To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

3850 Group (Spec.A)

Peripheral Function Application

1. Introduction

The following article introduces and shows an application example of peripheral function.

The explanation of this issue is applied to the following condition:
Applicable MCU: 3850 Group (Spec.A)

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

Address	
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)
0012_{16}	Port P0, P1, P2 pull-up control register (PULL012)
0013_{16}	Port P3 pull-up control register (PULL3)
0014_{16}	Port P4 pull-up control register (PULL4)

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

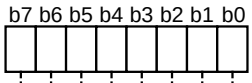
Port Pi (i = 0, 1, 2, 3, 4)
(Pi: addresses 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{16})

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

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

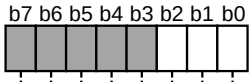


Port Pi direction register (i = 0, 1, 2, 3, 4)
(PiD: addresses 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{16})

b	Name	Functions	At reset	R	W
0	Port Pi direction register	0 : Port Pi0 input mode 1 : Port Pi0 output mode	0	×	○
1		0 : Port Pi1 input mode 1 : Port Pi1 output mode	0	×	○
2		0 : Port Pi2 input mode 1 : Port Pi2 output mode	0	×	○
3		0 : Port Pi3 input mode 1 : Port Pi3 output mode	0	×	○
4		0 : Port Pi4 input mode 1 : Port Pi4 output mode	0	×	○
5		0 : Port Pi5 input mode 1 : Port Pi5 output mode	0	×	○
6		0 : Port Pi6 input mode 1 : Port Pi6 output mode	0	×	○
7		0 : Port Pi7 input mode 1 : Port Pi7 output mode	0	×	○

Fig. 2.1.3 Structure of Port Pi direction register (i = 0, 1, 2, 3, 4)

Port P0, P1, P2 pull-up control register



Port P0, P1, P2, pull-up control register
(PULL012 : address 12_{16})

b	Name	Functions	At reset	R	W
0	P0 pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
1	P1 pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
2	P2 pull-up control bit	0 : No pull-up 1 : Pull-up	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. 2.1.4 Structure of Port P0, P1, P2 pull-up control register

Port P3 pull-up control register

b7 b6 b5 b4 b3 b2 b1 b0

Port P3 pull-up control register
(PULL3 : address 13_{16})

b	Name	Functions	At reset	R	W
0	$P3_0$ pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
1	$P3_1$ pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
2	$P3_2$ pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
3	$P3_3$ pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
4	$P3_4$ pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
5	Fix these bits to "0".		0	○	○
6			0	○	○
7			0	○	○

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

Port P4 pull-up control register

b7 b6 b5 b4 b3 b2 b1 b0

Port P4 pull-up control register
(PULL4 : address 14_{16})

b	Name	Functions	At reset	R	W
0	$P4_0$ pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
1	$P4_1$ pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
2	$P4_2$ pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
3	$P4_3$ pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
4	$P4_4$ pull-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
5	Fix these bits to "0".		0	○	○
6			0	○	○
7			0	○	○

Fig. 2.1.6 Structure of Port P4 pull-up control register

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. In the port which can select a internal pull-up resistor, the internal pull-up resistor can be used. • 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)

2.1.4 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 I/O ports :

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

② The AVss pin when not using the A-D converter :

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

(2) Termination remarks

① 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 ① shown on the above.

② I/O ports :
When setting for the input mode, do not connect to Vcc or Vss directly.

● **Reason**

If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between a port and Vcc (or Vss).

③ I/O ports :
When setting for the input mode, do not connect multiple ports in a lump to Vcc or Vss through a resistor.

● **Reason**

If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between ports.

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

2.2 Interrupt

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

2.2.1 Memory map

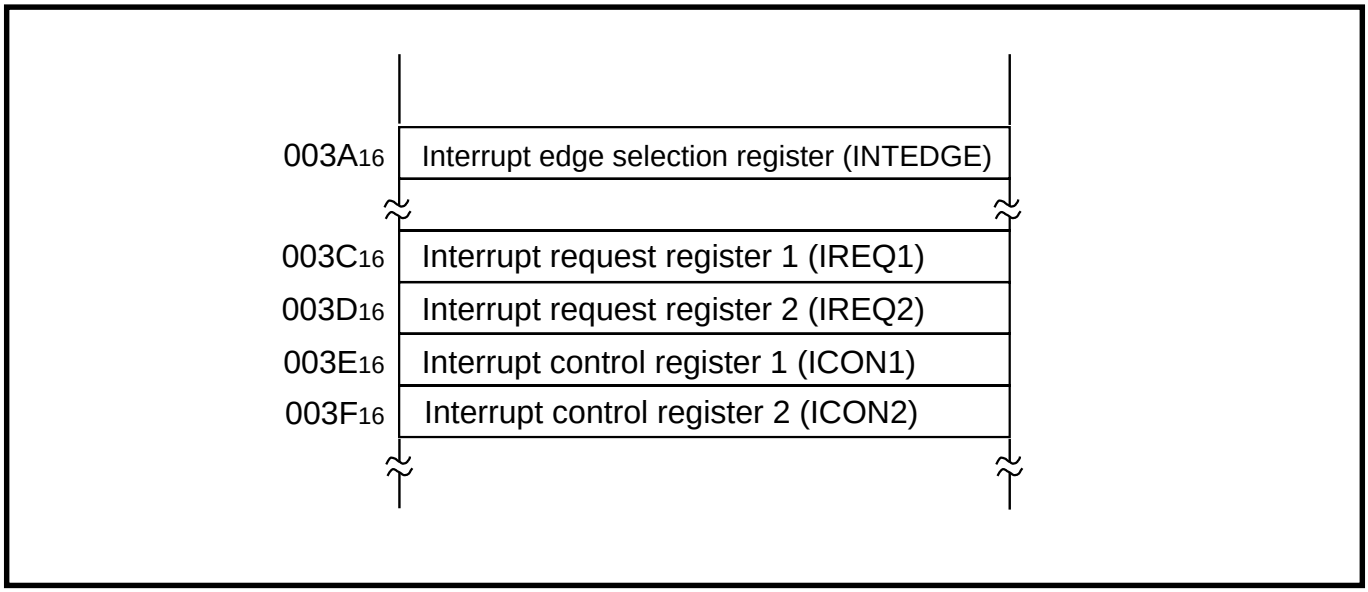


Fig. 2.2.1 Memory map of registers relevant to interrupt

2.2.2 Relevant registers

Interrupt edge selection register

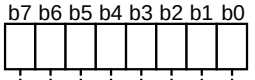
b7 b6 b5 b4 b3 b2 b1 b0

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 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

Interrupt request register 1



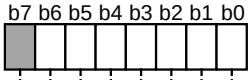
Interrupt request register 1
(IREQ1 : address $3C_{16}$)

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. 2.2.3 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. 2.2.4 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

2.2.3 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.

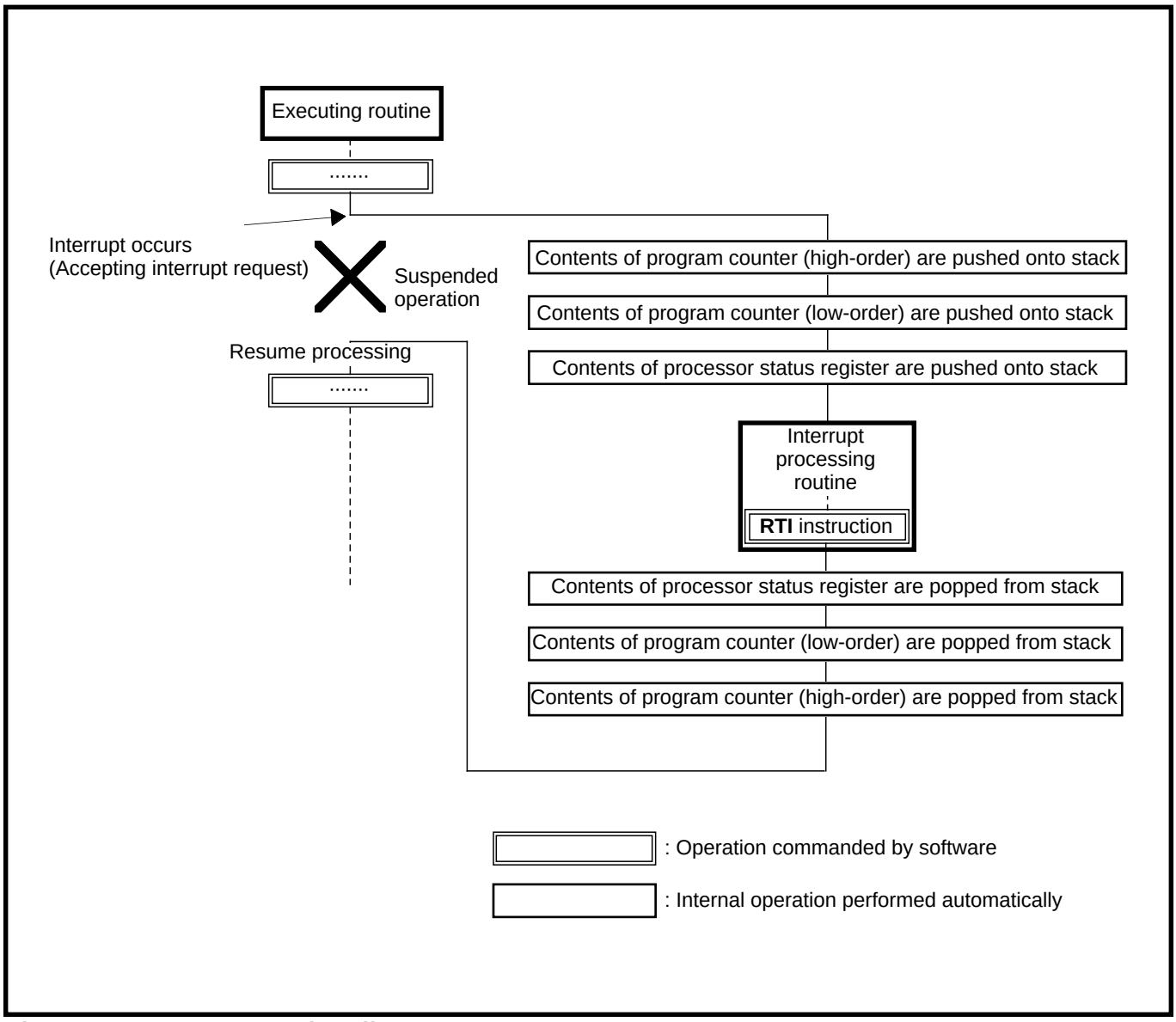


Fig. 2.2.7 Interrupt operation diagram

(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.

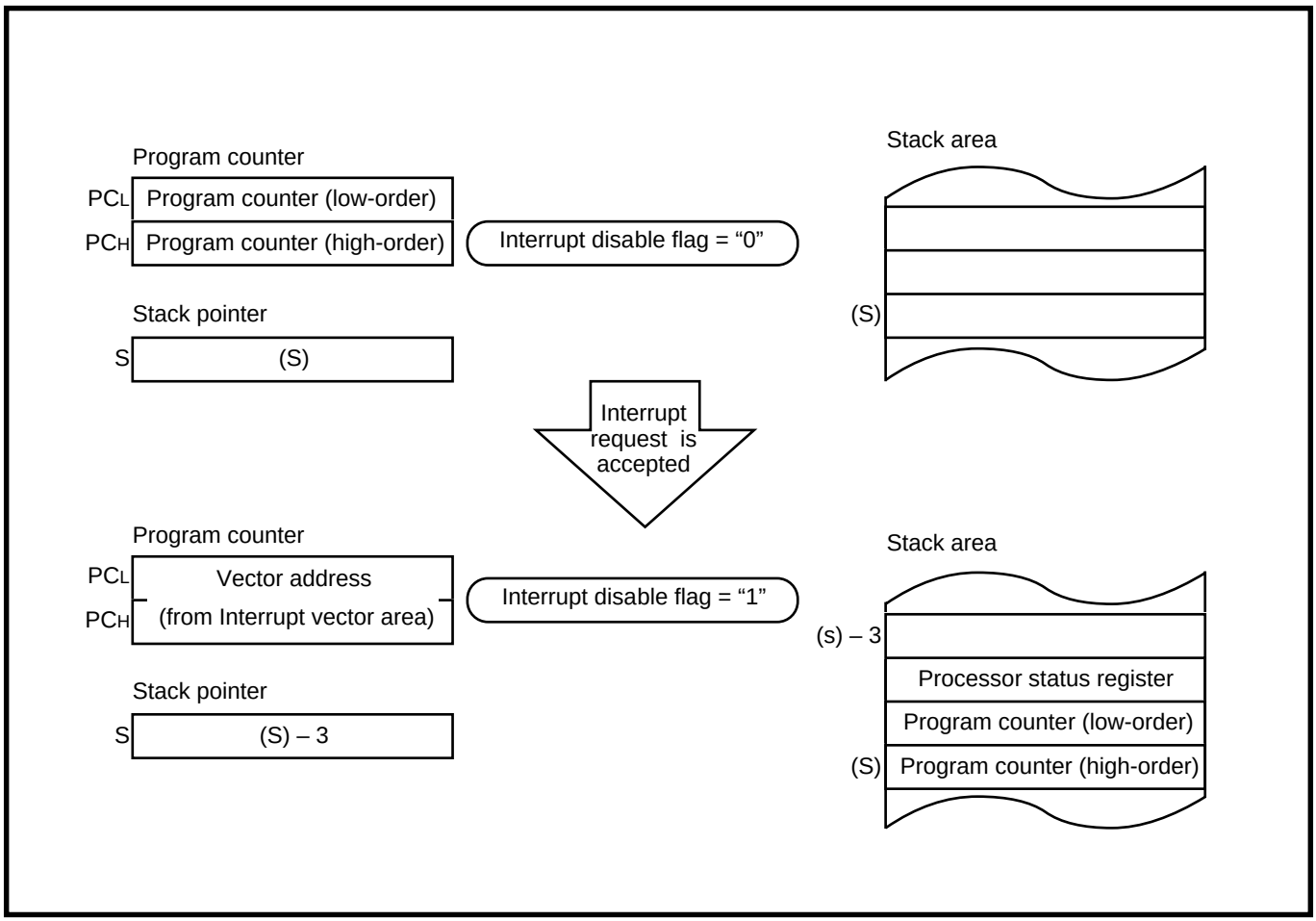


Fig. 2.2.8 Changes of stack pointer and program counter upon acceptance of interrupt request

(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.

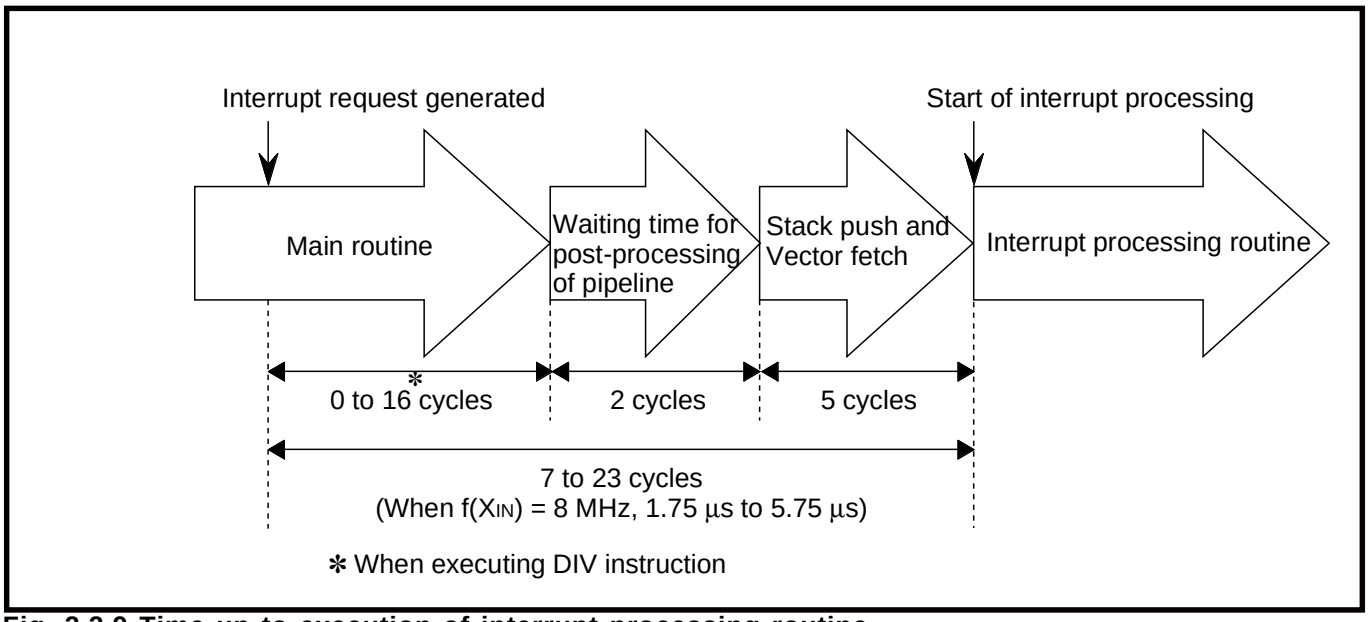


Fig. 2.2.9 Time up to execution of interrupt processing routine

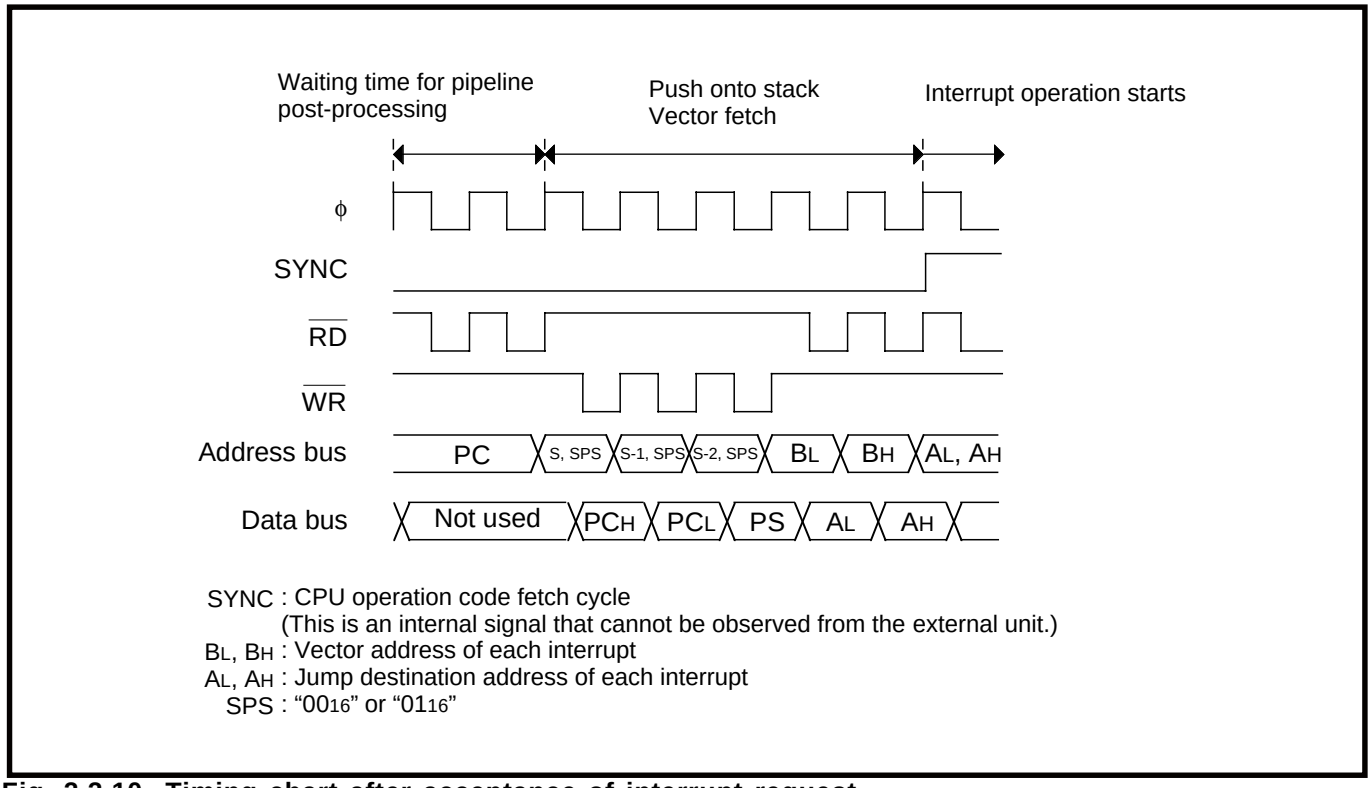


Fig. 2.2.10 Timing chart after acceptance of interrupt request

2.2.5 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.

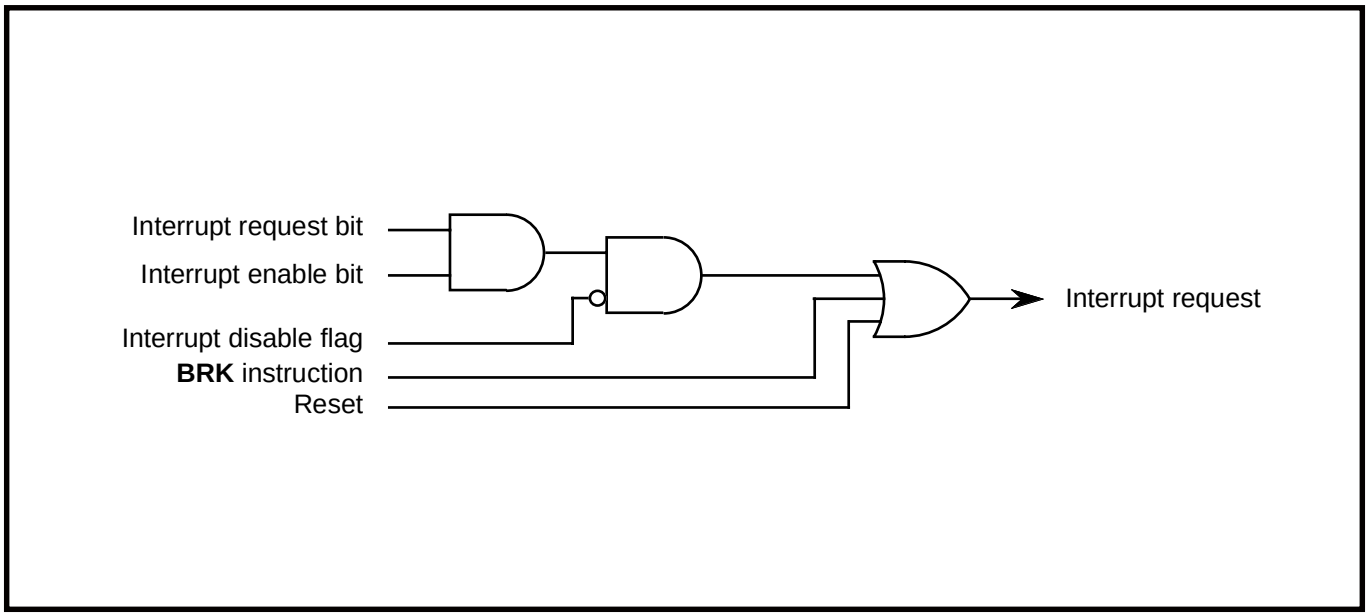


Fig. 2.2.11 Interrupt control diagram

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

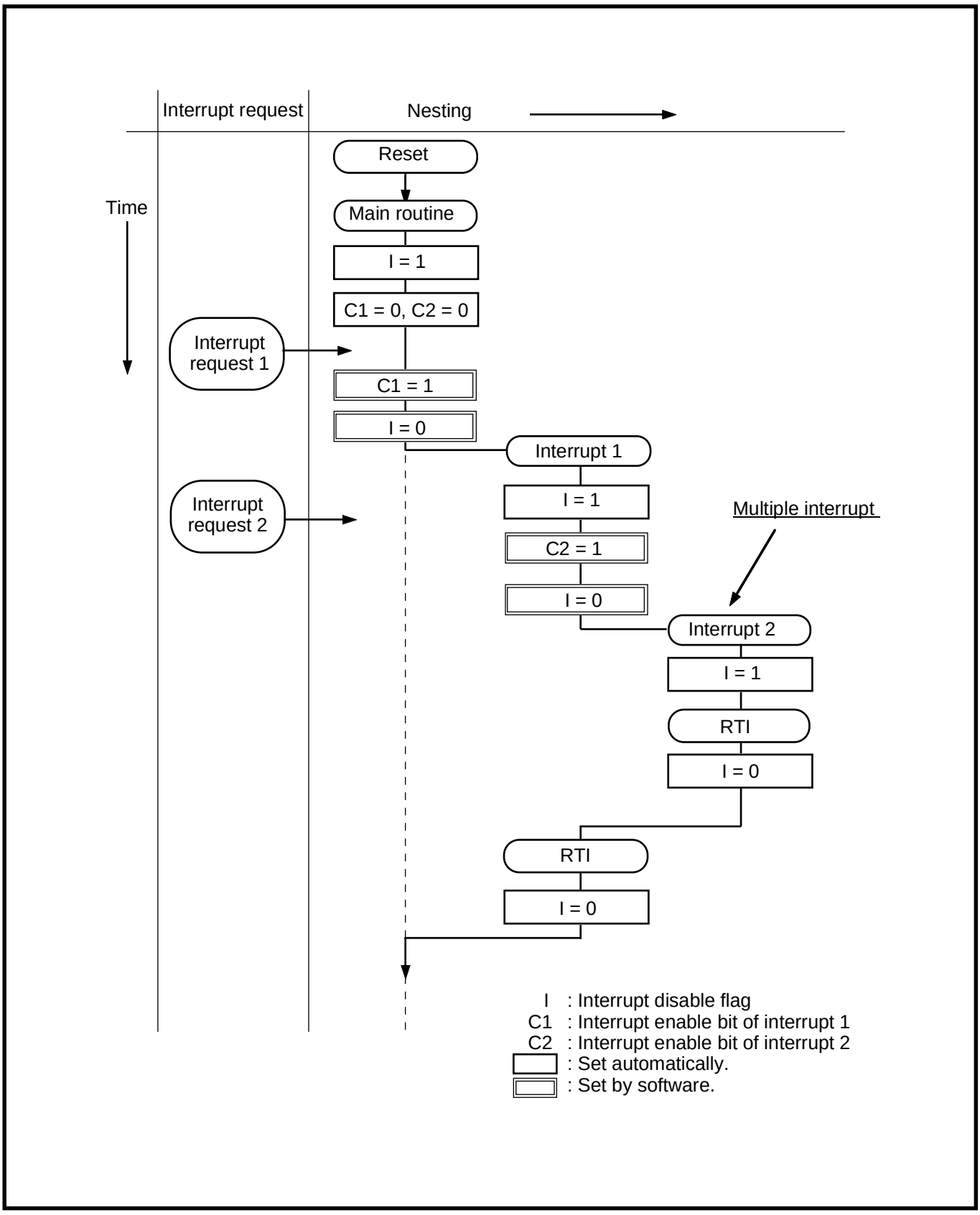


Fig. 2.2.12 Example of multiple interrupts

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.

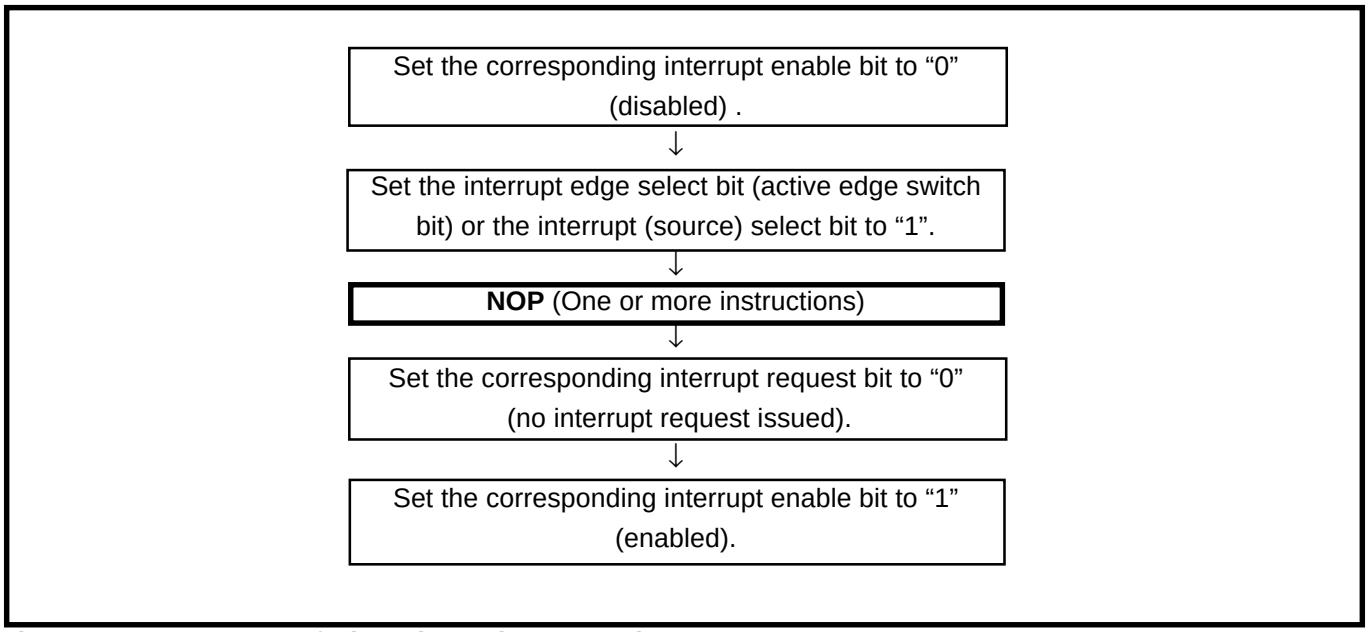


Fig. 2.2.13 Sequence of changing relevant register

■ **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.

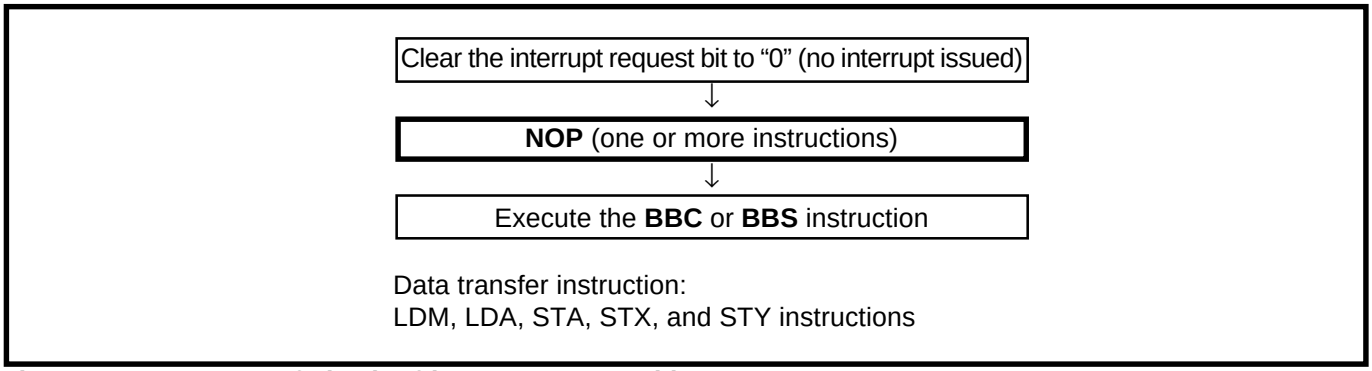


Fig. 2.2.14 Sequence of check of interrupt request bit

■ **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

Address	
0020₁₆	Prescaler 12 (PRE12)
0021₁₆	Timer 1 (T1)
0022₁₆	Timer 2 (T2)
0023₁₆	Timer XY mode register (TM)
0024₁₆	Prescaler X (PREX)
0025₁₆	Timer X (TX)
0026₁₆	Prescaler Y (PREY)
0027₁₆	Timer Y (TY)
0028₁₆	Timer count source selection register (TCSS)
≈	≈
003C₁₆	Interrupt request register 1 (IREQ1)
003D₁₆	Interrupt request register 2 (IREQ2)
003E₁₆	Interrupt control register 1 (ICON1)
003F₁₆	Interrupt control register 2 (ICON2)

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 20₁₆, 24₁₆, 26₁₆)

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

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

Timer 1

b7 b6 b5 b4 b3 b2 b1 b0

Timer 1
(T1: address 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

b7 b6 b5 b4 b3 b2 b1 b0

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

Timer X, Timer Y

b7 b6 b5 b4 b3 b2 b1 b0

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

b7 b6 b5 b4 b3 b2 b1 b0

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

b7 b6 b5 b4 b3 b2 b1 b0

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

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. 2.3.8 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. 2.3.9 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.

