

MITSUBISHI 8-BIT SINGLE-CHIP MICROCOMPUTER  
740 FAMILY / 38000 SERIES

3800  
Group

User's Manual

Keep safety first in your circuit designs!

- Mitsubishi Electric 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 non-flammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor 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 Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric 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 or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams and charts, represent information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.
- Mitsubishi Electric 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 Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.  
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

## Preface

This user's manual describes Mitsubishi's CMOS 8-bit microcomputers 3800 Group.

After reading this manual, the user should have a thorough knowledge of the functions and features of the 3800 Group, and should be able to fully utilize the product. The manual starts with specifications and ends with application examples.

For details of software, refer to the "SERIES MELPS 740 <SOFTWARE> USER'S MANUAL."

For details of development support tools, refer to the "DEVELOPMENT SUPPORT TOOLS FOR MICRO-COMPUTERS" data book.

# BEFORE USING THIS USER'S MANUAL

This user's manual consists of the following three chapters. Refer to the chapter appropriate to your conditions, such as hardware design or software development. Chapter 3 also includes necessary information for systems development. Be sure to refer to this chapter.

## 1. Organization

### ● CHAPTER 1 HARDWARE

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

### ● CHAPTER 2 APPLICATION

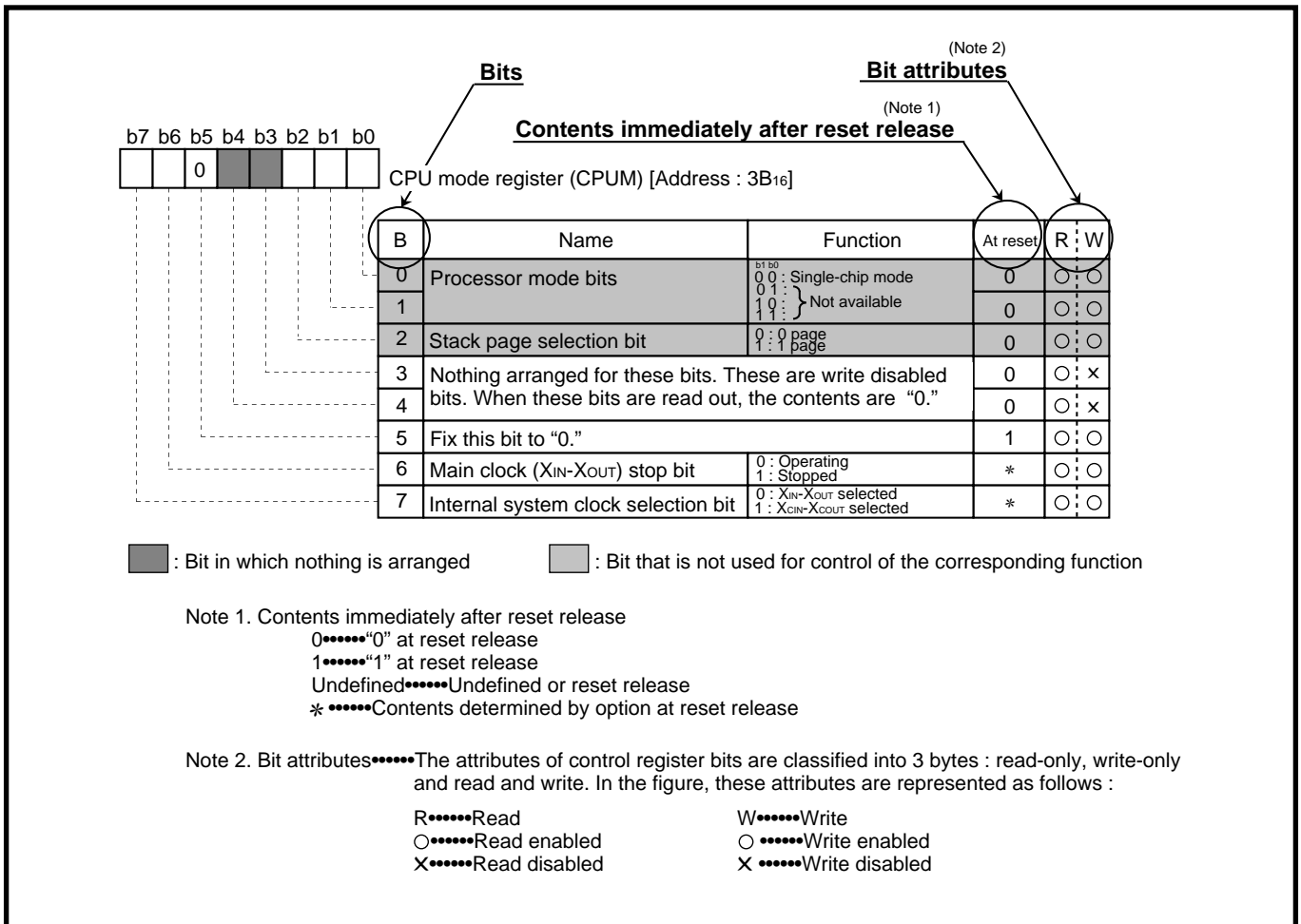
This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of related registers.

### ● CHAPTER 3 APPENDIX

This chapter includes necessary information for systems development using the microcomputer, electric characteristics, a list of registers, the masking confirmation (mask ROM version), and mark specifications which are to be submitted when ordering.

## 2. Structure of register

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



## LIST OF GROUPS HAVING THE SIMILAR FUNCTIONS

3800 group, one of the CMOS 8-bit microcomputer 38000 series presented in this user's manual is provided with standard functions.

The basic functions of the 3800, 3802, 3806 and 3807 groups having the same functions are shown below. For the detailed functions of each group, refer to the related data book and user's manual.

### List of groups having the same functions

As of September 1995

Group		3800 group					3802 group				3806 group				3807 group	
Function																
Pin (Package type)		64 pin • 64P4B • 64P6N-A • 64P6D-A					64 pin • 64P4B • 64P6N-A				80 pin • 80P6N-A • 80P6S-A • 80P6D-A				80 pin • 80P6N-A	
Clock generating circuit		1 circuit					1 circuit				1 circuit				2 circuit	
Timer		<8-bit> Prescaler : 3 Timer : 4					<8-bit> Prescaler : 3 Timer : 4				<8-bit> Prescaler : 3 Timer : 4				<8-bit> Timer : 3 <16-bit> Timer X/Y : 2 Timer A/B : 2	
Serial I/O		UART or Clock synchronous X 1					UART or Clock synchronous X 1				UART or Clock synchronous X 1				UART or Clock synchronous X 1	
		—					Clock synchronous X 1				Clock synchronous X 1				Clock synchronous X 1	
A-D converter		—					8-bit X 8-channel				8-bit X 8-channel				8-bit X 13-channel	
D-A converter		—					8-bit X 2-channel				8-bit X 2-channel				8-bit X 4-channel	
Memory type	Mask ROM	8K <small>(Note 1)</small>	16K <small>(Note 1)</small>	24K	32K <small>(Note 1)</small>	*	8K <small>(Note 1)</small>	16K <small>(Note 1)</small>	24K	32K <small>(Note 1)</small>	12K <small>(Note 1)</small>	16K <small>(Note 1)</small>	24K <small>(Note 3)</small>	32K <small>(Note 3)</small>	48K <small>(Note 3)</small>	16K
	One Time PROM	8K <small>(Note 1)</small>	16K <small>(Note 1)</small>	—	32K	—	—	—	—	32K <small>(Note 1)</small>	—	—	24K <small>(Note 2)</small>	—	48K <small>(Note 3)</small>	16K
	EPROM	—	16K	—	32K	—	—	—	—	32K	—	—	24K	—	48K <small>(Note 2)</small>	16K
	RAM	384	384	512	640	384	384	384	640	1024	384	384	512	1024	1024	512
Remarks							PWM output								Real time port output Analog comparator Watchdog timer	

**Notes 1:** Extended operating temperature version available

**2:** High-speed version available

**3:** Extended operating temperature version and High-speed version available

**\***. ROM expansion

# Table of contents

## CHAPTER 1. HARDWARE

<b>DESCRIPTION</b> .....	<b>1-2</b>
<b>FEATURES</b> .....	<b>1-2</b>
<b>APPLICATIONS</b> .....	<b>1-2</b>
<b>PIN CONFIGURATION</b> .....	<b>1-2</b>
<b>FUNCTIONAL BLOCK</b> .....	<b>1-4</b>
<b>PIN DESCRIPTION</b> .....	<b>1-5</b>
<b>PART NUMBERING</b> .....	<b>1-6</b>
<b>GROUP EXPANSION</b> .....	<b>1-7</b>
<b>GROUP EXPANSION (EXTENDED OPERATING TEMPERATURE VERSION)</b> .....	<b>1-9</b>
<b>FUNCTIONAL DESCRIPTION</b> .....	<b>1-10</b>
Central Processing Unit (CPU) .....	1-10
Memory .....	1-14
I/O Ports .....	1-16
Interrupts .....	1-18
Timers .....	1-20
Serial I/O .....	1-22
Reset Circuit .....	1-26
Clock Generating Circuit .....	1-28
Processor Modes .....	1-29
<b>NOTES ON PROGRAMMING</b> .....	<b>1-31</b>
Processor Status Register .....	1-31
Interrupts .....	1-31
Decimal Calculations.....	1-31
Timers .....	1-31
Multiplication and Division Instructions .....	1-31
Ports .....	1-31
Serial I/O .....	1-31
Instruction Execution Time .....	1-31
Memory Expansion Mode and Microprocessor Mode .....	1-31
<b>DATA REQUIRED FOR MASK ORDERS</b> .....	<b>1-32</b>
<b>ROM PROGRAMMING METHOD</b> .....	<b>1-32</b>
<b>FUNCTIONAL DESCRIPTION SUPPLEMENT</b> .....	<b>1-33</b>
Interrupt .....	1-33
Timing After Interrupt.....	1-34

## Table of contents

---

### CHAPTER 2. APPLICATION

<b>2.1 I/O port</b> .....	<b>2-2</b>
2.1.1 Memory map of I/O port .....	2-2
2.1.2 Related registers .....	2-3
2.1.3 Handling of unused pins .....	2-4
<b>2.2 Timer</b> .....	<b>2-5</b>
2.2.1 Memory map of timer .....	2-5
2.2.2 Related registers .....	2-6
2.2.3 Timer application examples .....	2-11
<b>2.3 Serial I/O</b> .....	<b>2-23</b>
2.3.1 Memory map of serial I/O .....	2-23
2.3.2 Related registers .....	2-24
2.3.3 Serial I/O connection examples .....	2-28
2.3.4 Setting of serial I/O transfer data format .....	2-30
2.3.5 Serial I/O application examples .....	2-31
<b>2.4 Processor mode</b> .....	<b>2-49</b>
2.4.1 Memory map of processor mode .....	2-49
2.4.2 Related register .....	2-49
2.4.3 Processor mode application examples .....	2-50
<b>2.5 Reset</b> .....	<b>2-54</b>
2.5.1 Connection example of reset IC .....	2-54

### CHAPTER 3. APPENDIX

<b>3.1 Electrical characteristics</b> .....	<b>3-2</b>
3.1.1 Absolute maximum ratings .....	3-2
3.1.2 Recommended operating conditions .....	3-2
3.1.3 Electrical characteristics .....	3-3
3.1.4 Timing requirements and Switching characteristics .....	3-4
3.1.5 Absolute maximum ratings (Extended operating temperature version) .....	3-8
3.1.6 Recommended operating conditions(Extended operating temperature version)....	3-8
3.1.7 Electrical characteristics (Extended operating temperature version) .....	3-9
3.1.8 Timing requirements and Switching characteristics (Extended operating temperature version).....	3-10
3.1.9 Timing diagram .....	3-12
<b>3.2 Standard characteristics</b> .....	<b>3-15</b>
3.2.1 Power source current characteristic examples .....	3-15
3.2.2 Port standard characteristic examples .....	3-16
<b>3.3 Notes on use</b> .....	<b>3-18</b>
3.3.1 Notes on interrupts .....	3-18
3.3.2 Notes on the serial I/O .....	3-18
3.3.3 Notes on the $\overline{\text{RESET}}$ pin .....	3-19
3.3.4 Notes on input and output pins .....	3-20

3.3.5 Notes on memory expansion mode and microprocessor mode .....	3-21
3.3.6 Notes on built-in PROM .....	3-22
<b>3.4 Countermeasures against noise .....</b>	<b>3-24</b>
3.4.1 Shortest wiring length .....	3-24
3.4.2 Connection of a bypass capacitor across the Vss line and the Vcc line .....	3-25
3.4.3 Consideration for oscillator .....	3-26
3.4.4 Setup for I/O ports .....	3-26
3.4.5 Providing of watchdog timer function by software .....	3-27
<b>3.5 List of registers .....</b>	<b>3-28</b>
<b>3.6 Mask ROM ordering method .....</b>	<b>3-37</b>
<b>3.7 Mark specification form .....</b>	<b>3-51</b>
<b>3.8 Package outline.....</b>	<b>3-53</b>
<b>3.9 Machine Instructions .....</b>	<b>3-56</b>
<b>3.10 List of instruction codes .....</b>	<b>3-66</b>
<b>3.11 SFR memory map.....</b>	<b>3-67</b>
<b>3.12 Pin configuration .....</b>	<b>3-68</b>



# List of figures

## CHAPTER 1 HARDWARE

Fig. 1 Pin configuration of M38002M4-XXXFP/M38003M6-XXXHP .....	1-2
Fig. 2 Pin configuration of M38002M4-XXXSP .....	1-3
Fig. 3 Functional block diagram .....	1-4
Fig. 4 Part numbering .....	1-6
Fig. 5 Memory expansion plan .....	1-7
Fig. 6 Memory expansion plan (Extended operating temperature version) .....	1-9
Fig. 7 740 Family CPU register structure .....	1-10
Fig. 8 Register push and pop at interrupt generation and subroutine call .....	1-11
Fig. 9 Structure of CPU mode register .....	1-13
Fig. 10 Memory map diagram .....	1-14
Fig. 11 Memory map of special function register (SFR) .....	1-15
Fig. 12 Port block diagram (single-chip mode) .....	1-17
Fig. 13 Interrupt control .....	1-19
Fig. 14 Structure of interrupt-related registers .....	1-19
Fig. 15 Structure of timer XY register .....	1-20
Fig. 16 Block diagram of timer X, timer Y, timer 1, and timer 2 .....	1-21
Fig. 17 Block diagram of clock synchronous serial I/O .....	1-22
Fig. 18 Operation of clock synchronous serial I/O function .....	1-22
Fig. 19 Block diagram of UART serial I/O .....	1-23
Fig. 20 Operation of UART serial I/O function .....	1-24
Fig. 21 Structure of serial I/O control registers .....	1-25
Fig. 22 Example of reset circuit .....	1-26
Fig. 23 Internal status of microcomputer after reset .....	1-26
Fig. 24 Timing of reset .....	1-27
Fig. 25 Ceramic resonator circuit .....	1-28
Fig. 26 External clock input circuit .....	1-28
Fig. 27 Block diagram of clock generating circuit .....	1-28
Fig. 28 Memory maps in various processor modes .....	1-29
Fig. 29 Structure of CPU mode register .....	1-29
Fig. 30 $\overline{ONW}$ function timing .....	1-30
Fig. 31 Programming and testing of One Time PROM version .....	1-32
Fig. 32 Timing chart after an interrupt occurs .....	1-34
Fig. 33 Time up to execution of the interrupt processing routine .....	1-34

## CHAPTER 2 APPLICATION

Fig. 2.1.1 Memory map of I/O port related registers .....	2-2
Fig. 2.1.2 Structure of Port Pi (i=0, 1, 2, 3, 4, 5, 6, 7) .....	2-3
Fig. 2.1.3 Structure of Port Pi direction register (i=0, 1, 2, 3, 4, 5, 6, 7) .....	2-3
Fig. 2.2.1 Memory map of timer related registers .....	2-5
Fig. 2.2.2 Structure of Prescaler 12, Prescaler X, Prescaler Y .....	2-6
Fig. 2.2.3 Structure of Timer 1 .....	2-6
Fig. 2.2.4 Structure of Timer 2, Timer X, Timer Y .....	2-7
Fig. 2.2.5 Structure of Timer XY mode register .....	2-8
Fig. 2.2.6 Structure of Interrupt request register 1 .....	2-9

## List of figures

---

Fig. 2.2.7 Structure of Interrupt request register 2 .....	2-9
Fig. 2.2.8 Structure of Interrupt control register 1 .....	2-10
Fig. 2.2.9 Structure of Interrupt control register 2 .....	2-10
Fig. 2.2.10 Connection of timers and setting of division ratios [Clock function] .....	2-12
Fig. 2.2.11 Setting of related registers [Clock function] .....	2-13
Fig. 2.2.12 Control procedure [Clock function] .....	2-14
Fig. 2.2.13 Example of a peripheral circuit .....	2-15
Fig. 2.2.14 Connection of the timer and setting of the division ratio [Piezoelectric buzzer output] .....	2-15
Fig. 2.2.15 Setting of related registers [Piezoelectric buzzer output] .....	2-16
Fig. 2.2.16 Control procedure [Piezoelectric buzzer output] .....	2-16
Fig. 2.2.17 A method for judging if input pulse exists .....	2-17
Fig. 2.2.18 Setting of related registers [Measurement of frequency] .....	2-18
Fig. 2.2.19 Control procedure [Measurement of frequency] .....	2-19
Fig. 2.2.20 Connection of the timer and setting of the division ratio [Measurement of pulse width] .....	2-20
Fig. 2.2.21 Setting of related registers [Measurement of pulse width] .....	2-21
Fig. 2.2.22 Control procedure [Measurement of pulse width] .....	2-22
Fig. 2.3.1 Memory map of serial I/O related registers .....	2-23
Fig. 2.3.2 Structure of Transmit/Receive buffer register .....	2-24
Fig. 2.3.3 Structure of Serial I/O status register .....	2-24
Fig. 2.3.4 Structure of Serial I/O control register .....	2-25
Fig. 2.3.5 Structure of UART control register .....	2-25
Fig. 2.3.6 Structure of Baud rate generator .....	2-26
Fig. 2.3.7 Structure of Interrupt edge selection register .....	2-26
Fig. 2.3.8 Structure of Interrupt request register 1 .....	2-27
Fig. 2.3.9 Structure of Interrupt control register 1 .....	2-27
Fig. 2.3.10 Serial I/O connection examples (1) .....	2-28
Fig. 2.3.11 Serial I/O connection examples (2) .....	2-29
Fig. 2.3.12 Setting of Serial I/O transfer data format .....	2-30
Fig. 2.3.13 Connection diagram [Communication using a clock synchronous serial I/O] .....	2-31
Fig. 2.3.14 Timing chart [Communication using a clock synchronous serial I/O] .....	2-31
Fig. 2.3.15 Setting of related registers at a transmitting side [Communication using a clock synchronous serial I/O] .....	2-32
Fig. 2.3.16 Setting of related registers at a receiving side [Communication using a clock synchronous serial I/O] .....	2-33
Fig. 2.3.17 Control procedure at a transmitting side [Communication using a clock synchronous serial I/O] .....	2-34
Fig. 2.3.18 Control procedure at a receiving side[Communication using a clock synchronous serial I/O] .	2-35
Fig. 2.3.19 Connection diagram [Output of serial data] .....	2-36
Fig. 2.3.20 Timing chart [Output of serial data] .....	2-36
Fig. 2.3.21 Setting of serial I/O related registers [Output of serial data] .....	2-37
Fig. 2.3.22 Setting of serial I/O transmission data [Output of serial data] .....	2-37
Fig. 2.3.23 Control procedure of serial I/O [Output of serial data] .....	2-38
Fig. 2.3.24 Connection diagram [Cyclic transmission or reception of block data between microcomputers]	2-39
Fig. 2.3.25 Timing chart [Cyclic transmission or reception of block data between microcomputers] .....	2-40
Fig. 2.3.26 Setting of related registers [Cyclic transmission or reception of block data between microcomputers] .	2-40
Fig. 2.3.27 Control in the master unit .....	2-41
Fig. 2.3.28 Control in the slave unit .....	2-42
Fig. 2.3.29 Connection diagram [Communication using UART] .....	2-43
Fig. 2.3.30 Timing chart [Communication using UART] .....	2-43

Fig. 2.3.31 Setting of related registers at a transmitting side [Communication using UART] .....	2-45
Fig. 2.3.32 Setting of related registers at a receiving side [Communication using UART] .....	2-46
Fig. 2.3.33 Control procedure at a transmitting side [Communication using UART] .....	2-47
Fig. 2.3.34 Control procedure at a receiving side [Communication using UART] .....	2-48
Fig. 2.4.1 Memory map of processor mode related register .....	2-49
Fig. 2.4.2 Structure of CPU mode register .....	2-49
Fig. 2.4.3 Expansion example of ROM and RAM .....	2-50
Fig. 2.4.4 Read-cycle (OE access, SRAM) .....	2-51
Fig. 2.4.5 Read-cycle (OE access, EPROM) .....	2-51
Fig. 2.4.6 Write-cycle (W control, SRAM) .....	2-52
Fig. 2.4.7 Application example of the $\overline{ONW}$ function .....	2-53
Fig. 2.5.1 Example of Poweron reset circuit .....	2-54
Fig. 2.5.2 RAM back-up system .....	2-54

## List of figures

---

### CHAPTER 3 APPENDIX

---

Fig. 3.1.1 Circuit for measuring output switching characteristics.....	3-11
Fig. 3.1.2 Timing diagram (in single-chip mode) .....	3-12
Fig. 3.1.3 Timing diagram (in memory expansion mode and microprocessor mode) (1) .....	3-13
Fig. 3.1.4 Timing diagram (in memory expansion mode and microprocessor mode) (2) .....	3-14
Fig. 3.2.1 Power source current characteristic example .....	3-15
Fig. 3.2.2 Power source current characteristic example (in wait mode).....	3-15
Fig. 3.2.3 Standard characteristic example of CMOS output port at P-channel drive(1). .....	3-16
Fig. 3.2.4 Standard characteristic example of CMOS output port at P-channel drive(2). .....	3-16
Fig. 3.2.5 Standard characteristic example of CMOS output port at N-channel drive(1) .....	3-17
Fig. 3.2.6 Standard characteristic example of CMOS output port at N-channel drive(2) .....	3-17
Fig. 3.3.1 Structure of interrupt control register 2.....	3-18
Fig. 3.4.1 Wiring for the $\overline{\text{RESET}}$ pin .....	3-24
Fig. 3.4.2 Wiring for clock I/O pins.....	3-25
Fig. 3.4.3 Wiring for the VPP pin of the One Time PROM and the EPROM version.....	3-25
Fig. 3.4.4 Bypass capacitor across the VSS line and the VCC line.....	3-25
Fig. 3.4.5 Wiring for a large current signal line .....	3-26
Fig. 3.4.6 Wiring to a signal line where potential levels change frequently .....	3-26
Fig. 3.4.7 Stepup for I/O ports .....	3-26
Fig. 3.4.8 Watchdog timer by software .....	3-27
Fig. 3.5.1 Structure of Port Pi (i=0, 1, 2, 3, 4, 5, 6, 7).....	3-28
Fig. 3.5.2 Structure of Port Pi direction register (i=0, 1, 2, 3, 4, 5, 6, 7).....	3-28
Fig. 3.5.3 Structure of Transmit/Receive buffer register .....	3-29
Fig. 3.5.4 Structure of Serial I/O status register .....	3-29
Fig. 3.5.5 Structure of Serial I/O control register.....	3-30
Fig. 3.5.6 Structure of UART control register .....	3-30
Fig. 3.5.7 Structure of Baud rate generator .....	3-31
Fig. 3.5.8 Structure of Prescaler 12, Prescaler X, Prescaler Y .....	3-31
Fig. 3.5.9 Structure of Timer 1 .....	3-32
Fig. 3.5.10 Structure of Timer 2, Timer X, Timer Y .....	3-32
Fig. 3.5.11 Structure of Timer XY mode register .....	3-33
Fig. 3.5.12 Structure of Interrupt edge selection register .....	3-34
Fig. 3.5.13 Structure of CPU mode register.....	3-34
Fig. 3.5.14 Structure of Interrupt request register 1 .....	3-35
Fig. 3.5.15 Structure of Interrupt request register 2 .....	3-35
Fig. 3.5.16 Structure of Interrupt control register 1 .....	3-36
Fig. 3.5.17 Structure of Interrupt control register 2 .....	3-36

# List of tables

## CHAPTER 1 HARDWARE

Table 1 Pin description .....	1-5
Table 2 List of supported products .....	1-8
Table 3 List of supported products (Extended operating temperature version) .....	1-9
Table 4 Push and pop instructions of accumulator or processor status register .....	1-11
Table 5 Set and clear instructions of each bit of processor status register.....	1-12
Table 6 List of I/O port functions .....	1-16
Table 7 Interrupt vector addresses and priority.....	1-18
Table 8 Functions of ports in memory expansion mode and microprocessor mode .....	1-29
Table 9 Programming adapter .....	1-32
Table 10 Interrupt sources, vector addresses and interrupt priority.....	1-33

## CHAPTER 2 APPLICATION

Table 2.1.1 Handling of unused pins (in single-chip mode) .....	2-4
Table 2.1.2 Handling of unused pins (in memory expansion mode and microprocessor mode) .....	2-4
Table 2.2.1 Function of CNTR0/CNTR1 edge switch bit .....	2-8
Table 2.3.1 Setting examples of Baud rate generator values and transfer bit rate values .....	2-44

## CHAPTER 3 APPENDIX

Table 3.1.1 Absolute maximum ratings .....	3-2
Table 3.1.2 Recommended operating conditions .....	3-2
Table 3.1.3 Electrical characteristics .....	3-3
Table 3.1.4 Timing requirements (1) .....	3-4
Table 3.1.5 Timing requirements (2) .....	3-4
Table 3.1.6 Switching characteristics (1) .....	3-5
Table 3.1.7 Switching characteristics (2) .....	3-5
Table 3.1.8 Timing requirements in memory expansion mode and microprocessor mode (1) .....	3-6
Table 3.1.9 Switching characteristics in memory expansion mode and microprocessor mode (1) .....	3-6
Table 3.1.10 Timing requirements in memory expansion mode and microprocessor mode (2) .....	3-7
Table 3.1.11 Switching characteristics in memory expansion mode and microprocessor mode (2) .....	3-7
Table 3.1.12 Absolute maximum ratings (Extended operating temperature version).....	3-8
Table 3.1.13 Recommended operating conditions (Extended operating temperature version) .....	3-8
Table 3.1.14 Electrical characteristics (Extended operating temperature version).....	3-9
Table 3.1.15 Timing requirements (Extended operating temperature version) .....	3-10
Table 3.1.16 Switching characteristics (Extended operating temperature version) .....	3-10
Table 3.1.17 Timing requirements in memory expansion mode and microprocessor mode (Extended operating temperature version) .....	3-11
Table 3.1.18 Switching characteristics in memory expansion mode and microprocessor mode (Extended operating temperature version) .....	3-11

## List of tables

---

Table 3.3.1 Programming adapter .....	3-22
Table 3.3.2 Setting of programming adapter switch .....	3-22
Table 3.3.3 Setting of PROM programmer address .....	3-23
Table 3.5.1 Function of CNTR0/CNTR1 edge switch bit.....	3-33



# CHAPTER 1

## **HARDWARE**

DESCRIPTION  
FEATURES  
APPLICATIONS  
PIN CONFIGURATION  
FUNCTIONAL BLOCK  
PIN DESCRIPTION  
PART NUMBERING  
GROUP EXPANSION  
FUNCTIONAL DESCRIPTION  
NOTES ON PROGRAMMING  
DATA REQUIRED FOR  
MASK ORDERS  
ROM PROGRAMMING  
METHOD  
FUNCTIONAL DESCRIPTION  
SUPPLEMENT

# HARDWARE

## DESCRIPTION/FEATURES/APPLICATIONS/PIN CONFIGURATION

### DESCRIPTION

The 3800 group is the 8-bit microcomputer based on the 740 family core technology.

The 3800 group is designed for office automation equipment, household appliances and include four timers, serial I/O function.

The various microcomputers in the 3800 group include variations of internal memory size and packaging. For details, refer to the section on part numbering.

For details on availability of microcomputers in the 3800 group, refer to the section on group expansion.

### FEATURES

- Basic machine-language instructions ..... 71
- The minimum instruction execution time ..... 0.5  $\mu$ s  
(at 8 MHz oscillation frequency)
- Memory size
  - ROM ..... 8 K to 32 K bytes
  - RAM ..... 384 to 1024 bytes
- Programmable input/output ports ..... 58
- Interrupts ..... 15 sources, 15 vectors
- Timers ..... 8 bit X 4
- Serial I/O ..... 8-bit X 1 (UART or Clock-synchronized)
- Clock generating circuit ..... Internal feedback resistor  
(connect to external ceramic resonator or quartz-crystal oscillator)

- Power source voltage ..... 3.0 to 5.5 V  
(Extended operating temperature version : 4.0 to 5.5 V)
- Power dissipation ..... 32 mW
- Memory expansion possible
- Operating temperature range ..... -20 to 85°C  
(Extended operating temperature version : -40 to 85°C)

### APPLICATIONS

Office automation, factory automation, household appliances, and other consumer applications, etc.

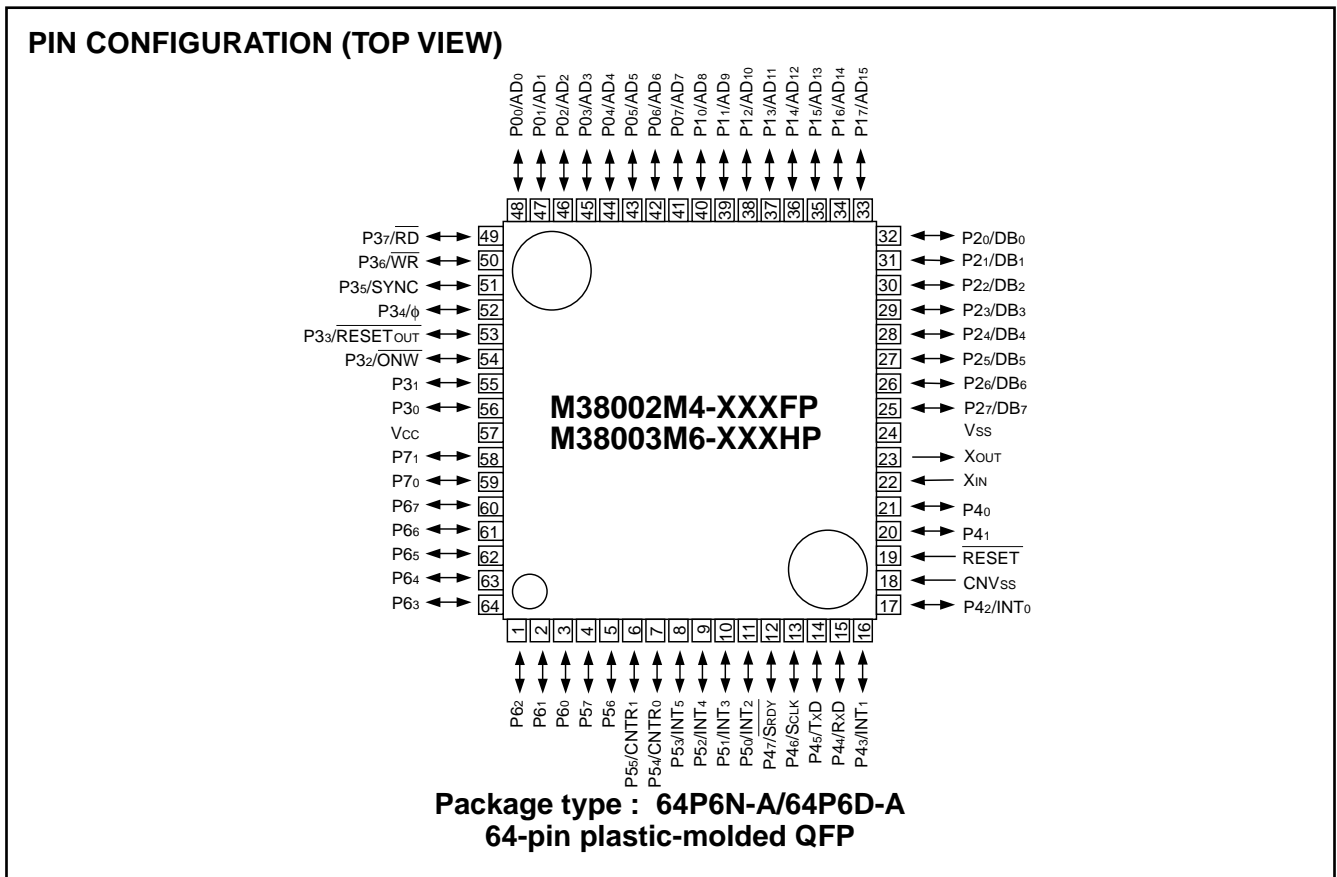


Fig. 1 Pin configuration of M38002M4-XXXFP/M38003M6-XXXHP



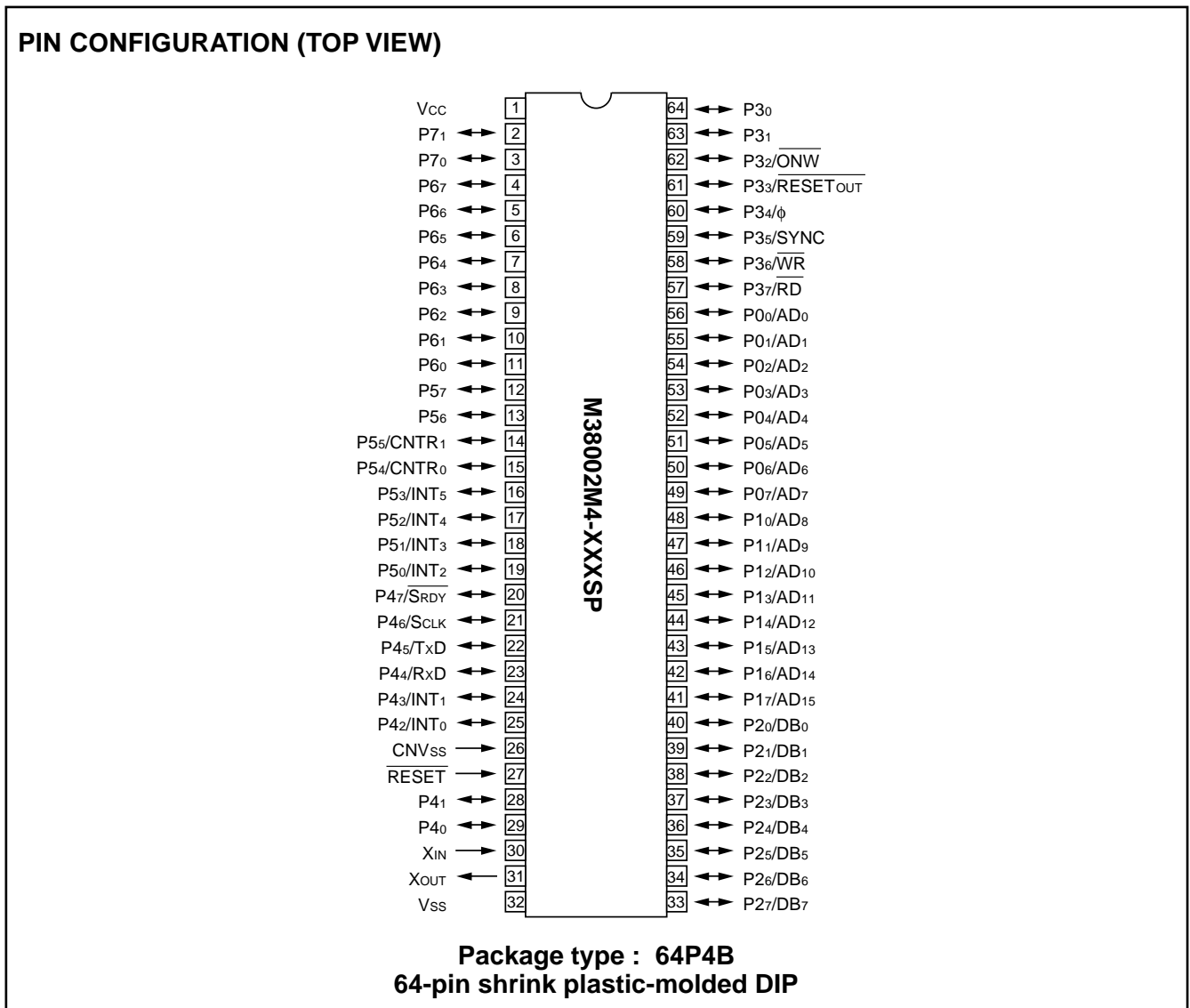


Fig. 2 Pin configuration of M38002M4-XXXSP

# HARDWARE

## FUNCTIONAL BLOCK

### FUNCTIONAL BLOCK

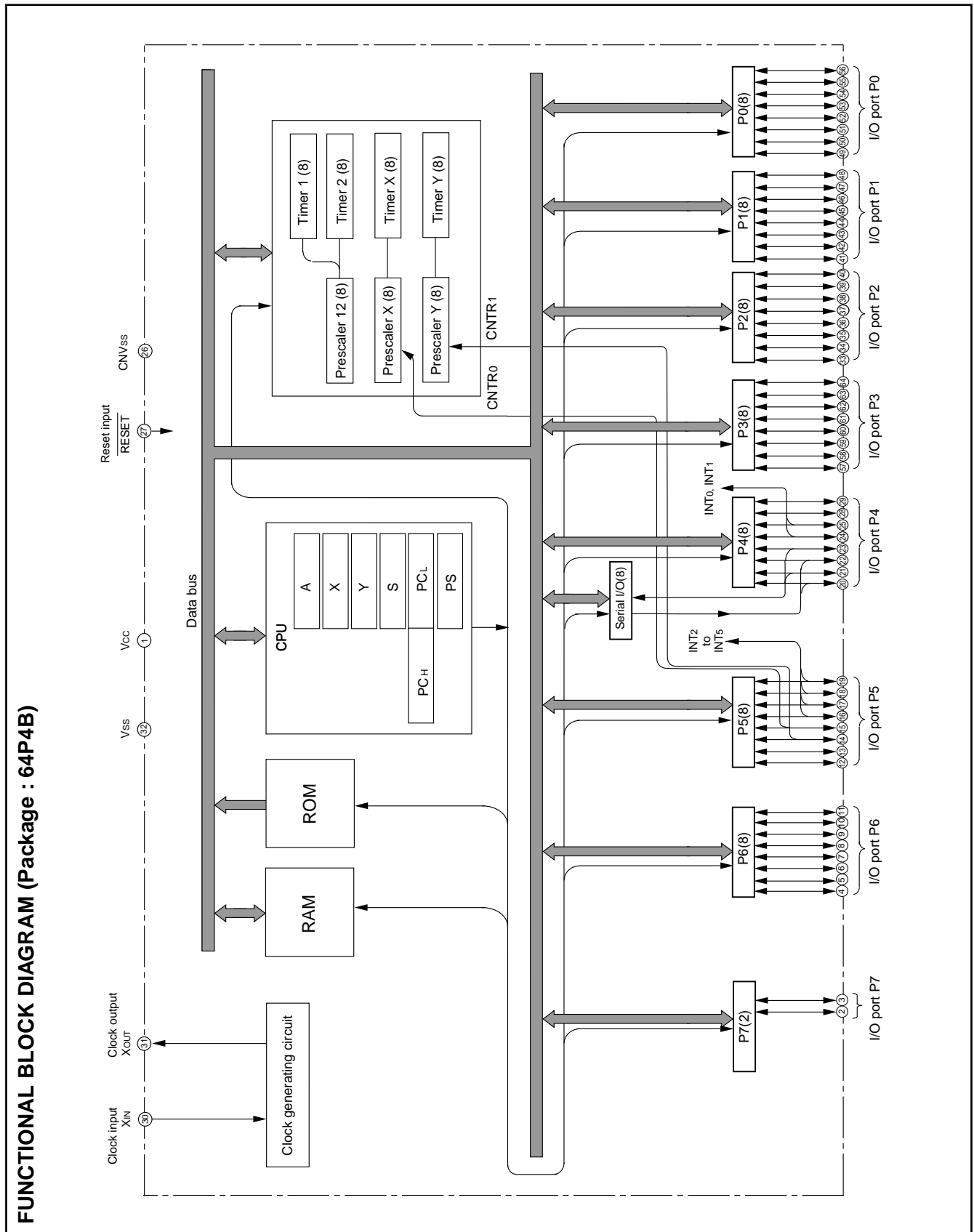


Fig. 3 Functional block diagram

### PIN DESCRIPTION

Table 1. Pin description

Pin	Name	Function	
			Function except a port function
Vcc	Power source	<ul style="list-style-type: none"> <li>Apply voltage of 3.0 V to 5.5 V to Vcc, and 0 V to Vss. (Extended operating temperature version : 4.0 V to 5.5 V)</li> </ul>	
Vss			
CNVss	CNVss	<ul style="list-style-type: none"> <li>This pin controls the operation mode of the chip.</li> <li>Normally connected to Vss.</li> <li>If this pin is connected to Vcc, the internal ROM is inhibited and external memory is accessed.</li> </ul>	
$\overline{\text{RESET}}$	Reset input	<ul style="list-style-type: none"> <li>Reset input pin for active "L"</li> </ul>	
XIN	Clock input	<ul style="list-style-type: none"> <li>Input and output signals for the internal clock generating circuit.</li> <li>Connect a ceramic resonator or quartz-crystal oscillator between the XIN and XOUT pins to set the oscillation frequency.</li> <li>If an external clock is used, connect the clock source to the XIN pin and leave the XOUT pin open.</li> <li>The clock is used as the oscillating source of system clock.</li> </ul>	
XOUT	Clock output		
P0 <sub>0</sub> – P0 <sub>7</sub>	I/O port P0	<ul style="list-style-type: none"> <li>8 bit CMOS I/O port</li> <li>I/O direction register allows each pin to be individually programmed as either input or output.</li> <li>At reset this port is set to input mode.</li> <li>In modes other than single-chip, these pins are used as address, data, and control bus I/O pins.</li> <li>CMOS compatible input level</li> <li>CMOS 3-state output structure</li> </ul>	
P1 <sub>0</sub> – P1 <sub>7</sub>	I/O port P1		
P2 <sub>0</sub> – P2 <sub>7</sub>	I/O port P2		
P3 <sub>0</sub> – P3 <sub>7</sub>	I/O port P3		
P4 <sub>0</sub> , P4 <sub>1</sub>	I/O port P4	<ul style="list-style-type: none"> <li>8-bit CMOS I/O port with the same function as port P0</li> <li>CMOS compatible input level</li> <li>CMOS 3-state output structure</li> </ul>	
P4 <sub>2</sub> /INT <sub>0</sub> , P4 <sub>3</sub> /INT <sub>1</sub>		<ul style="list-style-type: none"> <li>External interrupt input pins</li> </ul>	
P4 <sub>4</sub> /RxD, P4 <sub>5</sub> /TxD, P4 <sub>6</sub> /SCLK, P4 <sub>7</sub> /SRDY		<ul style="list-style-type: none"> <li>Serial I/O I/O pins</li> </ul>	
P5 <sub>0</sub> /INT <sub>2</sub> – P5 <sub>3</sub> /INT <sub>5</sub>	I/O port P5	<ul style="list-style-type: none"> <li>8-bit CMOS I/O port with the same function as port P0</li> <li>CMOS compatible input level</li> <li>CMOS 3-state output structure</li> </ul>	
P5 <sub>4</sub> /CNTR <sub>0</sub> , P5 <sub>5</sub> /CNTR <sub>1</sub>		<ul style="list-style-type: none"> <li>External interrupt input pins</li> </ul>	
P5 <sub>6</sub> , P5 <sub>7</sub>		<ul style="list-style-type: none"> <li>Timer X and Timer Y I/O pins</li> </ul>	
P6 <sub>0</sub> – P6 <sub>7</sub>	I/O port P6	<ul style="list-style-type: none"> <li>8-bit CMOS I/O port with the same function as port P0</li> <li>CMOS compatible input level</li> <li>CMOS 3-state output structure</li> </ul>	
P7 <sub>0</sub> , P7 <sub>1</sub>	I/O port P7	<ul style="list-style-type: none"> <li>2-bit CMOS I/O port with the same function as port P0</li> <li>CMOS compatible input level</li> <li>CMOS 3-state output structure</li> </ul>	

# HARDWARE

## PART NUMBERING

---

### PART NUMBERING

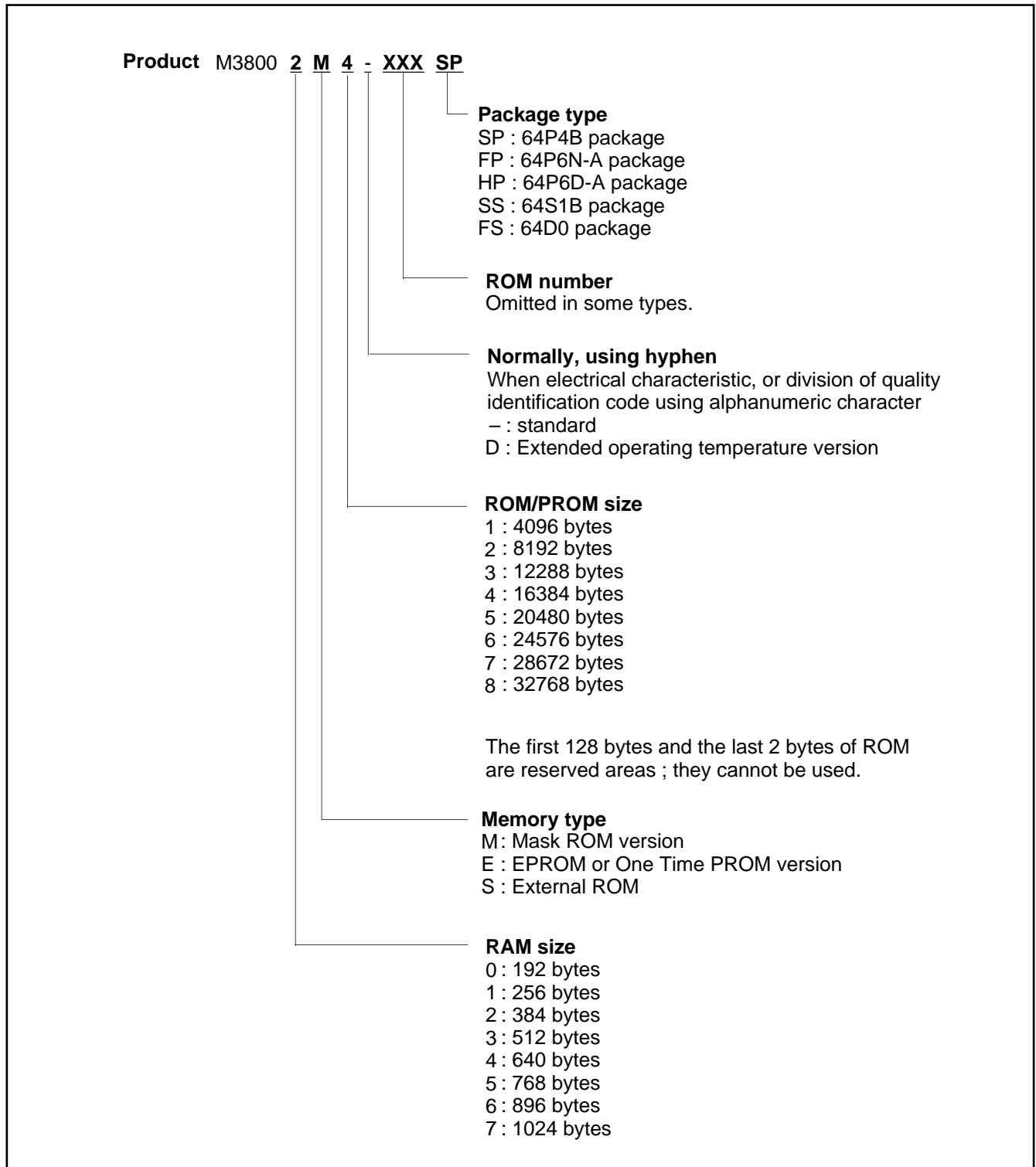


Fig. 4 Part numbering

### GROUP EXPANSION

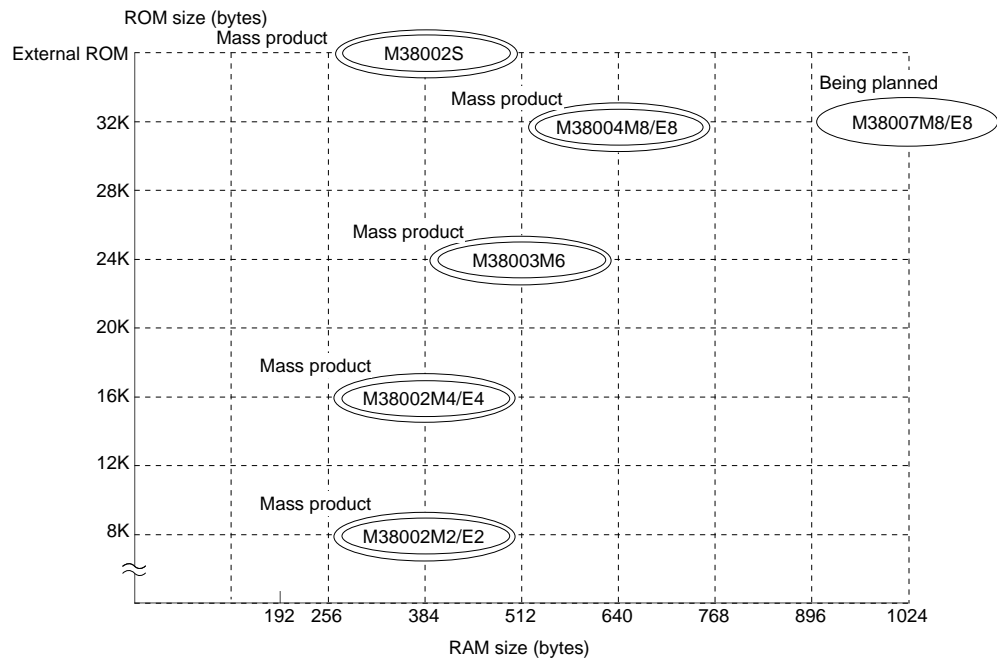
Mitsubishi plans to expand the 3800 group as follows:

- (1) Support for mask ROM, One Time PROM, EPROM, and external ROM versions  
 ROM/PROM capacity ..... 8 K to 32 K bytes  
 RAM capacity ..... 384 to 1024 bytes

(2) Packages

- 64P4B ..... Shrink plastic molded DIP  
 64P6N-A ..... 0.8 mm pitch plastic molded QFP  
 64P6D-A ..... 0.5 mm pitch plastic molded QFP  
 64S1B ..... Shrink ceramic DIP (EPROM version)  
 64D0 ..... 0.8 mm pitch ceramic LCC (EPROM version)

### Memory Expansion Plan



**Note :** Products under development or planning: the development schedule and specifications may be revised without notice.

Fig. 5 Memory expansion plan

# HARDWARE

## GROUP EXPANSION

Currently supported products are listed below.

**Table 2. List of supported products**

**As of September 1995**

Product	(P) ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38002M2-XXXSP	8192 (8062)	384	64P4B	Mask ROM version
M38002E2-XXXSP				One Time PROM version
M38002E2SP				One Time PROM version (blank)
M38002M2-XXXFP			64P6N-A	Mask ROM version
M38002E2-XXXFP				One Time PROM version
M38002E2FP				One Time PROM version (blank)
M38002M4-XXXSP	16384 (16254)	384	64P4B	Mask ROM version
M38002E4-XXXSP				One Time PROM version
M38002E4SP				One Time PROM version (blank)
M38002E4SS			64S1B-E	EPROM version
M38002M4-XXXFP			64P6N-A	Mask ROM version
M38002E4-XXXFP				One Time PROM version
M38002E4FP				One Time PROM version (blank)
M38002E4FS			64D0	EPROM version
M38003M6-XXXSP	24576 (24446)	512	64P4B	Mask ROM version
M38003M6-XXXFP			64P6N-A	Mask ROM version
M38003M6-XXXHP			64P6D-A	Mask ROM version
M38004M8-XXXSP	32768 (32638)	640	64P4B	Mask ROM version
M38004E8-XXXSP				One Time PROM version
M38004E8SP				One Time PROM version (blank)
M38004E8SS			64S1B-E	EPROM version
M38004M8-XXXFP			64P6N-A	Mask ROM version
M38004E8-XXXFP				One Time PROM version
M38004E8FP				One Time PROM version (blank)
M38004E8FS			64D0	EPROM version
M38002SSP	0	384	64P4B	External ROM type
M38002SFP			64P6N-A	External ROM type

### GROUP EXPANSION (EXTENDED OPERATING TEMPERATURE VERSION)

Mitsubishi plans to expand the 3800 group (extended operating temperature version) as follows:

- (1) Support for mask ROM, One Time PROM, and EPROM versions  
 ROM/PROM capacity ..... 8 K to 32 K bytes  
 RAM capacity ..... 384 to 640 bytes

- (2) Packages  
 64P4B ..... Shrink Plastic molded DIP  
 64P6N-A ..... 0.8 mm pitch plastic molded QFP

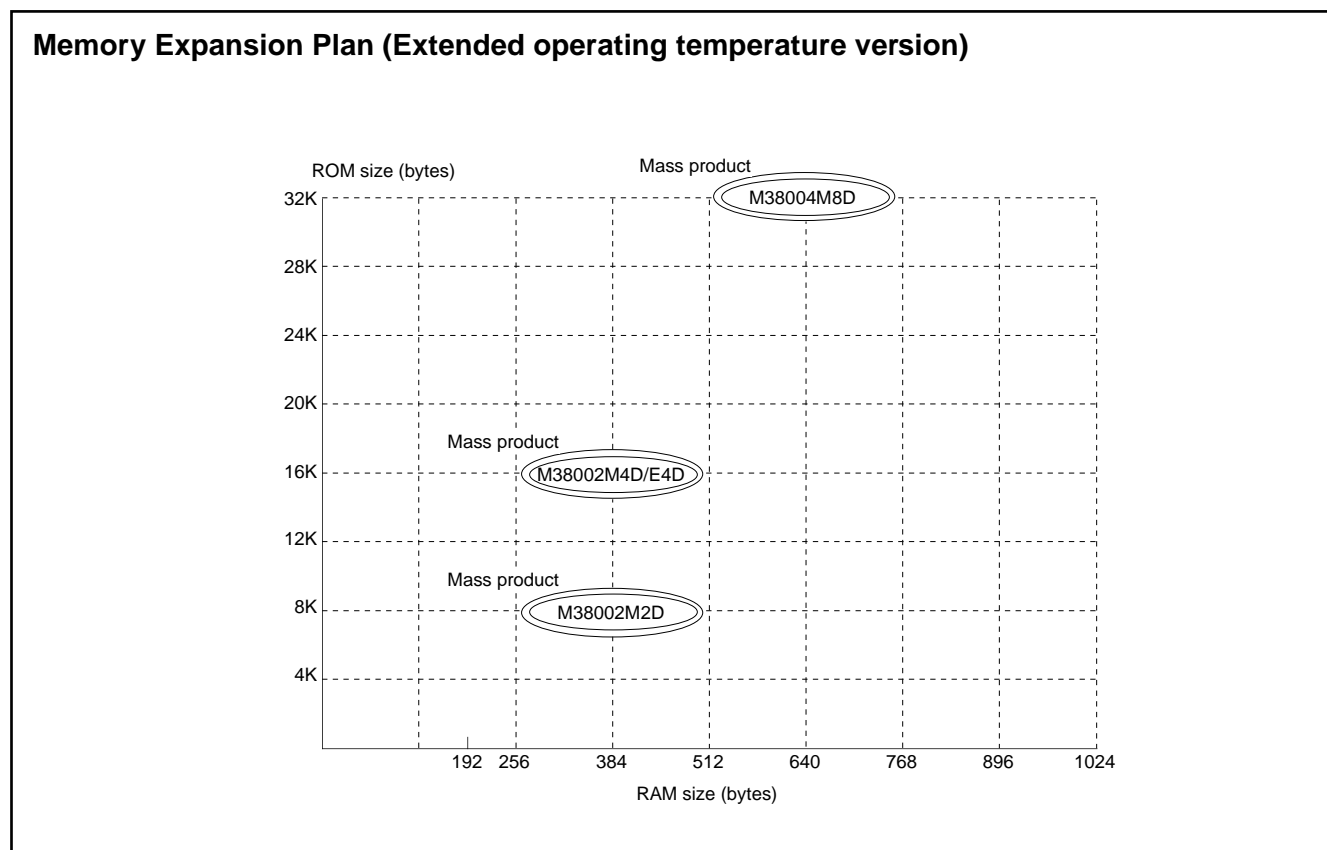


Fig. 6 Memory expansion plan (Extended operating temperature version)

Currently supported products are listed below.

Table 3. List of supported products (Extended operating temperature version)

As of September 1995

Product name	(P) ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38002M2DXXXSP	8192 (8062)	384	64P4B	Mask ROM version
M38002M2DXXXFP			64P6N-A	Mask ROM version
M38002M4DXXXSP	16384 (16254)	384	64P4B	Mask ROM version
M38002E4DXXXSP				One Time PROM version
M38002E4DSP				One Time PROM version (blank)
M38002M4DXXXFP			64P6N-A	Mask ROM version
M38002E4DXXXFP				One Time PROM version
M38002E4DFP				One Time PROM version (blank)
M38004M8DXXXSP				64P4B
M38004M8DXXXFP		64P6N-A	Mask ROM version	

# HARDWARE

## FUNCTIONAL DESCRIPTION

### FUNCTIONAL DESCRIPTION Central Processing Unit (CPU)

The 3800 group uses the standard 740 family instruction set. Refer to the table of 740 family addressing modes and machine instructions or the SERIES 740 <Software> User's Manual for details on the instruction set.

Machine-resident 740 family instructions are as follows:

The FST and SLW instruction cannot be used.

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

The central processing unit (CPU) has the six registers.

#### Accumulator (A)

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

#### Index register X (X), Index register Y (Y)

Both index register X and index register Y are 8-bit registers. In the index addressing modes, the value of the OPERAND is added to the contents of register X or register Y and specifies the real address.

When the T flag in the processor status register is set to "1", the value contained in index register X becomes the address for the second OPERAND.

#### Stack pointer (S)

The stack pointer is an 8-bit register used during subroutine calls and interrupts. The stack is used to store the current address data and processor status when branching to subroutines or interrupt routines.

The lower eight bits of the stack address are determined by the contents of the stack pointer. The upper eight bits of the stack address are determined by the Stack Page Selection Bit. If the Stack Page Selection Bit is "0", then the RAM in the zero page is used as the stack area. If the Stack Page Selection Bit is "1", then RAM in page 1 is used as the stack area.

The Stack Page Selection Bit is located in the SFR area in the zero page. Note that the initial value of the Stack Page Selection Bit varies with each microcomputer type. Also some microcomputer types have no Stack Page Selection Bit and the upper eight bits of the stack address are fixed.

The operations of pushing register contents onto the stack and popping them from the stack are shown in Fig. 8.

#### Program counter (PC)

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

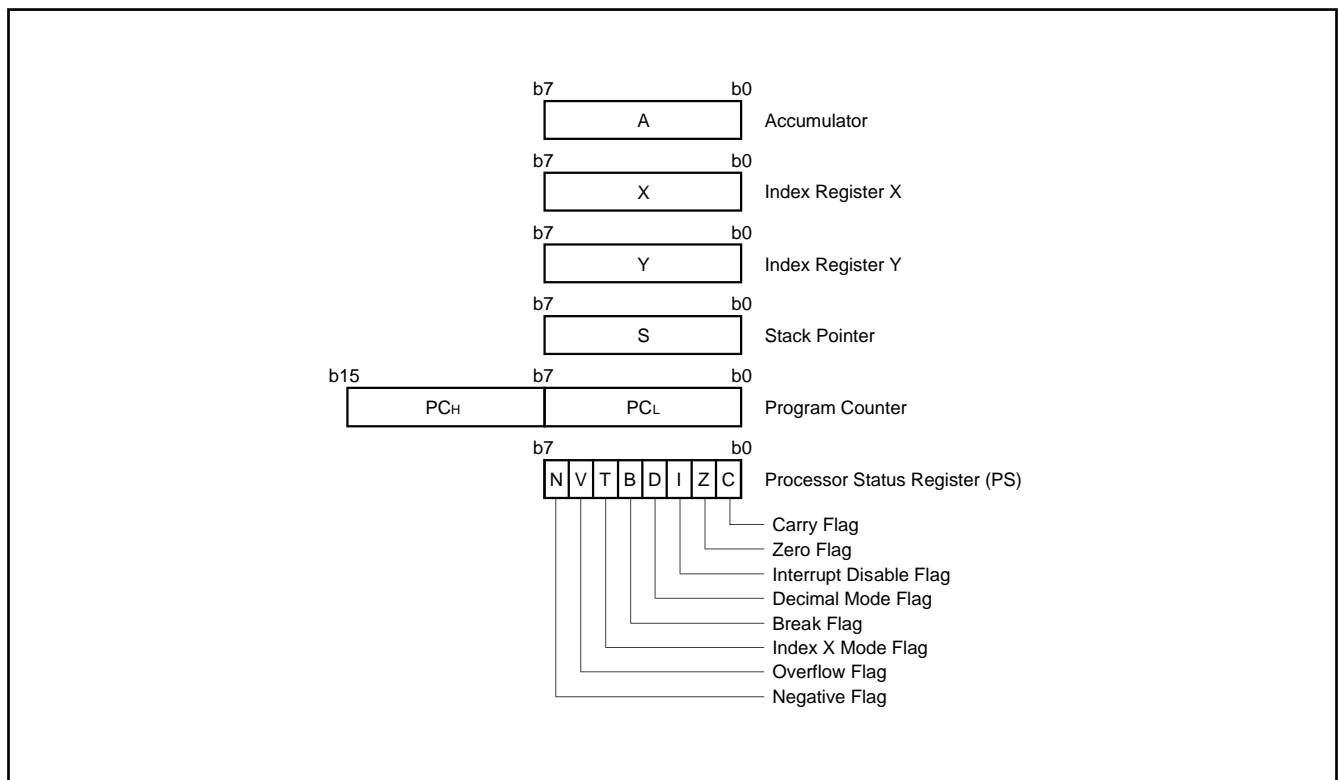


Fig. 7 740 Family CPU register structure



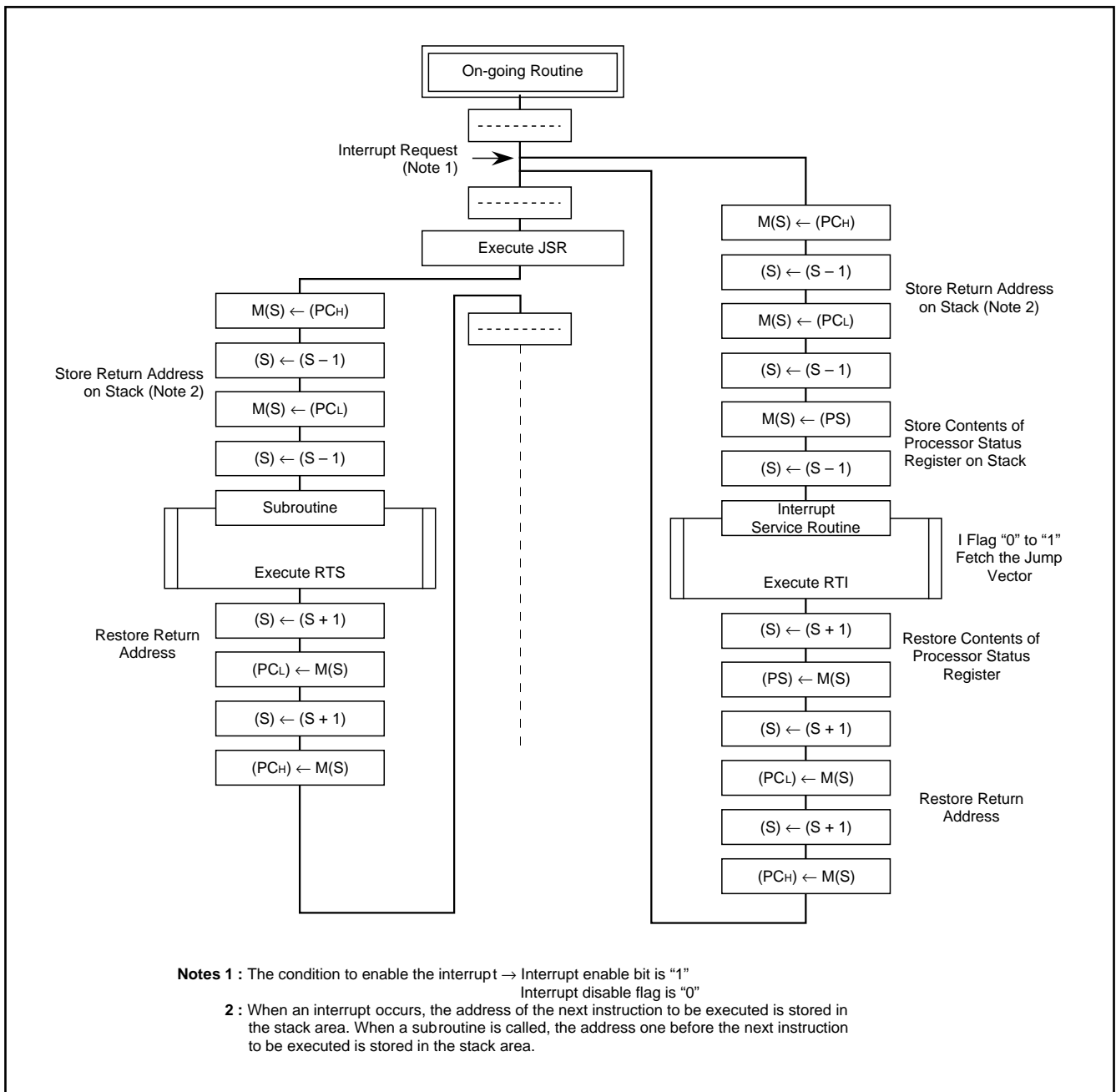


Fig. 8 Register push and pop at interrupt generation and subroutine call

Table 4. Push and pop instructions of accumulator or processor status register

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

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Processor status register (PS)

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

After reset, the Interrupt disable (I) flag is set to "1", but all other flags are undefined. Since the Index X mode (T) and Decimal mode (D) flags directly affect arithmetic operations, they should be initialized in the beginning of a program.

(1) Carry flag (C)

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

(2) Zero flag (Z)

The Z flag is set if the result of an immediate arithmetic operation or a data transfer is "0", and cleared if the result is anything other than "0".

(3) Interrupt disable flag (I)

The I flag disables all interrupts except for the interrupt generated by the BRK instruction.

Interrupts are disabled when the I flag is "1".

When an interrupt occurs, this flag is automatically set to "1" to prevent other interrupts from interfering until the current interrupt is serviced.

(4) Decimal mode flag (D)

The D flag determines whether additions and subtractions are executed in binary or decimal. Binary arithmetic is executed when this flag is "0"; decimal arithmetic is executed when it is "1". Decimal correction is automatic in decimal mode. Only the ADC and SBC instructions can be used for decimal arithmetic.

(5) Break flag (B)

The B flag is used to indicate that the current interrupt was generated by the BRK instruction. The BRK flag in the processor status register is always "0". When the BRK instruction is used to generate an interrupt, the processor status register is pushed onto the stack with the break flag set to "1". The saved processor status is the only place where the break flag is ever set.

(6) Index X mode flag (T)

When the T flag is "0", arithmetic operations are performed between accumulator and memory, e.g. the results of an operation between two memory locations is stored in the accumulator. When the T flag is "1", direct arithmetic operations and direct data transfers are enabled between memory locations, i.e. between memory and memory, memory and I/O, and I/O and I/O. In this case, the result of an arithmetic operation performed on data in memory location 1 and memory location 2 is stored in memory location 1. The address of memory location 1 is specified by index register X, and the address of memory location 2 is specified by normal addressing modes.

(7) Overflow flag (V)

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

(8) Negative flag (N)

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

Table 5. Set and clear instructions of each bit of processor status register

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

### CPU mode register

The CPU mode register is allocated at address 003B16.

The CPU mode register contains the stack page selection bit.

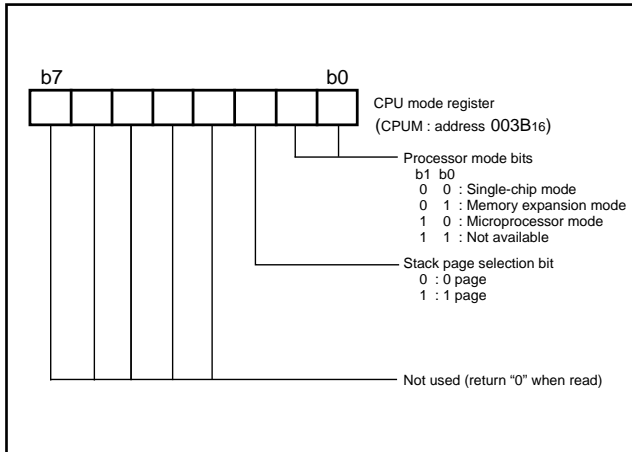


Fig. 9 Structure of CPU mode register

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Memory

#### Special function register (SFR) area

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

#### RAM

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

#### ROM

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

#### Interrupt vector area

The interrupt vector area contains reset and interrupt vectors.

#### Zero page

The 256 bytes from addresses 0000<sub>16</sub> to 00FF<sub>16</sub> are called the zero page area. The internal RAM and the special function registers (SFR) are allocated to this area.

The zero page addressing mode can be used to specify memory and register addresses in the zero page area. Access to this area with only 2 bytes is possible in the zero page addressing mode.

#### Special page

The 256 bytes from addresses FF00<sub>16</sub> to FFFF<sub>16</sub> are called the special page area. The special page addressing mode can be used to specify memory addresses in the special page area. Access to this area with only 2 bytes is possible in the special page addressing mode.

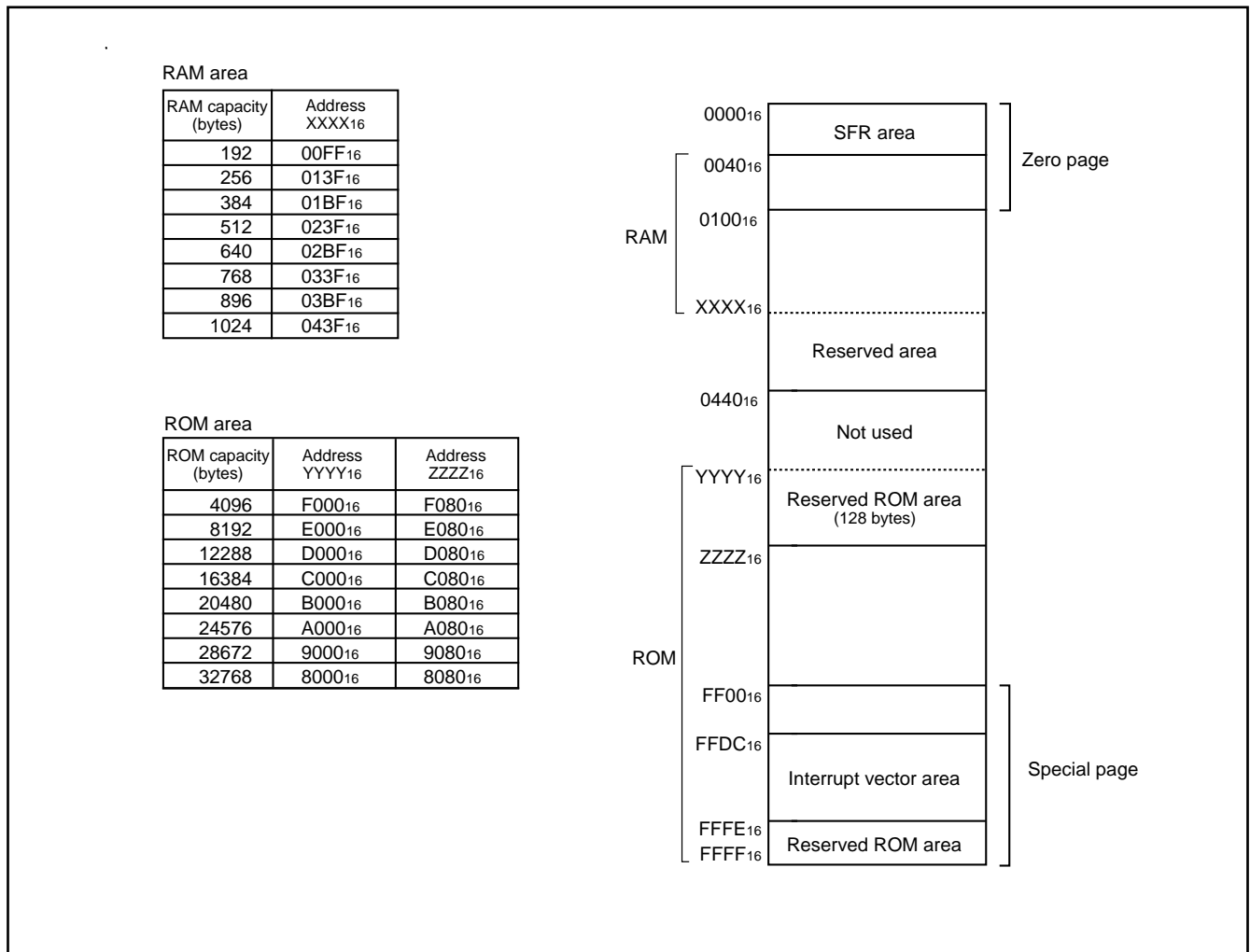


Fig. 10 Memory map diagram

0000 <sub>16</sub>	Port P0 (P0)	0020 <sub>16</sub>	Prescaler 12 (PRE12)
0001 <sub>16</sub>	Port P0 direction register (P0D)	0021 <sub>16</sub>	Timer 1 (T1)
0002 <sub>16</sub>	Port P1 (P1)	0022 <sub>16</sub>	Timer 2 (T2)
0003 <sub>16</sub>	Port P1 direction register (P1D)	0023 <sub>16</sub>	Timer XY mode register (TM)
0004 <sub>16</sub>	Port P2 (P2)	0024 <sub>16</sub>	Prescaler X (PREX)
0005 <sub>16</sub>	Port P2 direction register (P2D)	0025 <sub>16</sub>	Timer X (TX)
0006 <sub>16</sub>	Port P3 (P3)	0026 <sub>16</sub>	Prescaler Y (PREY)
0007 <sub>16</sub>	Port P3 direction register (P3D)	0027 <sub>16</sub>	Timer Y (TY)
0008 <sub>16</sub>	Port P4 (P4)	0028 <sub>16</sub>	
0009 <sub>16</sub>	Port P4 direction register (P4D)	0029 <sub>16</sub>	
000A <sub>16</sub>	Port P5 (P5)	002A <sub>16</sub>	
000B <sub>16</sub>	Port P5 direction register (P5D)	002B <sub>16</sub>	
000C <sub>16</sub>	Port P6 (P6)	002C <sub>16</sub>	
000D <sub>16</sub>	Port P6 direction register (P6D)	002D <sub>16</sub>	
000E <sub>16</sub>	Port P7 (P7)	002E <sub>16</sub>	
000F <sub>16</sub>	Port P7 direction register (P7D)	002F <sub>16</sub>	
0010 <sub>16</sub>		0030 <sub>16</sub>	
0011 <sub>16</sub>		0031 <sub>16</sub>	
0012 <sub>16</sub>		0032 <sub>16</sub>	
0013 <sub>16</sub>		0033 <sub>16</sub>	
0014 <sub>16</sub>		0034 <sub>16</sub>	
0015 <sub>16</sub>		0035 <sub>16</sub>	
0016 <sub>16</sub>		0036 <sub>16</sub>	
0017 <sub>16</sub>		0037 <sub>16</sub>	
0018 <sub>16</sub>	Transmit/Receive buffer register (TB/RB)	0038 <sub>16</sub>	
0019 <sub>16</sub>	Serial I/O status register (SIOSTS)	0039 <sub>16</sub>	
001A <sub>16</sub>	Serial I/O control register (SIOCON)	003A <sub>16</sub>	Interrupt edge selection register (INTEDGE)
001B <sub>16</sub>	UART control register (UARTCON)	003B <sub>16</sub>	CPU mode register (CPUM)
001C <sub>16</sub>	Baud rate generator (BRG)	003C <sub>16</sub>	Interrupt request register 1(IREQ1)
001D <sub>16</sub>		003D <sub>16</sub>	Interrupt request register 2(IREQ2)
001E <sub>16</sub>		003E <sub>16</sub>	Interrupt control register 1(ICON1)
001F <sub>16</sub>		003F <sub>16</sub>	Interrupt control register 2(ICON2)

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

# HARDWARE

## FUNCTIONAL DESCRIPTION

### I/O Ports

#### Direction registers

The 3800 group has 58 programmable I/O pins arranged in eight I/O ports (ports P0 to P7). The I/O ports have direction registers which determine the input/output direction of each individual pin. Each bit in a direction register corresponds to one pin, each pin can be set to be input port or output port.

