

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

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

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

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

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

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

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
  - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
  - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

To all our customers

---

## **Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.**

---

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.  
Customer Support Dept.  
April 1, 2003

# 3807 Group

User's Manual

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

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 3807 Group.

After reading this manual, the user should have a thorough knowledge of the functions and features of the 3807 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

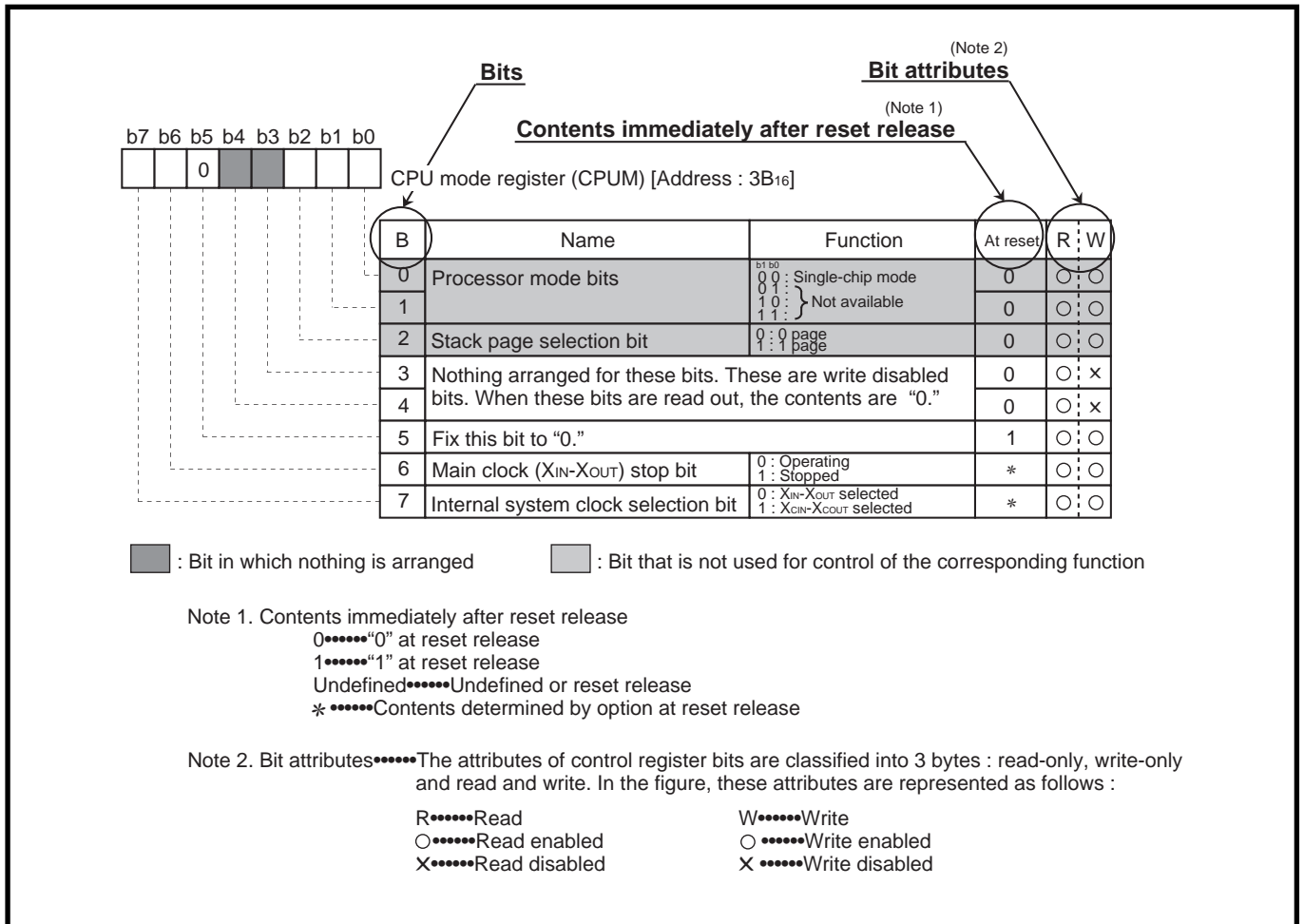
This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of related registers.

### ● 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

3807 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 1996

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 circuits	
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	—	32 K	—	—	—	—	32K	—	—	24 K	—	48K <small>(Note 2)</small>	16K
	RAM	384	384	512	640	384	384	384	64 0	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>APPLICATION</b> .....	<b>1-2</b>
<b>PIN CONFIGURATION</b> .....	<b>1-2</b>
<b>FUNCTIONAL BLOCK</b> .....	<b>1-3</b>
<b>PIN DESCRIPTION</b> .....	<b>1-4</b>
<b>PART NUMBERING</b> .....	<b>1-6</b>
<b>GROUP EXPANSION</b> .....	<b>1-7</b>
<b>FUNCTIONAL DESCRIPTION</b> .....	<b>1-8</b>
Central Processing Unit (CPU) .....	1-8
Memory .....	1-12
I/O Ports .....	1-14
Interrupts .....	1-20
Timers .....	1-23
Serial I/O .....	1-41
A-D Converter .....	1-47
D-A Converter .....	1-48
Analog Comparator .....	1-49
Watchdog Timer .....	1-50
Clock output function .....	1-51
Reset Circuit .....	1-52
Clock Generating Circuit .....	1-54
Processor Mode .....	1-57
<b>NOTES ON PROGRAMMING</b> .....	<b>1-59</b>
Processor Status Register .....	1-59
Interrupts .....	1-59
Decimal Calculations .....	1-59
Timers .....	1-59
Multiplication and Division Instructions .....	1-59
Ports .....	1-59
Serial I/O .....	1-59
A-D Converter .....	1-59
D-A Converter .....	1-59
Instruction Execution Time .....	1-59
<b>NOTES ON USAGE</b> .....	<b>1-60</b>
Handling of Source Pins .....	1-60

## Table of contents

---

P34 clock output function .....	1-60
Timer X and timer Y .....	1-60
EPROM version/One Time PROM version .....	1-60
<b>DATA REQUIRED FOR MASK ORDERS .....</b>	<b>1-61</b>
<b>ROM PROGRAMMING METHOD .....</b>	<b>1-61</b>
<b>FUNCTIONAL DESCRIPTION SUPPLEMENT .....</b>	<b>1-62</b>
Interrupt .....	1-62
Timing After Interrupt .....	1-63
A-D Converter .....	1-64

## 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 .....	2-5
<b>2.2 Timer .....</b>	<b>2-6</b>
2.2.1 Memory map of timer .....	2-6
2.2.2 Related registers .....	2-7
2.2.3 Timer application examples .....	2-14
<b>2.3 Serial I/O .....</b>	<b>2-29</b>
2.3.1 Memory map of serial I/O .....	2-29
2.3.2 Related registers .....	2-30
2.3.3 Serial I/O connection examples .....	2-37
2.3.4 Setting of serial I/O transfer data format .....	2-39
2.3.5 Serial I/O application examples .....	2-40
<b>2.4 Real time output port .....</b>	<b>2-62</b>
2.4.1 Memory map of real time output port .....	2-62
2.4.2 Related registers .....	2-62
2.4.3 Real time output port application examples .....	2-67
<b>2.5 A-D converter .....</b>	<b>2-80</b>
2.5.1 Memory map of A-D conversion .....	2-80
2.5.2 Related registers .....	2-81
2.5.3 A-D conversion application example .....	2-83
<b>2.6 Reset .....</b>	<b>2-87</b>
2.6.1 Connection example of reset IC .....	2-87
<b>2.7 Application circuit example .....</b>	<b>2-88</b>

---

**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 .....	3-3
3.1.3 Electrical characteristics .....	3-5
3.1.4 A-D converter characteristics .....	3-7
3.1.5 D-A converter characteristics .....	3-7
3.1.6 Analog comparator characteristics .....	3-7
3.1.7 Timing requirements .....	3-8
3.1.8 Switching characteristics .....	3-9
3.1.9 Timing requirements in memory expansion and microprocessor mode .....	3-10
3.1.10 Switching characteristics in memory expansion and microprocessor mode .....	3-10
<b>3.2 Standard characteristics</b> .....	<b>3-17</b>
3.2.1 Power source current characteristic examples .....	3-17
3.2.2 Port standard characteristic examples .....	3-18
3.2.3 Input current standard characteristic examples .....	3-21
3.2.4 A-D conversion standard characteristics .....	3-22
3.2.5 D-A conversion standard characteristics .....	3-23
<b>3.3 Notes on use</b> .....	<b>3-24</b>
3.3.1 Notes on interrupts .....	3-24
3.3.2 Notes on the serial I/O1 .....	3-24
3.3.3 Notes on the A-D converter .....	3-25
3.3.4 Notes on the $\overline{\text{RESET}}$ pin .....	3-26
3.3.5 Notes on input and output pins .....	3-26
3.3.6 Notes on memory expansion mode and microprocessor mode .....	3-27
3.3.7 Notes on built-in PROM .....	3-28
<b>3.4 Countermeasures against noise</b> .....	<b>3-29</b>
3.4.1 Shortest wiring length .....	3-29
3.4.2 Connection of a bypass capacitor across the Vss line and the Vcc line .....	3-30
3.4.3 Wiring to analog input pins .....	3-31
3.4.4 Consideration for oscillator .....	3-31
3.4.5 Setup for I/O ports .....	3-32
3.4.6 Providing of watchdog timer function by software .....	3-32
<b>3.5 List of registers</b> .....	<b>3-34</b>
<b>3.6 Mask ROM ordering method</b> .....	<b>3-56</b>
<b>3.7 Mark specification form</b> .....	<b>3-58</b>
<b>3.8 Package outline</b> .....	<b>3-59</b>
<b>3.9 Machine Instructions</b> .....	<b>3-60</b>
<b>3.10 List of instruction codes</b> .....	<b>3-70</b>
<b>3.11 SFR memory map</b> .....	<b>3-71</b>
<b>3.12 Pin configuration</b> .....	<b>3-72</b>

# List of figures

## CHAPTER 1 HARDWARE

Fig. 1 Pin configuration of M38073M4-XXXXFP .....	1-2
Fig. 2 Functional block diagram .....	1-3
Fig. 3 Part numbering .....	1-6
Fig. 4 Memory expansion plan .....	1-7
Fig. 5 740 Family CPU register structure .....	1-8
Fig. 6 Register push and pop at interrupt generation and subroutine call.....	1-9
Fig. 7 Structure of CPU mode register.....	1-11
Fig. 8 Memory map diagram .....	1-12
Fig. 9 Memory map of special function register (SFR) .....	1-13
Fig. 10 Structure of Port P2P3 control register.....	1-14
Fig. 11 Structure of Pull-up control register .....	1-14
Fig. 12 Port block diagram (1) .....	1-17
Fig. 13 Port block diagram (2) .....	1-18
Fig. 14 Port block diagram (3) .....	1-19
Fig. 15 Interrupt control .....	1-22
Fig. 16 Structure of interrupt-related registers .....	1-22
Fig. 17 Block diagram of Timer X and Timer Y.....	1-27
Fig. 18 Structure of Timer X mode register, Timer Y mode register, and Timer XY control register.....	1-28
Fig. 19 Timing chart of Timer • Event counter mode.....	1-29
Fig. 20 Timing chart of Pulse output mode .....	1-29
Fig. 21 Timing chart of Pulse period measurement mode.....	1-30
Fig. 22 Timing chart of Pulse width measurement mode .....	1-30
Fig. 23 Timing chart of Programmable waveform generating mode.....	1-31
Fig. 24 Timing chart of Programmable one-shot generating mode .....	1-31
Fig. 25 Timing chart of PWM mode.....	1-32
Fig. 26 Structure of Timer 123 mode register .....	1-33
Fig. 27 Block diagram of Timer .....	1-33
Fig. 28 Block diagram of Real time output port.....	1-36
Fig. 29 Structure of Real time output port related register (1) .....	1-37
Fig. 30 Structure of Real time output port related register (2) .....	1-38
Fig. 31 8 repeated load mode operation .....	1-39
Fig. 32 6 repeated load mode operation .....	1-39
Fig. 33 5 repeated load mode operation .....	1-40
Fig. 34 One-shot pulse generating mode operation .....	1-40
Fig. 35 Block diagram of clock synchronous serial I/O1.....	1-41
Fig. 36 Operation of clock synchronous serial I/O1 function .....	1-41
Fig. 37 Block diagram of UART serial I/O1 .....	1-42
Fig. 38 Operation of UART serial I/O1 function.....	1-42
Fig. 39 Structure of Serial I/O1 related register.....	1-43
Fig. 40 Structure of Serial I/O2 control register 1, 2 .....	1-44
Fig. 41 Block diagram of Serial I/O2 .....	1-45
Fig. 42 Timing of Serial I/O2 .....	1-45
Fig. 43 SCMP2 output operation .....	1-46
Fig. 44 Structure of A-D control register.....	1-47
Fig. 45 Block diagram of A-D converter .....	1-47

## List of figures

---

Fig. 46 Structure of D-A control register .....	1-48
Fig. 47 Block diagram of D-A converter .....	1-48
Fig. 48 Equivalent connection circuit of D-A converter .....	1-48
Fig. 49 Block diagram of Analog comparator .....	1-49
Fig. 50 Block diagram of Watchdog timer .....	1-50
Fig. 51 Structure of Watchdog timer control register .....	1-50
Fig. 52 Structure of Port P2P3 control register.....	1-51
Fig. 53 Block diagram of Clock output function .....	1-51
Fig. 54 Reset circuit example .....	1-52
Fig. 55 Reset sequence .....	1-52
Fig. 56 Internal status at reset .....	1-53
Fig. 57 Ceramic resonator circuit .....	1-54
Fig. 58 External clock input circuit .....	1-54
Fig. 59 System clock generating circuit block diagram (Single-chip mode) .....	1-55
Fig. 60 State transitions of system clock .....	1-56
Fig. 61 Memory maps in various processor modes .....	1-57
Fig. 62 Structure of CPU mode register .....	1-57
Fig. 63 $\overline{ONW}$ function timing .....	1-58
Fig. 64 Programming and testing of One Time PROM version .....	1-61
Fig. 65 Timing chart after an interrupt occurs .....	1-63
Fig. 66 Time up to execution of the interrupt processing routine .....	1-63
Fig. 67 A-D conversion equivalent circuit.....	1-65
Fig. 68 A-D conversion timing chart .....	1-65

## 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, 7, 8) .....	2-3
Fig. 2.1.3 Structure of Port Pi direction register (i=0, 1, 2, 3, 4, 5, 7, 8) .....	2-3
Fig. 2.1.4 Structure of Port P6 .....	2-4
Fig. 2.1.5 Structure of Port P6 direction register .....	2-4
Fig. 2.2.1 Memory map of timer related registers.....	2-6
Fig. 2.2.2 Structure of Timer XY control register .....	2-7
Fig. 2.2.3 Structure of Timer X Low-order, Timer X High-order, Timer Y Low-order, Timer Y High-order ....	2-7
Fig. 2.2.4 Structure of Timer 1, Timer 3 .....	2-8
Fig. 2.2.5 Structure of Timer 2 .....	2-8
Fig. 2.2.6 Structure of Timer X mode register .....	2-9
Fig. 2.2.7 Structure of Timer Y mode register .....	2-9
Fig. 2.2.8 Structure of Timer 123 mode register.....	2-11
Fig. 2.2.9 Structure of Interrupt edge selection register .....	2-11
Fig. 2.2.10 Structure of Interrupt request register 1 .....	2-12
Fig. 2.2.11 Structure of Interrupt request register 2 .....	2-12
Fig. 2.2.12 Structure of Interrupt control register 1 .....	2-13
Fig. 2.2.13 Structure of Interrupt control register 2 .....	2-13
Fig. 2.2.14 Connection of timers and setting of division ratios [Clock function] .....	2-15
Fig. 2.2.15 Setting of related registers [Clock function] .....	2-16
Fig. 2.2.16 Control procedure [Clock function] .....	2-17
Fig. 2.2.17 Example of a peripheral circuit .....	2-18
Fig. 2.2.18 Connection of the timer and setting of the division ratio [Piezoelectric buzzer output] .....	2-18

Fig. 2.2.19	Setting of related registers [Piezoelectric buzzer output].....	2-19
Fig. 2.2.20	Control procedure [Piezoelectric buzzer output].....	2-20
Fig. 2.2.21	A method for judging if input pulse exists .....	2-21
Fig. 2.2.22	Setting of related registers (1) [Measurement of frequency] .....	2-22
Fig. 2.2.23	Setting of related registers (2) [Measurement of frequency] .....	2-23
Fig. 2.2.24	Control procedure [Measurement of frequency].....	2-24
Fig. 2.2.25	Connection of the timer and setting of the division ratio [Measurement of pulse width] .....	2-25
Fig. 2.2.26	Setting of related registers [Measurement of pulse width] .....	2-26
Fig. 2.2.27	Control procedure (1) [Measurement of pulse width] .....	2-27
Fig. 2.2.28	Control procedure (2) [Measurement of pulse width] .....	2-28
Fig. 2.3.1	Memory map of serial I/O related registers .....	2-29
Fig. 2.3.2	Structure of Transmit/Receive buffer register .....	2-30
Fig. 2.3.3	Structure of Serial I/O1 status register .....	2-30
Fig. 2.3.4	Structure of Serial I/O1 control register.....	2-31
Fig. 2.3.5	Structure of UART control register .....	2-31
Fig. 2.3.6	Structure of Baud rate generator.....	2-32
Fig. 2.3.7	Structure of Serial I/O2 control register 1 .....	2-32
Fig. 2.3.8	Structure of Serial I/O2 control register 2 .....	2-33
Fig. 2.3.9	Structure of Serial I/O2 register.....	2-33
Fig. 2.3.10	Structure of Interrupt edge selection register .....	2-34
Fig. 2.3.11	Structure of Interrupt request register 1 .....	2-35
Fig. 2.3.12	Structure of Interrupt request register 2 .....	2-35
Fig. 2.3.13	Structure of Interrupt control register 1 .....	2-36
Fig. 2.3.14	Structure of Interrupt control register 2 .....	2-36
Fig. 2.3.15	Serial I/O connection examples (1).....	2-37
Fig. 2.3.16	Serial I/O connection examples (2).....	2-38
Fig. 2.3.17	Setting of Serial I/O transfer data format.....	2-39
Fig. 2.3.18	Connection diagram [Communication using a clock synchronous serial I/O] ..	2-40
Fig. 2.3.19	Timing chart [Communication using a clock synchronous serial I/O].....	2-40
Fig. 2.3.20	Setting of related registers at a transmitting side [Communication using a clock synchronous serial I/O] .....	2-41
Fig. 2.3.21	Setting of related registers at a receiving side [Communication using a clock synchronous serial I/O] .....	2-42
Fig. 2.3.22	Control procedure at a transmitting side [Communication using a clock synchronous serial I/O] .....	2-43
Fig. 2.3.23	Control procedure at a receiving side [Communication using a clock synchronous serial I/O] .....	2-44
Fig. 2.3.24	Connection diagram [Output of serial data] .....	2-45
Fig. 2.3.25	Timing chart [Output of serial data] .....	2-45
Fig. 2.3.26	Setting of serial I/O1 related registers [Output of serial data] .....	2-46
Fig. 2.3.27	Setting of serial I/O1 transmission data [Output of serial data].....	2-47
Fig. 2.3.28	Control procedure of serial I/O1 [Output of serial data] .....	2-48
Fig. 2.3.29	Setting of serial I/O2 related registers [Output of serial data] .....	2-49
Fig. 2.3.30	Setting of serial I/O2 transmission data [Output of serial data].....	2-50
Fig. 2.3.31	Control procedure of serial I/O2 [Output of serial data] .....	2-51
Fig. 2.3.32	Connection diagram [Cyclic transmission or reception of block data between microcomputers]..	2-52
Fig. 2.3.33	Timing chart [Cyclic transmission or reception of block data between microcomputers]..	2-53

## List of figures

---

Fig. 2.3.34 Setting of related registers [Cyclic transmission or reception of block data between microcomputers]..	2-53
Fig. 2.3.35 Control in the master unit .....	2-54
Fig. 2.3.36 Control in the slave unit .....	2-55
Fig. 2.3.37 Connection diagram [Communication using UART] .....	2-56
Fig. 2.3.38 Timing chart [Communication using UART] .....	2-56
Fig. 2.3.39 Setting of related registers at a transmitting side [Communication using UART].....	2-58
Fig. 2.3.40 Setting of related registers at a receiving side [Communication using UART].....	2-59
Fig. 2.3.41 Control procedure at a transmitting side [Communication using UART].....	2-60
Fig. 2.3.42 Control procedure at a receiving side [Communication using UART] .....	2-61
Fig. 2.4.1 Memory map of real time output port related registers.....	2-62
Fig. 2.4.2 Structure of Real time port register .....	2-62
Fig. 2.4.3 Structure of Real time port control register 0.....	2-63
Fig. 2.4.4 Structure of Real time port control register 1 .....	2-64
Fig. 2.4.5 Structure of Real time port control register 2.....	2-65
Fig. 2.4.6 Structure of Real time port control register 3.....	2-66
Fig. 2.4.7 Structure of Timer A Low-order, Timer A High-order, Timer B Low-order, Timer B High-order..	2-66
Fig. 2.4.8 Connection diagram .....	2-67
Fig. 2.4.9 Operation patterns of motor .....	2-67
Fig. 2.4.10 Example of timer table for acceleration and deceleration.....	2-68
Fig. 2.4.11 Example of operation pattern table .....	2-69
Fig. 2.4.12 Example of output data table.....	2-69
Fig. 2.4.13 Timing of Real time output.....	2-70
Fig. 2.4.14 Setting method and output timing .....	2-71
Fig. 2.4.15 Setting of related registers (1).....	2-72
Fig. 2.4.16 Setting of related registers (2).....	2-73
Fig. 2.4.17 Setting of related registers (3).....	2-74
Fig. 2.4.18 Setting of related registers (4).....	2-75
Fig. 2.4.19 Control procedure (1) .....	2-76
Fig. 2.4.20 Control procedure (2) .....	2-77
Fig. 2.4.21 Control procedure (3).....	2-78
Fig. 2.4.22 Control procedure (4).....	2-79
Fig. 2.5.1 Memory map of A-D conversion related registers.....	2-80
Fig. 2.5.2 Structure of A-D control register.....	2-81
Fig. 2.5.3 Structure of A-D conversion register.....	2-81
Fig. 2.5.4 Structure of Interrupt request register 2 .....	2-82
Fig. 2.5.5 Structure of Interrupt control register 2 .....	2-82
Fig. 2.5.6 Connection diagram [Read for analog signal using an internal trigger].....	2-83
Fig. 2.5.7 Setting of related registers [Read for analog signal using an internal trigger] .	2-83
Fig. 2.5.8 Control procedure [Read for analog signal using an internal trigger] .....	2-84
Fig. 2.5.9 Connection diagram [Read for analog signal using an external trigger] .....	2-85
Fig. 2.5.10 Setting of related registers [Read for analog signal using an external trigger] .....	2-85
Fig. 2.5.11 Control procedure [Read for analog signal using an external trigger] .....	2-86
Fig. 2.6.1 Example of Poweron reset circuit.....	2-87
Fig. 2.6.2 RAM back-up system .....	2-87
Fig. 2.7.1 Hot water supply system application example.....	2-89
Fig. 2.7.2 CD changer (car audio) application example.....	2-90
Fig. 2.7.3 Hot water washing toilet seat application example .....	2-91

## CHAPTER 3 APPENDIX

Fig. 3.1.1 Circuit for measuring output switching characteristics (1).....	3-11
Fig. 3.1.2 Circuit for measuring output switching characteristics (2).....	3-11
Fig. 3.1.3 Timing diagram (1) (in single-chip mode) .....	3-12
Fig. 3.1.4 Timing diagram (2) (in memory expansion mode and microprocessor mode) ..	3-13
Fig. 3.1.5 Timing diagram (3) (in memory expansion mode and microprocessor mode) ..	3-14
Fig. 3.1.6 Timing diagram (4) (in memory expansion mode and microprocessor mode) ..	3-15
Fig. 3.1.7 Timing diagram (5) (in memory expansion mode and microprocessor mode) ..	3-16
Fig. 3.2.1 Power source current characteristic example .....	3-17
Fig. 3.2.2 Power source current characteristic example (in wait mode).....	3-17
Fig. 3.2.3 Standard characteristic example of CMOS output port at P-channel drive(1) ..	3-18
Fig. 3.2.4 Standard characteristic example of CMOS output port at P-channel drive(2) ..	3-18
Fig. 3.2.5 Standard characteristic example of CMOS output port at N-channel drive(1) ..	3-19
Fig. 3.2.6 Standard characteristic example of CMOS output port at N-channel drive(2) ..	3-19
Fig. 3.2.7 Standard characteristic example of CMOS output port at N-channel drive(3) ..	3-20
Fig. 3.2.8 Standard characteristic example of CMOS output port at N-channel drive(4) ..	3-20
Fig. 3.2.9 Standard characteristic example of input current at connecting pull-up transistor (1) .....	3-21
Fig. 3.2.10 Standard characteristic example of input current at connecting pull-up transistor (2) .....	3-21
Fig. 3.2.11 A-D conversion standard characteristics .....	3-22
Fig. 3.2.12 D-A conversion standard characteristics .....	3-23
Fig. 3.3.1 Structure of interrupt control register 2 .....	3-24
Fig. 3.4.1 Wiring for the $\overline{\text{RESET}}$ pin.....	3-29
Fig. 3.4.2 Wiring for clock I/O pins .....	3-30
Fig. 3.4.3 Wiring for the $V_{PP}$ pin of the One Time PROM and the EPROM version .....	3-30
Fig. 3.4.4 Bypass capacitor across the $V_{SS}$ line and the $V_{CC}$ line .....	3-30
Fig. 3.4.5 Analog signal line and a resistor and a capacitor .....	3-31
Fig. 3.4.6 Wiring for a large current signal line .....	3-31
Fig. 3.4.7 Wiring to a signal line where potential levels change frequently.....	3-31
Fig. 3.4.8 Setup for I/O ports .....	3-32
Fig. 3.4.9 Watchdog timer by software.....	3-32
Fig. 3.5.1 Structure of Port Pi (i=0, 1, 2, 3, 4, 5, 7, 8).....	3-34
Fig. 3.5.2 Structure of Port Pi direction register (i=0, 1, 2, 3, 4, 5, 7, 8) .....	3-34
Fig. 3.5.3 Structure of Port P6 .....	3-35
Fig. 3.5.4 Structure of Port P6 direction register .....	3-35
Fig. 3.5.5 Structure of Timer XY control register .....	3-36
Fig. 3.5.6 Structure of Port P2P3 control register .....	3-36
Fig. 3.5.7 Structure of Pull-up control register .....	3-37
Fig. 3.5.8 Structure of Watchdog timer control register .....	3-37
Fig. 3.5.9 Structure of Transmit/Receive buffer register .....	3-38
Fig. 3.5.10 Structure of Serial I/O1 status register .....	3-38
Fig. 3.5.11 Structure of Serial I/O1 control register .....	3-39
Fig. 3.5.12 Structure of UART control register.....	3-39
Fig. 3.5.13 Structure of Baud rate generator.....	3-40
Fig. 3.5.14 Structure of Serial I/O2 control register 1.....	3-40
Fig. 3.5.15 Structure of Serial I/O2 control register 2.....	3-41
Fig. 3.5.16 Structure of Serial I/O2 register .....	3-41
Fig. 3.5.17 Structure of Timer X Low-order, Timer X High-order, Timer Y Low-order, Timer Y High-order	3-42



## List of figures

---

Fig. 3.5.18 Structure of Timer 1, Timer 3 .....	3-42
Fig. 3.5.19 Structure of Timer 2 .....	3-43
Fig. 3.5.20 Structure of Timer X mode register .....	3-44
Fig. 3.5.21 Structure of Timer Y mode register .....	3-44
Fig. 3.5.22 Structure of Timer 123 mode register .....	3-46
Fig. 3.5.23 Structure of Real time port register .....	3-46
Fig. 3.5.24 Structure of Real time port control register 0 .....	3-47
Fig. 3.5.25 Structure of Real time port control register 1 .....	3-48
Fig. 3.5.26 Structure of Real time port control register 2 .....	3-49
Fig. 3.5.27 Structure of Real time port control register 3 .....	3-50
Fig. 3.5.28 Structure of Timer A Low-order, Timer A High-order, Timer B Low-order, Timer B High-order	3-50
Fig. 3.5.29 Structure of D-A control register .....	3-51
Fig. 3.5.30 Structure of A-D control register .....	3-51
Fig. 3.5.31 Structure of A-D conversion register .....	3-52
Fig. 3.5.32 Structure of D-Ai conversion register (i = 1,2,3,4) .....	3-52
Fig. 3.5.33 Structure of Interrupt edge selection register .....	3-53
Fig. 3.5.34 Structure of CPU mode register .....	3-53
Fig. 3.5.35 Structure of Interrupt request register 1 .....	3-54
Fig. 3.5.35 Structure of Interrupt request register 2 .....	3-54
Fig. 3.5.36 Structure of Interrupt control register 1 .....	3-55
Fig. 3.5.36 Structure of Interrupt control register 2 .....	3-55

# List of tables

## CHAPTER 1 HARDWARE

Table 1 Pin description (1).....	1-4
Table 2 Pin description (2).....	1-5
Table 3 List of supported products .....	1-7
Table 4 Push and pop instructions of accumulator or processor status register .....	1-9
Table 5 Set and clear instructions of each bit of processor status register .....	1-10
Table 6 List of I/O port functions (1).....	1-15
Table 7 List of I/O port functions (2).....	1-16
Table 8 Interrupt vector addresses and priority .....	1-21
Table 9 Port functions in memory expansion mode and microprocessor mode .....	1-57
Table 10 Special programming adapter.....	1-61
Table 11 Interrupt sources, vector addresses and interrupt priority .....	1-62
Table 12 Change of A-D conversion register during A-D conversion .....	1-64

## CHAPTER 2 APPLICATION

Table 2.1.1 Handling of unused pins (in single-chip mode) .....	2-5
Table 2.1.2 Handling of unused pins (in memory expansion mode and microprocessor mode) .....	2-5
Table 2.2.1 Function of CNTR0/CNTR1 edge switch bit .....	2-10
Table 2.3.1 Setting examples of Baud rate generator values and transfer bit rate values .....	2-57

## CHAPTER 3 APPENDIX

Table 3.1.1 Absolute maximum ratings .....	3-2
Table 3.1.2 Recommended operating conditions (1) .....	3-3
Table 3.1.3 Recommended operating conditions (2) .....	3-3
Table 3.1.4 Recommended operating conditions (3) .....	3-4
Table 3.1.5 Electrical characteristics (1) .....	3-5
Table 3.1.6 Electrical characteristics (2) .....	3-6
Table 3.1.7 A-D converter characteristics .....	3-7
Table 3.1.8 D-A converter characteristics .....	3-7
Table 3.1.9 Analog comparator characteristics.....	3-7
Table 3.1.10 Timing requirements (1).....	3-8
Table 3.1.11 Timing requirements (2).....	3-8
Table 3.1.12 Switching characteristics (1).....	3-9
Table 3.1.13 Switching characteristics (2).....	3-9
Table 3.1.14 Timing requirements in memory expansion mode and microprocessor mode .....	3-10
Table 3.1.15 Switching characteristics in memory expansion mode and microprocessor mode .....	3-10
Table 3.3.1 Programming adapter .....	3-28
Table 3.3.2 Setting of programming adapter switch .....	3-28
Table 3.3.3 Setting of PROM programmer address.....	3-28
Table 3.5.1 Function of CNTR0/CNTR1 edge switch bit .....	3-45



# 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 3807 group is a 8-bit microcomputer based on the 740 family core technology.

The 3807 group has two serial I/Os, an A-D converter, a D-A converter, a real time output port function, a watchdog timer, and an analog comparator, which are available for a system controller which controls motors of office equipment and household appliances.

The various microcomputers in the 3807 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 3807 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 to 60 K bytes
  - RAM ..... 384 to 2048 bytes
- Programmable input/output ports ..... 68
- Software pull-up resistors (Ports P0 to P2) ..... 24
- Input ports (Ports P63 and P64) ..... 2
- Interrupts ..... 20 sources, 16 vectors
- Timers X, Y ..... 16-bit X 2
- Timers A, B (for real time output port function) ..... 16-bit X 2
- Timers 1–3 ..... 8-bit X 3

- Serial I/O1 (UART or Clock-synchronized) ..... 8-bit X 1
- Serial I/O2 (Clock-synchronized) ..... 8-bit X 1
- A-D converter ..... 8-bit X 13 channels
- D-A converter ..... 8-bit X 4 channels
- Watchdog timer ..... 16-bit X 1
- Analog comparator ..... 1 channel
- 2 Clock generating circuit
  - Main clock (X<sub>IN</sub>–X<sub>OUT</sub>) ..... Internal feedback resistor
  - Sub-clock (X<sub>CIN</sub>–X<sub>COUT</sub>) ..... Without internal feedback resistor (connect to external ceramic resonator or quartz-crystal oscillator)
- Power source voltage
  - In high-speed mode ..... 4.0 to 5.5 V (at 8 MHz oscillation frequency and high-speed selected)
  - In middle-speed mode ..... 2.7 to 5.5 V (at 8 MHz oscillation frequency and middle-speed selected)
  - In low-speed mode ..... 2.7 to 5.5 V (at 32 kHz oscillation frequency and low-speed selected)
- Power dissipation
  - In high-speed mode ..... 34 m W (at 8 MHz oscillation frequency, at 5 V power source voltage)
  - In low-speed mode ..... 60  $\mu$ W (at 32 kHz oscillation frequency, at 3 V power source voltage)
- Memory expansion ..... possible
- Operating temperature range ..... –20 to 85 °C

### APPLICATION

LBP engine control, PPC, FAX, office equipment, household appliances, consumer electronics, etc.

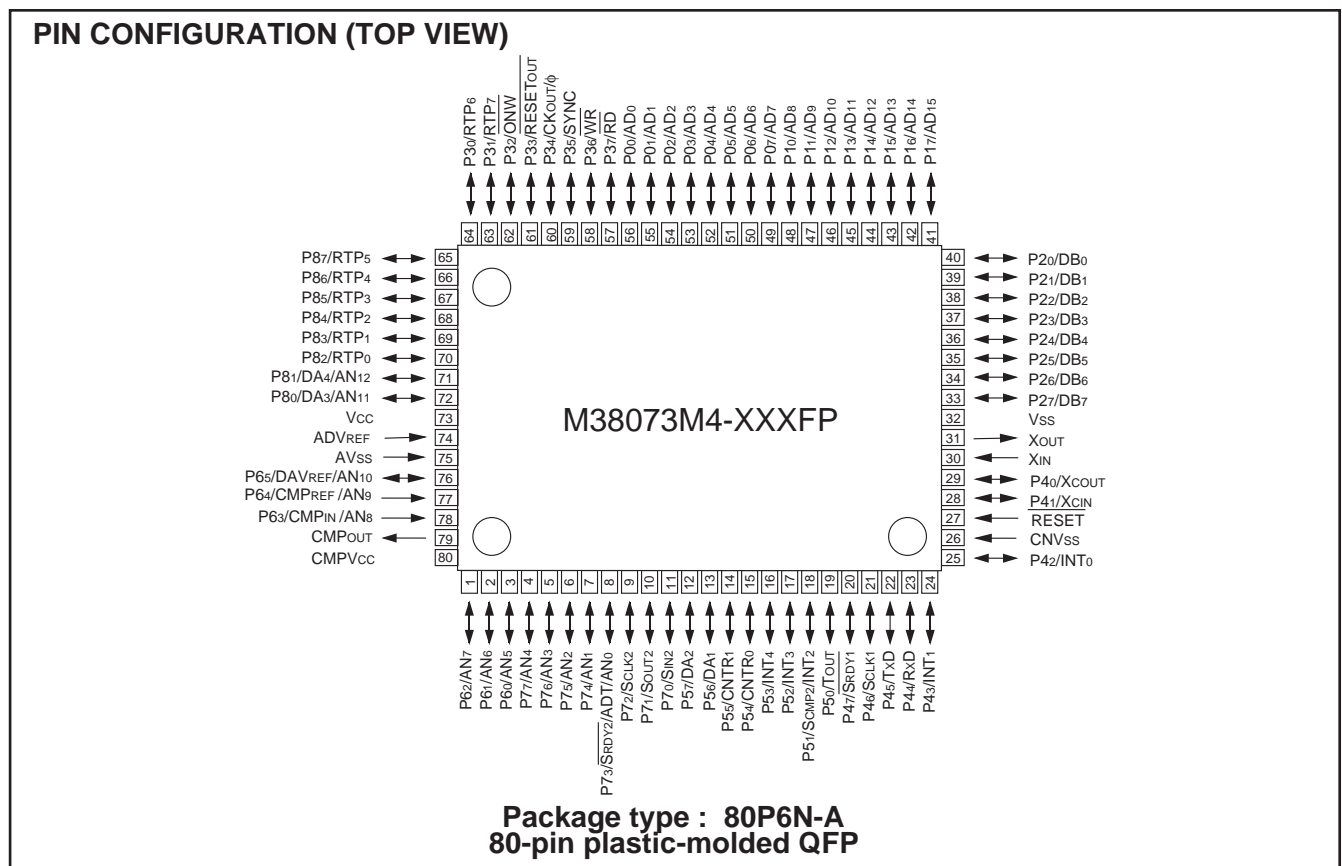


Fig. 1. Pin configuration of M38073M4-XXXXP

### FUNCTIONAL BLOCK

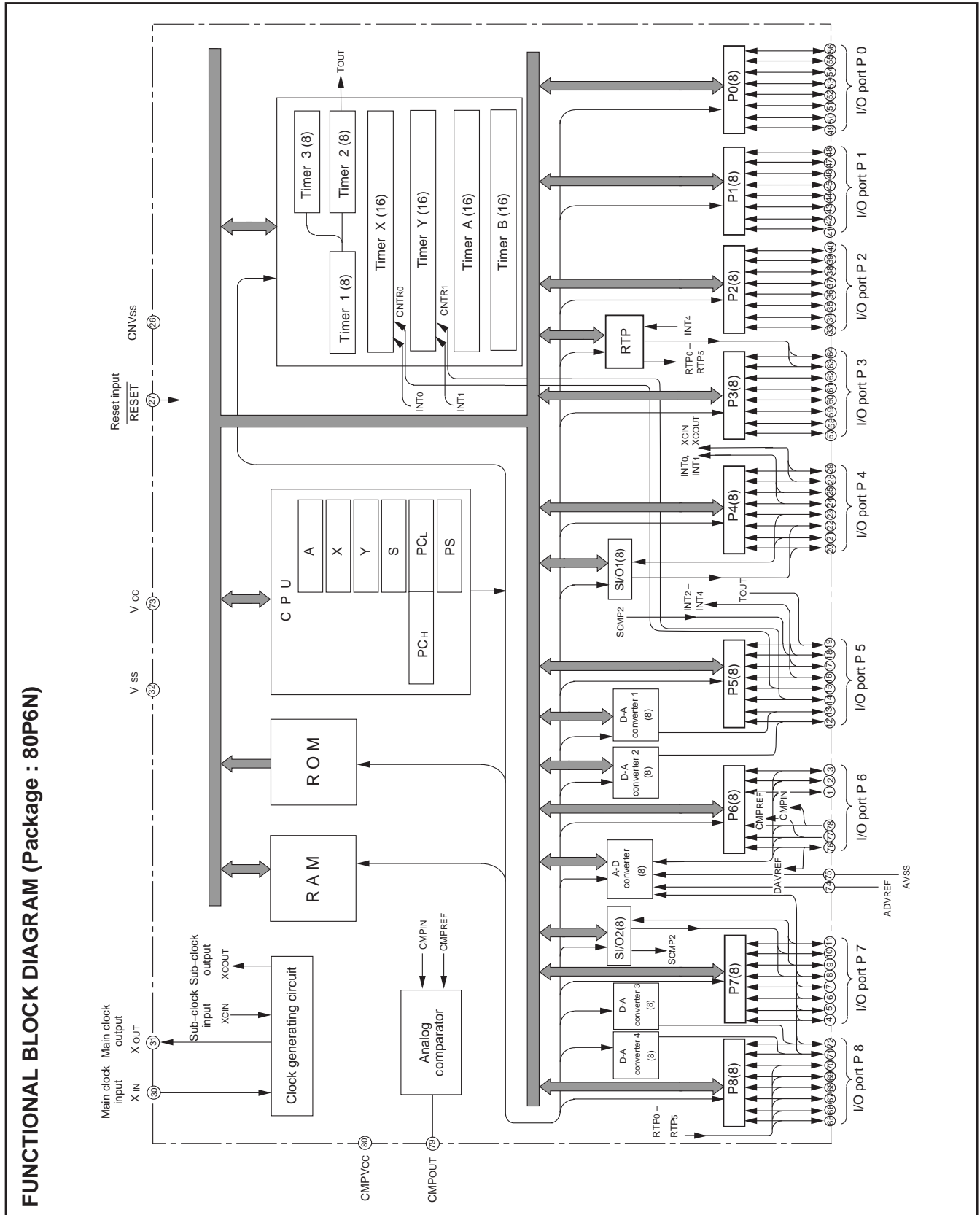


Fig. 2. Functional block diagram

# HARDWARE

## PIN DESCRIPTION

### PIN DESCRIPTION

Table 1. Pin description (1)

Pin	Name	Function	Function except a port function
Vcc, Vss	Power source	<ul style="list-style-type: none"> <li>Apply voltage of 2.7–5.5 V to Vcc, and 0 V to Vss.</li> </ul>	
CMPVcc	Analog comparator power source	<ul style="list-style-type: none"> <li>Power source input pin for an analog comparator</li> </ul>	
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>	
ADVREF	Analog reference voltage	<ul style="list-style-type: none"> <li>Reference voltage input pin for A-D converter.</li> </ul>	
AVss	Analog power source	<ul style="list-style-type: none"> <li>Analog power source input pin for A-D and D-A converter and an analog comparator</li> <li>Connect to Vss.</li> </ul>	
CMPOUT	Analog comparator output	<ul style="list-style-type: none"> <li>Output pin for an analog comparator</li> </ul>	
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> </ul>	
XOUT	Clock output	<ul style="list-style-type: none"> <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>	
P00–P07	I/O port P0	<ul style="list-style-type: none"> <li>8-bit CMOS I/O port</li> </ul>	
P10–P17	I/O port P1	<ul style="list-style-type: none"> <li>I/O direction register allows each pin to be individually programmed as either input or output.</li> </ul>	
P20–P27	I/O port P2	<ul style="list-style-type: none"> <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 bus I/O pins.</li> <li>CMOS compatible input level</li> <li>CMOS 3-state output structure</li> <li>Port P2 can be switched CMOS or TTL input level.</li> </ul>	
P30/RTP6, P31/RTP7 P34/CKOUT, P32, P33, P35–P37	I/O port P3	<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 control bus I/O pins.</li> <li>CMOS compatible input level</li> <li>CMOS 3-state output structure</li> <li>Port P32 can be switched CMOS or TTL input level.</li> </ul>	<ul style="list-style-type: none"> <li>Real time port function pins</li> <li>Clock output function pin</li> </ul>
P40/XCOUT, P41/XCIN P42/INT0, P43/INT1 P44/RxD, P45/TxD, P46/SCLK1, P47/SRDY1	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 structures</li> </ul>	<ul style="list-style-type: none"> <li>Sub-clock generating I/O pins(connect a resonator)</li> <li>Interrupt input pins</li> <li>Timer X, Timer Y function pins (INT0, INT1)</li> <li>Serial I/O1 function pins</li> </ul>
P50/TOUT P51/SCMP2/ INT2 P52/INT3, P53/INT4 P54/CNTR0, P55/CNTR1 P56/DA1, P57/DA2	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>	<ul style="list-style-type: none"> <li>Timer 2 output pin</li> <li>Interrupt input pin</li> <li>Serial I/O2 function pin</li> <li>Interrupt input pin</li> <li>Real time port function pin(INT4)</li> <li>Timer X, Timer Y function pins</li> <li>D-A conversion output pins</li> </ul>

**Table 2. Pin description (2)**

Pin	Name	Function	Function except a port function
P60/AN5– P62/AN7	I/O port P6	<ul style="list-style-type: none"> <li>• 3-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>	<ul style="list-style-type: none"> <li>• A-D conversion output pins</li> </ul>
P63/CMPIN/ AN8 P64/CMPREF/ AN9	Input port P6	<ul style="list-style-type: none"> <li>• 2-bit CMOS input port</li> <li>• CMOS compatible input level</li> </ul>	<ul style="list-style-type: none"> <li>• Analog comparator input pin</li> <li>• A-D conversion input pin</li> <li>• Reference voltage input pin for analog comparator</li> <li>• A-D conversion input pin</li> </ul>
P65/DAVREF/ AN10	I/O port P6	<ul style="list-style-type: none"> <li>• 1-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>	<ul style="list-style-type: none"> <li>• D-A conversion power source input pin</li> <li>• A-D conversion input pin</li> </ul>
P70/SIN2, P71/SOUT2, P72/SCLK2 P73/SRDY2/ ADT/AN0 P74/AN1– P77/AN4	I/O port P7	<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 structures</li> </ul>	<ul style="list-style-type: none"> <li>• Serial I/O2 function pins</li> <li>• Serial I/O2 function pin</li> <li>• A-D conversion input pin</li> <li>• A-D trigger input pin</li> <li>• A-D conversion input pin</li> </ul>
P80/DA3/ AN11, P81/DA4/ AN12, P82/RTP0– P87/RTP5	I/O port P8	<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 structures</li> </ul>	<ul style="list-style-type: none"> <li>• D-A conversion output pin</li> <li>• A-D conversion input pin</li> <li>• Realtime port function pins</li> </ul>

# HARDWARE

## PART NUMBERING

### PART NUMBERING

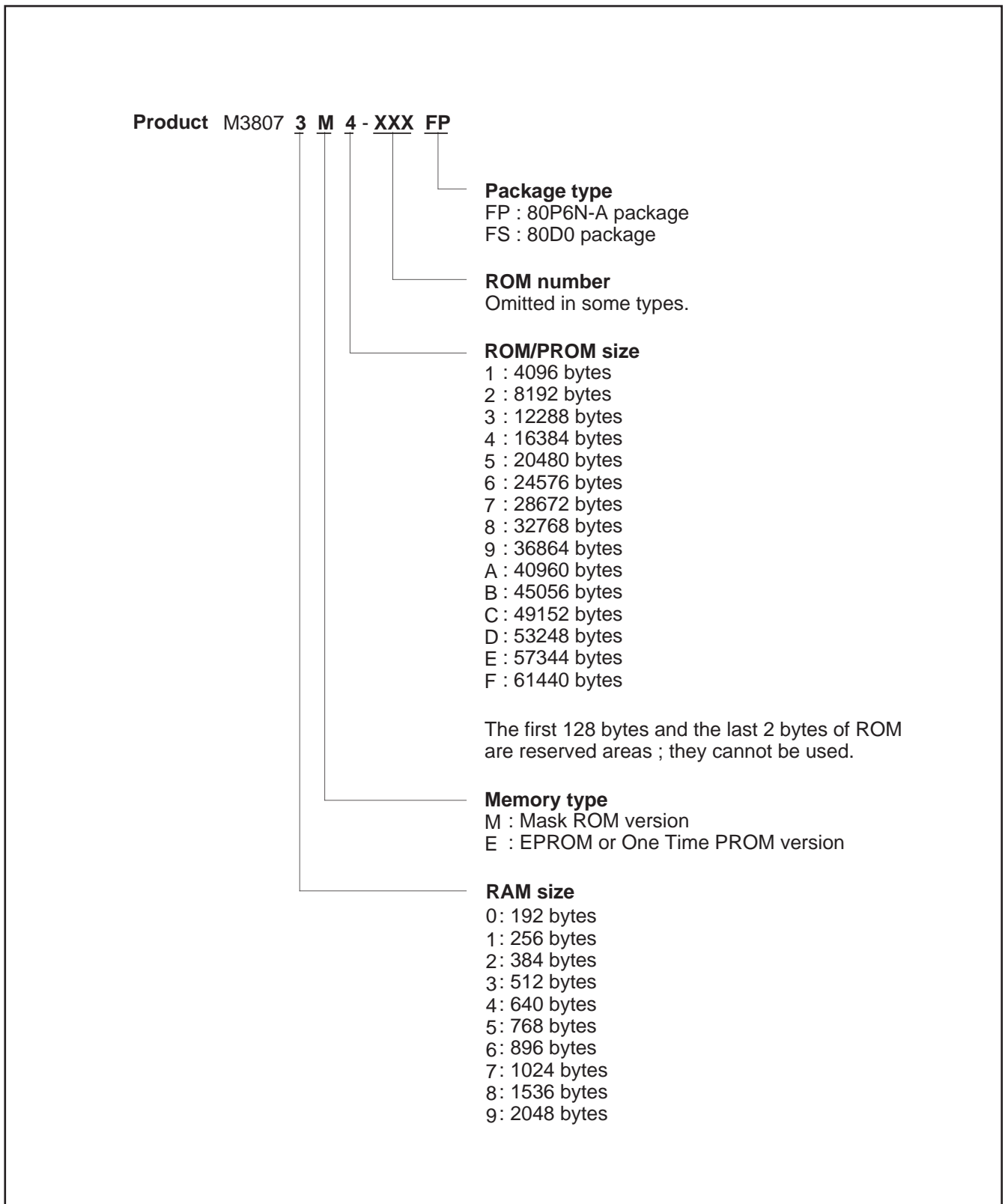


Fig. 3. Part numbering



### GROUP EXPANSION

Mitsubishi plans to expand the 3807 group as follows:

#### Memory Type

Support for Mask ROM, One Time PROM and EPROM versions.

#### Memory Size

ROM/PROM size ..... 8K to 60K bytes

RAM size ..... 384 to 2048 bytes

#### Package

80P6N-A ..... 0.8 mm-pitch plastic molded QFP

80D0 ..... 0.8 mm-pitch ceramic LCC (EPROM version)

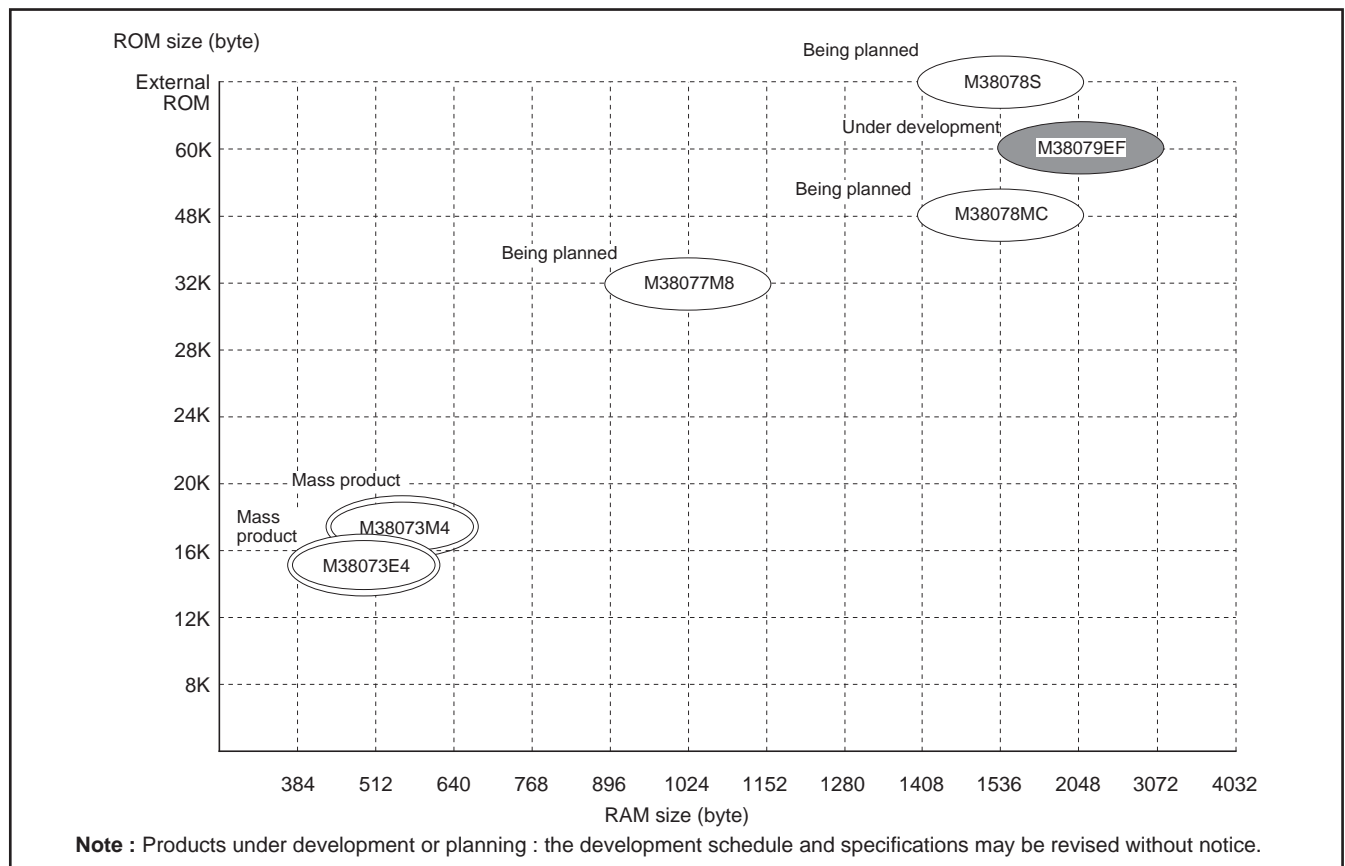


Fig. 4. Memory expansion plan

Currently supported products are listed below.

Table 3. List of supported products

As of May 1996

Product	(P) ROM size (bytes) ROM size for User ( )	RAM size (bytes)	Package	Remarks
M38073M4-XXXFP	16384 (16254)	512	80P6N-A	Mask ROM version
M38073E4-XXXFP				One Time PROM version
M38073E4FP				One Time PROM version (blank)
M38073E4FS			80D0	EPROM version

# HARDWARE

## FUNCTIONAL DESCRIPTION

### FUNCTIONAL DESCRIPTION Central Processing Unit (CPU)

The 3807 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 instructions cannot be used.

The MUL, DIV, WIT and STP 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 sub-routine 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.6.

### 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.

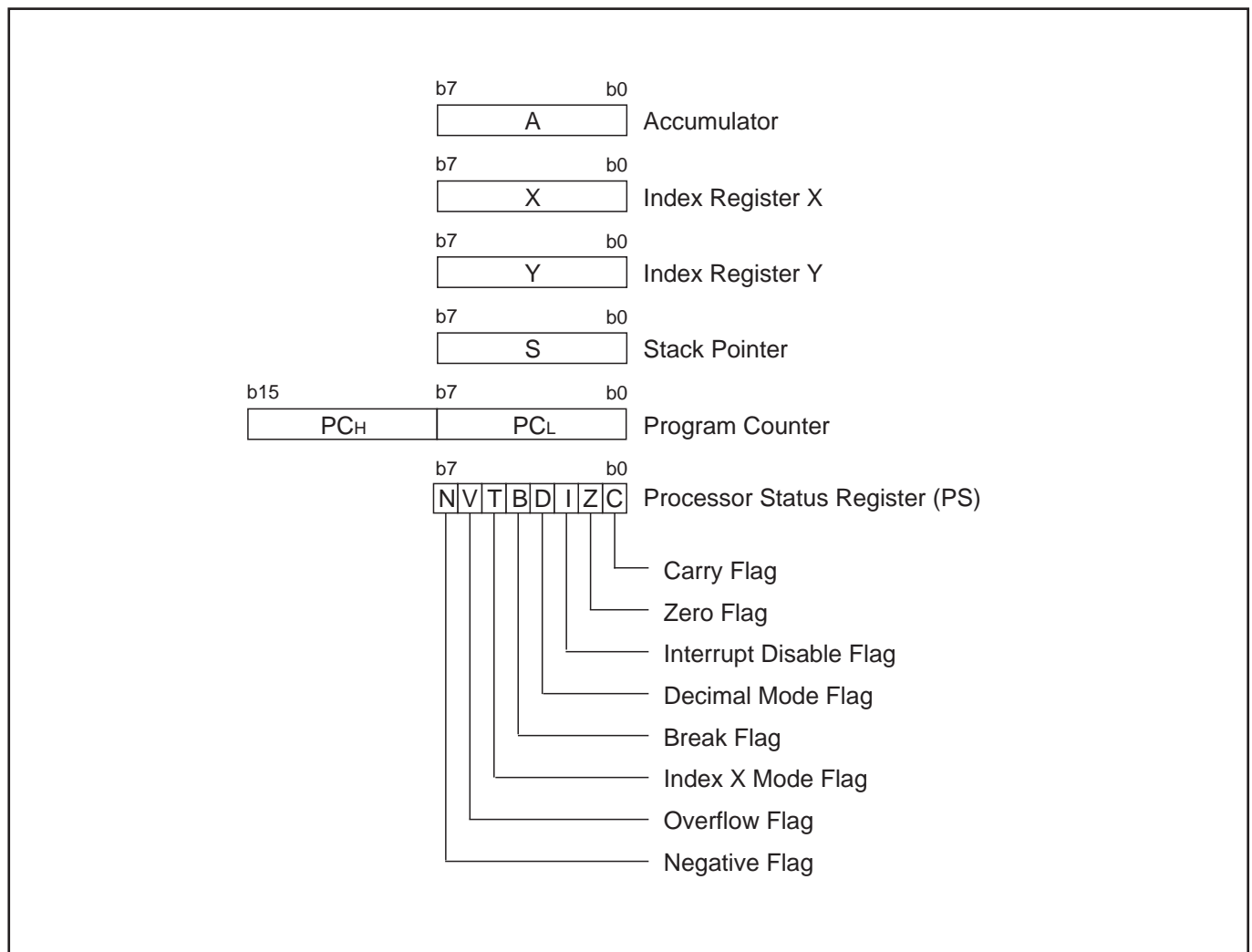


Fig. 5. 740 Family CPU register structure

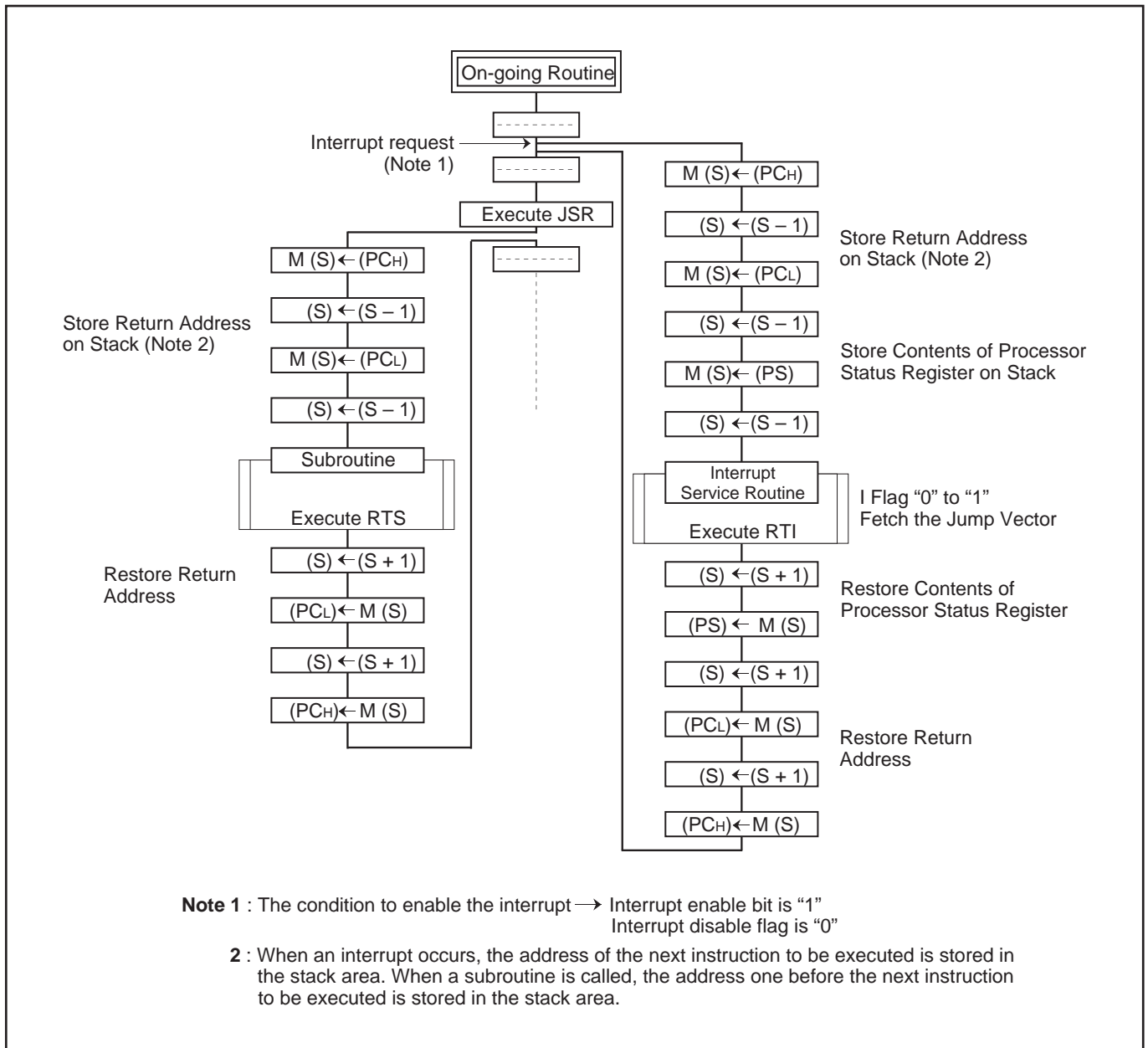


Fig. 6. Register push and pop at interrupt generation and subroutine call

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 contains the stack page selection bit and processor mode bits. The CPU mode register is allocated at address 003B16.

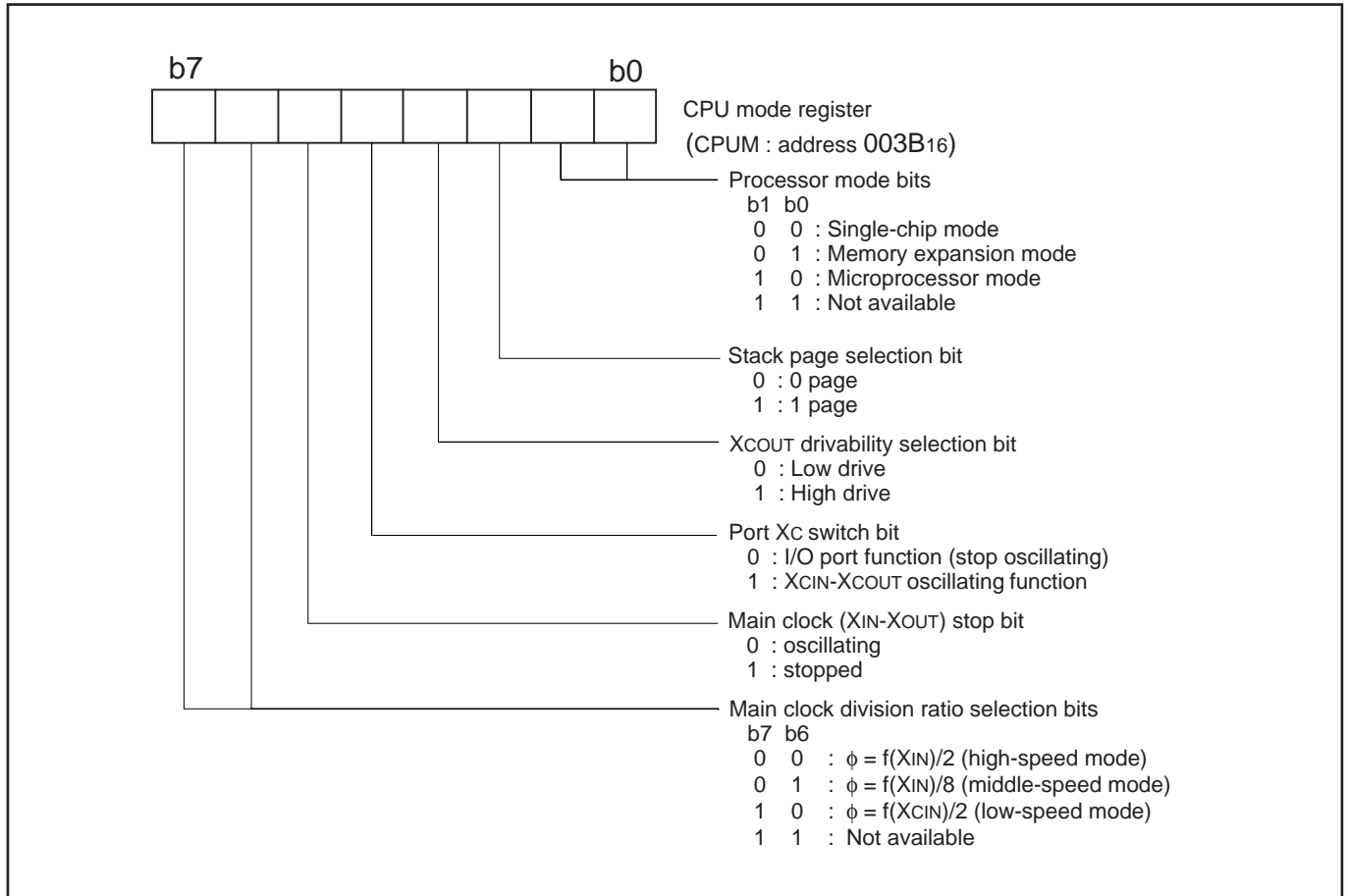


Fig. 7. Structure of CPU mode register

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Memory

#### Special function register (SFR) area

The special function register (SFR) 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 reset 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  $0000_{16}$  to  $00FF_{16}$  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  $FF00_{16}$  to  $FFFF_{16}$  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.

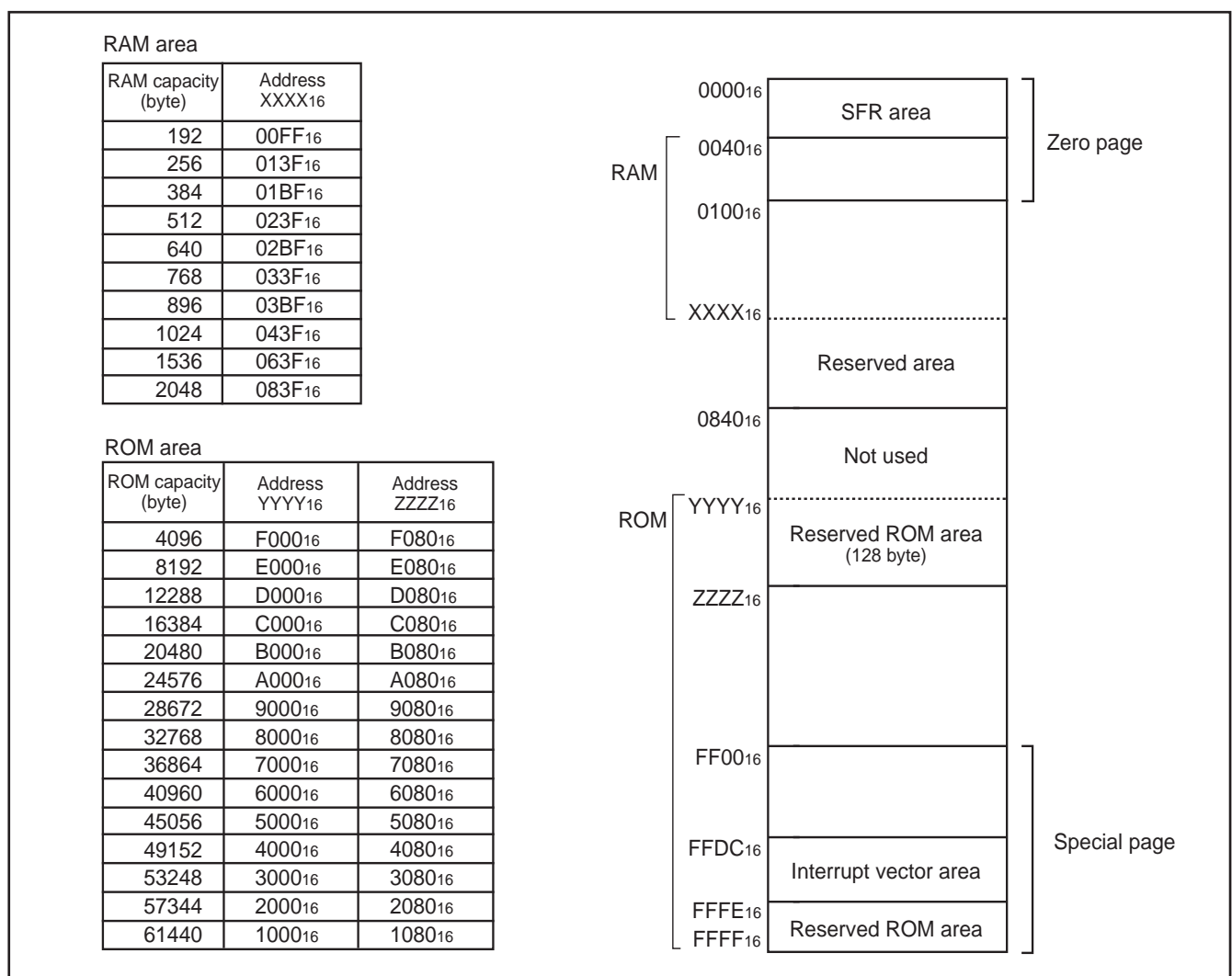


Fig. 8. Memory map diagram

0000 <sub>16</sub>	Port P0 (P0)	0020 <sub>16</sub>	Timer X (low-order) (TXL)
0001 <sub>16</sub>	Port P0 direction register (P0D)	0021 <sub>16</sub>	Timer X (high-order) (TXH)
0002 <sub>16</sub>	Port P1 (P1)	0022 <sub>16</sub>	Timer Y (low-order) (TYL)
0003 <sub>16</sub>	Port P1 direction register (P1D)	0023 <sub>16</sub>	Timer Y (high-order) (TYH)
0004 <sub>16</sub>	Port P2 (P2)	0024 <sub>16</sub>	Timer 1 (T1)
0005 <sub>16</sub>	Port P2 direction register (P2D)	0025 <sub>16</sub>	Timer 2 (T2)
0006 <sub>16</sub>	Port P3 (P3)	0026 <sub>16</sub>	Timer 3 (T3)
0007 <sub>16</sub>	Port P3 direction register (P3D)	0027 <sub>16</sub>	Timer X mode register (TXM)
0008 <sub>16</sub>	Port P4 (P4)	0028 <sub>16</sub>	Timer Y mode register (TYM)
0009 <sub>16</sub>	Port P4 direction register (P4D)	0029 <sub>16</sub>	Timer 123 mode register (T123M)
000A <sub>16</sub>	Port P5 (P5)	002A <sub>16</sub>	Real time port register (RTP)
000B <sub>16</sub>	Port P5 direction register (P5D)	002B <sub>16</sub>	Real time port control register 0 (RTPCON0)
000C <sub>16</sub>	Port P6 (P6)	002C <sub>16</sub>	Real time port control register 1 (RTPCON1)
000D <sub>16</sub>	Port P6 direction register (P6D)	002D <sub>16</sub>	Real time port control register 2 (RTPCON2)
000E <sub>16</sub>	Port P7 (P7)	002E <sub>16</sub>	Real time port control register 3 (RTPCON3)
000F <sub>16</sub>	Port P7 direction register (P7D)	002F <sub>16</sub>	Timer A (low-order) (TAL)
0010 <sub>16</sub>	Port P8 (P8)	0030 <sub>16</sub>	Timer A (high-order) (TAH)
0011 <sub>16</sub>	Port P8 direction register (P8D)	0031 <sub>16</sub>	Timer B (low-order) (TBL)
0012 <sub>16</sub>		0032 <sub>16</sub>	Timer B (high-order) (TBH)
0013 <sub>16</sub>		0033 <sub>16</sub>	D-A control register (DACON)
0014 <sub>16</sub>	Timer XY control register (TXYCON)	0034 <sub>16</sub>	A-D control register (ADCON)
0015 <sub>16</sub>	Port P2P3 control register (P2P3C)	0035 <sub>16</sub>	A-D conversion register (AD)
0016 <sub>16</sub>	Pull-up control register (PULL)	0036 <sub>16</sub>	D-A1 conversion register (DA1)
0017 <sub>16</sub>	Watchdog timer control register (WDTCN)	0037 <sub>16</sub>	D-A2 conversion register (DA2)
0018 <sub>16</sub>	Transmit/Receive buffer register (TB/RB)	0038 <sub>16</sub>	D-A3 conversion register (DA3)
0019 <sub>16</sub>	Serial I/O1 status register (SIO1STS)	0039 <sub>16</sub>	D-A4 conversion register (DA4)
001A <sub>16</sub>	Serial I/O1 control register (SIO1CON)	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>	Serial I/O2 control register 1 (SIO2CON1)	003D <sub>16</sub>	Interrupt request register 2 (IREQ2)
001E <sub>16</sub>	Serial I/O2 control register 2 (SIO2CON2)	003E <sub>16</sub>	Interrupt control register 1 (ICON1)
001F <sub>16</sub>	Serial I/O2 register (SIO2)	003F <sub>16</sub>	Interrupt control register 2 (ICON2)

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

# HARDWARE

## FUNCTIONAL DESCRIPTION

### I/O Ports

#### [Direction Registers] PiD

The 3807 group has 68 programmable I/O pins arranged in nine individual I/O ports (P0—P5, P60—P62, P65 and P7—P8). 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 pin, that pin becomes an output pin. If data is read from a pin set to output, the value of the port output latch is read, not the value of the pin itself. Pins set to input (the bit corresponding to that pin must be set to "0") are floating and the value of that pin can be written to. If a pin set to input is written to, only the port output latch is written to and the pin remains floating.

#### [Pull-up Control Register] PULL

Ports P0, P1 and P2 have built-in programmable pull-up resistors. The pull-up resistors are valid only in the case that the each control bit is set to "1" and the corresponding port direction registers are set to input mode.

#### (1) CMOS/TTL input level selection

Either CMOS input level or TTL input level can be selected as an input level for ports P20 to P27 and P32. The input level is selected by P2-P32 input level selection bit (b7) of the port P2P3 control register (address 0015<sub>16</sub>). When the bit is set to "0", CMOS input level is selected. When the bit is set to "1", the TTL input level is selected. After this bit is re-set, its initial value depends on the state of the CNVss pin. When the CNVss pin is connected to Vss, the initial value becomes "0". When the CNVss pin is connected to Vcc, the initial value becomes "1".

#### (2) Notes on STP instruction execution

Make sure that the input level at each pin is either 0V or to Vcc during execution of the STP instruction. When an input level is at an intermediate potential, a current will flow from Vcc to Vss through the input-stage gate.

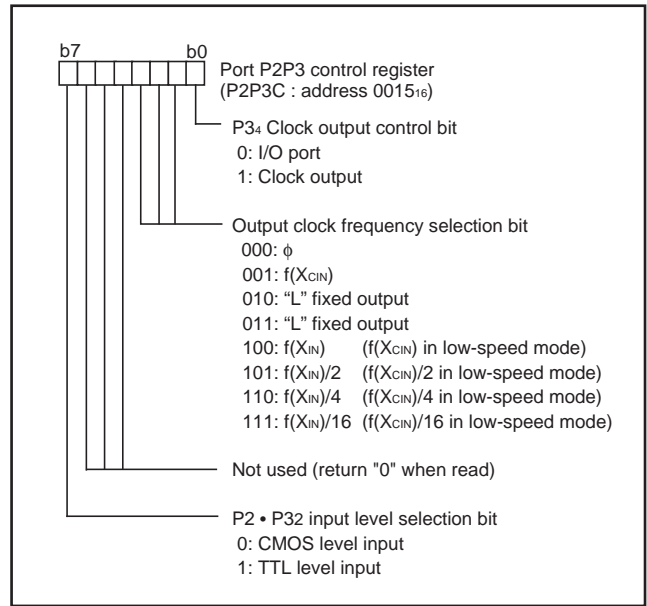


Fig. 10. Structure of Port P2P3 control register

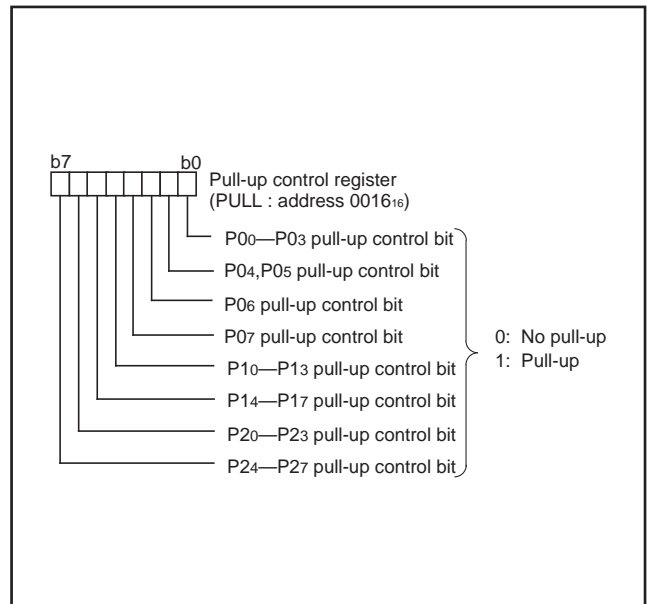


Fig. 11. Structure of Pull-up control register



**Table 6. List of I/O port functions (1)**

Pin	Name	Input/Output	I/O Format	Non-Port Function	Related SFRs	Ref.No.
P00–P07	Port P0	Input/output, individual bits	CMOS compatible input level	Address low-order byte output	CPU mode register	(1)
P10–P17	Port P1		CMOS 3-state output	Address high-order byte output	Pull-up control register	
P20–P27	Port P2		CMOS/TTL input level CMOS 3-state output	Data bus I/O	CPU mode register Pull-up control register Port P2P3 control register	
P30/RTP6, P31/RTP7	Port P3		CMOS compatible input level	Real time port output	CPU mode register	(2)
P32			CMOS 3-state output	Control signal input	CPU mode register Port P2P3 control register	
P33			CMOS/TTL input level CMOS 3-state output	Control signal output	CPU mode register	(3)
P34/CKOUT				Clock output, $\phi$ output	CPU mode register Port P2P3 control register	(4)
P35–P37				Control signal I/O	CPU mode register	(3)
P40/XCOUT, P41/XCIN	Port P4			Sub-clock generating circuit	CPU mode register	(5)
P42/INT0, P43/INT1				External interrupt input Timer X, Timer Y function input	Interrupt edge selection register	(6)
P44/RxD, P45/TxD, P46/SCLK1, P47/SRDY1				Serial I/O1 function I/O	Serial I/O1 control register UART control register	(7)
P50/TOUT				Timer 2 output	Timer 123 mode register	(8)
P51/SCMP2/ INT2				External interrupt input Serial I/O2 function I/O	Interrupt edge selection register Serial I/O2 control register	(9)
P52/INT3, P53/INT4	External interrupt input Real time port trigger input (INT4)	Interrupt edge selection register	(10)			
P54/CNTR0 P55/CNTR1	Timer X, Timer Y function I/O	Timer X mode register Timer Y mode register	(11)			
P56/DA1, P57/DA2	D-A conversion output	D-A control register	(12)			
P60/AN5— P62/AN7	Port P5			A-D conversion input	A-D control register	(13)
P63/CMPIN/ AN8	Port P6	Input	CMOS compatible input level	Analog comparator input pin A-D conversion input	A-D control register	(14)
P64/CMPREF/ AN9				Analog comparator reference voltage input pin A-D conversion input		
P65/DAVREF/ AN10		Input/output, individual bits	CMOS compatible input level CMOS 3-state output	D-A converter power source input A-D conversion input	A-D control register	(15)
P70/SIN2, P71/SOUT2, P72/SCLK2	Port P7			Serial I/O2 function I/O	Serial I/O2 control register	(16)
P73/SRDY2/ ADT/AN0				Serial I/O2 function I/O A-D trigger input A-D conversion input	Serial I/O2 control register A-D control register	(17)
P74/AN1— P77/AN4				A-D conversion input	A-D control register	(18)

# HARDWARE

## FUNCTIONAL DESCRIPTION

Table 7. List of I/O port functions (2)

Pin	Name	Input/Output	I/O Format	Non-Port Function	Related SFRs	Ref.No.
P80/DA3/ AN11	Port P8	Input/output, individual bits	CMOS compatible input level CMOS 3-state output	D-A conversion output	D-A control register	(14)
P81/DA4/ AN12				A-D conversion input	A-D control register	
P82/RTP0— P87/RTP5				Real time port output	Real time port control register	(23)

**Note1** : 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.

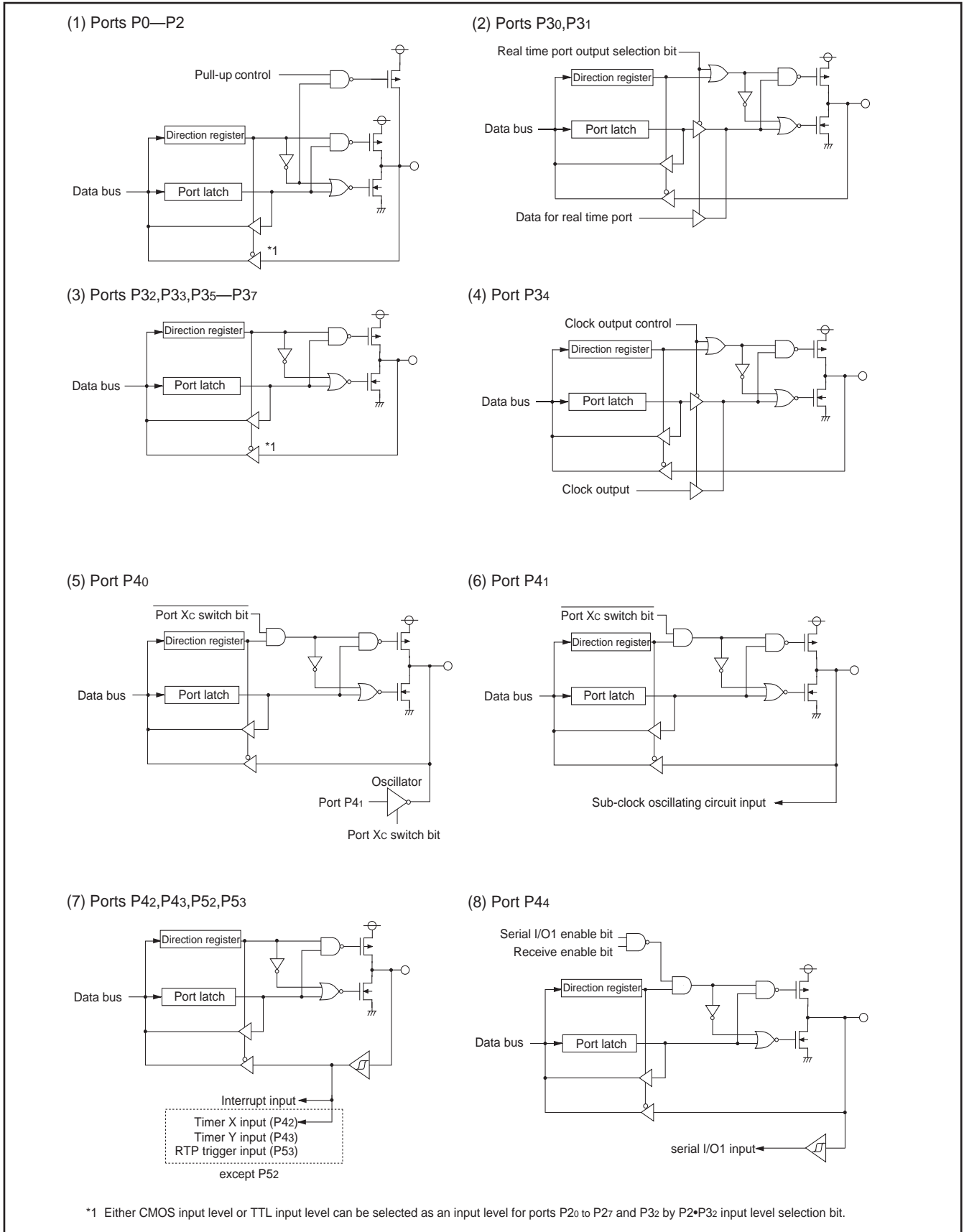


Fig. 12. Port block diagram (1)

# HARDWARE

## FUNCTIONAL DESCRIPTION

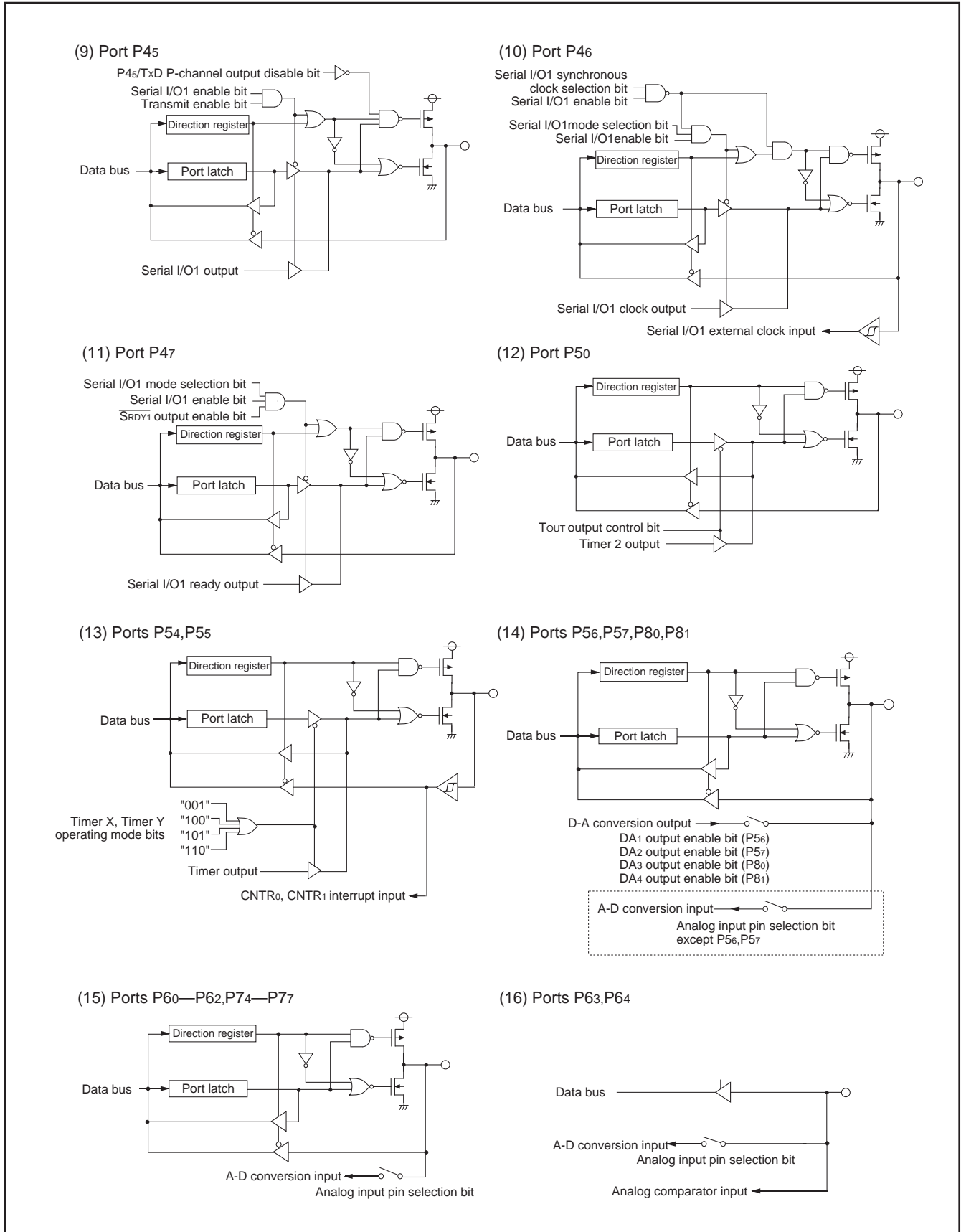


Fig. 13. Port block diagram (2)

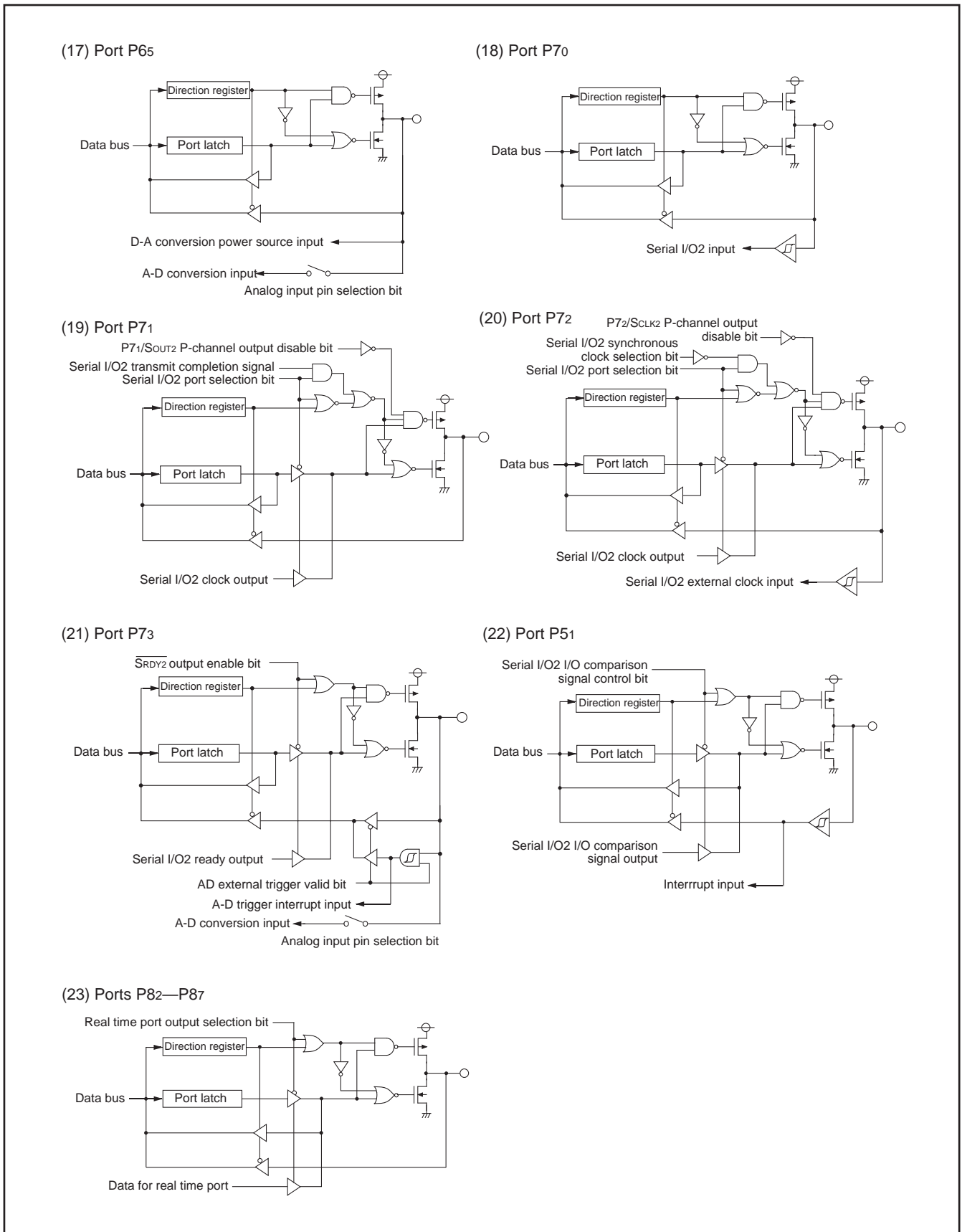


Fig. 14. Port block diagram (3)

# HARDWARE

## FUNCTIONAL DESCRIPTION

---

### Interrupts

Interrupts occur by twenty sources: eight external, eleven internal, and one software.

#### (1) Interrupt Control

Each interrupt except the BRK instruction interrupt have both an interrupt request bit and an interrupt enable bit, and is controlled by the interrupt disable flag. 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 interrupt and reset cannot be disabled with any flag or bit. The I flag disables all interrupts except the BRK instruction interrupt and reset. If several interrupts requests occurs at the same time the interrupt with highest priority is accepted first.

#### (2) Interrupt Operation

Upon acceptance of an interrupt the following operations are automatically performed:

1. The processing being executed is stopped.
2. The contents of the program counter and processor status register are automatically pushed onto the stack
3. Concurrently with the push operation, the interrupt jump destination address is read from the vector table into the program counter.
4. The interrupt disable flag is set and the corresponding interrupt request bit is cleared.

#### ■Notes on Use

When the active edge of an external interrupt (INT<sub>0</sub>—INT<sub>4</sub>, CNTR<sub>0</sub> or CNTR<sub>1</sub>) is set or the timer /INT interrupt source and the ADT/ A-D conversion interrupt source are 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 in interrupt edge selection register (in case of CNTR<sub>0</sub>: Timer X mode register ; in case of CNTR<sub>1</sub>: Timer Y mode register).
- (3) Clear the set interrupt request bit to "0."
- (4) Enable the external interrupt which is selected.

**Table 8. Interrupt vector addresses and priority**

Interrupt Source	Priority	Vector Addresses (Note 1)		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset (Note 2)	1	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/O1 receive	4	FFF7 <sub>16</sub>	FFF6 <sub>16</sub>	At completion of serial I/O1 data receive	Valid when serial I/O1 is selected
Serial I/O1 transmit	5	FFF5 <sub>16</sub>	FFF4 <sub>16</sub>	At completion of serial I/O1 data transmit shift or when transmit buffer is empty	Valid when serial I/O1 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	
INT <sub>3</sub>	8	FFEF <sub>16</sub>	FFEE <sub>16</sub>	At detection of either rising or falling edge of INT <sub>3</sub> input	External interrupt (active edge selectable) Valid when INT <sub>3</sub> interrupt is selected
Timer 2				At timer 2 underflow	Valid when timer 2 interrupt is selected
INT <sub>4</sub>	9	FFED <sub>16</sub>	FFEC <sub>16</sub>	At detection of either rising or falling edge of INT <sub>4</sub> input	External interrupt (active edge selectable) Valid when INT <sub>4</sub> interrupt is selected
Timer 3				At timer 3 underflow	Valid when timer 3 interrupt is selected
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)
Serial I/O2	12	FFE7 <sub>16</sub>	FFE6 <sub>16</sub>	At completion of serial I/O2 data transmit and receive	Valid when serial I/O2 is selected
INT <sub>2</sub>	13	FFE5 <sub>16</sub>	FFE4 <sub>16</sub>	At detection of either rising or falling edge of INT <sub>2</sub> input	External interrupt (active edge selectable) Valid when INT <sub>2</sub> interrupt is selected
Timer 1				At timer 1 underflow	Valid when timer 1 interrupt is selected
Timer A	14	FFE3 <sub>16</sub>	FFE2 <sub>16</sub>	At timer A underflow	
Timer B	15	FFE1 <sub>16</sub>	FFE0 <sub>16</sub>	At timer B underflow	
A-D conversion	16	FFDF <sub>16</sub>	FFDE <sub>16</sub>	At completion of A-D conversion	Valid when A-D interrupt is selected
ADT				At falling edge of ADT input	External interrupt(valid at falling) Valid when ADT interrupt is selected and when A-D external trigger is selected.
BRK instruction	17	FFDD <sub>16</sub>	FFDC <sub>16</sub>	At BRK instruction execution	Non-maskable software interrupt

**Note1** : Vector addresses contain interrupt jump destination addresses.

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

# HARDWARE

## FUNCTIONAL DESCRIPTION

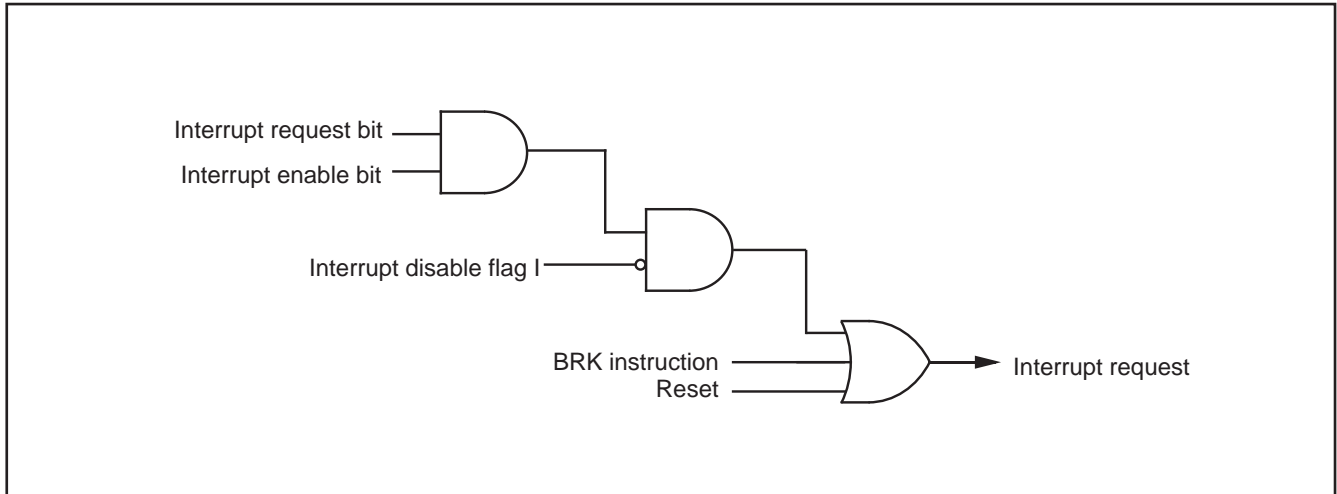


Fig. 15. Interrupt control

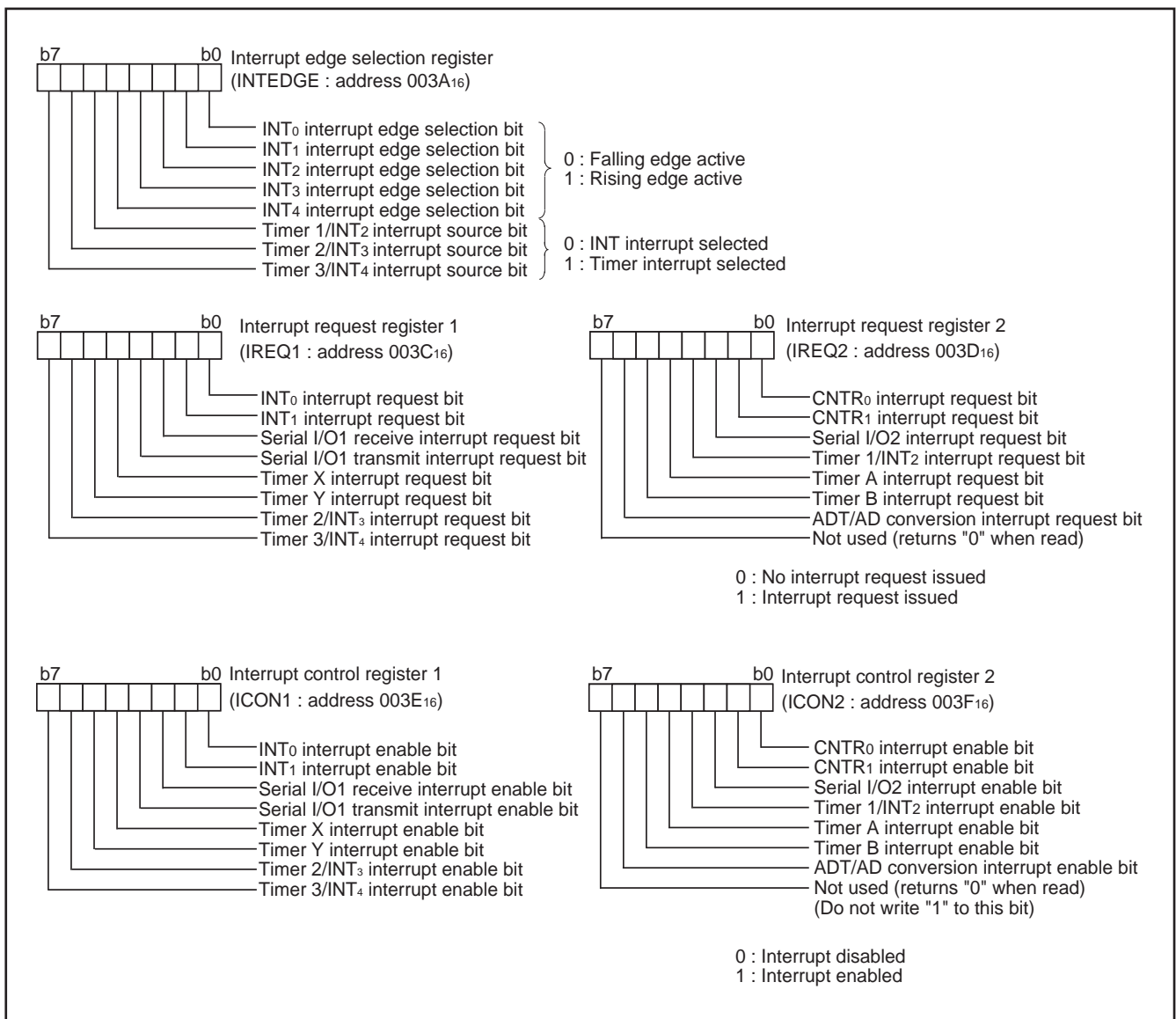


Fig. 16. Structure of Interrupt-related registers



### Timers

The 3807 group has seven timers : four 16-bit timers (Timer X, Timer Y, Timer A, and Timer B) and three 8-bit timers (Timer 1, Timer 2, and Timer 3).

All timers are down-counters. When the timer reaches either "0016" or "000016", an underflow occurs with the next count pulse. Then the contents of the timer latch is reloaded into the timer and the timer continues down-counting. When a timer underflows, the interrupt request bit corresponding to that timer is set to "1."

Read and write operation on 16-bit timer must be performed for both high- and low-order bytes. When reading a 16-bit timer, read from the high-order byte first. When writing to 16-bit timer, write to the low-order byte first. The 16-bit timer cannot perform the correct operation when reading during write operation, or when writing during read operation.

Timers A and B are real time output port timers. For details, refer to the section "Real time output port".

#### ●Timer X, Timer Y

Timer X and Y are independent 16-bit timers which can select enable seven different operation modes each by the setting of their mode registers. The related registers of timer X and Y are listed below. The following register abbreviations are used:

- Timer XY control register (TXYCON: address 0014<sub>16</sub>)
- Port P4 direction register (P4D: address 0009<sub>16</sub>)
- Port P5 direction register (P5D: address 000B<sub>16</sub>)
- Timer X (low-order) (TXL: address 0020<sub>16</sub>)
- Timer X (high-order) (TXH: address 0021<sub>16</sub>)
- Timer Y (low-order) (TYL: address 0022<sub>16</sub>)
- Timer Y (high-order) (TYH: address 0023<sub>16</sub>)
- Timer X mode register (TXM: address 0027<sub>16</sub>)
- Timer Y mode register (TYM: address 0028<sub>16</sub>)
- Interrupt edge selection register (INTEDGE: address 003A<sub>16</sub>)
- Interrupt request register 1 (IREQ1: address 003C<sub>16</sub>)
- Interrupt request register 2 (IREQ2: address 003D<sub>16</sub>)
- Interrupt control register 1 (ICON1: address 003E<sub>16</sub>)
- Interrupt control register 2 (ICON2: address 003F<sub>16</sub>)

For details, refer to the structures of each register.

The following is an explanation of the seven modes:

#### (1) Timer • event counter mode

##### ①Timer mode

###### • Mode selection

This mode can be selected by setting "000" to the following bits.

Timer X operating mode bit (bits 2 to 0) of TXM

Timer Y operating mode bit (bits 2 to 0) of TYM

###### • Count source selection

In high- or middle-speed mode,  $f(X_{IN})/2$ ,  $f(X_{IN})/16$ , or  $f(X_{CIN})$  can be selected as the count source.

In low-speed mode the count source is  $f(X_{CIN})$ .

A count source is selected by the following bit.

Timer X count source selection bit (bits 7 and 6) of TXM

Timer Y count source selection bit (bits 7 and 6) of TYM

###### • Interrupt

When an underflow is generated, the corresponding timer X interrupt request bit (b4) or timer Y interrupt request bit (b5) of IREQ1

is set to "1".

###### • Explanation of operation

After reset release, timer X stop control bit (b0) and timer Y stop control bit (b1) of TXYCON are set to "1" and the timer stops. During timer stop, a timer value written to the timer X or timer Y is set by writing data to the corresponding timer latch and timer at the same time. The timer operation is started by setting the bits 0 or 1 of TXYCON to "0". When the timer reaches "000016", an underflow occurs with the next count pulse. Then the contents of the timer latch is reloaded into the timer and the timer continues down-counting. For changing a timer value during count operation, a latch value must be changed by writing data only to the corresponding latch first. Then the timer is reloaded with the new latch value at the next underflow.

#### ②Event counter mode

##### • Mode selection

This mode can be selected by the following sequence.

1. Set "000" to the timer X operating mode bit (bits 2 to 0) of TXM, or to the timer Y operating mode bit (bits 2 to 0) of TYM.
2. Select an input signal from the CNTR<sub>0</sub> pin (in case of timer X ; set "11" to bits 7 and 6 of TXM), or from the CNTR<sub>1</sub> pin (in case of timer Y ; set "11" to bits 7 and 6 of TYM) as a count source.

The valid edge for the count operation is selected by the CNTR<sub>0</sub>/CNTR<sub>1</sub> active edge switch bit (b5) of TXM or TYM: if set to "0", counting starts with the rising edge or if set to "1", counting starts with the falling edge.

##### • Interrupt

The interrupt generation at underflow is the same as already explained for the timer mode.

##### • Explanation of operation

The operation is the same as already explained for the timer mode. In this mode, the double-function port of CNTR<sub>0</sub>/CNTR<sub>1</sub> pin must be set to input.

Figure 19 shows the timing chart for the timer • event counter mode.

#### (2) Pulse output mode

##### • Mode selection

This mode can be selected by setting "001" to the following bits.

Timer X operating mode bit (bits 2 to 0) of TXM

Timer Y operating mode bit (bits 2 to 0) of TYM

##### • Count source selection

In high- or middle-speed mode,  $f(X_{IN})/2$ ,  $f(X_{IN})/16$ , or  $f(X_{CIN})$  can be selected as the count source.

In low-speed mode the count source is  $f(X_{CIN})$ .

##### • Interrupt

The interrupt generation at underflow is the same as already explained for the timer mode.

##### • Explanation of operation

Counting operation is the same as in timer mode. Moreover the pulse which is inverted each time the timer underflows is output from CNTR<sub>0</sub>/CNTR<sub>1</sub> pin. When the CNTR<sub>0</sub>/CNTR<sub>1</sub> active edge switch bit (b5) of TXM or TYM is "0", output starts with "H" level. When set to "1", output starts with "L" level.

### ■Precautions

Set the double-function port of CNTR0/CNTR1 pin to output in this mode.

[During timer operation stop]

The output from CNTR0/CNTR1 pin is initialized to the level set through CNTR0/CNTR1 active edge switch bit.

[During timer operation enabled]

When the value of the CNTR0/CNTR1 active edge switch bit is written over, the output level of CNTR0/CNTR1 pin is inverted.

Figure 20 shows the timing chart of the pulse output mode.

### (3) Pulse period measurement mode

#### • Mode selection

This mode can be selected by setting "010" to the following bits.

Timer X operating mode bit (bits 2 to 0) of TXM

Timer Y operating mode bit (bits 2 to 0) of TYM

#### • Count source selection

In high- or middle-speed mode,  $f(X_{IN})/2$  or  $f(X_{IN})/16$  can be selected as the count source.

In low-speed mode the count source is  $f(X_{CIN})$ .

#### • Interrupt

The interrupt generation at underflow is the same as already explained for the timer mode. Bits 0 or 1 of IREQ2 is set to "1" synchronously to pulse period measurement completion.

#### • Explanation of operation

[During timer operation stop]

Select the count source. Next, select the interval of the pulse periods to be measured. When bit 5 of the TXM or TYM is set to "0", the timer counts during the interval of one falling edge of CNTR0/CNTR1 pin input until the next falling edge of input. If bits 5 are set to "1", the timer counts during the interval of one rising edge until the next rising edge.

[During timer operation enabled]

The pulse period measurement starts by setting bit 0 or 1 of TXYCON to "0" and the timer counts down from the value that was set to the timer before the start of measurement. When a valid edge of measurement start/stop is detected, the 1's complement of the timer value is written to the timer latch and "FFFF16" is set to the timer. Furthermore when the timer underflows, a timer X/Y interrupt request occurs and "FFFF16" is set to the timer. The measured value is held until the next measurement completion.

### ■Precautions

Set the double-function port of CNTR0/CNTR1 pin to input in this mode.

A read-out of timer value is impossible in this mode. The timer is written to only during timer stop (no measurement of pulse periods). Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operations during measurement.

The timer is set to "FFFF16" when the timer either underflows or a valid edge of pulse period measurement is detected. Due to that, the timer value at the start of measurement depends on the timer value before the start of measurement.

Figure 21 shows the timing chart of the pulse period measurement mode.

### (4) Pulse width measurement mode

#### • Mode selection

This mode can be selected by setting "011" to the following bits.

Timer X operating mode bit (bits 2 to 0) of TXM

Timer Y operating mode bit (bits 2 to 0) of TYM

#### • Count source selection

In high- or middle-speed mode,  $f(X_{IN})/2$  or  $f(X_{IN})/16$  can be selected as the count source.

In low-speed mode the count source is  $f(X_{CIN})$ .

#### • Interrupt

The interrupt generation at underflow is the same as already explained for the timer mode. Bit 0 or 1 of IREQ2 is set to "1" synchronously to pulse width measurement completion.

#### • Explanation of operation

[During timer operation stop]

Select the count source. Next, select the interval of the pulse widths to be measured. When bit 5 of TXM or TYM is set to "1", the timer counts during the interval of one falling edge of CNTR0/CNTR1 pin input until the next rising edge of input ("L" interval). If bit 5 is set to "0", the timer counts during the interval of one rising edge until the next falling edge ("H" interval).

[During timer operation enabled]

The pulse width measurement starts by setting bit 0 or 1 of TXYCON to "0" and the timer counts down from the value that was set to the timer before the start of measurement. When a valid edge of measurement completion is detected, the 1's complement of the timer value is written to the timer latch and "FFFF16" is set to the timer. Furthermore when the timer underflows, a timer X/Y interrupt request occurs and "FFFF16" is set to the timer. The measured value is held until the next measurement completion.

### ■Precautions

Set the double-function port of CNTR0/CNTR1 pin to input in this mode.

A read-out of timer value is impossible in this mode. The timer is written to only during timer stop (no measurement of pulse widths). Since the timer latch in this mode is specialized for the read-out of measured values, do not perform any write operations during measurement.

The timer value is set to "FFFF16" when the timer either underflows or a valid edge of pulse widths measurement is detected. Due to that, the timer value at the start of measurement depends on the timer value before the start of measurement.

Figure 22 shows the timing chart of the pulse width measurement mode.

### (5) Programmable waveform generation mode

#### • Mode selection

This mode can be selected by setting "100" to the following bits.

Timer X operating mode bit (bits 2 to 0) of TXM

Timer Y operating mode bit (bits 2 to 0) of TYM

#### • Count source selection

In high- or middle-speed mode,  $f(X_{IN})/2$ ,  $f(X_{IN})/16$ , or  $f(X_{CIN})$  can be selected as the count source.

In low-speed mode the count source is  $f(X_{CIN})$ .

#### • Interrupt

The interrupt generation at underflow is the same as already

explained for the timer mode.

- Explanation of operation

Counting operation is the same as in timer mode. Moreover the timer outputs the data set in the corresponding output level latch (bit 4 of TXM or TYM) to CNTR0/CNTR1 pin each time the timer underflows. After the timer underflows, the generation of optional waveform from CNTR0/CNTR1 pin is possible through a change of values in the output level latch and timer latch.

- Precautions

Set the double-function port of CNTR0/CNTR1 pin to output in this mode.

Figure 23 shows the timing chart of the programmable waveform generation mode.

### (6) Programmable one-shot generating mode

- Mode selection

This mode can be selected by setting "101" to the following bits.

Timer X operating mode bit (bits 2 to 0) of TXM

Timer Y operating mode bit (bits 2 to 0) of TYM

- Count source selection

In high- or middle-speed mode,  $f(X_{IN})/2$  or  $f(X_{IN})/16$  can be selected as the count source.

- Interrupt

The interrupt generation at underflow is the same as already explained for the timer mode. The one-shot generating trigger condition must be set to the INT0 interrupt edge selection bit (b0) and INT1 interrupt edge selection bit (b1) of INTEDGE. Setting these bits to "0" causes the interrupt request being triggered by a falling edge, setting them to "1" causes the interrupt request being triggered by a rising edge. The INT0 interrupt request bit (b0) and INT1 interrupt request bit (b1) of IREQ1 are set to "1" by detecting the active edge of the INT pin.

- Explanation of operation

For a "H" one-shot pulse, set bit 5 of TXM, TYM to "0".

[During timer operation stop]

The output level of CNTR0/CNTR1 pin is initialized to "L" at mode selection. Set the one-shot pulse width to TXH, TXL, TYH, TYL. A trigger generation during timer stop (input signal to INT0/INT1 pin) is invalid.

[During timer operation enabled]

When a trigger generation is detected, "H" is output, and at underflow "L" is output from CNTR0/CNTR1 pin.

For a "L" one-shot pulse set bit 5 of TXM, TYM to "1".

[During timer operation stop]

The output level of CNTR0/CNTR1 pin is initialized to "H" at mode selection. Set the one-shot pulse width to TXH, TXL, TYH, TYL. A trigger generation during timer stop (input signal to INT0/INT1 pin) is invalid.

[During timer operation enabled]

When a trigger generation is detected, "L" is output, and at underflow "H" is output from CNTR0/CNTR1 pin.

- Precautions

- Set the double-function port of CNTR0/CNTR1 pin to output and the double-function port of INT0/INT1 pin to input in this mode.
- This mode is unused in low-speed mode.

- During one-shot generation permission or one-shot generation the output level from CNTR0/CNTR1 pin changes if the value of the CNTR0/CNTR1 active edge switch bit is inverted.

Figure 24 shows the timing chart of the programmable one-shot generating mode.

### (7) PWM mode

- Mode selection

This mode can be selected by setting "110" to the following bits.

Timer X operating mode bit (bits 2 to 0) of TXM

Timer Y operating mode bit (bits 2 to 0) of TYM

- Count source selection

In high- or middle-speed mode,  $f(X_{IN})/2$  or  $f(X_{IN})/16$  can be selected as the count source.

- Interrupt

With a rising edge of CNTR0/CNTR1 output, the timer X interrupt request bit (b4) and timer Y interrupt request bit (b5) of IREQ1 are set to "1".

- Explanation of operation

PWM waveform is output from CNTR0 pin (in case of timer X) or from CNTR1 pin (in case of timer Y).

The "H" interval of PWM waveform is determined by the setting value  $m$  ( $m=0$  to 255) of TXH and TYH and the "L" interval of PWM waveform is determined by the setting value  $n$  ( $n=0$  to 255) of TXL and TYL.

The PWM cycles are:

$$\text{PWM cycle time} = (m+n) \cdot t_s$$

$$\text{PWM duty} = m / (m+n)$$

where:  $t_s$ : period of timer X/timer Y count source

[During count operation stop]

When a timer value is set to TXL, TXH, TYL, TYH by writing data to timer and timer latch at the same time. When setting this value, the output of CNTR0/CNTR1 pin is initialized to the "H" level.

[During count operation enabled]

By setting the bit 0 or 1 of TXYCON to "0", an "H" interval of TXH or TYH is output first, and after that a "L" level interval of TXL or TYL are output next. These operations are repeated continuously. The PWM output is changed after the underflow by setting a timer value, which is set by writing data to the timer latch only, to TXL, TXH, TYL, TYH.