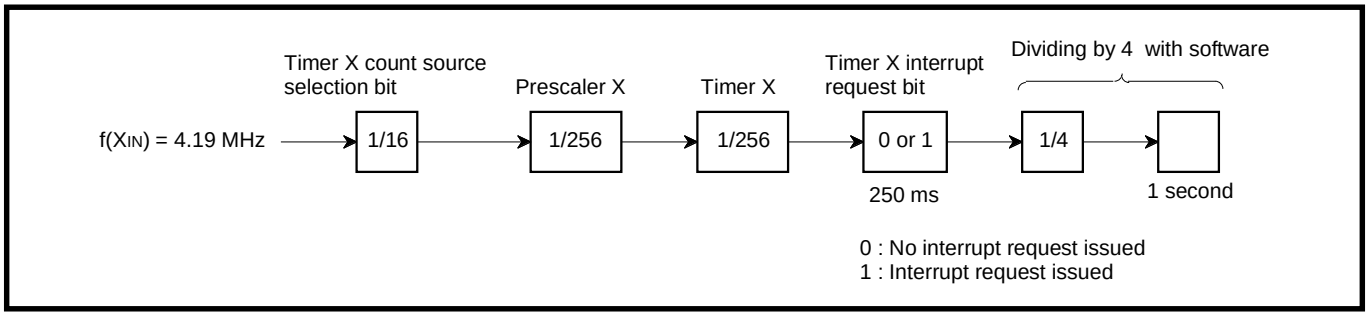


Fig. 2.3.12 Timers connection and setting of division ratios

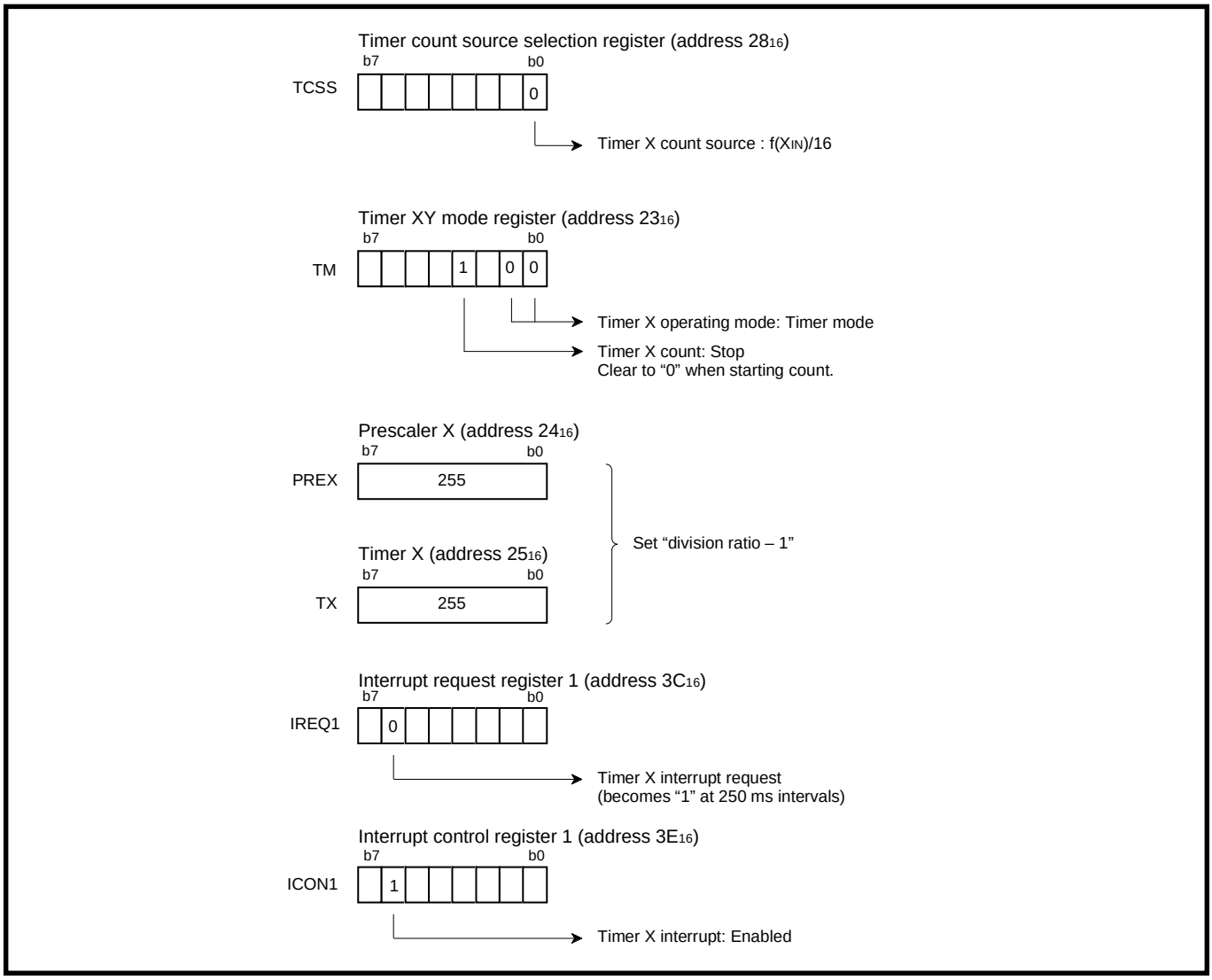


Fig. 2.3.13 Relevant registers setting

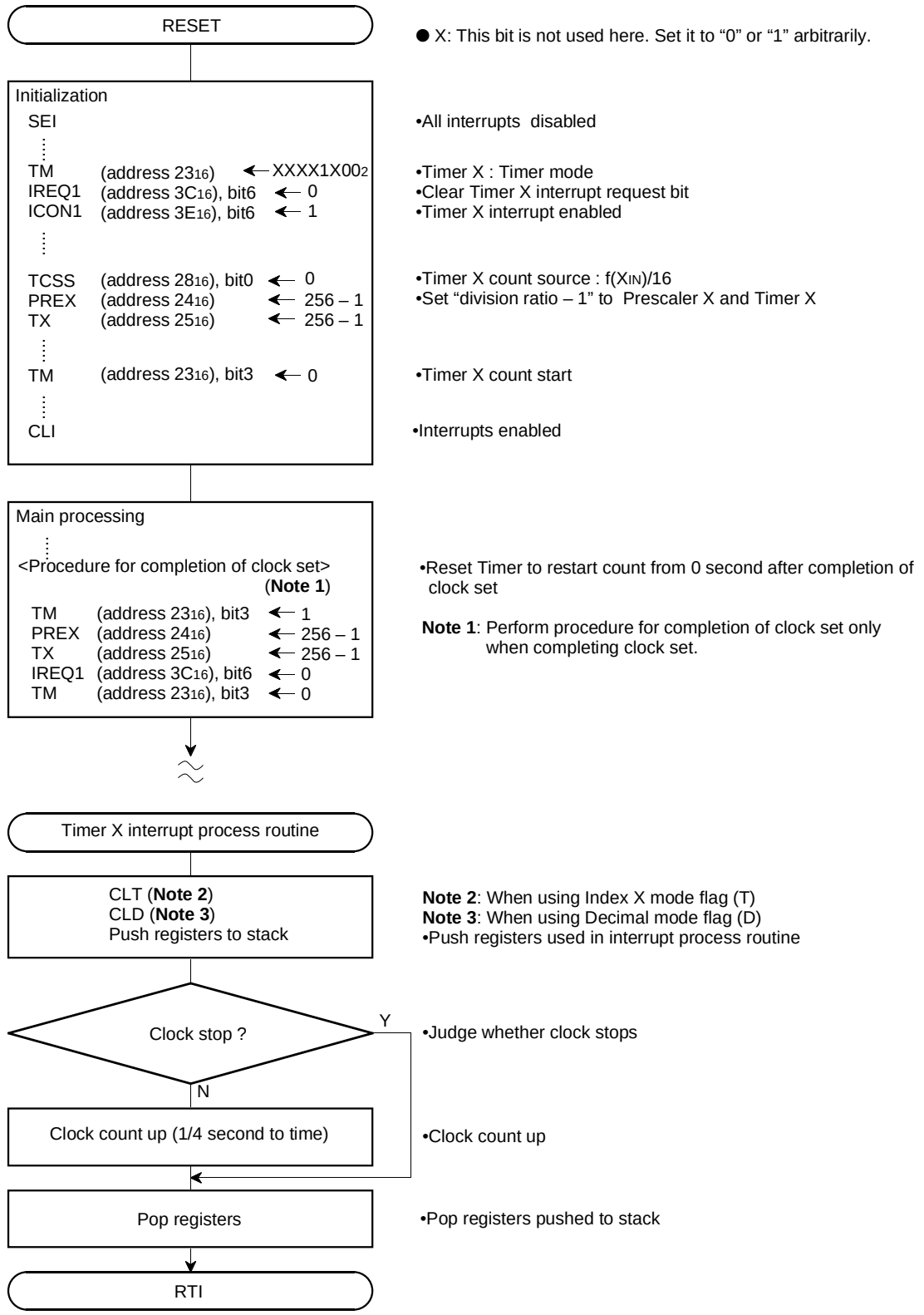


Fig. 2.3.14 Control procedure

(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.

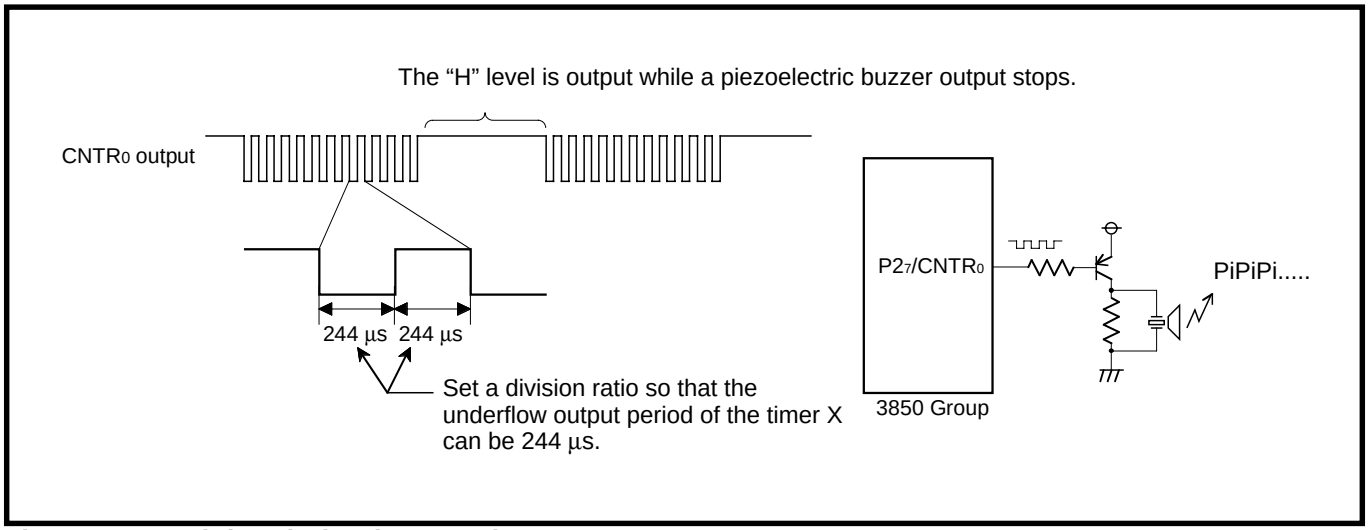


Fig. 2.3.15 Peripheral circuit example

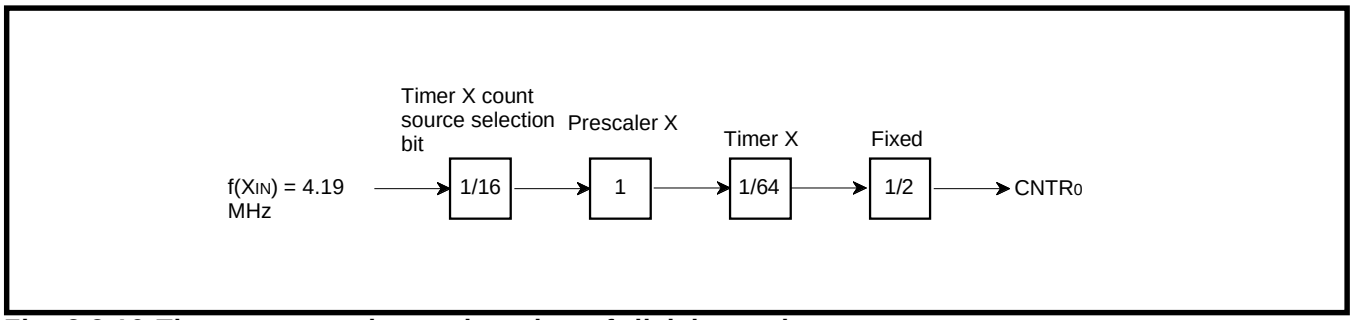


Fig. 2.3.16 Timers connection and setting of division ratios

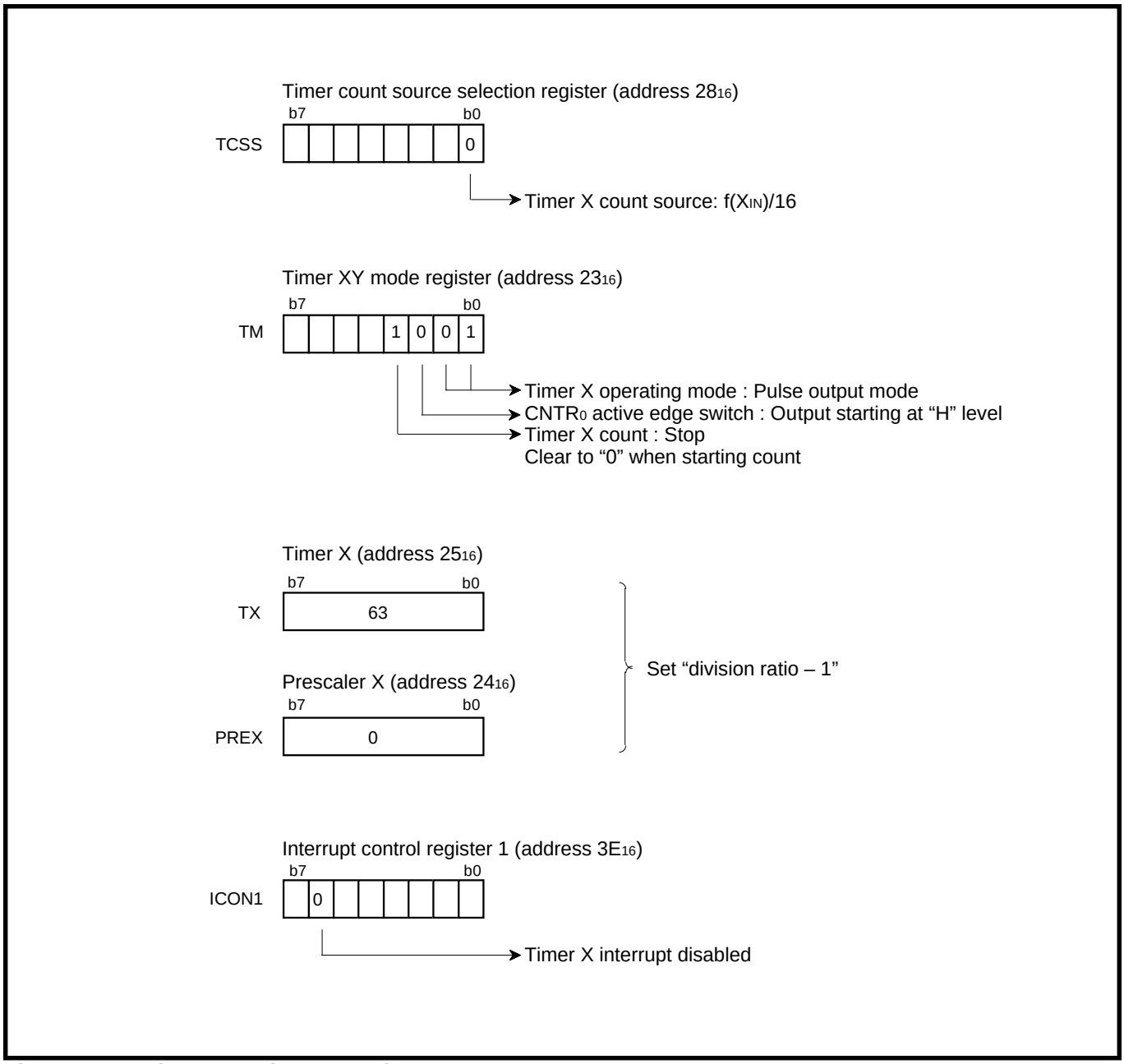


Fig. 2.3.17 Relevant registers setting

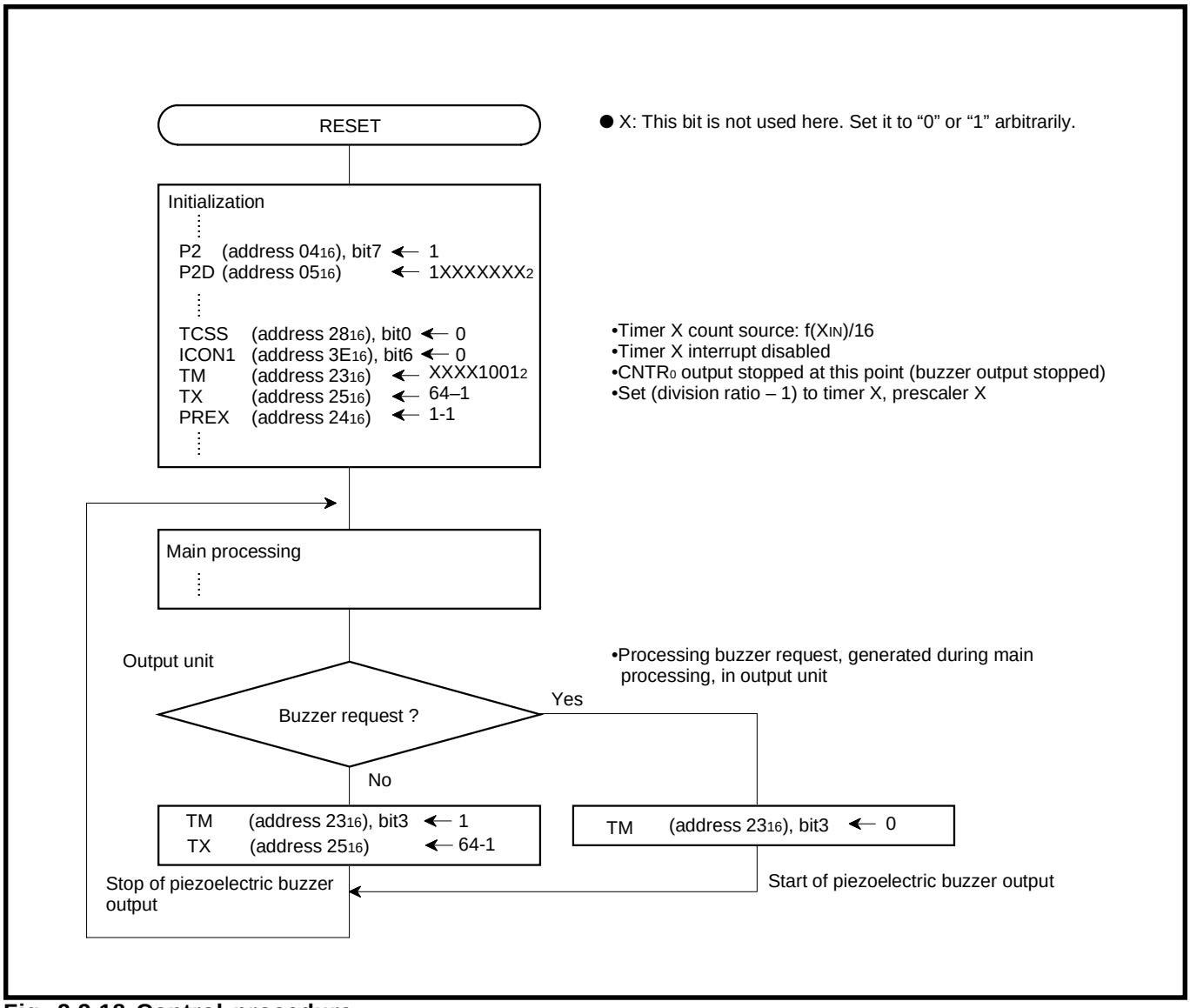


Fig. 2.3.18 Control procedure

(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.

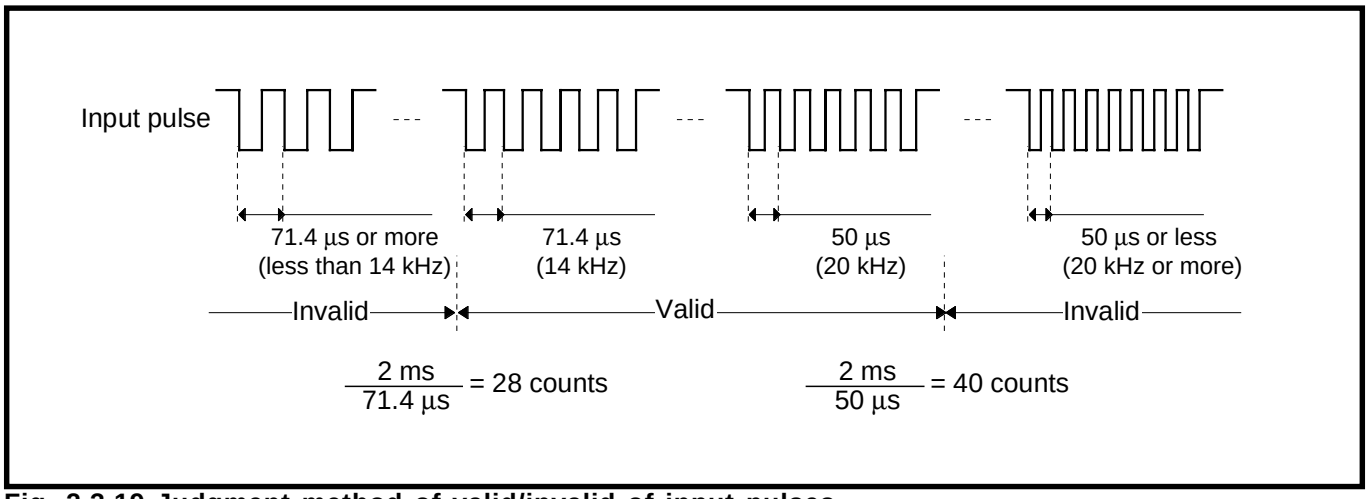


Fig. 2.3.19 Judgment method of valid/invalid of input pulses

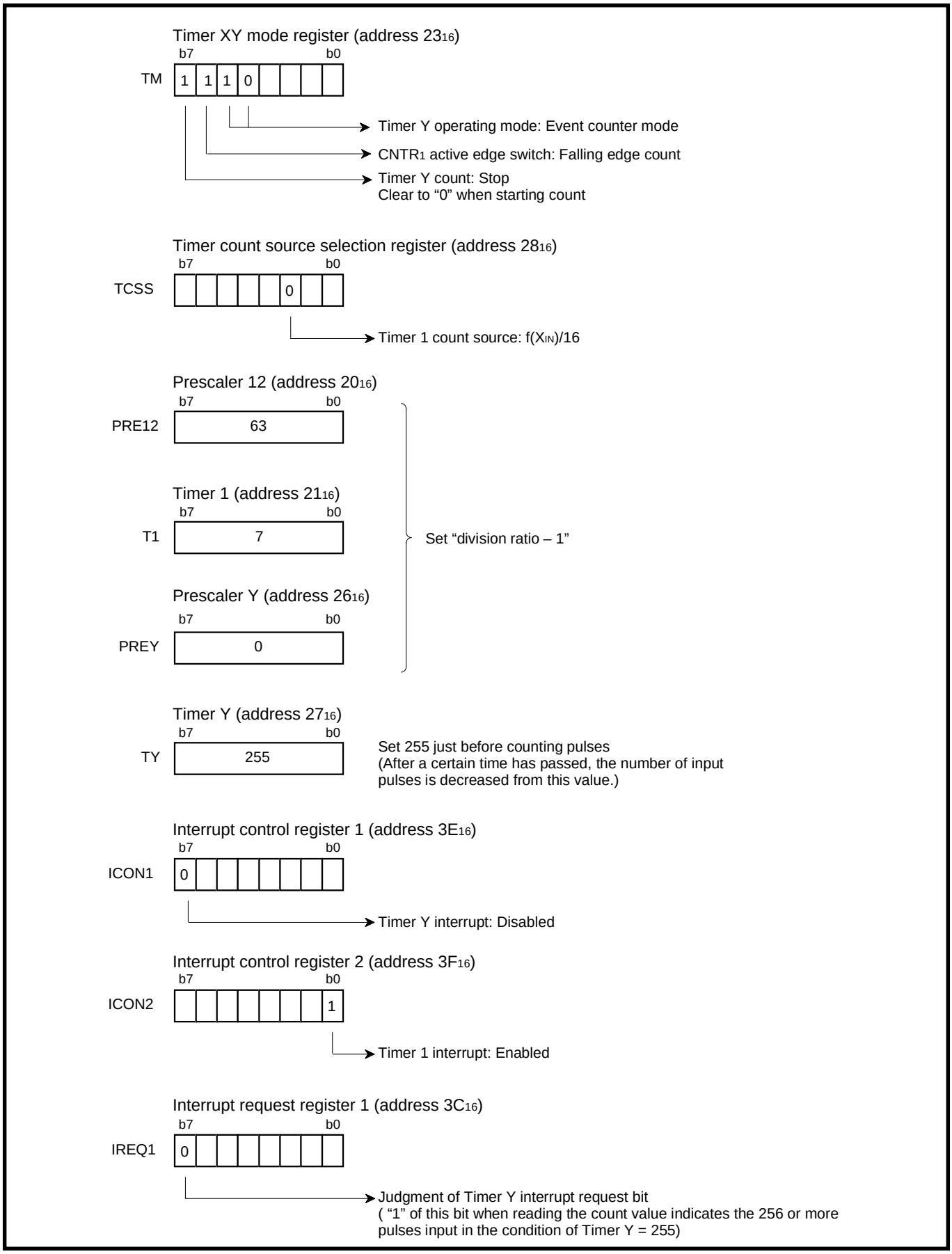


Fig. 2.3.20 Relevant registers setting

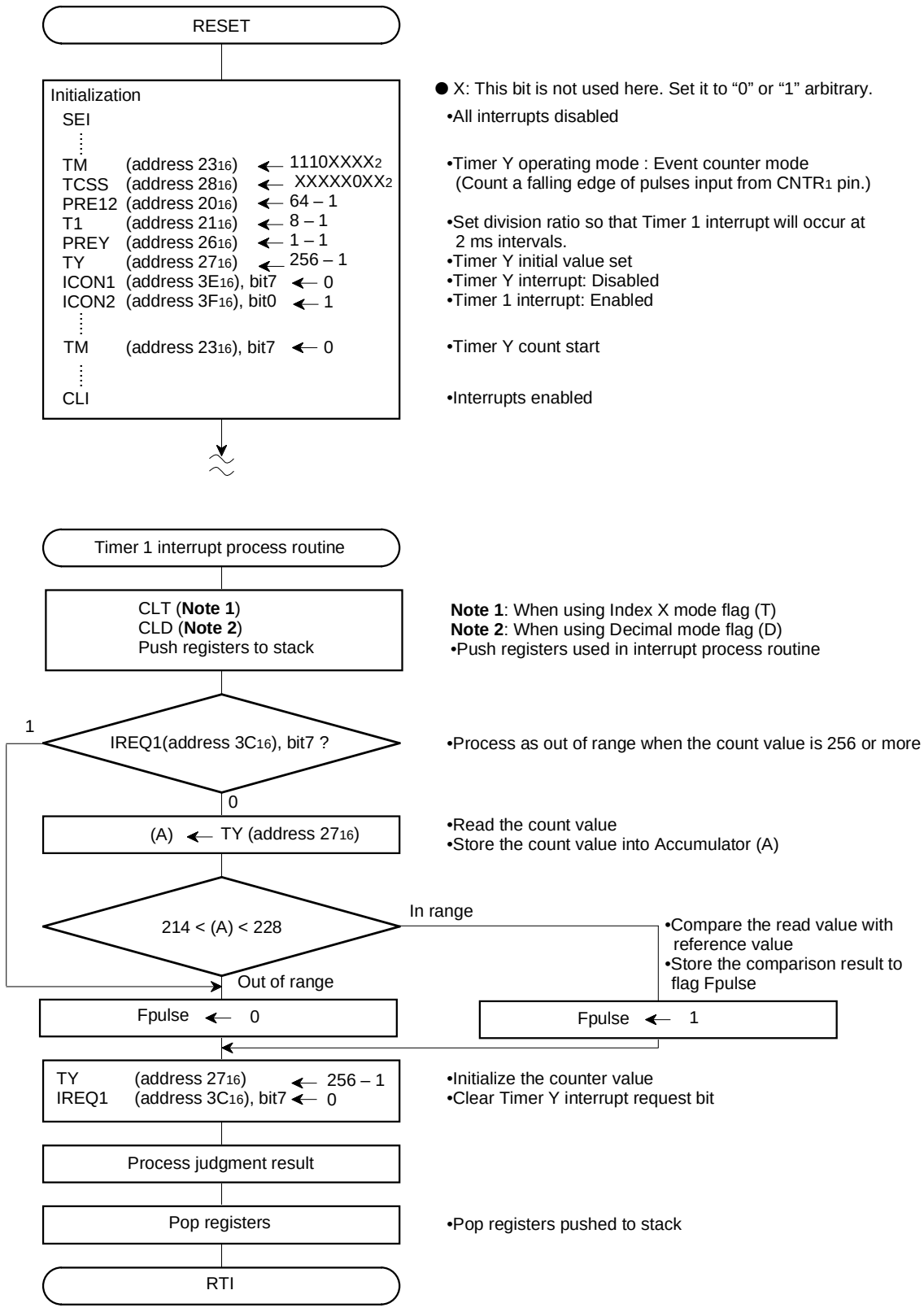


Fig. 2.3.21 Control procedure

(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.