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

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

**Table 6. List of I/O port functions**

Pin	Name	Input/Output	I/O Format	Non-Port Function	Related SFRs	Ref.No.		
P0 <sub>0</sub> – P0 <sub>7</sub>	Port P0	Input/output, individual bits	CMOS 3-state output CMOS compatible input level	Address low-order byte output	CPU mode register	(1)		
P1 <sub>0</sub> – P1 <sub>7</sub>	Port P1	Input/output, individual bits	CMOS 3-state output CMOS compatible input level	Address high-order byte output	CPU mode register			
P2 <sub>0</sub> – P2 <sub>7</sub>	Port P2	Input/output, individual bits	CMOS 3-state output CMOS compatible input level	Data bus I/O	CPU mode register			
P3 <sub>0</sub> – P3 <sub>7</sub>	Port P3	Input/output, individual bits	CMOS 3-state output CMOS compatible input level	Control signal I/O	CPU mode register			
P4 <sub>0</sub> , P4 <sub>1</sub>	Port P4	Input/output, individual bits	CMOS 3-state output CMOS compatible input level	External interrupt input	Interrupt edge selection register	(2)		
P4 <sub>2</sub> /INT <sub>0</sub> , P4 <sub>3</sub> /INT <sub>1</sub>						Serial I/O function I/O	Serial I/O control register UART control register	(3)
P4 <sub>4</sub> /RXD, P4 <sub>5</sub> /TXD, P4 <sub>6</sub> /SCLK, P4 <sub>7</sub> /SRDY								(4)
								(5)
								(6)
P5 <sub>0</sub> /INT <sub>2</sub> , P5 <sub>1</sub> /INT <sub>3</sub> , P5 <sub>2</sub> /INT <sub>4</sub> , P5 <sub>3</sub> /INT <sub>5</sub>	Port P5	Input/output, individual bits	CMOS 3-state output CMOS compatible input level	External interrupt input	Interrupt edge selection register	(2)		
P5 <sub>4</sub> /CNTR <sub>0</sub> , P5 <sub>5</sub> /CNTR <sub>1</sub>						Timer X and Timer Y function I/O	Timer XY mode register	(7)
P5 <sub>6</sub> , P5 <sub>7</sub>								
P6 <sub>0</sub> – P6 <sub>7</sub>	Port P6	Input/output, individual bits	CMOS 3-state output CMOS compatible input level			(1)		
P7 <sub>0</sub> , P7 <sub>1</sub>	Port P7	Input/output, individual bits	CMOS 3-state output CMOS compatible input level					

**Note 1:** For details of the functions of ports P0 to P3 in modes other than single-chip mode, and how to use double-function ports as function I/O ports, refer to the applicable sections.

**2:** Make sure that the input level at each pin is either 0 V or V<sub>CC</sub> during execution of the STP instruction.

When an input level is at an intermediate potential, a current will flow from V<sub>CC</sub> to V<sub>SS</sub> through the input-stage gate.

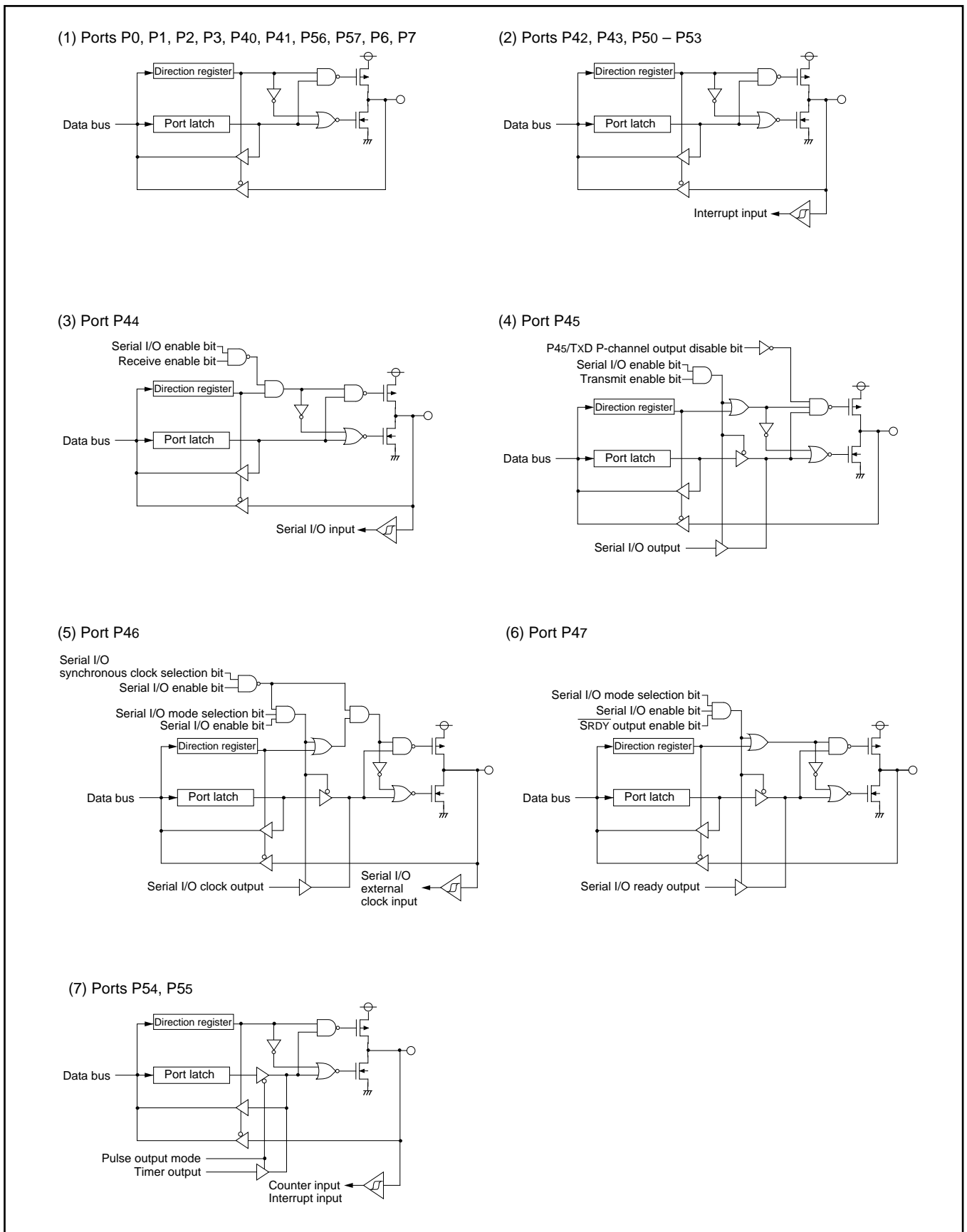


Fig. 12 Port block diagram (single-chip mode)

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Interrupts

Interrupts occur by fifteen sources: eight external, six internal, and one software.

### Interrupt control

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

Interrupt enable bits can be set or cleared by software.

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

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

### Interrupt operation

When an interrupt is received, the contents of the program counter and processor status register are automatically stored into the stack. The interrupt disable flag is set to inhibit other interrupts from interfering. The corresponding interrupt request bit is cleared and the interrupt jump destination address is read from the vector table into the program counter.

### Notes on use

When the active edge of an external interrupt (INT<sub>0</sub> to INT<sub>5</sub>, CNTR<sub>0</sub>, or CNTR<sub>1</sub>) is changed, the corresponding interrupt request bit may also be set. Therefore, please take following sequence;

- (1) Disable the external interrupt which is selected.
- (2) Change the active edge selection.
- (3) Clear the interrupt request bit which is selected to "0".
- (4) Enable the external interrupt which is selected.

**Table 7. Interrupt vector addresses and priority**

Interrupt Source	Priority	Vector Addresses (Note 1)		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset (Note 2)	1	FFFD <sub>16</sub>	FFFC <sub>16</sub>	At reset	Non-maskable
INT <sub>0</sub>	2	FFFB <sub>16</sub>	FFFA <sub>16</sub>	At detection of either rising or falling edge of INT <sub>0</sub> input	External interrupt (active edge selectable)
INT <sub>1</sub>	3	FFF9 <sub>16</sub>	FFF8 <sub>16</sub>	At detection of either rising or falling edge of INT <sub>1</sub> input	External interrupt (active edge selectable)
Serial I/O reception	4	FFF7 <sub>16</sub>	FFF6 <sub>16</sub>	At completion of serial I/O data reception	Valid when serial I/O is selected
Serial I/O transmission	5	FFF5 <sub>16</sub>	FFF4 <sub>16</sub>	At completion of serial I/O transfer shift or when transmission buffer is empty	Valid when serial I/O is selected
Timer X	6	FFF3 <sub>16</sub>	FFF2 <sub>16</sub>	At timer X underflow	
Timer Y	7	FFF1 <sub>16</sub>	FFF0 <sub>16</sub>	At timer Y underflow	
Timer 1	8	FFEF <sub>16</sub>	FFEE <sub>16</sub>	At timer 1 underflow	STP release timer underflow
Timer 2	9	FFED <sub>16</sub>	FFEC <sub>16</sub>	At timer 2 underflow	
CNTR <sub>0</sub>	10	FFEB <sub>16</sub>	FFEA <sub>16</sub>	At detection of either rising or falling edge of CNTR <sub>0</sub> input	External interrupt (active edge selectable)
CNTR <sub>1</sub>	11	FFE9 <sub>16</sub>	FFE8 <sub>16</sub>	At detection of either rising or falling edge of CNTR <sub>1</sub> input	External interrupt (active edge selectable)
INT <sub>2</sub>	12	FFE7 <sub>16</sub>	FFE6 <sub>16</sub>	At detection of either rising or falling edge of INT <sub>2</sub> input	External interrupt (active edge selectable)
INT <sub>3</sub>	13	FFE5 <sub>16</sub>	FFE4 <sub>16</sub>	At detection of either rising or falling edge of INT <sub>3</sub> input	External interrupt (active edge selectable)
INT <sub>4</sub>	14	FFE3 <sub>16</sub>	FFE2 <sub>16</sub>	At detection of either rising or falling edge of INT <sub>4</sub> input	External interrupt (active edge selectable)
INT <sub>5</sub>	15	FFE1 <sub>16</sub>	FFE0 <sub>16</sub>	At detection of either rising or falling edge of INT <sub>5</sub> input	External interrupt (active edge selectable)
BRK instruction	16	FFDD <sub>16</sub>	FFDC <sub>16</sub>	At BRK instruction execution	Non-maskable software interrupt

**Note 1:** Vector addresses contain interrupt jump destination addresses.

**2:** Reset function in the same way as an interrupt with the highest priority.



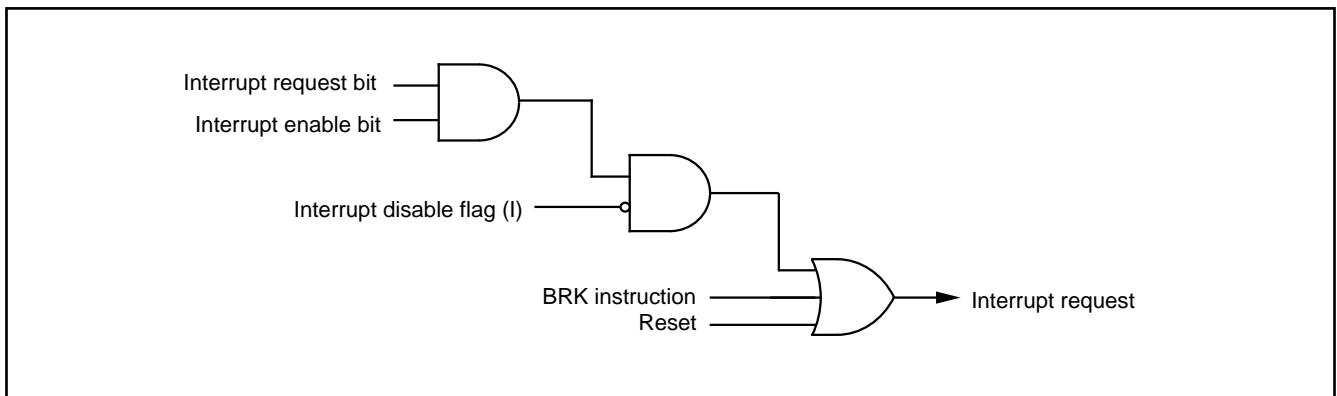


Fig. 13 Interrupt control

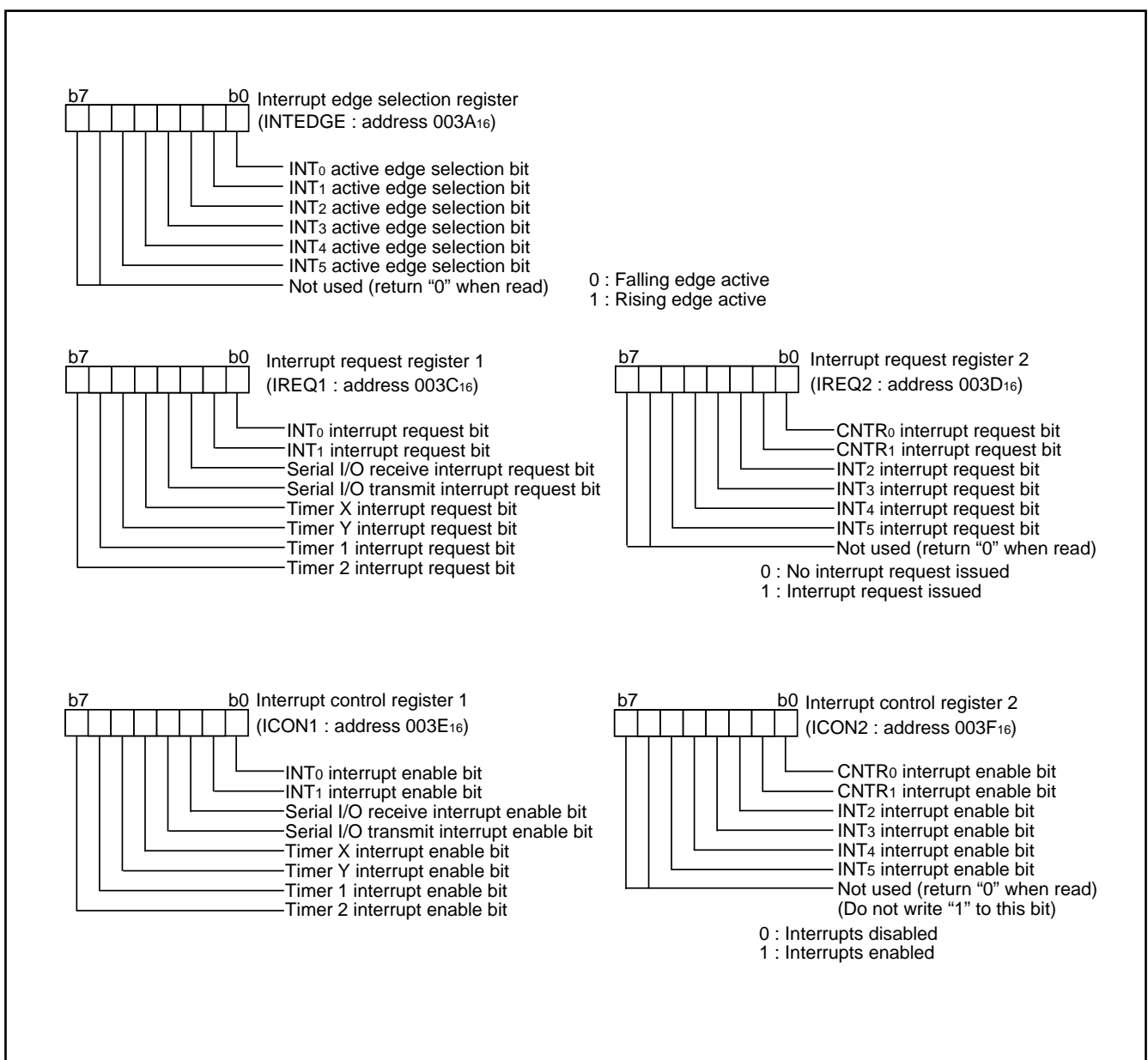


Fig. 14 Structure of interrupt-related registers

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Timers

The 3800 group has four timers: timer X, timer Y, timer 1, and timer 2.

All timers are count down. When the timer reaches “0016”, an underflow occurs at the next count pulse and the corresponding timer latch is reloaded into the timer and the count is continued. When a timer underflows, the interrupt request bit corresponding to that timer is set to “1”.

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

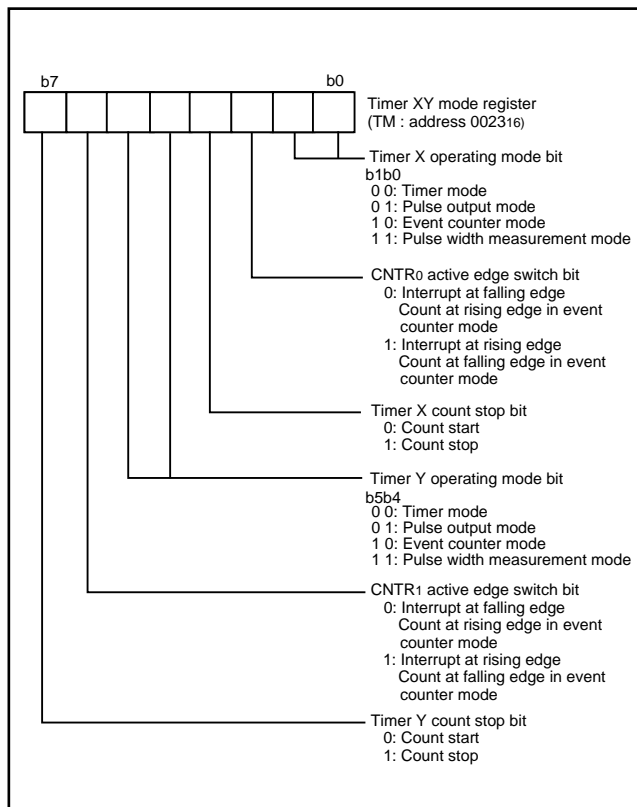


Fig. 15 Structure of timer XY register

### Timer 1 and Timer 2

The count source of prescaler 12 is the oscillation frequency divided by 16. The output of prescaler 12 is counted by timer 1 and timer 2, and a timer underflow sets the interrupt request bit.

### Timer X and Timer Y

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

#### Timer Mode

The timer counts  $f(X_{IN})/16$  in timer mode.

#### Pulse Output Mode

Timer X (or timer Y) counts  $f(X_{IN})/16$ . Whenever the contents of the timer reach “0016”, the signal output from the CNTR0 (or CNTR1) pin is inverted. If the CNTR0 (or CNTR1) active edge switch bit is “0”, output begins at “H”.

If it is “1”, output starts at “L”. When using a timer in this mode, set the corresponding port P54 ( or port P55) direction register to output mode.

#### Event Counter Mode

Operation in event counter mode is the same as in timer mode, except the timer counts signals input through the CNTR0 or CNTR1 pin.

#### Pulse Width Measurement Mode

If the CNTR0 (or CNTR1) active edge selection bit is “0”, the timer counts at the oscillation frequency divided by 16 while the CNTR0 (or CNTR1) pin is at “H”. If the CNTR0 (or CNTR1) active edge switch bit is “1”, the count continues during the time that the CNTR0 (or CNTR1) pin is at “L”.

In all of these modes, the count can be stopped by setting the timer X (timer Y) count stop bit to “1”. Every time a timer underflows, the corresponding interrupt request bit is set.

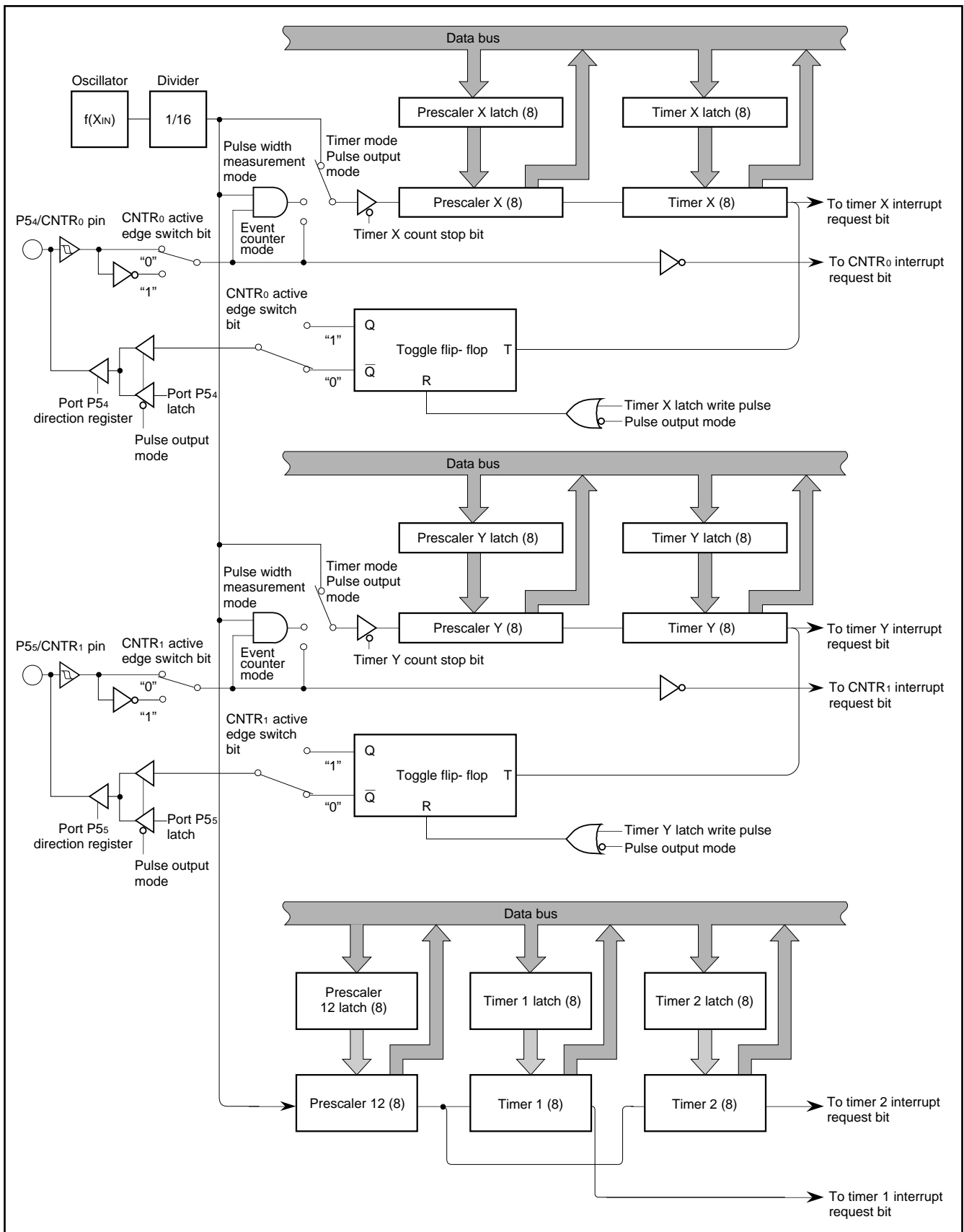


Fig. 16 Block diagram of timer X, timer Y, timer 1, and timer 2

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Serial I/O

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

### Clock synchronous serial I/O mode

Clock synchronous serial I/O mode can be selected by setting the mode selection bit of the serial I/O control register to "1". For clock synchronous serial I/O, the transmitter and the receiver must use the same clock. If an internal clock is used, transfer is started by a write signal to the TB/RB (address 0018<sub>16</sub>).

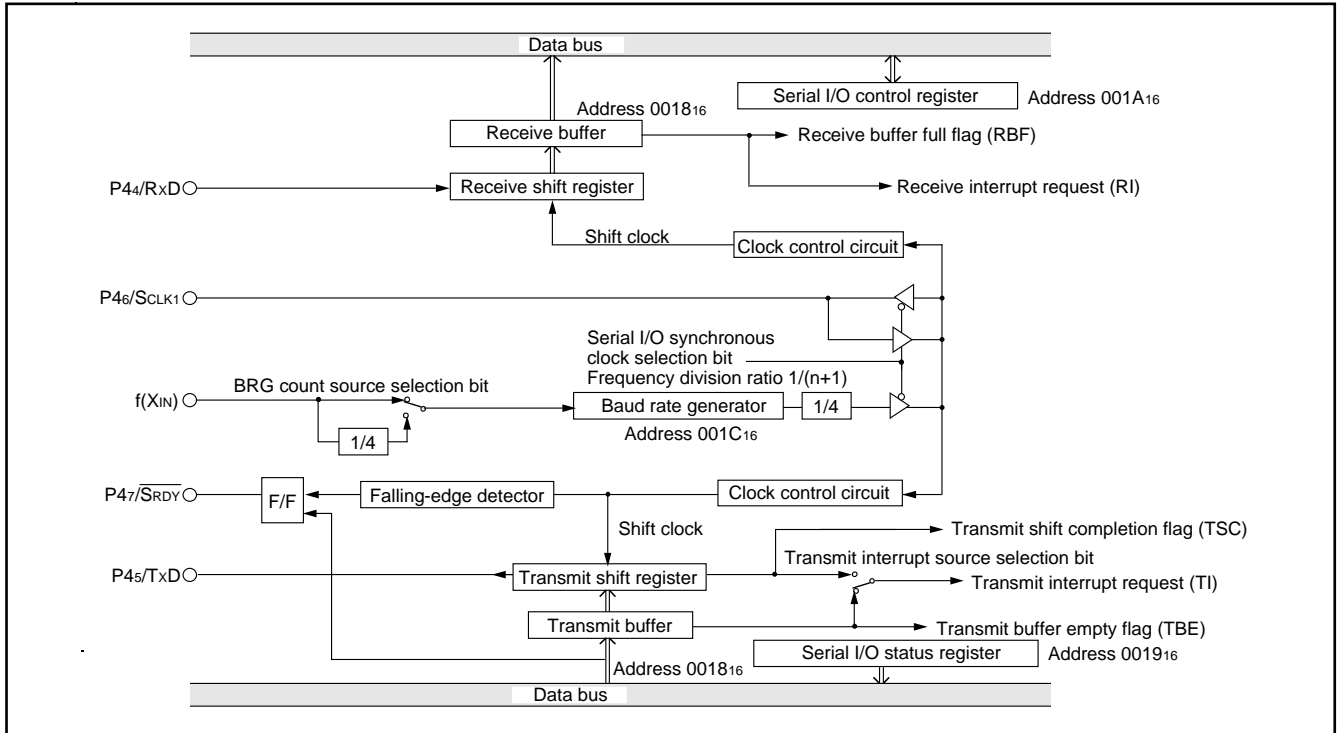


Fig. 17 Block diagram of clock synchronous serial I/O

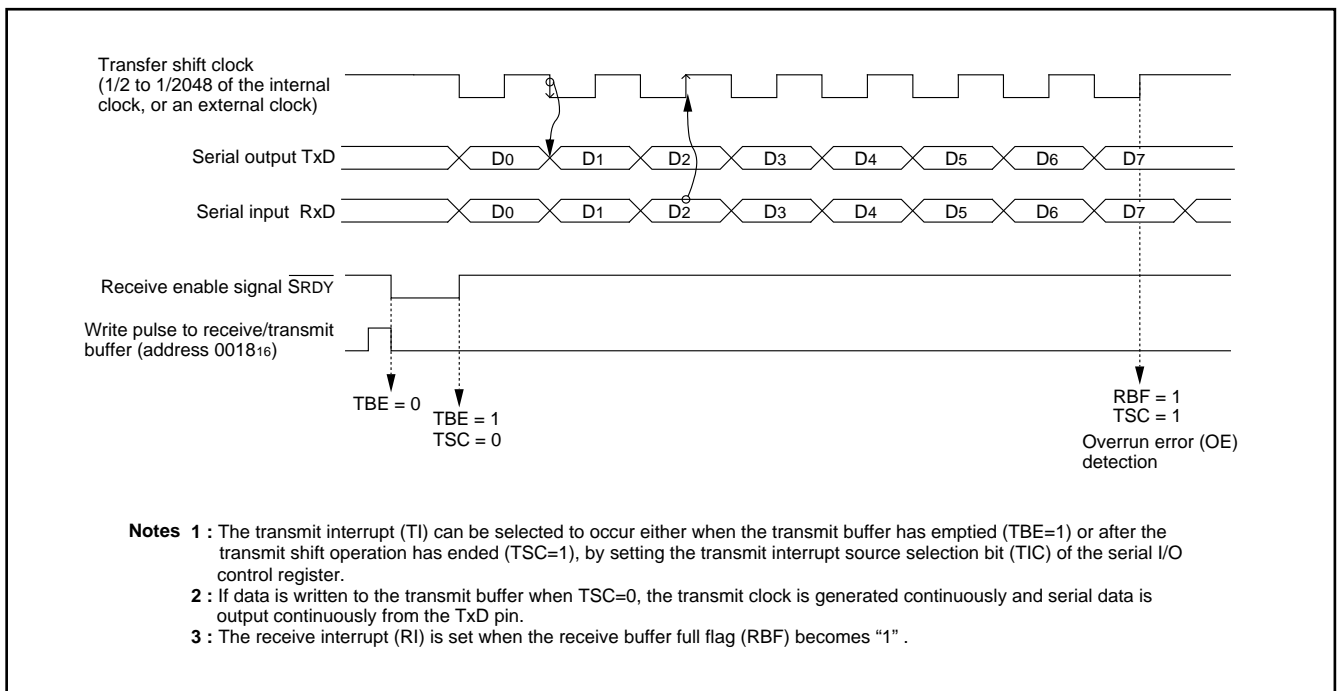


Fig. 18 Operation of clock synchronous serial I/O function

### Asynchronous serial I/O (UART) mode

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

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

The transmit and receive shift registers each have a buffer, but the

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

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

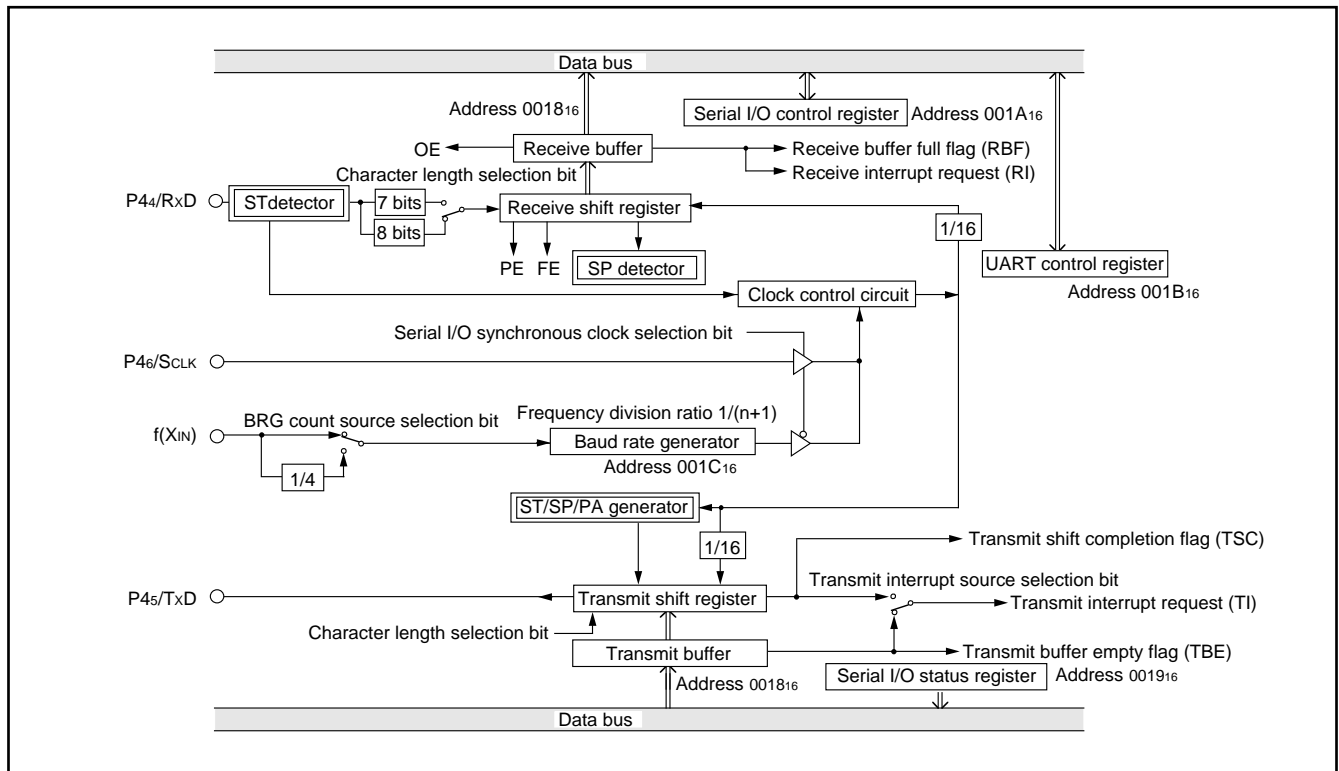


Fig. 19 Block diagram of UART serial I/O

# HARDWARE

## FUNCTIONAL DESCRIPTION

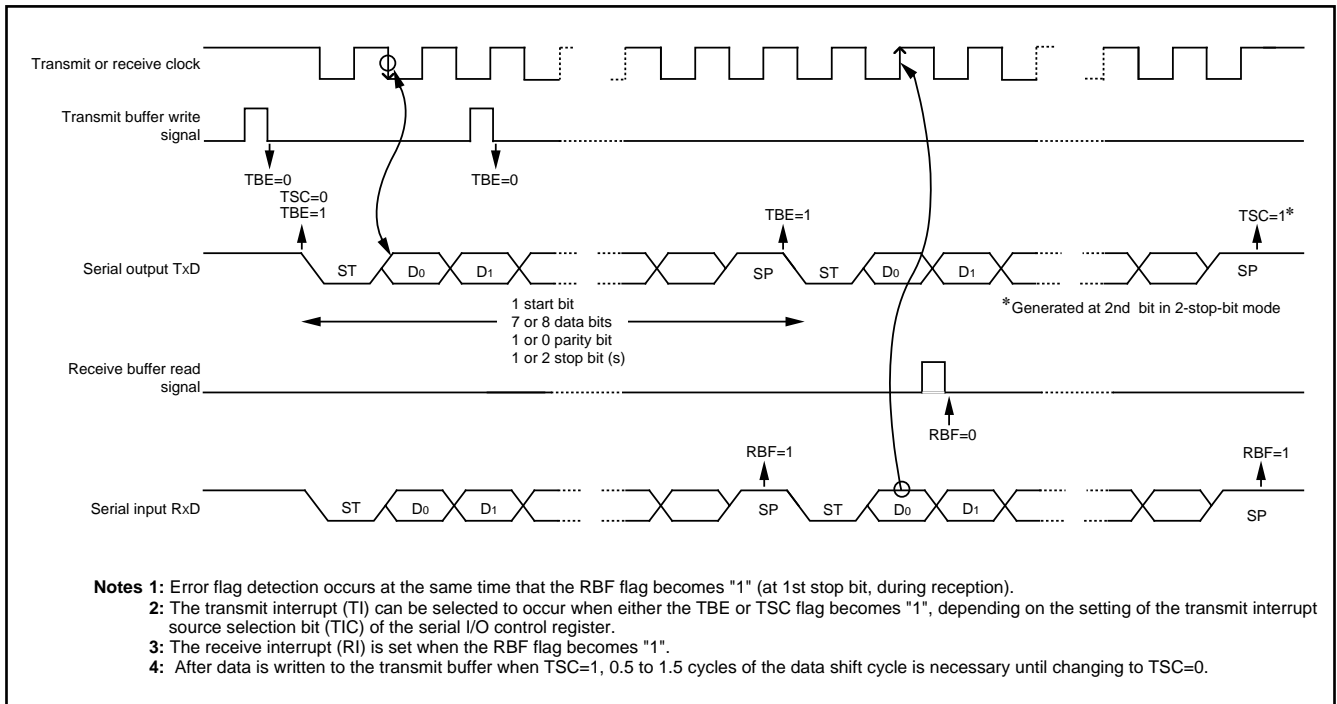


Fig. 20 Operation of UART serial I/O function

### Serial I/O control register (SIOCON) 001A16

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

### UART control register (UARTCON) 001B16

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

### Serial I/O status register (SIOSTS) 001916

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

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

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

If there is an error, it is detected at the same time that data is transferred from the receive shift register to the receive buffer, and the receive buffer full flag is set. A write to the serial I/O status register clears all the error flags OE, PE, FE, and SE (bit 3 to bit 6, re-

spectively). Writing "0" to the serial I/O enable bit SIOE (bit 7 of the Serial I/O Control Register) also clears all the status flags, including the error flags.

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

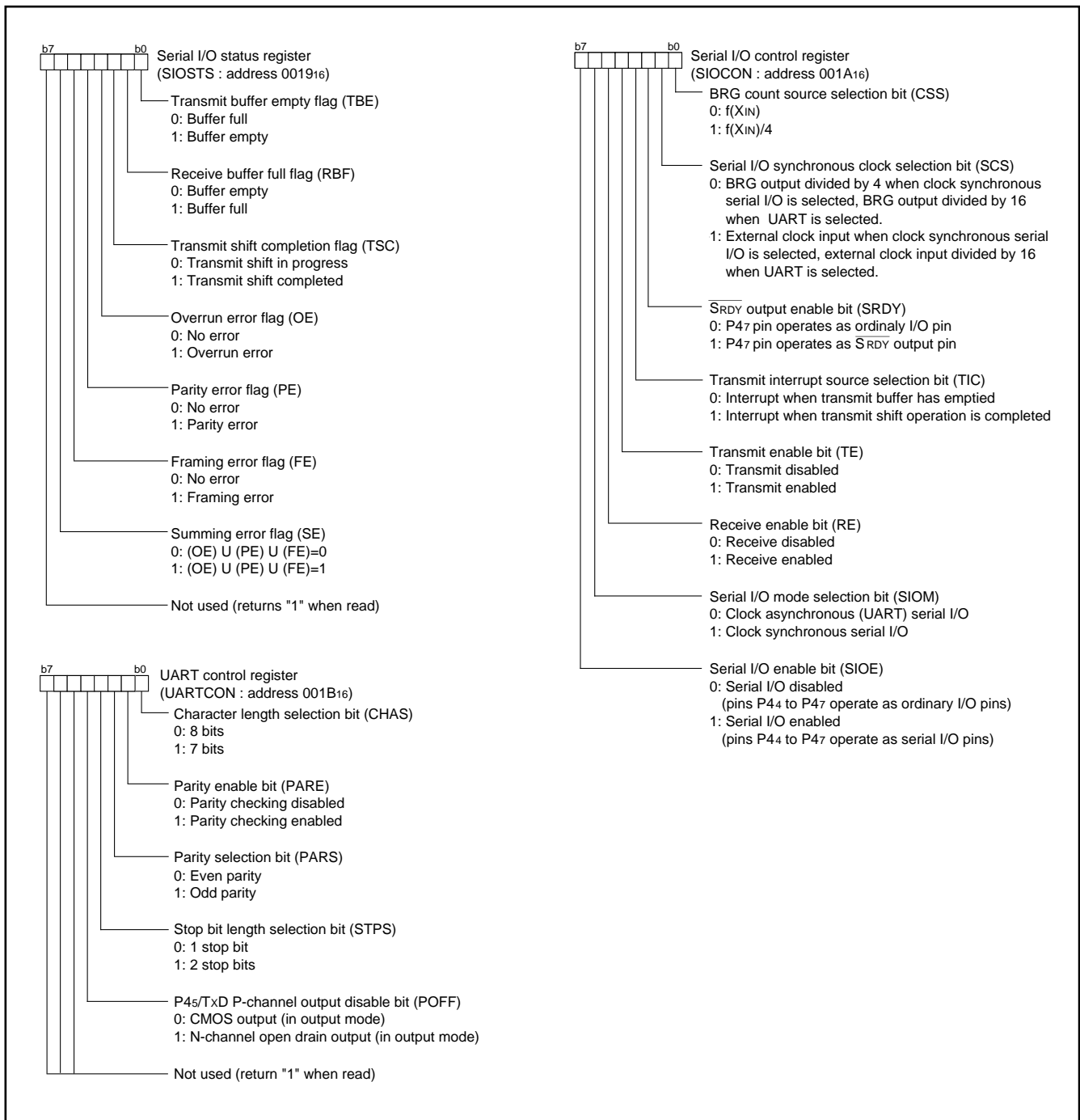
### Transmit buffer/Receive buffer register (TB/RB) 001816

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

### Baud rate generator (BRG) 001C16

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

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



**Fig. 21 Structure of serial I/O control registers**

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Reset Circuit

To reset the microcomputer, the  $\overline{\text{RESET}}$  pin should be held at an “L” level for 2  $\mu\text{s}$  or more. Then the  $\overline{\text{RESET}}$  pin is returned to an “H” level (the power source voltage should be between 3.0 V and 5.5 V, and between 4.0 V and 5.5 V for extended operating temperature version), reset is released. Internal operation does not begin until after 8 to 13  $X_{\text{IN}}$  clock cycles are completed. After the reset is completed, the program starts from the address contained in address  $\text{FFFD}_{16}$  (high-order byte) and address  $\text{FFFC}_{16}$  (low-order byte).

Make sure that the reset input voltage is less than 0.6 V for  $V_{\text{CC}}$  of 3.0 V (Extended operating temperature version: the reset input voltage is less than 0.8 V for  $V_{\text{CC}}$  of 4.0 V).

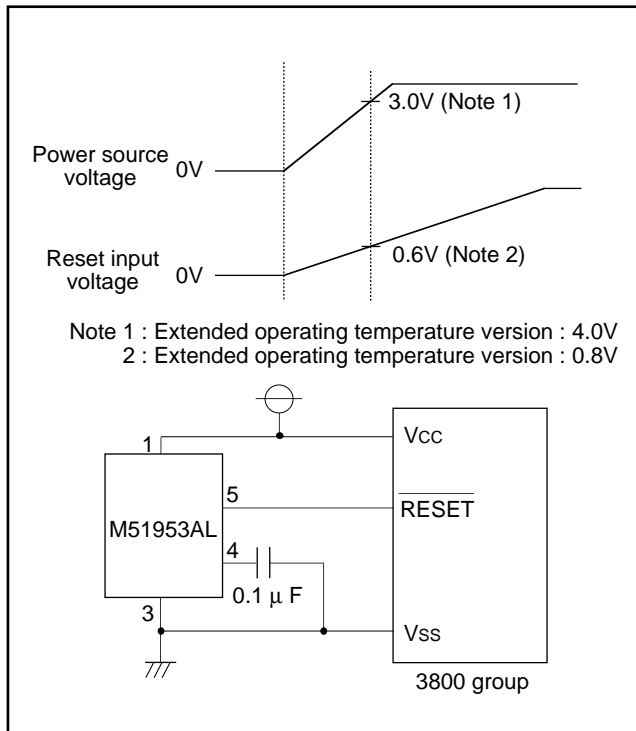


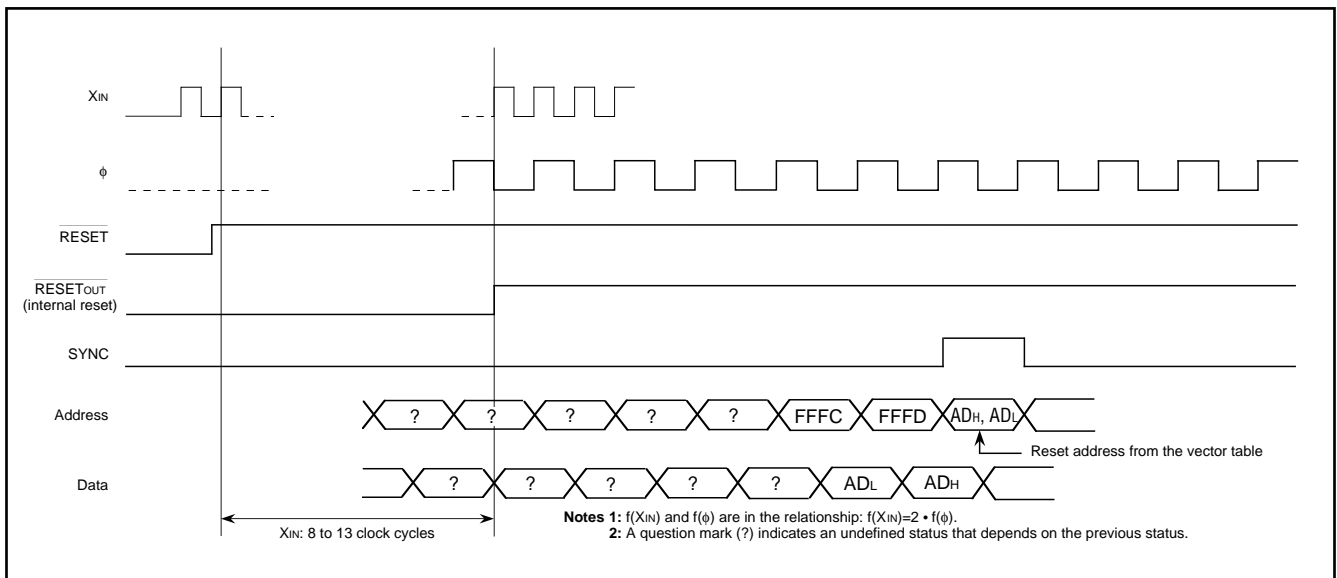
Fig. 22 Example of reset circuit

	Address	Register contents
(1) Port P0 direction register	(0001 <sub>16</sub> ) •••	00 <sub>16</sub>
(2) Port P1 direction register	(0003 <sub>16</sub> ) •••	00 <sub>16</sub>
(3) Port P2 direction register	(0005 <sub>16</sub> ) •••	00 <sub>16</sub>
(4) Port P3 direction register	(0007 <sub>16</sub> ) •••	00 <sub>16</sub>
(5) Port P4 direction register	(0009 <sub>16</sub> ) •••	00 <sub>16</sub>
(6) Port P5 direction register	(000B <sub>16</sub> ) •••	00 <sub>16</sub>
(7) Port P6 direction register	(000D <sub>16</sub> ) •••	00 <sub>16</sub>
(8) Port P7 direction register	(000F <sub>16</sub> ) •••	00 <sub>16</sub>
(9) Serial I/O status register	(0019 <sub>16</sub> ) •••	1 0 0 0 0 0 0 0
(10) Serial I/O control register	(001A <sub>16</sub> ) •••	00 <sub>16</sub>
(11) UART control register	(001B <sub>16</sub> ) •••	1 1 1 0 0 0 0 0
(12) Prescaler 12	(0020 <sub>16</sub> ) •••	FF <sub>16</sub>
(13) Timer 1	(0021 <sub>16</sub> ) •••	01 <sub>16</sub>
(14) Timer 2	(0022 <sub>16</sub> ) •••	FF <sub>16</sub>
(15) Timer XY mode register	(0023 <sub>16</sub> ) •••	00 <sub>16</sub>
(16) Prescaler X	(0024 <sub>16</sub> ) •••	FF <sub>16</sub>
(17) Timer X	(0025 <sub>16</sub> ) •••	FF <sub>16</sub>
(18) Prescaler Y	(0026 <sub>16</sub> ) •••	FF <sub>16</sub>
(19) Timer Y	(0027 <sub>16</sub> ) •••	FF <sub>16</sub>
(20) Interrupt edge selection register	(003A <sub>16</sub> ) •••	00 <sub>16</sub>
(21) CPU mode register	(003B <sub>16</sub> ) •••	0 0 0 0 0 0 0 * 0
(22) Interrupt request register 1	(003C <sub>16</sub> ) •••	00 <sub>16</sub>
(23) Interrupt request register 2	(003D <sub>16</sub> ) •••	00 <sub>16</sub>
(24) Interrupt control register 1	(003E <sub>16</sub> ) •••	00 <sub>16</sub>
(25) Interrupt control register 2	(003F <sub>16</sub> ) •••	00 <sub>16</sub>
(26) Processor status register	(PS)	x x x x x 1 x x
(27) Program counter	(PC <sub>H</sub> )	Contents of address $\text{FFFD}_{16}$
	(PC <sub>L</sub> )	Contents of address $\text{FFFC}_{16}$

Note. x : Undefined  
 \* : The initial values of  $\text{CM}_1$  are determined by the level at the  $\text{CNV}_{\text{SS}}$  pin.  
 The contents of all other registers and RAM are undefined after a reset, so they must be initialized by software.

Fig. 23 Internal status of microcomputer after reset





**Fig. 24** Timing of reset

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Clock Generating Circuit

An oscillation circuit can be formed by connecting a resonator between XIN and XOUT. To supply a clock signal externally, input it to the XIN pin and make the XOUT pin open.

### Oscillation control

#### Stop Mode

If the STP instruction is executed, the internal clock  $\phi$  stops at "H". Timer 1 is set to "0116" and prescaler 12 is set to "FF16". Oscillator restarts when an external interrupt is received, but the internal clock  $\phi$  remains at "H" until timer 1 underflows. This allows time for the clock circuit oscillation to stabilize. If oscillator is restarted by a reset, no wait time is generated, so keep the RESET pin at "L" level until oscillation has stabilized.

#### Wait Mode

If the WIT instruction is executed, the internal clock  $\phi$  stops at an "H" level, but the oscillator itself does not stop. The internal clock restarts if a reset occurs or when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted. To ensure that interrupts will be received to release the STP or WIT state, interrupt enable bits must be set to "1" before the STP or WIT instruction is executed.

When the STP status is released, prescaler 12 and timer 1 will start counting and reset will not be released until timer 1 underflows, so set the timer 1 interrupt enable bit to "0" before the STP instruction is executed.

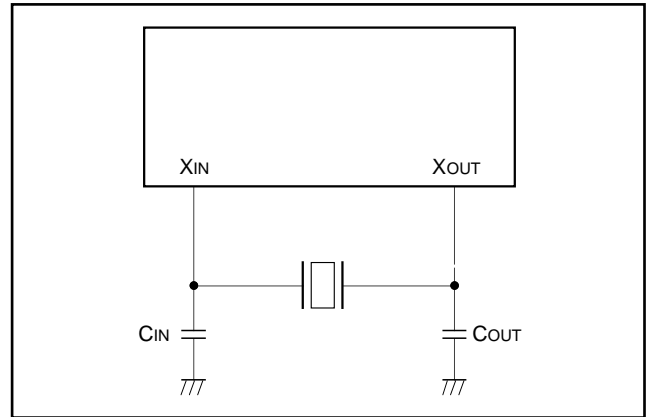


Fig. 25 Ceramic resonator circuit

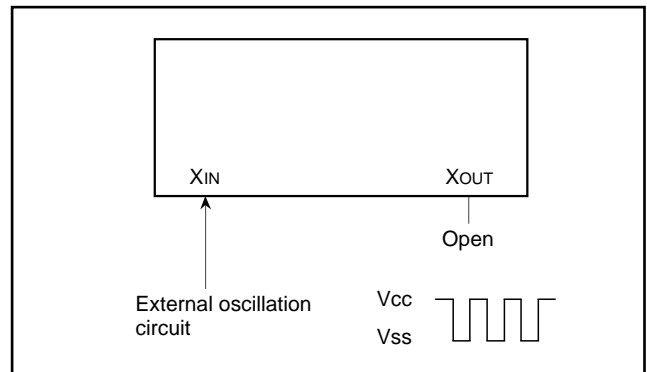


Fig. 26 External clock input circuit

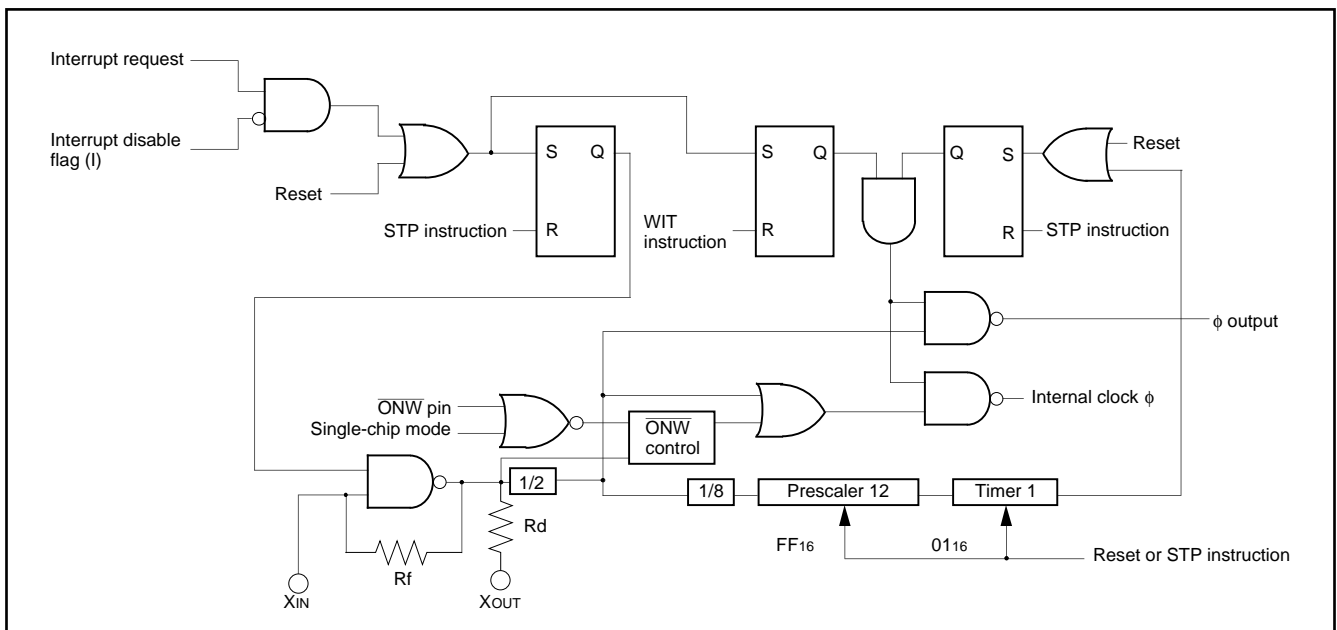


Fig. 27 Block diagram of clock generating circuit

### Processor Modes

Single-chip mode, memory expansion mode, and microprocessor mode can be selected by changing the contents of the processor mode bits CM0 and CM1 (bits 0 and 1 of address 003B<sub>16</sub>). In memory expansion mode and microprocessor mode, memory can be expanded externally through ports P0 to P3. In these modes, ports P0 to P3 lose their I/O port functions and become bus pins.

**Table 8. Functions of ports in memory expansion mode and microprocessor mode**

Port Name	Function
Port P0	Outputs low-order byte of address.
Port P1	Outputs high-order byte of address.
Port P2	Operates as I/O pins for data D7 to D0 (including instruction codes).
Port P3	P30 and P31 function only as output pins (except that the port latch cannot be read). P32 is the $\overline{ONW}$ input pin. P33 is the $\overline{RESETOUT}$ output pin. (Note) P34 is the $\phi$ output pin. P35 is the SYNC output pin. P36 is the $\overline{WR}$ output pin, and P37 is the $\overline{RD}$ output pin.

**Note:** If CNVss is connected to Vss, the microcomputer goes to single-chip mode after a reset, so this pin cannot be used as the  $\overline{RESETOUT}$  output pin.

### Single-Chip Mode

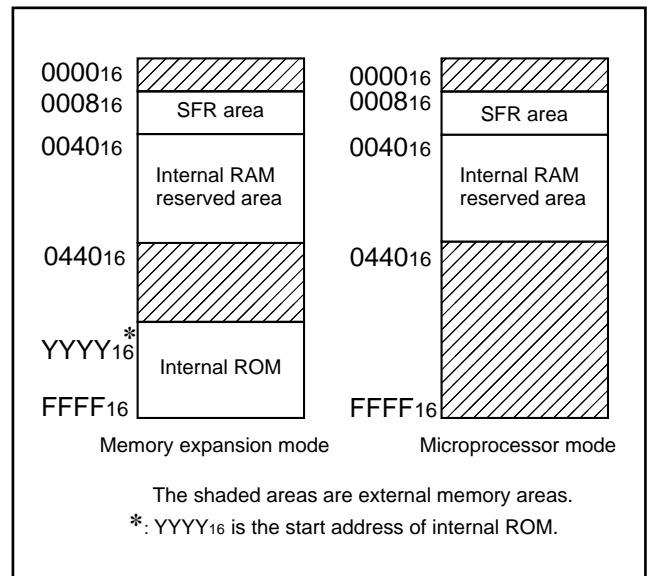
Select this mode by resetting the microcomputer with CNVss connected to Vss.

### Memory Expansion Mode

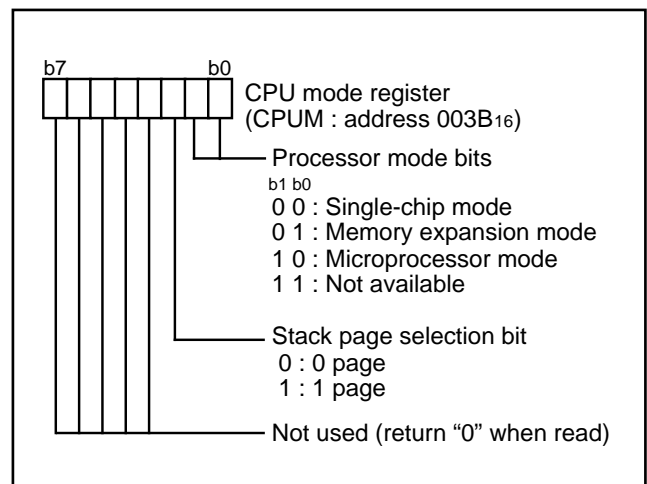
Select this mode by setting the processor mode bits to "01" in software with CNVss connected to Vss. This mode enables external memory expansion while maintaining the validity of the internal ROM. Internal ROM will take precedence over external memory if addresses conflict.

### Microprocessor Mode

Select this mode by resetting the microcomputer with CNVss connected to Vcc, or by setting the processor mode bits to "10" in software with CNVss connected to Vss. In microprocessor mode, the internal ROM is no longer valid and external memory must be used.



**Fig. 28 Memory maps in various processor modes**



**Fig. 29 Structure of CPU mode register**

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Bus control with memory expansion

The 3800 group has a built-in  $\overline{ONW}$  function to facilitate access to external memory and I/O devices in memory expansion mode or microprocessor mode.

If an "L" level signal is input to the  $\overline{ONW}$  pin when the CPU is in a read or write state, the corresponding read or write cycle is extended by one cycle of  $\phi$ . During this extended period, the  $\overline{RD}$  or  $\overline{WR}$  signal remains at "L". This extension period is valid only for writing to and reading from addresses 0000<sub>16</sub> to 0007<sub>16</sub> and 0440<sub>16</sub> to FFFF<sub>16</sub> in microprocessor mode, 0040<sub>16</sub> to YYYYY<sub>16</sub> in memory expansion mode, and only read and write cycles are extended.

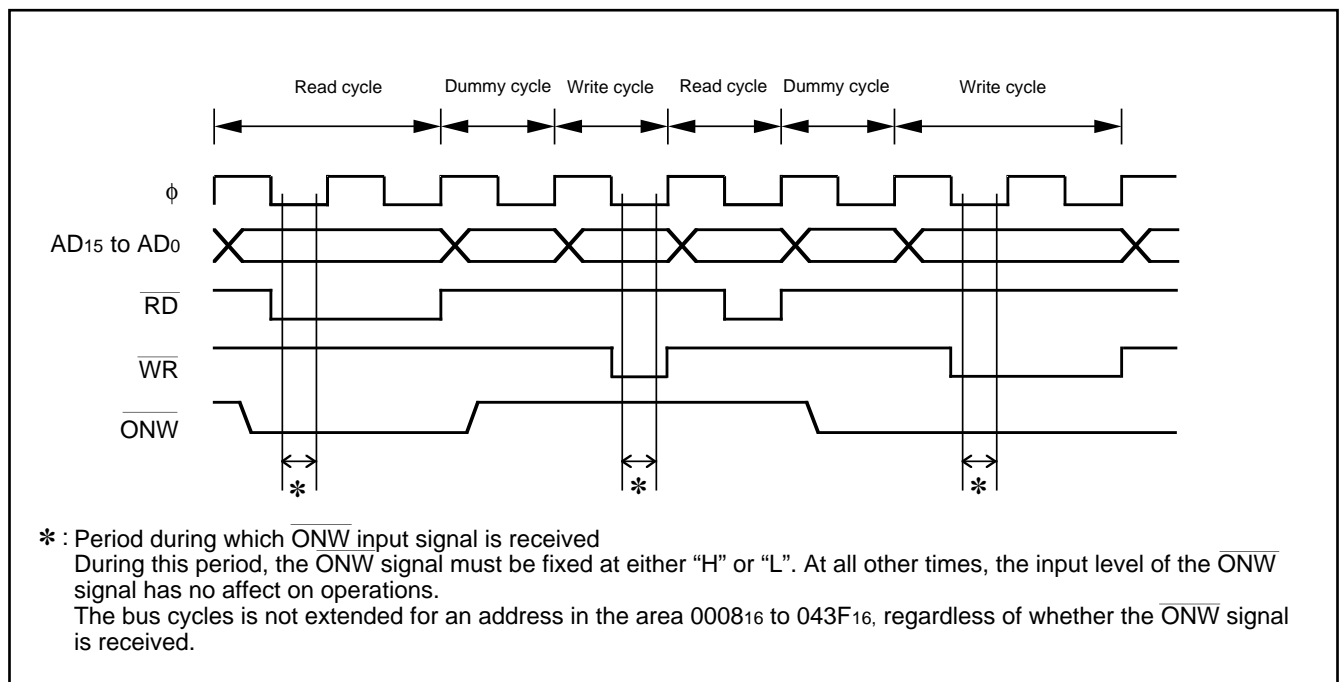


Fig. 30  $\overline{ONW}$  function timing

### NOTES ON PROGRAMMING

#### Processor Status Register

The contents of the processor status register (PS) after a reset are undefined, except for the interrupt disable flag (I) which is "1". After a reset, initialize flags which affect program execution.

In particular, it is essential to initialize the index X mode (T) and the decimal mode (D) flags because of their effect on calculations.

#### Interrupts

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

#### Decimal Calculations

To calculate in decimal notation, set the decimal mode flag (D) to "1", then execute an ADC or SBC instruction. Only the ADC and SBC instructions yield proper decimal results. After executing an ADC or SBC instruction, execute at least one instruction before executing a SEC, CLC, or CLD instruction.

In decimal mode, the values of the negative (N), overflow (V), and zero (Z) flags are invalid.

The carry flag can be used to indicate whether a carry or borrow has occurred. Initialize the carry flag before each calculation. Clear the carry flag before an ADC and set the flag before an SBC.

#### Timers

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

#### Multiplication and Division Instructions

The index X mode (T) and the decimal mode (D) flags do not affect the MUL and DIV instruction.

The execution of these instructions does not change the contents of the processor status register.

#### Ports

The contents of the port direction registers cannot be read.

The following cannot be used:

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

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

#### Serial I/O

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

Serial I/O continues to output the final bit from the TxD pin after transmission is completed.

#### Instruction Execution Time

The instruction execution time is obtained by multiplying the frequency of the internal clock  $\phi$  by the number of cycles needed to execute an instruction.

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

The frequency of the internal clock  $\phi$  is half of the XIN frequency.

When the  $\overline{\text{ONW}}$  function is used in modes other than single-chip mode, the frequency of the internal clock  $\phi$  may be one fourth the XIN frequency.

#### Memory Expansion Mode and Microprocessor Mode

Execute the LDM or STA instruction for writing to port P3 (address 000616) in memory expansion mode and microprocessor mode.

Set areas which can be read out and write to port P3 (address 000616) in a memory, using the read-modify-write instruction (SEB, CLB).

# HARDWARE

## DATA REQUIRED FOR MASK ORDERS/ROM PROGRAMMING METHOD

### DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

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

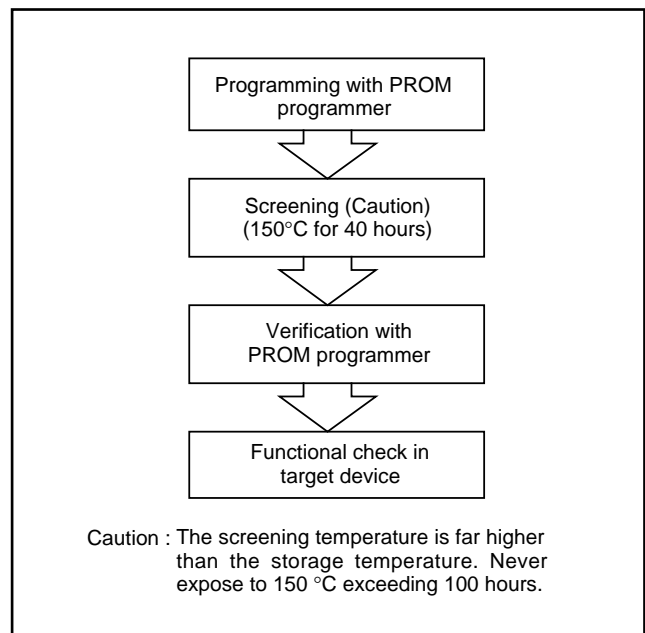
### ROM PROGRAMMING METHOD

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

**Table 9. Programming adapter**

Package	Name of Programming Adapter
64P4B, 64S1B	PCA4738S-64A
64P6N-A	PCA4738F-64A
64D0	PCA4738L-64A

The PROM of the blank One Time PROM version is not tested or screened in the assembly process and following processes. To ensure proper operation after programming, the procedure shown in Figure 31 is recommended to verify programming.



**Fig. 31 Programming and testing of One Time PROM version**

### FUNCTIONAL DESCRIPTION SUPPLEMENT Interrupt

3800 group permits interrupts on the basis of 15 sources. It is vector interrupts with a fixed priority system. Accordingly, when two or more interrupt requests occur during the same sampling, the higher-priority interrupt is accepted first. This priority is determined

by hardware, but variety of priority processing can be performed by software, using an interrupt enable bit and an interrupt disable flag.

For interrupt sources, vector addresses and interrupt priority, refer to “Table 10.”

**Table 10. Interrupt sources, vector addresses and interrupt priority**

Priority	Interrupt sources	Vector addresses		Remarks
		High-order	Low-order	
1	Reset (Note)	FFFD <sub>16</sub>	FFFC <sub>16</sub>	Non-maskable
2	INT <sub>0</sub> interrupt	FFFB <sub>16</sub>	FFFA <sub>16</sub>	External interrupt (active edge selectable)
3	INT <sub>1</sub> interrupt	FFF9 <sub>16</sub>	FFF8 <sub>16</sub>	External interrupt (active edge selectable)
4	Serial I/O receive interrupt	FFF7 <sub>16</sub>	FFF6 <sub>16</sub>	Valid when serial I/O is selected
5	Serial I/O transmit interrupt	FFF5 <sub>16</sub>	FFF4 <sub>16</sub>	Valid when serial I/O is selected
6	Timer X interrupt	FFF3 <sub>16</sub>	FFF2 <sub>16</sub>	
7	Timer Y interrupt	FFF1 <sub>16</sub>	FFF0 <sub>16</sub>	
8	Timer 1 interrupt	FFEF <sub>16</sub>	FFEE <sub>16</sub>	STP release timer underflow
9	Timer 2 interrupt	FFED <sub>16</sub>	FFEC <sub>16</sub>	
10	CNTR <sub>0</sub> interrupt	FFEB <sub>16</sub>	FFEA <sub>16</sub>	External interrupt (active edge selectable)
11	CNTR <sub>1</sub> interrupt	FFE9 <sub>16</sub>	FFE8 <sub>16</sub>	External interrupt (active edge selectable)
12	INT <sub>2</sub> interrupt	FFE7 <sub>16</sub>	FFE6 <sub>16</sub>	External interrupt (active edge selectable)
13	INT <sub>3</sub> interrupt	FFE5 <sub>16</sub>	FFE4 <sub>16</sub>	External interrupt (active edge selectable)
14	INT <sub>4</sub> interrupt	FFE3 <sub>16</sub>	FFE2 <sub>16</sub>	External interrupt (active edge selectable)
15	INT <sub>5</sub> interrupt	FFE1 <sub>16</sub>	FFE0 <sub>16</sub>	External interrupt (active edge selectable)
16	BRK instruction interrupt	FFDD <sub>16</sub>	FFDC <sub>16</sub>	Non-maskable software interrupt

**Note:** Reset functions in the same way as an interrupt with the highest priority.

# HARDWARE

## FUNCTIONAL DESCRIPTION SUPPLEMENT

### Timing After Interrupt

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

Figure 32 shows a timing chart after an interrupt occurs, and Figure 33 shows the time up to execution of the interrupt processing routine.

