REJ09B0080-0103Z

8

3850 Group (Spec. H)

User's Manual

RENESAS 8-BIT CISC SINGLE-CHIP MICROCOMPUTER

740 FAMILY / 38000 SERIES

Before using this material, please visit our website to confirm that this is the most current document available.

Rev. 1.03
Revision date: Sep. 18, 2003

RenesasTechnology
www.renesas.com

User's Manual

Keep safety first in your circuit designs!

- Renesas Technology Corporation 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 nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation 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 Corporation or a third party.
- Renesas Technology Corporation 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.
- 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 Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation 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 Corporation 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 Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
- 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 Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Renesas Technology Corporation 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 Corporation or an authorized Renesas Technology Corporation 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.
- The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
- 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.
- Please contact Renesas Technology Corporation for further details on these materials or t he products contained therein.

REVISION HISTORY	3850 Group (Spec. H) User's Manual

Rev.	Date	Description	
		Page	Summary
1.0	Aug. 30, 2001	–	First edition issued
1.1	Sep. 10, 2001	3-5	Limits and test conditions into Table 3.1.5 are partly added.
1.02	Aug. 29, 2003	1-6 1-7 1-7 1-37 1-38 2-103 2-104 2-105 2-106 2-107 3-4 3-20 3-22 3-24	Fig. 4 is partly revised. Table 2 is partly added. Note of Table 3 is added. Fig. 42 is partly revised. Fig. 43 is partly revised. Clause name of "2.11 Flash memory mode" is revised. Notes of Fig. 2.11.3 are partly revised. Table 2.11.2 is partly revised. Explanations of "[Beginning procedure]" of "2.11.6 CPU rewrite mode" are partly added. Explanations of this page are added. Parameter of Table 3.1.4 is partly revised. Fig. 3.2.16 is partly revised. Fig. 3.2.20 is partly revised. Fig. 3.2.24 is partly revised.
1.03	Sep. 18, 2003	1-51	Fig. 52 is partly revised.

Preface

This user's manual describes Renesas's CMOS 8-bit microcomputers 3850 Group (Spec. H).
After reading this manual, the user should have a through knowledge of the functions and features of the 3850 Group (Spec. H), and should be able to fully utilize the product. The manual starts with specifications and ends with application examples.
For details of software, refer to the "740 Family Software Manual".
The user who is using the 3850 Group (standard) needs to refer to not this manual but "3850/3851 Group User's Manual".

BEFORE USING THIS 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. Chapter 3 also includes necessary information for systems development. You must refer to that chapter.

1. Organization

● **CHAPTER 1 HARDWARE**
This chapter describes features of the microcomputer and operation of each peripheral function.

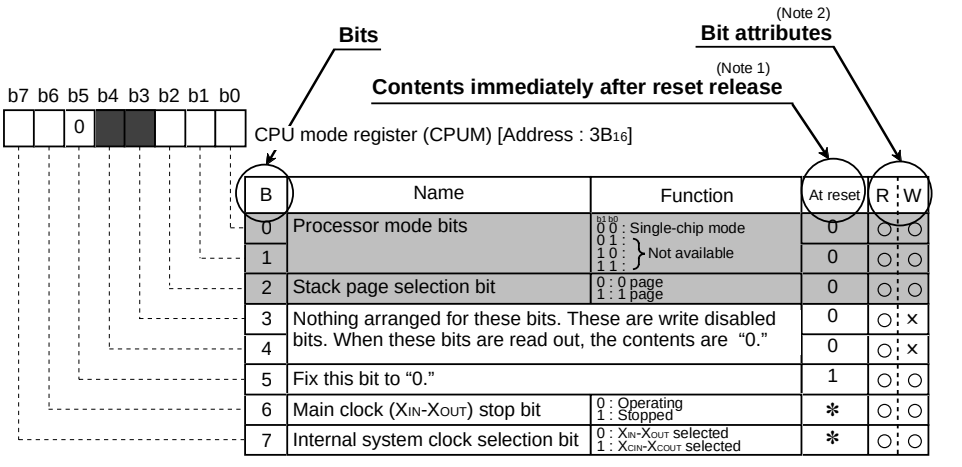
● **CHAPTER 2 APPLICATION**
This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of relevant registers.

● **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.

✻For the mask ROM confirmation form, the ROM programming confirmation form, and the mark specifications, refer to the "Renesas Technology Corp." Homepage (http://www.renesas.com/en/rom).

2. Structure of register

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



Note 1:. Contents immediately after reset release
0..... "0" at reset release
1..... "1" at reset release
?..... Undefined at reset release
✻.....Contents determined by option at reset release

Note 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

Table of contents

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

Table of contents

List of figures

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

List of figures

CHAPTER 3 APPENDIX

List of figures

List of tables

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

CHAPTER 1
HARDWARE

HARDWARE

DESCRIPTION/FEATURES/APPLICATION/PIN CONFIGURATION

DESCRIPTION

The 3850 group (spec. H) is the 8-bit microcomputer based on the 740 family core technology.
The 3850 group (spec. H) is designed for the household products and office automation equipment and includes serial I/O functions, 8-bit timer, and A-D converter.

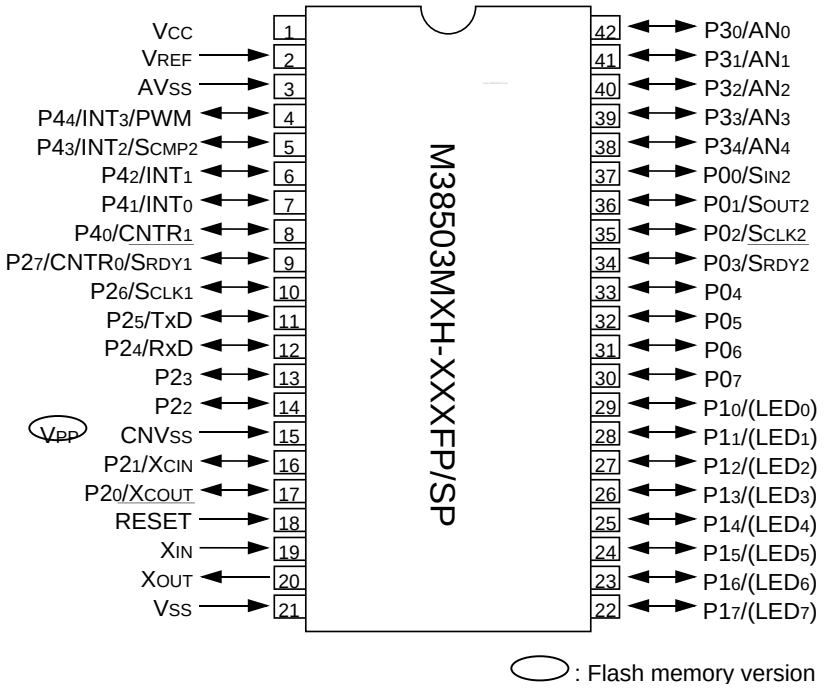
FEATURES

- Basic machine-language instructions 71
- Minimum instruction execution time ..... 0.5 µs
  (at 8 MHz oscillation frequency)
- Memory size
  ROM ..... 8K to 32K bytes
  RAM ..... 512 to 1024 bytes
- Programmable input/output ports 34
- Interrupts ..... 15 sources, 14 vectors
- Timers ..... 8-bit × 4
- Serial I/O1 ..... 8-bit × 1(UART or Clock-synchronized)
- Serial I/O2 ..... 8-bit × 1(Clock-synchronized)
- PWM ..... 8-bit × 1
- A-D converter ..... 10-bit × 5 channels
- Watchdog timer ..... 16-bit × 1
- Clock generating circuit ..... Built-in 2 circuits
  (connect to external ceramic resonator or quartz-crystal oscillator)
- Power source voltage
  In high-speed mode ..... 4.0 to 5.5 V
  (at 8 MHz oscillation frequency)
  In middle-speed mode ..... 2.7 to 5.5 V
  (at 8 MHz oscillation frequency)
  In low-speed mode ..... 2.7 to 5.5 V
  (at 32 kHz oscillation frequency)
- Power dissipation
  In high-speed mode ..... 34 mW
  (at 8 MHz oscillation frequency, at 5 V power source voltage)
  In low-speed mode
  Except M38507F8FP/SP ..... 60 µW
  M38507F8FP/SP ..... 450 µW
  (at 32 kHz oscillation frequency, at 3 V power source voltage)
- Operating temperature range ..... –20 to 85°C

APPLICATION

Office automation equipment, FA equipment, Household products, Consumer electronics, etc.

PIN CONFIGURATION (TOP VIEW)



Package type : FP ..... 42P2R-A/E (42-pin plastic-molded SSOP)
Package type : SP ..... 42P4B (42-pin plastic-molded SDIP)

Fig. 1 M38503MXH-XXXFP/SP pin configuration

FUNCTIONAL BLOCK

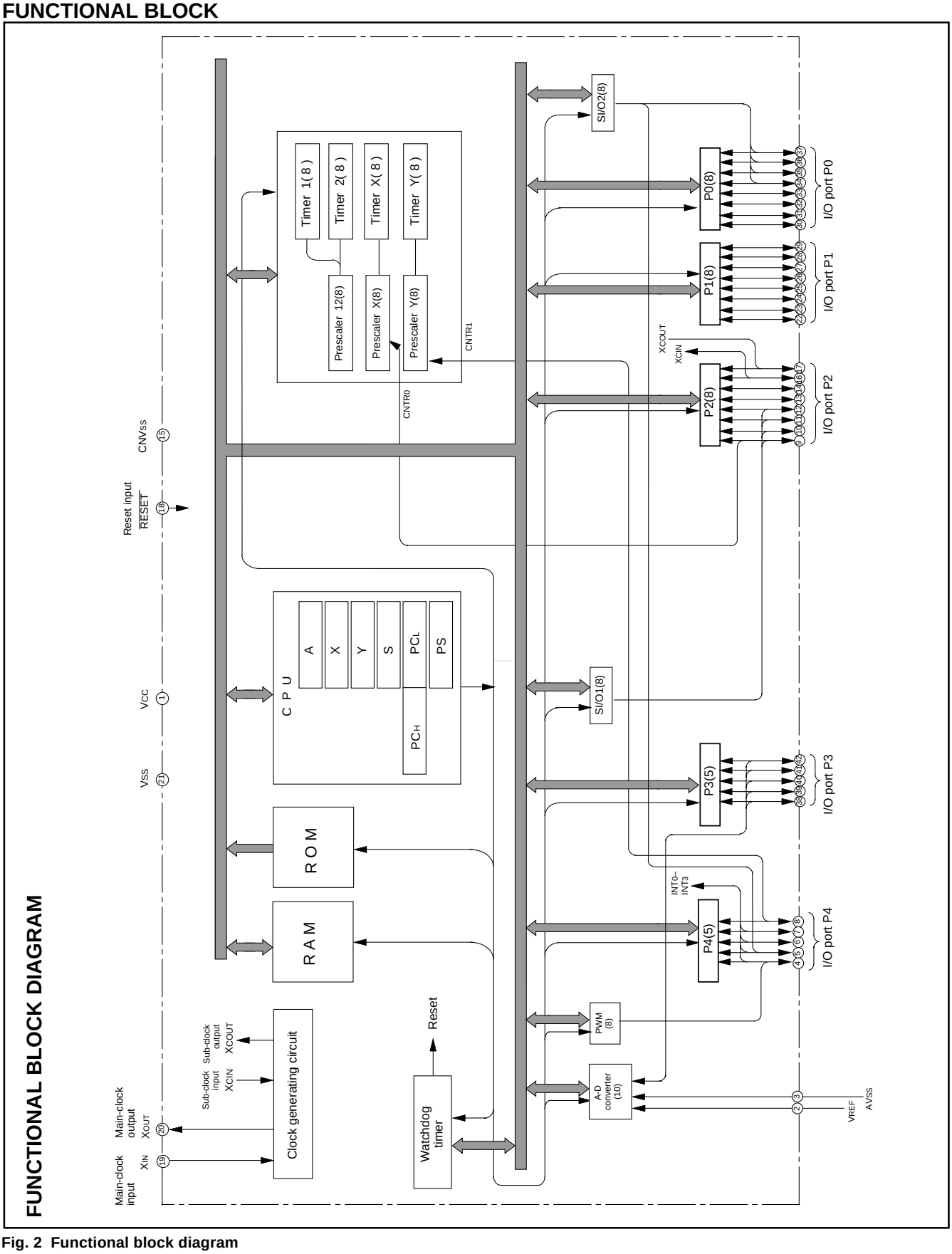


Fig. 2 Functional block diagram

PIN DESCRIPTION

Table 1 Pin description

Pin	Name	Functions	Function except a port function
V_{CC}, V_{SS}	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.	
$\overline{RESET}$	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/SIN2 P01/SOUT2 P02/SCLK2 P03/$\overline{SRDY2}$	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.	• Serial I/O2 function pin
P04–P07			
P10–P17	I/O port P1	•P10 to P17 (8 bits) are enabled to output large current for LED drive.	
P20/XCOUT P21/XCIN	I/O port P2	•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. •P20, P21, P24 to P27: CMOS3-state output structure. •P22, P23: N-channel open-drain structure.	• Sub-clock generating circuit I/O pins (connect a resonator)
P22 P23			
P24/RxD P25/TxD			• Serial I/O1 function pin
P26/SCLK1			
P27/CNTR0/$\overline{SRDY1}$			• Serial I/O1 function pin/ Timer X function pin
P30/AN0– P34/AN4	I/O port P3	•8-bit CMOS I/O port with the same function as port P0. •CMOS compatible input level. •CMOS 3-state output structure.	• A-D converter input pin
P40/CNTR1	I/O port P4	•8-bit CMOS I/O port with the same function as port P0. •CMOS compatible input level. •CMOS 3-state output structure.	• Timer Y function pin
P41/INT0 P42/INT1			• Interrupt input pins
P43/INT2/SCMP2			• Interrupt input pin • SCMP2 output pin
P44/INT3/PWM			• Interrupt input pin • PWM output pin

PART NUMBERING

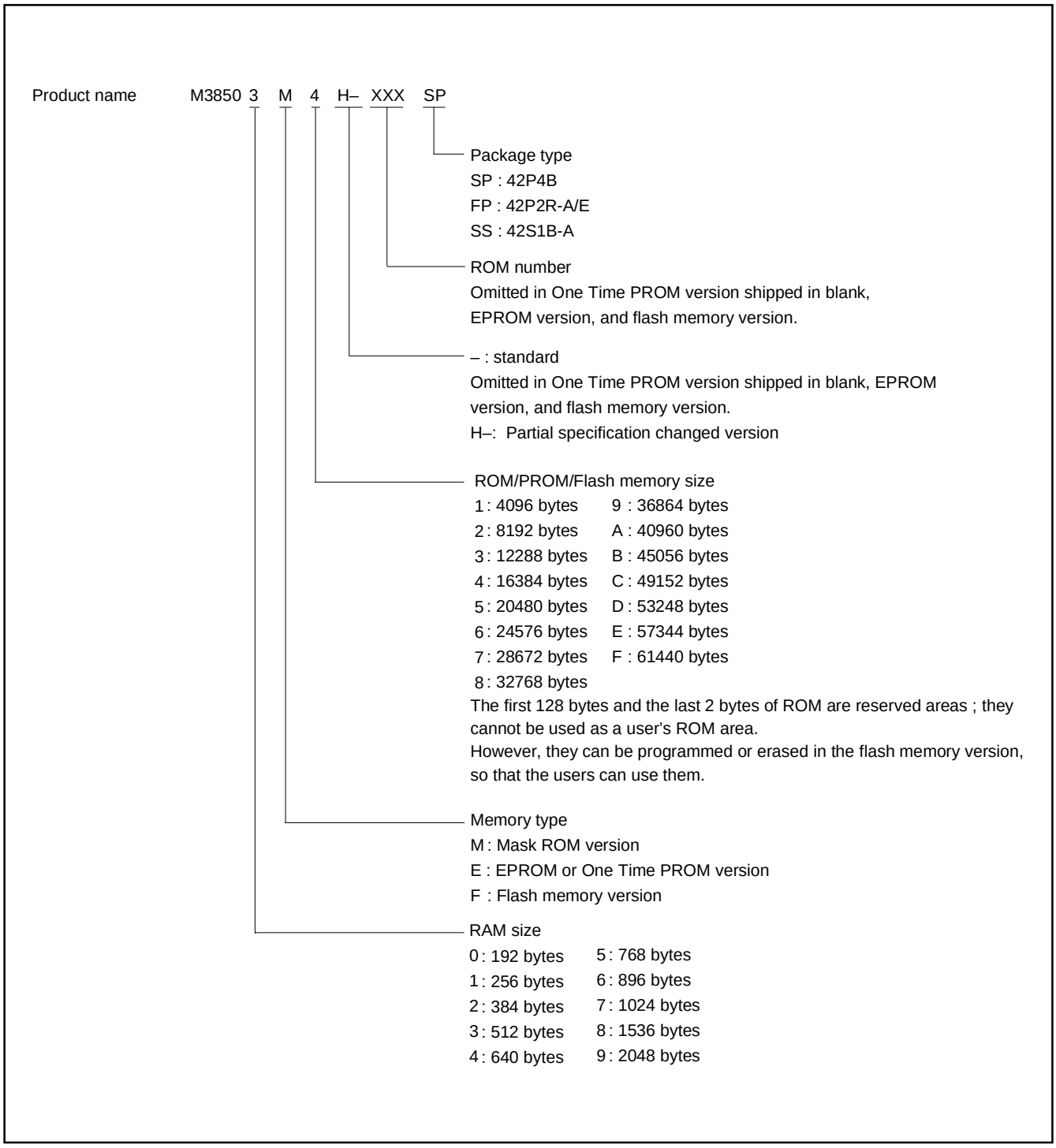


Fig. 3 Part numbering

GROUP EXPANSION

Renesas Technology plans to expand the 3850 group (spec. H) as follows.

Memory Type

Support for mask ROM, One Time PROM, and flash memory versions.

Memory Size

Flash memory size ..... 32 K bytes
One Time PROM size ..... 24 K bytes
Mask ROM size ..... 8 K to 32 K bytes
RAM size ..... 512 to 1 K bytes

Packages

42P4B ..... 42-pin shrink plastic-molded DIP
42P2R-A/E ..... 42-pin plastic-molded SSOP
42S1B-A ..... 42-pin shrink ceramic DIP (EPROM version)

Memory Expansion Plan

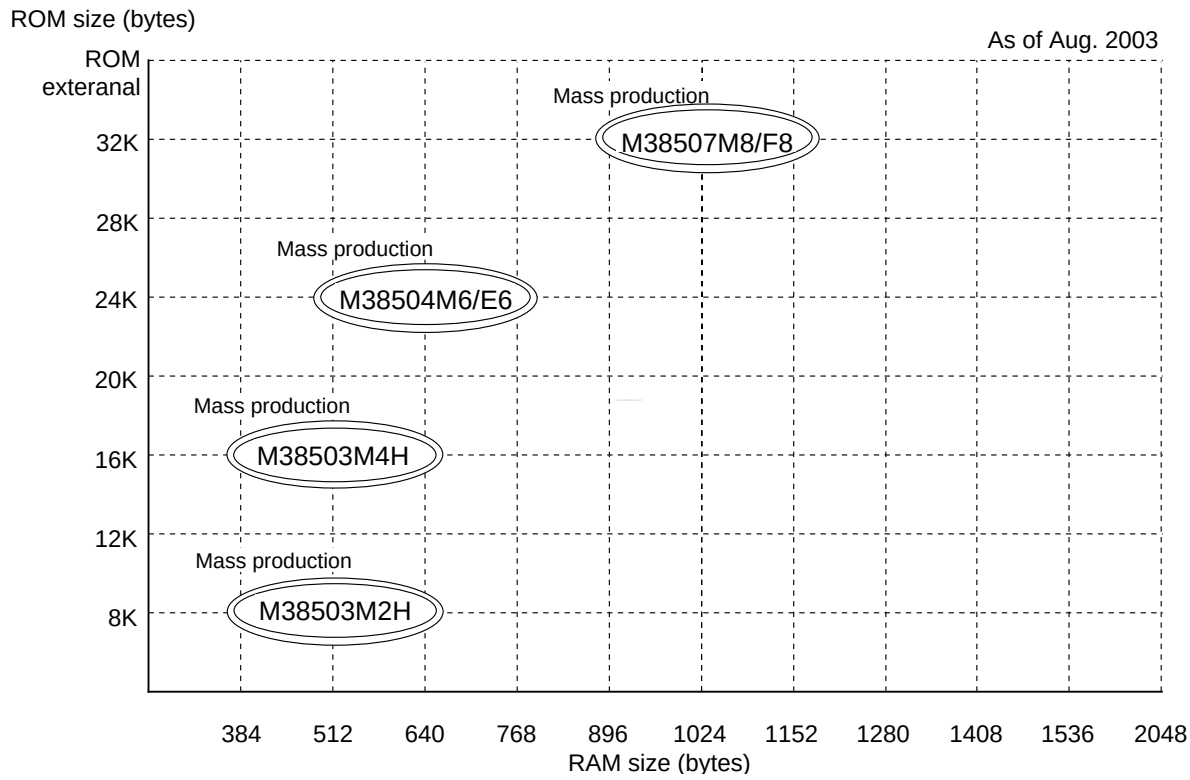


Fig. 4 Memory expansion plan

Currently support products are listed below.

Table 2 Support products **As of Aug. 2003**

Product name	ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38503M2H-XXXSP	8192 (8062)	512	42P4B	Mask ROM version
M38503M2H-XXXFP			42P2R-A/E	Mask ROM version
M38503M4H-XXXSP	16384 (16254)	512	424P4B	Mask ROM version
M38503M4H-XXXFP			42P2R-A/E	Mask ROM version
M38504M6-XXXSP	24576 (24446)	640	424P4B	Mask ROM version
M38504E6-XXXSP				One Time PROM version
M38504E6SP				One Time PROM version (blank)
M38504E6SS			42S1B-A	EPROM version
M38504M6-XXXFP			42P2R-A/E	Mask ROM version
M38504E6-XXXFP				One Time PROM version
M38504E6FP				One Time PROM version (blank)
M38507M8-XXXSP	32768 (32638)	1024	42P4B	Mask ROM version
M38507M8-XXXFP			42P2R-A/E	
M38507F8SP			424P4B	Flash memory version
M38507F8FP			42P2R-A/E	

Table 3 3850 group (standard) and 3850 group (spec. H) corresponding products

3850 group (standard) **(Note)**	3850 group (spec. H)
M38503M2-XXXFP/SP	M38503M2H-XXXFP/SP
M38503M4-XXXFP/SP	M38503M4H-XXXFP/SP
M38503E4-XXXFP/SP	M38504M6-XXXFP/SP
M38503E4FP/SP	M38504E6-XXXFP/SP
M38503E4SS	M38504E6FP/SP
	M38504E6SS
	M38507M8-XXXFP/SP
	M38507F8FP/SP

Note: The user who is using the 3850 Group (standard) needs to refer to not this manual but “3850/3851 Group User’s Manual”.

Table 4 Differences between 3850 group (standard) and 3850 group (spec. H)

	3850 group (standard)	3850 group (spec. H)
Serial I/O	1: Serial I/O (UART or Clock-synchronized)	2: Serial I/O1 (UART or Clock-synchronized) Serial I/O2 (Clock-synchronized)
A-D converter	Unserviceable in low-speed mode	Serviceable in low-speed mode
Large current port	5: $P1_3$–$P1_7$	8: $P1_0$–$P1_7$

Notes on differences between 3850 group (standard) and 3850 group (spec. H)

(1) The absolute maximum ratings of 3850 group (spec. H) is smaller than that of 3850 group (standard).
- •Power source voltage Vcc = –0.3 to 6.5 V
- •CNVss input voltage V_I = –0.3 to Vcc +0.3 V

(2) The oscillation circuit constants of X_{IN}-X_{OUT}, X_{CIN}-X_{COUT} may be some differences between 3850 group (standard) and 3850 group (spec. H).

(3) Do not write any data to the reserved area and the reserved bit. (Do not change the contents after reset.)

(4) Fix bit 3 of the CPU mode register to “1”.

(5) Be sure to perform the termination of unused pins.

FUNCTIONAL DESCRIPTION CENTRAL PROCESSING UNIT (CPU)

The 3850 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 6.

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

[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.

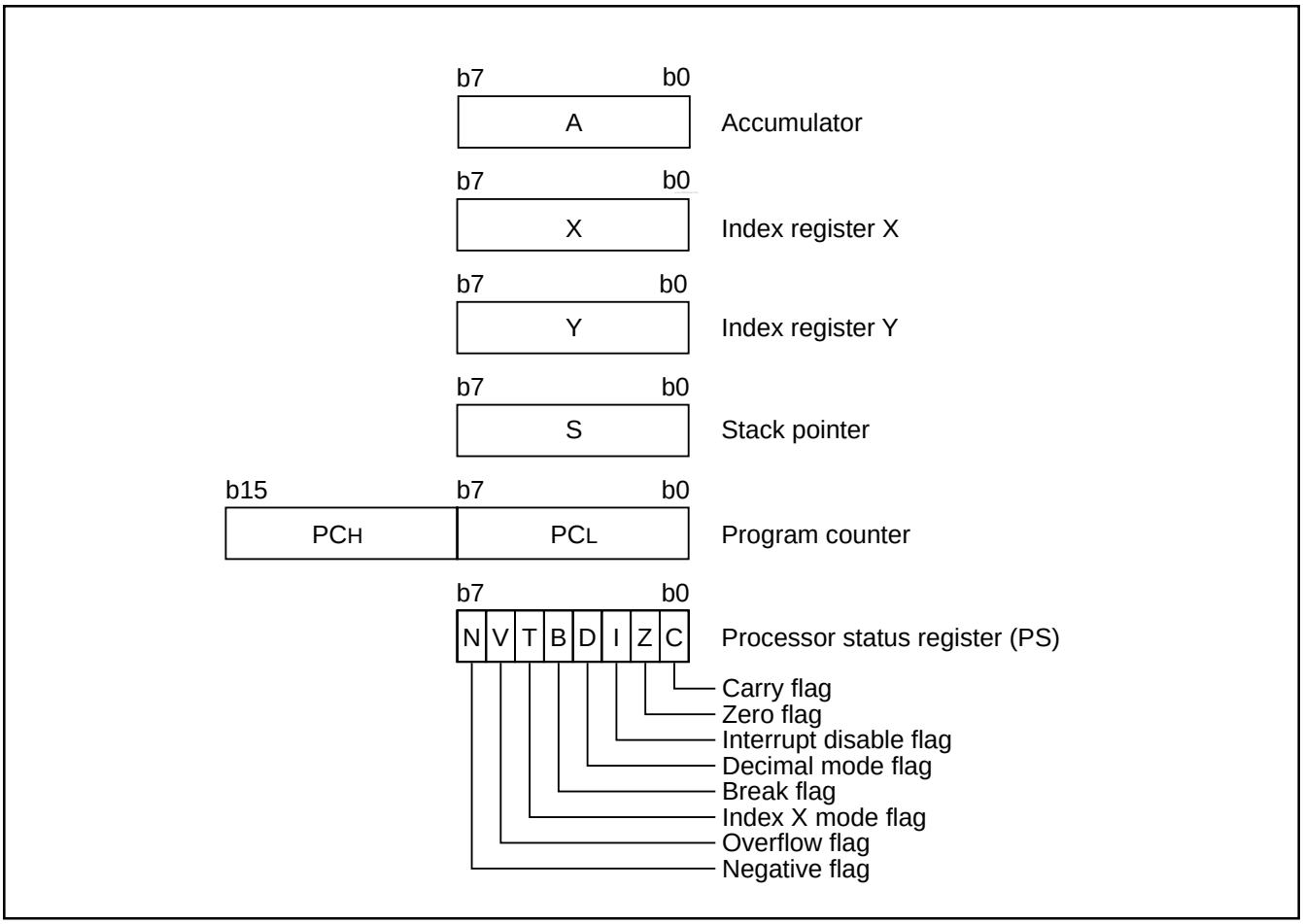


Fig. 5 740 Family CPU register structure

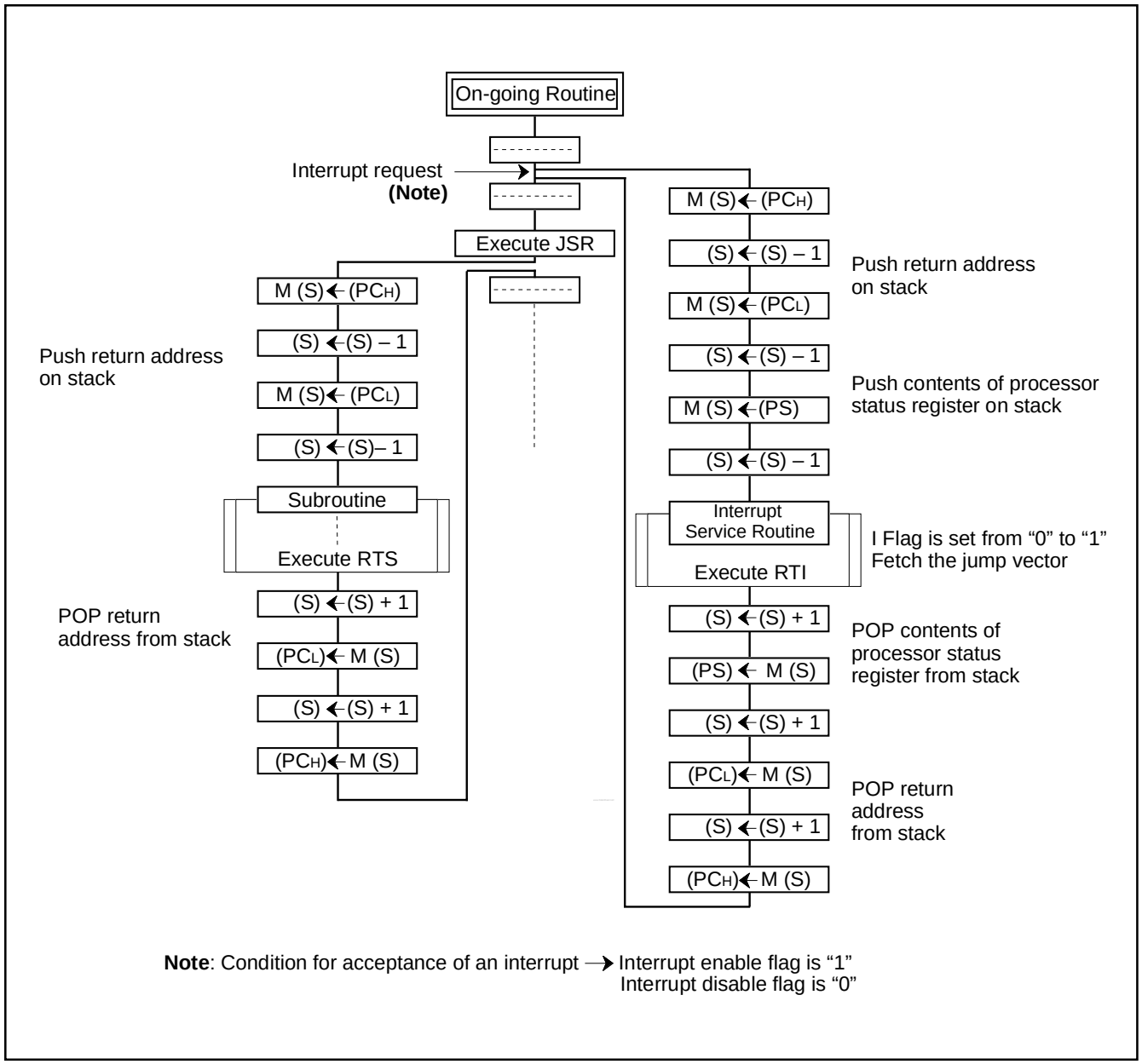


Fig. 6 Register push and pop at interrupt generation and subroutine call

Table 5 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

FUNCTIONAL DESCRIPTION

[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

•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 6 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.

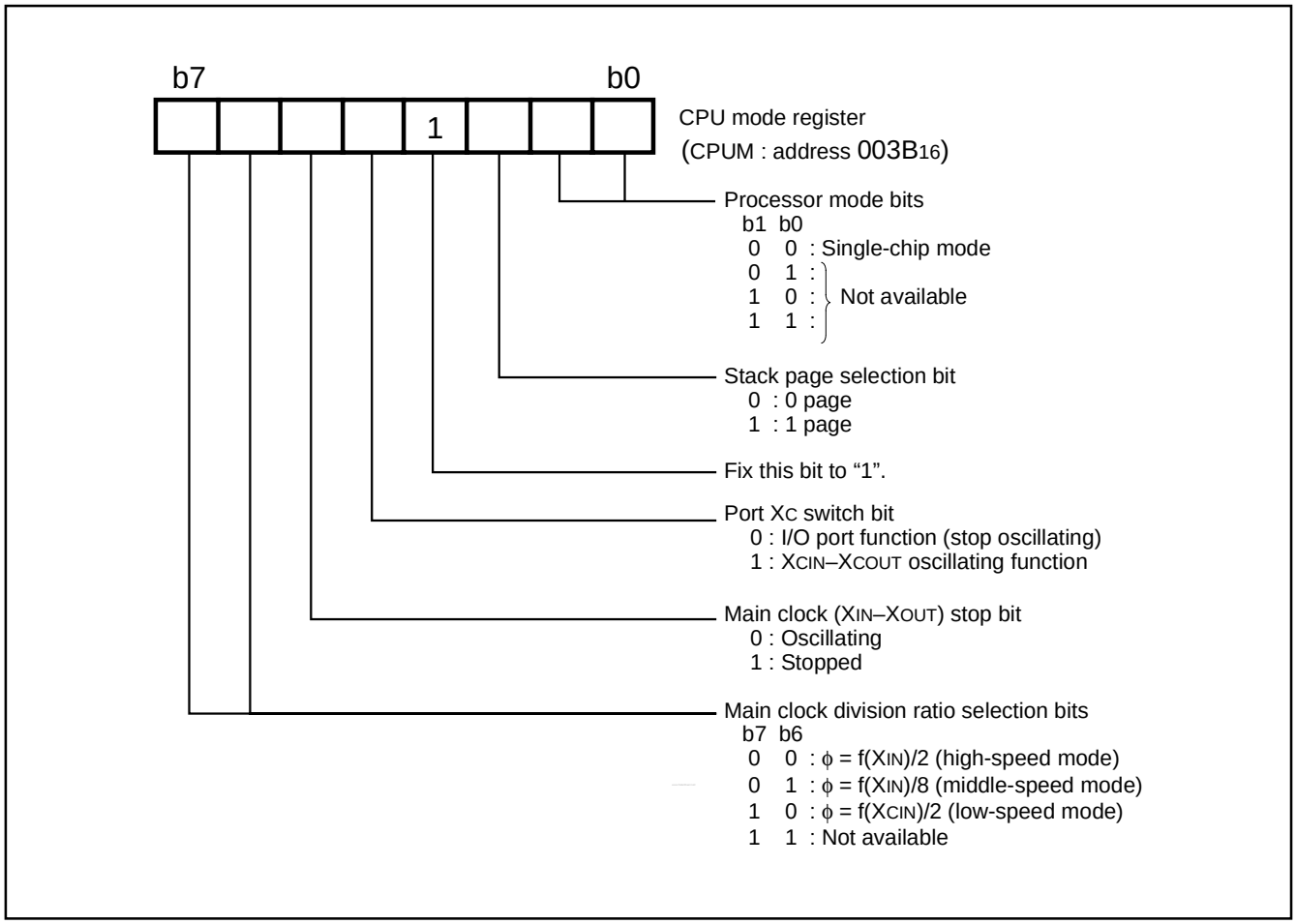


Fig. 7 Structure of CPU mode register

FUNCTIONAL DESCRIPTION

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

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

ROM

The first 128 bytes and the last 2 bytes of ROM are reserved for device testing and the rest is user area for storing programs.

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

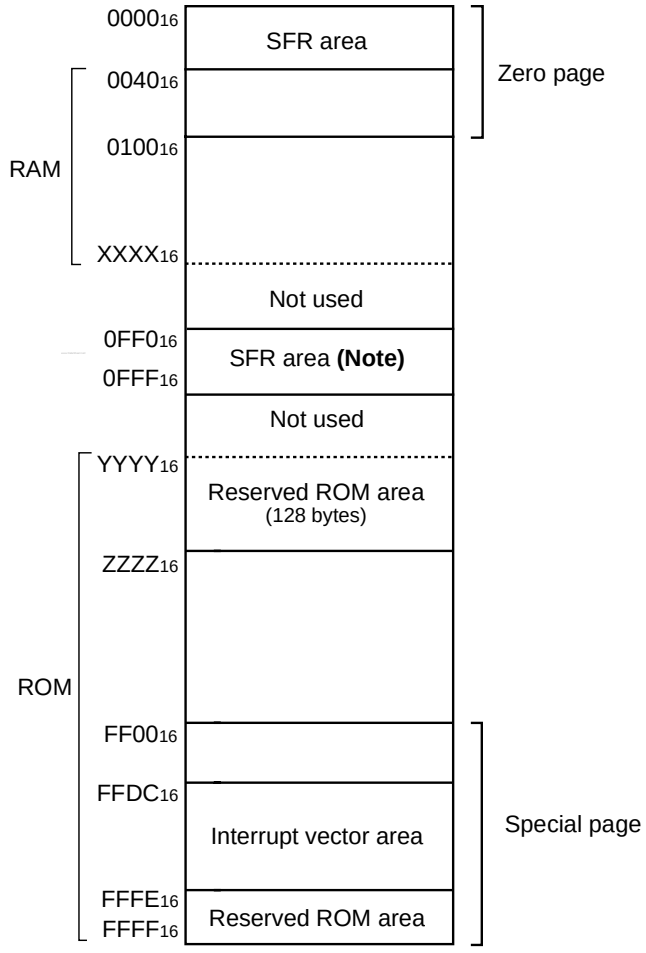
Access to this area with only 2 bytes is possible in the special page addressing mode.

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	Address ZZZZ16
4096	F00016	F08016
8192	E00016	E08016
12288	D00016	D08016
16384	C00016	C08016
20480	B00016	B08016
24576	A00016	A08016
28672	900016	908016
32768	800016	808016
36864	700016	708016
40960	600016	608016
45056	500016	508016
49152	400016	408016
53248	300016	308016
57344	200016	208016
61440	100016	108016



Note: Flash memory version only

Fig. 8 Memory map diagram

0000₁₆	Port P0 (P0)	0020₁₆	Prescaler 12 (PRE12)
0001₁₆	Port P0 direction register (P0D)	0021₁₆	Timer 1 (T1)
0002₁₆	Port P1 (P1)	0022₁₆	Timer 2 (T2)
0003₁₆	Port P1 direction register (P1D)	0023₁₆	Timer XY mode register (TM)
0004₁₆	Port P2 (P2)	0024₁₆	Prescaler X (PREX)
0005₁₆	Port P2 direction register (P2D)	0025₁₆	Timer X (TX)
0006₁₆	Port P3 (P3)	0026₁₆	Prescaler Y (PREY)
0007₁₆	Port P3 direction register (P3D)	0027₁₆	Timer Y (TY)
0008₁₆	Port P4 (P4)	0028₁₆	Timer count source selection register (TCSS)
0009₁₆	Port P4 direction register (P4D)	0029₁₆	
000A₁₆		002A₁₆	
000B₁₆		002B₁₆	Reserved ✻
000C₁₆		002C₁₆	Reserved ✻
000D₁₆		002D₁₆	Reserved ✻
000E₁₆		002E₁₆	Reserved ✻
000F₁₆		002F₁₆	Reserved ✻
0010₁₆		0030₁₆	Reserved ✻
0011₁₆		0031₁₆	Reserved ✻
0012₁₆	Reserved ✻	0032₁₆	
0013₁₆	Reserved ✻	0033₁₆	
0014₁₆	Reserved ✻	0034₁₆	A-D control register (ADCON)
0015₁₆	Serial I/O2 control register 1 (SIO2CON1)	0035₁₆	A-D conversion low-order register (ADL)
0016₁₆	Serial I/O2 control register 2 (SIO2CON2)	0036₁₆	A-D conversion high-order register (ADH)
0017₁₆	Serial I/O2 register (SIO2)	0037₁₆	Reserved ✻
0018₁₆	Transmit/Receive buffer register (TB/RB)	0038₁₆	MISRG
0019₁₆	Serial I/O1 status register (SIOSTS)	0039₁₆	Watchdog timer control register (WDTCON)
001A₁₆	Serial I/O1 control register (SIOCON)	003A₁₆	Interrupt edge selection register (INTEDGE)
001B₁₆	UART control register (UARTCON)	003B₁₆	CPU mode register (CPUM)
001C₁₆	Baud rate generator (BRG)	003C₁₆	Interrupt request register 1 (IREQ1)
001D₁₆	PWM control register (PWMCON)	003D₁₆	Interrupt request register 2 (IREQ2)
001E₁₆	PWM prescaler (PREPWM)	003E₁₆	Interrupt control register 1 (ICON1)
001F₁₆	PWM register (PWM)	003F₁₆	Interrupt control register 2 (ICON2)
		0FFE₁₆	Flash memory control register (FMCR)

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

Fig. 9 Memory map of special function register (SFR)

FUNCTIONAL DESCRIPTION

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.

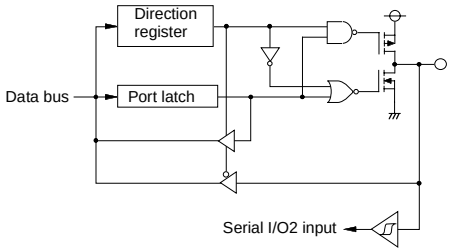
If data is read from a pin which is set to output, the value of the port output latch is read, not the value of the pin itself. Pins set to input are floating. If a pin set to input is written to, only the port output latch is written to and the pin remains floating.

Table 7 I/O port function

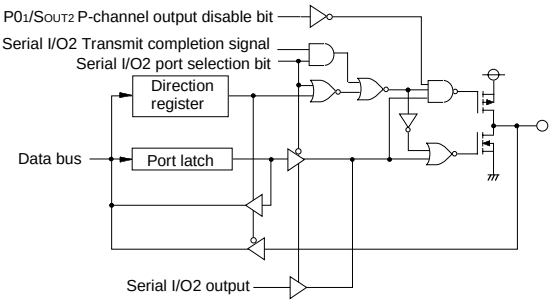
Pin	Name	Input/Output	I/O Structure	Non-Port Function	Related SFRs	Ref.No.
P0$_0$/SIN$_2$	Port P0	Input/output, individual bits	CMOS compatible input level CMOS 3-state output	Serial I/O2 function I/O	Serial I/O2 control register	(1)
P0$_1$/SOUT$_2$						(2)
P0$_2$/SCLK$_2$						(3)
P0$_3$/$\overline{\text{SRDY}_2}$						(4)
P0$_4$–P0$_7$						(5)
P1$_0$–P1$_7$	Port P1					
P2$_0$/XCOUT	Port P2			Sub-clock generating circuit	CPU mode register	(6)
P2$_1$/XCIN						(7)
P2$_2$ P2$_3$			CMOS compatible input level N-channel open-drain output			(8)
P2$_4$/RxD			CMOS compatible input level CMOS 3-state output	Serial I/O1 function I/O	Serial I/O1 control register	(9)
P2$_5$/TxD						(10)
P2$_6$/SCLK$_1$						(11)
P2$_7$/CNTR$_0$/$\overline{\text{SRDY}_1}$				Serial I/O1 function I/O Timer X function I/O	Serial I/O1 control register Timer XY mode register	(12)
P3$_0$/AN$_0$– P3$_4$/AN$_4$	Port P3			A-D conversion input	A-D control register	(13)
P4$_0$/CNTR$_1$	Port P4			Timer Y function I/O	Timer XY mode register	(14)
P4$_1$/INT$_0$ P4$_2$/INT$_1$				External interrupt input	Interrupt edge selection register	(15)
P4$_3$/INT$_2$/SCMP$_2$				External interrupt input SCMP$_2$ output	Interrupt edge selection register Serial I/O2 control register	(16)
P4$_4$/INT$_3$/PWM				External interrupt input PWM output	Interrupt edge selection register PWM control register	(17)

Note: When bits 5 to 7 of Ports P3 and P4 are read out, the contents are undefined.

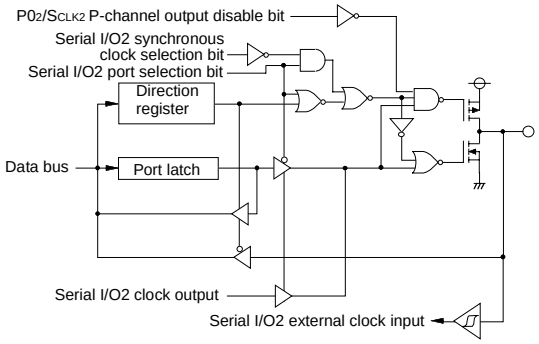
(1) Port $P0_0$



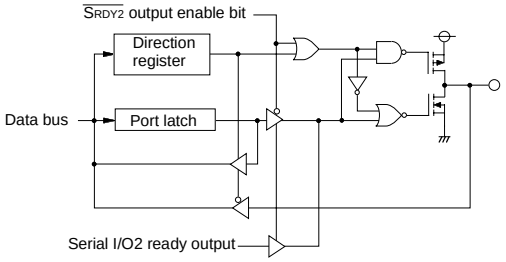
(2) Port $P0_1$



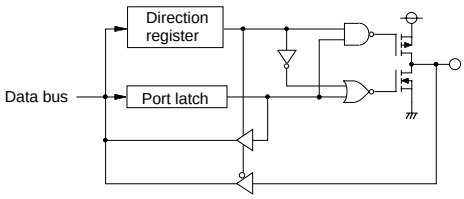
(3) Port $P0_2$



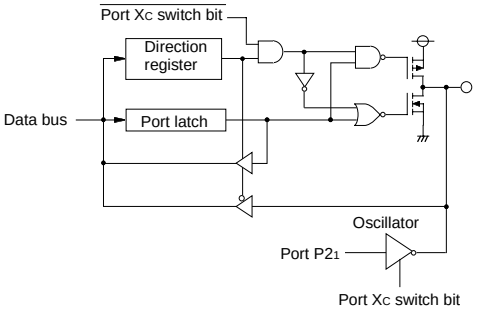
(4) Port $P0_3$



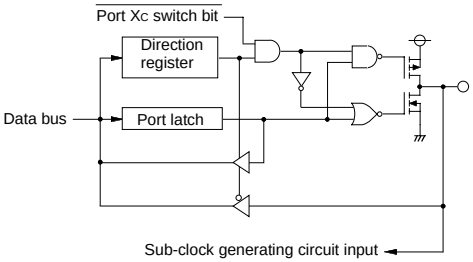
(5) Ports $P0_4$-$P0_7$,P1



(6) Port $P2_0$



(7) Port $P2_1$



(8) Ports $P2_2$,$P2_3$

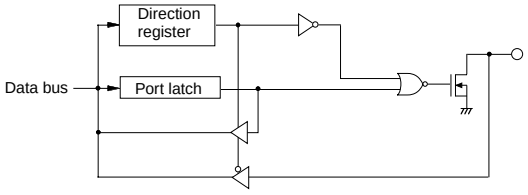
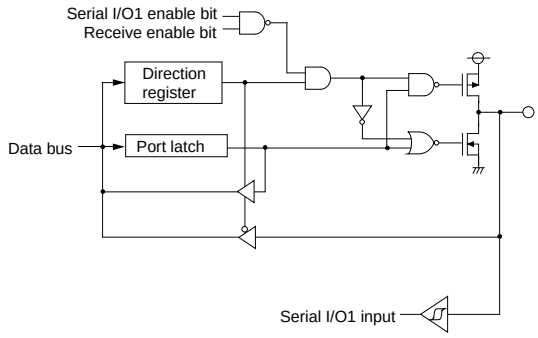
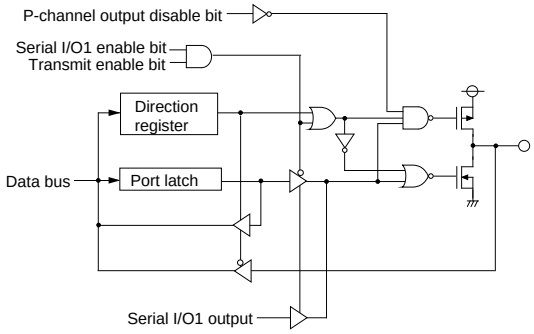


Fig. 10 Port block diagram (1)

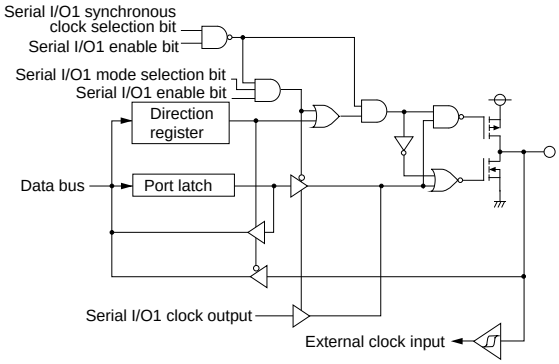
(9) Port $P2_4$



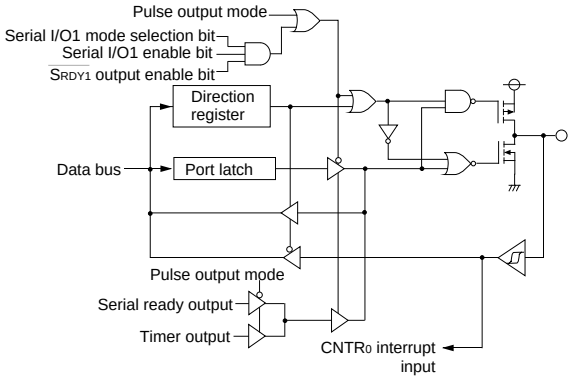
(10) Port $P2_5$



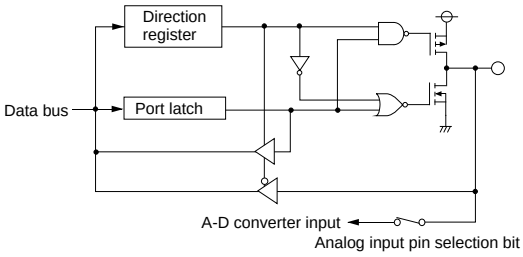
(11) Port $P2_6$



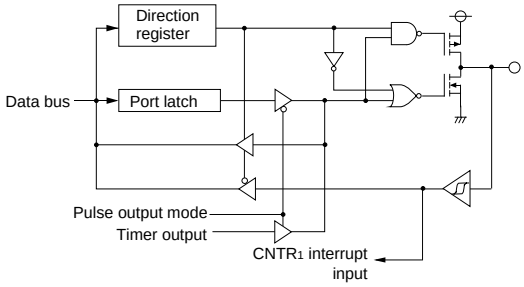
(12) Port $P2_7$



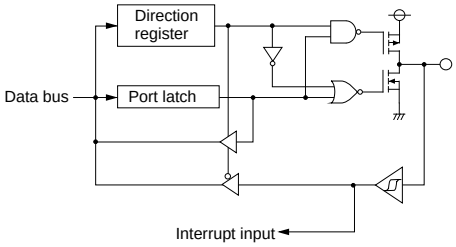
(13) Ports $P3_0$-$P3_4$



(14) Port $P4_0$



(15) Ports $P4_1$,$P4_2$



(16) Port $P4_3$

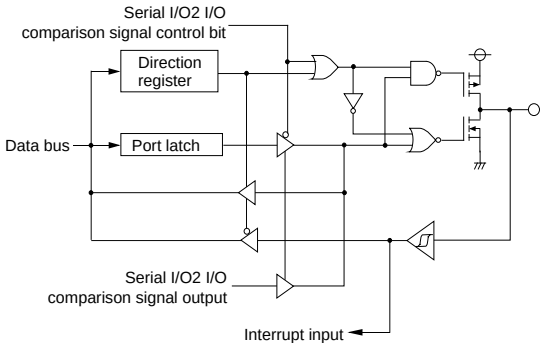


Fig. 11 Port block diagram (2)

(17) Port $P4_4$

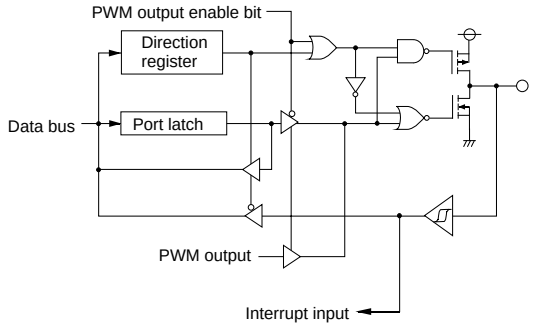


Fig. 12 Port block diagram (3)

INTERRUPTS

Interrupts occur by 15 sources among 15 sources: six external, eight 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 BRK instruction cannot be disabled with any flag or bit. The I (interrupt disable) flag disables all interrupts except the BRK instruction interrupt.

When several interrupts 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.

■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 $3A_{16}$)
Timer XY mode register (address 23_{16})

•When switching interrupt sources of an interrupt vector address where two or more interrupt sources are allocated
Related register: Interrupt edge selection register (address $3A_{16}$)

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 8 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)
Reserved	3	FFF916	FFF816	Reserved	
INT1	4	FFF716	FFF616	At detection of either rising or falling edge of INT1 input	External interrupt (active edge selectable)
INT2	5	FFF516	FFF416	At detection of either rising or falling edge of INT2 input	External interrupt (active edge selectable)
INT3/ Serial I/O2	6	FFF316	FFF216	At detection of either rising or falling edge of INT3 input/ At completion of serial I/O2 data reception/transmission	External interrupt (active edge selectable) Switch by Serial I/O2/INT3 interrupt source bit
Reserved	7	FFF116	FFF016	Reserved	
Timer X	8	FFEF16	FFEE16	At timer X underflow	
Timer Y	9	FFED16	FFEC16	At timer Y underflow	
Timer 1	10	FFEB16	FFEA16	At timer 1 underflow	STP release timer underflow
Timer 2	11	FFE916	FFE816	At timer 2 underflow	
Serial I/O1 reception	12	FFE716	FFE616	At completion of serial I/O1 data reception	Valid when serial I/O1 is selected
Serial I/O1 transmission	13	FFE516	FFE416	At completion of serial I/O1 transfer shift or when transmission buffer is empty	Valid when serial I/O1 is selected
CNTR0	14	FFE316	FFE216	At detection of either rising or falling edge of CNTR0 input	External interrupt (active edge selectable)
CNTR1	15	FFE116	FFE016	At detection of either rising or falling edge of CNTR1 input	External interrupt (active edge selectable)
A-D converter	16	FFDF16	FFDE16	At completion of A-D conversion	
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.

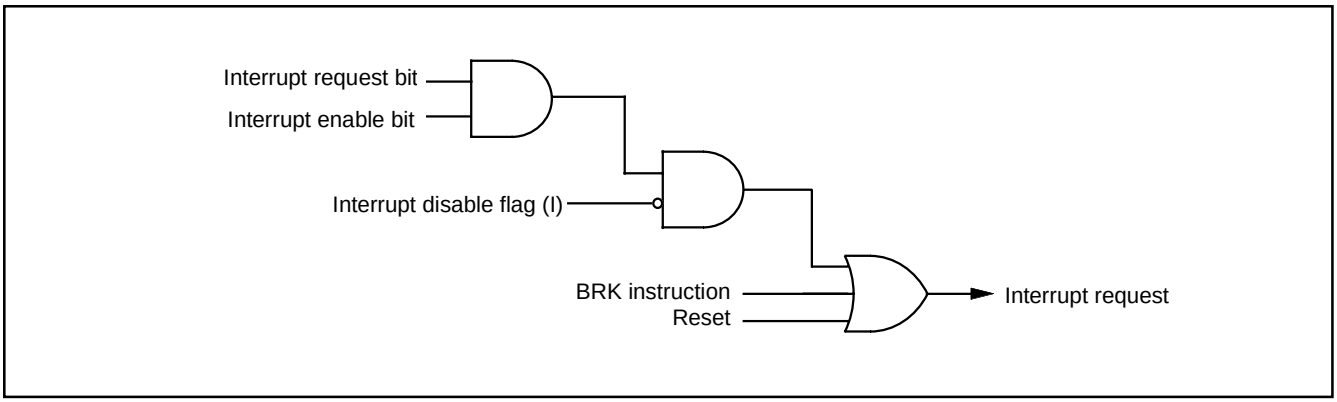


Fig. 13 Interrupt control

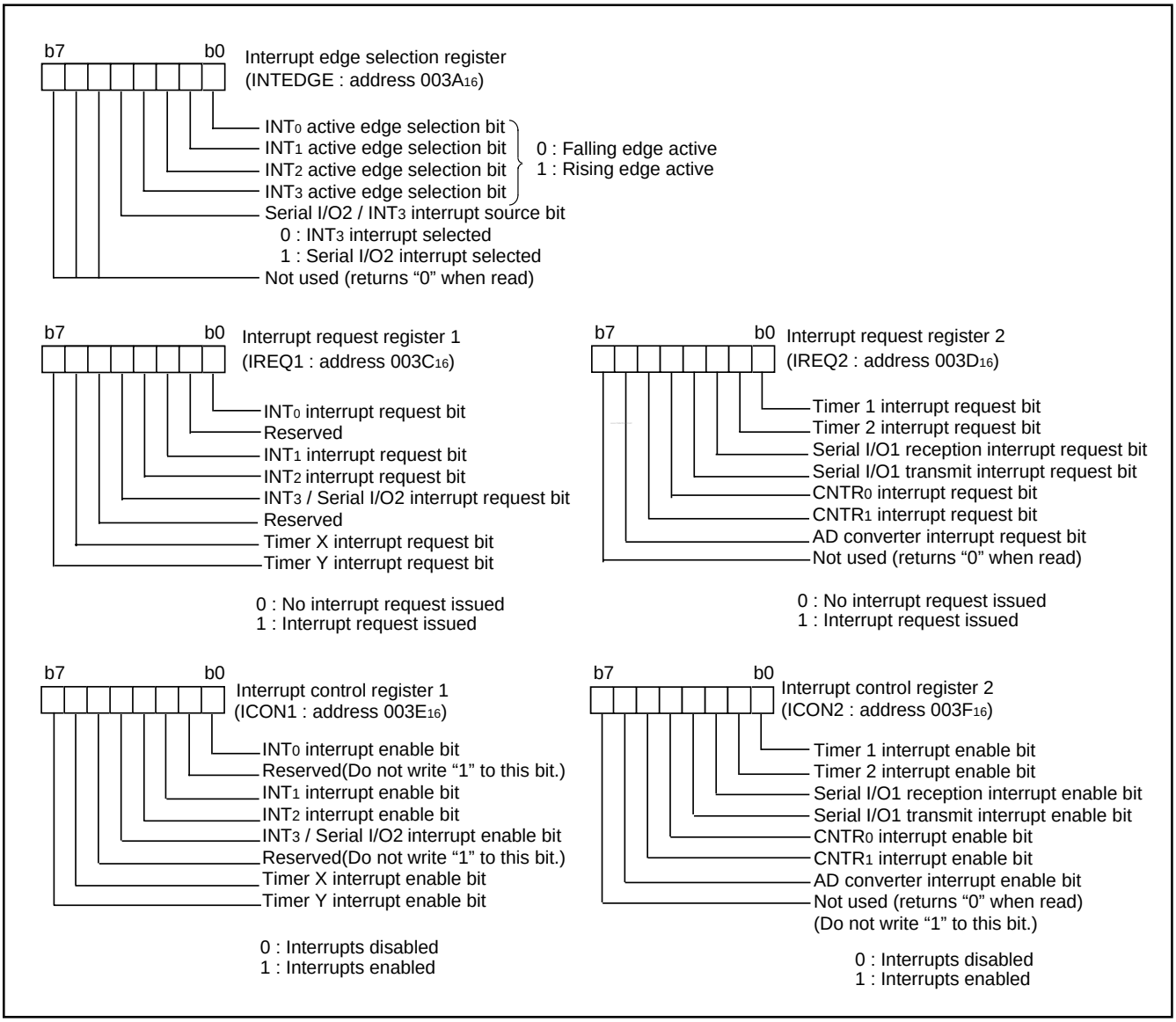


Fig. 14 Structure of interrupt-related registers

TIMERS

The 3850 group (spec. H) has four timers: timer X, timer Y, timer 1, and timer 2.

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 count down. When the timer reaches "00_{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 a timer underflows, the interrupt request bit corresponding to that timer is set to "1".

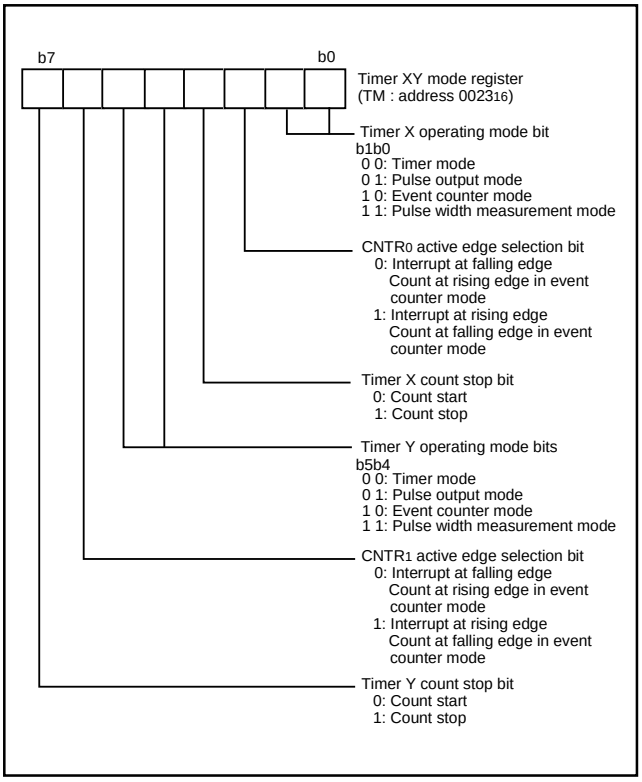


Fig. 15 Structure of timer XY mode register

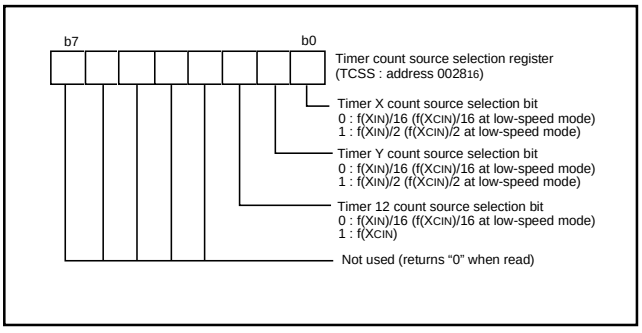


Fig. 16 Structure of timer count source selection register

Timer 1 and Timer 2

The count source of prescaler 12 is the oscillation frequency which is selected by timer 12 count source selection bit. The output of prescaler 12 is counted by timer 1 and timer 2, and a timer underflow sets the interrupt request bit.

Timer X and Timer Y

Timer X and Timer Y can each select in one of four operating modes by setting the timer XY mode register.

(1) Timer Mode

The timer counts the count source selected by Timer count source selection bit.

(2) Pulse Output Mode

The timer counts the count source selected by Timer count source selection bit. Whenever the contents of the timer reach "00_{16}", the signal output from the $CNTR_0$ (or $CNTR_1$) pin is inverted. If the $CNTR_0$ (or $CNTR_1$) active edge selection bit is "0", output begins at " H".

If it is "1", output starts at "L". When using a timer in this mode, set the corresponding port $P2_7$ (or port $P4_0$) direction register to output mode.

(3) Event Counter Mode

Operation in event counter mode is the same as in timer mode, except that the timer counts signals input through the $CNTR_0$ or $CNTR_1$ pin.

When the $CNTR_0$ (or $CNTR_1$) active edge selection bit is "0", the rising edge of the $CNTR_0$ (or $CNTR_1$) pin is counted.

When the $CNTR_0$ (or $CNTR_1$) active edge selection bit is "1", the falling edge of the $CNTR_0$ (or $CNTR_1$) pin is counted.

(4) Pulse Width Measurement Mode

If the $CNTR_0$ (or $CNTR_1$) active edge selection bit is "0", the timer counts the selected signals by the count source selection bit while the $CNTR_0$ (or $CNTR_1$) pin is at "H". If the $CNTR_0$ (or $CNTR_1$) active edge selection bit is "1", the timer counts it while the $CNTR_0$ (or $CNTR_1$) pin is at "L".

The count can be stopped by setting "1" to the timer X (or timer Y) count stop bit in any mode. The corresponding interrupt request bit is set each time a timer underflows.

■Note

When switching the count source by the timer 12, X and Y count source bit, the value of timer count is altered in unconsiderable amount owing to generating of a thin pulses in the count input signals.

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

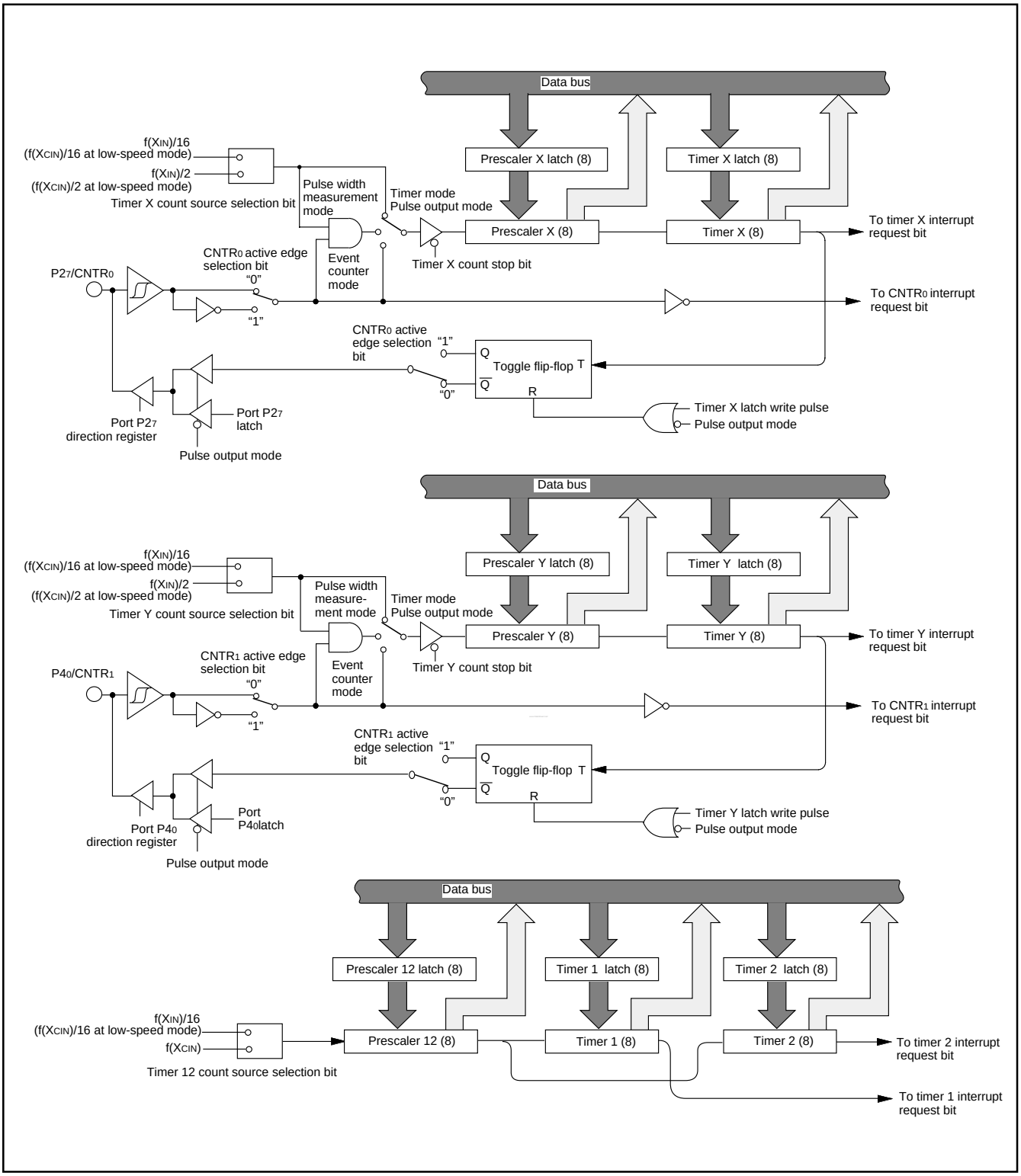


Fig. 17 Block diagram of timer X, timer Y, timer 1, and timer 2

SERIAL I/O
●SERIAL I/O1

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

(1) Clock Synchronous Serial I/O Mode

Clock synchronous serial I/O mode can be selected by setting the serial I/O1 mode selection bit of the serial I/O1 control register (bit 6 of address $001A_{16}$) 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 TB/RB.

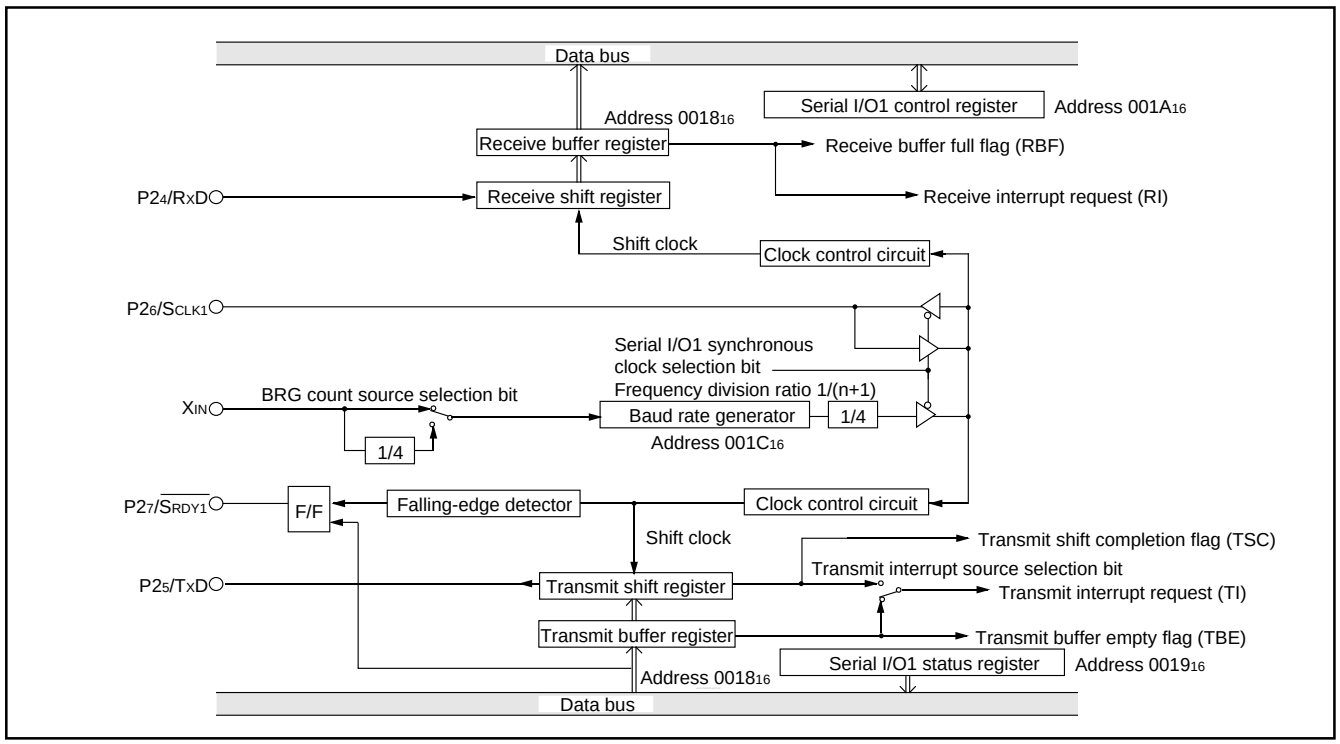
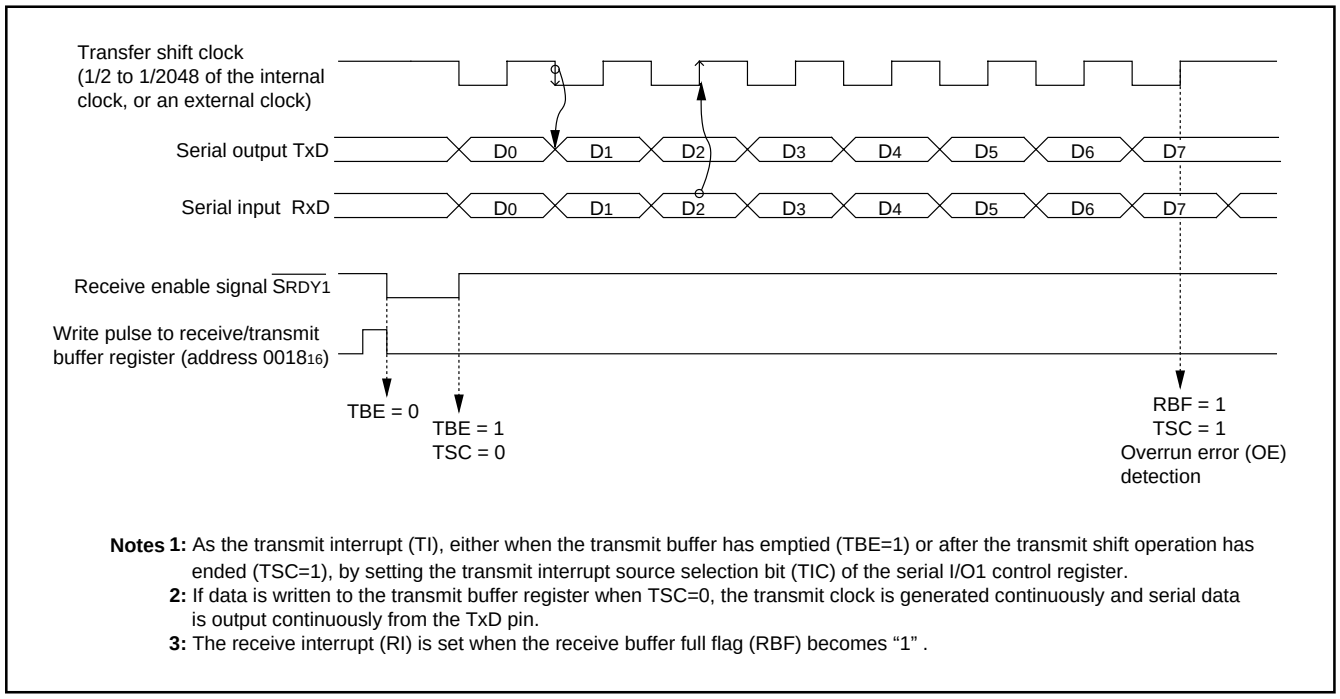


Fig. 18 Block diagram of clock synchronous serial I/O1



Notes 1: As the transmit interrupt (TI), 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. 19 Operation of clock synchronous serial I/O1 function

(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 (b6) 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 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.

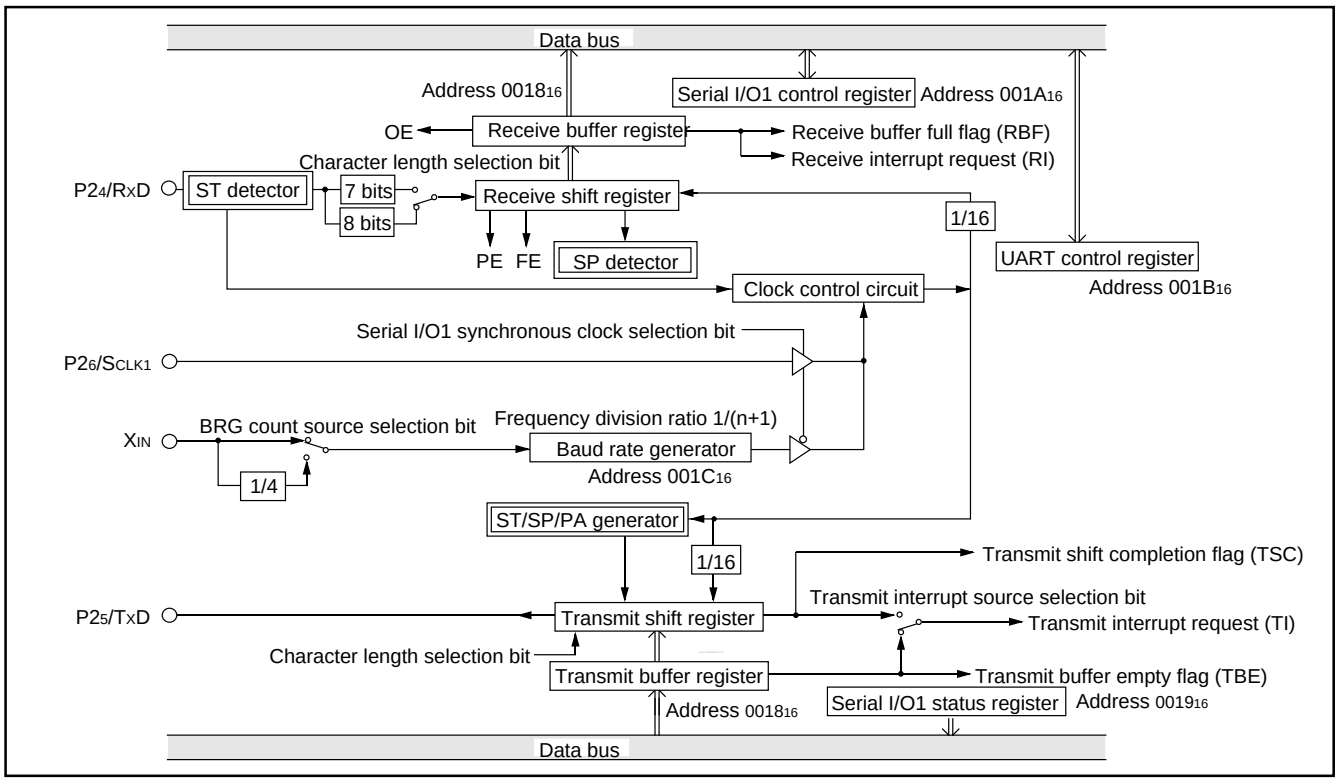
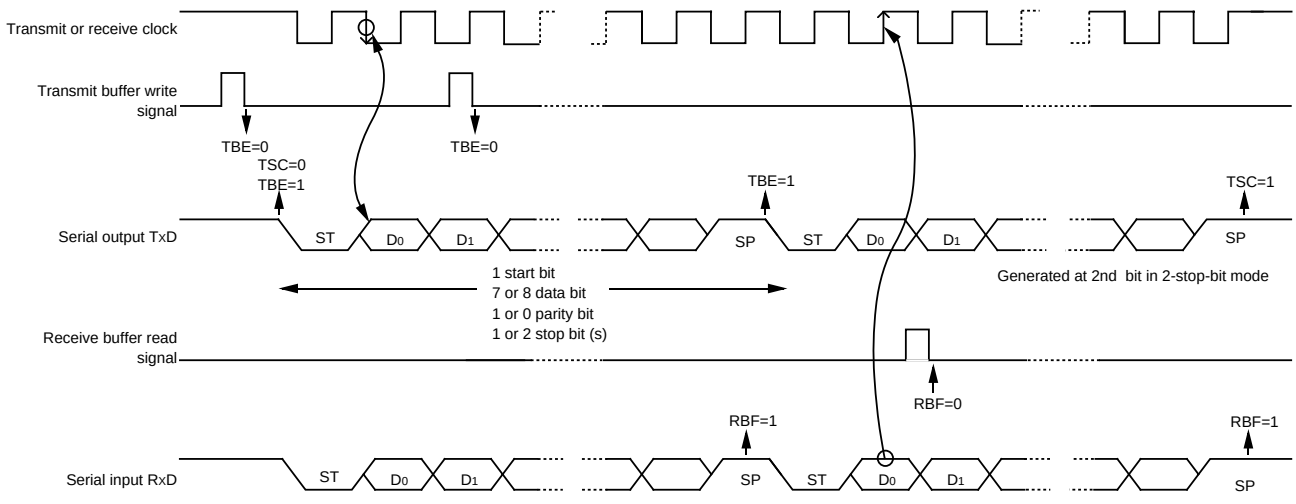


Fig. 20 Block diagram of UART serial I/O1



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 is necessary until changing to TSC=0.

Fig. 21 Operation of UART serial I/O1 function

[Transmit Buffer Register/Receive Buffer Register (TB/RB)] 0018$_{16}$

The transmit buffer register and the receive buffer register 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".

[Serial I/O1 Status Register (SIOSTS)] 0019$_{16}$

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".

[Serial I/O1 Control Register (SIOCON)] 001A$_{16}$

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

[UART Control Register (UARTCON)] 001B$_{16}$

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 P25/TXD pin.

[Baud Rate Generator (BRG)] 001C$_{16}$

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.

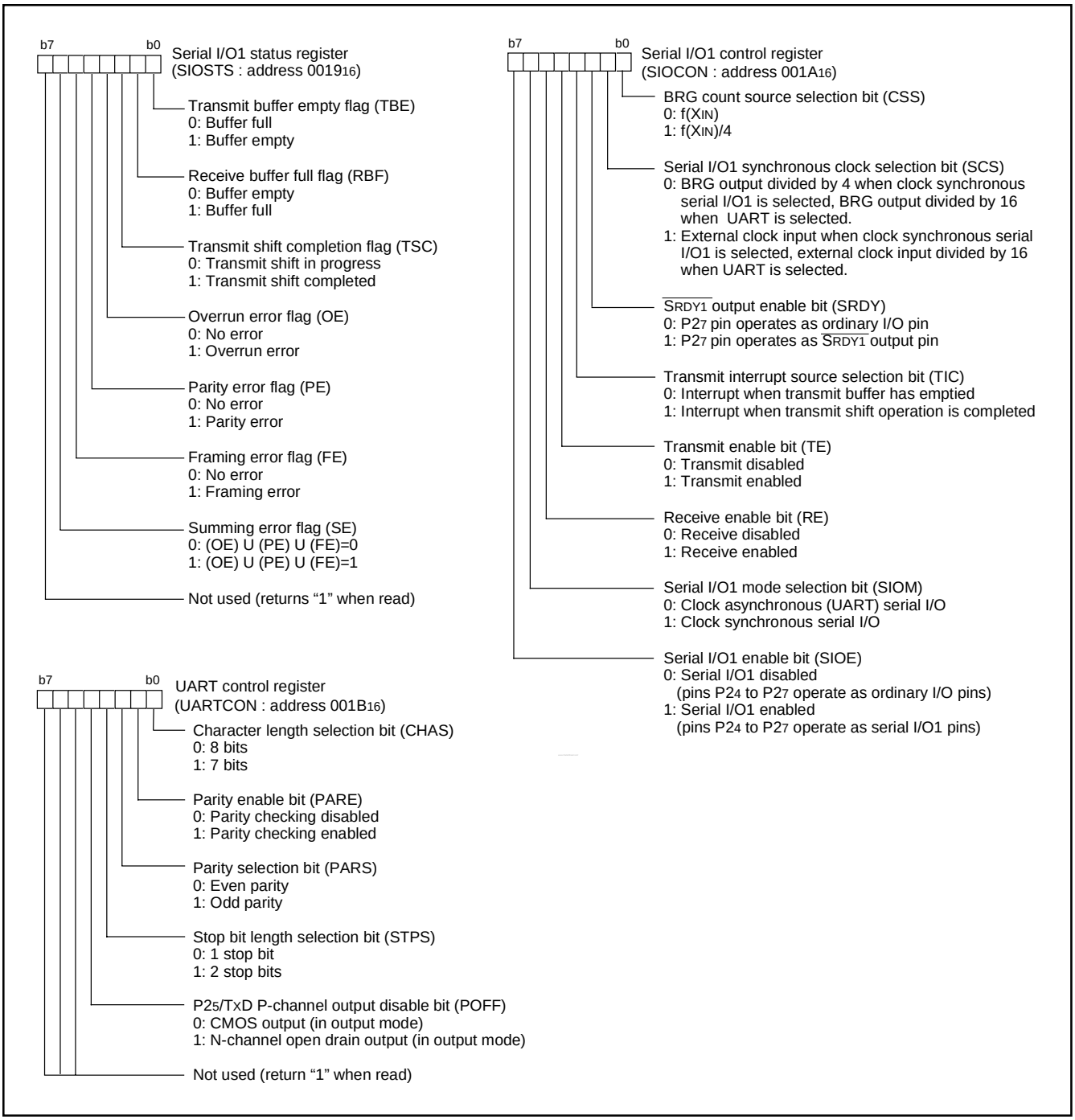


Fig. 22 Structure of serial I/O1 control registers

■Notes on serial I/O

When setting the transmit enable bit of serial I/O1 to "1", the serial I/O1 transmit interrupt request bit is automatically set to "1". When not requiring the interrupt occurrence synchronized with the transmission enalbed, take the following sequence.

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

●SERIAL I/O2

The serial I/O2 can be operated only as the clock synchronous type. As a synchronous clock for serial transfer, either internal clock or external clock can be selected by the serial I/O2 synchronous clock selection bit (b6) of serial I/O2 control register 1.

The internal clock incorporates a dedicated divider and permits selecting 6 types of clock by the internal synchronous clock selection bits (b2, b1, b0) of serial I/O2 control register 1.

Regarding S_{OUT2} and S_{CLK2} being output pins, either CMOS output format or N-channel open-drain output format can be selected by the $P0_1/S_{OUT2}$, $P0_2/S_{CLK2}$ P-channel output disable bit (b7) of serial I/O2 control register 1.

When the internal clock has been selected, a transfer starts by a write signal to the serial I/O2 register (address 0017_{16}). After completion of data transfer, the level of the S_{OUT2} pin goes to high impedance automatically but bit 7 of the serial I/O2 control register 2 is not set to "1" automatically.

When the external clock has been selected, the contents of the serial I/O2 register is continuously sifted while transfer clocks are input. Accordingly, control the clock externally. Note that the S_{OUT2} pin does not go to high impedance after completion of data transfer.

To cause the S_{OUT2} pin to go to high impedance in the case where the external clock is selected, set bit 7 of the serial I/O2 control register 2 to "1" when S_{CLK2} is "H" after completion of data transfer. After the next data transfer is started (the transfer clock falls), bit 7 of the serial I/O2 control register 2 is set to "0" and the S_{OUT2} pin is put into the active state.

Regardless of the internal clock to external clock, the interrupt request bit is set after the number of bits (1 to 8 bits) selected by the optional transfer bit is transferred. In case of a fractional number of bits less than 8 bits as the last data, the received data to be stored in the serial I/O2 register becomes a fractional number of bits close to MSB if the transfer direction selection bit of serial I/O2 control register 1 is LSB first, or a fractional number of bits close to LSB if the said bit is MSB first. For the remaining bits, the previously received data is shifted.

At transmit operation using the clock synchronous serial I/O, the S_{CMP2} signal can be output by comparing the state of the transmit pin S_{OUT2} with the state of the receive pin S_{IN2} in synchronization with a rise of the transfer clock. If the output level of the S_{OUT2} pin is equal to the input level to the S_{IN2} pin, "L" is output from the S_{CMP2} pin. If not, "H" is output. At this time, an INT_2 interrupt request can also be generated. Select a valid edge by bit 2 of the interrupt edge selection register (address $003A_{16}$).

[Serial I/O2 Control Registers 1, 2 (SIO2CON1 / SIO2CON2)] 0015_{16}, 0016_{16}

The serial I/O2 control registers 1 and 2 are containing various selection bits for serial I/O2 control as shown in Figure 23.