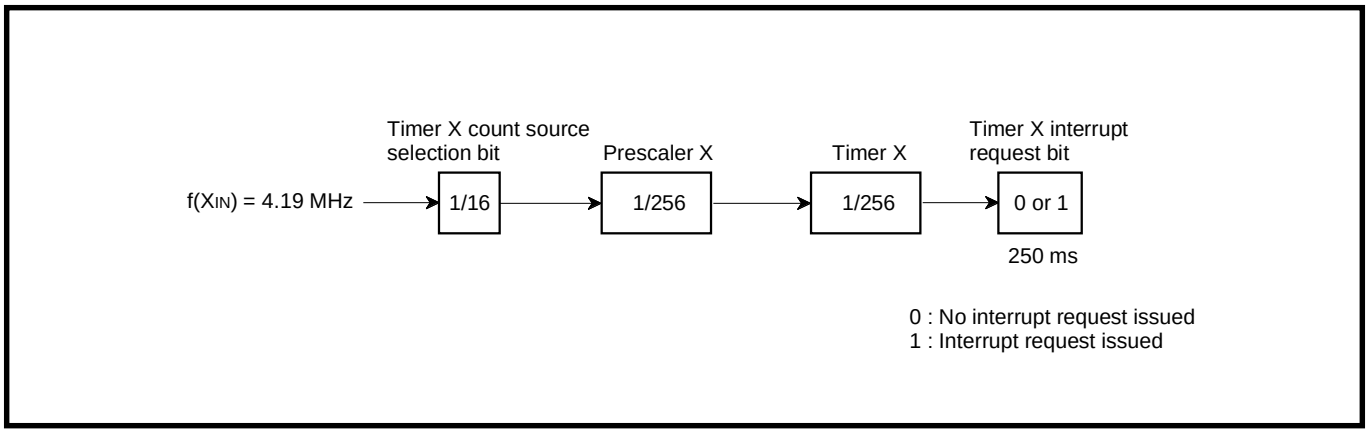


Fig. 2.3.22 Timers connection and setting of division ratios

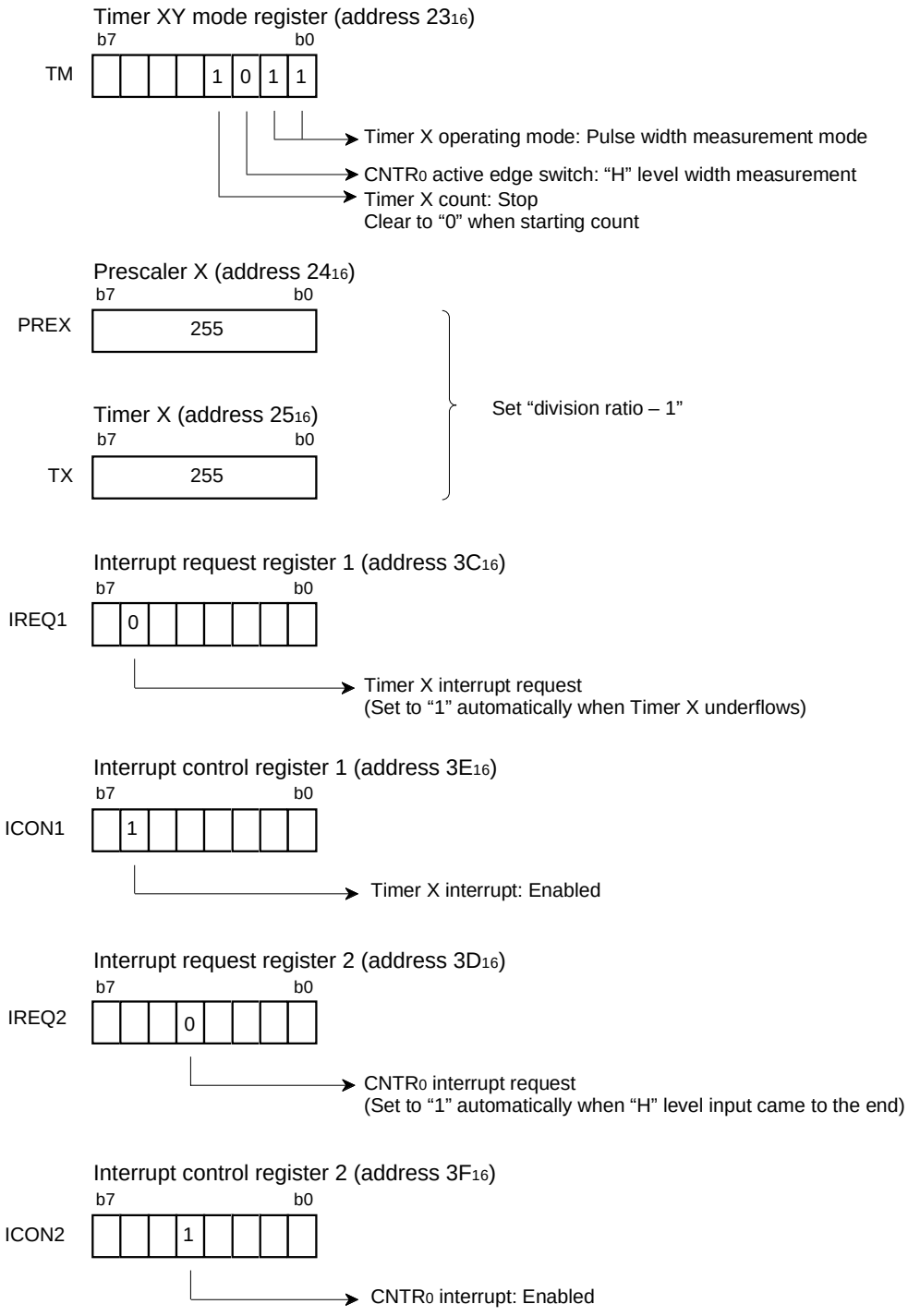


Fig. 2.3.23 Relevant registers setting

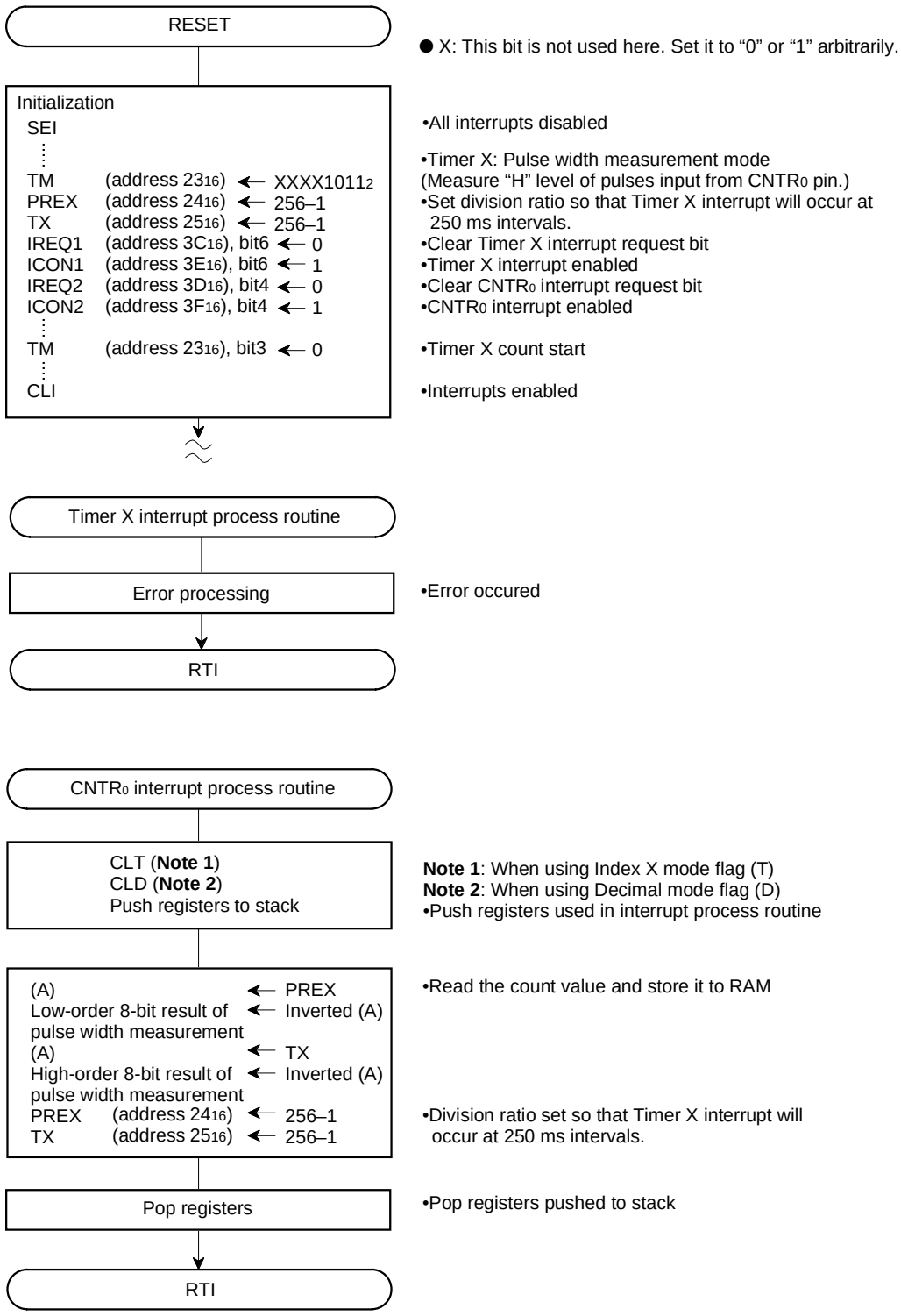


Fig. 2.3.24 Control procedure

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

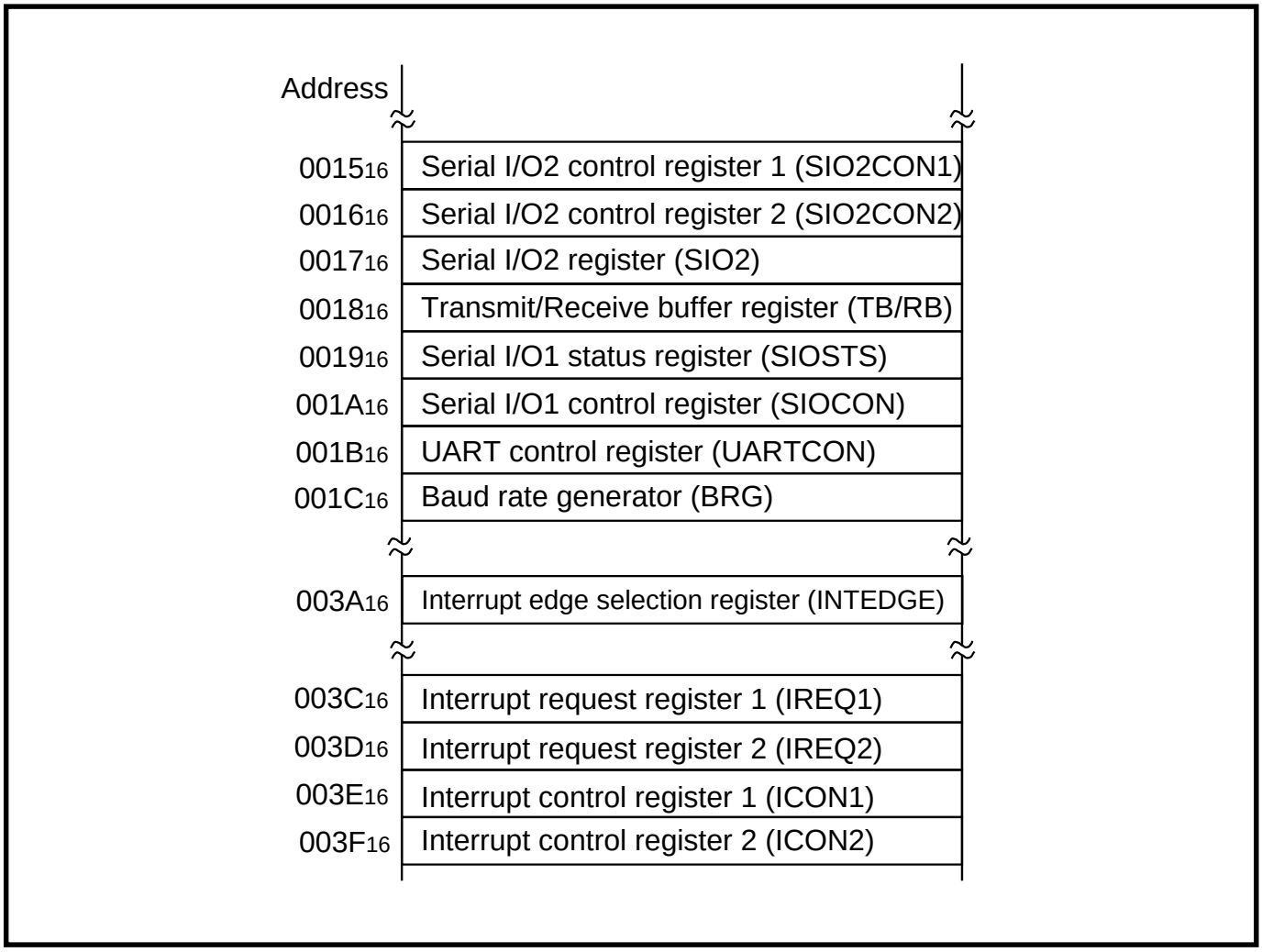


Fig. 2.4.1 Memory map of registers relevant to Serial I/O

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

b7 b6 b5 b4 b3 b2 b1 b0

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	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. 2.4.4 Structure of Serial I/O2 register

Transmit/Receive buffer register

b7 b6 b5 b4 b3 b2 b1 b0

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	O	O
1		Undefined	O	O
2		Undefined	O	O
3		Undefined	O	O
4		Undefined	O	O
5		Undefined	O	O
6		Undefined	O	O
7		Undefined	O	O

Note: 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

b7 b6 b5 b4 b3 b2 b1 b0

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

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. 2.4.13 Structure of Interrupt control register 1

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0
0

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

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.

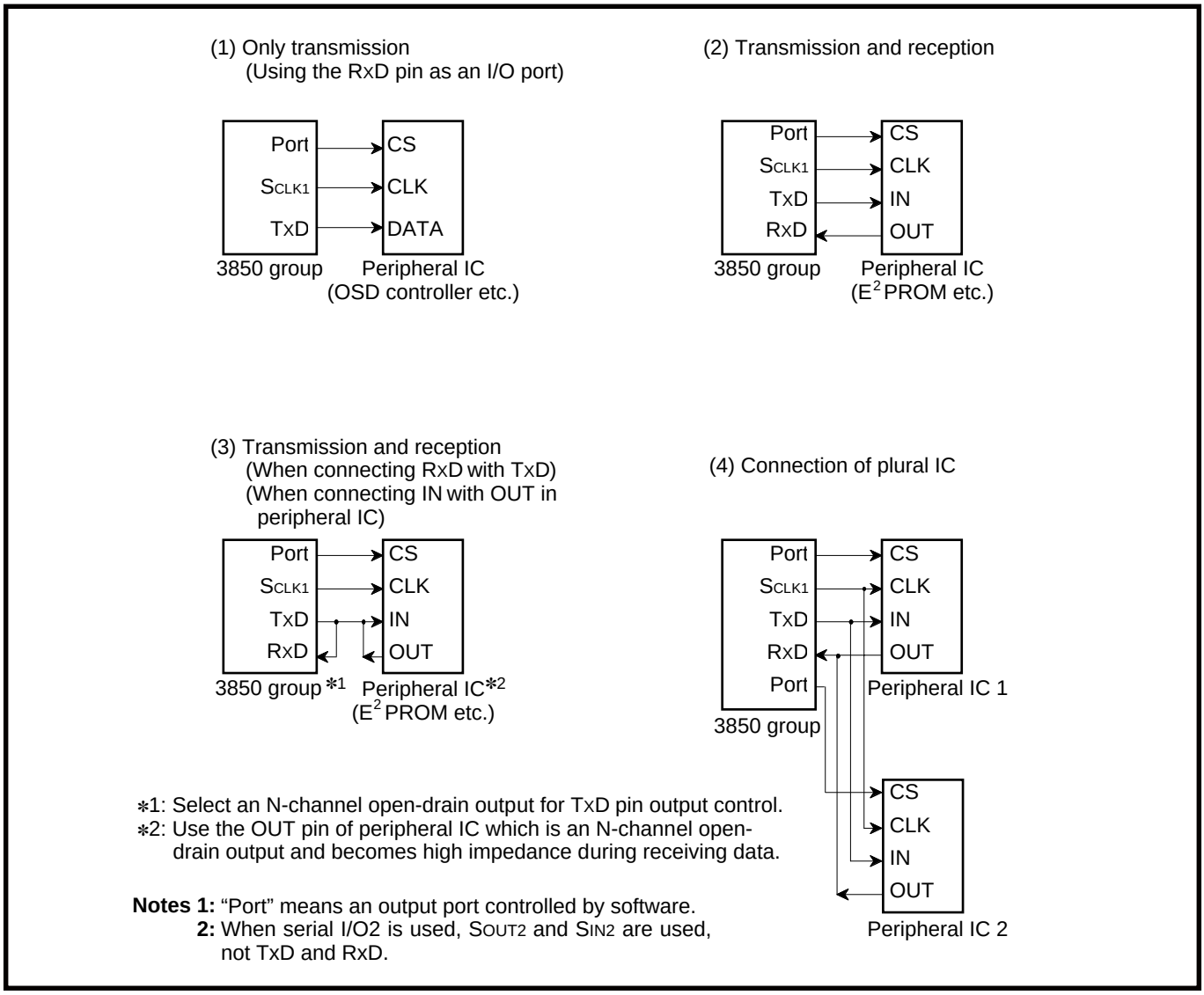
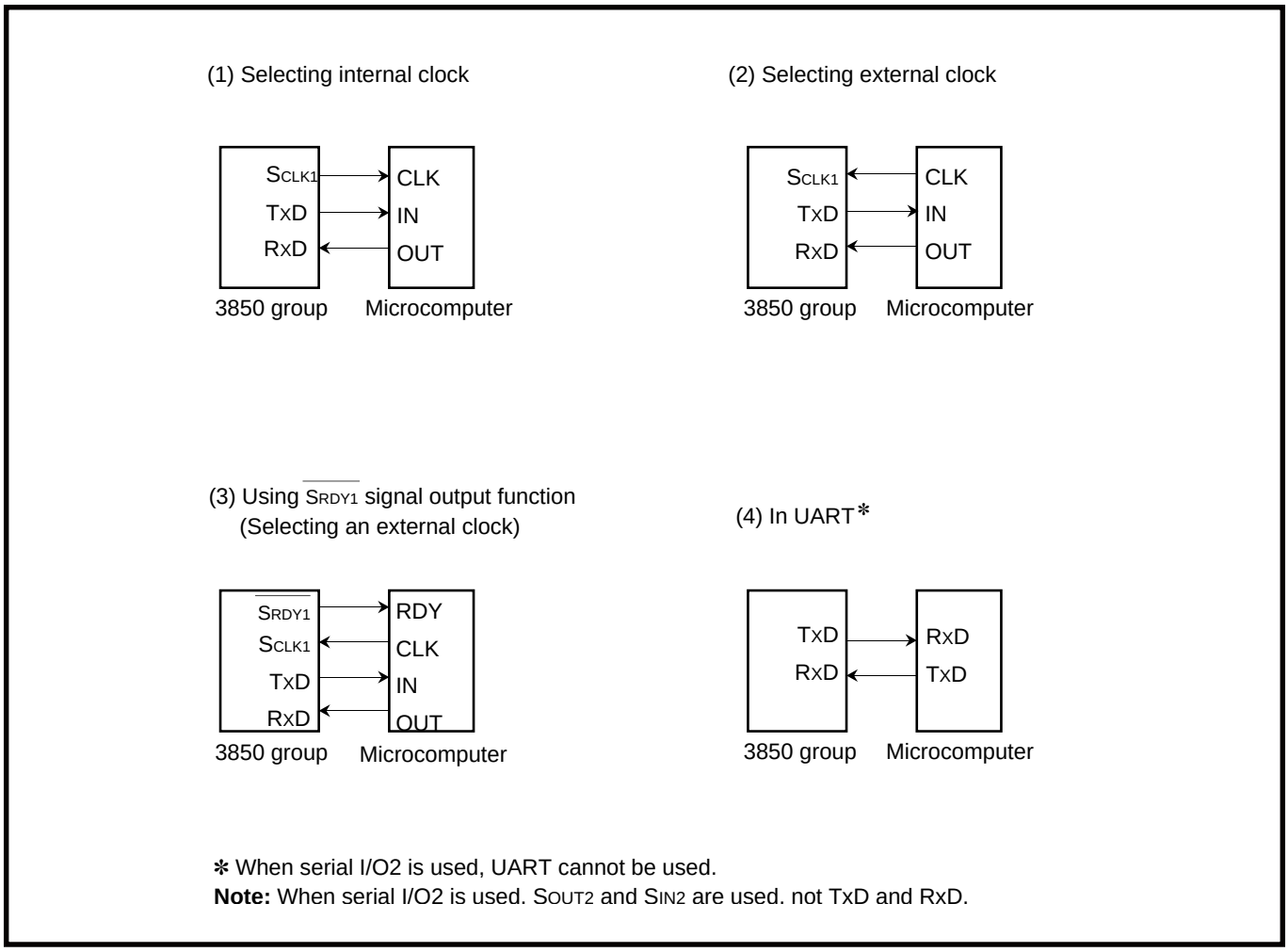


Fig. 2.4.15 Serial I/O connection examples (1)

(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.

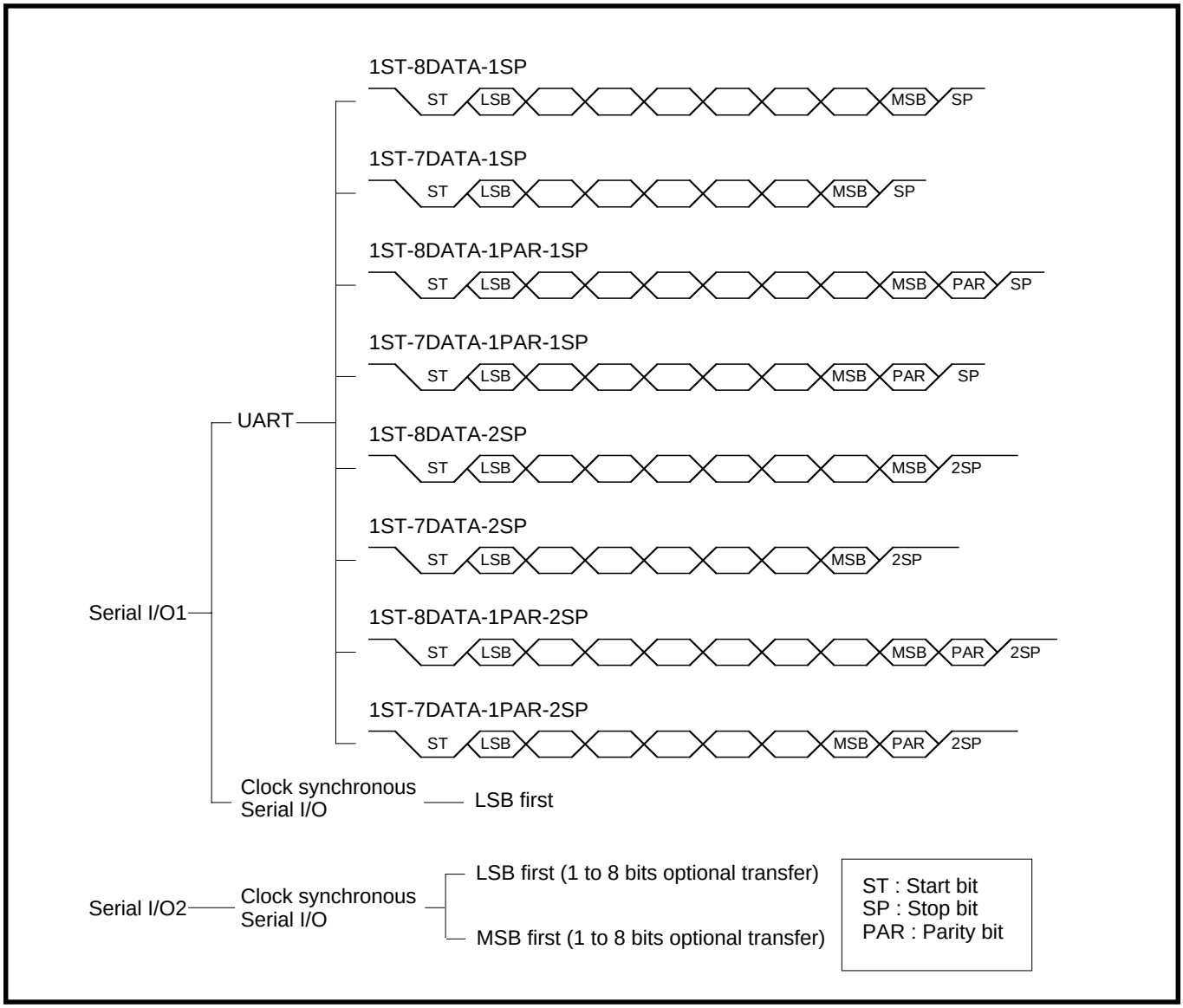


Fig. 2.4.17 Serial I/O transfer data format

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.

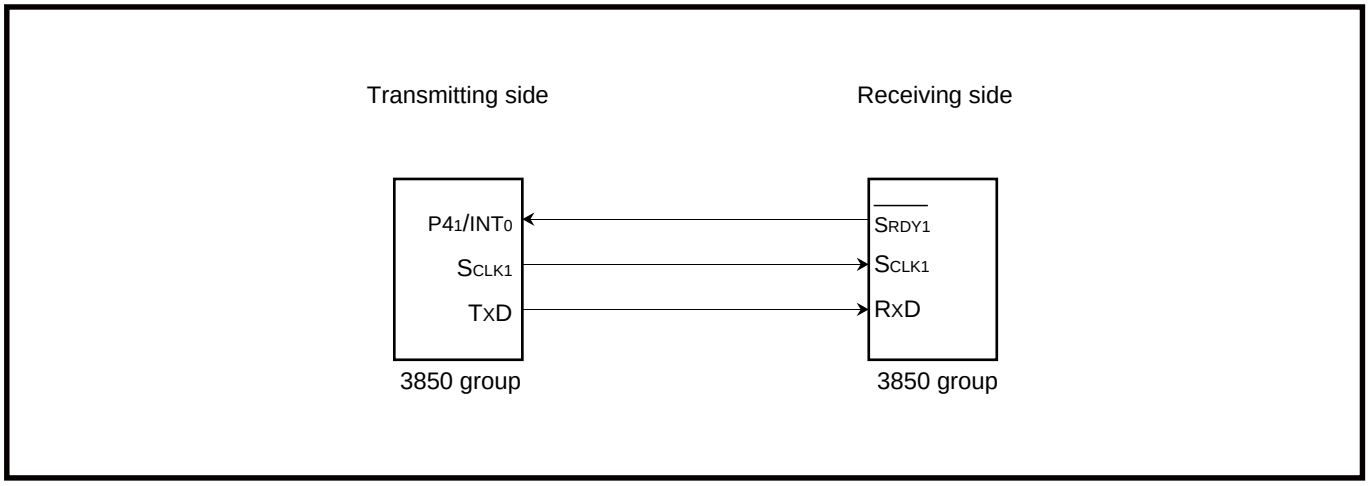


Fig. 2.4.18 Connection diagram

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.

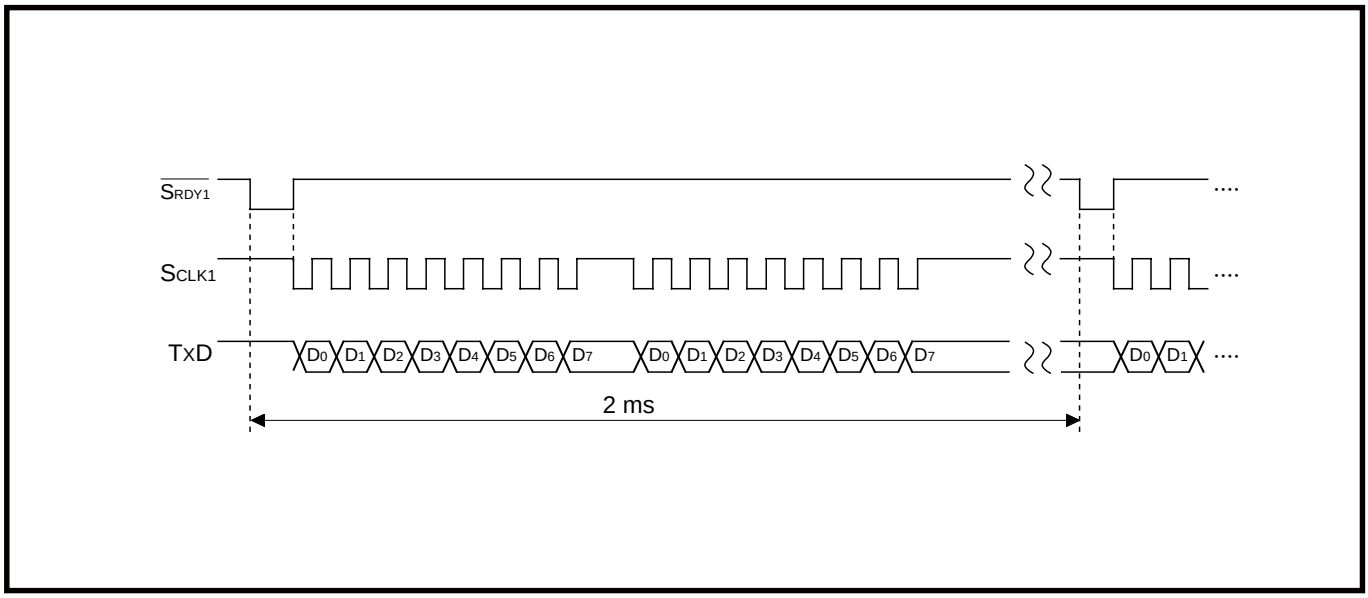
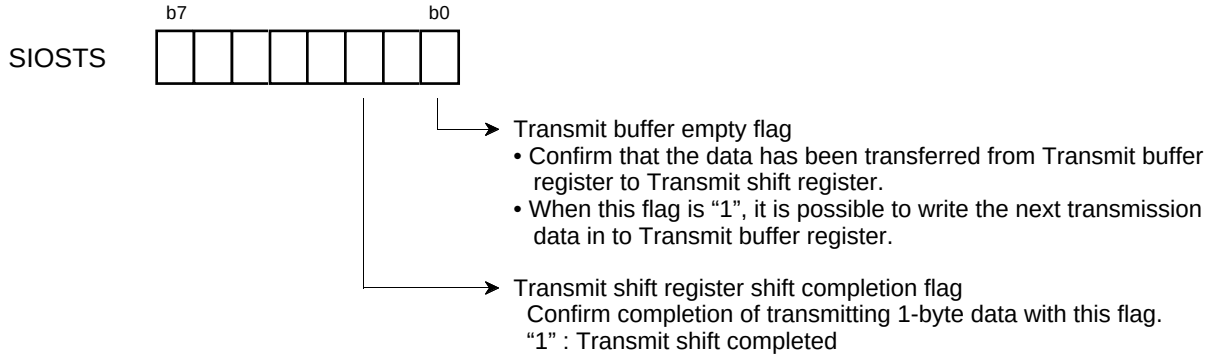


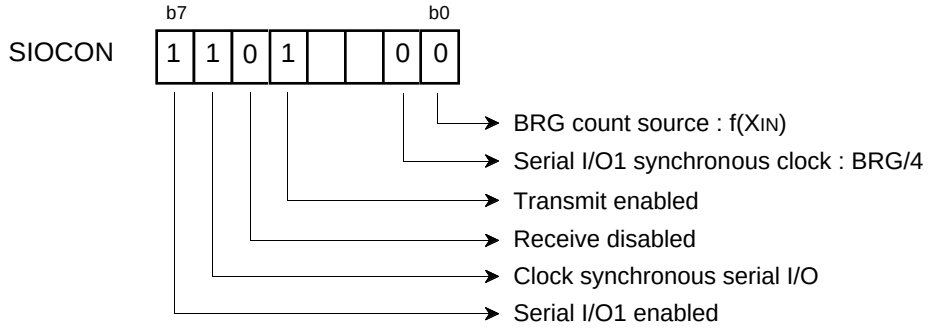
Fig. 2.4.19 Timing chart

Transmitting side

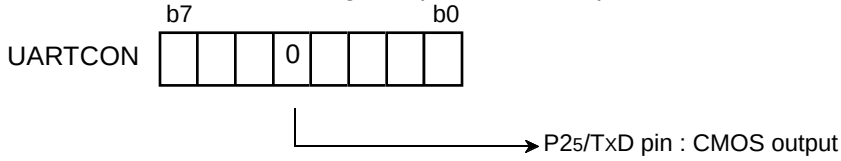
Serial I/O1 status register (Address : 19_{16})



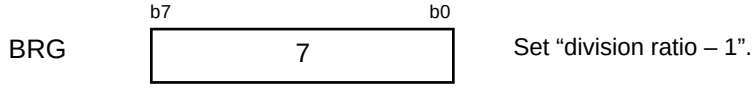
Serial I/O1 control register (Address : $1A_{16}$)



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



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



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

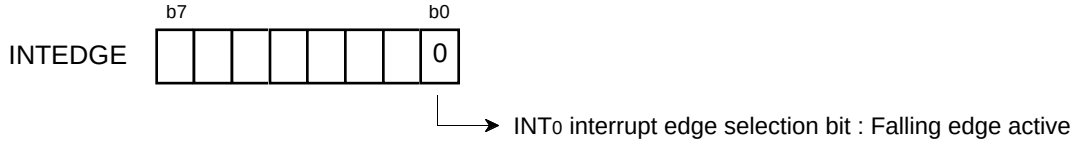


Fig. 2.4.20 Registers setting relevant to transmitting side

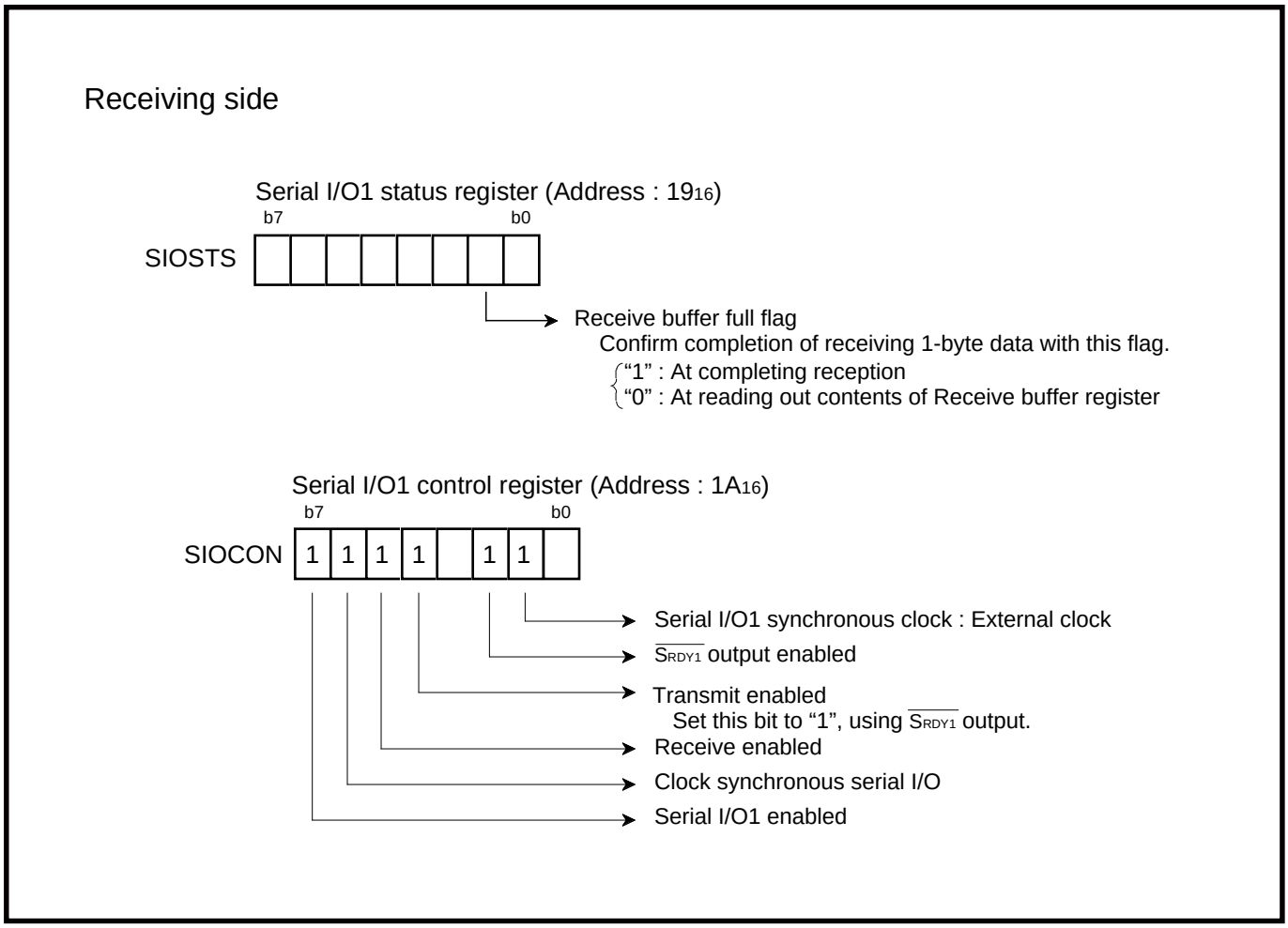


Fig. 2.4.21 Registers setting relevant to receiving side

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.

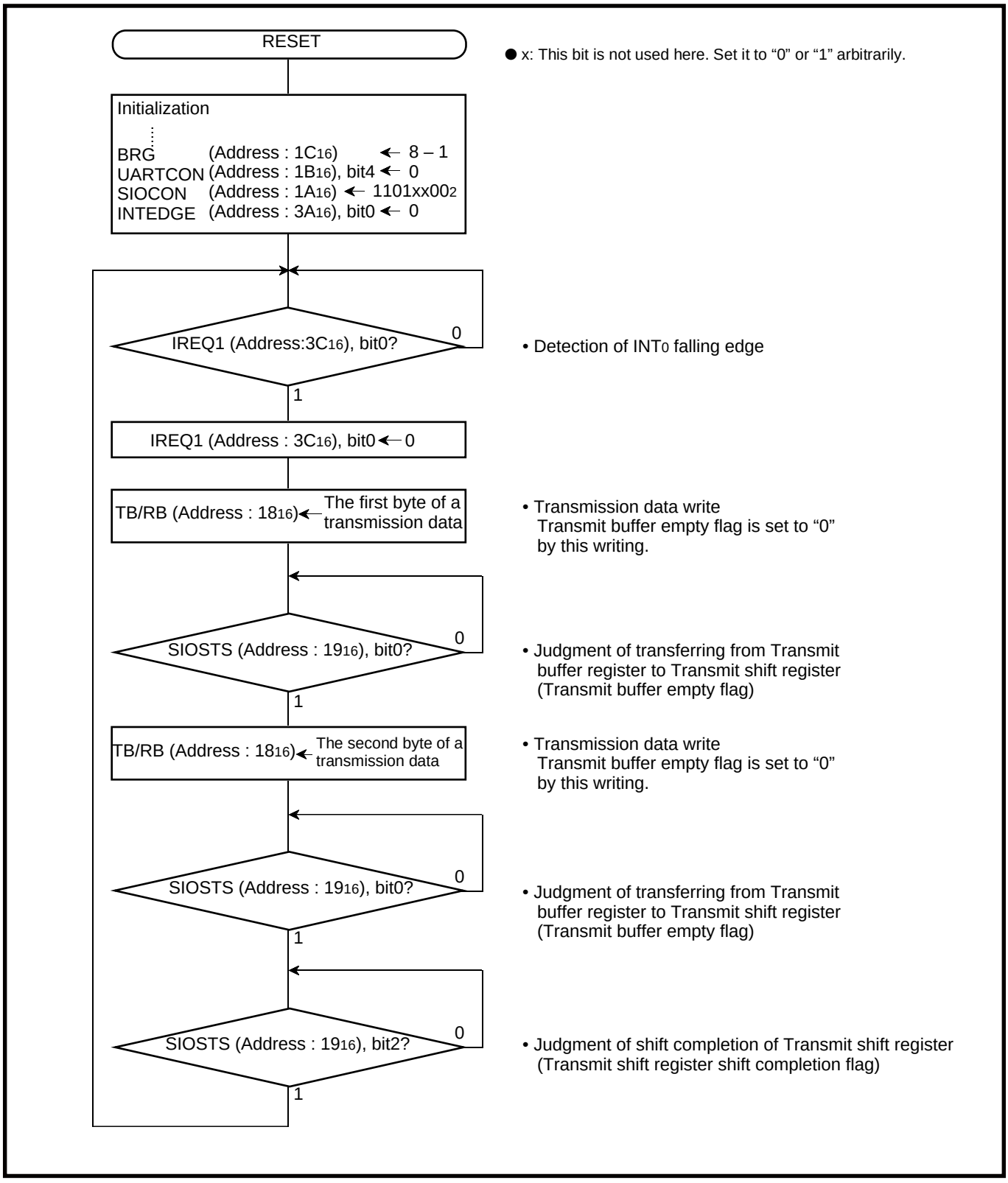


Fig. 2.4.22 Control procedure of transmitting side

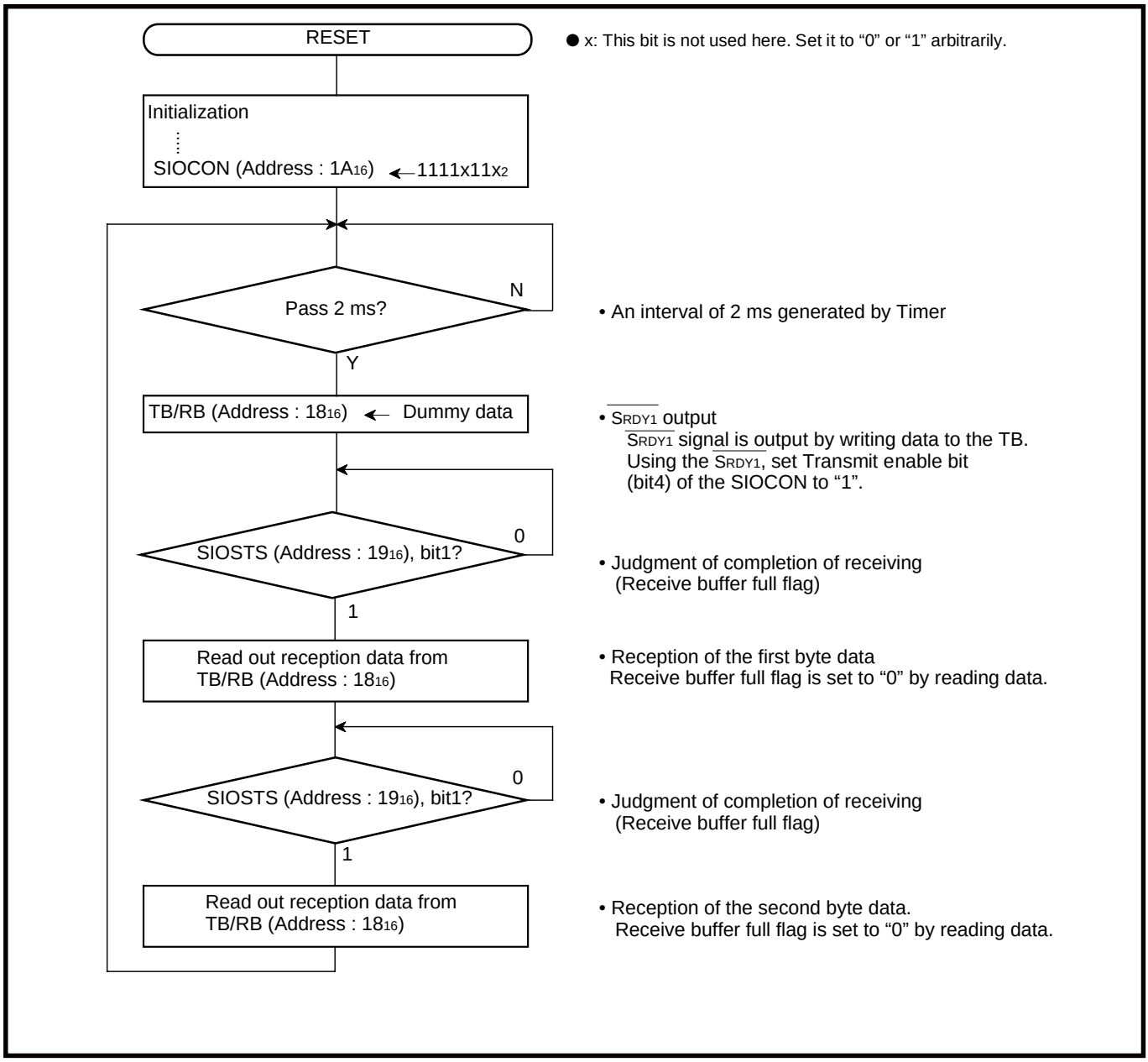


Fig. 2.4.23 Control procedure of receiving side

(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.

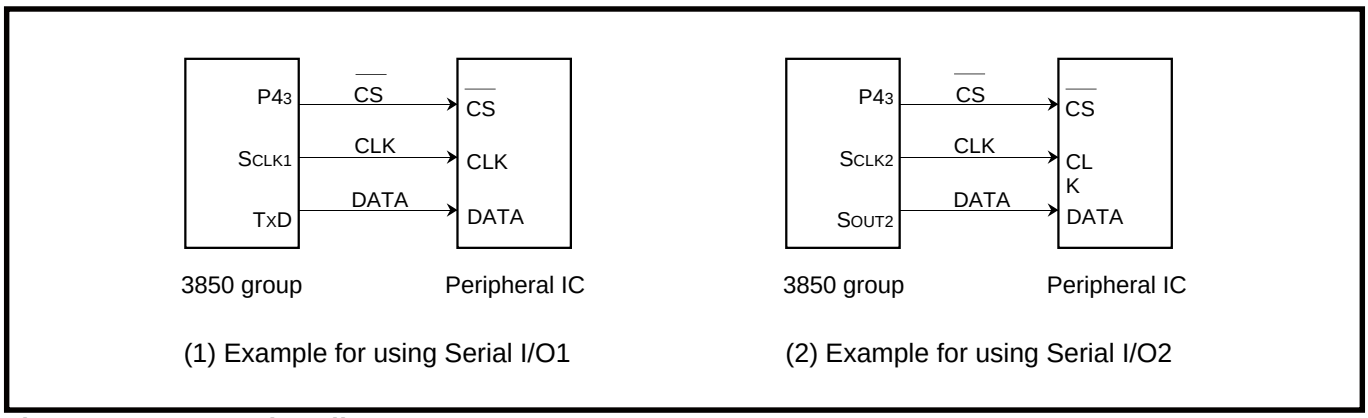
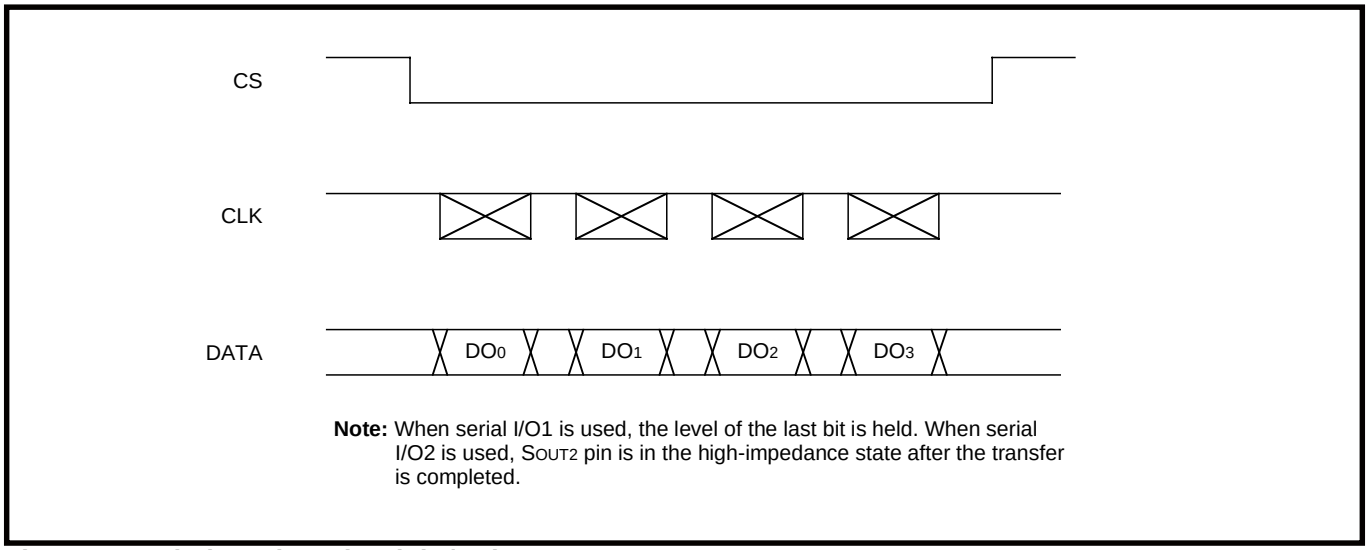


Fig. 2.4.24 Connection diagram

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.

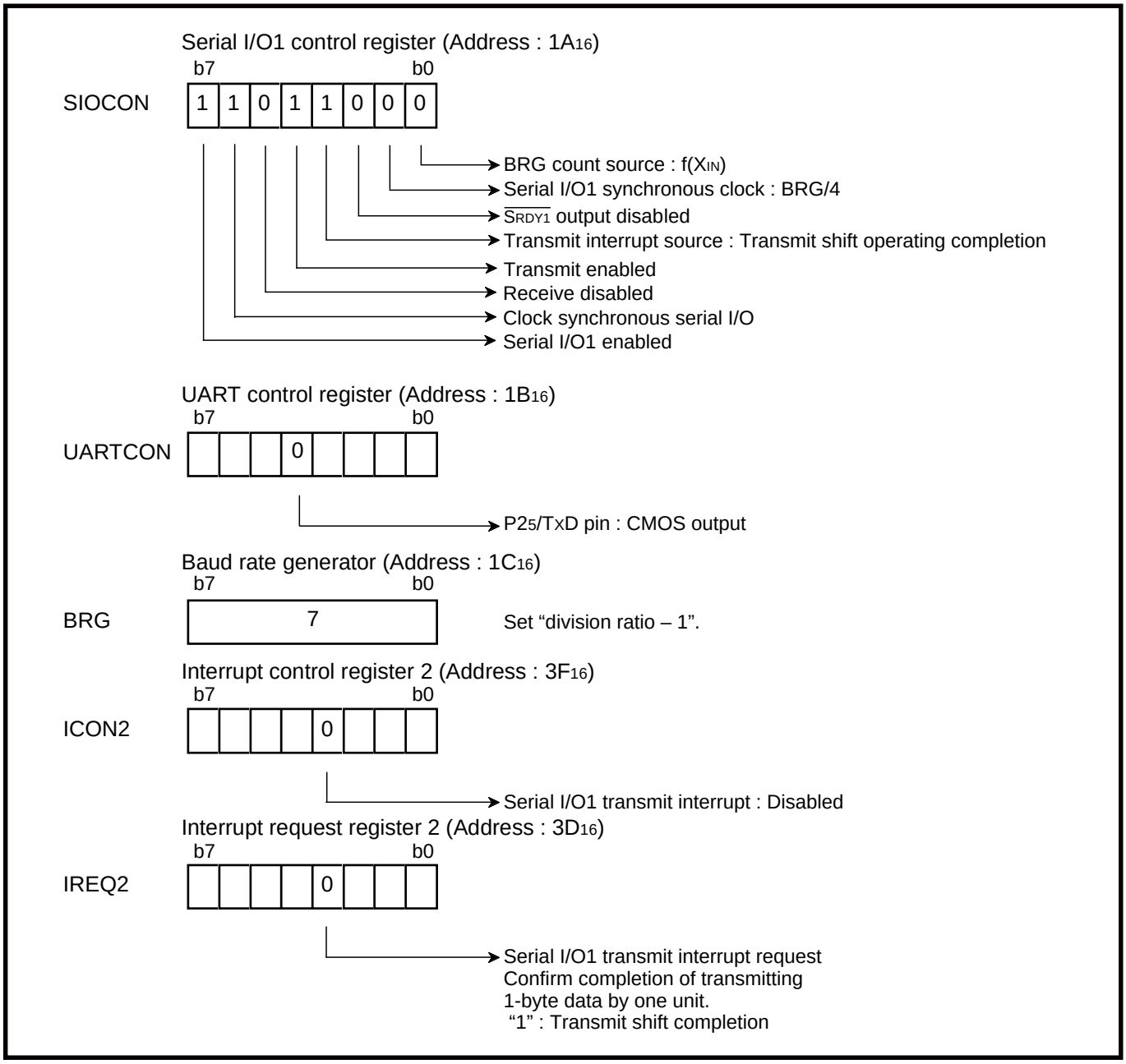


Fig. 2.4.26 Registers setting relevant to Serial I/O1

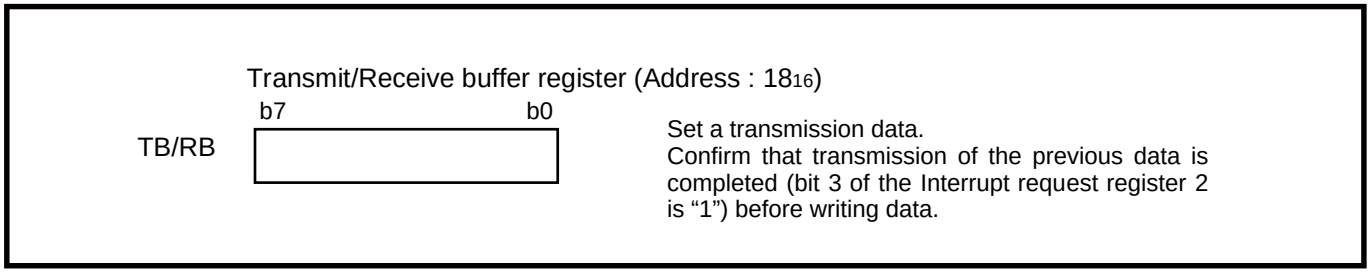


Fig. 2.4.27 Setting of serial I/O1 transmission data

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.

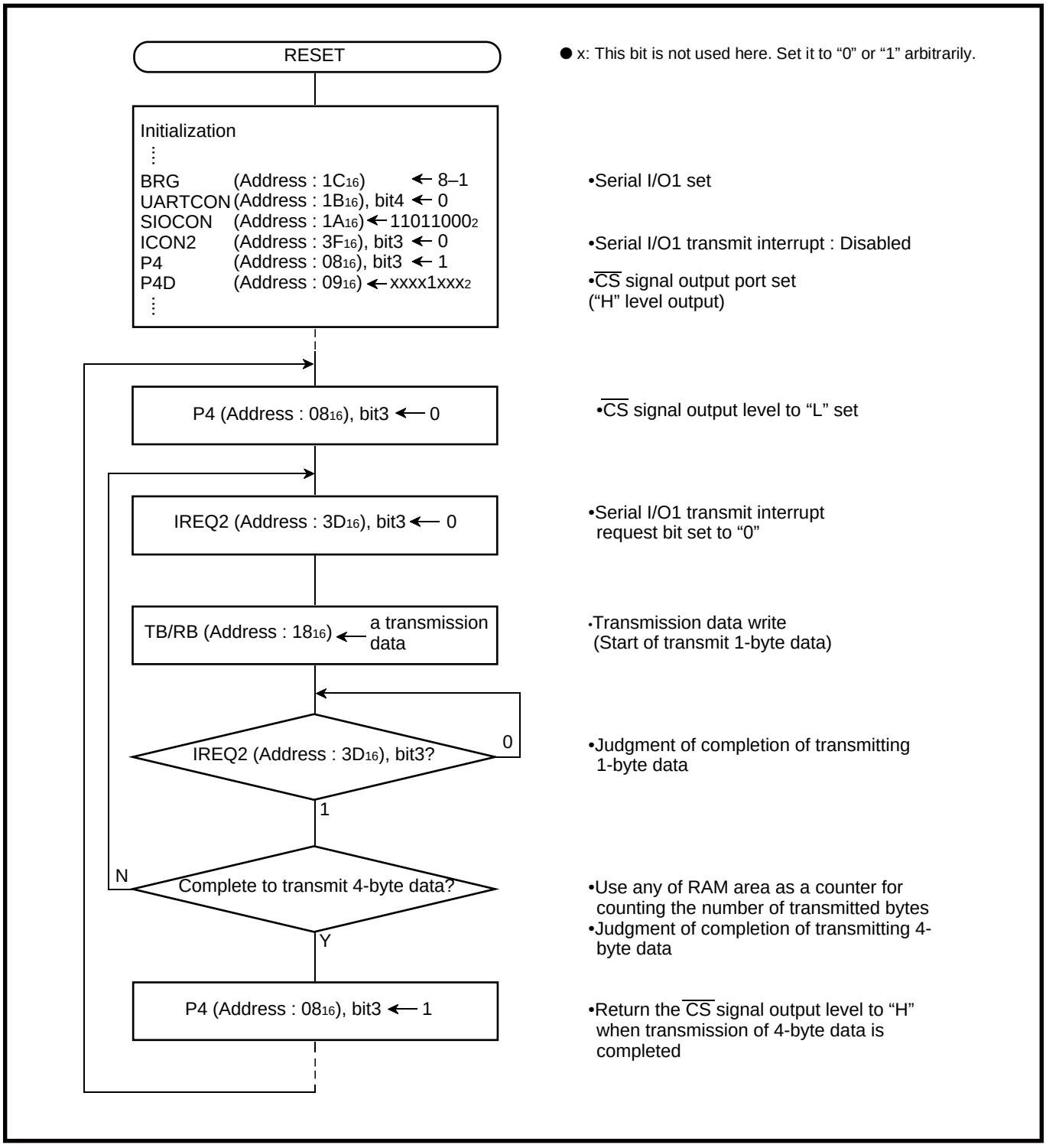


Fig. 2.4.28 Control procedure of Serial I/O1

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.

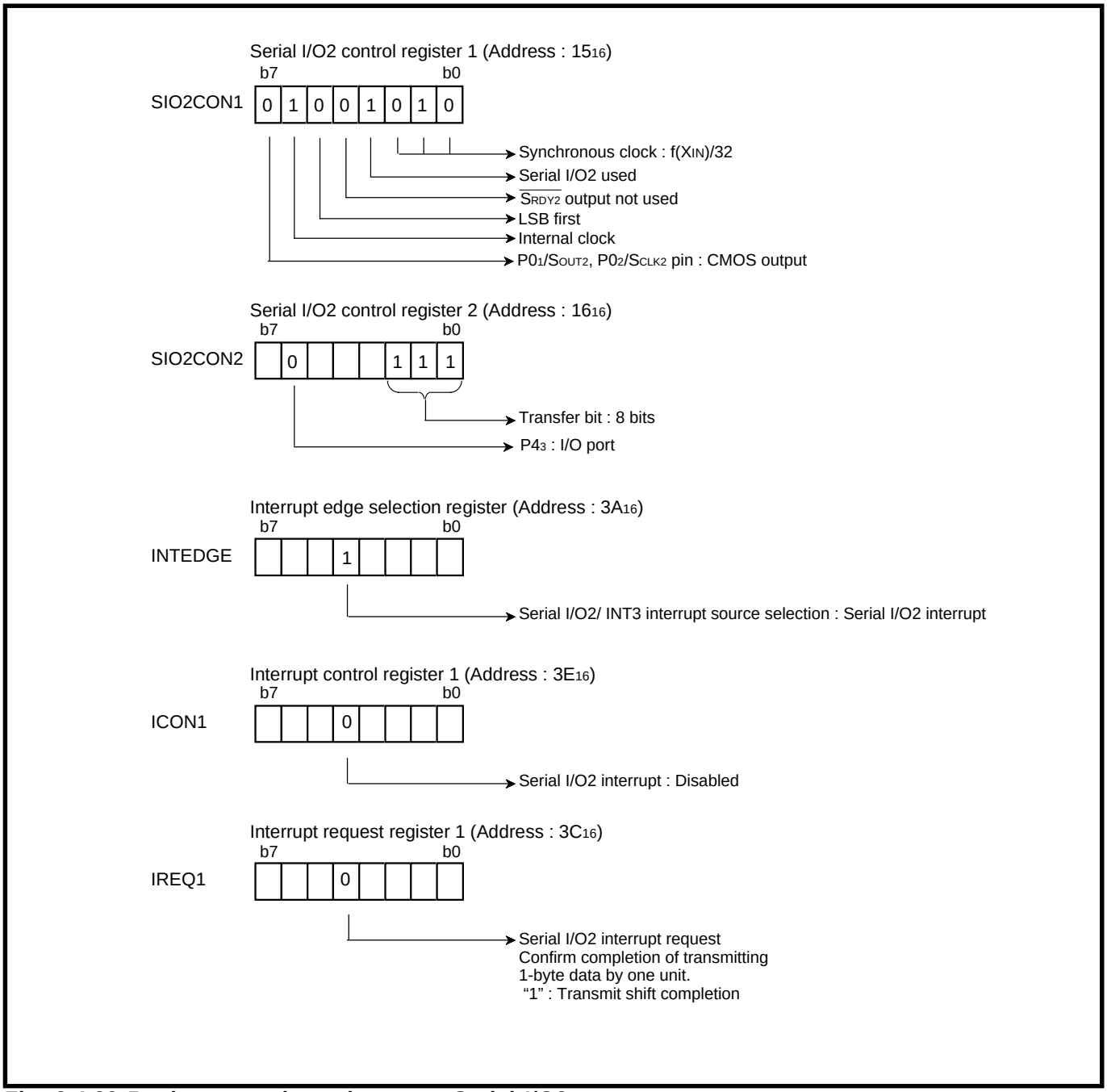


Fig. 2.4.29 Registers setting relevant to Serial I/O2

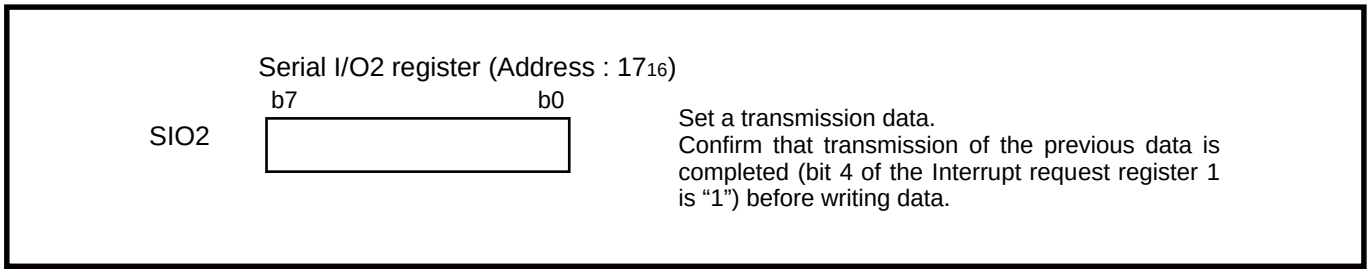


Fig. 2.4.30 Setting of serial I/O2 transmission data

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.

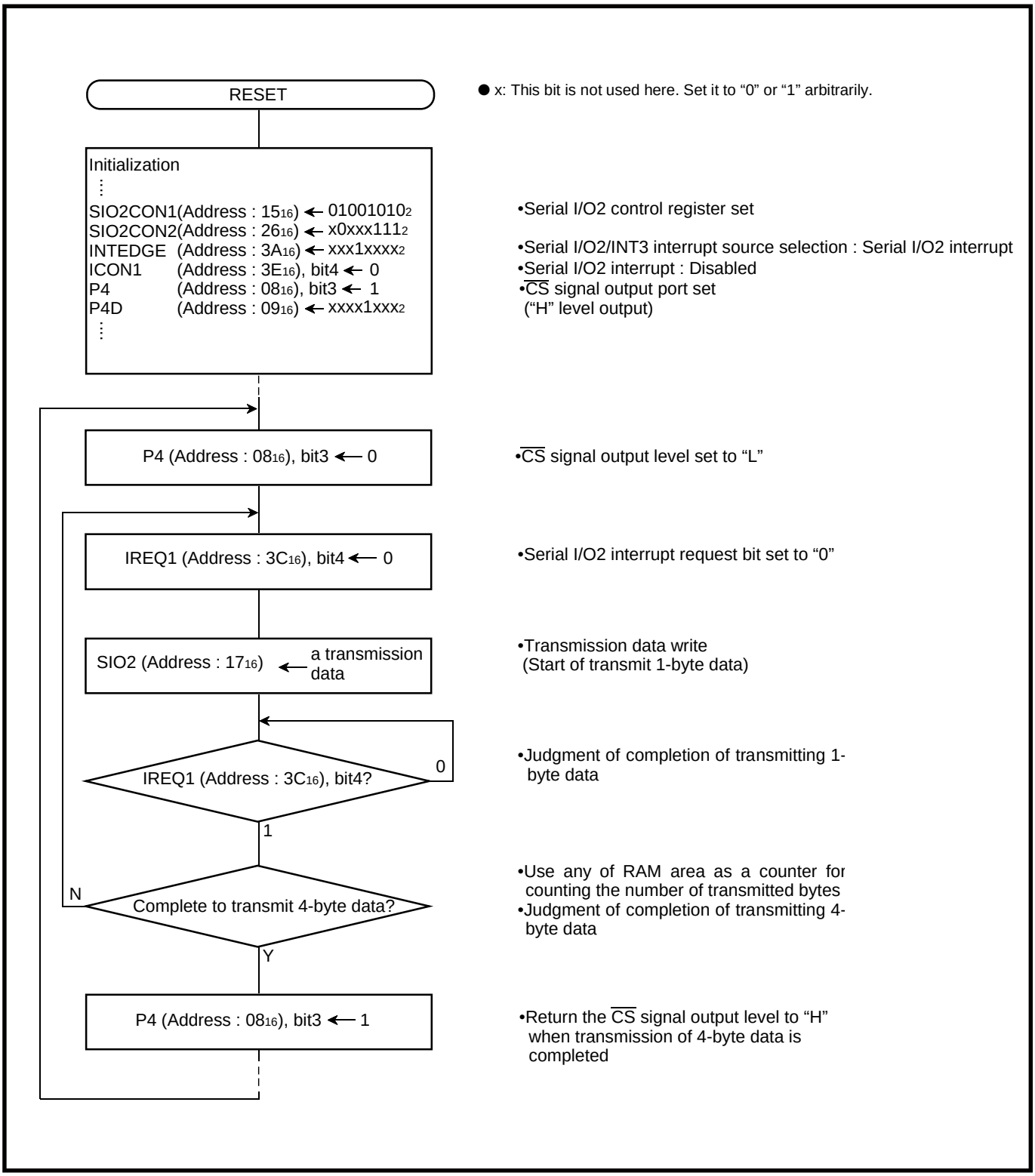


Fig. 2.4.31 Control procedure of Serial I/O2

(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.

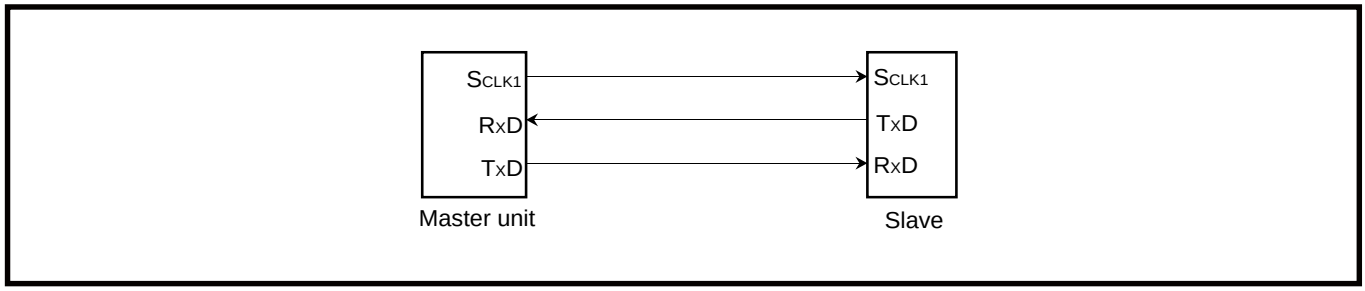


Fig. 2.4.32 Connection diagram

Specifications :

- The serial I/O1 is used (clock synchronous serial I/O is selected).
- Synchronous clock frequency : 125 kHz ($f(X_{IN})$ = 4 MHz is divided by 32)
- Byte cycle: 500 µs
- Number of bytes for transmission or reception : 8 byte/block
- Block transfer cycle : 16 ms
- Block transfer term : 4 ms
- Interval between blocks : 12 ms
- Heading adjustment time : 8 ms

Master side control

- Data is transmitted and received by interrupt routine executed every byte cycle (500 µs)

Slave side control

- Data is transmitted and received by serial I/O1 receive interrupt routine.
- The heading adjustment is carried out by interrupt routine executed every 1 ms.

Limitations of specifications :

- Reading of the reception data and writing 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 436 µs).
- "Heading adjustment time < interval between blocks" must be satisfied.

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.

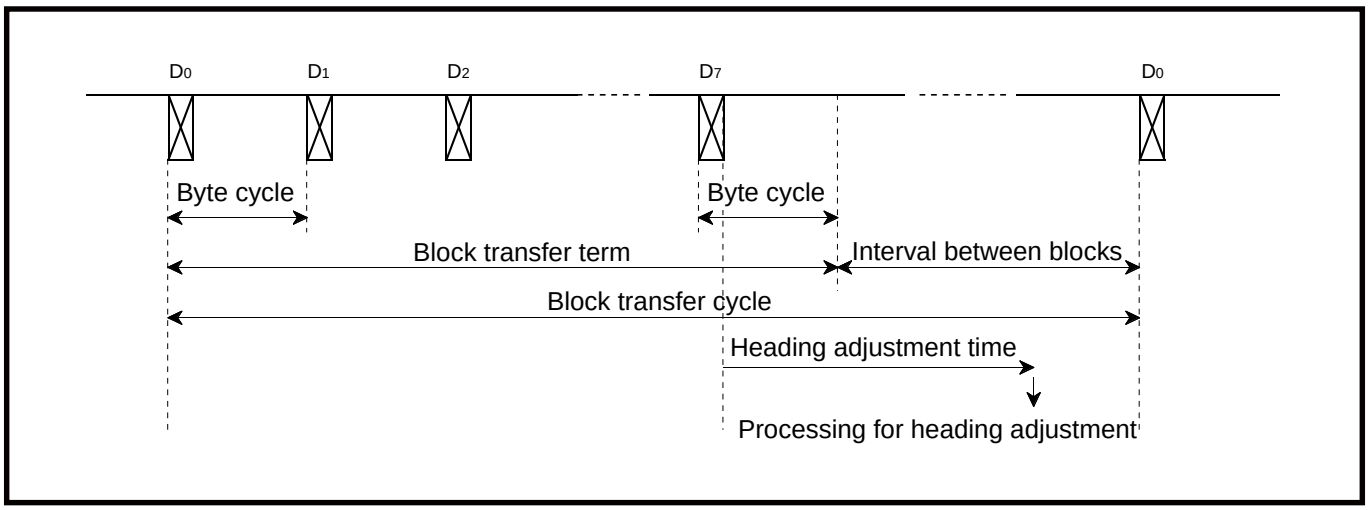


Fig. 2.4.33 Timing chart

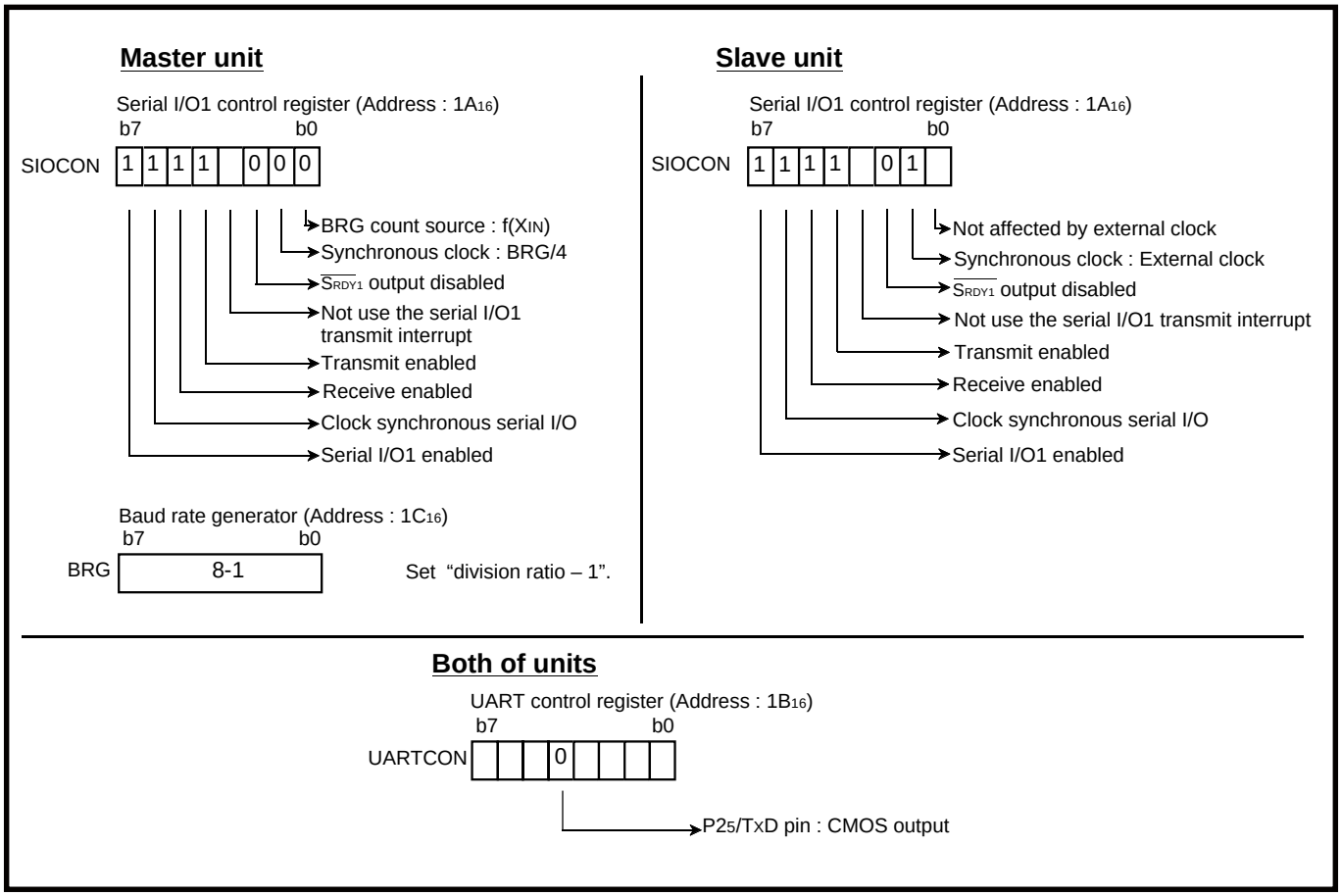


Fig. 2.4.34 Relevant registers setting

Control procedure :

● Control in the master unit

The master unit starts transmission or reception by writing transmission data to the transmit buffer register in the interrupt routine executed every 500 µs. In this interrupt routine, read the reception data before the next transmission data is written to the transmit buffer register. Additionally, transmission and reception of one block (8 bytes) is controlled and the block interval is generated. Figure 2.4.35 shows the control procedure of the master unit.

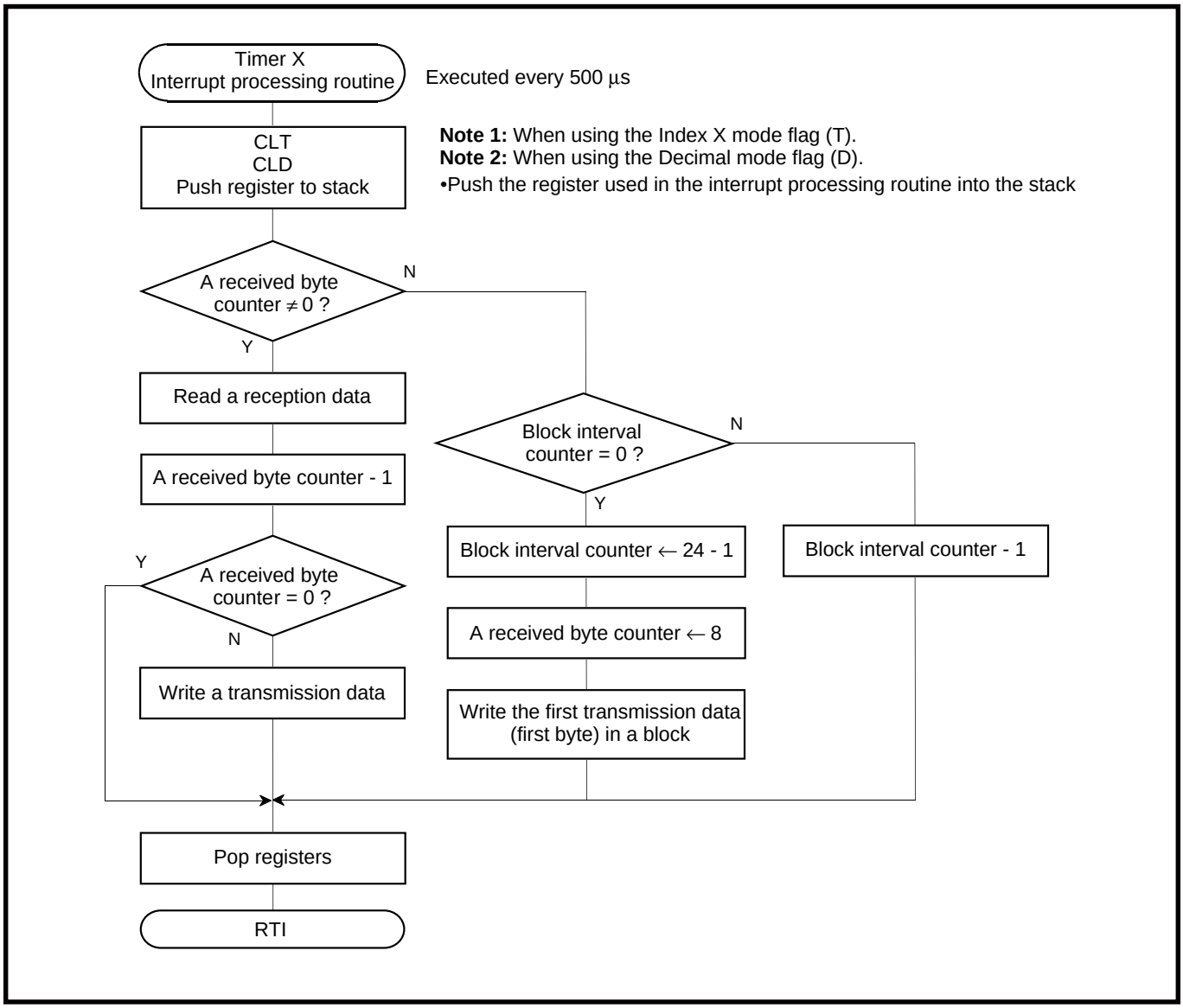


Fig. 2.4.35 Control procedure of master unit

● 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/O1 receive interrupt occurs for a certain time (heading adjustment time or more), the following processing will be performed in the interrupt routine executed every 1 ms.

1. Serial I/O1 is initialized.
2. The first 1-byte data of the transmission data in the block is written into the transmit buffer register.
3. Since the data to be received next is processed as the first 1 byte of the received data in the block, the receive byte counter is initialized.

Figure 2.4.36 shows a control procedure of the slave unit.

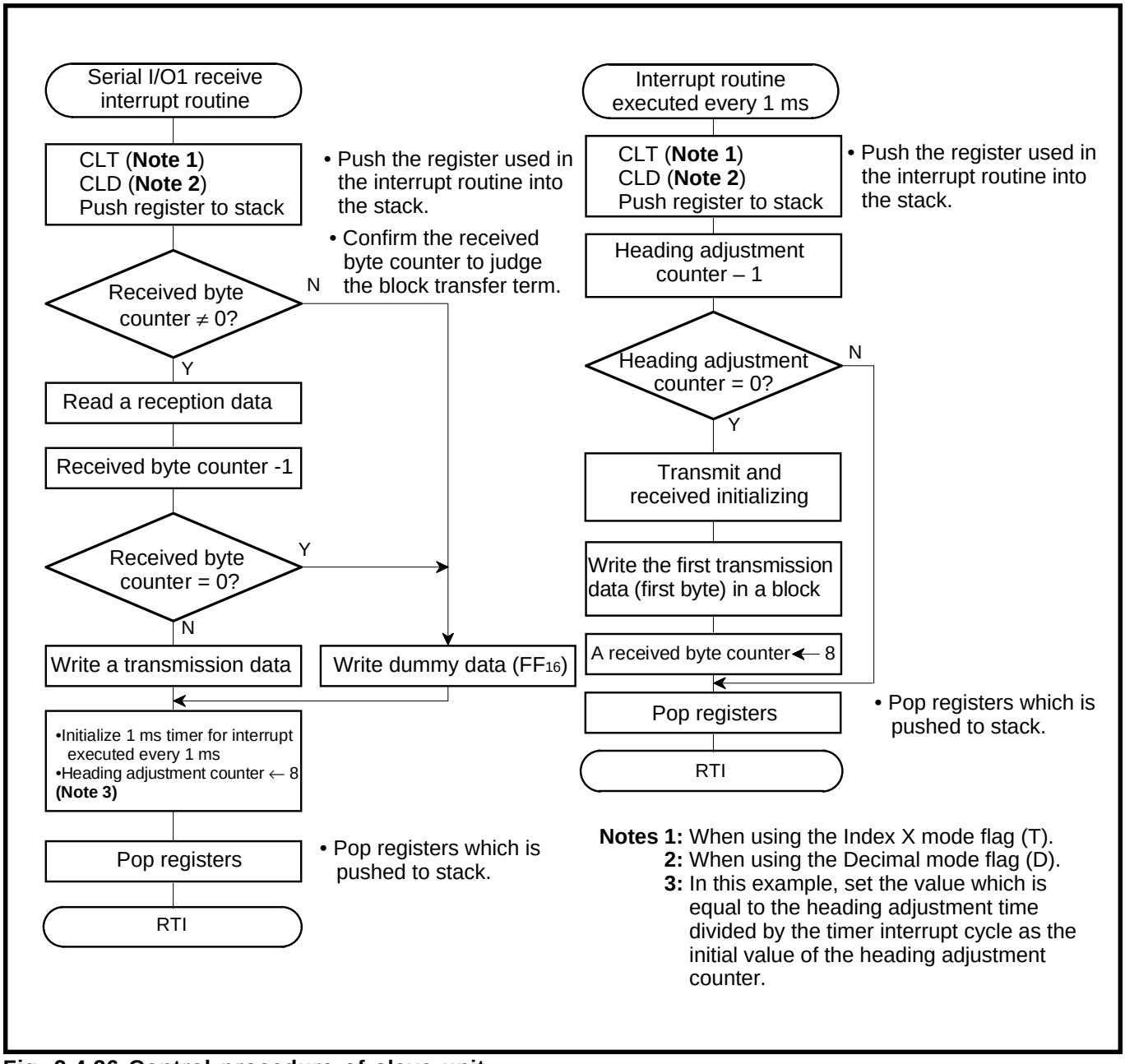


Fig. 2.4.36 Control procedure of slave unit

(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.

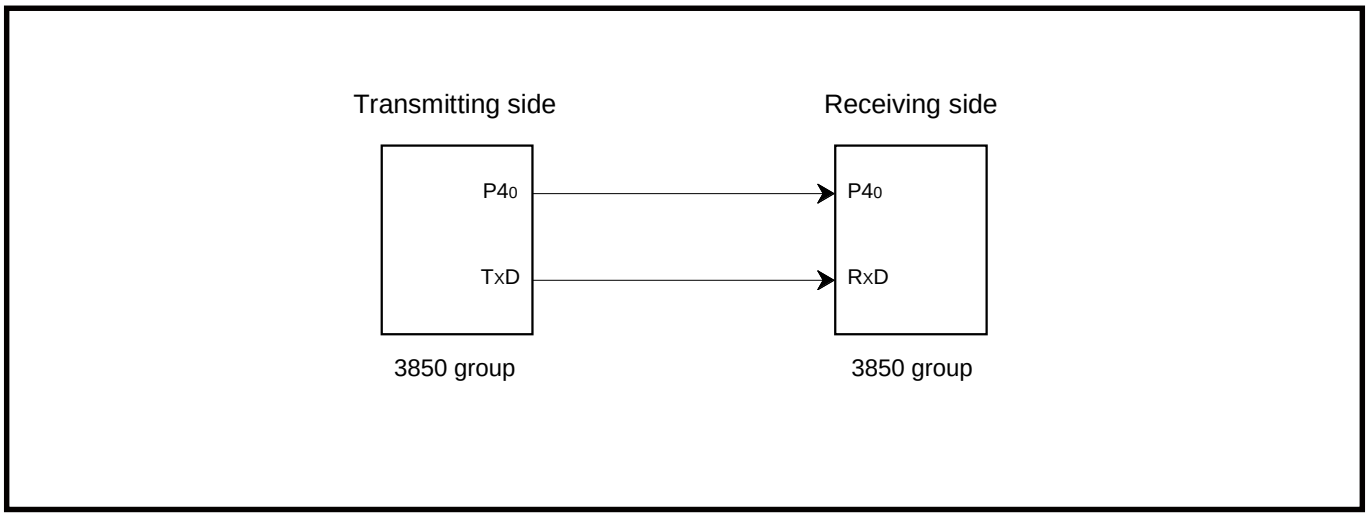


Fig. 2.4.37 Connection diagram

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.

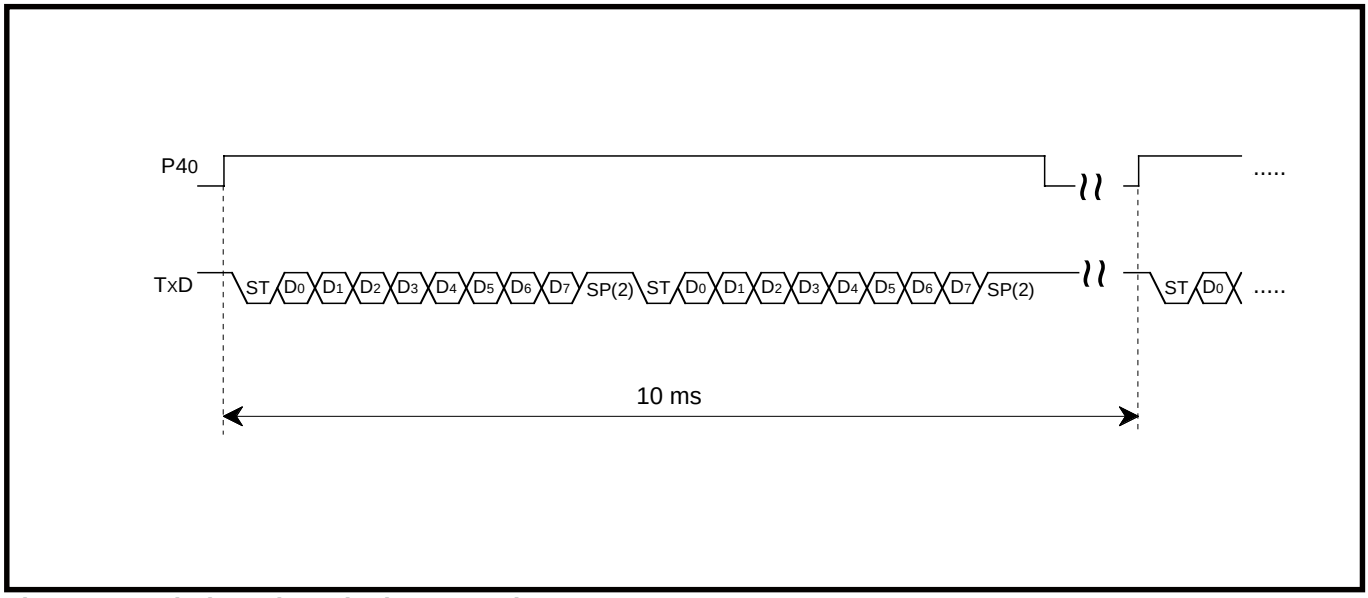


Fig. 2.4.38 Timing chart (using UART)

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(FF16)	300	488.28125
f(XIN)/4	127(7F16)	600	976.5625
f(XIN)/4	63(3F16)	1200	1953.125
f(XIN)/4	31(1F16)	2400	3906.25
f(XIN)/4	15(0F16)	4800	7812.5
f(XIN)/4	7(0716)	9600	15625
f(XIN)/4	3(0316)	19200	31250
f(XIN)/4	1(0116)	38400	62500
f(XIN)	3(0316)	76800	125000
f(XIN)	1(0116)	153600	250000
f(XIN)	0(0016)	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(CF16)	600
f(XIN)/4	103(6716)	1200
f(XIN)/4	51(3316)	2400
f(XIN)/4	25(1916)	4800
f(XIN)/4	12(0C16)	9600
f(XIN)	25(1916)	19200
f(XIN)	12(0C16)	38400

Notes 1: Select the BRG count source with bit 0 of the serial I/O1 control register (Address : 1A16).

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 : 1A16) is set to “0”, a value of m is 1.
When bit 0 of the serial I/O1 control register (Address : 1A16) is set to “1”, a value of m is 4.

Transmitting side

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

b7 b0

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 : $1A_{16}$)

b7 b0

SIOCON 1 0 0 1 _ 0 0 1

- BRG count source : f(X_{IN})/4
- Serial I/O1 synchronous clock : BRG/16
- $\overline{S_{RDY1}}$ output disabled
- Transmit enabled
- Receive disabled
- Asynchronous serial I/O (UART)
- Serial I/O1 enabled

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

b7 b0

UARTCON _ _ _ 0 1 _ 0 0

- Character length : 8 bits
- Parity checking disabled
- Stop bit length : 2 stop bits
- T_XD : CMOS output

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

b7 b0

BRG 7

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

✻ 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})

b7 b0

SIOSTS

- Receive buffer full flag
  Confirm completion of receiving 1-byte data with this flag.
  - "1" : At completing reception
  - "0" : At reading out contents of Receive buffer register
- Overrun error flag
  "1" : When data is ready in Receive shift register while Receive buffer register contains the data.
- Framing error flag
  "1" : When stop bits cannot be detected at the specified timing.
- Summing error flag
  "1" : When any one of the following errors occurs.
  - Overrun error
  - Framing error

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

b7 b0

SIOCON: 1 0 1 0 _ 0 0 1

- BRG count source : $f(X_{IN})/4$
- Serial I/O1 synchronous clock : BRG/16
- $\overline{S_{RDY1}}$ out disabled
- Transmit disabled
- Receive enabled
- Asynchronous serial I/O(UART)
- Serial I/O1 enabled

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

b7 b0

UARTCON: _ _ _ _ 1 _ 0 0

- Character length selection bit : 8 bits
- Parity enable bit : Parity checking disabled
- Stop bit length selection bit : 2 stop bits

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

b7 b0

BRG: 7

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

✻ 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.

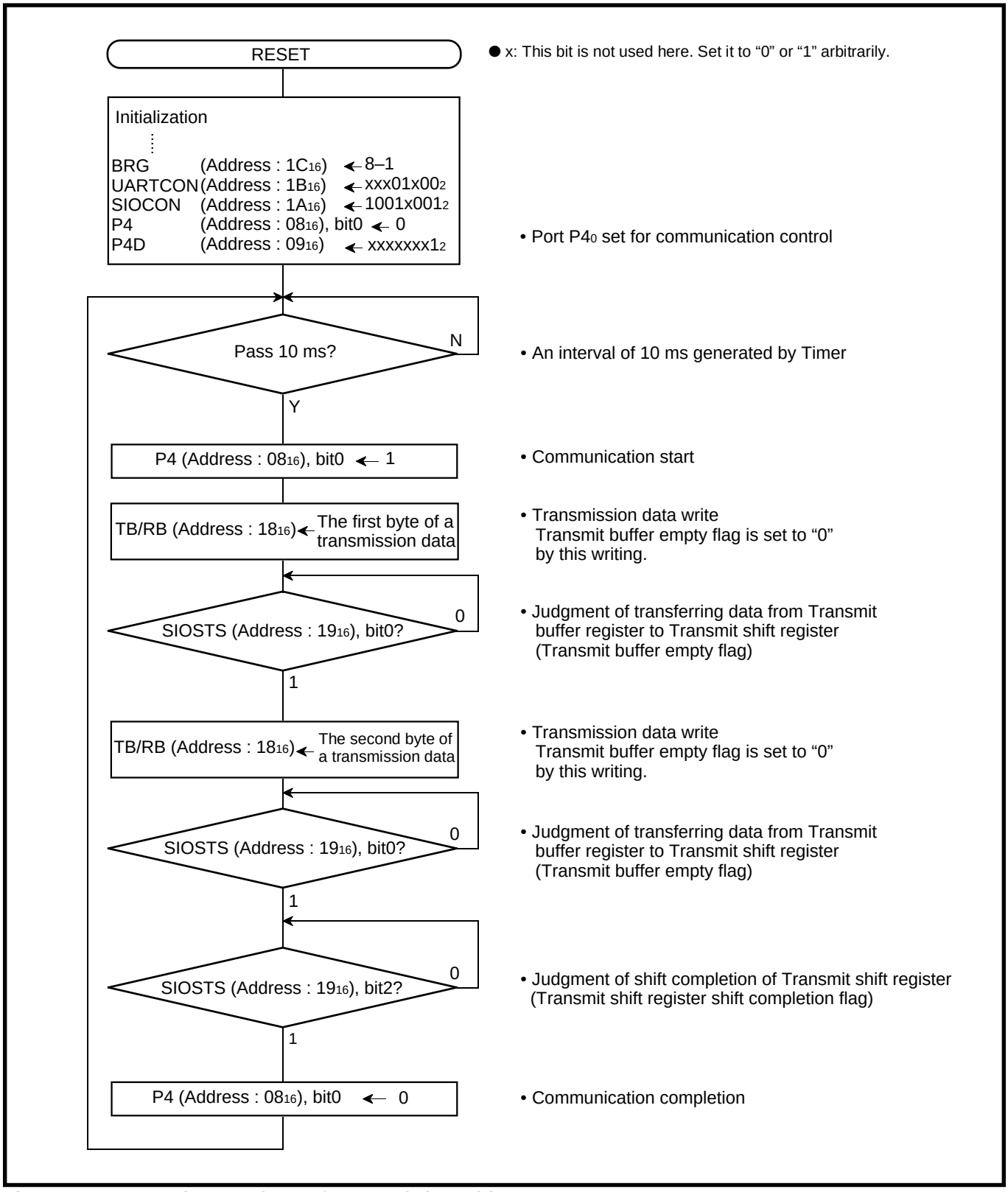


Fig. 2.4.41 Control procedure of transmitting side

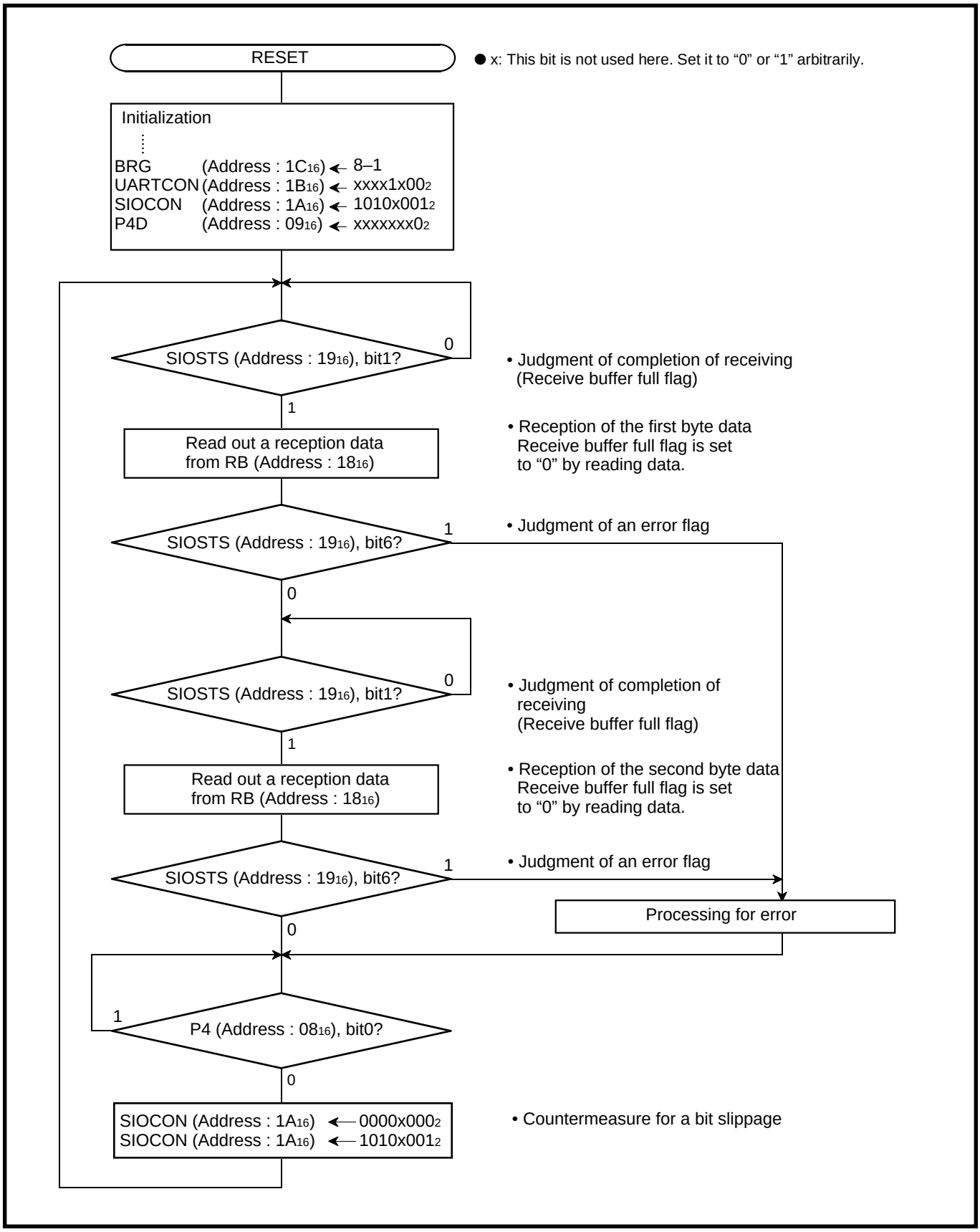


Fig. 2.4.42 Control procedure of receiving side

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”.

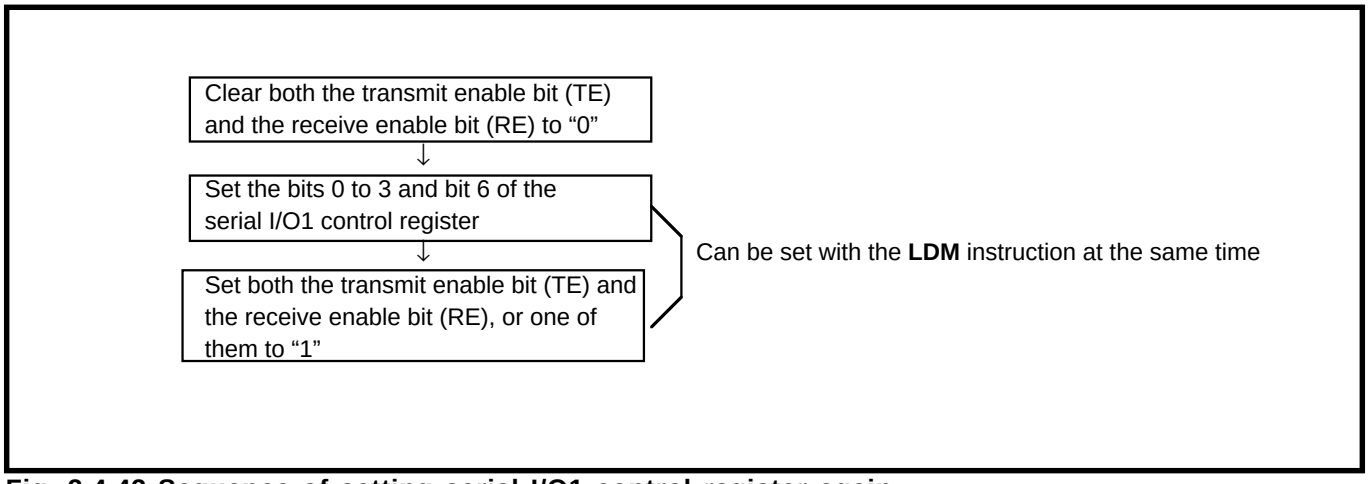


Fig. 2.4.43 Sequence of setting serial I/O1 control register again

(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 S_{CLK1} input level. Also, write the transmit data to the transmit buffer register at “H” of the S_{CLK1} 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

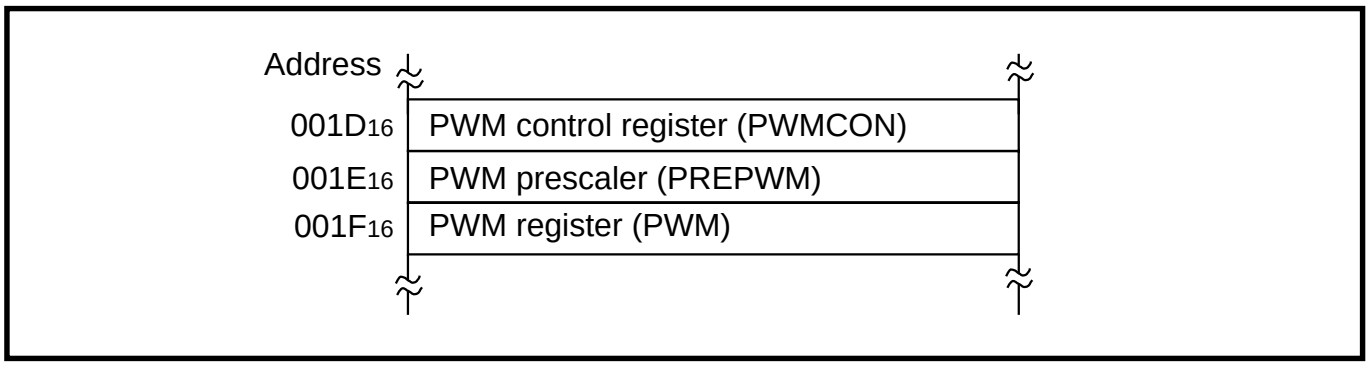


Fig. 2.5.1 Memory map of registers relevant to PWM

2.5.2 Relevant registers

PWM control register

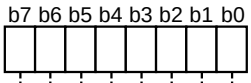
b7 b6 b5 b4 b3 b2 b1 b0

PWM control register (PWMCON: address 1D16)

b	Name	Functions	At reset	R	W
0	PWM function enable bit	0 : PWM disabled 1 : PWM enabled	0	○	○
1	Count source selection bit	0 : f(XIN) 1 : f(XIN)/2	0	○	○
2	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 2.5.2 Structure of PWM control register

PWM prescaler

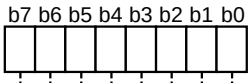


PWM prescaler
(PREPWM: address 1E16)

b	Functions	At reset	R	W
0	•Set the PWM period. •The value set in this register is written to both PWM prescaler pre-latch and PWM prescaler latch at the same time. • When data is written to this register during PWM output, the pulse corresponding to changed value is output at the next period. • When this register is read out, the count value of the PWM prescaler latch is read out.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 2.5.3 Structure of PWM prescaler

PWM register



PWM register
(PWM: address 1F16)

b	Functions	At reset	R	W
0	• Set the PWM “H” level output interval. • The value set in this register is written to both PWM register pre-latch and PWM register latch at the same time. • When data is written to this register during PWM output, the pulse corresponding to changed value is output at the next period. • When this register is read out, the contents of the PWM register latch is read out.	Undefined	○	○
1		Undefined	○	○
2		Undefined	○	○
3		Undefined	○	○
4		Undefined	○	○
5		Undefined	○	○
6		Undefined	○	○
7		Undefined	○	○

Fig. 2.5.4 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.

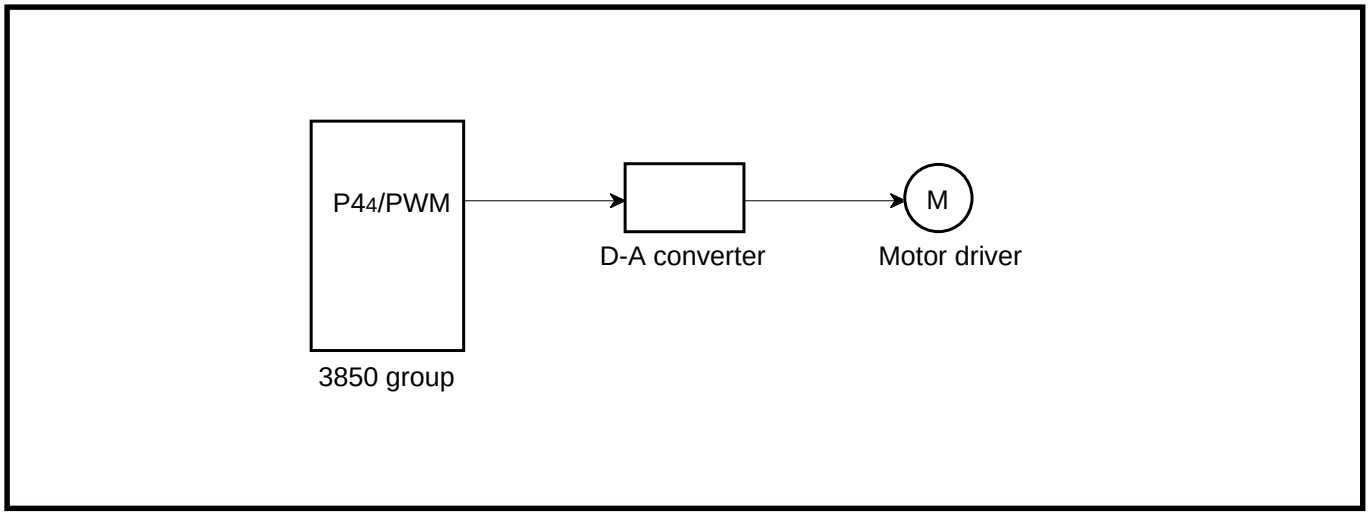


Fig. 2.5.5 Connection diagram

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.

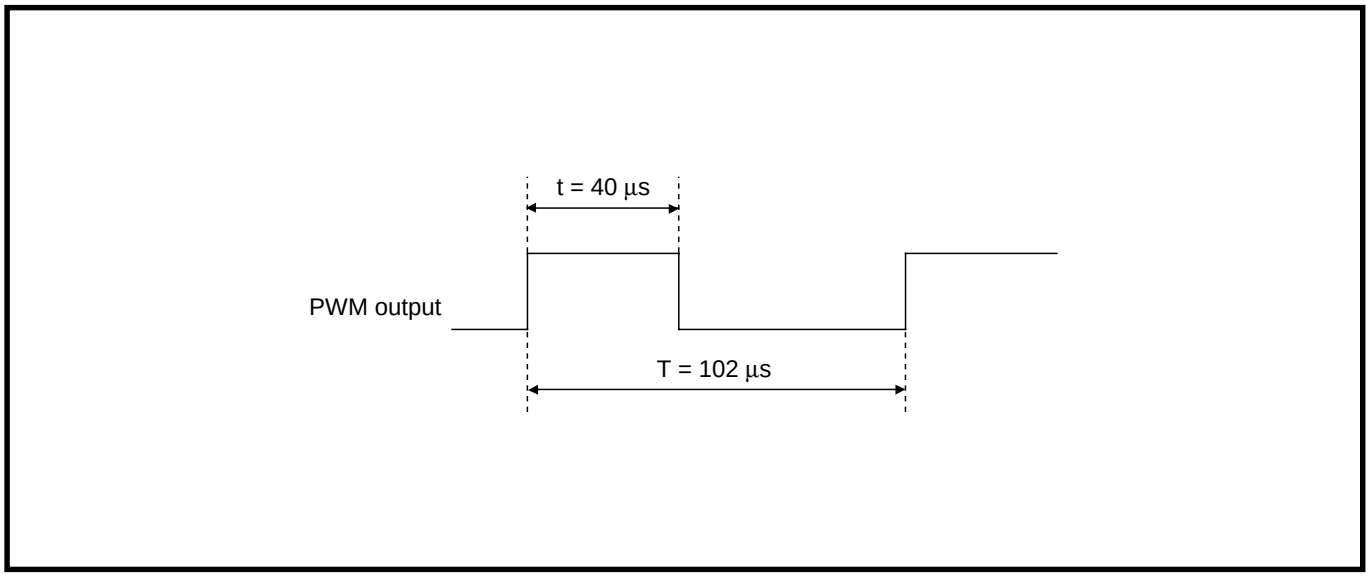


Fig. 2.5.6 PWM output timing

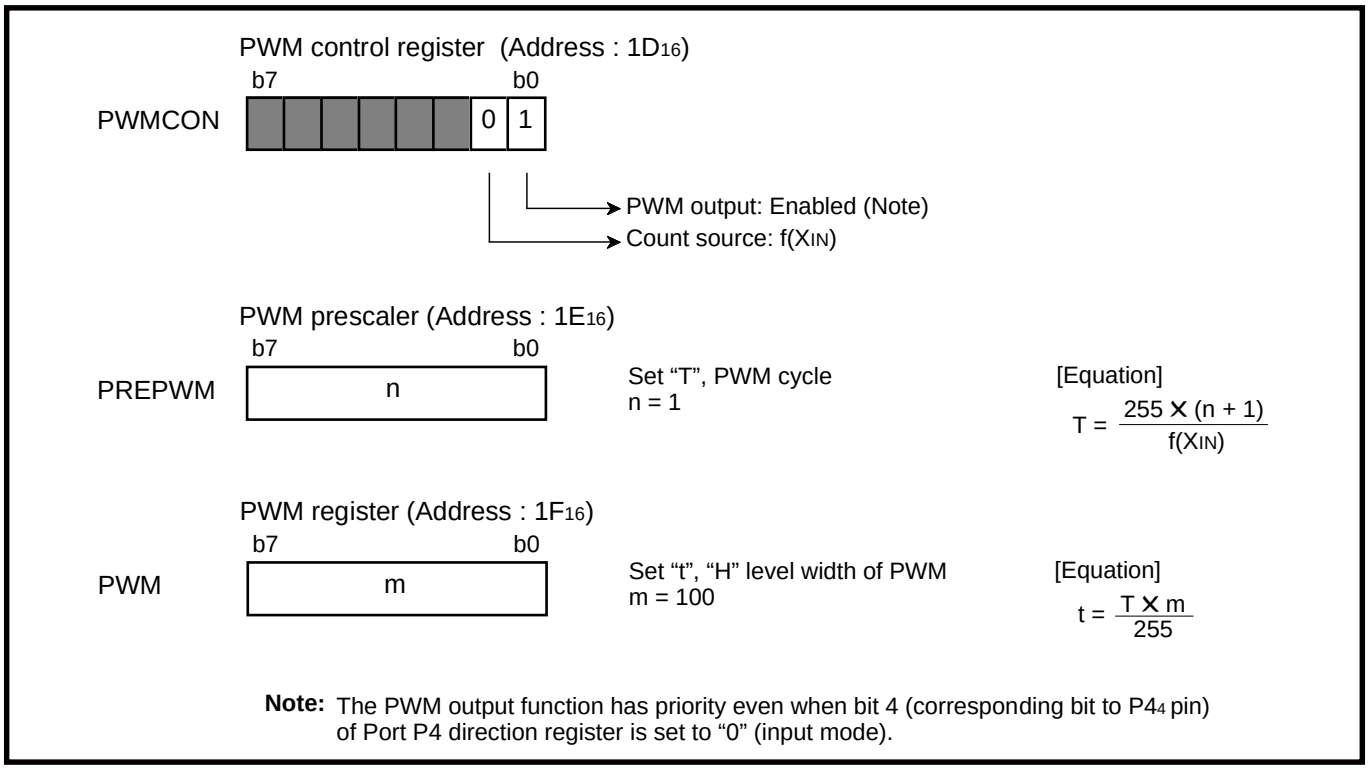


Fig. 2.5.7 Setting of relevant registers

<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.

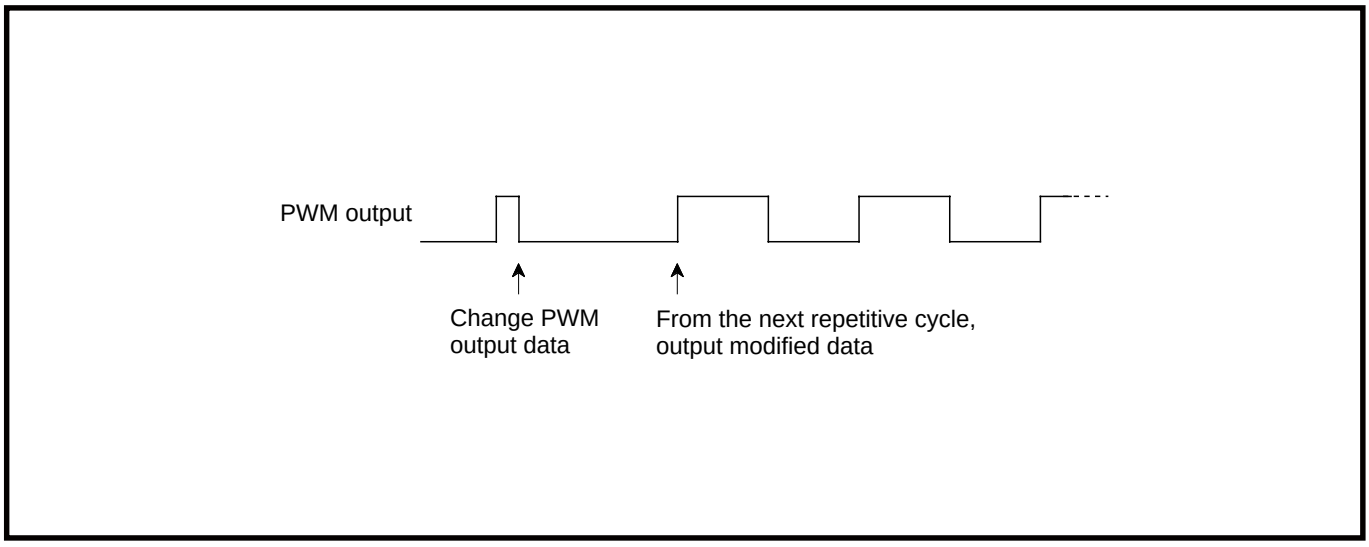


Fig. 2.5.8 PWM output

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.

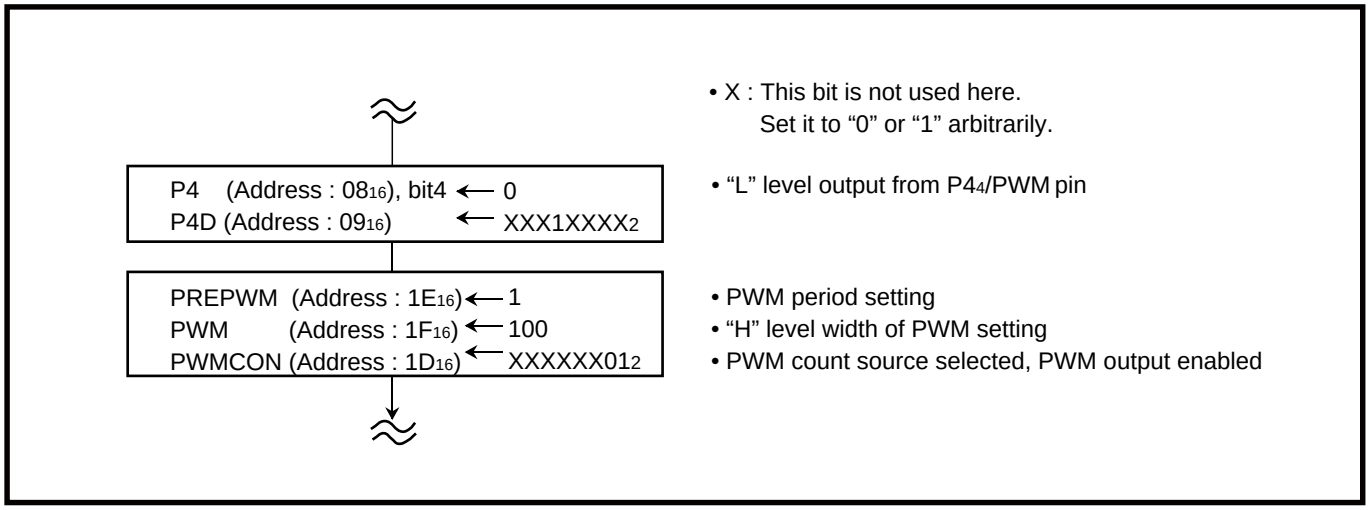


Fig. 2.5.9 Control procedure

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

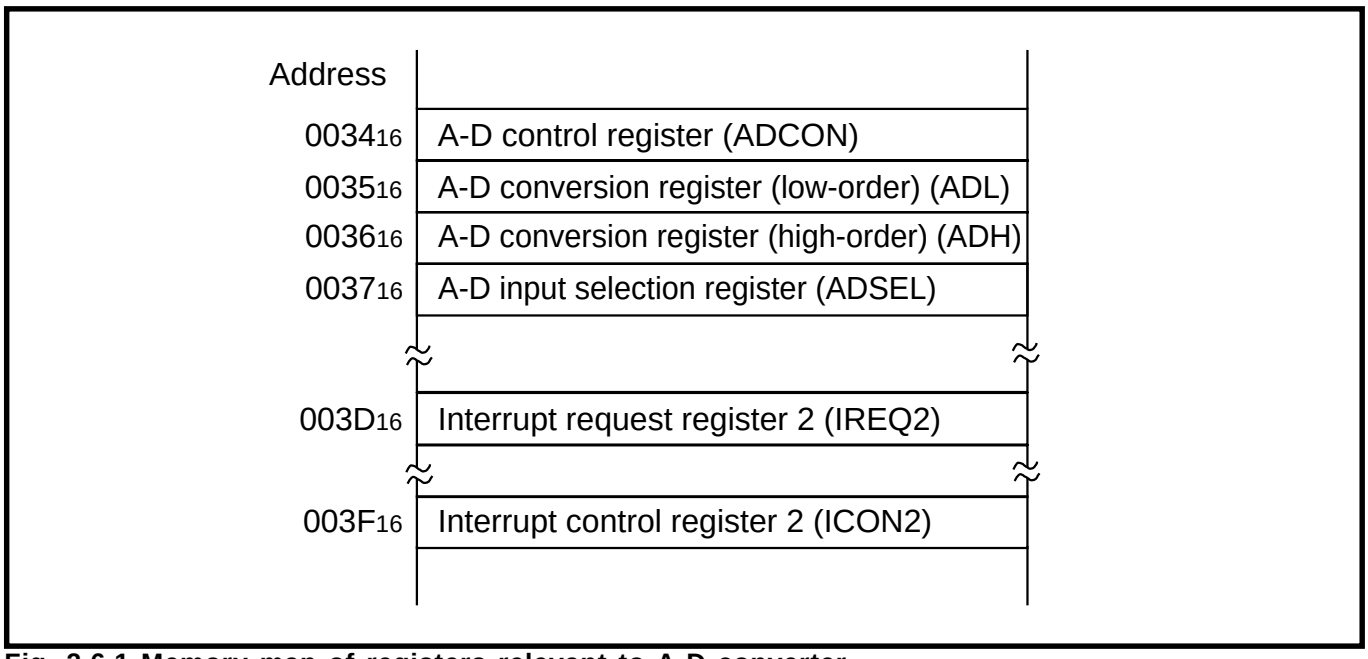


Fig. 2.6.1 Memory map of registers relevant to A-D converter

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 (Note 1) (Note 2) 0 0 0: P3₀/AN₀ or P0₄/AN₅ 0 0 1: P3₁/AN₁ or P0₅/AN₆ 0 1 0: P3₂/AN₂ or P0₆/AN₇ 0 1 1: P3₃/AN₃ or P0₇/AN₈ 1 0 0: P3₄/AN₄ or ————	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	○	×

Notes 1: When bit 0 of the A-D input selection register (address 37₁₆) is set to "0", it is selected here as the analog input pin.
2: When bit 0 of the A-D input selection register (address 371₆) is set to "1", it is selected here as the analog input pin.

Fig. 2.6.2 Structure of A-D control register

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.	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. 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.	Undefined	○	×
1		Undefined	○	×
2		Undefined	○	×
3	b7 8-bit read b0 [b9 b8 b7 b6 b5 b4 b3 b2]	Undefined	○	×
4		Undefined	○	×
5	b7 10-bit read b0 [b7 b6 b5 b4 b3 b2 b1 b0]	Undefined	○	×
6		Undefined	○	×
7		Undefined	○	×

Fig. 2.6.4 Structure of A-D conversion register (low-order)

A-D input selection register

b7 b6 b5 b4 b3 b2 b1 b0

A-D input selection register
(ADSEL: address 37_{16})

b	Name	Functions	At reset	R	W
0	Analog input port selection switch bit	0 : $P3_0/AN_0$-$P3_4/AN_4$ are selected as analog input pins. 1 : $P0_4/AN_5$-$P0_7/AN_8$ are selected as analog input pins.	0	○	○
1	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
2			0	○	×
3			0	○	×
4			0	○	×
5	Fix this bit to "0".		0	○	○
6	Nothing is arranged for this bit. This is write disabled bit. When this bit is read out, the contents are "0".		0	○	×
7	Fix this bit to "0".		0	○	○

Fig. 2.6.5 Structure of A-D input selection register

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. 2.6.6 Structure of Interrupt request register 2

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0
0

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

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. 2.6.7 Structure of Interrupt control register 2

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 2.6.8 shows a connection diagram, and Figure 2.6.9 shows the relevant registers setting.

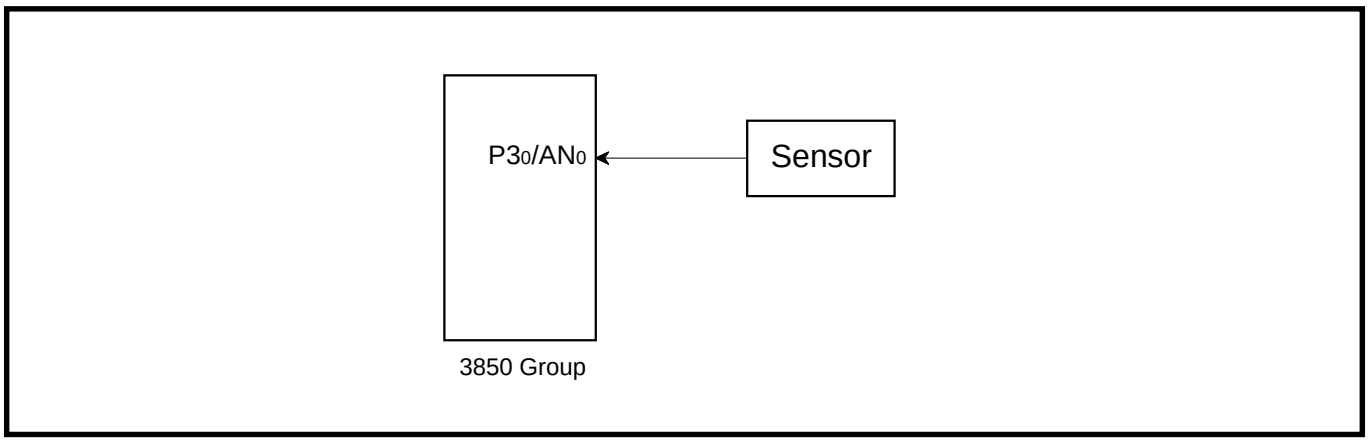


Fig. 2.6.8 Connection diagram

Specifications : •The analog input voltage input from a sensor is converted to digital values.
•P3₀/AN₀ pin is used as an analog input pin.

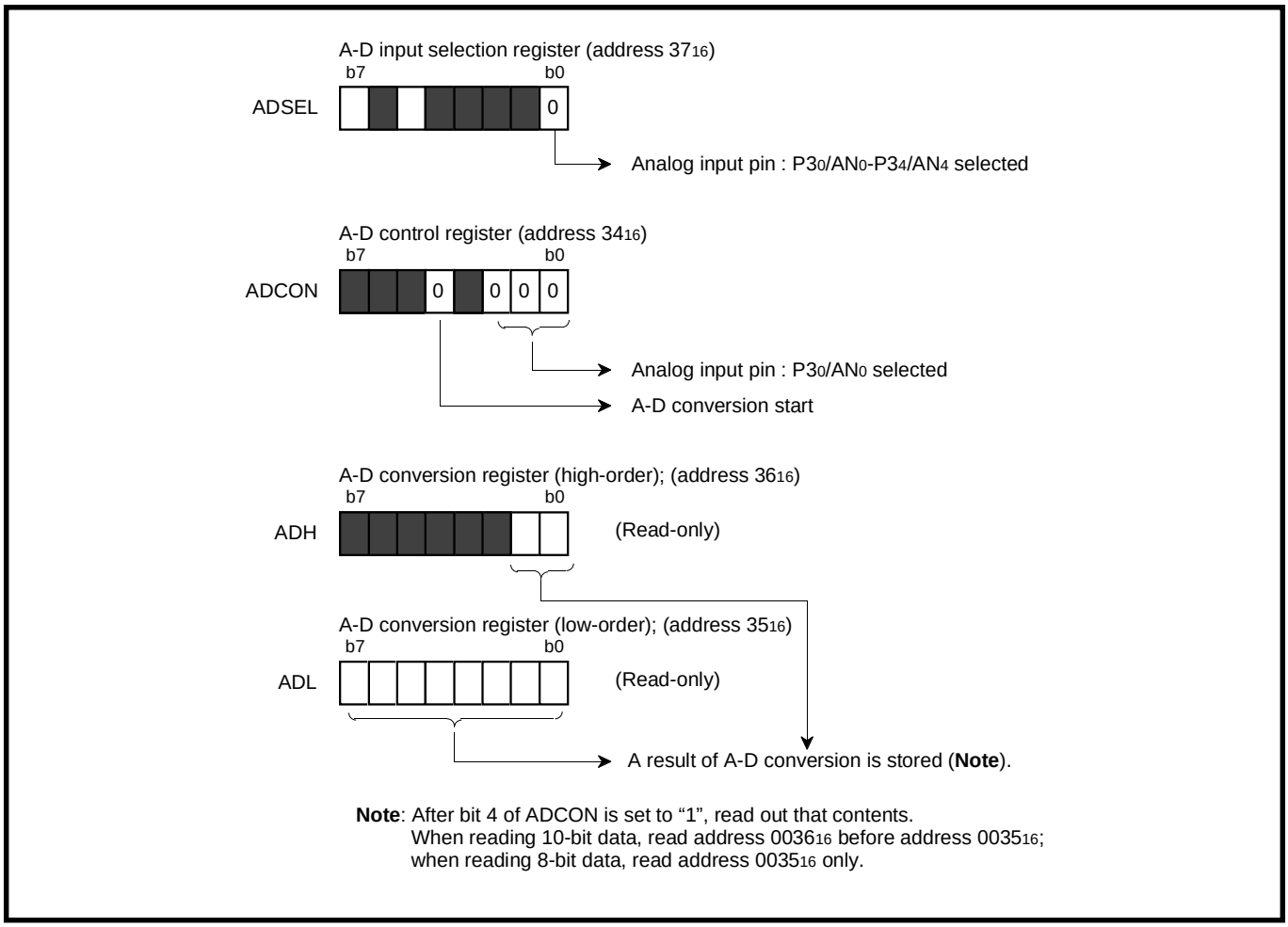


Fig. 2.6.9 Relevant registers setting

An analog input signal from a sensor is converted to the digital value according to the relevant registers setting shown by Figure 2.6.9 Figure 2.6.10 shows the control procedure for 8-bit read, and Figure 2.6.11 shows the control procedure for 10-bit read.

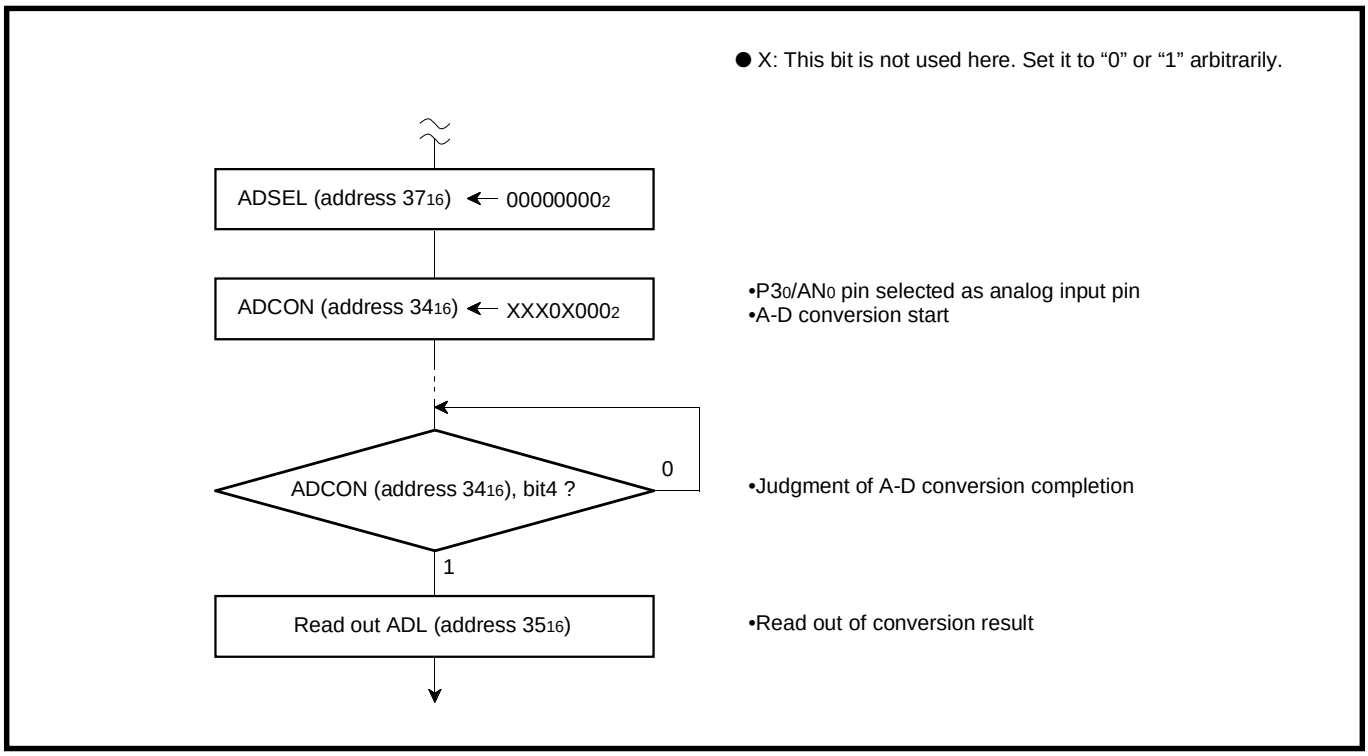


Fig. 2.6.10 Control procedure for 8-bit read

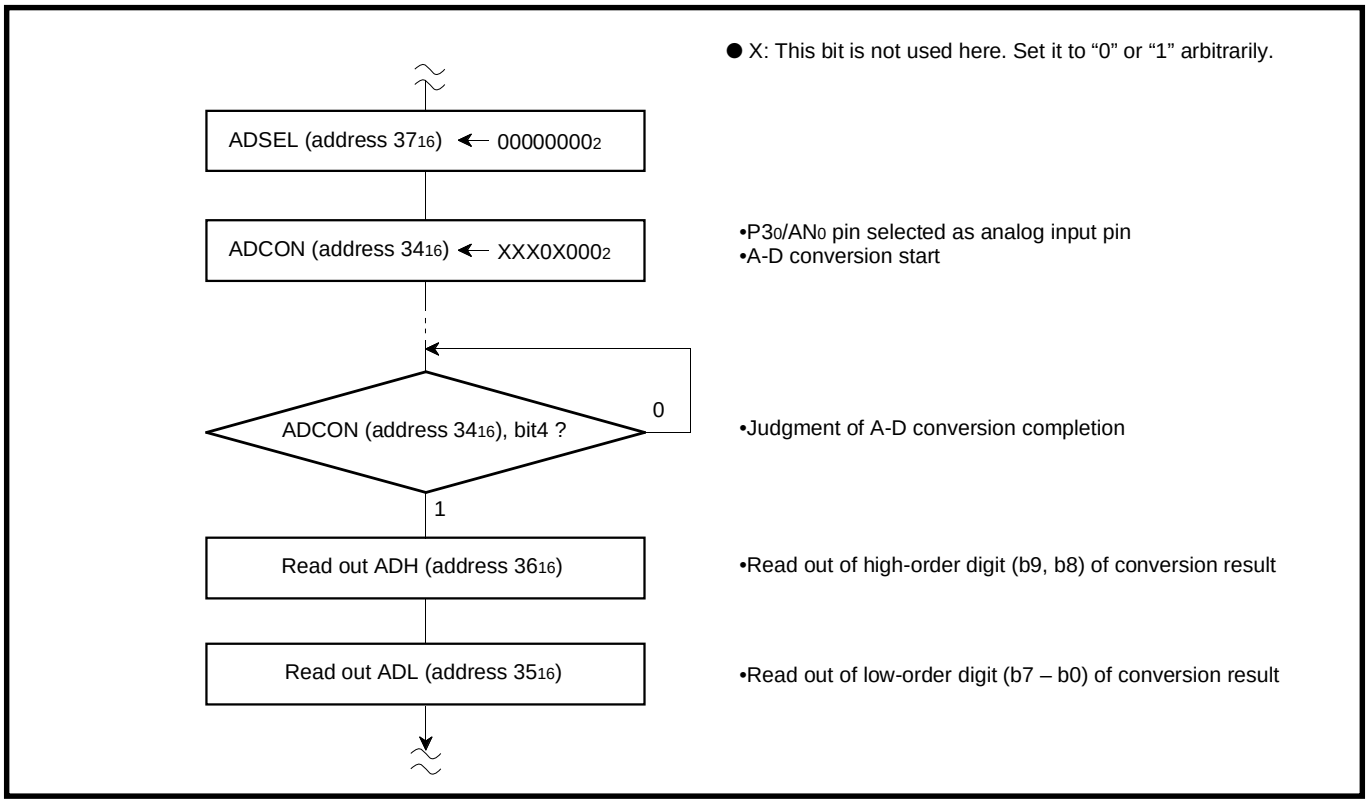


Fig. 2.6.11 Control procedure for 10-bit read

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(XIN) 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(XIN) 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

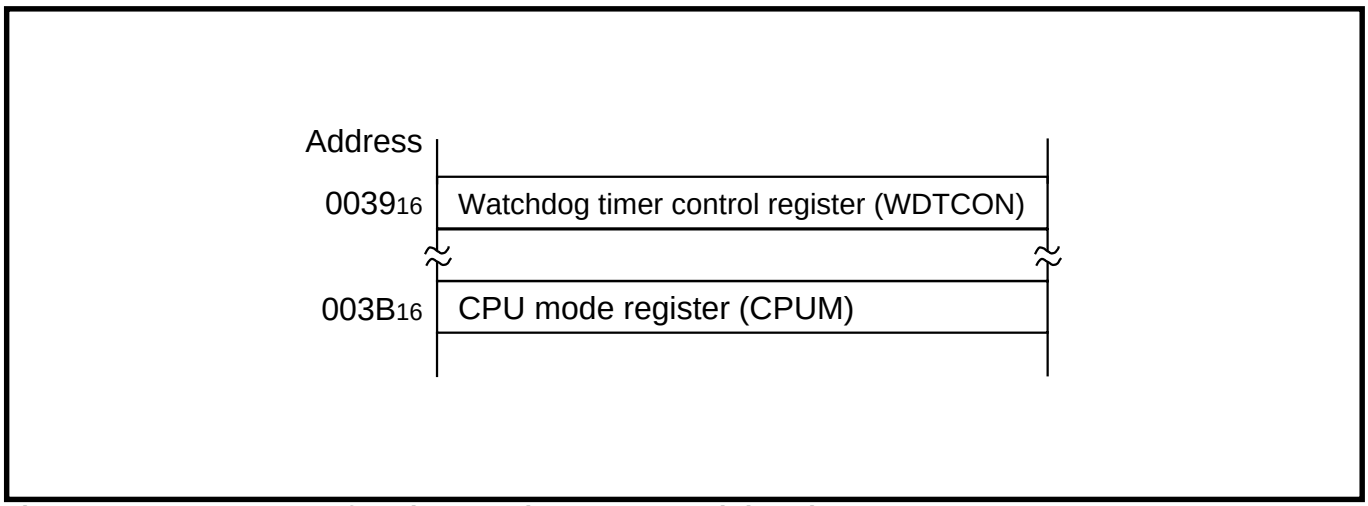


Fig. 2.7.1 Memory map of registers relevant to watchdog timer

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

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. 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.

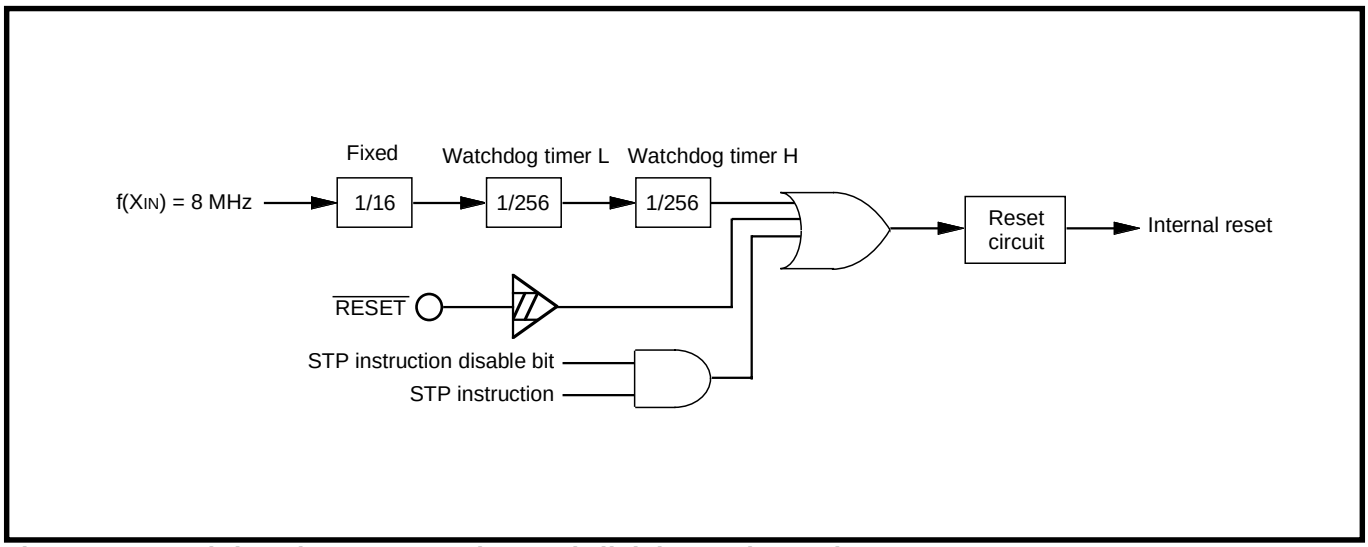


Fig. 2.7.4 Watchdog timer connection and division ratio setting

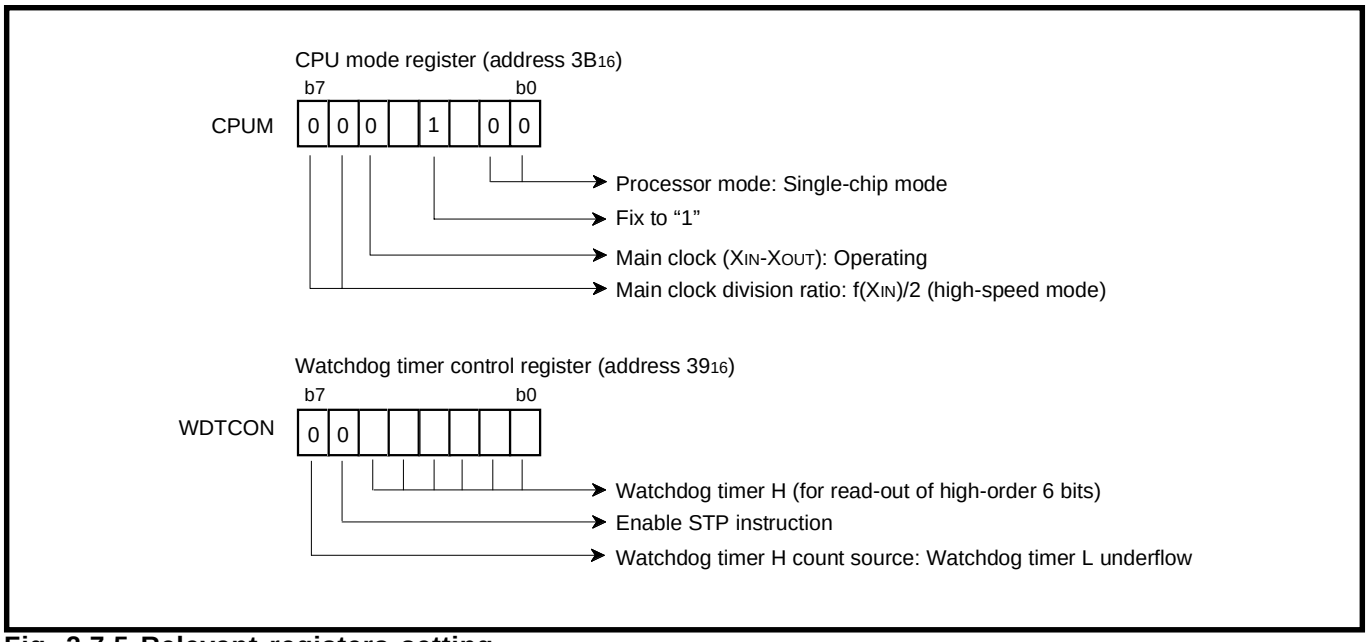


Fig. 2.7.5 Relevant registers setting

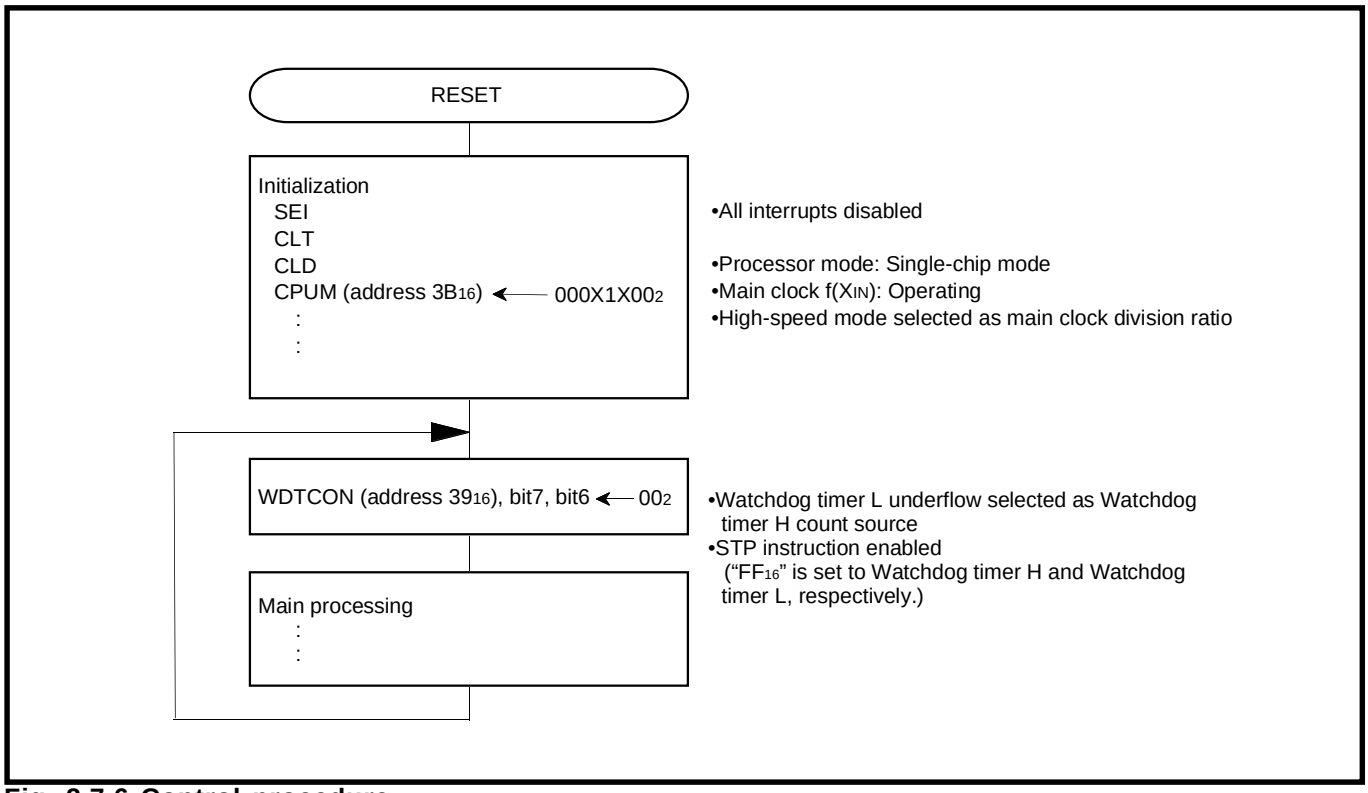


Fig. 2.7.6 Control procedure

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

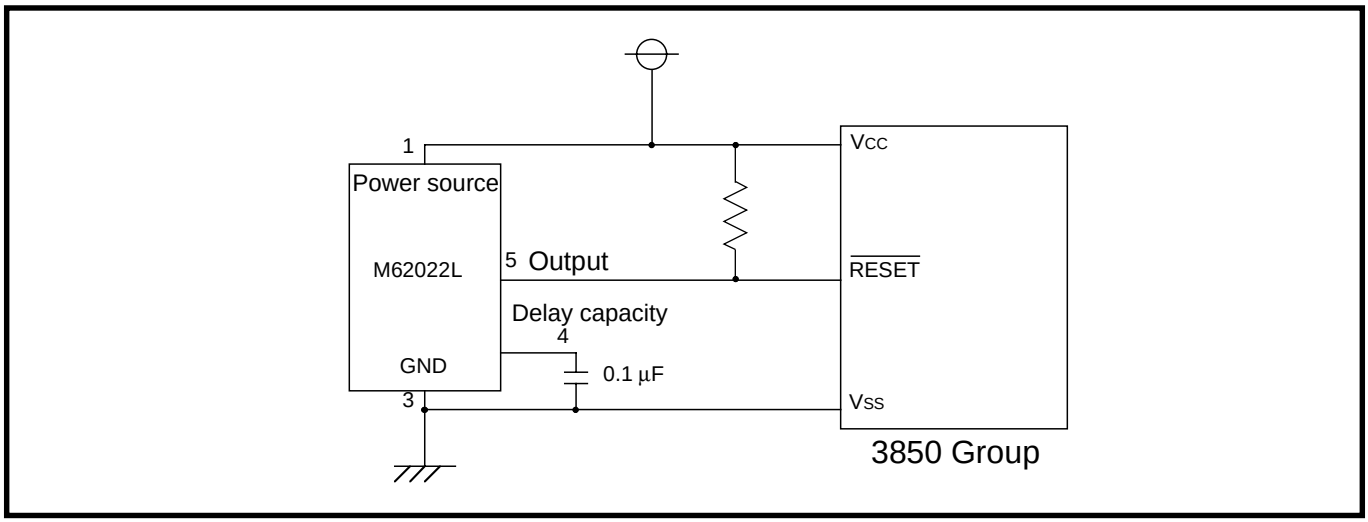


Fig. 2.8.1 Example of poweron reset circuit

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.

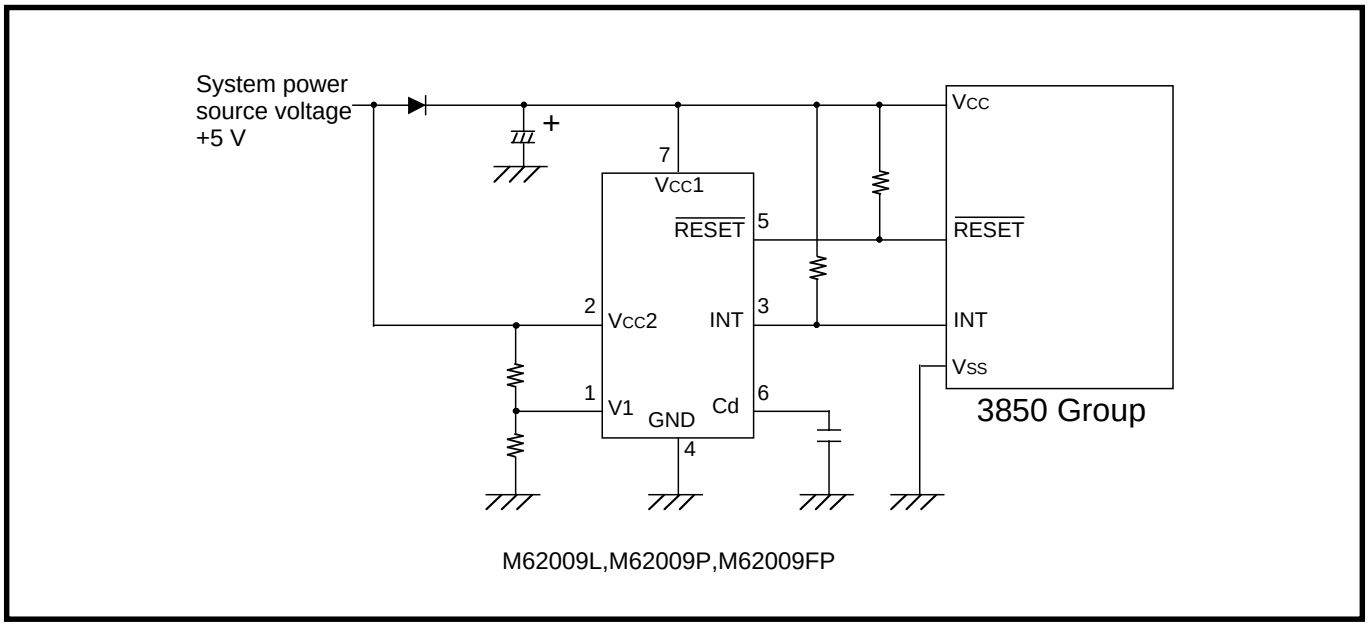


Fig. 2.8.2 RAM backup system

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

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: XCIN-XCOUT oscillation function	0	○	○
5	Main clock (XIN-XOUT) stop bit	0: Oscillating 1: Stopped	0	○	○
6	Main clock division ratio selection bits	b7 b6 0 0: ϕ=f(XIN)/2 (high-speed mode) 0 1: ϕ=f(XIN)/8 (middle-speed mode) 1 0: ϕ=f(XCIN)/2 (low-speed mode) 1 1: not available	1	○	○
7			0	○	○

Fig. 2.9.1 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.

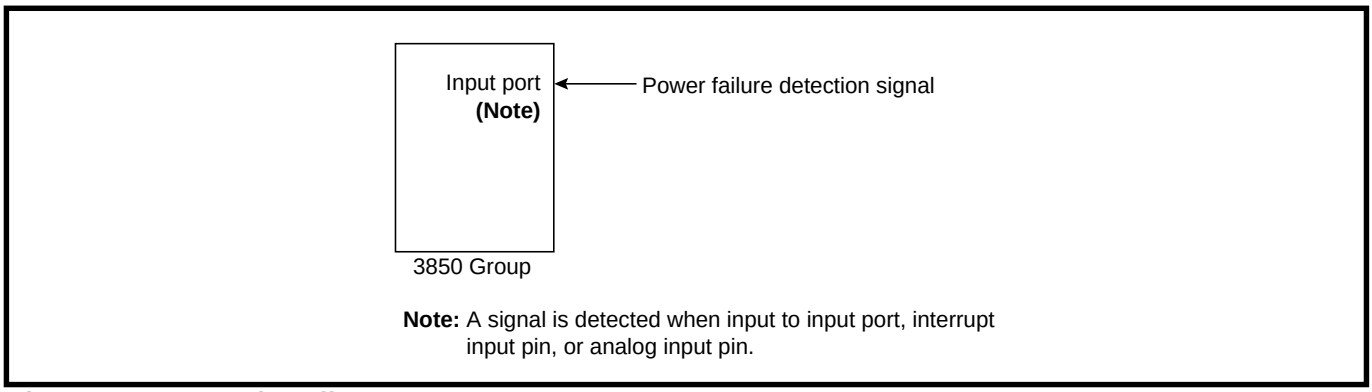


Fig. 2.9.2 Connection diagram

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.

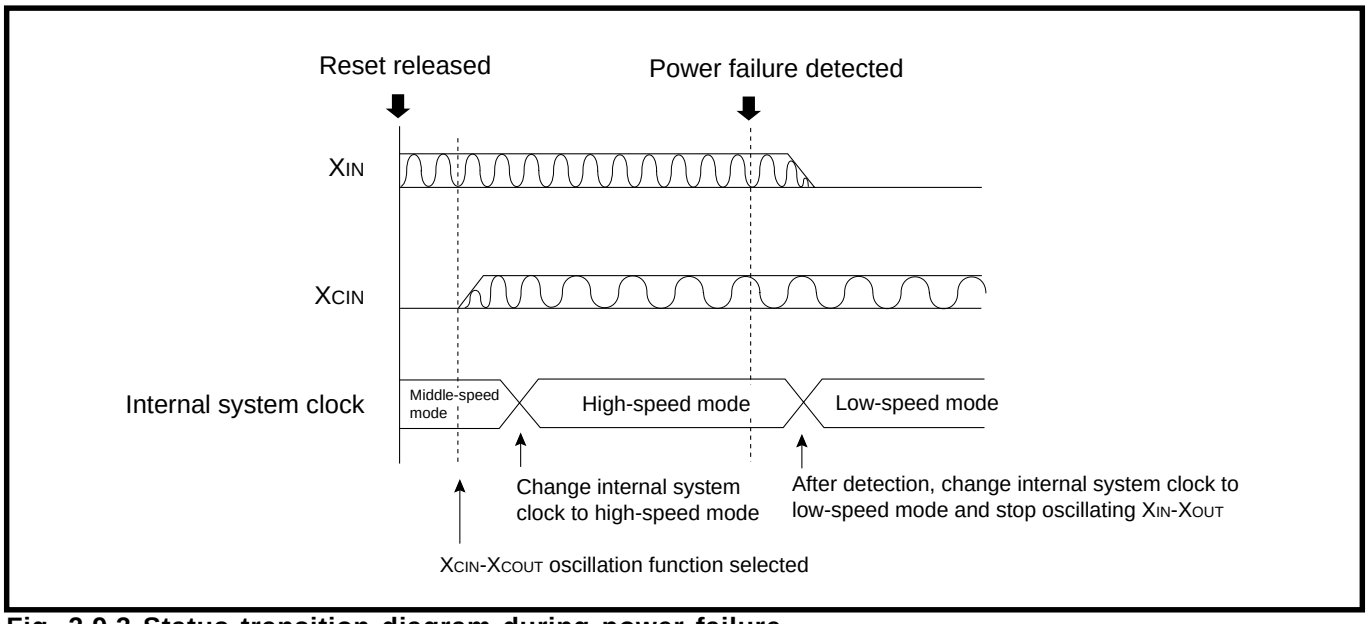
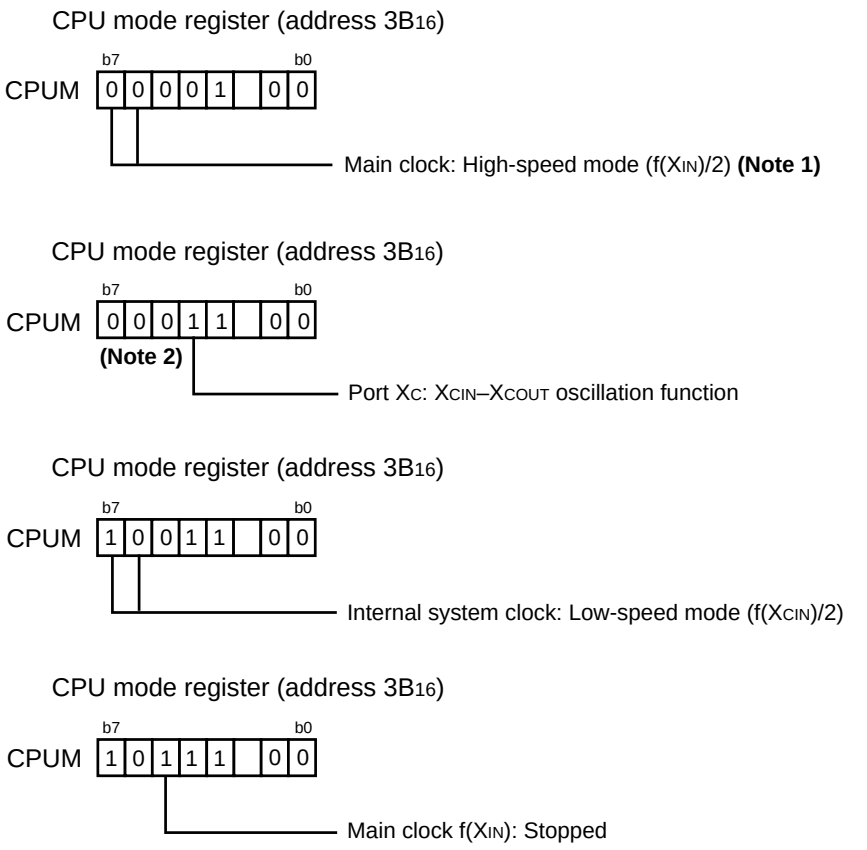


Fig. 2.9.3 Status transition diagram during power failure



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.

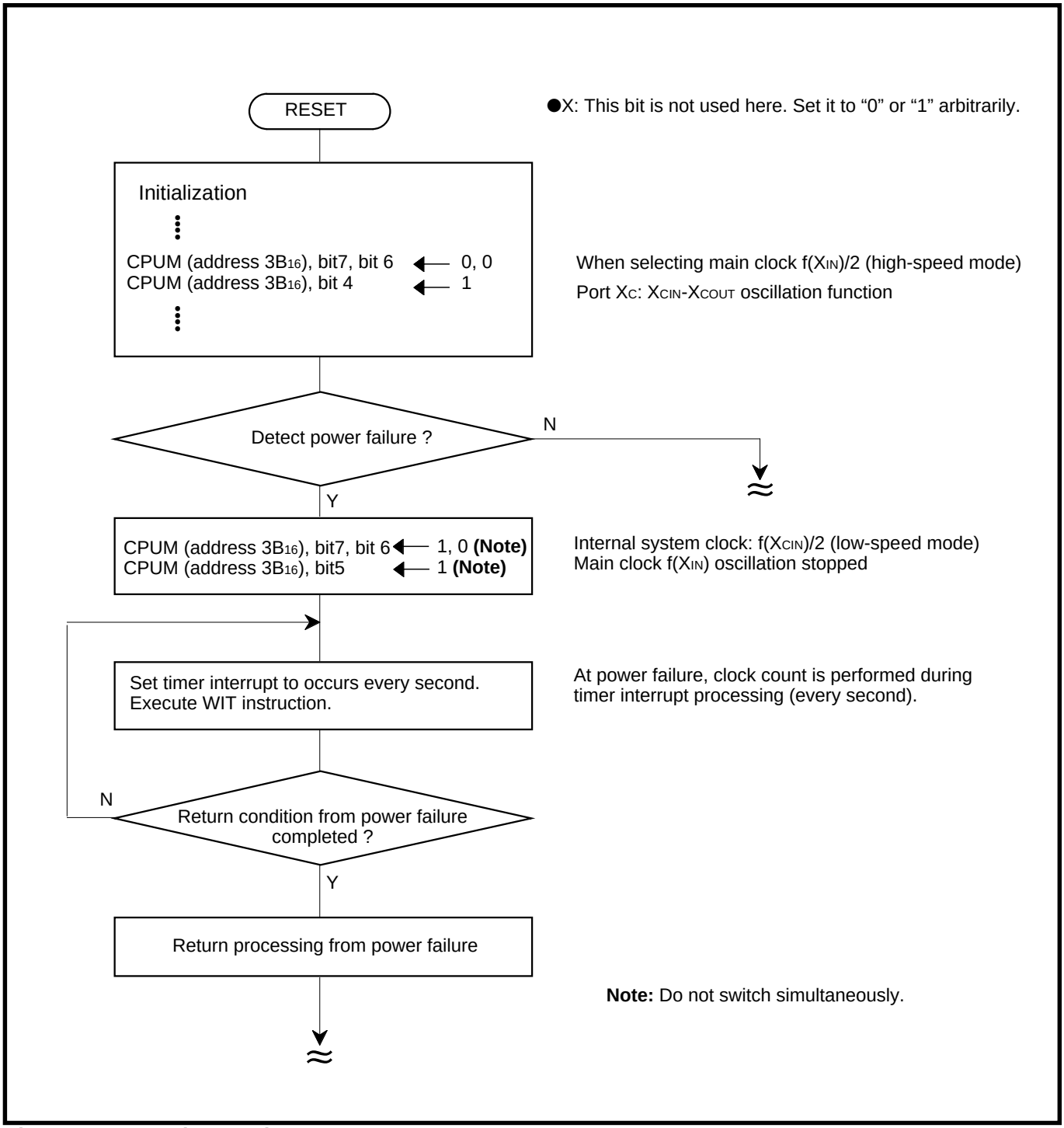


Fig. 2.9.5 Control procedure

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

b7 b6 b5 b4 b3 b2 b1 b0

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

2.10.2 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}).

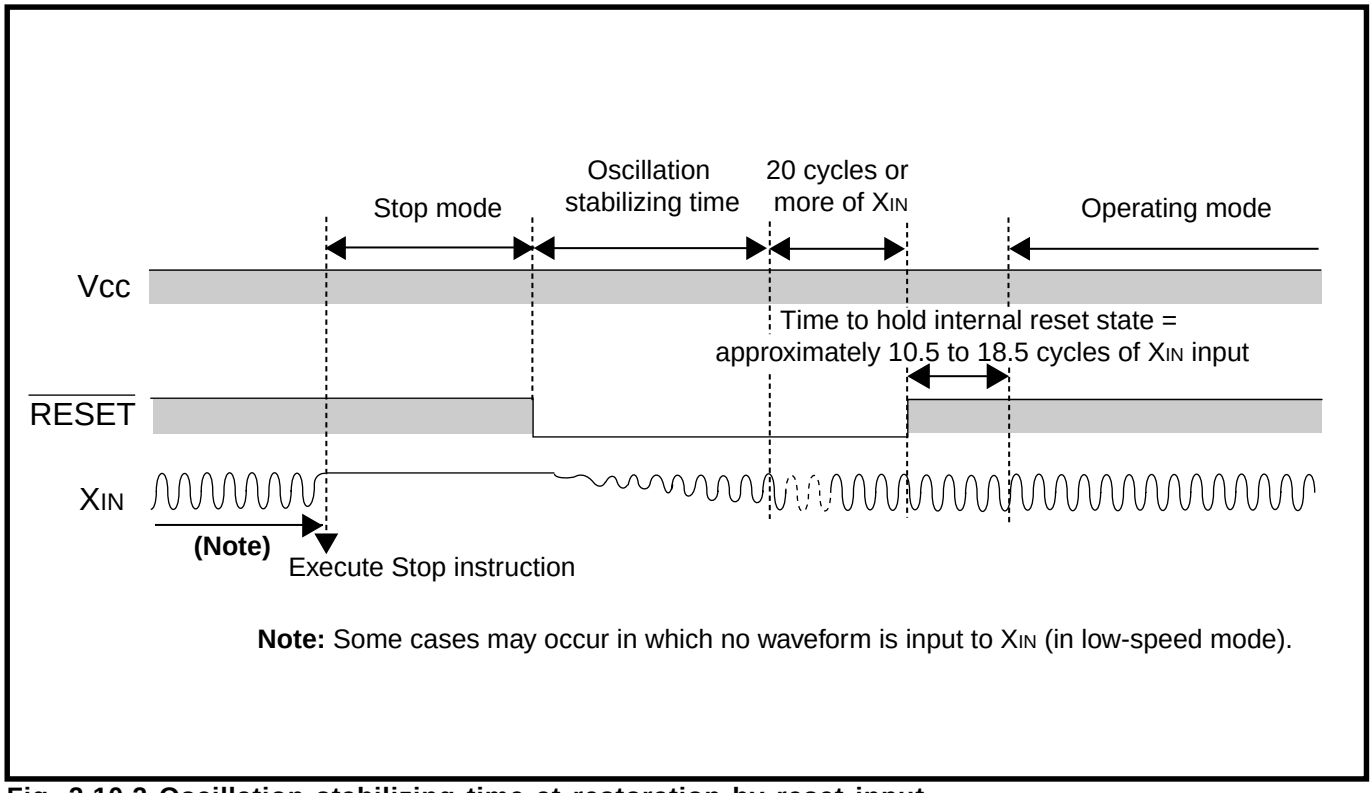


Fig. 2.10.2 Oscillation stabilizing time at restoration by reset input

■Restoration by interrupt request

The occurrence of an interrupt request in the stop mode releases the stop mode. As a result, oscillation is resumed. The interrupts available for restoration are:

•INT_0–INT_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]

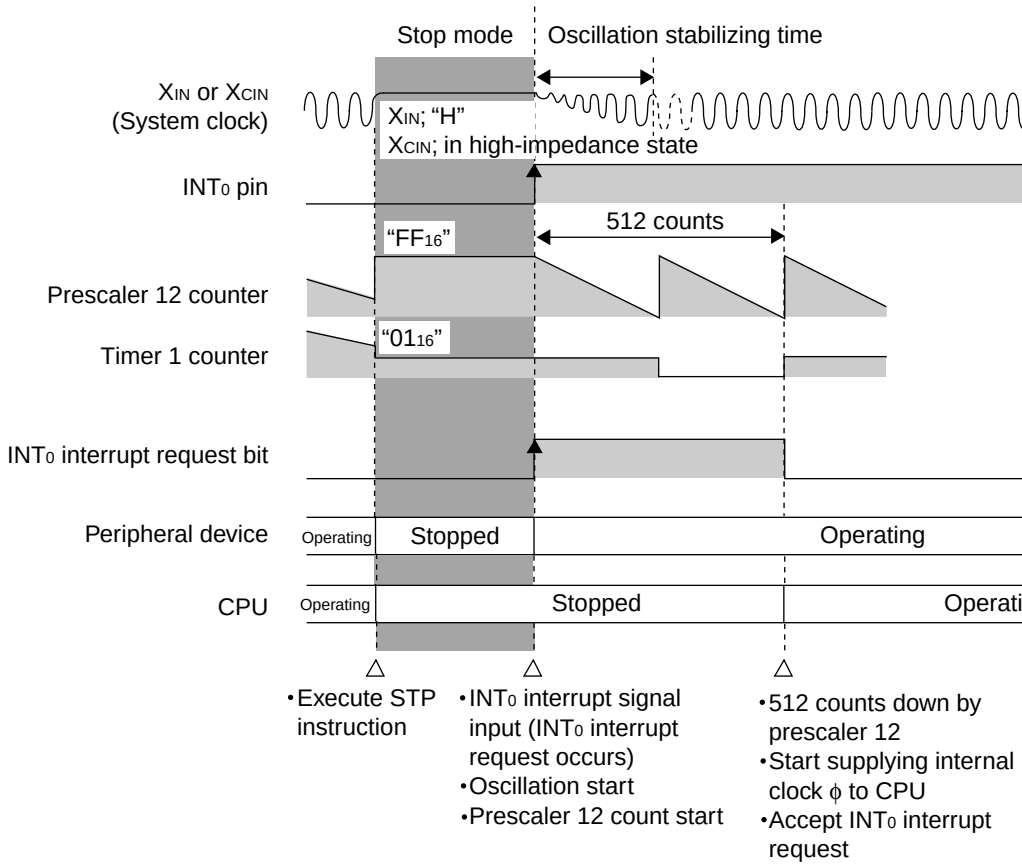
① Interrupt disable flag I = “0” (interrupt enabled)
② Timer 1 interrupt enable bit = “0” (interrupt disabled)
③ Interrupt request bit of interrupt source to be used for restoration = “0” (no interrupt request issued)
④ Interrupt enable bit of interrupt source to be used for restoration = “1” (interrupts 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.



Note: f(XIN)/16 or f(XCIN)/16 is input as the prescaler 12 count source.

Fig. 2.10.3 Execution sequence example at restoration by occurrence of INT0 interrupt request

(3) Notes on using stop mode

■Register setting

Since values of the prescaler 12 and Timer 1 are automatically reloaded when returning from the stop mode, set them again, respectively. (When the oscillation stabilizing time set after STP instruction released bit is "0")

■Clock restoration

After restoration from the stop mode to the normal mode by an interrupt request, the contents of the CPU mode register previous to the STP instruction execution are retained. Accordingly, if both main clock and sub clock were oscillating before execution of the STP instruction, the oscillation of both clocks is resumed at restoration.

In the above case, when the main clock side is set as a system clock, the oscillation stabilizing time for approximately 8,000 cycles of the XIN 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.

2.10.3 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.

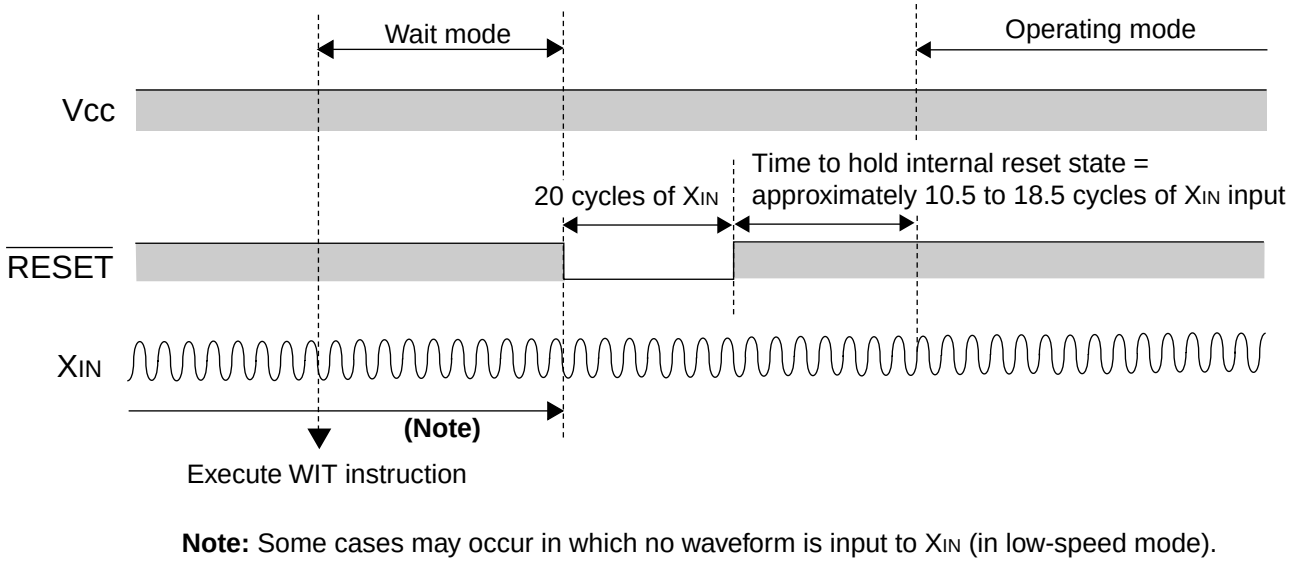
■Restoration by reset input

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.

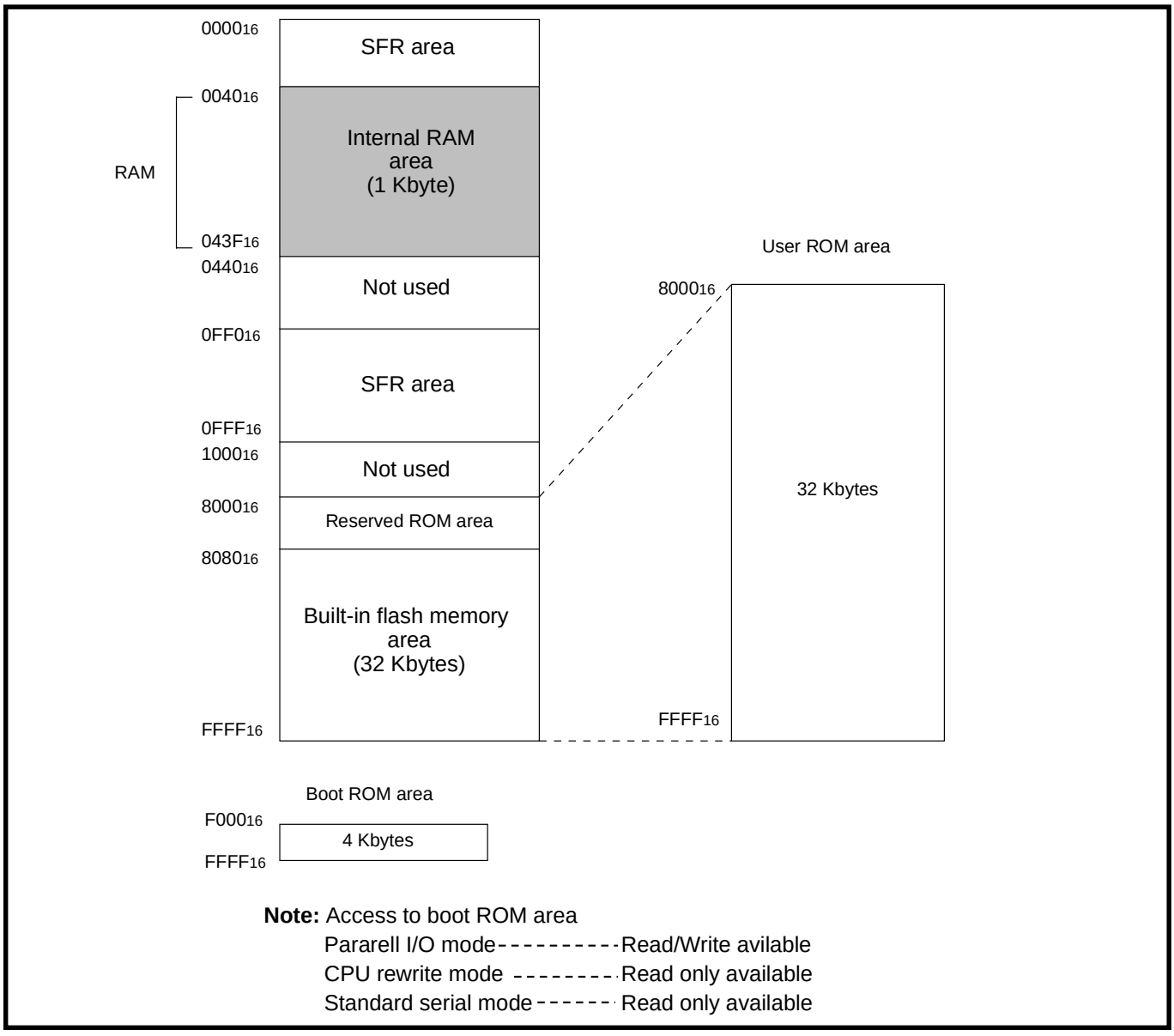


Fig. 2.11.1 Memory map of flash memory version for 3850 Group

2.11.3 Relevant registers

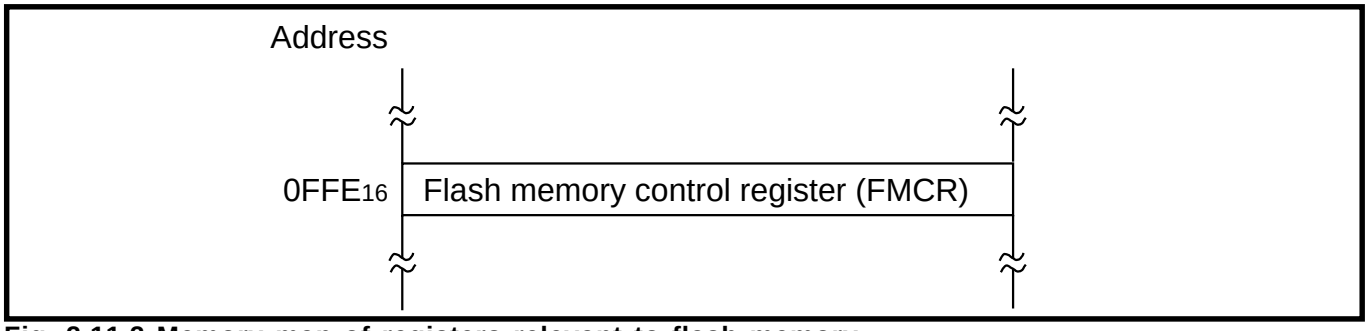


Fig. 2.11.2 Memory map of registers relevant to flash memory

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.

2.11.6 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.

2.11.7 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.

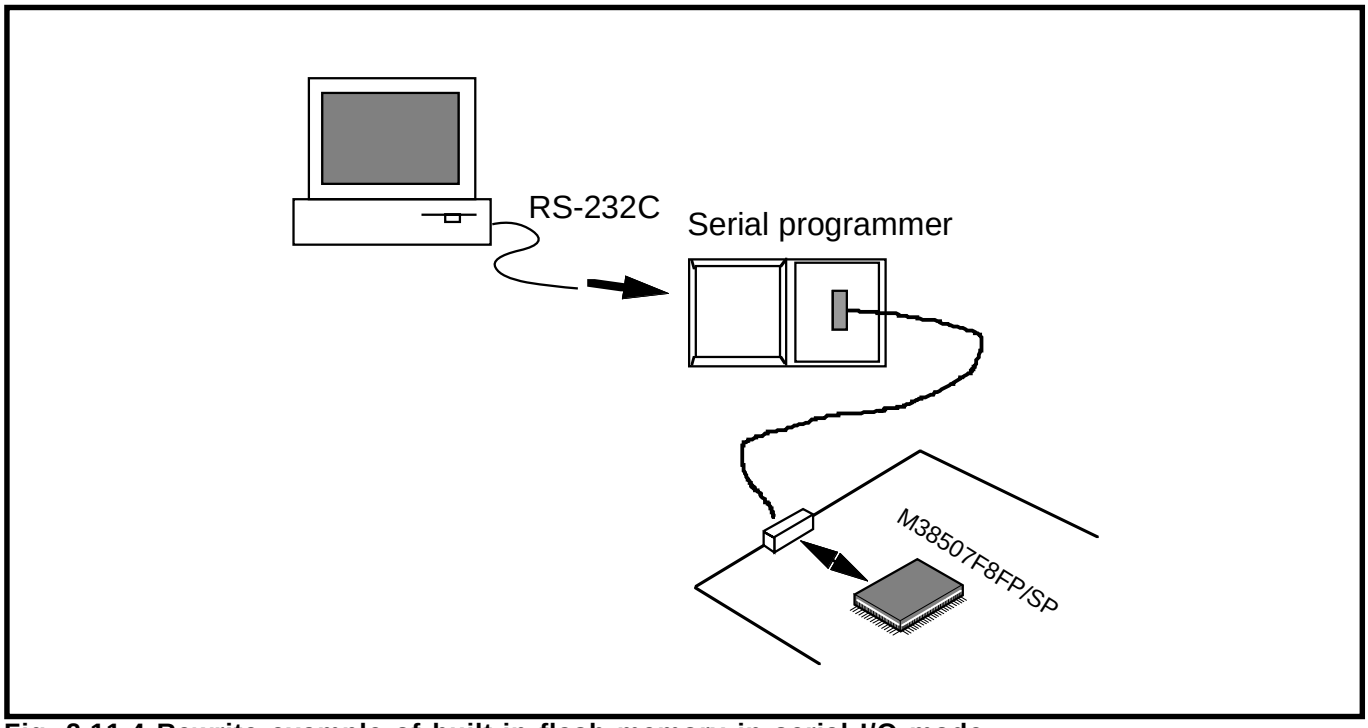


Fig. 2.11.4 Rewrite example of built-in flash memory in serial I/O mode

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.

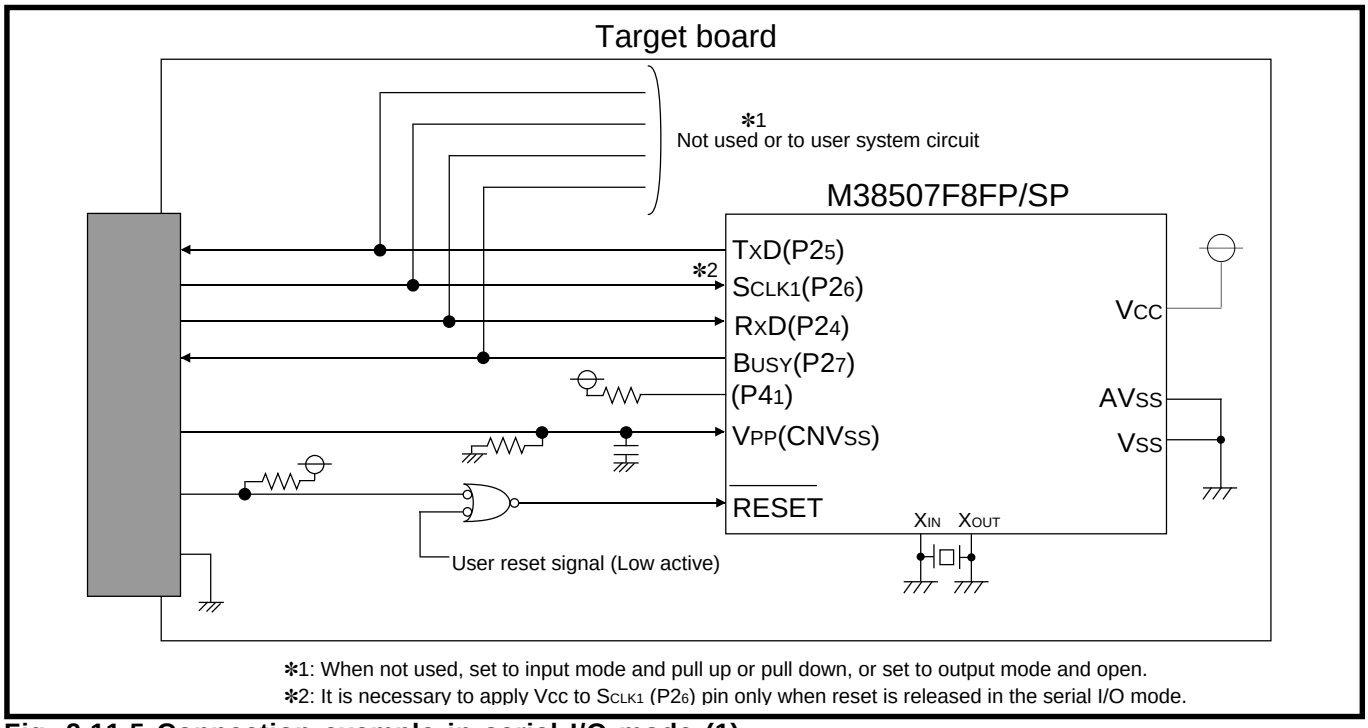


Fig. 2.11.5 Connection example in serial I/O mode (1)

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.

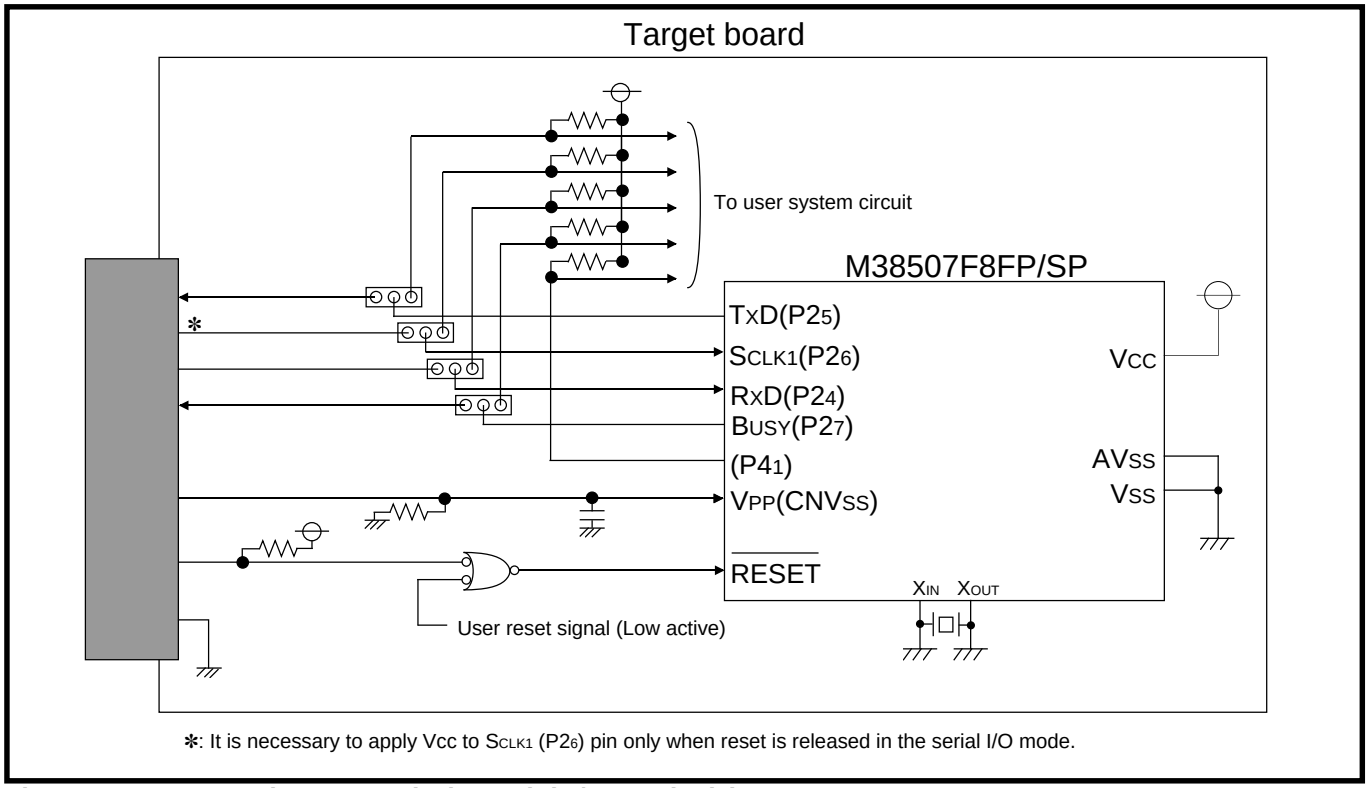
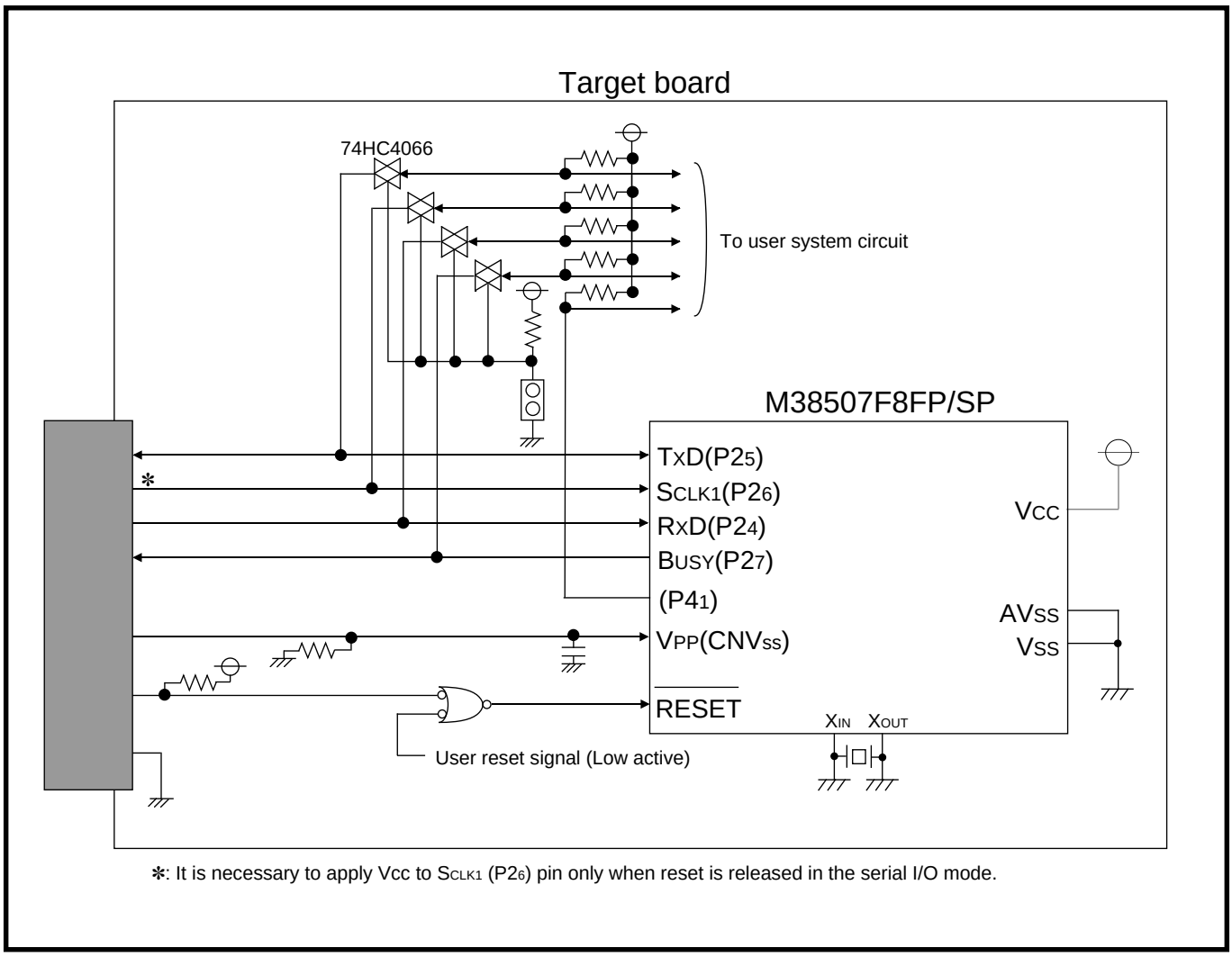


Fig. 2.11.6 Connection example in serial I/O mode (2)

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.



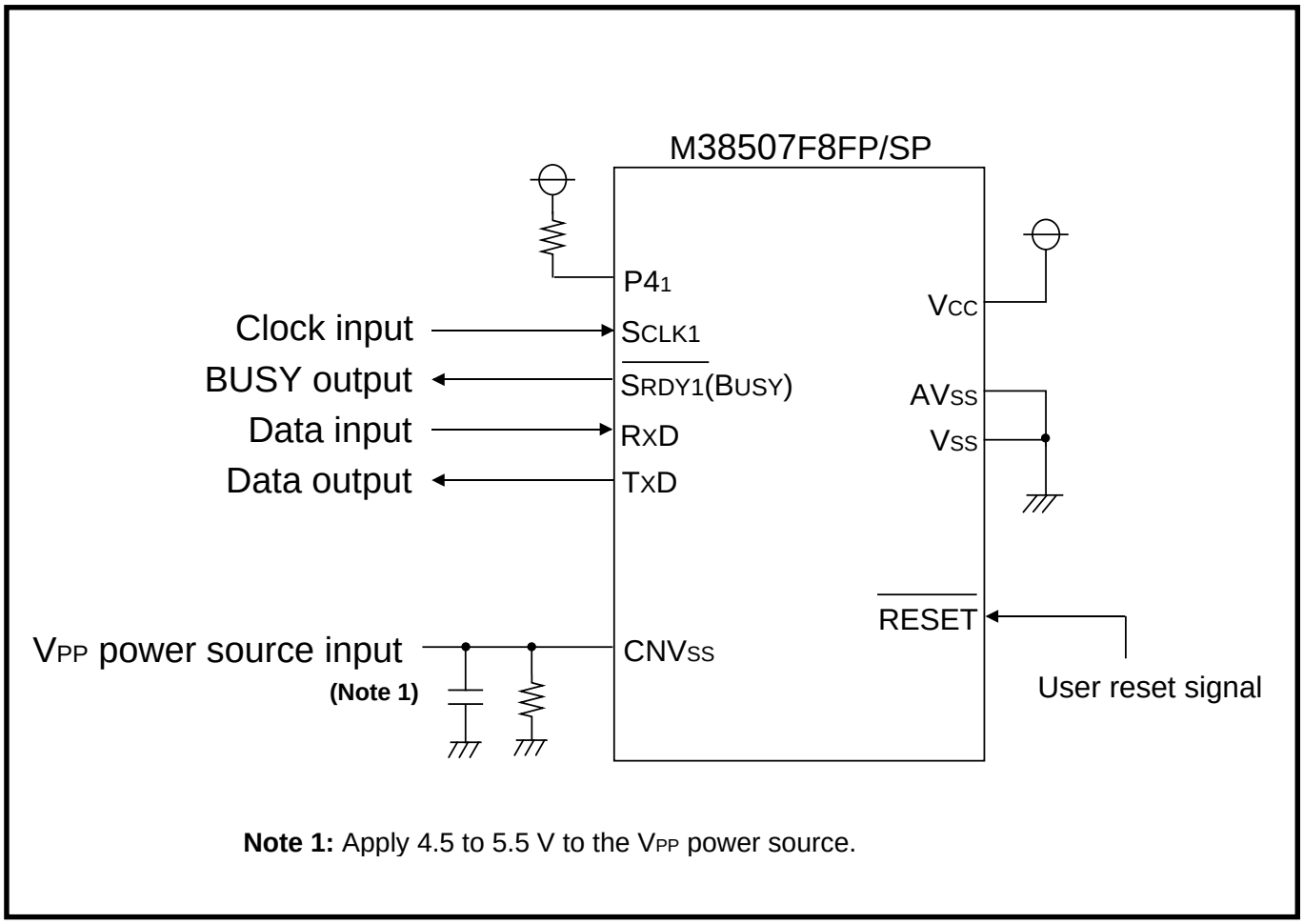
*: It is necessary to apply Vcc to SCLK1 (P26) pin only when reset is released in the serial I/O mode.

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

(2) Control example in CPU rewrite mode

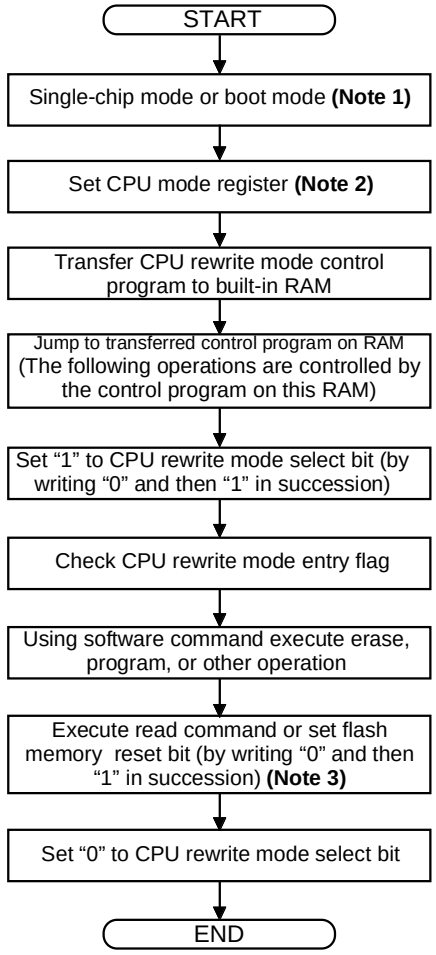
In this example, data is received by using serial I/O, and the data is programmed to the built-in flash memory in the CPU rewrite mode.

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



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.

3. Reference Program Example

Please find the reference program on the Renesas Technology website.
Click the upper left menu of the screen “Application Notes” on the 740 family.

4. Reference

Data Sheet
3850 Group (Spec.A) Data Sheet

Technical News/Technical Update
Before using this material, please visit our website to verify that this is the most updated document available.

5. Website and Support

Renesas Technology Corporation Semiconductor Home Page
http://www.renesas.com

E-mail Support
E-mail: csc@renesas.com

REVISION HISTORY	3850 Group (Spec.A) Peripheral Function Application

Rev.	Date	Description	
		Page	Summary
1.00	Nov 14, 2005	-	This application note is issued using the information of "Chapter 2 APPLICATION" in the 3850 Group (Spec.A) User's Manual Rev.1.00.
		30	Fig.2.3.14 Control procedure is partly revised
		53	Fig.2.4.20 Registers setting relevant to transmitting side is partly revised
		55	Fig.2.4.22 Control procedure of transmitting side is partly revised
		59	Fig.2.4.28 Control procedure of Serial I/O1 is partly revised
		61	Fig.2.4.31 Control procedure of Serial I/O2 is partly revised
		62	Specifications, Limitations of specifications are revised
		63	Fig.2.4.33 Timing chart is revised Fig.2.4.34 Relevant registers setting is partly revised
		64	●Control in the master unit is revised
		65	●Control in the slave unit is revised Fig.2.4.37 Control procedure of slave unit is revised
		70	Fig.2.4.41 Control procedure of transmitting side is partly revised
		71	Fig.2.4.42 Control procedure of receiving side is partly revised
		89	Fig.2.7.6 Control procedure is partly revised

Keep safety first in your circuit designs!

1. 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

1. 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.
2. 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.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology 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).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. 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.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.