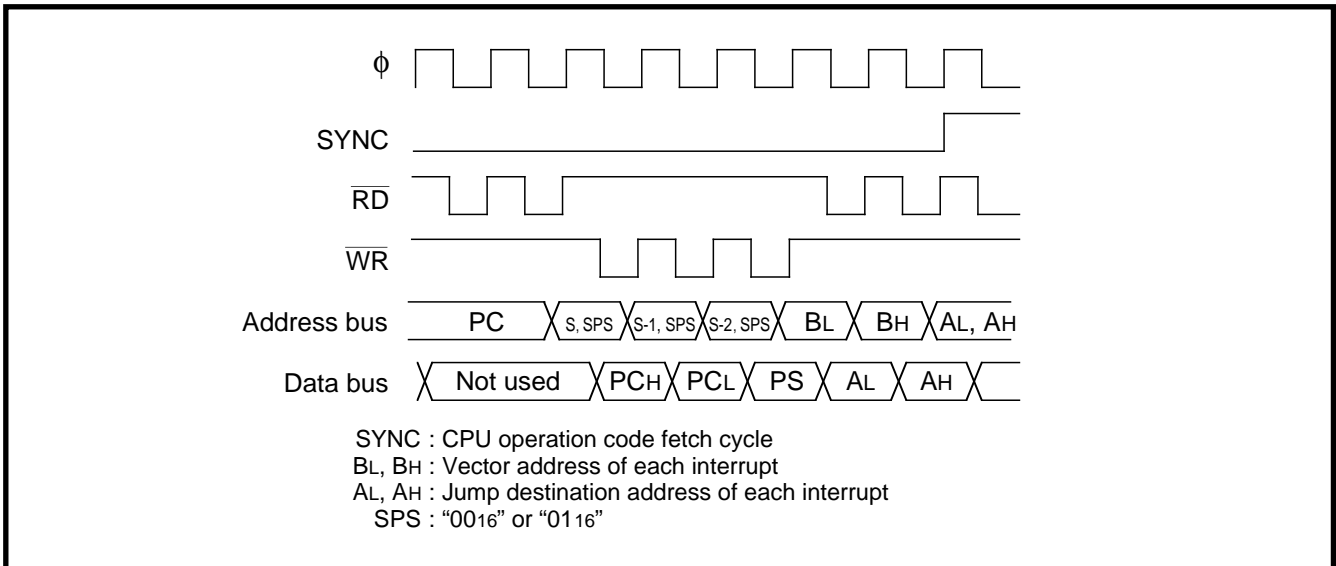


Fig. 32 Timing chart after an interrupt occurs

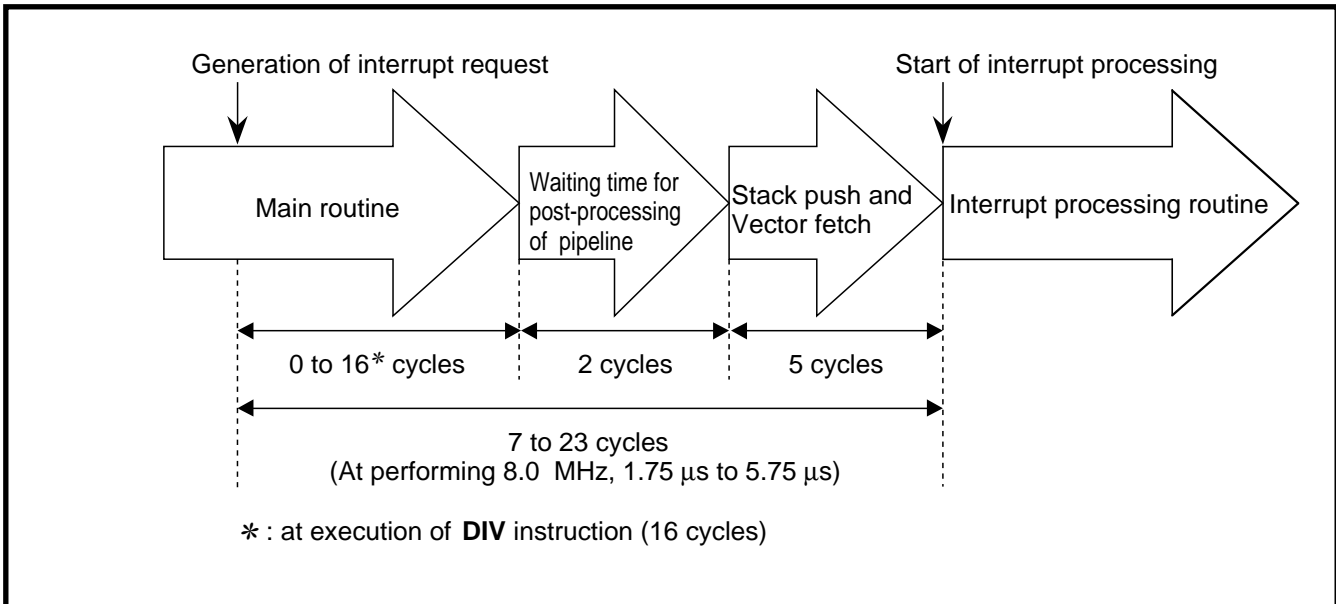


Fig. 33 Time up to execution of the interrupt processing routine





# CHAPTER 2

## **APPLICATION**

- 2.1 I/O port
- 2.2 Timer
- 2.3 Serial I/O
- 2.4 Processor mode
- 2.5 Reset

# APPLICATION

## 2.1 I/O port

---

### 2.1 I/O port

#### 2.1.1 Memory map of I/O port

0000 <sub>16</sub>	Port P0 (P0)
0001 <sub>16</sub>	Port P0 direction register (P0D)
0002 <sub>16</sub>	Port P1 (P1)
0003 <sub>16</sub>	Port P1 direction register (P1D)
0004 <sub>16</sub>	Port P2 (P2)
0005 <sub>16</sub>	Port P2 direction register (P2D)
0006 <sub>16</sub>	Port P3 (P3)
0007 <sub>16</sub>	Port P3 direction register (P3D)
0008 <sub>16</sub>	Port P4 (P4)
0009 <sub>16</sub>	Port P4 direction register (P4D)
000A <sub>16</sub>	Port P5 (P5)
000B <sub>16</sub>	Port P5 direction register (P5D)
000C <sub>16</sub>	Port P6 (P6)
000D <sub>16</sub>	Port P6 direction register (P6D)
000E <sub>16</sub>	Port P7 (P7)
000F <sub>16</sub>	Port P7 direction register (P7D)

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

### 2.1.2 Related registers

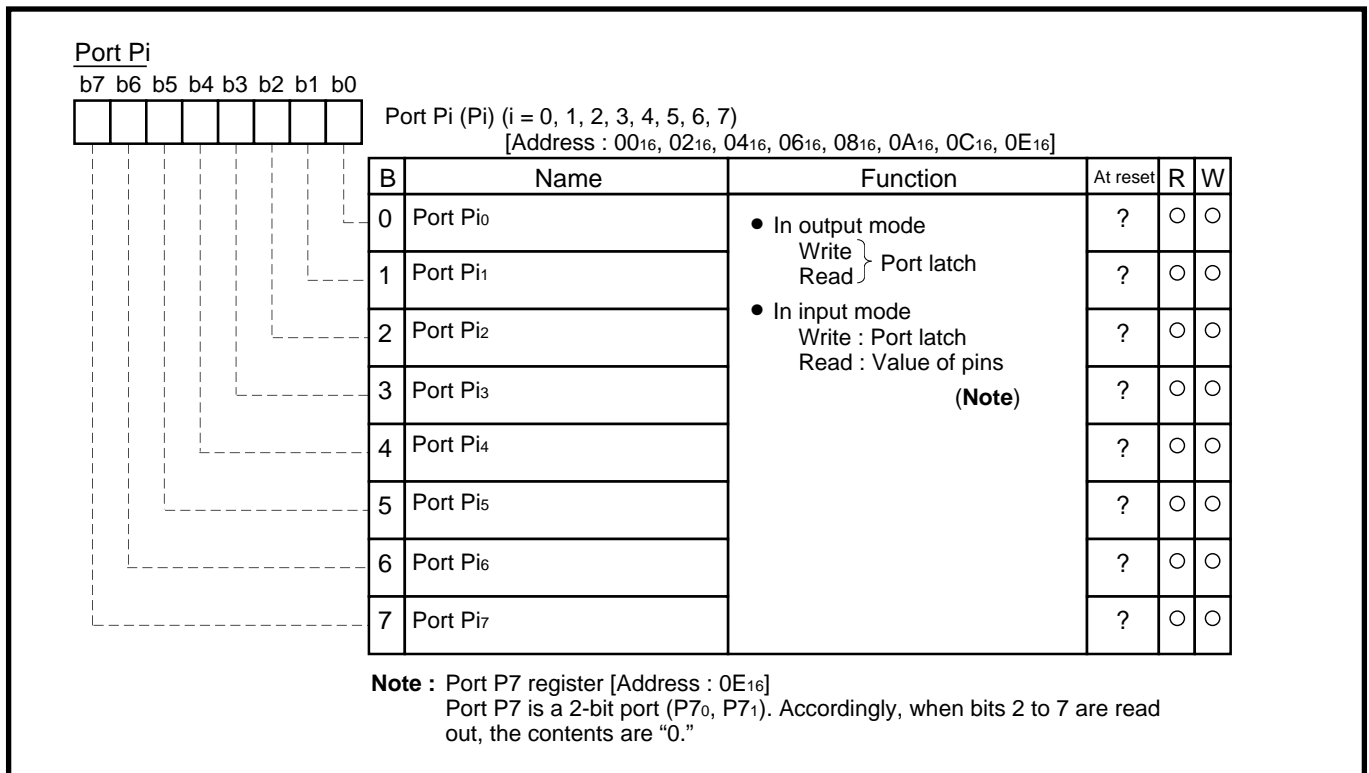


Fig. 2.1.2 Structure of Port Pi (i = 0, 1, 2, 3, 4, 5, 6, 7)

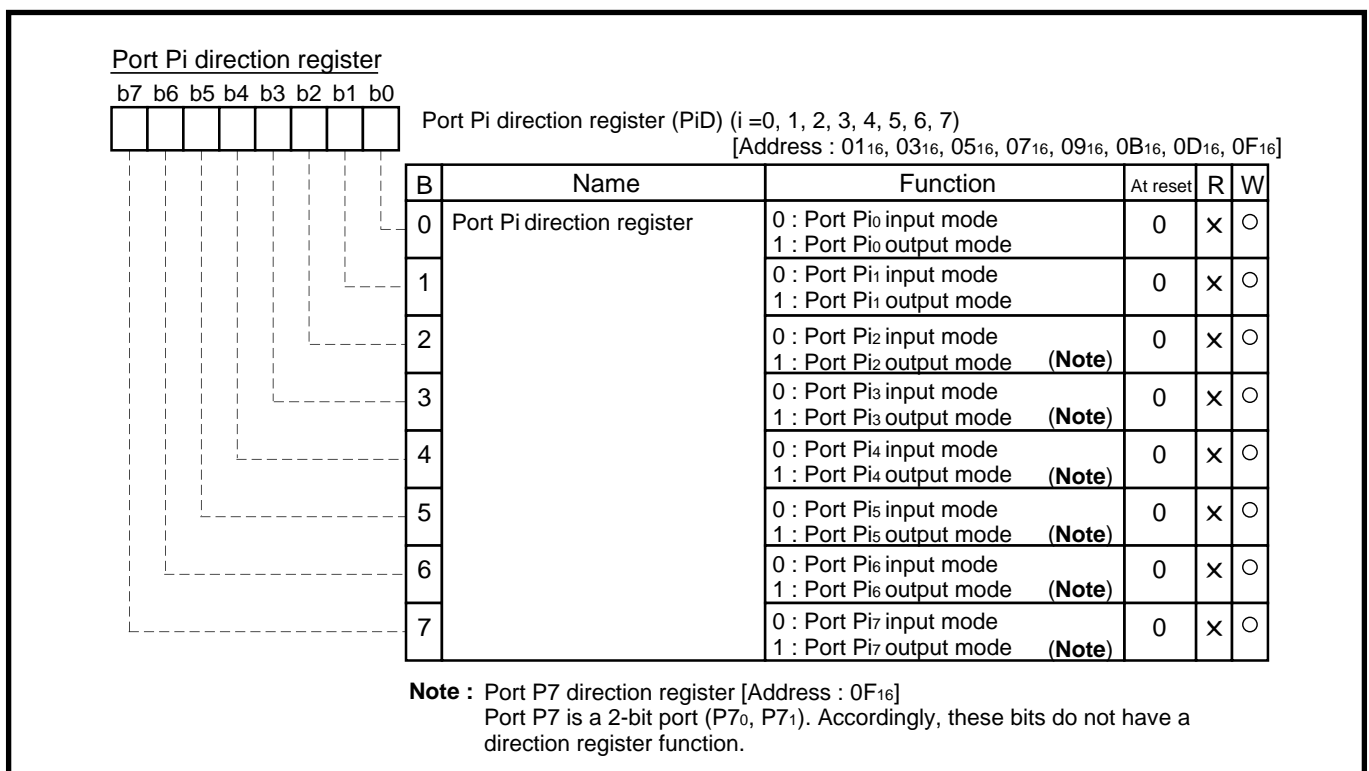


Fig. 2.1.3 Structure of Port Pi direction register (i = 0, 1, 2, 3, 4, 5, 6, 7)

# APPLICATION

## 2.1 I/O port

### 2.1.3 Handling of unused pins

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

Name of Pins/Ports	Handling
P0, P1, P2, P3, P4, P5, P6, P7	<ul style="list-style-type: none"><li>• Set to the input mode and connect to VCC or VSS through a resistor of 1 k<math>\Omega</math> to 10 k<math>\Omega</math>.</li><li>• Set to the output mode and open at "L" or "H."</li></ul>
XOUT	Open (only when using external clock).

**Table 2.1.2 Handling of unused pins (in memory expansion mode and microprocessor mode)**

Name of Pins/Ports	Handling
P30, P31	Open
P4, P5, P6, P7	<ul style="list-style-type: none"><li>• Set to the input mode and connect to VCC or VSS through a resistor of 1 k<math>\Omega</math> to 10 k<math>\Omega</math>.</li><li>• Set to the output mode and open at "L" or "H."</li></ul>
ONW	Connect to VCC through a resistor of 1 k $\Omega$ to 10 k $\Omega$ .
RESETOUT	Open
$\phi$	Open
SYNC	Open
XOUT	Open (only when using external clock).

## 2.2 Timer

### 2.2.1 Memory map of timer

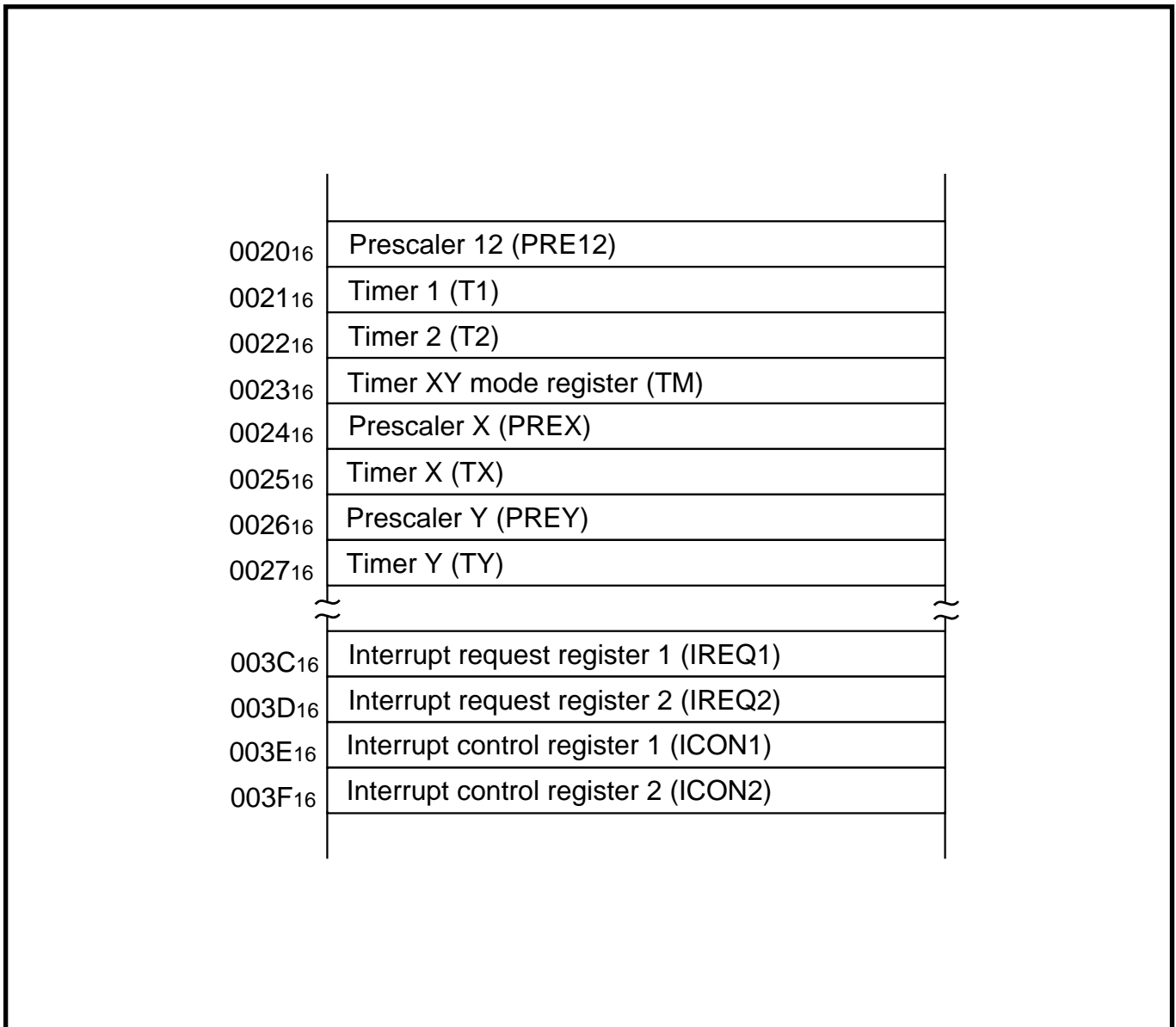


Fig. 2.2.1 Memory map of timer related registers

# APPLICATION

## 2.2 Timer

### 2.2.2 Related registers

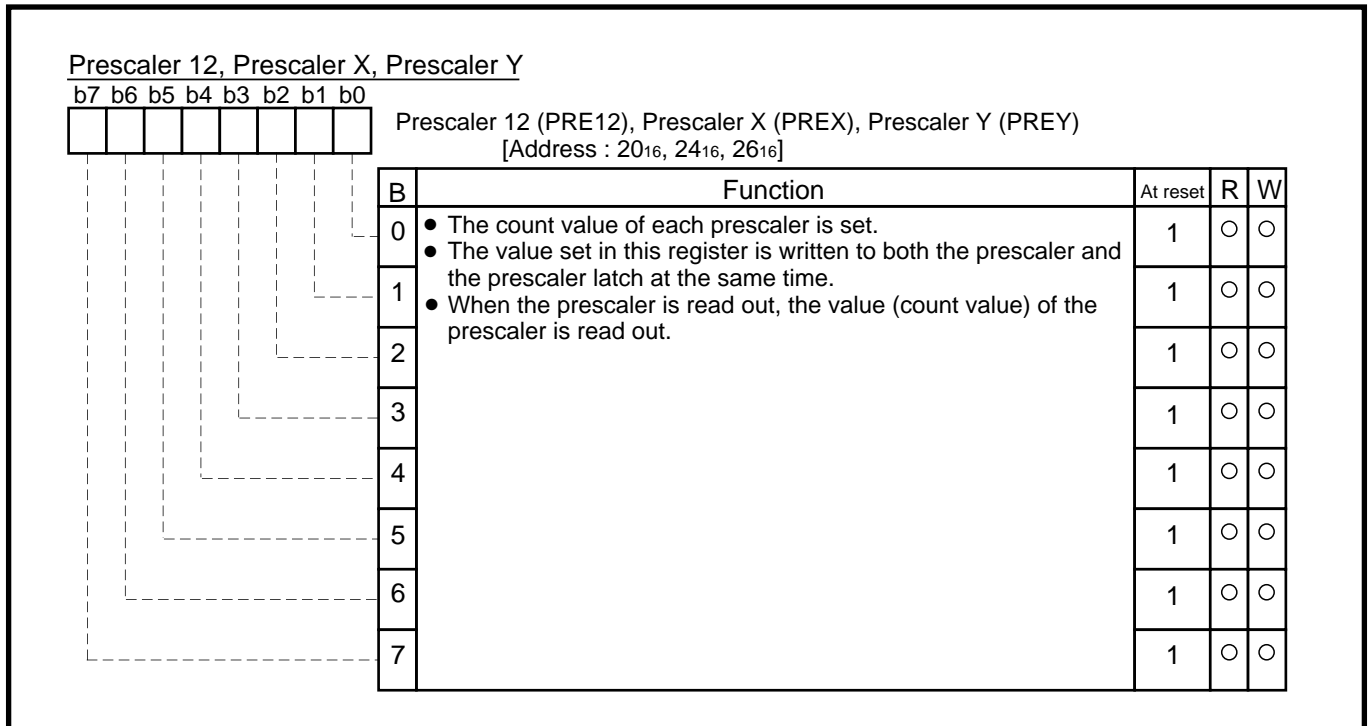


Fig. 2.2.2 Structure of Prescaler 12, Prescaler X, Prescaler Y

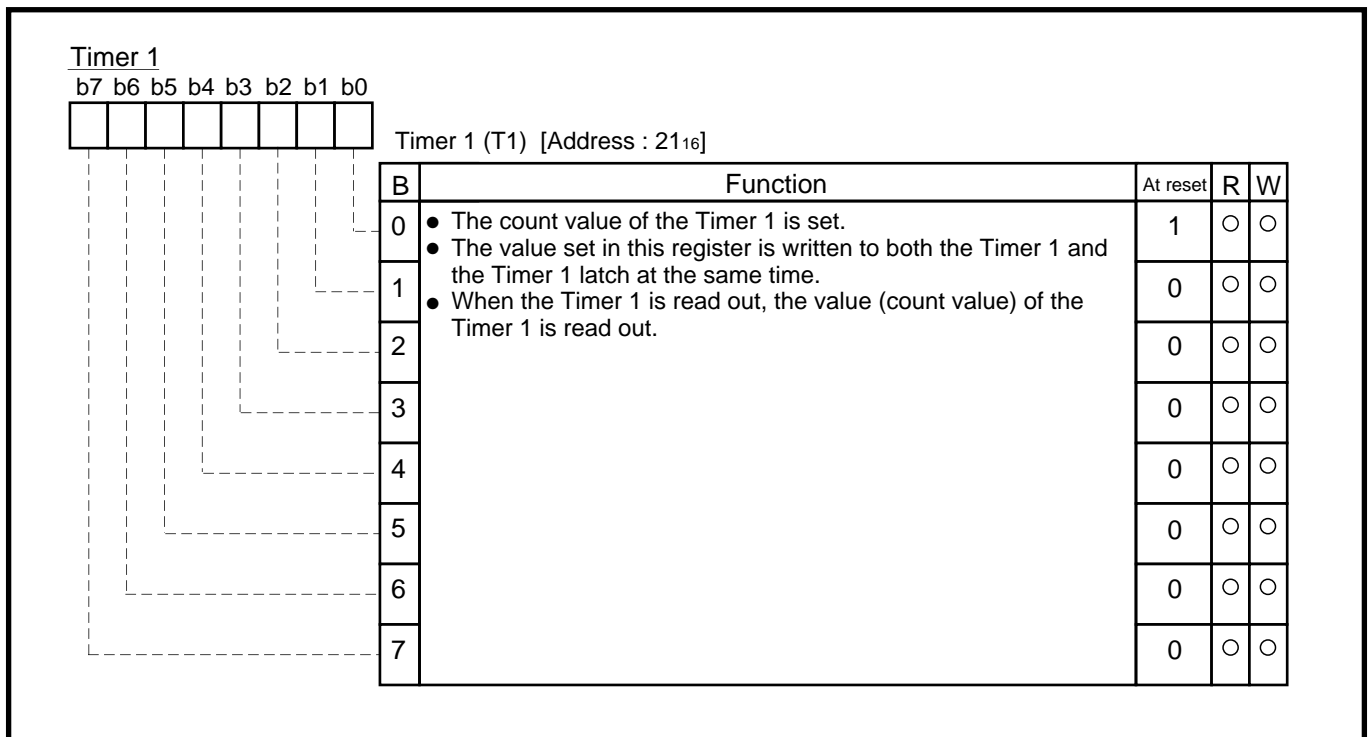


Fig. 2.2.3 Structure of Timer 1

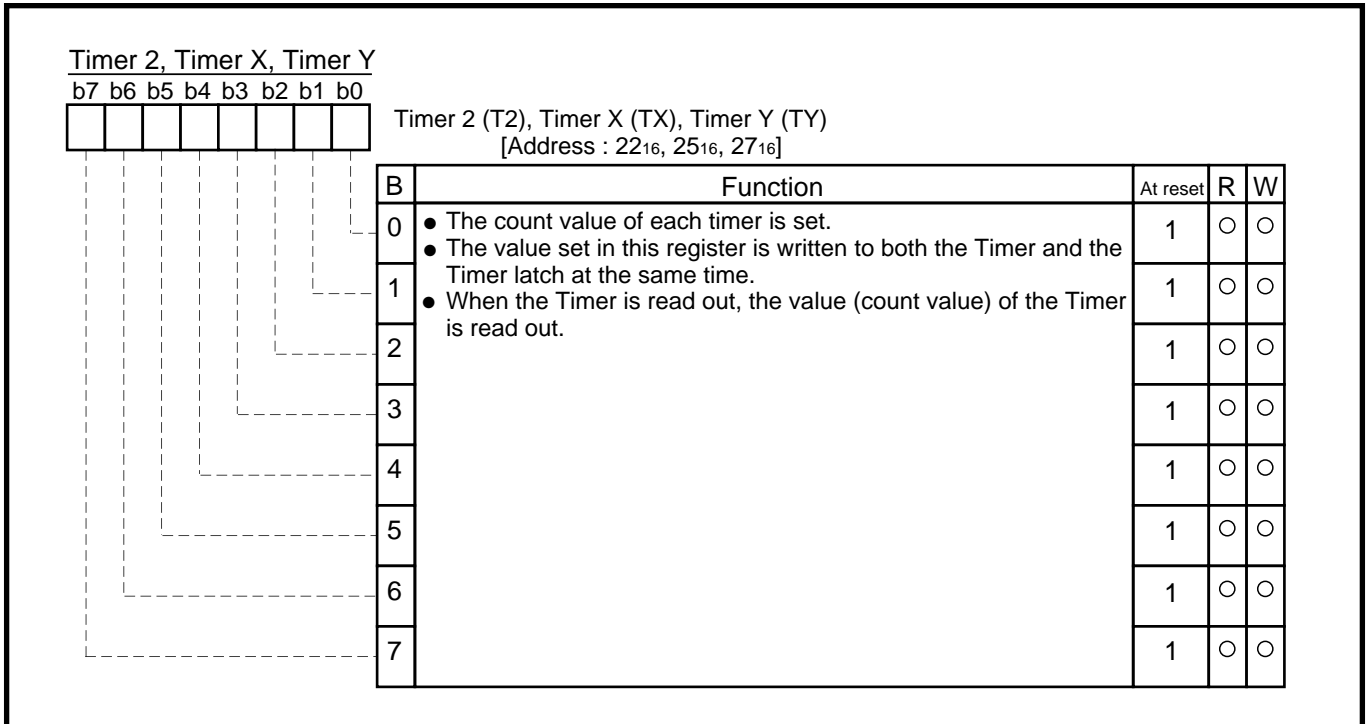


Fig. 2.2.4 Structure of Timer 2, Timer X, Timer Y

# APPLICATION

## 2.2 Timer

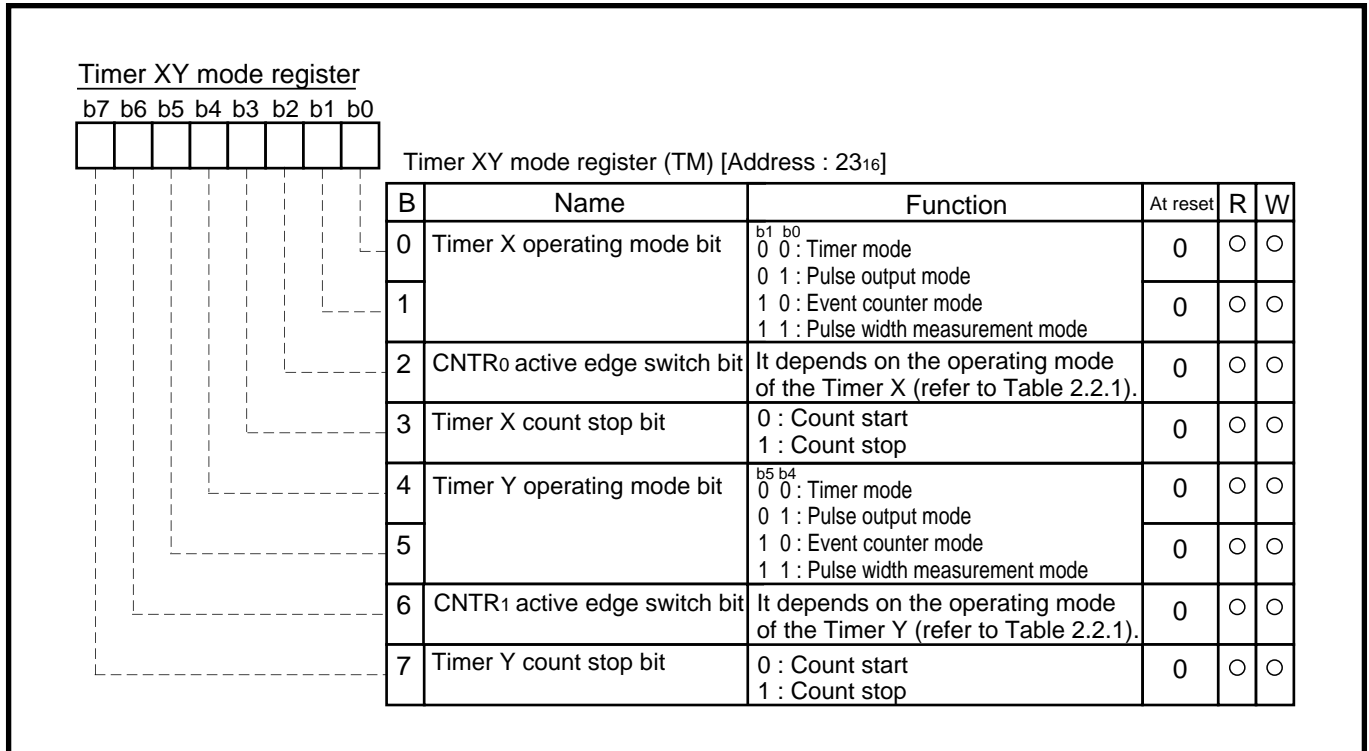


Fig. 2.2.5 Structure of Timer XY mode register

Table. 2.2.1 Function of CNTR0/CNTR1 edge switch bit

Operating mode of Timer X/Timer Y	Function of CNTR0/CNTR1 edge switch bit (bits 2 and 6)	
Timer mode	"0"	• Generation of CNTR0/CNTR1 interrupt request : Falling edge (No effect on timer count)
	"1"	• Generation of CNTR0/CNTR1 interrupt request : Rising edge (No effect on timer count)
Pulse output mode	"0"	• Start of pulse output : From "H" level • Generation of CNTR0/CNTR1 interrupt request : Falling edge
	"1"	• Start of pulse output : From "L" level • Generation of CNTR0/CNTR1 interrupt request : Rising edge
Event counter mode	"0"	• Timer X/Timer Y : Count of rising edge • Generation of CNTR0/CNTR1 interrupt request : Falling edge
	"1"	• Timer X/Timer Y : Count of falling edge • Generation of CNTR0/CNTR1 interrupt request : Rising edge
Pulse width measurement mode	"0"	• Timer X/Timer Y : Measurement of "H" level width • Generation of CNTR0/CNTR1 interrupt request : Falling edge
	"1"	• Timer X/Timer Y : Measurement of "L" level width • Generation of CNTR0/CNTR1 interrupt request : Rising edge



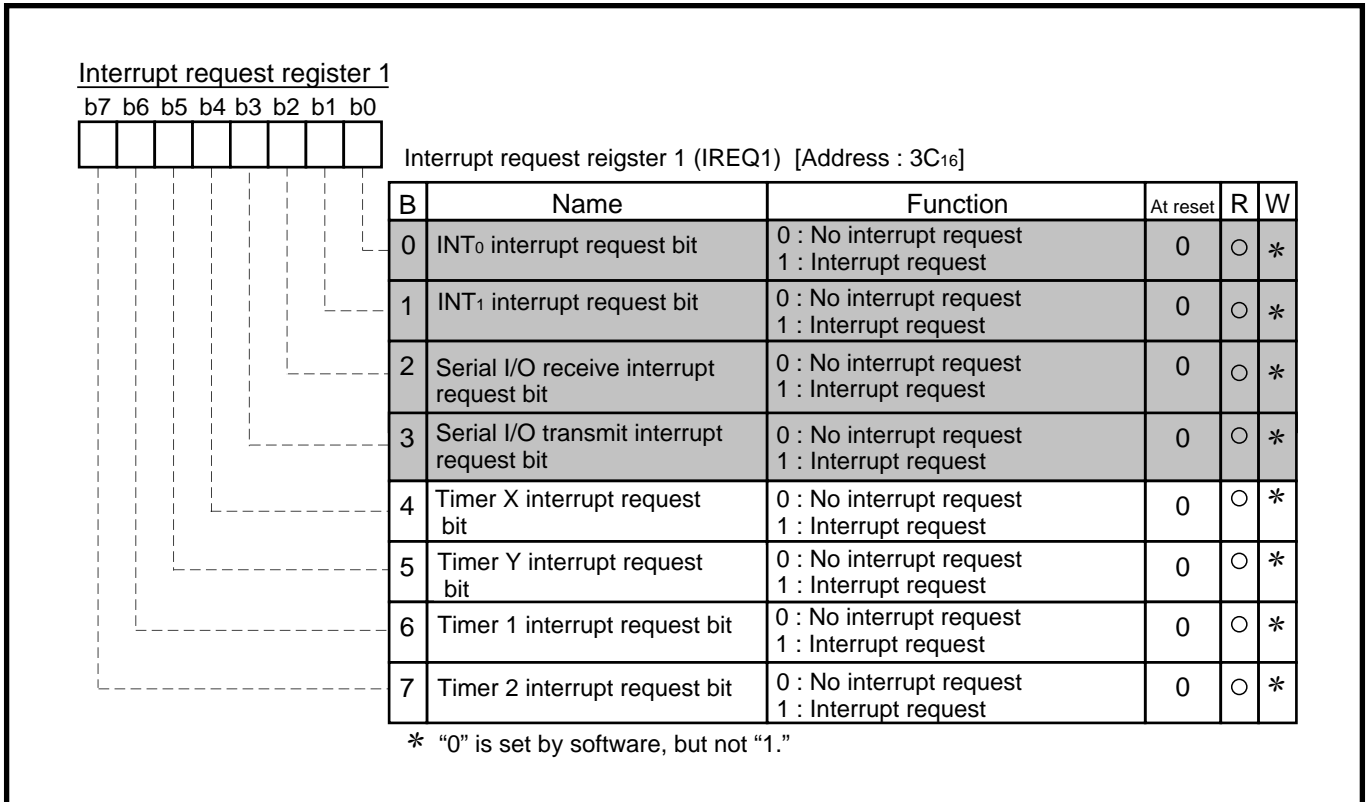


Fig. 2.2.6 Structure of Interrupt request register 1

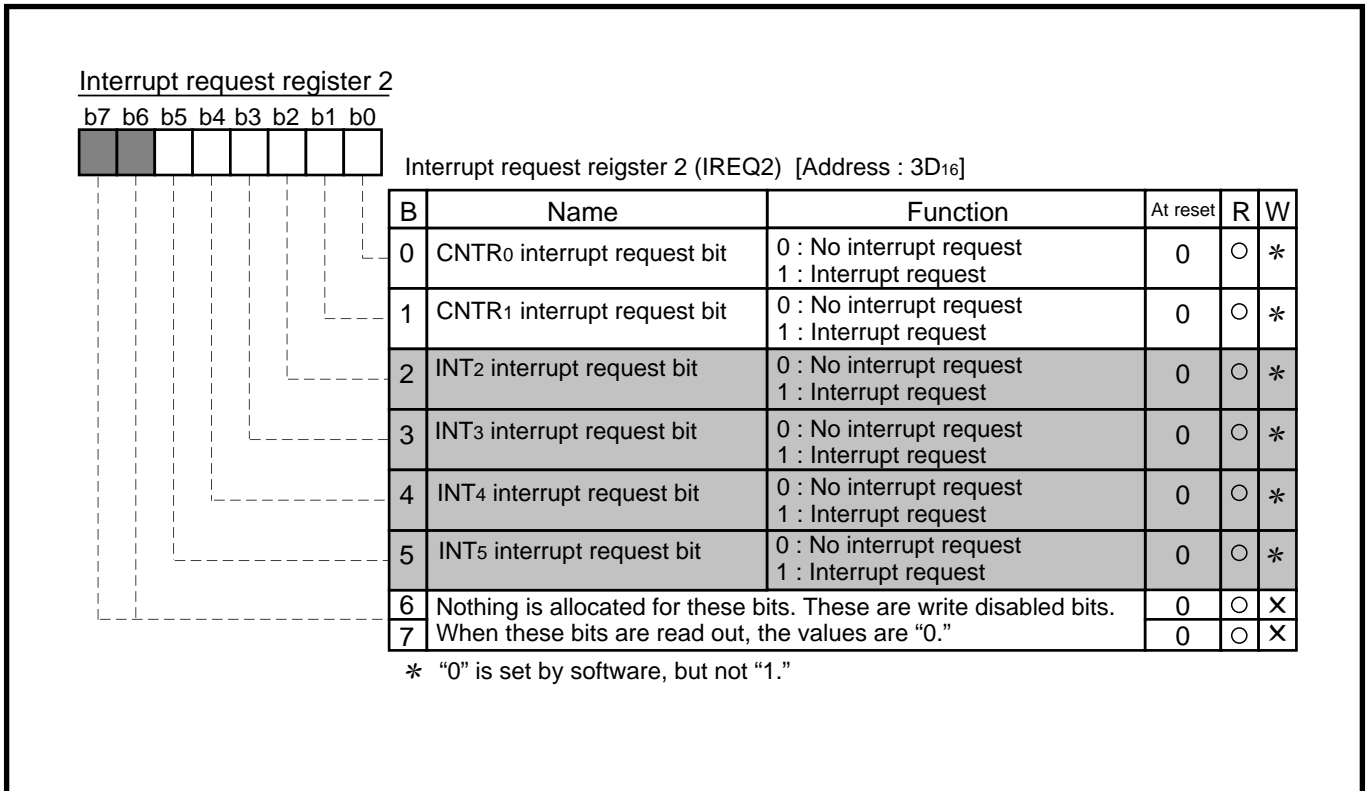


Fig. 2.2.7 Structure of Interrupt request register 2

# APPLICATION

## 2.2 Timer

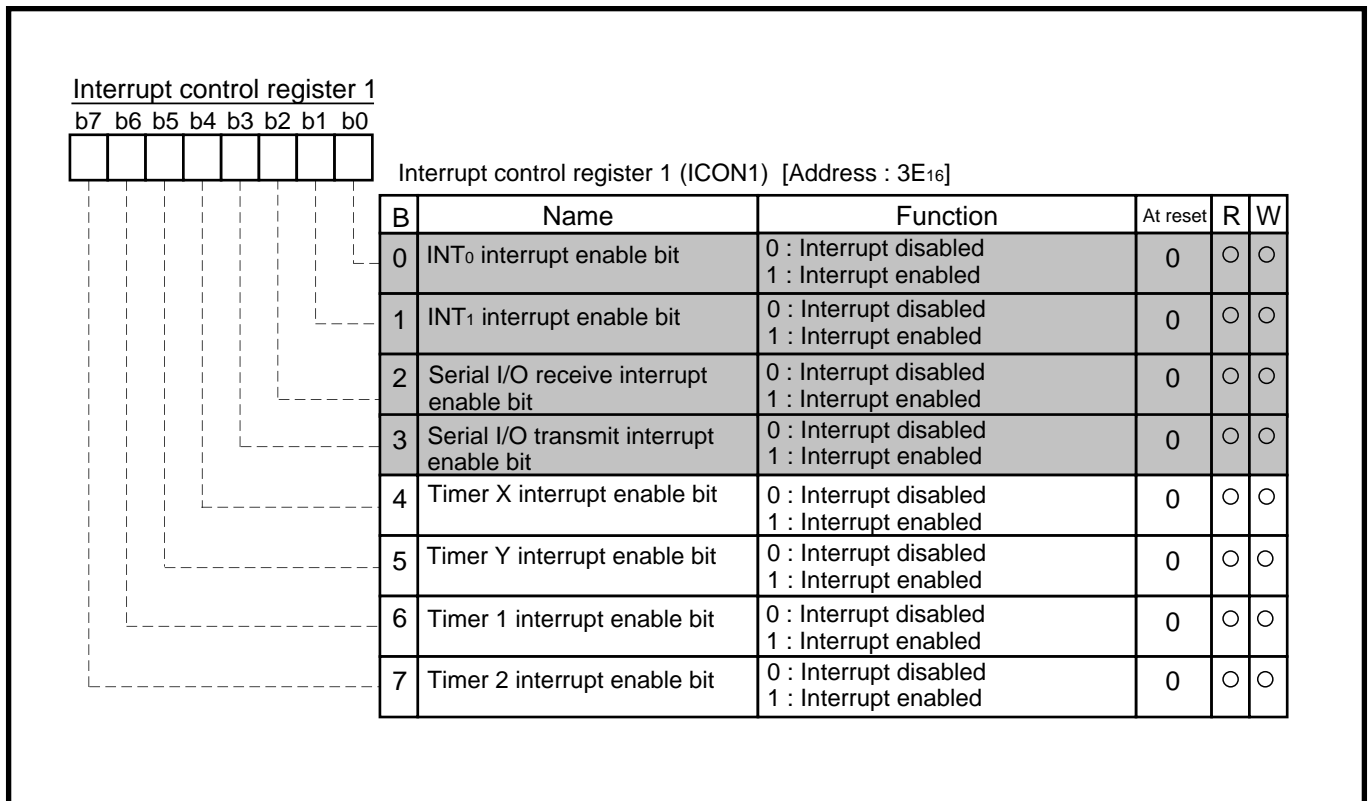


Fig. 2.2.8 Structure of Interrupt control register 1

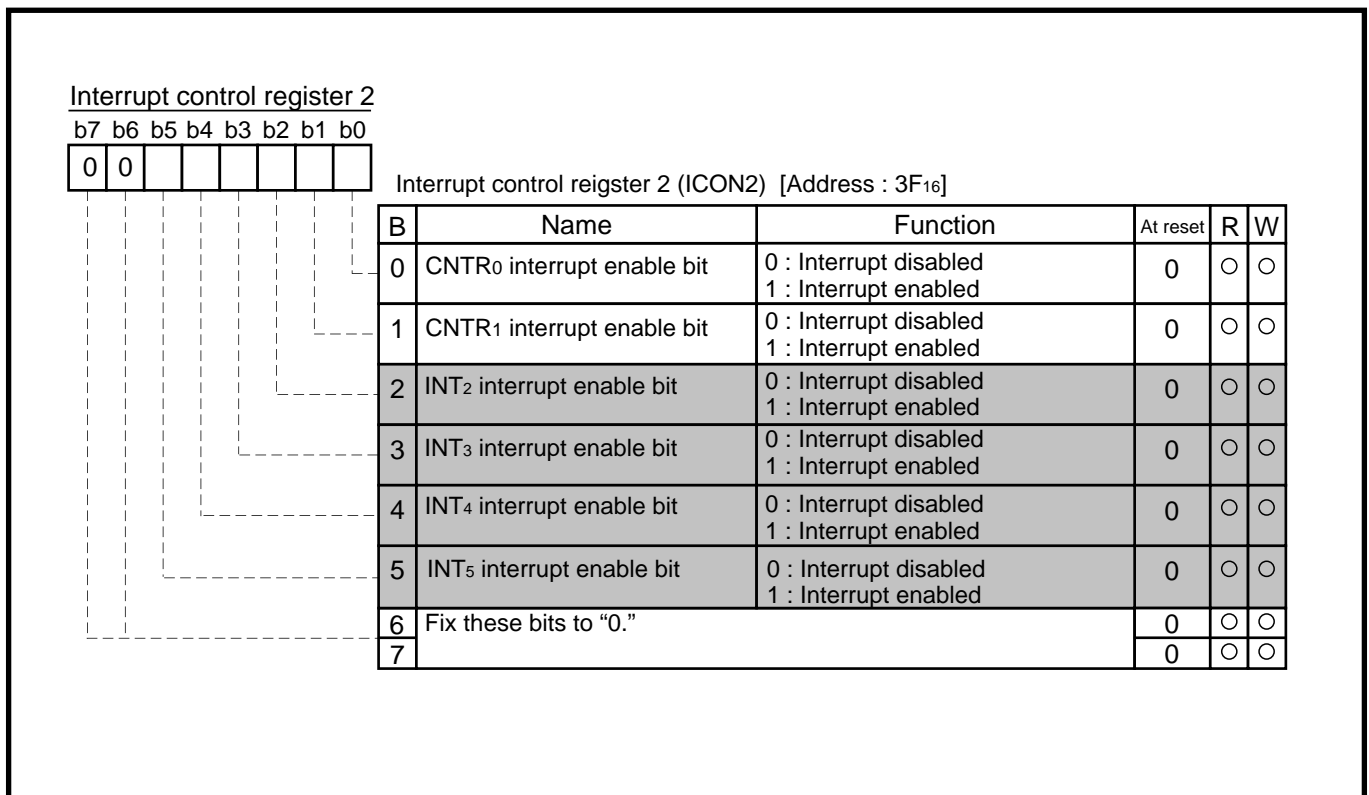


Fig. 2.2.9 Structure of Interrupt control register 2

### 2.2.3 Timer application examples

#### (1) Basic functions and uses

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

The Timer count stop bit is set to "0" after setting a count value to a timer. Then a timer interrupt request occurs after a certain period.

- [Use]**
- Generation of an output signal timing
  - Generation of a waiting time

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

The value of a timer latch is automatically written to a corresponding timer every time a timer underflows, and each cyclic timer interrupt request occurs.

- [Use]**
- Generation of cyclic interrupts
  - Clock function (measurement of 250m second) → 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 every time a timer underflows (Pulse output mode).

- [Use]**
- A piezoelectric buzzer output → Application example 2
  - Generation of the remote-control carrier waveforms

**[Function 4] Count of External pulse (Timer X, Timer Y)**

External pulses input to the CNTR pin are selected as a timer count source (Event counter mode).

- [Use]**
- Measurement of frequency → Application example 3
  - Division of external pulses.
  - Generation of interrupts in a cycle based on an external pulse.  
(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 (Pulse width measurement mode).

- [Use]**
- Measurement of external pulse frequency (Measurement of pulse width of FG pulse\* generated by motor) → 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.

# APPLICATION

## 2.2 Timer

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

**Outline :** The input clock is divided by a timer so that the clock counts up every 250 ms.

**Specifications :**

- The clock  $f(X_{IN}) = 4.19 \text{ MHz}$  ( $2^{22} \text{ Hz}$ ) is divided by a timer.
- The clock is counted at intervals of 250 ms by the Timer X interrupt.

Figure 2.2.10 shows a connection of timers and a setting of division ratios, Figures 2.2.11 show a setting of related registers, and Figure 2.2.12 shows a control procedure.

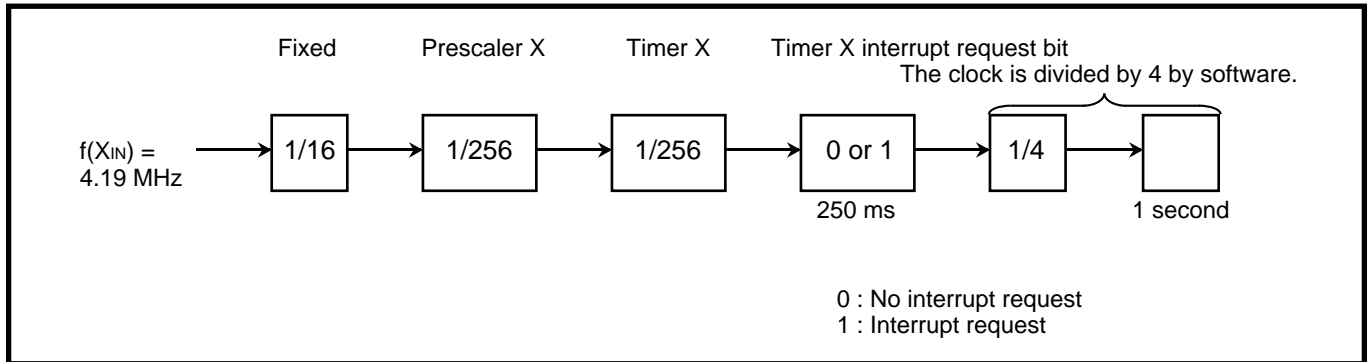


Fig. 2.2.10 Connection of timers and setting of division ratios [Clock function]

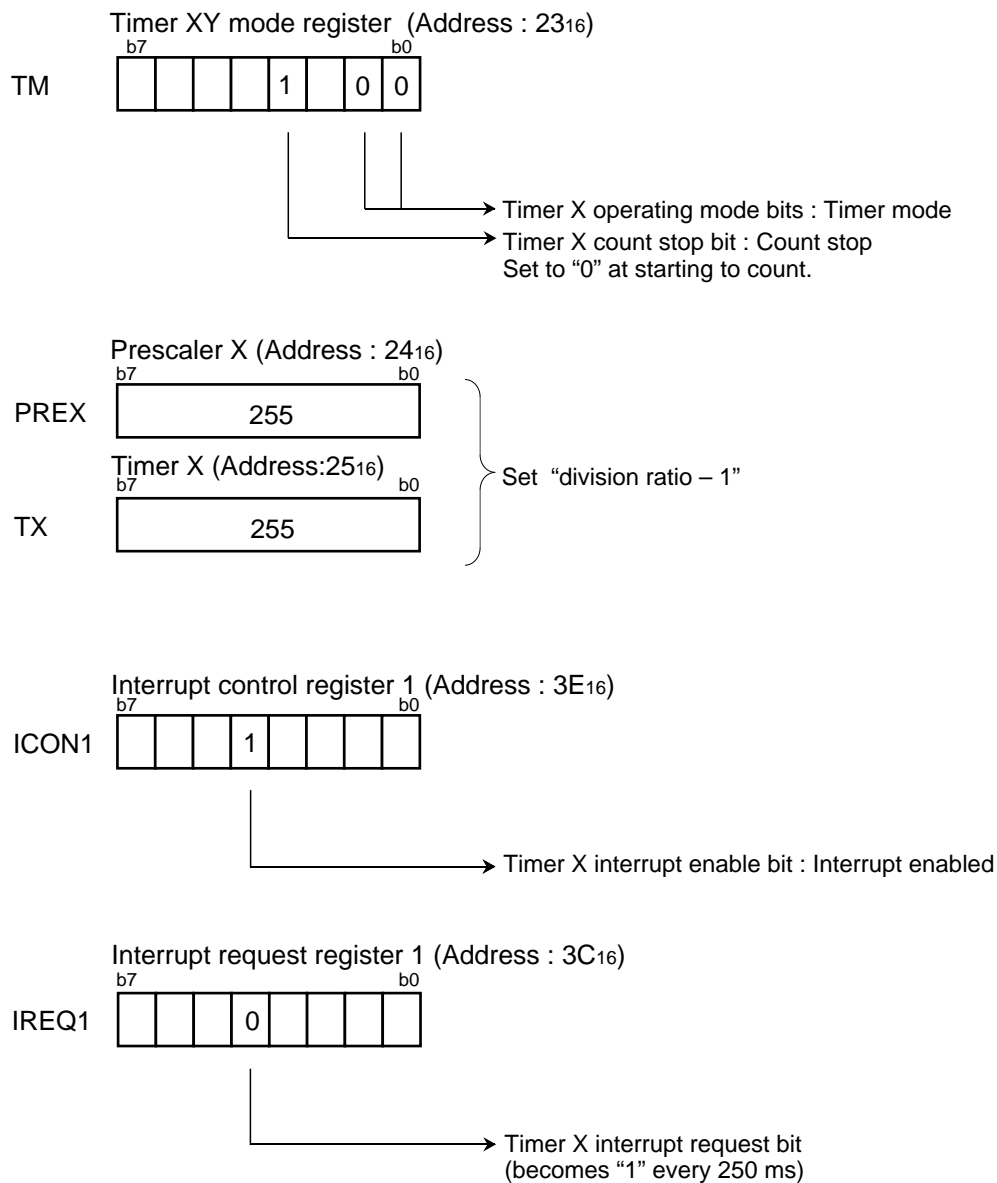


Fig. 2.2.11 Setting of related registers [Clock function]

# APPLICATION

## 2.2 Timer

### Control procedure :

Figure 2.2.12 shows a control procedure.

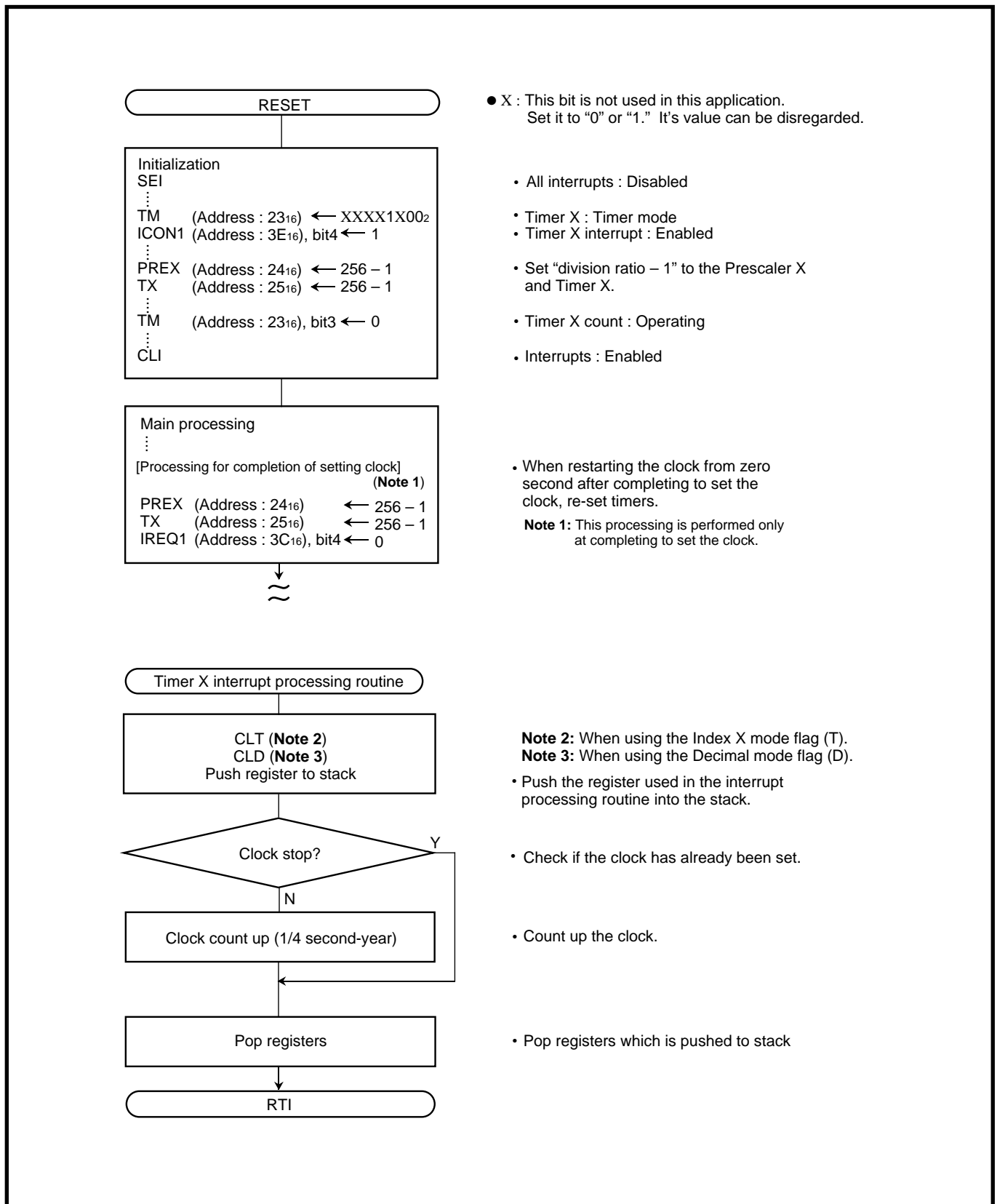


Fig. 2.2.12 Control procedure [Clock function]

### (3) Timer application example 2 : Piezoelectric buzzer output

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

- Specifications :**
- The rectangular waveform resulting from dividing clock  $f(X_{IN}) = 4.19 \text{ MHz}$  into about 2 kHz (2048 Hz) is output from the P54/CNTR0 pin.
  - The level of the P54/CNTR0 pin fixes to "H" while a piezoelectric buzzer output is stopped.

Figure 2.2.13 shows an example of a peripheral circuit, and Figure 2.2.14 shows a connection of the timer and setting of the division ratio.

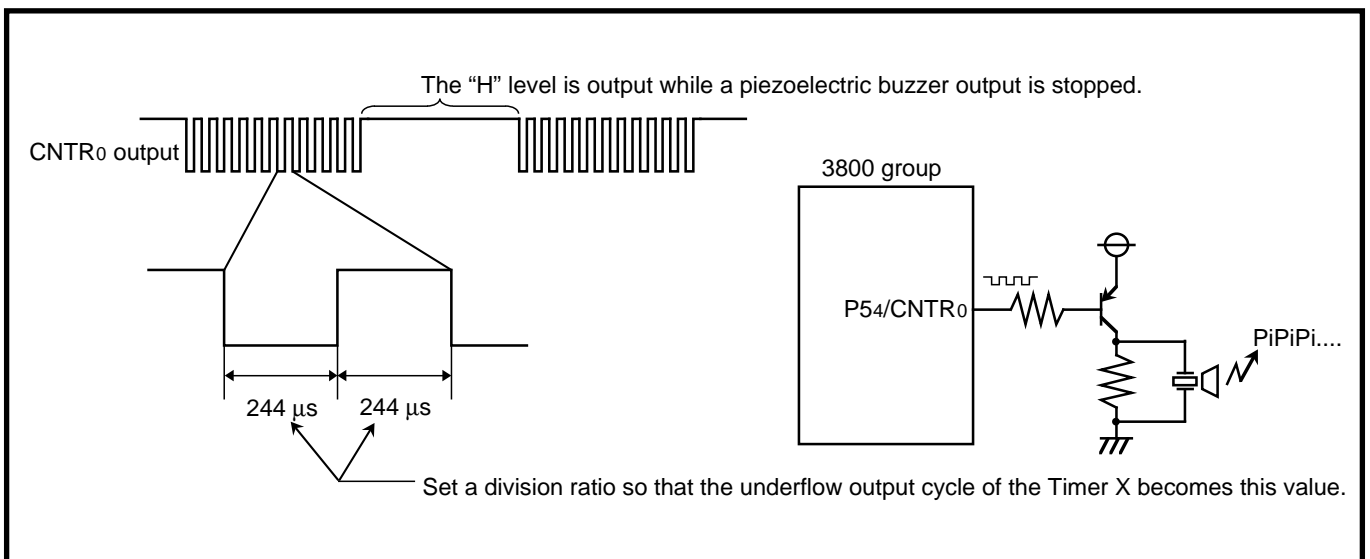


Fig. 2.2.13 Example of a peripheral circuit

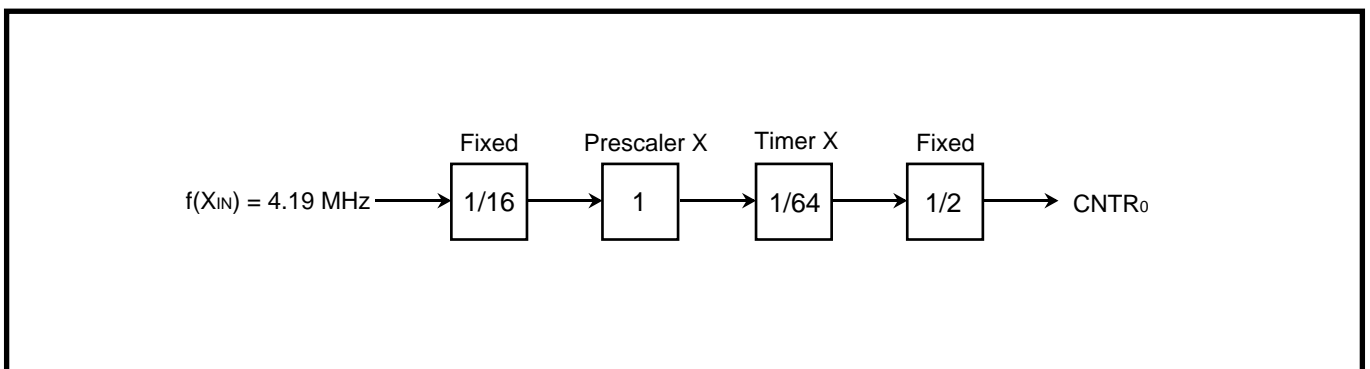


Fig. 2.2.14 Connection of the timer and setting of the division ratio [Piezoelectric buzzer output]

# APPLICATION

## 2.2 Timer

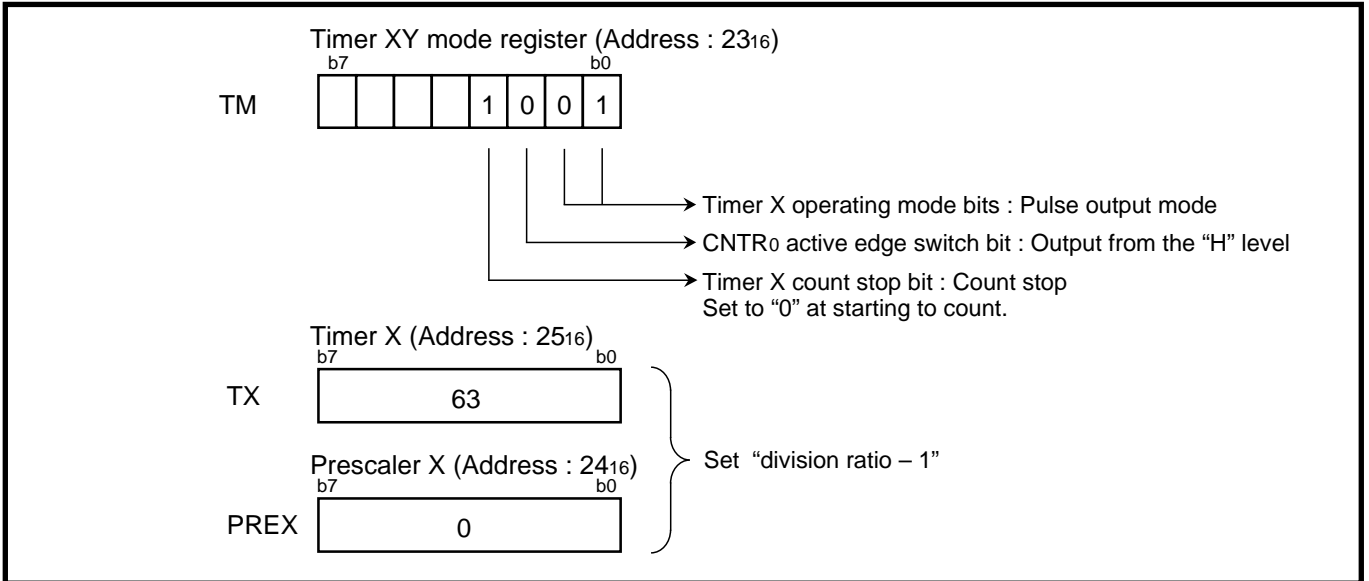


Fig. 2.2.15 Setting of related registers [Piezoelectric buzzer output]

### Control procedure :

Figure 2.2.16 shows a control procedure.

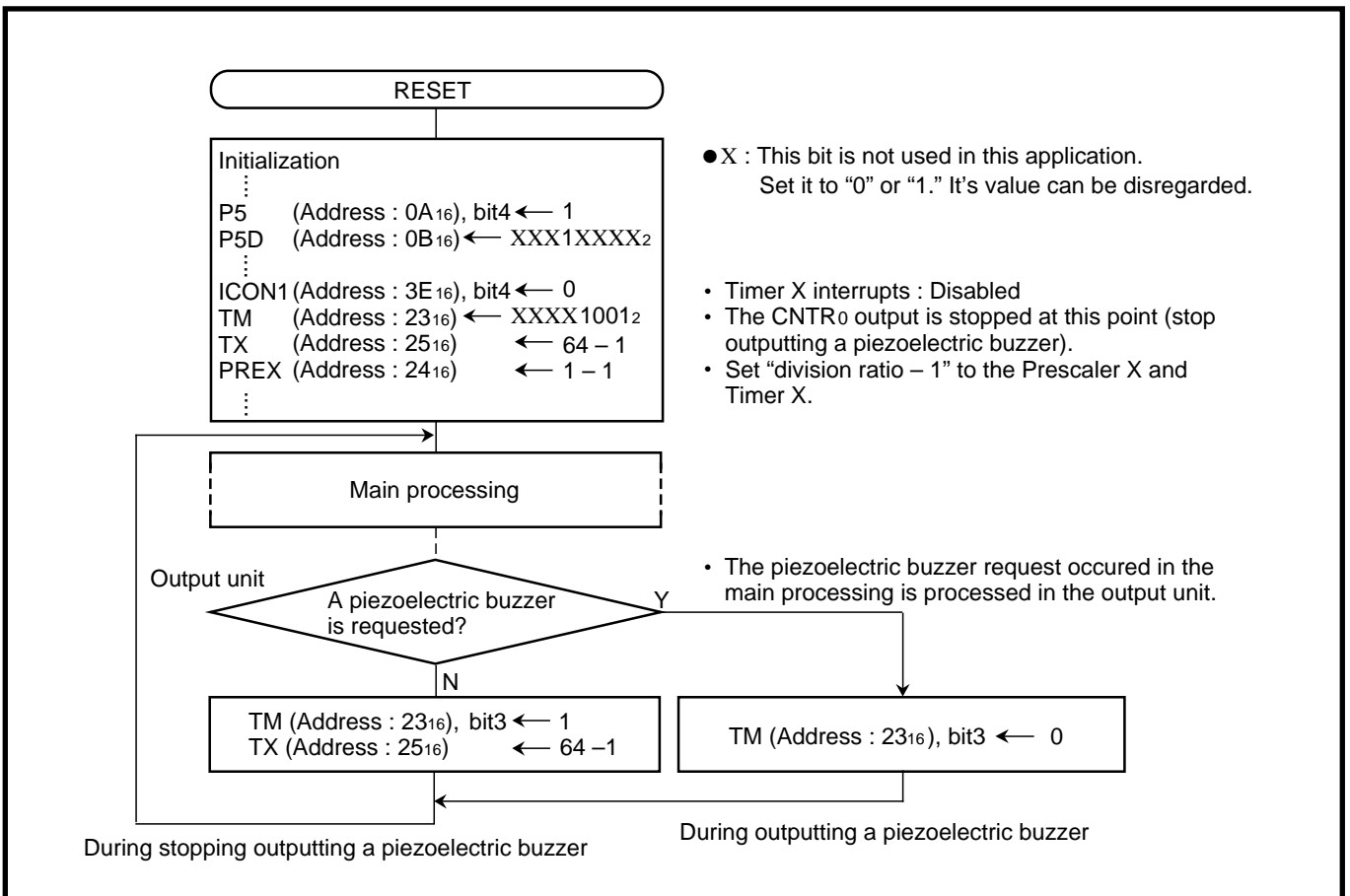


Fig. 2.2.16 Control procedure [Piezoelectric buzzer output]



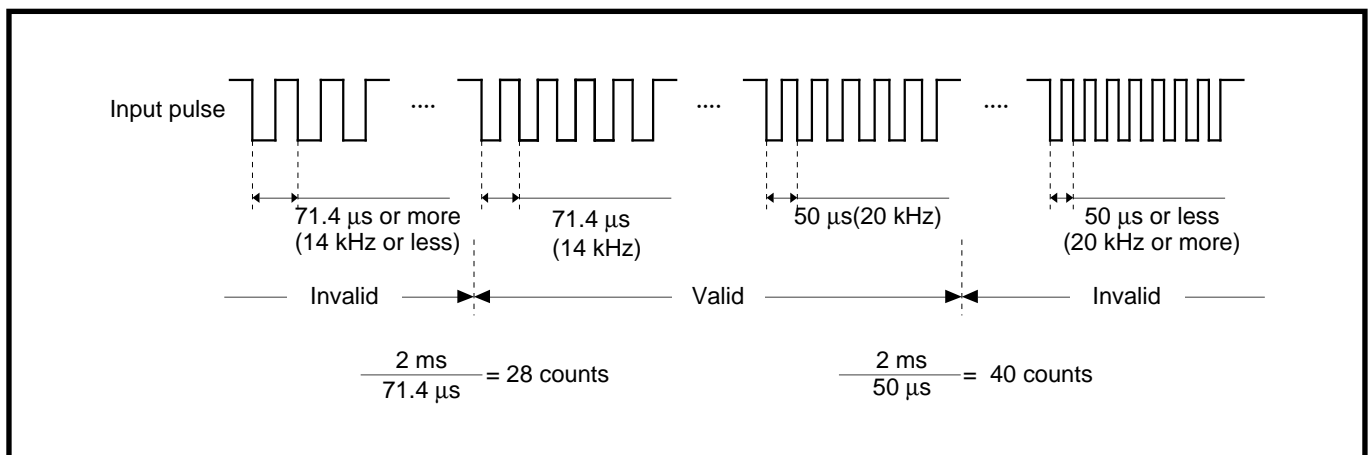
**(4) Timer application example 3 : Measurement of frequency**

**Outline :** The following two values are compared for judging if the frequency is within a certain range.

- A value counted a pulse which is input to P55/CNTR1 pin by a timer.
- A reference value

- Specifications :**
- The pulse is input to the P55/CNTR1 pin and counted by the Timer Y.
  - A count value is read out at the interval of about 2 ms (Timer 1 interrupt interval :  $244 \mu\text{s} \times 8$ ). When the count value is 28 to 40, it is regarded the input pulse as a valid.
  - Because the timer is a down-counter, the count value is compared with 227 to 215\* .
- \*  $227 \text{ to } 215 = 255 \text{ (initialized value of counter) } - 28 \text{ to } 40 \text{ (the number of valid value)}$ .

Figure 2.2.17 shows a method for judging if input pulse exists, and Figure 2.2.18 shows a setting of related registers.



**Fig 2.2.17 A method for judging if input pulse exists**

# APPLICATION

## 2.2 Timer

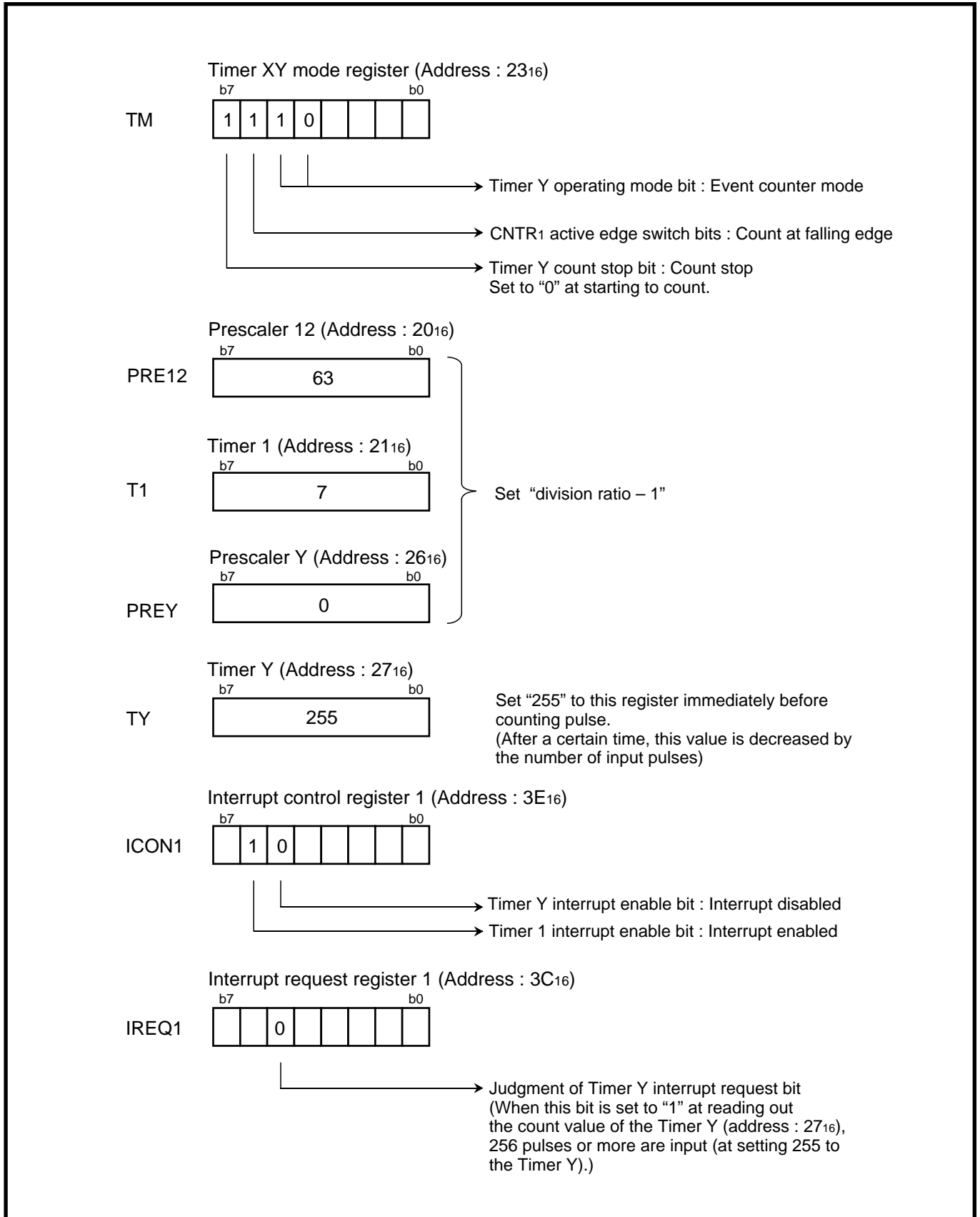


Fig. 2.2.18 Setting of related registers [Measurement of frequency]

### Control procedure :

Figure 2.2.19 shows a control procedure.

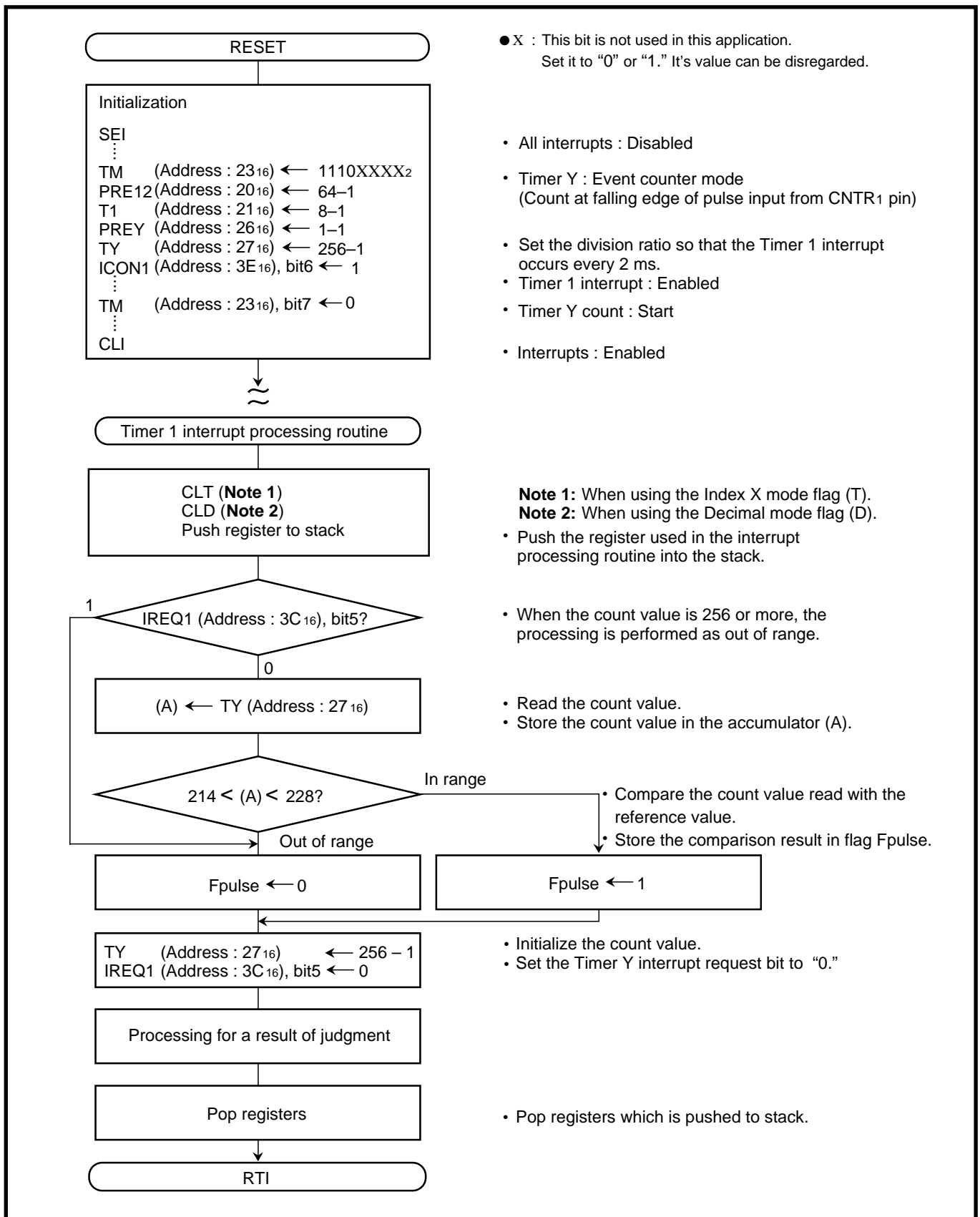


Fig. 2.2.19 Control procedure [Measurement of frequency]

# APPLICATION

## 2.2 Timer

### (5) Timer application example 4 : Measurement of pulse width of FG pulse generated by motor

**Outline** : The “H” level width of a pulse input to the P54/CNTR0 pin is counted by Timer X. An underflow is detected by Timer X interrupt and an end of the input pulse “H” level is detected by CNTR0 interrupt.

**Specifications** : • The “H” level width of a FG pulse input to the P54/CNTR0 pin is counted by Timer X.

(Example : When the clock frequency is 4.19 MHz, the count source would be  $3.8 \mu\text{s}$  that is obtained by dividing the clock frequency by 16.  
Measurement can be made up to 250 ms in the range of FFFF<sub>16</sub> to 0000<sub>16</sub>.)

Figure 2.2.20 shows a connection of the timer and a setting of the division ratio, and Figure 2.2.21 shows a setting of related registers.