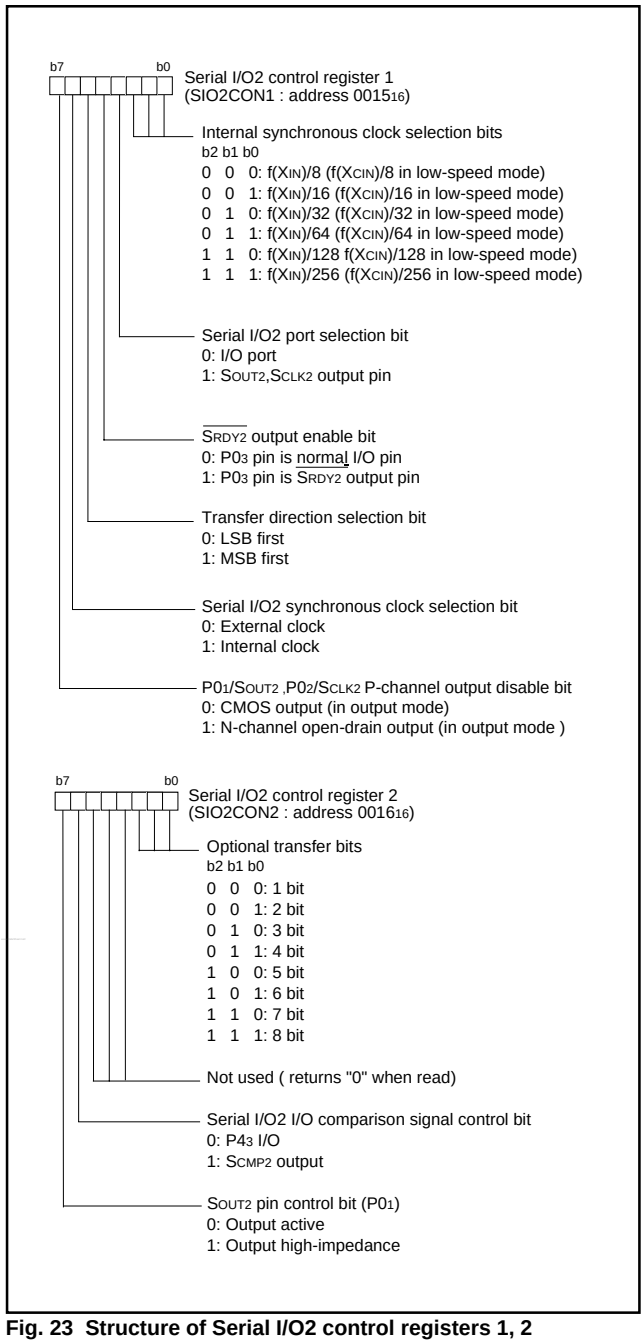
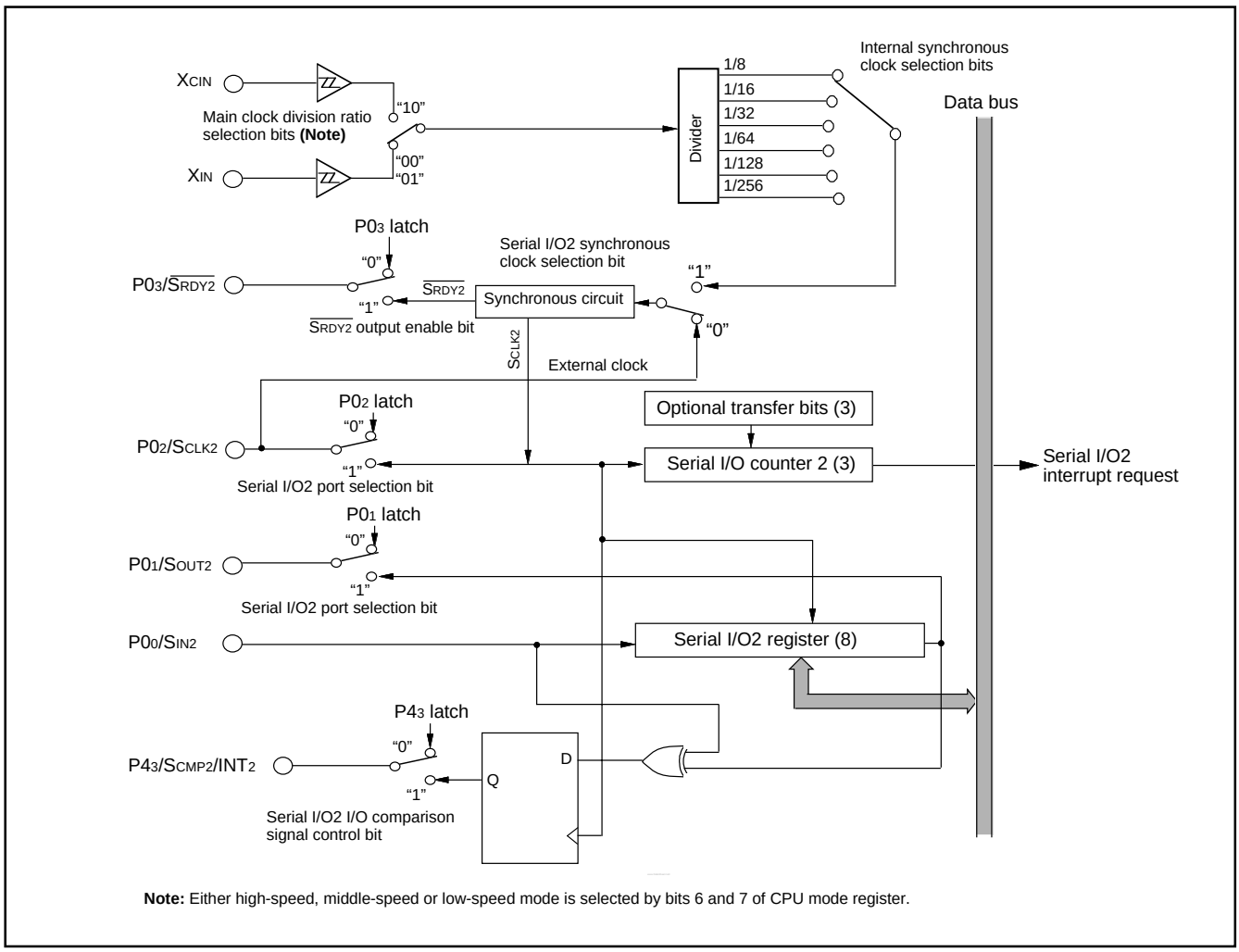
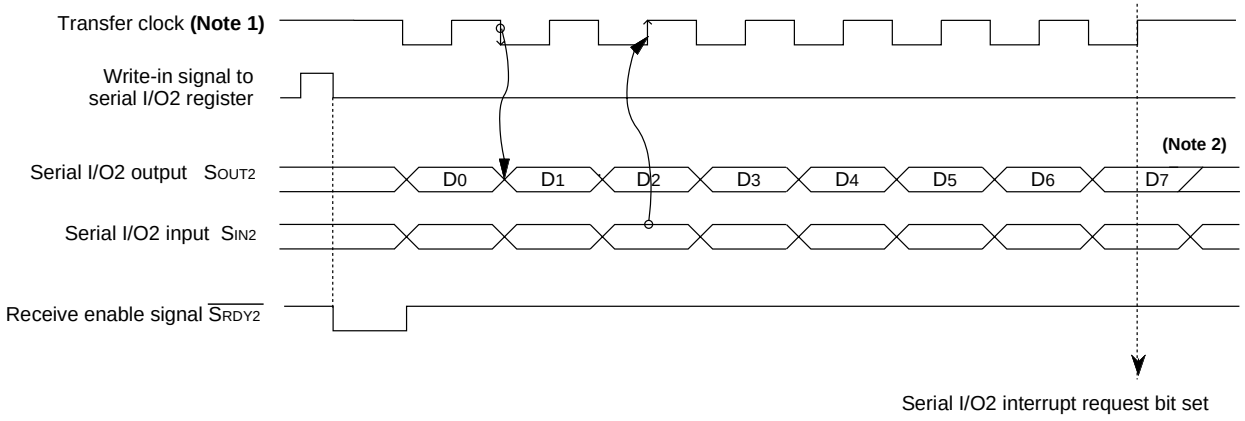


Fig. 23 Structure of Serial I/O2 control registers 1, 2



Note: Either high-speed, middle-speed or low-speed mode is selected by bits 6 and 7 of CPU mode register.

Fig. 24 Block diagram of Serial I/O2



Notes 1: When the internal clock is selected as a transfer clock, the f(XIN) clock division (f(XCIN) in low-speed mode) can be selected by setting bits 0 to 2 of serial I/O2 control register 1.
2: When the internal clock is selected as a transfer clock, the SOUT2 pin has high impedance after transfer completion.

Fig. 25 Timing chart of Serial I/O2

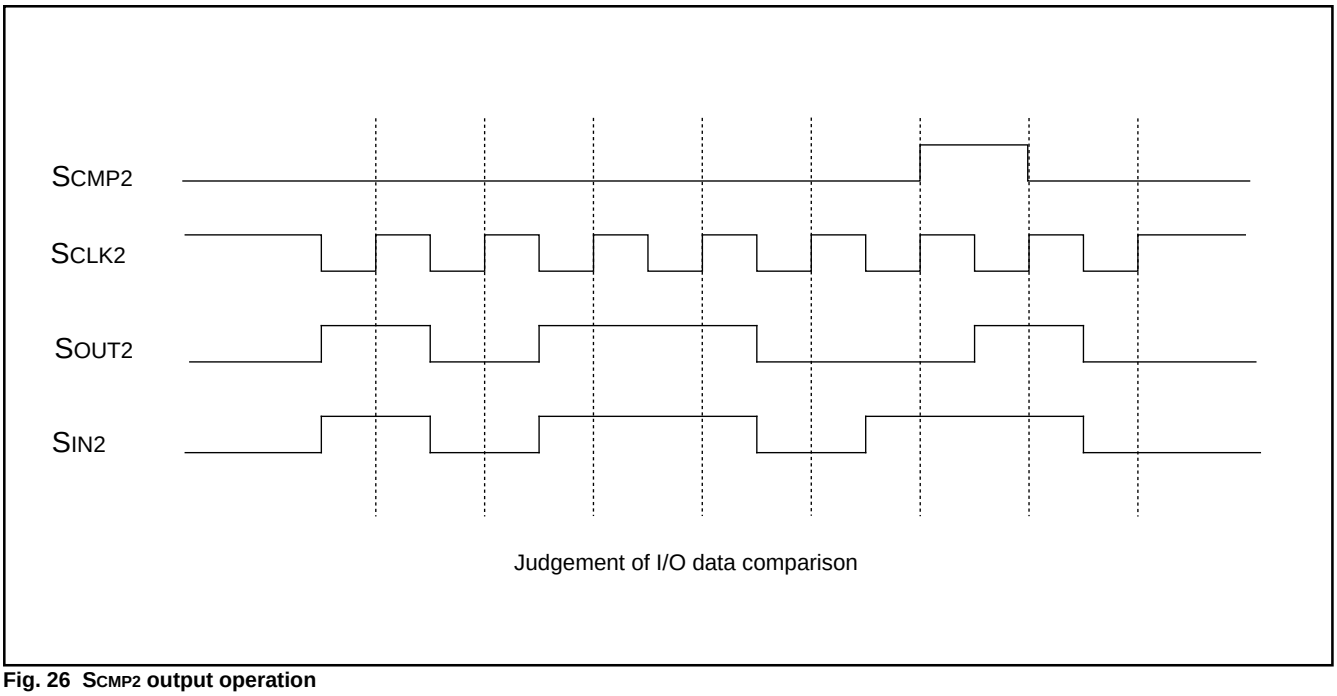


Fig. 26 SCMP2 output operation

PULSE WIDTH MODULATION (PWM)

The 3850 group (spec. H) has a PWM function with an 8-bit resolution, based on a signal that is the clock input XIN or that clock input divided by 2.

Data Setting

The PWM output pin also functions as port P44. 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,count source selection bit = "0")

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

= 0.125 × (n+1) × m μs

(when f(XIN) = 8 MHz,count source selection bit = "0")

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.

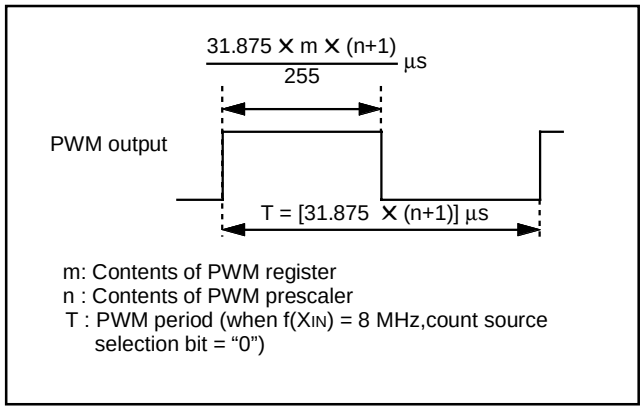


Fig. 27 Timing of PWM period

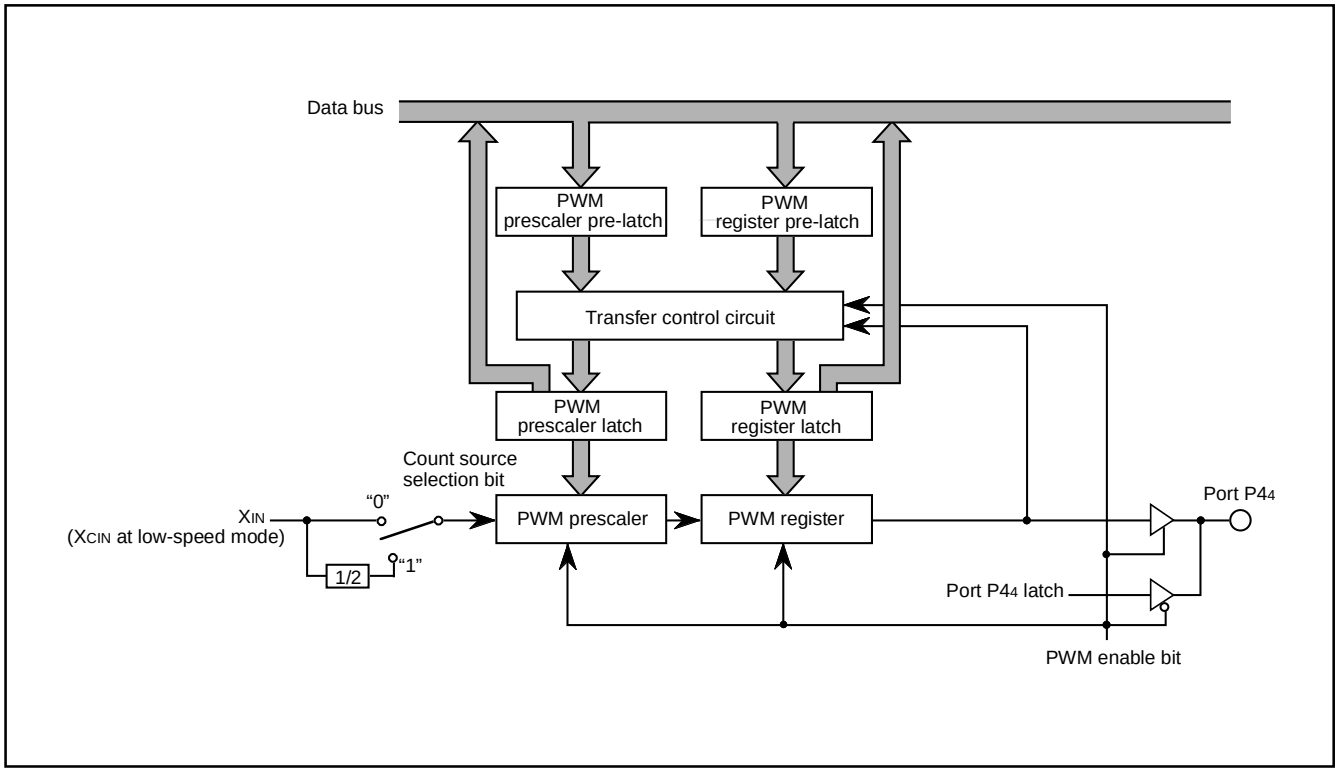


Fig. 28 Block diagram of PWM function

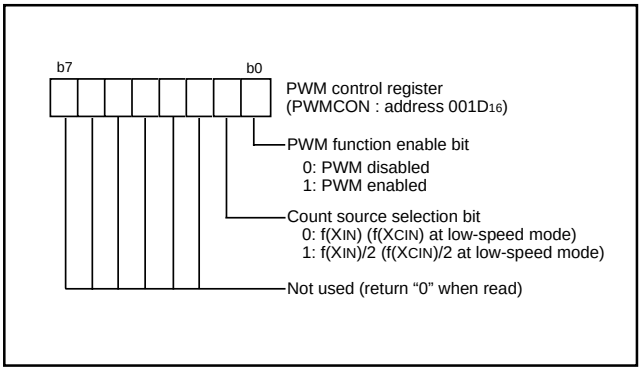


Fig. 29 Structure of PWM control register

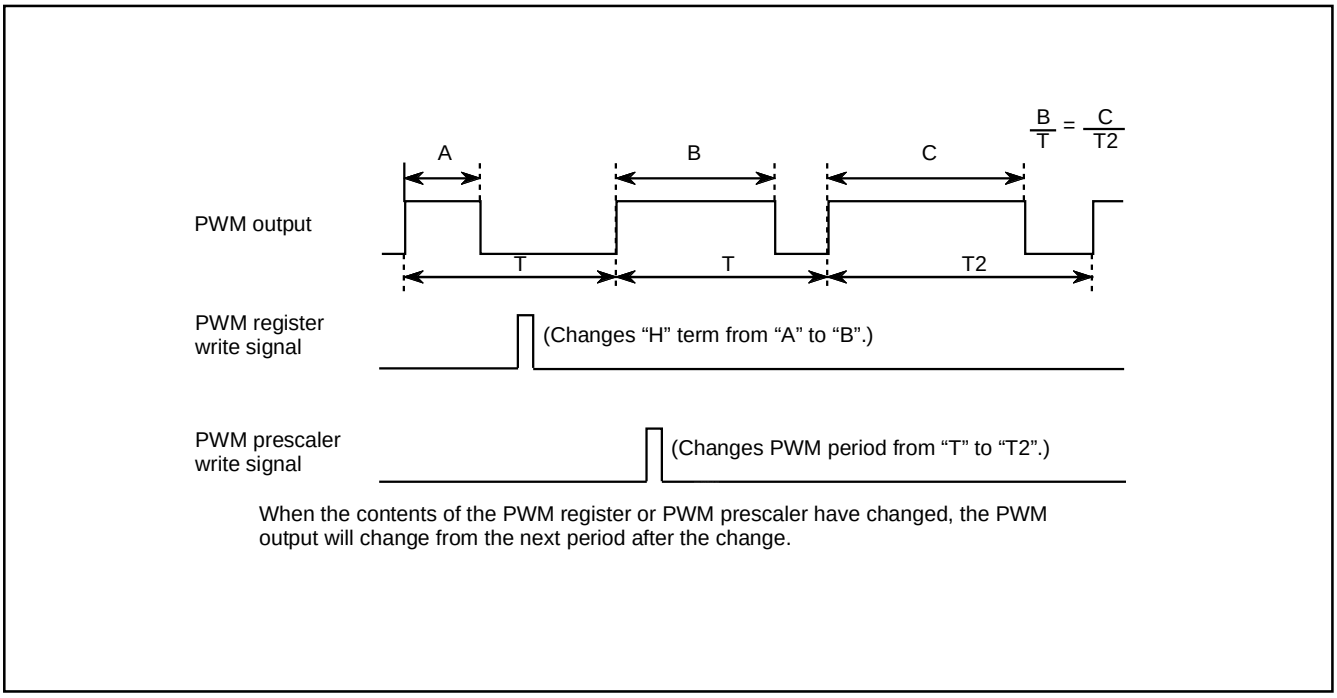


Fig. 30 PWM output timing when PWM register or PWM prescaler is changed

■Note

The PWM starts after the PWM function 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:

$\frac{n+1}{2 \cdot f(X_{IN})}$ sec (Count source selection bit = 0, where n is the value set in the prescaler)

$\frac{n+1}{f(X_{IN})}$ sec (Count source selection bit = 1, where n is the value set in the prescaler)

A-D CONVERTER
[A-D Conversion Registers (ADL, ADH)] 003516, 003616

The A-D conversion registers are read-only registers that store the result of an A-D conversion. Do not read these registers during an A-D conversion.

[AD Control Register (ADCON)] 003416

The AD control register controls the A-D conversion process. Bits 0 to 2 select a specific analog input pin. Bit 4 indicates 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 AVSS and VREF into 1024 and outputs the divided voltages.

Channel Selector

The channel selector selects one of ports P30/AN0 to P34/AN4 and inputs the voltage to the comparator.

Comparator and Control Circuit

The comparator and control circuit compare an analog input voltage with the comparison voltage, and the result is stored in the A-D conversion registers. When an A-D conversion is completed, the control circuit sets the A-D conversion completion bit and the A-D interrupt request bit to "1".

Note that because the comparator consists of a capacitor coupling, set f(XIN) to 500 kHz or more during an A-D conversion.

When the A-D converter is operated at low-speed mode, f(XIN) and f(XCIN) do not have the lower limit of frequency, because of the A-D converter has a built-in self-oscillation circuit.

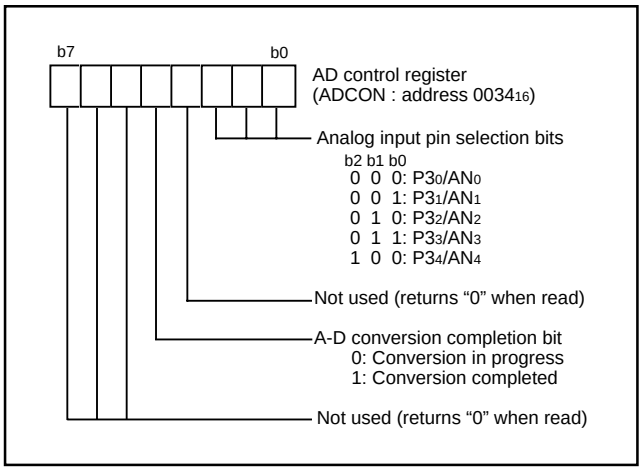


Fig. 31 Structure of AD control register

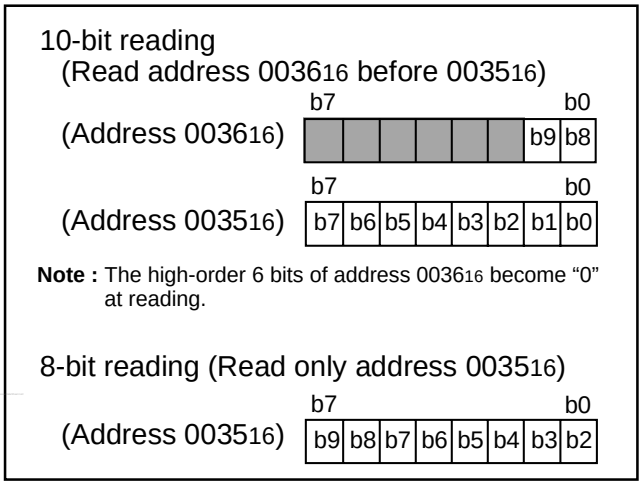


Fig. 32 Structure of A-D conversion registers

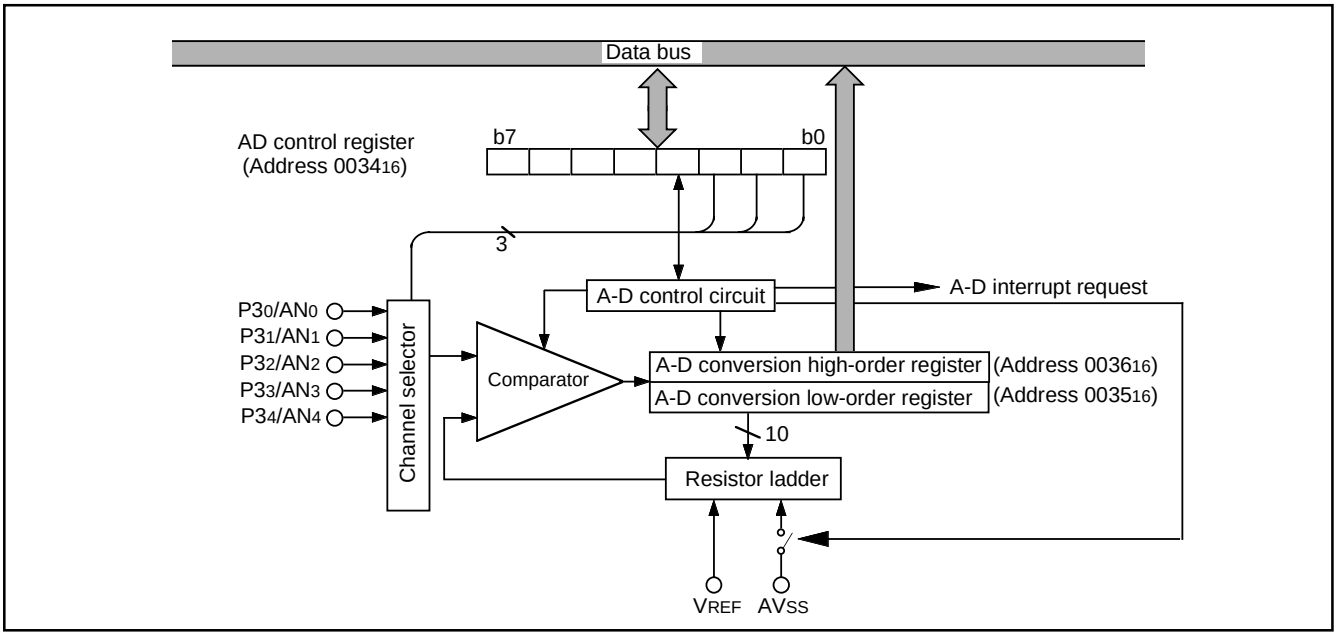


Fig. 33 Block diagram of A-D converter

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.

Standard Operation of Watchdog Timer

When any data is not written into the watchdog timer control register (address 0039_{16}) after reset, the watchdog timer is in the stop state. The watchdog timer starts to count down by writing an optional value into the watchdog timer control register (address 0039_{16}) and an internal reset occurs at an underflow of the watchdog timer H.

Accordingly, programming is usually performed so that writing to the watchdog timer control register (address 0039_{16}) may be started before an underflow. When the watchdog timer control register (address 0039_{16}) is read, the values of the high-order 6 bits of the watchdog timer H, STP instruction disable bit, and watchdog timer H count source selection bit are read.

●Initial value of watchdog timer

At reset or writing to the watchdog timer control register (address 0039_{16}), each watchdog timer H and L is set to "FF_{16}".

●Watchdog timer H count source selection bit operation

Bit 7 of the watchdog timer control register (address 0039_{16}) permits selecting a watchdog timer H count source. When this bit is set to "0", the count source becomes the underflow signal of watchdog timer L. The detection time is set to 131.072 ms at f(X_{IN}) = 8 MHz frequency and 32.768 s at f(X_{CIN}) = 32 kHz frequency.

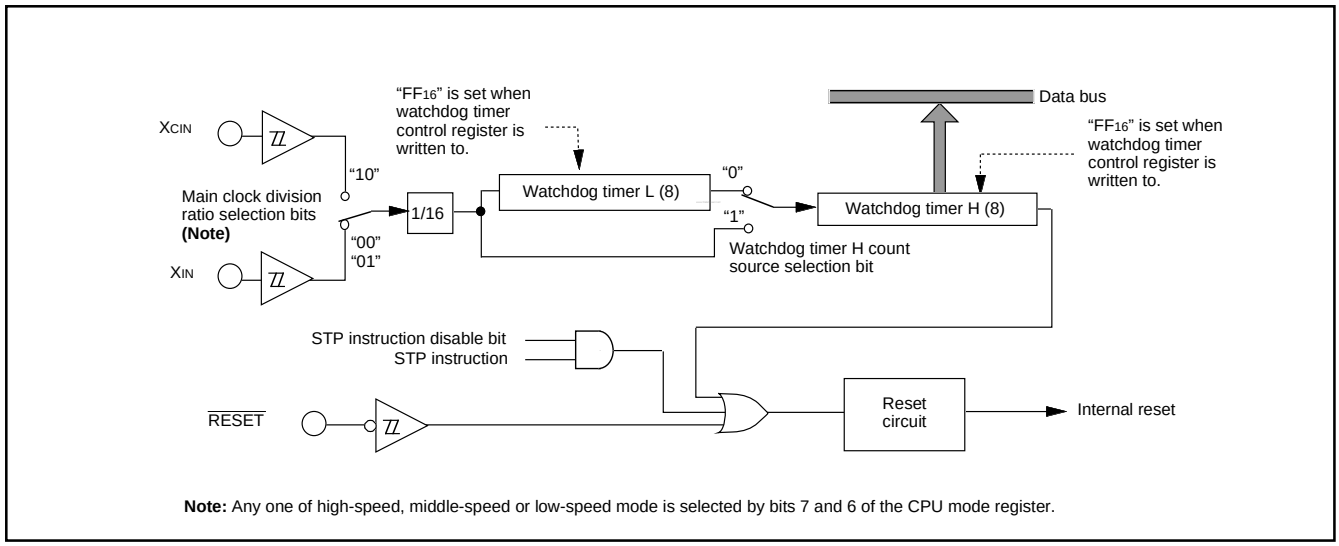
When this bit is set to "1", the count source becomes the signal divided by 16 for f(X_{IN}) (or f(X_{CIN})). The detection time in this case is set to 512 µs at f(X_{IN}) = 8 MHz frequency and 128 ms at f(X_{CIN}) = 32 kHz frequency. This bit is cleared to "0" after reset.

●Operation of STP instruction disable bit

Bit 6 of the watchdog timer control register (address 0039_{16}) permits disabling the STP instruction when the watchdog timer is in operation.

When this bit is "0", the STP instruction is enabled.

When this bit is "1", the STP instruction is disabled, once the STP instruction is executed, an internal reset occurs. When this bit is set to "1", it cannot be rewritten to "0" by program. This bit is cleared to "0" after reset.



Note: Any one of high-speed, middle-speed or low-speed mode is selected by bits 7 and 6 of the CPU mode register.

Fig. 34 Block diagram of Watchdog timer

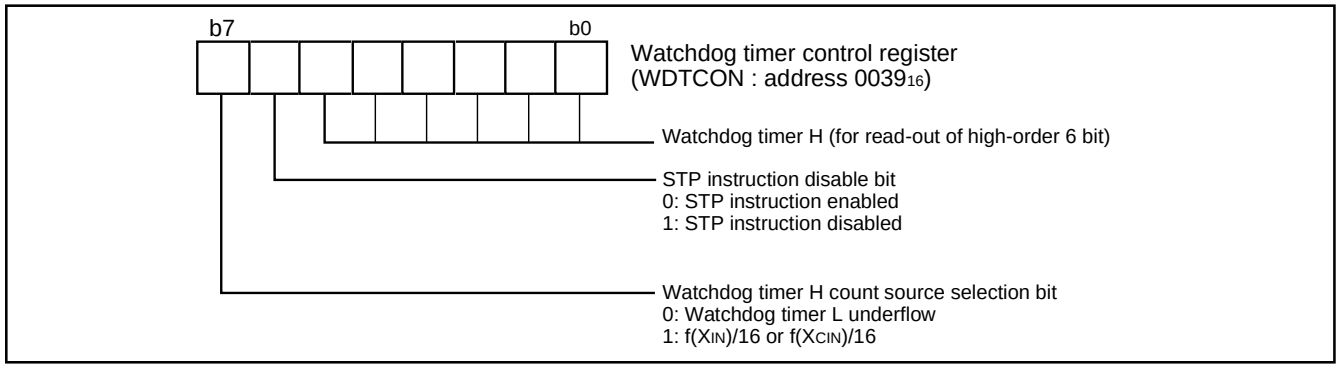


Fig. 35 Structure of Watchdog timer control register

RESET CIRCUIT

To reset the microcomputer, $\overline{\text{RESET}}$ pin must be held at an "L" level for 20 cycles or more of X_{IN}. Then the $\overline{\text{RESET}}$ pin is returned to an "H" level (the power source voltage must be between 2.7 V and 5.5 V, and the oscillation must be stable), reset is released. After the reset is completed, the program starts from the address contained in address $FFFD_{16}$ (high-order byte) and address $FFFC_{16}$ (low-order byte). Make sure that the reset input voltage is less than 0.54 V for V_{CC} of 2.7 V.

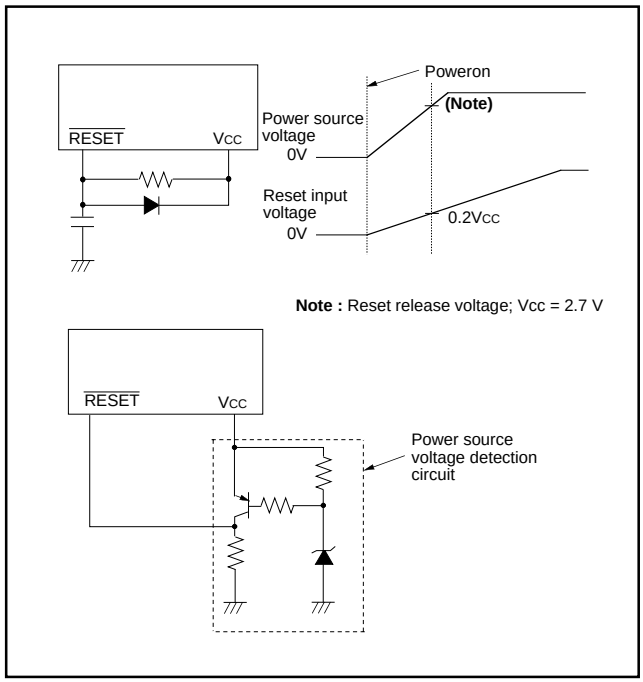


Fig. 36 Reset circuit example

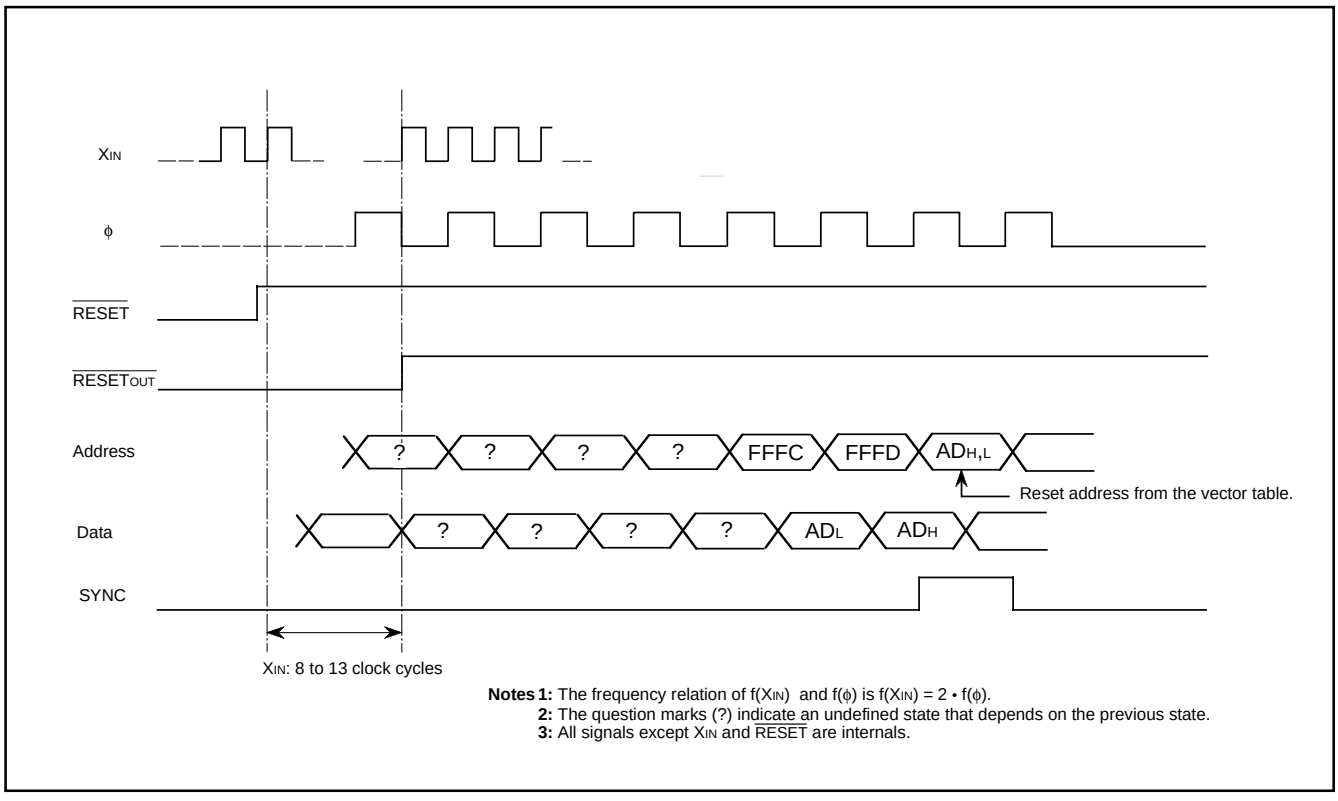


Fig. 37 Reset sequence

		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)	Serial I/O2 control register 1 (SIO2CON1)	0015_{16}	00_{16}
(12)	Serial I/O2 control register 2 (SIO2CON2)	0016_{16}	0 0 0 0 0 1 1 1
(13)	Serial I/O2 register (SIO2)	0017_{16}	X X X X X X X X
(14)	Transmit/Receive buffer register (TB/RB)	0018_{16}	X X X X X X X X
(15)	Serial I/O1 status register (SIOSTS)	0019_{16}	1 0 0 0 0 0 0 0
(16)	Serial I/O1 control register (SIOCON)	$001A_{16}$	00_{16}
(17)	UART control register (UARTCON)	$001B_{16}$	1 1 1 0 0 0 0 0
(18)	Baud rate generator (BRG)	$001C_{16}$	X X X X X X X X
(19)	PWM control register (PWMCON)	$001D_{16}$	00_{16}
(20)	PWM prescaler (PREPWM)	$001E_{16}$	X X X X X X X X
(21)	PWM register (PWM)	$001F_{16}$	X X X X X X X X
(22)	Prescaler 12 (PRE12)	0020_{16}	FF_{16}
(23)	Timer 1 (T1)	0021_{16}	01_{16}
(24)	Timer 2 (T2)	0022_{16}	00_{16}
(25)	Timer XY mode register (TM)	0023_{16}	00_{16}
(26)	Prescaler X (PREX)	0024_{16}	FF_{16}
(27)	Timer X (TX)	0025_{16}	FF_{16}
(28)	Prescaler Y (PREY)	0026_{16}	FF_{16}
(29)	Timer Y (TY)	0027_{16}	FF_{16}
(30)	Timer count source selection register (TCSS)	0028_{16}	00_{16}
(31)	A-D control register (ADCON)	0034_{16}	0 0 0 1 0 0 0 0
(32)	A-D conversion low-order register (ADL)	0035_{16}	X X X X X X X X
(33)	A-D conversion high-order register (ADH)	0036_{16}	0 0 0 0 0 0 X X

		Address	Register contents
(34)	MISRG	0038_{16}	00_{16}
(35)	Watchdog timer control register (WDTCON)	0039_{16}	0 0 1 1 1 1 1 1
(36)	Interrupt edge selection register (INTEDGE)	$003A_{16}$	00_{16}
(37)	CPU mode register (CPUM)	$003B_{16}$	0 1 0 0 1 0 0 0
(38)	Interrupt request register 1 (IREQ1)	$003C_{16}$	00_{16}
(39)	Interrupt request register 2 (IREQ2)	$003D_{16}$	00_{16}
(40)	Interrupt control register 1 (ICON1)	$003E_{16}$	00_{16}
(41)	Interrupt control register 2 (ICON2)	$003F_{16}$	00_{16}
(42)	Processor status register	(PS)	X X X X X 1 X X
(43)	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. 38 Internal status at reset

CLOCK GENERATING CIRCUIT

The 3850 group (spec. H) has two built-in oscillation circuits. 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. 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.

■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 the stop mode. When switching the mode between middle/high-speed and low-speed, set the frequency on condition that f(XIN) > 3•f(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.
The sub-clock XCIN-XCOUT oscillating circuit can not directly input clocks that are generated externally. Accordingly, make sure to cause an external resonator to oscillate.

Oscillation Control
(1) Stop mode

If the STP instruction is executed, the internal clock ϕ stops at an "H" level, and XIN and XCIN oscillation stops. 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.
Either XIN or XCIN divided by 16 is input to the prescaler 12 as count source. 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. This ensures time for the clock oscillation using the ceramic resonators to be stabilized. 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.

(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 at reset or when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

To ensure that the interrupts will be received to release the STP or WIT state, their interrupt enable bits must be set to "1" before executing of the STP or WIT instruction.
When releasing the STP state, the prescaler 12 and timer 1 will start counting the clock XIN divided by 16. Accordingly, set the timer 1 interrupt enable bit to "0" before executing the STP instruction.

■Note

When using the oscillation stabilizing time set after STP instruction released bit set to "1", evaluate time to stabilize oscillation of the used oscillator and set the value to the timer 1 and prescaler 12.

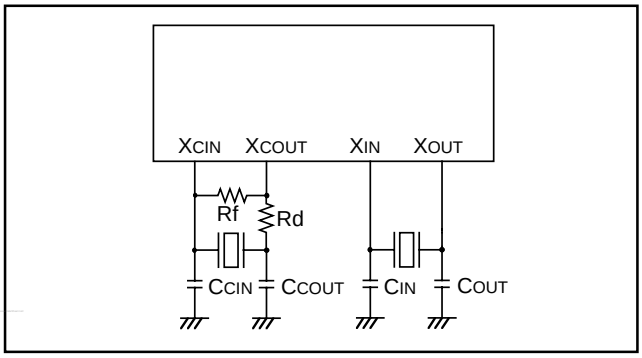


Fig. 39 Ceramic resonator circuit

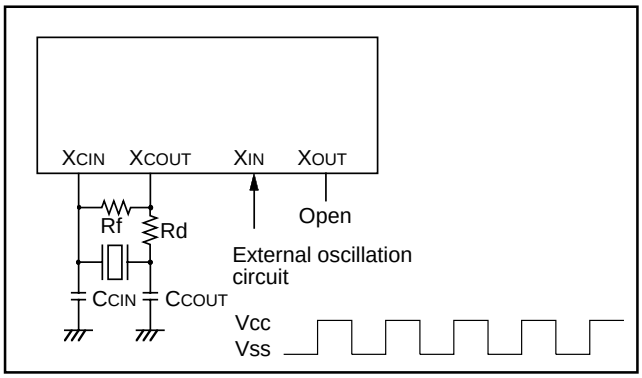


Fig. 40 External clock input circuit

[MISRG (MISRG)] 0038$_{16}$

MISRG consists of three control bits (bits 1 to 3) for middle-speed mode automatic switch and one control bit (bit 0) for oscillation stabilizing time set after STP instruction released.

By setting the middle-speed mode automatic switch start bit to "1" while operating in the low-speed mode and setting the middle-speed mode automatic switch set bit to "1", X_{IN} oscillation automatically starts and the mode is automatically switched to the middle-speed mode.

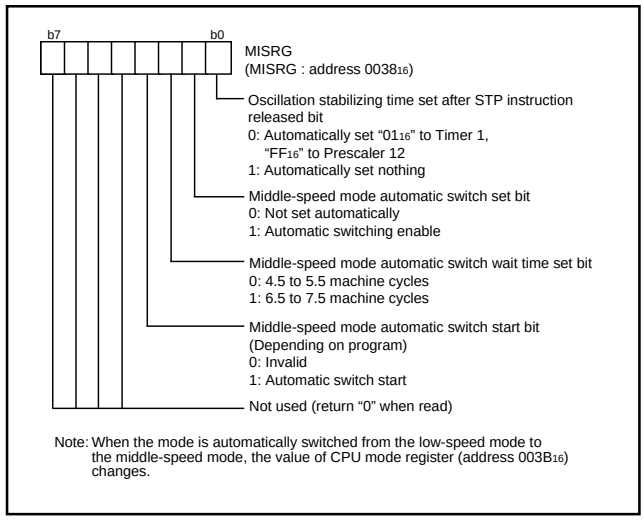
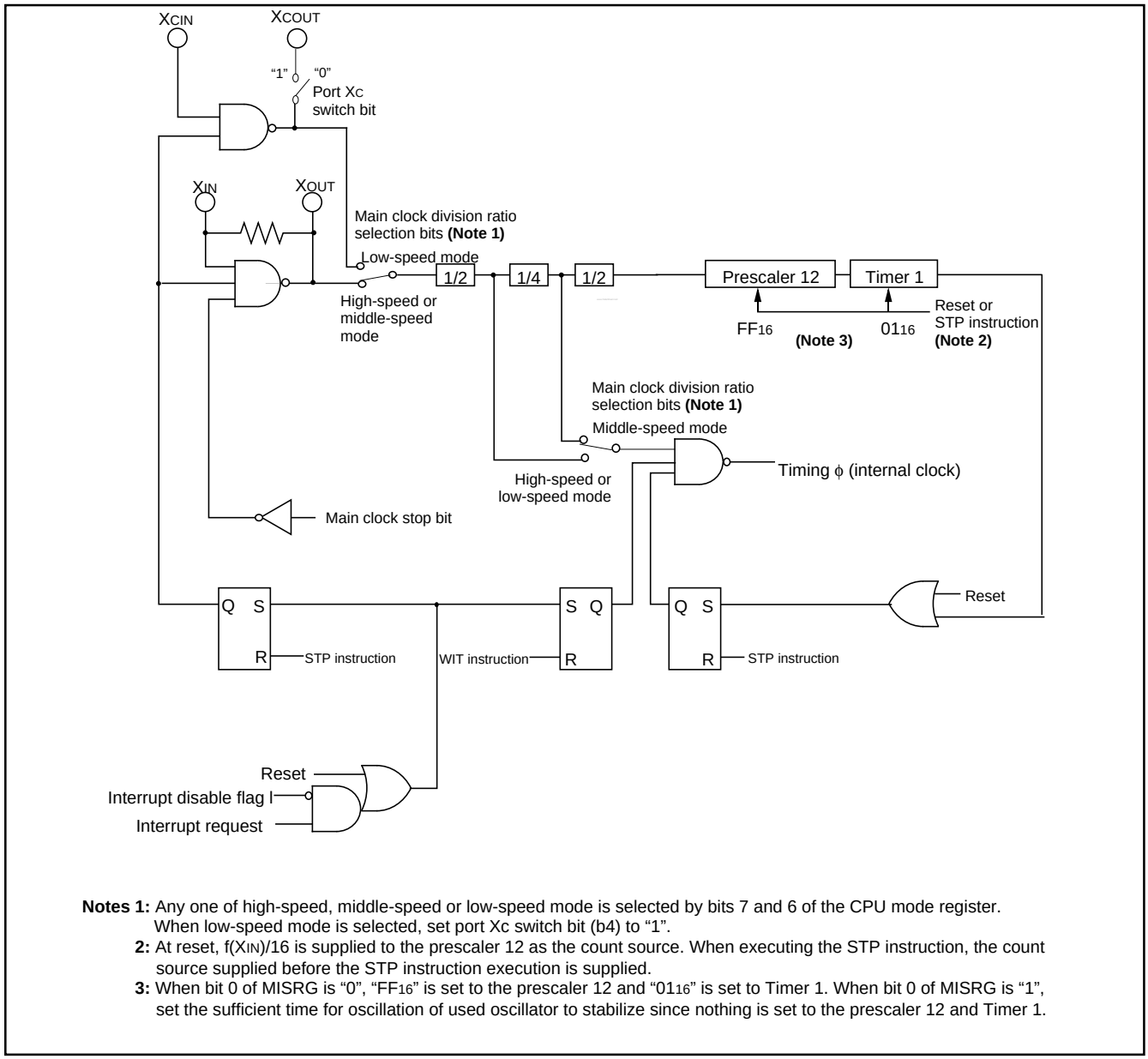


Fig. 41 Structure of MISRG

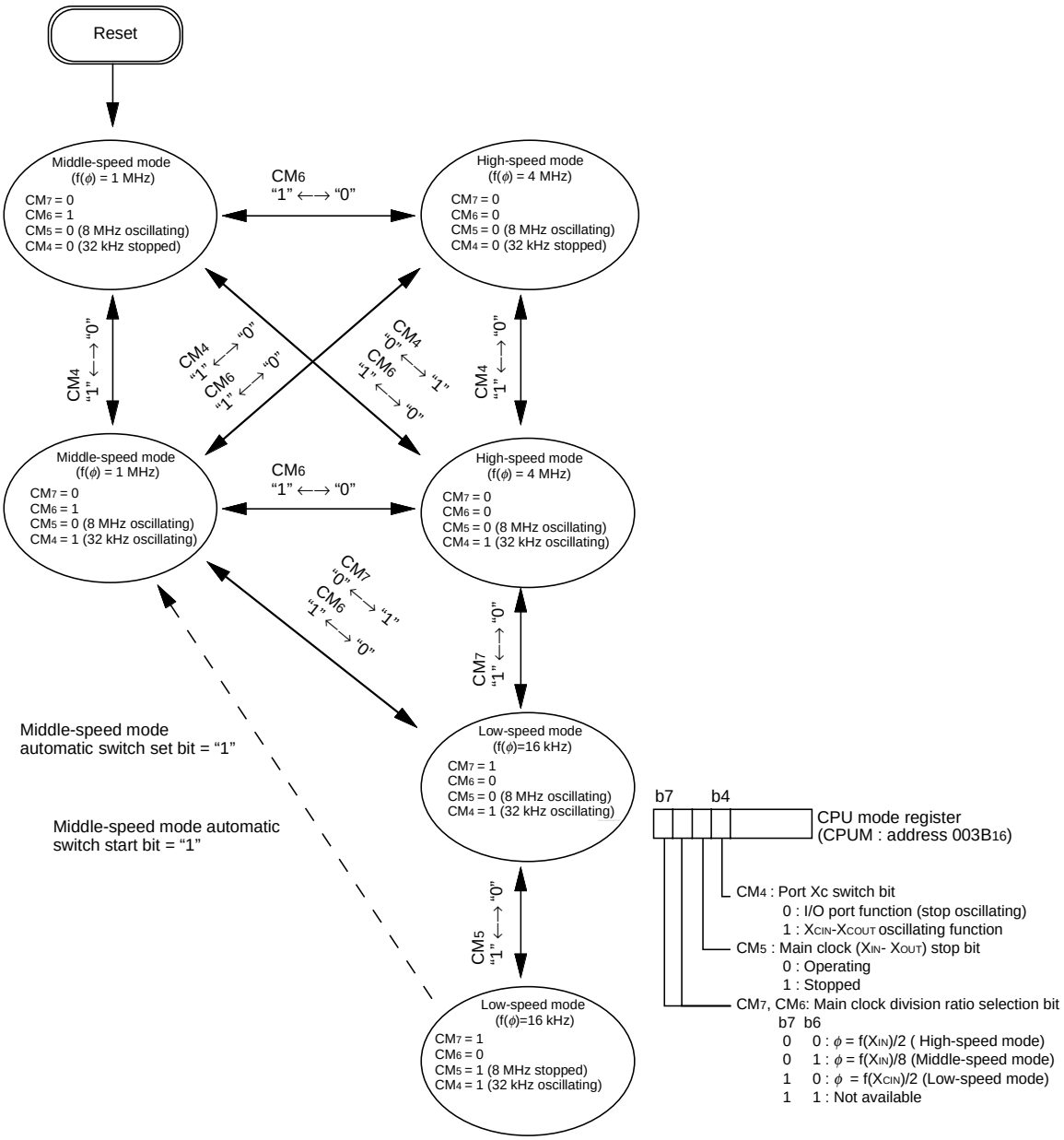


Notes 1: Any one of high-speed, middle-speed or low-speed mode is selected by bits 7 and 6 of the CPU mode register. When low-speed mode is selected, set port Xc switch bit (b4) to "1".

2: At reset, f(X_{IN})/16 is supplied to the prescaler 12 as the count source. When executing the STP instruction, the count source supplied before the STP instruction execution is supplied.

3: When bit 0 of MISRG is "0", "FF$_{16}$" is set to the prescaler 12 and "01$_{16}$" is set to Timer 1. When bit 0 of MISRG is "1", set the sufficient time for oscillation of used oscillator to stabilize since nothing is set to the prescaler 12 and Timer 1.

Fig. 42 System clock generating circuit block diagram (Single-chip mode)



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 bit 0 of MISRG is "0" and the stop mode is ended, a delay of approximately 1 ms occurs by connecting timer 1 in middle/high-speed mode.

5 : When bit 0 of MISRG is "0" and the stop mode is ended, the following is performed.
(1) After the clock is restarted, a delay of approximately 256 ms occurs in low-speed mode if Timer 12 count source selection bit is "0".
(2) After the clock is restarted, a delay of approximately 16 ms occurs in low-speed mode if Timer 12 count source selection bit is "1".

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: When switching to the middle-speed mode by the middle-speed mode automatic switch bit of MISRG, the waiting time set by the middle-speed mode automatic switch wait time set bit is generated automatically, and switch to the middle-speed mode.

8 : 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. 43 State transitions of system clock

FLASH MEMORY VERSION

Summary

Table 9 shows the summary of the M38507F8 (flash memory version).

Table 9 Summary of M38507F8 (flash memory version)

Item		Specification
Power source voltage		Vcc = 2.7–5.5 V **(Note 1)** Vcc = 2.7–3.6 V **(Note 2)**
Program/Erase V_{PP} voltage		4.5–5.5 V, f(X_{IN}) = 8 MHz
Flash memory mode		3 modes (Parallel I/O mode, Standard serial I/O mode, CPU rewrite mode)
Erase block division	User ROM area	1 block (32 Kbytes)
	Boot ROM area	1 block (4 Kbytes) **(Note 3)**
Program method		Byte program
Erase method		Batch erasing
Program/Erase control method		Program/Erase control by software command
Number of commands		6 commands
Number of program/Erase times		100 times
ROM code protection		Available in parallel I/O mode, and standard serial I/O mode

Notes 1: The power source voltage must be Vcc = 4.5–5.5 V at program and erase operation.
2: The power source voltage can be Vcc = 3.0–3.6 V also at program and erase operation.
3: 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 rewritten in only parallel I/O mode.

Flash Memory Mode

The M38507F8 (flash memory version) has an internal new DINOR (DIvided bit line NOR) flash memory that can be rewritten with a single power source when V_{CC} is 5 V, and 2 power sources when V_{CC} is 3.3-5.0 V.

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

The flash memory of the M38507F8 is divided into User ROM area and Boot ROM area as shown in Figure 44.

In addition to the ordinary user ROM area to store a microcomputer 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.

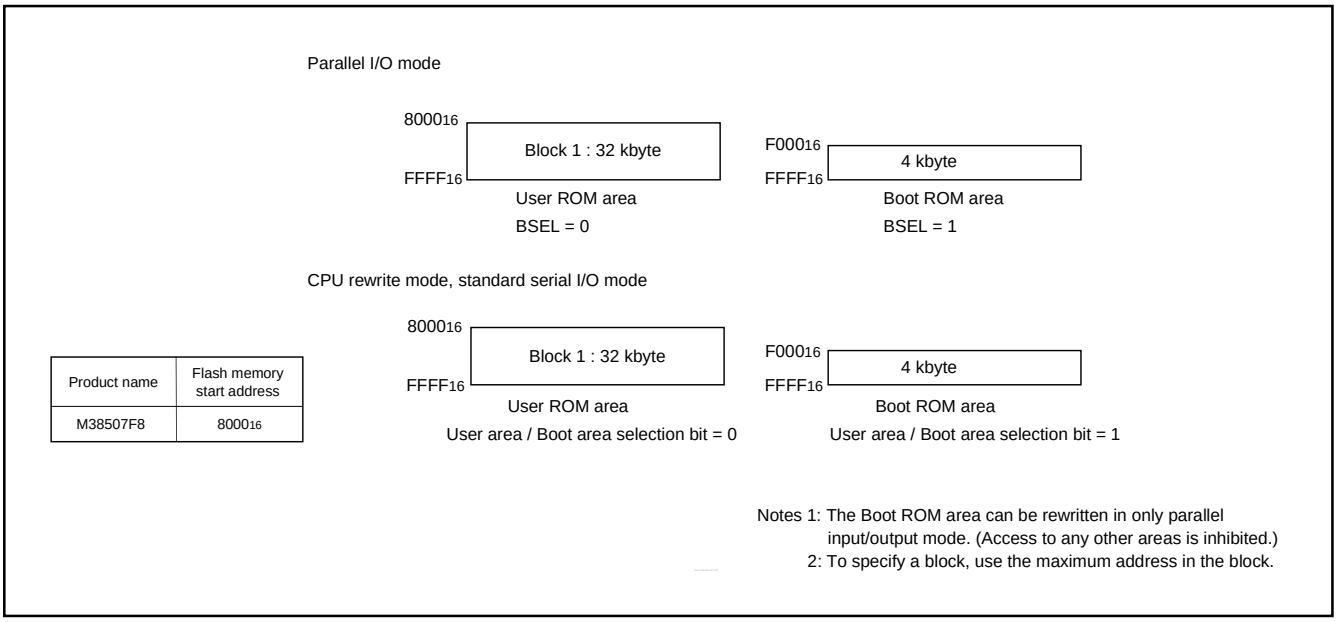


Fig. 44 Block diagram of flash memory version

CPU Rewrite Mode

In CPU rewrite mode, the on-chip 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 44 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.

Microcomputer Mode and 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 44 for details about the Boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNV_{SS} pin low. In this case, the CPU starts operating using the control program in the user ROM area.

When the microcomputer is reset by pulling the $P4_1/INT_0$ pin high, the CNV_{SS} pin high, the CPU starts operating using the control program in the Boot ROM area (program start address is $FFFC_{16}$, $FFFD_{16}$ fixation). This mode is called the "boot" mode.

Block Address

Block addresses refer to the maximum address of each block. These addresses are used in the block erase command. In case of the M38507F8, it has only one block.

Outline Performance (CPU Rewrite Mode)

In the 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 before it can be executed.

The CPU rewrite mode is accessed by applying 5V ± 10% to the CNV_{SS} pin and writing "1" for the CPU rewrite mode select bit (bit 1 in address $0FFE_{16}$). Software commands are accepted once the mode is accessed.

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 45 shows the flash memory control register.

Bit 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". Otherwise, it is "1".

Bit 1 is the CPU rewrite mode select bit. When this bit is set to "1" and 5V ± 10% are applied to the CNV_{SS} pin, the M38507F8 accesses the CPU rewrite mode. Software commands are accepted once the mode is accessed. In CPU rewrite mode, the CPU becomes unable to access the internal flash memory directly. Therefore, use the control program in RAM for write to bit 1. To set this bit to "1", it is necessary to write "0" and then write "1" in succession. The bit can be set to "0" by only writing a "0".

Bit 2 is the CPU rewrite mode entry flag. This bit can be read to check whether the CPU rewrite mode has been entered or not.

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

Bit 4 is the User area/Boot area selection bit. When this bit is set to "1", Boot ROM area is accessed, and CPU rewrite mode in Boot ROM area is available. In boot mode, this bit is set "1" automatically. Operation of this bit must be in RAM area.

Figure 46 shows a flowchart for setting/releasing the CPU rewrite mode.

Precautions on CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

(1) Operation speed

During CPU rewrite mode, set the internal clock frequency 4MHz or less using the main clock division ratio selection bits (bit 6, 7 at $003B_{16}$).

(2) Instructions inhibited against use

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

(3) Interrupts inhibited against use

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

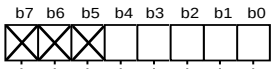
(4) Watchdog timer

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.

Flash memory control register

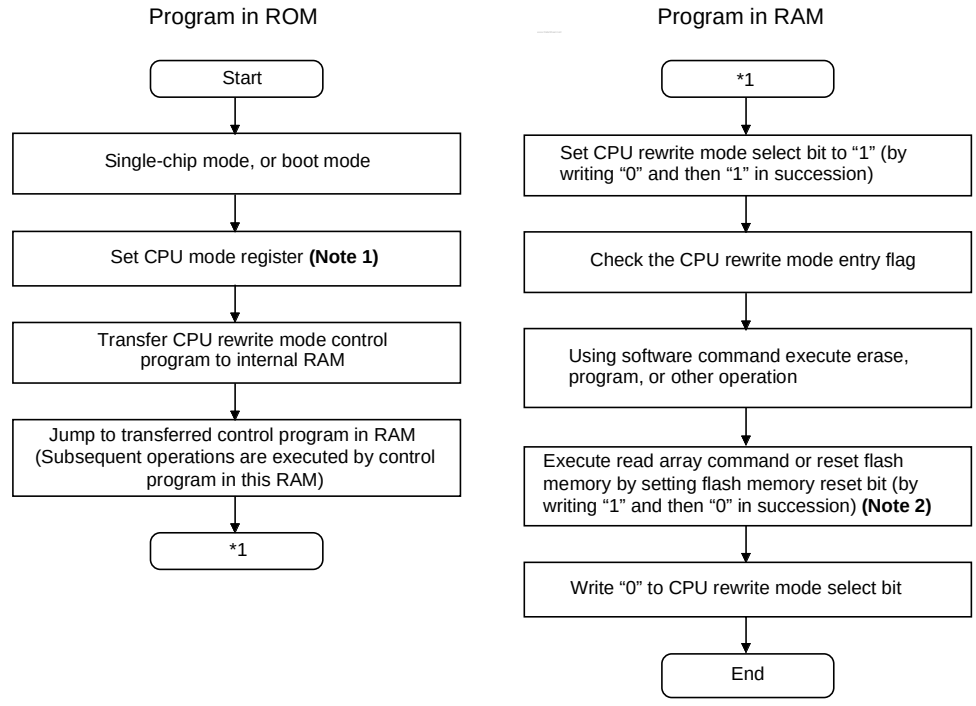


Symbol	Address	When reset
FMCR	0FFE16	XXX00001

Bit symbol	Bit name	Function	R	W
FMCR0	RY/$\overline{BY}$ status flag	0: Busy (being written or erased) 1: Ready	O	—
FMCR1	CPU rewrite mode select bit (Note 2)	0: Normal mode (Software commands invalid) 1: CPU rewrite mode (Software commands acceptable)	O	O
FMCR2	CPU rewrite mode entry flag	0: Normal mode (Software commands invalid) 1: CPU rewrite mode (Software commands acceptable)	O	—
FMCR3	Flash memory reset bit (Note 3)	0: Normal operation 1: Reset	O	O
FMCR4	User area / Boot area selection bit	0: User ROM area 1: Boot ROM area	O	O
Nothing is assigned. When write, set “0”. When read, values are indeterminate.			—	—

Notes 1: The contents of the flash memory control register after reset is released become “XXX00001”.
2: For this bit to be set to “1”, write “0” and then “1” to bit 1 in succession.
3: In order to perform flash memory reset by this bit setup, while the CPU rewriting mode selection bit is set to “1”, write “1” to bit 3. In order to reset release, write “0” to bit 3 in the next.

Fig. 45 Flash memory control registers



Notes 1: Set bit 6, 7 (Main clock division ratio selection bits) at CPU mode register (003B16).
2: Before exiting the CPU rewrite mode after completing erase or program operation, always be sure to execute a read array command or reset the flash memory.

Fig. 46 CPU rewrite mode set/reset flowchart

Software Commands

Table 10 lists the software commands.
After setting the CPU rewrite mode select bit to "1", write a software command to specify an erase or program operation.
The content of each software command is explained below.

Read Array Command (FF_{16})

The read array mode is entered by writing the command code "FF_{16}" in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the content of the specified address is read out at the data bus (D_0–D_7).
The read array mode is retained intact until another command is written. And after power on and after recover from deep power down mode, this mode is selected also.

Read Status Register Command (70_{16})

When the command code "70_{16}" is written in the first bus cycle, the content of the status register is read out at the data bus (D_0–D_7) by a read in the second bus cycle.
The status register is explained in the next section.

Clear Status Register Command (50_{16})

This command is used to clear the bits SR1,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 "50_{16}" in the first bus cycle.

Table 10 List of software commands (CPU rewrite mode)

Command	Cycle number	First bus cycle			Second bus cycle		
		Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read array	1	Write	X **(Note 4)**	FF_{16}			
Read status register	2	Write	X	70_{16}	Read	X	SRD **(Note 1)**
Clear status register	1	Write	X	50_{16}			
Program	2	Write	X	40_{16}	Write	WA **(Note 2)**	WD **(Note 2)**
Erase all block	2	Write	X	20_{16}	Write	X	20_{16}
Block erase	2	Write	X	20_{16}	Write	BA **(Note 3)**	$D0_{16}$

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

FUNCTIONAL DESCRIPTION

Program Command (40_{16})

Program operation starts when the command code "40_{16}" 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 reading the status register or the RY/$\overline{BY}$ status flag. When the program starts, the read status register mode is accessed automatically and the content of the status register is read into the data bus (D0–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 (FF_{16}) is written.

The RY/$\overline{BY}$ status flag 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.

Erase All Blocks Command (20_{16}/20_{16})

By writing the command code "20_{16}" in the first bus cycle and the confirmation command code "20_{16}" in the second bus cycle that follows, the system starts erase all blocks(erase and erase verify).

Whether the erase all blocks command is terminated can be confirmed by reading the status register or the RY/$\overline{BY}$ status flag. When the erase all blocks operation starts, the read status register mode is accessed automatically and the content of the status register can be read out. The status register bit 7 (SR7) is set to "0" at the same time the erase operation starts and is returned to "1" upon completion of the 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 erase operation and "1" when the erase operation is completed as is the status register bit 7.

At erase all blocks end, erase results can be checked by reading the status register. For details, refer to the section where the status register is detailed.

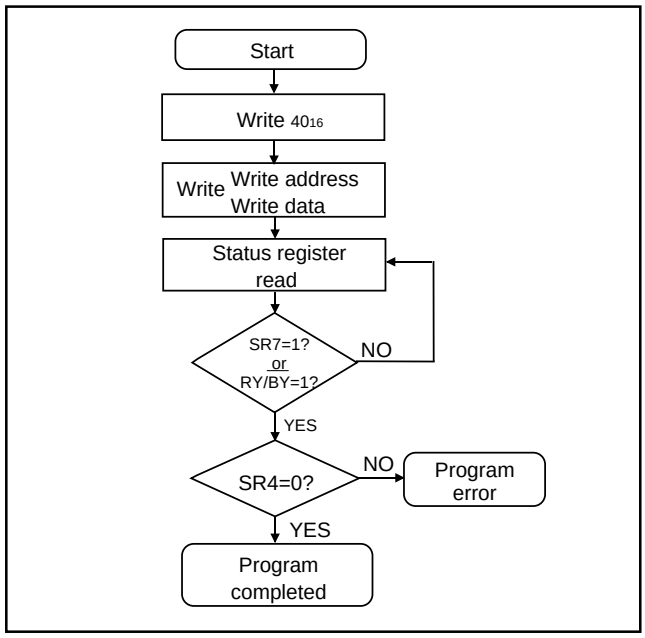


Fig. 47 Program flowchart

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

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

Whether the block erase operation is completed can be confirmed by reading the status register or the RY/$\overline{BY}$ status flag. At the same time the block erase operation starts, the read status register mode is automatically entered, so the content 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 operation is completed, the status register can be read out to know the result of the block erase operation. For de-

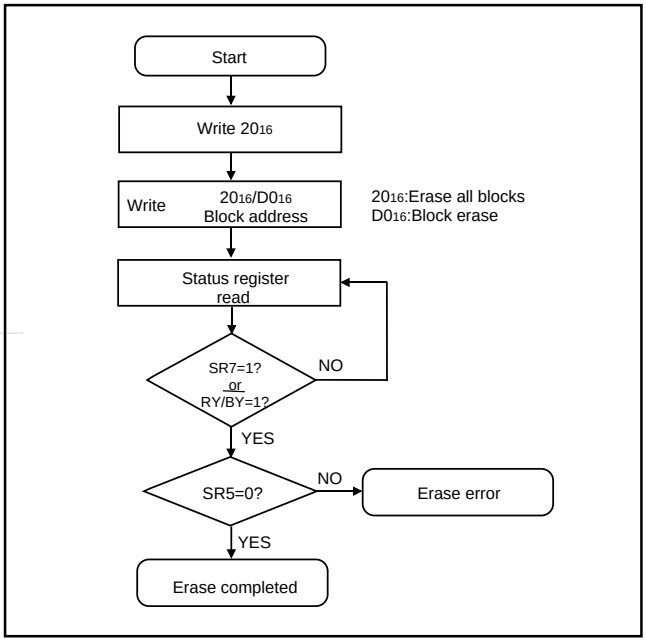


Fig. 48 Erase flowchart

Status Register

The status register shows the operating state 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

Table 11 shows the status register.
Also, the status register can be cleared in the following way.
(1) By writing the clear status register command (50_{16})
(2) In the deep power down mode
(3) In the power supply off state
After a reset, the status register is set to "80_{16}".
Each bit in this register is explained below.

Sequencer status (SR7)

After power-on, and after recover from deep power down mode, the sequencer status is set to "1"(ready).
The sequencer status indicates the operating status of the device. This status bit is set to "0" (busy) during write or erase operation and is set to "1" upon completion of these operations.

Erase status (SR5)

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

Program status (SR4)

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

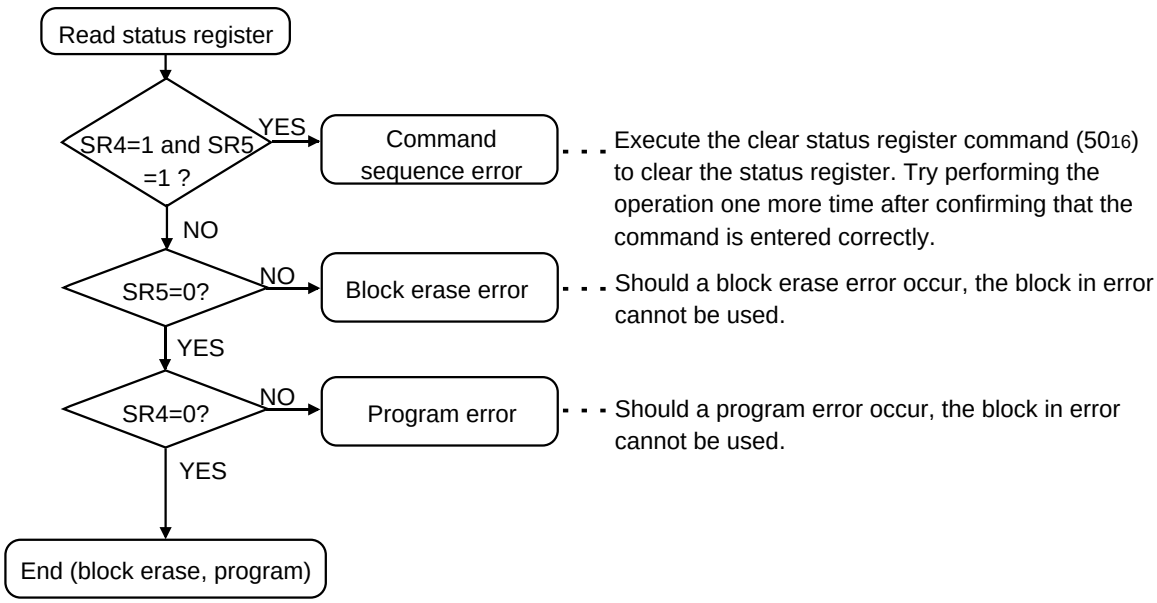
If "1" is written for any of the SR5 or SR4 bits, the program, erase all blocks, and block erase commands are not accepted. Before executing these commands, execute the clear status register command (50_{16}) and clear the status register.
Also, any commands are not correct, both SR5 and SR4 are set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

Table 11 Definition of each bit in status register

Each bit of SRD0 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 49 shows a full status check flowchart and the action to be taken when each error occurs.



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

Fig. 49 Full status check flowchart and remedial procedure for errors

Functions To Inhibit Rewriting Flash Memory Version

To prevent the contents of the flash memory version from being read out or rewritten easily, the device 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.

ROM code protect function

The ROM code protect function is the function inhibit reading out or modifying the contents of the flash memory version by using the ROM code protect control address ($FFDB_{16}$) during parallel I/O mode. Figure 50 shows the ROM code protect control address ($FFDB_{16}$). (This address exists in the user ROM area.)

If one of the pair of ROM code protect bits is set to "0", ROM code protect is turned on, so that the contents of the flash memory version are protected against readout and modification. 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", ROM code protect is turned off, so that the contents of the flash memory version can be read out or modified. Once 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 some other mode to rewrite the contents of the ROM code protect reset bits.

ROM code protect control address

b7	b6	b5	b4	b3	b2	b1	b0
						1	1

Symbol: ROMCP — Address: $FFDB_{16}$ — When reset: FF_{16}

Bit symbol	Bit name	Function
Reserved bit		Always set this bit to "1"
ROMCP2	ROM code protect level 2 set bit **(Note 1, 2)**	b3 b2 00: Protect enabled 01: Protect enabled 10: Protect enabled 11: Protect disabled
ROMCR	ROM code protect reset bit **(Note 3)**	b5 b4 00: Protect removed 01: Protect set bit effective 10: Protect set bit effective 11: Protect set bit effective
ROMCP1	ROM code protect level 1 set bit **(Note 1)**	b7 b6 00: Protect enabled 01: Protect enabled 10: Protect enabled 11: Protect disabled

Notes 1: When ROM code protect is turned on, the on-chip flash memory is protected against readout or modification in parallel input/output 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 changed in parallel input/output mode, they need to be rewritten in serial input/output mode or some other mode.

Fig. 50 ROM code protect control address

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 peripheral unit 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 peripheral unit are not accepted. The ID code consists of 8-bit data, the areas of which are $FFD4_{16}$ to $FFDA_{16}$. Write a program which has had the ID code preset at these addresses to the flash memory.

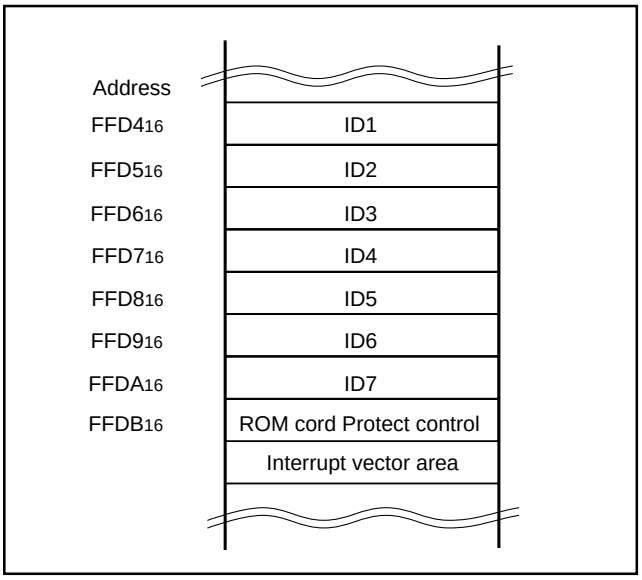


Fig. 51 ID code store addresses

Parallel I/O Mode

The parallel I/O mode is entered by making connections shown in Figure 52 and then turning the Vcc power supply on.

Address

The user ROM is only one block as shown in Figure 44. The block address referred to in this data sheet is the maximum address value of each block.

User ROM and Boot ROM Areas

In parallel I/O mode, the user ROM and boot ROM areas shown in Figure 44 can be rewritten. The BSEL pin is used to choose between these two areas. The user ROM area is selected by pulling the BSEL input low; the boot ROM area is selected by driving the BSEL input high. Both areas of flash memory can be operated on in the same way.

Program and block erase operations can be performed in the user ROM area. The user ROM area and its block is shown in Figure 44.

The user ROM area is 32 Kbytes in size. In parallel I/O mode, it is located at addresses 8000_{16} through $FFFF_{16}$. The boot ROM area is 4 Kbytes in size. In parallel I/O mode, it is 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 Renesas factory. Therefore, using the device in standard serial input/output mode, you do not need to write to the boot ROM area.

Functional Outline (Parallel I/O Mode)

In parallel I/O mode, bus operation modes—Read, Output Disable, Standby, Write, and Deep Power Down—are selected by the status of the $\overline{CE}$, $\overline{OE}$, $\overline{WE}$, and $\overline{RP}$ input pins.

The contents of erase, program, and other operations are selected by writing a software command. The data, status register, etc. in memory can only be read out by a read after software command input.

Program and erase operations are controlled using software commands.

The following explains about bus operation modes, software commands, and status register.

Bus Operation Modes

Read

The Read mode is entered by pulling the $\overline{OE}$ pin low when the $\overline{CE}$ pin is low and the $\overline{WE}$ and $\overline{RP}$ pins are high. There are two read modes: array, and status register, which are selected by software command input. In read mode, the data corresponding to each software command entered is output from the data I/O pins D_0–D_7. The read array mode is automatically selected when the device is powered on or after it exits deep power down mode.

Output Disable

The output disable mode is entered by pulling the $\overline{CE}$ pin low and the $\overline{WE}$, $\overline{OE}$, and $\overline{RP}$ pins high. Also, the data I/O pins are placed in the high-impedance state.

Standby

The standby mode is entered by driving the $\overline{CE}$ pin high when the $\overline{RP}$ pin is high. Also, the data I/O pins are placed in the high-impedance state. However, if the $\overline{CE}$ pin is set high during erase or program operation, the internal control circuit does not halt immediately and normal power consumption is required until the operation under way is completed.

Write

The write mode is entered by pulling the $\overline{WE}$ pin low when the $\overline{CE}$ pin is low and the $\overline{OE}$ and $\overline{RP}$ pins are high. In this mode, the device accepts the software commands or write data entered from the data I/O pins. A program, erase, or some other operation is initiated depending on the content of the software command entered here. The input data such as address and software command is latched at the rising edge of $\overline{WE}$ or $\overline{CE}$ whichever occurs earlier.

Deep Power Down

The deep power down is entered by pulling the $\overline{RP}$ pin low. Also, the data I/O pins are placed in the high-impedance state. When the device is freed from deep power down mode, the read array mode is selected and the content of the status register is set to "80_{16}". If the $\overline{RP}$ pin is pulled low during erase or program operation, the operation under way is canceled and the data in the relevant block becomes invalid.

Table 12 Relationship between control signals and bus operation modes

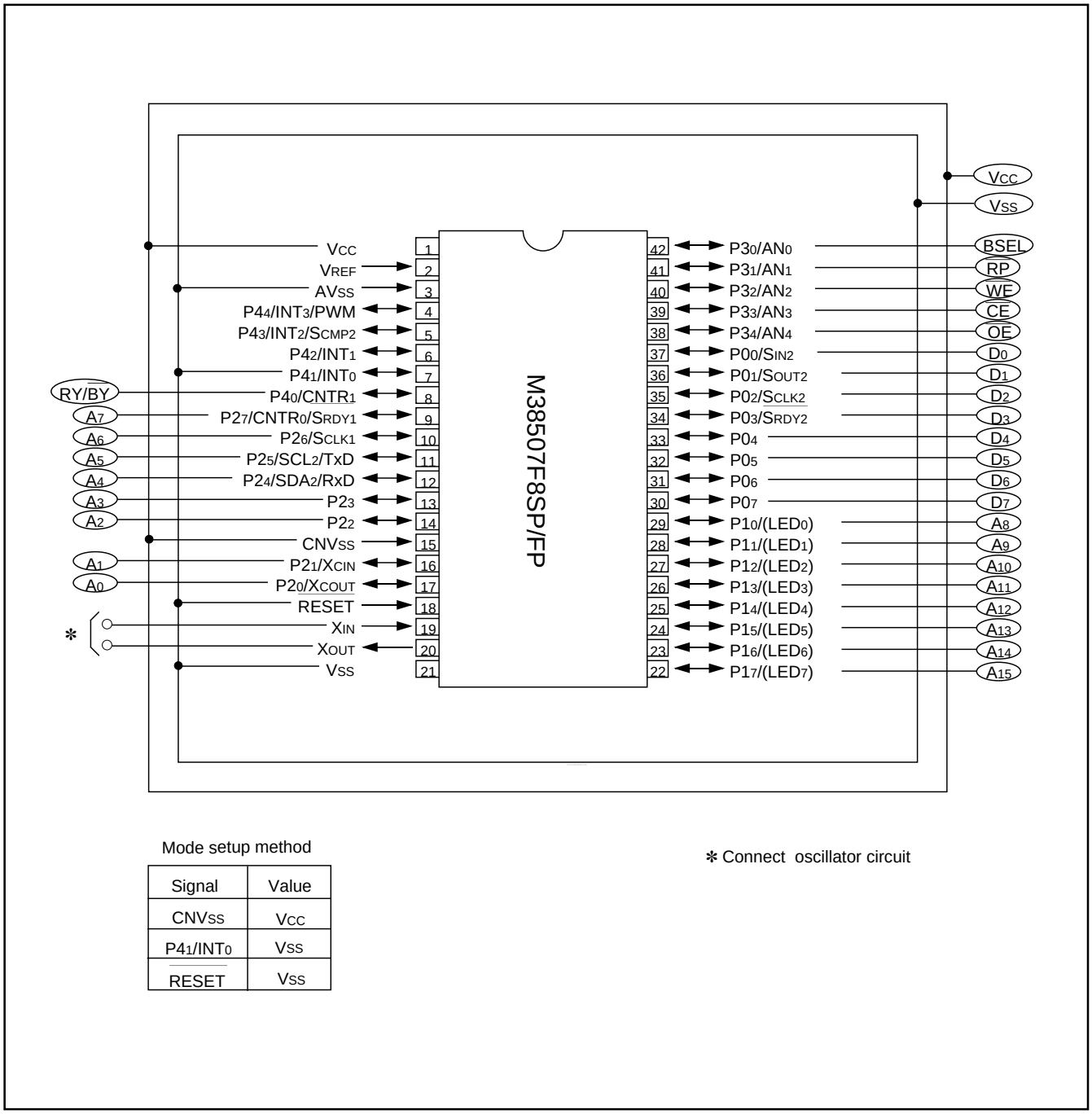
Mode \ Pin name		$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	$\overline{RP}$	D_0 to D_7
Read	Array	V_{IL}	V_{IL}	V_{IH}	V_{IH}	Data output
	Status register	V_{IL}	V_{IL}	V_{IH}	V_{IH}	Status register data output
Output disabled		V_{IL}	V_{IH}	V_{IH}	V_{IH}	High impedance
Stand by		V_{IH}	X	X	V_{IH}	High impedance
Write	Program	V_{IL}	V_{IH}	V_{IL}	V_{IH}	Command/data input
	Erase all blocks	V_{IL}	V_{IH}	V_{IL}	V_{IH}	Command input
	Block erase	V_{IL}	V_{IH}	V_{IL}	V_{IH}	Command input
Deep power down		X	X	X	V_{IL}	High impedance

Note : X can be V_{IL} or V_{IH}.

FUNCTIONAL DESCRIPTION

Table 13 Description of Pin Function (Flash Memory Parallel I/O Mode)

Pin name	Signal name	I/O	Function
V_{CC},V_{SS}	Power supply input		Apply 5.0 ± 0.5 V to the Vcc pin and 0 V to the Vss pin.
CNV$_{SS}$	CNV$_{SS}$	I	Connect this pin to V_{CC}.
$\overline{RESET}$	Reset input	I	Reset input pin. When reset is held low, more than 20 cycles of clock are required at the X_{IN} pin.
X_{IN}	Clock input	I	Connect a ceramic or crystal resonator between the X_{IN} and X_{OUT} pins. When entering an externally drived clock, enter it from X_{IN} and leave X_{OUT} open.
X_{OUT}	Clock output	O	
AV$_{SS}$	Analog power supply input		Connect AVss to Vss.
V_{REF}	Reference voltage input	I	Input AD reference voltage or keep open.
P0$_0$ to P0$_7$	Data I/O D_0 to D_7	I/O	These are data D_0–D_7 input/output pins.
P1$_0$ to P1$_7$	Address input A_8 to A_{15}	I	These are address A_8–A_{15} input pins.
P2$_0$ to P2$_7$	Address input A_0 to A_7	I	These are address A_0–A_7 input pins.
P3$_0$	BSEL input	I	This is a BSEL input pin.
P3$_1$	$\overline{RP}$ input	I	This is a $\overline{RP}$ input pin.
P3$_2$	$\overline{WE}$ input	I	This is a $\overline{WE}$ input pin.
P3$_3$	$\overline{CE}$ input	I	This is a $\overline{CE}$ input pin.
P3$_4$	$\overline{OE}$ input	I	This is a $\overline{OE}$ input pin.
P4$_0$	RY/$\overline{BY}$ output	O	This is a RY/$\overline{BY}$ output pin.
P4$_1$	Input P4$_1$	I	Enter low signals to this pin.
P4$_2$ to P4$_4$	Input P4	I	Input “H” or “L” or keep open.



Mode setup method

Signal	Value
CNVSS	VCC
P41/INT0	VSS
$\overline{\text{RESET}}$	VSS

* Connect oscillator circuit

Fig. 52 Pin connection diagram in parallel I/O mode

FUNCTIONAL DESCRIPTION

Software Commands

Table 14 lists the software commands. By entering a software command from the data I/O pins (D_0–D_7) in Write mode, specify the content of the operation, such as erase or program operation, to be performed.

The following explains the content of each software command.

Read Array Command (FF_{16})

The read array mode is entered by writing the command code "FF_{16}" in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the content of the specified address is output from the data I/O pins (D_0–D_7).

The read array mode is retained intact until another command is written.

The read array mode is also selected automatically when the device is powered on and after it exits deep power down mode.

Read Status Register Command (70_{16})

When the command code "70_{16}" is written in the first bus cycle, the content of the status register is output from the data I/O pins (D_0–D_7) by a read in the second bus cycle. Since the content of the status register is updated at the falling edge of $\overline{OE}$ or $\overline{CE}$, the $\overline{OE}$ or $\overline{CE}$ signal must be asserted each time the status is read. The status register is explained in the next section.

Clear Status Register Command (50_{16})

This command is used to clear the bits SR4,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 "50_{16}" in the first bus cycle.

Table 14 Software command list (parallel I/O mode)

Command	Cycle number	First bus cycle			Second bus cycle		
		Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read array	1	Write	X(Note 4)	FF_{16}			
Read status register	2	Write	X	70_{16}	Read	X	SRD(Note 1)
Clear status register	1	Write	X	50_{16}			
Program	2	Write	X	40_{16}	Write	WA(Note 2)	WD(Note 2)
All block erase	2	Write	X	20_{16}	Write	X	20_{16}
Block erase	2	Write	X	20_{16}	Write	BA(Note 3)	$D0_{16}$

Notes 1: SRD = Status Register Data

2: WA = Write Address, WD = Write Data

3: BA = Block Address (Enter the maximum address of each block)

4: X denotes a given address in the user ROM area or boot ROM area.

Program Command (40$_{16}$)

The program operation starts when the command code "40$_{16}$" 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 reading the status register or the RY/$\overline{BY}$ signal status. When the program starts, the read status register mode is accessed automatically and the content of the status register can be read out from the data bus (D_0–D_7). 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 (FF$_{16}$) is written.

The RY/$\overline{BY}$ pin is "L" during write operation and "H" 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.

Erase All Blocks Command (20$_{16}$/20$_{16}$)

By writing the command code "20$_{16}$" in the first bus cycle and the confirmation command code "20$_{16}$" in the second bus cycle that follows, the system starts erase all blocks(erase and erase verify).

Whether the erase all blocks command is terminated can be confirmed by reading the status register or the RY/$\overline{BY}$ signal status . When the erase all blocks operation starts, the read status register mode is accessed automatically and the content of the status register can be read out. The status register bit 7 (SR7) is set to "0" at the same time the erase operation starts and is returned to "1" upon completion of the erase operation. In this case, the read status register mode remains active until the read array command (FF$_{16}$) is written.

The RY/$\overline{BY}$ pin is "L" during erase operation and "H" when the erase operation is completed as is the status register bit 7.

At erase all blocks end, erase results can be checked by reading the status register. For details, refer to the section where the status register is detailed.

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$_{16}$" in the second bus cycle that follows to the block address of a flash memory block, the system initiates a block erase (erase and erase verify) operation.

Whether the block erase operation is completed can be confirmed by reading the status register or the RY/$\overline{BY}$ signal. At the same time the block erase operation starts, the read status register mode is automatically entered, so the content 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/$\overline{BY}$ pin is "L" during block erase operation and "H" when the block erase operation is completed as is the status register bit 7.

After the block erase operation is completed, the status register can be read out to know the result of the block erase operation. For details, refer to the section where the status register is detailed.