- Precautions

- Set the double-function port of CNTR0/CNTR1 pin to output in this mode.
- This mode is unused in low-speed mode.
- When the PWM "H" interval is set to "0016", PWM output is "L".
- When the PWM "L" interval is set to "0016", PWM output is "H".
- When the PWM "H" interval and "L" interval are set to "0016", PWM output is "L".
- When a PWM "H" interval or "L" interval is set to "0016" at least for a short time, timer X/timer Y interrupt request does not occur.
- When the value set to the timer latch is "0016", the value is undefined since the timer counts down by dummy count operation.

Figure 25 shows the timing chart of the PWM mode.

# HARDWARE

## FUNCTIONAL DESCRIPTION

---

### ■Precautions regarding all modes

#### • Timer X, timer Y writing control

One of the following operation is selected by bit 3 of TXM or TYM for timer X or timer Y.

Writing data to the corresponding latch and timer at the same time

Writing data to only corresponding latch

When the operation "writing data to only corresponding latch" is selected, the value is set to the timer latch by writing a value to timer X/Y address and a timer is renewed at the next underflow. After releasing a reset, "writing the corresponding latch and timer at the same time" is selected. When a value is written to timer X/Y address, a value is set to a timer and a timer latch at the same time. When "writing data to only corresponding latch" is selected, if writing to a reload latch and an underflow are performed at the same timing, the timer value is undefined.

#### • Timer X, timer Y read control

In pulse period measurement mode and pulse width measurement mode the timer value cannot be read-out. In all other modes read-out operations without effect to count operations/stops are possible. However, the timer latch value cannot be read-out.

#### • Precautions regarding the CNTR<sub>0</sub>/CNTR<sub>1</sub> active edge switch bit and the INT<sub>0</sub>/INT<sub>1</sub> interrupt edge selection bit:

The CNTR<sub>0</sub>/CNTR<sub>1</sub> active edge switch bit and the INT<sub>0</sub>/INT<sub>1</sub> interrupt edge selection bit settings have an effect also on each interrupt active edge.

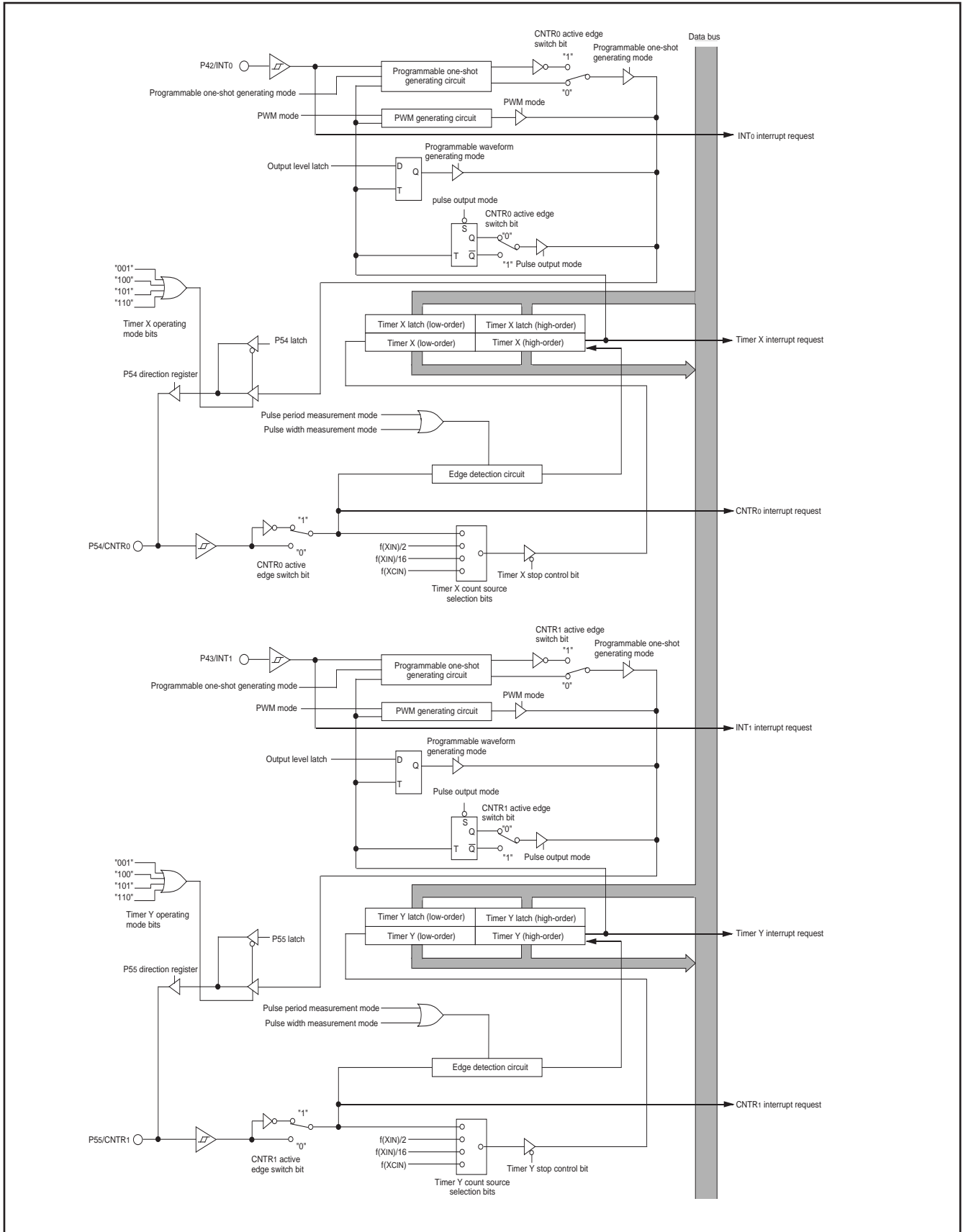


Fig. 17. Block diagram of Timer X and Timer Y

# HARDWARE

## FUNCTIONAL DESCRIPTION

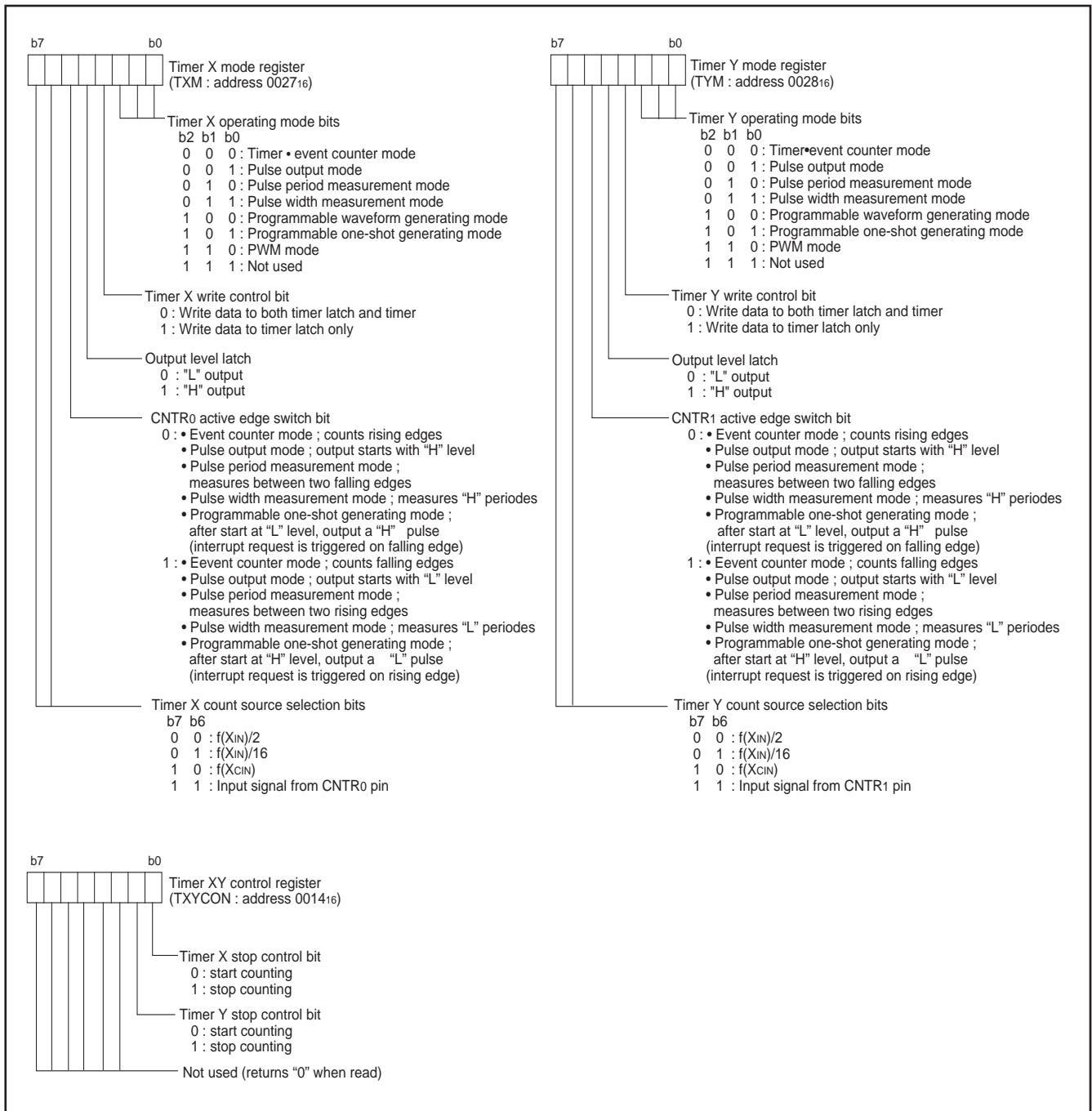


Fig. 18. Structure of Timer X mode register, Timer Y mode register, and Timer XY control register

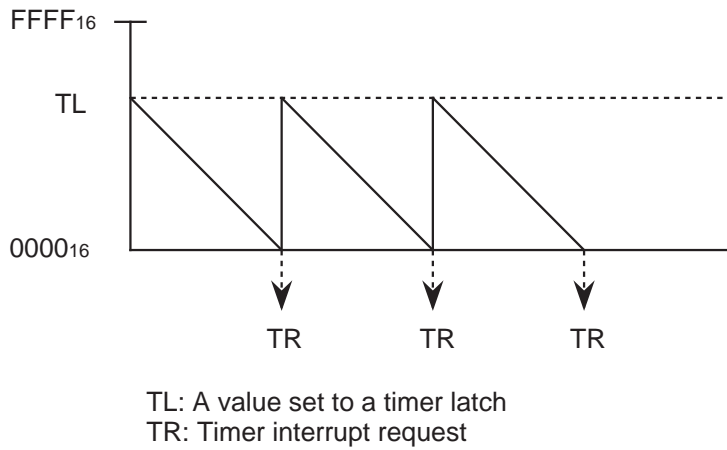
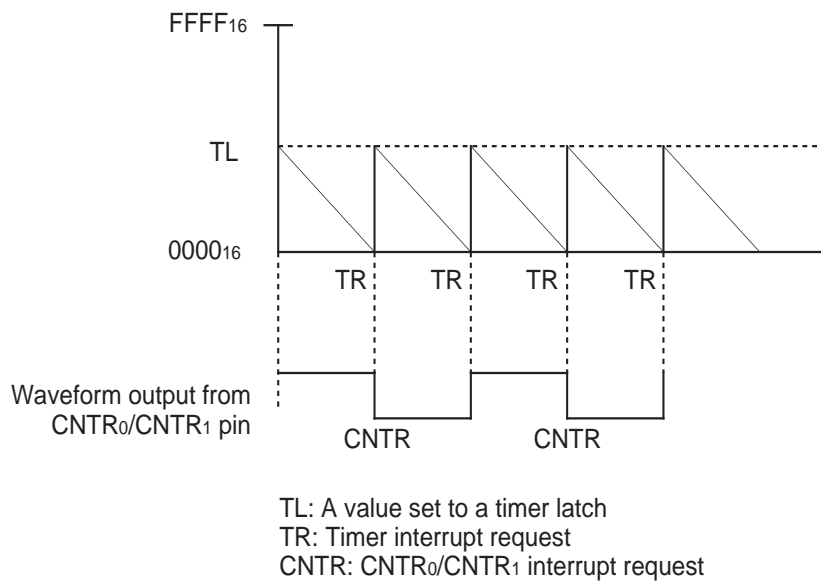


Fig. 19. Timing chart of Timer•Event counter mode



This example's condition:  
CNTR0/CNTR1 active edge switch bit "0":  
⇒ output starts with "H" level, interrupt at falling edge

Fig. 20. Timing chart of Pulse output mode

# HARDWARE

## FUNCTIONAL DESCRIPTION

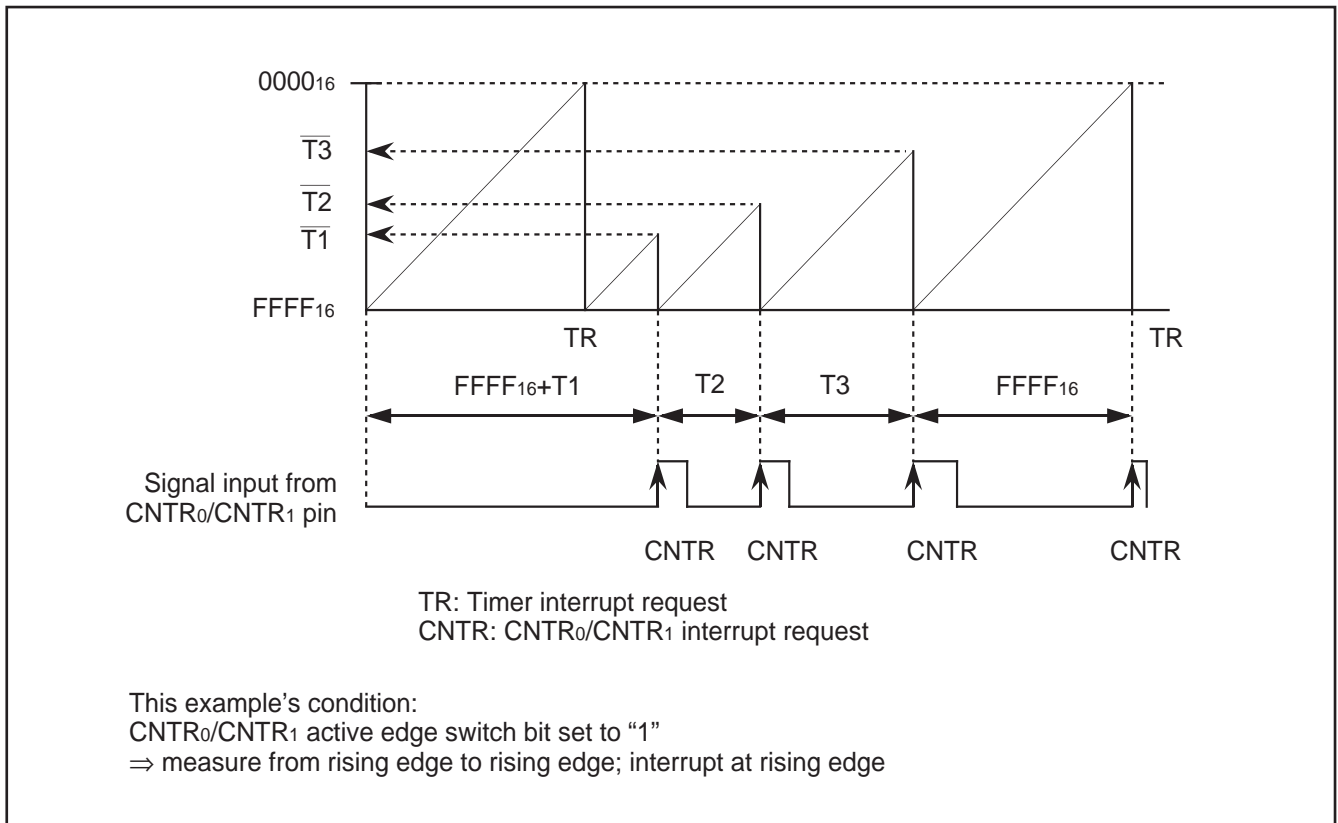


Fig. 21. Timing chart of Pulse period measurement mode

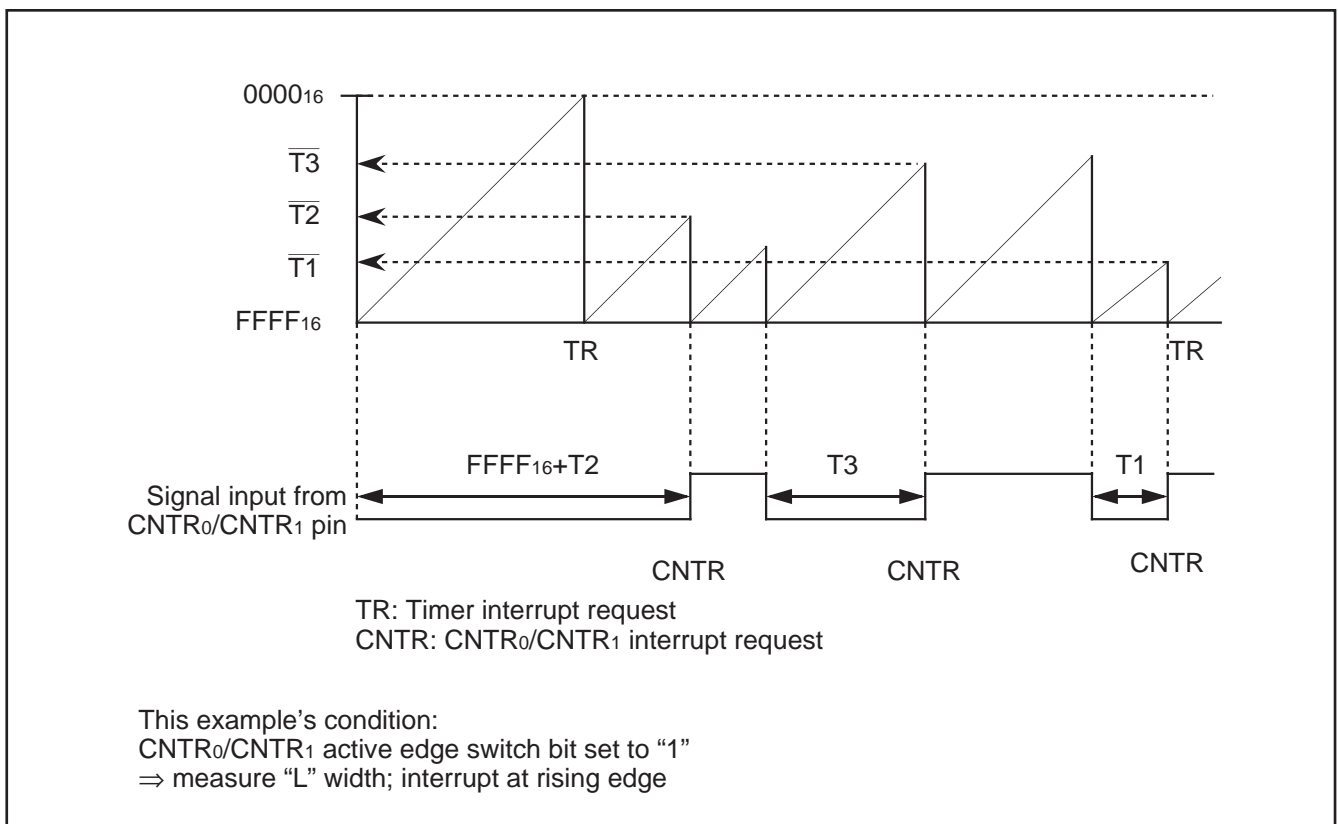
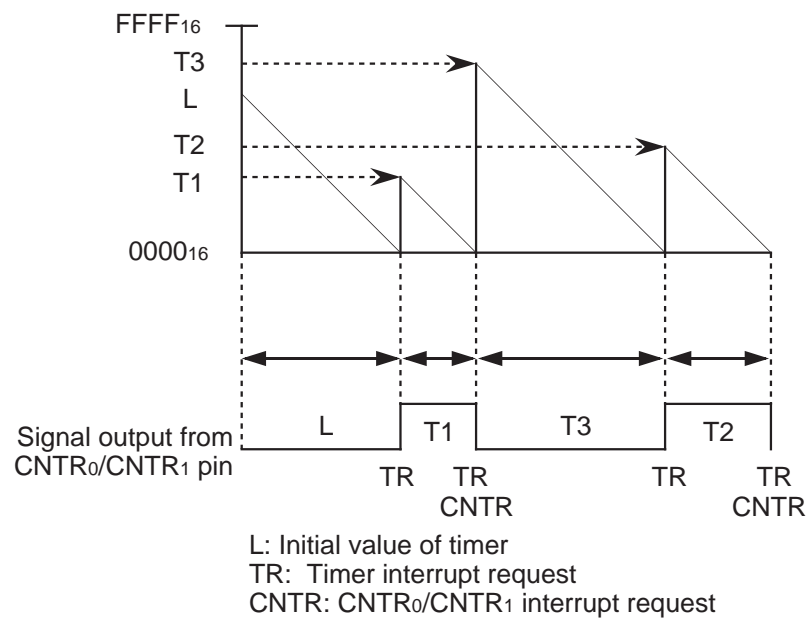


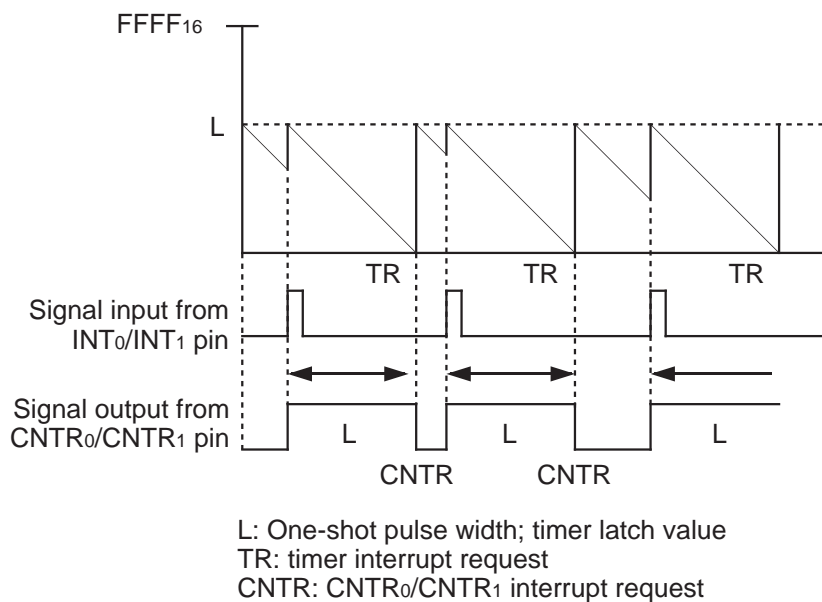
Fig. 22. Timing chart of Pulse width measurement mode





This example's condition:  
CNTR<sub>0</sub>/CNTR<sub>1</sub> active edge switch bit set to "0"  
⇒ output starts with "L" level; interrupt at falling edge

Fig. 23. Timing chart of Programmable waveform generating mode



This example's condition:  
CNTR<sub>0</sub>/CNTR<sub>1</sub> active edge switch bit set to "0"  
⇒ output a "H" pulse; interrupt at falling edge

Fig. 24. Timing chart of Programmable one-shot generating mode

# HARDWARE

## FUNCTIONAL DESCRIPTION

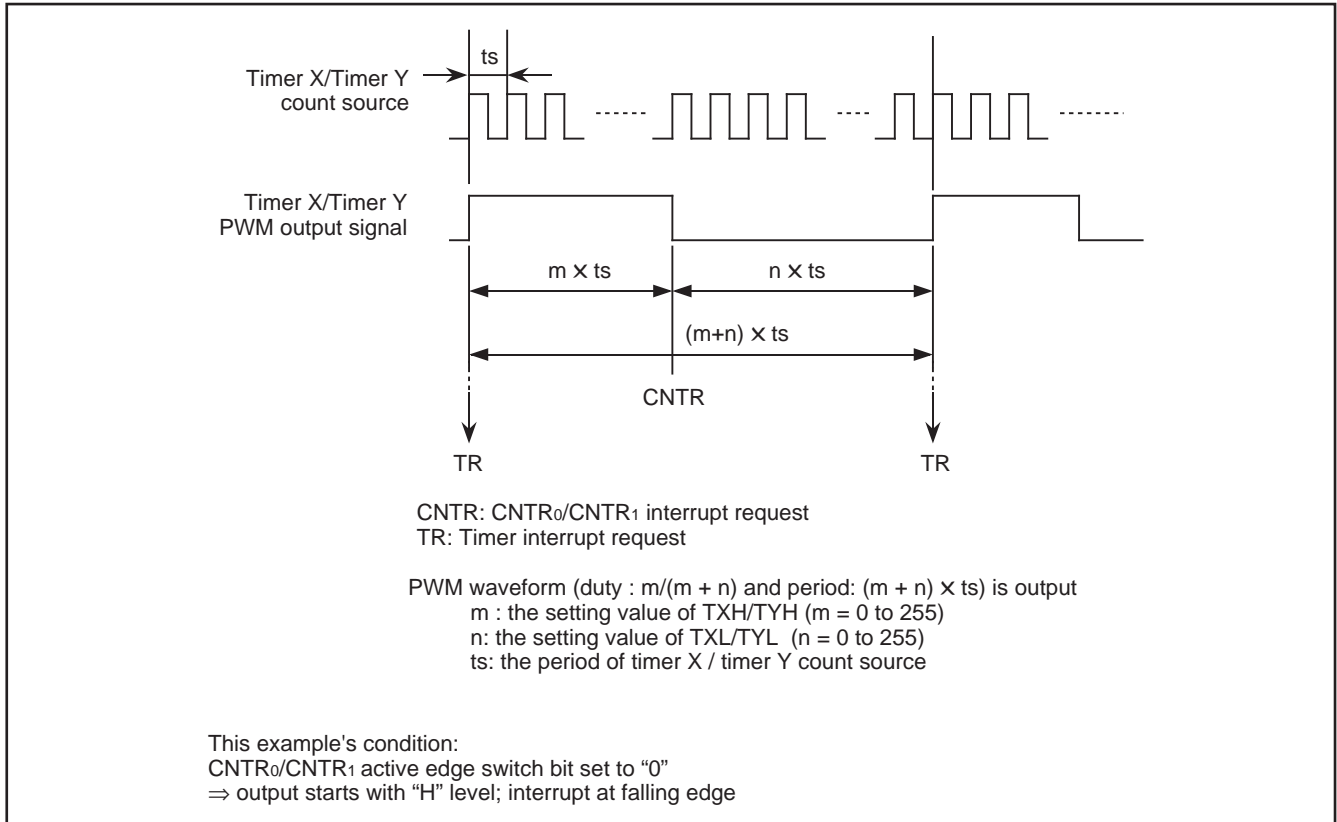


Fig. 25. Timing chart of PWM mode

### ●Timer 1, Timer 2, Timer 3

Timer 1 to 3 are 8-bit timers for which the count source can be selected through timer 123 mode register.

#### (1) Timer 2 write control

Timer 2 write control bit (b2) of timer 123 mode register allows to select whether a value written to timer 2 is written to timer latch and timer synchronously or to the timer latch only.

If only the timer latch is written to, the value is set only to the reload-latch by writing a value to the timer address at that time. The content of timer is reloaded with the next underflow. Usually writing operation to the timer latch and timer synchronously is selected. And a value is written to the timer latch and timer synchronously when a value is written to the timer address.

If only the timer latch is written to, it may occur that the value set to the counter is not constant, when the timing with which the reload-latch is written to and the underflow timing is nearly the same.

#### (2) Timer 2 output control

When timer 2 output (TOUT) is enabled, inverted signals are output from TOUT pin each time timer 2 has underflow. For this reason, set the double-function port of TOUT pin to output mode.

#### ■Precautions on timers 1 to 3

When the count source for timer 1 to 3 is switched, it may occur that short pulses are generated in count signals and that the timer count value shows big changes. When timer 1 output is selected as timer 2 or timer 3 count source, short pulses are generated to signals output from timer 1 through writing timer 1. Due to that, the count values for timer 2 and 3 may change very often.

Therefore, when the count sources for timer 1 to 3 are set, set the values in order starting from timer 1.

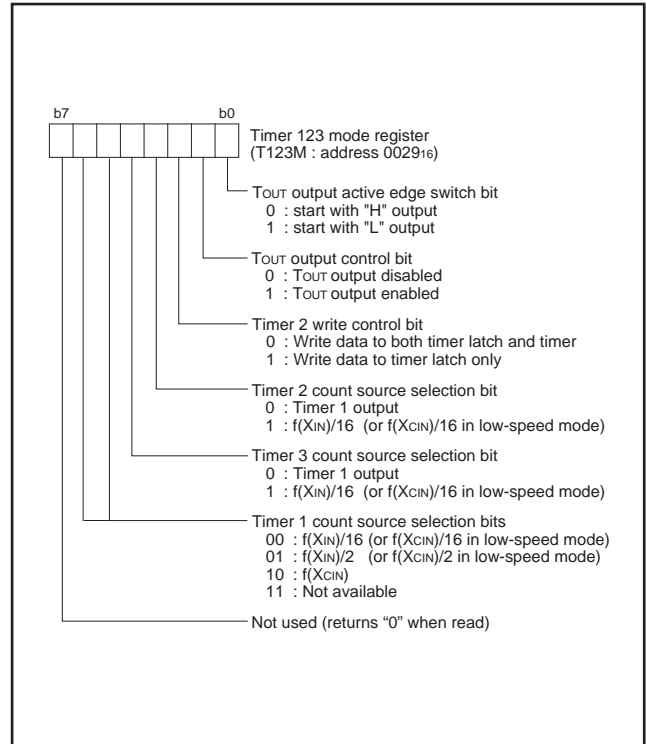


Fig. 26. Structure of Timer 123 mode register

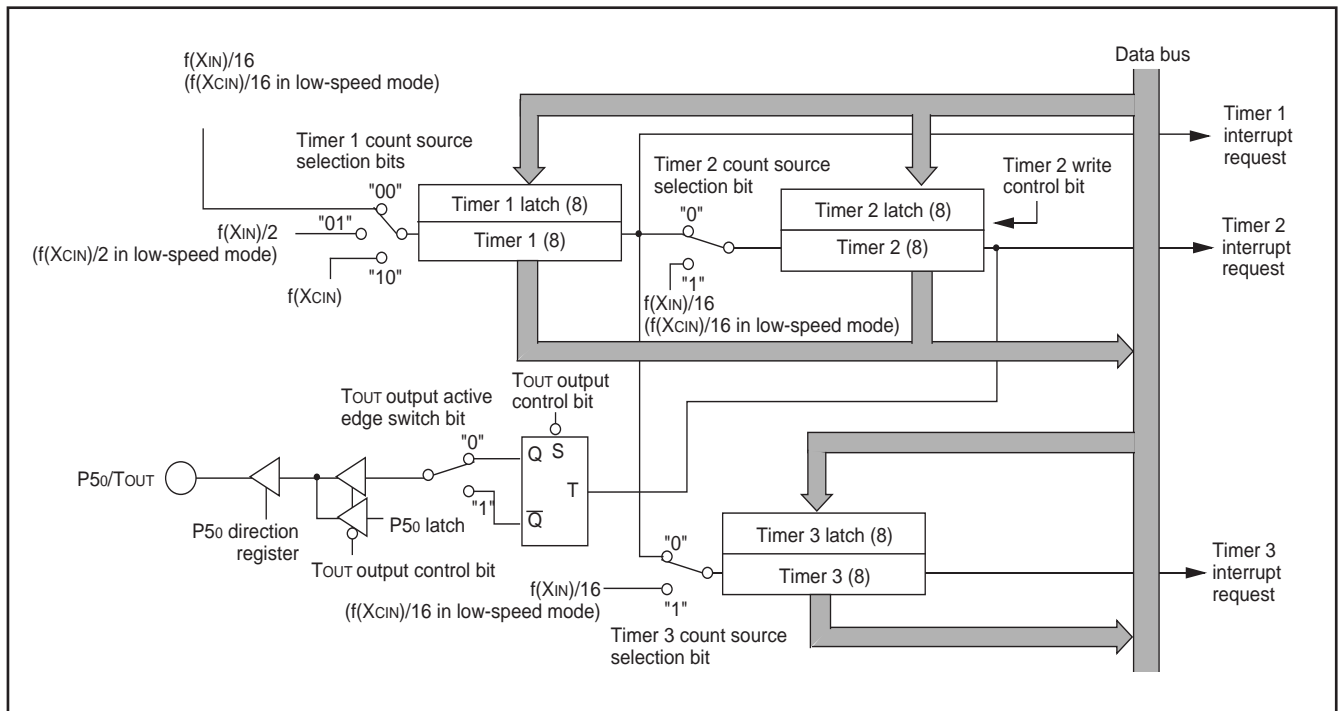


Fig. 27. Block diagram of Timer

### ●Real time output port

The 3807 group has two on-chip sets of real time output ports (RTP). The two sets of real time output ports consist of two 16-bit timers A and B and eight 8-bit real time port registers. Synchronous to the reloading of timers A and B, the real time port register values are output from ports P82 to P87, P30 and P31. The real time port registers consist of 8-bit register 0 to 7. Each port with its corresponding bits is shown in figure 26.

Timer A and timer B have each two 16-bit timer latches. Figure 28 shows the real time port block diagram and figure 29 and 30 show the structure of the real time port control registers 0 to 3.

There are four operating modes for real time ports which are: 8 repeated load mode, 6 repeated load mode, 5 repeated load mode and one-shot pulse generating mode. Each operating mode can be set for timer A and timer B separately. However, switch modes during timer count stop.

#### (1) 8 repeated load mode

The output operation for each value of the real time port registers 7 to 0 is performed repeatedly in association with an alternate underflow of the corresponding timer latch 1 or 0. The real time port output pointer changes in sequence as a cycle of 8 repeated load operations as "7, 6, 5, 4, 3, 2, 1, 0, 7, 6, 5, 4, 3, 2, 1, 0, 7, 6, 5, ...."

The initial value at the generation of a start trigger can be specified by setting a value in the output pointer. Figure 31 shows a timing chart of 8 repeated load mode.

#### (2) 6 repeated load mode

The output operation for each value of real time port registers 5 to 0 is performed repeatedly in association with an alternate underflow of the corresponding timer latch 1 to 0. The real time port output pointer changes in sequence as a cycle of 6 repeated load operations as "5, 4, 3, 2, 1, 0, 5, 4, 3, 2, 1, 0, 5, 4, ...."

The initial value at the generation of a start trigger can be specified by setting a value in the output pointer. Figure 32 shows a timing chart of the 6 repeated load mode.

#### (3) 5 repeated load mode

The output operation for each value of real time port registers 4 to 0 is performed repeatedly in association with an alternate underflow of the corresponding timer latch 1 or 0. The real time port output pointer changes in sequence as a cycle of 5 repeated load operations as "4, 3, 2, 1, 0, 4, 3, 2, 1, 0, 4, 3, 2, 1, ...." The initial value at the generation of a start trigger can be specified by setting a value in the output pointer. Figure 33 shows a timing chart of the 5 repeated load mode.

#### (4) One-shot pulse generation mode

The output operation for each value of real time port registers 2 to 0 is performed only once in association with trigger generation and an underflow of timer latch 1 or 0. After a trigger is generated, the value of real time port register 1 is output from the real time output port and the output pointer value becomes "000<sub>2</sub>". At each underflow of the timer, the each value of real time port registers 0 and 2 is output in ascending sequence, then the operation is completed.

After completion of the operation, the value of real time port register 2 is continuously output from the real time output port and the output pointer value continues to be "001<sub>2</sub>" until the next start trigger is

generated. In this condition, the real time port function is in the wait status.

When this mode is selected, the pointer value is not changed by writing a value into the output pointer. If external trigger is specified as trigger selection when this mode is selected, a rising and falling double edge trigger is generated regardless of the contents of the INT4 interrupt source bit (b7) of the interrupt edge selection register.

Figure 34 shows a timing chart of the one-shot pulse generation mode.

#### (5) Selection of timer interrupt mode

The timer is a count-down system. The contents of the timer latch are reloaded by the count pulse subsequent to the moment when the contents of the counter becomes "0000<sub>16</sub>". At the same time, the interrupt request bit corresponding to each timer is set to "1." The interrupt request corresponding to the value of the real time port output pointer can also be controlled. For controlling the interrupt request bit, refer to the item pertaining to the timer interrupt mode selection bit of the real time port control register 1,2 shown in figure 29 and 30.

#### (6) Switch of timer count source

The timer A and the timer B can select the system clock  $\phi$  divided by 2 or 16 as a count source with the timer A, B count source selection bit (b0) of real time port control register 0.

#### [Timer latches]

Each of the timer A and the timer B has two 16-bit timer latches. Data is written into the 8 low-order bits and the 8 high-order bits in this order. When the high-order side has been written, the next latch is automatically specified. The writing pointer changes in sequence as "1, 0, 1, 0, 1, ...." The timer latch to be written first can be specified by setting the timer writing pointer. Data is not written directly into the timer A and the timer B. When reading the contents of the timer, the count value at that point of time is read. Read the high-order side first and then the low-order side. The low-order side value is read with the same timing as that for the high-order side value and held at the timer read latch. The data held state is released by reading the low-order side. At a reload operation of the timer A or the timer B. Timer latch 1 is reloaded as the initial value after a trigger is generated. After that, the timer latch is reloaded in sequence as "0, 1, 0, 1, ...." The timer latch value cannot be read out.

#### [Start trigger]

The operation of the real time port is started by a start trigger. When a start trigger is generated, the value of the real time port register specified by the output pointer (the value of real time port register 1 in the one-shot pulse generation mode) is output from the real time output port.

The value of timer latch 1 is reloaded into the timer A or the timer B and the timer count A, B source stop bit is released, so that the timer count is started.

After that, when the timer underflows, data is transferred from the real port register to the real time output port.

As a start trigger, either internal trigger or external trigger can be selected by the timer A start trigger selection bit (b2) or timer B start trigger selection bit (b5) of real time port control register 0.

When the internal trigger is selected, a start trigger is generated by an input signal of the INT4 pin. The start trigger becomes a falling edge when the INT4 interrupt edge selection bit is "0" and a rising edge when this bit is "1".

When the external trigger is selected in the one-shot pulse generation mode, the start trigger becomes a rising/falling double edge trigger regardless of the contents of the INT4 interrupt edge selection bit.

### [Real time port registers] RTP

The data to be output to real time ports is written into 8 real time port registers 0 to 7. The correspondence between each bit of real time port registers and each port output is as follows :

P31: bit 7 of real time port registers 7 to 0

P30: bit 6 of real time port registers 7 to 0

P87: bit 5 of real time port registers 7 to 0

P86: bit 4 of real time port registers 7 to 0

P85: bit 3 of real time port registers 7 to 0

P84: bit 2 of real time port registers 7 to 0

P83: bit 1 of real time port registers 7 to 0

P82: bit 0 of real time port registers 7 to 0

It can be selected for each bit by real time port control register 3 whether the output of each port is to be used as an ordinary I/O port or a real time port output.

### [Real time port data pointer]

It can be optionally specified by the real time port data pointers A or B and the real time port data pointer A or B switching bit in which real time port register the output data is to be set or from which real time port register the data output is to be started.

When writing output data into the real time port register, set the real time port data pointer A, B switch bit to "0" (select the R/W pointer) and also write a value into the 3 bits of the real time port data pointers A, B. With this, the real time port register for writing will be specified. After that, when a value is written into the real time port register (address 002A16), the data is written into the specified real time port register and also the R/W pointer value is automatically decreased by 1. Then writing data is enabled into the next real time port register.

A value of "0002" to "1112" can be set into the R/W pointer regardless of the operating mode specified by the timer A, B operating mode selection bit, and the R/W pointer value is automatically decreased by 1 by writing data into the real time port register. However, when a value becomes "0002", the R/W pointer value is decreased by 1 in the numeral range of stages to be used in each operating mode unless the R/W pointer is set again at the subsequent write operation to the real time port register. When "1112 (=7)" is set in the R/W pointer, the R/W pointer operation in each selected mode is as follows :

•During 8 repeated load mode 7→6→5→4→3→2→1→0→7→6→5...

•During 6 repeated load mode 7→6→5→4→3→2→1→0→5→4→3...

•During 5 repeated load mode 7→6→5→4→3→2→1→0→4→3→2...

•During one-shot pulse generation mode

7→6→5→4→3→2→1→0→2→1→0...

When reading the real time port register, set the real time port data pointer A, B switch bit to "0" (select the R/W pointer) and also writing a value into the 3 bits of the real time port data pointer A, B to specify the real time port register for reading. After that, the value of the

specified real time port register can be read by reading the real time port register (address 002A16). In this case, however, the R/W pointer value is not counted down automatically. Accordingly, to read another real time port register, rewrite the R/W pointer beforehand.

To specify a read port register to be output to the real time output port, set the real time port data pointer A, B switch bit to "1" (select an output pointer) and also set a value in the 3 bits of the real time port data pointer A or B.

When a start trigger is generated, data is output beginning with the real time port register set in the output pointer and the output pointer value is automatically decreased by 1.

At each underflow of the timer A or timer B, the output pointer value is automatically decreased by 1. Regarding the case of the one-shot pulse generation mode, however, refer to the item pertaining to the one-shot pulse generation mode.

When the real time port data pointer A to B has been read, only the output pointer can be read.

### ■Notes regarding all modes

•When the trigger is generated again during timer count operation, the operation is started from the beginning. In this case, put an interval of 3 cycles or more between the generation of a trigger and the generation of the next trigger. If the generation of the next trigger occurs almost concurrently with the underflow timing of the timer, the next real time output may not be performed normally.

•To stop the timer count after generation of a start trigger, write "1" in the timer A, B count source stop bit of real time port control register 0 at an interval of 3 cycles or more of the timer count source.

•To change the contents of the real time port data pointer A, B switch bit, the real time port data pointer must be specified simultaneously. Therefore, use the LDM/STA instruction instead of the SEB/CLB instruction.

•If the timer A, B count source stop bit is changed ("1"→"0") by a start trigger between the read operation and the write operation of a read-modify-write instruction such as the SEB instruction which is used in real time port control register 0, the timer count will stop, having an effect on the real time output.

An maximum interval of 2 cycles of the count source is required before the timer A, B count source stop bit is cleared to "0" which indicates the count operation state after a start trigger is generated regardless of whether the start trigger is an internal trigger or an external trigger.

Accordingly, do not use the read-modify-write instruction for real time port control register 0 in this period. If a write operation for real time port control register 0 with any purpose other than stopping the timer count is performed concurrently with the generation of a start trigger, be sure to use such an instruction for writing "0" into the timer A, B count source stop bit as the LDM/STA instruction.

Even if "0" is written into the timer A, B count source stop bit, the timer count remains in the stop state without change.

•When the timing for writing to the high-order side reload latch is almost equal to the underflow timing, an undesirable value may be set in the timer A or timer B.

•If the real time output port is selected by real time port control register 3 after resetting, "L" is output from this pin until a start trigger is generated.

# HARDWARE

## FUNCTIONAL DESCRIPTION

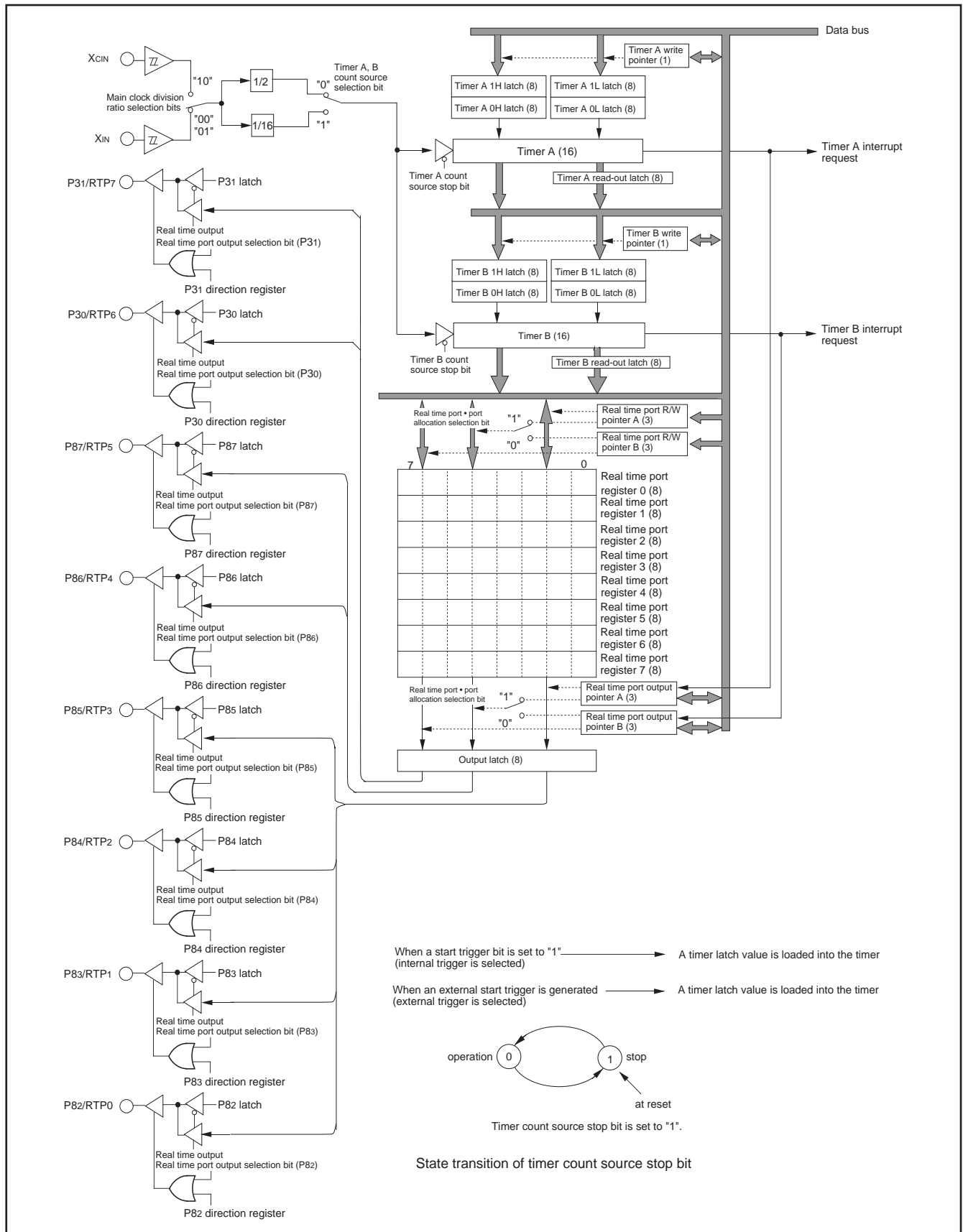


Fig. 28. Block diagram of Real time output port

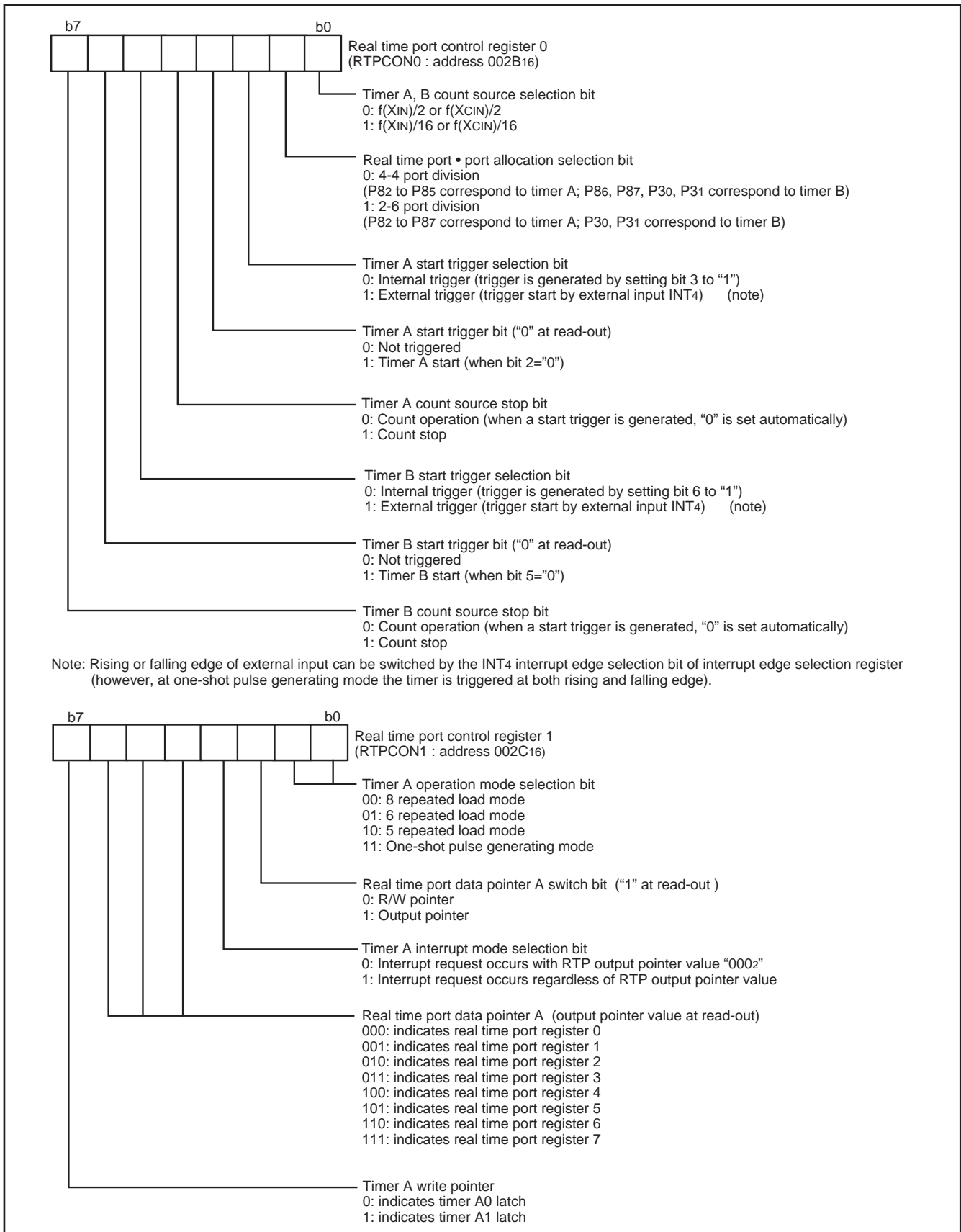


Fig. 29. Structure of Real time output port related register (1)

# HARDWARE

## FUNCTIONAL DESCRIPTION

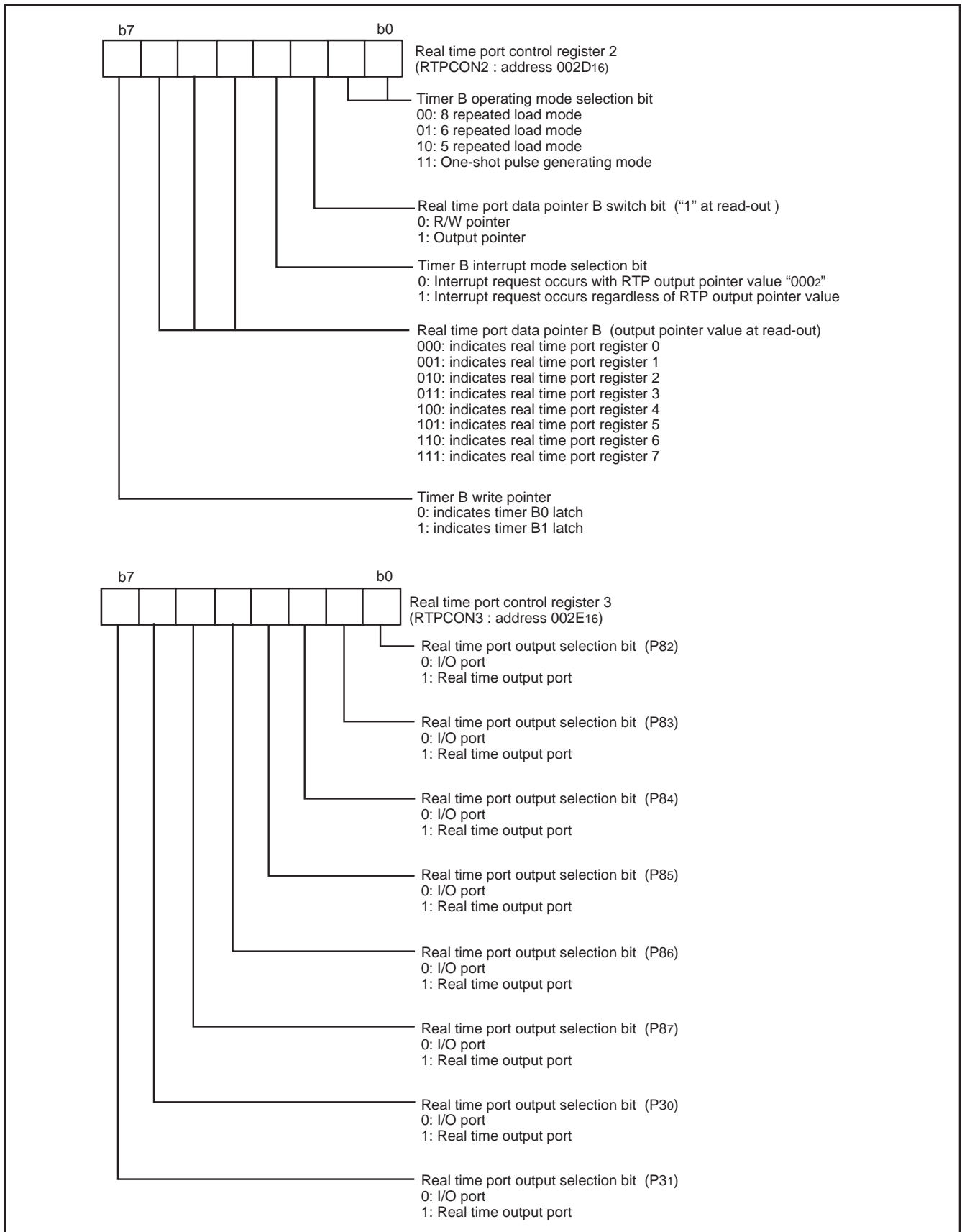
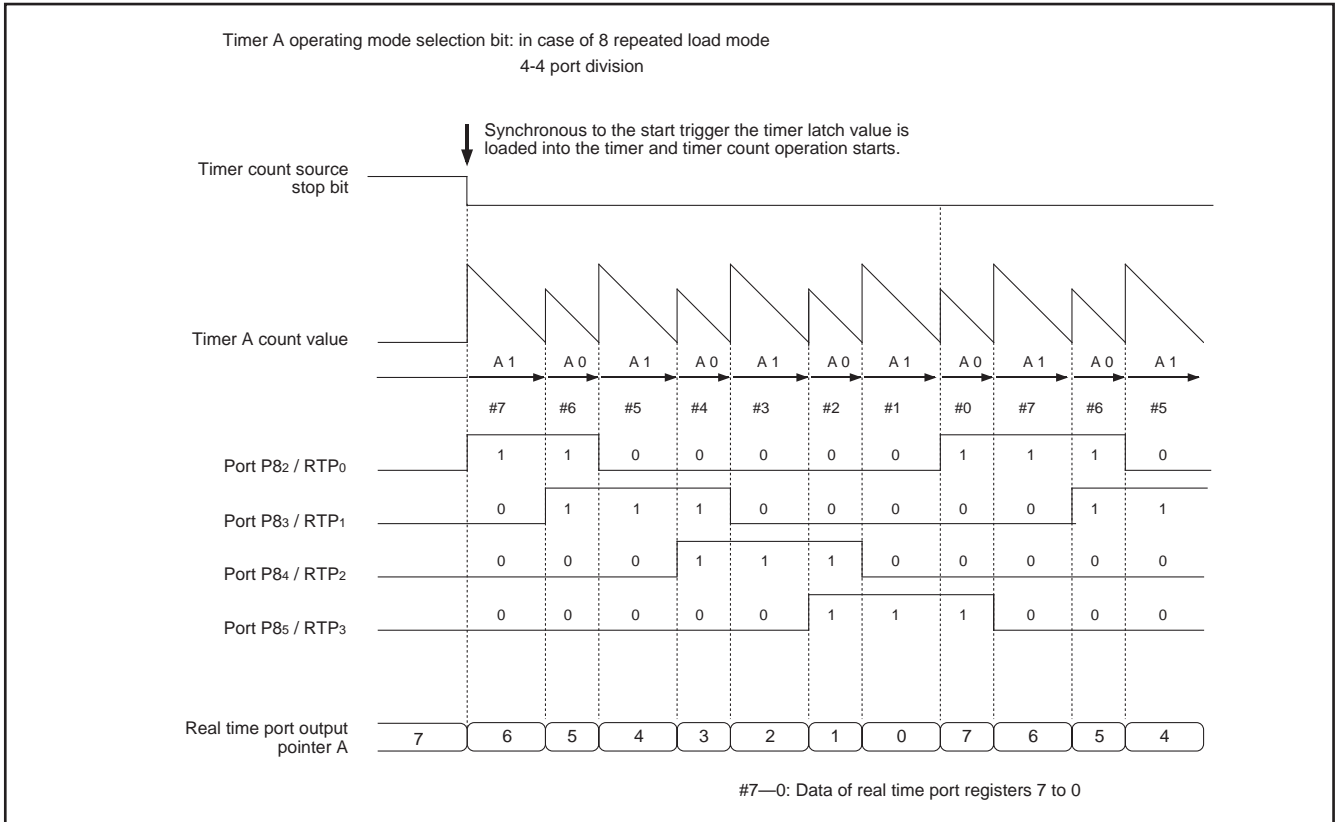
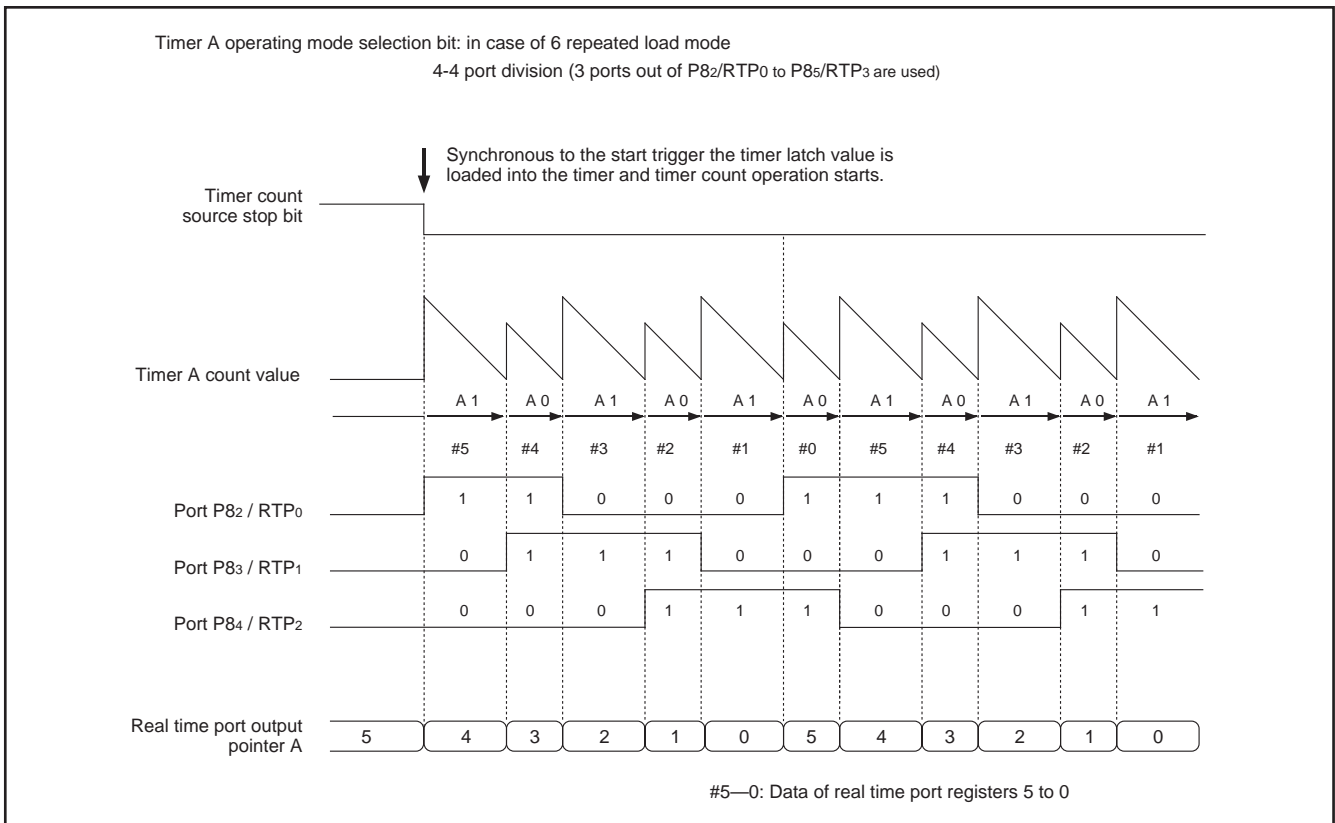


Fig. 30. Structure of Real time output port related register (2)





**Fig. 31. 8 repeated load mode operation**



**Fig. 32. 6 repeated load mode operation**

# HARDWARE

## FUNCTIONAL DESCRIPTION

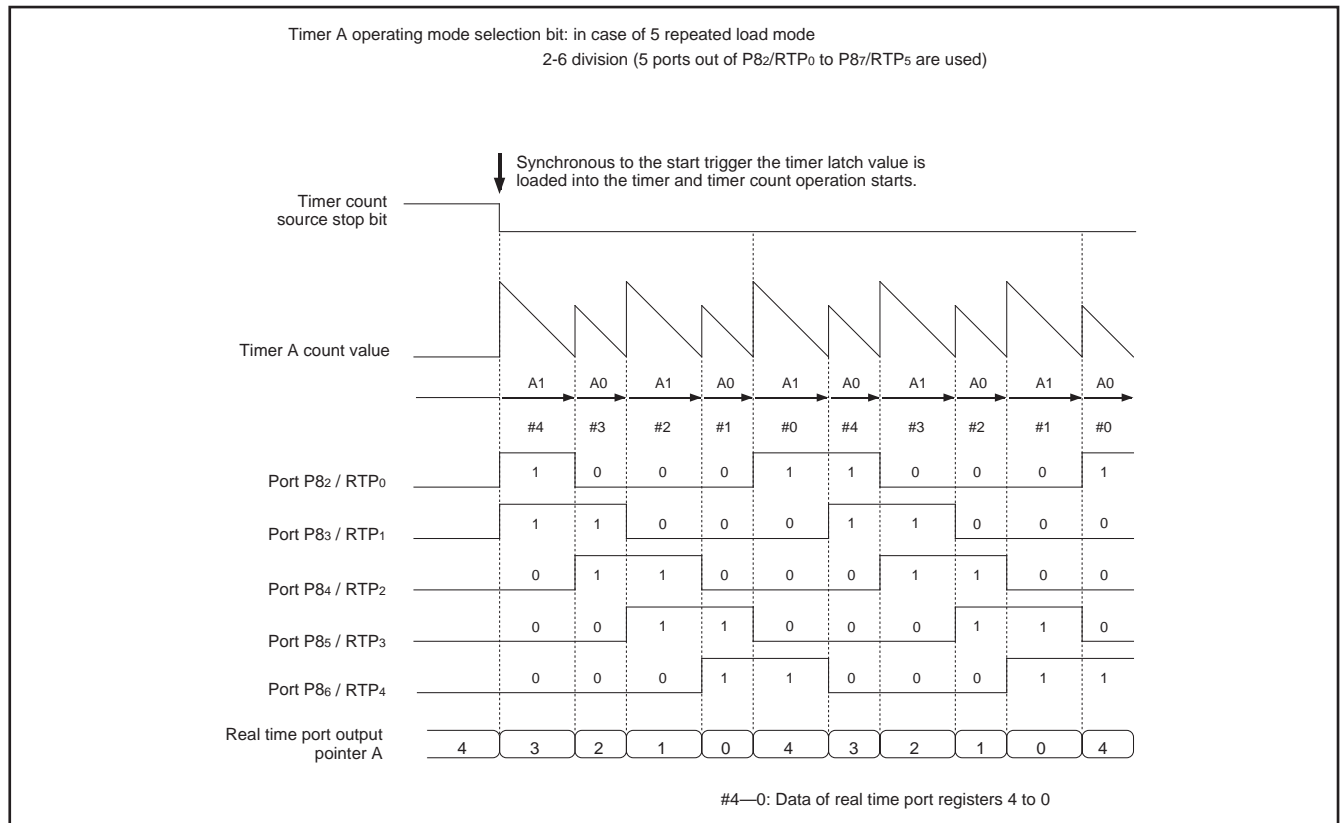


Fig. 33. 5 repeated load mode operation

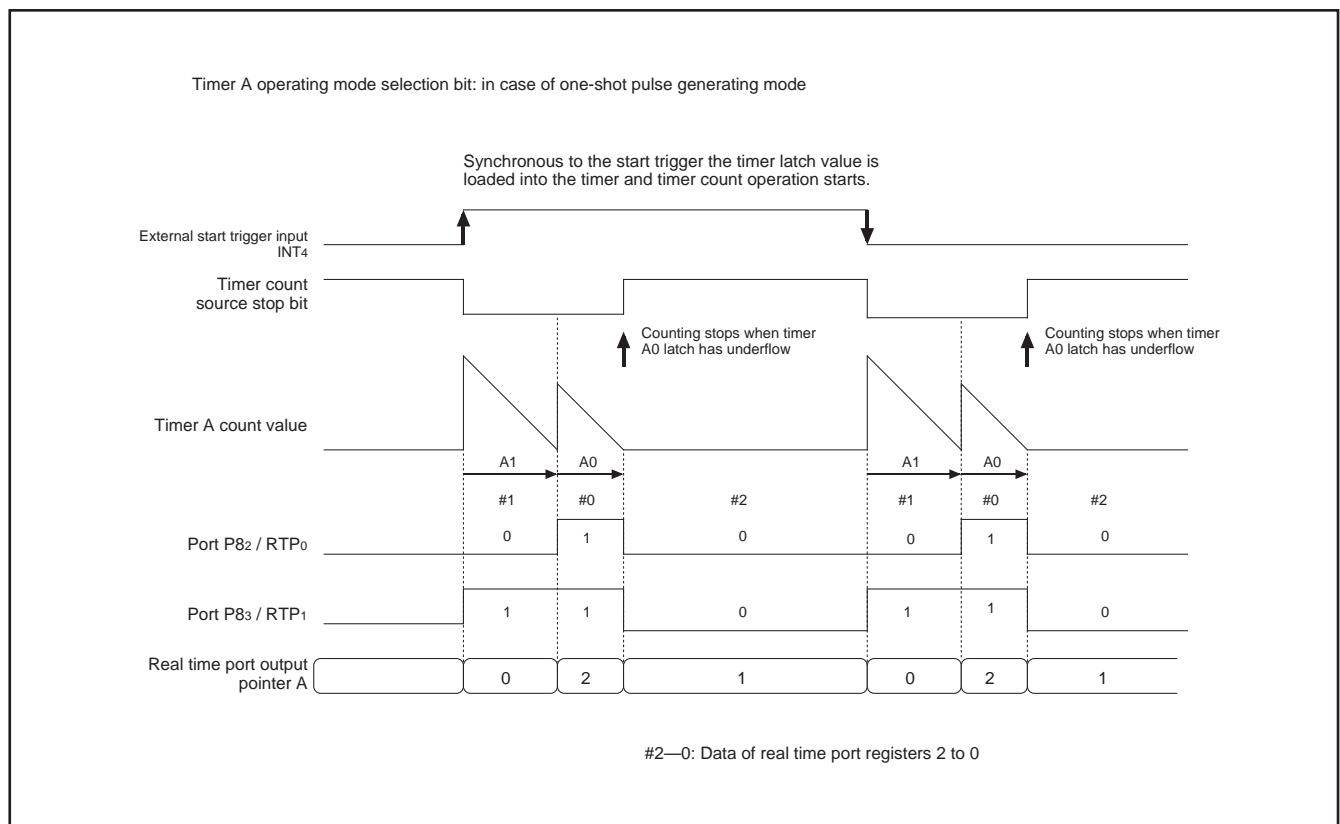


Fig. 34. One-shot pulse generating mode operation

### Serial I/O

#### ●Serial I/O1

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

#### (1) Clock Synchronous Serial I/O Mode

Clock synchronous serial I/O1 mode can be selected by setting the serial I/O1 mode selection bit (b6) of the serial I/O1 control register to "1." For clock synchronous serial I/O, the transmitter and the receiver must use the same clock for serial I/O1 operation. If an internal clock is used, transmit/receive is started by a write signal to the Transmit/Receive buffer register (TB/RB) (address:0018<sub>16</sub>).

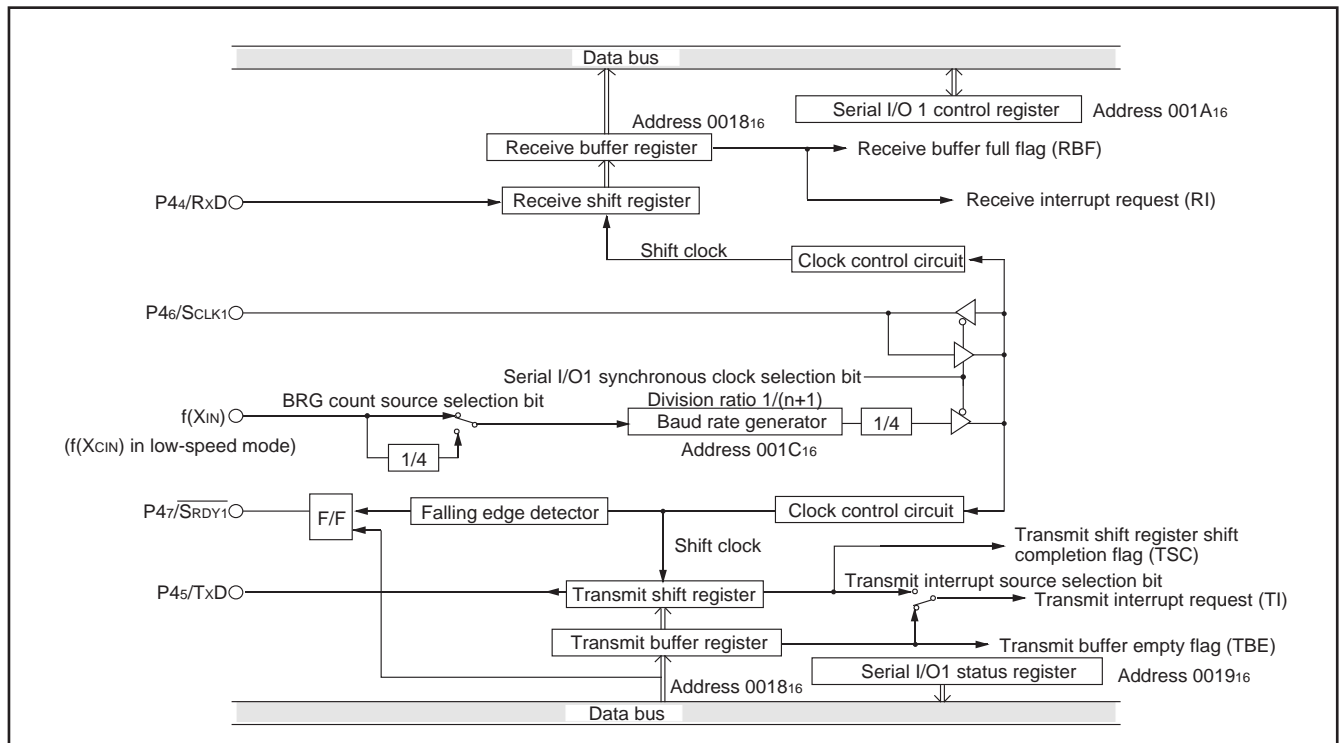


Fig. 35. Block diagram of clock synchronous serial I/O1

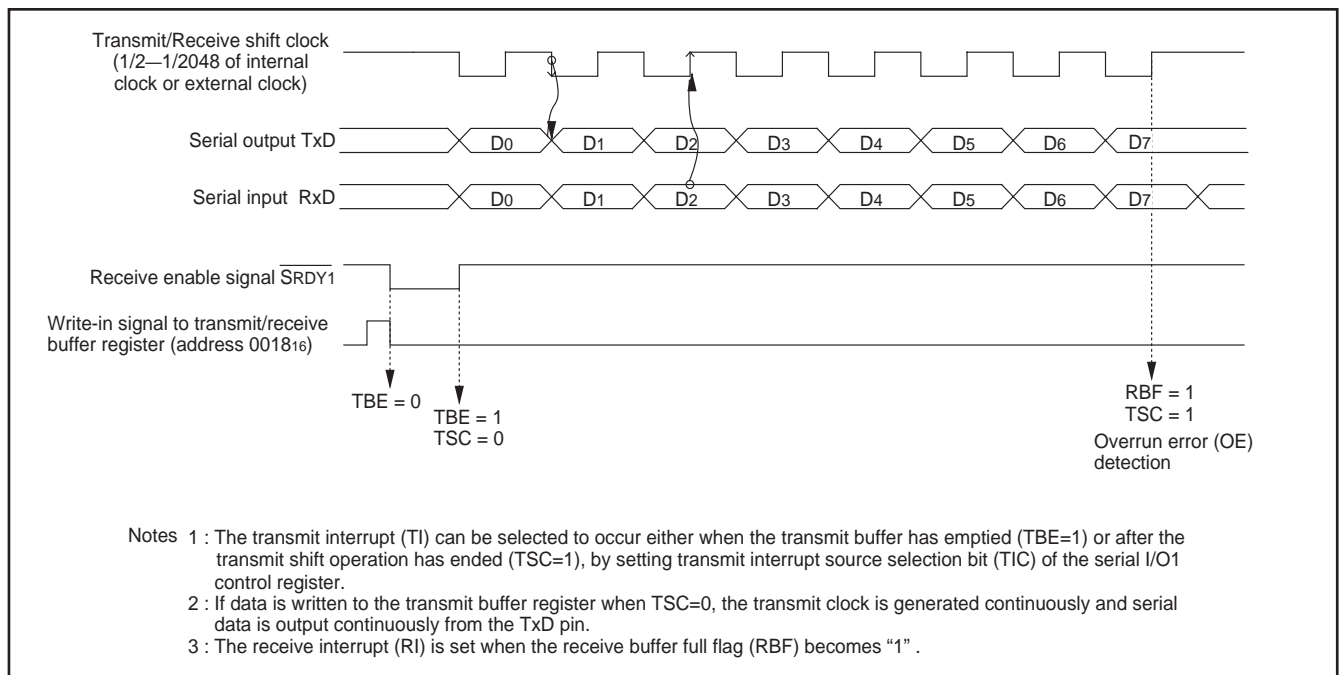


Fig. 36. Operation of clock synchronous serial I/O1 function

# HARDWARE

## FUNCTIONAL DESCRIPTION

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

Asynchronous serial I/O mode (UART) can be selected by clearing the Serial I/O mode selection bit (b6) 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 (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.

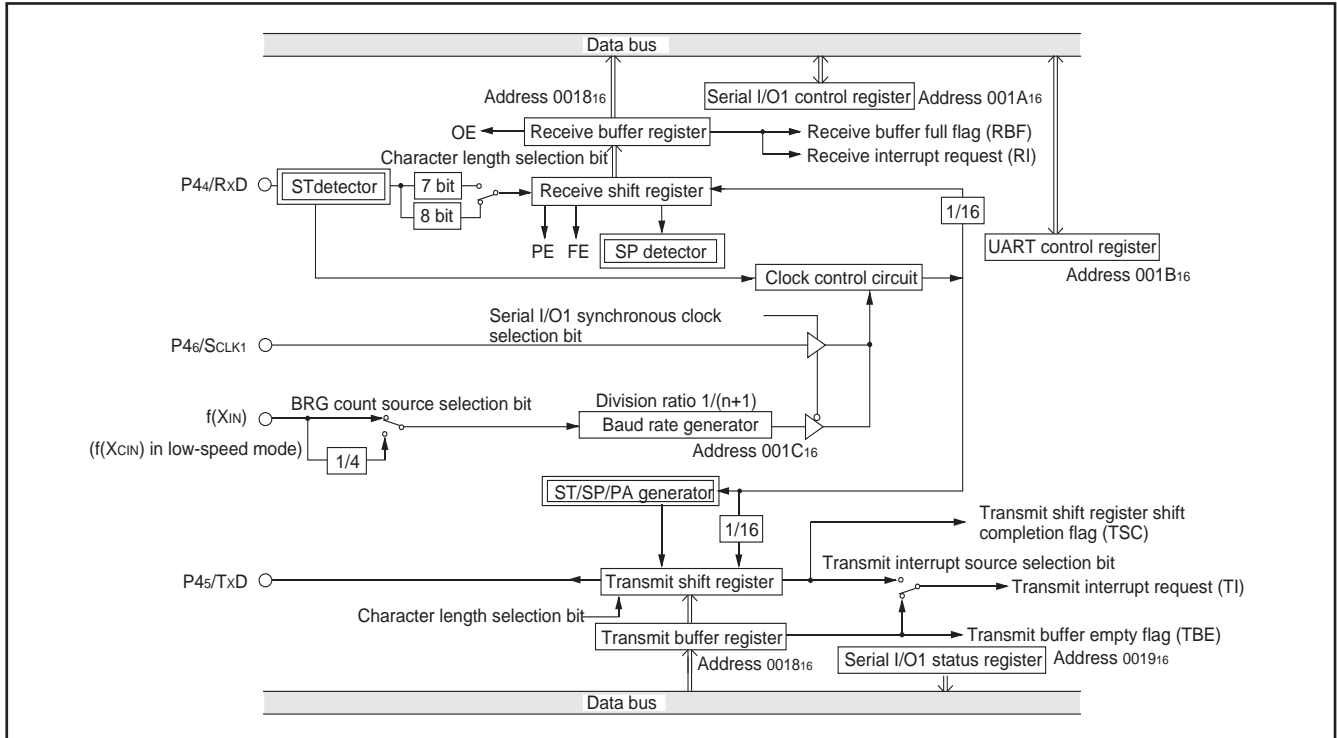


Fig. 37. Block diagram of UART serial I/O1

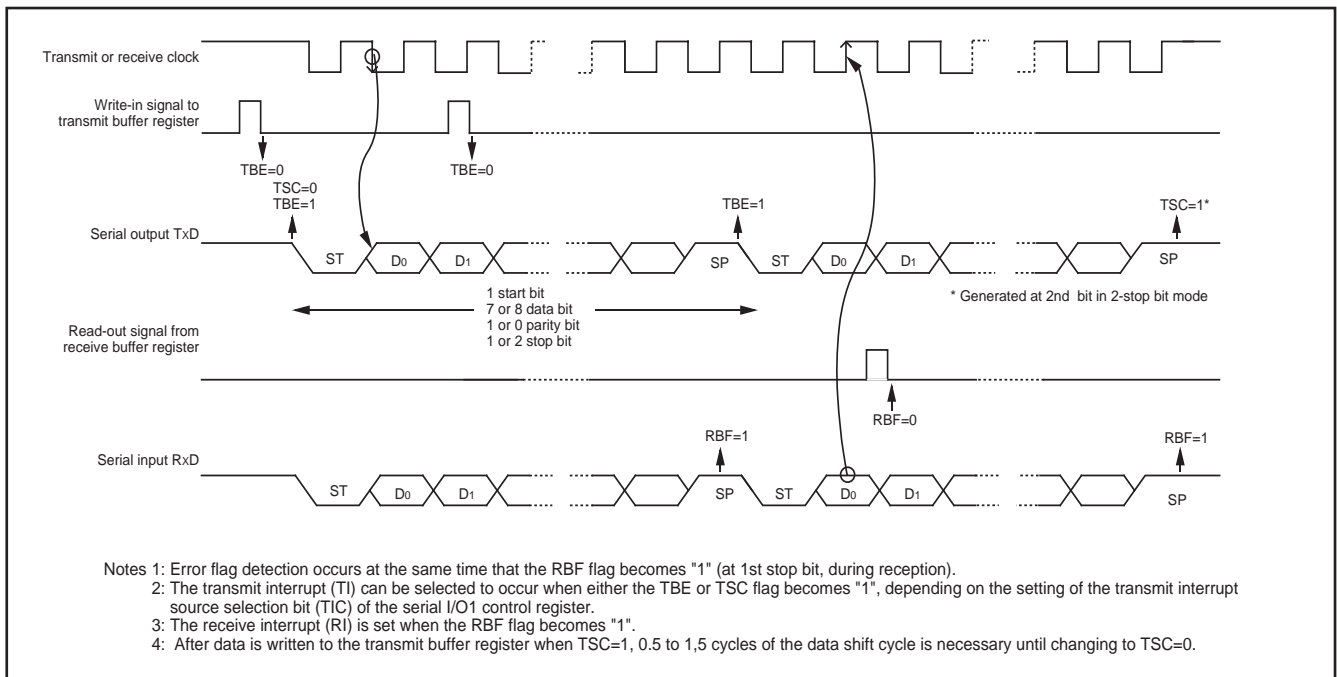


Fig. 38. Operation of UART serial I/O1 function

### [Transmit Buffer Register/Receive Buffer Register] TB/RB (0018<sub>16</sub>)

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

### [Serial I/O 1 Status Register] SIO1STS (0019<sub>16</sub>)

The read-only serial I/O1 status register consists of seven flags (b0 to b6) which indicate the operating status of the serial I/O1 function and various errors. Three of the flags (b4 to b6) are only valid in UART mode. The receive buffer full flag (b1) is cleared to "0" when the receive buffer is read.

The error detection is performed at the same time data is transferred from the receive shift register to the receive buffer register, and the receive buffer full flag is set. A writing to the serial I/O1 status register clears all the error flags OE, PE, FE, and SE (b3 to b6, respectively). Writing "0" to the serial I/O1 enable bit (SIOE : b7 of the serial I/O1 control register) also clears all the status flags, including the error flags.

All bits of the serial I/O1 status register are initialized to "0" at reset, but if the transmit enable bit (b4) of the serial I/O1 control register has been set to "1", the transmit shift register shift completion flag (b2) and the transmit buffer empty flag (b0) become "1."

### [Serial I/O1 Control Register] SIO1CON (001A<sub>16</sub>)

The serial I/O1 control register contains eight control bits for serial I/O1 functions.

### [UART Control Register] UARTCON (001B<sub>16</sub>)

The UART control register consists of four control bits (b0 to b3) which are valid when asynchronous serial I/O is selected and set the data format of an data transfer. One bit in this register (b4) is always valid and sets the output structure of the P4<sub>5</sub>/TxD pin.

### [Baud Rate Generator] BRG (001C<sub>16</sub>)

The baud rate generator determines the baud rate for serial transfer. With the 8-bit counter having a reload register 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.

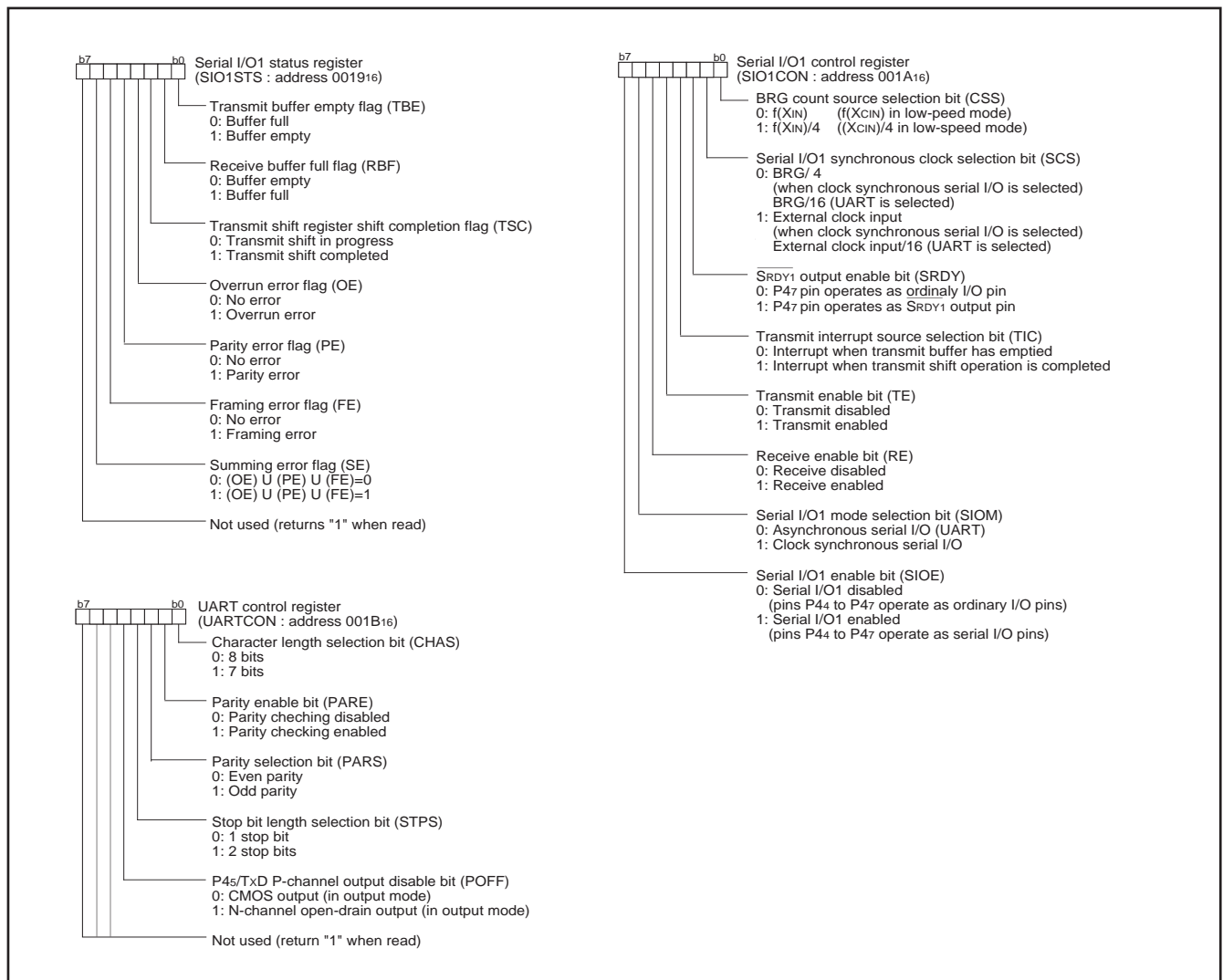


Fig. 39. Structure of serial I/O1 related register

# HARDWARE

## FUNCTIONAL DESCRIPTION

### ●Serial I/O2

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

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

Regarding SOUT2 and SCLK2 being output pins, either CMOS output format or N-channel open-drain output format can be selected by the P71/SOUT2, P72/SCLK2 P-channel output disable bit (b7) of serial I/O2 control register 1.

When the internal clock has been selected, a transfer starts by a write signal to the serial I/O2 register (address 001F16). After completion of data transfer, the level of the SOUT2 pin goes to high impedance automatically but bit 7 of the serial I/O2 control register 2 is not set to "1" automatically.

When the external clock has been selected, the contents of the serial I/O2 register is continuously sifted while transfer clocks are input. Accordingly, control the clock externally. Note that the SOUT2 pin does not go to high impedance after completion of data transfer.

To cause the SOUT2 pin to go to high impedance in the case where the external clock is selected, set bit 7 of the serial I/O2 control register 2 to "1" when SCLK2 is "H" after completion of data transfer. After the next data transfer is started (the transfer clock falls), bit 7 of the serial I/O2 control register 2 is set to "0" and the SOUT2 pin is put into the active state.

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

At transmit operation using the clock synchronous serial I/O, the SCMP2 signal can be output by comparing the state of the transmit pin SOUT2 with the state of the receive pin SIN2 in synchronization with a rise of the transfer clock. If the output level of the SOUT2 pin is equal to the input level to the SIN2 pin, "L" is output from the SCMP2 pin. If not, "H" is output. At this time, an INT2 interrupt request can also be generated. Select a valid edge by bit 2 of the interrupt edge selection register (address 003A16).

#### [Serial I/O2 Control Registers 1, 2] SIO2CON1 / SIO2CON2

The serial I/O2 control registers 1 and 2 are containing various selection bits for serial I/O2 control as shown in Figure 40.

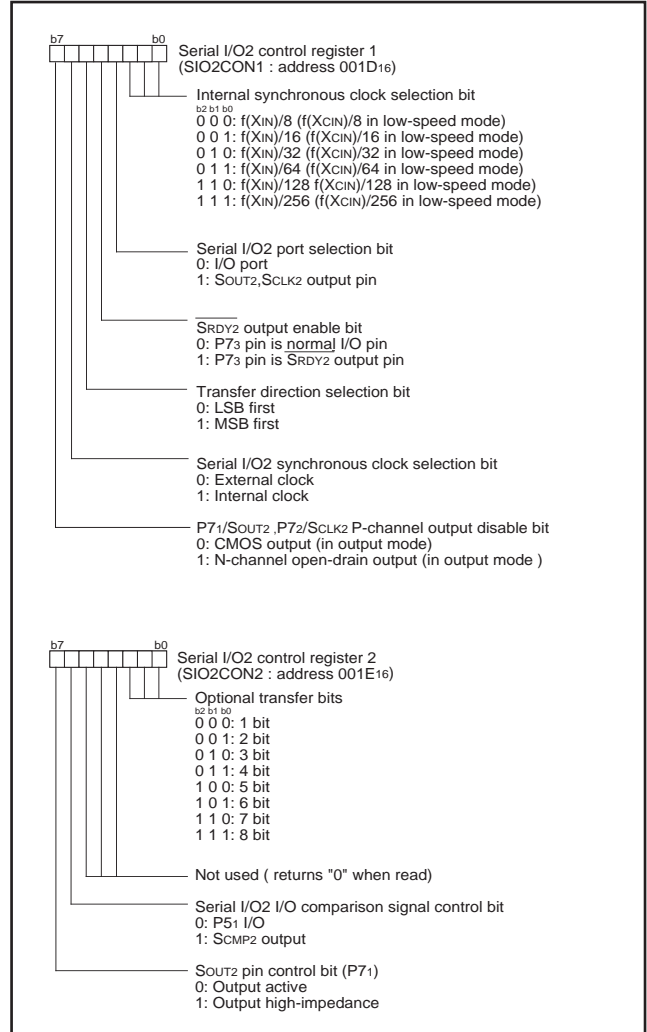


Fig. 40. Structure of Serial I/O2 control registers 1, 2

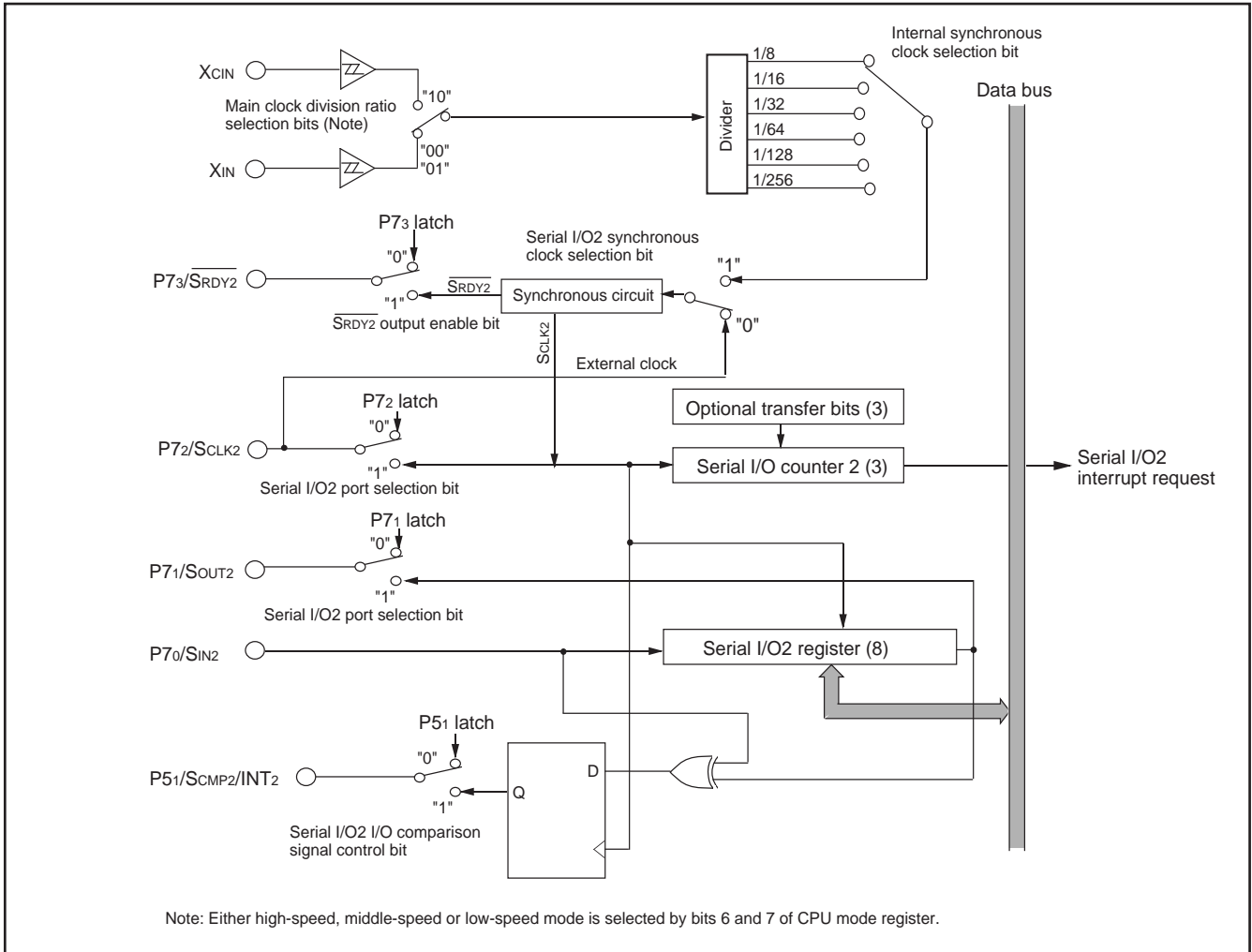


Fig. 41. Block diagram of Serial I/O2

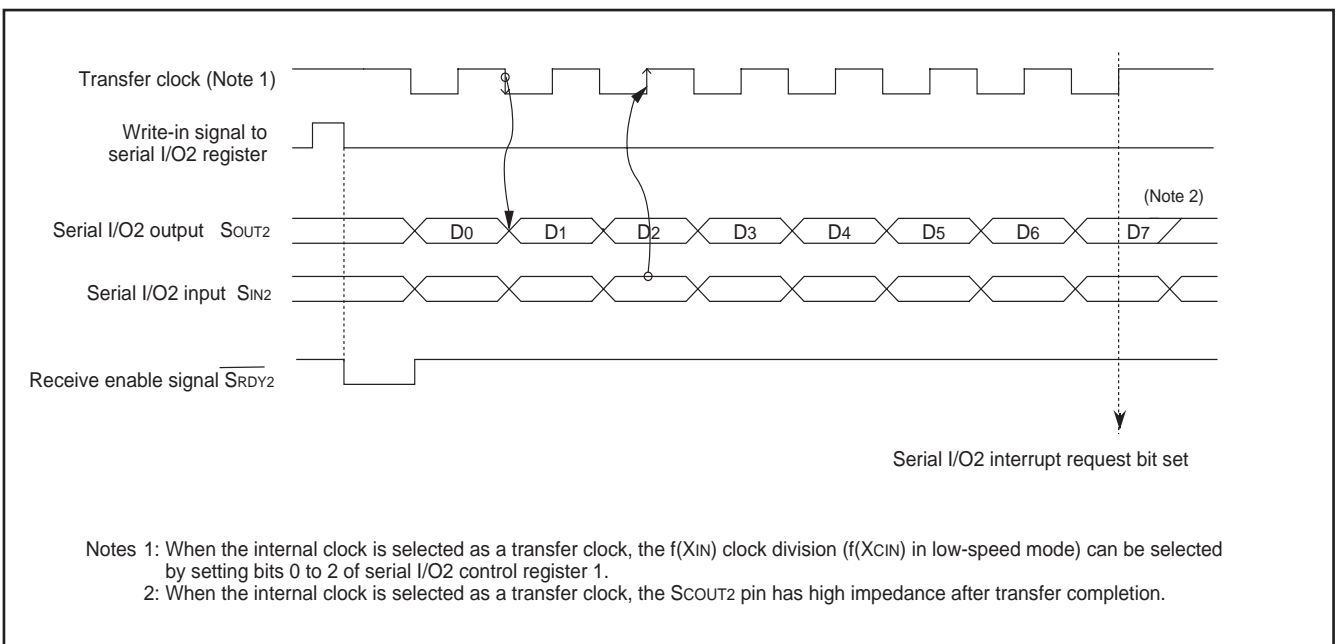


Fig. 42. Timing chart of Serial I/O2

# HARDWARE

## FUNCTIONAL DESCRIPTION

---

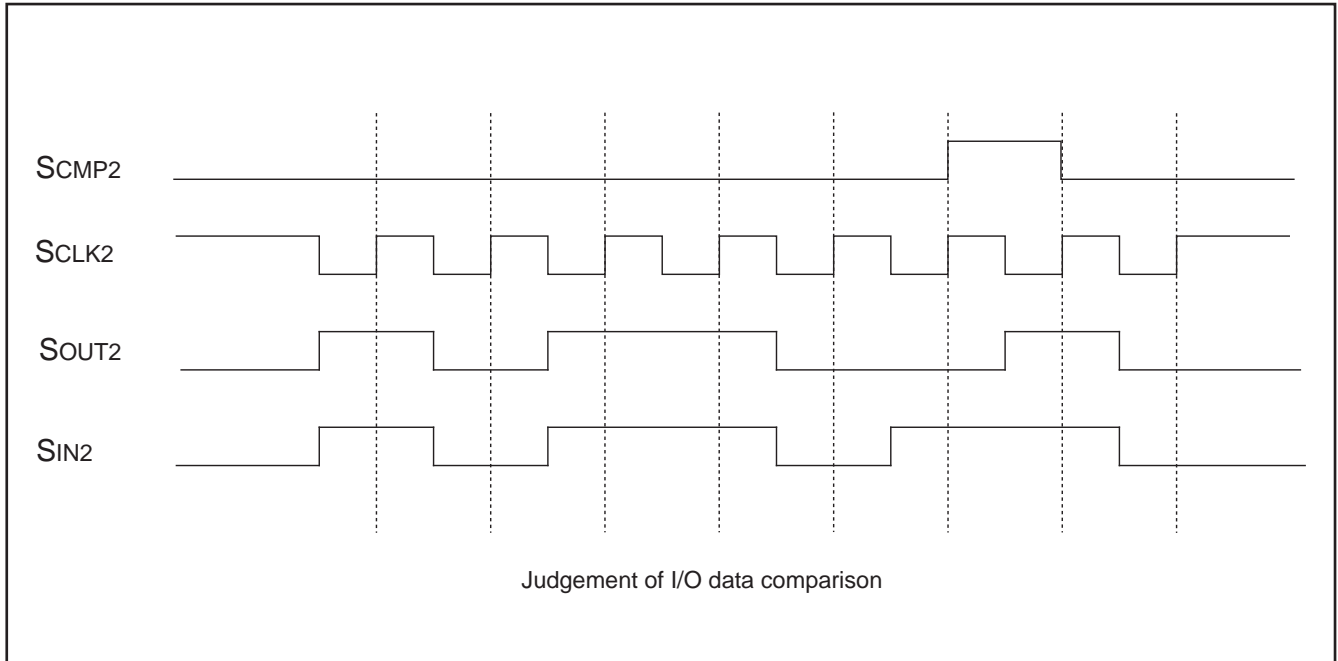


Fig. 43. SCMP2 output operation



### A-D Converter

#### [A-D Conversion Register] AD (address 003516)

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

#### [A-D Control Register] ADCON

The A-D control register controls the A-D conversion process. Bits 0 to 3 of this register select specific analog input pins. Bit 4 signals the completion of an A-D conversion. The value of this bit remains at "0" during an A-D conversion, then changes to "1" when the A-D conversion is completed. Writing "0" to this bit starts the A-D conversion. When bit 6, which is the AD external trigger valid bit, is set to "1", this bit enables A-D conversion at a falling edge of an ADT input. Set ports which is also used as ADT pins to input when using an A-D external trigger. Bit 5 is the ADVREF input switch bit. Writing "1" to this bit, this bit always causes ADVREF connection. Writing "0" to this bit causes ADVREF connection only during A-D conversion and cut off when A-D conversion is completed.

#### [Comparison Voltage Generator]

The comparison voltage generator divides the voltage between AVSS and ADVREF by 256, and outputs the divided voltages.

#### [Channel Selector]

The channel selector selects one of the input ports AN12 to AN0 and inputs it to the comparator.

#### [Comparator and Control Circuit]

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

Note that the comparator is constructed linked to a capacitor, so set

$f(X_{IN})$  to at least 500kHz during A-D conversion. Use a CPU system clock dividing the main clock  $X_{IN}$  as the internal clock  $\phi$ .

#### ■Note

When the A-D external trigger is invalidated by the AD external trigger valid bit, any interrupt request is not generated at a fall of the ADT input. When the AD external trigger valid bit is set to "1" beforehand, A-D conversion is not started by writing "0" into the AD conversion completion bit and "0" is not written into the AD conversion completion bit. Do not set "0" in the AD conversion completion bit concurrently with the timing at which the AD external trigger valid bit is rewritten. Put an interval of at least 50 cycles to more of the internal clock  $\phi$  between a start of A-D conversion and the next start of A-D conversion.

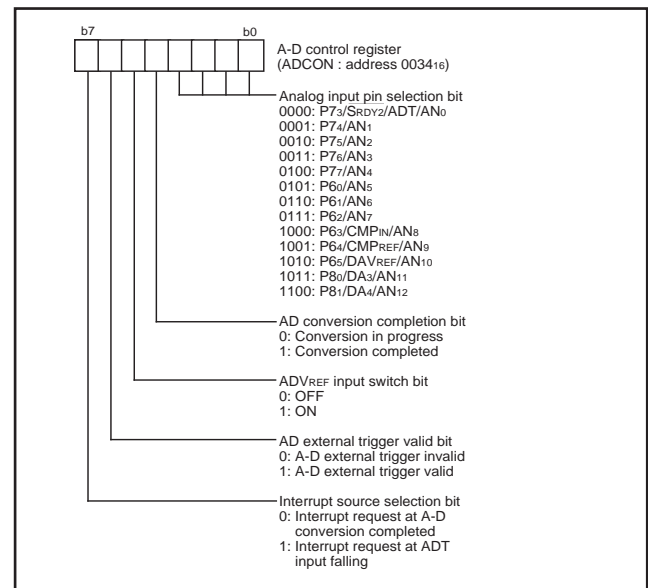


Fig. 44. Structure of A-D control register

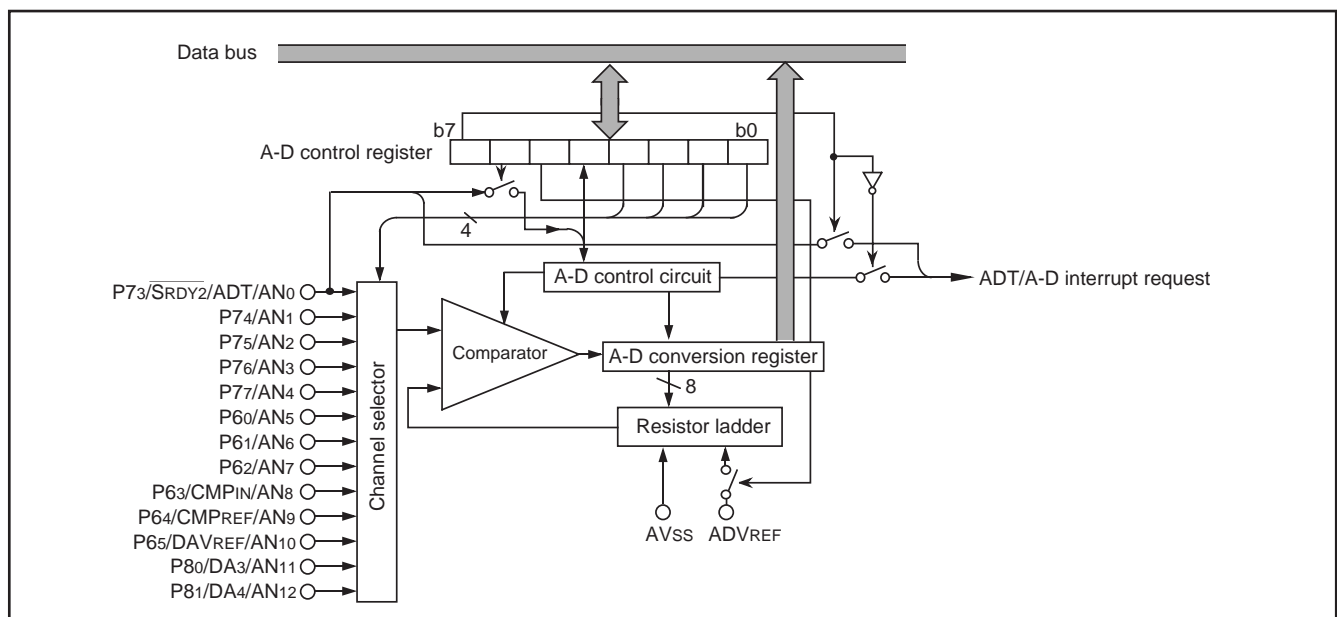


Fig. 45. Block diagram of A-D converter

# HARDWARE

## FUNCTIONAL DESCRIPTION

### D-A Converter

The 3807 group has an on-chip D-A converter with 8-bit resolution and 4 channels (DA<sub>i</sub> (i=1—4)). The D-A converter is performed by setting the value in the D-A conversion register. The result of D-A converter is output from DA<sub>i</sub> pin by setting the DA<sub>i</sub> output enable bits to "1." When using the D-A converter, the corresponding port direction register bit (P65/DAVREF/AN10, P56/DA1, P57/DA2, P80/DA3/AN11, P81/DA4/AN12) should be set to "0" (input status).

The output analog voltage V is determined by the value n (base 10) in the D-A conversion register as follows:

$$V = DAV_{REF} \times n / 256 \quad (n=0 \text{ to } 255)$$

Where DAV<sub>REF</sub> is the reference voltage.

At reset, the D-A conversion registers are cleared to "0016", the DA<sub>i</sub> output enable bits are cleared to "0", and DA<sub>i</sub> pin is set to input (high impedance). The DA output is not buffered, so connect an external buffer when driving a low-impedance load.

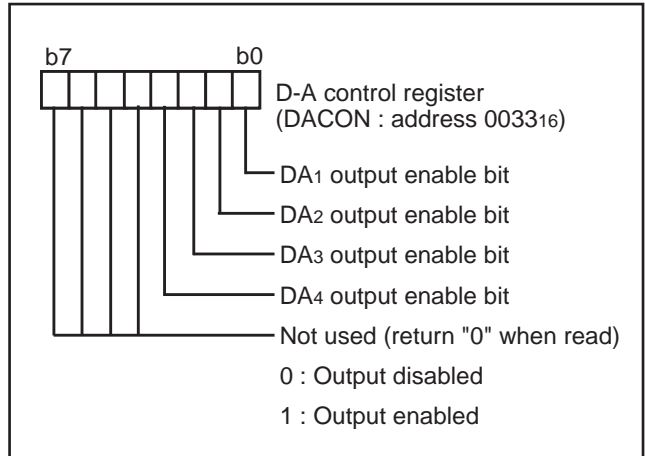


Fig. 46. Structure of D-A control register

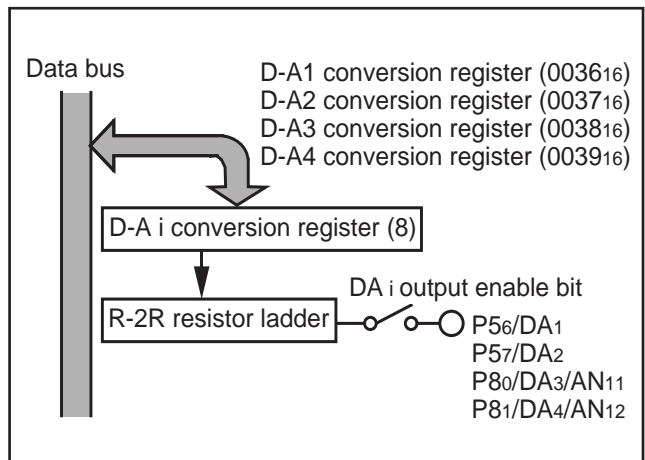


Fig. 47. Block diagram of D-A converter

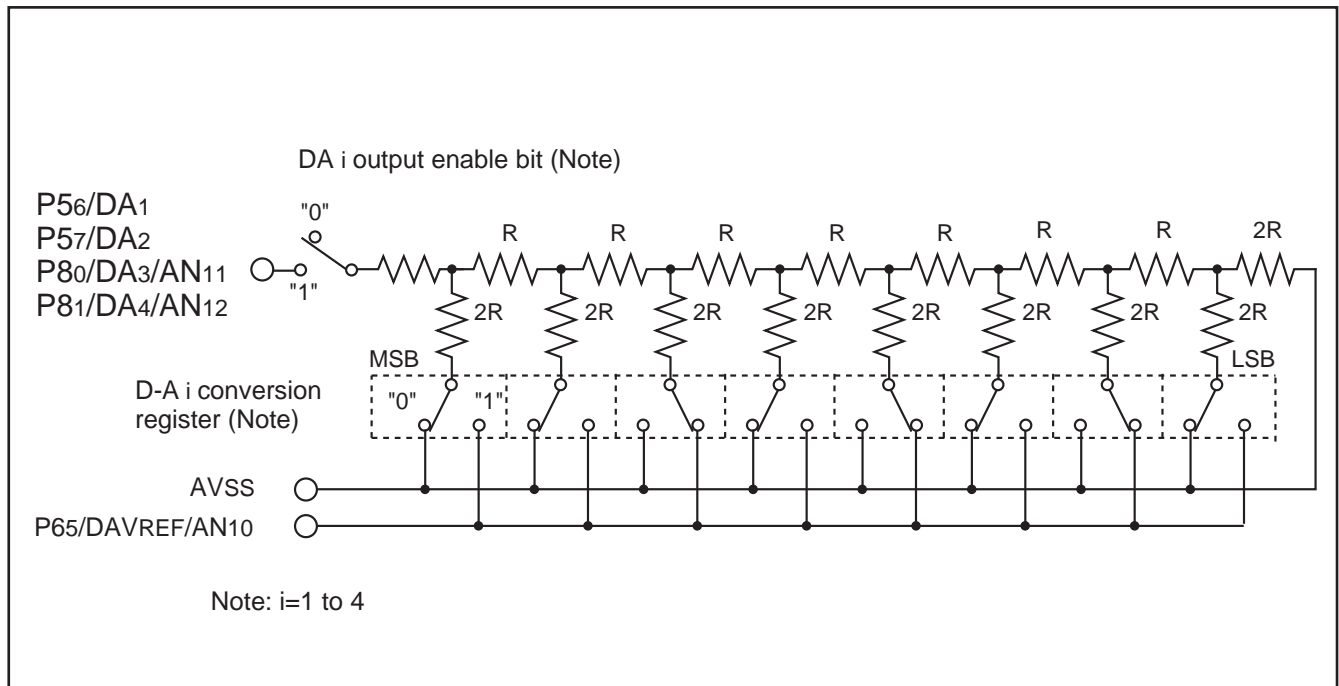


Fig. 48. Equivalent connection circuit of D-A converter

### Analog Comparator

An analog comparator circuit which is independent of peripheral circuits in the microcomputer is incorporated (**Note**).

An analog comparator outputs the result of comparison with an input voltage of CMPREF pin which is specified as a reference voltage and an input voltage of CMPIN pin to CMPOUT pin. The result is "1" when the input voltage to port CMPIN is higher than the voltage applied to port CMPREF and "0" when the voltage is lower.

Because the analog comparator consists of an analog MOS circuit, set the input voltage to the CMPIN pin and the CMPREF pin within the following range :

$V_{ss} + 1.2 \text{ V}$  to  $\text{CMPV}_{CC} - 0.5 \text{ V}$

#### ■Note

The analog comparator circuit is separated from the MCU internal peripheral circuit in the microcomputer. Accordingly, even if the microcomputer runs away, the analog comparator is still in operation. For this reason, the analog comparator can be used for safety circuit design.

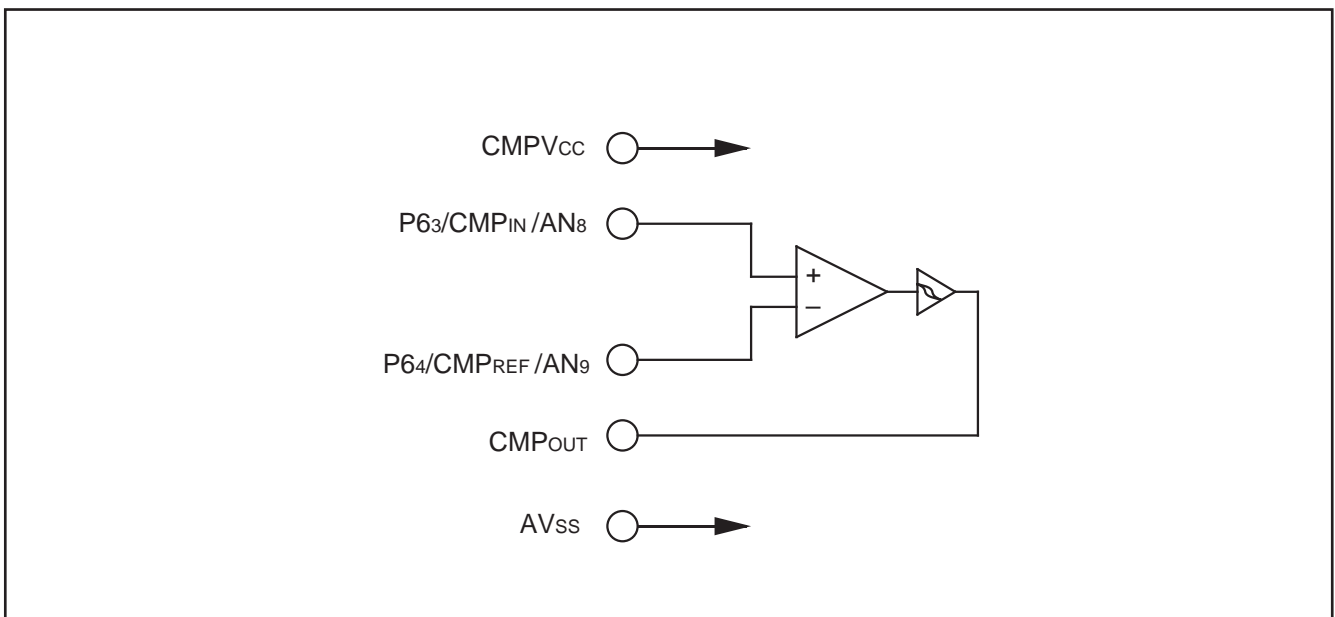


Fig. 49. Block diagram of Analog comparator

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Watchdog Timer

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

#### ●Standard operation of watchdog timer

When any data is not written into the watchdog timer control register (address 0017<sub>16</sub>) after resetting, the watchdog timer is in the stop state. The watchdog timer starts to count down by writing an optional value into the watchdog timer control register (address 0017<sub>16</sub>) and an internal resetting takes place at an underflow of the watchdog timer H.

Accordingly, programming is usually performed so that writing to the watchdog timer control register (address 0017<sub>16</sub>) may be started before an underflow. When the watchdog timer control register (address 0017<sub>16</sub>) is read, the values of the 6 high-order bits of the watchdog timer H, STP instruction disable bit, and watchdog timer H count source selection bit are read.

#### (1) Initial value of watchdog timer

At reset or writing to the watchdog timer control register (address 0017<sub>16</sub>), each watchdog timer H and L is set to "FF<sub>16</sub>."

#### (2) Watchdog timer H count source selection bit operation

Bit 7 of the watchdog timer control register (address 0017<sub>16</sub>) permits selecting a watchdog timer H count source. When this bit is set to "0", the count source becomes the underflow signal of watchdog timer L. The detection time is set then to  $f(X_{IN})=131.072$  ms at 8 MHz frequency and  $f(X_{CIN})=32.768$  s at 32 kHz frequency.

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

#### (3) Operation of STP instruction disable bit

Bit 6 of the watchdog timer control register (address 0017<sub>16</sub>) permits disabling the STP instruction when the watchdog timer is in operation.

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

When this bit is "1", the STP instruction is disabled.

Once the STP instruction is executed, an internal resetting takes place. When this bit is set to "1", it cannot be rewritten to "0" by program. This bit is cleared to "0" after resetting.

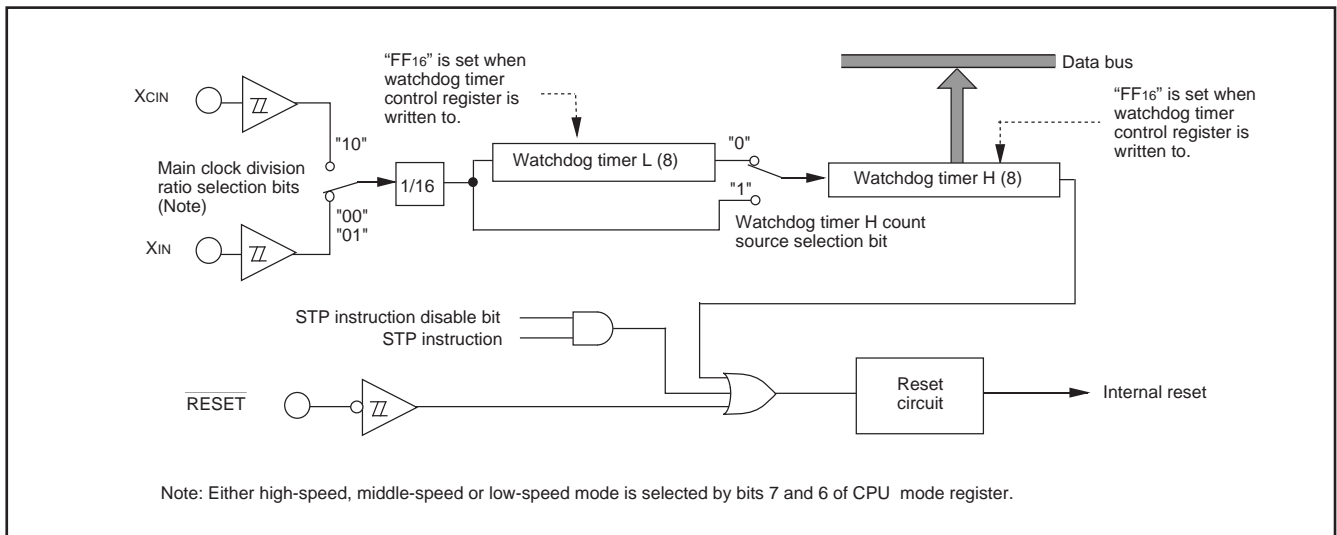


Fig. 50. Block diagram of Watchdog timer

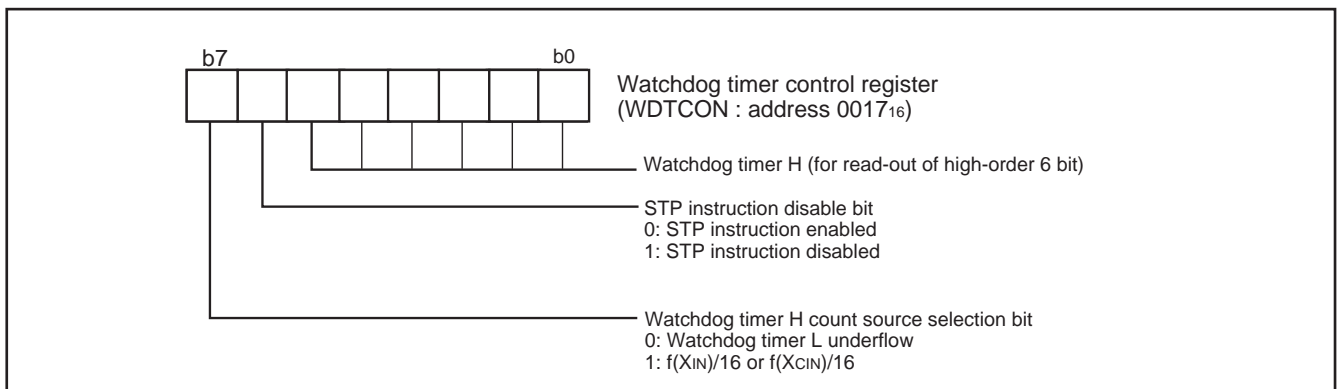


Fig. 51. Structure of Watchdog timer control register

### Clock output function

The internal clock  $\phi$  can be output from I/O port P34. Control of I/O ports and clock output function can be performed by port P2P3 control register (address 001516).

#### (1) I/O ports or clock output function selection

The P34 clock output control bit (b0) of port P2P3 control register selects the I/O port or clock output function. When clock output function is selected, the clock is output regardless of the port P34 direction register settings.

Directly after bit 0 is written to, the port or clock output is switched synchronous to a falling edge of clock frequency selected by the output clock frequency selection bit. When memory expansion mode or microprocessor mode is selected in CPU mode register (b1, b0), clock output is selected on regardless of P34 clock output control bit settings or port P34 direction register settings.

#### (2) Selection of output clock frequency

The output clock frequency selection bits (b3, b2, b1) of port P2P3 control register select the output clock frequency.

The output waveform when  $f(X_{IN})$  or  $f(X_{CIN})$  is selected, depends on  $X_{IN}$  or  $X_{CIN}$  input waveform however; all other output waveform settings have a duty cycle of 50%.

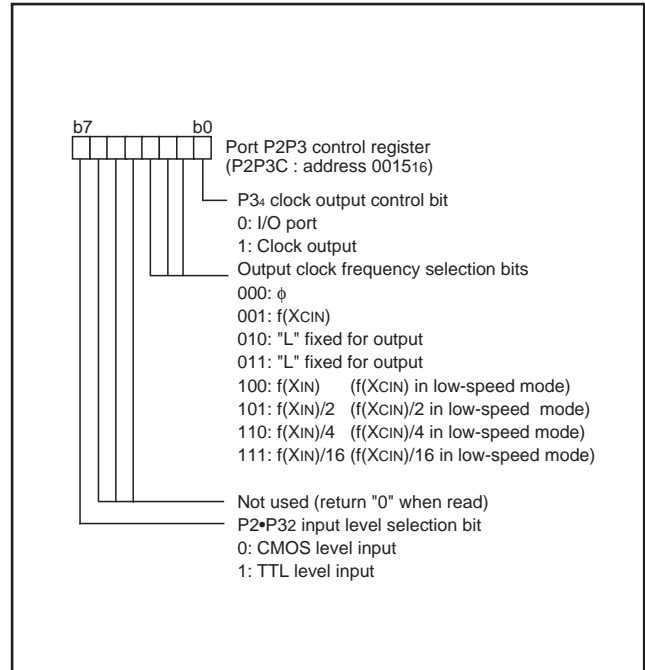


Fig. 52. Structure of Port P2P3 control register

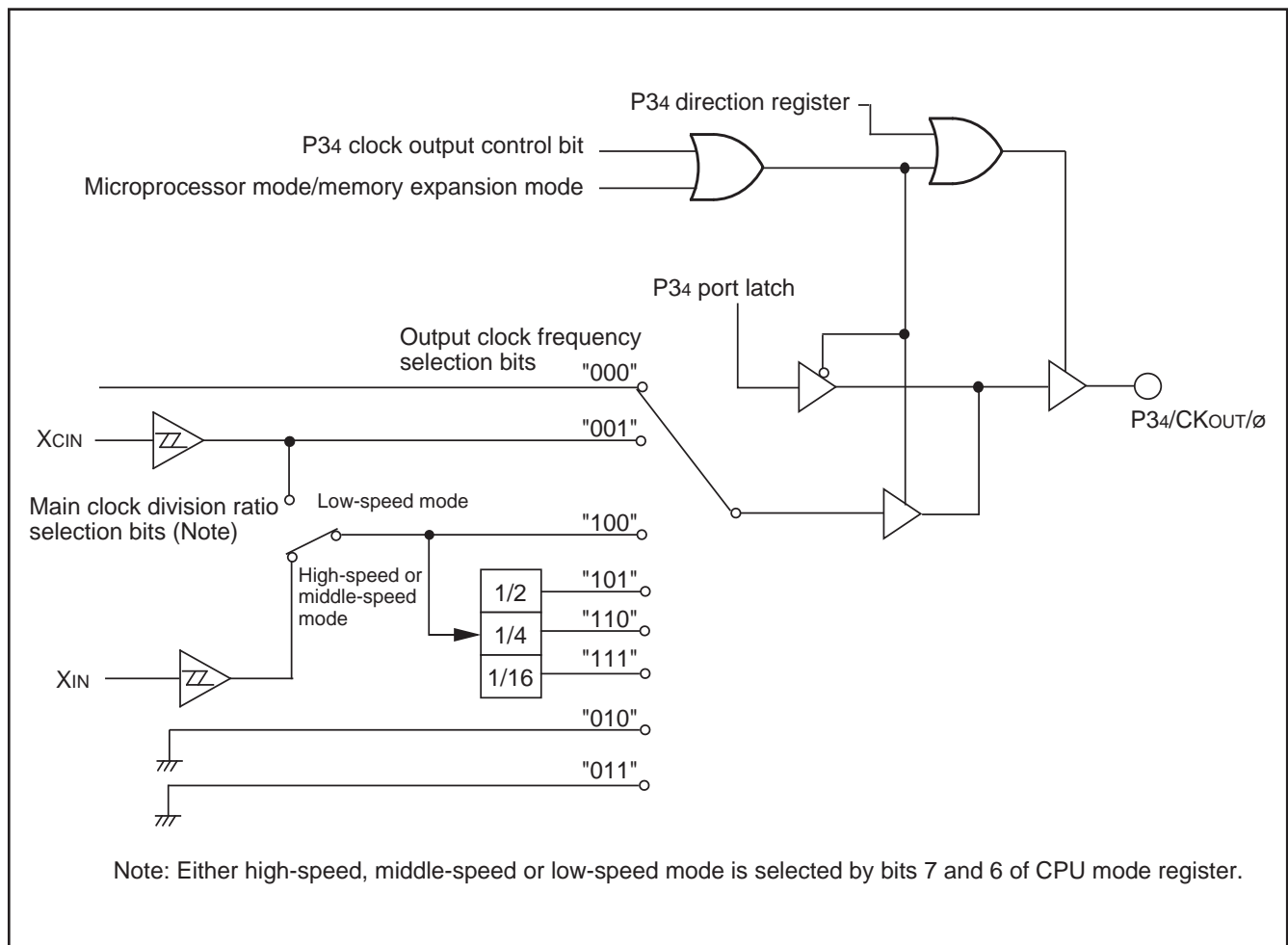


Fig. 53. Block diagram of Clock output function

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Reset Circuit

To reset the microcomputer,  $\overline{\text{RESET}}$  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 2.7 V and 5.5 V, and the oscillation should be stable), reset is released. After the reset is completed, the program starts from the address FFFD<sub>16</sub> (high-order byte) and address FFFC<sub>16</sub> (low-order byte). Make sure that the reset input voltage is less than 0.54 V for  $V_{\text{CC}}$  of 2.7 V.

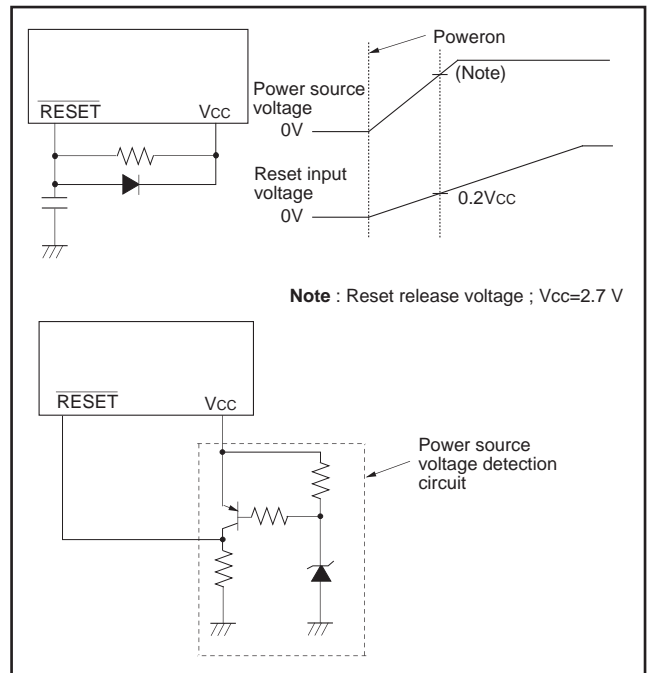


Fig. 54. Reset circuit example

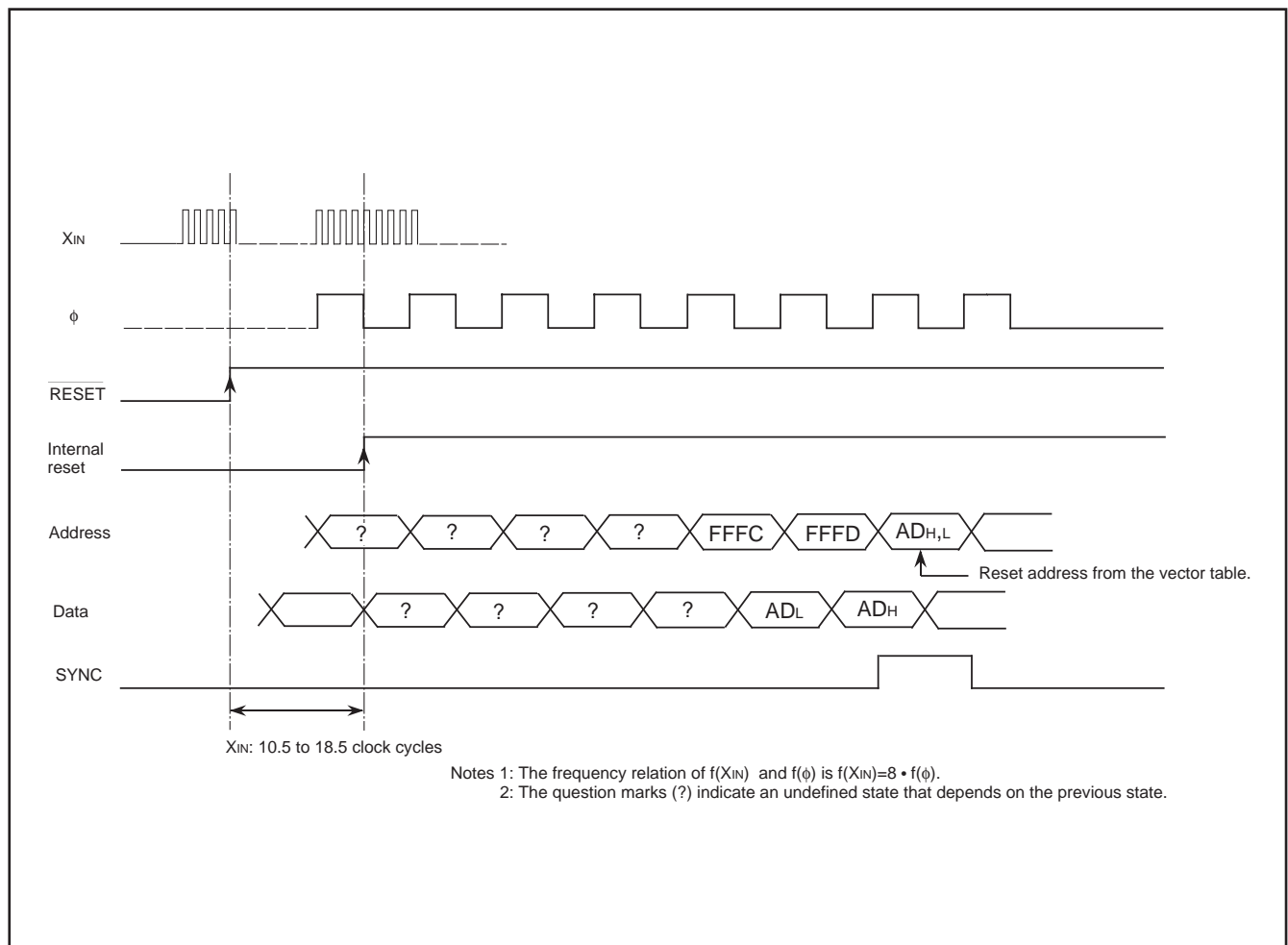


Fig. 55. Reset sequence

	Address	Register contents		Address	Register contents
(1) Port P0	0000 <sub>16</sub>	00 <sub>16</sub>	(34) Timer 3	0026 <sub>16</sub>	FF <sub>16</sub>
(2) Port P0 direction register	0001 <sub>16</sub>	00 <sub>16</sub>	(35) Timer X mode register	0027 <sub>16</sub>	00 <sub>16</sub>
(3) Port P1	0002 <sub>16</sub>	00 <sub>16</sub>	(36) Timer Y mode register	0028 <sub>16</sub>	00 <sub>16</sub>
(4) Port P1 direction register	0003 <sub>16</sub>	00 <sub>16</sub>	(37) Timer 123 mode register	0029 <sub>16</sub>	00 <sub>16</sub>
(5) Port P2	0004 <sub>16</sub>	00 <sub>16</sub>	(38) Real time port register 0—7	002A <sub>16</sub>	00 <sub>16</sub>
(6) Port P2 direction register	0005 <sub>16</sub>	00 <sub>16</sub>	(39) Real time port control register 0	002B <sub>16</sub>	1 0 0 1 0 0 0 0
(7) Port P3	0006 <sub>16</sub>	00 <sub>16</sub>	(40) Real time port control register 1	002C <sub>16</sub>	1 - - - 0 0 0 0
(8) Port P3 direction register	0007 <sub>16</sub>	00 <sub>16</sub>		R/W pointer	- 1 1 1 - - - -
(9) Port P4	0008 <sub>16</sub>	00 <sub>16</sub>		Output pointer	- 1 1 1 - - - -
(10) Port P4 direction register	0009 <sub>16</sub>	00 <sub>16</sub>	(41) Real time port control register 2	002D <sub>16</sub>	1 - - - 0 0 0 0
(11) Port P5	000A <sub>16</sub>	00 <sub>16</sub>		R/W pointer	- 1 1 1 - - - -
(12) Port P5 direction register	000B <sub>16</sub>	00 <sub>16</sub>		Output pointer	- 1 1 1 - - - -
(13) Port P6	000C <sub>16</sub>	00 <sub>16</sub>	(42) Real time port control register 3	002E <sub>16</sub>	00 <sub>16</sub>
(14) Port P6 direction register	000D <sub>16</sub>	00 <sub>16</sub>	(43) Timer A (low-order)	002F <sub>16</sub>	FF <sub>16</sub>
(15) Port P7	000E <sub>16</sub>	00 <sub>16</sub>	(44) Timer A (high-order)	0030 <sub>16</sub>	FF <sub>16</sub>
(16) Port P7 direction register	000F <sub>16</sub>	00 <sub>16</sub>	(45) Timer B (low-order)	0031 <sub>16</sub>	FF <sub>16</sub>
(17) Port P8	0010 <sub>16</sub>	00 <sub>16</sub>	(46) Timer B (high-order)	0032 <sub>16</sub>	FF <sub>16</sub>
(18) Port P8 direction register	0011 <sub>16</sub>	00 <sub>16</sub>	(47) D-A control register	0033 <sub>16</sub>	00 <sub>16</sub>
(19) Timer XY control register	0014 <sub>16</sub>	0 0 0 0 0 0 1 1	(48) A-D control register	0034 <sub>16</sub>	0 0 0 1 0 0 0 0
(20) Port P2P3 control register	0015 <sub>16</sub>	* 0 0 0 0 0 0 0 0	(49) D-A1 conversion register	0036 <sub>16</sub>	00 <sub>16</sub>
(21) Pull-up control register	0016 <sub>16</sub>	00 <sub>16</sub>	(50) D-A2 conversion register	0037 <sub>16</sub>	00 <sub>16</sub>
(22) Watchdog timer control register	0017 <sub>16</sub>	0 0 1 1 1 1 1 1	(51) D-A3 conversion register	0038 <sub>16</sub>	00 <sub>16</sub>
(23) Serial I/O1 status register	0019 <sub>16</sub>	1 0 0 0 0 0 0 0	(52) D-A4 conversion register	0039 <sub>16</sub>	00 <sub>16</sub>
(24) Serial I/O1 control register	001A <sub>16</sub>	00 <sub>16</sub>	(53) Interrupt edge selection register	003A <sub>16</sub>	00 <sub>16</sub>
(25) UART control register	001B <sub>16</sub>	1 1 1 0 0 0 0 0	(54) CPU mode register	003B <sub>16</sub>	0 1 0 0 1 0 * 0
(26) Serial I/O2 control register 1	001D <sub>16</sub>	00 <sub>16</sub>	(55) Interrupt request register 1	003C <sub>16</sub>	00 <sub>16</sub>
(27) Serial I/O2 control register 2	001E <sub>16</sub>	0 0 0 0 0 1 1 1	(56) Interrupt request register 2	003D <sub>16</sub>	00 <sub>16</sub>
(28) Timer X (low-order)	0020 <sub>16</sub>	FF <sub>16</sub>	(57) Interrupt control register 1	003E <sub>16</sub>	00 <sub>16</sub>
(29) Timer X (high-order)	0021 <sub>16</sub>	FF <sub>16</sub>	(58) Interrupt control register 2	003F <sub>16</sub>	00 <sub>16</sub>
(30) Timer Y (low-order)	0022 <sub>16</sub>	FF <sub>16</sub>	(59) Processor status register	(PS)	x x x x 1 x x x
(31) Timer Y (high-order)	0023 <sub>16</sub>	FF <sub>16</sub>	(60) Program counter	(PCH)	FFFD <sub>16</sub> contents
(32) Timer 1	0024 <sub>16</sub>	FF <sub>16</sub>		(PCL)	FFFC <sub>16</sub> contents
(33) Timer 2	0025 <sub>16</sub>	01 <sub>16</sub>			

\* The initial values depend on level of port CNV<sub>ss</sub>.  
X: Not fixed  
Since the initial values for other than above mentioned registers and RAM contents are indefinite at reset, they must be set.

Fig. 56. Internal status at reset

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Clock Generating Circuit

The 3807 group has two built-in oscillation circuits. An oscillation circuit can be formed by connecting a resonator between  $X_{IN}$  and  $X_{OUT}$  ( $X_{CIN}$  and  $X_{COUT}$ ). Use the circuit constants in accordance with the resonator manufacturer's recommended values. No external resistor is needed between  $X_{IN}$  and  $X_{OUT}$  since a feed-back resistor exists on-chip. However, an external feed-back resistor is needed between  $X_{CIN}$  and  $X_{COUT}$ .

Immediately after poweron, only the  $X_{IN}$  oscillation circuit starts oscillating, and  $X_{CIN}$  and  $X_{COUT}$  pins function as I/O ports.

#### ●Frequency control

##### (1) Middle-speed mode

The internal clock  $\phi$  is the frequency of  $X_{IN}$  divided by 8. After reset, this mode is selected.

##### (2) High-speed mode

The internal clock  $\phi$  is half the frequency of  $X_{IN}$ .

##### (3) Low-speed mode

The internal clock  $\phi$  is half the frequency of  $X_{CIN}$ .

#### ■Note

If you switch the mode between middle/high-speed and low-speed, stabilize both  $X_{IN}$  and  $X_{CIN}$  oscillations. The sufficient time is required for the sub clock to stabilize, especially immediately after poweron and at returning from stop mode. When switching the mode between middle/high-speed and low-speed, set the frequency on condition that  $f(X_{IN}) > 3f(X_{CIN})$ .

##### (4) Low power consumption mode

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

By clearing furthermore the  $X_{COUT}$  drivability selection bit (b3) of CPU mode register to "0", low power consumption operation of less than  $55 \mu A$  ( $V_{CC}=3 V$ ,  $X_{CIN}=32 kHz$ ) can be realized by reducing the drivability between  $X_{CIN}$  and  $X_{COUT}$ . At reset or during STP instruction execution this bit is set to "1" and a reduced drivability that has an easy oscillation start is set. The sub-clock  $X_{CIN}$ - $X_{COUT}$  oscillating circuit can not directly input clocks that are generated externally. Accordingly, make sure to cause an external resonator to oscillate.

#### ●Oscillation control

##### (1) Stop mode

If the STP instruction is executed, the internal clock  $\phi$  stops at an "H" level, and  $X_{IN}$  and  $X_{CIN}$  oscillators stop. Timer 1 is set to "FF<sub>16</sub>" and timer 2 is set to "01<sub>16</sub>."

Either  $X_{IN}$  or  $X_{CIN}$  divided by 16 is input to timer 1 as count source, and the output of timer 1 is connected to timer 2. The bits of the timer 123 mode register except timer 3 count source selection bit (b4) are cleared to "0". Set the timer 2/INT<sub>3</sub> interrupt source bit to "1" and timer 1/INT<sub>2</sub> as well as timer 2/INT<sub>3</sub> interrupt enable bit to disabled ("0") before executing the STP instruction. Oscillator restarts when an external interrupt is received, but the internal clock  $\phi$  is not sup-

plied to the CPU (remains at "H") until timer 2 underflows. This allows time for the clock circuit oscillation to stabilize. The internal clock  $\phi$  is supplied for the first time, when timer 2 underflows. Therefore make sure not to set the timer 2/INT<sub>3</sub> interrupt request bit to "1" before the STP instruction stops the oscillator. When the oscillator is restarted by reset apply "L" level to port  $\overline{RESET}$  until the oscillation is stable since a wait time will not be generated.

##### (2) Wait mode

If the WIT instruction is executed, the internal clock  $\phi$  stops at an "H" level. The states of  $X_{IN}$  and  $X_{CIN}$  are the same as the state before executing the WIT instruction. The internal clock restarts at reset or when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

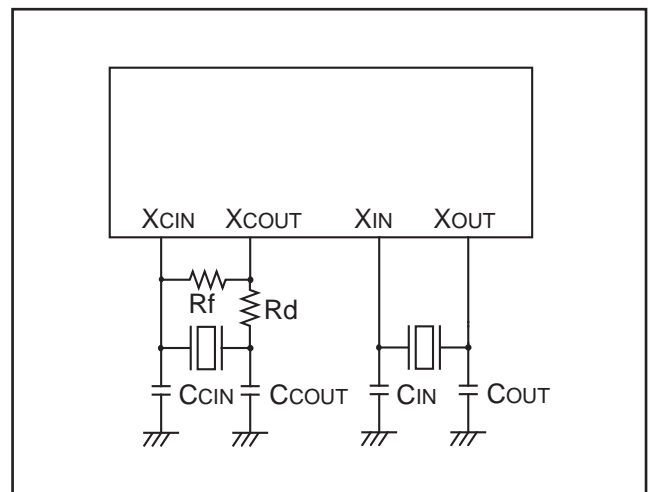


Fig. 57. Ceramic resonator circuit

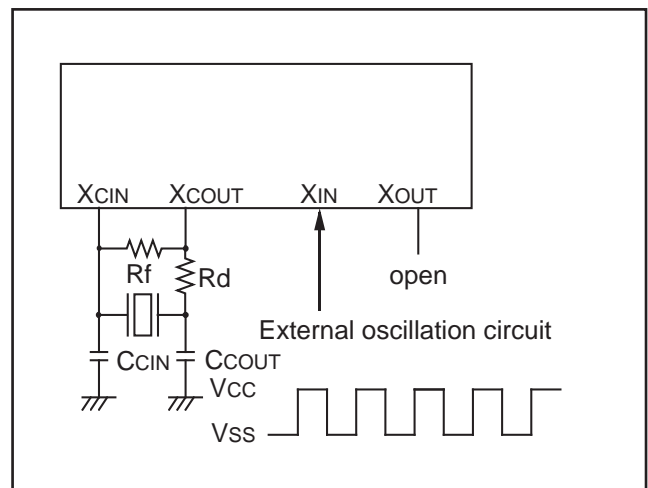


Fig. 58. External clock input circuit



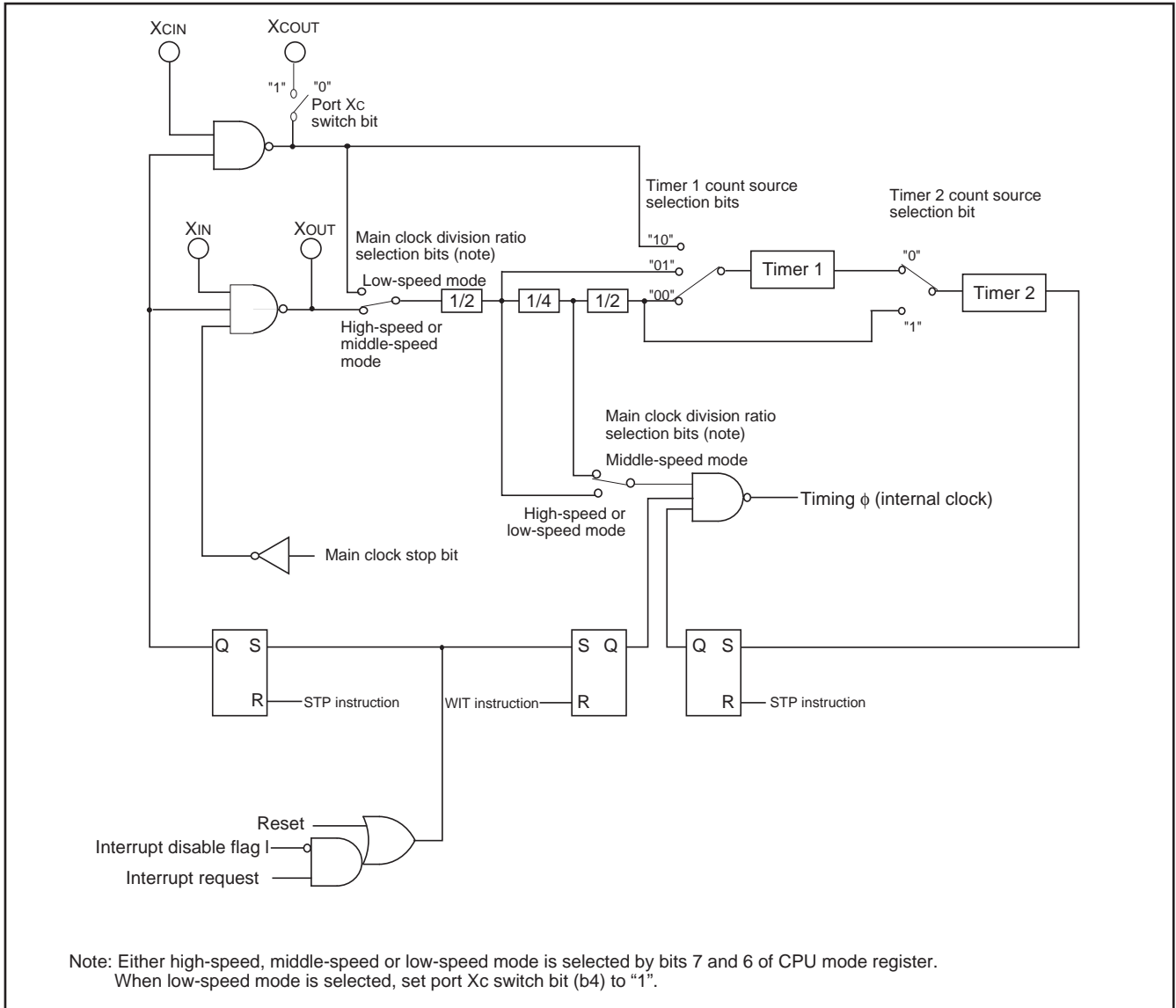
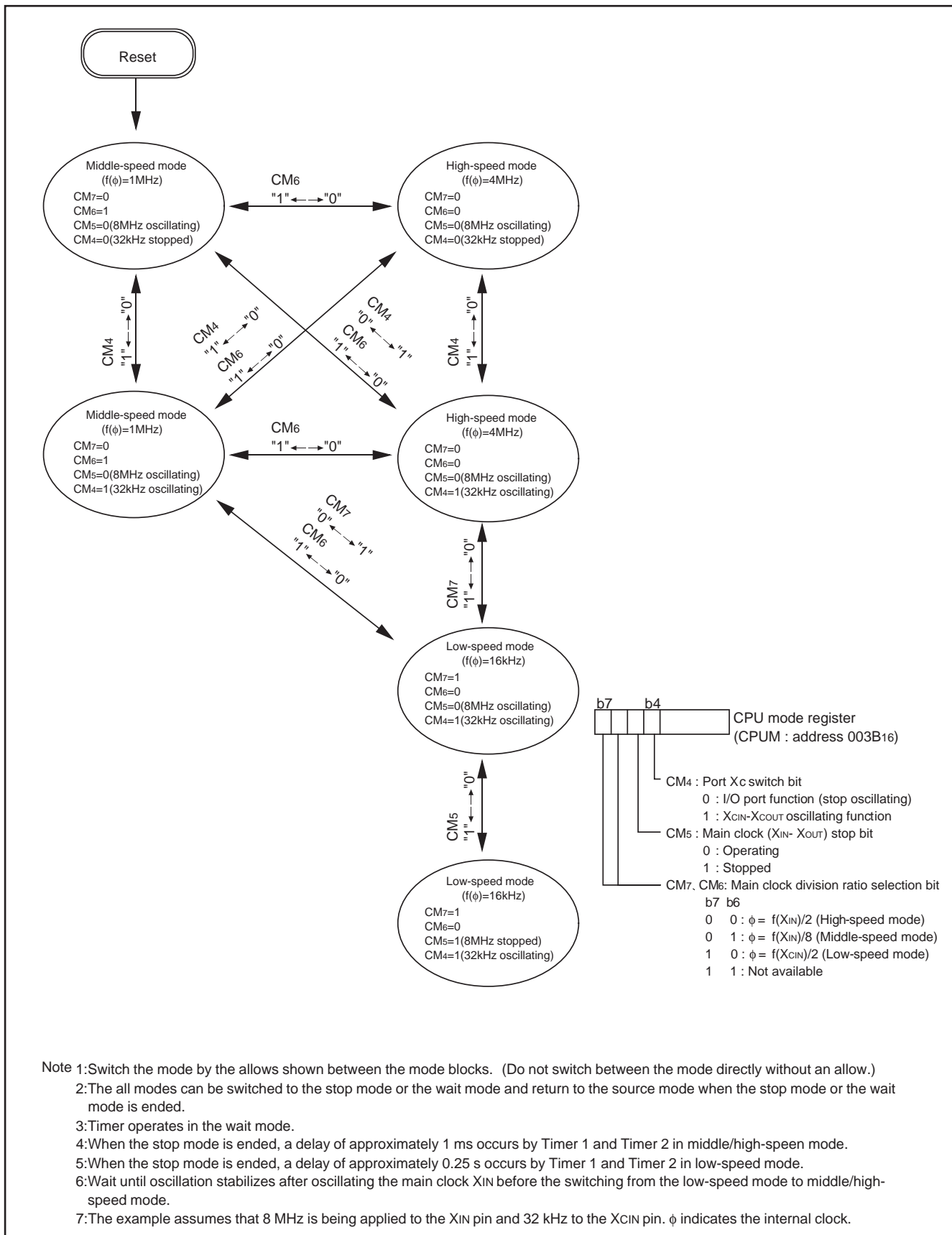


Fig. 59. System clock generating circuit block diagram (Single-chip mode)

# HARDWARE

## FUNCTIONAL DESCRIPTION



**Fig. 60. State transitions of system clock**

### Processor Mode

Single-chip mode, memory expansion mode, and microprocessor mode can be selected by changing the contents of the processor mode bits ( $CM_0$  and  $CM_1$  : b1 and b0 of address  $003B_{16}$ ). 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 9. Port functions in memory expansion mode and microprocessor mode**

Port Name	Function
Port P0	Outputs 8-bits low-order byte of address.
Port P1	Outputs 8-bits high-order byte of address.
Port P2	Operates as I/O pins for data D7 to D0 (including instruction code)
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 $\overline{REST}_{out}$ output pin. ( <b>Note</b> ) P34 is the $\phi$ output pin. P35 is the $\overline{SYNC}$ output pin. P36 is the $\overline{WR}$ output pin, and P37 is the $\overline{RD}$ output pin.

**Note :** If  $CNV_{ss}$  is connected to  $V_{ss}$ , the microcomputer goes to single-chip mode after a reset, so this pin cannot be used as the  $\overline{RESET}_{out}$  output pin.

#### (1) Single-chip mode

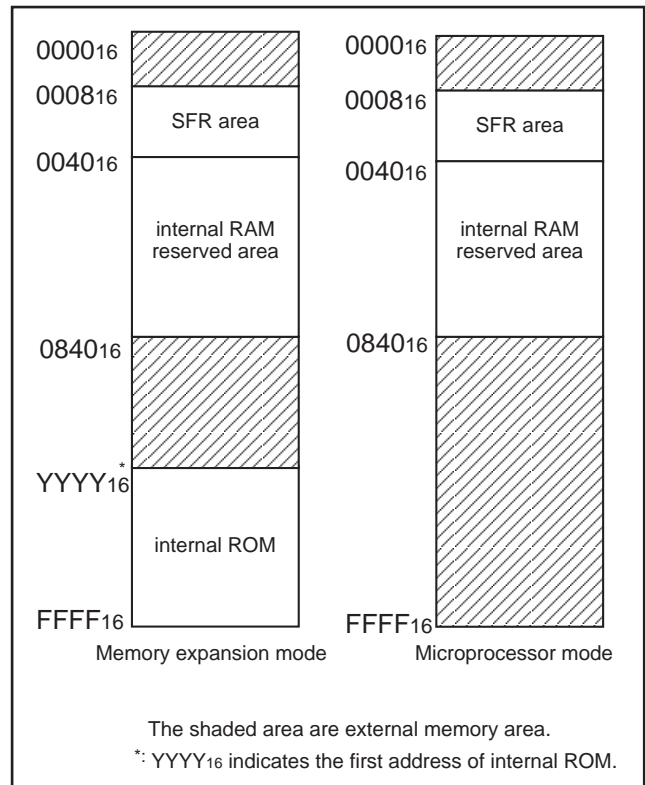
Select this mode by resetting the microcomputer with  $CNV_{ss}$  connected to  $V_{ss}$ .

#### (2) Memory expansion mode

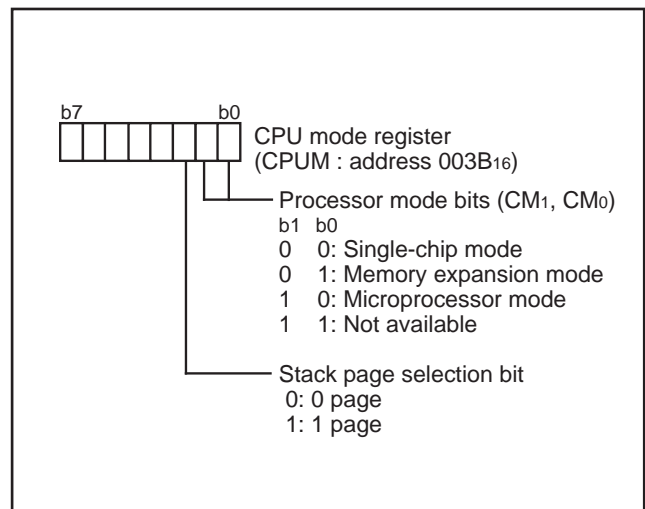
Select this mode by setting the processor mode bits (b1, b0) to "01" in software with  $CNV_{ss}$  connected to  $V_{ss}$ . This mode enables external memory expansion while maintaining the validity of the internal ROM. However, some I/O devices will not support the memory expansion mode. Internal ROM will take precedence over external memory if addresses conflict.

#### (3) Microprocessor mode

Select this mode by resetting the microcomputer with  $CNV_{ss}$  connected to  $V_{cc}$ , or by setting the processor mode bits to "10" in software with  $CNV_{ss}$  connected to  $V_{ss}$ . In microprocessor mode, the internal ROM is no longer valid and external memory must be used.



**Fig. 61. Memory maps in various processor modes**



**Fig. 62. Structure of CPU mode register**

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Bus control at memory expansion

The 3807 group has a built-in ONW function to facilitate access to external (expanded) memory and I/O devices in memory expansion mode or microprocessor mode.

If an "L" level signal is input to port P32/ $\overline{\text{ONW}}$  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  $\overline{\text{RD}}$  or  $\overline{\text{WR}}$  signal remains at "L". This extension function is valid only for writing to and reading from addresses 0000<sub>16</sub> to 0007<sub>16</sub> and 0840<sub>16</sub> to FFFF<sub>16</sub>, and only read and write cycles are extended.

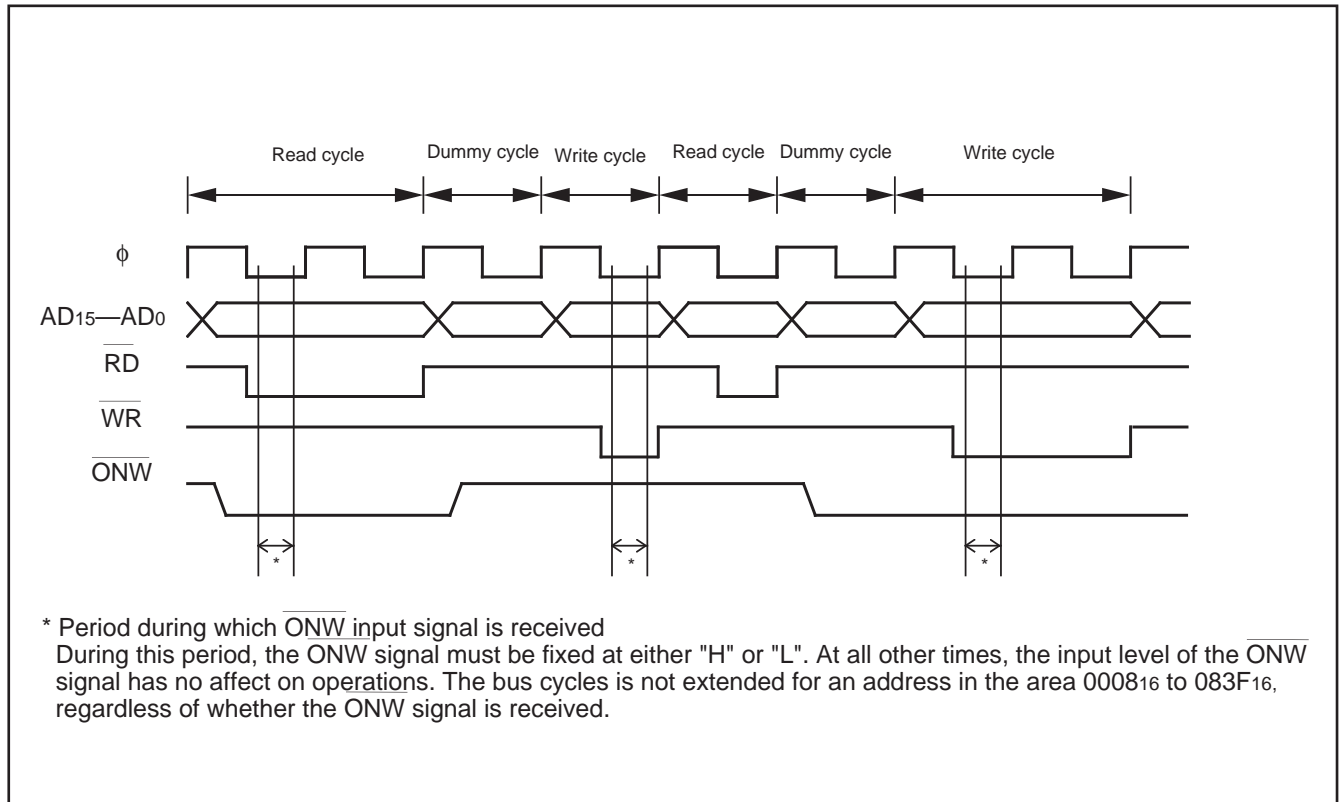


Fig. 63.  $\overline{\text{ONW}}$  function timing

### 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 performing a BBC or BBS instruction.

#### Decimal Calculations

- To calculate in decimal notation, set the decimal mode flag (D) to "1", then execute an ADC or SBC instruction. 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.

#### Timers

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

#### Multiplication and Division Instructions

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

#### Ports

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

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

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

#### Serial I/O

In clock synchronous serial I/O, if the receive side is using an external clock and it is to output the SRDY1 signal, set the transmit enable bit, the receive enable bit, and the SRDY1 output enable bit to "1."

Serial I/O1 continues to output the final bit from the TxD pin after transmission is completed. Sout2 pin for serial I/O2 goes to high impedance after transfer is completed.

When in serial I/O1 (clock-synchronous mode) or in serial I/O2 an external clock is used as synchronous clock, write transmission data to both the transmit buffer register and serial I/O2 register, during transfer clock is "H."

#### A-D Converter

The comparator uses internal capacitors whose charge will be lost if the clock frequency is too low.

Therefore, make sure that  $f(X_{IN})$  is at least on 500 kHz during an A-D conversion. (When the ONW pin has been set to "L", the A-D conversion will take twice as long to match the longer bus cycle, and so  $f(X_{IN})$  must be at least 1 MHz.)

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

#### D-A Converter

The accuracy of the D-A converter becomes rapidly poor under the  $V_{CC} = 4.0$  V or less condition; a supply voltage of  $V_{CC} \geq 4.0$  V is recommended. When a D-A converter is not used, set all values of D-Ai conversion registers (i=1 to 4) to "00<sub>16</sub>."

#### 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  $X_{IN}$  frequency in high-speed mode.

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

# HARDWARE

## NOTES ON USAGE

---

### NOTES ON USAGE

#### Handling of Source Pins

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

#### P34 clock output function

In the case of using an I/O port P34 as a clock output function, note the following : when an output clock frequency is changed during outputting a clock, the port may feed a noise having a shorter pulse width than the standard at the switch timing. Besides, it also may happen at the timing for switching the low-speed mode to the middle/high-speed mode.

#### Timer X and timer Y

In the pulse period measurement mode or the pulse width measurement mode for timers X and Y, set the "L" or "H" pulse width of input signal from CNTR0/CNTR1 pin to 2 cycles or more of a timer count source.

#### EPROM version/One Time PROM version

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

To improve the noise reduction, connect a track between CNVss pin and Vss pin or Vcc pin with 1 to 10 k $\Omega$  resistance.

The mask ROM version track of port CNVss has no operational interference even if it is connected via a resistor.

## 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)

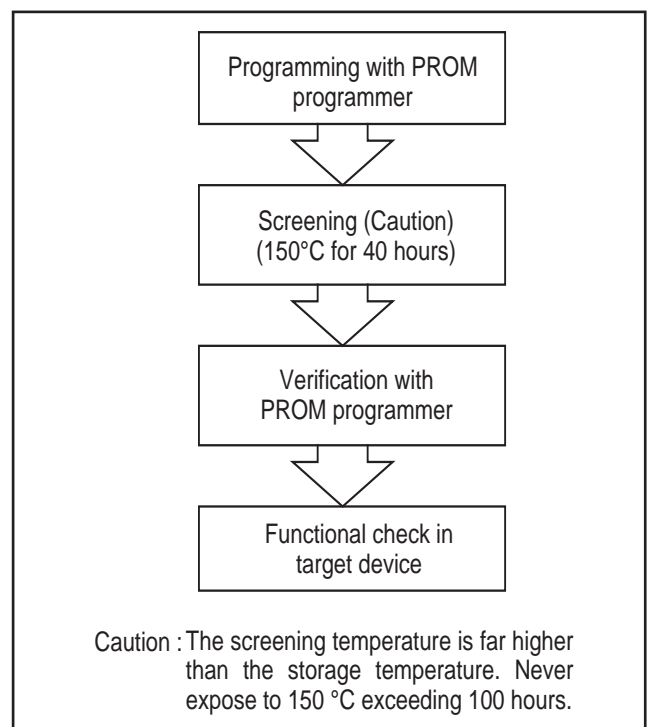
### ROM PROGRAMMING METHOD

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. 10. Special programming adapter**

Package	Name of Programming Adapter
80P6N-A	PCA4738F-80A
80D0	PCA4738L-80A

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 64 is recommended to verify programming.



**Fig. 64. Programming and testing of One Time PROM version**

# HARDWARE

## FUNCTIONAL DESCRIPTION SUPPLEMENT

### FUNCTIONAL DESCRIPTION SUPPLEMENT

#### Interrupt

3807 group permits interrupts on the basis of 16 sources. It is vector interrupts with a fixed priority system. Accordingly, when two or more interrupt

requests occur during the same sampling, the higher-priority interrupt is accepted first. This priority is determined by hardware, but variety of priority processing can be performed by software, using an interrupt enable bit and an interrupt disable flag. For interrupt sources, vector addresses and interrupt priority, refer to “**Table 11.**”

**Table 11. 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 <sub>1</sub> receive interrupt	FFF7 <sub>16</sub>	FFF6 <sub>16</sub>	Valid when serial I/O <sub>1</sub> is selected
5	Serial I/O <sub>1</sub> transmit interrupt	FFF5 <sub>16</sub>	FFF4 <sub>16</sub>	Valid when serial I/O <sub>1</sub> 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	INT <sub>3</sub> interrupt	FFEF <sub>16</sub>	FFEE <sub>16</sub>	External interrupt(active edge selectable)
	Valid when INT <sub>3</sub> interrupt is selected			
9	Timer 2 interrupt	FFED <sub>16</sub>	FFEC <sub>16</sub>	Valid when timer 2 interrupt is selected
	INT <sub>4</sub> interrupt			External interrupt(active edge selectable)
10	Timer 3 interrupt	FFEB <sub>16</sub>	FFEA <sub>16</sub>	Valid when INT <sub>4</sub> interrupt is selected
	CNTR <sub>0</sub> interrupt			External interrupt (active edge selectable)
11	CNTR <sub>1</sub> interrupt	FFE9 <sub>16</sub>	FFE8 <sub>16</sub>	External interrupt (active edge selectable)
	Serial I/O <sub>2</sub> interrupt			Valid when serial I/O <sub>2</sub> is selected
12	INT <sub>2</sub> interrupt	FFE7 <sub>16</sub>	FFE6 <sub>16</sub>	External interrupt(active edge selectable)
	Valid when INT <sub>2</sub> interrupt is selected			
13	Timer 1 interrupt	FFE5 <sub>16</sub>	FFE4 <sub>16</sub>	Valid when timer 1 interrupt is selected
	Timer A interrupt			FFE3 <sub>16</sub>
14	Timer B interrupt	FFE1 <sub>16</sub>	FFE0 <sub>16</sub>	
	A-D conversion interrupt			FFDF <sub>16</sub>
15	ADT interrupt	FFDD <sub>16</sub>	FFDC <sub>16</sub>	
				Valid when ADT interrupt and A-D external trigger valid are selected
16	BRK instruction interrupt			Non-maskable software interrupt

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



### Timing After Interrupt

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

Figure 65 shows a timing chart after an interrupt occurs, and Figure 66 shows the time up to execution of the interrupt processing routine.

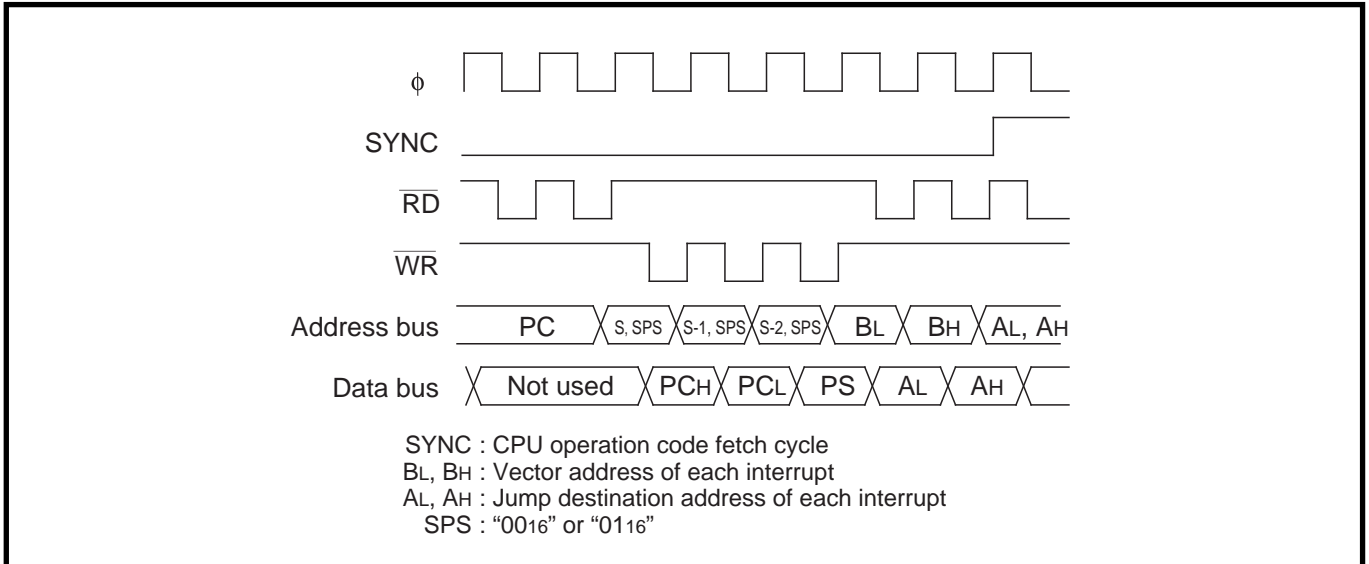


Fig. 65 Timing chart after an interrupt occurs

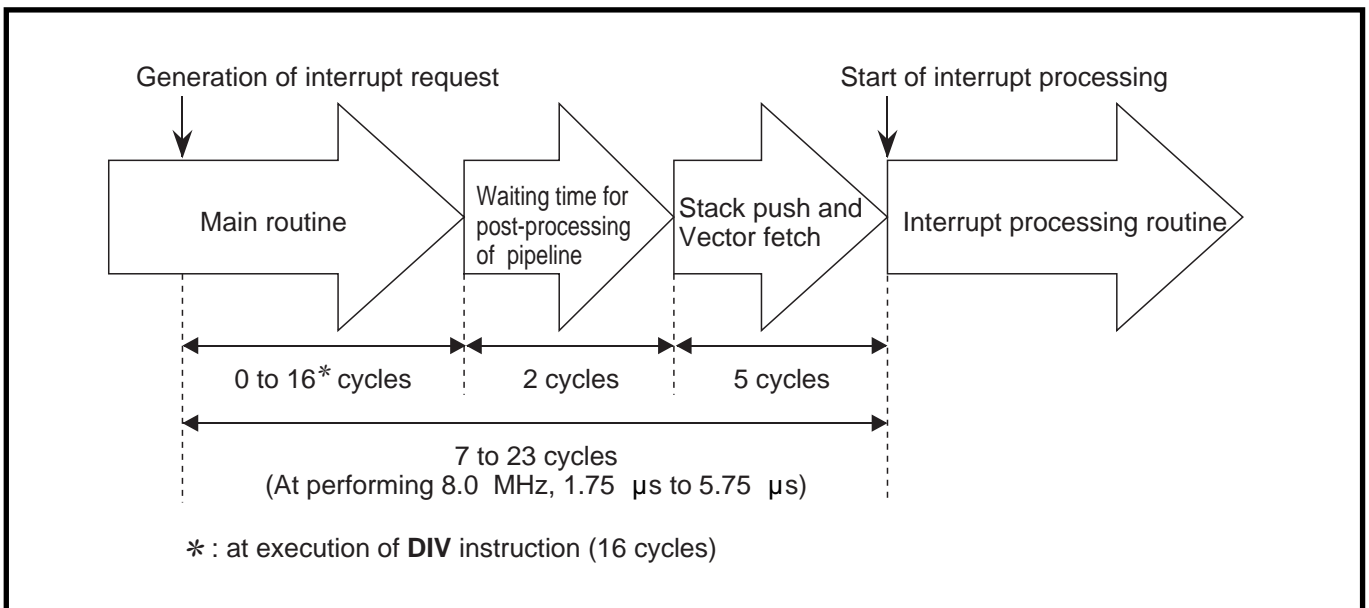


Fig. 66 Time up to execution of the interrupt processing routine

# HARDWARE

## FUNCTIONAL DESCRIPTION SUPPLEMENT

### A-D Converter

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

1. After the start of A-D conversion, A-D conversion register goes to "0016."
2. The highest-order bit of A-D conversion register is set to "1," and the comparison voltage Vref is input to the comparator. Then, Vref is compared with analog input voltage VIN.
3. As a result of comparison, when Vref < VIN, the highest-order bit of A-D conversion register becomes "1." When Vref > VIN, the highest-order bit becomes "0."

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

A-D conversion completes at 50 clock cycles (12.5 μs at f(XIN) = 8.0 MHz) after it is started, and the result of the conversion is stored into the A-D conversion register.

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

Relative formula for a reference voltage VREF of A-D converter and Vref	
When n = 0	Vref = 0
When n = 1 to 255	$V_{ref} = \frac{V_{REF}}{256} \times (n - 0.5)$
n : the value of A-D converter (decimal numeral)	

**Table 12. Change of A-D conversion register during A-D conversion**

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

- \*1: A result of the first comparison
- \*3: A result of the third comparison
- \*5: A result of the fifth comparison
- \*7: A result of the seventh comparison

- \*2: A result of the second comparison
- \*4: A result of the fourth comparison
- \*6: A result of the sixth comparison
- \*8: A result of the eighth comparison

Figure 67 shows A-D conversion equivalent circuit, and Figure 68 shows A-D conversion timing chart.

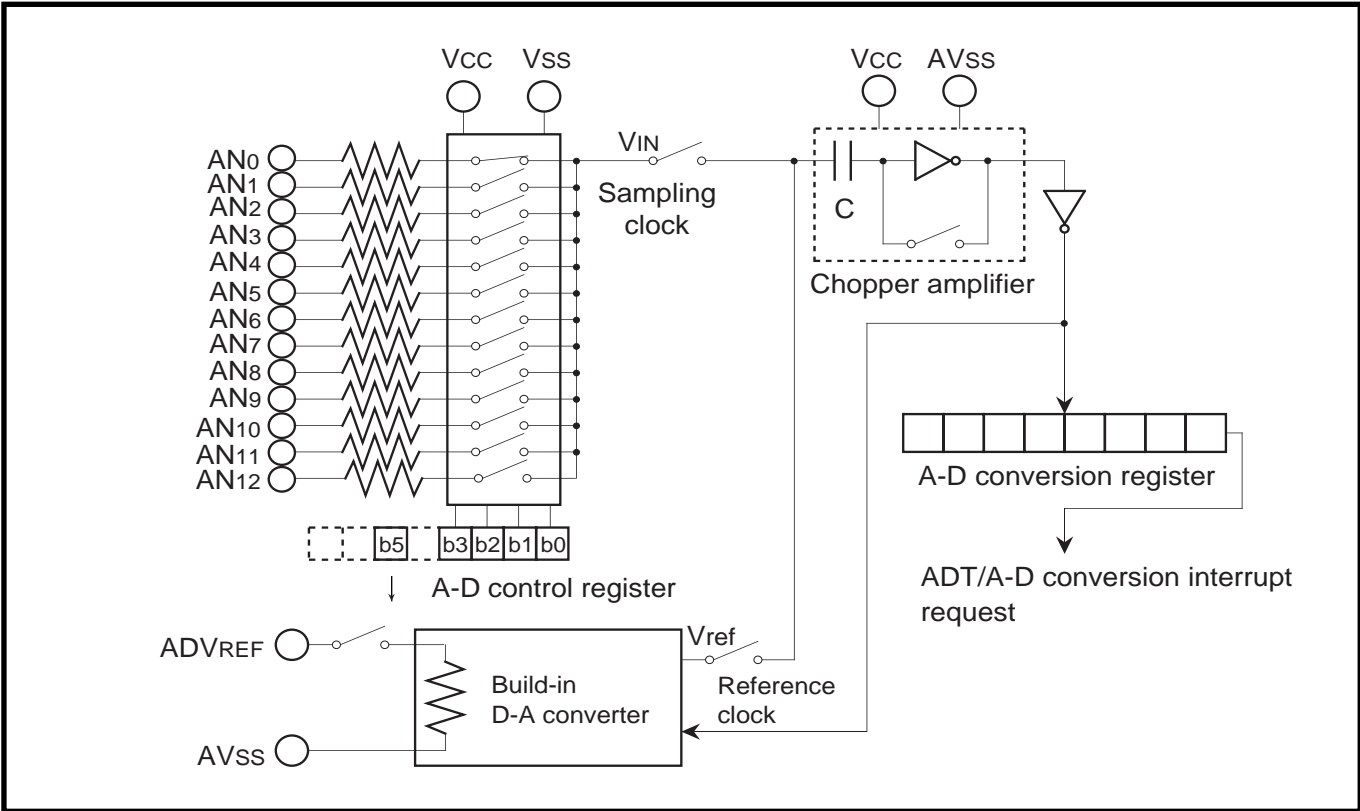


Fig. 67 A-D conversion equivalent circuit

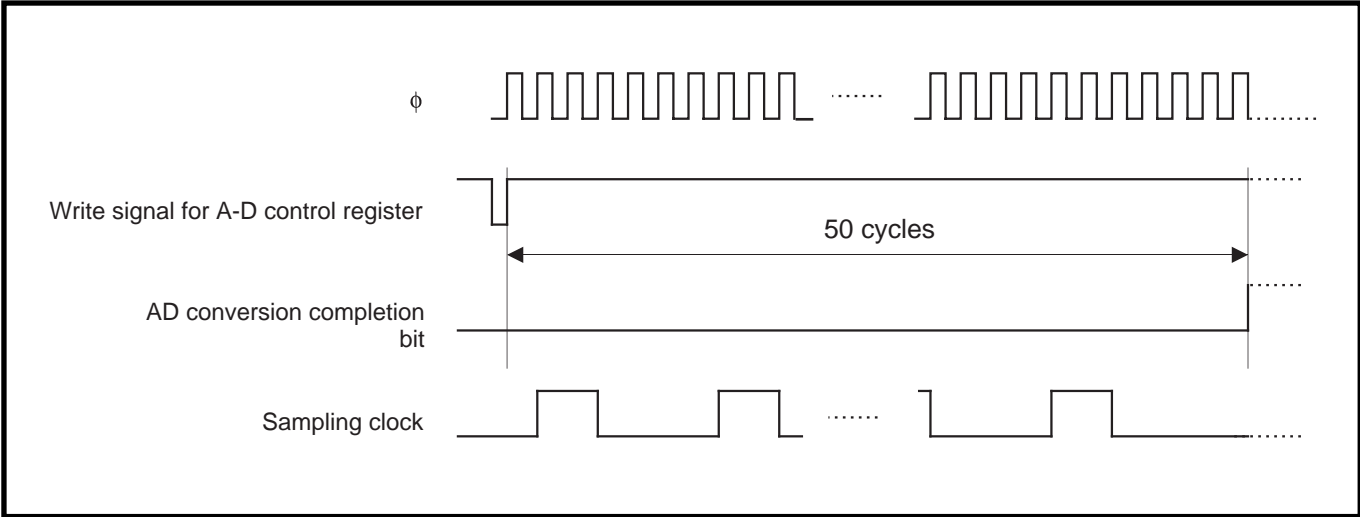


Fig. 68 A-D conversion timing chart



# CHAPTER 2

## **APPLICATION**

- 2.1 I/O port
- 2.2 Timer
- 2.3 Serial I/O
- 2.4 Real time output port
- 2.5 A-D converter
- 2.6 Reset
- 2.7 Application circuit example

# 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)
0010 <sub>16</sub>	Port P8 (P8)
0011 <sub>16</sub>	Port P8 direction register (P8D)

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

### 2.1.2 Related registers

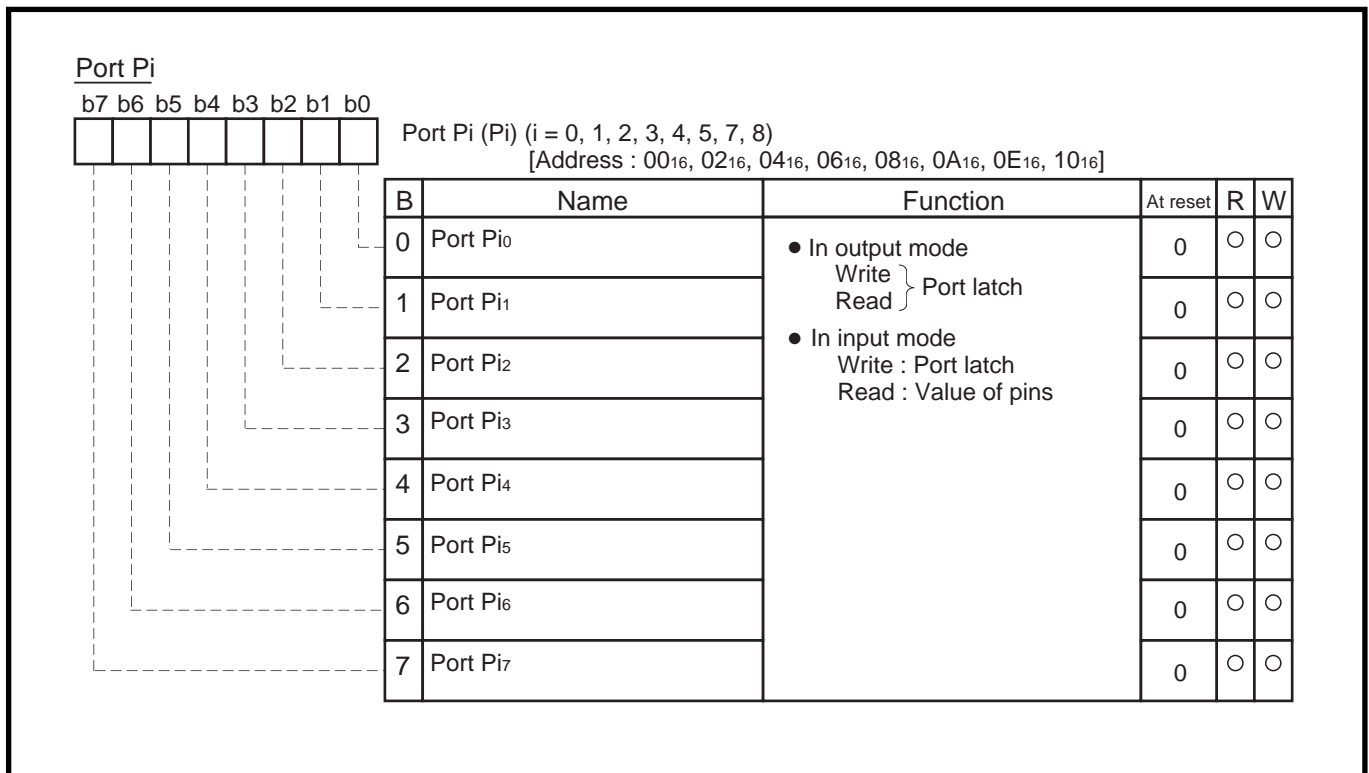


Fig. 2.1.2 Structure of Port Pi (i = 0, 1, 2, 3, 4, 5, 7, 8)

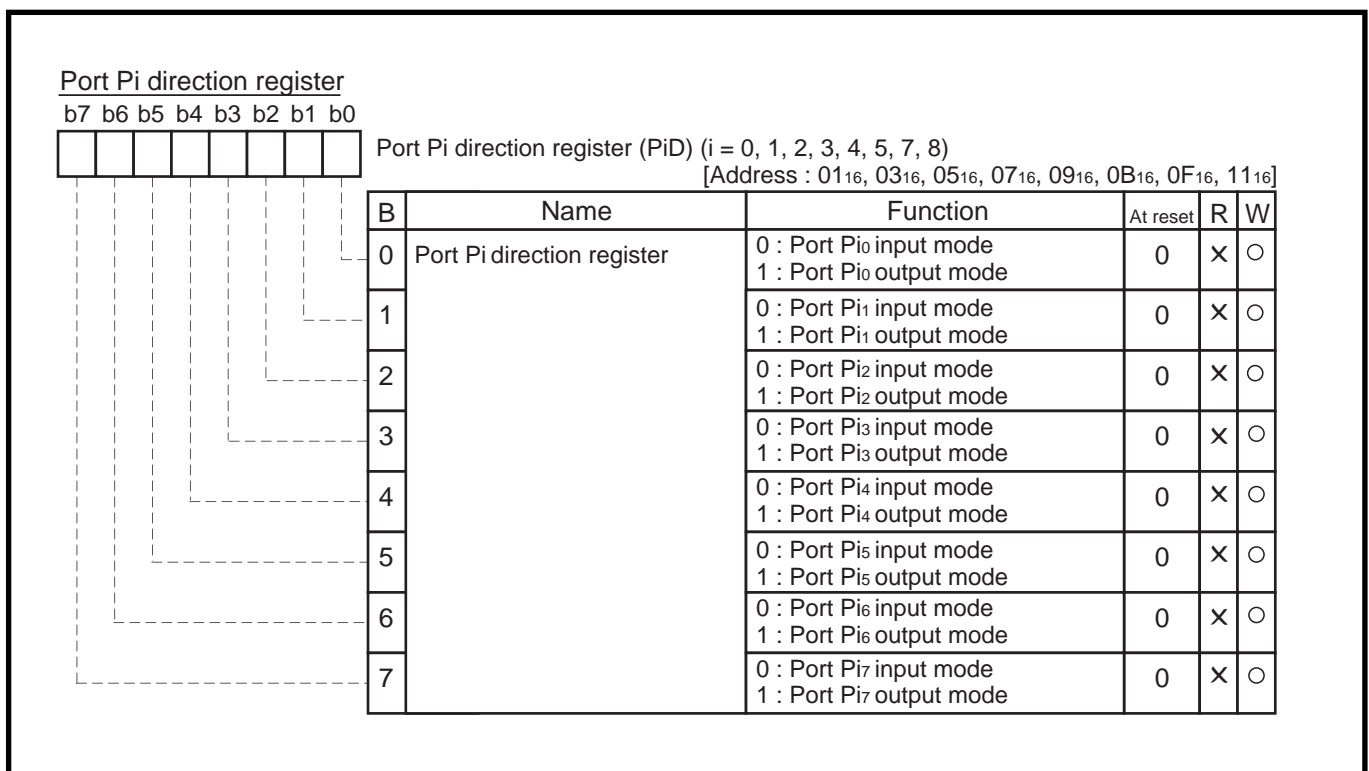


Fig. 2.1.3 Structure of Port Pi direction register (i = 0, 1, 2, 3, 4, 5, 7, 8)

# APPLICATION

## 2.1 I/O port

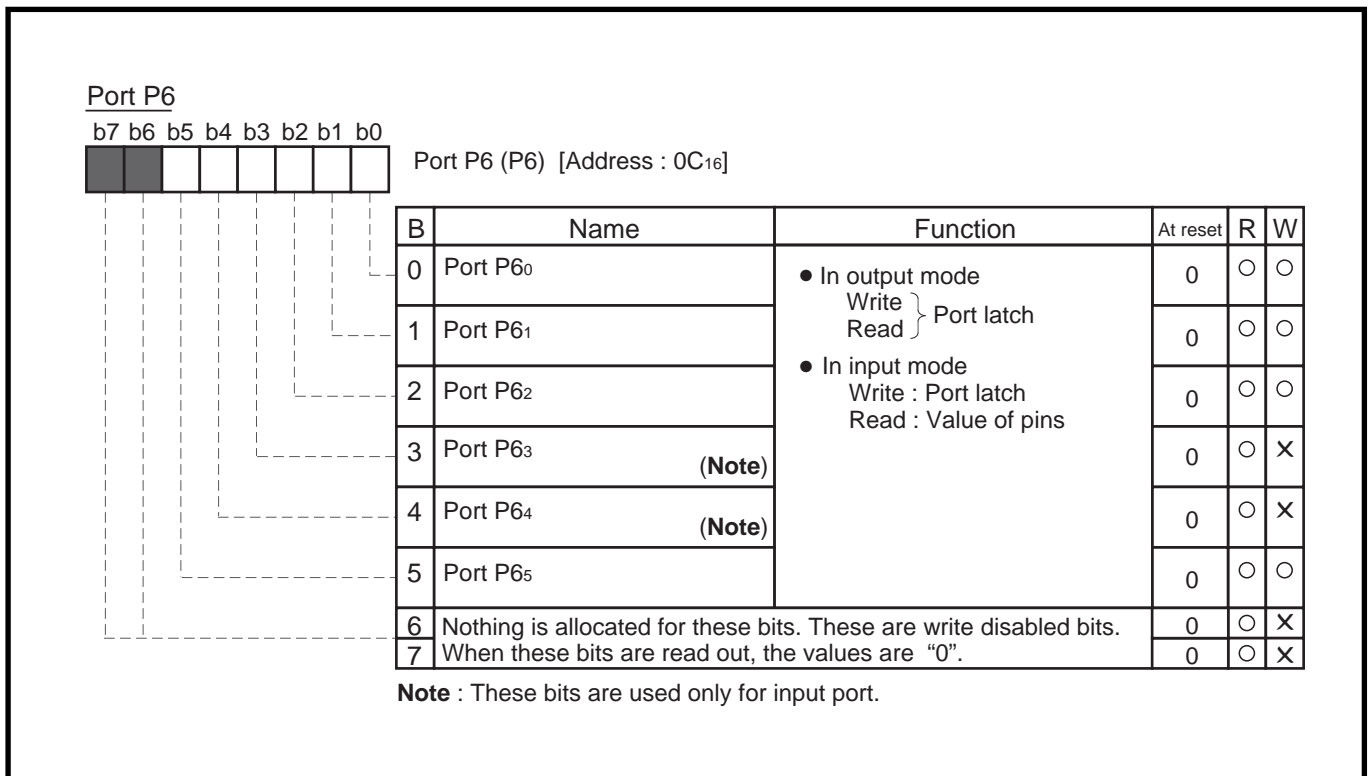


Fig. 2.1.4 Structure of Port P6

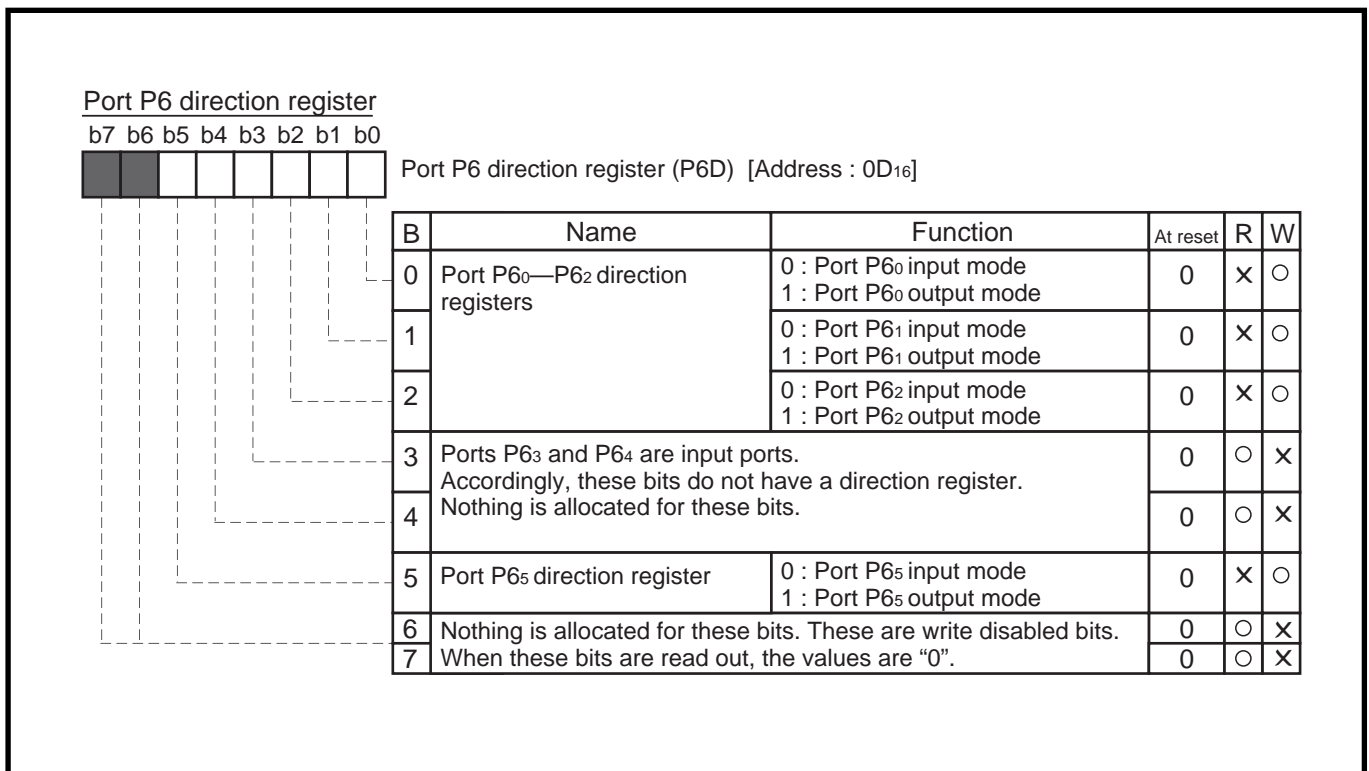


Fig. 2.1.5 Structure of Port P6 direction register

## 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, P8	<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>
ADVREF	Connect to Vss(GND) or open.
AVss	Connect to Vss(GND).
CMPVCC	Connect to Vss(GND).
CMPOUT	Open
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, P8	<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>
ADVREF	Connect to Vss(GND) or open.
$\overline{ONW}$	Connect to VCC through a resistor of 1 k $\Omega$ to 10 k $\Omega$ .
RESETOUT	Open
$\phi$	Open
SYNC	Open
AVss	Connect to Vss(GND).
CMPVCC	Connect to Vss(GND).
CMPOUT	Open
XOUT	Open (only when using external clock).



# APPLICATION

## 2.2 Timer

### 2.2 Timer

#### 2.2.1 Memory map of timer

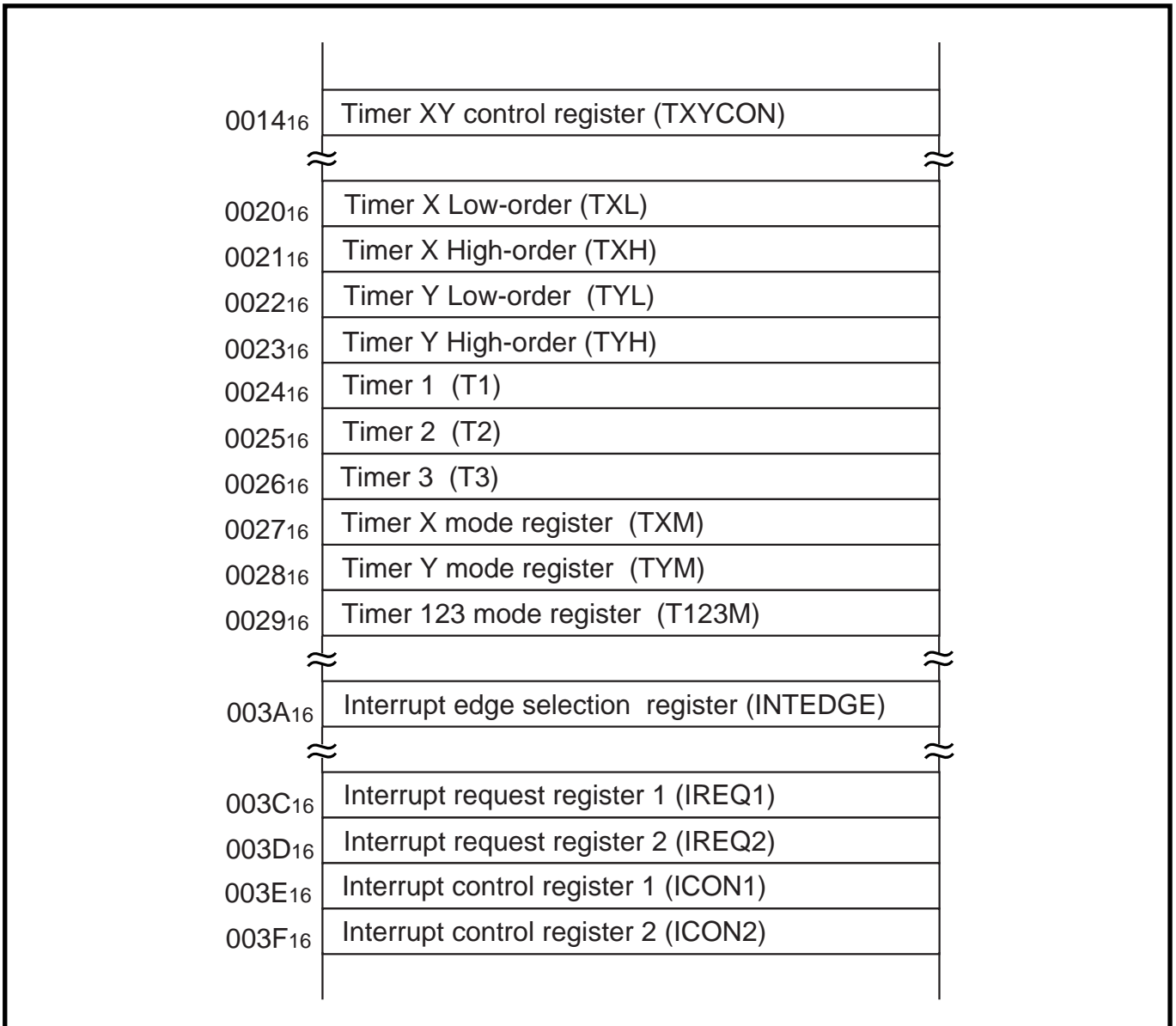


Fig. 2.2.1 Memory map of timer related registers

### 2.2.2 Related registers

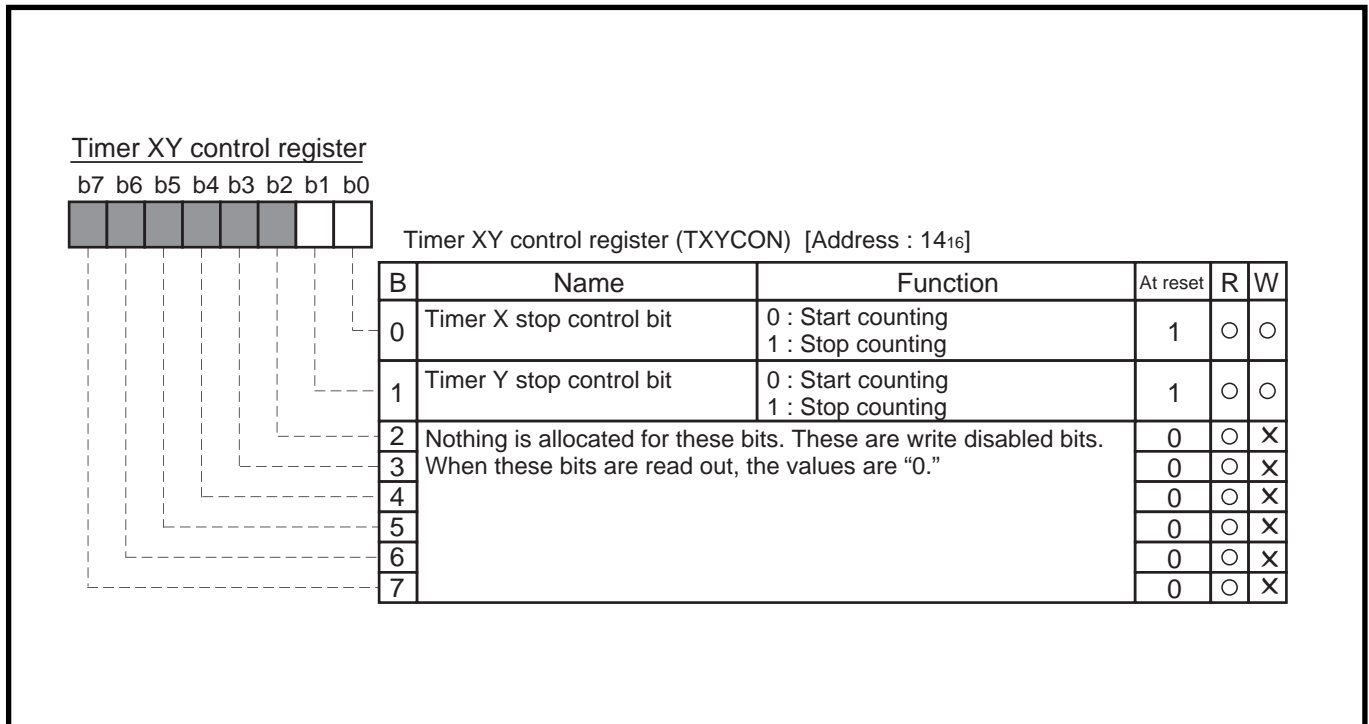


Fig. 2.2.2 Structure of Timer XY control register

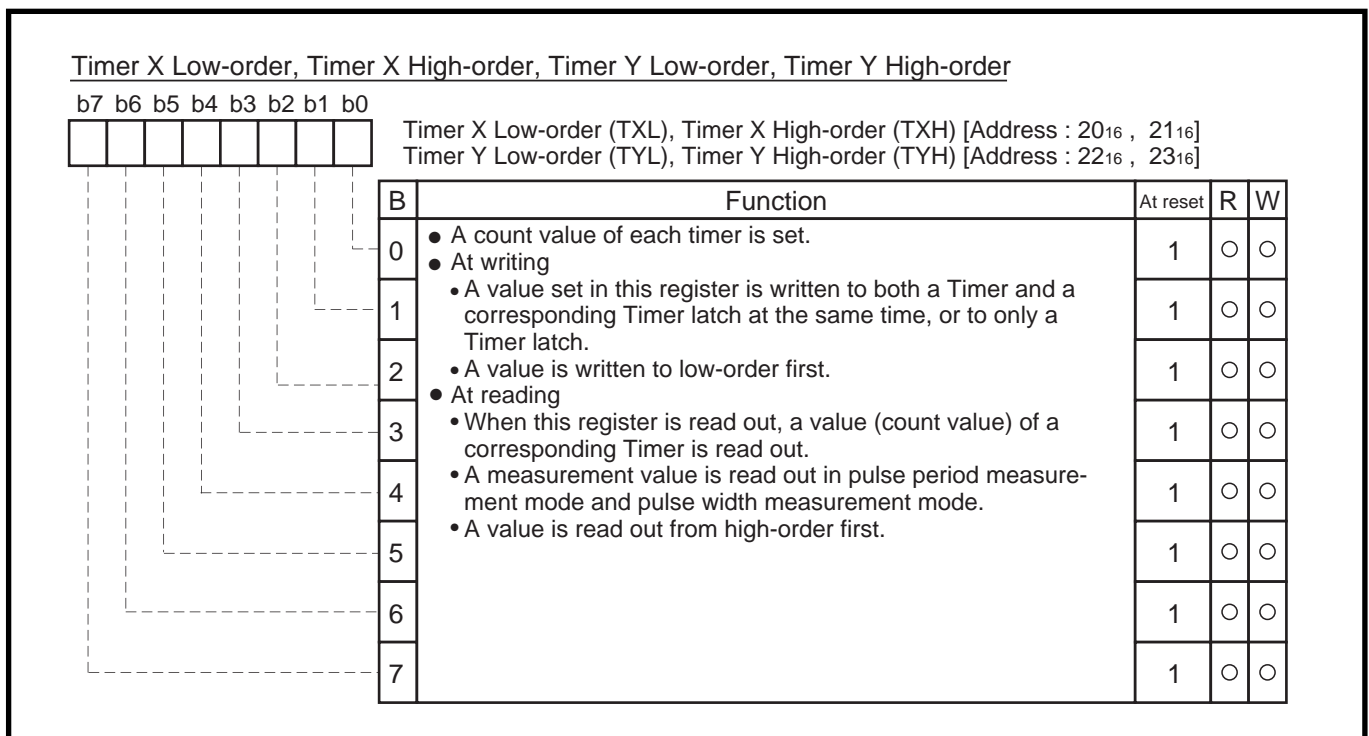


Fig. 2.2.3 Structure of Timer X Low-order, Timer X High-order, Timer Y Low-order, Timer Y High-order

# APPLICATION

## 2.2 Timer

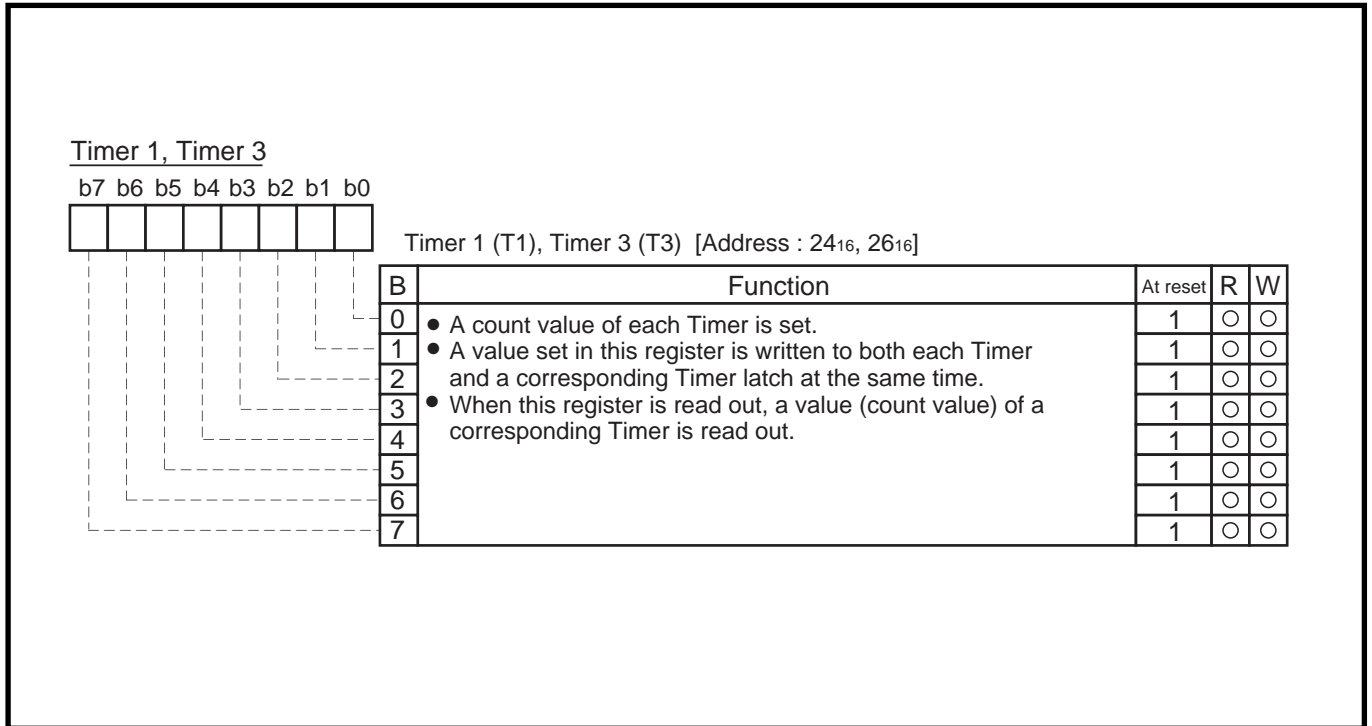


Fig. 2.2.4 Structure of Timer 1, Timer 3

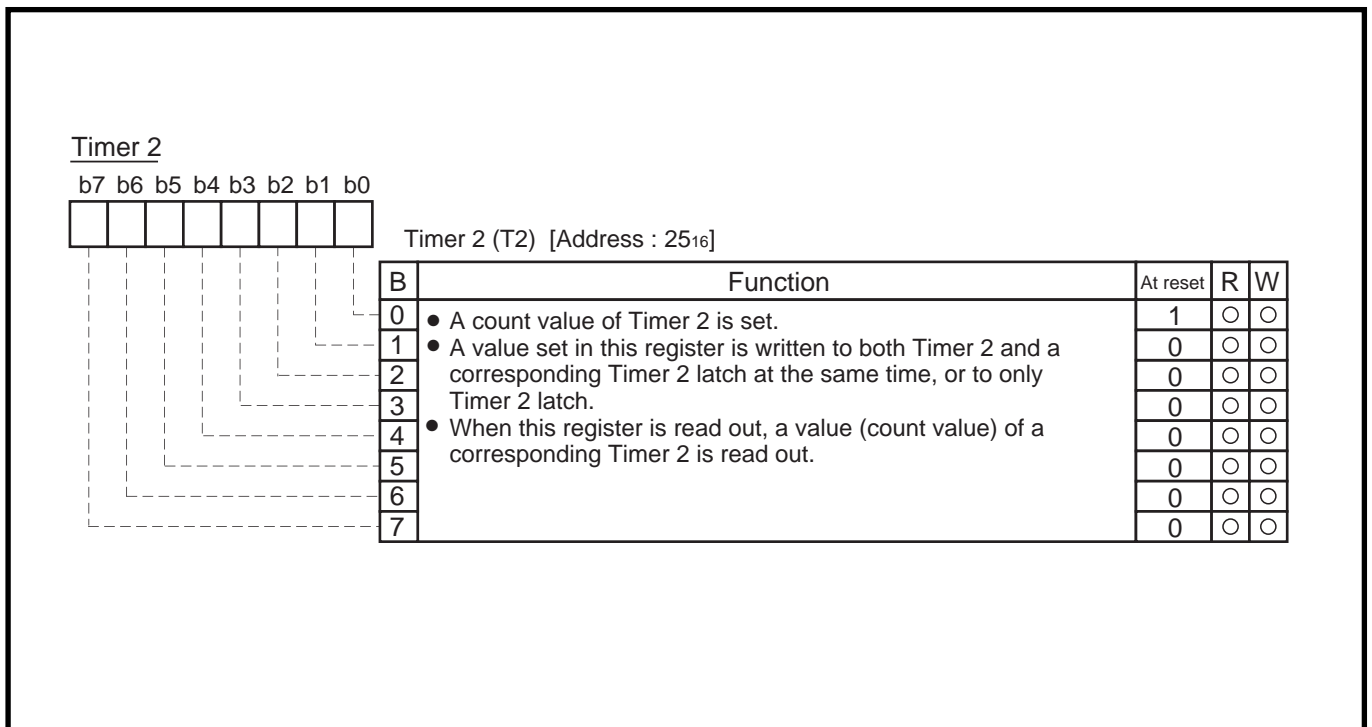


Fig. 2.2.5 Structure of Timer 2

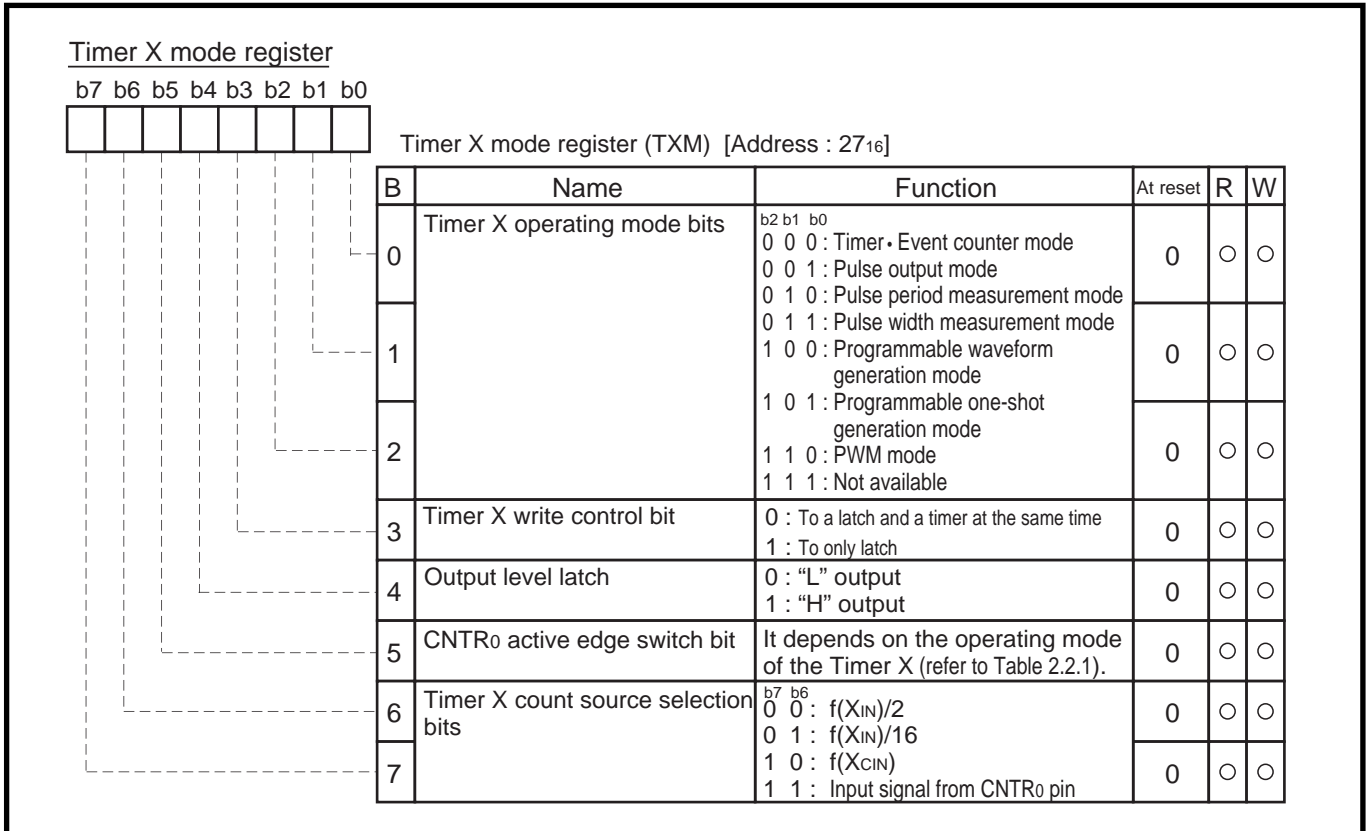


Fig. 2.2.6 Structure of Timer X mode register

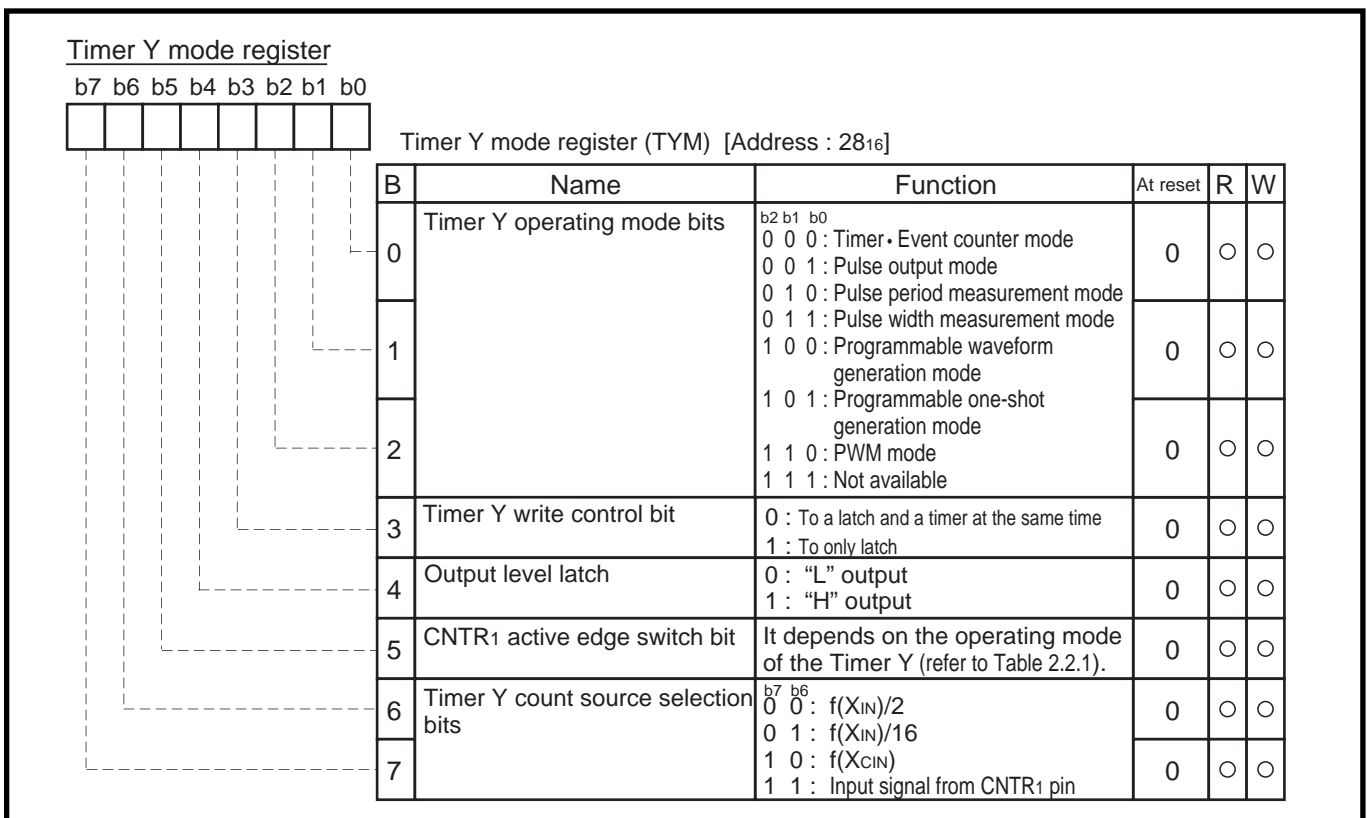


Fig. 2.2.7 Structure of Timer Y mode register

# APPLICATION

## 2.2 Timer

**Table. 2.2.1 Function of CNTR0/CNTR1 active edge switch bit**

Operating mode of Timer X/Timer Y	Function of CNTR0/CNTR1 edge switch bit (bit 5 of each address 27 <sub>16</sub> and 28 <sub>16</sub> )	
Timer mode	"0"	<ul style="list-style-type: none"> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge (No effect on timer count)</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge (No effect on timer count)</li> </ul>
Event counter mode	"0"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Count at rising edge</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Count at falling edge</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge</li> </ul>
Pulse output mode	"0"	<ul style="list-style-type: none"> <li>• Start of pulse output : From "H" level</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Start of pulse output : From "L" level</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge</li> </ul>
Pulse period measurement mode	"0"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Measurement of a period between a falling edge and the next falling edge</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Measurement of a period between a rising edge and the next rising edge</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge</li> </ul>
Pulse width measurement mode	"0"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Measurement of "H" level width</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Measurement of "L" level width</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge</li> </ul>
Programmable one-shot generation mode	"0"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Start of a pulse output at "L" level, and output of an one-shot "H" level pulse</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Start of a pulse output at "H" level, and output of an one-shot "L" level pulse</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge</li> </ul>

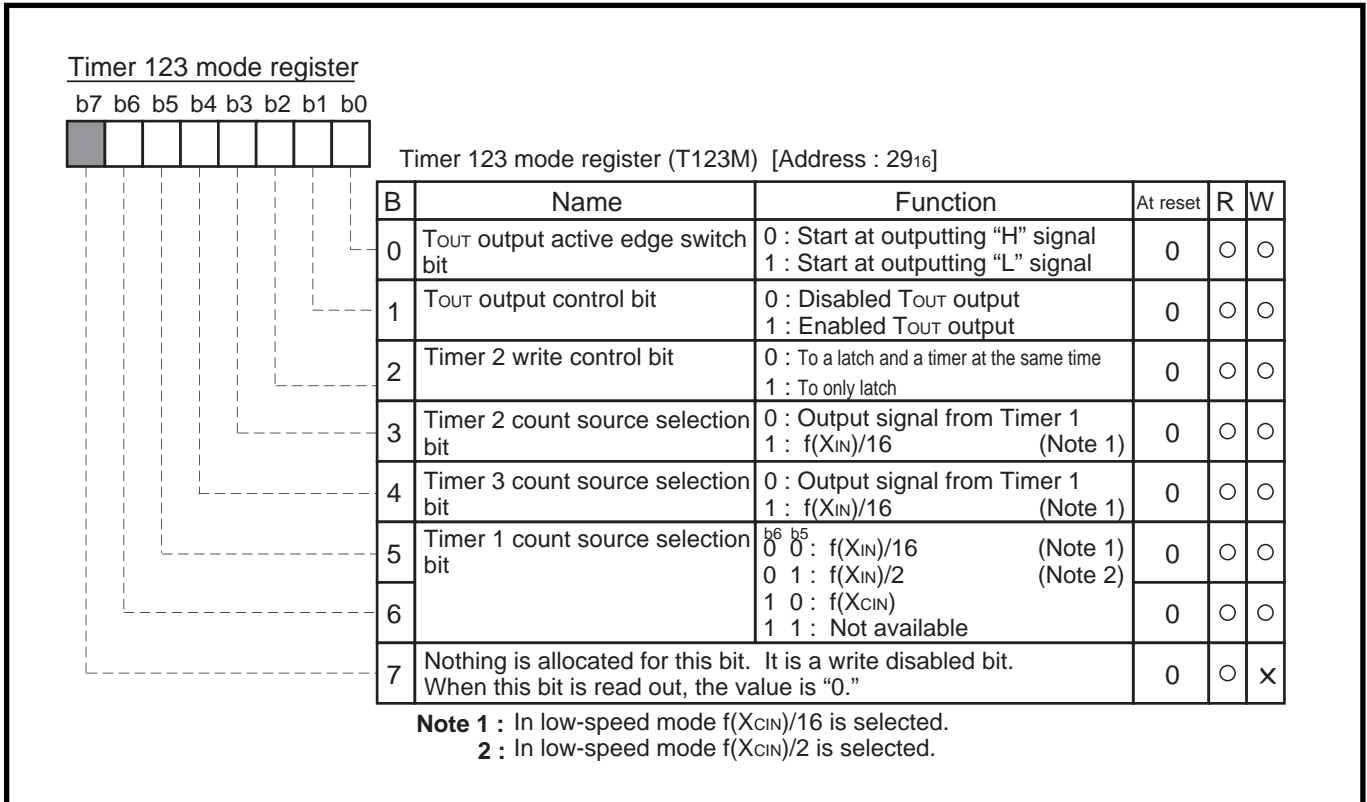


Fig. 2.2.8 Structure of Timer 123 mode register

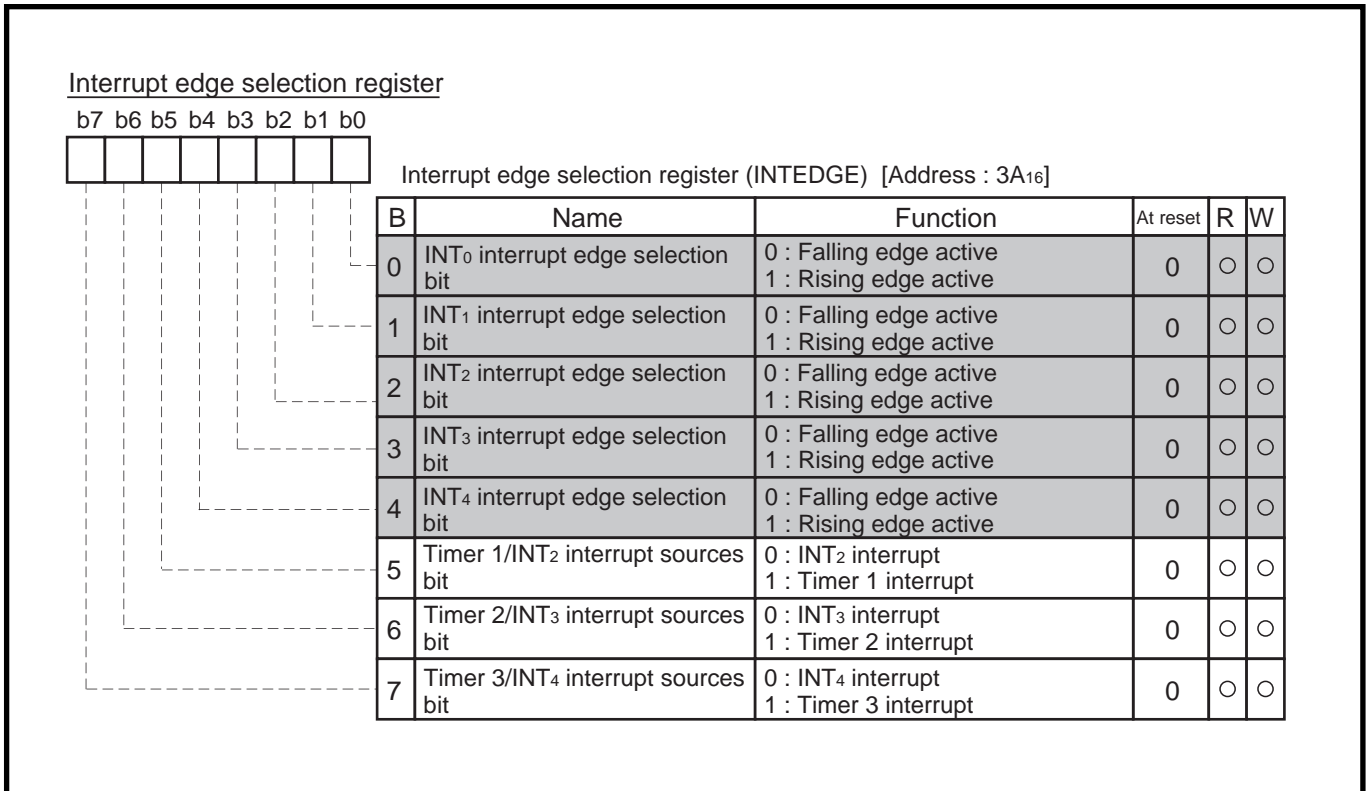


Fig. 2.2.9 Structure of Interrupt edge selection register

# APPLICATION

## 2.2 Timer

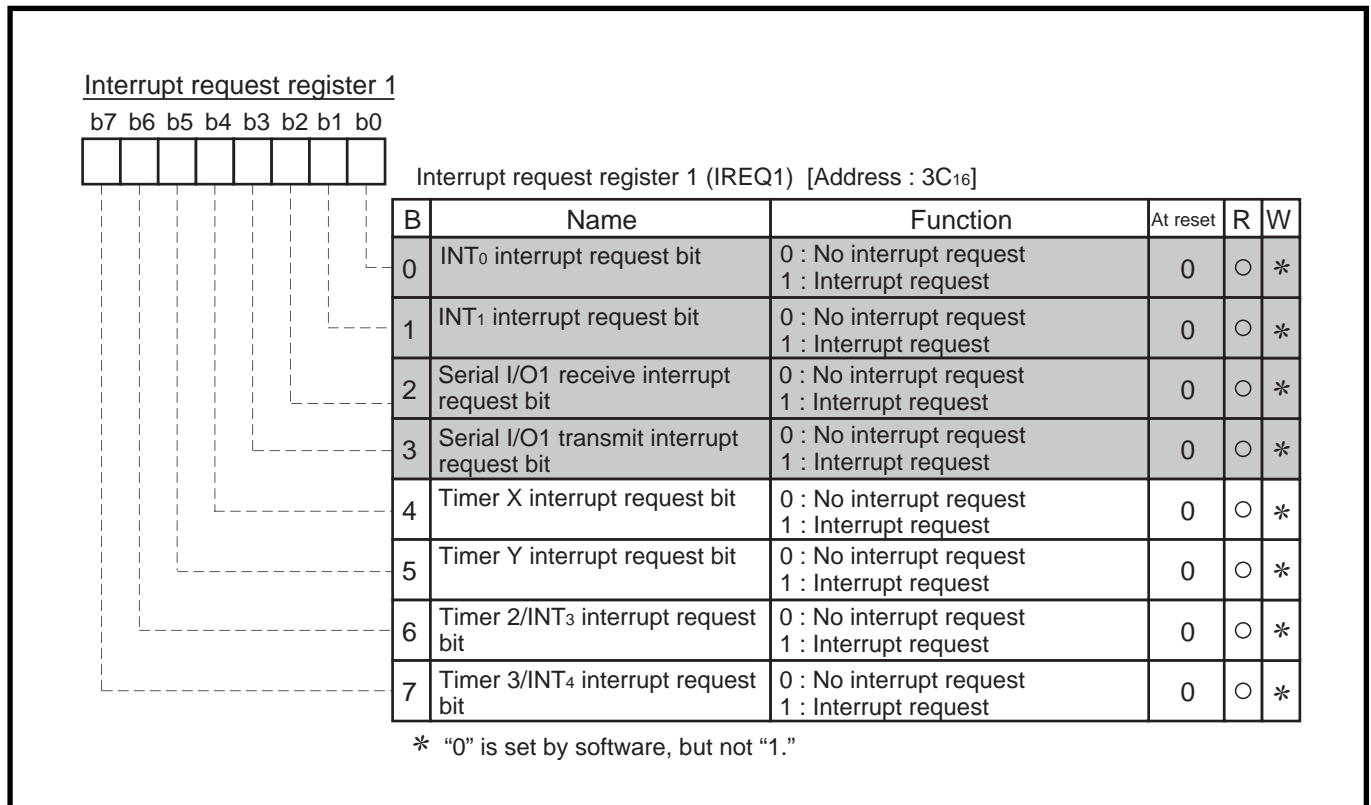


Fig. 2.2.10 Structure of Interrupt request register 1

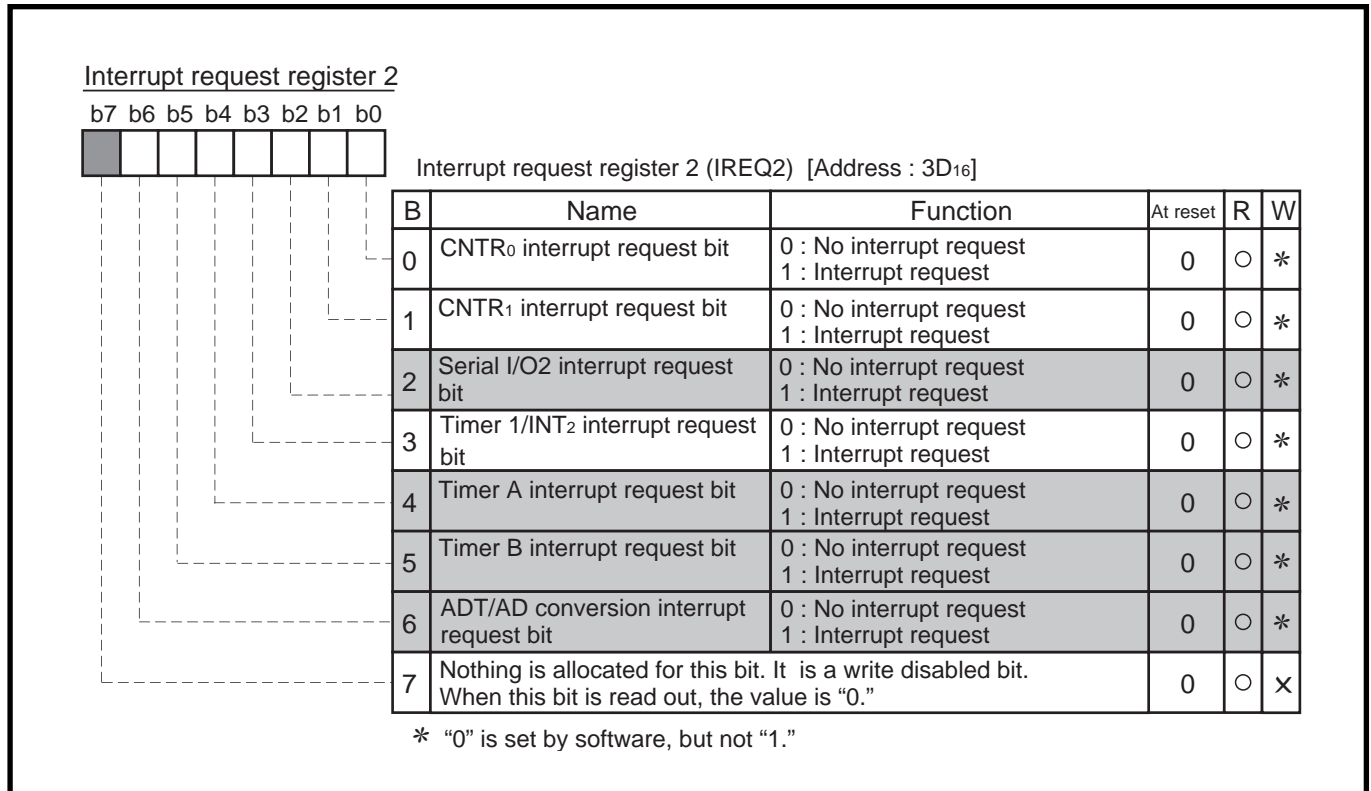
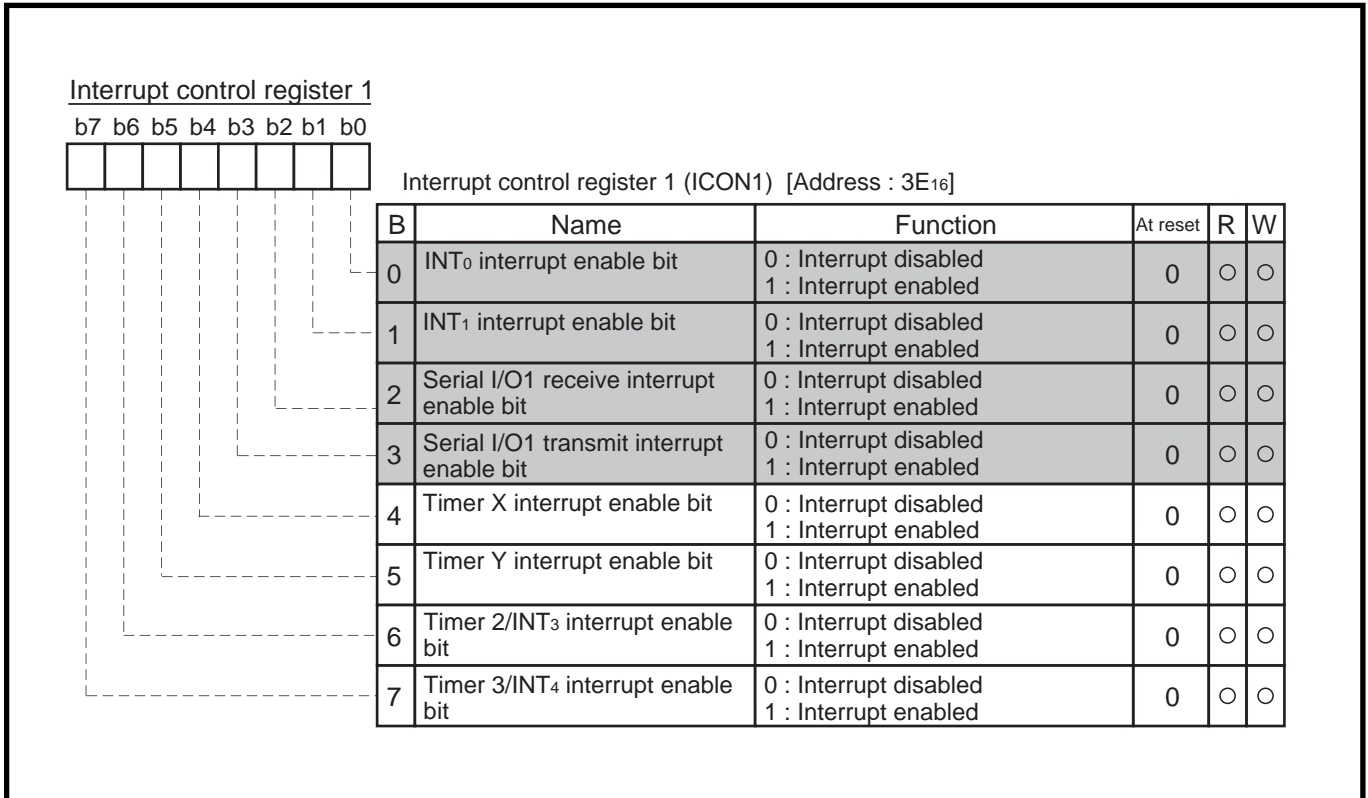
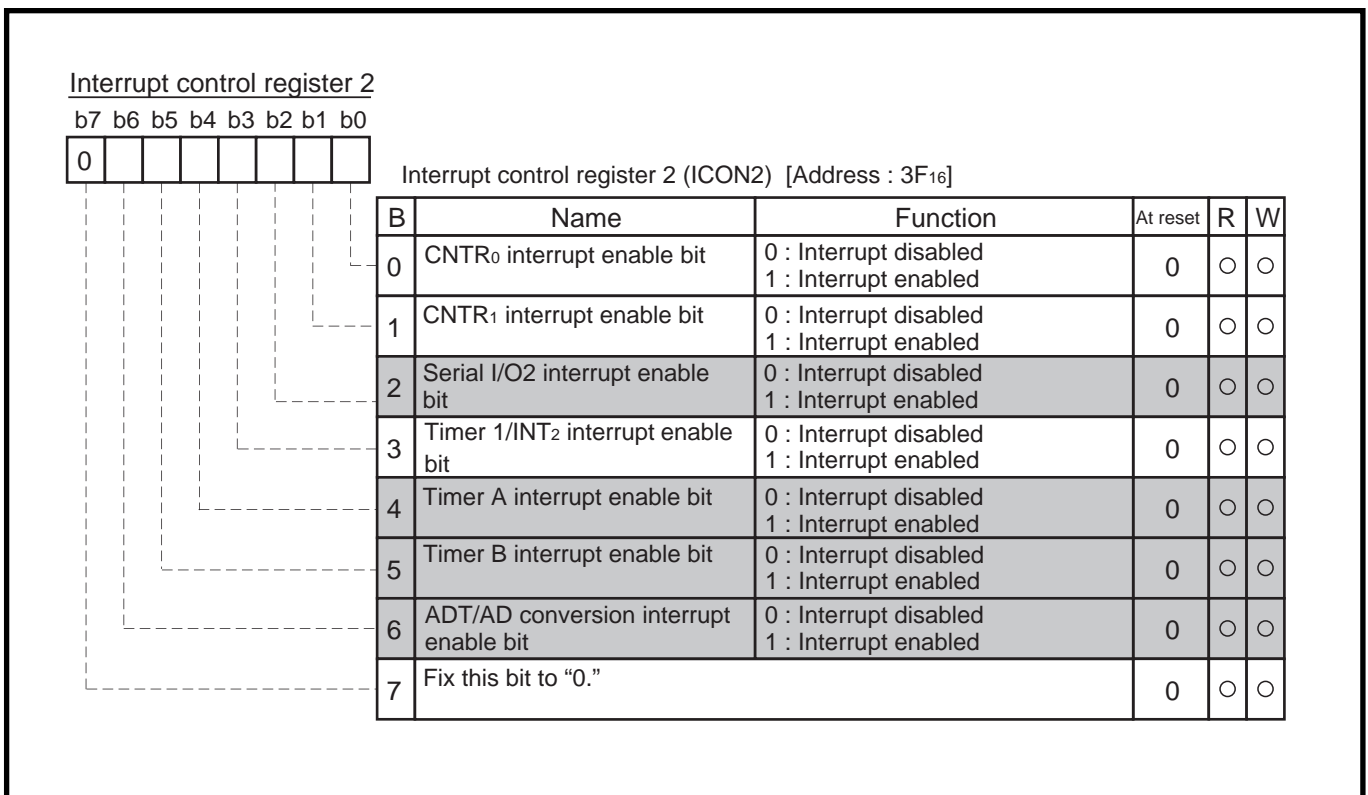


Fig. 2.2.11 Structure of Interrupt request register 2



**Fig. 2.2.12 Structure of Interrupt control register 1**



**Fig. 2.2.13 Structure of Interrupt control register 2**



# APPLICATION

## 2.2 Timer

---

### 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, Timer 3)**

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, Timer 3)**

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 25m second) → Application example 1
  - Control of a main routine cycle

**[Function 3] Output of Rectangular waveform (Timer X, Timer Y, Timer 2)**

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.

### (2) Timer application example 1 : Clock function (measurement of 25 ms)

**Outline :** The input clock is divided by a timer so that the clock counts up every 25 ms.

**Specifications :**

- The clock  $f(X_{IN}) = 8 \text{ MHz}$  is divided by a timer.
- The clock is counted at intervals of 25 ms by the Timer 3 interrupt.

Figure 2.2.14 shows a connection of timers and a setting of division ratios, Figures 2.2.15 show a setting of related registers, and Figure 2.2.16 shows a control procedure.

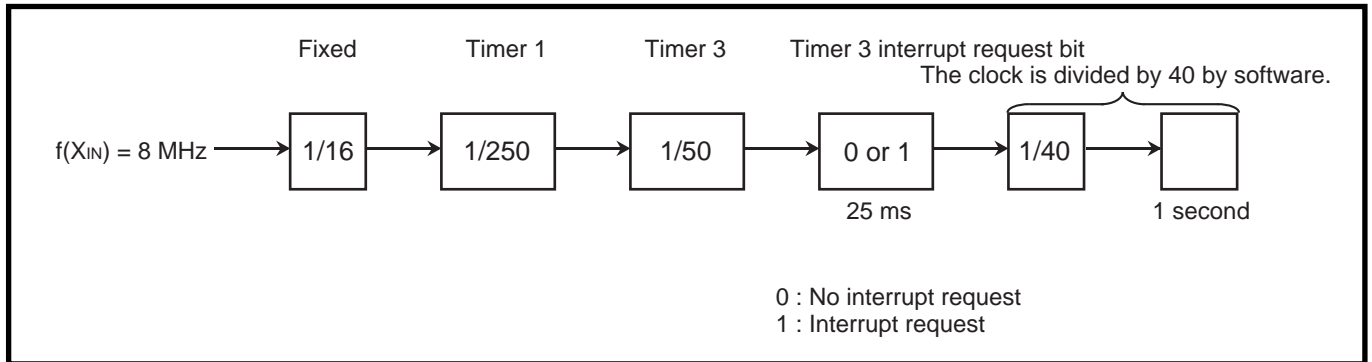


Fig. 2.2.14 Connection of timers and setting of division ratios [Clock function]

# APPLICATION

## 2.2 Timer

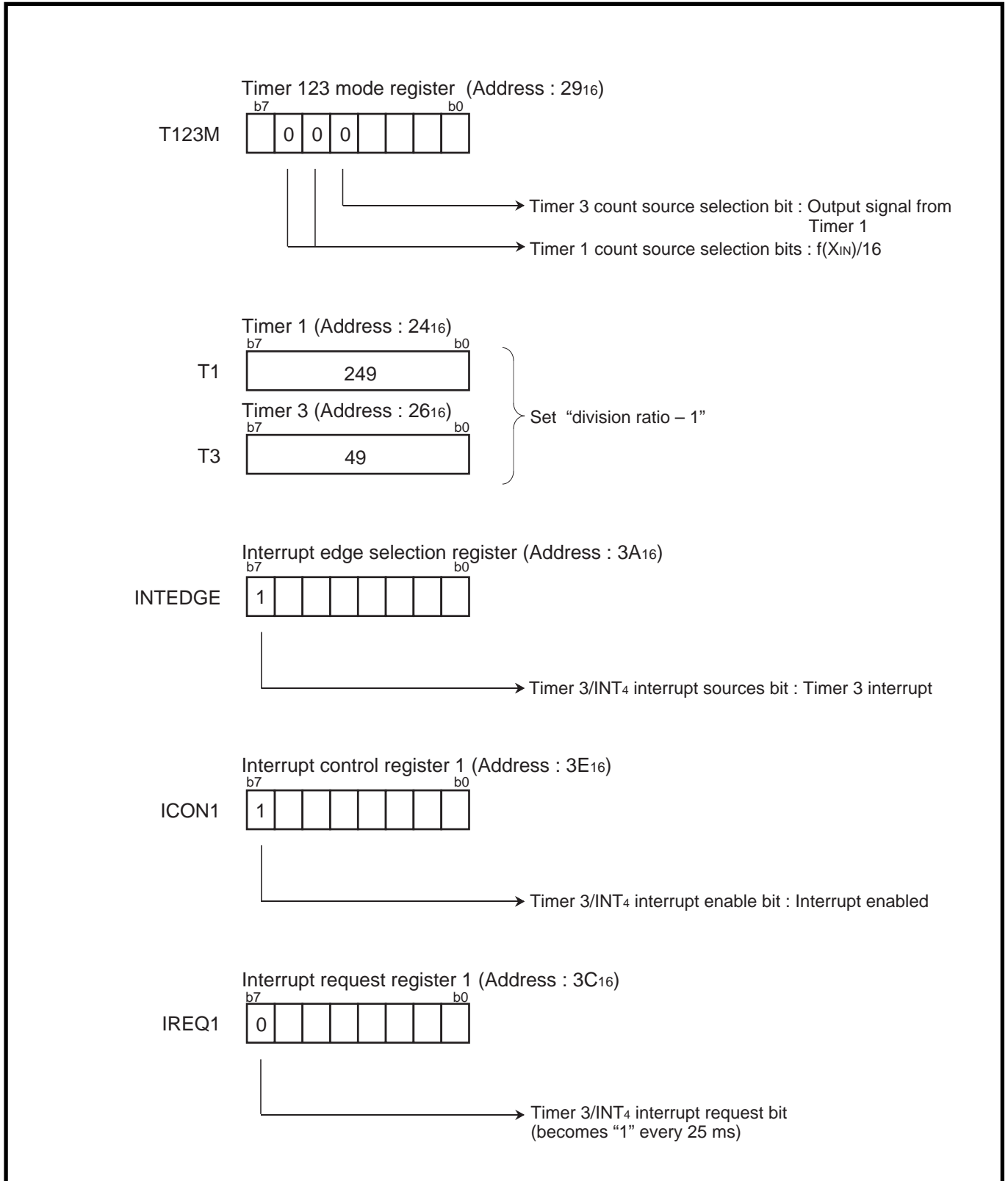
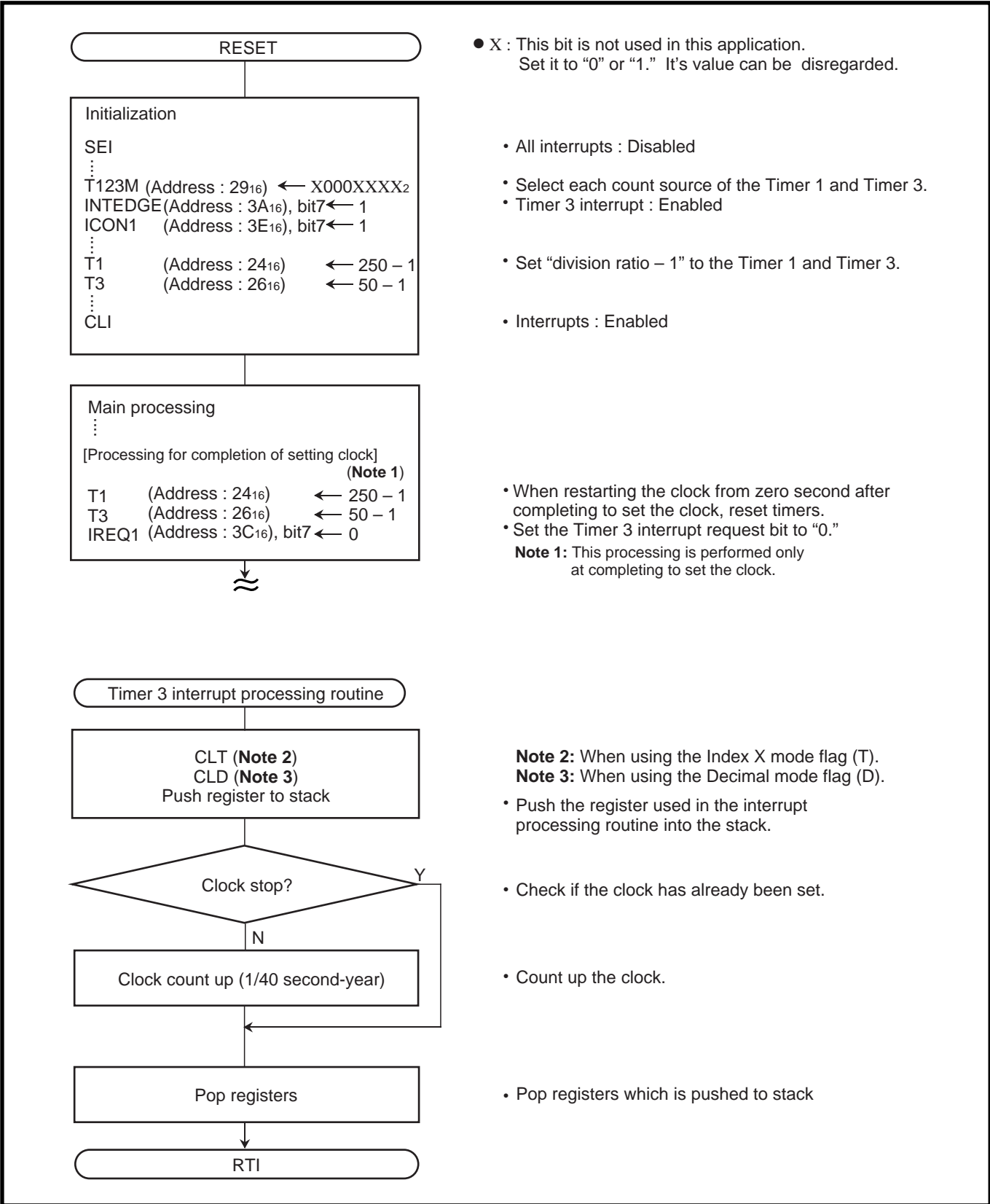


Fig. 2.2.15 Setting of related registers [Clock function]

Control procedure :

Figure 2.2.16 shows a control procedure.



• X : This bit is not used in this application. Set it to "0" or "1." It's value can be disregarded.

- All interrupts : Disabled
- Select each count source of the Timer 1 and Timer 3.
- Timer 3 interrupt : Enabled
- Set "division ratio - 1" to the Timer 1 and Timer 3.
- Interrupts : Enabled

- When restarting the clock from zero second after completing to set the clock, reset timers.
- Set the Timer 3 interrupt request bit to "0."

Note 1: This processing is performed only at completing to set the clock.

Note 2: When using the Index X mode flag (T).  
Note 3: When using the Decimal mode flag (D).

- Push the register used in the interrupt processing routine into the stack.
- Check if the clock has already been set.
- Count up the clock.
- Pop registers which is pushed to stack

Fig. 2.2.16 Control procedure [Clock function]

# APPLICATION

## 2.2 Timer

### (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  $f(X_{IN}) = 8 \text{ MHz}$  into about 2 kHz (2049 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 2.2.17 shows an example of a peripheral circuit, and Figure 2.2.18 shows a connection of the timer and setting of the division ratio.

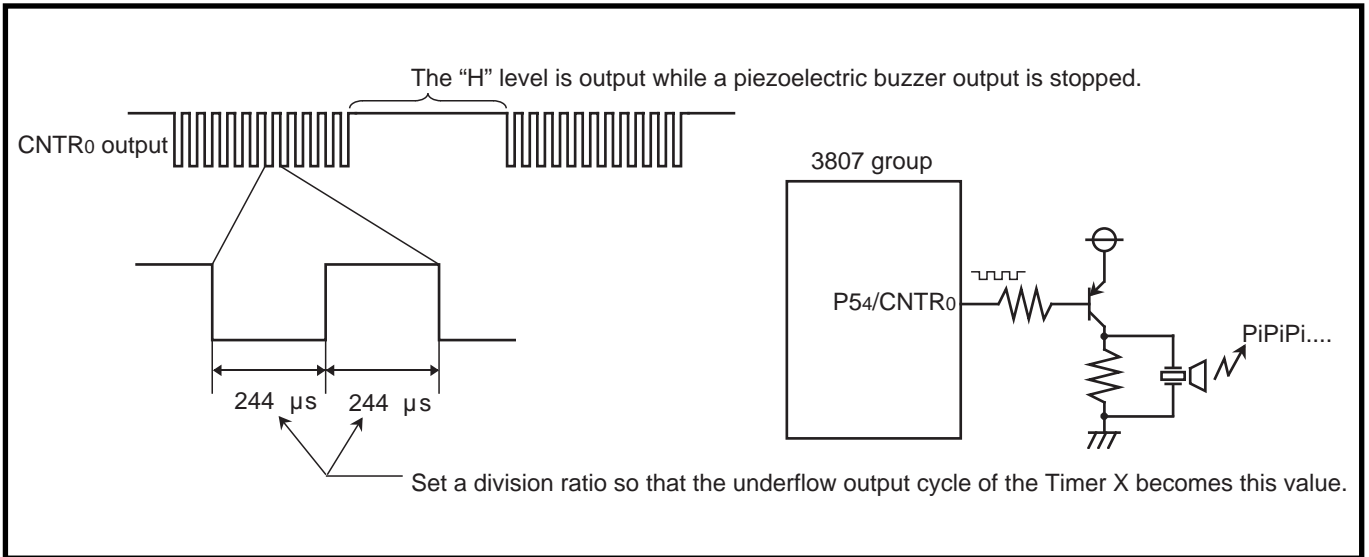


Fig. 2.2.17 Example of a peripheral circuit

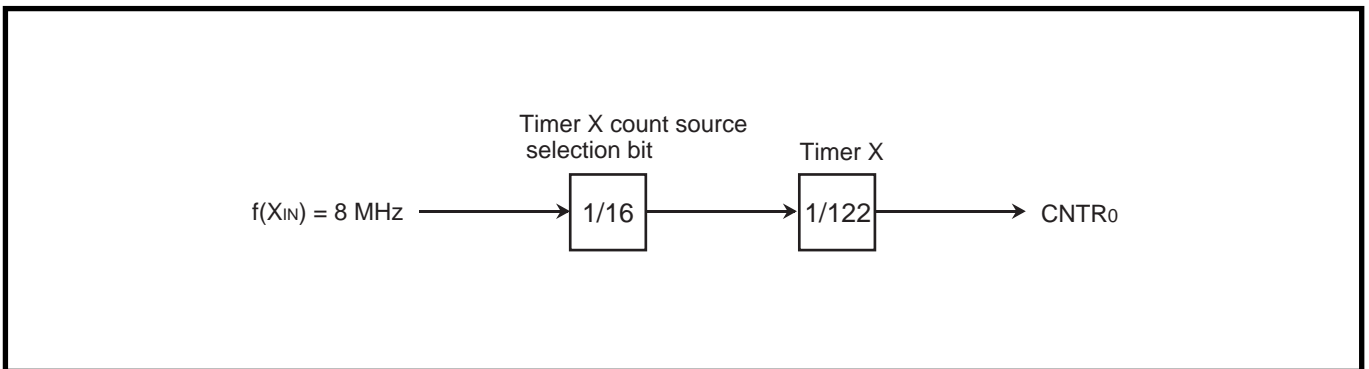


Fig. 2.2.18 Connection of the timer and setting of the division ratio [Piezoelectric buzzer output]

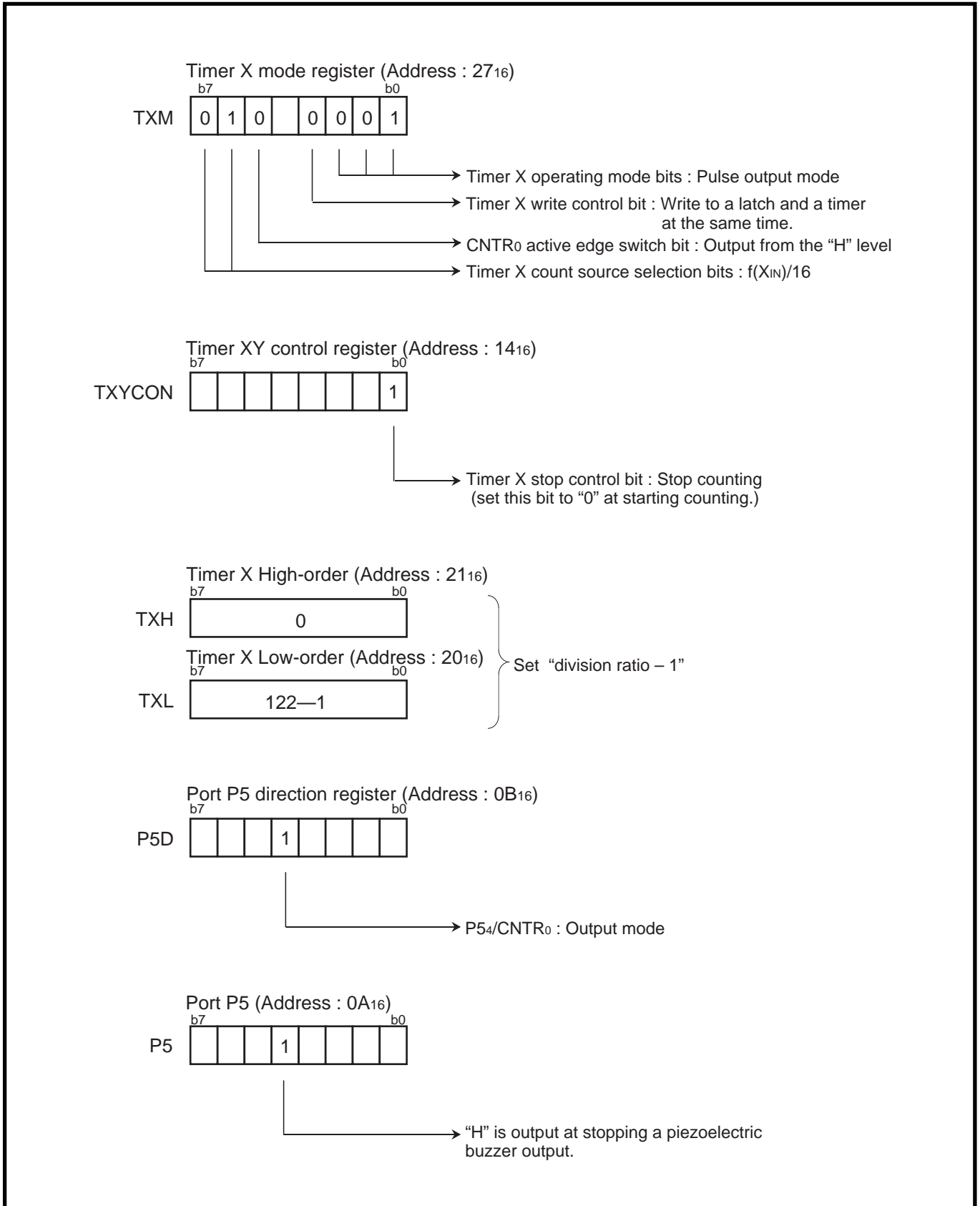


Fig. 2.2.19 Setting of related registers [Piezoelectric buzzer output]

# APPLICATION

## 2.2 Timer

### Control procedure :

Figure 2.2.20 shows a control procedure.

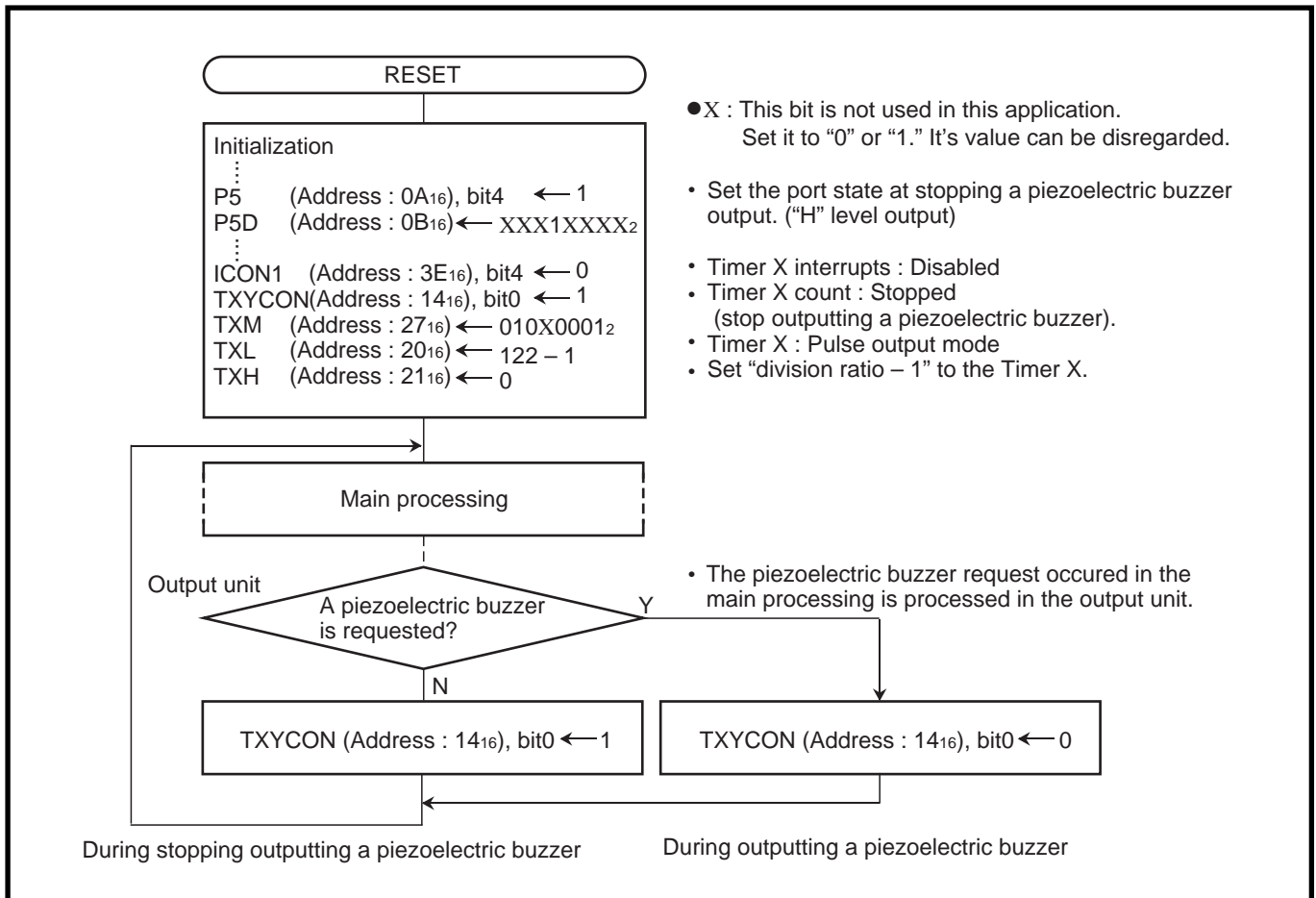


Fig. 2.2.20 Control procedure [Piezoelectric buzzer output]

**(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.

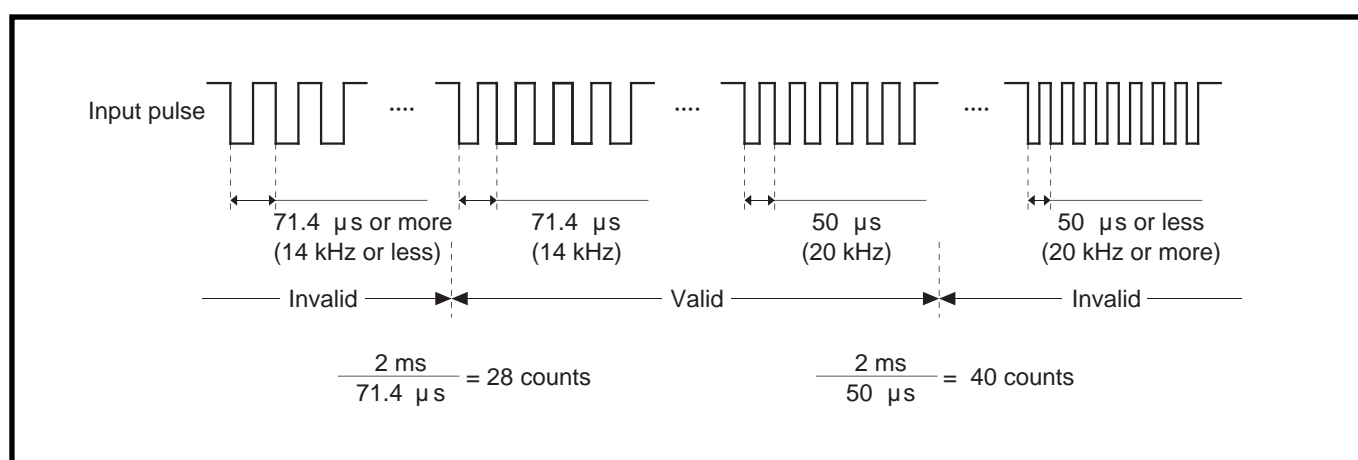
- A value counted a pulse which is input to P55/CNTR1 pin by a timer.
- A reference value

**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 X interrupt interval). 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 to 215 = 255 (initialized value of counter) – 28 to 40 (the number of valid value).

Figure 2.2.21 shows a method for judging if input pulse exists, and Figure 2.2.22 and Figure 2.2.23 show a setting of related registers.



**Fig 2.2.21 A method for judging if input pulse exists**



# APPLICATION

## 2.2 Timer

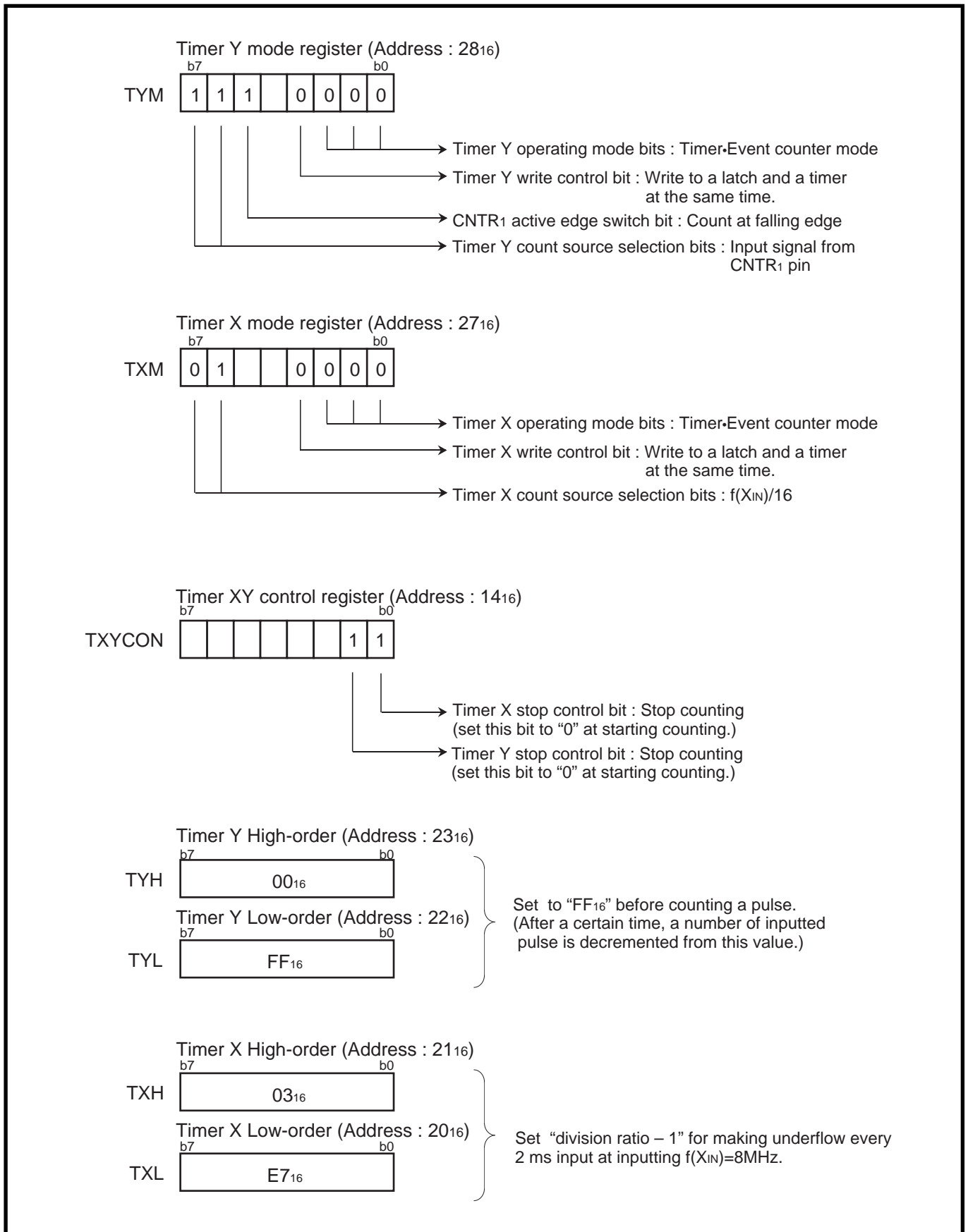
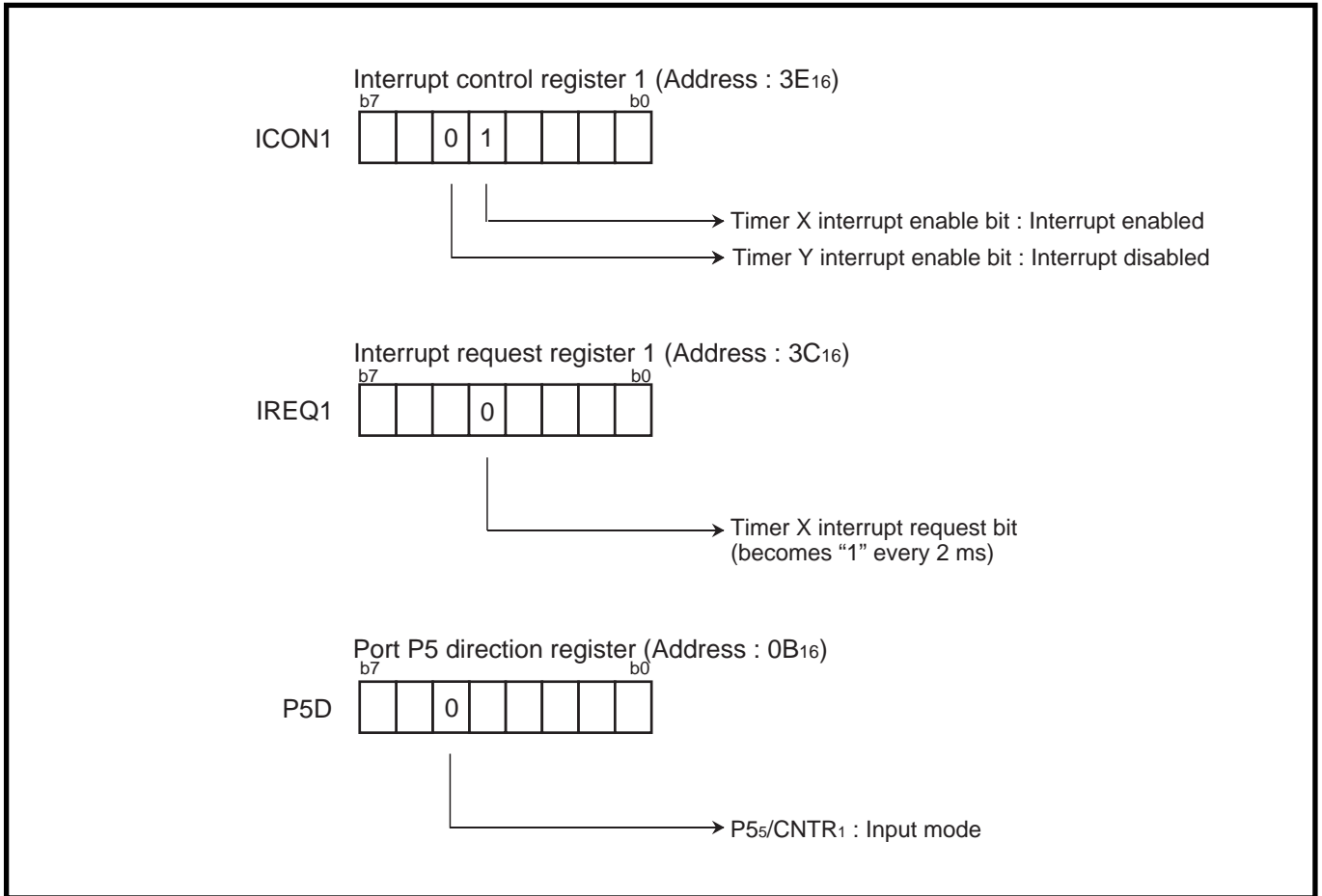


Fig. 2.2.22 Setting of related registers (1) [Measurement of frequency]



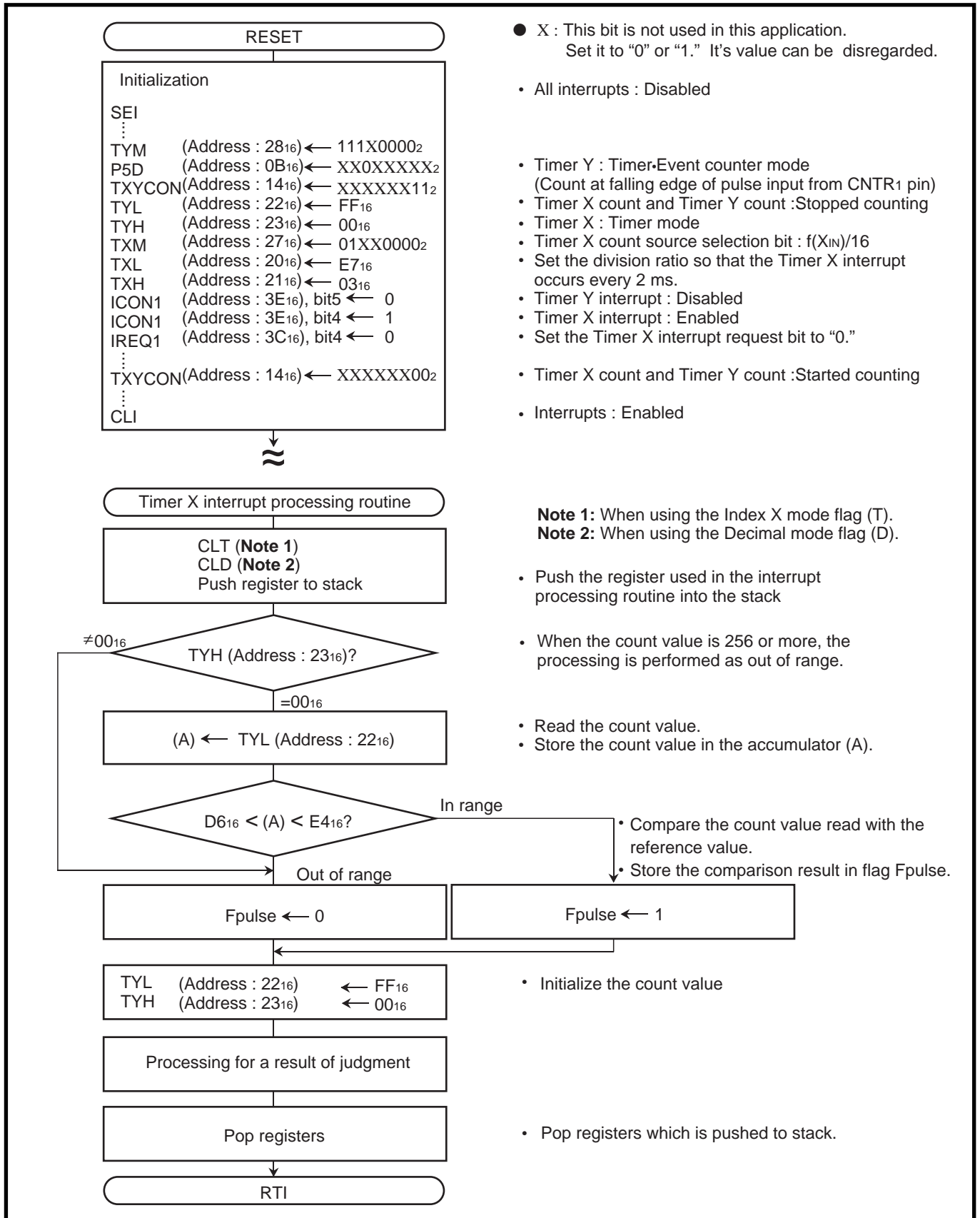
**Fig. 2.2.23 Setting of related registers (2) [Measurement of frequency]**

# APPLICATION

## 2.2 Timer

### Control procedure :

Figure 2.2.24 shows a control procedure.



● X : This bit is not used in this application.  
Set it to "0" or "1." It's value can be disregarded.

- All interrupts : Disabled
- Timer Y : Timer-Event counter mode  
(Count at falling edge of pulse input from CNTR1 pin)
- Timer X count and Timer Y count : Stopped counting
- Timer X : Timer mode
- Timer X count source selection bit :  $f(X_{IN})/16$
- Set the division ratio so that the Timer X interrupt occurs every 2 ms.
- Timer Y interrupt : Disabled
- Timer X interrupt : Enabled
- Set the Timer X interrupt request bit to "0."
- Timer X count and Timer Y count : Started counting
- Interrupts : Enabled

**Note 1:** When using the Index X mode flag (T).  
**Note 2:** When using the Decimal mode flag (D).

- Push the register used in the interrupt processing routine into the stack
- When the count value is 256 or more, the processing is performed as out of range.
- Read the count value.
- Store the count value in the accumulator (A).
- Compare the count value read with the reference value.
- Store the comparison result in flag Fpulse.
- Initialize the count value
- Pop registers which is pushed to stack.

Fig. 2.2.24 Control procedure [Measurement of frequency]

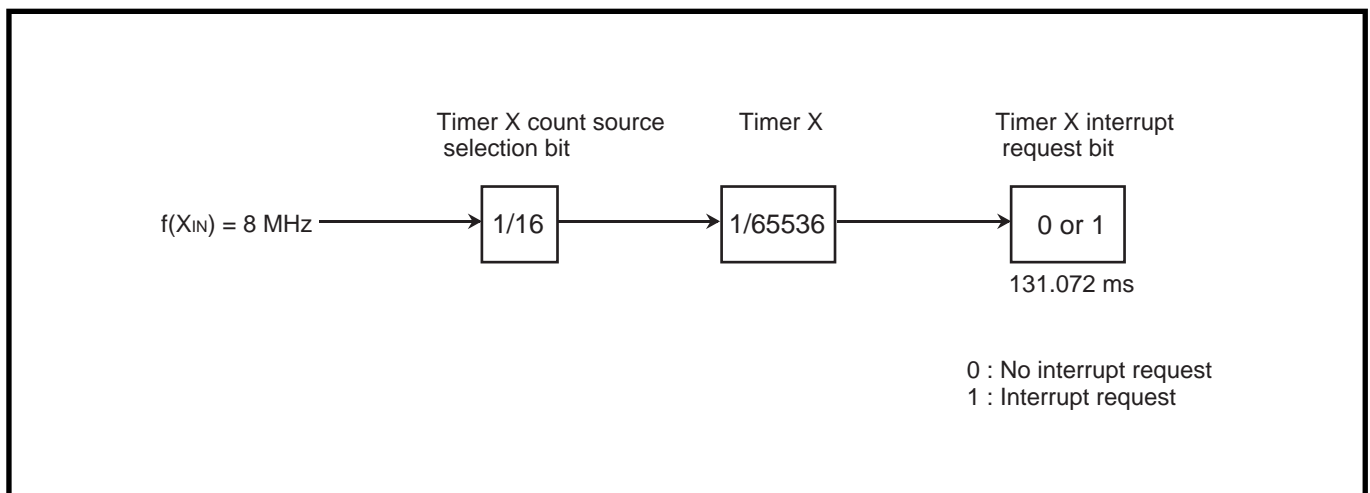
**(5) Timer application example 4 : Measurement of pulse width of FG pulse generated by motor**

**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 FG pulse input to the P54/CNTR0 pin is counted by Timer X.

(Example : When the clock frequency is 8 MHz, the count source would be  $2 \mu\text{s}$  that is obtained by dividing the clock frequency by 16. Measurement can be made up to 131.072 ms in the range of  $\text{FFFF}_{16}$  to  $0000_{16}$ .)

Figure 2.2.25 shows a connection of the timer and setting of the division ratio, and Figure 2.2.26 shows a setting of related registers.



**Fig. 2.2.25 Connection of the timer and setting of the division ratio [Measurement of pulse width]**

# APPLICATION

## 2.2 Timer

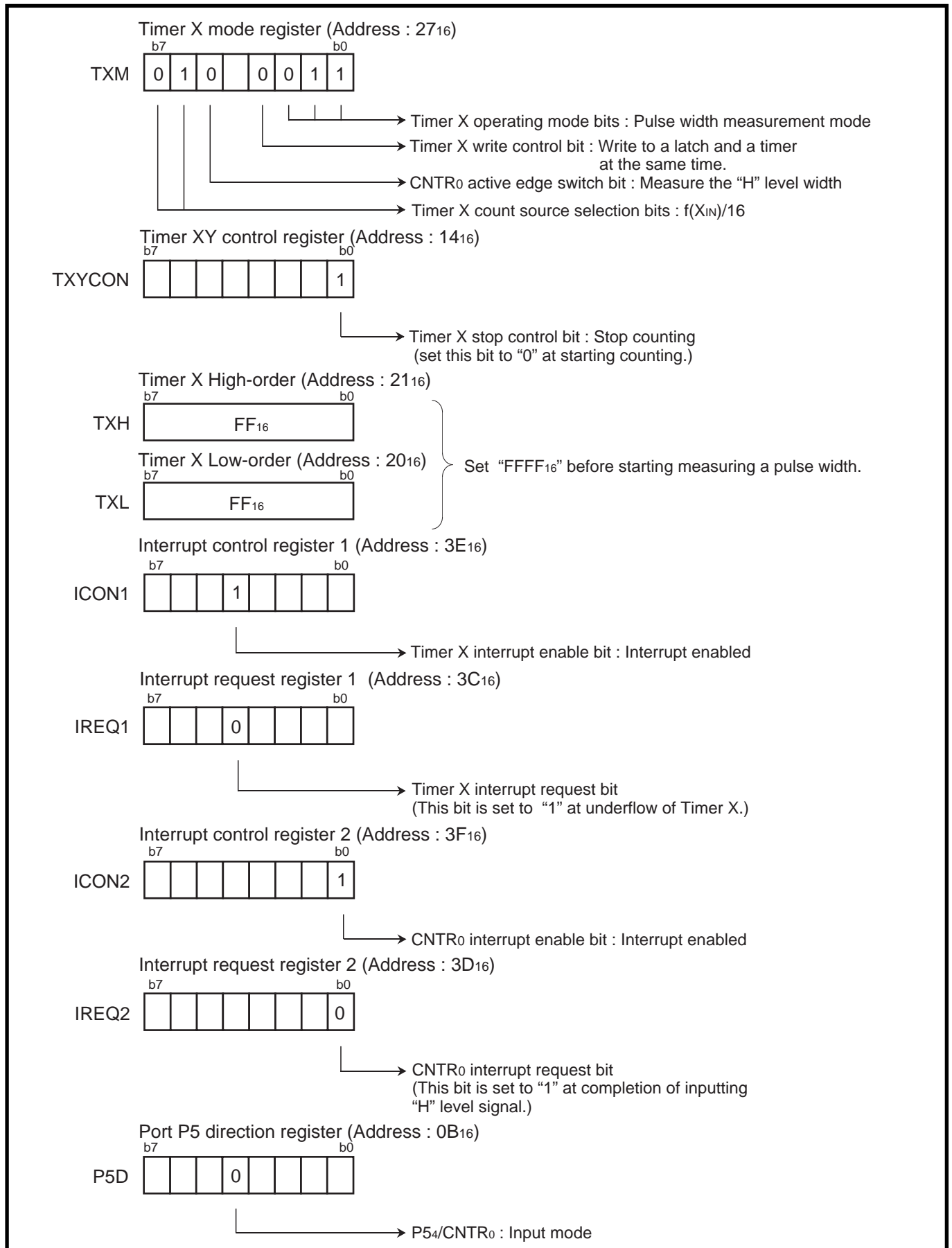


Fig. 2.2.26 Setting of related registers [Measurement of pulse width]

Figure 2.2.27 and Figure 2.2.28 show a control procedure.

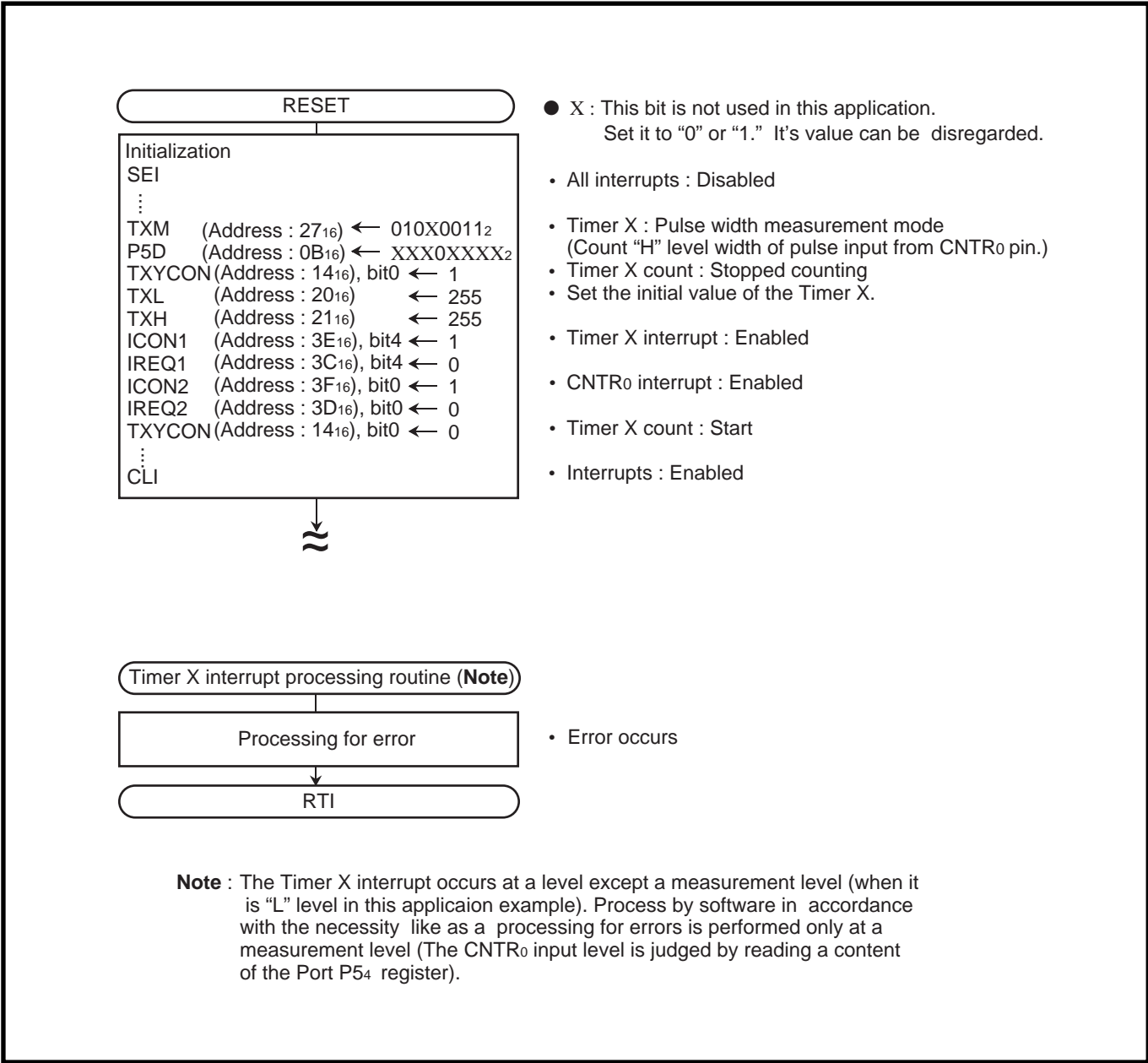
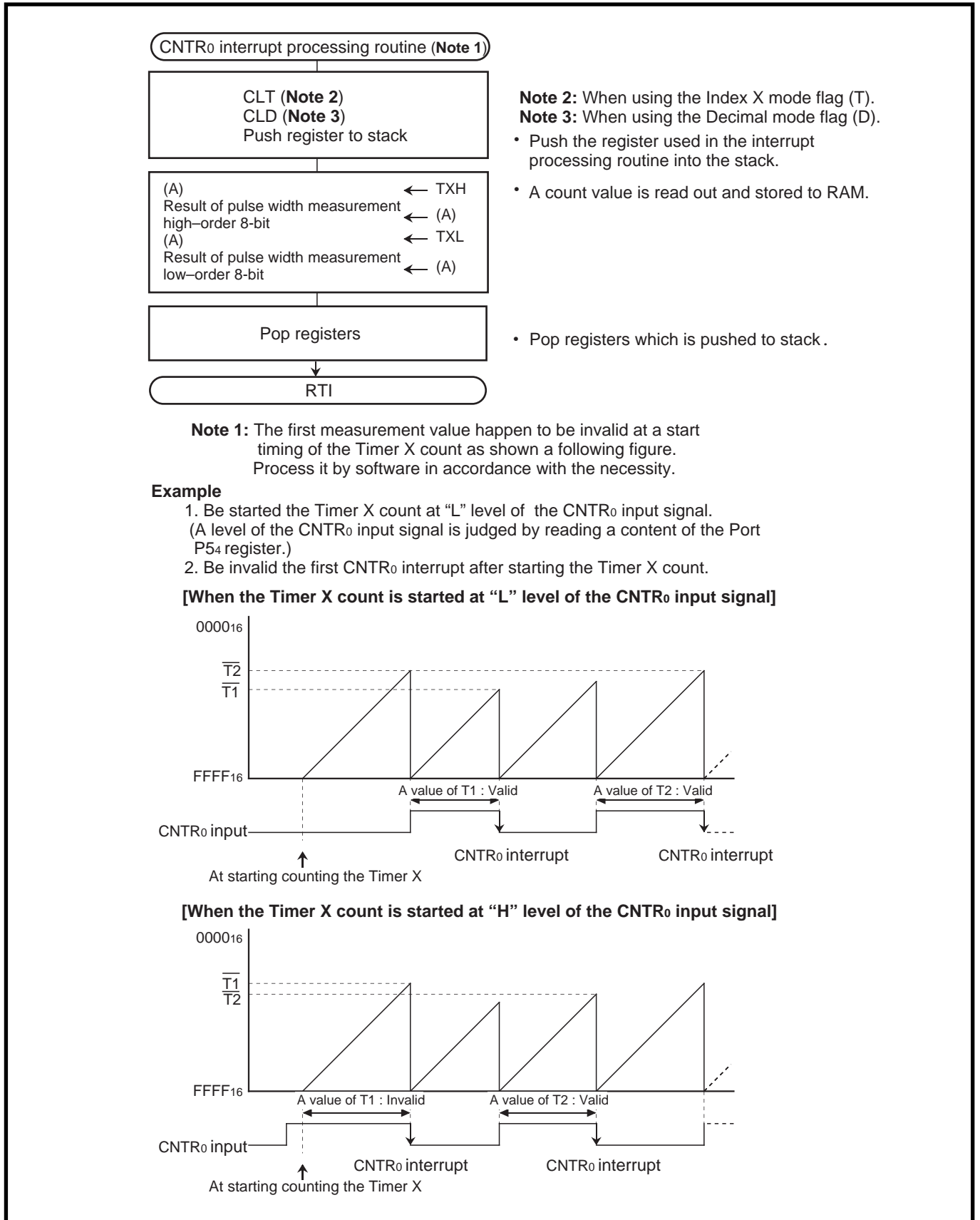


Fig. 2.2.27 Control procedure (1) [Measurement of pulse width]

# APPLICATION

## 2.2 Timer



## 2.3 Serial I/O

### 2.3.1 Memory map of serial I/O

0018 <sub>16</sub>	Transmit/Receive buffer register (TB/RB)
0019 <sub>16</sub>	Serial I/O1 status register (SIO1STS)
001A <sub>16</sub>	Serial I/O1 control register (SIO1CON)
001B <sub>16</sub>	UART control register (UARTCON)
001C <sub>16</sub>	Baud rate generator (BRG)
001D <sub>16</sub>	Serial I/O2 control register 1 (SIO2CON1)
001E <sub>16</sub>	Serial I/O2 control register 2 (SIO2CON2)
001F <sub>16</sub>	Serial I/O2 register (SIO2)
≈	≈
003A <sub>16</sub>	Interrupt edge selection register (INTEDGE)
≈	≈
003C <sub>16</sub>	Interrupt request register 1 (IREQ1)
003D <sub>16</sub>	Interrupt request register 2 (IREQ2)
003E <sub>16</sub>	Interrupt control register 1 (ICON1)
003F <sub>16</sub>	Interrupt control register 2 (ICON2)

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



# APPLICATION

## 2.3 Serial I/O

### 2.3.2 Related registers

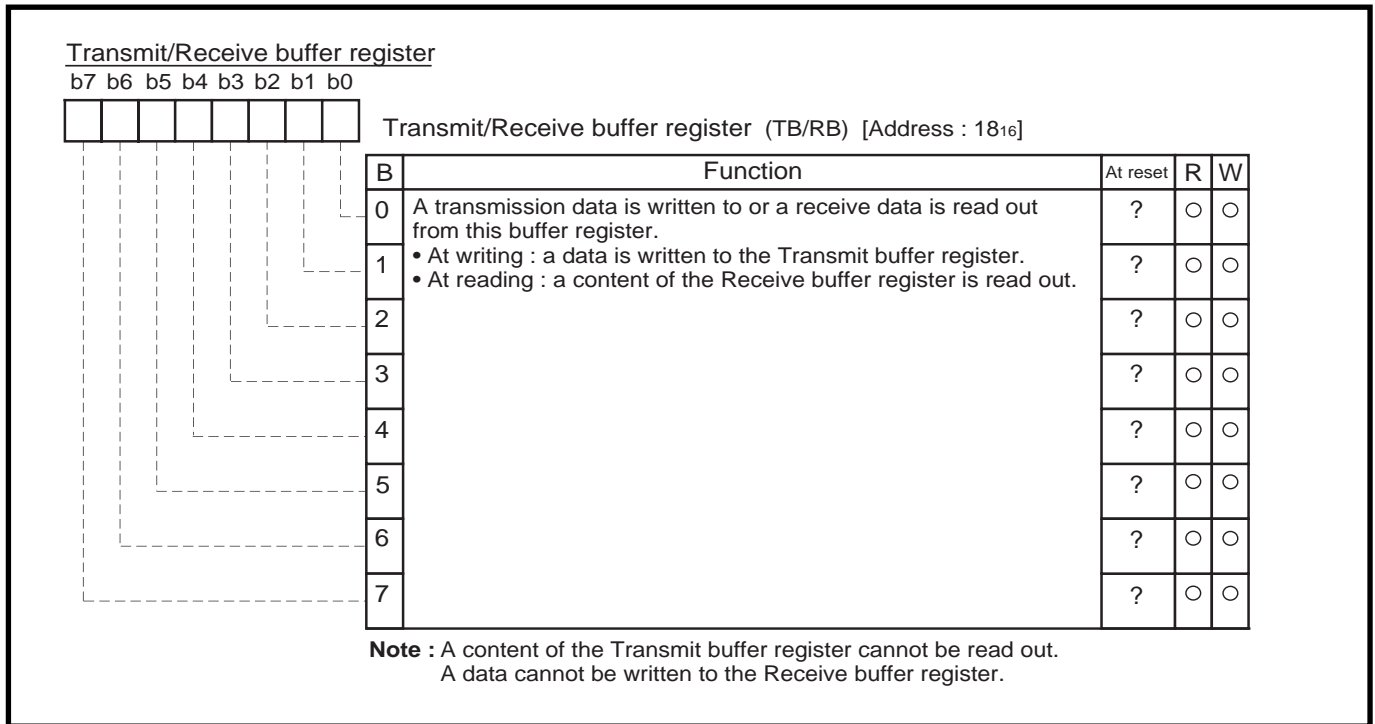


Fig. 2.3.2 Structure of Transmit/Receive buffer register

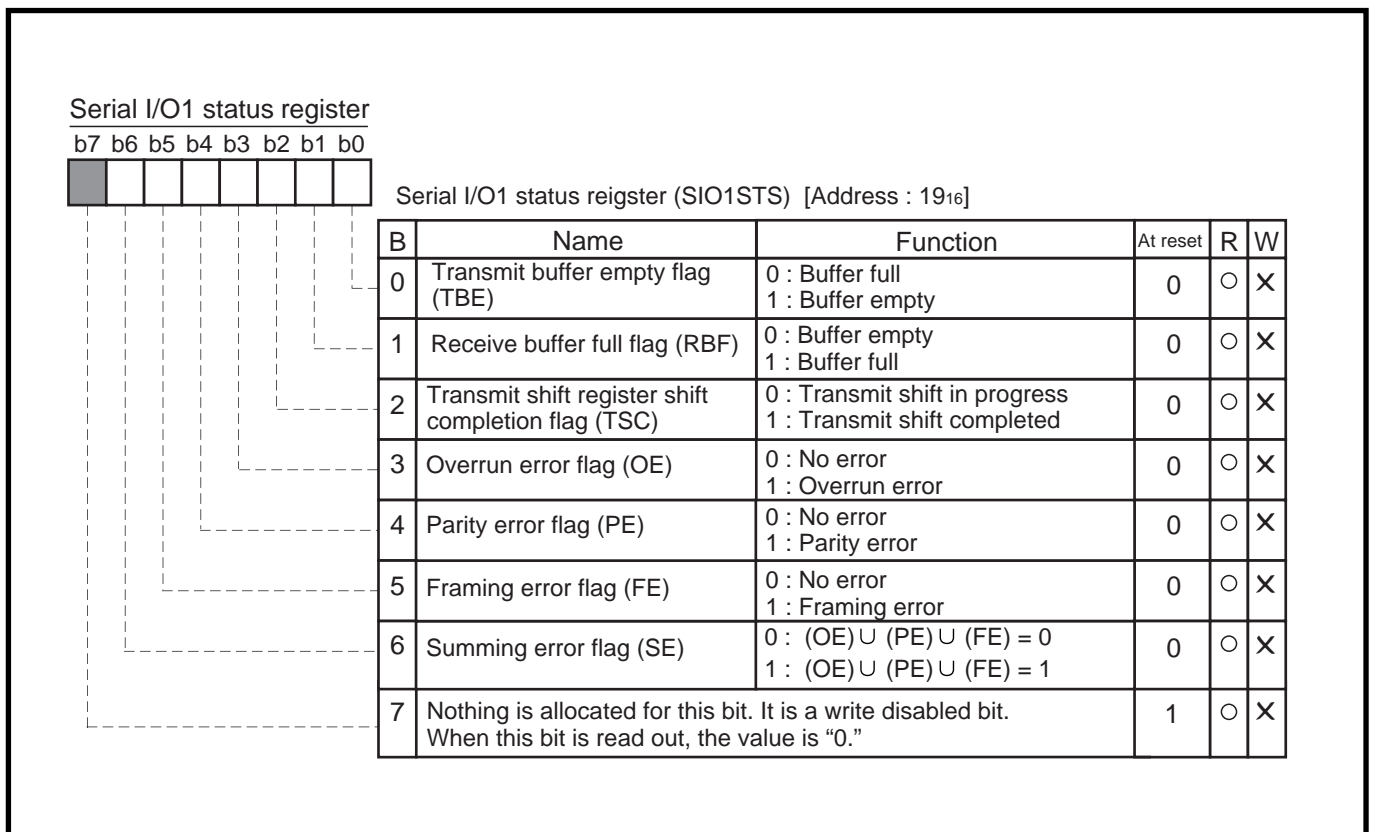


Fig. 2.3.3 Structure of Serial I/O1 status register

## Serial I/O1 control register

b7 b6 b5 b4 b3 b2 b1 b0



Serial I/O1 control register (SIO1CON) [Address : 1A16]

B	Name	Function	At reset	R	W
0	BRG count source selection bit (CSS)	0 : f(X <sub>IN</sub> ) 1 : f(X <sub>IN</sub> )/4	0	○	○
1	Serial I/O1 synchronous clock selection bit (SCS)	At selecting clock synchronous serial I/O 0 : BRG output divided by 4 1 : External clock input At selecting UART 0 : BRG output divided by 16 1 : External clock input divided by 16	0	○	○
2	SRDY <sub>1</sub> output enable bit (SRDY)	0 : I/O port (P47) 1 : SRDY <sub>1</sub> output pin	0	○	○
3	Transmit interrupt source selection bit (TIC)	0 : Transmit buffer empty 1 : Transmit shift operating completion	0	○	○
4	Transmit enable bit (TE)	0 : Transmit disabled 1 : Transmit enabled	0	○	○
5	Receive enable bit (RE)	0 : Receive disabled 1 : Receive enabled	0	○	○
6	Serial I/O1 mode selection bit (SIOM)	0 : UART 1 : Clock synchronous serial I/O	0	○	○
7	Serial I/O1 enable bit (SIOE)	0 : Serial I/O1 disabled (P44–P47 : I/O port) 1 : Serial I/O1 enabled (P44–P47 : Serial I/O function pin)	0	○	○

**Note 1** : In low-speed mode f(X<sub>CIN</sub>) is selected.  
**Note 2** : In low-speed mode f(X<sub>CIN</sub>)/4 is selected.

Fig. 2.3.4 Structure of Serial I/O1 control register

## UART control register

b7 b6 b5 b4 b3 b2 b1 b0



UART control register (UARTCON) [Address : 1B16]

B	Name	Function	At reset	R	W
0	Character length selection bit (CHAS)	0 : 8 bits 1 : 7 bits	0	○	○
1	Parity enable bit (PARE)	0 : Parity checking disabled 1 : Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0 : Even parity 1 : Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0 : 1 stop bit 1 : 2 stop bits	0	○	○
4	P45/TxD P-channel output disable bit (POFF)	In output mode 0 : CMOS output 1 : N-channel open-drain output	0	○	○
5	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "1."		1	○	X
6			1	○	X
7			1	○	X

Fig. 2.3.5 Structure of UART control register

# APPLICATION

## 2.3 Serial I/O

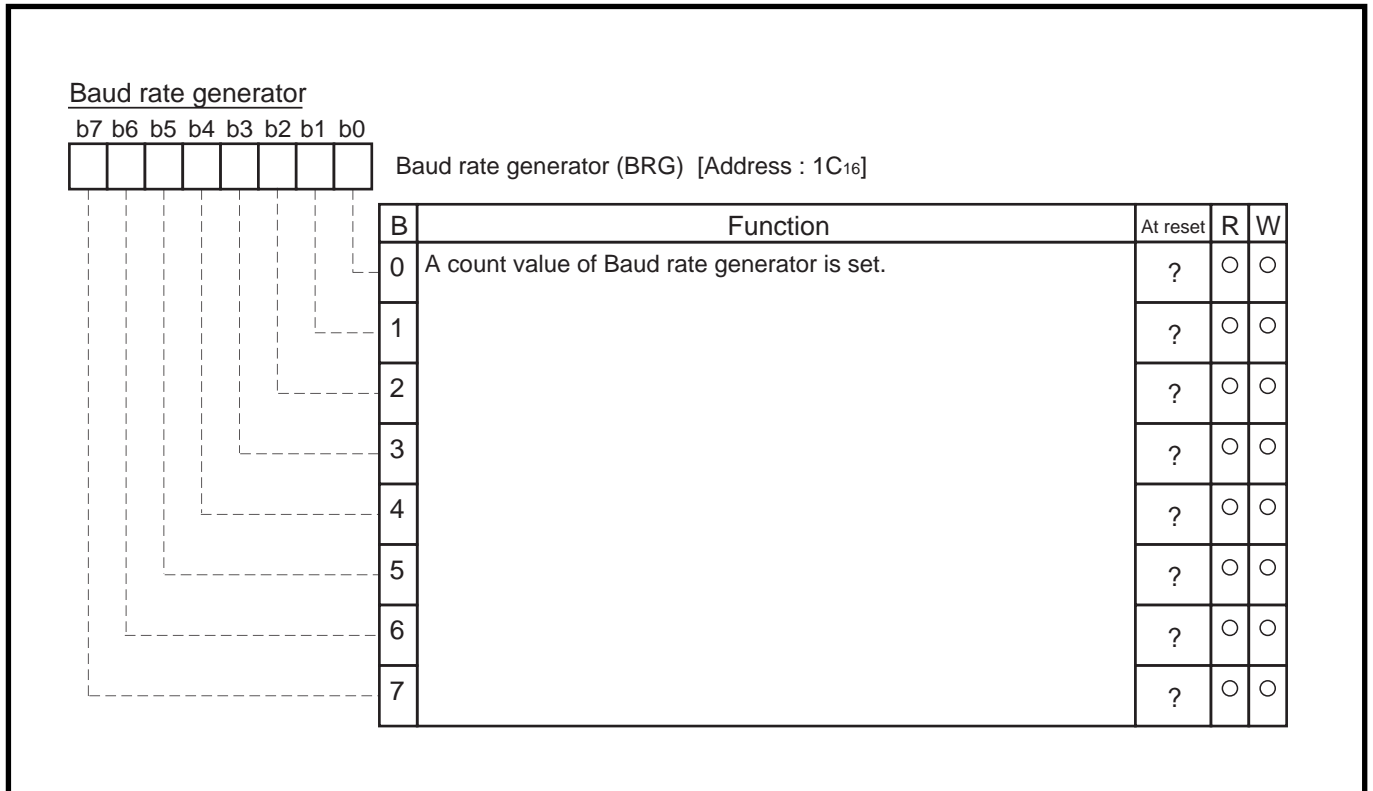


Fig. 2.3.6 Structure of Baud rate generator

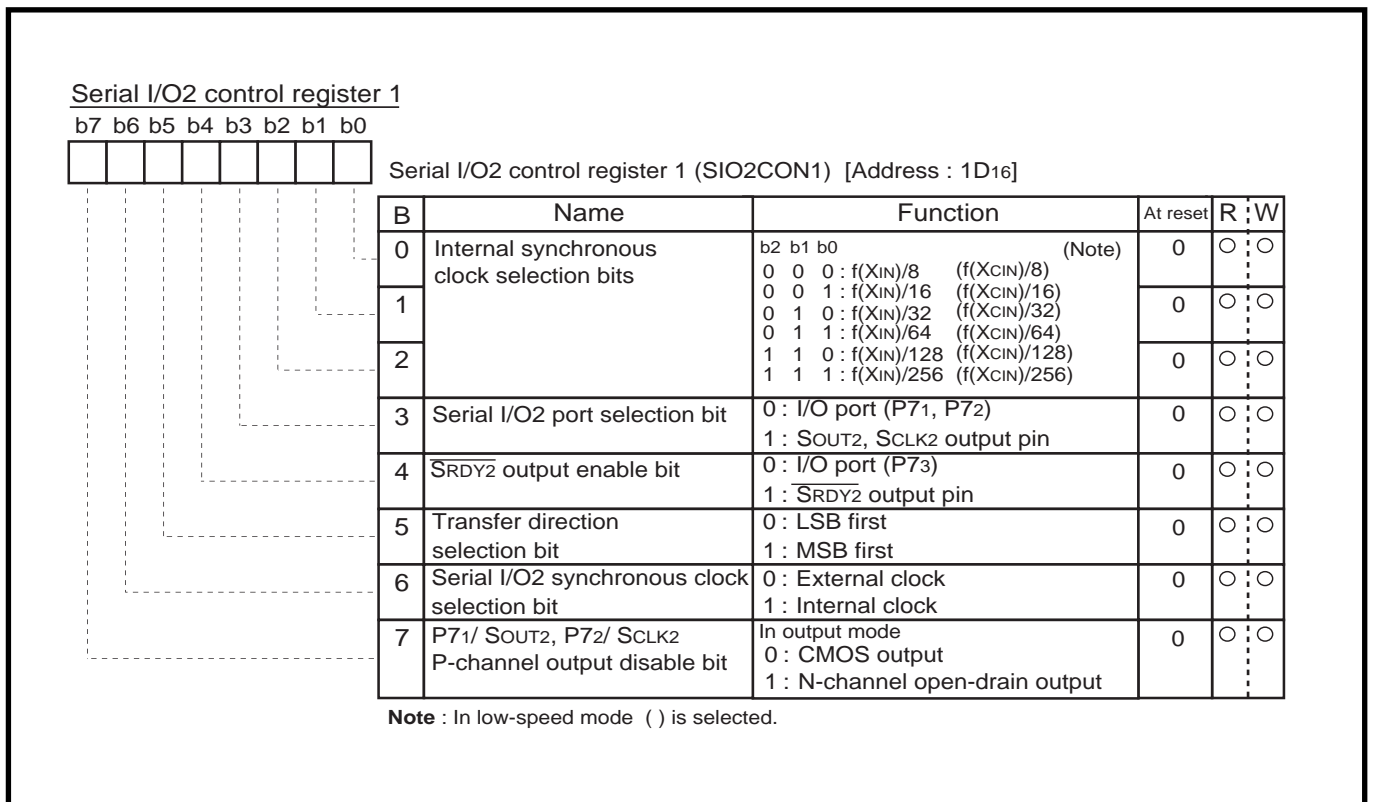


Fig. 2.3.7 Structure of Serial I/O2 control register 1

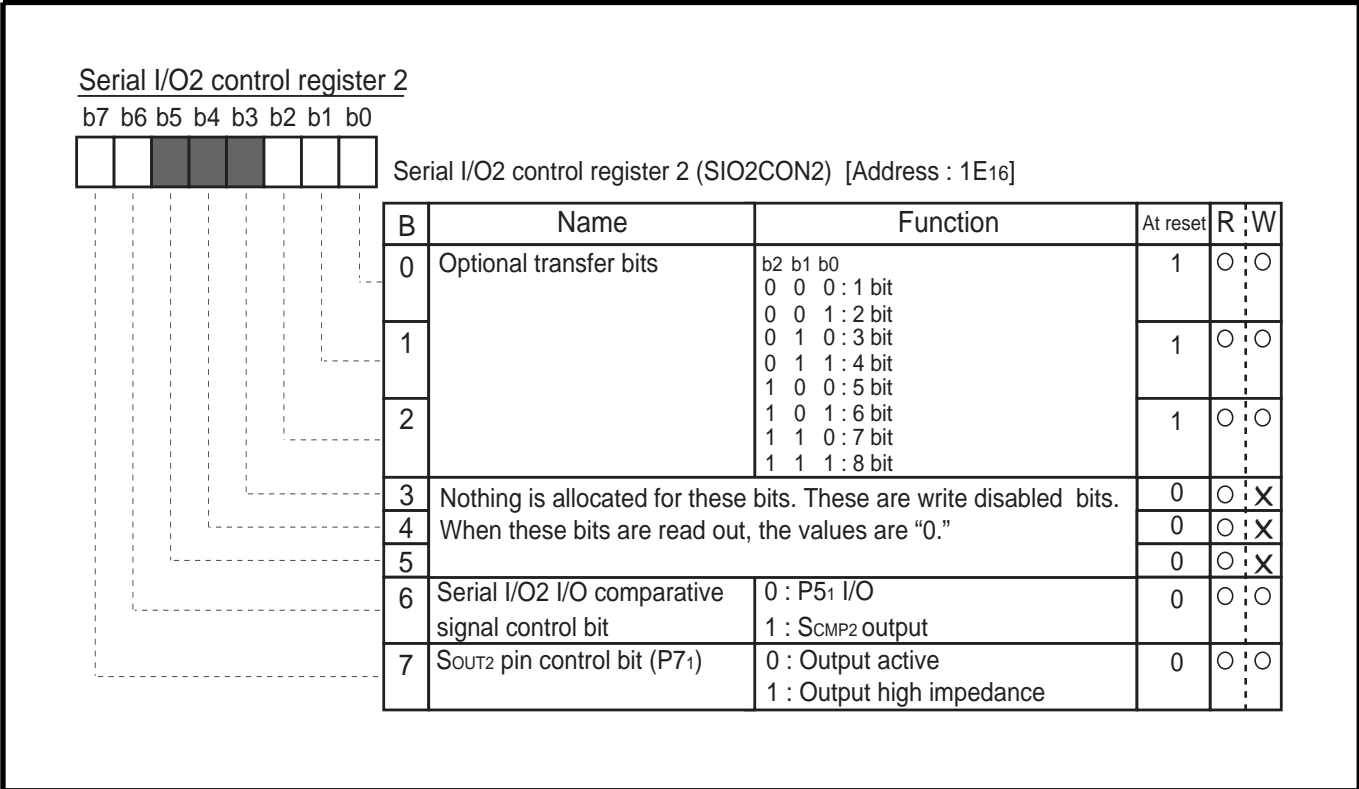


Fig. 2.3.8 Structure of Serial I/O2 control register 2

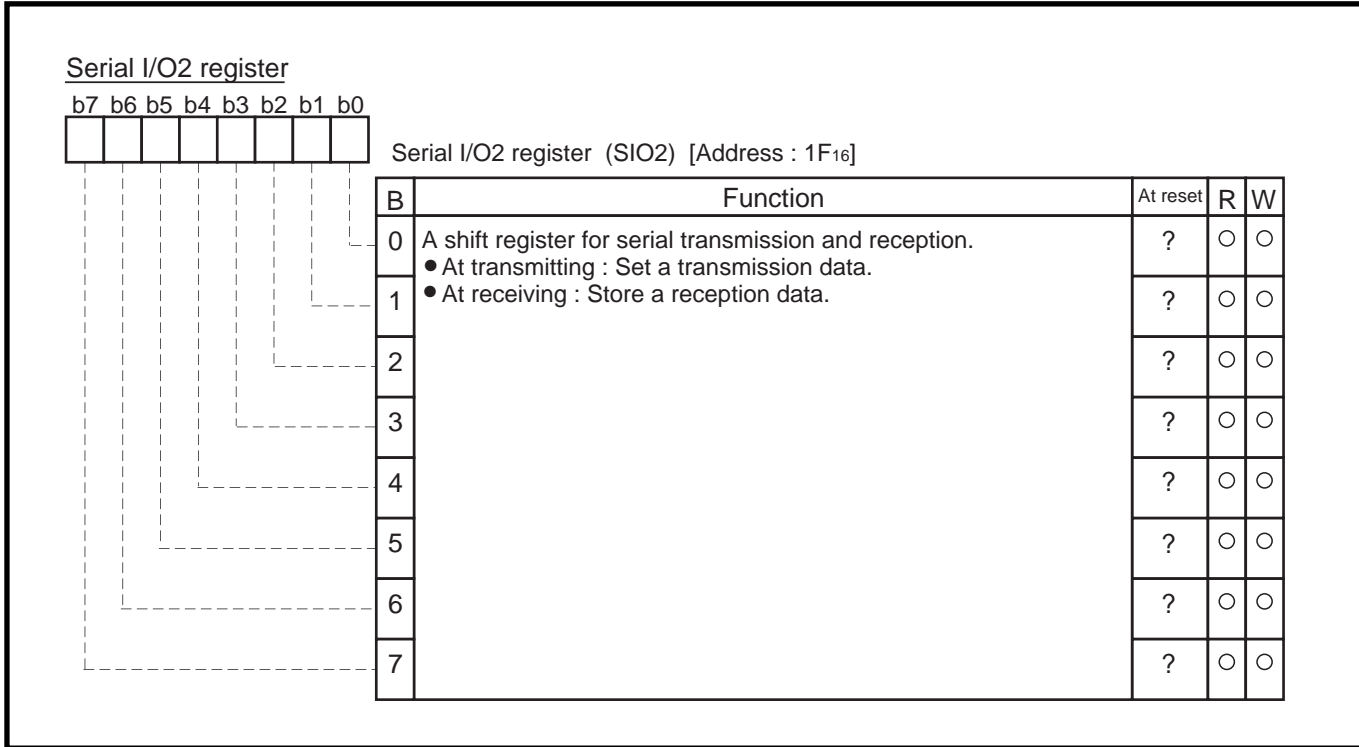


Fig. 2.3.9 Structure of Serial I/O2 register

# APPLICATION

## 2.3 Serial I/O

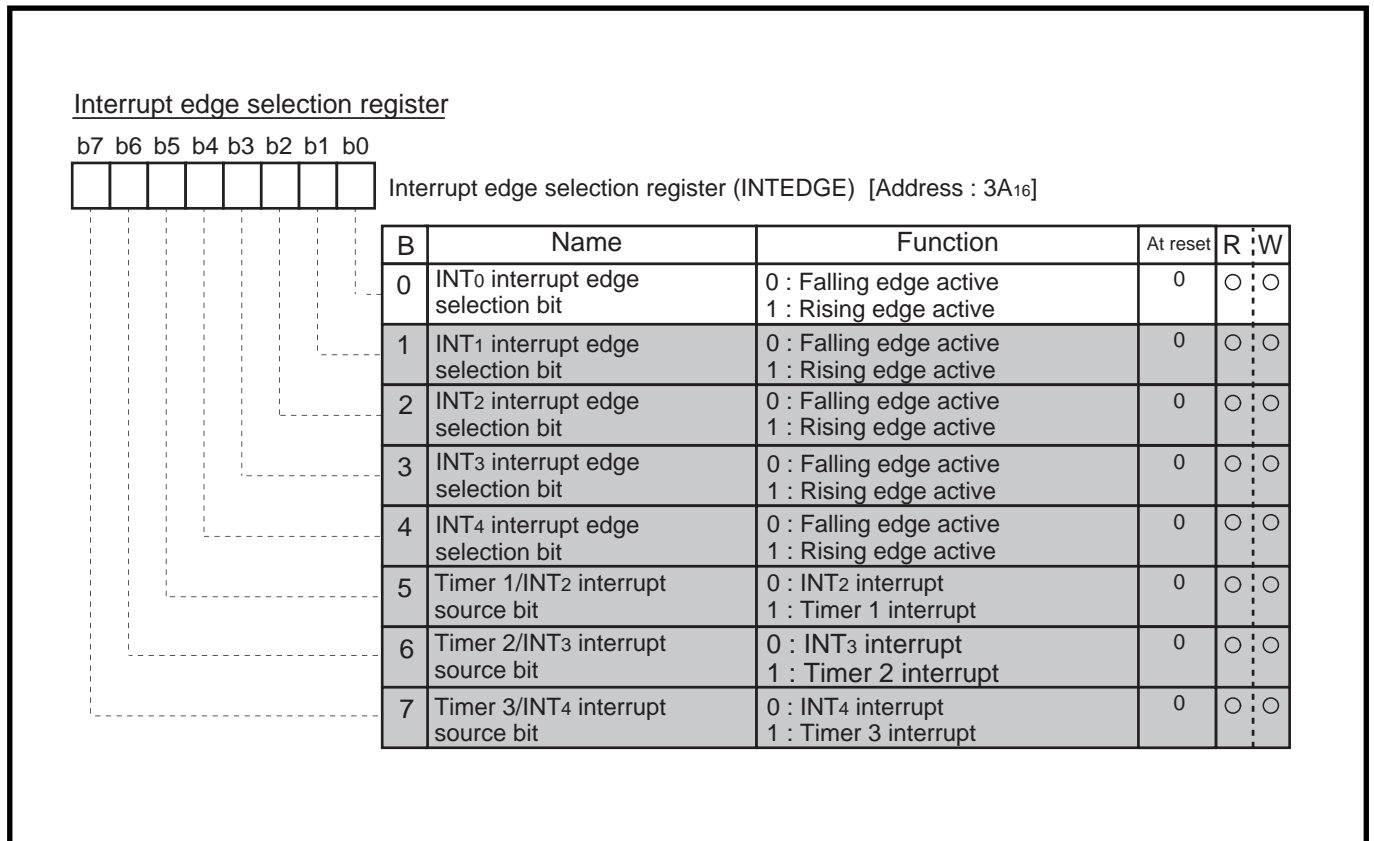


Fig. 2.3.10 Structure of Interrupt edge selection register

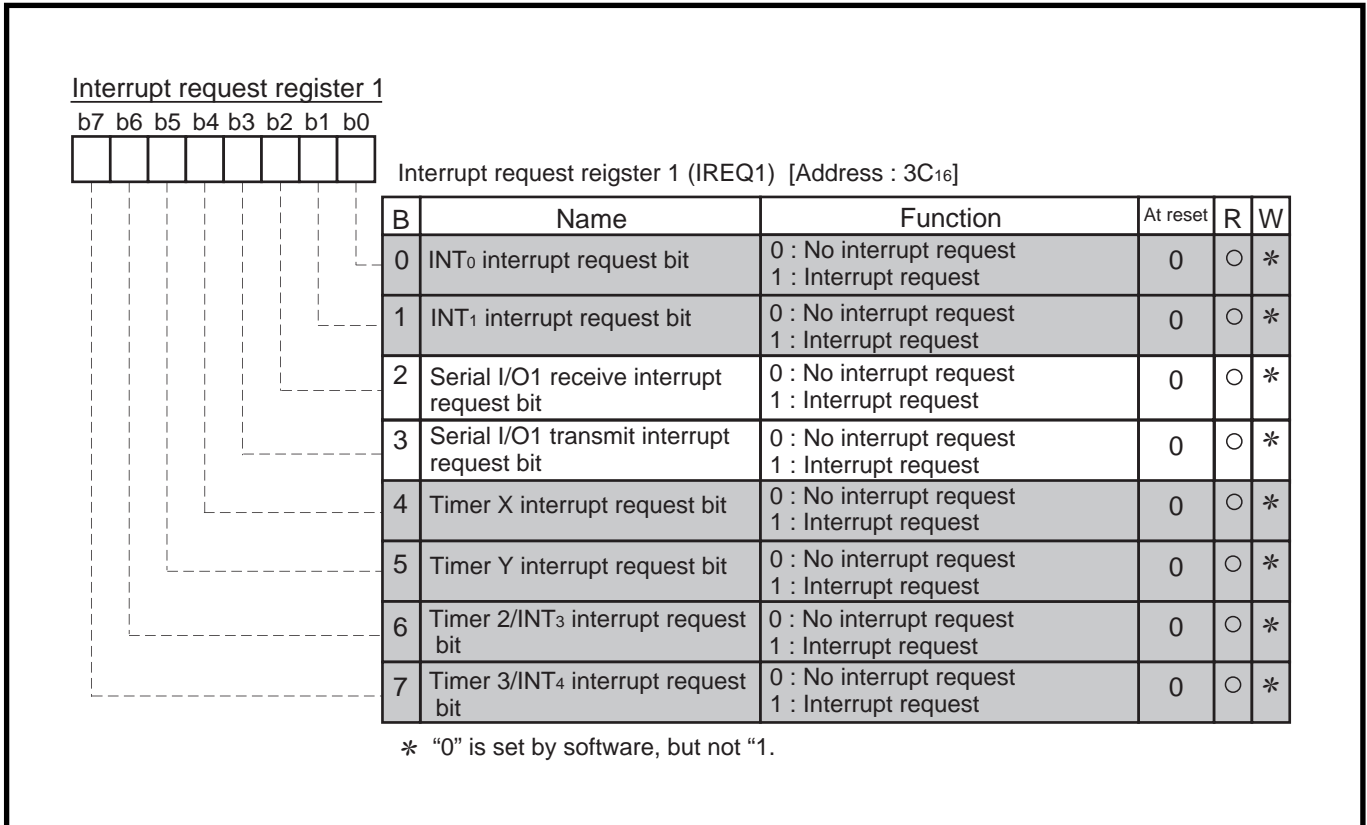


Fig. 2.3.11 Structure of Interrupt request register 1

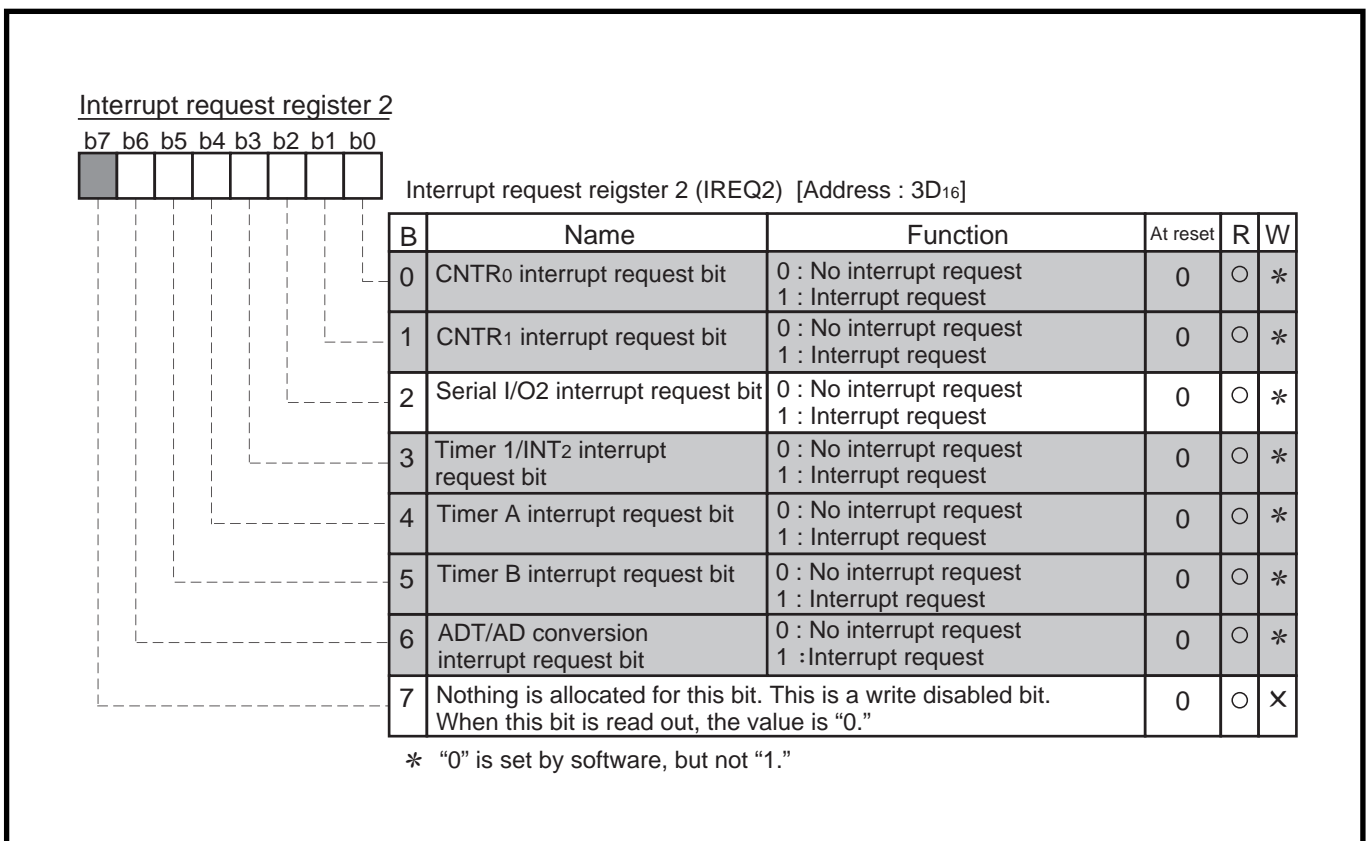


Fig. 2.3.12 Structure of Interrupt request register 2

# APPLICATION

## 2.3 Serial I/O

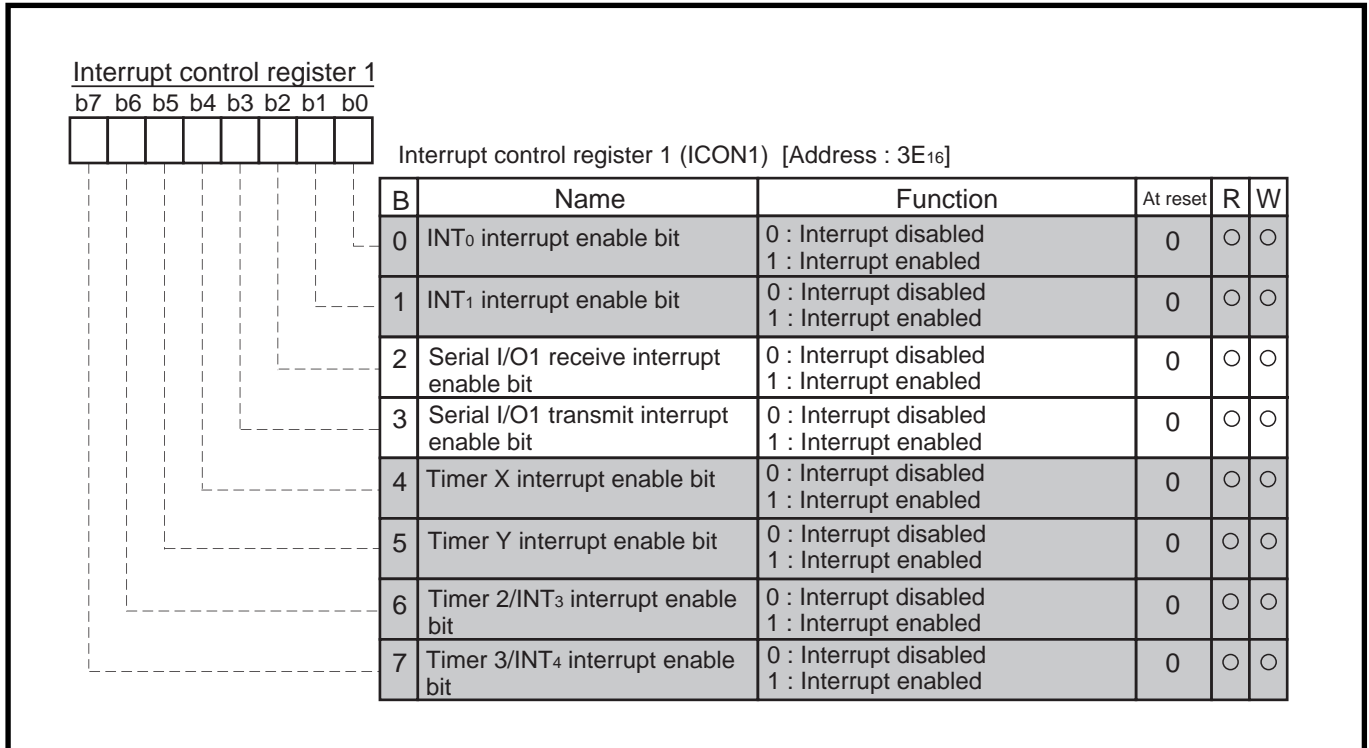


Fig. 2.3.13 Structure of Interrupt control register 1

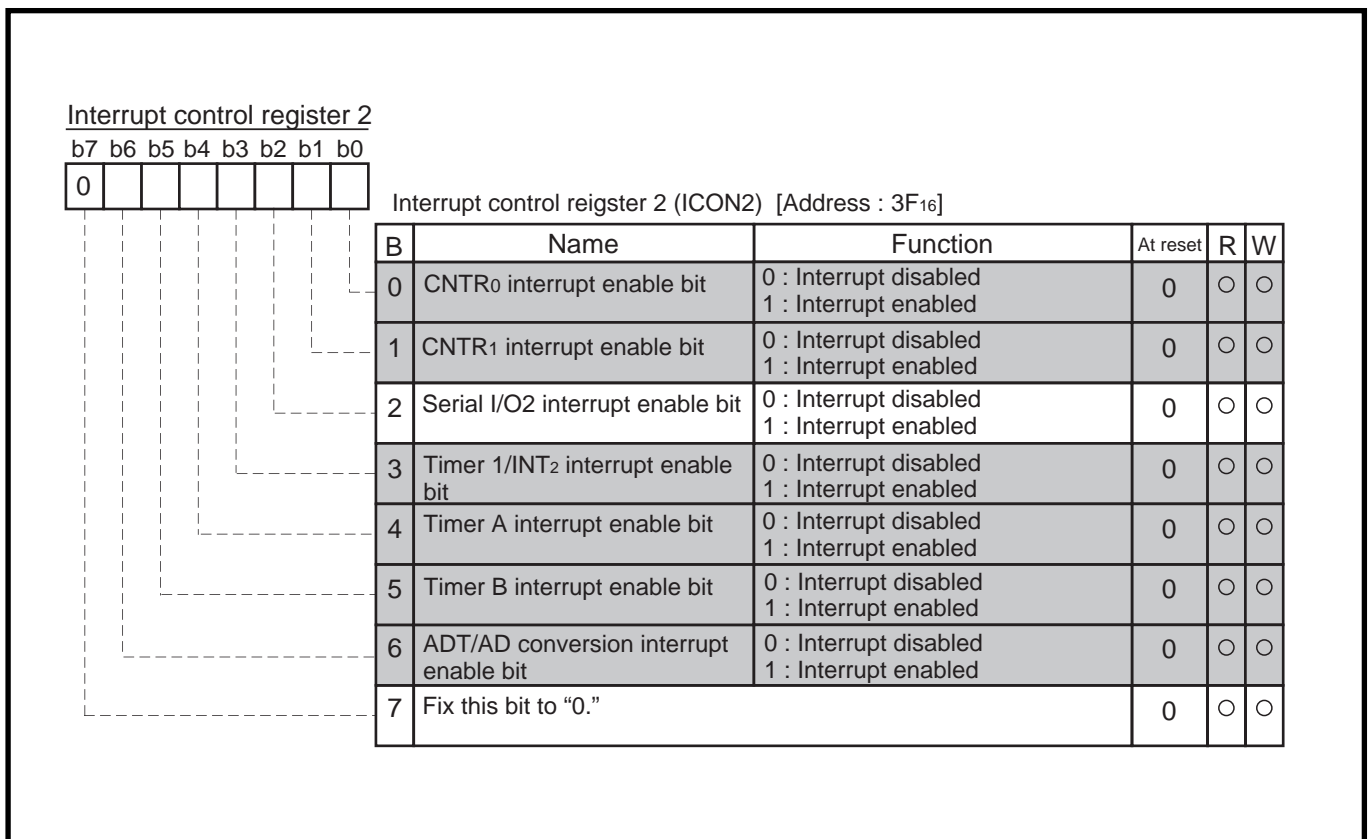


Fig. 2.3.14 Structure of Interrupt control register 2

### 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 2.3.15 shows connection examples of a peripheral IC equipped with the CS pin.

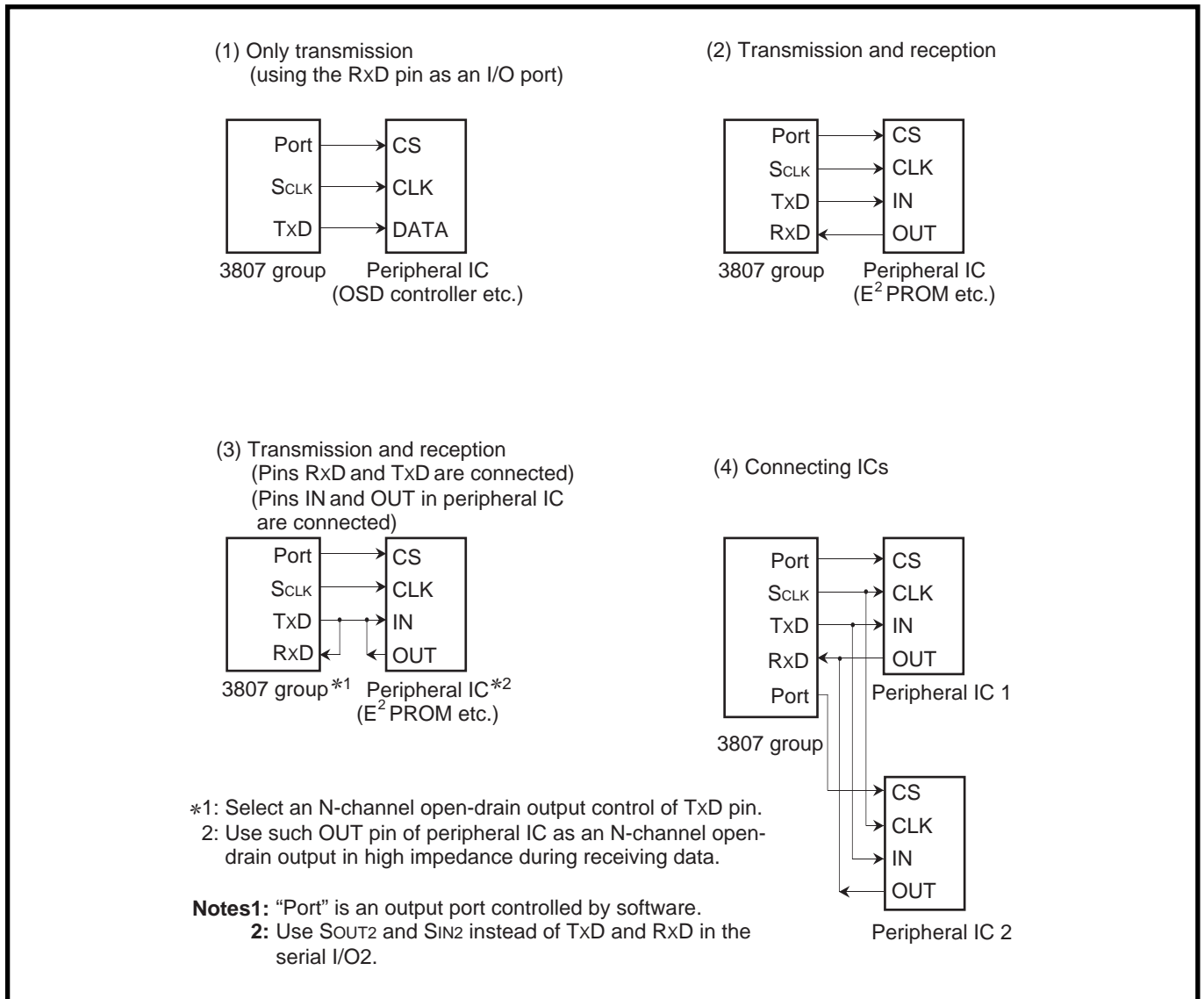


Fig. 2.3.15 Serial I/O connection examples (1)



# APPLICATION

## 2.3 Serial I/O

### (2) Connection with microcomputer

Figure 2.3.16 shows connection examples of the other microcomputers.

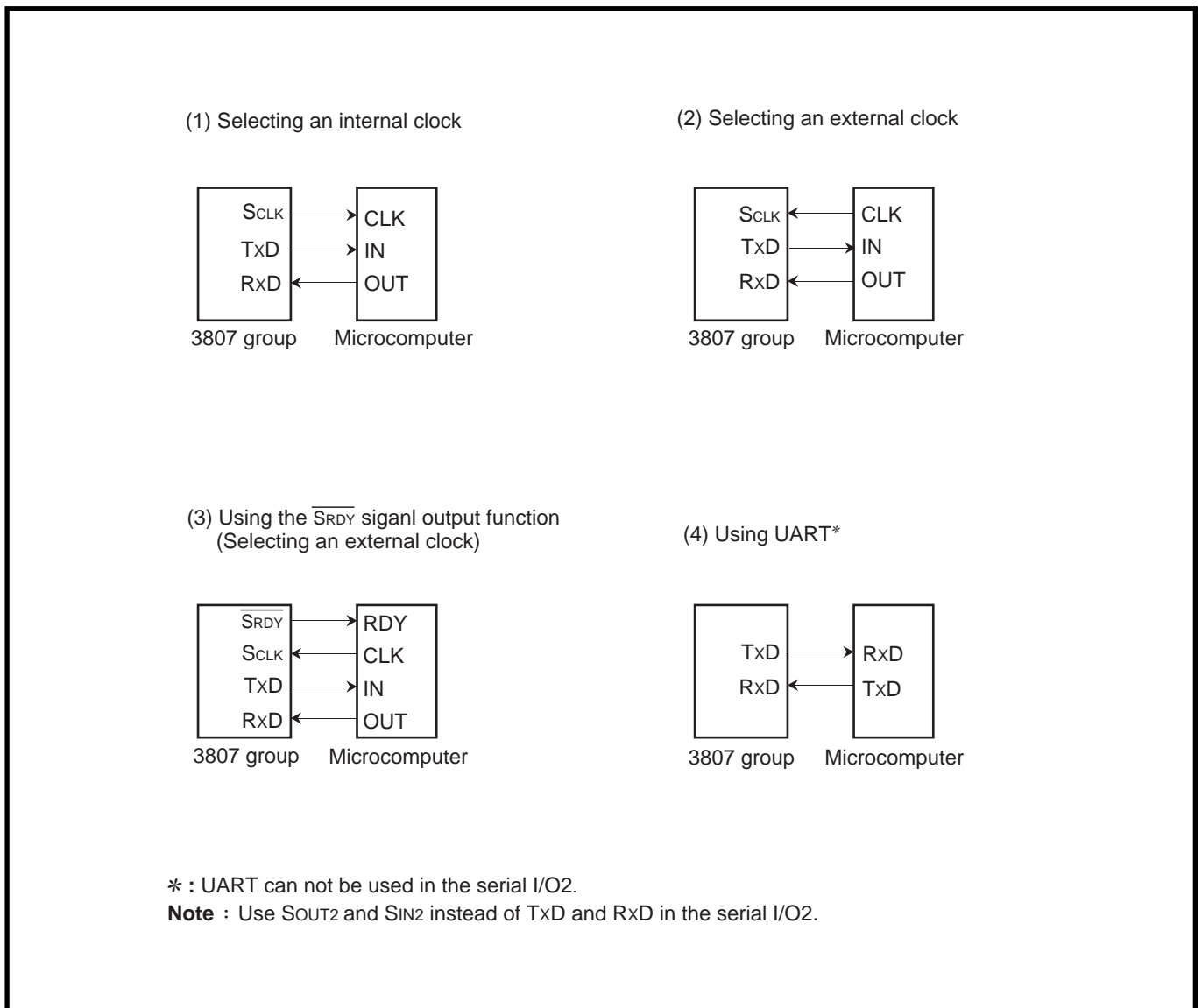


Fig. 2.3.16 Serial I/O connection examples (2)

### 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/O1. The serial I/O2 operates in a clock synchronous.

Figure 2.3.17 shows a setting of serial I/O transfer data format.

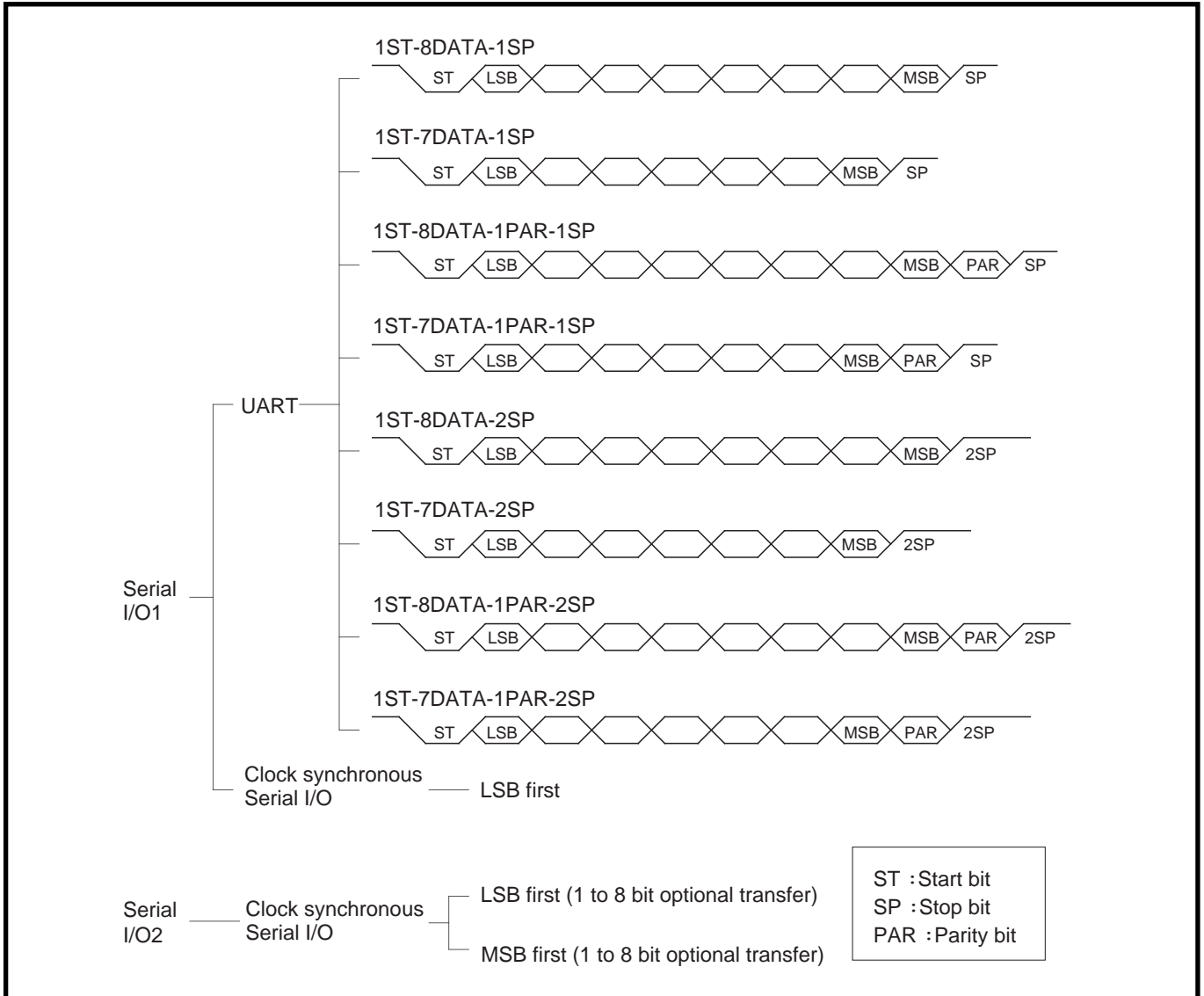


Fig. 2.3.17 Setting of Serial I/O transfer data format

# APPLICATION

## 2.3 Serial I/O

### 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  $\overline{\text{SRDY1}}$  signal is used for communication control.

Figure 2.3.18 shows a connection diagram, and Figure 2.3.19 shows a timing chart.

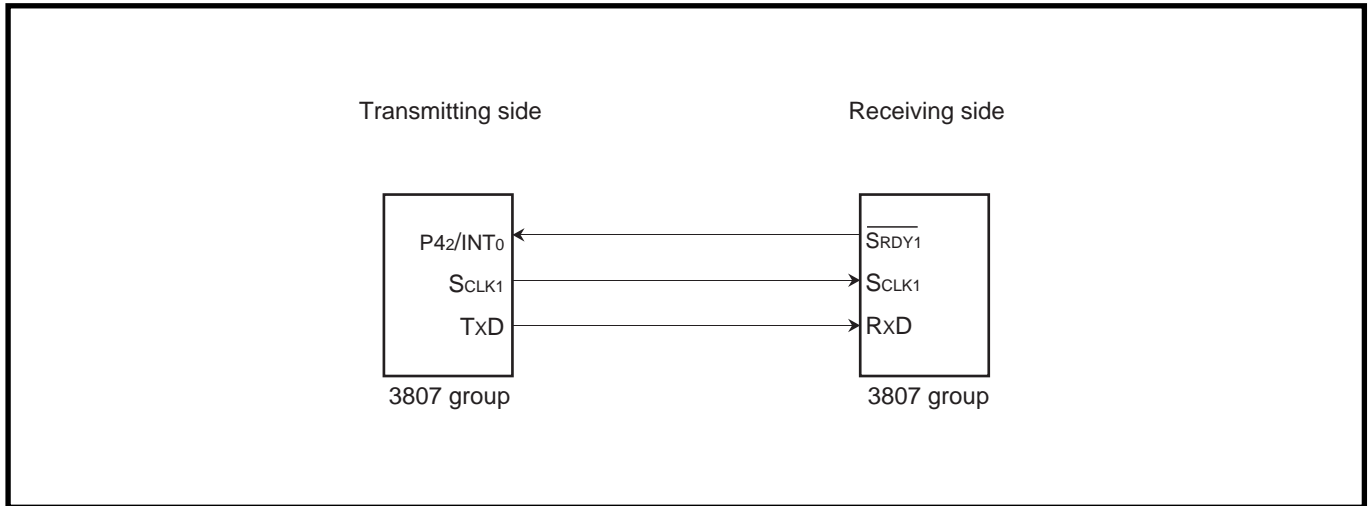


Fig. 2.3.18 Connection diagram [Communication using a clock synchronous serial I/O]

- Specifications :**
- The Serial I/O1 is used (clock synchronous serial I/O is selected)
  - Synchronous clock frequency : 125 kHz ( $f(\text{XIN}) = 8 \text{ MHz}$  is divided by 64)
  - The  $\overline{\text{SRDY1}}$  (receivable signal) is used.
  - The receiving side outputs the  $\overline{\text{SRDY1}}$  signal at intervals of 2 ms (generated by timer), and 2-byte data is transferred from the transmitting side to the receiving side.

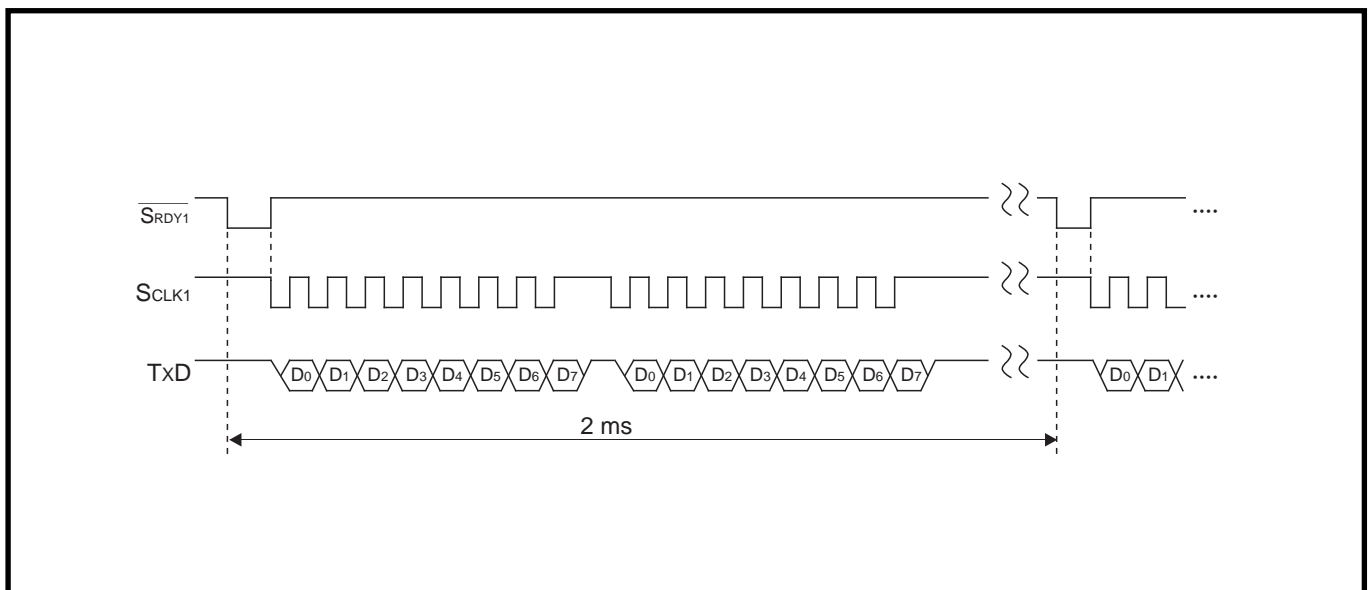


Fig. 2.3.19 Timing chart [Communication using a clock synchronous serial I/O]

## Transmitting side

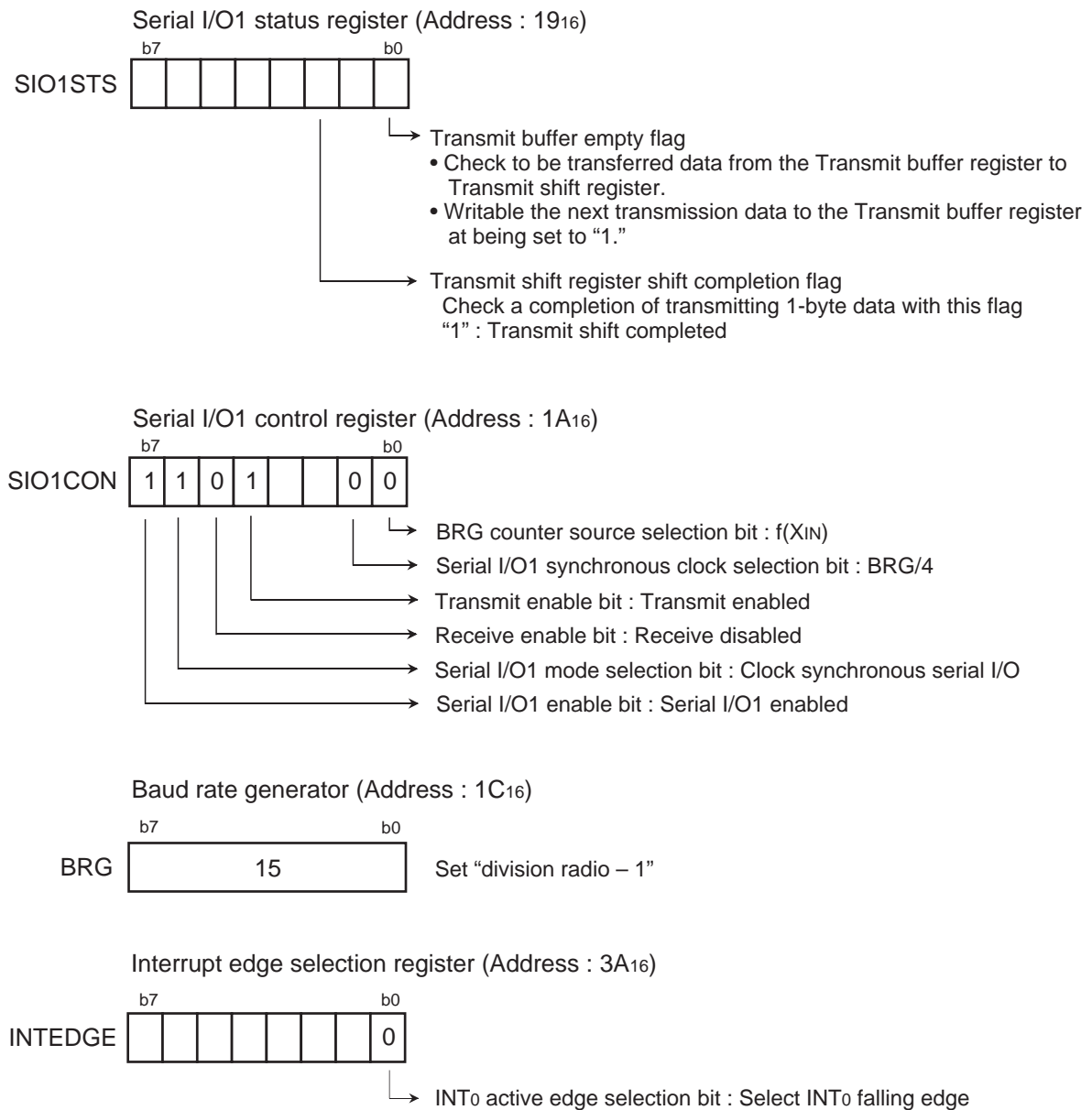


Fig. 2.3.20 Setting of related registers at a transmitting side [Communication using a clock synchronous serial I/O]

# APPLICATION

## 2.3 Serial I/O

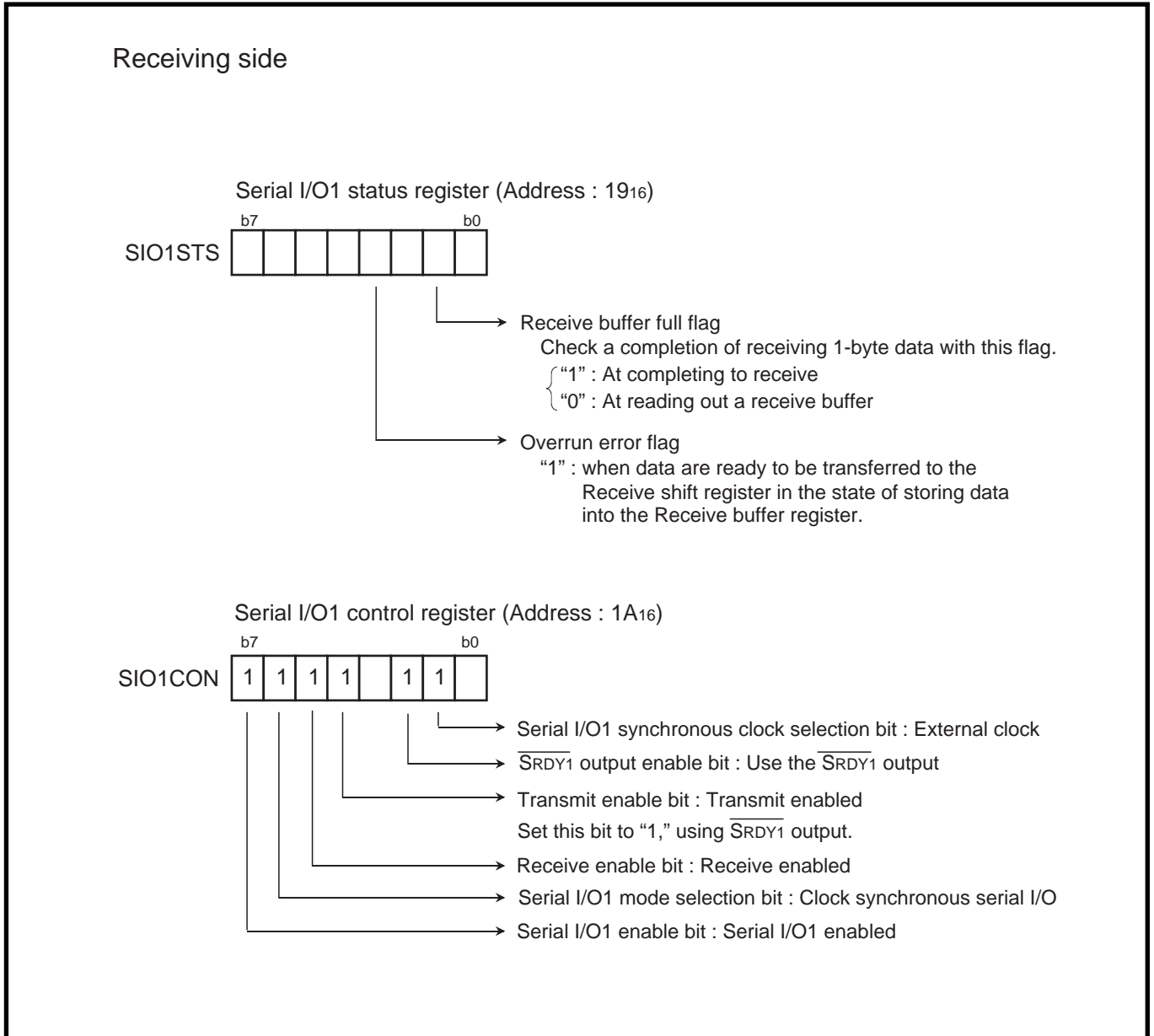
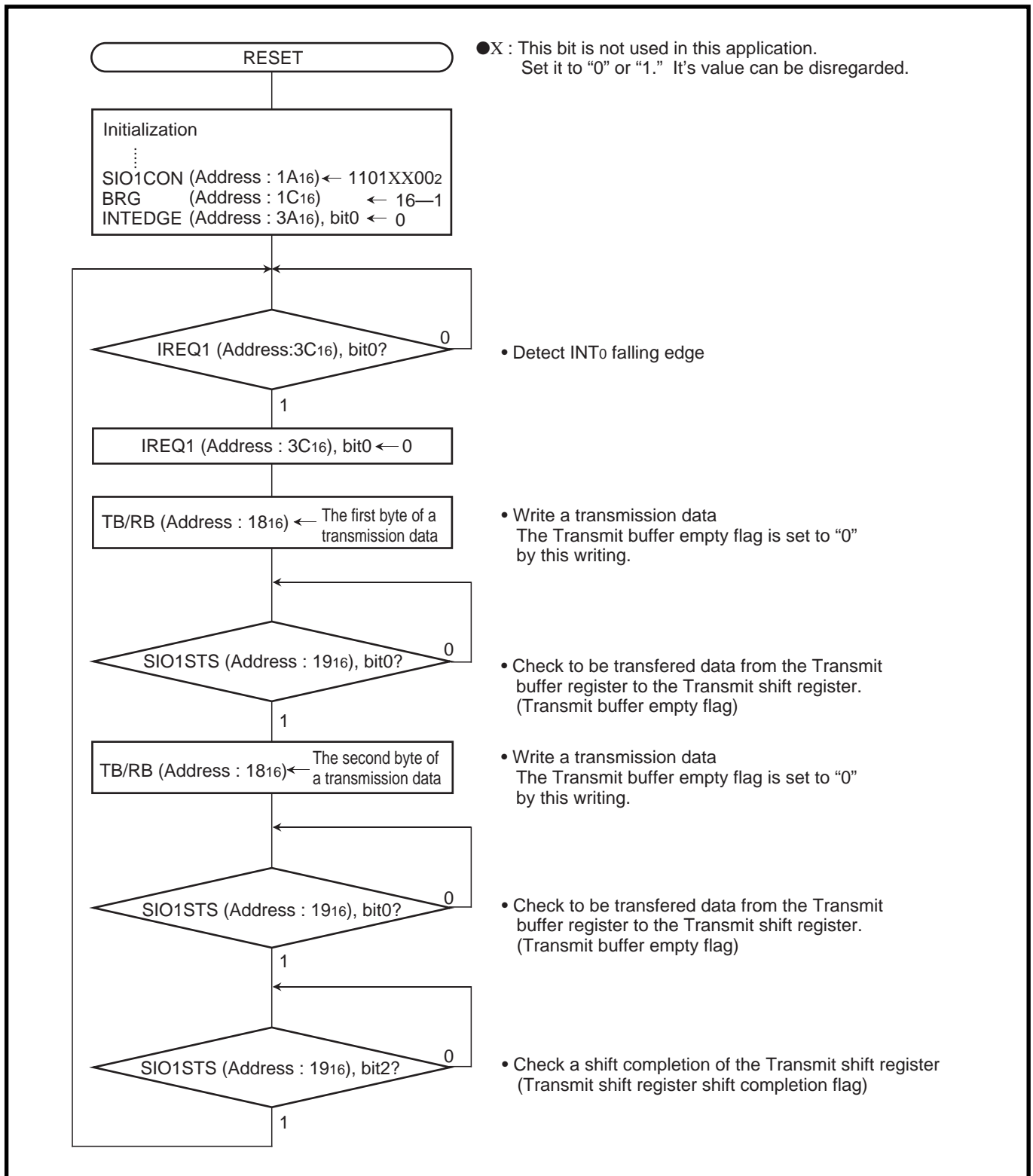


Fig. 2.3.21 Setting of related registers at a receiving side [Communication using a clock synchronous serial I/O]

**Control procedure :** Figure 2.3.22 shows a control procedure at a transmitting side, and Figure 2.3.23 shows a control procedure at a receiving side.



**Fig. 2.3.22 Control procedure at a transmitting side [Communication using a clock synchronous serial I/O]**

# APPLICATION

## 2.3 Serial I/O

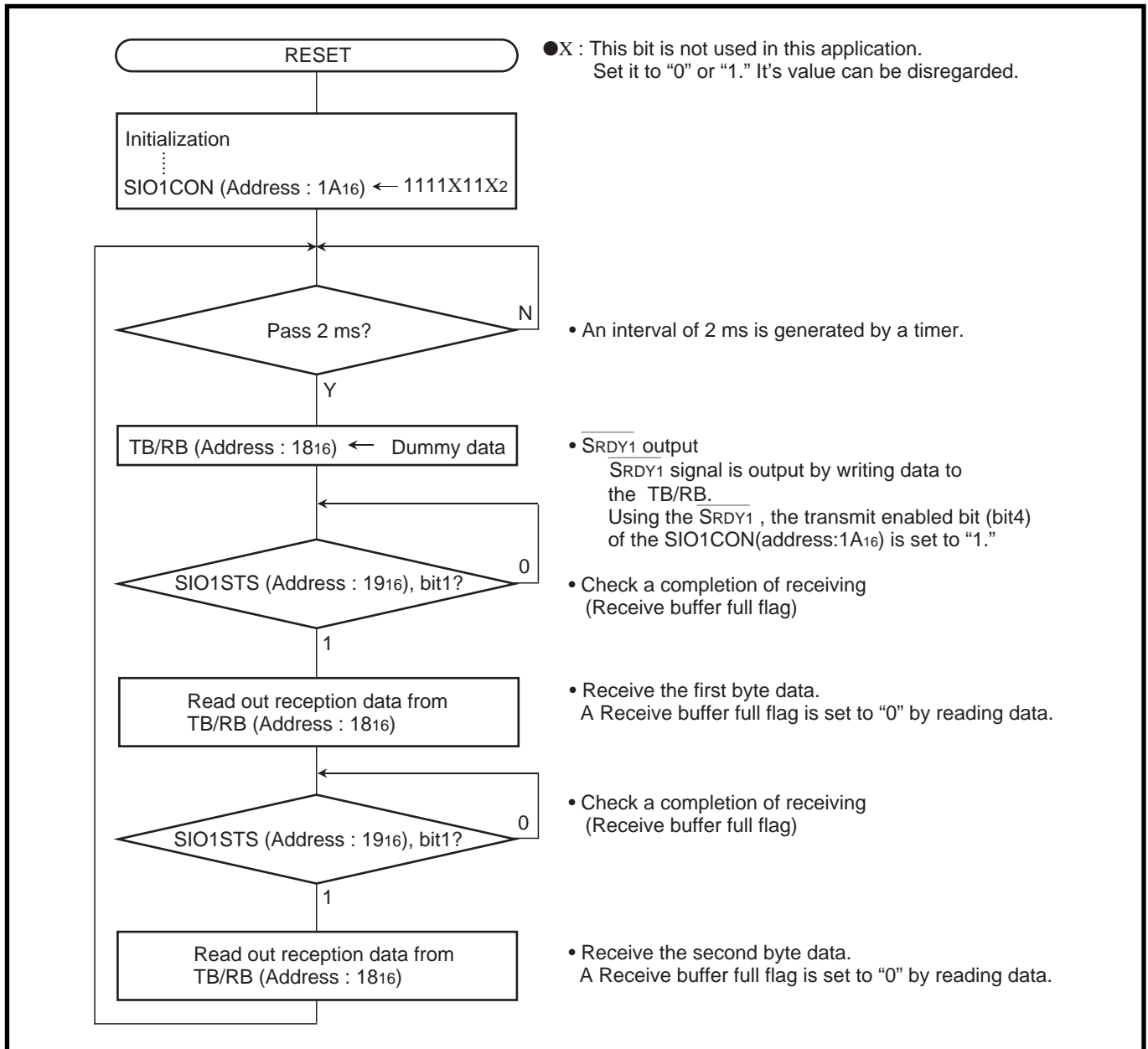


Fig. 2.3.23 Control procedure at a receiving side [Communication using a clock synchronous serial I/O]

(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.

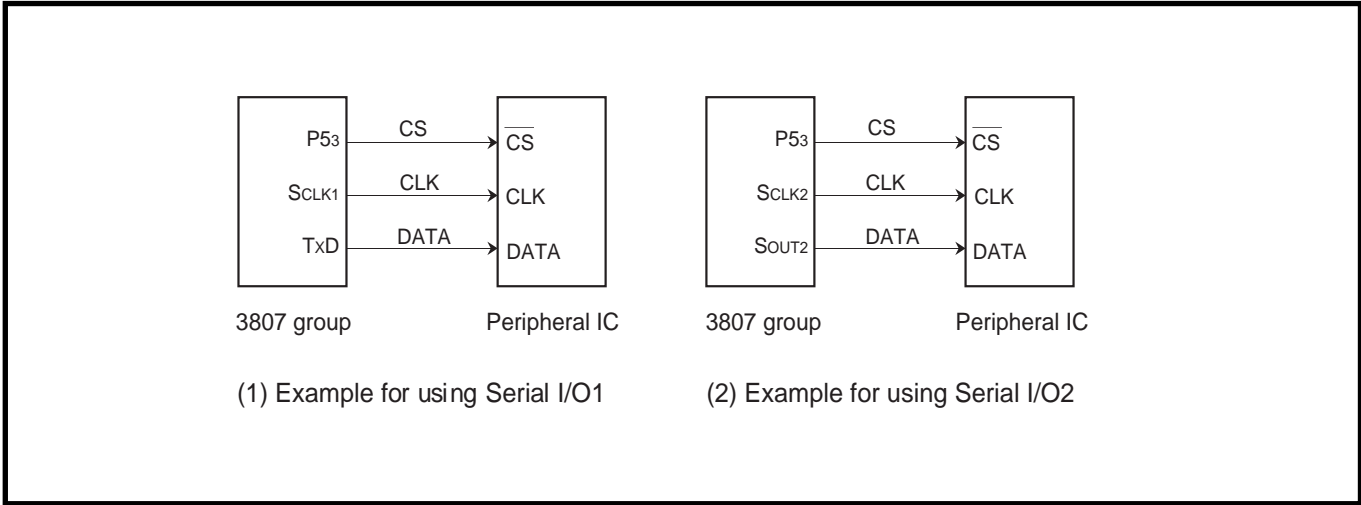


Fig. 2.3.24 Connection diagram [Output of serial data]

- Specifications :**
- The Serial I/O is used. (clock synchronous serial I/O is selected)
  - Synchronous clock frequency : 125 kHz ( $f(XIN) = 8 \text{ MHz}$  is divided by 64)
  - Transfer direction : LSB first
  - The Serial I/O 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 2.3.25 shows an output timing chart of serial data.

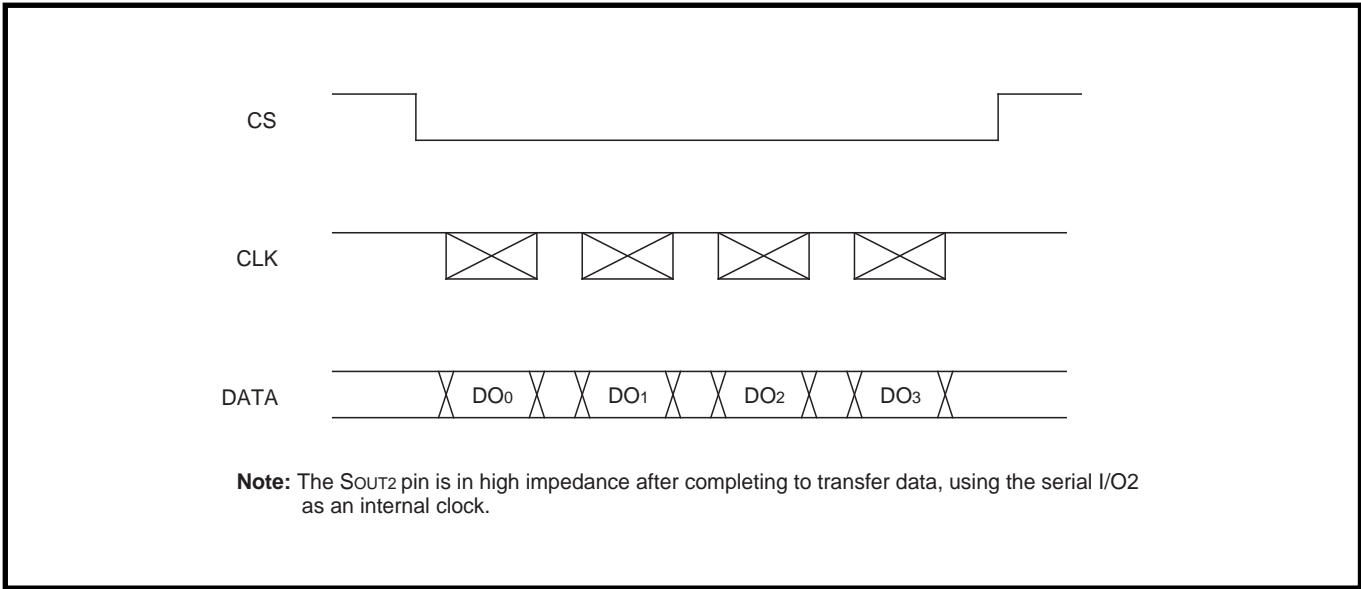


Fig. 2.3.25 Timing chart [Output of serial data]



# APPLICATION

## 2.3 Serial I/O

Figure 2.3.26 shows a setting of serial I/O1 related registers, and Figure 2.3.27 shows a setting of serial I/O1 transmission data.

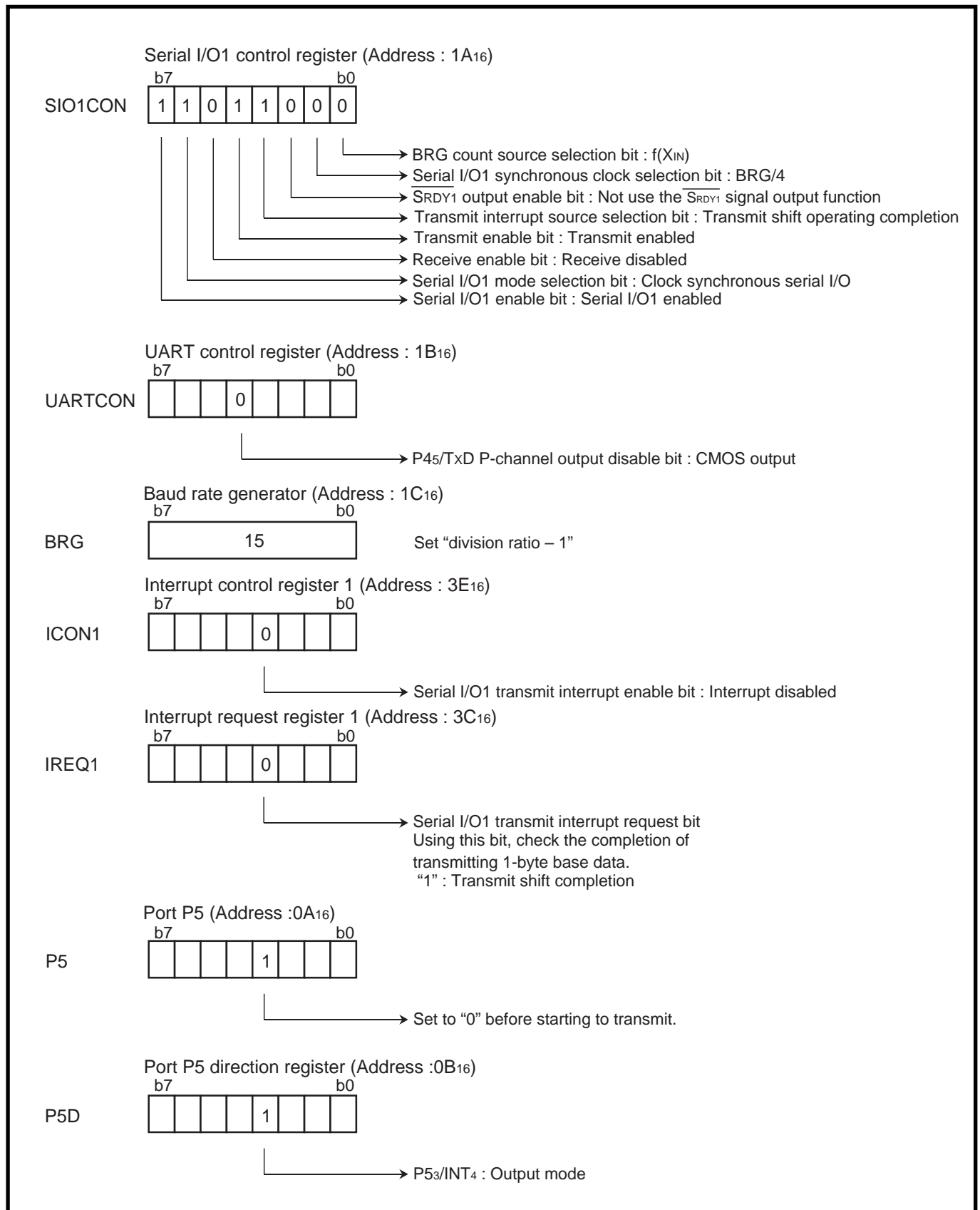


Fig. 2.3.26 Setting of serial I/O1 related registers [Output of serial data]

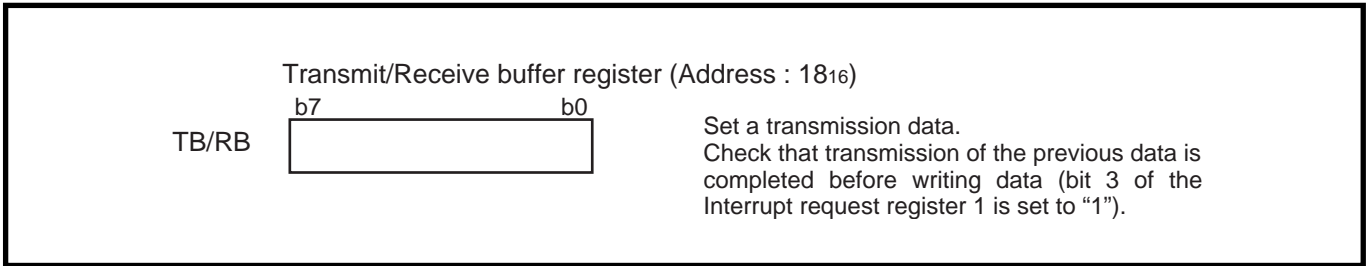


Fig. 2.3.27 Setting of serial I/O1 transmission data [Output of serial data]

# APPLICATION

## 2.3 Serial I/O

**Control procedure :** When the registers are set as shown in Fig. 2.3.26, the Serial I/O1 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 2.3.28 shows a control procedure of serial I/O1.

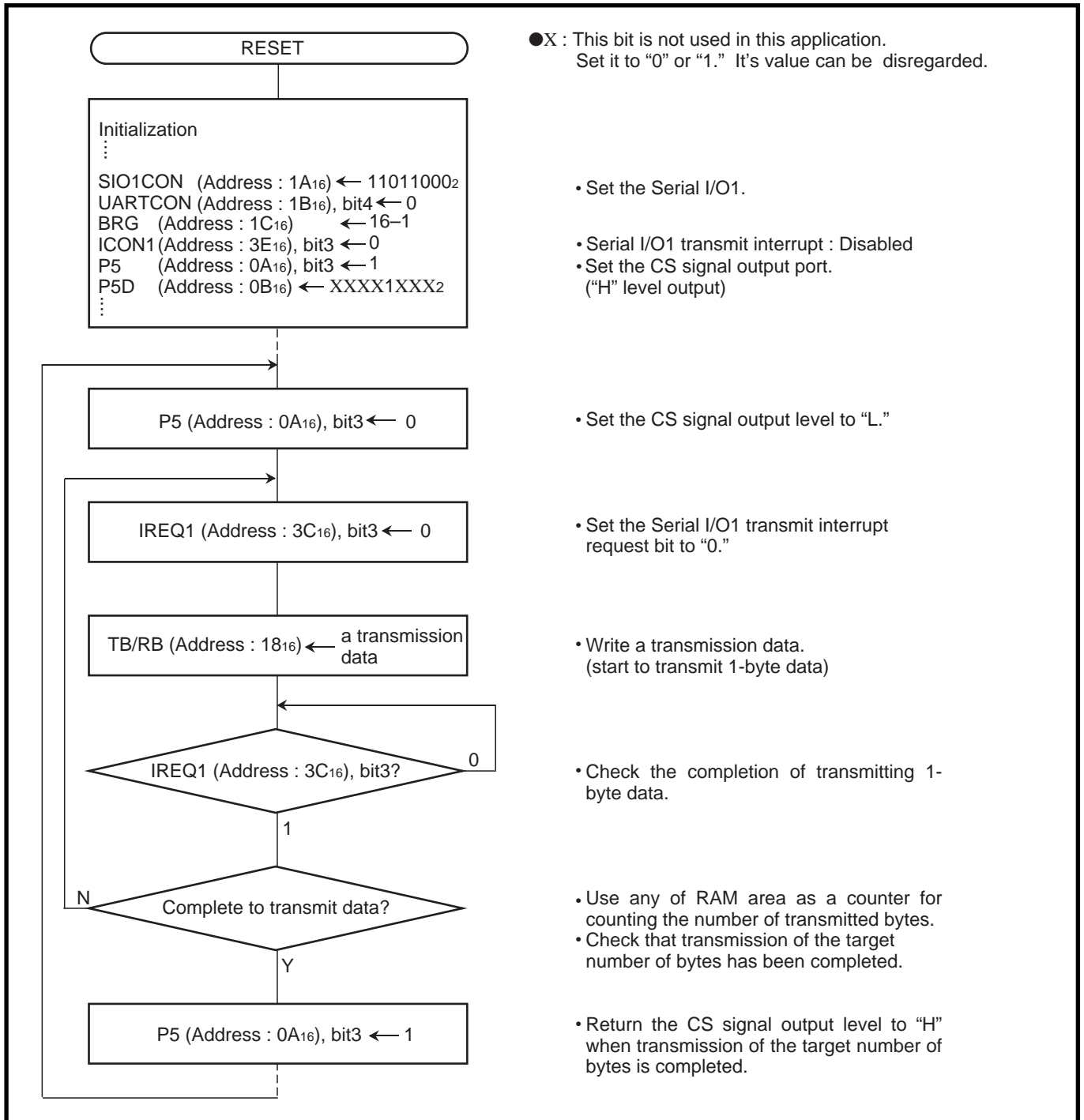


Fig. 2.3.28 Control procedure of serial I/O1 [Output of serial data]

Figure 2.3.29 shows a setting of serial I/O2 related registers, and Figure 2.3.30 shows a setting of serial I/O2 transmission data.

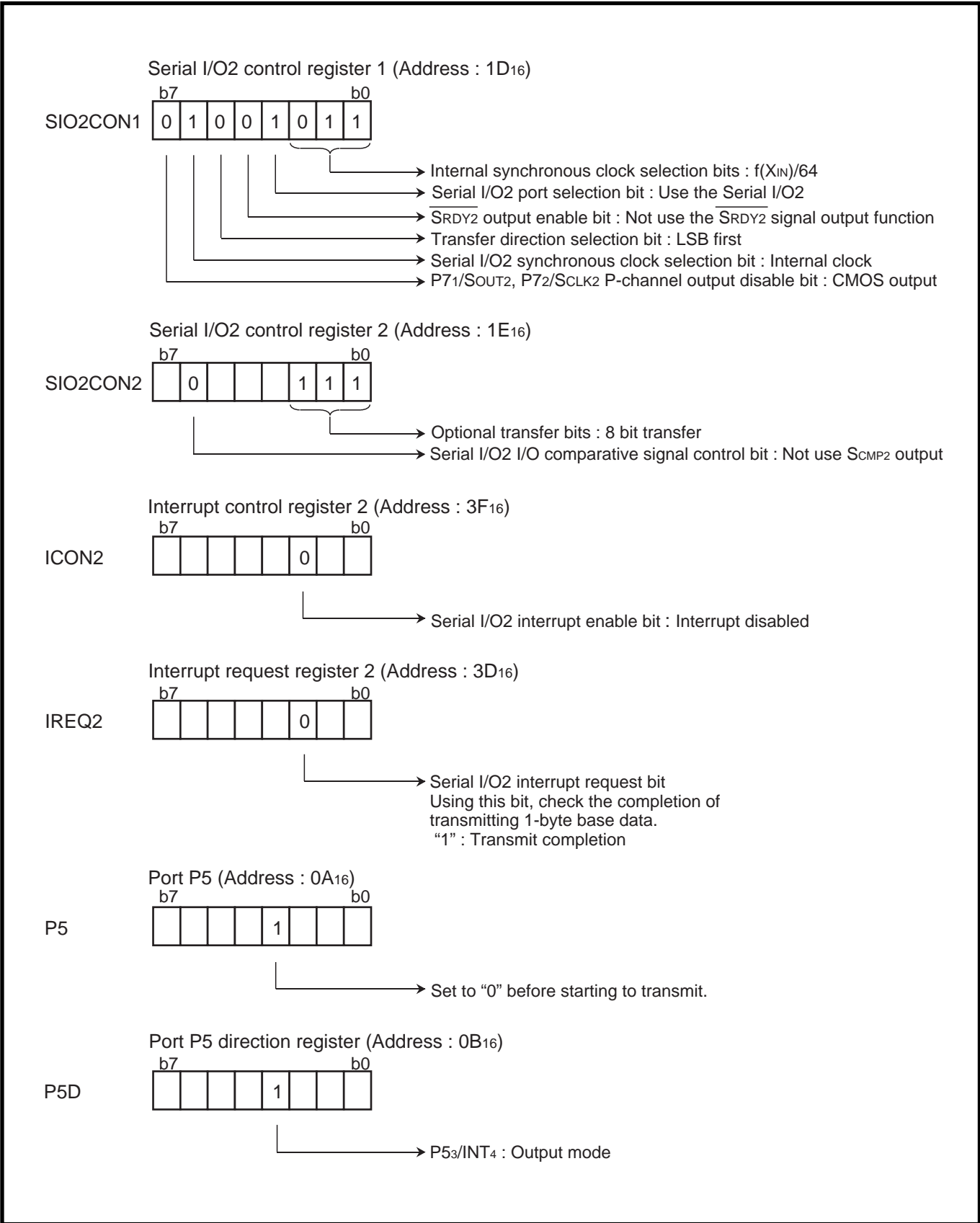


Fig. 2.3.29 Setting of serial I/O2 related registers [Output of serial data]

# APPLICATION

## 2.3 Serial I/O

---

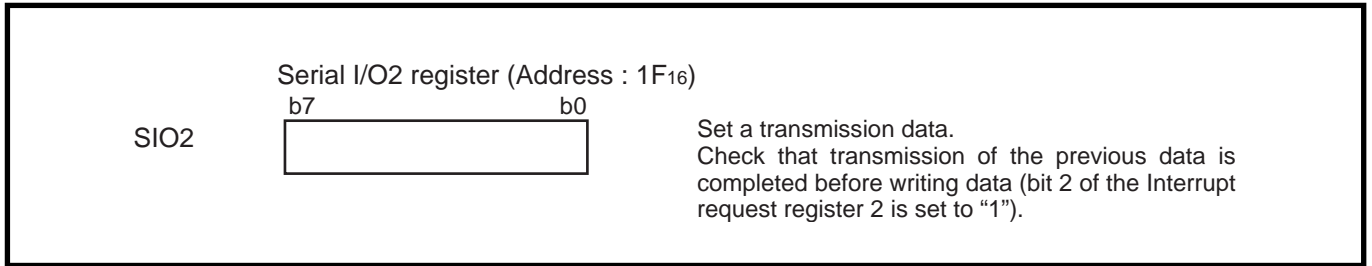


Fig. 2.3.30 Setting of serial I/O2 transmission data [Output of serial data]

**Control procedure :** When the registers are set as shown in Fig. 2.3.29, the Serial I/O2 can transmit 1-byte data simply by writing data to the Serial I/O2 register. Thus, after setting the CS signal to “L,” write the transmission data to the Serial I/O1 register on a 1-byte base, and return the CS signal to “H” when the desired number of bytes have been transmitted.

Figure 2.3.31 shows a control procedure of serial I/O2.

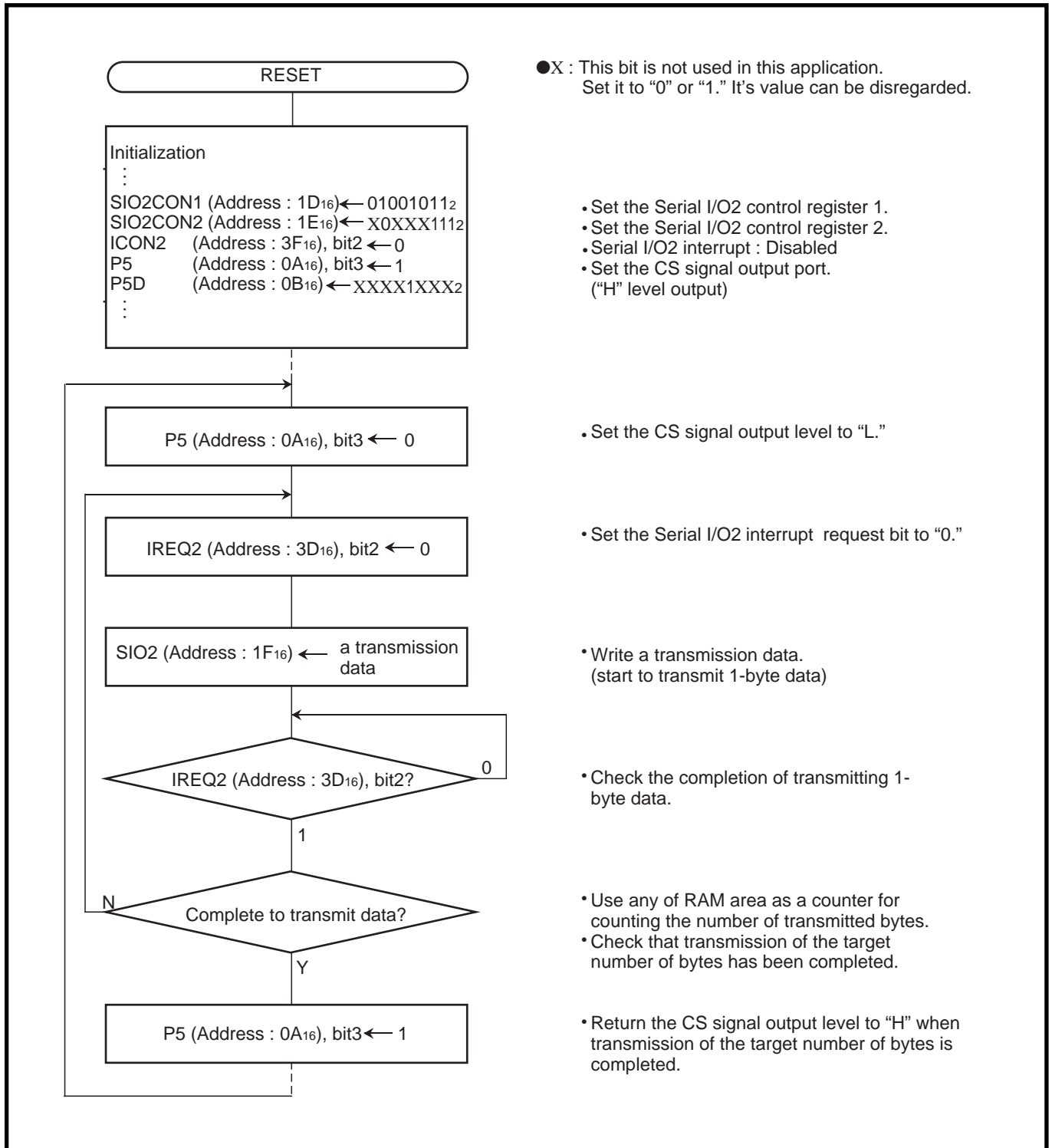


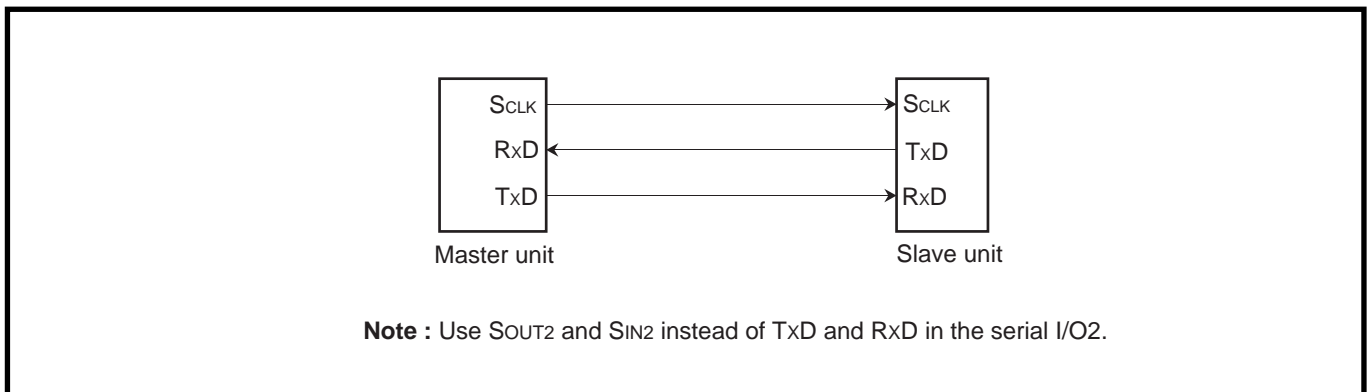
Fig. 2.3.31 Control procedure of serial I/O2 [Output of serial data]

# APPLICATION

## 2.3 Serial I/O

### (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. 2.3.32 Connection diagram [Cyclic transmission or reception of block data between microcomputers]**

- Specifications :**
- The serial I/O1 is used (clock synchronous serial I/O is selected).
  - Synchronous clock frequency : 125 kHz ( $f(XIN) = 8 \text{ MHz}$  is divided by 64)
  - 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/O1 receive interrupt request to generating of the next synchronizing clock is 428  $\mu\text{s}$ ).
2. "Heading adjustive time < interval between blocks" must be satisfied.

The communication is performed according to the timing shown below. In the slave unit, when a synchronizing clock is not input within a certain time (heading adjustive time), the next clock input is processed as the beginning (heading) of a block.

When a clock is input again after one block (8 byte) is received, the clock is ignored.

Figure 2.3.34 shows a setting of related registers.

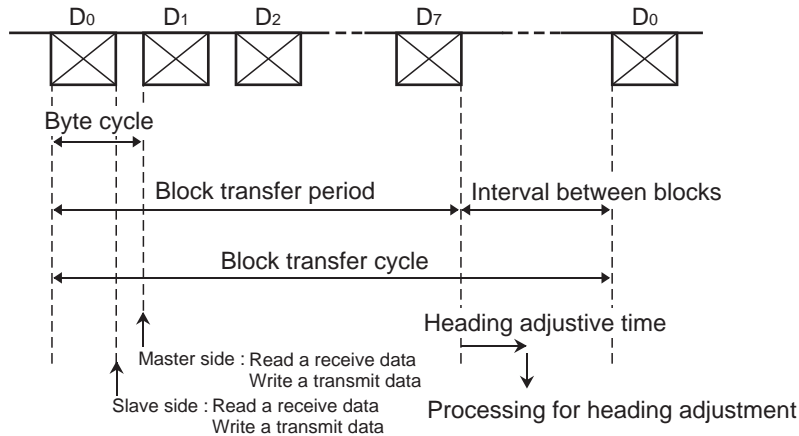


Fig. 2.3.33 Timing chart [Cyclic transmission or reception of block data between microcomputers]

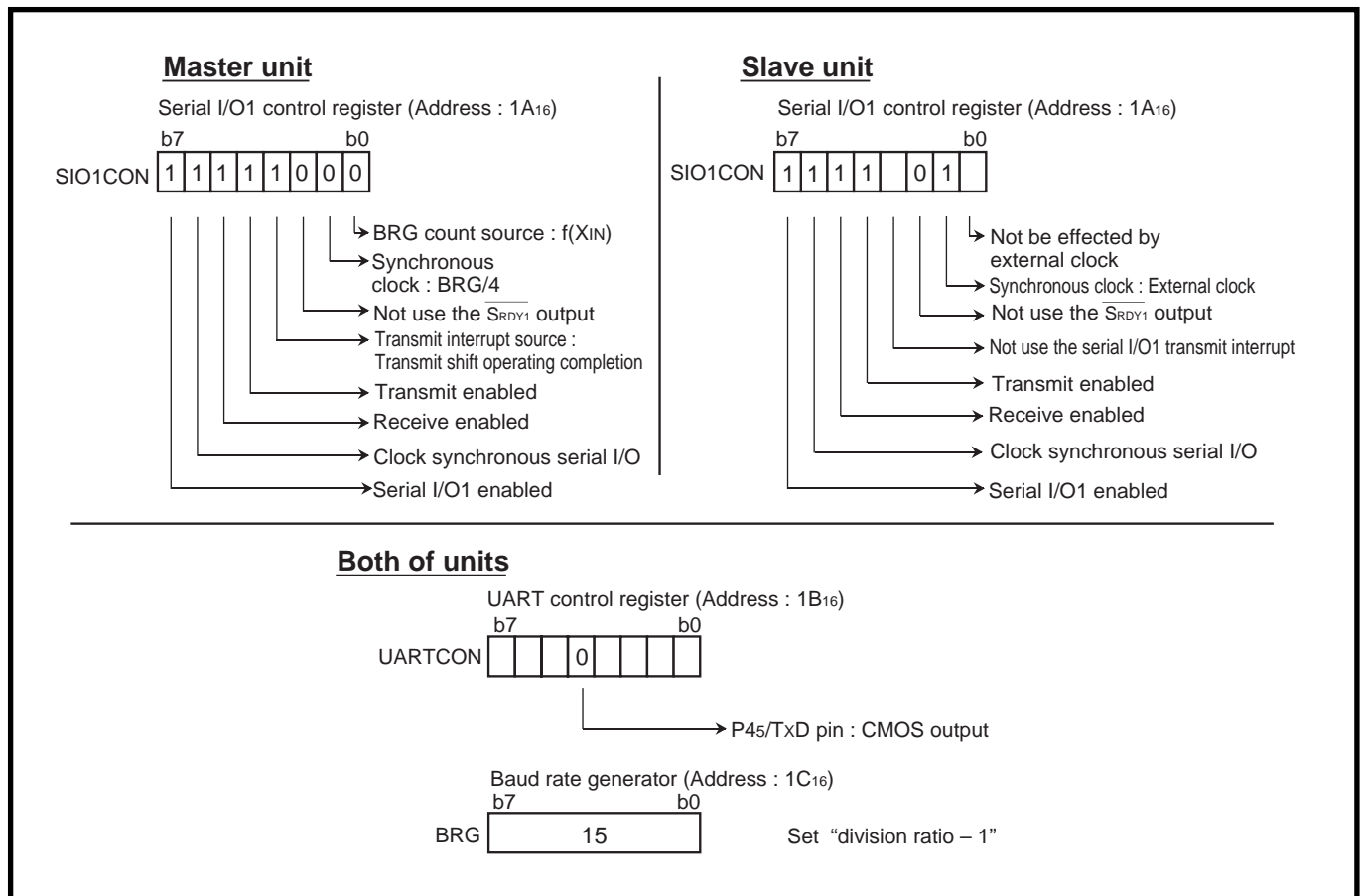


Fig. 2.3.34 Setting of related registers [Cyclic transmission or reception of block data between microcomputers]



# APPLICATION

## 2.3 Serial I/O

### Control procedure :

#### ① Control in the master unit

After a setting of the related registers is completed as shown in Figure 2.3.34, 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 2.3.33, therefore, take the timing into account and write transmission data. Read out the reception data when the Serial I/O1 transmit interrupt request bit is set to "1," or before the next transmission data is written to the Transmit buffer register.

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

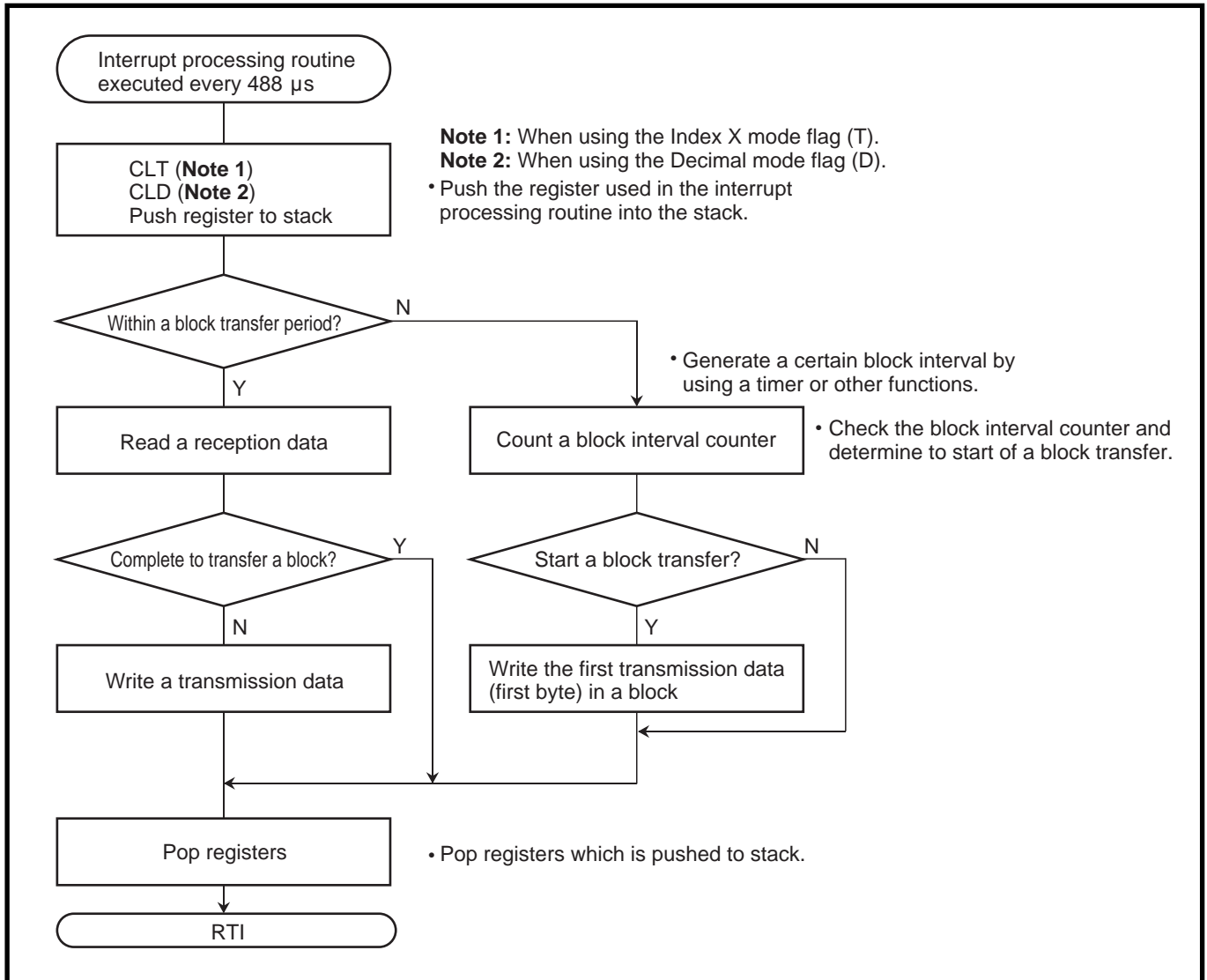


Fig. 2.3.35 Control in the master unit

## ② Control in the slave unit

After a setting of the related registers is completed as shown in Figure 2.3.34, the slave unit becomes the state which is received a synchronizing clock at all times, and the Serial I/O1 receive interrupt request bit is set to "1" every time an 8-bit synchronous clock is received.

By the serial I/O1 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/O1 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 Transmission buffer register.
2. The data to be received next is processed as the first 1 byte of the received data in the block.

Figure 2.3.36 shows the control in the slave unit using a serial I/O1 receive interrupt and any timer interrupt (for head adjustive).

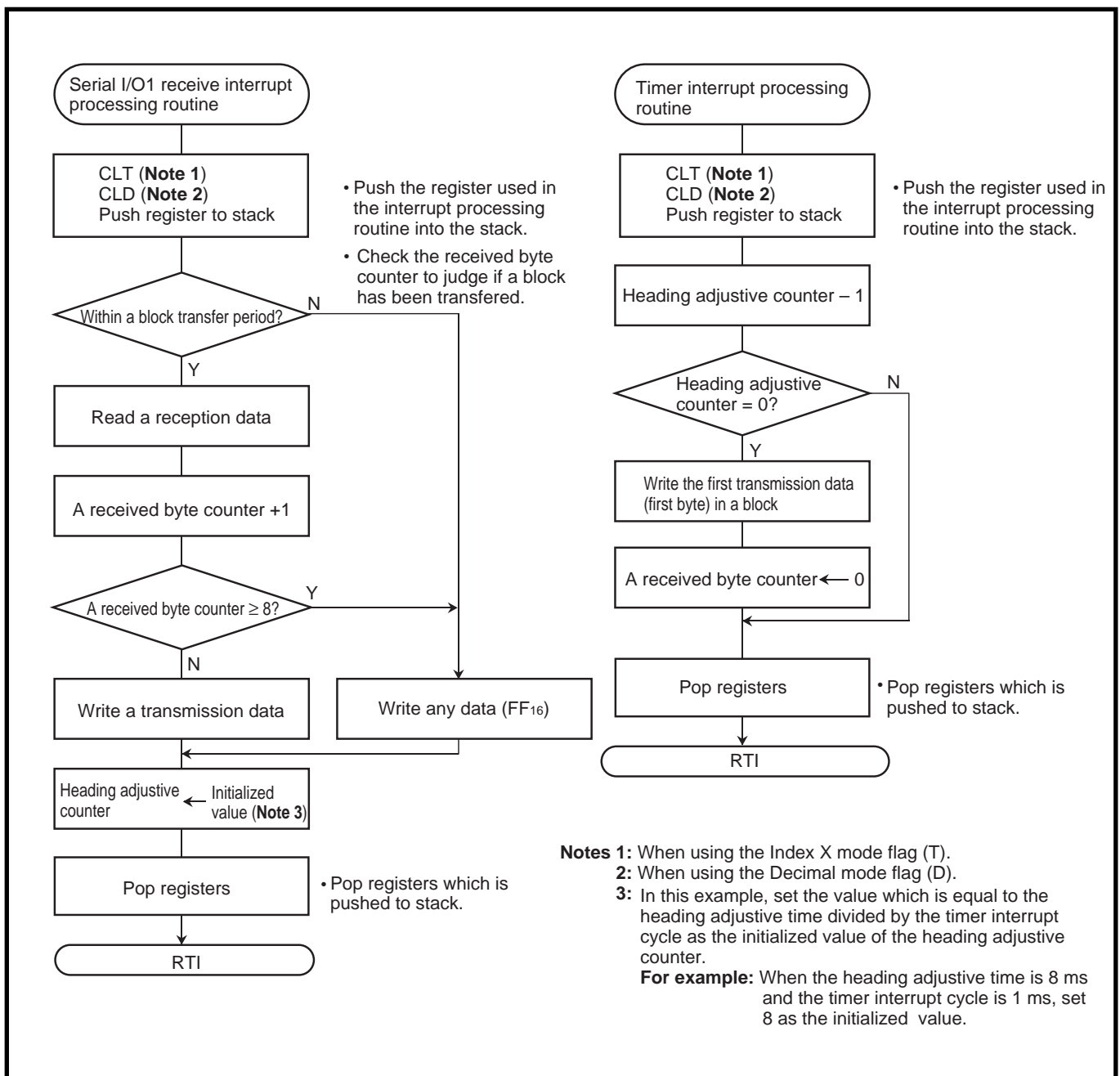


Fig. 2.3.36 Control in the slave unit

# APPLICATION

## 2.3 Serial I/O

### (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 P42 is used for communication control.

Figure 2.3.37 shows a connection diagram, and Figure 2.3.38 shows a timing chart.

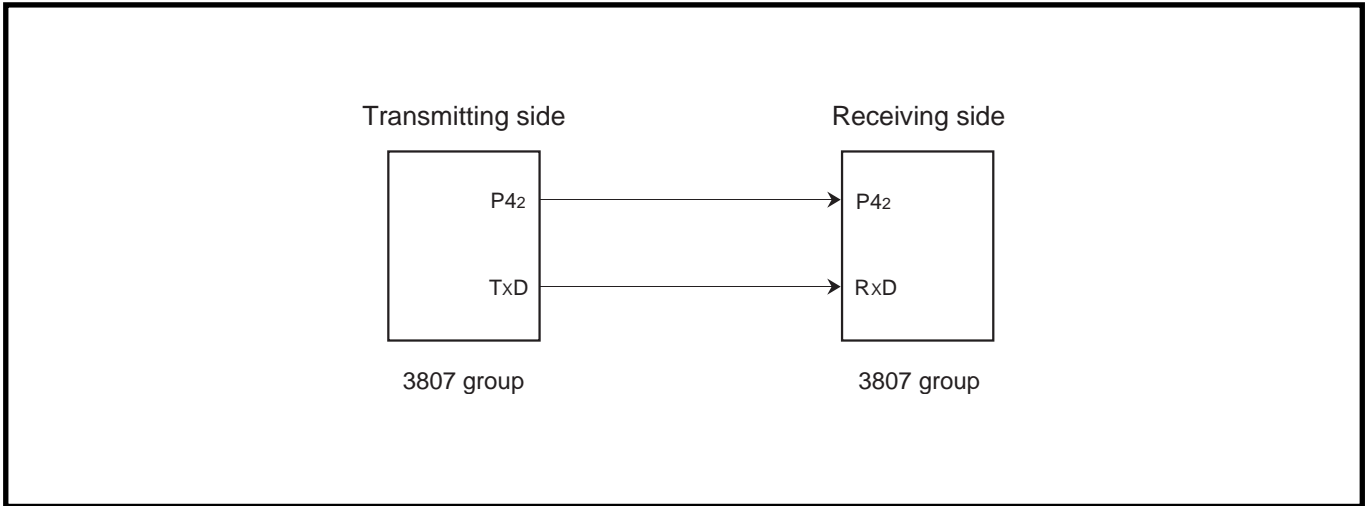


Fig. 2.3.37 Connection diagram [Communication using UART]

- Specifications :**
- The Serial I/O1 is used (UART is selected).
  - Transfer bit rate : 9600 bps ( $f(XIN) = 4.9152 \text{ MHz}$  is divided by 512)
  - Communication control using port P42  
(The output level of the port P42 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).

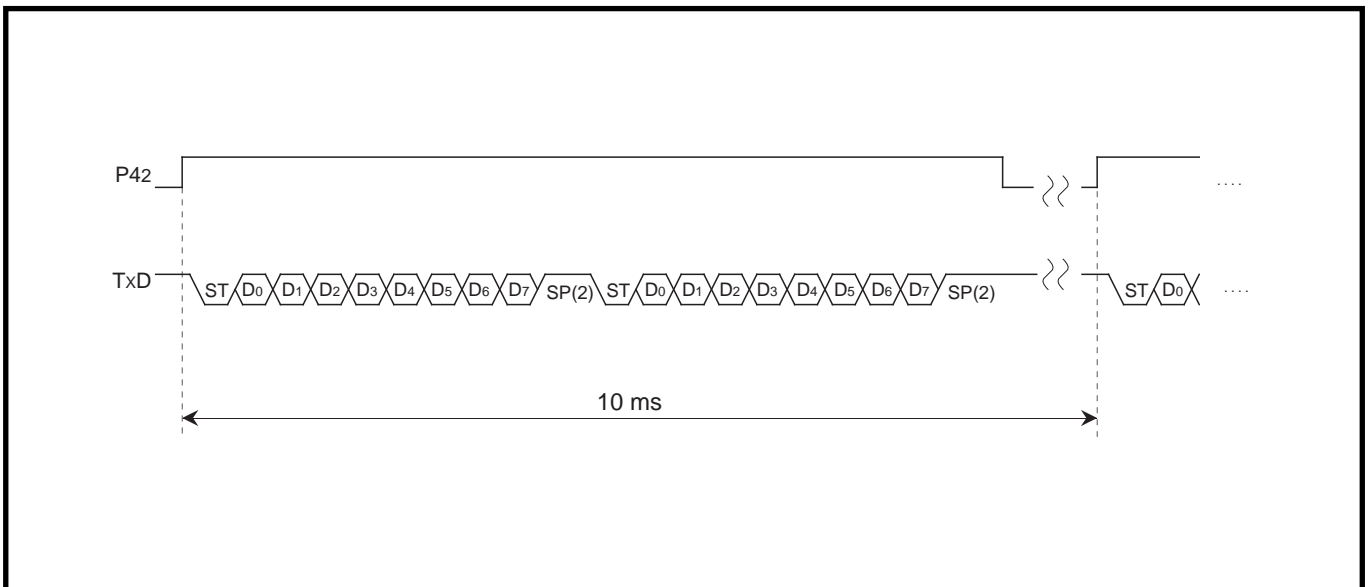


Fig. 2.3.38 Timing chart [Communication using UART]

Table 2.3.1 shows setting examples of Baud rate generator (BRG) values and transfer bit rate values, Figure 2.3.39 shows a setting of related registers at a transmitting side, and Figure 2.3.40 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/O1 control register (Address : 1A<sub>16</sub>) is set to “0,” a value of m is 1.

when bit 0 of the Serial I/O1 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/O1 control register (Address : 1A<sub>16</sub>).

# APPLICATION

## 2.3 Serial I/O

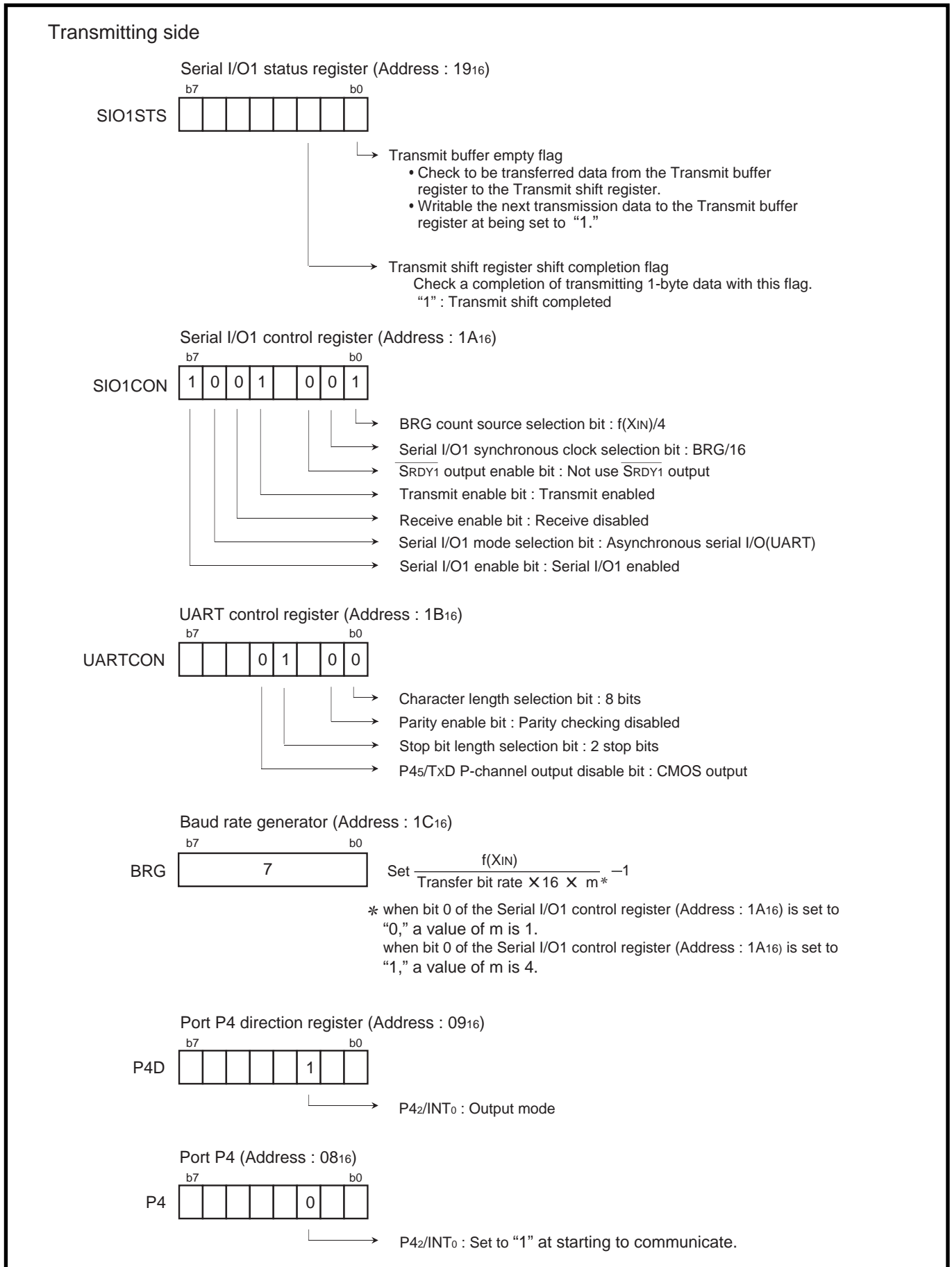


Fig. 2.3.39 Setting of related registers at a transmitting side [Communication using UART]

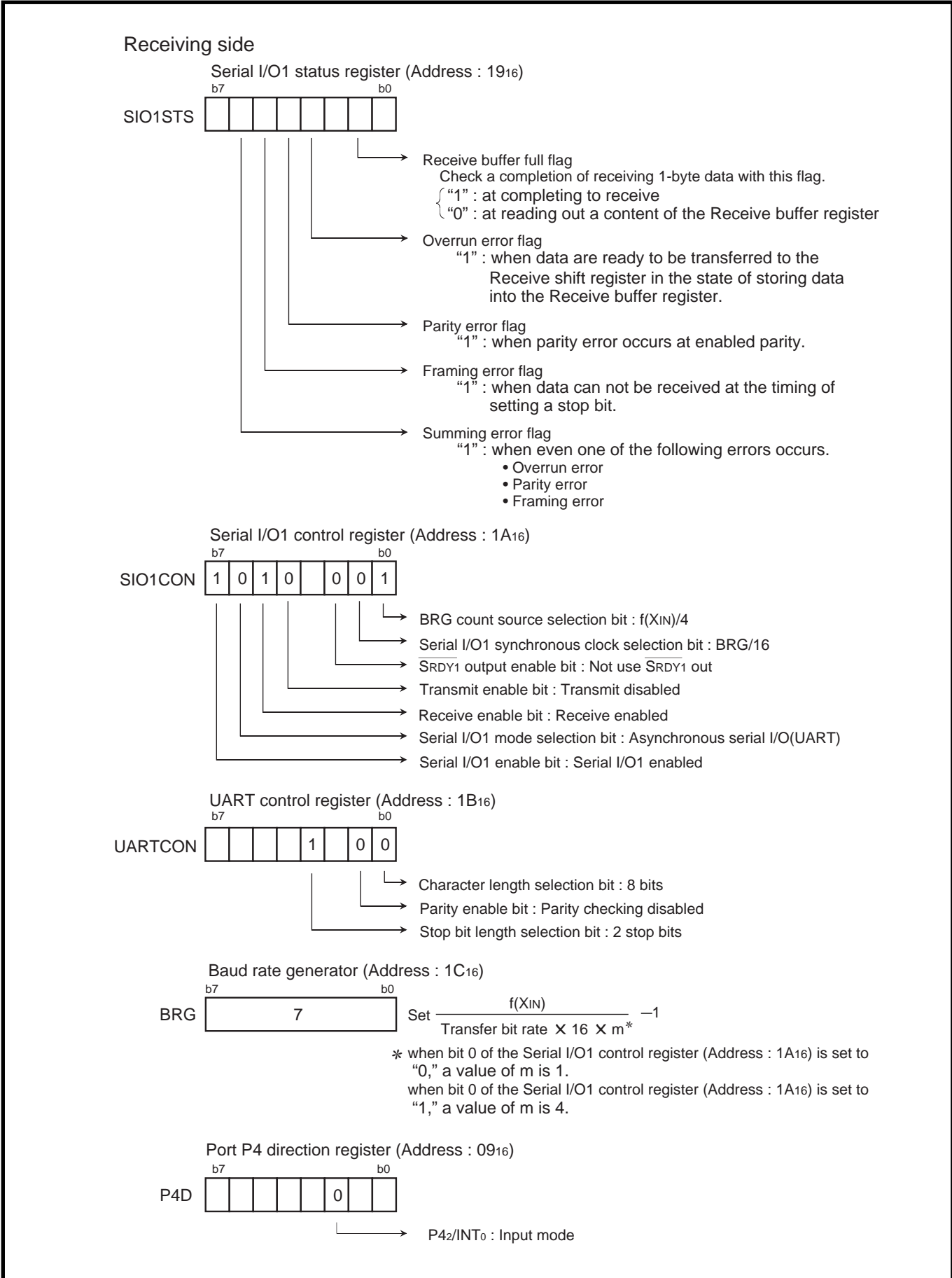
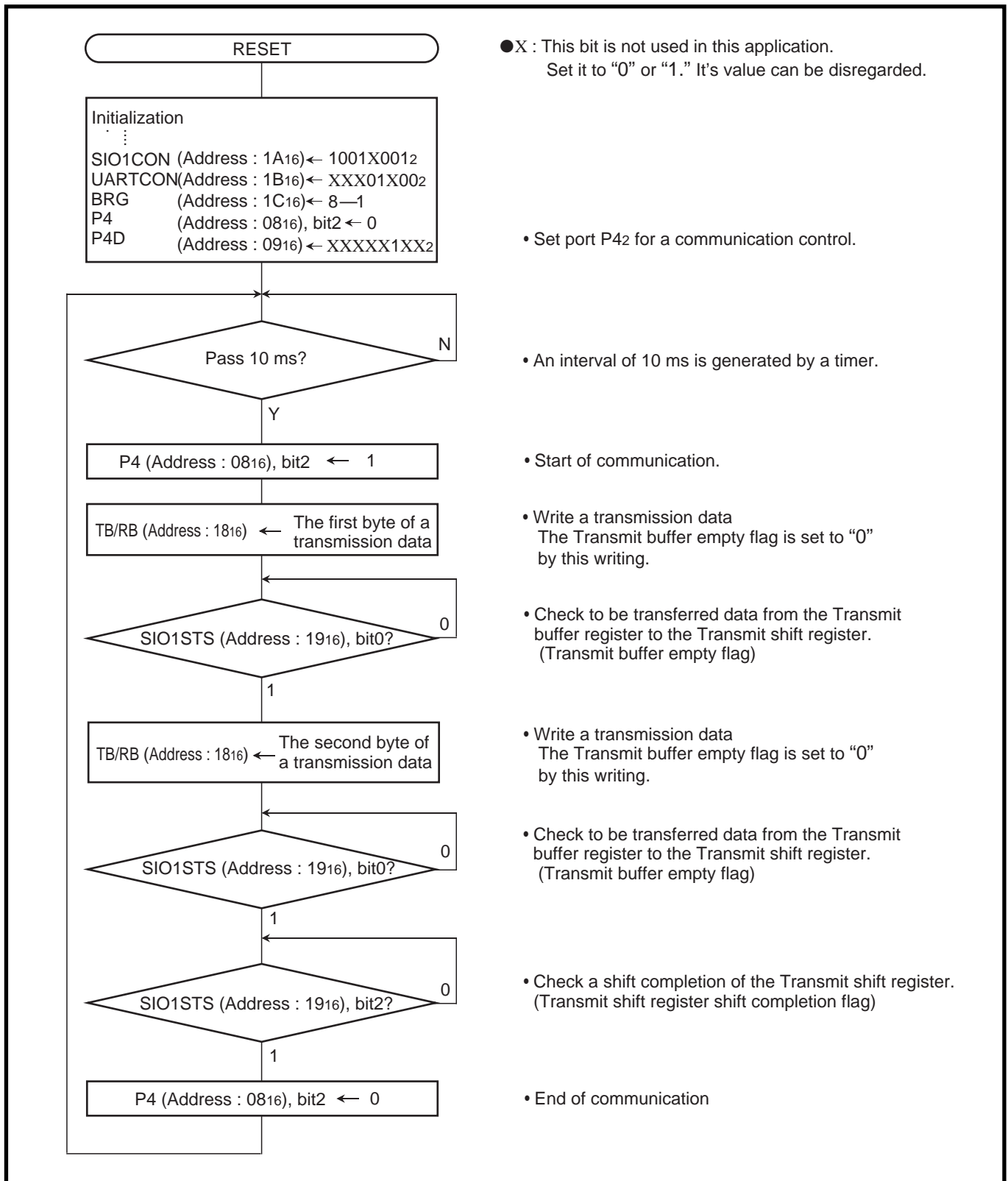


Fig. 2.3.40 Setting of related registers at a receiving side [Communication using UART]

# APPLICATION

## 2.3 Serial I/O

**Control procedure :** Figure 2.3.41 shows a control procedure at a transmitting side, and Figure 2.3.42 shows a control procedure at a receiving side.



**Fig. 2.3.41 Control procedure at a transmitting side [Communication using UART]**

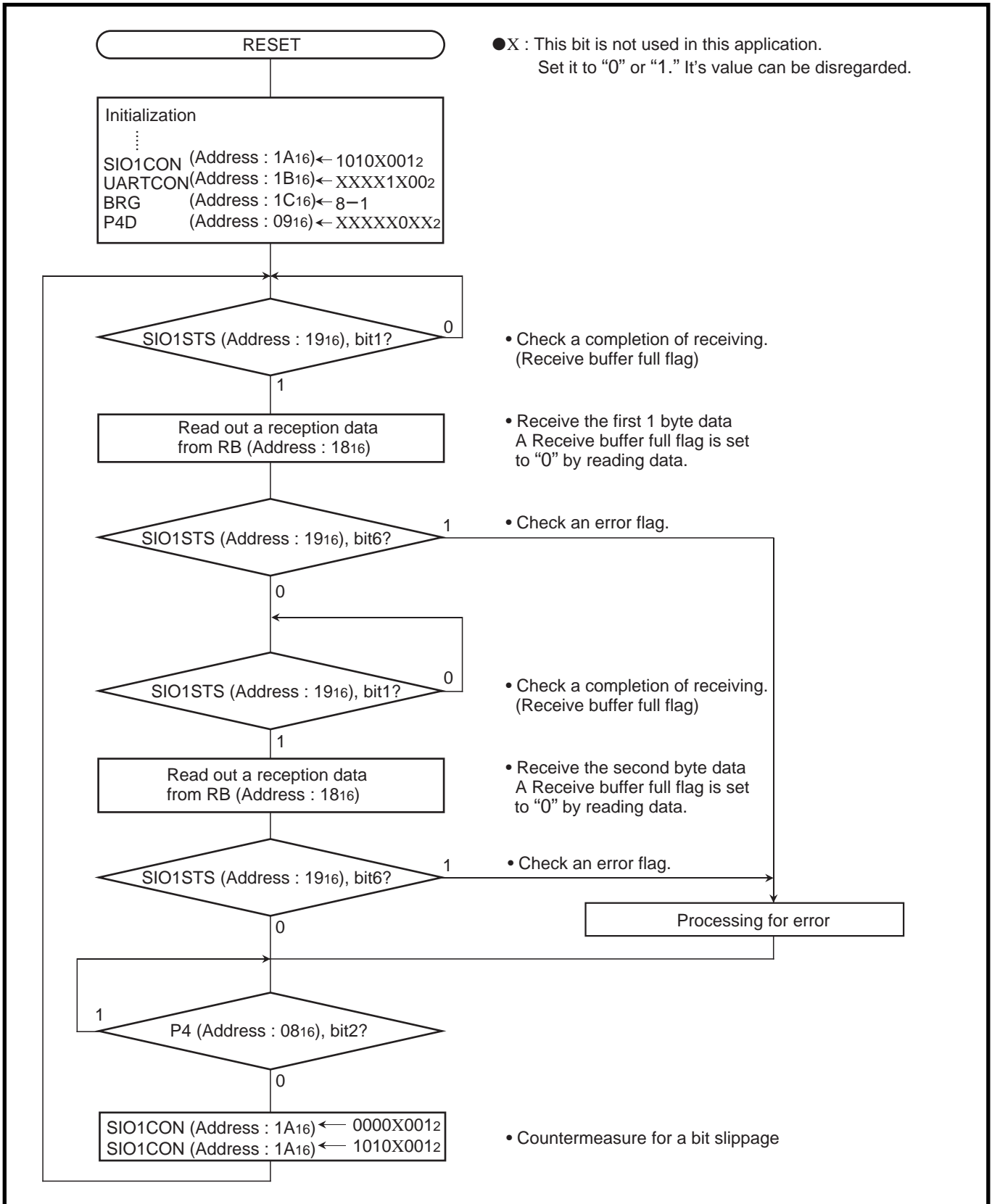


Fig. 2.3.42 Control procedure at a receiving side [Communication using UART]



# APPLICATION

## 2.4 Real time output port

### 2.4 Real time output port (RTP)

#### 2.4.1 Memory map of real time output port

002A <sub>16</sub>	Real time port register (RTP)
002B <sub>16</sub>	Real time port control register 0 (RTPCON0)
002C <sub>16</sub>	Real time port control register 1 (RTPCON1)
002D <sub>16</sub>	Real time port control register 2 (RTPCON2)
002E <sub>16</sub>	Real time port control register 3 (RTPCON3)
002F <sub>16</sub>	Timer A Low-order (TAL)
0030 <sub>16</sub>	Timer A High-order (TAH)
0031 <sub>16</sub>	Timer B Low-order (TBL)
0032 <sub>16</sub>	Timer B High-order (TBH)

Fig. 2.4.1 Memory map of real time output port related registers

#### 2.4.2 Related registers

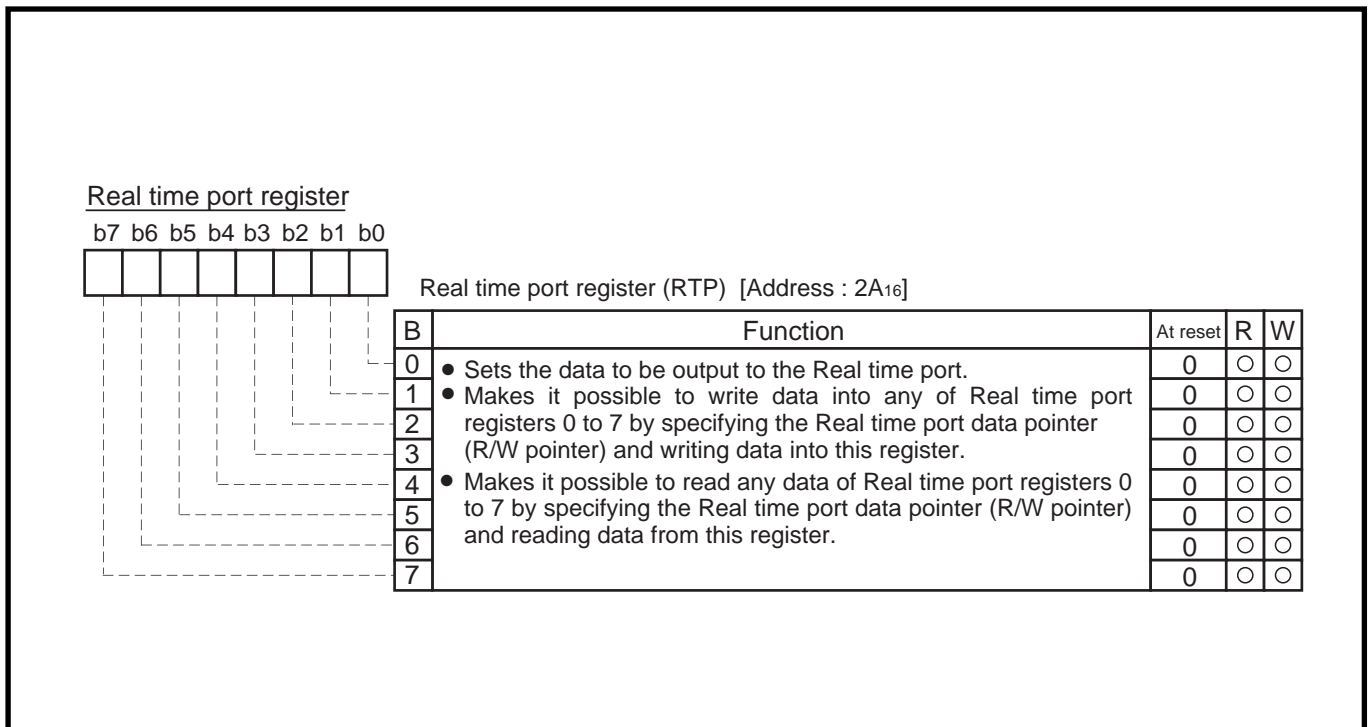
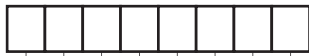


Fig. 2.4.2 Structure of Real time port register

### Real time port control register 0

b7 b6 b5 b4 b3 b2 b1 b0



Real time port control register 0 (RTPCON0) [Address : 2B<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Timer A, Timer B count source selection bit	0 : $f(X_{IN})/2$ (Note 1) 1 : $f(X_{IN})/16$ (Note 2)	0	○	○
1	Real time port • port allocation selection bit	0 : 4-4 division (Corresponding ports to the Timer A : P8 <sub>2</sub> -P8 <sub>5</sub> Corresponding ports to the Timer B : P8 <sub>6</sub> , P8 <sub>7</sub> , P3 <sub>0</sub> , P3 <sub>1</sub> ) 1 : 2-6 division (Corresponding ports to the Timer A : P8 <sub>2</sub> -P8 <sub>7</sub> Corresponding ports to the Timer B : P3 <sub>0</sub> , P3 <sub>1</sub> )	0	○	○
2	Timer A start trigger selection bit	0 : Internal trigger (occurs by writing "1" to bit 3.) 1 : External trigger (occurs by inputting trigger to the INT <sub>4</sub> pin.) (Note 3)	0	○	○
3	Timer A start trigger bit	0 : No operating by writing "0" 1 : Timer A starts counting by writing "1" (when bit 2 is set to "0")	0	○	○ (Note 4)
4	Timer A count source stop bit	0 : Operating (is set to "0" automatically at generating a start trigger.) 1 : Stop	1	○	○
5	Timer B start trigger selection bit	0 : Internal trigger (occurs by writing "1" to bit 6.) 1 : External trigger (occurs by inputting trigger to the INT <sub>4</sub> pin.) (Note 3)	0	○	○
6	Timer B start trigger bit	0 : No operating by writing "0" 1 : Timer B starts counting by writing "1" (when bit 5 is set to "0")	0	○	○ (Note 4)
7	Timer B count source stop bit	0 : Operating (is set to "0" automatically at generating a start trigger.) 1 : Stop	1	○	○

**Note 1:** In low-speed mode  $f(X_{CIN})/2$  is selected.

**Note 2:** In low-speed mode  $f(X_{CIN})/16$  is selected.

**Note 3:** The rising edge or falling edge of the external trigger is switched by the INT<sub>4</sub> interrupt edge selection bit (bit 4) of the interrupt edge selection register (Address : 3A<sub>16</sub>.) (However, when the One-shot pulse generation mode is selected, a rising/falling double edge trigger is generated in spite of the contents of the INT<sub>4</sub> interrupt edge selection bit.)

**Note 4:** At a read operation, "0" is always read out.

Fig. 2.4.3 Structure of Real time port control register 0

# APPLICATION

## 2.4 Real time output port

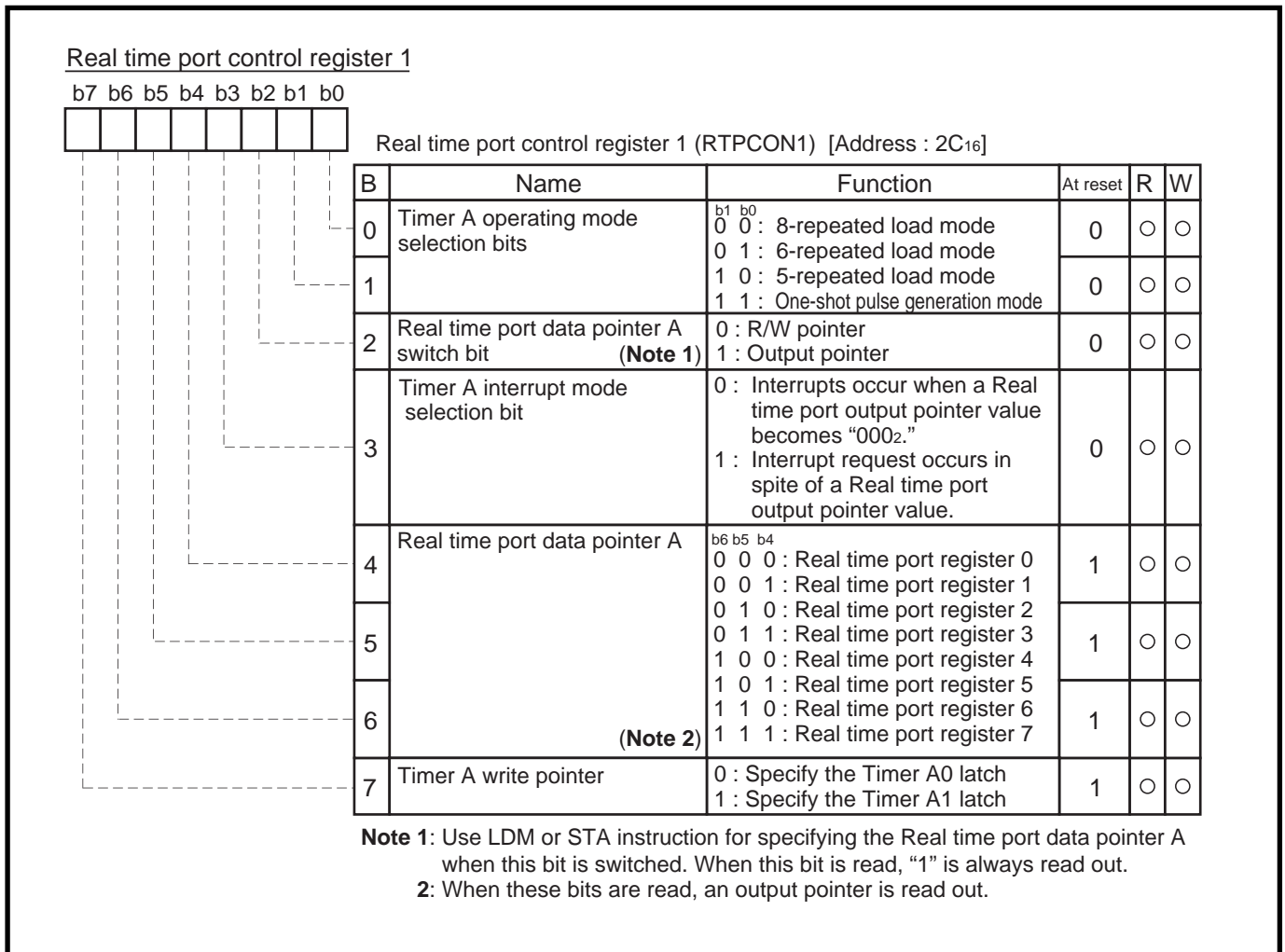


Fig. 2.4.4 Structure of Real time port control register 1

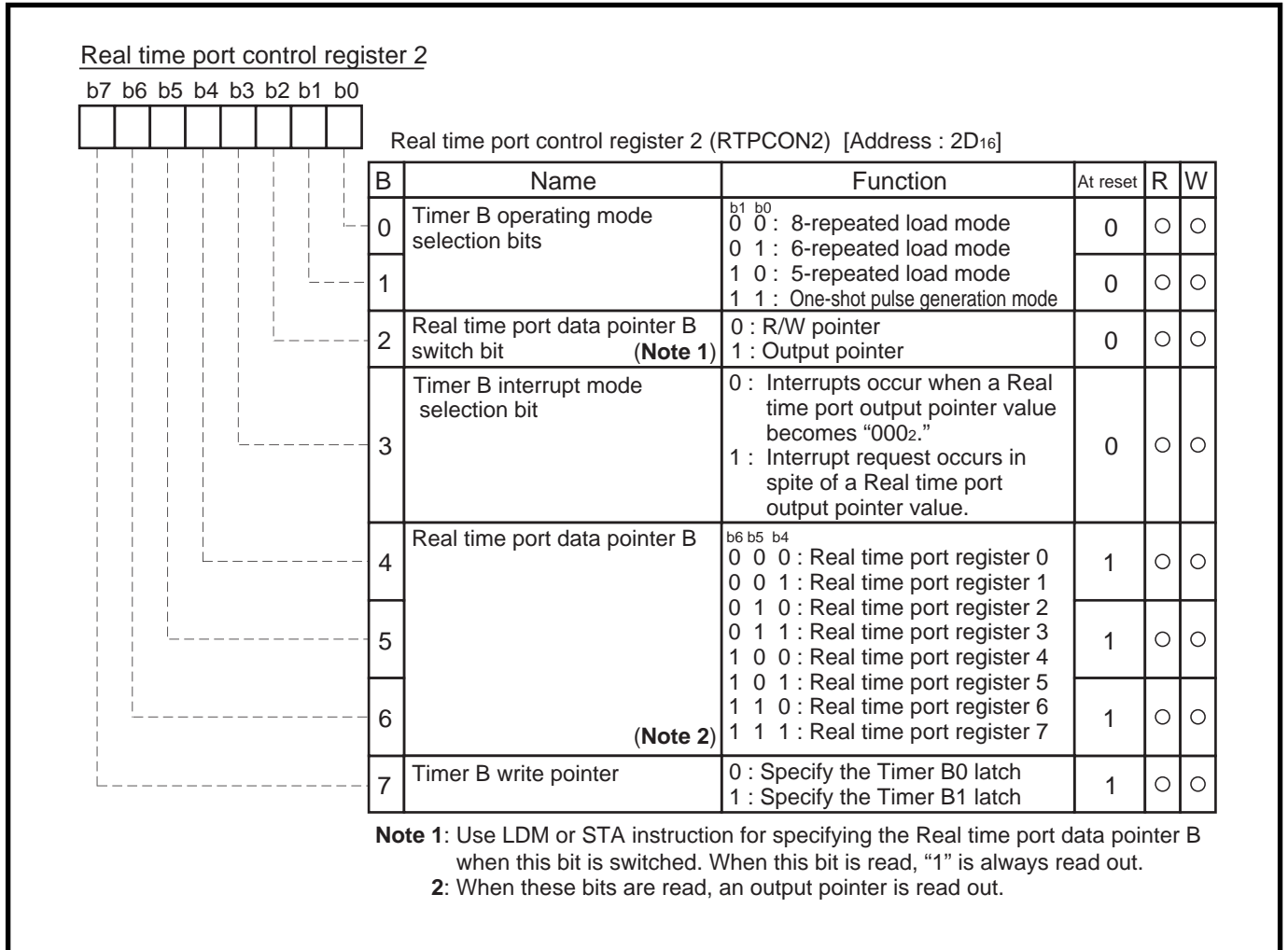


Fig. 2.4.5 Structure of Real time port control register 2

# APPLICATION

## 2.4 Real time output port

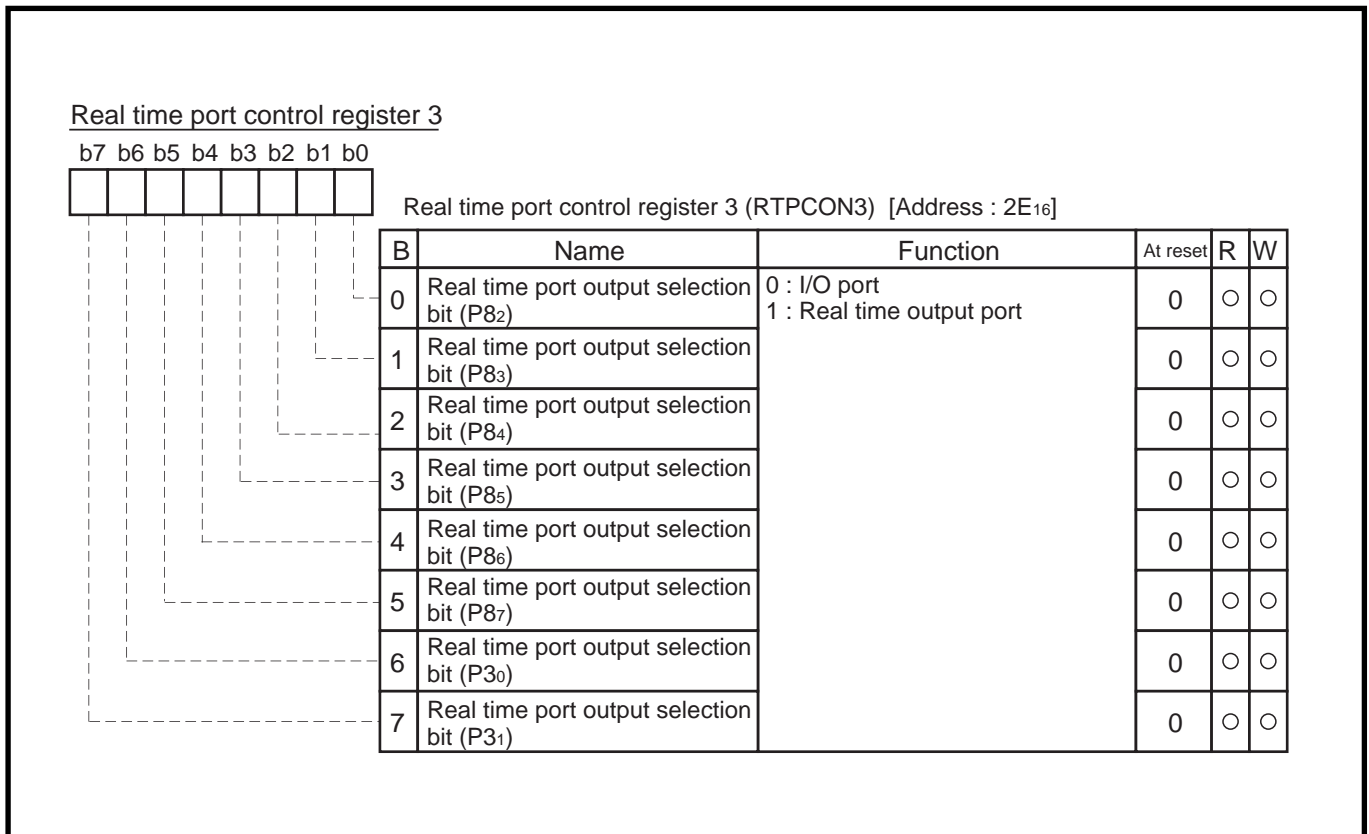


Fig. 2.4.6 Structure of Real time port control register 3

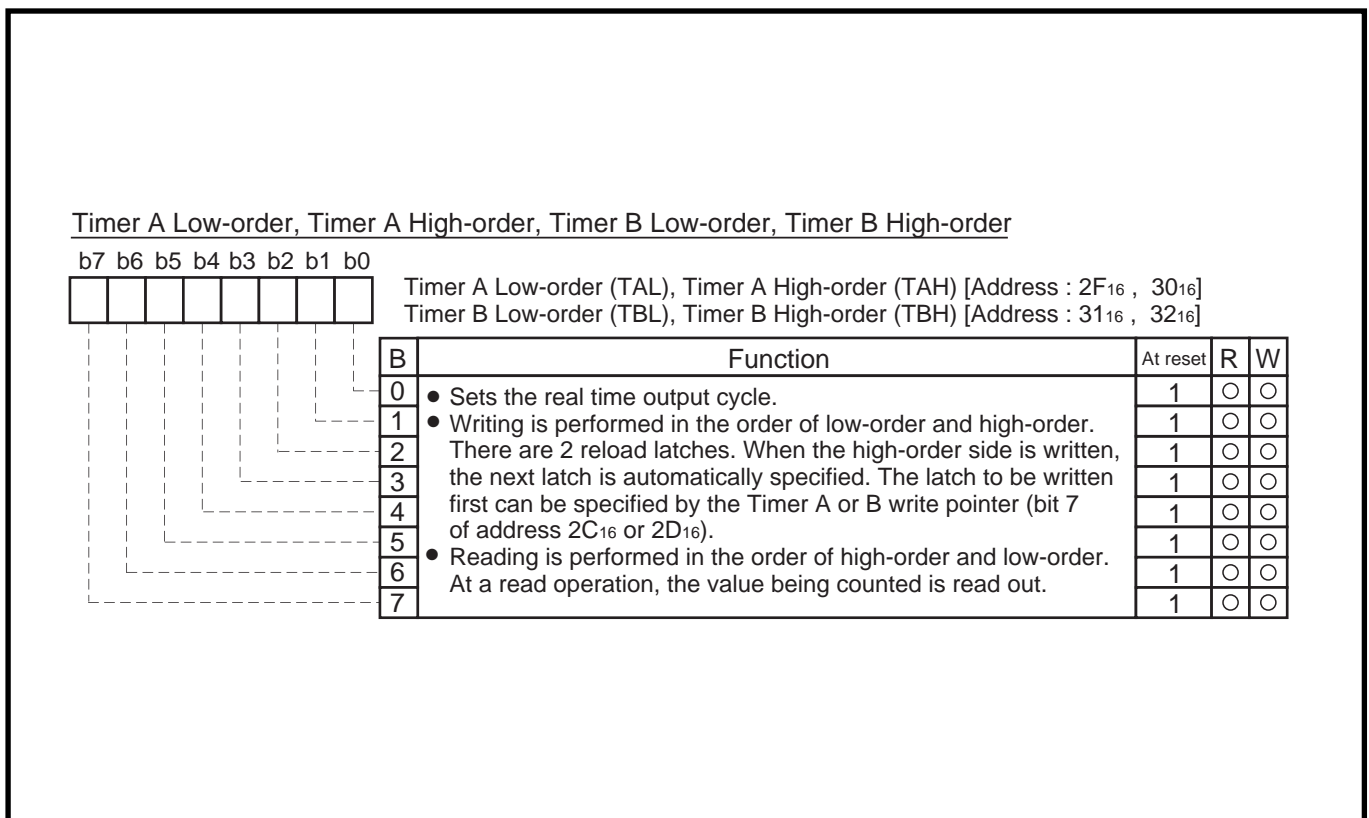


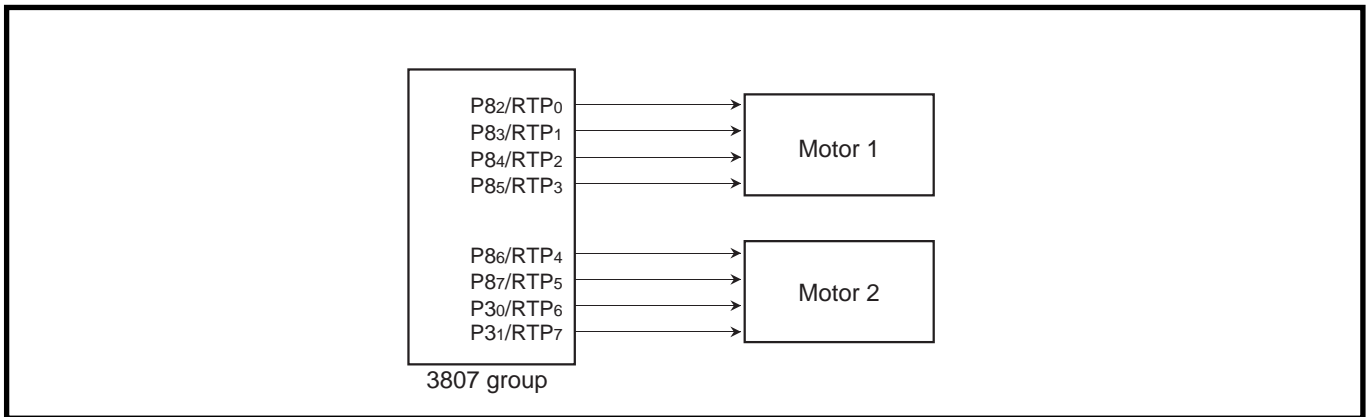
Fig. 2.4.7 Structure of Timer A Low-order, Timer A High-order, Timer B Low-order, Timer B High-order

### 2.4.3 Real time output port application examples

#### Control of stepping motor

**Outline :** The rotation of the stepping motor is controlled by using Real time output ports.

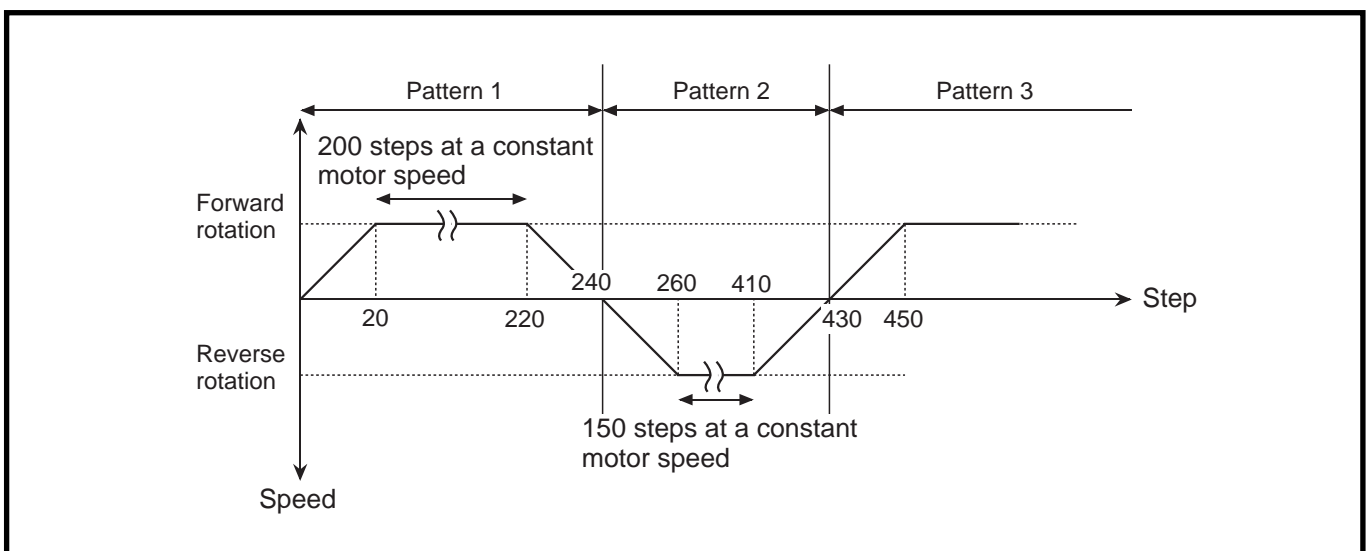
Figure 2.4.8 shows a connection diagram.



**Fig. 2.4.8 Connection diagram**

- Specifications :**
- Each of two motors is controlled by using four Real time output ports.
  - Clock  $f(XIN) = 8 \text{ MHz}$
  - The same data table is used for acceleration and deceleration. (20 steps, 500 pps max.)
  - The value of the Timer A and B are updated by each interrupt processing routine.
  - When the Timer A and/or B stops, the "L" level is output.

Figure 2.4.9 shows the operation patterns of the motor to be controlled in this application example. The Timer A and the Timer B can control the motor independently with different operation patterns.



**Fig. 2.4.9 Operation patterns of motor**

# APPLICATION

## 2.4 Real time output port

The motor is accelerated and decelerated by updating the timer value in the Timer interrupt processing routine. Figure 2.4.10 shows an example of timer table for acceleration and deceleration. A table common to both Timer A and Timer B is used in this application example.

As shown in the following figure, the motor speed is controlled by setting a value in the low-order side of the table first at acceleration and by setting a value the high-order side of the table first at deceleration. At a constant motor speed, the motor operation is continued with the last timer value of acceleration.

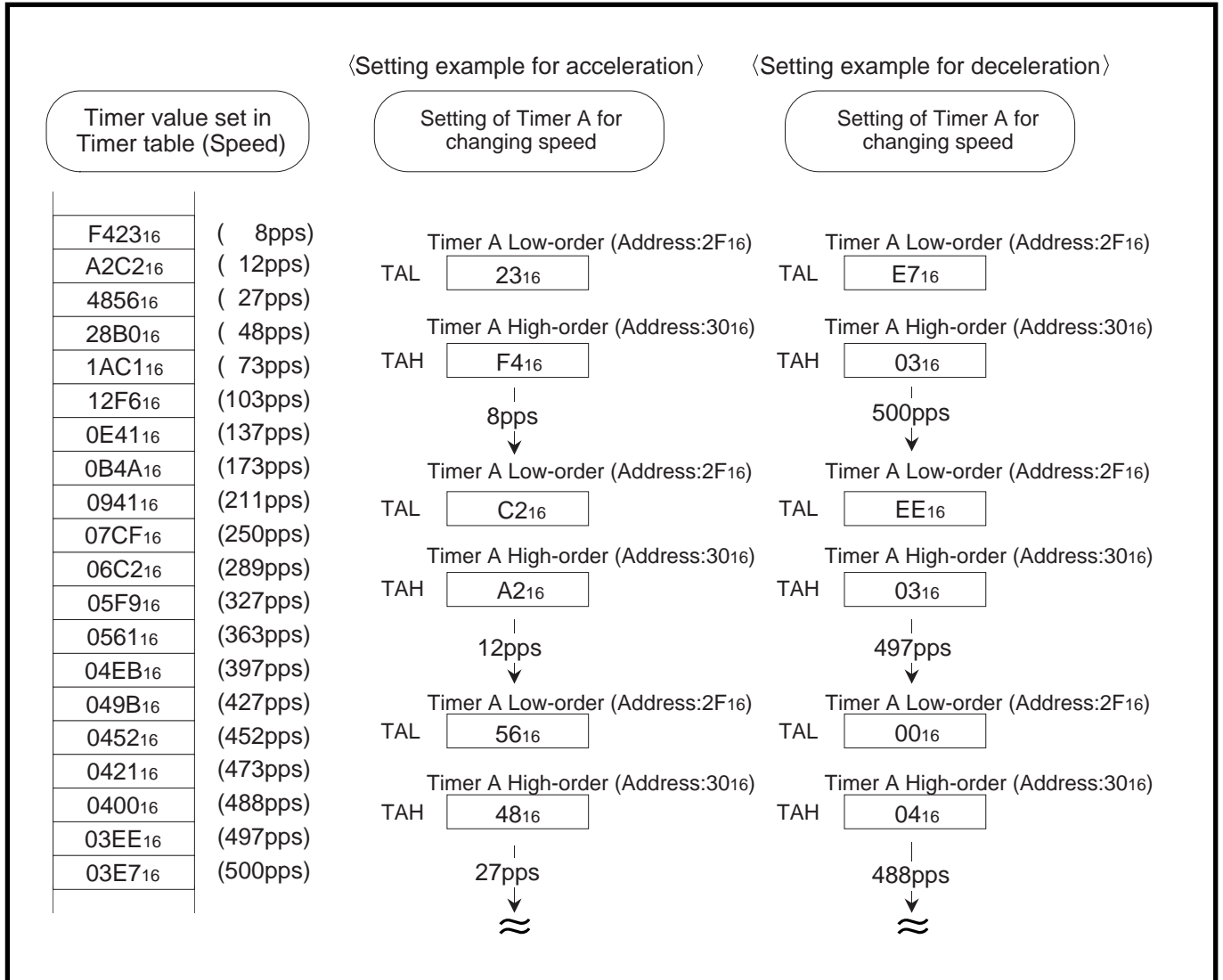


Fig. 2.4.10 Example of timer table for acceleration and deceleration

Figure 2.4.11 shows an example of operation pattern table to operate the motor by the operation patterns shown in Figure 2.4.9. The total number of operation patterns, the direction of motor rotation and number of steps at a constant motor speed are set in this table.

The motor can be rotated by an arbitrary distance by changing this number of steps. (In this application example, up to 255 steps can be set.)

An operation pattern table is set for each of the Timer A and the Timer B.

⋮			
04 <sub>16</sub>	Total number of operation patterns		
00 <sub>16</sub>	Forward rotation	}	Operation pattern 1
C8 <sub>16</sub>	200 steps at a constant motor speed		
FF <sub>16</sub>	Reverse rotation	}	Operation pattern 2
96 <sub>16</sub>	150 steps at a constant motor speed		
00 <sub>16</sub>	Forward rotation	}	Operation pattern 3
64 <sub>16</sub>	100 steps at a constant motor speed		
FF <sub>16</sub>	Reverse rotation	}	Operation pattern 4
FA <sub>16</sub>	250 steps at a constant motor speed		
⋮			

Note : "00<sub>16</sub>" is defined as a forward rotation in this example.  
"FF<sub>16</sub>" is defined as a reverse rotation in this example.

**Fig. 2.4.11 Example of operation pattern table**

Figure 2.4.12 shows an example of output data table. Output data is selected in the 4 types of tables shown in Figure 2.4.12 according to the information on forward rotation and reverse rotation referenced in the operation pattern table shown in Figure 2.4.11, and then set in Real time port registers 0 to 7. For example, in case the Timer B continues to control the motor in the forward direction when the data of operation pattern 2 is set after the Timer A has output operation pattern 1, the data of table 3 is set in Real time port registers 0 to 7.

	〈Table 1〉	〈Table 2〉	〈Table 3〉	〈Table 4〉
	RTP7–RTP4 : Forward rotation	RTP7–RTP4 : Reverse rotation	RTP7–RTP4 : Forward rotation	RTP7–RTP4 : Reverse rotation
	RTP3–RTP0 : Forward rotation	RTP3–RTP0 : Forward rotation	RTP3–RTP0 : Reverse rotation	RTP3–RTP0 : Reverse rotation
	b7                      b0	b7                      b0	b7                      b0	b7                      b0
Real time port register 7	0 0 0 1 0 0 0 1	1 0 0 1 0 0 0 1	0 0 0 1 1 0 0 1	1 0 0 1 1 0 0 1
	b7                      b0	b7                      b0	b7                      b0	b7                      b0
Real time port register 6	0 0 1 1 0 0 1 1	1 0 0 0 0 0 1 1	0 0 1 1 1 0 0 0	1 0 0 0 1 0 0 0
	b7                      b0	b7                      b0	b7                      b0	b7                      b0
Real time port register 5	0 0 1 0 0 0 1 0	1 1 0 0 0 0 1 0	0 0 1 0 1 1 0 0	1 1 0 0 1 1 0 0
	b7                      b0	b7                      b0	b7                      b0	b7                      b0
Real time port register 4	0 1 1 0 0 1 1 0	0 1 0 0 0 1 1 0	0 1 1 0 0 1 0 0	0 1 0 0 0 1 0 0
	b7                      b0	b7                      b0	b7                      b0	b7                      b0
Real time port register 3	0 1 0 0 0 1 0 0	0 1 1 0 0 1 0 0	0 1 0 0 0 1 1 0	0 1 1 0 0 1 1 0
	b7                      b0	b7                      b0	b7                      b0	b7                      b0
Real time port register 2	1 1 0 0 1 1 0 0	0 0 1 0 1 1 0 0	1 1 0 0 0 0 1 0	0 0 1 0 0 0 1 0
	b7                      b0	b7                      b0	b7                      b0	b7                      b0
Real time port register 1	1 0 0 0 1 0 0 0	0 0 1 1 1 0 0 0	1 0 0 0 0 0 1 1	0 0 1 1 0 0 1 1
	b7                      b0	b7                      b0	b7                      b0	b7                      b0
Real time port register 0	1 0 0 1 1 0 0 1	0 0 0 1 1 0 0 1	1 0 0 1 0 0 0 1	0 0 0 1 0 0 0 1

**Fig. 2.4.12 Example of output data table**



# APPLICATION

## 2.4 Real time output port

Figure 2.4.13 shows the waveforms which are output from RTP0 to RTP7 as a result that the Timer A and the Timer B are operated by using the data of Figure 2.4.10 to Figure 2.4.12. This timing chart is for the case where the Timer A controls operation pattern 1 and the Timer B controls operation pattern 3.

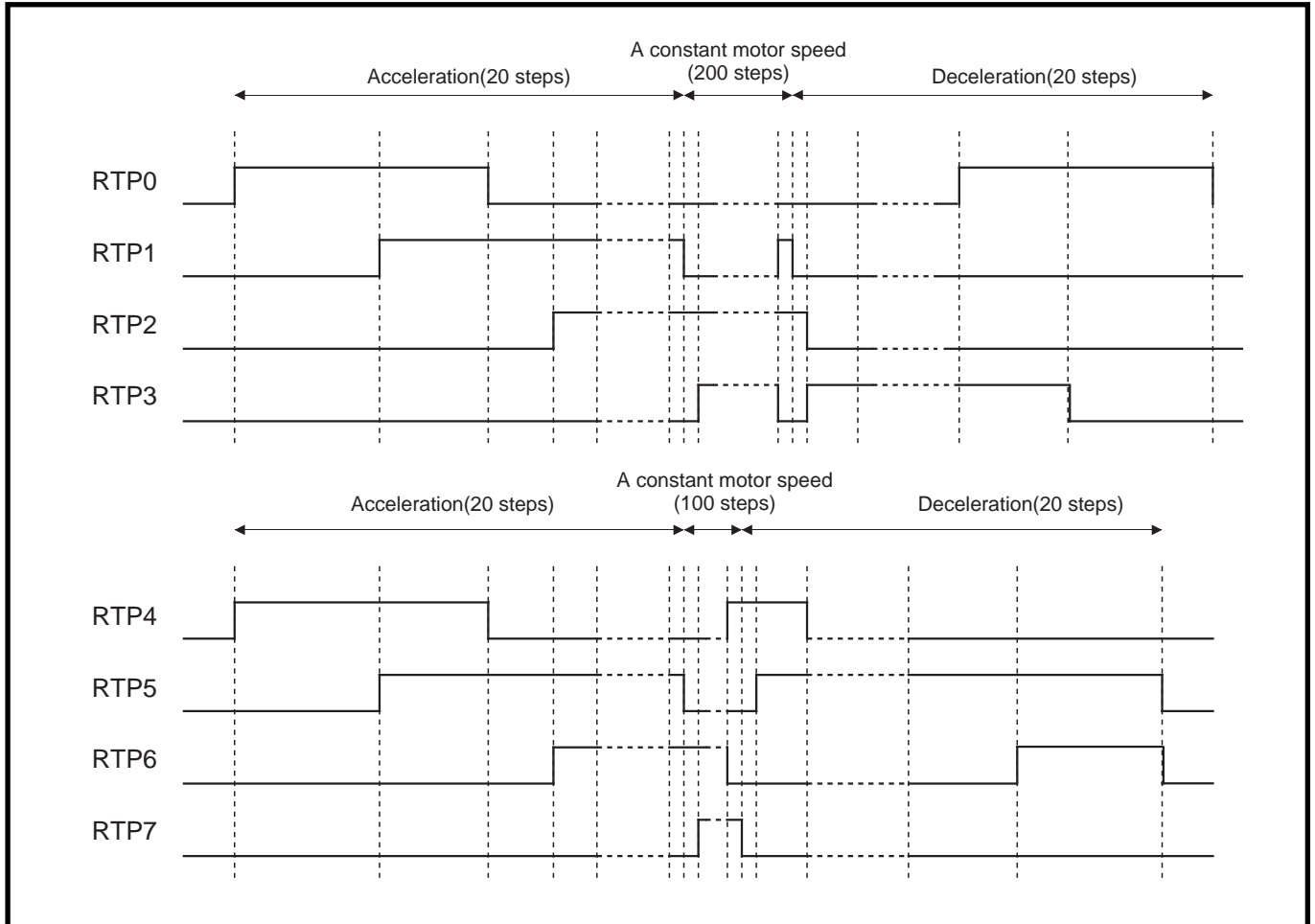


Fig. 2.4.13 Timing of Real time output

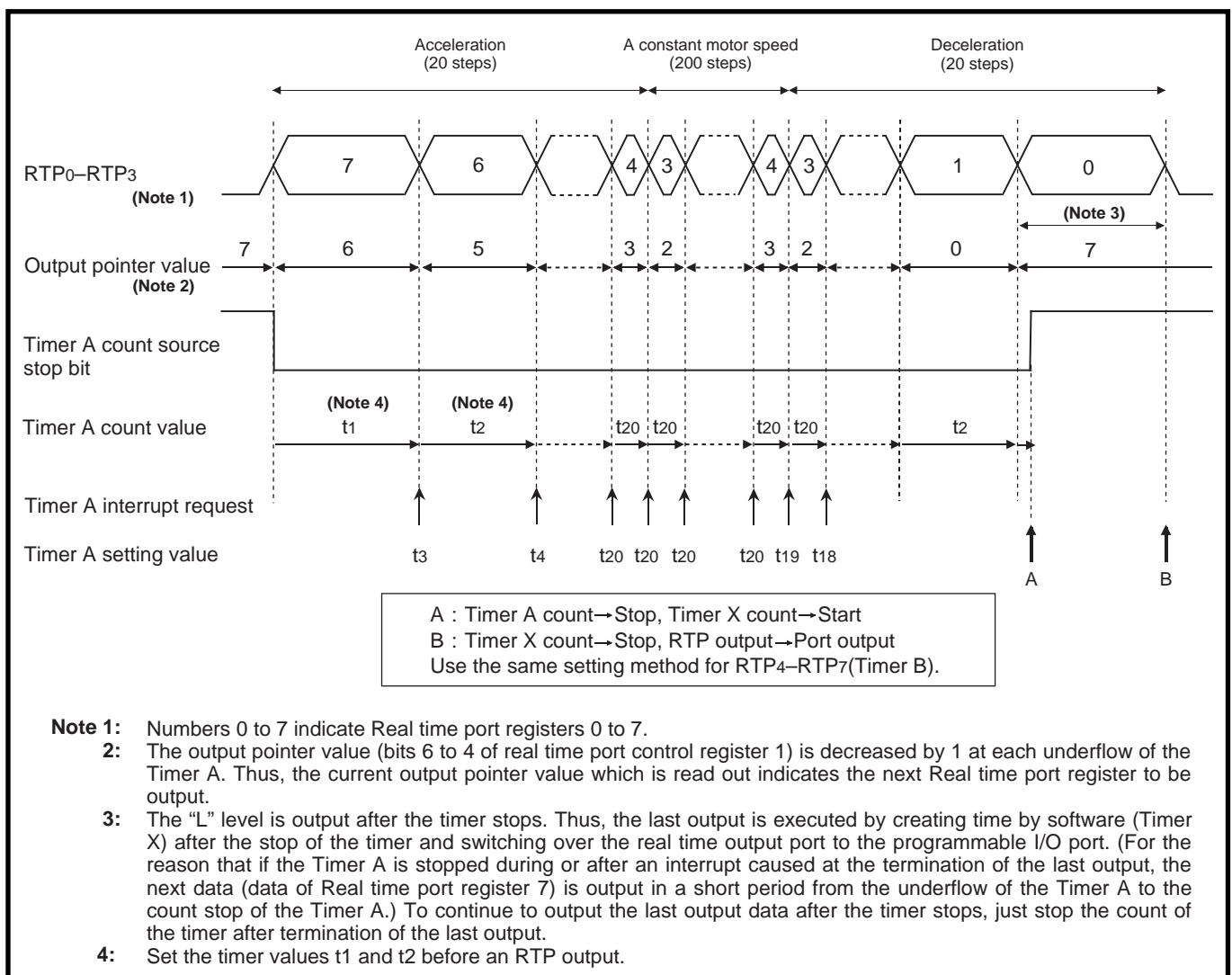
Figure 2.4.14 shows the setting method and output timing for the Timer A. The same setting method is used even for the Timer B. Before the count of the Timer A is started, initial values ( $t_1$ ,  $t_2$ ) are set in the Timer A1 latch and the Timer A0 latch. After the count of the Timer A is started, the timer value ( $t_3$ , ...) is updated in the Timer A interrupt processing routine.

The next latch is automatically specified each time a value is set in the timer, so it is not necessary to specify a write latch in bit 7 of RTPCON1 when the timer value is updated.

In this application example, the real time output port is switched over to the programmable I/O port after termination of the last output because the "L" level is output from RTP0 to RTP7 when the timer stops as a matter of specification. However, when the count of the Timer A is stopped and the real time output port is switched over to the programmable I/O port after termination of the last output, the next RTP data is output in a short period from an underflow of the Timer A till a count stop of the Timer A. To avoid outputting the next RTP data in the short period, the count of the Timer A is stopped at a start of the last output (though the last output data is output) and the last output period is counted by using different timers (Timer X for the Timer A and timer Y for the Timer B in this case). After that counting, when the Timer X underflows, the real time output port is switched over to the programmable I/O port and the "L" level is output.

To continue to output the last output data after the timer stops, just stop the count of the Timer A after termination of the last output.

Figure 2.4.14 shows the setting method and output timing and Figure 2.4.15 to Figure 2.4.18 show the control procedures for related registers.



**Fig. 2.4.14 Setting method and output timing**

# APPLICATION

## 2.4 Real time output port

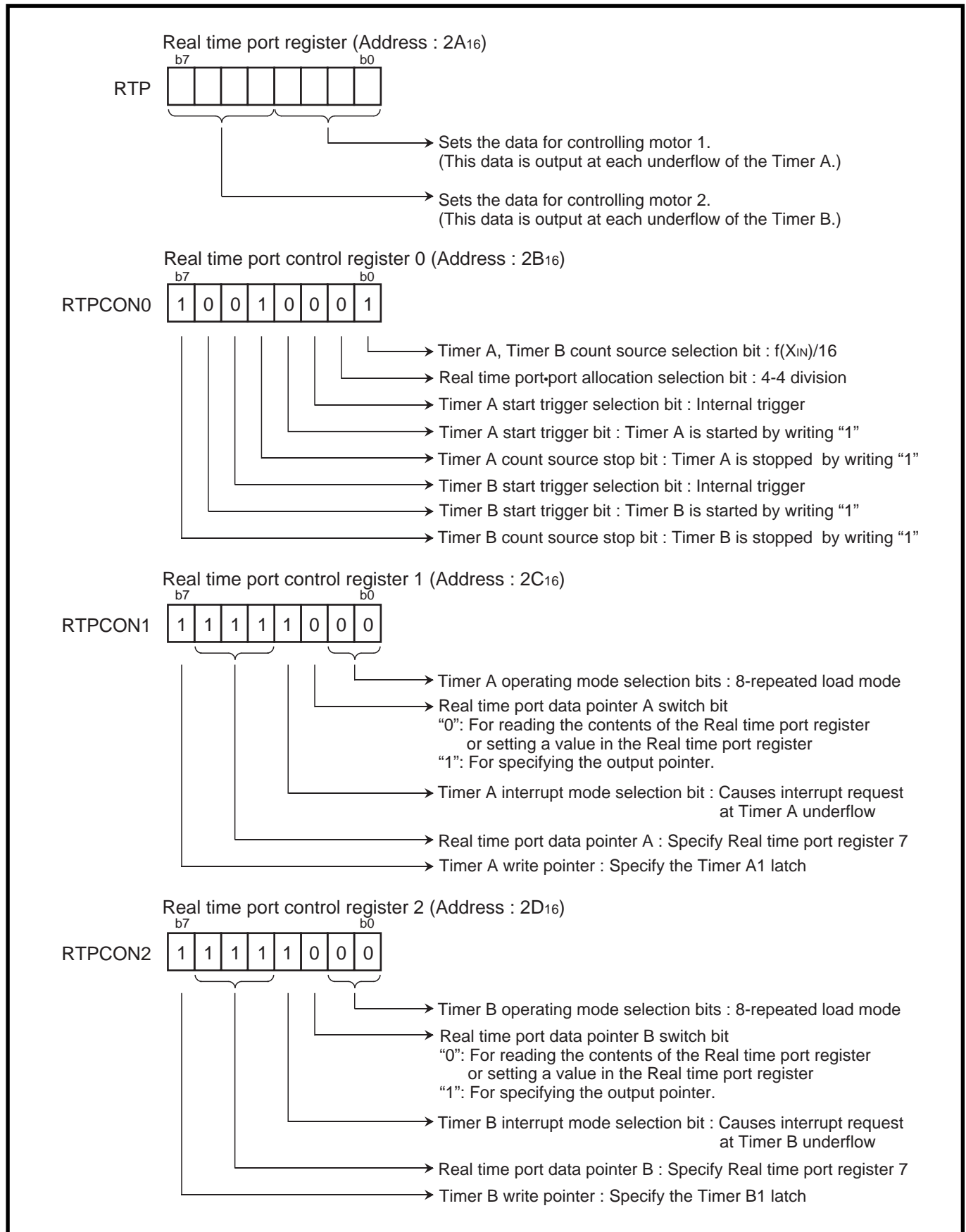


Fig. 2.4.15 Setting of related registers (1)

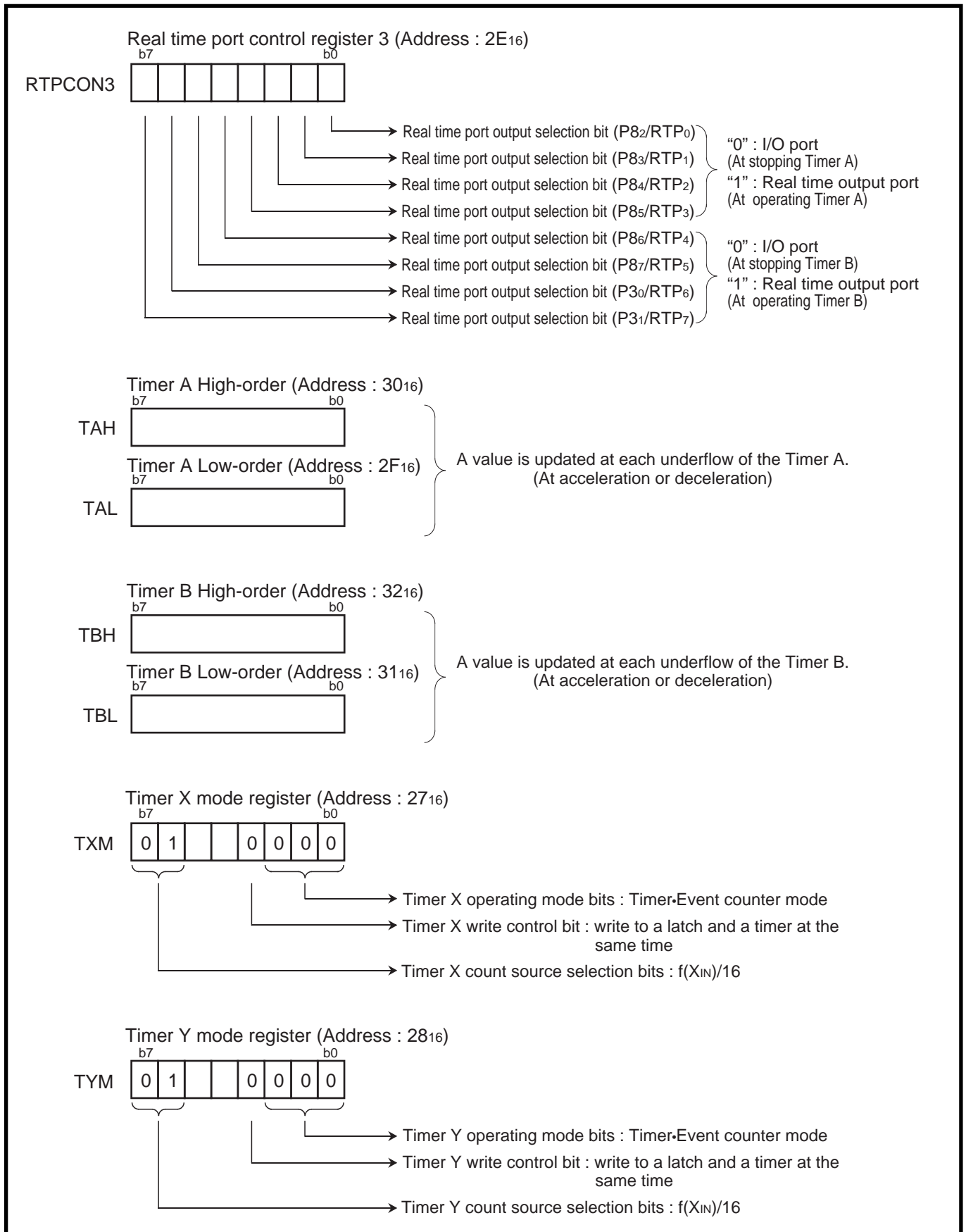


Fig. 2.4.16 Setting of related registers (2)

# APPLICATION

## 2.4 Real time output port

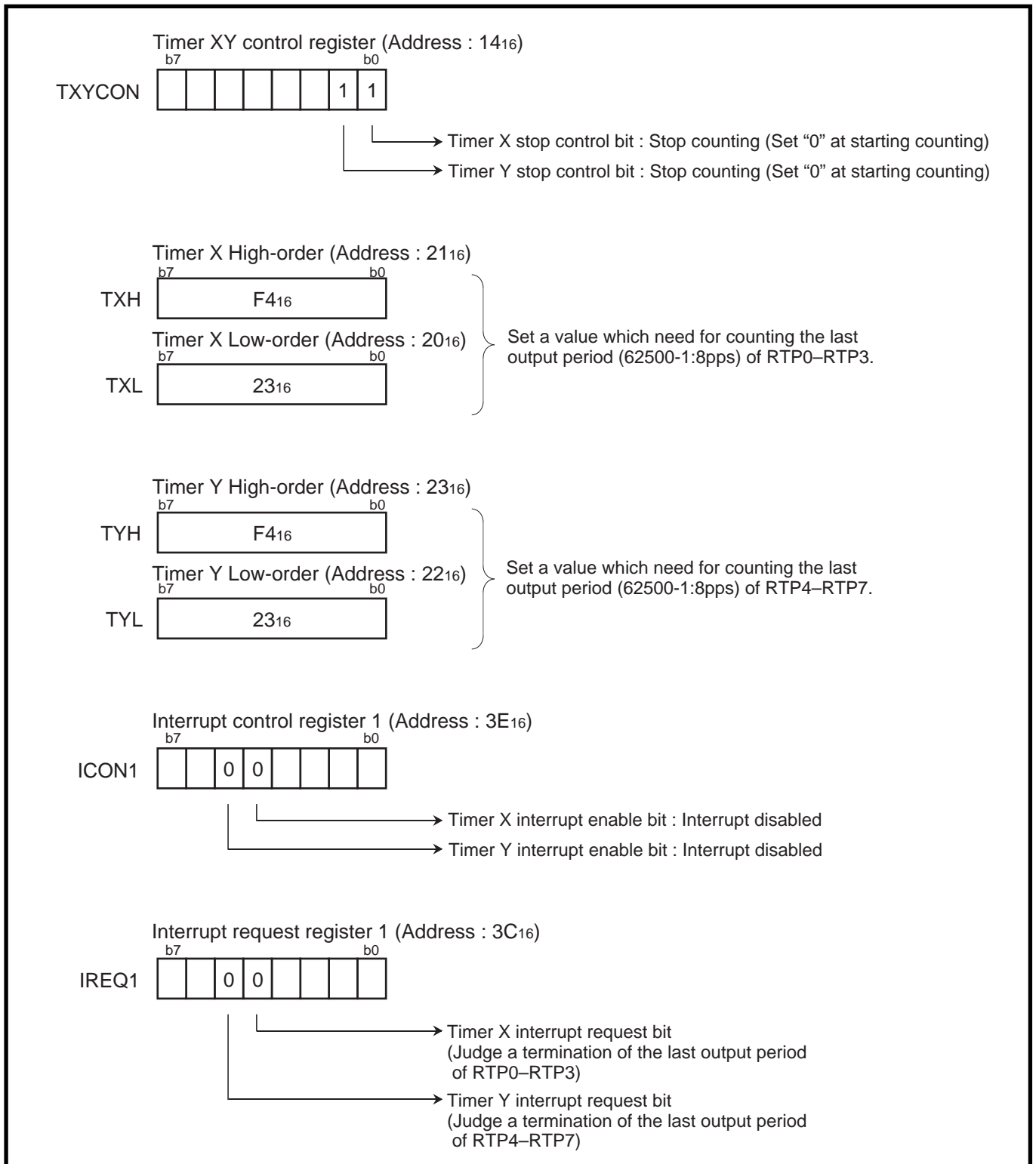


Fig. 2.4.17 Setting of related registers (3)

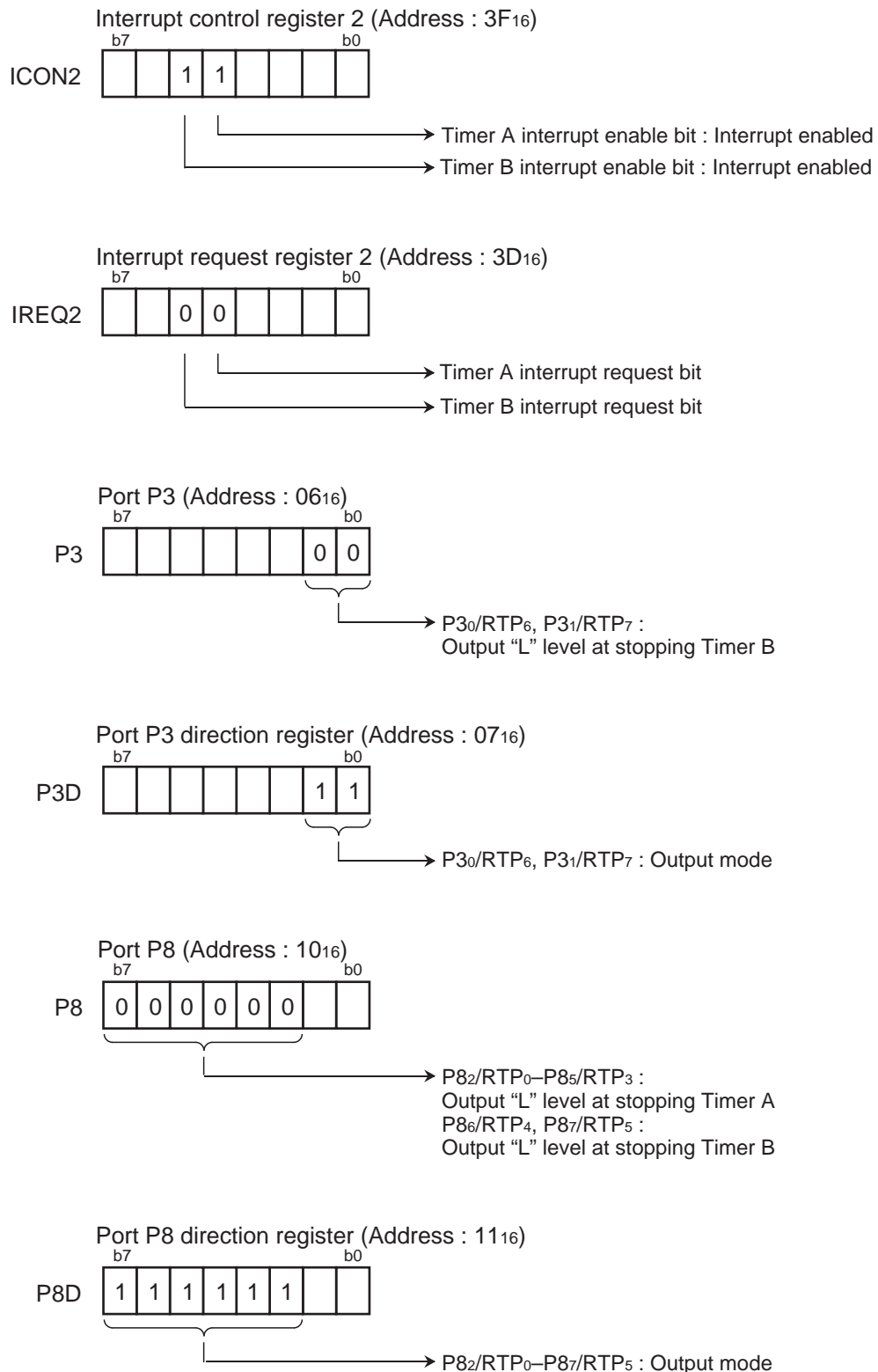


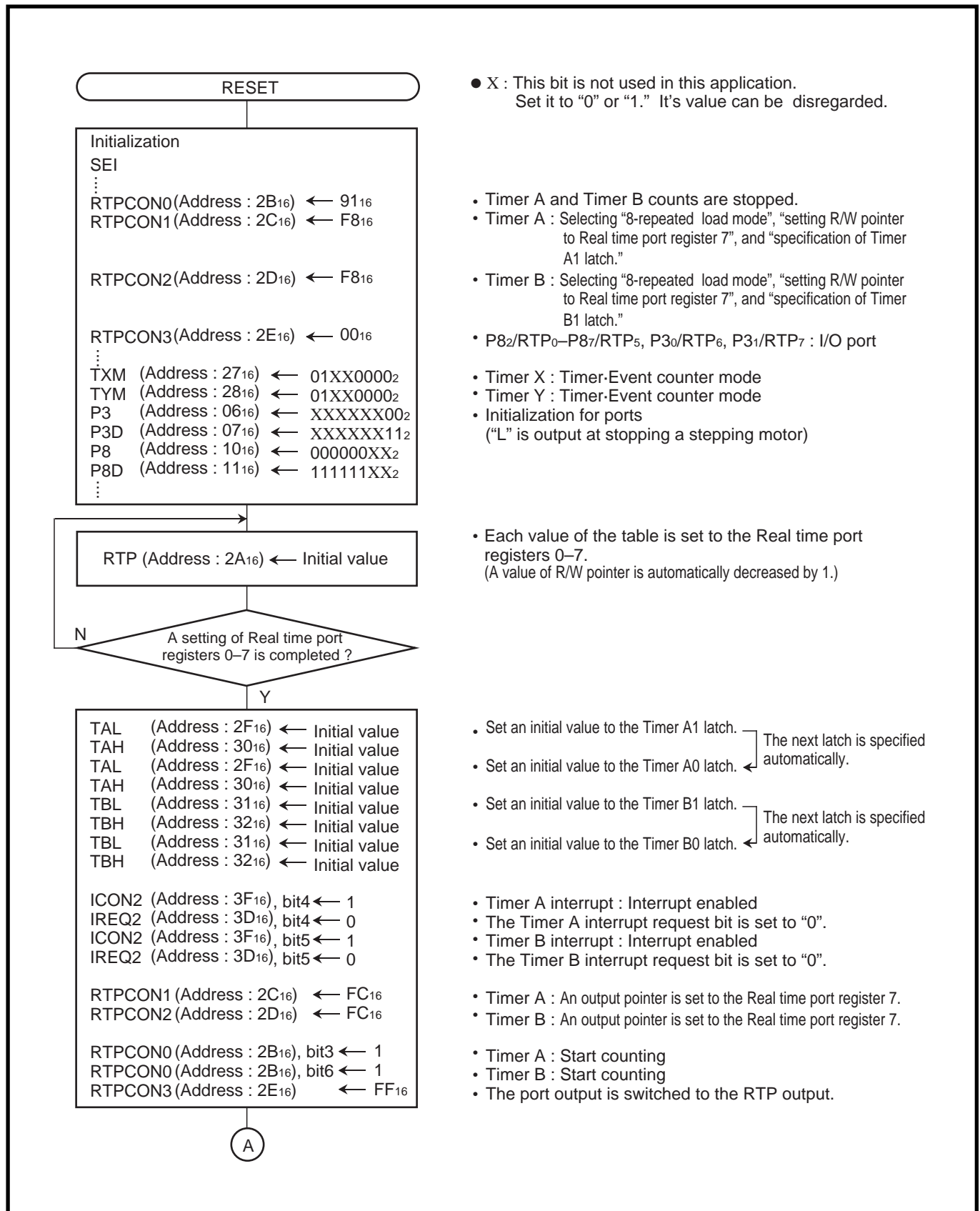
Fig. 2.4.18 Setting of related registers (4)

# APPLICATION

## 2.4 Real time output port

### Control procedure :

Figure 2.4.19–Figure 2.4.22 show control procedures.



**Fig. 2.4.19 Control procedure (1)**

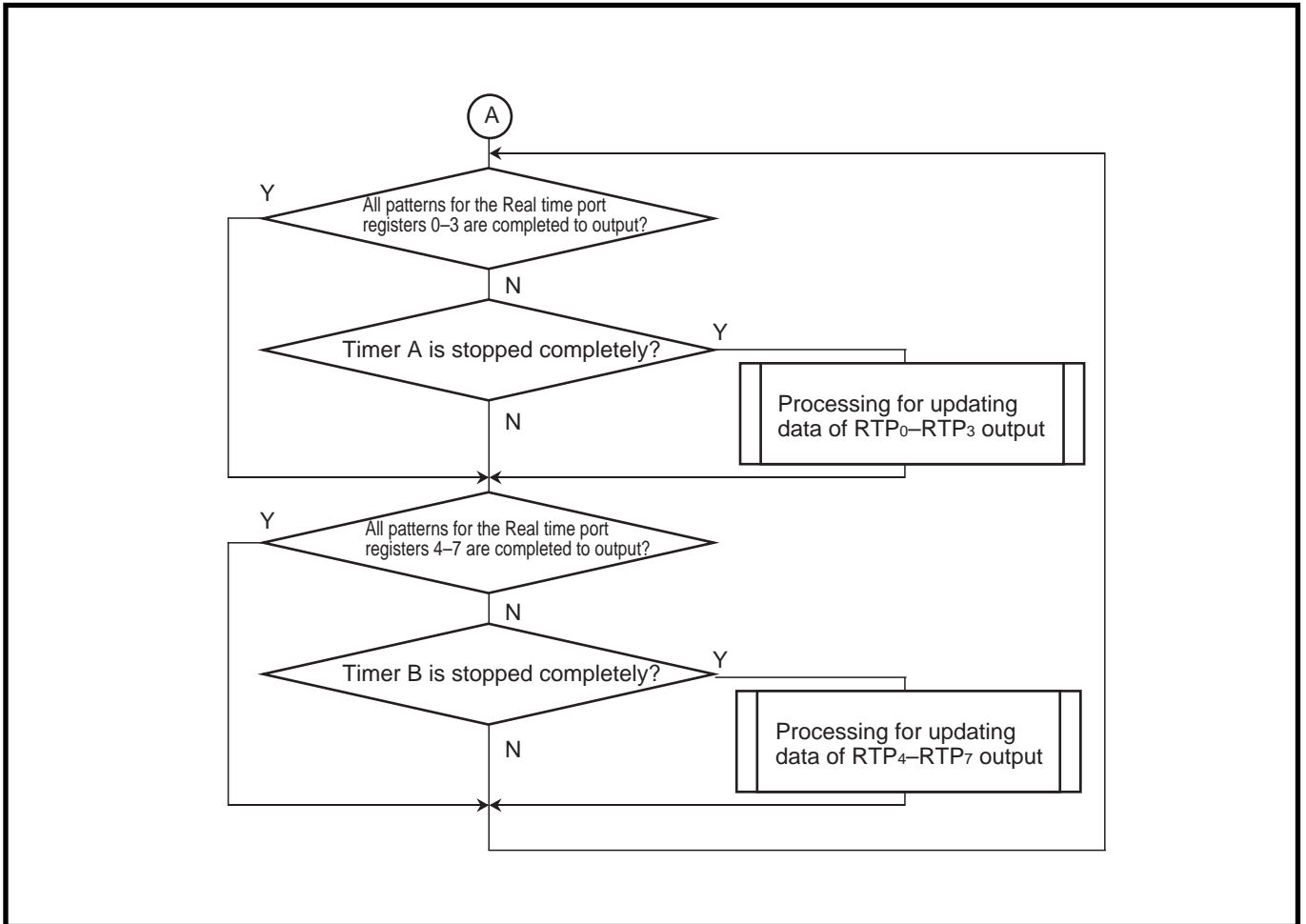
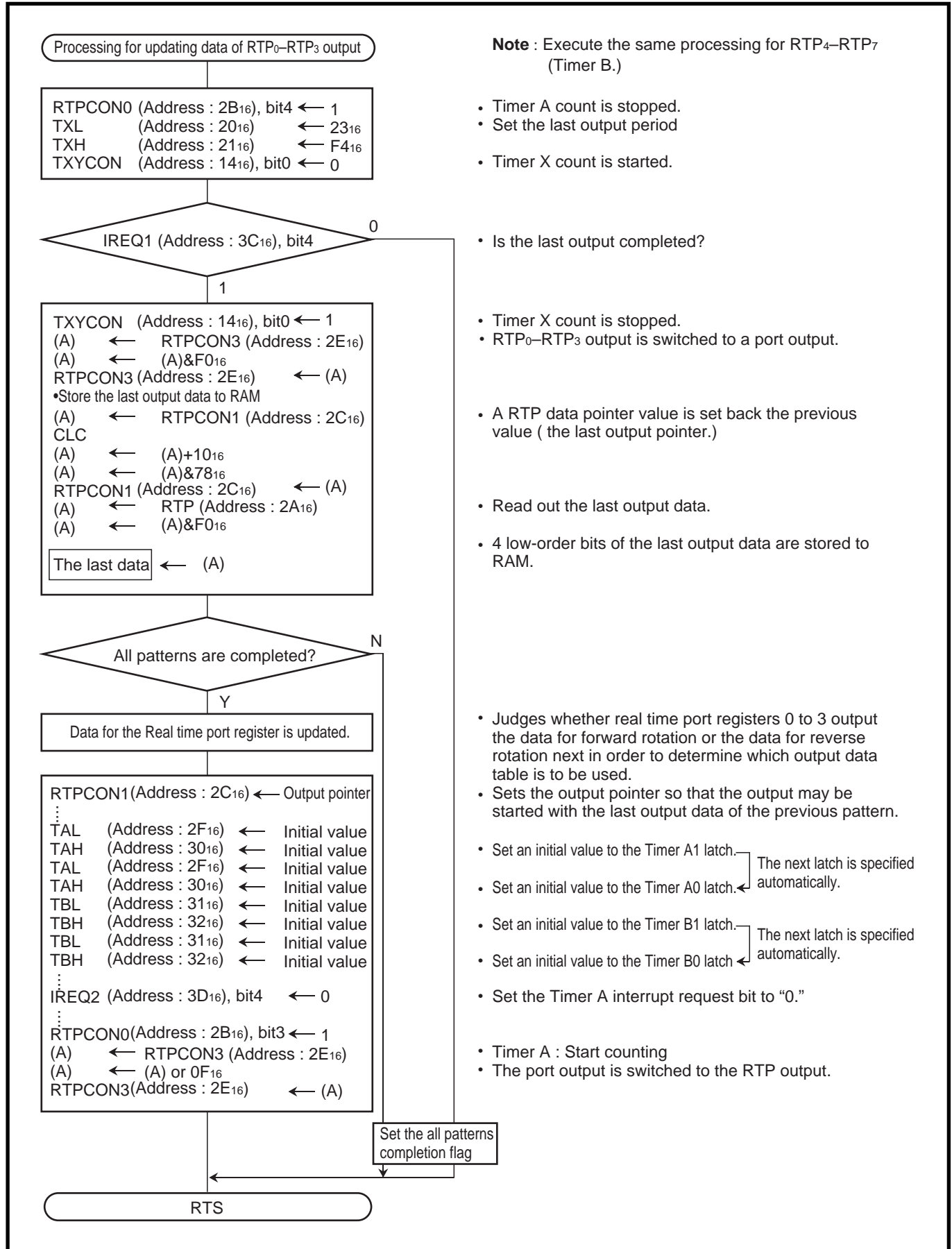


Fig. 2.4.20 Control procedure (2)



# APPLICATION

## 2.4 Real time output port



**Note :** Execute the same processing for RTP4-RTP7 (Timer B.)

- Timer A count is stopped.
- Set the last output period
- Timer X count is started.
- Is the last output completed?
- Timer X count is stopped.
- RTP0-RTP3 output is switched to a port output.
- A RTP data pointer value is set back the previous value ( the last output pointer.)
- Read out the last output data.
- 4 low-order bits of the last output data are stored to RAM.
- Judges whether real time port registers 0 to 3 output the data for forward rotation or the data for reverse rotation next in order to determine which output data table is to be used.
- Sets the output pointer so that the output may be started with the last output data of the previous pattern.
- Set an initial value to the Timer A1 latch. } The next latch is specified automatically.
- Set an initial value to the Timer A0 latch. }
- Set an initial value to the Timer B1 latch. } The next latch is specified automatically.
- Set an initial value to the Timer B0 latch. }
- Set the Timer A interrupt request bit to "0."
- Timer A : Start counting
- The port output is switched to the RTP output.

Fig. 2.4.21 Control procedure (3)

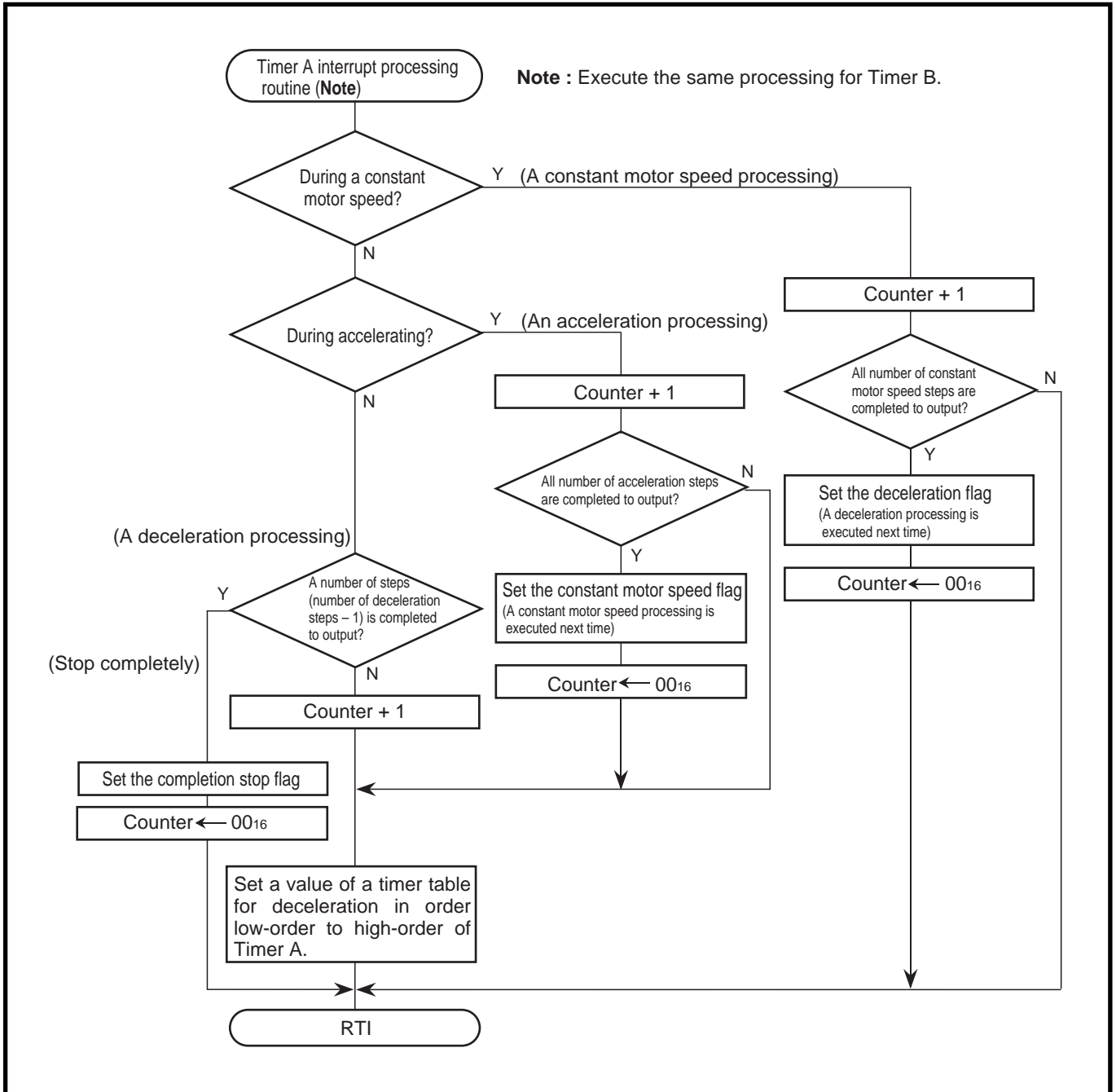


Fig. 2.4.22 Control procedure (4)

# APPLICATION

## 2.5 A-D converter

---

### 2.5 A-D converter

#### 2.5.1 Memory map of A-D conversion

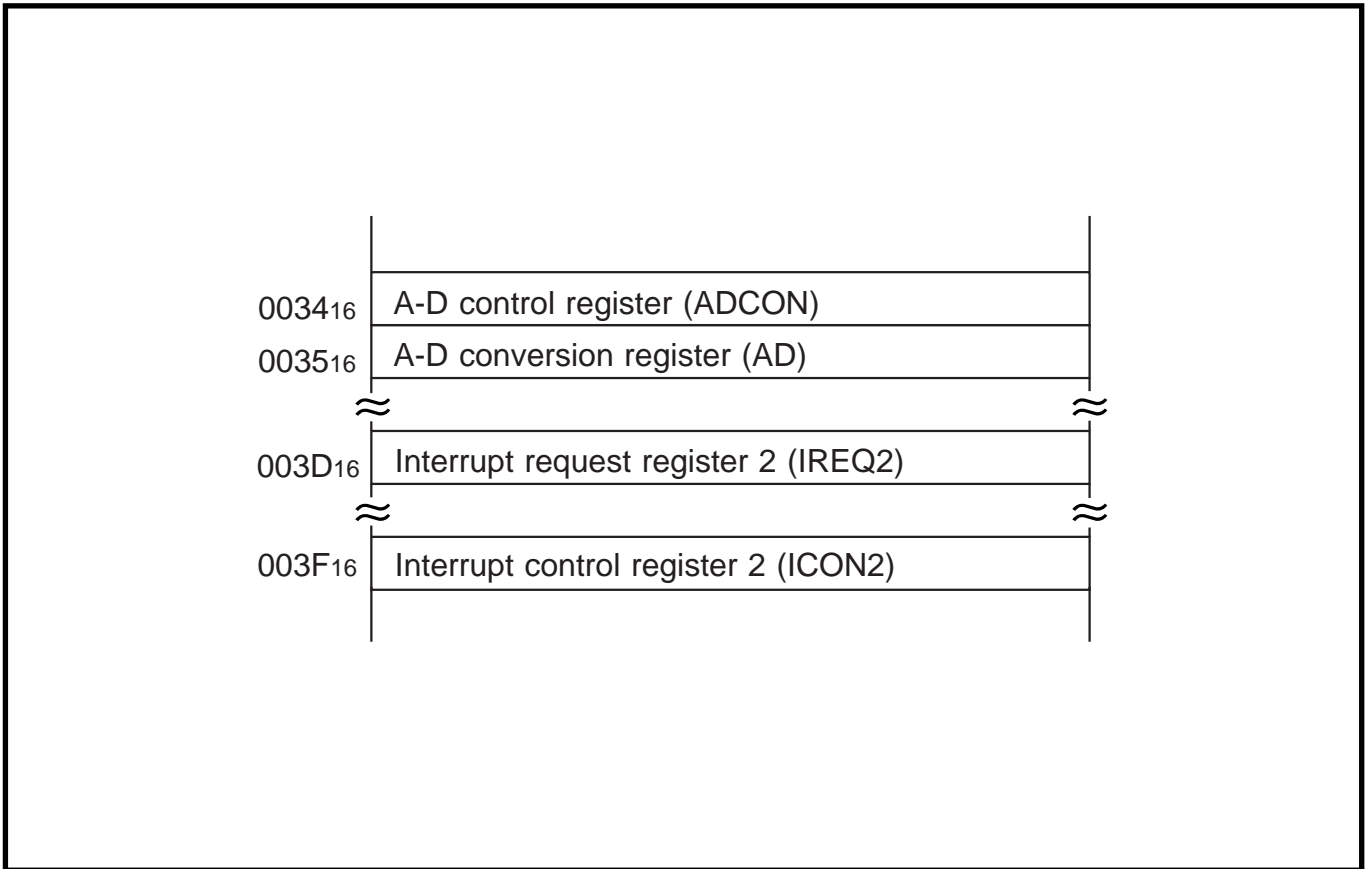


Fig. 2.5.1 Memory map of A-D conversion related registers

### 2.5.2 Related registers

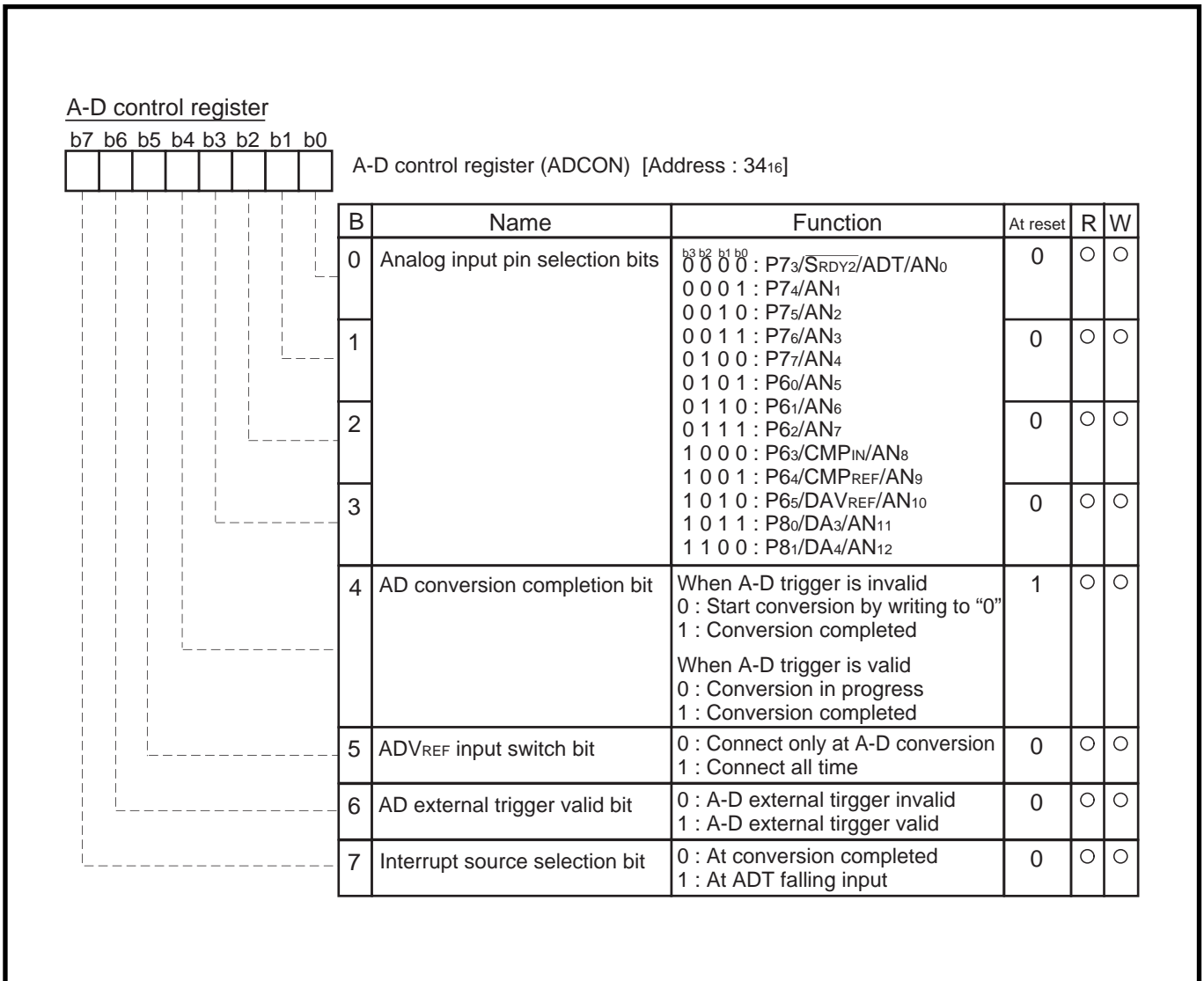


Fig. 2.5.2 Structure of A-D control register

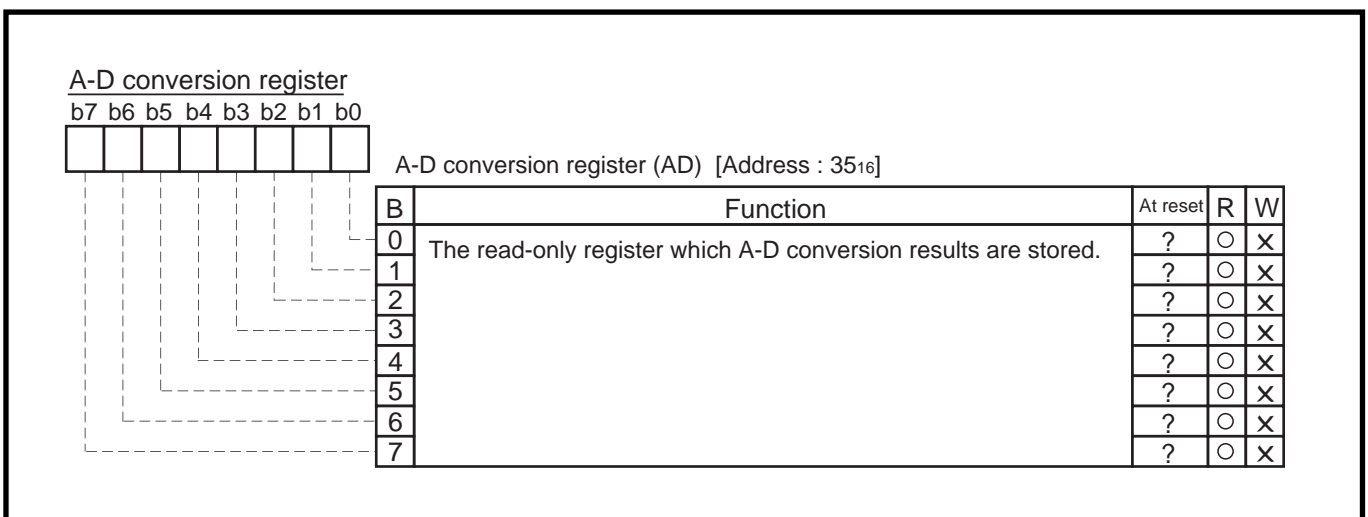


Fig. 2.5.3 Structure of A-D conversion register

# APPLICATION

## 2.5 A-D converter

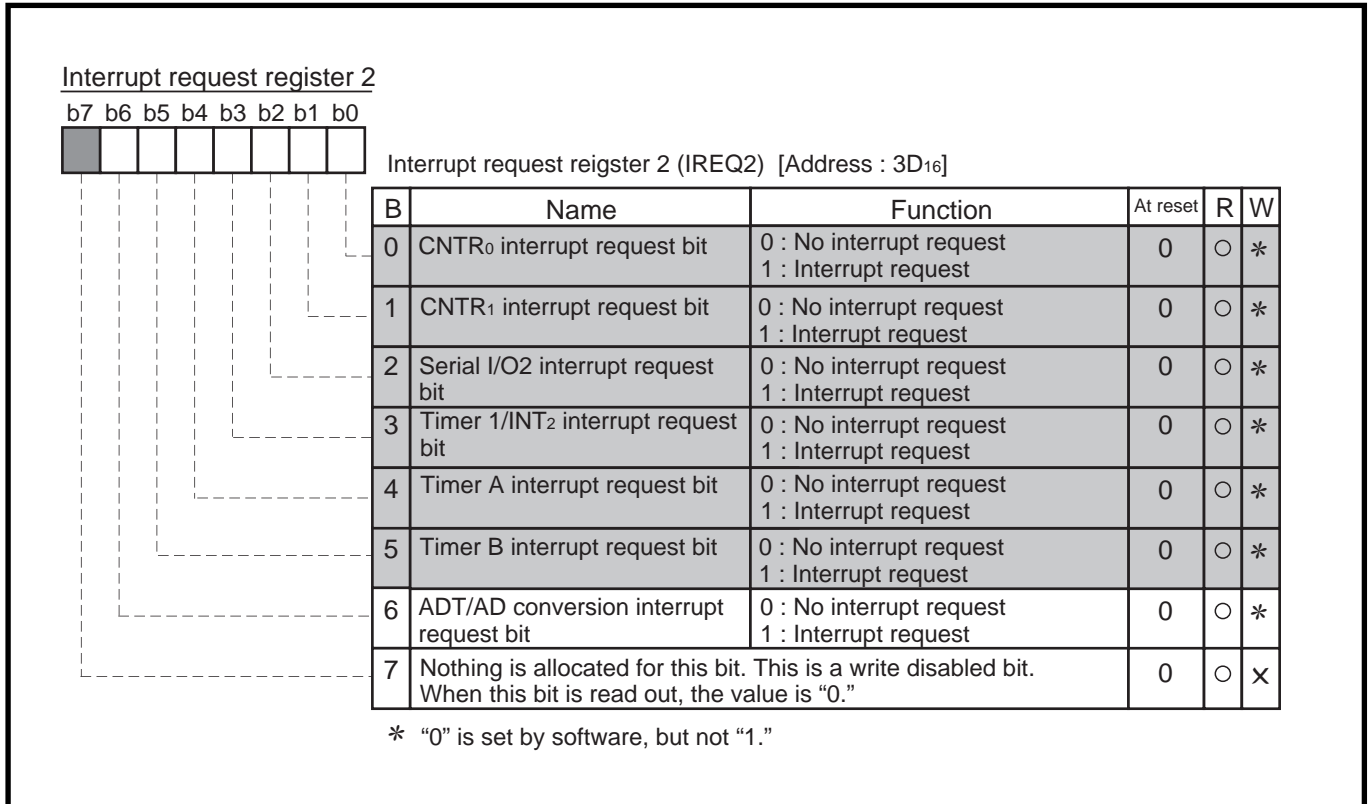


Fig. 2.5.4 Structure of Interrupt request register 2

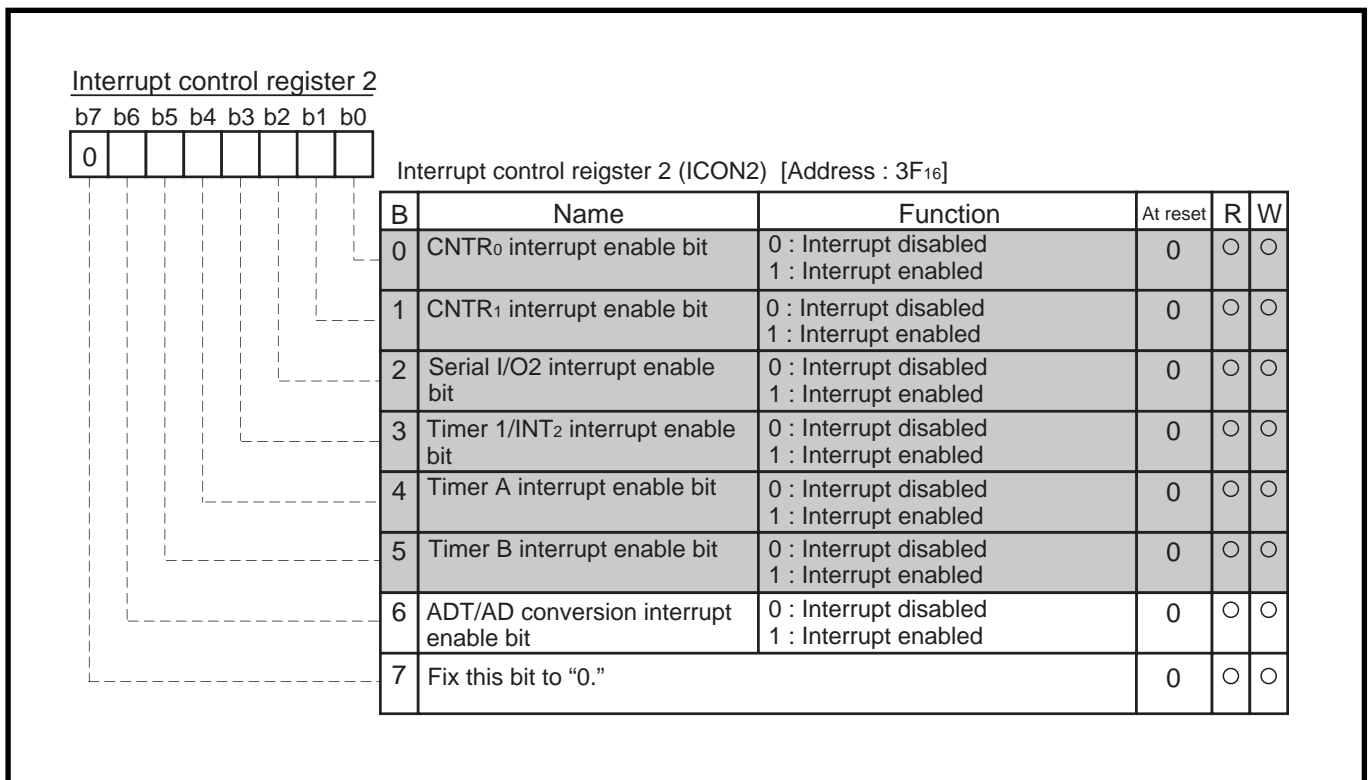


Fig. 2.5.5 Structure of Interrupt control register 2

### 2.5.3 A-D conversion application example

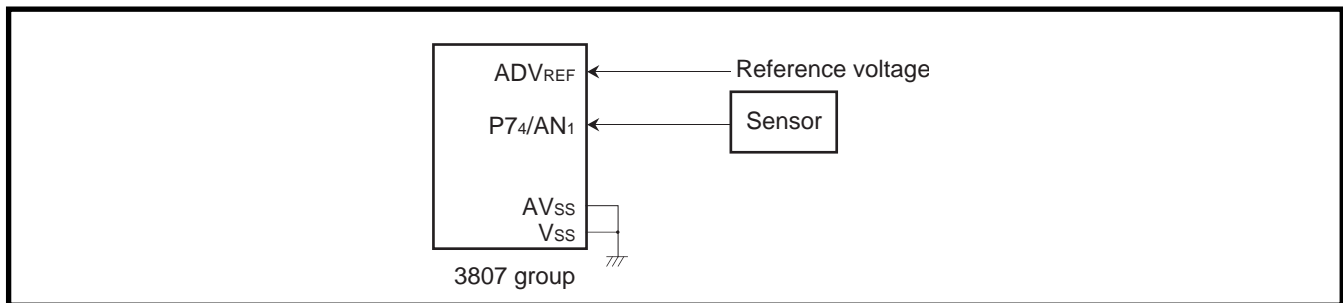
#### Conversion of Analog input voltage

**Outline :** The analog input voltage input from the sensor is converted into digital values.

Refer to the following examples for using an internal trigger or an external trigger.

#### (1) Read for analog signal using an internal trigger

Figure 2.5.6 shows a connection diagram, and Figure 2.5.7 shows a setting of related registers.



**Fig. 2.5.6 Connection diagram [Read for analog signal using an internal trigger]**

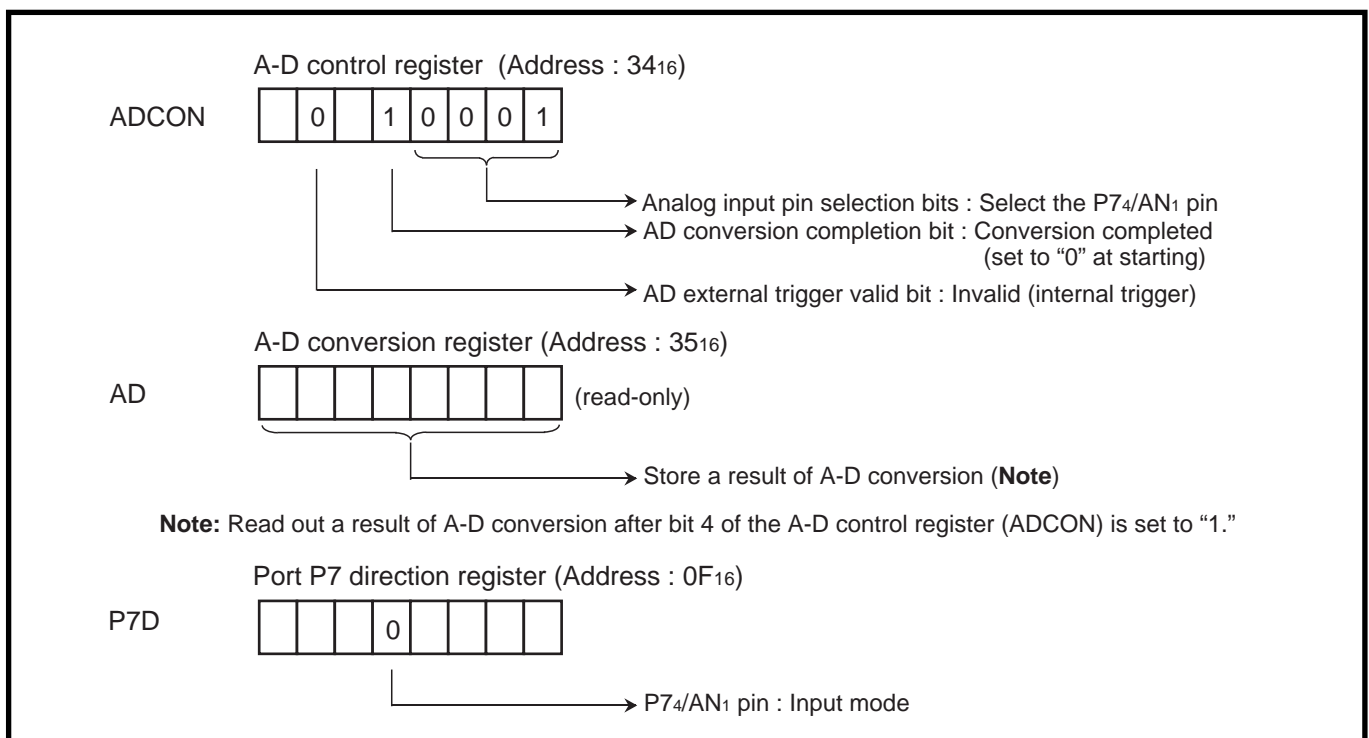
**Specifications :** • The analog input voltage input from the sensor is converted into digital values.  
**(Note)**

- The P74/AN1 pin is used as an analog input pin.
- A-D conversion start with an internal trigger (by setting bit 4 of A-D control register to "0" ).

**Note :** Example

When a reference voltage, 5.12 V is input to the ADVREF pin and a voltage, 4 V to the P74/AN1 pin, an input voltage is converted to a following value.

$$(256 / 5.12 \text{ V}) \times 4 \text{ V} = 200 \text{ (C816)}$$



**Fig. 2.5.7 Setting of related registers [Read for analog signal using an internal trigger]**

# APPLICATION

## 2.5 A-D converter

**Control procedure :** By setting the related registers as shown in Figure 2.5.8, the analog input voltage input from the sensor are converted into digital values.

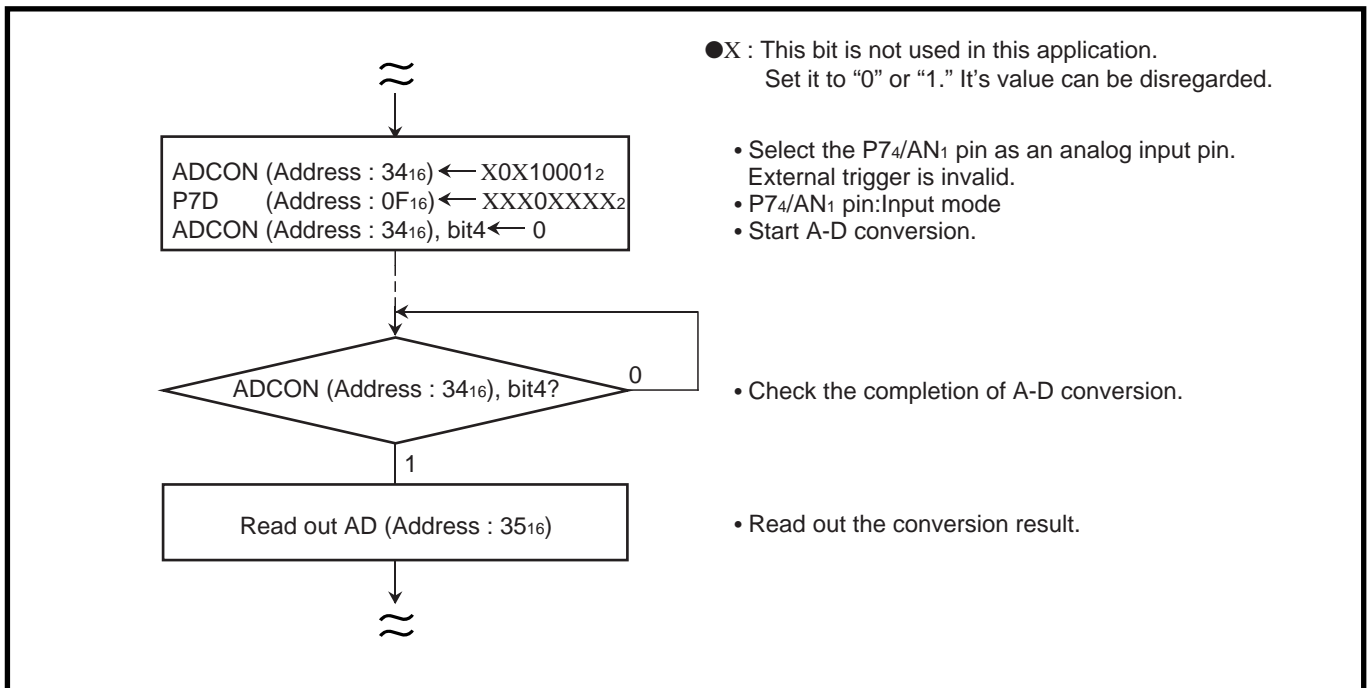
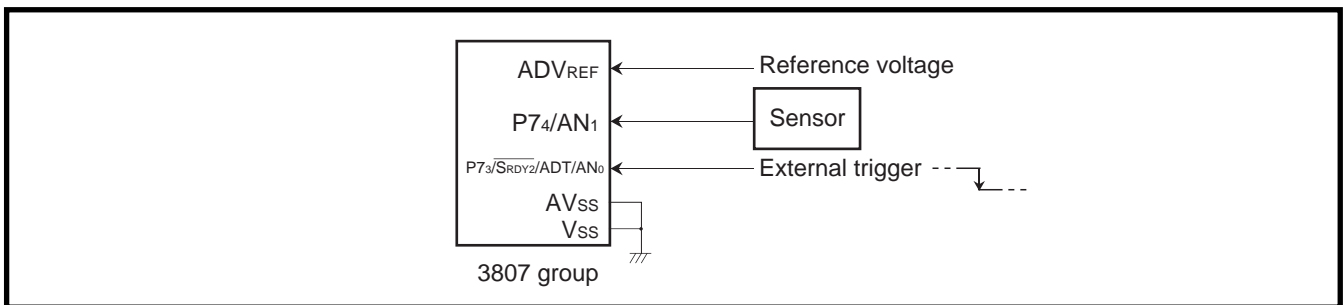


Fig. 2.5.8 Control procedure [Read for analog signal using an internal trigger]

### (2) Read for analog signal using an external trigger

Figure 2.5.9 shows a connection diagram, and Figure 2.5.10 shows a setting of related registers.



**Fig. 2.5.9 Connection diagram [Read for analog signal using an external trigger]**

**Specifications :**

- The analog input voltage input from the sensor is converted into digital values.

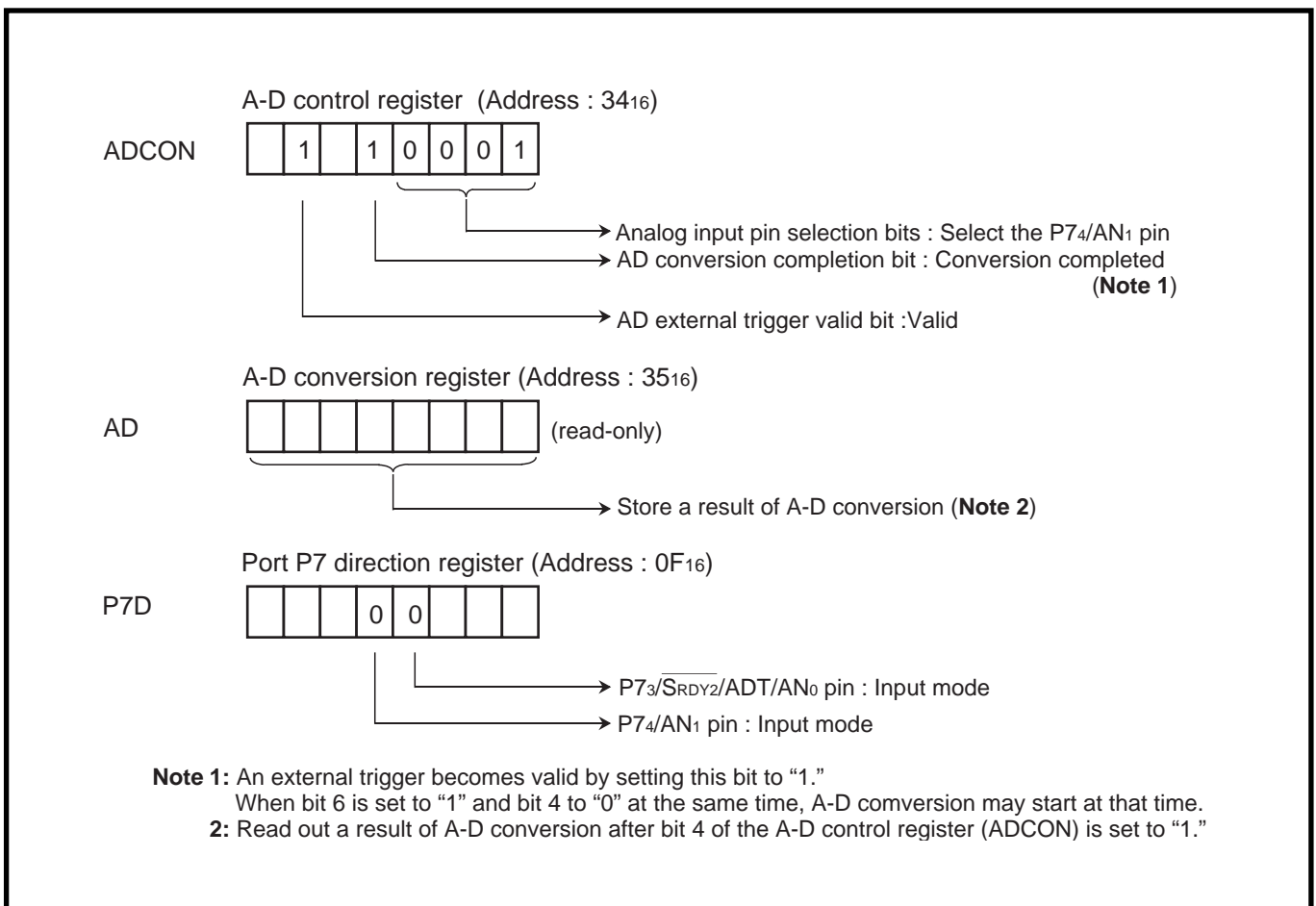
**(Note)**

- The P74/AN1 pin is used as an analog input pin.
- A-D conversion start with an external trigger (by inputting a falling edge to the P73/SRDY2/ADT/AN0 pin ).

**Note :** Example

When a reference voltage, 5.12 V is input to the ADVREF pin and a voltage, 4 V to the P74/AN1 pin, an input voltage is converted to a following value.

$$(256 / 5.12 \text{ V}) \times 4 \text{ V} = 200 \text{ (C816)}$$



**Fig. 2.5.10 Setting of related registers [Read for analog signal using an external trigger]**



# APPLICATION

## 2.5 A-D converter

**Control procedure :** By setting the related registers as shown in Figure 2.5.11, the analog input voltage input from the sensor are converted into digital values.

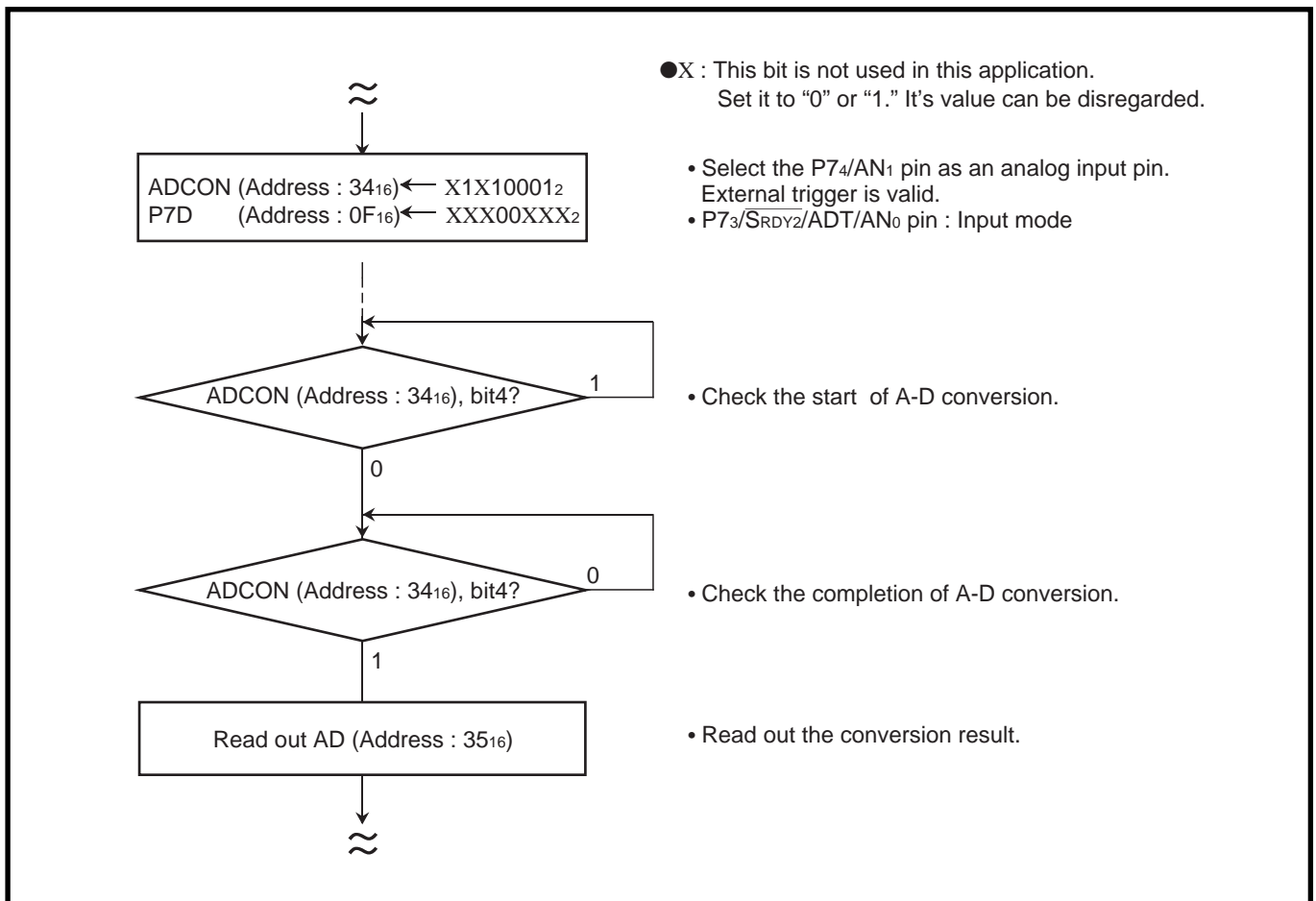


Fig. 2.5.11 Control procedure [Read for analog signal using an external trigger]

2.6 Reset

2.6.1 Connection example of reset IC

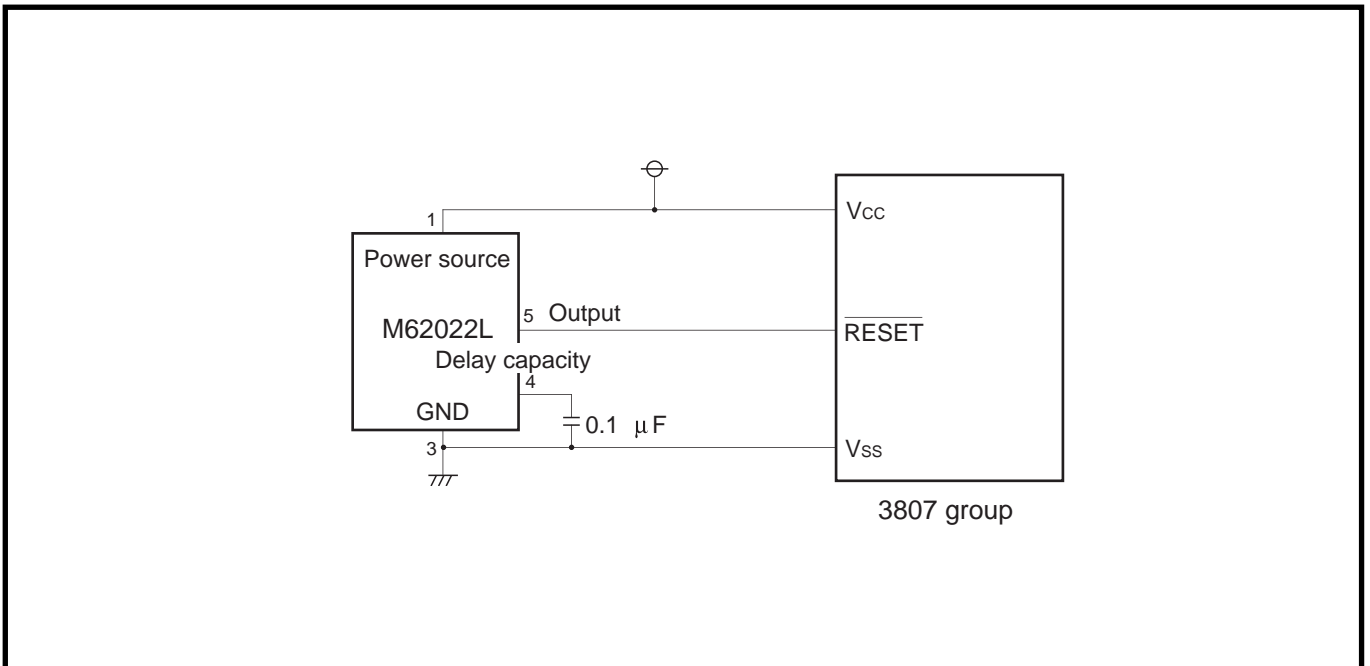


Fig. 2.6.1 Example of Poweron reset circuit

Figure 2.6.2 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.

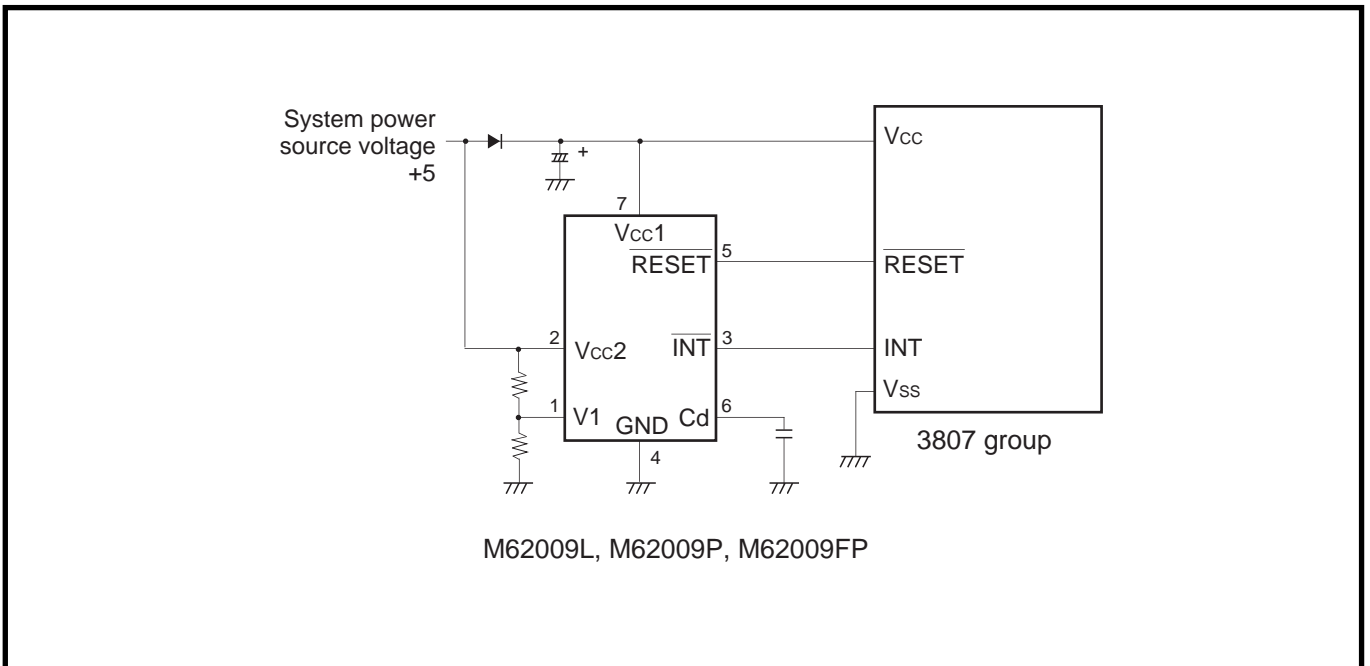


Fig. 2.6.2 RAM back-up system

# APPLICATION

## 2.7 Application circuit example

---

### 2.7 Application circuit example

Refer to the following applicaion circuit examples using the 3807 group microcomputer.

- Hot water supply system application example ..... Figure 2.7.1
- CD changer (car audio) application example ..... Figure 2.7.2
- Hot water washing toilet seat applicaiton example ..... Figure 2.7.3

### Hot water supply system application example

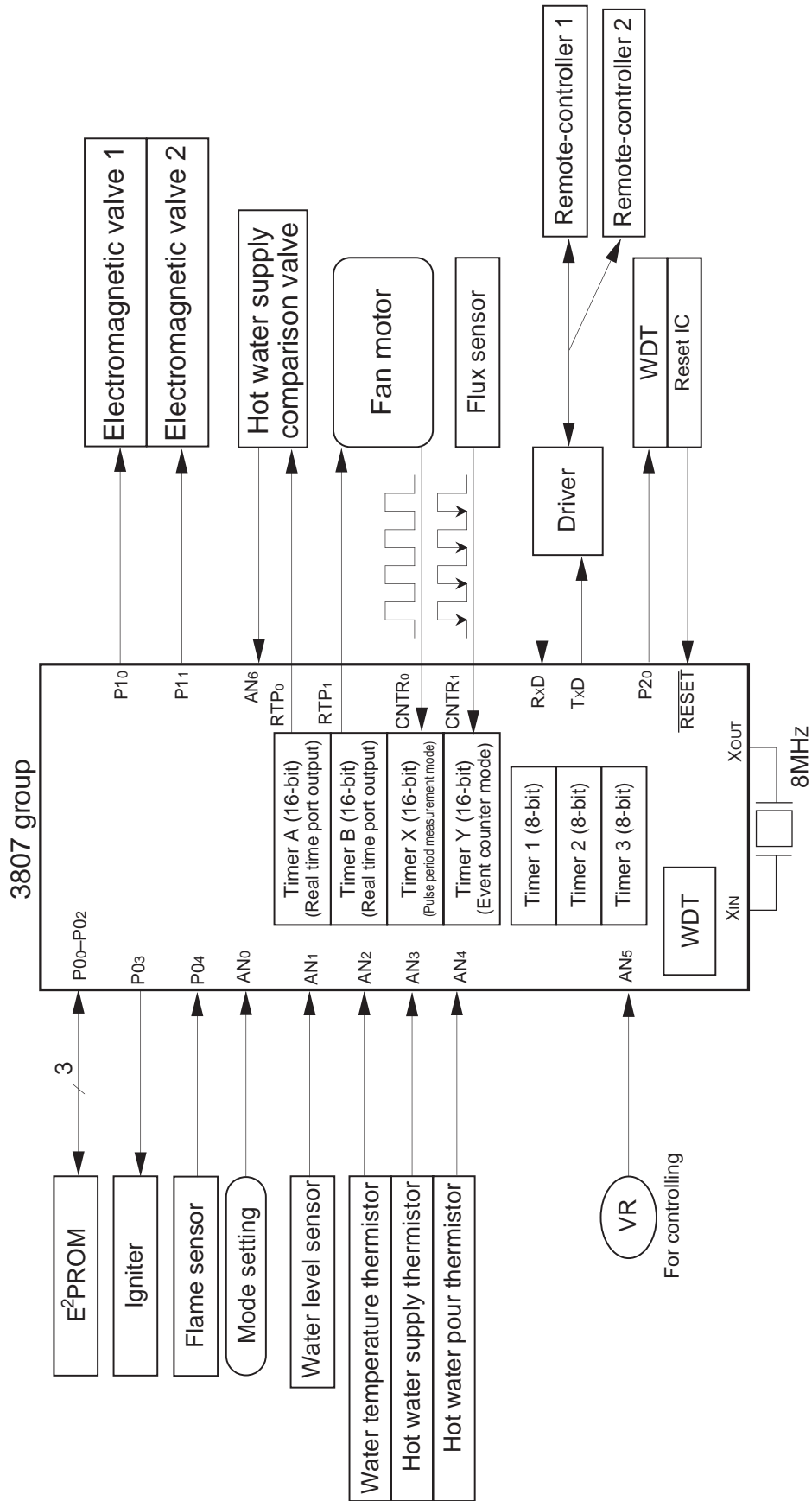


Fig. 2.7.1 Hot water supply system application example

# APPLICATION

## 2.7 Application circuit example

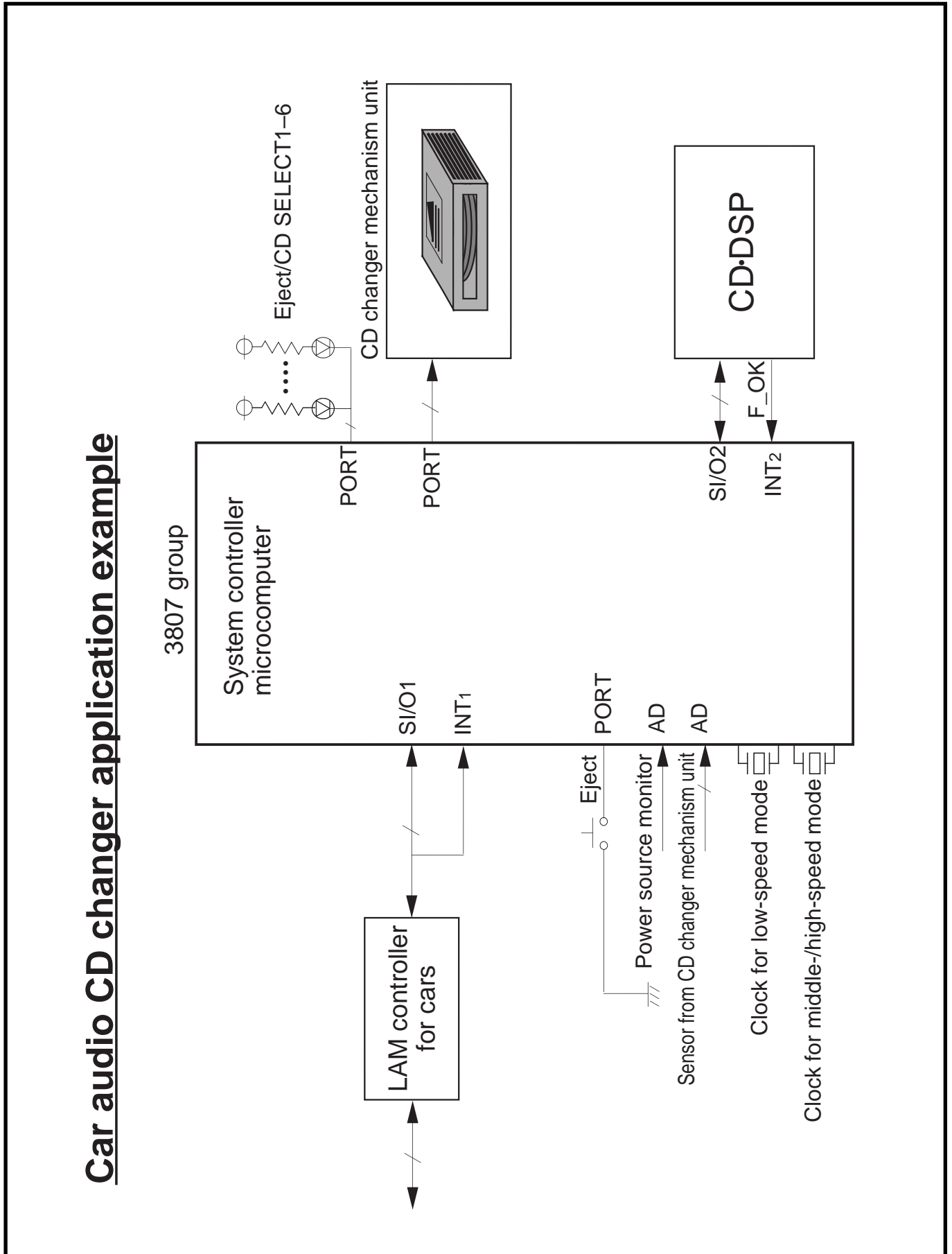


Fig. 2.7.2 CD changer (car audio) application example

### Warm water washing toilet seat application example

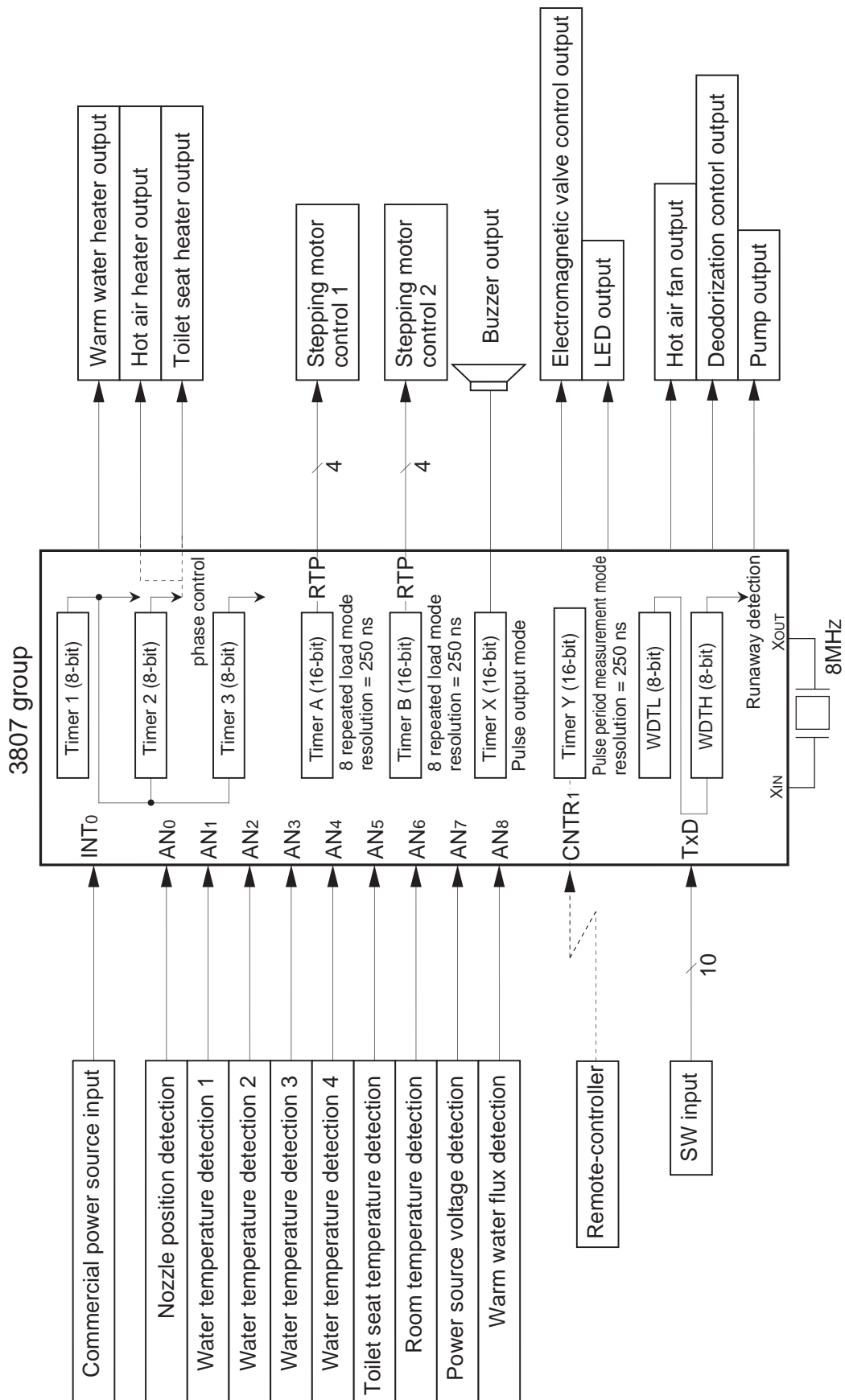


Fig. 2.7.3 Hot water washing toilet seat application example



# 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 Machine instructions
- 3.10 List of instruction codes
- 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
CMPV <sub>CC</sub>	Analog comparator power source voltage		-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-P65, P70-P77, P80-P87, ADVREF		-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> (ROM version)		-0.3 to 7	V
V <sub>I</sub>	Input voltage CNV <sub>SS</sub> (PROM version)		-0.3 to 13	V
V <sub>I</sub>	In-phase input voltage CMP <sub>IN</sub> , CMPREF		-0.3 to CMPV <sub>CC</sub> +0.3	V
V <sub>ID</sub>	Differential input voltage  CMP <sub>IN</sub> -CMPREF		CMPV <sub>CC</sub>	V
V <sub>O</sub>	Output voltage P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P62, P65, P70-P77, P80-P87, X <sub>OUT</sub>		-0.3 to V <sub>CC</sub> +0.3	V
V <sub>O</sub>	Output voltage CMP <sub>OUT</sub>		-0.3 to CMPV <sub>CC</sub> +0.3	V
P <sub>d</sub>	Power dissipation	T <sub>a</sub> = 25°C	500	mW
T <sub>opr</sub>	Operating temperature		-20 to 85	°C
T <sub>stg</sub>	Storage temperature		-40 to 125	°C



## 3.1 Electrical characteristics

## 3.1.2 Recommended operating conditions

**Table 3.1.2 Recommended operating conditions (1)**  
(V<sub>CC</sub> = 2.7 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	f(X <sub>IN</sub> ) ≤ 4.1MHz	2.7	5.0	5.5	V
		f(X <sub>IN</sub> ) = 8MHz	4.0	5.0	5.5	V
V <sub>SS</sub>	Power source voltage			0		V
ADVREF	A-D comparator reference voltage		2.0		V <sub>CC</sub>	V
DAVREF	D-A comparator reference voltage		2.7		V <sub>CC</sub>	V
CMPVCC	Analog comparator power source voltage			V <sub>CC</sub>		V
AVSS	Analog power source voltage			0		V
V <sub>IA</sub>	A-D comparator input voltage	AN <sub>0</sub> —AN <sub>12</sub>	AV <sub>SS</sub>		V <sub>CC</sub>	V
V <sub>IH</sub>	"H" input voltage	P <sub>00</sub> —P <sub>07</sub> , P <sub>10</sub> —P <sub>17</sub> , P <sub>30</sub> , P <sub>31</sub> , P <sub>33</sub> —P <sub>37</sub> , P <sub>40</sub> —P <sub>47</sub> , P <sub>50</sub> —P <sub>57</sub> , P <sub>60</sub> —P <sub>65</sub> , P <sub>70</sub> —P <sub>77</sub> , P <sub>80</sub> —P <sub>87</sub>	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IH</sub>	"H" input voltage (CMOS input level selected)	P <sub>20</sub> —P <sub>27</sub> , P <sub>32</sub>	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IH</sub>	"H" input voltage (TTL input level selected)	P <sub>20</sub> —P <sub>27</sub> , P <sub>32</sub> (Note)	2.0		V <sub>CC</sub>	V
V <sub>IH</sub>	"H" input voltage	RESET, X <sub>IN</sub> , CNV <sub>SS</sub>	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage	P <sub>00</sub> —P <sub>07</sub> , P <sub>10</sub> —P <sub>17</sub> , P <sub>30</sub> , P <sub>31</sub> , P <sub>33</sub> —P <sub>37</sub> , P <sub>40</sub> —P <sub>47</sub> , P <sub>50</sub> —P <sub>57</sub> , P <sub>60</sub> —P <sub>65</sub> , P <sub>70</sub> —P <sub>77</sub> , P <sub>80</sub> —P <sub>87</sub>	0		0.2V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage (CMOS input level selected)	P <sub>20</sub> —P <sub>27</sub> , P <sub>32</sub>	0		0.2V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage (TTL input level selected)	P <sub>20</sub> —P <sub>27</sub> , P <sub>32</sub> (Note)	0		0.8	V
V <sub>IL</sub>	"L" input voltage	RESET, CNV <sub>SS</sub>	0		0.2V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage	X <sub>IN</sub>	0		0.16V <sub>CC</sub>	V

Note: When V<sub>CC</sub> is 4.0 to 5.5 V.

**Table 3.1.3 Recommended operating conditions (2)**  
(V<sub>CC</sub> = 2.7 to 5.5 V, T<sub>a</sub> = – 20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
ΣI <sub>OH</sub> (peak)	"H" total peak output current (Note)	P <sub>00</sub> —P <sub>07</sub> , P <sub>10</sub> —P <sub>17</sub> , P <sub>20</sub> —P <sub>27</sub> , P <sub>30</sub> —P <sub>37</sub> , P <sub>80</sub> —P <sub>87</sub>			–80	mA
ΣI <sub>OH</sub> (peak)	"H" total peak output current (Note)	P <sub>40</sub> —P <sub>47</sub> , P <sub>50</sub> —P <sub>57</sub> , P <sub>60</sub> —P <sub>62</sub> , P <sub>65</sub> , CMP <sub>OUT</sub> , P <sub>70</sub> —P <sub>77</sub>			–80	mA
ΣI <sub>OL</sub> (peak)	"L" total peak output current (Note)	P <sub>00</sub> —P <sub>07</sub> , P <sub>10</sub> —P <sub>17</sub> , P <sub>20</sub> —P <sub>23</sub> , P <sub>30</sub> —P <sub>37</sub> , P <sub>80</sub> —P <sub>87</sub>			80	mA
ΣI <sub>OL</sub> (peak)	"L" total peak output current (Note) P <sub>24</sub> —P <sub>27</sub>	in single chip mode			80	mA
		in memory expansion mode and microprocessor mode			80	mA
ΣI <sub>OL</sub> (peak)	"L" total peak output current (Note)	P <sub>40</sub> —P <sub>47</sub> , P <sub>50</sub> —P <sub>57</sub> , P <sub>60</sub> —P <sub>62</sub> , P <sub>65</sub> , CMP <sub>OUT</sub> , P <sub>70</sub> —P <sub>77</sub>			80	mA
ΣI <sub>OH</sub> (avg)	"H" total average output current (Note)	P <sub>00</sub> —P <sub>07</sub> , P <sub>10</sub> —P <sub>17</sub> , P <sub>20</sub> —P <sub>27</sub> , P <sub>30</sub> —P <sub>37</sub> , P <sub>80</sub> —P <sub>87</sub>			–40	mA
ΣI <sub>OH</sub> (avg)	"H" total average output current (Note)	P <sub>40</sub> —P <sub>47</sub> , P <sub>50</sub> —P <sub>57</sub> , P <sub>60</sub> —P <sub>62</sub> , P <sub>65</sub> , CMP <sub>OUT</sub> , P <sub>70</sub> —P <sub>77</sub>			–40	mA
ΣI <sub>OL</sub> (avg)	"L" total average output current (Note)	P <sub>00</sub> —P <sub>07</sub> , P <sub>10</sub> —P <sub>17</sub> , P <sub>20</sub> —P <sub>23</sub> , P <sub>30</sub> —P <sub>37</sub> , P <sub>80</sub> —P <sub>87</sub>			40	mA
ΣI <sub>OL</sub> (avg)	"L" total average output current (Note) P <sub>24</sub> —P <sub>27</sub>	in single chip mode			40	mA
		in memory expansion mode and microprocessor mode			40	mA
ΣI <sub>OL</sub> (avg)	"L" total average output current (Note)	P <sub>40</sub> —P <sub>47</sub> , P <sub>50</sub> —P <sub>57</sub> , P <sub>60</sub> —P <sub>62</sub> , P <sub>65</sub> , CMP <sub>OUT</sub> , P <sub>70</sub> —P <sub>77</sub>			40	mA

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

# APPENDIX

## 3.1 Electrical characteristics

**Table 3.1.4 Recommended operating conditions (3)**  
(V<sub>CC</sub> = 2.7 to 5.5 V, T<sub>a</sub> = - 20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit	
			Min.	Typ.	Max.		
I <sub>OH</sub> (peak)	"H" peak output current (Note 1)	P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P62, P65, CMP <sub>OUT</sub> , P70–P77, P80–P87			-10	mA	
I <sub>OL</sub> (peak)	"L" peak output current (Note 1)	P00–P07, P10–P17, P20–P23, P30–P37, P40–P47, P50–P57, P60–P62, P65, CMP <sub>OUT</sub> , P70–P77, P80–P87			10	mA	
I <sub>OL</sub> (peak)	"L" peak output current (Note 1) P24–P27	in single chip mode			20	mA	
		in memory expansion mode and microprocessor mode			10	mA	
I <sub>OH</sub> (avg)	"H" average output current (Note 2)	P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P62, P65, CMP <sub>OUT</sub> , P70–P77, P80–P87			-5	mA	
I <sub>OL</sub> (avg)	"L" average output current (Note 2)	P00–P07, P10–P17, P20–P23, P30–P37, P40–P47, P50–P57, P60–P62, P65, CMP <sub>OUT</sub> , P70–P77, P80–P87			5	mA	
I <sub>OL</sub> (avg)	"L" average output current (Note 2) P24–P27	in single chip mode			15	mA	
		in memory expansion mode and microprocessor mode			5	mA	
f(X <sub>IN</sub> )	Main clock input oscillation frequency (Note 3)	High-speed mode 4.0V ≤ V <sub>CC</sub> ≤ 5.5V			8	MHz	
		High-speed mode 2.7V ≤ V <sub>CC</sub> ≤ 4.0V			3V <sub>CC</sub> -4	MHz	
		Middle-speed mode 4.0V ≤ V <sub>CC</sub> ≤ 5.5V			8	MHz	
		Middle-speed mode (Note 5) 2.7V ≤ V <sub>CC</sub> ≤ 4.0V			8	MHz	
		Middle-speed mode (Note 5) 2.7V ≤ V <sub>CC</sub> ≤ 4.0V			3V <sub>CC</sub> -4	MHz	
f(X <sub>CIN</sub> )	Sub-clock input oscillation frequency (Note 3, 4)				32.768	50	kHz

- Note 1:** The peak output current is the peak current flowing in each port.  
**2:** The average output current I<sub>OL</sub> (avg), I<sub>OH</sub> (avg) in an average value measured over 100ms.  
**3:** When the oscillation frequency has a duty cycle of 50%.  
**4:** When using the microcomputer in low-speed mode, set the sub-clock input oscillation frequency on condition that f(X<sub>CIN</sub>) < f(X<sub>IN</sub>)/3.  
**5:** When using the timer X/Y, timer A/B (real time output port), timer 1/2/3, serial I/O1, serial I/O2, and A-D converter, set the main clock input oscillation frequency to the max. 3 V<sub>CC</sub>-4 (MHz).

## 3.1 Electrical characteristics

### 3.1.3 Electrical characteristics

**Table 3.1.5 Electrical characteristics (1)**

(V<sub>CC</sub> = 2.7 to 5.5 V, V<sub>SS</sub> = 0 V, T<sub>a</sub> = - 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–P62, P65, P70–P77, P80–P87, CMP <sub>OUT</sub> (Note 1)	I <sub>OH</sub> = -10mA V <sub>CC</sub> = 4.0 to 5.5V	V <sub>CC</sub> -2.0			V
		I <sub>OH</sub> = -1.0mA V <sub>CC</sub> = 2.7 to 5.5V	V <sub>CC</sub> -1.0			V
VOL	“L” output voltage P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P62, P65, P70–P77, P80–P87, CMP <sub>OUT</sub>	I <sub>OL</sub> = 10mA V <sub>CC</sub> = 4.0 to 5.5V			2.0	V
		I <sub>OL</sub> = 1.6mA V <sub>CC</sub> = 2.7 to 5.5V			0.4	V
VT+–VT-	Hysteresis	P42, P43, P51–P55, P73 (Note 2), CNTR0, CNTR1, INT0–INT4, ADT		0.4		V
VT+–VT-	Hysteresis	RxD, SCLK1, SIN2, SCLK2		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–P65, P70–P77, P80–P87	V <sub>I</sub> = V <sub>CC</sub> (Pin floating. Pull-up transistors “off”)		5.0	μA
I <sub>IH</sub>	“H” input current	RESET, CNV <sub>SS</sub>	V <sub>I</sub> = V <sub>CC</sub>		5.0	μA
I <sub>IH</sub>	“H” input current	XIN	V <sub>I</sub> = V <sub>CC</sub>	4		μA
I <sub>IL</sub>	“L” input current	P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P65, P70–P77, P80–P87	V <sub>I</sub> = V <sub>SS</sub> (Pin floating. Pull-up transistors “off”)		-5.0	μA
I <sub>IL</sub>	“L” input current	RESET, CNV <sub>SS</sub>	V <sub>I</sub> = V <sub>SS</sub>		-5.0	μA
I <sub>IL</sub>	“L” input current	XIN	V <sub>I</sub> = V <sub>SS</sub>	-4		μA
I <sub>IL</sub>	“L” input current	P00–P07, P10–P17, P20–P27	Pull-up transistors “on” V <sub>I</sub> = V <sub>SS</sub>	-0.2		mA
V <sub>RAM</sub>	RAM hold voltage	When clock stopped	2.0		5.5	V

**Note 1:** P45 is measured when the P45/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B<sub>16</sub>) is “0”. P71, and P12 are measured when the P71/SOUT2 and P72/SCLK2 P-channel output disable bit of the serial I/O2 control register 1 (bit 7 of address 001D<sub>16</sub>).

**2:** P73 is measured when the AD external trigger valid bit of the A–D control register (bit 6 of address 0034<sub>16</sub>) is “1”.

# APPENDIX

## 3.1 Electrical characteristics

**Table 3.1.6 Electrical characteristics (2)**  
**(V<sub>CC</sub> = 2.7 to 5.5 V, V<sub>SS</sub> = 0 V, T<sub>a</sub> = - 20 to 85 °C, unless otherwise noted)**

Symbol	Parameter	Test conditions	Limits			Unit	
			Min.	Typ.	Max.		
I <sub>CC</sub>	Power source current	High-speed mode f(X <sub>IN</sub> ) = 8MHz f(X <sub>CIN</sub> ) = 32.768kHz Output transistors "off"		6.8	13	mA	
		High-speed mode f(X <sub>IN</sub> ) = 8MHz (in WIT state) f(X <sub>CIN</sub> ) = 32.768kHz Output transistors "off"		1.6		mA	
		Low-speed mode f(X <sub>IN</sub> ) = stopped f(X <sub>CIN</sub> ) = 32.768kHz Low-power dissipation mode (CM <sub>3</sub> = 0) Output transistors "off"		60	200	μA	
		Low-speed mode f(X <sub>IN</sub> ) = stopped f(X <sub>CIN</sub> ) = 32.768kHz (in WIT state) Low-power dissipation mode (CM <sub>3</sub> = 0) Output transistors "off"		20	40	μA	
		Low-speed mode (V <sub>CC</sub> = 3V) f(X <sub>IN</sub> ) = stopped f(X <sub>CIN</sub> ) = 32.768kHz Low-power dissipation mode (CM <sub>3</sub> = 0) Output transistors "off"		20	55	μA	
		Low-speed mode (V <sub>CC</sub> = 3V) f(X <sub>IN</sub> ) = stopped f(X <sub>CIN</sub> ) = 32.768kHz (in WIT state) Low-power dissipation mode (CM <sub>3</sub> = 0) Output transistors "off"		5.0	10.0	μA	
		Middle-speed mode f(X <sub>IN</sub> ) = 8MHz f(X <sub>CIN</sub> ) = stopped Output transistors "off"		4.0	7.0	mA	
		Middle-speed mode f(X <sub>IN</sub> ) = 8MHz (in WIT state) f(X <sub>CIN</sub> ) = stopped Output transistors "off"		1.5		mA	
		Increment when A-D conversion is executed f(X <sub>IN</sub> ) = 8MHz			800	μA	
		All oscillation stopped (in STP state) Output transistors "off"	T <sub>a</sub> = 25°C		0.1	1.0	μA
			T <sub>a</sub> = 85°C			10	μA
CMPI <sub>CC</sub>	Analog comparator Power source current			200	500	μA	

## 3.1 Electrical characteristics

### 3.1.4 A-D converter characteristics

**Table 3.1.7 A-D converter characteristics**

(V<sub>CC</sub> = 2.7 to 5.5 V, V<sub>SS</sub> = AV<sub>SS</sub> = 0 V, ADV<sub>REF</sub> = 2.0 V to V<sub>CC</sub>, T<sub>a</sub> = - 20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
—	Resolution					8	Bits
—	Absolute accuracy (excluding quantization error)		V <sub>CC</sub> = ADV <sub>REF</sub> = 5.0V			±2	LSB
t <sub>CONV</sub>	Conversion time					50	t <sub>c(φ)</sub>
RLADDER	Ladder resistor			12	35	100	kΩ
I <sub>ADVREF</sub>	Reference power source input current	ADV <sub>REF</sub> "on"	ADV <sub>REF</sub> = 5.0V	50	150	200	μA
		ADV <sub>REF</sub> "off"				5	μA
I <sub>I(AD)</sub>	A-D port input current					5.0	μA

### 3.1.5 D-A converter characteristics

**Table 3.1.8 D-A converter characteristics**

(V<sub>CC</sub> = 2.7 to 5.5 V, V<sub>SS</sub> = AV<sub>SS</sub> = 0 V, DAV<sub>REF</sub> = 2.7 V to V<sub>CC</sub>, T<sub>a</sub> = - 20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
—	Resolution					8	Bits
—	Absolute accuracy	V <sub>CC</sub> = 4.0 to 5.5V				1.0	%
		V <sub>CC</sub> = 2.7 to 4.0V				2.5	%
t <sub>su</sub>	Setting time					3	μs
R <sub>o</sub>	Output resistor			1	2.5	4	kΩ
I <sub>DAVREF</sub>	Reference power source input current (Note)					3.2	mA

**Note:** Using one D-A converter, with the value in the D-A conversion register of the other D-A converter being "0016".

### 3.1.6 Analog comparator characteristics

**Table 3.1.9 Analog comparator characteristics**

(V<sub>CC</sub> = 2.7 to 5.5 V, V<sub>SS</sub> = AV<sub>SS</sub> = 0 V, CMPV<sub>CC</sub> = 2.7 V to V<sub>CC</sub>, T<sub>a</sub> = - 20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
V <sub>IO</sub>	Input offset voltage		CMPV <sub>CC</sub> = 5.0V CMP <sub>REF</sub> = 2.5V, R <sub>s</sub> = 0Ω		3	50	mV
I <sub>B</sub>	Input bias current					5	μA
I <sub>IO</sub>	Input offset current					5	μA
V <sub>ICM</sub>	In-phase input voltage range			1.2		CMPV <sub>CC</sub> -0.5	V
A <sub>V</sub>	Voltage gain				∞		
t <sub>PD</sub>	Response time		CMPV <sub>CC</sub> = 5.0V CMP <sub>REF</sub> = 2.5V		60	2500	ns

# APPENDIX

## 3.1 Electrical characteristics

### 3.1.7 Timing requirements

**Table 3.1.10 Timing requirements (1)**

(V<sub>CC</sub> = 4.0 to 5.5 V, V<sub>SS</sub> = 0 V, T<sub>a</sub> = – 20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t <sub>w</sub> (RESET)	Reset input “L” pulse width	2			μs
t <sub>c</sub> (XIN)	External clock input cycle time	125			ns
t <sub>WH</sub> (XIN)	External clock input “H” pulse width	50			ns
t <sub>WL</sub> (XIN)	External clock input “L” pulse width	50			ns
t <sub>c</sub> (CNTR)	CNTR <sub>0</sub> , CNTR <sub>1</sub> input cycle time	200			ns
t <sub>WH</sub> (CNTR)	CNTR <sub>0</sub> , CNTR <sub>1</sub> input “H” pulse width	80			ns
t <sub>WL</sub> (CNTR)	CNTR <sub>0</sub> , CNTR <sub>1</sub> input “L” pulse width	80			ns
t <sub>WH</sub> (INT)	INT <sub>0</sub> to INT <sub>4</sub> input “H” pulse width	80			ns
t <sub>WL</sub> (INT)	INT <sub>0</sub> to INT <sub>4</sub> input “L” pulse width	80			ns
t <sub>c</sub> (SCLK1)	Serial I/O1 clock input cycle time (Note)	800			ns
t <sub>WH</sub> (SCLK1)	Serial I/O1 clock input “H” pulse width (Note)	370			ns
t <sub>WL</sub> (SCLK1)	Serial I/O1 clock input “L” pulse width (Note)	370			ns
t <sub>su</sub> (RXD–SCLK1)	Serial I/O1 clock input set up time	220			ns
t <sub>h</sub> (SCLK1–RXD)	Serial I/O1 clock input hold time	100			ns
t <sub>c</sub> (SCLK2)	Serial I/O2 clock input cycle time	1000			ns
t <sub>WH</sub> (SCLK2)	Serial I/O2 clock input “H” pulse width	400			ns
t <sub>WL</sub> (SCLK2)	Serial I/O2 clock input “L” pulse width	400			ns
t <sub>su</sub> (SIN2–SCLK2)	Serial I/O2 clock input set up time	200			ns
t <sub>h</sub> (SCLK2–SIN2)	Serial I/O2 clock input hold time	200			ns

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

**Table 3.1.11 Timing requirements (2)**

(V<sub>CC</sub> = 2.7 to 5.5 V, V<sub>SS</sub> = 0 V, T<sub>a</sub> = – 20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t <sub>w</sub> (RESET)	Reset input “L” pulse width	2			μs
t <sub>c</sub> (XIN)	External clock input cycle time	243			ns
t <sub>WH</sub> (XIN)	External clock input “H” pulse width	100			ns
t <sub>WL</sub> (XIN)	External clock input “L” pulse width	100			ns
t <sub>c</sub> (CNTR)	CNTR <sub>0</sub> , CNTR <sub>1</sub> input cycle time	500			ns
t <sub>WH</sub> (CNTR)	CNTR <sub>0</sub> , CNTR <sub>1</sub> input “H” pulse width	230			ns
t <sub>WL</sub> (CNTR)	CNTR <sub>0</sub> , CNTR <sub>1</sub> input “L” pulse width	230			ns
t <sub>WH</sub> (INT)	INT <sub>0</sub> to INT <sub>4</sub> input “H” pulse width	230			ns
t <sub>WL</sub> (INT)	INT <sub>0</sub> to INT <sub>4</sub> input “L” pulse width	230			ns
t <sub>c</sub> (SCLK1)	Serial I/O1 clock input cycle time (Note)	2000			ns
t <sub>WH</sub> (SCLK1)	Serial I/O1 clock input “H” pulse width (Note)	950			ns
t <sub>WL</sub> (SCLK1)	Serial I/O1 clock input “L” pulse width (Note)	950			ns
t <sub>su</sub> (RXD–SCLK1)	Serial I/O1 clock input set up time	400			ns
t <sub>h</sub> (SCLK1–RXD)	Serial I/O1 clock input hold time	200			ns
t <sub>c</sub> (SCLK2)	Serial I/O2 clock input cycle time	2000			ns
t <sub>WH</sub> (SCLK2)	Serial I/O2 clock input “H” pulse width	950			ns
t <sub>WL</sub> (SCLK2)	Serial I/O2 clock input “L” pulse width	950			ns
t <sub>su</sub> (SIN2–SCLK2)	Serial I/O2 clock input set up time	400			ns
t <sub>h</sub> (SCLK2–SIN2)	Serial I/O2 clock input hold time	300			ns

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

## 3.1 Electrical characteristics

### 3.1.8 Switching characteristics

**Table 3.1.12 Switching characteristics (1)**  
(V<sub>CC</sub> = 4.0 to 5.5 V, V<sub>SS</sub> = 0 V, T<sub>a</sub> = – 20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t <sub>WH</sub> (SCLK1)	Serial I/O1 clock output "H" pulse width	Fig. 3.1.1	tc(SCLK1)/2–30			ns
t <sub>WL</sub> (SCLK1)	Serial I/O1 clock output "L" pulse width		tc(SCLK1)/2–30			ns
t <sub>d</sub> (SCLK1–TXD)	Serial I/O1 output delay time (Note 1)				140	ns
t <sub>v</sub> (SCLK1–TXD)	Serial I/O1 output valid time (Note 1)		–30			ns
t <sub>r</sub> (SCLK1)	Serial I/O1 clock output rising time				30	ns
t <sub>f</sub> (SCLK1)	Serial I/O1 clock output falling time				30	ns
t <sub>WH</sub> (SCLK2)	Serial I/O2 clock output "H" pulse width	Fig. 3.1.1	tc(SCLK2)/2–160			ns
t <sub>WL</sub> (SCLK2)	Serial I/O2 clock output "L" pulse width		tc(SCLK2)/2–160			ns
t <sub>d</sub> (SCLK2–SOUT2)	Serial I/O2 output delay time (Note 2)				200	ns
t <sub>v</sub> (SCLK2–SOUT2)	Serial I/O2 output valid time (Note 2)		0			ns
t <sub>f</sub> (SCLK2)	Serial I/O2 clock output falling time				30	ns
t <sub>r</sub> (CMOS)	CMOS output rising time (Note 3)		Fig. 3.1.1		10	30
t <sub>f</sub> (CMOS)	CMOS output falling time (Note 3)			10	30	ns

**Note 1:** When the P45/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is "0".

**2:** When the P71/SOUT2, P72/SCLK2 P-channel output disable bit of the serial I/O2 control register1 (bit 7 of address 001D16) is "0".

**3:** XOUT pin is excluded.

**Table 3.1.13 Switching characteristics (2)**  
(V<sub>CC</sub> = 2.7 to 5.5 V, V<sub>SS</sub> = 0 V, T<sub>a</sub> = – 20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t <sub>WH</sub> (SCLK1)	Serial I/O1 clock output "H" pulse width	Fig. 3.1.1	tc(SCLK1)/2–50			ns
t <sub>WL</sub> (SCLK1)	Serial I/O1 clock output "L" pulse width		tc(SCLK1)/2–50			ns
t <sub>d</sub> (SCLK1–TXD)	Serial I/O1 output delay time (Note 1)				350	ns
t <sub>v</sub> (SCLK1–TXD)	Serial I/O1 output valid time (Note 1)		–30			ns
t <sub>r</sub> (SCLK1)	Serial I/O1 clock output rising time				50	ns
t <sub>f</sub> (SCLK1)	Serial I/O1 clock output falling time				50	ns
t <sub>WH</sub> (SCLK2)	Serial I/O2 clock output "H" pulse width	Fig. 3.1.1	tc(SCLK2)/2–240			ns
t <sub>WL</sub> (SCLK2)	Serial I/O2 clock output "L" pulse width		tc(SCLK2)/2–240			ns
t <sub>d</sub> (SCLK2–SOUT2)	Serial I/O2 output delay time (Note 2)				400	ns
t <sub>v</sub> (SCLK2–SOUT2)	Serial I/O2 output valid time (Note 2)		0			ns
t <sub>f</sub> (SCLK2)	Serial I/O2 clock output falling time				50	ns
t <sub>r</sub> (CMOS)	CMOS output rising time (Note 3)		Fig. 3.1.1		20	50
t <sub>f</sub> (CMOS)	CMOS output falling time (Note 3)			20	50	ns

**Note 1:** When the P45/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is "0".

**2:** When the P71/SOUT2, P72/SCLK2 P-channel output disable bit of the serial I/O2 control register1 (bit 7 of address 001D16) is "0".

**3:** XOUT pin is excluded.

# APPENDIX

## 3.1 Electrical characteristics

### 3.1.9 Timing requirements in memory expansion mode and microprocessor mode

**Table 3.1.14 Timing requirements in memory expansion and microprocessor mode**  
(Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = - 20 to 85 °C, in high-speed mode, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
tsu(ONW-φ)	ONW input set up time	-20			ns
th(φ-ONW)	ONW input hold time	-20			ns
tsu(DB-φ)	Data bus set up time	50			ns
th(φ-DB)	Data bus hold time	0			ns
tsu(ONW-RD), tsu(ONW-WR)	ONW input set up time	-20			ns
th(RD-ONW), th(WR-ONW)	ONW input hold time	-20			ns
tsu(DB-RD)	Data bus set up time	50			ns
th(RD-DB)	Data bus hold time	0			ns

### 3.1.10 Switching characteristics in memory expansion mode and microprocessor mode

**Table 3.1.15 Switching characteristics in memory expansion and microprocessor mode**  
(Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = - 20 to 85 °C, in high-speed mode, 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)	AD15-AD8 delay time			16	35	ns
td(φ-AL)	AD7-AD0 delay time			20	40	ns
tv(φ-AH)	AD15-AD8 valid time		2	5		ns
tv(φ-AL)	AD7-AD0 valid time		2	5		ns
td(φ-SYNC)	SYNC delay time			16		ns
tv(φ-SYNC)	SYNC valid time			5		ns
td(φ-DB)	Data bus delay time			15	30	ns
tv(φ-DB)	Data bus valid time		10			ns
tWL(RD), tWL(WR)	RD pulse width, WR pulse width		tc(XIN)-10			ns
	RD pulse width, WR pulse width (When one-wait is valid)		3tc(XIN)-10			ns
td(AH-RD), td(AH-WR)	AD15-AD8 delay time		tc(XIN)-35	tc(XIN)-16		ns
td(AL-RD), td(AL-WR)	AD7-AD0 delay time		tc(XIN)-40	tc(XIN)-20		ns
tv(RD-AH), tv(WR-AH)	AD15-AD8 valid time		2	5		ns
tv(RD-AL), tv(WR-AL)	AD7-AD0 valid time		2	5		ns
td(WR-DB)	Data bus delay time			15	30	ns
tv(WR-DB)	Data bus valid time		10			ns
td(RESET-RESETOUT)	RESETOUT output delay time				200	ns
tv(φ-RESETOUT)	RESETOUT output valid time (Note)	0		100	ns	

**Note:** The RESETOUT output 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

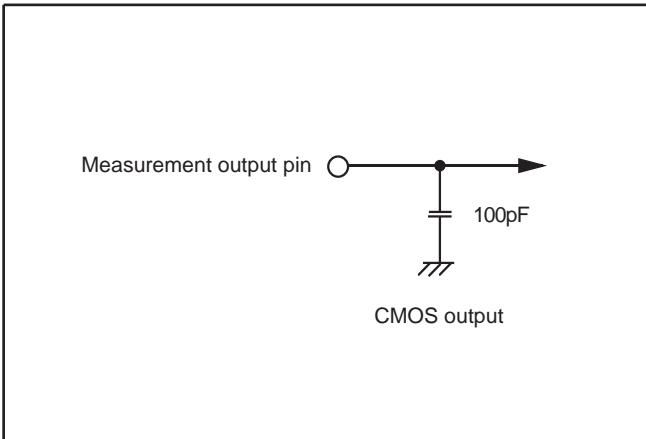


Fig. 3.1.1 Circuit for measuring output switching characteristics (1)

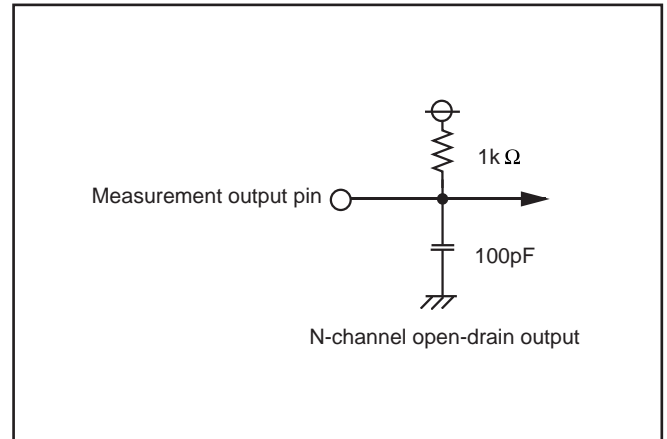


Fig. 3.1.2 Circuit for measuring output switching characteristics (2)

# APPENDIX

## 3.1 Electrical characteristics

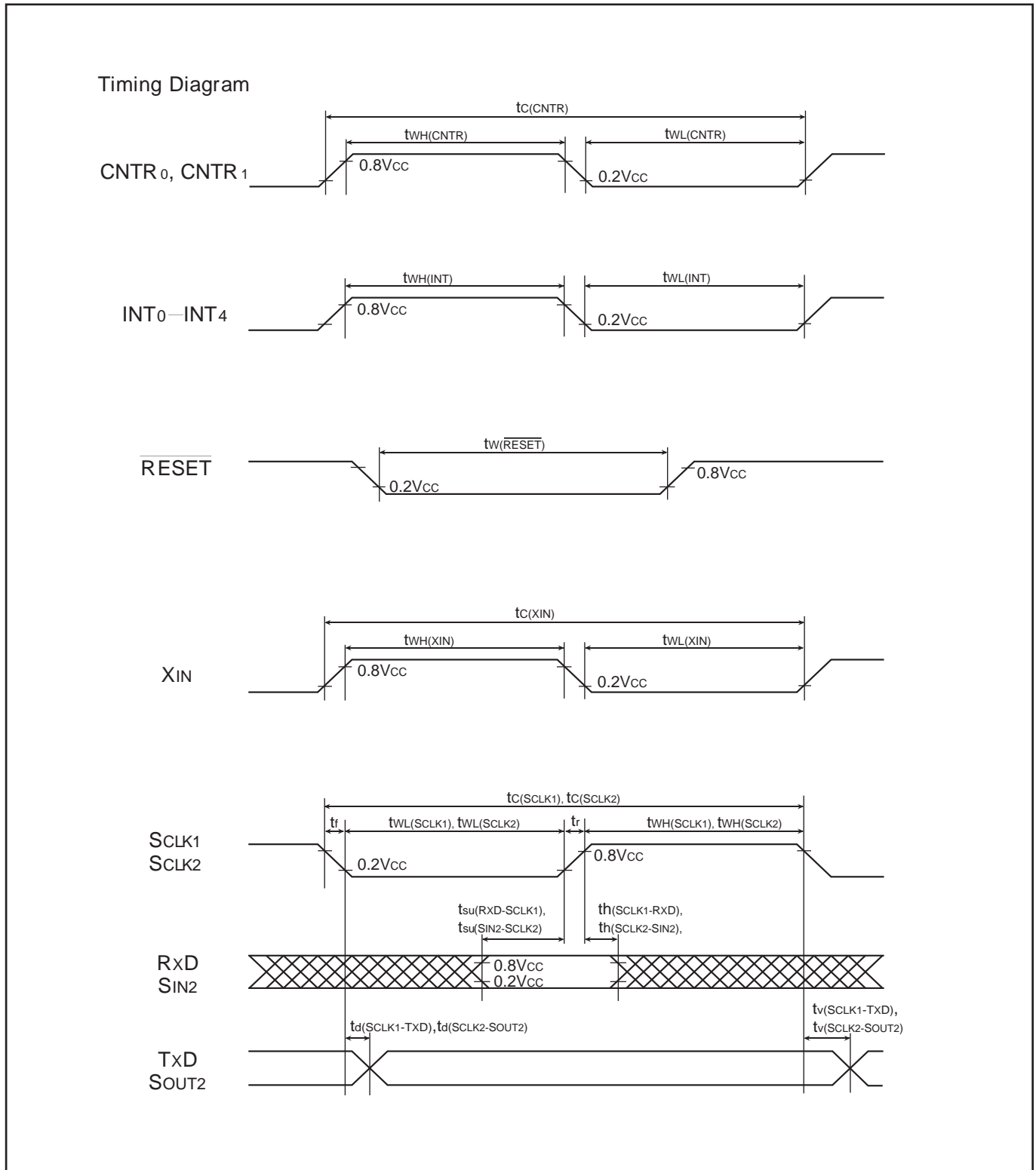
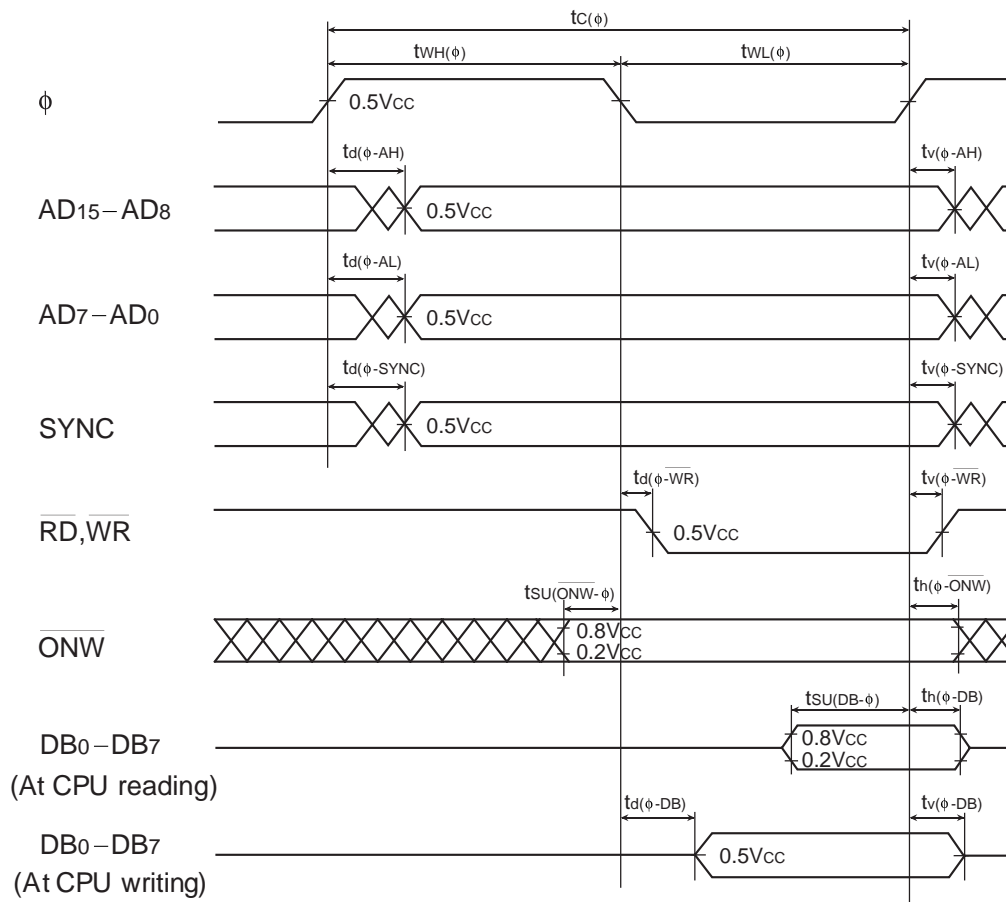


Fig. 3.1.3 Timing diagram (1) (in single-chip mode)

Timing Diagram in Memory Expansion Mode and Microprocessor Mode(CMOS level input)



Timing Diagram in Microprocessor Mode

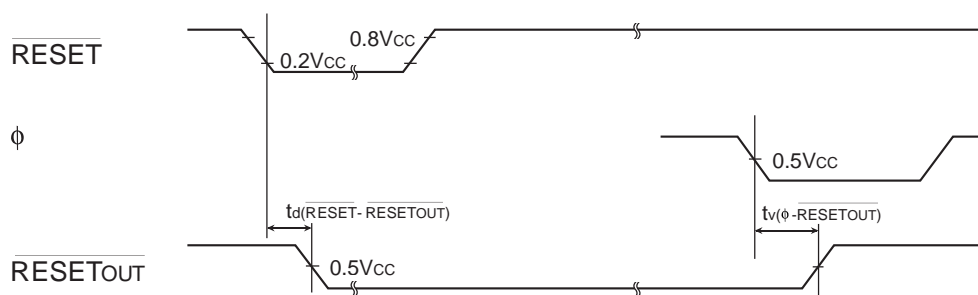


Fig. 3.1.4 Timing diagram (2) (in memory expansion mode and microprocessor mode)

# APPENDIX

## 3.1 Electrical characteristics

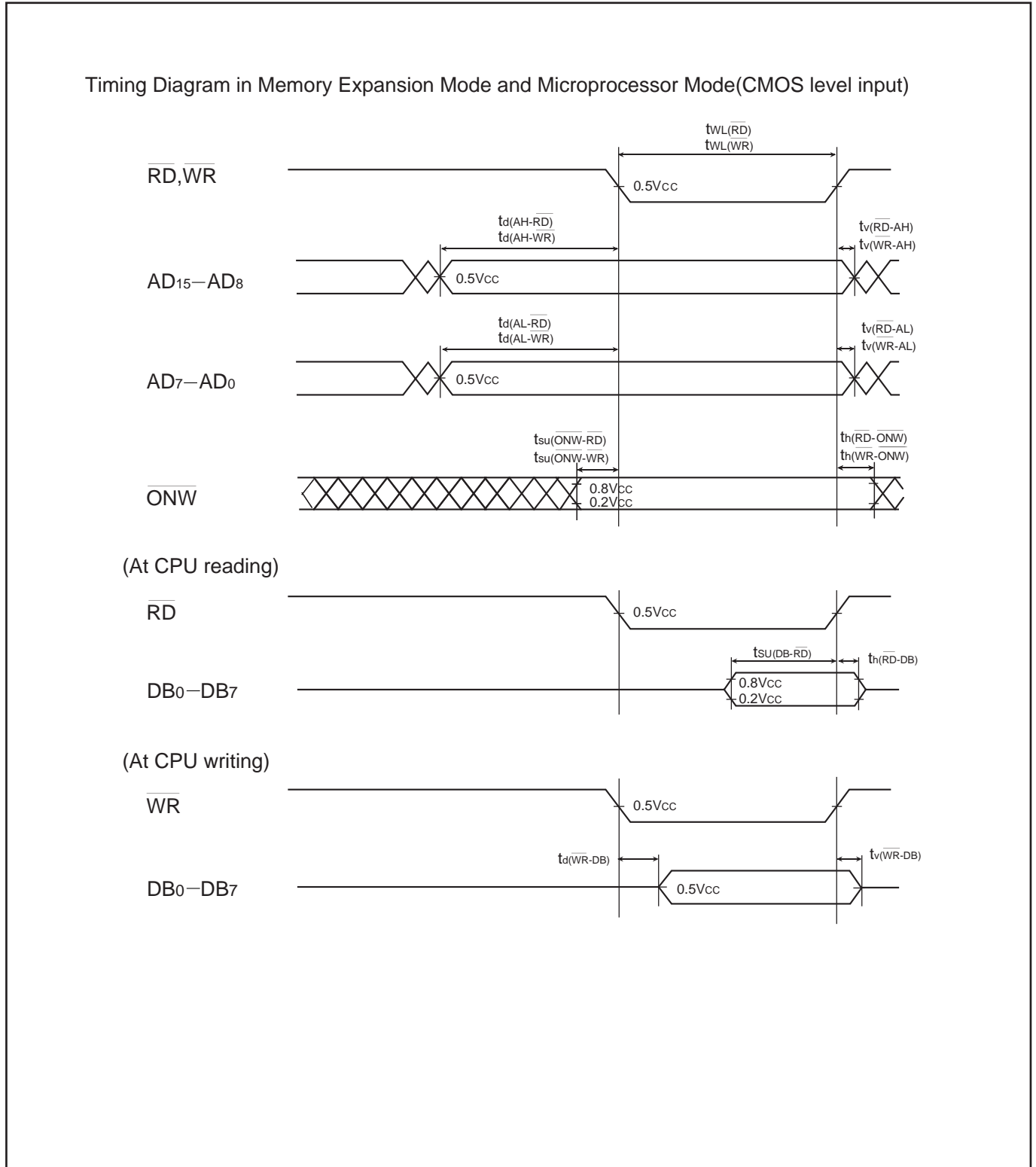
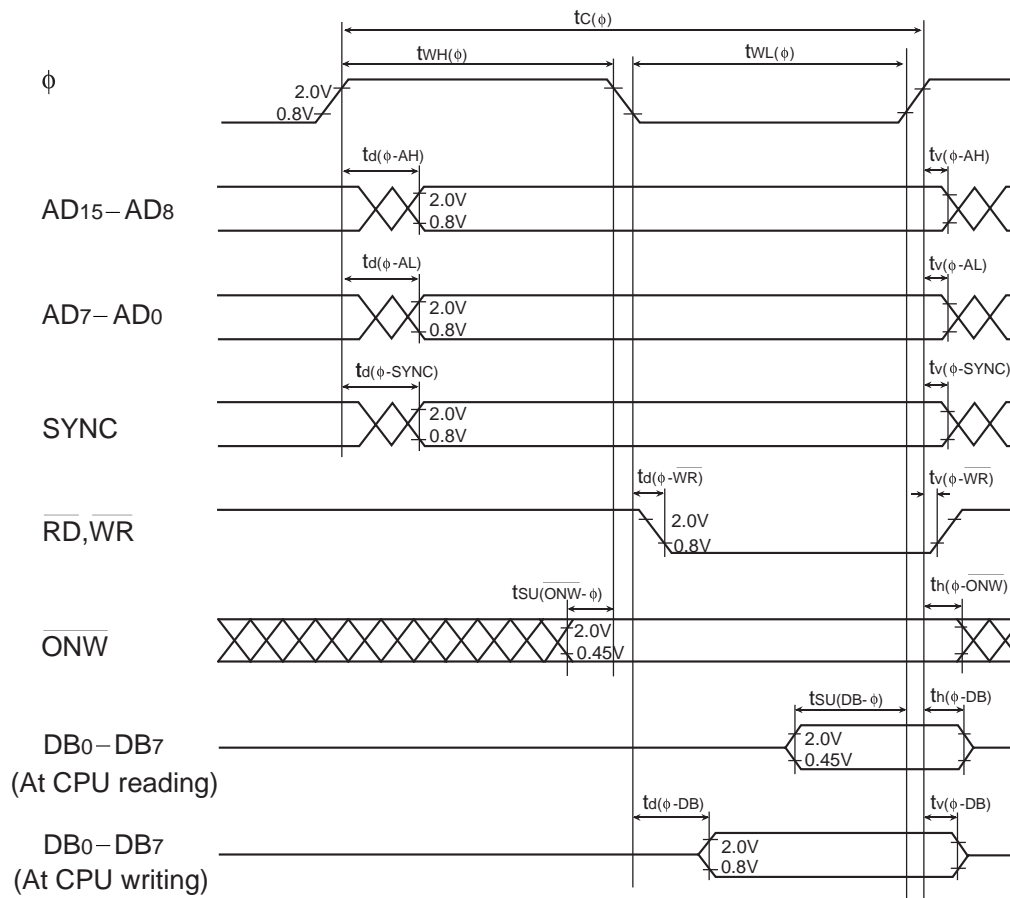


Fig. 3.1.5 Timing diagram (3) (in memory expansion mode and microprocessor mode)

Timing Diagram in Memory Expansion Mode and Microprocessor Mode(TTL level input)



Timing Diagram in Microprocessor Mode

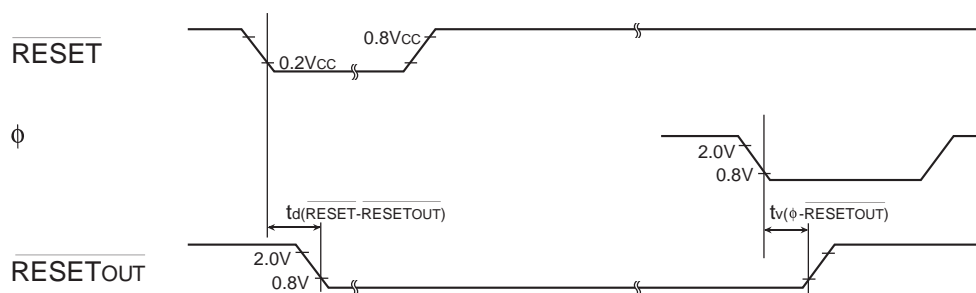


Fig. 3.1.6 Timing diagram (4) (in memory expansion mode and microprocessor mode)

# APPENDIX

## 3.1 Electrical characteristics

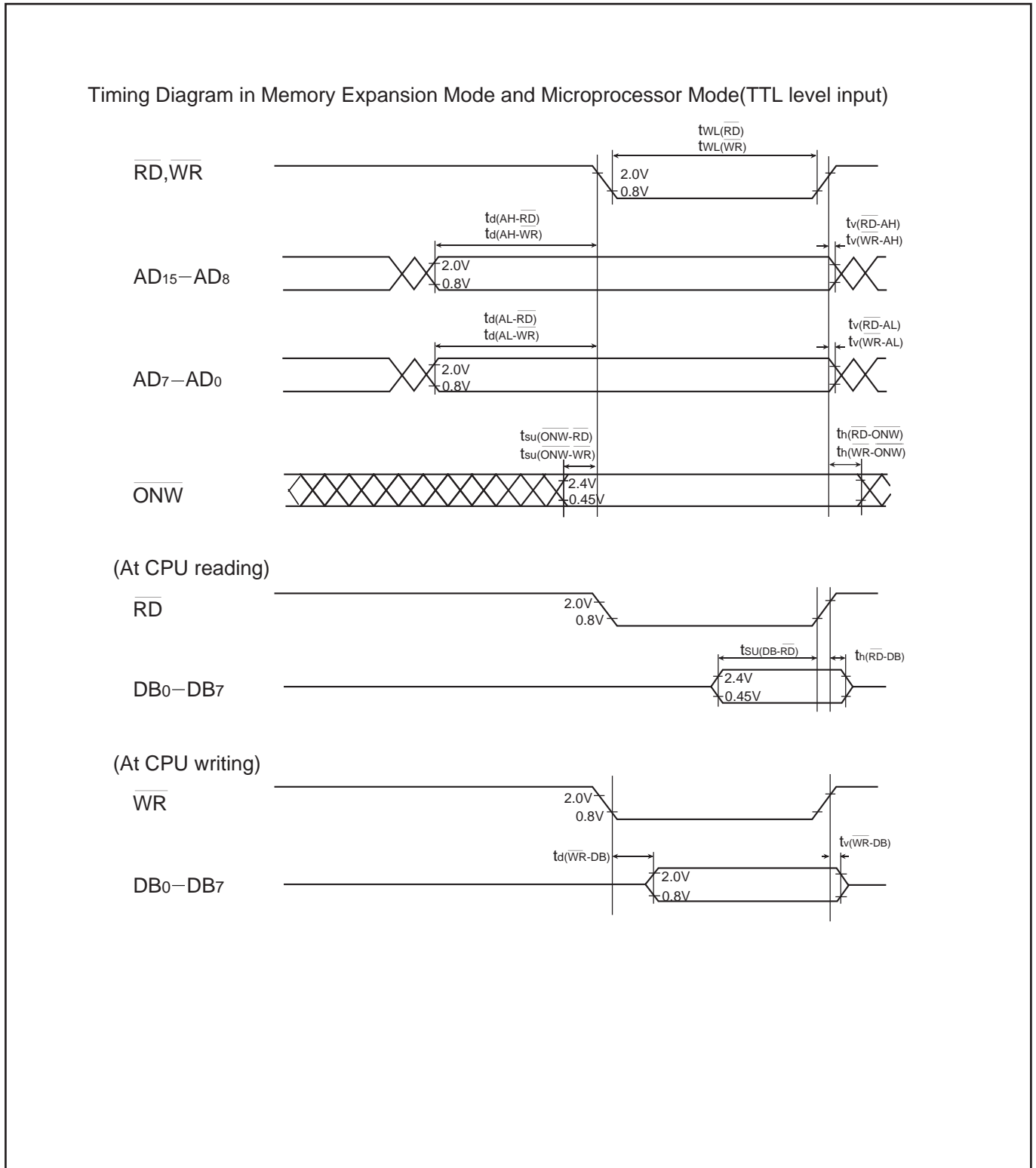


Fig. 3.1.7 Timing diagram (5) (in memory expansion mode and microprocessor mode)

### 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.

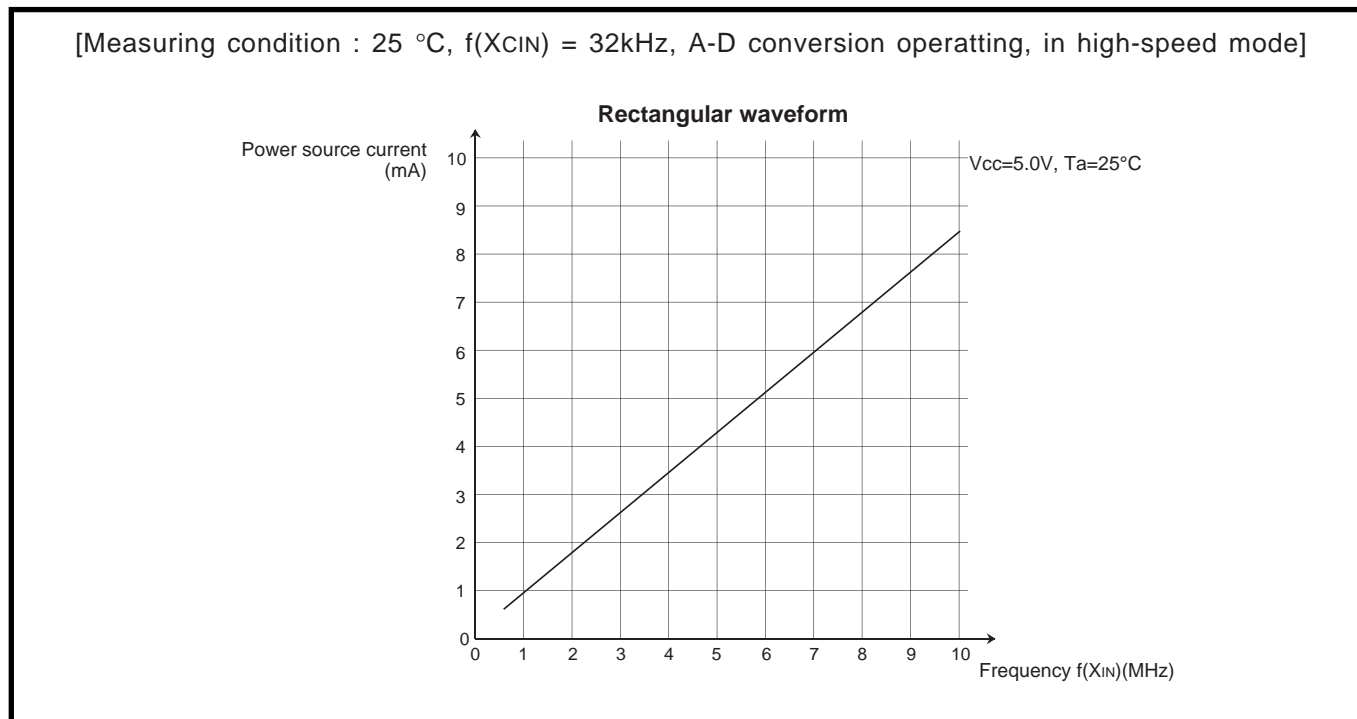


Fig. 3.2.1 Power source current characteristic example

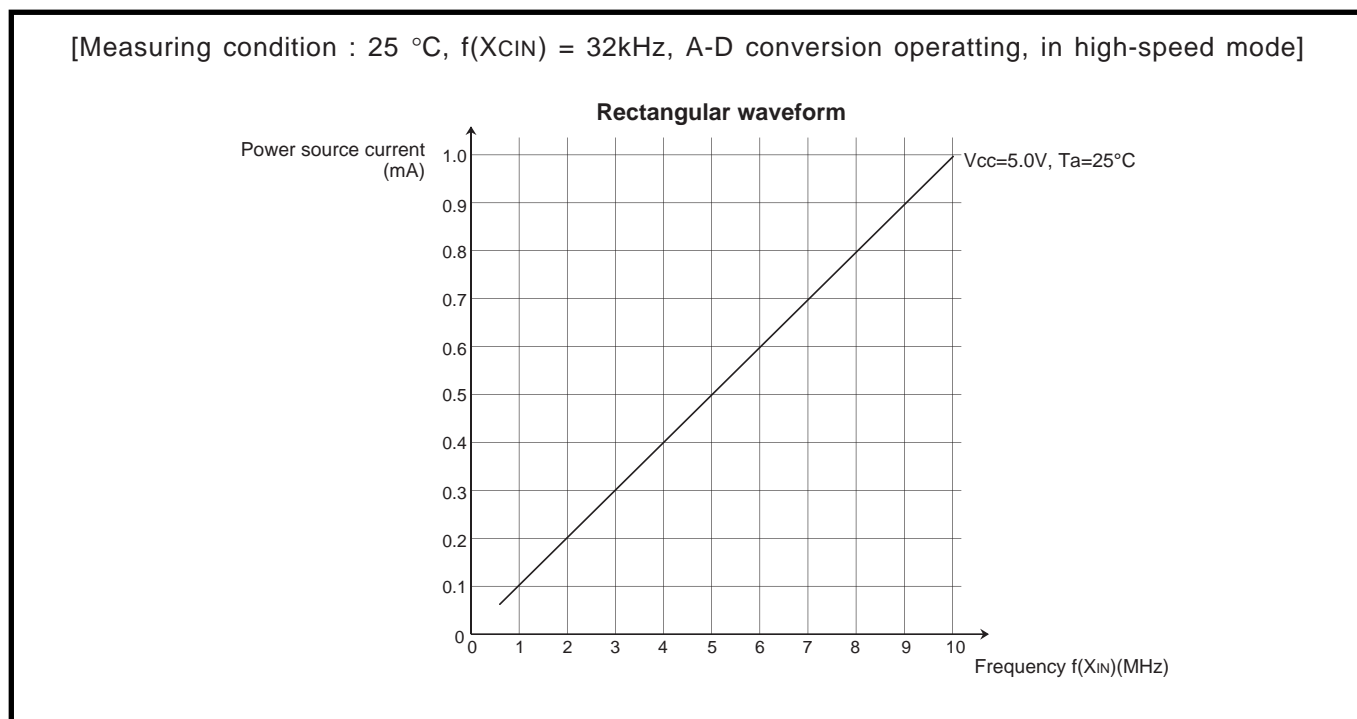


Fig. 3.2.2 Power source current characteristic example (in wait mode)

# 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.

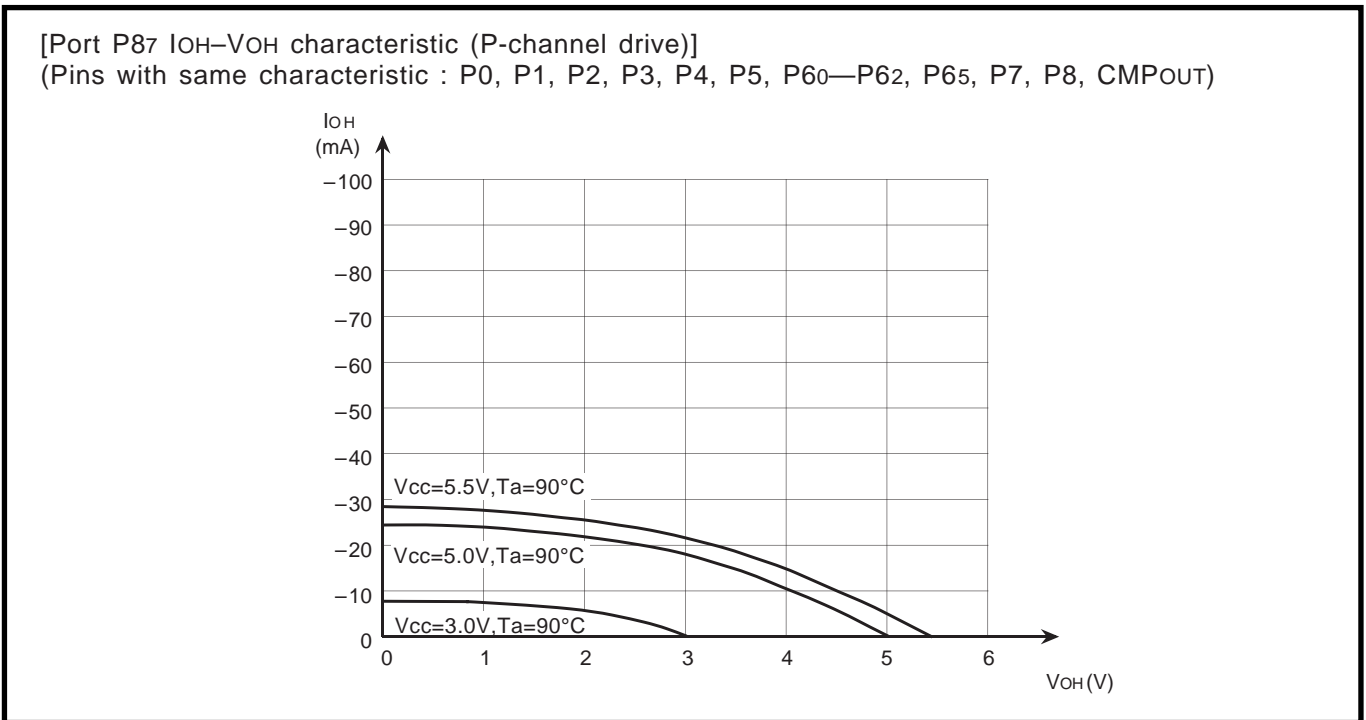


Fig. 3.2.3 Standard characteristic example of CMOS output port at P-channel drive (1)

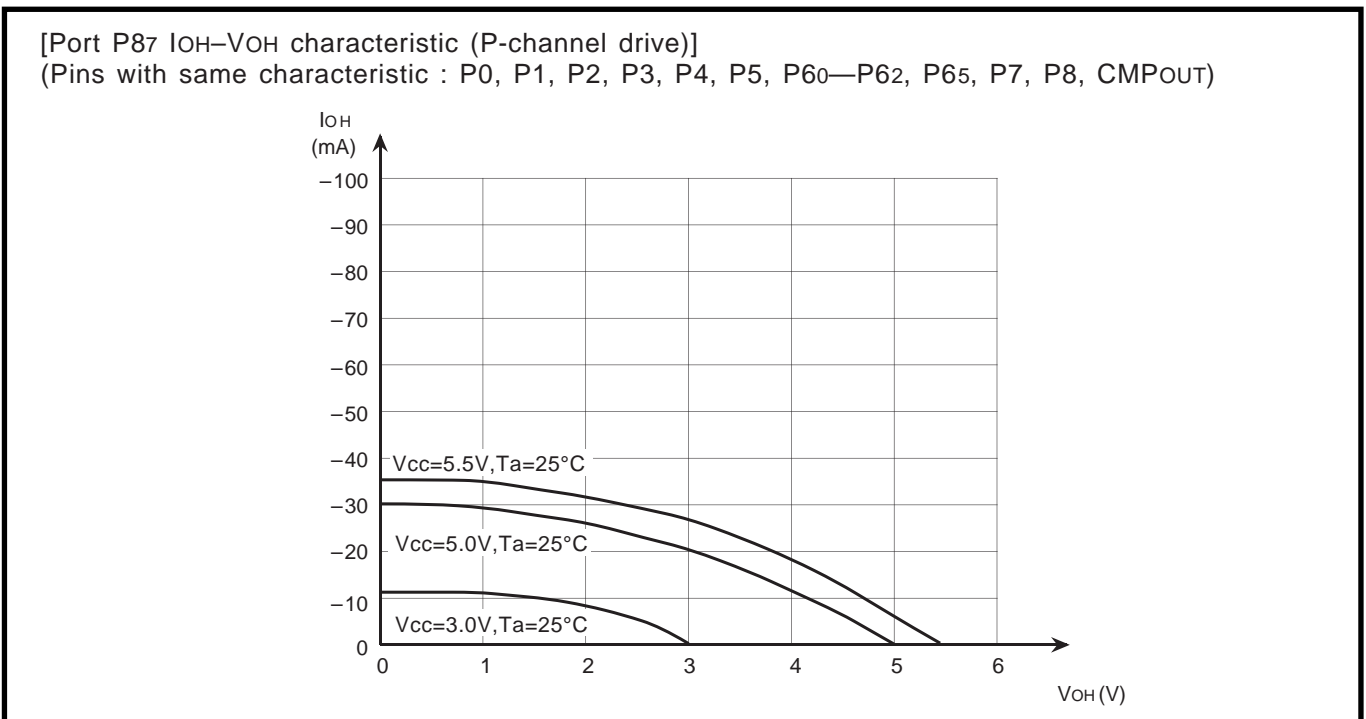


Fig. 3.2.4 Standard characteristic example of CMOS output port at P-channel drive (2)



## 3.2 Standard characteristics

[Port P87 IOL-VOL characteristic (N-channel drive)]

(Pins with same characteristic : P0, P1, P20—P23, P3, P4, P5, P60—P62, P65, P7, P8, CMPOUT, P24—P27 except in single-chip mode)

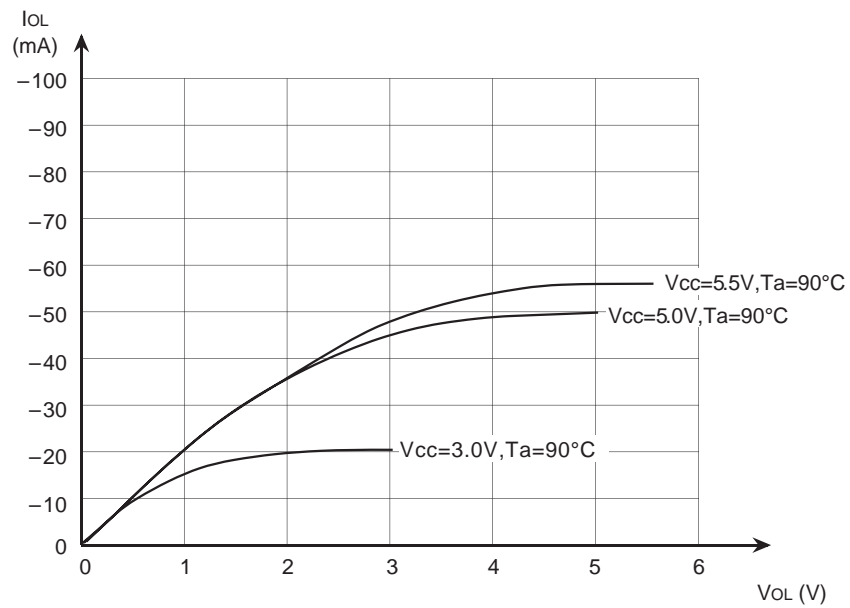


Fig. 3.2.5 Standard characteristic example of CMOS output port at N-channel drive (1)

[Port P87 IOL-VOL characteristic (N-channel drive)]

(Pins with same characteristic : P0, P1, P20—P23, P3, P4, P5, P60—P62, P65, P7, P8, CMPOUT, P24—P27 except in single-chip mode)

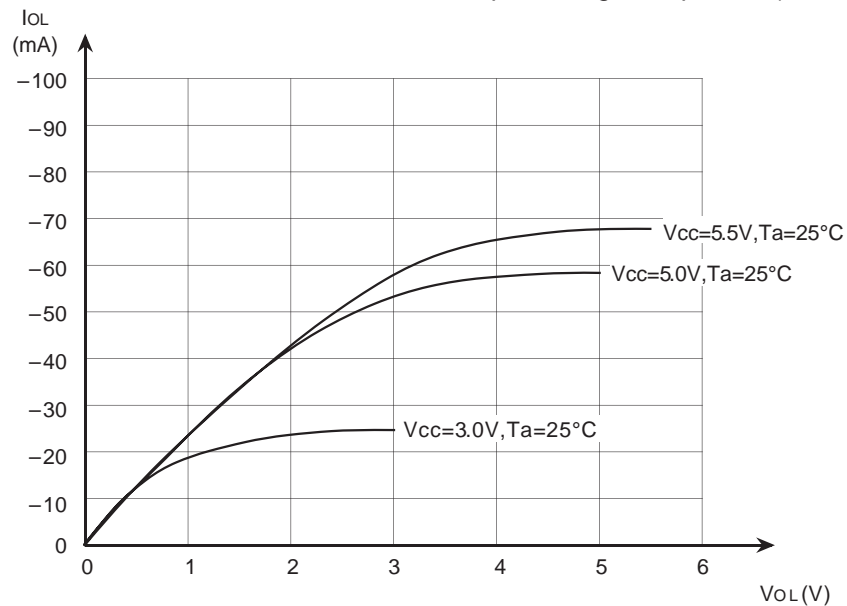


Fig. 3.2.6 Standard characteristic example of CMOS output port at N-channel drive (2)

# APPENDIX

## 3.2 Standard characteristics

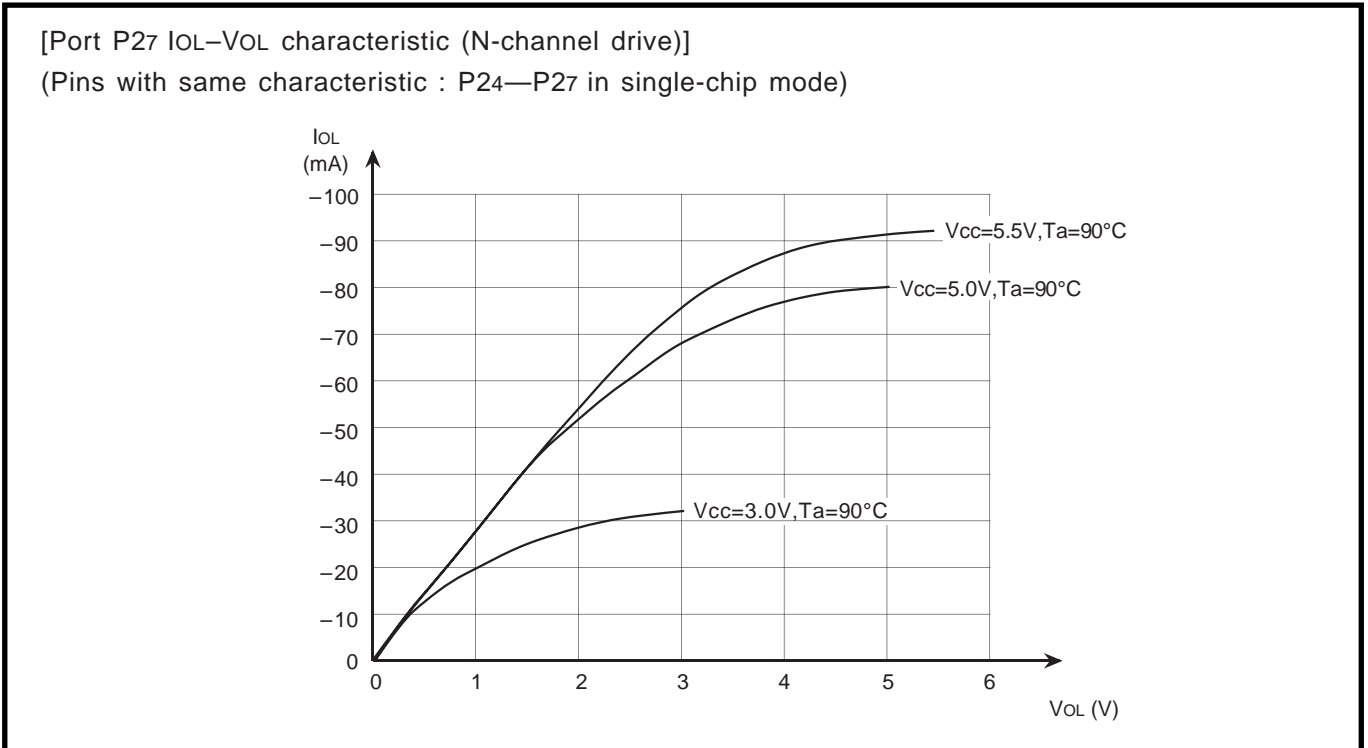


Fig. 3.2.7 Standard characteristic example of CMOS output port at N-channel drive (4)

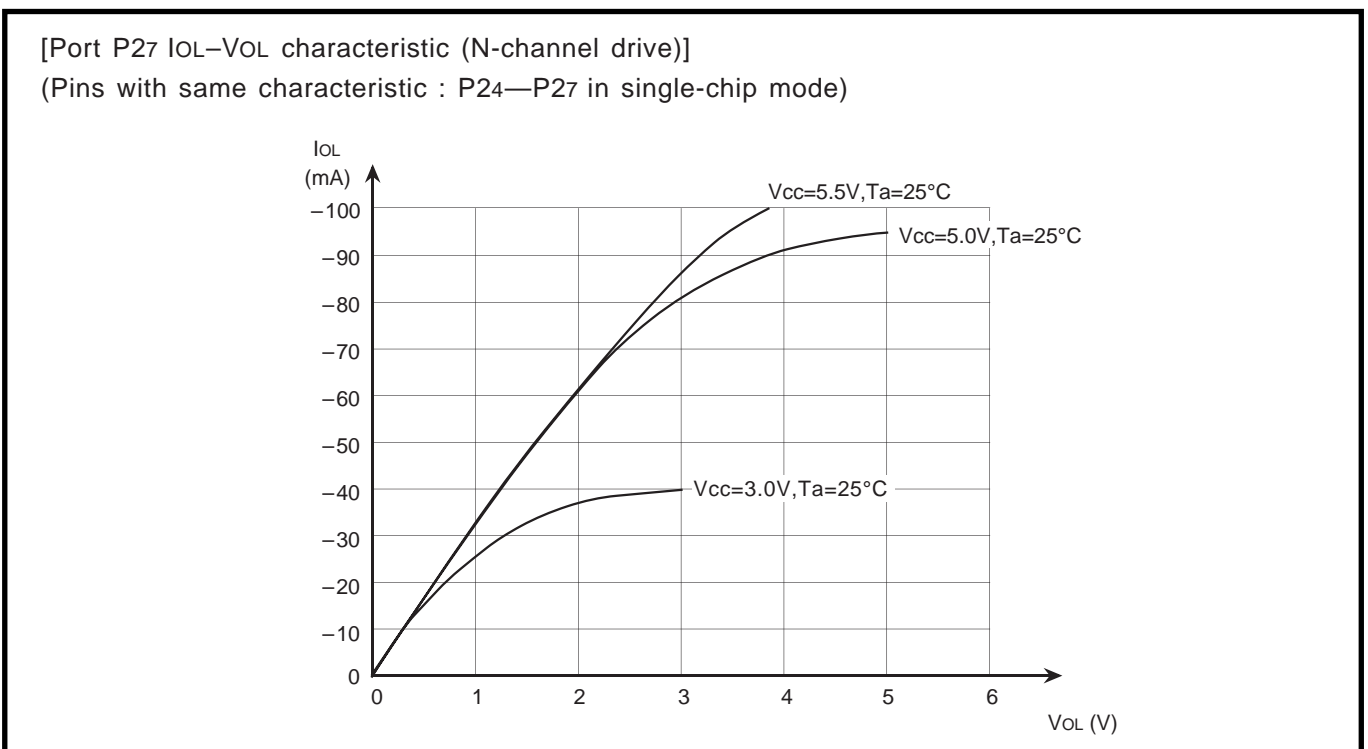


Fig. 3.2.8 Standard characteristic example of CMOS output port at N-channel drive (5)

## 3.2.3 Input current standard characteristic examples

Figure 3.2.9 and Figure 3.2.10 show input current standard characteristic examples.

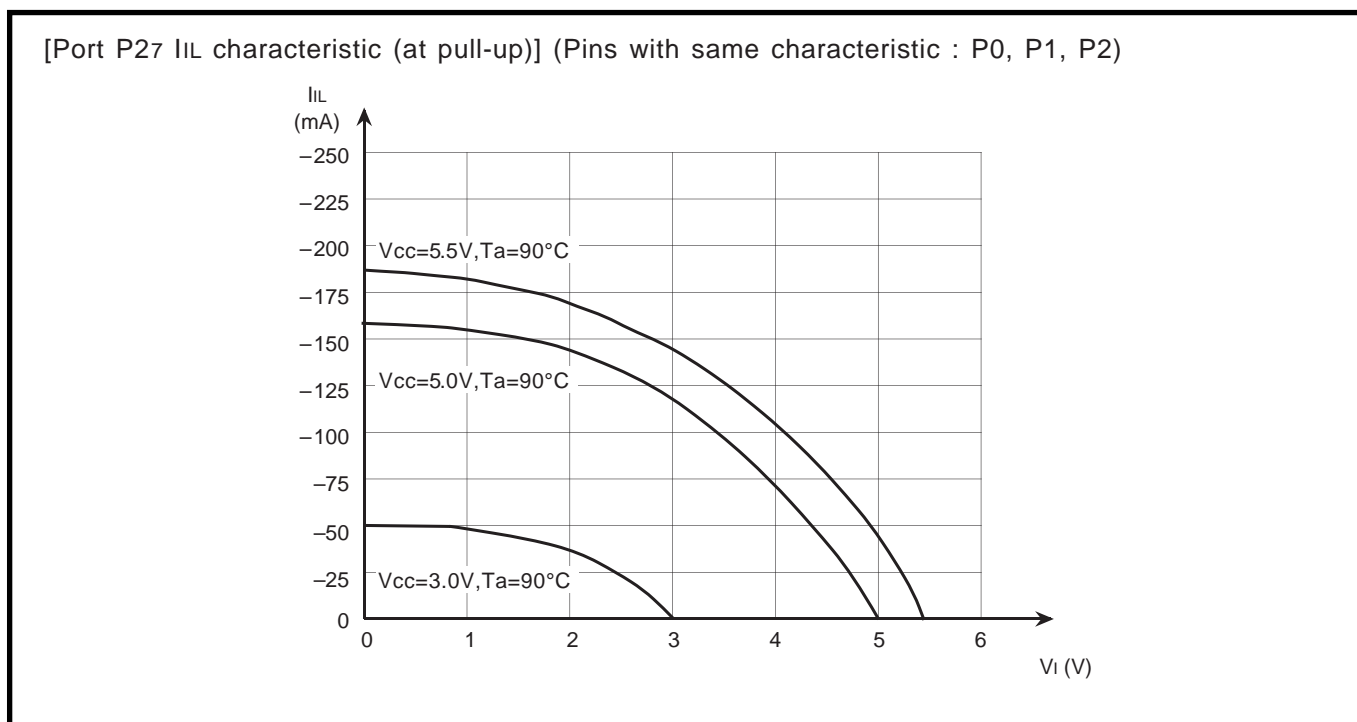


Fig. 3.2.9 Standard characteristic example of input current at connecting pull-up transistor (1)

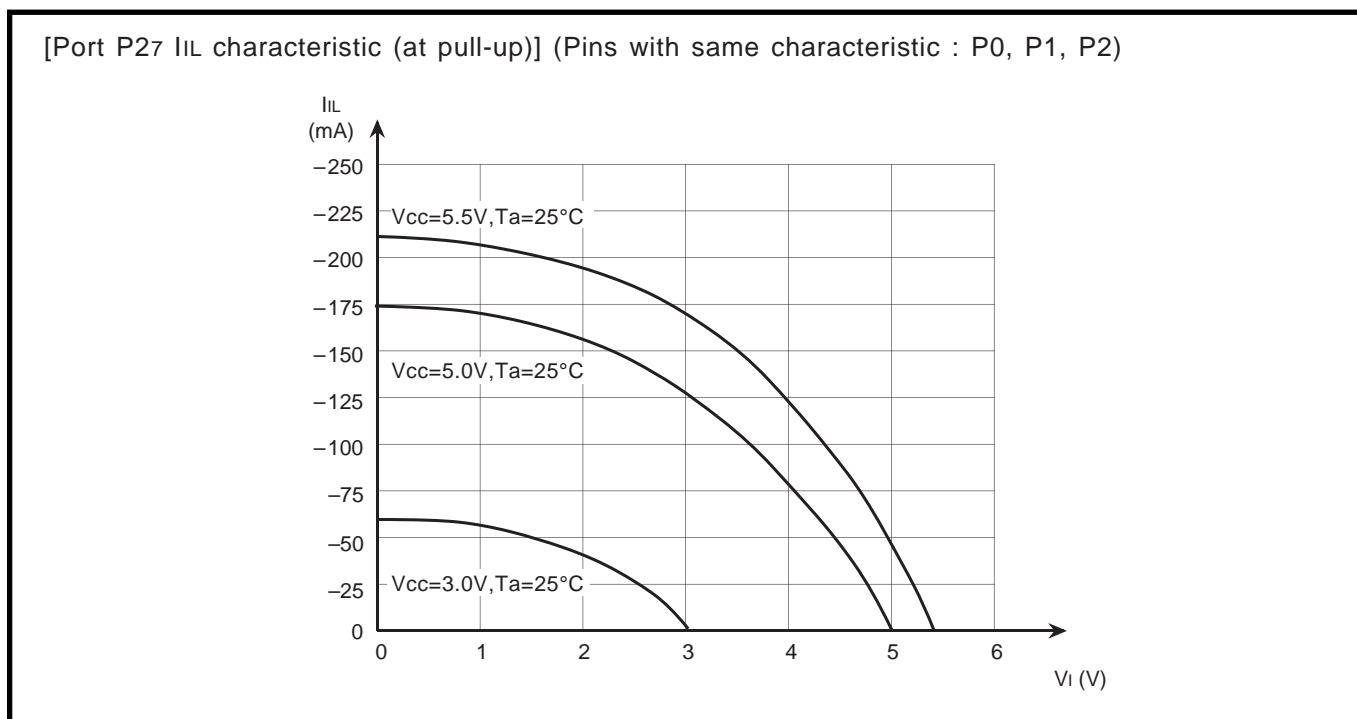


Fig. 3.2.10 Standard characteristic example of input current at connecting pull-up transistor (2)

# APPENDIX

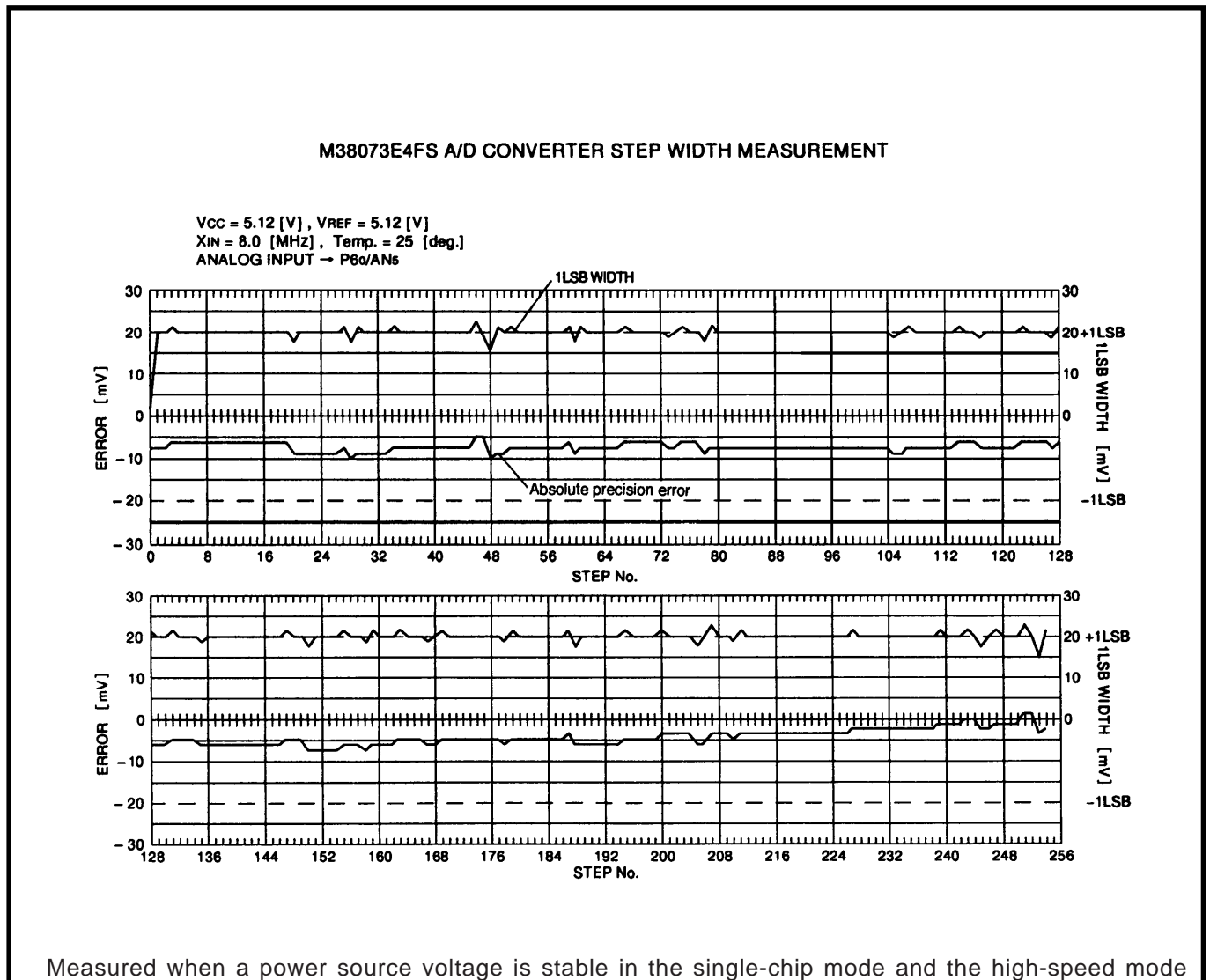
## 3.2 Standard characteristics

### 3.2.4 A-D conversion standard characteristics

Figure 3.2.11 shows the A-D conversion standard characteristics.

The lower-side line on the graph indicates the absolute precision error. It represents the deviation from the ideal value. For example, the conversion of output code from 0 to 1 occurs ideally at the point of AN0 = 10 mV, but the measured value is 0 mV. Accordingly, the measured point of conversion is represented as “10 – 0 = 10 mV.”

The upper-side line on the graph indicates the width of input voltages equivalent to output codes. For example, the measured width of the input voltage for output code 13 is 22 mV, so the differential nonlinear error is represented as “22 – 20 = 2 mV” (0.1 LSB).



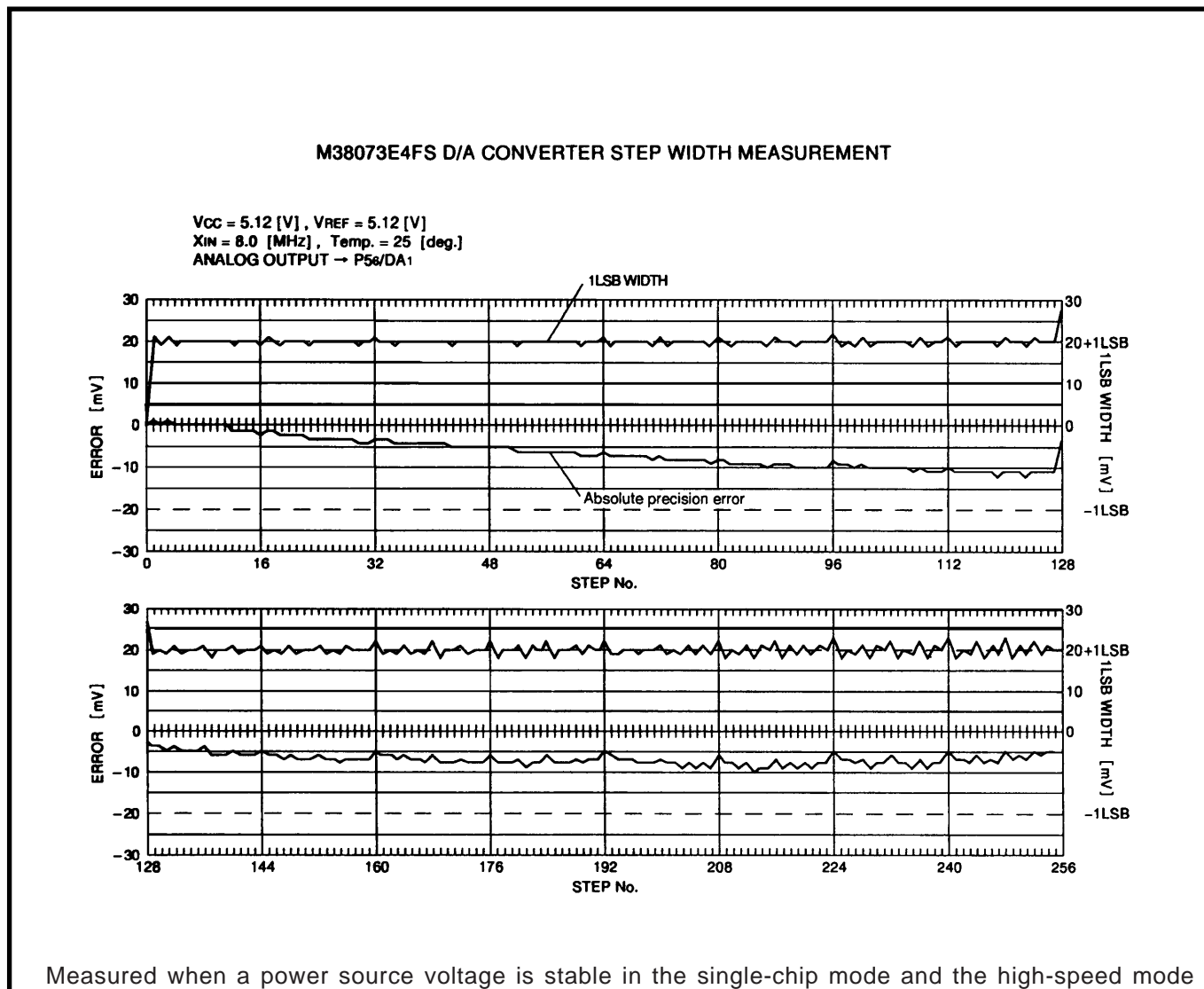
Measured when a power source voltage is stable in the single-chip mode and the high-speed mode

**Fig. 3.2.11 A-D conversion standard characteristics**

### 3.2.5 D-A conversion standard characteristics

Figure 3.2.12 shows the D-A conversion standard characteristics. The lower-side line on the graph indicates the absolute precision error. In this case, it represents the difference between the ideal analog output value for an input code and the measured value.

The upper-side line on the graph indicates the change width of output analog value to a one-bit change of input code.



Measured when a power source voltage is stable in the single-chip mode and the high-speed mode

Fig. 3.2.12 D-A conversion standard characteristics

# APPENDIX

## 3.3 Notes on use

### 3.3 Notes on use

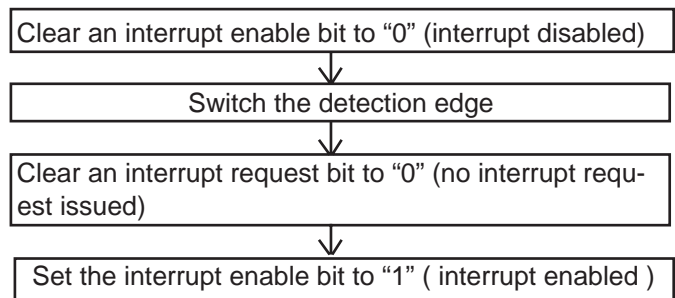
#### 3.3.1 Notes on interrupts

##### (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) Bit 7 of the interrupt control register 2

Fix the bit 7 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.

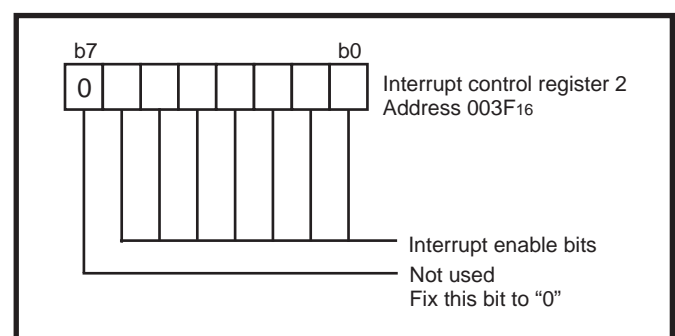


Fig. 3.3.1 Structure of interrupt control register 2

#### 3.3.2 Notes on the serial I/O1

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

As for the serial I/O1 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/O1 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/O1 enable bit is cleared to "0" (serial I/O1 disabled), the internal transmission is running (in this case, since pins Tx<sub>D</sub>, Rx<sub>D</sub>, SCLK<sub>1</sub>, and SRDY<sub>1</sub> 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/O1 enable bit is set to "1" at this time, the data during internally shifting is output to the Tx<sub>D</sub> pin and it may cause an operation failure to a microcomputer.

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

As for the serial I/O1 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/O1 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.

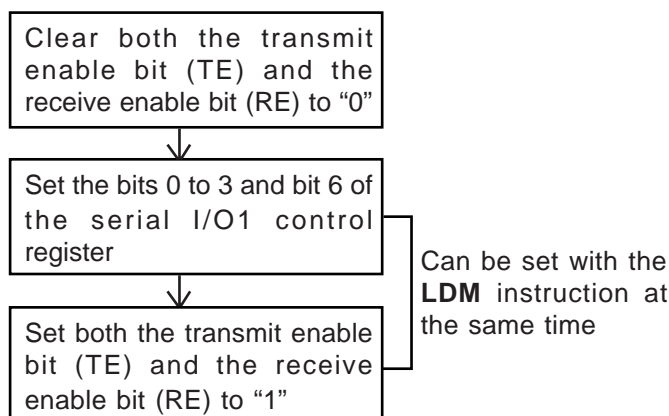
In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to "0" (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/O1 enable bit to "0" (serial I/O1 disabled) (refer to (1)).

### (4) The SRDY pin on a receiving side

When signals are output from the 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 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/O1 control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to "0."



### (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 A-D converter

#### (1) Input of signals from signal source with high impedance to an analog input pin

Make the signal source impedance for analog input low, or equip an analog input pin with an external capacitor of 0.01  $\mu$ F to 1  $\mu$ F. Further, make sure to check the operation of application products on the user side.

#### Reason

The A-D converter builds in the capacitor for analog voltage comparison. Accordingly, when signals from signal source with high impedance are input to an analog input pin, a charge and discharge noise generates. This may cause the A-D conversion precision to be worse.

# APPENDIX

## 3.3 Notes on use

---

### (2) AVss pin

Connect a power source for the A-D converter, AVss pin to the Vss line of the analog circuit.

### (3) A clock frequency during an A-D conversion

The comparator consists of a capacity coupling, and a charge of the capacity will be lost if the clock frequency is too low. Thus, make sure the following during an A-D conversion.

- $f(X_{IN})$  is 500 kHz or more .  
(When the ONW pin is "L",  $f(X_{IN})$  is 1 MHz or more.)
- Do not execute the STP instruction and WIT instruction.

### 3.3.4 Notes on the $\overline{RESET}$ pin

When a rising time of the reset signal is long, connect a ceramic capacitor or others across the  $\overline{RESET}$  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  $\overline{RESET}$  pin, a microcomputer may malfunction.

### 3.3.5 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) The AVss pin when not using the A-D converter

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

- AVss : Connect to the Vss pin

#### Reason

If the AVss pin is opened, the microcomputer may malfunction by effect of noise or others.

### 3.3.6 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

### 3.3.7 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
M38073E4FS	PCA4738L-80A
M38073E4FP (one-time blank)	PCA4738F-80A

#### (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
PCA4738F-80A	CMOS	CMOS	OFF
PCA4738L-80A			

**Table 3.3.3 Setting of PROM programmer address**

Microcomputer	PROM programmer start address	PROM programmer completion address
M38073E4FS	Address : 4080 <sub>16</sub> (Note 1)	Address : 7FFD <sub>16</sub> (Note 1)
M38073E4FP		

Note : 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) 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.

### 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.

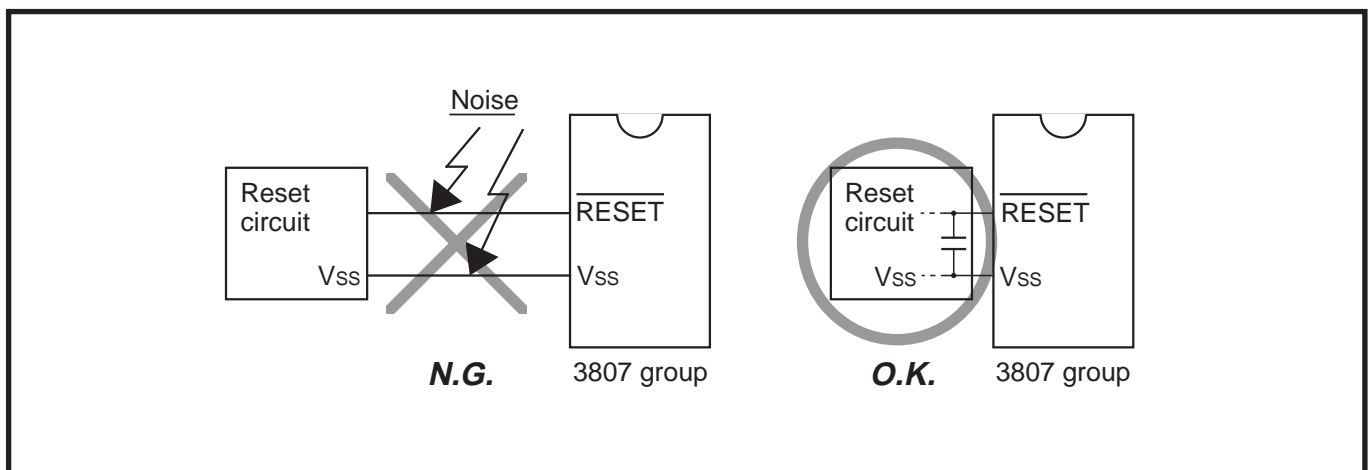


Fig. 3.4.1 Wiring for the  $\overline{\text{RESET}}$  pin

##### (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.

# APPENDIX

## 3.4 Countermeasures against noise

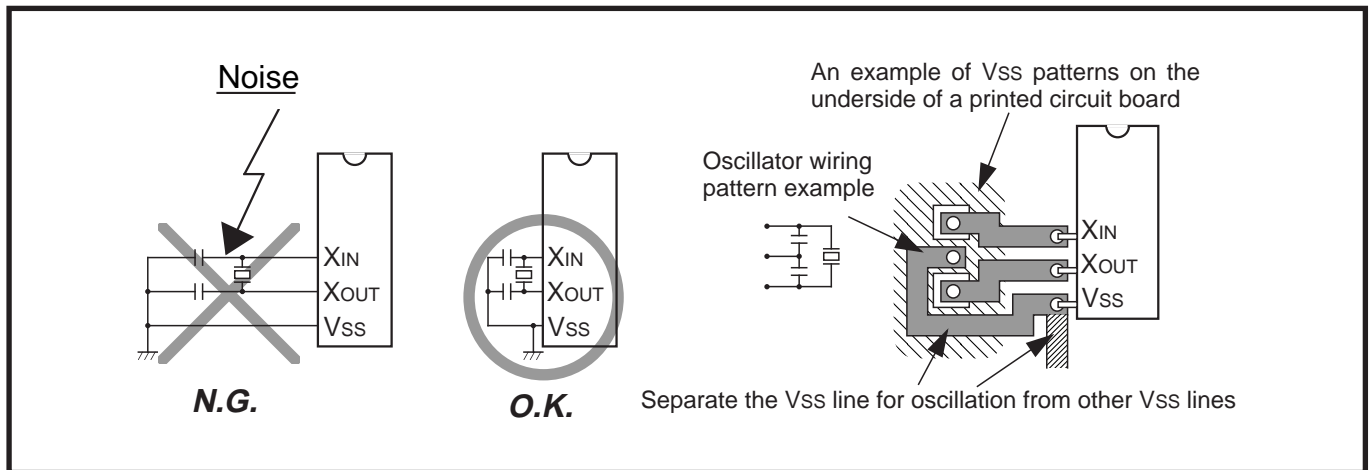


Fig. 3.4.2 Wiring for clock I/O pins

### (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.

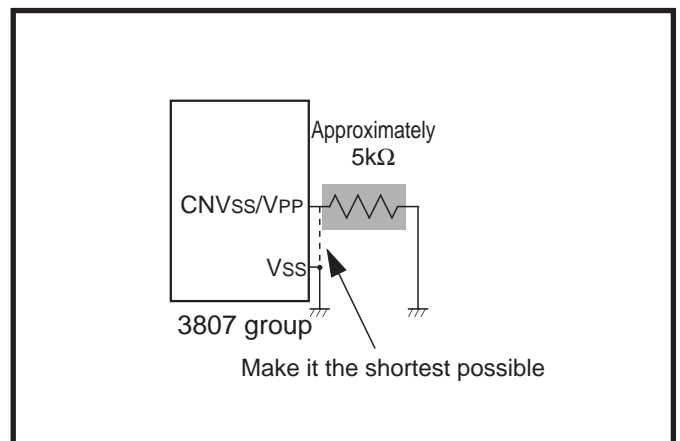


Fig. 3.4.3 Wiring for the VPP pin of the One Time PROM and the EPROM version

### 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.

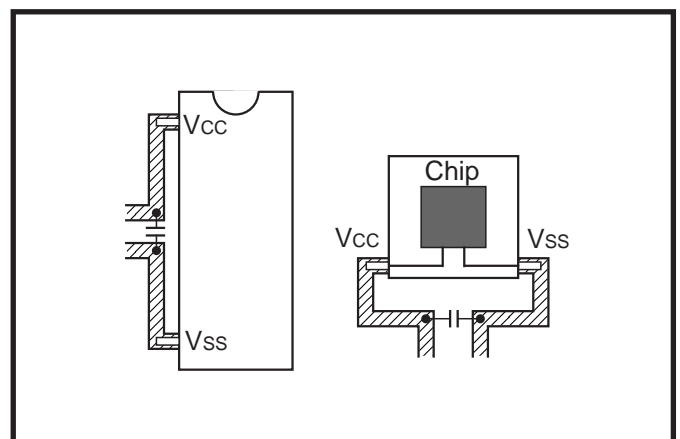


Fig. 3.4.4 Bypass capacitor across the Vss line and the Vcc line

## 3.4 Countermeasures against noise

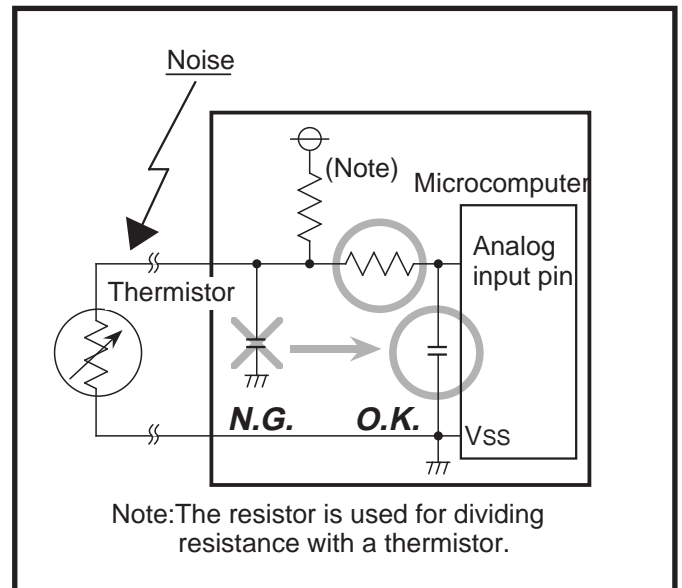
### 3.4.3 Wiring to analog input pins

- Connect an approximately 100 Ω to 1 kΩ resistor to an analog signal line which is connected to an analog input pin in series. Besides, connect the resistor to the microcomputer as close as possible.
- Connect an approximately 1000 pF capacitor across the Vss pin and the analog input pin. Besides, connect the capacitor to the Vss pin as close as possible. Also, connect the capacitor across the analog input pin and the Vss pin at equal length.

#### Reason

Signals which is input in an analog input pin (such as an A-D converter input pin) are usually output signals from sensor. The sensor which detects a change of event is installed far from the printed circuit board with a microcomputer, the wiring to an analog input pin is longer necessarily. This long wiring functions as an antenna which feeds noise into the microcomputer, which causes noise to an analog input pin.

If a capacitor between an analog input pin and the Vss pin is grounded at a position far away from the Vss pin, noise on the GND line may enter a microcomputer through the capacitor.



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

Fig.3.4.5 Analog signal line and a resistor and a capacitor

### 3.4.4. 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.

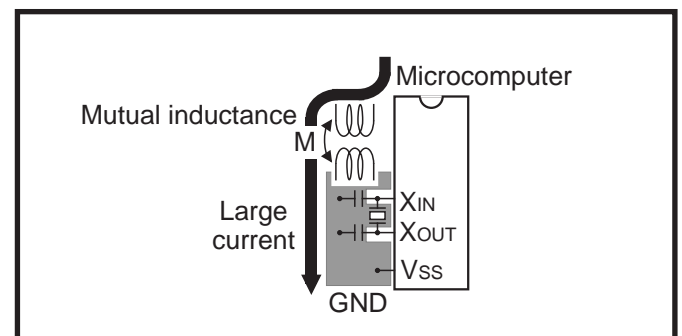


Fig.3.4.6 Wiring for a large current signal line

#### 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.

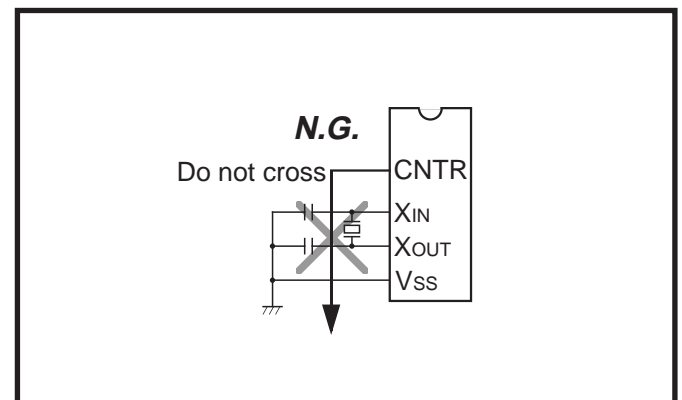


Fig.3.4.7 Wiring to a signal line where potential levels change frequently

# APPENDIX

## 3.4 Countermeasures against noise

### 3.4.5 Setup for I/O ports

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

<Hardware>

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

<Software>

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

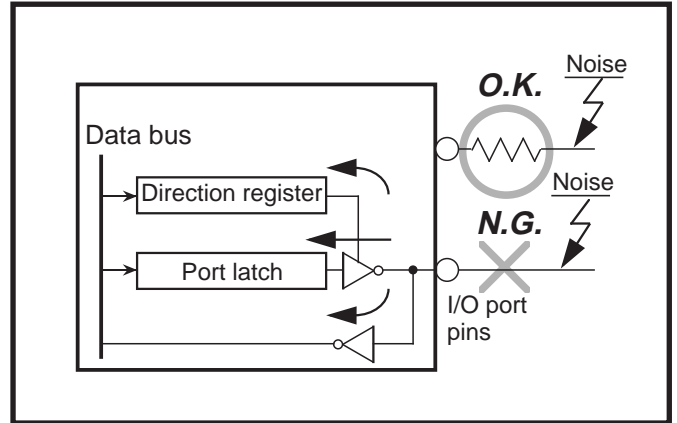


Fig. 3.4.8 Setup for I/O ports

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.

### 3.4.6 Providing of watchdog timer function by software

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

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

<The main routine>

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

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

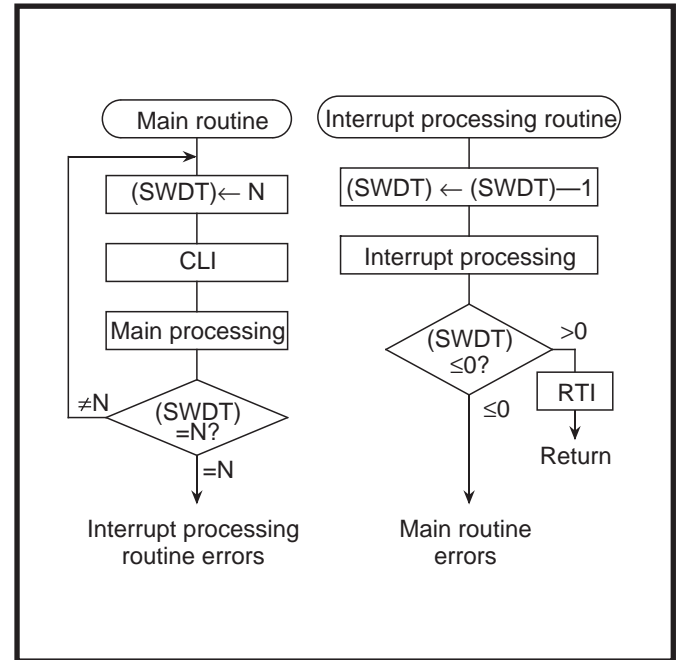


Fig. 3.4.9 Watchdog timer by software

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

## 3.4 Countermeasures against noise

---

<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 .

# APPENDIX

## 3.5 List of registers

### 3.5 List of registers

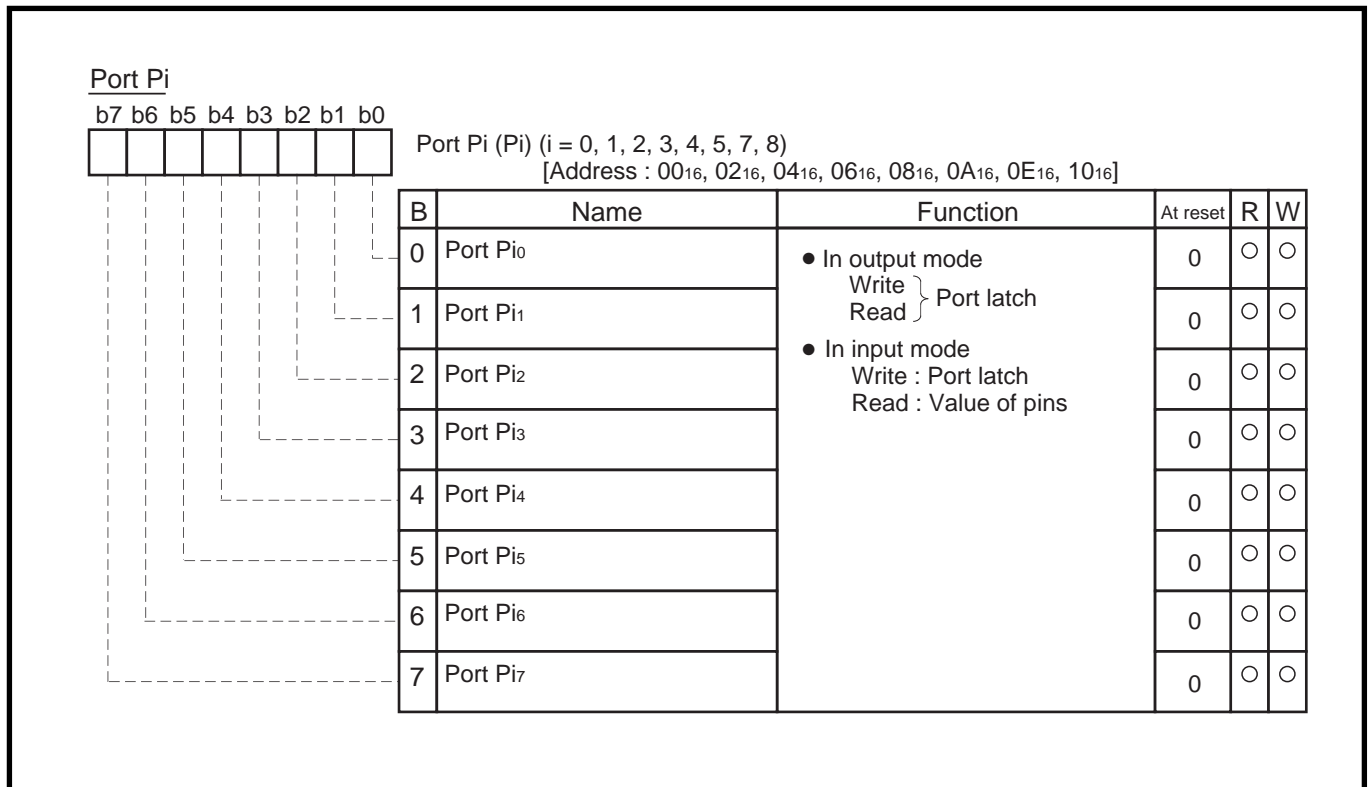


Fig. 3.5.1 Structure of Port Pi (i = 0, 1, 2, 3, 4, 5, 7, 8)

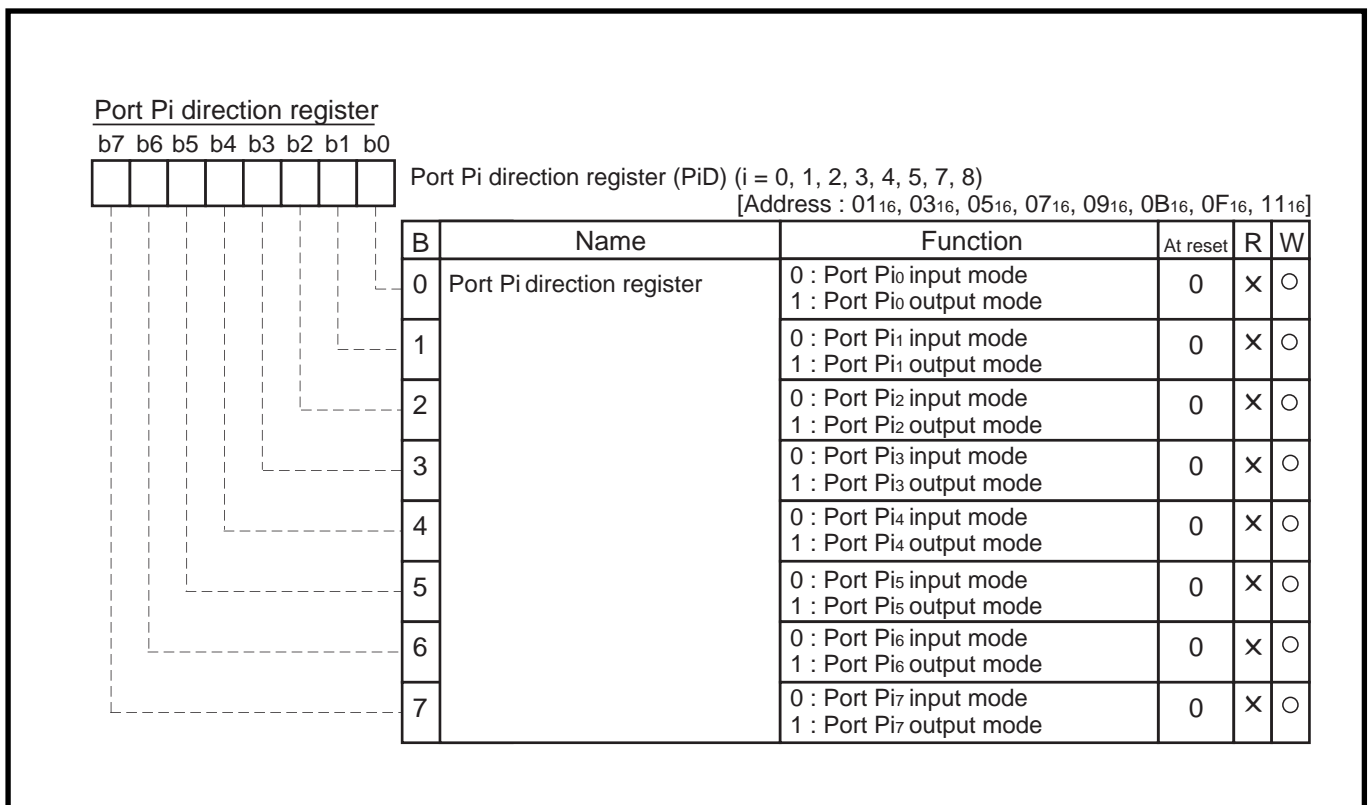


Fig. 3.5.2 Structure of Port Pi direction register (i = 0, 1, 2, 3, 4, 5, 7, 8)



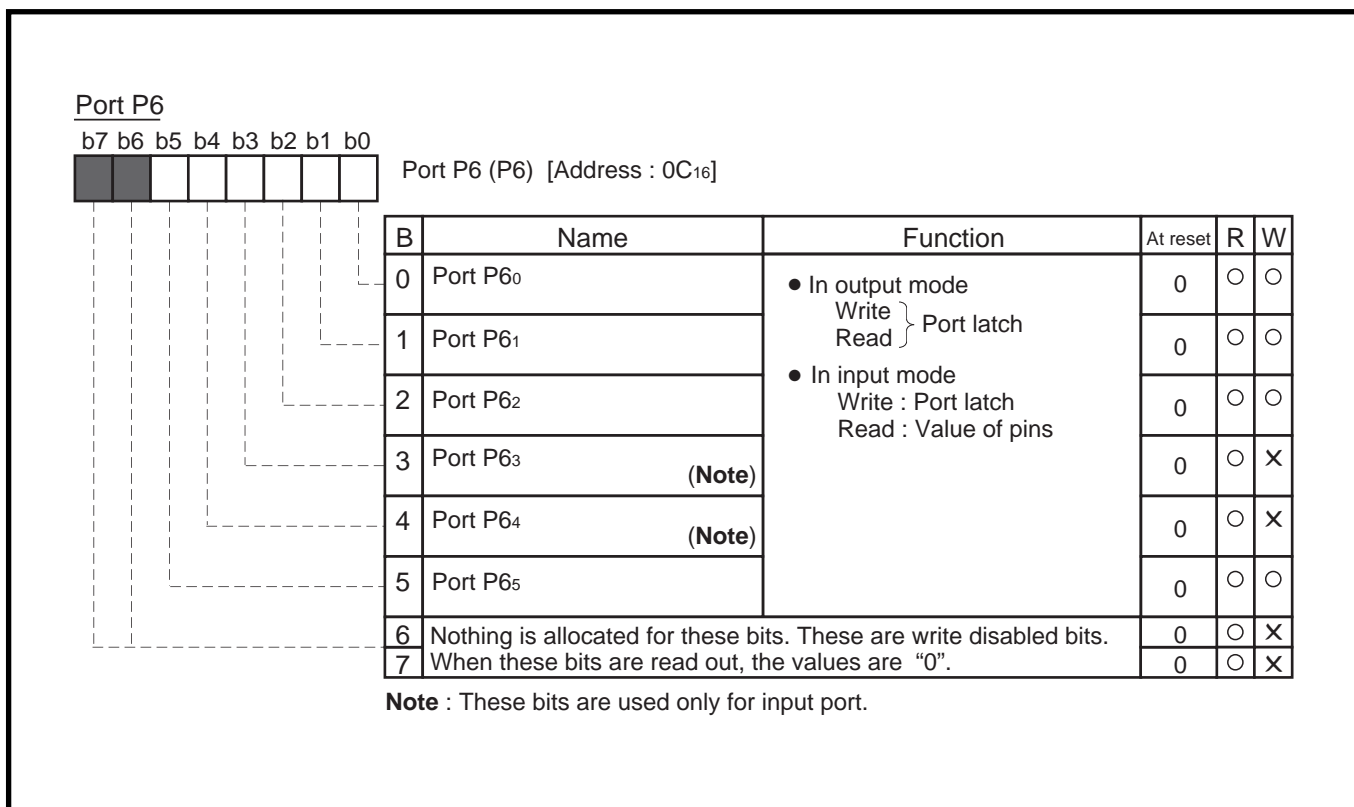


Fig. 3.5.3 Structure of Port P6

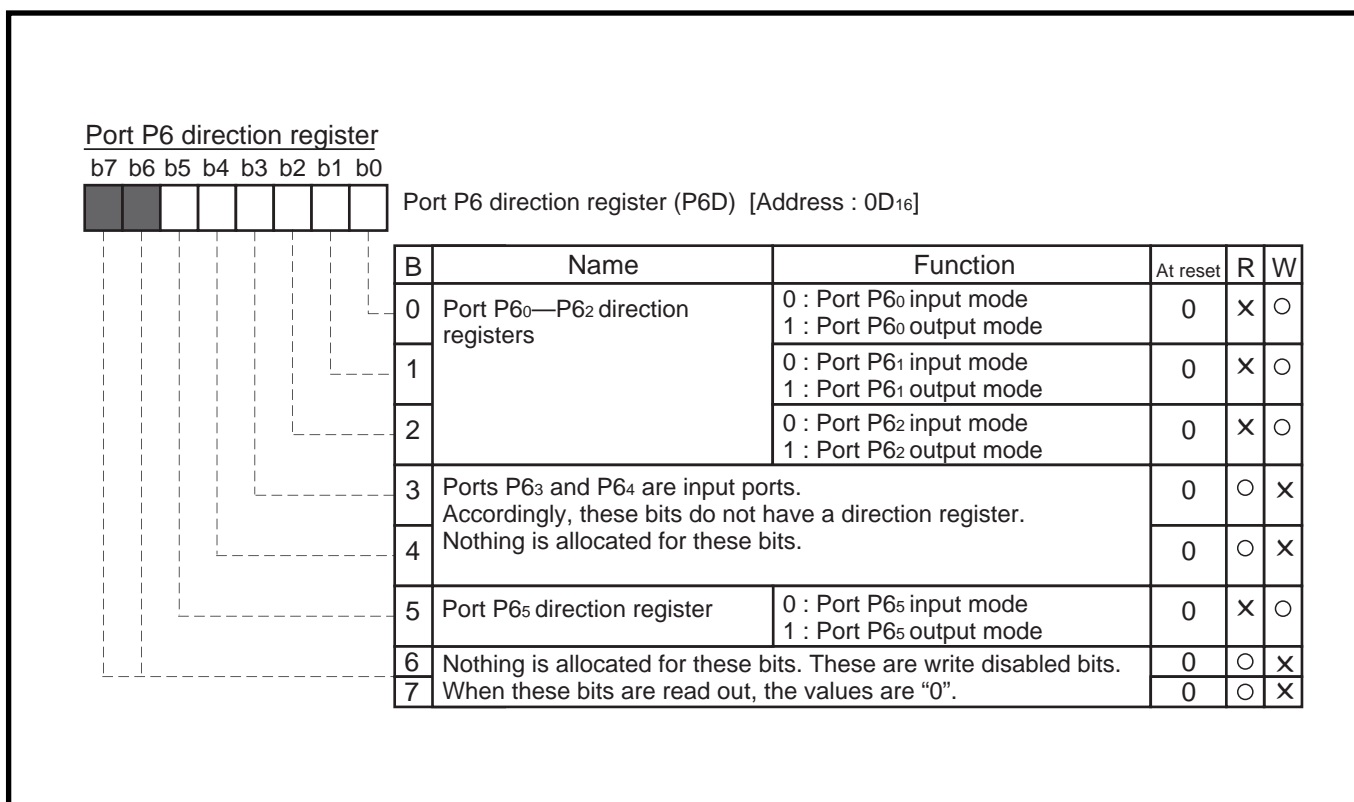


Fig. 3.5.4 Structure of Port P6 direction register

# APPENDIX

## 3.5 List of registers

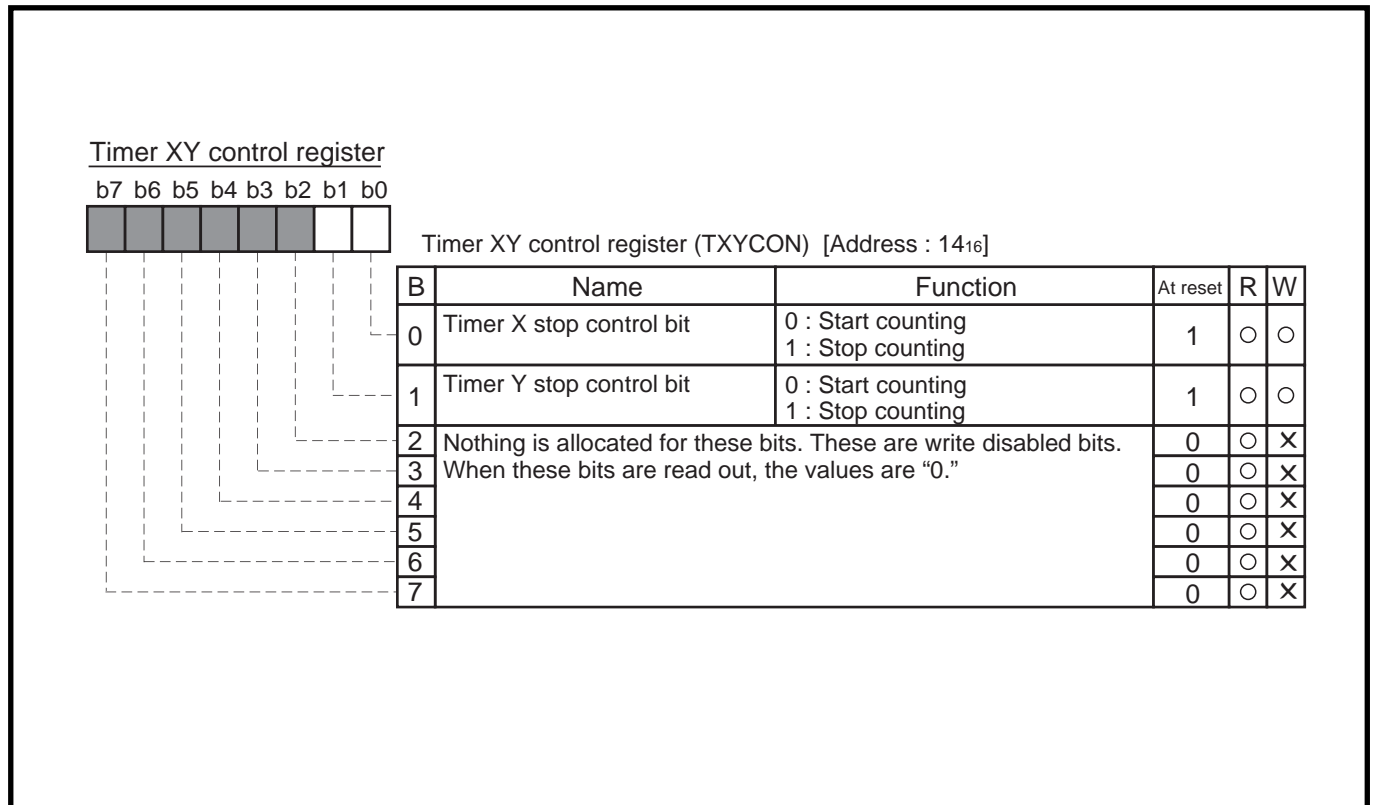


Fig. 3.5.5 Structure of Timer XY control register

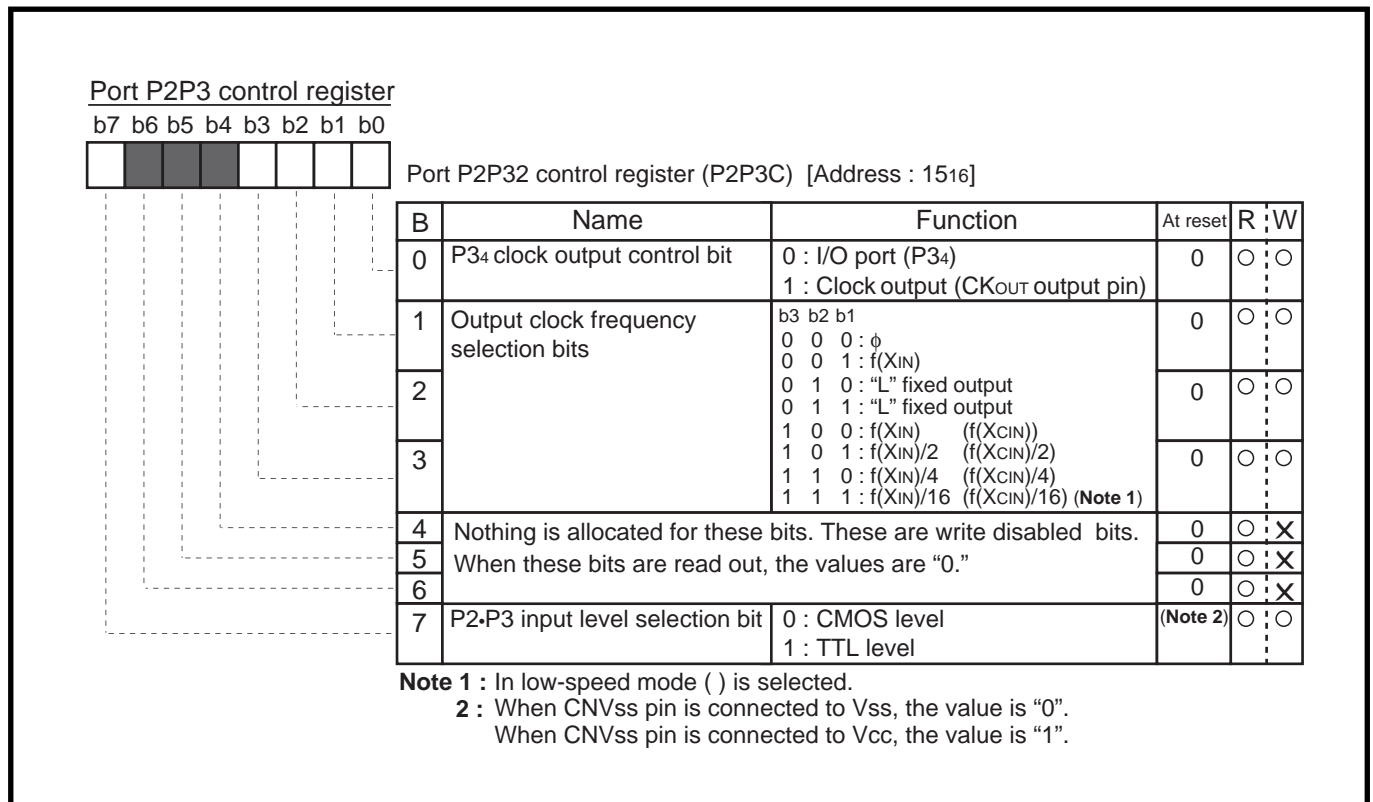


Fig. 3.5.6 Structure of Port P2P3 control register

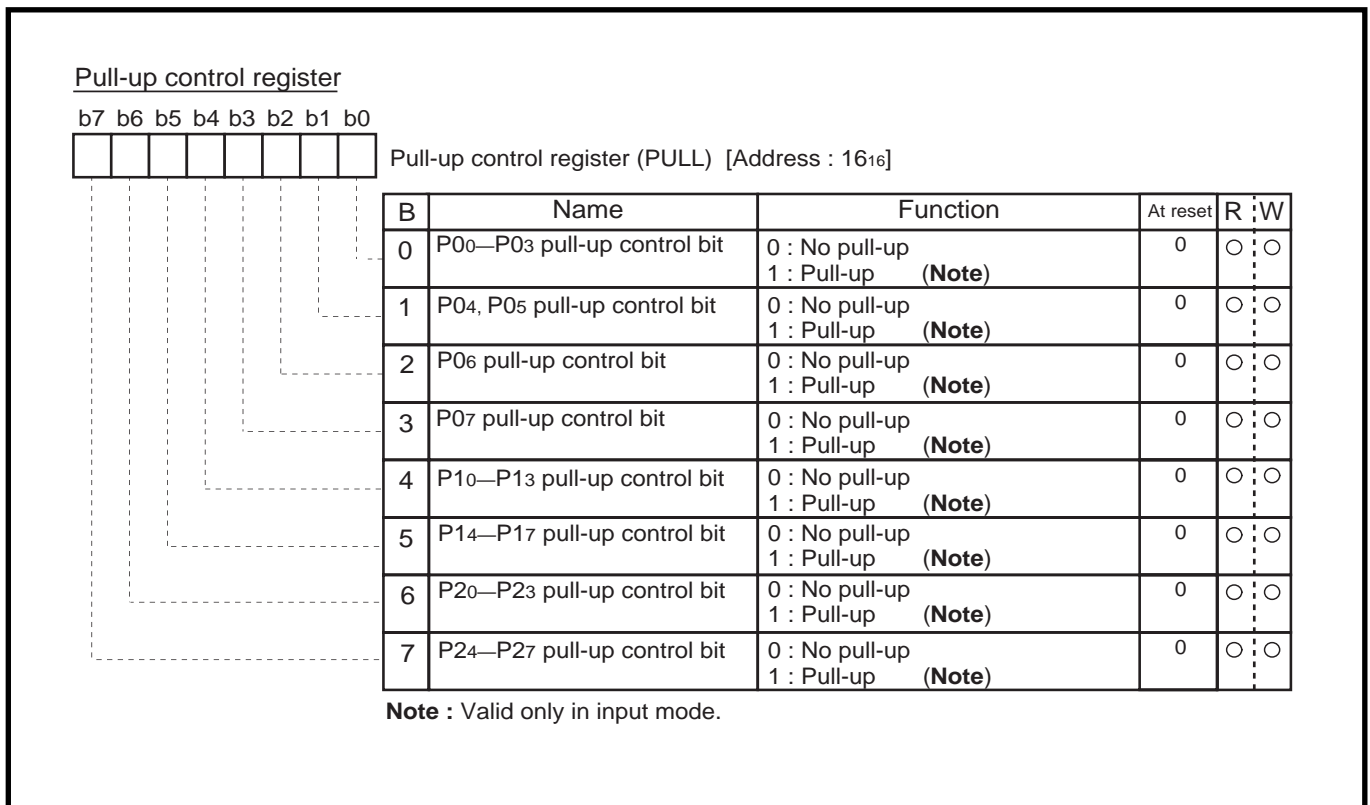


Fig. 3.5.7 Structure of Pull-up control register

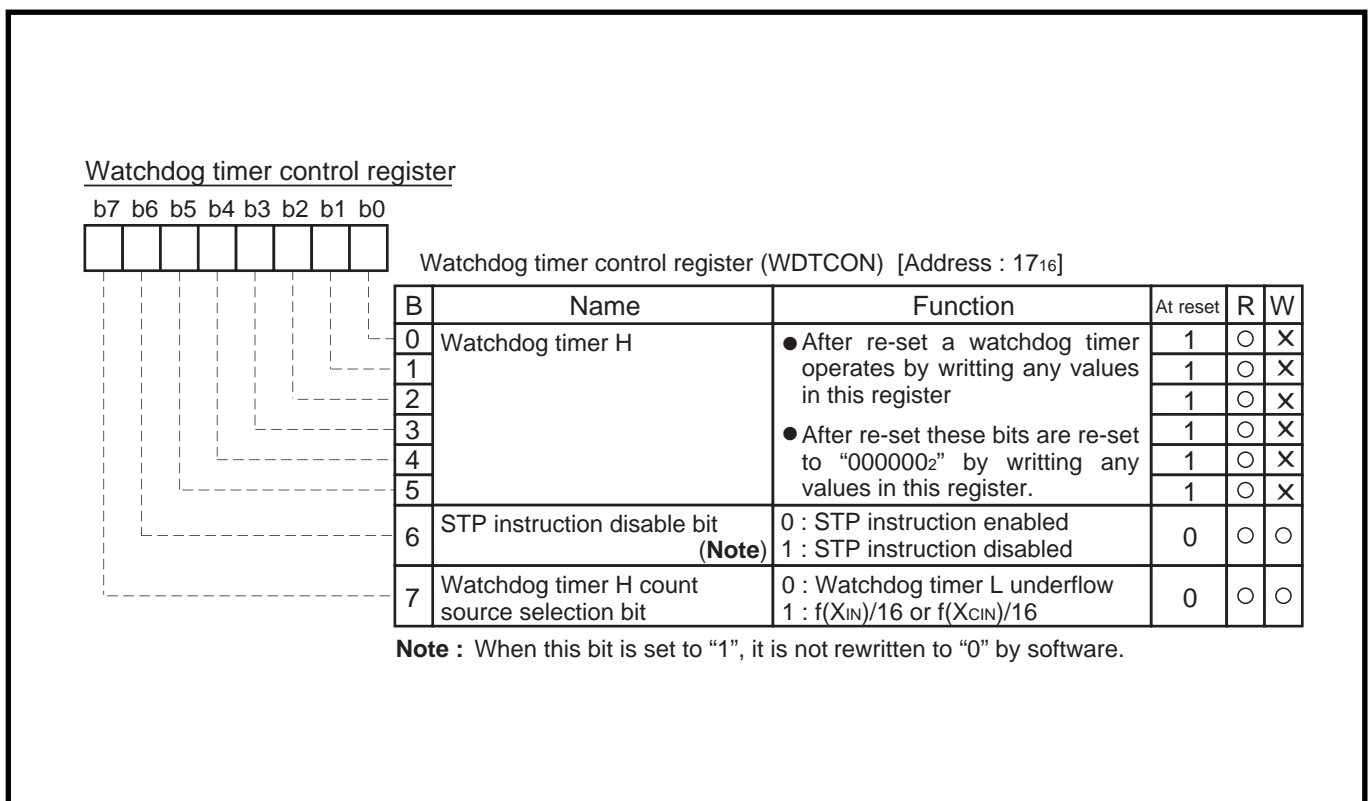


Fig. 3.5.8 Structure of Watchdog timer control register

# APPENDIX

## 3.5 List of registers

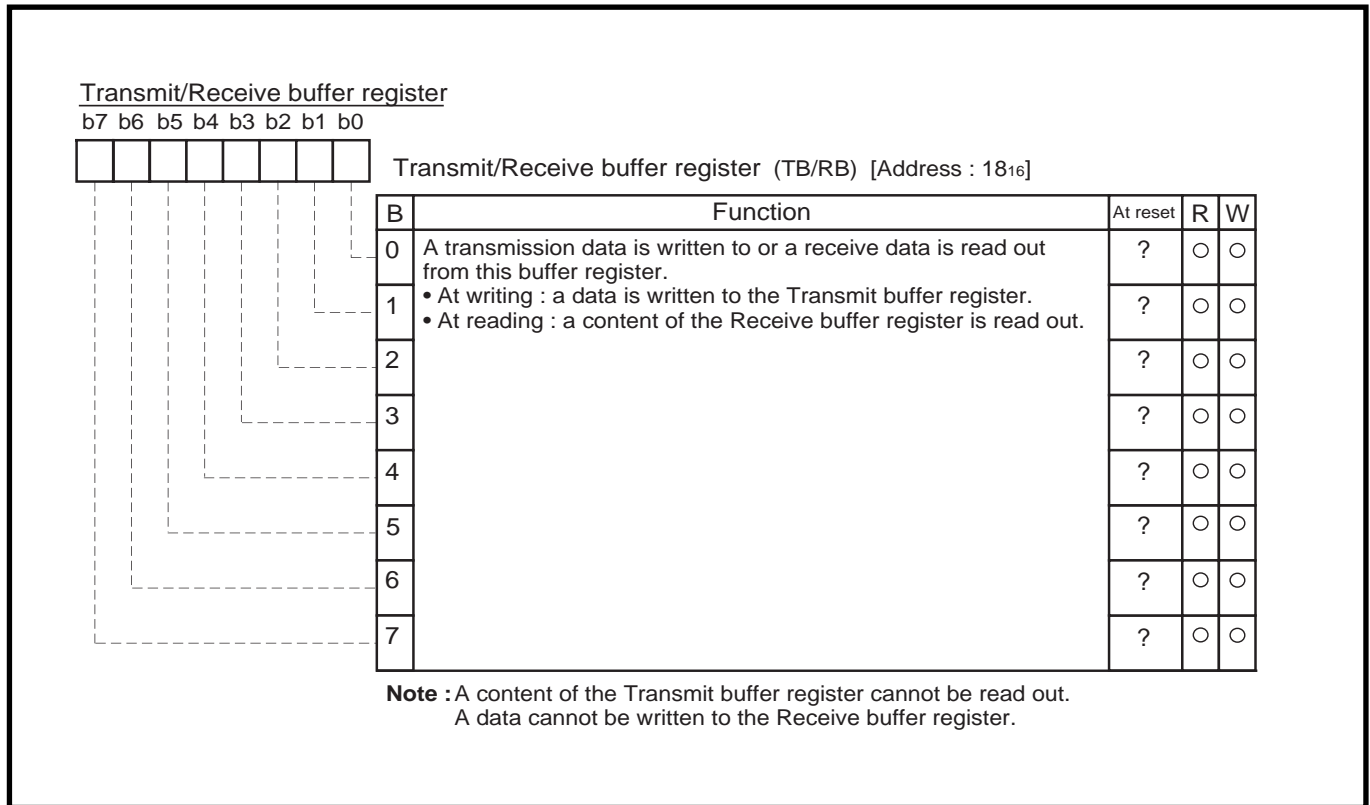


Fig. 3.5.9 Structure of Transmit/Receive buffer register

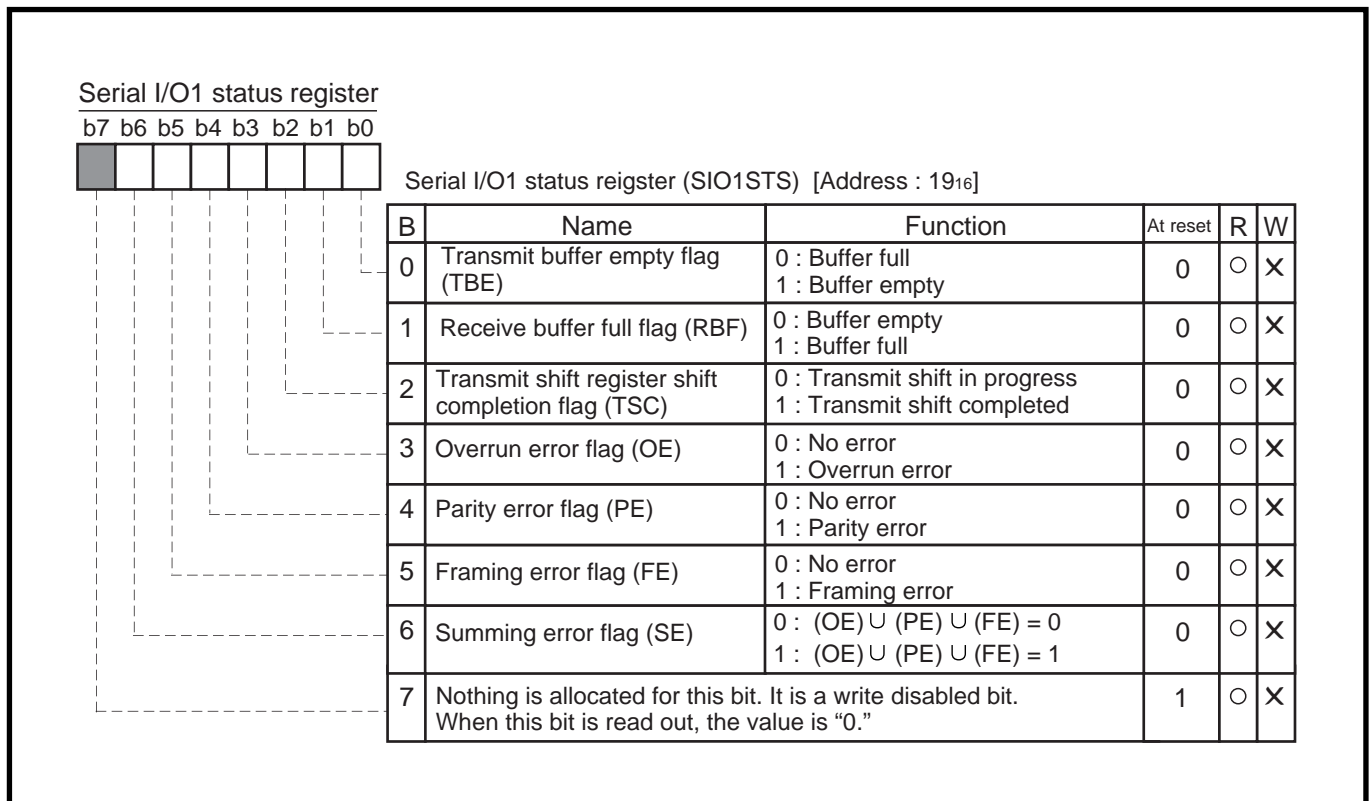


Fig. 3.5.10 Structure of Serial I/O1 status register

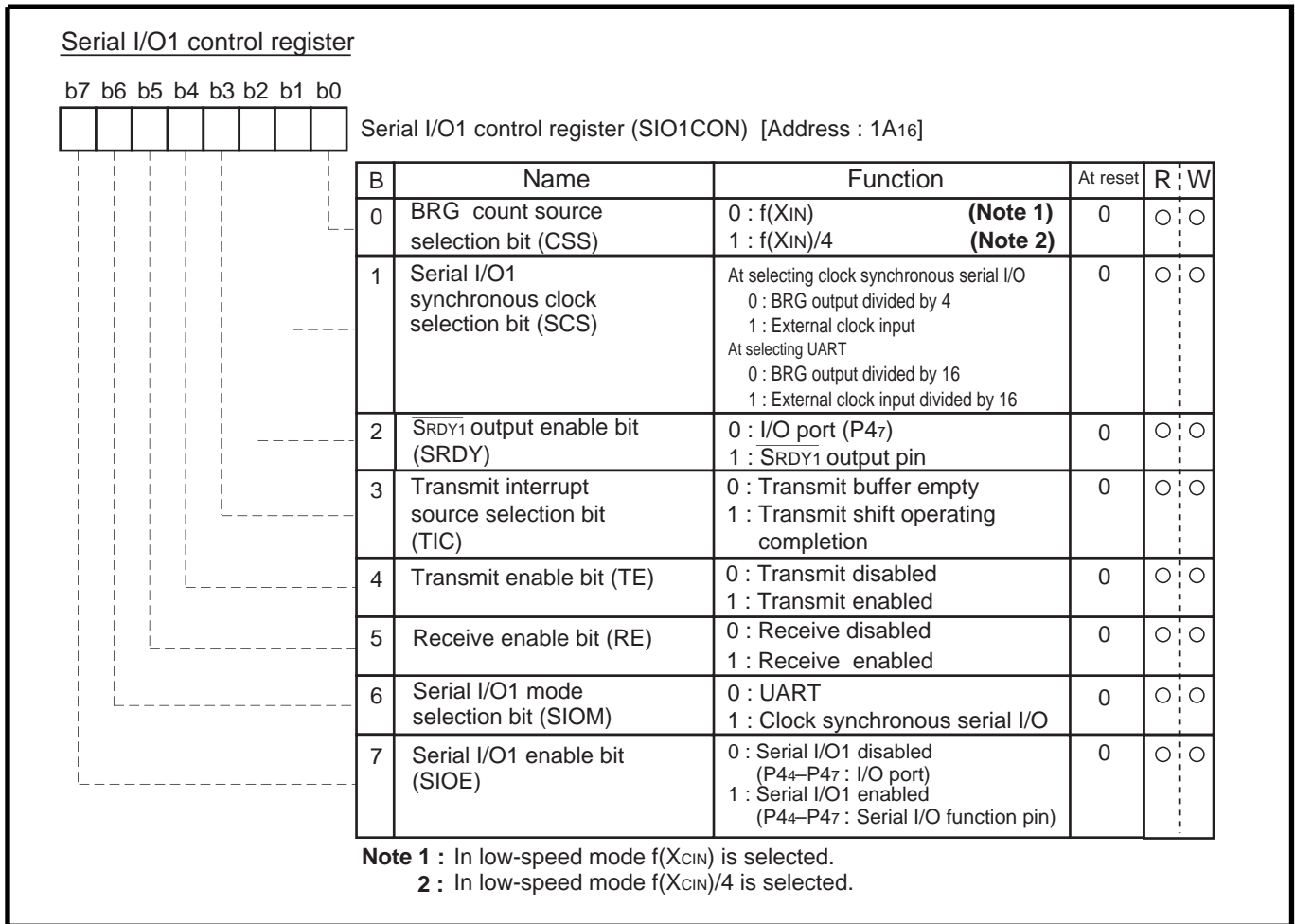


Fig. 3.5.11 Structure of Serial I/O1 control register

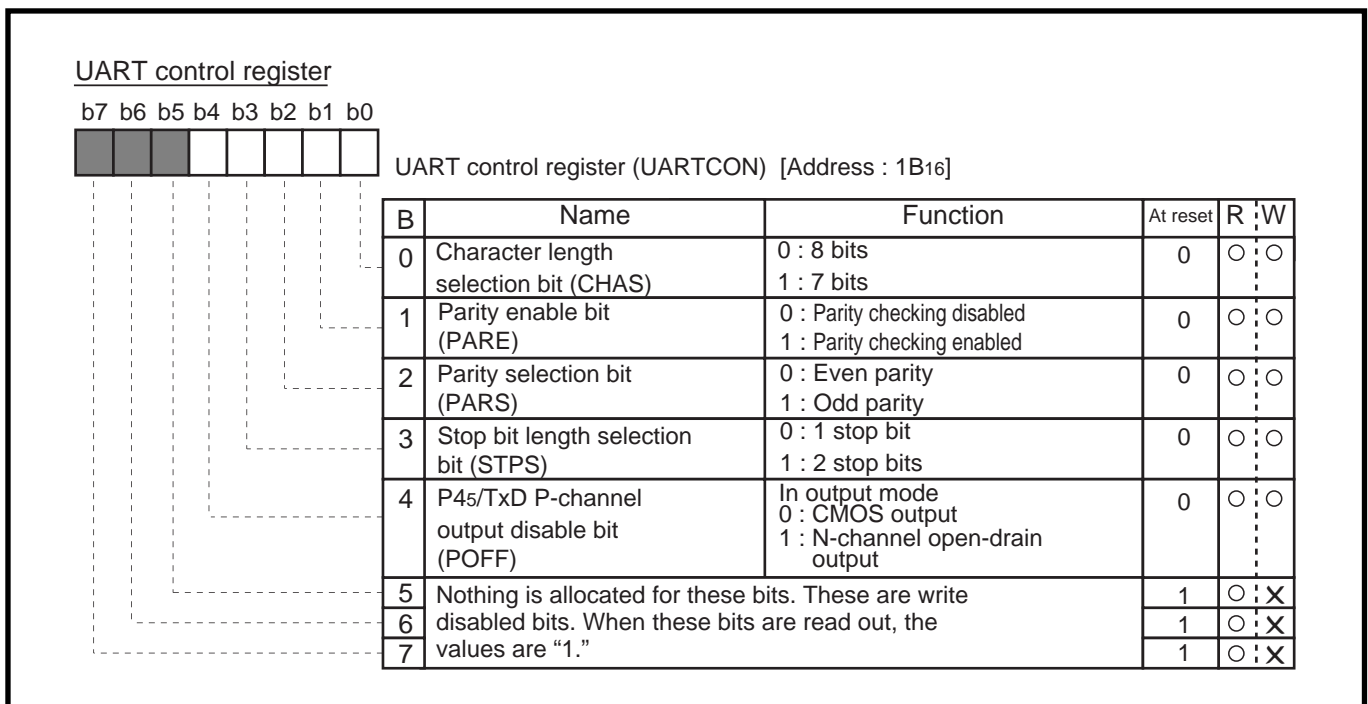


Fig. 3.5.12 Structure of UART control register

# APPENDIX

## 3.5 List of registers

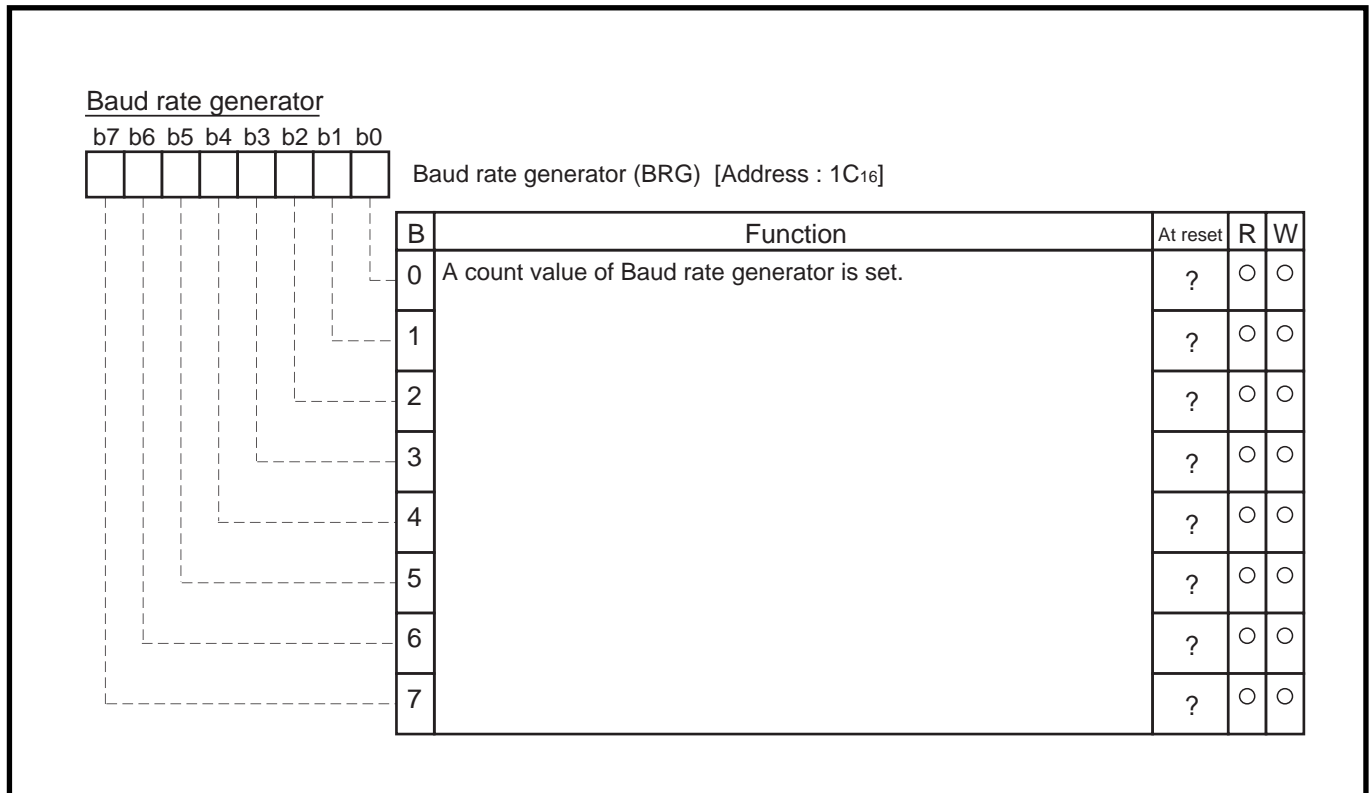


Fig. 3.5.13 Structure of Baud rate generator

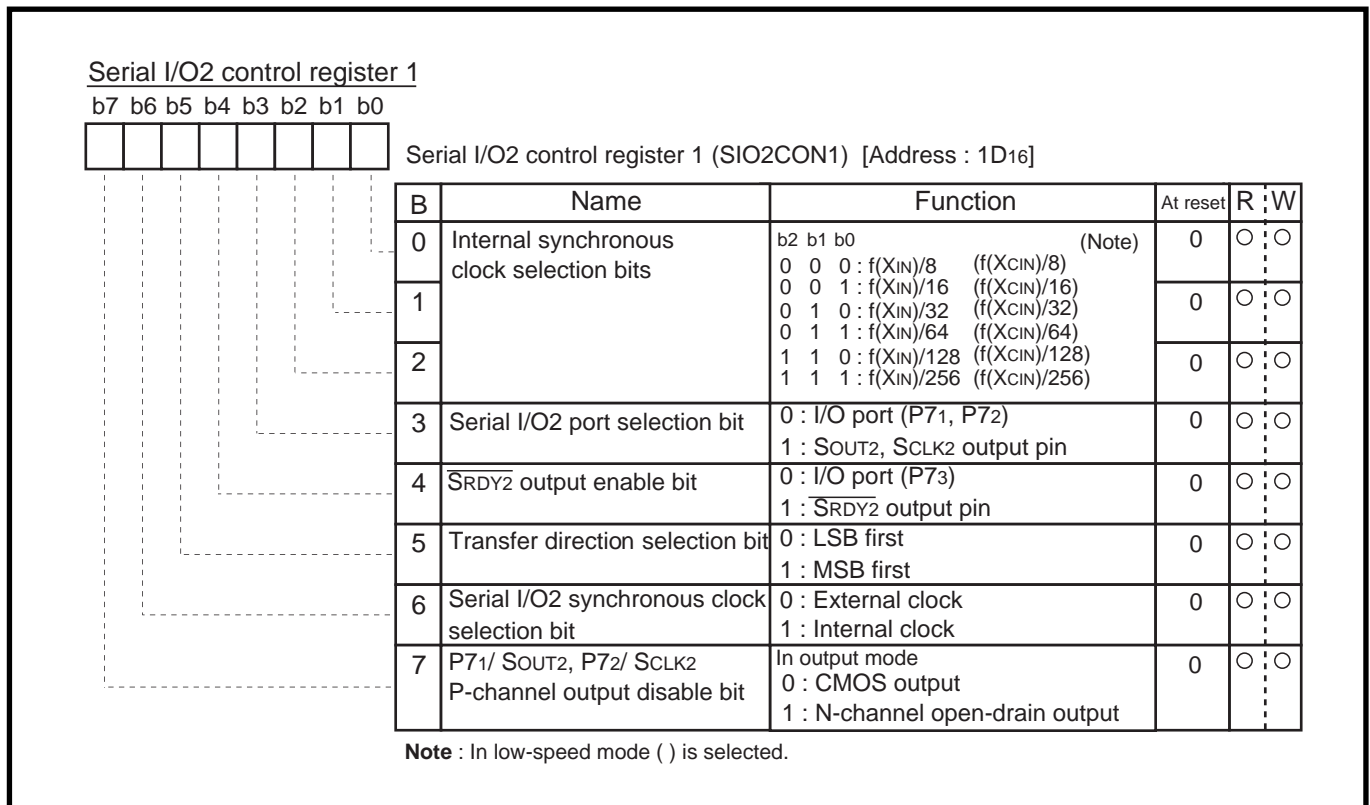


Fig. 3.5.14 Structure of Serial I/O2 control register 1

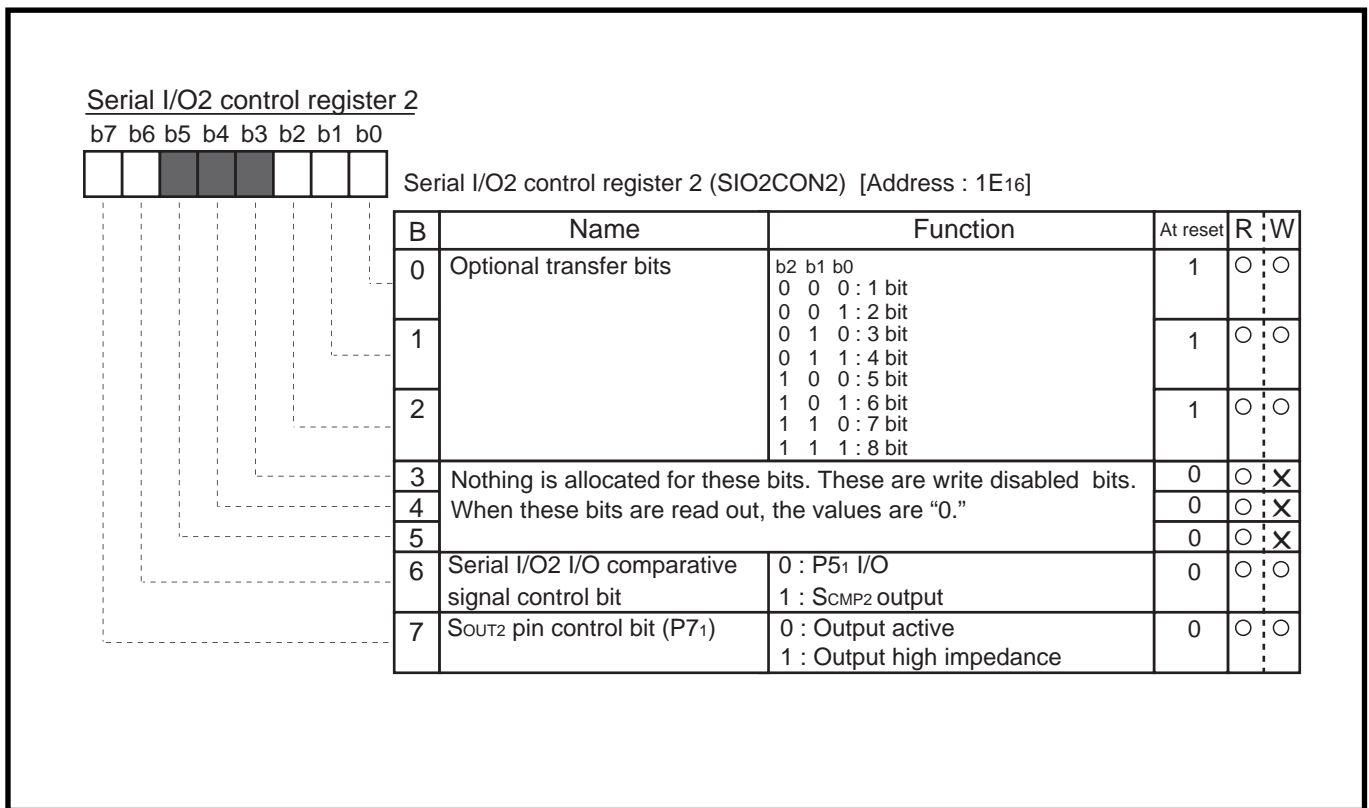


Fig. 3.5.15 Structure of Serial I/O2 control register 2

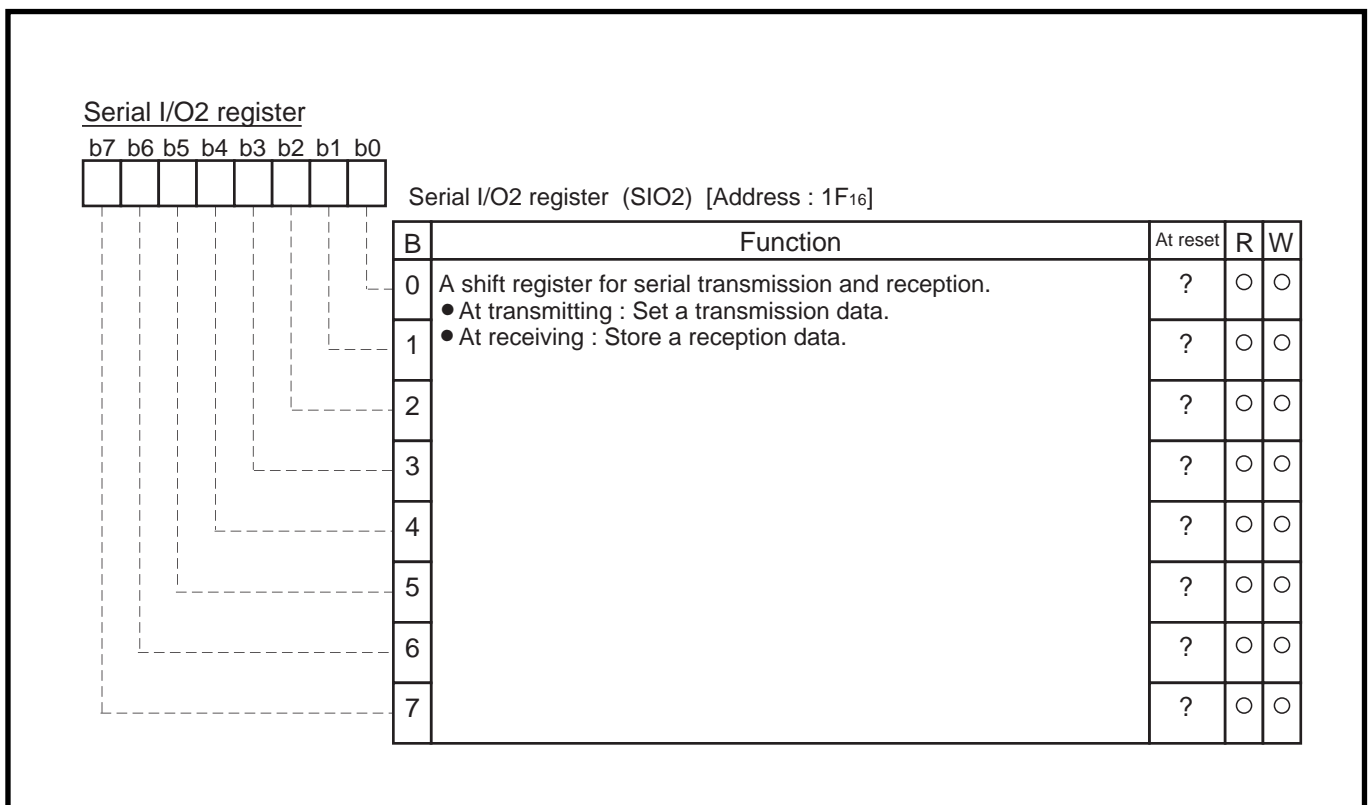


Fig. 3.5.16 Structure of Serial I/O2 register

# APPENDIX

## 3.5 List of registers

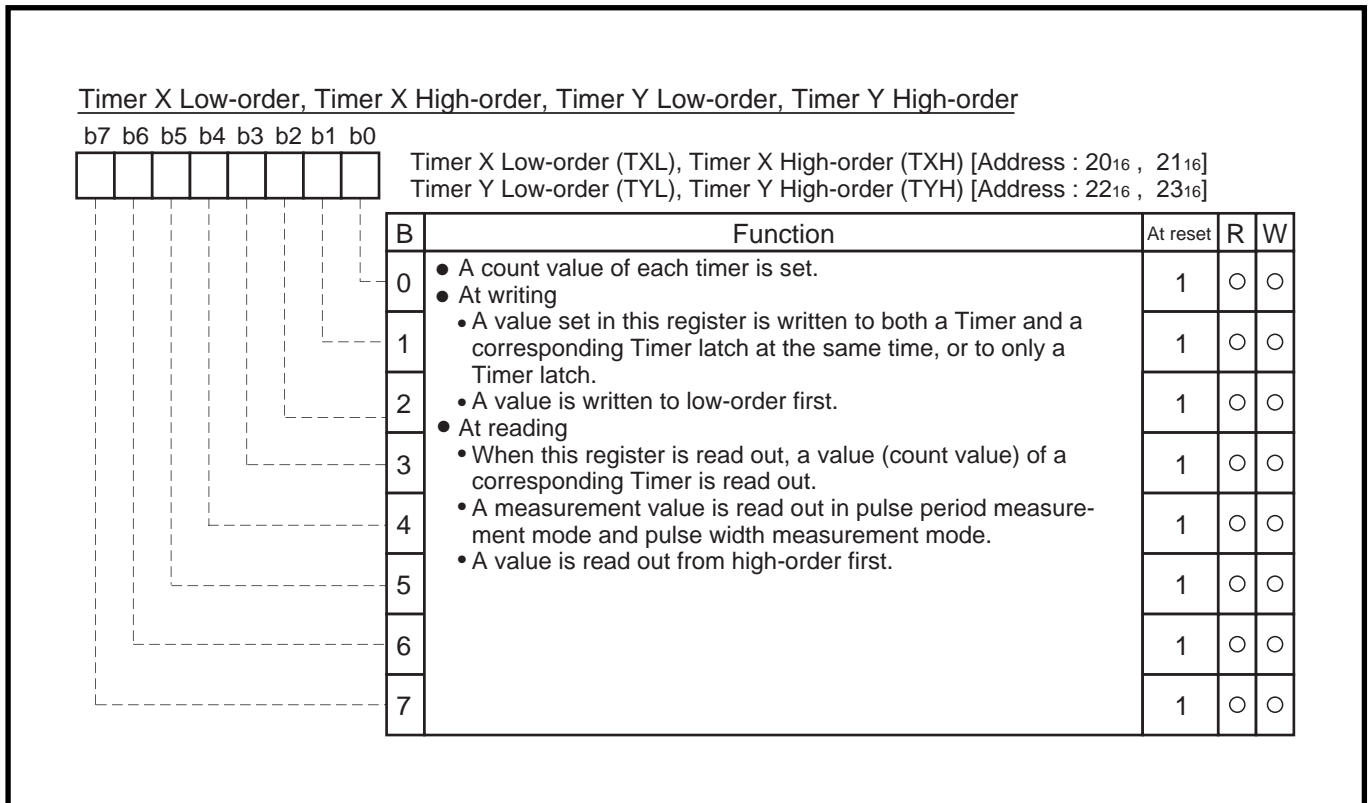


Fig. 3.5.17 Structure of Timer X Low-order, Timer X High-order, Timer Y Low-order, Timer Y High-order

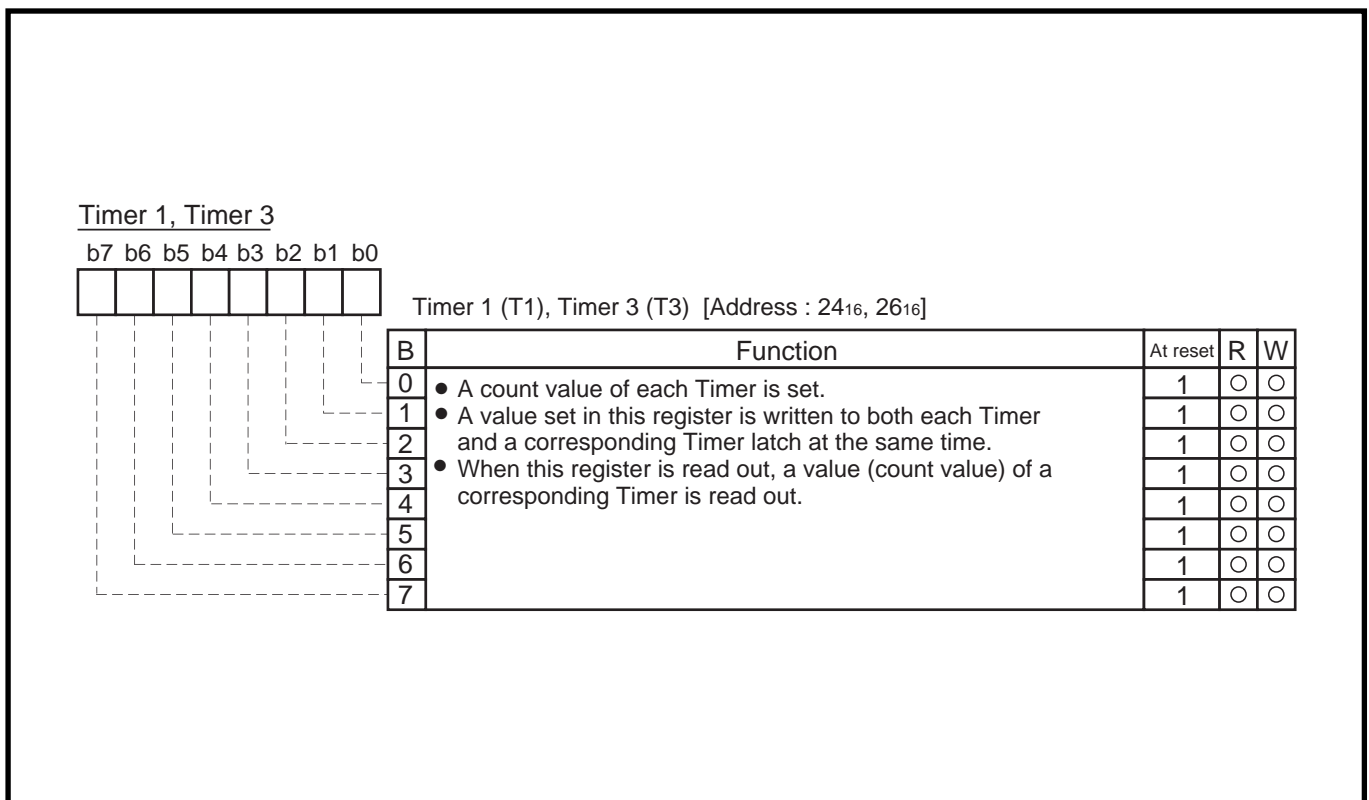


Fig. 3.5.18 Structure of Timer 1, Timer 3



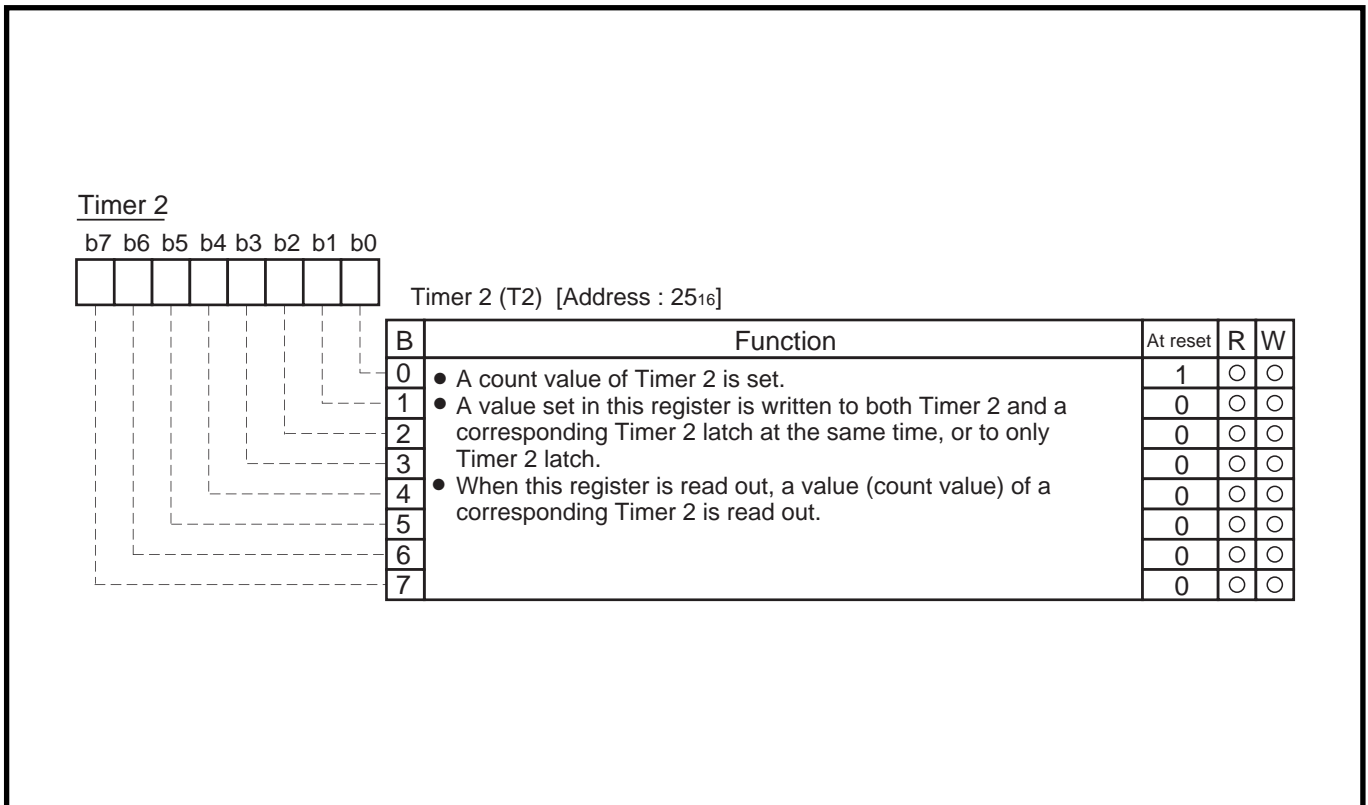


Fig. 3.5.19 Structure of Timer 2

# APPENDIX

## 3.5 List of registers

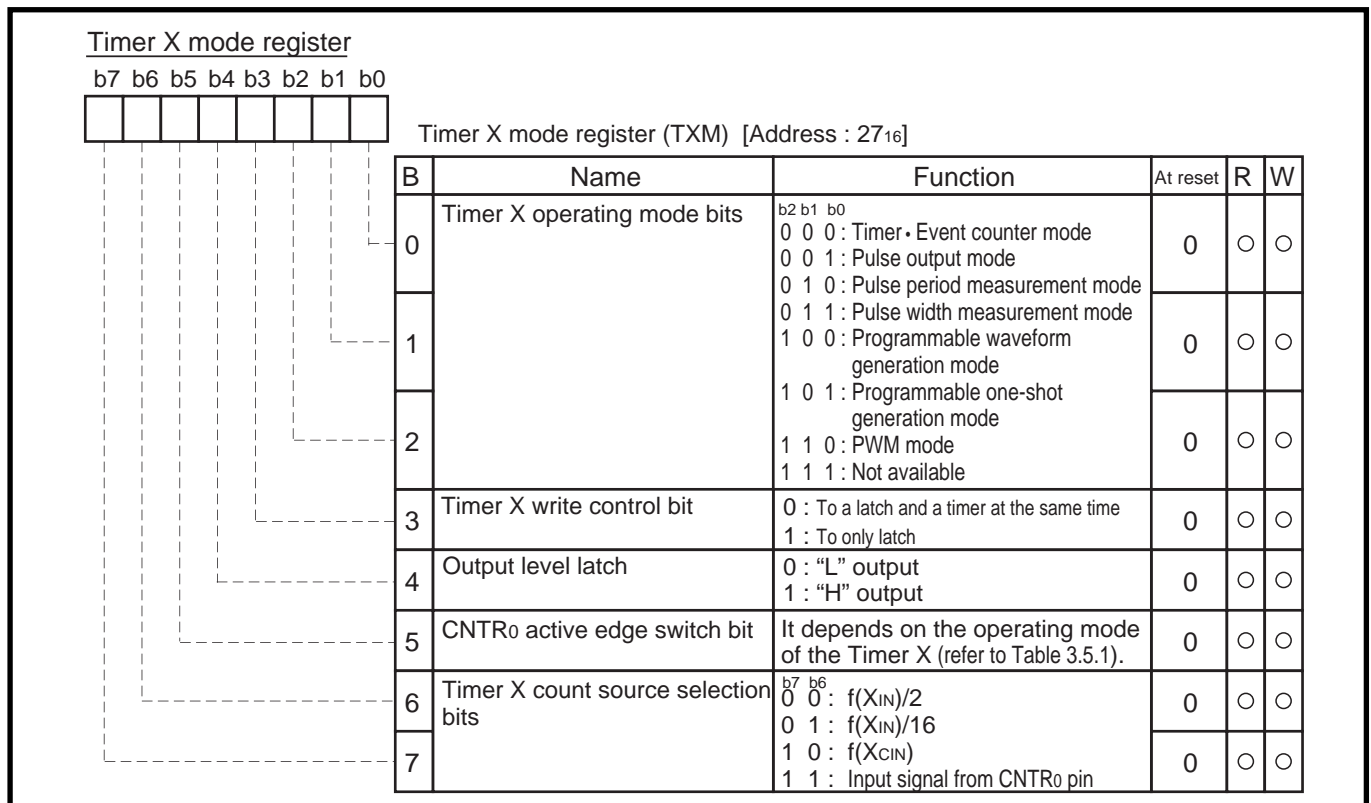


Fig. 3.5.20 Structure of Timer X mode register

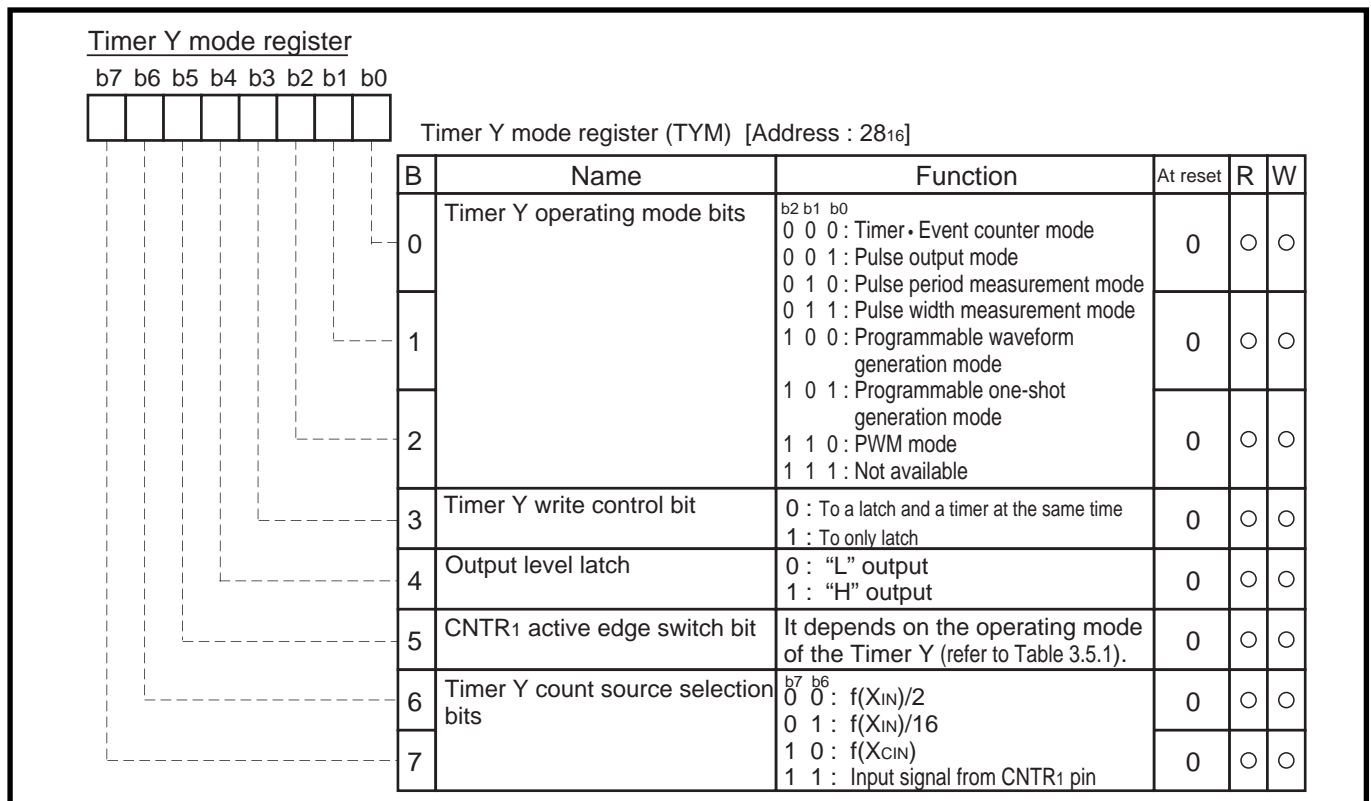


Fig. 3.5.21 Structure of Timer Y mode register

**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 (bit 5 of each address 27 <sub>16</sub> and 28 <sub>16</sub> )	
Timer mode	"0"	<ul style="list-style-type: none"> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge (No effect on timer count)</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge (No effect on timer count)</li> </ul>
Event counter mode	"0"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Count at rising edge</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Count at falling edge</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge</li> </ul>
Pulse output mode	"0"	<ul style="list-style-type: none"> <li>• Start of pulse output : From "H" level</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Start of pulse output : From "L" level</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge</li> </ul>
Pulse period measurement mode	"0"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Measurement of a period between a falling edge and the next falling edge</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Measurement of a period between a rising edge and the next rising edge</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge</li> </ul>
Pulse width measurement mode	"0"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Measurement of "H" level width</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Measurement of "L" level width</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge</li> </ul>
Programmable one-shot generation mode	"0"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Start of a pulse output at "L" level, and output of an one-shot "H" level pulse</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Falling edge</li> </ul>
	"1"	<ul style="list-style-type: none"> <li>• Timer X/Timer Y : Start of a pulse output at "H" level, and output of an one-shot "L" level pulse</li> <li>• Generation of CNTR0/CNTR1 interrupt request : Rising edge</li> </ul>

# APPENDIX

## 3.5 List of registers

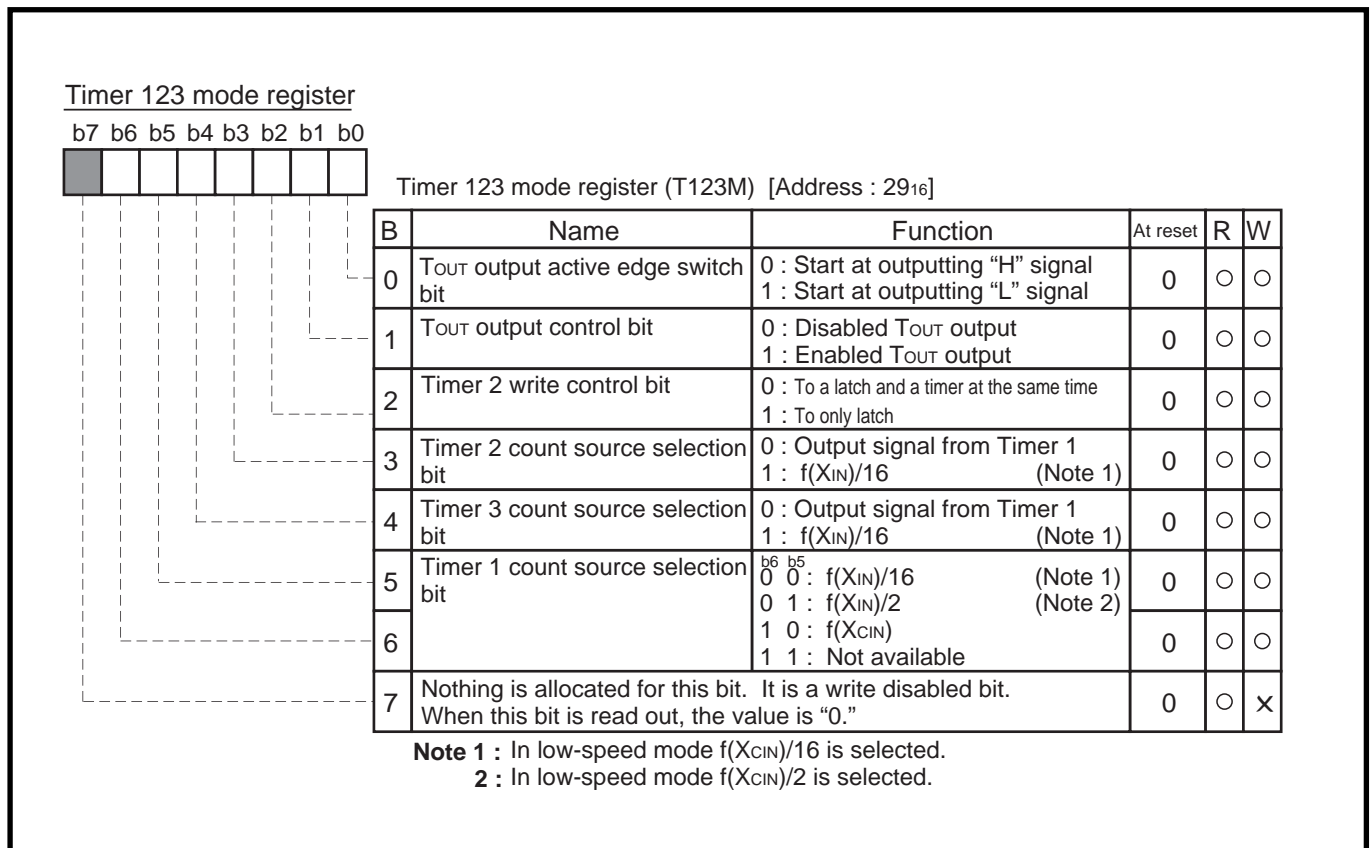


Fig. 3.5.22 Structure of Timer 123 mode register

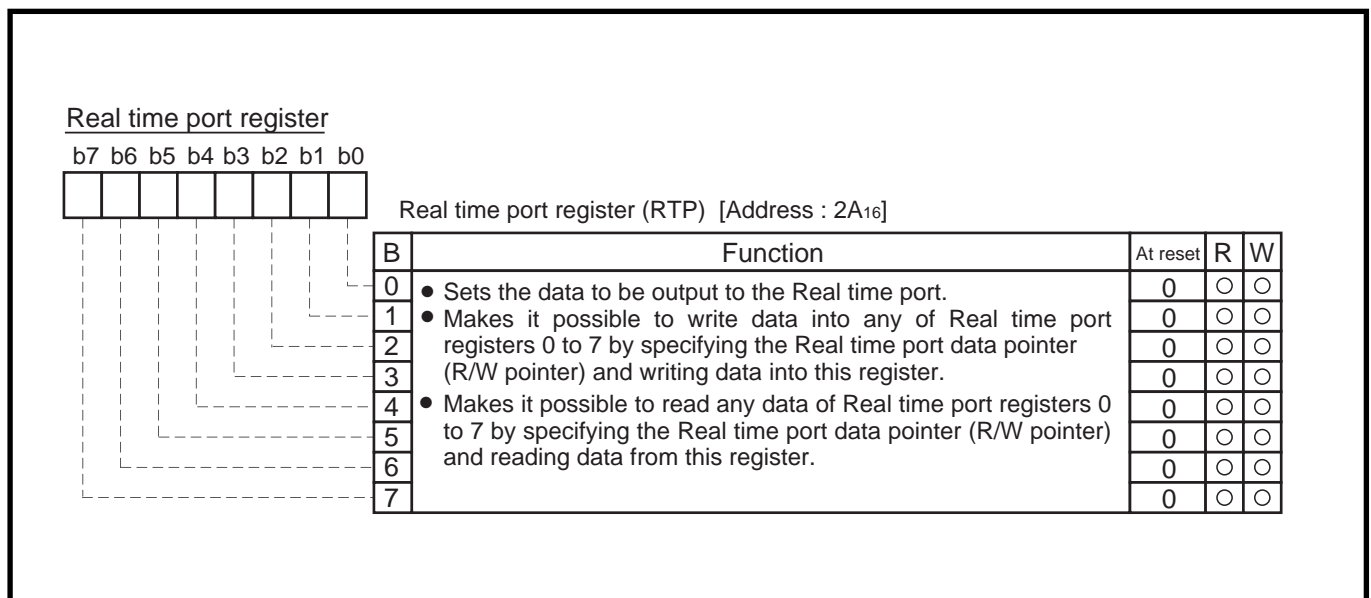


Fig. 3.5.23 Structure of Real time port register

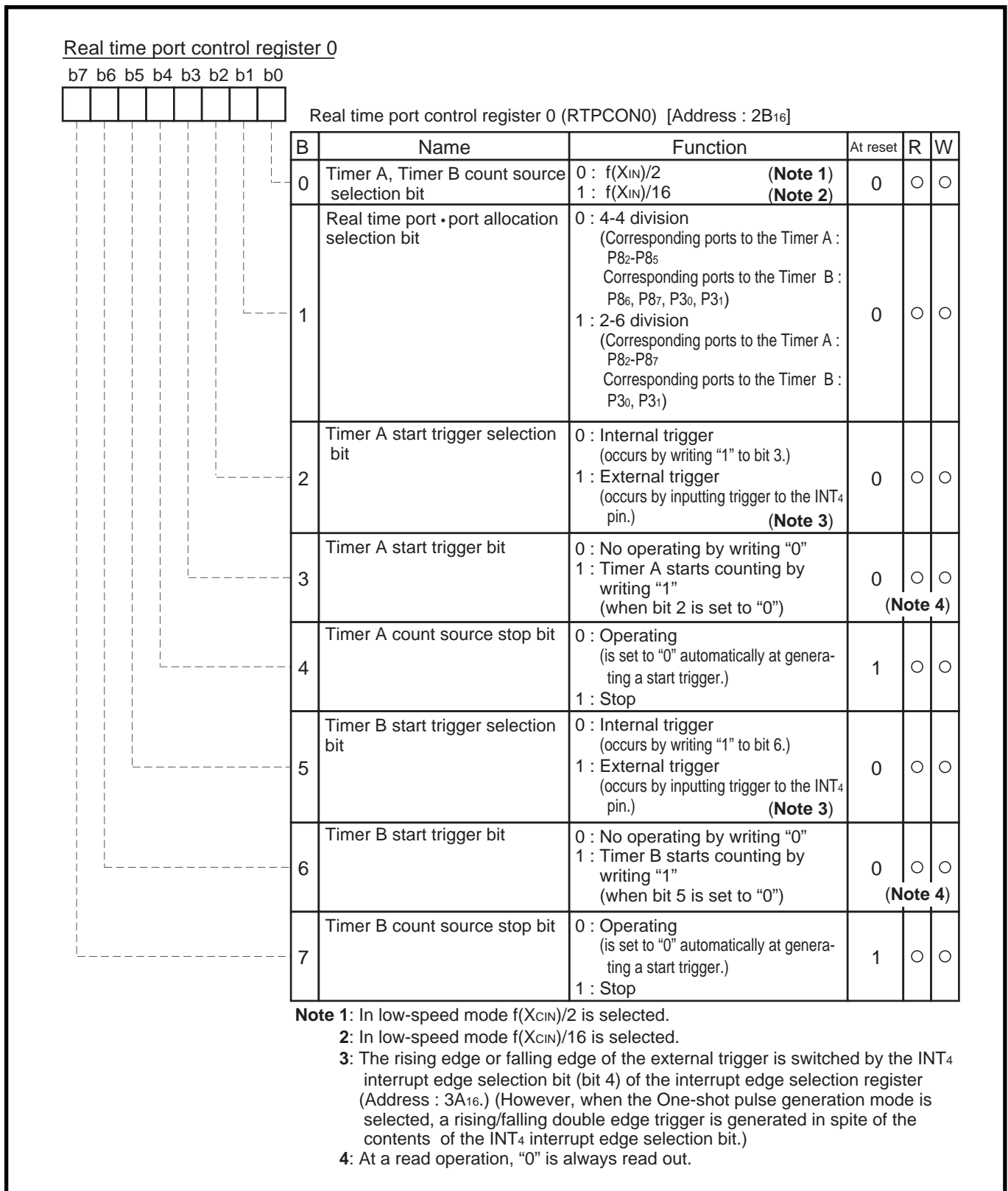


Fig. 3.5.24 Structure of Real time port control register 0

# APPENDIX

## 3.5 List of registers

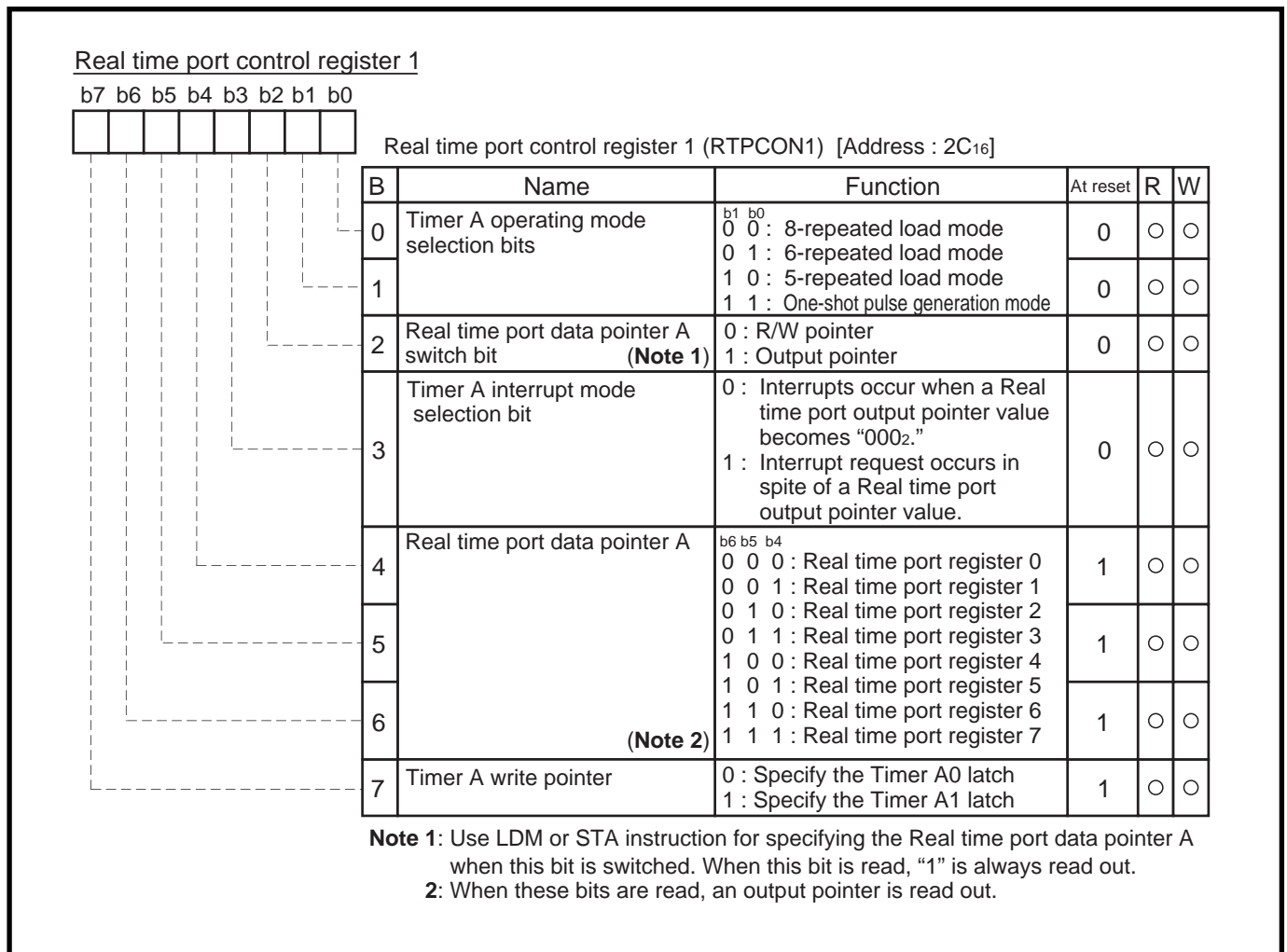


Fig. 3.5.25 Structure of Real time port control register 1

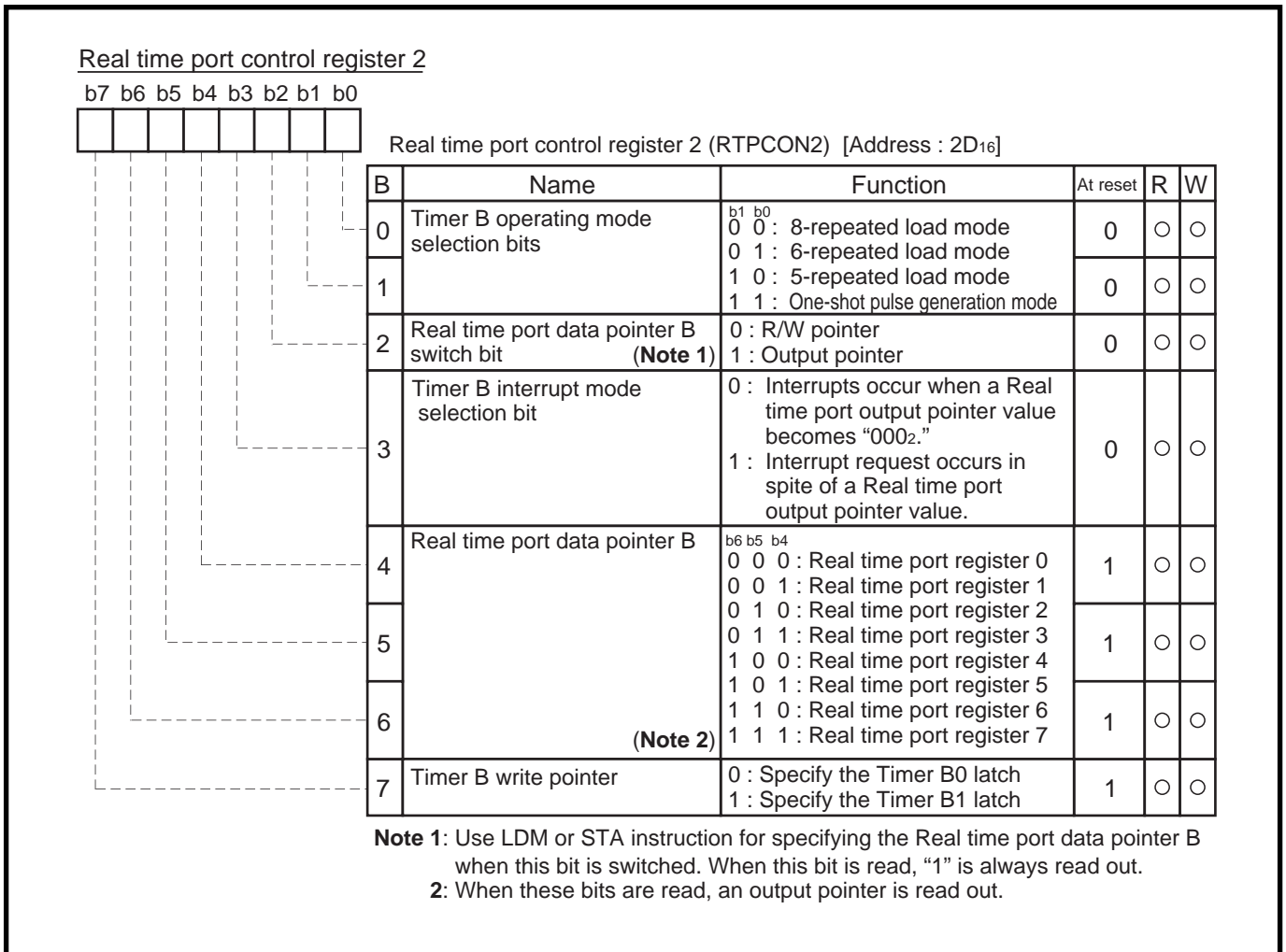


Fig. 3.5.26 Structure of Real time port control register 2

# APPENDIX

## 3.5 List of registers

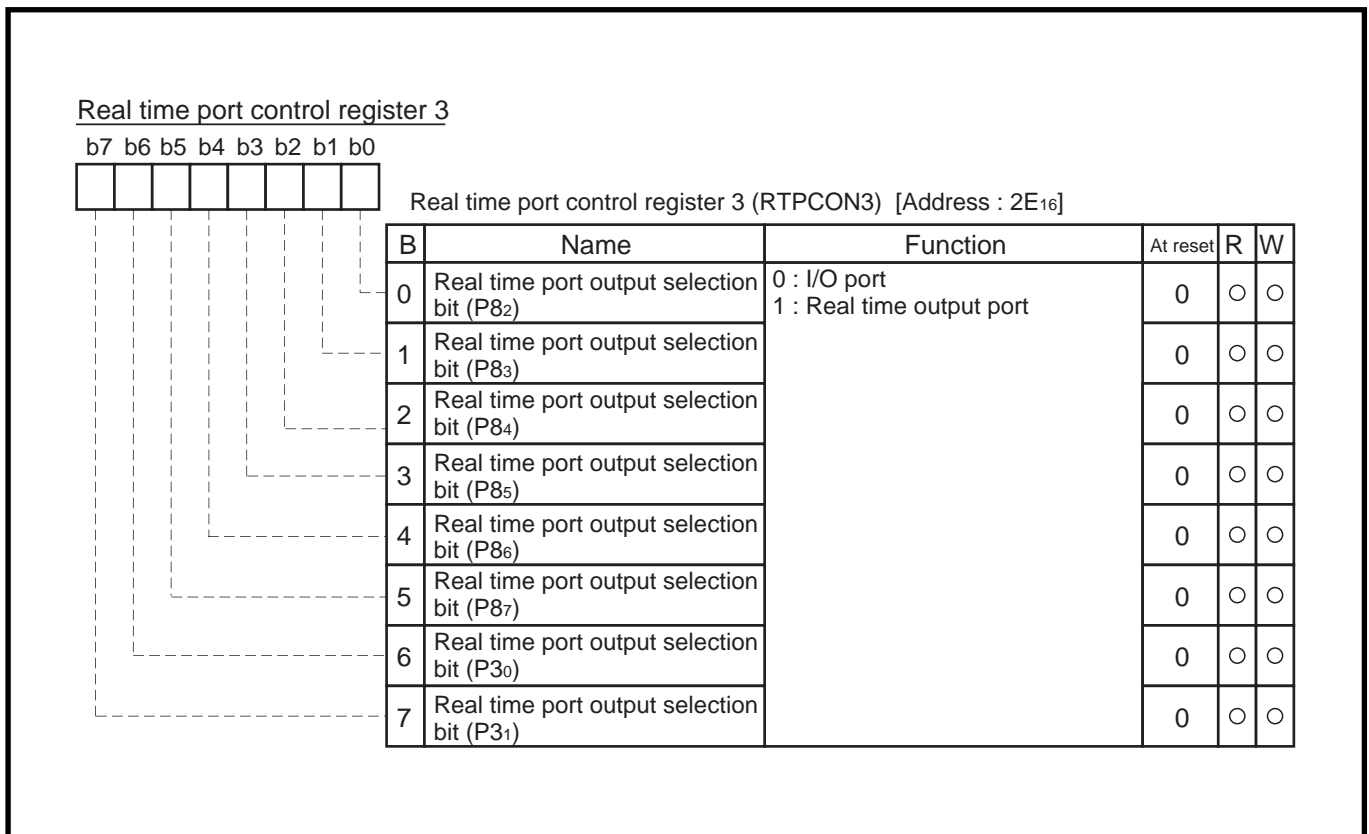


Fig. 3.5.27 Structure of Real time port control register 3

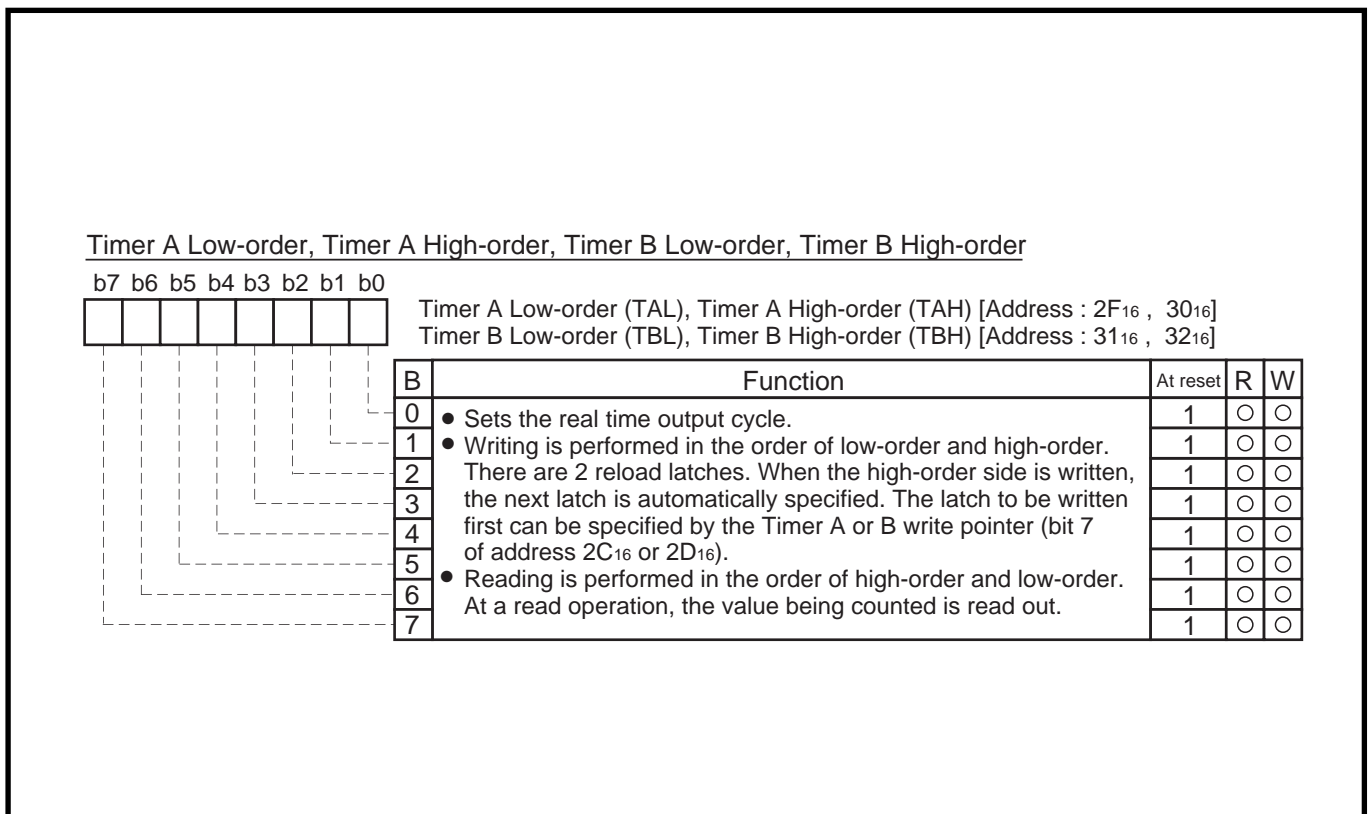
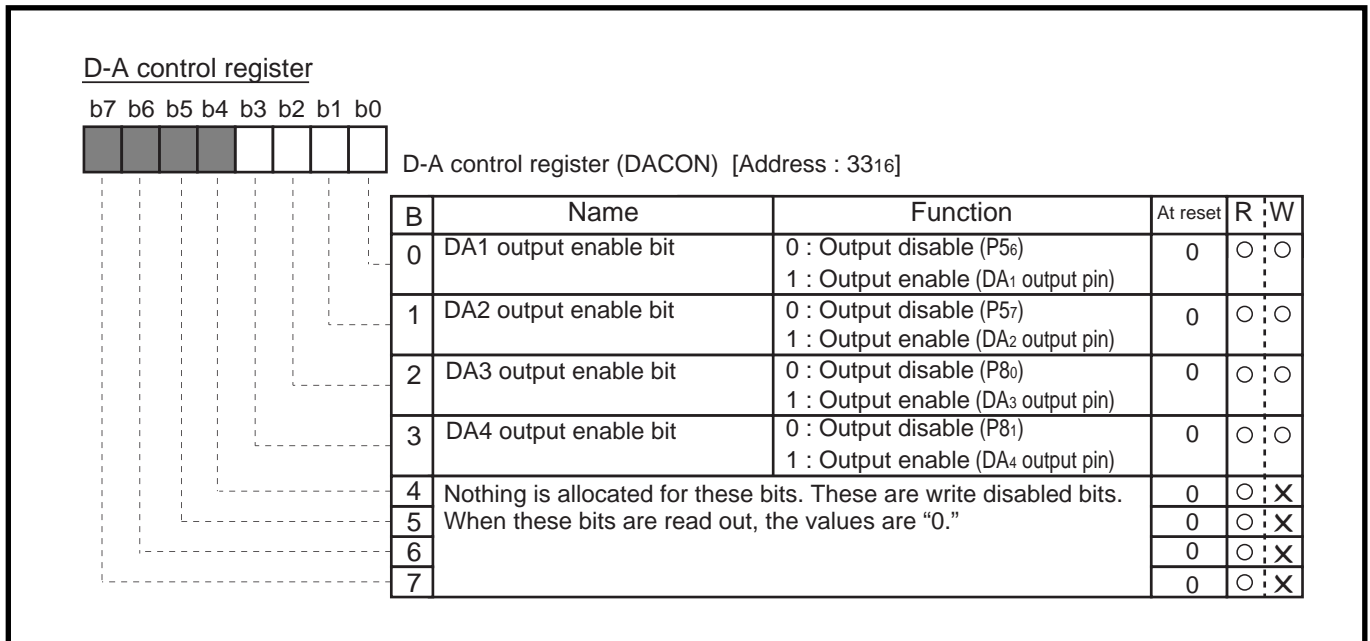
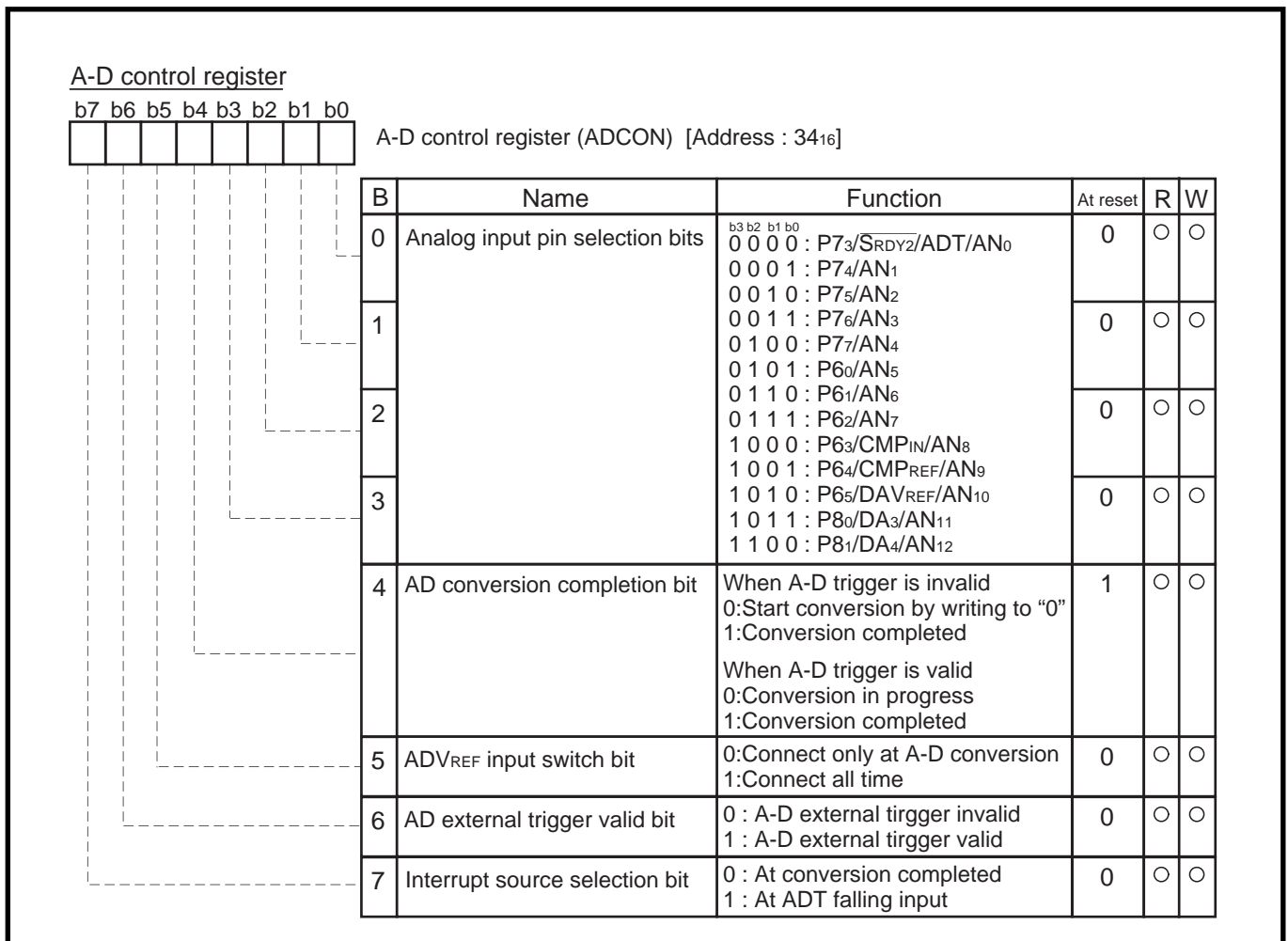


Fig. 3.5.28 Structure of Timer A Low-order, Timer A High-order, Timer B Low-order, Timer B High-order





**Fig. 3.5.29 Structure of D-A control register**



**Fig. 3.5.30 Structure of A-D control register**

# APPENDIX

## 3.5 List of registers

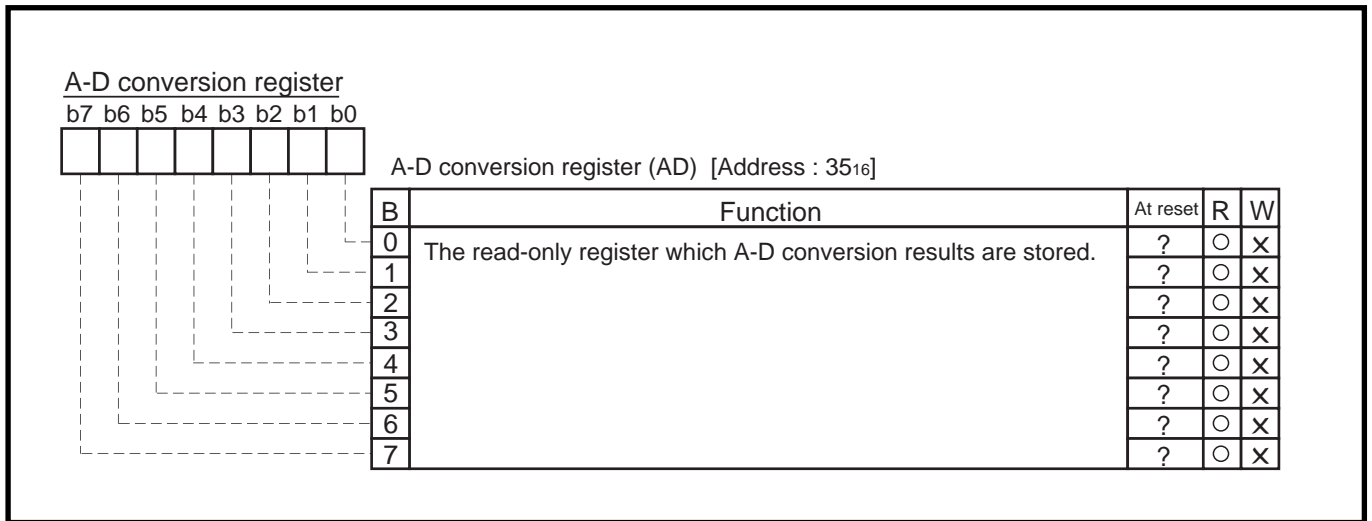


Fig. 3.5.31 Structure of A-D conversion register

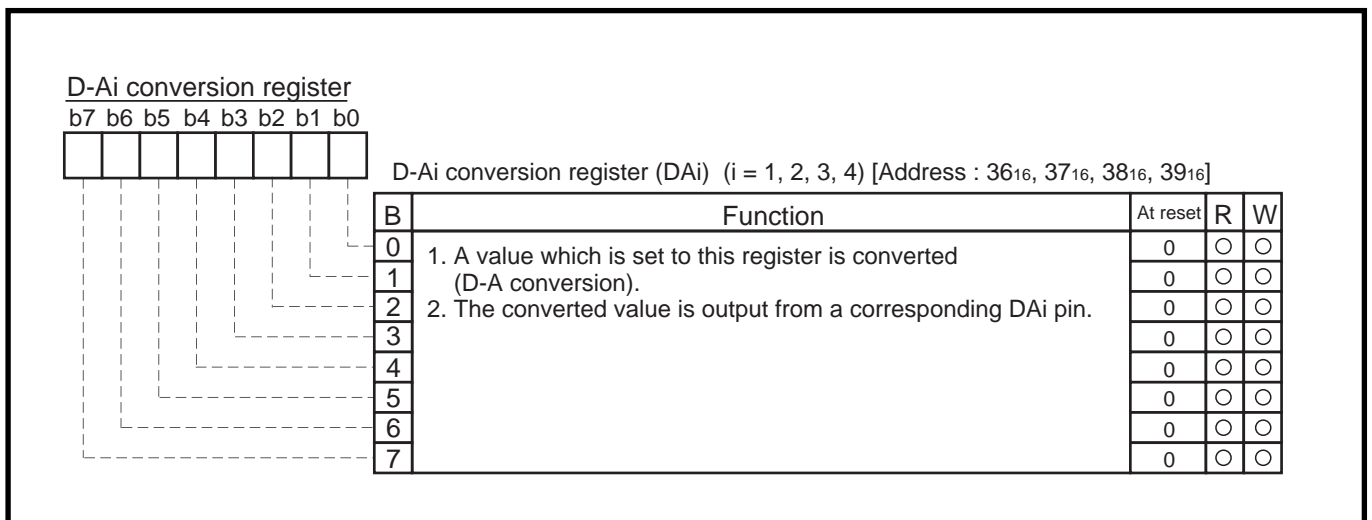


Fig. 3.5.32 Structure of D-Ai conversion register (i=1, 2, 3, 4)

### Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0



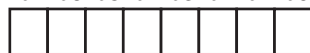
Interrupt edge selection register (INTEDGE) [Address : 3A16]

B	Name	Function	At reset	R	W
0	INT0 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT1 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
2	INT2 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
3	INT3 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
4	INT4 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
5	Timer 1/INT2 interrupt source bit	0 : INT2 interrupt 1 : Timer 1 interrupt	0	○	○
6	Timer 2/INT3 interrupt source bit	0 : INT3 interrupt 1 : Timer 2 interrupt	0	○	○
7	Timer 3/INT4 interrupt source bit	0 : INT4 interrupt 1 : Timer 3 interrupt	0	○	○

Fig. 3.5.33 Structure of Interrupt edge selection register

### CPU mode register

b7 b6 b5 b4 b3 b2 b1 b0



CPU mode register (CPUM) [Address : 3B16]

B	Name	Function	At reset	R	W
0	Processor mode bits	<sup>b1 b0</sup> 0 0 : Single-chip mode 0 1 : Memory expansion mode	0	○	○
1		1 0 : Microprocessor 1 1 : Not available			
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	X <sub>COU</sub> T drivability selection bit	0 : Low 1 : High	1	○	○
4	Port X <sub>C</sub> switch bit	0 : I/O port function 1 : X <sub>CIN</sub> -X <sub>COU</sub> T operating function	0	○	○
5	Main clock (X <sub>IN</sub> -X <sub>OU</sub> T) stop bit	0 : Operating 1 : Stopped	0	○	○
6	Main clock division ratio selection bits	<sup>b7 b6</sup> 0 0 : $\phi = f(X_{IN})/2$ (high-speed mode) 0 1 : $\phi = f(X_{IN})/8$ (middle-speed mode)	1	○	○
7		1 0 : $\phi = f(X_{CIN})/2$ (low-speed mode) 1 1 : Not available			

**Note :** An initial value of bit 1 is determined by a level of the CNVss pin.

Fig. 3.5.34 Structure of CPU mode register

# APPENDIX

## 3.5 List of registers

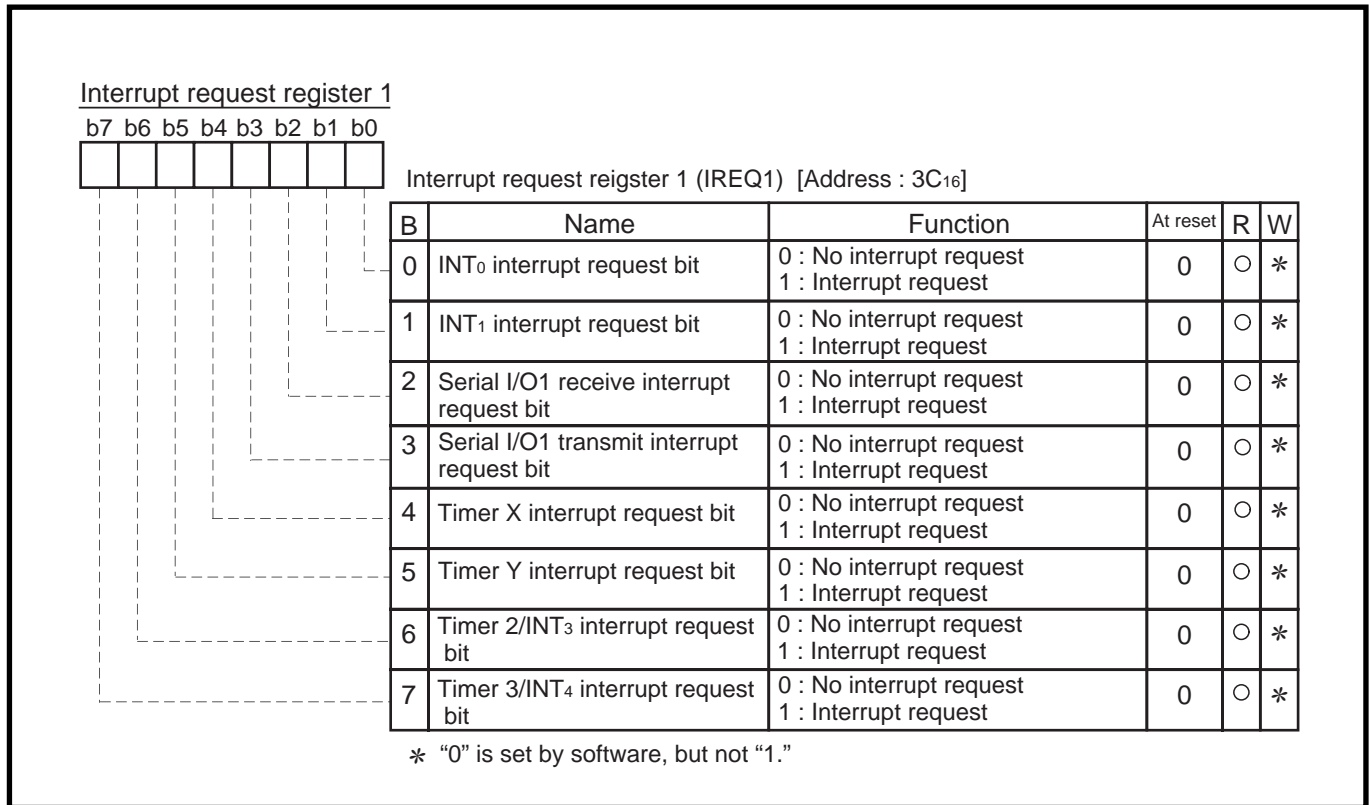


Fig. 3.5.35 Structure of Interrupt request register 1

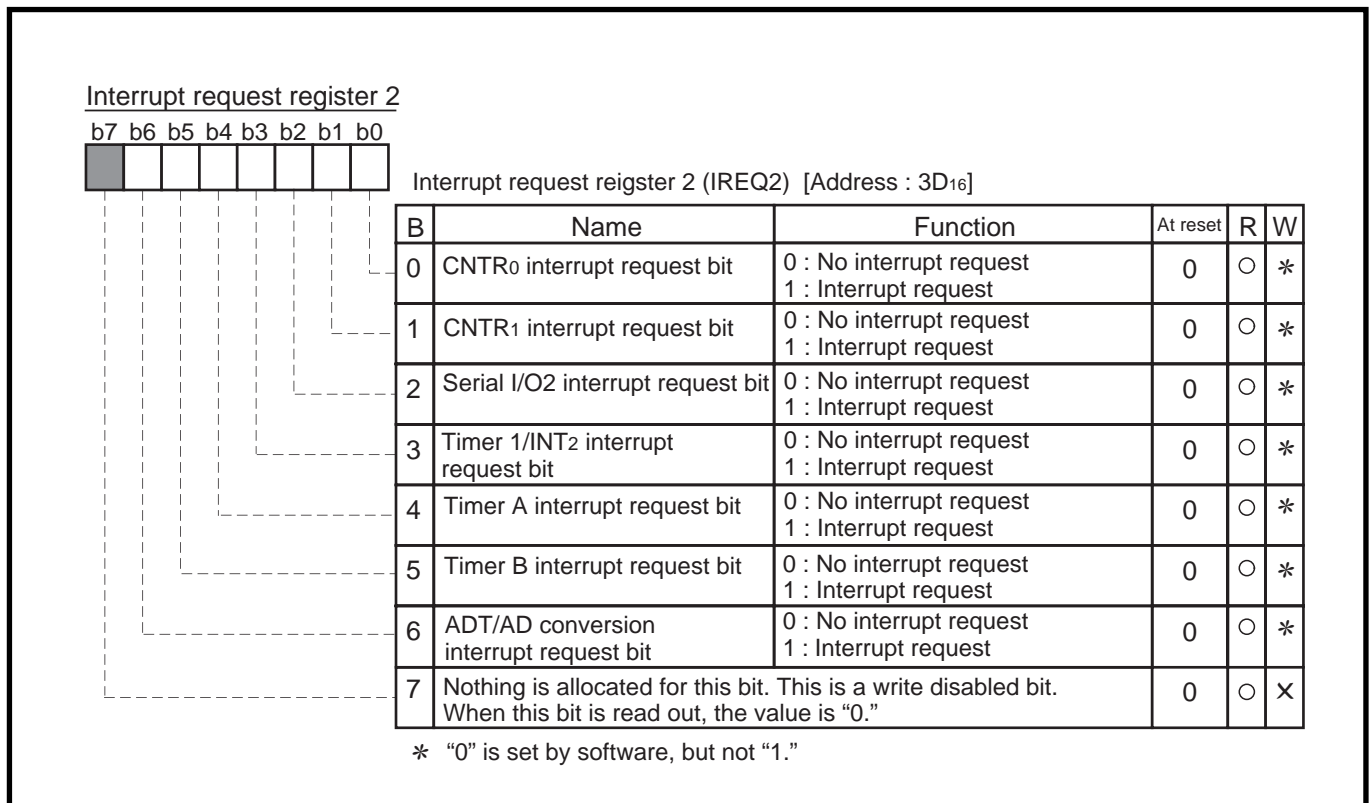


Fig. 3.5.36 Structure of Interrupt request register 2

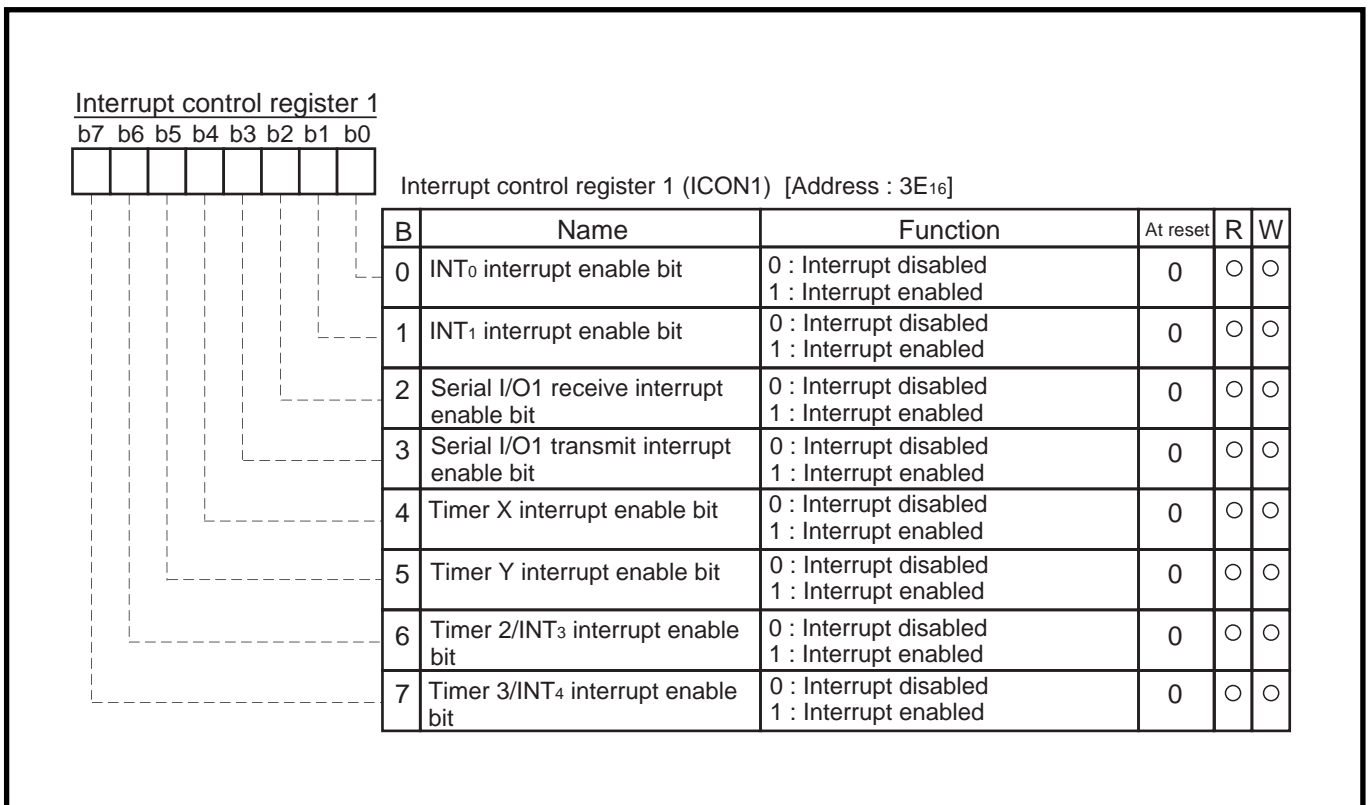


Fig. 3.5.37 Structure of Interrupt control register 1

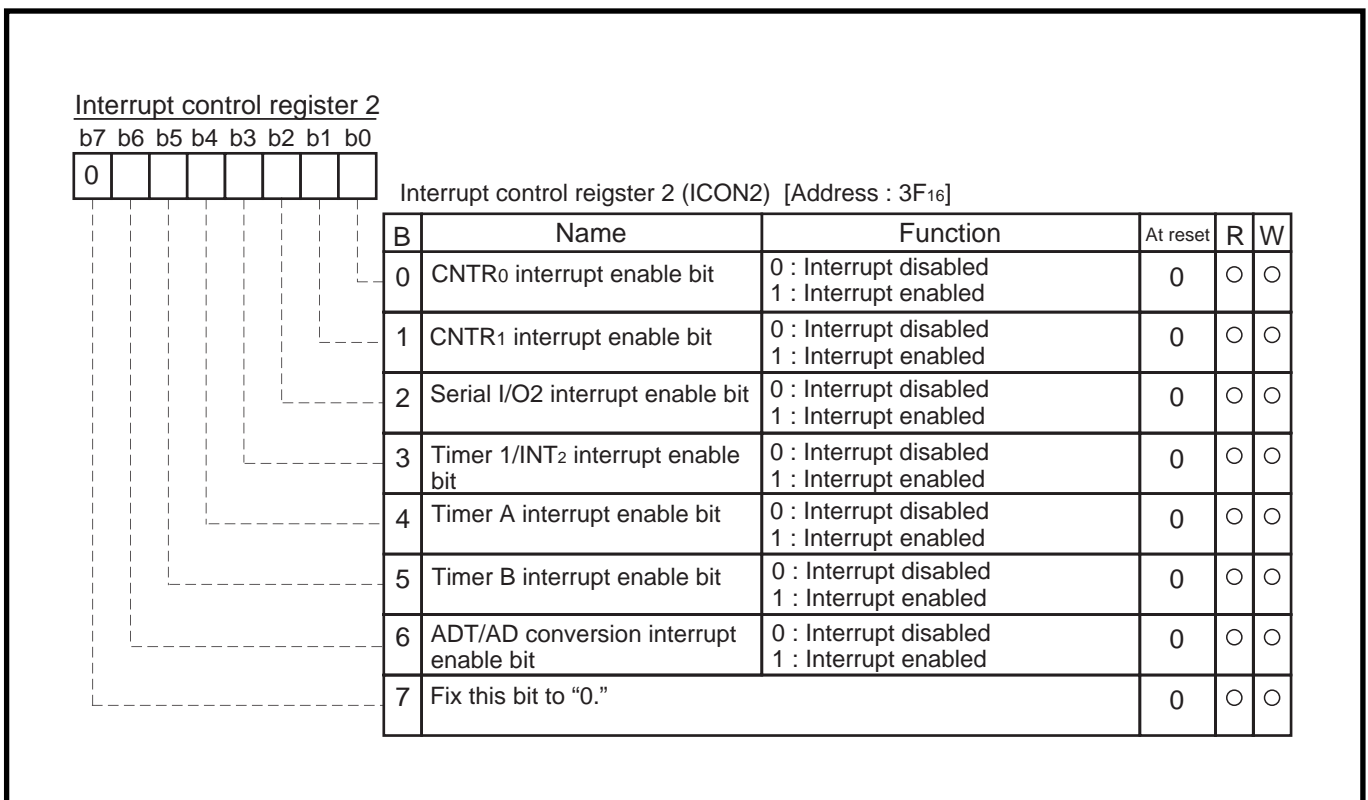


Fig. 3.5.38 Structure of Interrupt control register 2

# APPENDIX

## 3.6 Mask ROM ordering method

### 3.6 Mask ROM ordering method

GZZ-SH11-00B<68A0>

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

**740 FAMILY MASK ROM CONFIRMATION FORM  
SINGLE-CHIP MICROCOMPUTER M38073M4-XXXFP  
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 type of EPROMs submitted.

Three EPROMs are required for each pattern.

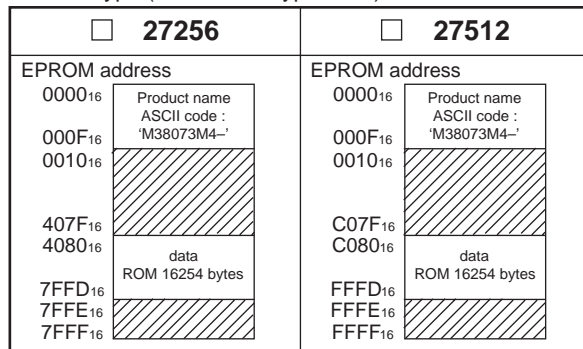
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the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

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 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>.

- (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 "M38073M4--" 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>	'-' = 2D <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>	'7' = 37 <sub>16</sub>	000C <sub>16</sub>	FF <sub>16</sub>
0005 <sub>16</sub>	'3' = 33 <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>

GZZ-SH11-00B<68A0>

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

**740 FAMILY MASK ROM CONFIRMATION FORM  
SINGLE-CHIP MICROCOMPUTER M38073M4-XXXFP  
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	*= Δ \$8000 .BYTE Δ 'M38073M4'	*= Δ \$0000 .BYTE Δ 'M38073M4'

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 (80P6N for M38073M4-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?

- Ceramic resonator                       Quartz crystal  
 External clock input                       Other (                      )

At what frequency?                      f(X<sub>IN</sub>) =  MHz

(2) Which function will you use the pins P<sub>41</sub>/X<sub>CIN</sub> and P<sub>40</sub>/X<sub>COU</sub>T as P<sub>41</sub> and P<sub>40</sub>, or X<sub>CIN</sub> and X<sub>COU</sub>T?

- Ports P<sub>41</sub> and P<sub>40</sub> function                       X<sub>CIN</sub> and X<sub>COU</sub>T function (external resonator)

※ 4. Comments

(2/2)

# APPENDIX

## 3.7 Mark specification form

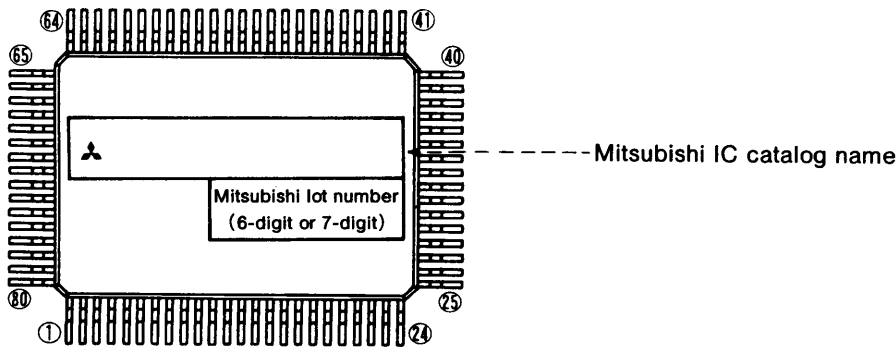
### 3.7 Mark specification form

#### 80P6N (80-PIN QFP) MARK SPECIFICATION FORM

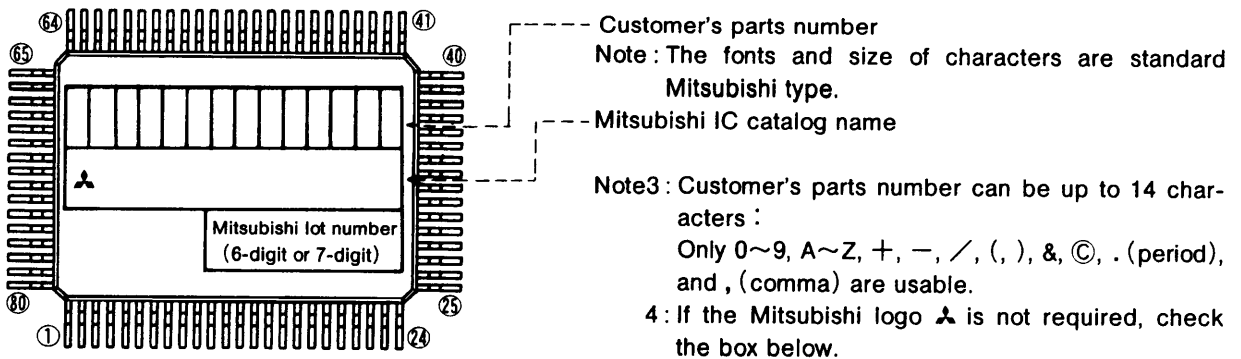
Mitsubishi IC catalog name

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



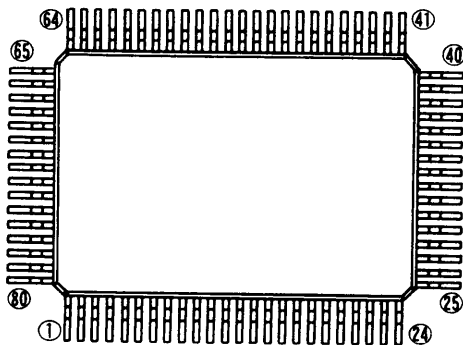
#### B. Customer's Parts Number + Mitsubishi Catalog Name



Mitsubishi logo is not required

Note1: The mark field should be written right aligned.  
2: The fonts and size of characters are standard Mitsubishi 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.



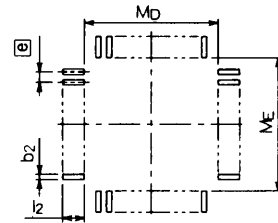
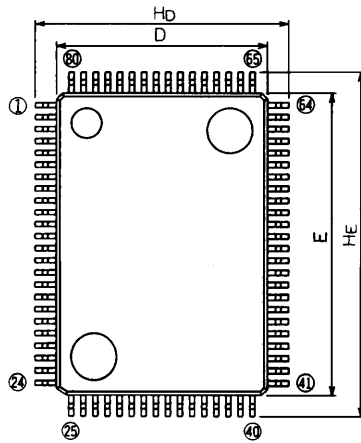
### 3.8 Package outline

#### 80P6N-A

Plastic 80pin 14x20mm body QFP

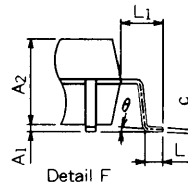
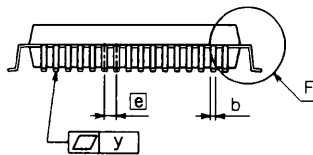
EIAJ Package Code	JEDEC Code	Weight (g)	Lead Material
QFP80-P-1420-0.80	—	1.58	Alloy 42

Scale : 2/1



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	3.05
A <sub>1</sub>	0	0.1	0.2
A <sub>2</sub>	—	2.8	—
b	0.3	0.35	0.45
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
ⓐ	—	0.8	—
HD	16.5	16.8	17.1
HE	22.5	22.8	23.1
L	0.4	0.6	0.8
L <sub>1</sub>	—	1.4	—
y	—	—	0.1
θ	0°	—	10°
b <sub>2</sub>	—	0.5	—
l <sub>2</sub>	1.3	—	—
MD	—	14.6	—
ME	—	20.6	—



# APPENDIX

## 3.9 Machine instructions

### 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	#			
ADC (Note 1) (Note 5)	When T = 0 $A \leftarrow A + M + C$  When T = 1 $M(X) \leftarrow M(X) + M + C$	Adds the carry, accumulator and memory contents. The results are entered into the accumulator.  Adds the contents of the memory in the address indicated by index register X, the contents of the memory specified by the addressing mode and the carry. The results are entered into the memory at the address indicated by index register X.				69	2	2										65	3	2			
AND (Note 1)	When T = 0 $A \leftarrow A \wedge M$  When T = 1 $M(X) \leftarrow M(X) \wedge M$	"AND's" the accumulator and memory contents. The results are entered into the accumulator.  "AND's" the contents of the memory of the address indicated by index register X and the contents of the memory specified by the addressing mode. The results are entered into the memory at the address indicated by index register X.				29	2	2										25	3	2			
ASL	$C \leftarrow \begin{matrix} 7 & 0 \\ \boxed{\phantom{00}} & \leftarrow 0 \end{matrix}$	Shifts the contents of accumulator or contents of memory one bit to the left. The low order bit of the accumulator or memory is cleared and the high order bit is shifted into the carry flag.							0A	2	1							06	5	2			
BBC (Note 4)	Ab or Mb = 0?	Branches when the contents of the bit specified in the accumulator or memory is "0".																$13_{20i}$	4	2			
BBS (Note 4)	Ab or Mb = 1?	Branches when the contents of the bit specified in the accumulator or memory is "1".																$03_{20i}$	4	2			
BCC (Note 4)	C = 0?	Branches when the contents of carry flag is "0".																					
BCS (Note 4)	C = 1?	Branches when the contents of carry flag is "1".																					
BEQ (Note 4)	Z = 1?	Branches when the contents of zero flag is "1".																					
BIT	$A \wedge M$	"AND's" the contents of accumulator and memory. The results are not entered anywhere.																24	3	2			
BMI (Note 4)	N = 1?	Branches when the contents of negative flag is "1".																					
BNE (Note 4)	Z = 0?	Branches when the contents of zero flag is "0".																					
BPL (Note 4)	N = 0?	Branches when the contents of negative flag is "0".																					
BRA	$PC \leftarrow PC \pm \text{offset}$	Jumps to address specified by adding offset to the program counter.																					
BRK	B ← 1 M(S) ← PCH S ← S - 1 M(S) ← PCL S ← S - 1 M(S) ← PS S ← S - 1 PCL ← ADL PCH ← ADH	Executes a software interrupt.	00	7	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	#	
BVC (Note 4)	$V = 0?$	Branches when the contents of overflow flag is "0".																			
BVS (Note 4)	$V = 1?$	Branches when the contents of overflow flag is "1".																			
CLB	$Ab \text{ or } Mb \leftarrow 0$	Clears the contents of the bit specified in the accumulator or memory to "0".										$1B_{20i}$	2	1					$1F_{20i}$	5	2
CLC	$C \leftarrow 0$	Clears the contents of the carry flag to "0".	18	2	1																
CLD	$D \leftarrow 0$	Clears the contents of decimal mode flag to "0".	D8	2	1																
CLI	$I \leftarrow 0$	Clears the contents of interrupt disable flag to "0".	58	2	1																
CLT	$T \leftarrow 0$	Clears the contents of index X mode flag to "0".	12	2	1																
CLV	$V \leftarrow 0$	Clears the contents overflow flag to "0".	B8	2	1																
CMP (Note 3)	When $T = 0$ $A - M$ When $T = 1$ $M(X) - M$	Compares the contents of accumulator and memory. Compares the contents of the memory specified by the addressing mode with the contents of the address indicated by index register X.							C9	2	2								C5	3	2
COM	$M \leftarrow \overline{M}$	Forms a one's complement of the contents of memory, and stores it into memory.																	44	5	2
CPX	$X - M$	Compares the contents of index register X and memory.							E0	2	2								E4	3	2
CPY	$Y - M$	Compares the contents of index register Y and memory.							C0	2	2								C4	3	2
DEC	$A \leftarrow A - 1$ or $M \leftarrow M - 1$	Decrements the contents of the accumulator or memory by 1.										1A	2	1					C6	5	2
DEX	$X \leftarrow X - 1$	Decrements the contents of index register X by 1.	CA	2	1																
DEY	$Y \leftarrow Y - 1$	Decrements the contents of index register Y by 1.	88	2	1																
DIV	$A \leftarrow (M(zz + X + 1),$ $M(zz + X)) / A$ $M(S) \leftarrow 1$ 's complement of Remainder $S \leftarrow S - 1$	Divides the 16-bit data that is the contents of M (zz + x + 1) for high byte and the contents of M (zz + x) for low byte by the accumulator. Stores the quotient in the accumulator and the 1's complement of the remainder on the stack.																			
EOR (Note 1)	When $T = 0$ $A \leftarrow A \vee M$ When $T = 1$ $M(X) \leftarrow M(X) \vee M$	"Exclusive-ORs" the contents of accumulator and memory. The results are stored in the accumulator. "Exclusive-ORs" the contents of the memory specified by the addressing mode and the contents of the memory at the address indicated by index register X. The results are stored into the memory at the address indicated by index register X.							49	2	2								45	3	2
INC	$A \leftarrow A + 1$ or $M \leftarrow M + 1$	Increments the contents of accumulator or memory by 1.										3A	2	1					E6	5	2
INX	$X \leftarrow X + 1$	Increments the contents of index register X by 1.	E8	2	1																
INY	$Y \leftarrow Y + 1$	Increments the contents of index register Y by 1.	C8	2	1																

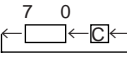
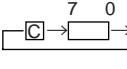







# APPENDIX

## 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	#	
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 $T = 0$ $A \leftarrow A - M - \bar{C}$  When $T = 1$ $M(X) \leftarrow M(X) - M - \bar{C}$	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	$Ab$ or $Mb \leftarrow 1$	Sets the specified bit in the accumulator or memory to "1".											$0B_{20i}$	2	1				$0F_{20i}$	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

D3 – D0 D7 – D4 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

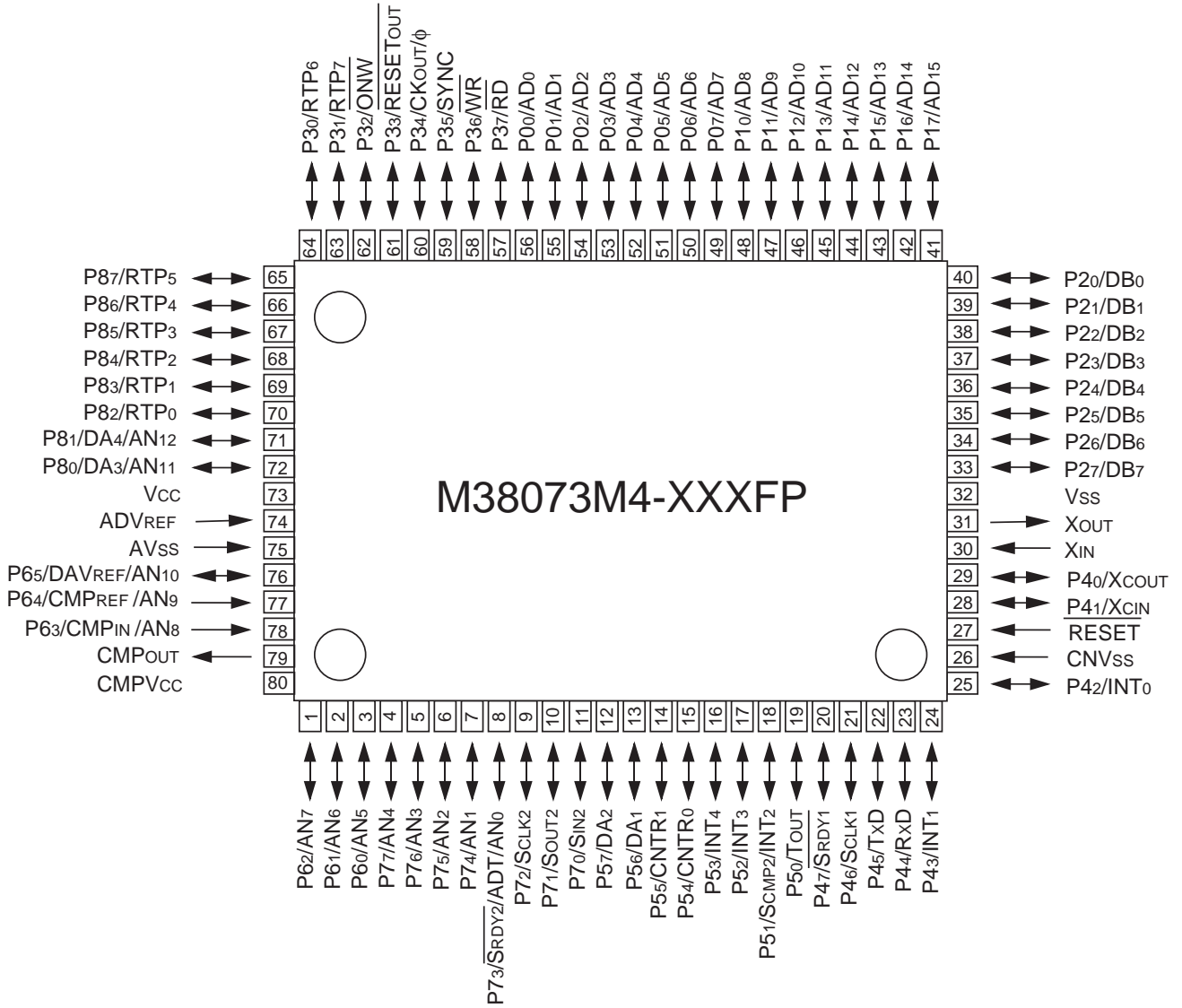
### 3.11 SFR memory map

0000 <sub>16</sub>	Port P0 (P0)	0020 <sub>16</sub>	Timer X (low-order) (TXL)
0001 <sub>16</sub>	Port P0 direction register (P0D)	0021 <sub>16</sub>	Timer X (high-order) (TXH)
0002 <sub>16</sub>	Port P1 (P1)	0022 <sub>16</sub>	Timer Y (low-order) (TYL)
0003 <sub>16</sub>	Port P1 direction register (P1D)	0023 <sub>16</sub>	Timer Y (high-order) (TYH)
0004 <sub>16</sub>	Port P2 (P2)	0024 <sub>16</sub>	Timer 1 (T1)
0005 <sub>16</sub>	Port P2 direction register (P2D)	0025 <sub>16</sub>	Timer 2 (T2)
0006 <sub>16</sub>	Port P3 (P3)	0026 <sub>16</sub>	Timer 3 (T3)
0007 <sub>16</sub>	Port P3 direction register (P3D)	0027 <sub>16</sub>	Timer X mode register (TXM)
0008 <sub>16</sub>	Port P4 (P4)	0028 <sub>16</sub>	Timer Y mode register (TYM)
0009 <sub>16</sub>	Port P4 direction register (P4D)	0029 <sub>16</sub>	Timer 123 mode register (T123M)
000A <sub>16</sub>	Port P5 (P5)	002A <sub>16</sub>	Real time port register (RTP)
000B <sub>16</sub>	Port P5 direction register (P5D)	002B <sub>16</sub>	Real time port control register 0 (RTPCON0)
000C <sub>16</sub>	Port P6 (P6)	002C <sub>16</sub>	Real time port control register 1 (RTPCON1)
000D <sub>16</sub>	Port P6 direction register (P6D)	002D <sub>16</sub>	Real time port control register 2 (RTPCON2)
000E <sub>16</sub>	Port P7 (P7)	002E <sub>16</sub>	Real time port control register 3 (RTPCON3)
000F <sub>16</sub>	Port P7 direction register (P7D)	002F <sub>16</sub>	Timer A (low-order) (TAL)
0010 <sub>16</sub>	Port P8 (P8)	0030 <sub>16</sub>	Timer A (high-order) (TAH)
0011 <sub>16</sub>	Port P8 direction register (P8D)	0031 <sub>16</sub>	Timer B (low-order) (TBL)
0012 <sub>16</sub>		0032 <sub>16</sub>	Timer B (high-order) (TBH)
0013 <sub>16</sub>		0033 <sub>16</sub>	D-A control register (DACON)
0014 <sub>16</sub>	Timer XY control register (TXYCON)	0034 <sub>16</sub>	A-D control register (ADCON)
0015 <sub>16</sub>	Port P2P3 control register (P2P3C)	0035 <sub>16</sub>	A-D conversion register (AD)
0016 <sub>16</sub>	Pull-up control register (PULL)	0036 <sub>16</sub>	D-A1 conversion register (DA1)
0017 <sub>16</sub>	Watchdog timer control register (WDTCON)	0037 <sub>16</sub>	D-A2 conversion register (DA2)
0018 <sub>16</sub>	Transmit/Receive buffer register (TB/RB)	0038 <sub>16</sub>	D-A3 conversion register (DA3)
0019 <sub>16</sub>	Serial I/O1 status register (SIO1STS)	0039 <sub>16</sub>	D-A4 conversion register (DA4)
001A <sub>16</sub>	Serial I/O1 control register (SIO1CON)	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>	Serial I/O2 control register 1 (SIO2CON1)	003D <sub>16</sub>	Interrupt request register 2 (IREQ2)
001E <sub>16</sub>	Serial I/O2 control register 2 (SIO2CON2)	003E <sub>16</sub>	Interrupt control register 1 (ICON1)
001F <sub>16</sub>	Serial I/O2 register (SIO2)	003F <sub>16</sub>	Interrupt control register 2 (ICON2)

# APPENDIX

## 3.12 Pin configuration

### 3.12 Pin configuration



**MITSUBISHI SEMICONDUCTORS  
USER'S MANUAL  
3807 Group**

---

Nov. 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**

# 3807 Group User's Manual



**Renesas Electronics Corporation**

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan



REVISION DESCRIPTION LIST

3807 Group User's Manual

Rev. No.	Revision Description	Rev. date
1.0	First Edition	971101