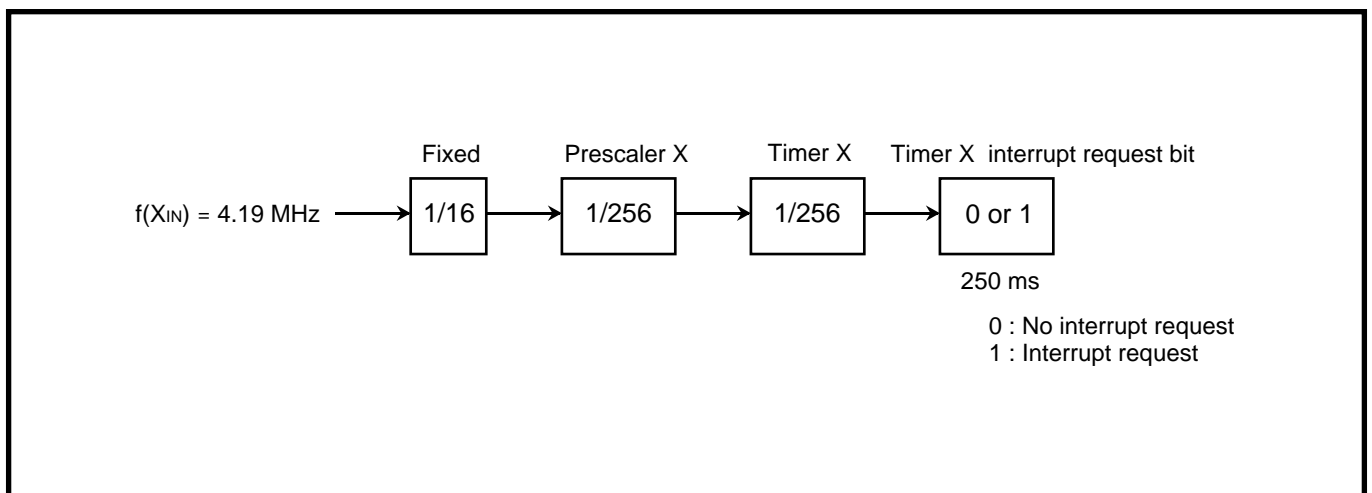
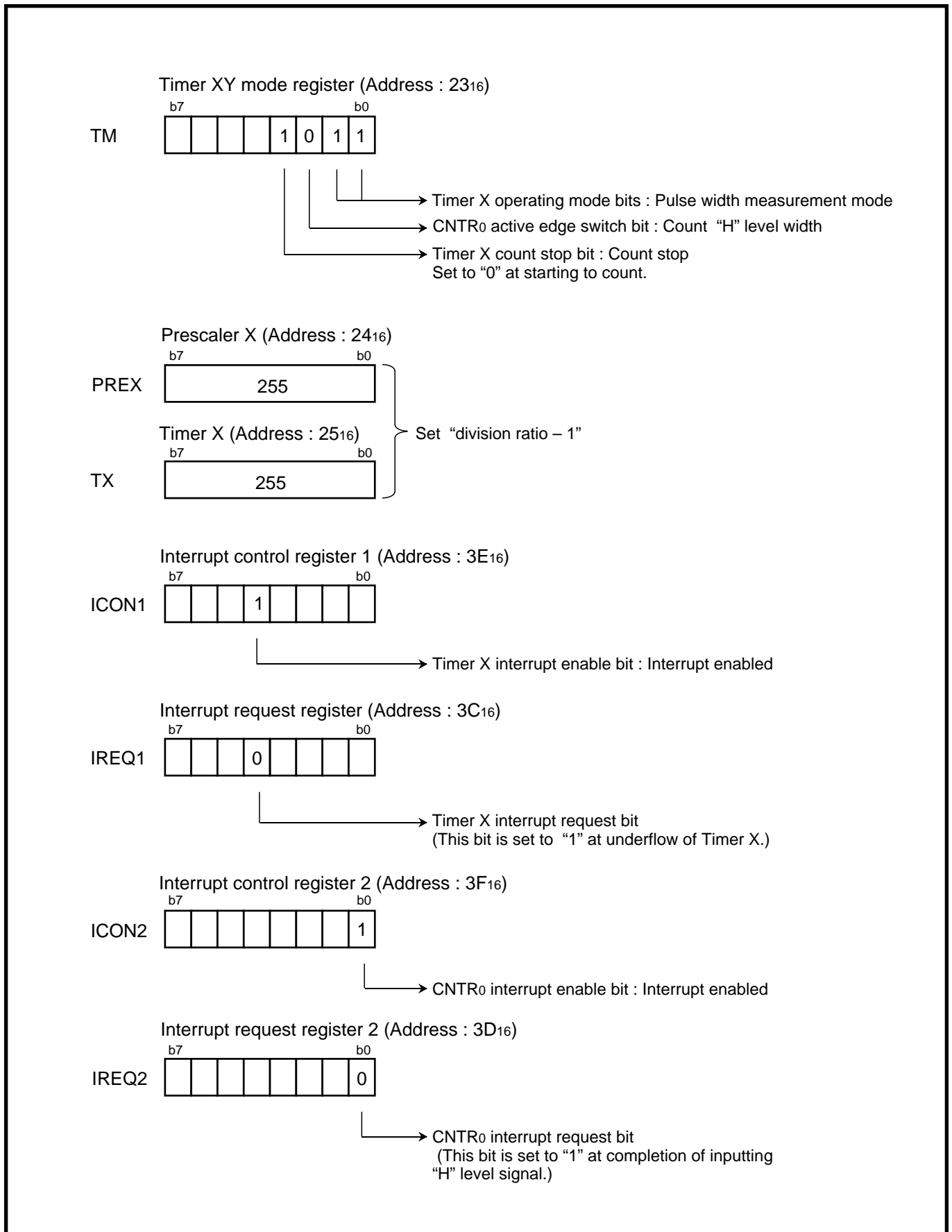


Fig. 2.2.20 Connection of the timer and setting of the division ratio [Measurement of pulse width]



**Fig. 2.2.21 Setting of related registers [Measurement of pulse width]**

# APPLICATION

## 2.2 Timer

Figure 2.2.22 shows a control procedure.

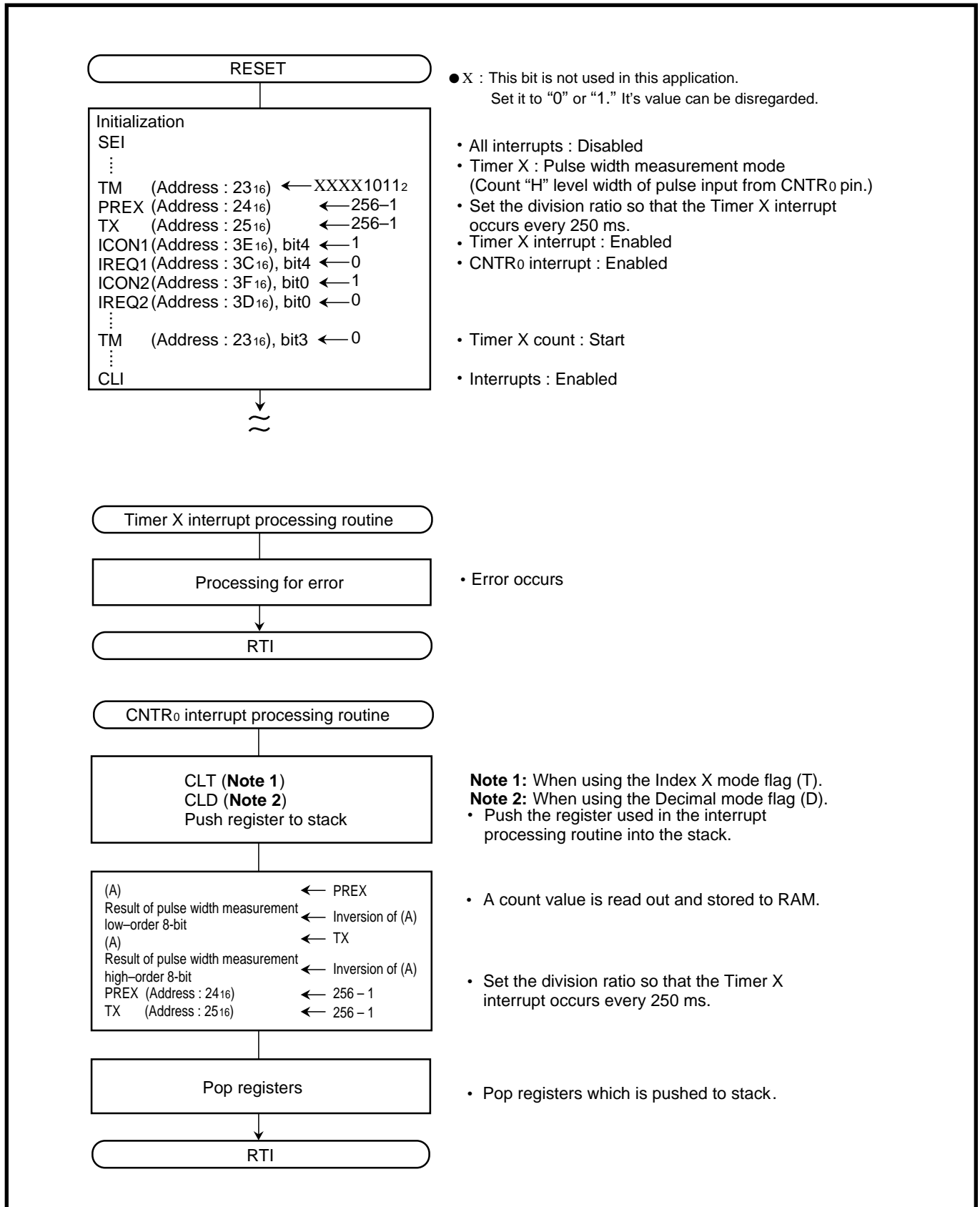


Fig. 2.2.22 Control procedure [Measurement of pulse width]

2.3 Serial I/O

2.3.1 Memory map of serial I/O

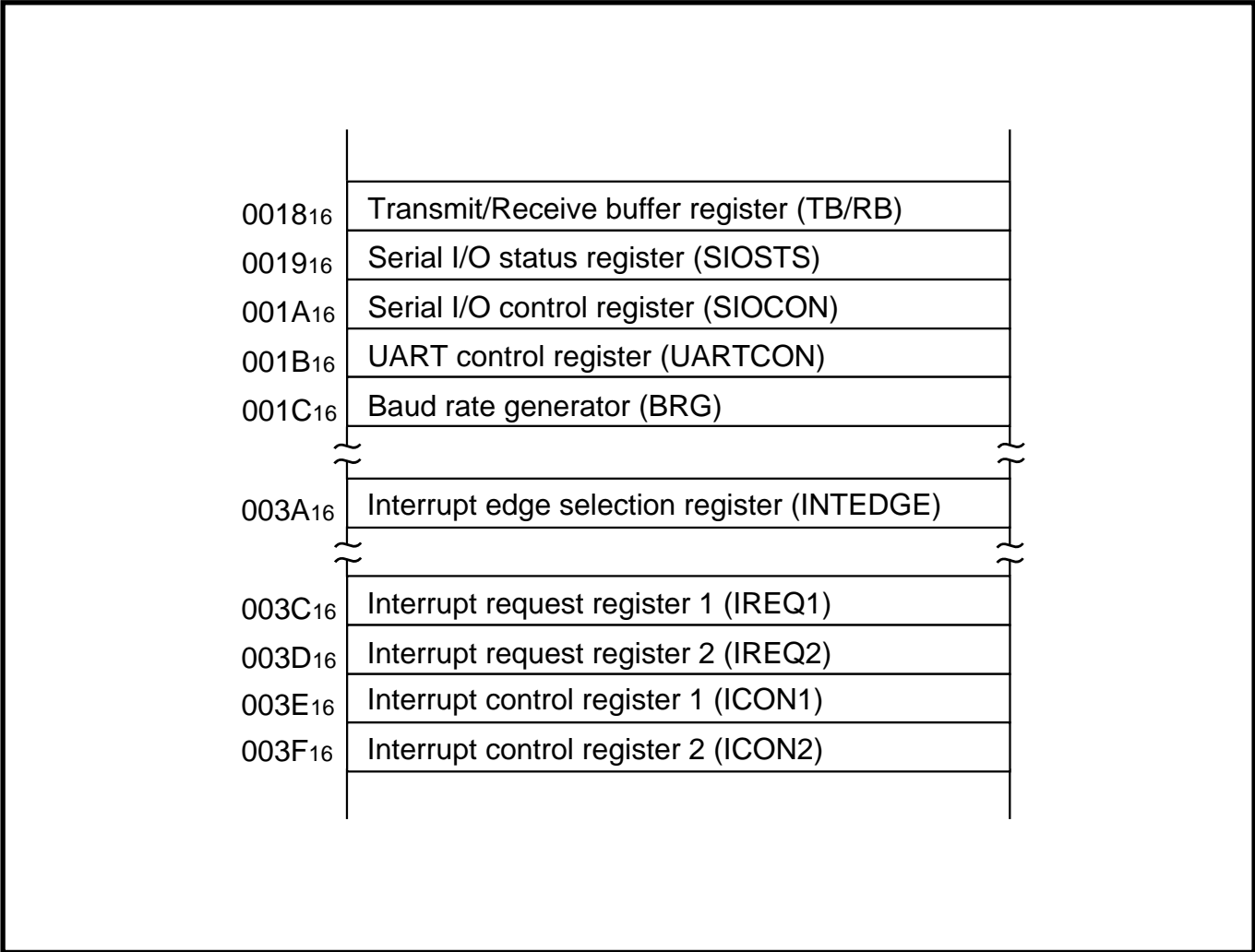


Fig. 2.3.1 Memory map of serial I/O related registers

# APPLICATION

## 2.3 Serial I/O

### 2.3.2 Related registers

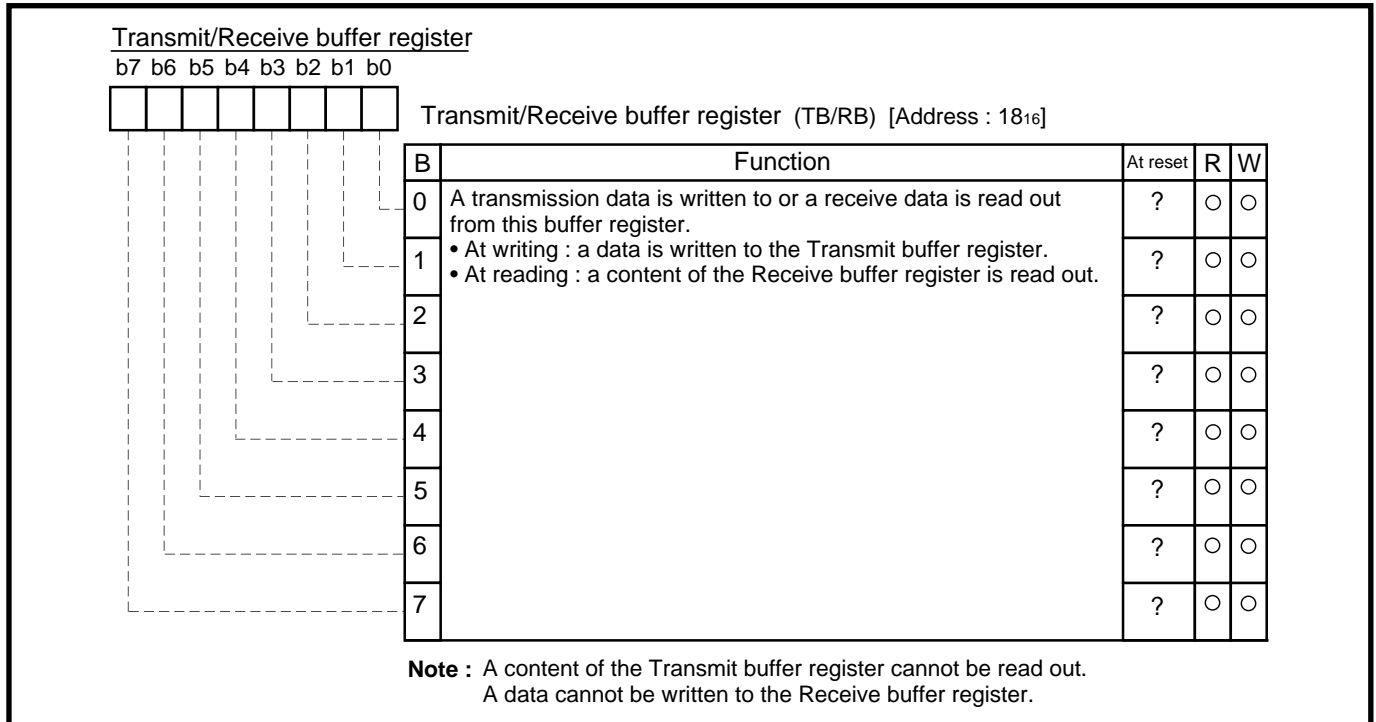


Fig. 2.3.2 Structure of Transmit/Receive buffer register

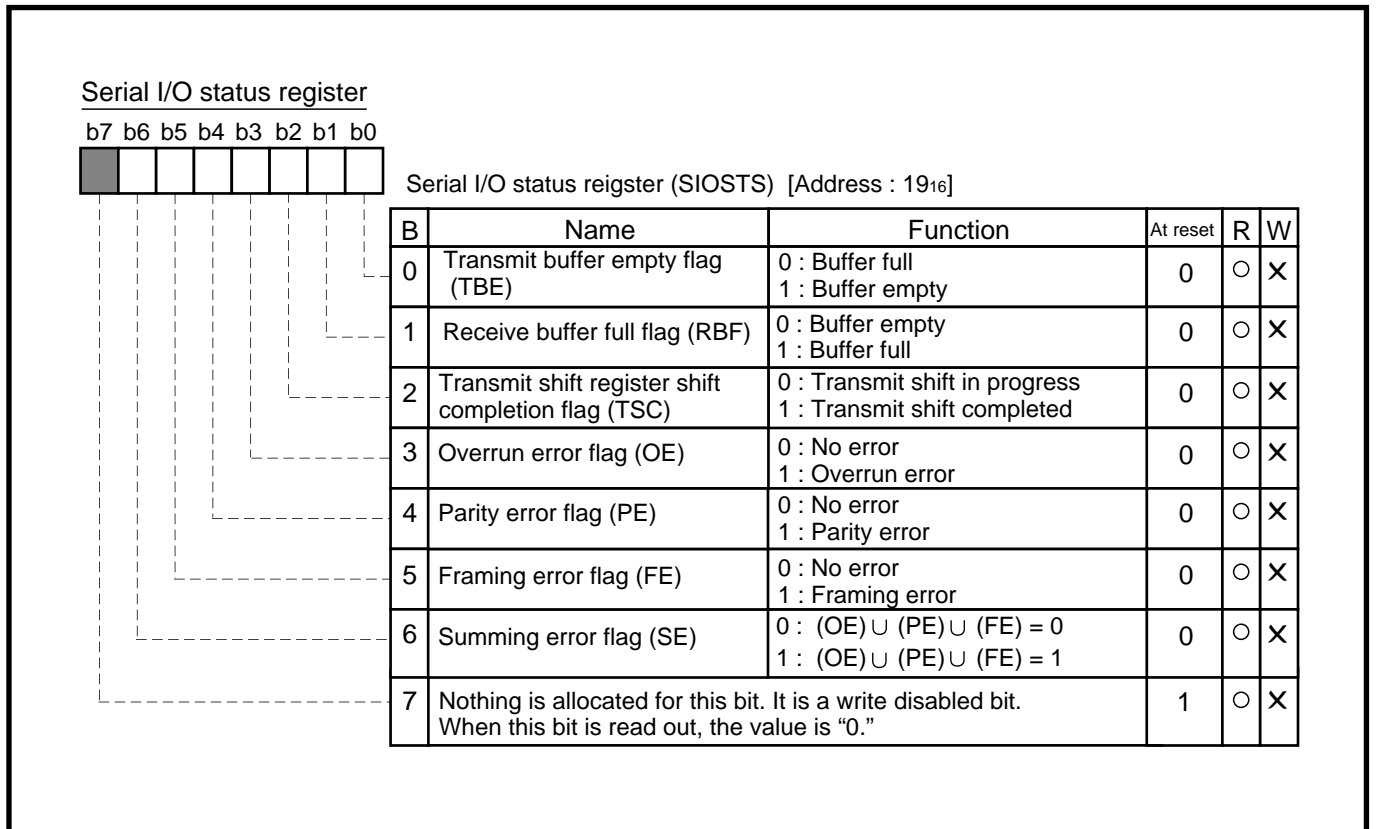


Fig. 2.3.3 Structure of Serial I/O status register



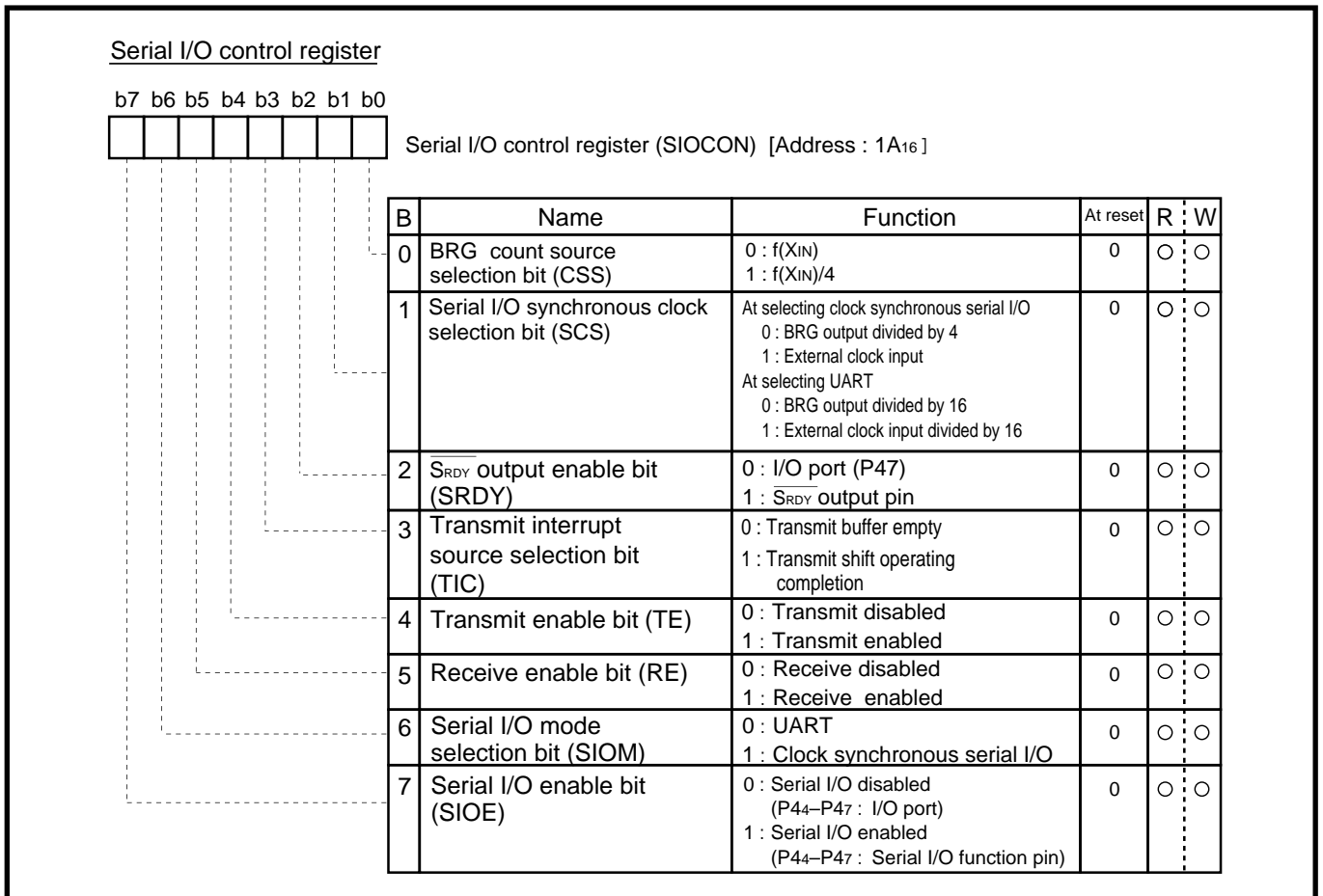


Fig. 2.3.4 Structure of Serial I/O control register

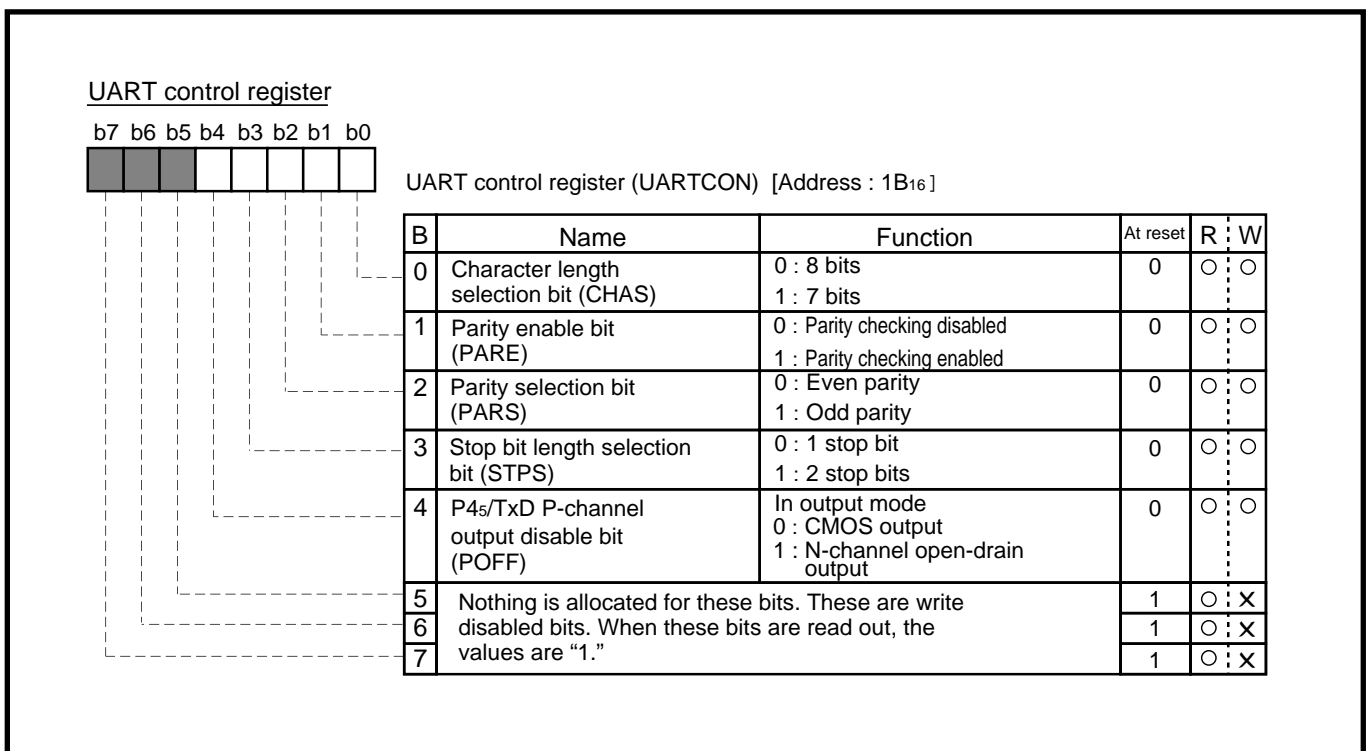


Fig. 2.3.5 Structure of UART control register

# APPLICATION

## 2.3 Serial I/O

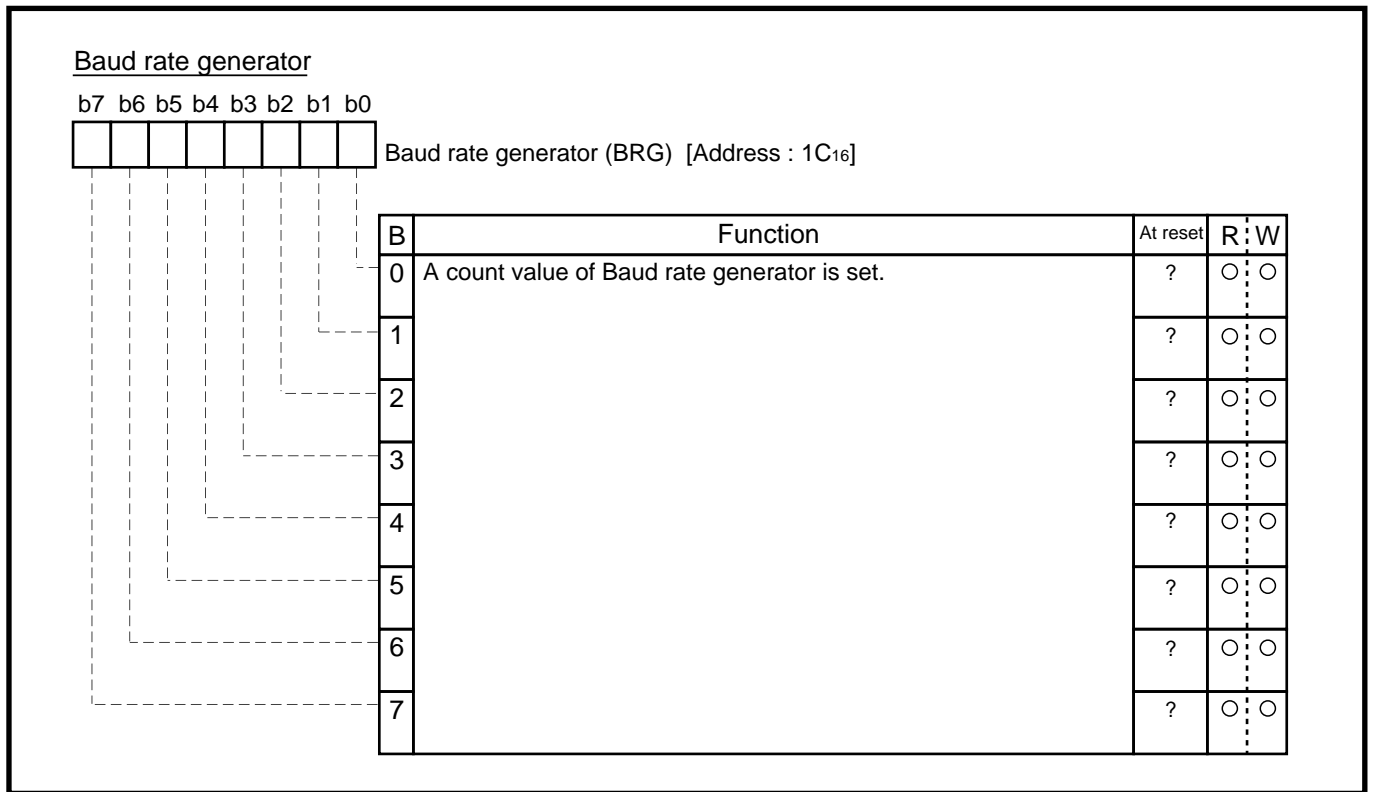


Fig. 2.3.6 Structure of Baud rate generator

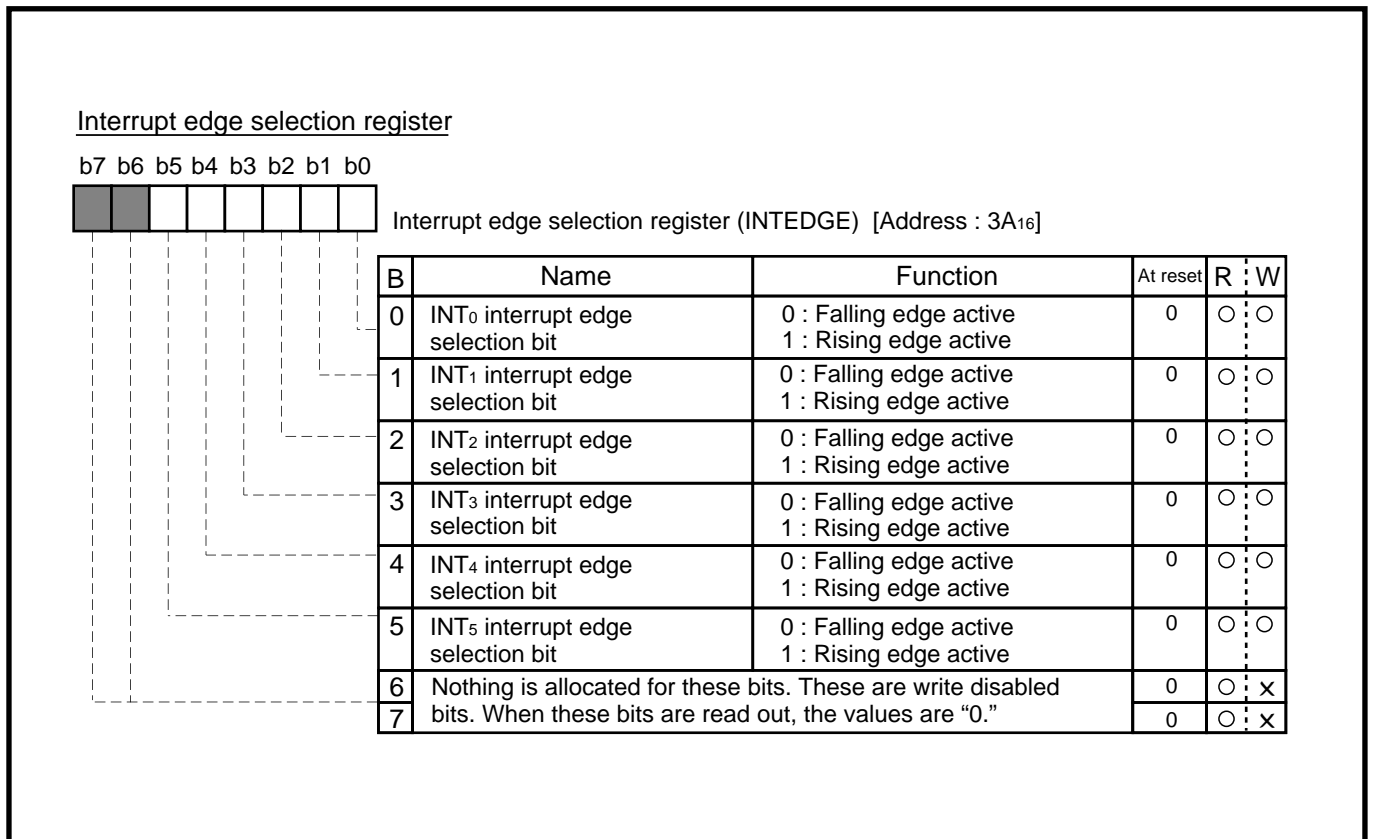


Fig. 2.3.7 Structure of Interrupt edge selection register

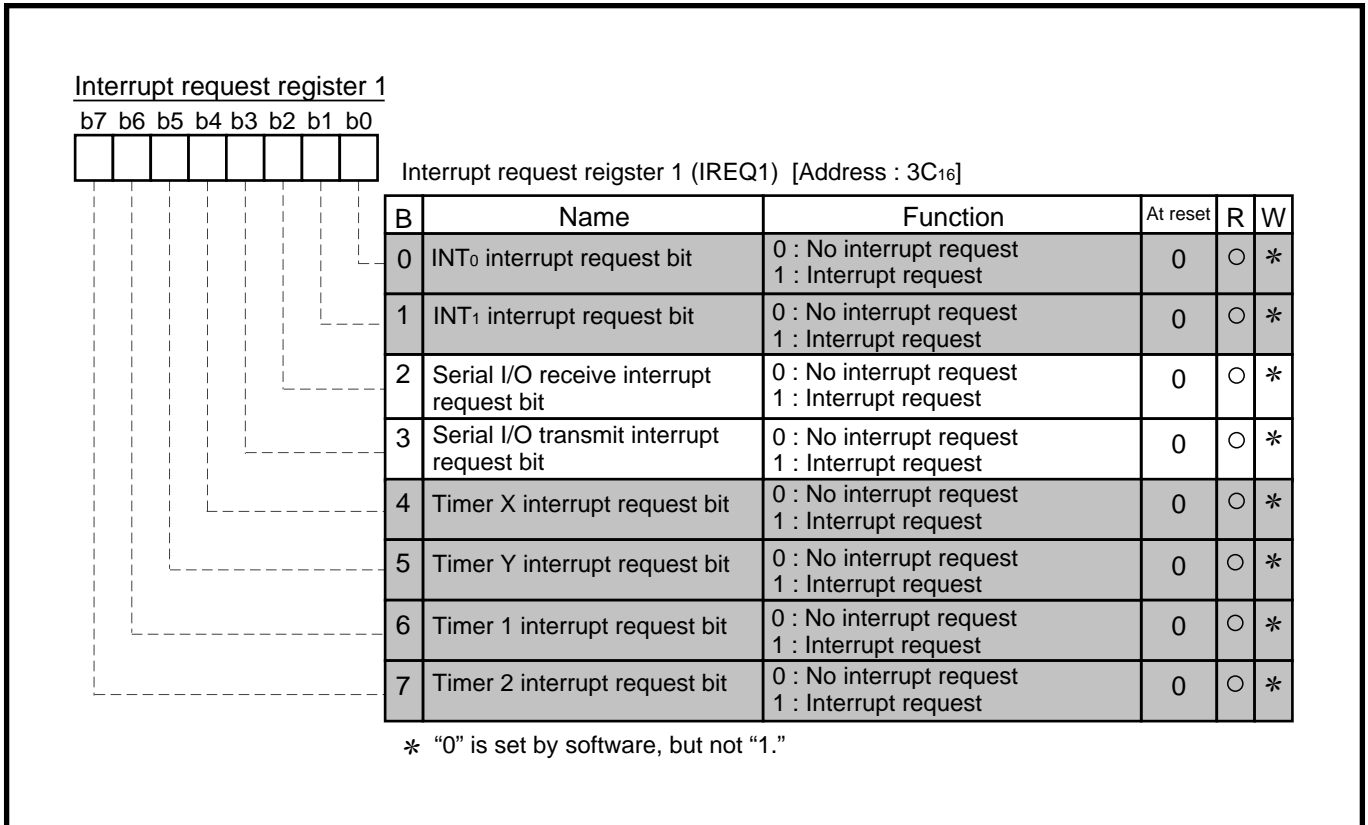


Fig. 2.3.8 Structure of Interrupt request register 1

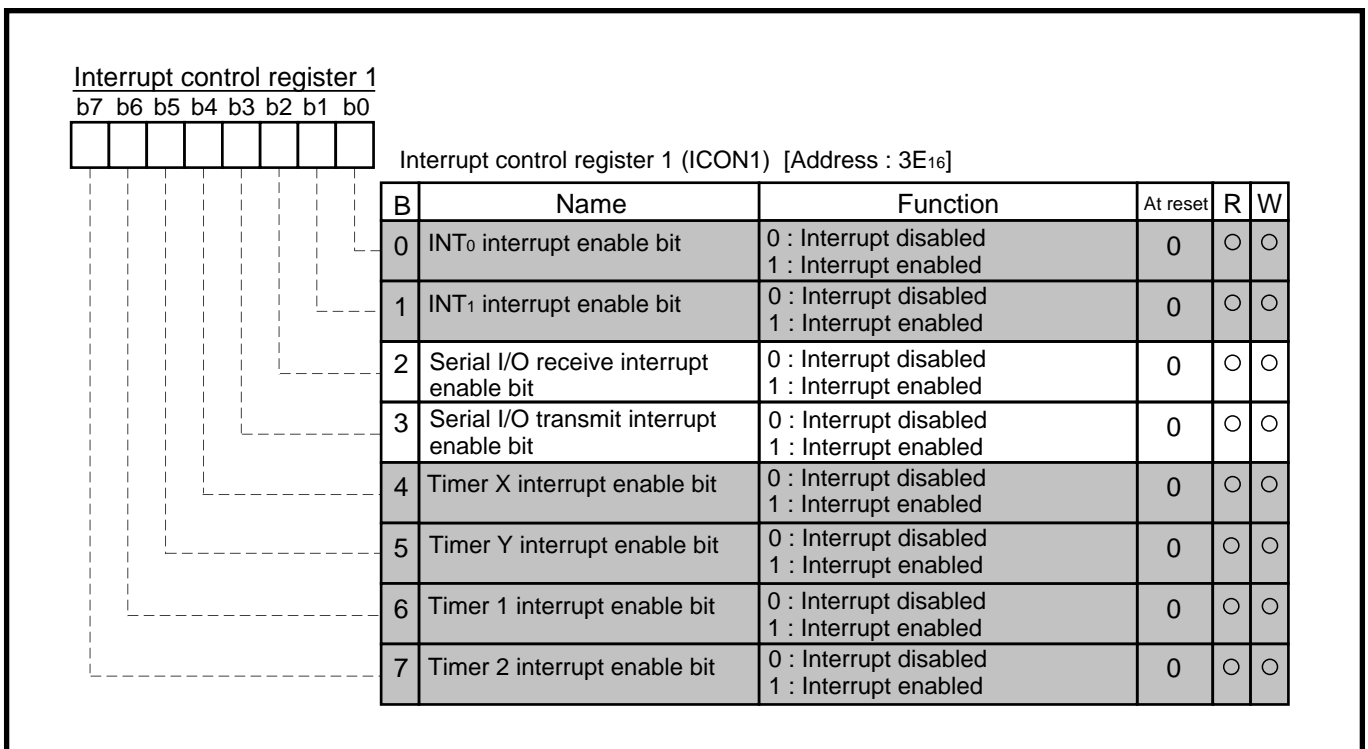


Fig. 2.3.9 Structure of Interrupt control register 1

# APPLICATION

## 2.3 Serial I/O

### 2.3.3 Serial I/O connection examples

#### (1) Control of peripheral IC equipped with CS pin

There are connection examples using a clock synchronous serial I/O mode.

Figure 2.3.10 shows connection examples of a peripheral IC equipped with the CS pin.

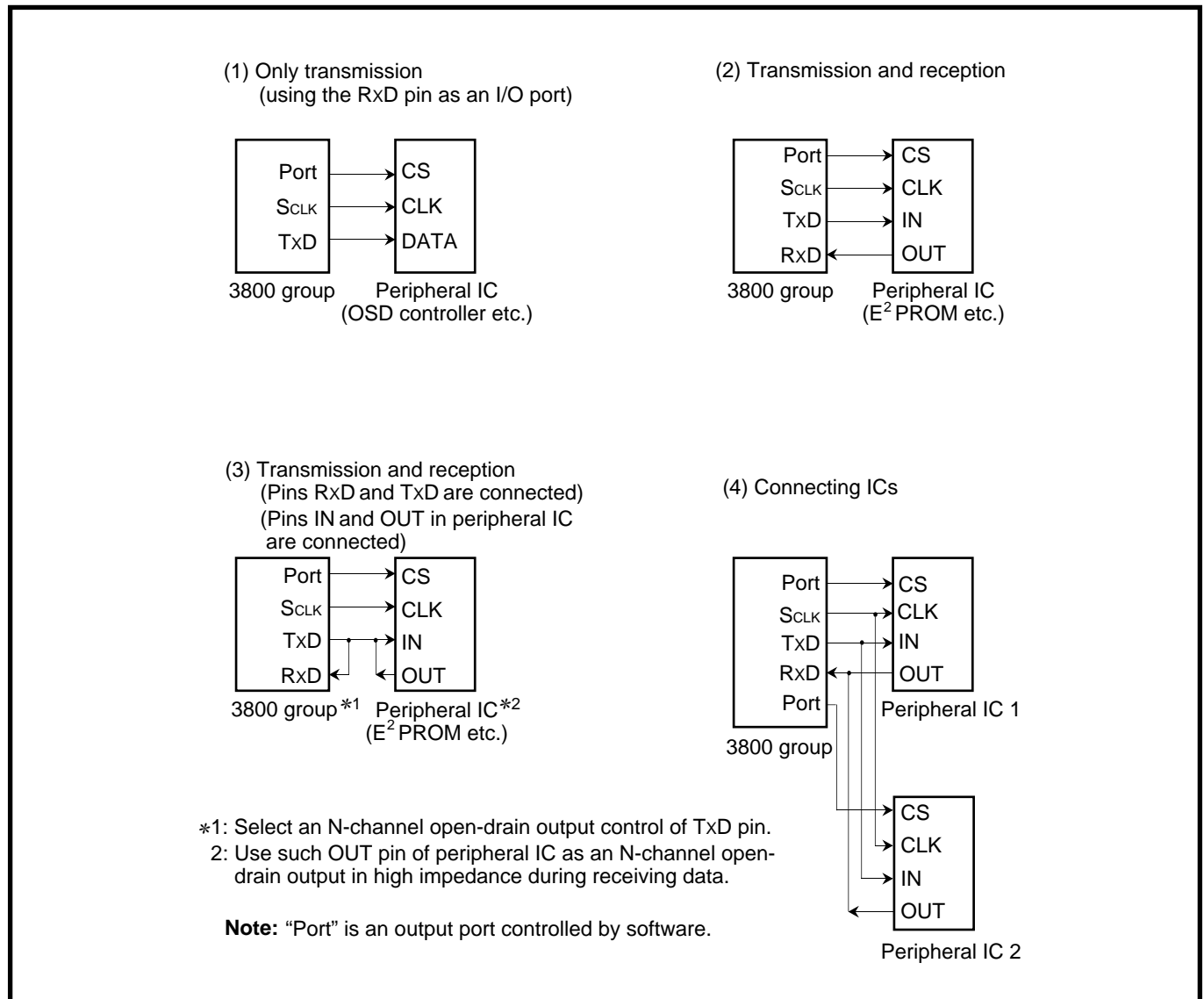


Fig. 2.3.10 Serial I/O connection examples (1)

### (2) Connection with microcomputer

Figure 2.3.11 shows connection examples of the other microcomputers.

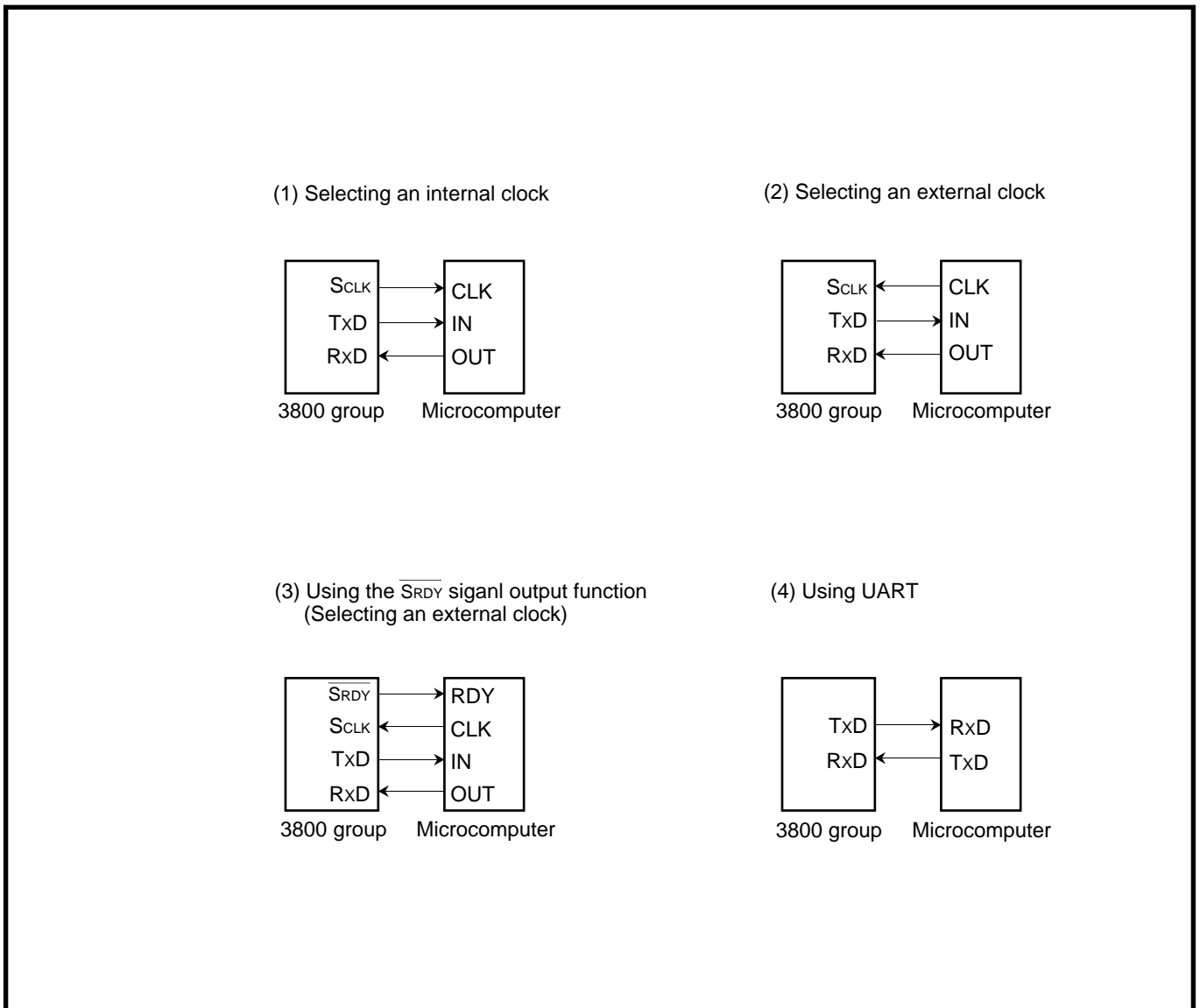


Fig. 2.3.11 Serial I/O connection examples (2)

# APPLICATION

## 2.3 Serial I/O

### 2.3.4 Setting of serial I/O transfer data format

A clock synchronous or clock asynchronous (UART) is selected as a data format of the serial I/O. Figure 2.3.12 shows a setting of serial I/O transfer data format.

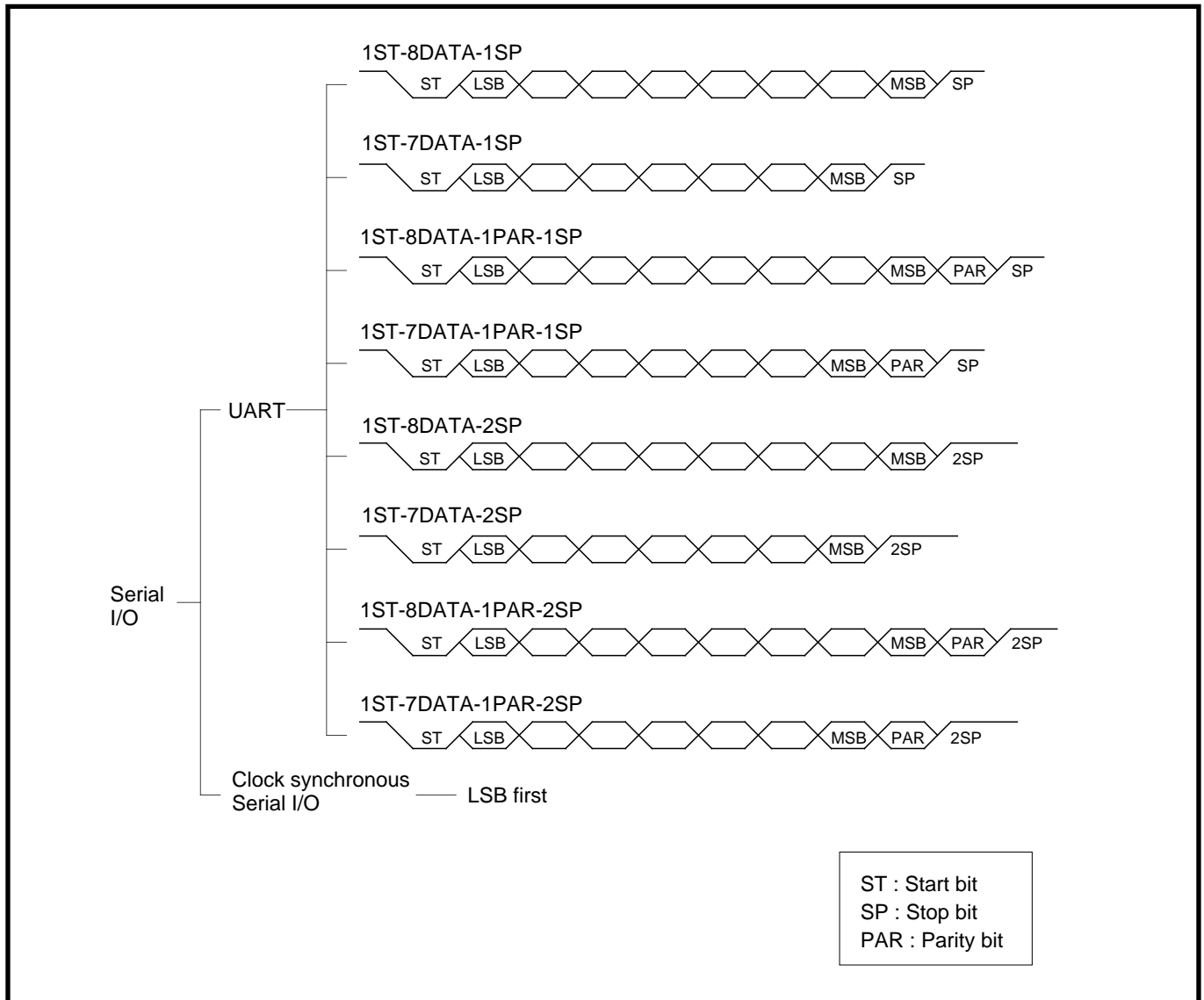


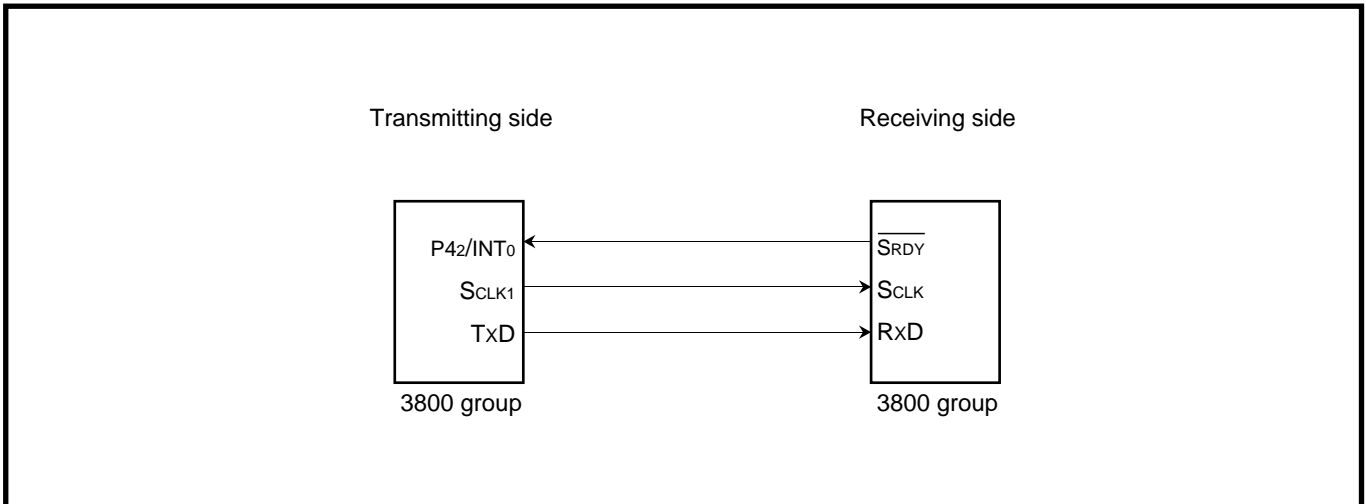
Fig. 2.3.12 Setting of Serial I/O transfer data format

### 2.3.5 Serial I/O application examples

#### (1) Communication using a clock synchronous serial I/O (transmit/receive)

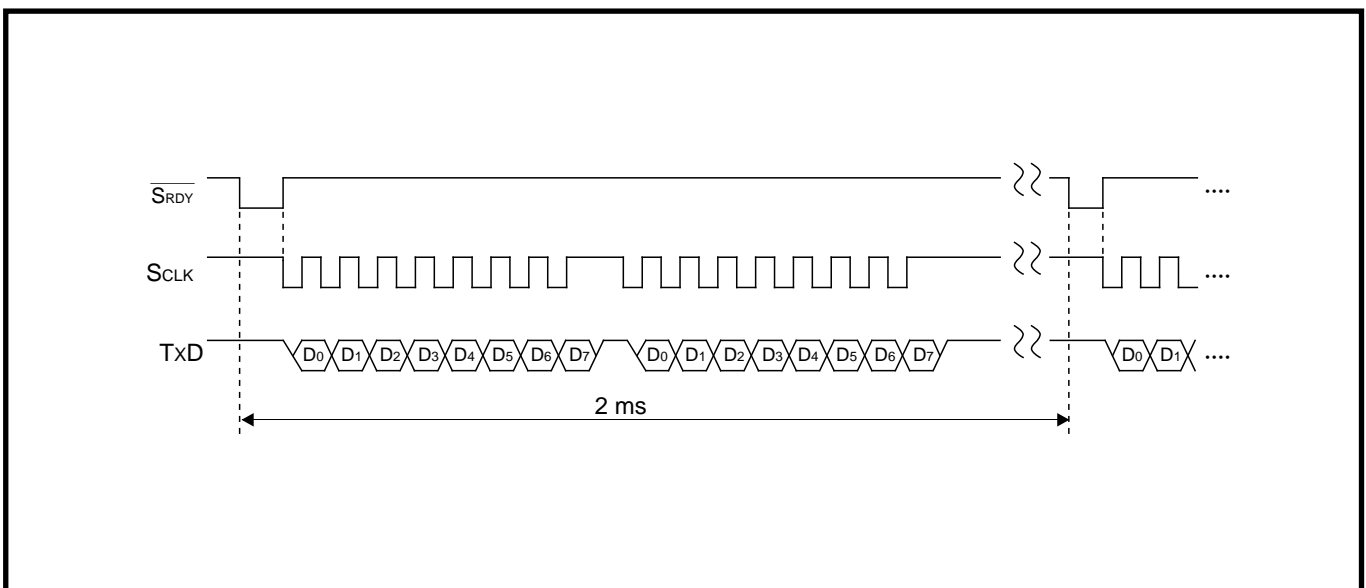
**Outline :** 2-byte data is transmitted and received through the clock synchronous serial I/O. The  $\overline{\text{SRDY}}$  signal is used for communication control.

Figure 2.3.13 shows a connection diagram, and Figure 2.3.14 shows a timing chart.



**Fig. 2.3.13 Connection diagram [Communication using a clock synchronous serial I/O]**

- Specifications :**
- The Serial I/O is used (clock synchronous serial I/O is selected)
  - Synchronous clock frequency : 125 kHz ( $f(\text{XIN}) = 4 \text{ MHz}$  is divided by 32)
  - The  $\overline{\text{SRDY}}$  (receivable signal) is used.
  - The receiving side outputs the  $\overline{\text{SRDY}}$  signal at intervals of 2 ms (generated by timer), and 2-byte data is transferred from the transmitting side to the receiving side.



**Fig. 2.3.14 Timing chart [Communication using a clock synchronous serial I/O]**

# APPLICATION

## 2.3 Serial I/O

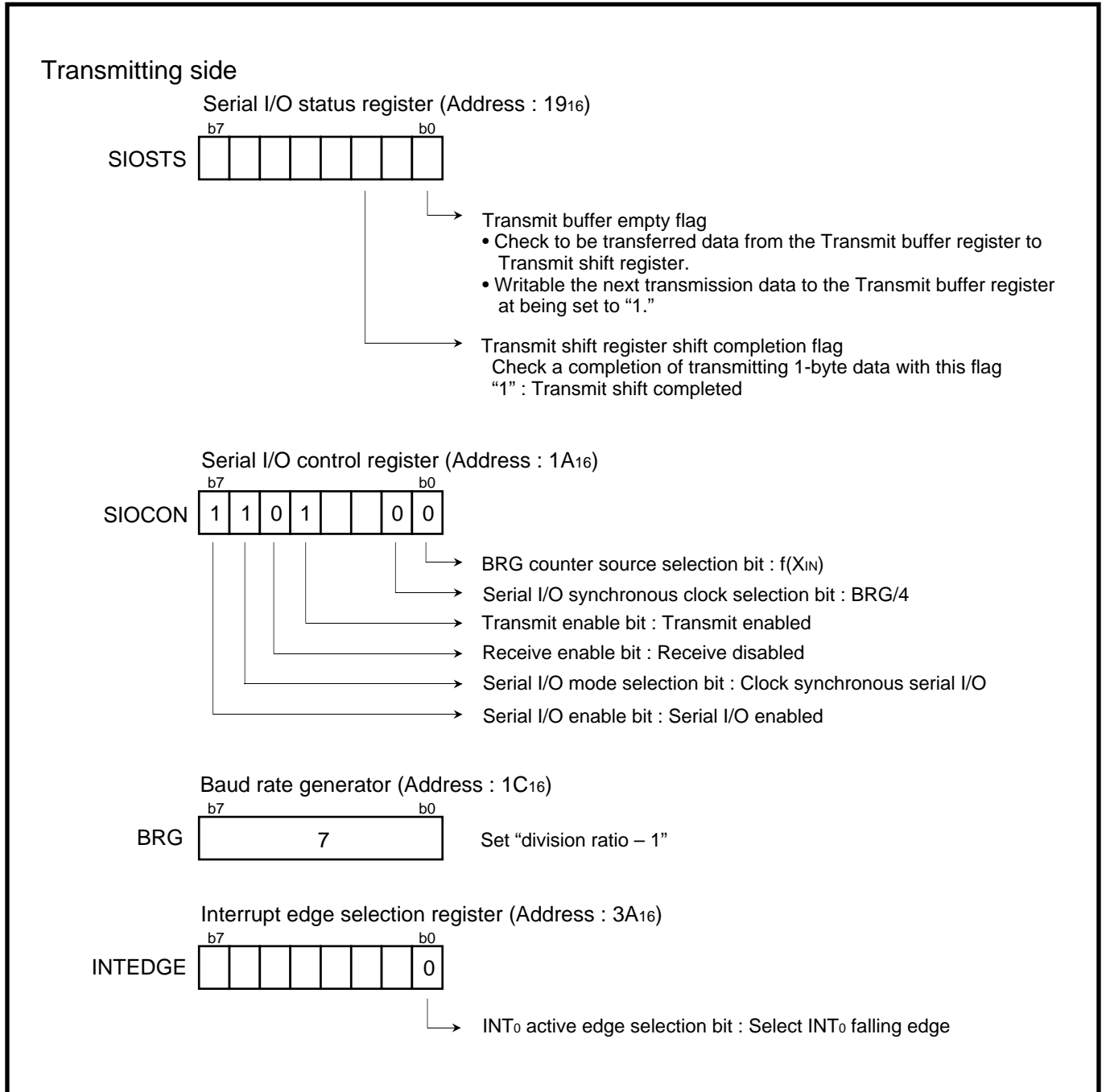
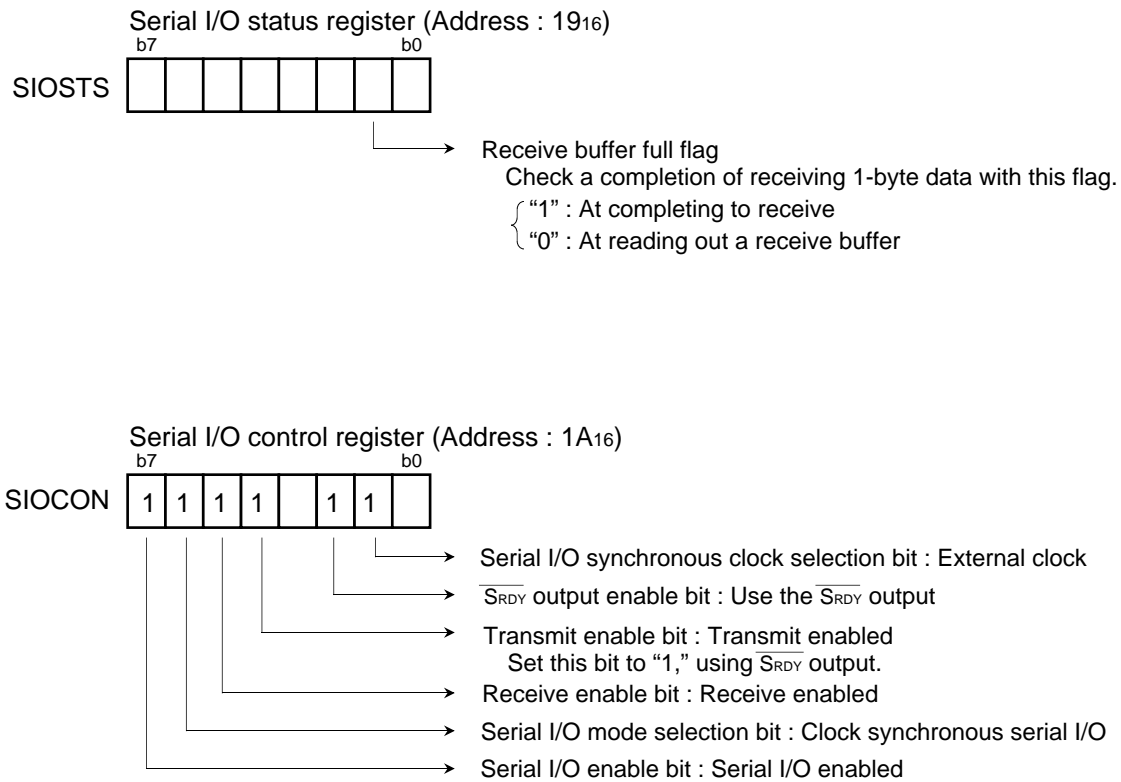


Fig. 2.3.15 Setting of related registers at a transmitting side [Communication using a clock synchronous serial I/O]



Receiving side

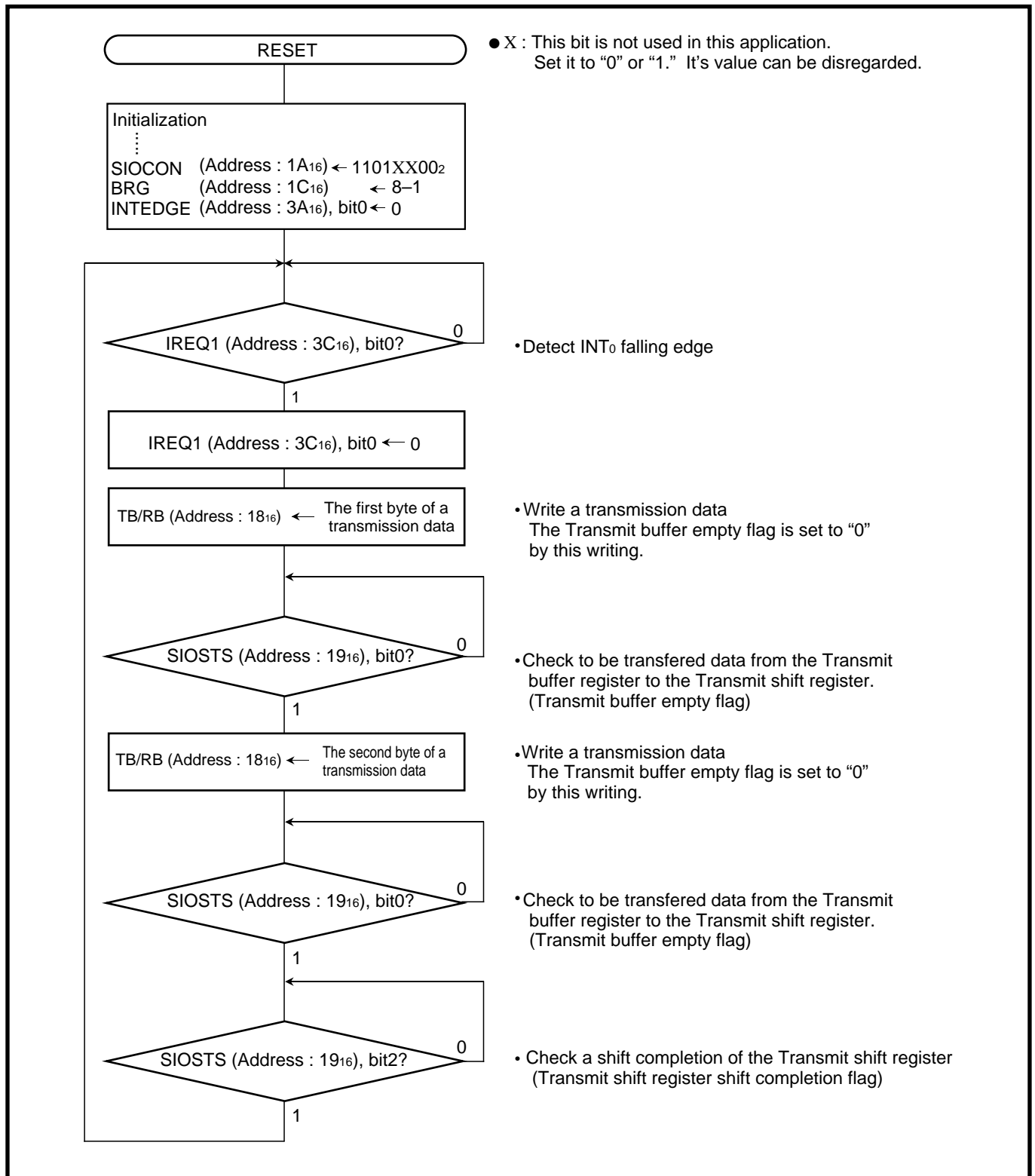


**Fig. 2.3.16 Setting of related registers at a receiving side [Communication using a clock synchronous serial I/O]**

# APPLICATION

## 2.3 Serial I/O

**Control procedure :** Figure 2.3.17 shows a control procedure at a transmitting side, and Figure 2.3.18 shows a control procedure at a receiving side.



**Fig. 2.3.17 Control procedure at a transmitting side [Communication using a clock synchronous serial I/O]**

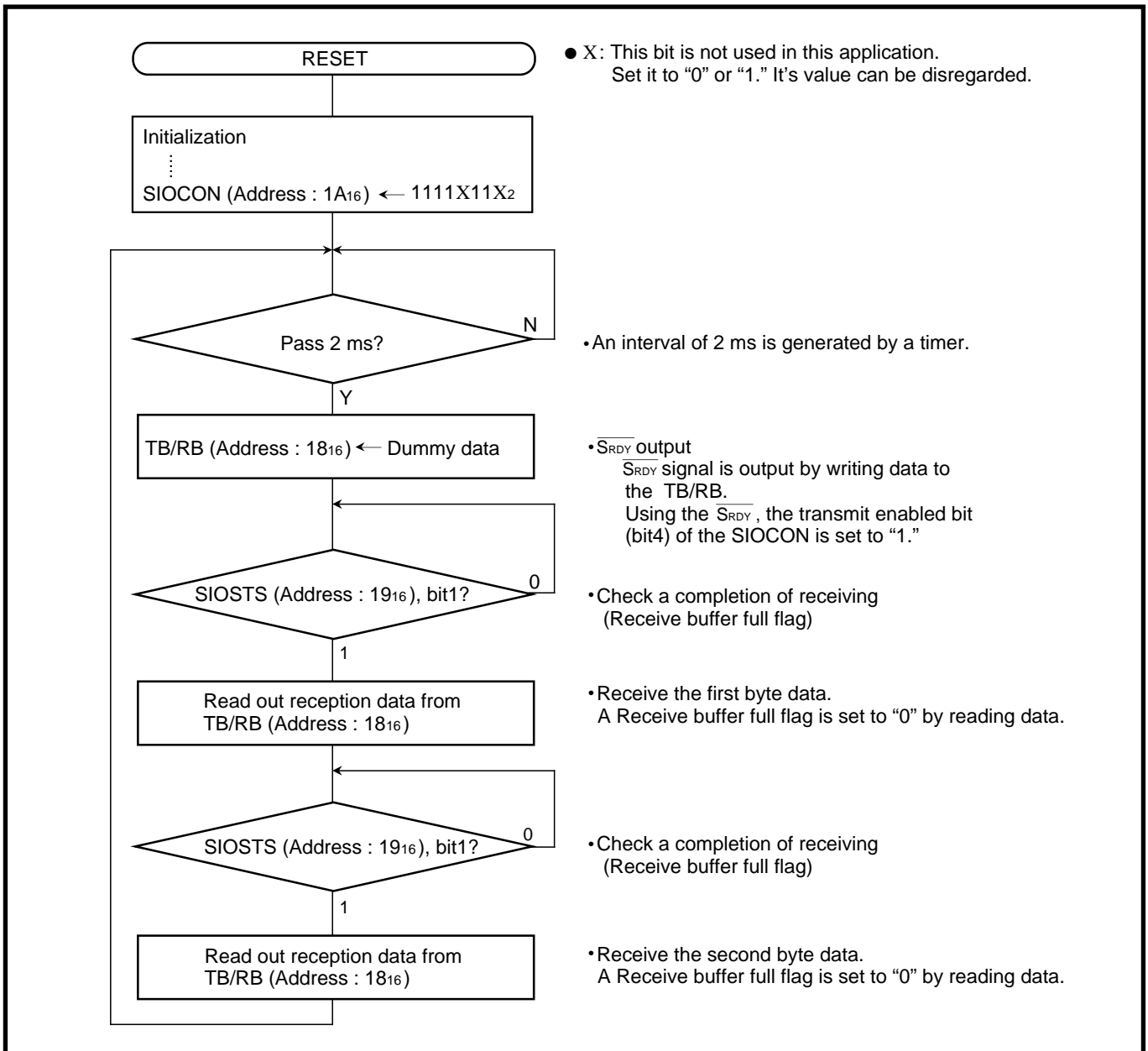


Fig. 2.3.18 Control procedure at a receiving side [Communication using a clock synchronous serial I/O]

# APPLICATION

## 2.3 Serial I/O

### (2) Output of serial data (control of a peripheral IC)

**Outline** : 4-byte data is transmitted and received through the clock synchronous serial I/O. The CS signal is output to a peripheral IC through the port P53.

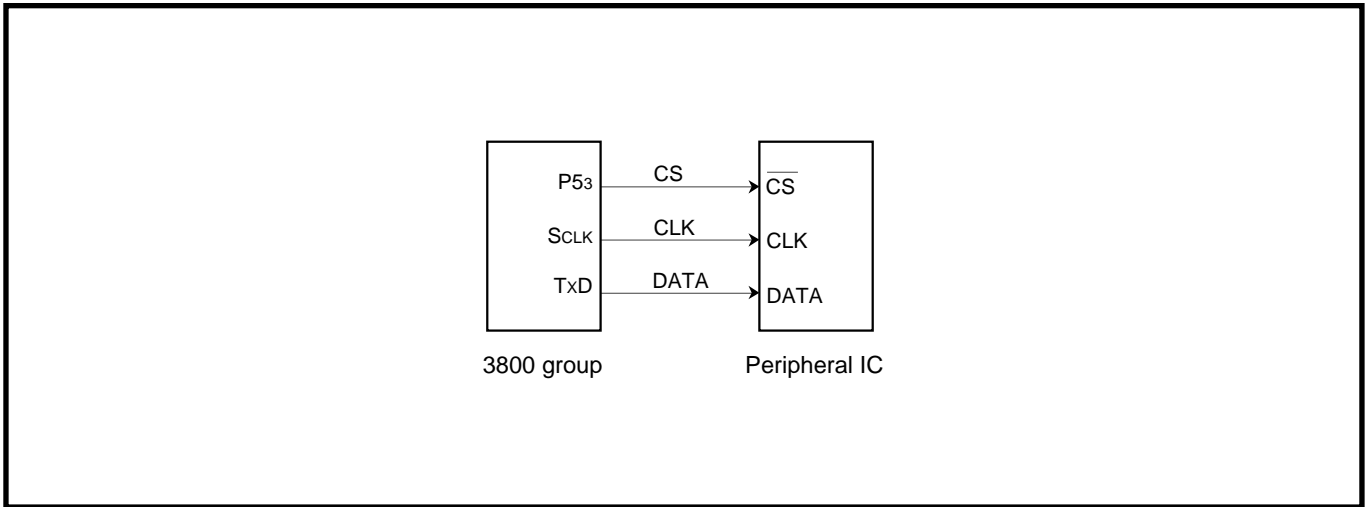


Fig. 2.3.19 Connection diagram [Output of serial data]

- Specifications** :
- The Serial I/O is used. (clock synchronous serial I/O is selected)
  - Synchronous clock frequency : 125 kHz ( $f(XIN) = 4 \text{ MHz}$  is divided by 32)
  - Transfer direction : LSB first
  - The Serial I/O1 interrupt is not used.
  - The Port P53 is connected to the  $\overline{CS}$  pin ("L" active) of the peripheral IC for a transmission control (the output level of the port P53 is controlled by software).

Figure 2.3.20 shows an output timing chart of serial data.

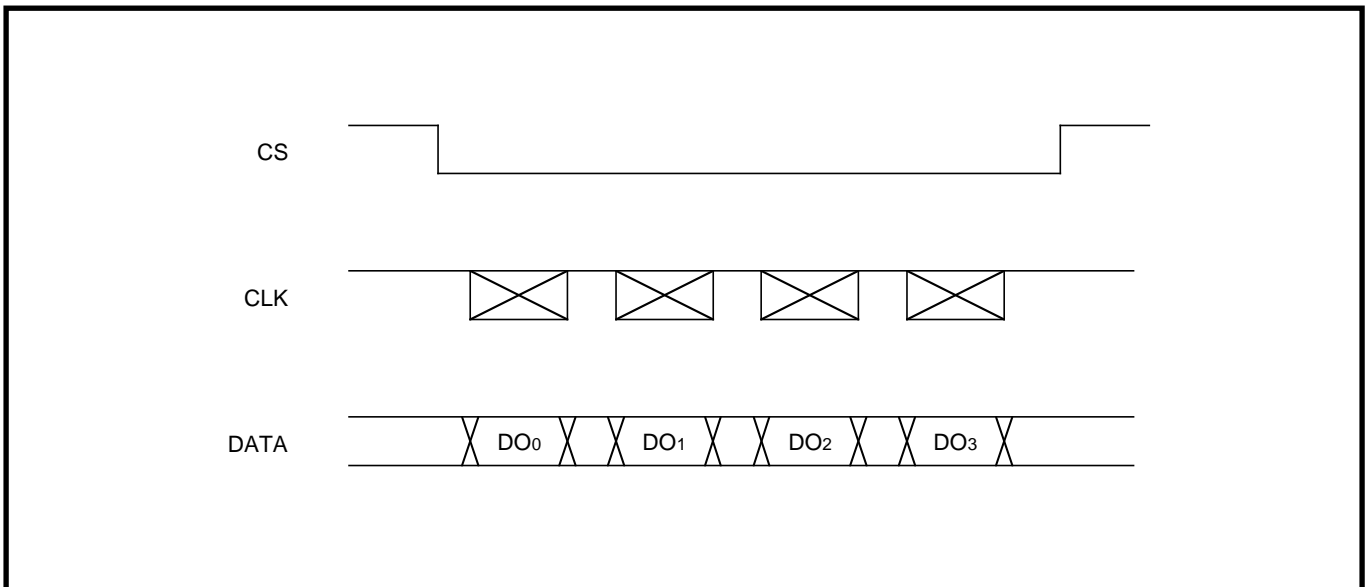


Fig. 2.3.20 Timing chart [Output of serial data]

Figure 2.3.21 shows a setting of serial I/O related registers, and Figure 2.3.22 shows a setting of serial I/O transmission data.

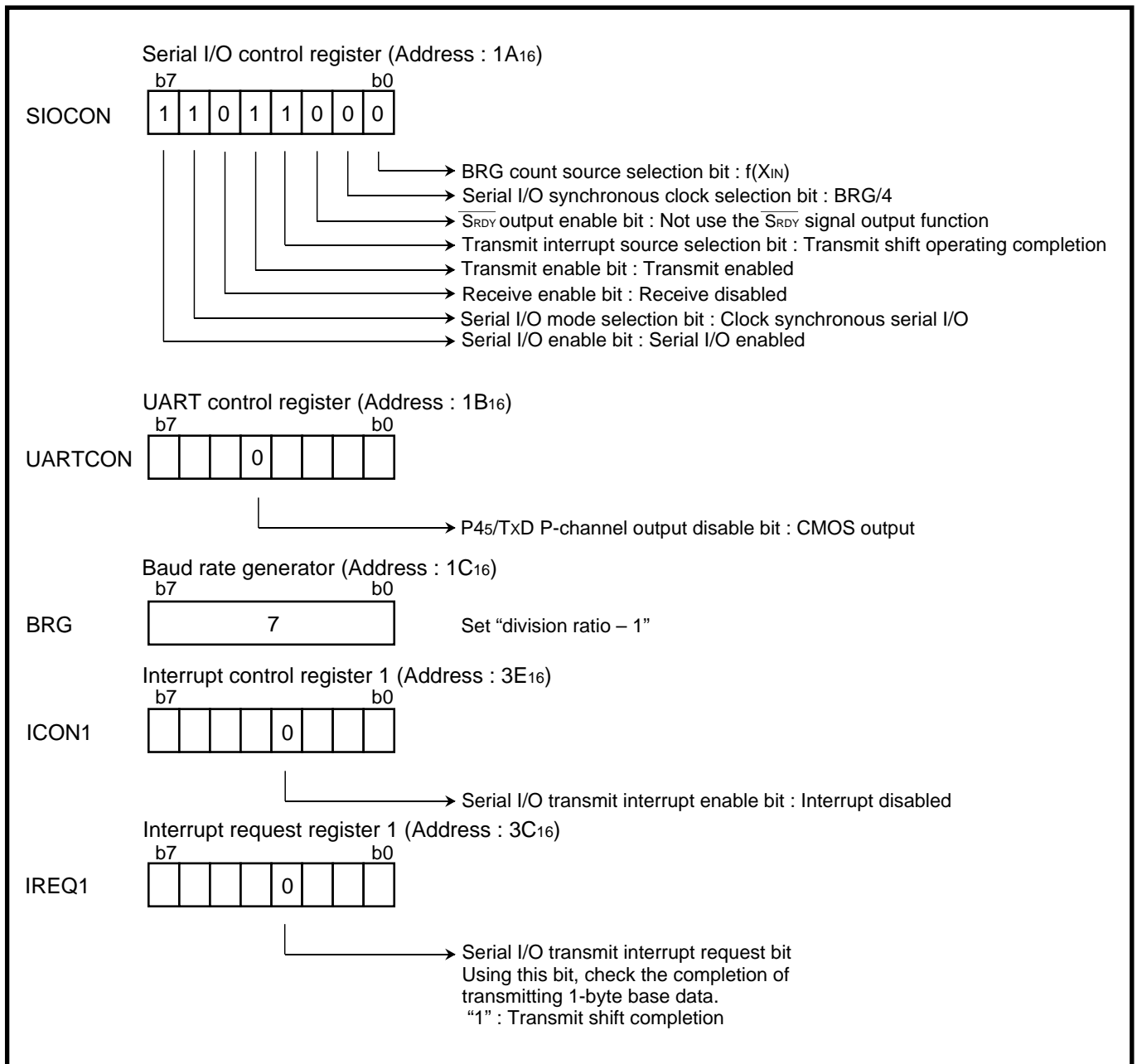


Fig. 2.3.21 Setting of serial I/O related registers [Output of serial data]

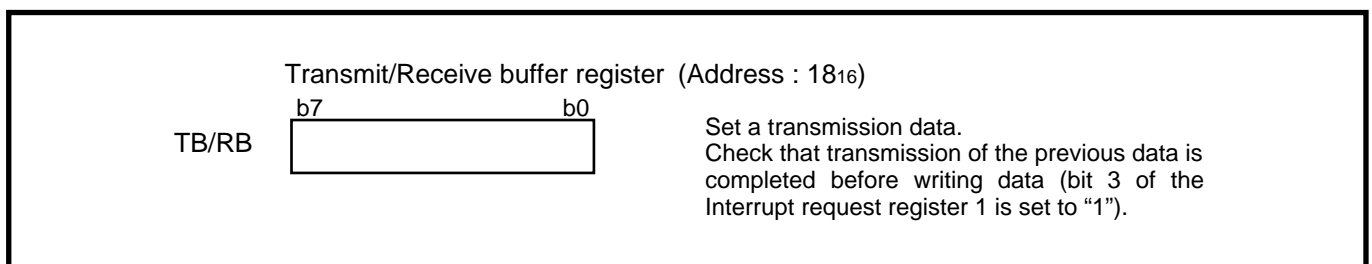


Fig. 2.3.22 Setting of serial I/O transmission data [Output of serial data]

# APPLICATION

## 2.3 Serial I/O

**Control procedure :** When the registers are set as shown in Figure 2.3.21, the Serial I/O can transmit 1-byte data simply by writing data to the Transmit buffer register. Thus, after setting the CS signal to “L,” write the transmission data to the Receive buffer register on a 1-byte base, and return the CS signal to “H” when the desired number of bytes have been transmitted. Figure 2.3.23 shows a control procedure of serial I/O.

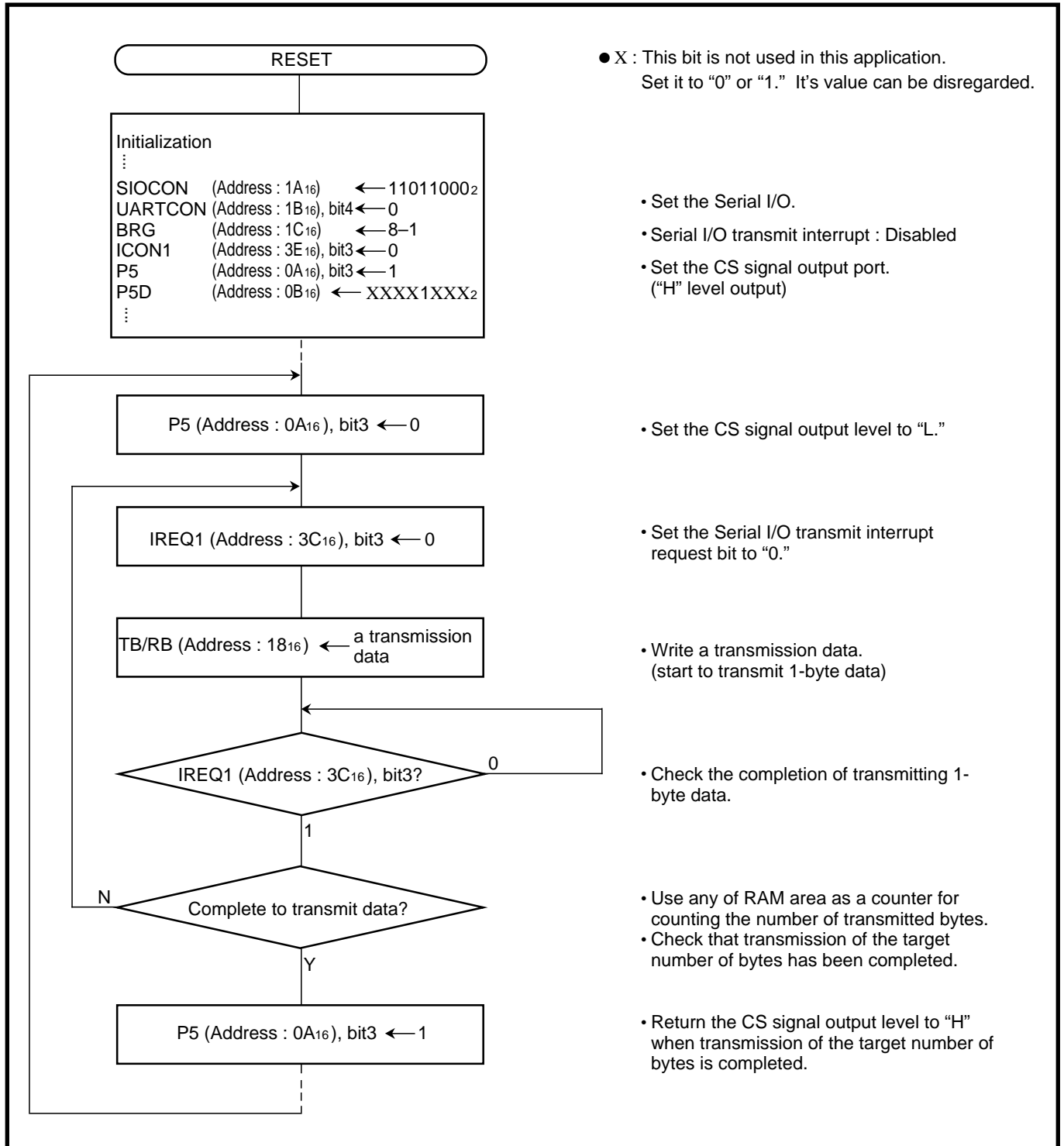
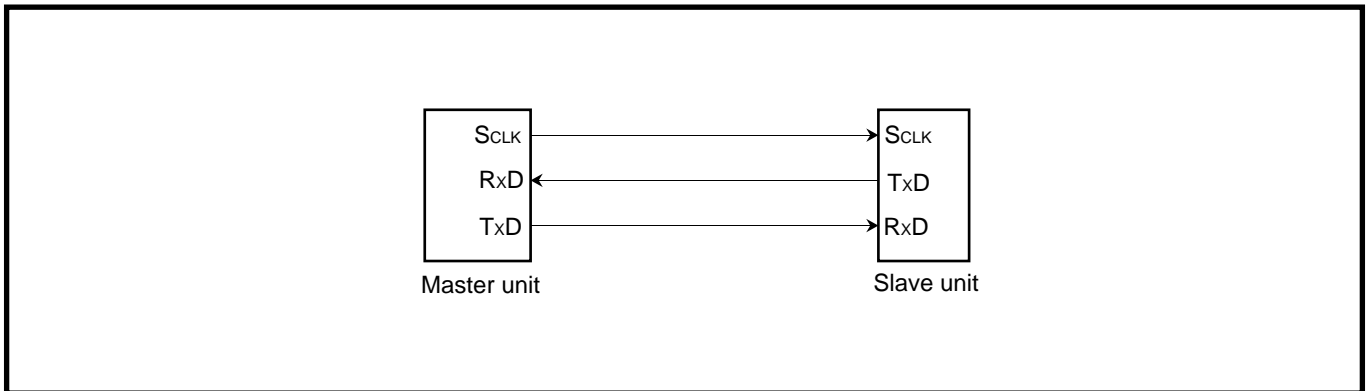


Fig. 2.3.23 Control procedure of serial I/O [Output of serial data]

**(3) Cyclic transmission or reception of block data (data of a specified number of bytes) between microcomputers [without using an automatic transfer]**

**Outline :** When a 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 synchronizing clock. Thus, it is necessary to be corrected constantly. This “heading adjustment” is carried out by using the interval between blocks in this example.



**Fig. 2.3.24 Connection diagram [Cyclic transmission or reception of block data between microcomputers]**

- Specifications :**
- The serial I/O is used (clock synchronous serial I/O is selected).
  - Synchronous clock frequency : 131 kHz ( $f(XIN) = 4.19 \text{ MHz}$  is divided by 32)
  - Byte cycle: 488  $\mu\text{s}$
  - Number of bytes for transmission or reception : 8 byte/block
  - Block transfer cycle : 16 ms
  - Block transfer period : 3.5 ms
  - Interval between blocks : 12.5 ms
  - Heading adjustive time : 8 ms

**Limitations of the specifications**

1. Reading of the reception data and setting of the next transmission data must be completed within the time obtained from “byte cycle – time for transferring 1-byte data” (in this example, the time taken from generating of the Serial I/O receive interrupt request to generating of the next synchronizing clock is 431  $\mu\text{s}$ ).
2. “Heading adjustive time < interval between blocks” must be satisfied.





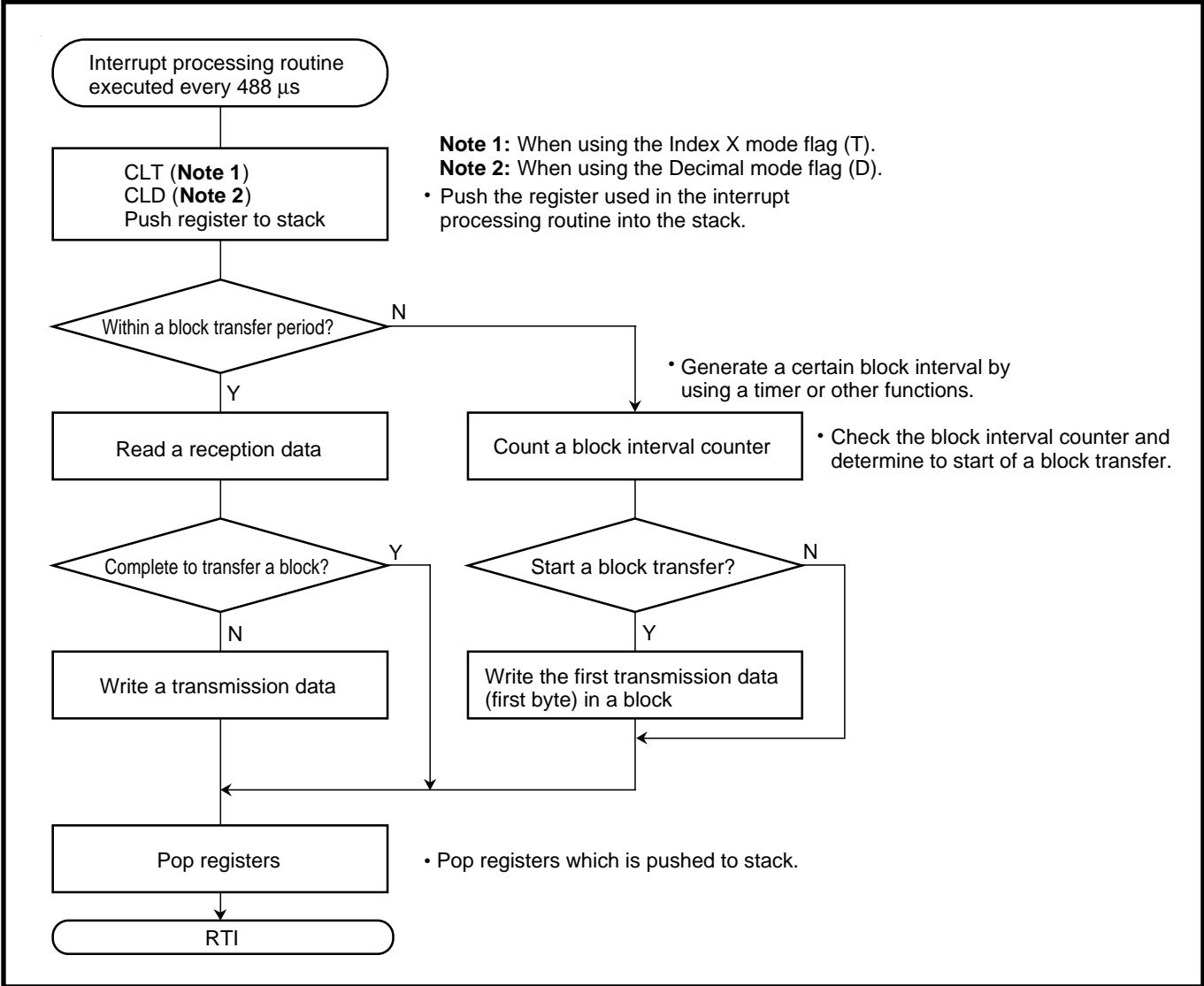
**Control procedure :**

① **Control in the master unit**

After a setting of the related registers is completed as shown in Figure 2.3.33, in the master unit transmission or reception of 1-byte data is started simply by writing transmission data to the Transmit buffer register.

To perform the communication in the timing shown in Figure 2.3.25, therefore, take the timing into account and write transmission data. Read out the reception data when the Serial I/O transmit interrupt request bit is set to "1," or before the next transmission data is written to the Transmit buffer register.

A processing example in the master unit using timer interrupts is shown below.



**Fig. 2.3.27 Control in the master unit**

# APPLICATION

## 2.3 Serial I/O

### ② Control in the slave unit

After a setting of the related registers is completed as shown in Figure 2.3.26, the slave unit becomes the state which is received a synchronizing clock at all times, and the Serial I/O receive interrupt request bit is set to "1" every time an 8-bit synchronous clock is received.

By the serial I/O receive interrupt processing routine, the data to be transmitted next is written to the Transmit buffer register after received data is read out.

However, if no serial I/O receive interrupt occurs for more than a certain time (head adjustive time), the following processing will be performed.

1. The first 1 byte data of the transmission data in the block is written into the Transmit buffer register.
2. The data to be received next is processed as the first 1 byte of the received data in the block.

Figure 2.3.28 shows the control in the slave unit using a serial I/O receive interrupt and any timer interrupt (for head adjustive).

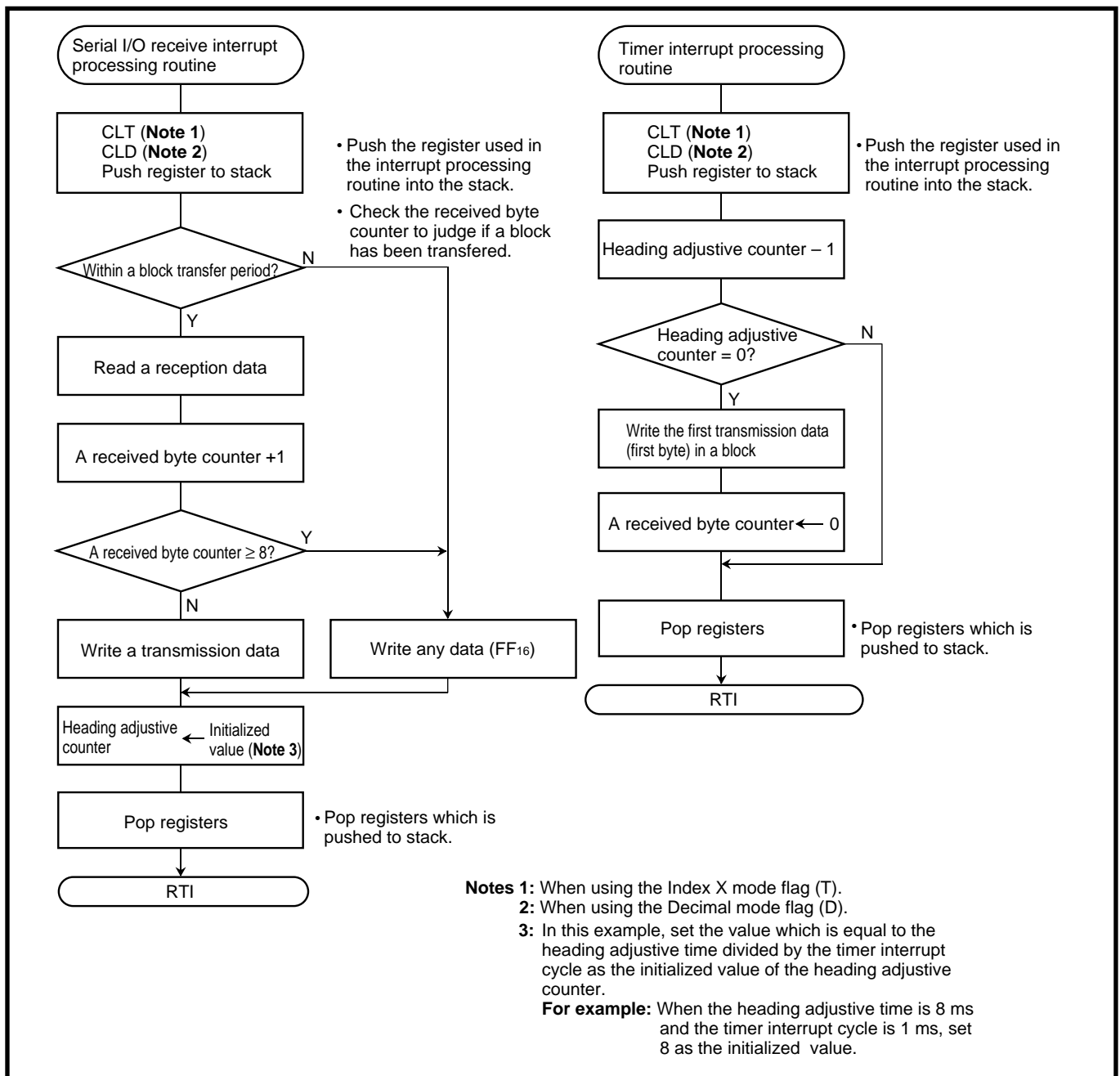
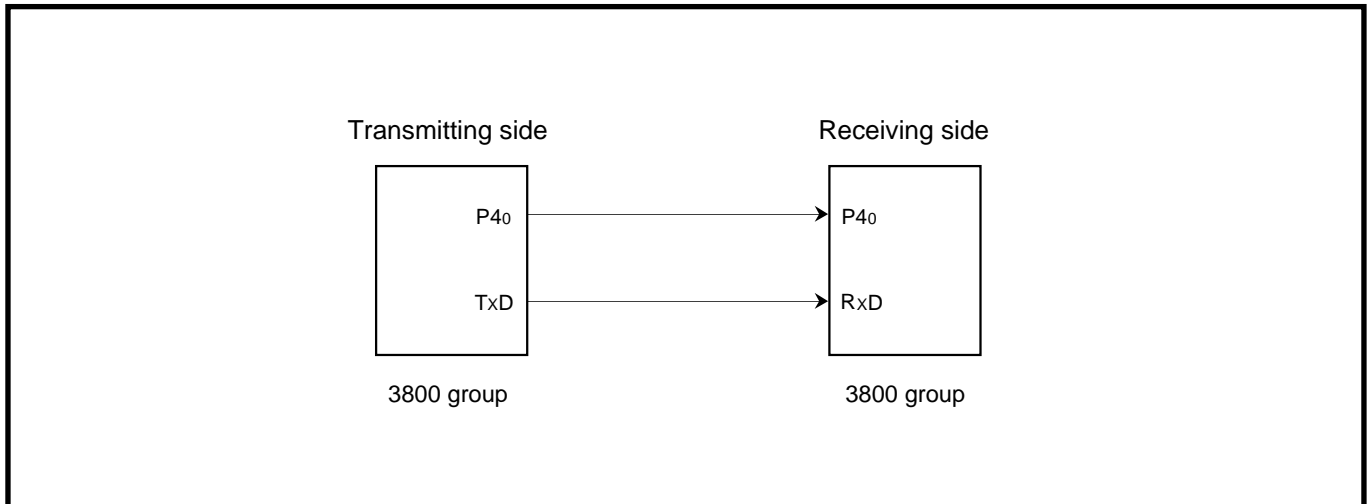


Fig. 2.3.28 Control in the slave unit

**(4) Communication (transmit/receive) using an asynchronous serial I/O (UART)**

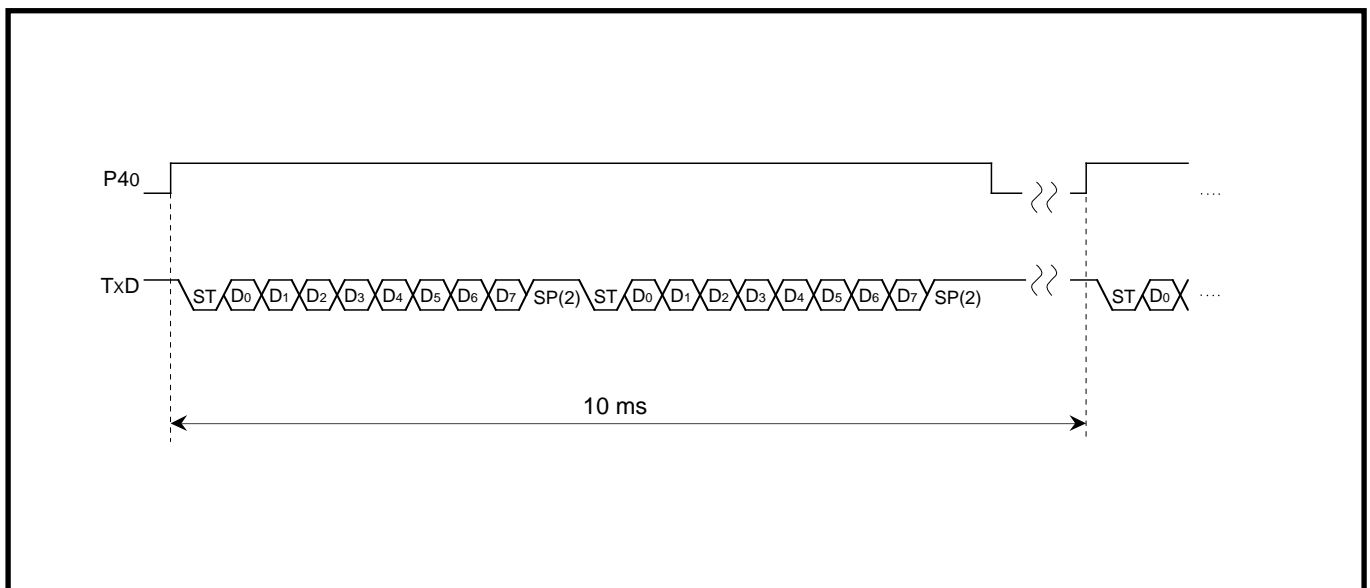
**Point** : 2-byte data is transmitted and received through an asynchronous serial I/O.  
The port P40 is used for communication control.

Figure 2.3.29 shows a connection diagram, and Figure 2.3.30 shows a timing chart.



**Fig. 2.3.29 Connection diagram [Communication using UART]**

- Specifications :**
- The Serial I/O is used (UART is selected).
  - Transfer bit rate : 9600 bps ( $f(X_{IN}) = 4.9152 \text{ MHz}$  is divided by 512)
  - Communication control using port P40  
(The output level of the port P40 is controlled by software.)
  - 2-byte data is transferred from the transmitting side to the receiving side at intervals of 10 ms (generated by timer).



**Fig. 2.3.30 Timing chart [Communication using UART]**

# APPLICATION

## 2.3 Serial I/O

Table 2.3.1 shows setting examples of Baud rate generator (BRG) values and transfer bit rate values, Figure 2.3.31 shows a setting of related registers at a transmitting side, and Figure 2.3.32 shows a setting of related registers at a receiving side.

**Table 2.3.1 Setting examples of Baud rate generator values and transfer bit rate values**

Transfer bit rate (bps) (Note 1)	BRG count source (Note 2)	at f(XIN) = 4.9152 MHz		at f(XIN) = 7.3728 MHz		at f(XIN) = 8 MHz	
		BRG setting value	Actual time (bps)	BRG setting value	Actual time (bps)	BRG setting value	Actual time (bps)
600	f(XIN)/4	127(7F <sub>16</sub> )	600.00	191(BF <sub>16</sub> )	600.00	207(CF <sub>16</sub> )	600.96
1200	f(XIN)/4	63(3F <sub>16</sub> )	1200.00	95(5F <sub>16</sub> )	1200.00	103(67 <sub>16</sub> )	1201.92
2400	f(XIN)/4	31(1F <sub>16</sub> )	2400.00	47(2F <sub>16</sub> )	2400.00	51(33 <sub>16</sub> )	2403.85
4800	f(XIN)/4	15(0F <sub>16</sub> )	4800.00	23(17 <sub>16</sub> )	4800.00	25(19 <sub>16</sub> )	4807.69
9600	f(XIN)/4	7(07 <sub>16</sub> )	9600.00	11(0B <sub>16</sub> )	9600.00	12(0C <sub>16</sub> )	9615.38
19200	f(XIN)/4	3(03 <sub>16</sub> )	19200.00	5(05 <sub>16</sub> )	19200.00	5(05 <sub>16</sub> )	20833.33
38400	f(XIN)/4	1(01 <sub>16</sub> )	38400.00	2(02 <sub>16</sub> )	38400.00	2(02 <sub>16</sub> )	41666.67
76800	f(XIN)	3(03 <sub>16</sub> )	76800.00	5(05 <sub>16</sub> )	76800.00	5(05 <sub>16</sub> )	83333.33
31250	f(XIN)	————	————	————	————	15(0F <sub>16</sub> )	31250.00
62500	f(XIN)	————	————	————	————	7(07 <sub>16</sub> )	62500.00

**Notes 1:** Equation of transfer bit rate

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

m: when bit 0 of the Serial I/O control register (Address : 1A<sub>16</sub>) is set to “0,” a value of m is 1.  
when bit 0 of the Serial I/O control register (Address : 1A<sub>16</sub>) is set to “1,” a value of m is 4.

**2:** A BRG count source is selected by bit 0 of the Serial I/O control register (Address : 1A<sub>16</sub>).

## Transmitting side

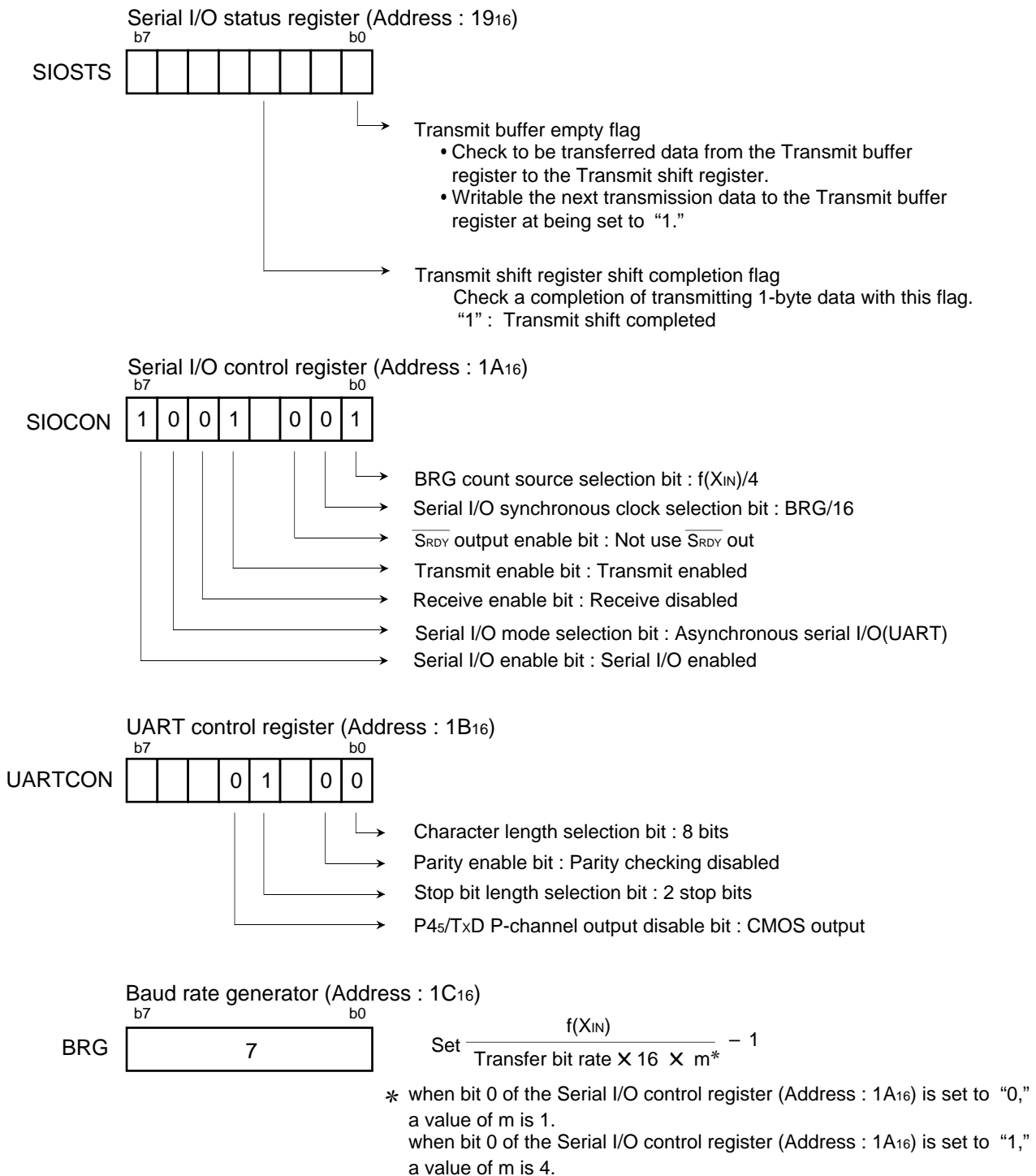


Fig. 2.3.31 Setting of related registers at a transmitting side [Communication using UART]

# APPLICATION

## 2.3 Serial I/O

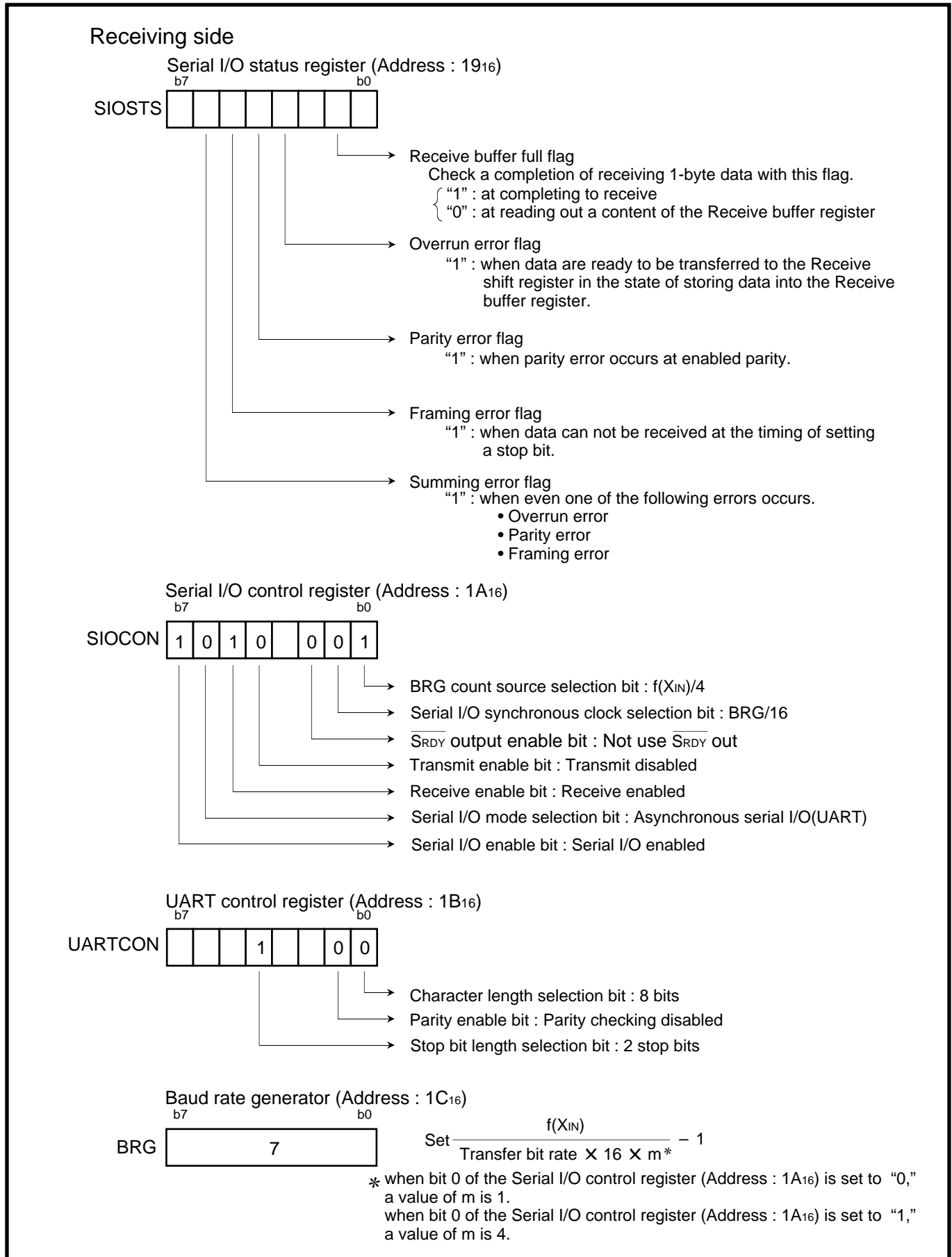
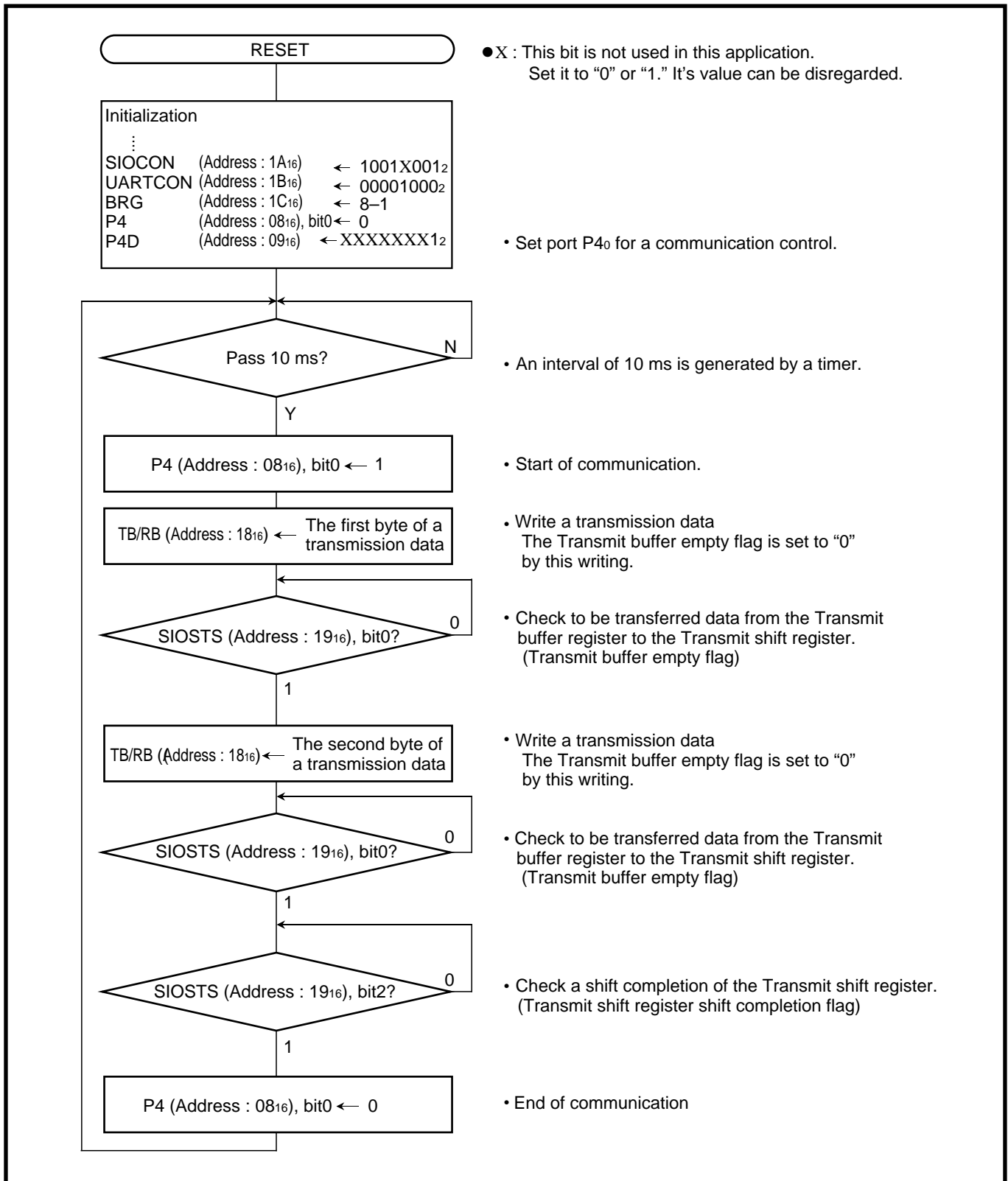


Fig. 2.3.32 Setting of related registers at a receiving side [Communication using UART]

**Control procedure** : Figure 2.3.33 shows a control procedure at a transmitting side, and Figure 2.3.34 shows a control procedure at a receiving side.



**Fig. 2.3.33 Control procedure at a transmitting side [Communication using UART]**

# APPLICATION

## 2.3 Serial I/O

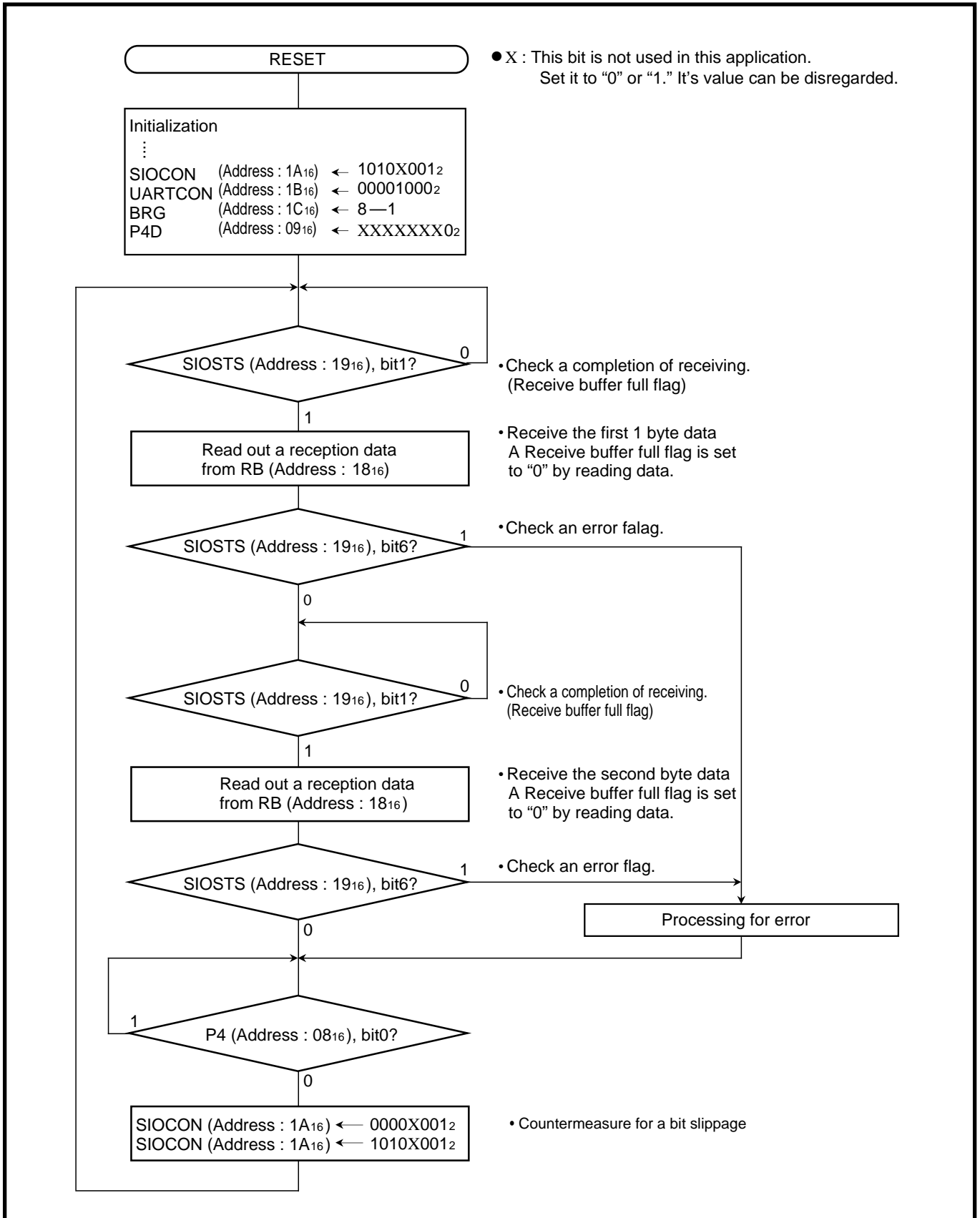


Fig. 2.3.34 Control procedure at a receiving side [Communication using UART]



2.4 Processor mode

2.4.1 Memory map of processor mode

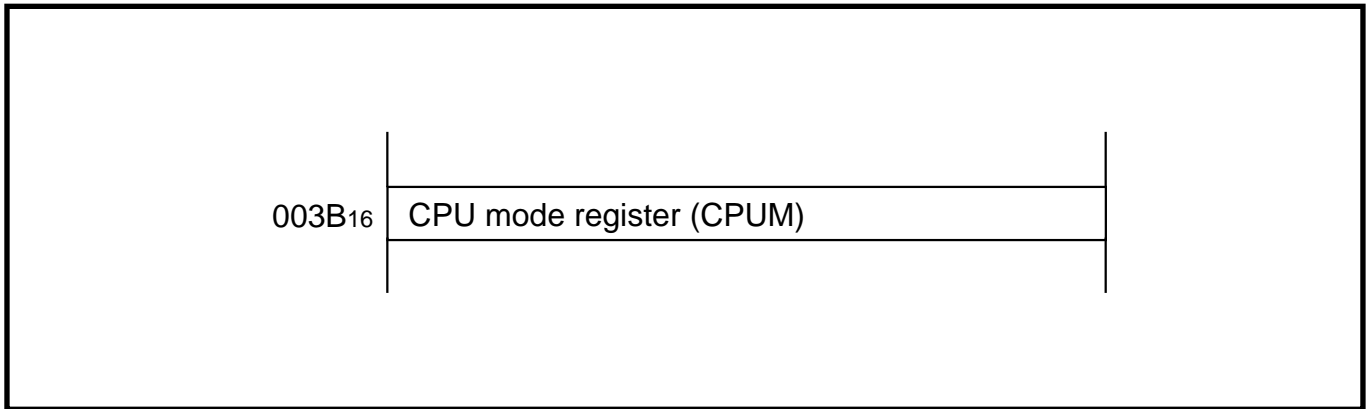


Fig. 2.4.1 Memory map of processor mode related register

2.4.2 Related register

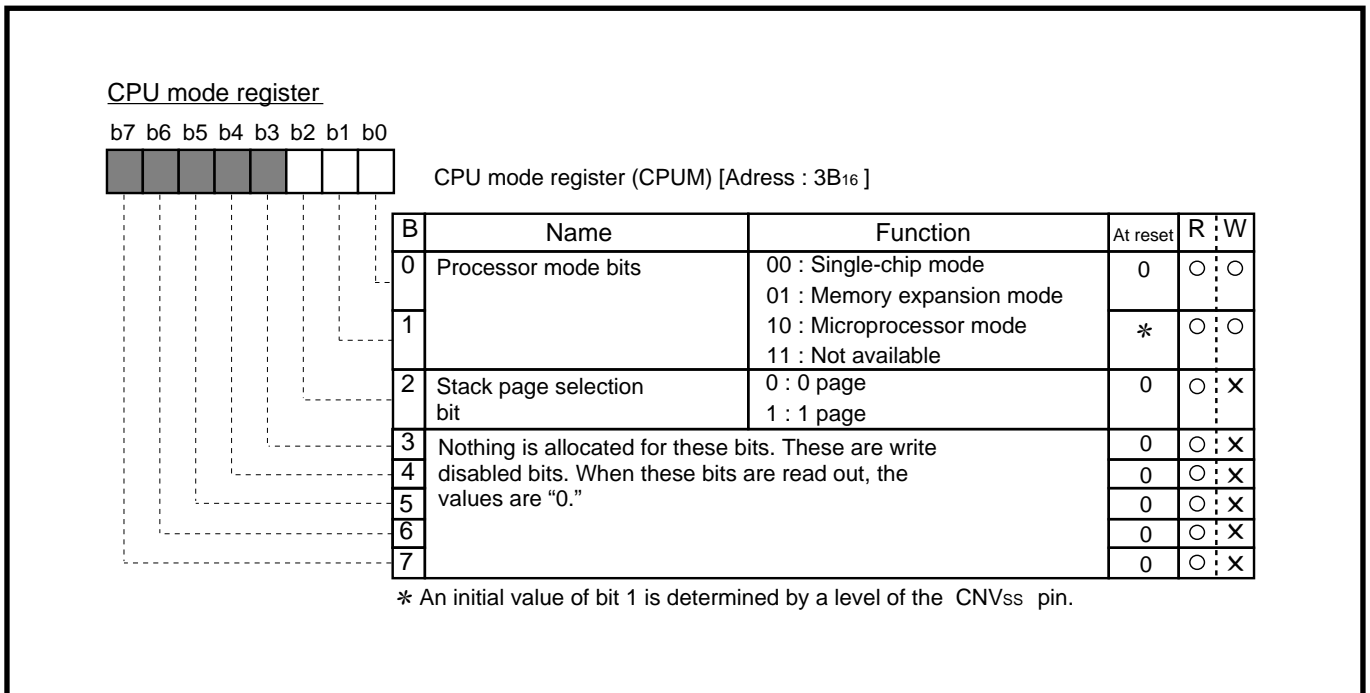


Fig. 2.4.2 Structure of CPU mode register

# APPLICATION

## 2.4 Processor mode

### 2.4.3 Processor mode application examples

#### (1) Application example of memory expansion in the case where the $\overline{\text{ONW}}$ (One-Wait) function is not used

**Outline :** The external memory is accessed in the microprocessor mode.

At  $f(\text{XIN}) = 8 \text{ MHz}$ , an available RAM is given by the following :

- $\overline{\text{OE}}$  access time :  $t_a (\text{OE}) \leq 50 \text{ ns}$
- Setup time for writing data :  $t_{\text{su}} (\text{D}) \leq 65 \text{ ns}$

For example, the M5M5256BP-10 whose address access is 100 ns is available.

Figure 2.4.3 shows an expansion example of a 32K byte ROM and a 32K byte RAM.

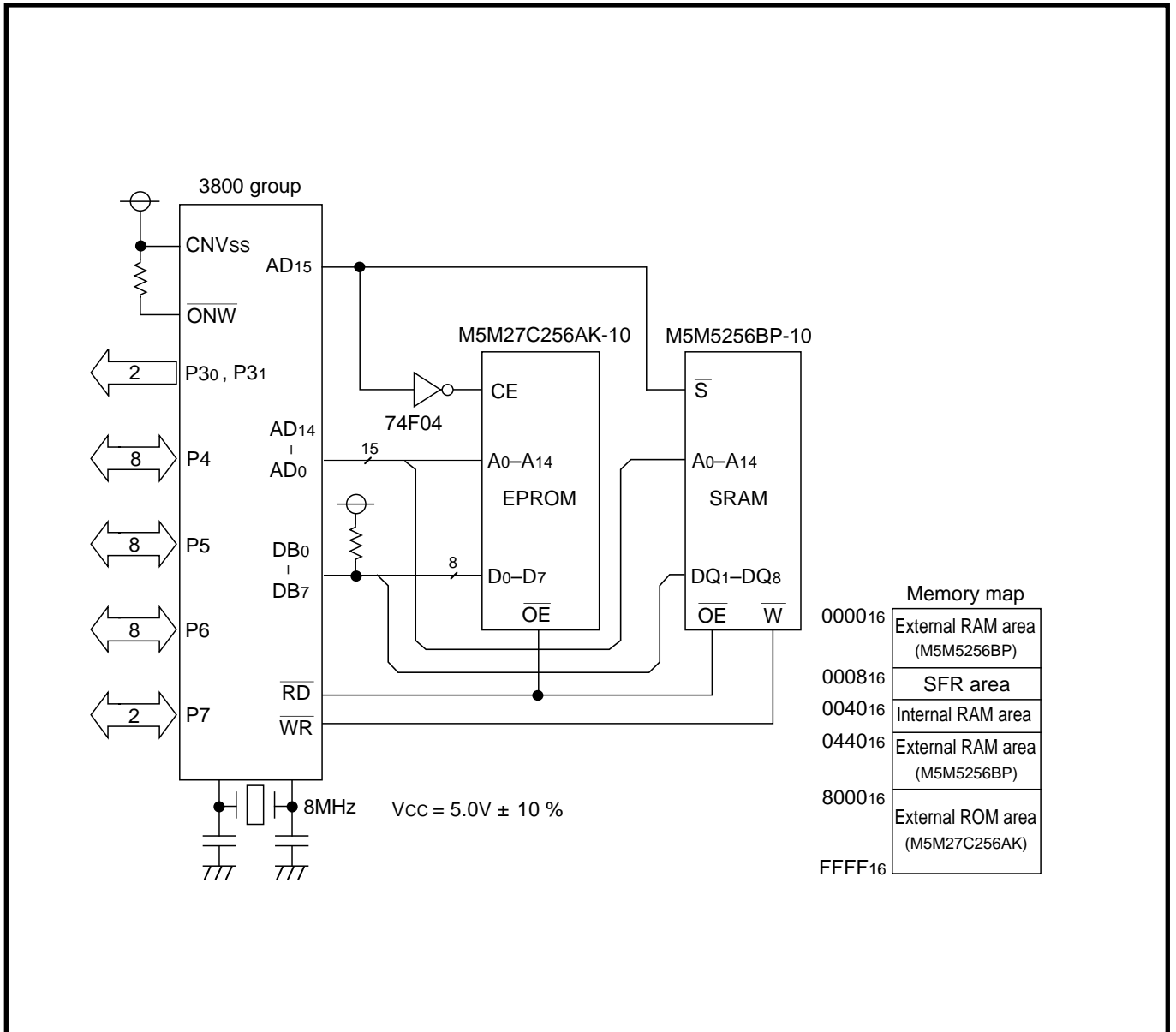


Fig. 2.4.3 Expansion example of ROM and RAM

Figure 2.4.4, Figure 2.4.5 and Figure 2.4.6 show a standard timing at 8 MHz (No-Wait).

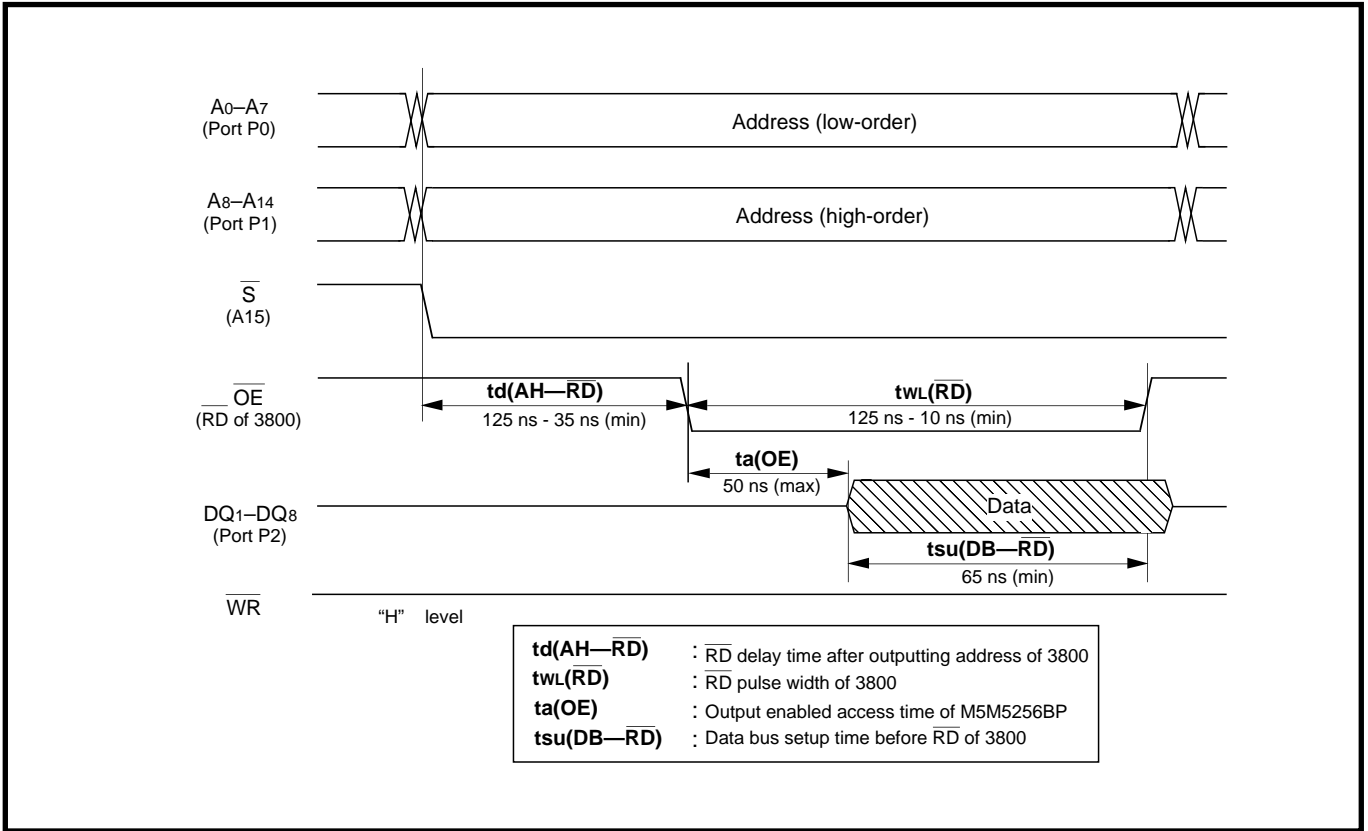


Fig. 2.4.4 Read-cycle (OE access, SRAM)

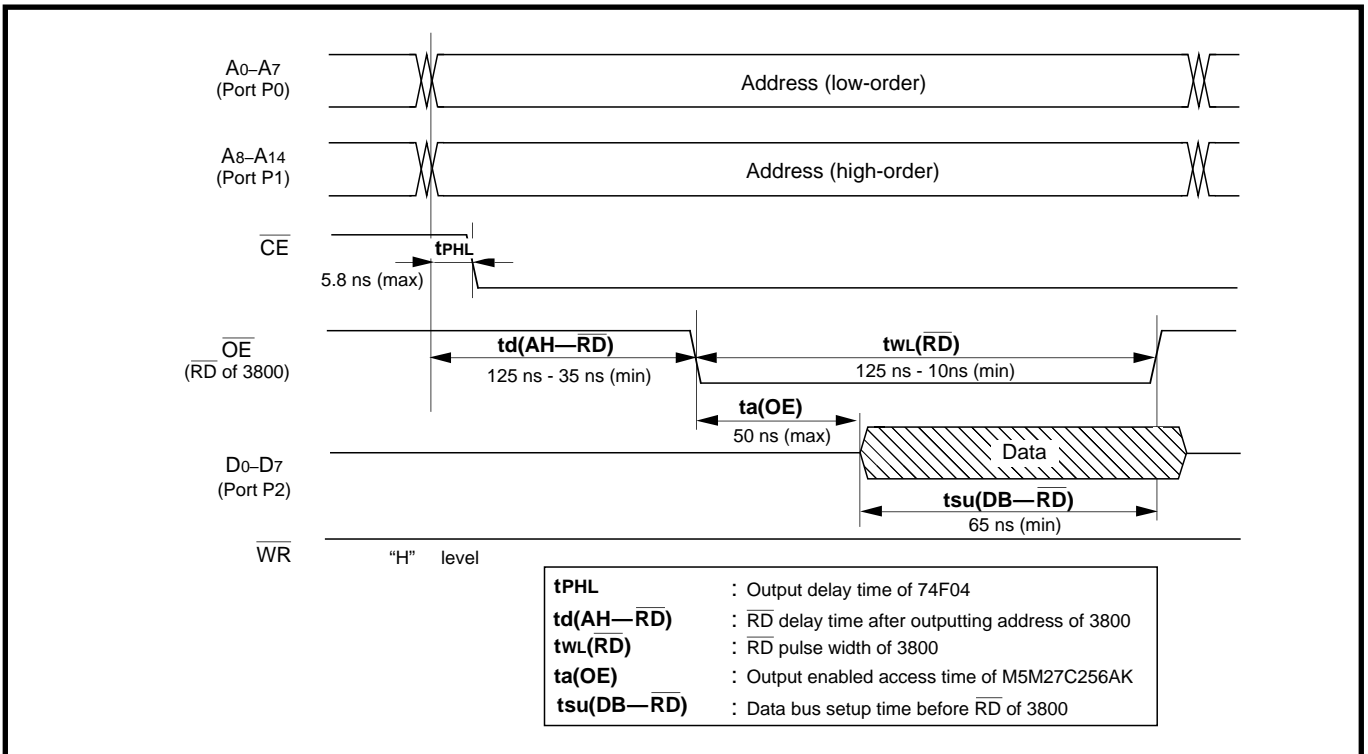


Fig. 2.4.5 Read-cycle (OE access, EPROM)

# APPLICATION

## 2.4 Processor mode

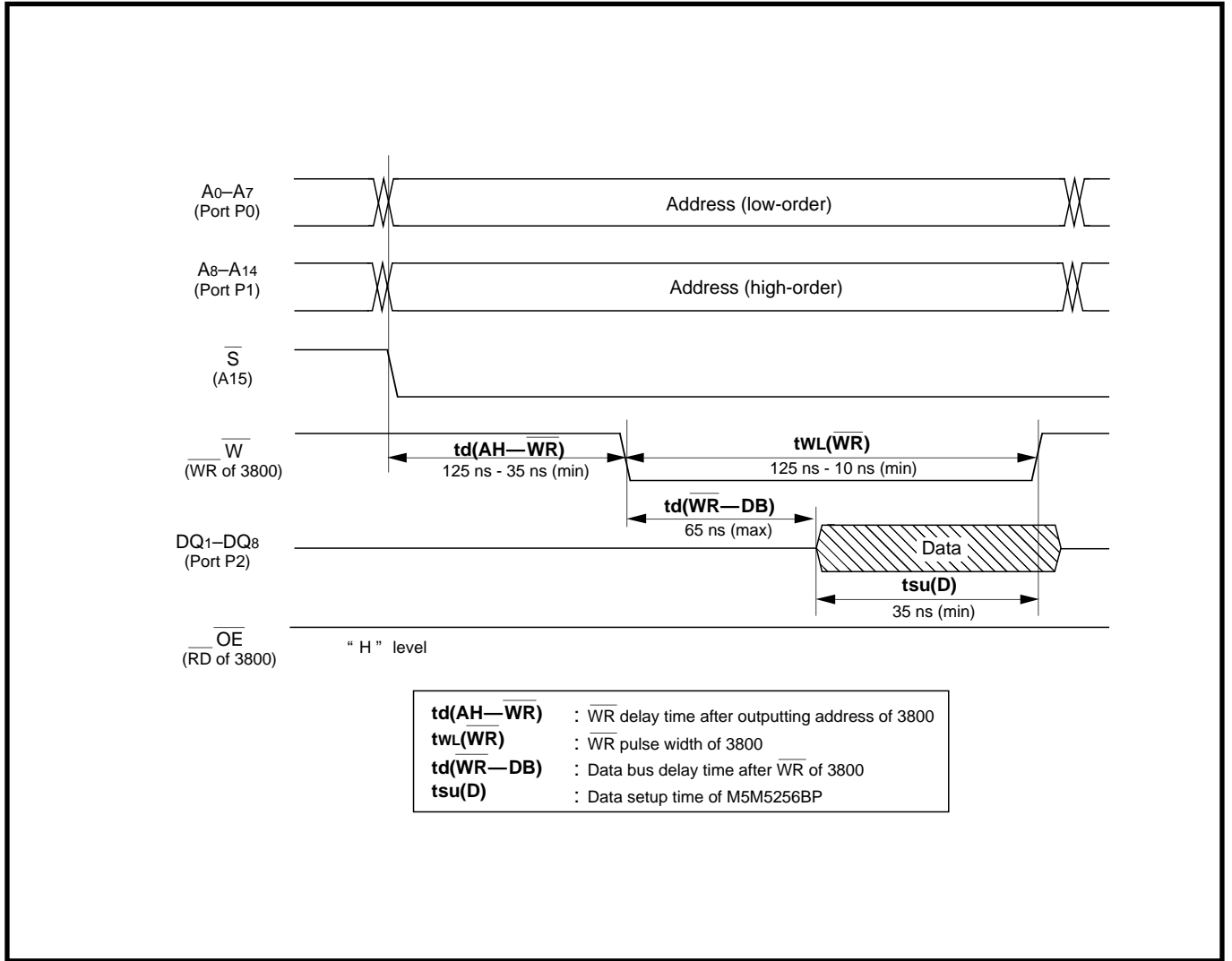


Fig. 2.4.6 Write-cycle (W control, SRAM)

### (2) Application example of memory expansion in the case where the $\overline{ONW}$ (One-Wait) function is used

**Outline :**  $\overline{ONW}$  function is used when the external memory access is slow.

If "L" level signal is input to the P32/ $\overline{ONW}$  pin while the CPU is in the read or write status, the read or write cycle corresponding to 1 cycle of  $\phi$  is extended. In the extended period, the  $\overline{RD}$  or  $\overline{WR}$  signal is kept at the "L" level. The  $\overline{ONW}$  function operates only when data is read from or written into addresses 0000<sub>16</sub> to 0007<sub>16</sub> and addresses 0440<sub>16</sub> to FFFF<sub>16</sub>.

Figure 2.4.7 shows an application example of the  $\overline{ONW}$  function.

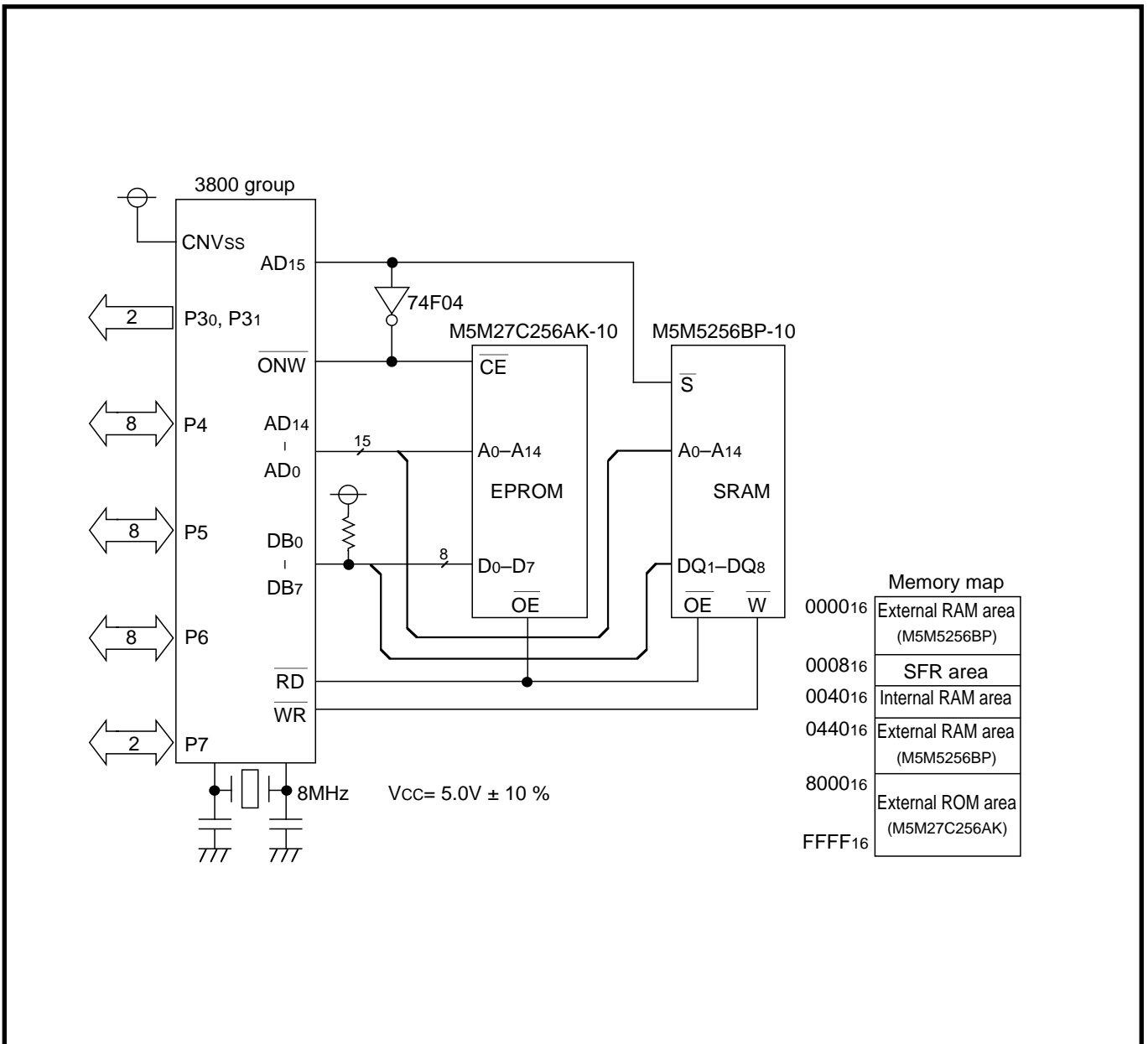


Fig. 2.4.7 Application example of the  $\overline{ONW}$  function

# APPLICATION

## 2.5 Reset

### 2.5 Reset

#### 2.5.1 Connection example of reset IC

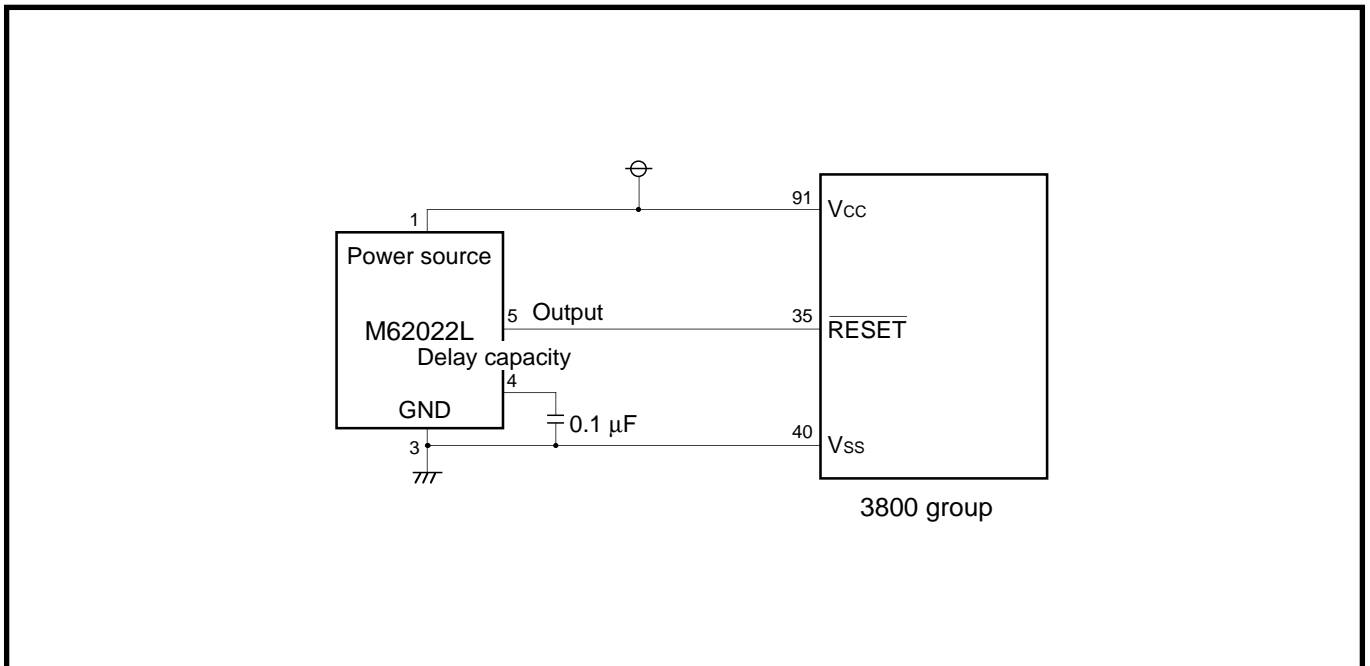


Fig. 2.5.1 Example of Poweron reset circuit

Figure 2.5.2 shows the system example which switch to the RAM backup mode by detecting a drop of the system power source voltage with the INT interrupt.

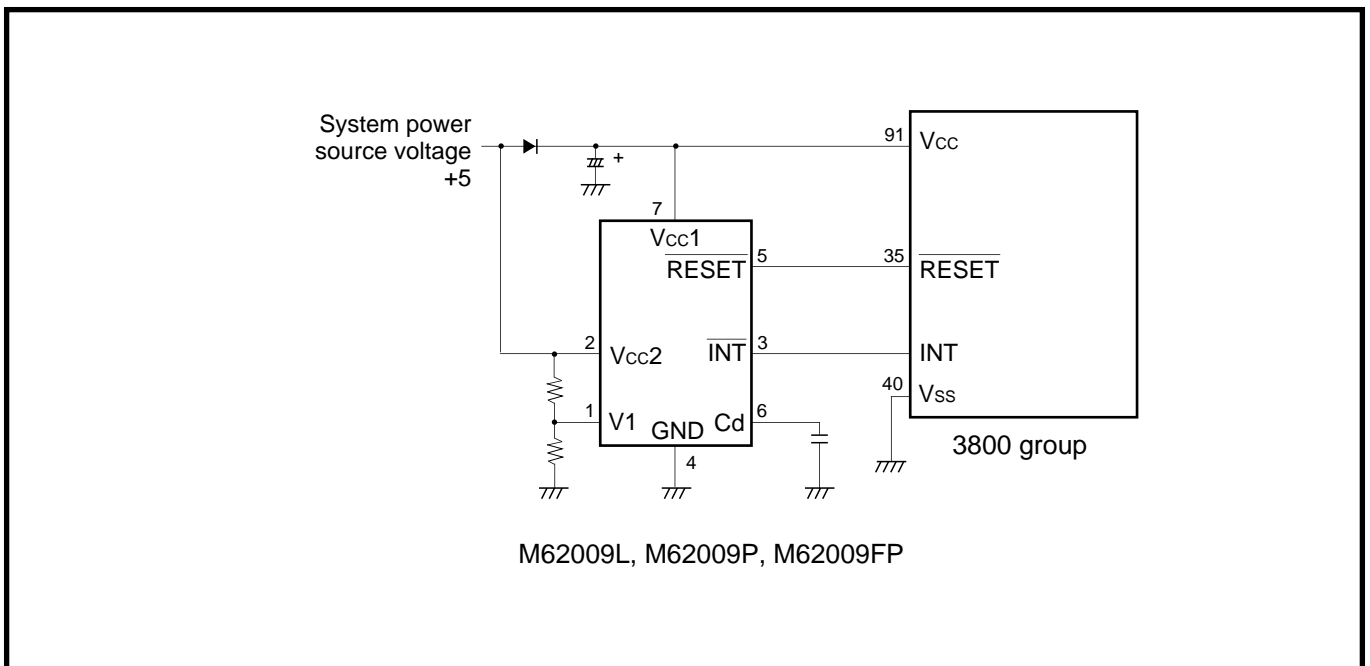


Fig. 2.5.2 RAM back-up system



# CHAPTER 3

## **APPENDIX**

- 3.1 Electrical characteristics
- 3.2 Standard characteristics
- 3.3 Notes on use
- 3.4 Countermeasures against noise
- 3.5 List of registers
- 3.6 Mask ROM ordering method
- 3.7 Mark specification form
- 3.8 Package outline
- 3.9 List of instruction codes
- 3.10 Machine instructions
- 3.11 SFR memory map
- 3.12 Pin configuration

# APPENDIX

## 3.1 Electrical characteristics

### 3.1 Electrical characteristics

#### 3.1.1 Absolute maximum ratings

Table 3.1.1 Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V <sub>CC</sub>	Power source voltage	All voltages are based on V <sub>SS</sub> . Output transistors are cut off.	-0.3 to 7.0	V
V <sub>I</sub>	Input voltage P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71		-0.3 to V <sub>CC</sub> +0.3	V
V <sub>I</sub>	Input voltage RESET, X <sub>IN</sub>		-0.3 to V <sub>CC</sub> +0.3	V
V <sub>I</sub>	Input voltage CNV <sub>SS</sub>		-0.3 to 13	V
V <sub>O</sub>	Output voltage P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71, X <sub>OUT</sub>		-0.3 to V <sub>CC</sub> +0.3	V
P <sub>d</sub>	Power dissipation	T <sub>a</sub> = 25 °C	1000(Note)	mW
T <sub>opr</sub>	Operating temperature		-20 to 85	°C
T <sub>stg</sub>	Storage temperature		-40 to 125	°C

Note : 300 mW in case of the flat package.

#### 3.1.2 Recommended operating conditions

Table 3.1.2 Recommended operating conditions (V<sub>CC</sub> = 3.0 to 5.5 V, T<sub>a</sub> = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V <sub>CC</sub>	Power source voltage (Note 1)	(f(X <sub>IN</sub> ) ≤ 2 MHz)	3.0	5.0	5.5	V
		(f(X <sub>IN</sub> ) = 8 MHz)	4.0	5.0	5.5	
V <sub>SS</sub>	Power source voltage			0		V
V <sub>IH</sub>	"H" input voltage	P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71	0.8 V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IH</sub>	"H" input voltage	RESET, X <sub>IN</sub> , CNV <sub>SS</sub>	0.8 V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage	P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71	0		0.2 V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage	RESET, CNV <sub>SS</sub>	0		0.2 V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage	X <sub>IN</sub>	0		0.16 V <sub>CC</sub>	V
ΣI <sub>OH(peak)</sub>	"H" total peak output current	P00–P07, P10–P17, P20–P27, P30–P37(Note 2)			-80	mA
ΣI <sub>OH(peak)</sub>	"H" total peak output current	P40–P47, P50–P57, P60–P67, P70, P71(Note 2)			-80	mA
ΣI <sub>OL(peak)</sub>	"L" total peak output current	P00–P07, P10–P17, P20–P27, P30–P37(Note 2)			80	mA
ΣI <sub>OL(peak)</sub>	"L" total peak output current	P40–P47, P50–P57, P60–P67, P70, P71(Note 2)			80	mA
ΣI <sub>OH(avg)</sub>	"H" total average output current	P00–P07, P10–P17, P20–P27, P30–P37(Note 2)			-40	mA
ΣI <sub>OH(avg)</sub>	"H" total average output current	P40–P47, P50–P57, P60–P67, P70, P71(Note 2)			-40	mA
ΣI <sub>OL(avg)</sub>	"L" total average output current	P00–P07, P10–P17, P20–P27, P30–P37(Note 2)			40	mA
ΣI <sub>OL(avg)</sub>	"L" total average output current	P40–P47, P50–P57, P60–P67, P70, P71(Note 2)			40	mA
I <sub>OH(peak)</sub>	"H" peak output current	P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71(Note 3)			-10	mA
I <sub>OL(peak)</sub>	"L" peak output current	P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71(Note 3)			10	mA
I <sub>OH(avg)</sub>	"H" average output current	P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71(Note 4)			-5	mA
I <sub>OL(avg)</sub>	"L" average output current	P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71(Note 4)			5	mA
f(X <sub>IN</sub> )	Internal clock oscillation frequency (4.0 V ≤ V <sub>CC</sub> ≤ 5.5 V)				8	MHz
	Internal clock oscillation frequency (3.0 V ≤ V <sub>CC</sub> ≤ 4.0 V)				6 V <sub>CC</sub> - 16	

Note 1: The minimum power source voltage is  $\frac{X+16}{6}$  [V] (f(X<sub>IN</sub>) = XMHz) on the condition of 2 MHz < f(X<sub>IN</sub>) < 8 MHz.

2: The total output current is the sum of all the currents flowing through all the applicable ports. The total average current is an average value measured over 100 ms. The total peak current is the peak value of all the currents.

3: The peak output current is the peak current flowing in each port.

4: The average output current I<sub>OL(avg)</sub>, I<sub>OH(avg)</sub> in an average value measured over 100 ms.



## 3.1 Electrical characteristics

### 3.1.3 Electrical characteristics

**Table 3.1.3 Electrical characteristics** ( $V_{CC} = 3.0$  to  $5.5$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85$  °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
VOH	“H” output voltage P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71 (Note)	IOH = –10 mA VCC = 4.0 to 5.5 V	VCC–2.0			V
		IOH = –1.0 mA VCC = 3.0 to 5.5 V	VCC–1.0			
VOL	“L” output voltage P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71	IO <sub>L</sub> = 10 mA VCC = 4.0 to 5.5 V			2.0	V
		IO <sub>L</sub> = 1.0 mA VCC = 3.0 to 5.5 V			1.0	
VT+ – VT–	Hysteresis	CNTR0, CNTR1, INT0–INT5		0.4		V
VT+ – VT–	Hysteresis	RxD, SCLK		0.5		V
VT+ – VT–	Hysteresis	RESET		0.5		V
I <sub>IH</sub>	“H” input current P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71	Vi = VCC			5.0	μA
I <sub>IH</sub>	“H” input current	RESET, CNVss			5.0	μA
I <sub>IH</sub>	“H” input current	XIN		4		μA
I <sub>IL</sub>	“L” input current P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71 RESET, CNVss	Vi = VSS			–5.0	μA
I <sub>IL</sub>	“L” input current	XIN		–4		μA
VRAM	RAM hold voltage	When clock stopped	2.0		5.5	V
I <sub>CC</sub>	Power source current	f(XIN) = 8 MHz, VCC = 5 V		6.4	13	mA
		f(XIN) = 5 MHz, VCC = 5 V		4	8	
		f(XIN) = 2 MHz, VCC = 3 V		0.8	2.0	
		When WIT instruction is executed with f(XIN) = 8 MHz, VCC = 5 V		1.5		
		When WIT instruction is executed with f(XIN) = 5 MHz, VCC = 5 V		1		
		When WIT instruction is executed with f(XIN) = 2 MHz, VCC = 3 V		0.2		
		When STP instruction is executed with clock stopped, output transistors isolated.	Ta = 25 °C		0.1	1
	Ta = 85 °C			10		

**Note :** P45 is measured when the P45/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is “0”.

# APPENDIX

## 3.1 Electrical characteristics

### 3.1.4 Timing requirements and Switching characteristics

**Table 3.1.4 Timing requirements (1)** ( $V_{CC} = 4.0$  to  $5.5$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85$  °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input "L" pulse width	2			$\mu$ s
$t_{c(XIN)}$	External clock input cycle time	125			ns
$t_{WH(XIN)}$	External clock input "H" pulse width	50			ns
$t_{WL(XIN)}$	External clock input "L" pulse width	50			ns
$t_{c(CNTR)}$	CNTR0, CNTR1 input cycle time	200			ns
$t_{WH(CNTR)}$	CNTR0, CNTR1 input "H" pulse width	80			ns
$t_{WL(CNTR)}$	CNTR0, CNTR1 input "L" pulse width	80			ns
$t_{WH(INT)}$	INT0 to INT5 input "H" pulse width	80			ns
$t_{WL(INT)}$	INT0 to INT5 input "L" pulse width	80			ns
$t_{c(SCLK)}$	Serial I/O clock input cycle time (Note)	800			ns
$t_{WH(SCLK)}$	Serial I/O clock input "H" pulse width (Note)	370			ns
$t_{WL(SCLK)}$	Serial I/O clock input "L" pulse width (Note)	370			ns
$t_{su(RxD-SCLK)}$	Serial I/O input set up time	220			ns
$t_{h(SCLK-RxD)}$	Serial I/O input hold time	100			ns

**Note:** When bit 6 of address 001A16 is "1". Divide this value by four when bit 6 of address 001A16 is "0".

**Table 3.1.5 Timing requirements (2)** ( $V_{CC} = 3.0$  to  $4.0$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85$  °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input "L" pulse width	2			$\mu$ s
$t_{c(XIN)}$	External clock input cycle time	500/ (3 $V_{CC}-8$ )			ns
$t_{WH(XIN)}$	External clock input "H" pulse width	200/ (3 $V_{CC}-8$ )			ns
$t_{WL(XIN)}$	External clock input "L" pulse width	200/ (3 $V_{CC}-8$ )			ns
$t_{c(CNTR)}$	CNTR0, CNTR1 input cycle time	500			ns
$t_{WH(CNTR)}$	CNTR0, CNTR1 input "H" pulse width	230			ns
$t_{WL(CNTR)}$	CNTR0, CNTR1 input "L" pulse width	230			ns
$t_{WH(INT)}$	INT0 to INT5 input "H" pulse width	230			ns
$t_{WL(INT)}$	INT0 to INT5 input "L" pulse width	230			ns
$t_{c(SCLK)}$	Serial I/O clock input cycle time (Note)	2000			ns
$t_{WH(SCLK)}$	Serial I/O clock input "H" pulse width (Note)	950			ns
$t_{WL(SCLK)}$	Serial I/O clock input "L" pulse width (Note)	950			ns
$t_{su(RxD-SCLK)}$	Serial I/O input set up time	400			ns
$t_{h(SCLK-RxD)}$	Serial I/O input hold time	200			ns

**Note:** When bit 6 of address 001A16 is "1" (clock synchronous mode). Divide this value by four when bit 6 of address 001A16 is "0" (UART mode).

## 3.1 Electrical characteristics

**Table 3.1.6 Switching characteristics (1)** ( $V_{CC} = 4.0$  to  $5.5$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85$  °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t <sub>WH</sub> (SCLK)	Serial I/O clock output "H" pulse width	Fig. 3.1.1	t <sub>c</sub> (SCLK)/2-30			ns
t <sub>WL</sub> (SCLK)	Serial I/O clock output "L" pulse width		t <sub>c</sub> (SCLK)/2-30			ns
t <sub>d</sub> (SCLK-TxD)	Serial I/O output delay time (Note 1)				140	ns
t <sub>v</sub> (SCLK-TxD)	Serial I/O output valid time (Note 1)		-30			ns
t <sub>r</sub> (SCLK)	Serial I/O clock output rising time				30	ns
t <sub>f</sub> (SCLK)	Serial I/O clock output falling time				30	ns
t <sub>r</sub> (CMOS)	CMOS output rising time (Note 2)			10	30	ns
t <sub>f</sub> (CMOS)	CMOS output falling time (Note 2)			10	30	ns

**Note1:** When the P45/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is "0".

**2:** XOUT pin is excluded.

**Table 3.1.7 Switching characteristics (2)** ( $V_{CC} = 3.0$  to  $4.0$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85$  °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t <sub>WH</sub> (SCLK)	Serial I/O clock output "H" pulse width	Fig. 3.1.1	t <sub>c</sub> (SCLK)/2-50			ns
t <sub>WL</sub> (SCLK)	Serial I/O clock output "L" pulse width		t <sub>c</sub> (SCLK)/2-50			ns
t <sub>d</sub> (SCLK-TxD)	Serial I/O output delay time (Note 1)				350	ns
t <sub>v</sub> (SCLK-TxD)	Serial I/O output valid time (Note 1)		-30			ns
t <sub>r</sub> (SCLK)	Serial I/O clock output rising time				50	ns
t <sub>f</sub> (SCLK)	Serial I/O clock output falling time				50	ns
t <sub>r</sub> (CMOS)	CMOS output rising time (Note 2)			20	50	ns
t <sub>f</sub> (CMOS)	CMOS output falling time (Note 2)			20	50	ns

**Note1:** When the P45/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is "0".

**2:** XOUT pin is excluded.

# APPENDIX

## 3.1 Electrical characteristics

**Table 3.1.8 Timing requirements in memory expansion mode and microprocessor mode (1)**

(VCC = 4.0 to 5.5 V, VSS = 0 V, Ta = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
tsu(ONW-φ)	Before φ ONW input set up time	-20			ns
th(φ-ONW)	After φ ONW input hold time	-20			ns
tsu(DB-φ)	Before φ data bus set up time	60			ns
th(φ-DB)	After φ data bus hold time	0			ns
tsu(ONW-RD)	Before RD ONW input set up time	-20			ns
tsu(ONW-WR)	Before WR ONW input set up time				
th(RD-ONW)	After RD ONW input hold time	-20			ns
th(WR-ONW)	After WR ONW input hold time				
tsu(DB-RD)	Before RD data bus set up time	65			ns
th(RD-DB)	After RD data bus hold time	0			ns

**Table 3.1.9 Switching characteristics in memory expansion mode and microprocessor mode (1)**

(VCC = 4.0 to 5.5 V, VSS = 0 V, Ta = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
tc(φ)	φ clock cycle time	Fig. 3.1.1		2tc(XIN)		ns
twH(φ)	φ clock "H" pulse width		tc(XIN)-10			ns
twL(φ)	φ clock "L" pulse width		tc(XIN)-10			ns
td(φ-AH)	After φ AD15-AD8 delay time			20	40	ns
tv(φ-AH)	After φ AD15-AD8 valid time		6	10		ns
td(φ-AL)	After φ AD7-AD0 delay time			25	45	ns
tv(φ-AL)	After φ AD7-AD0 valid time		6	10		ns
td(φ-SYNC)	SYNC delay time			20		ns
tv(φ-SYNC)	SYNC valid time			10		ns
td(φ-WR)	RD and WR delay time			10	20	ns
tv(φ-WR)	RD and WR valid time		3	5	10	ns
td(φ-DB)	After φ data bus delay time			20	70	ns
tv(φ-DB)	After φ data bus valid time		15			ns
twL(RD)	RD pulse width, WR pulse width		tc(XIN)-10			ns
twL(WR)	RD pulse width, WR pulse width (When one-wait is valid)		3tc(XIN)-10			ns
td(AH-RD)	After AD15-AD8 RD delay time		tc(XIN)-35	tc(XIN)-15		ns
td(AL-RD)	After AD7-AD0 RD delay time		tc(XIN)-40	tc(XIN)-20		ns
td(AH-WR)	After AD15-AD8 WR delay time					ns
td(AL-WR)	After AD7-AD0 WR delay time					ns
tv(RD-AH)	After RD AD15-AD8 valid time		0	5		ns
tv(WR-AH)	After WR AD15-AD8 valid time				ns	
tv(RD-AL)	After RD AD7-AD0 valid time	0	5		ns	
tv(WR-AL)	After WR AD7-AD0 valid time				ns	
td(WR-DB)	After WR data bus delay time		15	65	ns	
tv(WR-DB)	After WR data bus valid time	10			ns	
td(RESET-RESETOUT)	RESETOUT output delay time			200	ns	
tv(φ-RESET)	RESETOUT output valid time (Note)	0		200	ns	

**Note :** The RESETOUT goes "H" in sync with the fall of the φ clock that is anywhere between about 8 cycle and 13 cycles after the RESET input goes "H".

## 3.1 Electrical characteristics

Table 3.1.10 Timing requirements in memory expansion mode and microprocessor mode (2)

(VCC = 3.0 V, VSS = 0 V, Ta = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
tsu(ONW-φ)	Before φ ONW input set up time	-20			ns
th(φ-ONW)	After φ ONW input hold time	-20			ns
tsu(DB-φ)	Before φ data bus set up time	180			ns
th(φ-DB)	After φ data bus hold time	0			ns
tsu(ONW-RD)	Before RD ONW input set up time	-20			ns
tsu(ONW-WR)	Before WR ONW input set up time				
th(RD-ONW)	After RD ONW input hold time	-20			ns
th(WR-ONW)	After WR ONW input hold time				
tsu(DB-RD)	Before RD data bus set up time	185			ns
th(RD-DB)	After RD data bus hold time	0			ns

Table 3.1.11 Switching characteristics in memory expansion mode and microprocessor mode (2)

(VCC = 3.0 V, VSS = 0 V, Ta = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
tc(φ)	φ clock cycle time	Fig. 3.1.1		2tc(XIN)		ns
twH(φ)	φ clock "H" pulse width		tc(XIN)-20			ns
twL(φ)	φ clock "L" pulse width		tc(XIN)-20			ns
td(φ-AH)	After φ AD15-AD8 delay time				150	ns
tv(φ-AH)	After φ AD15-AD8 valid time		10	15		ns
td(φ-AL)	After φ AD7-AD0 delay time				150	ns
tv(φ-AL)	After φ AD7-AD0 valid time		10	15		ns
td(φ-SYNC)	SYNC delay time			40		ns
tv(φ-SYNC)	SYNC valid time			20		ns
td(φ-WR)	RD and WR delay time			15	25	ns
tv(φ-WR)	RD and WR valid time		3	7	15	ns
td(φ-DB)	After φ data bus delay time				200	ns
tv(φ-DB)	After φ data bus valid time		15			ns
twL(RD)	RD pulse width, WR pulse width		tc(XIN)-20			ns
twL(WR)	RD pulse width, WR pulse width (When one-wait is valid)		3tc(XIN)-20			ns
td(AH-RD)	After AD15-AD8 RD delay time		tc(XIN)-145			ns
td(AH-WR)	After AD15-AD8 WR delay time					
td(AL-RD)	After AD7-AD0 RD delay time		tc(XIN)-145			ns
td(AL-WR)	After AD7-AD0 WR delay time					
tv(RD-AH)	After RD AD15-AD8 valid time		5	10		ns
tv(WR-AH)	After WR AD15-AD8 valid time					
tv(RD-AL)	After RD AD7-AD0 valid time	5	10		ns	
tv(WR-AL)	After WR AD7-AD0 valid time					
td(WR-DB)	After WR data bus delay time			195	ns	
tv(WR-DB)	After WR data bus valid time	10			ns	
td(RESET-RESETOUT)	RESETOUT output delay time			300	ns	
tv(φ-RESET)	RESETOUT output valid time (Note)	0		300	ns	

**Note:** The RESETOUT goes "H" in sync with the fall of the φ clock that is anywhere between about 8 cycle and 13 cycles after the RESET input goes "H".

# APPENDIX

## 3.1 Electrical characteristics

### 3.1.5 Absolute maximum ratings (Extended operating temperature version)

Table 3.1.12 Absolute maximum ratings (Extended operating temperature version)

Symbol	Parameter	Conditions	Ratings	Unit
V <sub>CC</sub>	Power source voltage	All voltages are based on V <sub>SS</sub> . Output transistors are cut off.	-0.3 to 7.0	V
V <sub>I</sub>	Input voltage P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71		-0.3 to V <sub>CC</sub> +0.3	V
V <sub>I</sub>	Input voltage RESET, X <sub>IN</sub>		-0.3 to V <sub>CC</sub> +0.3	V
V <sub>I</sub>	Input voltage CNV <sub>SS</sub>		-0.3 to 13	V
V <sub>O</sub>	Output voltage P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71, X <sub>OUT</sub>		-0.3 to V <sub>CC</sub> +0.3	V
P <sub>d</sub>	Power dissipation	T <sub>a</sub> = 25 °C	1000(Note)	mW
T <sub>opr</sub>	Operating temperature		-40 to 85	°C
T <sub>stg</sub>	Storage temperature		-65 to 150	°C

Note : 300 mW in case of the flat package.

### 3.1.6 Recommended operating conditions (Extended operating temperature version)

Table 3.1.13 Recommended operating conditions (Extended operating temperature version)

(V<sub>CC</sub> = 4.0 to 5.5 V, T<sub>a</sub> = -40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V <sub>CC</sub>	Power source voltage	4.0	5.0	5.5	V
V <sub>SS</sub>	Power source voltage		0		V
V <sub>IH</sub>	"H" input voltage P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71	0.8 V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IH</sub>	"H" input voltage RESET, X <sub>IN</sub> , CNV <sub>SS</sub>	0.8 V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71	0		0.2 V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage RESET, CNV <sub>SS</sub>	0		0.2 V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage X <sub>IN</sub>	0		0.16 V <sub>CC</sub>	V
ΣI <sub>OH(peak)</sub>	"H" total peak output current P00–P07, P10–P17, P20–P27, P30–P37 (Note 1)			-80	mA
ΣI <sub>OH(peak)</sub>	"H" total peak output current P40–P47, P50–P57, P60–P67, P70, P71 (Note 1)			-80	mA
ΣI <sub>OL(peak)</sub>	"L" total peak output current P00–P07, P10–P17, P20–P27, P30–P37 (Note 1)			80	mA
ΣI <sub>OL(peak)</sub>	"L" total peak output current P40–P47, P50–P57, P60–P67, P70, P71 (Note 1)			80	mA
ΣI <sub>OH(avg)</sub>	"H" total average output current P00–P07, P10–P17, P20–P27, P30–P37 (Note 1)			-40	mA
ΣI <sub>OH(avg)</sub>	"H" total average output current P40–P47, P50–P57, P60–P67, P70, P71 (Note 1)			-40	mA
ΣI <sub>OL(avg)</sub>	"L" total average output current P00–P07, P10–P17, P20–P27, P30–P37 (Note 1)			40	mA
ΣI <sub>OL(avg)</sub>	"L" total average output current P40–P47, P50–P57, P60–P67, P70, P71 (Note 1)			40	mA
I <sub>OH(peak)</sub>	"H" peak output current P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71 (Note 2)			-10	mA
I <sub>OL(peak)</sub>	"L" peak output current P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71 (Note 2)			10	mA
I <sub>OH(avg)</sub>	"H" average output current P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71 (Note 3)			-5	mA
I <sub>OL(avg)</sub>	"L" average output current P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70, P71 (Note 3)			5	mA
f(X <sub>IN</sub> )	Internal clock oscillation frequency			8	MHz

Note 1: The total output current is the sum of all the currents flowing through all the applicable ports. The total average current is an average value measured over 100 ms. The total peak current is the peak value of all the currents.

2: The peak output current is the peak current flowing in each port.

3: The average output current I<sub>OL(avg)</sub>, I<sub>OH(avg)</sub> in an average value measured over 100 ms.

## 3.1 Electrical characteristics

### 3.1.7 Electrical characteristics (Extended operating temperature version)

**Table 3.1.14 Electrical characteristics (Extended operating temperature version)**

(VCC = 4.0 to 5.5 V, VSS = 0 V, Ta = -40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
VOH	"H" output voltage P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70, P71 (Note)	IOH = -10 mA	VCC-2.0			V
VOL	"L" output voltage P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70, P71	IOL = 10 mA			2.0	V
VT+ - VT-	Hysteresis CNTR0, CNTR1, INT0-INT5			0.4		V
VT+ - VT-	Hysteresis RXD, SCLK			0.5		V
VT+ - VT-	Hysteresis RESET			0.5		V
IiH	"H" input current P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70, P71	Vi = VCC			5.0	μA
IiH	"H" input current RESET, CNVSS	Vi = VCC			5.0	μA
IiH	"H" input current XIN	Vi = VCC		4		μA
IiL	"L" input current P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70, P71, RESET, CNVSS	Vi = VSS			-5.0	μA
IiL	"L" input current XIN	Vi = VSS		-4		μA
VRAM	RAM hold voltage	When clock stopped	2.0		5.5	V
ICC	Power source current	f(XIN) = 8 MHz		6.4	13	mA
		f(XIN) = 5 MHz		4	8	
		When WIT instruction is executed with f(XIN) = 8 MHz		1.5		
		When WIT instruction is executed with f(XIN) = 5 MHz		1		
		When STP instruction is executed with clock stopped, output transistors isolated.	Ta = 25 °C		0.1	1
	Ta = 85 °C			10		

**Note** : P45 is measured when the P45/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is "0".

# APPENDIX

## 3.1 Electrical characteristics

### 3.1.8 Timing requirements and Switching characteristics (Extended operating temperature version)

**Table 3.1.15 Timing requirements (Extended operating temperature version)**

(VCC = 4.0 to 5.5 V, VSS = 0 V, Ta = -40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
tW(RESET)	Reset input "L" pulse width	2			μs
tC(XIN)	External clock input cycle time	125			ns
tWH(XIN)	External clock input "H" pulse width	50			ns
tWL(XIN)	External clock input "L" pulse width	50			ns
tC(CNTR)	CNTR0, CNTR1 input cycle time	200			ns
tWH(CNTR)	CNTR0, CNTR1 input "H" pulse width	80			ns
tWL(CNTR)	CNTR0, CNTR1 input "L" pulse width	80			ns
tWH(INT)	INT0 to INT5 input "H" pulse width	80			ns
tWL(INT)	INT0 to INT5 input "L" pulse width	80			ns
tC(SCLK)	Serial I/O clock input cycle time (Note)	800			ns
tWH(SCLK)	Serial I/O clock input "H" pulse width (Note)	370			ns
tWL(SCLK)	Serial I/O clock input "L" pulse width (Note)	370			ns
tSU(RxD-SCLK)	Serial I/O input set up time	220			ns
tH(SCLK-RxD)	Serial I/O input hold time	100			ns

**Note:** Bit 6 of address 001A16 is "1". Divide this value by four bit 6 of address 001A16 is "0".

**Table 3.1.16 Switching characteristics (Extended operating temperature version)**

(VCC = 4.0 to 5.5 V, VSS = 0 V, Ta = -40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
tWH(SCLK)	Serial I/O clock output "H" pulse width	Fig. 3.1.1	tC(SCLK)/2-30			ns
tWL(SCLK)	Serial I/O clock output "L" pulse width		tC(SCLK)/2-30			ns
tD(SCLK-TxD)	Serial I/O output delay time (Note 1)				140	ns
tV(SCLK-TxD)	Serial I/O output valid time (Note 1)		-30			ns
tR(SCLK)	Serial I/O clock output rise time				30	ns
tF(SCLK)	Serial I/O clock output fall time				30	ns
tR(CMOS)	CMOS output rise time (Note 2)			10	30	ns
tF(CMOS)	CMOS output fall time (Note 2)			10	30	ns

**Note1:** When the P45/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is "0".

**2:** XOUT pin is excluded.



## 3.1 Electrical characteristics

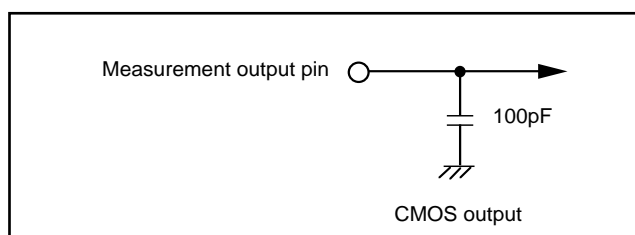
**Table 3.1.17 Timing requirements in memory expansion mode and microprocessor mode**  
(Extended operating temperature version) ( $V_{CC} = 4.0$  to  $5.5$  V,  $V_{SS} = 0$  V,  $T_a = -40$  to  $85$  °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{su}(\overline{ONW}-\phi)$	Before $\phi$ $\overline{ONW}$ input set up time	-20			ns
$t_h(\phi-\overline{ONW})$	After $\phi$ $\overline{ONW}$ input hold time	-20			ns
$t_{su}(\overline{DB}-\phi)$	Before $\phi$ data bus set up time	60			ns
$t_h(\phi-\overline{DB})$	After $\phi$ data bus hold time	0			ns
$t_{su}(\overline{ONW}-\overline{RD})$ $t_{su}(\overline{ONW}-\overline{WR})$	Before $\overline{RD}$ $\overline{ONW}$ input set up time Before $\overline{WR}$ $\overline{ONW}$ input set up time	-20			ns
$t_h(\overline{RD}-\overline{ONW})$ $t_h(\overline{WR}-\overline{ONW})$	After $\overline{RD}$ $\overline{ONW}$ input hold time After $\overline{WR}$ $\overline{ONW}$ input hold time	-20			ns
$t_{su}(\overline{DB}-\overline{RD})$	Before $\overline{RD}$ data bus set up time	65			ns
$t_h(\overline{RD}-\overline{DB})$	After $\overline{RD}$ data bus hold time	0			ns

**Table 3.1.18 Switching characteristics in memory expansion mode and microprocessor mode**  
(Extended operating temperature version) ( $V_{CC} = 4.0$  to  $5.5$  V,  $V_{SS} = 0$  V,  $T_a = -40$  to  $85$  °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_c(\phi)$	$\phi$ clock cycle time	Fig. 3.1.1		$2t_c(X_{IN})$		ns
$t_{wH}(\phi)$	$\phi$ clock "H" pulse width		$t_c(X_{IN})-10$			ns
$t_{wL}(\phi)$	$\phi$ clock "L" pulse width		$t_c(X_{IN})-10$			ns
$t_d(\phi-AH)$	After $\phi$ AD15-AD8 delay time			20	40	ns
$t_v(\phi-AH)$	After $\phi$ AD15-AD8 valid time		6	10		ns
$t_d(\phi-AL)$	After $\phi$ AD7-AD0 delay time			25	45	ns
$t_v(\phi-AL)$	After $\phi$ AD7-AD0 valid time		6	10		ns
$t_d(\phi-SYNC)$	SYNC delay time			20		ns
$t_v(\phi-SYNC)$	SYNC valid time			10		ns
$t_d(\phi-\overline{WR})$	$\overline{RD}$ and $\overline{WR}$ delay time			10	20	ns
$t_v(\phi-\overline{WR})$	$\overline{RD}$ and $\overline{WR}$ valid time		3	5	10	ns
$t_d(\phi-DB)$	After $\phi$ data bus delay time			20	70	ns
$t_v(\phi-DB)$	After $\phi$ data bus valid time		15			ns
$t_{wL}(\overline{RD})$ $t_{wL}(\overline{WR})$	$\overline{RD}$ pulse width, $\overline{WR}$ pulse width $\overline{RD}$ pulse width, $\overline{WR}$ pulse width (When one-wait is valid)		$t_c(X_{IN})-10$ $3t_c(X_{IN})-10$			ns
$t_d(AH-\overline{RD})$ $t_d(AH-\overline{WR})$	After AD15-AD8 $\overline{RD}$ delay time After AD15-AD8 $\overline{WR}$ delay time		$t_c(X_{IN})-35$	$t_c(X_{IN})-15$		ns
$t_d(AL-\overline{RD})$ $t_d(AL-\overline{WR})$	After AD7-AD0 $\overline{RD}$ delay time After AD7-AD0 $\overline{WR}$ delay time		$t_c(X_{IN})-40$	$t_c(X_{IN})-20$		ns
$t_v(\overline{RD}-AH)$ $t_v(\overline{WR}-AH)$	After $\overline{RD}$ AD15-AD8 valid time After $\overline{WR}$ AD15-AD8 valid time		0	5		ns
$t_v(\overline{RD}-AL)$ $t_v(\overline{WR}-AL)$	After $\overline{RD}$ AD7-AD0 valid time After $\overline{WR}$ AD7-AD0 valid time		0	5		ns
$t_d(\overline{WR}-DB)$	After $\overline{WR}$ data bus delay time			15	65	ns
$t_v(\overline{WR}-DB)$	After $\overline{WR}$ data bus valid time		10			ns
$t_d(\overline{RESET}-\overline{RESETOUT})$	$\overline{RESETOUT}$ output delay time			200	ns	
$t_v(\phi-\overline{RESET})$	$\overline{RESETOUT}$ output valid time (Note)	0		200	ns	

**Note :** The  $\overline{RESETOUT}$  output goes "H" in sync with the fall of the  $\phi$  clock that is anywhere between about 8 cycle and 13 cycles after the  $\overline{RESET}$  input goes "H".



**Fig. 3.1.1 Circuit for measuring output switching characteristics**

# APPENDIX

## 3.1 Electrical characteristics

### 3.1.9 Timing diagram

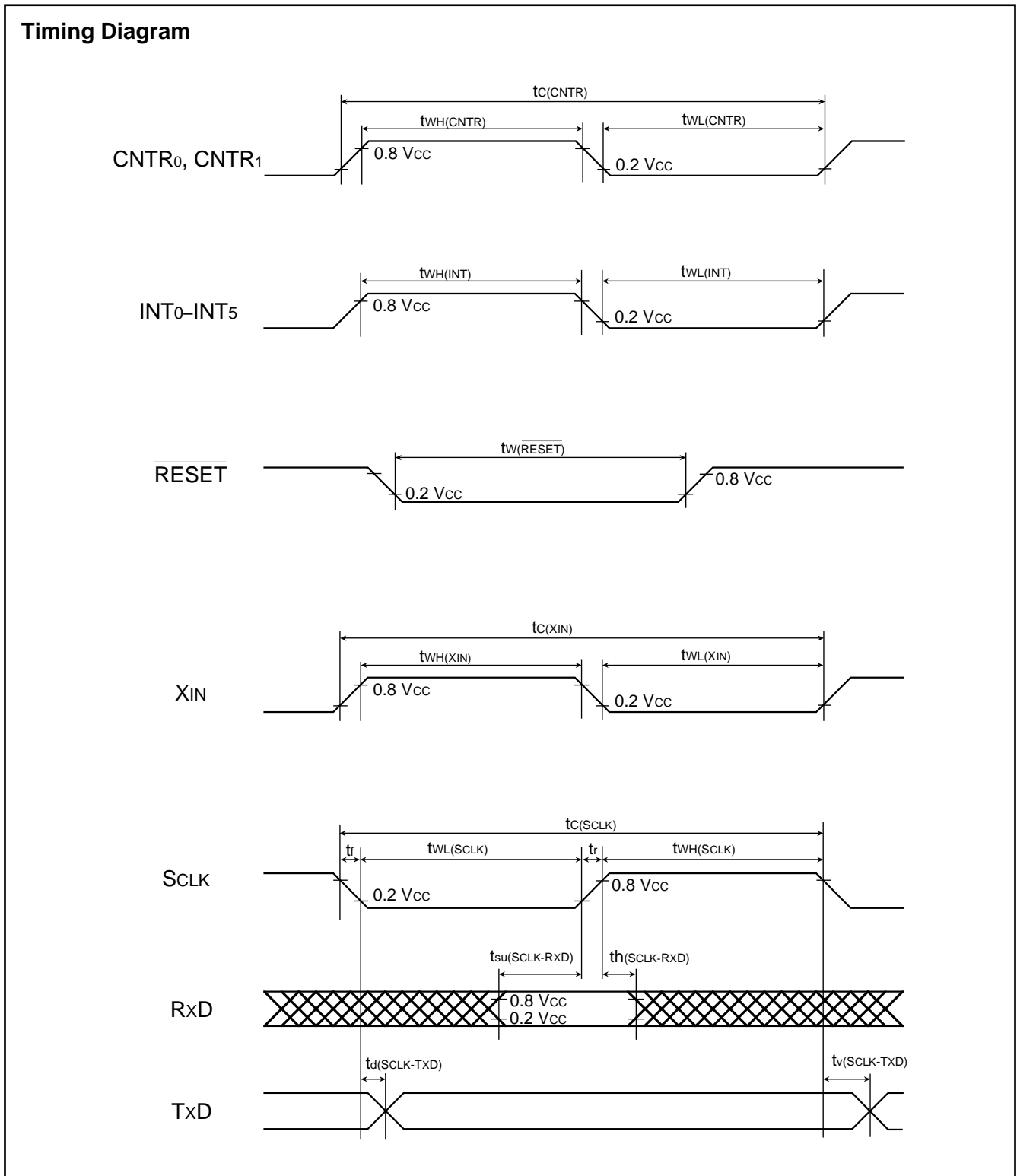


Fig. 3.1.2 Timing diagram (in single-chip mode)

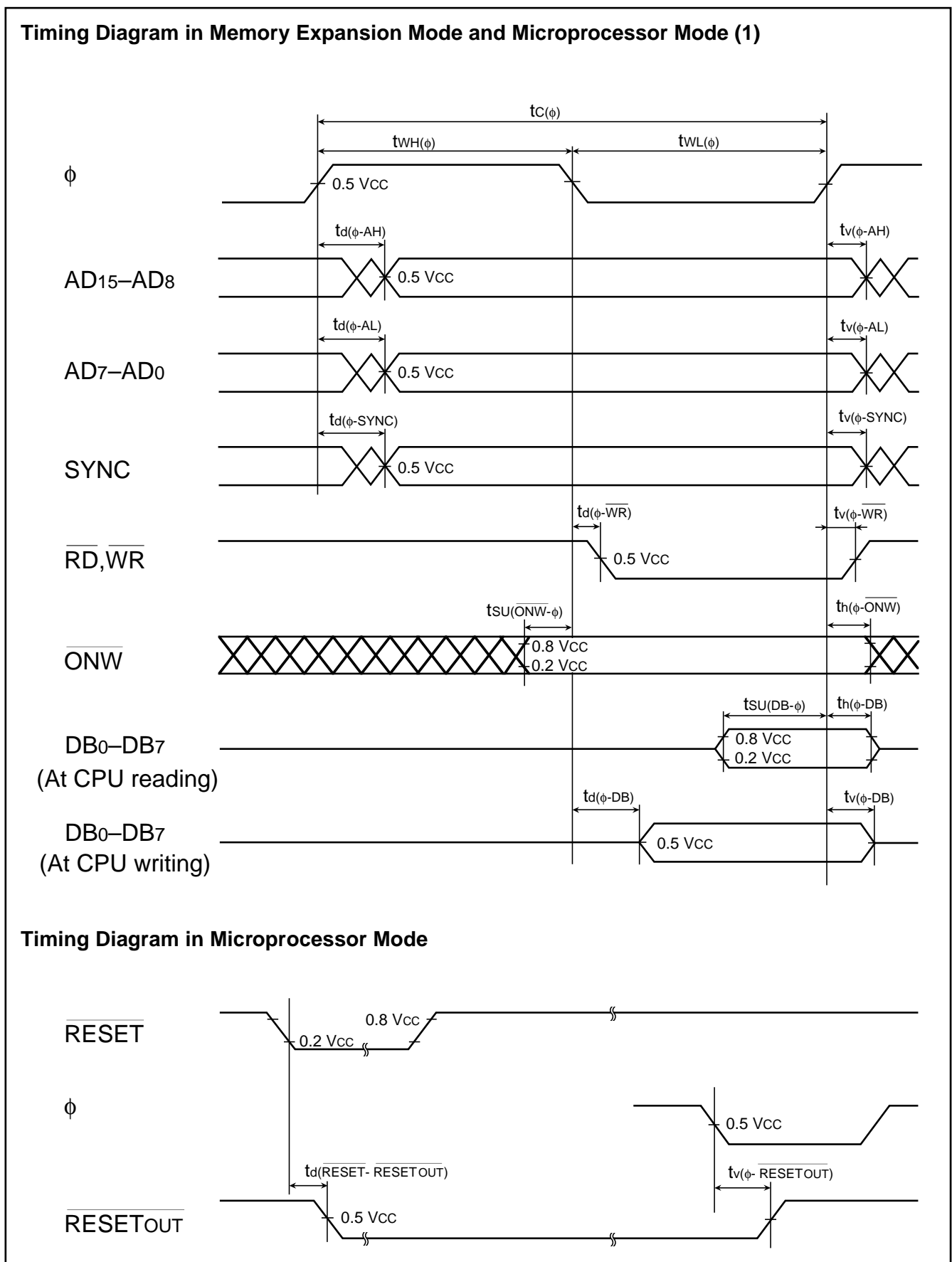


Fig. 3.1.3 Timing diagram (in memory expansion mode and microprocessor mode) (1)

# APPENDIX

## 3.1 Electrical characteristics

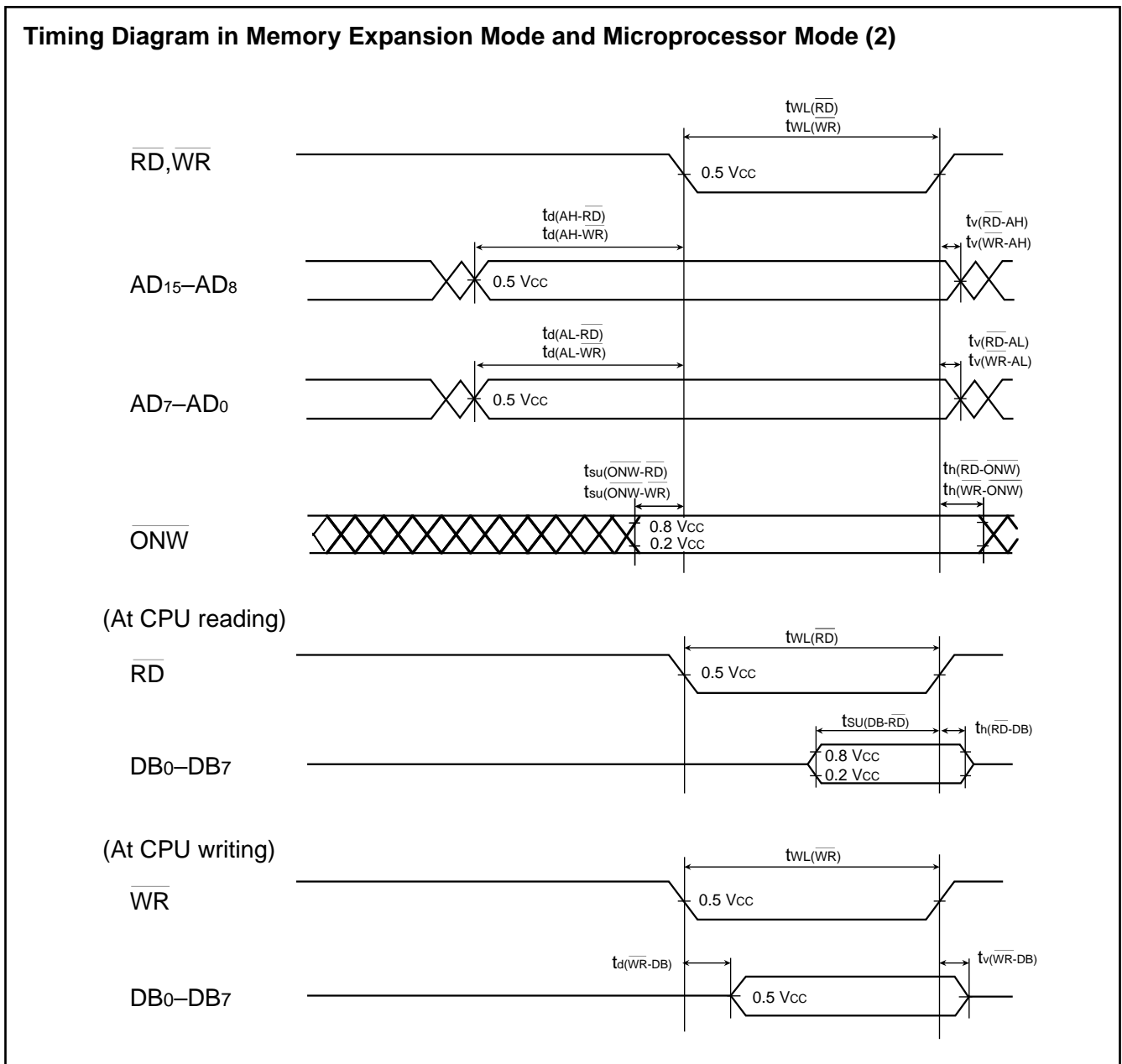


Fig. 3.1.4 Timing diagram (in memory expansion mode and microprocessor mode) (2)

## 3.2 Standard characteristics

### 3.2.1 Power source current characteristic examples

Figures 3.2.1 and Figure 3.2.2 show power source current characteristic examples.

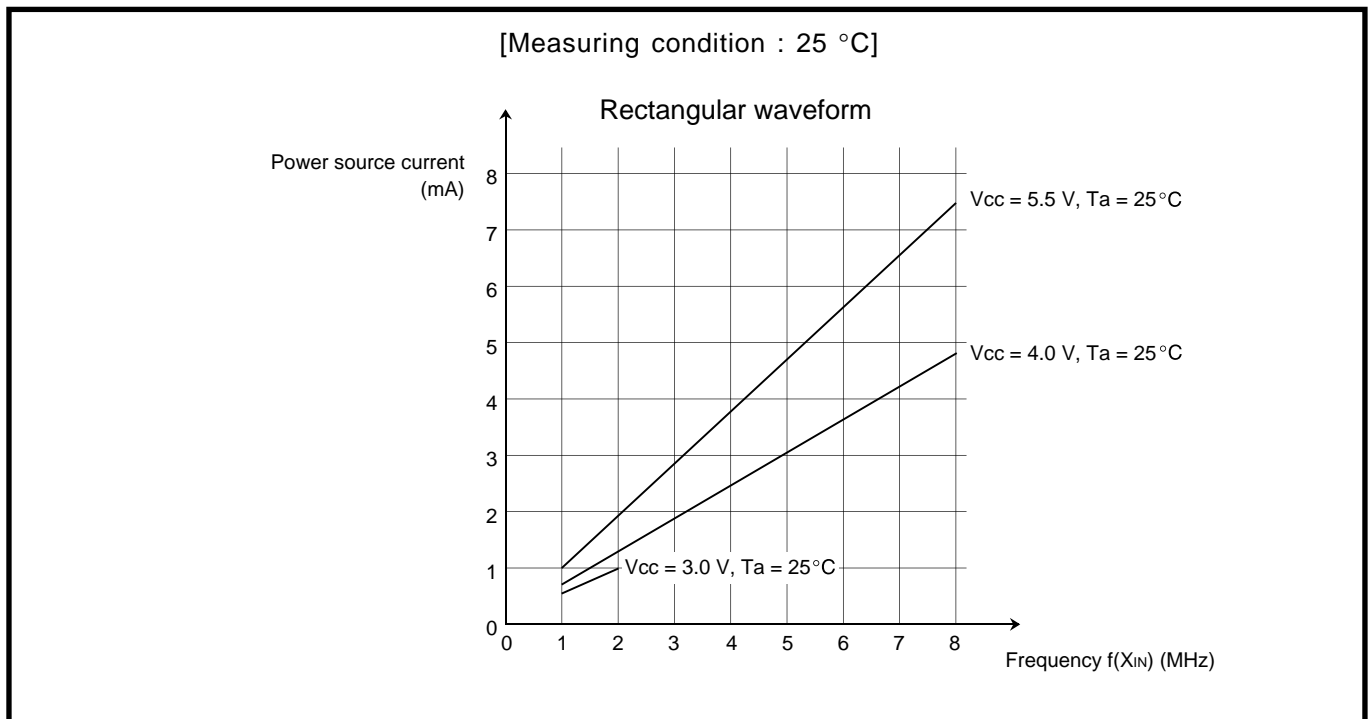


Fig. 3.2.1 Power source current characteristic example

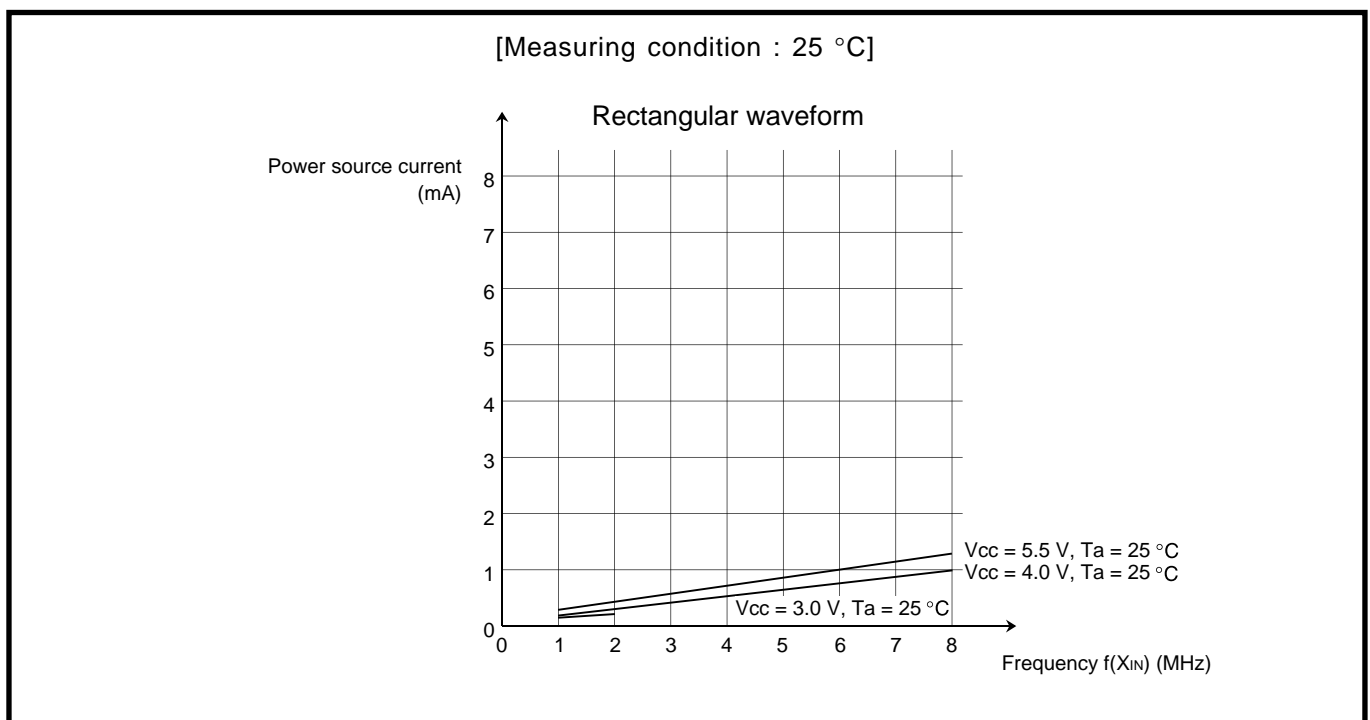


Fig. 3.2.2 Power source current characteristic example (in wait mode)

# APPENDIX

## 3.2 Standard characteristics

### 3.2.2 Port standard characteristic examples

Figures 3.2.3, Figure 3.2.4, Figure 3.2.5 and Figure 3.2.6 show port standard characteristic examples.

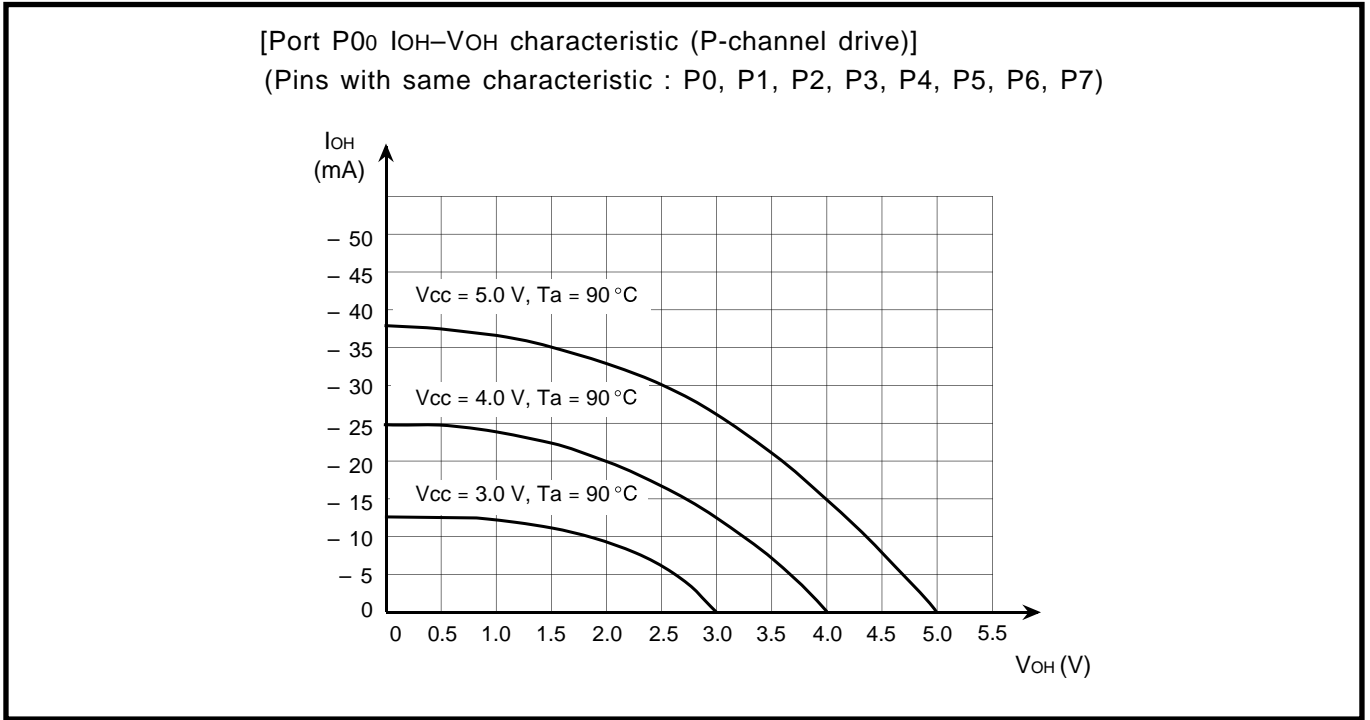


Fig. 3.2.3 Standard characteristic example of CMOS output port at P-channel drive (1)

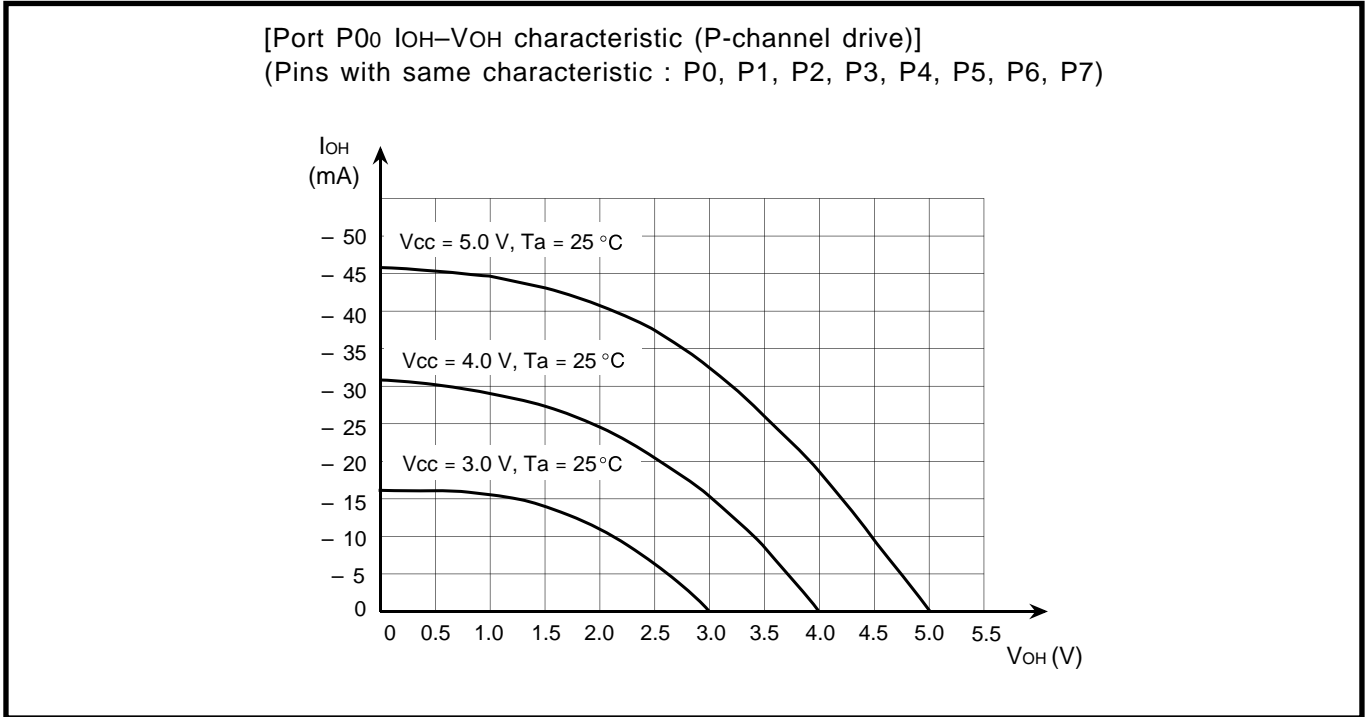


Fig. 3.2.4 Standard characteristic example of CMOS output port at P-channel drive (2)

3.2 Standard characteristics

[Port P00 IOL-VOL characteristic (N-channel drive)]

(Pins with same characteristic : P0, P1, P2, P3, P4, P5, P6, P7)

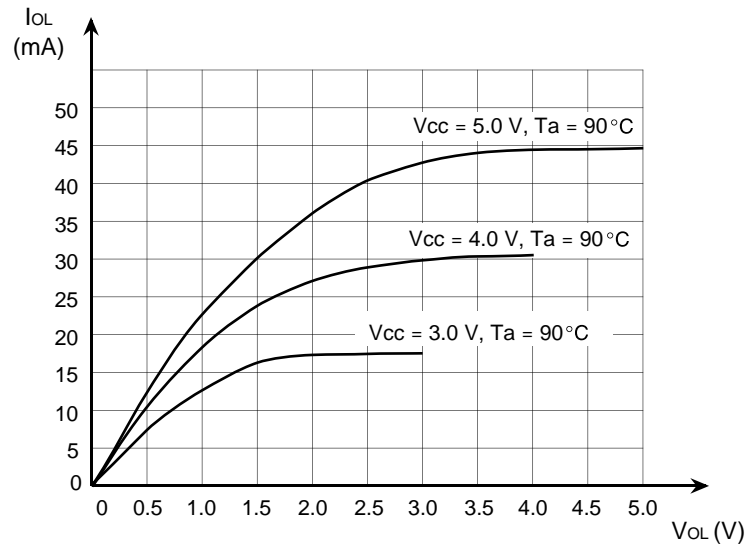


Fig. 3.2.5 Standard characteristic example of CMOS output port at N-channel drive (1)

[Port P00 IOL-VOL characteristic (N-channel drive)]

(Pins with same characteristic : P0, P1, P2, P3, P4, P5, P6, P7)

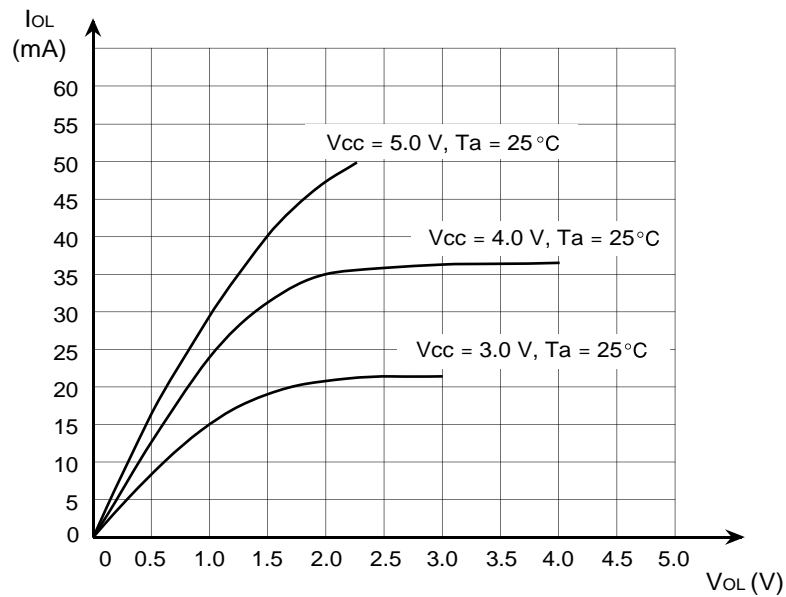


Fig. 3.2.6 Standard characteristic example of CMOS output port at N-channel drive (2)

# APPENDIX

## 3.3 Notes on use

### 3.3 Notes on use

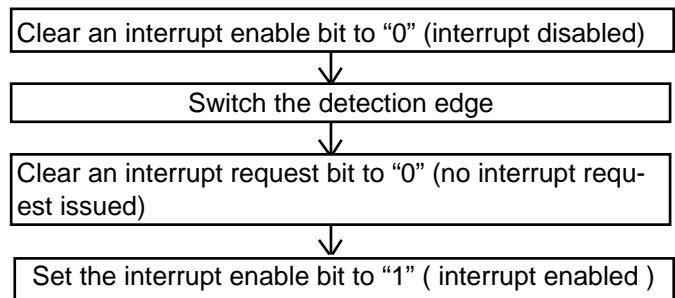
#### 3.3.1 Notes on interrupts

##### (1) Sequence for switching an external interrupt detection edge

When the external interrupt detection edge must be switched, make sure the following sequence.

##### Reason

The interrupt circuit recognizes the switching of the detection edge as the change of external input signals. This may cause an unnecessary interrupt.



##### (2) Bits 7 and 6 of the interrupt control register 2

Fix the bits 7 and 6 of the interrupt control register 2 (Address:003F<sub>16</sub>) to "0".

Figure 3.3.1 shows the structure of the interrupt control register 2.

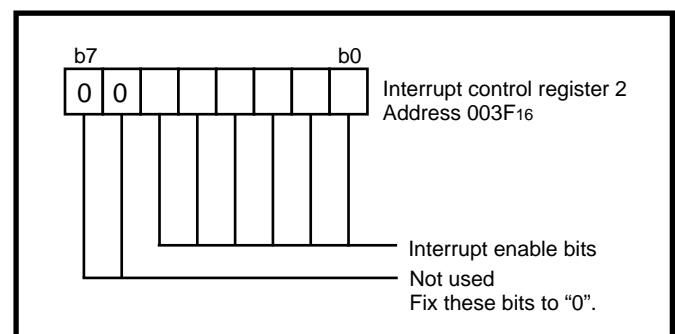


Fig. 3.3.1 Structure of interrupt control register 2

#### 3.3.2 Notes on the serial I/O

##### (1) Stop of data transmission

As for the serial I/O that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the transmit enable bit to "0" (transmit disabled), and clear the serial I/O enable bit to "0" (serial I/O disabled) in the following cases :

- when stopping data transmission during transmitting data in the clock synchronous serial I/O mode
- when stopping data transmission during transmitting data in the UART mode
- when stopping only data transmission during transmitting and receiving data in the UART mode

##### Reason

Since transmission is not stopped and the transmission circuit is not initialized even if the serial I/O enable bit is cleared to "0" (serial I/O disabled), the internal transmission is running (in this case, since pins TxD, RxD, SCLK, and  $\overline{\text{SRDY}}$  function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, the data is transferred to the transmit shift register and start to be shifted. When the serial I/O enable bit is set to "1" at this time, the data during internally shifting is output to the TxD pin and it may cause an operation failure to a microcomputer.

##### (2) Stop of data reception

As for the serial I/O that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear the receive enable bit to "0" (receive disabled), or clear the serial I/O enable bit to "0" (serial I/O disabled) in the following case :

- when stopping data reception during receiving data in the clock synchronous serial I/O mode

Clear the receive enable bit to "0" (receive disabled) in the following cases :

- when stopping data reception during receiving data in the UART mode
- when stopping only data reception during transmitting and receiving data in the UART mode



### (3) Stop of data transmission and reception in a clock synchronous serial I/O mode

As for the serial I/O that can be used as either a clock synchronous or an asynchronous (UART) serial I/O, clear both the transmit enable bit and receive enable bit to "0" (transmit and receive disabled) at the same time in the following case:

- when stopping data transmission and reception during transmitting and receiving data in the clock synchronous mode (when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

#### Reason

In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.

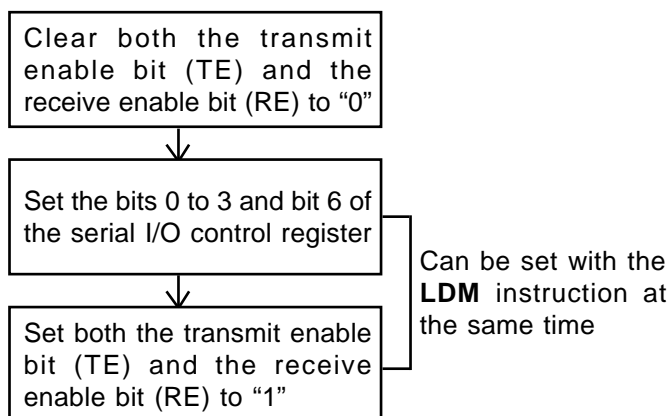
In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to "0" (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/O enable bit to "0" (serial I/O disabled) (refer to (1)).

### (4) The $\overline{\text{SRDY}}$ pin on a receiving side

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

### (5) Stop of data reception in a clock synchronous serial I/O mode

Set the serial I/O 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."



### (6) Control of data transmission using the transmit shift completion flag

The transmit shift completion flag changes from "1" to "0" with a delay of 0.5 to 1.5 shift clocks. When checking the transmit shift completion flag after writing a data to the transmit buffer register for controlling a data transmission, note this delay.

### (7) Control of data transmission using an external clock

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

### 3.3.3 Notes on the $\overline{\text{RESET}}$ pin

When a rising time of the reset signal is long, connect a ceramic capacitor or others across the  $\overline{\text{RESET}}$  pin and the Vss pin. And use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, make sure the following :

- Make the length of the wiring which is connected to a capacitor the shortest possible.
- Make sure to check 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, a microcomputer may malfunction.

# APPENDIX

## 3.3 Notes on use

---

### 3.3.4 Notes on input and output pins

#### (1) Fix of a port input level in stand-by state

Fix input levels of an input and an I/O port for getting effect of low-power dissipation in stand-by state\*, especially for the I/O ports of the N-channel open-drain.

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

When determining a resistance value, make sure the following:

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

\* stand-by state : the stop mode by executing the **STP** instruction  
the wait mode by executing the **WIT** instruction

#### Reason

Even when setting as an output port with its direction register, in the following state :

- N-channel.....when the content of the port latch is "1"

the transistor becomes the OFF state, which causes the ports to be the high-impedance state. Make sure that the level becomes "undefined" depending on external circuits.

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

#### (2) Modify of the content of I/O port latch

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

#### Reason

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

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

Make sure 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.
- Even when a bit of a port latch which is set as an input port is not specified with a bit managing instruction, its value may be changed in case where content of the pin differs from a content of the port latch.

\* bit managing instructions : **SEB** and **CLB** instruction

### 3.3.5 Notes on memory expansion mode and microprocessor mode

#### (1) Writing data to the port latch of port P3

In the memory expansion or the microprocessor mode, ports P30 and P31 can be used as the output port. Use the **LDM** or **STA** instruction for writing data to the port latch (address 0006<sub>16</sub>) of port P3.

When using a read-modify-write instruction (the **SEB** or the **CLB** instruction), allocate the read and the write enabled memory at address 0006<sub>16</sub>.

#### Reason

In the memory expansion or microprocessor mode, address 0006<sub>16</sub> is allocated in the external area.

Accordingly,

- Data is read from the external memory.
- Data is written to both the port latch of the port P3 and the external memory.

Accordingly, when executing a read-modify-write instruction for address 0006<sub>16</sub>, external memory data is read and modified, and the result is written in both the port latch of the port P3 and the external memory. If the read enabled memory is not allocated at address 0006<sub>16</sub>, the read data is undefined. The undefined data is modified and written to the port latch of the port P3. The port latch data of port P3 becomes "undefined."

#### (2) Overlap of an internal memory and an external memory

When the internal and the external memory are overlapped in the memory expansion mode, the internal memory is valid in this overlapped area. When the CPU writes or reads to this area, the following is performed :

- When reading data  
Only the data in the internal memory is read into the CPU and the data in the external memory is not read into the CPU. However, as the read signal and address are still valid, the external memory data of the corresponding address is output to the external data bus.
- When writing data  
Data is written in both the internal and the external memory.

# APPENDIX

## 3.3 Notes on use

### 3.3.6 Notes on built-in PROM

#### (1) Programming adapter

To write or read data into/from the internal PROM, use the dedicated programming adapter and general-purpose PROM programmer as shown in Table 3.3.1.

**Table 3.3.1 Programming adapter**

Microcomputer	Programming adapter
M38002E4SS	PCA4738S-64A
M38004E8SS	
M38002E2SP	
M38002E4SP	
M38004E8SP	
(one-time blank)	
M38002E4DSP	
(one-time blank)	
M38002E4FS	PCA4738L-64A
M38004E8FS	
M38002E2FP	PCA4738F-64A
M38002E4FP	
M38004E8FP	
(one-time blank)	
M38002E4DFP	
(one-time blank)	

#### (2) Write and read

In PROM mode, operation is the same as that of the M5M27C256AK, but programming conditions of PROM programmer are not set automatically because there are no internal device ID codes.

Accurately set the following conditions for data write/read. Take care not to apply 21 V to Vpp pin (is also used as the CNVss pin), or the product may be permanently damaged.

- Programming voltage : 12.5 V
- Setting of programming adapter switch : refer to table 3.3.2
- Setting of PROM programmer address : refer to table 3.3.3

**Table 3.3.2 Setting of programming adapter switch**

Programming adapter	SW 1	SW 2	SW 3
PCA4738S-64A	CMOS	CMOS	OFF
PCA4738L-64A			
PCA4738F-64A			

**Table 3.3.3 Setting of PROM programmer address**

Microcomputer	PROM programmer start address	PROM programmer completion address
M38002E2SP	Address : 6080 <sub>16</sub> (Note 1)	Address : 7FFD <sub>16</sub> (Note 1)
M38002E2FP		
M38002E4SS	Address : 4080 <sub>16</sub> (Note 2)	Address : 7FFD <sub>16</sub> (Note 2)
M38002E4SP		
M38002E4FS		
M38002E4FP		
M38002E4DSP		
M38002E4DFP		
M38004E8SS	Address : 0080 <sub>16</sub> (Note 3)	Address : 7FFD <sub>16</sub> (Note 3)
M38004E8SP		
M38004E8FS		
M38004E8FP		

Note1 : Addresses E080<sub>16</sub> to FFFD<sub>16</sub> in the internal PROM correspond to addresses 6080<sub>16</sub> to 7FFD<sub>16</sub> in the ROM programmer.

2 : Addresses C080<sub>16</sub> to FFFD<sub>16</sub> in the internal PROM correspond to addresses 4080<sub>16</sub> to 7FFD<sub>16</sub> in the ROM programmer.

3 : Addresses 8080<sub>16</sub> to FFFD<sub>16</sub> in the internal PROM correspond to addresses 0080<sub>16</sub> to 7FFD<sub>16</sub> in the ROM programmer.

### (3) Erasing

Contents of the windowed EPROM are erased through an ultraviolet light source of the wavelength 2537-Ångstrom . At least 15 W-sec/cm<sup>2</sup> are required to erase EPROM contents.

# APPENDIX

## 3.4 Countermeasures against noise

### 3.4 Countermeasures against noise

Countermeasures against noise are described below. The following countermeasures are effective against noise in theory, however, it is necessary not only to take measures as follows but to evaluate before actual use.

#### 3.4.1 Shortest wiring length

The wiring on a printed circuit board can be as an antenna which feeds noise into the microcomputer.

The shorter the total wiring length (by mm unit), the less the possibility of noise insertion into a microcomputer.

#### (1) Wiring for the $\overline{\text{RESET}}$ pin

Make the length of wiring which is connected to the  $\overline{\text{RESET}}$  pin as short as possible. Especially, connect a capacitor across the  $\overline{\text{RESET}}$  pin and the Vss pin with the shortest possible wiring (within 20mm).

#### Reason

The reset works to initialize a microcomputer.

The width of a pulse input into the  $\overline{\text{RESET}}$  pin is determined by the timing necessary conditions. If noise having a shorter pulse width than the standard is input to the  $\overline{\text{RESET}}$  pin, the reset is released before the internal state of the microcomputer is completely initialized. This may cause a program runaway.

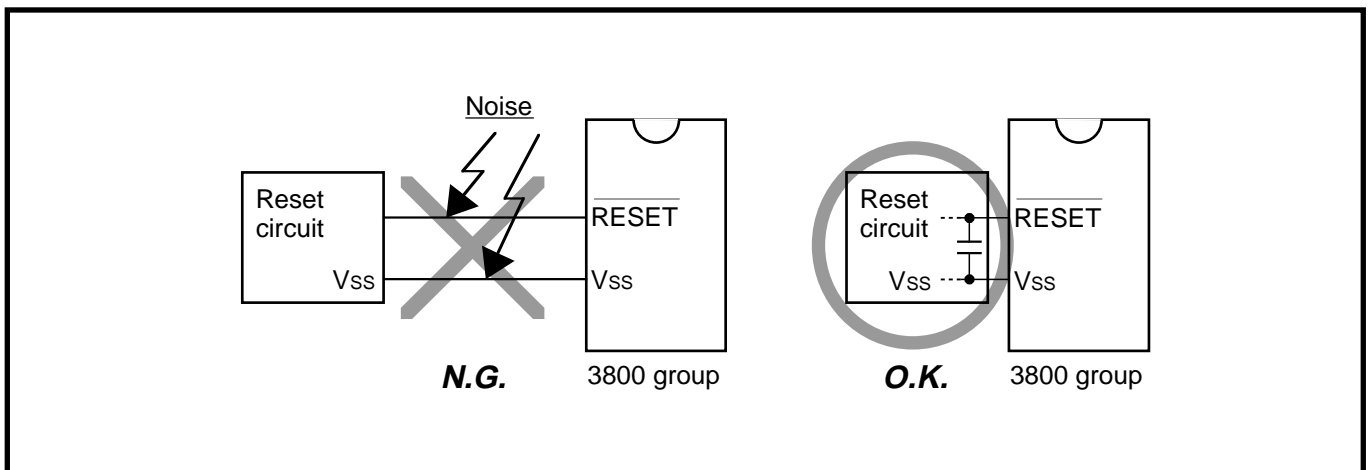


Fig. 3.4.1 Wiring for the  $\overline{\text{RESET}}$  pin

#### (2) Wiring for clock input/output pins

- Make the length of wiring which is connected to clock I/O pins as short as possible.
- Make the length of wiring (within 20mm) across the grounding lead of a capacitor which is connected to an oscillator and the Vss pin of a microcomputer as short as possible.
- Separate the Vss pattern only for oscillation from other Vss patterns.

#### Reason

A microcomputer's operation synchronizes with a clock generated by the oscillator (circuit). If noise enters clock I/O pins, clock waveforms may be deformed. This may cause a malfunction or program runaway.

Also, if a potential difference is caused by the noise between the Vss level of a microcomputer and the Vss level of an oscillator, the correct clock will not be input in the microcomputer.

## 3.4 Countermeasures against noise

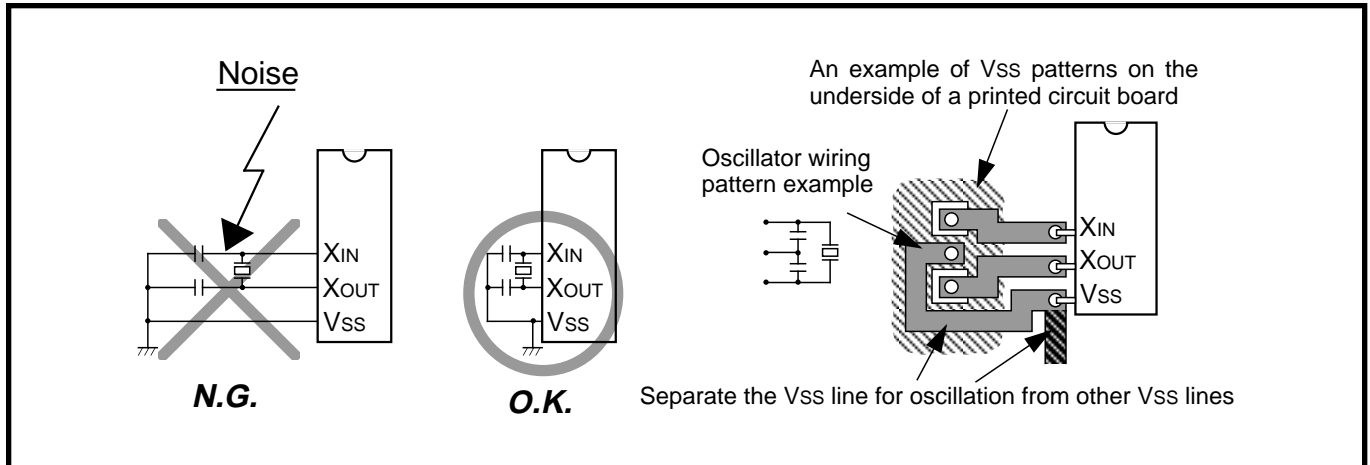


Fig. 3.4.2 Wiring for clock I/O pins

### (3) Wiring for the VPP pin of the One Time PROM version and the EPROM version (In this microcomputer the VPP pin is also used as the CNVss pin)

Connect an approximately 5 kΩ resistor to the VPP pin the shortest possible in series and also to the VSS pin. When not connecting the resistor, make the length of wiring between the VPP pin and the VSS pin the shortest possible.

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

#### Reason

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

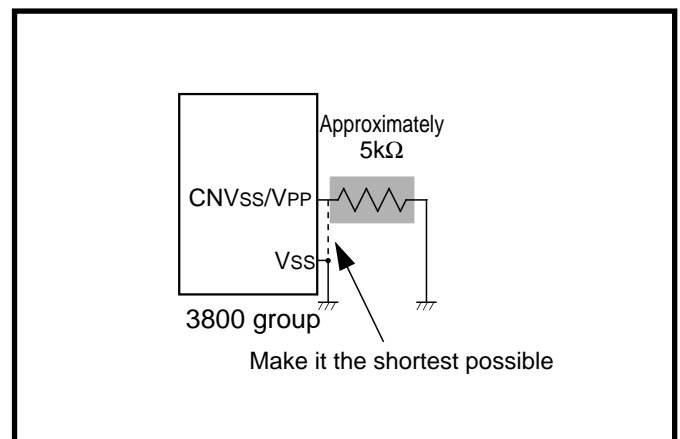


Fig. 3.4.3 Wiring for the VPP pin of the One Time PROM and the EPROM version

### 3.4.2 Connection of a bypass capacitor across the Vss line and the Vcc line

Connect an approximately 0.1 μF bypass capacitor across the VSS line and the VCC line as follows:

- Connect a bypass capacitor across the VSS pin and the VCC pin at equal length.
- Connect a bypass capacitor across the VSS pin and the VCC pin with the shortest possible wiring.
- Use lines with a larger diameter than other signal lines for VSS line and VCC line.

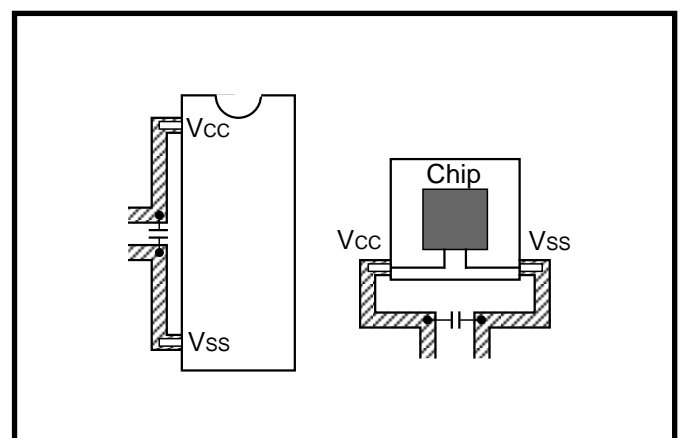


Fig. 3.4.4 Bypass capacitor across the Vss line and the Vcc line

# APPENDIX

## 3.4 Countermeasures against noise

### 3.4.3. Consideration for oscillator

Take care to prevent an oscillator that generates clocks for a microcomputer operation from being affected by other signals.

#### (1) Keeping an oscillator away from large current signal lines

Install a microcomputer (and especially an oscillator) as far as possible from signal lines where a current larger than the tolerance of current value flows.

#### Reason

In the system using a microcomputer, there are signal lines for controlling motors, LEDs, and thermal heads or others. When a large current flows through those signal lines, strong noise occurs because of mutual inductance.

#### (2) Keeping an oscillator away from signal lines where potential levels change frequently

Install an oscillator and a connecting pattern of an oscillator away from signal lines where potential levels change frequently. Also, do not cross such signal lines over the clock lines or the signal lines which are sensitive to noise.

#### Reason

Signal lines where potential levels change frequently (such as the CNTR pin line) may affect other lines at signal rising or falling edge. If such lines cross over a clock line, clock waveforms may be deformed, which causes a microcomputer failure or a program runaway.

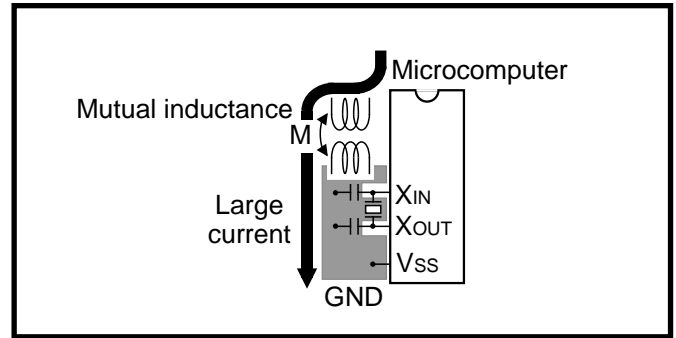


Fig.3.4.5 Wiring for a large current signal line

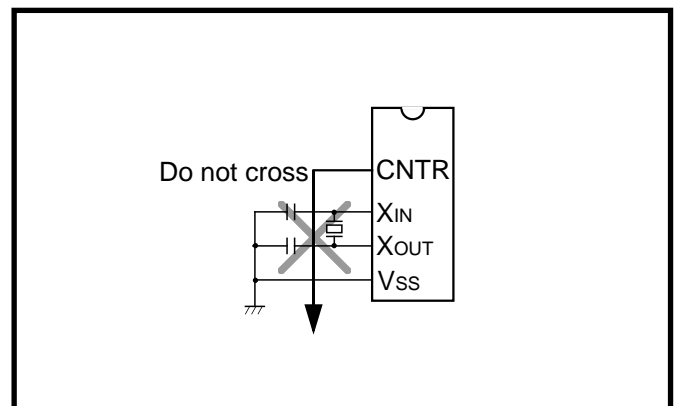


Fig.3.4.6 Wiring to a signal line where potential levels change frequently

### 3.4.4 Setup for I/O ports

Setup I/O ports using hardware and software as follows:

#### <Hardware>

- Connect a resistor of 100 Ω or more to an I/O port in series.

#### <Software>

- As for an input port, read data several times by a program for checking whether input levels are equal or not.
- As for an output port, since the output data may reverse because of noise, rewrite data to its port latch at fixed periods.
- Rewrite data to direction registers and pull-up control registers (only the product having it) at fixed periods.

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

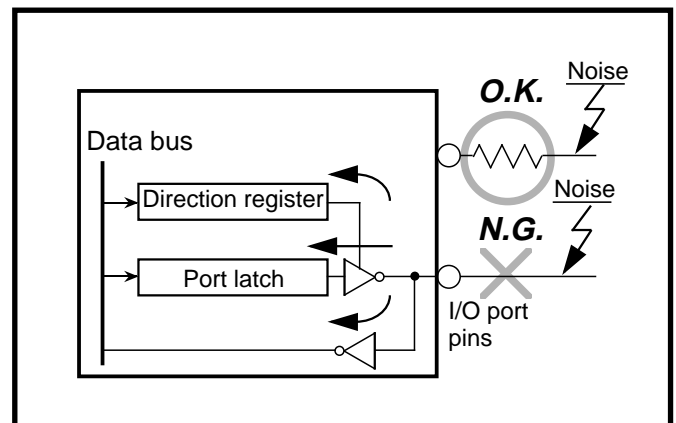


Fig. 3.4.7 Setup for I/O ports



### 3.4.5 Providing of watchdog timer function by software

If a microcomputer runs away because of noise or others, it can be detected by a software watchdog timer and the microcomputer can be reset to normal operation. This is equal to or more effective than program runaway detection by a hardware watchdog timer. The following shows an example of a watchdog timer provided by software.

In the following example, to reset a microcomputer to normal operation, the main routine detects errors of the interrupt processing routine and the interrupt processing routine detects errors of the main routine. This example assumes that interrupt processing is repeated multiple times in a single main routine processing.

<The main routine>

- Assigns a single byte of RAM to a software watchdog timer (SWDT) and writes the initial value N in the SWDT once at each execution of the main routine. The initial value N should satisfy the following condition:

$$N+1 \geq (\text{Counts of interrupt processing executed in each main routine})$$

As the main routine execution cycle may change because of an interrupt processing or others, the initial value N should have a margin.

- Watches the operation of the interrupt processing routine by comparing the SWDT contents with counts of interrupt processing count after the initial value N has been set.
- Detects that the interrupt processing routine has failed and determines to branch to the program initialization routine for recovery processing in the following cases:  
If the SWDT contents do not change after interrupt processing

<The interrupt processing routine>

- Decrements the SWDT contents by 1 at each interrupt processing.
- Determines that the main routine operates normally when the SWDT contents are reset to the initial value N at almost fixed cycles (at the fixed interrupt processing count).
- Detects that the main routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:  
When the contents of the SWDT reach 0 or less by continuative decrement without initializing to the initial value N.

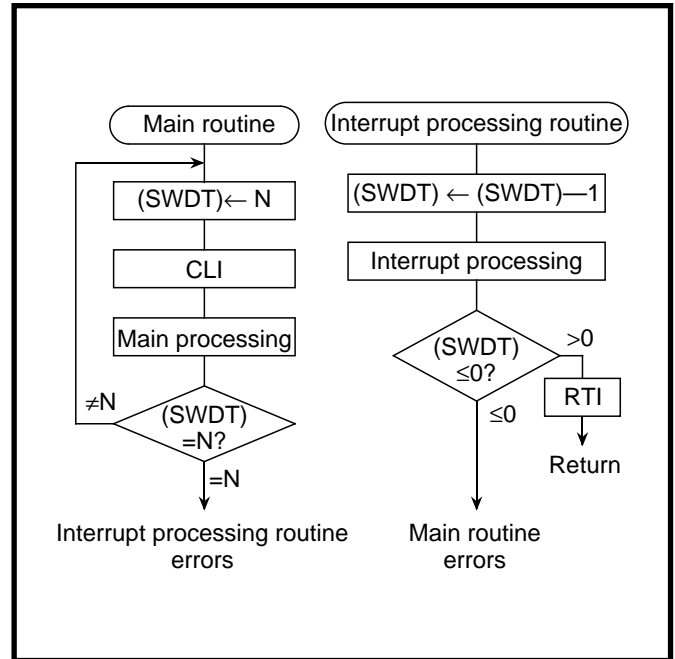


Fig. 3.4.8 Watchdog timer by software

# APPENDIX

## 3.5 List of registers

### 3.5 List of registers

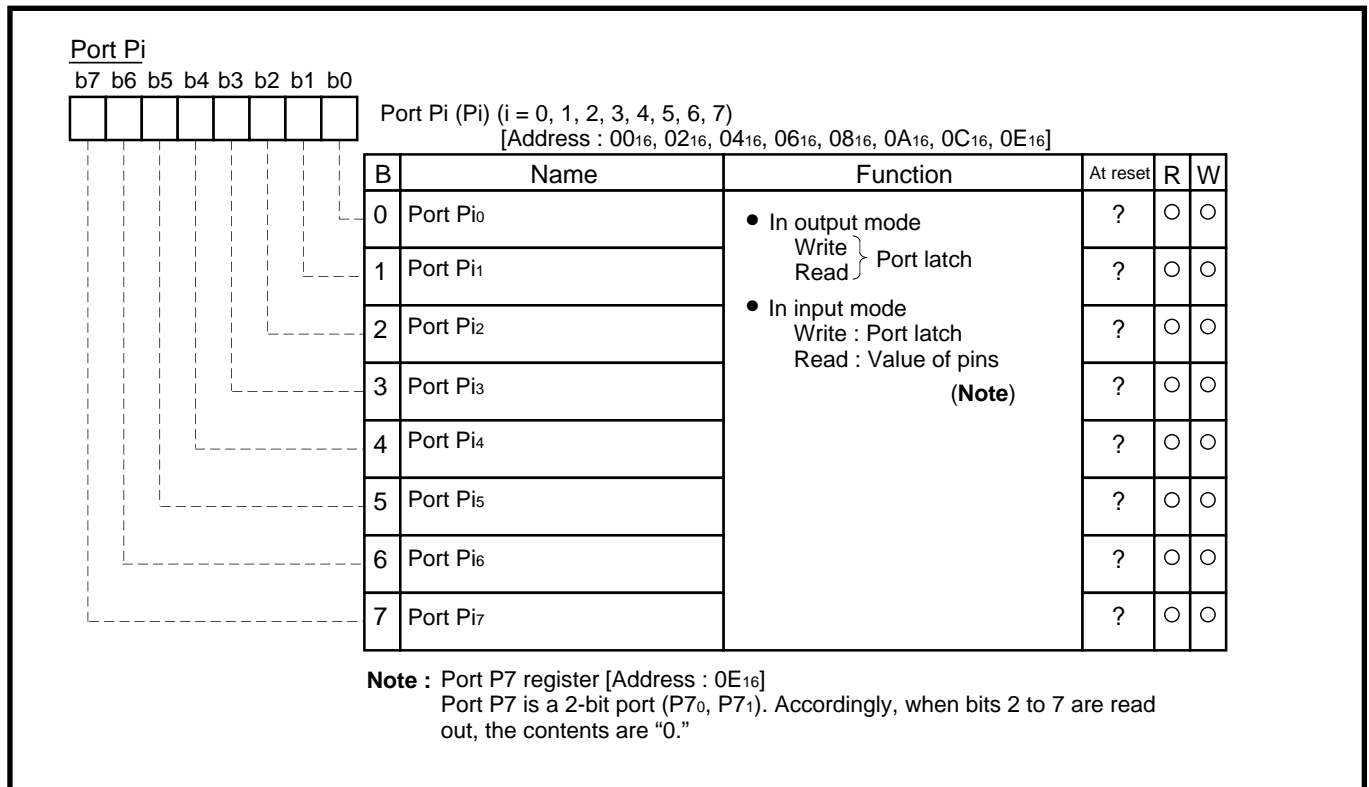


Fig. 3.5.1 Structure of Port Pi (i = 0, 1, 2, 3, 4, 5, 6, 7)

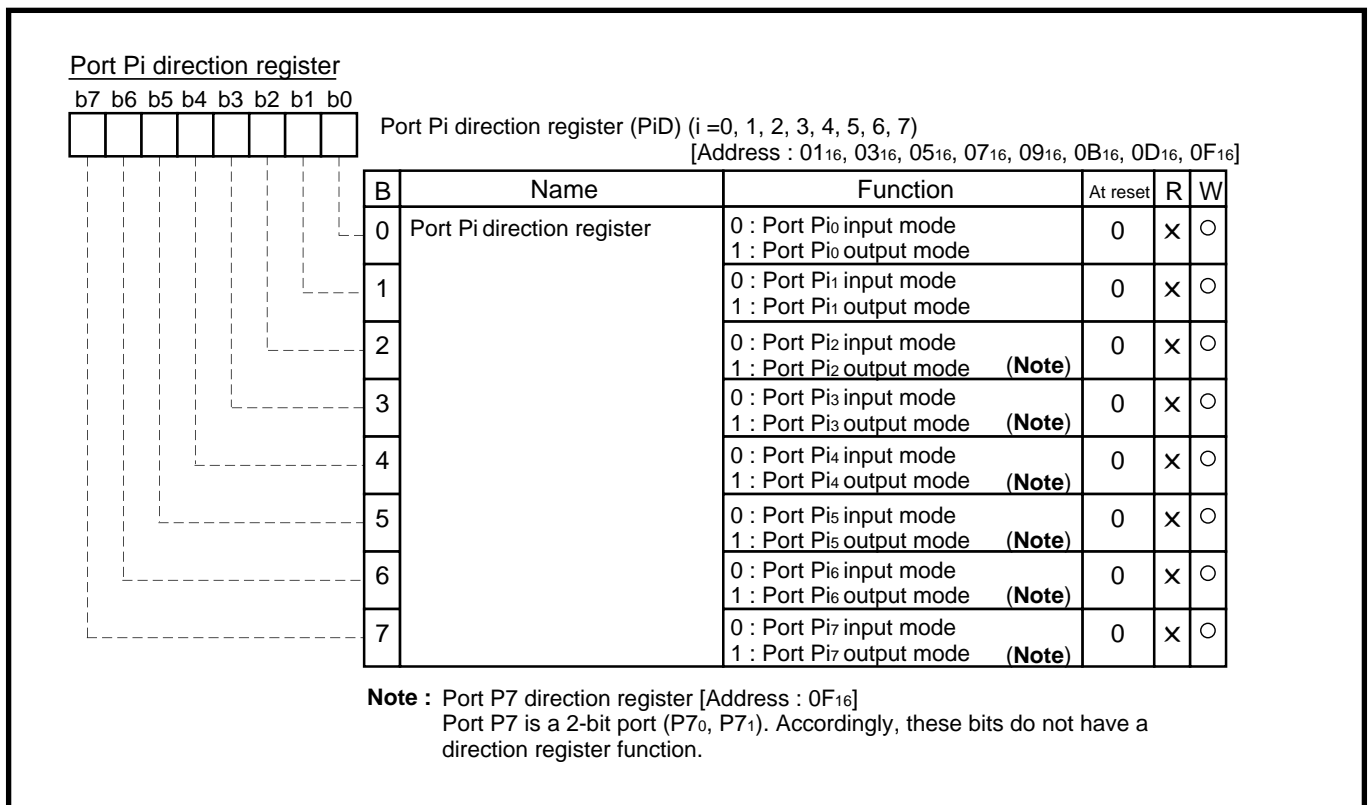


Fig. 3.5.2 Structure of Port Pi direction register (i = 0, 1, 2, 3, 4, 5, 6, 7)

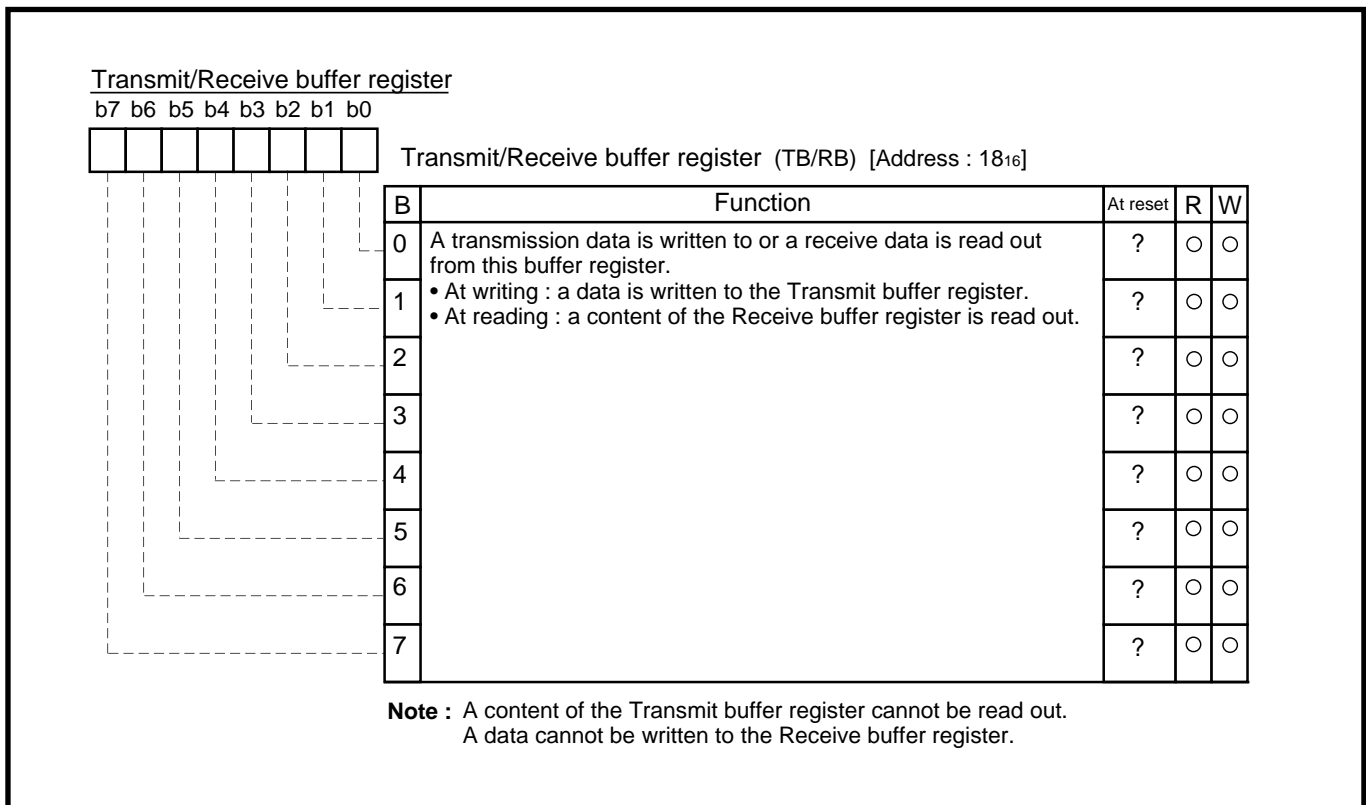


Fig. 3.5.3 Structure of Transmit/Receive buffer register

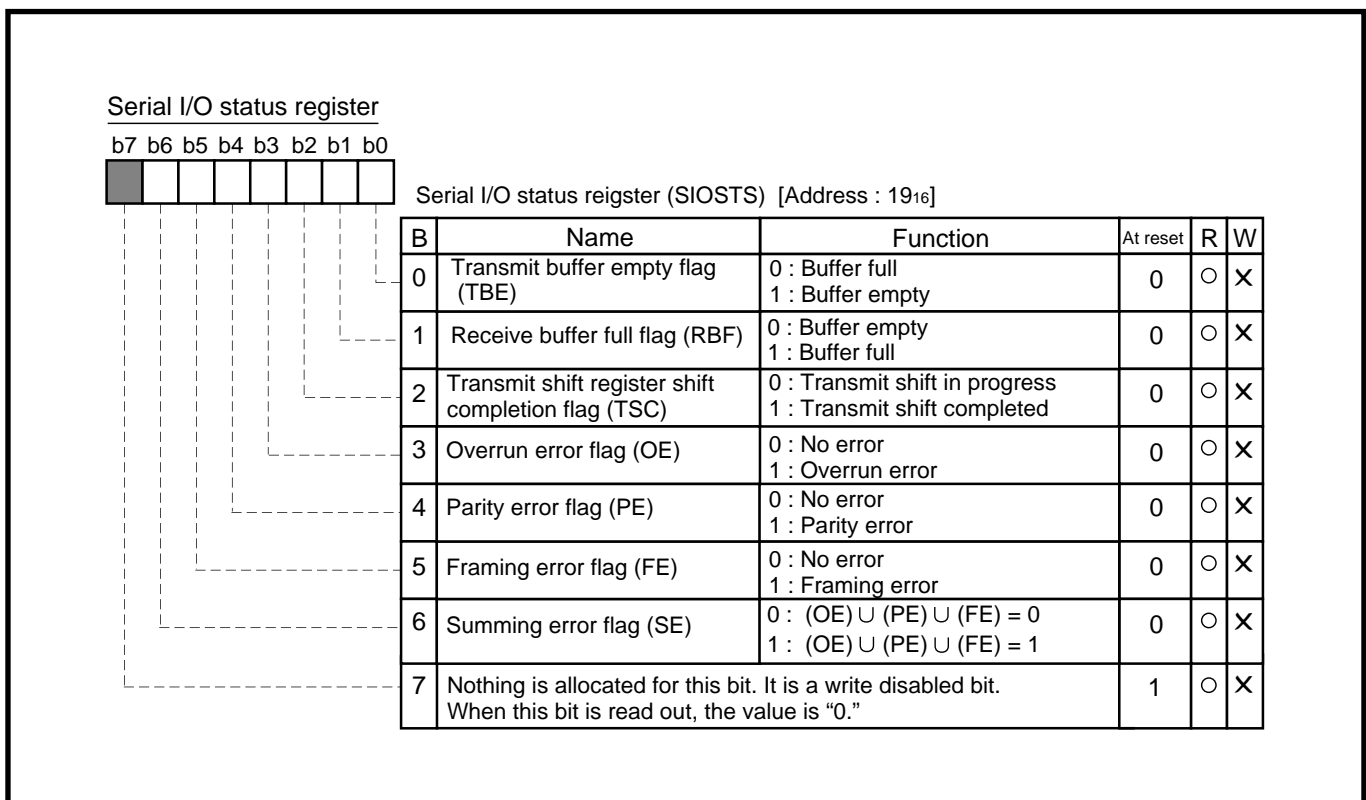


Fig. 3.5.4 Structure of Serial I/O status register

# APPENDIX

## 3.5 List of registers

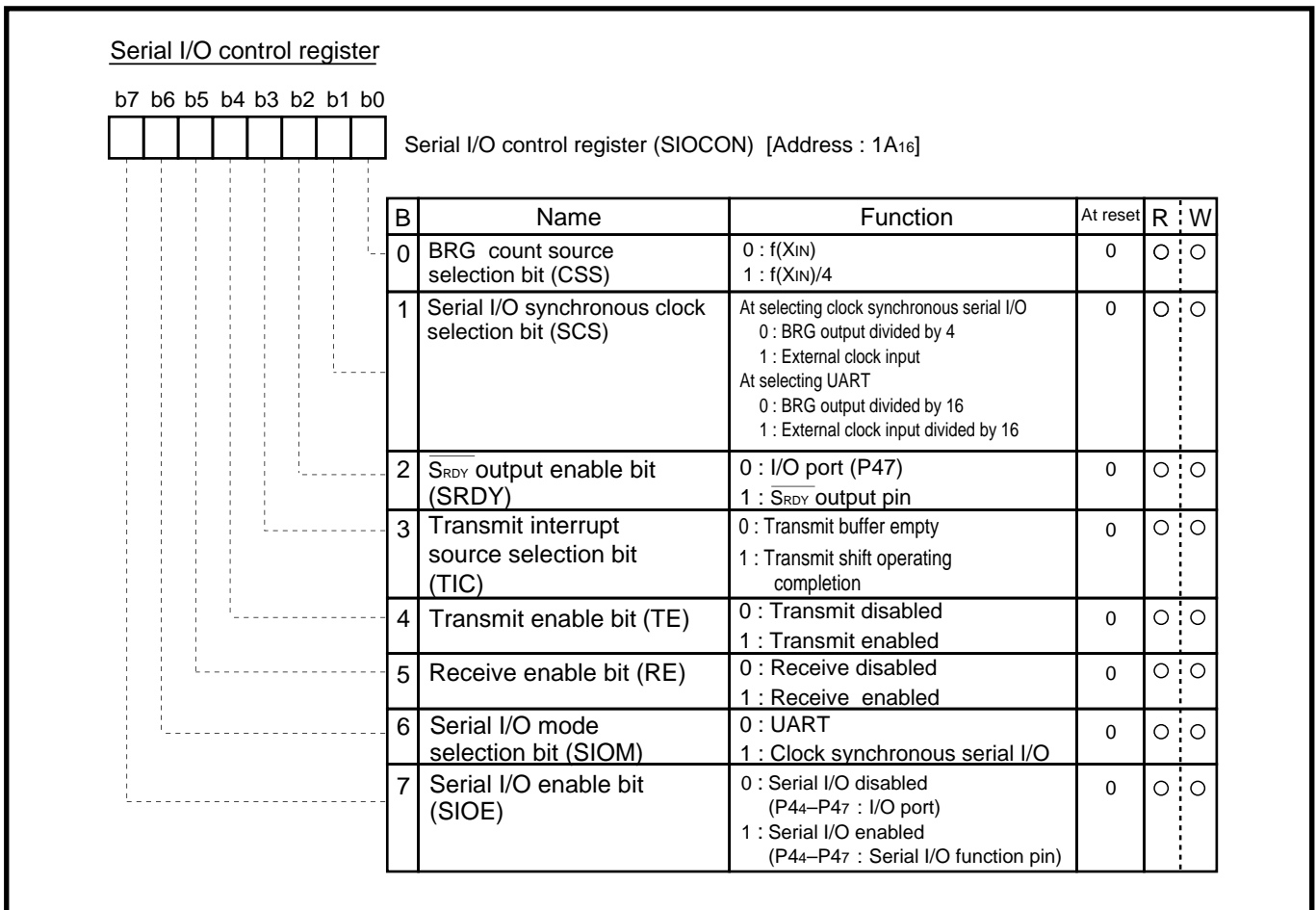


Fig. 3.5.5 Structure of Serial I/O control register

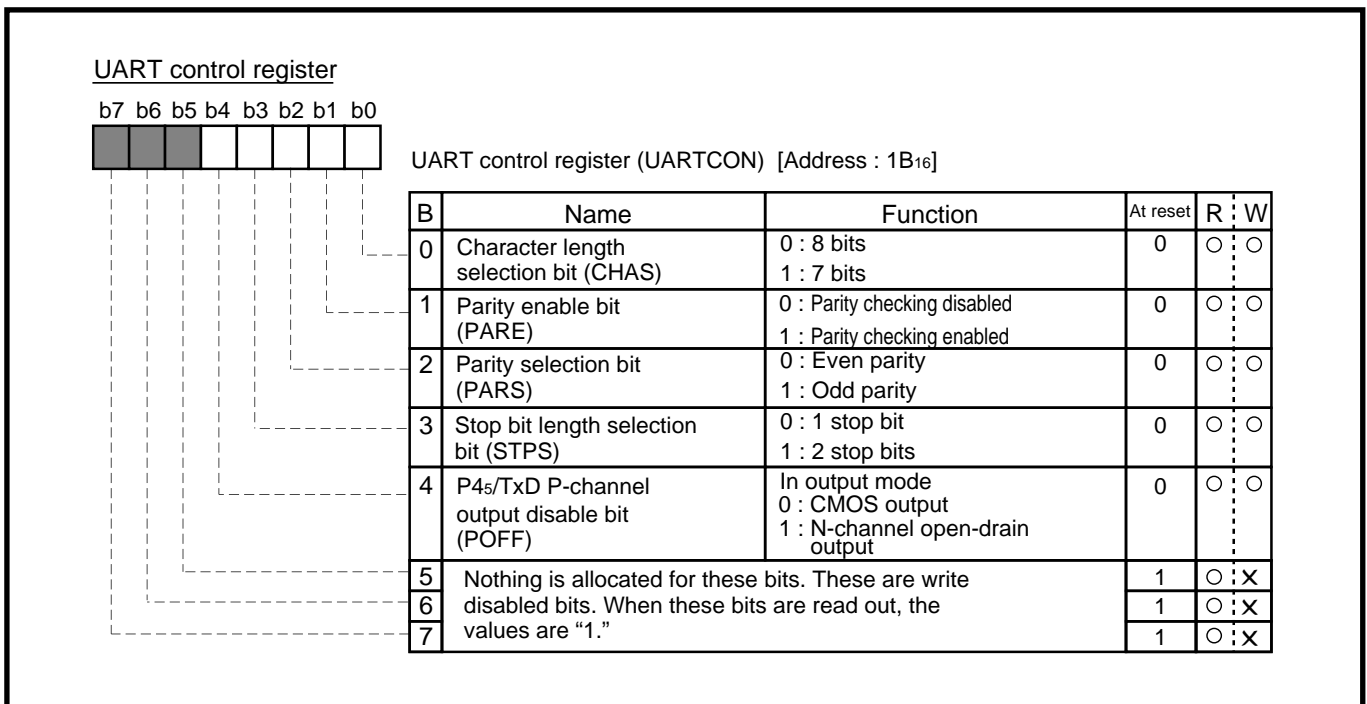


Fig. 3.5.6 Structure of UART control register

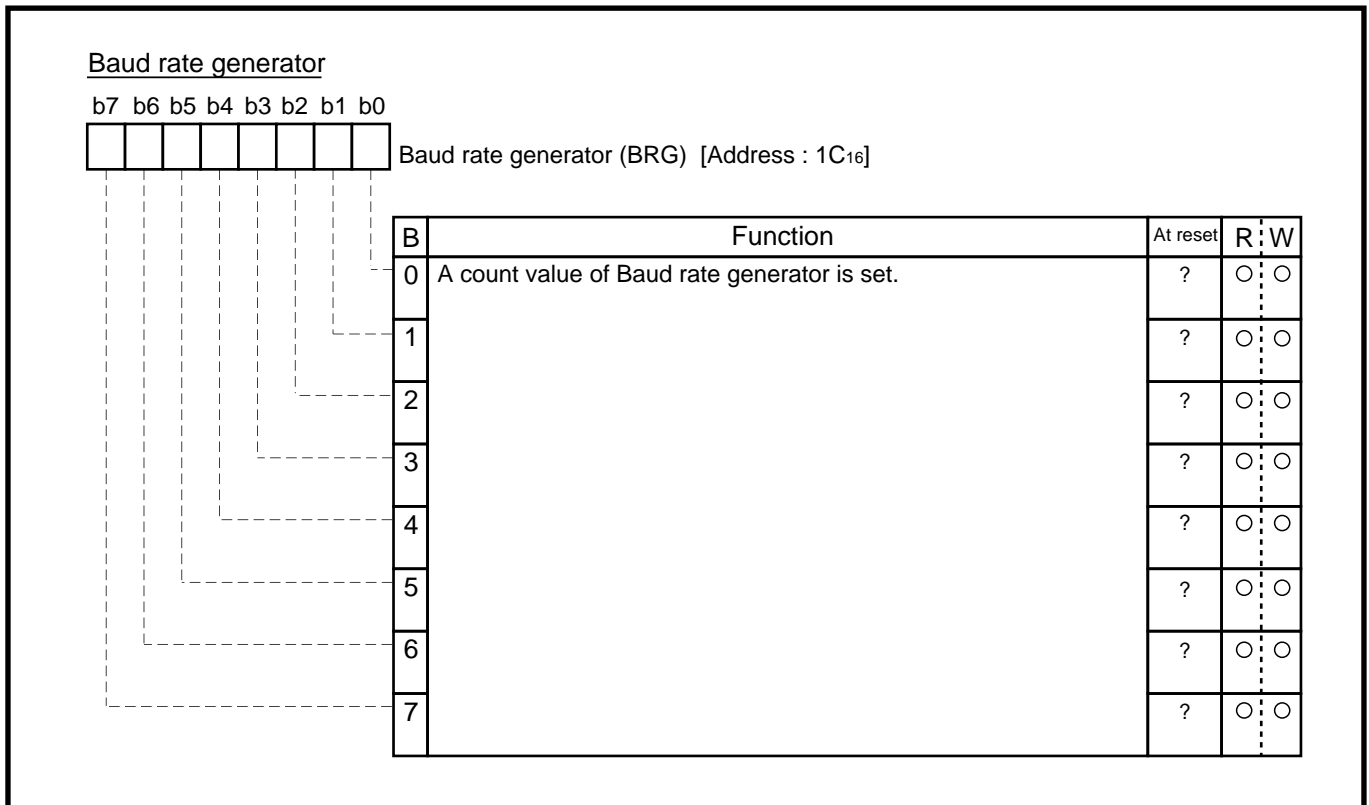


Fig. 3.5.7 Structure of Baud rate generator

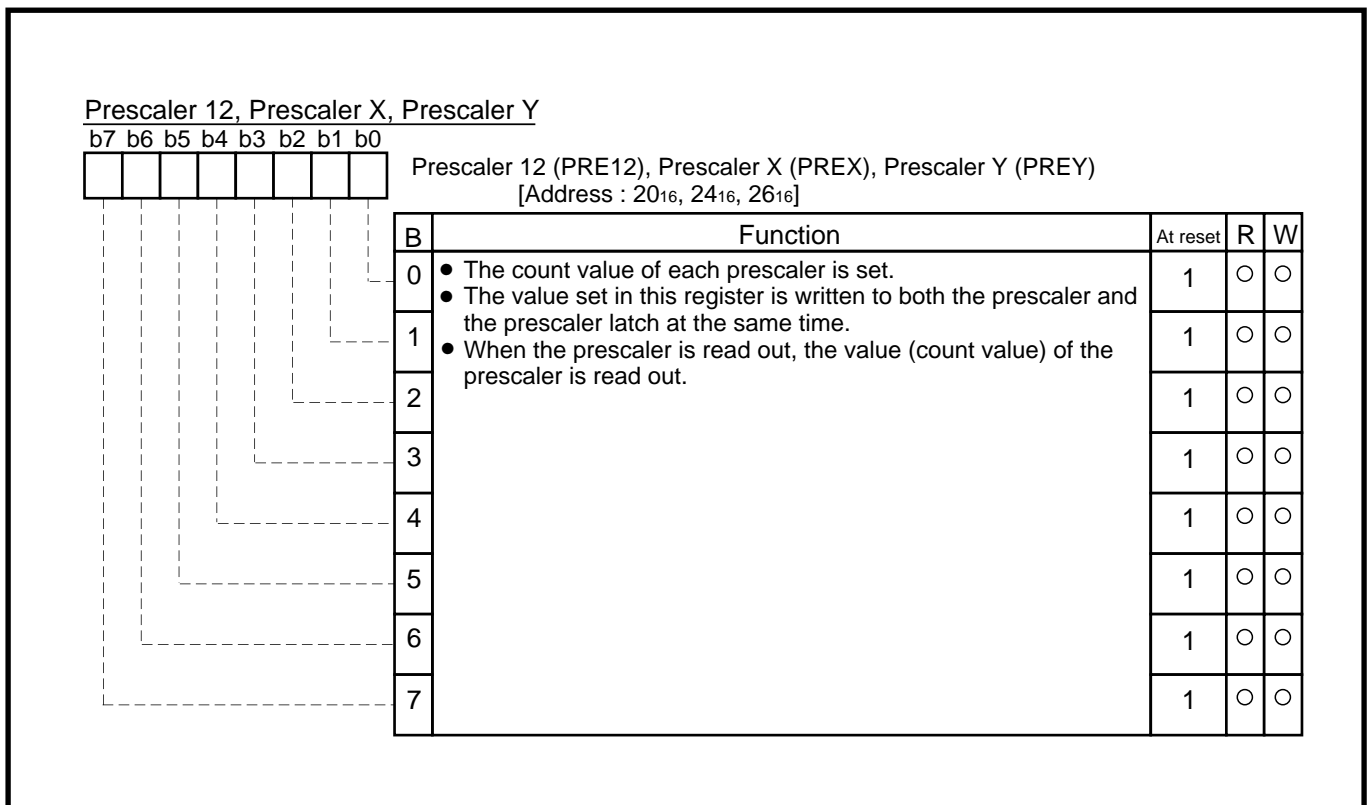


Fig. 3.5.8 Structure of Prescaler 12, Prescaler X, Prescaler Y

# APPENDIX

## 3.5 List of registers

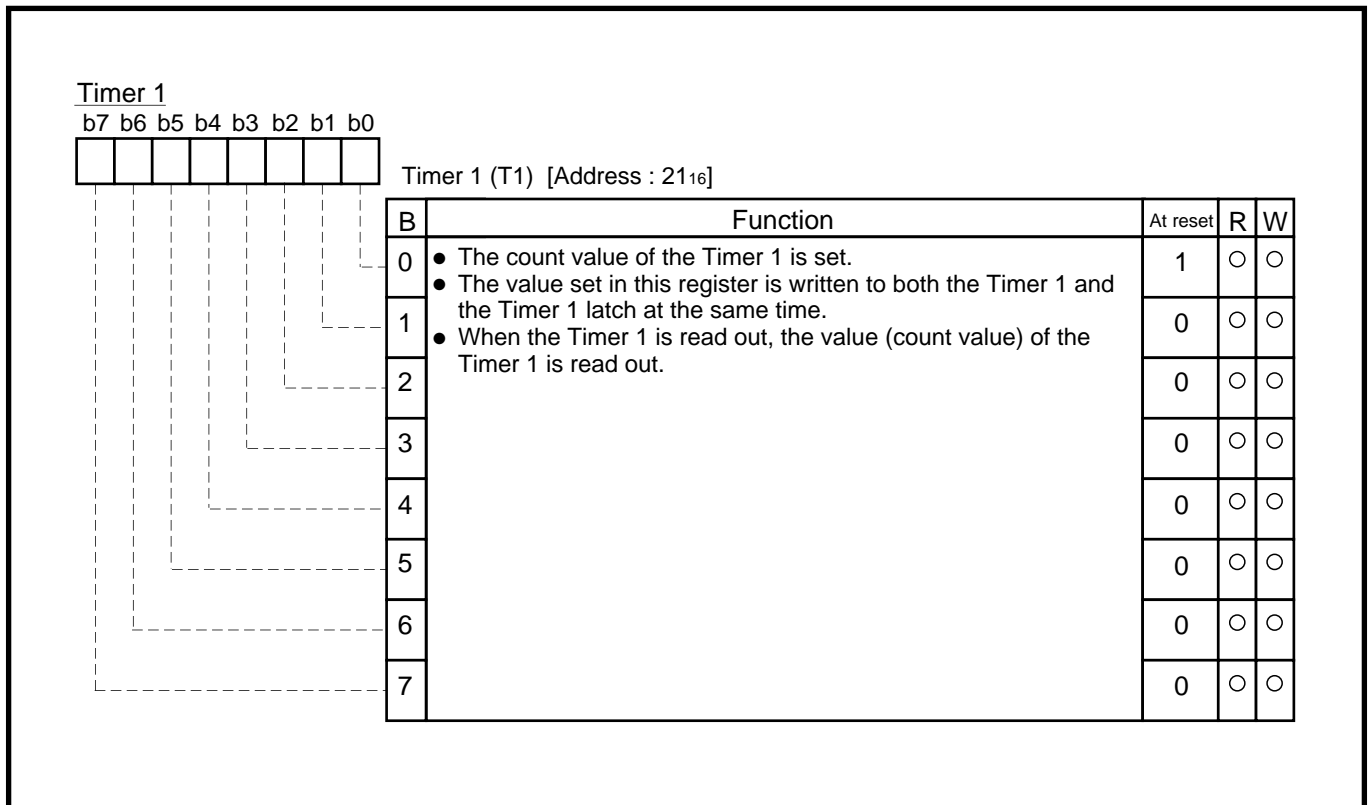


Fig. 3.5.9 Structure of Timer 1

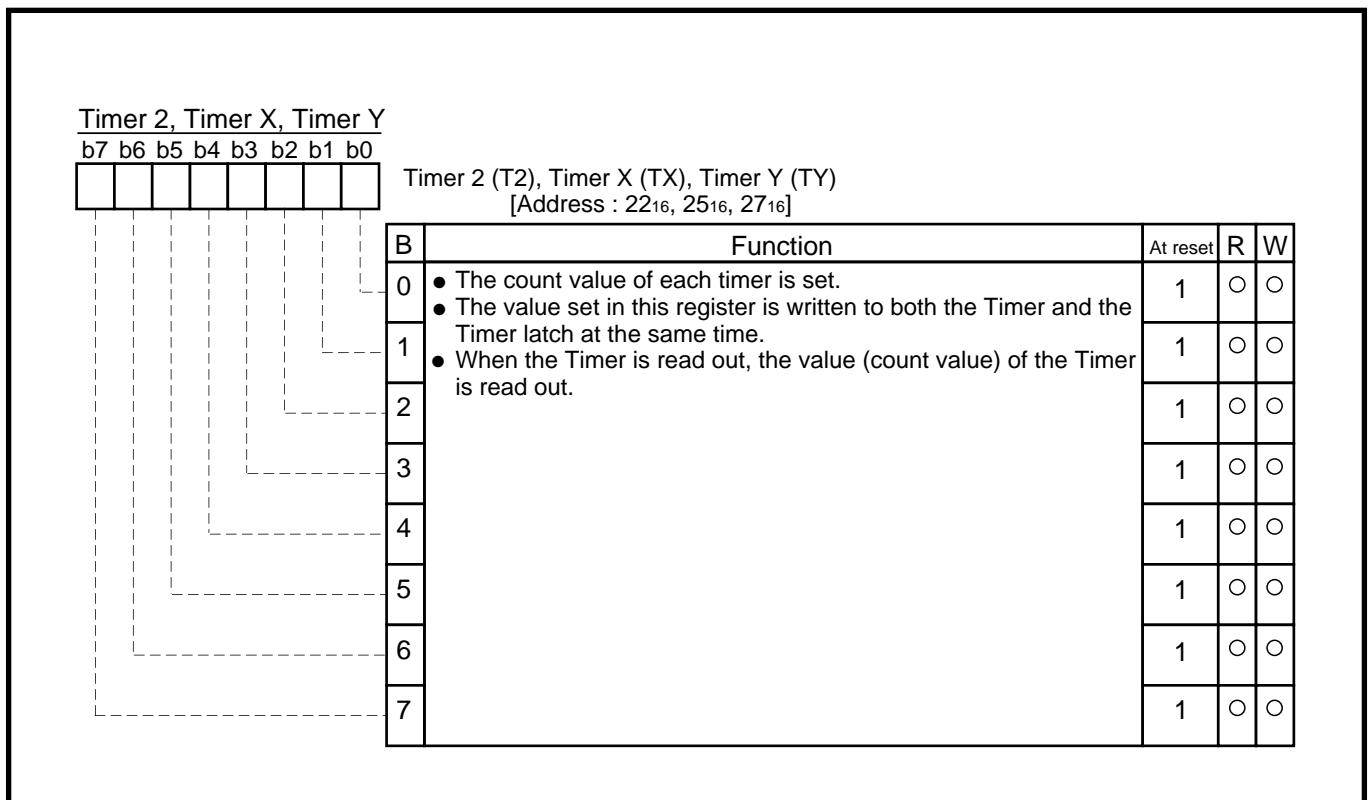


Fig. 3.5.10 Structure of Timer 2, Timer X, Timer Y

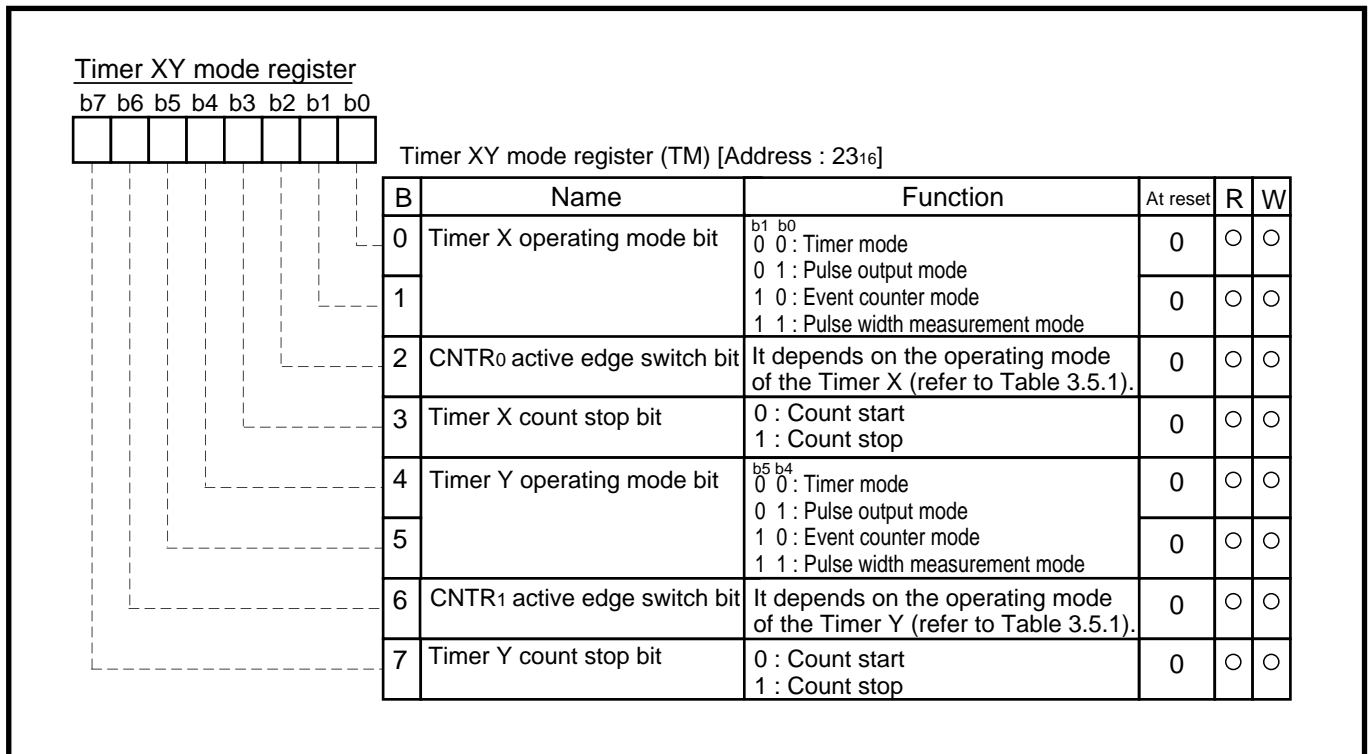


Fig. 3.5.11 Structure of Timer XY mode register

Table. 3.5.1 Function of CNTR0/CNTR1 edge switch bit

Operating mode of Timer X/Timer Y	Function of CNTR0/CNTR1 edge switch bit (bits 2 and 6)	
Timer mode	"0"	• Generation of CNTR0/CNTR1 interrupt request : Falling edge (No effect on timer count)
	"1"	• Generation of CNTR0/CNTR1 interrupt request : Rising edge (No effect on timer count)
Pulse output mode	"0"	• Start of pulse output : From "H" level • Generation of CNTR0/CNTR1 interrupt request : Falling edge
	"1"	• Start of pulse output : From "L" level • Generation of CNTR0/CNTR1 interrupt request : Rising edge
Event counter mode	"0"	• Timer X/Timer Y : Count of rising edge • Generation of CNTR0/CNTR1 interrupt request : Falling edge
	"1"	• Timer X/Timer Y : Count of falling edge • Generation of CNTR0/CNTR1 interrupt request : Rising edge
Pulse width measurement mode	"0"	• Timer X/Timer Y : Measurement of "H" level width • Generation of CNTR0/CNTR1 interrupt request : Falling edge
	"1"	• Timer X/Timer Y : Measurement of "L" level width • Generation of CNTR0/CNTR1 interrupt request : Rising edge

# APPENDIX

## 3.5 List of registers

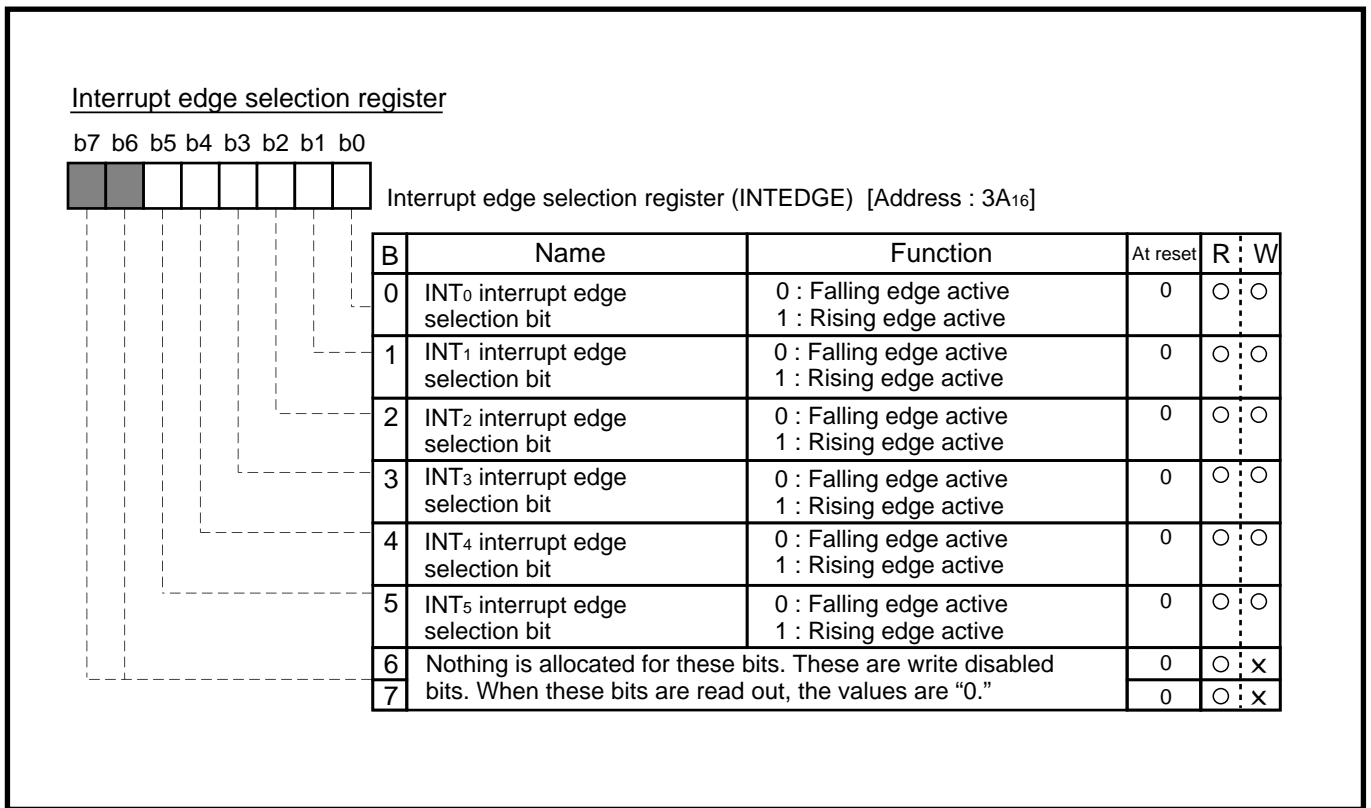


Fig. 3.5.12 Structure of Interrupt edge selection register

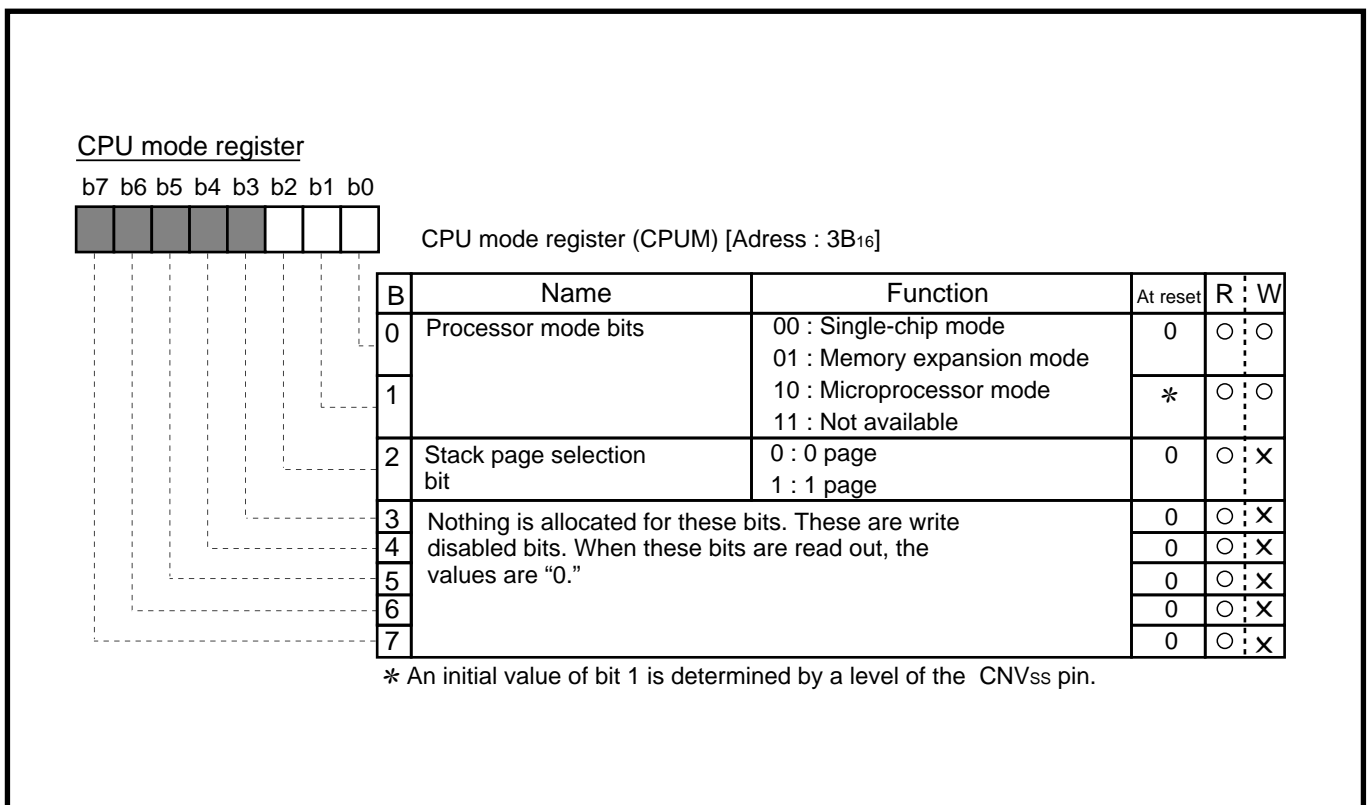


Fig. 3.5.13 Structure of CPU mode register



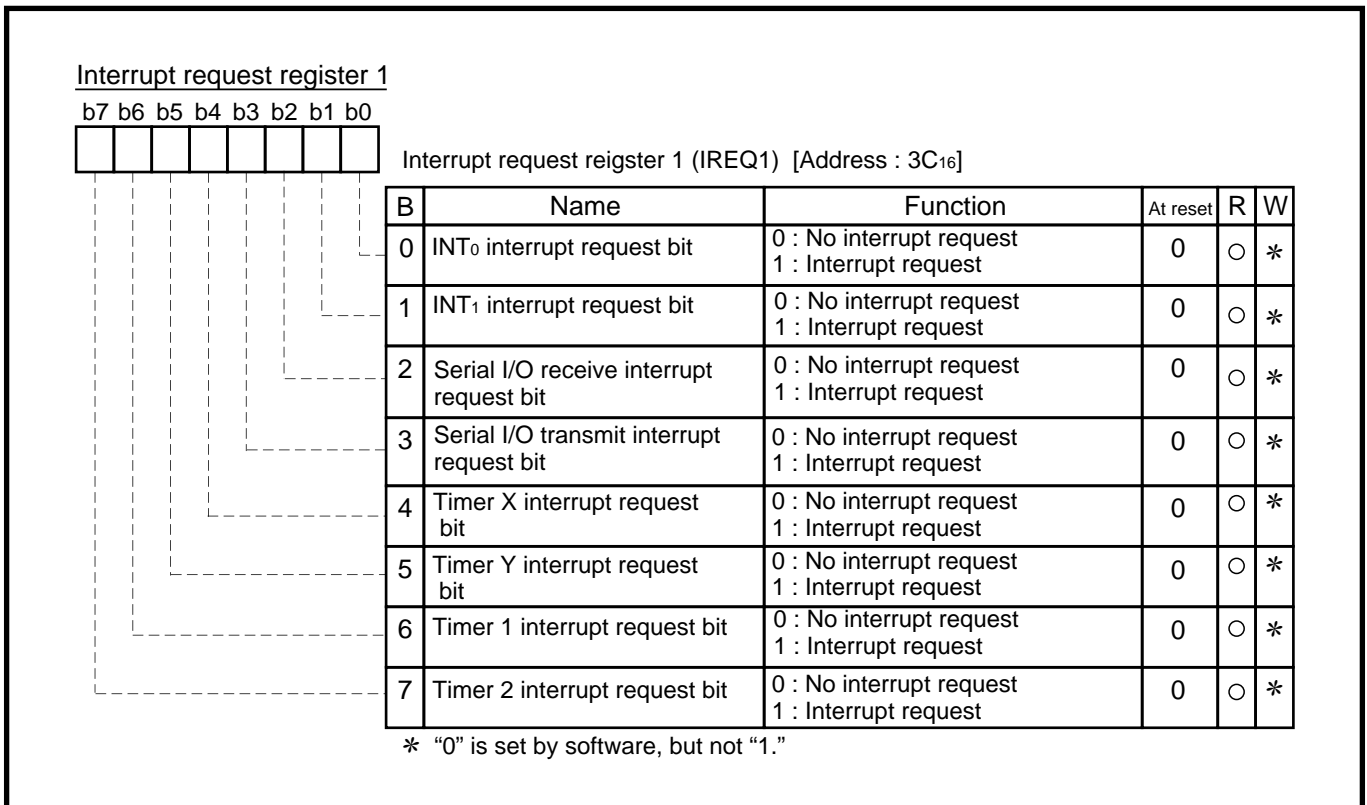


Fig. 3.5.14 Structure of Interrupt request register 1

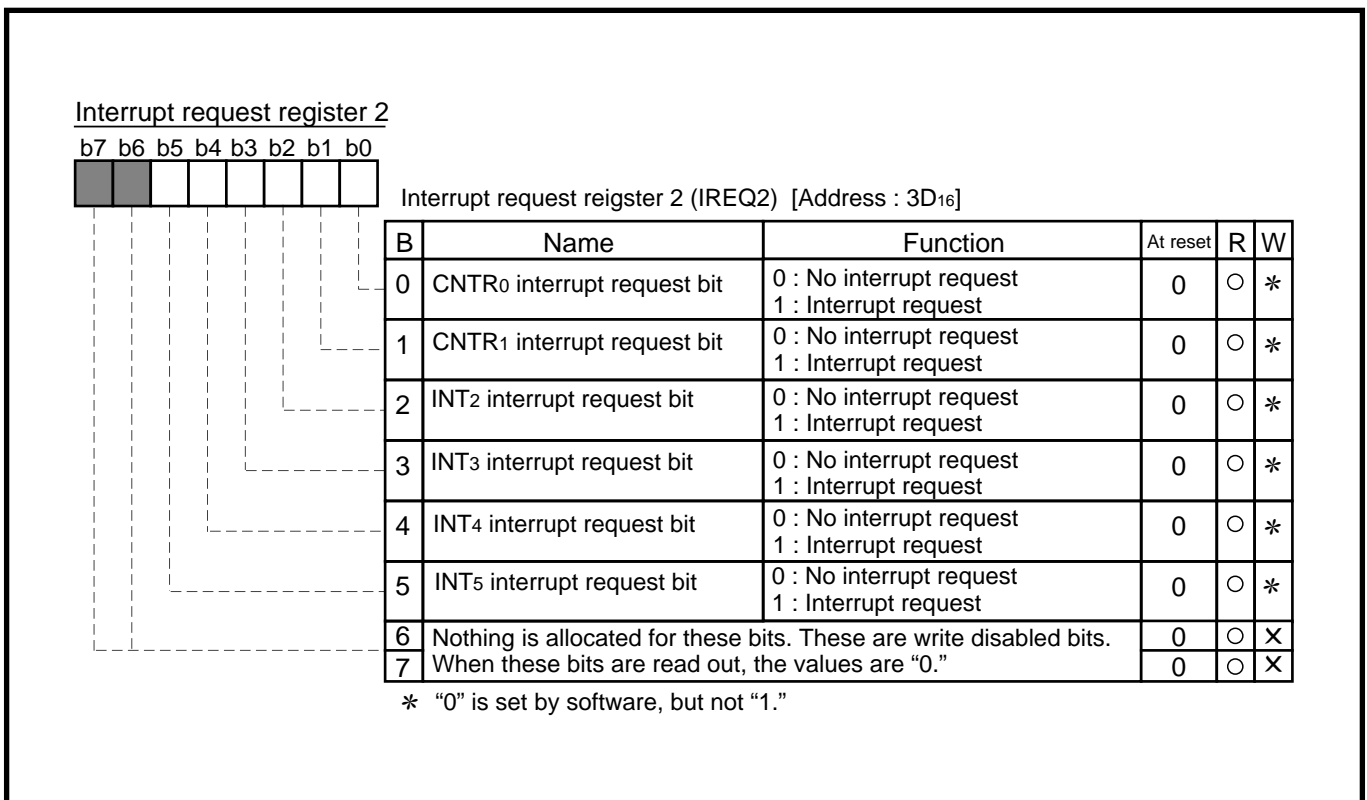


Fig. 3.5.15 Structure of Interrupt request register 2

# APPENDIX

## 3.5 List of registers

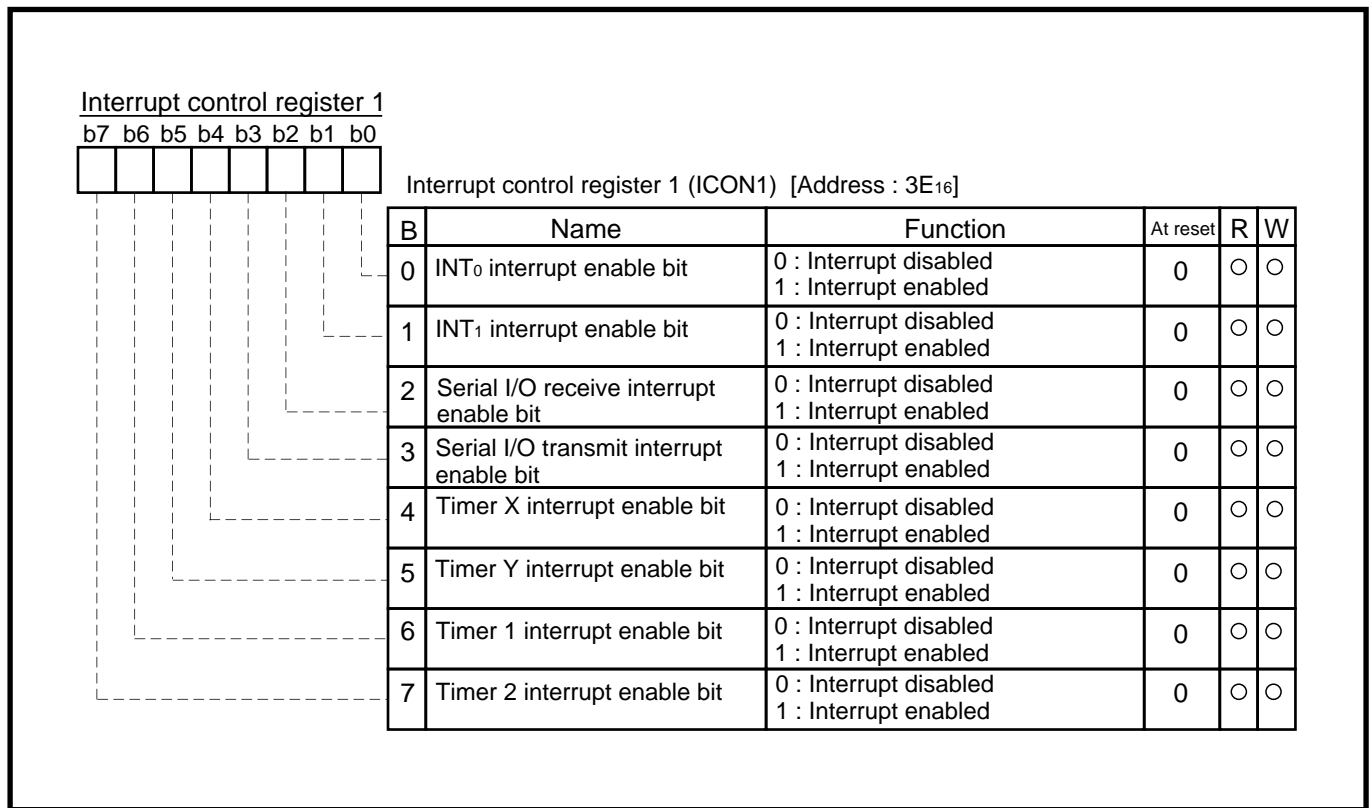


Fig. 3.5.16 Structure of Interrupt control register 1

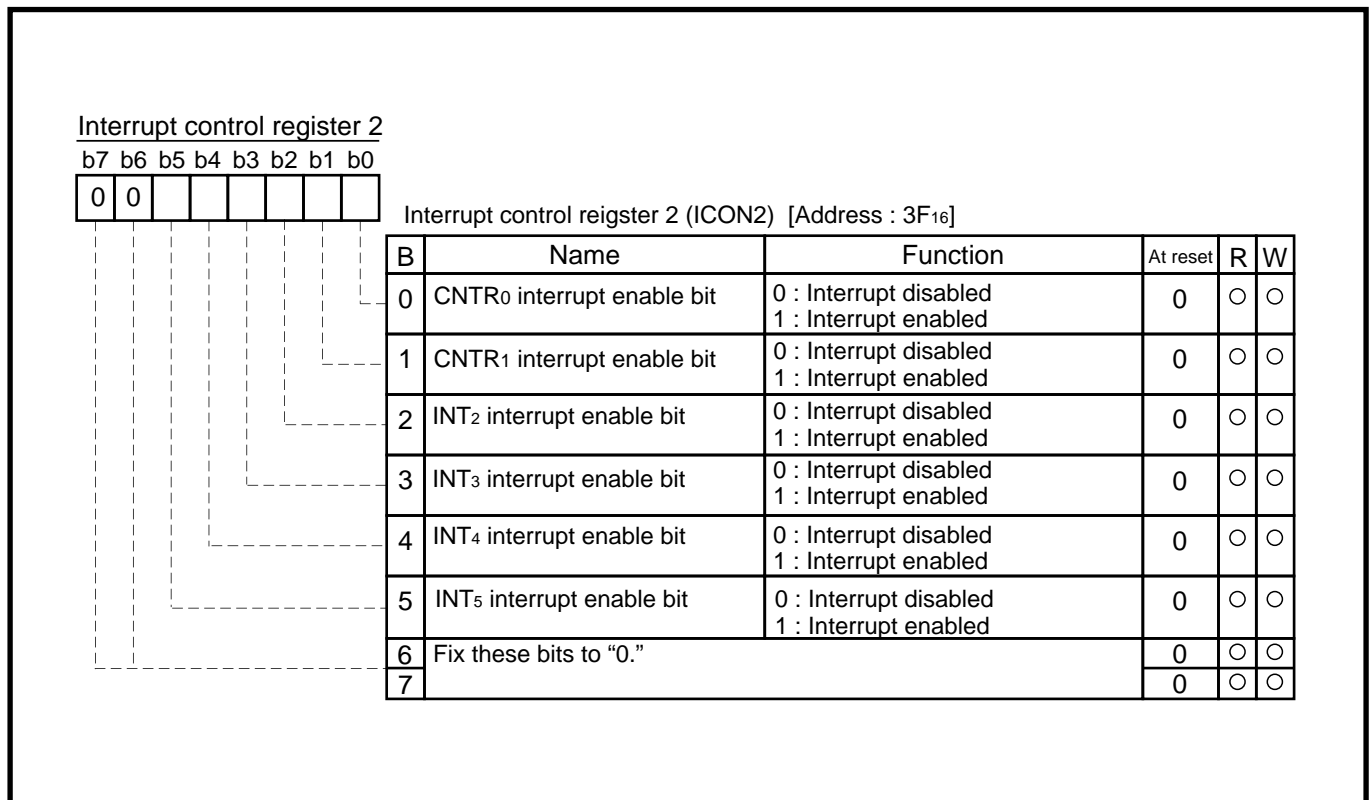


Fig. 3.5.17 Structure of Interrupt control register 2

### 3.6 Mask ROM ordering method

GZZ-SH04-34B<13B0>

Mask ROM number	
-----------------	--

**740 FAMILY MASK ROM CONFIRMATION FORM  
SINGLE-CHIP MICROCOMPUTER M38002M2-XXXSP/FP  
MITSUBISHI ELECTRIC**

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked \*.

* Customer	Company name	TEL (      )	Issuance signature	Submitted by	Supervisor
	Date issued	Date:			

\* 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.

Three EPROMs are required for each pattern.

If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

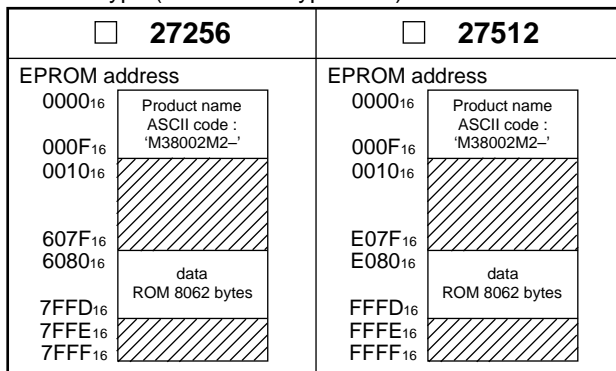
Microcomputer name :     M38002M2-XXXSP                       M38002M2-XXXFP

Checksum code for entire EPROM 

--	--	--	--

 (hexadecimal notation)

EPROM type (indicate the type used)



In the address space of the microcomputer, the internal ROM area is from address E080<sub>16</sub> to FFFD<sub>16</sub>. The reset vector is stored in addresses FFFC<sub>16</sub> and FFFD<sub>16</sub>.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF<sub>16</sub>".
- (2) The ASCII codes of the product name "M38002M2-" must be entered in addresses 0000<sub>16</sub> to 0008<sub>16</sub>. And set the data "FF<sub>16</sub>" in addresses 0009<sub>16</sub> to 000F<sub>16</sub>. The ASCII codes and addresses are listed to the right in hexadecimal notation.

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">Address</td><td></td></tr> <tr><td style="text-align: center;">0000<sub>16</sub></td><td style="text-align: center;">'M' = 4D<sub>16</sub></td></tr> <tr><td style="text-align: center;">0001<sub>16</sub></td><td style="text-align: center;">'3' = 33<sub>16</sub></td></tr> <tr><td style="text-align: center;">0002<sub>16</sub></td><td style="text-align: center;">'8' = 38<sub>16</sub></td></tr> <tr><td style="text-align: center;">0003<sub>16</sub></td><td style="text-align: center;">'0' = 30<sub>16</sub></td></tr> <tr><td style="text-align: center;">0004<sub>16</sub></td><td style="text-align: center;">'0' = 30<sub>16</sub></td></tr> <tr><td style="text-align: center;">0005<sub>16</sub></td><td style="text-align: center;">'2' = 32<sub>16</sub></td></tr> <tr><td style="text-align: center;">0006<sub>16</sub></td><td style="text-align: center;">'M' = 4D<sub>16</sub></td></tr> <tr><td style="text-align: center;">0007<sub>16</sub></td><td style="text-align: center;">'2' = 32<sub>16</sub></td></tr> </table>	Address		0000 <sub>16</sub>	'M' = 4D <sub>16</sub>	0001 <sub>16</sub>	'3' = 33 <sub>16</sub>	0002 <sub>16</sub>	'8' = 38 <sub>16</sub>	0003 <sub>16</sub>	'0' = 30 <sub>16</sub>	0004 <sub>16</sub>	'0' = 30 <sub>16</sub>	0005 <sub>16</sub>	'2' = 32 <sub>16</sub>	0006 <sub>16</sub>	'M' = 4D <sub>16</sub>	0007 <sub>16</sub>	'2' = 32 <sub>16</sub>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">Address</td><td></td></tr> <tr><td style="text-align: center;">0008<sub>16</sub></td><td style="text-align: center;">'-' = 2D<sub>16</sub></td></tr> <tr><td style="text-align: center;">0009<sub>16</sub></td><td style="text-align: center;">FF<sub>16</sub></td></tr> <tr><td style="text-align: center;">000A<sub>16</sub></td><td style="text-align: center;">FF<sub>16</sub></td></tr> <tr><td style="text-align: center;">000B<sub>16</sub></td><td style="text-align: center;">FF<sub>16</sub></td></tr> <tr><td style="text-align: center;">000C<sub>16</sub></td><td style="text-align: center;">FF<sub>16</sub></td></tr> <tr><td style="text-align: center;">000D<sub>16</sub></td><td style="text-align: center;">FF<sub>16</sub></td></tr> <tr><td style="text-align: center;">000E<sub>16</sub></td><td style="text-align: center;">FF<sub>16</sub></td></tr> <tr><td style="text-align: center;">000F<sub>16</sub></td><td style="text-align: center;">FF<sub>16</sub></td></tr> </table>	Address		0008 <sub>16</sub>	'-' = 2D <sub>16</sub>	0009 <sub>16</sub>	FF <sub>16</sub>	000A <sub>16</sub>	FF <sub>16</sub>	000B <sub>16</sub>	FF <sub>16</sub>	000C <sub>16</sub>	FF <sub>16</sub>	000D <sub>16</sub>	FF <sub>16</sub>	000E <sub>16</sub>	FF <sub>16</sub>	000F <sub>16</sub>	FF <sub>16</sub>
Address																																					
0000 <sub>16</sub>	'M' = 4D <sub>16</sub>																																				
0001 <sub>16</sub>	'3' = 33 <sub>16</sub>																																				
0002 <sub>16</sub>	'8' = 38 <sub>16</sub>																																				
0003 <sub>16</sub>	'0' = 30 <sub>16</sub>																																				
0004 <sub>16</sub>	'0' = 30 <sub>16</sub>																																				
0005 <sub>16</sub>	'2' = 32 <sub>16</sub>																																				
0006 <sub>16</sub>	'M' = 4D <sub>16</sub>																																				
0007 <sub>16</sub>	'2' = 32 <sub>16</sub>																																				
Address																																					
0008 <sub>16</sub>	'-' = 2D <sub>16</sub>																																				
0009 <sub>16</sub>	FF <sub>16</sub>																																				
000A <sub>16</sub>	FF <sub>16</sub>																																				
000B <sub>16</sub>	FF <sub>16</sub>																																				
000C <sub>16</sub>	FF <sub>16</sub>																																				
000D <sub>16</sub>	FF <sub>16</sub>																																				
000E <sub>16</sub>	FF <sub>16</sub>																																				
000F <sub>16</sub>	FF <sub>16</sub>																																				

# APPENDIX

## 3.6 Mask ROM ordering method

---

GZZ-SH04-34B<13B0>

Mask ROM number	
-----------------	--

### 740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M38002M2-XXXSP/FP MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program.

EPROM type	27256	27512
The pseudo-command	*= $\Delta$ \$8000 .BYTE $\Delta$ 'M38002M2-'	*= $\Delta$ \$0000 .BYTE $\Delta$ 'M38002M2-'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

※ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (64P4B for M38002M2-XXXSP, 64P6N for M38002M2-XXXFP) and attach it to the mask ROM confirmation form.

※ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the X<sub>IN</sub>-X<sub>OUT</sub> oscillator?

- |   |   |
|---|---|
| <input type="checkbox"/> Ceramic resonator    | <input type="checkbox"/> Quartz crystal       |
| <input type="checkbox"/> External clock input | <input type="checkbox"/> Other (            ) |

At what frequency?

f(X<sub>IN</sub>) =  MHz

(2) In which operation mode will you use your microcomputer?

- |  |  |
|--|--|
| <input type="checkbox"/> Single-chip mode    | <input type="checkbox"/> Memory expansion mode |
| <input type="checkbox"/> Microprocessor mode |  |

※ 4. Comments

(2/2)

GZZ-SH04-79B<16A0>

Mask ROM number	
-----------------	--

**740 FAMILY MASK ROM CONFIRMATION FORM  
SINGLE-CHIP MICROCOMPUTER M38002M2DXXXSP/FP  
MITSUBISHI ELECTRIC**

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked \*.

* Customer	Company name	TEL (      )	Issuance signature	Submitted by	Supervisor
	Date issued	Date:			

\* 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.

Three EPROMs are required for each pattern.

If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name :     M38002M2DXXXSP             M38002M2DXXXFP

Checksum code for entire EPROM 

--	--	--	--

 (hexadecimal notation)

EPROM type (indicate the type used)

<input type="checkbox"/> <b>27256</b>	<input type="checkbox"/> <b>27512</b>																										
<table border="1"> <tr><td colspan="2">EPROM address</td></tr> <tr><td>0000<sub>16</sub></td><td rowspan="3">Product name ASCII code : 'M38002M2D'</td></tr> <tr><td>000F<sub>16</sub></td></tr> <tr><td>0010<sub>16</sub></td></tr> <tr><td>607F<sub>16</sub></td><td rowspan="2">data ROM 8062 bytes</td></tr> <tr><td>6080<sub>16</sub></td></tr> <tr><td>7FFD<sub>16</sub></td><td rowspan="3">data ROM 8062 bytes</td></tr> <tr><td>7FFE<sub>16</sub></td></tr> <tr><td>7FFF<sub>16</sub></td></tr> </table>	EPROM address		0000 <sub>16</sub>	Product name ASCII code : 'M38002M2D'	000F <sub>16</sub>	0010 <sub>16</sub>	607F <sub>16</sub>	data ROM 8062 bytes	6080 <sub>16</sub>	7FFD <sub>16</sub>	data ROM 8062 bytes	7FFE <sub>16</sub>	7FFF <sub>16</sub>	<table border="1"> <tr><td colspan="2">EPROM address</td></tr> <tr><td>0000<sub>16</sub></td><td rowspan="3">Product name ASCII code : 'M38002M2D'</td></tr> <tr><td>000F<sub>16</sub></td></tr> <tr><td>0010<sub>16</sub></td></tr> <tr><td>E07F<sub>16</sub></td><td rowspan="2">data ROM 8062 bytes</td></tr> <tr><td>E080<sub>16</sub></td></tr> <tr><td>FFFD<sub>16</sub></td><td rowspan="3">data ROM 8062 bytes</td></tr> <tr><td>FFFE<sub>16</sub></td></tr> <tr><td>FFFF<sub>16</sub></td></tr> </table>	EPROM address		0000 <sub>16</sub>	Product name ASCII code : 'M38002M2D'	000F <sub>16</sub>	0010 <sub>16</sub>	E07F <sub>16</sub>	data ROM 8062 bytes	E080 <sub>16</sub>	FFFD <sub>16</sub>	data ROM 8062 bytes	FFFE <sub>16</sub>	FFFF <sub>16</sub>
EPROM address																											
0000 <sub>16</sub>	Product name ASCII code : 'M38002M2D'																										
000F <sub>16</sub>																											
0010 <sub>16</sub>																											
607F <sub>16</sub>	data ROM 8062 bytes																										
6080 <sub>16</sub>																											
7FFD <sub>16</sub>	data ROM 8062 bytes																										
7FFE <sub>16</sub>																											
7FFF <sub>16</sub>																											
EPROM address																											
0000 <sub>16</sub>	Product name ASCII code : 'M38002M2D'																										
000F <sub>16</sub>																											
0010 <sub>16</sub>																											
E07F <sub>16</sub>	data ROM 8062 bytes																										
E080 <sub>16</sub>																											
FFFD <sub>16</sub>	data ROM 8062 bytes																										
FFFE <sub>16</sub>																											
FFFF <sub>16</sub>																											

In the address space of the microcomputer, the internal ROM area is from address E080<sub>16</sub> to FFFD<sub>16</sub>. The reset vector is stored in addresses FFFC<sub>16</sub> and FFFD<sub>16</sub>.

- Set the data in the unused area (the shaded area of the diagram) to "FF<sub>16</sub>".
- The ASCII codes of the product name "M38002M2D" must be entered in addresses 0000<sub>16</sub> to 0008<sub>16</sub>. And set the data "FF<sub>16</sub>" in addresses 0009<sub>16</sub> to 000F<sub>16</sub>. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000 <sub>16</sub>	'M' = 4D <sub>16</sub>	0008 <sub>16</sub>	'D' = 44 <sub>16</sub>
0001 <sub>16</sub>	'3' = 33 <sub>16</sub>	0009 <sub>16</sub>	FF <sub>16</sub>
0002 <sub>16</sub>	'8' = 38 <sub>16</sub>	000A <sub>16</sub>	FF <sub>16</sub>
0003 <sub>16</sub>	'0' = 30 <sub>16</sub>	000B <sub>16</sub>	FF <sub>16</sub>
0004 <sub>16</sub>	'0' = 30 <sub>16</sub>	000C <sub>16</sub>	FF <sub>16</sub>
0005 <sub>16</sub>	'2' = 32 <sub>16</sub>	000D <sub>16</sub>	FF <sub>16</sub>
0006 <sub>16</sub>	'M' = 4D <sub>16</sub>	000E <sub>16</sub>	FF <sub>16</sub>
0007 <sub>16</sub>	'2' = 32 <sub>16</sub>	000F <sub>16</sub>	FF <sub>16</sub>

# APPENDIX

## 3.6 Mask ROM ordering method

---

GZZ-SH04-79B<16A0>

Mask ROM number	
-----------------	--

### 740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M38002M2DXXXSP/FP MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program.

EPROM type	27256	27512
The pseudo-command	*= $\Delta$ \$8000 .BYTE $\Delta$ 'M38002M2D'	*= $\Delta$ \$0000 .BYTE $\Delta$ 'M38002M2D'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

※ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (64P4B for M38002M2DXXXSP, 64P6N for M38002M2DXXXFP) and attach it to the mask ROM confirmation form.

※ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the X<sub>IN</sub>-X<sub>OUT</sub> oscillator?

- |   |   |
|---|---|
| <input type="checkbox"/> Ceramic resonator    | <input type="checkbox"/> Quartz crystal       |
| <input type="checkbox"/> External clock input | <input type="checkbox"/> Other (            ) |

At what frequency?

f(X<sub>IN</sub>) =  MHz

(2) In which operation mode will you use your microcomputer?

- |  |  |
|--|--|
| <input type="checkbox"/> Single-chip mode    | <input type="checkbox"/> Memory expansion mode |
| <input type="checkbox"/> Microprocessor mode |  |

※ 4. Comments

(2/2)



# APPENDIX

## 3.6 Mask ROM ordering method

---

GZZ-SH03-22B<9YB0>

Mask ROM number	
-----------------	--

### 740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M38002M4-XXXSP/FP MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program.

EPROM type	27256	27512
The pseudo-command	*= $\Delta$ \$8000 .BYTE $\Delta$ 'M38002M4-'	*= $\Delta$ \$0000 .BYTE $\Delta$ 'M38002M4-'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

※ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (64P4B for M38002M4-XXXSP, 64P6N for M38002M4-XXXFP) and attach it to the mask ROM confirmation form.

※ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the X<sub>IN</sub>-X<sub>OUT</sub> oscillator?

- |   |   |
|---|---|
| <input type="checkbox"/> Ceramic resonator    | <input type="checkbox"/> Quartz crystal       |
| <input type="checkbox"/> External clock input | <input type="checkbox"/> Other (            ) |

At what frequency?

f(X<sub>IN</sub>) =  MHz

(2) In which operation mode will you use your microcomputer?

- |  |  |
|--|--|
| <input type="checkbox"/> Single-chip mode    | <input type="checkbox"/> Memory expansion mode |
| <input type="checkbox"/> Microprocessor mode |  |

※ 4. Comments

(2/2)



GZZ-SH05-12B<21A0>

Mask ROM number	
-----------------	--

**740 FAMILY MASK ROM CONFIRMATION FORM  
SINGLE-CHIP MICROCOMPUTER M38002M4DXXXSP/FP  
MITSUBISHI ELECTRIC**

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked \*.

* Customer	Company name	TEL (      )	Issuance signature	Submitted by	Supervisor
	Date issued	Date:			

\* 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.

Three EPROMs are required for each pattern.

If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name :     M38002M4DXXXSP             M38002M4DXXXFP

Checksum code for entire EPROM 

--	--	--	--

 (hexadecimal notation)

EPROM type (indicate the type used)

<input type="checkbox"/> <b>27256</b>	<input type="checkbox"/> <b>27512</b>																																		
<table border="1"> <tr><td colspan="2">EPROM address</td></tr> <tr><td>0000<sub>16</sub></td><td rowspan="3">Product name ASCII code : 'M38002M4D'</td></tr> <tr><td>000F<sub>16</sub></td></tr> <tr><td>0010<sub>16</sub></td></tr> <tr><td> </td><td> </td></tr> <tr><td>407F<sub>16</sub></td><td rowspan="2">data ROM 16254 bytes</td></tr> <tr><td>4080<sub>16</sub></td></tr> <tr><td> </td><td> </td></tr> <tr><td>7FFD<sub>16</sub></td><td rowspan="3">data ROM 16254 bytes</td></tr> <tr><td>7FFE<sub>16</sub></td></tr> <tr><td>7FFF<sub>16</sub></td></tr> </table>	EPROM address		0000 <sub>16</sub>	Product name ASCII code : 'M38002M4D'	000F <sub>16</sub>	0010 <sub>16</sub>			407F <sub>16</sub>	data ROM 16254 bytes	4080 <sub>16</sub>			7FFD <sub>16</sub>	data ROM 16254 bytes	7FFE <sub>16</sub>	7FFF <sub>16</sub>	<table border="1"> <tr><td colspan="2">EPROM address</td></tr> <tr><td>0000<sub>16</sub></td><td rowspan="3">Product name ASCII code : 'M38002M4D'</td></tr> <tr><td>000F<sub>16</sub></td></tr> <tr><td>0010<sub>16</sub></td></tr> <tr><td> </td><td> </td></tr> <tr><td>C07F<sub>16</sub></td><td rowspan="2">data ROM 16254 bytes</td></tr> <tr><td>C080<sub>16</sub></td></tr> <tr><td> </td><td> </td></tr> <tr><td>FFFD<sub>16</sub></td><td rowspan="3">data ROM 16254 bytes</td></tr> <tr><td>FFFE<sub>16</sub></td></tr> <tr><td>FFFF<sub>16</sub></td></tr> </table>	EPROM address		0000 <sub>16</sub>	Product name ASCII code : 'M38002M4D'	000F <sub>16</sub>	0010 <sub>16</sub>			C07F <sub>16</sub>	data ROM 16254 bytes	C080 <sub>16</sub>			FFFD <sub>16</sub>	data ROM 16254 bytes	FFFE <sub>16</sub>	FFFF <sub>16</sub>
EPROM address																																			
0000 <sub>16</sub>	Product name ASCII code : 'M38002M4D'																																		
000F <sub>16</sub>																																			
0010 <sub>16</sub>																																			
407F <sub>16</sub>	data ROM 16254 bytes																																		
4080 <sub>16</sub>																																			
7FFD <sub>16</sub>	data ROM 16254 bytes																																		
7FFE <sub>16</sub>																																			
7FFF <sub>16</sub>																																			
EPROM address																																			
0000 <sub>16</sub>	Product name ASCII code : 'M38002M4D'																																		
000F <sub>16</sub>																																			
0010 <sub>16</sub>																																			
C07F <sub>16</sub>	data ROM 16254 bytes																																		
C080 <sub>16</sub>																																			
FFFD <sub>16</sub>	data ROM 16254 bytes																																		
FFFE <sub>16</sub>																																			
FFFF <sub>16</sub>																																			

In the address space of the microcomputer, the internal ROM area is from address C080<sub>16</sub> to FFFD<sub>16</sub>. The reset vector is stored in addresses FFFC<sub>16</sub> and FFFD<sub>16</sub>.

- Set the data in the unused area (the shaded area of the diagram) to "FF<sub>16</sub>".
- The ASCII codes of the product name "M38002M4D" must be entered in addresses 0000<sub>16</sub> to 0008<sub>16</sub>. And set the data "FF<sub>16</sub>" in addresses 0009<sub>16</sub> to 000F<sub>16</sub>. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000 <sub>16</sub>	'M' = 4D <sub>16</sub>	0008 <sub>16</sub>	'D' = 44 <sub>16</sub>
0001 <sub>16</sub>	'3' = 33 <sub>16</sub>	0009 <sub>16</sub>	FF <sub>16</sub>
0002 <sub>16</sub>	'8' = 38 <sub>16</sub>	000A <sub>16</sub>	FF <sub>16</sub>
0003 <sub>16</sub>	'0' = 30 <sub>16</sub>	000B <sub>16</sub>	FF <sub>16</sub>
0004 <sub>16</sub>	'0' = 30 <sub>16</sub>	000C <sub>16</sub>	FF <sub>16</sub>
0005 <sub>16</sub>	'2' = 32 <sub>16</sub>	000D <sub>16</sub>	FF <sub>16</sub>
0006 <sub>16</sub>	'M' = 4D <sub>16</sub>	000E <sub>16</sub>	FF <sub>16</sub>
0007 <sub>16</sub>	'4' = 34 <sub>16</sub>	000F <sub>16</sub>	FF <sub>16</sub>

# APPENDIX

## 3.6 Mask ROM ordering method

---

GZZ-SH05-12B<21A0>

Mask ROM number	
-----------------	--

### 740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M38002M4DXXXSP/FP MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program.

EPROM type	27256	27512
The pseudo-command	*= $\Delta$ \$8000 .BYTE $\Delta$ 'M38002M4D'	*= $\Delta$ \$0000 .BYTE $\Delta$ 'M38002M4D'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

※ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (64P4B for M38002M4DXXXSP, 64P6N for M38002M4DXXXFP) and attach it to the mask ROM confirmation form.

※ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the X<sub>IN</sub>-X<sub>OUT</sub> oscillator?

- |   |   |
|---|---|
| <input type="checkbox"/> Ceramic resonator    | <input type="checkbox"/> Quartz crystal       |
| <input type="checkbox"/> External clock input | <input type="checkbox"/> Other (            ) |

At what frequency?

f(X<sub>IN</sub>) =  MHz

(2) In which operation mode will you use your microcomputer?

- |  |  |
|--|--|
| <input type="checkbox"/> Single-chip mode    | <input type="checkbox"/> Memory expansion mode |
| <input type="checkbox"/> Microprocessor mode |  |

※ 4. Comments

(2/2)

## 3.6 Mask ROM ordering method

GZZ-SH04-62B<14B0>

Mask ROM number	
-----------------	--

### 740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M38003M6-XXXSP/FP/HP MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked \*.

* Customer	Company name	TEL ( )	Issuance signature	Submitted by	Supervisor
	Date issued	Date:			

\* 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.

Three EPROMs are required for each pattern.

If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name :     M38003M6-XXXSP     M38003M6-XXXFP     M38003M6-XXXHP

Checksum code for entire EPROM 

--	--	--	--

 (hexadecimal notation)

EPROM type (indicate the type used)

<input type="checkbox"/> <b>27256</b>	<input type="checkbox"/> <b>27512</b>												
<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">                     EPROM address                      0000<sub>16</sub>                      000F<sub>16</sub>                      0010<sub>16</sub>                        207F<sub>16</sub>                      2080<sub>16</sub>                        7FFD<sub>16</sub>                      7FFE<sub>16</sub>                      7FFF<sub>16</sub> </td> <td style="width: 50%;"> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Product name ASCII code : 'M38003M6-'</td> <td style="width: 50%;"></td> </tr> <tr> <td colspan="2" style="text-align: center;">data ROM 24446 bytes</td> </tr> </table> </td> </tr> </table>	EPROM address 0000 <sub>16</sub> 000F <sub>16</sub> 0010 <sub>16</sub>  207F <sub>16</sub> 2080 <sub>16</sub>  7FFD <sub>16</sub> 7FFE <sub>16</sub> 7FFF <sub>16</sub>	<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Product name ASCII code : 'M38003M6-'</td> <td style="width: 50%;"></td> </tr> <tr> <td colspan="2" style="text-align: center;">data ROM 24446 bytes</td> </tr> </table>	Product name ASCII code : 'M38003M6-'		data ROM 24446 bytes		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">                     EPROM address                      0000<sub>16</sub>                      000F<sub>16</sub>                      0010<sub>16</sub>                        A07F<sub>16</sub>                      A080<sub>16</sub>                        FFFD<sub>16</sub>                      FFFE<sub>16</sub>                      FFFF<sub>16</sub> </td> <td style="width: 50%;"> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Product name ASCII code : 'M38003M6-'</td> <td style="width: 50%;"></td> </tr> <tr> <td colspan="2" style="text-align: center;">data ROM 24446 bytes</td> </tr> </table> </td> </tr> </table>	EPROM address 0000 <sub>16</sub> 000F <sub>16</sub> 0010 <sub>16</sub>  A07F <sub>16</sub> A080 <sub>16</sub>  FFFD <sub>16</sub> FFFE <sub>16</sub> FFFF <sub>16</sub>	<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Product name ASCII code : 'M38003M6-'</td> <td style="width: 50%;"></td> </tr> <tr> <td colspan="2" style="text-align: center;">data ROM 24446 bytes</td> </tr> </table>	Product name ASCII code : 'M38003M6-'		data ROM 24446 bytes	
EPROM address 0000 <sub>16</sub> 000F <sub>16</sub> 0010 <sub>16</sub>  207F <sub>16</sub> 2080 <sub>16</sub>  7FFD <sub>16</sub> 7FFE <sub>16</sub> 7FFF <sub>16</sub>	<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Product name ASCII code : 'M38003M6-'</td> <td style="width: 50%;"></td> </tr> <tr> <td colspan="2" style="text-align: center;">data ROM 24446 bytes</td> </tr> </table>	Product name ASCII code : 'M38003M6-'		data ROM 24446 bytes									
Product name ASCII code : 'M38003M6-'													
data ROM 24446 bytes													
EPROM address 0000 <sub>16</sub> 000F <sub>16</sub> 0010 <sub>16</sub>  A07F <sub>16</sub> A080 <sub>16</sub>  FFFD <sub>16</sub> FFFE <sub>16</sub> FFFF <sub>16</sub>	<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Product name ASCII code : 'M38003M6-'</td> <td style="width: 50%;"></td> </tr> <tr> <td colspan="2" style="text-align: center;">data ROM 24446 bytes</td> </tr> </table>	Product name ASCII code : 'M38003M6-'		data ROM 24446 bytes									
Product name ASCII code : 'M38003M6-'													
data ROM 24446 bytes													

In the address space of the microcomputer, the internal ROM area is from address A080<sub>16</sub> to FFFD<sub>16</sub>. The reset vector is stored in addresses FFFC<sub>16</sub> and FFFD<sub>16</sub>.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF<sub>16</sub>".
- (2) The ASCII codes of the product name "M38003M6-" must be entered in addresses 0000<sub>16</sub> to 0008<sub>16</sub>. And set the data "FF<sub>16</sub>" in addresses 0009<sub>16</sub> to 000F<sub>16</sub>. The ASCII codes and addresses are listed to the right in hexadecimal notation.

<table border="1" style="width: 100%;"> <tr><td>Address</td><td></td></tr> <tr><td>0000<sub>16</sub></td><td>'M' = 4D<sub>16</sub></td></tr> <tr><td>0001<sub>16</sub></td><td>'3' = 33<sub>16</sub></td></tr> <tr><td>0002<sub>16</sub></td><td>'8' = 38<sub>16</sub></td></tr> <tr><td>0003<sub>16</sub></td><td>'0' = 30<sub>16</sub></td></tr> <tr><td>0004<sub>16</sub></td><td>'0' = 30<sub>16</sub></td></tr> <tr><td>0005<sub>16</sub></td><td>'3' = 33<sub>16</sub></td></tr> <tr><td>0006<sub>16</sub></td><td>'M' = 4D<sub>16</sub></td></tr> <tr><td>0007<sub>16</sub></td><td>'6' = 36<sub>16</sub></td></tr> </table>	Address		0000 <sub>16</sub>	'M' = 4D <sub>16</sub>	0001 <sub>16</sub>	'3' = 33 <sub>16</sub>	0002 <sub>16</sub>	'8' = 38 <sub>16</sub>	0003 <sub>16</sub>	'0' = 30 <sub>16</sub>	0004 <sub>16</sub>	'0' = 30 <sub>16</sub>	0005 <sub>16</sub>	'3' = 33 <sub>16</sub>	0006 <sub>16</sub>	'M' = 4D <sub>16</sub>	0007 <sub>16</sub>	'6' = 36 <sub>16</sub>	<table border="1" style="width: 100%;"> <tr><td>Address</td><td></td></tr> <tr><td>0008<sub>16</sub></td><td>'-' = 2D<sub>16</sub></td></tr> <tr><td>0009<sub>16</sub></td><td>FF<sub>16</sub></td></tr> <tr><td>000A<sub>16</sub></td><td>FF<sub>16</sub></td></tr> <tr><td>000B<sub>16</sub></td><td>FF<sub>16</sub></td></tr> <tr><td>000C<sub>16</sub></td><td>FF<sub>16</sub></td></tr> <tr><td>000D<sub>16</sub></td><td>FF<sub>16</sub></td></tr> <tr><td>000E<sub>16</sub></td><td>FF<sub>16</sub></td></tr> <tr><td>000F<sub>16</sub></td><td>FF<sub>16</sub></td></tr> </table>	Address		0008 <sub>16</sub>	'-' = 2D <sub>16</sub>	0009 <sub>16</sub>	FF <sub>16</sub>	000A <sub>16</sub>	FF <sub>16</sub>	000B <sub>16</sub>	FF <sub>16</sub>	000C <sub>16</sub>	FF <sub>16</sub>	000D <sub>16</sub>	FF <sub>16</sub>	000E <sub>16</sub>	FF <sub>16</sub>	000F <sub>16</sub>	FF <sub>16</sub>
Address																																					
0000 <sub>16</sub>	'M' = 4D <sub>16</sub>																																				
0001 <sub>16</sub>	'3' = 33 <sub>16</sub>																																				
0002 <sub>16</sub>	'8' = 38 <sub>16</sub>																																				
0003 <sub>16</sub>	'0' = 30 <sub>16</sub>																																				
0004 <sub>16</sub>	'0' = 30 <sub>16</sub>																																				
0005 <sub>16</sub>	'3' = 33 <sub>16</sub>																																				
0006 <sub>16</sub>	'M' = 4D <sub>16</sub>																																				
0007 <sub>16</sub>	'6' = 36 <sub>16</sub>																																				
Address																																					
0008 <sub>16</sub>	'-' = 2D <sub>16</sub>																																				
0009 <sub>16</sub>	FF <sub>16</sub>																																				
000A <sub>16</sub>	FF <sub>16</sub>																																				
000B <sub>16</sub>	FF <sub>16</sub>																																				
000C <sub>16</sub>	FF <sub>16</sub>																																				
000D <sub>16</sub>	FF <sub>16</sub>																																				
000E <sub>16</sub>	FF <sub>16</sub>																																				
000F <sub>16</sub>	FF <sub>16</sub>																																				



GZZ-SH04-30B<13B0>

Mask ROM number	
-----------------	--

**740 FAMILY MASK ROM CONFIRMATION FORM  
SINGLE-CHIP MICROCOMPUTER M38004M8-XXXSP/FP  
MITSUBISHI ELECTRIC**

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked \*.

* Customer	Company name	TEL (      )	Issuance signature	Submitted by	Supervisor
	Date issued	Date:			

\* 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.

Three EPROMs are required for each pattern.

If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

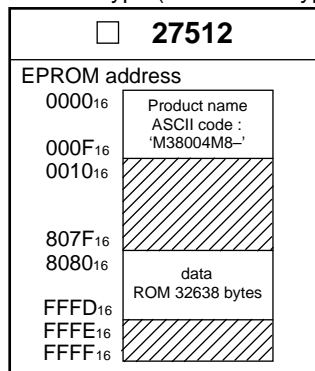
Microcomputer name :     M38004M8-XXXSP             M38004M8-XXXFP

Checksum code for entire EPROM 

--	--	--	--

 (hexadecimal notation)

EPROM type (indicate the type used)



In the address space of the microcomputer, the internal ROM area is from address 8080<sub>16</sub> to FFFD<sub>16</sub>. The reset vector is stored in addresses FFFC<sub>16</sub> and FFFD<sub>16</sub>.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF<sub>16</sub>".
- (2) The ASCII codes of the product name "M38004M8-" must be entered in addresses 0000<sub>16</sub> to 0008<sub>16</sub>. And set the data "FF<sub>16</sub>" in addresses 0009<sub>16</sub> to 000F<sub>16</sub>. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address	'M' = 4D <sub>16</sub>	Address	'-' = 2D <sub>16</sub>
0000 <sub>16</sub>		0008 <sub>16</sub>	FF <sub>16</sub>
0001 <sub>16</sub>	'3' = 33 <sub>16</sub>	0009 <sub>16</sub>	FF <sub>16</sub>
0002 <sub>16</sub>	'8' = 38 <sub>16</sub>	000A <sub>16</sub>	FF <sub>16</sub>
0003 <sub>16</sub>	'0' = 30 <sub>16</sub>	000B <sub>16</sub>	FF <sub>16</sub>
0004 <sub>16</sub>	'0' = 30 <sub>16</sub>	000C <sub>16</sub>	FF <sub>16</sub>
0005 <sub>16</sub>	'4' = 34 <sub>16</sub>	000D <sub>16</sub>	FF <sub>16</sub>
0006 <sub>16</sub>	'M' = 4D <sub>16</sub>	000E <sub>16</sub>	FF <sub>16</sub>
0007 <sub>16</sub>	'8' = 38 <sub>16</sub>	000F <sub>16</sub>	FF <sub>16</sub>

# APPENDIX

## 3.6 Mask ROM ordering method

---

GZZ-SH04-30B<13B0>

Mask ROM number	
-----------------	--

### 740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M38004M8-XXXSP/FP MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program.

EPROM type	27512
The pseudo-command	*= $\Delta$ \$0000 .BYTE $\Delta$ 'M38004M8-'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

※ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (64P4B for M38004M8-XXXSP, 64P6N for M38004M8-XXXFP) and attach it to the mask ROM confirmation form.

※ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the X<sub>IN</sub>-X<sub>OUT</sub> oscillator?

- |   |   |
|---|---|
| <input type="checkbox"/> Ceramic resonator    | <input type="checkbox"/> Quartz crystal       |
| <input type="checkbox"/> External clock input | <input type="checkbox"/> Other (            ) |

At what frequency?

f(X<sub>IN</sub>) =  MHz

(2) In which operation mode will you use your microcomputer?

- |  |  |
|--|--|
| <input type="checkbox"/> Single-chip mode    | <input type="checkbox"/> Memory expansion mode |
| <input type="checkbox"/> Microprocessor mode |  |

※ 4. Comments

(2/2)

GZZ-SH07-23B<33A0>

Mask ROM number	
-----------------	--

**740 FAMILY MASK ROM CONFIRMATION FORM  
SINGLE-CHIP MICROCOMPUTER M38004M8DXXXSP/FP  
MITSUBISHI ELECTRIC**

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked \*.

*	Customer	Company name	TEL (      )	Issuance signature	Submitted by	Supervisor
		Date issued	Date:			

\* 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.

Three EPROMs are required for each pattern.

If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

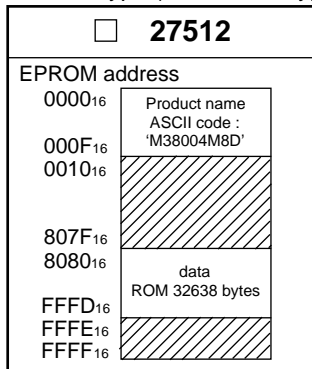
Microcomputer name :     M38004M8DXXXSP             M38004M8DXXXFP

Checksum code for entire EPROM 

--	--	--	--

 (hexadecimal notation)

EPROM type (indicate the type used)



In the address space of the microcomputer, the internal ROM area is from address 8080<sub>16</sub> to FFFD<sub>16</sub>. The reset vector is stored in addresses FFFC<sub>16</sub> and FFFD<sub>16</sub>.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF<sub>16</sub>".
- (2) The ASCII codes of the product name "M38004M8D" must be entered in addresses 0000<sub>16</sub> to 0008<sub>16</sub>. And set the data "FF<sub>16</sub>" in addresses 0009<sub>16</sub> to 000F<sub>16</sub>. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000 <sub>16</sub>	'M' = 4D <sub>16</sub>	0008 <sub>16</sub>	'D' = 44 <sub>16</sub>
0001 <sub>16</sub>	'3' = 33 <sub>16</sub>	0009 <sub>16</sub>	FF <sub>16</sub>
0002 <sub>16</sub>	'8' = 38 <sub>16</sub>	000A <sub>16</sub>	FF <sub>16</sub>
0003 <sub>16</sub>	'0' = 30 <sub>16</sub>	000B <sub>16</sub>	FF <sub>16</sub>
0004 <sub>16</sub>	'0' = 30 <sub>16</sub>	000C <sub>16</sub>	FF <sub>16</sub>
0005 <sub>16</sub>	'4' = 34 <sub>16</sub>	000D <sub>16</sub>	FF <sub>16</sub>
0006 <sub>16</sub>	'M' = 4D <sub>16</sub>	000E <sub>16</sub>	FF <sub>16</sub>
0007 <sub>16</sub>	'8' = 38 <sub>16</sub>	000F <sub>16</sub>	FF <sub>16</sub>

# APPENDIX

## 3.6 Mask ROM ordering method

---

GZZ-SH07-23B<33A0>

Mask ROM number	
-----------------	--

### 740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M38004M8DXXXSP/FP MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program.

EPROM type	27512
The pseudo-command	*= $\Delta$ \$0000 .BYTE $\Delta$ 'M38004M8D'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

※ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (64P4B for M38004M8DXXXSP, 64P6N for M38004M8DXXXFP) and attach it to the mask ROM confirmation form.

※ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the X<sub>IN</sub>-X<sub>OUT</sub> oscillator?

- |   |   |
|---|---|
| <input type="checkbox"/> Ceramic resonator    | <input type="checkbox"/> Quartz crystal       |
| <input type="checkbox"/> External clock input | <input type="checkbox"/> Other (            ) |

At what frequency?

f(X<sub>IN</sub>) =  MHz

(2) In which operation mode will you use your microcomputer?

- |  |  |
|--|--|
| <input type="checkbox"/> Single-chip mode    | <input type="checkbox"/> Memory expansion mode |
| <input type="checkbox"/> Microprocessor mode |  |

※ 4. Comments

(2/2)



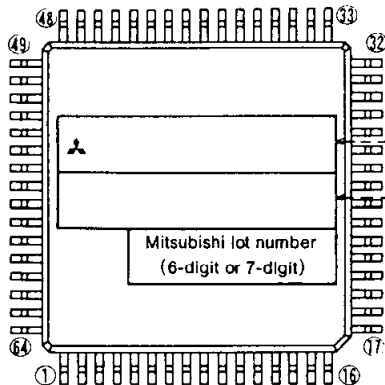
### 3.7 Mark specification form

#### 64P6N (64-PIN QFP) MARK SPECIFICATION FORM

Mitsubishi IC catalog name

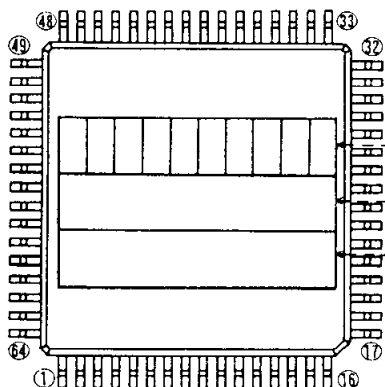
Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

##### A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

##### B. Customer's Parts Number + Mitsubishi Catalog Name



Customer's parts number

Note: The fonts and size of characters are standard Mitsubishi type.

Mitsubishi IC catalog name

Note3: Customer's parts number can be up to 10 characters:

Only 0~9, A~Z, +, -, /, (, ), &, ©, . (period), and , (comma) are usable.

4: If the Mitsubishi logo is not required, check the box below.

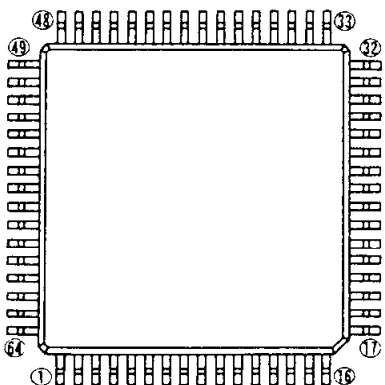
Mitsubishi logo is not required

5: Arrangement of Mitsubishi IC catalog name and Mitsubishi lot number is dependent on number of Mitsubishi IC catalog name and that Mitsubishi logo is required or not.

Note1: The mark field should be written right aligned.

2: The fonts and size of characters are standard Mitsubishi type. (The character size became smaller than A (standard Mitsubishi mark) type)

##### C. Special Mark Required



Note1: If the special mark is to be printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit or 7-digit) and mask ROM number (3-digit) are always marked.

2: If the customer's trade mark logo must be used in the special mark, check the box below. Please submit a clean original of the logo.

For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

Special logo required

The standard Mitsubishi font is used for all characters except for a logo.

# APPENDIX

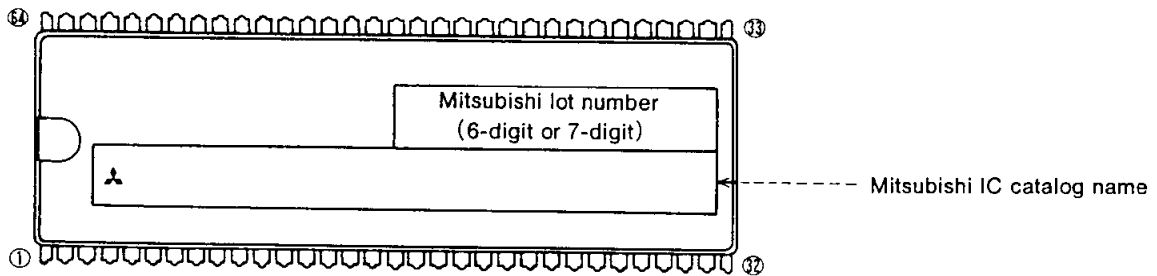
## 3.7 Mark specification form

### 64P4B (64-PIN SHRINK DIP) MARK SPECIFICATION FORM

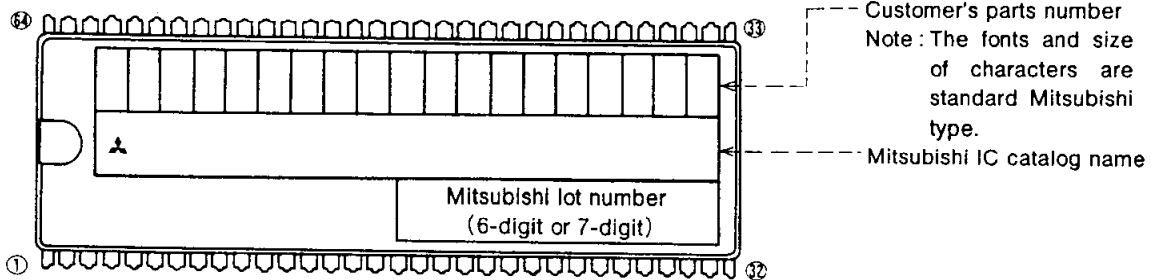
Mitsubishi IC catalog name

Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

#### A. Standard Mitsubishi Mark



#### B. Customer's Parts Number + Mitsubishi Catalog Name



Note 1: The mark field should be written right aligned.

2: The fonts and size of characters are standard Mitsubishi type.

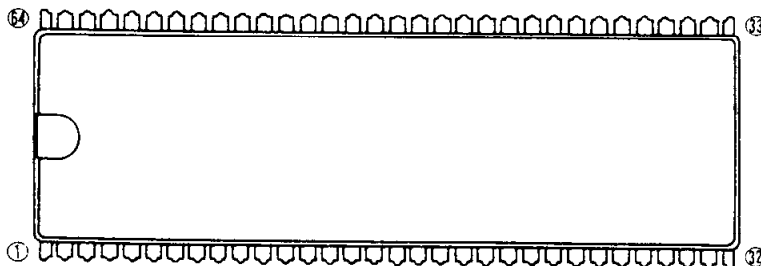
3: Customer's parts number can be up to 19 characters:

Only 0~9, A~Z, +, -, /, (, ), &, ©, . (period), and , (comma) are usable.

4: If the Mitsubishi logo is not required, check the box on the right.

Mitsubishi logo is not required

#### C. Special Mark Required



Note 1: If the special mark is to be printed, indicate the desired layout of the mark in the upper figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit or 7-digit) and mask ROM number (3-digit) are always marked.

2: If the customer's trade mark logo must be used in the special mark, check the box below. Please submit a clean original of the logo.

For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

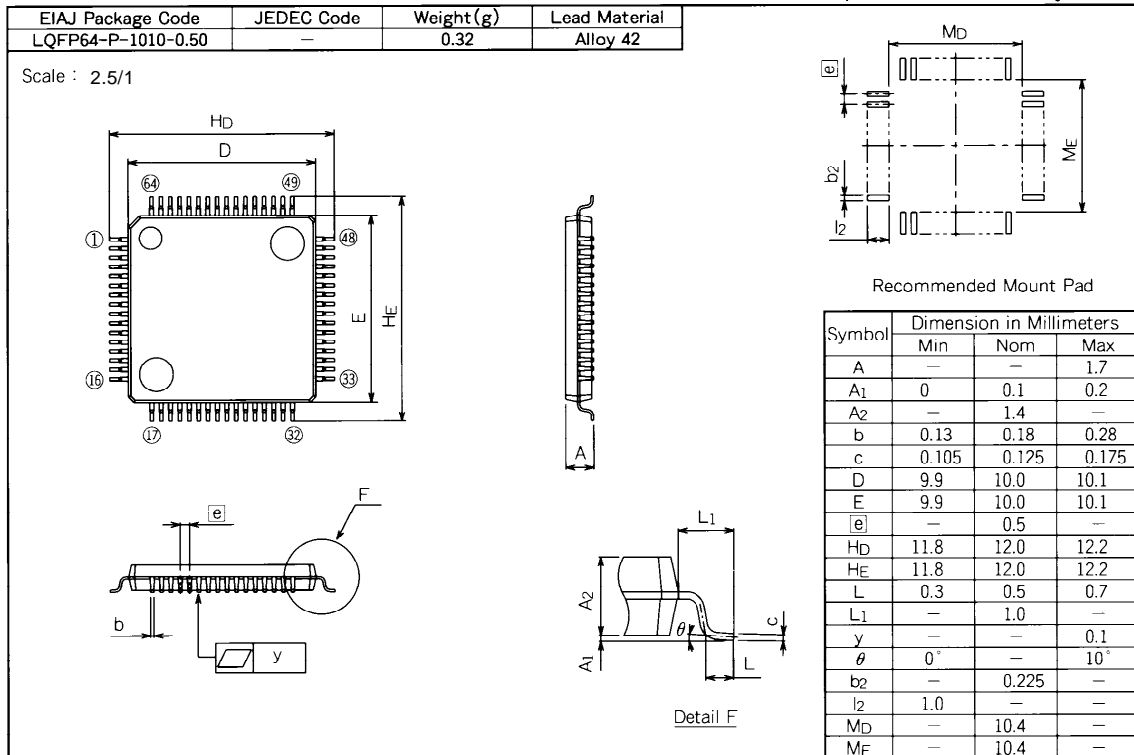
Special logo required

The standard Mitsubishi font is used for all characters except for a logo.

### 3.8 Package outline

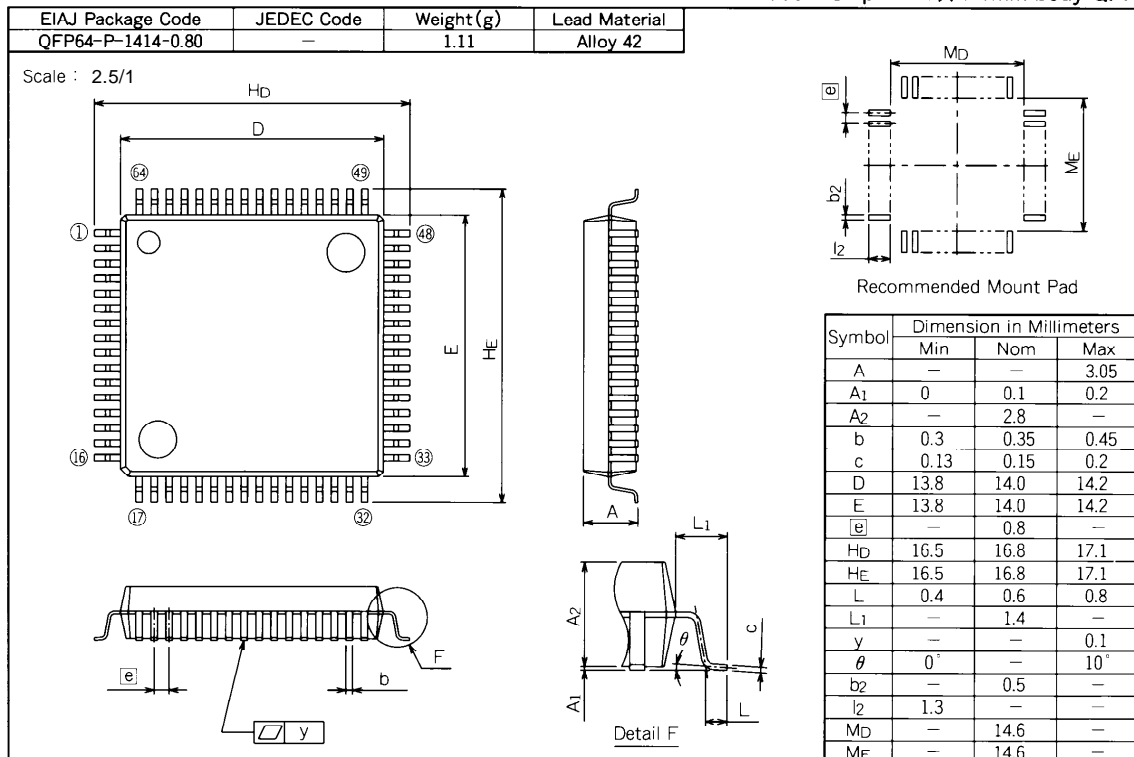
#### 64P6D-A

Plastic 64pin 10x10mm body LQFP



#### 64P6N-A

Plastic 64pin 14x14mm body QFP

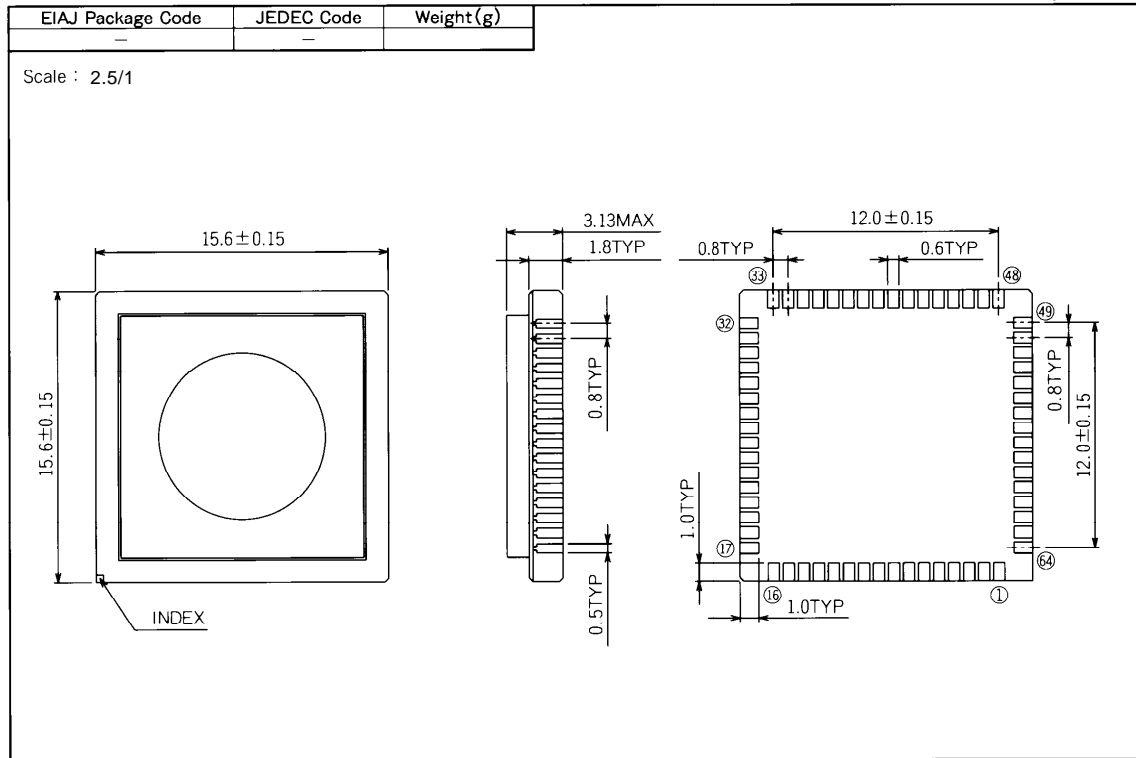


# APPENDIX

## 3.8 Package outline

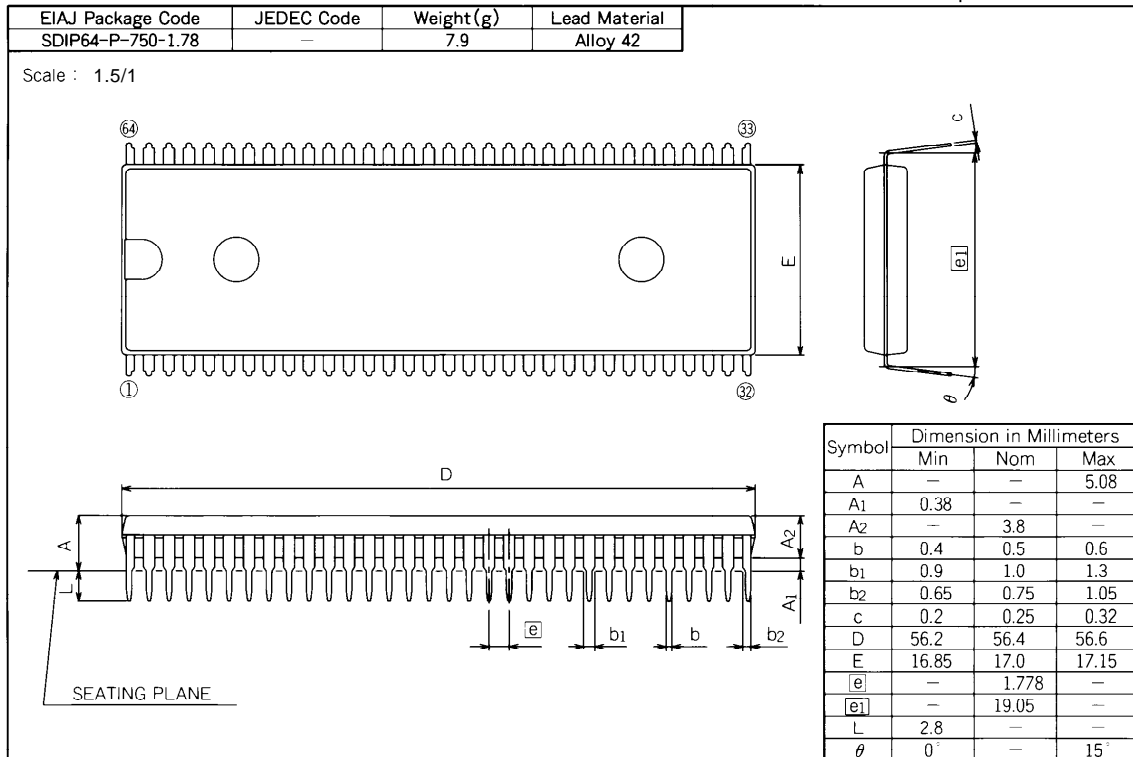
### 64D0

Metal seal 64pin QFN



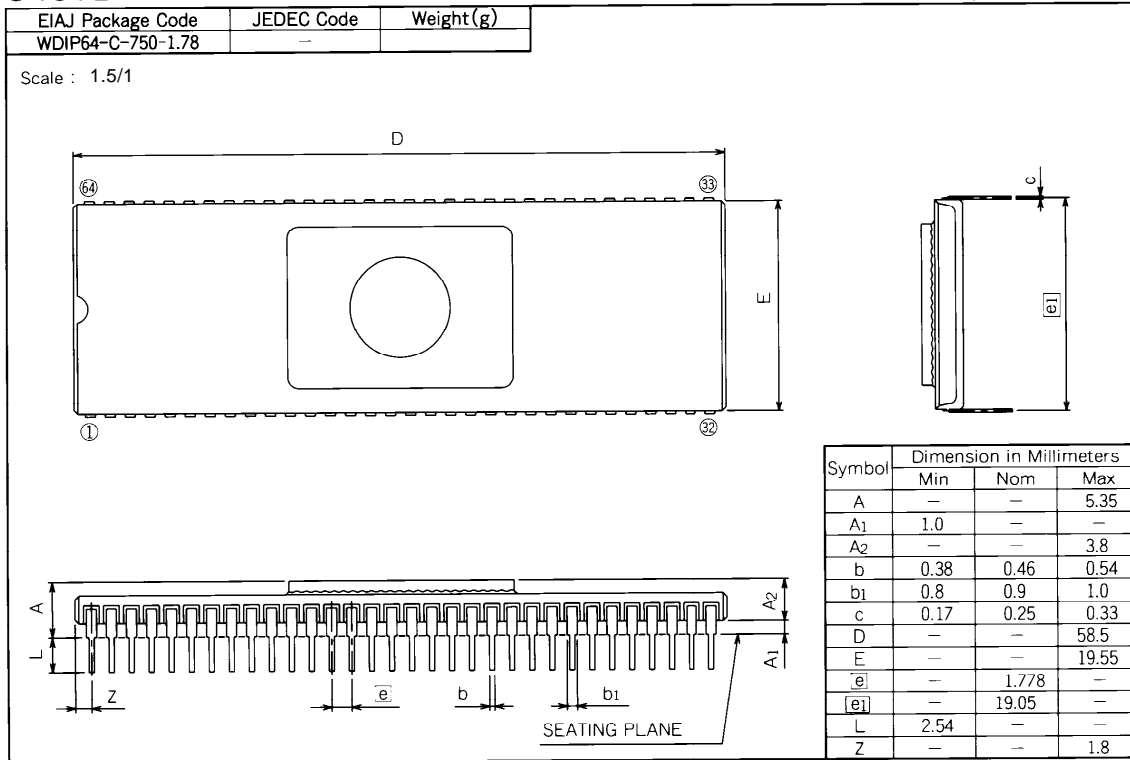
### 64P4B

Plastic 64pin 750mil SDIP



### 64S1B-E

Metal seal 64pin 750mil DIP











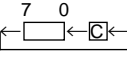
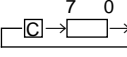







# APPENDIX

## 3.9 Machine instructions

Symbol	Function	Details	Addressing mode																			
			IMP			IMM			A			BIT, A			ZP			BIT, ZP				
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#		
PHA	$M(S) \leftarrow A$ $S \leftarrow S - 1$	Saves the contents of the accumulator in memory at the address indicated by the stack pointer and decrements the contents of stack pointer by 1.	48	3	1																	
PHP	$M(S) \leftarrow PS$ $S \leftarrow S - 1$	Saves the contents of the processor status register in memory at the address indicated by the stack pointer and decrements the contents of the stack pointer by 1.	08	3	1																	
PLA	$S \leftarrow S + 1$ $A \leftarrow M(S)$	Increments the contents of the stack pointer by 1 and restores the accumulator from the memory at the address indicated by the stack pointer.	68	4	1																	
PLP	$S \leftarrow S + 1$ $PS \leftarrow M(S)$	Increments the contents of stack pointer by 1 and restores the processor status register from the memory at the address indicated by the stack pointer.	28	4	1																	
ROL		Shifts the contents of the memory or accumulator to the left by one bit. The high order bit is shifted into the carry flag and the carry flag is shifted into the low order bit.							2A	2	1				26	5	2					
ROR		Shifts the contents of the memory or accumulator to the right by one bit. The low order bit is shifted into the carry flag and the carry flag is shifted into the high order bit.							6A	2	1				66	5	2					
RRF		Rotates the contents of memory to the right by 4 bits.													82	8	2					
RTI	$S \leftarrow S + 1$ $PS \leftarrow M(S)$ $S \leftarrow S + 1$ $PCL \leftarrow M(S)$ $S \leftarrow S + 1$ $PCH \leftarrow M(S)$	Returns from an interrupt routine to the main routine.	40	6	1																	
RTS	$S \leftarrow S + 1$ $PCL \leftarrow M(S)$ $S \leftarrow S + 1$ $PCH \leftarrow M(S)$	Returns from a subroutine to the main routine.	60	6	1																	
SBC (Note 1) (Note 5)	When $T = 0$ $A \leftarrow A - M - \overline{C}$  When $T = 1$ $M(X) \leftarrow M(X) - M - \overline{C}$	Subtracts the contents of memory and complement of carry flag from the contents of accumulator. The results are stored into the accumulator. Subtracts contents of complement of carry flag and contents of the memory indicated by the addressing mode from the memory at the address indicated by index register X. The results are stored into the memory of the address indicated by index register X.							E9	2	2				E5	3	2					
SEB	$Ab \text{ or } Mb \leftarrow 1$	Sets the specified bit in the accumulator or memory to "1".													0B 2i	2	1			0F 2i	5	2
SEC	$C \leftarrow 1$	Sets the contents of the carry flag to "1".	38	2	1																	
SED	$D \leftarrow 1$	Sets the contents of the decimal mode flag to "1".	F8	2	1																	
SEI	$I \leftarrow 1$	Sets the contents of the interrupt disable flag to "1".	78	2	1																	
SET	$T \leftarrow 1$	Sets the contents of the index X mode flag to "1".	32	2	1																	



# APPENDIX

## 3.9 Machine instructions

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
STA	$M \leftarrow A$	Stores the contents of accumulator in memory.													85	4	2			
STP		Stops the oscillator.	42	2	1															
STX	$M \leftarrow X$	Stores the contents of index register X in memory.													86	4	2			
STY	$M \leftarrow Y$	Stores the contents of index register Y in memory.													84	4	2			
TAX	$X \leftarrow A$	Transfers the contents of the accumulator to index register X.	AA	2	1															
TAY	$Y \leftarrow A$	Transfers the contents of the accumulator to index register Y.	A8	2	1															
TST	$M = 0?$	Tests whether the contents of memory are "0" or not.													64	3	2			
TSX	$X \leftarrow S$	Transfers the contents of the stack pointer to index register X.	BA	2	1															
TXA	$A \leftarrow X$	Transfers the contents of index register X to the accumulator.	8A	2	1															
TXS	$S \leftarrow X$	Transfers the contents of index register X to the stack pointer.	9A	2	1															
TYA	$A \leftarrow Y$	Transfers the contents of index register Y to the accumulator.	98	2	1															
WIT		Stops the internal clock.	C2	2	1															

- Notes**
- 1 : The number of cycles "n" is increased by 3 when T is 1.
  - 2 : The number of cycles "n" is increased by 2 when T is 1.
  - 3 : The number of cycles "n" is increased by 1 when T is 1.
  - 4 : The number of cycles "n" is increased by 2 when branching has occurred.
  - 5 : N, V, and Z flags are invalid in decimal operation mode.



# APPENDIX

## 3.10 List of instruction codes

### 3.10 List of instruction codes

D7 – D4	D3 – D0	Hexadecimal notation															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND, X	JSR ZP, IND	BBS 0, A	—	ORA ZP	ASL ZP	BBS 0, ZP	PHP	ORA IMM	ASL A	SEB 0, A	—	ORA ABS	ASL ABS	SEB 0, ZP
0001	1	BPL	ORA IND, Y	CLT	BBC 0, A	—	ORA ZP, X	ASL ZP, X	BBC 0, ZP	CLC	ORA ABS, Y	DEC A	CLB 0, A	—	ORA ABS, X	ASL ABS, X	CLB 0, ZP
0010	2	JSR ABS	AND IND, X	JSR SP	BBS 1, A	BIT ZP	AND ZP	ROL ZP	BBS 1, ZP	PLP	AND IMM	ROL A	SEB 1, A	BIT ABS	AND ABS	ROL ABS	SEB 1, ZP
0011	3	BMI	AND IND, Y	SET	BBC 1, A	—	AND ZP, X	ROL ZP, X	BBC 1, ZP	SEC	AND ABS, Y	INC A	CLB 1, A	LDM ZP	AND ABS, X	ROL ABS, X	CLB 1, ZP
0100	4	RTI	EOR IND, X	STP	BBS 2, A	COM ZP	EOR ZP	LSR ZP	BBS 2, ZP	PHA	EOR IMM	LSR A	SEB 2, A	JMP ABS	EOR ABS	LSR ABS	SEB 2, ZP
0101	5	BVC	EOR IND, Y	—	BBC 2, A	—	EOR ZP, X	LSR ZP, X	BBC 2, ZP	CLI	EOR ABS, Y	—	CLB 2, A	—	EOR ABS, X	LSR ABS, X	CLB 2, ZP
0110	6	RTS	ADC IND, X	MUL ZP, X	BBS 3, A	TST ZP	ADC ZP	ROR ZP	BBS 3, ZP	PLA	ADC IMM	ROR A	SEB 3, A	JMP IND	ADC ABS	ROR ABS	SEB 3, ZP
0111	7	BVS	ADC IND, Y	—	BBC 3, A	—	ADC ZP, X	ROR ZP, X	BBC 3, ZP	SEI	ADC ABS, Y	—	CLB 3, A	—	ADC ABS, X	ROR ABS, X	CLB 3, ZP
1000	8	BRA	STA IND, X	RRF ZP	BBS 4, A	STY ZP	STA ZP	STX ZP	BBS 4, ZP	DEY	—	TXA	SEB 4, A	STY ABS	STA ABS	STX ABS	SEB 4, ZP
1001	9	BCC	STA IND, Y	—	BBC 4, A	STY ZP, X	STA ZP, X	STX ZP, Y	BBC 4, ZP	TYA	STA ABS, Y	TXS	CLB 4, A	—	STA ABS, X	—	CLB 4, ZP
1010	A	LDY IMM	LDA IND, X	LDX IMM	BBS 5, A	LDY ZP	LDA ZP	LDX ZP	BBS 5, ZP	TAY	LDA IMM	TAX	SEB 5, A	LDY ABS	LDA ABS	LDX ABS	SEB 5, ZP
1011	B	BCS	LDA IND, Y	JMP ZP, IND	BBC 5, A	LDY ZP, X	LDA ZP, X	LDX ZP, Y	BBC 5, ZP	CLV	LDA ABS, Y	TSX	CLB 5, A	LDY ABS, X	LDA ABS, X	LDX ABS, Y	CLB 5, ZP
1100	C	CPY IMM	CMP IND, X	WIT	BBS 6, A	CPY ZP	CMP ZP	DEC ZP	BBS 6, ZP	INY	CMP IMM	DEX	SEB 6, A	CPY ABS	CMP ABS	DEC ABS	SEB 6, ZP
1101	D	BNE	CMP IND, Y	—	BBC 6, A	—	CMP ZP, X	DEC ZP, X	BBC 6, ZP	CLD	CMP ABS, Y	—	CLB 6, A	—	CMP ABS, X	DEC ABS, X	CLB 6, ZP
1110	E	CPX IMM	SBC IND, X	DIV ZP, X	BBS 7, A	CPX ZP	SBC ZP	INC ZP	BBS 7, ZP	INX	SBC IMM	NOP	SEB 7, A	CPX ABS	SBC ABS	INC ABS	SEB 7, ZP
1111	F	BEQ	SBC IND, Y	—	BBC 7, A	—	SBC ZP, X	INC ZP, X	BBC 7, ZP	SED	SBC ABS, Y	—	CLB 7, A	—	SBC ABS, X	INC ABS, X	CLB 7, ZP

- 3-byte instruction
- 2-byte instruction
- 1-byte instruction



### 3.11 SFR memory map

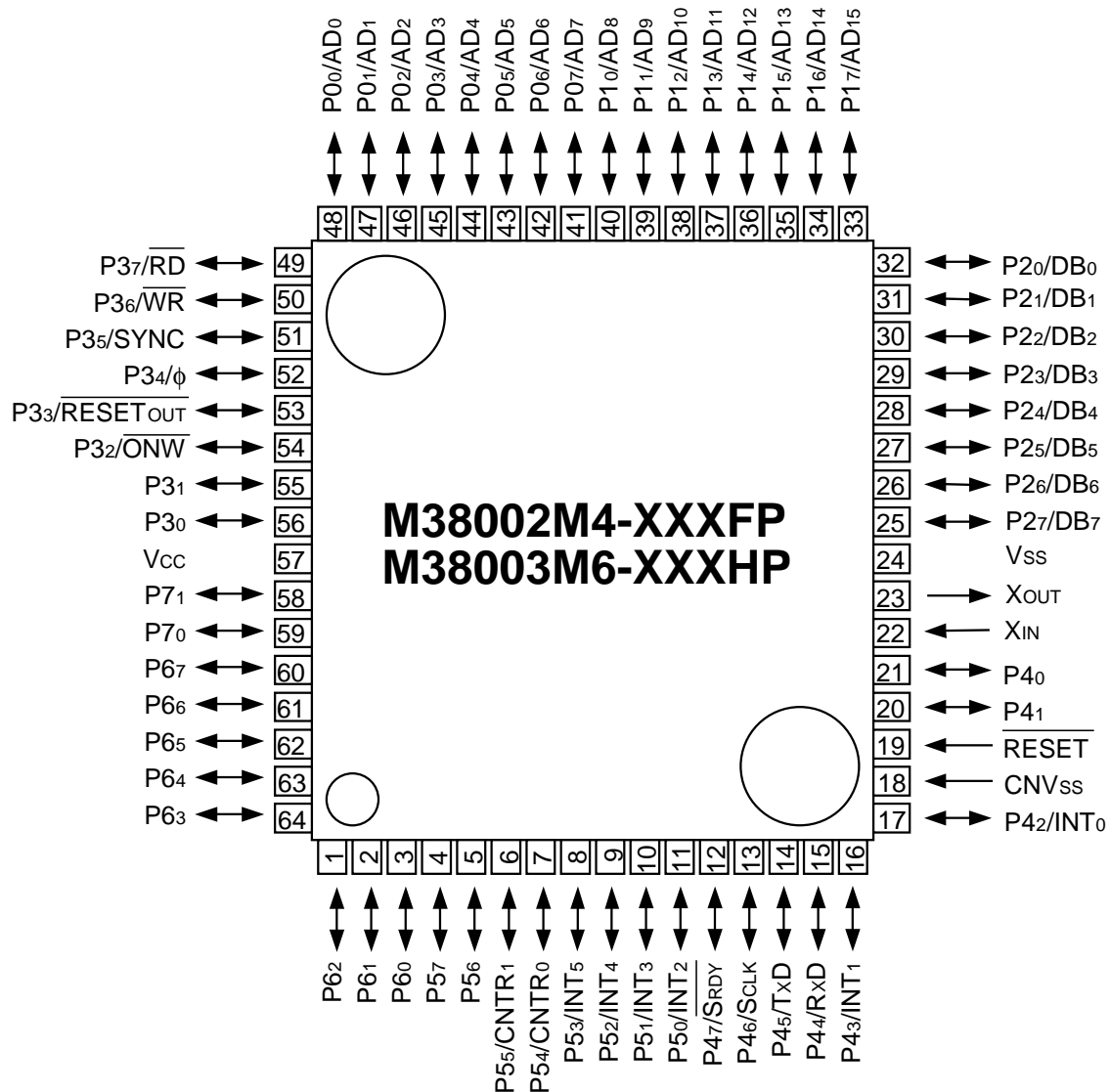
0000 <sub>16</sub>	Port P0 (P0)	0020 <sub>16</sub>	Prescaler 12 (PRE12)
0001 <sub>16</sub>	Port P0 direction register (P0D)	0021 <sub>16</sub>	Timer 1 (T1)
0002 <sub>16</sub>	Port P1 (P1)	0022 <sub>16</sub>	Timer 2 (T2)
0003 <sub>16</sub>	Port P1 direction register (P1D)	0023 <sub>16</sub>	Timer XY mode register (TM)
0004 <sub>16</sub>	Port P2 (P2)	0024 <sub>16</sub>	Prescaler X (PREX)
0005 <sub>16</sub>	Port P2 direction register (P2D)	0025 <sub>16</sub>	Timer X (TX)
0006 <sub>16</sub>	Port P3 (P3)	0026 <sub>16</sub>	Prescaler Y (PREY)
0007 <sub>16</sub>	Port P3 direction register (P3D)	0027 <sub>16</sub>	Timer Y (TY)
0008 <sub>16</sub>	Port P4 (P4)	0028 <sub>16</sub>	
0009 <sub>16</sub>	Port P4 direction register (P4D)	0029 <sub>16</sub>	
000A <sub>16</sub>	Port P5 (P5)	002A <sub>16</sub>	
000B <sub>16</sub>	Port P5 direction register (P5D)	002B <sub>16</sub>	
000C <sub>16</sub>	Port P6 (P6)	002C <sub>16</sub>	
000D <sub>16</sub>	Port P6 direction register (P6D)	002D <sub>16</sub>	
000E <sub>16</sub>	Port P7 (P7)	002E <sub>16</sub>	
000F <sub>16</sub>	Port P7 direction register (P7D)	002F <sub>16</sub>	
0010 <sub>16</sub>		0030 <sub>16</sub>	
0011 <sub>16</sub>		0031 <sub>16</sub>	
0012 <sub>16</sub>		0032 <sub>16</sub>	
0013 <sub>16</sub>		0033 <sub>16</sub>	
0014 <sub>16</sub>		0034 <sub>16</sub>	
0015 <sub>16</sub>		0035 <sub>16</sub>	
0016 <sub>16</sub>		0036 <sub>16</sub>	
0017 <sub>16</sub>		0037 <sub>16</sub>	
0018 <sub>16</sub>	Transmit/Receive buffer register (TB/RB)	0038 <sub>16</sub>	
0019 <sub>16</sub>	Serial I/O status register (SIOSTS)	0039 <sub>16</sub>	
001A <sub>16</sub>	Serial I/O control register (SIOCON)	003A <sub>16</sub>	Interrupt edge selection register (INTEDGE)
001B <sub>16</sub>	UART control register (UARTCON)	003B <sub>16</sub>	CPU mode register (CPUM)
001C <sub>16</sub>	Baud rate generator (BRG)	003C <sub>16</sub>	Interrupt request register 1 (IREQ1)
001D <sub>16</sub>		003D <sub>16</sub>	Interrupt request register 2 (IREQ2)
001E <sub>16</sub>		003E <sub>16</sub>	Interrupt control register 1 (ICON1)
001F <sub>16</sub>		003F <sub>16</sub>	Interrupt control register 2 (ICON2)

# APPENDIX

## 3.12 Pin configuration

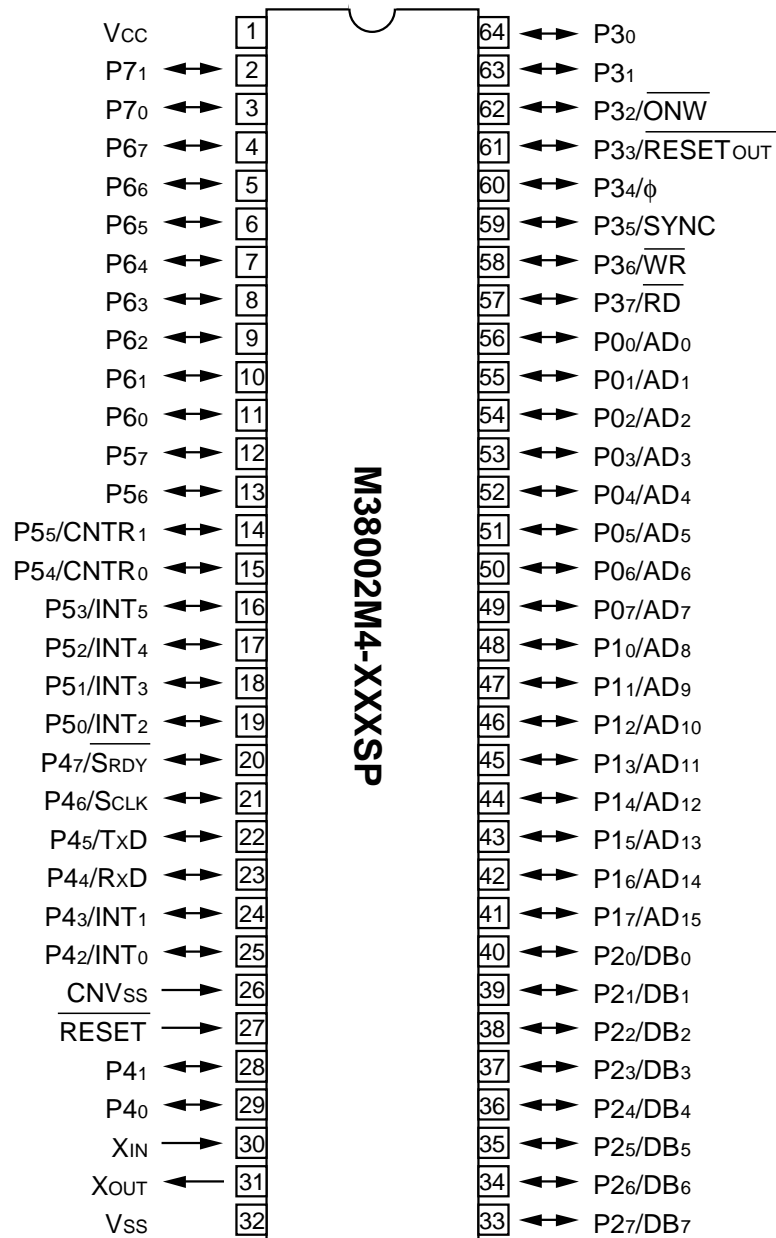
### 3.12 Pin configuration

#### PIN CONFIGURATION (TOP VIEW)



**Package type : 64P6N-A/64P6D-A**  
**64-pin plastic-molded QFP**

### PIN CONFIGURATION (TOP VIEW)



**Package type : 64P4B**  
**64-pin shrink plastic-molded DIP**

**MITSUBISHI SEMICONDUCTORS  
USER'S MANUAL  
3800Group**

---

Mar. First Edition 1996

Edited by  
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by  
Mitsubishi Electric Corp., Semiconductor Marketing Division

---

This book, or parts thereof, may not be reproduced in any form without permission of Mitsubishi Electric Corporation.

©1996 **MITSUBISHI ELECTRIC CORPORATION**

# User's Manual

## 3800 Group



**MITSUBISHI ELECTRIC CORPORATION**

HEAD OFFICE: MITSUBISHI DENKI BLDG., MARUNOUCHI, TOKYO 100. TELEX: J24532 CABLE: MELCO TOKYO