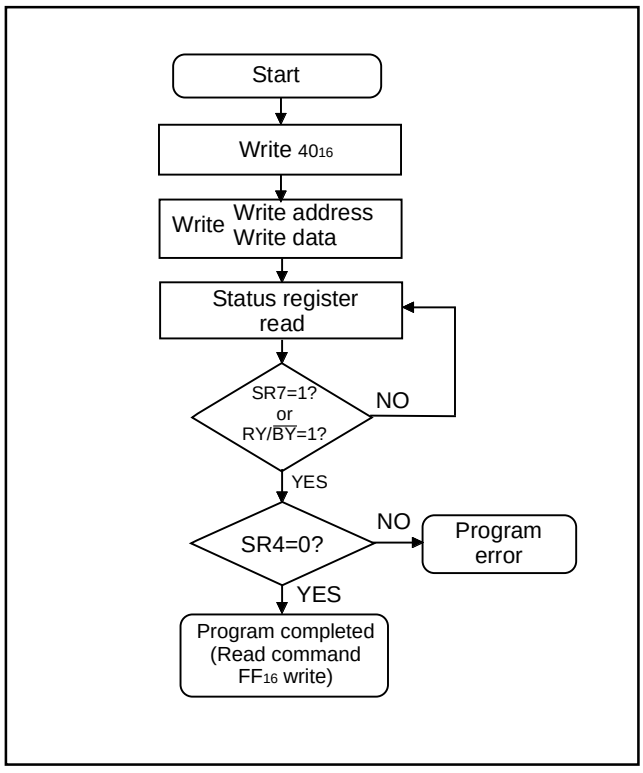


Fig. 53 Page program flowchart

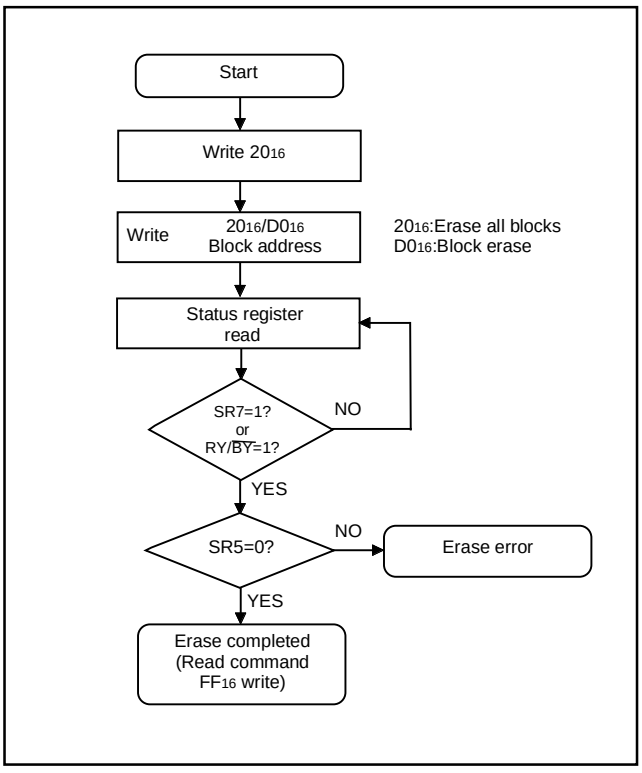


Fig. 54 Block erase flowchart

FUNCTIONAL DESCRIPTION

Status Register

The status register indicates status such as whether an erase operation or a program ended successfully or in error. It can be read under the following conditions.

(1) In the read array mode when the read status register command (70_{16}) is written and the block address is subsequently read.
(2) In the period from when the program write or auto erase starts to when the read array command (FF_{16})

The status register is cleared in the following situations.

(1) By writing the clear status register command (50_{16})
(2) In the deep power down mode
(3) In the power supply off state

Table 15 gives the definition of each status register bit. When power is turned on or returning from the deep power down mode, the status register outputs "80_{16}".

Sequencer status (SR7)

The sequencer status indicates the operating status of the flash memory. When power is turned on or returning from the deep power down mode, "1" is set for it. This bit is "0" (busy) during the write or erase operations and becomes "1" when these operations ends.

Erase Status (SR5)

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

Program Status (SR4)

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

If "1" is written for any of the SR5, SR4 bits, the program erase all blocks, block erase, commands are not accepted. Before executing these commands, execute the clear status register command (50_{16}) and clear the status register.

Also, any commands are not correct, both SR5 and SR4 are set to "1".

Full Status Check

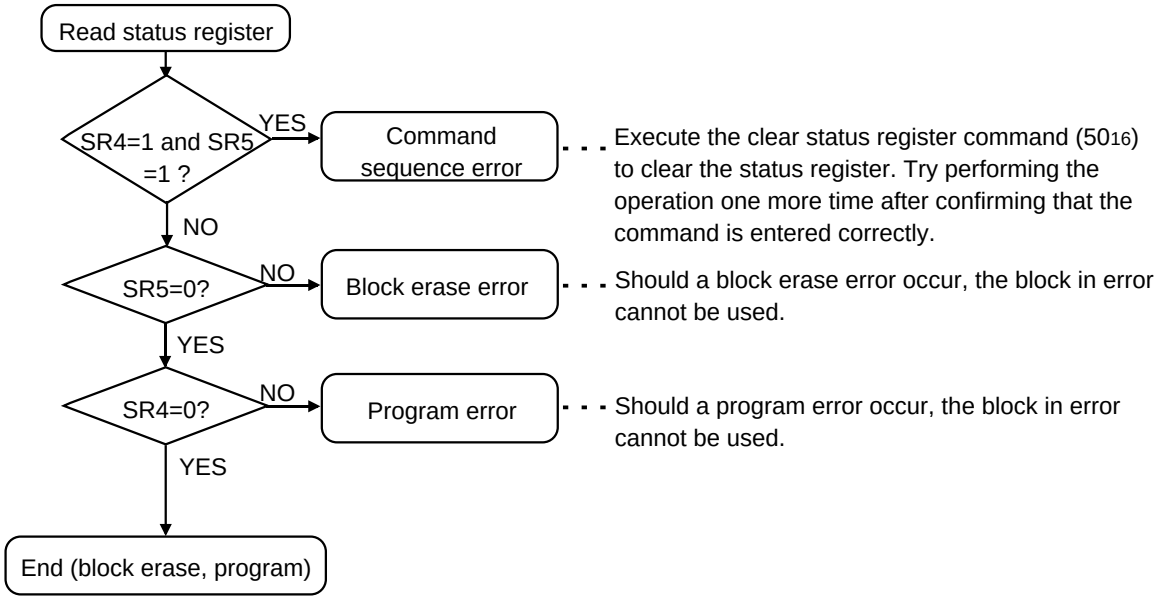
Results from executed erase and program operations can be known by running a full status check. Figure 55 shows a flowchart of the full status check and explains how to remedy errors which occur.

Ready/Busy (RY/$\overline{BY}$) pin

The RY/$\overline{BY}$ pin is an output pin (N-chanel open drain output) which, like the sequencer status (SR7), indicates the operating status of the flash memory. It is "L" level during auto program or auto erase operations and becomes to the high impedance state (ready state) when these operations end. The RY/$\overline{BY}$ pin requires an external pull-up.

Table 15 Status register

Each bit of SRD0 bits	Status name	Definition	
		"1"	"0"
SR7 (D_7)	Sequencer status	Ready	Busy
SR6 (D_6)	Reserved	-	-
SR5 (D_5)	Erase status	Ended in error	Ended successfully
SR4 (D_4)	Program status	Ended in error	Ended successfully
SR3 (D_3)	Reserved	-	-
SR2 (D_2)	Reserved	-	-
SR1 (D_1)	Reserved	-	-
SR0 (D_0)	Reserved	-	-



Note: When one of SR5 to SR4 is set to “1” , none of the program, all blocks erase, or block erase is accepted. Execute the clear status register command (50_{16}) before executing these commands.

Fig. 55 Full status check flowchart and remedial procedure for errors

FUNCTIONAL DESCRIPTION

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 modes require 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's rewrite mode), rewrite data input and so forth. The standard serial I/O mode is started by connecting “H” to the $P2_6$ (S_{CLK1}) pin and the $P4_1$(INT_0) pin and “H” to the CNV_{SS} pin (when V_{CC} = 4.5 V to 5.5 V, connect to V_{CC}; when V_{CC} = 2.7 V to 4.5 V, supply 4.5 V to 5.5 V to Vpp from an external source), and releasing the reset operation. (In the ordinary command mode, set CNVss pin to “L” level.)

This control program is written in the boot ROM area when the product is shipped from Renesas Technology Corp. Accordingly, make note of the fact that the standard serial I/O mode cannot be used if the boot ROM area is rewritten in the parallel I/O mode. Figure 56 shows the pin connections for the standard serial I/O mode. Serial data I/O uses SI/O1 data serially in 8-bit units.

To use standard serial I/O mode. The operation uses the four SI/O1 pins S_{CLK1}, RxD, TxD and $\overline{S_{RDY1}}$ (BUSY). The S_{CLK1} pin is the transfer clock input pin through which an external transfer clock is input. The TxD pin is for CMOS output. The $\overline{S_{RDY1}}$ (BUSY) pin outputs an “L” level when ready for reception and an “H” level when reception starts.

In the standard serial I/O mode, only the user ROM area indicated in Figure 44 can be rewritten. The boot ROM cannot.

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

Overview of standard serial I/O mode

In standard serial I/O mode, software commands, addresses and data are input and output between the MCU and peripheral units (serial programer, etc.) using 4-wire clock-synchronized serial I/O (SI/O1).

In reception, software commands, addresses and program data are synchronized with the rise of the transfer clock that is input to the S_{CLK1} pin, and are then input to the MCU via the RxD pin. In transmission, the read data and status are synchronized with the fall of the transfer clock, and output from the TxD pin.

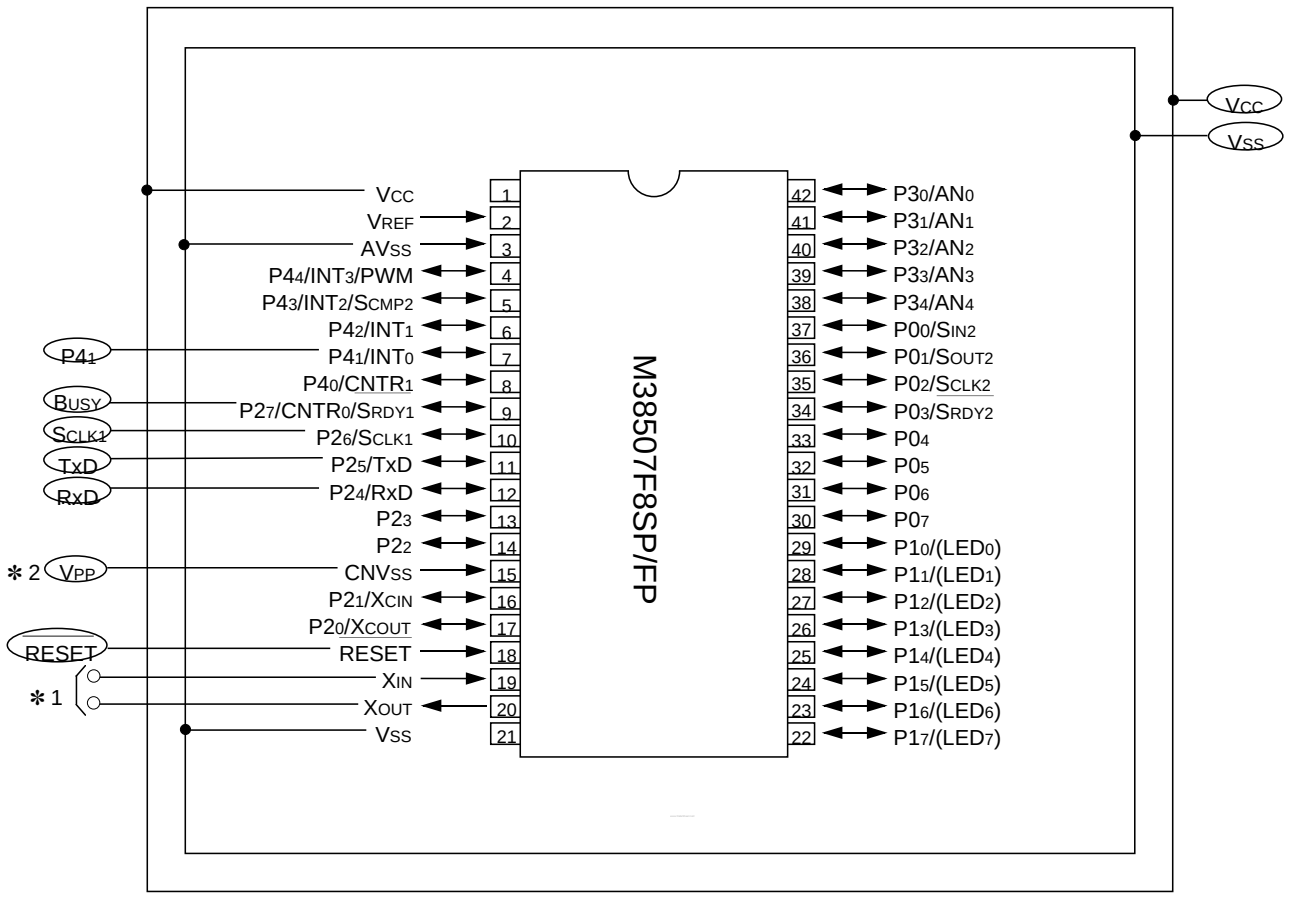
The TxD pin is for CMOS output. Transfer is in 8-bit units with LSB first.

When busy, such as during transmission, reception, erasing or program execution, the $\overline{S_{RDY1}}$ (BUSY) pin is “H” level. Accordingly, always start the next transfer after the $\overline{S_{RDY1}}$ (BUSY) pin is “L” level.

Also, data and status registers in memory can be read after inputting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. Here following are explained software commands, status registers, etc.

Table 16 Pin functions (Flash memory standard serial I/O mode)

Pin	Name	I/O	Description
V_{CC},V_{SS}	Power input		Apply program/erase protection voltage to Vcc pin and 0 V to Vss pin.
CNV_{SS}	CNV_{SS}	I	Connect to V_{CC} when V_{CC} = 4.5 V to 5.5 V. Connect to Vpp (=4.5 V to 5.5 V) when V_{CC} = 2.7 V to 4.5 V.
$\overline{RESET}$	Reset input	I	Reset input pin. While reset is "L" level, a 20 cycle or longer clock must be input to X_{IN} pin.
X_{IN}	Clock input	I	Connect a ceramic resonator or crystal oscillator between X_{IN} and X_{OUT} pins. To input an externally generated clock, input it to X_{IN} pin and open X_{OUT} pin.
X_{OUT}	Clock output	O	
AV_{SS}	Analog power supply input		Connect AV_{SS} to V_{SS} .
V_{REF}	Reference voltage input	I	Enter the reference voltage for AD from this pin.
$P0_0$ to $P0_7$	Input port P0	I	Input "H" or "L" level signal or open.
$P1_0$ to $P1_7$	Input port P1	I	Input "H" or "L" level signal or open.
$P2_0$ to $P2_3$	Input port P2	I	Input "H" or "L" level signal or open.
$P2_4$	RxD input	I	Serial data input pin
$P2_5$	TxD output	O	Serial data output pin
$P2_6$	S_{CLK1} input	I	Serial clock input pin
$P2_7$	BUSY output	O	BUSY signal output pin
$P3_0$ to $P3_4$	Input port P3	I	Input "H" or "L" level signal or open.
$P4_0$, $P4_2$ to $P4_4$	Input port P4	I	Input "H" or "L" level signal or open.
$P4_1$	Input $P4_1$	I	Input "H" level signal, when reset is released.



Mode setup method

Signal	Value
CNVss	4.5 to 5.5 V
P26/SCLK1	Vcc ✻3
P41/INT0	Vcc ✻3
RESET	Vss → Vcc

Notes 1: Connect oscillator circuit
2: Connect to Vcc when Vcc = 4.5 V to 5.5 V.
Connect to VPP (=4.5 V to 5.5 V) when Vcc = 2.7 V to 4.5 V.
3: It is necessary to apply Vcc only when reset is released.

Fig. 56 Connection for serial I/O mode

Software Commands

Table 17 lists software commands. In the standard serial I/O mode, erase operations, programs and reading are controlled by transferring software commands via the RxD pin. Software commands are explained here below. Basically, the software commands of the standard serial I/O mode is as same as that of the parallel I/O mode, but it is excluded 1 command of block erase, and it is added 3 command of ID check, download function, version data output function.

Table 17 Software commands (Standard serial I/O mode 1)

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verified
1	Page read	FF_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41_{16}	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Erase all blocks	$A7_{16}$	$D0_{16}$						Not acceptable
4	Read status register	70_{16}	SRD output	SRD1 output					Acceptable
5	Clear status register	50_{16}							Not acceptable
6	ID check	$F5_{16}$	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
7	Download function	FA_{16}	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
8	Version data output function	FB_{16}	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable

Notes 1: Shading indicates transfer from flash memory microcomputer to peripheral unit. All other data is transferred from the peripheral unit to the flash memory microcomputer.

2: SRD refers to status register data. SRD1 refers to status register 1 data.

3: All commands can be accepted when the flash memory is totally blank.

4: Address high (A_{16} to A_{23}) must be "00_{16}".

FUNCTIONAL DESCRIPTION

Page Read Command

This command reads the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page read command as explained here following.

(1) Transfer the "FF_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} ("00_{16}") with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first in sync with the fall of the clock.

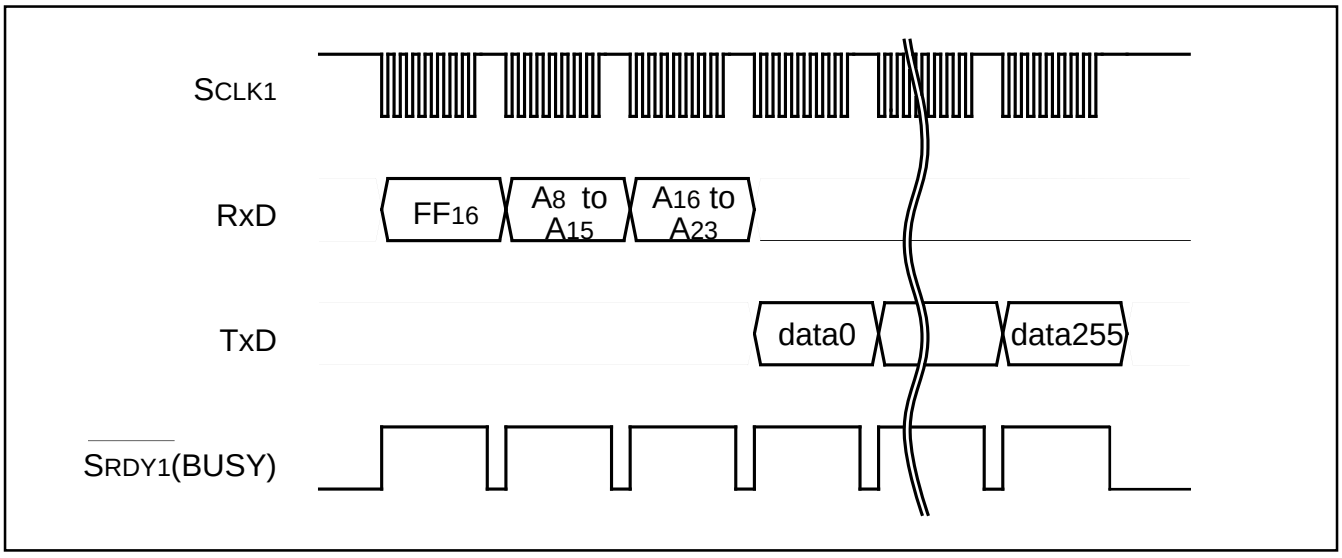


Fig. 57 Timing for page read

Read Status Register Command

This command reads status information. When the "70_{16}" command code is sent with the 1st byte, the contents of the status register (SRD) specified with the 2nd byte and the contents of status register 1 (SRD1) specified with the 3rd byte are read.

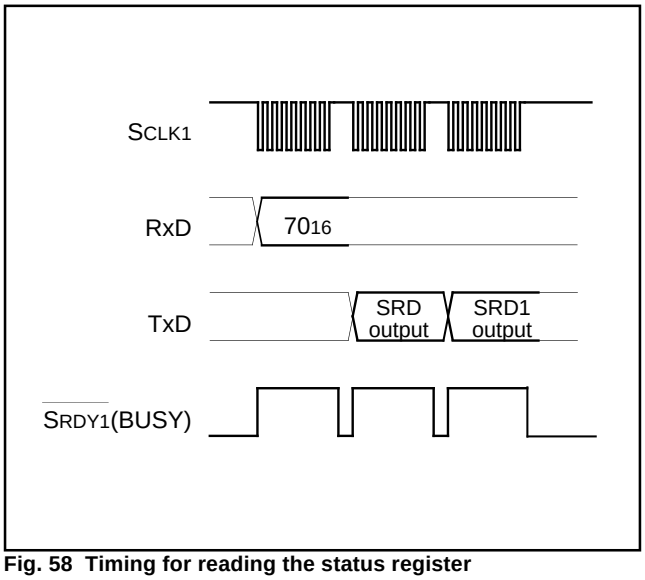


Fig. 58 Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR4–SR5) which are set when the status register operation ends in error. When the "50_{16}" command code is sent with the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the $\overline{SRDY1}$ (BUSY) signal changes from the "H" to the "L" level.

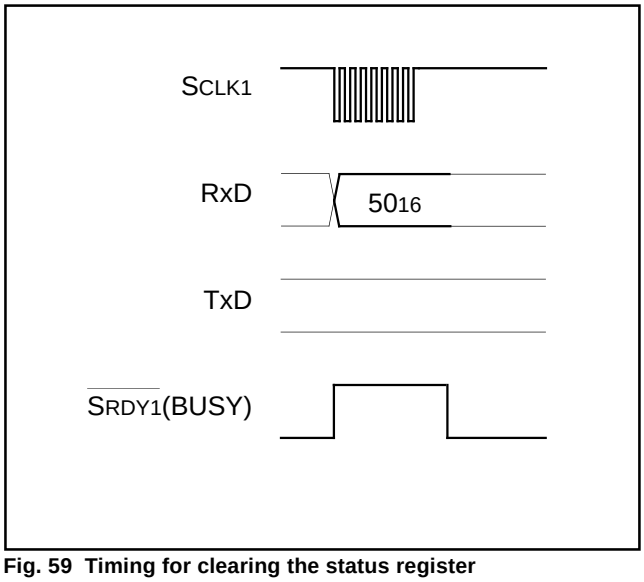


Fig. 59 Timing for clearing the status register

Page Program Command

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

(1) Transfer the "41$_{16}$" command code with the 1st byte.

(2) Transfer addresses A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ ("00$_{16}$") with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, as write data (D$_0$–D$_7$) for the page (256 bytes) specified with addresses A$_8$ to A$_{23}$ is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the $\overline{\text{SRDY1}}$ (BUSY) signal changes from the "H" to the "L" level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

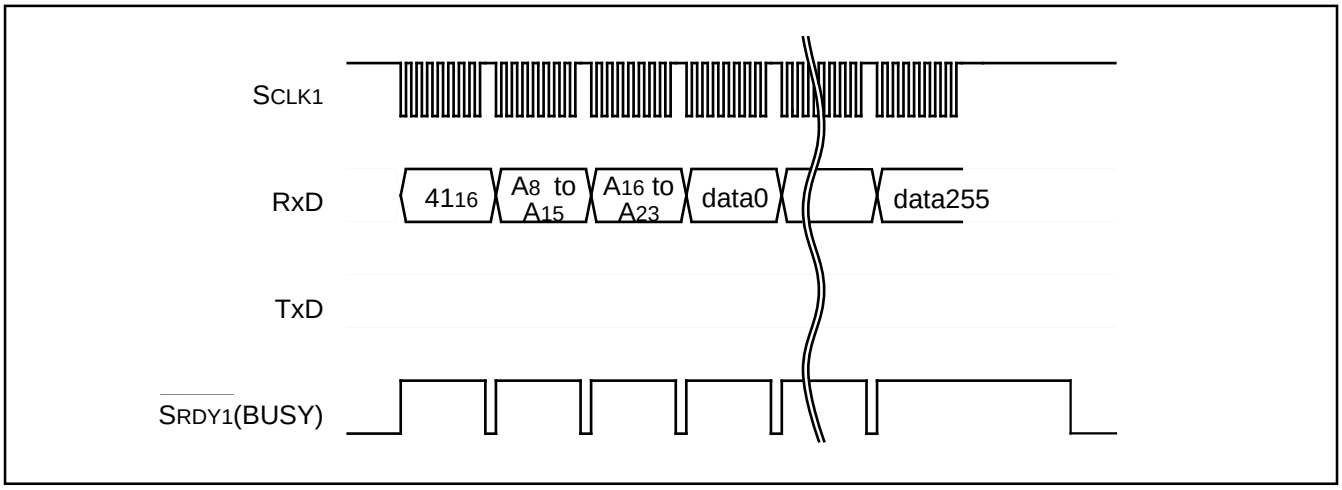


Fig. 60 Timing for the page program

Erase All Blocks Command

This command erases the content of all blocks. Execute the erase all blocks command as explained here following.

(1) Transfer the "A7$_{16}$" command code with the 1st byte.

(2) Transfer the verify command code "D0$_{16}$" with the 2nd byte. With the verify command code, the erase operation will start and continue for all blocks in the flash memory.
When block erasing ends, the $\overline{\text{SRDY1}}$ (BUSY) signal changes from the "H" to the "L" level . The result of the erase operation can be known by reading the status register.

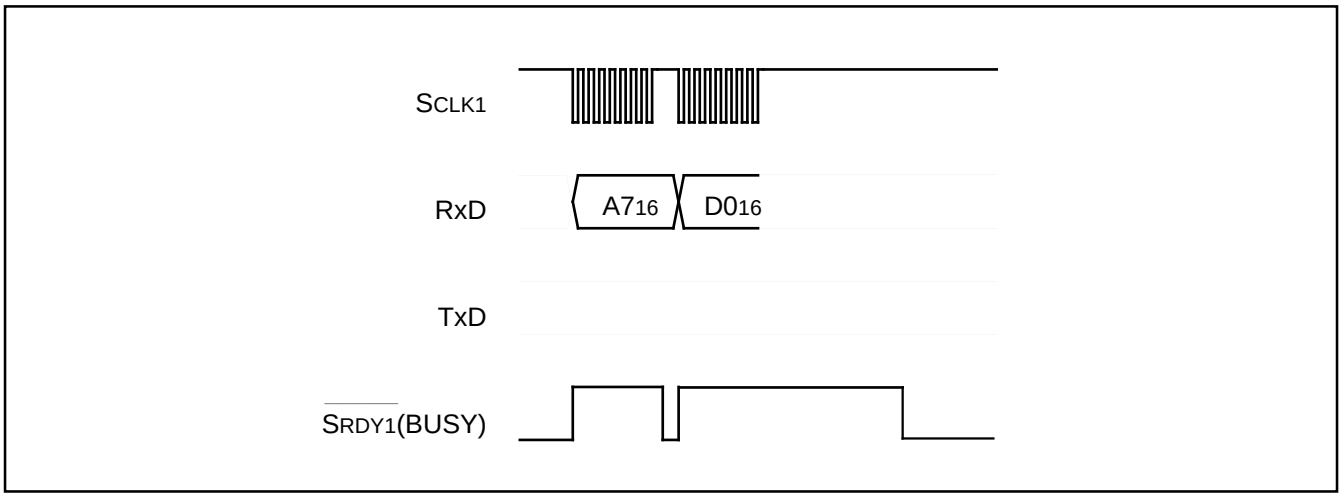


Fig. 61 Timing for erasing all blocks

FUNCTIONAL DESCRIPTION

Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

(1) Transfer the "FA16" command code with the 1st byte.
(2) Transfer the program size with the 2nd and 3rd bytes.
(3) Transfer the check sum with the 4th byte. The check sum is added to all data sent with the 5th byte onward.
(4) The program to execute is sent with the 5th byte onward.

When all data has been transmitted, if the check sum matches, the downloaded program is executed. The size of the program will vary according to the internal RAM.

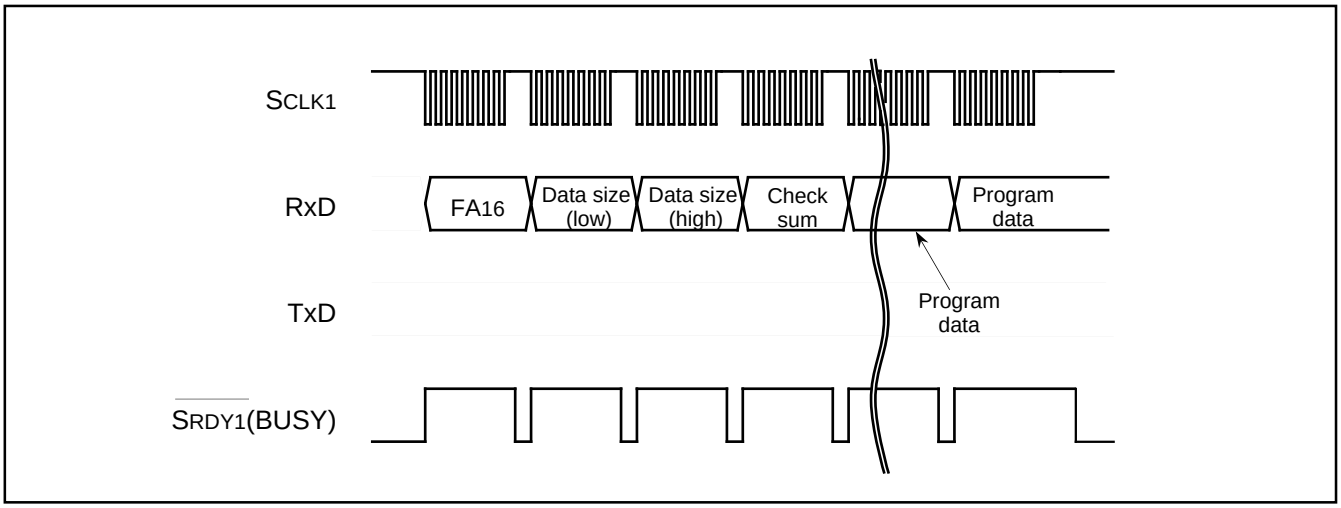


Fig. 62 Timing for download

Version Information Output Command

This command outputs the version information of the control program stored in the boot area. Execute the version information output command as explained here following.

(1) Transfer the "FB16" command code with the 1st byte.
(2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

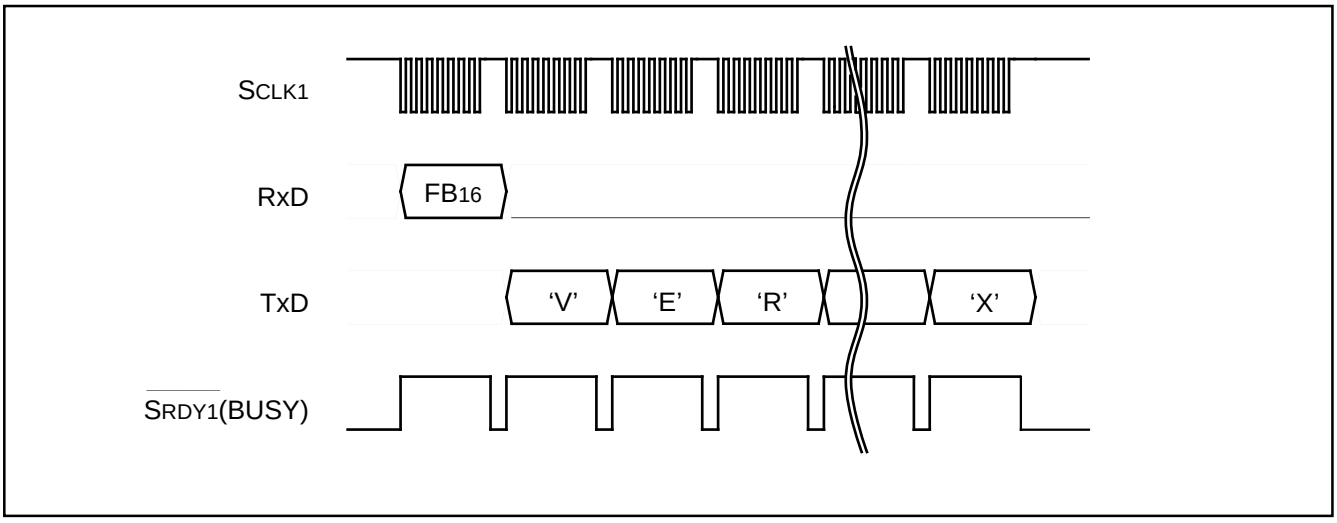


Fig. 63 Timing for version information output

ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

(1) Transfer the "F5$_{16}$" command code with the 1st byte.

(2) Transfer addresses A$_0$ to A$_7$, A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ ("00$_{16}$") of the 1st byte of the ID code with the 2nd, 3rd and 4th bytes respectively.

(3) Transfer the number of data sets of the ID code with the 5th byte.

(4) The ID code is sent with the 6th byte onward, starting with the 1st byte of the code.

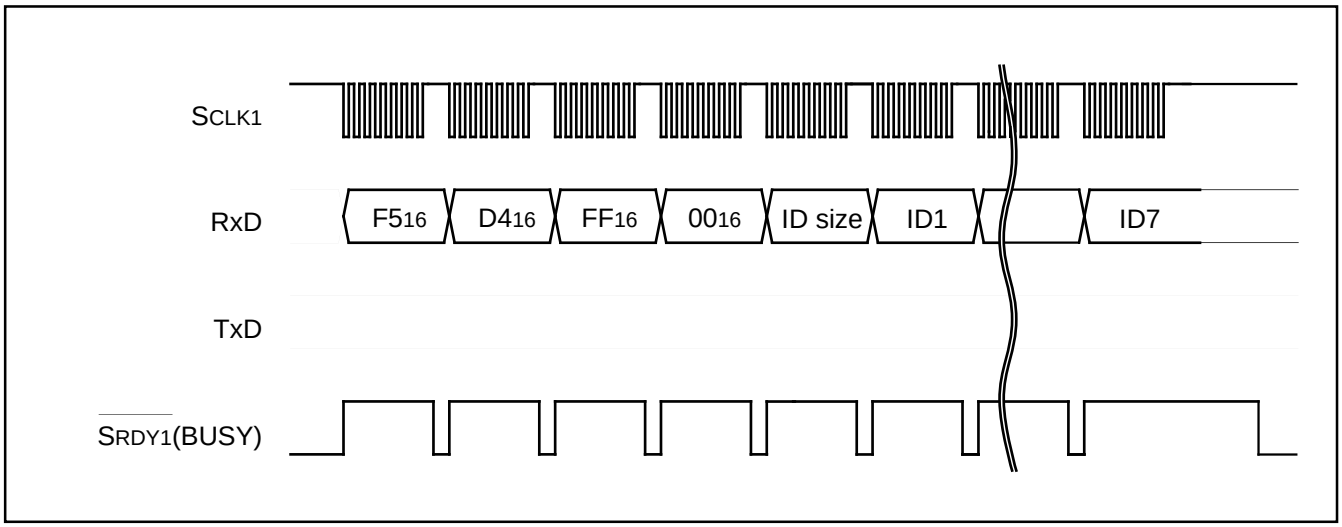


Fig. 64 Timing for the ID check

ID Code

When the flash memory is not blank, the ID code sent from the peripheral units and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the peripheral units is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses FFD4$_{16}$ to FFDA$_{16}$. Write a program into the flash memory, which already has the ID code set for these addresses.

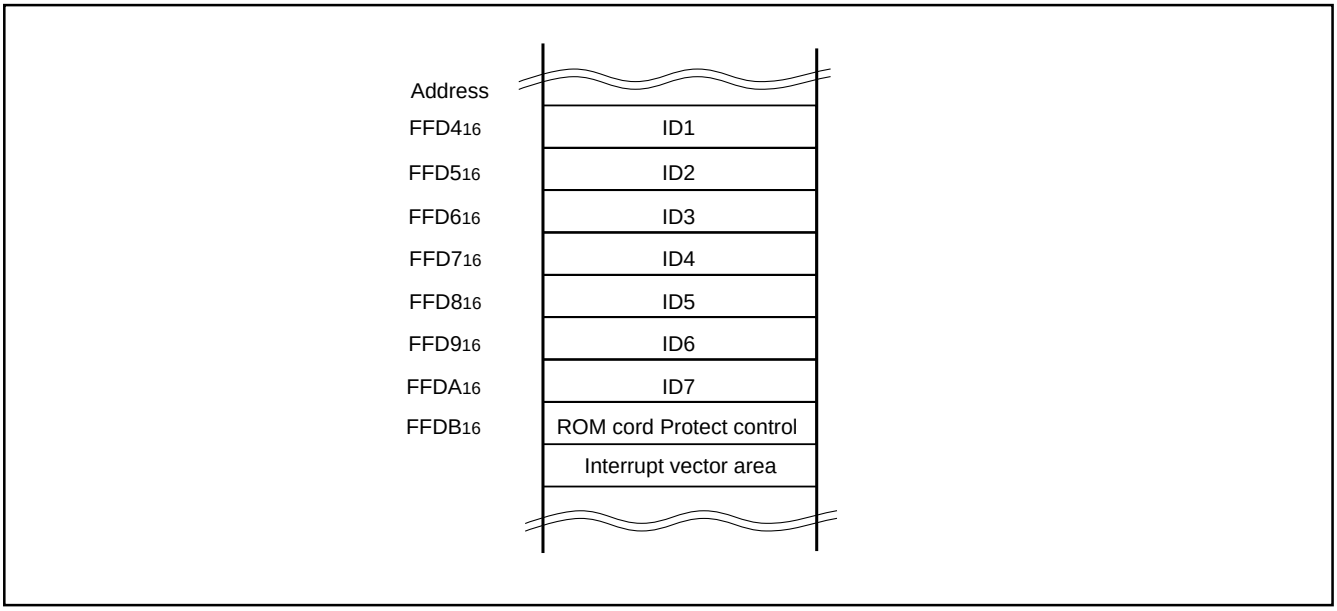


Fig. 65 ID code storage addresses

FUNCTIONAL DESCRIPTION

Status Register (SRD)

The status register indicates operating status of the flash memory and status such as whether an erase operation or a program ended successfully or in error. It can be read by writing the read status register command (70_{16}). Also, the status register is cleared by writing the clear status register command (50_{16}).
Table 18 gives the definition of each status register bit. After clearing the reset, the status register outputs "80_{16}".

Sequencer status (SR7)

After power-on and recover from deep power down mode, the sequencer status is set to "1"(ready).

The sequencer status indicates the operating status of the device. This status bit is set to "0" (busy) during write or erase operation and is set to "1" upon completion of these operations.

Erase Status (SR5)

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

Program Status (SR4)

The program status reports the operating status of the auto write operation. If a write error occurs, it is set to "1". When the program status is cleared, it is set to "0".

Table 18 Status register (SRD)

SRD0 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	-	-

Status Register 1 (SRD1)

Status register 1 indicates the status of serial communications, results from ID checks and results from check sum comparisons. It can be read after the SRD by writing the read status register command (70_{16}). Also, status register 1 is cleared by writing the clear status register command (50_{16}).
Table 19 gives the definition of each status register bit. "00_{16}" is output when power is turned on and the flag status is maintained even after the reset.

Boot Update Completed Bit (SR15)

This flag indicates whether the control program was downloaded to the RAM or not, using the download function.

Check Sum Consistency Bit (SR12)

This flag indicates whether the check sum matches or not when a program is downloaded for execution using the download function.

ID Check Completed Bits (SR11 and SR10)

These flags indicate the result of ID checks. Some commands cannot be accepted without an ID check.

Data Reception Time Out (SR9)

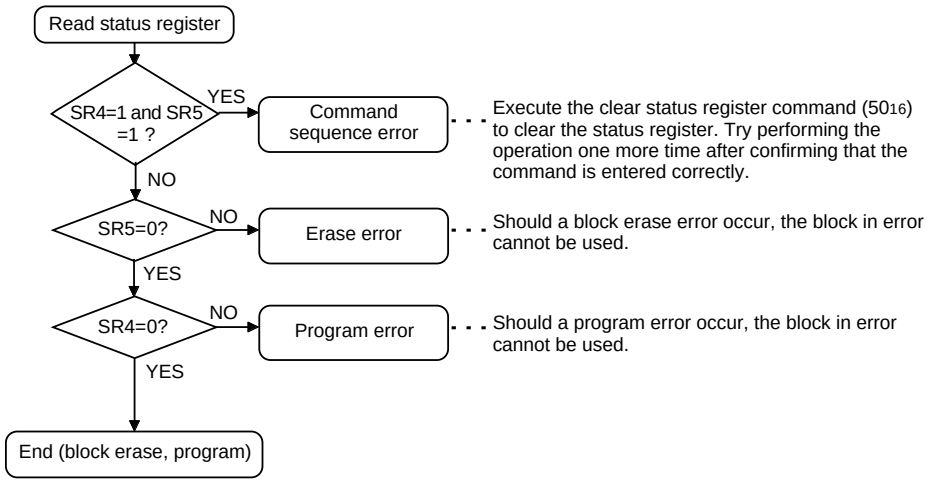
This flag indicates when a time out error is generated during data reception. If this flag is attached during data reception, the received data is discarded and the microcomputer returns to the command wait state.

Table 19 Status register 1 (SRD1)

SRD1 bits	Status name	Definition	
		"1"	"0"
SR15 (bit7)	Boot update completed bit	Update completed	Not Update
SR14 (bit6)	Reserved	-	-
SR13 (bit5)	Reserved	-	-
SR12 (bit4)	Checksum match bit	Match	Mismatch
SR11 (bit3) SR10 (bit2)	ID check completed bits	00 01 10 11	Not verified Verification mismatch Reserved Verified
SR9 (bit1)	Data reception time out	Time out	Normal operation
SR8 (bit0)	Reserved	-	-

Full Status Check

Results from executed erase and program operations can be known by running a full status check. Figure 66 shows a flowchart of the full status check and explains how to remedy errors which occur.

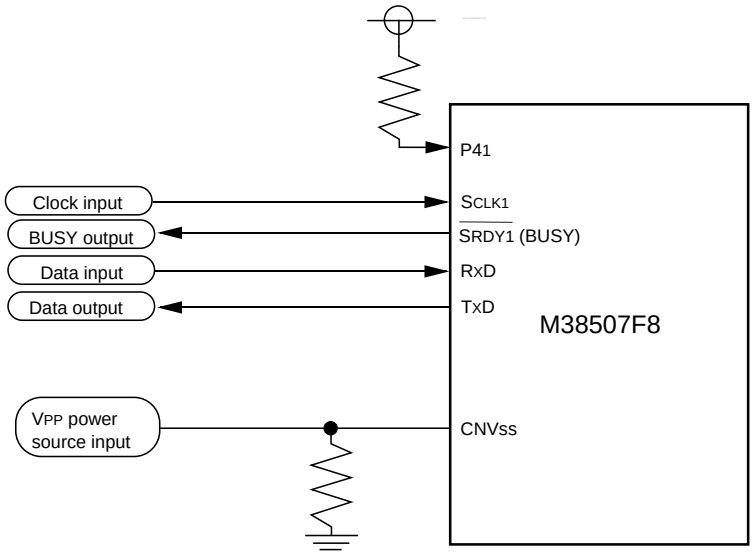


Note: When one of SR5 to SR4 is set to "1", none of the program, erase all blocks commands is accepted. Execute the clear status register command (50_{16}) before executing these commands.

Fig. 66 Full status check flowchart and remedial procedure for errors

Example Circuit Application for The Standard Serial I/O Mode

Figure 67 shows a circuit application for the standard serial I/O mode. Control pins will vary according to programmer, therefore see the peripheral unit manual for more information.



Notes 1: Control pins and external circuitry will vary according to peripheral unit. For more information, see the peripheral unit manual.
2: In this example, the Vpp power supply is supplied from an external source (writer). To use the user's power source, connect to 4.5 V to 5.5 V.
3: It is necessary to apply Vcc to SCLK pin only when reset is released.

Fig. 67 Example circuit application for the standard serial I/O mode

Flash memory Electrical characteristics

Table 20 Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Power source voltage	All voltages are based on V_{SS}. Output transistors are cut off.	–0.3 to 6.5	V
V_I	Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$, $P2_1$, $P2_4$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$, V_{REF}		–0.3 to V_{CC} +0.3	V
V_I	Input voltage $P2_2$, $P2_3$		–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 CNV_{SS}		–0.3 to V_{CC} +0.3	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$, $P2_1$, $P2_4$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$, X_{OUT}		–0.3 to V_{CC} +0.3	V
V_O	Output voltage $P2_2$, $P2_3$		–0.3 to 5.8	V
P_d	Power dissipation	T_a = 25 °C	1000 (Note)	mW
T_{opr}	Operating temperature		25±5	°C
T_{stg}	Storage temperature		–40 to 125	°C

Note: The rating becomes 300 mW at the 42P2R-A/E package.

Table 21 Flash memory mode Electrical characteristics
(Ta = 25°C, V_{CC} = 4.5 to 5.5V unless otherwise noted)

Symbol	Parameter	Conditions	Limits			Unit
			Min.	Typ.	Max.	
I_{PP1}	V_{PP} power source current (read)	V_{PP} = V_{CC}			100	μA
I_{PP2}	V_{PP} power source current (program)	V_{PP} = V_{CC}			60	mA
I_{PP3}	V_{PP} power source current (erase)	V_{PP} = V_{CC}			30	mA
V_{IL}	"L" input voltage (Note)		0		0.8	V
V_{IH}	"H" input voltage (Note)		2.0		V_{CC}	V
V_{PP}	V_{PP} power source voltage		4.5		5.5	V
V_{CC}	V_{CC} power source voltage	Microcomputer mode operation at V_{CC} = 2.7 to 5.5V	4.5		5.5	V
		Microcomputer mode operation at V_{CC} = 2.7 to 3.6V	3.0		3.6	V

Note: Input pins for parallel I/O mode.

AC Electrical characteristics
(Ta = 25°C, Vcc = 4.5 to 5.5V unless otherwise noted)

Table 22 Read-only mode

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_{RC}	Read cycle time	200			ns
$t_{a\ (AD)}$	Address access time			100	ns
$t_{a\ (CE)}$	$\overline{CE}$ access time			100	ns
$t_{a\ (OE)}$	$\overline{OE}$ access time			80	ns
t_{CLZ}	Output enable time (after $\overline{CE}$)	0			ns
$t_{DF(CE)}$	Output floating time (after $\overline{CE}$)			25	ns
t_{OLZ}	Output enable time (after $\overline{OE}$)	0			ns
$t_{DF(OE)}$	Output floating time (after $\overline{OE}$)			25	ns
t_{PHZ}	Output floating time (after $\overline{PR}$)			300	ns
t_{OH}	Output valid time (after $\overline{CE}$, $\overline{OE}$, address)	0			ns
t_{OEH}	Write recovery time (before read)	200			ns
t_{PS}	$\overline{RP}$ recovery time	10			µs

Note : Timing measurement condition is showed in Figure 68.

Table 23 Read / Write mode ($\overline{WE}$ control)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_{WC}	Write cycle time	200			ns
t_{AS}	Address set up time	100			ns
t_{AH}	Address hold time	25			ns
t_{DS}	Data set up time	100			ns
t_{DH}	Data hold time	25			ns
t_{CS}	$\overline{CE}$ set up time	0			ns
t_{CH}	$\overline{CE}$ hold time	0			ns
t_{WP}	$\overline{WE}$ pulse width	100			ns
t_{WPH}	"H" write pulse width	50			ns
t_{DAP}	Program time		25		µs
t_{DAE}	Erase all blocks time		1.5		s
t_{WHRL}	RY/$\overline{BY}$ delay time			200	ns
t_{PS}	$\overline{RP}$ recovery time	10			µs

Note : The read timing parameter in the command write operation mode is same as that of the read-only mode.
Typical value is at Vcc = 5.0 V, Ta = 25 °C condition.

FUNCTIONAL DESCRIPTION

Flash memory mode Electrical characteristics
(Ta = 25°C, Vcc = 4.5 to 5.5V unless otherwise noted)

Table 24 Read / Write mode ($\overline{CE}$ control)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t_{WC}	Write cycle time	200			ns
t_{AS}	Address set up time	100			ns
t_{AH}	Address hold time	25			ns
t_{DS}	Data set up time	100			ns
t_{DH}	Data hold time	25			ns
t_{WS}	$\overline{WE}$ set up time	0			ns
t_{WH}	$\overline{WE}$ hold time	0			ns
t_{CEP}	$\overline{CE}$ pulse width	100			ns
t_{CEPH}	"H" $\overline{CE}$ pulse width	50			ns
t_{DAP}	Program time		25		µs
t_{DAE}	Erase all blocks time		1.5		s
t_{EHRL}	RY/$\overline{BY}$ delay time			200	ns
t_{PS}	$\overline{RP}$ recovery time	10			µs

Note : The read timing parameter in the command write operation mode is same as that of the read-only mode.
Typical value is at V_{CC} = 5.0 V, Ta = 25 °C condition.

Table 25 Erase and program operation

Parameter	Min.	Typ.	Max.	Unit
Erase all blocks time		1.5		s
Block erase time		1.0		s
Program time (1byte)		25		µs

Table 26 V_{CC} power up / power down timing

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{VCS}	$\overline{RP}$ = V_{IH} set up time (after rised V_{CC} = V_{CC} min.)	10			µs

Note : Miserase or miswrite may happen, in case of noise pulse due to the power supply on or off is input to the control pins. Therefore disableing the write mode is need for prevent from memory data break at the power supply on or off. 10µs (min.) waiting time is need to initiate read or write operation after V_{CC} rises to V_{CC} min. at power supply on. The memory data is protected owing to keep the $\overline{RP}$ pin V_{IL} level at power supply off. The $\overline{RP}$ pin must be kept V_{IL} level for 10µs (min.) after V_{CC} rises to V_{CC} min. at the power supply on. The $\overline{RP}$ pin must be kept V_{IL} level until the V_{CC} falls to the GND level at power supply off. $\overline{RP}$ pin doesn't have latch mode, so $\overline{RP}$ pin must be kept V_{IH} level during read, erase and program operation.

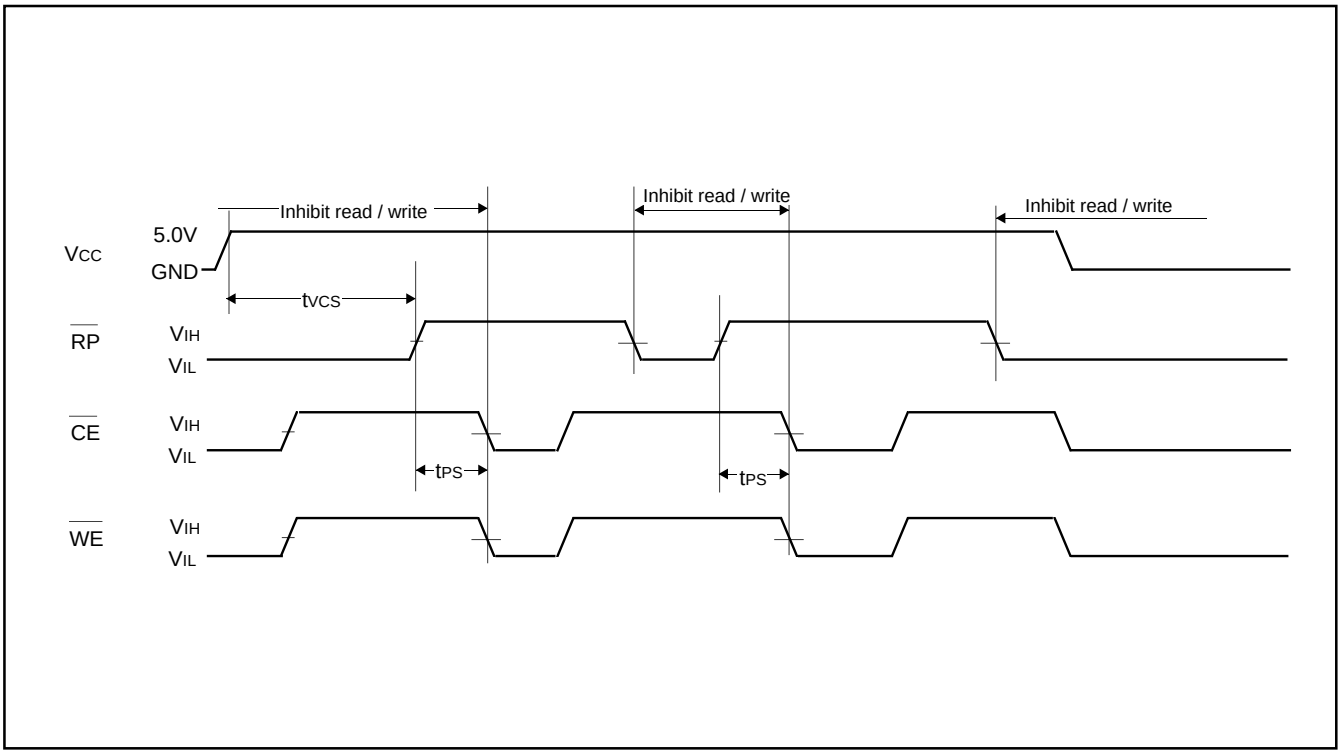


Fig. 68 Vcc power up / power down timing

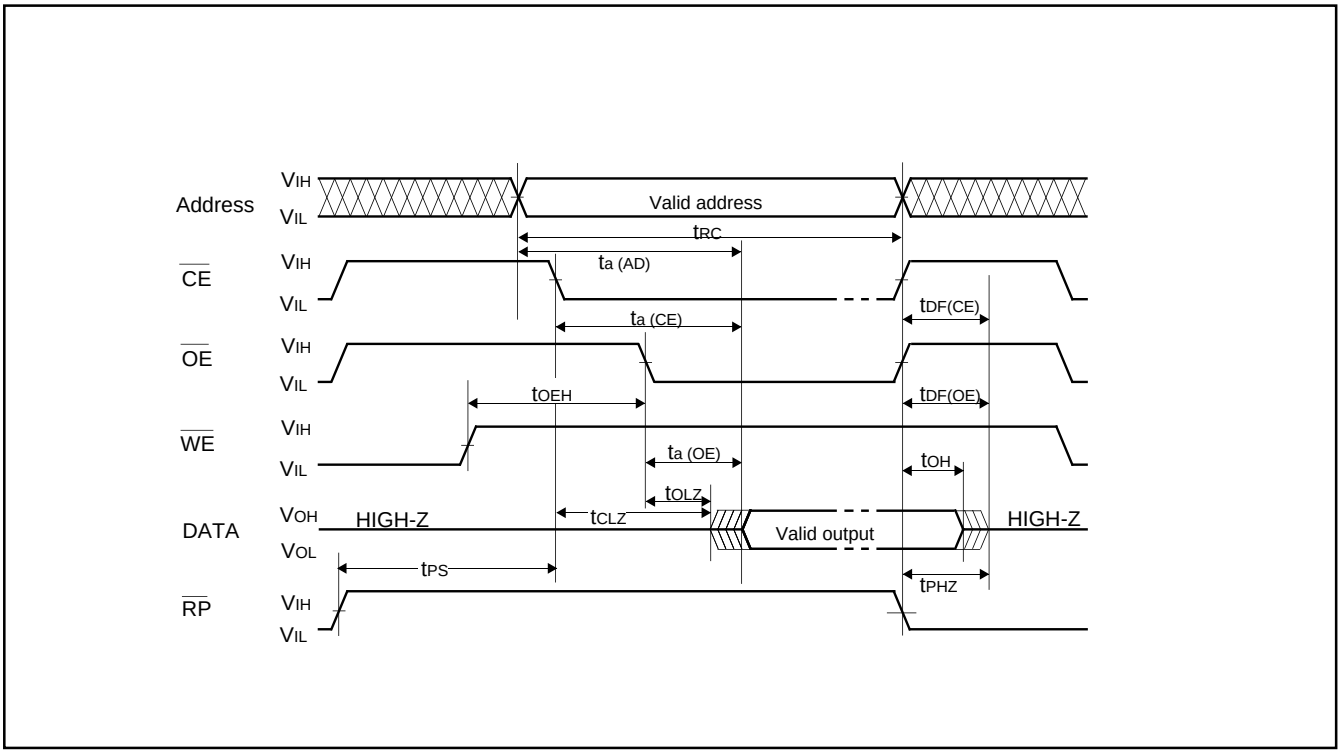


Fig. 69 AC wave for read operation

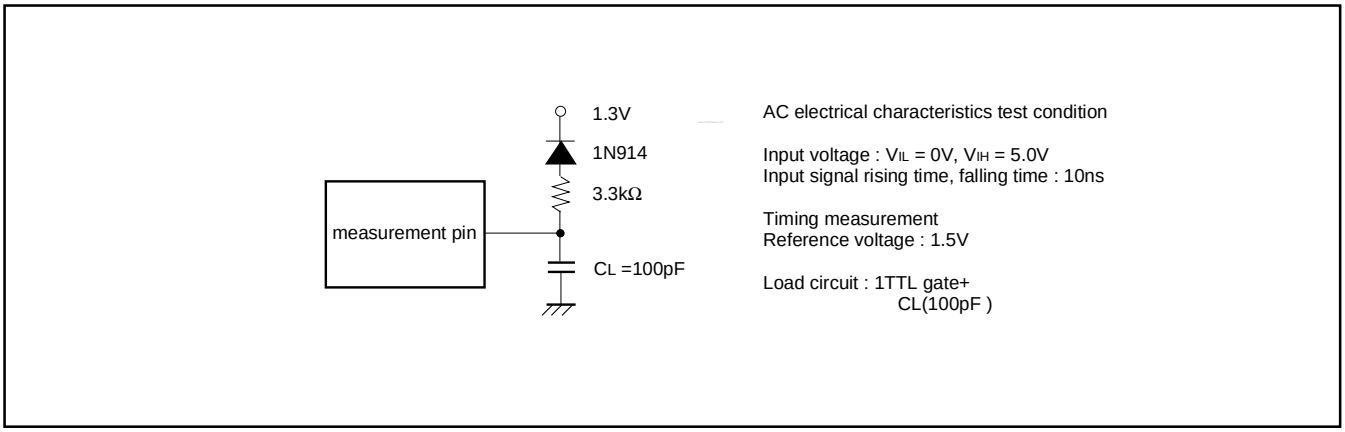


Fig. 70 AC electrical characteristics test condition for read operation

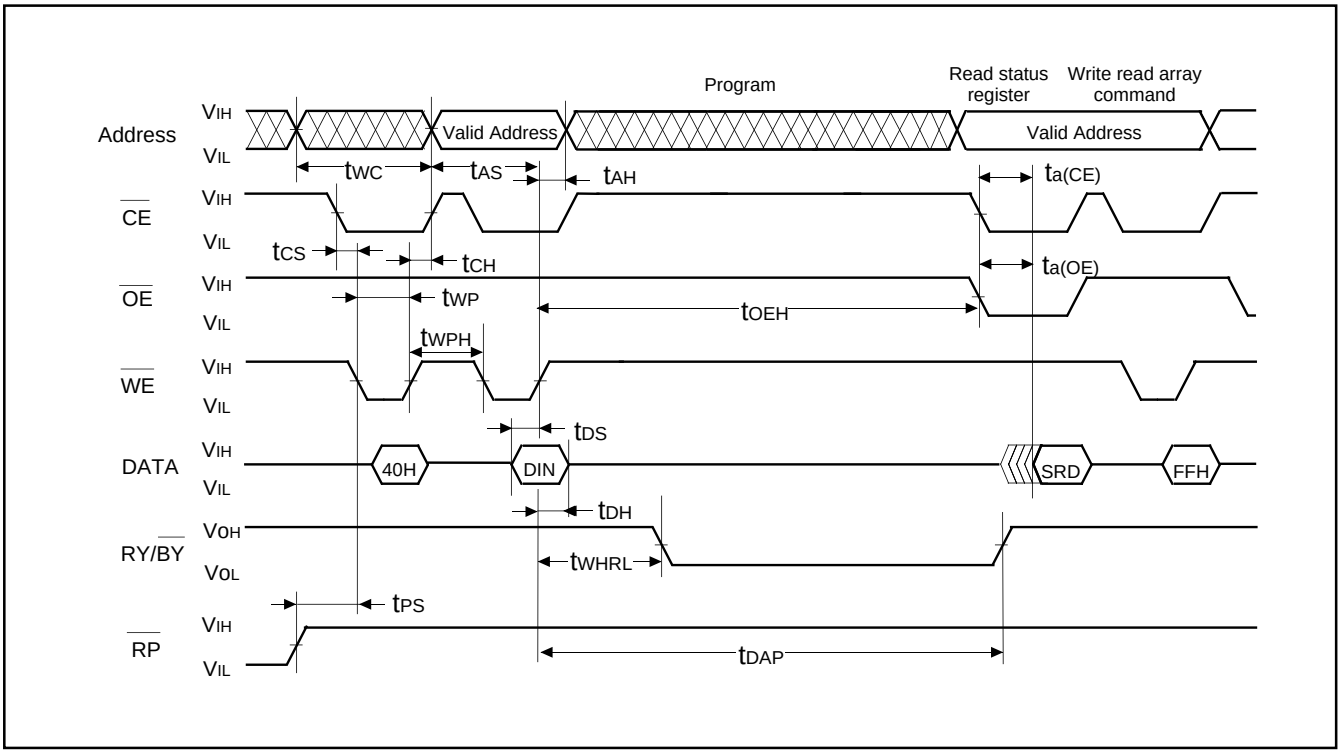


Fig. 71 AC wave for program operation (WE control)

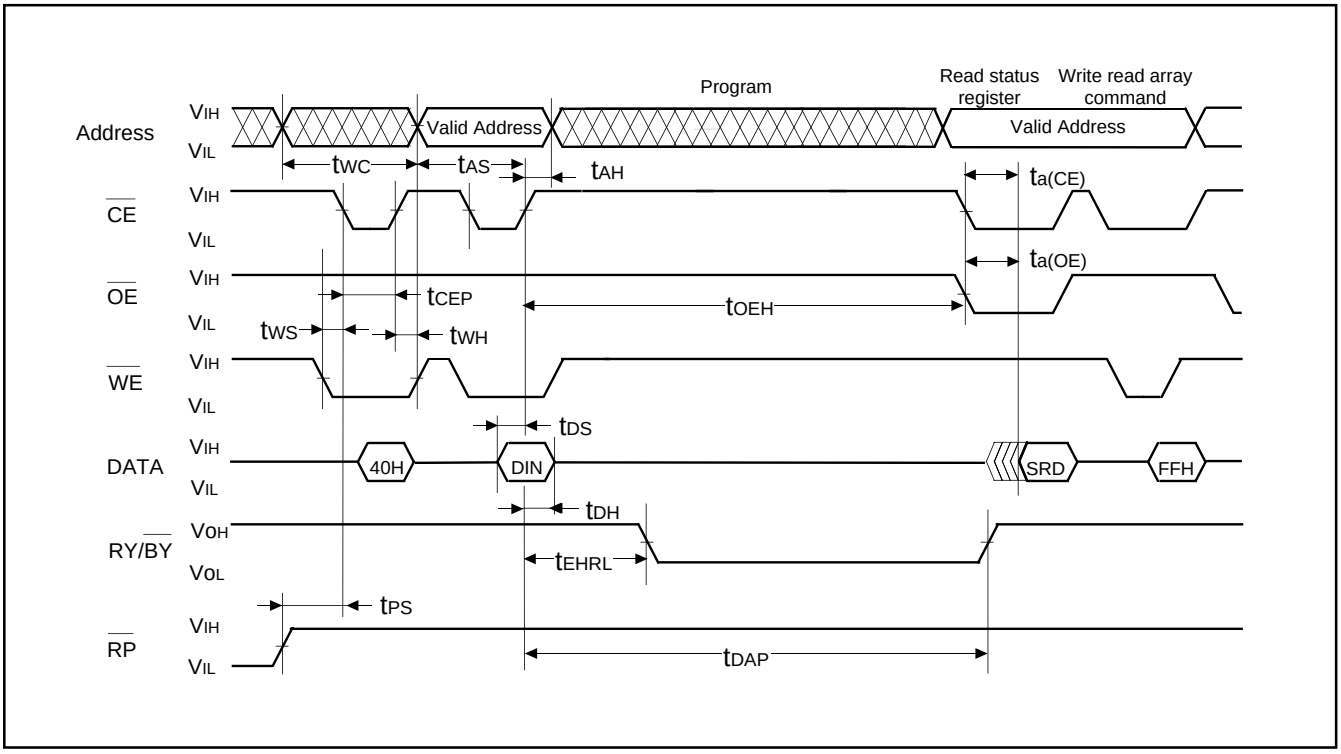


Fig. 72 AC wave for program operation (CE control)

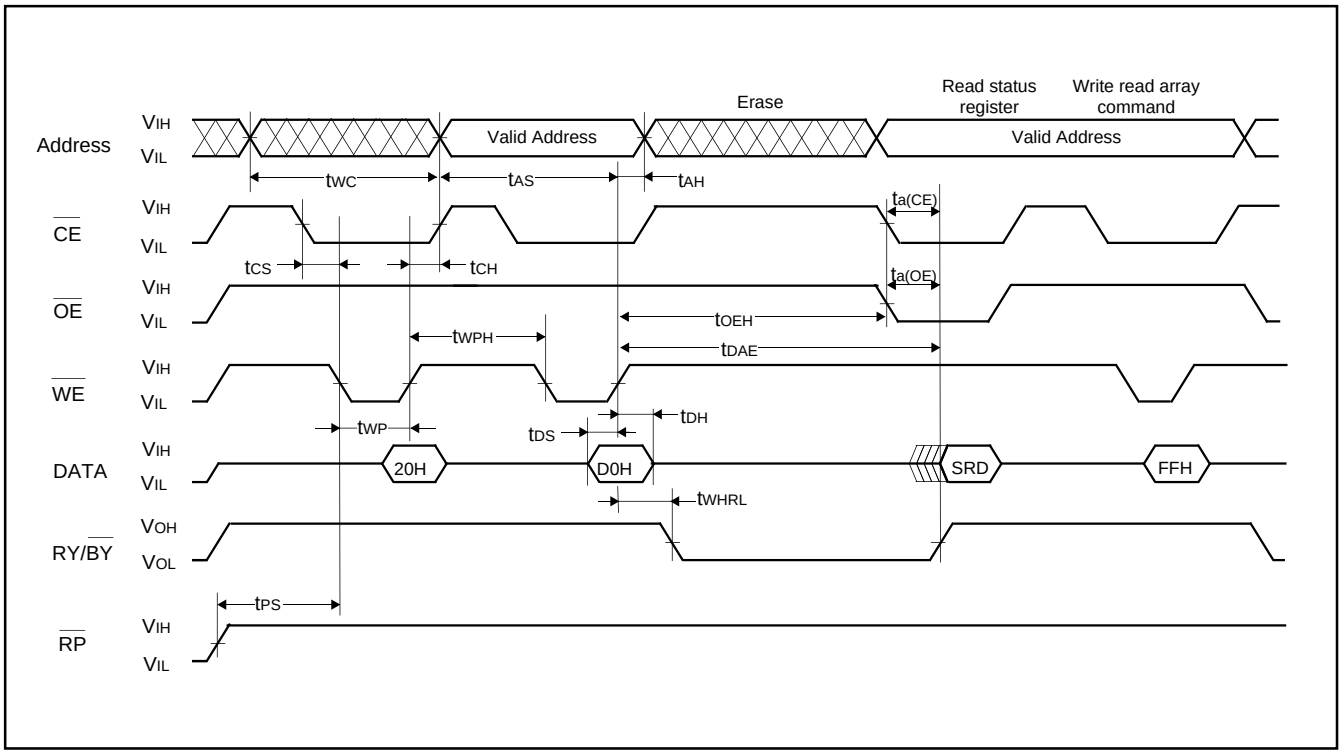


Fig. 73 AC wave for erase operation ($\overline{WE}$ control)

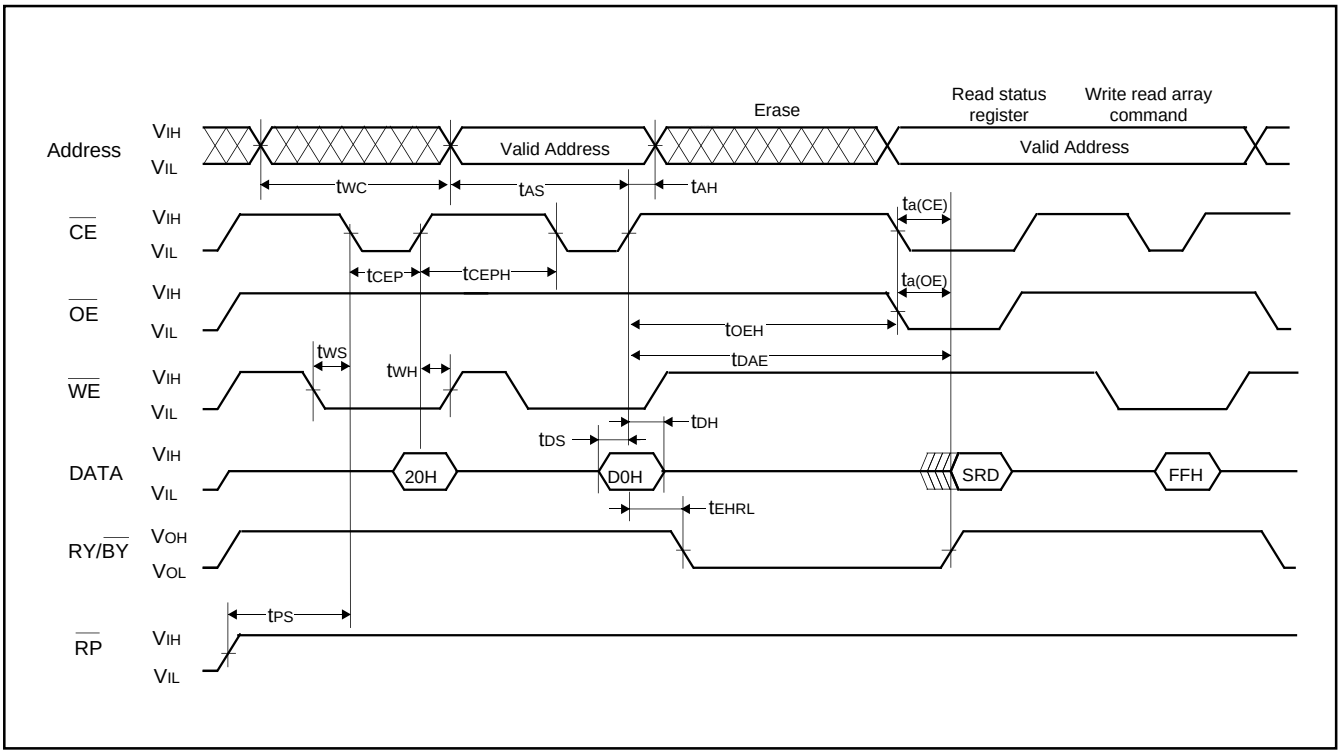


Fig. 74 AC wave for erase operation ($\overline{CE}$ control)

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 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 I/O

In clock synchronous serial I/O, if the receive side is using an external clock and it is to output the $\overline{S_{RDY1}}$ signal, set the transmit enable bit, the receive enable bit, and the $\overline{S_{RDY1}}$ output enable bit to "1".

Serial I/O1 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 transmission is completed.

When an external clock is used as synchronous clock in serial I/O1 or serial I/O2, write transmission data to the transmit buffer register or serial I/O2 register while the 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) in the middle/high-speed mode is at least on 500 kHz during an A-D conversion.

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

Instruction Execution Time

The instruction execution time is obtained by multiplying the frequency 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 frequency of the internal clock ϕ is half of the XIN frequency in high-speed mode.

NOTES ON USAGE

Differences between 3850 group (standard) and 3850 group (spec. H)

(1) The absolute maximum ratings of 3850 group (spec. H) is smaller than that of 3850 group (standard).
  •Power source voltage Vcc = –0.3 to 6.5 V
  •CNVss input voltage VI = –0.3 to Vcc +0.3 V

(2) The oscillation circuit constants of XIN-XOUT, XCIN-XCOUT may be some differences between 3850 group (standard) and 3850 group (spec. H).

(3) Do not write any data to the reserved area and the reserved bit. (Do not change the contents after rest.)

(4) Fix bit 3 of the CPU mode register to "1".

(5) Be sure to perform the termination of unused pins.

Handling of 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–0.1μF is recommended.

EPROM Version/One Time PROM Version/ Flash Memory Version

The CNVss pin is connected to the internal memory circuit block by a low-ohmic resistance, since it has the multiplexed function to be a programmable power source pin (VPP pin) as well.

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.

DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

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

DATA REQUIRED FOR One Time PROM PROGRAMMING ORDERS

The following are necessary when ordering a PROM programming service:

1. ROM Programming Confirmation Form*
2. Mark Specification Form* (only special mark with customer's trade mark logo)
3. Data to be programmed to PROM, in EPROM form (three identical copies) or one floppy disk.

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

ROM PROGRAMMING METHOD

The built-in PROM of the blank One Time PROM version and built-in EPROM version can be read or programmed with a general-purpose PROM programmer using a special programming adapter. Set the address of PROM programmer in the user ROM area.

Table 27 Programming adapter

Package	Name of Programming Adapter
42P4B, 42S1B	PCA4738S-42A
42P2R-A/E	PCA4738F-42A

The PROM of the blank One Time PROM version is not tested or screened in the assembly process and following processes. To ensure proper operation after programming, the procedure shown in Figure 75 is recommended to verify programming.

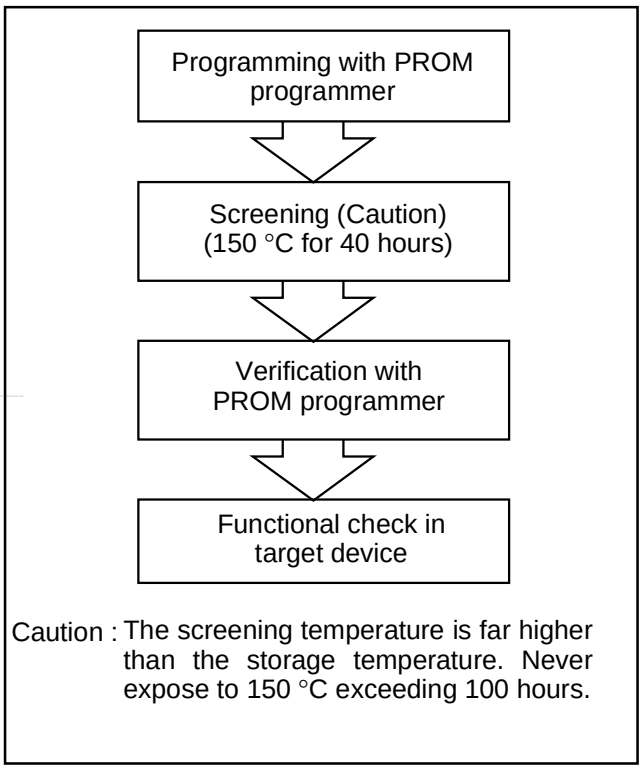


Fig. 75 Programming and testing of One Time PROM version

FUNCTIONAL DESCRIPTION SUPPLEMENT

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

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

A-D conversion completes at 61 clock cycles (15.25 μs at f(XIN) = 8 MHz) after it is started, and the result of the conversion is stored into the A-D conversion register.

Concurrently with the completion of A-D conversion, A-D conversion interrupt request occurs, so that the AD conversion interrupt request bit is set to "1".

Table 28 Relative formula for a reference voltage VREF of A-D converter and Vref

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

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

Table 29 Change of A-D conversion register during A-D conversion

	Change of A-D 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	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \cdots\cdots \pm \frac{V_{REF}}{1024}$

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

Figures 76 shows the A-D conversion equivalent circuit, and Figure 77 shows the A-D conversion timing chart.

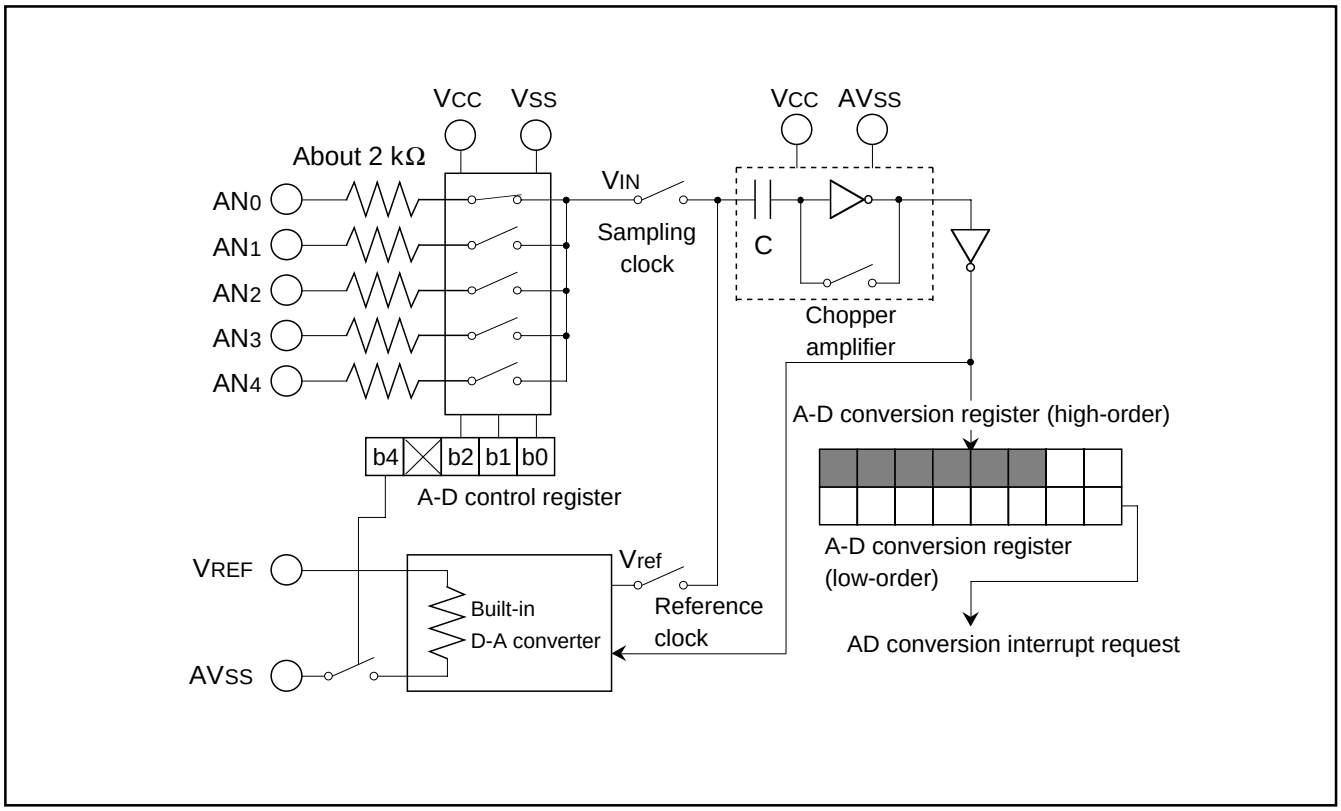


Fig. 76 A-D conversion equivalent circuit

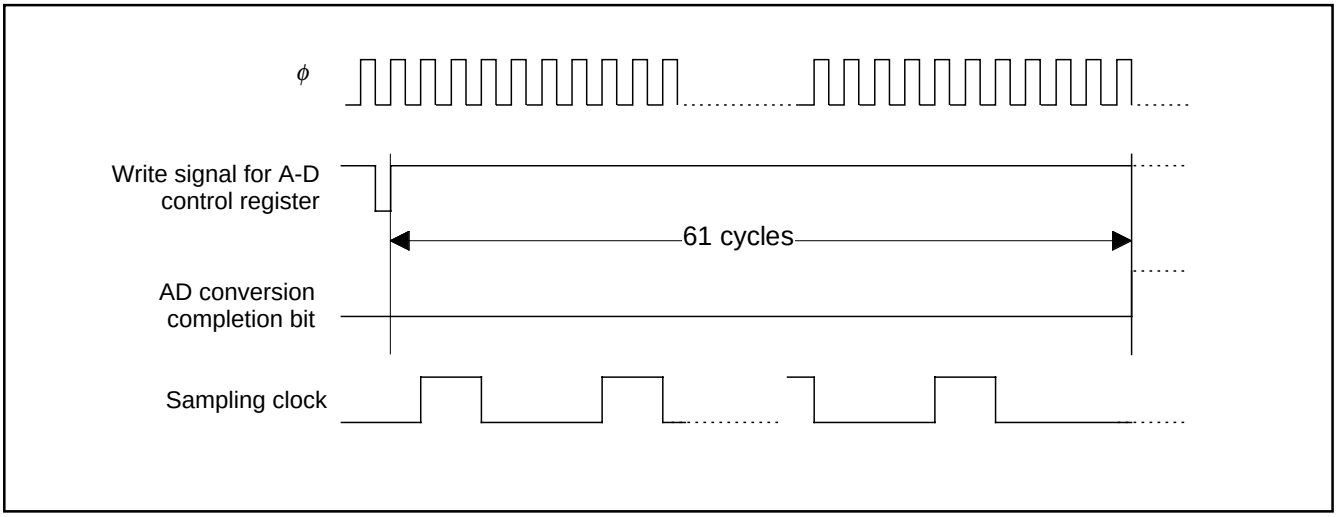


Fig. 77 A-D conversion timing chart

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

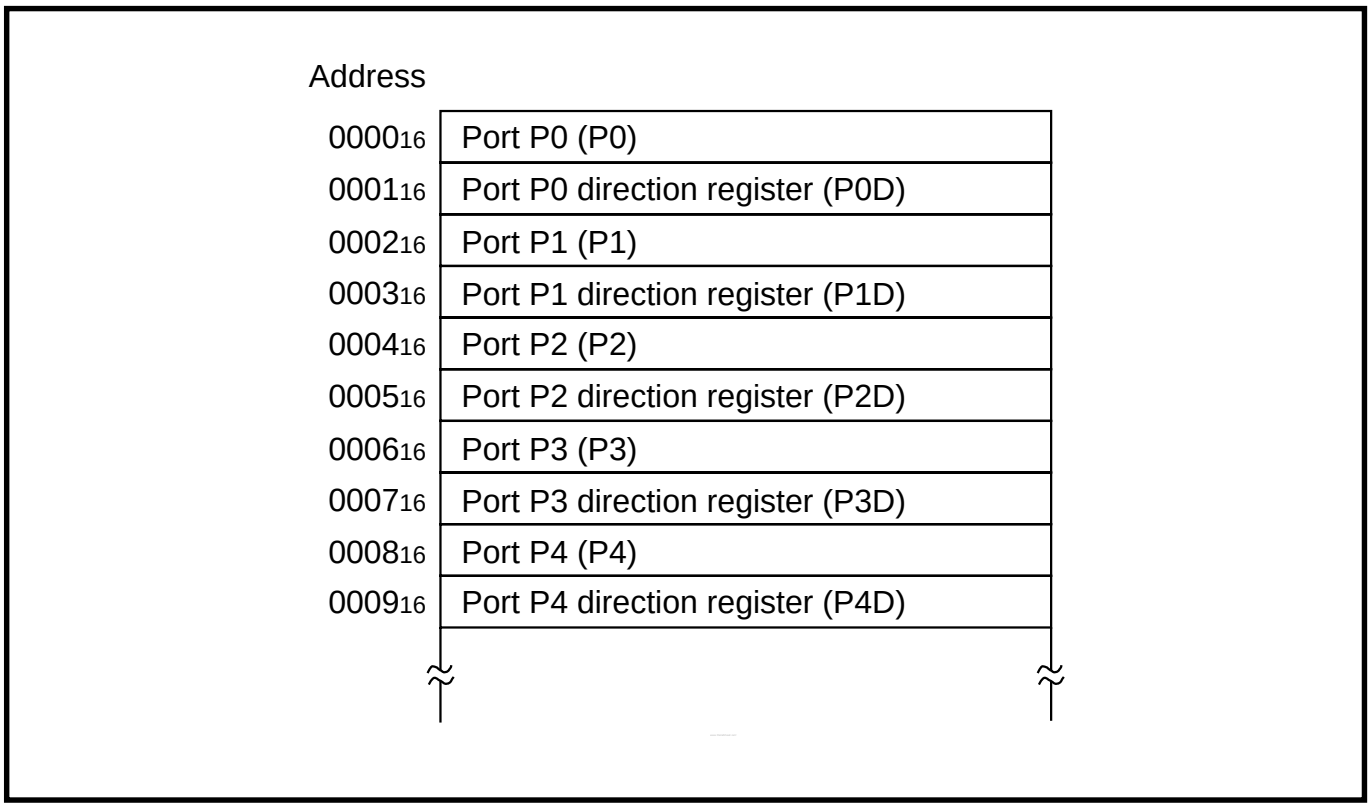


Fig. 2.1.1 Memory map of I/O port relevant registers

2.1.2 Relevant registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (i = 0, 1, 2, 3, 4)
(Pi: addresses 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{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	○	○

Note: When reading bit 5, 6 or 7 of ports 3 and 4, the contents are undefined.

Fig. 2.1.2 Structure of Port Pi (i = 0, 1, 2, 3, 4)

Port Pi direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi direction register (i = 0, 1, 2, 3, 4)
(PiD: addresses 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{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. 2.1.3 Structure of Port Pi direction register (i = 0, 1, 2, 3, 4)

2.1.3 Terminate unused pins

Table 2.1.1 Termination of unused pins

Pins/Ports name	Termination
P0, P1, P2, P3, P4	• 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 I/O port

2.1.4 Notes on I/O port

(1) Notes in standby state

In standby state*1, do not make input levels of an I/O port "undefined", especially for I/O ports of the N-channel open-drain. When setting the N-channel open-drain port as an output, do not make input levels of an I/O port "undefined", too.

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

● **Reason**

When setting as an input port with its direction register, the transistor becomes the OFF state, which causes the ports to be the high-impedance state.

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.

In I/O ports of N-channel open-drain, when the contents of the port latch are "1", even if it is set as an output port with its direction register, it becomes the same phenomenon as the case of an input port.

✻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

2.1.5 Termination of unused pins

(1) Terminate unused pins

① Output ports : Open

② Input ports :
Connect each pin to V_{CC} or V_{SS} through each resistor of 1 kΩ to 10 kΩ.
A for pins whose potential affects to operation modes such as pins CNV_{SS}, INT or others, select the V_{CC} pin or the V_{SS} pin according to their operation mode.

③ I/O ports :
- Set the I/O ports for the input mode and connect them to V_{CC} or V_{SS} through each resistor of 1 kΩ to 10 kΩ.
  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.

④ 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 Input ports and 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 ② and ③ shown on the above.

② I/O ports :
When setting for the input mode, do not connect to V_{CC} or V_{SS} 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 V_{CC} (or V_{SS}).

③ I/O ports :
When setting for the input mode, do not connect multiple ports in a lump to V_{CC} or V_{SS} 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

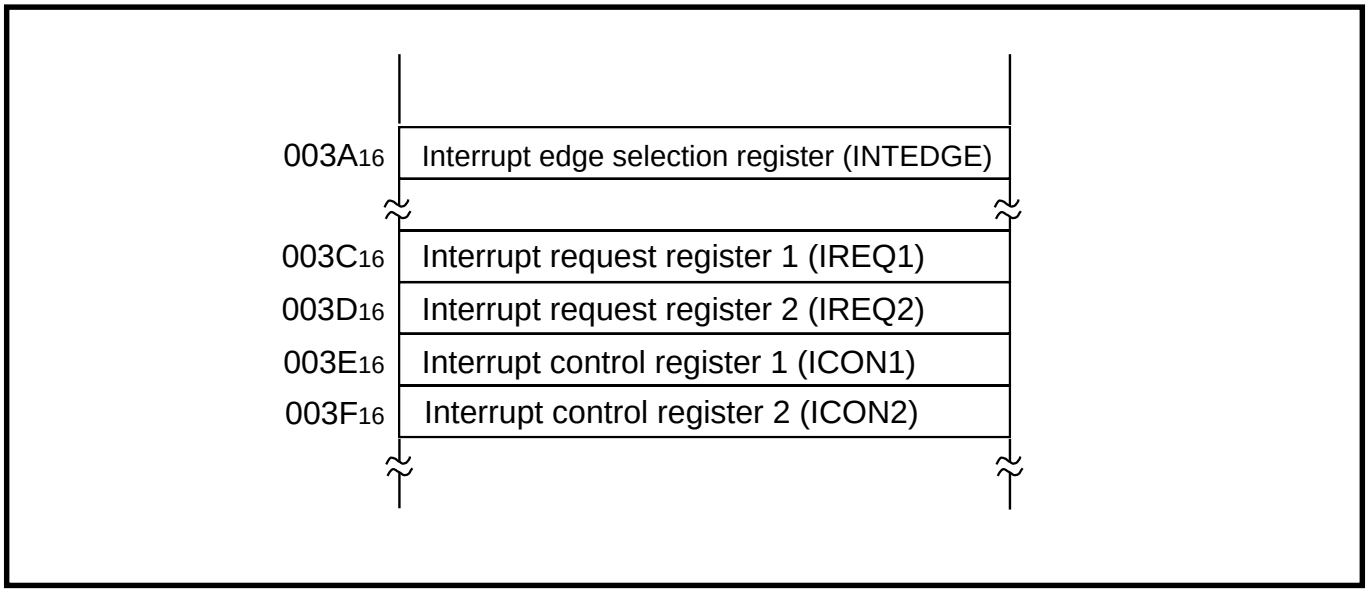
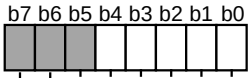


Fig. 2.2.1 Memory map of registers relevant to interrupt

2.2.2 Relevant registers

Interrupt edge selection register



Interrupt edge selection register
(INTEDGE: address $3A_{16}$)

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	INT_2 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
3	INT_3 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
4	Serial/O2/INT_3 interrupt source bit	0: INT_3 interrupt selected 1: Serial I/O2 interrupt selected	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			0	○	×
7			0	○	×

Fig. 2.2.2 Structure of Interrupt edge selection register

2.2 Interrupt

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1
(IREQ1 : address $3C_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	When writing to this bit, set “0” to this bit.		0	○	✻
2	INT_1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT_2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_3/Serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	When writing to this bit, set “0” to this bit.		0	○	✻
6	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer Y 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. 2.2.3 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2
(IREQ2 : address $3D_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	Timer 2 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 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	$CNTR_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	$CNTR_1$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	A-D converter 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. 2.2.4 Structure of Interrupt request register 2

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

0 0

Interrupt control register 1
(ICON1 : address $3E_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Fix this bit to "0".		0	○	○
2	INT_1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_3/Serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Fix this bit to "0".		0	○	○
6	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.2.5 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 $3F_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Timer 2 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 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	$CNTR_0$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	$CNTR_1$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	A-D converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to "0".		0	○	○

Fig. 2.2.6 Structure of Interrupt control register 2

2.2 Interrupt

2.2.3 Interrupt source

The 3850 group permits interrupts of 15 sources. 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 Table 2.2.1.

Table 2.2.1 Interrupt sources, vector addresses and priority of 3850 group

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)
Reserved	3	FFF916	FFF816	Reserved	
INT1	4	FFF716	FFF616	At detection of either rising or falling edge of INT1 input	External interrupt (active edge selectable)
INT2	5	FFF516	FFF416	At detection of either rising or falling edge of INT2 input	External interrupt (active edge selectable)
INT3	6	FFF316	FFF216	At detection of either rising or falling edge of INT3 input	External interrupt (active edge selectable)
Serial I/O2				At completion of serial I/O2 data transfer	Switch by Serial I/O2/INT3 interrupt source bit
Reserved	7	FFF116	FFF016	Reserved	
Timer X	8	FFEF16	FFEE16	At timer X underflow	
Timer Y	9	FFED16	FFEC16	At timer Y underflow	
Timer 1	10	FFEB16	FFEA16	At timer 1 underflow	STP release timer underflow
Timer 2	11	FFE916	FFE816	At timer 2 underflow	
Serial I/O1 received	12	FFE716	FFE616	At completion of serial I/O1 data reception	Valid when serial I/O1 is selected
Serial I/O1 transmit	13	FFE516	FFE416	At completion of serial I/O1 transfer shift or when transmission buffer is empty	Valid when serial I/O1 is selected
CNTR0	14	FFE316	FFE216	At detection of either rising or falling edge of CNTR0 input	External interrupt (active edge selectable)
CNTR1	15	FFE116	FFE016	At detection of either rising or falling edge of CNTR1 input	External interrupt (active edge selectable)
A-D converter	16	FFDF16	FFDE16	At completion of A-D conversion	
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 2.2.7 shows an interrupt operation diagram.

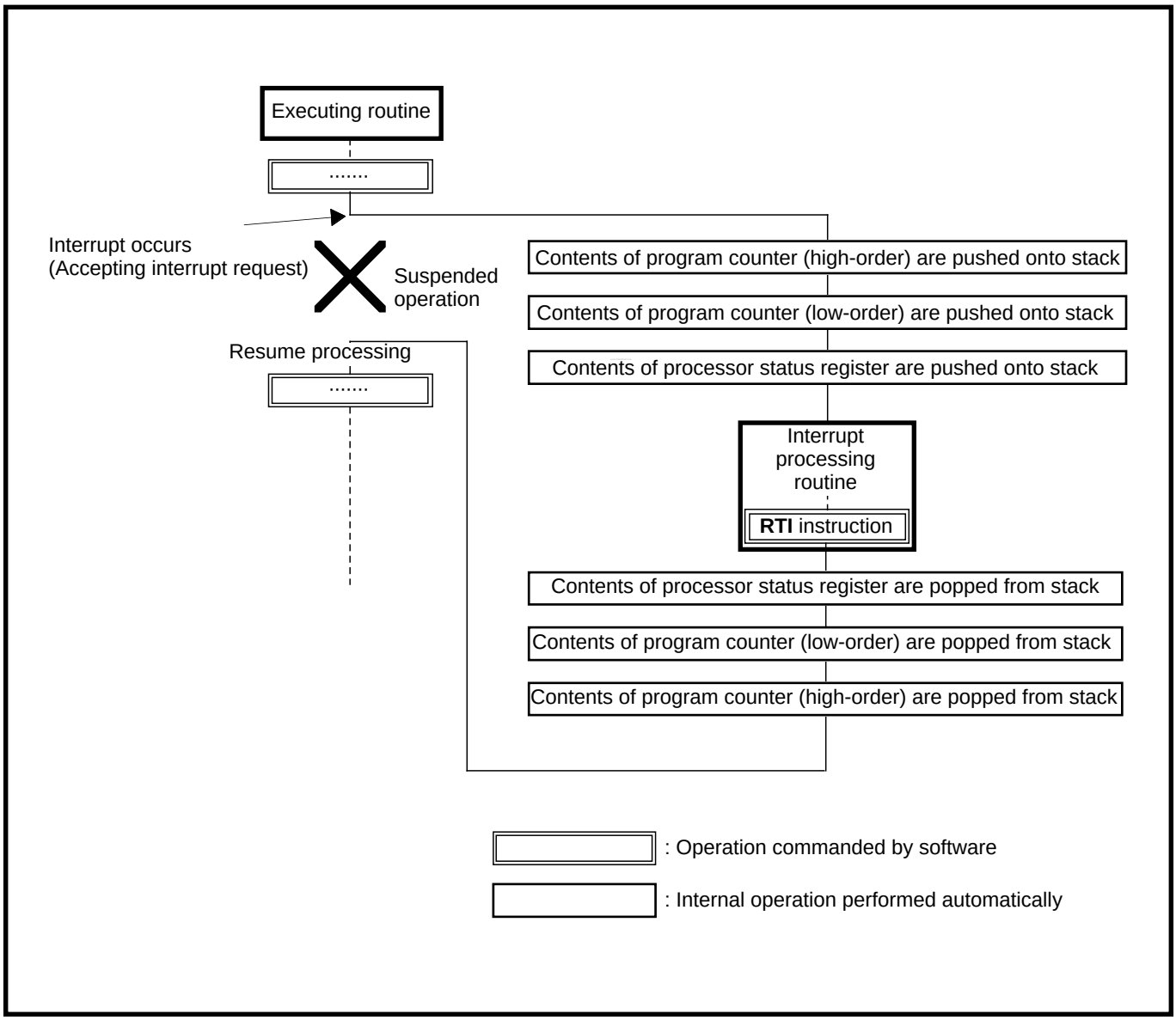


Fig. 2.2.7 Interrupt operation diagram

2.2 Interrupt

(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 2.2.8 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.

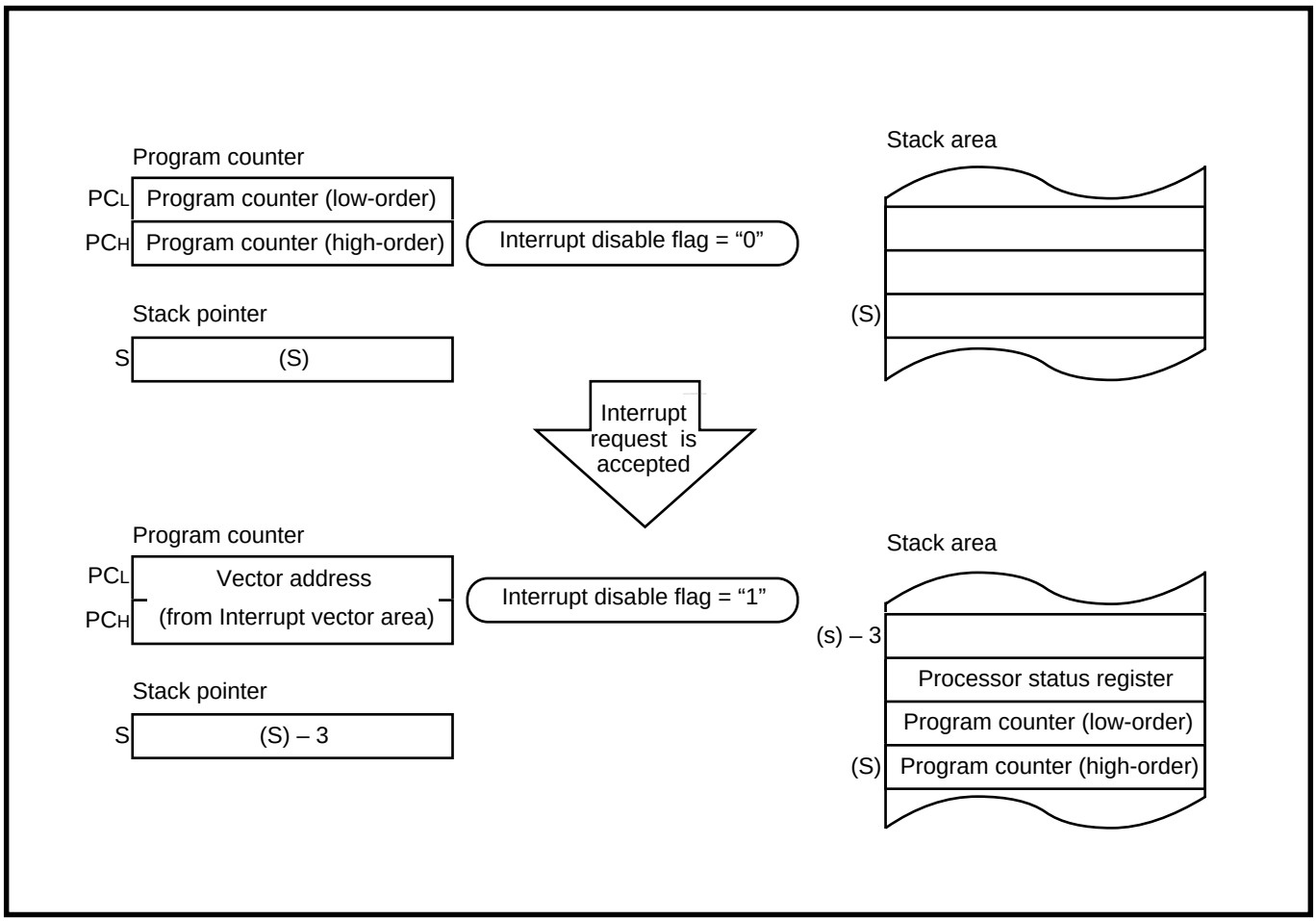


Fig. 2.2.8 Changes of stack pointer and program counter upon acceptance of interrupt request

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

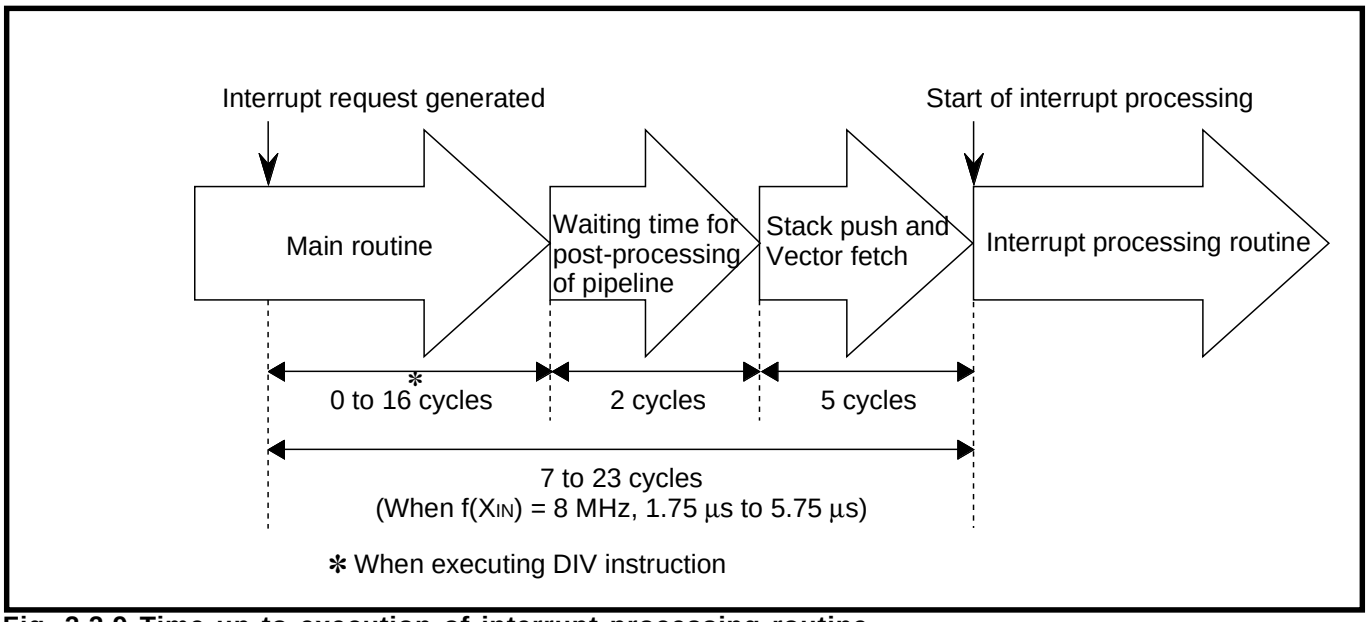


Fig. 2.2.9 Time up to execution of interrupt processing routine

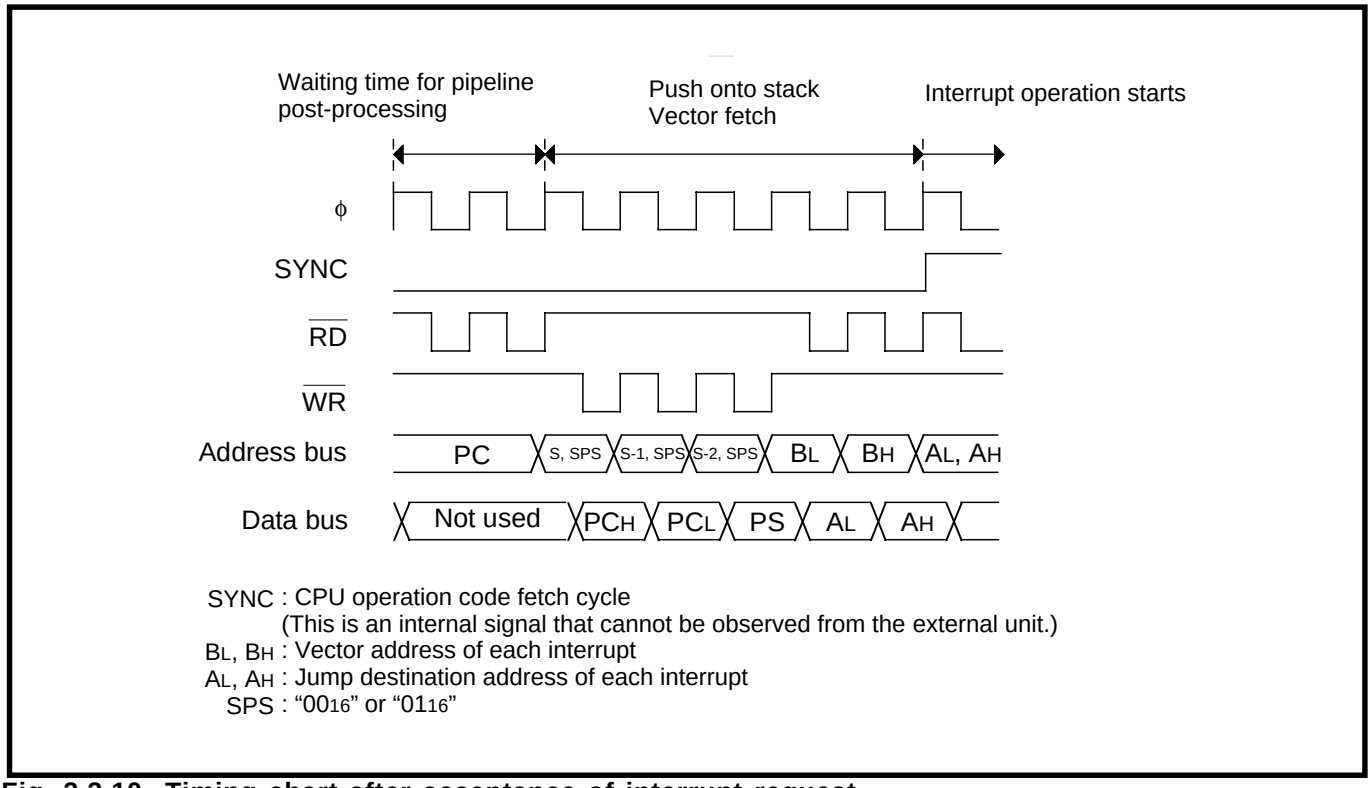


Fig. 2.2.10 Timing chart after acceptance of interrupt request

2.2 Interrupt

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 2.2.11 shows an interrupt control diagram.

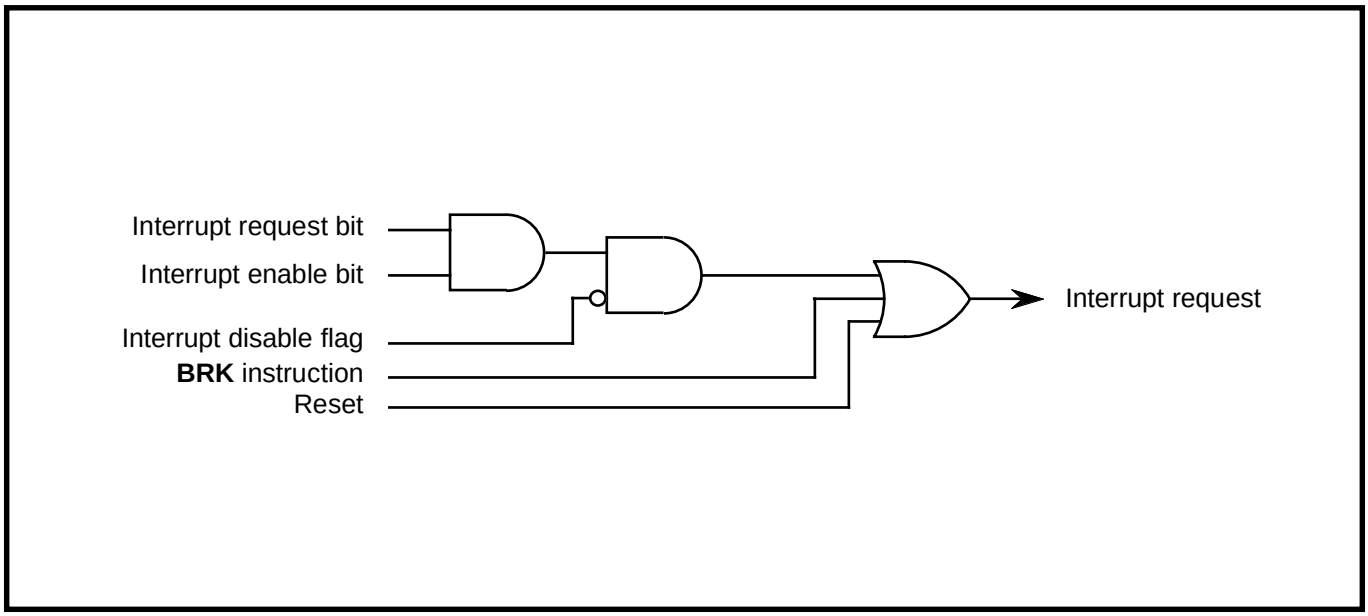


Fig. 2.2.11 Interrupt control diagram

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. Table 2.2.2 shows a 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 $3C_{16}$) and interrupt request register 2 (address $3D_{16}$).

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 $3F_{16}$).

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 2.2.12 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	$003E_{16}$	b0	$003C_{16}$	b0
INT_1	$003E_{16}$	b2	$003C_{16}$	b2
INT_2	$003E_{16}$	b3	$003C_{16}$	b3
INT_3/Serial I/O2	$003E_{16}$	b4	$003C_{16}$	b4
Timer X	$003E_{16}$	b6	$003C_{16}$	b6
Timer Y	$003E_{16}$	b7	$003C_{16}$	b7
Timer 1	$003F_{16}$	b0	$003D_{16}$	b0
Timer 2	$003F_{16}$	b1	$003D_{16}$	b1
Serial I/O1 reception	$003F_{16}$	b2	$003D_{16}$	b2
Serial I/O1 transmission	$003F_{16}$	b3	$003D_{16}$	b3
$CNTR_0$	$003F_{16}$	b4	$003D_{16}$	b4
$CNTR_1$	$003F_{16}$	b5	$003D_{16}$	b5
A-D converter	$003F_{16}$	b6	$003D_{16}$	b6

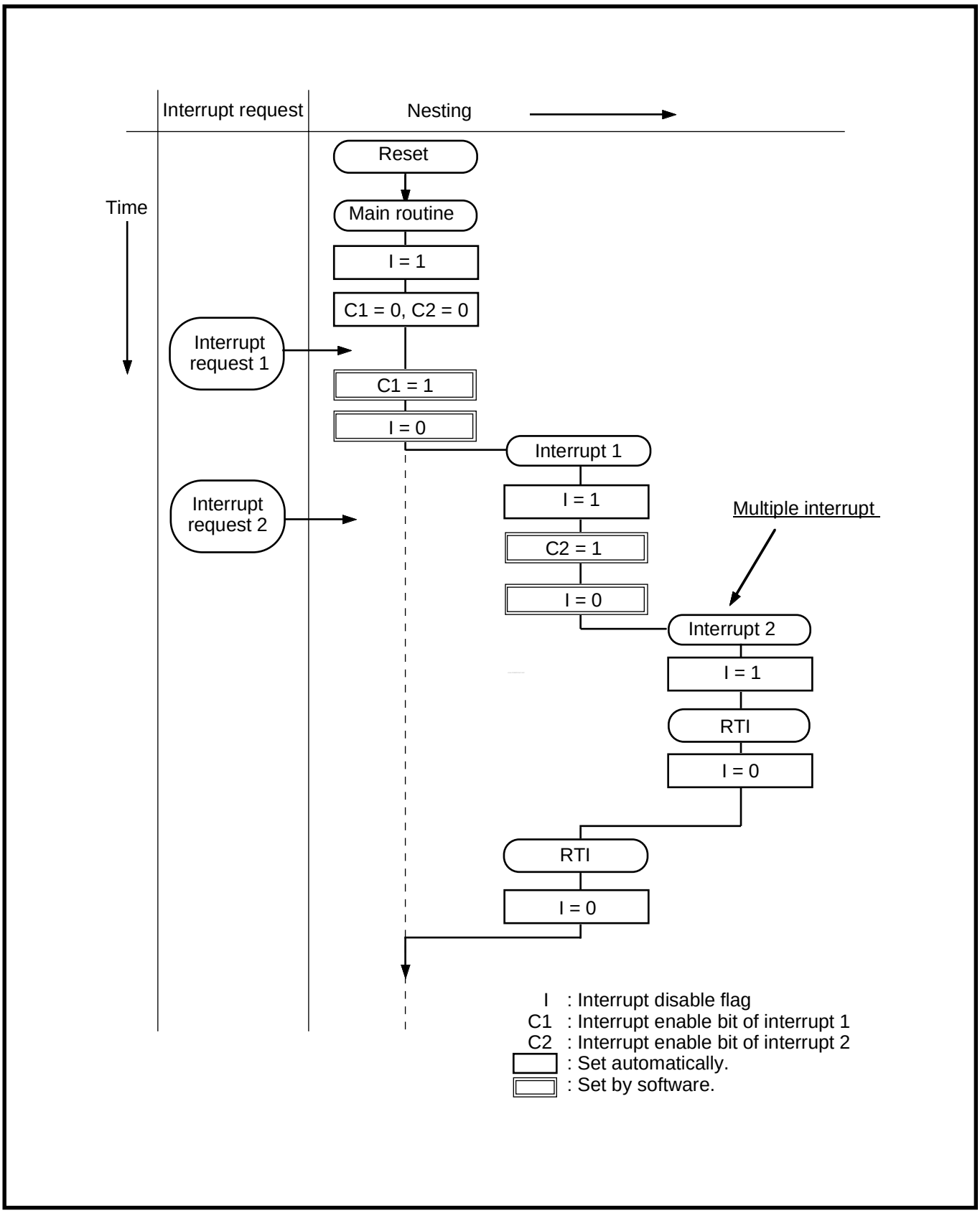


Fig. 2.2.12 Example of multiple interrupts

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_3).

(1) Active edge selection

INT_0–INT_3 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_3 interrupt source selection

Which of interrupt source of the serial I/O2/INT_3 interrupt source can be selected by the serial I/O2/INT_3 interrupt source bit (bit 4 of address $3A_{16}$). (Set this bit to “0” when using INT_3.)

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 $3A_{16}$)

•Timer XY mode register (address 23_{16})

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

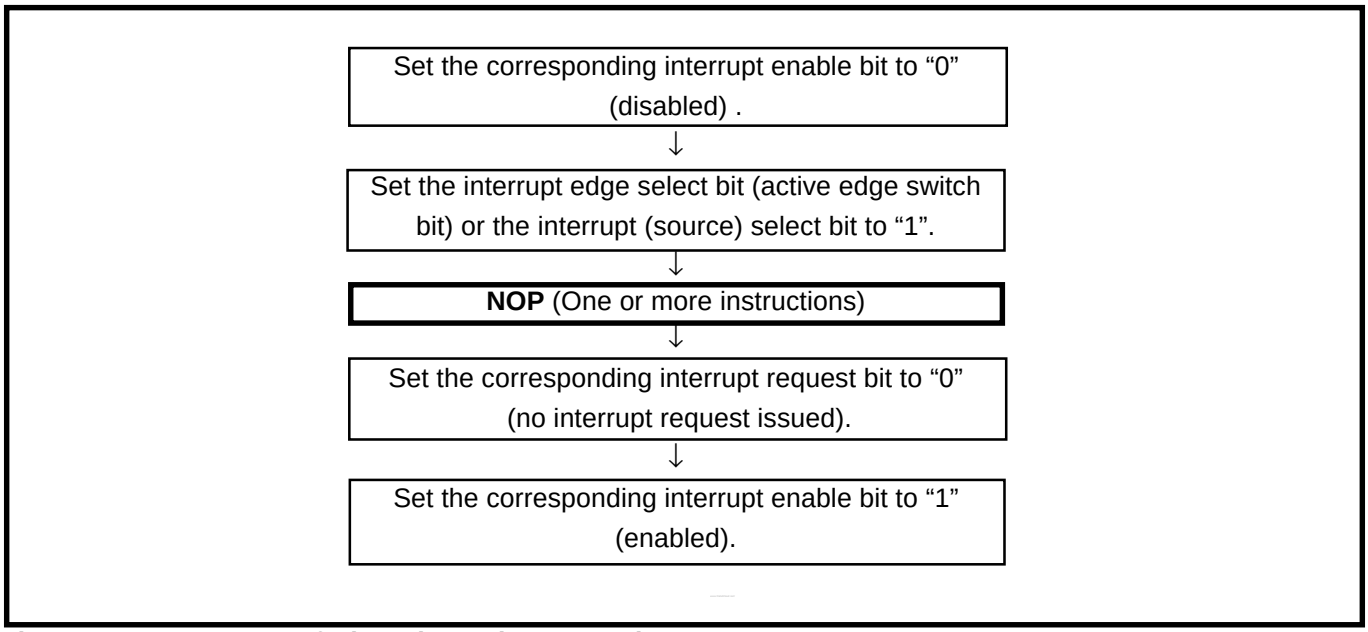


Fig. 2.2.13 Sequence of changing relevant register

■ **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 $3A_{16}$)
Timer XY mode register (address 23_{16})

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

Concerned register: Interrupt edge selection register (address $3A_{16}$)

(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” by using a data transfer instruction, execute one or more instructions before executing the **BBC** or **BBS** instruction.

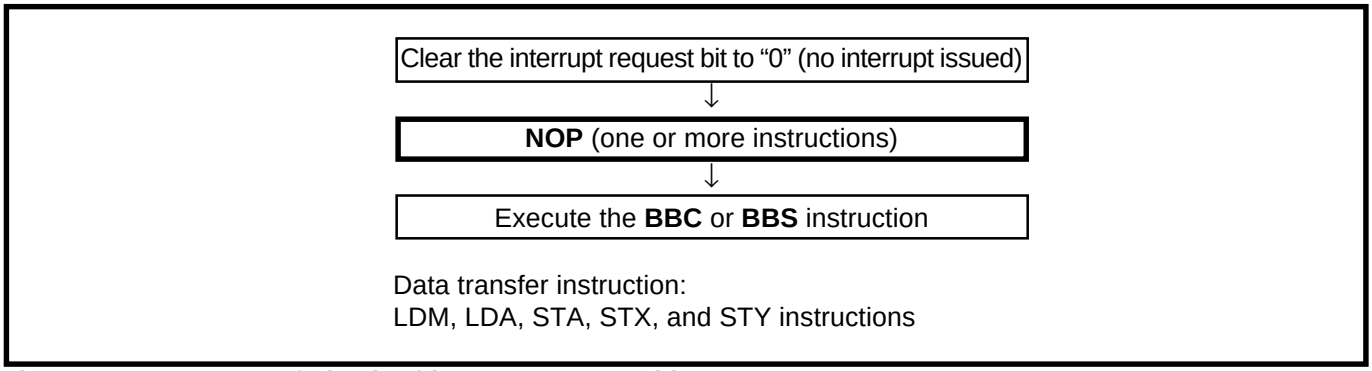


Fig. 2.2.14 Sequence of check of interrupt request bit

■ **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

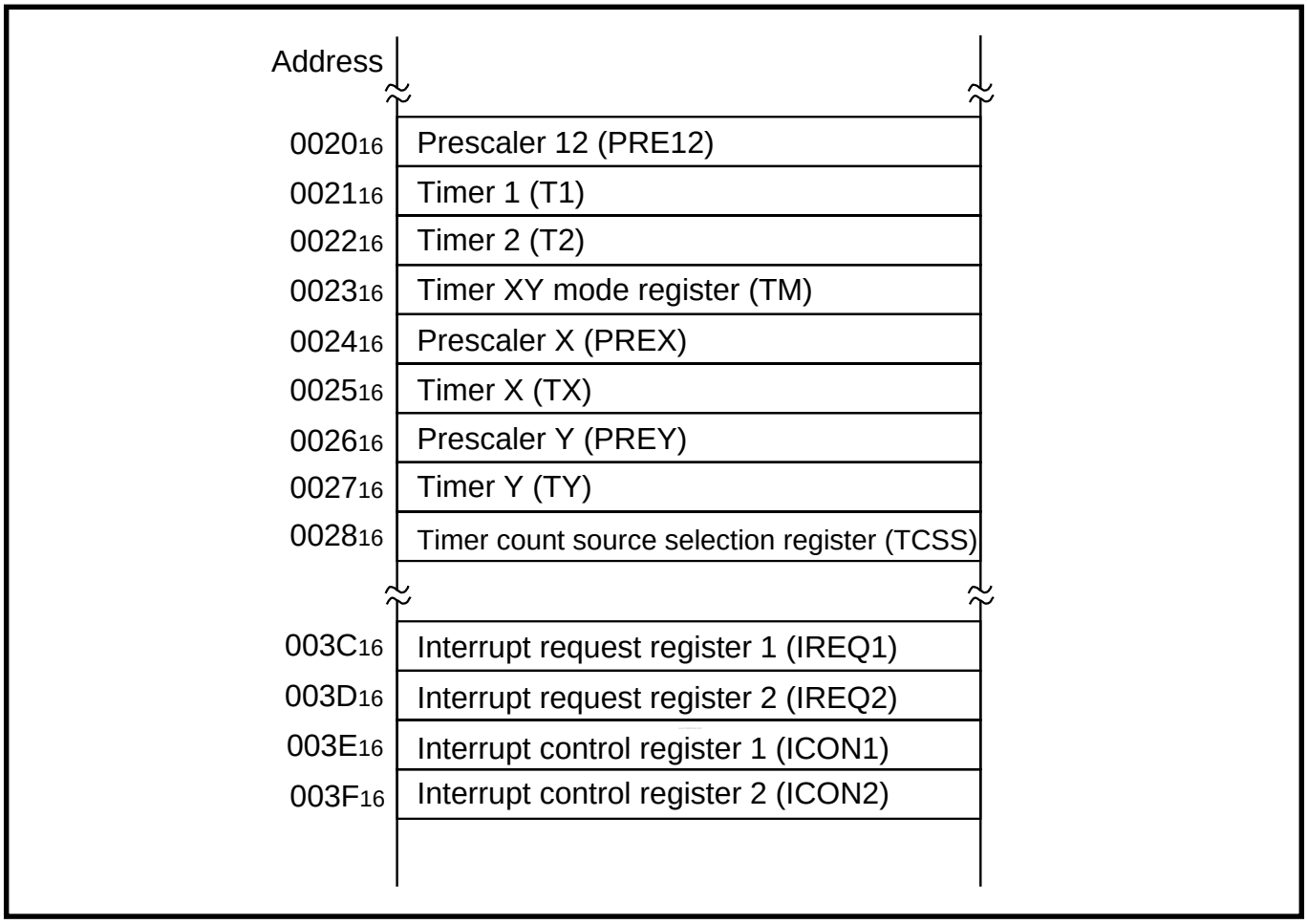


Fig. 2.3.1 Memory map of registers relevant to timers

2.3.2 Relevant registers

Prescaler 12, Prescaler X, Prescaler Y

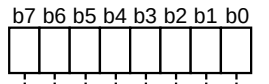
b7 b6 b5 b4 b3 b2 b1 b0

Prescaler 12 (PRE12), Prescaler X (PREX), Prescaler Y (PREY)
(addresses 20_{16}, 24_{16}, 26_{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

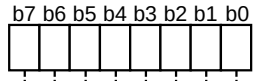


Timer 1
(T1: address 21_{16})

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



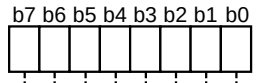
Timer 2
(T2: address 22_{16})

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

Fig. 2.3.4 Structure of Timer 2

2.3 Timer

Timer X, Timer Y

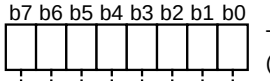


Timer X, Timer Y
(TX, TY: addresses 25_{16}, 27_{16})

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. 2.3.5 Structure of Timer X, Timer Y

Timer XY mode register



Timer XY mode register
(TM: address 23_{16})

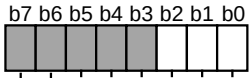
b	Name	Functions	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0: Timer mode 0 1: Pulse output mode 1 0: Event counter mode 1 1: Pulse width measurement mode	0	○	○
1			0	○	○
2	$CNTR_0$ active edge switch bit	Refer to Table 2.3.1	0	○	○
3	Timer X count stop bit	0: Count start 1: Count stop	0	○	○
4	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	$CNTR_1$ 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 $CNTR_0$/$CNTR_1$ active edge switch bit function

Timer X /Timer Y operation modes	Set value	Timer function	$CNTR_0$ / $CNTR_1$ interrupt request occurrence source
Timer mode	"0"	No influence to timer count	$CNTR_0$/$CNTR_1$ input signal falling edge
	"1"	No influence to timer count	$CNTR_0$/$CNTR_1$ input signal rising edge
Pulse output mode	"0"	Pulse output start: Beginning at "H" level	Output signal falling edge count
	"1"	Pulse output start: Beginning at "L" level	Output signal rising edge count
Event counter mode	"0"	Rising edge count	Input signal falling edge count
	"1"	Falling edge count	Input signal rising edge count
Pulse width measurement mode	"0"	"H" level width measurement	Input signal falling edge count
	"1"	"L" level width measurement	Input signal rising edge count

Timer count source selection register

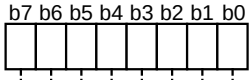


Timer count source selection register
(TCSS: address 28_{16})

b	Name	Functions	At reset	R	W
0	Timer X count source selection bit	0: $f(X_{IN})/16$ ($f(X_{CIN})/16$ at low-speed mode) 1: $f(X_{IN})/2$ ($f(X_{CIN})/2$ at low-speed mode)	0	○	○
1	Timer Y count source selection bit	0: $f(X_{IN})/16$ ($f(X_{CIN})/16$ at low-speed mode) 1: $f(X_{IN})/2$ ($f(X_{CIN})/2$ at low-speed mode)	0	○	○
2	Timer 12 count source selection bit	0: $f(X_{IN})/16$ ($f(X_{CIN})/16$ at low-speed mode) 1: $f(X_{CIN})$	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			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 2.3.7 Structure of Timer count source selection register

Interrupt request register 1



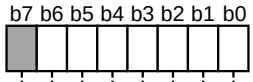
Interrupt request register 1
(IREQ1 : address $3C_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	When writing to this bit, set “0” to this bit.		0	○	✻
2	INT_1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT_2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_3/Serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	When writing to this bit, set “0” to this bit.		0	○	✻
6	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer Y 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. 2.3.8 Structure of Interrupt request register 1

Interrupt request register 2



Interrupt request register 2
(IREQ2 : address $3D_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	Timer 2 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 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	$CNTR_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	$CNTR_1$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	A-D converter 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. 2.3.9 Structure of Interrupt request register 2

Interrupt control register 1

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	

Interrupt control register 1
(ICON1 : address $3E_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Fix this bit to “0”.		0	○	○
2	INT_1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_2 nterrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_3/Serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Fix this bit to “0”.		0	○	○
6	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.3.10 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 $3F_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Timer 2 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 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	$CNTR_0$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	$CNTR_1$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	A-D converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.3.11 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 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 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)

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)

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)

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.

(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 X.
•The clock is counted up in the process routine of the timer X interrupt which occurs at 250 ms intervals.

Figure 2.3.12 shows the timers connection and setting of division ratios; Figure 2.3.13 shows the relevant registers setting; Figure 2.3.14 shows the control procedure.

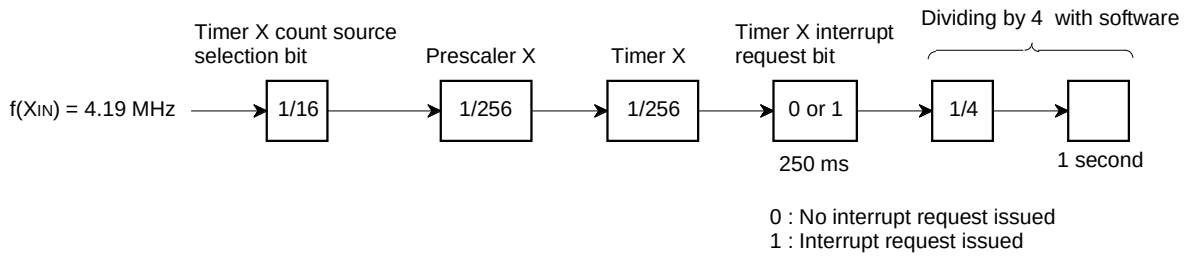


Fig. 2.3.12 Timers connection and setting of division ratios

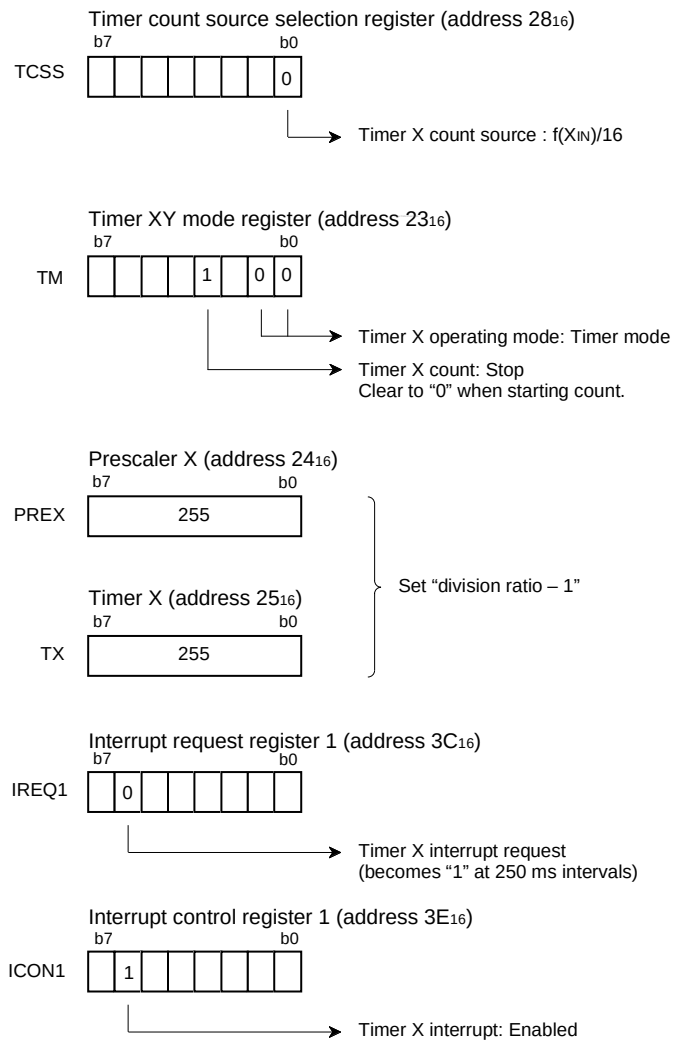


Fig. 2.3.13 Relevant registers setting

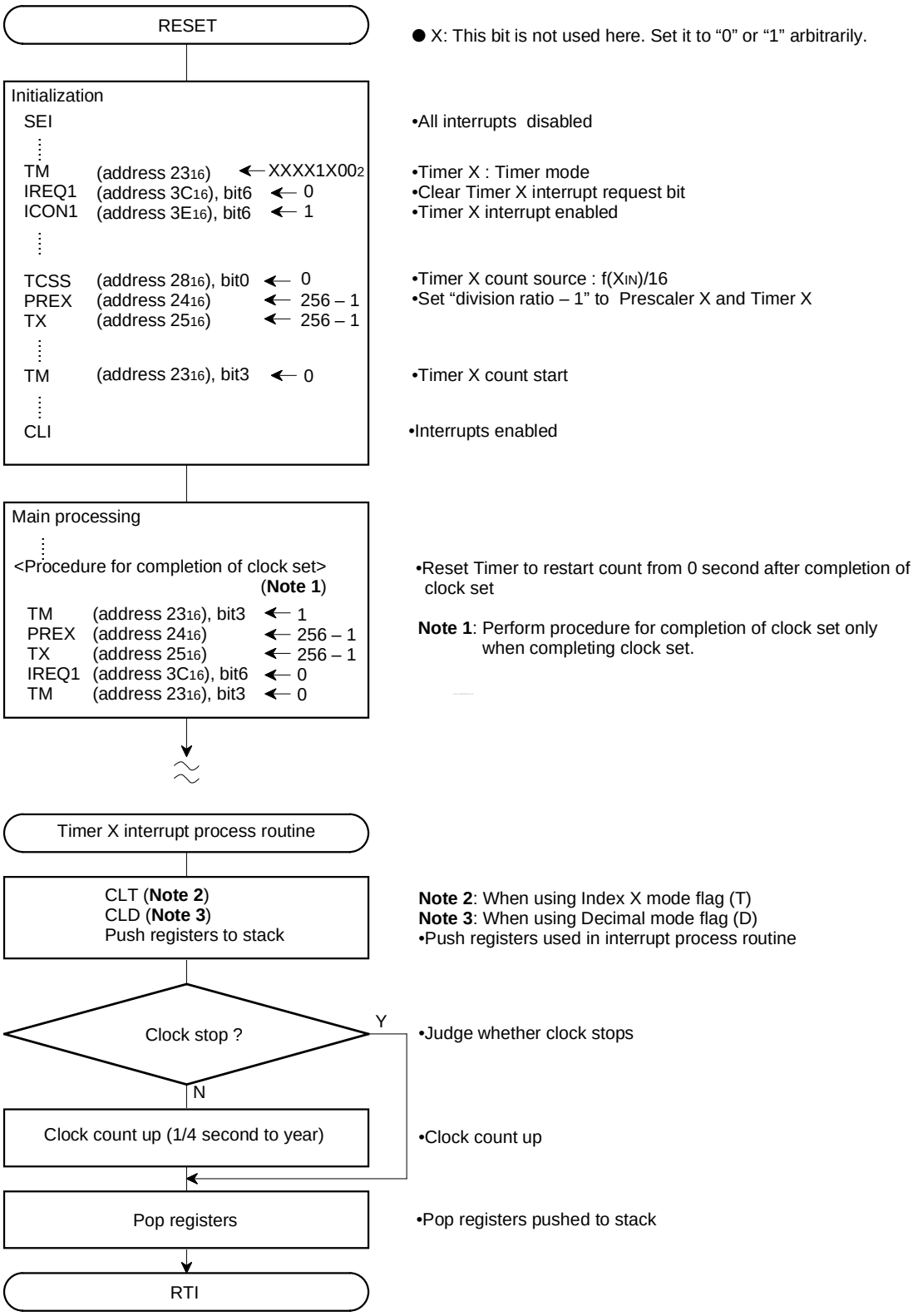


Fig. 2.3.14 Control procedure

(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 $f(X_{IN})$ = 4.19 MHz (2^{22} Hz) into about 2 kHz (2048 Hz), is output from the $P2_7/CNTR_0$ pin.
•The level of the $P2_7/CNTR_0$ pin is fixed to "H" while a piezoelectric buzzer output stops.

Figure 2.3.15 shows a peripheral circuit example, and Figure 2.3.16 shows the timers connection and setting of division ratios. Figure 2.3.17 shows the relevant registers setting, and Figure 2.3.18 shows the control procedure.

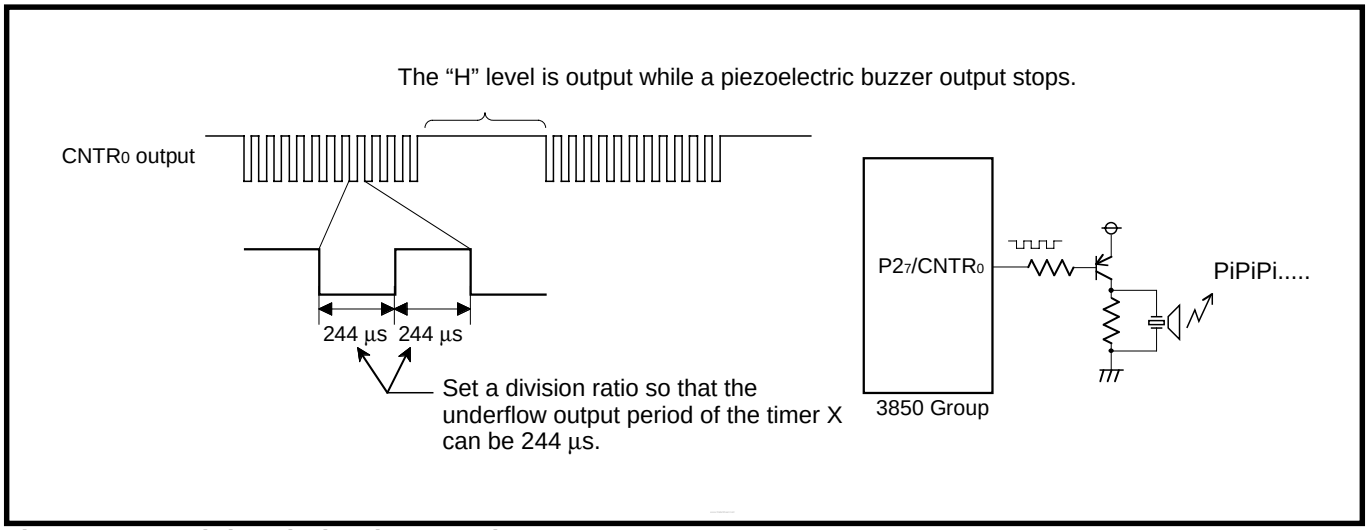


Fig. 2.3.15 Peripheral circuit example

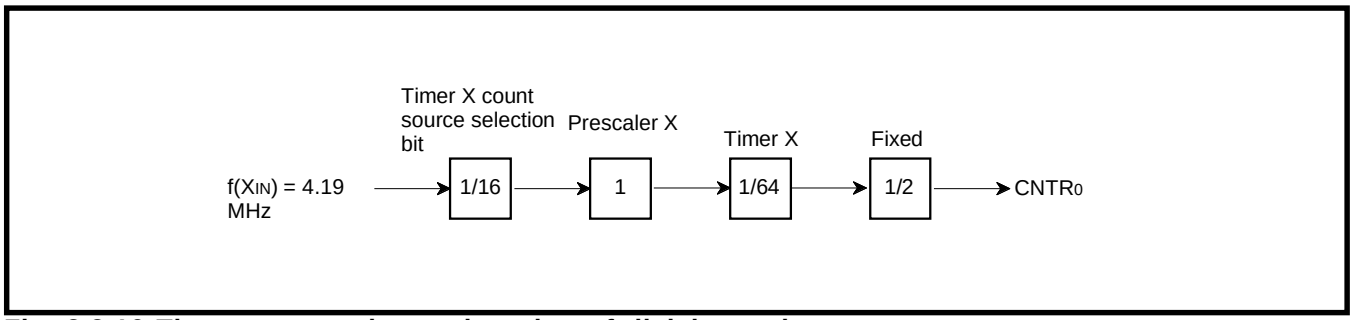


Fig. 2.3.16 Timers connection and setting of division ratios

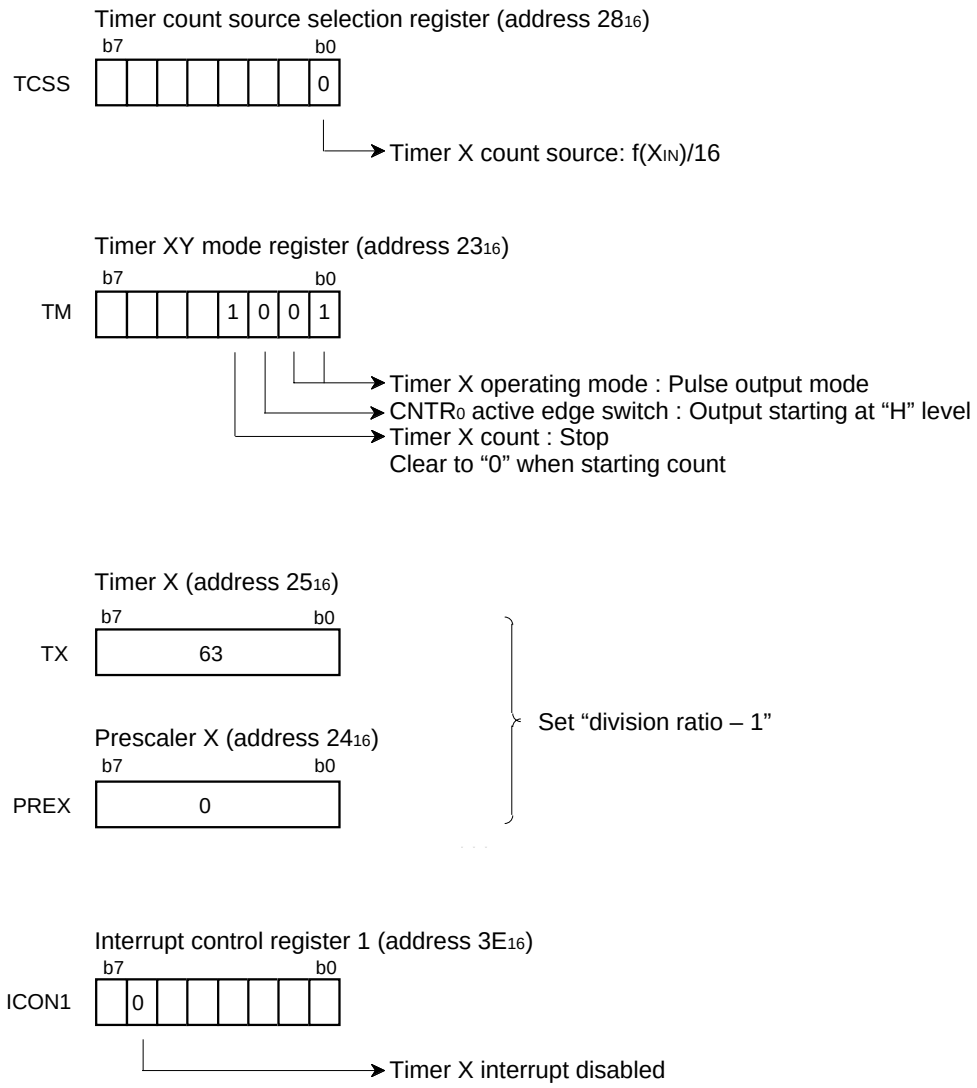


Fig. 2.3.17 Relevant registers setting

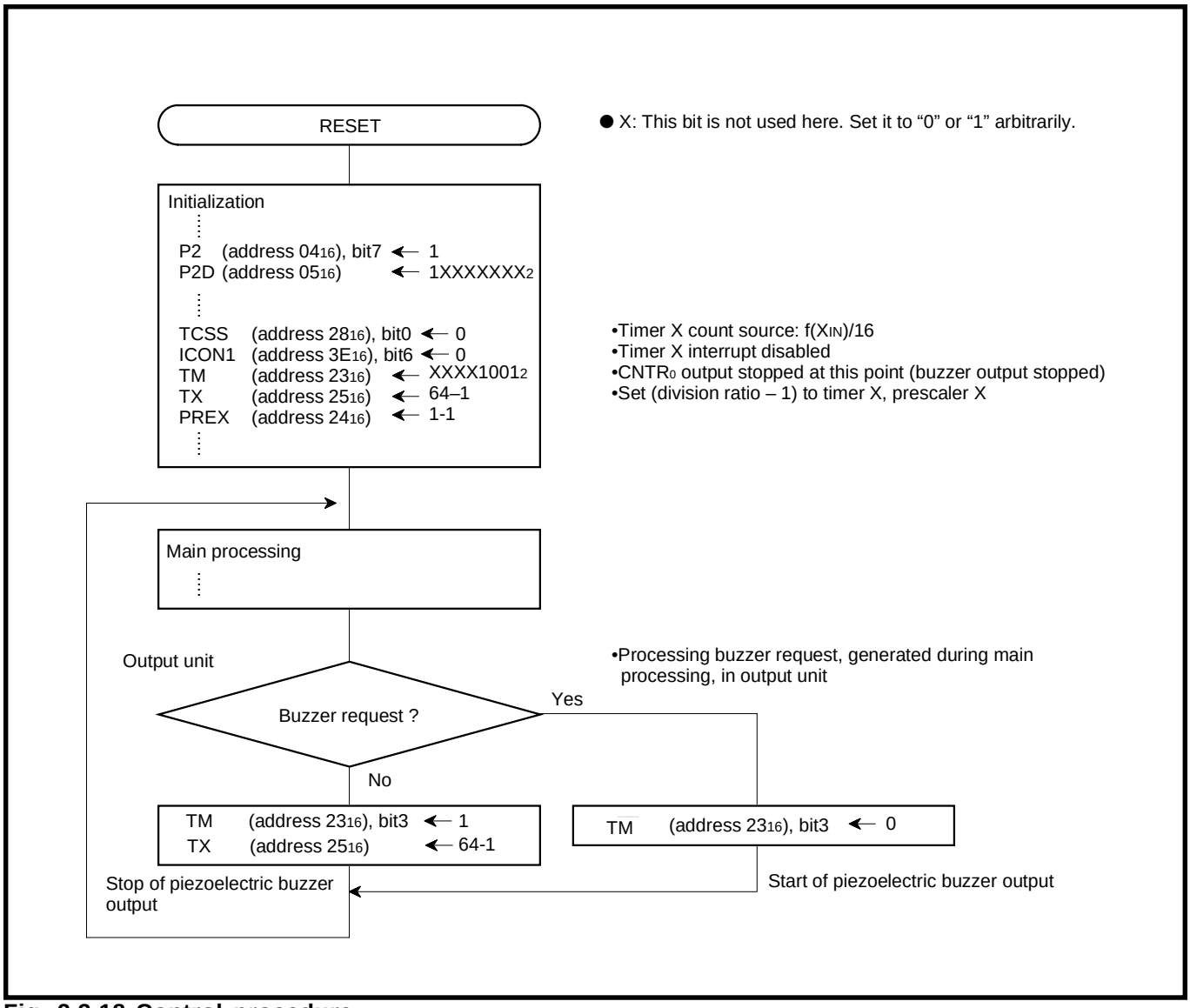


Fig. 2.3.18 Control procedure

(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 $P4_0/CNTR_1$ pin with the timer.
- A reference value

Specifications:
- Clock $f(X_{IN})$ = 4.19 MHz (2^{22} Hz)
- The pulse is input to the $P4_0/CNTR_1$ pin and counted by the timer Y.
- 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 value.

Figure 2.3.19 shows the judgment method of valid/invalid of input pulses; Figure 2.3.20 shows the relevant registers setting; Figure 2.3.21 shows the control procedure.

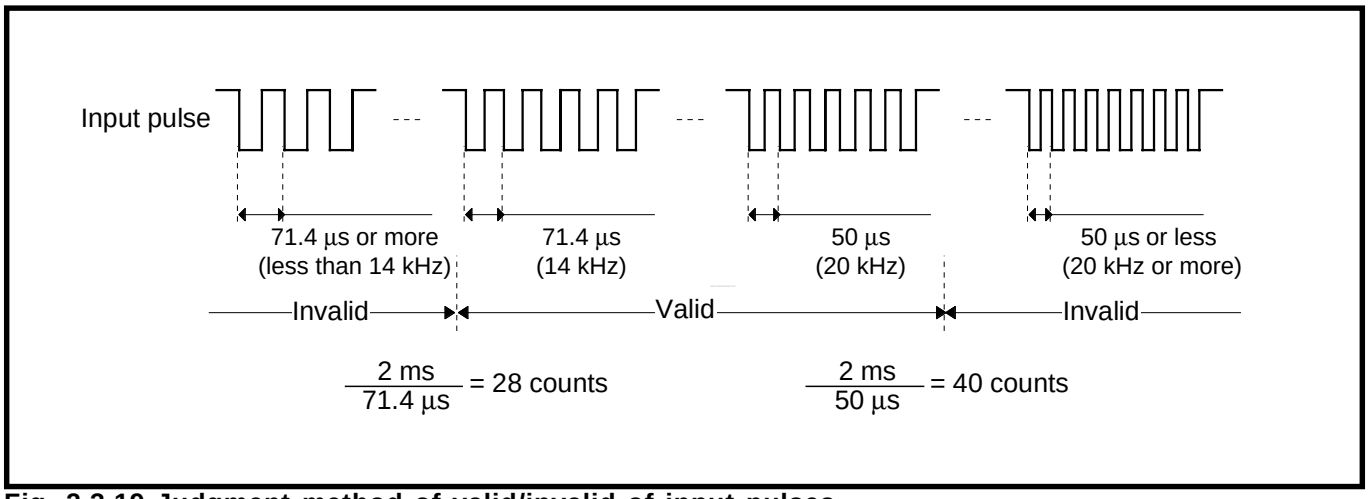


Fig. 2.3.19 Judgment method of valid/invalid of input pulses

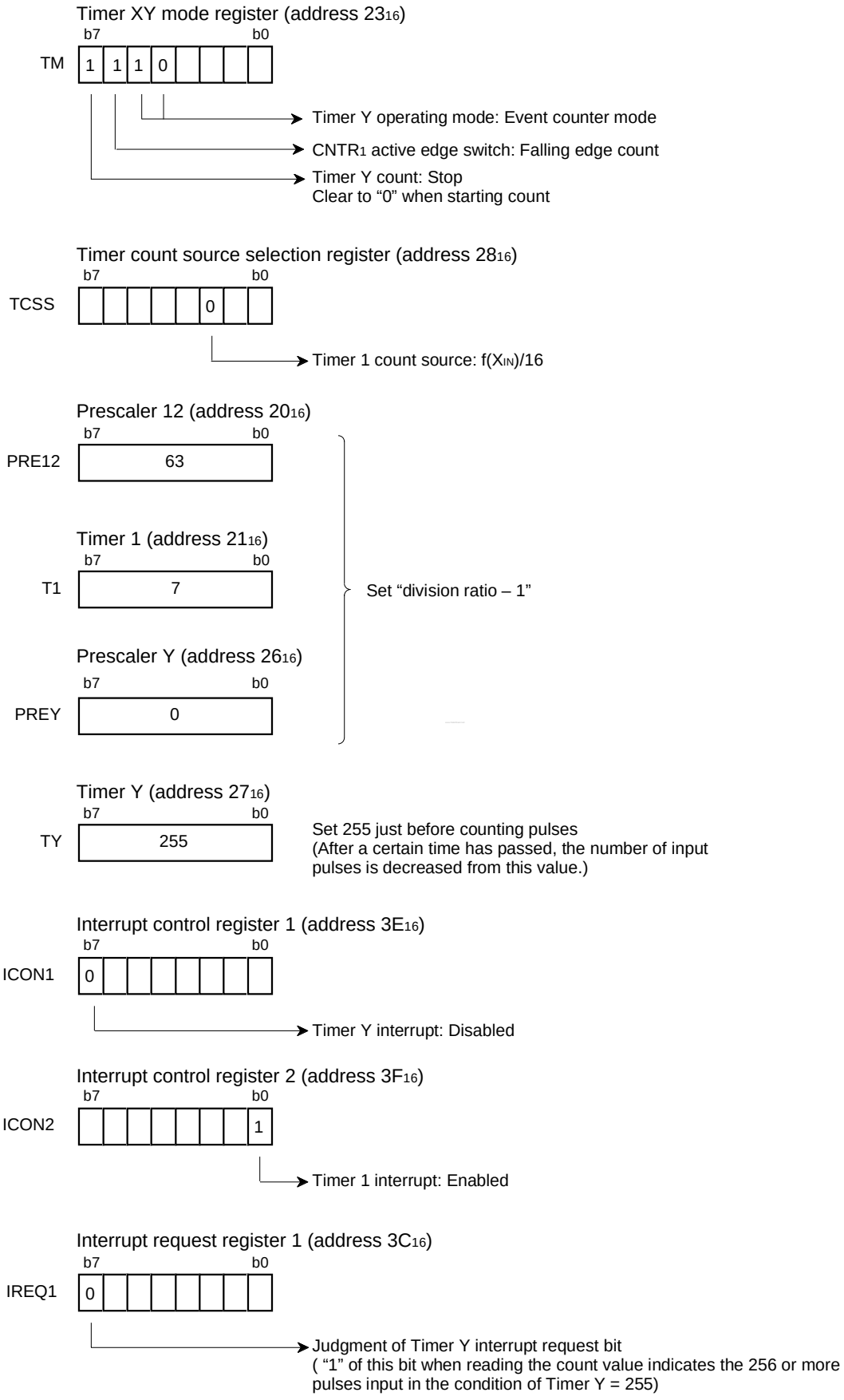


Fig. 2.3.20 Relevant registers setting

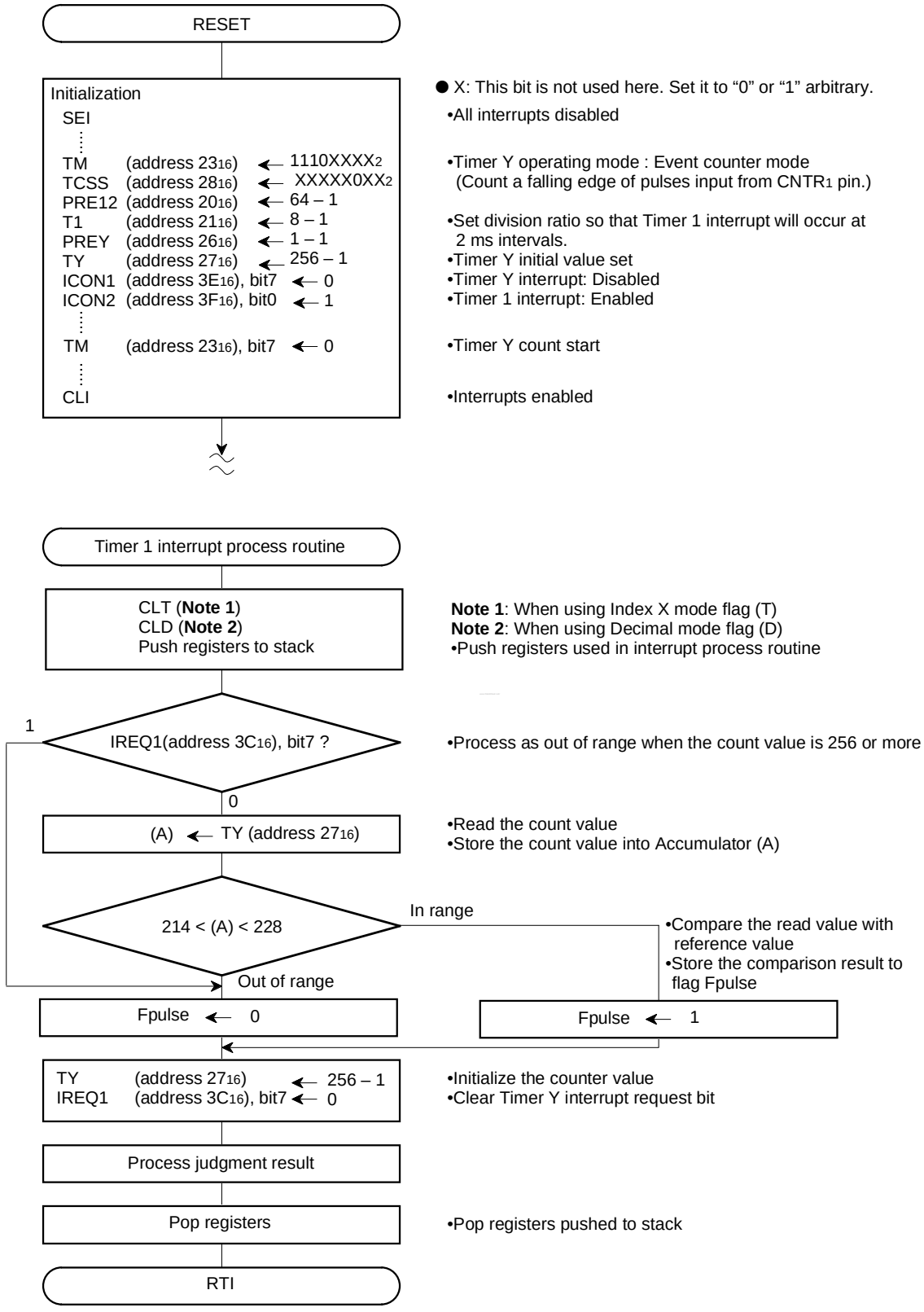


Fig. 2.3.21 Control procedure

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

Outline: The timer X counts the "H" level width of the pulses input to the $CNTR_0$ pin. An underflow is detected by the timer X interrupt and an end of the input pulse "H" level is detected by the $CNTR_0$ interrupt.

Specifications: •The timer X counts the "H" level width of the FG pulse input to the $CNTR_0$ pin.

<Example>

When the clock frequency is 4.19 MHz, the count source is 3.8 μs, which is obtained by dividing the clock frequency by 16. Measurement can be made up to 250 ms in the range of $FFFF_{16}$ to 0000_{16}.

Figure 2.3.22 shows the timers connection and setting of division ratio; Figure 2.3.23 shows the relevant registers setting; Figure 2.3.24 shows the control procedure.

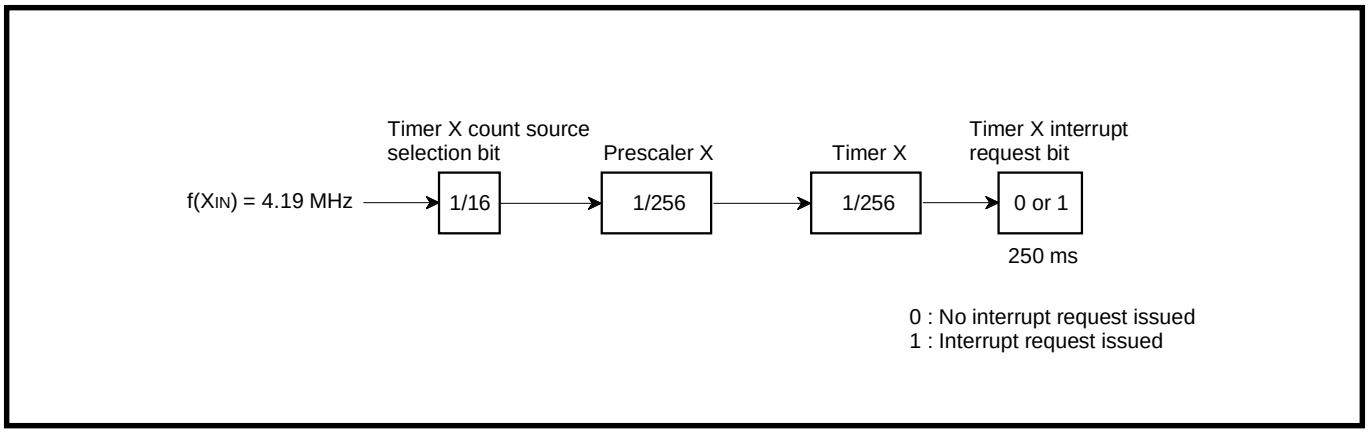


Fig. 2.3.22 Timers connection and setting of division ratios

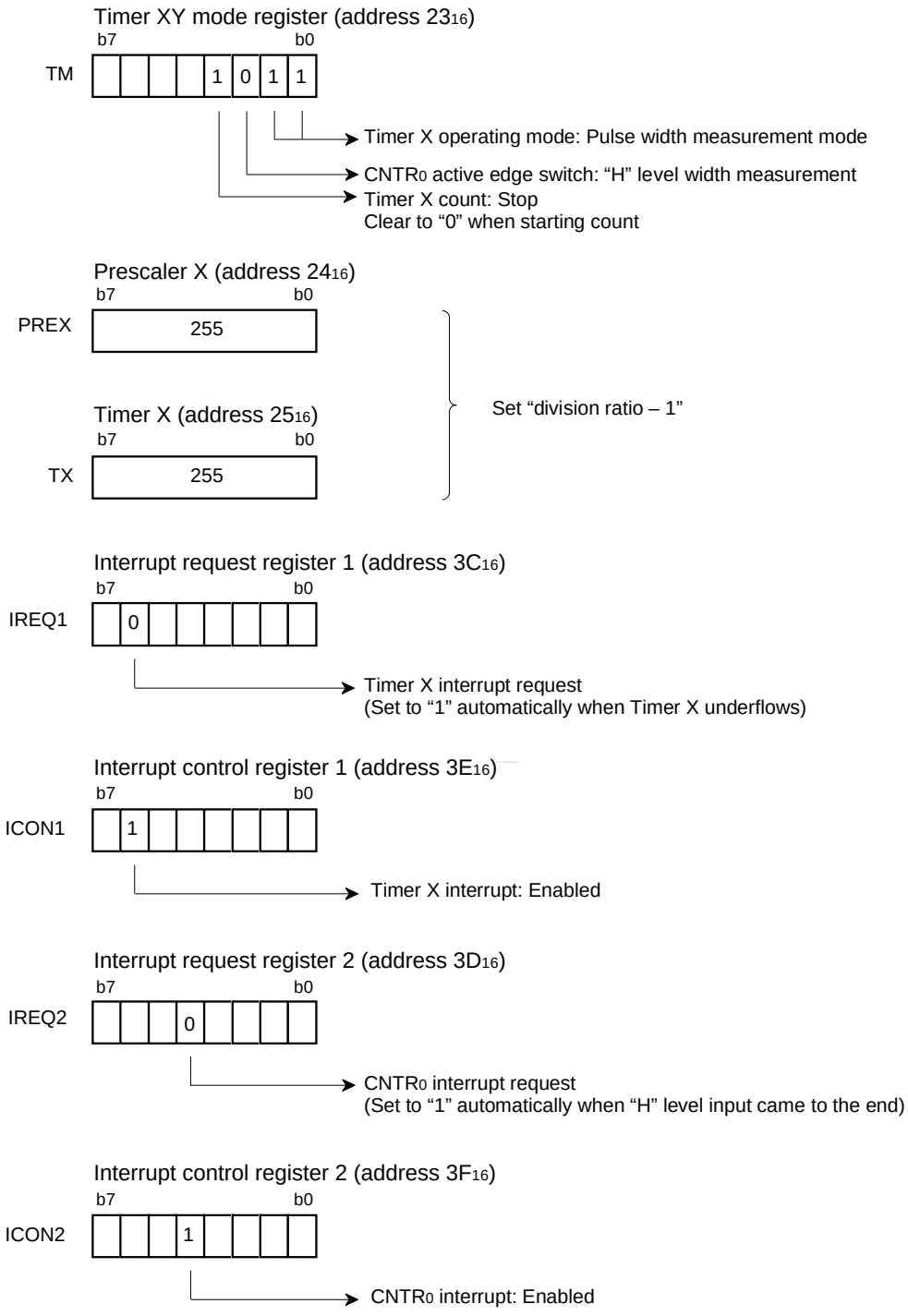


Fig. 2.3.23 Relevant registers setting

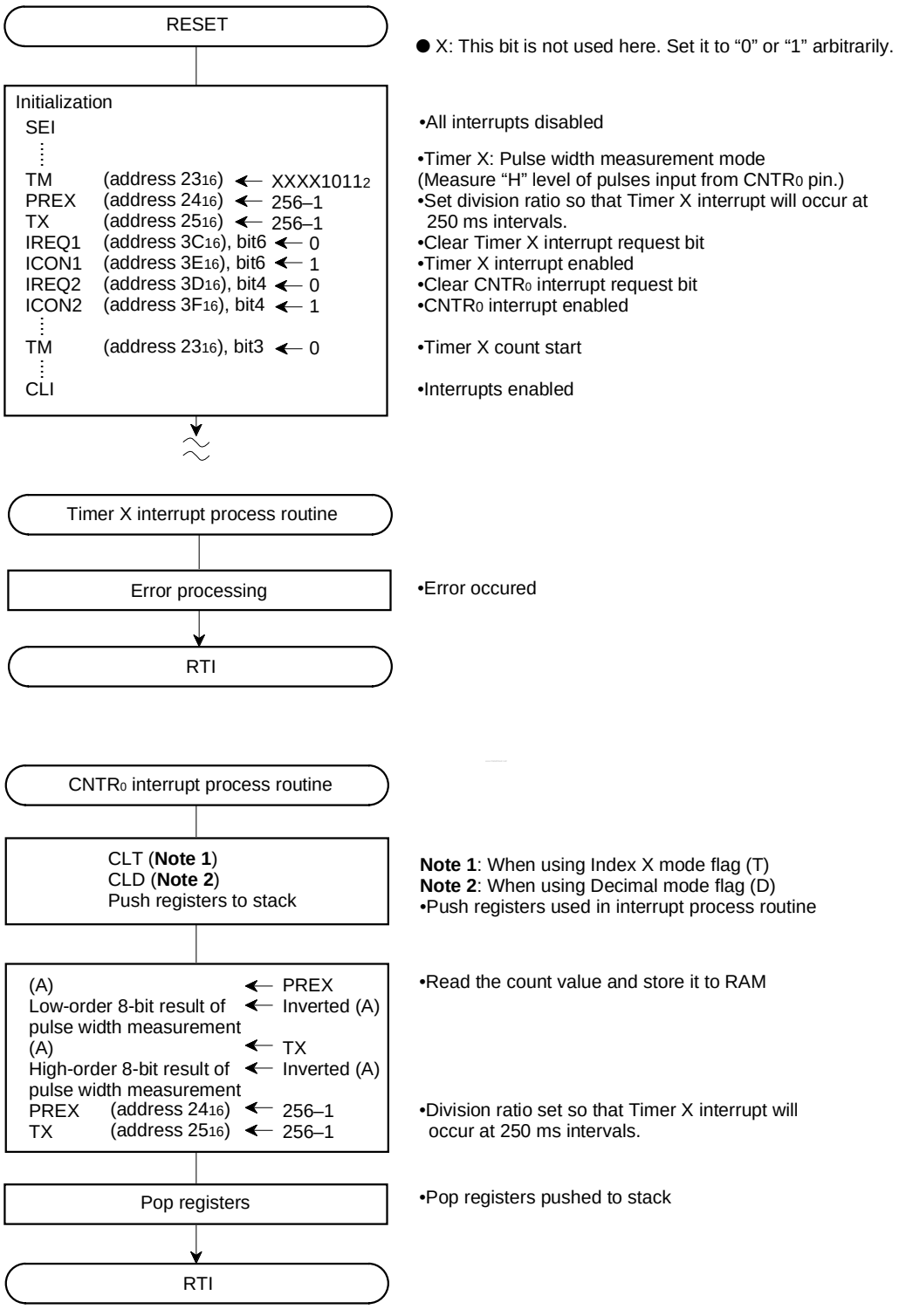


Fig. 2.3.24 Control procedure

2.3.4 Notes on timer

- 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.

2.4 Serial I/O

This paragraph explains the registers setting method and the notes relevant to the Serial I/O.

2.4.1 Memory map

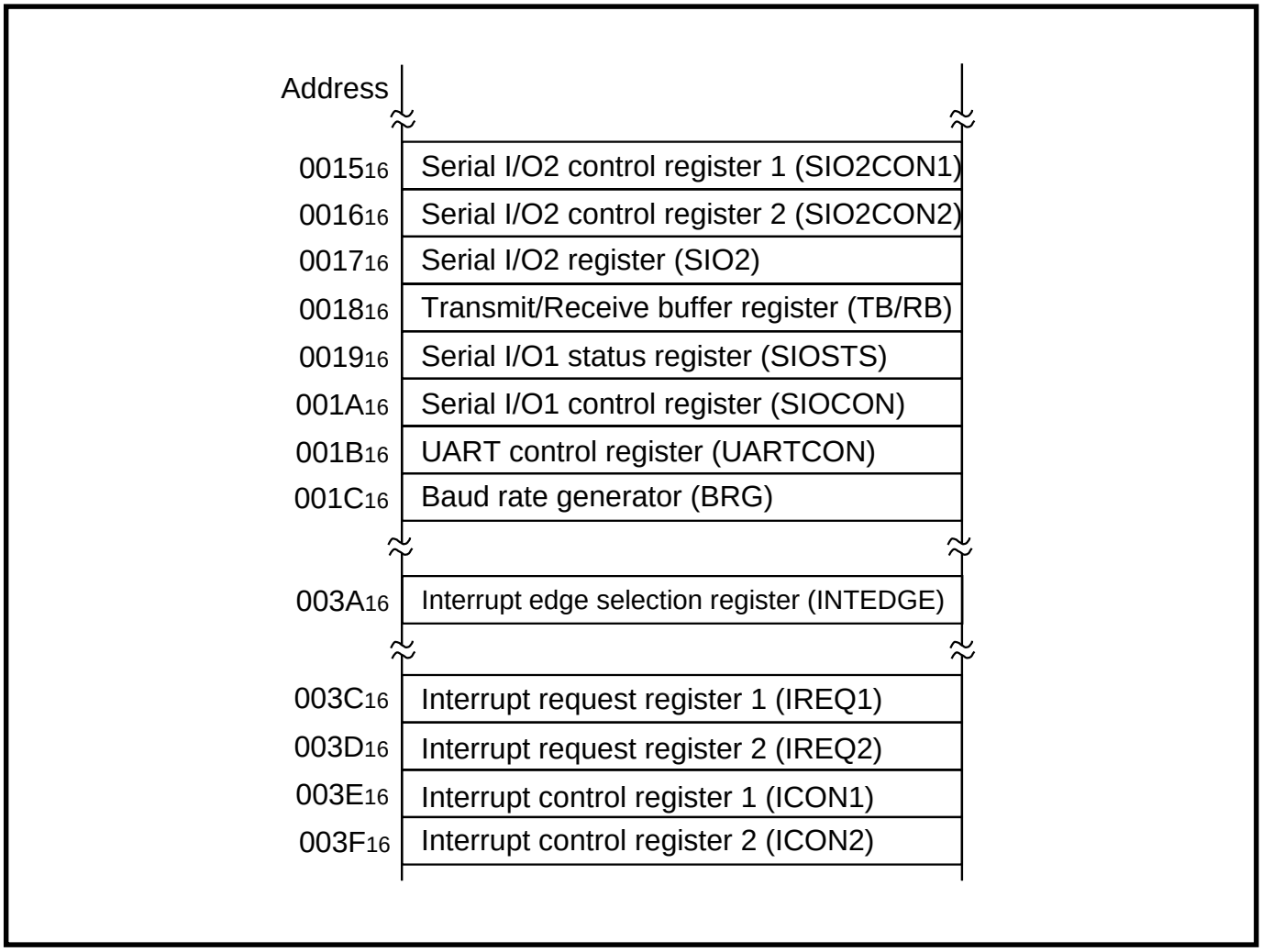


Fig. 2.4.1 Memory map of registers relevant to Serial I/O

2.4.2 Relevant registers

Serial I/O2 control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O2 control register 1 (SIO2CON1: address 15_{16})

b	Name	Functions	At reset	R	W
0	Internal synchronous clock selection bits	b2b1b0 0 0 0: f(X_{IN})/8 0 0 1: f(X_{IN})/16 0 1 0: f(X_{IN})/32 0 1 1: f(X_{IN})/64 1 1 0: f(X_{IN})/128 1 1 1: f(X_{IN})/256	0	○	○
1			0	○	○
2			0	○	○
3	Serial I/O2 port selection bit	0: I/O port ($P0_1$, $P0_2$) 1: S_{OUT2}, S_{CLK2} signal output	0	○	○
4	$\overline{S_{RDY2}}$ output enable bit	0: I/O port ($P0_3$) 1: $\overline{S_{RDY2}}$ 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	$P0_1/S_{OUT2}$, $P0_2/S_{CLK2}$ P-channel output disable bit	0: CMOS output 1: N-channel open-drain output	0	○	○

Fig. 2.4.2 Structure of Serial I/O2 control register 1

Serial I/O2 control register 2

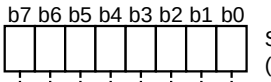
b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O2 control register 2 (SIO2CON2: address 16_{16})

b	Name	Functions	At reset	R	W
0	Optional transfer bits	b2b1b0 0 0 0: 1 bit 0 0 1: 2 bit 0 1 0: 3 bit 0 1 1: 4 bit 1 0 0: 5 bit 1 0 1: 6 bit 1 1 0: 7 bit 1 1 1: 8 bit	1	○	○
1			1	○	○
2			1	○	○
3	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
4			0	○	×
5			0	○	×
6	Serial I/O2 I/O comparison signal control bit	0: $P4_3$ I/O 1: S_{CMP2} output	0	○	○
7	S_{OUT2} pin control bit ($P0_1$)	0: Output active 1: Output high-impedance	0	○	○

Fig. 2.4.3 Structure of Serial I/O2 control register 2

Serial I/O2 register

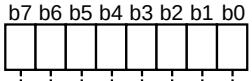


Serial I/O2 register
(SIO2: address 17_{16})

b	Name	Functions	At reset	R	W
0		This register becomes shift register.	Undefined	○	○
1		At transmit: Set transmit data to this register.	Undefined	○	○
2		At receive: Received data is stored to this register.	Undefined	○	○
3			Undefined	○	○
4			Undefined	○	○
5			Undefined	○	○
6			Undefined	○	○
7			Undefined	○	○

Fig. 2.4.4 Structure of Serial I/O2 register

Transmit/Receive buffer register



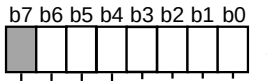
Transmit/Receive buffer register (TB/RB: address 18_{16})

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. 2.4.5 Structure of Transmit/Receive buffer register

Serial I/O1 status register



Serial I/O1 status register (SIOSTS: address 19_{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. 2.4.6 Structure of Serial I/O1 status register

Serial I/O1 control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O1 control register
(SIOCON: address $1A_{16}$)

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/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 ($P2_7$) 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 ($P2_4$ to $P2_7$: normal I/O pins) 1: Serial I/O1 enabled ($P2_4$ to $P2_7$: Serial I/O pins)	0	○	○

Fig. 2.4.7 Structure of Serial I/O1 control register

UART control register

b7 b6 b5 b4 b3 b2 b1 b0

UART control register
(UARTCON: address $1B_{16}$)

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	$P2_5$/TxD P-channel output disable bit (POFF)	In output mode 0: CMOS output 1: N-channel open-drain output	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. 2.4.8 Structure of UART control register

Baud rate generator

b7 b6 b5 b4 b3 b2 b1 b0

Baud rate generator(BRG : address $1C_{16}$)

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	○	○

Fig. 2.4.9 Structure of Baud rate generator

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt edge selection register
(INTEDGE: address $3A_{16}$)

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	INT_2 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
3	INT_3 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
4	Serial/O2/INT_3 interrupt source bit	0: INT_3 interrupt selected 1: Serial I/O2 interrupt selected	0	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
6			0	○	×
7			0	○	×

Fig. 2.4.10 Structure of Interrupt edge selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1
(IREQ1 : address $3C_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	When writing to this bit, set “0” to this bit.		0	○	✻
2	INT_1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT_2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_3/Serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	When writing to this bit, set “0” to this bit.		0	○	✻
6	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer Y 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. 2.4.11 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

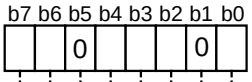
Interrupt request register 2
(IREQ2 : address $3D_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	Timer 2 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 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	$CNTR_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	$CNTR_1$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	A-D converter 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. 2.4.12 Structure of Interrupt request register 2

Interrupt control register 1

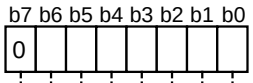


Interrupt control register 1
(ICON1 : address $3E_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Fix this bit to “0”.		0	○	○
2	INT_1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_2 nterrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_3/Serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Fix this bit to “0”.		0	○	○
6	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.4.13 Structure of Interrupt control register 1

Interrupt control register 2



Interrupt control register 2
(ICON2 : address $3F_{16}$)

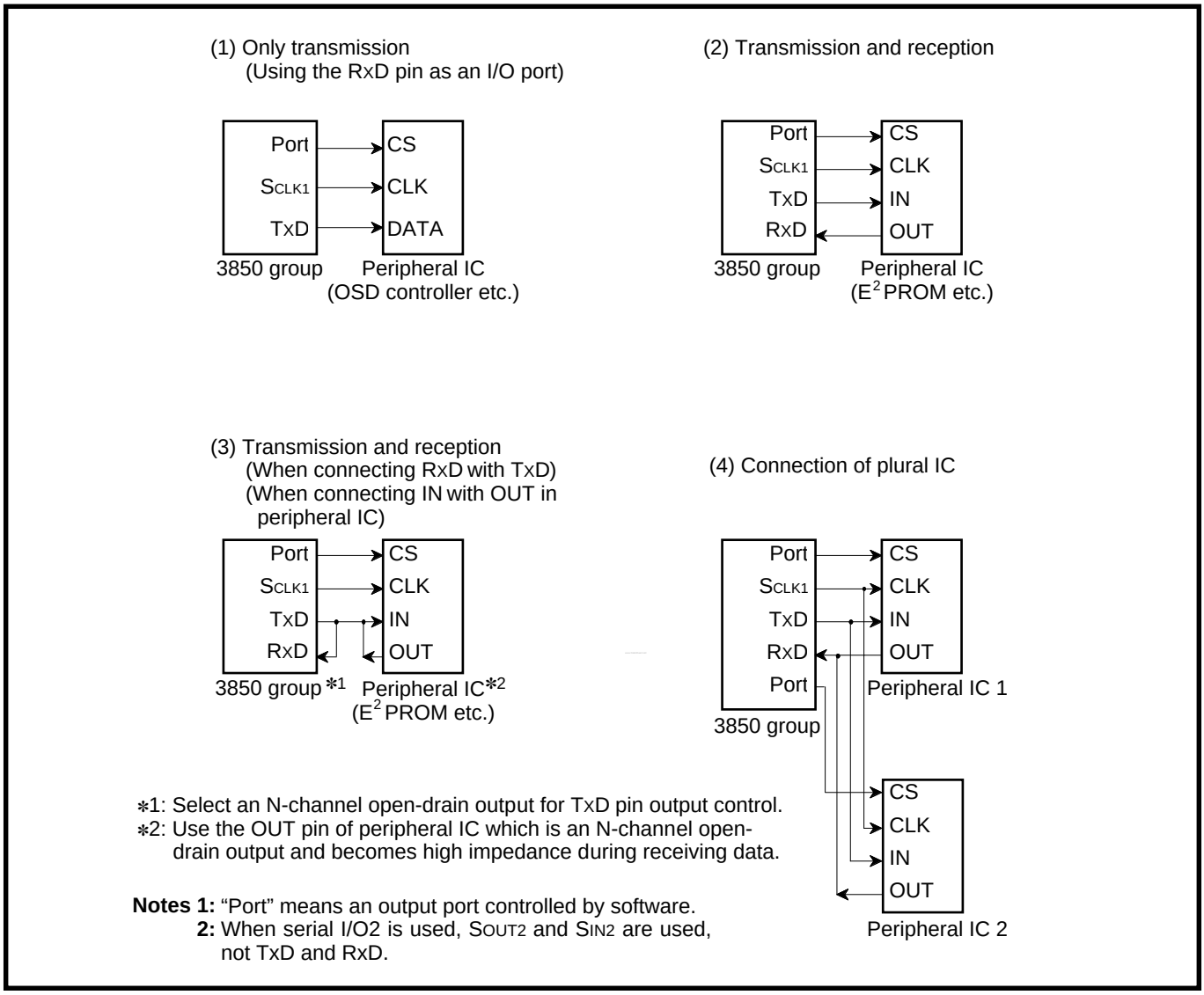
b	Name	Functions	At reset	R	W
0	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O1 receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O1 transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	$CNTR_0$ interrupt enable bit	0 : Interrupt disabled	0	○	○
5	$CNTR_1$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	A-D converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.4.14 Structure of Interrupt control register 2

2.4.3 Serial I/O connection examples

(1) Control of peripheral IC equipped with CS pin

Figure 2.4.15 shows connection examples of a peripheral IC equipped with the CS pin. There are connection examples using a clock synchronous serial I/O mode.



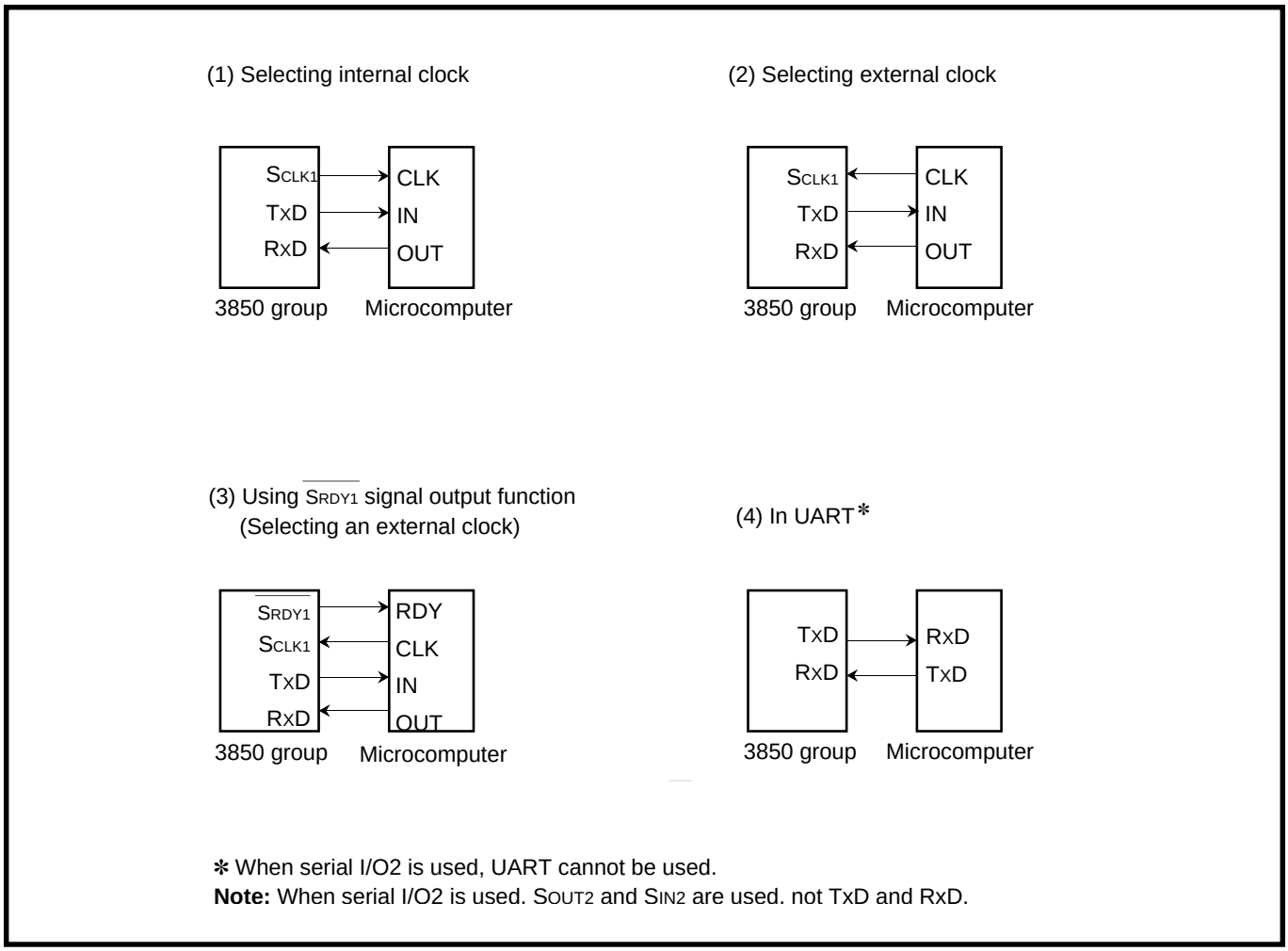
∗1: Select an N-channel open-drain output for TxD pin output control.
∗2: Use the OUT pin of peripheral IC which is an N-channel open-drain output and becomes high impedance during receiving data.

Notes 1: "Port" means an output port controlled by software.
2: When serial I/O2 is used, SOUT2 and SIN2 are used, not TxD and RxD.

Fig. 2.4.15 Serial I/O connection examples (1)

(2) Connection with microcomputer

Figure 2.4.16 shows connection examples with another microcomputer.



✻ When serial I/O2 is used, UART cannot be used.

Note: When serial I/O2 is used. SOUT2 and SIN2 are used. not TxD and RxD.

Fig. 2.4.16 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. A clock synchronous is used as a data format of Serial I/O2.

Figure 2.4.17 shows the serial I/O transfer data format.

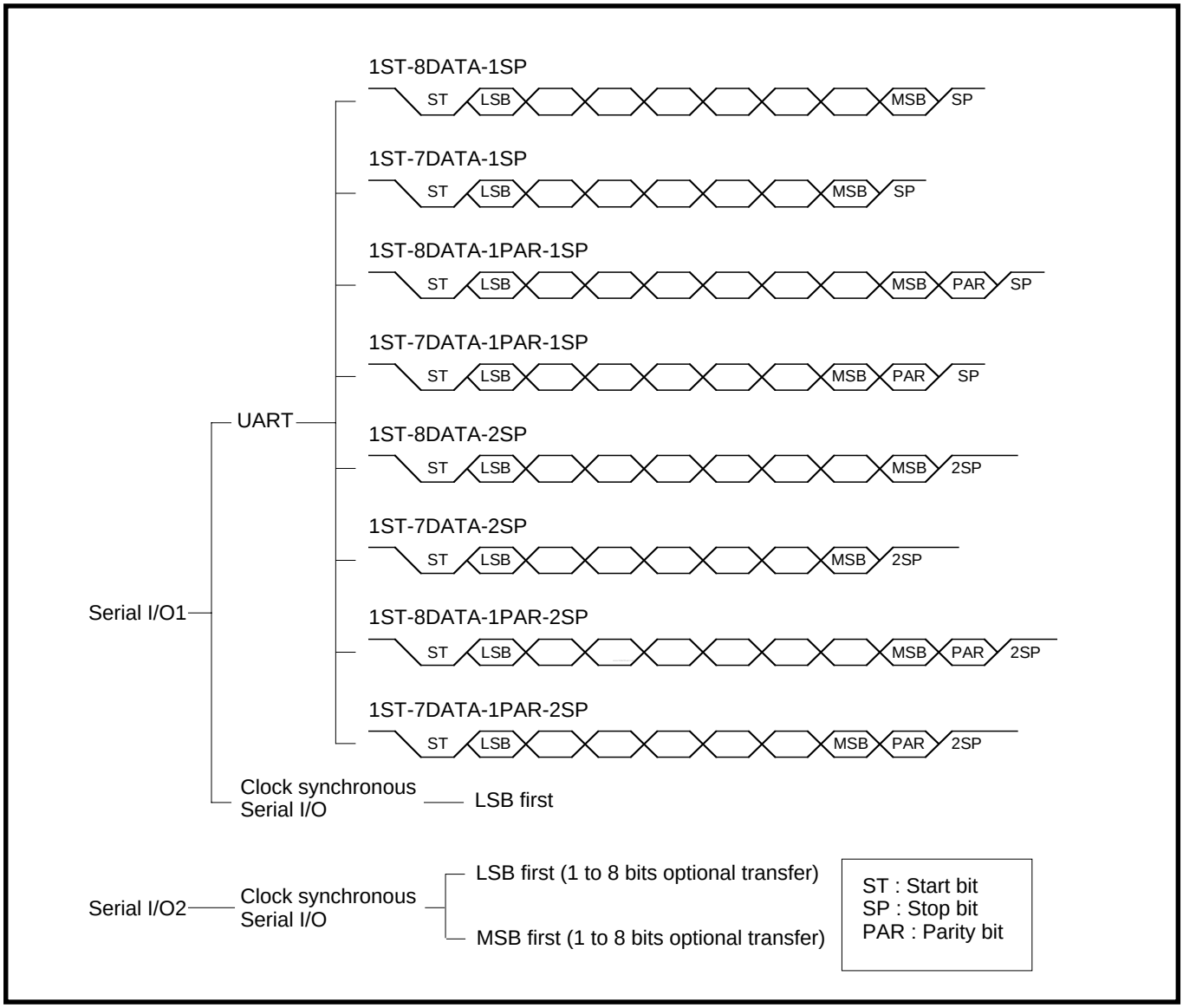


Fig. 2.4.17 Serial I/O transfer data format

2.4.5 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 2.4.18 shows a connection diagram, and Figure 2.4.19 shows a timing chart.
Figure 2.4.20 shows a registers setting relevant to the transmitting side, and Figure 2.4.21 shows registers setting relevant to the receiving side.

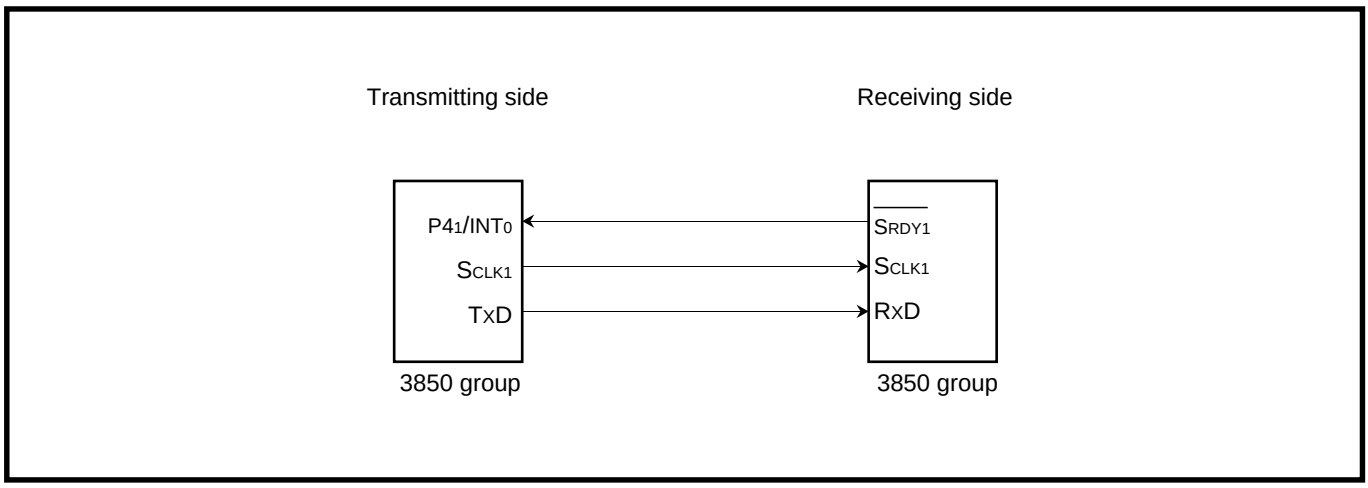


Fig. 2.4.18 Connection diagram

Specifications :
- The 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)
- The $\overline{S_{RDY1}}$ (receivable signal) is used.
- The receiving side outputs the $\overline{S_{RDY1}}$ signal at intervals of 2 ms (generated by timer), and 2-byte data is transferred from the transmitting side to the receiving side.

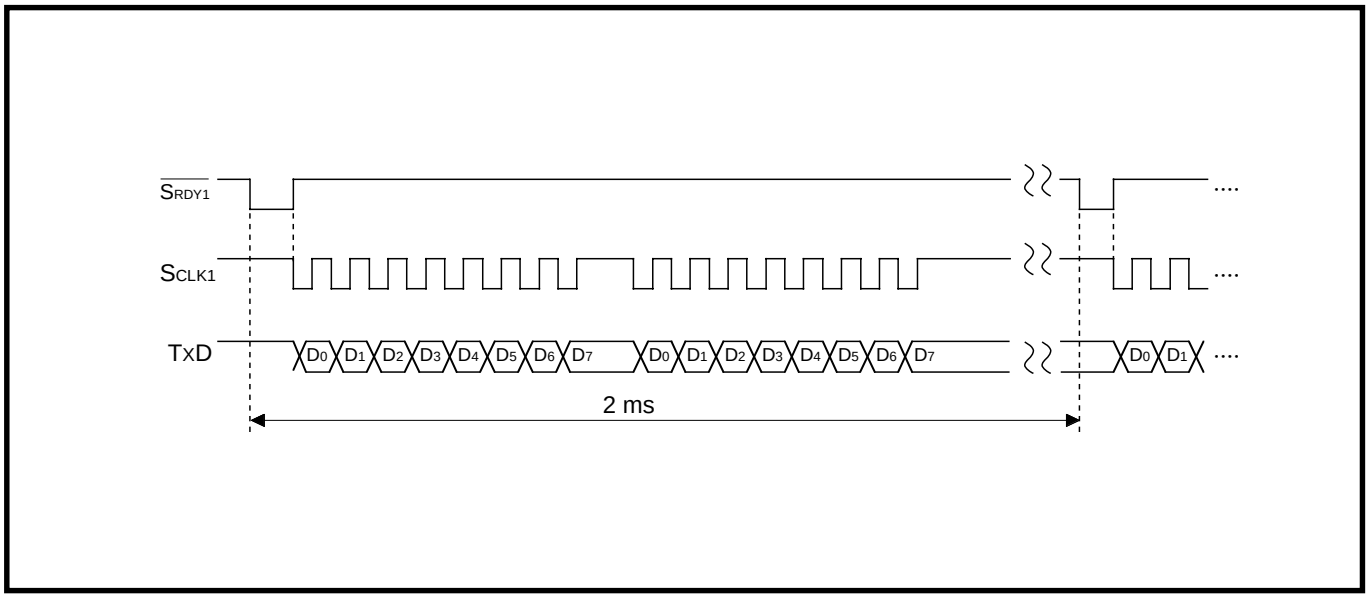
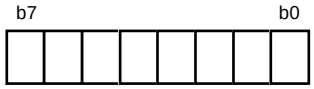


Fig. 2.4.19 Timing chart

Transmitting side

Serial I/O1 status register (Address : 19_{16})

SIOSTS

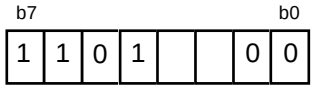


→ Transmit buffer empty flag
- Confirm that the data has been transferred from Transmit buffer register to Transmit shift register.
- When this flag is "1", it is possible to write the next transmission data in to Transmit buffer register.

→ 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 : $1A_{16}$)

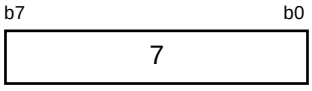
SIOCON



→ 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 (Address : $1C_{16}$)

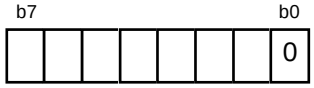
BRG



Set "division ratio – 1".

Interrupt edge selection register (Address : $3A_{16}$)

INTEDGE



→ INT_0 interrupt edge selection bit : Falling edge active

Fig. 2.4.20 Registers setting relevant to transmitting side

Receiving side

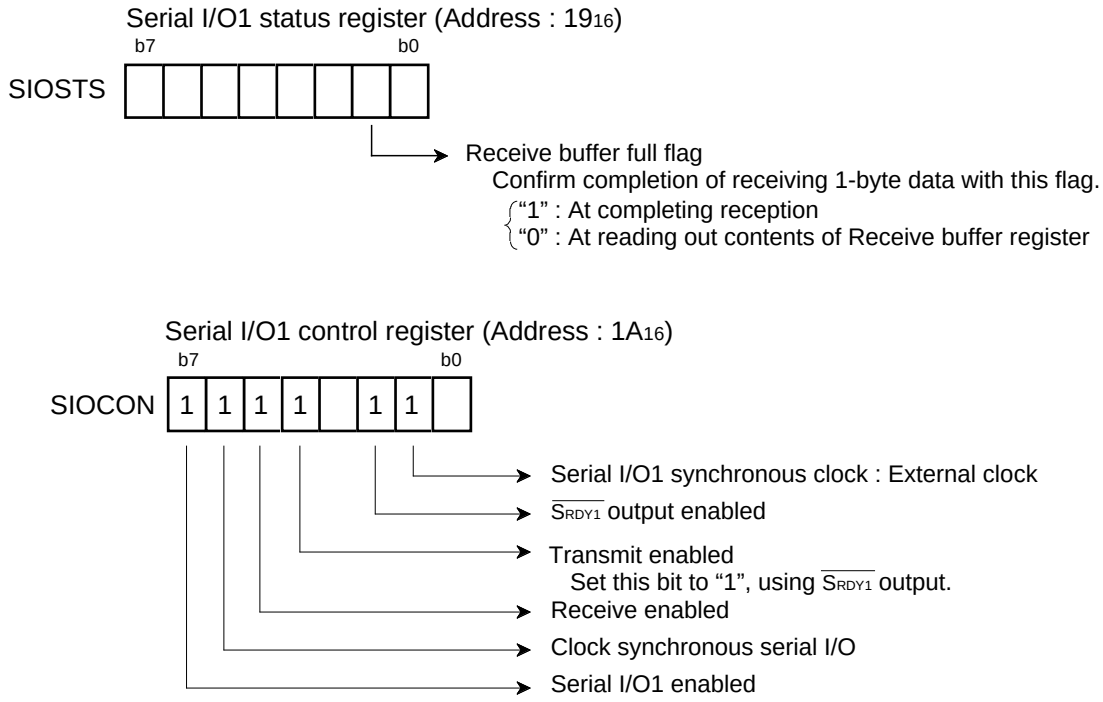


Fig. 2.4.21 Registers setting relevant to receiving side

Figure 2.4.22 shows a control procedure of the transmitting side, and Figure 2.4.23 shows a control procedure of the receiving side.

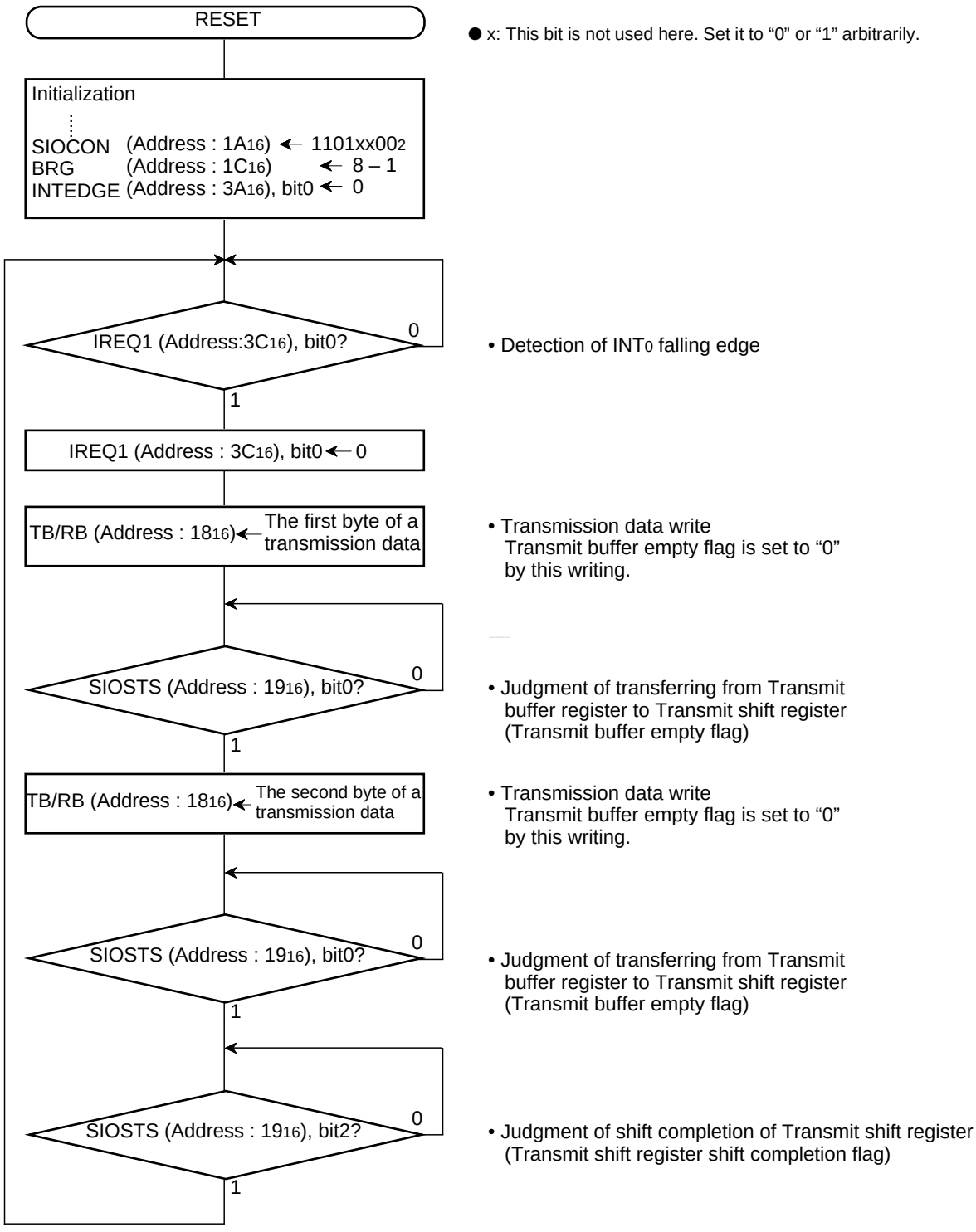


Fig. 2.4.22 Control procedure of transmitting side

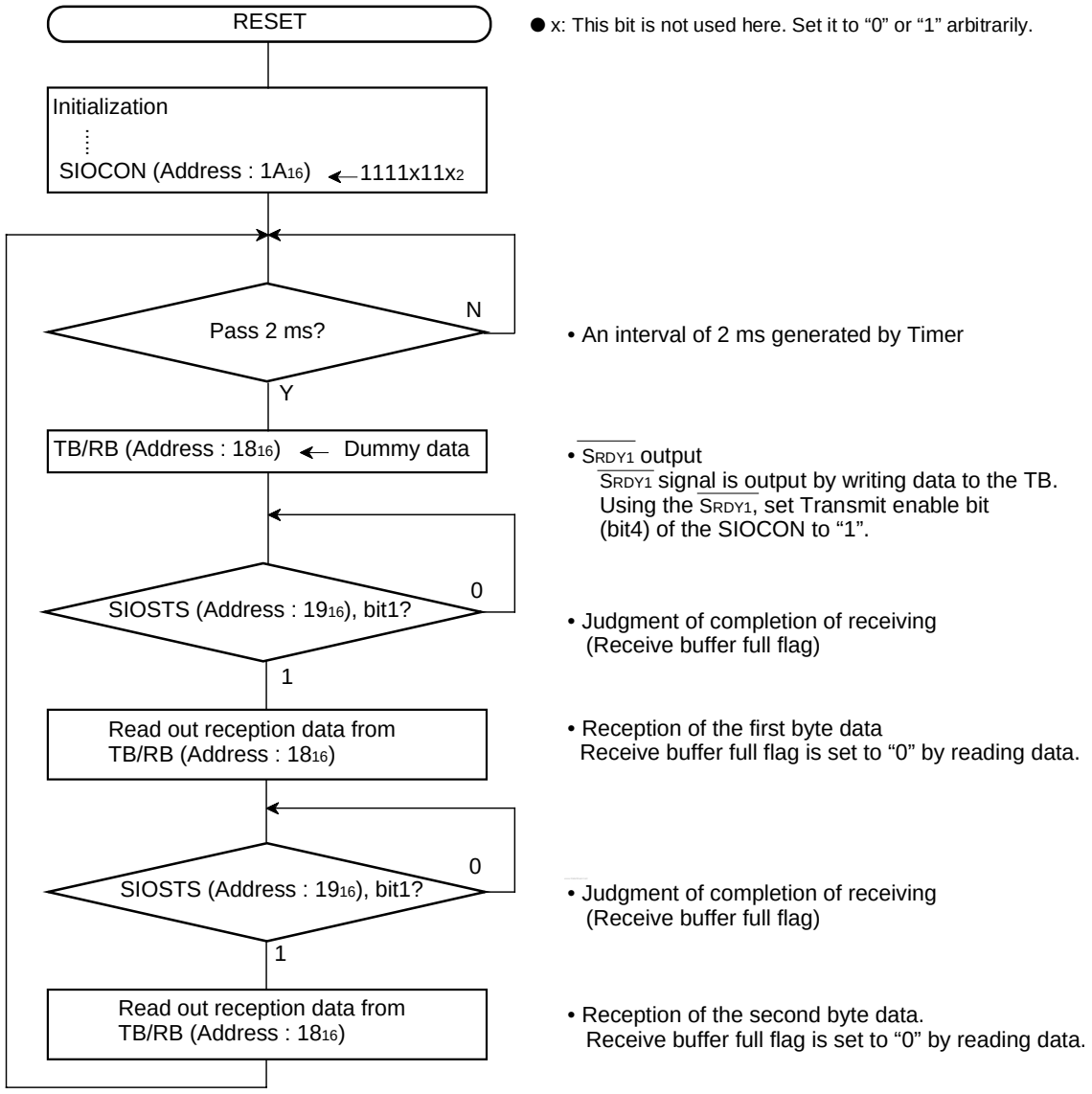


Fig. 2.4.23 Control procedure of receiving side

(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 CS signal is output to a peripheral IC through port $P4_3$.

Figure 2.4.24 shows a connection diagram, and Figure 2.4.25 shows a timing chart.

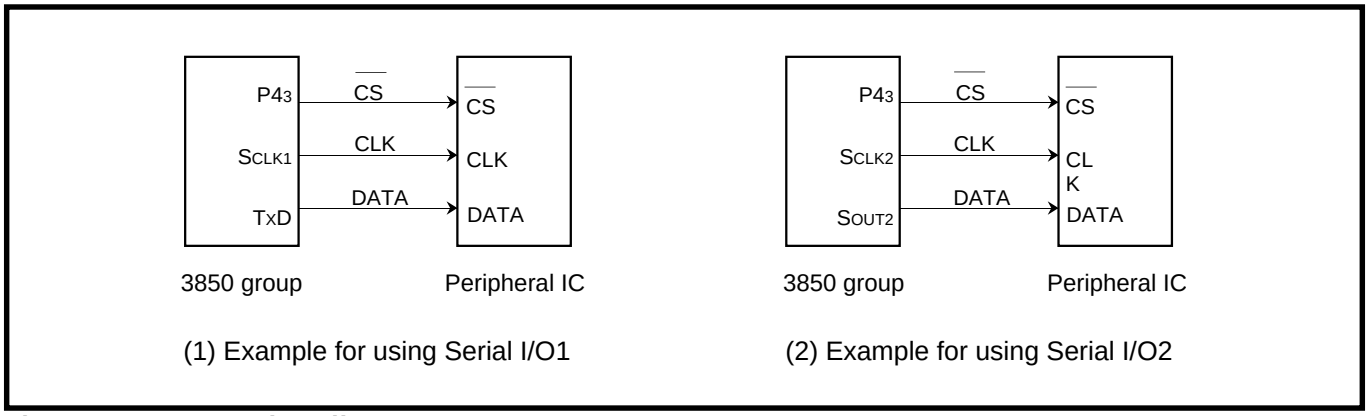
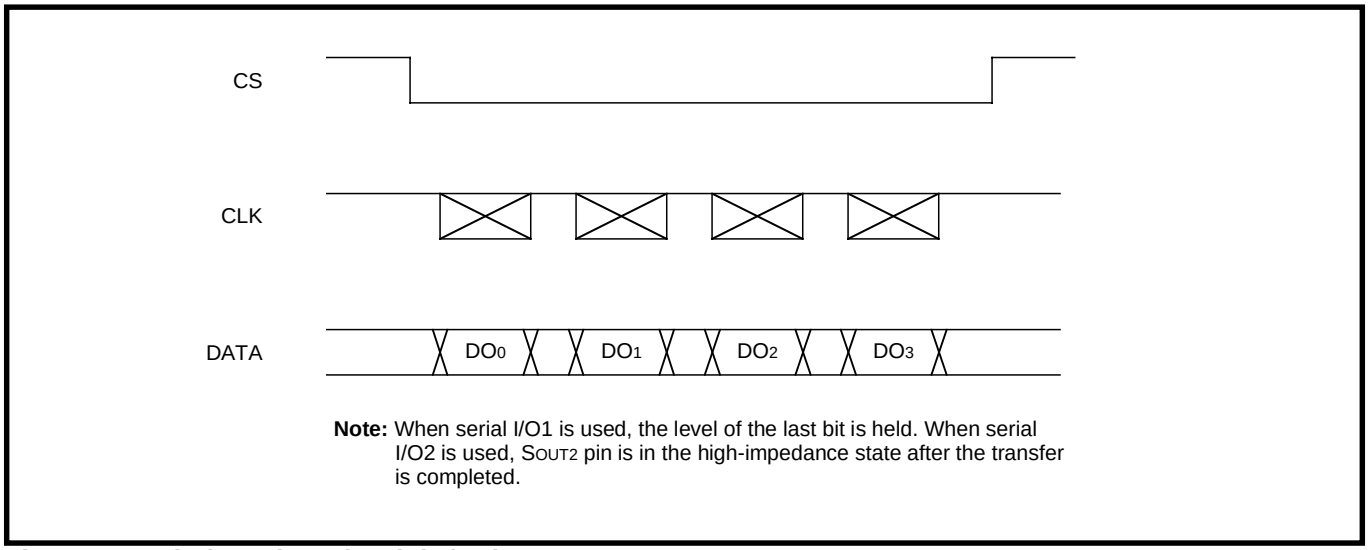


Fig. 2.4.24 Connection diagram

Specifications :
- The 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 $P4_3$ is connected to the $\overline{CS}$ pin ("L" active) of the peripheral IC for transmission control; the output level of port $P4_3$ is controlled by software.



Note: When serial I/O1 is used, the level of the last bit is held. When serial I/O2 is used, S_{OUT2} pin is in the high-impedance state after the transfer is completed.

Fig. 2.4.25 Timing chart (Serial I/O1)

Figure 2.4.26 shows registers setting relevant to Serial I/O1, and Figure 2.4.27 shows a setting of serial I/O1 transmission data.

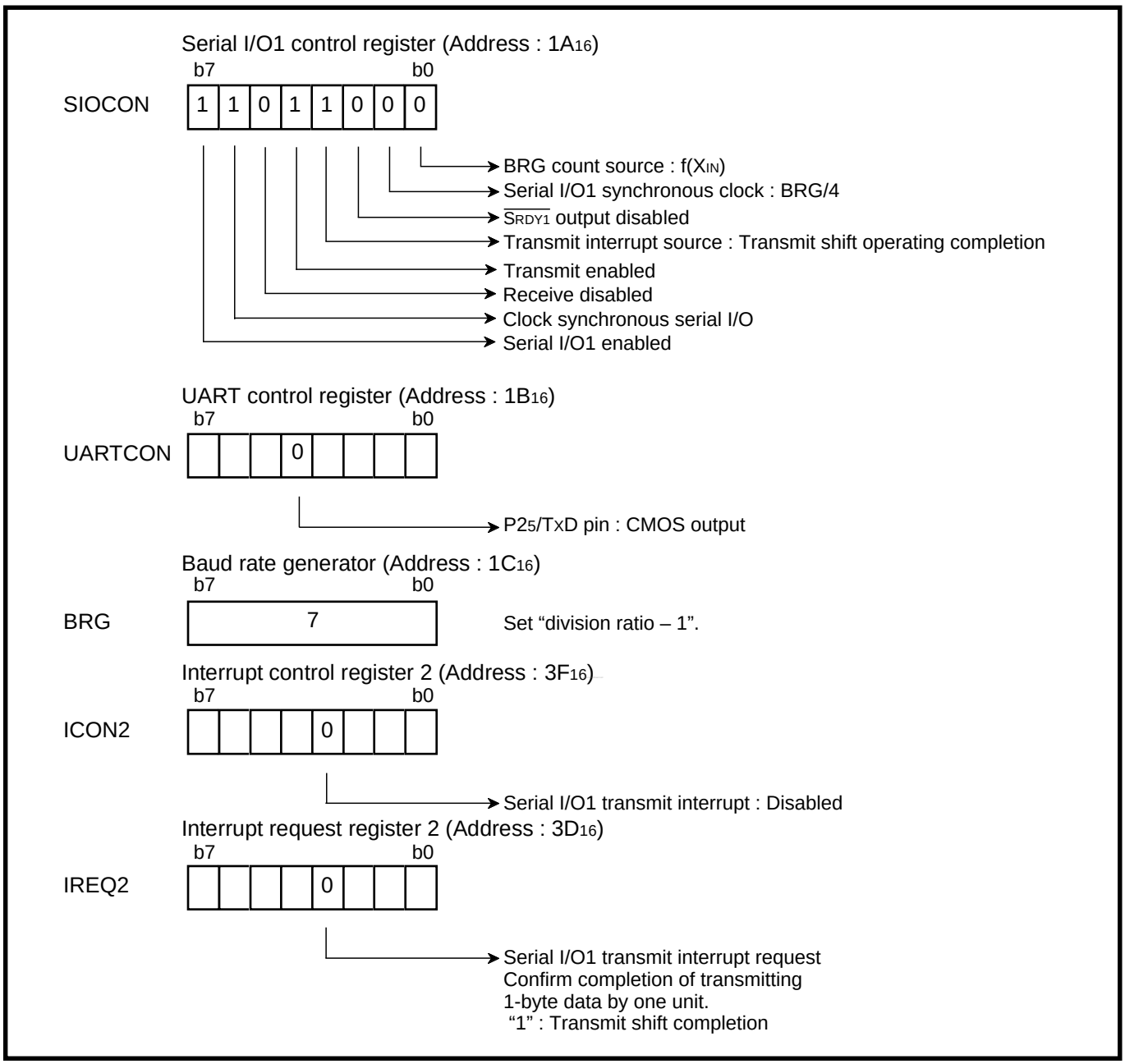


Fig. 2.4.26 Registers setting relevant to Serial I/O1

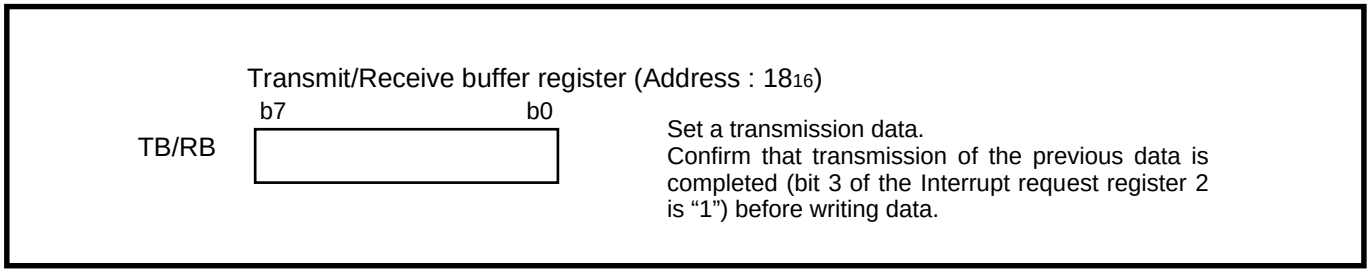


Fig. 2.4.27 Setting of serial I/O1 transmission data

Example for using Serial I/O1

When the registers are set as shown in Figure 2.4.26, the 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 4-byte data has been transmitted.

Figure 2.4.28 shows a control procedure of Serial I/O1.

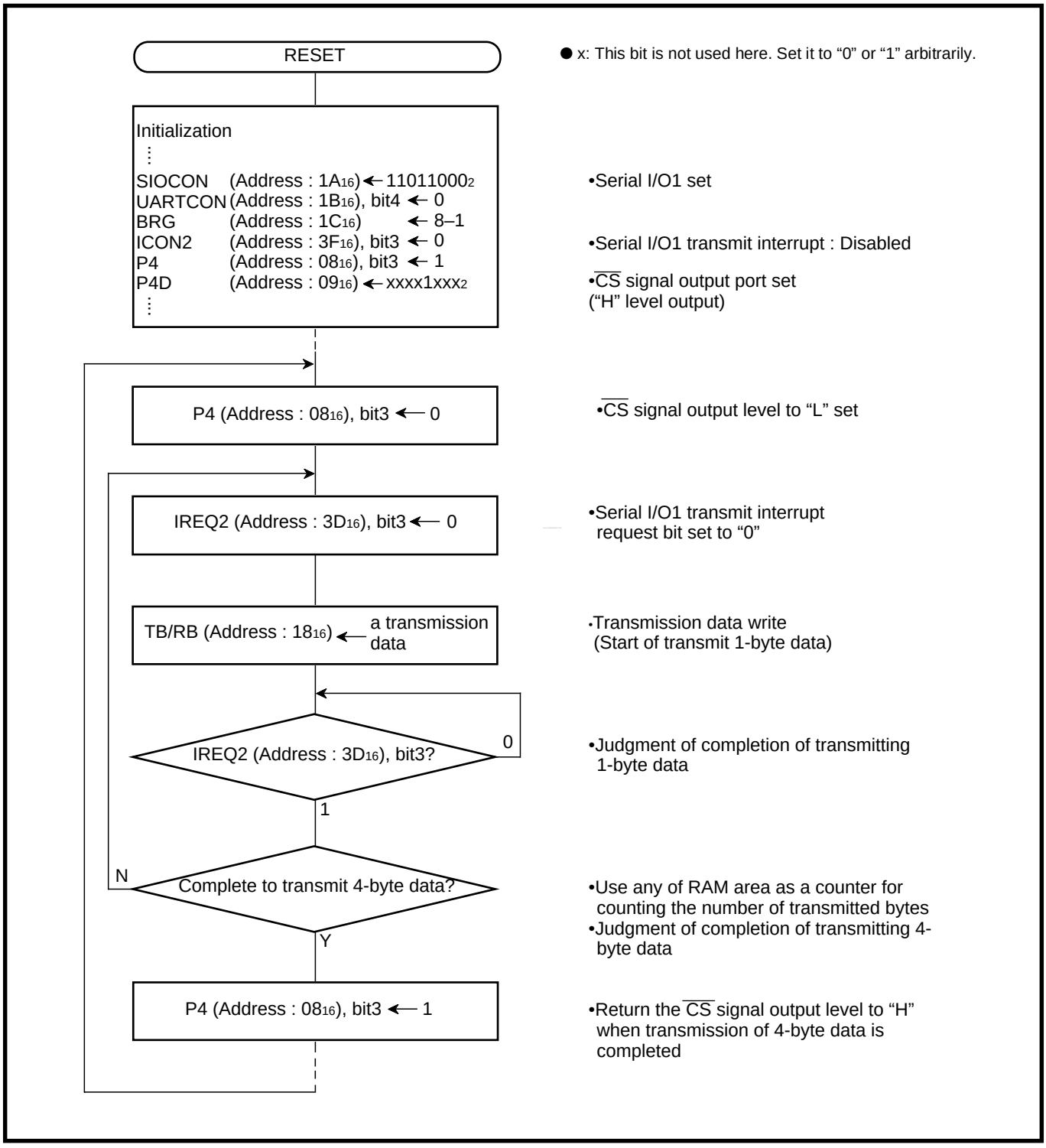


Fig. 2.4.28 Control procedure of Serial I/O1

Figure 2.4.29 shows registers setting relevant to Serial I/O2, and Figure 2.4.30 shows a setting of serial I/O2 transmission data.

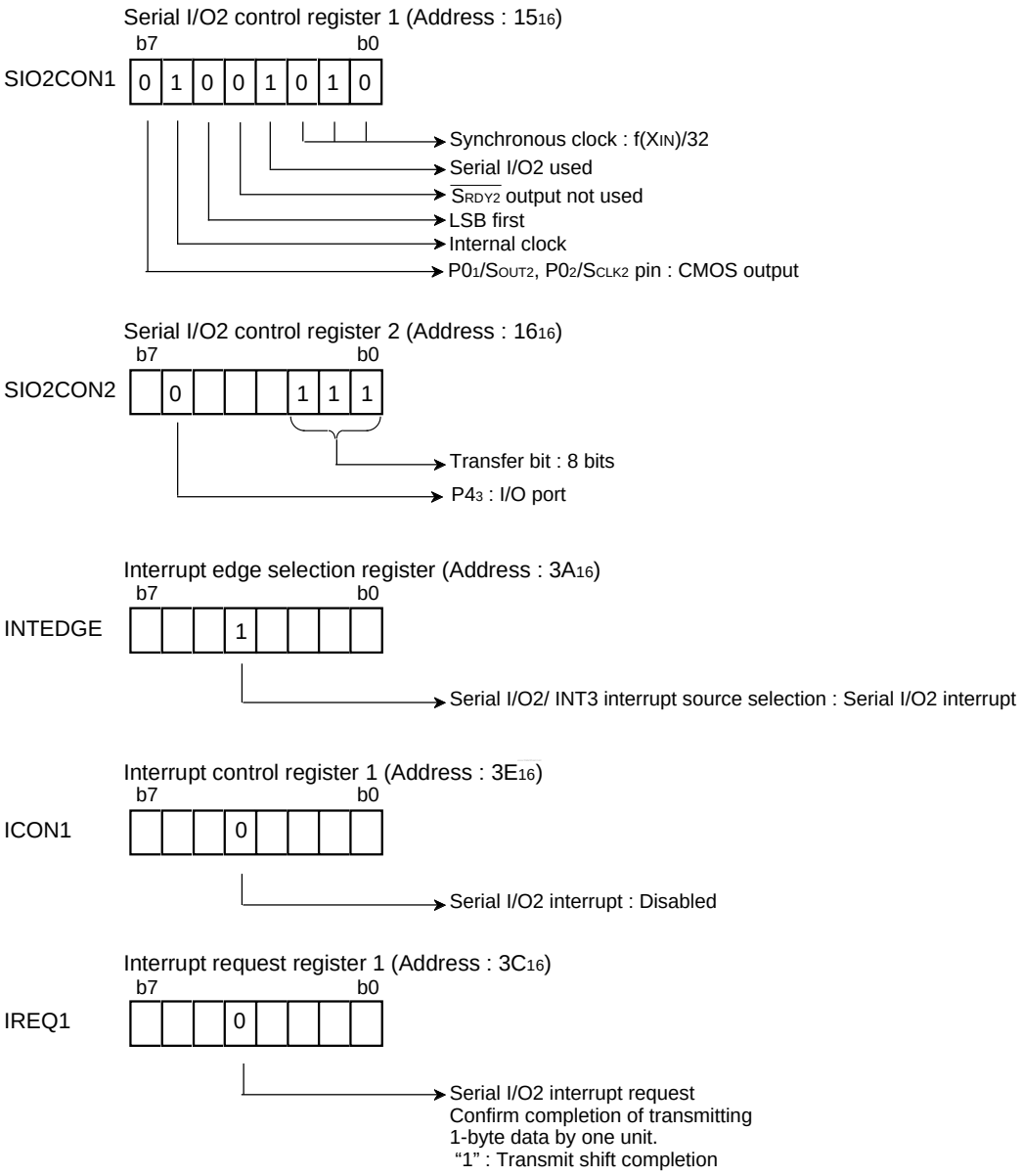


Fig. 2.4.29 Registers setting relevant to Serial I/O2

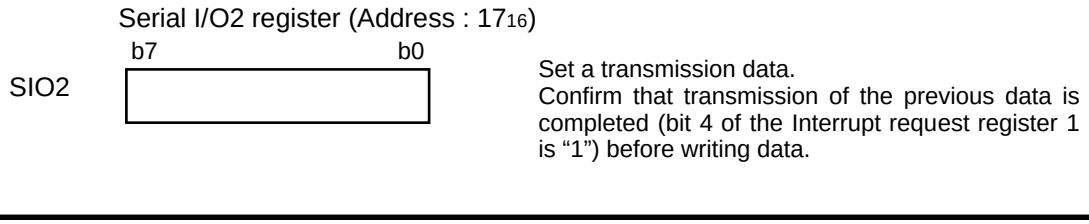


Fig. 2.4.30 Setting of serial I/O2 transmission data

Example for using Sreial I/O2

When the registers are set as shown in Fig. 2.4.29, the Serial I/O2 can transmit 1-byte data by writing data to the serial I/O2 register.

Thus, after setting the $\overline{CS}$ signal to "L", write the transmission data to Serial I/O2 by each 1 byte, and return the $\overline{CS}$ signal to "H" when 4-byte data has been transmitted.

Figure 2.4.31 shows a control procedure of Serial I/O2.

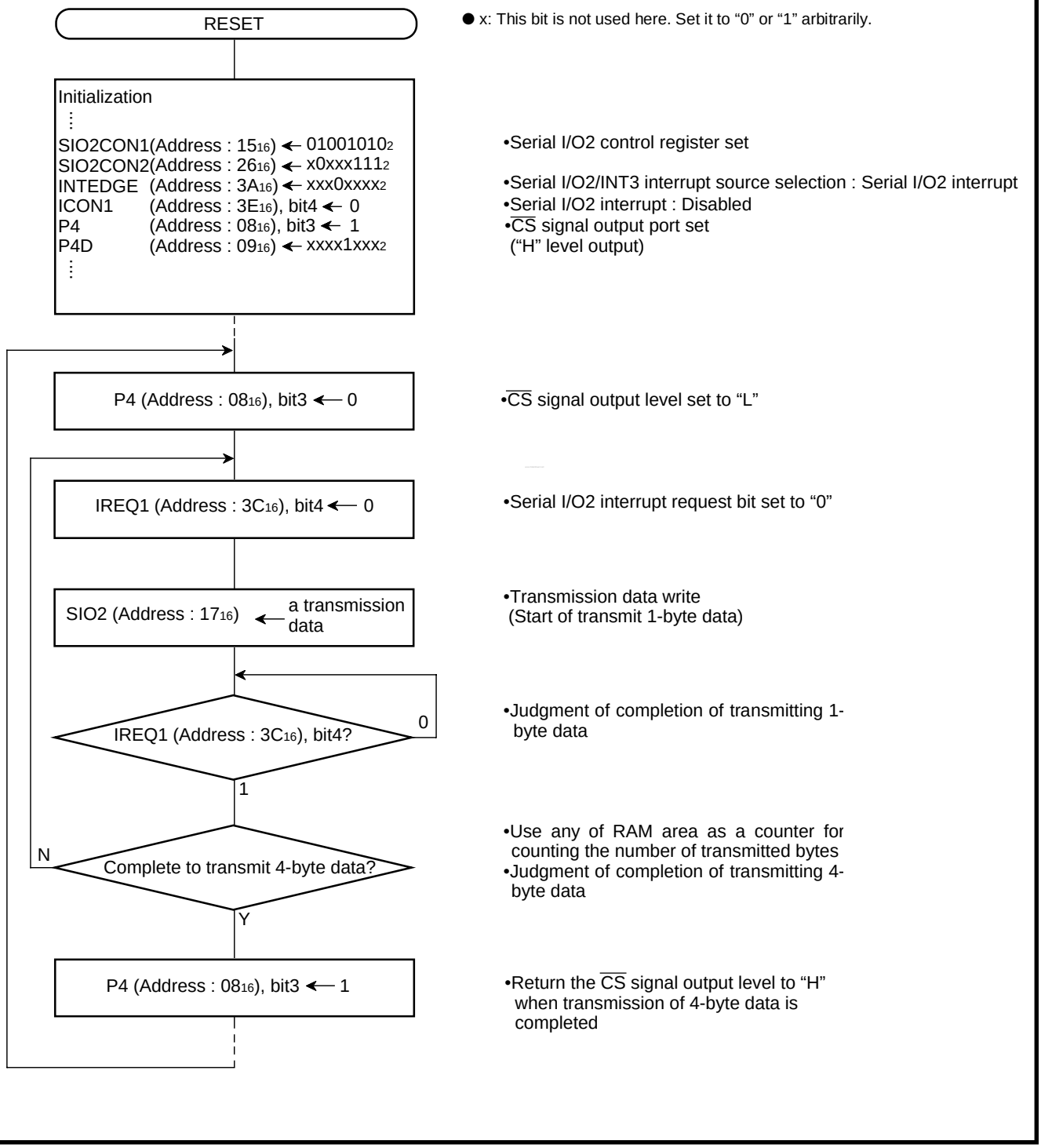


Fig. 2.4.31 Control procedure of Serial I/O2

(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.

Figure 2.4.32 shows a connection diagram.

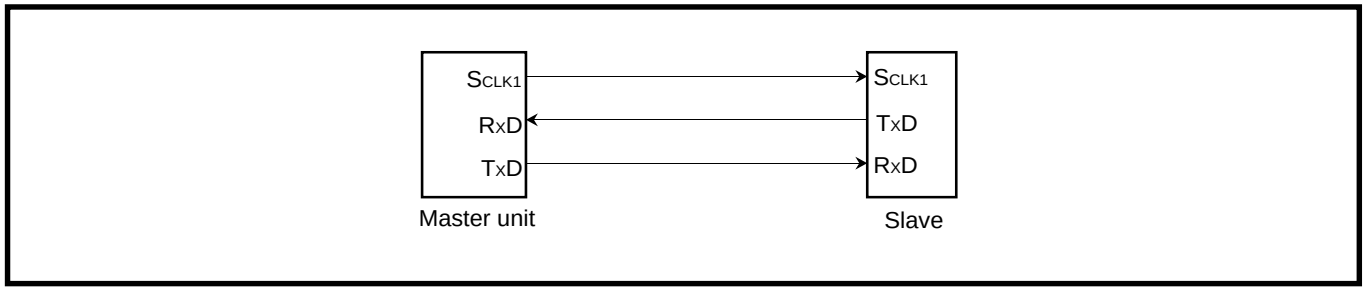


Fig. 2.4.32 Connection diagram

Specifications :

- The 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 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.

2.4 Serial I/O

The communication is performed according to the timing shown in Figure 2.4.33. 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 2.4.34 shows relevant registers setting.

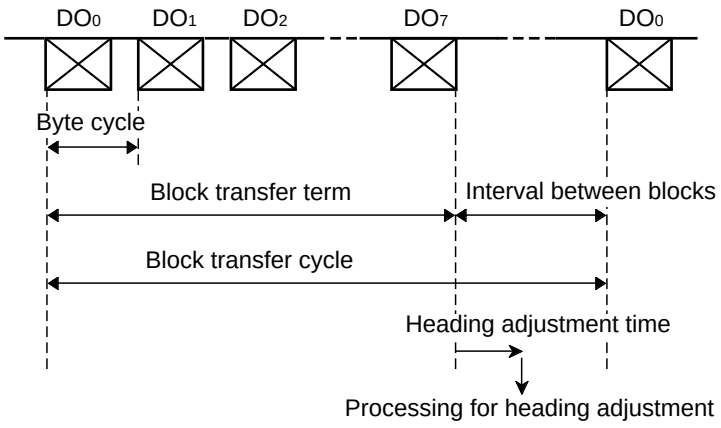


Fig. 2.4.33 Timing chart

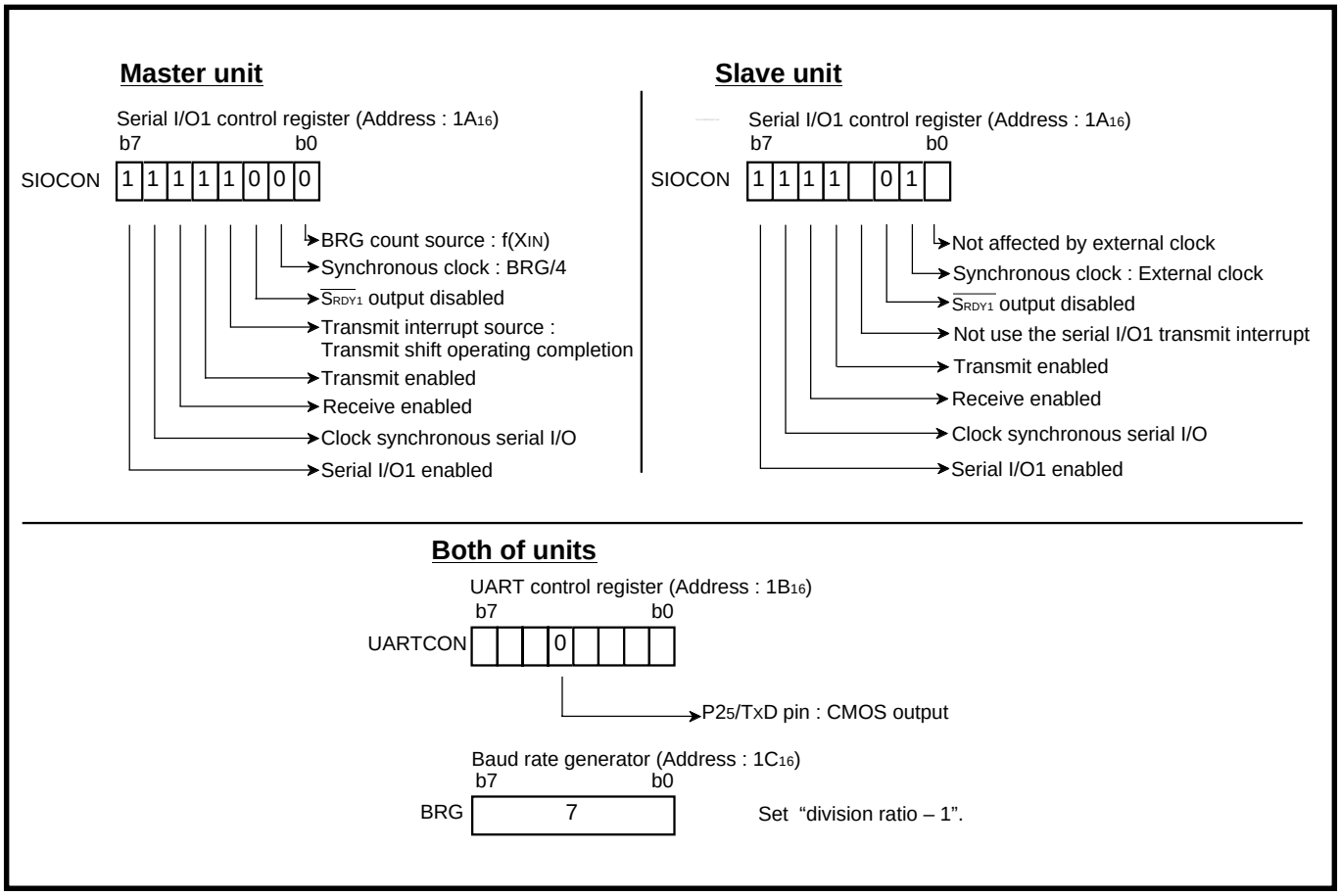


Fig. 2.4.34 Relevant registers setting

Control procedure :

● Control in the master unit

After setting the relevant registers shown in Figure 2.4.34, 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 2.4.33, 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 2.4.35 shows a control procedure of the master unit using timer interrupts.

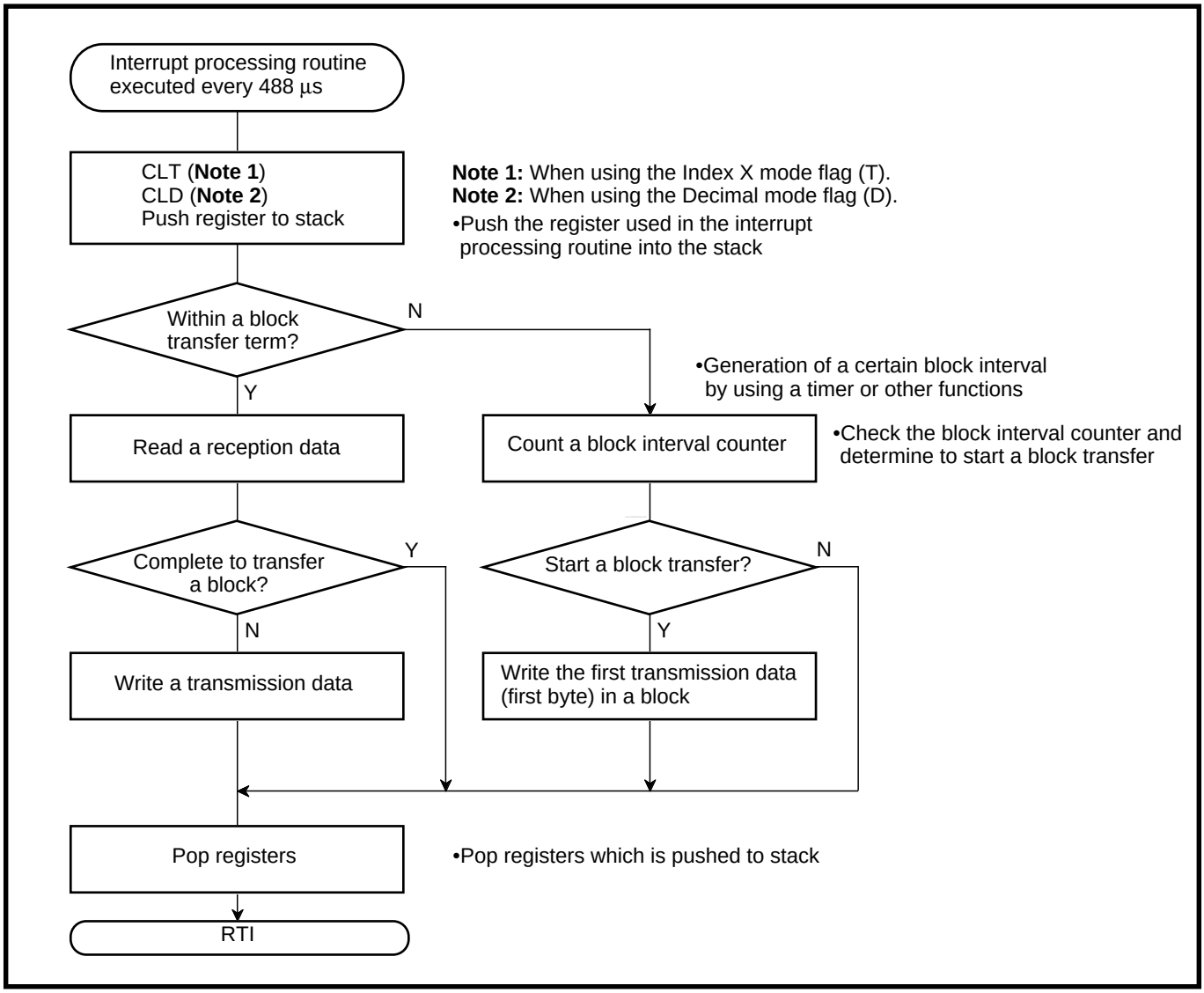


Fig. 2.4.35 Control procedure of master unit

● Control in the slave unit

After setting the relevant registers as shown in Figure 2.4.34, the slave unit becomes the state where a synchronous clock can be received at any time, and the serial I/O receive interrupt request bit is set to "1" each time an 8-bit synchronous clock is received.

In the serial I/O 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/O 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 2.4.36 shows a control procedure of the slave unit using the serial I/O receive interrupt and any timer interrupt (for heading adjustment).

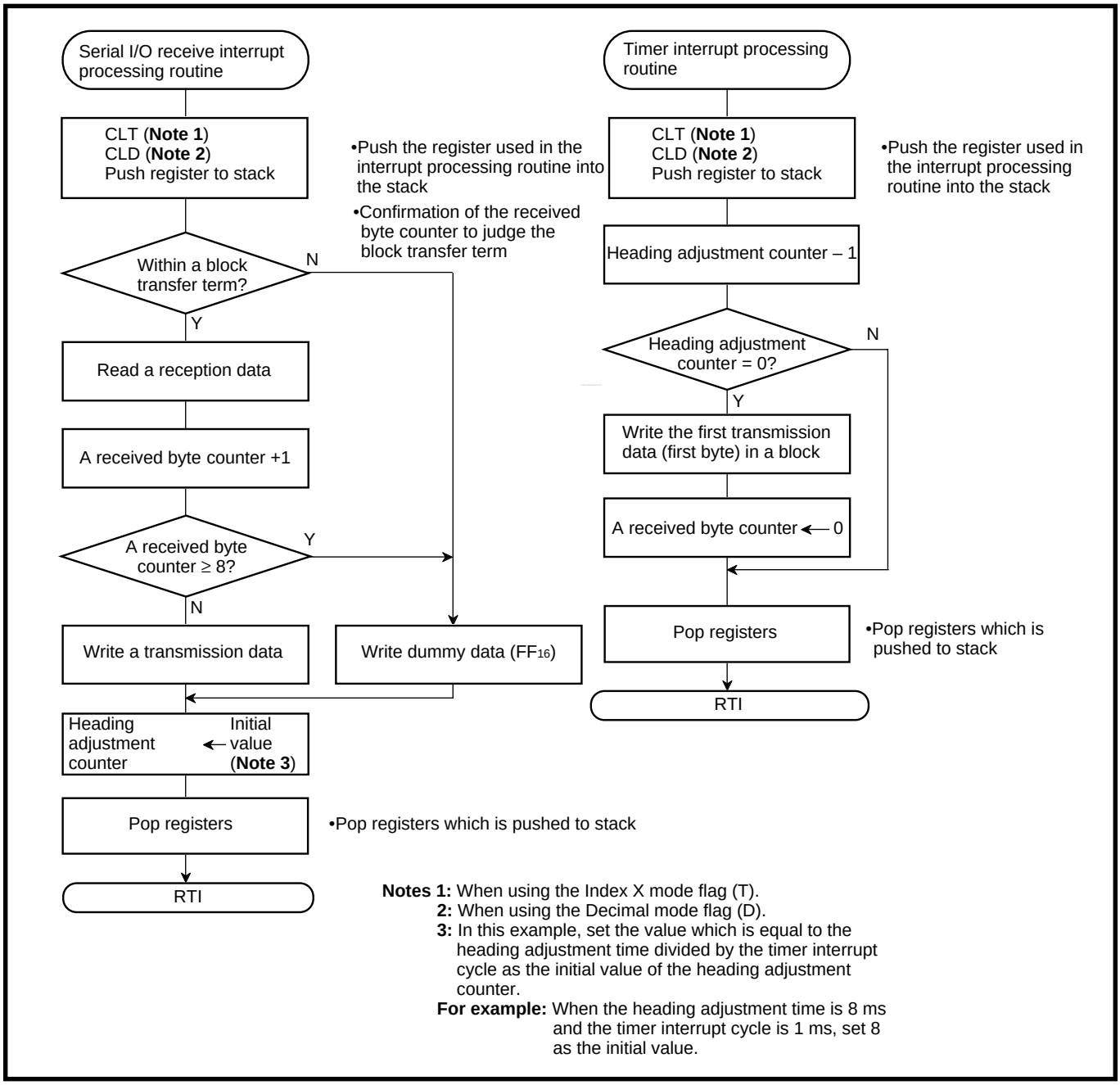


Fig. 2.4.36 Control procedure of slave unit

(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 2.4.37 shows a connection diagram, and Figure 2.4.38 shows a timing chart.

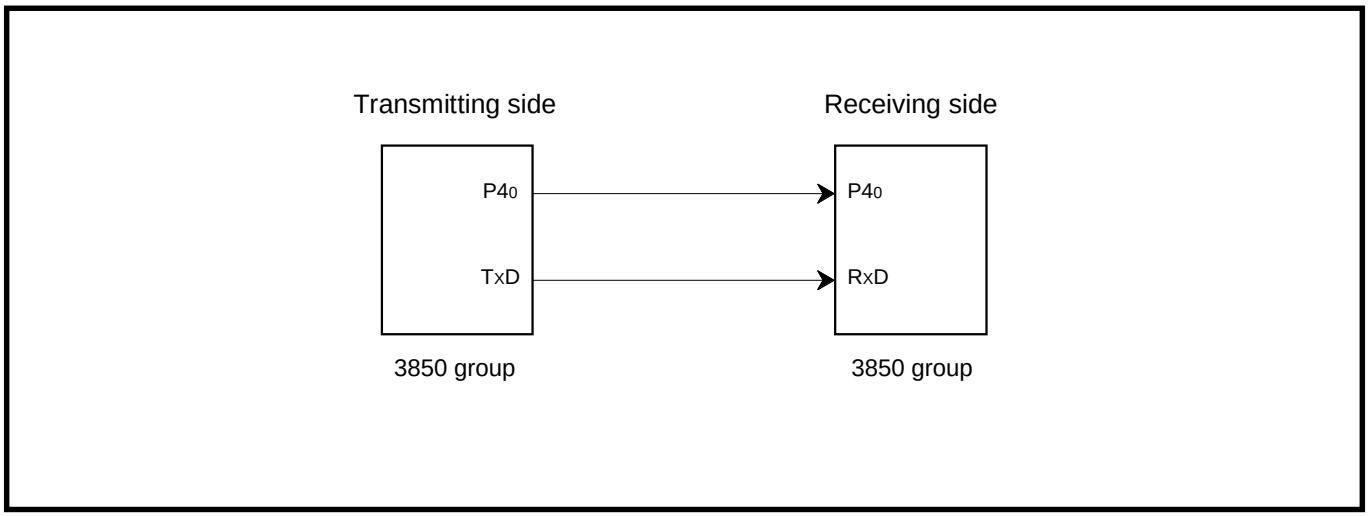


Fig. 2.4.37 Connection diagram

Specifications :
- The 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.

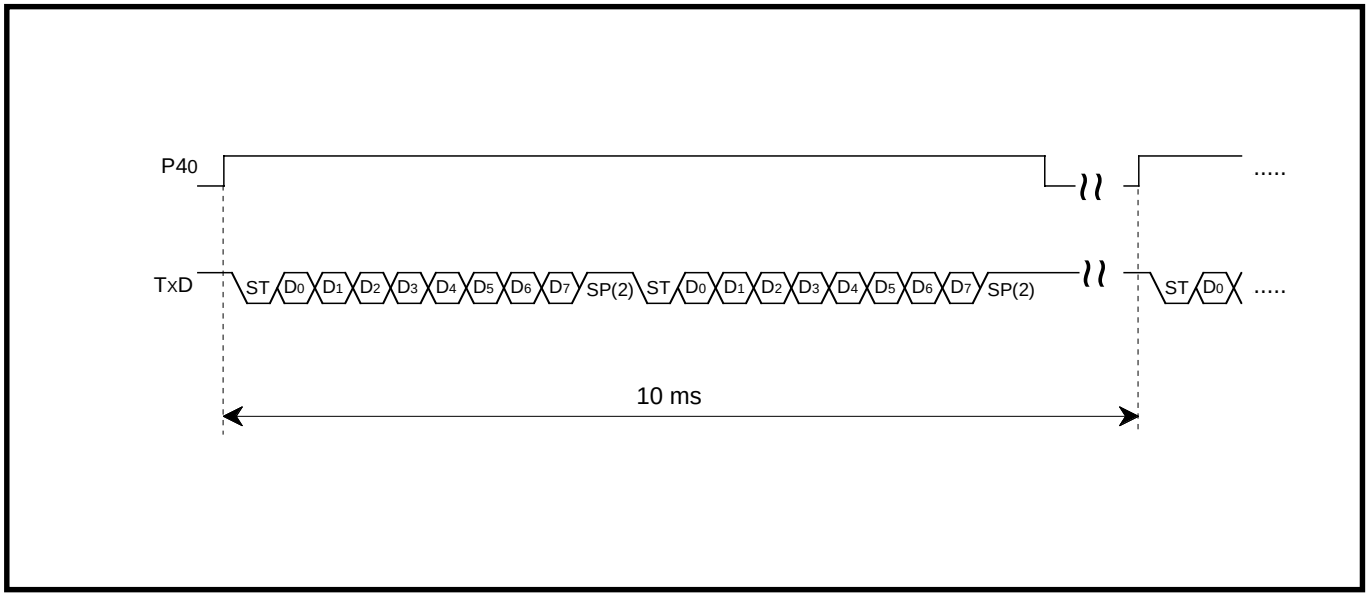


Fig. 2.4.38 Timing chart (using UART)

2.4 Serial I/O

Table 2.4.1 and Table 2.4.2 show setting examples of the baud rate generator (BRG) values and transfer bit rate values; Figure 2.4.39 shows registers setting relevant to the transmitting side; Figure 2.4.40 shows registers setting relevant to the receiving side.

Table 2.4.1 Setting examples of Baud rate generator values and transfer bit rate values (1)

BRG count source (Note 1)	BRG setting value	Transfer bit rate (bps) (Note 2)	
		at f(XIN) = 4.9152 MHz	at f(XIN) = 8 MHz
f(XIN)/4	255(FF_{16})	300	488.28125
f(XIN)/4	127($7F_{16}$)	600	976.5625
f(XIN)/4	63($3F_{16}$)	1200	1953.125
f(XIN)/4	31($1F_{16}$)	2400	3906.25
f(XIN)/4	15($0F_{16}$)	4800	7812.5
f(XIN)/4	7(07_{16})	9600	15625
f(XIN)/4	3(03_{16})	19200	31250
f(XIN)/4	1(01_{16})	38400	62500
f(XIN)	3(03_{16})	76800	125000
f(XIN)	1(01_{16})	153600	250000
f(XIN)	0(00_{16})	307200	500000

Table 2.4.2 Setting examples of Baud rate generator values and transfer bit rate values (2)

BRG count source (Note 1)	BRG setting value	Transfer bit rate (bps) (Note 2)
		at f(XIN) = 7.9872 MHz
f(XIN)/4	207(CF_{16})	600
f(XIN)/4	103(67_{16})	1200
f(XIN)/4	51(33_{16})	2400
f(XIN)/4	25(19_{16})	4800
f(XIN)/4	12($0C_{16}$)	9600
f(XIN)	25(19_{16})	19200
f(XIN)	12($0C_{16}$)	38400

Notes 1: Select the BRG count source with bit 0 of the serial I/O1 control register (Address : $1A_{16}$).

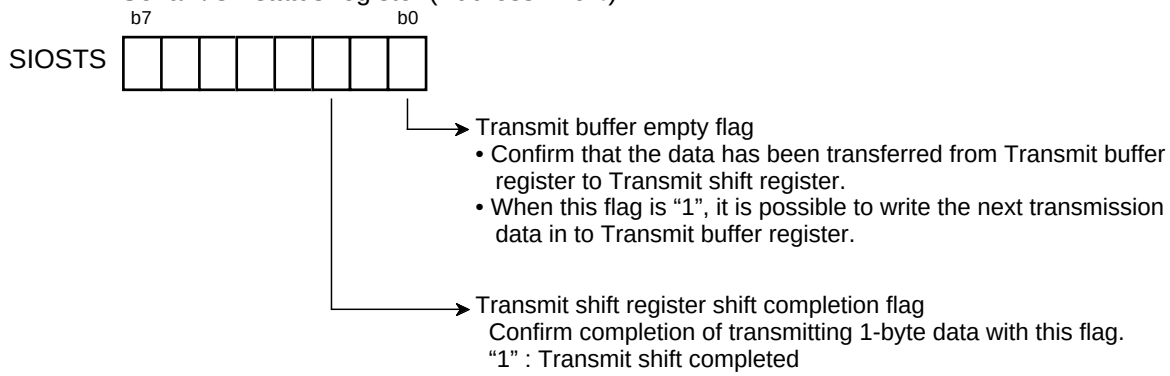
2: Equation of transfer bit rate:

$$\text{Transfer bit rate (bps)} = \frac{f(X_{IN})}{(\text{BRG setting value} + 1) \times 16 \times m^*}$$

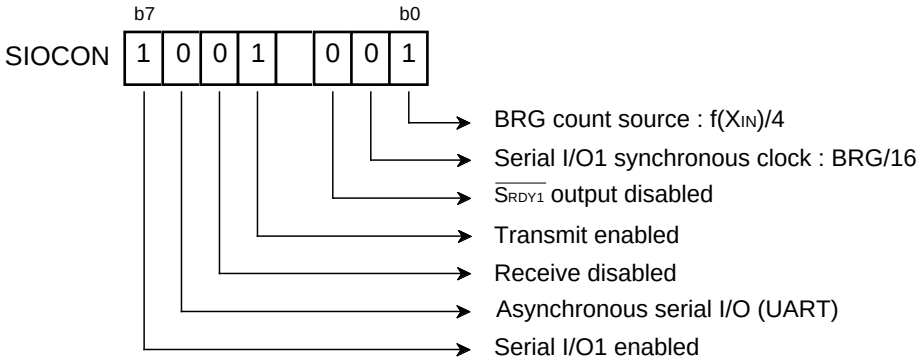
✻m: When bit 0 of the serial I/O1 control register (Address : $1A_{16}$) is set to “0”, a value of m is 1.
When bit 0 of the serial I/O1 control register (Address : $1A_{16}$) is set to “1”, a value of m is 4.

Transmitting side

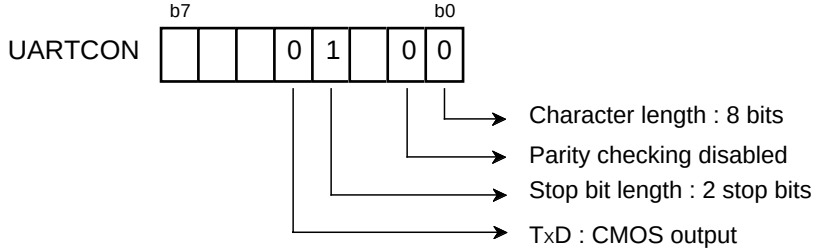
Serial I/O1 status register (Address : 19_{16})



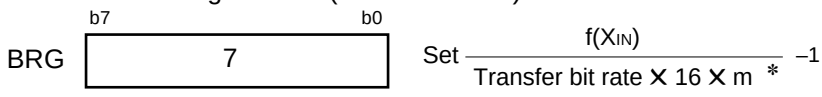
Serial I/O1 control register (Address : $1A_{16}$)



UART control register (Address : $1B_{16}$)



Baud rate generator (Address : $1C_{16}$)



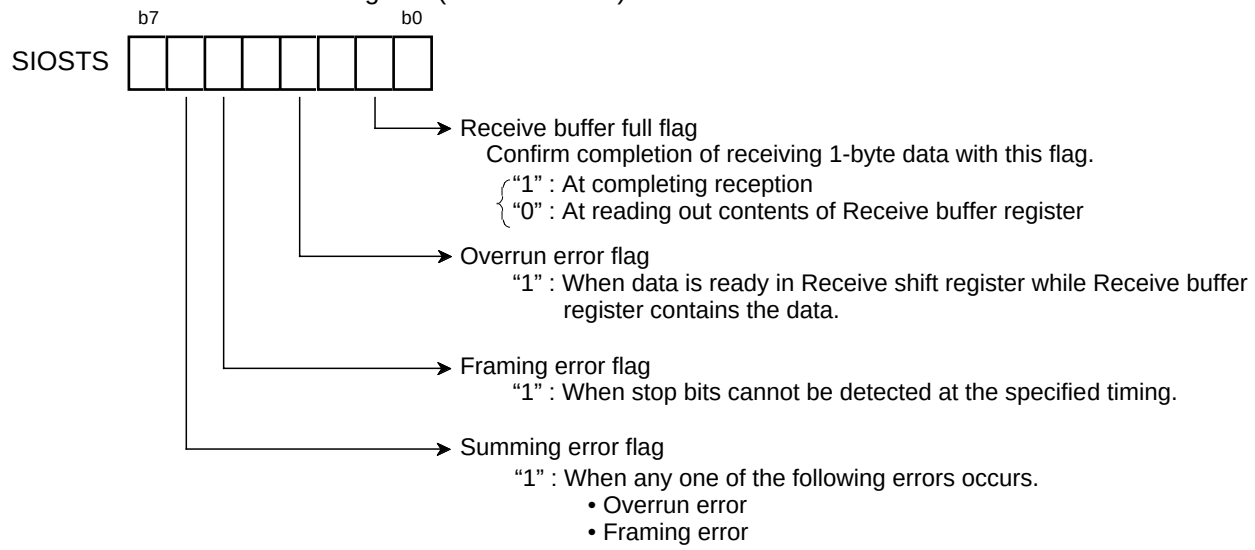
$$\text{Set } \frac{f(X_{IN})}{\text{Transfer bit rate} \times 16 \times m \ *} - 1$$

✻ When bit 0 of the Serial I/O1 control register (Address : $1A_{16}$) is set to "0", a value of m is 1.
When bit 0 of the Serial I/O1 control register (Address : $1A_{16}$) is set to "1", a value of m is 4.

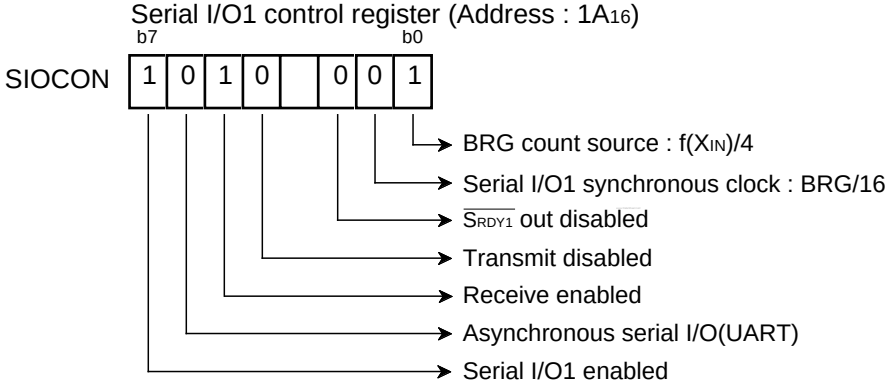
Fig. 2.4.39 Registers setting relevant to transmitting side

Receiving side

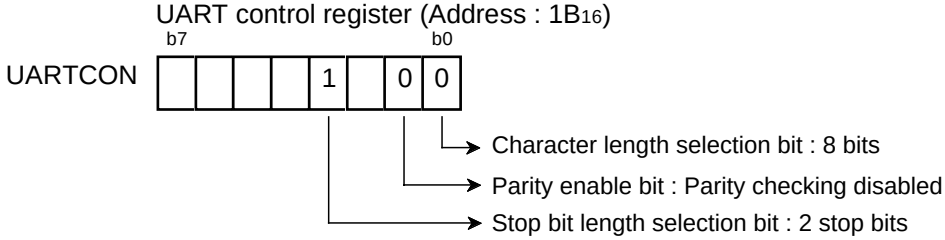
Serial I/O1 status register (Address : 19_{16})



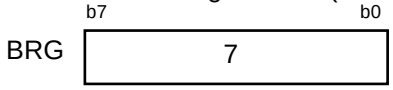
Serial I/O1 control register (Address : $1A_{16}$)



UART control register (Address : $1B_{16}$)



Baud rate generator (Address : $1C_{16}$)



$$\text{Set } \frac{f(X_{IN})}{\text{Transfer bit rate} \times 16 \times m^{*}} - 1$$

✻ When bit 0 of the Serial I/O1 control register (Address : $1A_{16}$) is set to "0", a value of m is 1.
When bit 0 of the Serial I/O1 control register (Address : $1A_{16}$) is set to "1", a value of m is 4.

Fig. 2.4.40 Registers setting relevant to receiving side

Figure 2.4.41 shows a control procedure of the transmitting side, and Figure 2.4.42 shows a control procedure of the receiving side.

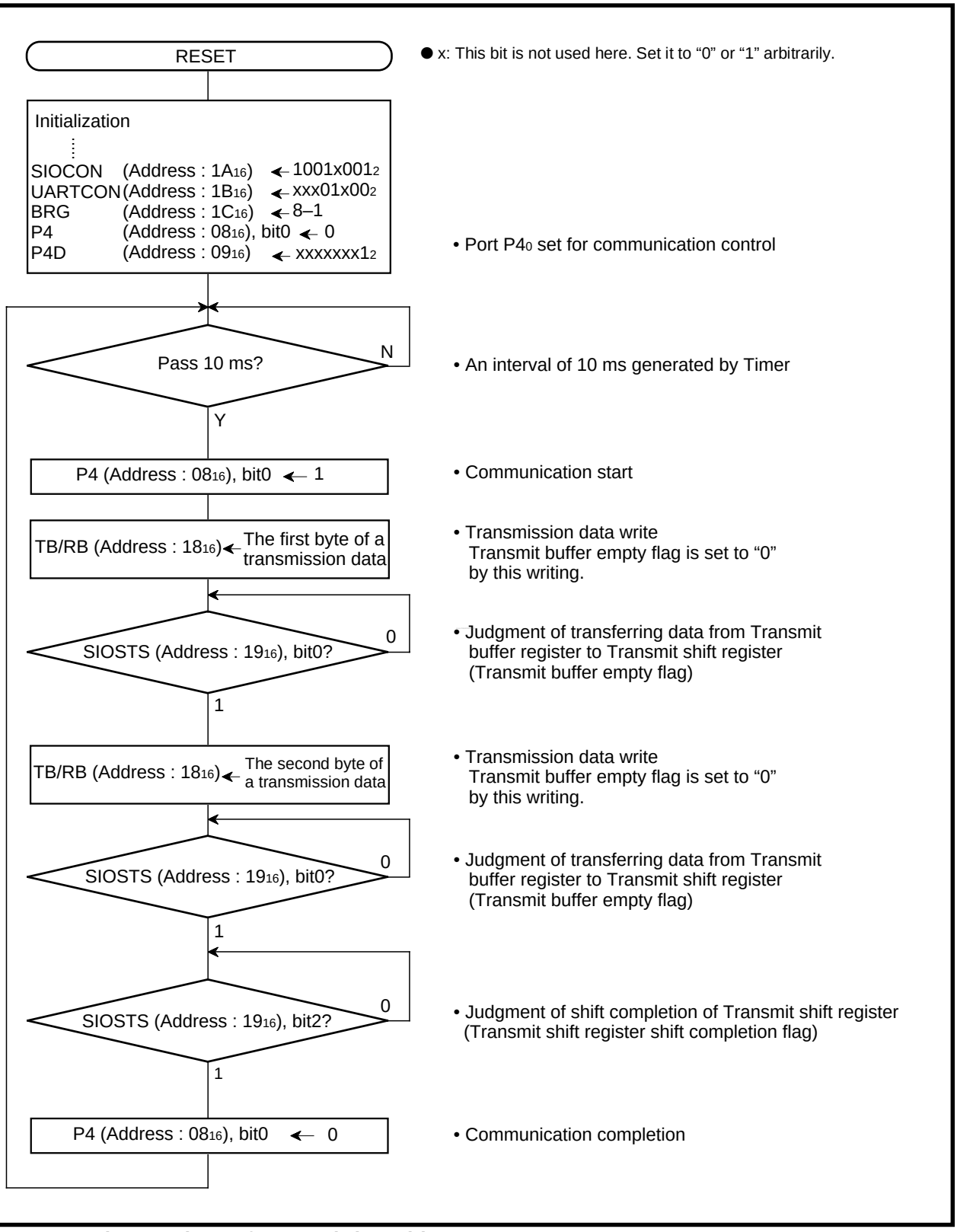


Fig. 2.4.41 Control procedure of transmitting side

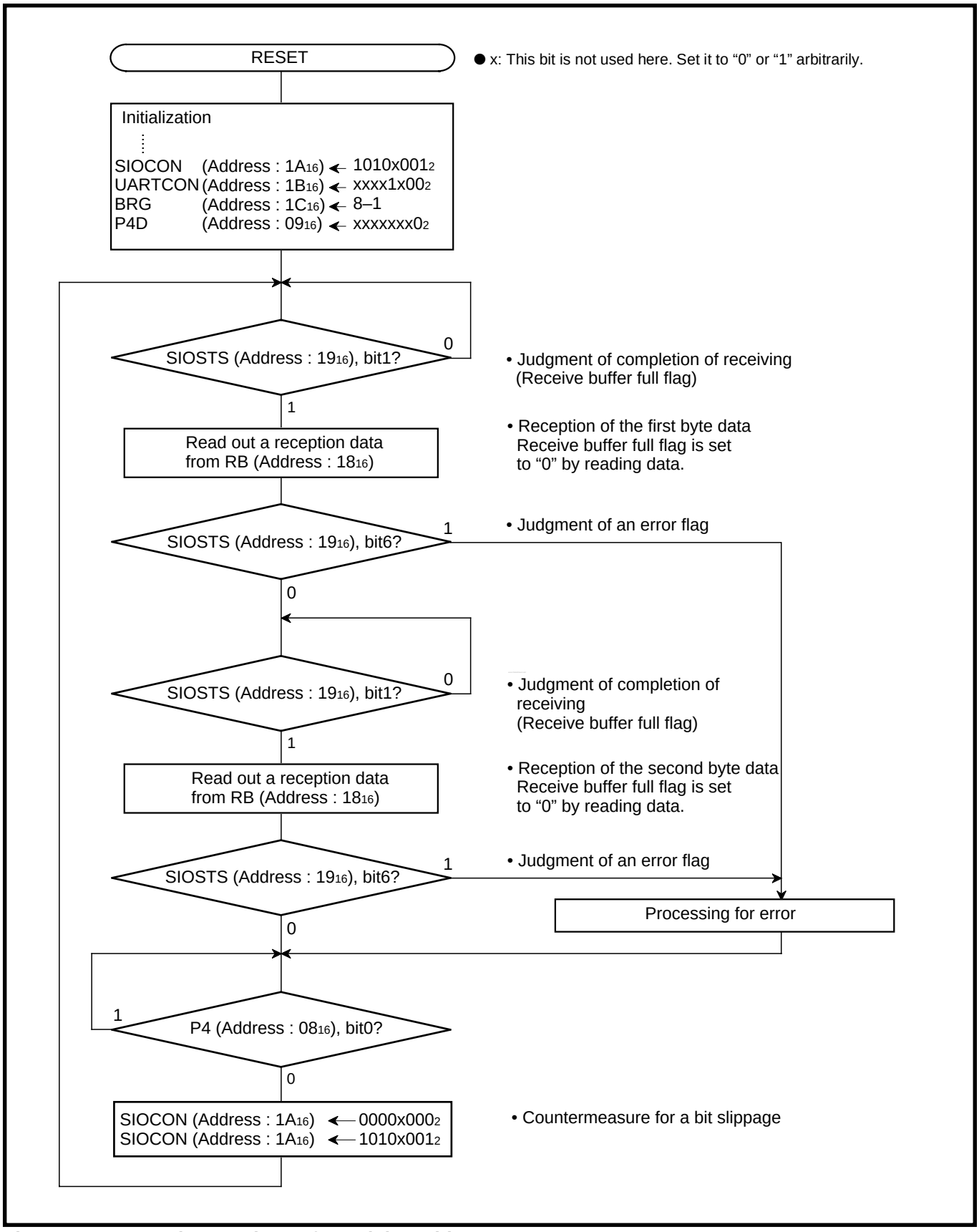


Fig. 2.4.42 Control procedure of receiving side

2.4.6 Notes on serial I/O

(1) Notes when selecting clock synchronous serial I/O (Serial I/O1)

① **Stop of transmission operation**
Clear the serial I/O1 enable bit and the transmit enable bit to “0” (Serial I/O1 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/O1 disabled), the internal transmission is running (in this case, since pins TxD, RxD, 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 TxD pin and an operation failure occurs.

② **Stop of receive operation**
Clear the receive enable bit to “0” (receive disabled), or clear the serial I/O1 enable bit to “0” (Serial I/O1 disabled).

③ **Stop of transmit/receive operation**
Clear the transmit enable bit and receive enable bit to “0” simultaneously (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/O1 disabled) (refer to (1) ①).

(2) Notes when selecting clock asynchronous serial I/O (Serial I/O1)

① **Stop of transmission operation**
Clear the transmit enable bit to “0” (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/O1 disabled), the internal transmission is running (in this case, since pins TxD, RxD, SCLK1, and 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 TxD pin and an operation failure occurs.

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).

● **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/O1 disabled), the internal transmission is running (in this case, since pins TxD, RxD, SCLK1, and 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 TxD pin and an operation failure occurs.

Only receive operation is stopped.
Clear the receive enable bit to “0” (receive disabled).

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

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 (Serial I/O1)

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”.

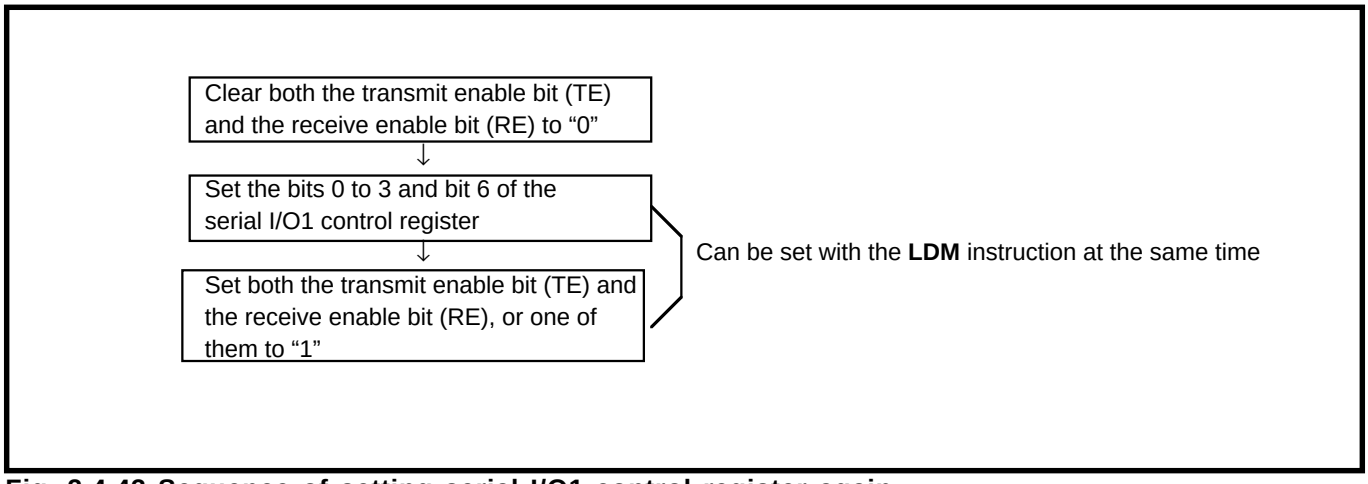


Fig. 2.4.43 Sequence of setting serial I/O1 control register again

(5) Data transmission control with referring to transmit shift register completion flag (Serial I/O1)

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 (Serial I/O1)

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 the transmit data to the transmit buffer register at "H" of the SCLK1 input level.

(7) Transmit interrupt request when transmit enable bit is set (Serial I/O1)

When the transmit interrupt is used, set the transmit interrupt enable bit to transmit enabled as shown in the following sequence.

① Set the interrupt enable bit to "0" (disabled) with CLB instruction.
② Prepare serial I/O for transmission/reception.
③ Set the interrupt request bit to "0" with CLB instruction after 1 or more instruction has been executed.
④ Set the interrupt enable bit to "1" (enabled).

● **Reason**

When the transmission enable bit is set to "1", the transmit buffer empty flag and transmit shift register completion flag are set to "1". The interrupt request is generated and the transmission interrupt bit is set regardless of which of the two timings listed below is selected as the timing for the transmission interrupt to be generated.

- Transmit buffer empty flag is set to "1"
- Transmit shift register completion flag is set to "1"

(8) Transmit data writing (Serial I/O2)

In the clock synchronous serial I/O, when selecting an external clock as synchronous clock, write the transmit data to the serial I/O2 register (serial I/O shift register) at "H" of the transfer clock input level.

2.5 PWM

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

2.5.1 Memory map

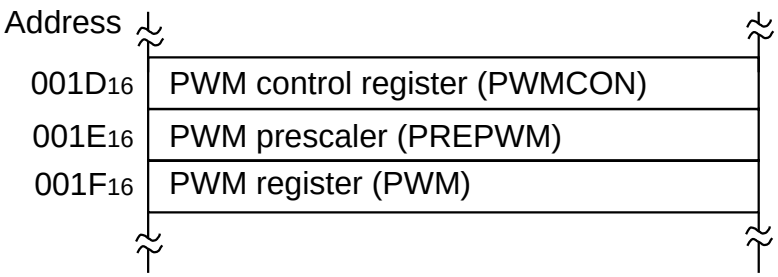


Fig. 2.5.1 Memory map of registers relevant to PWM

2.5.2 Relevant registers

PWM control register

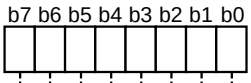
b7 b6 b5 b4 b3 b2 b1 b0

PWM control register (PWMCON: address $1D_{16}$)

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 : $f(X_{IN})$ 1 : $f(X_{IN})/2$	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. 2.5.2 Structure of PWM control register

PWM prescaler

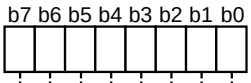


PWM prescaler
(PREPWM: address $1E_{16}$)

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	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

Fig. 2.5.3 Structure of PWM prescaler

PWM register



PWM register
(PWM: address $1F_{16}$)

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	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

Fig. 2.5.4 Structure of PWM register

2.5.3 PWM output circuit application example

<Motor control>

Outline : The rotation speed of the motor is controlled by using PWM (pulse width modulation) output.

Figure 2.5.5 shows a connection diagram ; Figures 2.5.6 shows PWM output timing, and Figure 2.5.7 shows a setting of the related registers.

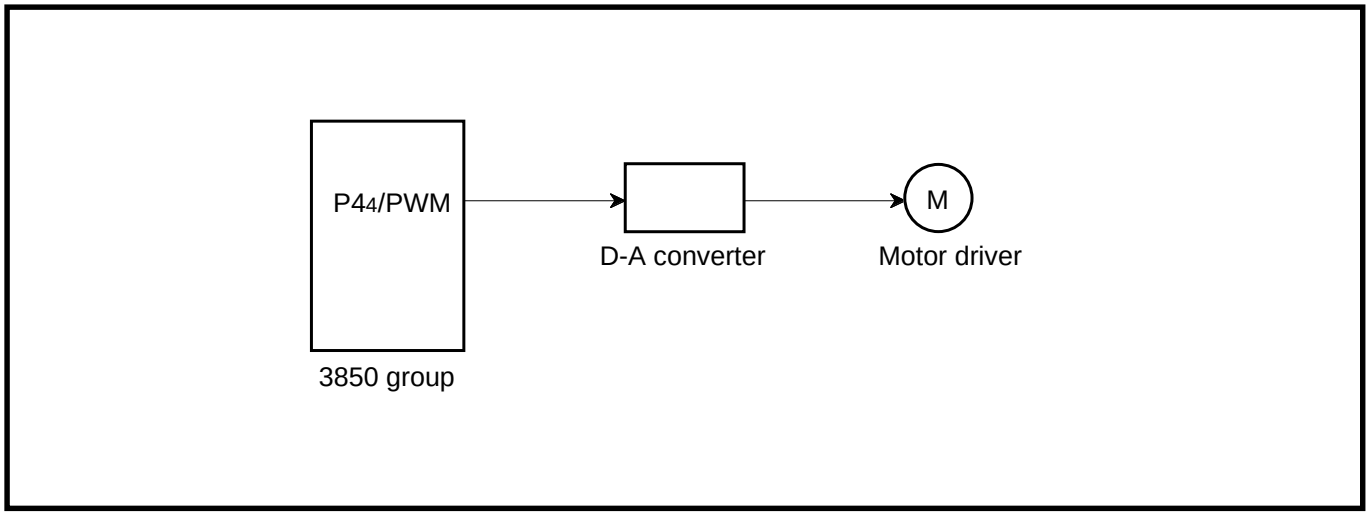


Fig. 2.5.5 Connection diagram

Specifications : • Motor is controlled by using the PWM output function of 8-bit resolution.
• Clock f(XIN) = 5.0 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.

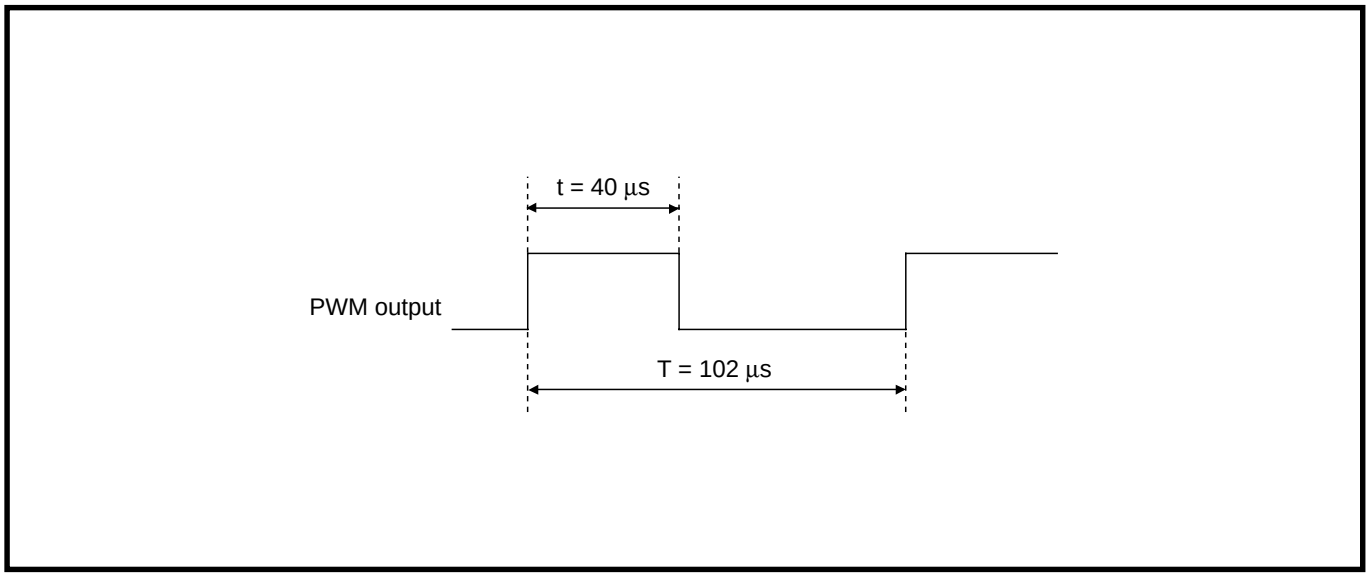


Fig. 2.5.6 PWM output timing

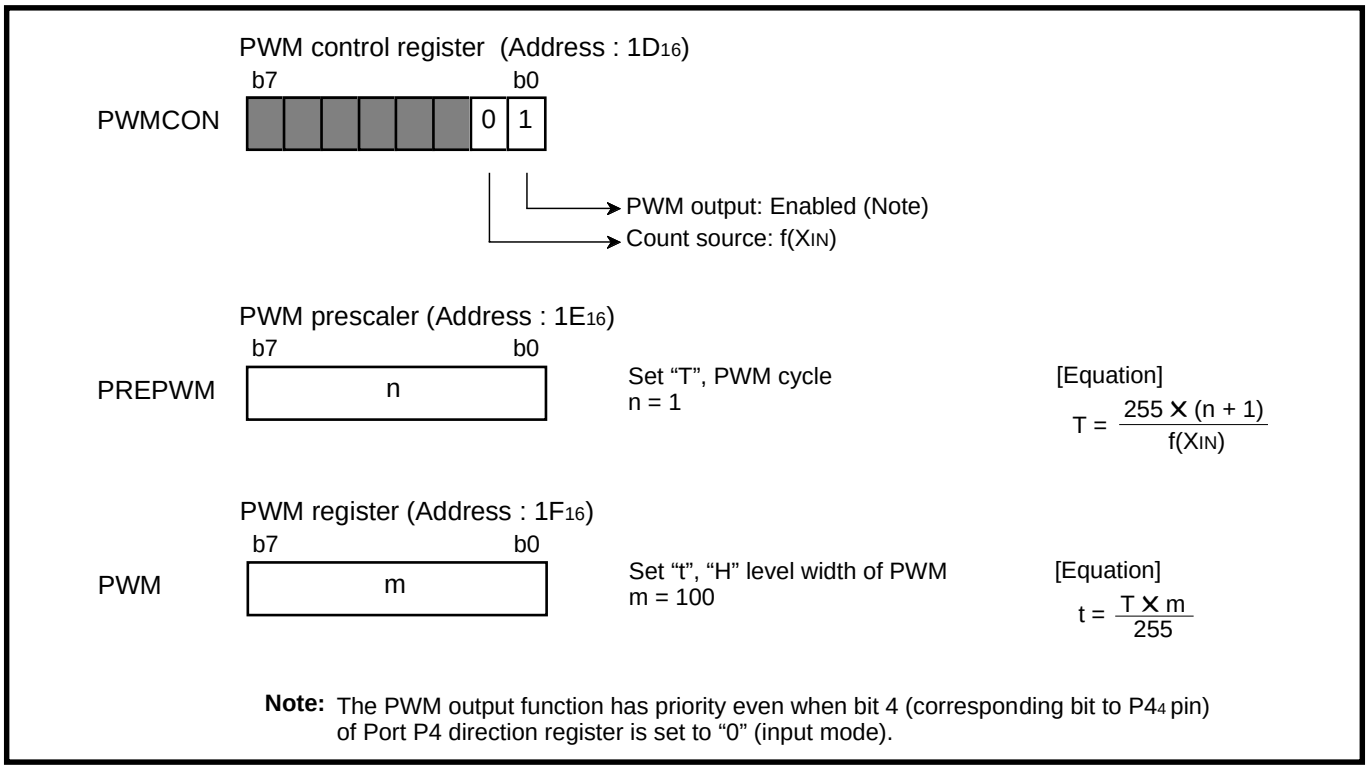


Fig. 2.5.7 Setting of relevant registers

<About PWM output>

1. Set the PWM function enable bit to "1" : The P44/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 P44/PWM pin is used as the port P44.
   Thus, when fixing the output level, take the following procedure:
   (1) Write an output value to bit 6 of the port P4 register.
   (2) Write "00010000$_2$" to the port P4 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.

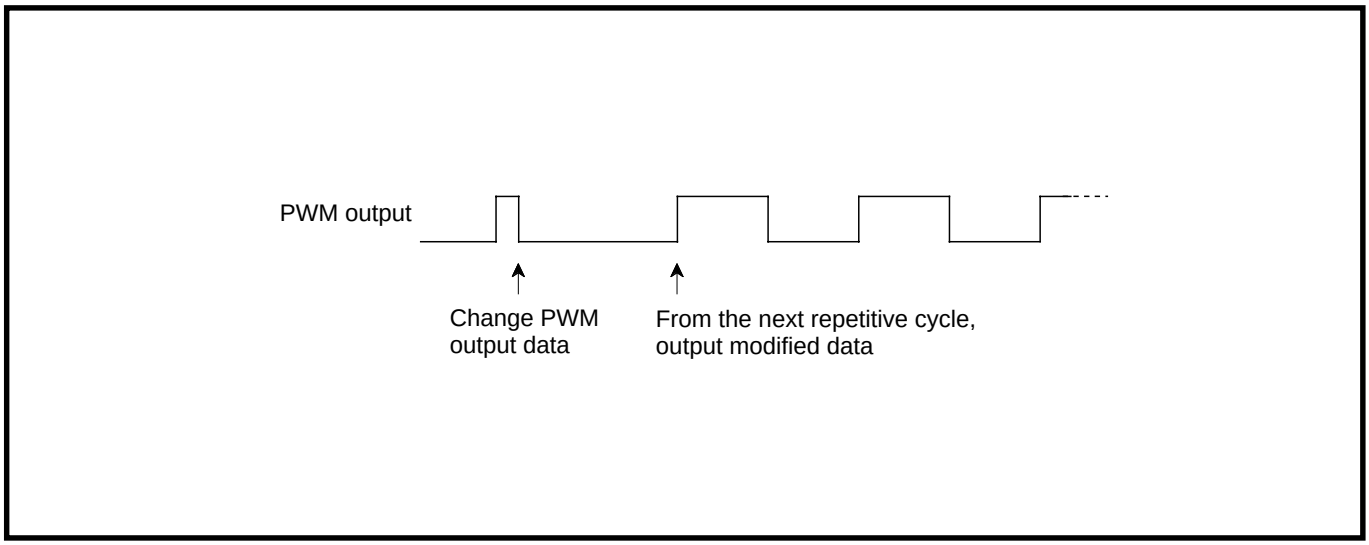


Fig. 2.5.8 PWM output

Control procedure : By setting the related registers as shown by Figure 2.5.7, 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 2.5.9 shows control procedure.

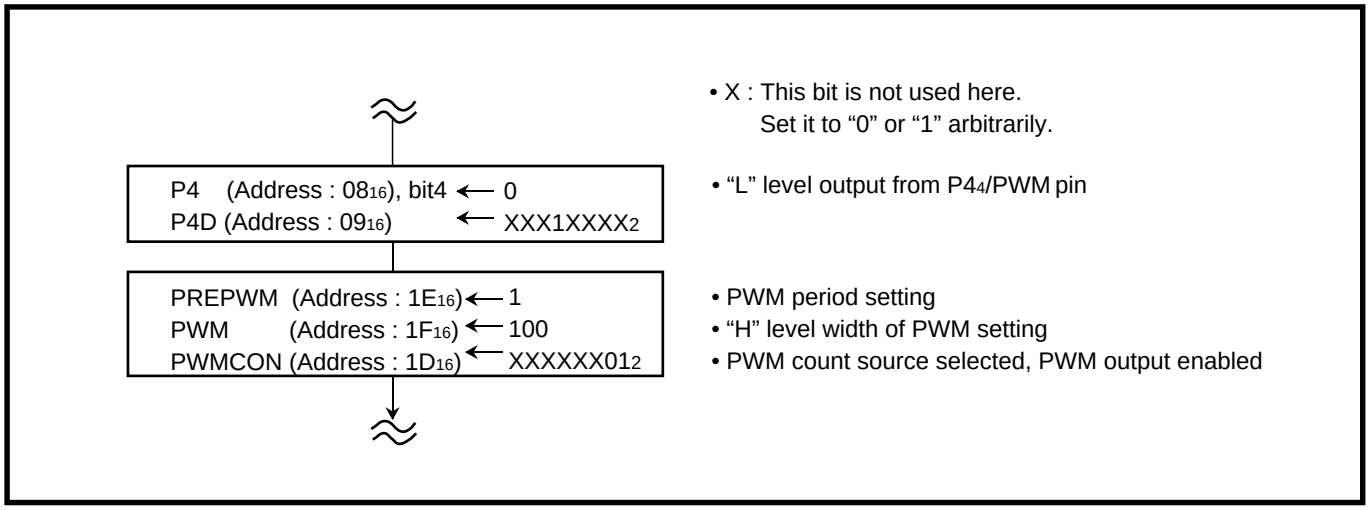


Fig. 2.5.9 Control procedure

2.5.4 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:

$$\frac{n + 1}{2 \cdot f(X_{IN})} \text{ sec.} \quad \text{(Count source selection bit = 0, where n is the value set in the prescaler)}$$

$$\frac{n + 1}{f(X_{IN})} \text{ sec.} \quad \text{(Count source selection bit = 1, where n is the value set in the prescaler)}$$

2.6 A-D converter

This paragraph explains the registers setting method and the notes relevant to the A-D converter.

2.6.1 Memory map

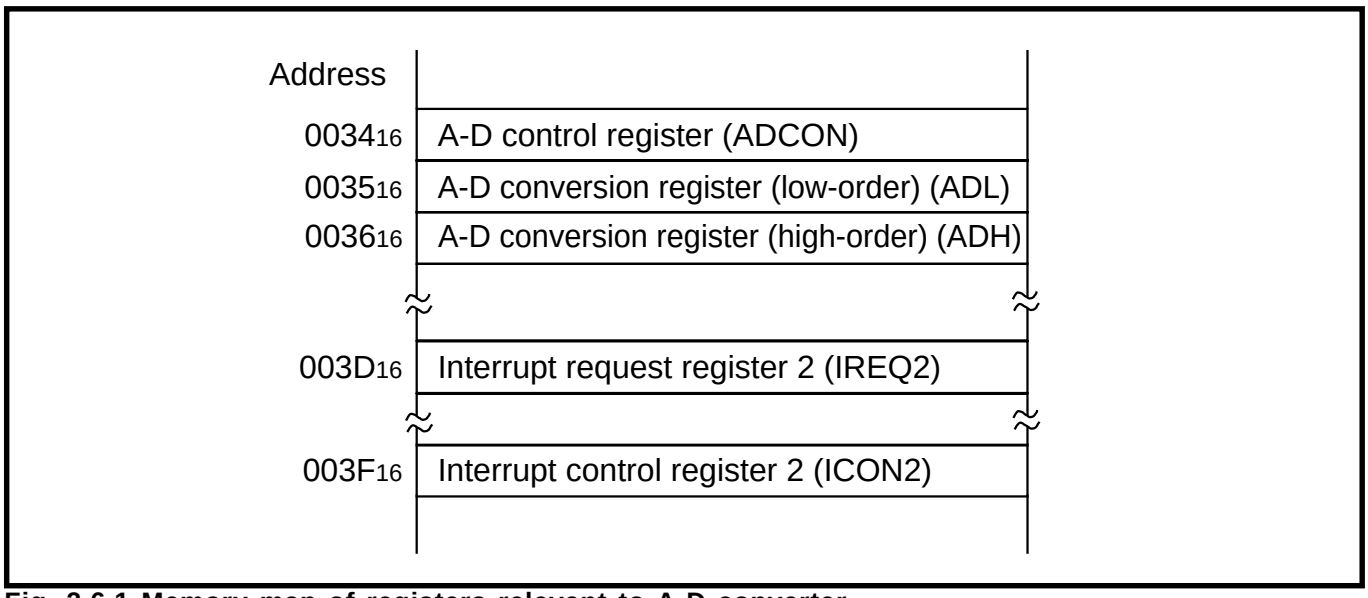


Fig. 2.6.1 Memory map of registers relevant to A-D converter

2.6.2 Relevant registers

A-D control register

b7 b6 b5 b4 b3 b2 b1 b0

A-D control register
(ADCON: address 34₁₆)

b	Name	Functions	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0: P3₀/AN₀ 0 0 1: P3₁/AN₁ 0 1 0: P3₂/AN₂ 0 1 1: P3₃/AN₃ 1 0 0: P3₄/AN₄	0	○	○
1			0	○	○
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	AD conversion completion bit	0: Conversion in progress 1: Conversion completed	1	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
6			0	○	×
7			0	○	×

Fig. 2.6.2 Structure of A-D control register

2.6 A-D converter

A-D conversion register (high-order)

b7 b6 b5 b4 b3 b2 b1 b0

A-D conversion register (high-order)
(ADH: address 36_{16})

b	Functions	At reset	R	W
0	This is A-D conversion result stored bits. This is read exclusive register. b7 10-bit read 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		0	○	×

Fig. 2.6.3 Structure of A-D conversion register (high-order)

A-D conversion register (low-order)

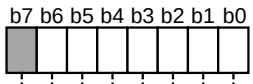
b7 b6 b5 b4 b3 b2 b1 b0

A-D conversion register (low-order)
(ADL: address 35_{16})

b	Functions	At reset	R	W
0	This is A-D conversion result stored bits. This is read exclusive register. b7 8-bit read b0 b9 b8 b7 b6 b5 b4 b3 b2 b7 10-bit read 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. 2.6.4 Structure of A-D conversion register (low-order)

Interrupt request register 2



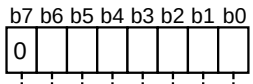
Interrupt request register 2
(IREQ2 : address $3D_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	Timer 2 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 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	$CNTR_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	$CNTR_1$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	A-D converter 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. 2.6.5 Structure of Interrupt request register 2

Interrupt control register 2



Interrupt control register 2
(ICON2 : address $3F_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Timer 2 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 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	$CNTR_0$ interrupt enable bit	0 : Interrupt disabled	0	○	○
5	$CNTR_1$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	A-D converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.6.6 Structure of Interrupt control register 2

2.6 A-D converter

2.6.3 A-D converter application examples

(1) Conversion of analog input voltage

Outline : The analog input voltage input from a sensor is converted to digital values.

Figure 2.6.7 shows a connection diagram, and Figure 2.6.8 shows the relevant registers setting.

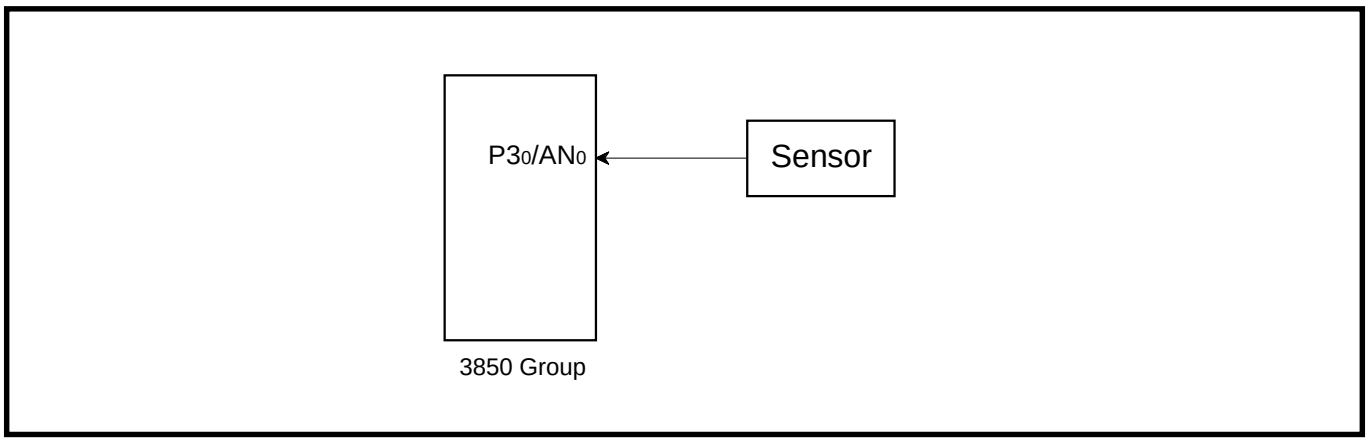


Fig. 2.6.7 Connection diagram

Specifications : •The analog input voltage input from a sensor is converted to digital values.
•$P3_0/AN_0$ pin is used as an analog input pin.

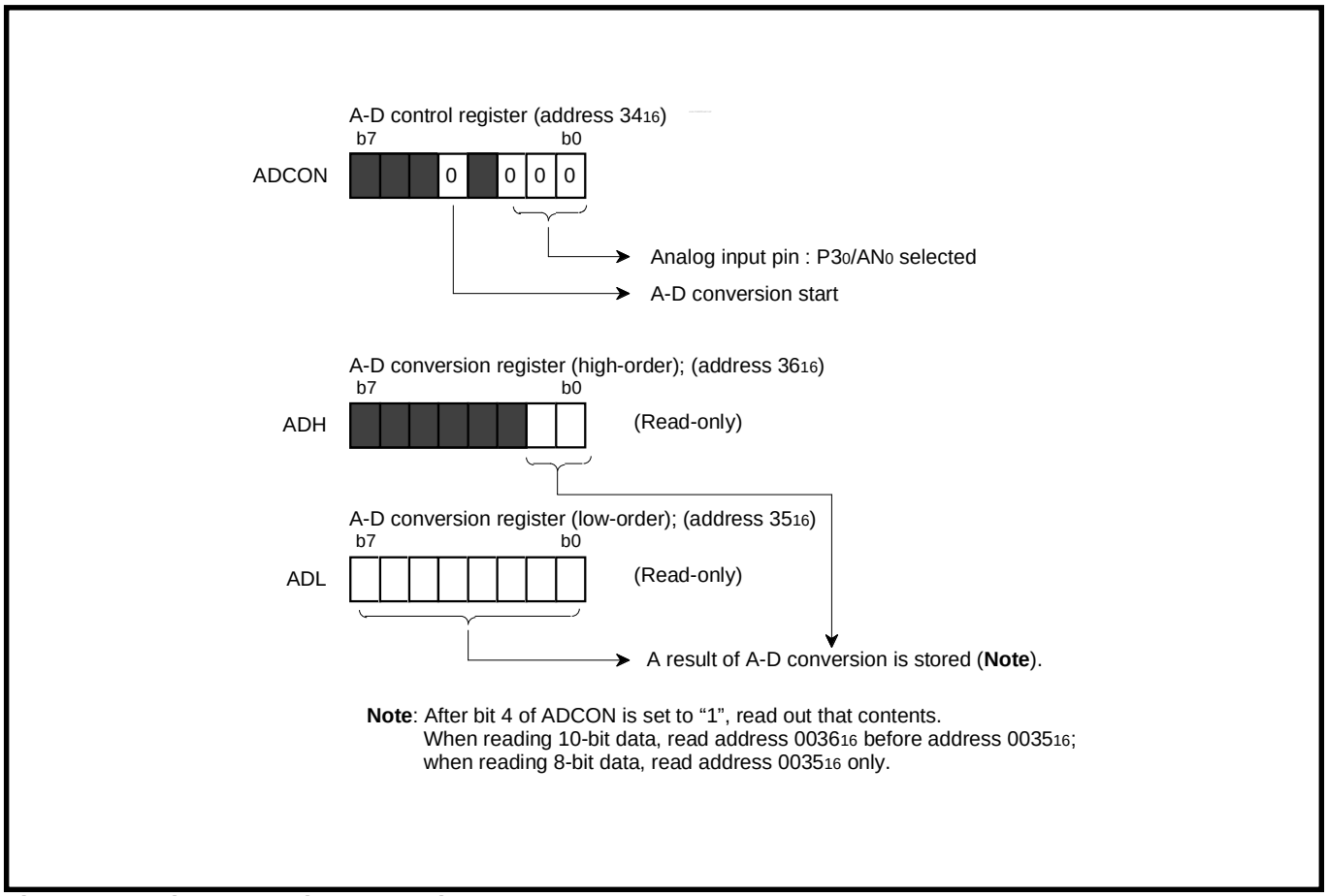


Fig. 2.6.8 Relevant registers setting

An analog input signal from a sensor is converted to the digital value according to the relevant registers setting shown by Figure 2.6.8. Figure 2.6.9 shows the control procedure for 8-bit read, and Figure 2.6.10 shows the control procedure for 10-bit read.

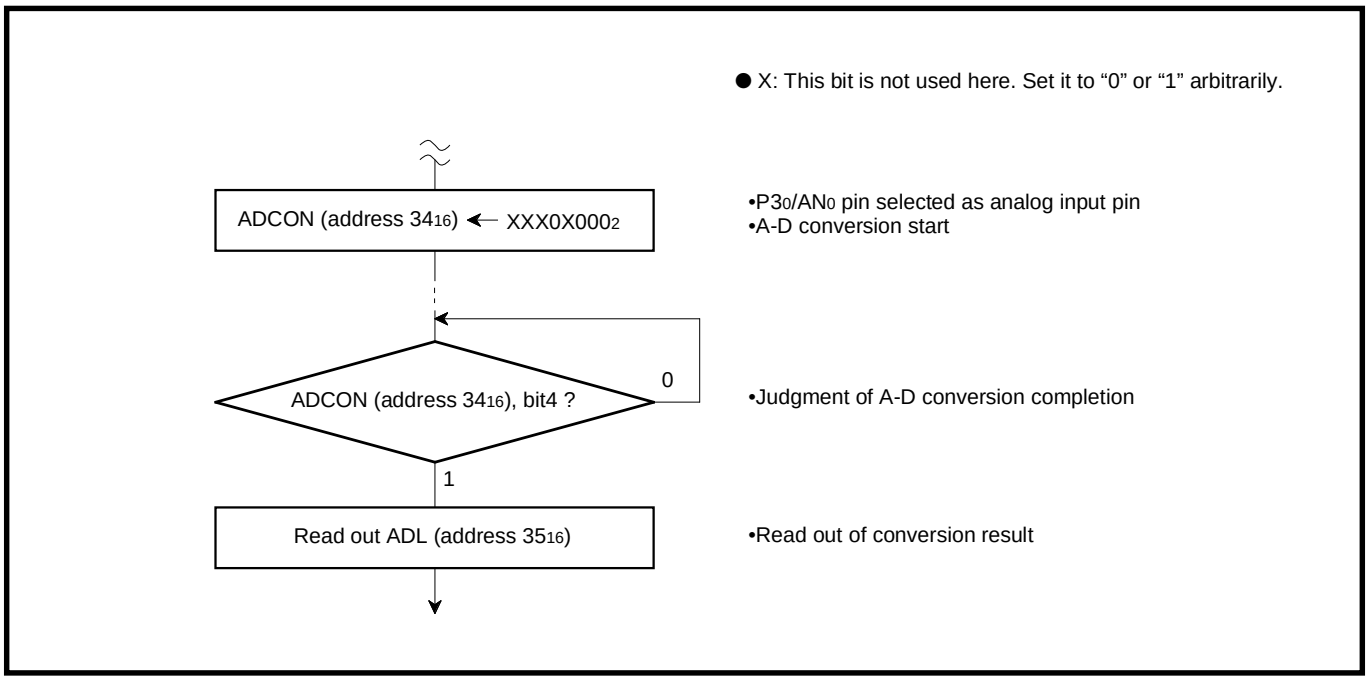


Fig. 2.6.9 Control procedure for 8-bit read

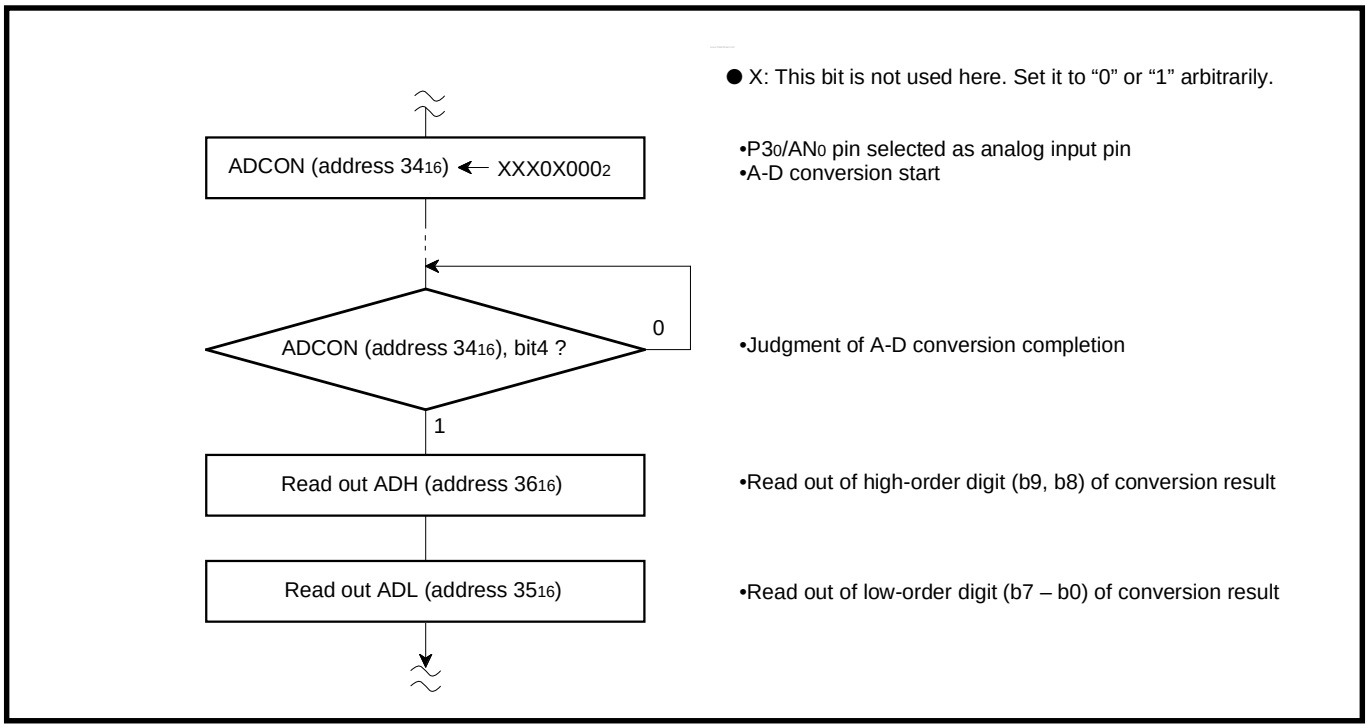


Fig. 2.6.10 Control procedure for 10-bit read

2.6 A-D converter

2.6.4 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 pin. 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(X_{IN}) is 500 kHz or more in middle-/high-speed mode.
• Do not execute the **STP** instruction.
• When the A-D converter is operated at low-speed mode, f(X_{IN}) do not have the lower limit of frequency, because of the A-D converter has a built-in self-oscillation circuit.

2.7 Watchdog timer

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

2.7.1 Memory map

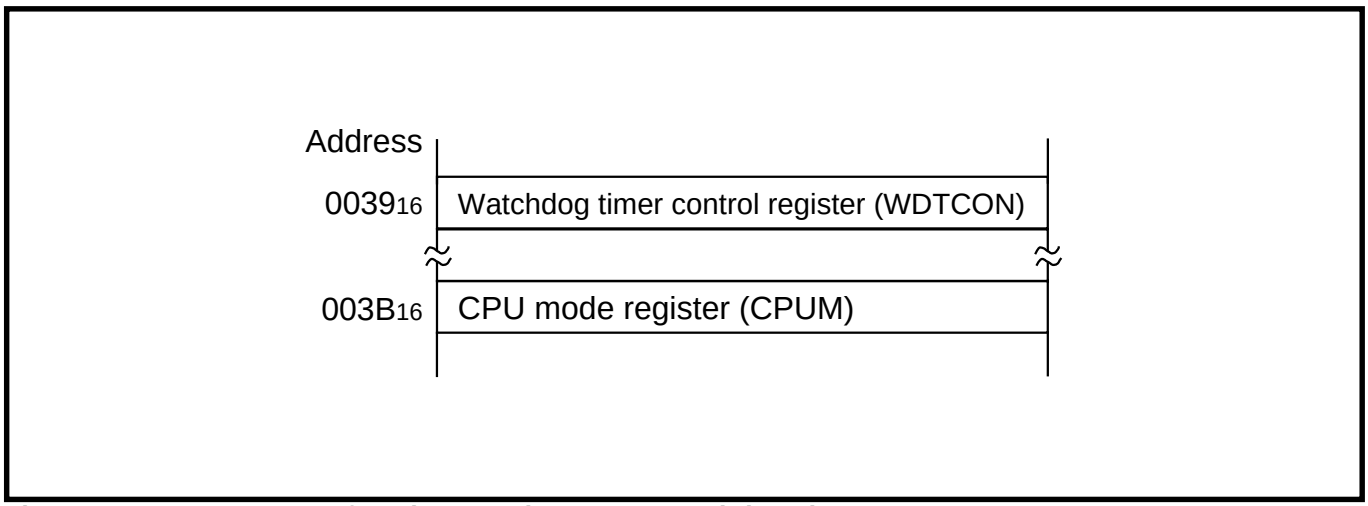


Fig. 2.7.1 Memory map of registers relevant to watchdog timer

2.7.2 Relevant registers

Watchdog timer control register

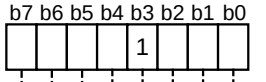
b7 b6 b5 b4 b3 b2 b1 b0

Watchdog timer control register
(WDTCON: address 39₁₆)

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. 2.7.2 Structure of Watchdog timer control register

CPU mode register



CPU mode register
(CPUM: address $3B_{16}$)

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. 2.7.3 Structure of CPU mode register

2.7.3 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:
- An underflow of watchdog timer H is judged to be program runaway, and the microcomputer is returned to the reset status.
- Before the watchdog timer underflows, "0" is set into bits 6 and 7 of the watchdog timer control register at every cycle in a main routine.
- 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.

Figure 2.7.4 shows a watchdog timer connection and division ratio setting; Figure 2.7.5 shows the relevant registers setting; Figure 2.7.6 shows the control procedure.

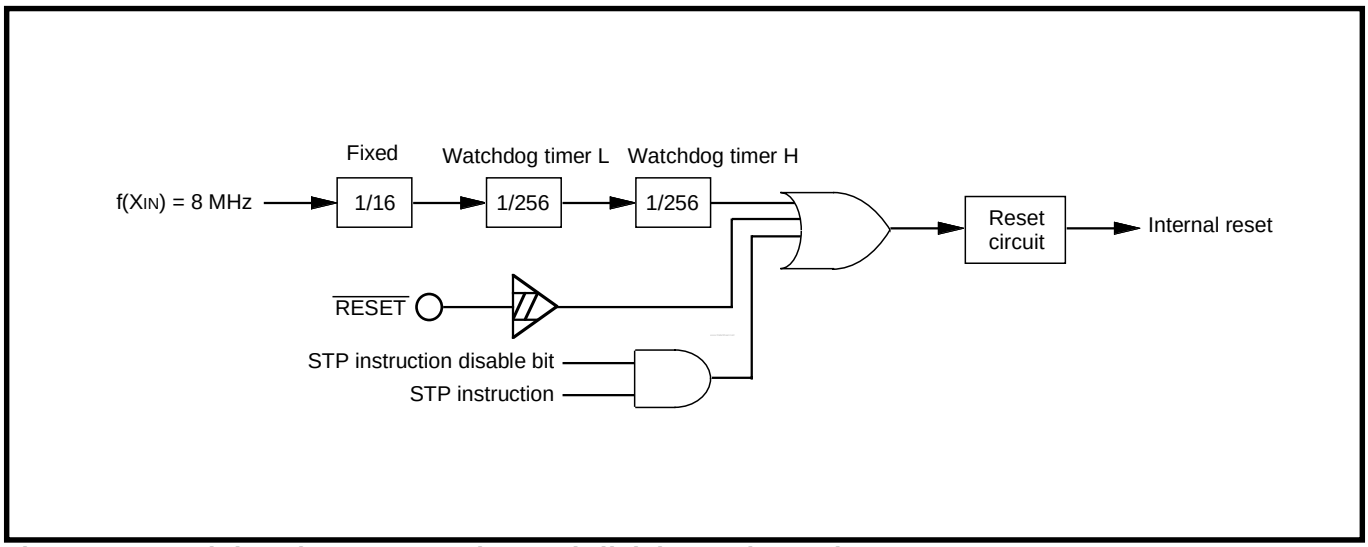


Fig. 2.7.4 Watchdog timer connection and division ratio setting

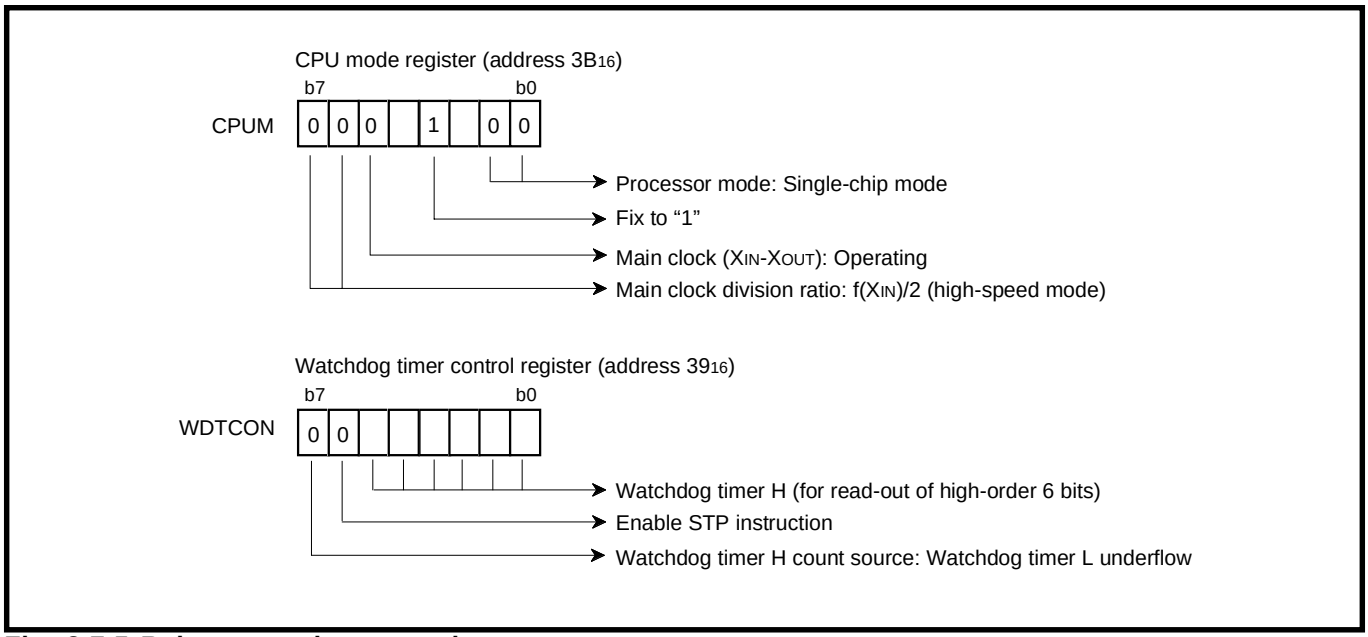


Fig. 2.7.5 Relevant registers setting

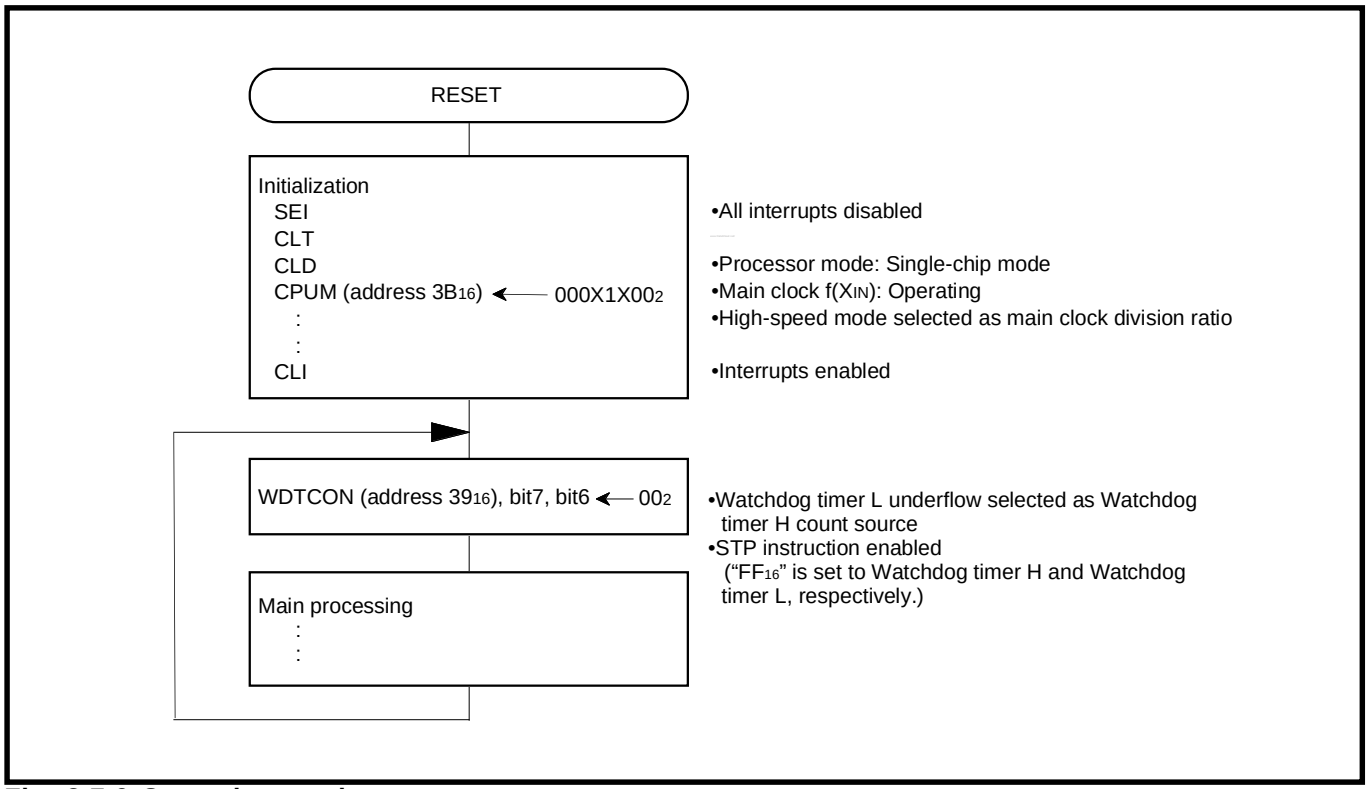


Fig. 2.7.6 Control procedure

2.7.4 Notes on watchdog timer

- Make sure that the watchdog timer 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.

2.8 Reset

2.8.1 Connection example of reset IC

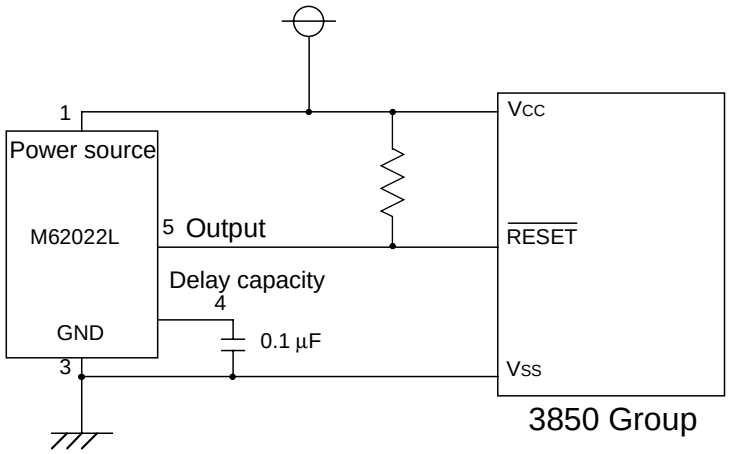


Fig. 2.8.1 Example of poweron reset circuit

Figure 2.8.2 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.

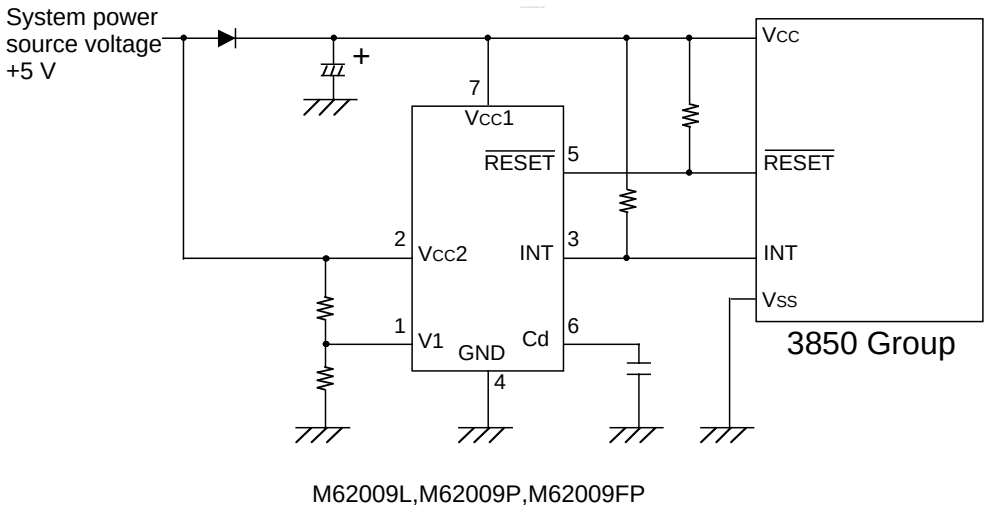


Fig. 2.8.2 RAM backup system

2.8.2 Notes on $\overline{\text{RESET}}$ pin

(1) 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) Reset release after power on

When releasing the reset after power on, such as power-on reset, release reset after X_{IN} passes more than 20 cycles in the state where the power supply voltage is 2.7 V or more and the X_{IN} oscillation is stable.

● **Reason**

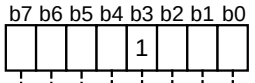
To release reset, the $\overline{\text{RESET}}$ pin must be held at an “L” level for 20 cycles or more of X_{IN} in the state where the power source voltage is between 2.7 V and 5.5 V, and X_{IN} oscillation is stable.

2.9 Clock generating circuit

This paragraph explains how to set the registers relevant to the clock generating circuit and describes an application example.

2.9.1 Relevant registers

CPU mode register



CPU mode register
(CPUM: address $3B_{16}$)

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. 2.9.1 Structure of CPU mode register

2.9.2 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.

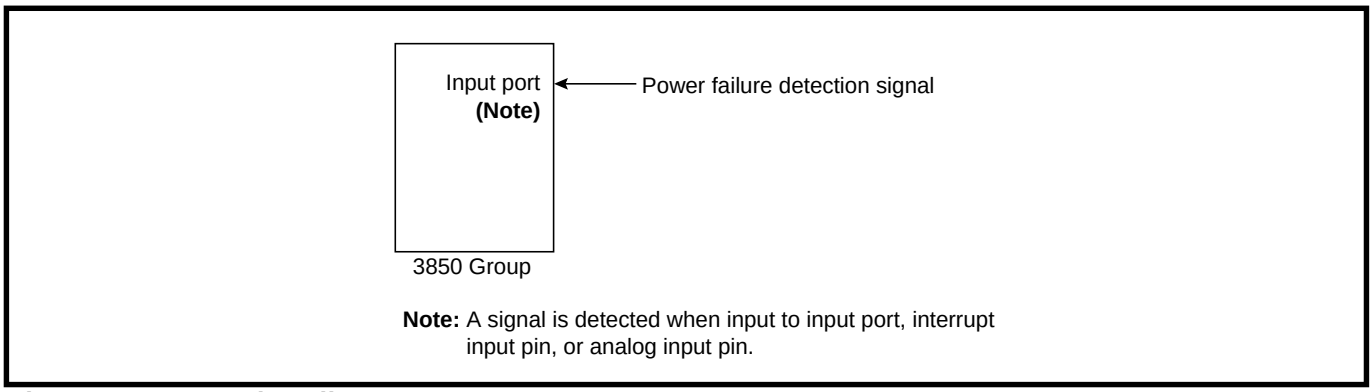


Fig. 2.9.2 Connection diagram

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: Stop V_{REF} current dissipation by terminating A-D conversion operation.

Figure 2.9.3 shows the status transition diagram during power failure and Figure 2.9.4 shows the setting of relevant registers.

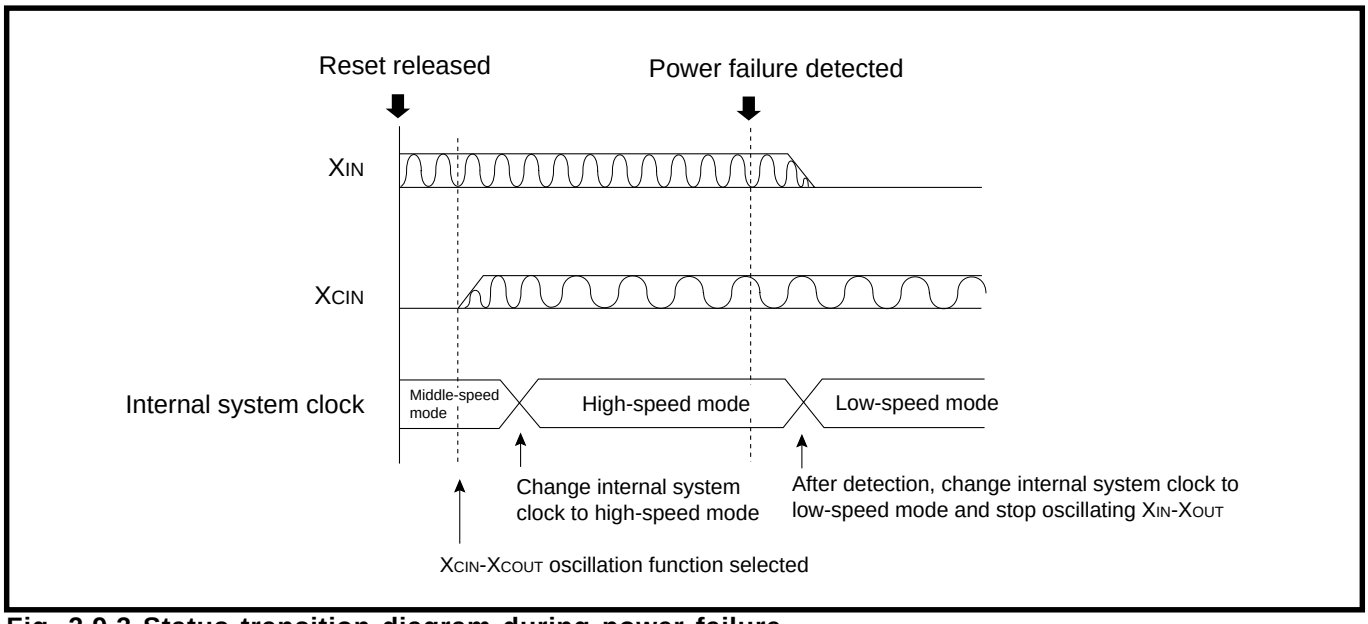
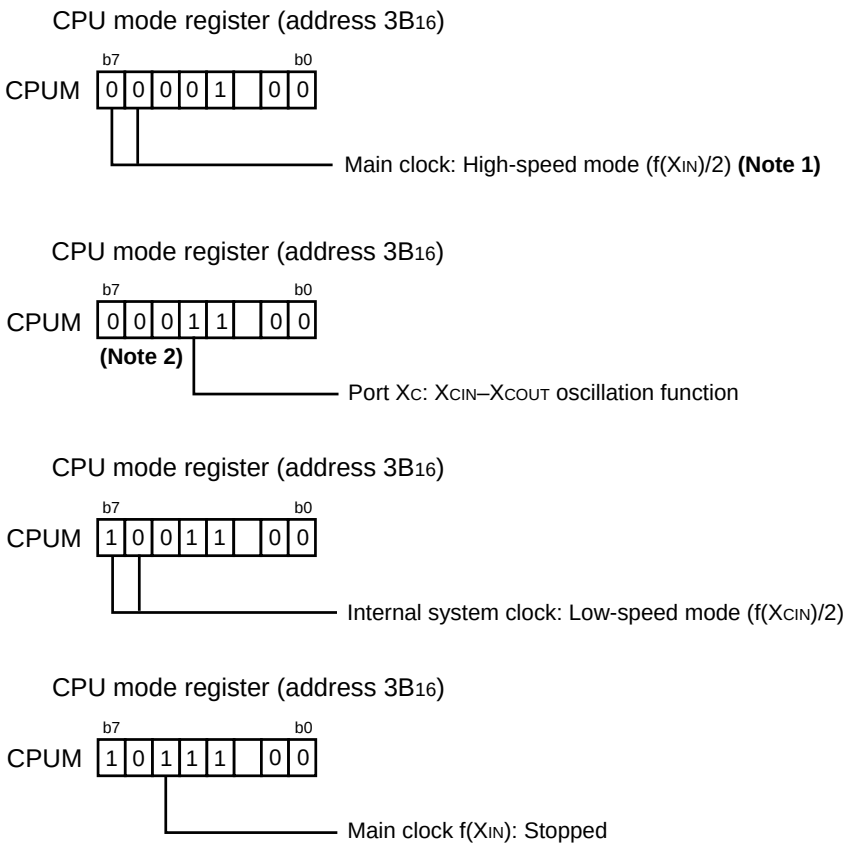


Fig. 2.9.3 Status transition diagram during power failure



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. 2.9.4 Setting of relevant registers

Control procedure: To prepare for a power failure, set the relevant registers in the order shown below.

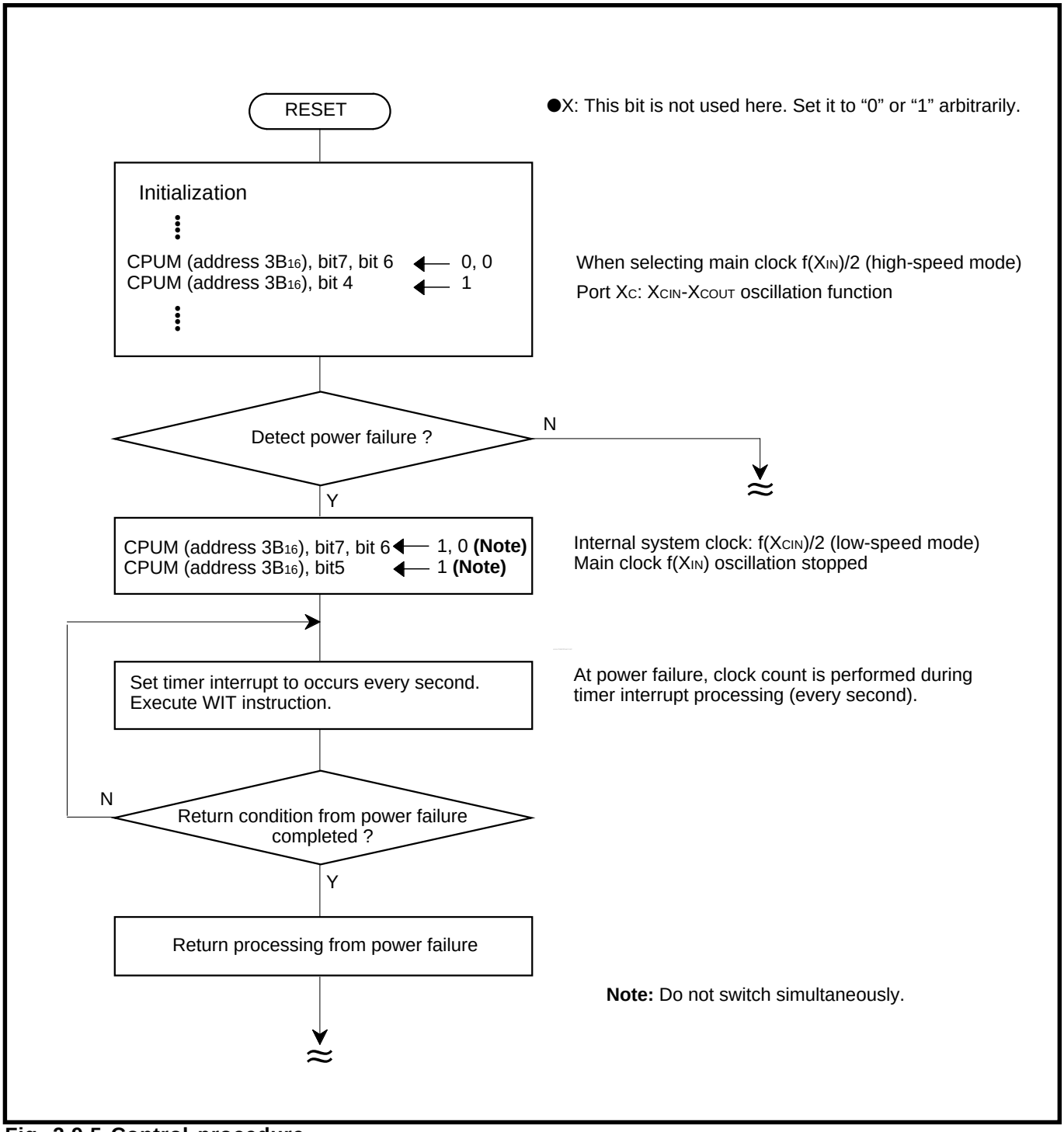


Fig. 2.9.5 Control procedure

2.10 Standby function

The 3850 group is provided with standby functions to stop the CPU by software and put the CPU into the low-power operation.

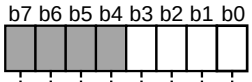
The following two types of standby functions are available.

•Stop mode using STP instruction

•Wait mode using WIT instruction

2.10.1 Relevant registers

MISRG



MISRG
(MISRG: address 38_{16})

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 (**Note 2**)	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 2**)	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: When automatic switch to middle-speed mode from low-speed mode occurs, the values of CPU mode register ($003B_{16}$) change.

Fig. 2.10.1 Structure of MISRG

2.10 Standby function

2.10.2 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 2.10.1 shows the state in the stop mode.

Table 2.10.1 State in stop mode

Item	State in stop mode
Oscillation	Stopped.
CPU	Stopped.
Internal clock ϕ	Stopped at "H" level.
I/O ports P0–P4	Retains the state at the STP instruction execution.
Timer	Stopped. (Timers 1, 2, X, Y) However, Timers X and Y can be operated in the event counter mode.
PWM	Stopped.
Watchdog timer	Stopped.
Serial I/O1, Serial I/O2	Stopped. However, these can be operated only when an external clock is selected.
A-D converter	Stopped.

(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.

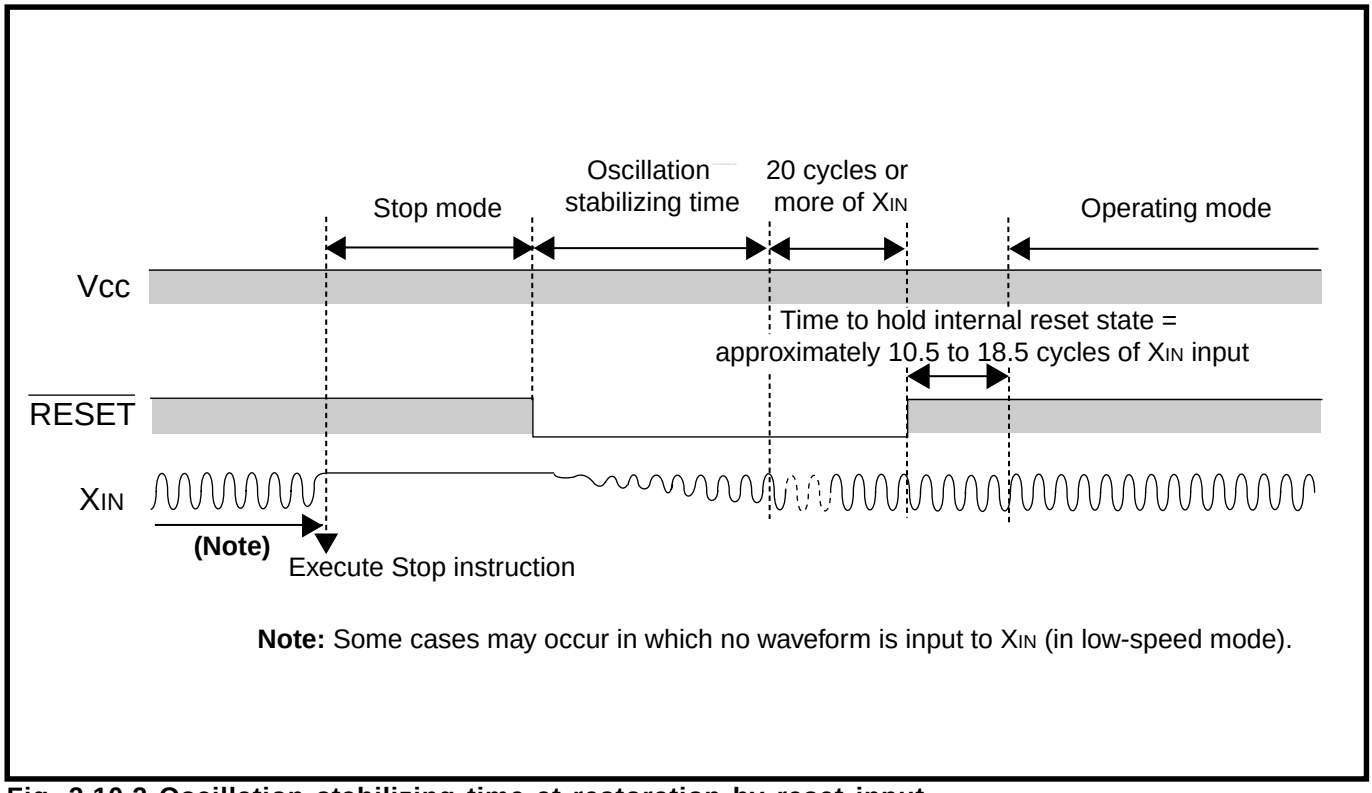
Oscillation is unstable when restarted. For this reason, time for stabilizing of oscillation (oscillation stabilizing time) **(Note)** is required. The input of the $\overline{\text{RESET}}$ pin should be held at the "L" level until oscillation stabilizes.

When the $\overline{\text{RESET}}$ pin is held at the "L" level for 20 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 2.10.2 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 "2.8 Reset".

Note: For the setting of oscillation stabilizing time, refer to MISRG (address 0038_{16}).



Note: Some cases may occur in which no waveform is input to X_{IN} (in low-speed mode).

Fig. 2.10.2 Oscillation stabilizing time at restoration by reset input

2.10 Standby function

■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_3
•$CNTR_0$, $CNTR_1$
•Serial I/O (1, 2) using an external clock
•Timer X, Y using an external event count

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)

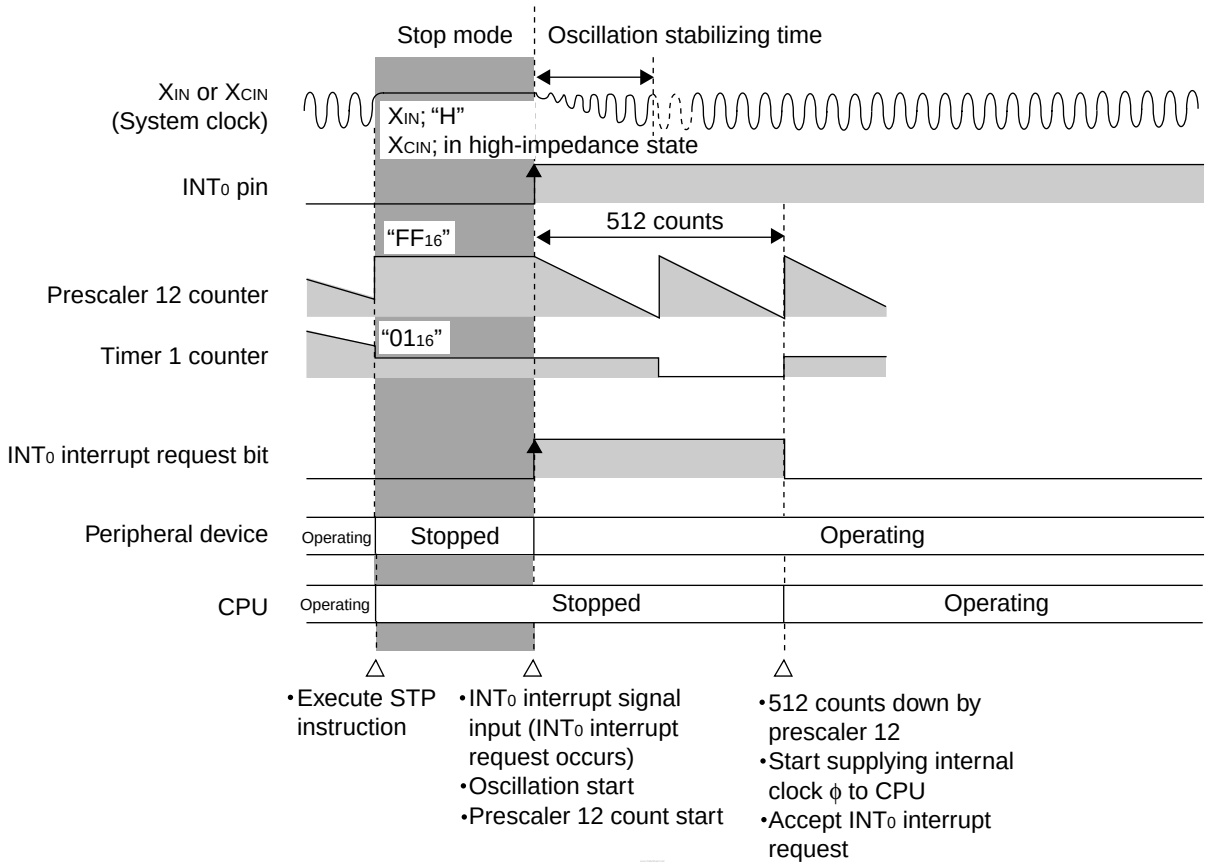
For more details concerning interrupts, refer to "2.2 Interrupts".

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 2.10.3 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.

●When restoring microcomputer from stop mode by INT_0 interrupt (rising edge selected)



Note: f(X_{IN})/16 or f(X_{CIN})/16 is input as the prescaler 12 count source.

Fig. 2.10.3 Execution sequence example at restoration by occurrence of INT_0 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 for approximately 8,000 cycles of the X_{IN} input is reserved at restoration from the stop mode. 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.

2.10 Standby function

2.10.3 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 2.10.2 shows the state in the wait mode.

Table 2.10.2 State in wait mode

Item	State in wait mode
Oscillation	Operating.
CPU	Stopped.
Internal clock ϕ	Stopped at "H" level.
I/O ports P0–P4	Retains the state at the WIT instruction execution.
Timer	Operating.
PWM	Operating.
Watchdog timer	Operating.
Serial I/O1, Serial I/O2	Operating.
A-D converter	Operating.

(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.

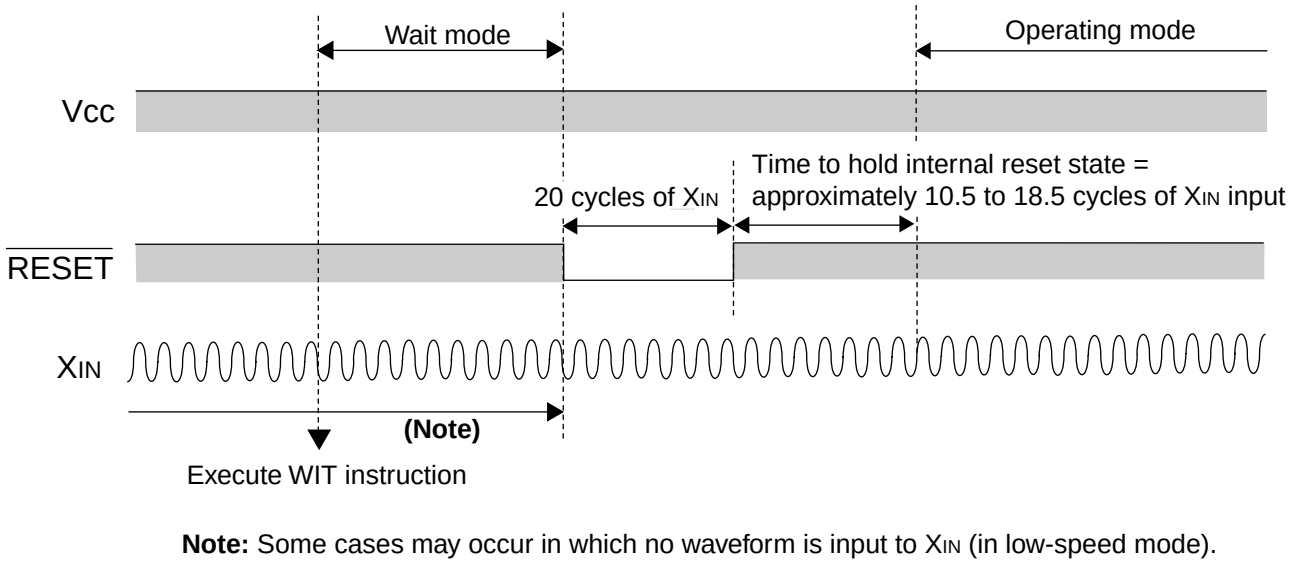
■Restoration by reset input

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 20 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 2.10.4 shows the reset input time.

For more details concerning reset, refer to "2.8 Reset".



Note: Some cases may occur in which no waveform is input to X_{IN} (in low-speed mode).

Fig. 2.10.4 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 WIT 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.

2.11 Flash memory mode

This paragraph explains the registers setting method and the notes relevant to the flash memory version.

2.11.1 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 "2.11.2 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

2.11.2 Memory map

M38507F8FP/SP have 32 Kbytes of built-in flash memory.

Figure 2.11.1 shows the memory map of the flash memory version.

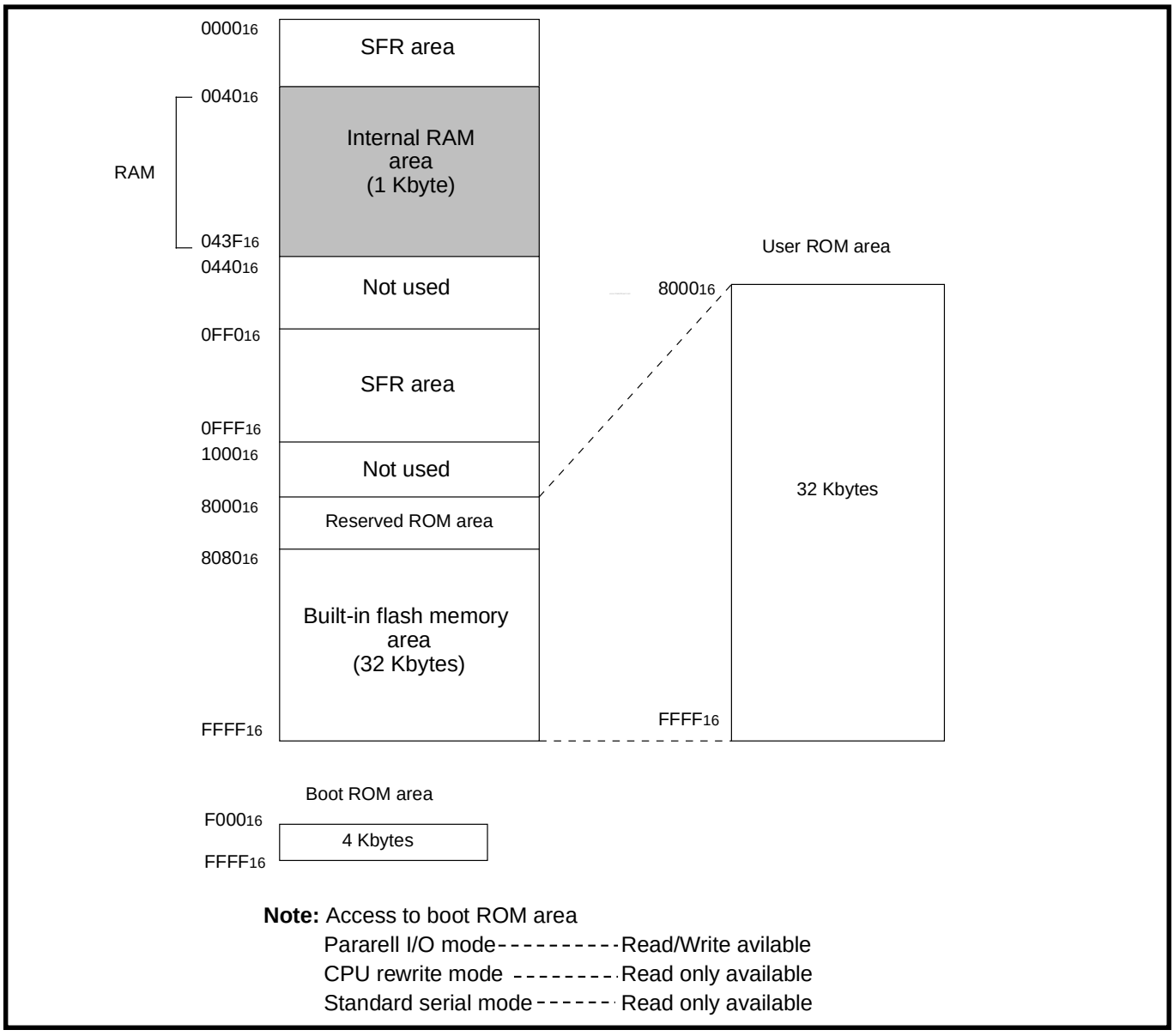


Fig. 2.11.1 Memory map of flash memory version for 3850 Group

2.11 Flash memory mode

2.11.3 Relevant registers

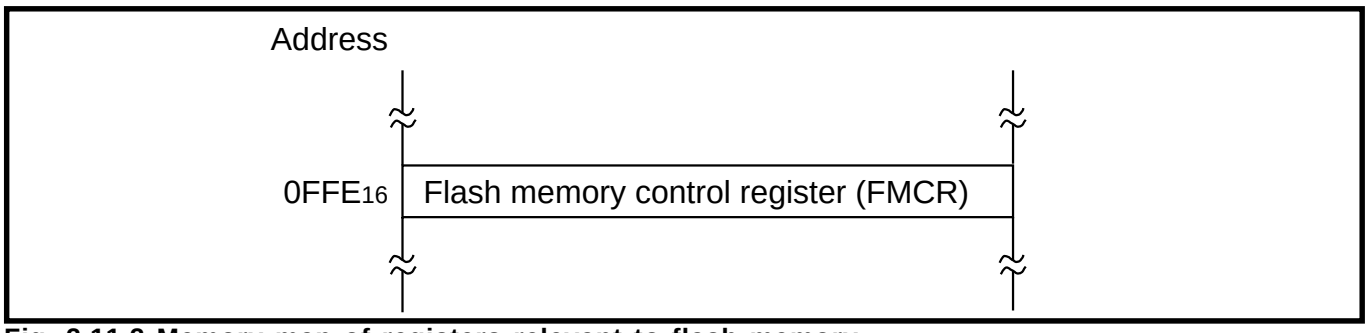


Fig. 2.11.2 Memory map of registers relevant to flash memory

Flash memory control register

b7 b6 b5 b4 b3 b2 b1 b0

Flash memory control register
(FMCR : address 0FFE16)

b	Name	Functions	At reset	R	W
0	RY/$\overline{BY}$ status flag	0 : Busy (being written or erased) 1 : Ready	1	○	×
1	CPU rewrite mode select bit **(Note 1)**	0 : Normal mode (Software commands invalid) 1 : CPU rewrite mode (Software commands acceptable)	0	○	○
2	CPU rewrite mode entry flag	0: Normal mode 1: CPU rewrite mode	0	○	×
3	Flash memory reset bit **(Note 2)**	0: Normal operation 1: Reset	0	○	○
4	User area/Boot area selection bit	0: User ROM area 1: Boot ROM area	0	○	○
5	Nothing is arranged for these bits. When write, set “0”. When these bits are read out, the contents are undefined.		Undefined	×	×
6			Undefined	×	×
7			Undefined	×	×

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: In order to perform flash memory reset by setting of this bit, set this bit to “1” in state of the CPU rewriting mode select bit = “1”, and then set to “0” to release the reset state.

Fig. 2.11.3 Structure of Flash memory control register

2.11.4 Parallel I/O mode

In the parallel I/O mode, program/erase to the built-in flash memory can be performed by a EPROM programmer (EFP-I).

The memory area of program/erase is from $0F000_{16}$ to $0FFFF_{16}$ (boot ROM area) or from 08000_{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 2.11.1 shows the setting of programmers when programming 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 2.11.1 Setting of programmers when parallel programming

Products	Parallel unit	Boot ROM area	User ROM area
M38507F8FP	EF3850F-42E	$0F000_{16}$ to $0FFFF_{16}$	08000_{16} to $0FFFF_{16}$
M38507F8SP	EF3850F-42S		

2.11.5 Standard serial I/O mode

Table 2.11.2 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 serial 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 2.11.2 Connection example to programmer when serial programming (4 wires)

Function	EFP-I (EF1SRP-01U)		3850 Group flash memory version	
	Signal name	EF1RP-01U side connector Line number	Pin name	Pin number
Transfer clock input	T_SCLK1	9	$P2_6/S_{CLK1}$	10
Serial data input	T_RXD	11	$P2_5$/TxD	11
Serial data output	T_TXD	10	$P2_4$/RxD	12
Transmit/Receive enable output	T_BUSY	12	$P2_7/CNTR_0/\overline{S_{RDY1}}$	9
5 V input	T_VPP	3	CNV_{SS}	15
Reset input	T_RESET	14	$\overline{RESET}$ (**Note 1**)	18
Target board power source monitor input	T_VDD **(Note 2)**	4	V_{CC} (**Note 2**)	1
GND	GND **(Note 3)**	1, 2, 15, 16	V_{SS}, AV_{SS} (**Note 3**)	21, 3

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.

2.11 Flash memory mode

2.11.6 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 the internal 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, erase all block, 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. As for the control example, refer to "2.11.7 (2) Control example in the CPU rewrite mode".

[Beginning procedure]

① Apply 5 V±10 % to the CNV_{SS}/V_{PP} 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.
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 Apply 5 V±10 % to the CNV_{SS}/V_{PP} pin (in single-chip mode).
⑥ 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.
⑦ Read the CPU rewrite mode entry flag (bit 2 of address $0FFE_{16}$) to confirm that the CPU rewrite mode is set to "1".
⑧ Flash memory operations are executed by using software commands.

Note: 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 command or set the flash memory reset bit (bit 3 of address $0FFE_{16}$).
② Set the CPU rewrite mode select bit (bit 0 of address $0FFE_{16}$) to "0".

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 $1E_{16}$) 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”, 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.

2.11 Flash memory mode

2.11.7 Flash memory mode application examples

The control pin processing example on the system board in the serial I/O mode and the control example in the CPU rewrite mode are described below.

(1) Control pin connection example on the system board in serial I/O mode

As shown in Figure 2.11.4, in the serial I/O mode, the built-in flash memory can be rewritten with the microcomputer mounted on board. Connection examples of control pins ($P2_4$/RxD, $P2_5$/TxD, $P2_6/S_{CLK1}$, $P2_7/\overline{S_{RDY1}}$, $P4_1$, CNV_{SS}, and $\overline{RESET}$ pin) in the serial I/O mode are described below.

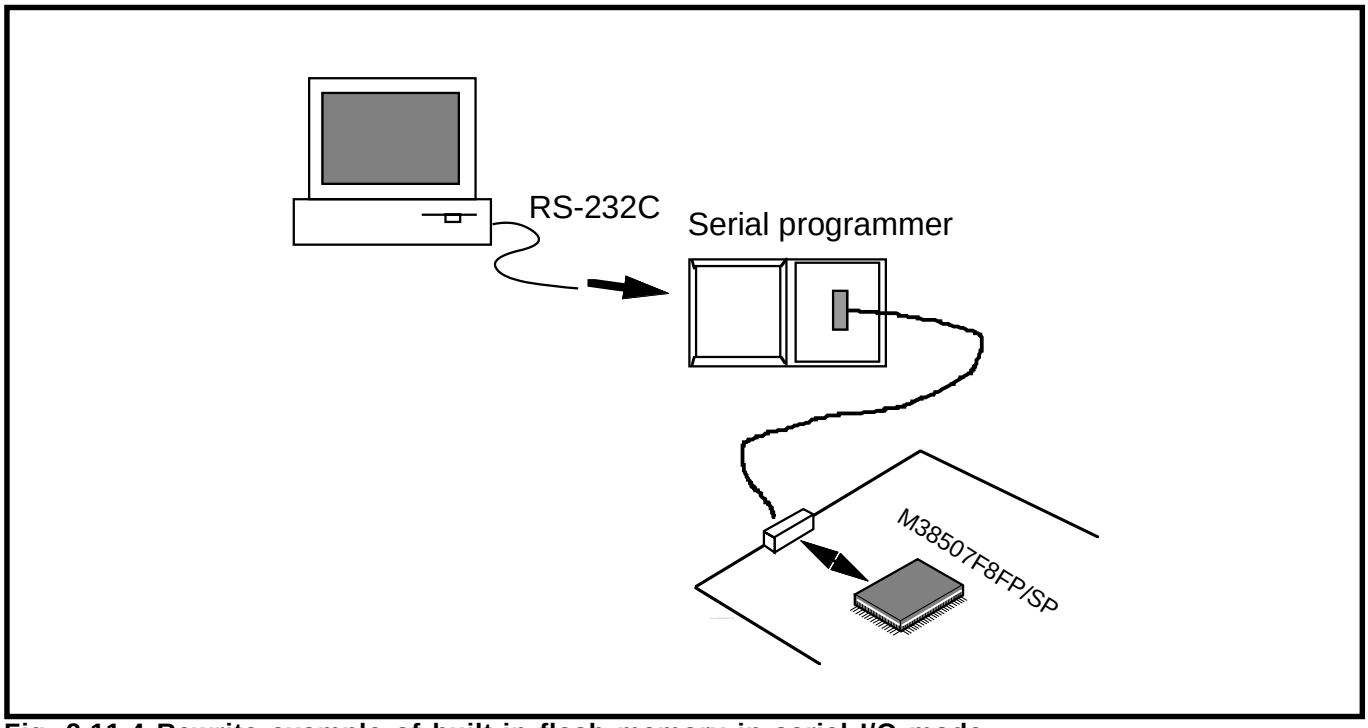
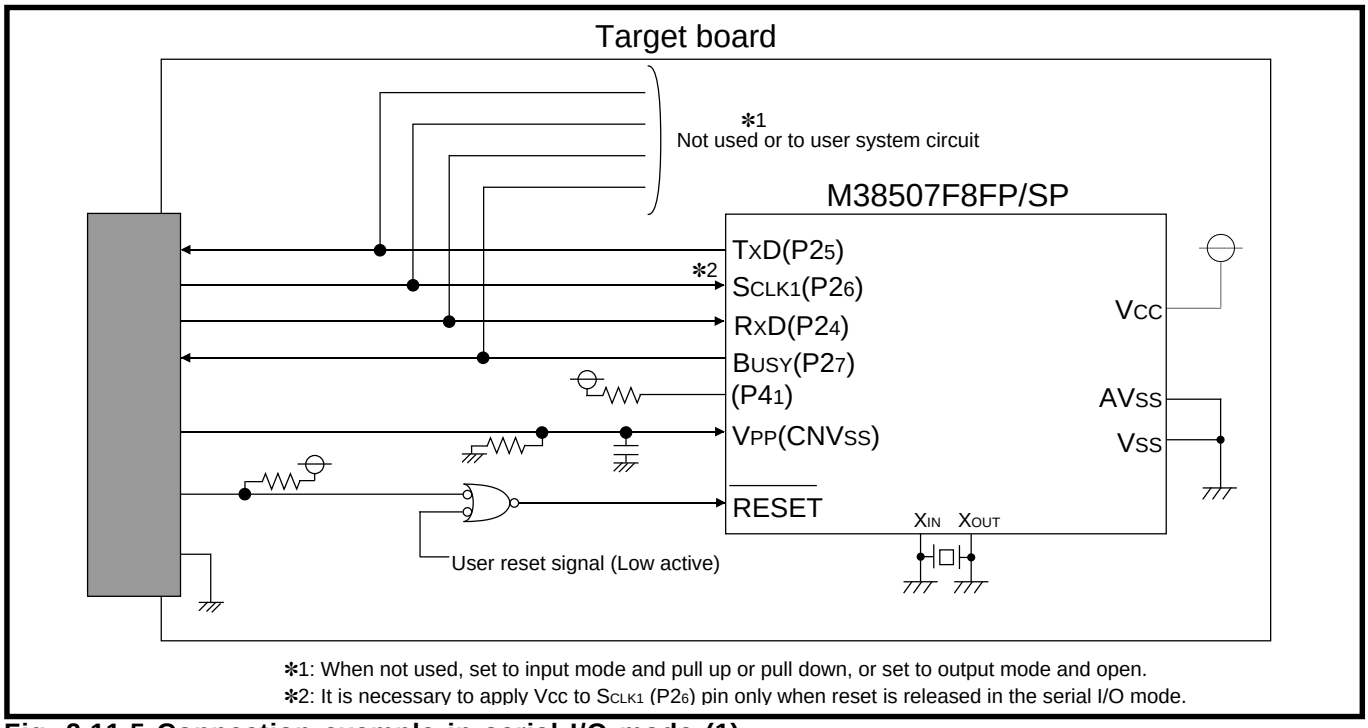


Fig. 2.11.4 Rewrite example of built-in flash memory in serial I/O mode

① **When control signals are not affected to user system circuit**

When the control signals in the serial I/O mode are not used or not affected to the user system circuit, they can be connected as shown in Figure 2.11.5.

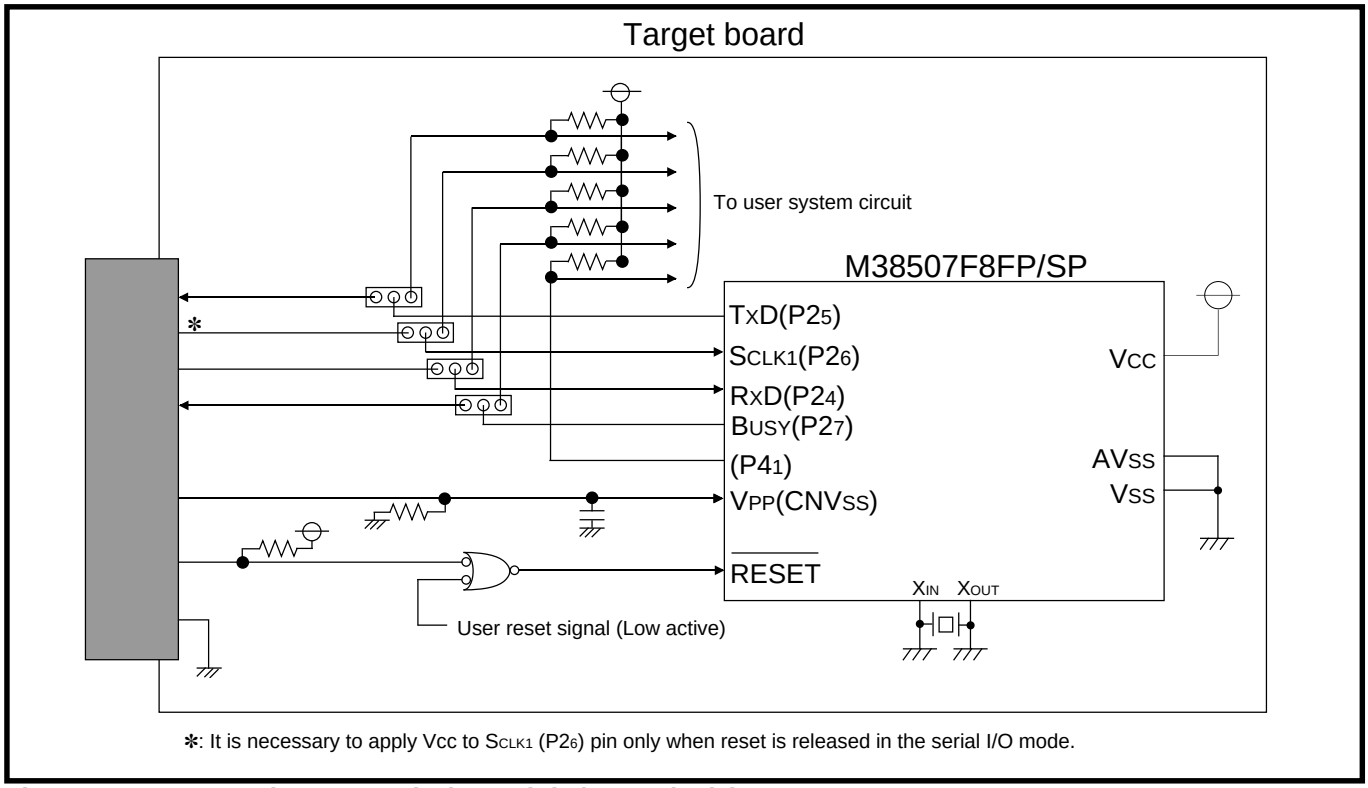


∗1: When not used, set to input mode and pull up or pull down, or set to output mode and open.
∗2: It is necessary to apply Vcc to S_{CLK1} ($P2_6$) pin only when reset is released in the serial I/O mode.

Fig. 2.11.5 Connection example in serial I/O mode (1)

② **When control signals are affected to user system circuit-1**

Figure 2.11.6 shows an example that the jumper switch cut-off the control signals not to supply to the user system circuit in the serial I/O mode.

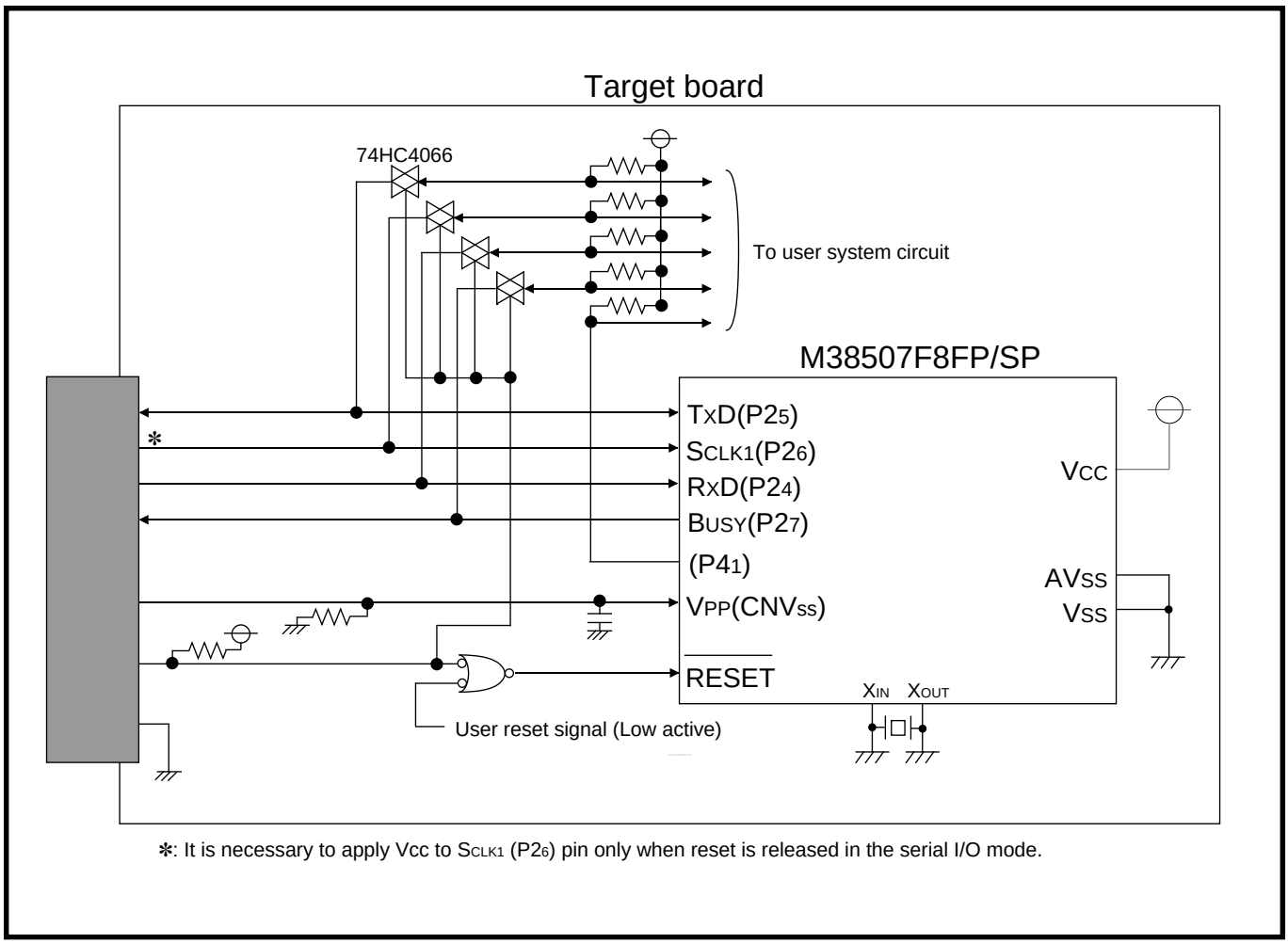


∗: It is necessary to apply Vcc to S_{CLK1} ($P2_6$) pin only when reset is released in the serial I/O mode.

Fig. 2.11.6 Connection example in serial I/O mode (2)

③ **When control signals are affected to user system circuit-2**

Figure 2.11.7 shows an example that the analog switch (74HC4066) cut-off the control signals not to supply to the user system circuit in the serial I/O mode.



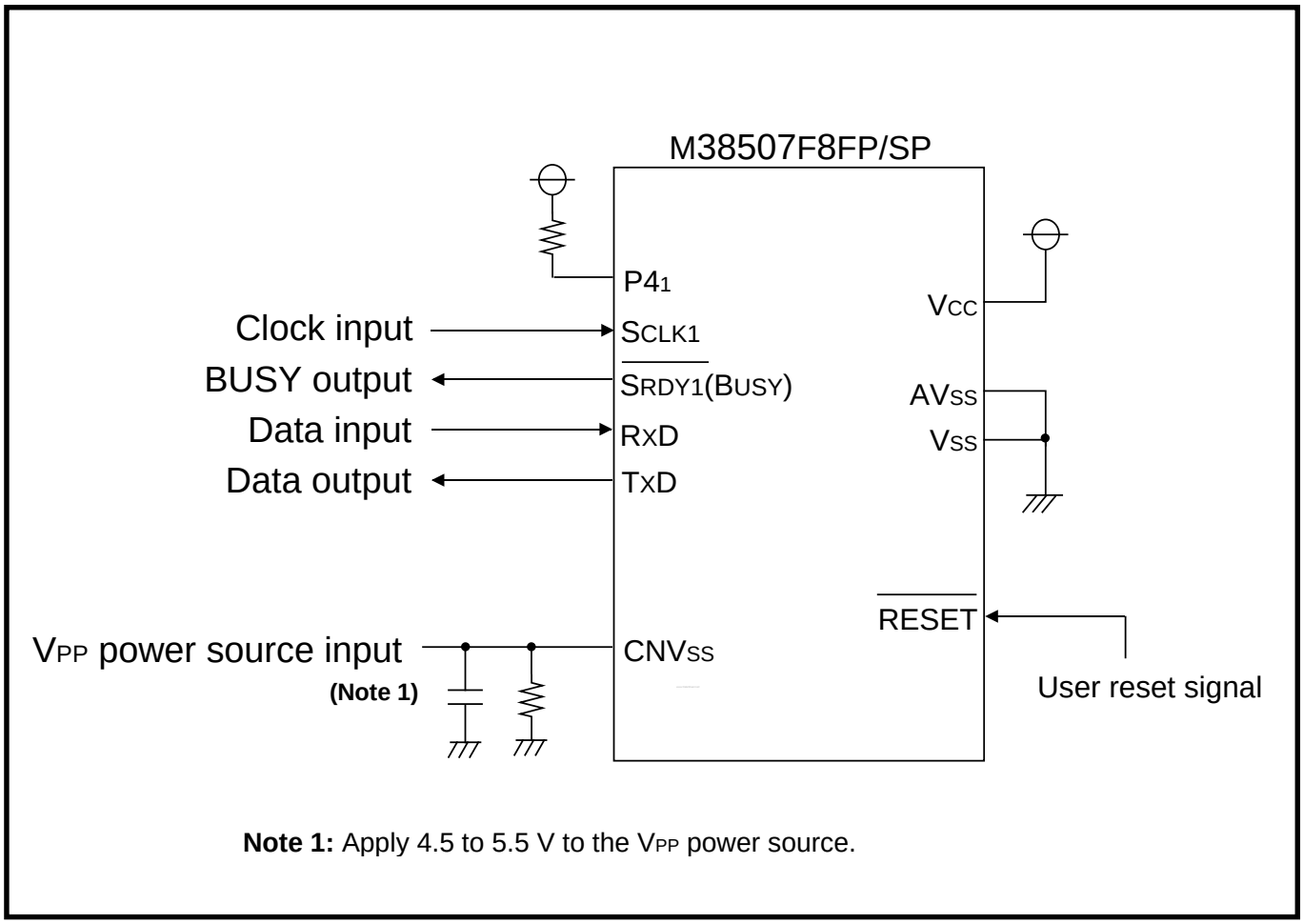
✻: It is necessary to apply Vcc to S_{CLK1} ($P2_6$) pin only when reset is released in the serial I/O mode.

Fig. 2.11.7 Connection example in serial I/O mode (3)

(2) Control 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.

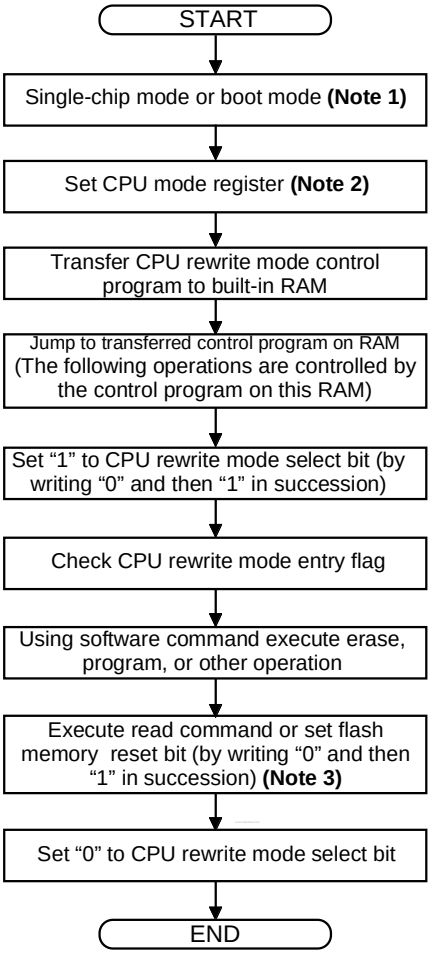
Figure 2.11.8 shows an example of the reprogramming system for the built-in flash memory in the CPU rewrite mode. Figure 2.11.9 shows the CPU rewrite mode beginning/release flowchart.



Note 1: Apply 4.5 to 5.5 V to the V_{PP} power source.

Fig. 2.11.8 Example of rewrite system for built-in flash memory in CPU rewrite mode

2.11 Flash memory mode



Notes 1: When MCU starts in the single-chip mode, it is necessary to apply 5 V ± 10 % to dhe CNVss pin until confirming of the CPU rewrite mode entry flag.
2: Set bits 6 and 7 (main clock division ratio selection bits) of the CPU mode register (address $003B_{16}$).
3: Before releasing the CPU rewrite mode after completing erase or program operation, always be sure to execute a read command or reset the flash memory.

Fig. 2.11.9 CPU rewrite mode beginning/release flowchart

2.11.8 Notes on CPU rewrite mode

(1) Operation speed

During CPU rewrite mode, set the internal clock ϕ 4 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.

MEMORANDUM

CHAPTER 3

APPENDIX

APPENDIX

3.1 Electrical characteristics

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 voltage		–0.3 to 6.5	V
V_I	Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$, $P2_1$, $P2_4$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$, V_{REF}	All voltages are based on V_{SS}. Output transistors are cut off.	–0.3 to V_{CC} +0.3	V
V_I	Input voltage $P2_2$, $P2_3$		–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 CNV_{SS}		–0.3 to V_{CC} +0.3	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$, $P2_1$, $P2_4$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$, X_{OUT}		–0.3 to V_{CC} +0.3	V
V_O	Output voltage $P2_2$, $P2_3$		–0.3 to 5.8	V
P_d	Power dissipation	T_a = 25 °C	1000 **(Note)**	mW
T_{opr}	Operating temperature		–20 to 85	°C
T_{stg}	Storage temperature		–40 to 125	°C

Note : The rating becomes 300mW at the 42P2R-A/E package.

3.1.2 Recommended operating conditions

Table 3.1.2 Recommended operating conditions (1)
(V_{CC} = 2.7 to 5.5 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage	8 MHz (high-speed mode)	4.0	5.0	5.5	V
		8 MHz (middle-speed mode), 4 MHz (high-speed mode)	2.7	5.0	5.5	
V_{SS}	Power source voltage			0		V
V_{REF}	A-D convert reference voltage		2.0		V_{CC}	V
AV_{SS}	Analog power source voltage			0		V
V_{IA}	Analog input voltage	AN_0–AN_4	AV_{SS}		V_{CC}	V
V_{IH}	“H” input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$	0.8V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage	$\overline{RESET}$, X_{IN}, CNV_{SS}	0.8V_{CC}		V_{CC}	V
V_{IL}	“L” input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$	0		0.2V_{CC}	V
V_{IL}	“L” input voltage	$\overline{RESET}$, CNV_{SS}	0		0.2V_{CC}	V
V_{IL}	“L” input voltage	X_{IN}	0		0.16V_{CC}	V
$\Sigma I_{OH(peak)}$	“H” total peak output current **(Note)**	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_4$			–80	mA
$\Sigma I_{OH(peak)}$	“H” total peak output current **(Note)**	$P2_0$, $P2_1$, $P2_4$–$P2_7$, $P4_0$–$P4_4$			–80	mA
$\Sigma I_{OL(peak)}$	“L” total peak output current **(Note)**	$P0_0$–$P0_7$, $P3_0$–$P3_4$			80	mA
$\Sigma I_{OL(peak)}$	“L” total peak output current **(Note)**	$P1_0$–$P1_7$			120	mA
$\Sigma I_{OL(peak)}$	“L” total peak output current **(Note)**	$P2_0$–$P2_7$,$P4_0$–$P4_4$			80	mA
$\Sigma I_{OH(avg)}$	“H” total average output current **(Note)**	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P3_0$–$P3_4$			–40	mA
$\Sigma I_{OH(avg)}$	“H” total average output current **(Note)**	$P2_0$, $P2_1$, $P2_4$–$P2_7$, $P4_0$–$P4_4$			–40	mA
$\Sigma I_{OL(avg)}$	“L” total average output current **(Note)**	$P0_0$–$P0_7$, $P3_0$–$P3_4$			40	mA
$\Sigma I_{OL(avg)}$	“L” total average output current **(Note)**	$P1_0$–$P1_7$			60	mA
$\Sigma I_{OL(avg)}$	“L” total average output current **(Note)**	$P2_0$–$P2_7$,$P4_0$–$P4_4$			40	mA

Note : 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.

APPENDIX

3.1 Electrical characteristics

Table 3.1.3 Recommended operating conditions (2)
(V_{CC} = 2.7 to 5.5 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol		Parameter	Limits			Unit
			Min.	Typ.	Max.	
$I_{OH(peak)}$	"H" peak output current **(Note 1)**	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$, $P2_1$, $P2_4$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$			–10	mA
$I_{OL(peak)}$	"L" peak output current **(Note 1)**	$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$			10	mA
$I_{OL(peak)}$	"L" peak output current **(Note 1)**	$P1_0$–$P1_7$			20	mA
$I_{OH(avg)}$	"H" average output current **(Note 2)**	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$, $P2_1$, $P2_4$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$			–5	mA
$I_{OL(avg)}$	"L" average output current **(Note 2)**	$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$			5	mA
$I_{OL(avg)}$	"L" average output current **(Note 2)**	$P1_0$–$P1_7$			15	mA
f(X_{IN})	Internal clock oscillation frequency (V_{CC} = 4.0 to 5.5V) **(Note 3)**				8	MHz
f(X_{IN})	Internal clock oscillation frequency (V_{CC} = 2.7 to 5.5V) **(Note 3)**				4	MHz

Notes 1: The peak output current is the peak current flowing in each port.
2: The average output current I_{OL}(avg), I_{OH}(avg) are average value measured over 100 ms.
3: When the oscillation frequency has a duty cycle of 50%.

3.1.3 Electrical characteristics

Table 3.1.4 Electrical characteristics (1)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	"H" output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$, $P2_1$, $P2_4$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$ **(Note)**	I_{OH} = –10 mA, V_{CC} = 4.0–5.5 V	V_{CC}–2.0			V
		I_{OH} = –1.0 mA, V_{CC} = 2.7–5.5 V	V_{CC}–1.0			V
V_{OL}	"L" output voltage $P0_0$–$P0_7$, $P2_0$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$	I_{OL} = 10 mA, V_{CC} = 4.0–5.5 V			2.0	V
		I_{OL} = 1.0 mA, V_{CC} = 2.7–5.5 V			1.0	V
V_{OL}	"L" output voltage $P1_0$–$P1_7$	I_{OL} = 20 mA, V_{CC} = 4.0–5.5 V			2.0	V
		I_{OL} = 10 mA, V_{CC} = 2.7–5.5 V			1.0	V
V_{T+}–V_{T-}	Hysteresis CNTR0, CNTR1, INT0–INT3			0.4		V
V_{T+}–V_{T-}	Hysteresis RxD, S_{CLK1}, S_{CLK2}, S_{IN2}			0.5		V
V_{T+}–V_{T-}	Hysteresis $\overline{RESET}$			0.5		V
I_{IH}	"H" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$, $P2_1$, $P2_4$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$	V_I = V_{CC}			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		µA
I_{IL}	"L" input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_4$, $P4_0$–$P4_4$	V_I = V_{SS}			–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		µA
V_{RAM}	RAM hold voltage	When clock stopped	2.0		5.5	V

Note: $P2_5$ is measured when the $P2_5$/TxD P-channel output disable bit of the UART control register (bit 4 of address $001B_{16}$) is "0".

Table 3.1.5 Electrical characteristics (2)
(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
I_{CC}	Power source current	High-speed mode $f(X_{IN})$ = 8 MHz $f(X_{CIN})$ = 32.768 kHz Output transistors “off”			6.8	13	mA
		High-speed mode $f(X_{IN})$ = 8 MHz (in WIT state) $f(X_{CIN})$ = 32.768 kHz Output transistors “off”			1.6		mA
		Low-speed mode $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz Output transistors “off”	Except M38507F8FP/SP		60	200	µA
			M38507F8FP/SP		250		µA
		Low-speed mode $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz (in WIT state) Output transistors “off”	Except M38507F8FP/SP		20	40	µA
			M38507F8FP/SP		70		µA
		Low-speed mode (V_{CC} = 3 V) $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz Output transistors “off”	Except M38507F8FP/SP		20	55	µA
			M38507F8FP/SP		150		µA
		Low-speed mode (V_{CC} = 3 V) $f(X_{IN})$ = stopped $f(X_{CIN})$ = 32.768 kHz (in WIT state) Output transistors “off”	Except M38507F8FP/SP		5.0	10.0	µA
			M38507F8FP/SP		20		µA
		Middle-speed mode $f(X_{IN})$ = 8 MHz $f(X_{CIN})$ = stopped Output transistors “off”			4.0	7.0	mA
		Middle-speed mode $f(X_{IN})$ = 8 MHz (in WIT state) $f(X_{CIN})$ = stopped Output transistors “off”			1.5		mA
		Increment when A-D conversion is executed $f(X_{IN})$ = 8 MHz			800		µA
		All oscillation stopped (in STP state) Output transistors “off”	Ta = 25 °C		0.1	1.0	µA
			Ta = 85 °C			10	µA

APPENDIX

3.1 Electrical characteristics

3.1.4 A-D converter characteristics

Table 3.1.6 A-D converter characteristics

(V_{CC} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 8 MHz, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
–	Resolution					10	bit
–	Absolute accuracy (excluding quantization error)					±4	LSB
t_{CONV}	Conversion time		High-speed mode, Middle-speed mode			61	2tc(X_{IN})
			Low-speed mode		40		µs
R_{LADDER}	Ladder resistor				35		kΩ
I_{VREF}	Reference power source input current	V_{REF} “on”	V_{REF} = 5.0 V	50	150	200	µA
		V_{REF} “off”				5.0	
$I_{I(AD)}$	A-D port input current				0.5	5.0	µA

3.1.5 Timing requirements and switching characteristics

Table 3.1.7 Timing requirements (1)

(Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input “L” pulse width	20			XIN cycle
$t_{C(XIN)}$	External clock input cycle time	125			ns
$t_{WH(XIN)}$	External clock input “H” pulse width	50			ns
$t_{WL(XIN)}$	External clock input “L” pulse width	50			ns
$t_{C(CNTR)}$	CNTR0, CNTR1 input cycle time	200			ns
$t_{WH(CNTR)}$	CNTR0, CNTR1 input “H” pulse width	80			ns
$t_{WL(CNTR)}$	CNTR0, CNTR1 input “L” pulse width	80			ns
$t_{WH(INT)}$	INT0 to INT3 input “H” pulse width	80			ns
$t_{WL(INT)}$	INT0 to INT3 input “L” pulse width	80			ns
$t_{C(SCLK1)}$	Serial I/O1 clock input cycle time **(Note)**	800			ns
$t_{WH(SCLK1)}$	Serial I/O1 clock input “H” pulse width **(Note)**	370			ns
$t_{WL(SCLK1)}$	Serial I/O1 clock input “L” pulse width **(Note)**	370			ns
$t_{su(RxD-SCLK1)}$	Serial I/O1 input setup time	220			ns
$t_{h(SCLK1-RxD)}$	Serial I/O1 input hold time	100			ns
$t_{C(SCLK2)}$	Serial I/O2 clock input cycle time	1000			ns
$t_{WH(SCLK2)}$	Serial I/O2 clock input “H” pulse width	400			ns
$t_{WL(SCLK2)}$	Serial I/O2 clock input “L” pulse width	400			ns
$t_{su(SIN2-SCLK2)}$	Serial I/O2 clock input setup time	200			ns
$t_{h(SCLK2-SIN2)}$	Serial I/O2 clock input hold time	200			ns

Note : When f(XIN) = 8 MHz and bit 6 of address 001A16 is “1” (clock synchronous).
Divide this value by four when f(XIN) = 8 MHz and bit 6 of address 001A16 is “0” (UART).

Table 3.1.8 Timing requirements (2)

(Vcc = 2.7 to 5.5 V, Vss = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input “L” pulse width	20			XIN cycle
$t_{C(XIN)}$	External clock input cycle time	250			ns
$t_{WH(XIN)}$	External clock input “H” pulse width	100			ns
$t_{WL(XIN)}$	External clock input “L” pulse width	100			ns
$t_{C(CNTR)}$	CNTR0, CNTR1 input cycle time	500			ns
$t_{WH(CNTR)}$	CNTR0, CNTR1 input “H” pulse width	230			ns
$t_{WL(CNTR)}$	CNTR0, CNTR1 input “L” pulse width	230			ns
$t_{WH(INT)}$	INT0 to INT3 input “H” pulse width	230			ns
$t_{WL(INT)}$	INT0 to INT3 input “L” pulse width	230			ns
$t_{C(SCLK1)}$	Serial I/O1 clock input cycle time **(Note)**	2000			ns
$t_{WH(SCLK1)}$	Serial I/O1 clock input “H” pulse width **(Note)**	950			ns
$t_{WL(SCLK1)}$	Serial I/O1 clock input “L” pulse width **(Note)**	950			ns
$t_{su(RxD-SCLK1)}$	Serial I/O1 input setup time	400			ns
$t_{h(SCLK1-RxD)}$	Serial I/O1 input hold time	200			ns
$t_{C(SCLK2)}$	Serial I/O2 clock input cycle time	2000			ns
$t_{WH(SCLK2)}$	Serial I/O2 clock input “H” pulse width	950			ns
$t_{WL(SCLK2)}$	Serial I/O2 clock input “L” pulse width	950			ns
$t_{su(SIN2-SCLK2)}$	Serial I/O2 clock input setup time	400			ns
$t_{h(SCLK2-SIN2)}$	Serial I/O2 clock input hold time	300			ns

Note : When f(XIN) = 4 MHz and bit 6 of address 001A16 is “1” (clock synchronous).
Divide this value by four when f(XIN) = 4 MHz and bit 6 of address 001A16 is “0” (UART).

APPENDIX

3.1 Electrical characteristics

Table 3.1.9 Switching characteristics (1)

(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t_{WH} (S_{CLK1})	Serial I/O1 clock output “H” pulse width	Fig.3.1.1	$t_{C(SCLK1)}$/2–30			ns
t_{WL} (S_{CLK1})	Serial I/O1 clock output “L” pulse width		$t_{C(SCLK1)}$/2–30			ns
td (S_{CLK1}-TxD)	Serial I/O1 output delay time **(Note 1)**				140	ns
tv (S_{CLK1}-TxD)	Serial I/O1 output valid time **(Note 1)**		–30			ns
tr (S_{CLK1})	Serial I/O1 clock output rising time				30	ns
tf (S_{CLK1})	Serial I/O1 clock output falling time				30	ns
t_{WH} (S_{CLK2})	Serial I/O2 clock output “H” pulse width		$t_{C(SCLK2)}$/2–160			ns
t_{WL} (S_{CLK2})	Serial I/O2 clock output “L” pulse width		$t_{C(SCLK2)}$/2–160			ns
td (S_{CLK2}-S_{OUT2})	Serial I/O2 output delay time **(Note 2)**				200	ns
tv (S_{CLK2}-S_{OUT2})	Serial I/O2 output valid time **(Note 2)**		0			ns
tf (S_{CLK2})	Serial I/O2 clock output falling time				30	ns
tr (CMOS)	CMOS output rising time **(Note 3)**			10	30	ns
tf (CMOS)	CMOS output falling time **(Note 3)**			10	30	ns

Notes 1: When the $P2_5$/TxD P-channel output disable bit of the UART control register (bit 4 of address $001B_{16}$) is “0”.
2: When the $P0_1$/S_{OUT2} and $P0_2$/S_{CLK2} P-channel output disable bit of the Serial I/O2 control register 1 (bit 7 of address 0015_{16}) is “0”.
3: The X_{OUT} pin is excluded.

Table 3.1.10 Switching characteristics (2)

(V_{CC} = 2.7 to 5.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t_{WH} (S_{CLK1})	Serial I/O1 clock output “H” pulse width	Fig.3.1.1	$t_{C(SCLK1)}$/2–50			ns
t_{WL} (S_{CLK1})	Serial I/O1 clock output “L” pulse width		$t_{C(SCLK1)}$/2–50			ns
td (S_{CLK1}-TxD)	Serial I/O1 output delay time **(Note 1)**				350	ns
tv (S_{CLK1}-TxD)	Serial I/O1 output valid time **(Note 1)**		–30			ns
tr (S_{CLK1})	Serial I/O1 clock output rising time				50	ns
tf (S_{CLK1})	Serial I/O1 clock output falling time				50	ns
t_{WH} (S_{CLK2})	Serial I/O2 clock output “H” pulse width		$t_{C(SCLK2)}$/2–240			ns
t_{WL} (S_{CLK2})	Serial I/O2 clock output “L” pulse width		$t_{C(SCLK2)}$/2–240			ns
td (S_{CLK2}-S_{OUT2})	Serial I/O2 output delay time **(Note 2)**				400	ns
tv (S_{CLK2}-S_{OUT2})	Serial I/O2 output valid time **(Note 2)**		0			ns
tf (S_{CLK2})	Serial I/O2 clock output falling time				50	ns
tr (CMOS)	CMOS output rising time **(Note 3)**			20	50	ns
tf (CMOS)	CMOS output falling time **(Note 3)**			20	50	ns

Notes 1: When the $P2_5$/TxD P-channel output disable bit of the UART control register (bit 4 of address $001B_{16}$) is “0”.
2: When the $P0_1$/S_{OUT2} and $P0_2$/S_{CLK2} P-channel output disable bit of the Serial I/O2 control register 1 (bit 7 of address 0015_{16}) is “0”.
3: The X_{OUT} pin is excluded.

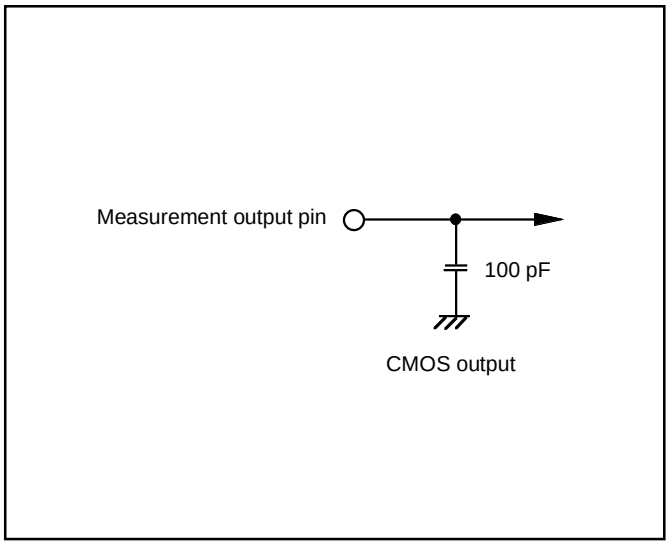


Fig. 3.1.1 Circuit for measuring output switching characteristics

APPENDIX

3.1 Electrical characteristics

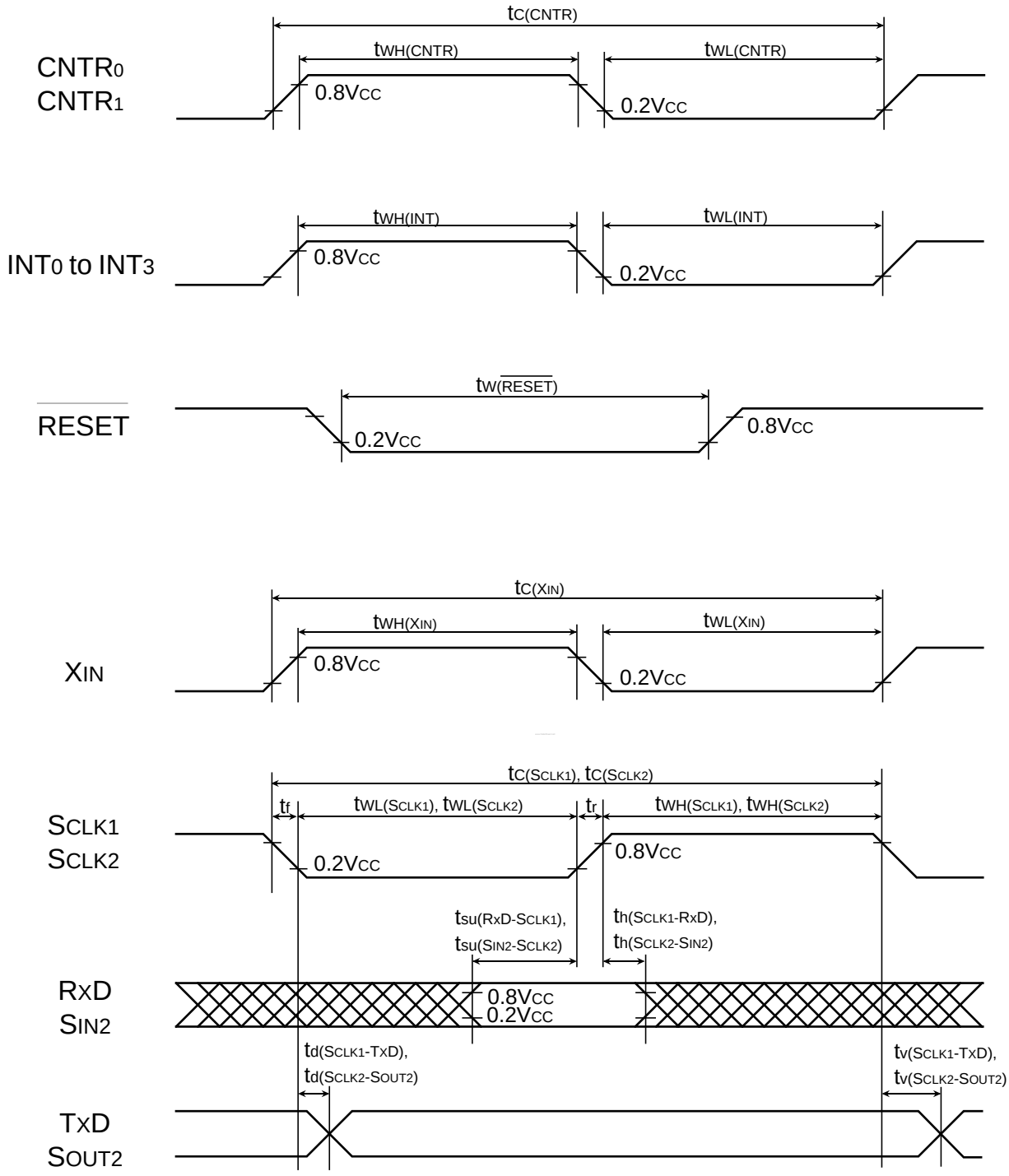


Fig. 3.1.2 Timing diagram

3.2 Standard characteristics

Standard characteristics described below are just examples of the 3850 Group (spec. H)'s characteristics and are not guaranteed. For rated values, refer to **"3.1 Electrical characteristics"**.

3.2.1 Flash memory version power source current standard characteristics

Figure 3.2.1, Figure 3.2.2, Figure 3.2.3, Figure 3.2.4, and Figure 3.2.5 show flash memory version (M38507F8) power source current standard characteristics.

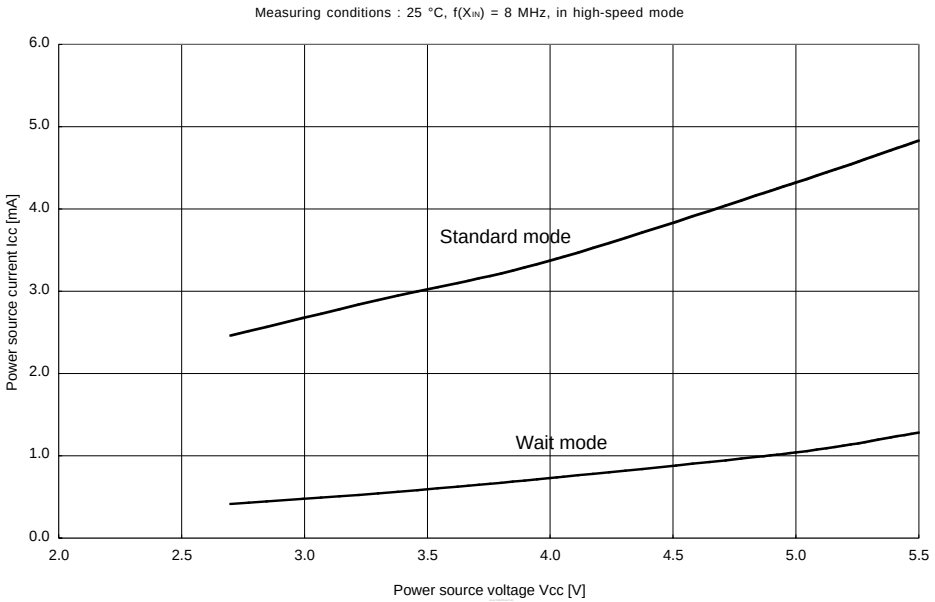


Fig. 3.2.1 Flash memory version power source current standard characteristics (in high-speed mode, $f(X_{IN})$ = 8 MHz)

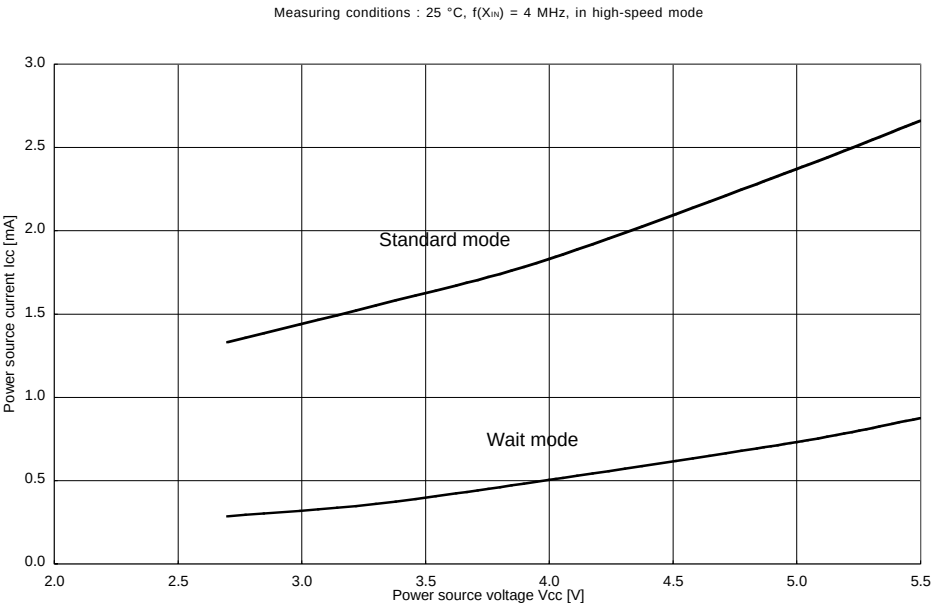


Fig. 3.2.2 Flash memory version power source current standard characteristics (in high-speed mode, $f(X_{IN})$ = 4 MHz)

APPENDIX

3.2 Standard characteristics

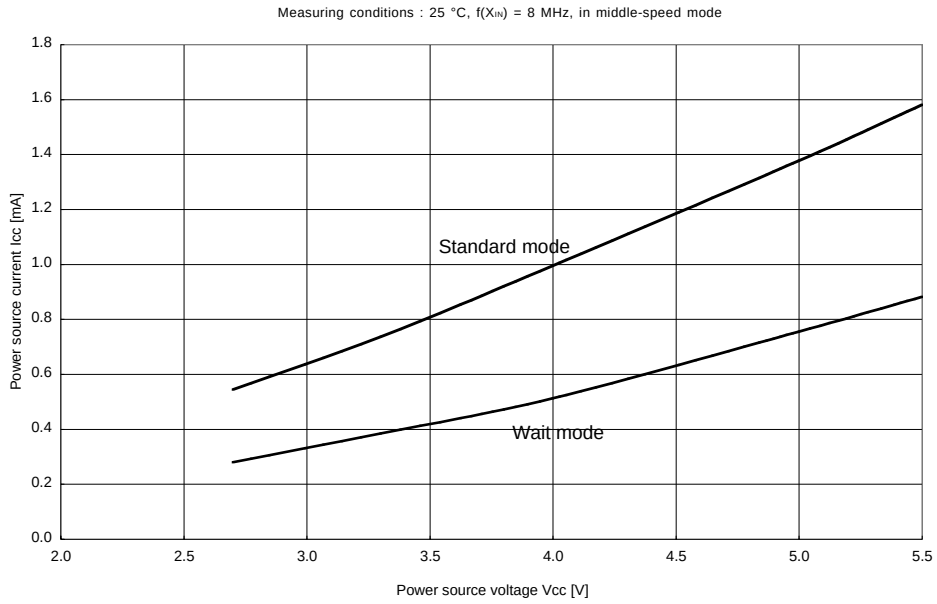


Fig. 3.2.3 Flash memory version power source current standard characteristics (in middle-speed mode, f(X_{IN}) = 8 MHz)

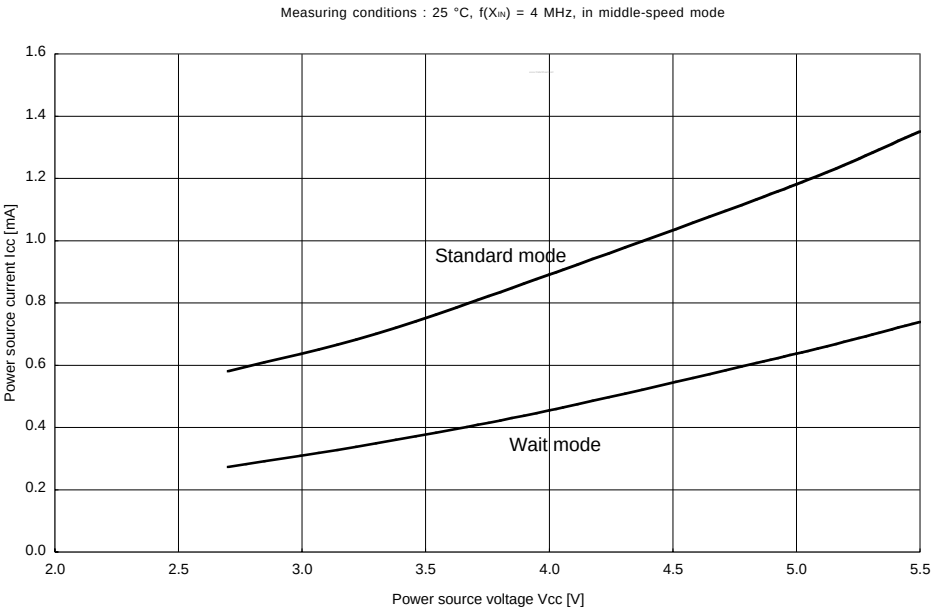


Fig. 3.2.4 Flash memory version power source current standard characteristics (in middle-speed mode, f(X_{IN}) = 4 MHz)

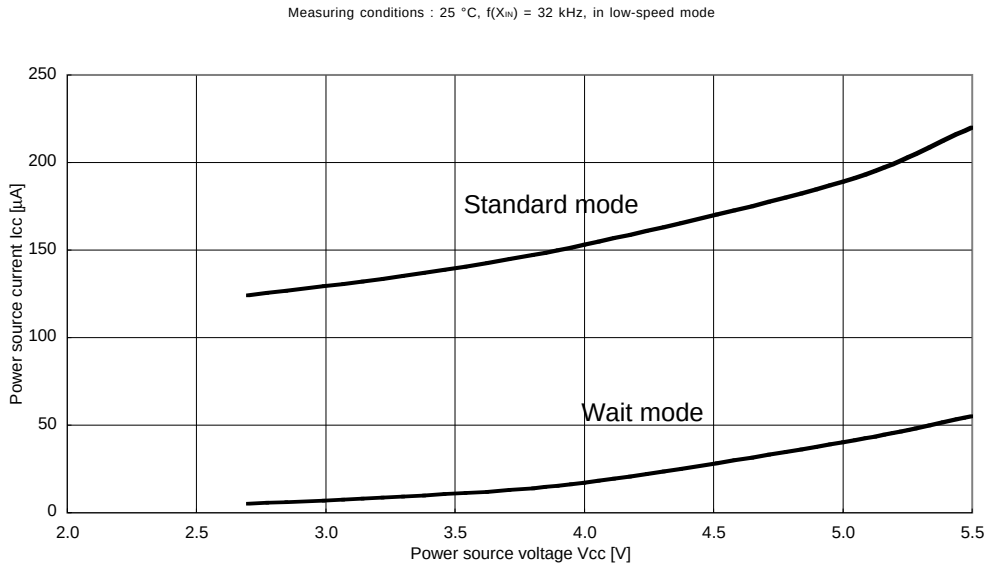


Fig. 3.2.5 Flash memory version power source current standard characteristics (in low-speed mode)

APPENDIX

3.2 Standard characteristics

3.2.2 Mask ROM version power source current standard characteristics

Figure 3.2.6, Figure 3.2.7, Figure 3.2.8, Figure 3.2.9 and Figure 3.2.10 show mask ROM version (M38503M2H, M38503M4H, M38504M6, M38507M8) power source current standard characteristics.

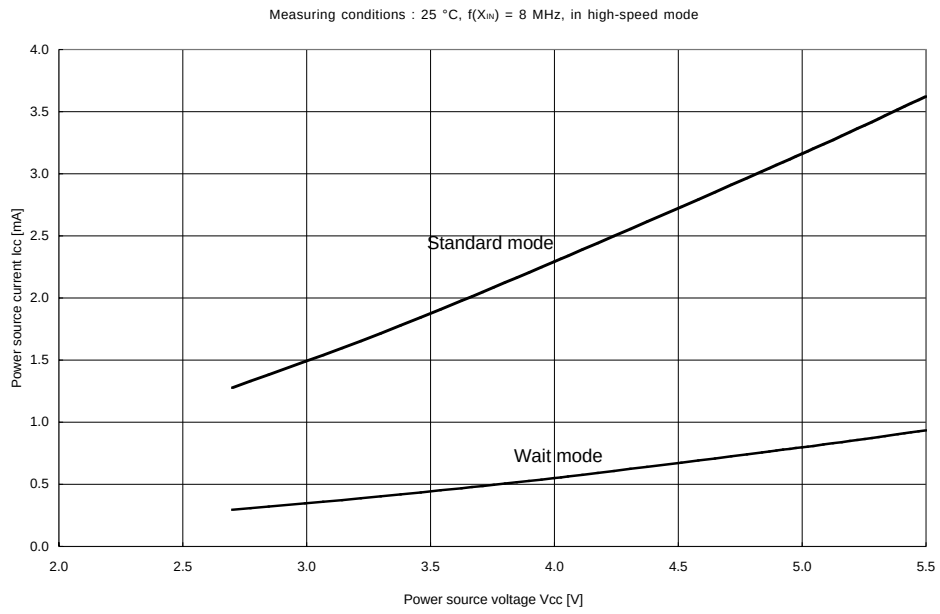


Fig. 3.2.6 Mask ROM version power source current standard characteristics (in high-speed mode, $f(X_{IN}) = 8$ MHz)

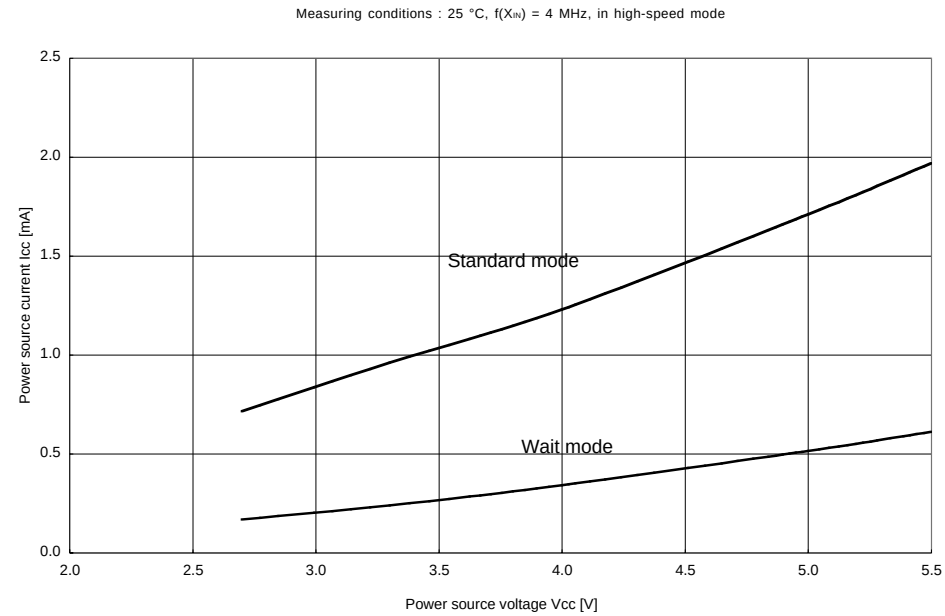


Fig. 3.2.7 Mask ROM version power source current standard characteristics (in high-speed mode, $f(X_{IN}) = 4$ MHz)

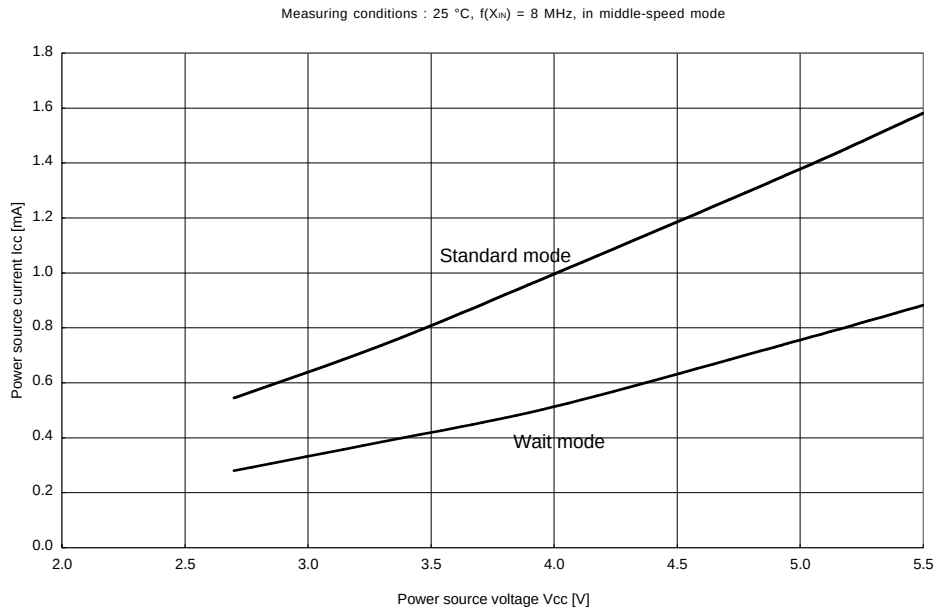


Fig. 3.2.8 Mask ROM version power source current standard characteristics (in middle-speed mode, f(XIN) = 8 MHz)

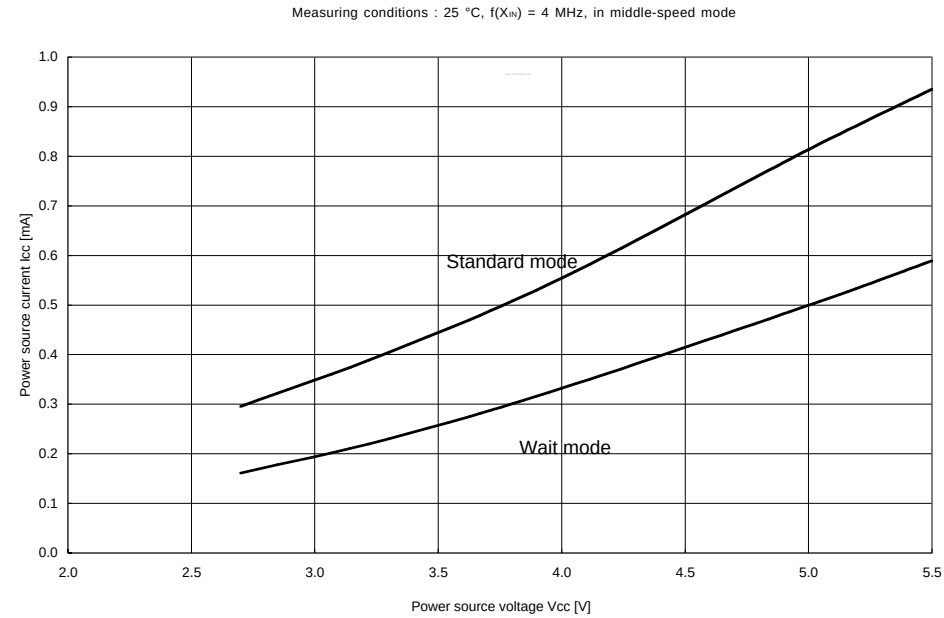


Fig. 3.2.9 Mask ROM version power source current standard characteristics (in middle-speed mode, f(XIN) = 4 MHz)

3.2 Standard characteristics

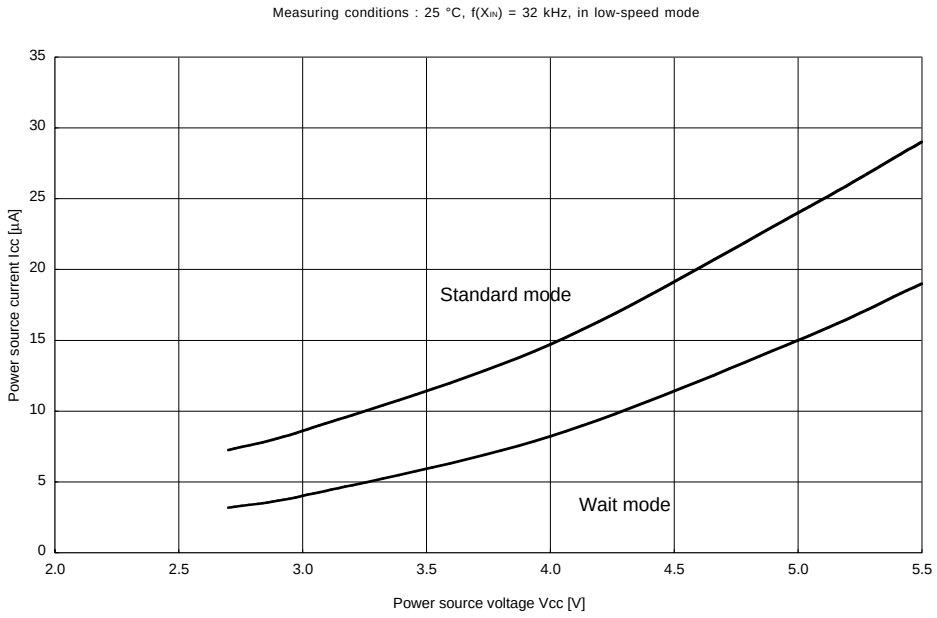


Fig. 3.2.10 Mask ROM version power source current standard characteristics (in low-speed mode)

3.2.3 PROM version power source current standard characteristics

Figure 3.2.11, Figure 3.2.12, Figure 3.2.13, Figure 3.2.14, and Figure 3.2.15 show flash memory version (M38504E6) power source current standard characteristics.

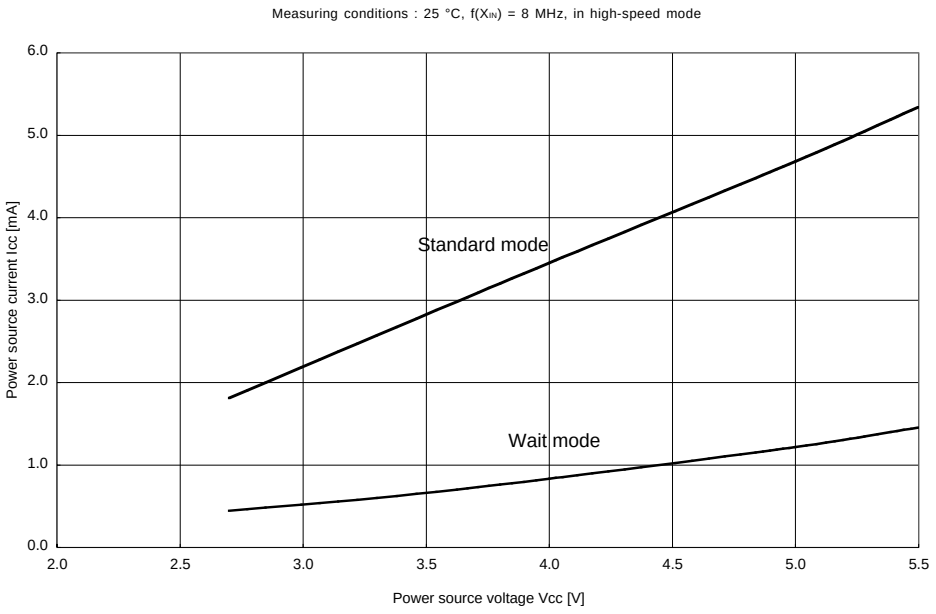


Fig. 3.2.11 PROM version power source current standard characteristics (in high-speed mode, f(X_{IN}) = 8 MHz)

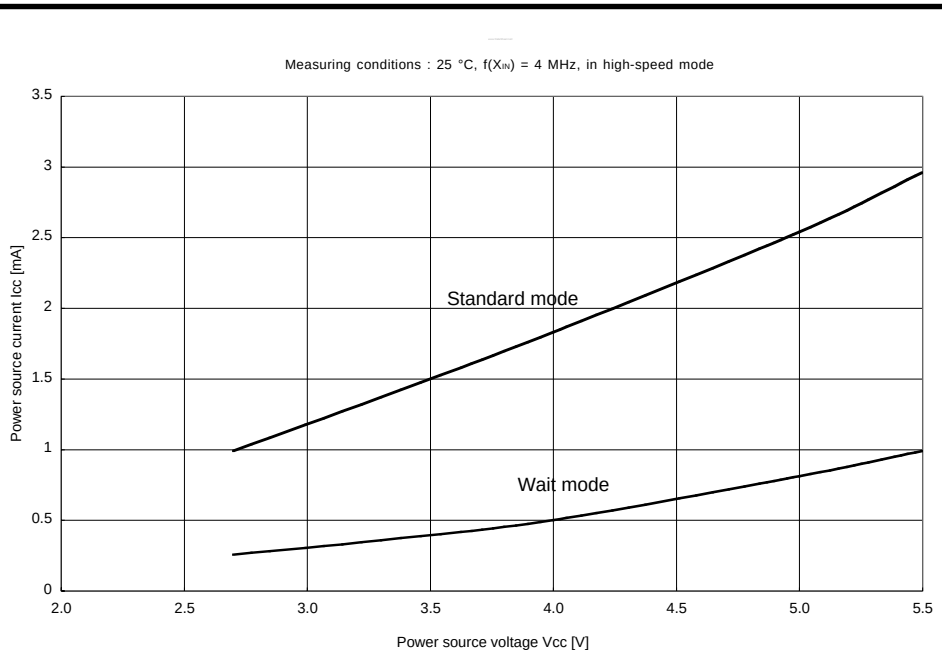


Fig. 3.2.12 PROM version power source current standard characteristics (in high-speed mode, f(X_{IN}) = 4 MHz)

APPENDIX

3.2 Standard characteristics

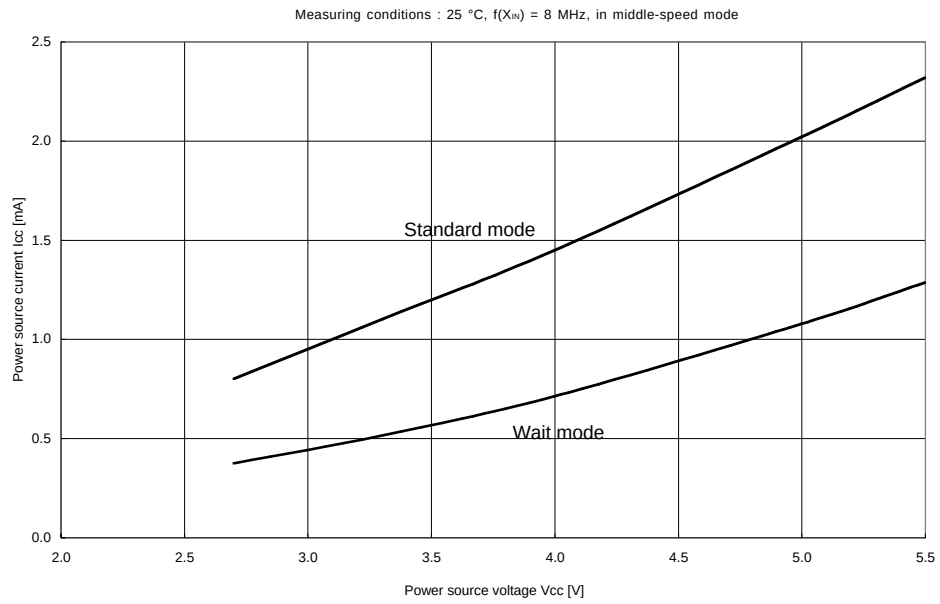


Fig. 3.2.13 PROM version power source current standard characteristics (in middle-speed mode, f(X_{IN}) = 8 MHz)

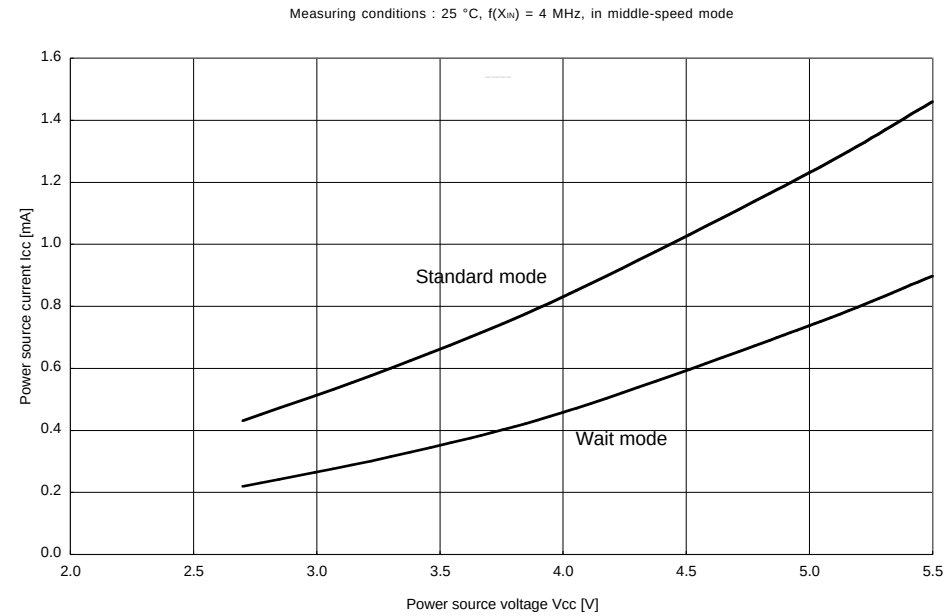


Fig. 3.2.14 PROM version power source current standard characteristics (in middle-speed mode, f(X_{IN}) = 4 MHz)

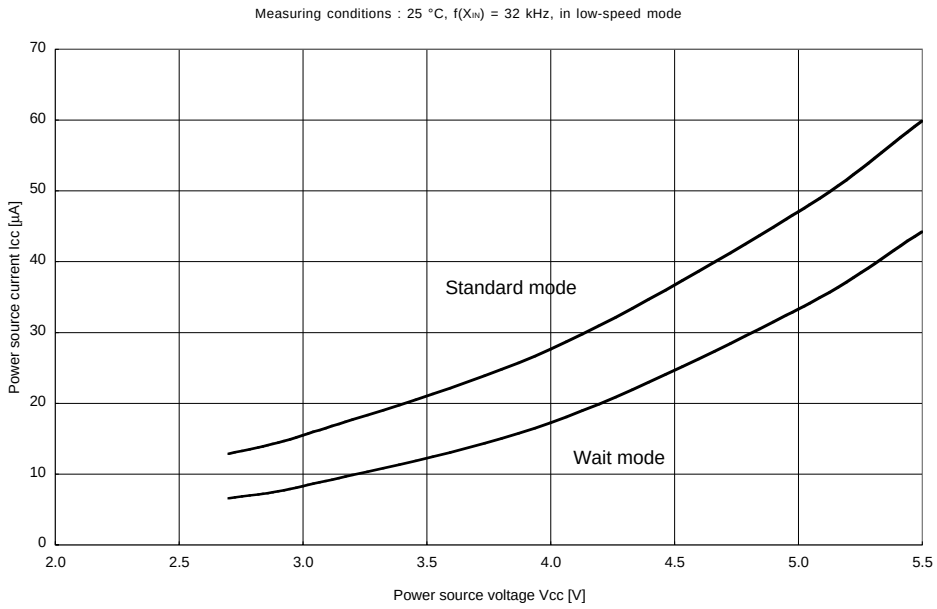


Fig. 3.2.15 PROM version power source current standard characteristics (in low-speed mode)

APPENDIX

3.2 Standard characteristics

3.2.4 Flash memory version port standard characteristics

Figure 3.2.16, Figure 3.2.17, Figure 3.2.18 and Figure 3.2.19 show flash memory version (M38507F8) port standard characteristics.

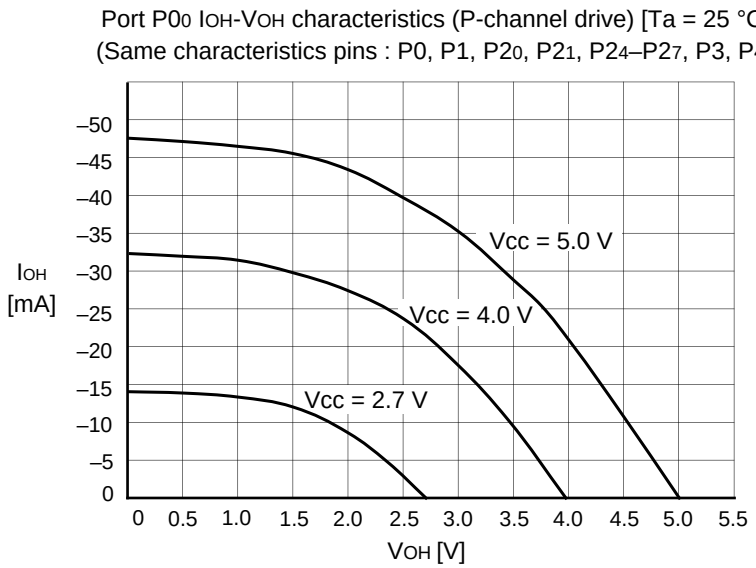


Fig. 3.2.16 CMOS output port P-channel side characteristics (Ta = 25 °C)

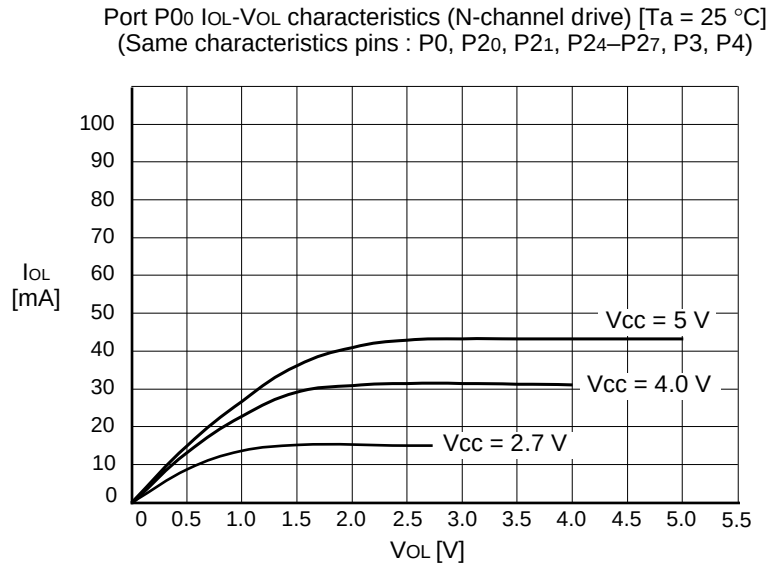


Fig. 3.2.17 CMOS output port N-channel side characteristics (Ta = 25 °C)

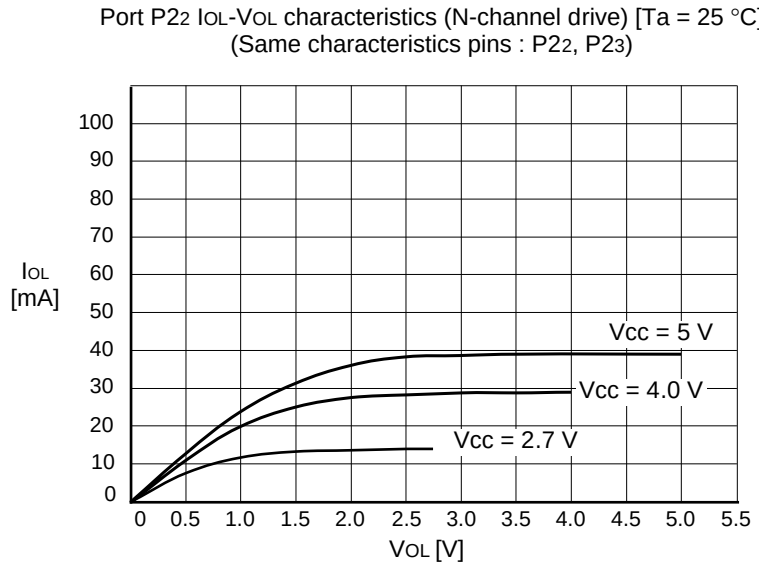


Fig. 3.2.18 N-channel open-drain output port N-channel side characteristics (Ta = 25 °C)

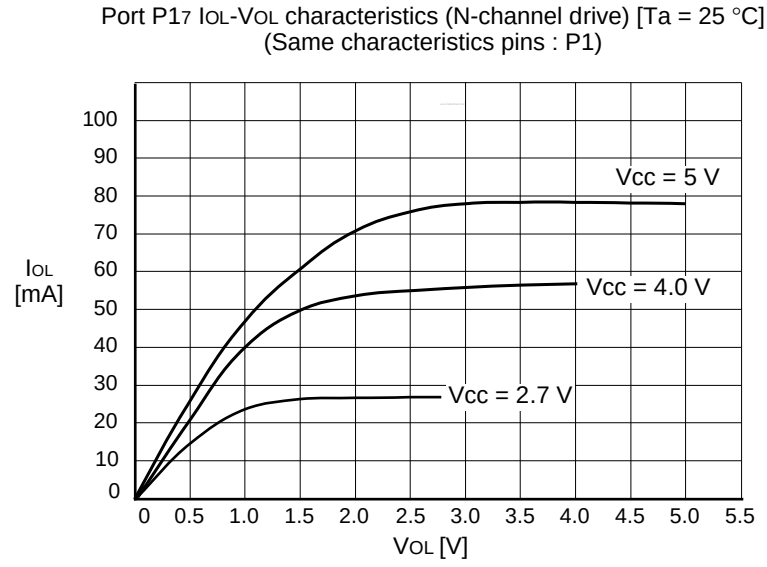


Fig. 3.2.19 CMOS large current output port N-channel side characteristics (Ta = 25 °C)

APPENDIX

3.2 Standard characteristics

3.2.5 Mask ROM version port standard characteristics

Figure 3.2.20, Figure 3.2.21, Figure 3.2.22 and Figure 3.2.23 show mask ROM version (M38503M2H, M38503M4H, M38504M6, M38507M8) port standard characteristics.

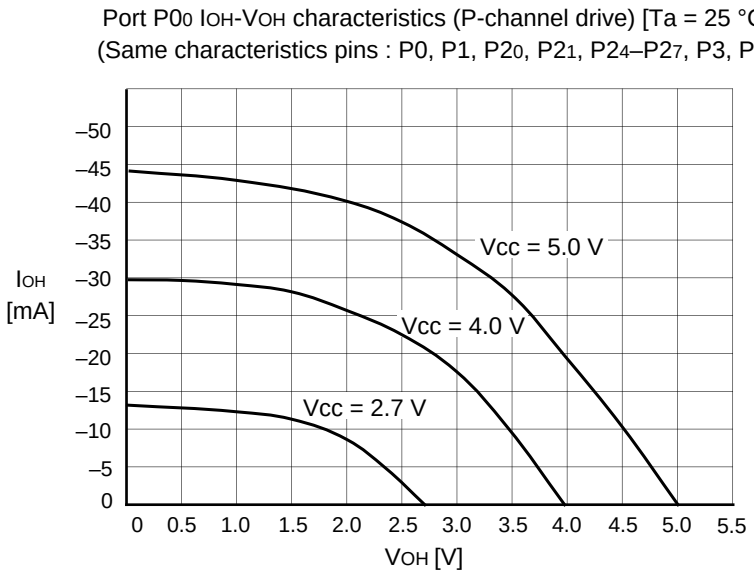


Fig. 3.2.20 CMOS output port P-channel side characteristics (Ta = 25 °C)

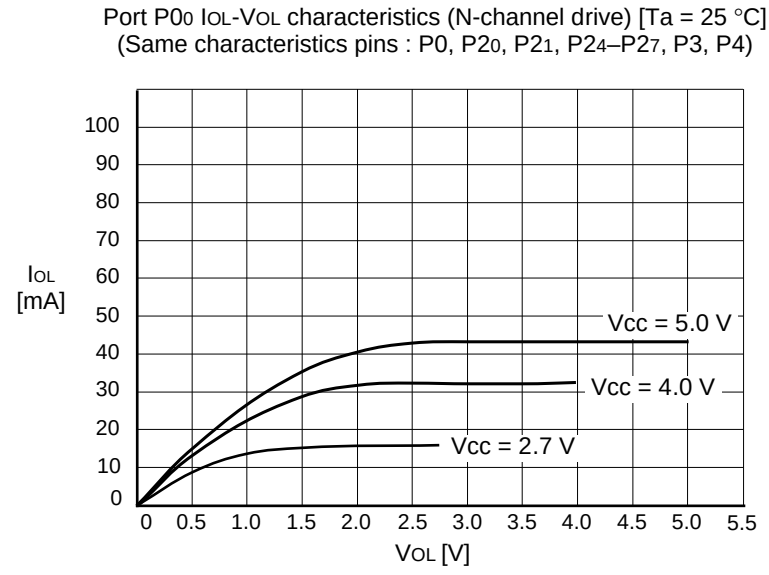


Fig. 3.2.21 CMOS output port N-channel side characteristics (Ta = 25 °C)

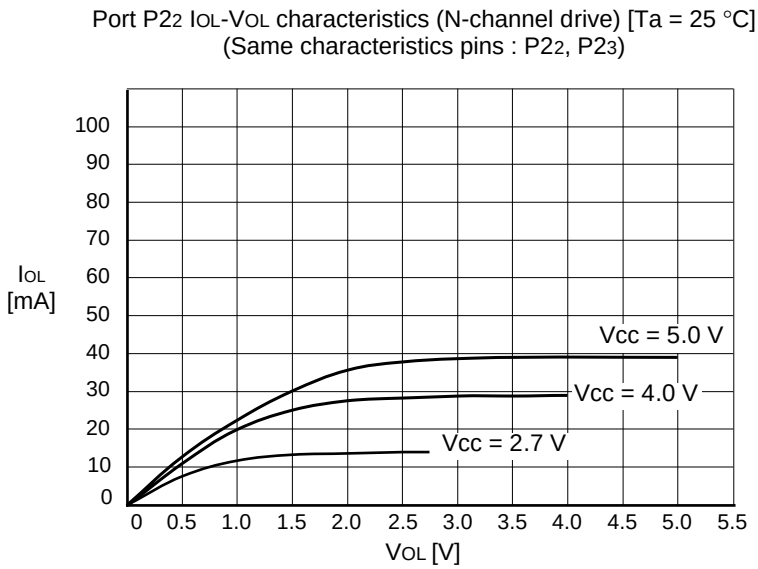


Fig. 3.2.22 N-channel open-drain output port N-channel side characteristics (Ta = 25 °C)

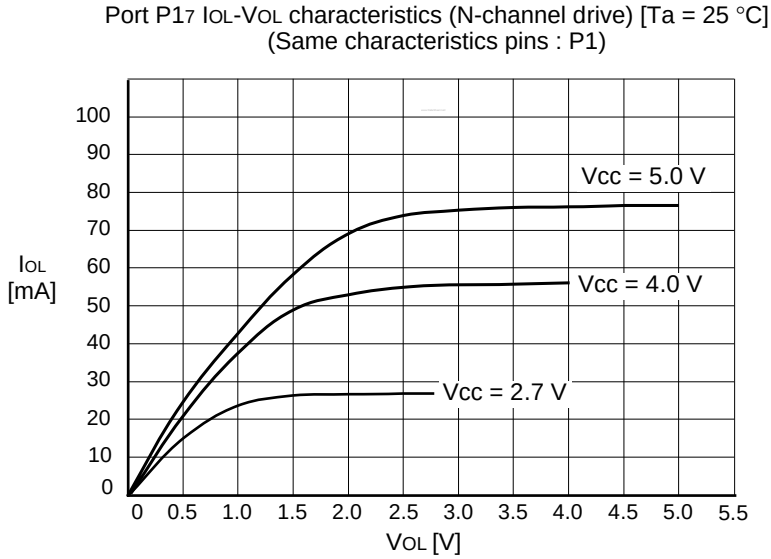


Fig. 3.2.23 CMOS large current output port N-channel side characteristics (Ta = 25 °C)

APPENDIX

3.2 Standard characteristics

3.2.6 PROM version port standard characteristics

Figure 3.2.24, Figure 3.2.25, Figure 3.2.26 and Figure 3.2.27 show PROM version (M38504E6) port standard characteristics.

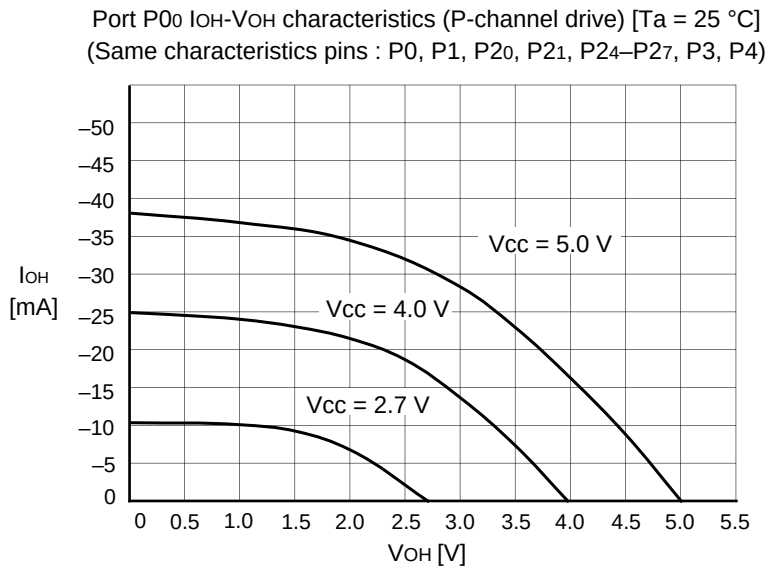


Fig. 3.2.24 CMOS output port P-channel side characteristics (Ta = 25 °C)

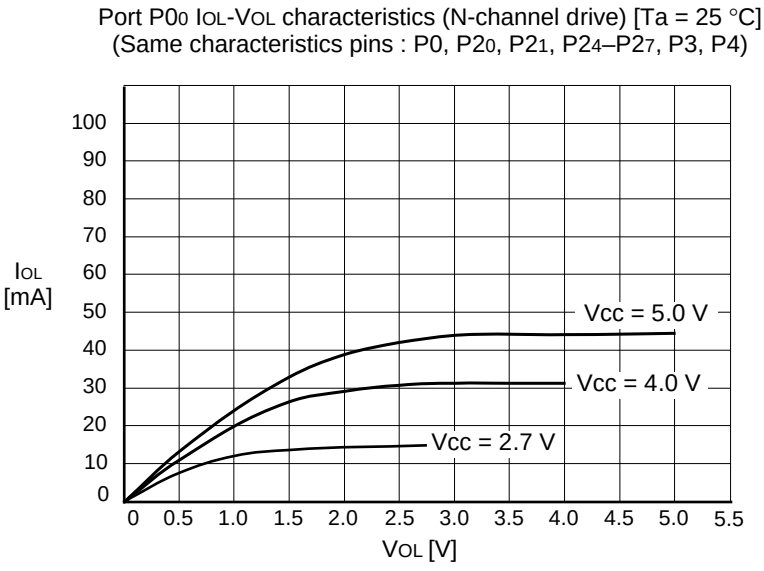


Fig. 3.2.25 CMOS output port N-channel side characteristics (Ta = 25 °C)

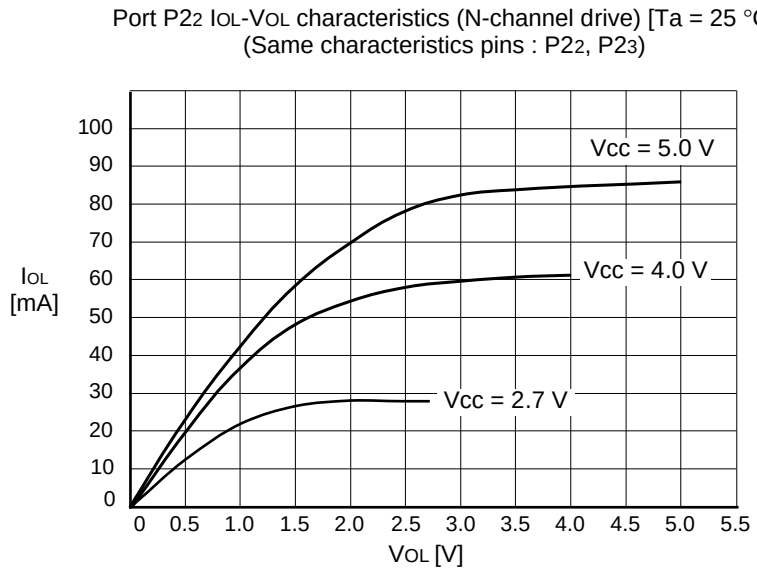


Fig. 3.2.26 N-channel open-drain output port N-channel side characteristics (Ta = 25 °C)

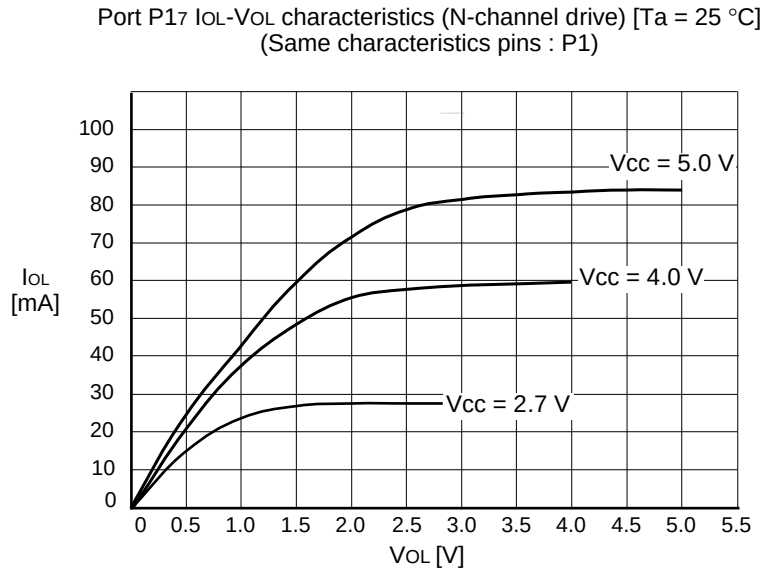


Fig. 3.2.27 CMOS large current output port N-channel side characteristics (Ta = 25 °C)

APPENDIX

3.2 Standard characteristics

3.2.7 A-D conversion standard characteristics

(1) Definition of A-D conversion accuracy

The A-D conversion accuracy is defined below.

●Relative accuracy

① **Zero transition voltage (V_{0T})**

This means an analog input voltage when the actual A-D conversion output data changes from "0" to "1".

② **Full-scale transition voltage (V_{FST})**

This means an analog input voltage when the actual A-D conversion output data changes from "1023" to "1022".

③ **Linearity error**

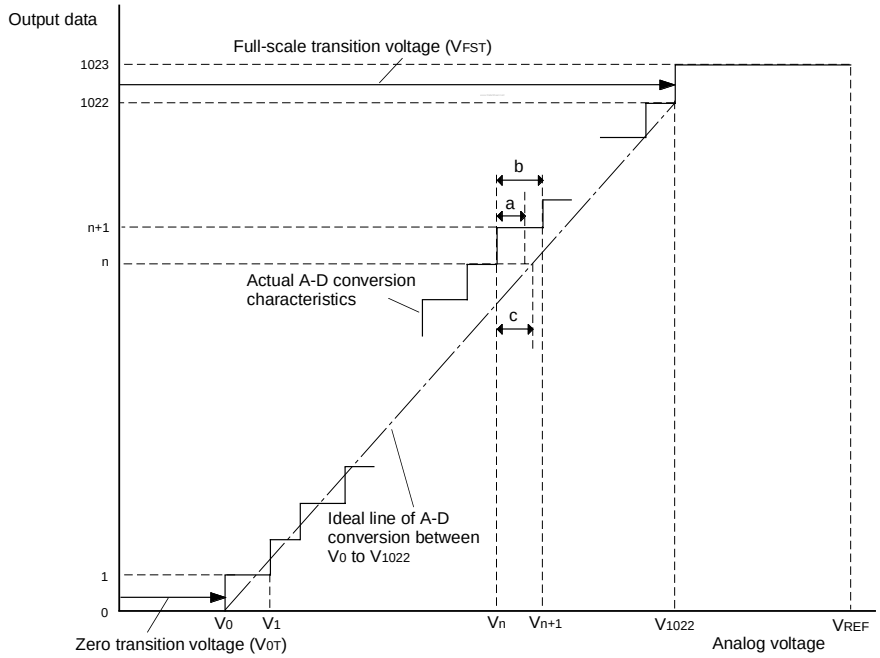
This means a deviation from the lone between V_{0T} and V_{FST} of a converted value between V_{0T} and V_{FST}.

④ **Differential non-linearity error**

This means a deviation from the input potential difference required to change a converted value between V_{0T} and V_{FST} by 1 LSB of the 1 LSB at the relative accuracy.

●Absolute accuracy

This means a deviation from the ideal characteristics between 0 to V_{REF} of actual A-D conversion characteristics.



Differential non-linearity error = $\frac{b-a}{a}$ [LSB]

Linearity error = $\frac{c}{a}$ [LSB]

1 LSB at relative accuracy = $\frac{V_{FST}-V_{0T}}{1022}$ [LSB]

1 LSB at absolute accuracy = $\frac{V_{REF}}{1024}$ [LSB]

V_n : Analog input voltage when the output data changes from "n" to "n + 1" (n = 0 to 1022)
a : 1 LSB at relative accuracy
b : $V_{n+1} - V_n$
c : Difference between the aideal V_n and actual V_n

Fig. 3.2.28 Definition of A-D conversion accuracy

(2) A-D conversion standard characteristics

Figure 3.2.29, Figure 3.2.30, and Figure 3.2.31 show the A-D conversion standard characteristics of flash memory version, mask ROM version, and PROM version, respectively.

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 256 to 257 should occur at 1280 mV, but the measured value is 2.5 mV. Accordingly, the measured point of change is 1280 + 2.5 = 1282.5 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 256 is 5.0 mV, so that the differential non-linear error is 5.0 – 5.0 = 0 mV (0 LSB).

M38507F8 A-D CONVERTER ERROR & STEP WIDTH MEASUREMENT

V_{CC} = 5.12 [V], V_{REF} = 5.12 [V]
X_{IN} = 8 [MHz], Ta = 25 [deg.]

Zero transition voltage	:	10.625 [mV]		
Full-scale transition voltage	:	5122.812 [mV]		
Differential non-linearity error	:	1.719 [mV]	:	0.344 [LSB]
Linearity error	:	−5.659 [mV]	:	−1.131 [LSB]
Absolute accuracy	:	8.906 [mV]	:	1.781 [LSB]

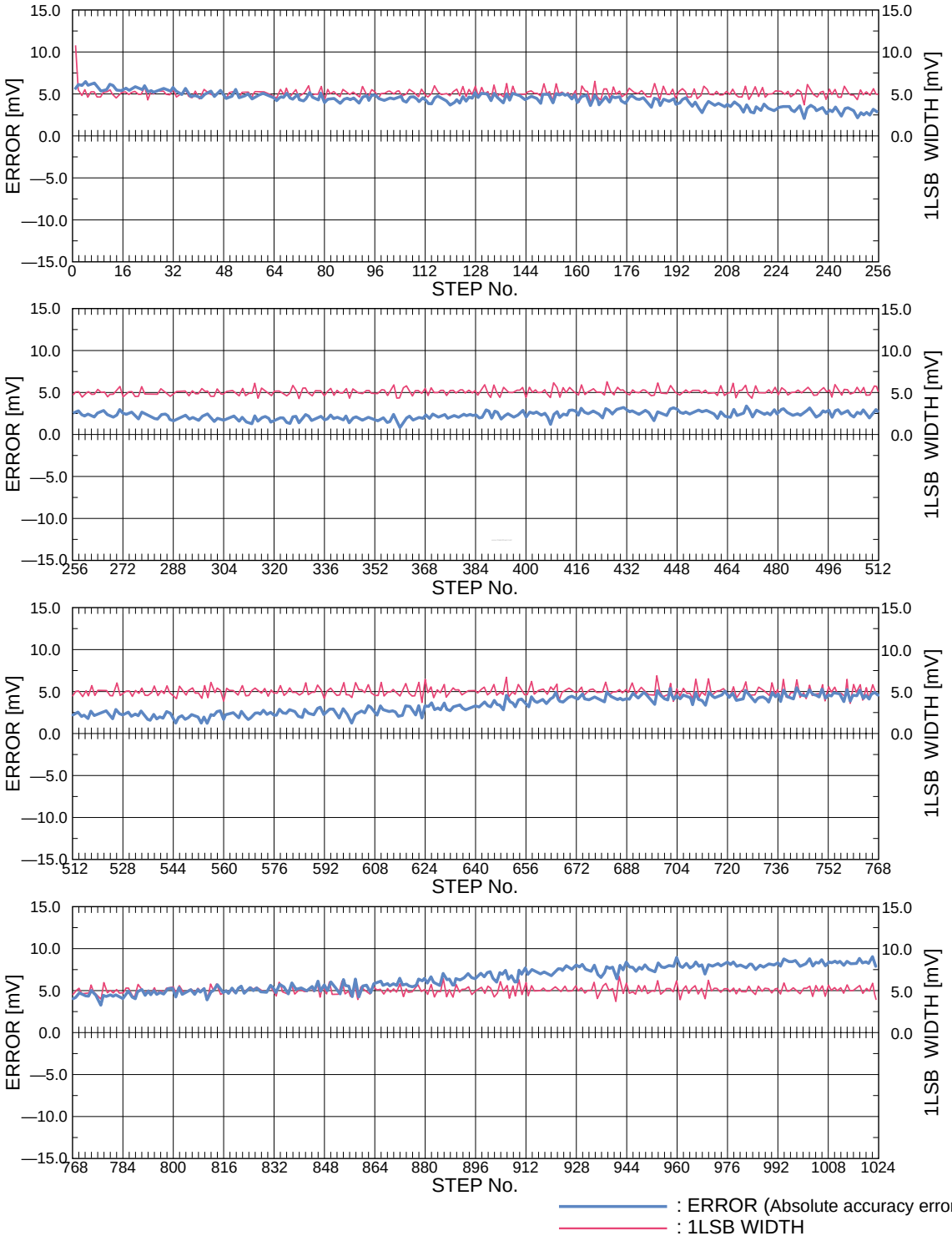


Fig. 3.2.29 Flash memory version (M38507F8) A-D conversion standard characteristics

M38503M2H, M38503M4H, M38504M6, M38507M8 A-D CONVERTER ERROR & STEP WIDTH MEASUREMENT

VCC = 5.12 [V], VREF = 5.12 [V]
XIN = 8 [MHz], Ta = 25 [deg.]

Zero transition voltage	:	10.31 [mV]	
Full-scale transition voltage	:	5118.12 [mV]	
Differential non-linearity error	:	–1.41 [mV] :	–0.28 [LSB]
Linearity error	:	–4.72 [mV] :	–0.94 [LSB]
Absolute accuracy	:	6.25 [mV] :	1.25 [LSB]

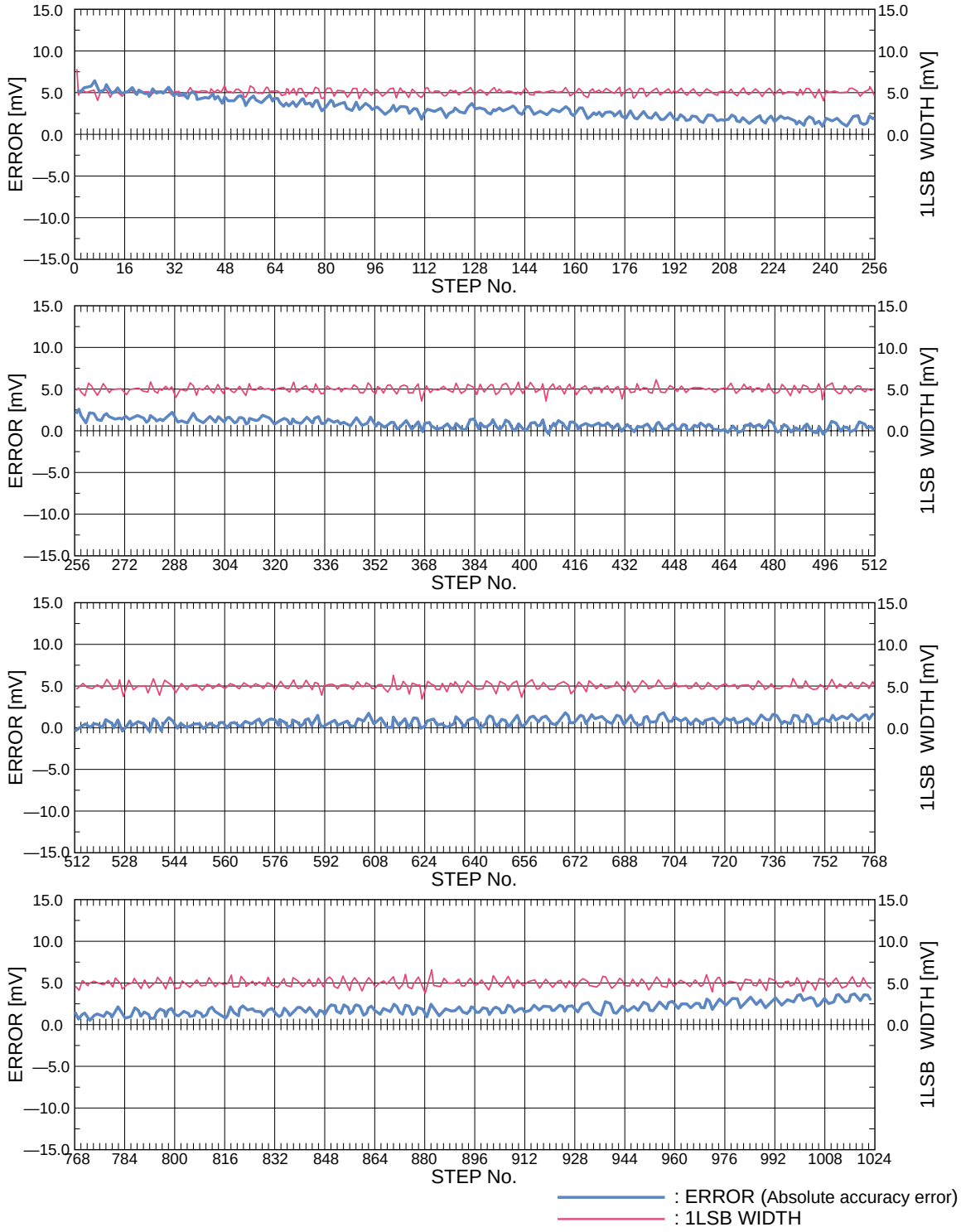


Fig. 3.2.30 Mask ROM version (M38503M2H, M38503M4H, M38504M6, M38507M8) A-D conversion standard characteristics

3.2 Standard characteristics

M38504E6 A-D CONVERTER ERROR & STEP WIDTH MEASUREMENT

V_{CC} = 5.12 [V], V_{REF} = 5.12 [V]
X_{IN} = 8 [MHz], Ta = 25 [deg.]

Zero transition voltage	:	−0.62 [mV]	
Full-scale transition voltage	:	5112.19 [mV]	
Differential non-linearity error	:	2.97 [mV] :	0.59 [LSB]
Linearity error	:	−3.41 [mV] :	−0.68 [LSB]
Absolute accuracy	:	−7.03 [mV] :	−1.41[LSB]

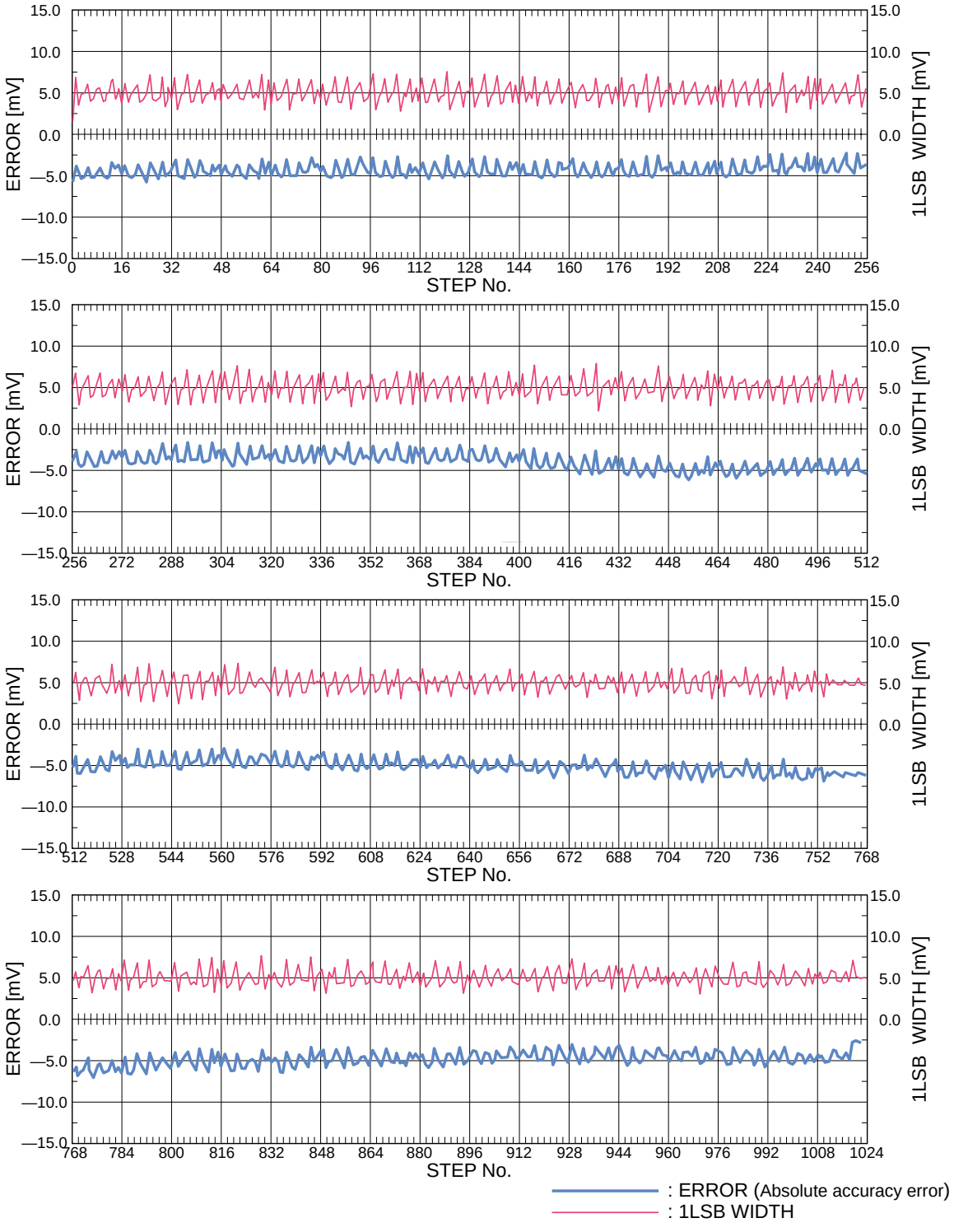


Fig. 3.2.31 PROM version (M38504E6) A-D conversion standard characteristics

3.3 Notes on use

3.3.1 Notes on input and output ports

(1) Notes in standby state

In standby state*1, do not make input levels of an I/O port "undefined", especially for I/O ports of the N-channel open-drain. When setting the N-channel open-drain port as an output, do not make input levels of an I/O port "undefined", too.

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

● **Reason**

When setting as an input port with its direction register, the transistor becomes the OFF state, which causes the ports to be the high-impedance state.

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.

In I/O ports of N-channel open-drain, when the contents of the port latch are "1", even if it is set as an output port with its direction register, it becomes the same phenomenon as the case of an input port.

✻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 Notes on use

3.3.2 Termination of unused pins

(1) Terminate unused pins

① Output ports : Open

② Input ports :
Connect each pin to V_{CC} or V_{SS} through each resistor of 1 kΩ to 10 kΩ.
As for pins whose potential affects to operation modes such as pins CNV_{SS}, INT or others, select the V_{CC} pin or the V_{SS} pin according to their operation mode.

③ I/O ports :
- Set the I/O ports for the input mode and connect them to V_{CC} or V_{SS} through each resistor of 1 kΩ to 10 kΩ.
  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.

④ 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 Input ports and 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 ② and ③ shown on the above.

② I/O ports :
When setting for the input mode, do not connect to V_{CC} or V_{SS} 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 V_{CC} (or V_{SS}).

③ I/O ports :
When setting for the input mode, do not connect multiple ports in a lump to V_{CC} or V_{SS} 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 $3A_{16}$)
•Timer XY mode register (address 23_{16})
Set the above listed registers or bits as the following sequence.

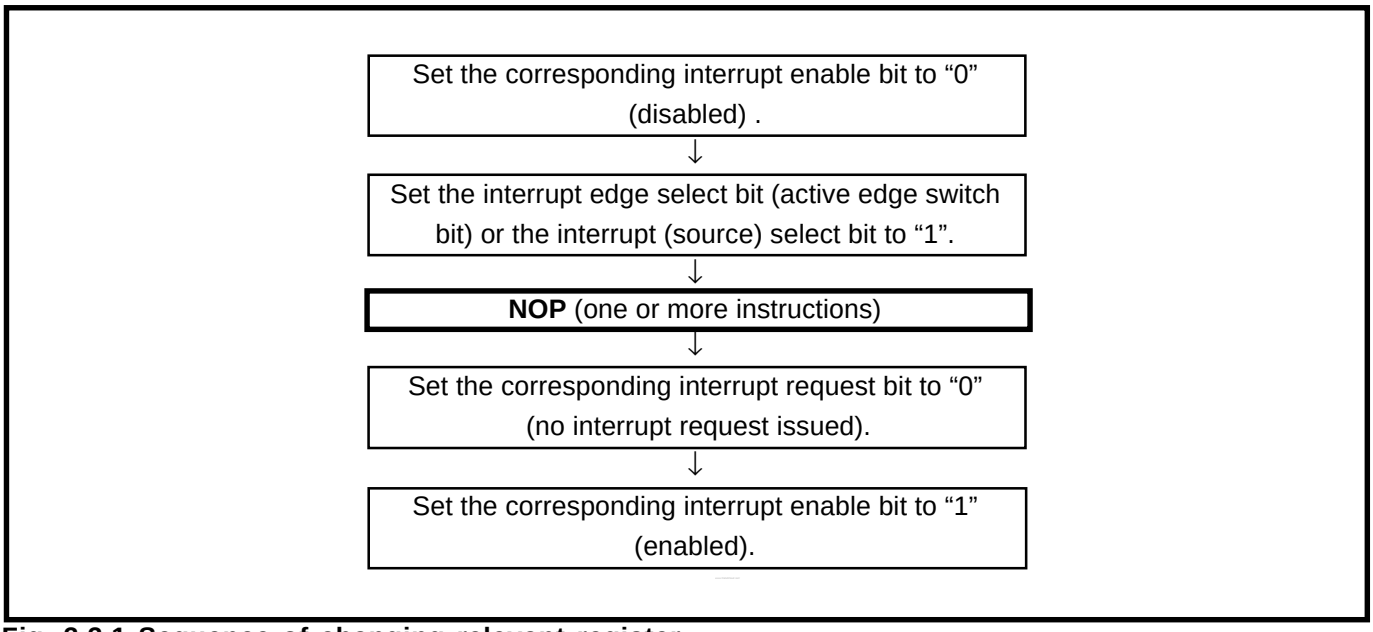


Fig. 3.3.1 Sequence of changing relevant register

■ **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 $3A_{16}$)
Timer XY mode register (address 23_{16})

•When switching interrupt sources of an interrupt vector address where two or more interrupt sources are allocated.
Concerned register: Interrupt edge selection register (address $3A_{16}$)

(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" by using a data transfer instruction, execute one or more instructions before executing the **BBC** or **BBS** instruction.

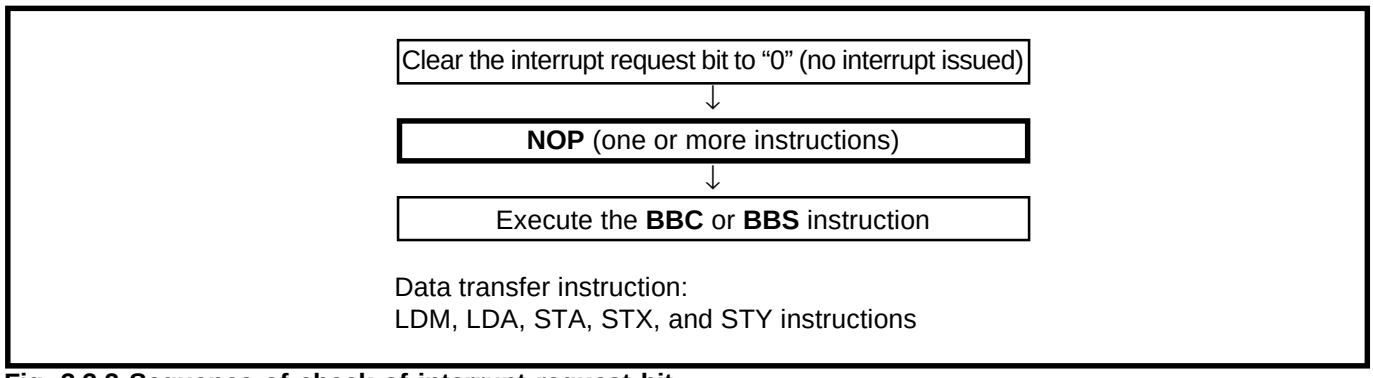


Fig. 3.3.2 Sequence of check of interrupt request bit

■ **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.

3.3.4 Notes on timer

● 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.

3.3.5 Notes on serial I/O

(1) Notes when selecting clock synchronous serial I/O (Serial I/O1)

① **Stop of transmission operation**

Clear the serial I/O1 enable bit and the transmit enable bit to "0" (Serial I/O1 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/O1 disabled), the internal transmission is running (in this case, since pins TxD, RxD, 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 TxD pin and an operation failure occurs.

② **Stop of receive operation**

Clear the receive enable bit to "0" (receive disabled), or clear the serial I/O1 enable bit to "0" (Serial I/O1 disabled).

③ **Stop of transmit/receive operation**
Clear the transmit enable bit and receive enable bit to “0” simultaneously (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/O1 disabled) (refer to (1) ①).

④ **$\overline{S_{RDY1}}$ output of reception side (Serial I/O1)**
When signals are output from the S_{RDY1} 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 $\overline{S_{RDY1}}$ output enable bit, and the transmit enable bit to “1” (transmit enabled).

(2) Notes when selecting clock asynchronous serial I/O (Serial I/O1)

① **Stop of transmission operation**
Clear the transmit enable bit to “0” (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/O1 disabled), the internal transmission is running (in this case, since pins TxD, RxD, S_{CLK1}, and S_{RDY1} 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 TxD pin and an operation failure occurs.

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).

● **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/O1 disabled), the internal transmission is running (in this case, since pins TxD, RxD, S_{CLK1}, and S_{RDY1} 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 TxD pin and an operation failure occurs.

Only receive operation is stopped.
Clear the receive enable bit to “0” (receive disabled).

(3) Setting serial I/O1 control register again (Serial I/O1)

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”.

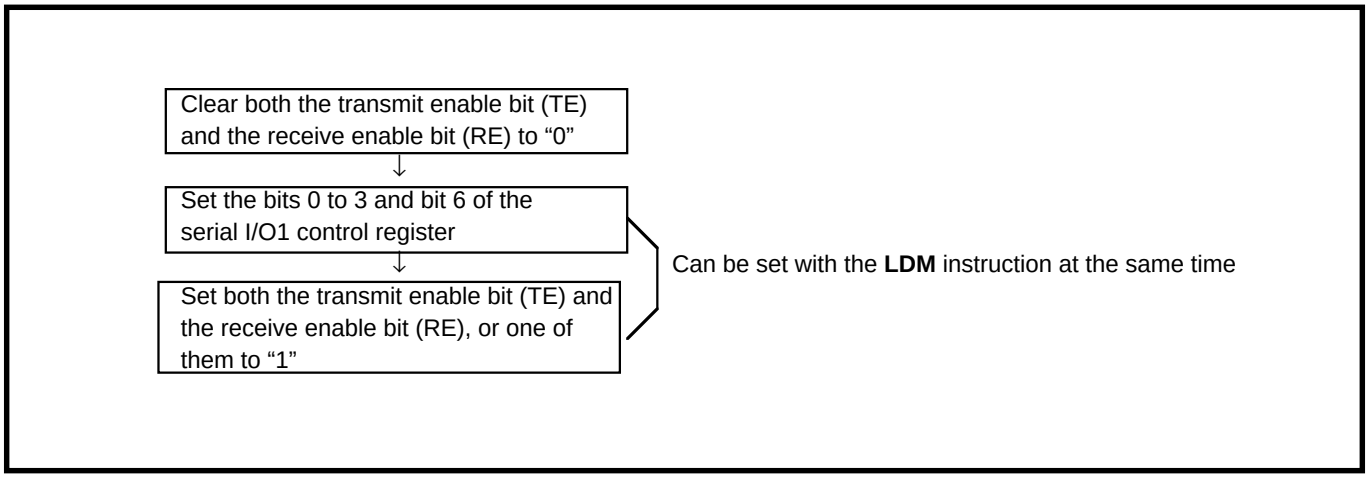


Fig. 3.3.3 Sequence of setting serial I/O1 control register again

(4) Data transmission control with referring to transmit shift register completion flag (Serial I/O1)

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.

(5) Transmit interrupt request when transmit enable bit is set (Serial I/O1)

When the transmit interrupt is used, set the transmit interrupt enable bit to transmit enabled as shown in the following sequence.

① Set the interrupt enable bit to “0” (disabled) with CLB instruction.
② Prepare serial I/O for transmission/reception.
③ Set the interrupt request bit to “0” with CLB instruction after 1 or more instruction has been executed.
④ Set the interrupt enable bit to “1” (enabled).

● Reason

When the transmission enable bit is set to “1”, the transmit buffer empty flag and transmit shift register completion flag are set to “1”. The interrupt request is generated and the transmission interrupt request bit is set regardless of which of the two timings listed below is selected as the timing for the transmission interrupt to be generated.

• Transmit buffer empty flag is set to “1”
• Transmit shift register completion flag is set to “1”

(6) Transmission control when external clock is selected (Serial I/O1 clock synchronous mode)

When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to “1” at “H” of the S_{CLK1} input level. Also, write the transmit data to the transmit buffer register (serial I/O shift register) at “H” of the S_{CLK1} input level.

(7) Transmit data writing (Serial I/O2)

In the clock synchronous serial I/O, when selecting an external clock as synchronous clock, write the transmit data to the serial I/O2 register (serial I/O shift register) at “H” of the transfer clock input level.

3.3.6 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:

$$\frac{n + 1}{2 \cdot f(X_{IN})} \text{ sec.}$$ (Count source selection bit = “0”, where n is the value set in the prescaler)

$$\frac{n + 1}{f(X_{IN})} \text{ sec.}$$ (Count source selection bit = “1”, where n is the value set in the prescaler)

3.3.7 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 AV_{SS} pin is A-D converter power source pin. Regardless of using the A-D conversion function or not, connect it as following :

• AV_{SS} : Connect to the V_{SS} line

● **Reason**

If the AV_{SS} 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.

• $f(X_{IN})$ is 500 kHz or more in middle-/high-speed mode.
• Do not execute the **STP** instruction.
• When the A-D converter is operated at low-speed mode, $f(X_{IN})$ do not have the lower limit of frequency, because of the A-D converter has a built-in self-oscillation circuit.

3.3.8 Notes on watchdog timer

●Make sure that the watchdog timer 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.

3.3.9 Notes on $\overline{\text{RESET}}$ pin

(1) 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 V_{SS} 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) Reset release after power on

When releasing the reset after power on, such as power-on reset, release reset after X_{IN} passes more than 20 cycles in the state where the power supply voltage is 2.7 V or more and the X_{IN} oscillation is stable.

● Reason

To release reset, the $\overline{\text{RESET}}$ pin must be held at an “L” level for 20 cycles or more of X_{IN} in the state where the power source voltage is between 2.7 V and 5.5 V, and X_{IN} oscillation is stable.

3.3.10 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”)

When using the oscillation stabilizing time set after STP instruction released bit set to “1”, evaluate time to stabilize oscillation of the used oscillator and set the value to the timer 1 and prescaler 12.

■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 for approximately 8,000 cycles of the X_{IN} input is reserved at restoration from the stop mode. 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.

3.3.11 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.

3.3.12 Notes on CPU rewrite mode of flash memory version

(1) Operation speed

During CPU rewrite mode, set the internal clock frequency 4MHz or less by using the main clock division ratio selection bits (bits 6, 7 at address $003B_{16}$).

(2) Instructions inhibited against use

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

(3) Interrupts inhibited against use

The interrupts cannot be used during 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 addresses $FFFC_{16}$ and $FFFD_{16}$ in boot ROM area.

3.3.13 Notes on restarting oscillation

■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 0038_{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.

3.3.14 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".

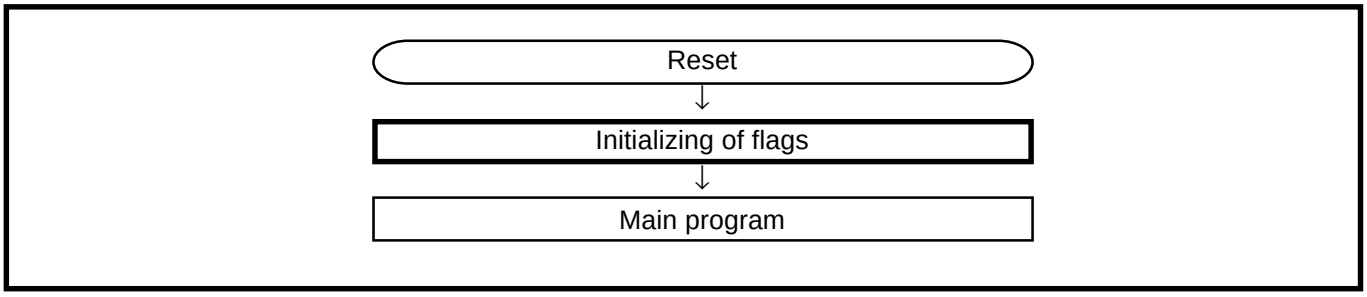


Fig. 3.3.4 Initialization of processor status register

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.

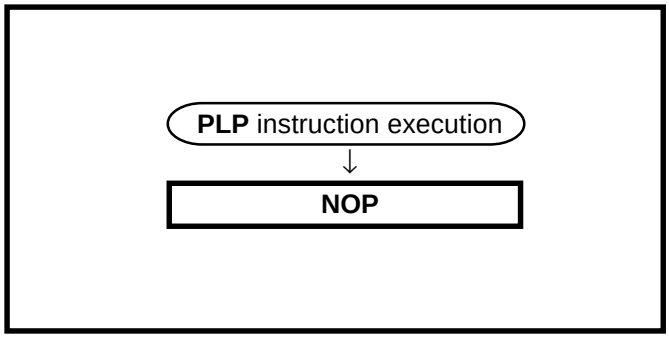


Fig. 3.3.5 Sequence of PLP instruction execution

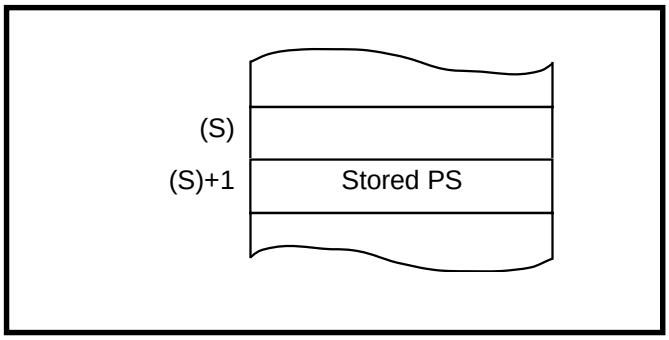


Fig. 3.3.6 Stack memory contents after PHP instruction execution

(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.

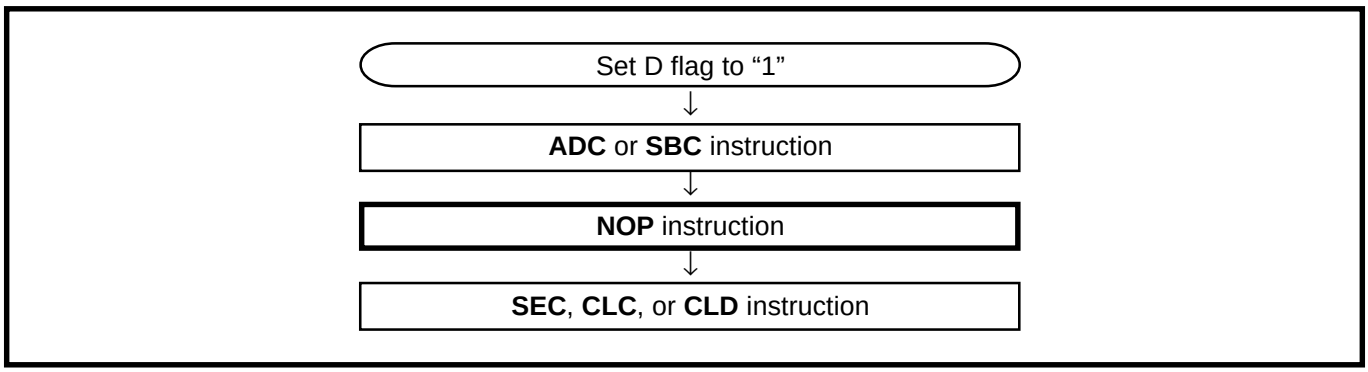


Fig. 3.3.7 Status flag at decimal calculations

(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 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.

(7) Instruction Execution Time

The instruction execution time is obtained by multiplying the frequency 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 frequency of the internal clock ϕ is half of the X_{IN} frequency in high-speed mode.

3.3 Notes on use

3.3.15 EPROM Version/One Time PROM Version/Flash Memory Version

The CNVss pin is connected to the internal memory circuit block by a low-ohmic resistance, since it has the multiplexed function to be a programmable power source pin (V_{PP} pin) as well.
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.

3.3.16 Handling of Source Pins

In order to avoid a latch-up occurrence, connect a capacitor suitable for high frequencies as bypass capacitor between power source pin (V_{CC} pin) and GND pin (V_{SS} pin) and between power source pin (V_{CC} pin) and analog power source input pin (AV_{SS} 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–0.1μF is recommended.

3.3.17 Differences between 3850 group (standard) and 3850 group (spec. H)

(1) The absolute maximum ratings of 3850 group (spec. H) is smaller than that of 3850 group (standard).
- •Power source voltage Vcc = 0.3 to 6.5 V
- •CNVss input voltage V_I = –0.3 to Vcc +0.3 V

(2) The oscillation circuit constants of X_{IN}-X_{OUT}, X_{CIN}-X_{COUT} may be some differences between 3850 group (standard) and 3850 group (spec. H).
(3) Do not write any data to the reserved area and the reserved bit. (Do not change the contents after reset.)
(4) Fix bit 3 of the CPU mode register to “1”.
(5) Be sure to perform the termination of unused pins.

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) Package

Select the smallest possible package to make the total wiring length short.

● **Reason**

The wiring length depends on a microcomputer package. Use of a small package, for example QFP and not DIP, makes the total wiring length short to reduce influence of noise.

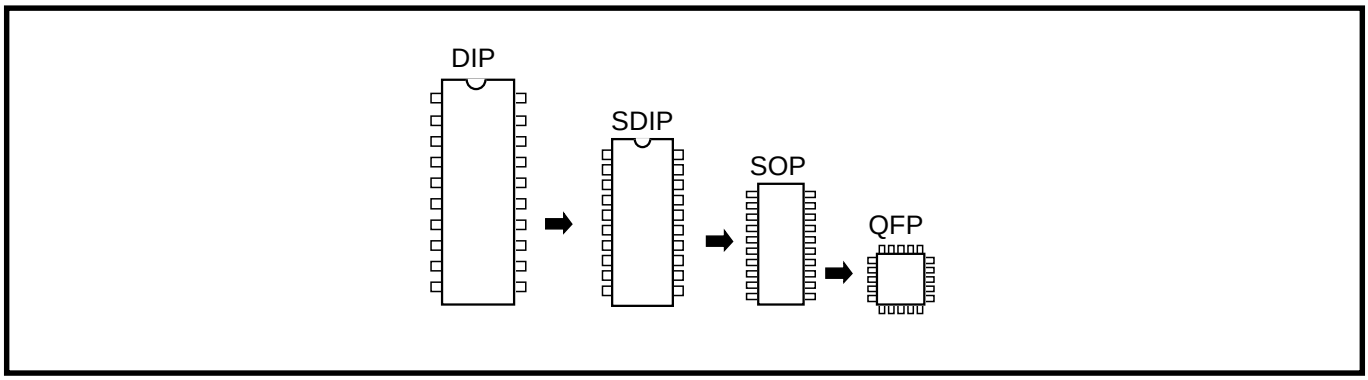


Fig. 3.4.1 Selection of packages

(2) 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 20mm).

● **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.

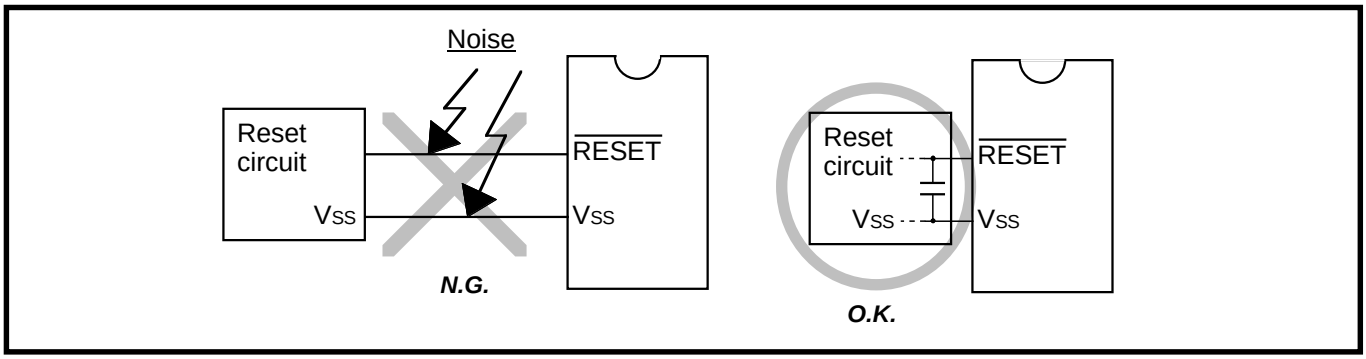


Fig. 3.4.2 Wiring for the $\overline{\text{RESET}}$ pin

3.4 Countermeasures against noise

(3) 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.

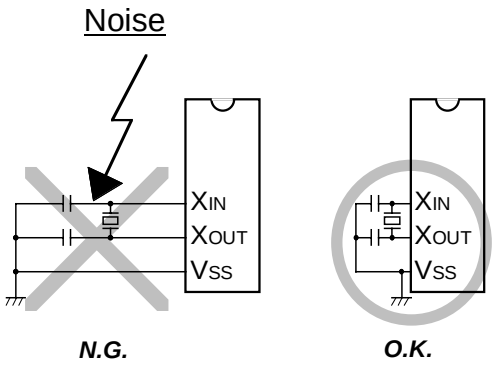


Fig. 3.4.3 Wiring for clock I/O pins

(4) 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.

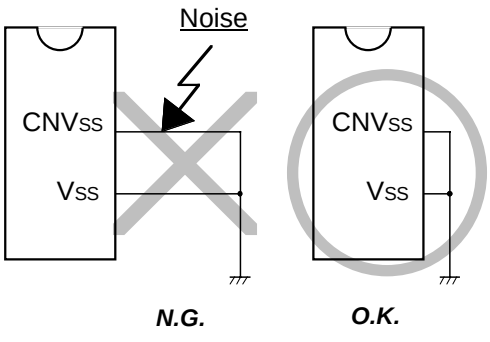


Fig. 3.4.4 Wiring for CNVss pin

(5) Wiring to V_{PP} pin of One Time PROM version, EPROM version, and Flash memory version

Connect an approximately 5 kΩ resistor to the V_{PP} pin the shortest possible in series and also to the V_{SS} pin. When not connecting the resistor, make the length of wiring between the V_{PP} pin and the V_{SS} pin the shortest possible.

Note: Even when a circuit which included an approximately 5 kΩ resistor is used in the Mask ROM version, the microcomputer operates correctly.

● **Reason**

The V_{PP} pin of the One Time PROM, the EPROM version, and the flash memory version is the power source input pin for the built-in PROM. When programming in the built-in PROM, the impedance of the V_{PP} pin is low to allow the electric current for writing flow into the PROM. Because of this, noise can enter easily. If noise enters the V_{PP} pin, abnormal instruction codes or data are read from the built-in PROM, which may cause a program runaway.

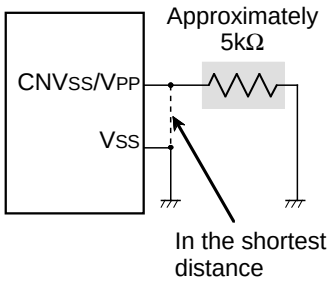


Fig. 3.4.5 Wiring for the V_{PP} pin of the One Time PROM version, the EPROM version, and the flash memory version

3.4.2 Connection of bypass capacitor across Vss line and Vcc line

Connect an approximately 0.1 μF bypass capacitor across the V_{SS} line and the V_{CC} line as follows:

- Connect a bypass capacitor across the V_{SS} pin and the V_{CC} pin at equal length.
- Connect a bypass capacitor across the V_{SS} pin and the V_{CC} pin with the shortest possible wiring.
- Use lines with a larger diameter than other signal lines for V_{SS} line and V_{CC} line.
- Connect the power source wiring via a bypass capacitor to the V_{SS} pin and the V_{CC} pin.

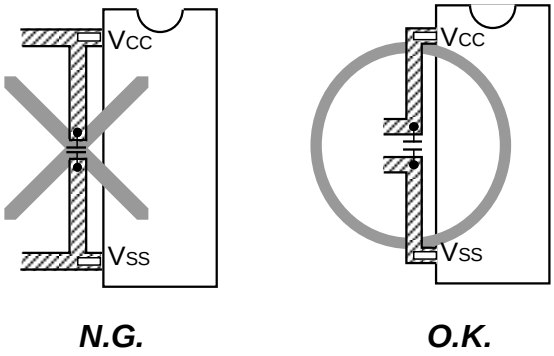


Fig. 3.4.6 Bypass capacitor across the Vss line and the Vcc line

3.4 Countermeasures against noise

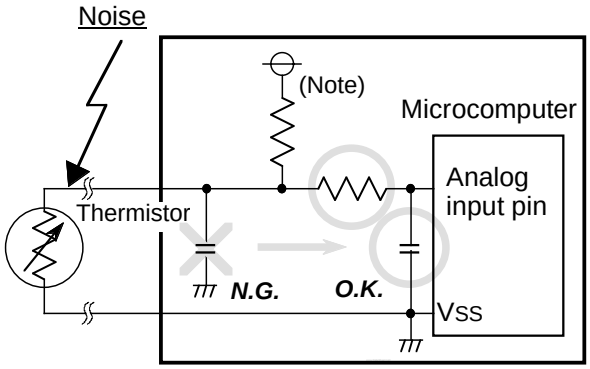
3.4.3 Wiring to analog input pins

- 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 V_{SS} pin and the analog input pin. Besides, connect the capacitor to the V_{SS} pin as close as possible. Also, connect the capacitor across the analog input pin and the V_{SS} 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 V_{SS} pin is grounded at a position far away from the V_{SS} pin, noise on the GND line may enter a microcomputer through the capacitor.



Note : The resistor is used for dividing resistance with a thermistor.

Fig. 3.4.7 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.

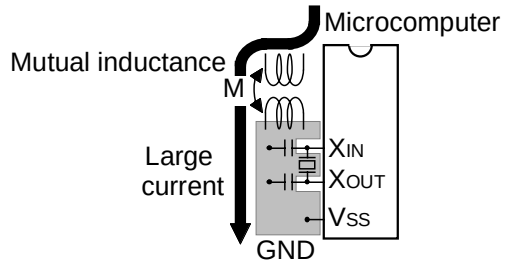


Fig. 3.4.8 Wiring for a large current signal line

(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.

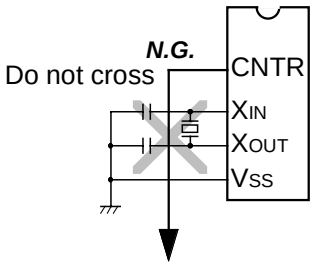


Fig. 3.4.9 Wiring of $\overline{\text{RESET}}$ pin

3.4 Countermeasures against noise

(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.

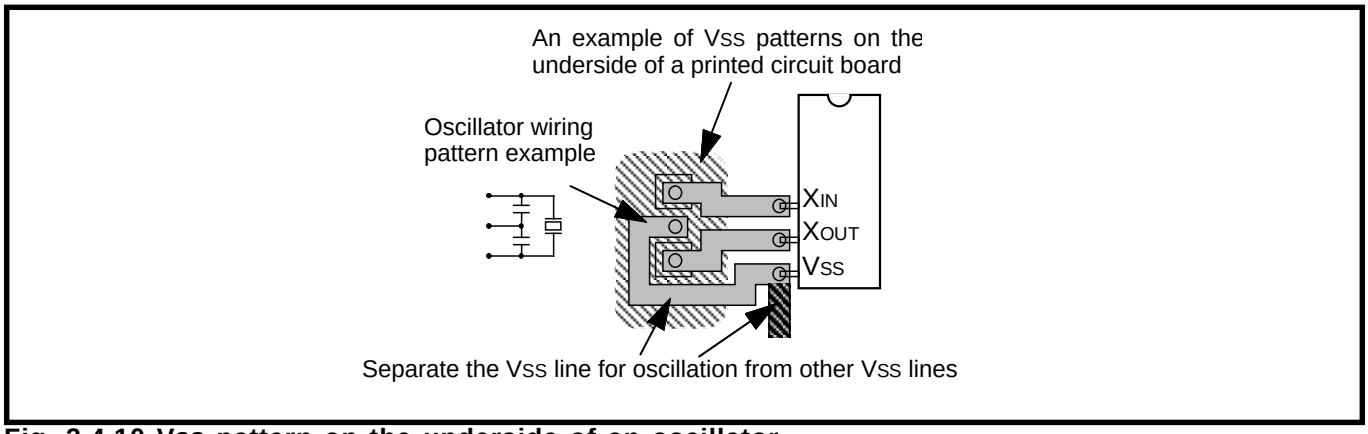


Fig. 3.4.10 Vss pattern on the underside of an oscillator

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 at fixed periods.

Note: When a direction register is set for input port again at fixed periods, a several-nanosecond short pulse may be output from this port. If this is undesirable, connect a capacitor to this port to remove the noise pulse.

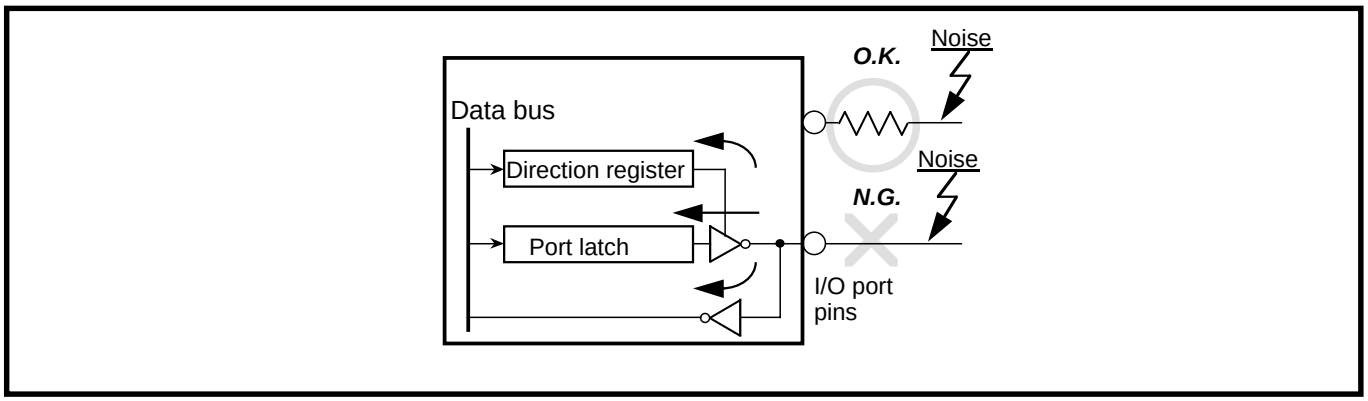


Fig. 3.4.11 Setup for I/O ports

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.

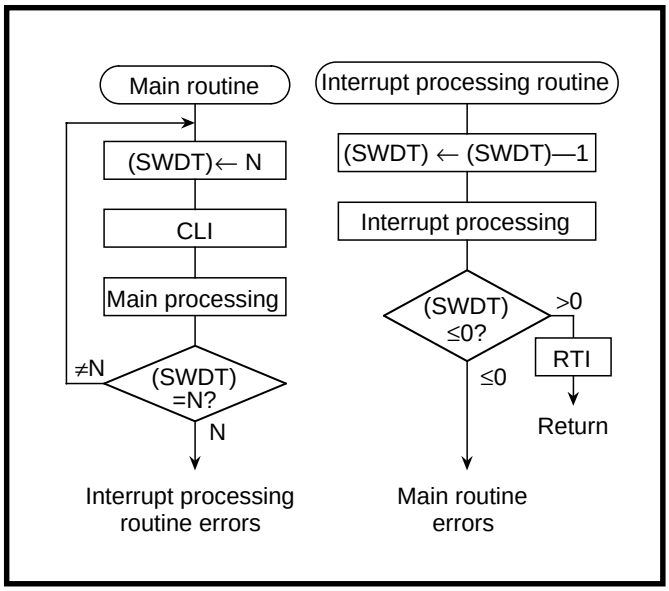


Fig. 3.4.12 Watchdog timer by software

3.5 List of registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (i = 0, 1, 2, 3, 4)
(Pi: addresses 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{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	○	○

Note: When reading bit 5, 6 or 7 of ports 3 and 4, the contents are undefined.

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)
(PiD: addresses 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{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

Serial I/O2 control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O2 control register 1 (SIO2CON1: address 15_{16})

b	Name	Functions	At reset	R	W
0	Internal synchronous clock selection bits	b2b1b0 0 0 0: f(X_{IN})/8 0 0 1: f(X_{IN})/16 0 1 0: f(X_{IN})/32 0 1 1: f(X_{IN})/64 1 1 0: f(X_{IN})/128 1 1 1: f(X_{IN})/256	0	○	○
1			0	○	○
2			0	○	○
3	Serial I/O2 port selection bit	0: I/O port ($P0_1$, $P0_2$) 1: S_{OUT2}, S_{CLK2} signal output	0	○	○
4	$\overline{S_{RDY2}}$ output enable bit	0: I/O port ($P0_3$) 1: $\overline{S_{RDY2}}$ 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	$P0_1/S_{OUT2}$, $P0_2/S_{CLK2}$ P-channel output disable bit	0: CMOS output 1: N-channel open-drain output	0	○	○

Fig. 3.5.3 Structure of Serial I/O2 control register 1

Serial I/O2 control register 2

b7 b6 b5 b4 b3 b2 b1 b0

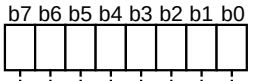
Serial I/O2 control register 2 (SIO2CON2: address 16_{16})

b	Name	Functions	At reset	R	W
0	Optional transfer bits	b2b1b0 0 0 0: 1 bit 0 0 1: 2 bit 0 1 0: 3 bit 0 1 1: 4 bit 1 0 0: 5 bit 1 0 1: 6 bit 1 1 0: 7 bit 1 1 1: 8 bit	1	○	○
1			1	○	○
2			1	○	○
3	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
4			0	○	×
5			0	○	×
6	Serial I/O2 I/O comparison signal control bit	0: $P4_3$ I/O 1: S_{CMP2} output	0	○	○
7	S_{OUT2} pin control bit ($P0_1$)	0: Output active 1: Output high-impedance	0	○	○

Fig. 3.5.4 Structure of Serial I/O2 control register 2

3.5 List of registers

Serial I/O2 register

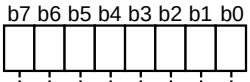


Serial I/O2 register
(SIO2: address 17_{16})

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. 3.5.5 Structure of Serial I/O2 register

Transmit/Receive buffer register



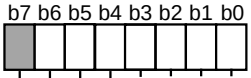
Transmit/Receive buffer register (TB/RB: address 18_{16})

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. 3.5.6 Structure of Transmit/Receive buffer register

Serial I/O1 status register



Serial I/O1 status register (SIOSTS: address 19_{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. 3.5.7 Structure of Seial I/O1 status register

Serial I/O1 control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O1 control register
(SIOCON: address $1A_{16}$)

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/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 ($P2_7$) 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 ($P2_4$ to $P2_7$: normal I/O pins) 1: Serial I/O1 enabled ($P2_4$ to $P2_7$: Serial I/O pins)	0	○	○

Fig. 3.5.8 Structure of Seial I/O1 control register

UART control register

b7 b6 b5 b4 b3 b2 b1 b0

UART control register
(UARTCON: address $1B_{16}$)

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	$P2_5$/TxD P-channel output disable bit (POFF)	In output mode 0: CMOS output 1: N-channel open-drain output	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. 3.5.9 Structure of UART control register

Baud rate generator

b7 b6 b5 b4 b3 b2 b1 b0

Baud rate generator(BRG : address $1C_{16}$)

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	○	○

Fig. 3.5.10 Structure of Baud rate generator

PWM control register

b7 b6 b5 b4 b3 b2 b1 b0

PWM control register (PWMCON: address $1D_{16}$)

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(X_{IN}) 1 : f(X_{IN})/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. 3.5.11 Structure of PWM control register

PWM prescaler

b7 b6 b5 b4 b3 b2 b1 b0

PWM prescaler
(PREPWM: address $1E_{16}$)

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. 3.5.12 Structure of PWM prescaler

3.5 List of registers

PWM register

b7 b6 b5 b4 b3 b2 b1 b0

PWM register
(PWM: address $1F_{16}$)

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. 3.5.13 Structure of PWM register

Prescaler 12, Prescaler X, Prescaler Y

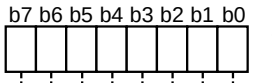
b7 b6 b5 b4 b3 b2 b1 b0

Prescaler 12 (PRE12), Prescaler X (PREX), Prescaler Y (PREY)
(addresses 20_{16}, 24_{16}, 26_{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. 3.5.14 Structure of Prescaler 12, Prescaler X, Prescaler Y

Timer 1

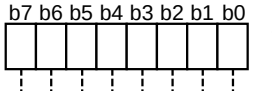


Timer 1
(T1: address 21_{16})

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. 3.5.15 Structure of Timer 1

Timer 2

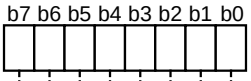


Timer 2
(T2: address 22_{16})

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

Fig. 3.5.16 Structure of Timer 2

Timer XY mode register



Timer XY mode register
(TM: address 23_{16})

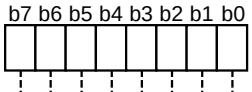
b	Name	Functions	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0: Timer mode 0 1: Pulse output mode 1 0: Event counter mode 1 1: Pulse width measurement mode	0	○	○
1			0	○	○
2	$CNTR_0$ active edge switch bit	Refer to Table 3.5.1	0	○	○
3	Timer X count stop bit	0: Count start 1: Count stop	0	○	○
4	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. 3.5.17 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	Set value	Timer function	$CNTR_0$ / $CNTR_1$ interrupt request occurrence source
Timer mode	"0"	No influence to timer count	$CNTR_0$/$CNTR_1$ input signal falling edge
	"1"	No influence to timer count	$CNTR_0$/$CNTR_1$ input signal rising edge
Pulse output mode	"0"	Pulse output start: Beginning at "H" level	Output signal falling edge count
	"1"	Pulse output start: Beginning at "L" level	Output signal rising edge count
Event counter mode	"0"	Rising edge count	Input signal falling edge count
	"1"	Falling edge count	Input signal rising edge count
Pulse width measurement mode	"0"	"H" level width measurement	Input signal falling edge count
	"1"	"L" level width measurement	Input signal rising edge count

Timer X, Timer Y

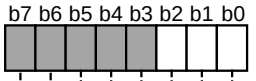


Timer X, Timer Y
(TX, TY: addresses 25_{16}, 27_{16})

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. 3.5.18 Structure of Timer X, Timer Y

Timer count source selection register



Timer count source selection register
(TCSS: address 28_{16})

b	Name	Functions	At reset	R	W
0	Timer X count source selection bit	0: $f(X_{IN})/16$ ($f(X_{CIN})/16$ at low-speed mode) 1: $f(X_{IN})/2$ ($f(X_{CIN})/2$ at low-speed mode)	0	○	○
1	Timer Y count source selection bit	0: $f(X_{IN})/16$ ($f(X_{CIN})/16$ at low-speed mode) 1: $f(X_{IN})/2$ ($f(X_{CIN})/2$ at low-speed mode)	0	○	○
2	Timer 12 count source selection bit	0: $f(X_{IN})/16$ ($f(X_{CIN})/16$ at low-speed mode) 1: $f(X_{CIN})$	0	○	○
3	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.19 Structure of Timer count source selection register

3.5 List of registers

A-D control register

b7 b6 b5 b4 b3 b2 b1 b0

A-D control register
(ADCON: address 34_{16})

b	Name	Functions	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0: $P3_0/AN_0$ 0 0 1: $P3_1/AN_1$ 0 1 0: $P3_2/AN_2$ 0 1 1: $P3_3/AN_3$ 1 0 0: $P3_4/AN_4$	0	○	○
1			0	○	○
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	AD conversion completion bit	0: Conversion in progress 1: Conversion completed	1	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bist are read out, the contents are “0”.		0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.20 Structure of A-D control register

A-D conversion register (low-order)

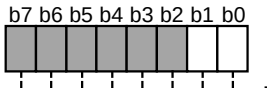
b7 b6 b5 b4 b3 b2 b1 b0

A-D conversion register (low-order)
(ADL: address 35_{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 … 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. 3.5.21 Structure of A-D conversion low-order register

A-D conversion register (high-order)

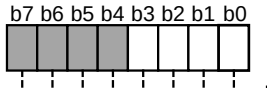


A-D conversion register (high-order)
(ADH: address 36_{16})

b	Functions	At reset	R	W
0	This is A-D conversion result stored bits. This is read exclusive register.	Undefined	○	×
1	b7 10-bit read b0 (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		0	○	×

Fig. 3.5.22 Structure of A-D conversion high-order register

MISRG



MISRG
(MISRG: address 38_{16})

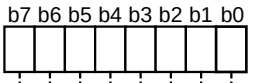
b	Name	Functions	At reset	R	W
0	Oscillatin stabilizing time set after STP instruction released bit	0: Automatically set (**Note 1**) 1: Autimatically set disabled	0	○	○
1	Middle-speed mode automatic switch set bit	0: Not set automatically 1: Automatic switching enabled (**Note 2**)	0	○	○
2	Middle-speed 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 2**)	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: When automatic switch to middle-speed mode from low-speed mode occurs, the values of CPU mode register ($3B_{16}$) change.

Fig. 3.5.23 Structure of MISRG

3.5 List of registers

Watchdog timer control register

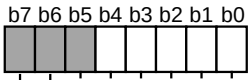


Watchdog timer control register
(WDTCON: address 39_{16})

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: $f(X_{IN})/16$ or $f(X_{CIN})/16$	0	○	○

Fig. 3.5.24 Structure of Watchdog timer control register

Interrupt edge selection register



Interrupt edge selection register
(INTEDGE: address $3A_{16}$)

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	INT_2 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
3	INT_3 active edge selection bit	0: Falling edge active 1: Rising edge active	0	○	○
4	Serial I/O2/INT_3 interrupt source bit	0: INT_3 interrupt selected 1: Serial I/O2 interrupt selected	0	○	○
5	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.25 Structure of Interrupt edge selection register

CPU mode register

b7	b6	b5	b4	b3	b2	b1	b0
				1			

CPU mode register
(CPUM: address $3B_{16}$)

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. 3.5.26 Structure of CPU mode register

Interrupt request register 1

b7	b6	b5	b4	b3	b2	b1	b0

Interrupt request register 1
(IREQ1 : address $3C_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	When writing to this bit, set "0" to this bit.		0	○	✻
2	INT_1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	INT_2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_3/Serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	When writing to this bit, set "0" to this bit.		0	○	✻
6	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer Y 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. 3.5.27 Structure of Interrupt request register 1

3.5 List of registers

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2
(IREQ2 : address $3D_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	Timer 2 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 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	$CNTR_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	$CNTR_1$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	A-D converter 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. 3.5.28 Structure of Interrupt request register 2

Interrupt control register 1

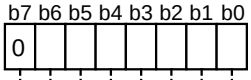
b7 b6 b5 b4 b3 b2 b1 b0
0 0

Interrupt control register 1
(ICON1 : address $3E_{16}$)

b	Name	Functions	At reset	R	W
0	INT_0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Fix this bit to “0”.		0	○	○
2	INT_1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_3/Serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Fix this bit to “0”.		0	○	○
6	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 3.5.29 Structure of Interrupt control register 1

Interrupt control register 2

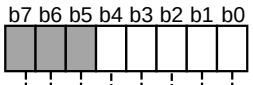


Interrupt control register 2
(ICON2 : address $3F_{16}$)

b	Name	Functions	At reset	R	W
0	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	Timer 2 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 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	$CNTR_0$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	$CNTR_1$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	A-D converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 3.5.30 Structure of Interrupt control register 2

Flash memory control register



Flash memory control register
(FMCR : address $0FFE_{16}$)

b	Name	Functions	At reset	R	W
0	RY/$\overline{BY}$ status flag	0 : Busy (being written or erased) 1 : Ready	1	○	×
1	CPU rewrite mode select bit **(Note 1)**	0 : Normal mode (Software commands invalid) 1 : CPU rewrite mode (Software commands acceptable)	0	○	○
2	CPU rewrite mode entry flag	0: Normal mode 1: CPU rewrite mode	0	○	×
3	Flash memory reset bit **(Note 2)**	0: Normal operation 1: Reset	0	○	○
4	User area/Boot area selection bit	0: User ROM area 1: Boot ROM area	0	○	○
5	Nothing is arranged for these bits. When write, set “0”. When these bits are read out, the contents are undefined.		Undefined	×	×
6			Undefined	×	×
7			Undefined	×	×

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”. Set this bit to “0” subsequently after setting it to “1” (reset).

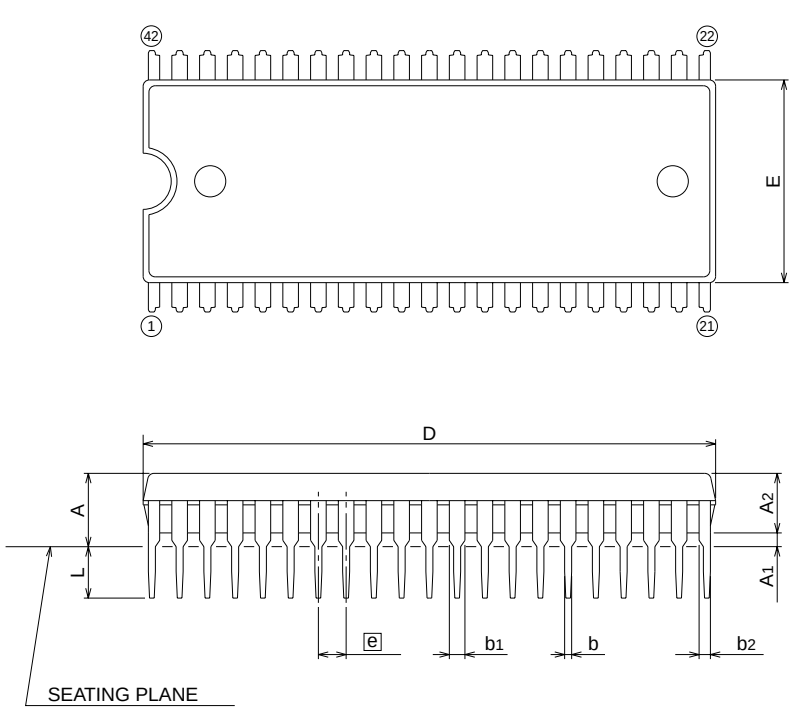
Fig. 3.5.31 Structure of Flash memory control register

3.6 Package outline

42P4B (MMP)

Plastic 42pin 600mil SDIP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SDIP42-P-600-1.78	–	4.1	Alloy 42/Cu Alloy

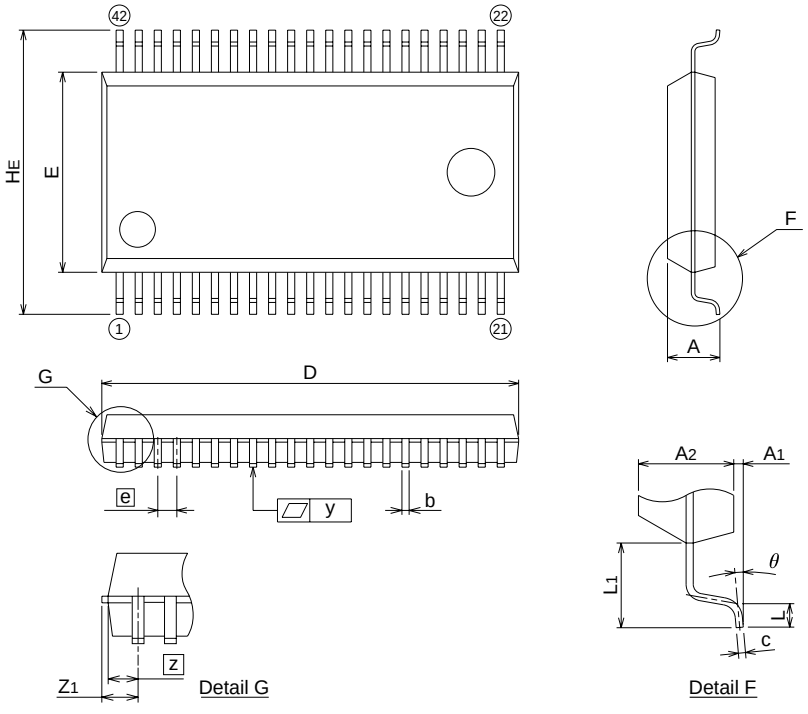


Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	5.5
A_1	0.51	–	–
A_2	–	3.8	–
b	0.35	0.45	0.55
b_1	0.9	1.0	1.3
b_2	0.63	0.73	1.03
c	0.22	0.27	0.34
D	36.5	36.7	36.9
E	12.85	13.0	13.15
e	–	1.778	–
e_1	–	15.24	–
L	3.0	–	–
θ	0°	–	15°

42P2R-A/E

Plastic 42pin 450mil SSOP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SSOP42-P-450-0.80	–	0.63	Alloy 42

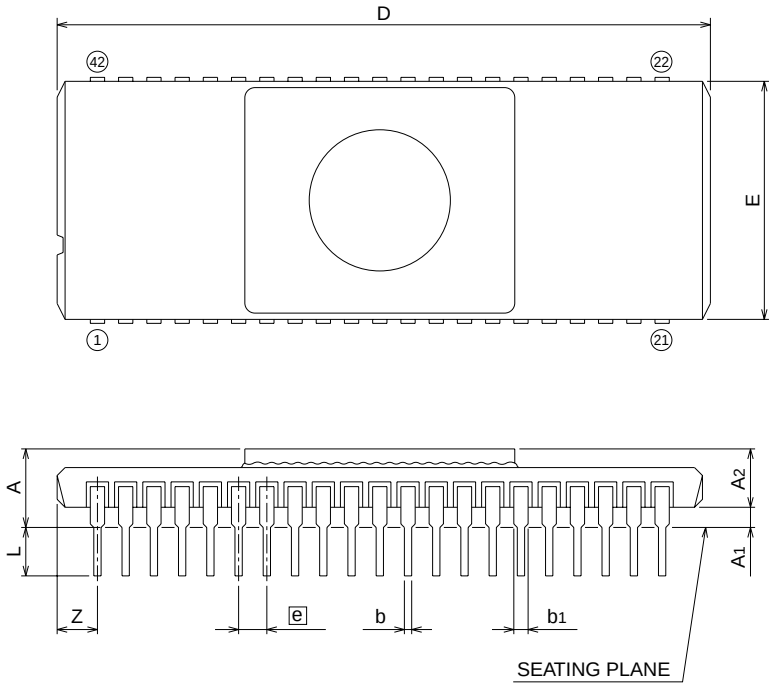


Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	2.4
A_1	0.05	–	–
A_2	–	2.0	–
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	17.3	17.5	17.7
E	8.2	8.4	8.6
e	–	0.8	–
H_E	11.63	11.93	12.23
L	0.3	0.5	0.7
L_1	–	1.765	–
z	–	0.75	–
Z_1	–	–	0.9
y	–	–	0.15
θ	0°	–	10°
b_2	–	0.5	–
e_1	–	11.43	–
l_2	1.27	–	–

42S1B-A

Metal seal 42pin 600mil DIP

EIAJ Package Code	JEDEC Code	Weight(g)
WDIP42-C-600-1.78	–	



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	5.0
A_1	1.0	–	–
A_2	–	–	3.44
b	0.38	0.46	0.54
b_1	0.7	0.8	0.9
c	0.17	0.25	0.33
D	–	–	41.1
E	–	–	15.8
e	–	1.778	–
e_1	–	15.24	–
L	3.05	–	–
Z	–	–	3.05

3.7 Machine instructions

Symbol	Function	Details	Addressing mode																		Processor status register							
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R			7	6	5	4	3	2	1	0
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
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				N	V	•	•	•	•	Z	C
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				N	•	•	•	•	•	Z	•
ASL	7 0 C ← ← 0	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				N	•	•	•	•	•	Z	C
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				M_7	M_6	•	•	•	•	Z	•
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																									M_7	M_6	•	•	•	•	Z	•
																											30	2	2				•	•	•	•	•	•	•	•
																											D0	2	2				•	•	•	•	•	•	•	•

3.7 Machine instructions

Symbol	Function	Details	Addressing mode																													Processor status register							
			IMP			IMM			A			BIT, A			ZP			BIT, ZP			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	#	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.																																													10	2	2				•	•	•	•	•	•	•	•
BRA	PC ← PC ± offset	This instruction branches to the appointed address. The branch address is specified by a relative address.																																													80	4	2				•	•	•	•	•	•	•	•
BRK	B ← 1 (PC) ← (PC) + 2 M(S) ← PCH S ← S – 1 M(S) ← PCL S ← S – 1 M(S) ← PS S ← S – 1 I← 1 PCL ← ADL PCH ← ADH	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																																													•	•	•	1	•	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.																																												50	2	2				•	•	•	•	•	•	•	•
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.																																												70	2	2				•	•	•	•	•	•	•	•
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																																													•	•	•	•	•	•	•	0
CLD	D ← 0	This instruction clears D.	D8	2	1																																													•	•	•	•	0	•	•	•
CLI	I ← 0	This instruction clears I.	58	2	1																																													•	•	•	•	•	0	•	•
CLT	T ← 0	This instruction clears T.	12	2	1																																													•	•	0	•	•	•	•	•
CLV	V ← 0	This instruction clears V.	B8	2	1																																													•	0	•	•	•	•	•	•
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				D5	4	2				CD	4	3	DD	5	3	D9	5	3							C1	6	2	D1	6	2							N	•	•	•	•	•	Z	C
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																																	N	•	•	•	•	•	Z	•
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										EC	4	3																					N	•	•	•	•	•	Z	C
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										CC	4	3																					N	•	•	•	•	•	Z	C
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				D6	6	2				CE	6	3	DE	7	3																		N	•	•	•	•	•	Z	•

APPENDIX

3.7 Machine instructions

3.7 Machine instructions

Symbol	Function	Details	Addressing mode																										Processor status register							
			IMP			IMM			A			BIT, A			ZP			BIT, ZP			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	#	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																																														N	•	•	•	•	•	Z	•
DEY	Y ← Y – 1	This instruction subtracts one from the current contents of Y.	88	2	1																																														N	•	•	•	•	•	Z	•
DIV	A ← (M(zz + X + 1), M(zz + X)) / A M(S) ← one's complement of Remainder S ← S – 1	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				55	4	2				4D	4	3	5D	5	3	59	5	3							41	6	2	51	6	2							N	•	•	•	•	•	Z	•
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				F6	6	2				EE	6	3	FE	7	3																		N	•	•	•	•	•	Z	•
INX	X ← X + 1	This instruction adds one to the contents of X.	E8	2	1																																														N	•	•	•	•	•	Z	•
INY	Y ← Y + 1	This instruction adds one to the contents of Y.	C8	2	1																																														N	•	•	•	•	•	Z	•
JMP	If addressing mode is ABS PCL ← ADL PCH ← ADH If addressing mode is IND PCL ← M (ADH, ADL) PCH ← M (ADH, ADL + 1) If addressing mode is ZP, IND PCL ← M(00, ADL) PCH ← M(00, ADL + 1)	This instruction jumps to the address designated by the following three addressing modes: Absolute Indirect Absolute Zero Page Indirect Absolute																								4C	3	3										6C	5	3	B2	4	2										•	•	•	•	•	•	•	•
JSR	M(S) ← PCH S ← S – 1 M(S) ← PCL S ← S – 1 After executing the above, if addressing mode is ABS, PCL ← ADL PCH ← ADH if addressing mode is SP, PCL ← ADL PCH ← FF If addressing mode is ZP, IND, PCL ← M(00, ADL) PCH ← M(00, ADL + 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																								20	6	3													02	7	2							22	5	2	•	•	•	•	•	•	•	•
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				B5	4	2				AD	4	3	BD	5	3	B9	5	3							A1	6	2	B1	6	2							N	•	•	•	•	•	Z	•
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							B6	4	2	AE	4	3				BE	5	3																	N	•	•	•	•	•	Z	•
LDY	Y ← M	This instruction loads the contents of M in Y.				A0	2	2							A4	3	2				B4	4	2				AC	4	3	BC	5	3																					N	•	•	•	•	•	Z	•

APPENDIX
3.7 Machine instructions

Symbol	Function	Details	Addressing mode																										Processor status register							
			IMP			IMM			A			BIT, A			ZP			BIT, ZP			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	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
LSR	7 0 0→□→C	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				56	6	2				4E	6	3	5E	7	3																				0	•	•	•	•	•	Z	C
MUL	M(S) • A ← A ✻ M(zz + X) S ← S – 1	Multiplies 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.																			62	15	2																											•	•	•	•	•	•	•	•
NOP	PC ← PC + 1	This instruction adds one to the PC but does no otheroperation.	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				15	4	2				0D	4	3	1D	5	3	19	5	3							01	6	2	11	6	2							N	•	•	•	•	•	Z	•
PHA	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																																													N	•	•	•	•	•	Z	•
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																																													(Value saved in stack)							
ROL	7 0 ←□←C←	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				36	6	2				2E	6	3	3E	7	3																				N	•	•	•	•	•	Z	C
ROR	7 0 C→□→	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				76	6	2				6E	6	3	7E	7	3																				N	•	•	•	•	•	Z	C
RRF	7 0 →□→	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																																													(Value saved in stack)							
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																																													•	•	•	•	•	•	•	•

3.7 Machine instructions

Symbol	Function	Details	Addressing mode																						Processor status register							
			IMP			IMM			A			BIT, A			ZP			BIT, ZP			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	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
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				F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2							N	V	•	•	•	•	Z	C
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																																													•	•	•	•	•	•	•	1
SED	D ← 1	This instruction set D.	F8	2	1																																													•	•	•	•	1	•	•	•
SEI	I ← 1	This instruction set I.	78	2	1																																													•	•	•	•	•	1	•	•
SET	T ← 1	This instruction set T.	32	2	1																																													•	•	1	•	•	•	•	•
STA	M ← A	This instruction stores the contents of A in M. The contents of A does not change.													85	4	2				95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	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							96	5	2	8E	5	3																					•	•	•	•	•	•	•	•
STY	M ← Y	This instruction stores the contents of Y in M. The contents of Y does not change.													84	4	2				94	5	2				8C	5	3																					•	•	•	•	•	•	•	•
TAX	X ← A	This instruction stores the contents of A in X. The contents of A does not change.	AA	2	1																																												N	•	•	•	•	•	Z	•
TAY	Y ← A	This instruction stores the contents of A in Y. The contents of A does not change.	A8	2	1																																												N	•	•	•	•	•	Z	•
TST	M = 0?	This instruction tests whether the contents of M are “0” or not and modifies the N and Z.													64	3	2																																	N	•	•	•	•	•	Z	•
TSX	X ← S	This instruction transfers the contents of S in X.	BA	2	1																																												N	•	•	•	•	•	Z	•
TXA	A ← X	This instruction stores the contents of X in A.	8A	2	1																																												N	•	•	•	•	•	Z	•
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																																												N	•	•	•	•	•	Z	•
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.

APPENDIX

3.7 Machine instructions

APPENDIX

3.7 Machine instructions

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

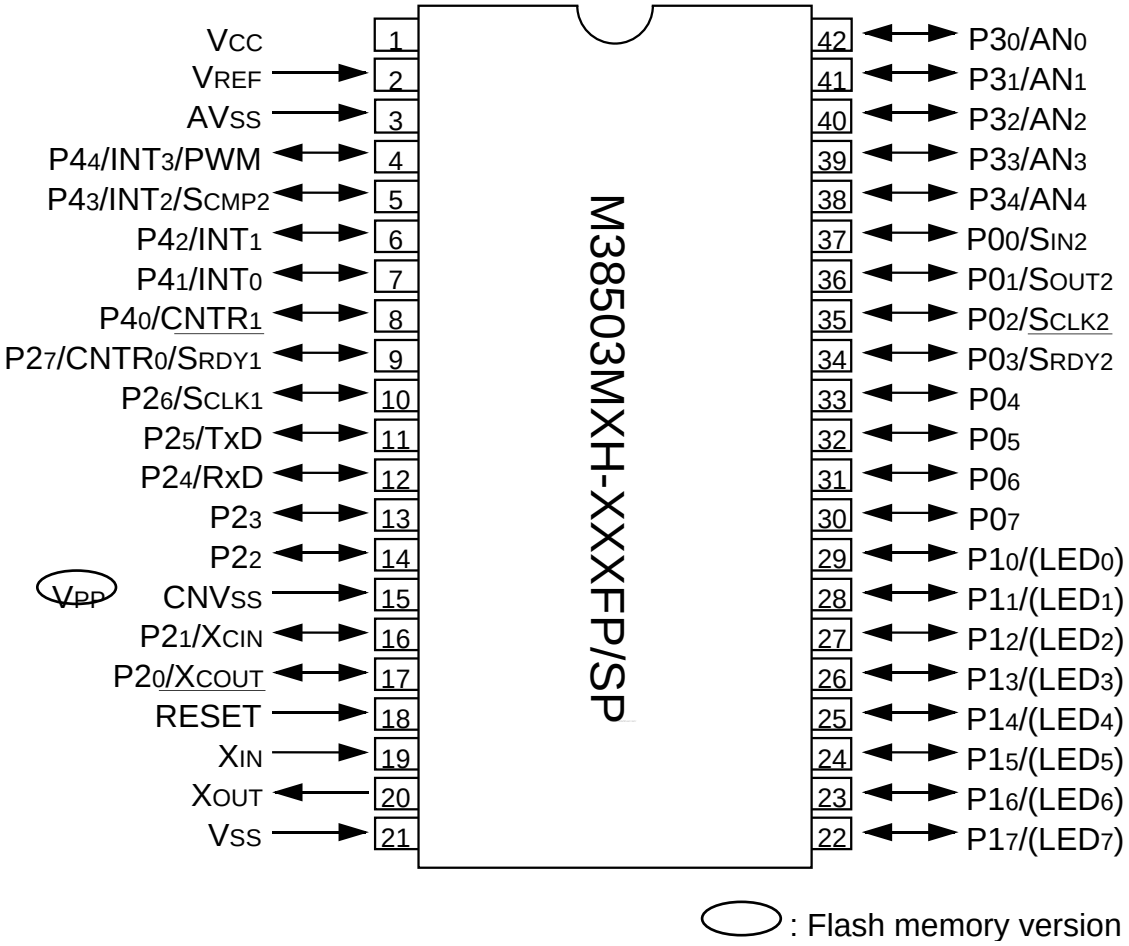
0000_{16}	Port P0 (P0)
0001_{16}	Port P0 direction register (P0D)
0002_{16}	Port P1 (P1)
0003_{16}	Port P1 direction register (P1D)
0004_{16}	Port P2 (P2)
0005_{16}	Port P2 direction register (P2D)
0006_{16}	Port P3 (P3)
0007_{16}	Port P3 direction register (P3D)
0008_{16}	Port P4 (P4)
0009_{16}	Port P4 direction register (P4D)
$000A_{16}$	
$000B_{16}$	
$000C_{16}$	
$000D_{16}$	
$000E_{16}$	
$000F_{16}$	
0010_{16}	
0011_{16}	
0012_{16}	Reserved ✻
0013_{16}	Reserved ✻
0014_{16}	Reserved ✻
0015_{16}	Serial I/O2 control register 1 (SIO2CON1)
0016_{16}	Serial I/O2 control register 2 (SIO2CON2)
0017_{16}	Serial I/O2 register (SIO2)
0018_{16}	Transmit/Receive buffer register (TB/RB)
0019_{16}	Serial I/O1 status register (SIOSTS)
$001A_{16}$	Serial I/O1 control register (SIOCON)
$001B_{16}$	UART control register (UARTCON)
$001C_{16}$	Baud rate generator (BRG)
$001D_{16}$	PWM control register (PWMCON)
$001E_{16}$	PWM prescaler (PREPWM)
$001F_{16}$	PWM register (PWM)

0020_{16}	Prescaler 12 (PRE12)
0021_{16}	Timer 1 (T1)
0022_{16}	Timer 2 (T2)
0023_{16}	Timer XY mode register (TM)
0024_{16}	Prescaler X (PREX)
0025_{16}	Timer X (TX)
0026_{16}	Prescaler Y (PREY)
0027_{16}	Timer Y (TY)
0028_{16}	Timer count source selection register (TCSS)
0029_{16}	
$002A_{16}$	
$002B_{16}$	Reserved ✻
$002C_{16}$	Reserved ✻
$002D_{16}$	Reserved ✻
$002E_{16}$	Reserved ✻
$002F_{16}$	Reserved ✻
0030_{16}	Reserved ✻
0031_{16}	Reserved ✻
0032_{16}	
0033_{16}	
0034_{16}	A-D control register (ADCON)
0035_{16}	A-D conversion low-order register (ADL)
0036_{16}	A-D conversion high-order register (ADH)
0037_{16}	Reserved ✻
0038_{16}	MISRG
0039_{16}	Watchdog timer control register (WDTCON)
$003A_{16}$	Interrupt edge selection register (INTEDGE)
$003B_{16}$	CPU mode register (CPUM)
$003C_{16}$	Interrupt request register 1 (IREQ1)
$003D_{16}$	Interrupt request register 2 (IREQ2)
$003E_{16}$	Interrupt control register 1 (ICON1)
$003F_{16}$	Interrupt control register 2 (ICON2)

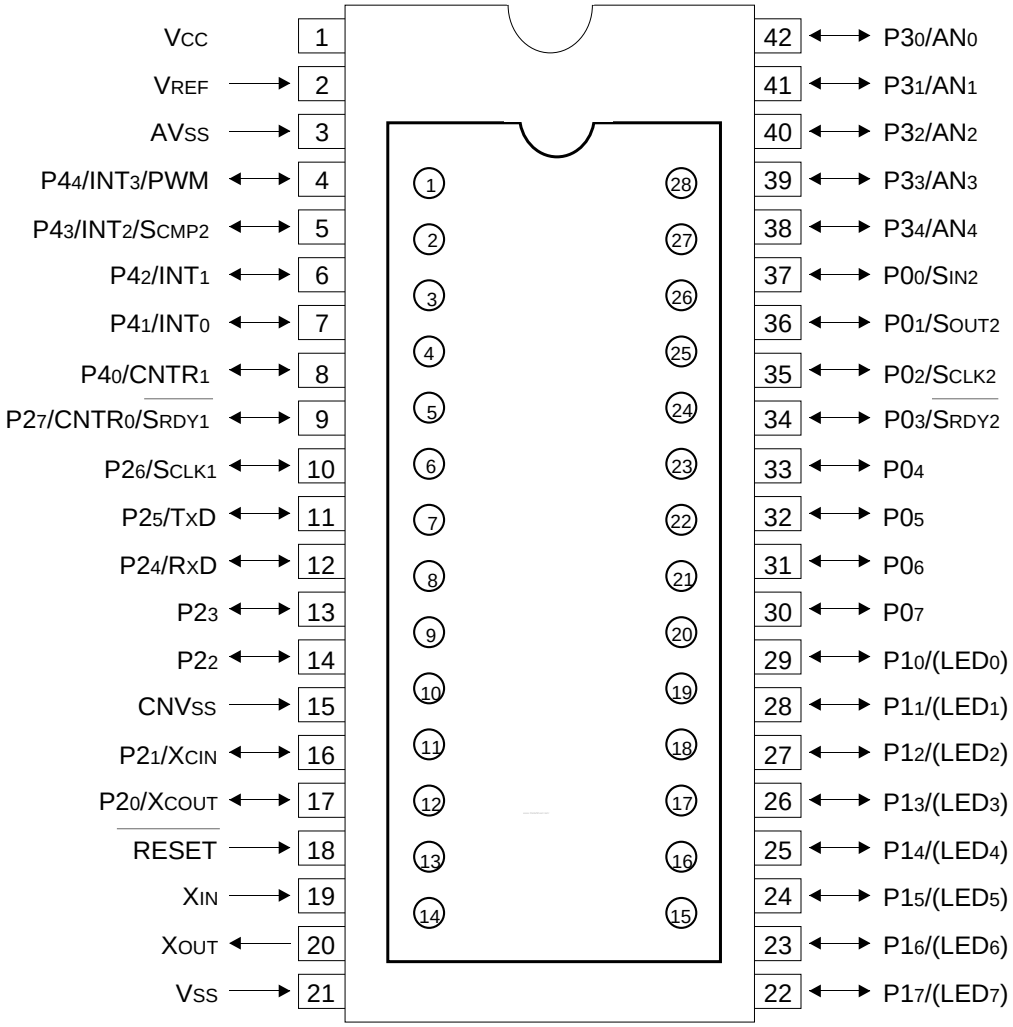
$0FFE_{16}$	Flash memory control register (FMCR)

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

3.10 Pin configurations



M38517RSS PIN CONFIGURATION (TOP VIEW)

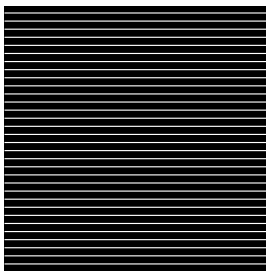


Outline : 42S1M

RENESAS 8-BIT CISC SINGLE-CHIP MICROCOMPUTER
USER'S MANUAL
3850 Group (Spec. H) Rev.1.03

Editioned by
Committee of editing of RENESAS Semiconductor User's Manual

3850 Group (Spec. H)
User's Manual



RENESAS

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan