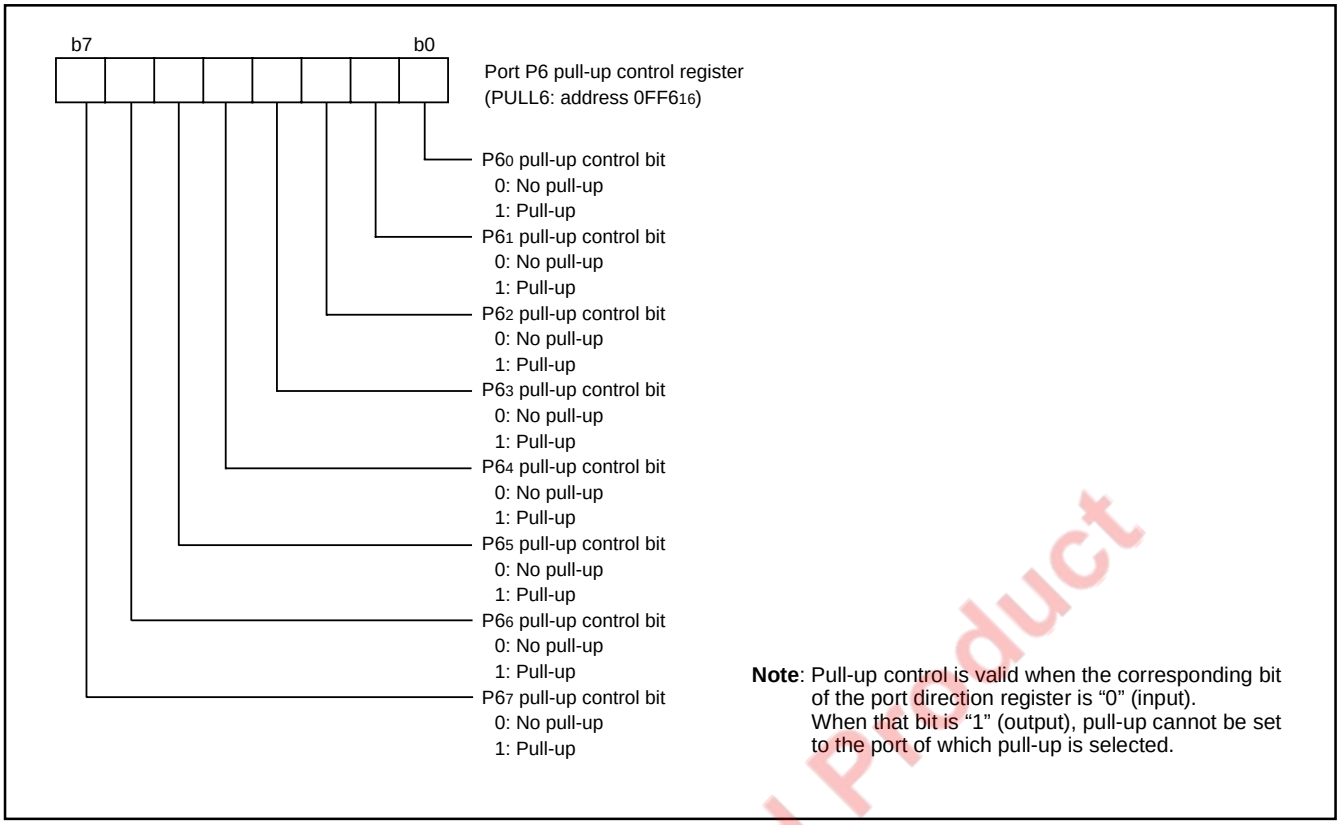


Fig. 18 Structure of port pull-up control register (4)

INTERRUPTS

The 3804 group (Spaec. H)'s interrupts are a type of vector and occur by 16 sources among 23 sources: nine external, thirteen internal, and one software.

Interrupt Control

Each interrupt is controlled by an interrupt request bit, an interrupt enable bit, and the interrupt disable flag except for the software interrupt set by the BRK instruction. An interrupt occurs if the corresponding interrupt request and enable bits are "1" and the interrupt disable flag is "0".

Interrupt enable bits can be set or cleared by software.

Interrupt request bits can be cleared by software, but cannot be set by software.

The reset and the BRK instruction cannot be disabled with any flag or bit. The I (interrupt disable) flag disables all interrupts except the reset and the BRK instruction interrupt.

When several interrupt requests occur at the same time, the interrupts are received according to priority.

Interrupt Operation

By acceptance of an interrupt, the following operations are automatically performed:

1. The contents of the program counter and the processor status register are automatically pushed onto the stack.
2. The interrupt disable flag is set and the corresponding interrupt request bit is cleared.
3. The interrupt jump destination address is read from the vector table into the program counter.

Interrupt Source Selection

Which of each combination of the following interrupt sources can be selected by the interrupt source selection register (address 0039_{16}).

1. INT_0 or Timer Z
2. Serial I/O1 transmission or SCL, SDA
3. $CNTR_0$ or SCL, SDA
4. $CNTR_1$ or Serial I/O3 reception
5. Serial I/O2 or Timer Z
6. INT_2 or I^2C
7. INT_4 or $CNTR_2$
8. A/D converter or serial I/O3 transmission

External Interrupt Pin Selection

The occurrence sources of the external interrupt INT_0 and INT_4 can be selected from either input from INT_{00} and INT_{40} pin, or input from INT_{01} and INT_{41} pin by the INT_0, INT_4 interrupt switch bit of interrupt edge selection register (bit 6 of address $003A_{16}$).

■ Notes

When setting the followings, the interrupt request bit may be set to "1".

•When setting external interrupt active edge
Related register: Interrupt edge selection register (address $003A_{16}$)
Timer XY mode register (address 0023_{16})
Timer Z mode register (address $002A_{16}$)
I^2C START/STOP condition control register (address 0016_{16})

•When switching interrupt sources of an interrupt vector address where two or more interrupt sources are allocated
Related register: Interrupt source selection register (address 0039_{16})

When not requiring for the interrupt occurrence synchronized with these setting, take the following sequence.

①Set the corresponding interrupt enable bit to "0" (disabled).
②Set the interrupt edge select bit or the interrupt source select bit to "1".
③Set the corresponding interrupt request bit to "0" after 1 or more instructions have been executed.
④Set the corresponding interrupt enable bit to "1" (enabled).

Table 6 Interrupt vector addresses and priority

Interrupt Source	Priority	Vector Addresses (Note 1)		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset (Note 2)	1	FFFD16	FFFC16	At reset	Non-maskable
INT0	2	FFFB16	FFFA16	At detection of either rising or falling edge of INT0 input	External interrupt (active edge selectable)
Timer Z				At timer Z underflow	
INT1	3	FFF916	FFF816	At detection of either rising or falling edge of INT1 input	External interrupt (active edge selectable)
Serial I/O1 reception	4	FFF716	FFF616	At completion of serial I/O1 data reception	Valid when serial I/O1 is selected
Serial I/O1 transmission	5	FFF516	FFF416	At completion of serial I/O1 transmission shift or when transmission buffer is empty	Valid when serial I/O1 is selected
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
Timer X	6	FFF316	FFF216	At timer X underflow	
Timer Y	7	FFF116	FFF016	At timer Y underflow	
Timer 1	8	FFEF16	FFEE16	At timer 1 underflow	STP release timer underflow
Timer 2	9	FFED16	FFEC16	At timer 2 underflow	
CNTR0	10	FFEB16	FFEA16	At detection of either rising or falling edge of CNTR0 input	External interrupt (active edge selectable)
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
CNTR1	11	FFE916	FFE816	At detection of either rising or falling edge of CNTR1 input	External interrupt (active edge selectable)
Serial I/O3 reception				At completion of serial I/O3 data reception	Valid when serial I/O3 is selected
Serial I/O2	12	FFE716	FFE616	At completion of serial I/O2 data transmission or reception	Valid when serial I/O2 is selected
Timer Z				At timer Z underflow	
INT2	13	FFE516	FFE416	At detection of either rising or falling edge of INT2 input	External interrupt (active edge selectable)
I^2C				At completion of data transfer	
INT3	14	FFE316	FFE216	At detection of either rising or falling edge of INT3 input	External interrupt (active edge selectable)
INT4	15	FFE116	FFE016	At detection of either rising or falling edge of INT4 input	External interrupt (active edge selectable)
CNTR2				At detection of either rising or falling edge of CNTR2 input	External interrupt (active edge selectable)
A/D converter	16	FFDF16	FFDE16	At completion of A/D conversion	
Serial I/O3 transmission				At completion of serial I/O3 transmission shift or when transmission buffer is empty	Valid when serial I/O3 is selected
BRK instruction	17	FFDD16	FFDC16	At BRK instruction execution	Non-maskable software interrupt

Notes 1: Vector addresses contain interrupt jump destination addresses.
2: Reset function in the same way as an interrupt with the highest priority.

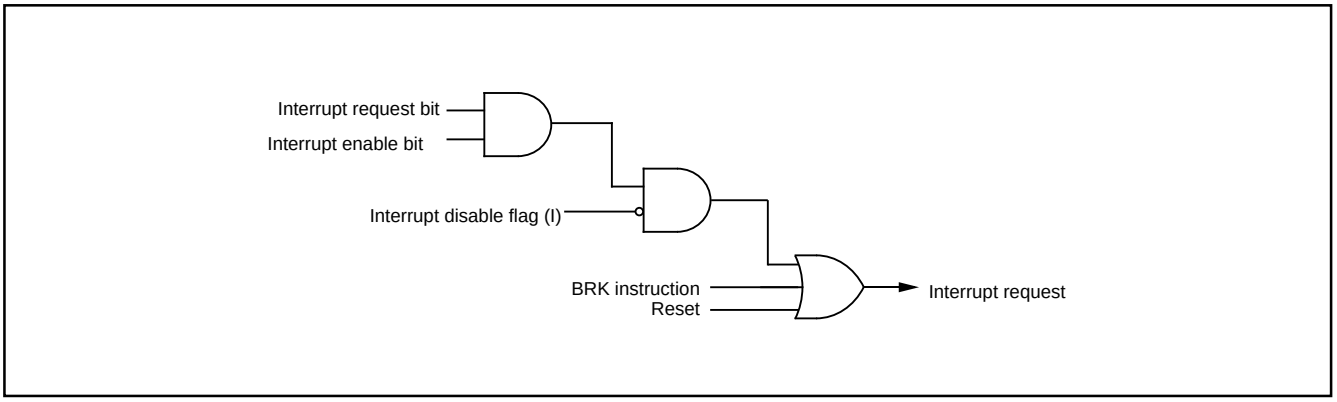


Fig. 19 Interrupt control

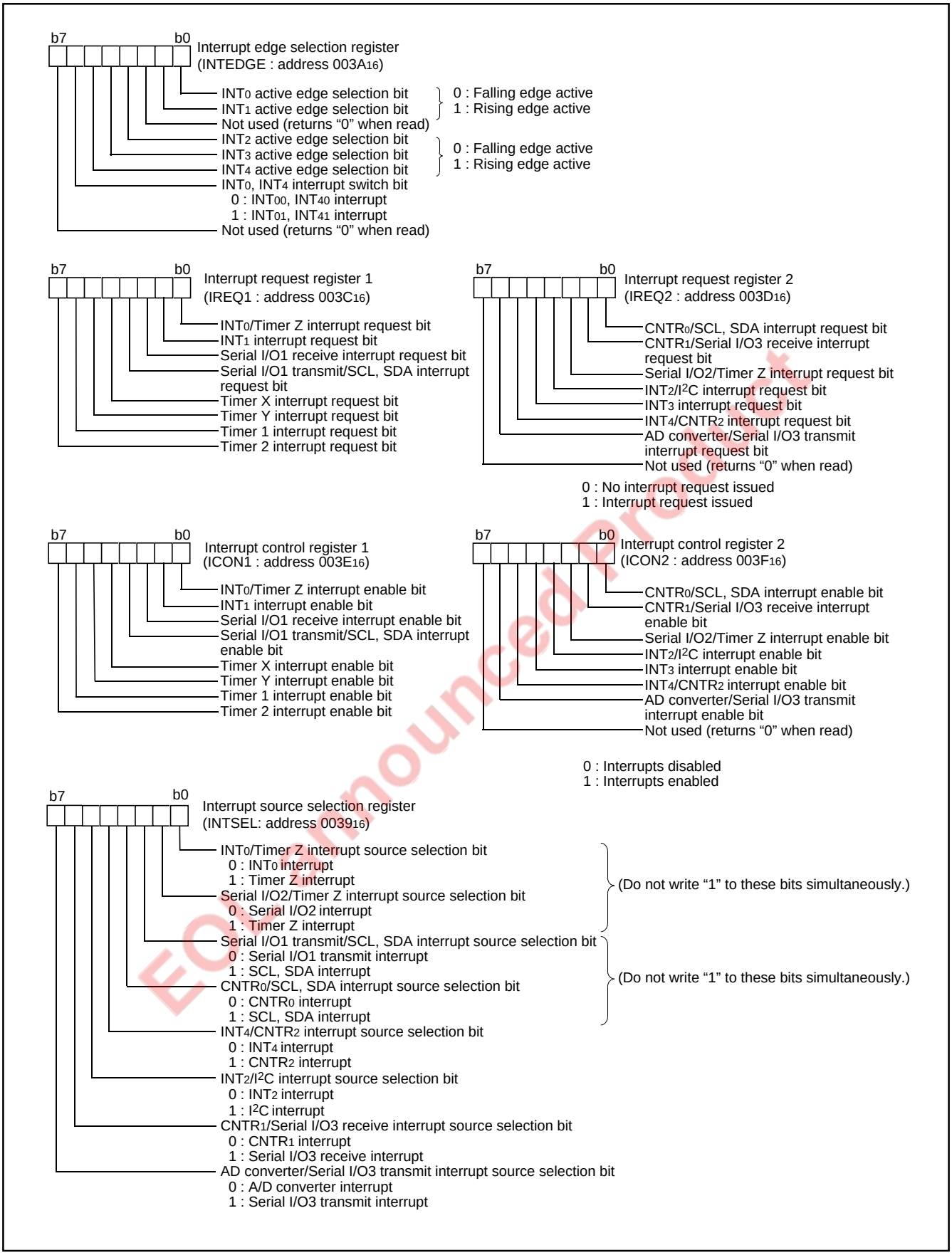


Fig. 20 Structure of interrupt-related registers

TIMERS

●8-bit Timers

The 3804 group (Spec. H) has four 8-bit timers: timer 1, timer 2, timer X, and timer Y.

The timer 1 and timer 2 use one prescaler in common, and the timer X and timer Y use each prescaler. Those are 8-bit prescalers. Each of the timers and prescalers has a timer latch or a prescaler latch.

The division ratio of each timer or prescaler is given by 1/(n + 1), where n is the value in the corresponding timer or prescaler latch.

All timers are down-counters. When the timer reaches "00_{16}", an underflow occurs at the next count pulse and the contents of the corresponding timer latch are reloaded into the timer and the count is continued. When the timer underflows, the interrupt request bit corresponding to that timer is set to "1".

●Timer divider

The divider count source is switched by the main clock division ratio selection bits of CPU mode register (bits 7 and 6 at address $003B_{16}$). When these bits are "00" (high-speed mode) or "01" (middle-speed mode), X_{IN} is selected. When these bits are"10" (low-speed mode), X_{CIN} is selected.

●Prescaler 12

The prescaler 12 counts the output of the timer divider. The count source is selected by the timer 12, X count source selection register among 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 of f(X_{IN}) or f(X_{CIN}).

Timer 1 and Timer 2

The timer 1 and timer 2 counts the output of prescaler 12 and periodically set the interrupt request bit.

●Prescaler X and prescaler Y

The prescaler X and prescaler Y count the output of the timer divider or f(X_{CIN}). The count source is selected by the timer 12, X count source selection register (address $000E_{16}$) and the timer Y, Z count source selection register (address $000F_{16}$) among 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, and 1/1024 of f(X_{IN}) or f(X_{CIN}); and f(X_{CIN}).

Timer X and Timer Y

The timer X and timer Y can each select one of four operating modes by setting the timer XY mode register (address 0023_{16}).

(1) Timer mode

●Mode selection

This mode can be selected by setting "00" to the timer X operating mode bits (bits 1 and 0) and the timer Y operating mode bits (bits 5 and 4) of the timer XY mode register (address 0023_{16}).

●Explanation of operation

The timer count operation is started by setting "0" to the timer X count stop bit (bit 3) and the timer Y count stop bit (bit 7) of the timer XY mode register (address 0023_{16}).

When the timer reaches "00_{16}", an underflow occurs at the next count pulse and the contents of timer latch are reloaded into the timer and the count is continued.

(2) Pulse output mode

●Mode selection

This mode can be selected by setting "01" to the timer X operating mode bits (bits 1 and 0) and the timer Y operating mode bits (bits 5 and 4) of the timer XY mode register (address 0023_{16}).

●Explanation of operation

The operation is the same as the timer mode's. Moreover the pulse which is inverted each time the timer underflows is output from $CNTR_0$/$CNTR_1$ pin. Regardless of the timer counting or not the output of $CNTR_0$/$CNTR_1$ pin is initialized to the level of specified by their active edge switch bits when writing to the timer. When the $CNTR_0$ active edge switch bit (bit 2) and the $CNTR_1$ active edge switch bit (bit 6) of the timer XY mode register (address 0023_{16}) is "0", the output starts with "H" level. When it is "1", the output starts with "L" level.

Switching the $CNTR_0$ or $CNTR_1$ active edge switch bit will reverse the output level of the corresponding $CNTR_0$ or CNTR1 pin.

■Precautions

Set the double-function port of $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to output in this mode.

(3) Event counter mode

●Mode selection

This mode can be selected by setting “10” to the timer X operating mode bits (bits 1 and 0) and the timer Y operating mode bits (bits 5 and 4) of the timer XY mode register (address 0023_{16}).

●Explanation of operation

The operation is the same as the timer mode’s except that the timer counts signals input from the $CNTR_0$ or $CNTR_1$ pin. The valid edge for the count operation depends on the $CNTR_0$ active edge switch bit (bit 2) or the $CNTR_1$ active edge switch bit (bit 6) of the timer XY mode register (address 0023_{16}). When it is “0”, the rising edge is valid. When it is “1”, the falling edge is valid.

■Precautions

Set the double-function port of $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to input in this mode.

(4) Pulse width measurement mode

●Mode selection

This mode can be selected by setting “11” to the timer X operating mode bits (bits 1 and 0) and the timer Y operating mode bits (bits 5 and 4) of the timer XY mode register (address 0023_{16}).

●Explanation of operation

When the $CNTR_0$ active edge switch bit (bit 2) or the $CNTR_1$ active edge switch bit (bit 6) of the timer XY mode register (address 0023_{16}) is “1”, the timer counts during the term of one falling edge of $CNTR_0$/$CNTR_1$ pin input until the next rising edge of input (“L” term). When it is “0”, the timer counts during the term of one rising edge input until the next falling edge input (“H” term).

■Precautions

Set the double-function port of $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to input in this mode.

The count operation can be stopped by setting “1” to the timer X count stop bit (bit 3) and the timer Y count stop bit (bit 7) of the timer XY mode register (address 0023_{16}). The interrupt request bit is set to “1” each time the timer underflows.

•Precautions when switching count source

When switching the count source by the timer 12, X and Y count source selection bits, the value of timer count is altered in inconsiderable amount owing to generating of thin pulses on the count input signals.

Therefore, select the timer count source before setting the value to the prescaler and the timer.

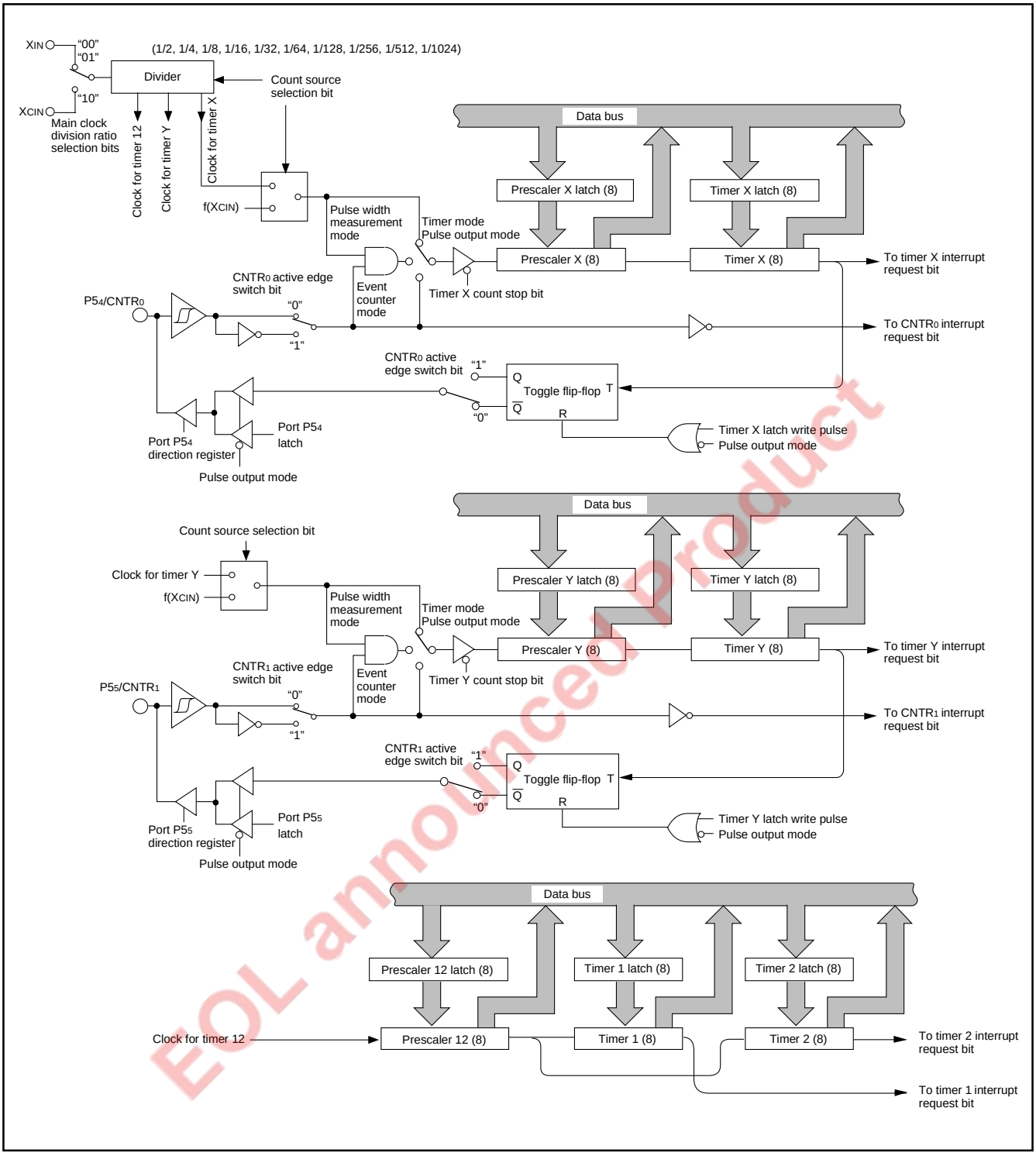


Fig. 21 Block diagram of timer X, timer Y, timer 1, and timer 2

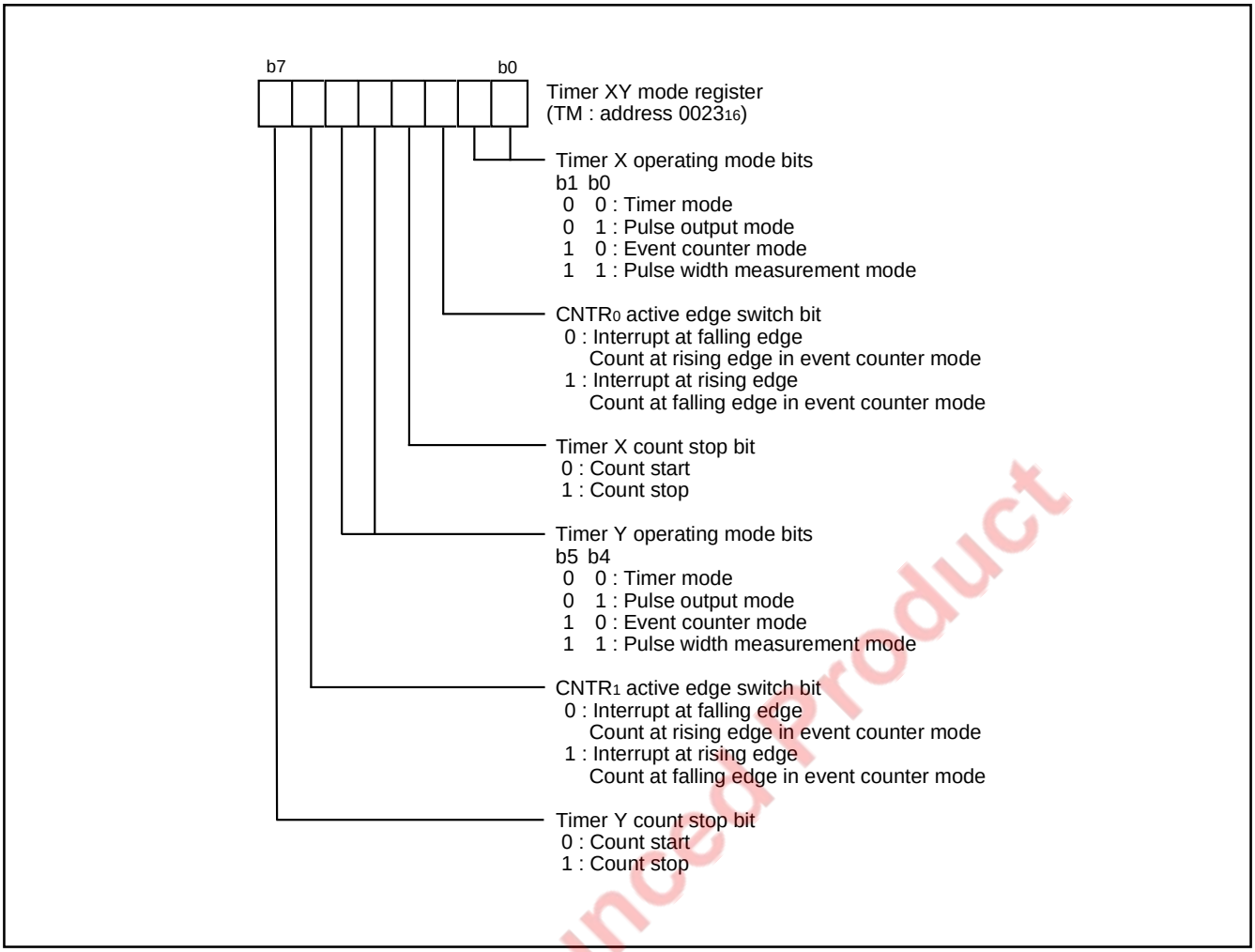


Fig. 22 Structure of timer XY mode register

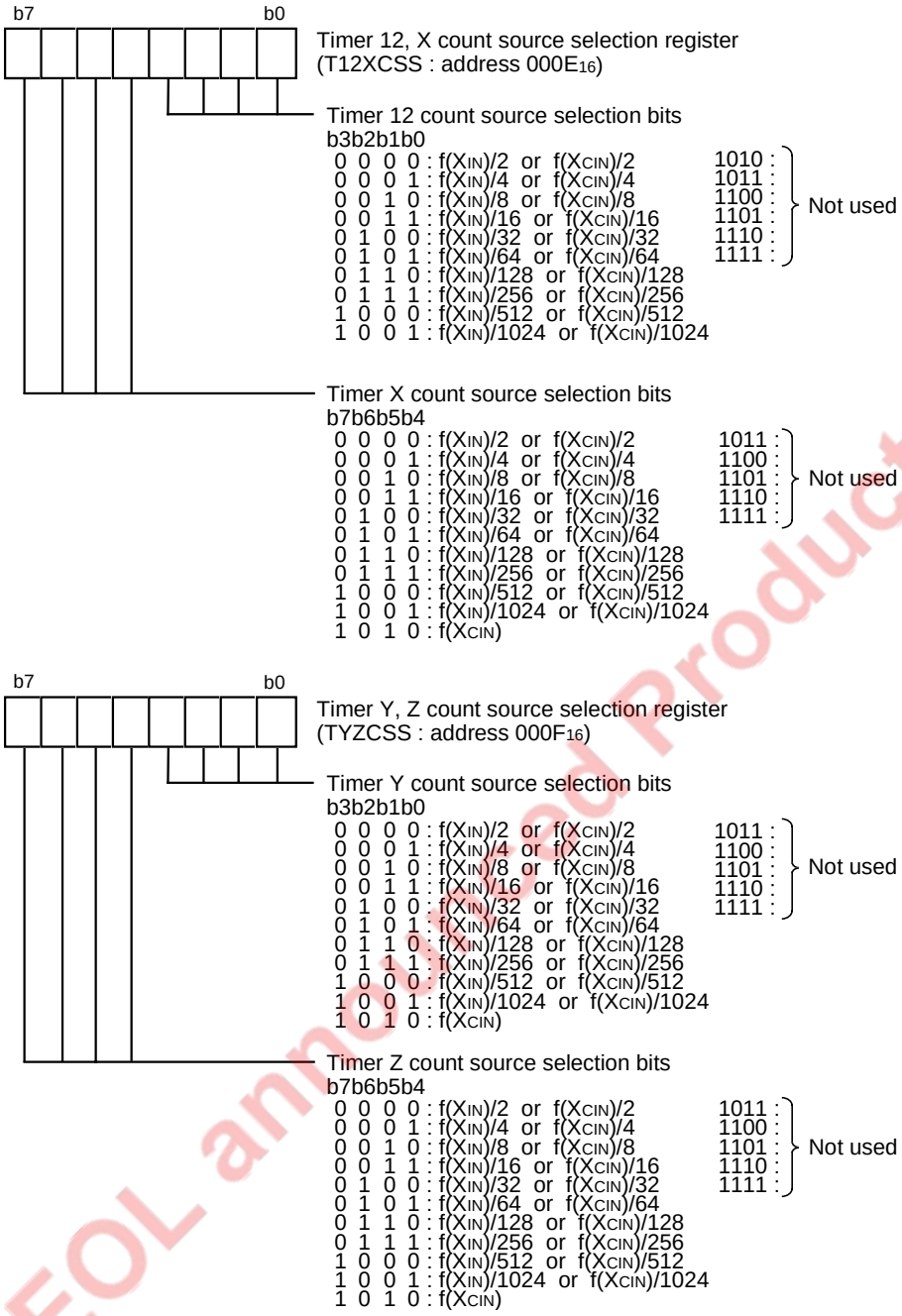


Fig. 23 Structure of timer 12, X and timer Y, Z count source selection registers

●16-bit Timer

The timer Z is a 16-bit timer. When the timer reaches "0000$_{16}$", an underflow occurs at the next count pulse and the corresponding timer latch is reloaded into the timer and the count is continued. When the timer underflows, the interrupt request bit corresponding to the timer Z is set to "1".

When reading/writing to the timer Z, perform reading/writing to both the high-order byte and the low-order byte. When reading the timer Z, read from the high-order byte first, followed by the low-order byte. Do not perform the writing to the timer Z between read operation of the high-order byte and read operation of the low-order byte. When writing to the timer Z, write to the low-order byte first, followed by the high-order byte. Do not perform the reading to the timer Z between write operation of the low-order byte and write operation of the high-order byte.

The timer Z can select the count source by the timer Z count source selection bits of timer Y, Z count source selection register (bits 7 to 4 at address 000F$_{16}$).

Timer Z can select one of seven operating modes by setting the timer Z mode register (address 002A$_{16}$).

(1) Timer mode

●Mode selection

This mode can be selected by setting "000" to the timer Z operating mode bits (bits 2 to 0) and setting "0" to the timer/event counter mode switch bit (b7) of the timer Z mode register (address 002A$_{16}$).

●Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XIN); or f(XCIN) can be selected as the count source.

In low-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XCIN); or f(XCIN) can be selected as the count source.

●Interrupt

When an underflow occurs, the INT$_0$/timer Z interrupt request bit (bit 0) of the interrupt request register 1 (address 003C$_{16}$) is set to "1".

●Explanation of operation

During timer stop, usually write data to a latch and a timer at the same time to set the timer value.

The timer count operation is started by setting "0" to the timer Z count stop bit (bit 6) of the timer Z mode register (address 002A$_{16}$).

When the timer reaches "0000$_{16}$", an underflow occurs at the next count pulse and the contents of timer latch are reloaded into the timer and the count is continued.

When writing data to the timer during operation, the data is written only into the latch. Then the new latch value is reloaded into the timer at the next underflow.

(2) Event counter mode

●Mode selection

This mode can be selected by setting "000" to the timer Z operating mode bits (bits 2 to 0) and setting "1" to the timer/event counter mode switch bit (bit 7) of the timer Z mode register (address 002A$_{16}$).

The valid edge for the count operation depends on the CNTR$_2$ active edge switch bit (bit 5) of the timer Z mode register (address 002A$_{16}$). When it is "0", the rising edge is valid. When it is "1", the falling edge is valid.

●Interrupt

The interrupt at an underflow is the same as the timer mode's.

●Explanation of operation

The operation is the same as the timer mode's.

Set the double-function port of CNTR$_2$ pin and port P4$_7$ to input in this mode.

Figure 26 shows the timing chart of the timer/event counter mode.

(3) Pulse output mode

●Mode selection

This mode can be selected by setting "001" to the timer Z operating mode bits (bits 2 to 0) and setting "0" to the timer/event counter mode switch bit (b7) of the timer Z mode register (address 002A$_{16}$).

●Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XIN); or f(XCIN) can be selected as the count source.

In low-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XCIN); or f(XCIN) can be selected as the count source.

●Interrupt

The interrupt at an underflow is the same as the timer mode's.

●Explanation of operation

The operation is the same as the timer mode's. Moreover the pulse which is inverted each time the timer underflows is output from CNTR$_2$ pin. When the CNTR$_2$ active edge switch bit (bit 5) of the timer Z mode register (address 002A$_{16}$) is "0", the output starts with "H" level. When it is "1", the output starts with "L" level.

■Precautions

The double-function port of CNTR$_2$ pin and port P4$_7$ is automatically set to the timer pulse output port in this mode.

The output from CNTR$_2$ pin is initialized to the level depending on CNTR$_2$ active edge switch bit by writing to the timer.

When the value of the CNTR$_2$ active edge switch bit is changed, the output level of CNTR$_2$ pin is inverted.

Figure 27 shows the timing chart of the pulse output mode.

(4) Pulse period measurement mode

●Mode selection

This mode can be selected by setting “010” to the timer Z operating mode bits (bits 2 to 0) and setting “0” to the timer/event counter mode switch bit (b7) of the timer Z mode register (address $002A_{16}$).

●Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(X_{IN}); or f(X_{CIN}) can be selected as the count source.

In low-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(X_{CIN}); or f(X_{CIN}) can be selected as the count source.

●Interrupt

The interrupt at an underflow is the same as the timer mode’s. When the pulse period measurement is completed, the INT_4/$CNTR_2$ interrupt request bit (bit 5) of the interrupt request register 2 (address $003D_{16}$) is set to “1”.

●Explanation of operation

The cycle of the pulse which is input from the $CNTR_2$ pin is measured. When the $CNTR_2$ active edge switch bit (bit 5) of the timer Z mode register (address $002A_{16}$) is “0”, the timer counts during the term from one falling edge of $CNTR_2$ pin input to the next falling edge. When it is “1”, the timer counts during the term from one rising edge input to the next rising edge input.

When the valid edge of measurement completion/start is detected, the 1’s complement of the timer value is written to the timer latch and “$FFFF_{16}$” is set to the timer.

Furthermore when the timer underflows, the timer Z interrupt request occurs and “$FFFF_{16}$” is set to the timer. When reading the timer Z, the value of the timer latch (measured value) is read. The measured value is retained until the next measurement completion.

■Precautions

Set the double-function port of $CNTR_2$ pin and port $P4_7$ to input in this mode.

A read-out of timer value is impossible in this mode. The timer can be written to only during timer stop (no measurement of pulse period).

Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operation during measurement.

“$FFFF_{16}$” is set to the timer when the timer underflows or when the valid edge of measurement start/completion is detected. Consequently, the timer value at start of pulse period measurement depends on the timer value just before measurement start.

Figure 28 shows the timing chart of the pulse period measurement mode.

(5) Pulse width measurement mode

●Mode selection

This mode can be selected by setting “011” to the timer Z operating mode bits (bits 2 to 0) and setting “0” to the timer/event counter mode switch bit (b7) of the timer Z mode register (address $002A_{16}$).

●Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(X_{IN}); or f(X_{CIN}) can be selected as the count source.

In low-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(X_{CIN}); or f(X_{CIN}) can be selected as the count source.

●Interrupt

The interrupt at an underflow is the same as the timer mode’s. When the pulse widths measurement is completed, the INT_4/$CNTR_2$ interrupt request bit (bit 5) of the interrupt request register 2 (address $003D_{16}$) is set to “1”.

●Explanation of operation

The pulse width which is input from the $CNTR_2$ pin is measured. When the $CNTR_2$ active edge switch bit (bit 5) of the timer Z mode register (address $002A_{16}$) is “0”, the timer counts during the term from one rising edge input to the next falling edge input (“H” term). When it is “1”, the timer counts during the term from one falling edge of $CNTR_2$ pin input to the next rising edge of input (“L” term).

When the valid edge of measurement completion is detected, the 1’s complement of the timer value is written to the timer latch.

When the valid edge of measurement completion/start is detected, “$FFFF_{16}$” is set to the timer.

When the timer Z underflows, the timer Z interrupt occurs and “$FFFF_{16}$” is set to the timer Z. When reading the timer Z, the value of the timer latch (measured value) is read. The measured value is retained until the next measurement completion.

■Precautions

Set the double-function port of $CNTR_2$ pin and port $P4_7$ to input in this mode.

A read-out of timer value is impossible in this mode. The timer can be written to only during timer stop (no measurement of pulse widths).

Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operation during measurement.

“$FFFF_{16}$” is set to the timer when the timer underflows or when the valid edge of measurement start/completion is detected. Consequently, the timer value at start of pulse width measurement depends on the timer value just before measurement start.

Figure 29 shows the timing chart of the pulse width measurement mode.

(6) Programmable waveform generating mode

●Mode selection

This mode can be selected by setting "100" to the timer Z operating mode bits (bits 2 to 0) and setting "0" to the timer/event counter mode switch bit (b7) of the timer Z mode register (address 002A16).

●Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XIN); or f(XCIN) can be selected as the count source.

In low-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XCIN); or f(XCIN) can be selected as the count source.

●Interrupt

The interrupt at an underflow is the same as the timer mode's.

●Explanation of operation

The operation is the same as the timer mode's. Moreover the timer outputs the data set in the output level latch (bit 4) of the timer Z mode register (address 002A16) from the CNTR2 pin each time the timer underflows.

Changing the value of the output level latch and the timer latch after an underflow makes it possible to output an optional waveform from the CNTR2 pin.

■Precautions

The double-function port of CNTR2 pin and port P47 is automatically set to the programmable waveform generating port in this mode.

Figure 30 shows the timing chart of the programmable waveform generating mode.

(7) Programmable one-shot generating mode

●Mode selection

This mode can be selected by setting "101" to the timer Z operating mode bits (bits 2 to 0) and setting "0" to the timer/event counter mode switch bit (b7) of the timer Z mode register (address 002A16).

●Count source selection

In high- or middle-speed mode, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 or 1/1024 of f(XIN); or f(XCIN) can be selected as the count source.

●Interrupt

The interrupt at an underflow is the same as the timer mode's.

The trigger to generate one-shot pulse can be selected by the INT1 active edge selection bit (bit 1) of the interrupt edge selection register (address 003A16). When it is "0", the falling edge active is selected; when it is "1", the rising edge active is selected.

When the valid edge of the INT1 pin is detected, the INT1 interrupt request bit (bit 1) of the interrupt request register 1 (address 003C16) is set to "1".

●Explanation of operation

•"H" one-shot pulse; Bit 5 of timer Z mode register = "0"

The output level of the CNTR2 pin is initialized to "L" at mode selection. When trigger generation (input signal to INT1 pin) is detected, "H" is output from the CNTR2 pin. When an underflow occurs, "L" is output. The "H" one-shot pulse width is set by the setting value to the timer Z register low-order and high-order. When trigger generating is detected during timer count stop, although "H" is output from the CNTR2 pin, "H" output state continues because an underflow does not occur.

•"L" one-shot pulse; Bit 5 of timer Z mode register = "1"

The output level of the CNTR2 pin is initialized to "H" at mode selection. When trigger generation (input signal to INT1 pin) is detected, "L" is output from the CNTR2 pin. When an underflow occurs, "H" is output. The "L" one-shot pulse width is set by the setting value to the timer Z low-order and high-order. When trigger generating is detected during timer count stop, although "L" is output from the CNTR2 pin, "L" output state continues because an underflow does not occur.

■Precautions

Set the double-function port of INT1 pin and port P42 to input in this mode.

Set the double-function port of CNTR2 pin and port P22 is automatically set to the programmable one-shot generating port in this mode.

This mode cannot be used in low-speed mode.

If the value of the CNTR2 active edge switch bit is changed during one-shot generating enabled or generating one-shot pulse, then the output level from CNTR2 pin changes.

Figure 31 shows the timing chart of the programmable one-shot generating mode.

■Notes regarding all modes

●Timer Z write control

Which write control can be selected by the timer Z write control bit (bit 3) of the timer Z mode register (address 002A16), writing data to both the latch and the timer at the same time or writing data only to the latch.

When the operation "writing data only to the latch" is selected, the value is set to the timer latch by writing data to the address of timer Z and the timer is updated at next underflow. After reset release, the operation "writing data to both the latch and the timer at the same time" is selected, and the value is set to both the latch and the timer at the same time by writing data to the address of timer Z.

In the case of writing data only to the latch, if writing data to the latch and an underflow are performed almost at the same time, the timer value may become undefined.

●Timer Z read control

A read-out of timer value is impossible in pulse period measurement mode and pulse width measurement mode. In the other modes, a read-out of timer value is possible regardless of count operating or stopped.

However, a read-out of timer latch value is impossible.

●Switch of interrupt active edge of CNTR2 and INT1

Each interrupt active edge depends on setting of the CNTR2 active edge switch bit and the INT1 active edge selection bit.

●Switch of count source

When switching the count source by the timer Z count source selection bits, the value of timer count is altered in inconsiderable amount owing to generating of thin pulses on the count input signals.

Therefore, select the timer count source before setting the value to the prescaler and the timer.

●Usage of CNTR2 pin as normal I/O port

To use the CNTR2 pin as normal I/O port P47, set timer Z operating mode bits (b2, b1, b0) of timer Z mode register (address 002A16) to "000".

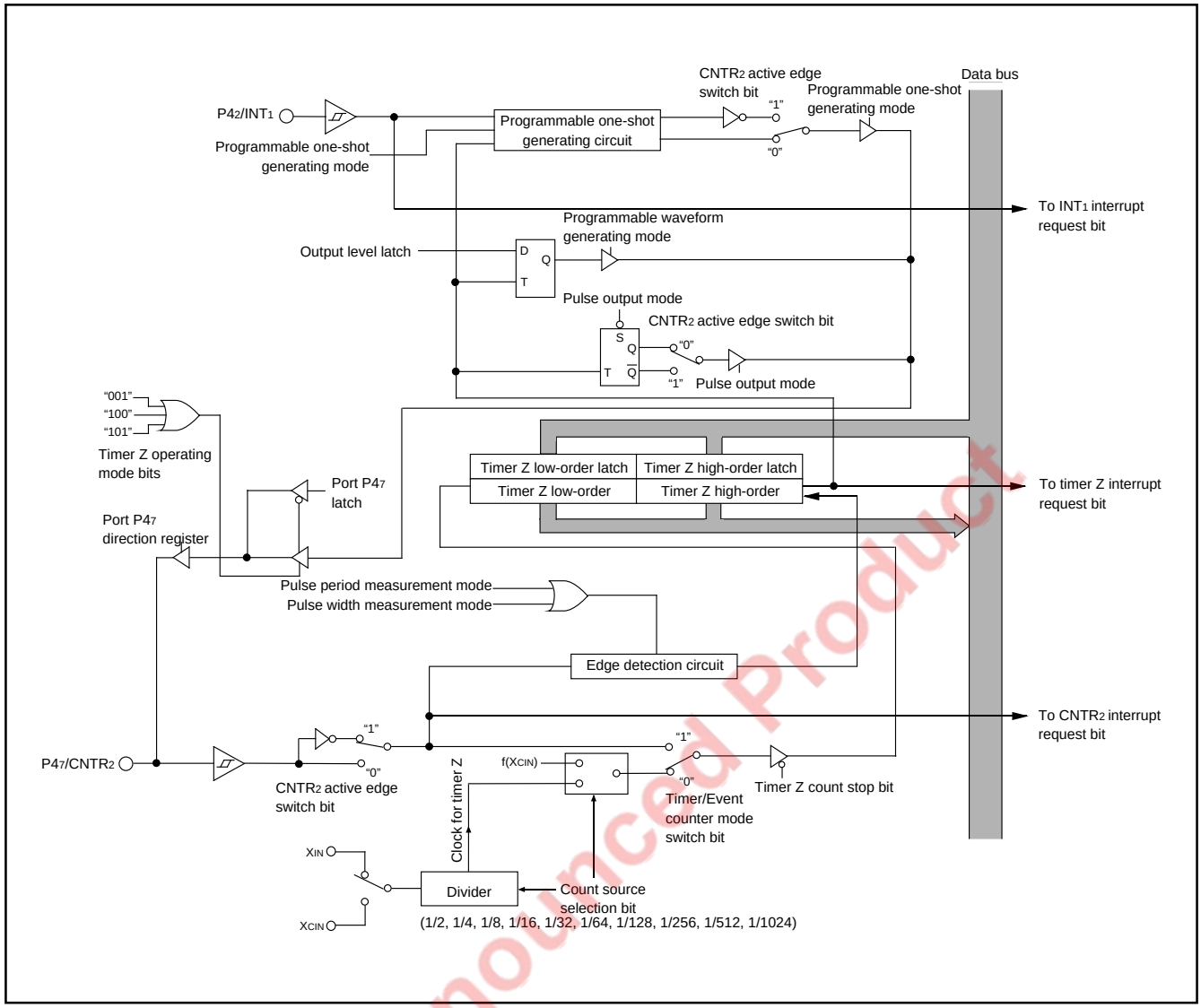


Fig. 24 Block diagram of timer Z

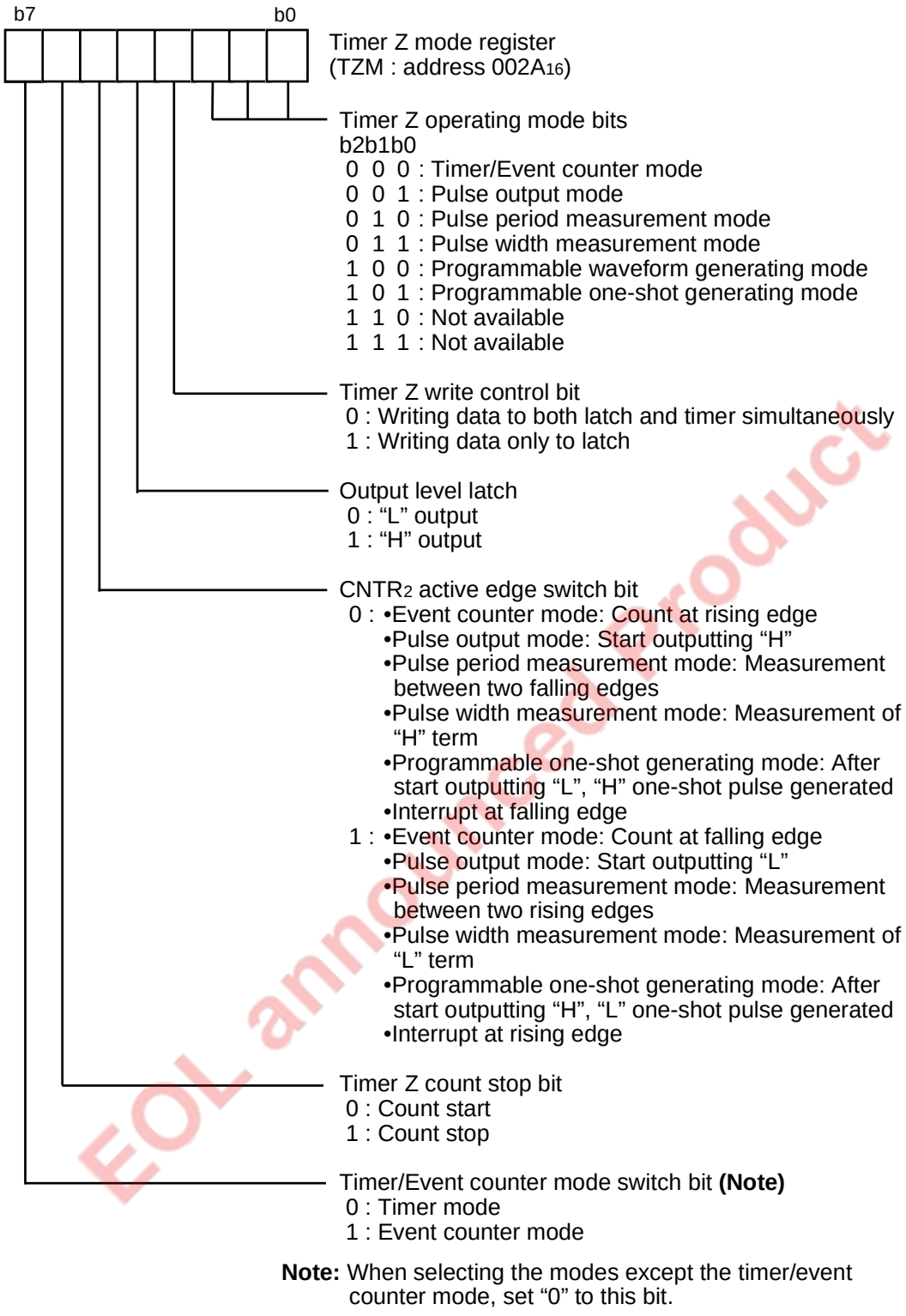
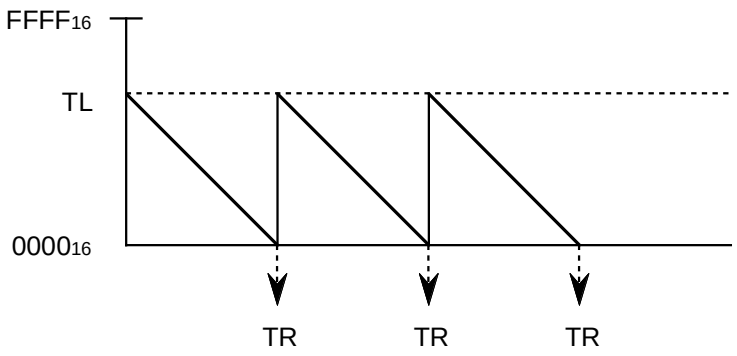
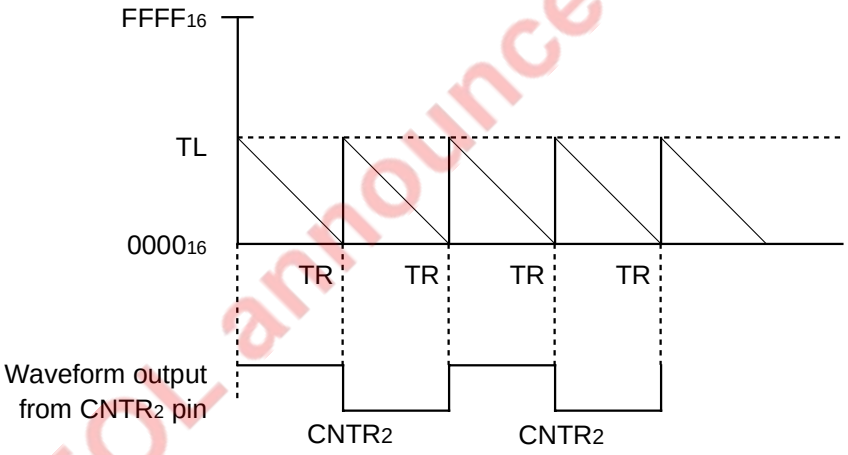


Fig. 25 Structure of timer Z mode register



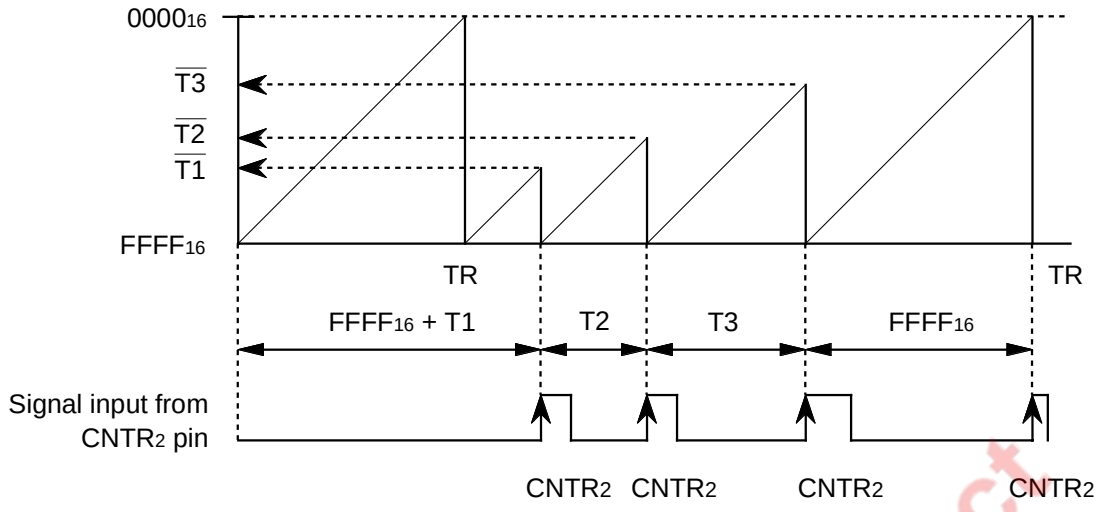
TL : Value set to timer latch
TR : Timer interrupt request

Fig. 26 Timing chart of timer/event counter mode



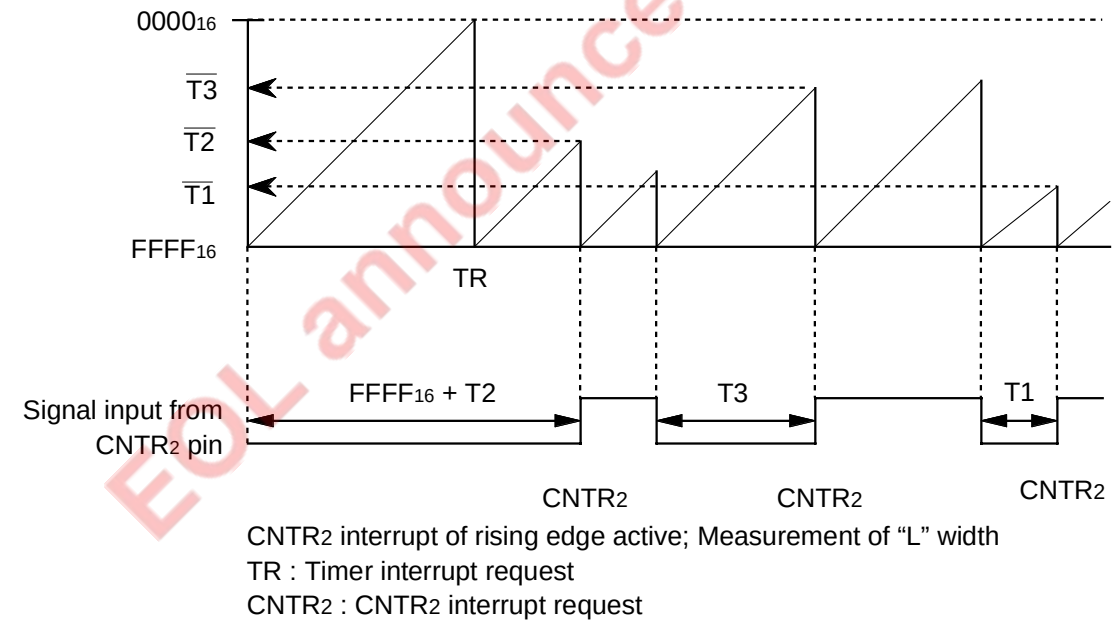
TL : Value set to timer latch
TR : Timer interrupt request
CNTR2 : CNTR2 interrupt request
(CNTR2 active edge switch bit = “0”; Falling edge active)

Fig. 27 Timing chart of pulse output mode



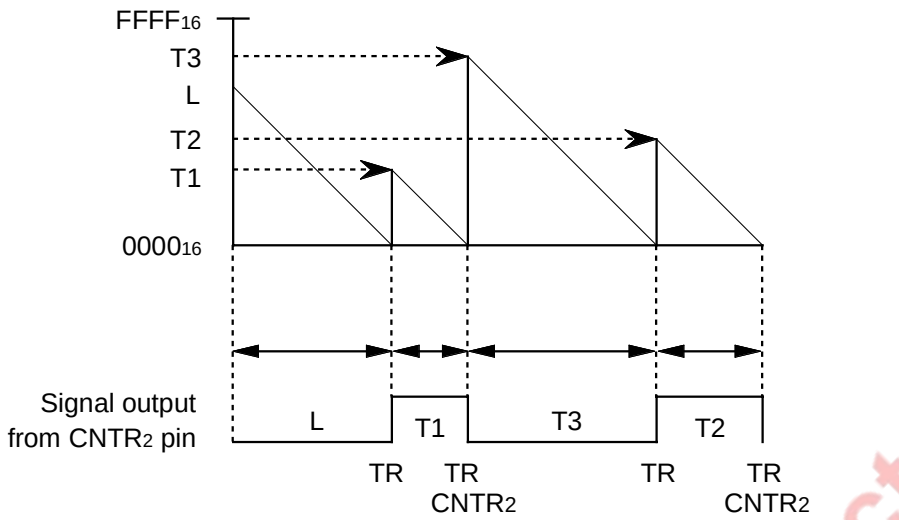
CNTR2 of rising edge active
TR : Timer interrupt request
CNTR2 : CNTR2 interrupt request

Fig. 28 Timing chart of pulse period measurement mode (Measuring term between two rising edges)



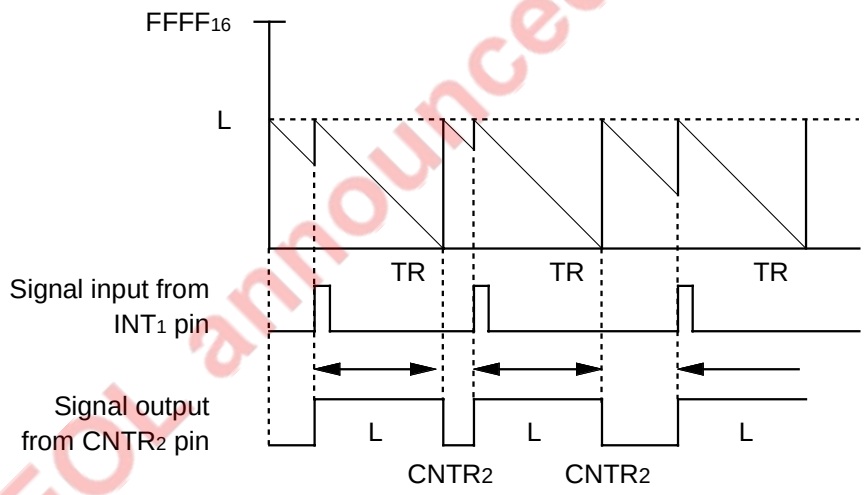
CNTR2 interrupt of rising edge active; Measurement of “L” width
TR : Timer interrupt request
CNTR2 : CNTR2 interrupt request

Fig. 29 Timing chart of pulse width measurement mode (Measuring “L” term)



L : Timer initial value
TR : Timer interrupt request
CNTR2 : CNTR2 interrupt request
(CNTR2 active edge switch bit = “0”; Falling edge active)

Fig. 30 Timing chart of programmable waveform generating mode



L : One-shot pulse width
TR : Timer interrupt request
CNTR2 : CNTR2 interrupt request
(CNTR2 active edge switch bit = “0”; Falling edge active)

Fig. 31 Timing chart of programmable one-shot generating mode (“H” one-shot pulse generating)

SERIAL INTERFACE
Serial I/O1

Serial I/O1 can be used as either clock synchronous or asynchronous (UART) serial I/O1. A dedicated timer is also provided for baud rate generation.

(1) Clock Synchronous Serial I/O Mode

Clock synchronous serial I/O1 mode can be selected by setting the serial I/O1 mode selection bit of the serial I/O1 control register (bit 6 of address 001A16) to "1".

For clock synchronous serial I/O, the transmitter and the receiver must use the same clock. If an internal clock is used, transfer is started by a write signal to the transmit/receive buffer register.

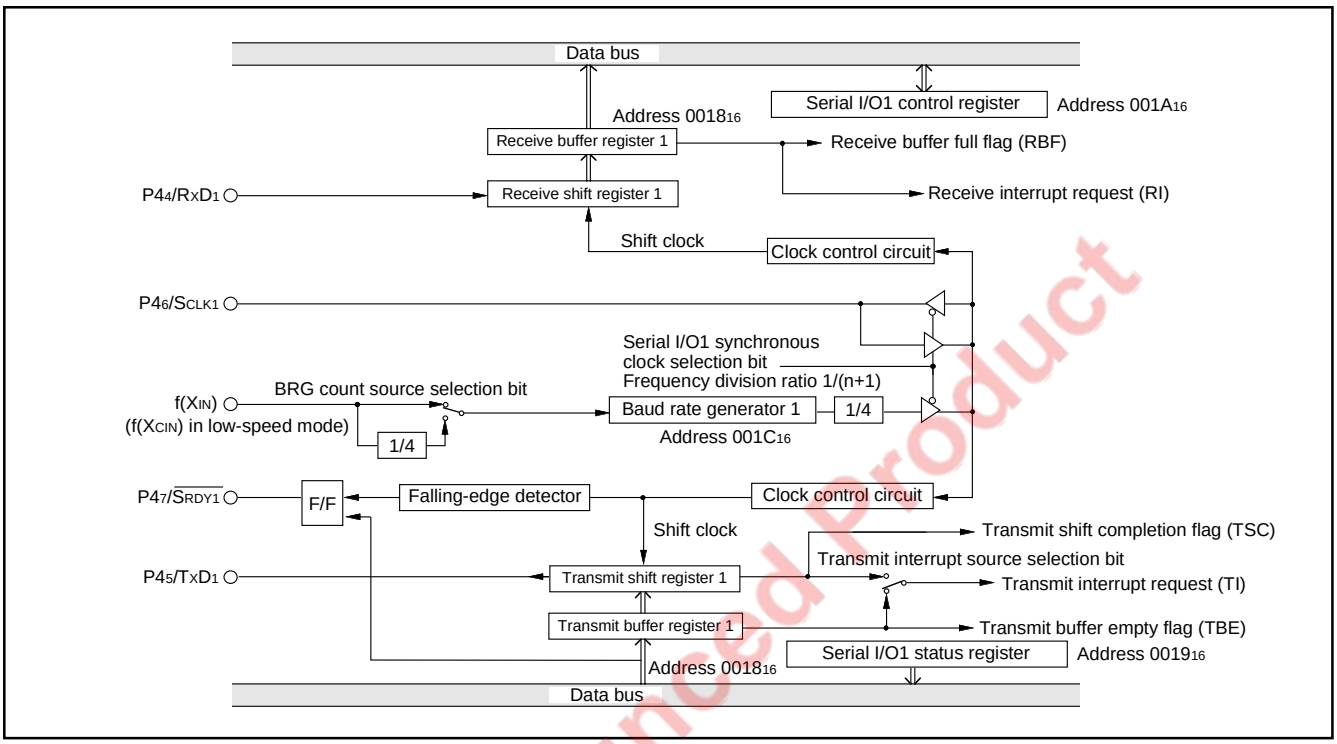
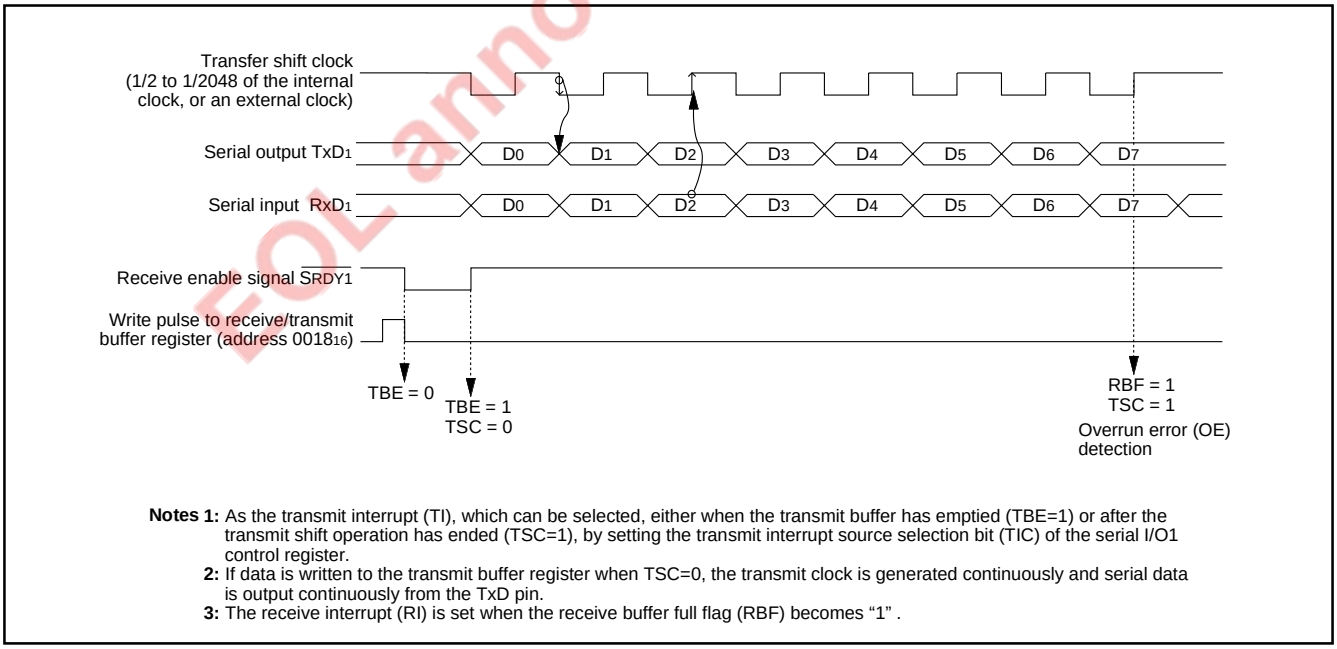


Fig. 32 Block diagram of clock synchronous serial I/O1



Notes 1: As the transmit interrupt (TI), which can be selected, either when the transmit buffer has emptied (TBE=1) or after the transmit shift operation has ended (TSC=1), by setting the transmit interrupt source selection bit (TIC) of the serial I/O1 control register.
2: If data is written to the transmit buffer register when TSC=0, the transmit clock is generated continuously and serial data is output continuously from the TxD pin.
3: The receive interrupt (RI) is set when the receive buffer full flag (RBF) becomes "1" .

Fig. 33 Operation of clock synchronous serial I/O1

(2) Asynchronous Serial I/O (UART) Mode

Clock asynchronous serial I/O mode (UART) can be selected by clearing the serial I/O1 mode selection bit of the serial I/O1 control register to “0”.

Eight serial data transfer formats can be selected, and the transfer formats used by a transmitter and receiver must be identical.

The transmit and receive shift registers each have a buffer, but the two buffers have the same address in a memory. Since the shift register cannot be written to or read from directly, transmit data is written to the transmit buffer register, and receive data is read from the receive buffer register.

The transmit buffer register can also hold the next data to be transmitted, and the receive buffer register can hold a character while the next character is being received.

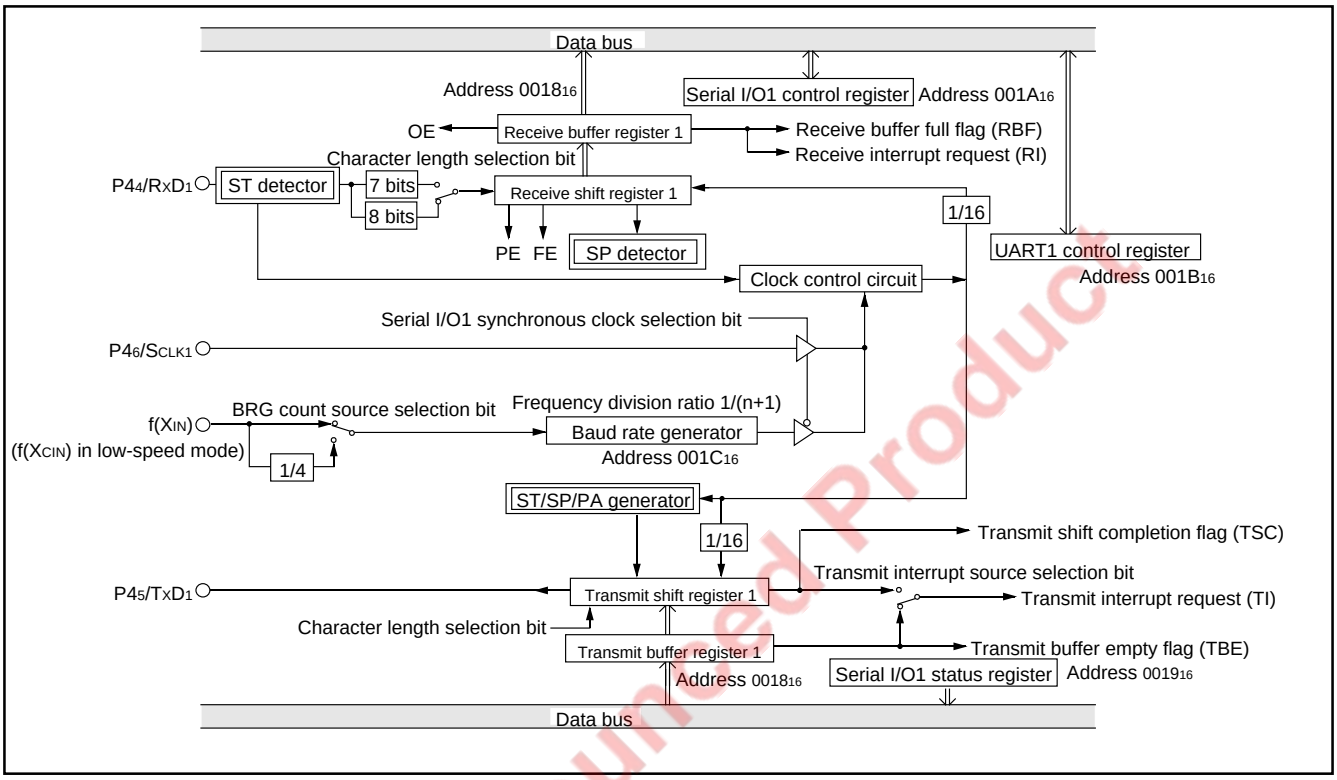
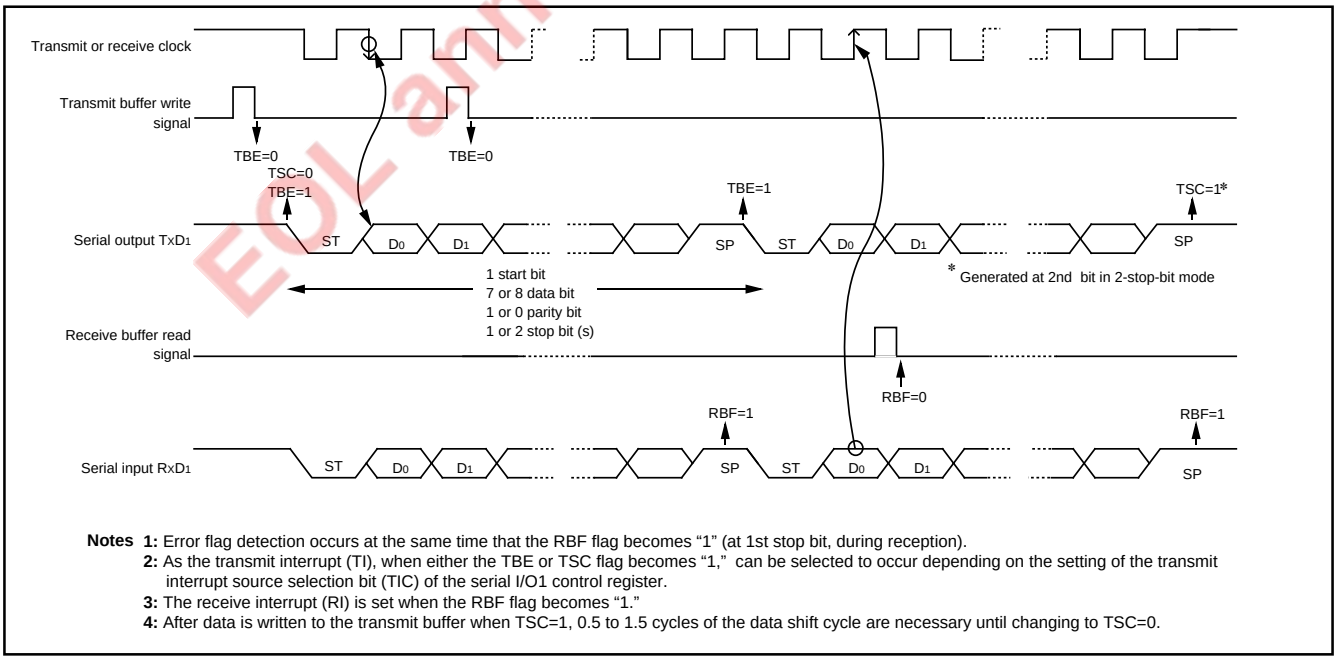


Fig. 34 Block diagram of UART serial I/O1



Notes
1: Error flag detection occurs at the same time that the RBF flag becomes “1” (at 1st stop bit, during reception).
2: As the transmit interrupt (TI), when either the TBE or TSC flag becomes “1,” can be selected to occur depending on the setting of the transmit interrupt source selection bit (TIC) of the serial I/O1 control register.
3: The receive interrupt (RI) is set when the RBF flag becomes “1.”
4: After data is written to the transmit buffer when TSC=1, 0.5 to 1.5 cycles of the data shift cycle are necessary until changing to TSC=0.

Fig. 35 Operation of UART serial I/O1

[Serial I/O1 Control Register (SIO1CON)] 001A16

The serial I/O1 control register consists of eight control bits for the serial I/O1 function.

[UART1 Control Register (UART1CON)] 001B16

The UART control register consists of four control bits (bits 0 to 3) which are valid when asynchronous serial I/O is selected and set the data format of an data transfer, and one bit (bit 4) which is always valid and sets the output structure of the P45/TXD1 pin.

[Serial I/O1 Status Register (SIO1STS)] 001916

The read-only serial I/O1 status register consists of seven flags (bits 0 to 6) which indicate the operating status of the serial I/O1 function and various errors.

Three of the flags (bits 4 to 6) are valid only in UART mode.

The receive buffer full flag (bit 1) is cleared to “0” when the receive buffer register is read.

If there is an error, it is detected at the same time that data is transferred from the receive shift register to the receive buffer register, and the receive buffer full flag is set. A write to the serial I/O1 status register clears all the error flags OE, PE, FE, and SE (bit 3 to bit 6, respectively). Writing “0” to the serial I/O1 enable bit SIOE (bit 7 of the serial I/O1 control register) also clears all the status flags, including the error flags.

Bits 0 to 6 of the serial I/O1 status register are initialized to “0” at reset, but if the transmit enable bit (bit 4) of the serial I/O1 control register has been set to “1”, the transmit shift completion flag (bit 2) and the transmit buffer empty flag (bit 0) become “1”.

[Transmit Buffer Register 1/Receive Buffer Register 1 (TB1/RB1)] 001816

The transmit buffer register 1 and the receive buffer register 1 are located at the same address. The transmit buffer is write-only and the receive buffer is read-only. If a character bit length is 7 bits, the MSB of data stored in the receive buffer is “0”.

[Baud Rate Generator 1 (BRG1)] 001C16

The baud rate generator determines the baud rate for serial transfer.

The baud rate generator divides the frequency of the count source by 1/(n + 1), where n is the value written to the baud rate generator.

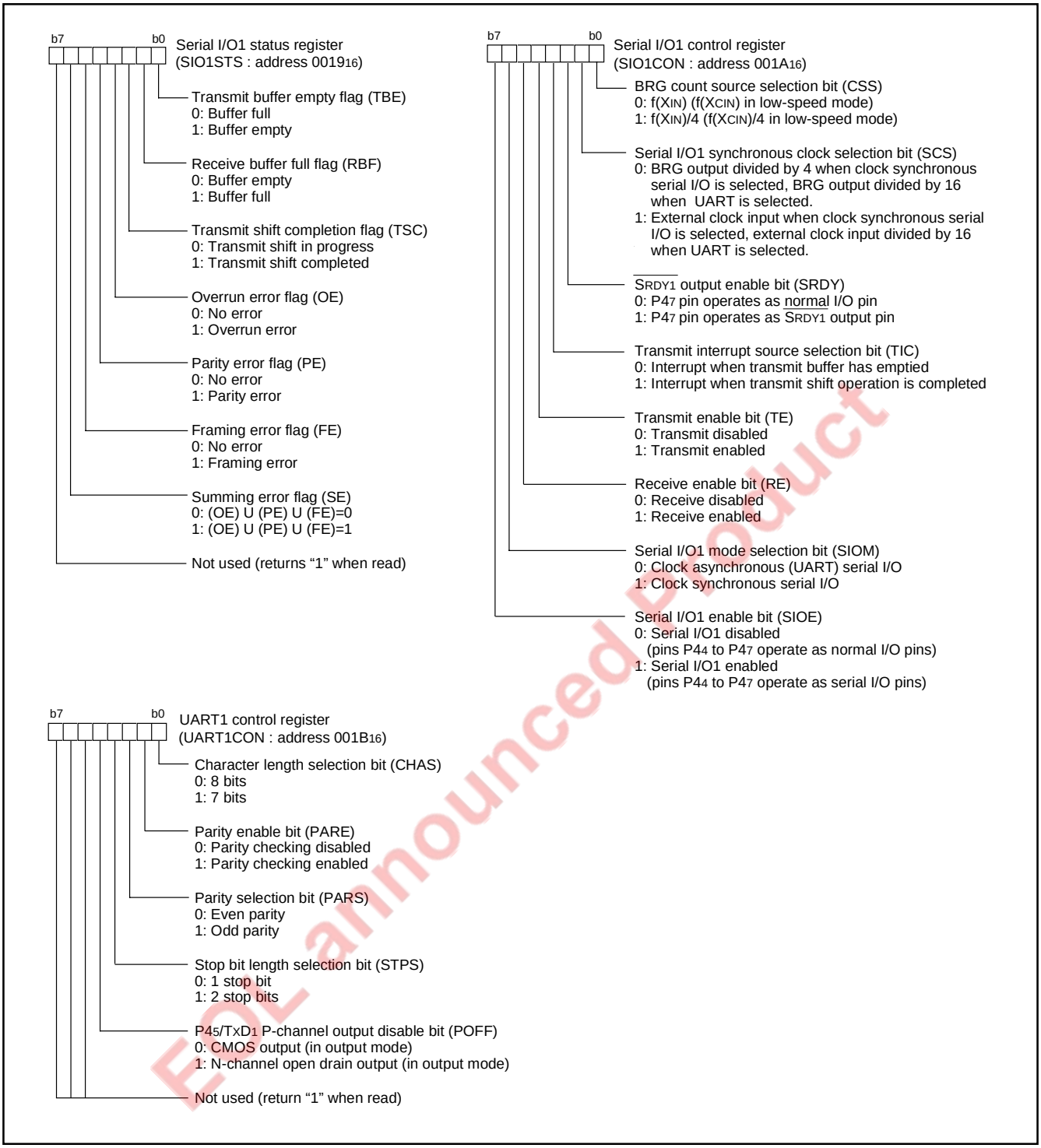


Fig. 36 Structure of serial I/O1 control registers

■ Notes concerning serial I/O1

1. Notes when selecting clock synchronous serial I/O

1.1 Stop of transmission operation

● Note

Clear the serial I/O1 enable bit and the transmit enable bit to “0” (serial I/O and transmit disabled).

● Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O1 enable bit is cleared to “0” (serial I/O disabled), the internal transmission is running (in this case, since pins TxD1, RxD1, SCLK1, and $\overline{\text{SRDY1}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O1 enable bit is set to “1” at this time, the data during internally shifting is output to the TxD1 pin and an operation failure occurs.

1.2 Stop of receive operation

● Note

Clear the receive enable bit to “0” (receive disabled), or clear the serial I/O1 enable bit to “0” (serial I/O disabled).

1.3 Stop of transmit/receive operation

● Note

Clear both the transmit enable bit and receive enable bit to “0” (transmit and receive disabled).
(when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

● Reason

In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.
In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to “0” (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/O1 enable bit to “0” (serial I/O disabled) (refer to **1.1**).

2. Notes when selecting clock asynchronous serial I/O

2.1 Stop of transmission operation

● Note

Clear the transmit enable bit to “0” (transmit disabled). The transmission operation does not stop by clearing the serial I/O1 enable bit to “0”.

● Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O1 enable bit is cleared to “0” (serial I/O disabled), the internal transmission is running (in this case, since pins TxD1, RxD1, SCLK1, and $\overline{\text{SRDY1}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O1 enable bit is set to “1” at this time, the data during internally shifting is output to the TxD1 pin and an operation failure occurs.

2.2 Stop of receive operation

● Note

Clear the receive enable bit to “0” (receive disabled).

2.3 Stop of transmit/receive operation

● Note 1 (only transmission operation is stopped)

Clear the transmit enable bit to “0” (transmit disabled). The transmission operation does not stop by clearing the serial I/O1 enable bit to “0”.

● Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O1 enable bit is cleared to “0” (serial I/O disabled), the internal transmission is running (in this case, since pins TxD1, RxD1, SCLK1, and $\overline{\text{SRDY1}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O1 enable bit is set to “1” at this time, the data during internally shifting is output to the TxD1 pin and an operation failure occurs.

● Note 2 (only receive operation is stopped)

Clear the receive enable bit to “0” (receive disabled).

3. $\overline{\text{SRDY1}}$ output of reception side

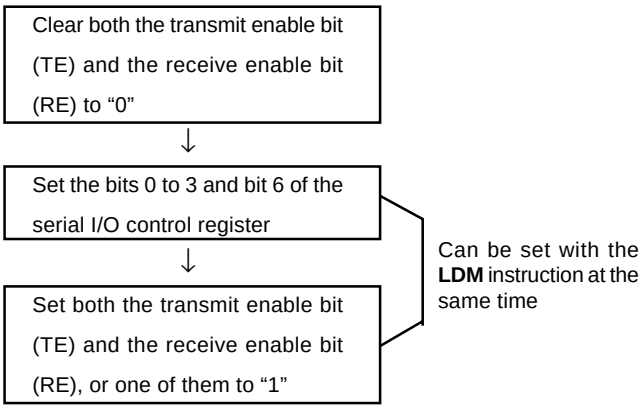
● **Note**

When signals are output from the $\overline{\text{SRDY1}}$ pin on the reception side by using an external clock in the clock synchronous serial I/O mode, set all of the receive enable bit, the $\overline{\text{SRDY1}}$ output enable bit, and the transmit enable bit to "1" (transmit enabled).

4. Setting serial I/O1 control register again

● **Note**

Set the serial I/O1 control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to "0."



5. Data transmission control with referring to transmit shift register completion flag

● **Note**

After the transmit data is written to the transmit buffer register, the transmit shift register completion flag changes from "1" to "0" with a delay of 0.5 to 1.5 shift clocks. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

6. Transmission control when external clock is selected

● **Note**

When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to "1" at "H" of the SCLK1 input level. Also, write data to the transmit buffer register at "H" of the SCLK1 input level.

7. Transmit interrupt request when transmit enable bit is set

● **Note**

When using the transmit interrupt, take the following sequence.

① Set the serial I/O1 transmit interrupt enable bit to "0" (disabled).
② Set the transmit enable bit to "1".
③ Set the serial I/O1 transmit interrupt request bit to "0" after 1 or more instruction has executed.
④ Set the serial I/O1 transmit interrupt enable bit to "1" (enabled).

● **Reason**

When the transmit enable bit is set to "1", the transmit buffer empty flag and the transmit shift register shift completion flag are also set to "1". Therefore, regardless of selecting which timing for the generating of transmit interrupts, the interrupt request is generated and the transmit interrupt request bit is set at this point.

Serial I/O2

The serial I/O2 function can be used only for clock synchronous serial I/O2.

For clock synchronous serial I/O2, the transmitter and the receiver must use the same clock. If the internal clock is used, transfer is started by a write signal to the serial I/O2 register.

[Serial I/O2 Control Register (SIO2CON)] 001D16

The serial I/O2 control register contains eight bits which control various serial I/O2 functions.

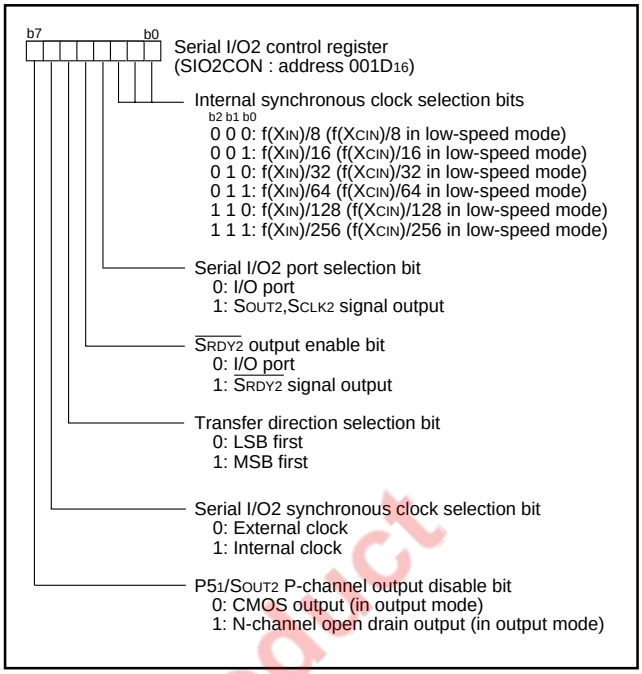


Fig. 37 Structure of serial I/O2 control register

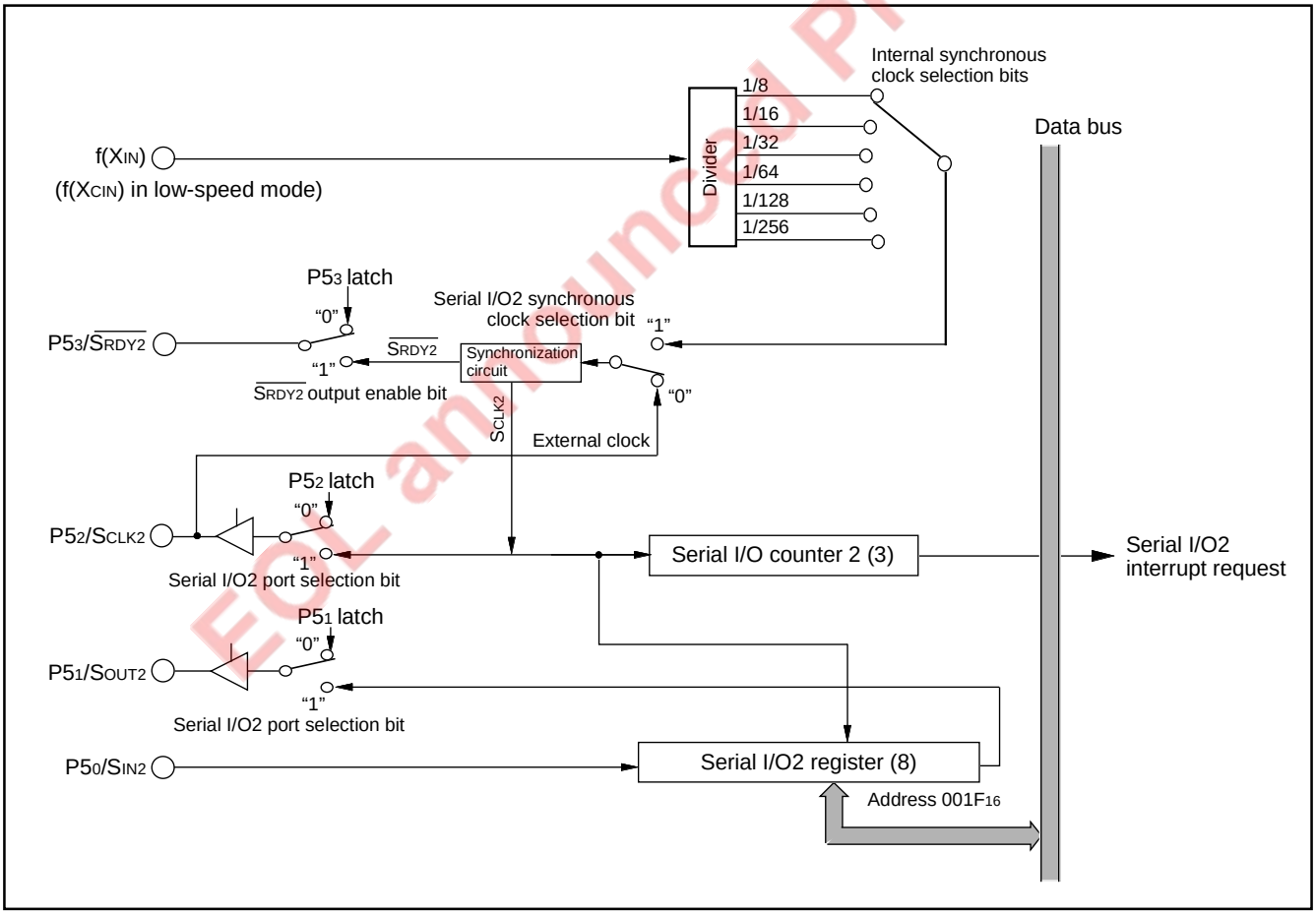
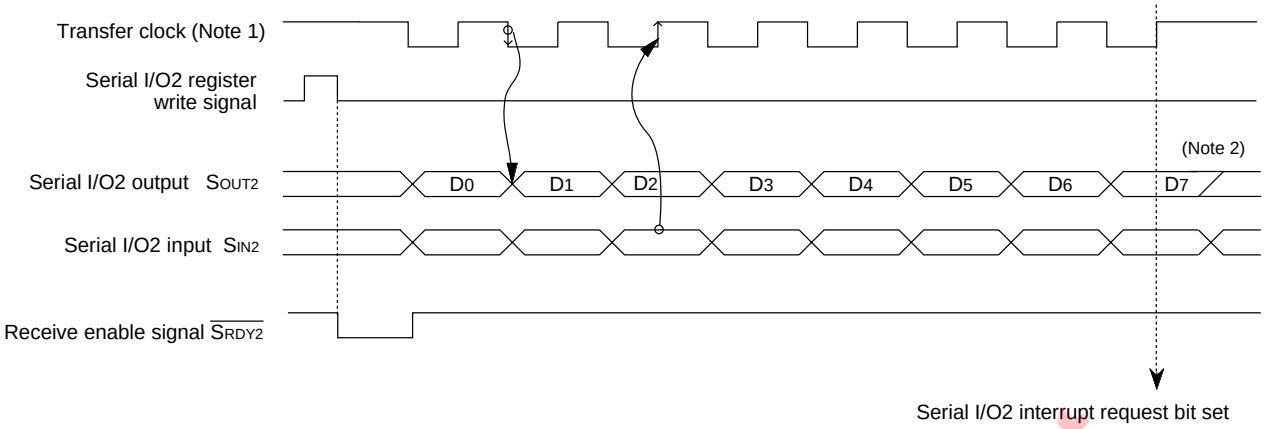


Fig. 38 Block diagram of serial I/O2



Notes 1: When the internal clock is selected as the transfer clock, the divide ratio of f(X_{IN}), or f(X_{CIN}) in low-speed mode, can be selected by setting bits 0 to 2 of the serial I/O2 control register.
2: When the internal clock is selected as the transfer clock, the S_{OUT2} pin goes to high impedance after transfer completion.

Fig. 39 Timing of serial I/O2

Serial I/O3

Serial I/O3 can be used as either clock synchronous or asynchronous (UART) serial I/O3. A dedicated timer is also provided for baud rate generation.

(1) Clock Synchronous Serial I/O Mode

Clock synchronous serial I/O3 mode can be selected by setting the serial I/O3 mode selection bit of the serial I/O3 control register (bit 6 of address 0032_{16}) to "1".

For clock synchronous serial I/O, the transmitter and the receiver must use the same clock. If an internal clock is used, transfer is started by a write signal to the transmit/receive buffer register.

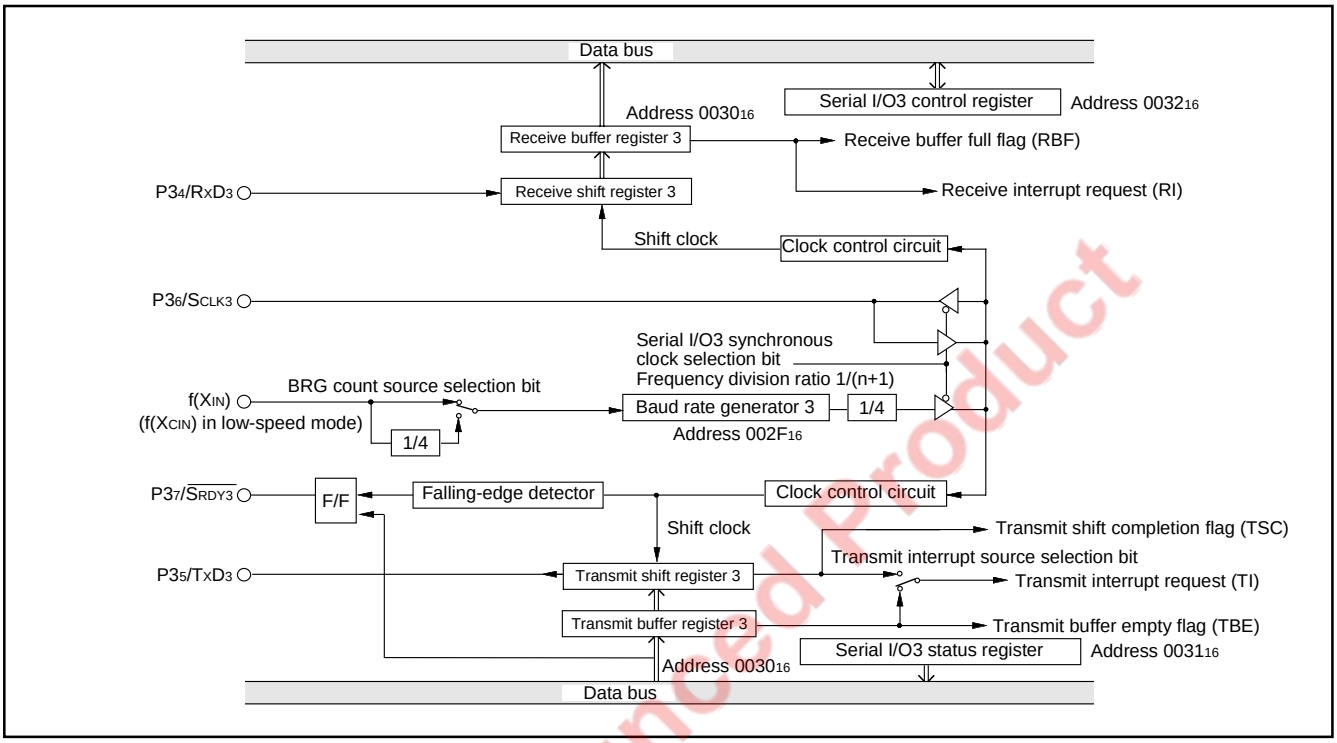
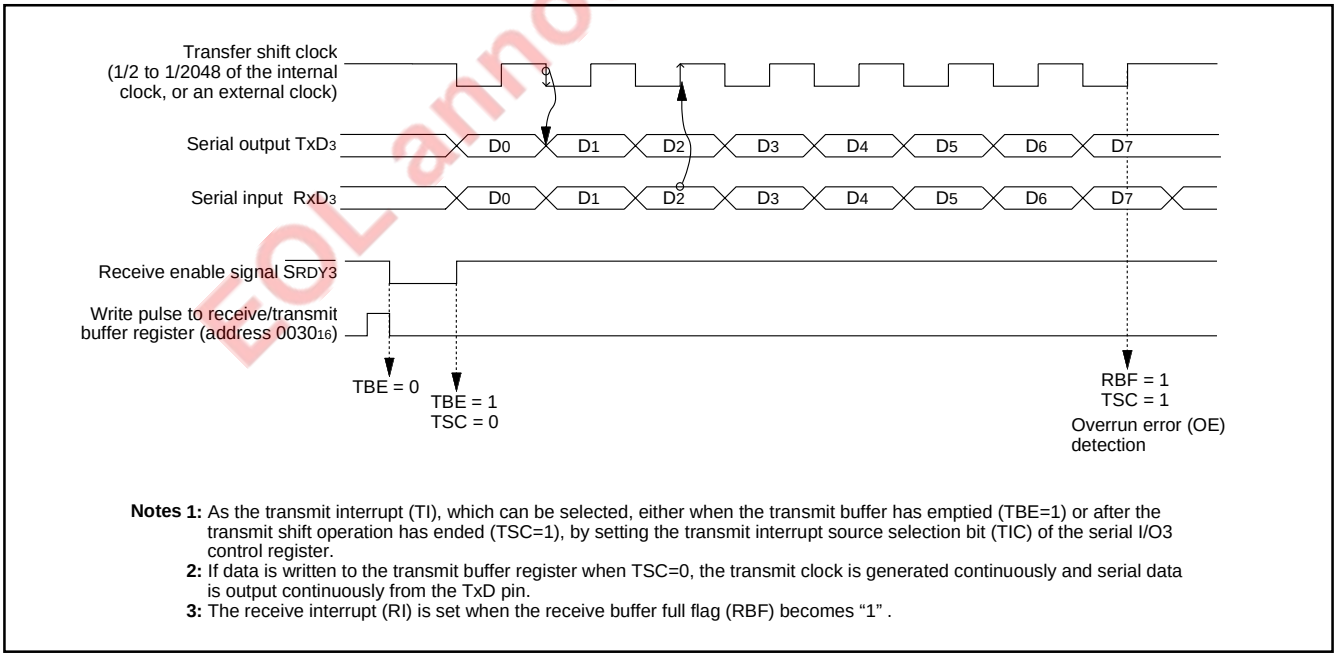


Fig. 40 Block diagram of clock synchronous serial I/O3



Notes 1: As the transmit interrupt (TI), which can be selected, either when the transmit buffer has emptied (TBE=1) or after the transmit shift operation has ended (TSC=1), by setting the transmit interrupt source selection bit (TIC) of the serial I/O3 control register.
2: If data is written to the transmit buffer register when TSC=0, the transmit clock is generated continuously and serial data is output continuously from the TxD pin.
3: The receive interrupt (RI) is set when the receive buffer full flag (RBF) becomes "1" .

Fig. 41 Operation of clock synchronous serial I/O3

(2) Asynchronous Serial I/O (UART) Mode

Clock asynchronous serial I/O mode (UART) can be selected by clearing the serial I/O3 mode selection bit of the serial I/O3 control register to “0”.

Eight serial data transfer formats can be selected, and the transfer formats used by a transmitter and receiver must be identical.

The transmit and receive shift registers each have a buffer, but the two buffers have the same address in a memory. Since the shift register cannot be written to or read from directly, transmit data is written to the transmit buffer register, and receive data is read from the receive buffer register.

The transmit buffer register can also hold the next data to be transmitted, and the receive buffer register can hold a character while the next character is being received.

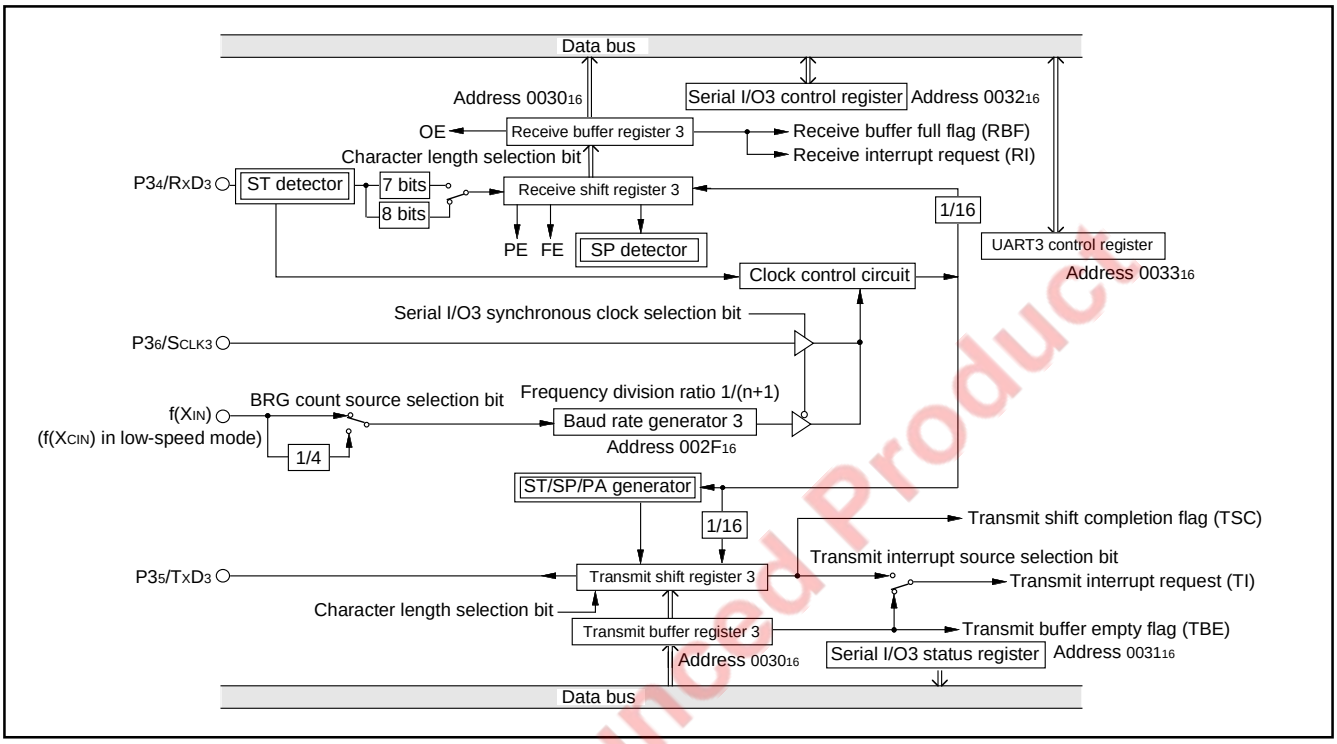
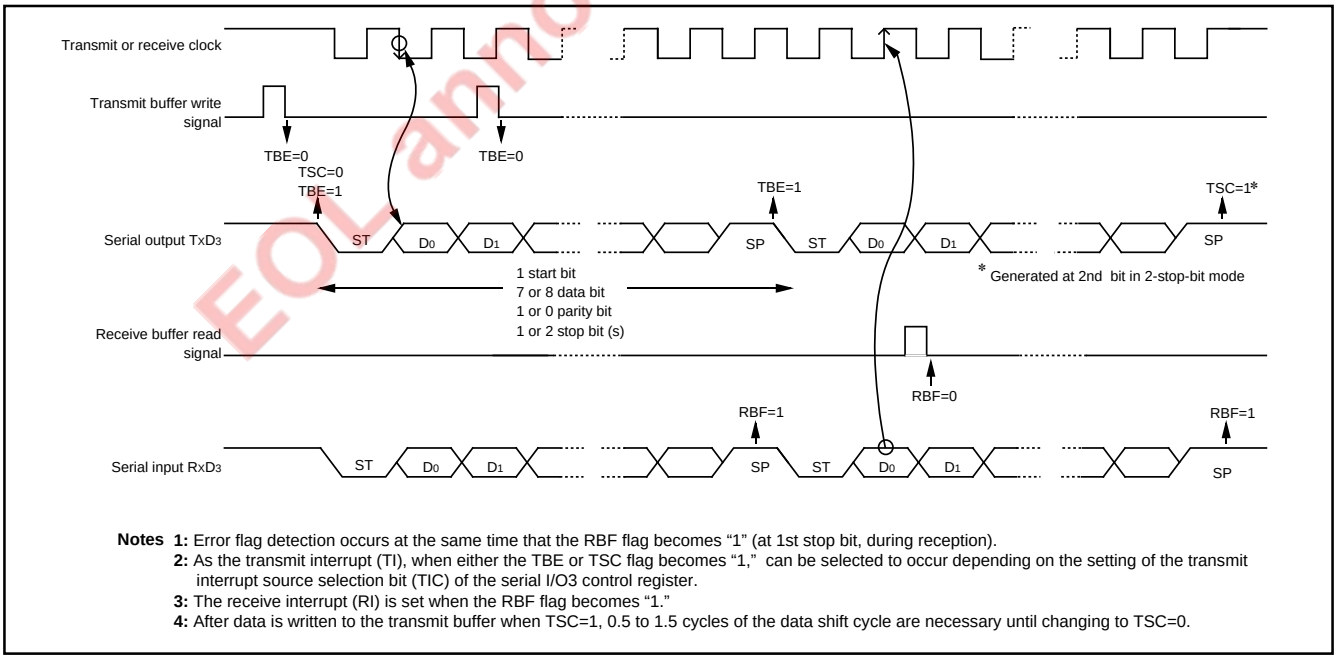


Fig. 42 Block diagram of UART serial I/O3



Notes
1: Error flag detection occurs at the same time that the RBF flag becomes “1” (at 1st stop bit, during reception).
2: As the transmit interrupt (TI), when either the TBE or TSC flag becomes “1,” can be selected to occur depending on the setting of the transmit interrupt source selection bit (TIC) of the serial I/O3 control register.
3: The receive interrupt (RI) is set when the RBF flag becomes “1.”
4: After data is written to the transmit buffer when TSC=1, 0.5 to 1.5 cycles of the data shift cycle are necessary until changing to TSC=0.

Fig. 43 Operation of UART serial I/O3

[Serial I/O3 Control Register (SIO3CON)] 0032$_{16}$

The serial I/O3 control register consists of eight control bits for the serial I/O3 function.

[UART3 Control Register (UART3CON)] 0033$_{16}$

The UART control register consists of four control bits (bits 0 to 3) which are valid when asynchronous serial I/O is selected and set the data format of an data transfer, and one bit (bit 4) which is always valid and sets the output structure of the P3$_5$/TxD$_3$ pin.

[Serial I/O3 Status Register (SIO3STS)] 0031$_{16}$

The read-only serial I/O3 status register consists of seven flags (bits 0 to 6) which indicate the operating status of the serial I/O3 function and various errors.

Three of the flags (bits 4 to 6) are valid only in UART mode.

The receive buffer full flag (bit 1) is cleared to “0” when the receive buffer register is read.

If there is an error, it is detected at the same time that data is transferred from the receive shift register to the receive buffer register, and the receive buffer full flag is set. A write to the serial I/O3 status register clears all the error flags OE, PE, FE, and SE (bit 3 to bit 6, respectively). Writing “0” to the serial I/O3 enable bit SIOE (bit 7 of the serial I/O3 control register) also clears all the status flags, including the error flags.

Bits 0 to 6 of the serial I/O3 status register are initialized to “0” at reset, but if the transmit enable bit (bit 4) of the serial I/O3 control register has been set to “1”, the transmit shift completion flag (bit 2) and the transmit buffer empty flag (bit 0) become “1”.

[Transmit Buffer Register 3/Receive Buffer Register 3 (TB3/RB3)] 0030$_{16}$

The transmit buffer register 3 and the receive buffer register 3 are located at the same address. The transmit buffer is write-only and the receive buffer is read-only. If a character bit length is 7 bits, the MSB of data stored in the receive buffer is “0”.

[Baud Rate Generator 3 (BRG3)] 002F$_{16}$

The baud rate generator determines the baud rate for serial transfer.

The baud rate generator divides the frequency of the count source by 1/(n + 1), where n is the value written to the baud rate generator.

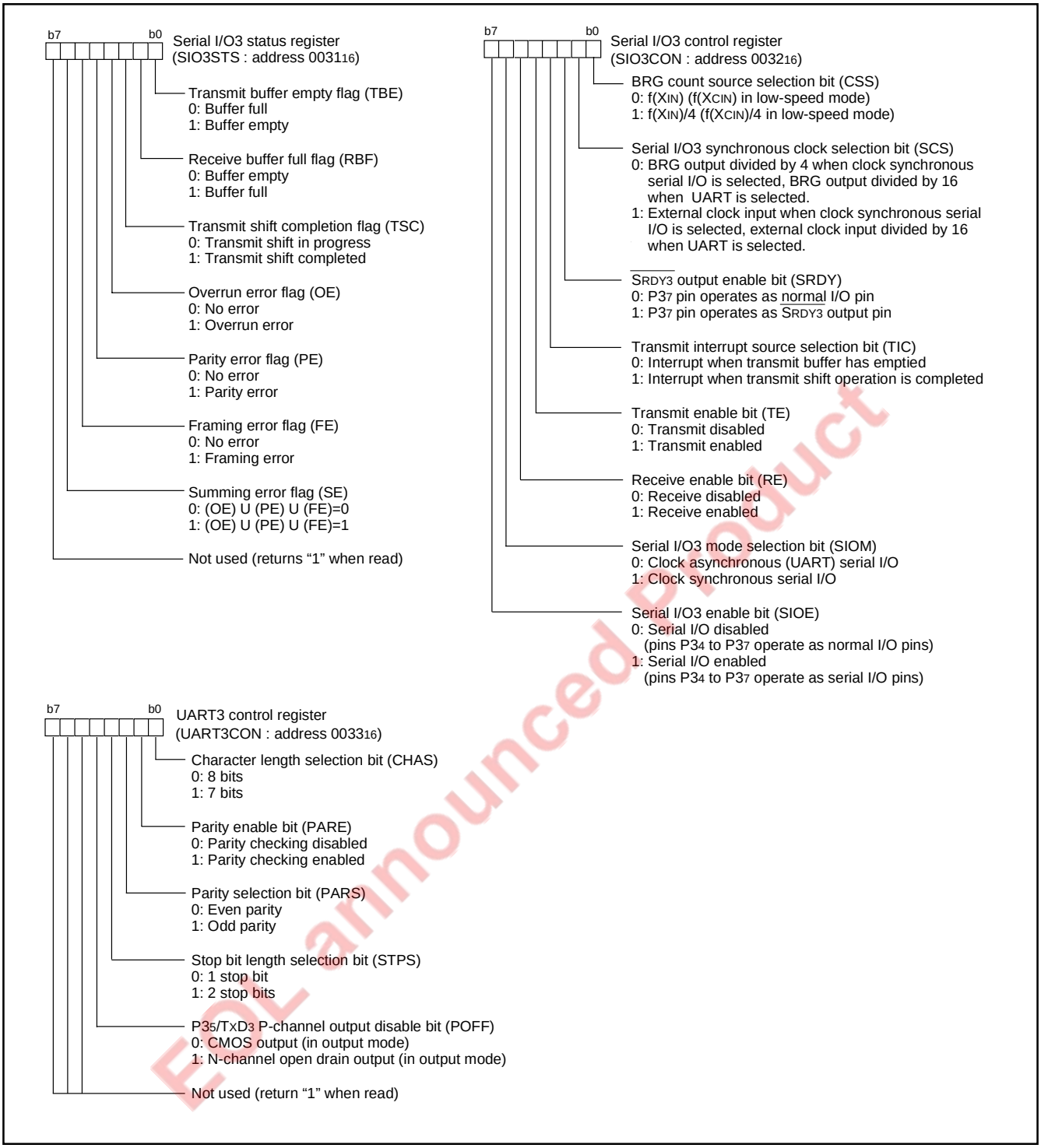


Fig. 44 Structure of serial I/O3 control registers

■ Notes concerning serial I/O3

1. Notes when selecting clock synchronous serial I/O

1.1 Stop of transmission operation

● **Note**

Clear the serial I/O3 enable bit and the transmit enable bit to “0” (serial I/O and transmit disabled).

● **Reason**

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O3 enable bit is cleared to “0” (serial I/O disabled), the internal transmission is running (in this case, since pins TxD3, RxD3, SCLK3, and $\overline{\text{SRDY3}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O enable bit is set to “1” at this time, the data during internally shifting is output to the TxD3 pin and an operation failure occurs.

1.2 Stop of receive operation

● **Note**

Clear the receive enable bit to “0” (receive disabled), or clear the serial I/O3 enable bit to “0” (serial I/O disabled).

1.3 Stop of transmit/receive operation

● **Note**

Clear both the transmit enable bit and receive enable bit to “0” (transmit and receive disabled).
(when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

● **Reason**

In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.
In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to “0” (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/O3 enable bit to “0” (serial I/O disabled) (refer to **1.1**).

2. Notes when selecting clock asynchronous serial I/O

2.1 Stop of transmission operation

● **Note**

Clear the transmit enable bit to “0” (transmit disabled). The transmission operation does not stop by clearing the serial I/O3 enable bit to “0”.

● **Reason**

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O3 enable bit is cleared to “0” (serial I/O disabled), the internal transmission is running (in this case, since pins TxD3, RxD3, SCLK3, and $\overline{\text{SRDY3}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O3 enable bit is set to “1” at this time, the data during internally shifting is output to the TxD3 pin and an operation failure occurs.

2.2 Stop of receive operation

● **Note**

Clear the receive enable bit to “0” (receive disabled).

2.3 Stop of transmit/receive operation

● **Note 1 (only transmission operation is stopped)**

Clear the transmit enable bit to “0” (transmit disabled). The transmission operation does not stop by clearing the serial I/O3 enable bit to “0”.

● **Reason**

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O3 enable bit is cleared to “0” (serial I/O disabled), the internal transmission is running (in this case, since pins TxD3, RxD3, SCLK3, and $\overline{\text{SRDY3}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O3 enable bit is set to “1” at this time, the data during internally shifting is output to the TxD3 pin and an operation failure occurs.

● **Note 2 (only receive operation is stopped)**

Clear the receive enable bit to “0” (receive disabled).

3. $\overline{\text{SRDY3}}$ output of reception side

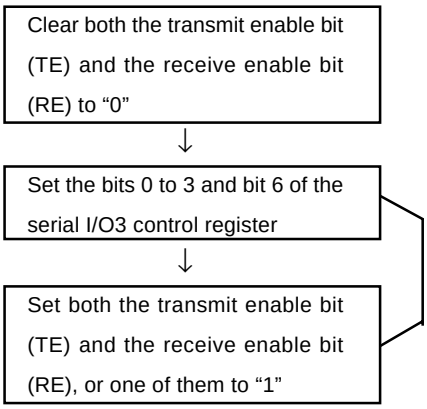
● **Note**

When signals are output from the $\overline{\text{SRDY3}}$ pin on the reception side by using an external clock in the clock synchronous serial I/O mode, set all of the receive enable bit, the $\overline{\text{SRDY3}}$ output enable bit, and the transmit enable bit to "1" (transmit enabled).

4. Setting serial I/O3 control register again

● **Note**

Set the serial I/O3 control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to "0."



5. Data transmission control with referring to transmit shift register completion flag

● **Note**

After the transmit data is written to the transmit buffer register, the transmit shift register completion flag changes from "1" to "0" with a delay of 0.5 to 1.5 shift clocks. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

6. Transmission control when external clock is selected

● **Note**

When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to "1" at "H" of the SCLK3 input level. Also, write data to the transmit buffer register at "H" of the SCLK input level.

7. Transmit interrupt request when transmit enable bit is set

● **Note**

When using the transmit interrupt, take the following sequence.

① Set the serial I/O3 transmit interrupt enable bit to "0" (disabled).
② Set the transmit enable bit to "1".
③ Set the serial I/O3 transmit interrupt request bit to "0" after 1 or more instruction has executed.
④ Set the serial I/O3 transmit interrupt enable bit to "1" (enabled).

● **Reason**

When the transmit enable bit is set to "1", the transmit buffer empty flag and the transmit shift register shift completion flag are also set to "1". Therefore, regardless of selecting which timing for the generating of transmit interrupts, the interrupt request is generated and the transmit interrupt request bit is set at this point.

PULSE WIDTH MODULATION (PWM)

The 3804 group (Spec. H) has PWM functions with an 8-bit resolution, based on a signal that is the clock input XIN or that clock input divided by 2 or the clock input XCIN or that clock input divided by 2 in low-speed mode.

Data Setting

The PWM output pin also functions as port P56. Set the PWM period by the PWM prescaler, and set the "H" term of output pulse by the PWM register.

If the value in the PWM prescaler is n and the value in the PWM register is m (where n = 0 to 255 and m = 0 to 255) :

PWM period = 255 × (n+1) / f(XIN)

= 31.875 × (n+1) µs (when f(XIN) = 8 MHz)

Output pulse "H" term = PWM period × m / 255

= 0.125 × (n+1) × m µs

(when f(XIN) = 8 MHz)

PWM Operation

When bit 0 (PWM enable bit) of the PWM control register is set to "1", operation starts by initializing the PWM output circuit, and pulses are output starting at an "H".

If the PWM register or PWM prescaler is updated during PWM output, the pulses will change in the cycle after the one in which the change was made.

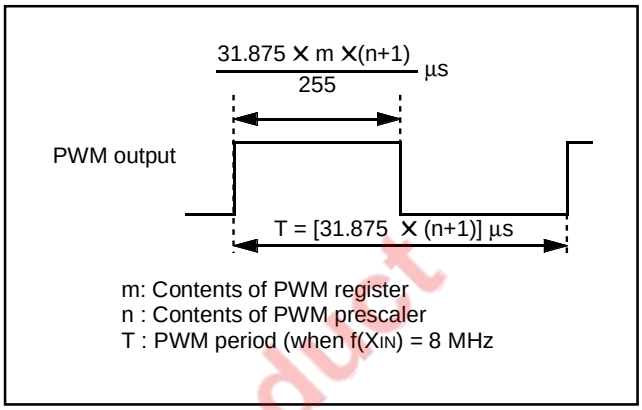


Fig. 45 Timing of PWM period

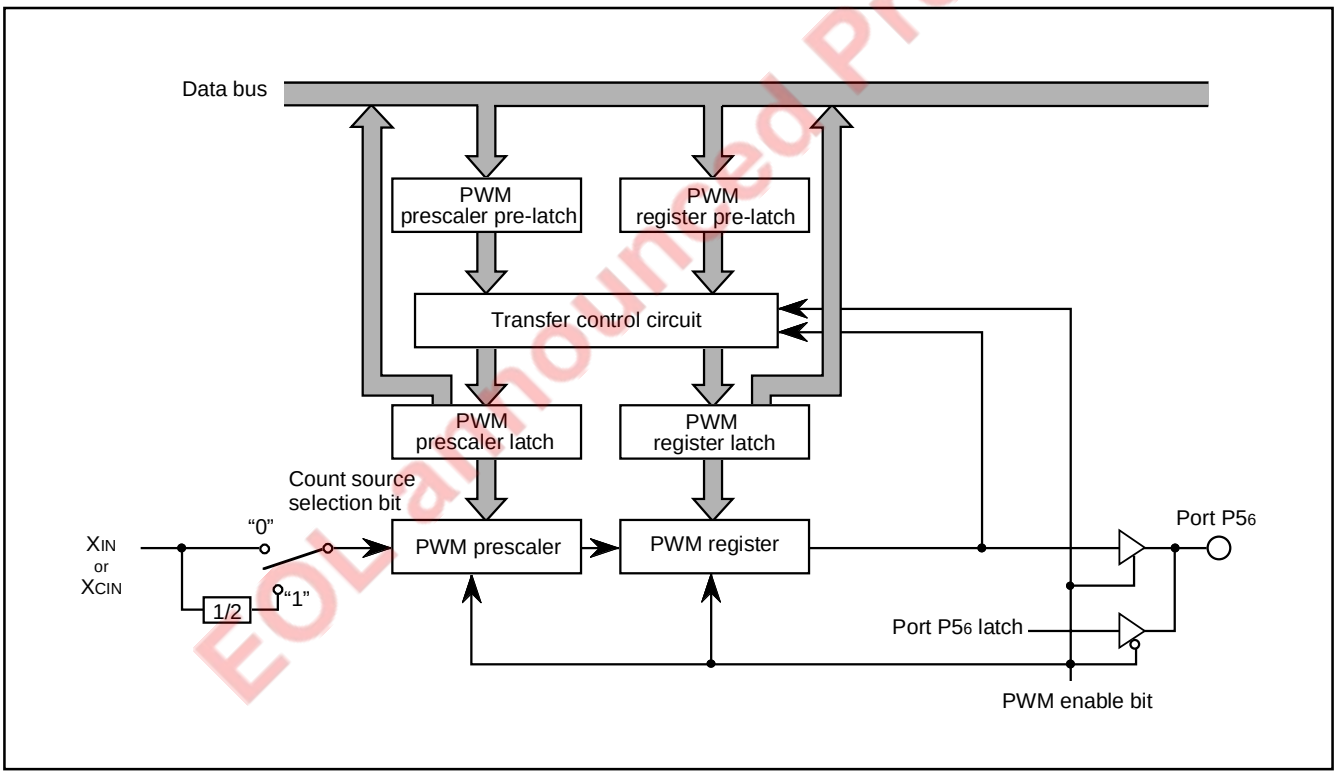


Fig. 46 Block diagram of PWM function

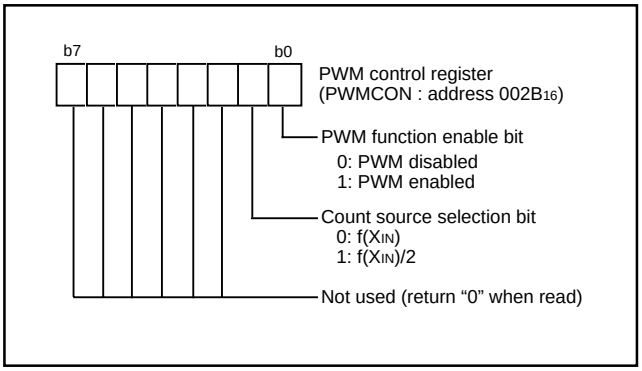


Fig. 47 Structure of PWM control register

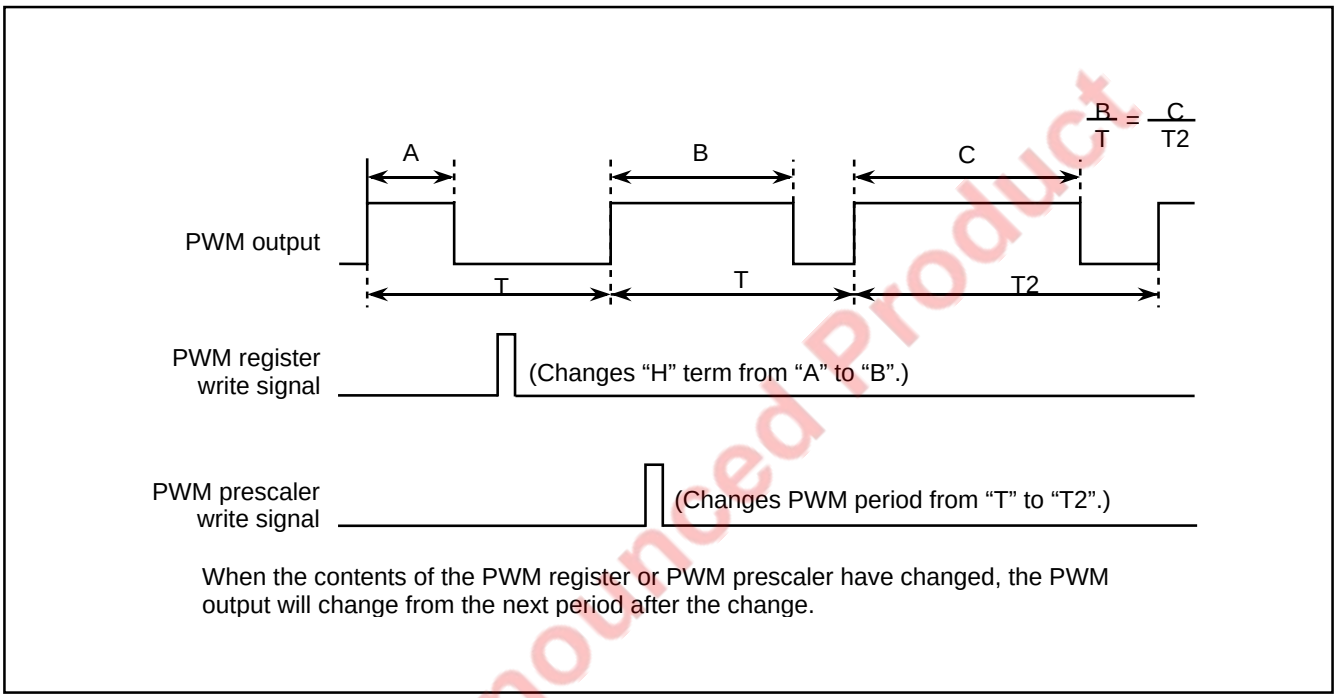


Fig. 48 PWM output timing when PWM register or PWM prescaler is changed

A/D CONVERTER

[AD Conversion Register 1, 2 (AD1, AD2)] 003516, 003816

The AD conversion register is a read-only register that stores the result of an A/D conversion. When reading this register during an A/D conversion, the previous conversion result is read.

Bit 7 of the AD conversion register 2 is the conversion mode selection bit. When this bit is set to "0," the A/D converter becomes the 10-bit A/D mode. When this bit is set to "1," that becomes the 8-bit A/D mode. The conversion result of the 8-bit A/D mode is stored in the AD conversion register 1. As for 10-bit A/D mode, not only 10-bit reading but also only high-order 8-bit reading of conversion result can be performed by selecting the reading procedure of the AD conversion registers 1, 2 after A/D conversion is completed (in Figure 50).

As for 10-bit A/D mode, the 8-bit reading inclined to MSB is performed when reading the AD converter register 1 after A/D conversion is started; and when the AD converter register 1 is read after reading the AD converter register 2, the 8-bit reading inclined to LSB is performed.

[AD/DA Control Register (ADCON)] 003416

The AD/DA control register controls the A/D conversion process. Bits 0 to 2 and bit 4 select a specific analog input pin. Bit 3 signals the completion of an A/D conversion. The value of this bit remains at "0" during an A/D conversion, and changes to "1" when an A/D conversion ends. Writing "0" to this bit starts the A/D conversion.

Comparison Voltage Generator

The comparison voltage generator divides the voltage between V_{REF} and AV_{SS} into 1024, and that outputs the comparison voltage in the 10-bit A/D mode (256 division in 8-bit A/D mode).

The A/D converter successively compares the comparison voltage V_{ref} in each mode, dividing the V_{REF} voltage (see below), with the input voltage.

- 10-bit A/D mode (10-bit reading)

$$V_{ref} = \frac{V_{REF}}{1024} \times n \quad (n = 0–1023)$$

- 10-bit A/D mode (8-bit reading)

$$V_{ref} = \frac{V_{REF}}{256} \times n \quad (n = 0–255)$$

- 8-bit A/D mode

$$V_{ref} = \frac{V_{REF}}{256} \times (n–0.5) \quad (n = 1–255)$$
$$= 0 \qquad (n = 0)$$

Channel Selector

The channel selector selects one of ports $P6_7/AN_7$ to $P6_0/AN_0$ or $P0_7/AN_{15}$ to $P0_0/AN_8$, and inputs the voltage to the comparator.

Comparator and Control Circuit

The comparator and control circuit compares an analog input voltage with the comparison voltage, and then stores the result in the AD conversion registers 1, 2. When an A/D conversion is completed, the control circuit sets the AD conversion completion bit and the AD interrupt request bit to "1".

Note that because the comparator consists of a capacitor coupling, set $f(X_{IN})$ to 500 kHz or more during an A/D conversion.

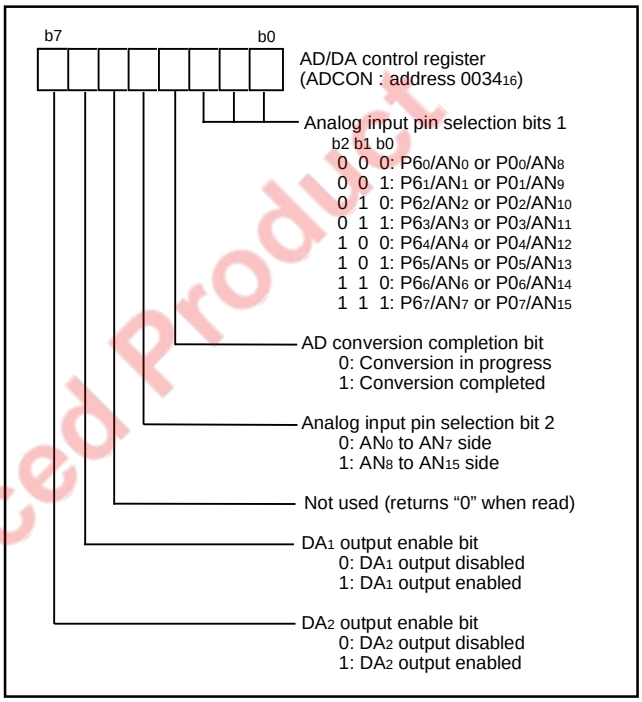


Fig. 49 Structure of AD/DA control register

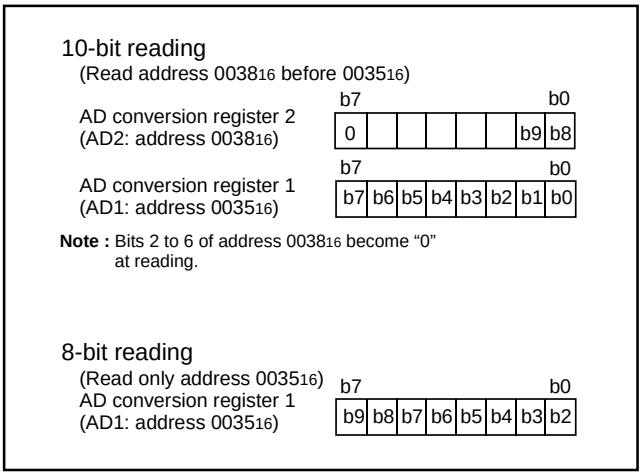


Fig. 50 Structure of 10-bit A/D mode reading

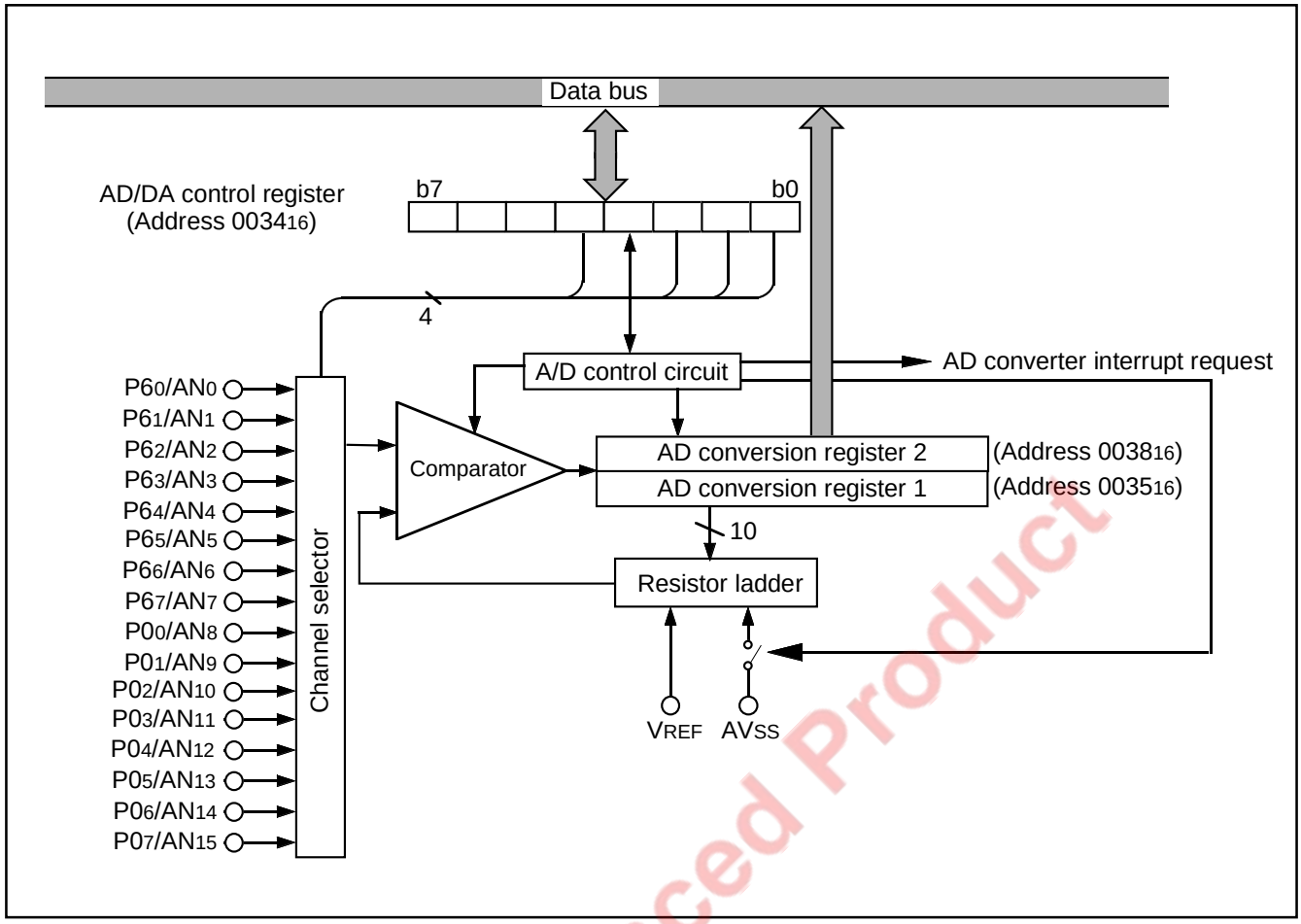


Fig. 51 Block diagram of A/D converter

D/A CONVERTER

The 3804 group (Spec. H) has two internal D/A converters (DA1 and DA2) with 8-bit resolution.
The D/A conversion is performed by setting the value in each DA conversion register. The result of D/A conversion is output from the DA_1 or DA_2 pin by setting the DA output enable bit to "1".
When using the D/A converter, the corresponding port direction register bit ($P3_0/DA_1$ or $P3_1/DA_2$) must be set to "0" (input status).
The output analog voltage V is determined by the value n (decimal notation) in the DA conversion register as follows:

$V = V_{REF} \times n/256$ (n = 0 to 255)
Where V_{REF} is the reference voltage.

At reset, the DA conversion registers are cleared to "00_{16}", and the DA output enable bits are cleared to "0", and the $P3_0/DA_1$ and $P3_1/DA_2$ pins become high impedance.
The DA output does not have buffers. Accordingly, connect an external buffer when driving a low-impedance load.

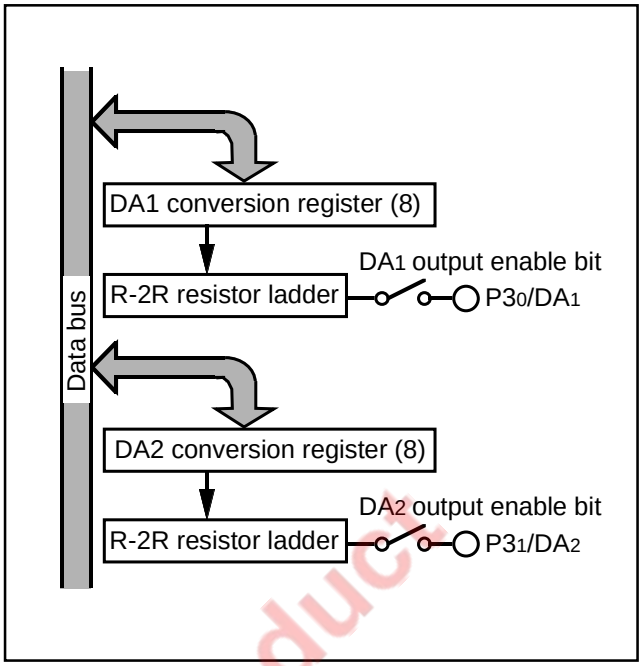


Fig. 52 Block diagram of D/A converter

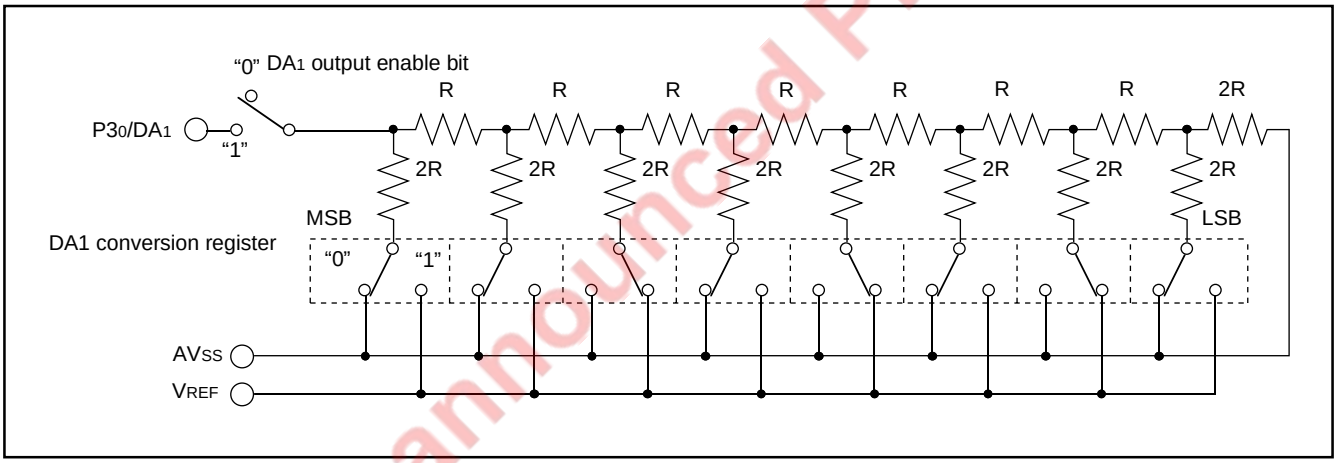


Fig. 53 Equivalent connection circuit of D/A converter (DA1)

WATCHDOG TIMER

The watchdog timer gives a mean of returning to the reset status when a program cannot run on a normal loop (for example, because of a software run-away). The watchdog timer consists of an 8-bit watchdog timer L and an 8-bit watchdog timer H.

Watchdog Timer Initial Value

Watchdog timer L is set to "FF16" and watchdog timer H is set to "FF16" by writing to the watchdog timer control register (address 001E16) or at a reset. Any write instruction that causes a write signal can be used, such as the STA, LDM, CLB, etc. Data can only be written to bits 6 and 7 of the watchdog timer control register. Regardless of the value written to bits 0 to 5, the above-mentioned value will be set to each timer.

Watchdog Timer Operations

The watchdog timer stops at reset and a countdown is started by the writing to the watchdog timer control register. An internal reset occurs when watchdog timer H underflows. The reset is released after its release time. After the release, the program is restarted from the reset vector address. Usually, write to the watchdog timer control register by software before an underflow of the watchdog timer H. The watchdog timer does not function if the watchdog timer control register is not written to at least once.

When bit 6 of the watchdog timer control register is kept at "0", the STP instruction is enabled. When that is executed, both the clock and the watchdog timer stop. Count re-starts at the same time as the release of stop mode **(Note)**. The watchdog timer does not stop while a WIT instruction is executed. In addition, the STP instruction is disabled by writing "1" to this bit again. When the STP instruction is executed at this time, it is processed as an undefined instruction, and an internal reset occurs. Once a "1" is written to this bit, it cannot be programmed to "0" again.
The following shows the period between the write execution to the watchdog timer control register and the underflow of watchdog timer H.

Bit 7 of the watchdog timer control register is "0":
when XCIN = 32.768 kHz; 32 s
when XIN = 16 MHz; 65.536 ms

Bit 7 of the watchdog timer control register is "1":
when XCIN = 32.768 kHz; 125 ms
when XIN = 16 MHz; 256 µs

Note: The watchdog timer continues to count even while waiting for a stop release. Therefore, make sure that watchdog timer H does not underflow during this period.

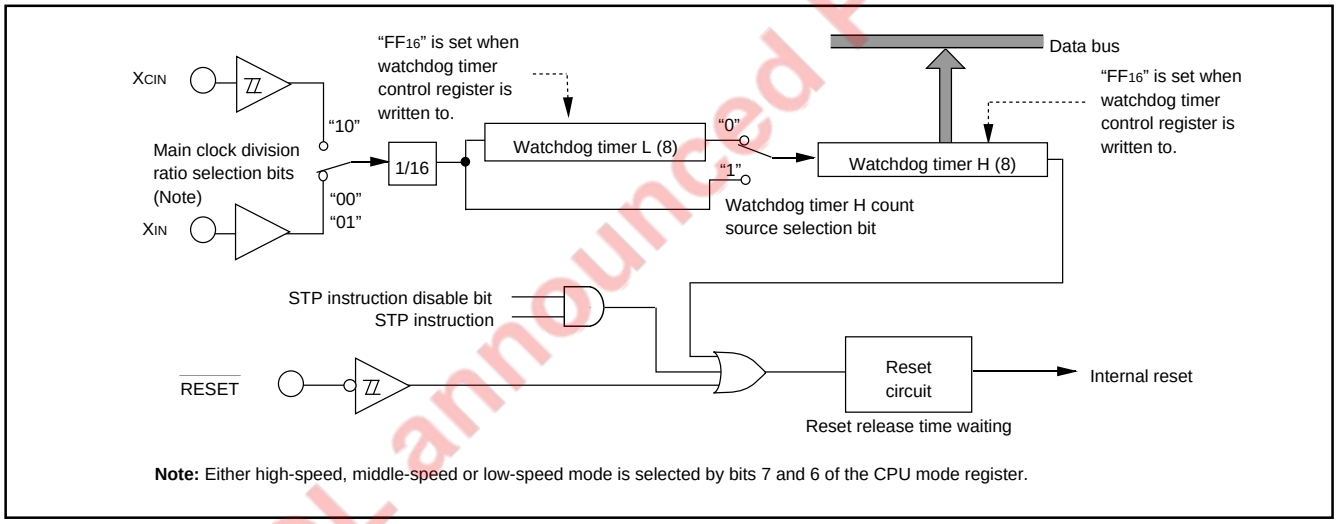


Fig. 54 Block diagram of Watchdog timer

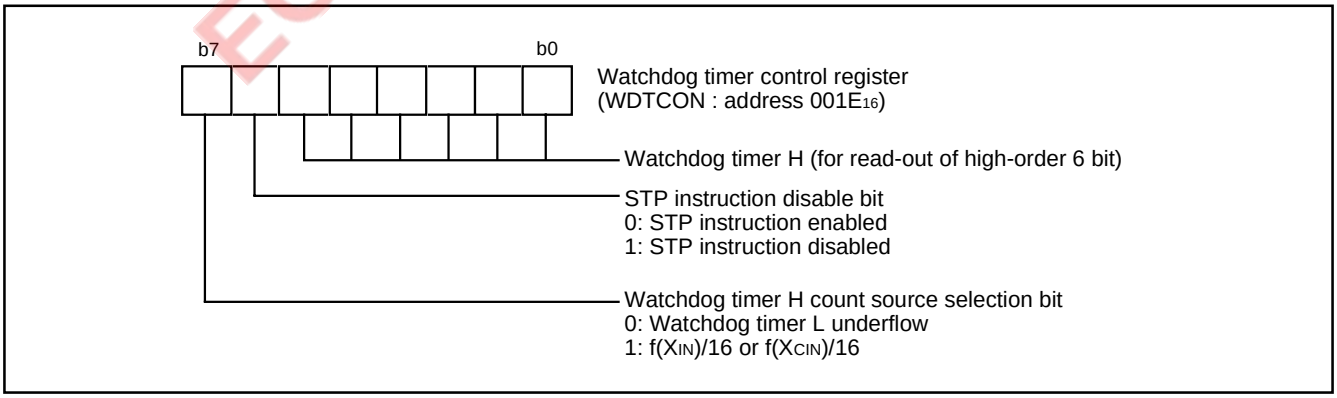


Fig. 55 Structure of Watchdog timer control register

MULTI-MASTER I²C-BUS INTERFACE

The 3804 group (Spec. H) has the multi-master I²C-BUS interface. The multi-master I²C-BUS interface is a serial communications circuit, conforming to the Philips I²C-BUS data transfer format. This interface, offering both arbitration lost detection and a synchronous functions, is useful for the multi-master serial communications.

Figure 56 shows a block diagram of the multi-master I²C-BUS interface and Table 7 lists the multi-master I²C-BUS interface functions.

This multi-master I²C-BUS interface consists of the I²C slave address registers 0 to 2, the I²C data shift register, the I²C clock control register, the I²C control register, the I²C status register, the I²C START/STOP condition control register, the I²C special mode control register, the I²C special mode status register, and other control circuits.

When using the multi-master I²C-BUS interface, set 1 MHz or more to the internal clock ϕ.

Table 7 Multi-master I²C-BUS interface functions

Item	Function
Format	In conformity with Philips I²C-BUS standard: 10-bit addressing format 7-bit addressing format High-speed clock mode Standard clock mode
Communication mode	In conformity with Philips I²C-BUS standard: Master transmission Master reception Slave transmission Slave reception
SCL clock frequency	16.1 kHz to 400 kHz (at ϕ= 4 MHz)

System clock ϕ = f(XIN)/2 (high-speed mode)
ϕ = f(XIN)/8 (middle-speed mode)

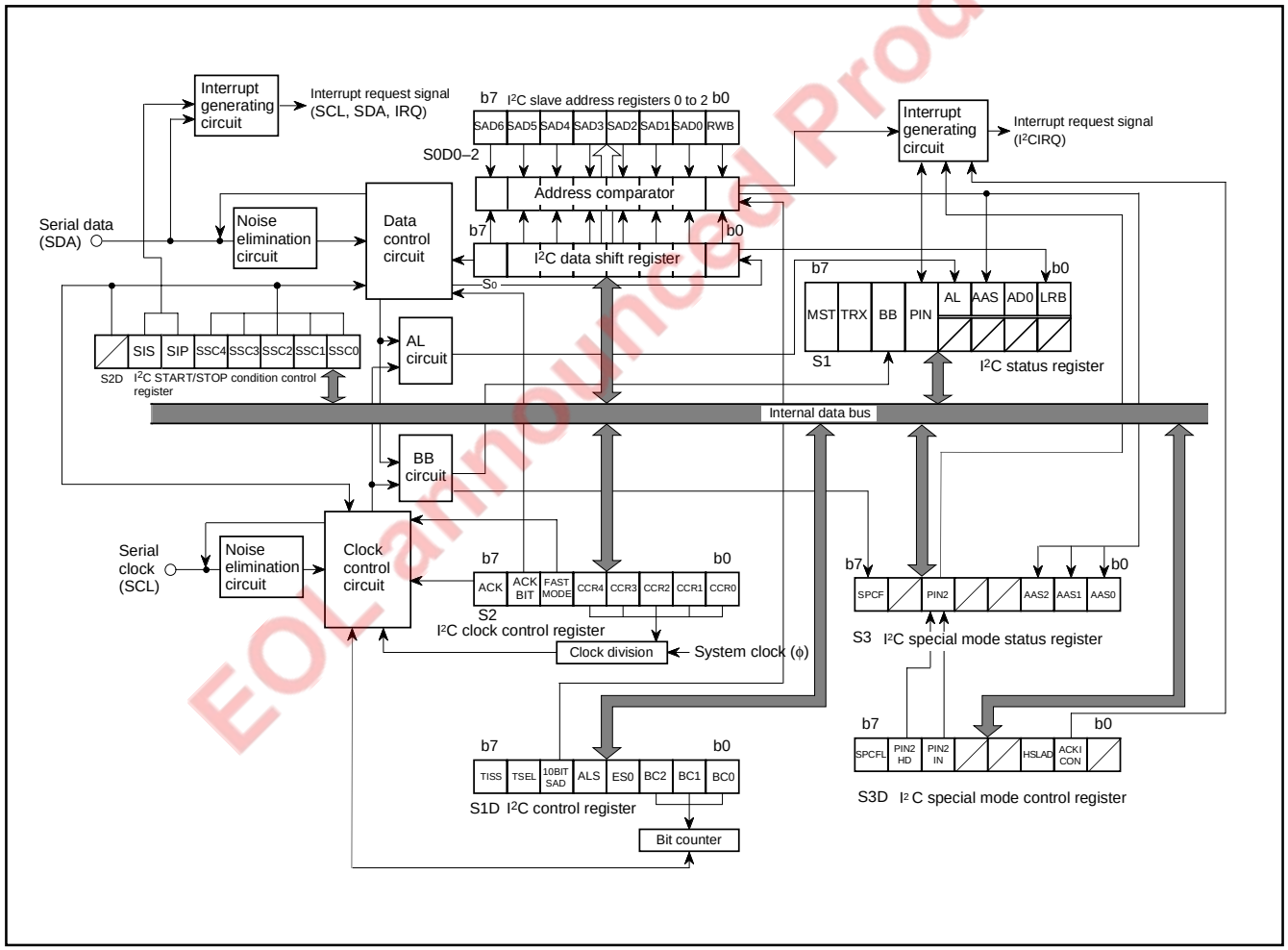


Fig. 56 Block diagram of multi-master I²C-BUS interface

✻ : Purchase of MITSUBISHI ELECTRIC CORPORATIONS I²C components conveys a license under the Philips I²C Patent Rights to use these components an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

[I²C Data Shift Register (S0)] 0011₁₆

The I²C data shift register (S0: address 0011₁₆) is an 8-bit shift register to store receive data and write transmit data.

When transmit data is written into this register, it is transferred to the outside from bit 7 in synchronization with the SCL, and each time one-bit data is output, the data of this register are shifted by one bit to the left. When data is received, it is input to this register from bit 0 in synchronization with the SCL, and each time one-bit data is input, the data of this register are shifted by one bit to the left. The minimum 2 cycles of the internal clock ϕ are required from the rising of the SCL until input to this register.

The I²C data shift register is in a write enable status only when the I²C-BUS interface enable bit (ES0 bit) of the I²C control register (S1D: address 0014₁₆) is "1". The bit counter is reset by a write instruction to the I²C data shift register. When both the ES0 bit and the MST bit of the I²C status register (S1: address 0013₁₆) are "1," the SCL is output by a write instruction to the I²C data shift register. Reading data from the I²C data shift register is always enabled regardless of the ES0 bit value.

[I²C Slave Address Registers 0 to 2 (S0D0 to S0D2)] 0FF7₁₆ to 0FF9₁₆

The I²C slave address registers 0 to 2 (S0D0 to S0D2: addresses 0FF7₁₆ to 0FF9₁₆) consists of a 7-bit slave address and a read/$\overline{\text{write}}$ bit. In the addressing mode, the slave address written in this register is compared with the address data to be received immediately after the START condition is detected.

•Bit 0: Read/$\overline{\text{write}}$ bit (RWB)

This is not used in the 7-bit addressing mode. In the 10-bit addressing mode, set RWB to "0" because the first address data to be received is compared with the contents (SAD6 to SAD0 + RWB) of the I²C slave address registers 0 to 2.

When 2-byte address data match slave address, a 7-bit slave address which is received after restart condition has detected and R/$\overline{\text{W}}$ data can be matched by setting "1" to RWB with software. The RWB is cleared to "0" automatically when the stop condition is detected.

•Bits 1 to 7: Slave address (SAD0–SAD6)

These bits store slave addresses. Regardless of the 7-bit addressing mode or the 10-bit addressing mode, the address data transmitted from the master is compared with these bits' contents.

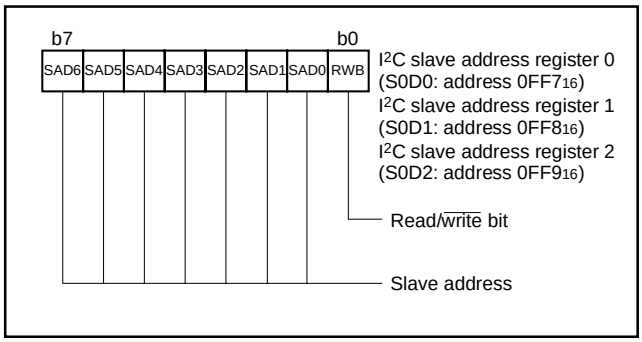


Fig. 57 Structure of I²C slave address registers 0 to 2

[I²C Clock Control Register (S2)] 0015₁₆

The I²C clock control register (S2: address 0015₁₆) is used to set ACK control, SCL mode and SCL frequency.

•Bits 0 to 4: SCL frequency control bits (CCR0–CCR4)

These bits control the SCL frequency. Refer to Table 8.

•Bit 5: SCL mode specification bit (FAST MODE)

This bit specifies the SCL mode. When this bit is set to "0," the standard clock mode is selected. When the bit is set to "1," the high-speed clock mode is selected.

When connecting the bus of the high-speed mode I²C bus standard (maximum 400 kbits/s), use 8 MHz or more oscillation frequency f(XIN) in the high-speed mode (2 division clock).

•Bit 6: ACK bit (ACK BIT)

This bit sets the SDA status when an ACK clock* is generated. When this bit is set to "0," the ACK return mode is selected and SDA goes to "L" at the occurrence of an ACK clock. When the bit is set to "1," the ACK non-return mode is selected. The SDA is held in the "H" status at the occurrence of an ACK clock.

However, when the slave address agree with the address data in the reception of address data at ACK BIT = "0," the SDA is automatically made "L" (ACK is returned). If there is a disagreement between the slave address and the address data, the SDA is automatically made "H" (ACK is not returned).

*ACK clock: Clock for acknowledgment

•Bit 7: ACK clock bit (ACK)

This bit specifies the mode of acknowledgment which is an acknowledgment response of data transfer. When this bit is set to "0," the no ACK clock mode is selected. In this case, no ACK clock occurs after data transmission. When the bit is set to "1," the ACK clock mode is selected and the master generates an ACK clock each completion of each 1-byte data transfer. The device for transmitting address data and control data releases the SDA at the occurrence of an ACK clock (makes SDA "H") and receives the ACK bit generated by the data receiving device.

Note: Do not write data into the I²C clock control register during transfer. If data is written during transfer, the I²C clock generator is reset, so that data cannot be transferred normally.

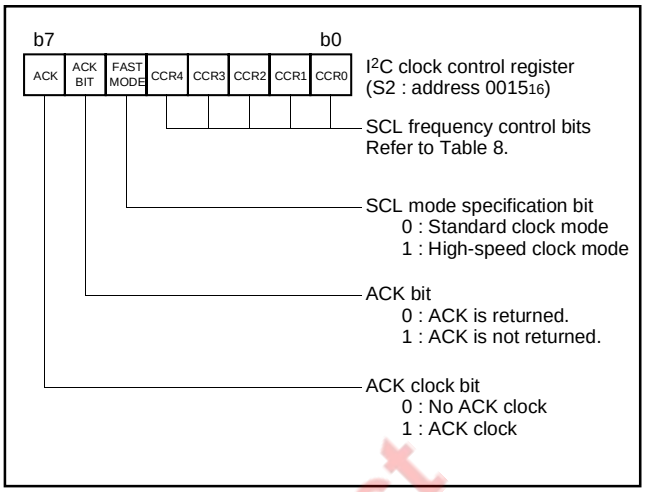


Fig. 58 Structure of I²C clock control register

Table 8 Set values of I²C clock control register and SCL frequency

Setting value of CCR4–CCR0					SCL frequency (at ϕ = 4 MHz, unit : kHz) **(Note 1)**	
CCR4	CCR3	CCR2	CCR1	CCR0	Standard clock mode	High-speed clock mode
0	0	0	0	0	Setting disabled	Setting disabled
0	0	0	0	1	Setting disabled	Setting disabled
0	0	0	1	0	Setting disabled	Setting disabled
0	0	0	1	1	– **(Note 2)**	333
0	0	1	0	0	– **(Note 2)**	250
0	0	1	0	1	100	400 **(Note 3)**
0	0	1	1	0	83.3	166
⋮	⋮	⋮	⋮	⋮	500/CCR value **(Note 3)**	1000/CCR value **(Note 3)**
1	1	1	0	1	17.2	34.5
1	1	1	1	0	16.6	33.3
1	1	1	1	1	16.1	32.3

Notes 1: Duty of SCL output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at ϕ = 4 MHz). "H" duration of the clock fluctuates from –4 to +2 machine cycles in the standard clock mode, and fluctuates from –2 to +2 machine cycles in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because "L" duration is extended instead of "H" duration reduction.
These are values when SCL synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

2: Each value of SCL frequency exceeds the limit at ϕ = 4 MHz or more. When using these setting value, use ϕ of 4 MHz or less.

3: The data formula of SCL frequency is described below:
ϕ/(8 × CCR value) Standard clock mode
ϕ/(4 × CCR value) High-speed clock mode (CCR value ≠ 5)
ϕ/(2 × CCR value) High-speed clock mode (CCR value = 5)
Do not set 0 to 2 as CCR value regardless of ϕ frequency.
Set 100 kHz (max.) in the standard clock mode and 400 kHz (max.) in the high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

[I²C Control Register (S1D)] 0014₁₆

The I²C control register (S1D: address 0014₁₆) controls data communication format.

•Bits 0 to 2: Bit counter (BC0–BC2)

These bits decide the number of bits for the next 1-byte data to be transmitted. The I²C interrupt request signal occurs immediately after the number of count specified with these bits (ACK clock is added to the number of count when ACK clock is selected by ACK clock bit (bit 7 of S2, address 0015₁₆) have been transferred, and BC0 to BC2 are returned to "000₂".

Also when a START condition is received, these bits become "000₂" and the address data is always transmitted and received in 8 bits.

•Bit 3: I²C interface enable bit (ES0)

This bit enables to use the multi-master I²C-BUS interface. When this bit is set to "0," the use disable status is provided, so that the SDA and the SCL become high-impedance. When the bit is set to "1," use of the interface is enabled.

When ES0 = "0," the following is performed.

- PIN = "1," BB = "0" and AL = "0" are set (which are bits of the I²C status register, S1, at address 0013₁₆).
- Writing data to the I²C data shift register (S0: address 0011₁₆) is disabled.

•Bit 4: Data format selection bit (ALS)

This bit decides whether or not to recognize slave addresses. When this bit is set to "0," the addressing format is selected, so that address data is recognized. When a match is found between a slave address and address data as a result of comparison or when a general call (refer to "I²C Status Register," bit 1) is received, transfer processing can be performed. When this bit is set to "1," the free data format is selected, so that slave addresses are not recognized.

•Bit 5: Addressing format selection bit (10BIT SAD)

This bit selects a slave address specification format. When this bit is set to "0," the 7-bit addressing format is selected. In this case, only the high-order 7 bits (slave address) of the I²C slave address registers 0 to 2 are compared with address data. When this bit is set to "1," the 10-bit addressing format is selected, and all the bits of the I²C slave address registers 0 to 2 are compared with address data.

•Bit 7: I²C-BUS interface pin input level selection bit (TISS)

This bit selects the input level of the SCL and SDA pins of the multi-master I²C-BUS interface.

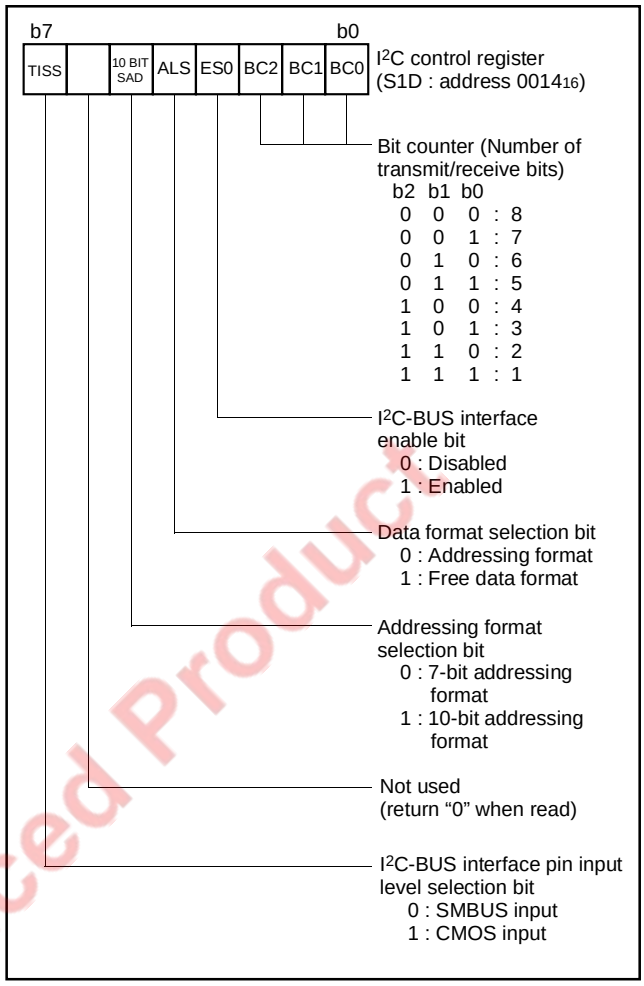


Fig. 59 Structure of I²C control register

[I²C Status Register (S1)] 0013₁₆

The I²C status register (S1: address 0013₁₆) controls the I²C-BUS interface status. The low-order 4 bits are read-only bits and the high-order 4 bits can be read out and written to.
Set "0000₂" to the low-order 4 bits, because these bits become the reserved bits at writing.

•Bit 0: Last receive bit (LRB)

This bit stores the last bit value of received data and can also be used for ACK receive confirmation. If ACK is returned when an ACK clock occurs, the LRB bit is set to "0." If ACK is not returned, this bit is set to "1." Except in the ACK mode, the last bit value of received data is input. The state of this bit is changed from "1" to "0" by executing a write instruction to the I²C data shift register (S0: address 0011₁₆).

•Bit 1: General call detecting flag (AD0)

When the ALS bit is "0", this bit is set to "1" when a general call* whose address data is all "0" is received in the slave mode. By a general call of the master device, every slave device receives control data after the general call. The AD0 bit is set to "0" by detecting the STOP condition or START condition, or reset.

*General call: The master transmits the general call address "00₁₆" to all slaves.

•Bit 2: Slave address comparison flag (AAS)

This flag indicates a comparison result of address data when the ALS bit is "0".

① In the slave receive mode, when the 7-bit addressing format is selected, this bit is set to "1" in one of the following conditions:
- The address data immediately after occurrence of a START condition agrees with the slave address stored in the high-order 7 bits of the I²C slave address register.
- A general call is received.

② In the slave receive mode, when the 10-bit addressing format is selected, this bit is set to "1" with the following condition:
- When the address data is compared with the I²C slave address register (8 bits consisting of slave address and RWB bit), the first bytes agree.

③ This bit is set to "0" by executing a write instruction to the I²C data shift register (S0: address 0011₁₆) when ES0 is set to "1" or reset.

•Bit 3: Arbitration lost* detecting flag (AL)

In the master transmission mode, when the SDA is made "L" by any other device, arbitration is judged to have been lost, so that this bit is set to "1." At the same time, the TRX bit is set to "0," so that immediately after transmission of the byte whose arbitration was lost is completed, the MST bit is set to "0." The arbitration lost can be detected only in the master transmission mode. When arbitration is lost during slave address transmission, the TRX bit is set to "0" and the reception mode is set. Consequently, it becomes possible to detect the agreement of its own slave address and address data transmitted by another master device.

The AL bit is set to "0" in one of the following conditions:
- Executing a write instruction to the I²C data shift register (S0: address 0011₁₆)
- When the ES0 bit is "0"
- At reset

*Arbitration lost : The status in which communication as a master is disabled.

•Bit 4: SCL pin low hold bit (PIN)

This bit generates an interrupt request signal. Each time 1-byte data is transmitted, the PIN bit changes from "1" to "0." At the same time, an interrupt request signal occurs to the CPU. The PIN bit is set to "0" in synchronization with a falling of the last clock (including the ACK clock) of an internal clock and an interrupt request signal occurs in synchronization with a falling of the PIN bit. When the PIN bit is "0," the SCL is kept in the "0" state and clock generation is disabled. Figure 61 shows an interrupt request signal generating timing chart.
The PIN bit is set to "1" in one of the following conditions:
- Executing a write instruction to the I²C data shift register (S0: address 0011₁₆). (This is the only condition which the prohibition of the internal clock is released and data can be communicated except for the start condition detection.)
- When the ES0 bit is "0"
- At reset
- When writing "1" to the PIN bit by software

The PIN bit is set to "0" in one of the following conditions:
- Immediately after completion of 1-byte data transmission (including when arbitration lost is detected)
- Immediately after completion of 1-byte data reception
- In the slave reception mode, with ALS = "0" and immediately after completion of slave address agreement or general call address reception
- In the slave reception mode, with ALS = "1" and immediately after completion of address data reception

•Bit 5: Bus busy flag (BB)

This bit indicates the status of use of the bus system. When this bit is set to "0," this bus system is not busy and a START condition can be generated. The BB flag is set/reset by the SCL, SDA pins input signal regardless of master/slave. This flag is set to "1" by detecting the START condition, and is set to "0" by detecting the STOP condition. The condition of these detecting is set by the START/STOP condition setting bits (SSC4–SSC0) of the I²C START/STOP condition control register (S2D: address 0016₁₆). When the ES0 bit of the I²C control register (bit 3 of S1D, address 0014₁₆) is "0" or reset, the BB flag is set to "0."
For the writing function to the BB flag, refer to the sections "START Condition Generating Method" and "STOP Condition Generating Method" described later.

•Bit 6: Communication mode specification bit (transfer direction specification bit: TRX)

This bit decides a direction of transfer for data communication. When this bit is “0,” the reception mode is selected and the data of a transmitting device is received. When the bit is “1,” the transmission mode is selected and address data and control data are output onto the SDA in synchronization with the clock generated on the SCL.

This bit is set/reset by software and hardware. About set/reset by hardware is described below. This bit is set to “1” by hardware when all the following conditions are satisfied:

- When ALS is “0”
- In the slave reception mode or the slave transmission mode
- When the R/$\overline{W}$ bit reception is “1”

This bit is set to “0” in one of the following conditions:

- When arbitration lost is detected.
- When a STOP condition is detected.
- When writing “1” to this bit by software is invalid by the START condition duplication preventing function **(Note)**.
- With MST = “0” and when a START condition is detected.
- With MST = “0” and when ACK non-return is detected.
- At reset

•Bit 7: Communication mode specification bit (master/slave specification bit: MST)

This bit is used for master/slave specification for data communication. When this bit is “0,” the slave is specified, so that a START condition and a STOP condition generated by the master are received, and data communication is performed in synchronization with the clock generated by the master. When this bit is “1,” the master is specified and a START condition and a STOP condition are generated. Additionally, the clocks required for data communication are generated on the SCL.

This bit is set to “0” in one of the following conditions.

- Immediately after completion of the byte which has lost arbitration when arbitration lost is detected
- When a STOP condition is detected.
- Writing “1” to this bit by software is invalid by the START condition duplication preventing function **(Note)**.
- At reset

Note: START condition duplication preventing function
The MST, TRX, and BB bits is set to “1” at the same time after confirming that the BB flag is “0” in the procedure of a START condition occurrence. However, when a START condition by another master device occurs and the BB flag is set to “1” immediately after the contents of the BB flag is confirmed, the START condition duplication preventing function makes the writing to the MST and TRX bits invalid. The duplication preventing function becomes valid from the rising of the BB flag to reception completion of slave address.

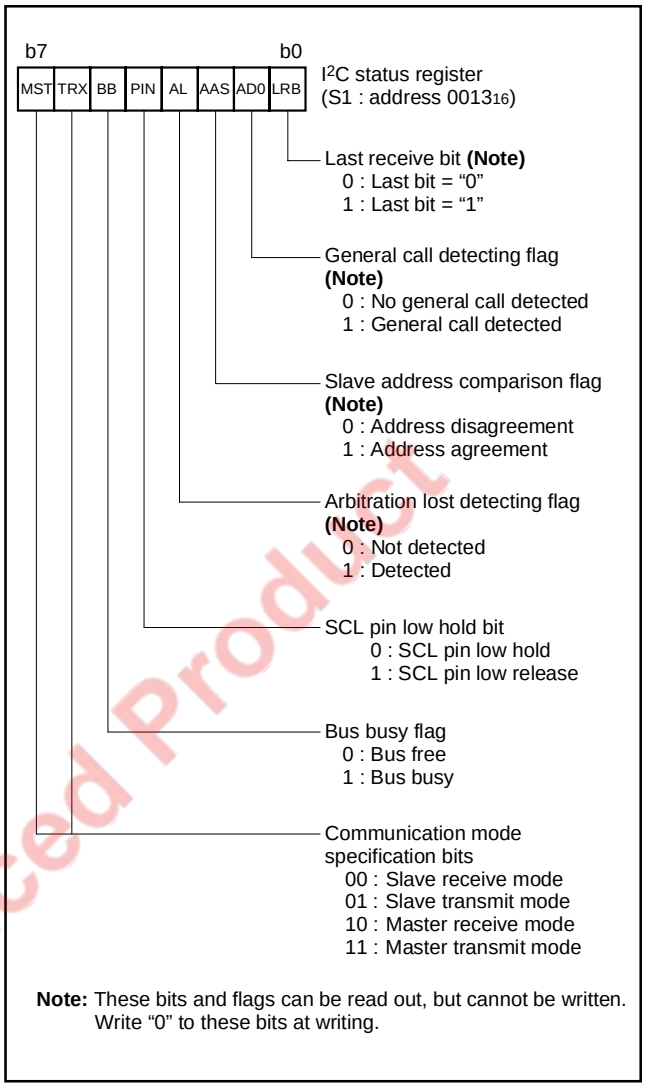


Fig. 60 Structure of I2C status register

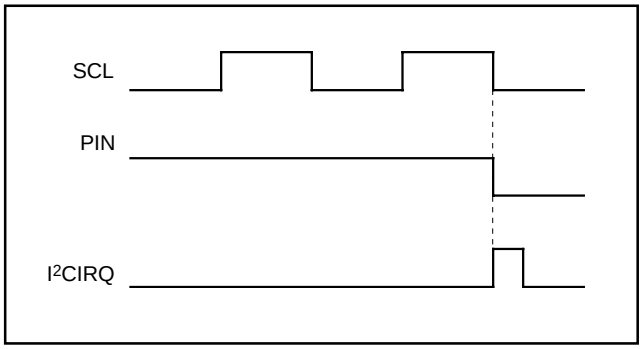


Fig. 61 Interrupt request signal generating timing

START Condition Generating Method

When writing "1" to the MST, TRX, and BB bits of the I²C status register (S1: address 0013₁₆) at the same time after writing the slave address to the I²C data shift register (S0: address 0011₁₆) with the condition in which the ES0 bit of the I²C control register (S1D: address 0014₁₆) is "1" and the BB flag is "0", a START condition occurs. After that, the bit counter becomes "000₂" and an SCL for 1 byte is output. The START condition generating timing is different in the standard clock mode and the high-speed clock mode. Refer to Figure 62, the START condition generating timing diagram, and Table 9, the START condition generating timing table.

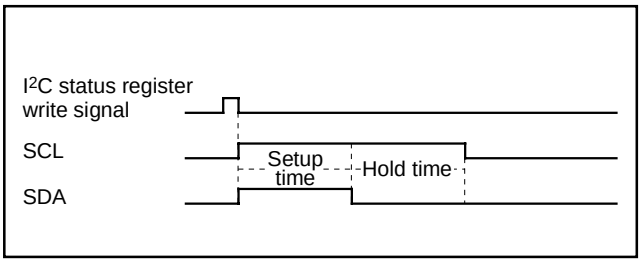


Fig. 62 START condition generating timing diagram

Table 9 START condition generating timing table

Item	Standard clock mode	High-speed clock mode
Setup time	5.0 µs (20 cycles)	2.5 µs (10 cycles)
Hold time	5.0 µs (20 cycles)	2.5 µs (10 cycles)

Note: Absolute time at ϕ = 4 MHz. The value in parentheses denotes the number of ϕ cycles.

STOP Condition Generating Method

When the ES0 bit of the I²C control register (S1D: address 0014₁₆) is "1," write "1" to the MST and TRX bits, and write "0" to the BB bit of the I²C status register (S1: address 0013₁₆) simultaneously. Then a STOP condition occurs. The STOP condition generating timing is different in the standard clock mode and the high-speed clock mode. Refer to Figure 63, the STOP condition generating timing diagram, and Table 10, the STOP condition generating timing table.

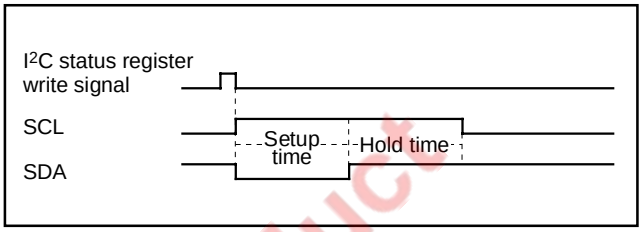


Fig. 63 STOP condition generating timing diagram

Table 10 STOP condition generating timing table

Item	Standard clock mode	High-speed clock mode
Setup time	5.0 µs (20 cycles)	3.0 µs (12 cycles)
Hold time	4.5 µs (18 cycles)	2.5 µs (10 cycles)

Note: Absolute time at ϕ = 4 MHz. The value in parentheses denotes the number of ϕ cycles.

START/STOP Condition Detecting Operation

The START/STOP condition detection operations are shown in Figures 64, 65, and Table 11. The START/STOP condition is set by the START/STOP condition set bit.

The START/STOP condition can be detected only when the input signal of the SCL and SDA pins satisfy three conditions: SCL release time, setup time, and hold time (see Table 11).

The BB flag is set to "1" by detecting the START condition and is reset to "0" by detecting the STOP condition.

The BB flag set/reset timing is different in the standard clock mode and the high-speed clock mode. Refer to Table 11, the BB flag set/reset time.

Note: When a STOP condition is detected in the slave mode (MST = 0), an interrupt request signal "I^2CIRQ" occurs to the CPU.

Table 11 START condition/STOP condition detecting conditions

	Standard clock mode	High-speed clock mode
SCL release time	SSC value + 1 cycle (6.25 µs)	4 cycles (1.0 µs)
Setup time	$\frac{\text{SSC value} + 1}{2}$ cycle < 4.0 µs (3.125 µs)	2 cycles (0.5 µs)
Hold time	$\frac{\text{SSC value} + 1}{2}$ cycle < 4.0 µs (3.125 µs)	2 cycles (0.5 µs)
BB flag set/ reset time	$\frac{\text{SSC value} - 1}{2}$ + 2 cycles (3.375 µs)	3.5 cycles (0.875 µs)

Note: Unit : Cycle number of internal clock ϕ
SSC value is the decimal notation value of the START/STOP condition set bits SSC4 to SSC0. Do not set "0" or an odd number to SSC value. The value in parentheses is an example when the I^2C START/STOP condition control register is set to "18_{16}" at ϕ = 4 MHz.

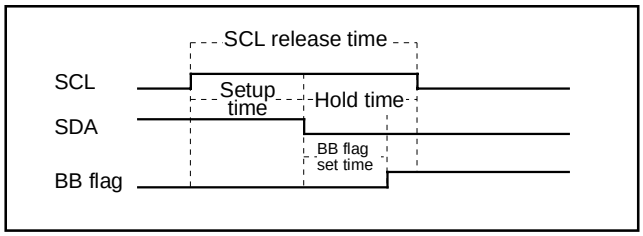


Fig. 64 START/STOP condition detecting timing diagram

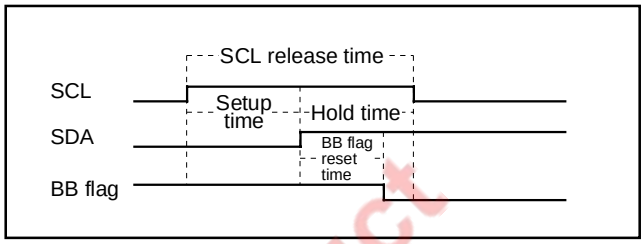


Fig. 65 STOP condition detecting timing diagram

[I²C START/STOP Condition Control Register (S2D)] 0016₁₆

The I²C START/STOP condition control register (S2D: address 0016₁₆) controls START/STOP condition detection.

•Bits 0 to 4: START/STOP condition set bits (SSC4–SSC0)

SCL release time, setup time, and hold time change the detection condition by value of the main clock divide ratio selection bit and the oscillation frequency f(XIN) because these time are measured by the internal system clock. Accordingly, set the proper value to the START/STOP condition set bits (SSC4 to SSC0) in considered of the system clock frequency. Refer to Table 11.

Do not set "00000₂" or an odd number to the START/STOP condition set bits (SSC4 to SSC0).

Refer to Table 12, the recommended set value to START/STOP condition set bits (SSC4–SSC0) for each oscillation frequency.

•Bit 5: SCL/SDA interrupt pin polarity selection bit (SIP)

An interrupt can occur when detecting the falling or rising edge of the SCL or SDA pin. This bit selects the polarity of the SCL or SDA pin interrupt pin.

•Bit 6: SCL/SDA interrupt pin selection bit (SIS)

This bit selects the pin of which interrupt becomes valid between the SCL pin and the SDA pin.

Note: When changing the setting of the SCL/SDA interrupt pin polarity selection bit, the SCL/SDA interrupt pin selection bit, or the I²C-BUS interface enable bit ES0, the SCL/SDA interrupt request bit may be set. When selecting the SCL/SDA interrupt source, disable the interrupt before the SCL/SDA interrupt pin polarity selection bit, the SCL/SDA interrupt pin selection bit, or the I²C-BUS interface enable bit ES0 is set. Reset the request bit to "0" after setting these bits, and enable the interrupt.

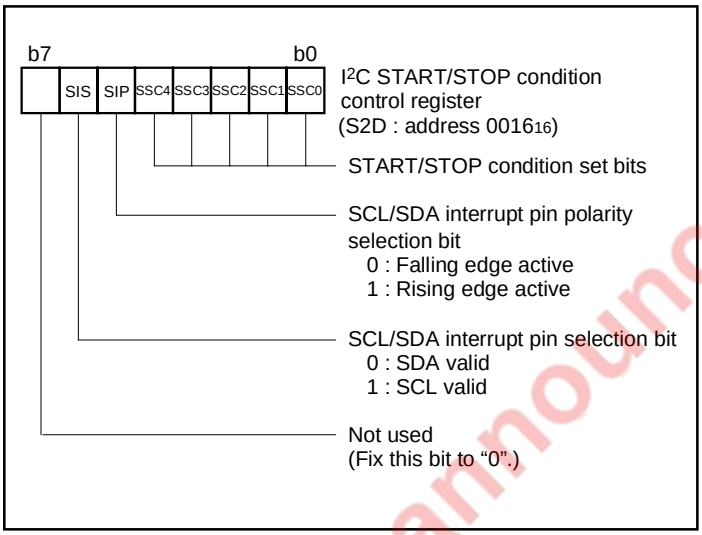


Fig. 66 Structure of I²C START/STOP condition control register

Table 12 Recommended set value to START/STOP condition set bits (SSC4–SSC0) for each oscillation frequency

Oscillation frequency f(XIN) (MHz)	Main clock divide ratio	Internal clock ϕ (MHz)	START/STOP condition control register	SCL release time (µs)	Setup time (µs)	Hold time (µs)
8	2	4	XXX11010	6.75 µs (27 cycles)	3.5 µs (14 cycles)	3.25 µs (13 cycles)
			XXX11000	6.25 µs (25 cycles)	3.25 µs (13 cycles)	3.0 µs (12 cycles)
8	8	1	XXX00100	5.0 µs (5 cycles)	3.0 µs (3 cycles)	2.0 µs (2 cycles)
4	2	2	XXX01100	6.5 µs (13 cycles)	3.5 µs (7 cycles)	3.0 µs (6 cycles)
			XXX01010	5.5 µs (11 cycles)	3.0 µs (6 cycles)	2.5 µs (5 cycles)
2	2	1	XXX00100	5.0 µs (5 cycles)	3.0 µs (3 cycles)	2.0 µs (2 cycles)

Note: Do not set an odd number to the START/STOP condition set bits (SSC4 to SSC0) and "00000₂".

[I²C Special Mode Status Register (S3)] 0012₁₆

The I²C special mode status register (S3: address 0012₁₆) consists of the flags indicating I²C operating state in the I²C special mode, which is set by the I²C special mode control register (S3D: address 0017₁₆).

The stop condition flag is valid in all operating modes.

•Bit 0: Slave address 0 comparison flag (AAS0)
Bit 1: Slave address 1 comparison flag (AAS1)
Bit 2: Slave address 2 comparison flag (AAS2)

These flags indicate a comparison result of address data. These flags are valid only when the slave address control bit (MSLAD) is "1".

In the 7-bit addressing format of the slave reception mode, the respective slave address i (i = 0, 1, 2) comparison flags corresponding to the I²C slave address registers 0 to 2 are set to "1" when an address data immediately after an occurrence of a START condition agrees with the high-order 7-bit slave address stored in the I²C slave address registers 0 to 2 (addresses 0FF7₁₆ to 0FF9₁₆).

In the 10-bit addressing format of the slave mode, the respective slave address i (i = 0, 1, 2) comparison flags corresponding to the I²C slave address registers are set to "1" when an address data is compared with the 8 bits consisting of the slave address stored in the I²C slave address registers 0 to 2 and the RWB bit, and the first byte agrees.

These flags are initialized to "0" at reset, when the slave address control bit (MSLAD) is "0", or when writing data to the I²C data shift register (S0: address 0011₁₆).

•Bit 5: SCL pin low hold 2 flag (PIN2)

When the ACK interrupt control bit (ACKICON) and the ACK clock bit (ACK) are "1", this flag is set to "0" in synchronization with the falling of the data's last SCL clock, just before the ACK clock. The SCL pin is simultaneously held low, and the I²C interrupt request occurs.

This flag is initialized to "1" at reset, when the ACK interrupt control bit (ACKICON) is "0", or when writing "1" to the SCL pin low hold 2 flag set bit (PIN2IN).

The SCL pin is held low when either the SCL pin low hold bit (PIN) or the SCL pin low hold 2 flag (PIN2) becomes "0". The low hold state of the SCL pin is released when both the SCL pin low hold bit (PIN) and the SCL pin low hold 2 flag (PIN2) are "1".

•Bit 7: Stop condition flag (SPCF)

This flag is set to "1" when a STOP condition occurs.

This flag is initialized to "0" at reset, when the I²C-BUS interface enable bit (ES0) is "0", or when writing "1" to the STOP condition flag clear bit (SPFCL).

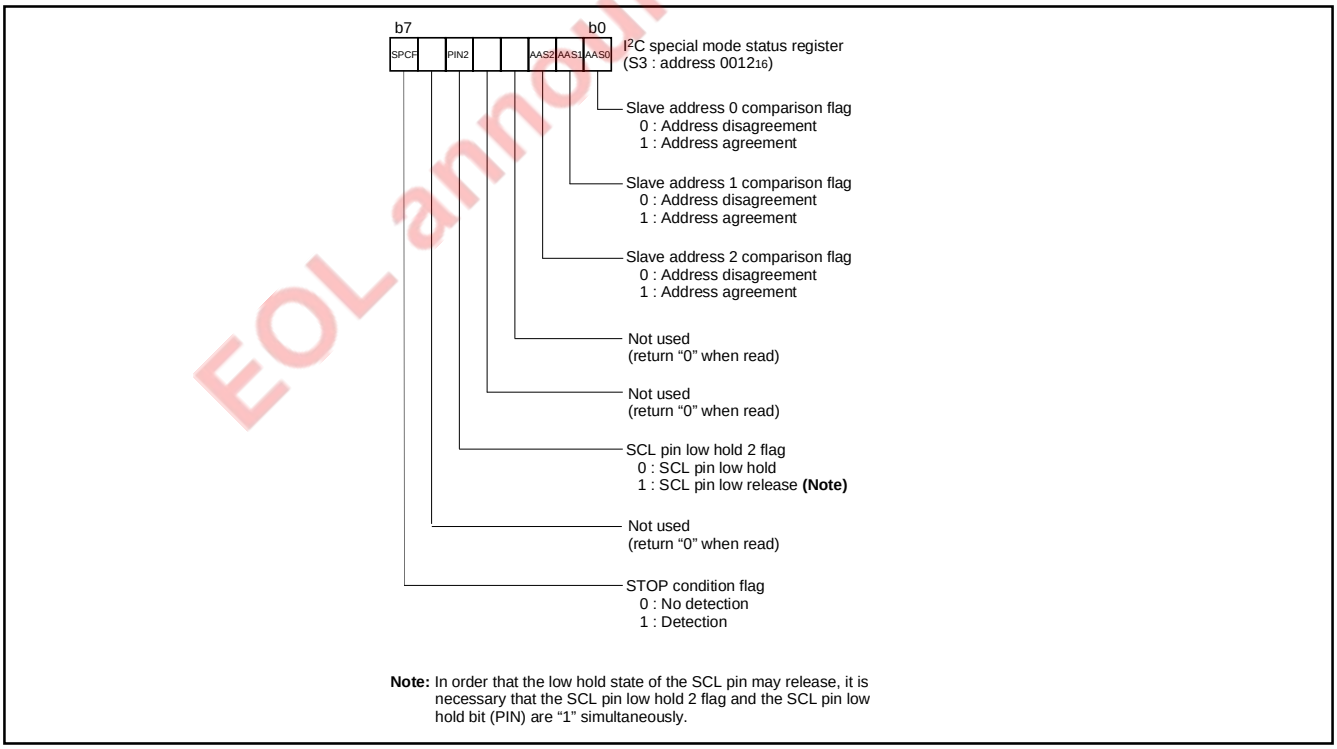


Fig. 67 Structure of I²C special mode status register

[I²C Special Mode Control Register (S3D)] 001716

The I²C special mode control register (S3D: address 001716) controls special functions such as occurrence timing of reception interrupt request and extending slave address comparison to 3 bytes.

•Bit 1: ACK interrupt control bit (ACKICON)

This bit controls the timing of I²C interrupt request occurrence at completion of data receiving due to master reception or slave reception.

When this bit is "0", the SCL pin low hold bit (PIN) is set to "0" in synchronization with the falling of the last SCL clock, including the ACK clock. The SCL pin is simultaneously held low, and the I²C interrupt request occurs.

When this bit is "1" and the ACK clock bit (ACK) is "1", the SCL pin low hold 2 flag (PIN2) is set to "0" in synchronization with the falling of the data's last SCL clock, just before the ACK clock. The SCL pin is simultaneously held low, and the I²C interrupt request occurs again. The ACK bit can be changed after the contents of data are confirmed by using this function.

•Bit 2: I²C slave address control bit (MSLAD)

This bit controls a slave address. When this bit is "0", only the I²C slave address register 0 (address 0FF716) becomes valid as a slave address and a read/write bit.

When this bit is "1", all of the I²C slave address registers 0 to 2 (addresses 0FF716 to 0FF916) become valid as a slave address and a read/write bit. In this case, when an address data agrees with any one of the I²C slave address registers 0 to 2, the slave address comparison flag (AAS) is set to "1" and the I²C slave address comparison flag corresponding to the agreed I²C slave address registers 0 to 2 is also set to "1".

•Bit 5: SCL pin low hold 2 flag set bit (PIN2IN)

Writing "1" to this bit initializes the SCL pin low hold 2 flag (PIN2) to "1".

When writing "0", nothing is generated.

•Bit 6: SCL pin low hold set bit (PIN2HD)

When the SCL pin low hold bit (PIN) becomes "0", the SCL pin is held low. However, the SCL pin low hold bit (PIN) cannot be set to "0" by software. The SCL pin low hold set bit (PIN2HD) is used to , hold the SCL pin in the low state by software. When writing "1" to this bit, the SCL pin low hold 2 flag (PIN2) becomes "0", and the SCL pin is held low. When writing "0", nothing occurs.

•Bit 7: STOP condition flag clear bit (SPFCL)

Writing "1" to this bit initializes the STOP condition flag (SPCF) to "0".

When writing "0", nothing is generated.

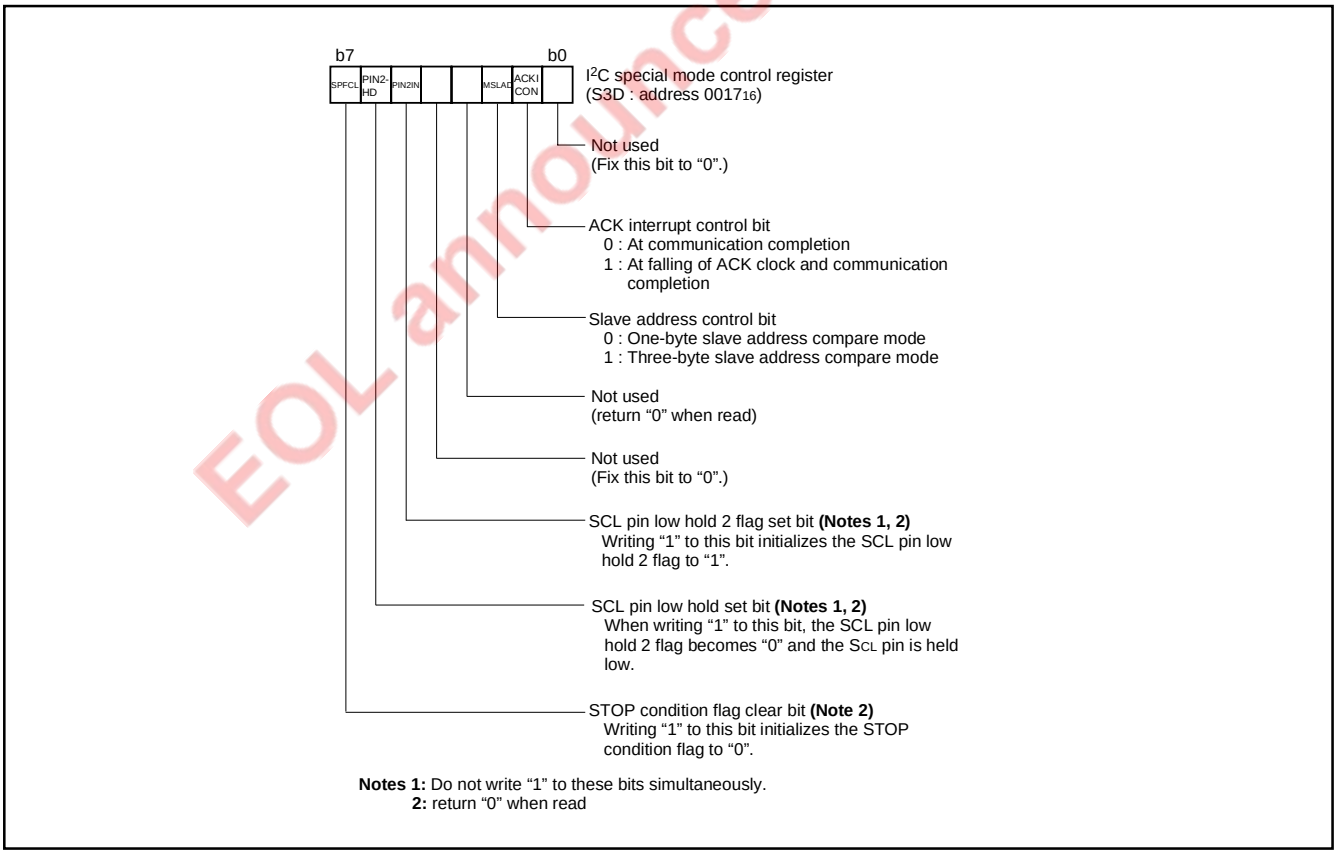


Fig. 68 Structure of I²C special mode control register

Address Data Communication

There are two address data communication formats, namely, 7-bit addressing format and 10-bit addressing format. The respective address communication formats are described below.

① 7-bit addressing format

To adapt the 7-bit addressing format, set the 10BIT SAD bit of the I²C control register (S1D: address 0014₁₆) to "0". The first 7-bit address data transmitted from the master is compared with the high-order 7-bit slave address stored in the I²C slave address register. At the time of this comparison, address comparison of the RWB bit of the I²C slave address register is not performed. For the data transmission format when the 7-bit addressing format is selected, refer to Figure 69, (1) and (2).

② 10-bit addressing format

To adapt the 10-bit addressing format, set the 10BIT SAD bit of the I²C control register (S1D: address 0014₁₆) to "1." An address comparison is performed between the first-byte address data transmitted from the master and the 8-bit slave address stored in the I²C slave address register. At the time of this comparison, an address comparison between the RWB bit of the I²C slave address register and the R/$\overline{W}$ bit which is the last bit of the address data transmitted from the master is made. In the 10-bit addressing mode, the RWB bit which is the last bit of the address data not only specifies the direction of communication for control data, but also is processed as an address data bit. When the first-byte address data agree with the slave address, the AAS bit of the I²C status register (S1: address 0013₁₆) is set to "1." After the second-byte address data is stored into the I²C data shift register (S0: address 0011₁₆), perform an address comparison between the second-byte data and the slave address by software. When the address data of the 2 bytes agree with the slave address, set the RWB bit of the I²C slave address register to "1" by software. This processing can make the 7-bit slave address and R/$\overline{W}$ data agree, which are received after a RESTART condition is detected, with the value of the I²C slave address register. For the data transmission format when the 10-bit addressing format is selected, refer to Figure 69, (3) and (4).

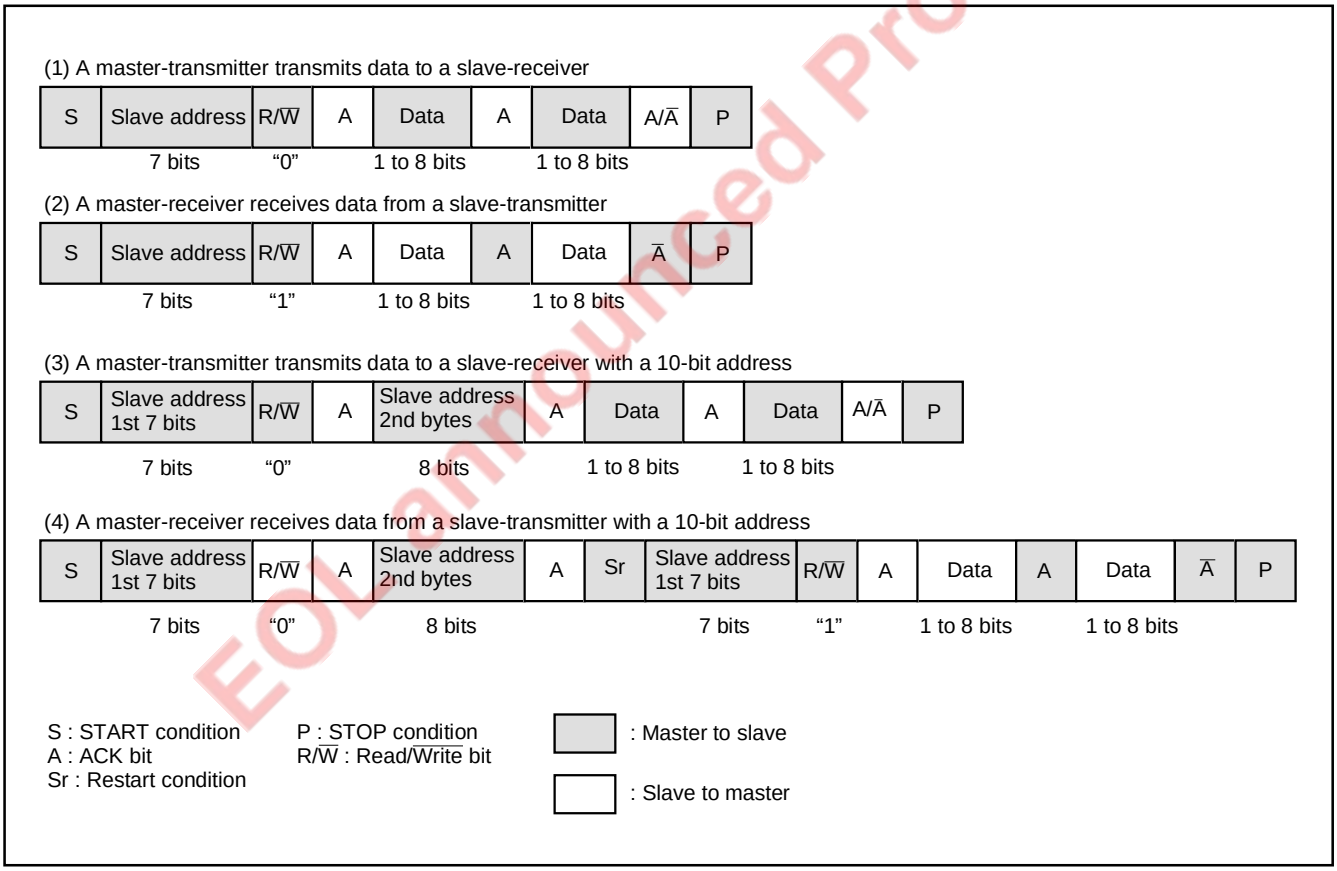


Fig. 69 Address data communication format

Example of Master Transmission

An example of master transmission in the standard clock mode, at the SCL frequency of 100 kHz and in the ACK return mode is shown below.

① Set a slave address in the high-order 7 bits of the I^2C slave address register and "0" into the RWB bit.
② Set the ACK return mode and SCL = 100 kHz by setting "85_{16}" in the I^2C clock control register (S2: address 0015_{16}).
③ Set "00_{16}" in the I^2C status register (S1: address 0013_{16}) so that transmission/reception mode can become initializing condition.
④ Set a communication enable status by setting "08_{16}" in the I^2C control register (S1D: address 0014_{16}).
⑤ Confirm the bus free condition by the BB flag of the I^2C status register (S1: address 0013_{16}).
⑥ Set the address data of the destination of transmission in the high-order 7 bits of the I^2C data shift register (S0: address 0011_{16}) and set "0" in the least significant bit.
⑦ Set "$F0_{16}$" in the I^2C status register (S1: address 0013_{16}) to generate a START condition. At this time, an SCL for 1 byte and an ACK clock automatically occur.
⑧ Set transmit data in the I^2C data shift register (S0: address 0011_{16}). At this time, an SCL and an ACK clock automatically occur.
⑨ When transmitting control data of more than 1 byte, repeat step ⑧.
⑩ Set "$D0_{16}$" in the I^2C status register (S1: address 0013_{16}) to generate a STOP condition if ACK is not returned from slave reception side or transmission ends.

Example of Slave Reception

An example of slave reception in the high-speed clock mode, at the SCL frequency of 400 kHz, in the ACK non-return mode and using the addressing format is shown below.

① Set a slave address in the high-order 7 bits of the I^2C slave address register and "0" in the RWB bit.
② Set the no ACK clock mode and SCL = 400 kHz by setting "25_{16}" in the I^2C clock control register (S2: address 0015_{16}).
③ Set "00_{16}" in the I^2C status register (S1: address 0013_{16}) so that transmission/reception mode can become initializing condition.
④ Set a communication enable status by setting "08_{16}" in the I^2C control register (S1D: address 0014_{16}).
⑤ When a START condition is received, an address comparison is performed.
⑥ •When all transmitted addresses are "0" (general call):
AD0 of the I^2C status register (S1: address 0013_{16}) is set to "1" and an interrupt request signal occurs.
• When the transmitted addresses agree with the address set in ①:
AAS of the I^2C status register (S1: address 0013_{16}) is set to "1" and an interrupt request signal occurs.
• In the cases other than the above AD0 and AAS of the I^2C status register (S1: address 0013_{16}) are set to "0" and no interrupt request signal occurs.
⑦ Set dummy data in the I^2C data shift register (S0: address 0011_{16}).
⑧ When receiving control data of more than 1 byte, repeat step ⑦.
⑨ When a STOP condition is detected, the communication ends.

■Precautions when using multi-master I²C-BUS interface

(1) Read-modify-write instruction

The precautions when the read-modify-write instruction such as SEB, CLB etc. is executed for each register of the multi-master I²C-BUS interface are described below.

- I²C data shift register (S0: address 0011₁₆)
  When executing the read-modify-write instruction for this register during transfer, data may become a value not intended.
- I²C slave address registers 0 to 2 (S0D0 to S0D2: addresses 0FF7₁₆ to0FF9₁₆)
  When the read-modify-write instruction is executed for this register at detecting the STOP condition, data may become a value not intended. It is because H/W changes the read/write bit (RWB) at the above timing.
- I²C status register (S1: address 0013₁₆)
  Do not execute the read-modify-write instruction for this register because all bits of this register are changed by H/W.
- I²C control register (S1D: address 0014₁₆)
  When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become a value not intended. Because H/W changes the bit counter (BC0-BC2) at the above timing.
- I²C clock control register (S2: address 0015₁₆)
  The read-modify-write instruction can be executed for this register.
- I²C START/STOP condition control register (S2D: address 0016₁₆)
  The read-modify-write instruction can be executed for this register.

(2) START condition generating procedure using multi-master

1. Procedure example (The necessary conditions of the generating procedure are described as the following 2 to 5.

```
      ⋮
    LDA —                 (Taking out of slave address value)
    SEI                   (Interrupt disabled)
    BBS 5, S1, BUSBUSY    (BB flag confirming and branch process)
  BUSFREE:
    STA S0                (Writing of slave address value)
    LDM #$F0, S1          (Trigger of START condition generating)
    CLI                   (Interrupt enabled)
      ⋮
  BUSBUSY:
    CLI                   (Interrupt enabled)
      ⋮
```

2. Use “Branch on Bit Set” of “BBS 5, S1, –” for the BB flag confirming and branch process.
3. Use “STA $12, STX $12” or “STY $12” of the zero page addressing instruction for writing the slave address value to the I²C data shift register.
4. Execute the branch instruction of above 2 and the store instruction of above 3 continuously shown the above procedure example.
5. Disable interrupts during the following three process steps:
   • BB flag confirming
   • Writing of slave address value
   • Trigger of START condition generating
   When the condition of the BB flag is bus busy, enable interrupts immediately.

(3) RESTART condition generating procedure

1. Procedure example (The necessary conditions of the generating procedure are described as the following 2 to 4.)
   Execute the following procedure when the PIN bit is “0.”

```
      ⋮
    LDM #$00, S1          (Select slave receive mode)
    LDA —                 (Taking out of slave address value)
    SEI                   (Interrupt disabled)
    STA S0                (Writing of slave address value)
    LDM #$F0, S1          (Trigger of RESTART condition generating)
    CLI                   (Interrupt enabled)
      ⋮
```

2. Select the slave receive mode when the PIN bit is “0.” Do not write “1” to the PIN bit. Neither “0” nor “1” is specified for the writing to the BB bit.
   The TRX bit becomes “0” and the SDA pin is released.
3. The SCL pin is released by writing the slave address value to the I²C data shift register.
4. Disable interrupts during the following two process steps:
   • Writing of slave address value
   • Trigger of RESTART condition generating

(4) Writing to I²C status register

Do not execute an instruction to set the PIN bit to “1” from “0” and an instruction to set the MST and TRX bits to “0” from “1” simultaneously. It is because it may enter the state that the SCL pin is released and the SDA pin is released after about one machine cycle. Do not execute an instruction to set the MST and TRX bits to “0” from “1” simultaneously when the PIN bit is “1.” It is because it may become the same as above.

(5) Process of after STOP condition generating

Do not write data in the I²C data shift register S0 and the I²C status register S1 until the bus busy flag BB becomes “0” after generating the STOP condition in the master mode. It is because the STOP condition waveform might not be normally generated. Reading to the above registers does not have the problem.

RESET CIRCUIT

To reset the microcomputer, $\overline{\text{RESET}}$ pin should be held at an “L” level for 16 cycles or more of XIN. Then the $\overline{\text{RESET}}$ pin is returned to an “H” level (the power source voltage should be between 2.7 V to 5.5 V, and the oscillation should be stable), reset is released. After the reset is completed, the program starts from the address contained in address FFFD16 (high-order byte) and address FFFC16 (low-order byte).

Input to the $\overline{\text{RESET}}$ pin in the following procedure.

●When power source is stabilized

(1) Input “L” level to $\overline{\text{RESET}}$ pin.
(2) Input “L” level for 16 cycles or more to XIN pin.
(3) Input “H” level to $\overline{\text{RESET}}$ pin.

●At power-on

(1) Input “L” level to $\overline{\text{RESET}}$ pin.
(2) Increase the power source voltage to 2.7 V.
(3) Wait for td(P-R) until internal power source has stabilized.
(4) Input “L” level for 16 cycles or more to XIN pin.
(5) Input “H” level to $\overline{\text{RESET}}$ pin.

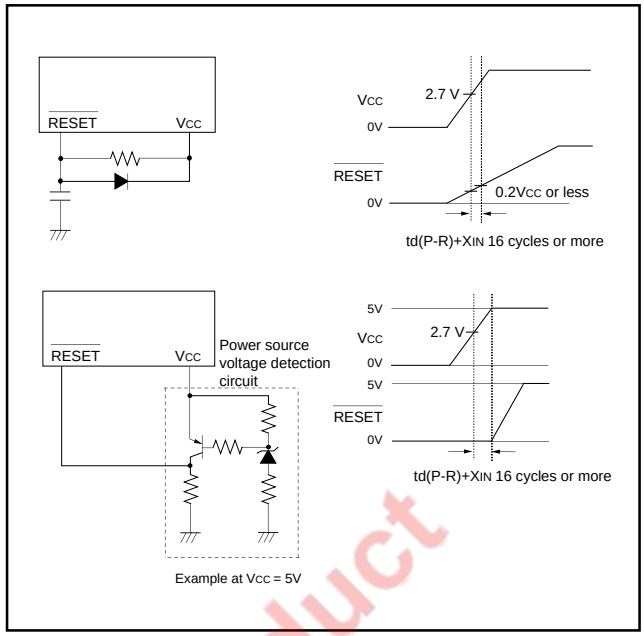


Fig. 70 Reset circuit example

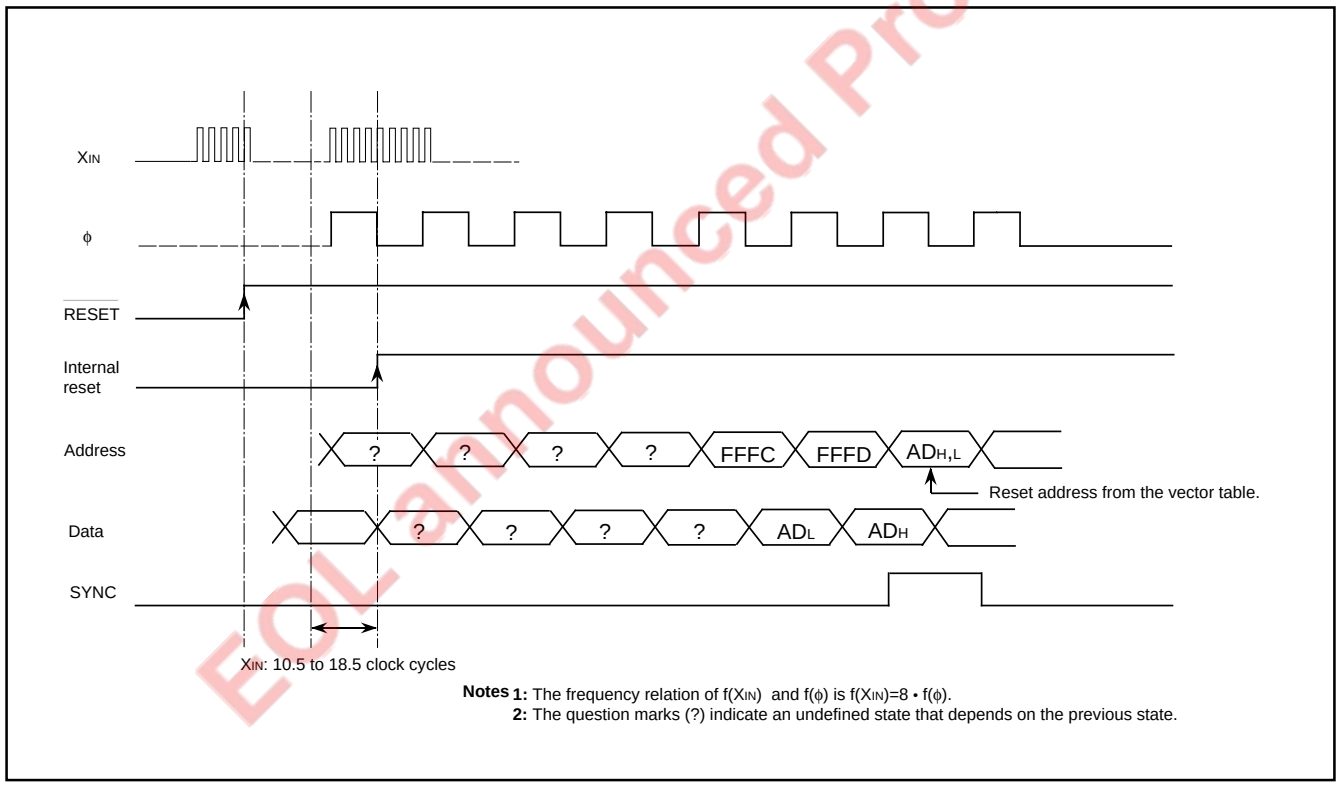


Fig. 71 Reset sequence

		Address	Register contents
(1)	Port P0 (P0)	0000_{16}	00_{16}
(2)	Port P0 direction register (P0D)	0001_{16}	00_{16}
(3)	Port P1 (P1)	0002_{16}	00_{16}
(4)	Port P1 direction register (P1D)	0003_{16}	00_{16}
(5)	Port P2 (P2)	0004_{16}	00_{16}
(6)	Port P2 direction register (P2D)	0005_{16}	00_{16}
(7)	Port P3 (P3)	0006_{16}	00_{16}
(8)	Port P3 direction register (P3D)	0007_{16}	00_{16}
(9)	Port P4 (P4)	0008_{16}	00_{16}
(10)	Port P4 direction register (P4D)	0009_{16}	00_{16}
(11)	Port P5 (P5)	$000A_{16}$	00_{16}
(12)	Port P5 direction register (P5D)	$000B_{16}$	00_{16}
(13)	Port P6 (P6)	$000C_{16}$	00_{16}
(14)	Port P6 direction register (P6D)	$000D_{16}$	00_{16}
(15)	Timer 12, X count source selection register (T12XCSS)	$000E_{16}$	0 0 1 1 0 0 1 1
(16)	Timer Y, Z count source selection register (TYZCSS)	$000F_{16}$	0 0 1 1 0 0 1 1
(17)	MISRG	0010_{16}	00_{16}
(18)	I^2C data shift register (S0)	0011_{16}	X X X X X X X X
(19)	I^2C special mode status register (S3)	0012_{16}	0 0 1 0 0 0 0 0
(20)	I^2C status register (S1)	0013_{16}	0 0 0 1 0 0 0 X
(21)	I^2C control register (S1D)	0014_{16}	00_{16}
(22)	I^2C clock control register (S2)	0015_{16}	00_{16}
(23)	I^2C START/STOP condition control register (S2D)	0016_{16}	0 0 0 1 1 0 1 0
(24)	I^2C special mode control register (S3D)	0017_{16}	00_{16}
(25)	Transmit/Receive buffer register 1 (TB1/RB1)	0018_{16}	X X X X X X X X
(26)	Serial I/O1 status register (SIO1STS)	0019_{16}	1 0 0 0 0 0 0 0
(27)	Serial I/O1 control register (SIO1CON)	$001A_{16}$	00_{16}
(28)	UART1 control register (UART1CON)	$001B_{16}$	1 1 1 0 0 0 0 0
(29)	Baud rate generator 1 (BRG1)	$001C_{16}$	X X X X X X X X
(30)	Serial I/O2 control register (SIO2CON)	$001D_{16}$	00_{16}
(31)	Watchdog timer control register (WDTCON)	$001E_{16}$	0 0 1 1 1 1 1 1
(32)	Serial I/O2 register (SIO2)	$001F_{16}$	X X X X X X X X
(33)	Prescaler 12 (PRE12)	0020_{16}	FF_{16}
(34)	Timer 1 (T1)	0021_{16}	01_{16}
(35)	Timer 2 (T2)	0022_{16}	FF_{16}
(36)	Timer XY mode register (TM)	0023_{16}	00_{16}
(37)	Prescaler X (PREX)	0024_{16}	FF_{16}
(38)	Timer X (TX)	0025_{16}	FF_{16}
(39)	Prescaler Y (PREY)	0026_{16}	FF_{16}
(40)	Timer Y (TY)	0027_{16}	FF_{16}

		Address	Register contents
(41)	Timer Z (low-order) (TZL)	0028_{16}	FF_{16}
(42)	Timer Z (high-order) (TZH)	0029_{16}	FF_{16}
(43)	Timer Z mode register (TZM)	$002A_{16}$	00_{16}
(44)	PWM control register (PWMCON)	$002B_{16}$	00_{16}
(45)	PWM prescaler (PREPWM)	$002C_{16}$	X X X X X X X X
(46)	PWM register (PWM)	$002D_{16}$	X X X X X X X X
(47)	Baud rate generator 3 (BRG3)	$002F_{16}$	X X X X X X X X
(48)	Transmit/Receive buffer register 3 (TB3/RB3)	0030_{16}	X X X X X X X X
(49)	Serial I/O3 status register (SIO3STS)	0031_{16}	1 0 0 0 0 0 0 0
(50)	Serial I/O3 control register (SIO3CON)	0032_{16}	00_{16}
(51)	UART3 control register (UART3CON)	0033_{16}	1 1 1 0 0 0 0 0
(52)	AD/DA control register (ADCON)	0034_{16}	0 0 0 0 1 0 0 0
(53)	AD conversion register 1 (AD1)	0035_{16}	X X X X X X X X
(54)	DA1 conversion register (DA1)	0036_{16}	00_{16}
(55)	DA2 conversion register (DA2)	0037_{16}	00_{16}
(56)	AD conversion register 2 (AD2)	0038_{16}	0 0 0 0 0 0 X X
(57)	Interrupt source selection register (INTSEL)	0039_{16}	00_{16}
(58)	Interrupt edge selection register (INTEDGE)	$003A_{16}$	00_{16}
(59)	CPU mode register (CPUM)	$003B_{16}$	0 1 0 0 1 0 0 0
(60)	Interrupt request register 1 (IREQ1)	$003C_{16}$	00_{16}
(61)	Interrupt request register 2 (IREQ2)	$003D_{16}$	00_{16}
(62)	Interrupt control register 1 (ICON1)	$003E_{16}$	00_{16}
(63)	Interrupt control register 2 (ICON2)	$003F_{16}$	00_{16}
(64)	Flash memory control register 0 (FMCR0)	$0FE0_{16}$	01_{16}
(65)	Flash memory control register 1 (FMCR1)	$0FE1_{16}$	40_{16}
(66)	Flash memory control register 2 (FMCR2)	$0FE2_{16}$	45_{16}
(67)	Port P0 pull-up control register (PULL0)	$0FF0_{16}$	00_{16}
(68)	Port P1 pull-up control register (PULL1)	$0FF1_{16}$	00_{16}
(69)	Port P2 pull-up control register (PULL2)	$0FF2_{16}$	00_{16}
(70)	Port P3 pull-up control register (PULL3)	$0FF3_{16}$	00_{16}
(71)	Port P4 pull-up control register (PULL4)	$0FF4_{16}$	00_{16}
(72)	Port P5 pull-up control register (PULL5)	$0FF5_{16}$	00_{16}
(73)	Port P6 pull-up control register (PULL6)	$0FF6_{16}$	00_{16}
(74)	I^2C slave address register 0 (S0D0)	$0FF7_{16}$	00_{16}
(75)	I^2C slave address register 1 (S0D1)	$0FF8_{16}$	00_{16}
(76)	I^2C slave address register 2 (S0D3)	$0FF9_{16}$	00_{16}
(77)	Processor status register	(PS)	X X X X X 1 X X
(78)	Program counter	(PC_H)	$FFFD_{16}$ contents
		(PC_L)	$FFFC_{16}$ contents

Note : X : Not fixed
Since the initial values for other than above mentioned registers and RAM contents are indefinite at reset, they must be set.

Fig. 72 Internal status at reset

CLOCK GENERATING CIRCUIT

The 3804 group (Spec. H) has two built-in oscillation circuits: main clock XIN-XOUT oscillation circuit and sub clock XCIN-XCOUT oscillation circuit. An oscillation circuit can be formed by connecting a resonator between XIN and XOUT (XCIN and XCOUT). Use the circuit constants in accordance with the resonator manufacturer's recommended values. No external resistor is needed between XIN and XOUT since a feed-back resistor exists on-chip.(An external feed-back resistor may be needed depending on conditions.) However, an external feed-back resistor is needed between XCIN and XCOUT.

Immediately after power on, only the XIN oscillation circuit starts oscillating, and XCIN and XCOUT pins function as I/O ports.

Frequency Control
(1) Middle-speed mode

The internal clock ϕ is the frequency of XIN divided by 8. After reset is released, this mode is selected.

(2) High-speed mode

The internal clock ϕ is half the frequency of XIN.

(3) Low-speed mode

The internal clock ϕ is half the frequency of XCIN.

(4) Low power dissipation mode

The low power consumption operation can be realized by stopping the main clock XIN in low-speed mode. To stop the main clock, set bit 5 of the CPU mode register to "1." When the main clock XIN is restarted (by setting the main clock stop bit to "0"), set sufficient time for oscillation to stabilize.

Oscillation Control
(1) Stop mode

If the STP instruction is executed, the internal clock ϕ stops at an "H" level, and XIN and XCIN oscillators stop. When the oscillation stabilizing time set after STP instruction released bit is "0," the prescaler 12 is set to "FF16" and timer 1 is set to "0116." When the oscillation stabilizing time set after STP instruction released bit is "1," set the sufficient time for oscillation of used oscillator to stabilize since nothing is set to the prescaler 12 and timer 1.

After STP instruction is released, the input of the prescaler 12 is connected to count source which had set at executing the STP instruction, and the output of the prescaler 12 is connected to timer 1. Set the timer 1 interrupt enable bit to disabled ("0") before executing the STP instruction. Oscillator restarts when an external interrupt is received, but the internal clock ϕ is not supplied to the CPU (remains at "H") until timer 1 underflows. The internal clock ϕ is supplied for the first time, when timer 1 underflows. Therefore make sure not to set the timer 1 interrupt request bit to "1" before the STP instruction stops the oscillator. When the oscillator is restarted by reset, apply "L" level to the $\overline{\text{RESET}}$ pin until the oscillation is stable since a wait time will not be generated.

The internal power supply circuit is changed to low power consumption mode for consumption current reduction at the time of STP instruction execution.

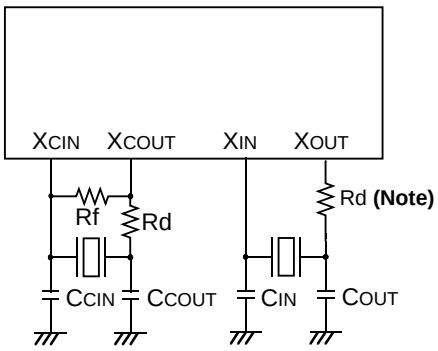
Although an internal power supply circuit is usually changed to the normal operation mode at the time of the return from an STP instruction, since a certain time is required to start the power supply to the flash memory and operation of flash memory to be enabled, set wait time 100 μs or more by the oscillation stabilization time set function after release of the STP instruction which used the timer 1.

(2) Wait mode

If the WIT instruction is executed, the internal clock ϕ stops at an "H" level, but the oscillator does not stop. The internal clock ϕ restarts when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

■Note

- If you switch the mode between middle/high-speed and low-speed, stabilize both XIN and XCIN oscillations. The sufficient time is required for the sub clock to stabilize, especially immediately after power on and at returning from stop mode. When switching the mode between middle/high-speed and low-speed, set the frequency on condition that f(XIN) > 3f(XCIN).
- When using the quartz-crystal oscillator of high frequency, such as 16 MHz etc., it may be necessary to select a specific oscillator with the specification demanded.



Notes : Insert a damping resistor if required.
The resistance will vary depending on the oscillator and the oscillation drive capacity setting.
Use the value recommended by the maker of the oscillator.
Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip though a feedback resistor exists on-chip, insert a feedback resistor between XIN and XOUT following the instruction.

Fig. 73 Ceramic resonator circuit

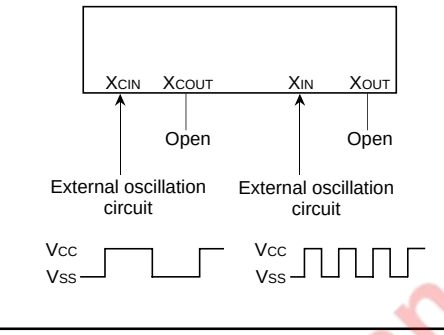
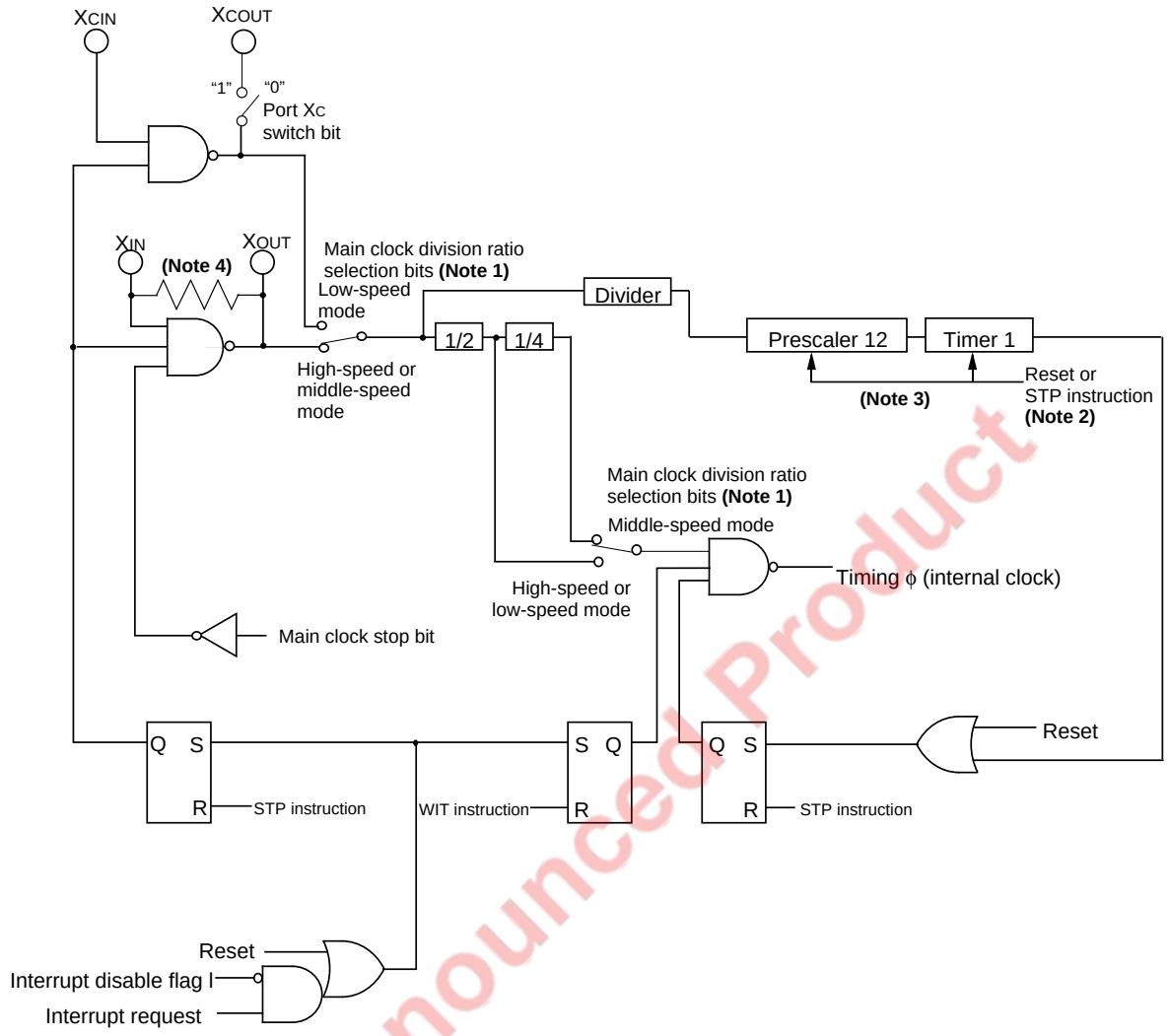
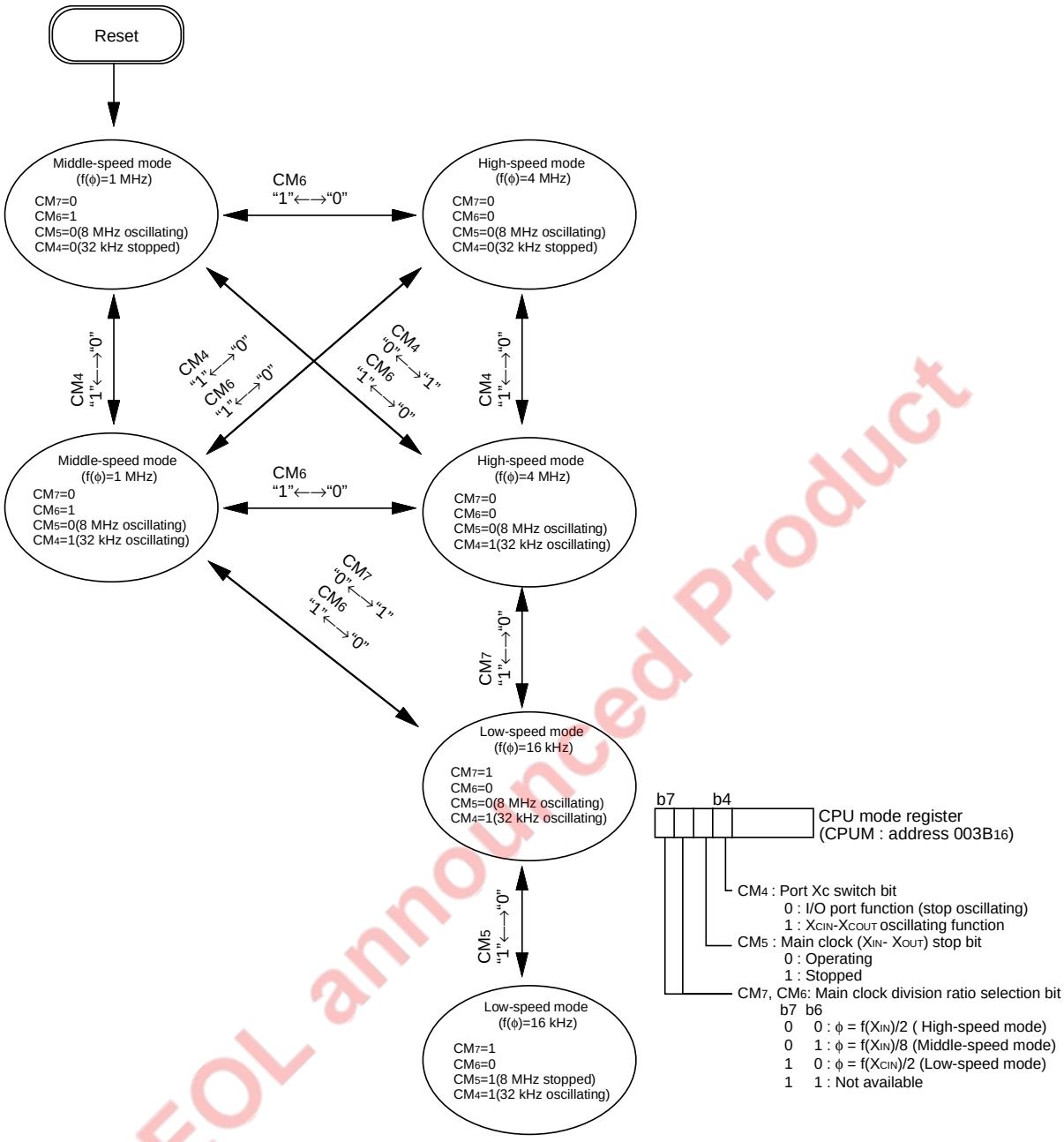


Fig. 74 External clock input circuit



Notes 1: Either high-speed, middle-speed or low-speed mode is selected by bits 7 and 6 of the CPU mode register. When low-speed mode is selected, set port Xc switch bit (b4) to "1".
2: f(XIN)/16 is supplied as the count source to the prescaler 12 at reset. The count source before executing the STP instruction is supplied as the count source at executing STP instruction.
3: When bit 0 of MISRG is "0", timer 1 is set "01$_{16}$" and prescaler 12 is set "FF$_{16}$" automatically. When bit 0 of MISRG is "1", set the appropriate value to them in accordance with oscillation stablizing time required by the using oscillator because nothing is automatically set into timer 1 and prescaler 12.
4: Although a feed-back resistor exists on-chip, an external feed-back resistor may be needed depending on conditions.

Fig. 75 System clock generating circuit block diagram



Notes 1 : Switch the mode by the allows shown between the mode blocks. (Do not switch between the modes directly without an allow.)
2 : The all modes can be switched to the stop mode or the wait mode and return to the source mode when the stop mode or the wait mode is ended.
3 : Timer operates in the wait mode.
4 : When the stop mode is ended, a delay of approximately 1 ms occurs by connecting prescaler 12 and Timer 1 in middle/high-speed mode.
5 : When the stop mode is ended, a delay of approximately 0.25 s occurs by Timer 1 and Timer 2 in low-speed mode.
6 : Wait until oscillation stabilizes after oscillating the main clock XIN before the switching from the low-speed mode to middle/high-speed mode.
7 : The example assumes that 8 MHz is being applied to the XIN pin and 32 kHz to the XCIN pin. ϕ indicates the internal clock.

Fig. 76 State transitions of system clock

FLASH MEMORY MODE

The 3804 group (spec. H) has the flash memory that can be rewritten with a single power source.

For this flash memory, three flash memory modes are available in which to read, program, and erase: the parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and the CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU).

This flash memory has some blocks on it as shown in Figure 77 and each block can be erased.

In addition to the ordinary User ROM area to store the MCU operation control program, the flash memory has a Boot ROM area that is used to store a program to control rewriting in CPU rewrite and standard serial I/O modes. This Boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. However, the user can write a rewrite control program in this area that suits the user's application system. This Boot ROM area can be rewritten in only parallel I/O mode.

● Summary

Table 13 lists the summary of the 3804 Group (spec. H).

Table 13 Summary of 3804 group (spec. H)

Item		Specifications
Power source voltage (Vcc)		VCC = 2.7 to 5.5 V
Program/Erase VPP voltage (VPP)		VCC = 2.7 to 5.5 V
Flash memory mode		3 modes; Parallel I/O mode, Standard serial I/O mode, CPU rewrite mode
Erase block division	User ROM area/Data ROM area	Refer to Fig. 77.
	Boot ROM area (Note)	Not divided (4K bytes)
Program method		In units of bytes
Erase method		Block erase
Program/Erase control method		Program/Erase control by software command
Number of commands		5 commands
Number of program/Erase times		100
ROM code protection		Available in parallel I/O mode and standard serial I/O mode

Note: The Boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. This Boot ROM area can be erased and written in only parallel I/O mode.

● Boot Mode

The control program for CPU rewrite mode must be written into the User ROM or Boot ROM area in parallel I/O mode beforehand. (If the control program is written into the Boot ROM area, the standard serial I/O mode becomes unusable.)
See Figure 77 for details about the Boot ROM area.
Normal microcomputer mode is entered when the microcomputer is reset with pulling CNVss pin low. In this case, the CPU starts operating using the control program in the User ROM area.
When the microcomputer is reset and the CNVss pin high after pulling the P45/TxD1 pin and CNVss pin high, the CPU starts operating (start address of program is stored into addresses FFFC16 and FFFD16) using the control program in the Boot ROM area. This mode is called the “Boot mode”. Also, User ROM area can be rewritten using the control program in the Boot ROM area.

● Block Address

Block addresses refer to the maximum address of each block. These addresses are used in the block erase command.

● CPU Rewrite Mode

In CPU rewrite mode, the internal flash memory can be operated on (read, program, or erase) under control of the Central Processing Unit (CPU).
In CPU rewrite mode, only the User ROM area shown in Figure 77 can be rewritten; the Boot ROM area cannot be rewritten. Make sure the program and block erase commands are issued for only the User ROM area and each block area.
The control program for CPU rewrite mode can be stored in either User ROM or Boot ROM area. In the CPU rewrite mode, because the flash memory cannot be read from the CPU, the rewrite control program must be transferred to internal RAM area before it can be executed.

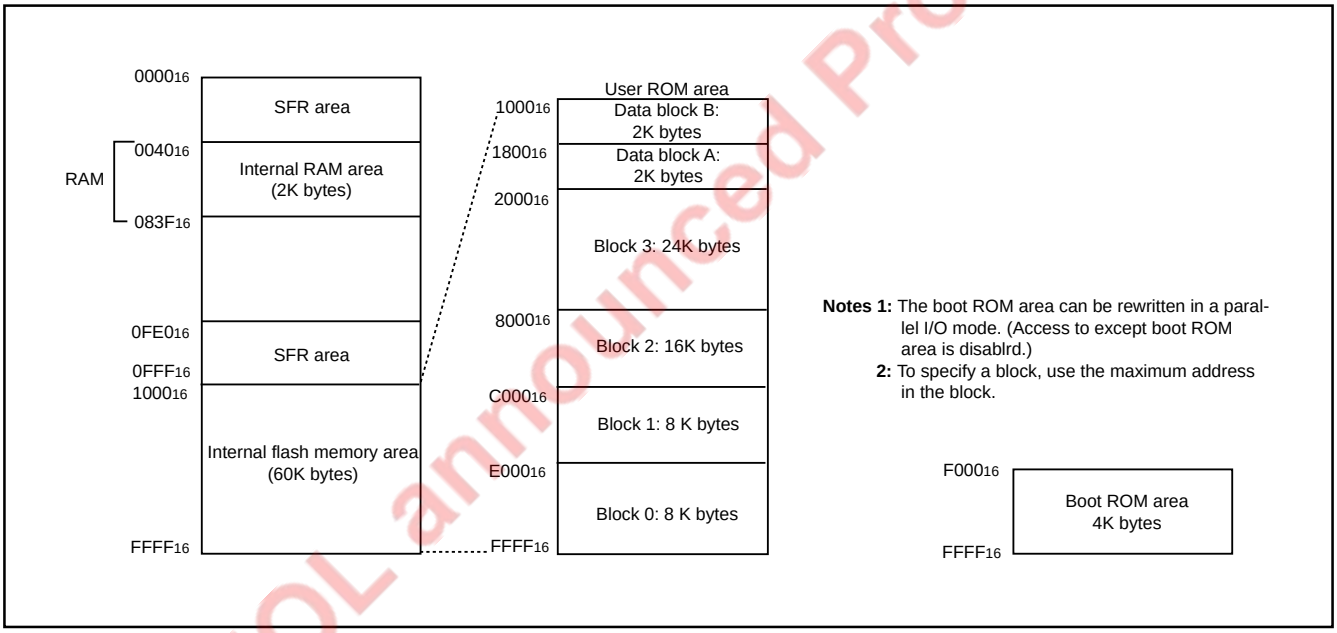


Fig. 77 Block diagram of built-in flash memory

●Outline Performance

CPU rewrite mode is usable in the single-chip or Boot mode. The only User ROM area can be rewritten.

In CPU rewrite mode, the CPU erases, programs and reads the internal flash memory as instructed by software commands. This rewrite control program must be transferred to internal RAM area before it can be executed.

The MCU enters CPU rewrite mode by setting “1” to the CPU rewrite mode select bit (bit 1 of address 0FE0$_{16}$). Then, software commands can be accepted.

Use software commands to control program and erase operations. Whether a program or erase operation has terminated normally or in error can be verified by reading the status register.

Figure 78 shows the flash memory control register 0.

Bit 0 of the flash memory control register 0 is the RY/$\overline{BY}$ status flag used exclusively to read the operating status of the flash memory. During programming and erase operations, it is “0” (busy). Otherwise, it is “1” (ready).

Bit 1 of the flash memory control register 0 is the CPU rewrite mode select bit. When this bit is set to “1”, the MCU enters CPU rewrite mode. And then, software commands can be accepted. In CPU rewrite mode, the CPU becomes unable to access the internal flash memory directly. Therefore, use the control program in the internal RAM for write to bit 1. To set this bit 1 to “1”, it is necessary to write “0” and then write “1” in succession to bit 1. The bit can be set to “0” by only writing “0”.

Bit 2 of the flash memory control register 0 is the 8 KB user block E/W enable bit. By setting combination of bit 4 of the flash memory control register 2 and this bit as shown in Table 14, E/W is disabled to user block in the CPU rewriting mode.

Bit 3 of the flash memory control register 0 is the flash memory reset bit used to reset the control circuit of internal flash memory. This bit is used when flash memory access has failed. When the CPU rewrite mode select bit is “1”, setting “1” for this bit resets the control circuit. To release the reset, it is necessary to set this bit to “0”.

Bit 5 of the flash memory control register 0 is the User ROM area select bit and is valid only in the boot mode. Setting this bit to “1” in the boot mode switches an accessible area from the boot ROM area to the user ROM area. To use the CPU rewrite mode in the boot mode, set this bit to “1”. To rewrite bit 5, execute the user-original reprogramming control software transferred to the internal RAM in advance.

Bit 6 of the flash memory control register 0 is the program status flag. This bit is set to “1” when writing to flash memory is failed. When program error occurs, the block cannot be used.

Bit 7 of the flash memory control register 0 is the erase status flag. This bit is set to “1” when erasing flash memory is failed. When erase error occurs, the block cannot be used.

Figure 79 shows the flash memory control register 1.

Bit 0 of the flash memory control register 1 is the Erase suspend enable bit. By setting this bit to “1”, the erase suspend mode to suspend erase processing temporaly when block erase command is executed can be used. In order to set this bit to “1”, writing “0” and “1” in succession to bit 0. In order to set this bit to “0”, write “0” only to bit 0.

Bit 1 of the flash memory control register 1 is the erase suspend request bit. By setting this bit to “1” when erase suspend enable bit is “1”, the erase processing is suspended.

Bit 6 of the flash memory control register 1 is the erase suspend flag. This bit is cleared to “0” at the flash erasing.

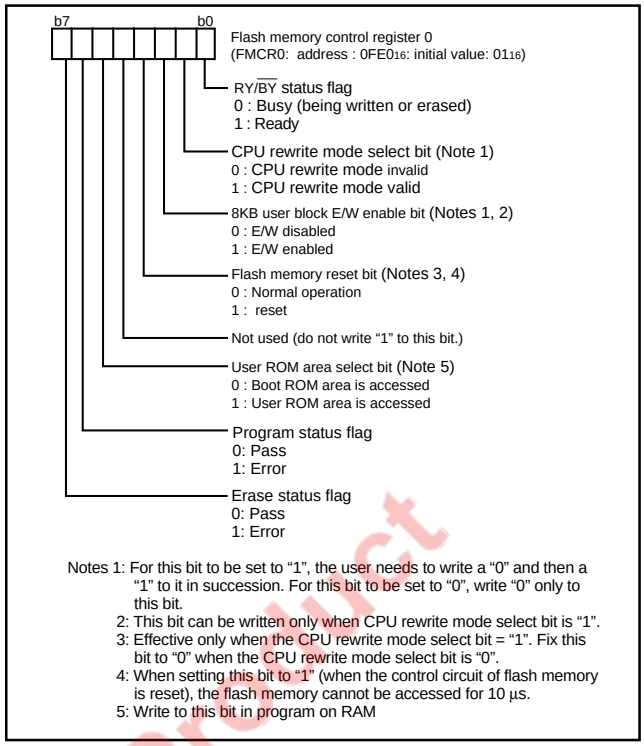


Fig. 78 Structure of flash memory control register 0

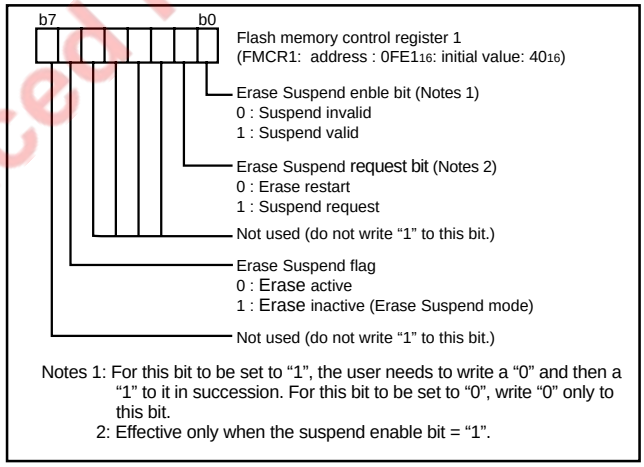
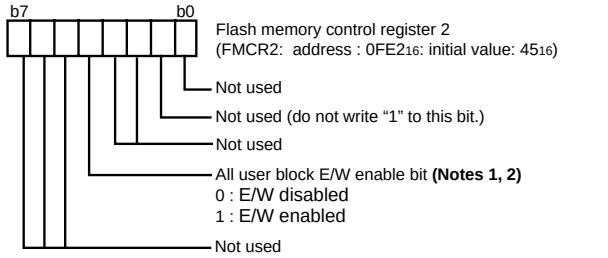


Fig. 79 Structure of flash memory control register 1



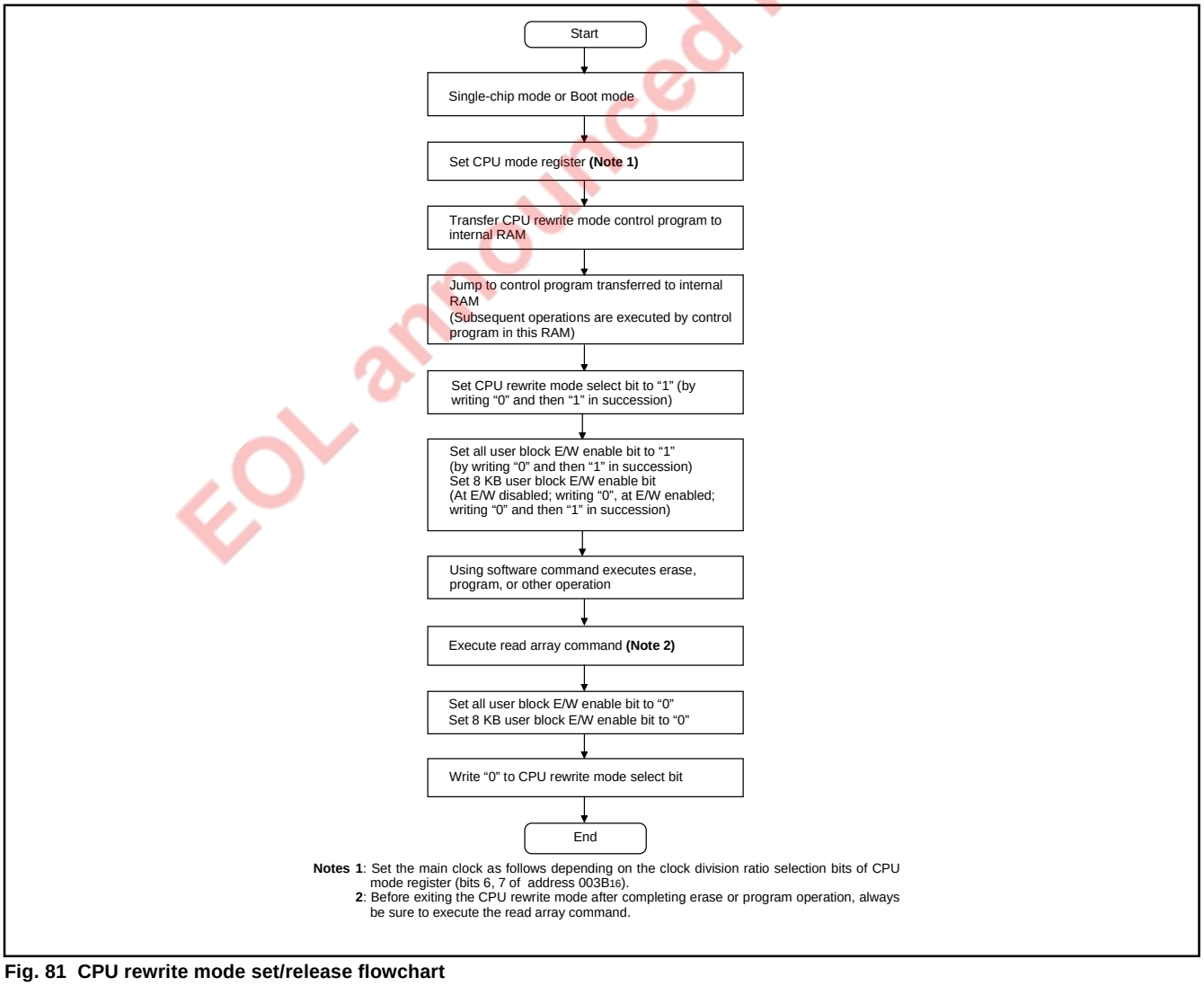
Notes 1: For this bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession. For this bit to be set to “0”, write “0” only to this bit.
2: Effective only when the CPU rewrite mode select bit = “1”.

Fig. 80 Structure of flash memory control register 2

Table 14 State of E/W inhibition function

All user block E/W enable bit	8 KB user block E/W enable bit	8 KB × 2 block Addresses C000₁₆ to FFFF₁₆	16 KB + 24 KB block Addresses 2000₁₆ to BFFF₁₆	Data block Addresses 1000₁₆ to 1FFF₁₆
0	0	E/W disabled	E/W disabled	E/W enabled
0	1	E/W disabled	E/W disabled	E/W enabled
1	0	E/W disabled	E/W enabled	E/W enabled
1	1	E/W enabled	E/W enabled	E/W enabled

Figure 81 shows a flowchart for setting/releasing CPU rewrite mode.



Notes 1: Set the main clock as follows depending on the clock division ratio selection bits of CPU mode register (bits 6, 7 of address 003B₁₆).
2: Before exiting the CPU rewrite mode after completing erase or program operation, always be sure to execute the read array command.

Fig. 81 CPU rewrite mode set/release flowchart

■ Notes on CPU Rewrite Mode

Take the notes described below when rewriting the flash memory in CPU rewrite mode.

●Operation speed

During CPU rewrite mode, set the system clock ϕ to 4.0 MHz or less using the clock division ratio selection bits (bits 6 and 7 of address $003B_{16}$).

●Instructions inhibited against use

The instructions which refer to the internal data of the flash memory cannot be used during CPU rewrite mode.

●Interrupts

The interrupts cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory.

●Watchdog timer

If the watchdog timer has been already activated, internal reset due to an underflow will not occur because the watchdog timer is surely cleared during program or erase.

●Reset

Reset is always valid. The MCU is activated using the boot mode at release of reset in the condition of CNVss = “H”, so that the program will begin at the address which is stored in addresses $FFFC_{16}$ and $FFFD_{16}$ of the boot ROM area.

● Software Commands

Table 15 lists the software commands.
After setting the CPU rewrite mode select bit to “1”, execute a software command to specify an erase or program operation.
Each software command is explained below.

• Read Array Command (FF16)

The read array mode is entered by writing the command code “FF16” in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the contents of the specified address are read out at the data bus (D0 to D7).
The read array mode is retained until another command is written.

• Read Status Register Command (7016)

When the command code “7016” is written in the first bus cycle, the contents of the status register are read out at the data bus (D0 to D7) by a read in the second bus cycle.
The status register is explained in the next section.

• Clear Status Register Command (5016)

This command is used to clear the bits SR4 and SR5 of the status register after they have been set. These bits indicate that operation has ended in an error. To use this command, write the command code “5016” in the first bus cycle.

• Program Command (4016)

Program operation starts when the command code “4016” is written in the first bus cycle. Then, if the address and data to program are written in the 2nd bus cycle, program operation (data programming and verification) will start.
Whether the write operation is completed can be confirmed by read status register or the RY/$\overline{BY}$ status flag. When the program starts, the read status register mode is entered automatically and the contents of the status register is read at the data bus (D0 to D7). The status register bit 7 (SR7) is set to “0” at the same time the write operation starts and is returned to “1” upon completion of the write operation. In this case, the read status register mode remains active until the read array command (FF16) is written.

The RY/$\overline{BY}$ status flag of the flash memory control register is “0” during write operation and “1” when the write operation is completed as is the status register bit 7.
At program end, program results can be checked by reading the status register.

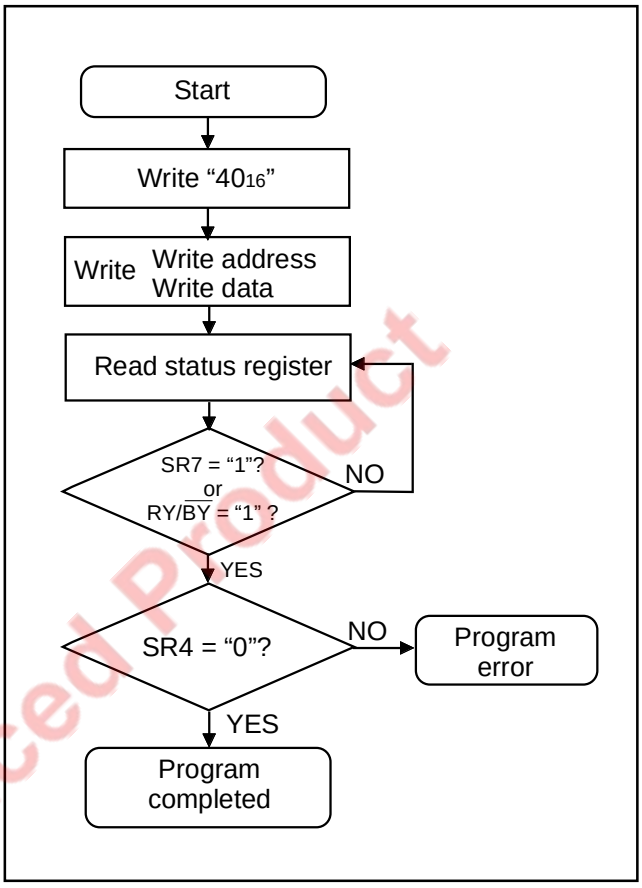


Fig. 82 Program flowchart

Table 15 List of software commands (CPU rewrite mode)

Command	Cycle number	First bus cycle			Second bus cycle		
		Mode	Address	Data (D0 to D7)	Mode	Address	Data (D0 to D7)
Read array	1	Write	X (Note 4)	FF16			
Read status register	2	Write	X	7016	Read	X	SRD (Note 1)
Clear status register	1	Write	X	5016			
Program	2	Write	X	4016	Write	WA (Note 2)	WD (Note 2)
Block erase	2	Write	X	2016	Write	BA (Note 3)	D016

Notes 1: SRD = Status Register Data
2: WA = Write Address, WD = Write Data
3: BA = Block Address to be erased (Input the maximum address of each block.)
4: X denotes a given address in the User ROM area.

• Block Erase Command (20_{16}/$D0_{16}$)

By writing the command code "20_{16}" in the first bus cycle and the confirmation command code "$D0_{16}$" and the block address in the second bus cycle that follows, the block erase (erase and erase verify) operation starts for the block address of the flash memory to be specified.

Whether the block erase operation is completed can be confirmed by read status register or the RY/$\overline{BY}$ status flag of flash memory control register. At the same time the block erase operation starts, the read status register mode is automatically entered, so that the contents of the status register can be read out. The status register bit 7 (SR7) is set to "0" at the same time the block erase operation starts and is returned to "1" upon completion of the block erase operation. In this case, the read status register mode remains active until the read array command (FF_{16}) is written.

The RY/$\overline{BY}$ status flag is "0" during block erase operation and "1" when the block erase operation is completed as is the status register bit 7.

After the block erase ends, erase results can be checked by reading the status register. For details, refer to the section where the status register is detailed.

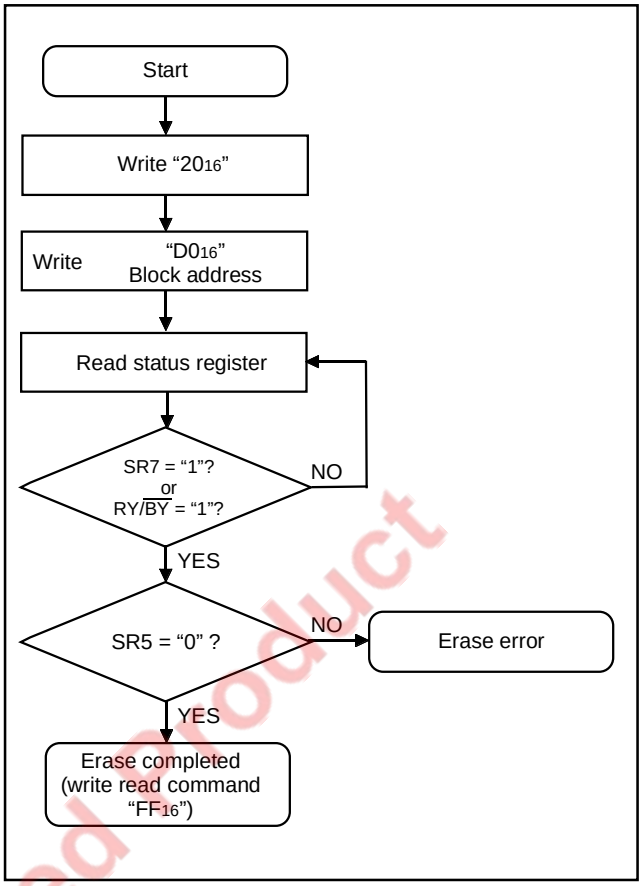


Fig. 83 Erase flowchart

● Status Register

The status register shows the operating status of the flash memory and whether erase operations and programs ended successfully or in error. It can be read in the following ways:

(1) By reading an arbitrary address from the User ROM area after writing the read status register command (70_{16})
(2) By reading an arbitrary address from the User ROM area in the period from when the program starts or erase operation starts to when the read array command (FF_{16}) is input.

Also, the status register can be cleared by writing the clear status register command (50_{16}).
After reset, the status register is set to "80_{16}".
Table 16 shows the status register. Each bit in this register is explained below.

•Sequencer status (SR7)

The sequencer status indicates the operating status of the flash memory. This bit is set to "0" (busy) during write or erase operation and is set to "1" when these operations ends.
After power-on, the sequencer status is set to "1" (ready).

•Erase status (SR5)

The erase status indicates the operating status of erase operation. If an erase error occurs, it is set to "1". When the erase status is cleared, it is reset to "0".

•Program status (SR4)

The program status indicates the operating status of write operation. When a write error occurs, it is set to "1".
The program status is reset to "0" when it is cleared.

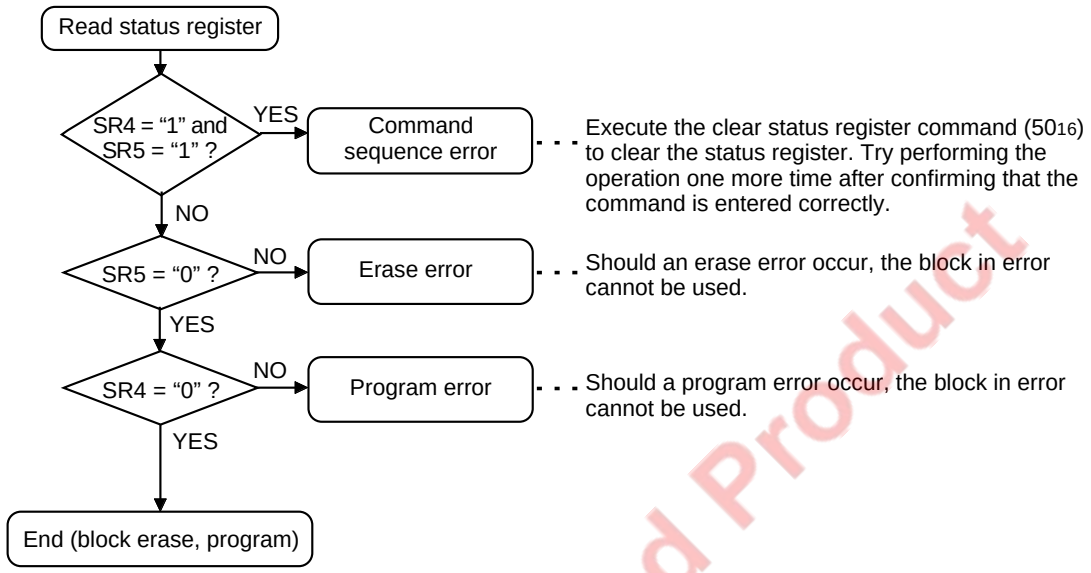
If "1" is written for any of the SR5 and SR4 bits, the read array, program, and block erase commands are not accepted. Before executing these commands, execute the clear status register command (50_{16}) and clear the status register.
Also, if any commands are not correct, both SR5 and SR4 are set to "1".

Table 16 Definition of each bit in status register

Each bit of SRD bits	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Sequencer status	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR3 (bit3)	Reserved	-	-
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

● Full Status Check

By performing full status check, it is possible to know the execution results of erase and program operations. Figure 84 shows a full status check flowchart and the action to be taken when each error occurs.



Note: When one of SR5 and SR4 is set to "1", none of the read array, program, and block erase commands is accepted. Execute the clear status register command (50_{16}) before executing these commands.

Fig. 84 Full status check flowchart and remedial procedure for errors

● Functions To Inhibit Rewriting Flash Memory Version

To prevent the contents of internal flash memory from being read out or rewritten easily, this MCU incorporates a ROM code protect function for use in parallel I/O mode and an ID code check function for use in standard serial I/O mode.

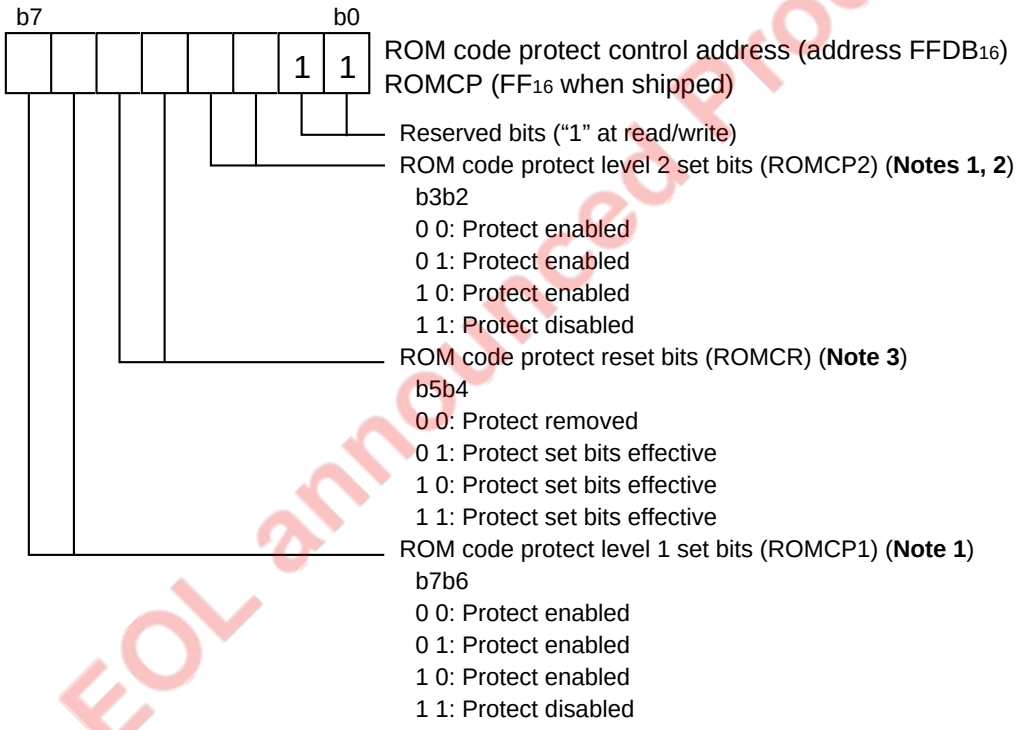
(1) ROM Code Protect Function

The ROM code protect function is the function to inhibit reading out or modifying the contents of internal flash memory by using the ROM code protect control address (address FFDB16) in parallel I/O mode. Figure 85 shows the ROM code protect control address (address FFDB16). (This address exists in the User ROM area.)

If one or both of the pair of ROM code protect bits is set to "0", the ROM code protect is turned on, so that the contents of internal flash memory are protected against readout and modification. The ROM code protect is implemented in two levels. If level 2 is selected, the flash memory is protected even against readout by a shipment inspection LSI tester, etc. When an attempt is made to select both level 1 and level 2, level 2 is selected by default.

If both of the two ROM code protect reset bits are set to "00", the ROM code protect is turned off, so that the contents of internal flash memory can be readout or modified. Once the ROM code protect is turned on, the contents of the ROM code protect reset bits cannot be modified in parallel I/O mode. Use the serial I/O or CPU rewrite mode to rewrite the contents of the ROM code protect reset bits.

Rewriting of only the ROM code protect control address (address FFDB16) cannot be performed. When rewriting the ROM code protect reset bit, rewrite the whole user ROM area (block 0) containing the ROM code protect control address.



Notes 1: When ROM code protect is turned on, the internal flash memory is protected against readout or modification in parallel I/O mode.
2: When ROM code protect level 2 is turned on, ROM code readout by a shipment inspection LSI tester, etc. also is inhibited.
3: The ROM code protect reset bits can be used to turn off ROM code protect level 1 and ROM code protect level 2. However, since these bits cannot be modified in parallel I/O mode, they need to be rewritten in serial I/O mode or CPU rewrite mode.

Fig. 85 Structure of ROM code protect control address

(2) ID Code Check Function

Use this function in standard serial I/O mode. When the contents of the flash memory are not blank, the ID code sent from the programmer is compared with the ID code written in the flash memory to see if they match. If the ID codes do not match, the commands sent from the programmer are not accepted. The ID code consists of 8-bit data, and its areas are FFD4$_{16}$ to FFDA$_{16}$. Write a program which has had the ID code preset at these addresses to the flash memory.

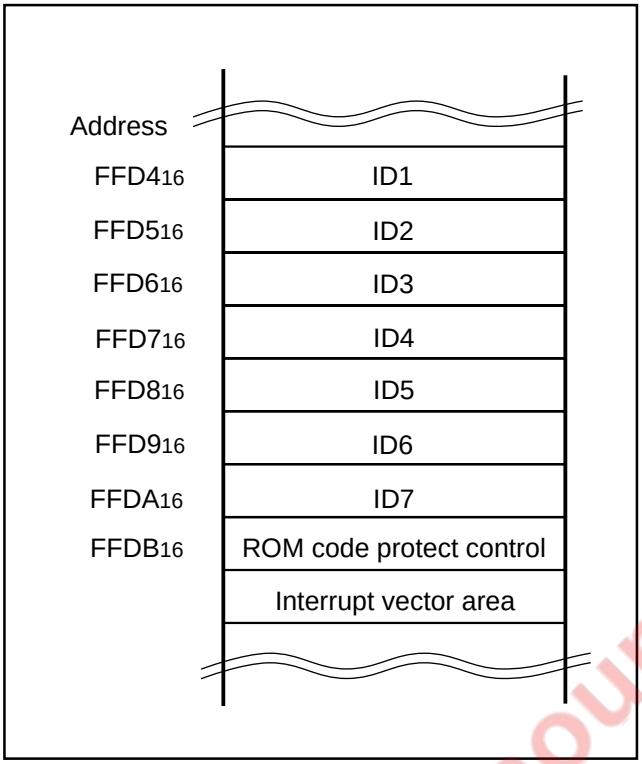


Fig. 86 ID code store addresses

● Parallel I/O Mode

The parallel I/O mode is used to input/output software commands, address and data in parallel for operation (read, program and erase) to internal flash memory.
Use the external device (writer) only for 3804 Group (spec. H). For details, refer to the user's manual of each writer manufacturer.

• User ROM and Boot ROM Areas

In parallel I/O mode, the User ROM and Boot ROM areas shown in Figure 77 can be rewritten. Both areas of flash memory can be operated on in the same way.
The Boot ROM area is 4 Kbytes in size and located at addresses $F000_{16}$ through $FFFF_{16}$. Make sure program and block erase operations are always performed within this address range. (Access to any location outside this address range is prohibited.)
In the Boot ROM area, an erase block operation is applied to only one 4 Kbyte block. The boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the fac-tory. Therefore, using the MCU in standard serial I/O mode, do not rewrite to the Boot ROM area.

● Standard serial I/O Mode

The standard serial I/O mode inputs and outputs the software commands, addresses and data needed to operate (read, program, erase, etc.) the internal flash memory. This I/O is clock synchronized serial. This mode requires a purpose-specific peripheral unit.

The standard serial I/O mode is different from the parallel I/O mode in that the CPU controls flash memory rewrite (uses the CPU rewrite mode), rewrite data input and so forth. The standard serial I/O mode is started by connecting “H” to the CNVss pin and “H” to the $P4_5$ (BOOTENT) pin, and releasing the reset operation. (In the ordinary microcomputer mode, set CNVss pin to “L” level.)

This control program is written in the Boot ROM area when the product is shipped from Renesas. Accordingly, make note of the fact that the standard serial I/O mode cannot be used if the Boot ROM area is rewritten in parallel I/O mode. The standard serial I/O mode has standard serial I/O mode 1 of the clock synchronous serial and standard serial I/O mode 2 of the clock asynchronous serial. Tables 17 and 18 show description of pin function (standard serial I/O mode). Figures 87 to 90 show the pin connections for the standard serial I/O mode.

In standard serial I/O mode, only the User ROM area shown in Figure 77 can be rewritten. The Boot ROM area cannot be written.

In standard serial I/O mode, a 7-byte ID code is used. When there is data in the flash memory, this function determines whether the ID code sent from the peripheral unit (programmer) and those written in the flash memory match. The commands sent from the peripheral unit (programmer) are not accepted unless the ID code matches.

Table 17 Description of pin function (Flash Memory Serial I/O Mode 1)

Pin name	Signal name	I/O	Function
VCC,VSS	Power supply	I	Apply 2.7 to 5.5 V to the Vcc pin and 0 V to the Vss pin.
CNVSS	CNVSS	I	After input of port is set, input “H” level.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. To reset the microcomputer, $\overline{\text{RESET}}$ pin should be held at an “L” level for 16 cycles or more of XIN.
XIN	Clock input	I	Connect an oscillation circuit between the XIN and XOUT pins.
XOUT	Clock output	O	As for the connection method, refer to the “clock generating circuit”.
AVSS	Analog power supply input		Connect AVss to Vss.
VREF	Reference voltage input	I	Apply reference voltage of A/D to this pin.
P00–P07,P10–P17, P20–P27,P30–P37, P40–P43,P50–P57, P60–P67	I/O port	I/O	Input “L” or “H” level, or keep open.
P44	RxD input	I	Serial data input pin.
P45	TxD output	O	Serial data output pin.
P46	SCLK input	I	Serial clock input pin.
P47	BUSY output	O	BUSY signal output pin.

Table 18 Description of pin function (Flash Memory Serial I/O Mode 2)

Pin name	Signal name	I/O	Function
VCC,VSS	Power supply	I	Apply 2.7 to 5.5 V to the Vcc pin and 0 V to the Vss pin.
CNVSS	CNVSS	I	After input of port is set, input “H” level.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. To reset the microcomputer, $\overline{\text{RESET}}$ pin should be held at an “L” level for 16 cycles or more of XIN.
XIN	Clock input	I	Connect an oscillation circuit between the XIN and XOUT pins.
XOUT	Clock output	O	As for the connection method, refer to the “clock generating circuit”.
AVss	Analog power supply input		Connect AVss to Vss.
VREF	Reference voltage input	I	Apply reference voltage of A/D to this pin.
P00–P07,P10–P17, P20–P27,P30–P37, P40–P43,P50–P57, P60–P67	I/O port	I/O	Input “L” or “H” level, or keep open.
P44	RxD input	I	Serial data input pin.
P45	TxD output	O	Serial data output pin.
P46	SCLK input	I	Input “L” level.
P47	BUSY output	O	BUSY signal output pin.

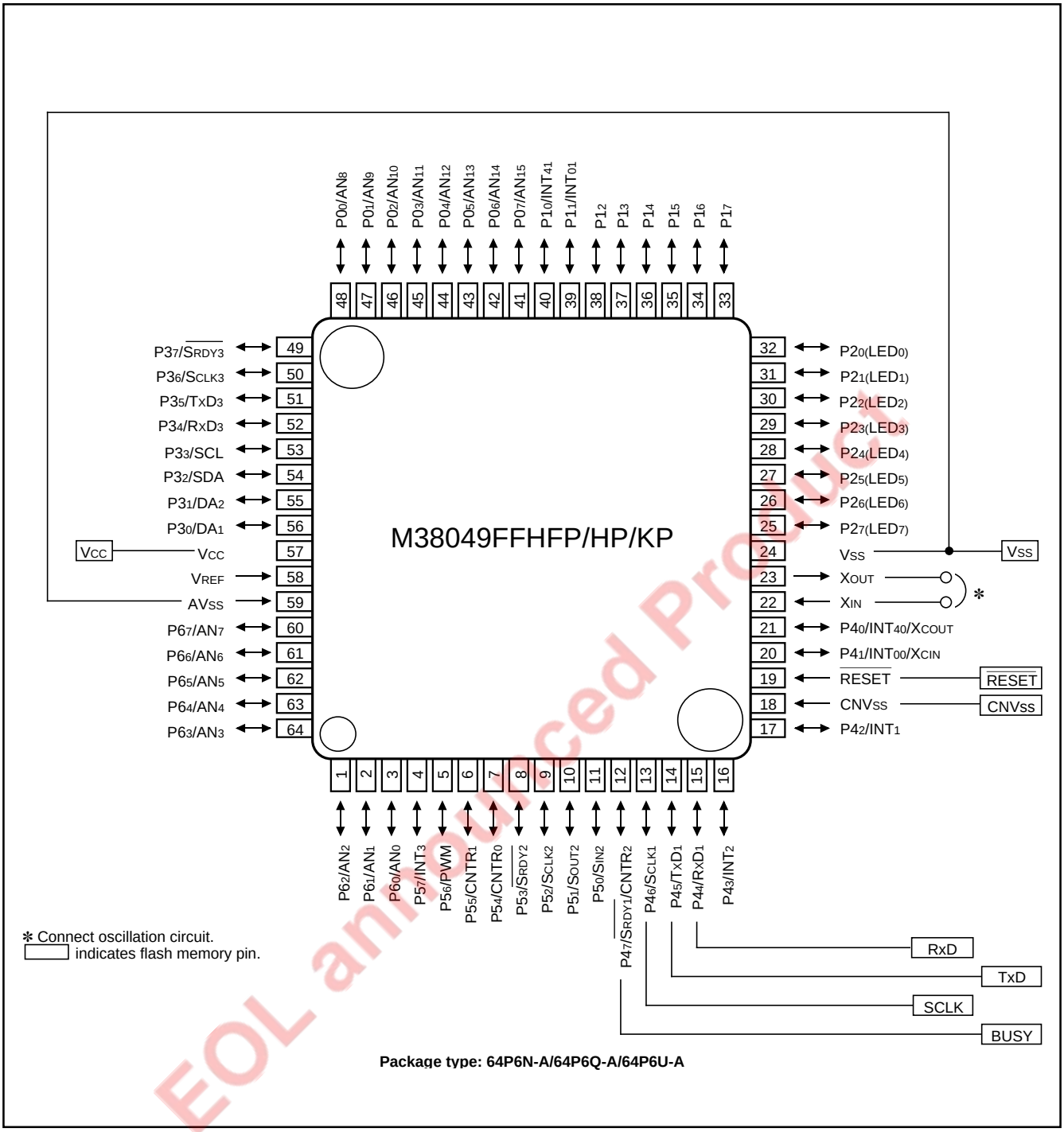


Fig. 87 Connection for standard serial I/O mode 1 (M38049FFHFP/HP/KP)

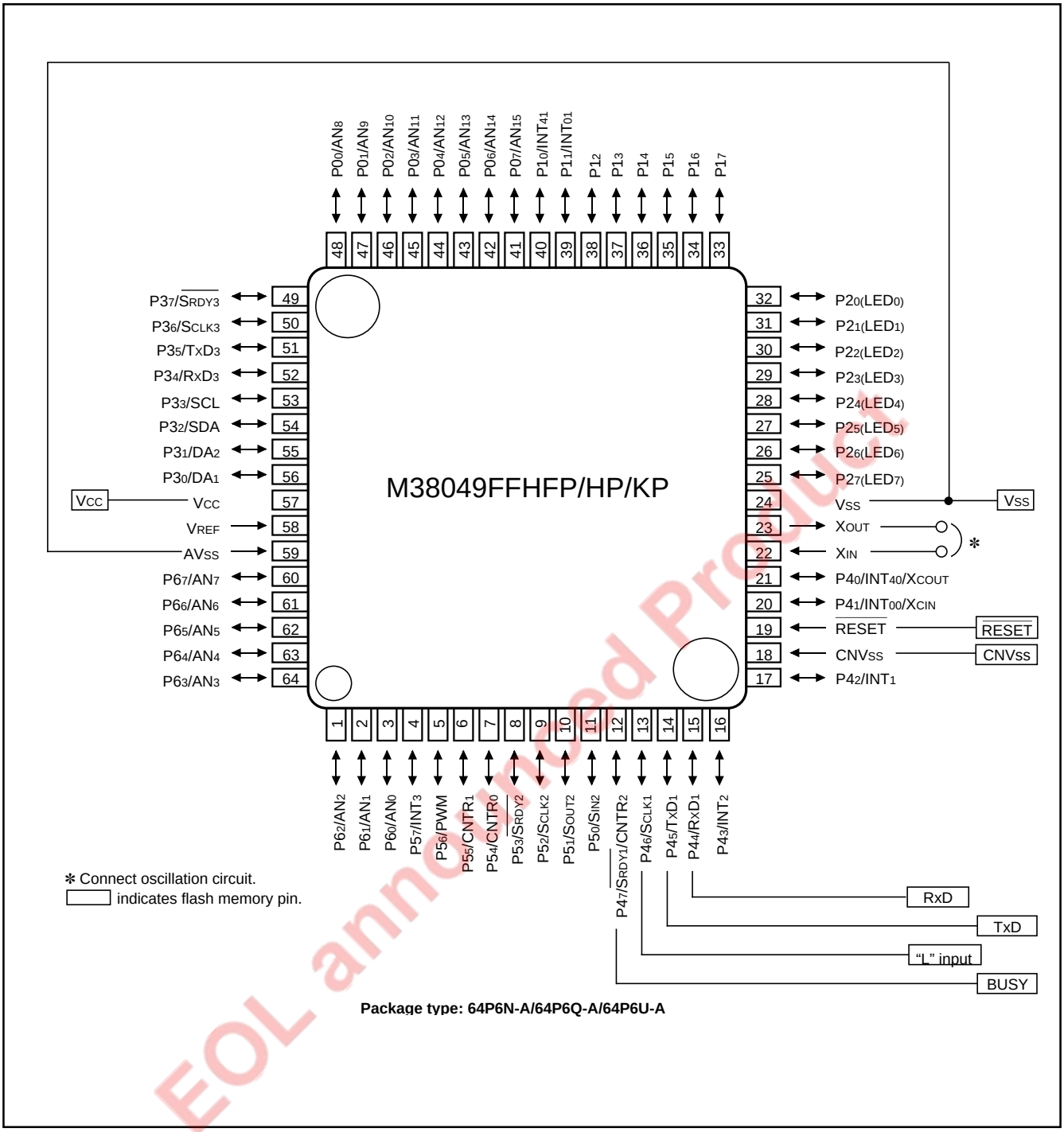


Fig. 88 Connection for standard serial I/O mode 2 (M38049FFHFP/HP/KP)

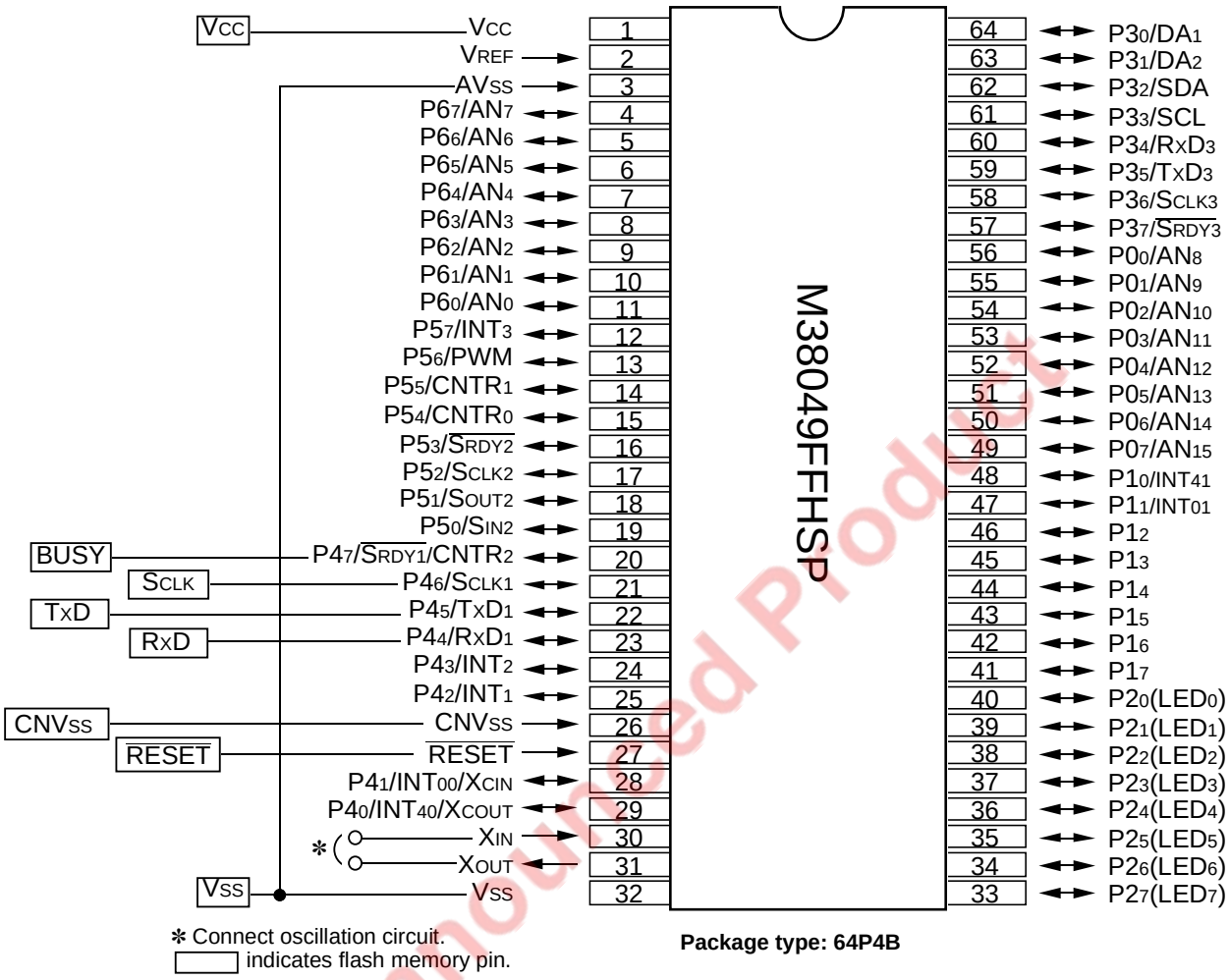


Fig. 89 Connection for standard serial I/O mode 1 (M38049FFHSP)

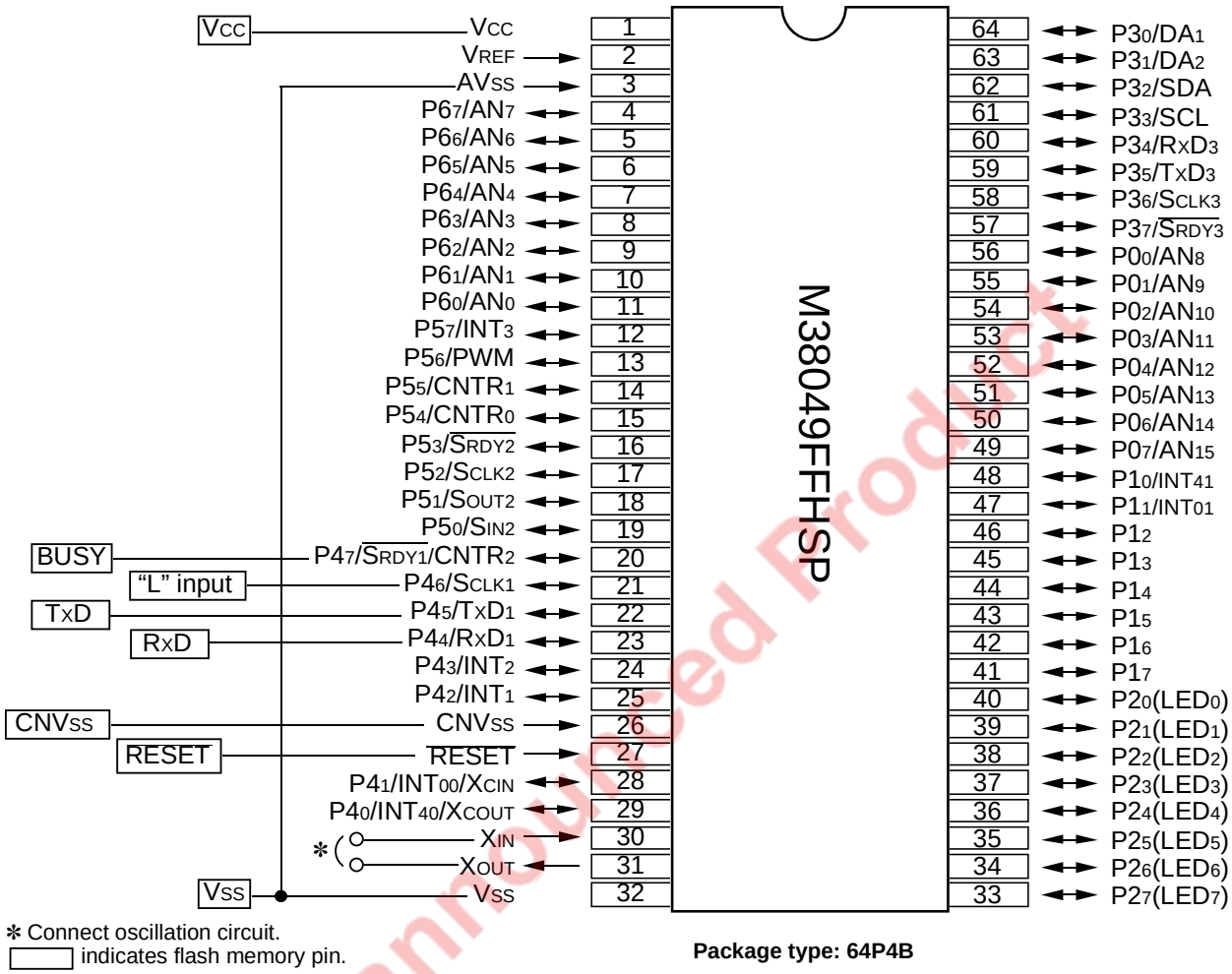
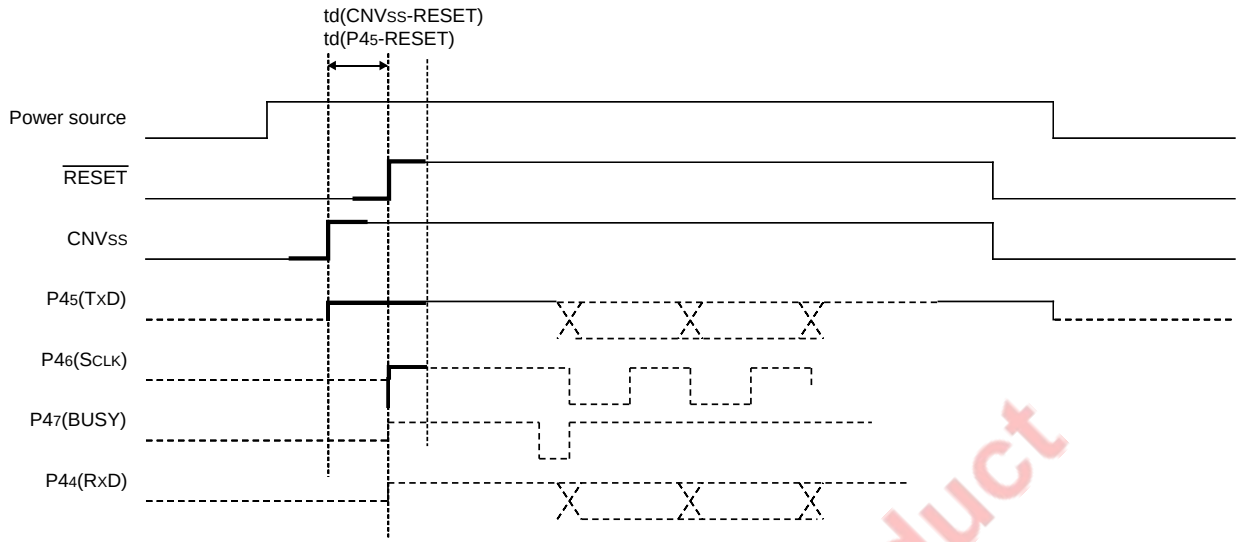


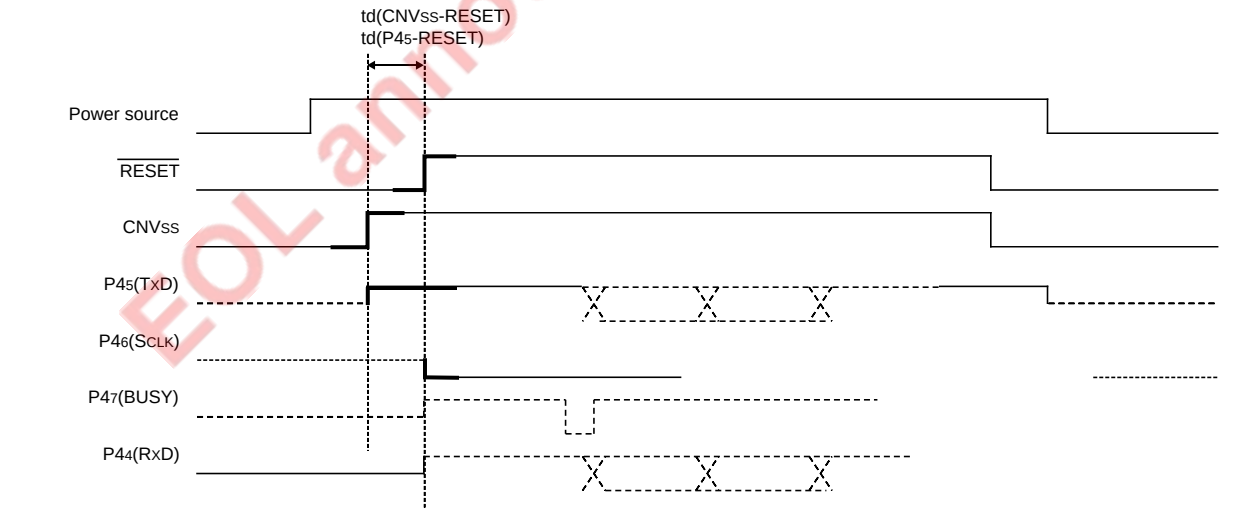
Fig. 90 Connection for standard serial I/O mode 2 (M38049FFHSP)



Symbol	Limits			Unit
	Min.	Typ.	Max.	
td(CNVss-RESET)	0	–	–	ms
td(P45-RESET)	0			ms

Notes: In the standard serial I/O mode 1, input "H" to the P46 pin.
Be sure to set the CNVss pin to "H" before rising RESET.
Be sure to set the P45 pin to "H" before rising RESET.

Fig. 91 Operating waveform for standard serial I/O mode 1



Symbol	Limits			Unit
	Min.	Typ.	Max.	
td(CNVss-RESET)	0	–	–	ms
td(P45-RESET)	0			ms

Notes: In the standard serial I/O mode 2, input "H" to the P46 pin.
Be sure to set the CNVss pin to "H" before rising RESET.
Be sure to set the P45 pin to "H" before rising RESET.

Fig. 92 Operating waveform for standard serial I/O mode 2

NOTES ON PROGRAMMING

Processor Status Register

The contents of the processor status register (PS) after a reset are undefined, except for the interrupt disable flag (I) which is "1." After a reset, initialize flags which affect program execution. In particular, it is essential to initialize the index X mode (T) and the decimal mode (D) flags because of their effect on calculations.

Interrupts

The contents of the interrupt request bits do not change immediately after they have been written. After writing to an interrupt request register, execute at least one instruction before performing a BBC or BBS instruction.

Decimal Calculations

- To calculate in decimal notation, set the decimal mode flag (D) to "1", then execute an ADC or SBC instruction. After executing an ADC or SBC instruction, execute at least one instruction before executing a SEC, CLC, or CLD instruction.
- In decimal mode, the values of the negative (N), overflow (V), and zero (Z) flags are invalid.

Timers

If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is 1/(n+1).

Multiplication and Division Instructions

- The index X mode (T) and the decimal mode (D) flags do not affect the MUL and DIV instruction.
- The execution of these instructions does not change the contents of the processor status register.

Ports

The contents of the port direction registers cannot be read. The following cannot be used:

- The data transfer instruction (LDA, etc.)
- The operation instruction when the index X mode flag (T) is "1"
- The instruction with the addressing mode which uses the value of a direction register as an index
- The bit-test instruction (BBC or BBS, etc.) to a direction register
- The read-modify-write instructions (ROR, CLB, or SEB, etc.) to a direction register.

Use instructions such as LDM and STA, etc., to set the port direction registers.

Serial Interface

In clock synchronous serial I/O, if the receive side is using an external clock and it is to output the $\overline{\text{SRDY}}$ signal, set the transmit enable bit, the receive enable bit, and the $\overline{\text{SRDY}}$ output enable bit to "1."

Serial I/O continues to output the final bit from the TxD pin after transmission is completed. SOUT2 pin for serial I/O2 goes to high impedance after transfer is completed.

When in serial I/Os 1 and 3 (clock-synchronous mode) or in serial I/O2, an external clock is used as synchronous clock, write transmission data to the transmit buffer register or serial I/O2 register, during transfer clock is "H."

A/D Converter

The comparator uses capacitive coupling amplifier whose charge will be lost if the clock frequency is too low.

Therefore, make sure that f(XIN) is at least on 500 kHz during an A/D conversion.

Do not execute the STP instruction during an A/D conversion.

D/A Converter

The accuracy of the D/A converter becomes rapidly poor under the V_{CC} = 4.0 V or less condition; a supply voltage of $V_{CC} \geq 4.0$ V is recommended. When a D/A converter is not used, set all values of D/Ai conversion registers (i=1, 2) to "00_{16}."

Instruction Execution Time

The instruction execution time is obtained by multiplying the period of the internal clock ϕ by the number of cycles needed to execute an instruction.

The number of cycles required to execute an instruction is shown in the list of machine instructions.

The period of the internal clock ϕ is double of the XIN period in high-speed mode.

NOTES ON USAGE

Handling of Power Source Pins

In order to avoid a latch-up occurrence, connect a capacitor suitable for high frequencies as bypass capacitor between power source pin (Vcc pin) and GND pin (Vss pin), and between power source pin (Vcc pin) and analog power source input pin (AVss pin). Besides, connect the capacitor to as close as possible. For bypass capacitor which should not be located too far from the pins to be connected, a ceramic capacitor of 0.01 µF–0.1 µF is recommended.

Power Source Voltage

When the power source voltage value of a microcomputer is less than the value which is indicated as the recommended operating conditions, the microcomputer does not operate normally and may perform unstable operation.
In a system where the power source voltage drops slowly when the power source voltage drops or the power supply is turned off, reset a microcomputer when the power source voltage is less than the recommended operating conditions and design a system not to cause errors to the system by this unstable operation.

Flash Memory Version

The CNVss pin determines the flash memory mode. To improve the noise reduction, connect a track between CNVss pin and Vss pin or Vcc pin with 1 to 10 kΩ resistance. The mask ROM version track of CNVss pin has no operational interference even if it is connected to Vss pin or Vcc pin via a resistor.

Electric Characteristic Differences Between Mask ROM and Flash Memory Version MCUs

There are differences in electric characteristics, operation margin, noise immunity, and noise radiation between Mask ROM and Flash Memory version MCUs due to the difference in the manufacturing processes, built-in ROM, and layout pattern etc.When manufacturing an application system with the Flash Memory version and then switching to use of the Mask ROM version, please conduct evaluations equivalent to the system evaluations conducted for the flash memory version.

DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

1. Mask ROM Confirmation Form *
2. Mark Specification Form *
3. Data to be written to ROM, in EPROM form (three identical copies)

* For the mask ROM confirmation and the mark specifications, refer to the “Renesas Technology Corp.” Homepage (http://www.renesas.com/en/rom).

FUNCTIONAL DESCRIPTION SUPPLEMENT

Interrupt

The 3804 group (Spec. H) permits interrupts on the basis of 16 sources. It is vector interrupts with a fixed priority system. Accordingly, when two or more interrupt requests occur during the same sampling, the higher-priority interrupt is accepted first. This priority is determined by hardware, but variety of priority processing can be performed by software, using an interrupt enable bit and an interrupt disable flag.

For interrupt sources, vector addresses and interrupt priority, refer to “Table 19”.

Table 19 Interrupt sources, vector addresses and priority

Interrupt Source	Priority	Vector Addresses (Note 1)		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset (Note 2)	1	FFFD16	FFFC16	At reset	Non-maskable
INT0	2	FFFB16	FFFA16	At detection of either rising or falling edge of INT0 input	External interrupt (active edge selectable)
Timer Z				At timer Z underflow	
INT1	3	FFF916	FFF816	At detection of either rising or falling edge of INT1 input	External interrupt (active edge selectable)
Serial I/O1 reception	4	FFF716	FFF616	At completion of serial I/O1 data reception	Valid when serial I/O1 is selected
Serial I/O1 transmission	5	FFF516	FFF416	At completion of serial I/O1 transmission shift or when transmission buffer is empty	Valid when serial I/O1 is selected
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
Timer X	6	FFF316	FFF216	At timer X underflow	
Timer Y	7	FFF116	FFF016	At timer Y underflow	
Timer 1	8	FFEF16	FFEE16	At timer 1 underflow	STP release timer underflow
Timer 2	9	FFED16	FFEC16	At timer 2 underflow	
CNTR0	10	FFEB16	FFEA16	At detection of either rising or falling edge of CNTR0 input	External interrupt (active edge selectable)
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
CNTR1	11	FFE916	FFE816	At detection of either rising or falling edge of CNTR1 input	External interrupt (active edge selectable)
Serial I/O3 reception				At completion of serial I/O3 data reception	Valid when serial I/O3 is selected
Serial I/O2	12	FFE716	FFE616	At completion of serial I/O2 data transmission or reception	Valid when serial I/O2 is selected
Timer Z				At timer Z underflow	
INT2	13	FFE516	FFE416	At detection of either rising or falling edge of INT2 input	External interrupt (active edge selectable)
I^2C				At completion of data transfer	
INT3	14	FFE316	FFE216	At detection of either rising or falling edge of INT3 input	External interrupt (active edge selectable)
INT4	15	FFE116	FFE016	At detection of either rising or falling edge of INT4 input	External interrupt (active edge selectable)
CNTR2				At detection of either rising or falling edge of CNTR2 input	External interrupt (active edge selectable)
A/D converter	16	FFDF16	FFDE16	At completion of A/D conversion	
Serial I/O3 transmission				At completion of serial I/O3 transmission shift or when transmission buffer is empty	Valid when serial I/O3 is selected
BRK instruction	17	FFDD16	FFDC16	At BRK instruction execution	Non-maskable software interrupt

Notes 1: Vector addresses contain interrupt jump destination addresses.
2: Reset function in the same way as an interrupt with the highest priority.

Timing After Interrupt

The interrupt processing routine begins with the machine cycle following the completion of the instruction that is currently in execution.

Figure 93 shows a timing chart after an interrupt occurs, and Figure 94 shows the time up to execution of the interrupt processing routine.

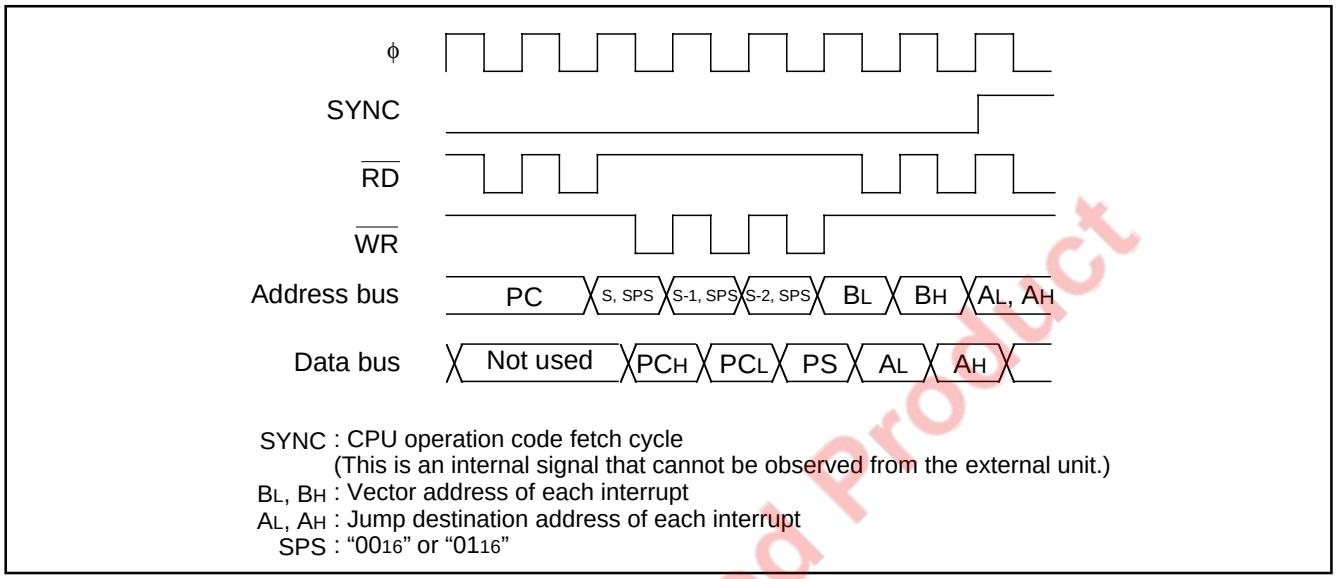


Fig. 93 Timing chart after an interrupt occurs

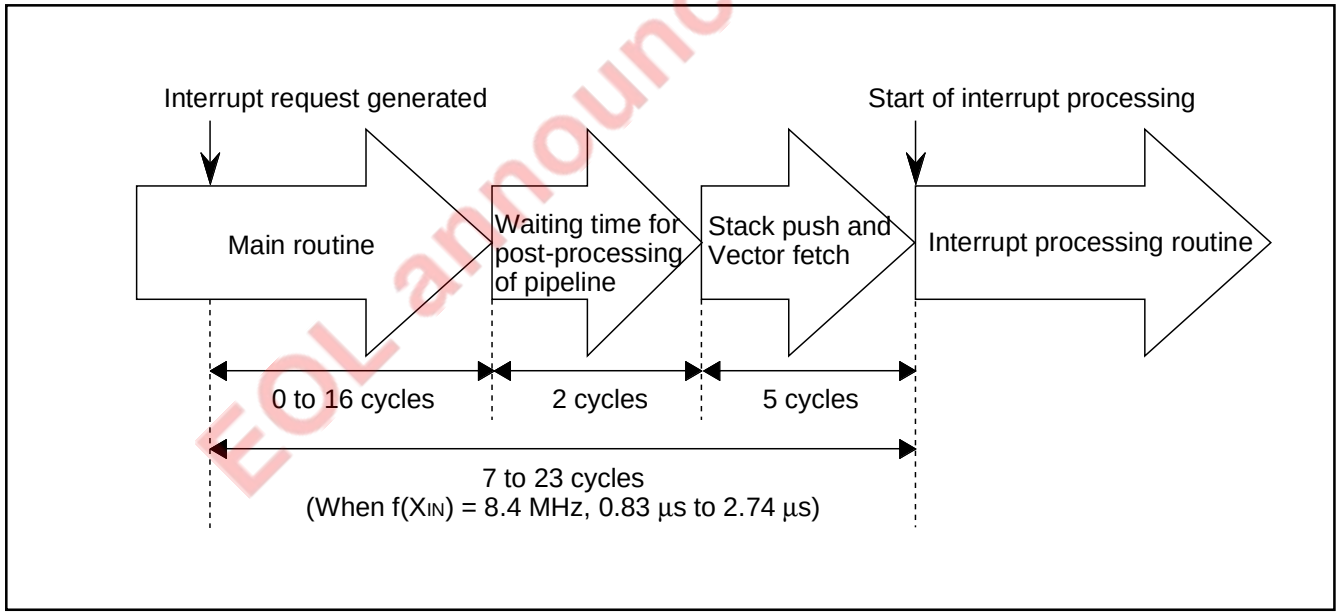


Fig. 94 Time up to execution of the interrupt processing routine

A/D Converter

A/D conversion is started by setting AD conversion completion bit to “0”. During A/D conversion, internal operations are performed as follows.

1. After the start of A/D conversion, AD conversion register goes to “00_{16}”.
2. The highest-order bit of AD conversion register is set to “1”. and the comparison voltage Vref is input to the comparator. Then, Vref is compared with analog input voltage V_{IN}.
3. As a result of comparison, when Vref < V_{IN}, the highest-order bit of AD conversion register becomes “1.” When Vref > V_{IN}, the highest-order bit becomes “0.”

By repeating the above operations up to the lowest-order bit of the AD conversion register, an analog value converts into a digital value.

In the 10-bit A/D mode, A/D conversion completes at 61 cycles of 2tc(X_{IN})* (15.25 µs at f(X_{IN}) = 8.0 MHz) after it is started. In the 8-bit A/D mode, A/D conversion completes at 50 cycles of 2tc(X_{IN}) (12.5 µs at f(X_{IN}) = 8.0 MHz) after it is started. And the result of the conversion is stored into the AD conversion register.

Concurrently with the completion of A/D conversion, the A/D conversion completion bit is set to “1” and an A/D conversion interrupt request occurs, so that the AD conversion interrupt request bit is set to “1”.

* tc(X_{IN}) = Main clock input cycle time

Table 20 Relative formula for a reference voltage V_{REF} of A/D converter and Vref (at 10-bit A/D mode)

When n = 0	Vref = 0
When n = 1 to 1023	$Vref = \frac{V_{REF}}{1024} \times n$

n : Value of A/D converter (decimal numeral)

Table 21 Relative formula for a reference voltage V_{REF} of A/D converter and Vref (at 8-bit A/D mode)

When n = 0	Vref = 0
When n = 1 to 255	$Vref = \frac{V_{REF}}{256} \times (n - 0.5)$

n : Value of A/D converter (decimal numeral)

Table 22 Change of AD conversion register during A/D conversion (at 10-bit A/D mode)

	Change of AD conversion register	Value of comparison voltage (Vref)
At start of conversion	0 0 0 0 0 0 0 0 0 0	0
First comparison	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$
Second comparison	*1 1 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4}$
Third comparison	*1 *2 1 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8}$
⋮	⋮	⋮
After completion of tenth comparison	A result of A/D conversion: *1 *2 *3 *4 *5 *6 *7 *8 *9 *10	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \bullet\bullet\bullet \pm \frac{V_{REF}}{1024}$

✻1–✻10: A result of the first to tenth comparison

Table 23 Change of AD conversion register during A/D conversion (at 8-bit A/D mode)

	Change of AD conversion register	Value of comparison voltage (Vref)
At start of conversion	0 0 0 0 0 0 0 0	0
First comparison	1 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{512}$
Second comparison	*1 1 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{512}$
Third comparison	*1 *2 1 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{512}$
⋮	⋮	⋮
After completion of eighth comparison	A result of A/D conversion *1 *2 *3 *4 *5 *6 *7 *8	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \bullet\bullet\bullet \pm \frac{V_{REF}}{256} - \frac{V_{REF}}{512}$

*1–*8: A result of the first to eighth comparison

Figure 95 shows A/D conversion equivalent circuit, and Figure 96 shows A/D conversion timing chart.

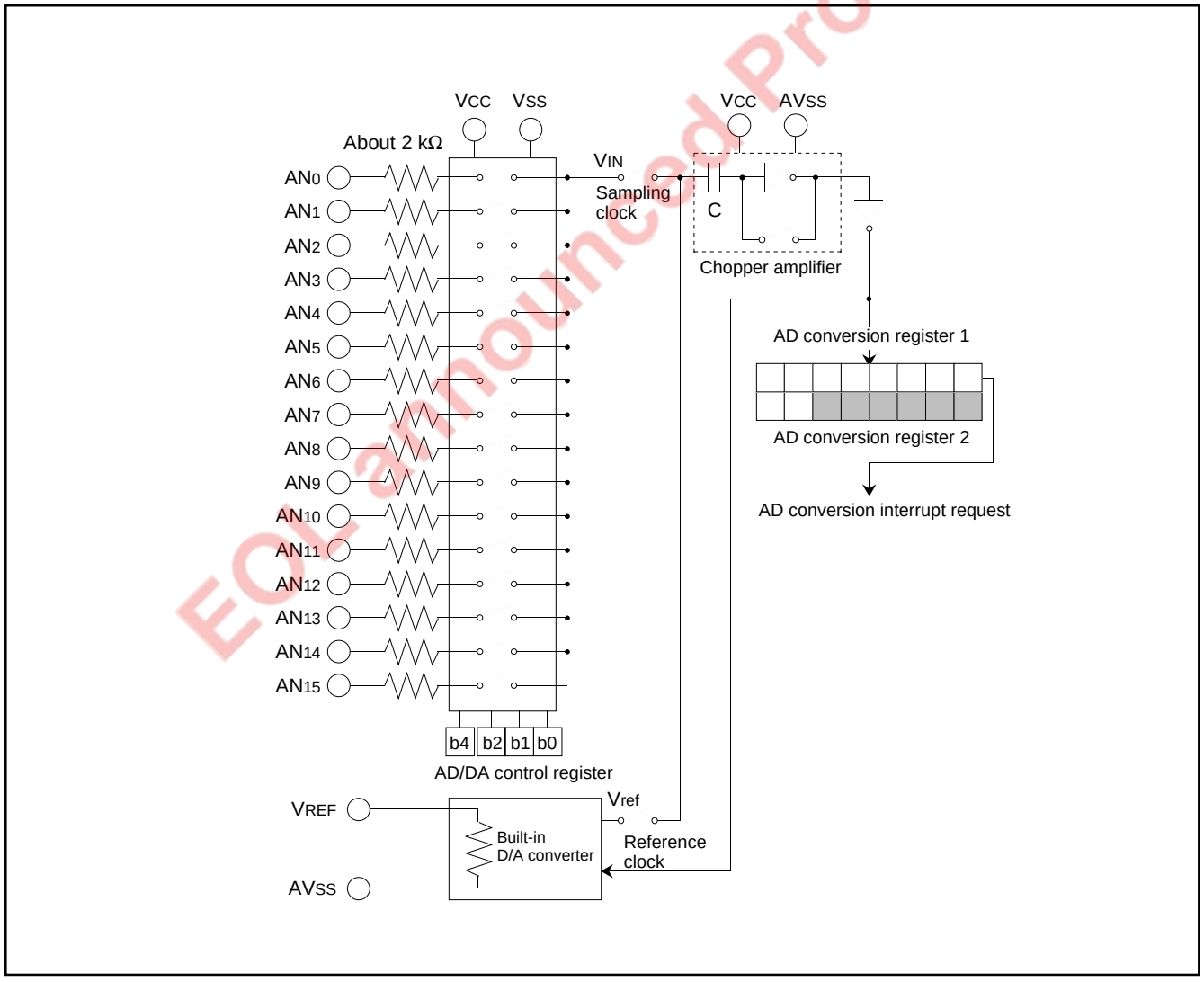


Fig. 95 A/D conversion equivalent circuit

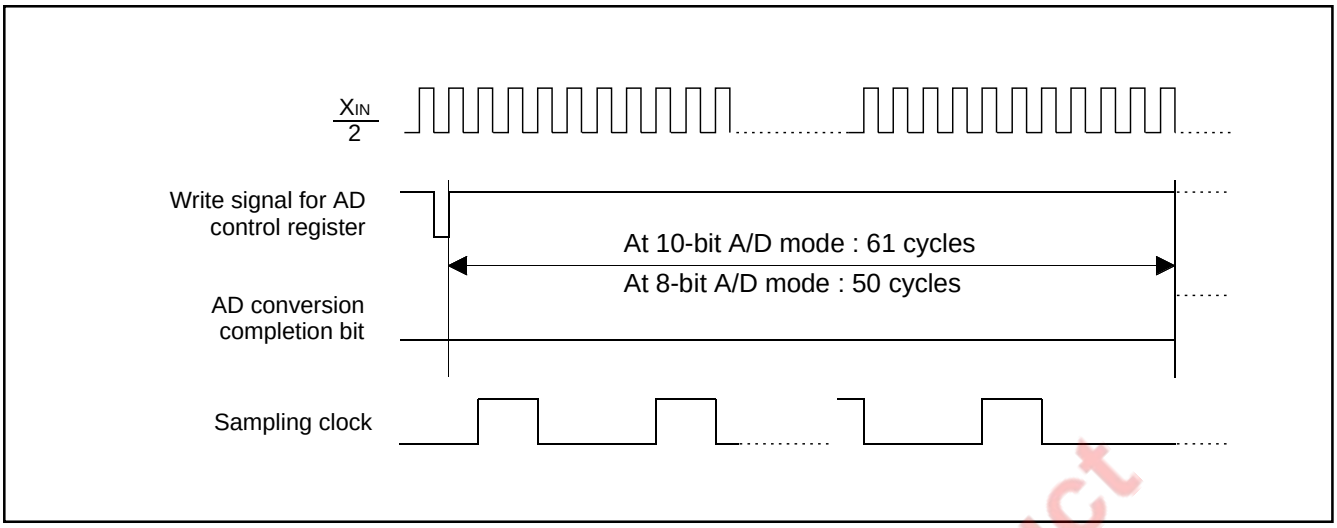


Fig. 96 A/D conversion timing chart

Memo

CHAPTER 2

APPLICATION

2.1 I/O port

This paragraph describes the setting method of I/O port relevant registers, notes etc.

2.1.1 Memory map

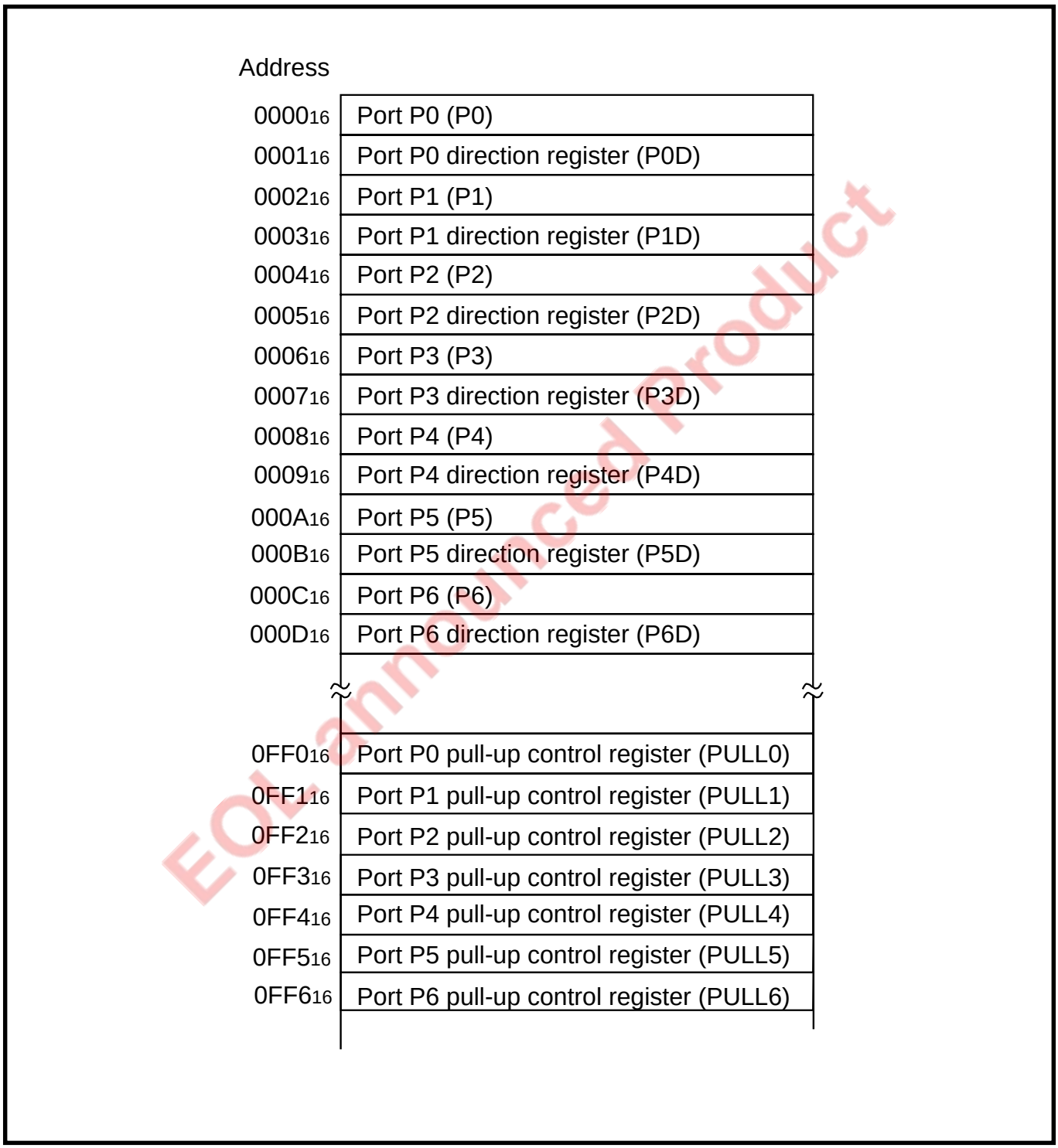


Fig. 2.1.1 Memory map of I/O port relevant registers

2.1.2 Relevant registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (i = 0, 1, 2, 3, 4, 5, 6)
(Pi: addresses 0000_{16}, 0002_{16}, 0004_{16}, 0006_{16}, 0008_{16}, $000A_{16}$, $000C_{16}$)

b	Name	Functions	At reset	R	W
0	Port P_{i0}	●In output mode Write •••••••• Port latch Read •••••••• Port latch ●In input mode Write •••••••• Port latch Read •••••••• Value of pin	0	○	○
1	Port P_{i1}		0	○	○
2	Port P_{i2}		0	○	○
3	Port P_{i3}		0	○	○
4	Port P_{i4}		0	○	○
5	Port P_{i5}		0	○	○
6	Port P_{i6}		0	○	○
7	Port P_{i7}		0	○	○

Fig. 2.1.2 Structure of Port Pi (i = 0 to 6)

Port Pi direction register

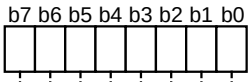
b7 b6 b5 b4 b3 b2 b1 b0

Port Pi direction register (i = 0, 1, 2, 3, 4, 5, 6)
(PiD: addresses 0001_{16}, 0003_{16}, 0005_{16}, 0007_{16}, 0009_{16}, $000B_{16}$, $000D_{16}$)

b	Name	Functions	At reset	R	W
0	Port Pi direction register	0 : Port P_{i0} input mode 1 : Port P_{i0} output mode	0	×	○
1		0 : Port P_{i1} input mode 1 : Port P_{i1} output mode	0	×	○
2		0 : Port P_{i2} input mode 1 : Port P_{i2} output mode	0	×	○
3		0 : Port P_{i3} input mode 1 : Port P_{i3} output mode	0	×	○
4		0 : Port P_{i4} input mode 1 : Port P_{i4} output mode	0	×	○
5		0 : Port P_{i5} input mode 1 : Port P_{i5} output mode	0	×	○
6		0 : Port P_{i6} input mode 1 : Port P_{i6} output mode	0	×	○
7		0 : Port P_{i7} input mode 1 : Port P_{i7} output mode	0	×	○

Fig. 2.1.3 Structure of Port Pi direction register (i = 0 to 6)

Port Pi pull-up control register (i = 0 to 2, 4 to 6)

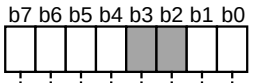


Port Pi pull-up control register (i = 0 to 2, 4 to 6)
(PULLi: addresses 0FF0$_{16}$, 0FF1$_{16}$, 0FF2$_{16}$, 0FF4$_{16}$, 0FF5$_{16}$, 0FF6$_{16}$)

b	Name	Functions	At reset	R	W
0	Port Pi$_0$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
1	Port Pi$_1$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
2	Port Pi$_2$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
3	Port Pi$_3$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
4	Port Pi$_4$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
5	Port Pi$_5$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
6	Port Pi$_6$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
7	Port Pi$_7$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○

Fig. 2.1.4 Structure of Port Pi pull-up control register (i = 0, 1, 2, 4, 5, 6)

Port P3 pull-up control register



Port P3 pull-up control register
(PULL3: address 0FF3$_{16}$)

b	Name	Functions	At reset	R	W
0	Port P3$_0$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
1	Port P3$_1$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
2	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
3			0	○	×
4	Port P3$_4$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
5	Port P3$_5$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
6	Port P3$_6$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
7	Port P3$_7$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○

Fig. 2.1.5 Structure of Port P3 pull-up control register

2.1.3 Port Pi pull-up control register

Valid/Invalid of pull-up resistor can be set by the pull-up control register by a bit unit. Pull-up control is valid only when each direction register is set to the input mode.

Note: Ports $P3_2$ and $P3_3$ do not have pull-up control bit because they are N-channel open-drain output.

2.1.4 Terminate unused pins

Table 2.1.1 Termination of unused pins (in single-chip mode)

Pins	Termination
P0, P1, P2, P3, P4, P5, P6	• Set to the input mode and connect each to V_{CC} or V_{SS} through a resistor of 1 kΩ to 10 kΩ. • Set to the output mode and open at “L” or “H” output state.
V_{REF}	Connect to Vss (GND).
AV_{SS}	Connect to Vss (GND).
X_{OUT}	Open (only when using external clock)

2.1.5 Notes on I/O port

(1) Notes in standby state

In standby state*1 for low-power dissipation, do not make input levels of an I/O port “undefined”, especially for I/O ports of the N-channel open-drain.

Pull-up (connect the port to VCC) or pull-down (connect the port to VSS) these ports through a resistor.

When determining a resistance value, note the following points:

- External circuit
- Variation of output levels during the ordinary operation

When using built-in pull-up resistor, note on varied current values:

- When setting as an input port : Fix its input level
- When setting as an output port : Prevent current from flowing out to external

● **Reason**

Even when setting as an output port with its direction register, when the content of the port latch is “1”, the transistor becomes the OFF state, which causes the ports to be the high-impedance state. Note that the level becomes “undefined” depending on external circuits.

Accordingly, the potential which is input to the input buffer in a microcomputer is unstable in the state that input levels of an I/O port are “undefined”. This may cause power source current.

✻1 standby state: stop mode by executing **STP** instruction
wait mode by executing **WIT** instruction

(2) Modifying output data with bit managing instruction

When the port latch of an I/O port is modified with the bit managing instruction*2, the value of the unspecified bit may be changed.

● **Reason**

The bit managing instructions are read-modify-write form instructions for reading and writing data by a byte unit. Accordingly, when these instructions are executed on a bit of the port latch of an I/O port, the following is executed to all bits of the port latch.

- As for bit which is set for input port:
  The pin state is read in the CPU, and is written to this bit after bit managing.
- As for bit which is set for output port:
  The bit value is read in the CPU, and is written to this bit after bit managing.

Note the following:

- Even when a port which is set as an output port is changed for an input port, its port latch holds the output data.
- As for a bit of which is set for an input port, its value may be changed even when not specified with a bit managing instruction in case where the pin state differs from its port latch contents.

✻2 Bit managing instructions: **SEB** and **CLB** instructions

2.1.6 Termination of unused pins

(1) Terminate unused pins

① I/O ports :
- Set the I/O ports for the input mode and connect them to Vcc or Vss through each resistor of 1 kΩ to 10 kΩ.
  Ports that permit the selecting of a built-in pull-up resistor can also use this resistor. Set the I/O ports for the output mode and open them at “L” or “H”.
- When opening them in the output mode, the input mode of the initial status remains until the mode of the ports is switched over to the output mode by the program after reset. Thus, the potential at these pins is undefined and the power source current may increase in the input mode. With regard to an effects on the system, thoroughly perform system evaluation on the user side.
- Since the direction register setup may be changed because of a program runaway or noise, set direction registers by program periodically to increase the reliability of program.

② The AVss pin when not using the A/D converter :
- When not using the A/D converter, handle a power source pin for the A/D converter, AVss pin as follows:
  AVss: Connect to the Vss pin.

(2) Termination remarks

① I/O ports :
Do not open in the input mode.
● **Reason**
- The power source current may increase depending on the first-stage circuit.
- An effect due to noise may be easily produced as compared with proper termination ① and shown on the above.

② I/O ports :
When setting for the input mode, do not connect to Vcc or Vss directly.
● **Reason**
If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between a port and Vcc (or Vss).

③ I/O ports :
When setting for the input mode, do not connect multiple ports in a lump to Vcc or Vss through a resistor.
● **Reason**
If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between ports.

- At the termination of unused pins, perform wiring at the shortest possible distance (20 mm or less) from microcomputer pins.

2.2 Interrupt

This paragraph explains the registers setting method and the notes relevant to the interrupt.

2.2.1 Memory map

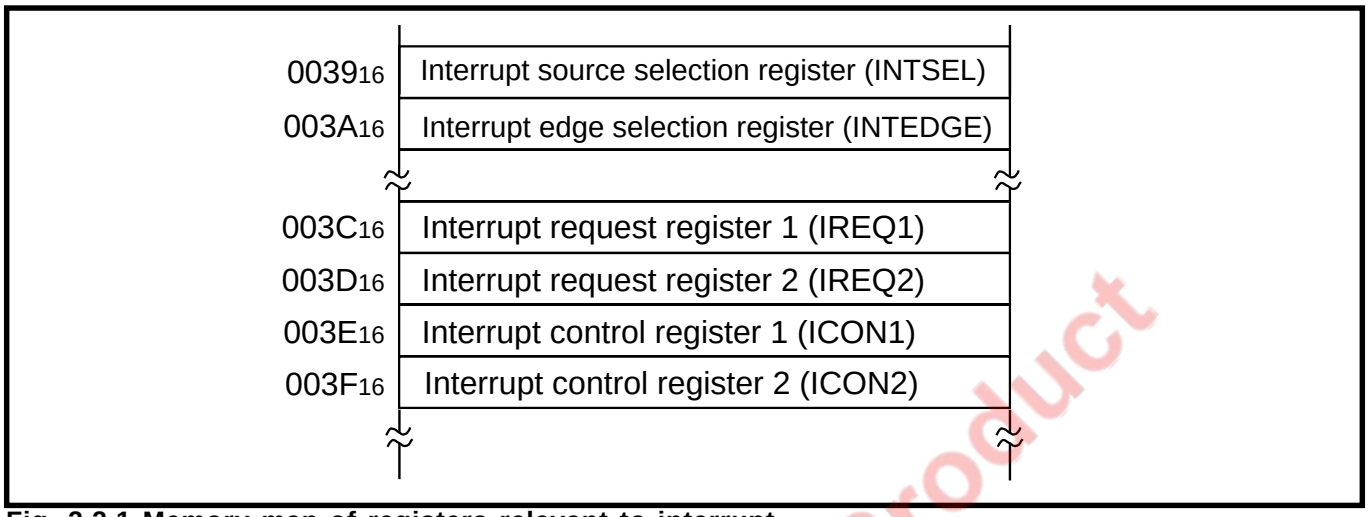


Fig. 2.2.1 Memory map of registers relevant to interrupt

2.2.2 Relevant registers

Interrupt source selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt source selection register (INTSEL: address 0039₁₆)

b	Name	Functions	At reset	R	W
0	INT_0/Timer Z interrupt source selection bit (*1)	0: INT_0 interrupt 1: Timer Z interrupt	0	○	○
1	Serial I/O2/Timer Z interrupt source selection bit (*1)	0: Serial I/O2 interrupt 1: Timer Z interrupt	0	○	○
2	Serial I/O1 transmit/ SCL, SDA interrupt source selection bit (*2)	0: Serial I/O1 transmit interrupt 1: SCL, SDA interrupt	0	○	○
3	$CNTR_0$/SCL, SDA interrupt source selection bit (*2)	0: $CNTR_0$ interrupt 1: SCL, SDA interrupt	0	○	○
4	INT_4/$CNTR_2$ interrupt source selection bit	0: INT_4 interrupt 1: $CNTR_2$ interrupt	0	○	○
5	INT_2/I^2C interrupt source selection bit	0: INT_2 interrupt 1: I^2C interrupt	0	○	○
6	$CNTR_1$/Serial I/O3 receive interrupt source selection bit	0: $CNTR_1$ interrupt 1: Serial I/O3 receive interrupt	0	○	○
7	AD converter/Serial I/O3 transmit interrupt source selection bit	0: A/D converter interrupt 1: Serial I/O3 transmit interrupt	0	○	○

*1: Do not write 1 to these bits simultaneously.
*2: Do not write 1 to these bits simultaneously.

Fig. 2.2.2 Structure of Interrupt source selection register

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt edge selection register
(INTEDGE: address 003A16)

b	Name	Functions	At reset	R	W
0	INT_0 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
1	INT_1 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
2	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are "0".		0	○	×
3	INT_2 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
4	INT_3 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
5	INT_4 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
6	INT_0, INT_4 interrupt switch bit	0: INT_{00}, INT_{40} interrupt 1: INT_{01}, INT_{41} interrupt	0	○	○
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are "0".		0	○	×

Fig. 2.2.3 Structure of Interrupt edge selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

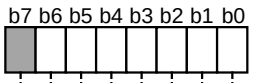
Interrupt request register 1
(IREQ1 : address 003C16)

b	Name	Functions	At reset	R	W
0	INT_0/Timer Z interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
1	INT_1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
2	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
3	Serial I/O1 transmit/SCL, SDA interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*

*: 0 can be set by software, but 1 cannot be set.

Fig. 2.2.4 Structure of Interrupt request register 1

Interrupt request register 2



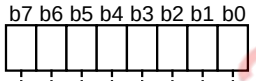
Interrupt request register 2
(IREQ2 : address 003D16)

b	Name	Functions	At reset	R	W
0	CNTR0/SCL, SDA interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	CNTR1/Serial I/O3 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O2/Timer Z interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT2/I2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	INT4/CNTR2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	AD converter/Serial I/O3 transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are 0 .		0	○	×

✻: 0 can be set by software, but 1 cannot be set.

Fig. 2.2.5 Structure of Interrupt request register 2

Interrupt control register 1



Interrupt control register 1
(ICON1 : address 003E16)

b	Name	Functions	At reset	R	W
0	INT0/Timer Z interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit/SCL, SDA interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.2.6 Structure of Interrupt control register 1

Interrupt control register 2

b7	b6	b5	b4	b3	b2	b1	b0
0							

Interrupt control register 2
(ICON2 : address 003F16)

b	Name	Functions	At reset	R	W
0	CNTR0/SCL, SDA interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	CNTR1/ Serial I/O3 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/ Timer Z interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT2/I2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT3 interrupt enable bit	0 : Interrupt disabled	0	○	○
5	INT4/CNTR2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/Serial I/O3 transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.2.7 Structure of Interrupt control register 2

2.2.3 Interrupt source

The 3804 group (Spec. H) 's interrupts are a type of vector and occur by 16 sources among 23 sources: nine external, thirteen internal, and one software. These are vector interrupts with a fixed priority system. Accordingly, when two or more interrupt requests occur during the same sampling, the higher-priority interrupt is accepted first. This priority is determined by hardware, but a variety of priority processing can be performed by software, using an interrupt enable bit and an interrupt disable flag.
For interrupt sources, vector addresses and interrupt priority, refer to Tables 2.2.1.

Table 2.2.1 Interrupt sources, vector addresses and priority

Interrupt Source	Priority	Vector Addresses (Note 1)		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset (Note 2)	1	FFFD16	FFFC16	At reset	Non-maskable
INT0	2	FFFB16	FFFA16	At detection of either rising or falling edge of INT0 input	External interrupt (active edge selectable)
Timer Z				At timer Z underflow	
INT1	3	FFF916	FFF816	At detection of either rising or falling edge of INT1 input	External interrupt (active edge selectable)
Serial I/O1 reception	4	FFF716	FFF616	At completion of serial I/O1 data reception	Valid when serial I/O1 is selected
Serial I/O1 transmission	5	FFF516	FFF416	At completion of serial I/O1 transmission shift or when transmission buffer is empty	Valid when serial I/O1 is selected
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
Timer X	6	FFF316	FFF216	At timer X underflow	
Timer Y	7	FFF116	FFF016	At timer Y underflow	
Timer 1	8	FFEF16	FFEE16	At timer 1 underflow	STP release timer underflow
Timer 2	9	FFED16	FFEC16	At timer 2 underflow	
CNTR0	10	FFEB16	FFEA16	At detection of either rising or falling edge of CNTR0 input	External interrupt (active edge selectable)
SCL, SDA				At detection of either rising or falling edge of SCL or SDA	External interrupt (active edge selectable)
CNTR1	11	FFE916	FFE816	At detection of either rising or falling edge of CNTR1 input	External interrupt (active edge selectable)
Serial I/O3 reception				At completion of serial I/O3 data reception	Valid when serial I/O3 is selected
Serial I/O2	12	FFE716	FFE616	At completion of serial I/O2 data transmission or reception	Valid when serial I/O2 is selected
Timer Z				At timer Z underflow	
INT2	13	FFE516	FFE416	At detection of either rising or falling edge of INT2 input	External interrupt (active edge selectable)
I^2C				At completion of data transfer	
INT3	14	FFE316	FFE216	At detection of either rising or falling edge of INT3 input	External interrupt (active edge selectable)
INT4	15	FFE116	FFE016	At detection of either rising or falling edge of INT4 input	External interrupt (active edge selectable)
CNTR2				At detection of either rising or falling edge of CNTR2 input	External interrupt (active edge selectable)
A/D converter	16	FFDF16	FFDE16	At completion of A/D conversion	
Serial I/O3 transmission				At completion of serial I/O3 transmission shift or when transmission buffer is empty	Valid when serial I/O3 is selected
BRK instruction	17	FFDD16	FFDC16	At BRK instruction execution	Non-maskable software interrupt

Notes 1: Vector addresses contain interrupt jump destination addresses.
2: Reset function in the same way as an interrupt with the highest priority.

2.2.4 Interrupt operation

When an interrupt request is accepted, the contents of the following registers just before acceptance of the interrupt requests are automatically pushed onto the stack area in the order of ①, ② and ③.

①High-order contents of program counter (PC_H)
②Low-order contents of program counter (PC_L)
③Contents of processor status register (PS)

After the contents of the above registers are pushed onto the stack area, the accepted interrupt vector address enters the program counter and consequently the interrupt processing routine is executed.
When the RTI instruction is executed at the end of the interrupt processing routine, the contents of the above registers pushed onto the stack area are restored to the respective registers in the order of ③, ② and ①; and the microcomputer resumes the processing executed just before acceptance of the interrupts.
Figure 2.2.8 shows an interrupt operation diagram.

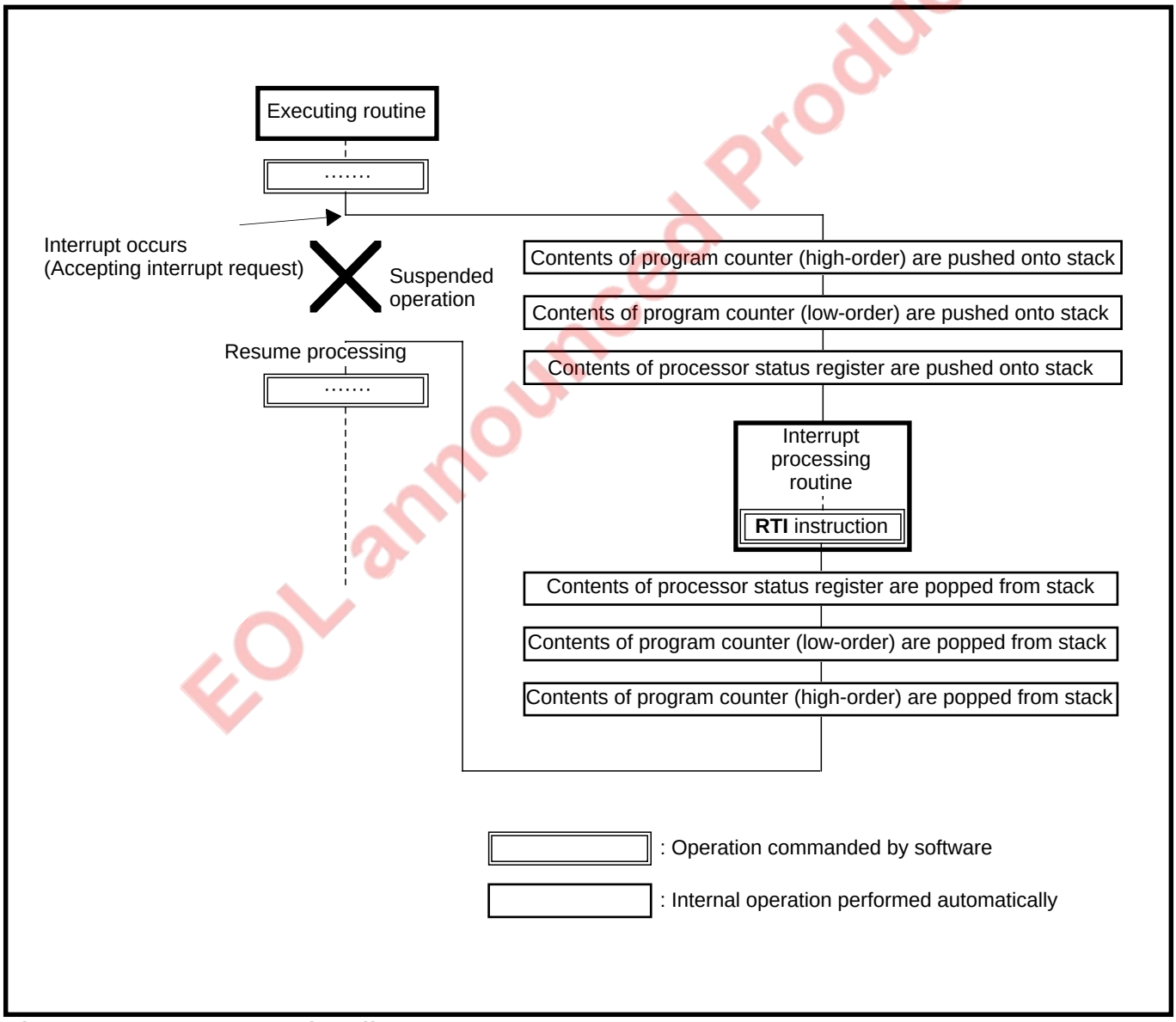


Fig. 2.2.8 Interrupt operation diagram

(1) Processing upon acceptance of interrupt request

Upon acceptance of an interrupt request, the following operations are automatically performed.

①The processing being executed is stopped.

②The contents of the program counter and the processor status register are pushed onto the stack area. Figure 2.2.9 shows the changes of the stack pointer and the program counter upon acceptance of an interrupt request.

③Concurrently with the push operation, the jump destination address (the beginning address of the interrupt processing routine) of the occurring interrupt stored in the vector address is set in the program counter, then the interrupt processing routine is executed.

④After the interrupt processing routine is started, the corresponding interrupt request bit is automatically cleared to “0”. The interrupt disable flag is set to “1” so that multiple interrupts are disabled.

Accordingly, for executing the interrupt processing routine, it is necessary to set the jump destination address in the vector area corresponding to each interrupt.

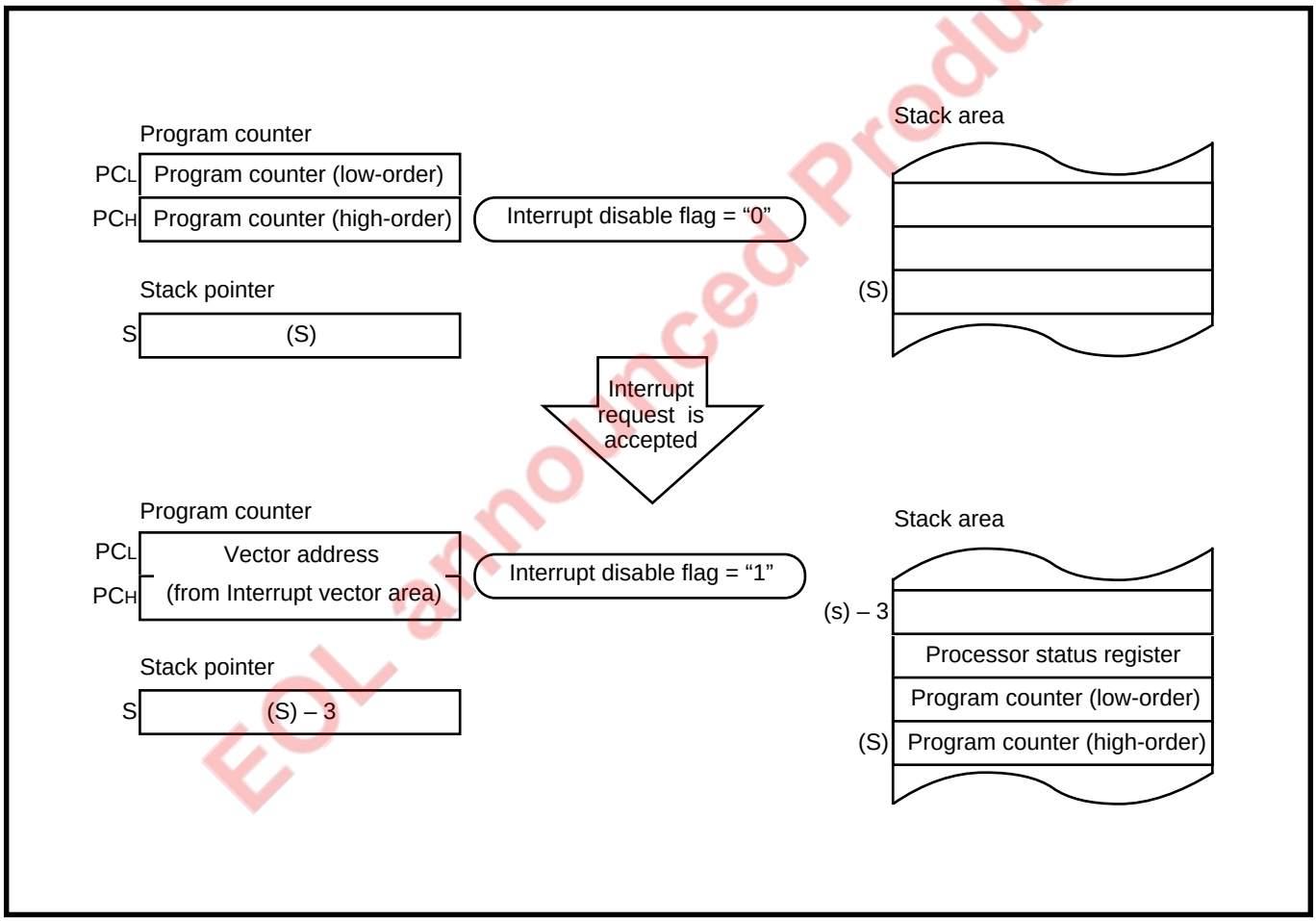


Fig. 2.2.9 Changes of stack pointer and program counter upon acceptance of interrupt request

(2) Timing after acceptance of interrupt request

The interrupt processing routine begins with the machine cycle following the completion of the instruction that is currently being executed.

Figure 2.2.10 shows the time up to execution of interrupt processing routine and Figure 2.2.11 shows the timing chart after acceptance of interrupt request.

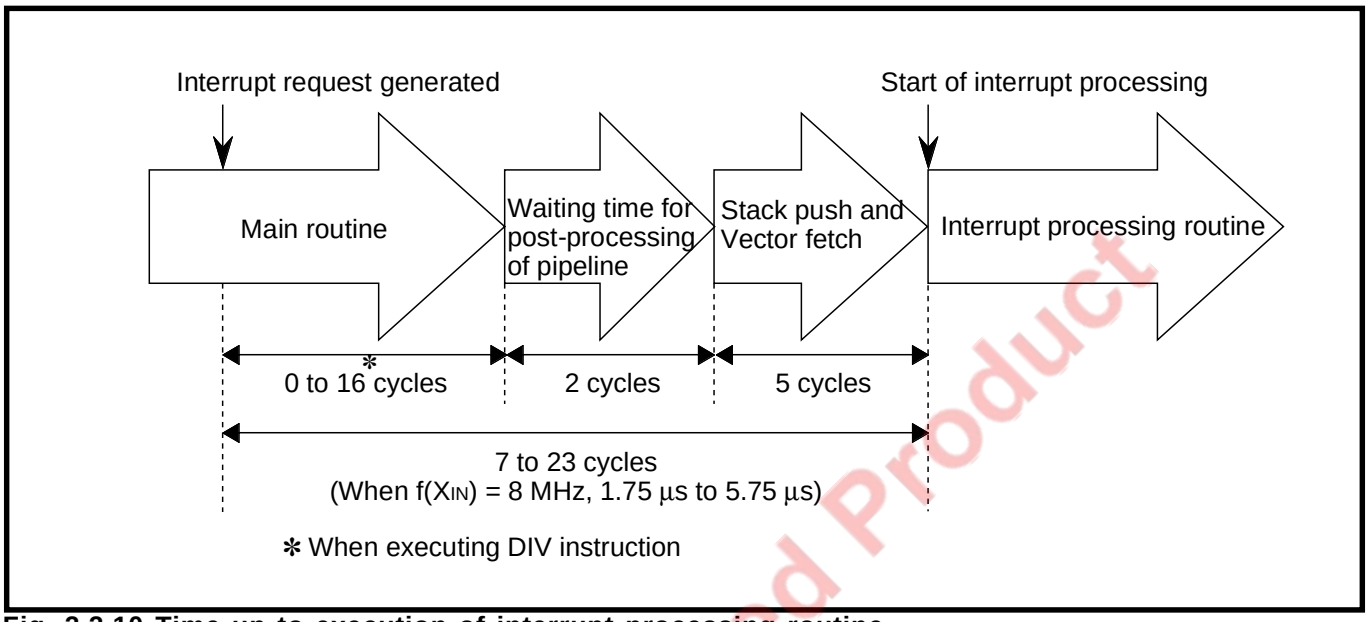


Fig. 2.2.10 Time up to execution of interrupt processing routine

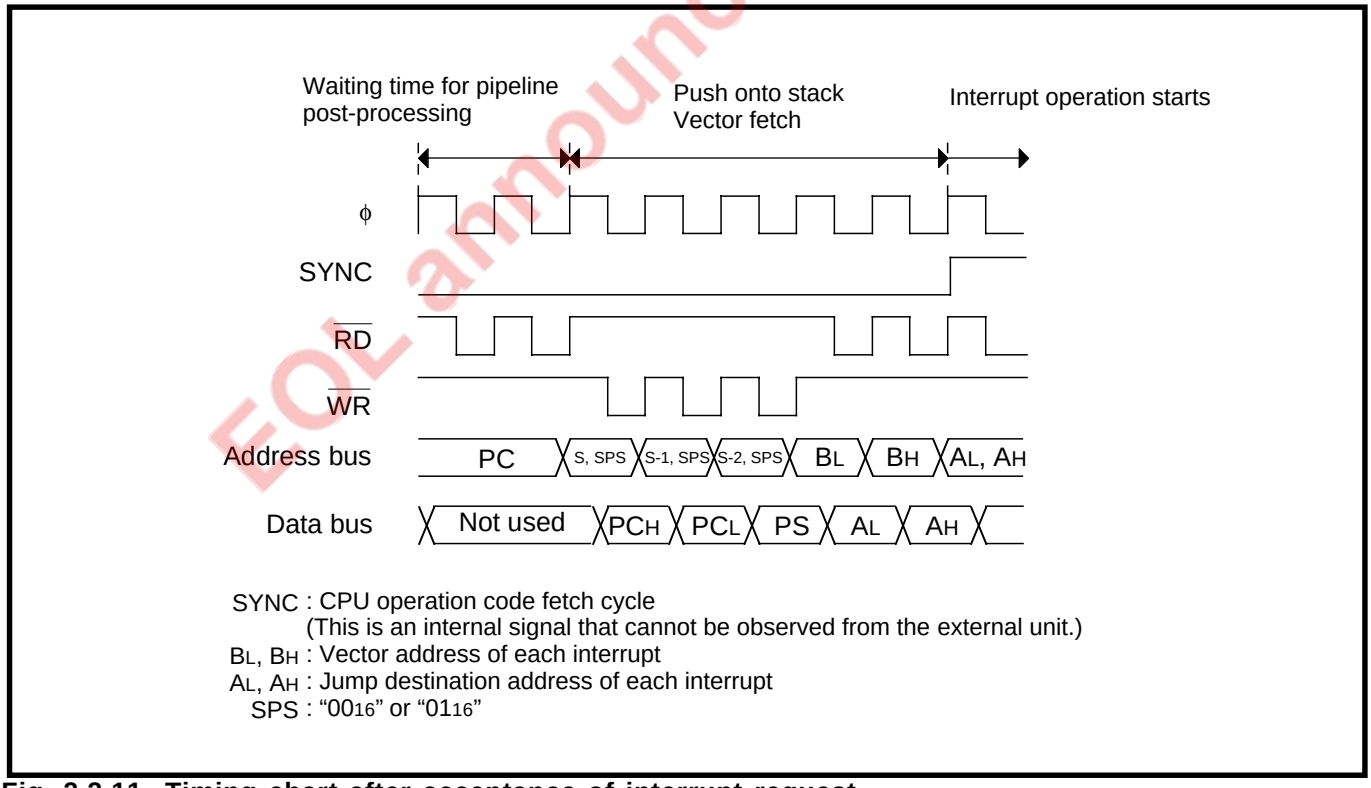


Fig. 2.2.11 Timing chart after acceptance of interrupt request

2.2.5 Interrupt control

The acceptance of all interrupts, excluding the BRK instruction interrupt, can be controlled by the interrupt request bit, interrupt enable bit, and an interrupt disable flag, as described in detail below. Figure 2.2.12 shows an interrupt control diagram.

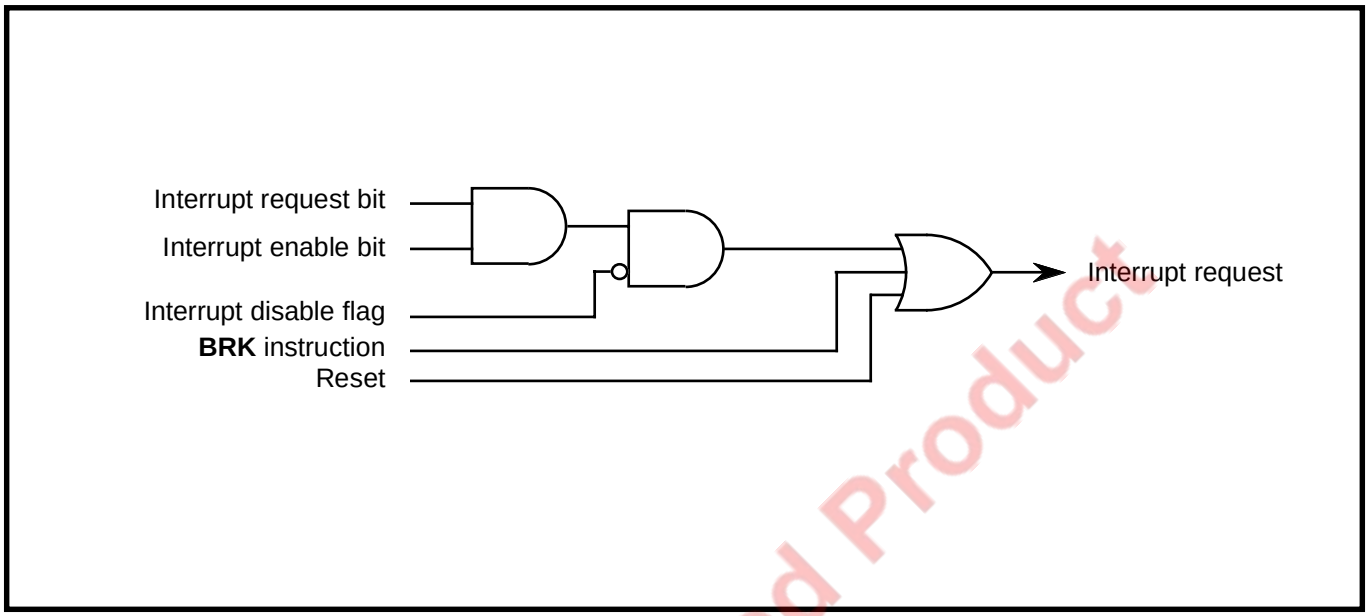


Fig. 2.2.12 Interrupt control diagram

The interrupt request bit, interrupt enable bit and interrupt disable flag function independently and do not affect each other. An interrupt is accepted when all the following conditions are satisfied.

- Interrupt request bit ..... "1"
- Interrupt enable bit ..... "1"
- Interrupt disable flag ..... "0"

Though the interrupt priority is determined by hardware, a variety of priority processing can be performed by software using the above bits and flag. Tables 2.2.2 shows list of interrupt control bits according to the interrupt source.

(1) Interrupt request bits

The interrupt request bits are allocated to the interrupt request register 1 (address $003C_{16}$) and interrupt request register 2 (address $003D_{16}$).

The occurrence of an interrupt request causes the corresponding interrupt request bit to be set to "1". The interrupt request bit is held in the "1" state until the interrupt is accepted. When the interrupt is accepted, this bit is automatically cleared to "0".

Each interrupt request bit can be set to "0", but cannot be set to "1", by software.

(2) Interrupt enable bits

The interrupt enable bits are allocated to the interrupt control register 1 (address $003E_{16}$) and the interrupt control register 2 (address $003F_{16}$).

The interrupt enable bits control the acceptance of the corresponding interrupt request.

When an interrupt enable bit is "0", the corresponding interrupt request is disabled. If an interrupt request occurs when this bit is "0", the corresponding interrupt request bit is set to "1" but the interrupt is not accepted. In this case, unless the interrupt request bit is set to "0" by software, the interrupt request bit remains in the "1" state.

When an interrupt enable bit is "1", the corresponding interrupt is enabled. If an interrupt request occurs when this bit is "1", the interrupt is accepted (when interrupt disable flag = "0").

Each interrupt enable bit can be set to "0" or "1" by software.

(3) Interrupt disable flag

The interrupt disable flag is allocated to bit 2 of the processor status register. The interrupt disable flag controls the acceptance of interrupt request except BRK instruction.

When this flag is “1”, the acceptance of interrupt requests is disabled. When the flag is “0”, the acceptance of interrupt requests is enabled. This flag is set to “1” with the SEI instruction and is set to “0” with the CLI instruction.

When a main routine branches to an interrupt processing routine, this flag is automatically set to “1”, so that multiple interrupts are disabled. To use multiple interrupts, set this flag to “0” with the CLI instruction within the interrupt processing routine. Figure 2.2.13 shows an example of multiple interrupts.

Table 2.2.2 List of interrupt bits according to interrupt source

Interrupt source	Interrupt enable bit		Interrupt request bit	
	Address	Bit	Address	Bit
INT_0/Timer Z	$003E_{16}$	b0	$003C_{16}$	b0
INT_1	$003E_{16}$	b1	$003C_{16}$	b1
Serial I/O1 reception	$003E_{16}$	b2	$003C_{16}$	b2
Serial I/O1 transmission/SCL, SDA	$003E_{16}$	b3	$003C_{16}$	b3
Timer X	$003E_{16}$	b4	$003C_{16}$	b4
Timer Y	$003E_{16}$	b5	$003C_{16}$	b5
Timer 1	$003E_{16}$	b6	$003C_{16}$	b6
Timer 2	$003E_{16}$	b7	$003C_{16}$	b7
$CNTR_0$/SCL, SDA	$003F_{16}$	b0	$003D_{16}$	b0
$CNTR_1$/Serial I/O3 reception	$003F_{16}$	b1	$003D_{16}$	b1
Serial I/O2/Timer Z	$003F_{16}$	b2	$003D_{16}$	b2
INT_2/I^2C	$003F_{16}$	b3	$003D_{16}$	b3
INT_3	$003F_{16}$	b4	$003D_{16}$	b4
INT_4/$CNTR_2$	$003F_{16}$	b5	$003D_{16}$	b5
A/D converter/Serial I/O3 transmission	$003F_{16}$	b6	$003D_{16}$	b6

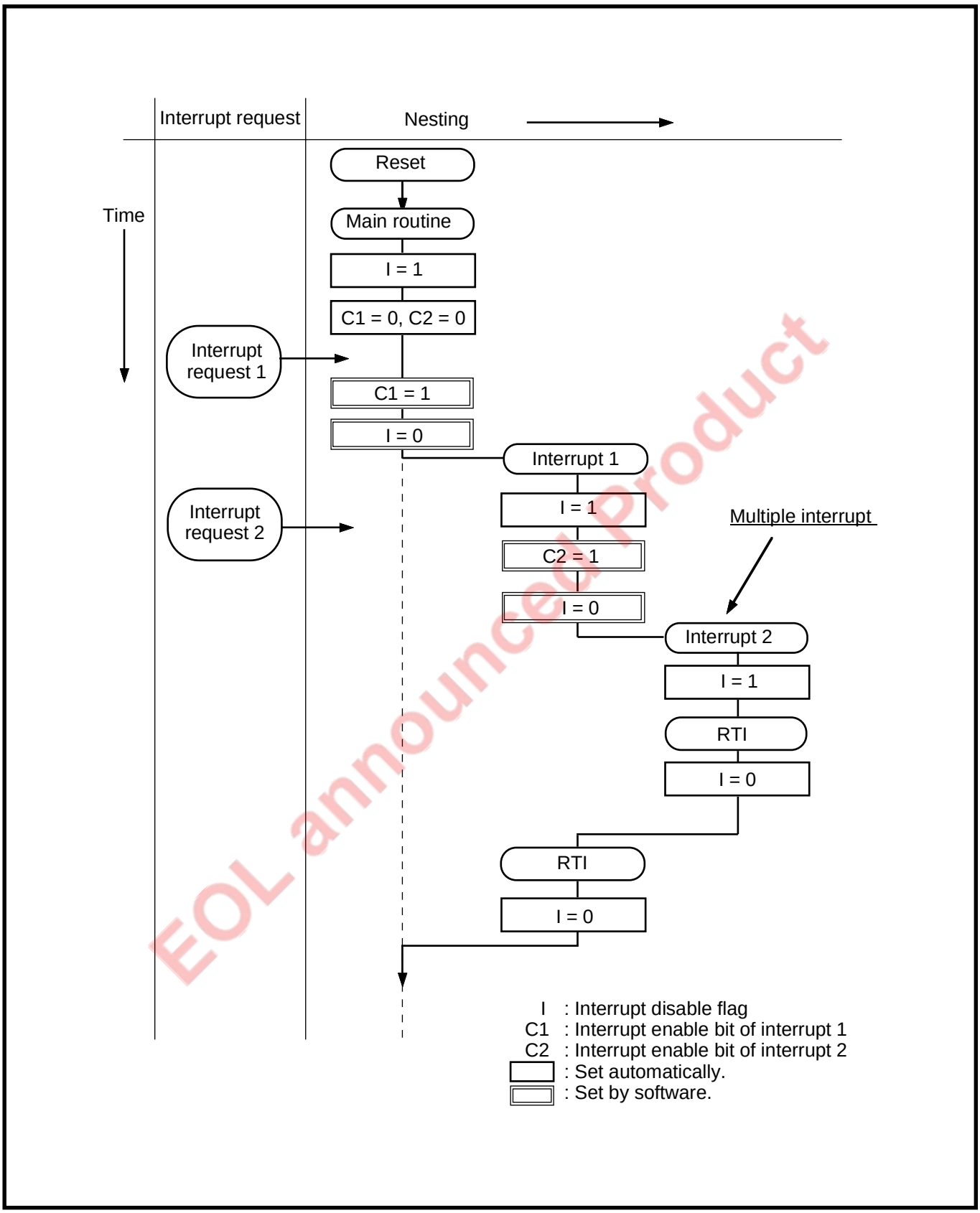


Fig. 2.2.13 Example of multiple interrupts

2.2.6 INT interrupt

The INT interrupt requests is generated when the microcomputer detects a level change of each INT pin (INT_0–INT_4).

(1) Active edge selection

INT_0–INT_4 can be selected from either a falling edge or rising edge detection as an active edge by the interrupt edge selection register. In the "0" state, the falling edge of the corresponding pin is detected. In the "1" state, the rising edge of the corresponding pin is detected.

(2) INT_0, INT_2, INT_4 interrupt source selection

When using the following interrupt source, select which of the interrupt source by the interrupt source selection register (address 0039_{16}). (Set these bits to "0" when using INT.)

- INT_0 or timer Z (bit 0)
- INT_4 or $CNTR_2$ (bit 4)
- INT_2 or I^2C (bit 5)

2.2.7 Notes on interrupts

(1) Change of relevant register settings

When the setting of the following registers or bits is changed, the interrupt request bit may be set to “1”. When not requiring the interrupt occurrence synchronized with these setting, take the following sequence.

- Interrupt edge selection register (address $003A_{16}$)
- Timer XY mode register (address 0023_{16})
- Timer Z mode register (address $002A_{16}$)
- I^2C START/STOP condition control register (address 0016_{16})

Set the above listed registers or bits as the following sequence.

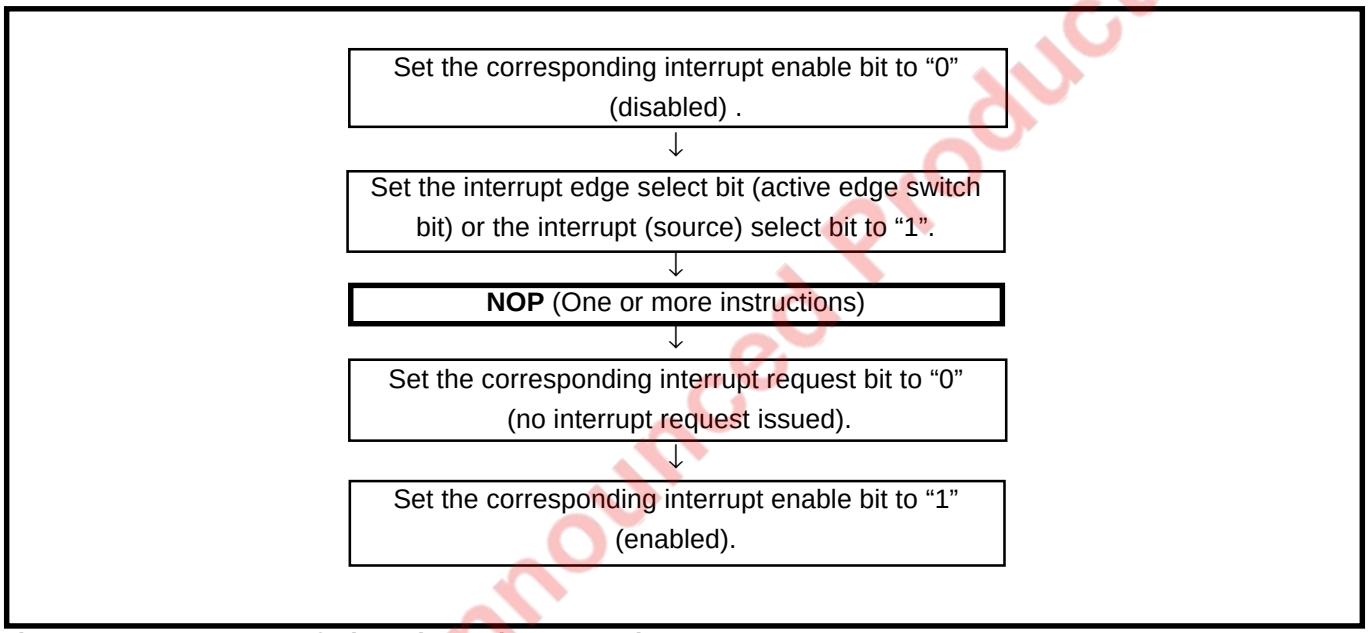


Fig. 2.2.14 Sequence of changing relevant register

■ **Reason**

When setting the followings, the interrupt request bit may be set to “1”.

- When setting external interrupt active edge

  Concerned register: Interrupt edge selection register (address $003A_{16}$)
  Timer XY mode register (address 0023_{16})
  Timer Z mode register (address $002A_{16}$)
  I^2C START/STOP condition control register (address 0016_{16})

- When switching interrupt sources of an interrupt vector address where two or more interrupt sources are allocated.

  Concerned register: Interrupt source selection register (address 0039_{16})

(2) Check of interrupt request bit

- When executing the **BBC** or **BBS** instruction to an interrupt request bit of an interrupt request register immediately after this bit is set to “0”, execute one or more instructions before executing the **BBC** or **BBS** instruction.

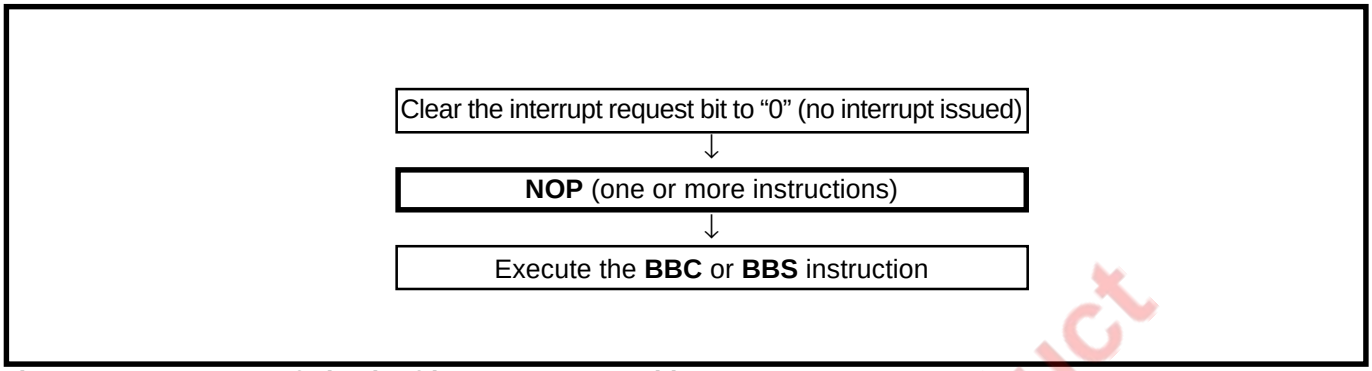


Fig. 2.2.15 Sequence of check of interrupt request bit

■ **Reason**

If the BBC or BBS instruction is executed immediately after an interrupt request bit of an interrupt request register is cleared to “0”, the value of the interrupt request bit before being cleared to “0” is read.

2.3 Timer

This paragraph explains the registers setting method and the notes relevant to the timers.

2.3.1 Memory map

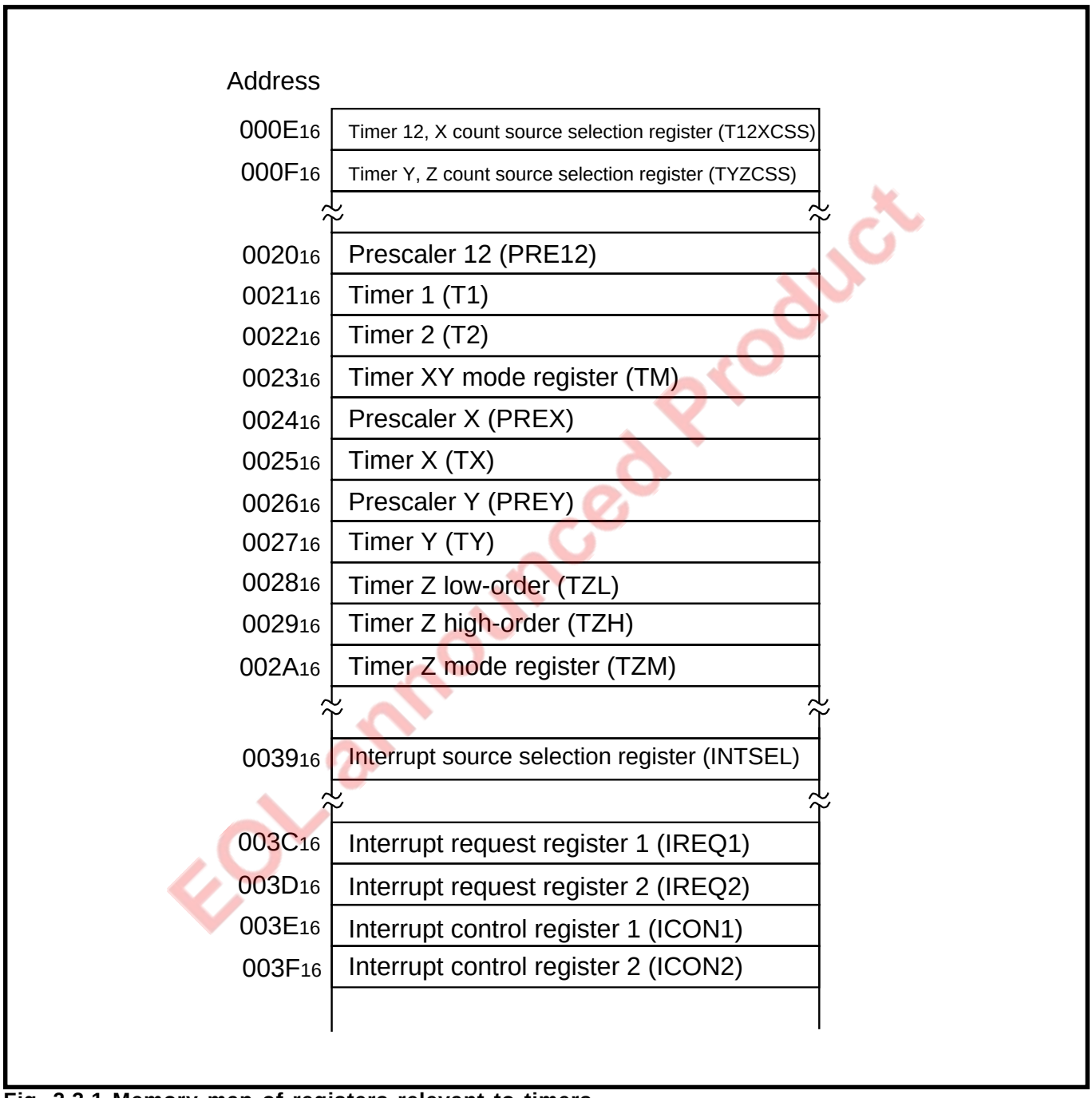


Fig. 2.3.1 Memory map of registers relevant to timers

2.3.2 Relevant registers

Prescaler 12, Prescaler X, Prescaler Y

b7 b6 b5 b4 b3 b2 b1 b0

Prescaler 12 (PRE12), Prescaler X (PREX), Prescaler Y (PREY)
(addresses 0020_{16}, 0024_{16}, 0026_{16})

b	Functions	At reset	R	W
0	• Set a count value of each prescaler. • The value set in this register is written to both each prescaler and the corresponding prescaler latch at the same time. • When this register is read out, the count value of the corresponding prescaler is read out.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Fig. 2.3.2 Structure of Prescaler 12, Prescaler X, Prescaler Y

Timer 1

b7 b6 b5 b4 b3 b2 b1 b0

Timer 1
(T1: address 0021_{16})

b	Functions	At reset	R	W
0	• Set timer 1 count value. • The value set in this register is written to both the timer 1 and the timer 1 latch at the same time. • When the timer 1 is read out, the count value of the timer 1 is read out.	1	○	○
1		0	○	○
2		0	○	○
3		0	○	○
4		0	○	○
5		0	○	○
6		0	○	○
7		0	○	○

Fig. 2.3.3 Structure of Timer 1

Timer 2, Timer X, Timer Y

b7 b6 b5 b4 b3 b2 b1 b0

Timer 2 (T2), Timer X (TX), Timer Y (TY)
(addresses 0022_{16}, 0025_{16}, 0027_{16})

b	Functions	At reset	R	W
0	• Set each timer count value. • The value set in this register is written to both each timer and the corresponding timer latch at the same time. • When each timer is read out, the count value of the corresponding timer is read out.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Fig. 2.3.4 Structure of Timer 2, Timer X, Timer Y

Timer Z low-order, Timer Z high-order

b7 b6 b5 b4 b3 b2 b1 b0

Timer Z low-order (TZL), Timer Z high-order (TZH)
(addresses 0028_{16}, 0029_{16})

b	Functions	At reset	R	W
0	• Set each timer count value. [At write] • Depending on the write control bit (bit 3 of TZM), the value set to this register is written to each timer and the corresponding timer latch at the same time, or is written only to the latch. [At read] • The corresponding timer count value is read out by reading this register. • Read both registers in order of TZH and TZL following. • Write both registers in order of TZL and TZH following.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Fig. 2.3.5 Structure of Timer Z (low-order, high-order)

Timer XY mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer XY mode register
(TM: address 0023₁₆)

b	Name	Functions	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0: Timer mode 0 1: Pulse output mode 1 0: Event counter mode 1 1: Pulse width measurement mode	0	○	○
1			0	○	○
2	CNTR₀ active edge switch bit	Refer to Table 2.3.1	0	○	○
3	Timer X count stop bit	0: Count start 1: Count stop	0	○	○
4	Timer Y operating mode bits	b5 b4 0 0: Timer mode 0 1: Pulse output mode 1 0: Event counter mode 1 1: Pulse width measurement mode	0	○	○
5			0	○	○
6	CNTR₁ active edge switch bit	Refer to Table 2.3.1	0	○	○
7	Timer Y count stop bit	0: Count start 1: Count stop	0	○	○

Fig. 2.3.6 Structure of Timer XY mode register

Table 2.3.1 CNTR₀ /CNTR₁ active edge switch bit function

Timer X /Timer Y operation modes		CNTR₀ / CNTR₁ active edge switch bit (bits 2, 6 of address 0023₁₆) contents
Timer mode	"0"	CNTR₀ / CNTR₁ interrupt request occurrence: Falling edge ; No influence to timer count
	"1"	CNTR₀ / CNTR₁ interrupt request occurrence: Rising edge ; No influence to timer count
Pulse output mode	"0"	Pulse output start: Beginning at "H" level CNTR₀ / CNTR₁ interrupt request occurrence: Falling edge
	"1"	Pulse output start: Beginning at "L" level CNTR₀ / CNTR₁ interrupt request occurrence: Rising edge
Event counter mode	"0"	Timer X / Timer Y: Rising edge count CNTR₀ / CNTR₁ interrupt request occurrence: Falling edge
	"1"	Timer X / Timer Y: Falling edge count CNTR₀ / CNTR₁ interrupt request occurrence: Rising edge
Pulse width measurement mode	"0"	Timer X / Timer Y: "H" level width measurement CNTR₀ / CNTR₁ interrupt request occurrence: Falling edge
	"1"	Timer X / Timer Y: "L" level width measurement CNTR₀ / CNTR₁ interrupt request occurrence: Rising edge

Timer Z mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer Z mode register
(TZM: address 002A16)

b	Name	Functions	At reset	R	W
0	Timer Z operating mode bits	b2b1b0 0 0 0: Timer/Event counter mode 0 0 1: Pulse output mode 0 1 0: Pulse period measurement mode 0 1 1: Pulse width measurement mode 1 0 0: Programmable waveform generating mode 1 0 1: Programmable one-shot generating mode 1 1 0: Not available 1 1 1: Not available	0	○	○
1			0	○	○
2			0	○	○
3	Timer Z write control bit	0: Writing data to both latch and timer simultaneously 1: Writing data only to latch	0	○	○
4	Output level latch	0: "L" output 1: "H" output	0	○	○
5	$CNTR_2$ active edge switch bit	Refer to Table 2.3.2.	0	○	○
6	Timer Z count stop bit	0: Count start 1: Count stop	0	○	○
7	Timer/Event counter mode switch bit **(Note)**	0: Timer mode 1: Event counter mode	0	○	○

Note: When selecting the modes except the timer/event counter mode, set "0" to this bit.

Fig. 2.3.7 Structure of Timer Z mode register

Table 2.3.2 $CNTR_2$ active edge switch bit function

Timer Z operation modes	$CNTR_2$ active edge switch bit (bit 5 of address $002A_{16}$) contents	
Timer mode	"0"	$CNTR_2$ interrupt request occurrence: Falling edge ; No influence to timer count
	"1"	$CNTR_2$ interrupt request occurrence: Rising edge ; No influence to timer count
Event counter mode	"0"	Timer Z: Rising edge count $CNTR_2$ interrupt request occurrence: Falling edge
	"1"	Timer Z: Falling edge count $CNTR_2$ interrupt request occurrence: Rising edge
Pulse output mode	"0"	Pulse output start: Beginning at "H" level $CNTR_2$ interrupt request occurrence: Falling edge
	"1"	Pulse output start: Beginning at "L" level $CNTR_2$ interrupt request occurrence: Rising edge
Pulse period measurement mode	"0"	Timer Z : Term from one falling edge to next falling edge measurement $CNTR_2$ interrupt request occurrence: Falling edge
	"1"	Timer Z : Term from one rising edge to next rising edge measurement $CNTR_2$ interrupt request occurrence: Rising edge
Pulse width measurement mode	"0"	Timer Z: "H" level width measurement $CNTR_2$ interrupt request occurrence: Falling edge
	"1"	Timer Z: "L" level width measurement $CNTR_2$ interrupt request occurrence: Rising edge
Programmable one-shot generating mode	"0"	Timer Z : Pulse output start from "L" level, and "H" level one-shot pulse is output. $CNTR_2$ interrupt request occurrence: Falling edge
	"1"	Timer Z : Pulse output start from "H" level, and "L" level one-shot pulse is output. $CNTR_2$ interrupt request occurrence: Rising edge

Timer 12, X count source selection register

b7 b6 b5 b4 b3 b2 b1 b0

Timer 12, X count source selection register
(T12XCSS: address 000E16)

b	Name	Functions	At reset	R	W
0	Timer 12 count source selection bits	b3b2b1b0 0 0 0 0: f(XIN)/2 or f(XCIN)/2 0 0 0 1: f(XIN)/4 or f(XCIN)/4 0 0 1 0: f(XIN)/8 or f(XCIN)/8 0 0 1 1: f(XIN)/16 or f(XCIN)/16 0 1 0 0: f(XIN)/32 or f(XCIN)/32 0 1 0 1: f(XIN)/64 or f(XCIN)/64 0 1 1 0: f(XIN)/128 or f(XCIN)/128 0 1 1 1: f(XIN)/256 or f(XCIN)/256 1 0 0 0: f(XIN)/512 or f(XCIN)/512 1 0 0 1: f(XIN)/1024 or f(XCIN)/1024 1010 to 1111: Not available	1	O	O
1			1	O	O
2			0	O	O
3			0	O	O
4	Timer X count source selection bits	b7b6b5b4 0 0 0 0: f(XIN)/2 or f(XCIN)/2 0 0 0 1: f(XIN)/4 or f(XCIN)/4 0 0 1 0: f(XIN)/8 or f(XCIN)/8 0 0 1 1: f(XIN)/16 or f(XCIN)/16 0 1 0 0: f(XIN)/32 or f(XCIN)/32 0 1 0 1: f(XIN)/64 or f(XCIN)/64 0 1 1 0: f(XIN)/128 or f(XCIN)/128 0 1 1 1: f(XIN)/256 or f(XCIN)/256 1 0 0 0: f(XIN)/512 or f(XCIN)/512 1 0 0 1: f(XIN)/1024 or f(XCIN)/1024 1 0 1 0: f(XCIN) 1011 to 1111: Not available	1	O	O
5			1	O	O
6			0	O	O
7			0	O	O

Fig. 2.3.8 Structure of Timer 12, X count source selection register

Timer Y, Z count source selection register

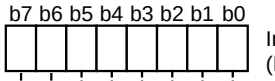
b7 b6 b5 b4 b3 b2 b1 b0

Timer Y, Z count source selection register
(TYZCSS: address 000F16)

b	Name	Functions	At reset	R	W
0	Timer Y count source selection bits	b3b2b1b0 0 0 0 0: f(XIN)/2 or f(XCIN)/2 0 0 0 1: f(XIN)/4 or f(XCIN)/4 0 0 1 0: f(XIN)/8 or f(XCIN)/8 0 0 1 1: f(XIN)/16 or f(XCIN)/16 0 1 0 0: f(XIN)/32 or f(XCIN)/32 0 1 0 1: f(XIN)/64 or f(XCIN)/64 0 1 1 0: f(XIN)/128 or f(XCIN)/128 0 1 1 1: f(XIN)/256 or f(XCIN)/256 1 0 0 0: f(XIN)/512 or f(XCIN)/512 1 0 0 1: f(XIN)/1024 or f(XCIN)/1024 1 0 1 0: f(XCIN) 1011 to 1111: Not available	1	O	O
1			1	O	O
2			0	O	O
3			0	O	O
4	Timer Z count source selection bits	b7b6b5b4 0 0 0 0: f(XIN)/2 or f(XCIN)/2 0 0 0 1: f(XIN)/4 or f(XCIN)/4 0 0 1 0: f(XIN)/8 or f(XCIN)/8 0 0 1 1: f(XIN)/16 or f(XCIN)/16 0 1 0 0: f(XIN)/32 or f(XCIN)/32 0 1 0 1: f(XIN)/64 or f(XCIN)/64 0 1 1 0: f(XIN)/128 or f(XCIN)/128 0 1 1 1: f(XIN)/256 or f(XCIN)/256 1 0 0 0: f(XIN)/512 or f(XCIN)/512 1 0 0 1: f(XIN)/1024 or f(XCIN)/1024 1 0 1 0: f(XCIN) 1011 to 1111: Not available	1	O	O
5			1	O	O
6			0	O	O
7			0	O	O

Fig. 2.3.9 Structure of Timer Y, Z count source selection register

Interrupt source selection register



Interrupt source selection register
(INTSEL: address 0039$_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0/Timer Z interrupt source selection bit (*1)	0: INT_0 interrupt 1: Timer Z interrupt	0	○	○
1	Serial I/O2/Timer Z interrupt source selection bit (*1)	0: Serial I/O2 interrupt 1: Timer Z interrupt	0	○	○
2	Serial I/O1 transmit/ SCL, SDA interrupt source selection bit (*2)	0: Serial I/O1 transmit interrupt 1: SCL, SDA interrupt	0	○	○
3	$CNTR_0$/SCL, SDA interrupt source selection bit (*2)	0: $CNTR_0$ interrupt 1: SCL, SDA interrupt	0	○	○
4	INT_4/$CNTR_2$ interrupt source selection bit	0: INT_4 interrupt 1: $CNTR_2$ interrupt	0	○	○
5	INT_2/I^2C interrupt source selection bit	0: INT_2 interrupt 1: I^2C interrupt	0	○	○
6	$CNTR_1$/Serial I/O3 receive interrupt source selection bit	0: $CNTR_1$ interrupt 1: Serial I/O3 receive interrupt	0	○	○
7	AD converter/Serial I/O3 transmit interrupt source selection bit	0: A/D converter interrupt 1: Serial I/O3 transmit interrupt	0	○	○

*1: Do not write 1 to these bits simultaneously.
*2: Do not write 1 to these bits simultaneously.

Fig. 2.3.10 Structure of Interrupt source selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1
(IREQ1 : address 003C16)

b	Name	Functions	At reset	R	W
0	INT0/Timer Z interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O1 transmit/SCL, SDA interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻: 0 can be set by software, but 1 cannot be set.

Fig. 2.3.11 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2
(IREQ2 : address 003D16)

b	Name	Functions	At reset	R	W
0	CNTR0/SCL, SDA interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	CNTR1/Serial I/O3 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O2/Timer Z interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT2/I2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	INT4/CNTR2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	AD converter/Serial I/O3 transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are 0 .		0	○	×

✻: 0 can be set by software, but 1 cannot be set.

Fig. 2.3.12 Structure of Interrupt request register 2

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1
(ICON1 : address 003E16)

b	Name	Functions	At reset	R	W
0	INT0/Timer Z interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit/SCL, SDA interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.3.13 Structure of Interrupt control register 1

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0
0

Interrupt control register 2
(ICON2 : address 003F16)

b	Name	Functions	At reset	R	W
0	CNTR0/SCL, SDA interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	CNTR1/ Serial I/O3 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/ Timer Z interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT2/I2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT3 interrupt enable bit	0 : Interrupt disabled	0	○	○
5	INT4/CNTR2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/Serial I/O3 transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.3.14 Structure of Interrupt control register 2

2.3.3 Timer application examples

(1) Basic functions and uses

[Function 1] Control of Event interval (Timer X, Timer Y, Timer Z, Timer 1, Timer 2)

When a certain time, by setting a count value to each timer, has passed, the timer interrupt request occurs.

<Use>

- •Generation of an output signal timing
- •Generation of a wait time

[Function 2] Control of Cyclic operation (Timer X, Timer Y, Timer Z, Timer 1, Timer 2)

The value of the timer latch is automatically written to the corresponding timer each time the timer underflows, and each timer interrupt request occurs in cycles.

<Use>

- •Generation of cyclic interrupts
- •Clock function (measurement of 250 ms); see Application example 1
- •Control of a main routine cycle

[Function 3] Output of Rectangular waveform (Timer X, Timer Y, Timer Z)

The output level of the CNTR pin is inverted each time the timer underflows (in the pulse output mode).

<Use>

- •Piezoelectric buzzer output; see Application example 2
- •Generation of the remote control carrier waveforms

[Function 4] Count of External pulses (Timer X, Timer Y, Timer Z)

External pulses input to the CNTR pin are counted as the timer count source (in the event counter mode).

<Use>

- •Frequency measurement; see Application example 3
- •Division of external pulses
- •Generation of interrupts due to a cycle using external pulses as the count source; count of a reel pulse

[Function 5] Measurement of External pulse width (Timer X, Timer Y, Timer Z)

The “H” or “L” level width of external pulses input to CNTR pin is measured (in the pulse width measurement mode).

<Use>

- •Measurement of external pulse frequency (measurement of pulse width of FG pulse* for a motor); see Application example 4
- •Measurement of external pulse duty (when the frequency is fixed)

FG pulse*: Pulse used for detecting the motor speed to control the motor speed.

[Function 6] Output of Arbitrary waveform (Timer Z)

The value which is set to the output level latch is output from the CNTR pin each time the timer underflows. (programmable waveform generating mode)

[Function 7] One-shot pulse output by external trigger (Timer Z)

The value of timer latch is set to timer by trigger signal which is input from the INT pin, and timer is counted down. When trigger signal is input, “H” or “L” is output from the CNTR pin at the same time, and “L” or “H” is output by underflow of timer. (programmable one-shot generating mode)

(2) Timer application example 1: Clock function (measurement of 250 ms)

Outline: The input clock is divided by the timer so that the clock can count up at 250 ms intervals.

Specifications: •The clock f(X_{IN}) = 4.19 MHz (2^{22} Hz) is divided by the timer.
•The clock is counted up in the process routine of the timer X interrupt which occurs at 250 ms intervals.

Figure 2.3.15 shows the timers connection and setting of division ratios; Figure 2.3.16 shows the relevant registers setting; Figure 2.3.17 shows the control procedure.

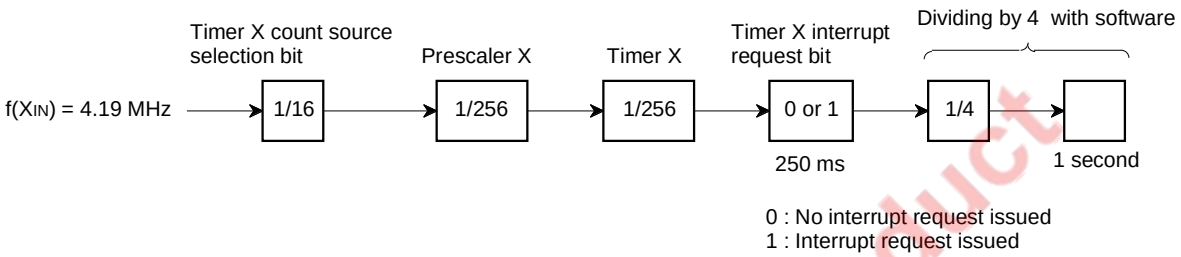


Fig. 2.3.15 Timers connection and setting of division ratios

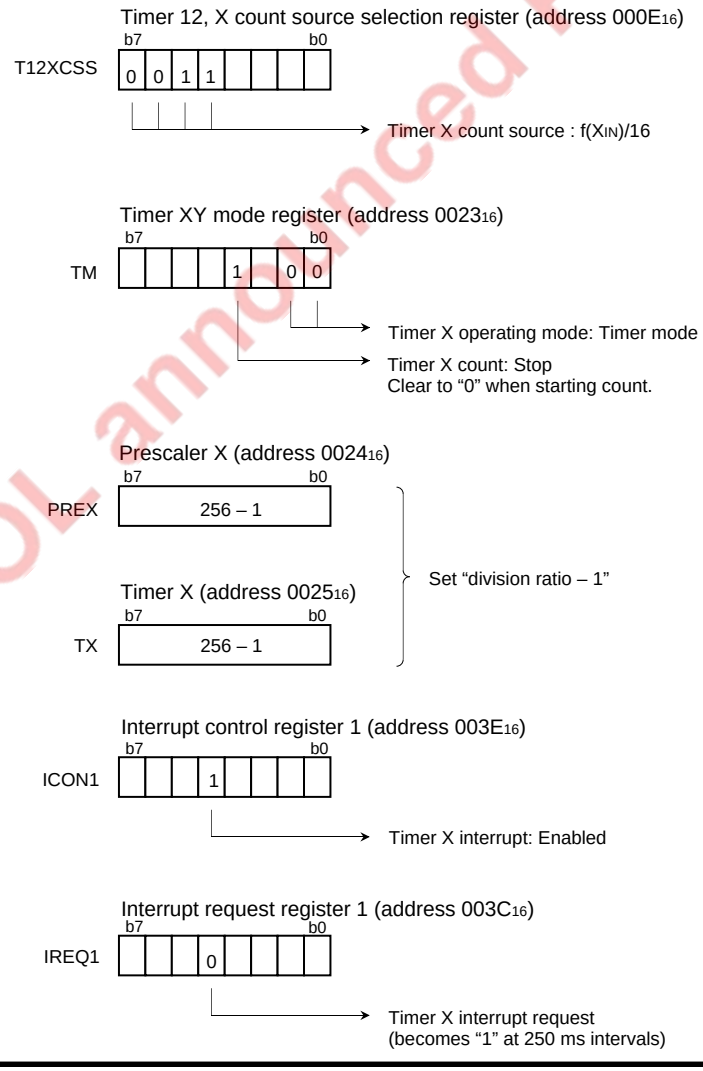


Fig. 2.3.16 Relevant registers setting

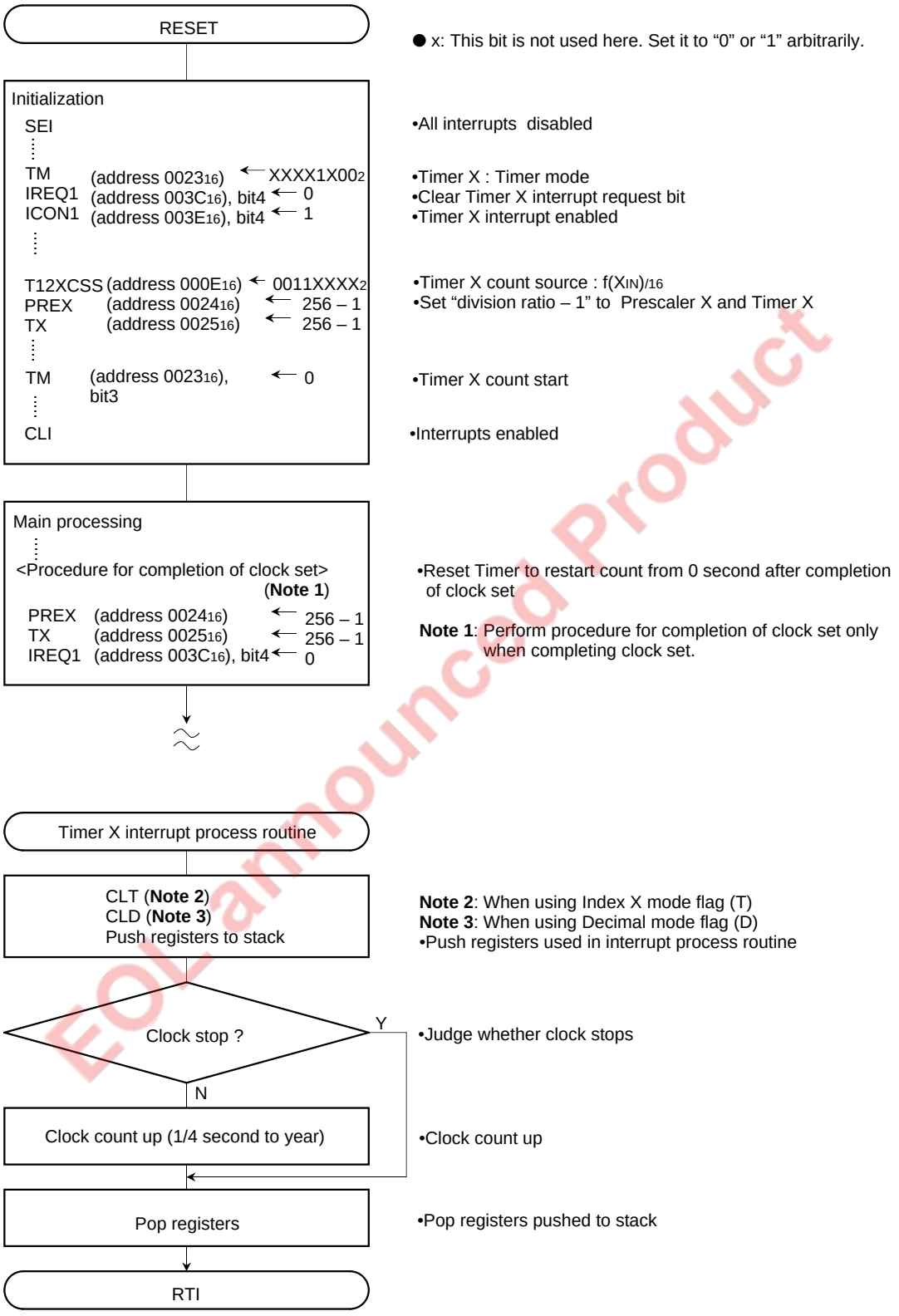


Fig. 2.3.17 Control procedure

(3) Timer application example 2: Piezoelectric buzzer output

Outline: The rectangular waveform output function of the timer is applied for a piezoelectric buzzer output.

Specifications: •The rectangular waveform, dividing the clock f(X_{IN}) = 8 MHz into about 2 kHz (2049 Hz), is output from the $P4_7$/$CNTR_2$ pin.
•The level of the $P4_7$/$CNTR_2$ pin is fixed to "H" while a piezoelectric buzzer output stops.

Figure 2.3.18 shows a peripheral circuit example, and Figure 2.3.19 shows the timers connection and setting of division ratios. Figure 2.3.20 shows the relevant registers setting, and Figure 2.3.21 shows the control procedure.

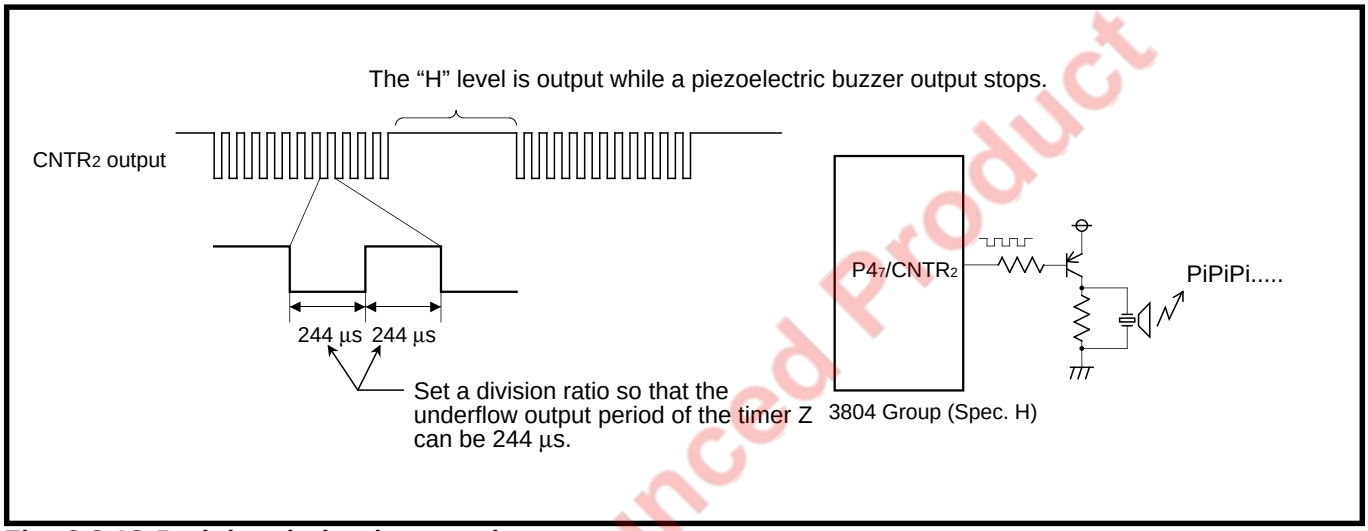


Fig. 2.3.18 Peripheral circuit example

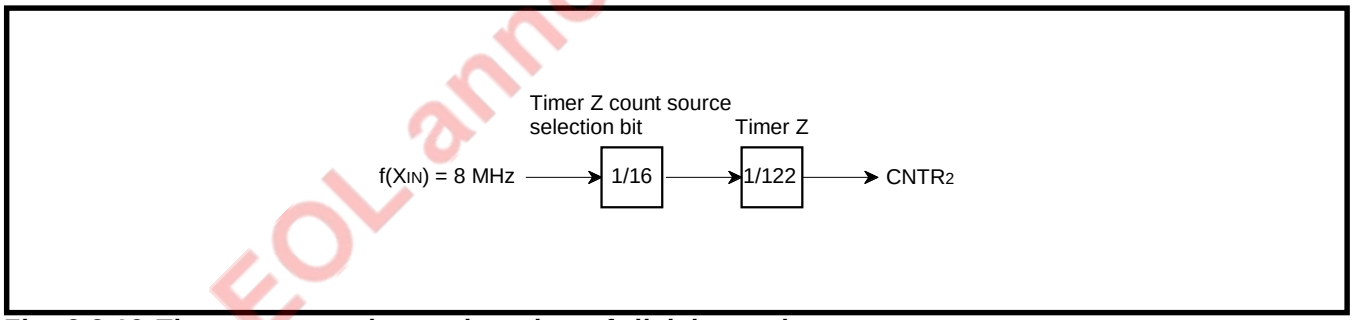


Fig. 2.3.19 Timers connection and setting of division ratios

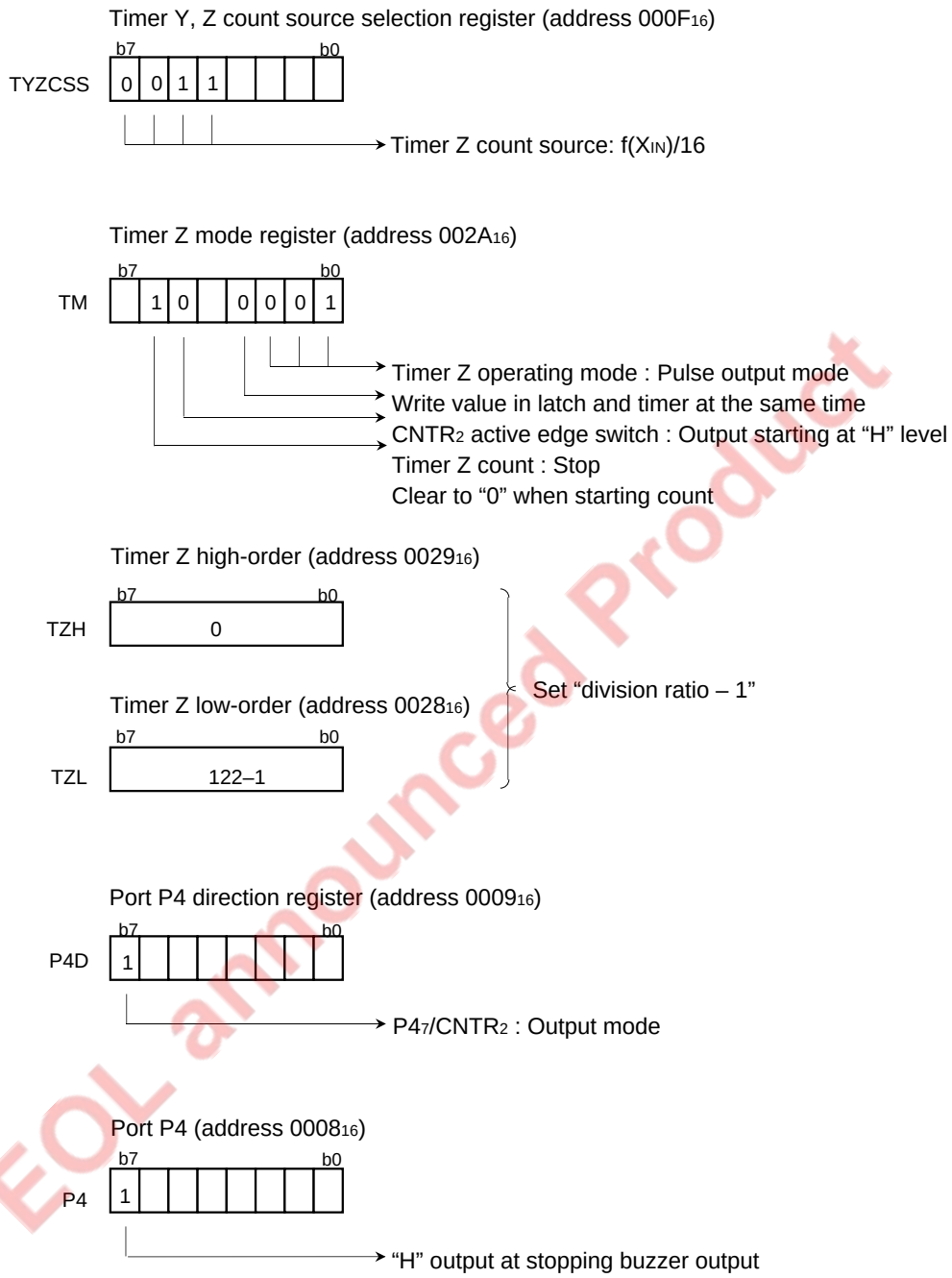


Fig. 2.3.20 Relevant registers setting

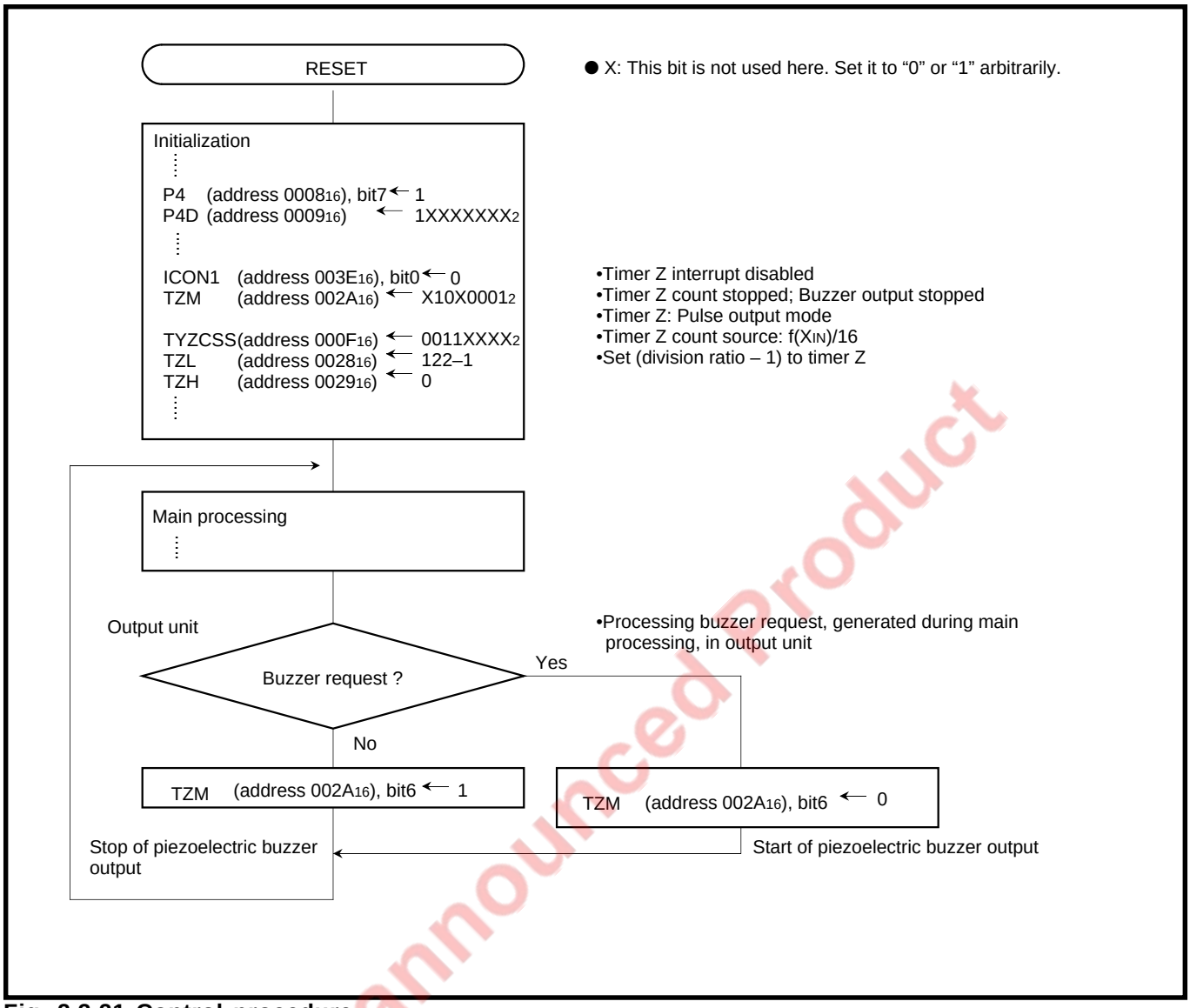


Fig. 2.3.21 Control procedure

(4) Timer application example 3: Frequency measurement

Outline: The following two values are compared to judge whether the frequency is within a valid range.
- A value by counting pulses input to $P5_5/CNTR_1$ pin with the timer.
- A reference value

Specifications:
- The pulse is input to the $P5_5/CNTR_1$ pin and counted by the timer Y.
- The clock f(X_{IN}) = 8 MHz is dividing by the timer 1, and the interrupt request occurs at about 2 ms intervals.
- A count value is read out at about 2 ms intervals, the timer 1 interrupt interval. When the count value is 28 to 40, it is judged that the input pulse is valid.
- Because the timer is a down-counter, the count value is compared with 227 to 215 **(Note)**.

Note: 227 to 215 = {255 (initial value of counter) – 28} to {255 – 40}; 28 to 40 means the number of valid count value.

Figure 2.3.22 shows the judgment method of valid/invalid of input pulses; Figure 2.3.23 shows the relevant registers setting; Figure 2.3.24 shows the control procedure.

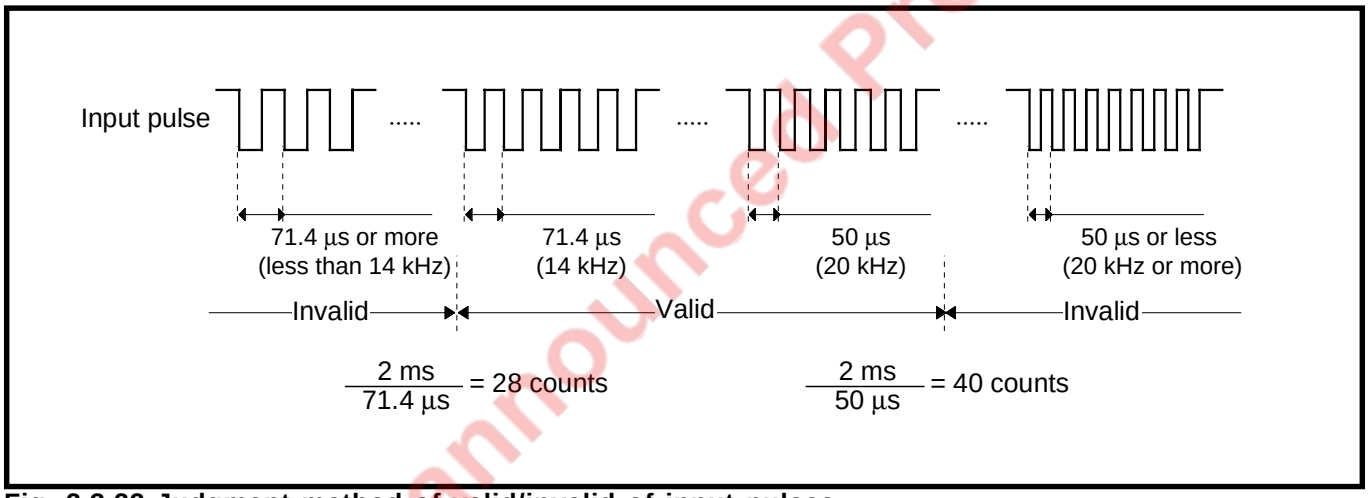


Fig. 2.3.22 Judgment method of valid/invalid of input pulses

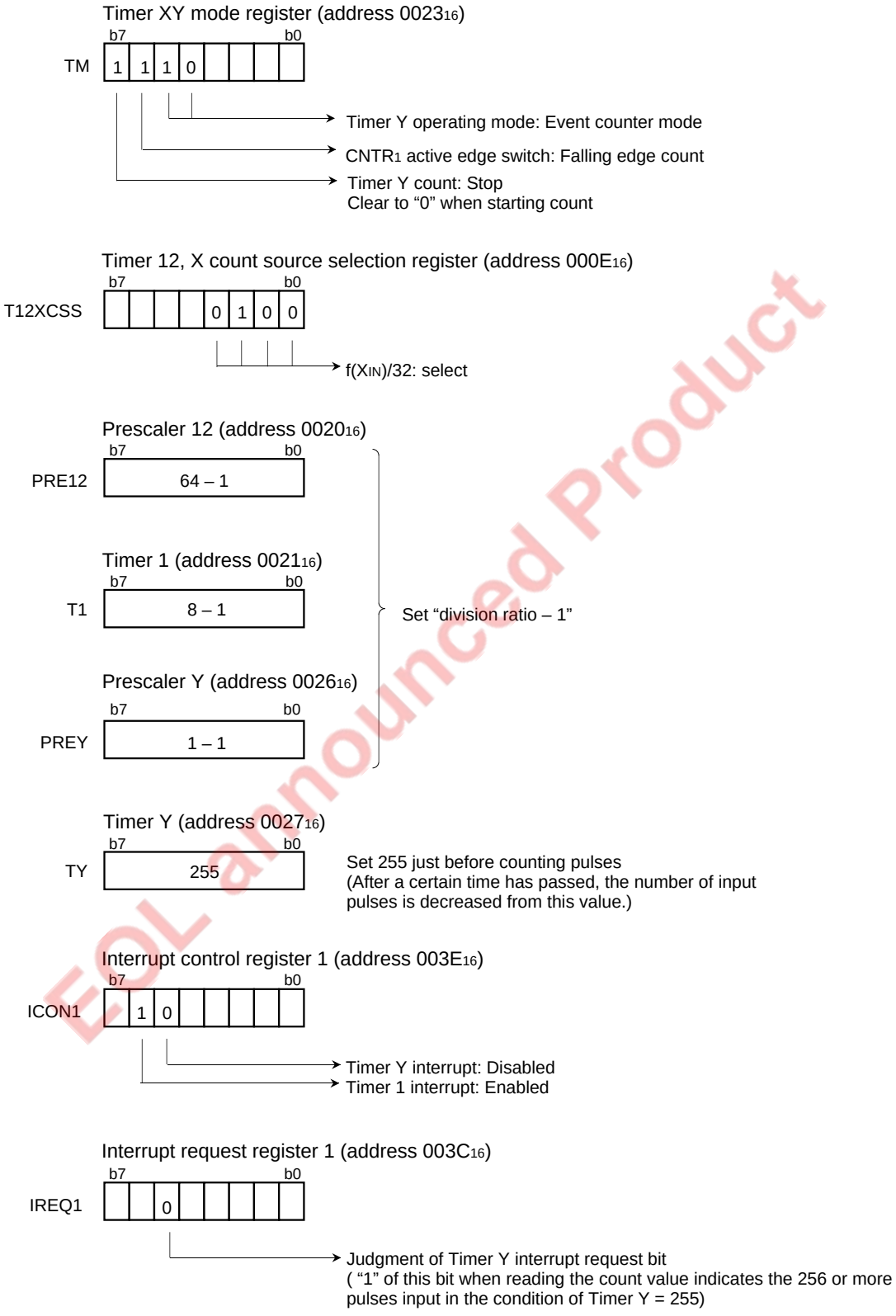


Fig. 2.3.23 Relevant registers setting

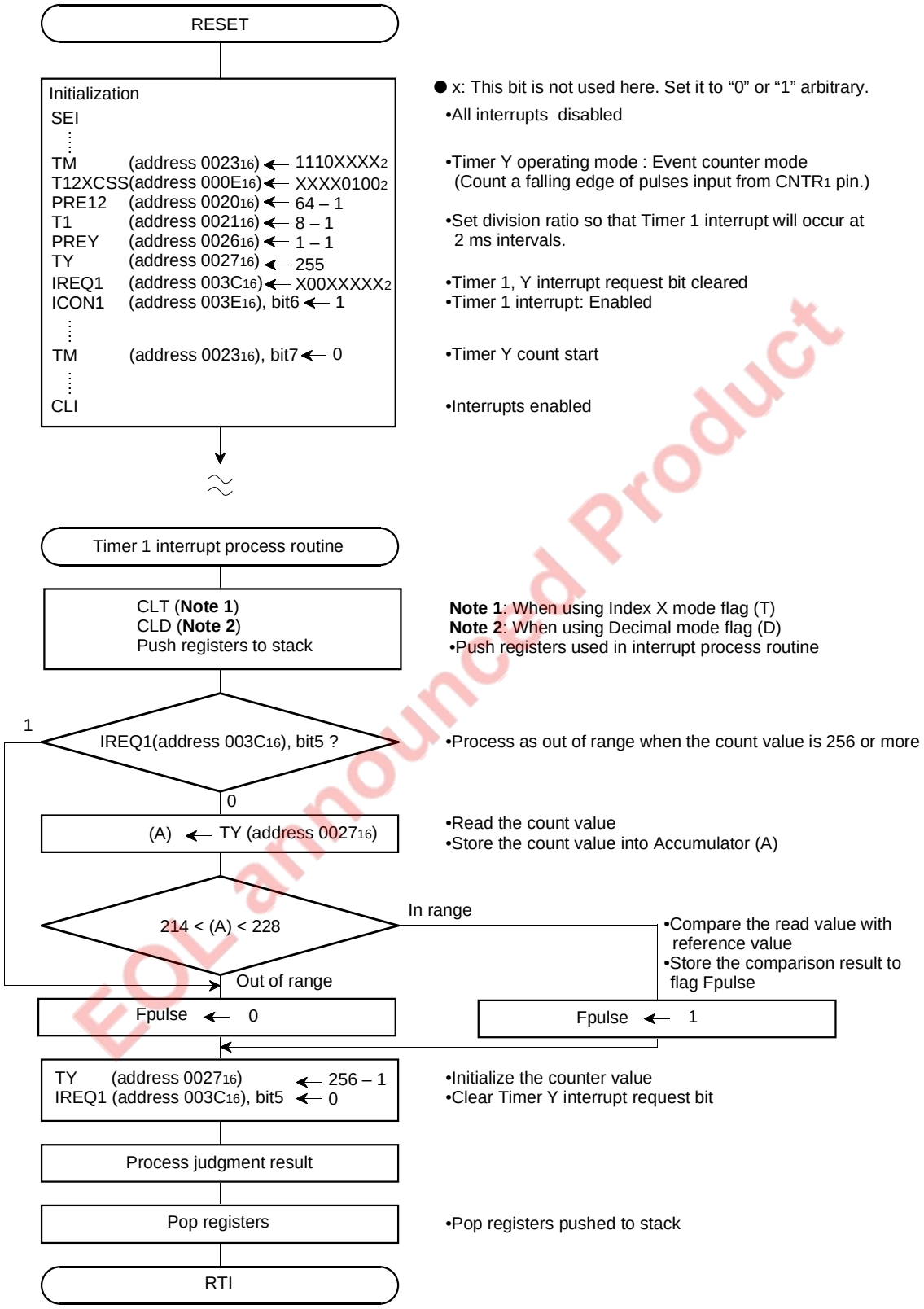


Fig. 2.3.24 Control procedure

(5) Timer application example 4: Measurement of FG pulse width for motor

Outline: The timer Z counts the "H" level width of the pulses input to the $P4_7/CNTR_2$ pin. An underflow is detected by the timer Z interrupt and an end of the input pulse "H" level is detected by the $P4_7/CNTR_2$ interrupt.

Specifications: •The timer Z counts the "H" level width of the FG pulse input to the $P4_7/CNTR_2$ pin.

<Example>

When the clock frequency is 8 MHz, the count source is 2 µs, which is obtained by dividing the clock frequency by 16. Measurement can be made up to 131.072 ms in the range of $FFFF_{16}$ to 0000_{16}.

Figure 2.3.25 shows the timers connection and setting of division ratio; Figure 2.3.26 shows the relevant registers setting; Figure 2.3.27 and Figure 2.3.28 show the control procedure.

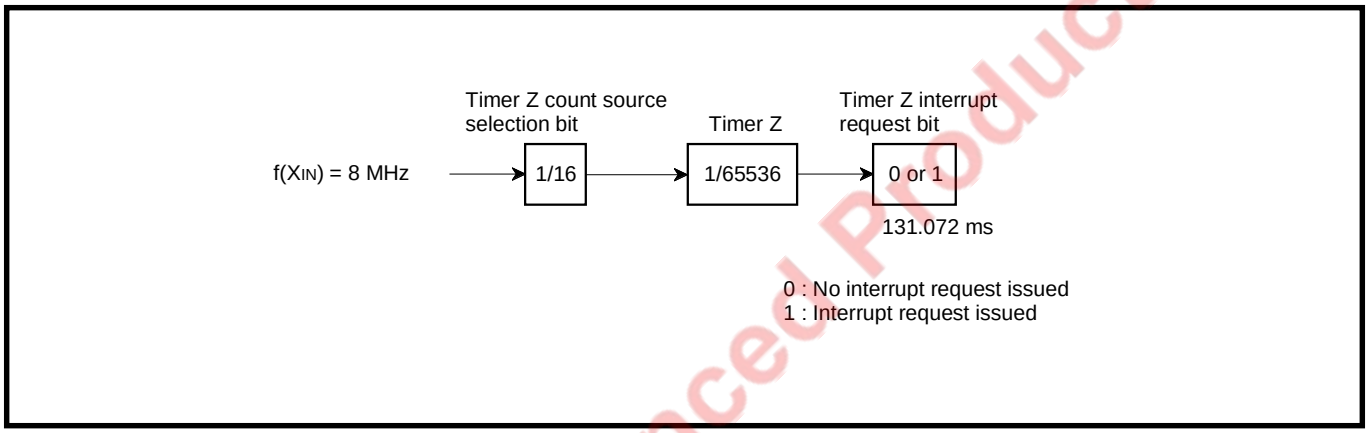


Fig. 2.3.25 Timers connection and setting of division ratios

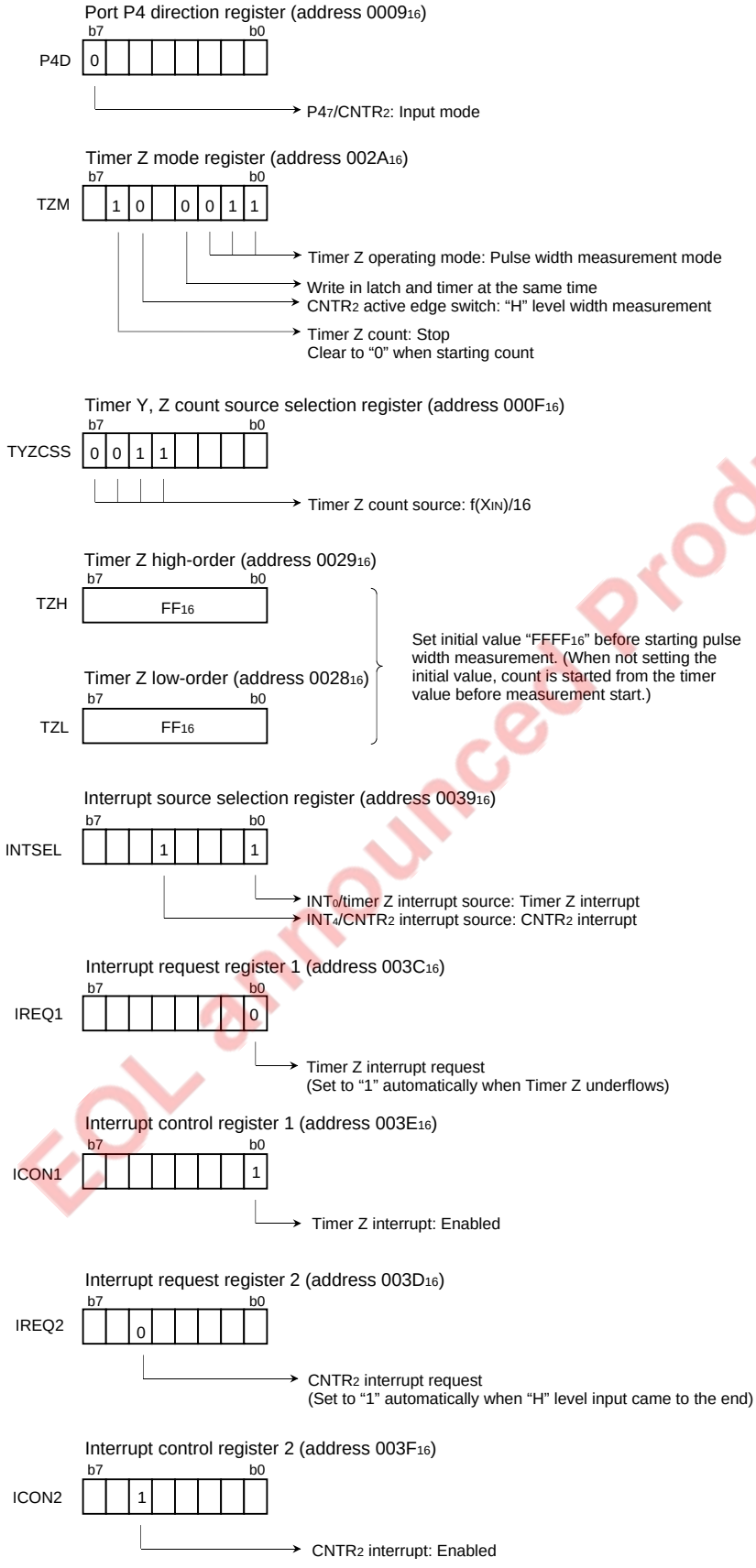


Fig. 2.3.26 Relevant registers setting

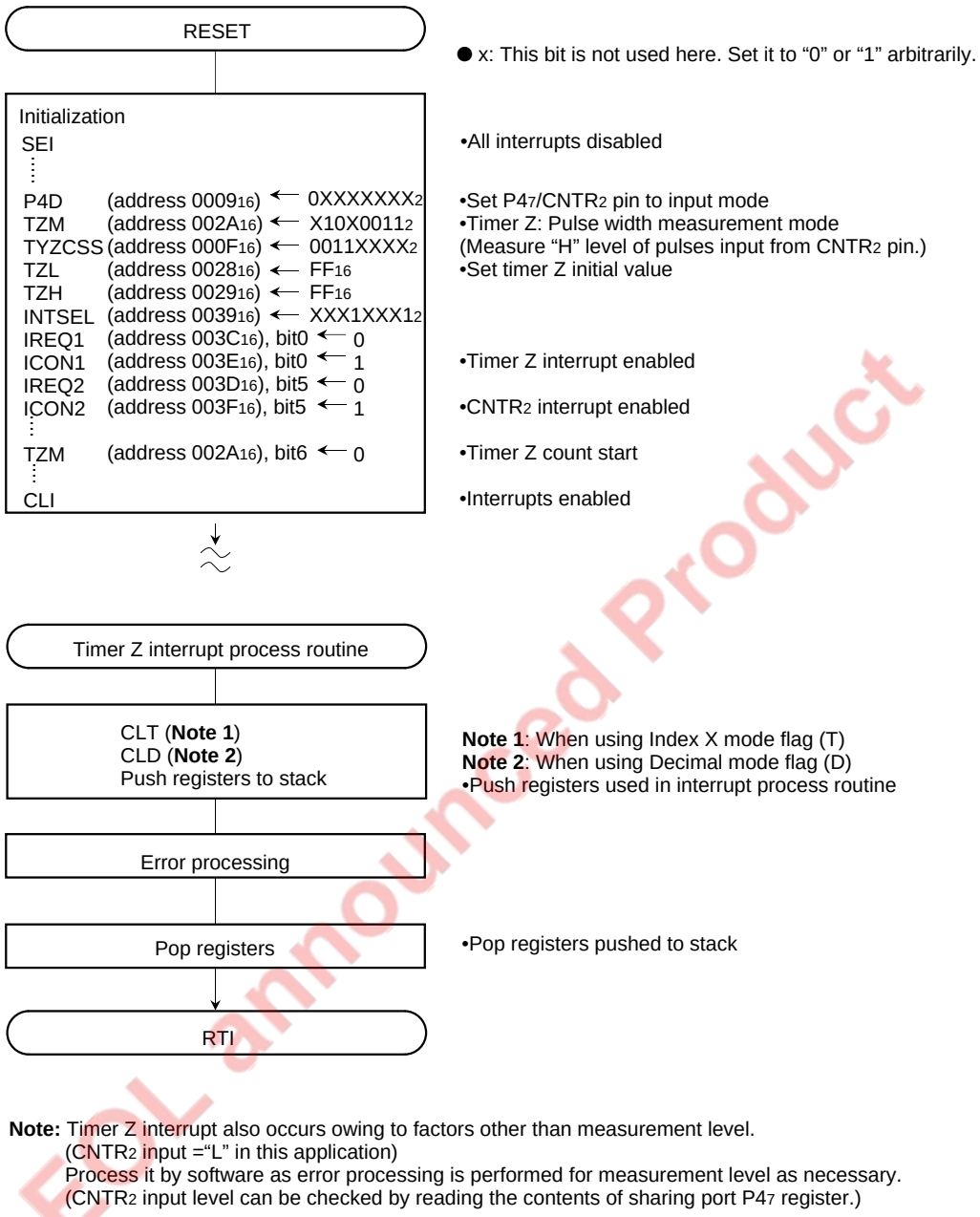
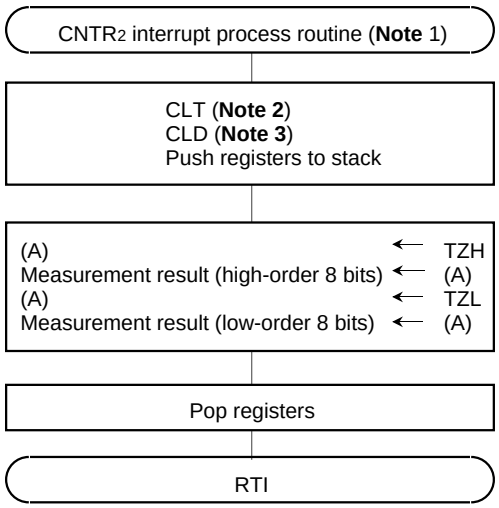


Fig. 2.3.27 Control procedure (1)



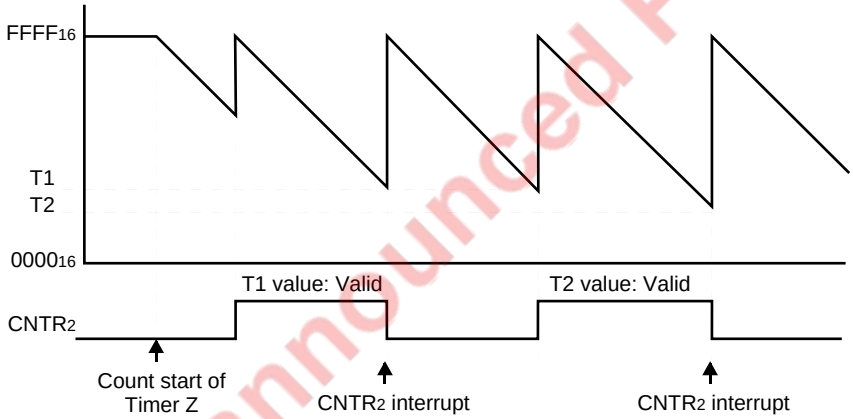
Notes **2**: When using Index X mode flag (T)
3: When using Decimal mode flag (D)
•Pushing registers used in interrupt process routine

•Count value read and storing it to RAM

•Popping registers pushed to stack

Note 1: The first value becomes invalid depending on start timing of Timer Z count shown by the following figure.
Process it by software as necessary.

[Example 1] • Start Timer Z count when $CNTR_2$ input level is "L".
($CNTR_2$ input level can be checked by reading the contents of sharing port $P4_7$ register.)



[Example 2] • Start Timer Z count when $CNTR_2$ input level is "H".
Invalidate the first $CNTR_2$ interrupt after start of Timer Z count.

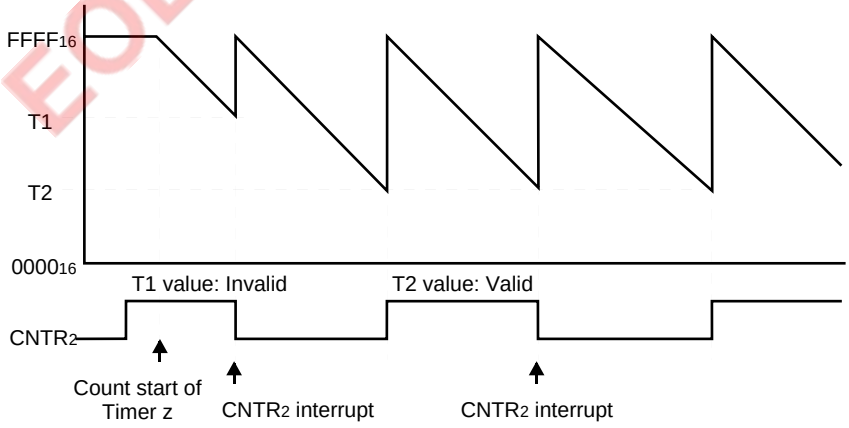


Fig. 2.3.28 Control procedure (2)

2.3.4 Notes on timer

Notes on 8-bit timer (timer 1, 2, X, Y)

- If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is 1/(n+1).
- When switching the count source by the timer 12, X and Y count source selection bits, the value of timer count is altered in unconsiderable amount owing to generating of thin pulses in the count input signals.
  Therefore, select the timer count source before set the value to the prescaler and the timer.
- Set the double-function port of the $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to output in the pulse output mode.
- Set the double-function port of $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to input in the event counter mode and the pulse width measurement mode.

Notes on 16-bit timer (timer Z)

(1) Pulse output mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to output.

(2) Pulse period measurement mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to input.
- A read-out of timer value is impossible in this mode. The timer can be written to only during timer stop (no measurement of pulse period).
- Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operation during measurement.
- "$FFFF_{16}$" is set to the timer when the timer underflows or when the valid edge of measurement start/completion is detected. Consequently, the timer value at start of pulse period measurement depends on the timer value just before measurement start.

(3) Pulse width measurement mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to input.
- A read-out of timer value is impossible in this mode. The timer can be written to only during timer stop (no measurement of pulse period).
- Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operation during measurement.
- "$FFFF_{16}$" is set to the timer when the timer underflows or when the valid edge of measurement start/completion is detected. Consequently, the timer value at start of pulse width measurement depends on the timer value just before measurement start.

(4) Programmable waveform generating mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to output.

(5) Programmable one-shot generating mode

- Set the double-function port of $CNTR_2$ pin and port $P4_7$ to output, and of INT_1 pin and port $P4_2$ to input in this mode.
- This mode cannot be used in low-speed mode.
- If the value of the $CNTR_2$ active edge switch bit is changed during one-shot generating enabled or generating one-shot pulse, then the output level from $CNTR_2$ pin changes.

(6) All modes

●Timer Z write control

Which write control can be selected by the timer Z write control bit (bit 3) of the timer Z mode register (address $002A_{16}$), writing data to both the latch and the timer at the same time or writing data only to the latch.

When the operation “writing data only to the latch” is selected, the value is set to the timer latch by writing data to the address of timer Z and the timer is updated at next underflow. After reset release, the operation “writing data to both the latch and the timer at the same time” is selected, and the value is set to both the latch and the timer at the same time by writing data to the address of timer Z.

In the case of writing data only to the latch, if writing data to the latch and an underflow are performed almost at the same time, the timer value may become undefined.

●Timer Z read control

A read-out of timer value is impossible in pulse period measurement mode and pulse width measurement mode. In the other modes, a read-out of timer value is possible regardless of count operating or stopped.

However, a read-out of timer latch value is impossible.

●Switch of interrupt active edge of $CNTR_2$ and INT_1

Each interrupt active edge depends on setting of the $CNTR_2$ active edge switch bit and the INT_1 active edge selection bit.

●Switch of count source

When switching the count source by the timer Z count source selection bits, the value of timer count is altered in inconsiderable amount owing to generating of thin pulses on the count input signals.

Therefore, select the timer count source before setting the value to the prescaler and the timer.

2.4 Serial interface

This paragraph explains the registers setting method and the notes relevant to Serial I/O.

2.4.1 Memory map

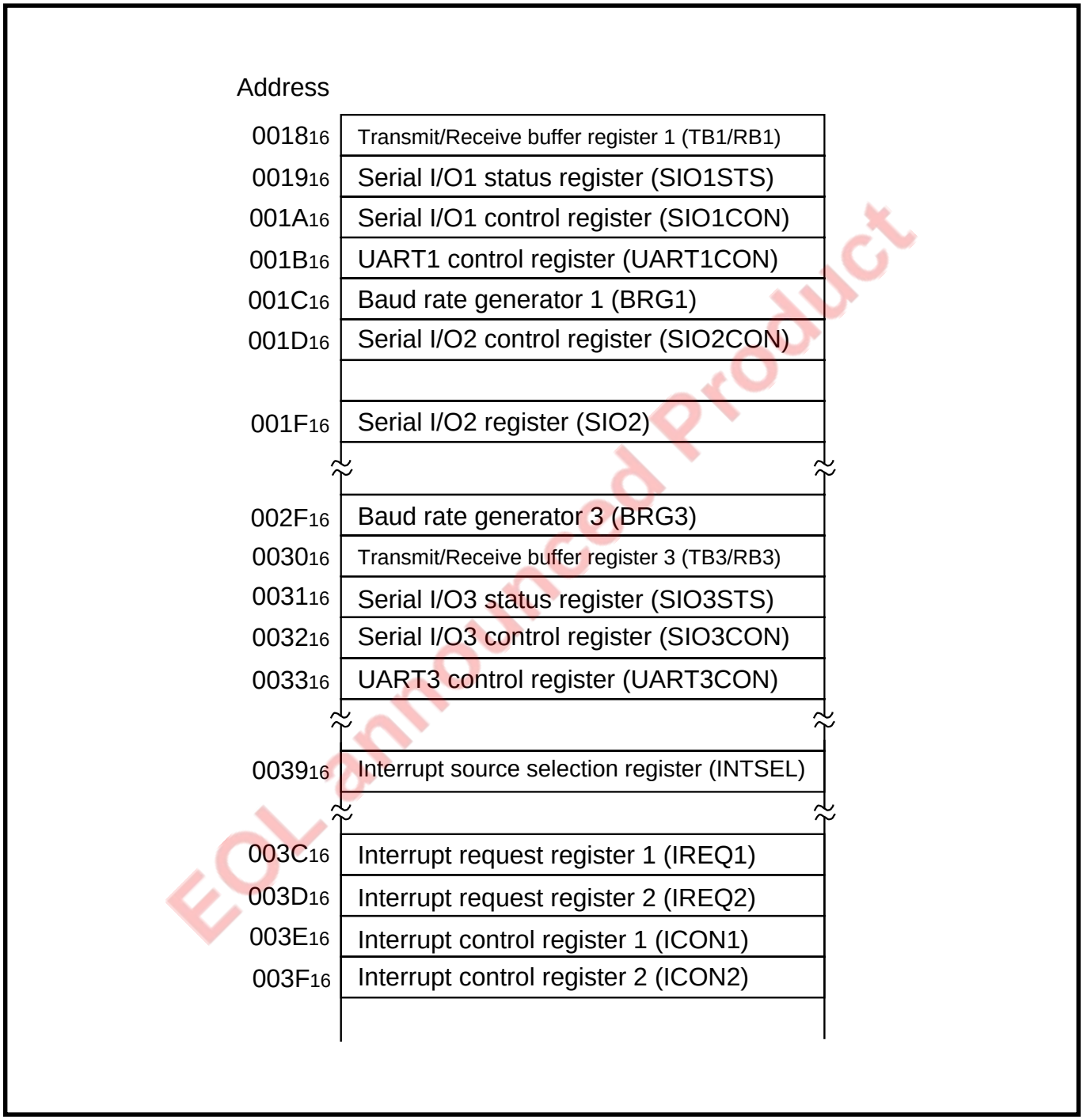
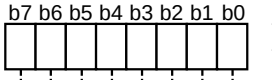


Fig. 2.4.1 Memory map of registers relevant to Serial I/O

2.4.2 Relevant registers

Transmit/Receive buffer register 1, Transmit/Receive buffer register 3



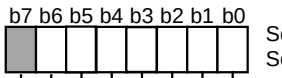
Transmit/Receive buffer register 1 (TB1/RB1: address 0018_{16})
Transmit/Receive buffer register 3 (TB3/RB3: address 0030_{16})

b	Functions	At reset	R	W
0	The transmission data is written to or the receive data is read out from this buffer register. • At write: A data is written to the transmit buffer register. • At read: The contents of the receive buffer register are read out.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Note: The contents of transmit buffer register cannot be read out.
The data cannot be written to the receive buffer register.

Fig. 2.4.2 Structure of Transmit/Receive buffer register 1 and Transmit/Receive buffer register 3

Serial I/O1 status register, Serial I/O3 status register



Serial I/O1 status register (SIO1STS: address 0019_{16})
Serial I/O3 status register (SIO3STS: address 0031_{16})

b	Name	Functions	At reset	R	W
0	Transmit buffer empty flag (TBE)	0: Buffer full 1: Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0: Buffer empty 1: Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0: Transmit shift in progress 1: Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0: No error 1: Overrun error	0	○	×
4	Parity error flag (PE)	0: No error 1: Parity error	0	○	×
5	Framing error flag (FE)	0: No error 1: Framing error	0	○	×
6	Summing error flag (SE)	0: (OE) U (PE) U (FE) = 0 1: (OE) U (PE) U (FE) = 1	0	○	×
7	Nothing is arranged for this bit. This bit is a write disabled bit. When this bit is read out, the contents are “1”.		1	○	×

Fig. 2.4.3 Structure of Serial I/O1 status register and Serial I/O3 status register

Serial I/O1 control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O1 control register
(SIO1CON: address 001A16)

b	Name	Functions	At reset	R	W
0	BRG count source selection bit (CSS)	0: f(XIN) 1: f(XIN)/4	0	○	○
1	Serial I/O1 synchronous clock selection bit (SCS)	When clock synchronous serial I/O is selected, 0: BRG output divided by 4 1: External clock input When UART is selected, 0: BRG output divided by 16 1: External clock input divided by16	0	○	○
2	$\overline{SRDY1}$ output enable bit (SRDY)	0: I/O port (P47) 1: $\overline{SRDY1}$ output pin	0	○	○
3	Transmit interrupt source selection bit (TIC)	0: Transmit buffer empty 1: Transmit shift operation completion	0	○	○
4	Transmit enable bit (TE)	0: Transmit disabled 1: Transmit enabled	0	○	○
5	Receive enable bit (RE)	0: Receive disabled 1: Receive enabled	0	○	○
6	Serial I/O1 mode selection bit (SIOM)	0: UART 1: Clock synchronous serial I/O	0	○	○
7	Serial I/O1 enable bit (SIOE)	0: Serial I/O1 disabled (P44 to P47: normal I/O pins) 1: Serial I/O1 enabled (P44 to P47: Serial I/O pins)	0	○	○

Fig. 2.4.4 Structure of Serial I/O1 control register

Serial I/O3 control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O3 control register
(SIO3CON: address 0032_{16})

b	Name	Functions	At reset	R	W
0	BRG count source selection bit (CSS)	0: f(X_{IN}) 1: f(X_{IN})/4	0	○	○
1	Serial I/O3 synchronous clock selection bit (SCS)	When clock synchronous serial I/O is selected, 0: BRG output divided by 4 1: External clock input When UART is selected, 0: BRG output divided by 16 1: External clock input divided by16	0	○	○
2	$\overline{S_{RDY3}}$ output enable bit (SRDY)	0: I/O port ($P3_7$) 1: $\overline{S_{RDY3}}$ output pin	0	○	○
3	Transmit interrupt source selection bit (TIC)	0: Transmit buffer empty 1: Transmit shift operation completion	0	○	○
4	Transmit enable bit (TE)	0: Transmit disabled 1: Transmit enabled	0	○	○
5	Receive enable bit (RE)	0: Receive disabled 1: Receive enabled	0	○	○
6	Serial I/O3 mode selection bit (SIOM)	0: UART 1: Clock synchronous serial I/O	0	○	○
7	Serial I/O3 enable bit (SIOE)	0: Serial I/O3 disabled ($P3_4$ to $P3_7$: normal I/O pins) 1: Serial I/O3 enabled ($P3_4$ to $P3_7$: Serial I/O pins)	0	○	○

Fig. 2.4.5 Structure of Serial I/O3 control register

UART1 control register

b7 b6 b5 b4 b3 b2 b1 b0

UART1 control register
(UART1CON: address 001B16)

b	Name	Functions	At reset	R	W
0	Character length selection bit (CHAS)	0: 8 bits 1: 7 bits	0	○	○
1	Parity enable bit (PARE)	0: Parity checking disabled 1: Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0: Even parity 1: Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0: 1 stop bit 1: 2 stop bits	0	○	○
4	P45/TxD1 P-channel output disable bit (POFF)	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "1".		1	○	×
6			1	○	×
7			1	○	×

Fig. 2.4.6 Structure of UART1 control register

UART3 control register

b7 b6 b5 b4 b3 b2 b1 b0

UART3 control register
(UART3CON: address 003316)

b	Name	Functions	At reset	R	W
0	Character length selection bit (CHAS)	0: 8 bits 1: 7 bits	0	○	○
1	Parity enable bit (PARE)	0: Parity checking disabled 1: Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0: Even parity 1: Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0: 1 stop bit 1: 2 stop bits	0	○	○
4	P35/TxD3 P-channel output disable bit (POFF)	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "1".		1	○	×
6			1	○	×
7			1	○	×

Fig. 2.4.7 Structure of UART3 control register

Baud rate generator i (i = 1, 3)

b7 b6 b5 b4 b3 b2 b1 b0

Baud rate generator i
(BRGi (i=1, 3): address 001C16, 002F16)

b	Functions	At reset	R	W
0	Set a count value of baud rate generator.	Undefined	O	O
1		Undefined	O	O
2		Undefined	O	O
3		Undefined	O	O
4		Undefined	O	O
5		Undefined	O	O
6		Undefined	O	O
7		Undefined	O	O

Note: Write to this register while transmit/receive operation is stopped.

Fig. 2.4.8 Structure of Baud rate generator 1 and Baud rate generator 3

Serial I/O2 control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O2 control register
(SIO2CON: address 001D16)

b	Name	Functions	At reset	R	W
0	Internal synchronous clock selection bits	b2b1b0 0 0 0: f(XIN)/8 0 0 1: f(XIN)/16 0 1 0: f(XIN)/32 0 1 1: f(XIN)/64 1 1 0: f(XIN)/128 1 1 1: f(XIN)/256	0	O	O
1			0	O	O
2			0	O	O
3	Serial I/O2 port selection bit	0: I/O port (P51, P52) 1: SOUT2, SCLK2 signal output	0	O	O
4	$\overline{S_{RDY2}}$ output enable bit	0: I/O port (P53) 1: $\overline{S_{RDY2}}$ signal output	0	O	O
5	Transfer direction selection bit	0: LSB first 1: MSB first	0	O	O
6	Serial I/O2 synchronous clock selection bit	0: External clock 1: Internal clock	0	O	O
7	P51/SOUT2 P-channel output disable bit	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	O	O

Fig. 2.4.9 Structure of Serial I/O2 control register

Serial I/O2 register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O2 register
(SIO2: address 001F16)

b	Name	Functions	At reset	R	W
0		This register becomes shift register. At transmit: Set transmit data to this register. At receive: Received data is stored to this register.	Undefined	○	○
1			Undefined	○	○
2			Undefined	○	○
3			Undefined	○	○
4			Undefined	○	○
5			Undefined	○	○
6			Undefined	○	○
7			Undefined	○	○

Fig. 2.4.10 Structure of Serial I/O2 register

Interrupt source selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt source selection register
(INTSEL: address 003916)

b	Name	Functions	At reset	R	W
0	INT_0/Timer Z interrupt source selection bit (*1)	0: INT_0 interrupt 1: Timer Z interrupt	0	○	○
1	Serial I/O2/Timer Z interrupt source selection bit (*1)	0: Serial I/O2 interrupt 1: Timer Z interrupt	0	○	○
2	Serial I/O1 transmit/ SCL, SDA interrupt source selection bit (*2)	0: Serial I/O1 transmit interrupt 1: SCL, SDA interrupt	0	○	○
3	$CNTR_0$/SCL, SDA interrupt source selection bit (*2)	0: $CNTR_0$ interrupt 1: SCL, SDA interrupt	0	○	○
4	INT_4/$CNTR_2$ interrupt source selection bit	0: INT_4 interrupt 1: $CNTR_2$ interrupt	0	○	○
5	INT_2/I^2C interrupt source selection bit	0: INT_2 interrupt 1: I^2C interrupt	0	○	○
6	$CNTR_1$/Serial I/O3 receive interrupt source selection bit	0: $CNTR_1$ interrupt 1: Serial I/O3 receive interrupt	0	○	○
7	AD converter/Serial I/O3 transmit interrupt source selection bit	0: A/D converter interrupt 1: Serial I/O3 transmit interrupt	0	○	○

*1: Do not write 1 to these bits simultaneously.
*2: Do not write 1 to these bits simultaneously.

Fig. 2.4.11 Structure of Interrupt source selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1
(IREQ1 : address 003C16)

b	Name	Functions	At reset	R	W
0	INT0/Timer Z interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O1 transmit/SCL, SDA interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻: 0 can be set by software, but 1 cannot be set.

Fig. 2.4.12 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2
(IREQ2 : address 003D16)

b	Name	Functions	At reset	R	W
0	CNTR0/SCL, SDA interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	CNTR1/Serial I/O3 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O2/Timer Z interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT2/I2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	INT4/CNTR2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	AD converter/Serial I/O3 transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are 0 .		0	○	×

✻: 0 can be set by software, but 1 cannot be set.

Fig. 2.4.13 Structure of Interrupt request register 2

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1
(ICON1 : address 003E16)

b	Name	Functions	At reset	R	W
0	INT0/Timer Z interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit/SCL, SDA interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.4.14 Structure of Interrupt control register 1

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0
0

Interrupt control register 2
(ICON2 : address 003F16)

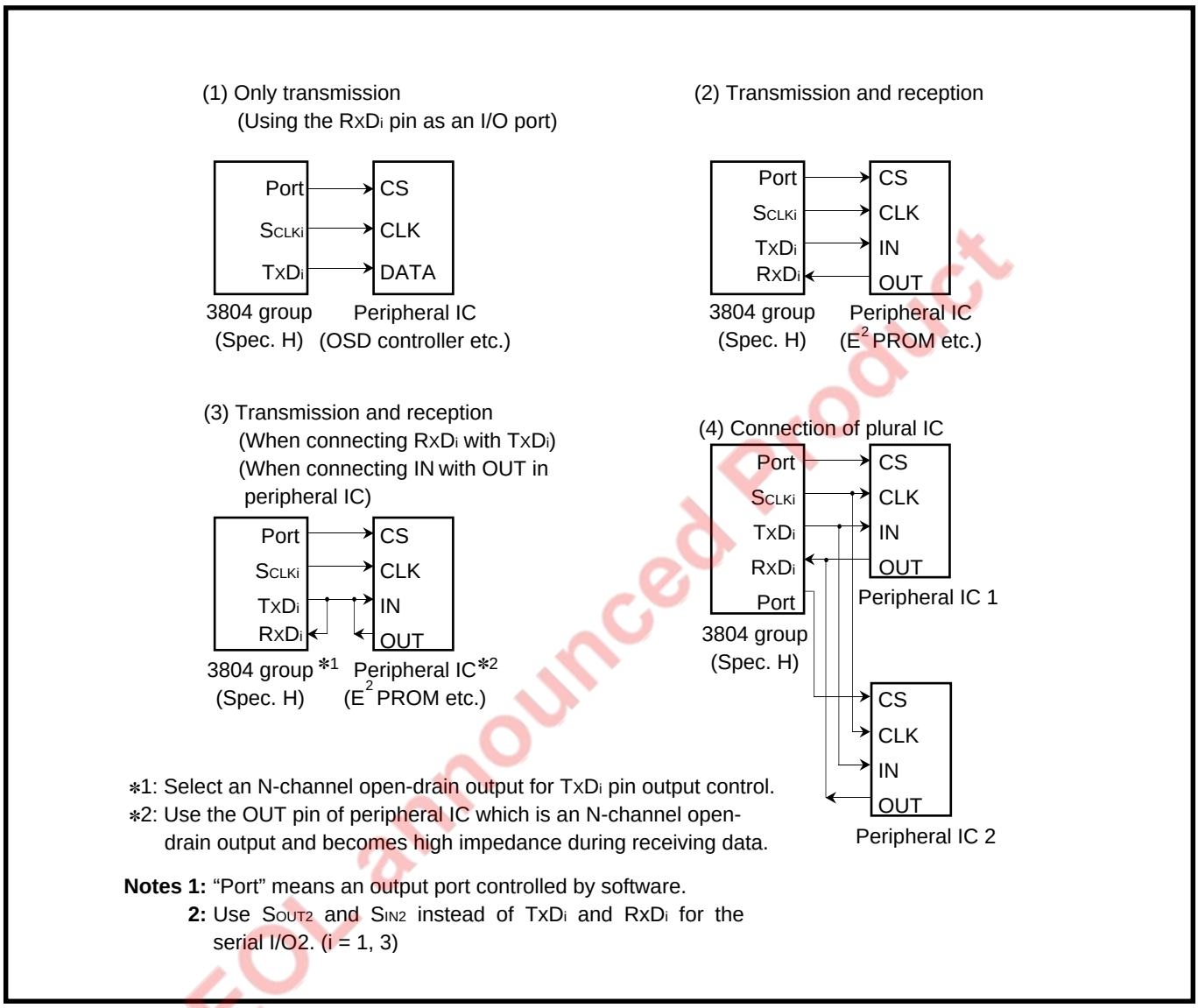
b	Name	Functions	At reset	R	W
0	CNTR0/SCL, SDA interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	CNTR1/ Serial I/O3 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/ Timer Z interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT2/I2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT3 interrupt enable bit	0 : Interrupt disabled	0	○	○
5	INT4/CNTR2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/Serial I/O3 transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.4.15 Structure of Interrupt control register 2

2.4.3 Serial I/O connection examples

(1) Control of peripheral IC equipped with CS pin

Figure 2.4.16 shows connection examples of a peripheral IC equipped with the CS pin. There are connection examples using a clock synchronous serial I/O mode.



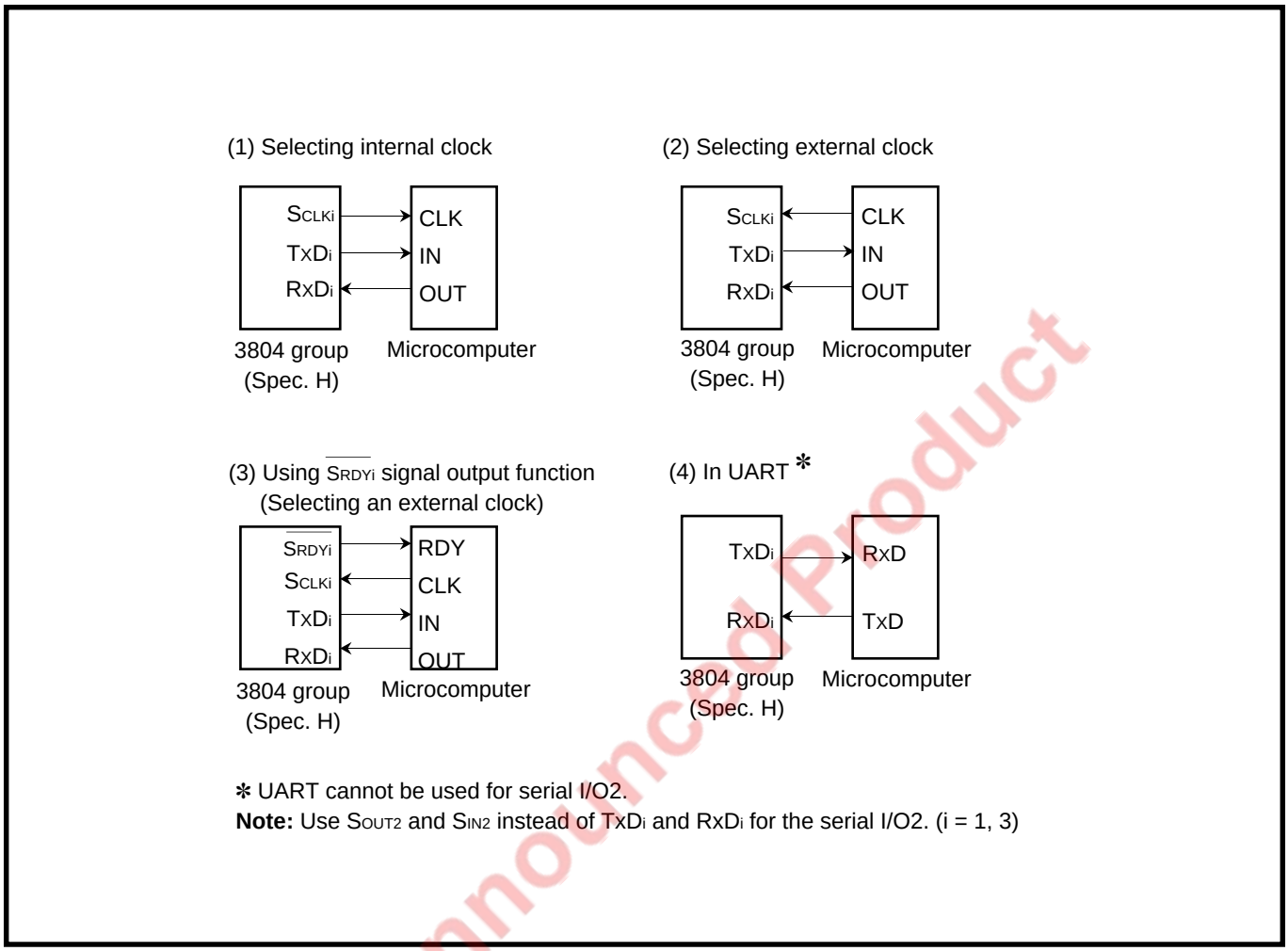
∗1: Select an N-channel open-drain output for TxDi pin output control.
∗2: Use the OUT pin of peripheral IC which is an N-channel open-drain output and becomes high impedance during receiving data.

Notes 1: "Port" means an output port controlled by software.
2: Use SOUT2 and SIN2 instead of TxDi and RxDi for the serial I/O2. (i = 1, 3)

Fig. 2.4.16 Serial I/O connection examples (1)

(2) Connection with microcomputer

Figure 2.4.17 shows connection examples with another microcomputer.



* UART cannot be used for serial I/O2.

Note: Use S_{OUT2} and S_{IN2} instead of TxD_i and RxD_i for the serial I/O2. (i = 1, 3)

Fig. 2.4.17 Serial I/O connection examples (2)

2.4.4 Setting of serial I/O transfer data format

A clock synchronous or clock asynchronous (UART) can be selected as a data format of serial I/O1 and serial I/O3.
Serial I/O2 operates in a clock synchronous.
Figure 2.4.18 shows the serial I/O transfer data format.

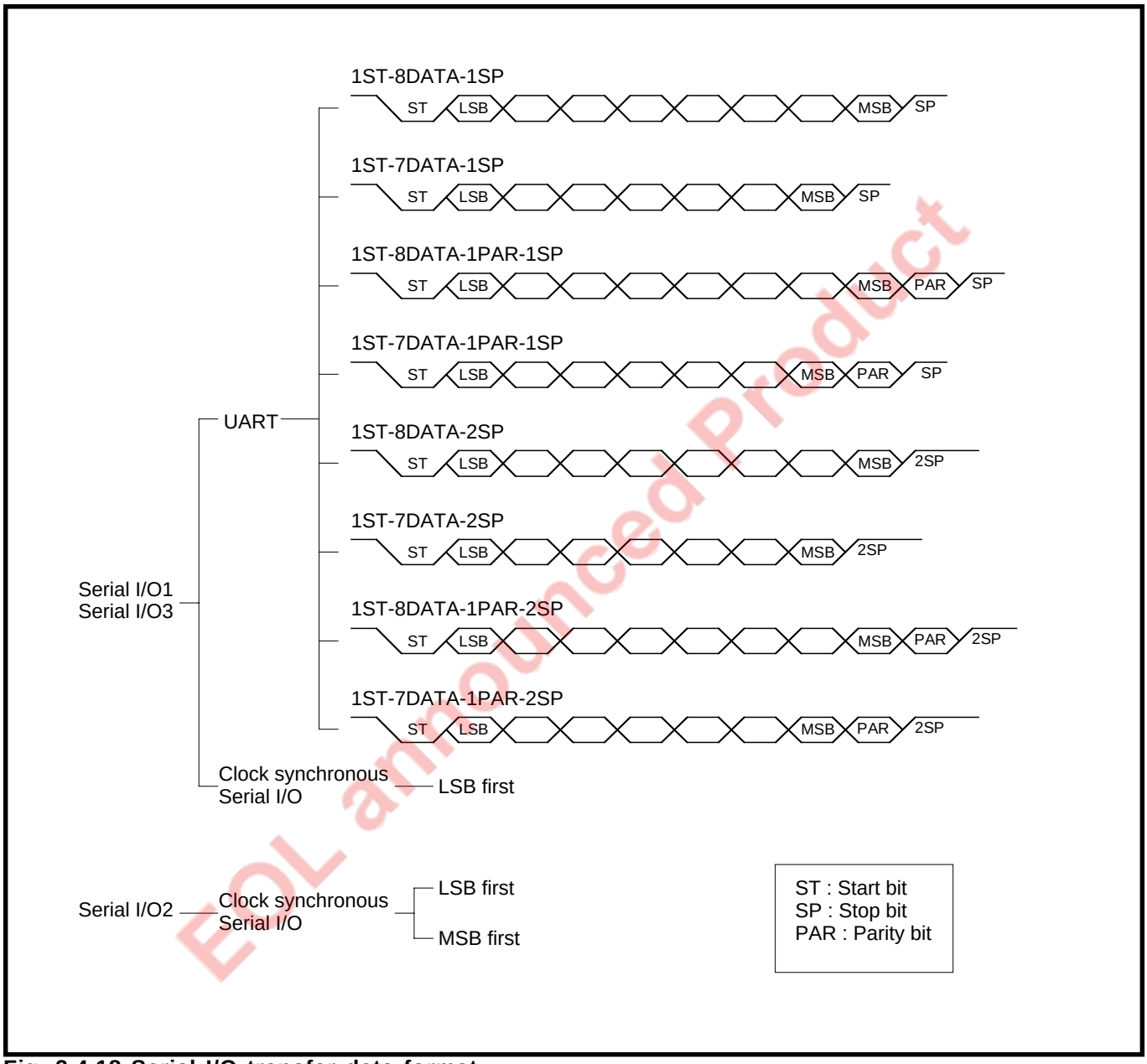


Fig. 2.4.18 Serial I/O transfer data format

2.4.5 Serial I/O1, serial I/O3 operation: stop and initialize

Serial I/O1 and serial I/O3 perform the same operation. In the following explanations when names of serial I/O1 and serial I/O3 are different, serial I/O1s' are showed first and then serial I/O3s' in the marked ( ).

(1) Clock synchronous serial I/O mode

■Stop/initialize transmit operation when only transmitting

When using an internal clock, set the transmit enable bit and serial I/O1 enable bit (serial I/O3 enable bit) to "0".
When using an external clock, set the transmit enable bit to "0".

By setting the transmit enable bit to "0", the transmit operations listed below will be stopped or initialized. However, when using an internal clock, the clock is output in 8 pulses, even if the transmit enable bit is set to "0" during transmit operations.

- Stop supply of shift clock to transmit shift register
- Initialize clock control circuit for transmit
- Transmit buffer empty flag = "0"
- Transmit shift register shift complete flag = "0"
- $P4_5/TxD_1$ pin: input/output port $P4_5$ ($P3_5/TxD_3$ pin: input/output port $P3_5$)

By setting the serial I/O1 enable bit (serial I/O3 enable bit) to "0", pins $P4_4/RxD_1$, $P4_5/TxD_1$, $P4_6/S_{CLK1}$, and $P4_7/\overline{S_{RDY1}}$ ($P3_4/RxD_3$, $P3_5/TxD_3$, $P3_6/S_{CLK3}$, $P3_7/\overline{S_{RDY3}}$ pins) all become I/O ports. As a result, the internal clock cannot be output externally.

■Stop/initialize receive operation when only receiving

When using an internal clock, set the receive enable bit and serial I/O1 enable bit (serial I/O3 enable bit) to "0".
When using an external clock, set the receive enable bit or serial I/O1 enable bit (serial I/O3 enable bit) to "0".

By setting the receive enable bit to "0", the receive operations listed below will be stopped or initialized. However, when using an internal clock, the clock is output in 8 pulses, even if the receive enable bit is set to "0" during receive operations.

- Stop supply of shift clock to receive shift register
- Initialize clock control circuit for receive
- Error flags (over-run, parity, framing, and summing error flags) = "0"
- Receive buffer full flag = "0"
- $P4_4/RxD_1$ pin: input/output port $P4_4$ ($P3_4/RxD_3$ pin: input/output port $P3_4$)

By setting the serial I/O1 enable bit (serial I/O3 enable bit) to "0", the receive operations listed below will be stopped or initialized. As a result, the internal clock cannot be output externally.

- Stop supply of shift clock to receive shift register
- Initialize clock control circuit for receive
- Error flags (over-run, parity, framing, and summing error flags) = "0"
- Receive buffer full flag = "0"
- $P4_4/RxD_1$, $P4_5/TxD_1$, $P4_6/S_{CLK1}$, $P4_7/\overline{S_{RDY1}}$ pins: I/O ports $P4_4$, $P4_5$, $P4_6$, $P4_7$
  ($P3_4/RxD_3$, $P3_5/TxD_3$, $P3_6/S_{CLK3}$, $P3_7/\overline{S_{RDY3}}$ pins: I/O ports $P3_4$, $P3_5$, $P3_6$, $P3_7$)

■Stop/initialize receive/transmit operation when both receiving and transmitting

Set the transmit enable bit and receive enable bit to "0" simultaneously.
When using an internal clock, also set the serial I/O1 enable bit (serial I/O3 enable bit) to "0".

(2) UART Mode

■Stop/initialize transmit operation

Set the transmit enable bit to "0".

■Stop/initialize receive operation

Set the receive enable bit to "0".

2.4.6 Serial I/O pin function and selection method

(1) Serial I/O1, serial I/O3

Table 2.4.1 shows the pin function in the clock synchronous serial I/O mode, Table 2.4.2 shows the pin function in the UART mode.

Table 2.4.1 Pin function in clock synchronous serial I/O mode

Pin name (Serial I/O1)	Pin name (Serial I/O3)	Function	Serial I/O1 control register (address 001A16) Serial I/O3 control register (address 003216)								Corresponding direction register
			b7 (Note 1)	b6	b5	b4	b3	b2	b1	b0	
			SIOE	SIOM	RE	TE	TIC	SRDY	SCS	CSS	
P44/RxD1	P34/RxD3	RxD1, RxD3	1	1	1	1	×	×	×	×	×
		P44, P34	1	1	0	×	×	×	×	×	0/1
P45/TxD1	P35/TxD3	TxD1, TxD3	1	1	×	1	×	×	×	×	×
		P45, P35	1	1	×	0	×	×	×	×	0/1
P46/SCLK1	P36/SCLK3	SCLK1 (External clock input)	1	1	×	1	×	×	1	×	×
		SCLK1 (Internal clock output)	1	1	×	1	×	×	0	×	×
P47/SRDY1 /CNTR2 (Note 2)	P37/SRDY3	SRDY1, SRDY3	1	1	1	1	×	1	×	×	×
		P47, P37	1	1	×	×	×	0	×	×	0/1

Note 1: When SIOE is set to "0", all pins become I/O ports regardless of set value of b6–b0.
Note 2: In the pulse output mode, the programmable waveform generating mode, or the programmable one-shot generating mode of the timer Z, this pin functions as the timer Z function output pin regardless of b7-b0 setting.
×: This is not used for the pin's function setting.

Table 2.4.2 Pin function in UART mode

Pin name (Serial I/O1)	Pin name (Serial I/O3)	Function	Serial I/O1 control register (address 001A16)								Corresponding direction register
			b7 (Note 1)	b6	b5	b4	b3	b2	b1	b0	
			SIOE	SIOM	RE	TE	TIC	SRDY	SCS	CSS	
P44/RxD1	P34/RxD3	RxD	1	0	1	×	×	×	×	×	×
		P44	1	0	0	×	×	×	×	×	0/1
P45/TxD1	P35/TxD3	TxD	1	0	×	1	×	×	×	×	×
		P45	1	0	×	0	×	×	×	×	0/1
P46/SCLK1	P36/SCLK3	SCLK1 (External clock input)	1	0	×	×	×	×	1	×	×
		P46	1	0	×	×	×	×	0	×	0/1
P47/SRDY1 /CNTR2 (Note 2)	P37/SRDY3	P47	1	0	×	×	×	×	×	×	0/1

Note 1: When SIOE is set to "0", all pins become I/O ports regardless of set value of b6–b0.
Note 2: In the pulse output mode, the programmable waveform generating mode, or the programmable one-shot generating mode of the timer Z, this pin functions as the timer Z function output pin regardless of b7-b0 setting.
×: This is not used for the pin's function setting.

(2) Serial I/O2

Table 2.4.3 shows the pin function in the clock synchronous serial I/O mode.

Table 2.4.3 Pin function in clock synchronous serial I/O mode

Pin name	Function		Serial I/O2 control register (address 001D16)								Corresponding direction register
			b7	b6	b5	b4	b3	b2	b1	b0	
P50/SIN2	SIN2 (Note 1)		×	×	×	×	1	×	×	×	0
	P50		×	×	×	×	×	×	×	×	0/1
P51/SOUT2	SOUT2	CMOS output	0	×	×	×	1	×	×	×	×
		N-channel open-drain output	1	×	×	×	1	×	×	×	×
	P51	(Note 3)	×	×	×	0	×	×	×	0/1	
P52/SCLK2	SCLK2 (External clock input) (Note 2)		×	0	×	×	1	×	×	×	×
	SCLK2 (Internal clock output)		×	1	×	×	1	×	×	×	×
	P52		×	×	×	×	0	×	×	×	0/1
P53/SRDY2	SRDY2		×	×	×	1	×	×	×	×	×
	P53		×	×	×	0	×	×	×	×	0/1

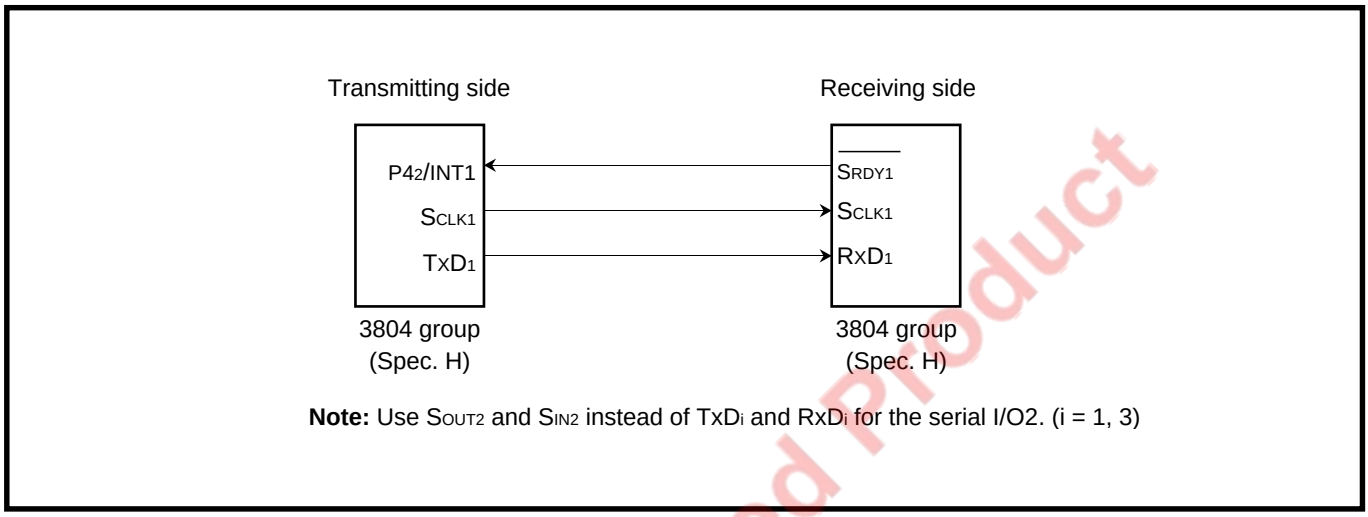
Notes 1: Although this pin functions as SIN2 when b3 is set to "0", set "1" to b3.
Notes 2: Although this pin functions as SCLK2 when b3 and the corresponding direction register are set to "0", set "1" to b3.
Notes 3: When the corresponding direction register bit is "1", the b7 setting is valid.
×: This is not used for the pin's function setting.

2.4.7 Serial I/O application examples

(1) Communication using clock synchronous serial I/O (transmit/receive)

Outline : 2-byte data is transmitted and received, using the clock synchronous serial I/O. The $\overline{S_{RDY1}}$ signal is used for communication control.

Figure 2.4.19 shows a connection diagram, and Figure 2.4.20 shows a timing chart.
Figure 2.4.21 shows a registers setting relevant to the transmitting side, and Figure 2.4.22 shows registers setting relevant to the receiving side.



Note: Use S_{OUT2} and S_{IN2} instead of TxD_i and RxD_i for the serial I/O2. (i = 1, 3)

Fig. 2.4.19 Connection diagram

Specifications :
- Serial I/O is used (clock synchronous serial I/O is selected.)
- Synchronous clock frequency : 125 kHz (f(X_{IN}) = 4 MHz is divided by 32)
- $\overline{S_{RDY1}}$ (receivable signal) is used.
- The receiving side outputs $\overline{S_{RDY1}}$ signal at intervals of 2 ms (generated by timer), and 2-byte data is transferred from the transmitting side to the receiving side.

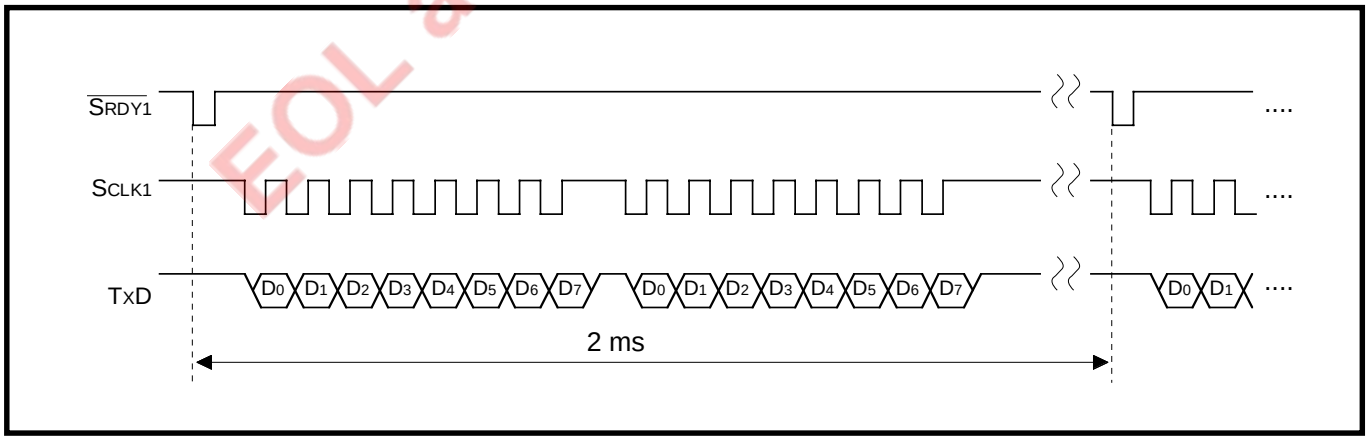


Fig. 2.4.20 Timing chart (using clock synchronous serial I/O)

Transmitting side

Port P4 direction register (Address : 0009_{16})

P4D

b7							b0
					0		

→ Port $P4_2/INT_1$: Input mode

Serial I/O1 status register (Address : 0019_{16})

SIO1STS

b7							b0

→ Transmit buffer empty flag
- Confirm that the data has been transferred from Transmit buffer register to Transmit shift register.
- When this flag is "1", it is possible to write the next transmission data in to Transmit buffer register.

→ Transmit shift register shift completion flag
Confirm completion of transmitting 1-byte data with this flag.
"1" : Transmit shift completed

Serial I/O1 control register (Address : $001A_{16}$)

SIO1CON

b7							b0
1	1	0	1			0	0

→ BRG count source : $f(X_{IN})$
→ Serial I/O1 synchronous clock : BRG/4
→ Transmit enabled
→ Receive disabled
→ Clock synchronous serial I/O
→ Serial I/O1 enabled

Baud rate generator 1 (Address : $001C_{16}$)

BRG1

b7							b0
8 – 1							

Set "division ratio – 1"

Interrupt edge selection register (Address : $003A_{16}$)

INTEDGE

b7							b0
						0	

→ INT_1 falling edge active

Fig. 2.4.21 Registers setting relevant to transmitting side

Receiving side

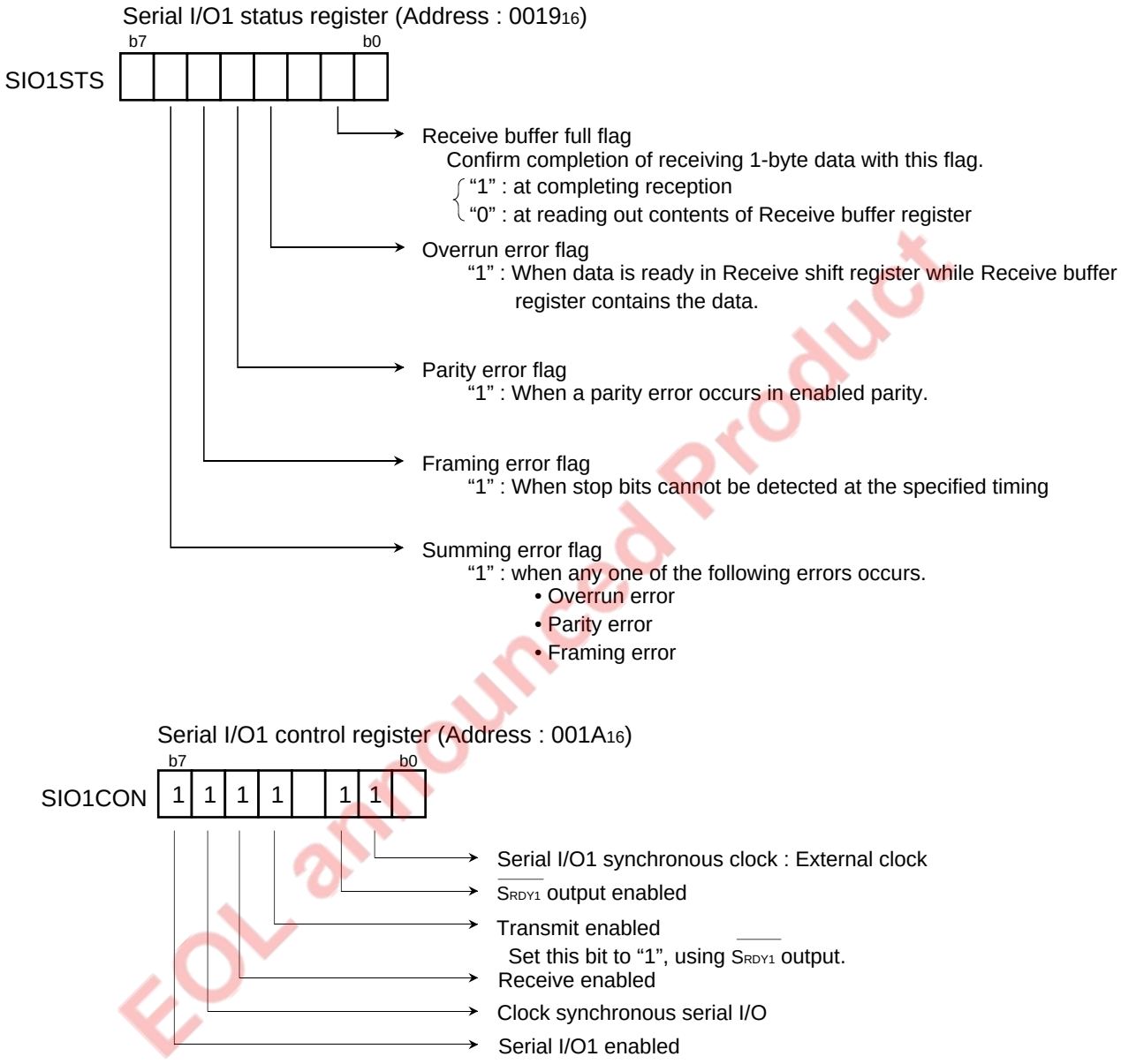


Fig. 2.4.22 Registers setting relevant to receiving side

Figure 2.4.23 shows a control procedure of the transmitting side, and Figure 2.4.24 shows a control procedure of the receiving side.

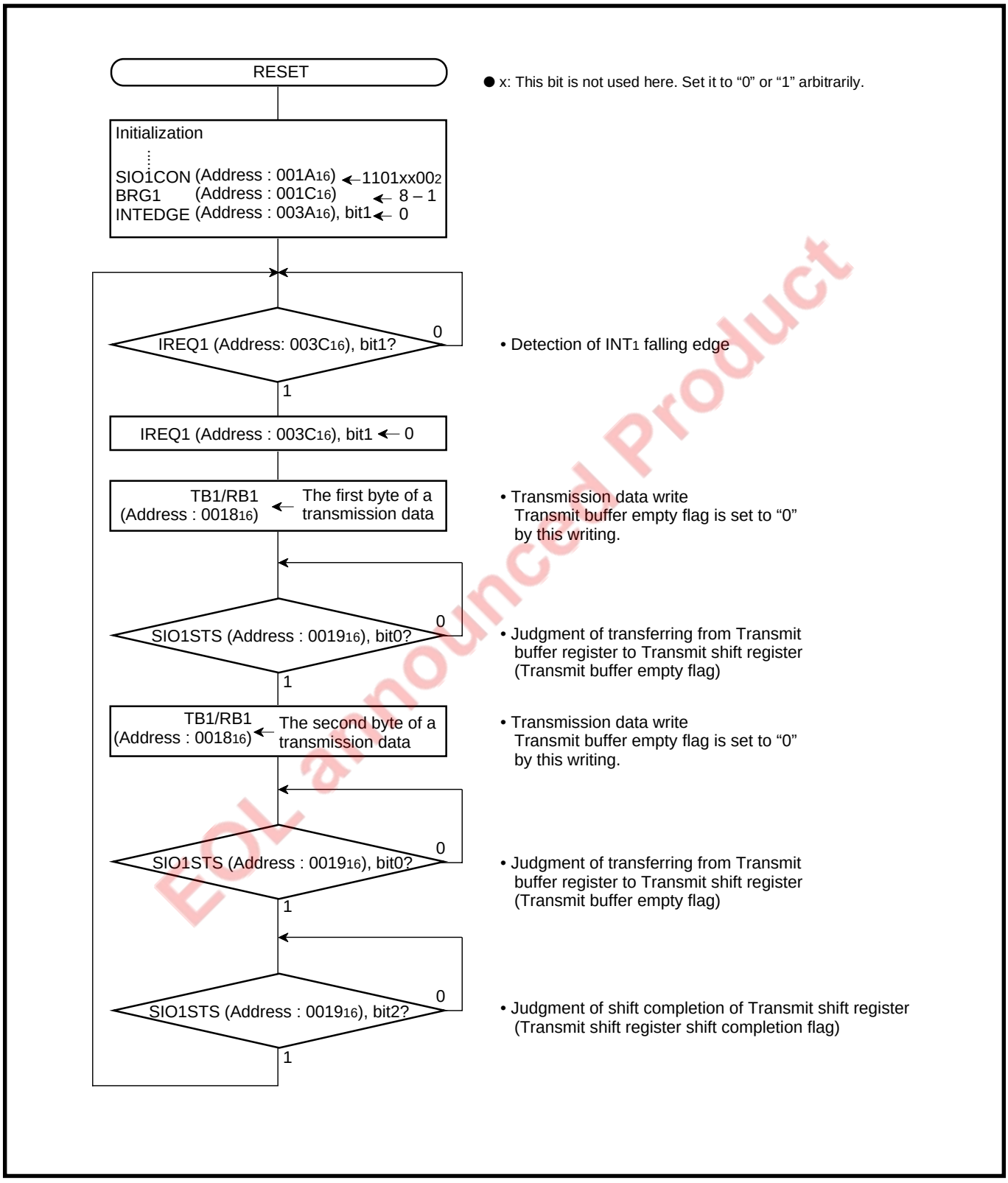


Fig. 2.4.23 Control procedure of transmitting side

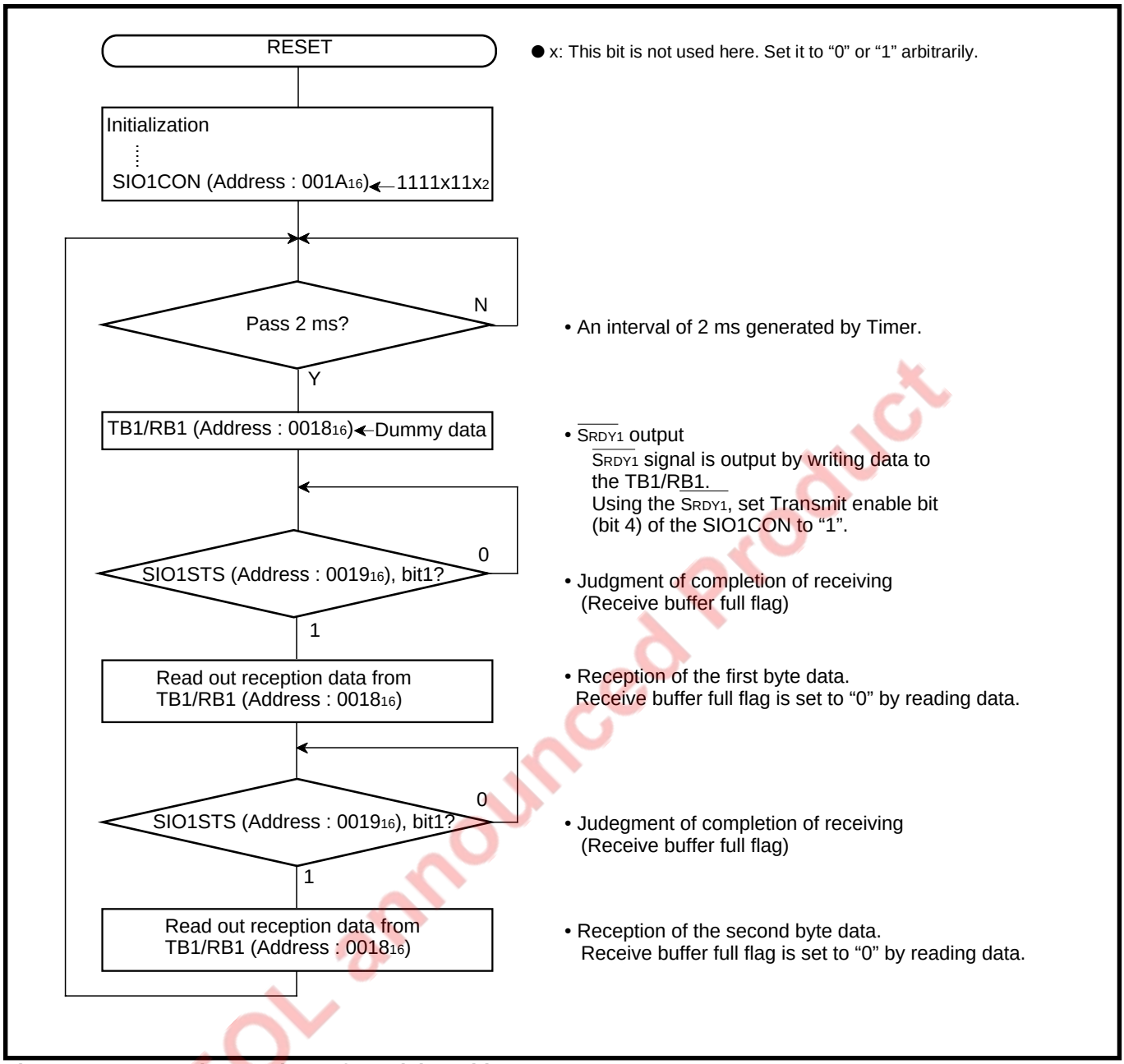


Fig. 2.4.24 Control procedure of receiving side

(2) Output of serial data (control of peripheral IC)

Outline : 4-byte data is transmitted and received, using the clock synchronous serial I/O. The $\overline{CS}$ signal is output to a peripheral IC through port $P6_3$.

Figure 2.4.25 shows connection diagrams of example for using serial I/O1 and example for using serial I/O2 with the same specification, and Figure 2.4.26 shows a timing chart.

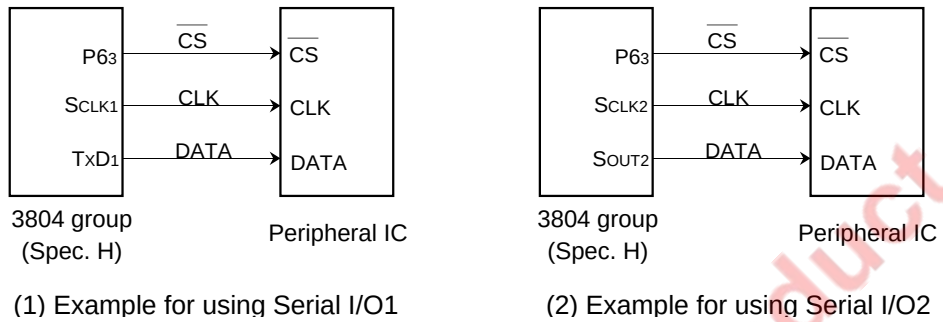
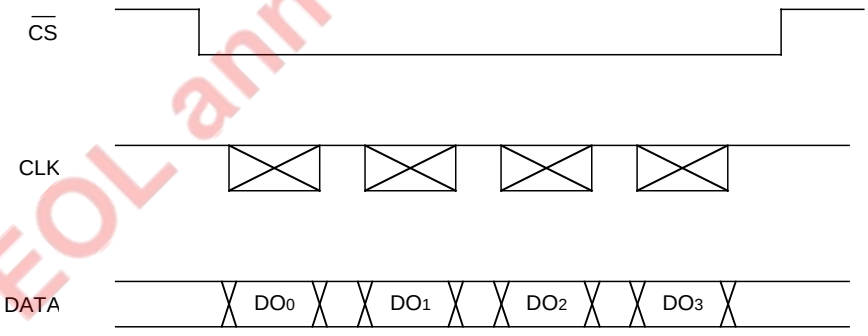


Fig. 2.4.25 Connection diagrams

Specifications :
- Serial I/O is used (clock synchronous serial I/O is selected.)
- Synchronous clock frequency : 125 kHz ($f(X_{IN})$ = 4 MHz is divided by 32)
- Transfer direction : LSB first
- The Serial I/O interrupt is not used.
- Port $P6_3$ is connected to the $\overline{CS}$ pin ("L" active) of the peripheral IC for transmission control; the output level of port $P6_3$ is controlled by software.



Note: When using serial I/O2, the S_{OUT2} pin becomes the high-impedance state after transfer is completed.

Fig. 2.4.26 Timing chart (serial I/O1)

Figure 2.4.27 shows registers setting relevant to serial I/O1, and Figure 2.4.28 shows a setting of serial I/O1 transmission data.

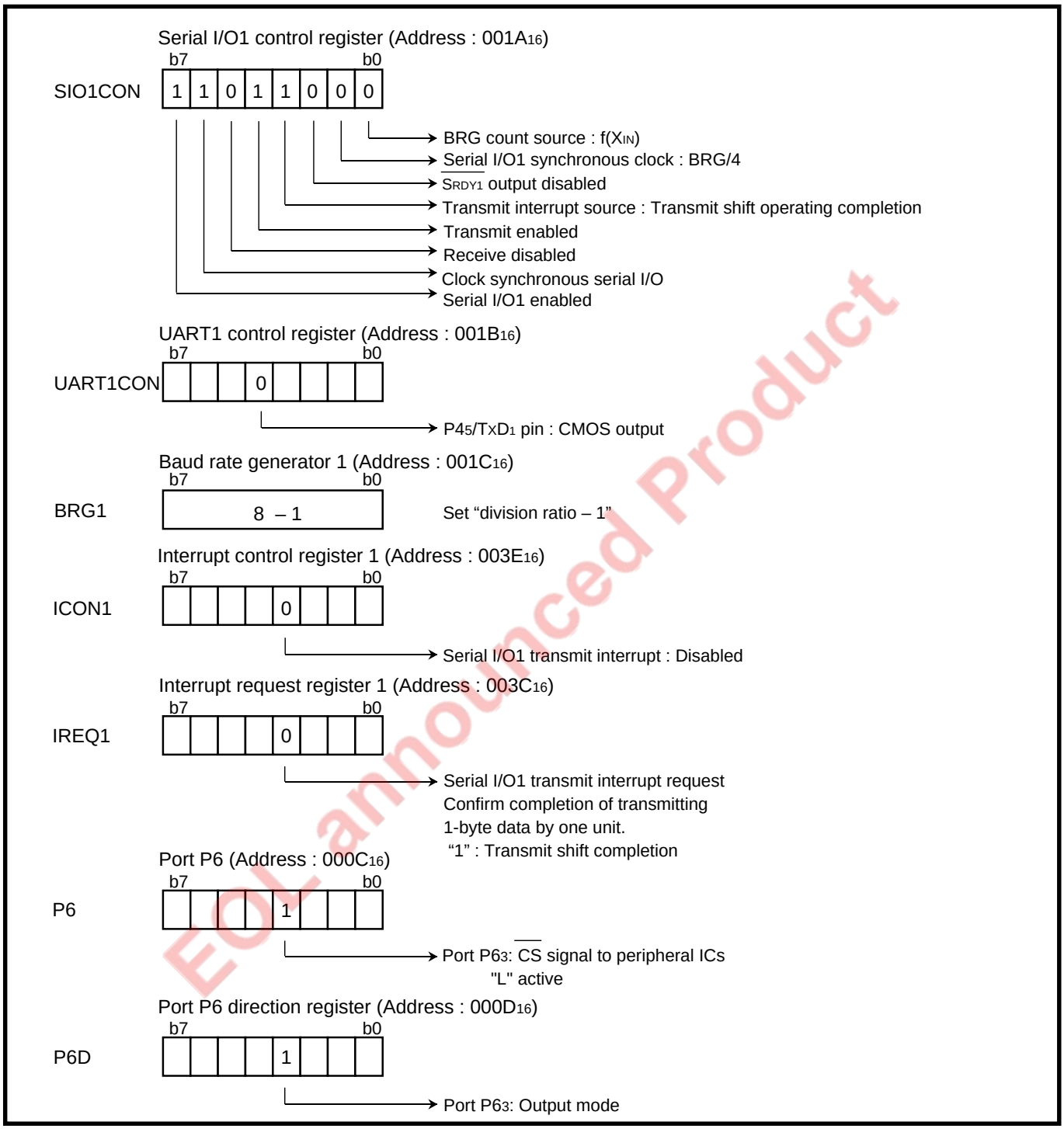


Fig. 2.4.27 Registers setting relevant to serial I/O1

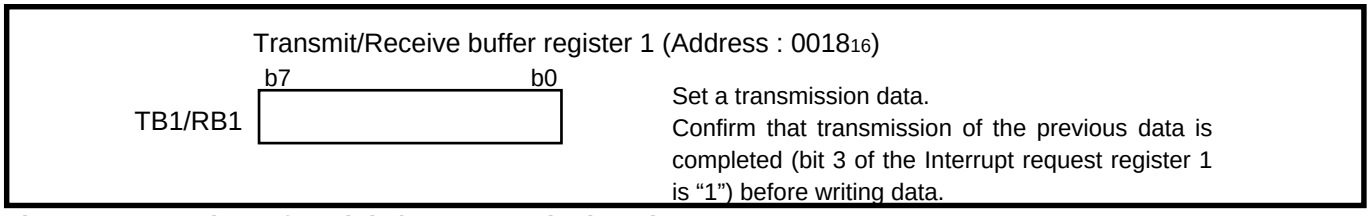


Fig. 2.4.28 Setting of serial I/O1 transmission data

When the registers are set as shown in Fig. 2.4.27, serial I/O1 can transmit 1-byte data by writing data to the transmit buffer register.
Thus, after setting the $\overline{CS}$ signal to "L", write the transmission data to the transmit buffer register by each 1 byte, and return the $\overline{CS}$ signal to "H" when the target number of bytes has been transmitted. Figure 2.4.29 shows a control procedure of serial I/O1.

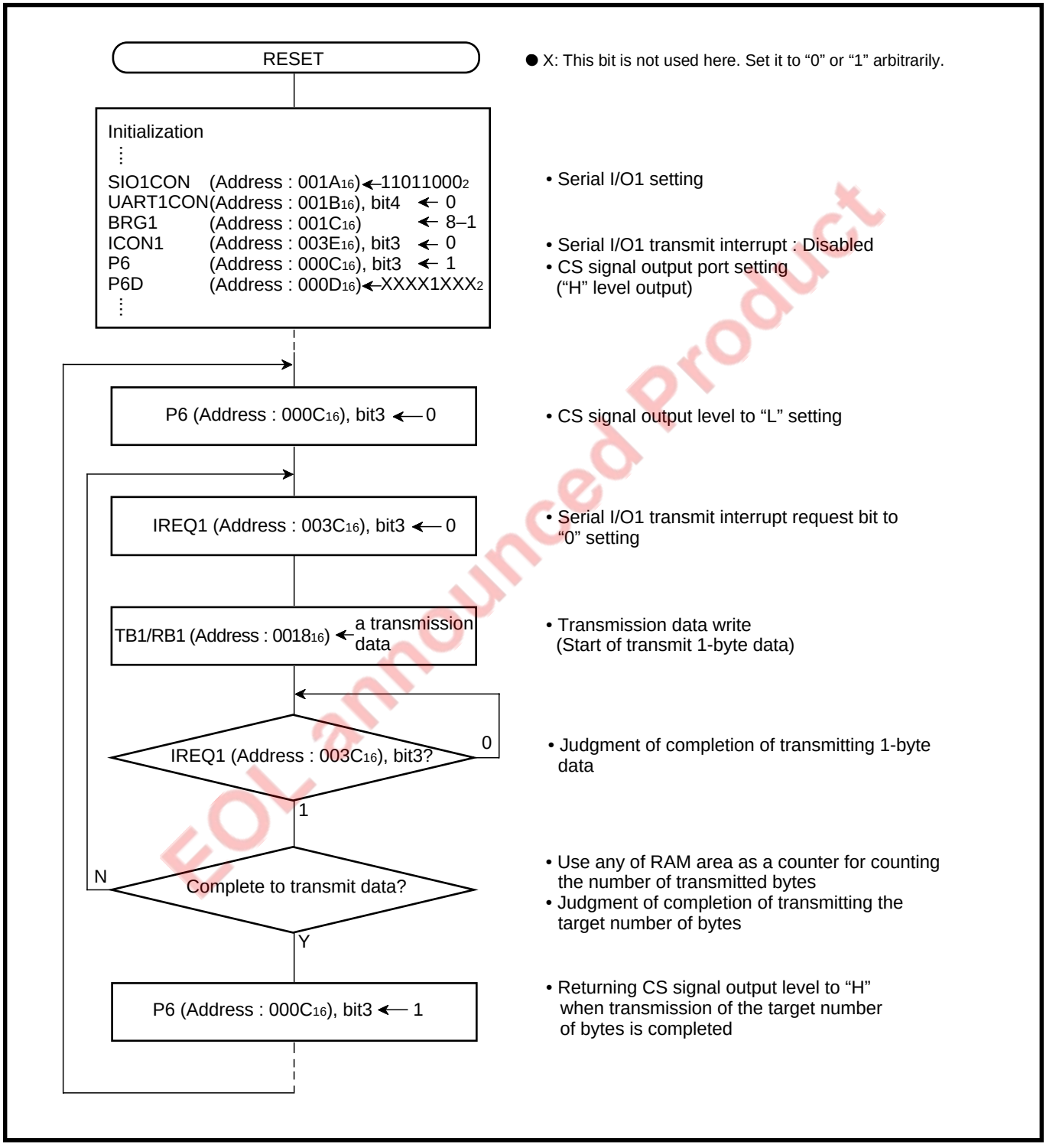


Fig. 2.4.29 Control procedure of serial I/O1

Figure 2.4.30 shows registers setting relevant to serial I/O2, and Figure 2.4.31 shows a setting of serial I/O2 transmission data.

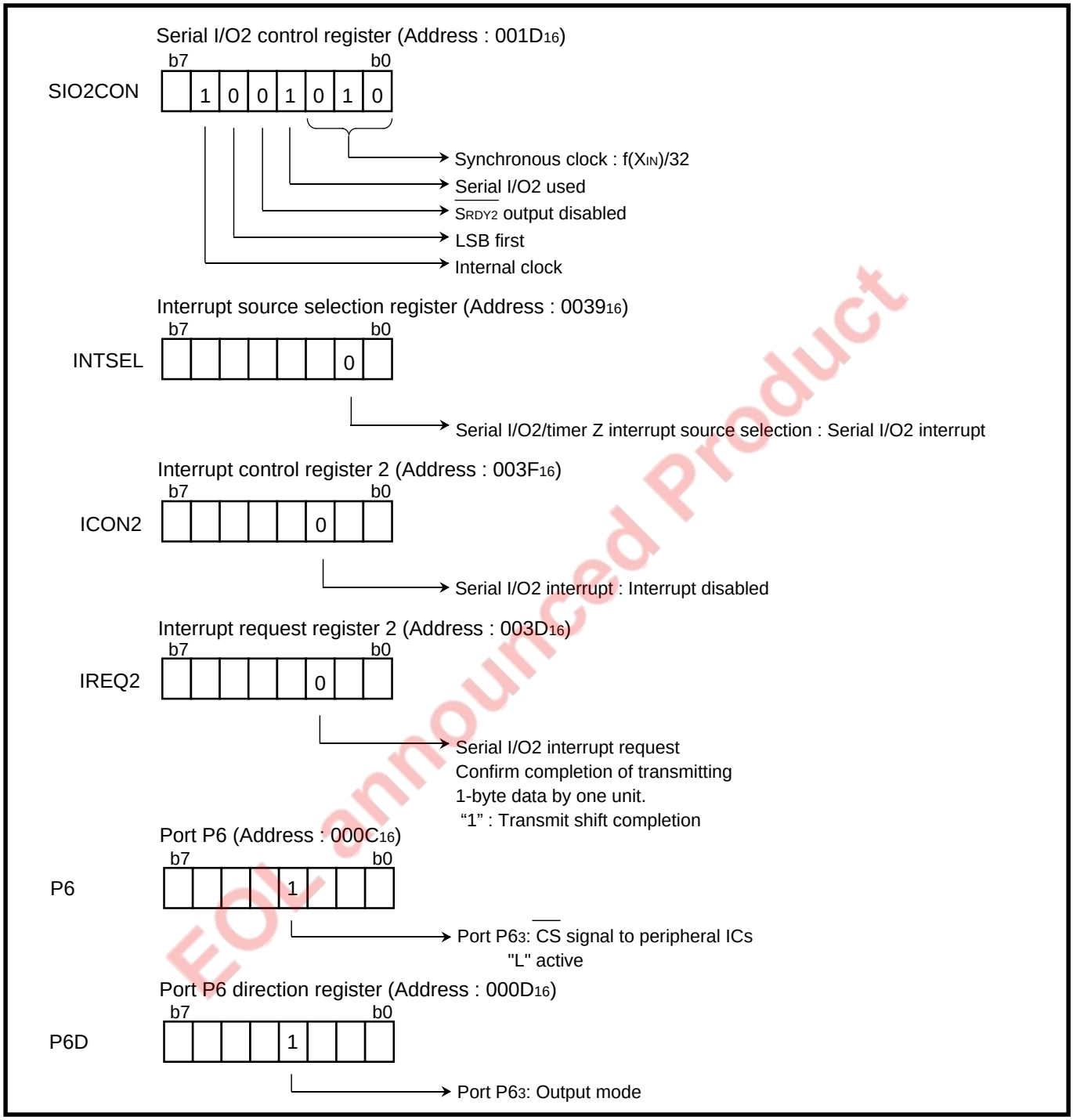


Fig. 2.4.30 Registers setting relevant to serial I/O2

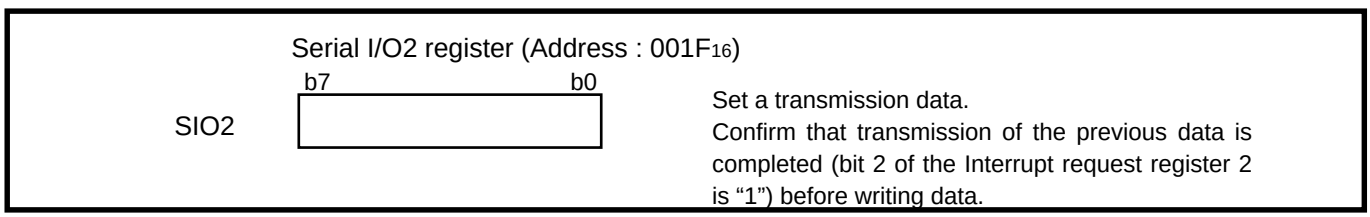


Fig. 2.4.31 Setting of serial I/O2 transmission data

When the registers are set as shown in Fig. 2.4.30, serial I/O2 can transmit 1-byte data by writing data to the serial I/O2 register.
Thus, after setting the CS signal to “L”, write the transmission data to the serial I/O2 register by each 1 byte, and return the CS signal to “H” when the target number of bytes has been transmitted.
Figure 2.4.32 shows a control procedure of serial I/O2.

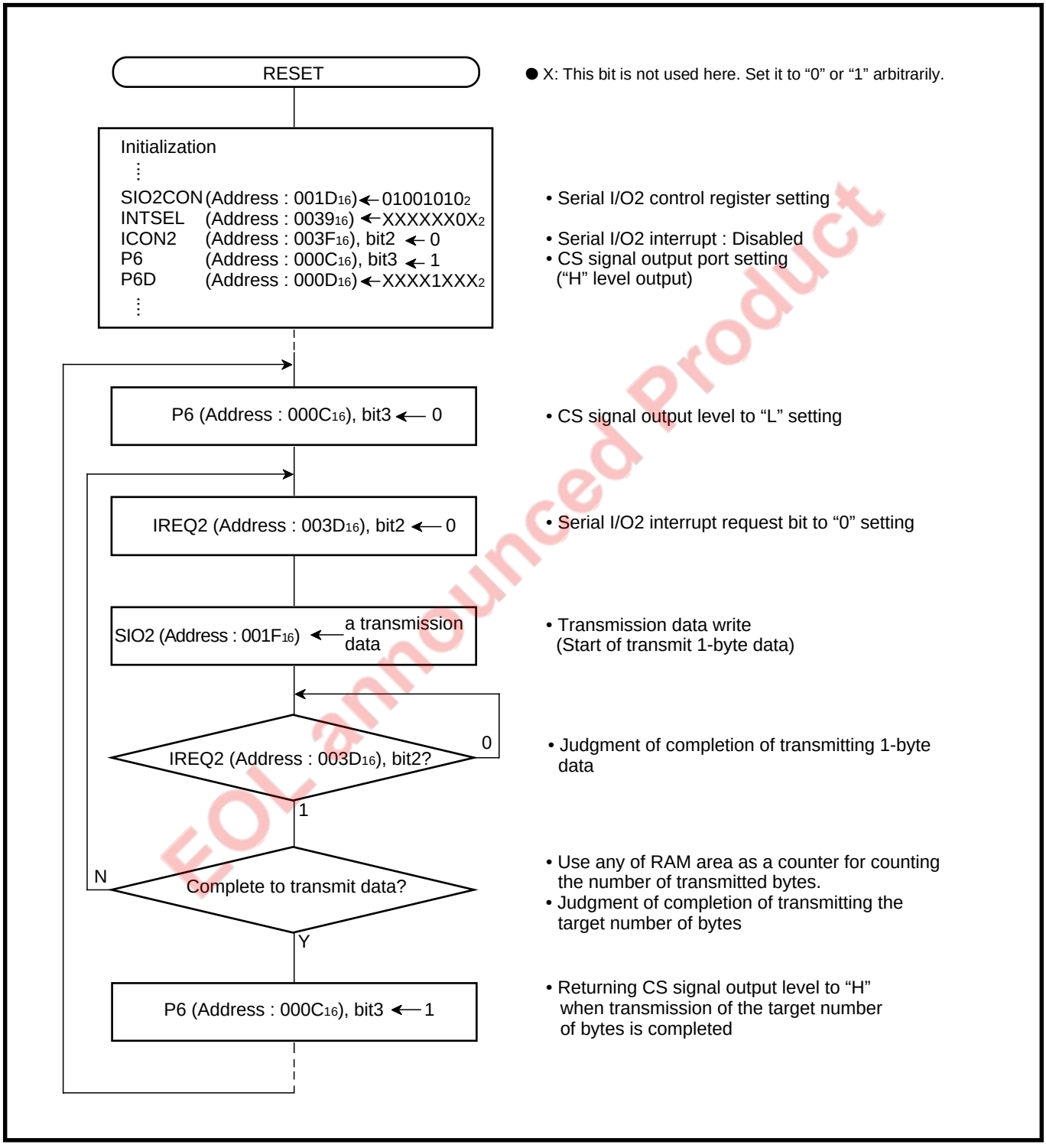


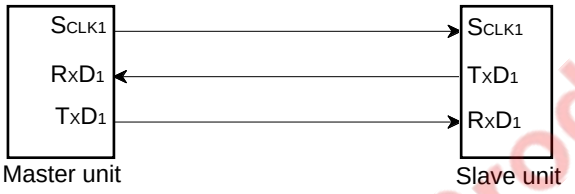
Fig. 2.4.32 Control procedure of serial I/O2

(3) Cyclic transmission or reception of block data (data of specified number of bytes) between two microcomputers

Outline : When the clock synchronous serial I/O is used for communication, synchronization of the clock and the data between the transmitting and receiving sides may be lost because of noise included in the synchronous clock. It is necessary to correct that constantly, using "heading adjustment".

This "heading adjustment" is carried out by using the interval between blocks in this example. This example is described for serial I/O1, but this example also can apply serial I/O3.

Figure 2.4.33 shows connection diagram.



Note: Use S_{OUT2} and S_{IN2} instead of TxD_1 and RxD_1 for serial I/O2.

Fig. 2.4.33 Connection diagram

Specifications :

- Serial I/O is used (clock synchronous serial I/O is selected).
- Synchronous clock frequency : 131 kHz ($f(X_{IN})$ = 4.19 MHz is divided by 32)
- Byte cycle: 488 µs
- Number of bytes for transmission or reception : 8 byte/block
- Block transfer cycle : 16 ms
- Block transfer term : 3.5 ms
- Interval between blocks : 12.5 ms
- Heading adjustment time : 8 ms

Limitations of the specifications :

- Reading of the reception data and setting of the next transmission data must be completed within the time obtained from "byte cycle – time for transferring 1-byte data" (in this example, the time taken from generating of the serial I/O1 receive interrupt request to input of the next synchronous clock is 431 µs).
- "Heading adjustment time < interval between blocks" must be satisfied.

The communication is performed according to the timing shown in Figure 2.4.34. In the slave unit, when a synchronous clock is not input within a certain time (heading adjustment time), the next clock input is processed as the beginning (heading) of a block.
When a clock is input again after one block (8 byte) is received, the clock is ignored.
Figure 2.4.35 shows relevant registers setting.

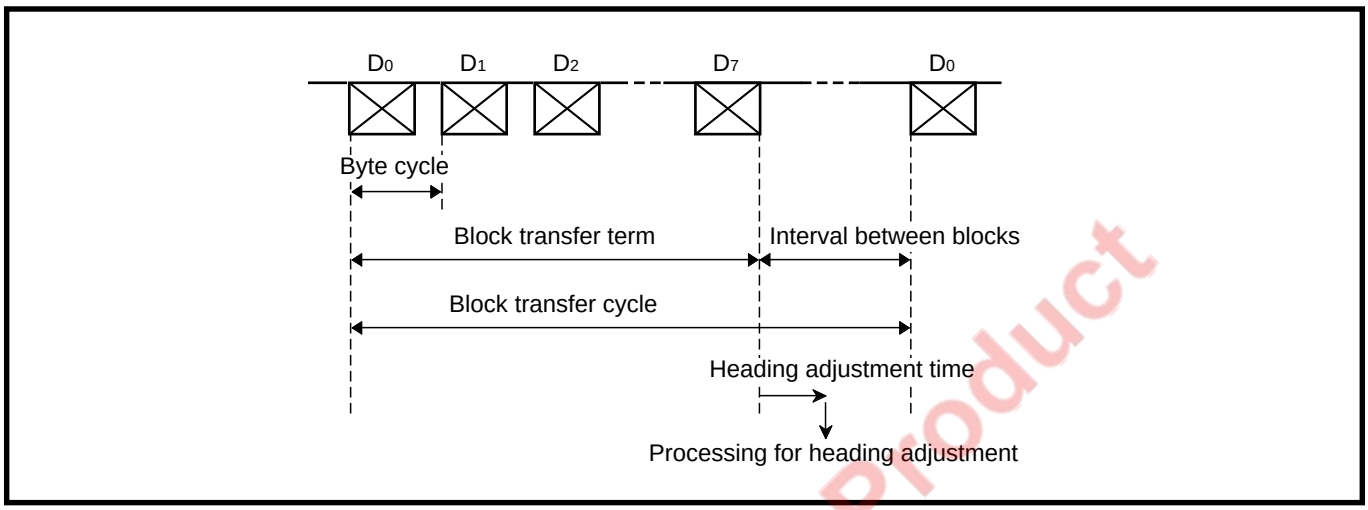


Fig. 2.4.34 Timing chart

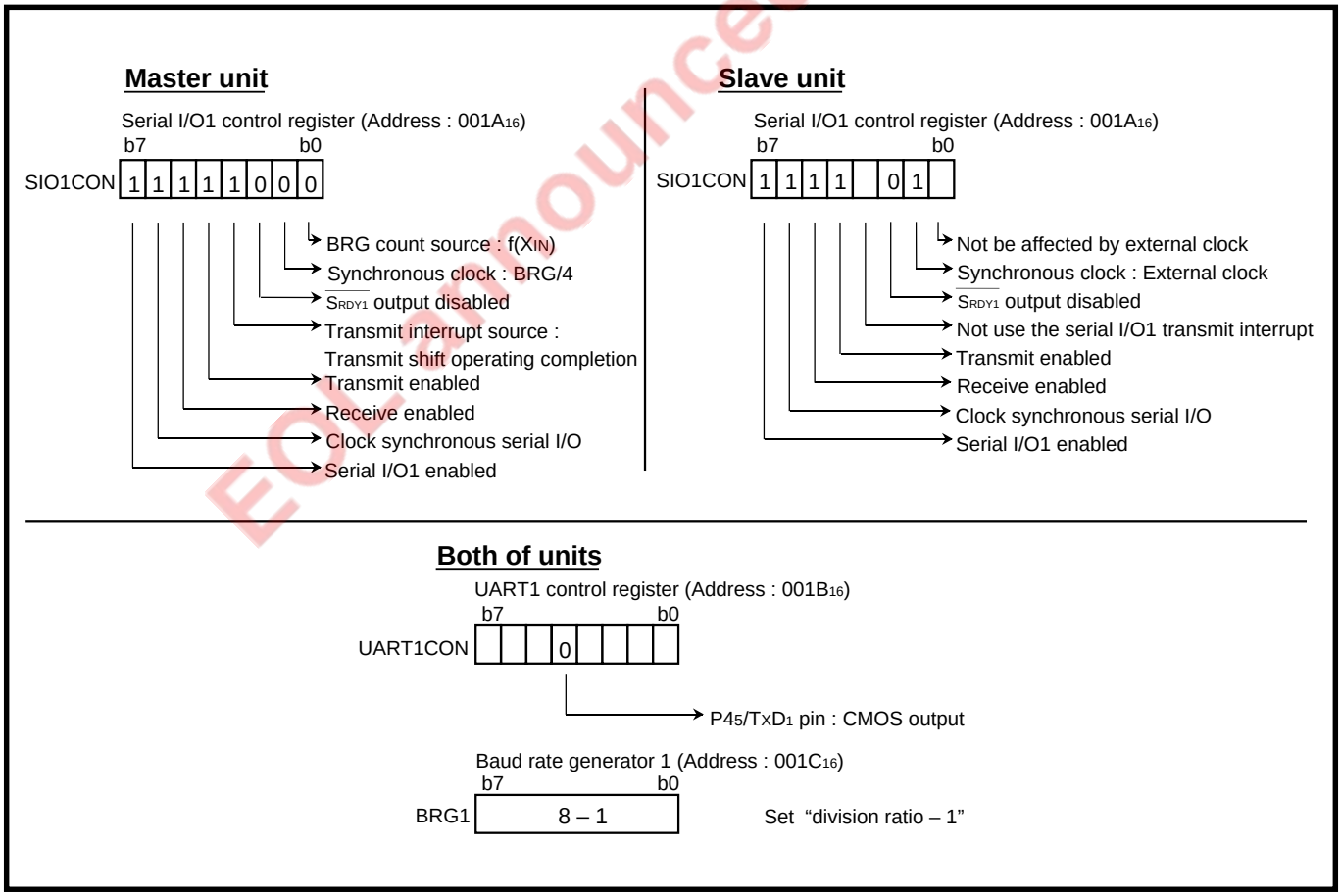


Fig. 2.4.35 Relevant registers setting

Control procedure :

● Control in the master unit

After setting the relevant registers shown in Figure 2.4.35, the master unit starts transmission or reception of 1-byte data by writing transmission data to the transmit buffer register.

To perform the communication in the timing shown in Figure 2.4.34, take the timing into account and write transmission data. Additionally, read out the reception data when the serial I/O1 transmit interrupt request bit is set to "1," or before the next transmission data is written to the transmit buffer register.

Figure 2.4.36 shows a control procedure of the master unit using timer interrupts.

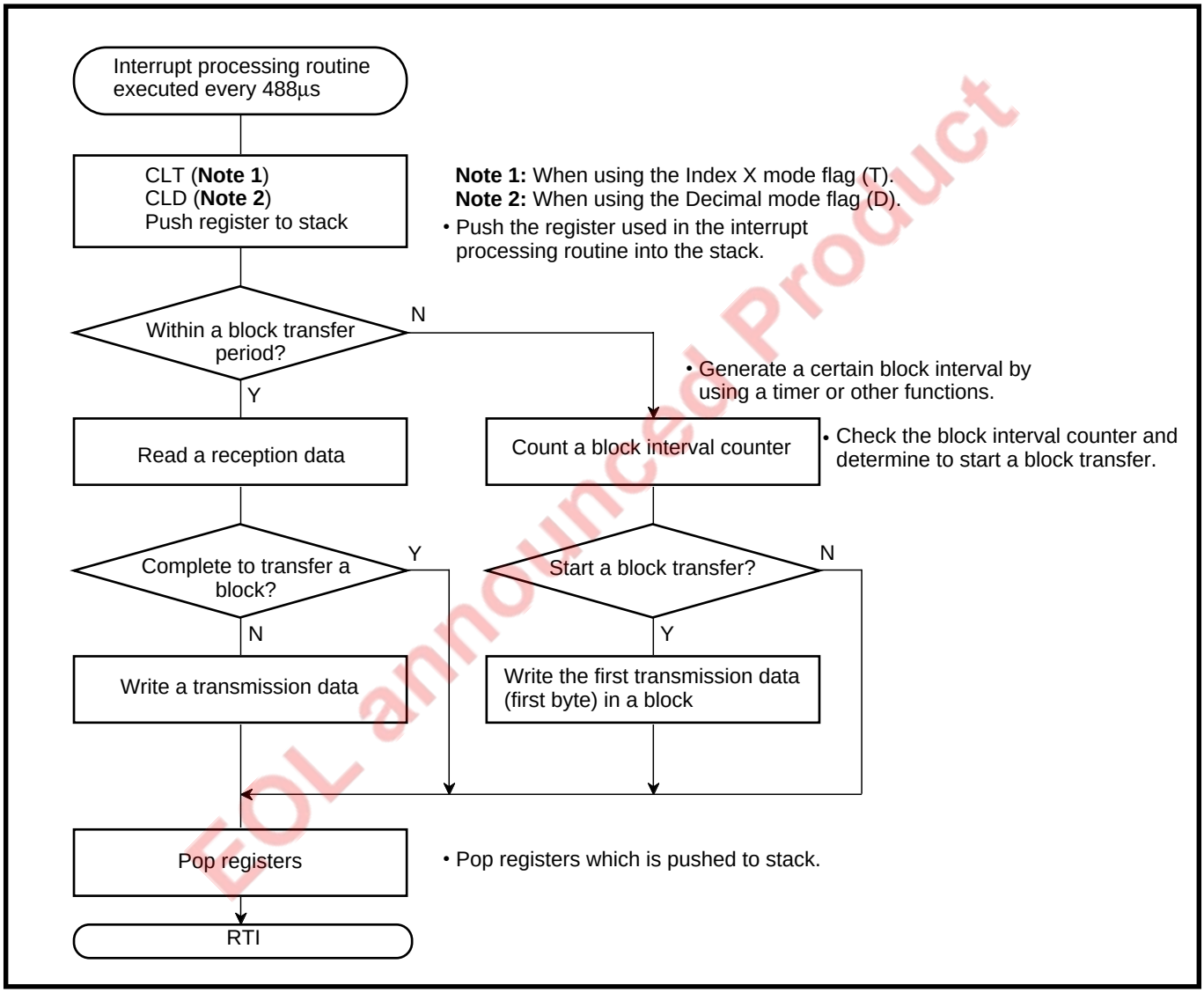


Fig. 2.4.36 Control procedure of master unit

● Control in the slave unit

After setting the relevant registers as shown in Figure 2.4.35, the slave unit becomes the state where a synchronous clock can be received at any time, and the serial I/O1 receive interrupt occurs each time an 8-bit synchronous clock is received.

In the serial I/O1 receive interrupt processing routine, the data to be transmitted next is written to the transmit buffer register after the received data is read out.

However, if no serial I/O1 receive interrupt occurs for a certain time (heading adjustment time or more), the following processing will be performed.

1. The first 1-byte data of the transmission data in the block is written into the transmit buffer register.
2. The data to be received next is processed as the first 1 byte of the received data in the block.

Figure 2.4.37 shows a control procedure of the slave unit using the serial I/O1 receive interrupt and any timer interrupt (for heading adjustment).

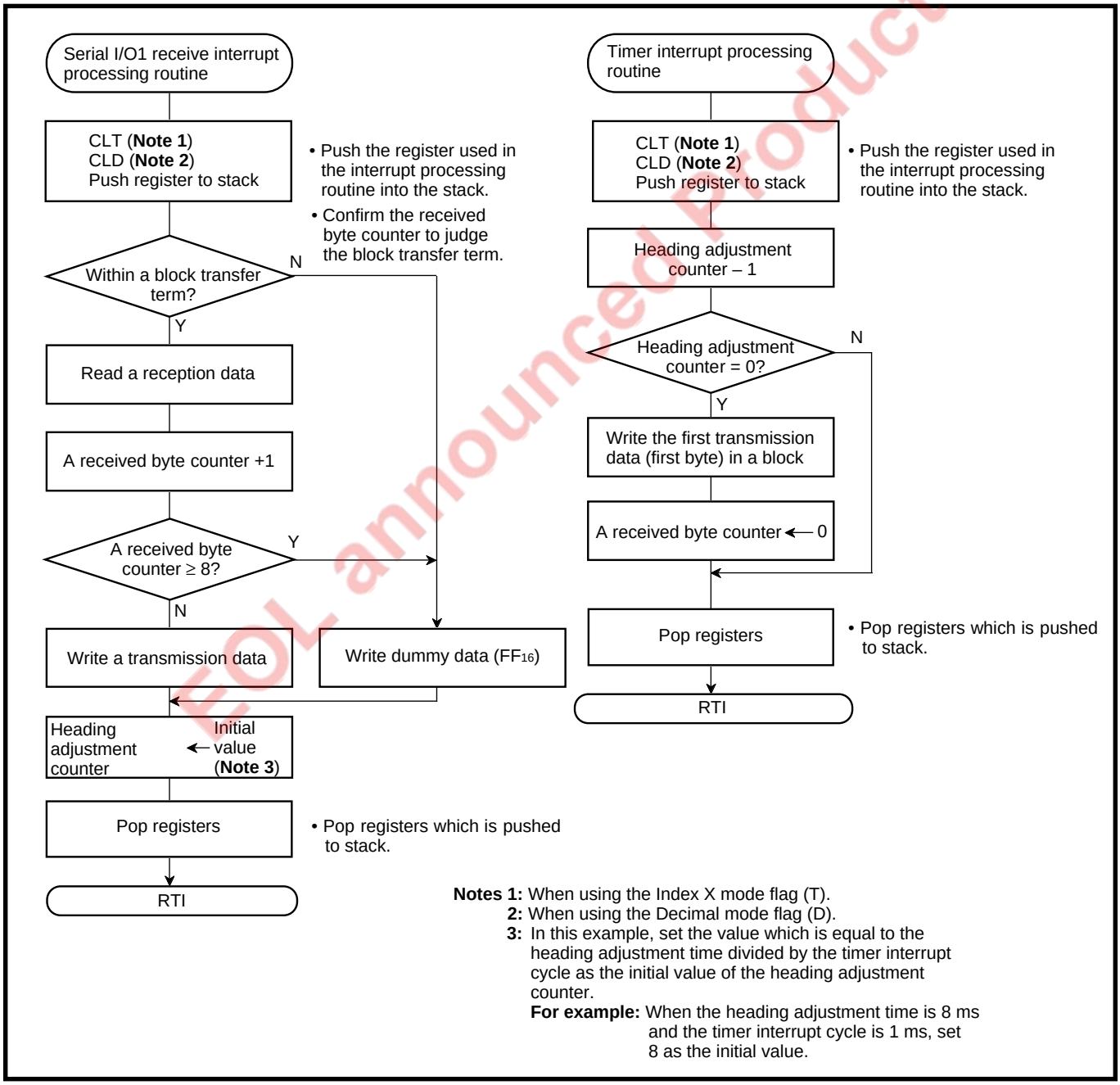


Fig. 2.4.37 Control procedure of slave unit

(4) Communication (transmit/receive) using asynchronous serial I/O (UART)

Outline : 2-byte data is transmitted and received, using the asynchronous serial I/O. Port $P4_0$ is used for communication control.

Figure 2.4.38 shows a connection diagram, and Figure 2.4.39 shows a timing chart.

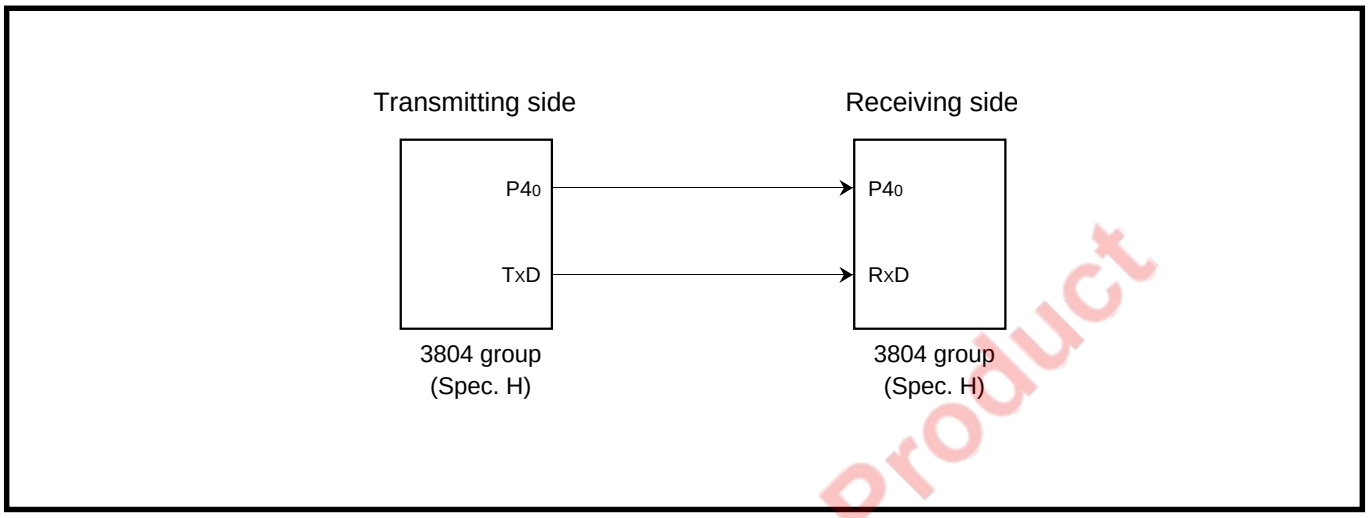


Fig. 2.4.38 Connection diagram (Communication using UART)

Specifications :
- Serial I/O1 is used (UART is selected).
- Transfer bit rate : 9600 bps ($f(X_{IN})$ = 4.9152 MHz is divided by 512)
- Communication control using port $P4_0$
  (The output level of port $P4_0$ is controlled by software.)
- 2-byte data is transferred from the transmitting side to the receiving side at intervals of 10 ms generated by the timer.

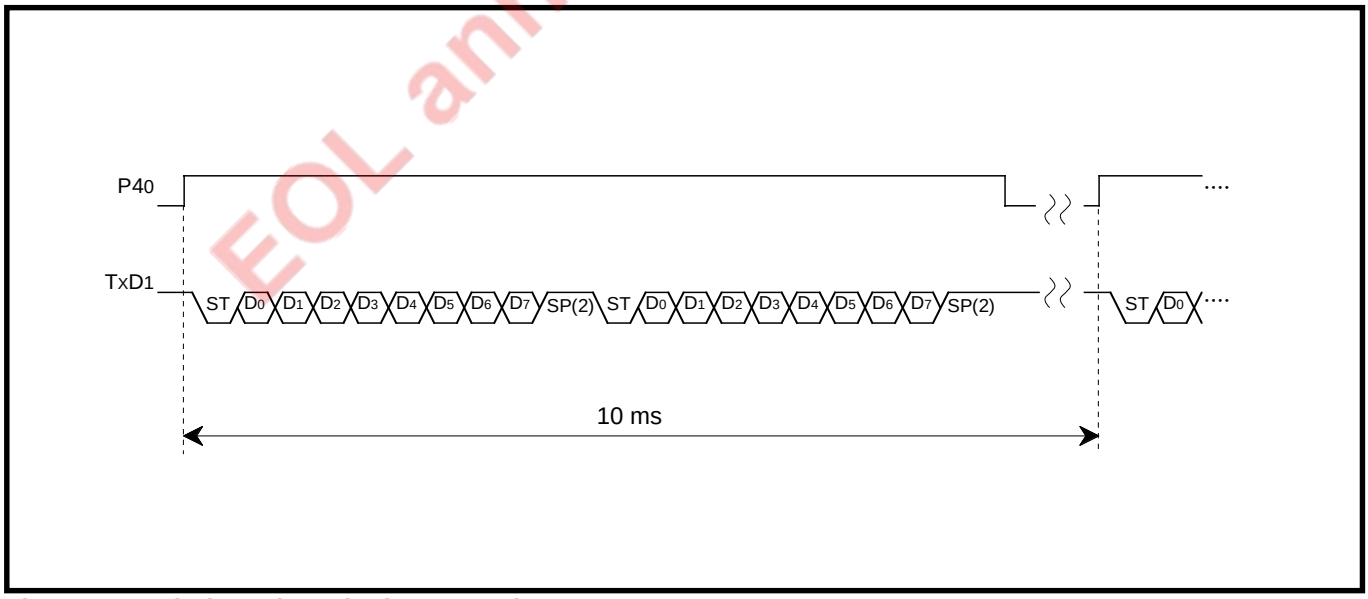


Fig. 2.4.39 Timing chart (using UART)

Table 2.4.4 shows setting examples of the baud rate generator (BRG) values and transfer bit rate values; Figure 2.4.40 shows registers setting relevant to the transmitting side; Figure 2.4.41 shows registers setting relevant to the receiving side.

Table 2.4.4 Setting examples of Baud rate generator (BRG) values and transfer bit rate values

BRG count source **(Note 1)**	BRG setting value	Transfer bit rate (bps) **(Note 2)**	
		at f(XIN) = 4.9152 MHz	at f(XIN) = 16 MHz
f(XIN)/4	255(FF16)	300	976.5625
f(XIN)/4	127(7F16)	600	1953.125
f(XIN)/4	63(3F16)	1200	3906.25
f(XIN)/4	31(1F16)	2400	7812.5
f(XIN)/4	15(0F16)	4800	15625
f(XIN)/4	7(0716)	9600	31250
f(XIN)/4	3(0316)	19200	62500
f(XIN)/4	1(0116)	38400	125000
f(XIN)	3(0316)	76800	250000
f(XIN)	1(0116)	153600	500000
f(XIN)	0(0016)	307200	1000000

Notes 1: Select the BRG count source with bit 0 of the serial I/O1 control register (Address : 001A16).

2: Equation of transfer bit rate:

$$\text{Transfer bit rate (bps)} = \frac{f(X_{IN})}{(\text{BRG setting value} + 1) \times 16 \times m^*}$$

*m: When bit 0 of the serial I/O1 control register (Address : 001A16) is set to “0,” a value of m is 1.
When bit 0 of the serial I/O1 control register (Address : 001A16) is set to “1,” a value of m is 4.

Transmitting side

Serial I/O1 status register (Address : 0019_{16})

SIO1STS: b7 b0

- Transmit buffer empty flag
  - Confirm that the data has been transferred from Transmit buffer register to Transmit shift register.
  - When this flag is "1", it is possible to write the next transmission data in to Transmit buffer register.
- Transmit shift register shift completion flag
  - Confirm completion of transmitting 1-byte data with this flag.
  - "1" : Transmit shift completed

Serial I/O1 control register (Address : $001A_{16}$)

SIO1CON: b7 b0 — 1 0 0 1 [ ] 0 0 1

- BRG count source : f(X_{IN})/4
- Synchronous clock : BRG/16
- $\overline{S_{RDY1}}$ output disabled
- Transmit enabled
- Receive disabled
- Asynchronous serial I/O(UART)
- Serial I/O1 enabled

UART1 control register (Address : $001B_{16}$)

UART1CON: b7 b0 — [ ] [ ] [ ] 0 1 [ ] 0 0

- Character length : 8 bits
- Parity checking disabled
- Stop bit length : 2 stop bits
- $P4_5/TxD_1$ pin : CMOS output

Baud rate generator 1 (Address : $001C_{16}$)

BRG1: b7 b0 — 8 – 1

$$\text{Set } \frac{f(X_{IN})}{\text{Transfer bit rate} \times 16 \times m\ *} - 1$$

✻ When bit 0 of Serial I/O1 control register (Address : $001A_{16}$) is set to "0," a value of m is 1.
When bit 0 of Serial I/O1 control register (Address : $001A_{16}$) is set to "1," a value of m is 4.

Port P4 (Address : 0008_{16})

P4: b7 b0 — [ ] [ ] [ ] [ ] [ ] [ ] [ ] 0

- Port $P4_0$: Communication control: "H" active

Port P4 direction register (Address : 0009_{16})

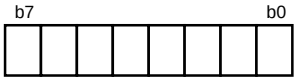
P4D: b7 b0 — [ ] [ ] [ ] [ ] [ ] [ ] [ ] 1

- Port $P4_0$: Communication control: Output mode

Fig. 2.4.40 Registers setting relevant to transmitting side

Receiving side

Serial I/O1 status register (Address : 0019$_{16}$)



→ Receive buffer full flag
Confirm completion of receiving 1-byte data with this flag.
"1" : at completing reception
"0" : at reading out contents of Receive buffer register

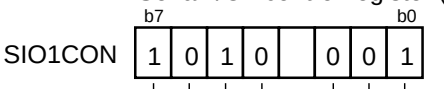
→ Overrun error flag
"1" : When data is ready in Receive shift register while Receive buffer register contains the data.

→ Parity error flag
"1" : When a parity error occurs in enabled parity.

→ Framing error flag
"1" : When stop bits cannot be detected at the specified timing

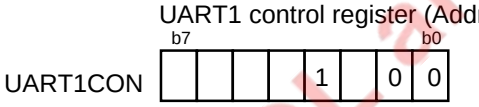
→ Summing error flag
"1" : when any one of the following errors occurs.
- Overrun error
- Parity error
- Framing error

Serial I/O1 control register (Address : 001A$_{16}$)



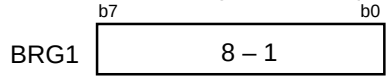
→ BRG count source : f(X$_{IN}$)/4
→ Synchronous clock : BRG/16
→ $\overline{S_{RDY1}}$ output disabled
→ Transmit disabled
→ Receive enabled
→ Asynchronous serial I/O(UART)
→ Serial I/O1 enabled

UART1 control register (Address : 001B$_{16}$)



→ Character length : 8 bits
→ Parity checking disabled
→ Stop bit length : 2 stop bits

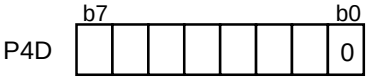
Baud rate generator 1 (Address : 001C$_{16}$)



Set $\dfrac{f(X_{IN})}{\text{Transfer bit rate} \times 16 \times m \ *} - 1$

✻ When bit 0 of Serial I/O1 control register (Address : 001A$_{16}$) is set to "0," a value of m is 1.
When bit 0 of Serial I/O1 control register (Address : 001A$_{16}$) is set to "1," a value of m is 4.

Port P4 direction register (Address : 0009$_{16}$)



→ Port P4$_0$: Communication control: Input mode

Fig. 2.4.41 Registers setting relevant to receiving side

Figure 2.4.42 shows a control procedure of the transmitting side, and Figure 2.4.43 shows a control procedure of the receiving side.

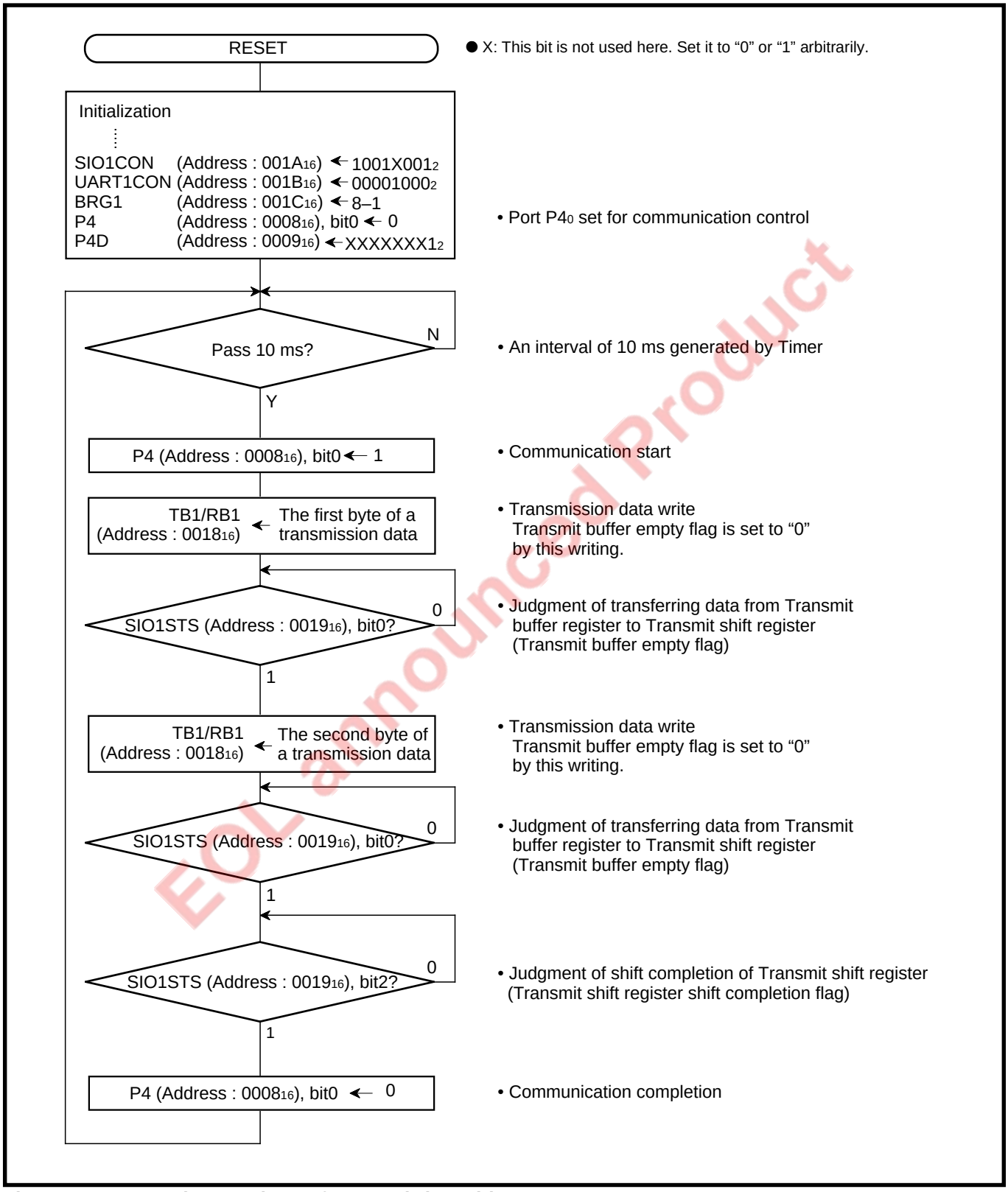


Fig. 2.4.42 Control procedure of transmitting side

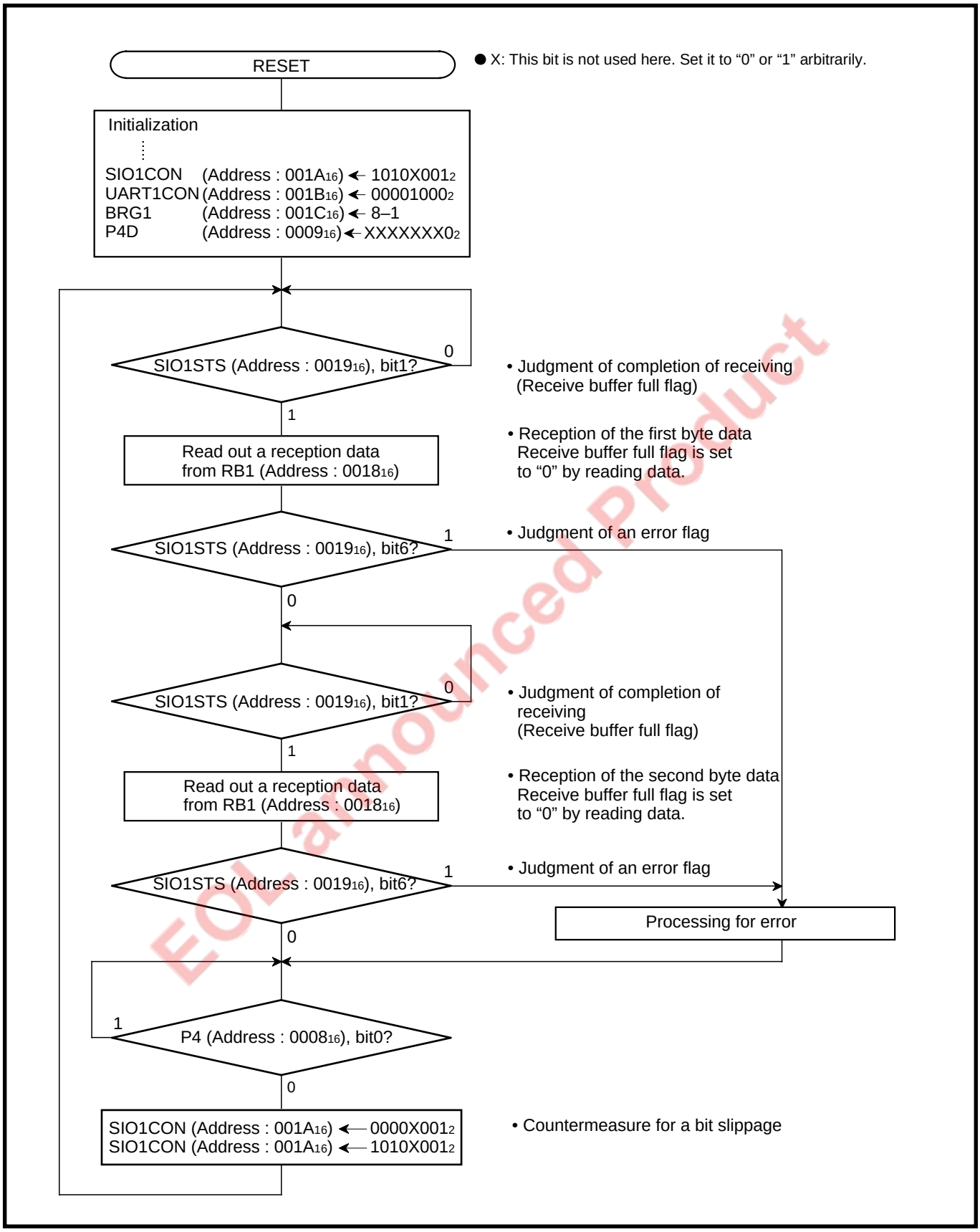


Fig. 2.4.43 Control procedure of receiving side

2.4.8 Notes on serial interface

(1) Notes when selecting clock synchronous serial I/O

① **Stop of transmission operation**
As for serial I/Oi (i = 1, 3) that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the serial I/Oi enable bit and the transmit enable bit to “0” (serial I/Oi and transmit disabled).

● **Reason**
Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/Oi enable bit is cleared to “0” (serial I/Oi disabled), the internal transmission is running (in this case, since pins TxDi, RxDi, SCLKi, and $\overline{\text{SRDY}i}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/Oi enable bit is set to “1” at this time, the data during internally shifting is output to the TxDi pin and an operation failure occurs.

② **Stop of receive operation**
As for serial I/Oi (i = 1, 3) that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the receive enable bit to “0” (receive disabled), or clear the serial I/Oi enable bit to “0” (serial I/Oi disabled).

③ **Stop of transmit/receive operation**
As for serial I/Oi (i = 1, 3) that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear both the transmit enable bit and receive enable bit to “0” simultaneously (transmit and receive disabled) in the clock synchronous serial I/O mode.
(When data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

● **Reason**
In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.
In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to “0” (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/Oi enable bit to “0” (serial I/Oi disabled) (refer to ① in (1)).

(2) Notes when selecting clock asynchronous serial I/O

① **Stop of transmission operation**
Clear the transmit enable bit to “0” (transmit disabled). Transmission operation does not stop by setting the serial I/Oi enable bit (i = 1, 3) to “0”.

● **Reason**
This is the same as ① in (1).

② **Stop of receive operation**
Clear the receive enable bit to “0” (receive disabled).

③ **Stop of transmit/receive operation**

Only transmission operation is stopped.

Clear the transmit enable bit to “0” (transmit disabled). Transmission operation does not stop by setting the serial I/Oi enable bit (i = 1, 3) to “0”.

● **Reason**

This is the same as ① in (1).

Only receive operation is stopped.

Clear the receive enable bit to “0” (receive disabled).

(3) $\overline{\text{SRDY}}$i (i = 1, 3) output of reception side

When signals are output from the $\overline{\text{SRDY}}$i pin on the reception side by using an external clock in the clock synchronous serial I/O mode, set all of the receive enable bit, the $\overline{\text{SRDY}}$i output enable bit, and the transmit enable bit to “1” (transmit enabled).

(4) Setting serial I/Oi (i = 1, 3) control register again

Set the serial I/Oi control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to “0.”

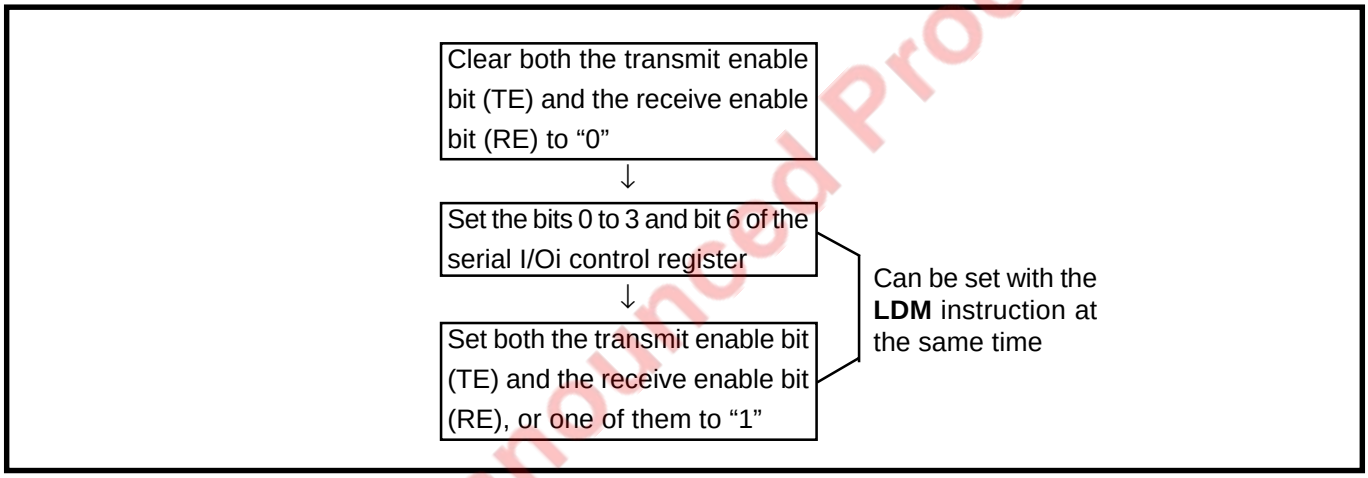


Fig. 2.4.44 Sequence of setting serial I/Oi (i = 1, 3) control register again

(5) Data transmission control with referring to transmit shift register completion flag

After the transmit data is written to the transmit buffer register, the transmit shift register completion flag changes from “1” to “0” with a delay of 0.5 to 1.5 shift clocks. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

(6) Transmission control when external clock is selected

When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to “1” at “H” of the SCLKi (i = 1, 3) input level. Also, write the transmit data to the transmit buffer register at “H” of the SCLKi input level.

(7) Transmit interrupt request when transmit enable bit is set

When the transmit interrupt is used, take the following sequence.

① Set the serial I/Oi transmit interrupt enable bit (i = 1, 3) to “0” (disabled).
② Set the tranasmit enable bit to “1”.
③ Set the serial I/Oi transmit interrupt request bit (i = 1, 3) to “0” after 1 or more instruction has executed.
④ Set the serial I/Oi transmit interrupt enable bit (i = 1, 3) to “1” (enabled).

● **Reason**

When the transmission enable bit is set to “1”, the transmit buffer empty flag and transmit shift register shift completion flag are also set to “1”.
Therefore, regardless of selecting which timing for the generating of transmit interrupts, the interrupt request is generated and the transmit interrupt request bit is set at this point.

(8) Writing to baud rate generator i (BRGi) (i = 1, 3)

Write data to the baud rate generator i (BRGi) (i = 1, 3) while the transmission/reception operation is stopped.

2.5 Multi-master I²C-BUS interface

The only 3804 group has functions of the multi-master I²C-BUS interface.
The multi-master I²C-BUS interface is a serial communication circuit, conforming to the Philips I²C-BUS data transfer format. This paragraph explains the I²C-BUS overview and communication examples.

2.5.1 Memory map

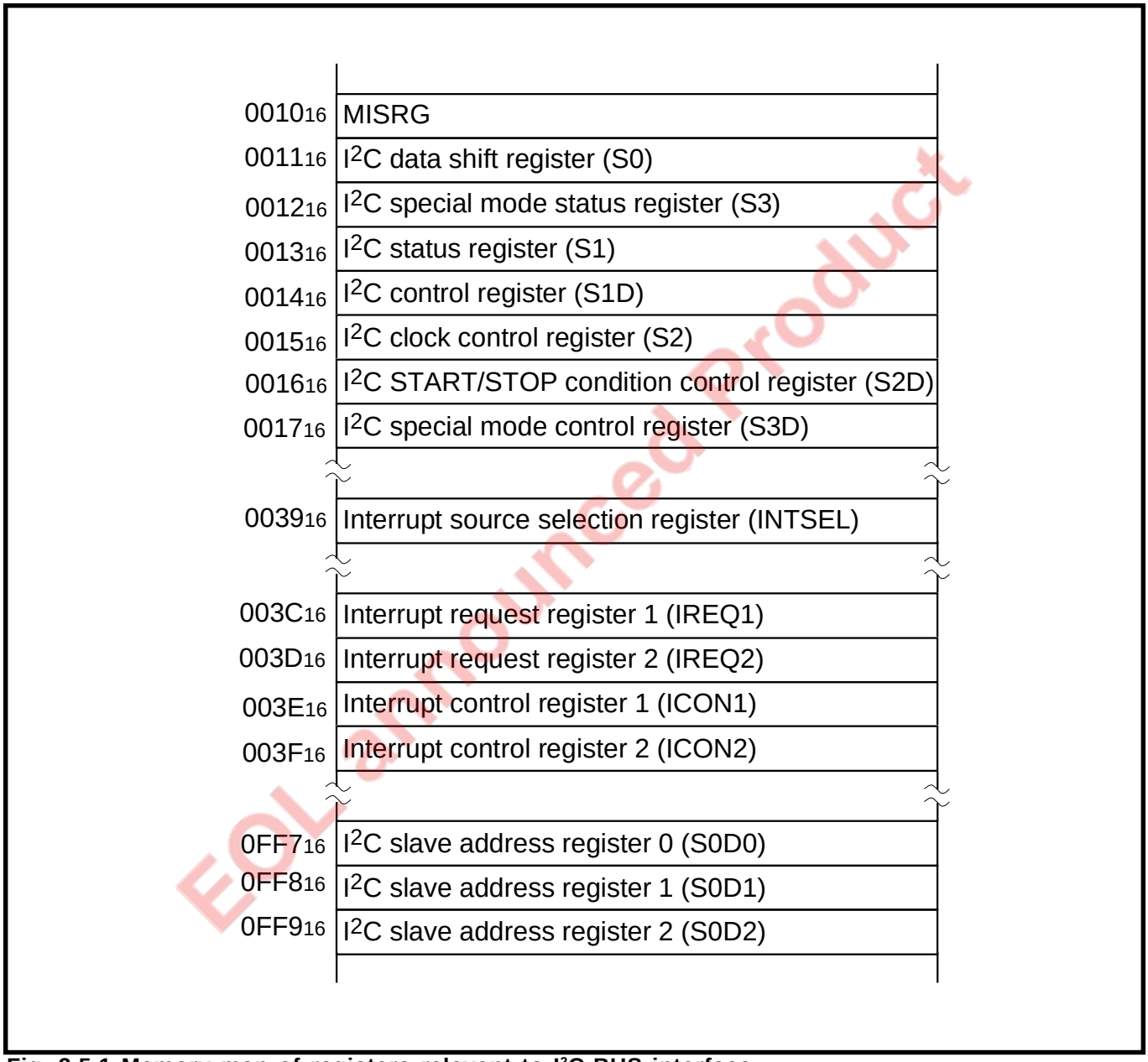


Fig. 2.5.1 Memory map of registers relevant to I²C-BUS interface

2.5.2 Relevant registers

MISRG

b7 b6 b5 b4 b3 b2 b1 b0

MISRG
(MISRG: address 0010₁₆)

b	Name	Functions	At reset	R	W
0	Oscillation stabilizing time set after STP instruction released bit	0: Automatically set **(Note 1)** 1: Autimatically set disabled	0	○	○
1	Middle-speed mode automatic switch set bit	0: Not set automatically 1: Automatic switching enabled **(Notes 2, 3)**	0	○	○
2	Middle-speed mode automatic switch wait time set bit	0: 4.5 to 5.5 machine cycles 1: 6.5 to 7.5 machine cycles	0	○	○
3	Middle-speed mode automatic switch start bit (Depending on program)	0: Invalid 1: Automatic switch start **(Note 3)**	0	○	○
4	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are 0 .		0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Notes 1: 01₁₆ is set to Timer 1, FF₁₆ is set to Prescaler 12.
2: During operation in low-speed mode, it is possible automatically to switch to middle-speed mode owing to the rising of SCL/SDA.
3: When automatic switch to middle-speed mode from low-speed mode occurs, the values of CPU mode register (003B₁₆) change.

Fig. 2.5.2 Structure of MISRG

I²C data shift register

b7 b6 b5 b4 b3 b2 b1 b0

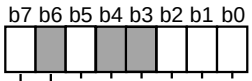
I²C data shift register
(S0: address 0011₁₆)

b	Functions	At reset	R	W
0	• 8-bit shift register to store receive data and write transmit data.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Note: When data is written to I²C data shift register after the MST bit is set to "0" (slave mode), keep the interval for 8 machine cycles or more.
Also, when the read-modify-write instructions (SEB, CLB) are used during data transfer, the values may be undefined.

Fig. 2.5.3 Structure of I²C data shift register

I²C special mode status register



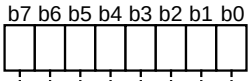
I²C special mode status register
(S3: address 0012₁₆)

b	Name	Functions	At reset	R	W
0	Slave address 0 comparison flag (AAS0)	0: Address disagreement 1: Address agreement **(Notes 1, 2)**	0	○	×
1	Slave address 1 comparison flag (AAS1)	0: Address disagreement 1: Address agreement **(Notes 1, 2)**	0	○	×
2	Slave address 2 comparison flag (AAS2)	0: Address disagreement 1: Address agreement **(Notes 1, 2)**	0	○	×
3	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are 0 .		0	○	×
4	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are undefined.		0	○	×
5	SCL pin low hold 2 flag (PIN2)	0: SCL pin low hold 1: SCL pin low release **(Notes 1, 3)**	1	○	×
6	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are 0 .		0	○	×
7	STOP condition flag (SPCF)	0: No detection 1: Detection **(Notes 1, 4)**	0	○	×

Notes 1: These bits and flags can be read out, but cannot be written.
2: These bits can be detected only when the data format selection bit (ALS) of I²C control register is set to 0 .
3: This bit is initialized to 1 at reset, when the ACK interrupt control bit is 0 , or when writing 1 to the SCL pin low hold 2 flag set bit.
4: This bit is initialized to 0 at reset, when the I²C-BUS interface enable bit (ES0) is 0 , or when writing 1 to the STOP condition flag clear bit.

Fig. 2.5.4 Structure of I²C special mode status register

I²C status register



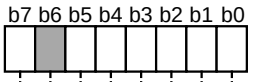
I²C status register
(S1: address 0013₁₆)

b	Name	Functions	At reset	R	W
0	Last receive bit (LRB)	0: Last bit = 0 1: Last bit = 1 (**Note 1**)	Undefined	○	×
1	General call detection flag (AD0)	0: No general call detected 1: General call detected (**Notes 1, 2**)	0	○	×
2	Slave address comparison flag (AAS)	0: Address disagreement 1: Address agreement (**Notes 1, 2**)	0	○	×
3	Arbitration lost detection flag (AL)	0: Not detected 1: Detected (**Note 1**)	0	○	×
4	SCL pin low hold bit (PIN)	0: SCL pin low hold (**Note 3**) 1: SCL pin low release	1	○	○
5	Bus busy flag (BB)	0: Bus free 1: Bus busy	0	○	○
6	Communication mode specification bits (TRX, MST)	b7 b6 0 0: Slave receive mode 0 1: Slave transmit mode 1 0: Master receive mode 1 1: Master transmit mode	0	○	○
7			0	○	○

Notes 1: These flags and bits are exclusive to input. When writing to these bits, write 0 to these bits.
2: These bits can be detected only when the data format selection bit (ALS) of I²C control register is set to 0 .
3: This bit can be set to 1 by program, but cannot be cleared to 0 .
4: All bits are changed by hardware. Do not use the read-modify-write instructions (SEB, CLB).

Fig. 2.5.5 Structure of I²C status register

I²C control register



I²C control register
(S1D: address 0014_{16})

b	Name	Functions	At reset	R	W
0	Bit counter (Number of transmit/receive bits) (BC0, BC1, BC2)	b2b1b0 0 0 0: 8 0 0 1: 7 0 1 0: 6 0 1 1: 5 1 0 0: 4 1 0 1: 3 1 1 0: 2 1 1 1: 1	0	○	○
1			0	○	○
2			0	○	○
3	I²C-BUS interface enable bit (ES0)	0: Disabled 1: Enabled	0	○	○
4	Data format selection bit (ALS)	0: Addressing format 1: Free data format	0	○	○
5	Addressing format selection bit (10BIT SAD)	0: 7-bit addressing format 1: 10-bit addressing format	0	○	○
6	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are 0 .		0	○	×
7	I²C-BUS interface pin input level selection bit (TISS)	0: SMBUS input 1: CMOS input	0	○	○

Note: Do not use the read-modify-write instruction because some bits change by hardware when the start condition is detected and the byte-transfer is completed.

Fig. 2.5.6 Structure of I²C control register

I²C clock control register

b7 b6 b5 b4 b3 b2 b1 b0

I²C clock control register
(S2: address 0015₁₆)

b	Name	Functions			At reset	R	W
0	SCL frequency control bits (CCR0, CCR1, CCR2, CCR3, CCR4)	Setting value b4b3b2b1b0	Standard clock mode	High-speed clock mode	0	○	○
1		00 to 02	Disabled	Disabled	0	○	○
2		03	Disabled	333			
		04	(**Note 2**)	250			
		05	100	400 (**Note 3**)	0	○	○
		06	83.3	166			
3		⋮	500/CCR value (**Note 3**)	1000/CCR value (**Note 3**)	0	○	○
		1D	17.2	34.5			
		1E	16.6	33.3			
4		1F	16.1	32.3	0	○	○
		(ϕ = 4 MHz, Unit: kHz) (**Note 1**)					
5	SCL mode specification bit (FAST MODE)	0: Standard clock mode 1: High-speed clock mode			0	○	○
6	ACK bit (ACK BIT)	0: ACK is returned. 1: ACK is not returned.			0	○	○
7	ACK clock bit (ACK)	0: No ACK clock 1: ACK clock			0	○	○

Notes 1: Duty of SCL clock output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at ϕ = 4 MHz). H duration of the clock fluctuates from —4 to +2 machine cycles in the standard clock mode, and fluctuates from —2 to +2 machine cycles in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because L duration is extended instead of H duration reduction.
These are values when SCL clock synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

2: Each value of SCL frequency exceeds the limit at ϕ = 4 MHz or more. When using these setting value, use ϕ of 4 MHz or less.

3: The data formula of SCL frequency is described below:
ϕ/(8 × CCR value) Standard clock mode
ϕ/(4 × CCR value) High-speed clock mode (CCR value ≠ 5)
ϕ/(2 × CCR value) High-speed clock mode (CCR value = 5)
Do not set 0 to 2 as CCR value regardless of ϕ frequency.
Set 100 kHz (max.) in the standard clock mode and 400 kHz (max.) in the high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

Fig. 2.5.7 Structure of I²C clock control register

I²C START/STOP condition control register

b7 b6 b5 b4 b3 b2 b1 b0
0

I²C START/STOP condition control register
(S2D: address 0016₁₆)

b	Name	Functions	At reset	R	W
0	START/STOP condition set bits (SSC0, SSC1, SSC2, SSC3, SSC4)	SCL release time = ϕ (µs) × (SSC+1) Setup time = ϕ (µs) × (SSC+1)/2 Hold time = ϕ (µs) × (SSC+1)/2	0	○	○
1			1	○	○
2			0	○	○
3			1	○	○
4			1	○	○
5	SCL/SDA interrupt pin polarity selection bit (SIP)	0: Falling edge active 1: Rising edge active	0	○	○
6	SCL/SDA interrupt pin selection bit (SIS)	0: SDA valid 1: SCL valid	0	○	○
7	Fix this bit to 0 .		0	○	○

Note: Fix SSC0 to 0 . Also, do not set SSC4 to SSC0 to odd values or 00000₂ .

Fig. 2.5.8 Structure of I²C START/STOP condition control register

I²C special mode control register

b7 b6 b5 b4 b3 b2 b1 b0
0 0

I²C special mode control register
(S3D: address 0017₁₆)

b	Name	Functions	At reset	R	W
0	Fix this bit to 0 .		0	○	○
1	ACK interrupt control bit (ACKICON)	0: At communication completion 1: At falling of ACK clock and communication completion	0	○	○
2	Slave address control bit (MSLAD)	0: One-byte slave address compare mode 1: Three-byte slave address compare mode	0	○	○
3	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are 0 .		0	○	×
4	Fix this bit to 0 .		0	○	○
5	SCL pin low hold 2 flag set bit (PIN2IN)	Writing 1 to this bit initializes the SCL pin low hold 2 flag to 1 . **(Notes 1, 2)**	0	○	○
6	SCL pin low hold set bit (PIN2HD)	Writing 1 to this bit clears the SCL pin low hold 2 flag to 0 and holds the SCL pin low. **(Notes 1, 2)**	0	○	○
7	STOP condition flag clear bit (SPFCL)	Writing 1 to this bit initializes the STOP condition flag to 0 . **(Note 1)**	0	○	○

Notes 1: When 0 is written to these bits, nothing is happened.
2: Do not write 1 to these bits at the same time.

Fig. 2.5.9 Structure of I²C special mode control register

I²C slave address register i (i = 0 to 2)

b7 b6 b5 b4 b3 b2 b1 b0

I²C slave address register i (i = 0 to 2)
(S0D0, S0D1, S0D2: addresses 0FF7₁₆, 0FF8₁₆, 0FF9₁₆)

b	Name	Functions	At reset	R	W
0	Read/$\overline{\text{Write}}$ bit (RWB)	0: Write bit 1: Read bit	0	○	○
1	Slave address (SAD0, SAD1, SAD2, SAD3, SAD4, SAD5, SAD6)	The contents of these bits are compared with the address data transmitted from master.	0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Note: When the read-modify-write instructions (SEB, CLB) are used at detection of stop condition, the values may be undefined.

Fig. 2.5.10 Structure of I²C slave address register i (i = 0 to 2)

Interrupt source selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt source selection register
(INTSEL: address 0039₁₆)

b	Name	Functions	At reset	R	W
0	INT_0/Timer Z interrupt source selection bit (*1)	0: INT_0 interrupt 1: Timer Z interrupt	0	○	○
1	Serial I/O2/Timer Z interrupt source selection bit (*1)	0: Serial I/O2 interrupt 1: Timer Z interrupt	0	○	○
2	Serial I/O1 transmit/ SCL, SDA interrupt source selection bit (*2)	0: Serial I/O1 transmit interrupt 1: SCL, SDA interrupt	0	○	○
3	$CNTR_0$/SCL, SDA interrupt source selection bit (*2)	0: $CNTR_0$ interrupt 1: SCL, SDA interrupt	0	○	○
4	INT_4/$CNTR_2$ interrupt source selection bit	0: INT_4 interrupt 1: $CNTR_2$ interrupt	0	○	○
5	INT_2/I²C interrupt source selection bit	0: INT_2 interrupt 1: I²C interrupt	0	○	○
6	$CNTR_1$/Serial I/O3 receive interrupt source selection bit	0: $CNTR_1$ interrupt 1: Serial I/O3 receive interrupt	0	○	○
7	AD converter/Serial I/O3 transmit interrupt source selection bit	0: A/D converter interrupt 1: Serial I/O3 transmit interrupt	0	○	○

*1: Do not write 1 to these bits simultaneously.
*2: Do not write 1 to these bits simultaneously.

Fig. 2.5.11 Structure of Interrupt source selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1
(IREQ1 : address 003C16)

b	Name	Functions	At reset	R	W
0	INT0/Timer Z interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	INT1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O1 transmit/SCL, SDA interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻: 0 can be set by software, but 1 cannot be set.

Fig. 2.5.12 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2
(IREQ2 : address 003D16)

b	Name	Functions	At reset	R	W
0	CNTR0/SCL, SDA interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	CNTR1/Serial I/O3 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O2/Timer Z interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT2/I²C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	INT4/CNTR2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	AD converter/Serial I/O3 transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are 0 .		0	○	×

✻: 0 can be set by software, but 1 cannot be set.

Fig. 2.5.13 Structure of Interrupt request register 2

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1
(ICON1 : address 003E16)

b	Name	Functions	At reset	R	W
0	INT0/Timer Z interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit/SCL, SDA interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.5.14 Structure of Interrupt control register 1

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0
0

Interrupt control register 2
(ICON2 : address 003F16)

b	Name	Functions	At reset	R	W
0	CNTR0/SCL, SDA interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	CNTR1/ Serial I/O3 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/ Timer Z interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT2/I²C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT3 interrupt enable bit	0 : Interrupt disabled	0	○	○
5	INT4/CNTR2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/Serial I/O3 transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to 0 .		0	○	○

Fig. 2.5.15 Structure of Interrupt control register 2

2.5.3 I²C-BUS overview

The I²C-BUS is a both directions serial bus connected with two signal lines; the SCL which transmits a clock and the SDA which transmits data.

Each port of the 3804 group has an N-channel open-drain structure for output and a CMOS structure for input. The devices connected with the I²C-BUS interface use an open drain, so that external pull-up resistors are required. Accordingly, while any one of devices always outputs “L”, other devices cannot output “H”.

Figure 2.5.16 shows the I²C-BUS connection structure.

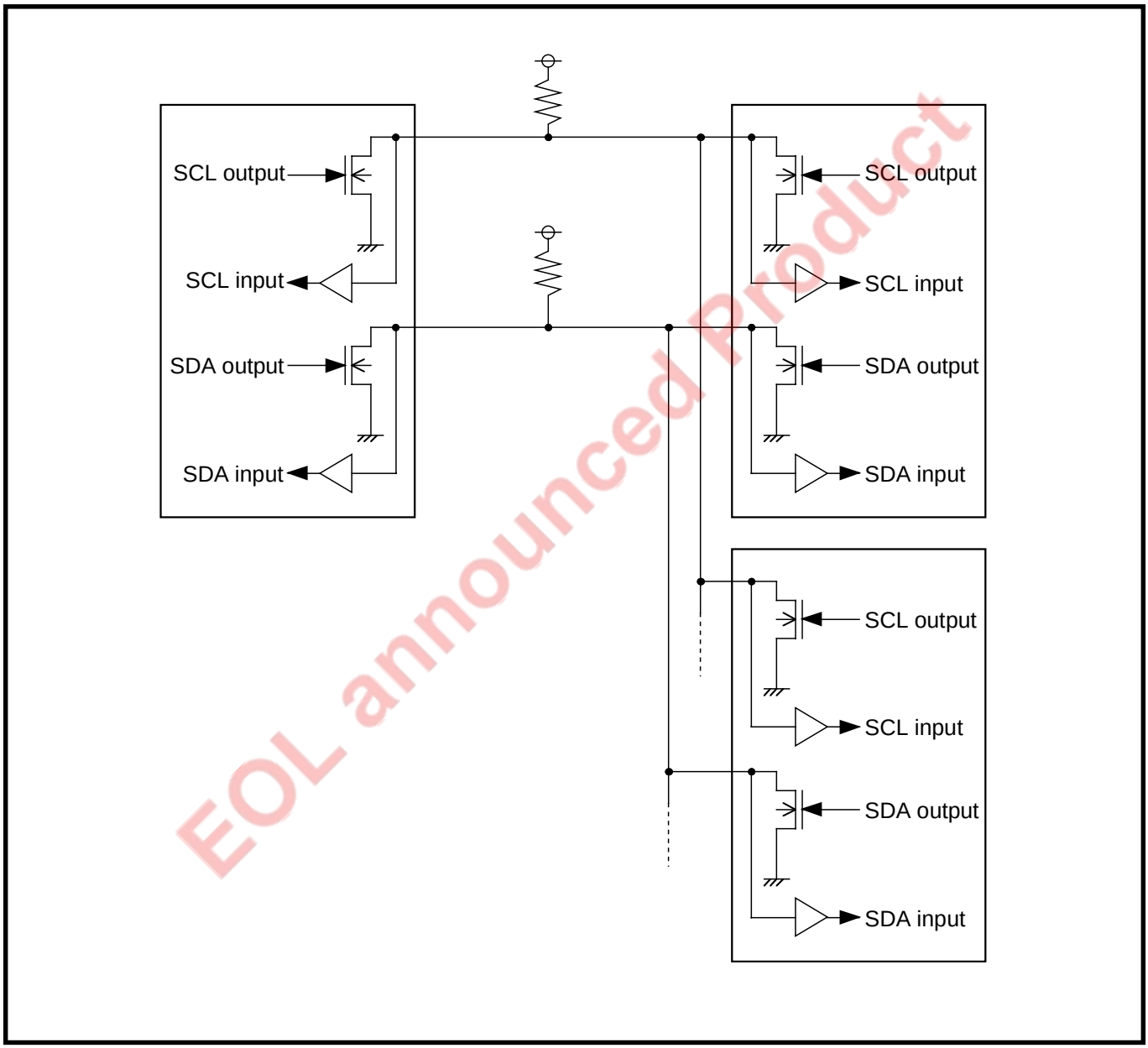


Fig. 2.5.16 I²C-BUS connection structure

2.5.4 Communication format

Figure 2.5.17 shows an I²C-BUS communication format example.
The I²C-BUS consists of the following:
•START condition to indicate communication start
•Slave address and data to specify each device
•ACK to indicate acknowledgment of address and data
•STOP condition to indicate communication completion.

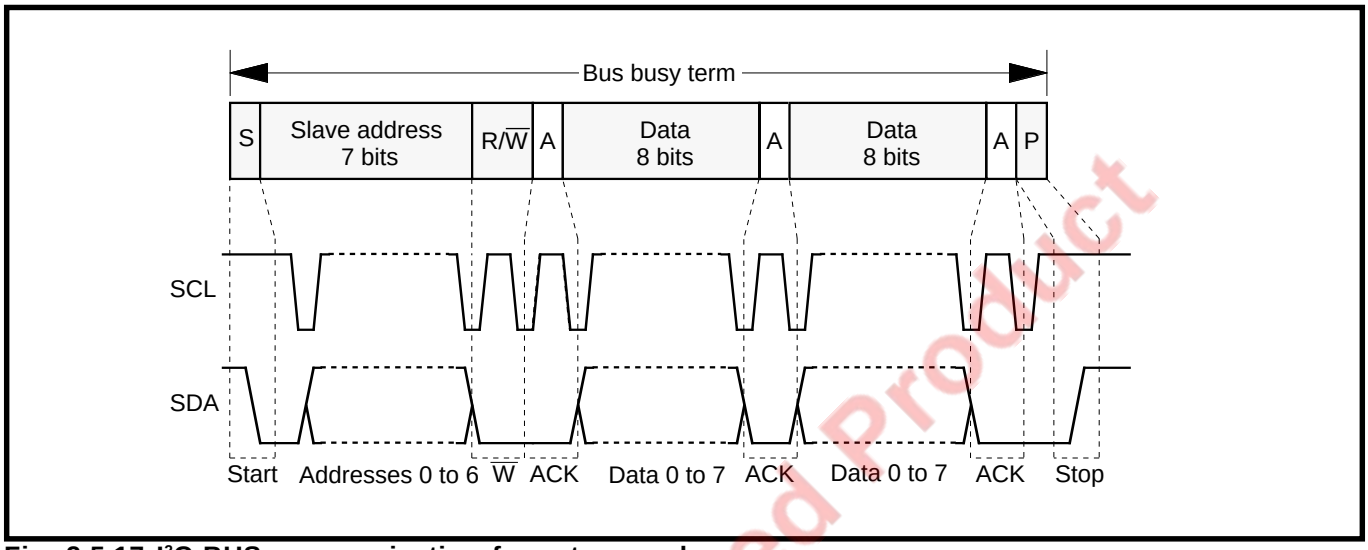


Fig. 2.5.17 I²C-BUS communication format example

(1) START condition

When communication starts, the master device outputs the START condition to the slave device. The I²C-BUS defines that data can be changed when a clock line is “L”. Accordingly, data change when a clock line is “H” is treated as STOP or START condition.
The data line change from “H” to “L” when a clock line is “H” is START condition.

(2) STOP condition

Just as in START condition, the data line change from “L” to “H” when a clock line is “H” is STOP condition.
The term from START condition to STOP condition is called “Bus busy”. The master device is inhibited from starting data transfer during that term.
The Bus busy status can be judged by using the BB flag of I²C status register (bit 5 of address 0013_{16}).

(3) Slave address

The slave address is transmitted after START condition. This address consists of 7 bits and the 7-th bit functions as the read/write (R/$\overline{W}$) bit which indicates a data transmission method. The slave devices connected with the same I²C-BUS must have their addresses, individually. It is because that address is defined for the master to specify the transmitted/received slave device.
The read/write (R/$\overline{W}$) bit indicates a data transmission direction; “L” means write from the master to the slave, and “H” means read in.

(4) Data

The data has an 8-bit length. There are two cases depending on the read/write (R/$\overline{W}$) bit of a slave address; one is from the master to the slave and the other is from the slave to the master.

(5) ACK bit

The ACK bit clock is generated by the master. This is used for indication of acknowledgment on the SDA line, the slave's busy and the data end.

For example, the slave device makes the SDA line "L" for acknowledgment when confirming the slave address following the START condition. The built-in I²C-BUS interface has the slave address automatic judgment function and the ACK acknowledgment function. "L" is automatically output when the ACK bit of I²C clock control register (bit 6 of address 0015_{16}) is "0" and an address data is received. When the slave address and the address data do not correspond, "H" (NACK) is automatically output.

In case the slave device cannot receive owing to an interrupt process, performing operation or others, the master can output STOP condition and complete data transfer by making the ACK data of the slave address "H" for acknowledgment. Even in case the slave device cannot receive data during data transferring, the communication can be interrupted by performing NACK acknowledgment to the following data.

When the master is receiving the data from the slave, the master can notify the slave of completion of data reception by performing NACK acknowledgment to the last data received from the slave.

(6) RESTART condition

The master can receive or transmit data without transmission of STOP condition while the master is transmitting or receiving a data.

For example, after the master transmitted a data to the slave, transmitting a slave address + R (Read) following RESTART condition can make the following data treat as a reception data.

Additionally, transmitting a slave address + $\overline{W}$ (Write) following RESTART condition can make the following data treat as a transmission data.

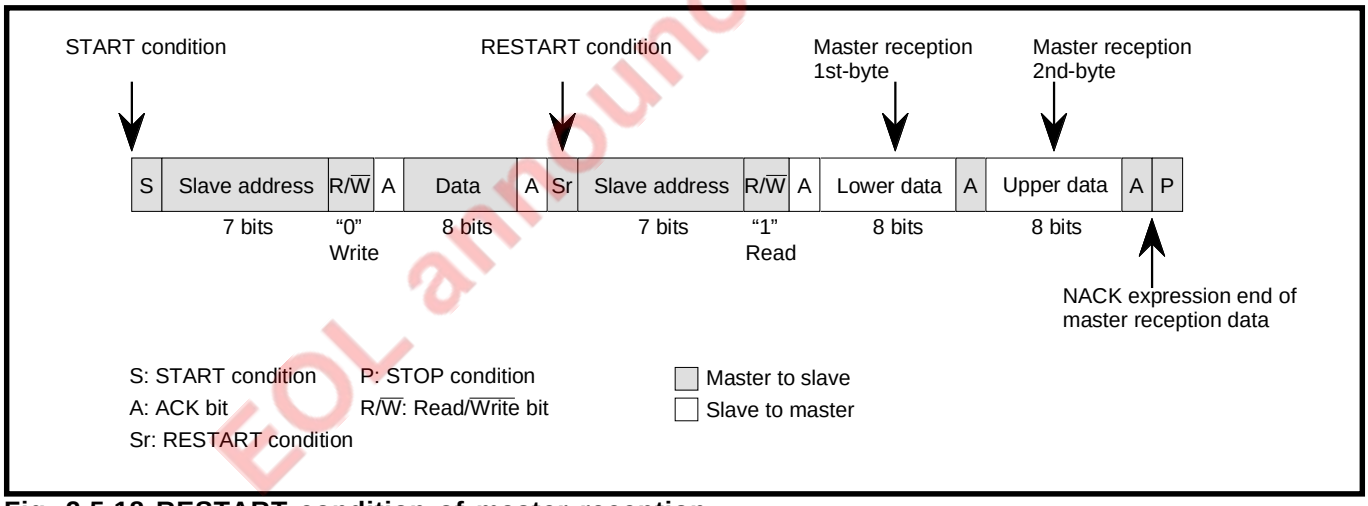


Fig. 2.5.18 RESTART condition of master reception

2.5.5 Synchronization and arbitration lost

(1) Synchronization

When a plural master exists on the I²C-BUS and the masters, which have different speed, are going to simultaneously communicate; there is a rule to unify clocks so that a clock of each bit can be output correctly.

Figure 2.5.19 shows a synchronized SCL line example. The SCL (A) and the SCL (B) are the master devices having a different speed. The SCL is synchronized waveforms.

As shown by Figure 2.5.19, the SCL lines can be synchronized by the following method; the device which first finishes "H" term makes the SCL line "L" and the device which last remains "L" makes the SCL line "H".

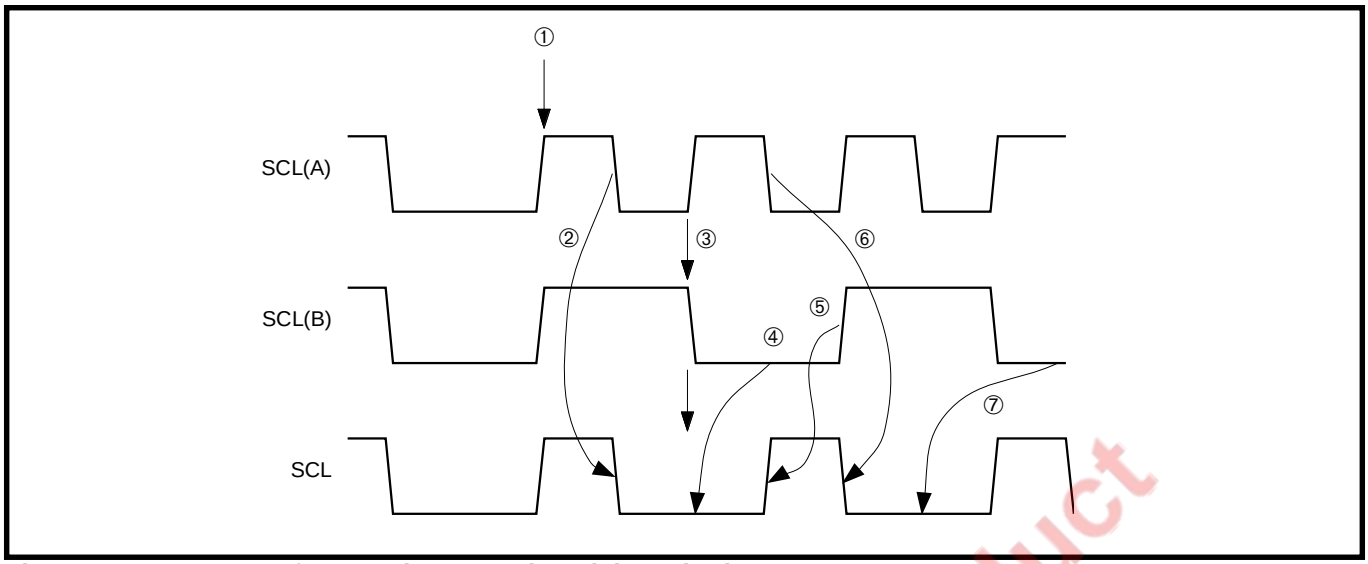


Fig. 2.5.19 SCL waveforms when synchronizing clocks

① After START condition, the masters, which have different speed, simultaneously start clock transmission.
② The SCL outputs “L” because (A) finished counting “H” output; then (B)’s “H” output counting is interrupted and (B) starts counting “L” output.
③ The (A) outputs “H” because (A) finished counting “L” term; the SCL level does not become “H” because (B) outputs “L”, and counting “H” term does not start but stop.
④ (B) outputs “L” term.
⑤ The SCL outputs “H” because (B) finished counting “L” term; then (B)’s “H” output counting is started at the same time as (A).
⑥ The SCL outputs “L” because (A) first finished counting “H” output; then (B)’s “H” output counting is interrupted and (B) starts counting “L” output.
⑦ The above are repeatedly performed.

(2) Clock synchronization during communication

In the I²C-BUS, the slave device is permitted to retain the SCL line “L” and become waiting status for transmission from the master. By byte unit, for the reception preparation of the slave device, the master can become waiting status by making the SCL line “L”, which is after completion of byte reception or the ACK.
By bit unit, it is possible to slow down a clock speed by retaining the clock line “L” for slave devices having limited hardware.

The 3804 group can transmit data correctly without reduction of data bits toward waiting status request from the slave device. It is because the synchronization circuit is included for the case when retaining the SCL line “L” as an internal hardware.
After the last bit, including the ACK bit, of a transmission/reception data byte, the SCL line automatically remains “L” and waiting status is generated until completion of an interrupt process or reception preparation.

(3) Arbitration lost

A plural master exists on the same bus in the I²C-BUS and there are possibility to start communication simultaneously. Even when the master devices having the same transmission frequency start communication simultaneously, which device must transmit data correctly. Accordingly, there is the definition to detect a communication confliction on the SDA line in the I²C-BUS.
The SDA line is output at the timing synchronized by the SCL, however, the synchronization among the SDA signals is not performed.

2.5.6 SMBUS communication usage example

This clause explains a SMBUS communication control example using the I²C-BUS. This is a control example as the master device and the slave device in the Read Word protocol of SMBUS protocol.
The following is a communication example of the “Voltage () command” of the Smart battery data.

Communication specifications:

- Communication frequency = 100 kHz
- Slave address of itself, battery, = “0001011X$_2$” (X means the read/$\overline{\text{write}}$ bit)
- Slave address of communication destination, host, = “0001000X$_2$” (X means the read/$\overline{\text{write}}$ bit)
- Voltage () command = “09$_{16}$”
- Voltage value of acknowledgment = “2EE0$_{16}$” (12000 mV)
- The communication process is performed in the interrupt process. However, the main process performs an occurrence of the first START condition and a slave address set.
- A communication buffer is established. Data transfer between the main process and the interrupt process is performed through the communication buffer.

(1) Initial setting

Figure 2.5.20 shows an initial setting example using SMBUS communication.

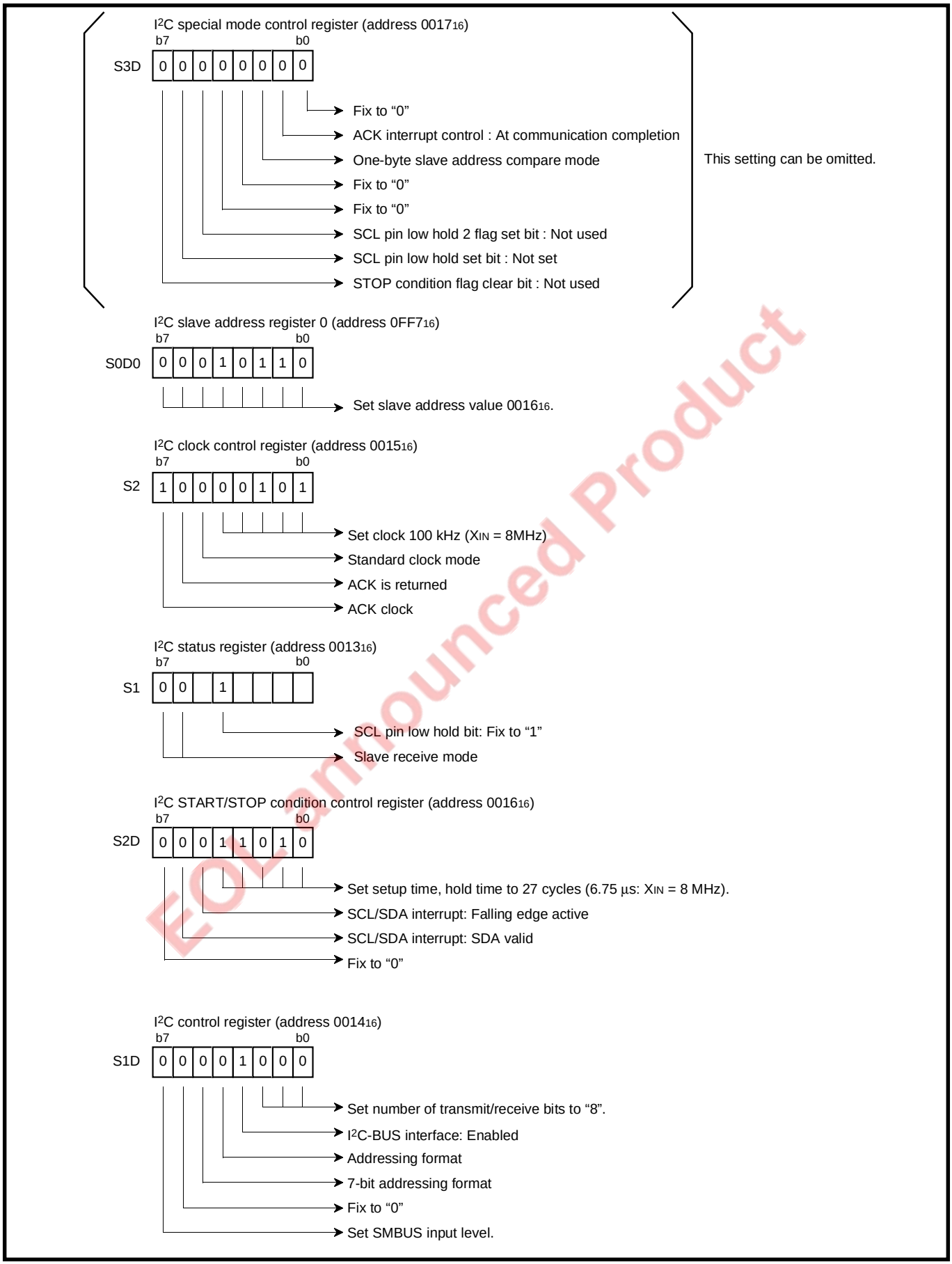


Fig. 2.5.20 Initial setting example for SMBUS communication

(2) Communication example in master device

The master device follows the procedures ① to ⑥ shown by Figure 2.5.21.
Additionally, the shaded area in the figure is a transmission data from the master device and the white area is a transmission data from the slave device.

① Generating of START condition; Transmission of slave address + write bit
② Transmission of command
③ Generating of RESTART condition; Transmission of slave address + read bit
④ Reception of lower data
⑤ Reception of upper data
⑥ Generating of STOP condition

Figures 2.5.22 to 2.5.27 show the procedures ① to ⑥.

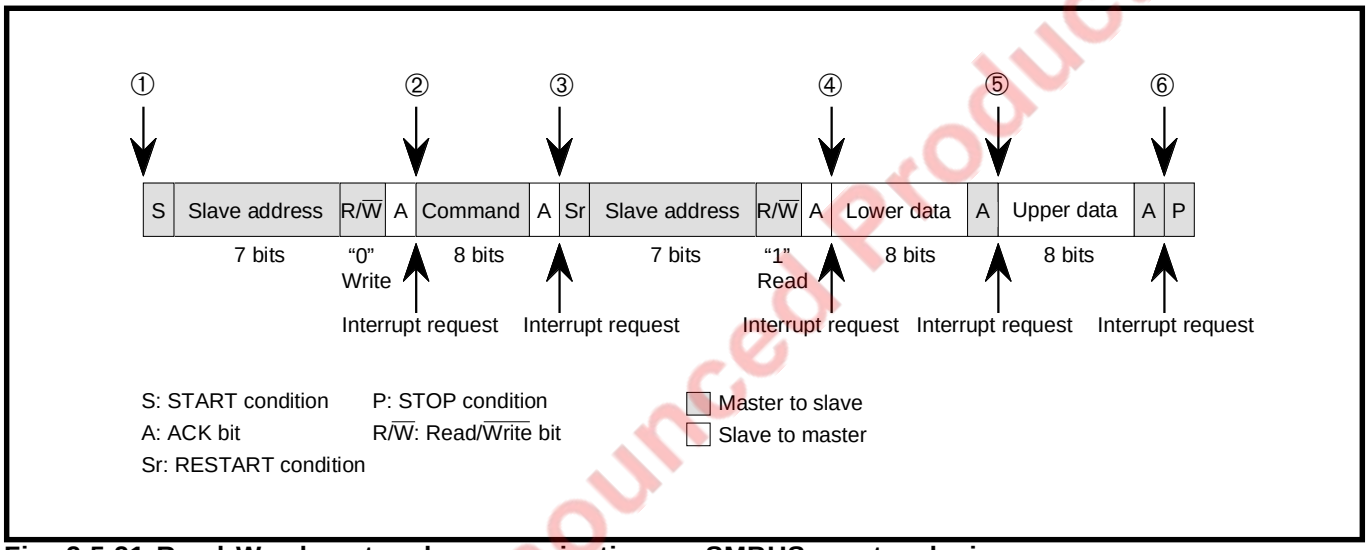


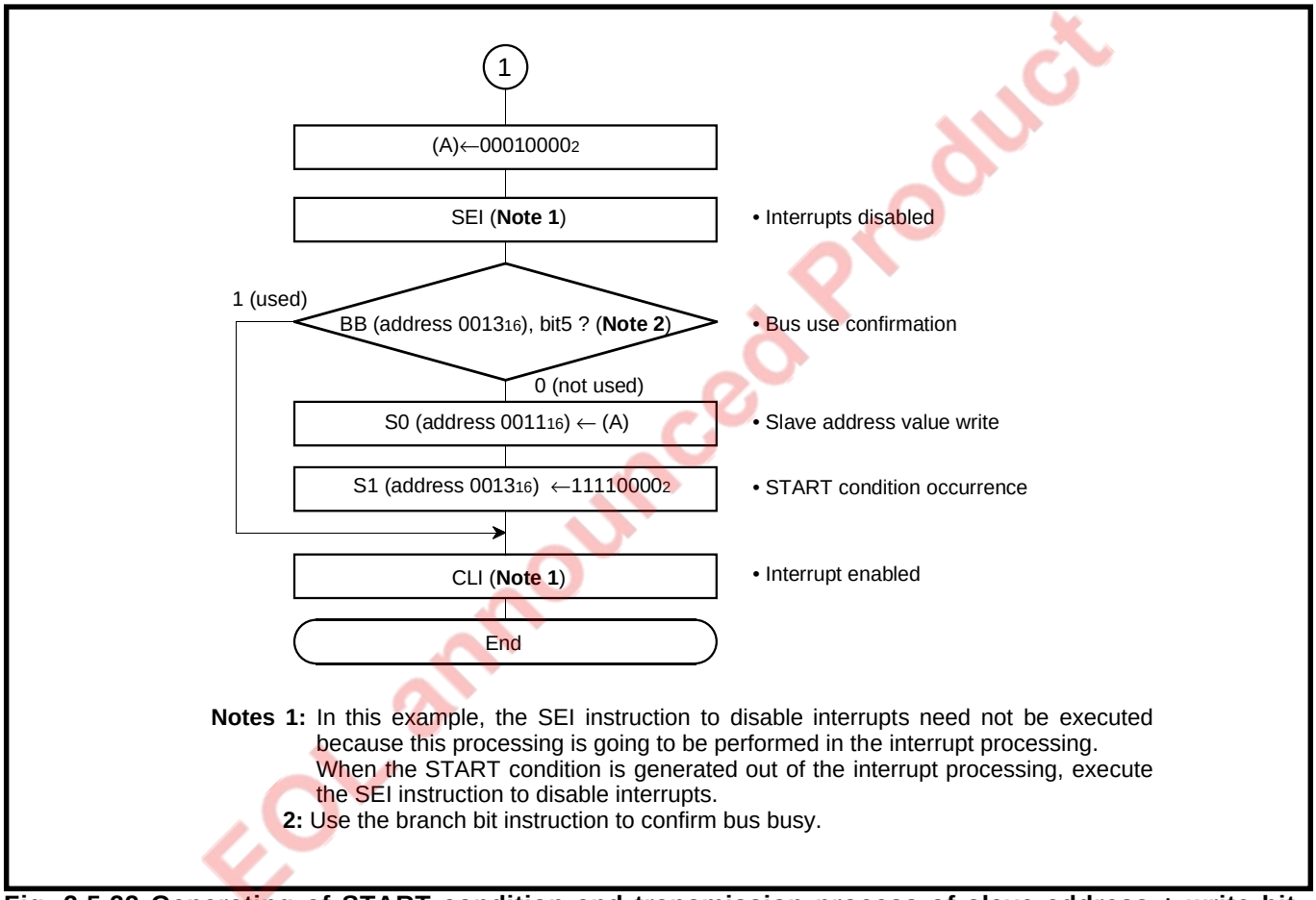
Fig. 2.5.21 Read Word protocol communication as SMBUS master device

① **Generating of START condition; Transmission of slave address + write bit**

After confirming that other master devices do not use the bus, generate the START condition, because the SMBUS is a multi-master.

Write “slave address + write bit” to the I²C data shift register (address 0011_{16}) before performing to make the START condition generate. It is because the SCL of 1-byte unit is output, following occurrence of the START condition.

If other master devices start communication until an occurrence of the START condition after confirming the bus use, it cannot communicate correctly. However in this case, that situation does not affect other master devices owing to detection of an arbitration lost or the START condition duplication preventing function.



Notes 1: In this example, the SEI instruction to disable interrupts need not be executed because this processing is going to be performed in the interrupt processing. When the START condition is generated out of the interrupt processing, execute the SEI instruction to disable interrupts.

2: Use the branch bit instruction to confirm bus busy.

Fig. 2.5.22 Generating of START condition and transmission process of slave address + write bit

② **Transmission of command**

Confirm correct completion of communication at ① before command transmission. When receiving the STOP condition, a process not to transmit a command is required, because the internal I²C-BUS generates an interrupt request also owing to the STOP condition transmitted to other devices.

After confirming correct completion of communication, write a command to the I²C data shift register (address 0011_{16}).
In case the AL bit (bit 3 of address 0013_{16}) is "1", check the slave address comparison flag (AAS bit; bit 2 of address 0013_{16}) to judge whether the device given a right of master transmission owing to an arbitration specifies itself as a slave address. When it is "1", perform the slave reception; when "0", wait for a STOP condition occurrence caused by other devices and the communication completion.
In case the AL bit is "0", check the last received bit (LRB bit; bit 0 of address 0013_{16}). When it is "1", make the STOP condition generate and release the bus use, because the specified slave device does not exist on the SMBUS.

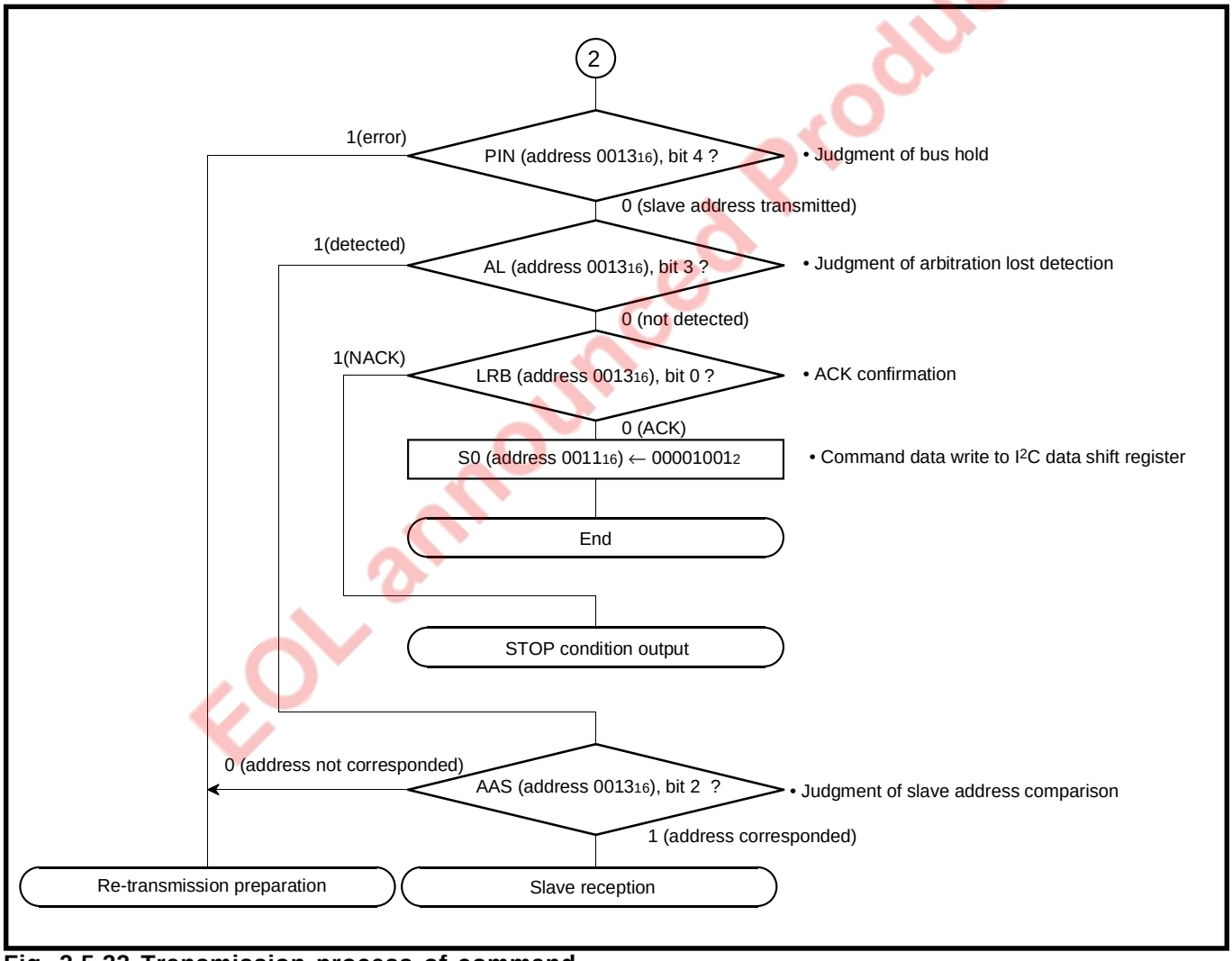


Fig. 2.5.23 Transmission process of command

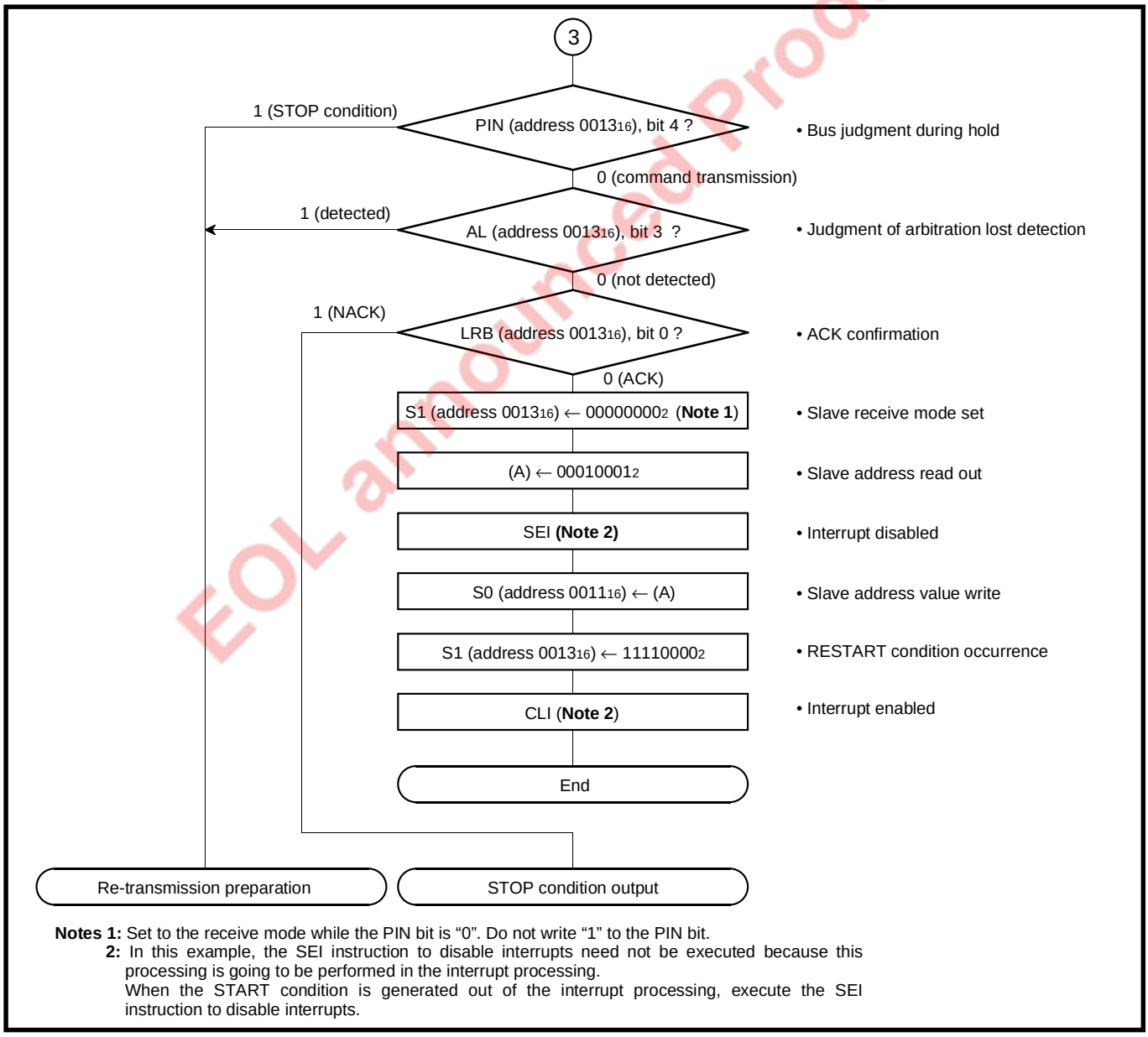
③ **Generating of RESTART condition; Transmission of slave address + read bit**

Confirm correct completion of communication at ② before generating the RESTART condition. After confirming correct completion, generate the RESTART condition and perform the transmission process of "slave address + read bit". Note that procedure because that is different from ①'s process.

As the same reason as ①, write "slave address + read bit" to the I²C data shift register (address 0011_{16}) before performing to make the START condition generate. However, when writing a slave address to the I²C data shift register in this condition, a slave address is output at that time. Consequently, the RESTART condition cannot be generated. Therefore, follow the slave reception procedure before those processes.

In case the arbitration lost detecting flag (AL bit, bit 3 of address 0013_{16}) is "1", return to the process ①, because other master devices will have priority to communicate.

When the last received bit (LRB bit; bit 0 of address 0013_{16}) is "1", generate the STOP condition and make the bus release, because acknowledgment cannot be done owing to BUSY status of the slave device specified on the SMBUS or other reasons.



Notes 1: Set to the receive mode while the PIN bit is "0". Do not write "1" to the PIN bit.

2: In this example, the SEI instruction to disable interrupts need not be executed because this processing is going to be performed in the interrupt processing.
When the START condition is generated out of the interrupt processing, execute the SEI instruction to disable interrupts.

Fig. 2.5.24 Transmission process of RESTART condition and slave address + read bit

④ **Reception of lower data**

Confirm correct completion of communication at ③ before receiving the lower data. After confirming correct completion, clear the ACK bit (bit 6 of address 0015_{16}) to "0", in which ACK is returned, and set to the master receive mode. After that, write dummy data to the I²C data shift register (address 0011_{16}).

When the MST bit (bit 7 of address 0013_{16}) is "0", perform the error process explained as follows and return to the process ①.

When the last receive bit (LRB bit; bit 0 of address 0013_{16}) is "1", generate the STOP condition and make the bus release, because the slave device specified on the SMBUS does not exist.

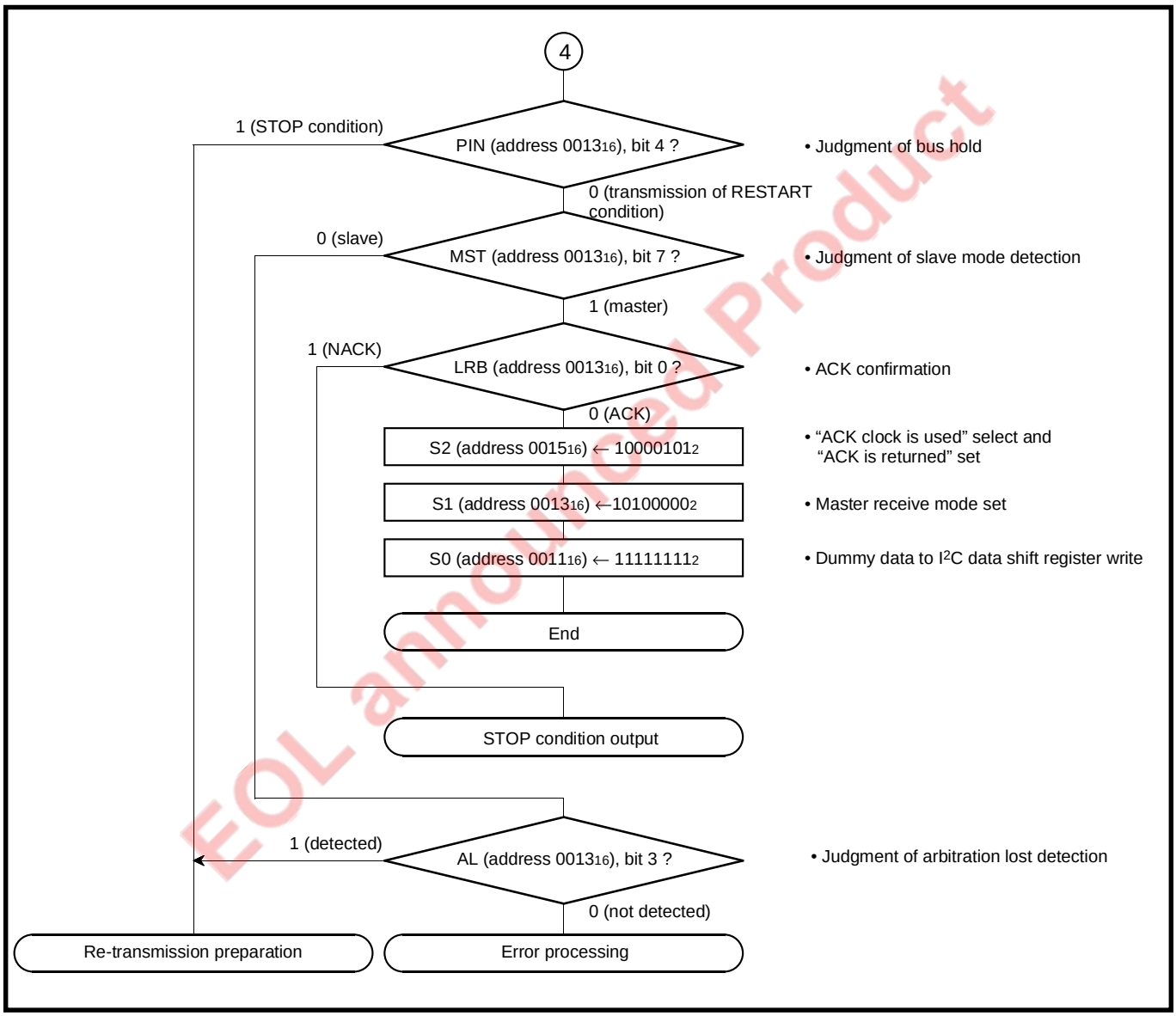


Fig. 2.5.25 Reception process of lower data

⑤ **Transmission of upper data**

Confirm correct completion of communication at ④ before receiving the upper data. After confirming correct completion, store the received data (lower data).

Set the ACK bit (bit 6 of address 0015_{16}) to "1", in which ACK is not returned and write dummy data to the I²C data shift register (address 0011_{16}).

When the MST bit (bit 7 of address 0013_{16}) is "0", return to the process ①, because other devices have priority to communicate.

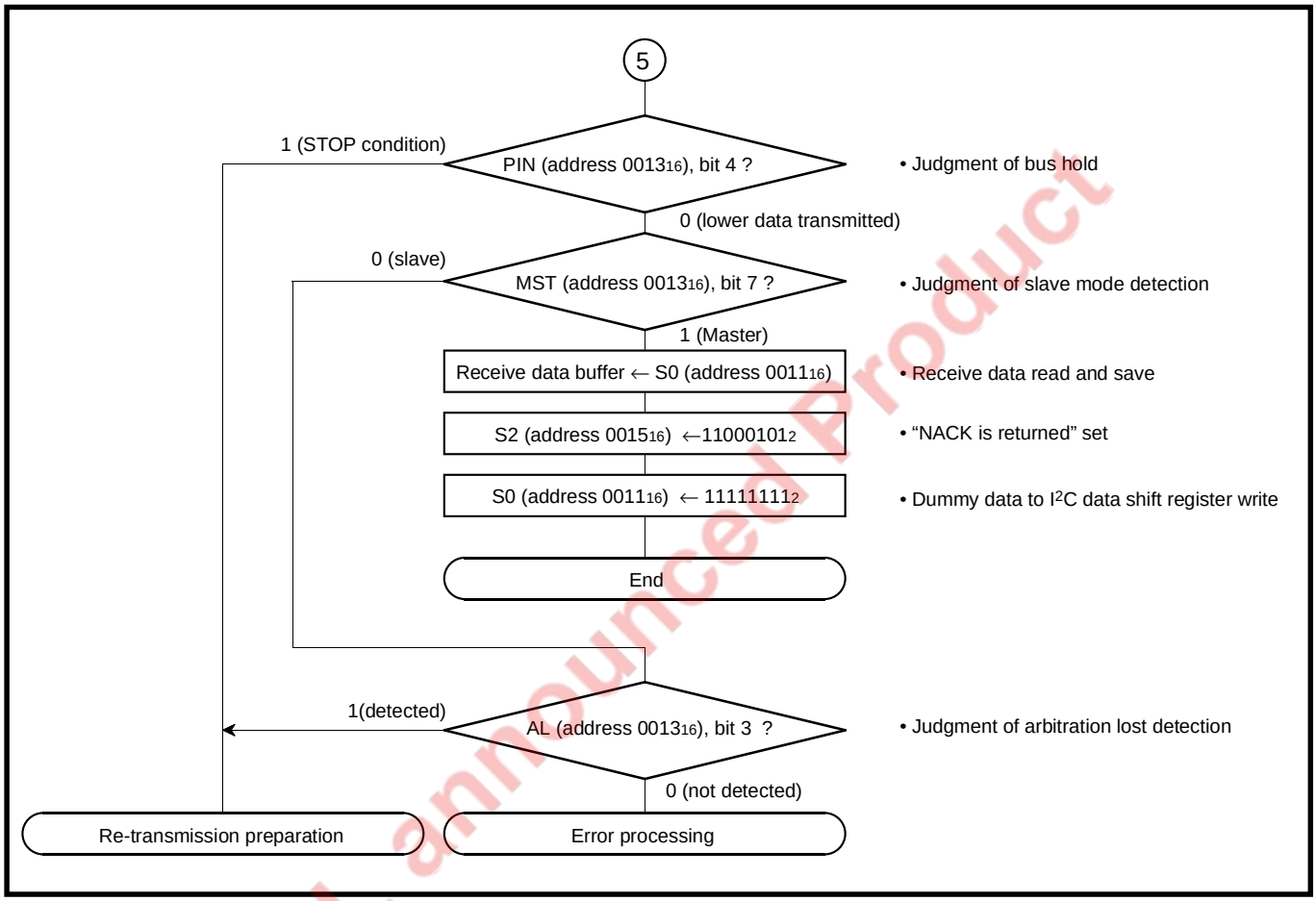


Fig. 2.5.26 Reception process of upper data

⑥ **Generating of STOP condition**

Confirm correct completion of communication at ⑤ before generating the STOP condition. After confirming correct completion, store the received data (upper data).

Clear the ACK bit (bit 6 of address 0015_{16}) to "0", in which ACK is returned, and generate the STOP condition. The communication mode is set to the slave receive mode by the occurrence of STOP condition.

When the MST bit (bit 7 of address 0013_{16}) is "0", return to the process ①, because other devices have priority to communicate.

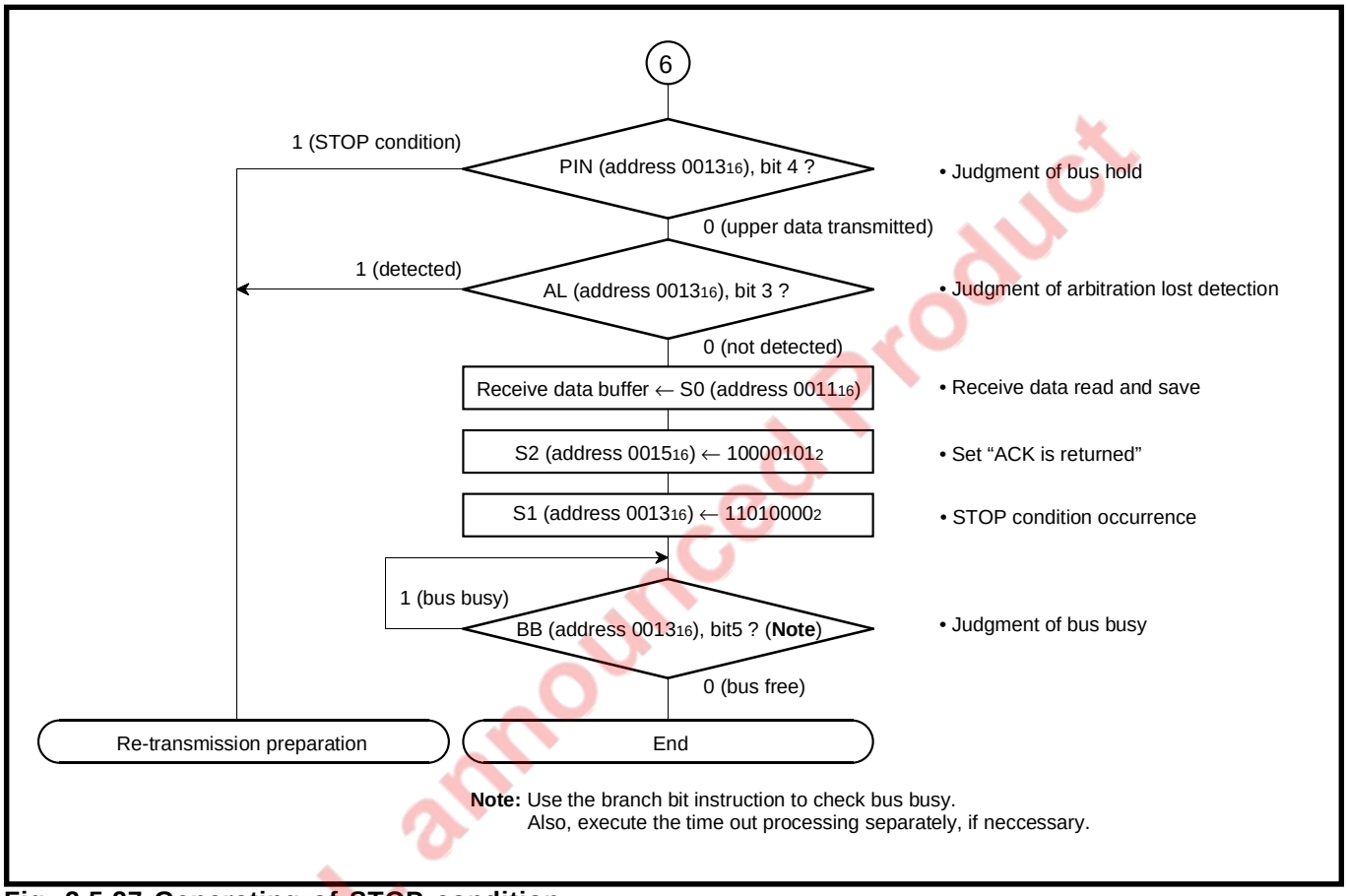


Fig. 2.5.27 Generating of STOP condition

(3) Communication example in slave device

The slave device follows the procedures ① to ⑥ shown by Figure 2.5.28.
The only difference from the master device's communication is an occurrence of interrupt request after detection of STOP condition.

① Reception of START condition; Transmission of ACK bit due to slave address correspondence
② Reception of command
③ Reception of RESTART condition; Reception of slave address + read bit
④ Transmission of lower data
⑤ Transmission of upper data
⑥ Reception of STOP condition

Figures 2.5.29 to 2.5.34 show the procedures ① to ⑥.

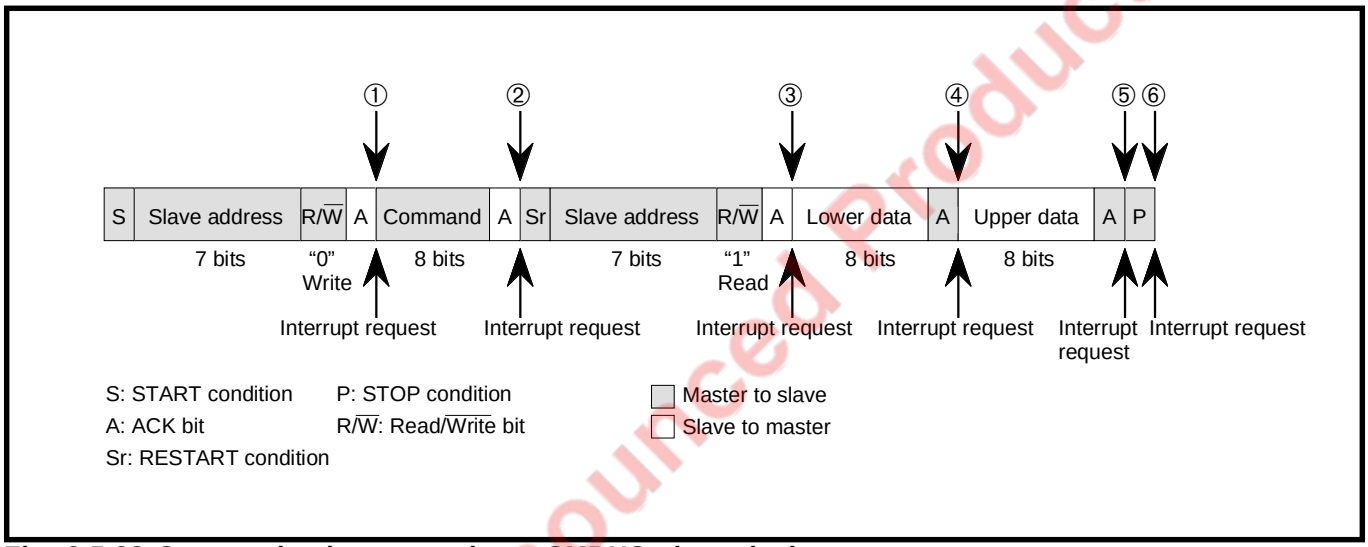


Fig. 2.5.28 Communication example as SMBUS slave device

① **Reception of START condition; Transmission of ACK bit due to slave address correspondence**
In the case of operation as the slave, all processes are performed in the interrupt after setting of the slave reception in the main process, because an interrupt request does not occur until correspondence of a slave address.
In the first interrupt, after confirming correspondence of the slave address, write dummy data to receive a command into the I²C data shift register.

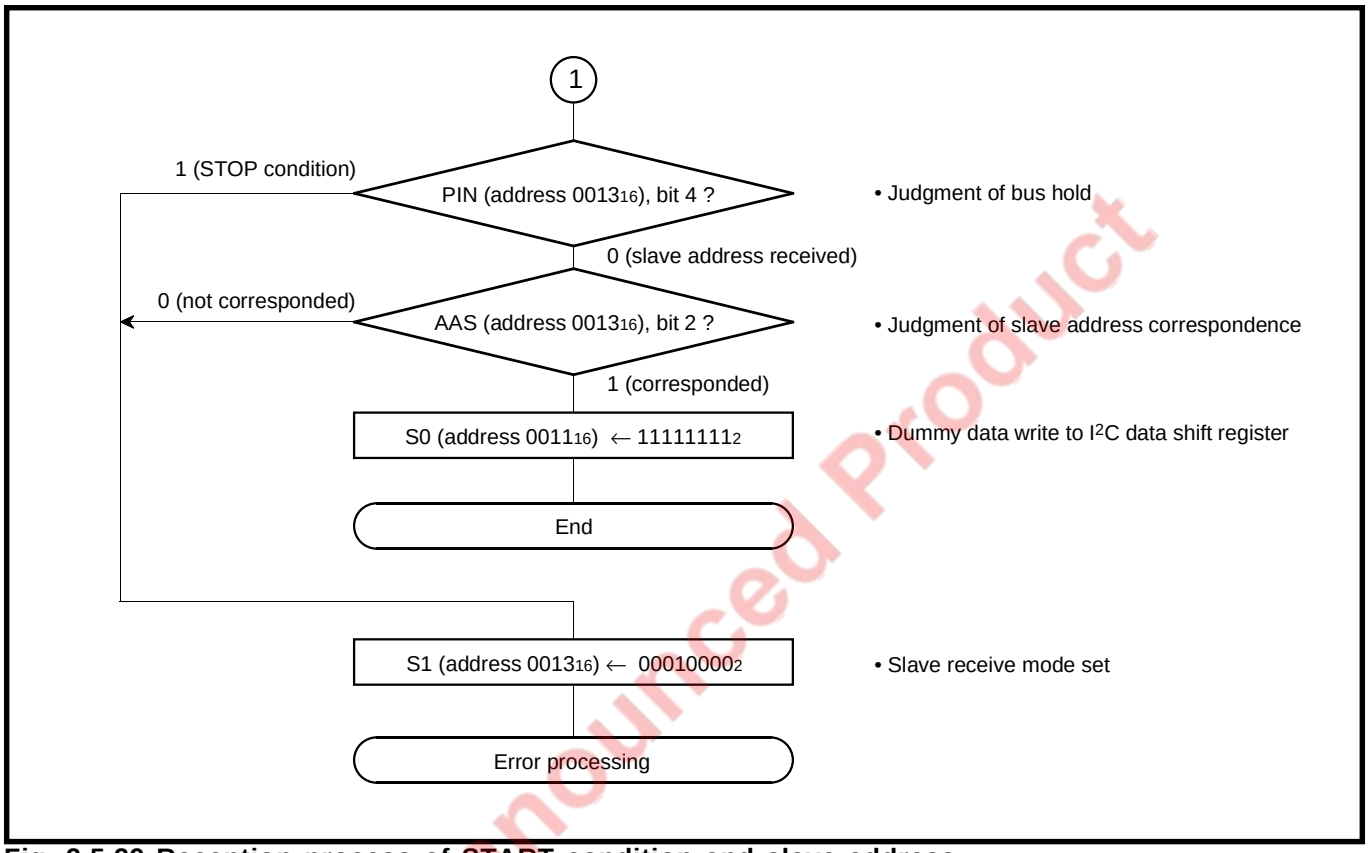


Fig. 2.5.29 Reception process of START condition and slave address

② **Reception of command**

Confirm correct completion of the command reception in the interrupt after receiving the command. After confirming correct command from the host, write dummy data to the I²C data shift register to wait for reception of the next slave address.

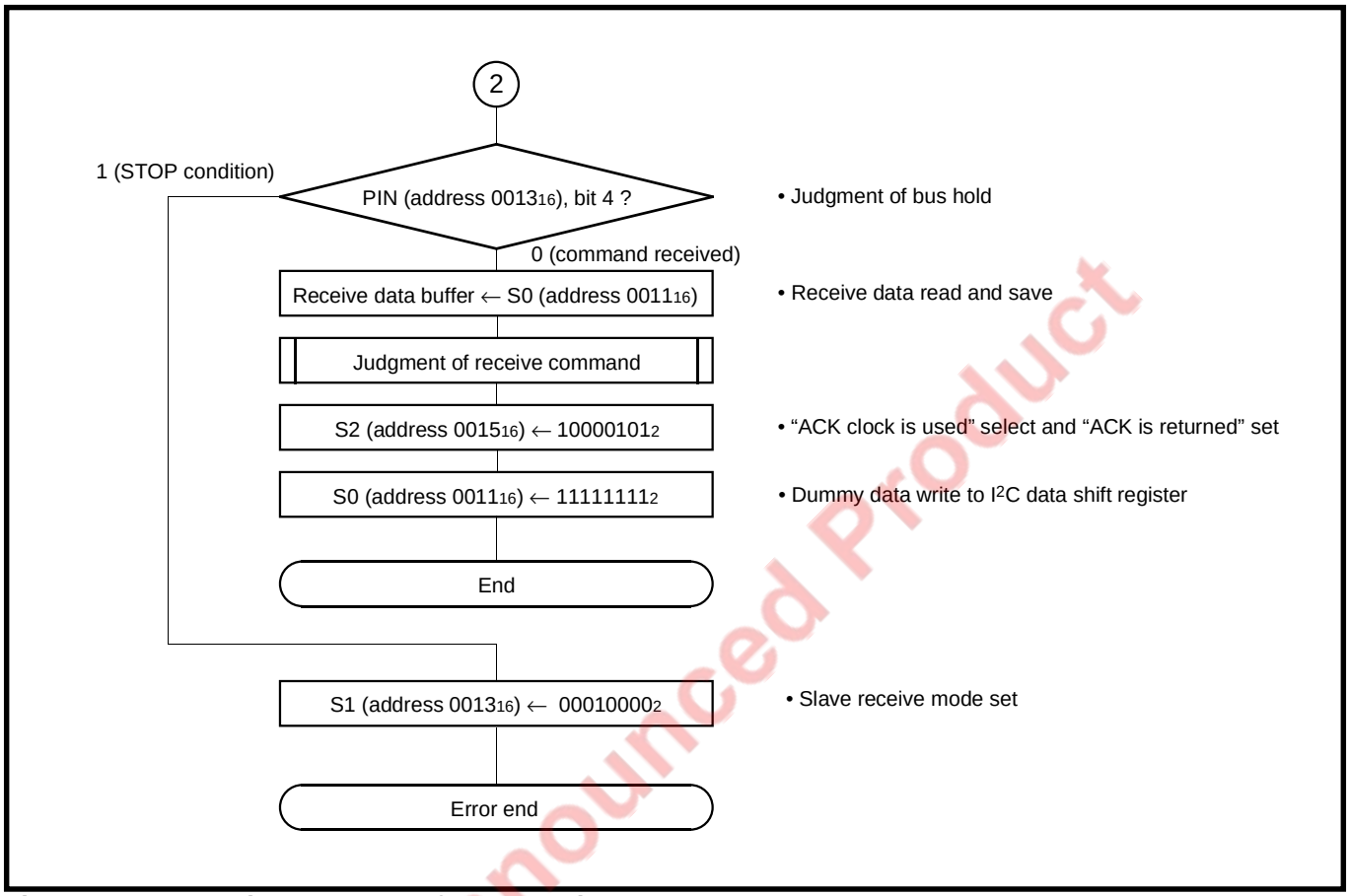


Fig. 2.5.30 Reception process of command

③ **Reception of RESTART condition and slave address**

After receiving a slave address, prepare transmission data.

Judgment whether receiving data or transmitting is required, because the mode is automatically switched between the receive mode and the transmit mode depending on the R/$\overline{W}$ bit of the received slave address. Accordingly, judge whether read or write referring the slave address comparison flag (AAS bit; bit 2 of address 0013_{16}).

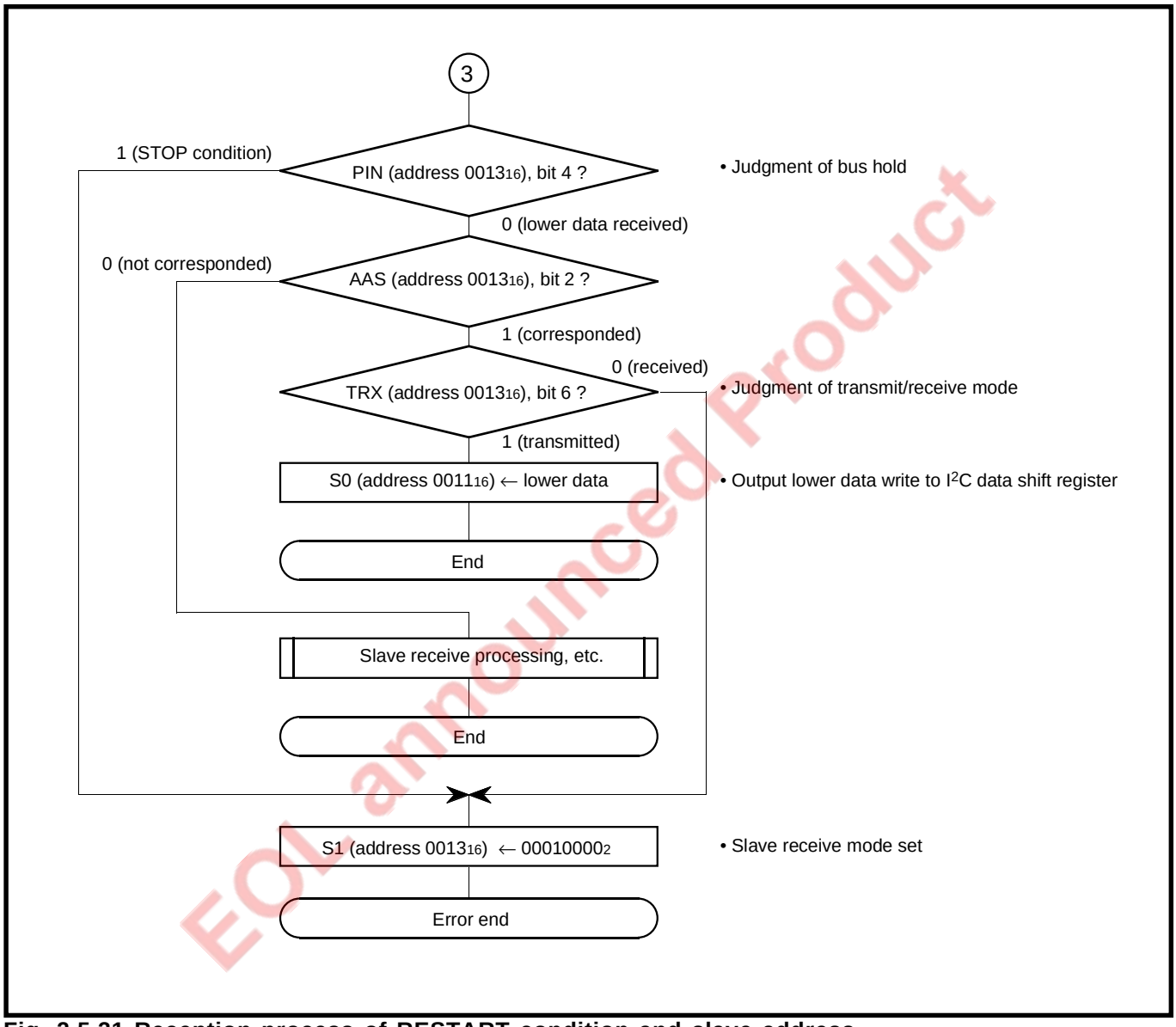


Fig. 2.5.31 Reception process of RESTART condition and slave address

④ **Transmission of lower data**

Before transmitting the upper data, restart to transmit the data at ④ and confirm correct completion of transmission of the lower data set in the slave address reception interrupt.
After that, transmit the upper data.

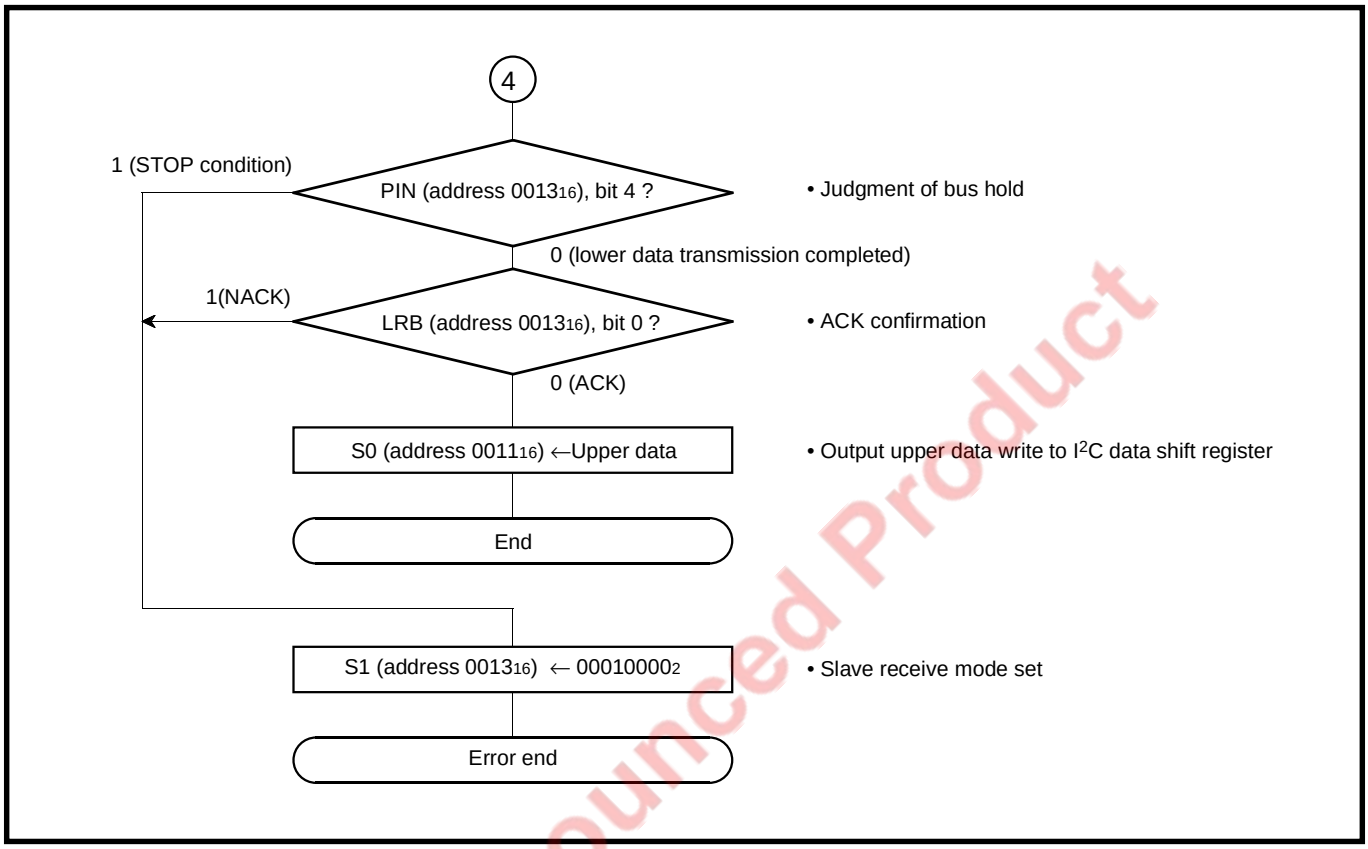


Fig. 2.5.32 Transmission process of lower data

⑤ **Transmission of upper data**

Confirm correct completion of the upper data transmission. The master returns the NACK toward the transmitted second-byte data, the upper data. Accordingly, confirm that the last receive bit (LRB bit; bit 0 of address 0013_{16}) is “1”.

After that, write dummy data to the I²C data shift register and wait for the interrupt of STOP condition.

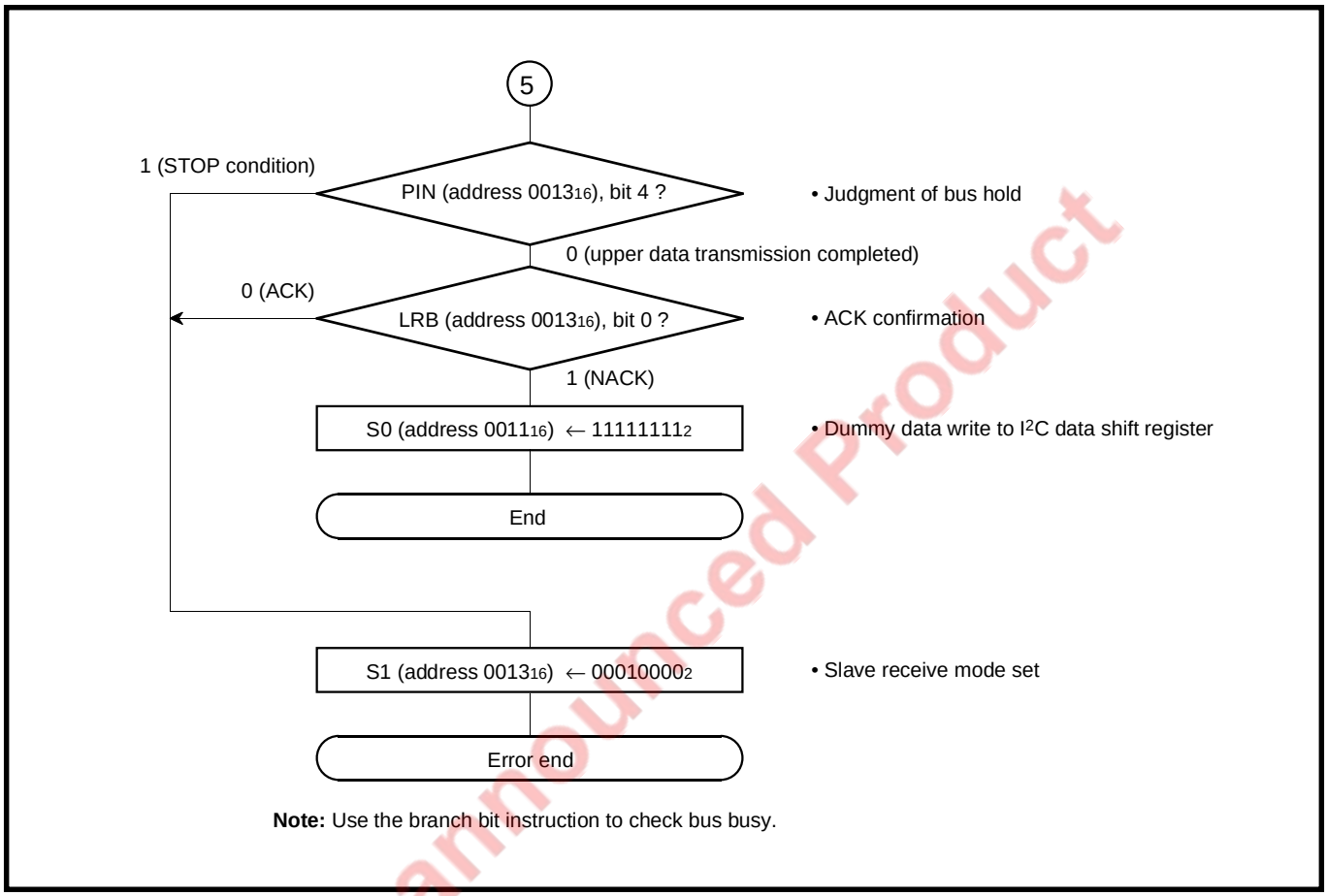


Fig. 2.5.33 Transmission process of upper data

⑥ **Reception of STOP condition**

Confirm that the STOP condition is correctly output and the bus is released.

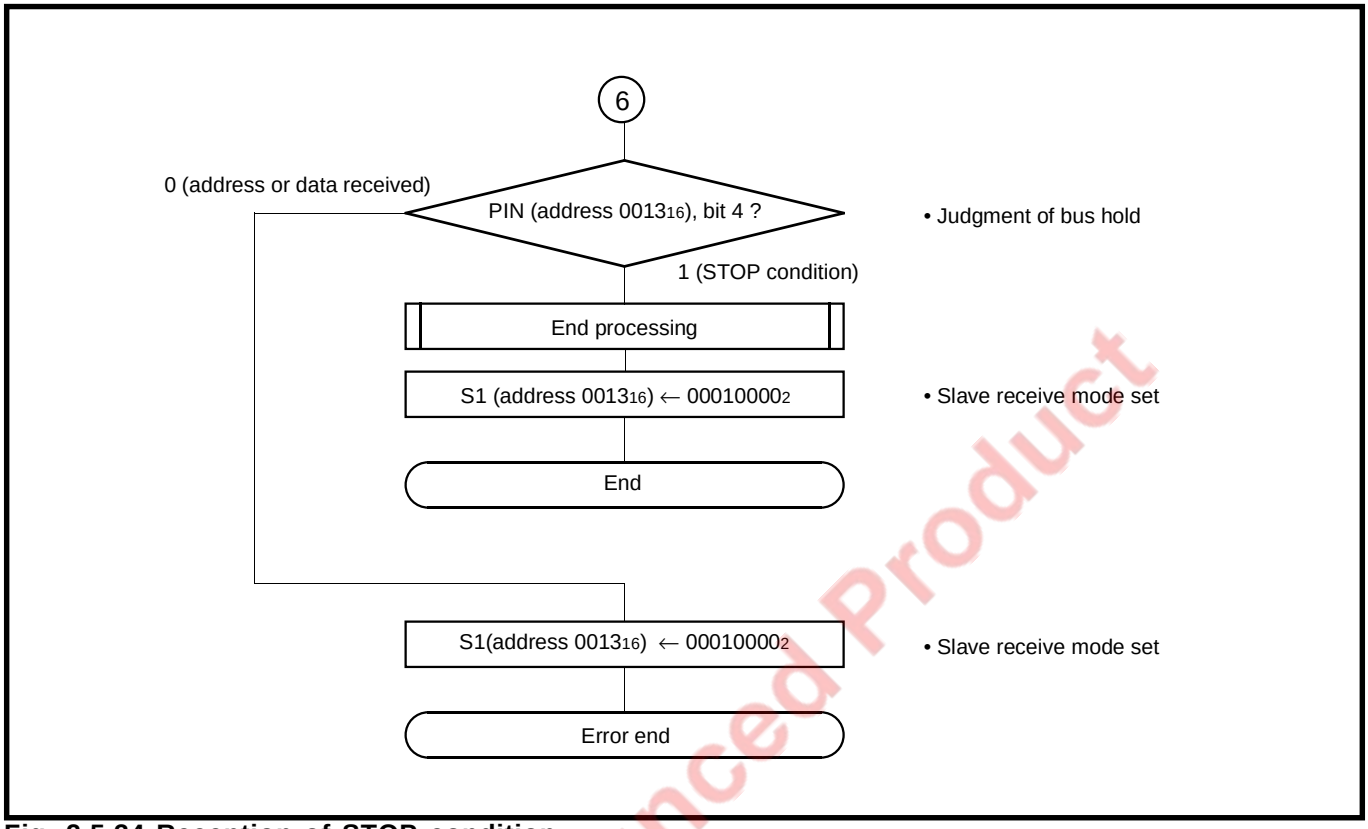


Fig. 2.5.34 Reception of STOP condition

2.5.7 Notes on multi-master I²C-BUS interface

(1) Read-modify-write instruction

Each register of the multi-master I²C-BUS interface has bits to change by hardware. The precautions when the read-modify-write instruction such as **SEB**, **CLB** etc. is executed for each register of the multi-master I²C-BUS interface are described below.

① **I²C data shift register (S0: address 0011_{16})**
When executing the read-modify-write instruction for this register during transfer, data may become a value not intended.

② **I²C slave address registers 0 to 2 (S0D0 to S0D2: addresses $0FF7_{16}$ to $0FF9_{16}$)**
When the read-modify-write instruction is executed for this register at detecting the STOP condition, data may become a value not intended.

● **Reason**
It is because hardware changes the read/$\overline{\text{write}}$ bit (RWB) at detecting the STOP condition.

③ **I²C status register (S1: address 0013_{16})**
Do not execute the read-modify-write instruction for this register because all bits of this register are changed by hardware.

④ **I²C control register (S1D: address 0014_{16})**
When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become a value not intended.

● **Reason**
Because hardware changes the bit counter (BC0 to BC2).

⑤ **I²C clock control register (S2: address 0015_{16})**
The read-modify-write instruction can be executed for this register.

⑥ **I²C START/STOP condition control register (S2D: address 0016_{16})**
The read-modify-write instruction can be executed for this register.

(2) START condition generating procedure using multi-master

① Procedure example (The necessary conditions of the generating procedure are described as the following ② to ⑤).

```
        LDA #SLADR          (Taking out of slave address value)
        SEI                 (Interrupt disabled)
        BBS 5, S1, BUSBUSY  (BB flag confirming and branch process)
BUSFREE:
        STA S0              (Writing of slave address value)
        LDM #$F0, S1        (Trigger of START condition generating)
        CLI                 (Interrupt enabled)
        :
        :
BUSBUSY:
        CLI                 (Interrupt enabled)
        :
        :
```

② Use “Branch on Bit Set” of “BBS 5, S1, –” for the BB flag confirming and branch process.

③ Use “STA, STX” or “STY” of the zero page addressing instruction for writing the slave address value to the I²C data shift register (S0: address 0011_{16}).

④ Execute the branch instruction of above ② and the store instruction of above ③ continuously shown by the above procedure example.

⑤ Disable interrupts during the following three process steps:

- BB flag confirming
- Writing of slave address value
- Trigger of START condition generating

(3) RESTART condition generating procedure in master

① Procedure example (The necessary conditions of the generating procedure are described as the following ② to ④). Execute the following procedure when the PIN bit is “0”.

```
        LDM #$00, S1        (Select slave receive mode)
        LDA #SLADR          (Taking out of slave address value)
        SEI                 (Interrupt disabled)
        STA S0              (Writing of slave address value)
        LDM #$F0, S1        (Trigger of RESTART condition generating)
        CLI                 (Interrupt enabled)
        :
        :
```

② Select the slave receive mode when the PIN bit is “0”. Do not write “1” to the PIN bit. The TRX bit becomes “0” and the SDA pin is released.

③ The SCL pin is released by writing the slave address value to the I²C data shift register.

④ Disable interrupts during the following two process steps:

- Writing of slave address value
- Trigger of RESTART condition generating

(4) Writing to I²C status register (S1: address 0013_{16})

Do not execute an instruction to set the PIN bit to “1” from “0” and an instruction to set the MST and TRX bits to “0” from “1” simultaneously. It is because it may enter the state that the SCL pin is released and the SDA pin is released after about one machine cycle. Do not execute an instruction to set the MST and TRX bits to “0” from “1” simultaneously when the PIN bit is “1”. It is because it may become the same as above.

(5) Writing to I²C clock control register (S2: address 0015_{16})

Do not write data into the I²C clock control register during transfer. If data is written during transfer, the I²C clock generator is reset, so that data cannot be transferred normally.

(6) Switching of SCL/SDA interrupt pin polarity selection bit, SCL/SDA interrupt pin selection bit, I²C-BUS interface enable bit

When changing the setting of the SCL/SDA interrupt pin polarity selection bit, the SCL/SDA interrupt pin selection bit, or the I²C-BUS interface enable bit ES0, the SCL/SDA interrupt request bit may be set. When selecting the SCL/SDA interrupt source, disable the interrupt before the SCL/SDA interrupt pin polarity selection bit, the SCL/SDA interrupt pin selection bit, or the I²C-BUS interface enable bit ES0 is set. Reset the request bit to "0" after setting these bits, and enable the interrupt.

(7) Process of after STOP condition generating in master mode

Do not write data in the I²C data shift register (S0) and the I²C status register (S1) until the bus busy flag BB becomes "0" after generating the STOP condition in the master mode. It is because the STOP condition waveform might not be normally generated. Reading to the above registers does not have the problem.

(8) ES0 bit switch

In standard clock mode when SSC = "00010_2" or in high-speed clock mode, flag BB may switch to "1" if ES0 bit is set to "1" when SDA is "L".

Countermeasure:

Set ES0 to "1" when SDA is "H".

2.5.8 Notes on programming for SMBUS interface

(1) Time out process

For a smart battery system, the time out process with a program is required so that the communication can be completed even when communication is interrupted. It is because there is possibility of extracting a battery from a PC.

The specifications are defined so that communication has been able to be completed within 25 ms from START condition to STOP condition and within 10 ms from the ACK pulse to the ACK pulse of each byte. Accordingly, the following two should be considered as count start conditions.

① **SDA falling edge caused by SCL/SDA interrupt**

This is the countermeasure for a communication interrupt in the middle of from START condition to a slave address. However, the detection condition must be considered because a interrupt is also generated by communication from other masters to other slaves.

② **SMBUS interrupt after receiving slave address**

This is the countermeasure for when communication is interrupted from receiving a slave address until receiving a command.

(2) Low hold of communication line

The I²C-BUS interface conforms to the I²C-BUS Standard Specifications. However, because the use condition of SMBUS differs from the I²C-BUS's, there is possibility of occurrence of the following problem.

① **Low hold of SDA line caused by ACK pulse at voltage drop of communication line**

When the SMBUS voltage slowly drops, that is caused by extracting a battery from equipment or turning off a PC's power or etc., it might be incorrectly treated as the SCL pulse near the threshold level voltage.

When the SDA is judged "L" in that condition, it becomes the general call and the ACK is transmitted. However, when the SCL remains "L" at the ACK pulse, the SDA continuously remains "L" until input of the next SCL pulse.

Countermeasure:

As explained before, start the time out count at the falling of SDA line of START condition and reset ES0 bit of the S1D register when the time out is satisfied (**Note**).

Note: Do not use the read-modify-write instruction at this time. Furthermore, when the ES0 bit is set to "0", it becomes a general-purpose port ; so that the port must be set to input mode or "H".

2.6 PWM

This paragraph explains the registers setting method and the notes relevant to the PWM.

2.6.1 Memory map

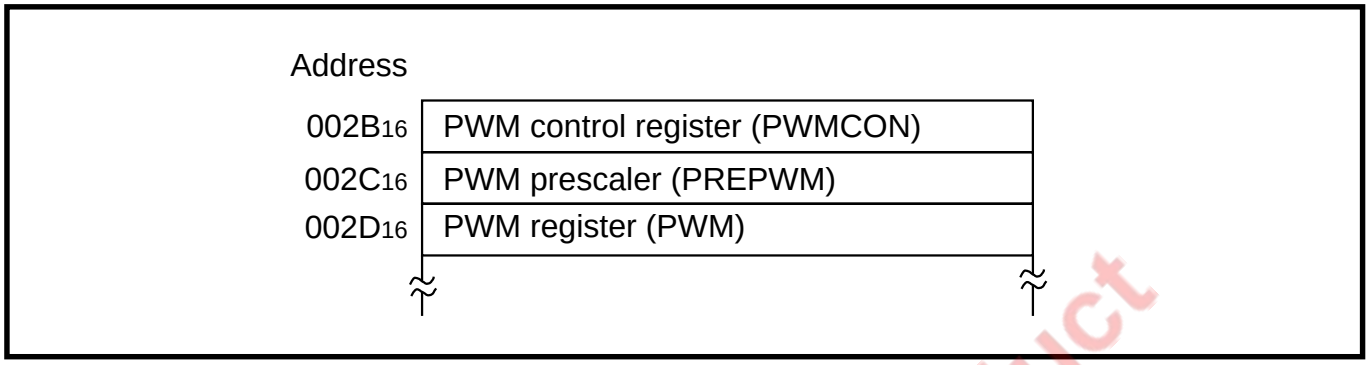


Fig. 2.6.1 Memory map of registers relevant to PWM

2.6.2 Relevant registers

PWM control register

b7 b6 b5 b4 b3 b2 b1 b0

PWM control register (PWMCON: address 002B16)

b	Name	Functions	At reset	R	W
0	PWM function enable bit	0 : PWM disabled 1 : PWM enabled	0	○	○
1	Count source selection bit	0 : f(XIN) 1 : f(XIN)/2	0	○	○
2	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 2.6.2 Structure of PWM control register

PWM prescaler

b7 b6 b5 b4 b3 b2 b1 b0

PWM prescaler
(PREPWM: address 002C16)

b	Functions	At reset	R	W
0	•Set the PWM period. •The value set in this register is written to both PWM prescaler pre-latch and PWM prescaler latch at the same time. • When data is written to this register during PWM output, the pulse corresponding to changed value is output at the next period. • When this register is read out, the count value of the PWM prescaler latch is read out.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 2.6.3 Structure of PWM prescaler

PWM register

b7 b6 b5 b4 b3 b2 b1 b0

PWM register
(PWM: address 002D16)

b	Functions	At reset	R	W
0	• Set the PWM “H” level output interval. • The value set in this register is written to both PWM register pre-latch and PWM register latch at the same time. • When data is written to this register during PWM output, the pulse corresponding to changed value is output at the next period. • When this register is read out, the contents of the PWM register latch is read out.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 2.6.4 Structure of PWM register

2.6.3 PWM output circuit application example

<Motor control>

Outline : The rotation speed of the motor is controlled by using PWM (pulse width modulation) output.

Figure 2.6.5 shows a connection diagram ; Figures 2.6.6 shows PWM output timing, and Figure 2.6.7 shows a setting of the related registers.

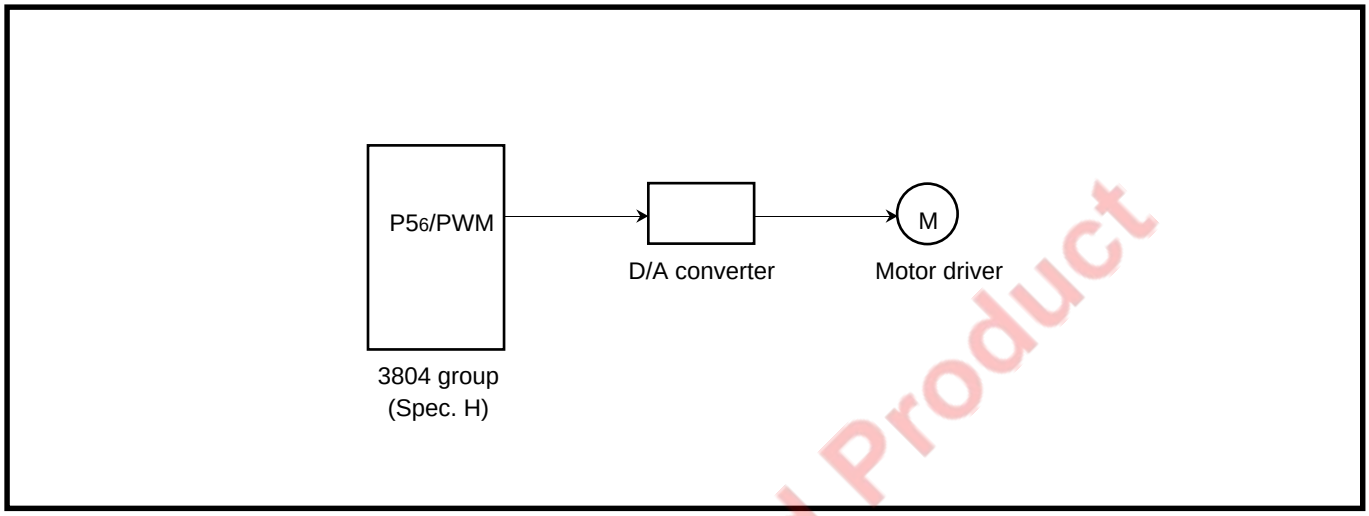


Fig. 2.6.5 Connection diagram

Specifications : • Motor is controlled by using the PWM output function of 8-bit resolution.
• Clock f(XIN) = 5 MHz
• "T", PWM cycle : 102 μs
• "t", "H" level width of output pulse : 40 μs (Fixed speed)

✻ A motor speed can be changed by modifying the "H" level width of output pulse.

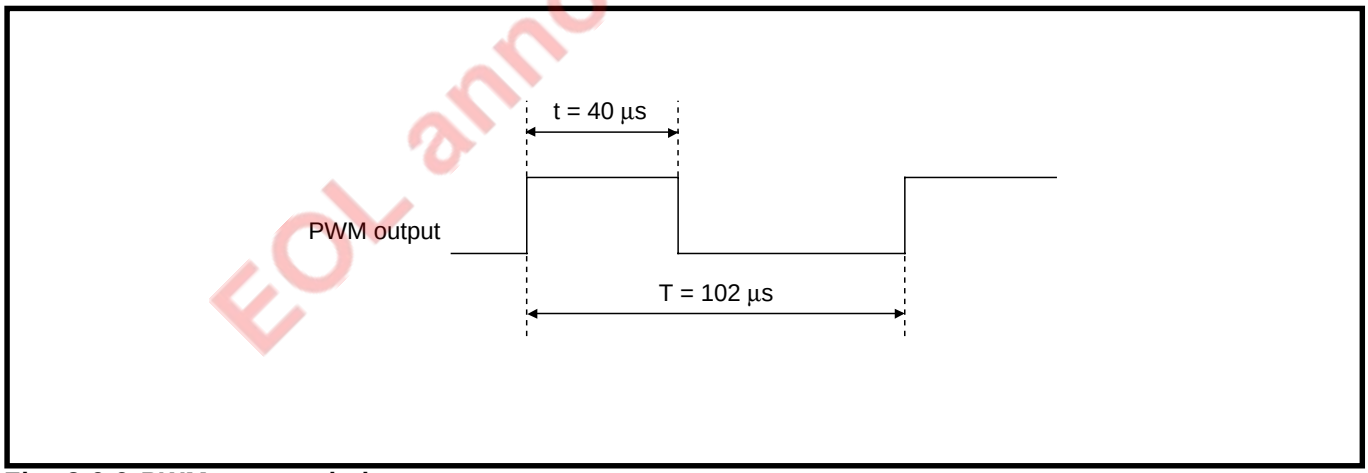


Fig. 2.6.6 PWM output timing

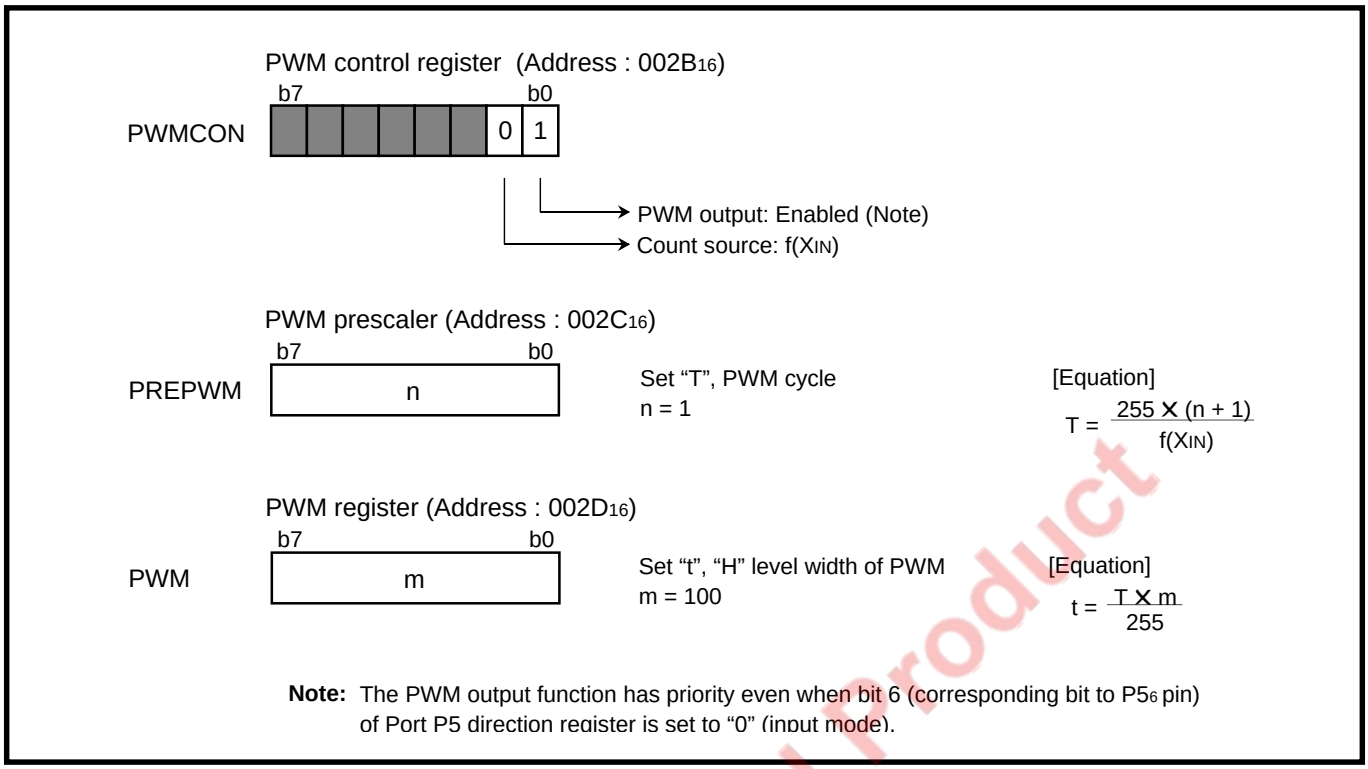


Fig. 2.6.7 Setting of relevant registers

<About PWM output>

1. Set the PWM function enable bit to "1" : The P56/PWM pin is used as the PWM pin.
   The pulse beginning with "H" level pulse is output.
2. Set the PWM function enable bit to "0" : The P56/PWM pin is used as the port P56.
   Thus, when fixing the output level, take the following procedure:
   (1) Write an output value to bit 6 of the port P5 register.
   (2) Write "01000000$_2$" to the port P5 direction register.
3. After data is set to the PWM prescaler and the PWM register, the PWM waveforms corresponding to updated data will be output from the next repetitive cycle.

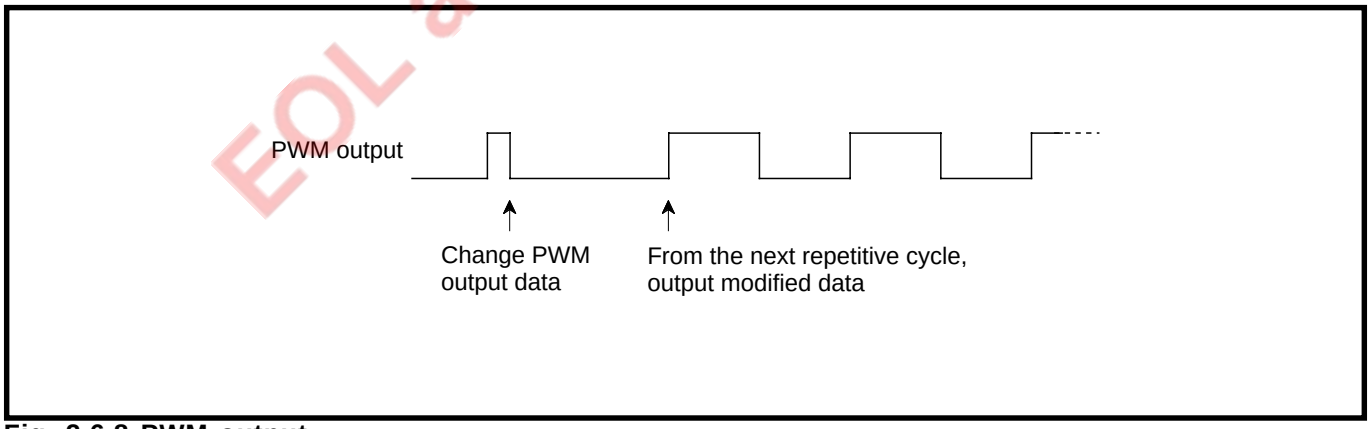


Fig. 2.6.8 PWM output

By setting the related registers as shown by Figure 2.6.7, PWM waveforms are output to the externals. This PWM output is integrated through the low pass filter, and that converted into DC signals is used for control of the motor. Figure 2.6.9 shows control procedure.

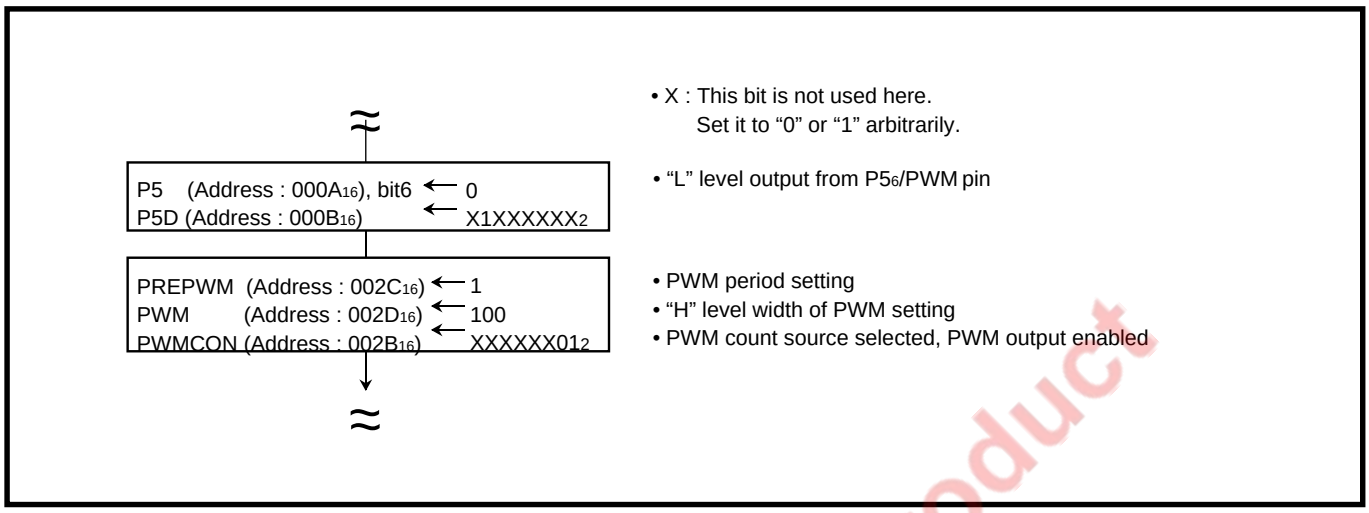


Fig. 2.6.9 Control procedure

2.6.4 Notes on PWM

The PWM starts after the PWM enable bit is set to enable and "L" level is output from the PWM pin. The length of this "L" level output is as follows:

$$\frac{n + 1}{2 \cdot f(X_{IN})} \text{ (s)} \quad \text{(Count source selection bit = 0, where n is the value set in the prescaler)}$$

$$\frac{n + 1}{f(X_{IN})} \text{ (s)} \quad \text{(Count source selection bit = 1, where n is the value set in the prescaler)}$$

2.7 A/D converter

This paragraph explains the registers setting method and the notes relevant to the A/D converter.

2.7.1 Memory map

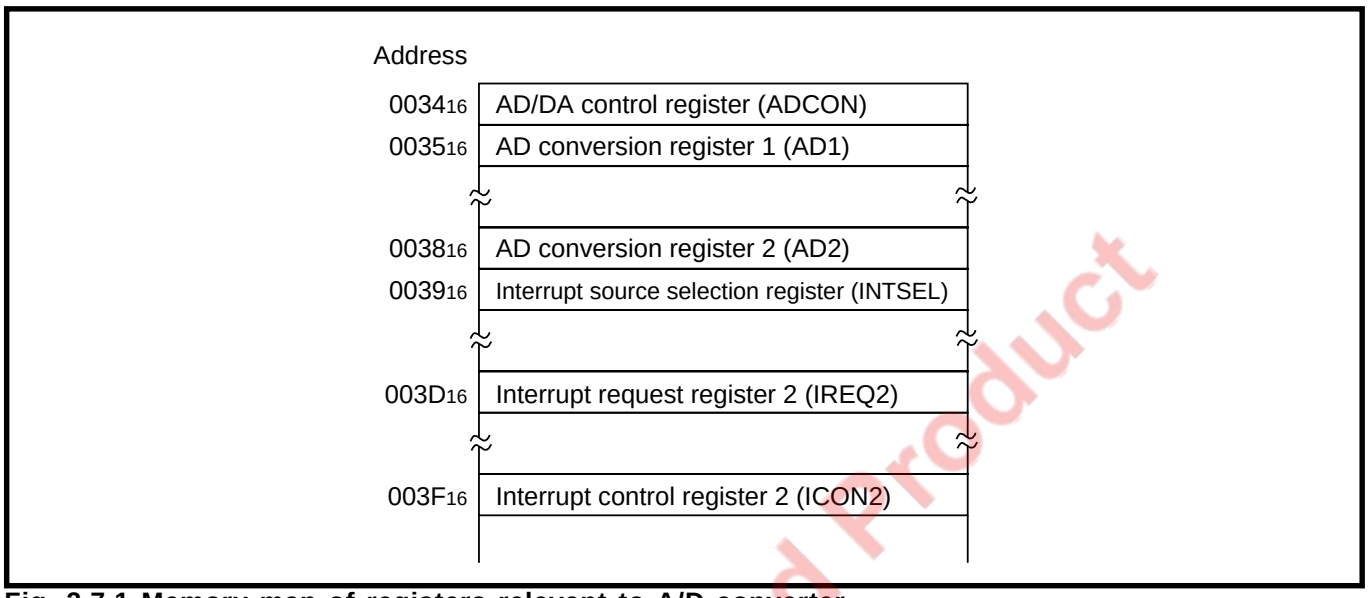


Fig. 2.7.1 Memory map of registers relevant to A/D converter

2.7.2 Relevant registers

AD/DA control register

b7 b6 b5 b4 b3 b2 b1 b0

AD/DA control register
(ADCON: address 0034₁₆)

b	Name	Functions	At reset	R	W
0	Analog input pin selection bits 1	b2 b1 b0 0 0 0: P6₀/AN₀ or P0₀/AN₈ 0 0 1: P6₁/AN₁ or P0₁/AN₉ 0 1 0: P6₂/AN₂ or P0₂/AN₁₀ 0 1 1: P6₃/AN₃ or P0₃/AN₁₁ 1 0 0: P6₄/AN₄ or P0₄/AN₁₂ 1 0 1: P6₅/AN₅ or P0₅/AN₁₃ 1 1 0: P6₆/AN₆ or P0₆/AN₁₄ 1 1 1: P6₇/AN₇ or P0₇/AN₁₅	0	○	○
1			0	○	○
2			0	○	○
3	AD conversion completion bit	0: Conversion in progress 1: Conversion completed	1	○	○
4	Analog input pin selection bit 2	0: AN₀ to AN₇ side 1: AN₈ to AN₁₅ side	0	○	○
5	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
6	DA₁ output enable bit	0: DA₁ output disabled 1: DA₁ output enabled	0	○	○
7	DA₂ output enable bit	0: DA₂ output disabled 1: DA₂ output enabled	0	○	○

Fig. 2.7.2 Structure of AD/DA control register

AD conversion register 1

b7 b6 b5 b4 b3 b2 b1 b0

AD conversion register 1
(AD1: address 0035$_{16}$)

b	Functions	At reset	R	W
0	This is A/D conversion result stored bits. This is read exclusive register. 8-bit read: b7 b6 b5 b4 b3 b2 b1 b0 = b9 b8 b7 b6 b5 b4 b3 b2 10-bit read: b7 b6 b5 b4 b3 b2 b1 b0	Undefined	○	×
1		Undefined	○	×
2		Undefined	○	×
3		Undefined	○	×
4		Undefined	○	×
5		Undefined	○	×
6		Undefined	○	×
7		Undefined	○	×

Fig. 2.7.3 Structure of AD conversion register 1

AD conversion register 2

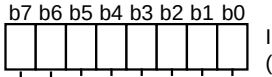
b7 b6 b5 b4 b3 b2 b1 b0

AD conversion register 2
(AD2: address 0038$_{16}$)

b	Functions		At reset	R	W
0	This is A/D conversion result stored bits. This is read exclusive register. 10-bit read: b7 b6 b5 b4 b3 b2 b1 b0 = b9 b8		Undefined	○	×
1			Undefined	○	×
2	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7	Conversion mode selection bit	0: 10-bit A/D mode 1: 8-bit A/D mode	0	○	○

Fig. 2.7.4 Structure of AD conversion register 2

Interrupt source selection register



Interrupt source selection register
(INTSEL: address 0039$_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0/Timer Z interrupt source selection bit (*1)	0: INT_0 interrupt 1: Timer Z interrupt	0	○	○
1	Serial I/O2/Timer Z interrupt source selection bit (*1)	0: Serial I/O2 interrupt 1: Timer Z interrupt	0	○	○
2	Serial I/O1 transmit/ SCL, SDA interrupt source selection bit (*2)	0: Serial I/O1 transmit interrupt 1: SCL, SDA interrupt	0	○	○
3	$CNTR_0$/SCL, SDA interrupt source selection bit (*2)	0: $CNTR_0$ interrupt 1: SCL, SDA interrupt	0	○	○
4	INT_4/$CNTR_2$ interrupt source selection bit	0: INT_4 interrupt 1: $CNTR_2$ interrupt	0	○	○
5	INT_2/I^2C interrupt source selection bit	0: INT_2 interrupt 1: I^2C interrupt	0	○	○
6	$CNTR_1$/Serial I/O3 receive interrupt source selection bit	0: $CNTR_1$ interrupt 1: Serial I/O3 receive interrupt	0	○	○
7	AD converter/Serial I/O3 transmit interrupt source selection bit	0: A/D converter interrupt 1: Serial I/O3 transmit interrupt	0	○	○

*1: Do not write 1 to these bits simultaneously.
*2: Do not write 1 to these bits simultaneously.

Fig. 2.7.5 Structure of Interrupt source selection register

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2
(IREQ2 : address 003D16)

b	Name	Functions	At reset	R	W
0	CNTR0/SCL, SDA interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	CNTR1/Serial I/O3 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O2/Timer Z interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT2/I2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	INT4/CNTR2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	AD converter/Serial I/O3 transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are 0 .		0	○	×

✻: 0 can be set by software, but 1 cannot be set.

Fig. 2.7.6 Structure of Interrupt request register 2

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0
0

Interrupt control register 2
(ICON2 : address 003F16)

b	Name	Functions	At reset	R	W
0	CNTR0/SCL, SDA interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	CNTR1/ Serial I/O3 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/ Timer Z interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT2/I2C interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT3 interrupt enable bit	0 : Interrupt disabled	0	○	○
5	INT4/CNTR2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/Serial I/O3 transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.7.7 Structure of Interrupt control register 2

2.7.3 A/D converter application examples

(1) Conversion of analog input voltage 1

Outline : The analog input voltage input from a sensor is converted to digital values.

Figure 2.7.8 shows a connection diagram, and Figure 2.7.9 shows the relevant registers setting.

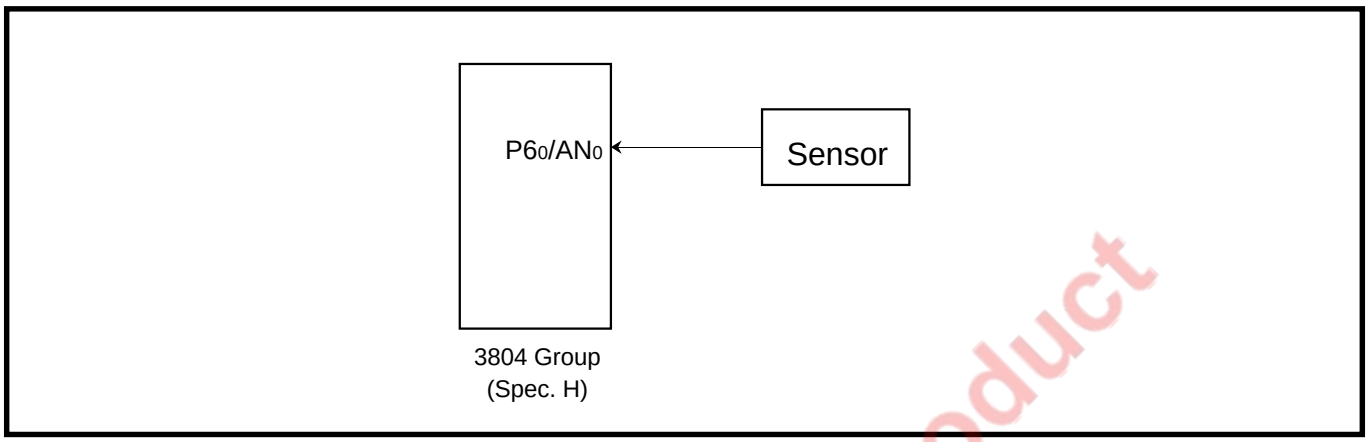


Fig. 2.7.8 Connection diagram

Specifications : •The analog input voltage input from a sensor is converted to digital values.
•P6₀/AN₀ pin is used as an analog input pin.
•10-bit A/D mode

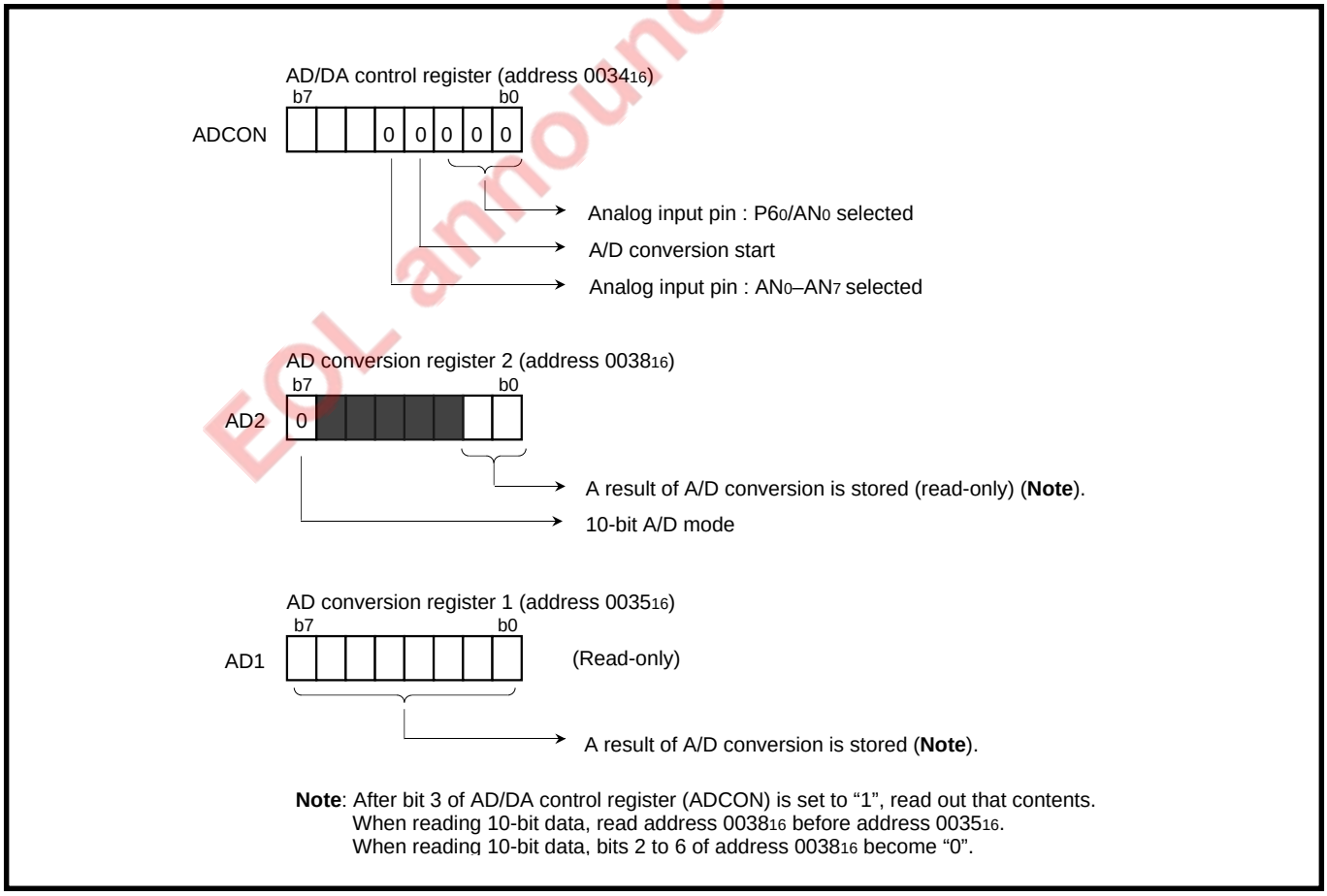


Fig. 2.7.9 Relevant registers setting

An analog input signal from a sensor is converted to the digital value according to the relevant registers setting shown by Figure 2.7.9. Figure 2.7.10 shows the control procedure for 10-bit A/D mode.

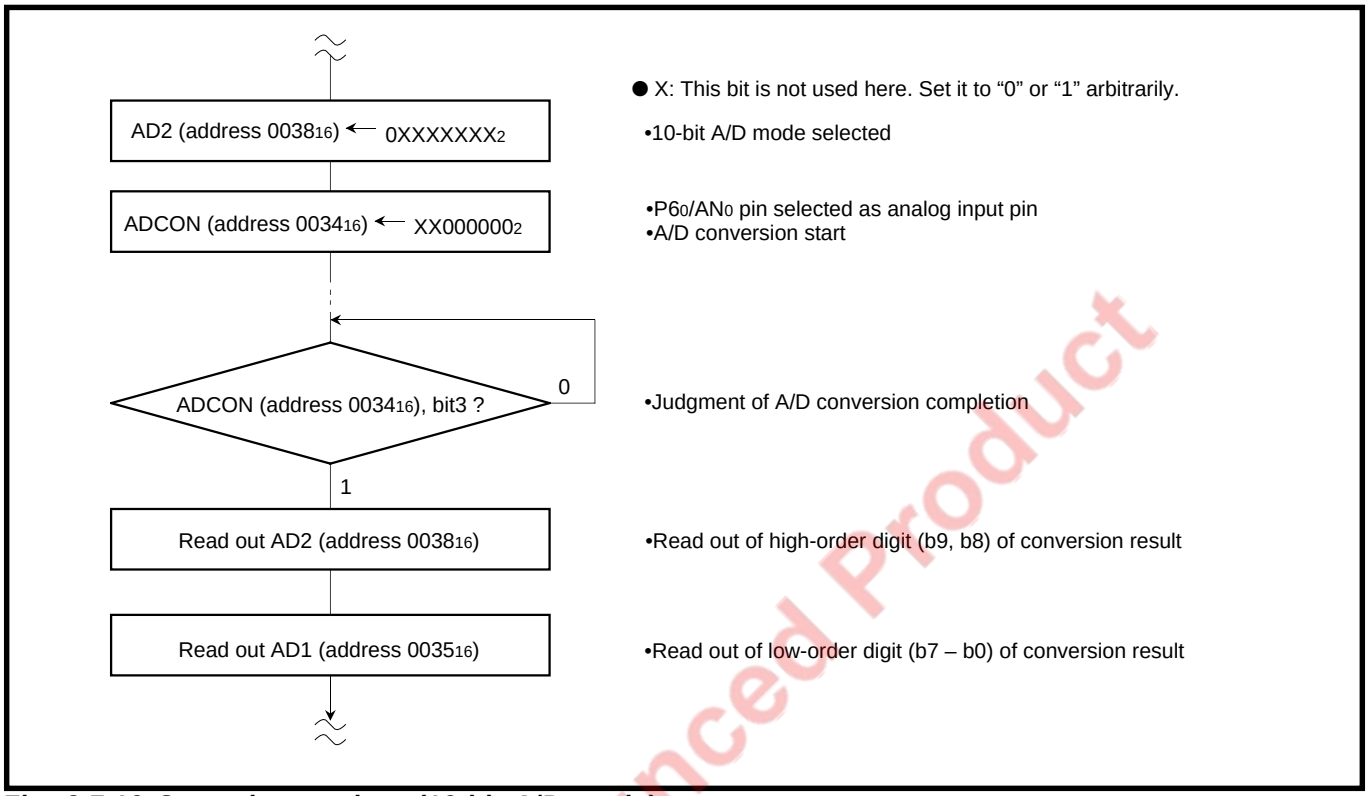


Fig. 2.7.10 Control procedure (10-bit A/D mode)

(2) Conversion of analog input voltage 2

Outline : The analog input voltage input from a sensor is converted to digital values.

Figure 2.7.11 shows a connection diagram, and Figure 2.7.12 shows the relevant registers setting.

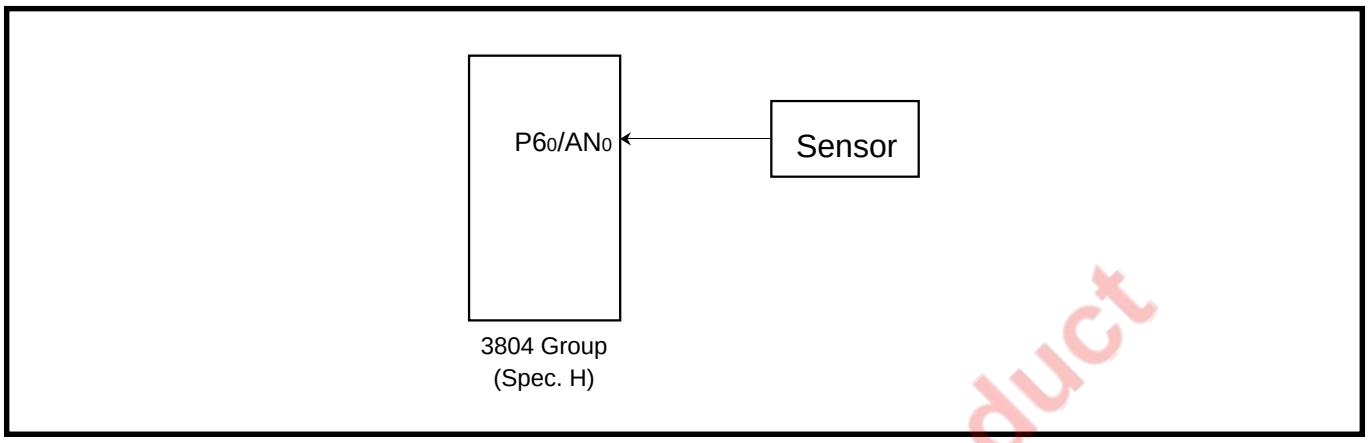
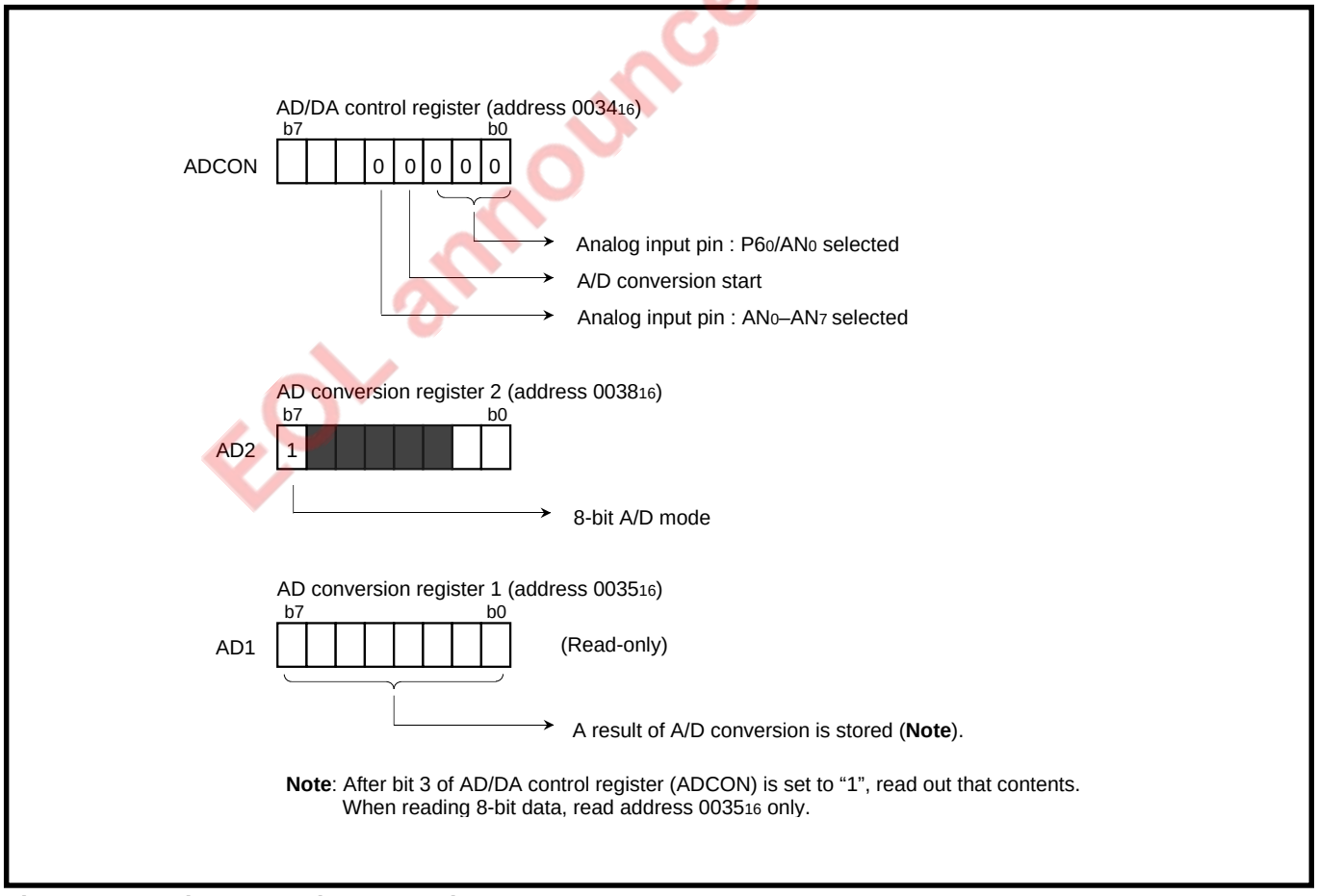


Fig. 2.7.11 Connection diagram

Specifications : •The analog input voltage input from a sensor is converted to digital values.
•P60/AN0 pin is used as an analog input pin.
•8-bit A/D mode



Note: After bit 3 of AD/DA control register (ADCON) is set to "1", read out that contents.
When reading 8-bit data, read address 003516 only.

Fig. 2.7.12 Relevant registers setting

An analog input signal from a sensor is converted to the digital value according to the relevant registers setting shown by Figure 2.7.12. Figure 2.7.13 shows the control procedure for 8-bit A/D mode.

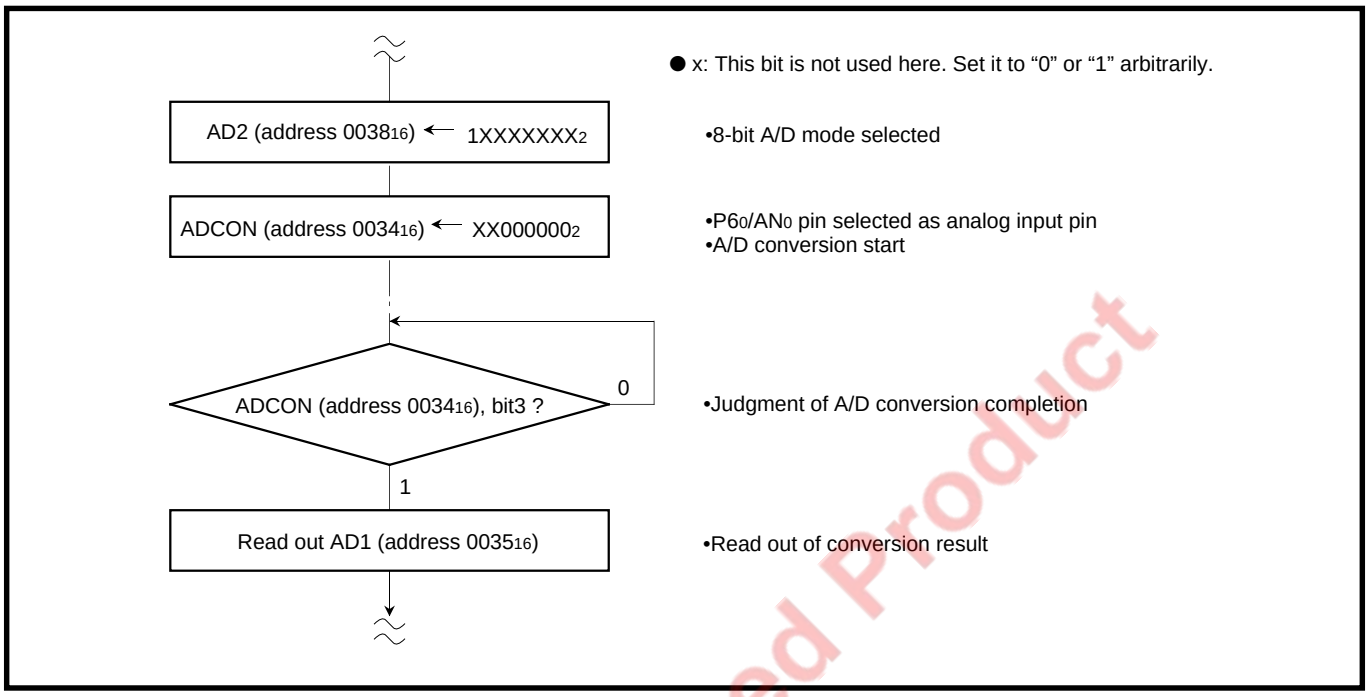


Fig. 2.7.13 Control procedure (8-bit A/D mode)

2.7.4 Notes on A/D converter

(1) Analog input pin

Make the signal source impedance for analog input low, or equip an analog input pin with an external capacitor of 0.01 µF to 1 µF. Further, be sure to verify the operation of application products on the user side.

● **Reason**

An analog input pin includes the capacitor for analog voltage comparison. Accordingly, when signals from signal source with high impedance are input to an analog input pin, charge and discharge noise generates. This may cause the A/D conversion precision to be worse.

(2) A/D converter power source pin

The AVSS pin is A/D converter power source pins. Regardless of using the A/D conversion function or not, connect it as following :

• AVSS : Connect to the VSS line

● **Reason**

If the AVSS pin is opened, the microcomputer may have a failure because of noise or others.

(3) Clock frequency during A/D conversion

The comparator consists of a capacity coupling, and a charge of the capacity will be lost if the clock frequency is too low. Thus, make sure the following during an A/D conversion.

• f(XIN) is 500 kHz or more

• Do not execute the **STP** instruction

(4) Difference between at 8-bit reading in 10-bit A/D mode and at 8-bit A/D mode

At 8-bit reading in the 10-bit A/D mode, "–1/2 LSB" correction is not performed to the A/D conversion result.

In the 8-bit A/D mode, the A/D conversion characteristics is the same as 3802 group's characteristics because "–1/2 LSB" correction is performed.

2.8 D/A Converter

This paragraph explains the registers setting method and the notes relevant to the D/A converter.

2.8.1 Memory map

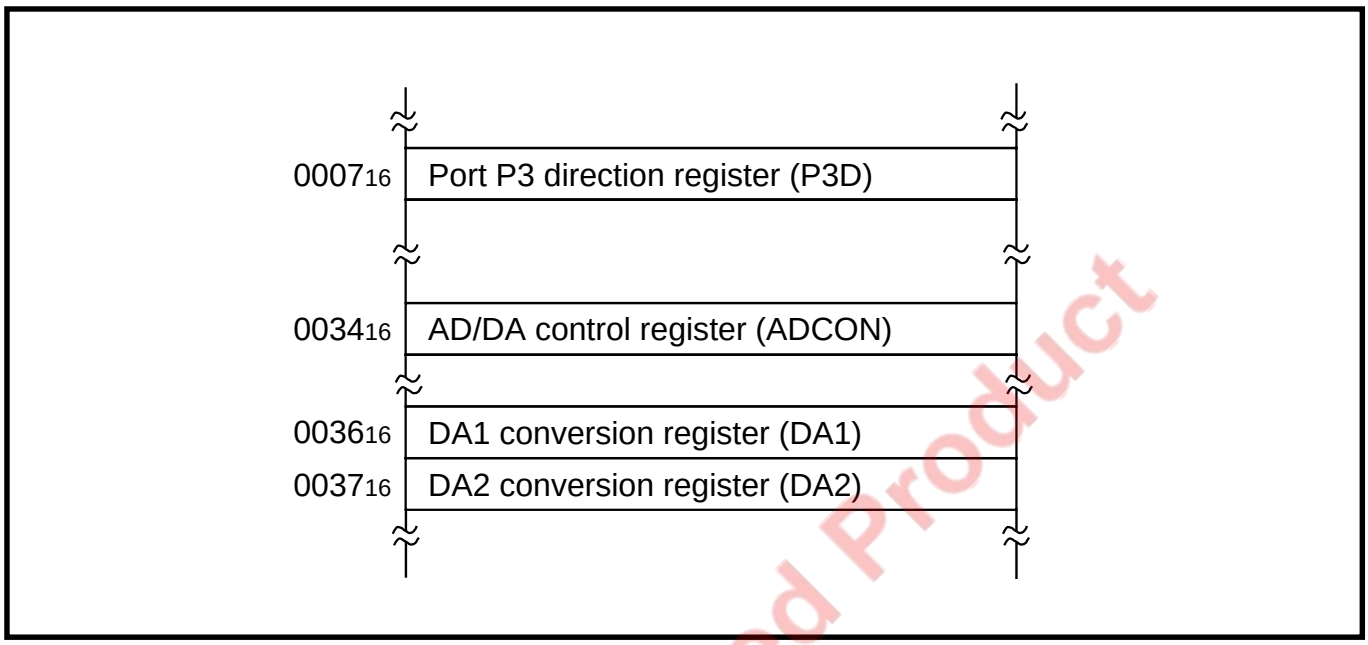


Fig. 2.8.1 Memory map of registers relevant to D/A converter

2.8.2 Relevant registers

Port P3 direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port P3 direction register
(P3D: address 0007_{16})

b	Name	Functions	At reset	R	W
0	Port P3 direction register	0 : Port $P3_0$ input mode 1 : Port $P3_0$ output mode	0	×	○
1		0 : Port $P3_1$ input mode 1 : Port $P3_1$ output mode	0	×	○
2		0 : Port $P3_2$ input mode 1 : Port $P3_2$ output mode	0	×	○
3		0 : Port $P3_3$ input mode 1 : Port $P3_3$ output mode	0	×	○
4		0 : Port $P3_4$ input mode 1 : Port $P3_4$ output mode	0	×	○
5		0 : Port $P3_5$ input mode 1 : Port $P3_5$ output mode	0	×	○
6		0 : Port $P3_6$ input mode 1 : Port $P3_6$ output mode	0	×	○
7		0 : Port $P3_7$ input mode 1 : Port $P3_7$ output mode	0	×	○

Fig. 2.8.2 Structure of Port P5 direction register

AD/DA control register

b7 b6 b5 b4 b3 b2 b1 b0

AD/DA control register
(ADCON: address 0034_{16})

b	Name	Functions	At reset	R	W
0	Analog input pin selection bits 1	b2 b1 b0 0 0 0: $P6_0/AN_0$ or $P0_0/AN_8$ 0 0 1: $P6_1/AN_1$ or $P0_1/AN_9$ 0 1 0: $P6_2/AN_2$ or $P0_2/AN_{10}$ 0 1 1: $P6_3/AN_3$ or $P0_3/AN_{11}$ 1 0 0: $P6_4/AN_4$ or $P0_4/AN_{12}$ 1 0 1: $P6_5/AN_5$ or $P0_5/AN_{13}$ 1 1 0: $P6_6/AN_6$ or $P0_6/AN_{14}$ 1 1 1: $P6_7/AN_7$ or $P0_7/AN_{15}$	0	○	○
1			0	○	○
2			0	○	○
3	AD conversion completion bit	0: Conversion in progress 1: Conversion completed	1	○	○
4	Analog input pin selection bit 2	0: AN_0 to AN_7 side 1: AN_8 to AN_{15} side	0	○	○
5	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are "0".		0	○	×
6	DA_1 output enable bit	0: DA_1 output disabled 1: DA_1 output enabled	0	○	○
7	DA_2 output enable bit	0: DA_2 output disabled 1: DA_2 output enabled	0	○	○

Fig. 2.8.3 Structure of AD/DA control register

DAi conversion register

b7 b6 b5 b4 b3 b2 b1 b0

DAi conversion register (i = 1, 2)
(DAi: addresses 0036_{16}, 0037_{16})

b	Functions	At reset	R	W
0	This is D/A output value stored bits. This is write exclusive register.	0	×	○
1		0	×	○
2		0	×	○
3		0	×	○
4		0	×	○
5		0	×	○
6		0	×	○
7		0	×	○

Fig. 2.8.4 Structure of DAi converter register

2.8.3 D/A converter application example

(1) Speaker output volume modulation

Outline: The volume of a speaker output is modulated by using D/A converter.

Specifications: •Timer X modulates the period of sound for the pitch interval, so that a fixed pitch ("la": approx. 440 Hz) can be output. Modulating the amplitude with the D/A output value controls the volume.
•Use f(X_{IN}) = 6 MHz.
•Use DA1 ($P3_0/DA_1$ pin) as D/A converter.

Figure 2.8.5 shows a peripheral circuit example and Figure 2.8.6 shows a speaker output example. Figure 2.8.7 shows the relevant registers setting.

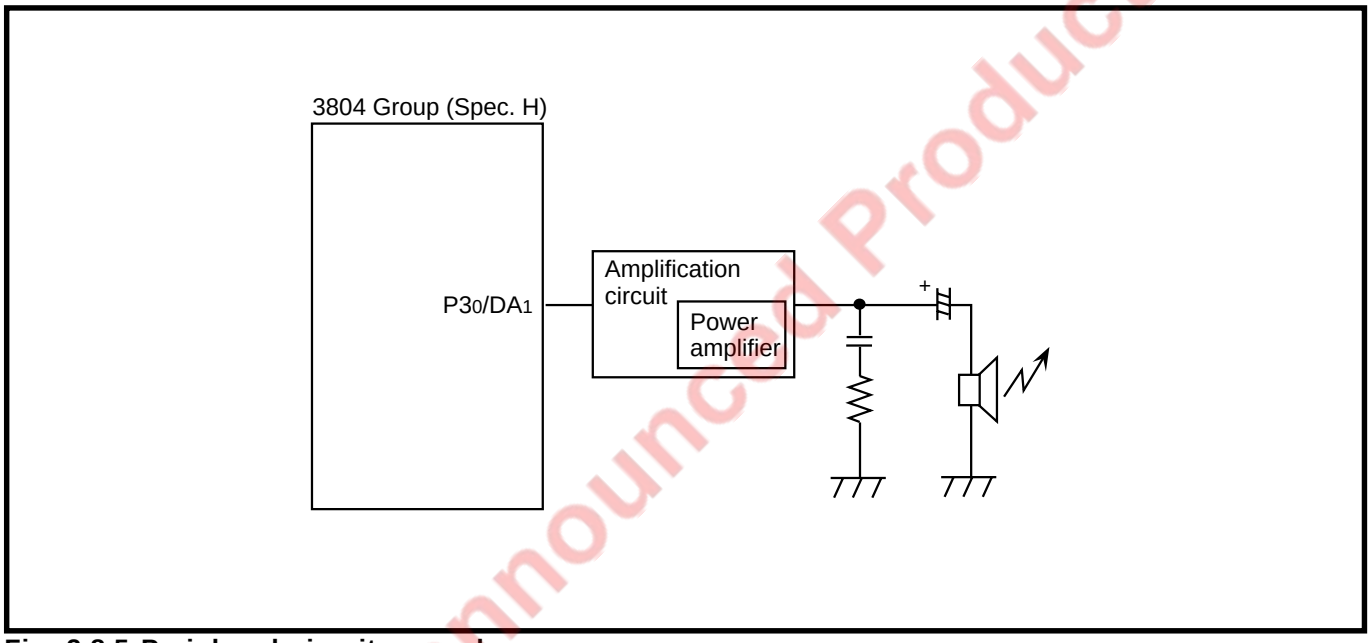


Fig. 2.8.5 Peripheral circuit example

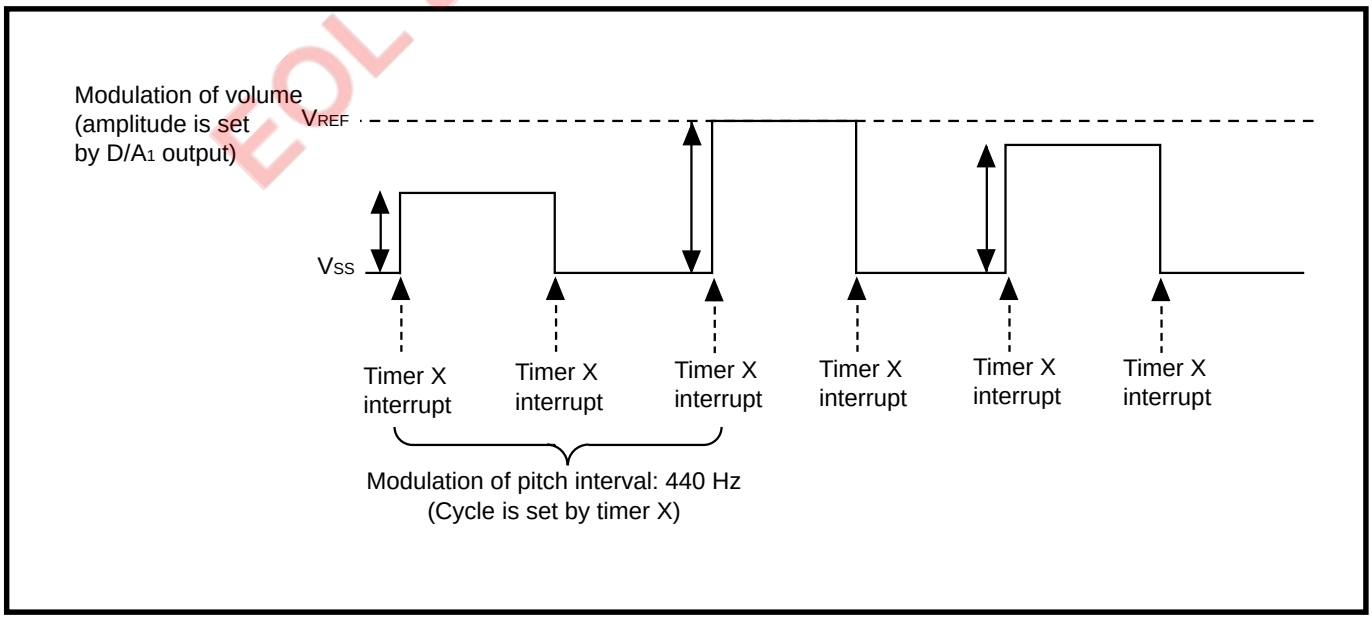


Fig. 2.8.6 Speaker output example

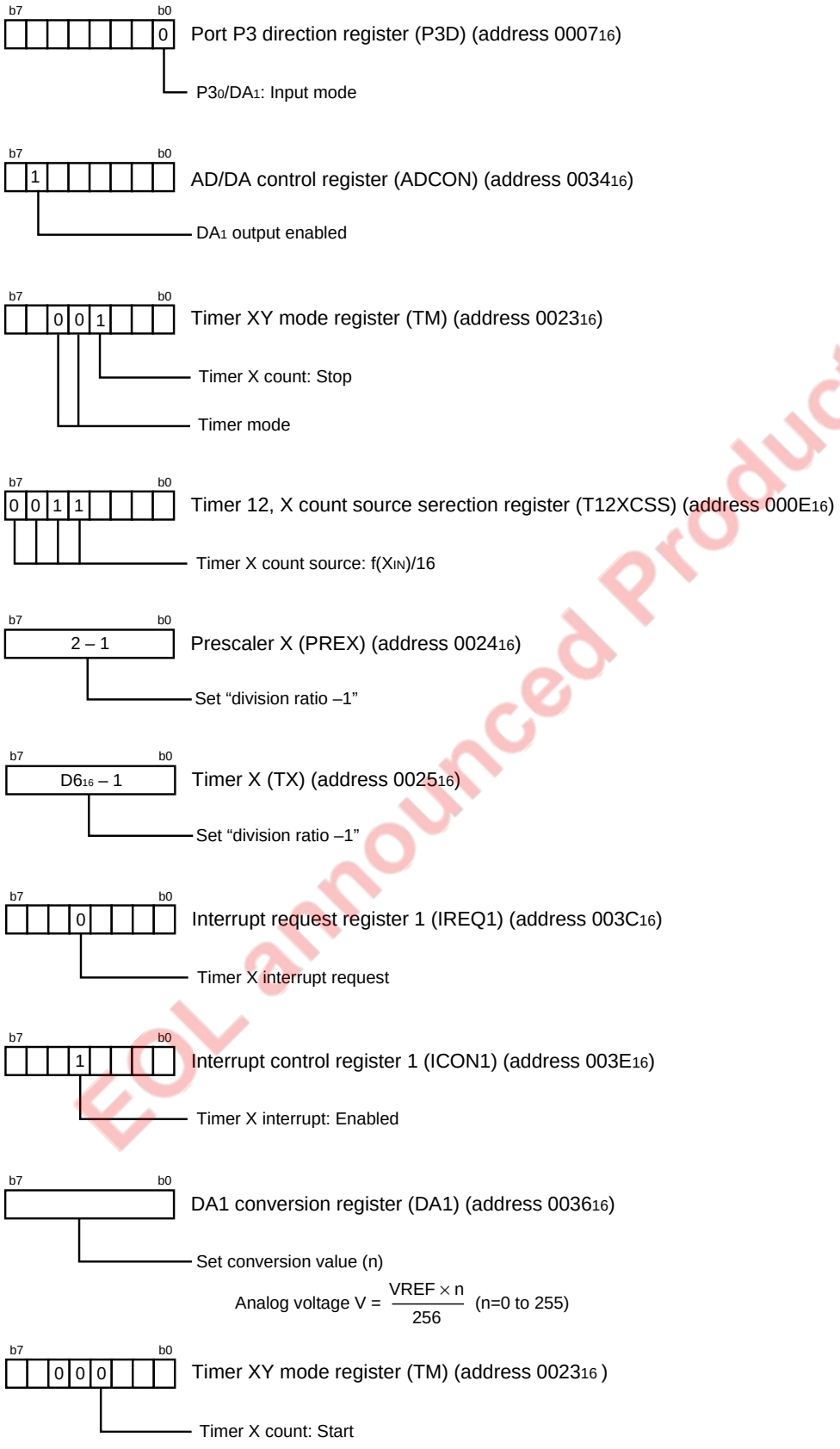


Fig. 2.8.7 Relevant registers setting

When the registers are set as shown in Figure 2.8.7, the speaker output volume is modulated by the D/A output value. Figure 2.8.8 shows the control procedure.

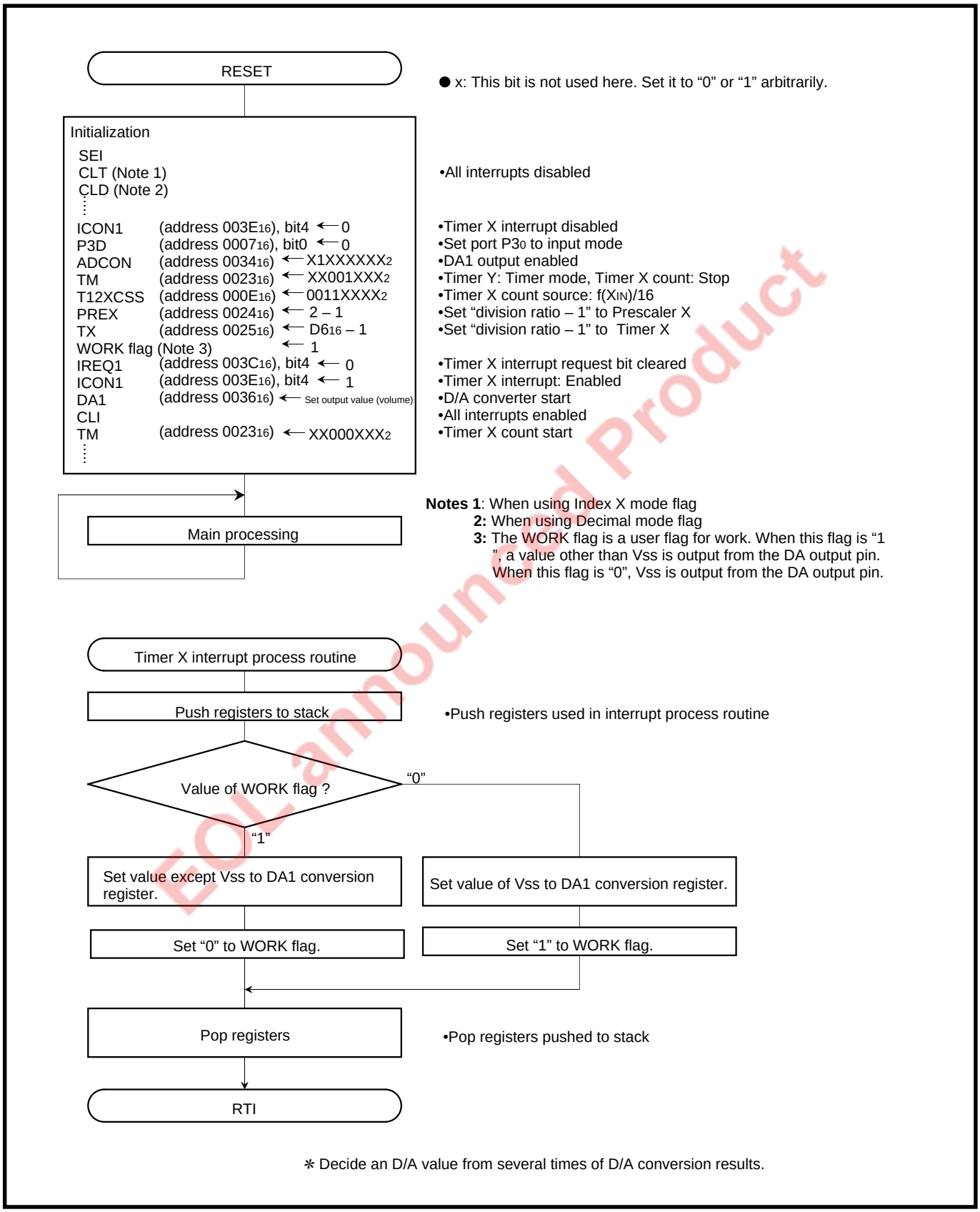


Fig. 2.8.8 Control procedure

2.8.4 Notes on D/A converter

(1) Vcc when using D/A converter

The D/A converter accuracy when Vcc is 4.0 V or less differs from that of when Vcc is 4.0 V or more. When using the D/A converter, we recommend using a Vcc of 4.0 V or more.

(2) DAi conversion register when not using D/A converter

When a D/A converter is not used, set all values of the DAi conversion registers (i = 1, 2) to "00_{16}". The initial value after reset is "00_{16}".

2.9 Watchdog timer

This paragraph explains the registers setting method and the notes relevant to the watchdog timer.

2.9.1 Memory map

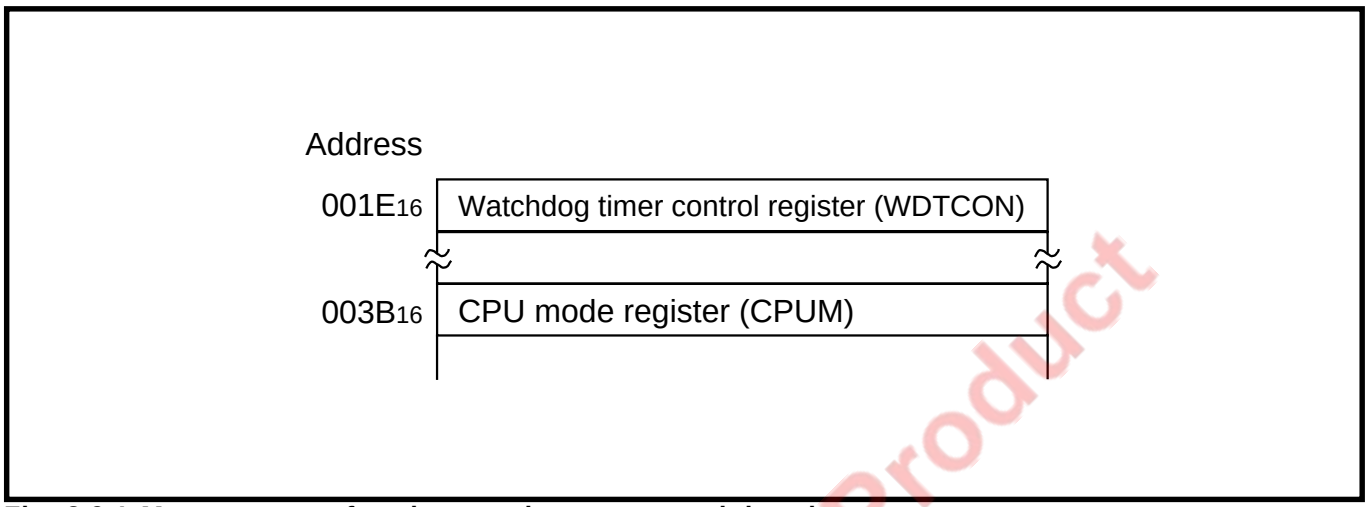


Fig. 2.9.1 Memory map of registers relevant to watchdog timer

2.9.2 Relevant registers

Watchdog timer control register

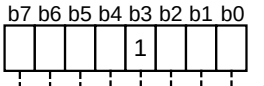
b7 b6 b5 b4 b3 b2 b1 b0

Watchdog timer control register
(WDTCON: address 001E16)

b	Name	Functions	At reset	R	W
0	Watchdog timer H (for read-out of high-order 6 bit)		1	○	×
1			1	○	×
2			1	○	×
3			1	○	×
4			1	○	×
5			1	○	×
6	STP instruction disable bit	0: STP instruction enabled 1: STP instruction disabled	0	○	○
7	Watchdog timer H count source selection bit	0: Watchdog timer L underflow 1: f(XIN)/16 or f(XCIN)/16	0	○	○

Fig. 2.9.2 Structure of Watchdog timer control register

CPU mode register



CPU mode register
(CPUM: address 003B16)

b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 00 : Single-chip mode 01 : 10 : } Not available 11 :	0	○	○
1			0	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to “1”.		1	○	○
4	Port Xc switch bit	0: I/O port function (stop oscillating) 1: XCIN-XCOUT oscillation function	0	○	○
5	Main clock (XIN-XOUT) stop bit	0: Oscillating 1: Stopped	0	○	○
6	Main clock division ratio selection bits	b7 b6 0 0: ϕ=f(XIN)/2 (high-speed mode) 0 1: ϕ=f(XIN)/8 (middle-speed mode) 1 0: ϕ=f(XCIN)/2 (low-speed mode) 1 1: Not available	1	○	○
7			0	○	○

Fig. 2.9.3 Structure of CPU mode register

2.9.3 Watchdog timer application examples

(1) Detection of program runaway

Outline: If program runaway occurs, let the microcomputer reset, using the internal timer for detection of program runaway.

Specifications:
- High-speed mode is used as a main clock division ratio.
- An underflow signal of the watchdog timer L is supplied as the count source of watchdog timer H.
- 1 cycle of main routine is 65.536 ms or less.
- Before the watchdog timer H underflows, “0” is set into bit 7 of the watchdog timer control register at every cycle in a main routine.
- An underflow of watchdog timer H is judged to be program runaway, and the microcomputer is returned to the reset status.

Figure 2.9.4 shows a watchdog timer connection and division ratio setting; Figure 2.9.5 shows the relevant registers setting; Figure 2.9.6 shows the control procedure.

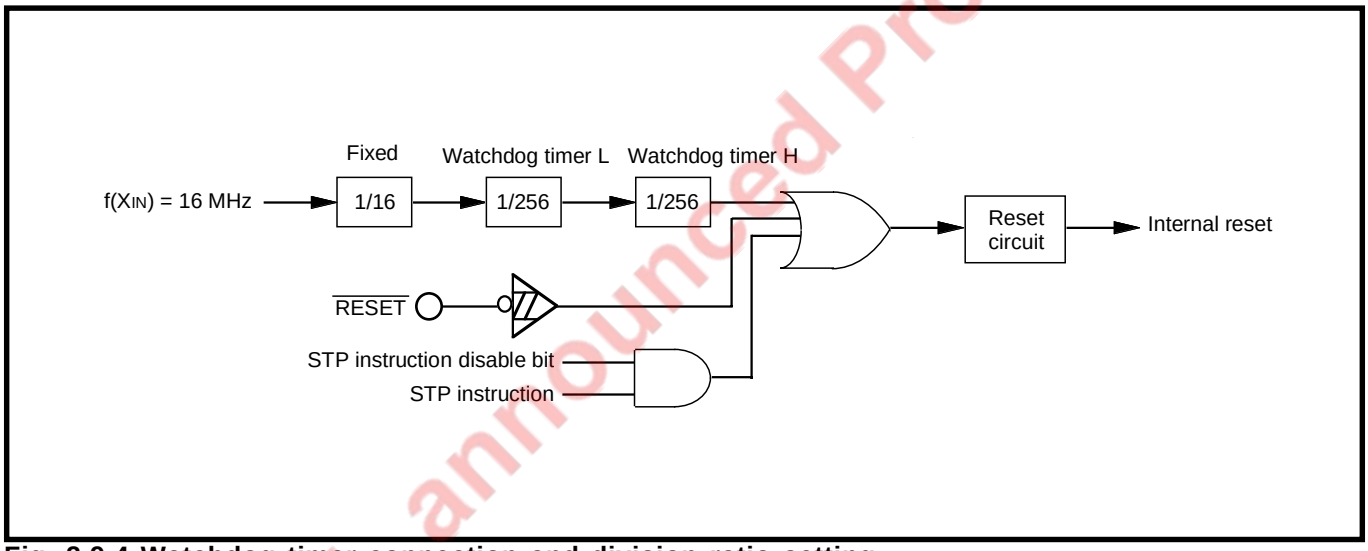


Fig. 2.9.4 Watchdog timer connection and division ratio setting

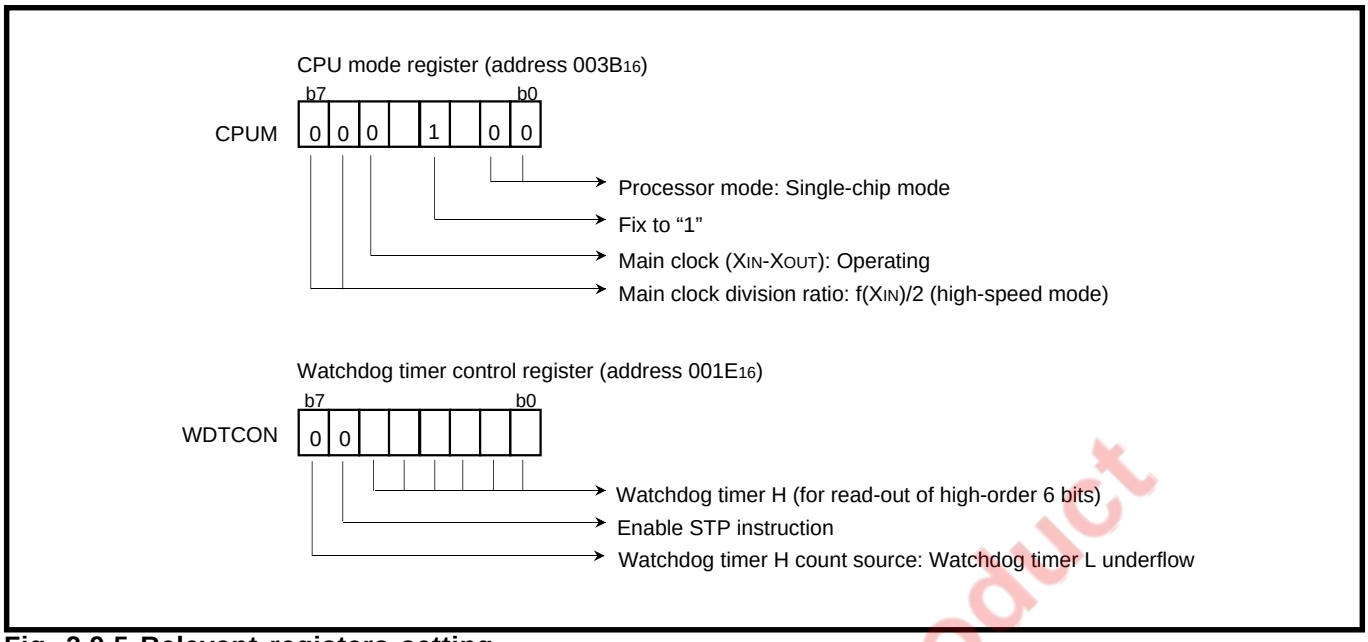


Fig. 2.9.5 Relevant registers setting

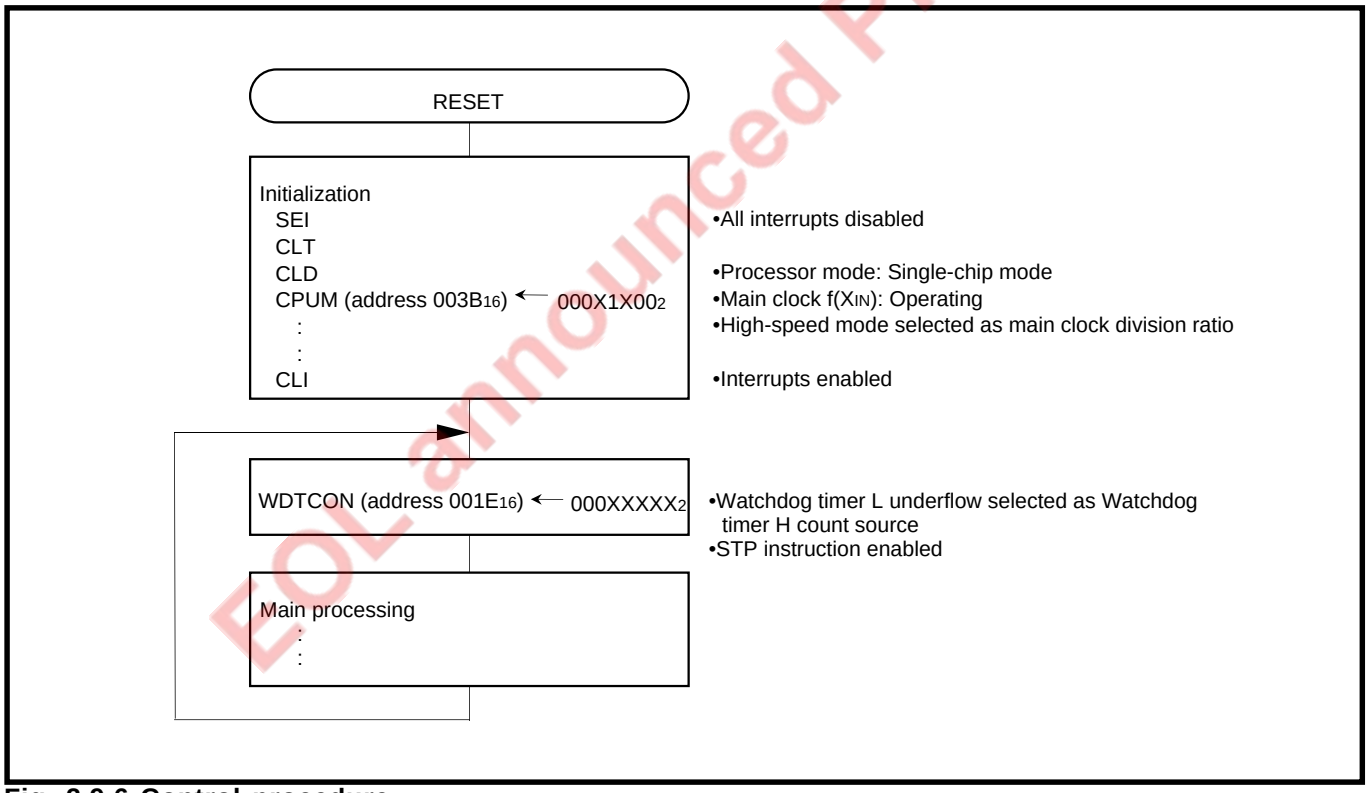


Fig. 2.9.6 Control procedure

2.9.4 Notes on watchdog timer

- Make sure that the watchdog timer H does not underflow while waiting Stop release, because the watchdog timer keeps counting during that term.
- When the STP instruction disable bit has been set to "1", it is impossible to switch it to "0" by a program.

2.10 Reset

2.10.1 Connection example of reset IC

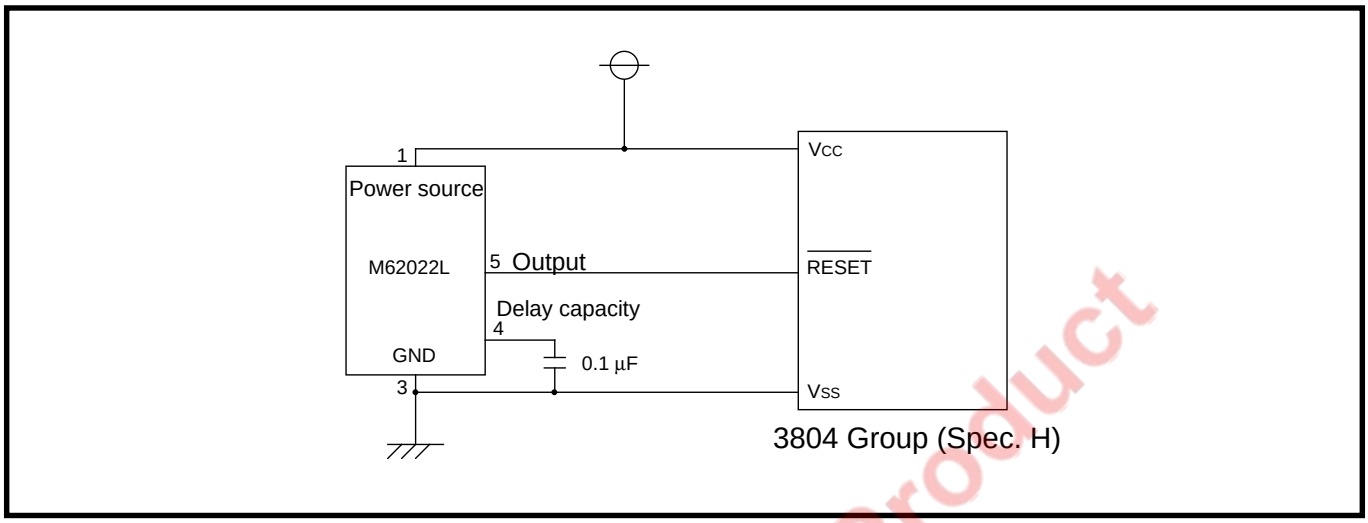


Fig. 2.10.1 Example of poweron reset circuit

Figure 2.10.2 shows the system example which switches to the RAM backup mode by detecting a drop of the system power source voltage with the INT interrupt.

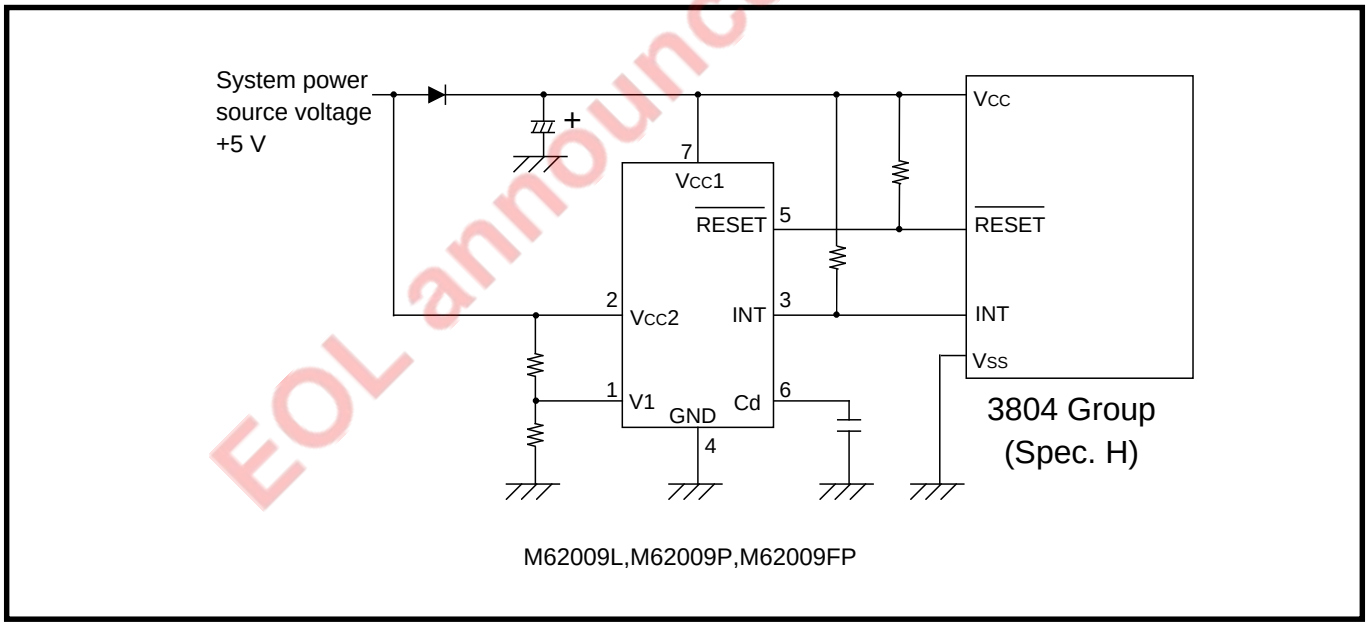


Fig. 2.10.2 RAM backup system

2.10.2 Notes on $\overline{\text{RESET}}$ pin

Connecting capacitor

In case where the $\overline{\text{RESET}}$ signal rise time is long, connect a ceramic capacitor or others across the $\overline{\text{RESET}}$ pin and the Vss pin. Use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, note the following :

- Make the length of the wiring which is connected to a capacitor as short as possible.
- Be sure to verify the operation of application products on the user side.

● **Reason**

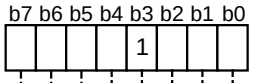
If the several nanosecond or several ten nanosecond impulse noise enters the $\overline{\text{RESET}}$ pin, it may cause a microcomputer failure.

2.11 Clock generating circuit

This paragraph explains how to set the registers relevant to the clock generating circuit and describes an application example.

2.11.1 Relevant registers

CPU mode register



CPU mode register
(CPUM: address 003B16)

b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 00 : Single-chip mode 01 : Not available 10 : Not available 11 : Not available	0	○	○
1			0	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to "1".		1	○	○
4	Port Xc switch bit	0: I/O port function (oscillation stopped) 1: XCIN-XCOUT oscillation function	0	○	○
5	Main clock (XIN-XOUT) stop bit	0: Oscillating 1: Stopped	0	○	○
6	Main clock division ratio selection bits	b7 b6 0 0: ϕ=f(XIN)/2 (high-speed mode) 0 1: ϕ=f(XIN)/8 (middle-speed mode) 1 0: ϕ=f(XCIN)/2 (low-speed mode) 1 1: Not available	1	○	○
7			0	○	○

Fig. 2.11.1 Structure of CPU mode register

2.11.2 Clock generating circuit application example

(1) Status transition during power failure

Outline: The clock counts up every second by using the timer interrupt during a power failure.

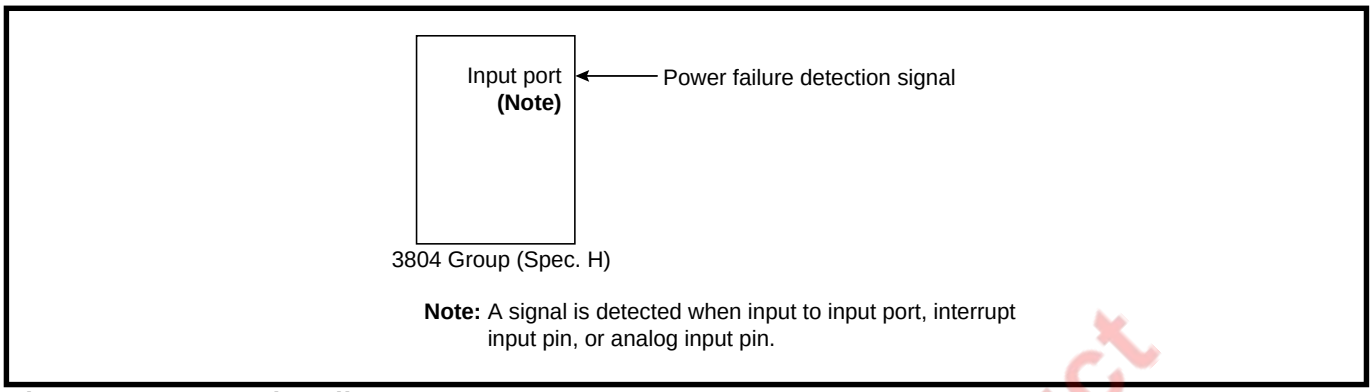


Fig. 2.11.2 Connection diagram

Specifications: •Reducing power dissipation as low as possible while maintaining clock function
•Clock: f(X_{IN}) = 8 MHz, f(X_{CIN}) = 32.768 kHz
•Port processing
Input port: Fixed to "H" or "L" level externally.
Output port: Fixed to output level that does not cause current flow to the external.
(Example) Fix to "H" for an LED circuit that turns on at "L" output level.
I/O port: Input port → Fixed to "H" or "L" level externally.
Output port → Output of data that does not consume current
V_{REF} pin: Terminate A/D conversion operation
Stop V_{REF} current dissipation by setting value of DAi conversion register to "00_{16}".

Figure 2.11.3 shows the status transition diagram during power failure and Figure 2.11.4 shows the setting of relevant registers.

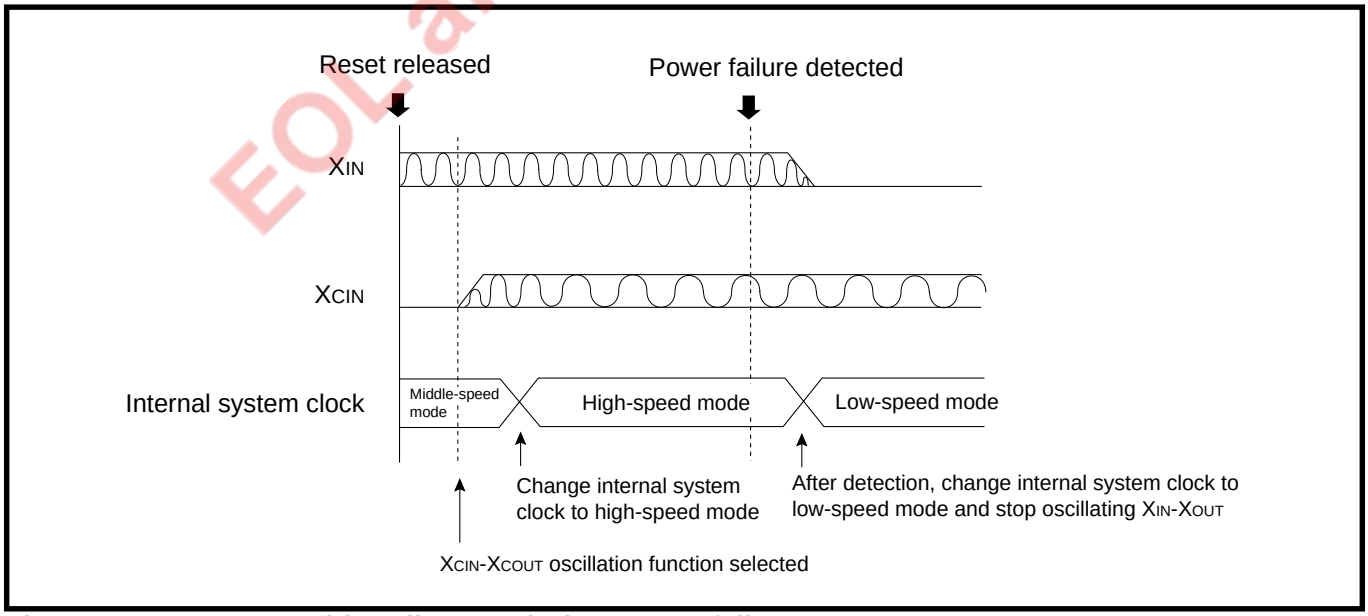
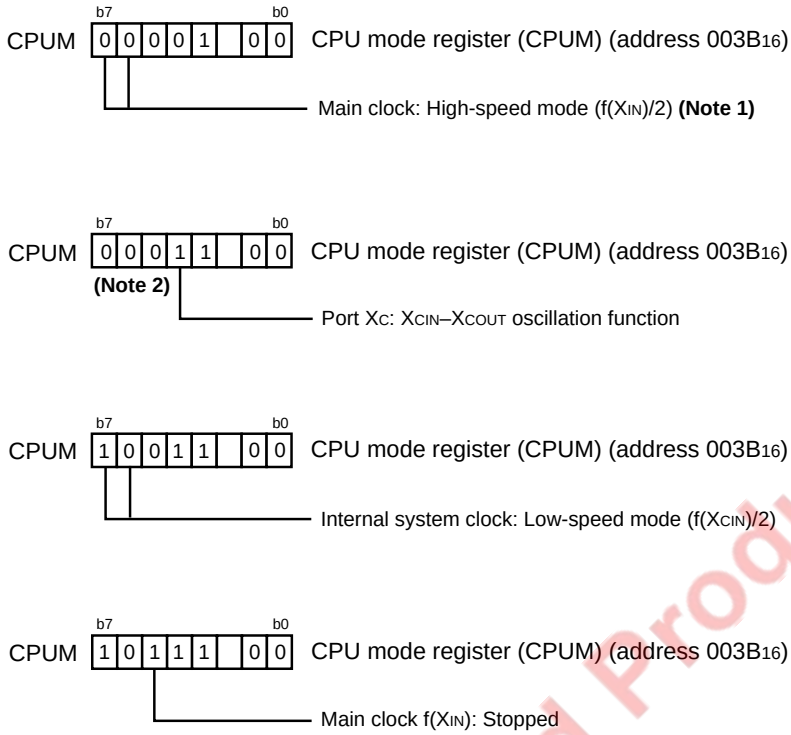


Fig. 2.11.3 Status transition diagram during power failure



Notes 1: This setting is necessary only when selecting the high-speed mode.
2: When selecting the middle-speed mode, bit 6 is "1".

Fig. 2.11.4 Setting of relevant registers

Control procedure: To prepare for a power failure, set the relevant registers in the order shown below.

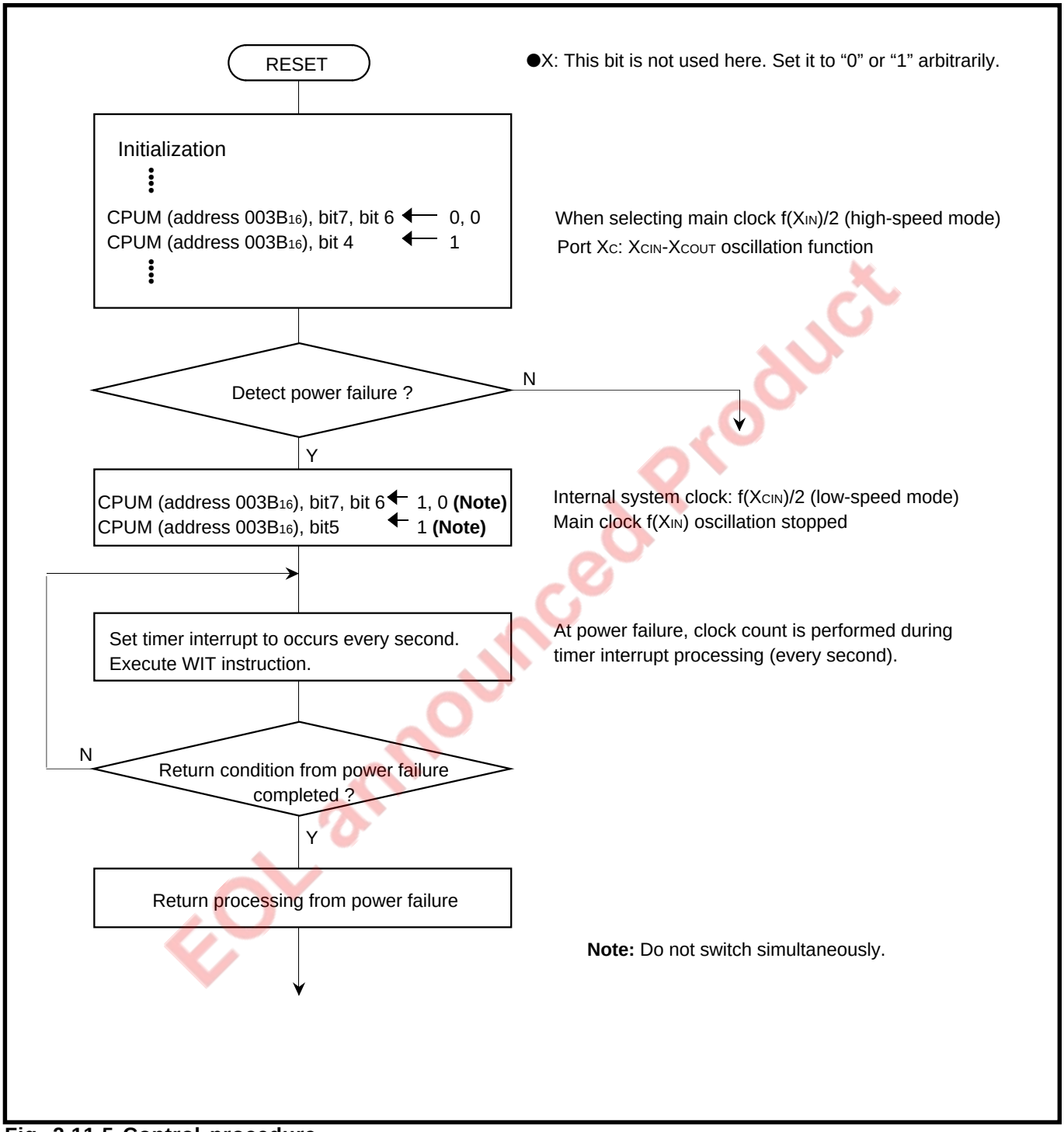


Fig. 2.11.5 Control procedure

2.12 Standby function

The 3804 group (Spec. H) is provided with standby functions to stop the CPU by software and put the CPU into the low-power operation.
The following two types of standby functions are available.
•Stop mode using STP instruction
•Wait mode using WIT instruction

2.12.1 Stop mode

The stop mode is set by executing the STP instruction. In the stop mode, the oscillation of both clocks (X_{IN}–X_{OUT}, X_{CIN}–X_{COUT}) stop and the internal clock ϕ stops at the "H" level. The CPU stops and peripheral units stop operating. As a result, power dissipation is reduced.

(1) State in stop mode

Table 2.12.1 shows the state in the stop mode.

Table 2.12.1 State in stop mode

Item	State in stop mode
Oscillation	Stopped.
CPU	Stopped.
Internal clock ϕ	Stopped at "H" level.
I/O ports P0–P6	Retains the state at the STP instruction execution.
Timer	Stopped. (Timers 1, 2, X, Y, Z) However, Timers X, Y, Z can be operated in the event counter mode.
PWM	Stopped.
Watchdog timer	Stopped.
Serial I/O1, Serial I/O2, Serial I/O3	Stopped. However, these can be operated only when an external clock is selected.
I^2C-BUS interface	Stopped.
A/D converter	Stopped.
D/A converter	Retains output voltage.

(2) Release of stop mode

The stop mode is released by a reset input or by the occurrence of an interrupt request. Note the differences in the restoration process according to reset input or interrupt request, as described below.

■Restoration by reset input

The stop mode is released by holding the $\overline{\text{RESET}}$ pin to the "L" input level during the stop mode. Oscillation is started when all ports are in the input state and the stop mode of the main clock (X_{IN}-X_{OUT}) is released.

Oscillation is unstable when restarted. For this reason, time for stabilizing of oscillation (oscillation stabilizing time) is required. The input of the $\overline{\text{RESET}}$ pin should be held at the "L" level until oscillation stabilizes.

When the $\overline{\text{RESET}}$ pin is held at the "L" level for 16 cycles or more of X_{IN} after the oscillation has stabilized, the microcomputer will go to the reset state. After the input level of the $\overline{\text{RESET}}$ pin is returned to "H", the reset state is released in approximately 10.5 to 18.5 cycles of the X_{IN} input.

Figure 2.12.1 shows the oscillation stabilizing time at restoration by reset input.

At release of the stop mode by reset input, the internal RAM retains its contents previous to the reset. However, the previous contents of the CPU register and SFR are not retained.

For more details concerning reset, refer to "2.10 Reset".

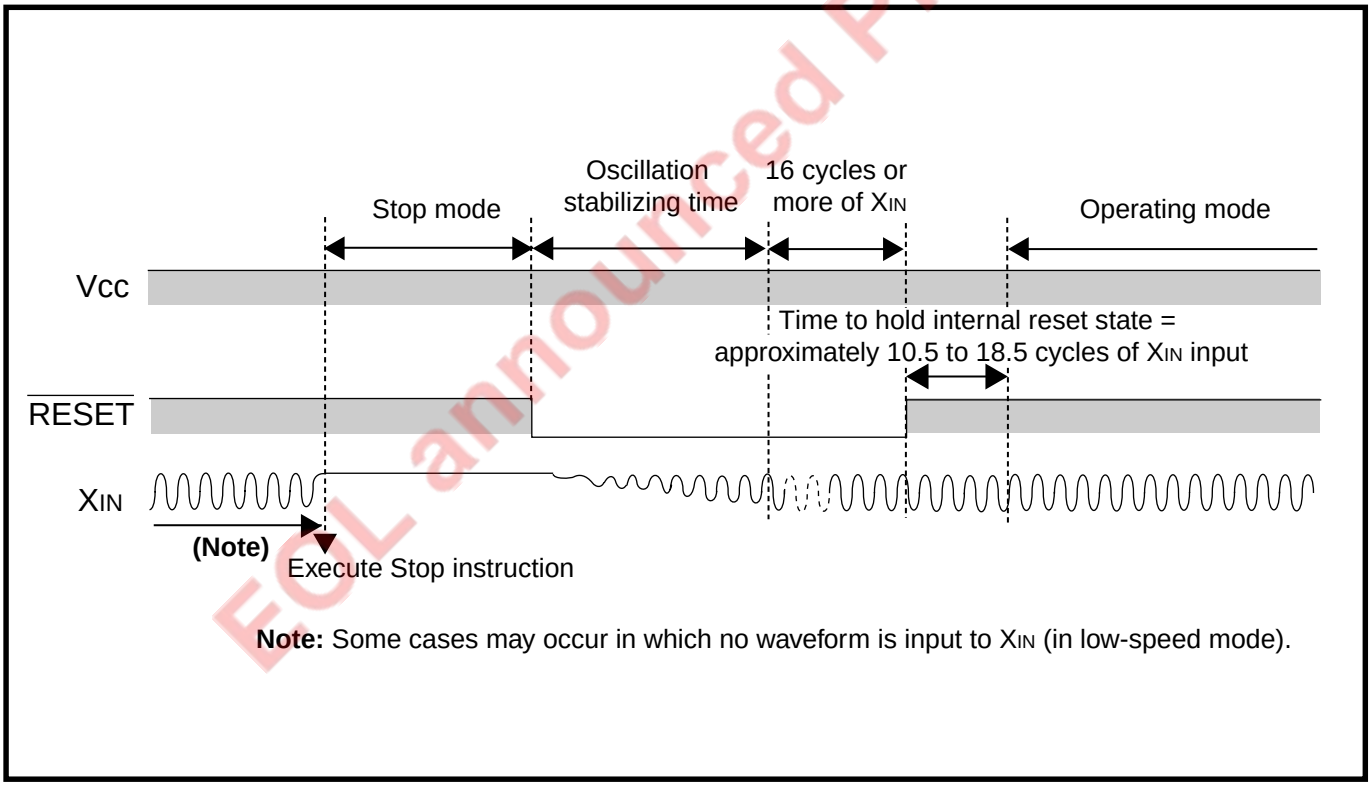


Fig. 2.12.1 Oscillation stabilizing time at restoration by reset input

■Restoration by interrupt request

The occurrence of an interrupt request in the stop mode releases the stop mode. As a result, oscillation is resumed. The interrupts available for restoration are:

•INT_0–INT_4
•$CNTR_0$, $CNTR_1$, $CNTR_2$
•Serial I/O (1, 2, 3) using an external clock
•Timer X, Y, Z using an external event count
•SCL/SDA

However, when using any of these interrupt requests for restoration from the stop mode, in order to enable the selected interrupt, you must execute the STP instruction after setting the following conditions.

[Necessary register setting]

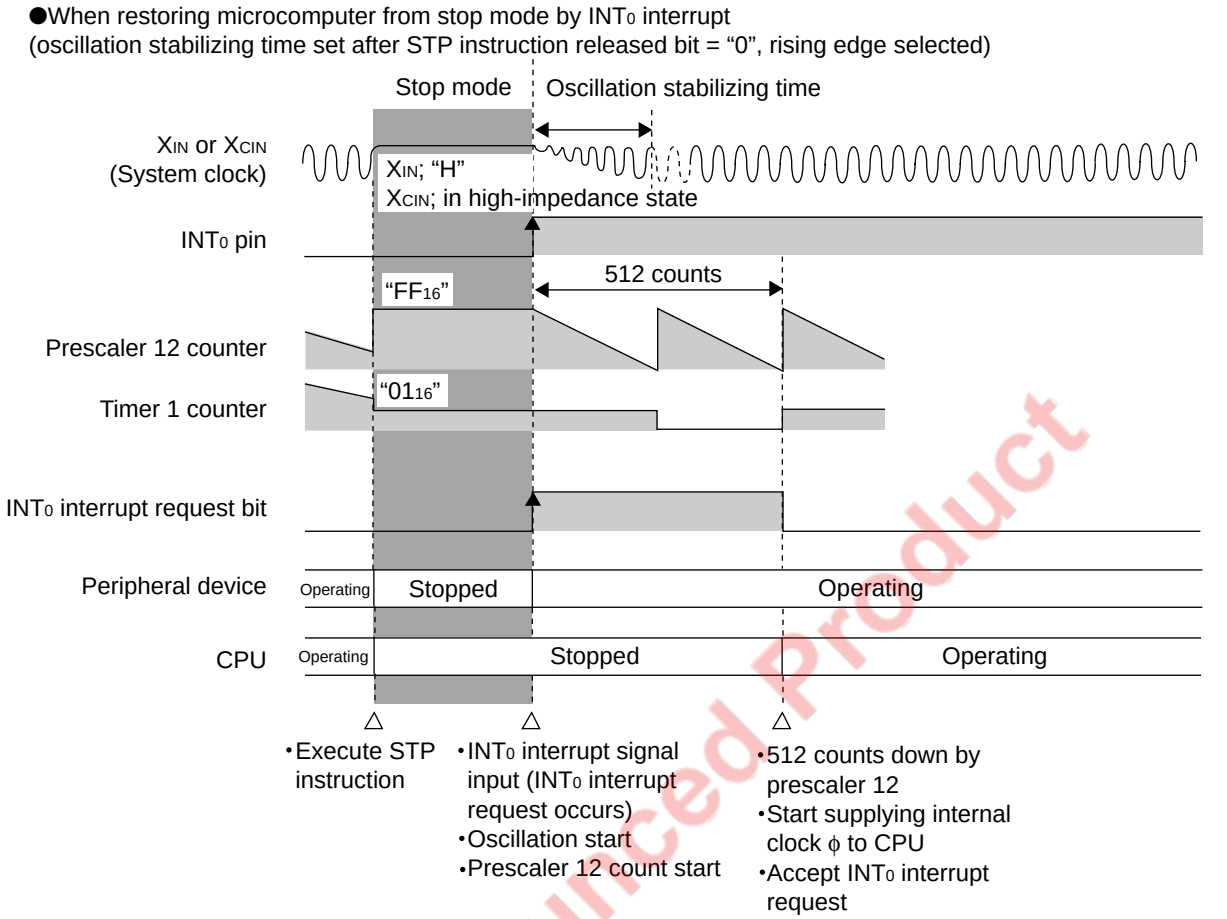
① Interrupt disable flag I = "0" (interrupt enabled)
② Timer 1 interrupt enable bit = "0" (interrupt disabled)
③ Interrupt request bit of interrupt source to be used for restoration = "0" (no interrupt request issued)
④ Interrupt enable bit of interrupt source to be used for restoration = "1" (interrupts enabled)

For more details concerning interrupts, refer to "2.2 Interrupts".

Oscillation is unstable when restarted. For this reason, time for stabilizing of oscillation (oscillation stabilizing time) is required. For restoration by an interrupt request, waiting time prior to supplying internal clock ϕ to the CPU is automatically generated*2 by Prescaler 12 and Timer 1*1. This waiting time is reserved as the oscillation stabilizing time on the system clock side. The supply of internal clock ϕ to the CPU is started at the Timer 1 underflow.

Figure 2.12.2 shows an execution sequence example at restoration by the occurrence of an INT_0 interrupt request.

∗1: If the STP instruction is executed when the oscillation stabilizing time set after STP instruction released bit is "0", "FF_{16}" and "01_{16}" are automatically set in the Prescaler 12 counter/latch and Timer 1 counter/latch, respectively. When the oscillation stabilizing time set after STP instruction released bit is "1", nothing is automatically set to either Prescaler 12 or Timer 1. For this reason, any suitable value can be set to Prescaler 12 and Timer 1 for the oscillation stabilizing time.
∗2: Immediately after the oscillation is started, the count source is supplied to the prescaler 12 so that a count operation is started.



Note: The count source set at STP instruction execution is connected as the prescaler 12 count source.

Fig. 2.12.2 Execution sequence example at restoration by occurrence of INT0 interrupt request

(3) Notes on using stop mode

■Register setting

Since values of the prescaler 12 and Timer 1 are automatically reloaded when returning from the stop mode, set them again, respectively. (When the oscillation stabilizing time set after STP instruction released bit is "0")

■Clock restoration

After restoration from the stop mode to the normal mode by an interrupt request, the contents of the CPU mode register previous to the STP instruction execution are retained. Accordingly, if both main clock and sub clock were oscillating before execution of the STP instruction, the oscillation of both clocks is resumed at restoration.

In the above case, when the main clock side is set as a system clock, the oscillation stabilizing time until the timer 1 underflow is reserved at restoration from the stop mode.

When the oscillation stabilizing time set after STP instruction released bit is "0", the time for 512 counts of the count source become the oscillation stabilizing time. When the oscillation stabilizing time set after STP instruction released bit is "1", an arbitrarily count value set to the prescaler 12 and the timer 1 become the oscillation stabilizing time.

At this time, note that the oscillation on the sub clock side may not be stabilized even after the lapse of the oscillation stabilizing time of the main clock side.

2.12.2 Wait mode

The wait mode is set by execution of the WIT instruction. In the wait mode, oscillation continues, but the internal clock ϕ stops at the "H" level.
The CPU stops, but most of the peripheral units continue operating.

(1) State in wait mode

The continuation of oscillation permits clock supply to the peripheral units. Table 2.12.2 shows the state in the wait mode.

Table 2.12.2 State in wait mode

Item	State in wait mode
Oscillation	Operating.
CPU	Stopped.
Internal clock ϕ	Stopped at "H" level.
I/O ports P0–P6	Retains the state at the WIT instruction execution.
Timer	Operating.
PWM	Operating.
Watchdog timer	Operating.
Serial I/O1, Serial I/O2, Serial I/O3	Operating.
I^2C-BUS interface	Stopped.
A/D converter	Operating.
D/A converter	Retains output voltage.

(2) Release of wait mode

The wait mode is released by reset input or by the occurrence of an interrupt request. Note the differences in the restoration process according to reset input or interrupt request, as described below.

In the wait mode, oscillation is continued, so an instruction can be executed immediately after the wait mode is released.

■Restoration by reset input

The wait mode is released by holding the input level of the $\overline{\text{RESET}}$ pin at "L" in the wait mode. Upon release of the wait mode, all ports are in the input state, and supply of the internal clock ϕ to the CPU is started. To reset the microcomputer, the $\overline{\text{RESET}}$ pin should be held at an "L" level for 16 cycles or more of X_{IN}. The reset state is released in approximately 10.5 cycles to 18.5 cycles of the X_{IN} input after the input of the $\overline{\text{RESET}}$ pin is returned to the "H" level.

At release of wait mode, the internal RAM retains its contents previous to the reset. However, the previous contents of the CPU register and SFR are not retained.

Figure 2.12.3 shows the reset input time.

For more details concerning reset, refer to "2.10 Reset".

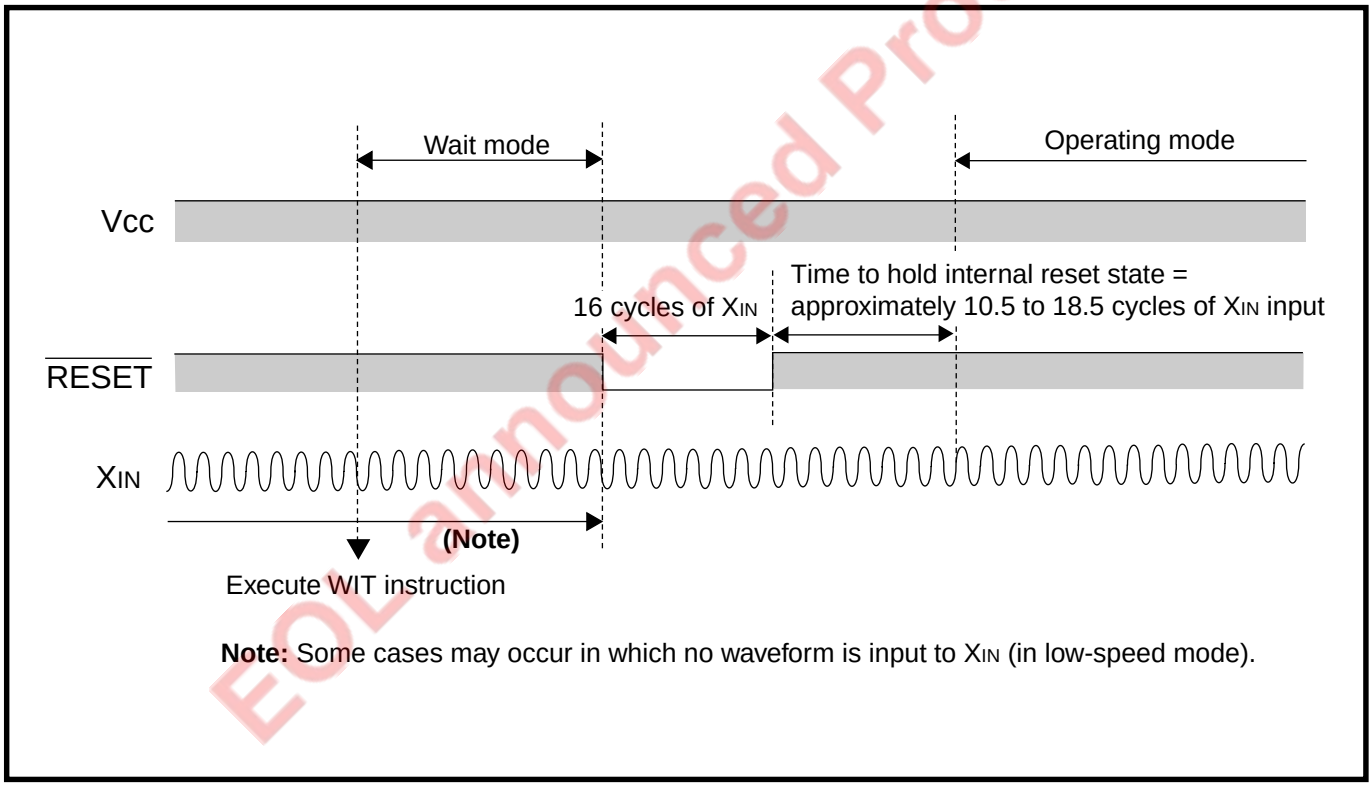


Fig. 2.12.3 Reset input time

■Restoration by interrupt request

In the wait mode, the occurrence of an interrupt request releases the wait mode and supply of the internal clock ϕ to the CPU is started. At the same time, the interrupt request used for restoration is accepted, so the interrupt processing routine is executed.

However, when using an interrupt request for restoration from the wait mode, in order to enable the selected interrupt, you must execute the STP instruction after setting the following conditions.

[Necessary register setting]

① Interrupt disable flag I = "0" (interrupt enabled)

② Interrupt request bit of interrupt source to be used for restoration = "0" (no interrupt request issued)

③ Interrupt enable bit of interrupt source to be used for restoration = "1" (interrupts enabled)

For more details concerning interrupts, refer to "2.2 Interrupts".

(3) Notes on wait mode

■Clock restoration

If the wait mode is released by a reset when X_{CIN} is set as the system clock and X_{IN} oscillation is stopped during execution of the WIT instruction, X_{CIN} oscillation stops, X_{IN} oscillations starts, and X_{IN} is set as the system clock.

In the above case, the $\overline{\text{RESET}}$ pin should be held at "L" until the oscillation is stabilized.

2.13 Flash memory mode

This paragraph explains the registers setting method and the notes relevant to the flash memory mode of M38049FFHSP/FP/HP/KP.

2.13.1 Overview

The functions of the flash memory version are similar to those of the mask ROM version except that the flash memory is built-in and some of the SFR area differ from that of the mask ROM version (refer to "2.13.2 Memory map").

In the flash memory version, the built-in flash memory can be programmed or erased by using the following three modes.

- CPU rewrite mode
- Parallel I/O mode
- Standard serial I/O mode

2.13.2 Memory map

M38049FFHSP/FP/HP/KP have 60 Kbytes of built-in flash memory.

Figure 2.13.1 shows the memory map of the flash memory version.

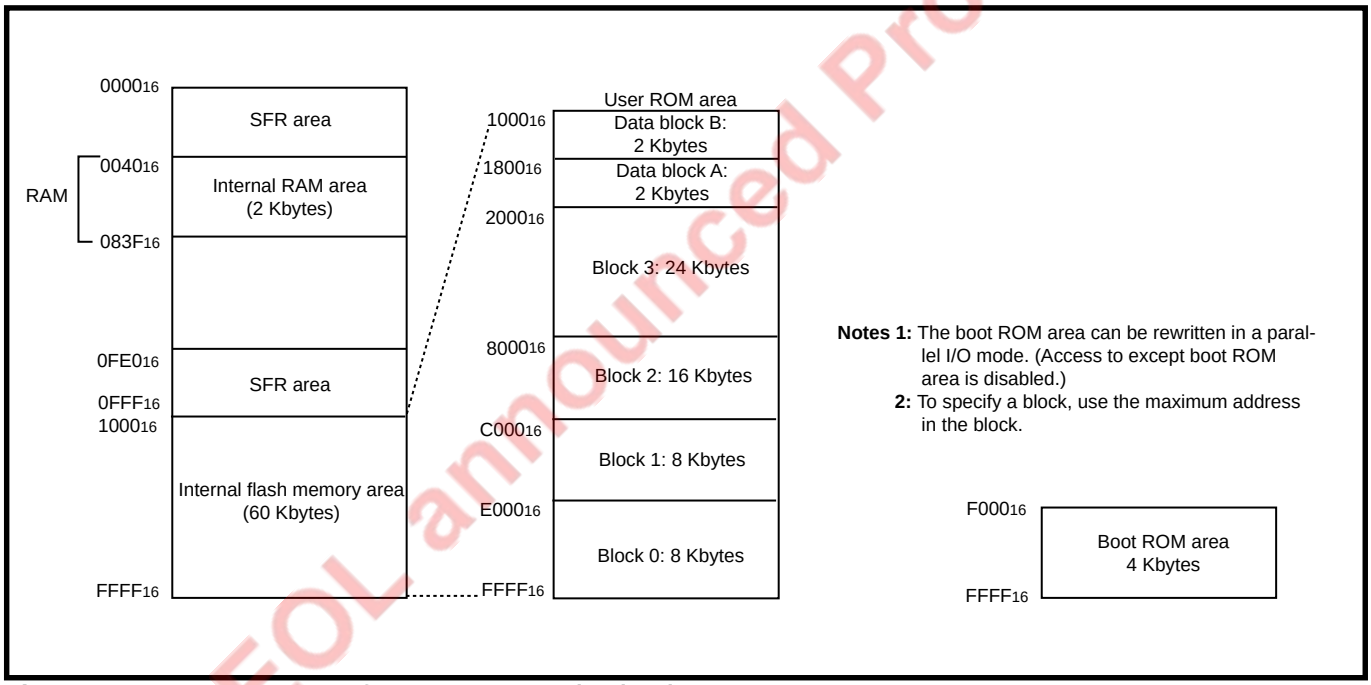


Fig. 2.13.1 Memory map of M38049FFHSP/FP/HP/KP

2.13.3 Relevant registers

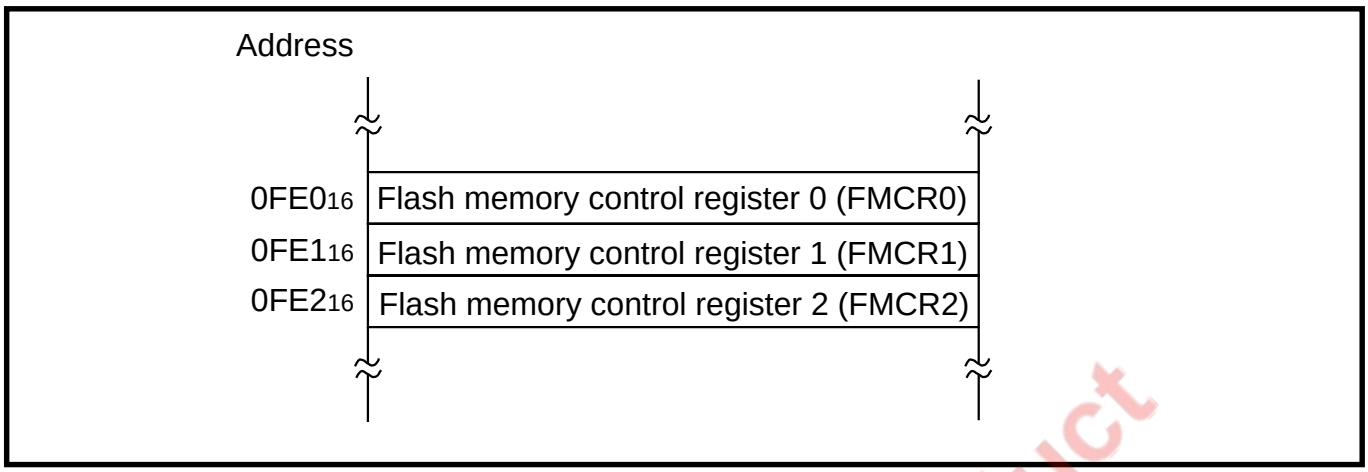


Fig. 2.13.2 Memory map of registers relevant to flash memory

Flash memory control register 0

b7 b6 b5 b4 b3 b2 b1 b0

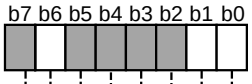
Flash memory control register 0
(FMCR0 : address 0FE016)

b	Name	Functions	At reset	R	W
0	RY/$\overline{BY}$ status flag	0 : Busy (being automatic written or automatic erased) 1 : Ready	1	○	×
1	CPU rewrite mode select bit **(Note 1)**	0 : CPU rewrite mode invalid (software commandes invalid) 1 : CPU rewrite mode valid (Software commands acceptable)	0	○	○
2	8 KB user block E/W enable bit **(Notes 1, 2)**	0: E/W disabled 1: E/W enabled	0	○	○
3	Flash memory reset bit **(Notes 3, 4)**	0: Normal operation 1: Reset	0	○	○
4	Not used (Do not write "1" to this bit.)		0	○	○
5	User ROM area select bit **(Note 5)**	0: User ROM area is accessed 1: Boot ROM area is accessed	0	○	○
6	Program status flag	0: Pass 1: Error	0	○	×
7	Erase status flag	0: Pass 1: Error	0	○	×

Notes 1: For this bit to be set to "1", the user needs to write a "0" and then a "1" to it in succession. For this bit to be set to "0", write "0" only to this bit.
2: This bit can be written only when the CPU rewrite mode select bit is "1".
3: Effective only when the CPU rewrite mode select bit = "1". Fix this bit to "0" when the CPU rewrite mode select bit is "0".
4: When setting this bit to "1" (when the control circuit of flash memory is reset), the flash memory cannot be accessed for 10 μs.
5: Write to this bit from program on RAM.

Fig. 2.13.3 Structure of Flash memory control register 0

Flash memory control register 1



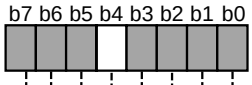
Flash memory control register 1
(FMCR1 : address 0FE1$_{16}$)

b	Name	Functions	At reset	R	W
0	Erase suspend enable bit **(Note 1)**	0 : Suspend invalid 1 : Suspend valid	0	○	○
1	Erase suspend request bit **(Note 2)**	0 : Erase restart (no request issued) 1 : Suspend request (request issued)	0	○	○
2	Nothing is arranged for these bits. If writing to these bits, write “0”. The contents are undefined at reading.		0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6	Erase suspend flag	0 : Erase active 1 : Erase inactive (Erase suspend mode)	1	○	×
7	Nothing is arranged for these bits. If writing to these bits, write “0”. The contents are undefined at reading.		0	○	×

Notes 1: For this bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession.
2: Only when the erase suspend bit is “1”, this bit is valid.

Fig. 2.13.4 Structure of Flash memory control register 1

Flash memory control register 2



Flash memory control register 2
(FMCR2 : address 0FE2$_{16}$)

b	Name	Functions	At reset	R	W
0	Nothing is arranged for these bits. If writing to these bits, write “0”. The contents are undefined at reading.		1	○	×
1			0	○	×
2			1	○	×
3			0	○	×
4	All user block E/W enable bit **(Notes 1, 2)**	0 : E/W disabled 1 : E/W enabled	0	○	○
5	Nothing is arranged for these bits. If writing to these bits, write “0”. The contents are undefined at reading.		0	○	×
6			1	○	×
7			0	○	×

Notes 1: For this bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession.
2: Effective only when the CPU rewrite mode select bit = “1”.

Fig. 2.13.5 Structure of Flash memory control register 2

2.13.4 Parallel I/O mode

In the parallel I/O mode, program/erase to the built-in flash memory can be performed by a flash memory programmer (EFP-I etc.).

The memory area of program/erase is from $0F000_{16}$ to $0FFFF_{16}$ (boot ROM area) or from 01000_{16} to $0FFFF_{16}$ (user ROM area). Be especially careful when erasing; if the memory area is not set correctly, the products will be damaged eternally.

Table 2.13.1 shows the parallel unit when programming by EFP-I in the parallel I/O mode.

•EFP-I provided by Suisei Electronics System Co., Ltd. (http://www.suisei.co.jp/index_e.htm) (product available in Asia and Oceania only)

Table 2.13.1 Parallel unit when parallel programming (when using EFP-I provided by Suisei Electronics System Co., Ltd.)

Products	Parallel unit	Boot ROM area	User ROM area
M38049FFHSP	EF3803F-64S	$0F000_{16}$ to $0FFFF_{16}$	01000_{16} to $0FFFF_{16}$
M38049FFHFP	EF3803F-64F		
M38049FFHHP	EF3803F-64H		
M38049FFHKP	EF3803F-64U		

2.13.5 Standard serial I/O mode

Table 2.13.2 shows a pin connection example (4 wires) between the programmer (EFP-I; Serial unit EF1SRP-01U is required additionally) and the microcomputer when programming in the standard serial I/O mode 1.

•EFP-I provided by Suisei Electronics System Co., Ltd. (http://www.suisei.co.jp/index_e.htm) (product available in Asia and Oceania only)

Table 2.13.2 Connection example to programmer when serial programming (4 wires)

Function	EFP-I (EF1SRP-01U)		Flash memory version		
	Signal name	EF1RP-01U side connector Line number	Pin name	M38049FFHSP pin number	M38049FFHSP pin number
Transfer clock input	T_SCLK	9	$P4_6/S_{CLK1}$	21	13
Serial data input	T_TXD	10	$P4_4/RxD_1$	23	15
Serial data output	T_RXD	11	$P4_5/TxD_1$	22	14
Transmit/Receive enable output	T_BUSY	12	$P4_7/\overline{S_{RDY1}}/CNTR_2$	20	12
"H" input	T_VPP	3	CNV_{SS}	26	18
Reset input	T_RESET	14	$\overline{RESET}$ (**Note 1**)	27	19
Target board power source monitor input	T_VDD **(Note 2)**	4	V_{CC} (**Note 2**)	1	57
GND	GND **(Note 3)**	1, 2, 15, 16	V_{SS}, AV_{SS} (**Note 3**)	32, 3	24, 59

Notes 1: Since reset release after write verification is not performed, when operating MCU after writing, separate a target connection cable.

2: Supply Vcc of EFP-I side from user side so that the power supply voltage of the output buffer used by the EFP-I side becomes the same as user side power supply voltage (Vcc).

3: Four pins (No. 1, 2, 15, and 16) of the EF1SRP-01U side connector are prepared for GND signal. When connecting with a target board, although connection of only one pin does not have a problem, we recommend connecting with two or more pins.

2.13.6 CPU rewrite mode

In the CPU rewrite mode, issuing software commands through the Central Processing Unit (CPU) can rewrite the built-in flash memory. Accordingly, the contents of the built-in flash memory can be rewritten with the microcomputer itself mounted on board, without using the programmer.
Store the rewrite control program to the built-in flash memory in advance. The built-in flash memory cannot be read in the CPU rewrite mode. Accordingly, after transferring the rewrite control program to RAM, execute it on the RAM.
The following commands can be used in the CPU rewrite mode: read array, read status register, clear status register, program, and block erase. For details concerning each command, refer to “CHAPTER 1 Flash memory mode (CPU rewrite mode)”.

(1) CPU rewrite mode beginning/release procedures

Operation procedure in the CPU rewrite mode for the built-in flash memory is described below.

[Beginning procedure]

① Apply “H” to the CNV_{SS} pin and $P4_5/TxD_1$ pin (at selecting boot ROM area).
② Release reset.
③ Set bits 6 and 7 (main clock division ratio selection bits) of the CPU mode register. (Make sure that system clock ϕ is less than 4.0 MHz.)
④ After CPU rewrite mode control program is transferred to internal RAM, jump to this control program on RAM. (The following operations are controlled by this control program).
⑤ Set “1” to the CPU rewrite mode select bit (bit 1 of address $0FFE_{16}$).
For this bit to be set to “1”, the user needs to write “0” and then “1” to it in succession.
⑥ Set “1” to the all user block E/W enable bit (bit 4 of address $0FE2_{16}$). Set the 8 KB user block E/W enable bit. (Set to “0” when E/W is disabled, and set to “1” when E/W is enabled.)*
* For these bits to be set to “1”, the user needs to write “0” and then “1” to those in succession.
⑦ Flash memory operations are executed by using software commands.

Note 1: The following procedures are also necessary.
- Control for data which is input from the external (serial I/O etc.) and to be programmed to the flash memory.
- Initial setting for ports, etc.
- Writing to the watchdog timer

[Release procedure]

① Execute the read array command.
② In order to disable E/W to the user ROM area (except for data block), set “0” to the all user block E/W enable bit (bit 4 of $0FE2_{16}$) and the 8 KB user block E/W enable bit (bit 2 of $0FE0_{16}$) **(Note 2)**.
③ Set the CPU rewrite mode select bit (bit 1 of address $0FFE_{16}$) to “0”.
④ Jump from the CPU rewriting control program on RAM to the user program on the flash memory.

Note 2: Although E/W inhibition is not indispensable, the safety of system improves by disabling E/W except the time of E/W execution.

Also, execute the following processing before the CPU reprogramming mode is selected so that interrupts will not occur during the CPU reprogramming mode.

• Set the interrupt disable flag (I) to “1”

When the watchdog timer has already started, write to the watchdog timer control register (address $001E_{16}$) periodically during the CPU reprogramming mode in order not to generate the reset by the underflow of the watchdog timer H.

During the program or erase execution, watchdog timer is automatically cleared. Accordingly, the inernal reset by underflow does not occur.

When the interrupt request or reset occurs in the CPU reprogramming mode, the microcomputer enters the following state;

• Interrupt occurs

This may cause a program runaway because the read from the flash memory which has the interrupt vector area cannot be performed.

• Underflow of watchdog timer H, reset

This may cause a microcomputer reset; the built-in flash memory control circuit and the flash memory control register are reset. When reset state is released with CNVss = “H” and $P4_5/TxD_1$ = “H”, CPU starts in the boot mode.

Also, when the above interrupt and reset occur during program/erase, error data may still exist after reset release because the reprogramming of the flash memory is not completed, so that reprogramming of the flash memory in the parallel I/O mode or serial I/O mode is required.

2.13.7 Flash memory mode application examples

The control pin processing example on the system board in the standard serial I/O mode and the control example in the CPU rewrite mode are described below.

(1) Control pin connection example on system board in standard serial I/O mode

As shown in Figure 2.13.6, in the standard serial I/O mode, the built-in flash memory can be rewritten with the microcomputer mounted on board. Connection examples of control pins ($P4_4$/RxD, $P4_5$/TxD, $P4_6$/S_{CLK1}, $P4_7$/$\overline{S_{RDY1}}$/$CNTR_2$, CNV_{SS}, and $\overline{RESET}$ pin) in the standard serial I/O mode are described below.

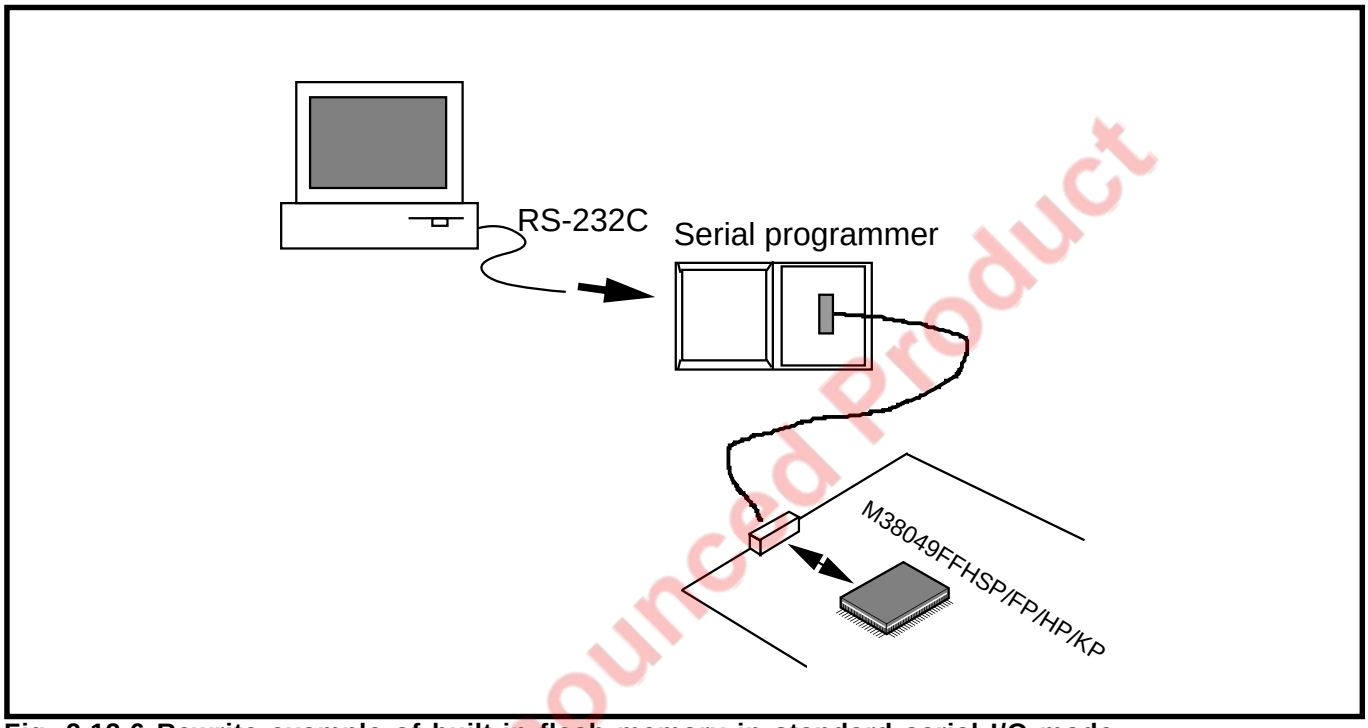
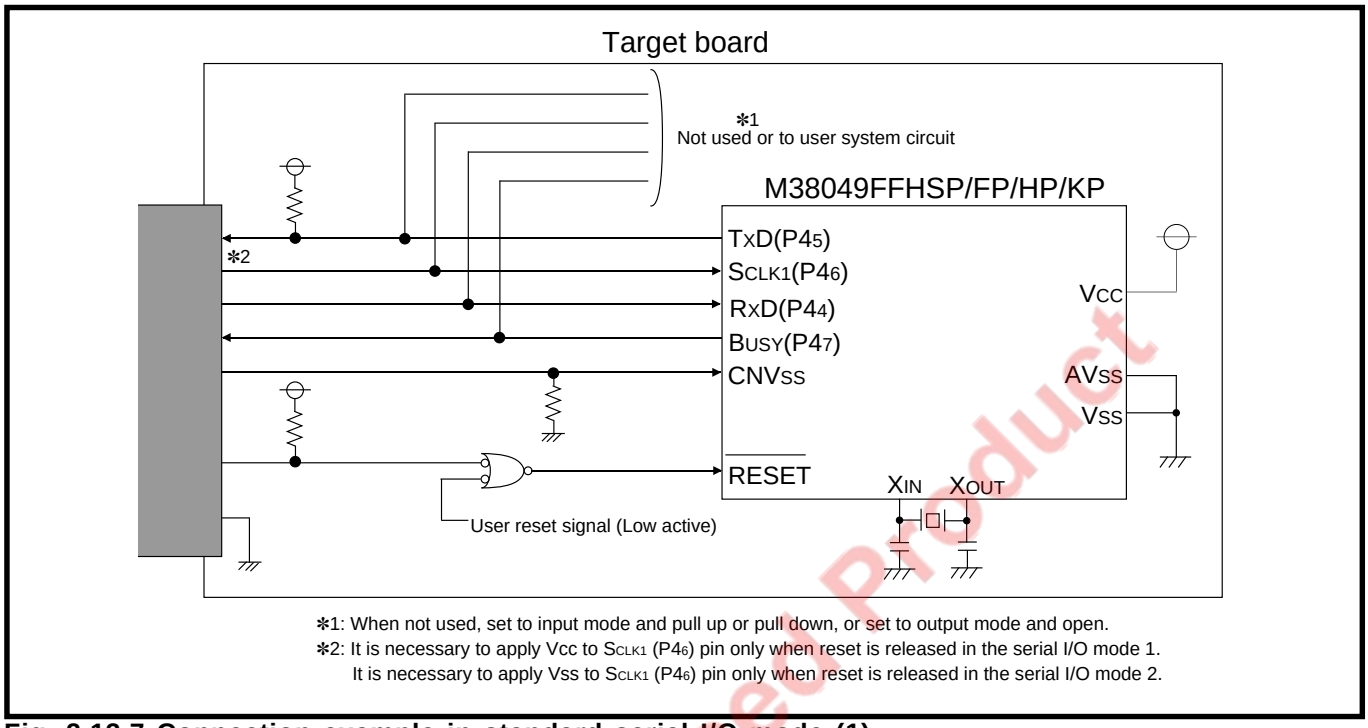


Fig. 2.13.6 Rewrite example of built-in flash memory in standard serial I/O mode

① **When control signals are not affected to user system circuit**

When the control signals in the standard serial I/O mode are not used or not affected to the user system circuit, they can be connected as shown in Figure 2.13.7.

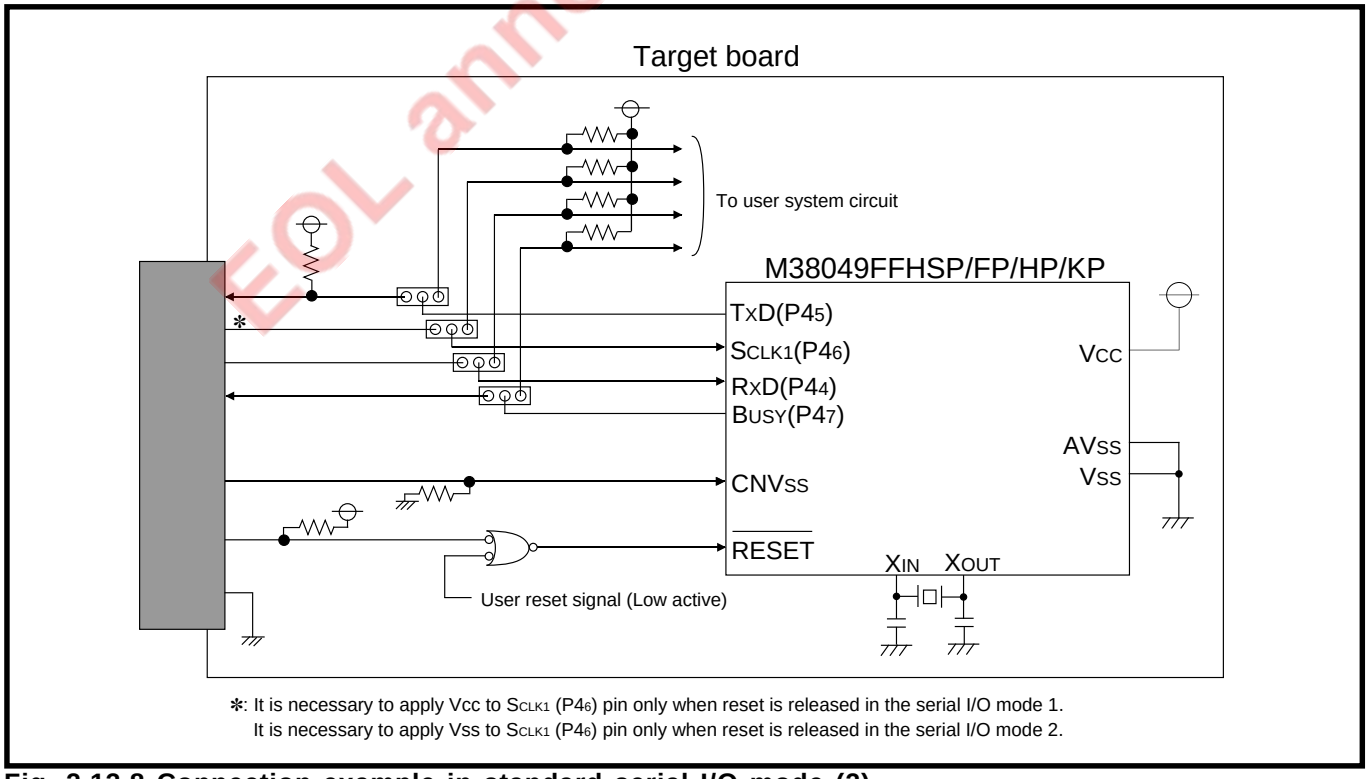


*1: When not used, set to input mode and pull up or pull down, or set to output mode and open.
*2: It is necessary to apply Vcc to SCLK1 (P46) pin only when reset is released in the serial I/O mode 1.
It is necessary to apply Vss to SCLK1 (P46) pin only when reset is released in the serial I/O mode 2.

Fig. 2.13.7 Connection example in standard serial I/O mode (1)

② **When control signals are affected to user system circuit-1**

Figure 2.13.8 shows an example that the jumper switch cut-off the control signals not to supply to the user system circuit in the standard serial I/O mode.

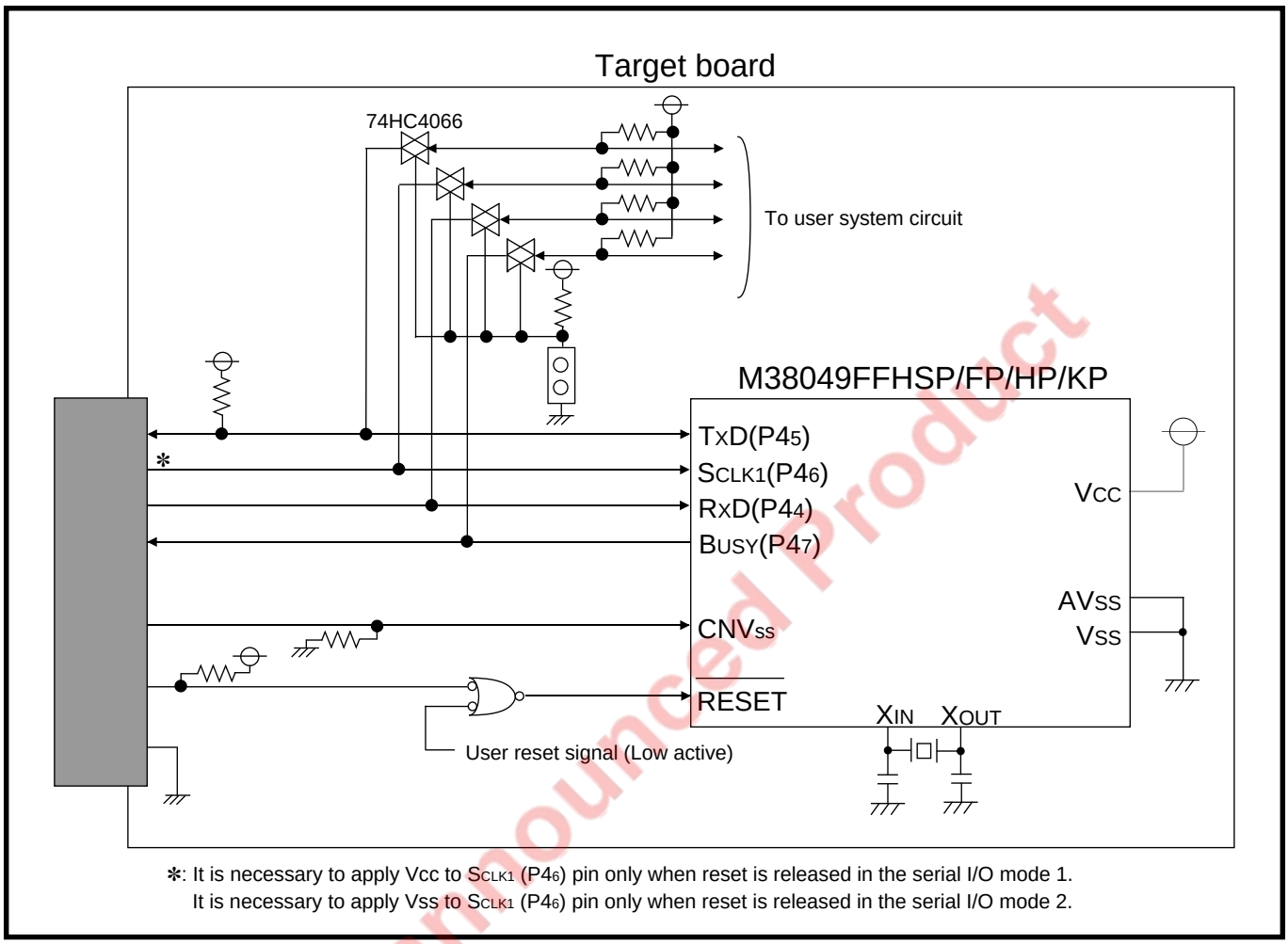


*: It is necessary to apply Vcc to SCLK1 (P46) pin only when reset is released in the serial I/O mode 1.
It is necessary to apply Vss to SCLK1 (P46) pin only when reset is released in the serial I/O mode 2.

Fig. 2.13.8 Connection example in standard serial I/O mode (2)

③ **When control signals are affected to user system circuit-2**

Figure 2.13.9 shows an example that the analog switch (74HC4066) cut-off the control signals not to supply to the user system circuit in the standard serial I/O mode.



*: It is necessary to apply Vcc to SCLK1 (P46) pin only when reset is released in the serial I/O mode 1.
It is necessary to apply Vss to SCLK1 (P46) pin only when reset is released in the serial I/O mode 2.

Fig. 2.13.9 Connection example in standard serial I/O mode (3)

(2) Control pin termination example in CPU rewrite mode

In this example, data is received by using serial I/O, and the data is programmed to the built-in flash memory in the CPU rewrite mode.

Figure 2.13.10 shows an example of the reprogramming system for the built-in flash memory in the CPU rewrite mode. For the CPU rewrite mode beginning/release method, refer to “2.13.6 CPU rewrite mode.”

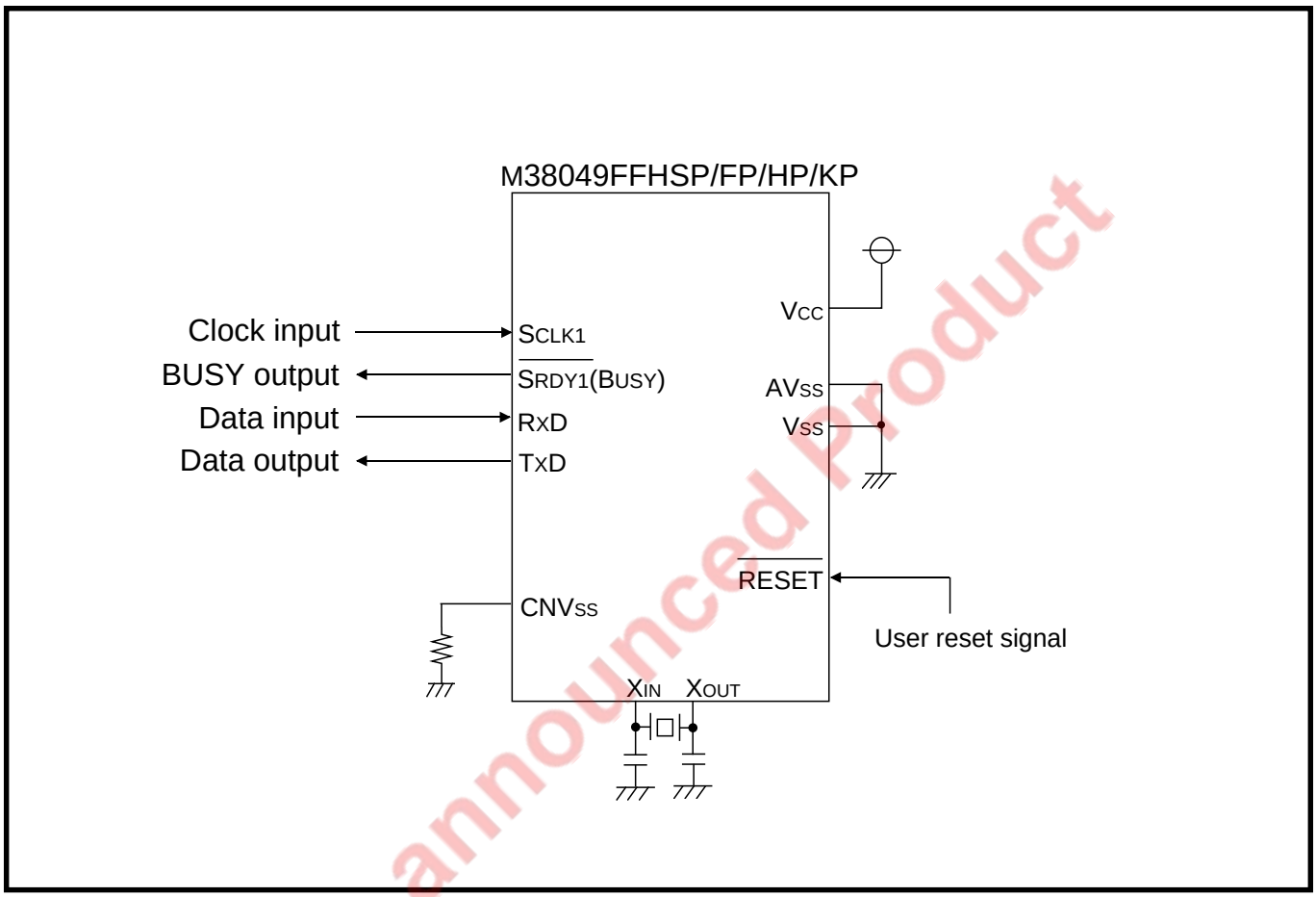


Fig. 2.13.10 Example of rewrite system for built-in flash memory in CPU rewrite mode (single-chip mode)

2.13.8 Notes on CPU rewrite mode

(1) Operation speed
During CPU rewrite mode, set the system clock ϕ 4.0 MHz or less using the main clock division ratio selection bits (bits 6 and 7 of address $003B_{16}$).

(2) Instructions inhibited against use
The instructions which refer to the internal data of the flash memory cannot be used during the CPU rewrite mode.

(3) Interrupts inhibited against use
The interrupts cannot be used during the CPU rewrite mode because they refer to the internal data of the flash memory.

(4) Watchdog timer
In case of the watchdog timer has been running already, the internal reset generated by watchdog timer underflow does not happen, because of watchdog timer is always clearing during program or erase operation.

(5) Reset
Reset is always valid. In case of CNV_{SS} = "H" when reset is released, boot mode is active. So the program starts from the address contained in address $FFFC_{16}$ and $FFFD_{16}$ in boot ROM area.

CHAPTER 3

APPENDIX

3.1 ELECTRICAL CHARACTERISTICS

3.1.1 Absolute maximum ratings

Table 3.1.1 Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Power source voltages	All voltages are based on Vss.	–0.3 to 6.5	V
V_I	Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_4$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, V_{REF}	Output transistors are cut off.	–0.3 to V_{CC} +0.3	V
V_I	Input voltage $P3_2$, $P3_3$		–0.3 to 5.8	V
V_I	Input voltage $\overline{RESET}$, X_{IN}		–0.3 to V_{CC} +0.3	V
V_I	Input voltage CNVss		–0.3 to V_{CC} +0.3	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_4$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, X_{OUT}		–0.3 to V_{CC} +0.3	V
V_O	Output voltage $P3_2$, $P3_3$		–0.3 to 5.8	V
P_d	Power dissipation	Ta = 25°C	1000 **(Note)**	mW
T_{opr}	Operating temperature		–20 to 85	°C
T_{stg}	Storage temperature		–65 to 125	°C

Note: This value is 300 mW except SP package.

3.1.2 Recommended operating conditions

Table 3.1.2 Recommended operating conditions (1)

(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{CC}	Power source voltage **(Note 1)**	When start oscillating **(Note 2)**		2.7	5.0	5.5	V
		High-speed mode $f(\phi) = f(X_{IN})/2$	$f(X_{IN}) \le 8.4$ MHz	2.7	5.0	5.5	V
			$f(X_{IN}) \le 12.5$ MHz	4.0	5.0	5.5	V
			$f(X_{IN}) \le 16.8$ MHz	4.5	5.0	5.5	V
		Middle-speed mode $f(\phi) = f(X_{IN})/8$	$f(X_{IN}) \le 12.5$ MHz	2.7	5.0	5.5	V
			$f(X_{IN}) \le 16.8$ MHz	4.5	5.0	5.5	V
V_{SS}	Power source voltage				0		V
V_{IH}	“H” input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_4$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$			0.8V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage $P3_2$, $P3_3$			0.8V_{CC}		5.5	V
V_{IH}	“H” input voltage (when I^2C-BUS input level is selected) SDA, SCL			0.7V_{CC}		5.5	V
V_{IH}	“H” input voltage (when SMBUS input level is selected) SDA, SCL			1.4		5.5	V
V_{IH}	“H” input voltage $\overline{RESET}$, X_{IN}, CNV_{SS}			0.8V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage X_{CIN}			2		V_{CC}	V
V_{IL}	“L” input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$,$P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$			0		0.2V_{CC}	V
V_{IL}	“L” input voltage (when I^2C-BUS input level is selected) SDA, SCL			0		0.3Vcc	V
V_{IL}	“L” input voltage (when SMBUS input level is selected) SDA, SCL			0		0.6	V
V_{IL}	“L” input voltage $\overline{RESET}$, CNV_{SS}			0		0.2V_{CC}	V
V_{IL}	“L” input voltage X_{IN}					0.16V_{CC}	V
V_{IL}	“L” input voltage X_{CIN}					0.4	V

Notes 1: When using A/D converter, see A/D converter recommended operating conditions.

2: The start voltage and the start time for oscillation depend on the using oscillator, oscillation circuit constant value and operating temperature range, etc.. Particularly a high-frequency oscillator might require some notes in the low voltage operation.

Table 3.1.3 Recommended operating conditions (2)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
$f(X_{IN})$	Main clock input oscillation frequency **(Note 1)**	High-speed mode $f(\phi) = f(X_{IN})/2$	2.7 ≤ V_{CC} < 4.0 V			$\frac{(9 \times V_{CC}-0.3) \times 1.05}{3}$	MHz
			4.0 ≤ V_{CC} < 4.5 V			$\frac{(24 \times V_{CC}-60) \times 1.05}{3}$	MHz
			4.5 ≤ V_{CC} ≤ 5.5 V			16.8	MHz
		Middle-speed mode $f(\phi) = f(X_{IN})/8$	2.7 ≤ V_{CC} < 4.5 V			$\frac{(15 \times V_{CC}+39) \times 1.1}{7}$	MHz
			4.5 ≤ V_{CC} ≤ 5.5 V			16.8	MHz
$f(X_{CIN})$	Sub-clock input oscillation frequency **(Notes 1, 2)**				32.768	50	kHz

Notes 1: When the oscillation frequency has a duty cycle of 50%.
2: When using the microcomputer in low-speed mode, set the sub-clock input oscillation frequency on condition that $f(X_{CIN}) < f(X_{IN})/3$.

Table 3.1.4 Recommended operating conditions (3)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$\Sigma I_{OH(peak)}$	"H" total peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_4$–$P3_7$ **(Note 1)**			–80	mA
$\Sigma I_{OH(peak)}$	"H" total peak output current	$P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$ **(Note 1)**			–80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_7$ **(Note 1)**			80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current	$P2_0$–$P2_7$ **(Note 1)**			80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$ **(Note 1)**			80	mA
$\Sigma I_{OH(avg)}$	"H" total average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_4$–$P3_7$ **(Note 1)**			–40	mA
$\Sigma I_{OH(avg)}$	"H" total average output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$ **(Note 1)**			–40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_7$ **(Note 1)**			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current	$P2_0$–$P2_7$ **(Note 1)**			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$ **(Note 1)**			40	mA
$I_{OH(peak)}$	"H" peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_4$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$ **(Note 2)**			–10	mA
$I_{OL(peak)}$	"L" peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$ **(Note 2)**			10	mA
$I_{OL(peak)}$	"L" peak output current	$P2_0$–$P2_7$ **(Note 2)**			20	mA
$I_{OH(avg)}$	"H" average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P3_1$, $P3_4$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$ **(Note 3)**			–5	mA
$I_{OL(avg)}$	"L" average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$ **(Note 3)**			5	mA
$I_{OL(avg)}$	"L" average output current	$P2_0$–$P2_7$ **(Note 3)**			10	mA

Notes 1: The total output current is the sum of all the currents flowing through all the applicable ports. The total average current is an average value measured over 100 ms. The total peak current is the peak value of all the currents.
2: The peak output current is the peak current flowing in each port.
3: The average output current I_{OL}(avg), I_{OH}(avg) are average value measured over 100 ms.

3.1.3 Electrical characteristics

Table 3.1.5 Electrical characteristics (1)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	"H" output voltage P0₀–P0₇, P1₀–P1₇, P2₀–P2₇, P3₀, P3₁, P3₄–P3₇, P4₀–P4₇, P5₀–P5₇, P6₀–P6₇ **(Note 1)**	I_{OH} = –10 mA V_{CC} = 4.0 to 5.5 V	V_{CC}–2.0			V
		I_{OH} = –1.0 mA V_{CC} = 1.8 to 5.5 V	V_{CC}–1.0			V
V_{OL}	"L" output voltage P0₀–P0₇, P1₀–P1₇, P2₀–P2₇, P3₀–P3₇, P4₀–P4₇, P5₀–P5₇, P6₀–P6₇	I_{OL} = 10 mA V_{CC} = 4.0 to 5.5 V			2.0	V
		I_{OL} = 1.6 mA V_{CC} = 1.8 to 5.5 V			1.0	V
V_{OL}	"L" output voltage P2₀–P2₇	I_{OL} = 20 mA V_{CC} = 4.0 to 5.5 V			2.0	V
		I_{OL} = 1.6 mA V_{CC} = 1.8 to 5.5 V			0.4	V
V_{T+}–V_{T-}	Hysteresis CNTR₀, CNTR₁, CNTR₂, INT₀–INT₄			0.4		V
V_{T+}–V_{T-}	Hysteresis RxD1, SCLK1, SIN2, SCLK2, RxD3, SCLK3			0.5		V
V_{T+}–V_{T-}	Hysteresis $\overline{RESET}$			0.5		V
I_{IH}	"H" input current P0₀–P0₇, P1₀–P1₇, P2₀–P2₇, P3₀–P3₇, P4₀–P4₇, P5₀–P5₇, P6₀–P6₇	V_I = V_{CC} (Pin floating. Pull-up transistors "off")			5.0	µA
I_{IH}	"H" input current $\overline{RESET}$, CNV_{SS}	V_I = V_{CC}			5.0	µA
I_{IH}	"H" input current X_{IN}	V_I = V_{CC}		4.0		µA
I_{IL}	"L" input current P0₀–P0₇, P1₀–P1₇, P2₀–P2₇, P3₀–P3₇, P4₀–P4₇, P5₀–P5₇, P6₀–P6₇	V_I = V_{SS} (Pin floating. Pull-up transistors "off")			–5.0	µA
I_{IL}	"L" input current $\overline{RESET}$,CNV_{SS}	V_I = V_{SS}			–5.0	µA
I_{IL}	"L" input current X_{IN}	V_I = V_{SS}		–4.0		µA
I_{IL}	"L" input current (at Pull-up) P0₀–P0₇, P1₀–P1₇, P2₀–P2₇, P3₀, P3₁, P3₄–P3₇, P4₀–P4₇, P5₀–P5₇, P6₀–P6₇	V_I = V_{SS} V_{CC} = 5.0 V	–80	–210	–420	µA
		V_I = V_{SS} V_{CC} = 3.0 V	–30	–70	–140	µA
V_{RAM}	RAM hold voltage	When clock stopped	1.8		V_{CC}	V

Note 1: P3₅ is measured when the P3₅/TxD_3 P-channel output disable bit of the UART3 control register (bit 4 of address 0033₁₆) is "0".
P4₅ is measured when the P4₅/TxD_1 P-channel output disable bit of the UART1 control register (bit 4 of address 001B₁₆) is "0".

Table 3.1.6 Electrical characteristics (2)
(V_{CC} = 2.7 to 5.5 V, T_a = –20 to 85 °C, f(X_{CIN})=32.768kHz (Stoped in middle-speed mode), Output transistors "off", AD converter not operated)

Symbol	Parameter	Test conditions			Limits			Unit
					Min.	Typ.	Max.	
I_{CC}	Power source current	High-speed mode	V_{CC} = 5V	f(X_{IN}) = 16.8 MHz		5.5	8,3	mA
				f(X_{IN}) = 12.5 MHz		4.5	6.8	mA
				f(X_{IN}) = 8.4 MHz		3.5	5.3	mA
				f(X_{IN}) = 4.2 MHz		2.2	3.3	mA
				f(X_{IN}) = 16.8 MHz (in WIT state)		2.2	3.3	mA
			V_{CC} = 3V	f(X_{IN}) = 8.4 MHz		2.7	4.1	mA
				f(X_{IN}) = 4.2 MHz		1.8	2.7	mA
				f(X_{IN}) = 2.1 MHz		1.1	1.7	mA
		Middle-speed mode	V_{CC} = 5V	f(X_{IN}) = 16.8 MHz		3.0	4.5	mA
				f(X_{IN}) = 12.5 MHz		2.4	3.6	mA
				f(X_{IN}) = 8.4 MHz		2.0	3.0	mA
				f(X_{IN}) = 16.8 MHz (in WIT state)		2.1	3.2	mA
			V_{CC} = 3V	f(X_{IN}) = 12.5 MHz		1.7	2.6	mA
				f(X_{IN}) = 8.4 MHz		1.5	2.3	mA
				f(X_{IN}) = 6.3 MHz		1.3	2.0	mA
		Low-speed mode	V_{CC} = 5V	f(X_{IN}) = stopped		410	630	µA
				In WIT state		4.5	6.8	µA
			V_{CC} = 3V	f(X_{IN}) = stopped		400	600	µA
				In WIT state		3.7	5.6	µA
		In STP state (All oscillation stopped)		Ta = 25 °C		0.55	3.0	µA
				Ta = 85 °C		0.75		µA
		Increment when A/D conversion is executed		f(X_{IN}) = 16.8 MHz, V_{CC} = 5V In Middle-, high-speed mode		1000		µA

3.1.4 A/D converter characteristics

Table 3.1.7 A/D converter recommended operating conditions

(V_{CC} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage (When A/D converter is used)	8-bit A/D mode **(Note 1)**	2.7	5.0	5.5	V
		10-bit A/D mode **(Note 2)**	2.7	5.0	5.5	
V_{REF}	Analog reference voltage		2.0		V_{CC}	V
AV_{SS}	Analog power source voltage			0		V
V_{IA}	Analog input voltage		0		V_{CC}	V
f(X_{IN})	Main clock oscillation frequency (When A/D converter is used)	2.7 ≤ V_{CC} < 4.0 V	0.5		$\frac{(9\times V_{CC}-0.3)\times 1.05}{3}$	MHz
		4.0 ≤ V_{CC} < 4.5 V	0.5		$\frac{(24\times V_{CC}-60)\times 1.05}{3}$	
		4.5 ≤ V_{CC} ≤ 5.5 V	0.5		16.8	

Note 1: 8-bit A/D mode: When the conversion mode selection bit (bit 7 of address 003816) is "1".
2: 10-bit A/D mode: When the conversion mode selection bit (bit 7 of address 003816) is "0".

Table 3.1.8 A/D converter characteristics

(V_{CC} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
–	Resolution	8-bit A/D mode **(Note 1)**				8	bit
		10-bit A/D mode **(Note 2)**				10	
–	Absolute accuracy (excluding quantization error)	8-bit A/D mode **(Note 1)**	2.7 ≤ V_{REF} ≤ 5.5 V			±2	LSB
		10-bit A/D mode **(Note 2)**	2.7 ≤ V_{REF} ≤ 5.5 V			±4	
t_{CONV}	Conversion time	8-bit A/D mode **(Note 1)**				50	2tc(X_{IN})
		10-bit A/D mode **(Note 2)**				61	
R_{LADDER}	Ladder resistor			12	35	100	kΩ
I_{VREF}	Reference power source input current	at A/D converter operated	V_{REF} = 5.0 V	50	150	200	µA
		at A/D converter stopped	V_{REF} = 5.0 V			5	µA
$I_{I(AD)}$	A/D port inout current					5	µA

Note 1: 8-bit A/D mode: When the conversion mode selection bit (bit 7 of address 003816) is "1".
2: 10-bit A/D mode: When the conversion mode selection bit (bit 7 of address 003816) is "0".

3.1.5 D/A converter characteristics

Table 3.1.9 D/A converter characteristics

(V_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 V to V_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
–	Resolution					8	bit
–	Absolute accuracy	4.0 ≤ V_{REF} ≤ 5.5 V				1.0	%
		2.7 ≤ V_{REF} < 4.0 V				2.5	%
tsu	Setting time					3	µs
RO	Output resistor			2	3.5	5	kΩ
I_{VREF}	Reference power source input current **(Note 1)**					3.2	mA

Note 1: Using one D/A converter, with the value in the DA conversion register of the other D/A converter being "0016".

3.1.6 Power source circuit timing characteristics

Table 3.1.10 Power source circuit timing characteristics

(V_{CC} = 2.7 to 5.5 V, V_{REF} = 2.7 V to V_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
td(P–R)	Internal power source stable time at power-on	2.7 ≤ V_{CC} < 5.5 V			2	ms

3.1.7 Timing requirements and switching characteristics

Table 3.1.11 Timing requirements (1)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$t_W(\overline{RESET})$	Reset input “L” pulse width		td(P-R) ms + 16			X_{IN} cycle
$t_C(X_{IN})$	Main clock X_{IN} input cycle time	4.5≤Vcc≤5.5 V	59.5			ns
		4.0≤Vcc<4.5 V	10000/(86Vcc-219)			
		2.7≤Vcc<4.0 V	26×10^3/(82Vcc-3)			
$t_{WH}(X_{IN})$	Main clock X_{IN} input “H” pulse width	4.5≤Vcc≤5.5 V	25			ns
		4.0≤Vcc<4.5 V	4000/(86Vcc-219)			
		2.7≤Vcc<4.0 V	10000/(82Vcc-3)			
$t_{WL}(X_{IN})$	Main clock X_{IN} input “L” pulse width	4.5≤Vcc≤5.5 V	25			ns
		4.0≤Vcc<4.5 V	4000/(86Vcc-219)			
		2.7≤Vcc<4.0 V	10000/(82Vcc-3)			
$t_C(X_{CIN})$	Sub-clock X_{CIN} input cycle time		20			µs
$t_{WH}(X_{CIN})$	Sub-clock X_{CIN} input “H” pulse width		5			µs
$t_{WL}(X_{CIN})$	Sub-clock X_{CIN} input “L” pulse width		5			µs
t_C(CNTR)	$CNTR_0$–$CNTR_2$ input cycle time	4.5≤Vcc≤5.5 V	120			ns
		4.0≤Vcc<4.5 V	160			
		2.7≤Vcc<4.0 V	250			
t_{WH}(CNTR)	$CNTR_0$–$CNTR_2$ input “H” pulse width	4.5≤Vcc≤5.5 V	48			ns
		4.0≤Vcc<4.5 V	64			
		2.7≤Vcc<4.0 V	115			
t_{WL}(CNTR)	$CNTR_0$–$CNTR_2$ input “L” pulse width	4.5≤Vcc≤5.5 V	48			ns
		4.0≤Vcc<4.5 V	64			
		2.7≤Vcc<4.0 V	115			
t_{WH}(INT)	INT_{00}, INT_{01}, INT_1, INT_2, INT_3, INT_{40}, INT_{41} input “H” pulse width	4.5≤Vcc≤5.5 V	48			ns
		4.0≤Vcc<4.5 V	64			
		2.7≤Vcc<4.0 V	115			
t_{WL}(INT)	INT_{00}, INT_{01}, INT_1, INT_2, INT_3, INT_{40}, INT_{41} input “L” pulse width	4.5≤Vcc≤5.5 V	48			ns
		4.0≤Vcc<4.5 V	64			
		2.7≤Vcc<4.0 V	115			

Table 3.1.12 Timing requirements (2)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$t_{C(SCLK1)}$, $t_{C(SCLK3)}$	Serial I/O1, serial I/O3 clock input cycle time **(Note)**	4.5≤Vcc≤5.5 V	250			ns
		4.0≤Vcc<4.5 V	320			
		2.7≤Vcc<4.0 V	500			
$t_{WH(SCLK1)}$, $t_{WH(SCLK3)}$	Serial I/O1, serial I/O3 clock input “H” pulse width **(Note)**	4.5≤Vcc≤5.5 V	120			ns
		4.0≤Vcc<4.5 V	150			
		2.7≤Vcc<4.0 V	240			
$t_{WL(SCLK1)}$, $t_{WL(SCLK3)}$	Serial I/O1, serial I/O3 clock input “L” pulse width **(Note)**	4.5≤Vcc≤5.5 V	120			ns
		4.0≤Vcc<4.5 V	150			
		2.7≤Vcc<4.0 V	240			
$t_{su(RxD1-SCLK1)}$, $t_{su(RxD3-SCLK3)}$	Serial I/O1, serial I/O3 clock input setup time	4.5≤Vcc≤5.5 V	70			ns
		4.0≤Vcc<4.5 V	90			
		2.7≤Vcc<4.0 V	100			
$t_{h(SCLK1-RxD1)}$, $t_{h(SCLK3-RxD3)}$	Serial I/O1, serial I/O3 clock input hold time	4.5≤Vcc≤5.5 V	32			ns
		4.0≤Vcc<4.5 V	40			
		2.7≤Vcc<4.0 V	50			
$t_{C(SCLK2)}$	Serial I/O2 clock input cycle time	4.5≤Vcc≤5.5 V	500			ns
		4.0≤Vcc<4.5 V	650			
		2.7≤Vcc<4.0 V	1000			
$t_{WH(SCLK2)}$	Serial I/O2 clock input “H” pulse width	4.5≤Vcc≤5.5 V	200			ns
		4.0≤Vcc<4.5 V	260			
		2.7≤Vcc<4.0 V	400			
$t_{WL(SCLK2)}$	Serial I/O2 clock input “L” pulse width	4.5≤Vcc≤5.5 V	200			ns
		4.0≤Vcc<4.5 V	260			
		2.7≤Vcc<4.0 V	400			
$t_{su(SIN2-SCLK2)}$	Serial I/O2 clock input setup time	4.5≤Vcc≤5.5 V	100			ns
		4.0≤Vcc<4.5 V	130			
		2.7≤Vcc<4.0 V	200			
$t_{h(SCLK2-SIN2)}$	Serial I/O2 clock input hold time	4.5≤Vcc≤5.5 V	100			ns
		4.0≤Vcc<4.5 V	130			
		2.7≤Vcc<4.0 V	150			

Note : When bit 6 of address $001A_{16}$ and bit 6 of address 0032_{16} are “1” (clock synchronous).
Divide this value by four when bit 6 of address $001A_{16}$ and bit 6 of address 0032_{16} are “0” (UART).

Table 3.1.13 Switching characteristics
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
$t_{WH}(S_{CLK1})$ $t_{WH}(S_{CLK3})$	Serial I/O1, serial I/O3 clock output “H” pulse width	4.5≤Vcc≤5.5 V	Fig. 3.1.1	$tC(S_{CLK1})$2-30, $tC(S_{CLK3})$/2-30			ns
		4.0≤Vcc<4.5 V		$tC(S_{CLK1})$2-35, $tC(S_{CLK3})$/2-35			
		2.7≤Vcc<4.0 V		$tC(S_{CLK1})$2-40, $tC(S_{CLK3})$/2-40			
$t_{WL}(S_{CLK1})$ $t_{WL}(S_{CLK3})$	Serial I/O1, serial I/O3 clock output “L” pulse width	4.5≤Vcc≤5.5 V		$tC(S_{CLK1})$2-30, $tC(S_{CLK3})$/2-30			ns
		4.0≤Vcc<4.5 V		$tC(S_{CLK1})$2-35, $tC(S_{CLK3})$/2-35			
		2.7≤Vcc<4.0 V		$tC(S_{CLK1})$2-40, $tC(S_{CLK3})$/2-40			
$t_d(S_{CLK1}\text{-}TxD1)$ $t_d(S_{CLK3}\text{-}TxD3)$	Serial I/O1, serial I/O3 output delay time **(Note)**	4.5≤Vcc≤5.5 V				140	ns
		4.0≤Vcc<4.5 V				200	
		2.7≤Vcc<4.0 V				350	
$t_v(S_{CLK1}\text{-}TxD1)$ $t_v(S_{CLK3}\text{-}TxD3)$	Serial I/O1, serial I/O3 output valid time **(Note)**	4.5≤Vcc≤5.5 V		-30			ns
		4.0≤Vcc<4.5 V		-30			
		2.7≤Vcc<4.0 V		-30			
$t_r(S_{CLK1})$ $t_r(S_{CLK3})$	Serial I/O1, serial I/O3 rise time of clock output	4.5≤Vcc≤5.5 V				30	ns
		4.0≤Vcc<4.5 V				35	
		2.7≤Vcc<4.0 V				40	
$t_f(S_{CLK1})$ $t_f(S_{CLK3})$	Serial I/O1, serial I/O3 fall time of clock output	4.5≤Vcc≤5.5 V				30	ns
		4.0≤Vcc<4.5 V				35	
		2.7≤Vcc<4.0 V				40	
$t_{WH}(S_{CLK2})$	Serial I/O2 clock output “H” pulse width	4.5≤Vcc≤5.5 V		$tC(S_{CLK2})$/2-160			ns
		4.0≤Vcc<4.5 V		$tC(S_{CLK2})$/2-200			
		2.7≤Vcc<4.0 V		$tC(S_{CLK2})$/2-240			
$t_{WL}(S_{CLK2})$	Serial I/O2 clock output “L” pulse width	4.5≤Vcc≤5.5 V		$tC(S_{CLK2})$/2-160			ns
		4.0≤Vcc<4.5 V		$tC(S_{CLK2})$/2-200			
		2.7≤Vcc<4.0 V		$tC(S_{CLK2})$/2-240			
$t_d(S_{CLK2}\text{-}S_{OUT2})$	Serial I/O2 output delay time	4.5≤Vcc≤5.5 V				200	ns
		4.0≤Vcc<4.5 V				250	
		2.7≤Vcc<4.0 V				300	
$t_v(S_{CLK2}\text{-}S_{OUT2})$	Serial I/O2 output valid time	4.5≤Vcc≤5.5 V			0		ns
		4.0≤Vcc<4.5 V			0		
		2.7≤Vcc<4.0 V			0		
$t_f(S_{CLK2})$	Serial I/O2 fall time of clock output	4.5≤Vcc≤5.5 V				30	ns
		4.0≤Vcc<4.5 V				35	
		2.7≤Vcc<4.0 V				40	
t_r(CMOS)	CMOS rise time of output **(Note)**	4.5≤Vcc≤5.5 V			10	30	ns
		4.0≤Vcc<4.5 V			12	35	
		2.7≤Vcc<4.0 V			15	40	
t_f(CMOS)	CMOS fall time of output **(Note)**	4.5≤Vcc≤5.5 V			10	30	ns
		4.0≤Vcc<4.5 V			12	35	
		2.7≤Vcc<4.0 V			15	40	

Note: When the $P4_5/TxD_1$ P-channel output disable bit of the UART1 control register (bit 4 of address $001B_{16}$) is “0”.

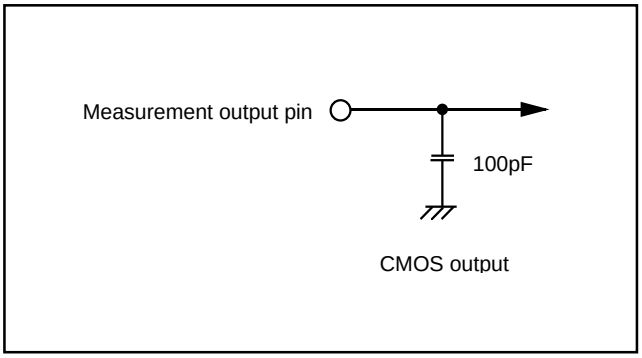


Fig. 3.1.1 Circuit for measuring output switching characteristics (1)

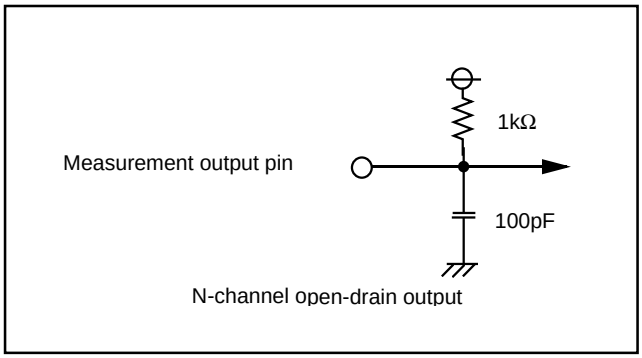


Fig. 3.1.2 Circuit for measuring output switching characteristics (2)

Single-chip mode timing diagram

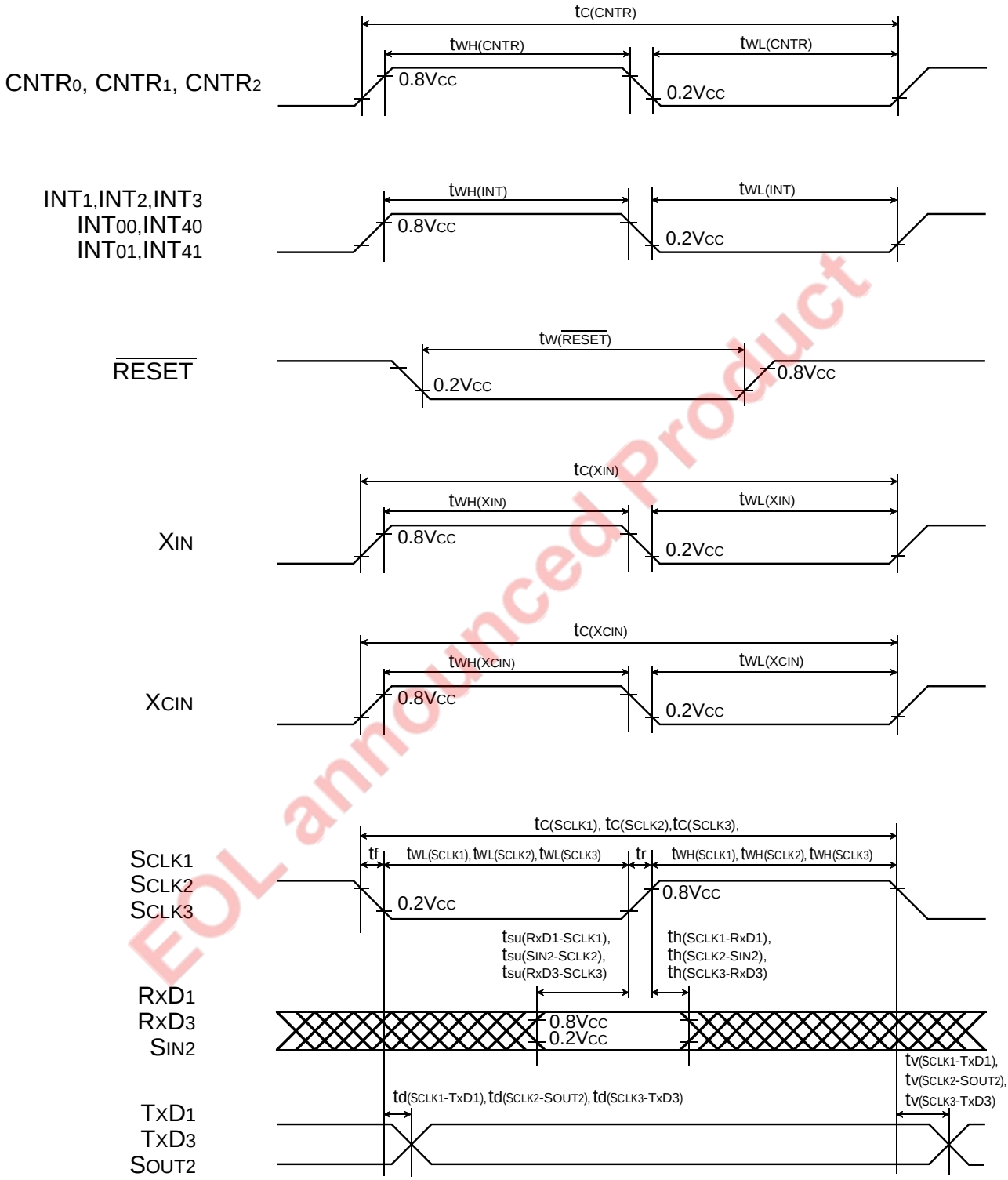


Fig. 3.1.3 Timing diagram (in single-chip mode)

3.1.8 Multi-master I²C-BUS bus line characteristics

Table 3.1.14 Multi-master I²C-BUS bus line characteristics

Symbol	Parameter	Standard clock mode		High-speed clock mode		Unit
		Min.	Max.	Min.	Max.	
t_{BUF}	Bus free time	4.7		1.3		µs
$t_{HD;STA}$	Hold time for START condition	4.0		0.6		µs
t_{LOW}	Hold time for SCL clock = “0”	4.7		1.3		µs
t_R	Rising time of both SCL and SDA signals		1000	20+0.1C_b	300	ns
$t_{HD;DAT}$	Data hold time	0		0	0.9	µs
t_{HIGH}	Hold time for SCL clock = “1”	4.0		0.6		µs
t_F	Falling time of both SCL and SDA signals		300	20+0.1C_b	300	ns
$t_{SU;DAT}$	Data setup time	250		100		ns
$t_{SU;STA}$	Setup time for repeated START condition	4.7		0.6		µs
$t_{SU;STO}$	Setup time for STOP condition	4.0		0.6		µs

Note: C_b = total capacitance of 1 bus line

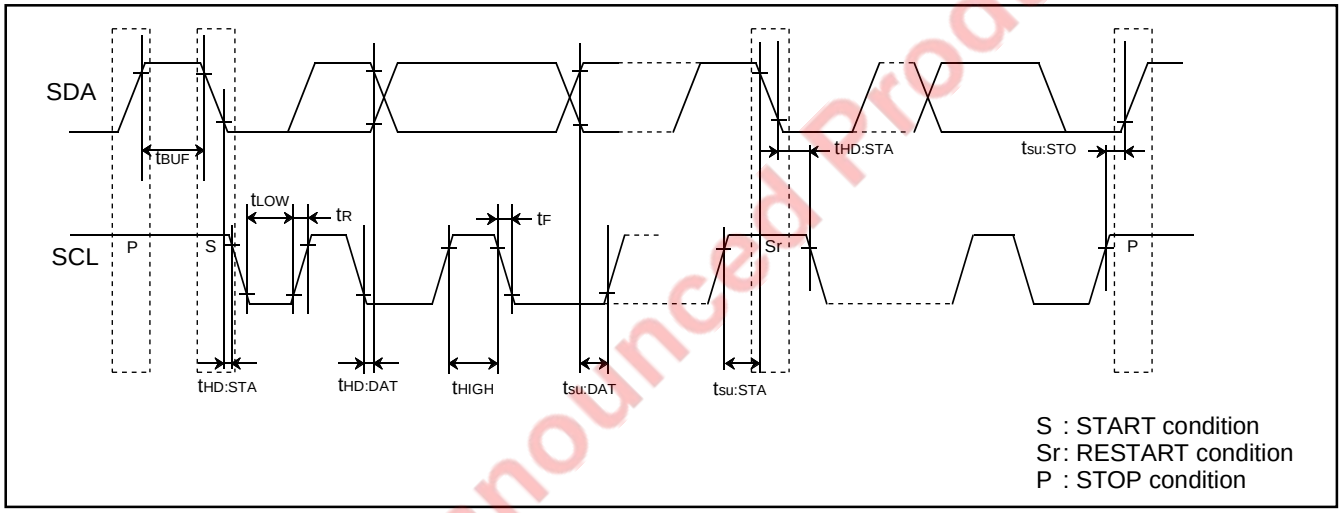


Fig. 3.1.4 Timing diagram of multi-master I²C-BUS

3.2 Standard characteristics

Standard characteristics described below are just examples of the 3804 Group (spec. H)'s characteristics and are not guaranteed. For rated values, refer to **"3.1 Electrical characteristics"**.

3.2.1 Power source current standard characteristics

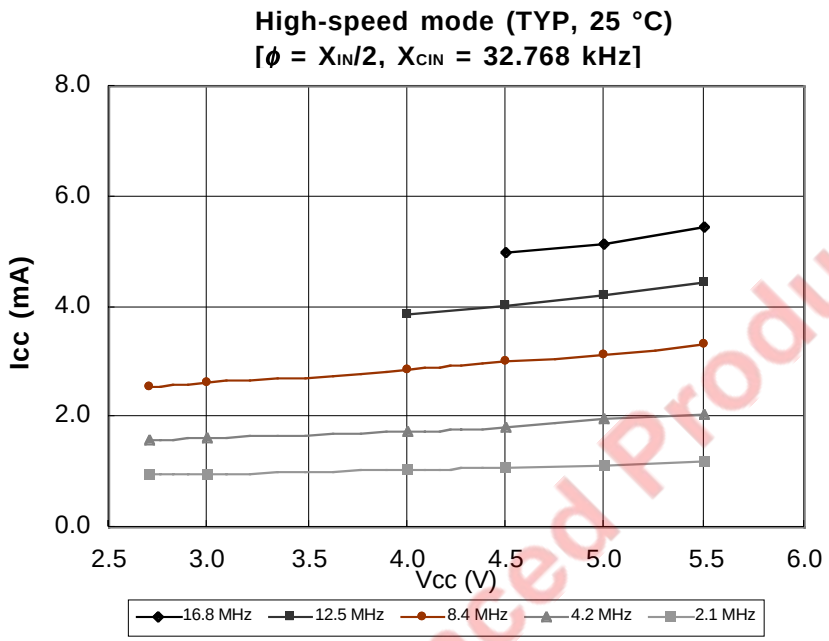


Fig. 3.2.1 Power source current standard characteristics (in high-speed mode)

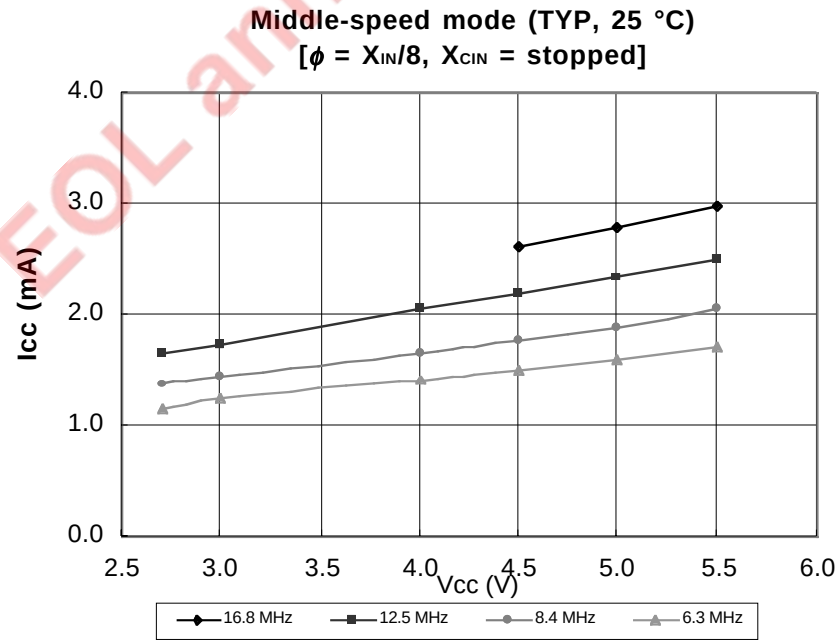


Fig. 3.2.2 Power source current standard characteristics (in middle-speed mode)

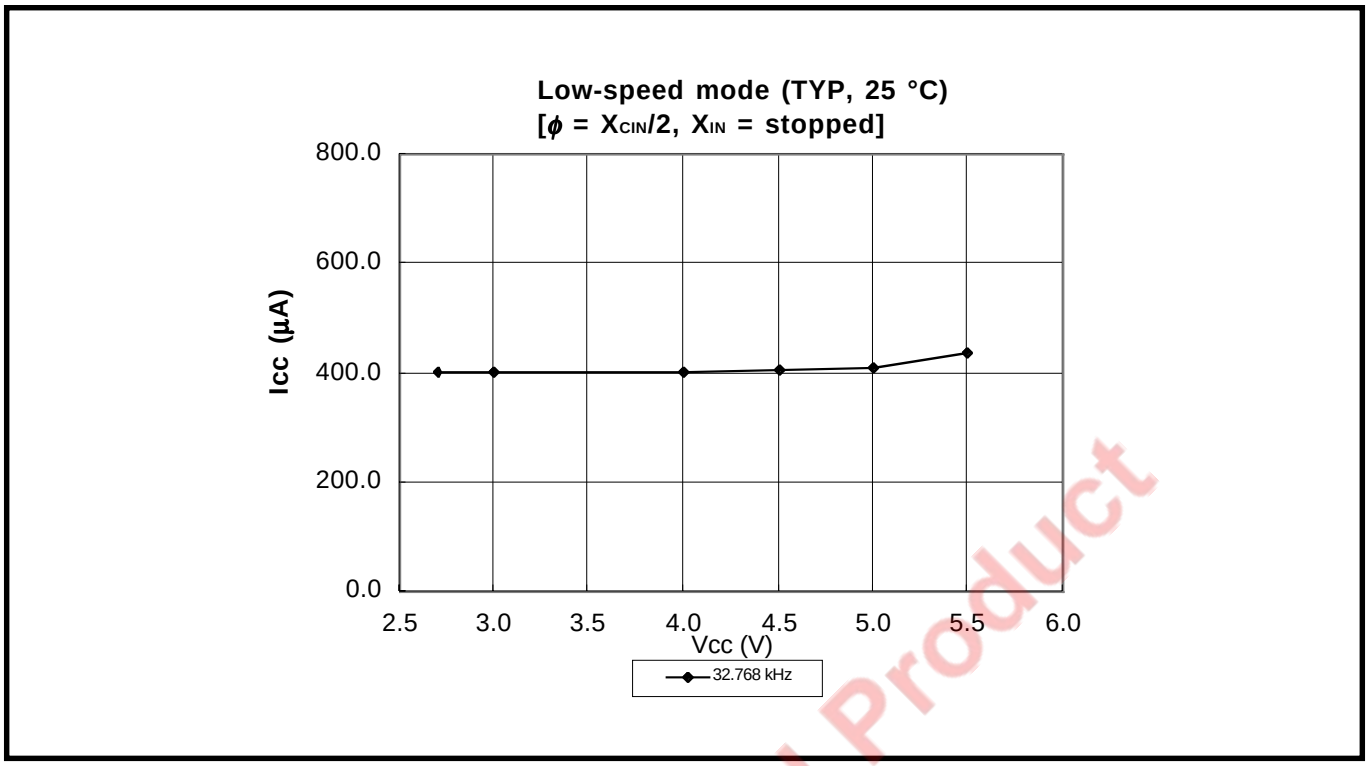


Fig. 3.2.3 Power source current standard characteristics (in low-speed mode)

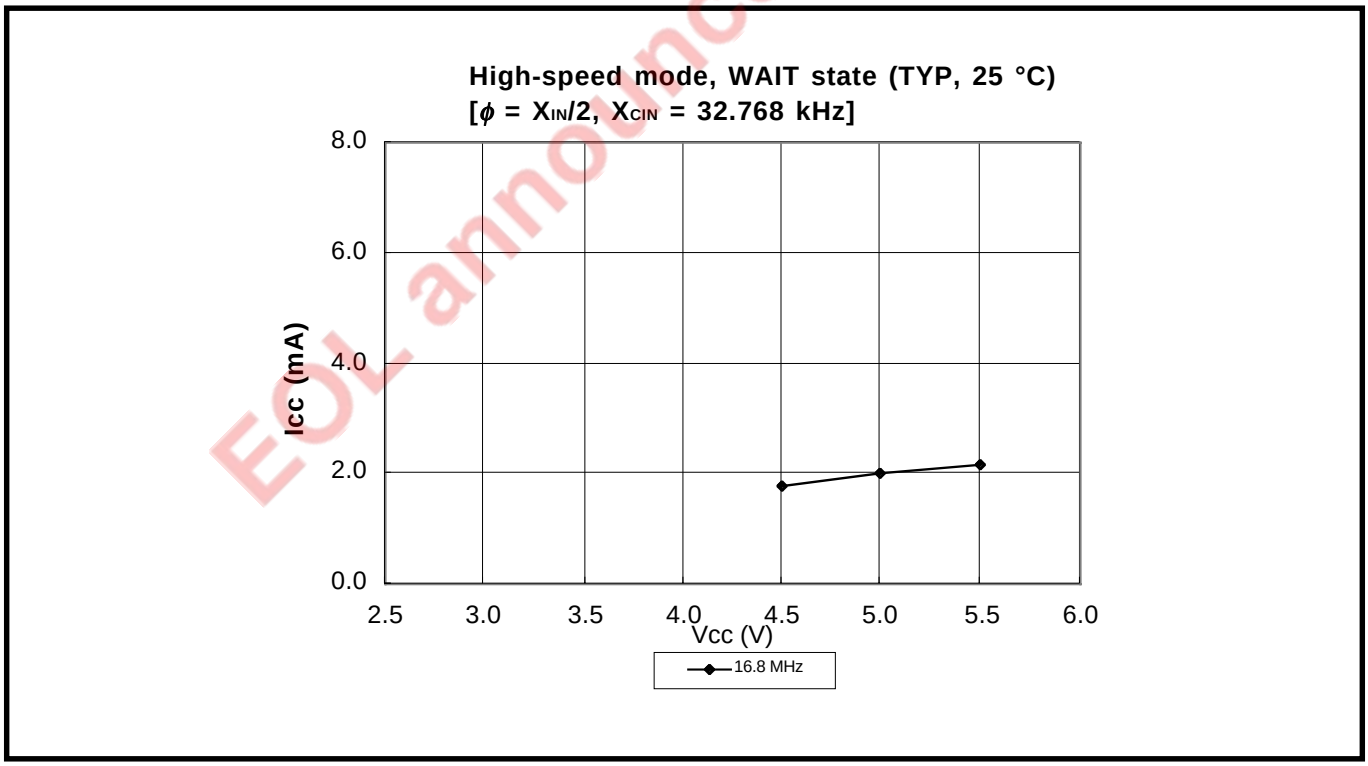


Fig. 3.2.4 Power source current standard characteristics (in high-speed mode, f(XIN) = 16.8 MHz, WAIT state)

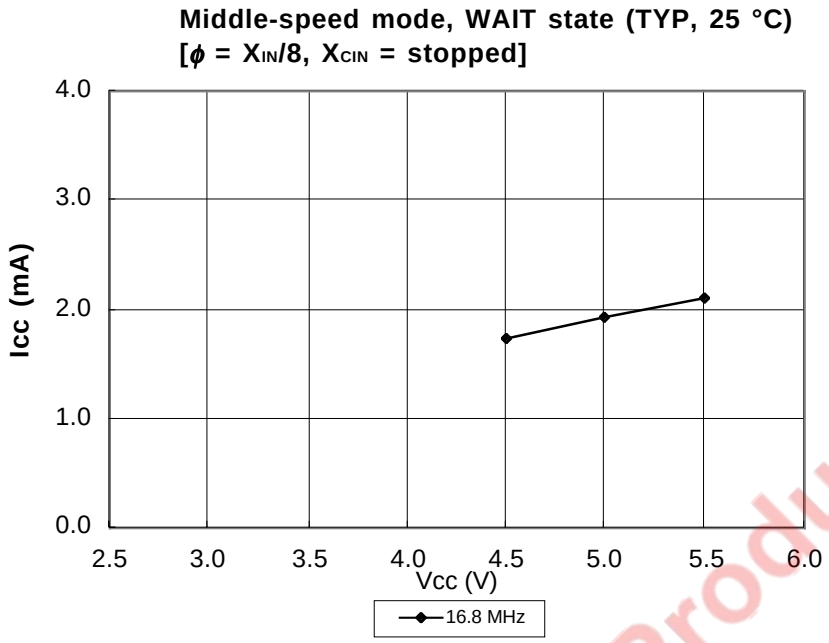


Fig. 3.2.5 Power source current standard characteristics (in middle-speed mode, f(XIN) = 16.8 MHz, WAIT state)

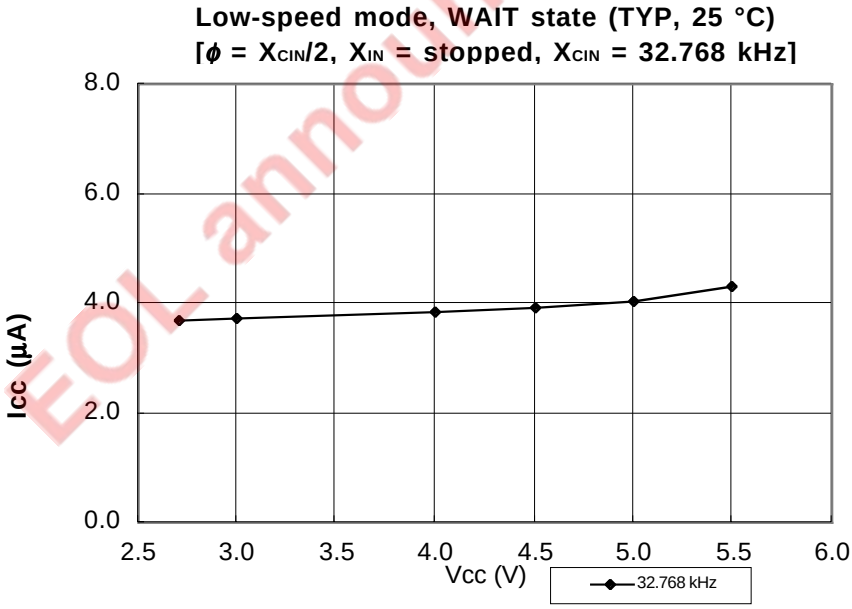


Fig. 3.2.6 Power source current standard characteristics (in low-speed mode, WAIT state)

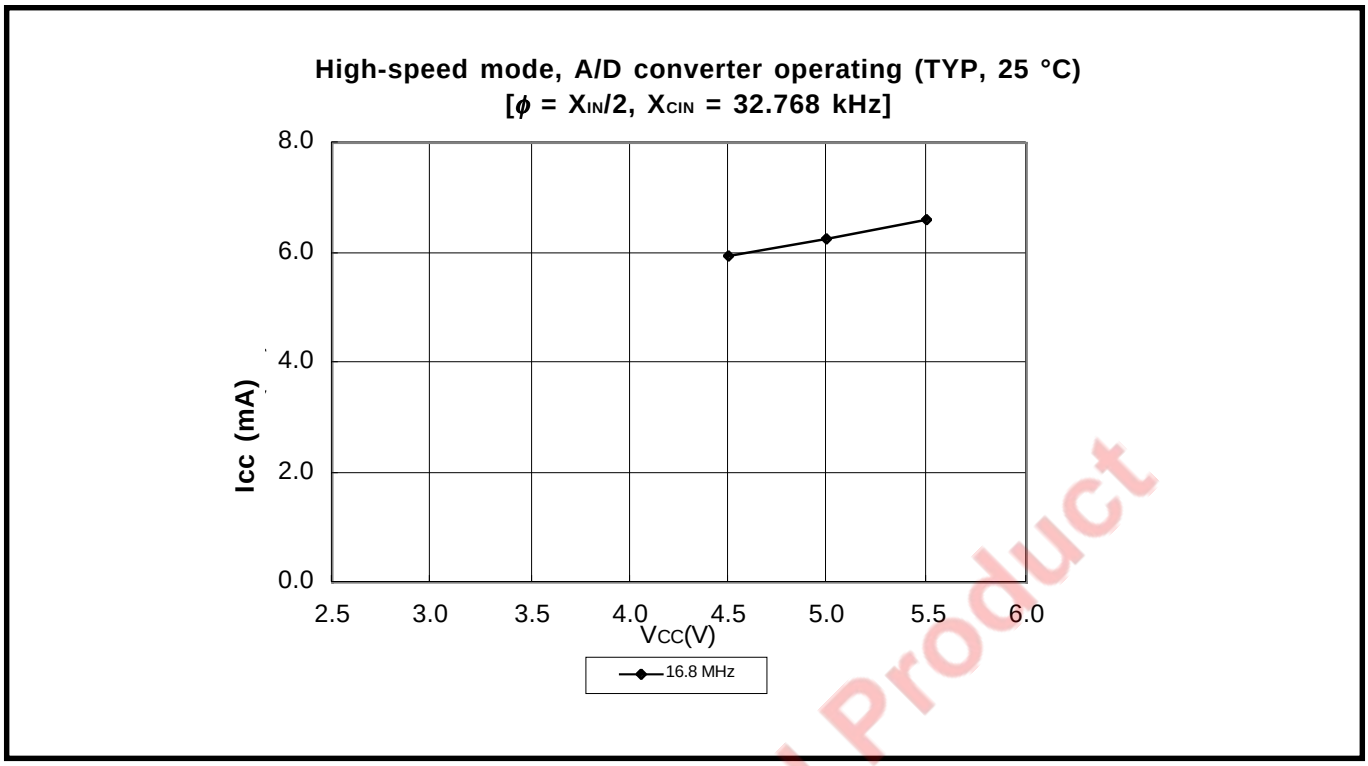


Fig. 3.2.7 Power source current standard characteristics (in high-speed mode, f(XIN) = 16.8 MHz, A/D converter operating)

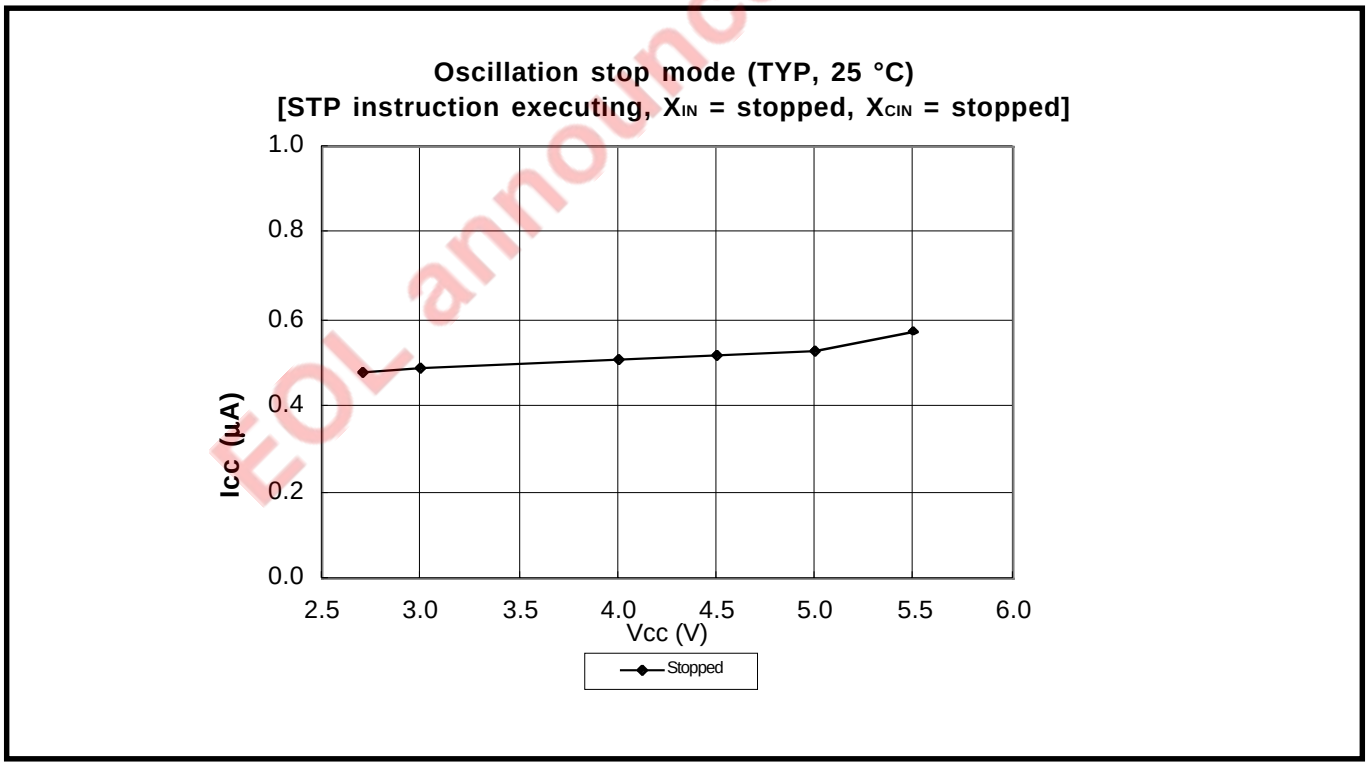


Fig. 3.2.8 Power source current standard characteristics (at oscillation stopping)

3.2.2 Port standard characteristics

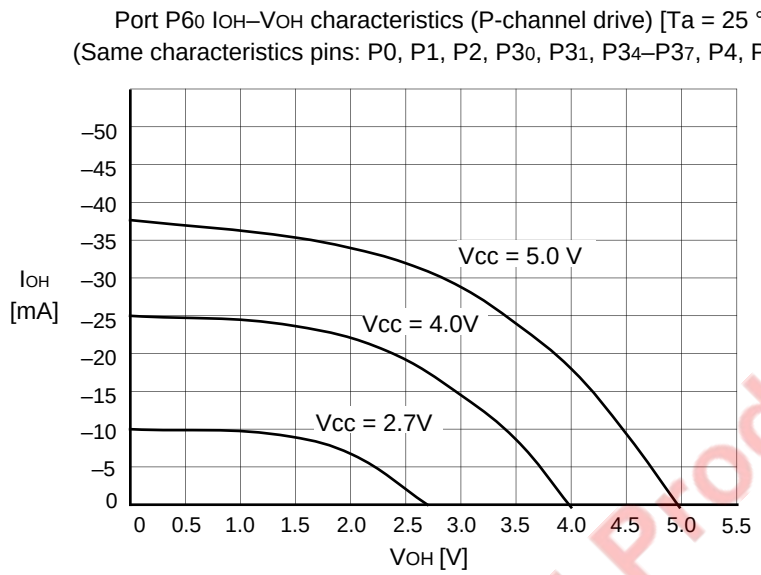


Fig. 3.2.9 CMOS output port P-channel side characteristics (Ta = 25 °C)

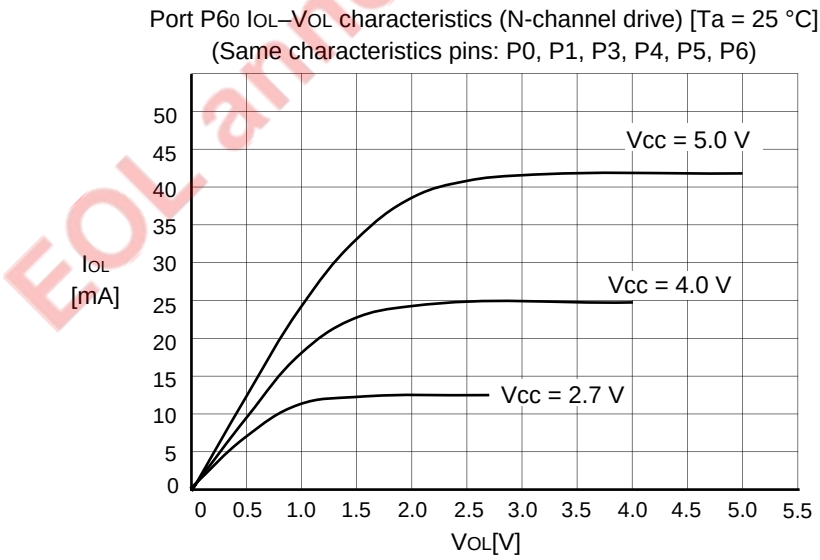


Fig. 3.2.10 CMOS output port N-channel side characteristics (Ta = 25 °C)

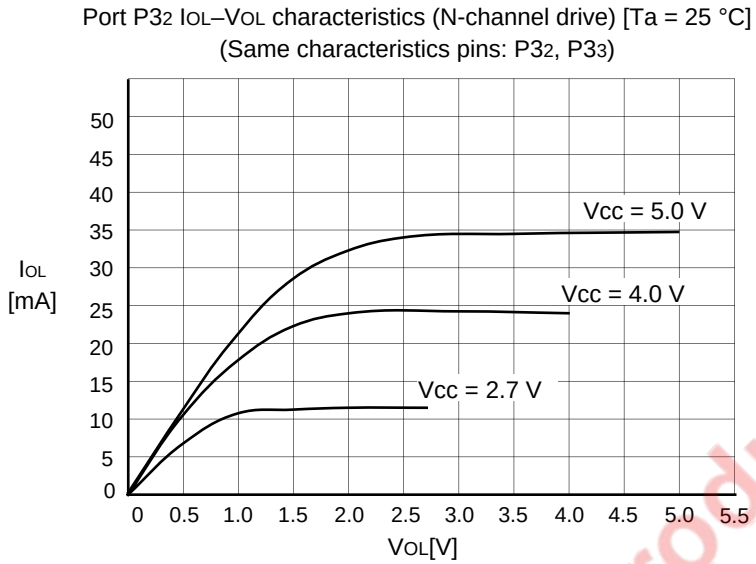


Fig. 3.2.11 N-channel open-drain output port N-channel side characteristics (Ta = 25 °C)

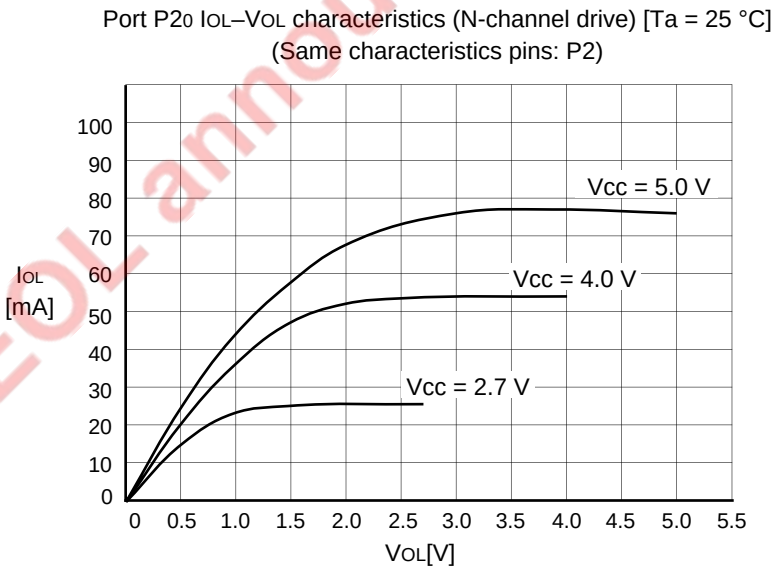


Fig. 3.2.12 CMOS large current output port N-channel side characteristics (Ta = 25 °C)

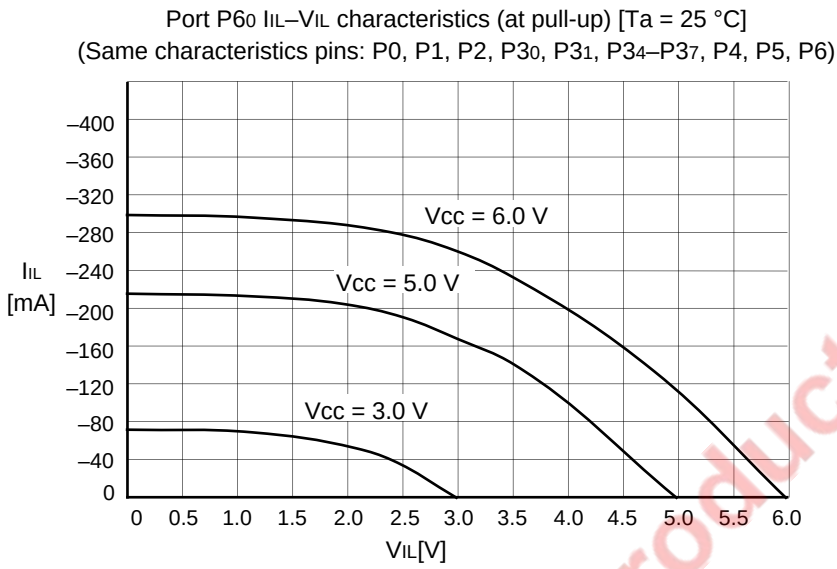


Fig. 3.2.13 CMOS input port at pull-up characteristics (Ta = 25 °C)

3.2.3 A/D conversion standard characteristics

Figure3.2.14, Figure 3.2.15, and Figure 3.2.16 show the A/D conversion standard characteristics.
The thick lines of the graph indicate the absolute precision errors, These are expressed as the deviation from the ideal value when the output code changes. For example, the change in output code from 512 to 513 should occur at 2560 mV, but the measured value is –10 mV. Accordingly, the measured point of change is 2560 – 10 = 2550 mV.

The thin lines of the graph indicate the input voltage width for which the output code is constant. For example, the measured input voltage width for which the output code is 512 is 5.0 mV, so that the differential non-linear error is 5.0 – 5.0 = 0.0 mV (0 LSB).

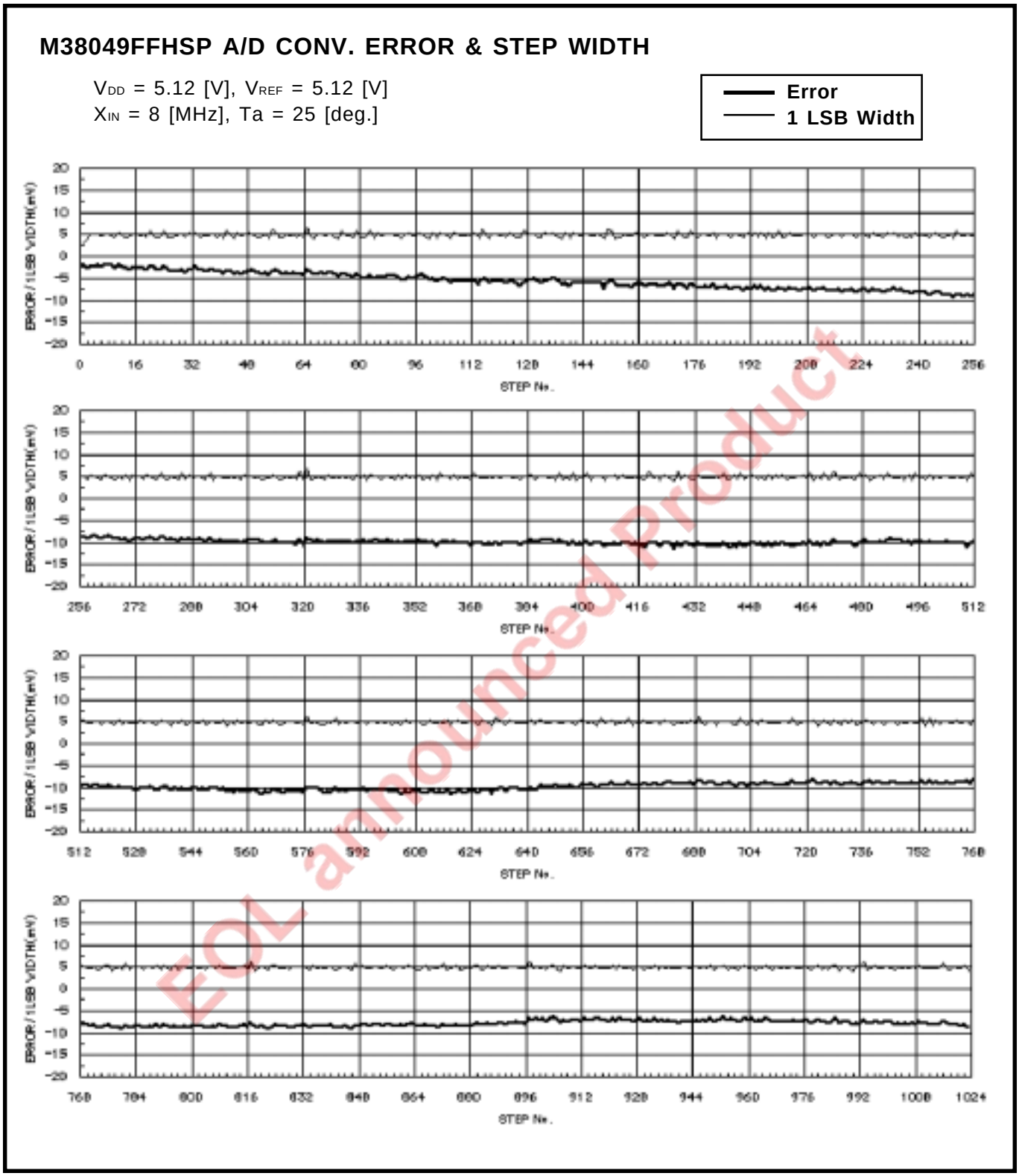


Fig. 3.2.14 A/D conversion standard characteristics (f(XIN) = 8 MHz)

M38049FFHSP A/D CONV. ERROR & STEP WIDTH

V_{DD} = 5.12 [V], V_{REF} = 5.12 [V]
X_{IN} = 12 [MHz], Ta = 25 [deg.]

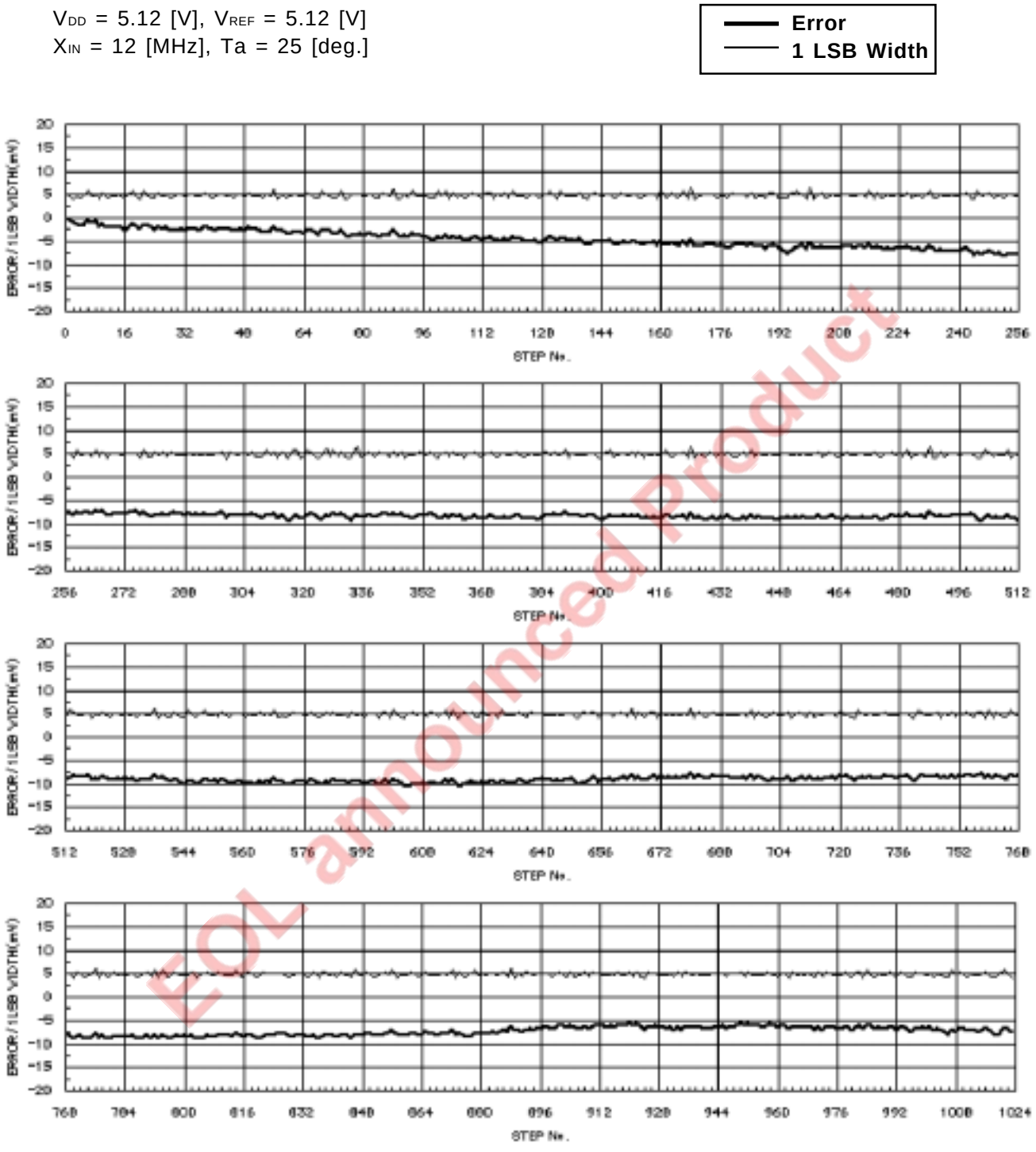


Fig. 3.2.15 A/D conversion standard characteristics (f(X_{IN}) = 12 MHz)

M38049FFHSP A/D CONV. ERROR & STEP WIDTH

V_{DD} = 5.12 [V], V_{REF} = 5.12 [V]
X_{IN} = 16 [MHz], Ta = 25 [deg.]

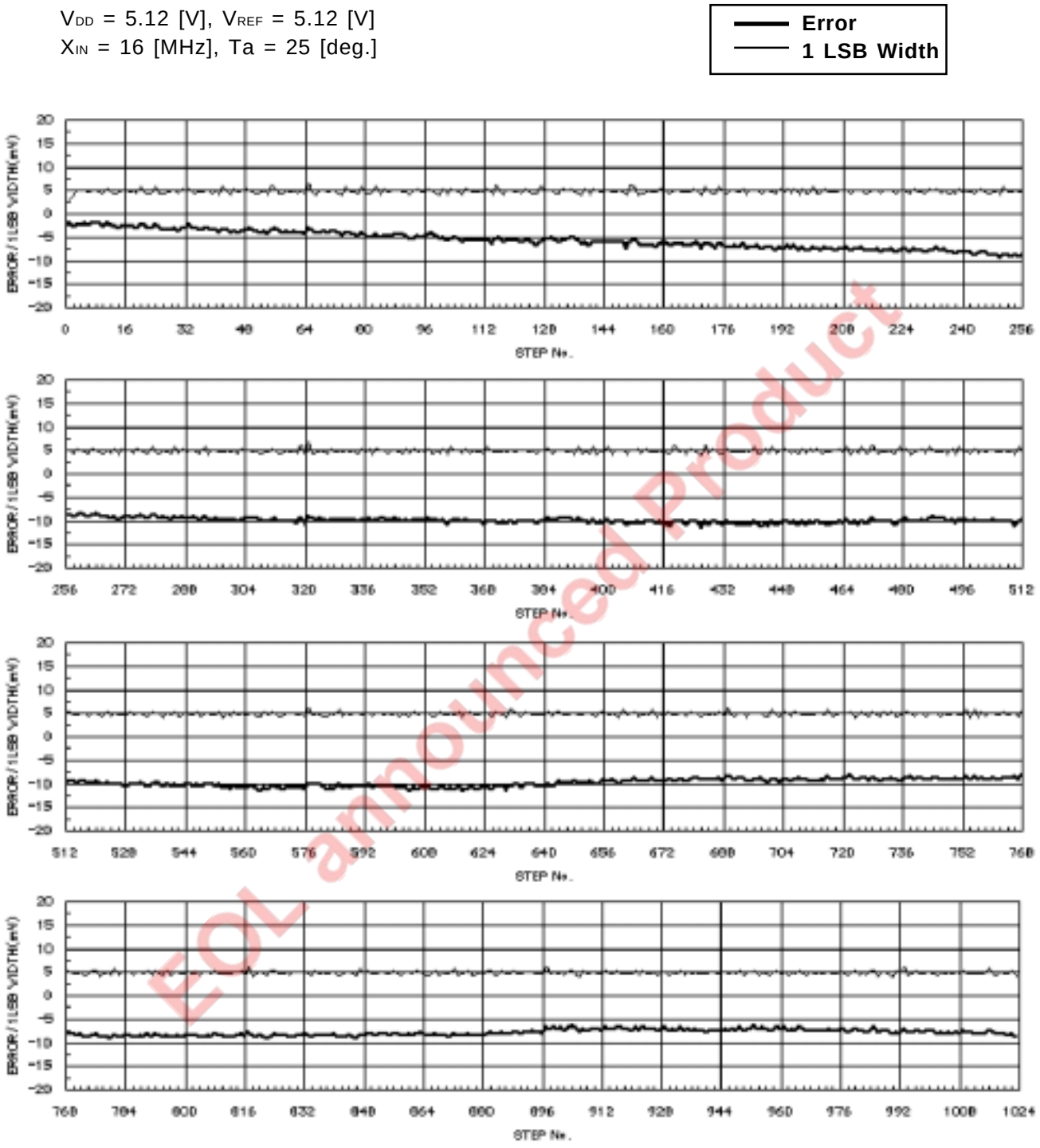


Fig. 3.2.16 A/D conversion standard characteristics (f(X_{IN}) = 16 MHz)

3.2.4 D/A conversion standard characteristics

Figure 3.2.17 shows the D/A conversion standard characteristics.

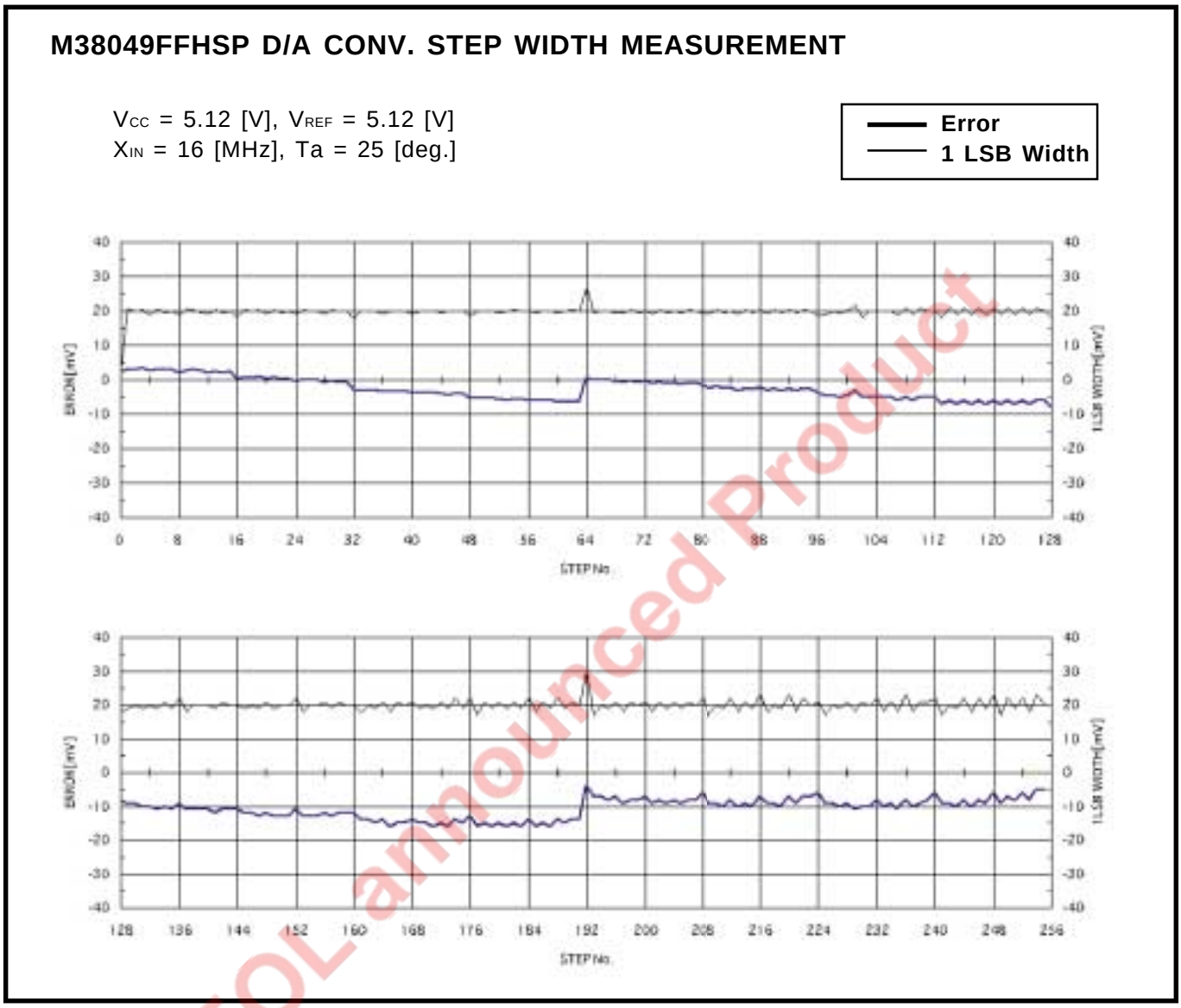


Fig. 3.2.17 D/A conversion standard characteristics

3.3 Notes on use

3.3.1 Notes on input and output ports

(1) Notes in standby state

In standby state*1 for low-power dissipation, do not make input levels of an I/O port “undefined”. Even when an I/O port of N-channel open-drain is set as output mode, if output data is “1”, the aforementioned notes are necessary.
Pull-up (connect the port to V_{CC}) or pull-down (connect the port to V_{SS}) these ports through a resistor.
When determining a resistance value, note the following points:
• External circuit
• Variation of output levels during the ordinary operation

When using built-in pull-up resistor, note on varied current values:
• When setting as an input port : Fix its input level
• When setting as an output port : Prevent current from flowing out to external

● **Reason**

Exclusive input ports are always in a high-impedance state. An output transistor becomes an OFF state when an I/O port is set as input mode by the direction register, so that the port enter a high-impedance state. At this time, the potential which is input to the input buffer in a microcomputer is unstable in the state that input levels are “undefined”. This may cause power source current. Even when an I/O port of N-channel open-drain is set as output mode by the direction register, if the contents of the port latch is “1”, the same phenomenon as that of an input port will occur.

✻1 standby state: Stop mode by executing **STP** instruction
Wait mode by executing **WIT** instruction

(2) Modifying output data with bit managing instruction

When the port latch of an I/O port is modified with the bit managing instruction*2, the value of the unspecified bit may be changed.

● **Reason**

The bit managing instructions are read-modify-write form instructions for reading and writing data by a byte unit. Accordingly, when these instructions are executed on a bit of the port latch of an I/O port, the following is executed to all bits of the port latch.
•As for bit which is set for input port:
The pin state is read in the CPU, and is written to this bit after bit managing.
•As for bit which is set for output port:
The bit value is read in the CPU, and is written to this bit after bit managing.

Note the following:
•Even when a port which is set as an output port is changed for an input port, its port latch holds the output data.
•As for a bit of which is set for an input port, its value may be changed even when not specified with a bit managing instruction in case where the pin state differs from its port latch contents.

✻2 Bit managing instructions: **SEB** and **CLB** instructions

3.3.2 Termination of unused pins

(1) Terminate unused pins

① I/O ports :
- Set the I/O ports for the input mode and connect them to Vcc or Vss through each resistor of 1 kΩ to 10 kΩ.
  Ports that permit the selecting of a built-in pull-up resistor can also use this resistor. Set the I/O ports for the output mode and open them at “L” or “H”.
- When opening them in the output mode, the input mode of the initial status remains until the mode of the ports is switched over to the output mode by the program after reset. Thus, the potential at these pins is undefined and the power source current may increase in the input mode. With regard to an effects on the system, thoroughly perform system evaluation on the user side.
- Since the direction register setup may be changed because of a program runaway or noise, set direction registers by program periodically to increase the reliability of program.

② The AVss pin when not using the A/D converter :
- When not using the A/D converter, handle a power source pin for the A/D converter, AVss pin as follows:
  AVss: Connect to the Vss pin.

(2) Termination remarks

① I/O ports :
Do not open in the input mode.
- **Reason**
  - The power source current may increase depending on the first-stage circuit.
  - An effect due to noise may be easily produced as compared with proper termination ① and shown on the above.

② I/O ports :
When setting for the input mode, do not connect to Vcc or Vss directly.
- **Reason**
  If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between a port and Vcc (or Vss).

③ I/O ports :
When setting for the input mode, do not connect multiple ports in a lump to Vcc or Vss through a resistor.
- **Reason**
  If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between ports.

- At the termination of unused pins, perform wiring at the shortest possible distance (20 mm or less) from microcomputer pins.

3.3.3 Notes on interrupts

(1) Change of relevant register settings

When the setting of the following registers or bits is changed, the interrupt request bit may be set to “1”. When not requiring the interrupt occurrence synchronized with these setting, take the following sequence.

•Interrupt edge selection register (address $003A_{16}$)
•Timer XY mode register (address 0023_{16})
•Timer Z mode register (address $002A_{16}$)
•I^2C START/STOP condition control register (address 0016_{16})

Set the above listed registers or bits as the following sequence.

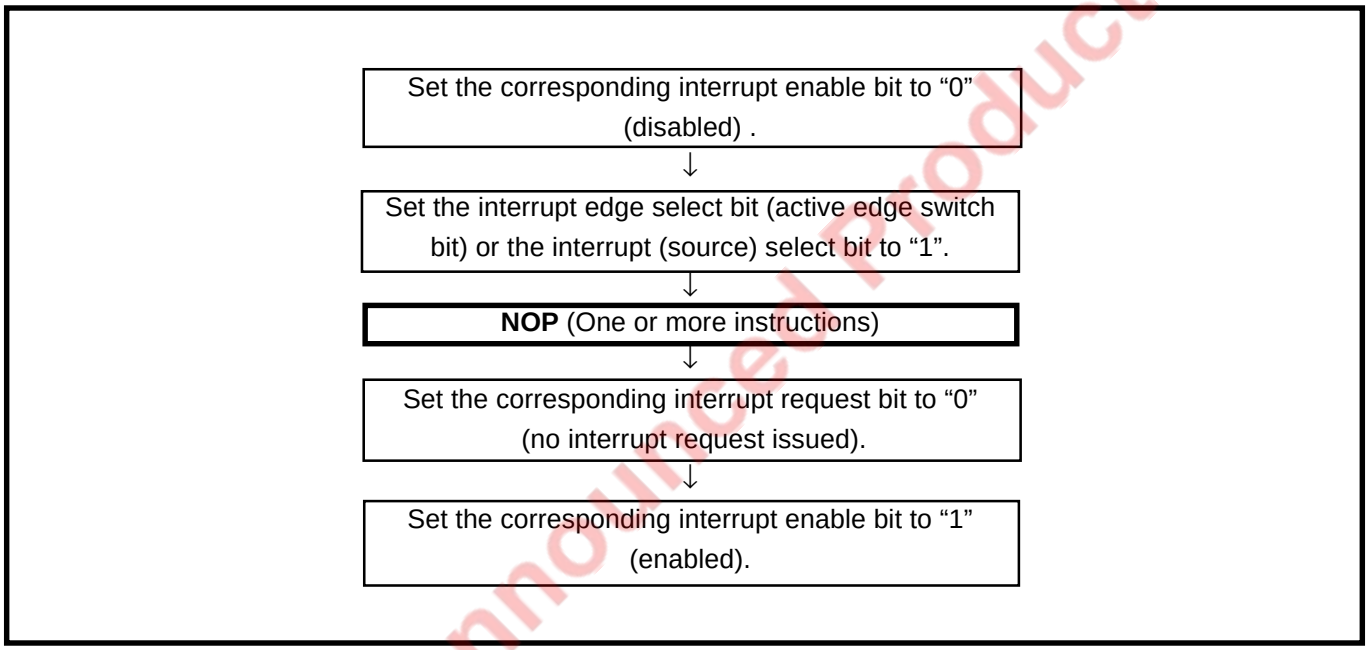


Fig. 3.3.1 Sequence of changing relevant register

■ **Reason**

When setting the followings, the interrupt request bit may be set to “1”.

•When setting external interrupt active edge
Concerned register: Interrupt edge selection register (address $003A_{16}$)
Timer XY mode register (address 0023_{16})
Timer Z mode register (address $002A_{16}$)
I^2C START/STOP condition control register (address 0016_{16})

•When switching interrupt sources of an interrupt vector address where two or more interrupt sources are allocated.
Concerned register: Interrupt source selection register (address 0039_{16})

(2) Check of interrupt request bit

● When executing the **BBC** or **BBS** instruction to an interrupt request bit of an interrupt request register immediately after this bit is set to “0”, execute one or more instructions before executing the **BBC** or **BBS** instruction.

■ **Reason**

If the BBC or BBS instruction is executed immediately after an interrupt request bit of an interrupt request register is cleared to “0”, the value of the interrupt request bit before being cleared to “0” is read.

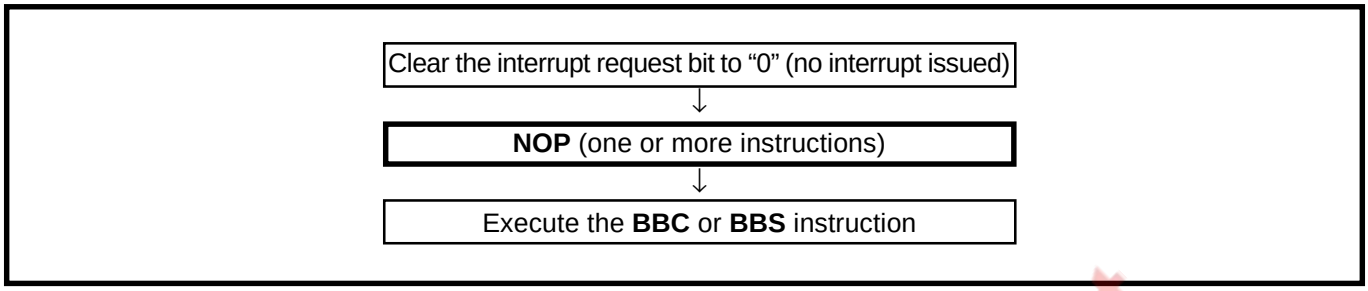


Fig. 3.3.2 Sequence of check of interrupt request bit

3.3.4 Notes on 8-bit timer (timer 1, 2, X, Y)

- If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is 1/(n+1).
- When switching the count source by the timer 12, X and Y count source selection bits, the value of timer count is altered in unconsiderable amount owing to generating of thin pulses in the count input signals.
  Therefore, select the timer count source before set the value to the prescaler and the timer.
- Set the double-function port of the $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to output in the pulse output mode.
- Set the double-function port of $CNTR_0$/$CNTR_1$ pin and port $P5_4$/$P5_5$ to input in the event counter mode and the pulse width measurement mode.

3.3.5 Notes on 16-bit timer (timer Z)

(1) Pulse output mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to output.

(2) Pulse period measurement mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to input.
- A read-out of timer value is impossible in this mode. The timer can be written to only during timer stop (no measurement of pulse period).
- Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operation during measurement.
- “$FFFF_{16}$” is set to the timer when the timer underflows or when the valid edge of measurement start/completion is detected. Consequently, the timer value at start of pulse period measurement depends on the timer value just before measurement start.

(3) Pulse width measurement mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to input.
- A read-out of timer value is impossible in this mode. The timer can be written to only during timer stop (no measurement of pulse period).
- Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operation during measurement.
- “$FFFF_{16}$” is set to the timer when the timer underflows or when the valid edge of measurement start/completion is detected. Consequently, the timer value at start of pulse width measurement depends on the timer value just before measurement start.

(4) Programmable waveform generating mode

- Set the double-function port of the $CNTR_2$ pin and port $P4_7$ to output.

(5) Programmable one-shot generating mode

- Set the double-function port of $CNTR_2$ pin and port $P4_7$ to output, and of INT_1 pin and port $P4_2$ to input in this mode.
- This mode cannot be used in low-speed mode.
- If the value of the $CNTR_2$ active edge switch bit is changed during one-shot generating enabled or generating one-shot pulse, then the output level from $CNTR_2$ pin changes.

(6) All modes

●Timer Z write control

Which write control can be selected by the timer Z write control bit (bit 3) of the timer Z mode register (address $002A_{16}$), writing data to both the latch and the timer at the same time or writing data only to the latch.

When the operation “writing data only to the latch” is selected, the value is set to the timer latch by writing data to the address of timer Z and the timer is updated at next underflow. After reset release, the operation “writing data to both the latch and the timer at the same time” is selected, and the value is set to both the latch and the timer at the same time by writing data to the address of timer Z.

In the case of writing data only to the latch, if writing data to the latch and an underflow are performed almost at the same time, the timer value may become undefined.

●Timer Z read control

A read-out of timer value is impossible in pulse period measurement mode and pulse width measurement mode. In the other modes, a read-out of timer value is possible regardless of count operating or stopped.

However, a read-out of timer latch value is impossible.

●Switch of interrupt active edge of $CNTR_2$ and INT_1

Each interrupt active edge depends on setting of the $CNTR_2$ active edge switch bit and the INT_1 active edge selection bit.

●Switch of count source

When switching the count source by the timer Z count source selection bits, the value of timer count is altered in inconsiderable amount owing to generating of thin pulses on the count input signals.

Therefore, select the timer count source before setting the value to the prescaler and the timer.

3.3.6 Notes on serial interface

(1) Notes when selecting clock synchronous serial I/O

① **Stop of transmission operation**

As for serial I/Oi (i = 1, 3) that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the serial I/Oi enable bit and the transmit enable bit to “0” (serial I/Oi and transmit disabled).

● **Reason**

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/Oi enable bit is cleared to “0” (serial I/Oi disabled), the internal transmission is running (in this case, since pins TxDi, RxDi, SCLKi, and $\overline{\text{SRDYi}}$ function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/Oi enable bit is set to “1” at this time, the data during internally shifting is output to the TxDi pin and an operation failure occurs.

② **Stop of receive operation**

As for serial I/Oi (i = 1, 3) that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the receive enable bit to “0” (receive disabled), or clear the serial I/Oi enable bit to “0” (serial I/Oi disabled).

③ **Stop of transmit/receive operation**

As for serial I/Oi (i = 1, 3) that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear both the transmit enable bit and receive enable bit to “0” (transmit and receive disabled).

(when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

● **Reason**

In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.

In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to “0” (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/Oi enable bit to “0” (serial I/Oi disabled) (refer to ① in (1)).

(2) Notes when selecting clock asynchronous serial I/O

① **Stop of transmission operation**
Clear the transmit enable bit to “0” (transmit disabled). The transmission operation does not stop by clearing the serial I/Oi enable bit (i = 1, 3) to “0”.

● **Reason**
This is the same as ① in (1).

② **Stop of receive operation**
Clear the receive enable bit to “0” (receive disabled).

③ **Stop of transmit/receive operation**
Only transmission operation is stopped.
Clear the transmit enable bit to “0” (transmit disabled). The transmission operation does not stop by clearing the serial I/Oi enable bit (i = 1, 3) to “0”.

● **Reason**
This is the same as ① in (1).

Only receive operation is stopped.
Clear the receive enable bit to “0” (receive disabled).

(3) $\overline{\text{SRDYi}}$ (i = 1, 3) output of reception side
When signals are output from the $\overline{\text{SRDYi}}$ pin on the reception side by using an external clock in the clock synchronous serial I/O mode, set all of the receive enable bit, the $\overline{\text{SRDYi}}$ output enable bit, and the transmit enable bit to “1” (transmit enabled).

(4) Setting serial I/Oi (i = 1, 3) control register again
Set the serial I/Oi control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to “0.”

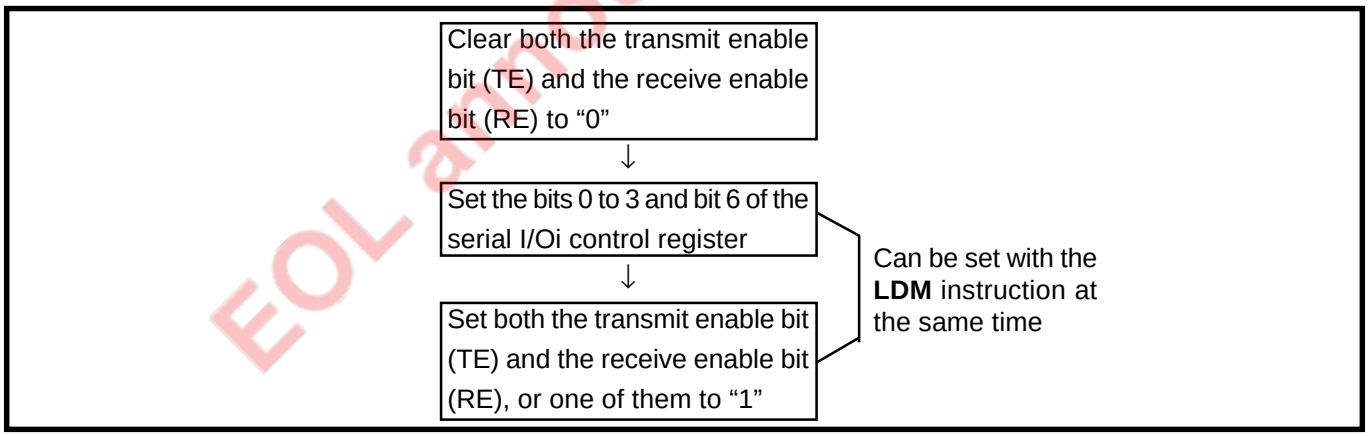


Fig. 3.3.3 Sequence of setting serial I/Oi (i = 1, 3) control register again

(5) Data transmission control with referring to transmit shift register completion flag
After the transmit data is written to the transmit buffer register, the transmit shift register completion flag changes from “1” to “0” with a delay of 0.5 to 1.5 shift clocks. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

(6) Transmission control when external clock is selected
When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to “1” at “H” of the SCLKi (i = 1, 3) input level. Also, write the transmit data to the transmit buffer register at “H” of the SCLKi input level.

(7) Transmit interrupt request when transmit enable bit is set

When using the transmit interrupt, take the following sequence.

① Set the serial I/Oi transmit interrupt enable bit (i = 1, 3) to "0" (disabled).
② Set the tranasmit enable bit to "1".
③ Set the serial I/Oi transmit interrupt request bit (i = 1, 3) to "0" after 1 or more instruction has executed.
④ Set the serial I/Oi transmit interrupt enable bit (i = 1, 3) to "1" (enabled).

● **Reason**

When the transmission enable bit is set to "1", the transmit buffer empty flag and transmit shift register shift completion flag are also set to "1".
Therefore, regardless of selecting which timing for the generating of transmit interrupts, the interrupt request is generated and the transmit interrupt request bit is set at this point.

(8) Writing to baud rate generator i (BRGi) (i = 1, 3)

Write data to the baud rate generator i (BRGi) (i = 1, 3) while the transmission/reception operation is stopped.

3.3.7 Notes on multi-master I²C-BUS interface

(1) Read-modify-write instruction

Each register of the multi-master I²C-BUS interface has bits to change by hardware. The precautions when the read-modify-write instruction such as **SEB**, **CLB** etc. is executed for each register of the multi-master I²C-BUS interface are described below.

① **I²C data shift register (S0: address 0011_{16})**
When executing the read-modify-write instruction for this register during transfer, data may become a value not intended.
② **I²C slave address registers 0 to 2 (S0D0 to S0D2: addresses $0FF7_{16}$ to $0FF9_{16}$)**
When the read-modify-write instruction is executed for this register at detecting the STOP condition, data may become a value not intended.
● **Reason**
It is because hardware changes the read/$\overline{\text{write}}$ bit (RWB) at detecting the STOP condition.
③ **I²C status register (S1: address 0013_{16})**
Do not execute the read-modify-write instruction for this register because all bits of this register are changed by hardware.
④ **I²C control register (S1D: address 0014_{16})**
When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become a value not intended.
● **Reason**
Because hardware changes the bit counter (BC0 to BC2).
⑤ **I²C clock control register (S2: address 0015_{16})**
The read-modify-write instruction can be executed for this register.
⑥ **I²C START/STOP condition control register (S2D: address 0016_{16})**
The read-modify-write instruction can be executed for this register.

(2) START condition generating procedure using multi-master

① Procedure example (The necessary conditions of the generating procedure are described as the following ② to ⑤).

```
        LDA #SLADR          (Taking out of slave address value)
        SEI                 (Interrupt disabled)
        BBS 5, S1, BUSBUSY  (BB flag confirming and branch process)
BUSFREE:
        STA S0              (Writing of slave address value)
        LDM #$F0, S1        (Trigger of START condition generating)
        CLI                 (Interrupt enabled)
        :
        :
BUSBUSY:
        CLI                 (Interrupt enabled)
        :
        :
```

② Use “Branch on Bit Set” of “BBS 5, S1, –” for the BB flag confirming and branch process.

③ Use “STA, STX” or “STY” of the zero page addressing instruction for writing the slave address value to the I²C data shift register (S0: address 0011_{16}).

④ Execute the branch instruction of above ② and the store instruction of above ③ continuously shown by the above procedure example.

⑤ Disable interrupts during the following three process steps:

- BB flag confirming
- Writing of slave address value
- Trigger of START condition generating

(3) RESTART condition generating procedure in master

① Procedure example (The necessary conditions of the generating procedure are described as the following ② to ④). Execute the following procedure when the PIN bit is “0”.

```
        LDM #$00, S1        (Select slave receive mode)
        LDA #SLADR          (Taking out of slave address value)
        SEI                 (Interrupt disabled)
        STA S0              (Writing of slave address value)
        LDM #$F0, S1        (Trigger of RESTART condition generating)
        CLI                 (Interrupt enabled)
        :
        :
```

② Select the slave receive mode when the PIN bit is “0”. Do not write “1” to the PIN bit. The TRX bit becomes “0” and the SDA pin is released.

③ The SCL pin is released by writing the slave address value to the I²C data shift register.

④ Disable interrupts during the following two process steps:

- Writing of slave address value

(4) Writing to I²C status register (S1: address 0013_{16})

Do not execute an instruction to set the PIN bit to “1” from “0” and an instruction to set the MST and TRX bits to “0” from “1” simultaneously. It is because it may enter the state that the SCL pin is released and the SDA pin is released after about one machine cycle. Do not execute an instruction to set the MST and TRX bits to “0” from “1” simultaneously when the PIN bit is “1”. It is because it may become the same as above.

(5) Writing to I²C clock control register (S2: address 0015_{16})

Do not write data into the I²C clock control register during transfer. If data is written during transfer, the I²C clock generator is reset, so that data cannot be transferred normally.

(6) Switching of SCL/SDA interrupt pin polarity selection bit, SCL/SDA interrupt pin selection bit, I²C-BUS interface enable bit

When changing the setting of the SCL/SDA interrupt pin polarity selection bit, the SCL/SDA interrupt pin selection bit, or the I²C-BUS interface enable bit ES0, the SCL/SDA interrupt request bit may be set. When selecting the SCL/SDA interrupt source, disable the interrupt before the SCL/SDA interrupt pin polarity selection bit, the SCL/SDA interrupt pin selection bit, or the I²C-BUS interface enable bit ES0 is set. Reset the request bit to "0" after setting these bits, and enable the interrupt.

(7) Process of after STOP condition generating in master mode

Do not write data in the I²C data shift register (S0) and the I²C status register (S1) until the bus busy flag BB becomes "0" after generating the STOP condition in the master mode. It is because the STOP condition waveform might not be normally generated. Reading to the above registers does not have the problem.

(8) ES0 bit switch

In standard clock mode when SSC = "00010_2" or in high-speed clock mode, flag BB may switch to "1" if ES0 bit is set to "1" when SDA is "L".

Countermeasure:

Set ES0 to "1" when SDA is "H".

- Trigger of RESTART condition generating

3.3.8 Notes on programming for SMBUS interface

(1) Time out process

For a smart battery system, the time out process with a program is required so that the communication can be completed even when communication is interrupted. It is because there is possibility of extracting a battery from a PC.

The specifications are defined so that communication has been able to be completed within 25 ms from START condition to STOP condition and within 10 ms from the ACK pulse to the ACK pulse of each byte. Accordingly, the following two should be considered as count start conditions.

① **SDA falling edge caused by SCL/SDA interrupt**

This is the countermeasure for a communication interrupt in the middle of from START condition to a slave address. However, the detection condition must be considered because a interrupt is also generated by communication from other masters to other slaves.

② **SMBUS interrupt after receiving slave address**

This is the countermeasure for when communication is interrupted from receiving a slave address until receiving a command.

(2) Low hold of communication line

The I²C-BUS interface conforms to the I²C-BUS Standard Specifications. However, because the use condition of SMBUS differs from the I²C-BUS's, there is possibility of occurrence of the following problem.

① **Low hold of SDA line caused by ACK pulse at voltage drop of communication line**

When the SMBUS voltage slowly drops, that is caused by extracting a battery from equipment or turning off a PC's power or etc., it might be incorrectly treated as the SCL pulse near the threshold level voltage.

When the SDA is judged "L" in that condition, it becomes the general call and the ACK is transmitted. However, when the SCL remains "L" at the ACK pulse, the SDA continuously remains "L" until input of the next SCL pulse.

Countermeasure:
As explained before, start the time out count at the falling of SDA line of START condition and reset ES0 bit of the S1D register when the time out is satisfied (**Note**).

Note: Do not use the read-modify-write instruction at this time. Furthermore, when the ES0 bit is set to "0", it becomes a general-purpose port ; so that the port must be set to input mode or "H".

3.3.9 Notes on PWM

The PWM starts from "H" level after the PWM enable bit is set to enable and "L" level is temporarily output from the PWM pin.
The length of this "L" level output is as follows:

$$\frac{n + 1}{2 \cdot f(X_{IN})} \text{ (s)} \quad \text{(Count source selection bit = "0", where n is the value set in the prescaler)}$$

$$\frac{n + 1}{f(X_{IN})} \text{ (s)} \quad \text{(Count source selection bit = "1", where n is the value set in the prescaler)}$$

3.3.10 Notes on A/D converter

(1) Analog input pin
Make the signal source impedance for analog input low, or equip an analog input pin with an external capacitor of 0.01 µF to 1 µF. Further, be sure to verify the operation of application products on the user side.

● **Reason**
An analog input pin includes the capacitor for analog voltage comparison. Accordingly, when signals from signal source with high impedance are input to an analog input pin, charge and discharge noise generates. This may cause the A/D conversion precision to be worse.

(2) A/D converter power source pin
The AVSS pin is A/D converter power source pins. Regardless of using the A/D conversion function or not, connect it as following :
• AVSS : Connect to the VSS line

● **Reason**
If the AVSS pin is opened, the microcomputer may have a failure because of noise or others.

(3) Clock frequency during A/D conversion
The comparator consists of a capacity coupling, and a charge of the capacity will be lost if the clock frequency is too low. Thus, make sure the following during an A/D conversion.
• f(XIN) is 500 kHz or more
• Do not execute the **STP** instruction

(4) Difference between at 8-bit reading in 10-bit A/D mode and at 8-bit A/D mode
At 8-bit reading in the 10-bit A/D mode, "–1/2 LSB" correction is not performed to the A/D conversion result.
In the 8-bit A/D mode, the A/D conversion characteristics is the same as 3802 group's characteristics because "–1/2 LSB" correction is performed.

3.3.11 Notes on D/A converter

(1) Vcc when using D/A converter

The D/A converter accuracy when Vcc is 4.0 V or less differs from that of when Vcc is 4.0 V or more. When using the D/A converter, we recommend using a Vcc of 4.0 V or more.

(2) DAi conversion register when not using D/A converter

When a D/A converter is not used, set all values of the DAi conversion registers (i = 1, 2) to "00_{16}". The initial value after reset is "00_{16}".

3.3.12 Notes on watchdog timer

- Make sure that the watchdog timer H does not underflow while waiting Stop release, because the watchdog timer keeps counting during that term.
- When the STP instruction disable bit has been set to "1", it is impossible to switch it to "0" by a program.

3.3.13 Notes on $\overline{\text{RESET}}$ pin

Connecting capacitor

In case where the $\overline{\text{RESET}}$ signal rise time is long, connect a ceramic capacitor or others across the $\overline{\text{RESET}}$ pin and the Vss pin. Use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, note the following :

- Make the length of the wiring which is connected to a capacitor as short as possible.
- Be sure to verify the operation of application products on the user side.

● **Reason**

If the several nanosecond or several ten nanosecond impulse noise enters the $\overline{\text{RESET}}$ pin, it may cause a microcomputer failure.

3.3.14 Notes on low-speed operation mode

(1) Using sub-clock

To use a sub-clock, fix bit 3 of the CPU mode register to "1" or control the Rd (refer to Figure 3.3.4) resistance value to a certain level to stabilize an oscillation. For resistance value of Rd, consult the oscillator manufacturer.

● **Reason**

When bit 3 of the CPU mode register is set to "0", the sub-clock oscillation may stop.

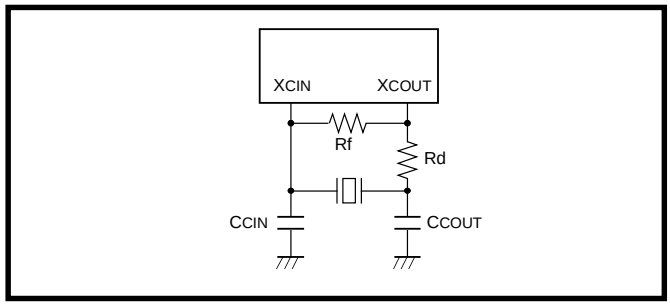


Fig. 3.3.4 Ceramic resonator circuit

(2) Switch between middle/high-speed mode and low-speed mode

If you switch the mode between middle/high-speed and low-speed, stabilize both X_{IN} and X_{CIN} oscillations. The sufficient time is required for the sub clock to stabilize, especially immediately after power on and at returning from stop mode. When switching the mode between middle/high-speed and low-speed, set the frequency on condition that $f(X_{IN}) > 3f(X_{CIN})$.

3.3.15 Quartz-crystal oscillator

When using the quartz-crystal oscillator of high frequency, such as 16 MHz etc., it may be necessary to select a specific oscillator with the specification demanded.

3.3.16 Notes on restarting oscillation

(1) Restarting oscillation

Usually, when the MCU stops the clock oscillation by STP instruction and the STP instruction has been released by an external interrupt source, the fixed values of Timer 1 and Prescaler 12 (Timer 1 = "01_{16}", Prescaler 12 = "FF_{16}") are automatically reloaded in order for the oscillation to stabilize. The user can inhibit the automatic setting by writing "1" to bit 0 of MISRG (address 0010_{16}).
However, by setting this bit to "1", the previous values, set just before the STP instruction was executed, will remain in Timer 1 and Prescaler 12. Therefore, you will need to set an appropriate value to each register, in accordance with the oscillation stabilizing time, before executing the STP instruction.

● **Reason**

Oscillation will restart when an external interrupt is received. However, internal clock ϕ is supplied to the CPU only when Timer 1 starts to underflow. This ensures time for the clock oscillation using the ceramic resonators to be stabilized.

3.3.17 Notes on using stop mode

■Register setting

Since values of the prescaler 12 and Timer 1 are automatically reloaded when returning from the stop mode, set them again, respectively. (When the oscillation stabilizing time set after STP instruction released bit is "0")

■Clock restoration

After restoration from the stop mode to the normal mode by an interrupt request, the contents of the CPU mode register previous to the STP instruction execution are retained. Accordingly, if both main clock and sub clock were oscillating before execution of the STP instruction, the oscillation of both clocks is resumed at restoration.
In the above case, when the main clock side is set as a system clock, the oscillation stabilizing time until the timer 1 underflow is reserved at restoration from the stop mode.
When the oscillation stabilizing time set after STP instruction released bit is "0", the time for 512 counts of the count source become the oscillation stabilizing time. When the oscillation stabilizing time set after STP instruction released bit is "1", an arbitrarily count value set to the prescaler 12 and the timer 1 become the oscillation stabilizing time.
At this time, note that the oscillation on the sub clock side may not be stabilized even after the lapse of the oscillation stabilizing time of the main clock side.

3.3.18 Notes on wait mode

■**Clock restoration**

If the wait mode is released by a reset when X_{CIN} is set as the system clock and X_{IN} oscillation is stopped during execution of the WIT instruction, X_{CIN} oscillation stops, X_{IN} oscillations starts, and X_{IN} is set as the system clock.

In the above case, the RESET pin should be held at "L" until the oscillation is stabilized.

3.3.19 Notes on CPU rewrite mode

(1) Operation speed

During CPU rewrite mode, set the system clock ϕ 4.0 MHz or less using the main clock division ratio selection bits (bits 6 and 7 of address $003B_{16}$).

(2) Instructions inhibited against use

The instructions which refer to the internal data of the flash memory cannot be used during the CPU rewrite mode.

(3) Interrupts inhibited against use

The interrupts cannot be used during the CPU rewrite mode because they refer to the internal data of the flash memory.

(4) Watchdog timer

In case of the watchdog timer has been running already, the internal reset generated by watchdog timer underflow does not happen, because of watchdog timer is always clearing during program or erase operation.

(5) Reset

Reset is always valid. In case of CNV_{SS} = "H" when reset is released, boot mode is active. So the program starts from the address contained in address $FFFC_{16}$ and $FFFD_{16}$ in boot ROM area.

3.3.20 Notes on programming

(1) Processor status register

① **Initializing of processor status register**

Flags which affect program execution must be initialized after a reset.

In particular, it is essential to initialize the T and D flags because they have an important effect on calculations.

● **Reason**

After a reset, the contents of the processor status register (PS) are undefined except for the I flag which is "1".

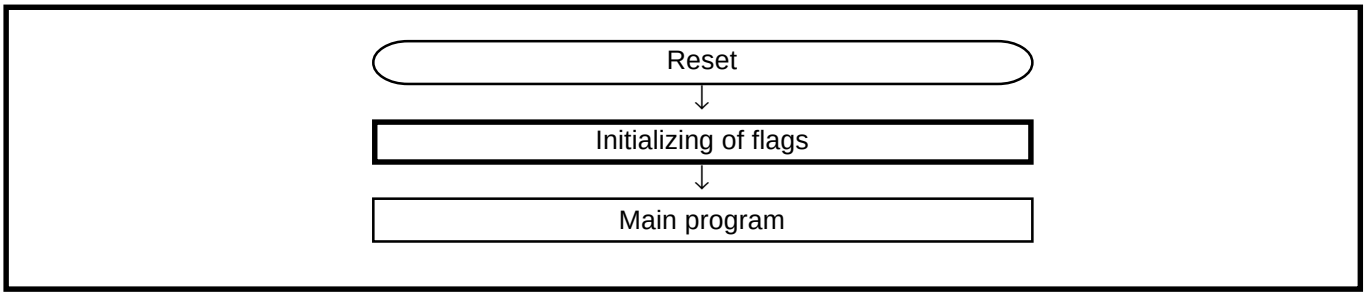


Fig. 3.3.5 Initialization of processor status register

② **How to reference the processor status register**

To reference the contents of the processor status register (PS), execute the **PHP** instruction once then read the contents of (S+1). If necessary, execute the **PLP** instruction to return the PS to its original status.

A **NOP** instruction should be executed after every **PLP** instruction.

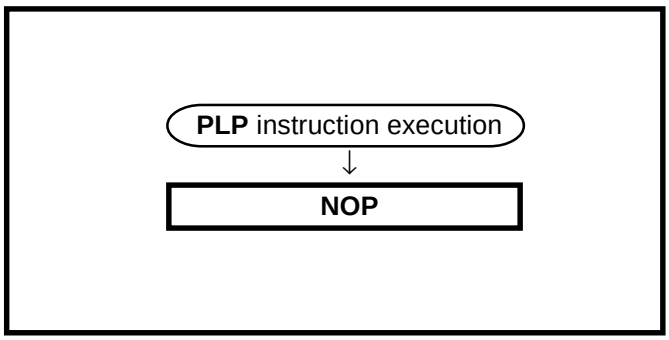


Fig. 3.3.6 Sequence of PLP instruction execution

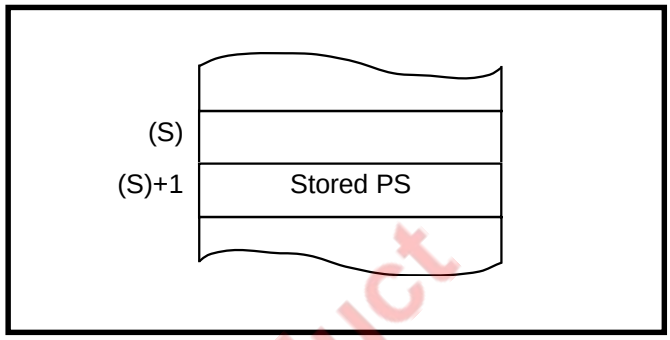


Fig. 3.3.7 Stack memory contents after PHP instruction execution

(2) BRK instruction

① **Interrupt priority level**

When the BRK instruction is executed with the following conditions satisfied, the interrupt execution is started from the address of interrupt vector which has the highest priority.

- Interrupt request bit and interrupt enable bit are set to “1”.
- Interrupt disable flag (I) is set to “1” to disable interrupt.

(3) Decimal calculations

① **Execution of decimal calculations**

The **ADC** and **SBC** are the only instructions which will yield proper decimal notation, set the decimal mode flag (D) to “1” with the **SED** instruction. After executing the **ADC** or **SBC** instruction, execute another instruction before executing the **SEC**, **CLC**, or **CLD** instruction.

② **Notes on status flag in decimal mode**

When decimal mode is selected, the values of three of the flags in the status register (the N, V, and Z flags) are invalid after a **ADC** or **SBC** instruction is executed.

The carry flag (C) is set to “1” if a carry is generated as a result of the calculation, or is cleared to “0” if a borrow is generated. To determine whether a calculation has generated a carry, the C flag must be initialized to “0” before each calculation. To check for a borrow, the C flag must be initialized to “1” before each calculation.

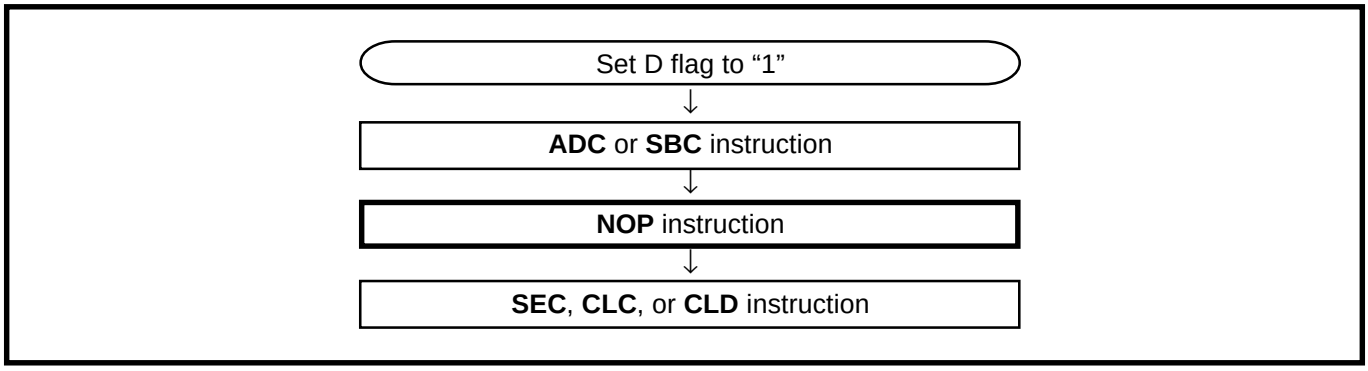


Fig. 3.3.8 Status flag at decimal calculations

(4) JMP instruction

When using the **JMP** instruction in indirect addressing mode, do not specify the last address on a page as an indirect address.

(5) Multiplication and Division Instructions

- The index X mode (T) and the decimal mode (D) flags do not affect the MUL and DIV instruction.
- The execution of these instructions does not change the contents of the processor status register.

(6) Ports

The contents of the port direction registers cannot be read. The following cannot be used:

- The data transfer instruction (LDA, etc.)
- The operation instruction when the index X mode flag (T) is “1”
- The instruction with the addressing mode which uses the value of a direction register as an index
- The bit-test instruction (BBC or BBS, etc.) to a direction register
- The read-modify-write instructions (ROR, CLB, or SEB, etc.) to a direction register.

Use instructions such as LDM and STA, etc., to set the port direction registers.

3.3.21 Notes on flash memory version

The CNVss pin determines the flash memory mode. To improve the noise reduction, connect a track between CNVss pin and Vss pin or Vcc pin with 1 to 10km resistance. The mask ROM version track of CNVss pin has no operational interference even if it is connected to Vss pin or Vcc pin via a resistor.

3.3.22 Notes on electric characteristic differences between mask ROM and flash nemory version MCUs

There are differences in electric characteristics, operation margin, noise immunity, and noise radiation between Mask ROM and Flash Memory version MCUs due to the difference in the manufacturing processes, built-in ROM, and layout pattern etc. When manufacturing an application system with the Flash Memory version and then switching to use of the Mask ROM version, please conduct evaluations equivalent to the system evaluations conducted for the flash memory version.

3.3.23 Notes on handling of power source pins

In order to avoid a latch-up occurrence, connect a capacitor suitable for high frequencies as bypass capacitor between power source pin (Vcc pin) and GND pin (Vss pin), and between power source pin (Vcc pin) and analog power source input pin (AVss pin). Besides, connect the capacitor to as close as possible. For bypass capacitor which should not be located too far from the pins to be connected, a ceramic capacitor of 0.01 μF–0.1 μF is recommended.

3.3.24 Power Source Voltage

When the power source voltage value of a microcomputer is less than the value which is indicated as the recommended operating conditions, the microcomputer does not operate normally and may perform unstable operation.

In a system where the power source voltage drops slowly when the power source voltage drops or the power supply is turned off, reset a microcomputer when the power source voltage is less than the recommended operating conditions and design a system not to cause errors to the system by this unstable operation.

3.4 Countermeasures against noise

Countermeasures against noise are described below. The following countermeasures are effective against noise in theory, however, it is necessary not only to take measures as follows but to evaluate before actual use.

3.4.1 Shortest wiring length

The wiring on a printed circuit board can function as an antenna which feeds noise into the microcomputer. The shorter the total wiring length (by mm unit), the less the possibility of noise insertion into a microcomputer.

(1) Wiring for $\overline{\text{RESET}}$ pin

Make the length of wiring which is connected to the $\overline{\text{RESET}}$ pin as short as possible. Especially, connect a capacitor across the $\overline{\text{RESET}}$ pin and the V_{SS} pin with the shortest possible wiring (within 20 mm).

● **Reason**

The width of a pulse input into the $\overline{\text{RESET}}$ pin is determined by the timing necessary conditions. If noise having a shorter pulse width than the standard is input to the $\overline{\text{RESET}}$ pin, the reset is released before the internal state of the microcomputer is completely initialized. This may cause a program runaway.

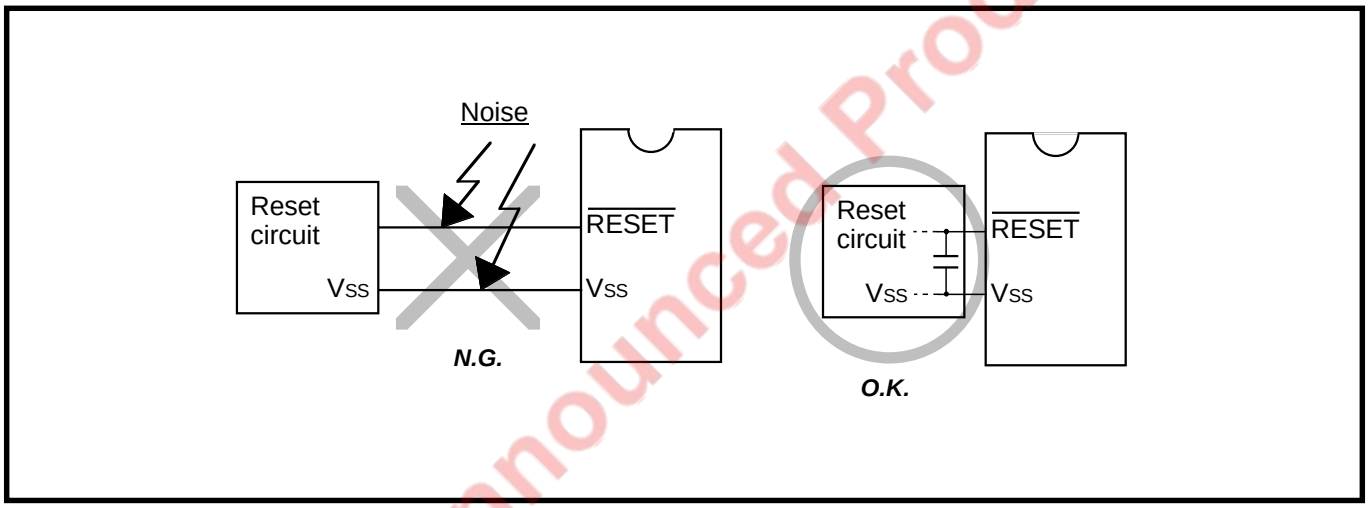


Fig. 3.4.1 Wiring for the $\overline{\text{RESET}}$ pin

(2) Wiring for clock input/output pins

- Make the length of wiring which is connected to clock I/O pins as short as possible.
- Make the length of wiring (within 20 mm) across the grounding lead of a capacitor which is connected to an oscillator and the Vss pin of a microcomputer as short as possible.
- Separate the Vss pattern only for oscillation from other Vss patterns.

● **Reason**

If noise enters clock I/O pins, clock waveforms may be deformed. This may cause a program failure or program runaway. Also, if a potential difference is caused by the noise between the Vss level of a microcomputer and the Vss level of an oscillator, the correct clock will not be input in the microcomputer.

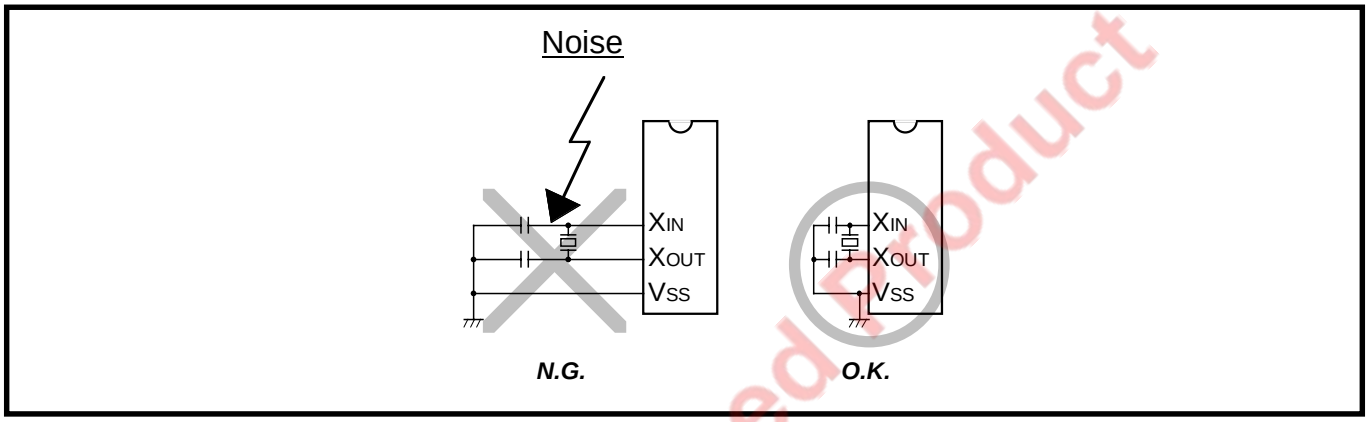


Fig. 3.4.2 Wiring for clock I/O pins

(3) Wiring to CNVss pin

Connect the CNVss pin to the Vss pin with the shortest possible wiring.

● **Reason**

The processor mode of a microcomputer is influenced by a potential at the CNVss pin. If a potential difference is caused by the noise between pins CNVss and Vss, the processor mode may become unstable. This may cause a microcomputer malfunction or a program runaway.

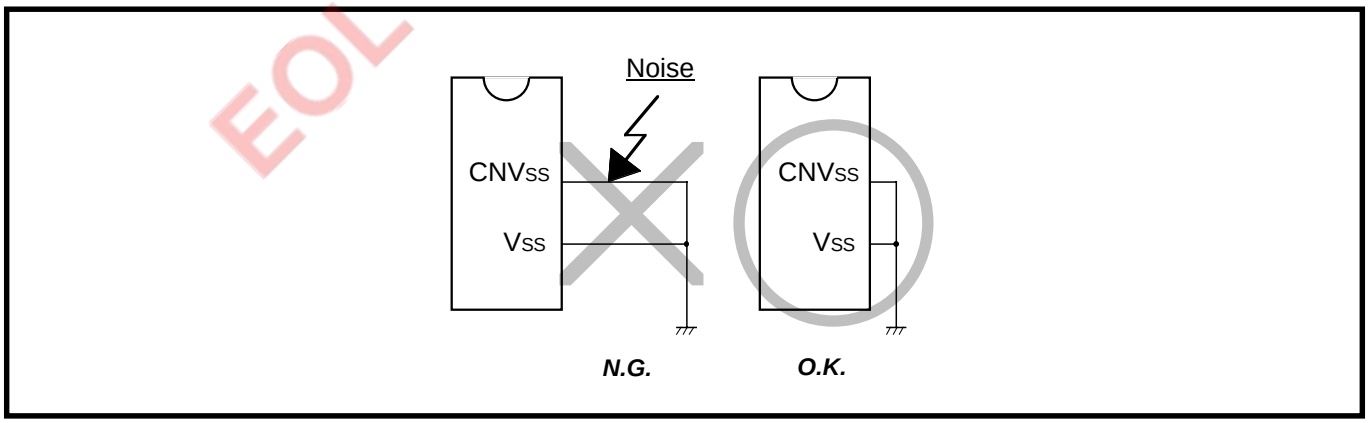


Fig. 3.4.3 Wiring for CNVss pin

3.4.2 Connection of bypass capacitor across Vss line and Vcc line

Connect an approximately 0.1 μF bypass capacitor across the Vss line and the Vcc line as follows:

- Connect a bypass capacitor across the Vss pin and the Vcc pin at equal length.
- Connect a bypass capacitor across the Vss pin and the Vcc pin with the shortest possible wiring.
- Use lines with a larger diameter than other signal lines for Vss line and Vcc line.
- Connect the power source wiring via a bypass capacitor to the Vss pin and the Vcc pin.

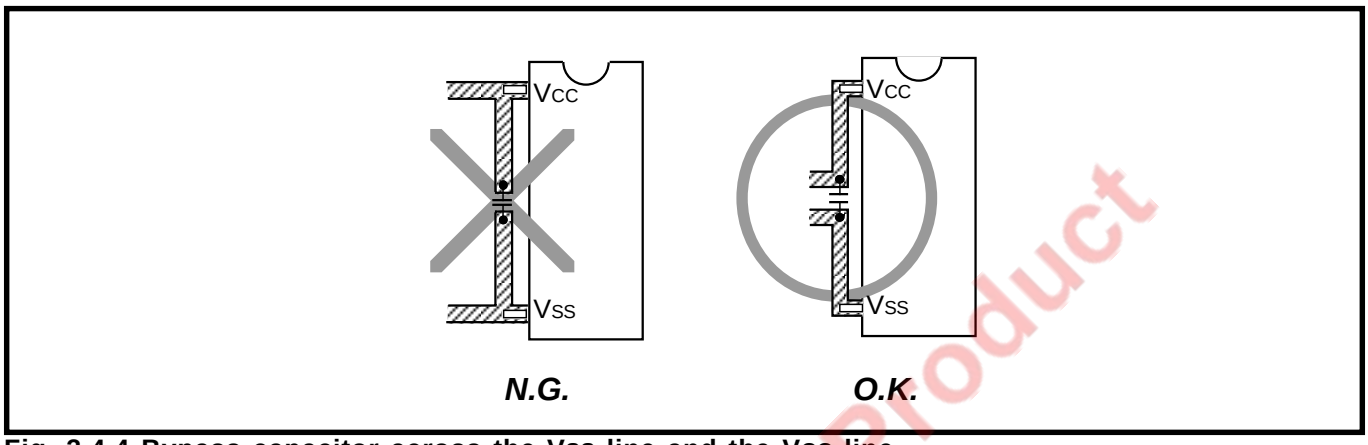


Fig. 3.4.4 Bypass capacitor across the Vss line and the Vcc line

3.4.3 Wiring to analog input pins

- Connect an approximately 100 Ω to 1 kΩ resistor to an analog signal line which is connected to an analog input pin in series. Besides, connect the resistor to the microcomputer as close as possible.
- Connect an approximately 1000 pF capacitor across the Vss pin and the analog input pin. Besides, connect the capacitor to the Vss pin as close as possible. Also, connect the capacitor across the analog input pin and the Vss pin at equal length.

● **Reason**

Signals which is input in an analog input pin (such as an A/D converter/comparator input pin) are usually output signals from sensor. The sensor which detects a change of event is installed far from the printed circuit board with a microcomputer, the wiring to an analog input pin is longer necessarily. This long wiring functions as an antenna which feeds noise into the microcomputer, which causes noise to an analog input pin.

If a capacitor between an analog input pin and the Vss pin is grounded at a position far away from the Vss pin, noise on the GND line may enter a microcomputer through the capacitor.

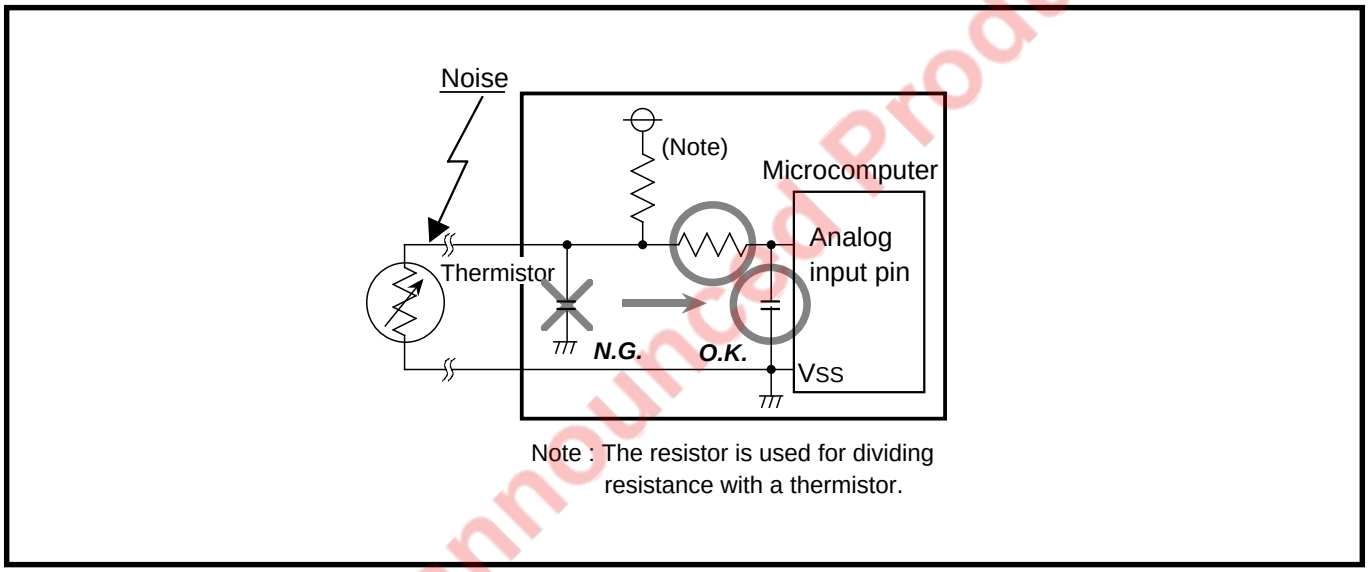


Fig. 3.4.5 Analog signal line and a resistor and a capacitor

3.4.4 Oscillator concerns

Take care to prevent an oscillator that generates clocks for a microcomputer operation from being affected by other signals.

(1) Keeping oscillator away from large current signal lines

Install a microcomputer (and especially an oscillator) as far as possible from signal lines where a current larger than the tolerance of current value flows.

● **Reason**

In the system using a microcomputer, there are signal lines for controlling motors, LEDs, and thermal heads or others. When a large current flows through those signal lines, strong noise occurs because of mutual inductance.

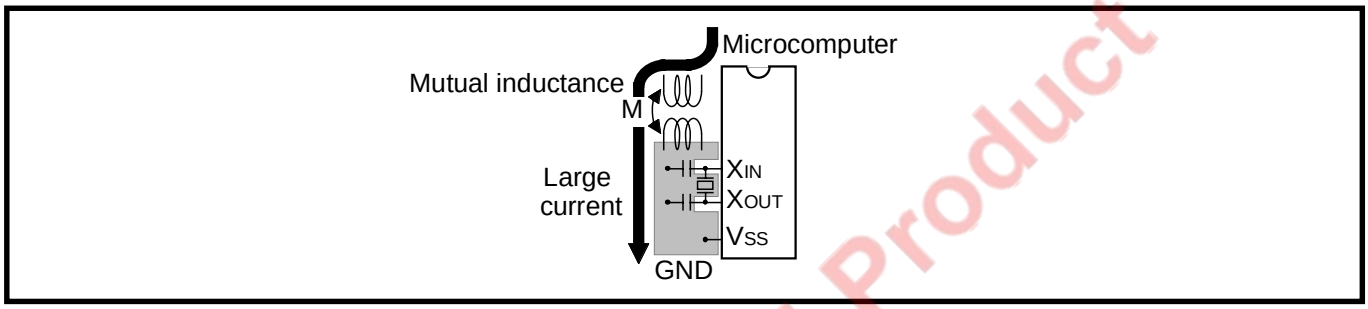


Fig. 3.4.6 Wiring for a large current signal line

(2) Installing oscillator away from signal lines where potential levels change frequently

Install an oscillator and a connecting pattern of an oscillator away from signal lines where potential levels change frequently. Also, do not cross such signal lines over the clock lines or the signal lines which are sensitive to noise.

● **Reason**

Signal lines where potential levels change frequently (such as the CNTR pin signal line) may affect other lines at signal rising edge or falling edge. If such lines cross over a clock line, clock waveforms may be deformed, which causes a microcomputer failure or a program runaway.

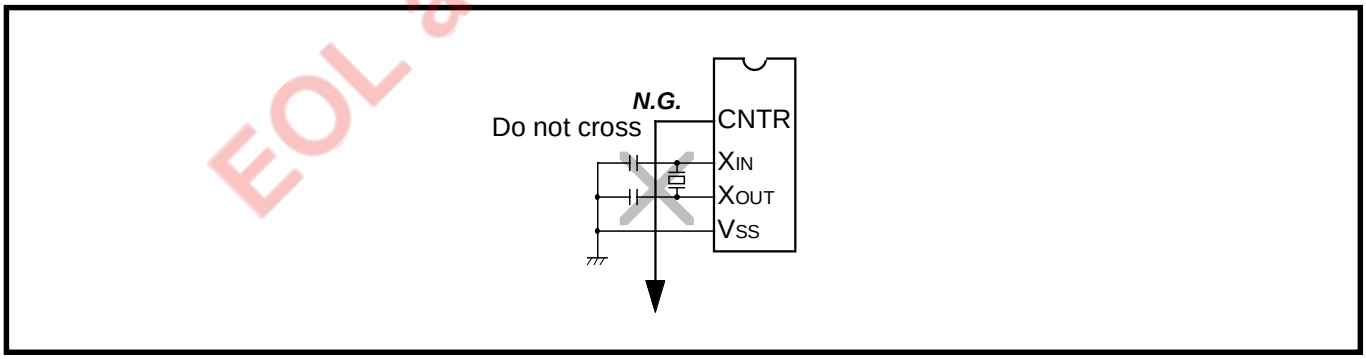


Fig. 3.4.7 Wiring of signal lines where potential levels change frequently

(3) Oscillator protection using Vss pattern

As for a two-sided printed circuit board, print a Vss pattern on the underside (soldering side) of the position (on the component side) where an oscillator is mounted.

Connect the Vss pattern to the microcomputer Vss pin with the shortest possible wiring. Besides, separate this Vss pattern from other Vss patterns.

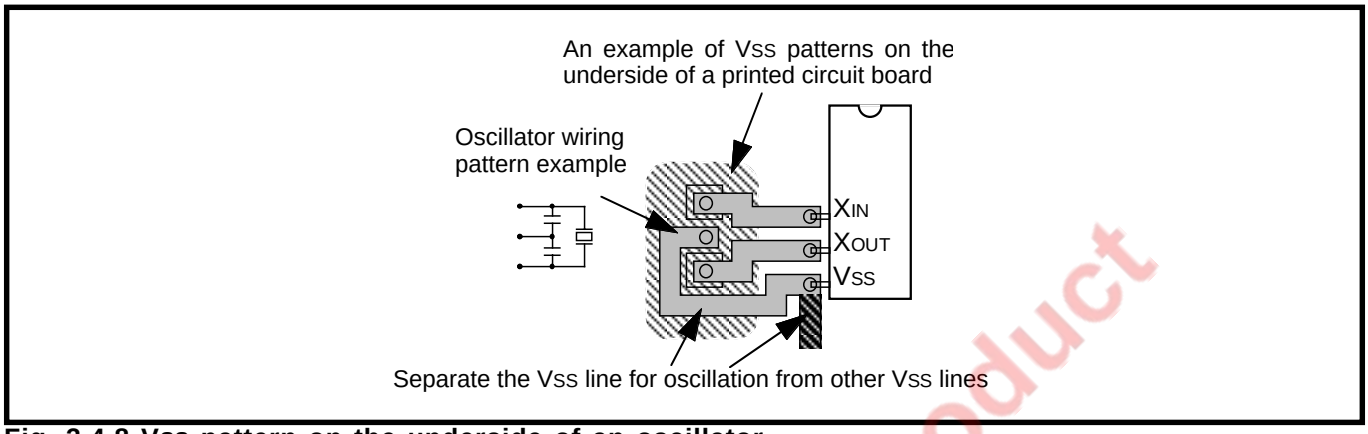


Fig. 3.4.8 Vss pattern on the underside of an oscillator

3.4.5 Setup for I/O ports

Setup I/O ports using hardware and software as follows:

<Hardware>

- Connect a resistor of 100 Ω or more to an I/O port in series.

<Software>

- As for an input port, read data several times by a program for checking whether input levels are equal or not.
- As for an output port, since the output data may reverse because of noise, rewrite data to its port latch at fixed periods.
- Rewrite data to direction registers and pull-up control registers at fixed periods.

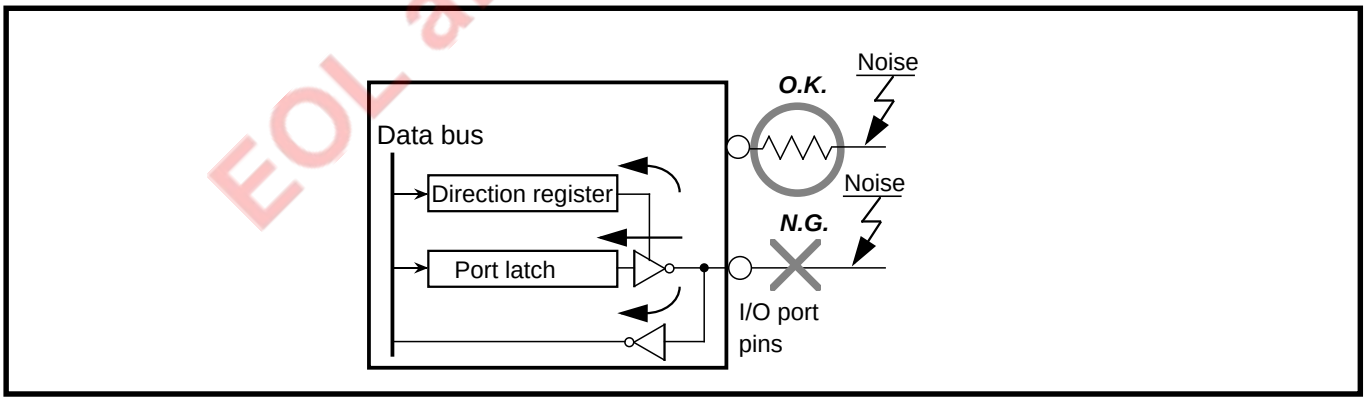


Fig. 3.4.9 Setup for I/O ports

3.4.6 Providing of watchdog timer function by software

If a microcomputer runs away because of noise or others, it can be detected by a software watchdog timer and the microcomputer can be reset to normal operation. This is equal to or more effective than program runaway detection by a hardware watchdog timer. The following shows an example of a watchdog timer provided by software.

In the following example, to reset a microcomputer to normal operation, the main routine detects errors of the interrupt processing routine and the interrupt processing routine detects errors of the main routine. This example assumes that interrupt processing is repeated multiple times in a single main routine processing.

<The main routine>

- Assigns a single byte of RAM to a software watchdog timer (SWDT) and writes the initial value N in the SWDT once at each execution of the main routine. The initial value N should satisfy the following condition:
  N+1 ≥ (Counts of interrupt processing executed in each main routine)
  As the main routine execution cycle may change because of an interrupt processing or others, the initial value N should have a margin.
- Watches the operation of the interrupt processing routine by comparing the SWDT contents with counts of interrupt processing after the initial value N has been set.
- Detects that the interrupt processing routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
  If the SWDT contents do not change after interrupt processing.

<The interrupt processing routine>

- Decrements the SWDT contents by 1 at each interrupt processing.
- Determines that the main routine operates normally when the SWDT contents are reset to the initial value N at almost fixed cycles (at the fixed interrupt processing count).
- Detects that the main routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
  If the SWDT contents are not initialized to the initial value N but continued to decrement and if they reach 0 or less.

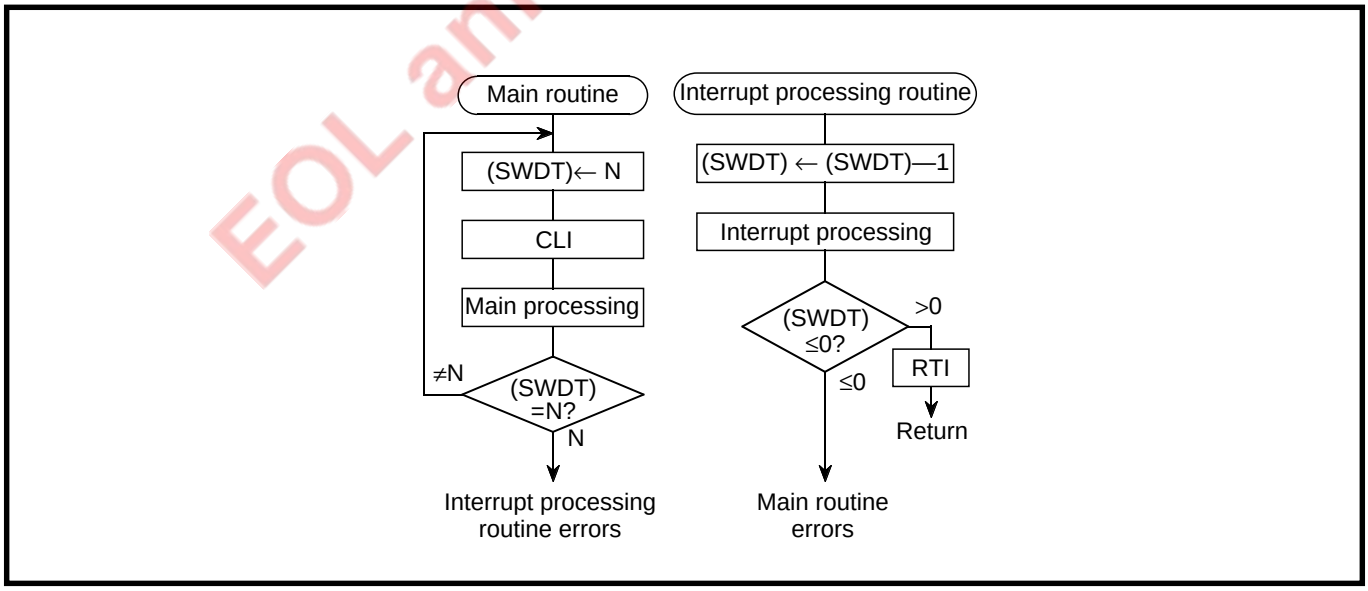


Fig. 3.4.10 Watchdog timer by software

3.5 Control registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (i = 0, 1, 2, 3, 4, 5, 6)
(Pi: addresses 0000_{16}, 0002_{16}, 0004_{16}, 0006_{16}, 0008_{16}, $000A_{16}$, $000C_{16}$)

b	Name	Functions	At reset	R	W
0	Port Pi_0	●In output mode Write •••••••• Port latch Read •••••••• Port latch ●In input mode Write •••••••• Port latch Read •••••••• Value of pin	0	○	○
1	Port Pi_1		0	○	○
2	Port Pi_2		0	○	○
3	Port Pi_3		0	○	○
4	Port Pi_4		0	○	○
5	Port Pi_5		0	○	○
6	Port Pi_6		0	○	○
7	Port Pi_7		0	○	○

Fig. 3.5.1 Structure of Port Pi

Port Pi direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi direction register (i = 0, 1, 2, 3, 4, 5, 6)
(PiD: addresses 0001_{16}, 0003_{16}, 0005_{16}, 0007_{16}, 0009_{16}, $000B_{16}$, $000D_{16}$)

b	Name	Functions	At reset	R	W
0	Port Pi direction register	0 : Port Pi_0 input mode 1 : Port Pi_0 output mode	0	×	○
1		0 : Port Pi_1 input mode 1 : Port Pi_1 output mode	0	×	○
2		0 : Port Pi_2 input mode 1 : Port Pi_2 output mode	0	×	○
3		0 : Port Pi_3 input mode 1 : Port Pi_3 output mode	0	×	○
4		0 : Port Pi_4 input mode 1 : Port Pi_4 output mode	0	×	○
5		0 : Port Pi_5 input mode 1 : Port Pi_5 output mode	0	×	○
6		0 : Port Pi_6 input mode 1 : Port Pi_6 output mode	0	×	○
7		0 : Port Pi_7 input mode 1 : Port Pi_7 output mode	0	×	○

Fig. 3.5.2 Structure of Port Pi direction register

Timer 12, X count source selection register

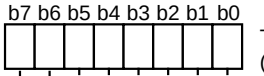
b7 b6 b5 b4 b3 b2 b1 b0

Timer 12, X count source selection register
(T12XCSS: address 000E16)

b	Name	Functions	At reset	R	W
0	Timer 12 count source selection bits	b3b2b1b0 0 0 0 0: f(XIN)/2 or f(XCIN)/2 0 0 0 1: f(XIN)/4 or f(XCIN)/4 0 0 1 0: f(XIN)/8 or f(XCIN)/8 0 0 1 1: f(XIN)/16 or f(XCIN)/16 0 1 0 0: f(XIN)/32 or f(XCIN)/32 0 1 0 1: f(XIN)/64 or f(XCIN)/64 0 1 1 0: f(XIN)/128 or f(XCIN)/128 0 1 1 1: f(XIN)/256 or f(XCIN)/256 1 0 0 0: f(XIN)/512 or f(XCIN)/512 1 0 0 1: f(XIN)/1024 or f(XCIN)/1024 1010 to 1111: Not available	1	○	○
1			1	○	○
2			0	○	○
3			0	○	○
4	Timer X count source selection bits	b7b6b5b4 0 0 0 0: f(XIN)/2 or f(XCIN)/2 0 0 0 1: f(XIN)/4 or f(XCIN)/4 0 0 1 0: f(XIN)/8 or f(XCIN)/8 0 0 1 1: f(XIN)/16 or f(XCIN)/16 0 1 0 0: f(XIN)/32 or f(XCIN)/32 0 1 0 1: f(XIN)/64 or f(XCIN)/64 0 1 1 0: f(XIN)/128 or f(XCIN)/128 0 1 1 1: f(XIN)/256 or f(XCIN)/256 1 0 0 0: f(XIN)/512 or f(XCIN)/512 1 0 0 1: f(XIN)/1024 or f(XCIN)/1024 1 0 1 0: f(XCIN) 1011 to 1111: Not available	1	○	○
5			1	○	○
6			0	○	○
7			0	○	○

Fig. 3.5.3 Structure of Timer 12, X count source selection register

Timer Y, Z count source selection register



Timer Y, Z count source selection register
(TYZCSS: address 000F16)

b	Name	Functions	At reset	R	W
0	Timer Y count source selection bits	b3b2b1b0 0 0 0 0: f(XIN)/2 or f(XCIN)/2 0 0 0 1: f(XIN)/4 or f(XCIN)/4 0 0 1 0: f(XIN)/8 or f(XCIN)/8 0 0 1 1: f(XIN)/16 or f(XCIN)/16 0 1 0 0: f(XIN)/32 or f(XCIN)/32 0 1 0 1: f(XIN)/64 or f(XCIN)/64 0 1 1 0: f(XIN)/128 or f(XCIN)/128 0 1 1 1: f(XIN)/256 or f(XCIN)/256 1 0 0 0: f(XIN)/512 or f(XCIN)/512 1 0 0 1: f(XIN)/1024 or f(XCIN)/1024 1 0 1 0: f(XCIN) 1011 to 1111: Not available	1	○	○
1			1	○	○
2			0	○	○
3			0	○	○
4	Timer Z count source selection bits	b7b6b5b4 0 0 0 0: f(XIN)/2 or f(XCIN)/2 0 0 0 1: f(XIN)/4 or f(XCIN)/4 0 0 1 0: f(XIN)/8 or f(XCIN)/8 0 0 1 1: f(XIN)/16 or f(XCIN)/16 0 1 0 0: f(XIN)/32 or f(XCIN)/32 0 1 0 1: f(XIN)/64 or f(XCIN)/64 0 1 1 0: f(XIN)/128 or f(XCIN)/128 0 1 1 1: f(XIN)/256 or f(XCIN)/256 1 0 0 0: f(XIN)/512 or f(XCIN)/512 1 0 0 1: f(XIN)/1024 or f(XCIN)/1024 1 0 1 0: f(XCIN) 1011 to 1111: Not available	1	○	○
5			1	○	○
6			0	○	○
7			0	○	○

Fig. 3.5.4 Structure of Timer Y, Z count source selection register

MISRG

b7 b6 b5 b4 b3 b2 b1 b0

MISRG
(MISRG: address 0010_{16})

b	Name	Functions	At reset	R	W
0	Oscillatin stabilizing time set after STP instrution released bit	0: Automatically set (**Note 1**) 1: Autimatically set disabled	0	○	○
1	Middle-speed mode automatic switch set bit	0: Not set automatically 1: Automatic switching enabled (**Notes 2, 3**)	0	○	○
2	Middle-speed mode automatic switch wait time set bit	0: 4.5 to 5.5 machine cycles 1: 6.5 to 7.5 machine cycles	0	○	○
3	Middle-speed mode automatic switch start bit (Depending on program)	0: Invalid 1: Automatic switch start (**Note 3**)	0	○	○
4	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Notes 1: "01_{16}" is set to Timer 1, "FF_{16}" is set to Prescaler 12.
2: During operation in low-speed mode, it is possible automatically to switch to middle-speed mode owing to the rising of SCL/SDA.
3: When automatic switch to middle-speed mode from low-speed mode occurs, the values of CPU mode register ($003B_{16}$) change.

Fig. 3.5.5 Structure of MISRG

I²C data shift register

b7 b6 b5 b4 b3 b2 b1 b0

I²C data shift register
(S0: address 0011_{16})

b	Functions	At reset	R	W
0	• 8-bit shift register to store receive data and write transmit data.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Note: When data is written to I²C data shift register after the MST bit is set to "0" (slave mode), keep the interval for 8 machine cycles or more.
Also, when the read-modify-write instructions (SEB, CLB) are used during data transfer, the values may be undefined.

Fig. 3.5.6 Structure of I²C data shift register

I²C special mode status register

b7 b6 b5 b4 b3 b2 b1 b0

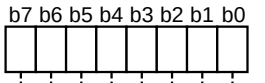
I²C special mode status register
(S3: address 0012₁₆)

b	Name	Functions	At reset	R	W
0	Slave address 0 comparison flag (AAS0)	0: Address disagreement 1: Address agreement **(Notes 1, 2)**	0	○	×
1	Slave address 1 comparison flag (AAS1)	0: Address disagreement 1: Address agreement **(Notes 1, 2)**	0	○	×
2	Slave address 2 comparison flag (AAS2)	0: Address disagreement 1: Address agreement **(Notes 1, 2)**	0	○	×
3	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are "0".		0	○	×
4	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are undefined.		0	○	×
5	SCL pin low hold 2 flag (PIN2)	0: SCL pin low hold 1: SCL pin low release **(Notes 1, 3)**	1	○	×
6	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are "0".		0	○	×
7	STOP condition flag (SPCF)	0: No detection 1: Detection **(Notes 1, 4)**	0	○	×

Notes 1: These bits and flags can be read out, but cannot be written.
2: These bits can be detected only when the data format selection bit (ALS) of I²C control register is set to "0".
3: This bit is initialized to "1" at reset, when the ACK interrupt control bit is "0", or when writing "1" to the SCL pin low hold 2 flag set bit.
4: This bit is initialized to "0" at reset, when the I²C-BUS interface enable bit (ES0) is "0", or when writing "1" to the STOP condition flag clear bit.

Fig. 3.5.7 Structure of I²C special mode status register

I²C status register



I²C status register
(S1: address 0013_{16})

b	Name	Functions	At reset	R	W
0	Last receive bit (LRB)	0: Last bit = "0" 1: Last bit = "1" (**Note 1**)	Undefined	○	×
1	General call detection flag (AD0)	0: No general call detected 1: General call detected (**Notes 1, 2**)	0	○	×
2	Slave address comparison flag (AAS)	0: Address disagreement 1: Address agreement (**Notes 1, 2**)	0	○	×
3	Arbitration lost detection flag (AL)	0: Not detected 1: Detected (**Note 1**)	0	○	×
4	SCL pin low hold bit (PIN)	0: SCL pin low hold (**Note 3**) 1: SCL pin low release	1	○	○
5	Bus busy flag (BB)	0: Bus free 1: Bus busy	0	○	○
6	Communication mode specification bits (TRX, MST)	b7 b6 0 0: Slave receive mode 0 1: Slave transmit mode 1 0: Master receive mode 1 1: Master transmit mode	0	○	○
7			0	○	○

Notes 1: These flags and bits are exclusive to input. When writing to these bits, write "0" to these bits.
2: These bits can be detected only when the data format selection bit (ALS) of I²C control register is set to "0".
3: This bit can be set to "1" by program, but cannot be cleared to "0".
4: All bits are changed by hardware. Do not use the read-modify-write instructions (SEB, CLB).

Fig. 3.5.8 Structure of I²C status register

I²C control register

b7 b6 b5 b4 b3 b2 b1 b0

I²C control register
(S1D: address 0014₁₆)

b	Name	Functions	At reset	R	W
0	Bit counter (Number of transmit/receive bits) (BC0, BC1, BC2)	b2b1b0 0 0 0: 8 0 0 1: 7 0 1 0: 6 0 1 1: 5 1 0 0: 4 1 0 1: 3 1 1 0: 2 1 1 1: 1	0	○	○
1			0	○	○
2			0	○	○
3	I²C-BUS interface enable bit (ES0)	0: Disabled 1: Enabled	0	○	○
4	Data format selection bit (ALS)	0: Addressing format 1: Free data format	0	○	○
5	Addressing format selection bit (10BIT SAD)	0: 7-bit addressing format 1: 10-bit addressing format	0	○	○
6	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
7	I²C-BUS interface pin input level selection bit (TISS)	0: SMBUS input 1: CMOS input	0	○	○

Note: Do not use the read-modify-write instruction because some bits change by hardware when the start condition is detected and the byte-transfer is completed.

Fig. 3.5.9 Structure of I²C control register

I²C clock control register

b7 b6 b5 b4 b3 b2 b1 b0

I²C clock control register
(S2: address 0015₁₆)

b	Name	Functions			At reset	R	W
0	SCL frequency control bits (CCR0, CCR1, CCR2, CCR3, CCR4)	Setting value b4b3b2b1b0	Standard clock mode	High-speed clock mode	0	○	○
		00 to 02	Disabled	Disabled			
1		03	Disabled	333	0	○	○
		04	(**Note 2**)	250			
2		05	100	400 (**Note 3**)	0	○	○
		06	83.3	166			
3		⋮	500/CCR value (**Note 3**)	1000/CCR value (**Note 3**)	0	○	○
		1D	17.2	34.5			
4		1E	16.6	33.3	0	○	○
		1F	16.1	32.3			
		(ϕ = 4 MHz, Unit: kHz) (**Note 1**)					
5	SCL mode specification bit (FAST MODE)	0: Standard clock mode 1: High-speed clock mode			0	○	○
6	ACK bit (ACK BIT)	0: ACK is returned. 1: ACK is not returned.			0	○	○
7	ACK clock bit (ACK)	0: No ACK clock 1: ACK clock			0	○	○

Notes 1: Duty of SCL clock output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at ϕ = 4 MHz). H duration of the clock fluctuates from —4 to +2 machine cycles in the standard clock mode, and fluctuates from —2 to +2 machine cycles in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because L duration is extended instead of H duration reduction.
These are values when SCL clock synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

2: Each value of SCL frequency exceeds the limit at ϕ = 4 MHz or more. When using these setting value, use ϕ of 4 MHz or less.

3: The data formula of SCL frequency is described below:
ϕ/(8 × CCR value) Standard clock mode
ϕ/(4 × CCR value) High-speed clock mode (CCR value ≠ 5)
ϕ/(2 × CCR value) High-speed clock mode (CCR value = 5)
Do not set 0 to 2 as CCR value regardless of ϕ frequency.
Set 100 kHz (max.) in the standard clock mode and 400 kHz (max.) in the high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

Fig. 3.5.10 Structure of I²C clock control register

I²C START/STOP condition control register

	b7	b6	b5	b4	b3	b2	b1	b0
	0							

I²C START/STOP condition control register
(S2D: address 0016₁₆)

b	Name	Functions	At reset	R	W
0	START/STOP condition set bits (SSC0, SSC1, SSC2, SSC3, SSC4)	SCL release time = ϕ (μs) × (SSC+1) Setup time = ϕ (μs) × (SSC+1)/2 Hold time = ϕ (μs) × (SSC+1)/2	0	○	○
1			1	○	○
2			0	○	○
3			1	○	○
4			1	○	○
5	SCL/SDA interrupt pin polarity selection bit (SIP)	0: Falling edge active 1: Rising edge active	0	○	○
6	SCL/SDA interrupt pin selection bit (SIS)	0: SDA valid 1: SCL valid	0	○	○
7	Fix this bit to 0 .		0	○	○

Note: Fix SSC0 to 0 . Also, do not set SSC4 to SSC0 to odd values or 00000₂ .

Fig. 3.5.11 Structure of I²C START/STOP condition control register

I²C special mode control register

	b7	b6	b5	b4	b3	b2	b1	b0
				0				0

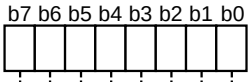
I²C special mode control register
(S3D: address 0017₁₆)

b	Name	Functions	At reset	R	W
0	Fix this bit to 0 .		0	○	○
1	ACK interrupt control bit (ACKICON)	0: At communication completion 1: At falling of ACK clock and communication completion	0	○	○
2	Slave address control bit (MSLAD)	0: One-byte slave address compare mode 1: Three-byte slave address compare mode	0	○	○
3	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are 0 .		0	○	×
4	Fix this bit to 0 .		0	○	○
5	SCL pin low hold 2 flag set bit (PIN2IN)	Writing 1 to this bit initializes the SCL pin low hold 2 flag to 1 . (**Notes 1, 2**)	0	○	○
6	SCL pin low hold set bit (PIN2HD)	Writing 1 to this bit clears the SCL pin low hold 2 flag to 0 and holds the SCL pin low. (**Notes 1, 2**)	0	○	○
7	STOP condition flag clear bit (SPFCL)	Writing 1 to this bit initializes the STOP condition flag to 0 . (**Note 1**)	0	○	○

Notes 1: When 0 is written to these bits, nothing is happened.
2: Do not write 1 to these bits at the same time.

Fig. 3.5.12 Structure of I²C special mode control register

Transmit/Receive buffer register 1, Transmit/Receive buffer register 3



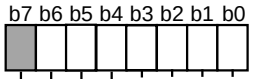
Transmit/Receive buffer register 1 (TB1/RB1: address 0018_{16})
Transmit/Receive buffer register 3 (TB3/RB3: address 0030_{16})

b	Functions	At reset	R	W
0	The transmission data is written to or the receive data is read out from this buffer register. • At write: A data is written to the transmit buffer register. • At read: The contents of the receive buffer register are read out.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Note: The contents of transmit buffer register cannot be read out.
The data cannot be written to the receive buffer register.

Fig. 3.5.13 Structure of Transmit/Receive buffer register 1, Transmit/Receive buffer register 3

Serial I/O1 status register, Serial I/O3 status register

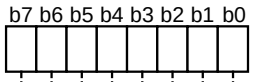


Serial I/O1 status register (SIO1STS: address 0019_{16})
Serial I/O3 status register (SIO3STS: address 0031_{16})

b	Name	Functions	At reset	R	W
0	Transmit buffer empty flag (TBE)	0: Buffer full 1: Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0: Buffer empty 1: Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0: Transmit shift in progress 1: Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0: No error 1: Overrun error	0	○	×
4	Parity error flag (PE)	0: No error 1: Parity error	0	○	×
5	Framing error flag (FE)	0: No error 1: Framing error	0	○	×
6	Summing error flag (SE)	0: (OE) U (PE) U (FE) = 0 1: (OE) U (PE) U (FE) = 1	0	○	×
7	Nothing is arranged for this bit. This bit is a write disabled bit. When this bit is read out, the contents are “1”.		1	○	×

Fig. 3.5.14 Structure of Serial I/O1 status register, Serial I/O3 status register

Serial I/O1 control register



Serial I/O1 control register
(SIO1CON: address 001A16)

b	Name	Functions	At reset	R	W
0	BRG count source selection bit (CSS)	0: f(XIN) 1: f(XIN)/4	0	○	○
1	Serial I/O1 synchronous clock selection bit (SCS)	When clock synchronous serial I/O is selected, 0: BRG output divided by 4 1: External clock input When UART is selected, 0: BRG output divided by 16 1: External clock input divided by16	0	○	○
2	$\overline{SRDY1}$ output enable bit (SRDY)	0: I/O port (P47) 1: $\overline{SRDY1}$ output pin	0	○	○
3	Transmit interrupt source selection bit (TIC)	0: Transmit buffer empty 1: Transmit shift operation completion	0	○	○
4	Transmit enable bit (TE)	0: Transmit disabled 1: Transmit enabled	0	○	○
5	Receive enable bit (RE)	0: Receive disabled 1: Receive enabled	0	○	○
6	Serial I/O1 mode selection bit (SIOM)	0: UART 1: Clock synchronous serial I/O	0	○	○
7	Serial I/O1 enable bit (SIOE)	0: Serial I/O1 disabled (P44 to P47: normal I/O pins) 1: Serial I/O1 enabled (P44 to P47: Serial I/O pins)	0	○	○

Fig. 3.5.15 Structure of Serial I/O1 control register

UART1 control register

b7 b6 b5 b4 b3 b2 b1 b0

UART1 control register
(UART1CON: address $001B_{16}$)

b	Name	Functions	At reset	R	W
0	Character length selection bit (CHAS)	0: 8 bits 1: 7 bits	0	○	○
1	Parity enable bit (PARE)	0: Parity checking disabled 1: Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0: Even parity 1: Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0: 1 stop bit 1: 2 stop bits	0	○	○
4	$P4_5/TxD_1$ P-channel output disable bit (POFF)	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "1".		1	○	×
6			1	○	×
7			1	○	×

Fig. 3.5.16 Structure of UART1 control register

Baud rate generator i (i = 1, 3)

b7 b6 b5 b4 b3 b2 b1 b0

Baud rate generator i
(BRGi (i=1, 3): address $001C_{16}$, $002F_{16}$)

b	Functions	At reset	R	W
0	Set a count value of baud rate generator.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Note: Write to this register while transmit/receive operation is stopped.

Fig. 3.5.17 Structure of Baud rate generator i

Serial I/O2 control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O2 control register
(SIO2CON: address 001D16)

b	Name	Functions	At reset	R	W
0	Internal synchronous clock selection bits	b2b1b0 0 0 0: f(XIN)/8 0 0 1: f(XIN)/16 0 1 0: f(XIN)/32 0 1 1: f(XIN)/64 1 1 0: f(XIN)/128 1 1 1: f(XIN)/256	0	○	○
1			0	○	○
2			0	○	○
3	Serial I/O2 port selection bit	0: I/O port (P51, P52) 1: SOUT2, SCLK2 signal output	0	○	○
4	$\overline{\text{SRDY2}}$ output enable bit	0: I/O port (P53) 1: $\overline{\text{SRDY2}}$ signal output	0	○	○
5	Transfer direction selection bit	0: LSB first 1: MSB first	0	○	○
6	Serial I/O2 synchronous clock selection bit	0: External clock 1: Internal clock	0	○	○
7	P51/SOUT2 P-channel output disable bit	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○

Fig. 3.5.18 Structure of Serial I/O2 control register

Watchdog timer control register

b7 b6 b5 b4 b3 b2 b1 b0

Watchdog timer control register
(WDTCON: address 001E16)

b	Name	Functions	At reset	R	W
0	Watchdog timer H (for read-out of high-order 6 bit)		1	○	×
1			1	○	×
2			1	○	×
3			1	○	×
4			1	○	×
5			1	○	×
6	STP instruction disable bit	0: STP instruction enabled 1: STP instruction disabled	0	○	○
7	Watchdog timer H count source selection bit	0: Watchdog timer L underflow 1: f(XIN)/16 or f(XCIN)/16	0	○	○

Fig. 3.5.19 Structure of Watchdog timer control register

Serial I/O2 register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O2 register
(SIO2: address 001F16)

b	Name	Functions	At reset	R	W
0		This register becomes shift register. At transmit: Set transmit data to this register. At receive: Received data is stored to this register.	Undefined	○	○
1			Undefined	○	○
2			Undefined	○	○
3			Undefined	○	○
4			Undefined	○	○
5			Undefined	○	○
6			Undefined	○	○
7			Undefined	○	○

Fig. 3.5.20 Structure of Serial I/O2 register

Prescaler 12, Prescaler X, Prescaler Y

b7 b6 b5 b4 b3 b2 b1 b0

Prescaler 12 (PRE12), Prescaler X (PREX), Prescaler Y (PREY)
(addresses 002016, 002416, 002616)

b	Functions	At reset	R	W
0	• Set a count value of each prescaler. • The value set in this register is written to both each prescaler and the corresponding prescaler latch at the same time. • When this register is read out, the count value of the corresponding prescaler is read out.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Fig. 3.5.21 Structure of Prescaler 12, Prescaler X, Prescaler Y

Timer 1

b7 b6 b5 b4 b3 b2 b1 b0

Timer 1
(T1: address 0021_{16})

b	Functions	At reset	R	W
0	• Set timer 1 count value. • The value set in this register is written to both the timer 1 and the timer 1 latch at the same time. • When the timer 1 is read out, the count value of the timer 1 is read out.	1	○	○
1		0	○	○
2		0	○	○
3		0	○	○
4		0	○	○
5		0	○	○
6		0	○	○
7		0	○	○

Fig. 3.5.22 Structure of Timer 1

Timer 2, Timer X, Timer Y

b7 b6 b5 b4 b3 b2 b1 b0

Timer 2 (T2), Timer X (TX), Timer Y (TY)
(addresses 0022_{16}, 0025_{16}, 0027_{16})

b	Functions	At reset	R	W
0	• Set each timer count value. • The value set in this register is written to both each timer and the corresponding timer latch at the same time. • When each timer is read out, the count value of the corresponding timer is read out.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Fig. 3.5.23 Structure of Timer 2, Timer X, Timer Y

Timer XY mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer XY mode register
(TM: address 0023_{16})

b	Name	Functions	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0: Timer mode 0 1: Pulse output mode 1 0: Event counter mode 1 1: Pulse width measurement mode	0	○	○
1			0	○	○
2	$CNTR_0$ active edge switch bit	Refer to Table 3.5.1	0	○	○
3	Timer X count stop bit	0: Count start 1: Count stop	0	○	○
4	Timer Y operating mode bits	b5 b4 0 0: Timer mode 0 1: Pulse output mode 1 0: Event counter mode 1 1: Pulse width measurement mode	0	○	○
5			0	○	○
6	$CNTR_1$ active edge switch bit	Refer to Table 3.5.1	0	○	○
7	Timer Y count stop bit	0: Count start 1: Count stop	0	○	○

Fig. 3.5.24 Structure of Timer XY mode register

Table 3.5.1 $CNTR_0$/$CNTR_1$ active edge switch bit function

Timer X/Timer Y operation modes	$CNTR_0$/$CNTR_1$ active edge switch bit (bits 2 and 6 of address 0023_{16}) contents	
Timer mode	"0"	$CNTR_0$/$CNTR_1$ interrupt request occurrence: Falling edge ; No influence to timer count
	"1"	$CNTR_0$/$CNTR_1$ interrupt request occurrence: Rising edge ; No influence to timer count
Pulse output mode	"0"	Pulse output start: Beginning at "H" level $CNTR_0$/$CNTR_1$ interrupt request occurrence: Falling edge
	"1"	Pulse output start: Beginning at "L" level $CNTR_0$/$CNTR_1$ interrupt request occurrence: Rising edge
Event counter mode	"0"	Timer X/Timer Y: Rising edge count $CNTR_0$/$CNTR_1$ interrupt request occurrence: Falling edge
	"1"	Timer X/Timer Y: Falling edge count $CNTR_0$/$CNTR_1$ interrupt request occurrence: Rising edge
Pulse width measurement mode	"0"	Timer X/Timer Y: "H" level width measurement $CNTR_0$/$CNTR_1$ interrupt request occurrence: Falling edge
	"1"	Timer X/Timer Y: "L" level width measurement $CNTR_0$/$CNTR_1$ interrupt request occurrence: Rising edge

Timer Z low-order, Timer Z high-order

b7 b6 b5 b4 b3 b2 b1 b0

Timer Z low-order (TZL), Timer Z high-order (TZH)
(addresses 0028_{16}, 0029_{16})

b	Functions	At reset	R	W
0	• Set each timer count value. [At write] • Depending on the write control bit (bit 3 of TZM), the value set to this register is written to each timer and the corresponding timer latch at the same time, or is written only to the latch. [At read] • The corresponding timer count value is read out by reading this register. • Read both registers in order of TZH and TZL following. • Write both registers in order of TZL and TZH following.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Fig. 3.5.25 Structure of Timer Z low-order, Timer Z high-order

Timer Z mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer Z mode register
(TZM: address $002A_{16}$)

b	Name	Functions	At reset	R	W
0	Timer Z operating mode bits	b2b1b0 0 0 0: Timer/Event counter mode 0 0 1: Pulse output mode 0 1 0: Pulse period measurement mode 0 1 1: Pulse width measurement mode 1 0 0: Programmable waveform generating mode 1 0 1: Programmable one-shot generating mode 1 1 0: Not available 1 1 1: Not available	0	○	○
1			0	○	○
2			0	○	○
3	Timer Z write control bit	0: Writing data to both latch and timer simultaneousuly 1: Writing data only to latch	0	○	○
4	Output level latch	0: "L" output 1: "H" output	0	○	○
5	$CNTR_2$ active edge switch bit	Refer to Table 3.5.2.	0	○	○
6	Timer Z count stop bit	0: Count start 1: Count stop	0	○	○
7	Timer/Event counter mode switch bit **(Note)**	0: Timer mode 1: Event counter mode	0	○	○

Note: When selecting the modes except the timer/event counter mode, set "0" to this bit.

Fig. 3.5.26 Structure of Timer Z mode register

Table 3.5.2 $CNTR_2$ active edge switch bit function

Timer Z operation modes	$CNTR_2$ active edge switch bit (bit 5 of address $002A_{16}$) contents	
Timer mode	"0"	$CNTR_2$ interrupt request occurrence: Falling edge ; No influence to timer count
	"1"	$CNTR_2$ interrupt request occurrence: Rising edge ; No influence to timer count
Event counter mode	"0"	Timer Z: Rising edge count $CNTR_2$ interrupt request occurrence: Falling edge
	"1"	Timer Z: Falling edge count $CNTR_2$ interrupt request occurrence: Rising edge
Pulse output mode	"0"	Pulse output start: Beginning at "H" level $CNTR_2$ interrupt request occurrence: Falling edge
	"1"	Pulse output start: Beginning at "L" level $CNTR_2$ interrupt request occurrence: Rising edge
Pulse period measurement mode	"0"	Timer Z: Period from falling edge to the next falling edge measurement $CNTR_2$ interrupt request occurrence: Falling edge
	"1"	Timer Z: Period from rising edge to the next rising edge measurement $CNTR_2$ interrupt request occurrence: Rising edge
Pulse width measurement mode	"0"	Timer Z: "H" level width measurement $CNTR_2$ interrupt request occurrence: Falling edge
	"1"	Timer Z: "L" level width measurement $CNTR_2$ interrupt request occurrence: Rising edge
Programmable one-shot generating mode	"0"	Timer Z: after start outputting "L", "H" one-shot pulse generated $CNTR_2$ interrupt request occurrence: Falling edge
	"1"	Timer Z: after start outputting "H", "L" one-shot pulse generated $CNTR_2$ interrupt request occurrence: Rising edge

PWM control register

b7 b6 b5 b4 b3 b2 b1 b0

PWM control register (PWMCON: address 002B16)

b	Name	Functions	At reset	R	W
0	PWM function enable bit	0 : PWM disabled 1 : PWM enabled	0	○	○
1	Count source selection bit	0 : f(XIN) 1 : f(XIN)/2	0	○	○
2	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.27 Structure of PWM control register

PWM prescaler

b7 b6 b5 b4 b3 b2 b1 b0

PWM prescaler
(PREPWM: address 002C16)

b	Functions	At reset	R	W
0	•Set the PWM period. •The value set in this register is written to both PWM prescaler pre-latch and PWM prescaler latch at the same time. • When data is written to this register during PWM output, the pulse corresponding to changed value is output at the next period. • When this register is read out, the count value of the PWM prescaler latch is read out.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 3.5.28 Structure of PWM prescaler

PWM register

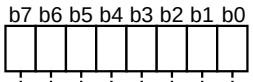
b7 b6 b5 b4 b3 b2 b1 b0

PWM register
(PWM: address 002D16)

b	Functions	At reset	R	W
0	• Set the PWM "H" level output interval. • The value set in this register is written to both PWM register pre-latch and PWM register latch at the same time. • When data is written to this register during PWM output, the pulse corresponding to changed value is output at the next period. • When this register is read out, the contents of the PWM register latch is read out.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 3.5.29 Structure of PWM register

Serial I/O3 control register

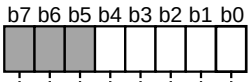


Serial I/O3 control register
(SIO3CON: address 0032_{16})

b	Name	Functions	At reset	R	W
0	BRG count source selection bit (CSS)	0: f(XIN) 1: f(XIN)/4	0	○	○
1	Serial I/O3 synchronous clock selection bit (SCS)	When clock synchronous serial I/O is selected, 0: BRG output divided by 4 1: External clock input When UART is selected, 0: BRG output divided by 16 1: External clock input divided by16	0	○	○
2	$\overline{SRDY3}$ output enable bit (SRDY)	0: I/O port (P37) 1: $\overline{SRDY3}$ output pin	0	○	○
3	Transmit interrupt source selection bit (TIC)	0: Transmit buffer empty 1: Transmit shift operation completion	0	○	○
4	Transmit enable bit (TE)	0: Transmit disabled 1: Transmit enabled	0	○	○
5	Receive enable bit (RE)	0: Receive disabled 1: Receive enabled	0	○	○
6	Serial I/O3 mode selection bit (SIOM)	0: UART 1: Clock synchronous serial I/O	0	○	○
7	Serial I/O3 enable bit (SIOE)	0: Serial I/O3 disabled (P34 to P37: normal I/O pins) 1: Serial I/O3 disabled (P34 to P37: Serial I/O pins)	0	○	○

Fig. 3.5.30 Structure of Serial I/O3 control register

UART3 control register

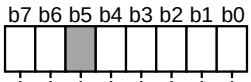


UART3 control register
(UART3CON: address 0033_{16})

b	Name	Functions	At reset	R	W
0	Character length selection bit (CHAS)	0: 8 bits 1: 7 bits	0	○	○
1	Parity enable bit (PARE)	0: Parity checking disabled 1: Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0: Even parity 1: Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0: 1 stop bit 1: 2 stop bits	0	○	○
4	$P3_5$/TxD_3 P-channel output disable bit (POFF)	0: CMOS output (in output mode) 1: N-channel open-drain output (in output mode)	0	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “1”.		1	○	×
6			1	○	×
7			1	○	×

Fig. 3.5.31 Structure of UART3 control register

AD/DA control register



AD/DA control register
(ADCON: address 0034_{16})

b	Name	Functions	At reset	R	W
0	Analog input pin selection bits 1	b2 b1 b0 0 0 0: $P6_0$/AN_0 or $P0_0$/AN_8 0 0 1: $P6_1$/AN_1 or $P0_1$/AN_9 0 1 0: $P6_2$/AN_2 or $P0_2$/AN_{10} 0 1 1: $P6_3$/AN_3 or $P0_3$/AN_{11} 1 0 0: $P6_4$/AN_4 or $P0_4$/AN_{12} 1 0 1: $P6_5$/AN_5 or $P0_5$/AN_{13} 1 1 0: $P6_6$/AN_6 or $P0_6$/AN_{14} 1 1 1: $P6_7$/AN_7 or $P0_7$/AN_{15}	0	○	○
1			0	○	○
2			0	○	○
3	AD conversion completion bit	0: Conversion in progress 1: Conversion completed	1	○	○
4	Analog input pin selection bit 2	0: AN_0 to AN_7 side 1: AN_8 to AN_{15} side	0	○	○
5	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
6	DA_1 output enable bit	0: DA_1 output disabled 1: DA_1 output enabled	0	○	○
7	DA_2 output enable bit	0: DA_2 output disabled 1: DA_2 output enabled	0	○	○

Fig. 3.5.32 Structure of AD/DA control register

AD conversion register 1

b7 b6 b5 b4 b3 b2 b1 b0

AD conversion register 1
(AD1: address 0035₁₆)

b	Functions	At reset	R	W
0	This is A/D conversion result stored bits. This is read exclusive register. 8-bit read: b7 … b0 = b9 b8 b7 b6 b5 b4 b3 b2 10-bit read: b7 … b0 = b7 b6 b5 b4 b3 b2 b1 b0	Undefined	○	×
1		Undefined	○	×
2		Undefined	○	×
3		Undefined	○	×
4		Undefined	○	×
5		Undefined	○	×
6		Undefined	○	×
7		Undefined	○	×

Fig. 3.5.33 Structure of AD conversion register 1

DAi conversion register

b7 b6 b5 b4 b3 b2 b1 b0

DAi conversion register (i = 1, 2)
(DAi: addresses 0036₁₆, 0037₁₆)

b	Functions	At reset	R	W
0	This is D/A output value stored bits. This is write exclusive register.	0	×	○
1		0	×	○
2		0	×	○
3		0	×	○
4		0	×	○
5		0	×	○
6		0	×	○
7		0	×	○

Fig. 3.5.34 Structure of DAi conversion register (i = 1, 2)

AD conversion register 2

b7 b6 b5 b4 b3 b2 b1 b0

AD conversion register 2
(AD2: address 0038₁₆)

b	Functions	At reset	R	W
0	This is A/D conversion result stored bits. This is read exclusive register.	Undefined	○	×
1	10-bit read: b7 … b0 = (b7 b6 b5 b4 b3 b2) b9 b8	Undefined	○	×
2	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.	0	○	×
3		0	○	×
4		0	○	×
5		0	○	×
6		0	○	×
7	Conversion mode selection bit 0: 10-bit A/D mode 1: 8-bit A/D mode	0	○	○

Fig. 3.5.35 Structure of AD conversion register 2

Interrupt source selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt source selection register
(INTSEL: address 0039_{16})

b	Name	Functions	At reset	R	W
0	INT_0/Timer Z interrupt source selection bit (*1)	0: INT_0 interrupt 1: Timer Z interrupt	0	○	○
1	Serial I/O2/Timer Z interrupt source selection bit (*1)	0: Serial I/O2 interrupt 1: Timer Z interrupt	0	○	○
2	Serial I/O1 transmit/ SCL, SDA interrupt source selection bit (*2)	0: Serial I/O1 transmit interrupt 1: SCL, SDA interrupt	0	○	○
3	$CNTR_0$/SCL, SDA interrupt source selection bit (*2)	0: $CNTR_0$ interrupt 1: SCL, SDA interrupt	0	○	○
4	INT_4/$CNTR_2$ interrupt source selection bit	0: INT_4 interrupt 1: $CNTR_2$ interrupt	0	○	○
5	INT_2/I^2C interrupt source selection bit	0: INT_2 interrupt 1: I^2C interrupt	0	○	○
6	$CNTR_1$/Serial I/O3 receive interrupt source selection bit	0: $CNTR_1$ interrupt 1: Serial I/O3 receive interrupt	0	○	○
7	AD converter/Serial I/O3 transmit interrupt source selection bit	0: A/D converter interrupt 1: Serial I/O3 transmit interrupt	0	○	○

*1: Do not write 1 to these bits simultaneously.
*2: Do not write 1 to these bits simultaneously.

Fig. 3.5.36 Structure of Interrupt source selection register

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt edge selection register
(INTEDGE: address 003A16)

b	Name	Functions	At reset	R	W
0	INT_0 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
1	INT_1 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
2	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are "0".		0	○	×
3	INT_2 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
4	INT_3 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
5	INT_4 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
6	INT_0, INT_4 interrupt switch bit	0: INT_{00}, INT_{40} interrupt 1: INT_{01}, INT_{41} interrupt	0	○	○
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are "0".		0	○	×

Fig. 3.5.37 Structure of Interrupt edge selection register

CPU mode register

b7 b6 b5 b4 b3 b2 b1 b0
(b3 = 1)

CPU mode register
(CPUM: address 003B16)

b	Name	Functions	At reset	R	W
0	Processor mode bits	b1 b0 00 : Single-chip mode 01 : 10 : } Not available 11 :	0	○	○
1			0	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to "1".		1	○	○
4	Port Xc switch bit	0: I/O port function (stop oscillating) 1: X_{CIN}-X_{COUT} oscillation function	0	○	○
5	Main clock (X_{IN}-X_{OUT}) stop bit	0: Oscillating 1: Stopped	0	○	○
6	Main clock division ratio selection bits	b7 b6 0 0: ϕ=f(X_{IN})/2 (high-speed mode) 0 1: ϕ=f(X_{IN})/8 (middle-speed mode) 1 0: ϕ=f(X_{CIN})/2 (low-speed mode) 1 1: not available	1	○	○
7			0	○	○

Fig. 3.5.38 Structure of CPU mode register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1
(IREQ1 : address 003C16)

b	Name	Functions	At reset	R	W
0	INT0/Timer Z interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
1	INT1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
2	Serial I/O1 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
3	Serial I/O1 transmit/SCL, SDA interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
4	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
5	Timer Y interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*

*: 0 can be set by software, but 1 cannot be set.

Fig. 3.5.39 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2
(IREQ2 : address 003D16)

b	Name	Functions	At reset	R	W
0	CNTR0/SCL, SDA interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
1	CNTR1/Serial I/O3 receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
2	Serial I/O2/Timer Z interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
3	INT2/I2C interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
4	INT3 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
5	INT4/CNTR2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
6	AD converter/Serial I/O3 transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
7	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are 0 .		0	○	×

*: 0 can be set by software, but 1 cannot be set.

Fig. 3.5.40 Structure of Interrupt request register 2

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1
(ICON1 : address 003E16)

b	Name	Functions	At reset	R	W
0	INT0/Timer Z interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit/SCL, SDA interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 3.5.41 Structure of Interrupt control register 1

Interrupt control register 2

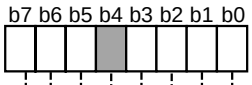
b7 b6 b5 b4 b3 b2 b1 b0
0

Interrupt control register 2
(ICON2 : address 003F16)

b	Name	Functions	At reset	R	W
0	CNTR0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	CNTR1/ Serial I/O3 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O2/ Timer Z interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT3 interrupt enable bit	0 : Interrupt disabled	0	○	○
5	INT4/CNTR2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	AD converter/Serial I/O3 transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 3.5.42 Structure of Interrupt control register 2

Flash memory control register 0



Flash memory control register 0
(FMCR0 : address 0FE0$_{16}$)

b	Name	Functions	At reset	R	W
0	RY/$\overline{BY}$ status flag	0 : Busy (being automatic written or automatic erased) 1 : Ready	1	○	×
1	CPU rewrite mode select bit **(Note 1)**	0 : CPU rewrite mode invalid (software commandes invalid) 1 : CPU rewrite mode valid (Software commands acceptable)	0	○	○
2	8 KB user block E/W enable bit **(Notes 1, 2)**	0: E/W disabled 1: E/W enabled	0	○	○
3	Flash memory reset bit **(Note 3)**	0: Normal operation 1: Reset	0	○	○
4	Not used (Do not write “1” to this bit.)		0	○	○
5	User ROM area select bit	0: User ROM area is accessed 1: Boot ROM area is accessed	0	○	○
6	Program status flag	0: Pass 1: Error	0	○	×
7	Erase status flag	0: Pass 1: Error	0	○	×

Notes 1: For this bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession. For this bit to be set to “0”, write “0” only to this bit.
2: This bit can be written only when the CPU rewrite mode select bit is “1”.
3: Effective only when the CPU rewrite mode select bit = “1”. Fix this bit to “0” when the CPU rewrite mode select bit is “0”.
4: When setting this bit to “1” (when the control circuit of flash memory is reset), the flash memory cannot be accessed for 10 µs.
5: Write to this bit from program on RAM.

Fig. 3.5.43 Structure of Flash memory control register 0

Flash memory control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Flash memory control register 1
(FMCR1 : address 0FE1$_{16}$)

b	Name	Functions	At reset	R	W
0	Erase suspend enable bit **(Note 1)**	0 : Suspend invalid 1 : Suspend valid	0	○	○
1	Erase suspend request bit **(Note 2)**	0 : Erase restart (no request issued) 1 : Suspend request (request issued)	0	○	○
2	Nothing is arranged for these bits. If writing to these bits, write “0”. The contents are undefined at reading.		0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6	Erase suspend flag	0 : Erase active 1 : Erase inactive (Erase suspend mode)	1	○	×
7	Nothing is arranged for these bits. If writing to these bits, write “0”. The contents are undefined at reading.		0	○	×

Notes 1: For this bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession.
2: Only when the erase suspend bit is “1”, this bit is valid.

Fig. 3.5.44 Structure of Flash memory control register 1

Flash memory control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Flash memory control register 2
(FMCR2 : address 0FE2$_{16}$)

b	Name	Functions	At reset	R	W
0	Nothing is arranged for these bits. If writing to these bits, write “0”. The contents are undefined at reading.		1	○	×
1			0	○	×
2			1	○	×
3			0	○	×
4	All user block E/W enable bit **(Notes 1, 2)**	0 : E/W disabled 1 : E/W enabled	0	○	○
5	Nothing is arranged for these bits. If writing to these bits, write “0”. The contents are undefined at reading.		0	○	×
6			1	○	×
7			0	○	×

Notes 1: For this bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession.
2: Effective only when the CPU rewrite mode select bit = “1”.

Fig. 3.5.45 Structure of Flash memory control register 2

Port Pi pull-up control register (i = 0 to 2, 4 to 6)

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi pull-up control register (i = 0 to 2, 4 to 6)
(PULLi: addresses 0FF0$_{16}$, 0FF1$_{16}$, 0FF2$_{16}$, 0FF4$_{16}$, 0FF5$_{16}$, 0FF6$_{16}$)

b	Name	Functions	At reset	R	W
0	Port Pi$_0$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
1	Port Pi$_1$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
2	Port Pi$_2$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
3	Port Pi$_3$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
4	Port Pi$_4$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
5	Port Pi$_5$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
6	Port Pi$_6$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
7	Port Pi$_7$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○

Fig. 3.5.46 Structure of Port Pi pull-up control register (i = 0 to 2, 4 to 6)

Port P3 pull-up control register

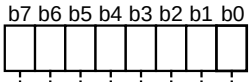
b7 b6 b5 b4 b3 b2 b1 b0

Port P3 pull-up control register
(PULL3: address 0FF3$_{16}$)

b	Name	Functions	At reset	R	W
0	Port P3$_0$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
1	Port P3$_1$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
2	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
3			0	○	×
4	Port P3$_4$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
5	Port P3$_5$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
6	Port P3$_6$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○
7	Port P3$_7$ pull-up control bit	0: No pull-up 1: Pull-up	0	○	○

Fig. 3.5.47 Structure of Port P3 pull-up control register

I²C slave address register i (i = 0 to 2)



I²C slave address register i (i = 0 to 2)
(S0D0, S0D1, S0D2: addresses 0FF7₁₆, 0FF8₁₆, 0FF9₁₆)

b	Name	Functions	At reset	R	W
0	Read/$\overline{\text{Write}}$ bit (RWB)	0: Write bit 1: Read bit	0	○	○
1	Slave address (SAD0, SAD1, SAD2, SAD3, SAD4, SAD5, SAD6)	The contents of these bits are compared with the address data transmitted from master.	0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Note: When the read-modify-write instructions (SEB, CLB) are used at detection of stop condition, the values may be undefined.

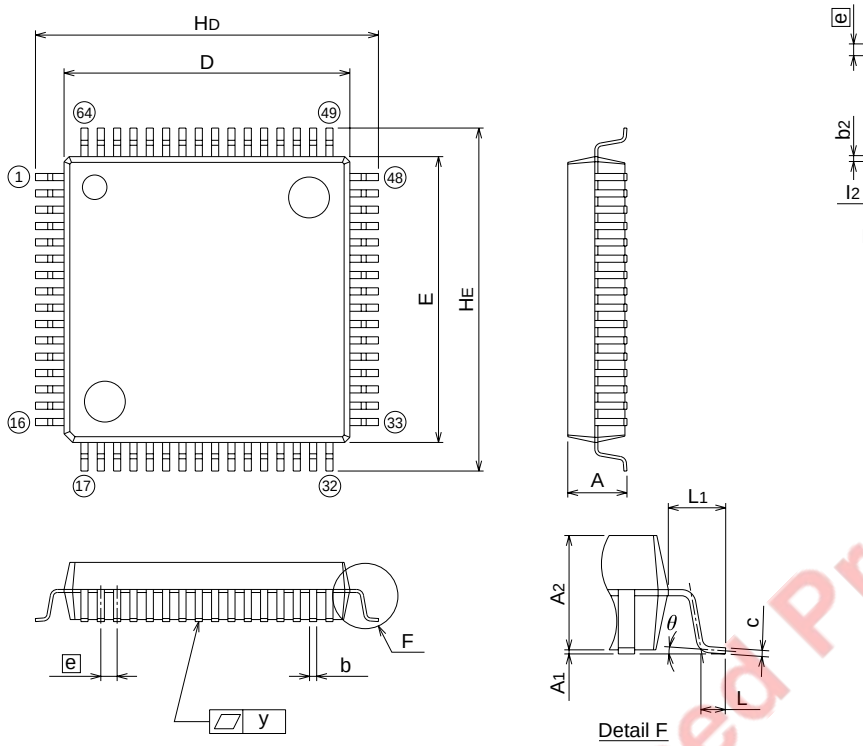
Fig. 3.5.48 Structure of I²C slave address register i (i = 0 to 2)

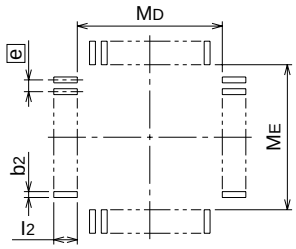
3.6 Package outline

64P6N-A

Plastic 64pin 14×14mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP64-P-1414-0.80	–	1.11	Alloy 42





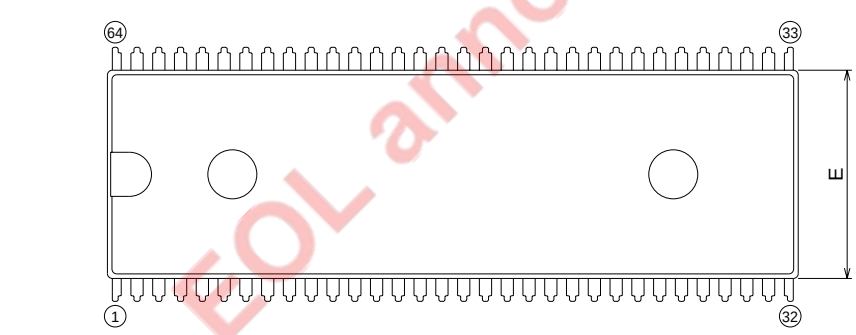
Recommended Mount Pad

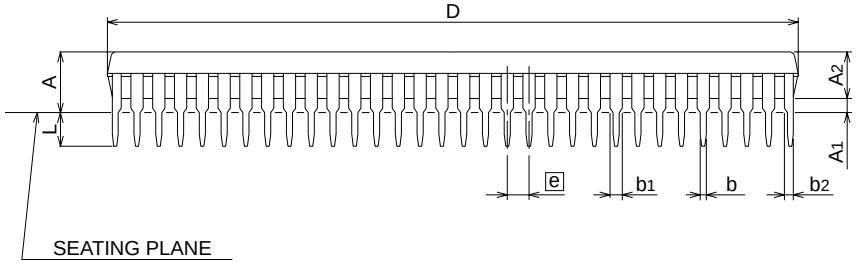
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	3.05
A1	0	0.1	0.2
A2	–	2.8	–
b	0.3	0.35	0.45
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	13.8	14.0	14.2
e	–	0.8	–
HD	16.5	16.8	17.1
HE	16.5	16.8	17.1
L	0.4	0.6	0.8
L1	–	1.4	–
y	–	–	0.1
θ	0°	–	10°
b2	–	0.5	–
l2	1.3	–	–
MD	–	14.6	–
ME	–	14.6	–

64P4B

Plastic 64pin 750mil SDIP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SDIP64-P-750-1.78	–	7.9	Alloy 42/Cu Alloy



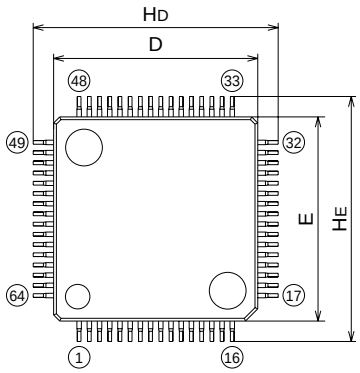


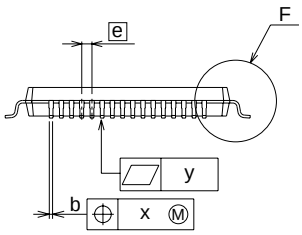
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	5.08
A1	0.38	–	–
A2	–	3.8	–
b	0.4	0.5	0.59
b1	0.9	1.0	1.3
b2	0.65	0.75	1.05
c	0.2	0.25	0.32
D	56.2	56.4	56.6
E	16.85	17.0	17.15
e	–	1.778	–
e1	–	19.05	–
L	2.8	–	–
θ	0°	–	15°

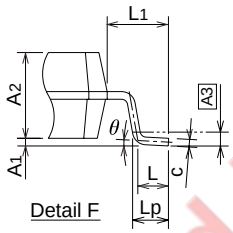
64P6Q-A

Plastic 64pin 10×10mm body LQFP

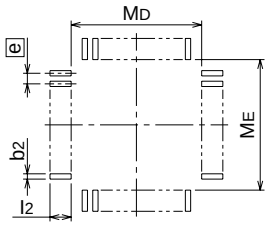
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP64-P-1010-0.5	—		Cu Alloy







Detail F



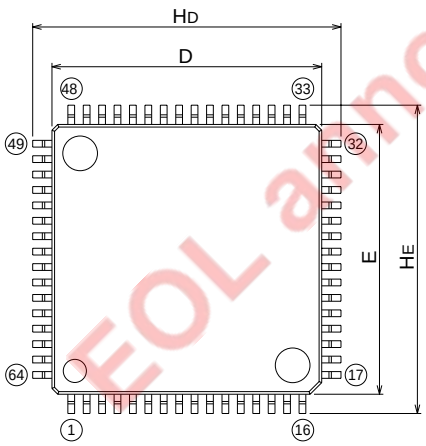
Recommended Mount Pad

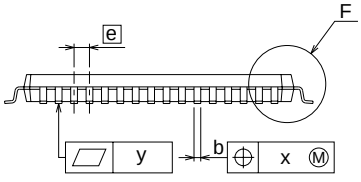
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	1.7
A1	0	0.1	0.2
A2	—	1.4	—
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	9.9	10.0	10.1
E	9.9	10.0	10.1
e	—	0.5	—
HD	11.8	12.0	12.2
HE	11.8	12.0	12.2
L	0.3	0.5	0.7
L1	—	1.0	—
Lp	0.45	0.6	0.75
A3	—	0.25	—
x	—	—	0.08
y	—	—	0.1
θ	0i	—	10i
b2	—	0.225	—
I2	1.0	—	—
MD	—	10.4	—
ME	—	10.4	—

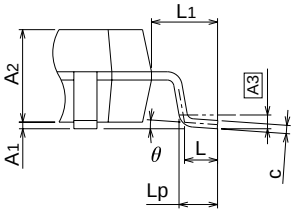
64P6U-A

Plastic 64pin 14×14mm body LQFP

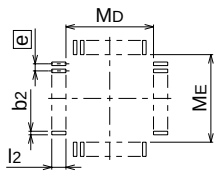
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP64-P-1414-0.8	—		Cu Alloy







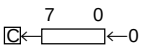
Detail F



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	1.7
A1	0	0.1	0.2
A2	—	1.4	—
b	0.32	0.37	0.45
c	0.105	0.125	0.175
D	13.9	14.0	14.1
E	13.9	14.0	14.1
e	—	0.8	—
HD	15.8	16.0	16.2
HE	15.8	16.0	16.2
L	0.3	0.5	0.7
L1	—	1.0	—
Lp	0.45	0.6	0.75
A3	—	0.25	—
x	—	—	0.2
y	—	—	0.1
θ	0i	—	8i
b2	—	0.5	—
I2	0.95	—	—
MD	—	14.4	—
ME	—	14.4	—

3.7 Machine instructions

Symbol	Function	Details	Addressing mode																
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
ADC (Note 1) (Note 5)	When T = 0 A ← A + M + C When T = 1 M(X) ← M(X) + M + C	When T = 0, this instruction adds the contents M, C, and A; and stores the results in A and C. When T = 1, this instruction adds the contents of M(X), M and C; and stores the results in M(X) and C. When T=1, the contents of A remain unchanged, but the contents of status flags are changed. M(X) represents the contents of memory where is indicated by X.				69	2	2							65	3	2			
AND (Note 1)	When T = 0 A ← A ∧ M When T = 1 M(X) ← M(X) ∧ M	When T = 0, this instruction transfers the contents of A and M to the ALU which performs a bit-wise AND operation and stores the result back in A. When T = 1, this instruction transfers the contents M(X) and M to the ALU which performs a bit-wise AND operation and stores the results back in M(X). When T = 1, the contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				29	2	2							25	3	2			
ASL		This instruction shifts the content of A or M by one bit to the left, with bit 0 always being set to 0 and bit 7 of A or M always being contained in C.							0A	2	1				06	5	2			
BBC (Note 4)	Ai or Mi = 0?	This instruction tests the designated bit i of M or A and takes a branch if the bit is 0. The branch address is specified by a relative address. If the bit is 1, next instruction is executed.										13+20i	4	2				17+20i	5	3
BBS (Note 4)	Ai or Mi = 1?	This instruction tests the designated bit i of the M or A and takes a branch if the bit is 1. The branch address is specified by a relative address. If the bit is 0, next instruction is executed.										03+20i	4	2				07+20i	5	3
BCC (Note 4)	C = 0?	This instruction takes a branch to the appointed address if C is 0. The branch address is specified by a relative address. If C is 1, the next instruction is executed.																		
BCS (Note 4)	C = 1?	This instruction takes a branch to the appointed address if C is 1. The branch address is specified by a relative address. If C is 0, the next instruction is executed.																		
BEQ (Note 4)	Z = 1?	This instruction takes a branch to the appointed address when Z is 1. The branch address is specified by a relative address. If Z is 0, the next instruction is executed.																		
BIT	A ∧ M	This instruction takes a bit-wise logical AND of A and M contents; however, the contents of A and M are not modified. The contents of N, V, Z are changed, but the contents of A, M remain unchanged.													24	3	2			
BMI (Note 4)	N = 1?	This instruction takes a branch to the appointed address when N is 1. The branch address is specified by a relative address. If N is 0, the next instruction is executed.																		
BNE (Note 4)	Z = 0?	This instruction takes a branch to the appointed address if Z is 0. The branch address is specified by a relative address. If Z is 1, the next instruction is executed.																		

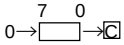
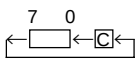
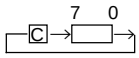
Addressing mode																											Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
75	4	2				6D	4	3	7D	5	3	79	5	3							61	6	2	71	6	2							N	V	•	•	•	•	Z	C
35	4	2				2D	4	3	3D	5	3	39	5	3							21	6	2	31	6	2							N	•	•	•	•	•	Z	•
16	6	2				0E	6	3	1E	7	3																						N	•	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																											90	2	2				•	•	•	•	•	•	•	•
																											B0	2	2				•	•	•	•	•	•	•	•
																											F0	2	2				•	•	•	•	•	•	•	•
						2C	4	3																										M7	M6	•	•	•	•	Z	•
																											30	2	2				•	•	•	•	•	•	•	•
																											D0	2	2				•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
BPL (Note 4)	N = 0?	This instruction takes a branch to the appointed address if N is 0. The branch address is specified by a relative address. If N is 1, the next instruction is executed.																		
BRA	PC ← PC ± offset	This instruction branches to the appointed address. The branch address is specified by a relative address.																		
BRK	B ← 1 (PC) ← (PC) + 2 M(S) ← PC_H S ← S – 1 M(S) ← PC_L S ← S – 1 M(S) ← PS S ← S – 1 I← 1 PC_L ← AD_L PC_H ← AD_H	When the BRK instruction is executed, the CPU pushes the current PC contents onto the stack. The BADRS designated in the interrupt vector table is stored into the PC.	00	7	1															
BVC (Note 4)	V = 0?	This instruction takes a branch to the appointed address if V is 0. The branch address is specified by a relative address. If V is 1, the next instruction is executed.																		
BVS (Note 4)	V = 1?	This instruction takes a branch to the appointed address when V is 1. The branch address is specified by a relative address. When V is 0, the next instruction is executed.																		
CLB	Ai or Mi ← 0	This instruction clears the designated bit i of A or M.										1B + 20i	2	1				1F + 20i	5	2
CLC	C ← 0	This instruction clears C.	18	2	1															
CLD	D ← 0	This instruction clears D.	D8	2	1															
CLI	I ← 0	This instruction clears I.	58	2	1															
CLT	T ← 0	This instruction clears T.	12	2	1															
CLV	V ← 0	This instruction clears V.	B8	2	1															
CMP (Note 3)	When T = 0 A – M When T = 1 M(X) – M	When T = 0, this instruction subtracts the contents of M from the contents of A. The result is not stored and the contents of A or M are not modified. When T = 1, the CMP subtracts the contents of M from the contents of M(X). The result is not stored and the contents of X, M, and A are not modified. M(X) represents the contents of memory where is indicated by X.				C9	2	2							C5	3	2			
COM	M ← $\overline{M}$	This instruction takes the one's complement of the contents of M and stores the result in M.													44	5	2			
CPX	X – M	This instruction subtracts the contents of M from the contents of X. The result is not stored and the contents of X and M are not modified.				E0	2	2							E4	3	2			
CPY	Y – M	This instruction subtracts the contents of M from the contents of Y. The result is not stored and the contents of Y and M are not modified.				C0	2	2							C4	3	2			
DEC	A ← A – 1 or M ← M – 1	This instruction subtracts 1 from the contents of A or M.							1A	2	1				C6	5	2			

Addressing mode																											Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
																											10	2	2				•	•	•	•	•	•	•	•
																											80	4	2				•	•	•	•	•	•	•	•
																																		•	•	•	1	•	1	•	•
																											50	2	2				•	•	•	•	•	•	•	•
																											70	2	2				•	•	•	•	•	•	•	•
																																		•	•	•	•	•	•	•	•
																																		•	•	•	•	•	•	•	0
																																		•	•	•	•	0	•	•	•
																																		•	•	•	•	•	0	•	•
																																		•	•	0	•	•	•	•	•
																																		•	0	•	•	•	•	•	•
D5	4	2				CD	4	3	DD	5	3	D9	5	3							C1	6	2	D1	6	2							N	•	•	•	•	•	Z	C
																																		N	•	•	•	•	•	Z	•
						EC	4	3																										N	•	•	•	•	•	Z	C
						CC	4	3																										N	•	•	•	•	•	Z	C
D6	6	2				CE	6	3	DE	7	3																							N	•	•	•	•	•	Z	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
DEX	X ← X – 1	This instruction subtracts one from the current contents of X.	CA	2	1															
DEY	Y ← Y – 1	This instruction subtracts one from the current contents of Y.	88	2	1															
DIV	A ← (M(zz + X + 1), M(zz + X)) / A M(S) ← one's complement of Remainder S ← S – 1	This instruction divides the 16-bit data in M(zz+(X)) (low-order byte) and M(zz+(X)+1) (high-order byte) by the contents of A. The quotient is stored in A and the one's complement of the remainder is pushed onto the stack.																		
EOR (Note 1)	When T = 0 A ← A ∀ M When T = 1 M(X) ← M(X) ∀ M	When T = 0, this instruction transfers the contents of the M and A to the ALU which performs a bit-wise Exclusive OR, and stores the result in A. When T = 1, the contents of M(X) and M are transferred to the ALU, which performs a bit-wise Exclusive OR and stores the results in M(X). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				49	2	2							45	3	2			
INC	A ← A + 1 or M ← M + 1	This instruction adds one to the contents of A or M.							3A	2	1				E6	5	2			
INX	X ← X + 1	This instruction adds one to the contents of X.	E8	2	1															
INY	Y ← Y + 1	This instruction adds one to the contents of Y.	C8	2	1															
JMP	If addressing mode is ABS PC_L ← AD_L PC_H ← AD_H If addressing mode is IND PC_L ← M (AD_H, AD_L) PC_H ← M (AD_H, AD_L + 1) If addressing mode is ZP, IND PC_L ← M(00, AD_L) PC_H ← M(00, AD_L + 1)	This instruction jumps to the address designated by the following three addressing modes: Absolute Indirect Absolute Zero Page Indirect Absolute																		
JSR	M(S) ← PC_H S ← S – 1 M(S) ← PC_L S ← S – 1 After executing the above, if addressing mode is ABS, PC_L ← AD_L PC_H ← AD_H if addressing mode is SP, PC_L ← AD_L PC_H ← FF If addressing mode is ZP, IND, PC_L ← M(00, AD_L) PC_H ← M(00, AD_L + 1)	This instruction stores the contents of the PC in the stack, then jumps to the address designated by the following addressing modes: Absolute Special Page Zero Page Indirect Absolute																		
LDA (Note 2)	When T = 0 A ← M When T = 1 M(X) ← M	When T = 0, this instruction transfers the contents of M to A. When T = 1, this instruction transfers the contents of M to (M(X)). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				A9	2	2							A5	3	2			
LDM	M ← nn	This instruction loads the immediate value in M.													3C	4	3			
LDX	X ← M	This instruction loads the contents of M in X.				A2	2	2							A6	3	2			
LDY	Y ← M	This instruction loads the contents of M in Y.				A0	2	2							A4	3	2			

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
E2	16	2																															•	•	•	•	•	•	•	•
55	4	2				4D	4	3	5D	5	3	59	5	3							41	6	2	51	6	2							N	•	•	•	•	•	Z	•
F6	6	2				EE	6	3	FE	7	3																						N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
						4C	3	3							6C	5	3	B2	4	2													•	•	•	•	•	•	•	•
						20	6	3										02	7	2										22	5	2	•	•	•	•	•	•	•	•
B5	4	2				AD	4	3	BD	5	3	B9	5	3							A1	6	2	B1	6	2							N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
			B6	4	2	AE	4	3				BE	5	3																			N	•	•	•	•	•	Z	•
B4	4	2				AC	4	3	BC	5	3																						N	•	•	•	•	•	Z	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
LSR		This instruction shifts either A or M one bit to the right such that bit 7 of the result always is set to 0, and the bit 0 is stored in C.							4A	2	1				46	5	2			
MUL	M(S) • A ← A ✻ M(zz + X) S ← S – 1	This instruction multiply Accumulator with the memory specified by the Zero Page X address mode and stores the high-order byte of the result on the Stack and the low-order byte in A.																		
NOP	PC ← PC + 1	This instruction adds one to the PC but does no other operation.	EA	2	1															
ORA (Note 1)	When T = 0 A ← A V M When T = 1 M(X) ← M(X) V M	When T = 0, this instruction transfers the contents of A and M to the ALU which performs a bit-wise “OR”, and stores the result in A. When T = 1, this instruction transfers the contents of M(X) and the M to the ALU which performs a bit-wise OR, and stores the result in M(X). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				09	2	2							05	3	2			
PHA	M(S) ← A S ← S – 1	This instruction pushes the contents of A to the memory location designated by S, and decrements the contents of S by one.	48	3	1															
PHP	M(S) ← PS S ← S – 1	This instruction pushes the contents of PS to the memory location designated by S and decrements the contents of S by one.	08	3	1															
PLA	S ← S + 1 A ← M(S)	This instruction increments S by one and stores the contents of the memory designated by S in A.	68	4	1															
PLP	S ← S + 1 PS ← M(S)	This instruction increments S by one and stores the contents of the memory location designated by S in PS.	28	4	1															
ROL		This instruction shifts either A or M one bit left through C. C is stored in bit 0 and bit 7 is stored in C.							2A	2	1				26	5	2			
ROR		This instruction shifts either A or M one bit right through C. C is stored in bit 7 and bit 0 is stored in C.							6A	2	1				66	5	2			
RRF		This instruction rotates 4 bits of the M content to the right.													82	8	2			
RTI	S ← S + 1 PS ← M(S) S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S)	This instruction increments S by one, and stores the contents of the memory location designated by S in PS. S is again incremented by one and stores the contents of the memory location designated by S in PCL. S is again incremented by one and stores the contents of memory location designated by S in PCH.	40	6	1															
RTS	S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S) (PC) ← (PC) + 1	This instruction increments S by one and stores the contents of the memory location designated by S in PCL. S is again incremented by one and the contents of the memory location is stored in PCH. PC is incremented by 1.	60	6	1															

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
56	6	2				4E	6	3	5E	7	3																						0	•	•	•	•	•	Z	C
62	15	2																															•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
15	4	2				0D	4	3	1D	5	3	19	5	3							01	6	2	11	6	2							N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	(Value saved in stack)							
36	6	2				2E	6	3	3E	7	3																						N	•	•	•	•	•	Z	C
76	6	2				6E	6	3	7E	7	3																						N	•	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	(Value saved in stack)							
																																	•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
SBC (Note 1) (Note 5)	When T = 0 A ← A – M – C When T = 1 M(X) ← M(X) – M – C	When T = 0, this instruction subtracts the value of M and the complement of C from A, and stores the results in A and C. When T = 1, the instruction subtracts the contents of M and the complement of C from the contents of M(X), and stores the results in M(X) and C. A remain unchanged, but status flag are changed. M(X) represents the contents of memory where is indicated by X.				E9	2	2							E5	3	2			
SEB	Ai or Mi ← 1	This instruction sets the designated bit i of A or M.										0B + 20i	2	1				0F + 20i	5	2
SEC	C ← 1	This instruction sets C.	38	2	1															
SED	D ← 1	This instruction set D.	F8	2	1															
SEI	I ← 1	This instruction set I.	78	2	1															
SET	T ← 1	This instruction set T.	32	2	1															
STA	M ← A	This instruction stores the contents of A in M. The contents of A does not change.													85	4	2			
STP		This instruction resets the oscillation control F/F and the oscillation stops. Reset or interrupt input is needed to wake up from this mode.	42	2	1															
STX	M ← X	This instruction stores the contents of X in M. The contents of X does not change.													86	4	2			
STY	M ← Y	This instruction stores the contents of Y in M. The contents of Y does not change.													84	4	2			
TAX	X ← A	This instruction stores the contents of A in X. The contents of A does not change.	AA	2	1															
TAY	Y ← A	This instruction stores the contents of A in Y. The contents of A does not change.	A8	2	1															
TST	M = 0?	This instruction tests whether the contents of M are “0” or not and modifies the N and Z.													64	3	2			
TSX	X ← S	This instruction transfers the contents of S in X.	BA	2	1															
TXA	A ← X	This instruction stores the contents of X in A.	8A	2	1															
TXS	S ← X	This instruction stores the contents of X in S.	9A	2	1															
TYA	A ← Y	This instruction stores the contents of Y in A.	98	2	1															
WIT		The WIT instruction stops the internal clock but not the oscillation of the oscillation circuit is not stopped. CPU starts its function after the Timer X over flows (comes to the terminal count). All registers or internal memory contents except Timer X will not change during this mode. (Of course needs VDD).	C2	2	1															

Notes 1 : The number of cycles “n” is increased by 3 when T is 1.
2 : The number of cycles “n” is increased by 2 when T is 1.
3 : The number of cycles “n” is increased by 1 when T is 1.
4 : The number of cycles “n” is increased by 2 when branching has occurred.
5 : N, V, and Z flags are invalid in decimal operation mode.

Addressing mode																											Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2							N	V	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	1
																																	•	•	•	•	1	•	•	•
																																	•	•	•	•	•	1	•	•
																																	•	•	1	•	•	•	•	•
95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	2							•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
			96	5	2	8E	5	3																									•	•	•	•	•	•	•	•
94	5	2				8C	5	3																									•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•

Symbol	Contents	Symbol	Contents
IMP	Implied addressing mode	+	Addition
IMM	Immediate addressing mode	–	Subtraction
A	Accumulator or Accumulator addressing mode	✻	Multiplication
BIT, A	Accumulator bit addressing mode	/	Division
BIT, A, R	Accumulator bit relative addressing mode	Λ	Logical OR
ZP	Zero page addressing mode	V	Logical AND
BIT, ZP	Zero page bit addressing mode	∀	Logical exclusive OR
BIT, ZP, R	Zero page bit relative addressing mode	—	Negation
ZP, X	Zero page X addressing mode	←	Shows direction of data flow
ZP, Y	Zero page Y addressing mode	X	Index register X
ABS	Absolute addressing mode	Y	Index register Y
ABS, X	Absolute X addressing mode	S	Stack pointer
ABS, Y	Absolute Y addressing mode	PC	Program counter
IND	Indirect absolute addressing mode	PS	Processor status register
		PC_H	8 high-order bits of program counter
ZP, IND	Zero page indirect absolute addressing mode	PC_L	8 low-order bits of program counter
		AD_H	8 high-order bits of address
IND, X	Indirect X addressing mode	AD_L	8 low-order bits of address
IND, Y	Indirect Y addressing mode	FF	FF in Hexadecimal notation
REL	Relative addressing mode	nn	Immediate value
SP	Special page addressing mode	zz	Zero page address
C	Carry flag	M	Memory specified by address designation of any addressing mode
Z	Zero flag		
I	Interrupt disable flag	M(X)	Memory of address indicated by contents of index register X
D	Decimal mode flag		
B	Break flag	M(S)	Memory of address indicated by contents of stack pointer
T	X-modified arithmetic mode flag		
V	Overflow flag	$M(AD_H, AD_L)$	Contents of memory at address indicated by AD_H and AD_L, in AD_H is 8 high-order bits and AD_L is 8 low-order bits.
N	Negative flag		
		$M(00, AD_L)$	Contents of address indicated by zero page AD_L
		Ai	Bit i (i = 0 to 7) of accumulator
		Mi	Bit i (i = 0 to 7) of memory
		OP	Opcode
		n	Number of cycles
		#	Number of bytes

3.8 List of instruction code

D7 – D4 \ D3 – D0	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND, X	JSR ZP, IND	BBS 0, A	—	ORA ZP	ASL ZP	BBS 0, ZP	PHP	ORA IMM	ASL A	SEB 0, A	—	ORA ABS	ASL ABS	SEB 0, ZP
0001	1	BPL	ORA IND, Y	CLT	BBC 0, A	—	ORA ZP, X	ASL ZP, X	BBC 0, ZP	CLC	ORA ABS, Y	DEC A	CLB 0, A	—	ORA ABS, X	ASL ABS, X	CLB 0, ZP
0010	2	JSR ABS	AND IND, X	JSR SP	BBS 1, A	BIT ZP	AND ZP	ROL ZP	BBS 1, ZP	PLP	AND IMM	ROL A	SEB 1, A	BIT ABS	AND ABS	ROL ABS	SEB 1, ZP
0011	3	BMI	AND IND, Y	SET	BBC 1, A	—	AND ZP, X	ROL ZP, X	BBC 1, ZP	SEC	AND ABS, Y	INC A	CLB 1, A	LDM ZP	AND ABS, X	ROL ABS, X	CLB 1, ZP
0100	4	RTI	EOR IND, X	STP	BBS 2, A	COM ZP	EOR ZP	LSR ZP	BBS 2, ZP	PHA	EOR IMM	LSR A	SEB 2, A	JMP ABS	EOR ABS	LSR ABS	SEB 2, ZP
0101	5	BVC	EOR IND, Y	—	BBC 2, A	—	EOR ZP, X	LSR ZP, X	BBC 2, ZP	CLI	EOR ABS, Y	—	CLB 2, A	—	EOR ABS, X	LSR ABS, X	CLB 2, ZP
0110	6	RTS	ADC IND, X	MUL ZP, X	BBS 3, A	TST ZP	ADC ZP	ROR ZP	BBS 3, ZP	PLA	ADC IMM	ROR A	SEB 3, A	JMP IND	ADC ABS	ROR ABS	SEB 3, ZP
0111	7	BVS	ADC IND, Y	—	BBC 3, A	—	ADC ZP, X	ROR ZP, X	BBC 3, ZP	SEI	ADC ABS, Y	—	CLB 3, A	—	ADC ABS, X	ROR ABS, X	CLB 3, ZP
1000	8	BRA	STA IND, X	RRF ZP	BBS 4, A	STY ZP	STA ZP	STX ZP	BBS 4, ZP	DEY	—	TXA	SEB 4, A	STY ABS	STA ABS	STX ABS	SEB 4, ZP
1001	9	BCC	STA IND, Y	—	BBC 4, A	STY ZP, X	STA ZP, X	STX ZP, Y	BBC 4, ZP	TYA	STA ABS, Y	TXS	CLB 4, A	—	STA ABS, X	—	CLB 4, ZP
1010	A	LDY IMM	LDA IND, X	LDX IMM	BBS 5, A	LDY ZP	LDA ZP	LDX ZP	BBS 5, ZP	TAY	LDA IMM	TAX	SEB 5, A	LDY ABS	LDA ABS	LDX ABS	SEB 5, ZP
1011	B	BCS	LDA IND, Y	JMP ZP, IND	BBC 5, A	LDY ZP, X	LDA ZP, X	LDX ZP, Y	BBC 5, ZP	CLV	LDA ABS, Y	TSX	CLB 5, A	LDY ABS, X	LDA ABS, X	LDX ABS, Y	CLB 5, ZP
1100	C	CPY IMM	CMP IND, X	WIT	BBS 6, A	CPY ZP	CMP ZP	DEC ZP	BBS 6, ZP	INY	CMP IMM	DEX	SEB 6, A	CPY ABS	CMP ABS	DEC ABS	SEB 6, ZP
1101	D	BNE	CMP IND, Y	—	BBC 6, A	—	CMP ZP, X	DEC ZP, X	BBC 6, ZP	CLD	CMP ABS, Y	—	CLB 6, A	—	CMP ABS, X	DEC ABS, X	CLB 6, ZP
1110	E	CPX IMM	SBC IND, X	DIV ZP, X	BBS 7, A	CPX ZP	SBC ZP	INC ZP	BBS 7, ZP	INX	SBC IMM	NOP	SEB 7, A	CPX ABS	SBC ABS	INC ABS	SEB 7, ZP
1111	F	BEQ	SBC IND, Y	—	BBC 7, A	—	SBC ZP, X	INC ZP, X	BBC 7, ZP	SED	SBC ABS, Y	—	CLB 7, A	—	SBC ABS, X	INC ABS, X	CLB 7, ZP

: 3-byte instruction

: 2-byte instruction

: 1-byte instruction

3.9 SFR memory map

0000₁₆	Port P0 (P0)
0001₁₆	Port P0 direction register (P0D)
0002₁₆	Port P1 (P1)
0003₁₆	Port P1 direction register (P1D)
0004₁₆	Port P2 (P2)
0005₁₆	Port P2 direction register (P2D)
0006₁₆	Port P3 (P3)
0007₁₆	Port P3 direction register (P3D)
0008₁₆	Port P4 (P4)
0009₁₆	Port P4 direction register (P4D)
000A₁₆	Port P5 (P5)
000B₁₆	Port P5 direction register (P5D)
000C₁₆	Port P6 (P6)
000D₁₆	Port P6 direction register (P6D)
000E₁₆	Timer 12, X count source selection register (T12XCSS)
000F₁₆	Timer Y, Z count source selection register (TYZCSS)
0010₁₆	MISRG
0011₁₆	I²C data shift register (S0)
0012₁₆	I²C special mode status register (S3)
0013₁₆	I²C status register (S1)
0014₁₆	I²C control register (S1D)
0015₁₆	I²C clock control register (S2)
0016₁₆	I²C START/STOP condition control register (S2D)
0017₁₆	I²C special mode control register (S3D)
0018₁₆	Transmit/Receive buffer register 1 (TB1/RB1)
0019₁₆	Serial I/O1 status register (SIO1STS)
001A₁₆	Serial I/O1 control register (SIO1CON)
001B₁₆	UART1 control register (UART1CON)
001C₁₆	Baud rate generator 1 (BRG1)
001D₁₆	Serial I/O2 control register (SIO2CON)
001E₁₆	Watchdog timer control register (WDTCON)
001F₁₆	Serial I/O2 register (SIO2)

0020₁₆	Prescaler 12 (PRE12)
0021₁₆	Timer 1 (T1)
0022₁₆	Timer 2 (T2)
0023₁₆	Timer XY mode register (TM)
0024₁₆	Prescaler X (PREX)
0025₁₆	Timer X (TX)
0026₁₆	Prescaler Y (PREY)
0027₁₆	Timer Y (TY)
0028₁₆	Timer Z low-order (TZL)
0029₁₆	Timer Z high-order (TZH)
002A₁₆	Timer Z mode register (TZM)
002B₁₆	PWM control register (PWMCON)
002C₁₆	PWM prescaler (PREPWM)
002D₁₆	PWM register (PWM)
002E₁₆	
002F₁₆	Baud rate generator 3 (BRG3)
0030₁₆	Transmit/Receive buffer register 3 (TB3/RB3)
0031₁₆	Serial I/O3 status register (SIO3STS)
0032₁₆	Serial I/O3 control register (SIO3CON)
0033₁₆	UART3 control register (UART3CON)
0034₁₆	AD/DA control register (ADCON)
0035₁₆	AD conversion register 1 (AD1)
0036₁₆	DA1 conversion register (DA1)
0037₁₆	DA2 conversion register (DA2)
0038₁₆	AD conversion register 2 (AD2)
0039₁₆	Interrupt source selection register (INTSEL)
003A₁₆	Interrupt edge selection register (INTEDGE)
003B₁₆	CPU mode register (CPUM)
003C₁₆	Interrupt request register 1 (IREQ1)
003D₁₆	Interrupt request register 2 (IREQ2)
003E₁₆	Interrupt control register 1 (ICON1)
003F₁₆	Interrupt control register 2 (ICON2)

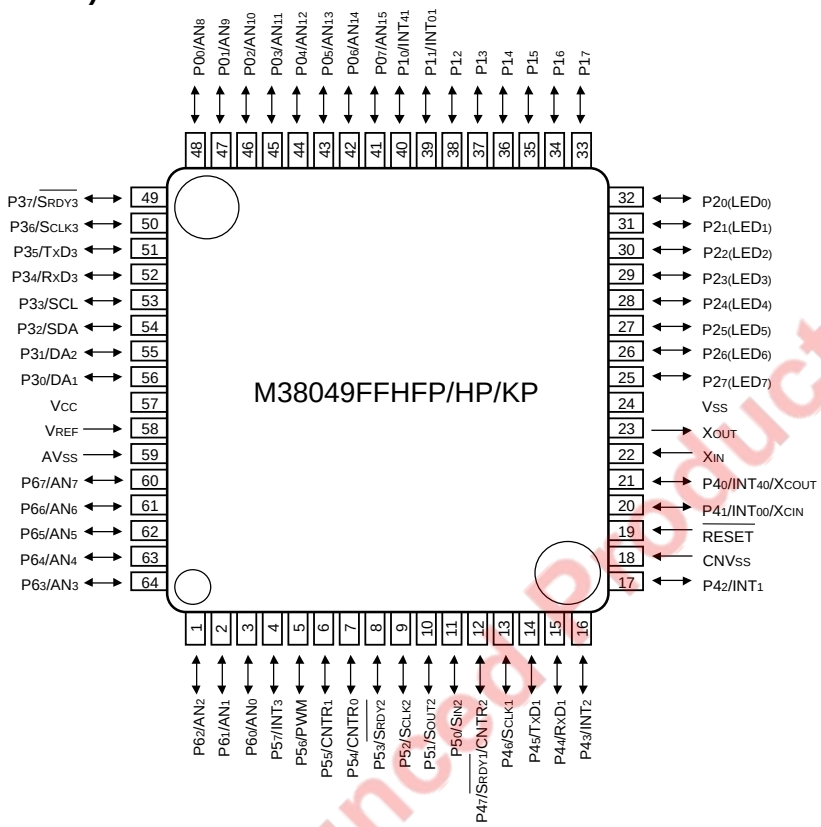
0FE0₁₆	Flash memory control register 0 (FMCR0)
0FE1₁₆	Flash memory control register 1 (FMCR1)
0FE2₁₆	Flash memory control register 2 (FMCR2)
0FE3₁₆	Reserved ✻
0FE4₁₆	Reserved ✻
0FE5₁₆	Reserved ✻
0FE6₁₆	Reserved ✻
0FE7₁₆	Reserved ✻
0FE8₁₆	Reserved ✻
0FE9₁₆	Reserved ✻
0FEA₁₆	Reserved ✻
0FEB₁₆	Reserved ✻
0FEC₁₆	Reserved ✻
0FED₁₆	Reserved ✻
0FEE₁₆	Reserved ✻
0FEF₁₆	Reserved ✻

0FF0₁₆	Port P0 pull-up control register (PULL0)
0FF1₁₆	Port P1 pull-up control register (PULL1)
0FF2₁₆	Port P2 pull-up control register (PULL2)
0FF3₁₆	Port P3 pull-up control register (PULL3)
0FF4₁₆	Port P4 pull-up control register (PULL4)
0FF5₁₆	Port P5 pull-up control register (PULL5)
0FF6₁₆	Port P6 pull-up control register (PULL6)
0FF7₁₆	I²C slave address register 0 (S0D0)
0FF8₁₆	I²C slave address register 1 (S0D1)
0FF9₁₆	I²C slave address register 2 (S0D2)

✻ Reserved area: Do not write any data to these addresses, because these areas are reserved.

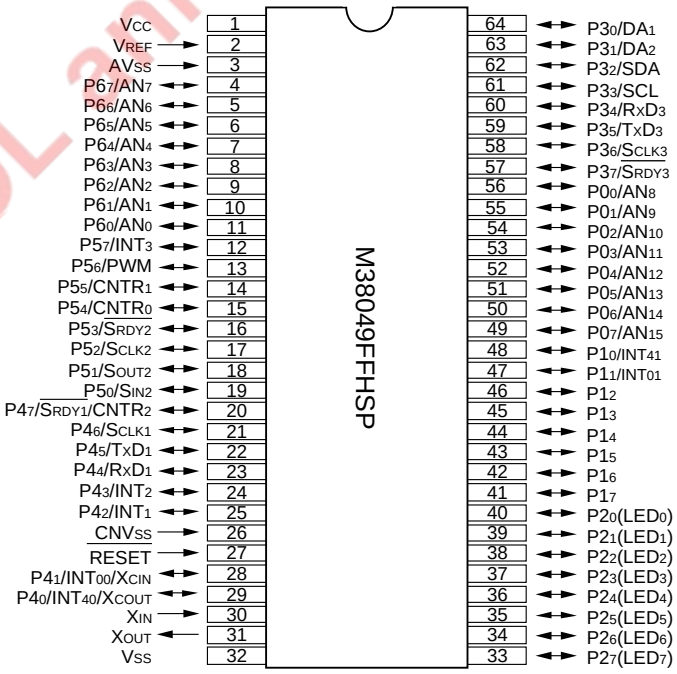
3.10 Pin configurations

(TOP VIEW)



Package type : 64P6N-A/64P6Q-A/64P6U-A

(TOP VIEW)



Package type : 64P4B

3804 Group (Spec. H) User's Manual

Publication Data : Rev.1.00 Jan 14, 2005

Published by : Sales Strategic Planning Div.
Renesas Technology Corp.

3804 Group (Spec. H)
User's Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan