To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

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

User's Manual

8

3800 Group

User's Manual

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

Renesas Electronics
www.renesas.com

New publication, 1996.03

Keep safety first in your circuit designs!

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

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams and charts, represent information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
  Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

Preface

This user's manual describes Mitsubishi's CMOS 8-bit microcomputers 3800 Group.
After reading this manual, the user should have a through knowledge of the functions and features of the 3800 Group, and should be able to fully utilize the product. The manual starts with specifications and ends with application examples.
For details of software, refer to the "SERIES MELPS 740 <SOFTWARE> USER'S MANUAL."
For details of development support tools, refer to the "DEVELOPMENT SUPPORT TOOLS FOR MICRO-COMPUTERS" data book.

BEFORE USING THIS USER'S MANUAL

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

1. Organization

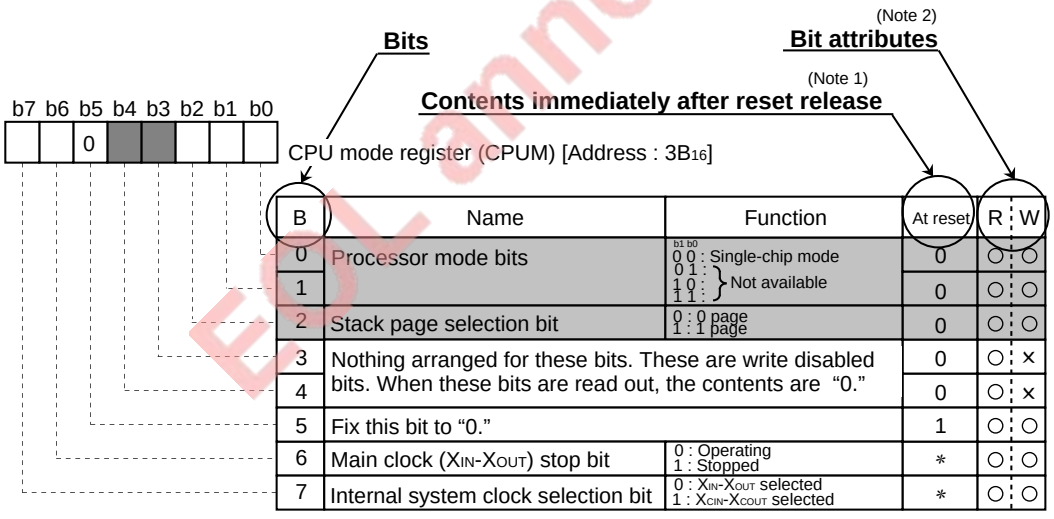
● CHAPTER 1 HARDWARE
This chapter describes features of the microcomputer and operation of each peripheral function.

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

● CHAPTER 3 APPENDIX
This chapter includes necessary information for systems development using the microcomputer, electric characteristics, a list of registers, the masking confirmation (mask ROM version), and mark specifications which are to be submitted when ordering.

2. Structure of register

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



B	Name	Function	At reset	R	W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : 1 0 : 1 1 : } Not available	0	○	○
1			0	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Nothing arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0."		0	○	×
4			0	○	×
5	Fix this bit to "0."		1	○	○
6	Main clock (X_{IN}-X_{OUT}) stop bit	0 : Operating 1 : Stopped	*	○	○
7	Internal system clock selection bit	0 : X_{IN}-X_{OUT} selected 1 : X_{CIN}-X_{COUT} selected	*	○	○

: Bit in which nothing is arranged : Bit that is not used for control of the corresponding function

Note 1. Contents immediately after reset release
0••••••"0" at reset release
1••••••"1" at reset release
Undefined••••••Undefined or reset release
* ••••••Contents determined by option at reset release

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

R••••••Read
○••••••Read enabled
×••••••Read disabled
W••••••Write
○ ••••••Write enabled
× ••••••Write disabled

LIST OF GROUPS HAVING THE SIMILAR FUNCTIONS

3800 group, one of the CMOS 8-bit microcomputer 38000 series presented in this user's manual is provided with standard functions.
The basic functions of the 3800, 3802, 3806 and 3807 groups having the same functions are shown below. For the detailed functions of each group, refer to the related data book and user's manual.

List of groups having the same functions

As of September 1995

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

Notes 1: Extended operating temperature version available
2: High-speed version available
3: Extended operating temperature version and High-speed version available
✻. ROM expansion

Table of contents

CHAPTER 1. HARDWARE

DESCRIPTION **1-2**

FEATURES **1-2**

APPLICATIONS **1-2**

PIN CONFIGURATION **1-2**

FUNCTIONAL BLOCK **1-4**

PIN DESCRIPTION **1-5**

PART NUMBERING **1-6**

GROUP EXPANSION **1-7**

GROUP EXPANSION (EXTENDED OPERATING TEMPERATURE VERSION) **1-9**

FUNCTIONAL DESCRIPTION **1-10**
- Central Processing Unit (CPU) 1-10
- Memory 1-14
- I/O Ports 1-16
- Interrupts 1-18
- Timers 1-20
- Serial I/O 1-22
- Reset Circuit 1-26
- Clock Generating Circuit 1-28
- Processor Modes 1-29

NOTES ON PROGRAMMING **1-31**
- Processor Status Register 1-31
- Interrupts 1-31
- Decimal Calculations 1-31
- Timers 1-31
- Multiplication and Division Instructions 1-31
- Ports 1-31
- Serial I/O 1-31
- Instruction Execution Time 1-31
- Memory Expansion Mode and Microprocessor Mode 1-31

DATA REQUIRED FOR MASK ORDERS **1-32**

ROM PROGRAMMING METHOD **1-32**

FUNCTIONAL DESCRIPTION SUPPLEMENT **1-33**
- Interrupt 1-33
- Timing After Interrupt 1-34

Table of contents

CHAPTER 2. APPLICATION

2.1 I/O port 2-2
2.1.1 Memory map of I/O port 2-2
2.1.2 Related registers 2-3
2.1.3 Handling of unused pins 2-4

2.2 Timer 2-5
2.2.1 Memory map of timer 2-5
2.2.2 Related registers 2-6
2.2.3 Timer application examples 2-11

2.3 Serial I/O 2-23
2.3.1 Memory map of serial I/O 2-23
2.3.2 Related registers 2-24
2.3.3 Serial I/O connection examples 2-28
2.3.4 Setting of serial I/O transfer data format 2-30
2.3.5 Serial I/O application examples 2-31

2.4 Processor mode 2-49
2.4.1 Memory map of processor mode 2-49
2.4.2 Related register 2-49
2.4.3 Processor mode application examples 2-50

2.5 Reset 2-54
2.5.1 Connection example of reset IC 2-54

CHAPTER 3. APPENDIX

3.1 Electrical characteristics 3-2
3.1.1 Absolute maximum ratings 3-2
3.1.2 Recommended operating conditions 3-2
3.1.3 Electrical characteristics 3-3
3.1.4 Timing requirements and Switching characteristics 3-4
3.1.5 Absolute maximum ratings (Extended operating temperature version) 3-8
3.1.6 Recommended operating conditions(Extended operating temperature version) 3-8
3.1.7 Electrical characteristics (Extended operating temperature version) 3-9
3.1.8 Timing requirements and Switching characteristics (Extended operating temperature version) 3-10
3.1.9 Timing diagram 3-12

3.2 Standard characteristics 3-15
3.2.1 Power source current characteristic examples 3-15
3.2.2 Port standard characteristic examples 3-16

3.3 Notes on use 3-18
3.3.1 Notes on interrupts 3-18
3.3.2 Notes on the serial I/O 3-18
3.3.3 Notes on the $\overline{\text{RESET}}$ pin 3-19
3.3.4 Notes on input and output pins 3-20

3.3.5 Notes on memory expansion mode and microprocessor mode 3-21
3.3.6 Notes on built-in PROM 3-22

3.4 Countermeasures against noise 3-24
3.4.1 Shortest wiring length 3-24
3.4.2 Connection of a bypass capacitor across the Vss line and the Vcc line 3-25
3.4.3 Consideration for oscillator 3-26
3.4.4 Setup for I/O ports 3-26
3.4.5 Providing of watchdog timer function by software 3-27

3.5 List of registers 3-28

3.6 Mask ROM ordering method 3-37

3.7 Mark specification form 3-51

3.8 Package outline 3-53

3.9 Machine Instructions 3-56

3.10 List of instruction codes 3-66

3.11 SFR memory map 3-67

3.12 Pin configuration 3-68

List of figures

CHAPTER 1 HARDWARE

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

CHAPTER 2 APPLICATION

Fig. 2.1.1 Memory map of I/O port related registers 2-2
Fig. 2.1.2 Structure of Port Pi (i=0, 1, 2, 3, 4, 5, 6, 7) 2-3
Fig. 2.1.3 Structure of Port Pi direction register (i=0, 1, 2, 3, 4, 5, 6, 7) 2-3

Fig. 2.2.1 Memory map of timer related registers 2-5
Fig. 2.2.2 Structure of Prescaler 12, Prescaler X, Prescaler Y 2-6
Fig. 2.2.3 Structure of Timer 1 2-6
Fig. 2.2.4 Structure of Timer 2, Timer X, Timer Y 2-7
Fig. 2.2.5 Structure of Timer XY mode register 2-8
Fig. 2.2.6 Structure of Interrupt request register 1 2-9

List of figures

Fig. 2.2.7 Structure of Interrupt request register 2 2-9
Fig. 2.2.8 Structure of Interrupt control register 1 2-10
Fig. 2.2.9 Structure of Interrupt control register 2 2-10
Fig. 2.2.10 Connection of timers and setting of division ratios [Clock function] 2-12
Fig. 2.2.11 Setting of related registers [Clock function] 2-13
Fig. 2.2.12 Control procedure [Clock function] 2-14
Fig. 2.2.13 Example of a peripheral circuit 2-15
Fig. 2.2.14 Connection of the timer and setting of the division ratio [Piezoelectric buzzer output] 2-15
Fig. 2.2.15 Setting of related registers [Piezoelectric buzzer output] 2-16
Fig. 2.2.16 Control procedure [Piezoelectric buzzer output] 2-16
Fig. 2.2.17 A method for judging if input pulse exists 2-17
Fig. 2.2.18 Setting of related registers [Measurement of frequency] 2-18
Fig. 2.2.19 Control procedure [Measurement of frequency] 2-19
Fig. 2.2.20 Connection of the timer and setting of the division ratio [Measurement of pulse width] 2-20
Fig. 2.2.21 Setting of related registers [Measurement of pulse width] 2-21
Fig. 2.2.22 Control procedure [Measurement of pulse width] 2-22

Fig. 2.3.1 Memory map of serial I/O related registers 2-23
Fig. 2.3.2 Structure of Transmit/Receive buffer register 2-24
Fig. 2.3.3 Structure of Serial I/O status register 2-24
Fig. 2.3.4 Structure of Serial I/O control register 2-25
Fig. 2.3.5 Structure of UART control register 2-25
Fig. 2.3.6 Structure of Baud rate generator 2-26
Fig. 2.3.7 Structure of Interrupt edge selection register 2-26
Fig. 2.3.8 Structure of Interrupt request register 1 2-27
Fig. 2.3.9 Structure of Interrupt control register 1 2-27
Fig. 2.3.10 Serial I/O connection examples (1) 2-28
Fig. 2.3.11 Serial I/O connection examples (2) 2-29
Fig. 2.3.12 Setting of Serial I/O transfer data format 2-30
Fig. 2.3.13 Connection diagram [Communication using a clock synchronous serial I/O] 2-31
Fig. 2.3.14 Timing chart [Communication using a clock synchronous serial I/O] 2-31
Fig. 2.3.15 Setting of related registers at a transmitting side [Communication using a clock synchronous serial I/O] 2-32
Fig. 2.3.16 Setting of related registers at a receiving side [Communication using a clock synchronous serial I/O] 2-33
Fig. 2.3.17 Control procedure at a transmitting side [Communication using a clock synchronous serial I/O] 2-34
Fig. 2.3.18 Control procedure at a receiving side[Communication using a clock synchronous serial I/O] . 2-35
Fig. 2.3.19 Connection diagram [Output of serial data] 2-36
Fig. 2.3.20 Timing chart [Output of serial data] 2-36
Fig. 2.3.21 Setting of serial I/O related registers [Output of serial data] 2-37
Fig. 2.3.22 Setting of serial I/O transmission data [Output of serial data] 2-37
Fig. 2.3.23 Control procedure of serial I/O [Output of serial data] 2-38
Fig. 2.3.24 Connection diagram [Cyclic transmission or reception of block data between microcomputers] 2-39
Fig. 2.3.25 Timing chart [Cyclic transmission or reception of block data between microcomputers] 2-40
Fig. 2.3.26 Setting of related registers [Cyclic transmission or reception of block data between microcomputers] . 2-40
Fig. 2.3.27 Control in the master unit 2-41
Fig. 2.3.28 Control in the slave unit 2-42
Fig. 2.3.29 Connection diagram [Communication using UART] 2-43
Fig. 2.3.30 Timing chart [Communication using UART] 2-43

Fig. 2.3.31 Setting of related registers at a transmitting side [Communication using UART] 2-45
Fig. 2.3.32 Setting of related registers at a receiving side [Communication using UART] 2-46
Fig. 2.3.33 Control procedure at a transmitting side [Communication using UART] 2-47
Fig. 2.3.34 Control procedure at a receiving side [Communication using UART] 2-48

Fig. 2.4.1 Memory map of processor mode related register 2-49
Fig. 2.4.2 Structure of CPU mode register 2-49
Fig. 2.4.3 Expansion example of ROM and RAM 2-50
Fig. 2.4.4 Read-cycle (OE access, SRAM) 2-51
Fig. 2.4.5 Read-cycle (OE access, EPROM) 2-51
Fig. 2.4.6 Write-cycle (W control, SRAM) 2-52
Fig. 2.4.7 Application example of the $\overline{\text{ONW}}$ function 2-53

Fig. 2.5.1 Example of Poweron reset circuit 2-54
Fig. 2.5.2 RAM back-up system 2-54

List of figures

CHAPTER 3 APPENDIX

Fig. 3.1.1 Circuit for measuring output switching characteristics 3-11
Fig. 3.1.2 Timing diagram (in single-chip mode) 3-12
Fig. 3.1.3 Timing diagram (in memory expansion mode and microprocessor mode) (1) 3-13
Fig. 3.1.4 Timing diagram (in memory expansion mode and microprocessor mode) (2) 3-14

Fig. 3.2.1 Power source current characteristic example 3-15
Fig. 3.2.2 Power source current characteristic example (in wait mode) 3-15
Fig. 3.2.3 Standard characteristic example of CMOS output port at P-channel drive(1). 3-16
Fig. 3.2.4 Standard characteristic example of CMOS output port at P-channel drive(2). 3-16
Fig. 3.2.5 Standard characteristic example of CMOS output port at N-channel drive(1) 3-17
Fig. 3.2.6 Standard characteristic example of CMOS output port at N-channel drive(2) 3-17

Fig. 3.3.1 Structure of interrupt control register 2 3-18

Fig. 3.4.1 Wiring for the $\overline{\text{RESET}}$ pin 3-24
Fig. 3.4.2 Wiring for clock I/O pins 3-25
Fig. 3.4.3 Wiring for the V_{PP} pin of the One Time PROM and the EPROM version 3-25
Fig. 3.4.4 Bypass capacitor across the V_{SS} line and the V_{CC} line 3-25
Fig. 3.4.5 Wiring for a large current signal line 3-26
Fig. 3.4.6 Wiring to a signal line where potential levels change frequently 3-26
Fig. 3.4.7 Stepup for I/O ports 3-26
Fig. 3.4.8 Watchdog timer by software 3-27

Fig. 3.5.1 Structure of Port Pi (i=0, 1, 2, 3, 4, 5, 6, 7) 3-28
Fig. 3.5.2 Structure of Port Pi direction register (i=0, 1, 2, 3, 4, 5, 6, 7) 3-28
Fig. 3.5.3 Structure of Transmit/Receive buffer register 3-29
Fig. 3.5.4 Structure of Serial I/O status register 3-29
Fig. 3.5.5 Structure of Serial I/O control register 3-30
Fig. 3.5.6 Structure of UART control register 3-30
Fig. 3.5.7 Structure of Baud rate generator 3-31
Fig. 3.5.8 Structure of Prescaler 12, Prescaler X, Prescaler Y 3-31
Fig. 3.5.9 Structure of Timer 1 3-32
Fig. 3.5.10 Structure of Timer 2, Timer X, Timer Y 3-32
Fig. 3.5.11 Structure of Timer XY mode register 3-33
Fig. 3.5.12 Structure of Interrupt edge selection register 3-34
Fig. 3.5.13 Structure of CPU mode register 3-34
Fig. 3.5.14 Structure of Interrupt request register 1 3-35
Fig. 3.5.15 Structure of Interrupt request register 2 3-35
Fig. 3.5.16 Structure of Interrupt control register 1 3-36
Fig. 3.5.17 Structure of Interrupt control register 2 3-36

List of tables

CHAPTER 1 HARDWARE

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

CHAPTER 2 APPLICATION

Table 2.1.1 Handling of unused pins (in single-chip mode) 2-4
Table 2.1.2 Handling of unused pins (in memory expansion mode and microprocessor mode) 2-4

Table 2.2.1 Function of $CNTR_0$/$CNTR_1$ edge switch bit 2-8

Table 2.3.1 Setting examples of Baud rate generator values and transfer bit rate values 2-44

CHAPTER 3 APPENDIX

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

List of tables

Table 3.3.1 Programming adapter 3-22
Table 3.3.2 Setting of programming adapter switch 3-22
Table 3.3.3 Setting of PROM programmer address 3-23

Table 3.5.1 Function of $CNTR_0$/$CNTR_1$ edge switch bit 3-33

CHAPTER 1
HARDWARE

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

DESCRIPTION/FEATURES/APPLICATIONS/PIN CONFIGURATION

DESCRIPTION

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

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

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

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

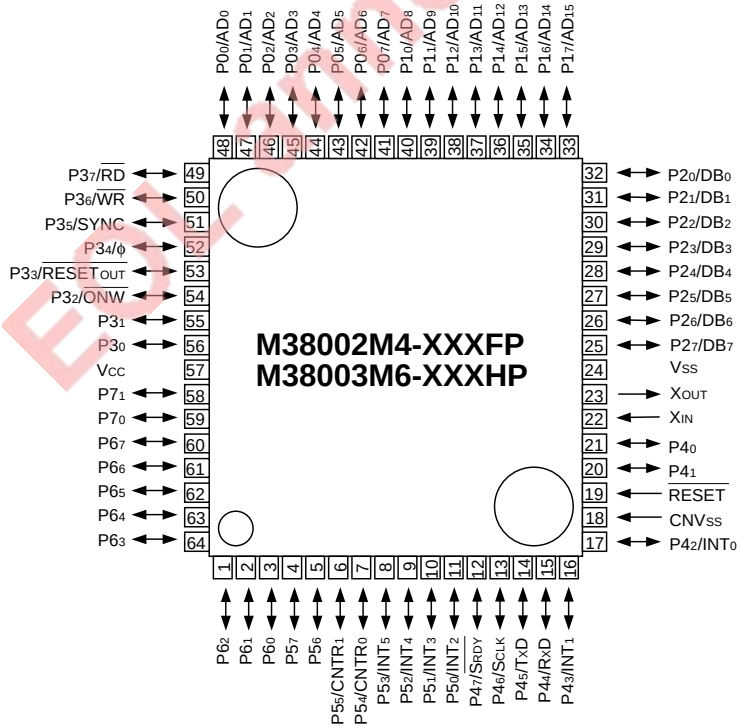
FEATURES

- Basic machine-language instructions 71
- The minimum instruction execution time ..... 0.5 µs
  (at 8 MHz oscillation frequency)
- Memory size
  ROM ..... 8 K to 32 K bytes
  RAM ..... 384 to 1024 bytes
- Programmable input/output ports 58
- Interrupts ..... 15 sources, 15 vectors
- Timers ..... 8 bit × 4
- Serial I/O .....8-bit × 1 (UART or Clock-synchronized)
- Clock generating circuit ..... Internal feedback resistor
  (connect to external ceramic resonator or quartz-crystal oscillator)
- Power source voltage .....3.0 to 5.5 V
  (Extended operating temperature version : 4.0 to 5.5 V)
- Power dissipation ..... 32 mW
- Memory expansion possible
- Operating temperature range ..... –20 to 85°C
  (Extended operating temperature version : –40 to 85°C)

APPLICATIONS

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

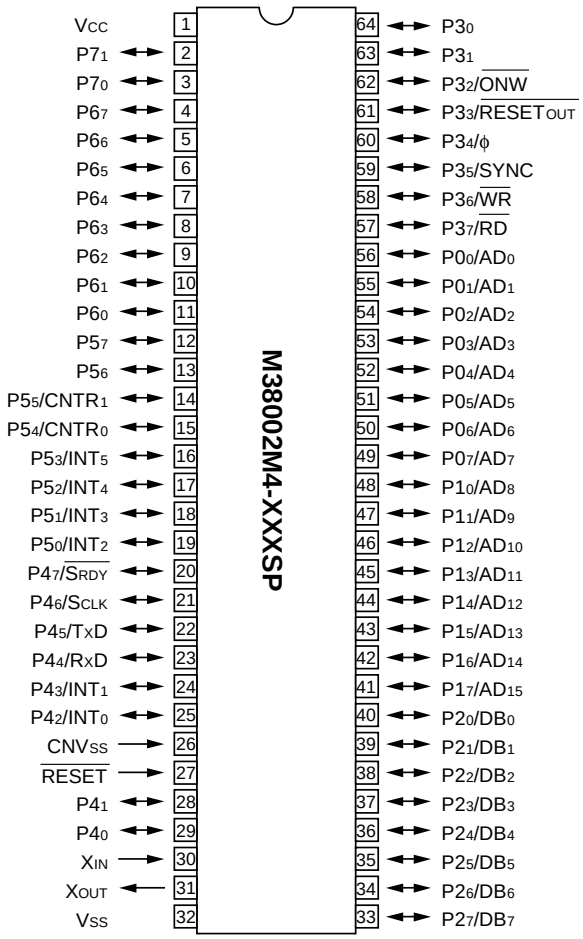
PIN CONFIGURATION (TOP VIEW)



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

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

PIN CONFIGURATION (TOP VIEW)



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

Fig. 2 Pin configuration of M38002M4-XXXSP

FUNCTIONAL BLOCK

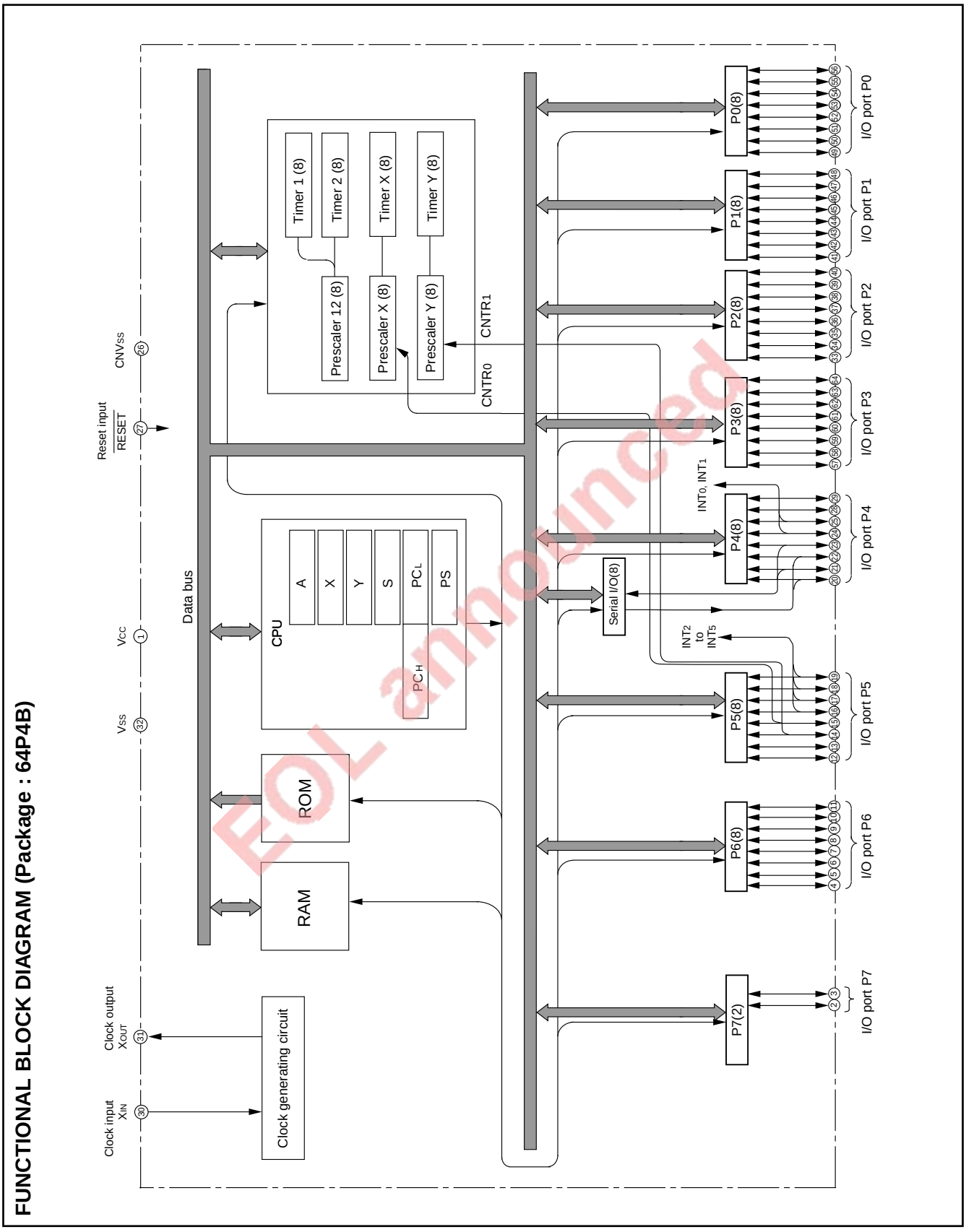


Fig. 3 Functional block diagram

PIN DESCRIPTION

Table 1. Pin description

Pin	Name	Function	Function except a port function
V_{CC}	Power source	• Apply voltage of 3.0 V to 5.5 V to V_{CC}, and 0 V to V_{SS}. (Extended operating temperature version : 4.0 V to 5.5 V)	
V_{SS}			
CNV_{SS}	CNV_{SS}	• This pin controls the operation mode of the chip. • Normally connected to V_{SS}. • If this pin is connected to V_{CC}, the internal ROM is inhibited and external memory is accessed.	
$\overline{RESET}$	Reset input	• Reset input pin for active “L”	
X_{IN}	Clock input	• Input and output signals for the internal clock generating circuit. • Connect a ceramic resonator or quartz-crystal oscillator between the X_{IN} and X_{OUT} pins to set the oscillation frequency. • If an external clock is used, connect the clock source to the X_{IN} pin and leave the X_{OUT} pin open. • The clock is used as the oscillating source of system clock.	
X_{OUT}	Clock output		
$P0_0$ – $P0_7$	I/O port P0	• 8 bit CMOS I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • At reset this port is set to input mode. • In modes other than single-chip, these pins are used as address, data, and control bus I/O pins. • CMOS compatible input level • CMOS 3-state output structure	
$P1_0$ – $P1_7$	I/O port P1		
$P2_0$ – $P2_7$	I/O port P2		
$P3_0$ – $P3_7$	I/O port P3		
$P4_0$, $P4_1$	I/O port P4	• 8-bit CMOS I/O port with the same function as port P0 • CMOS compatible input level • CMOS 3-state output structure	
$P4_2/INT_0$, $P4_3/INT_1$			• External interrupt input pins
$P4_4/RxD$, $P4_5/TxD$, $P4_6/S_{CLK}$, $P4_7/\overline{S_{RDY}}$			• Serial I/O I/O pins
$P5_0/INT_2$ – $P5_3/INT_5$	I/O port P5	• 8-bit CMOS I/O port with the same function as port P0 • CMOS compatible input level • CMOS 3-state output structure	• External interrupt input pins
$P5_4/CNTR_0$, $P5_5/CNTR_1$			• Timer X and Timer Y I/O pins
$P5_6$, $P5_7$			
$P6_0$ – $P6_7$	I/O port P6	• 8-bit CMOS I/O port with the same function as port P0 • CMOS compatible input level • CMOS 3-state output structure	
$P7_0$, $P7_1$	I/O port P7	• 2-bit CMOS I/O port with the same function as port P0 • CMOS compatible input level • CMOS 3-state output structure	

PART NUMBERING

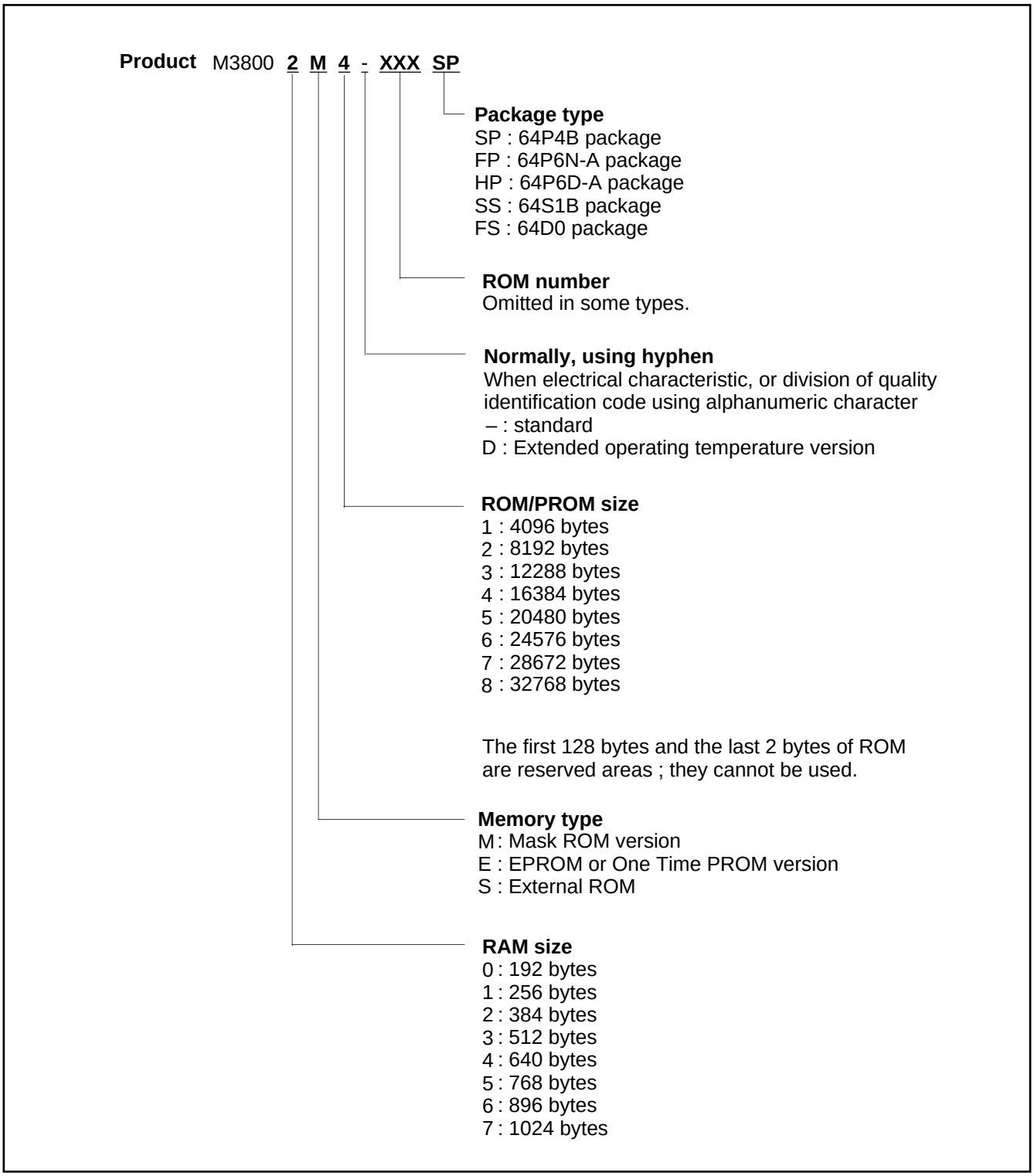


Fig. 4 Part numbering

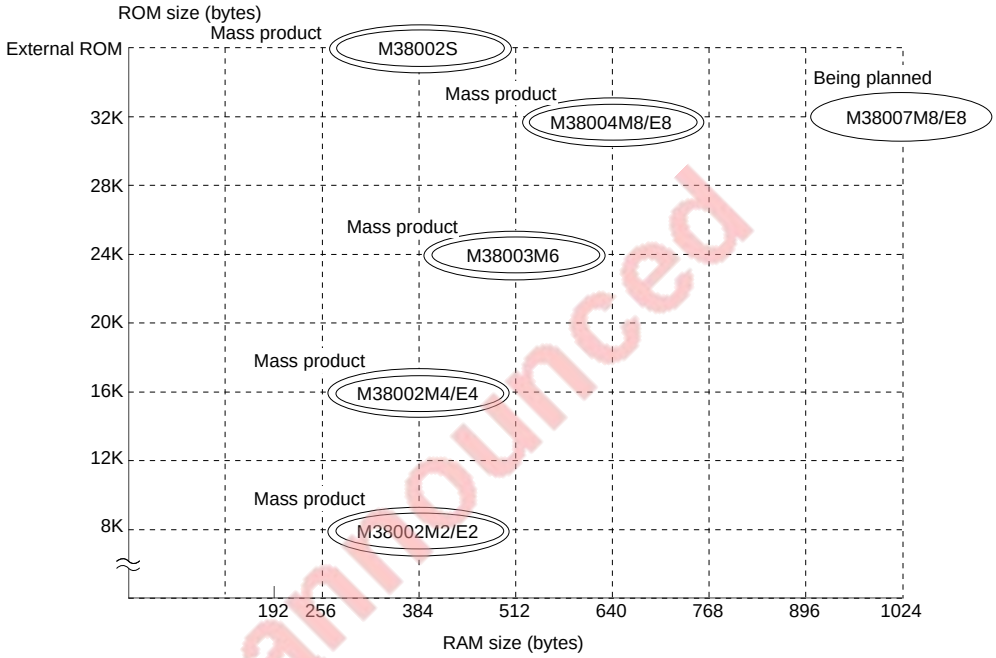
GROUP EXPANSION

Mitsubishi plans to expand the 3800 group as follows:

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

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

Memory Expansion Plan



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

Fig. 5 Memory expansion plan

GROUP EXPANSION

Currently supported products are listed below.

Table 2. List of supported products **As of September 1995**

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

GROUP EXPANSION (EXTENDED OPERATING TEMPERATURE VERSION)

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

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

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

Memory Expansion Plan (Extended operating temperature version)

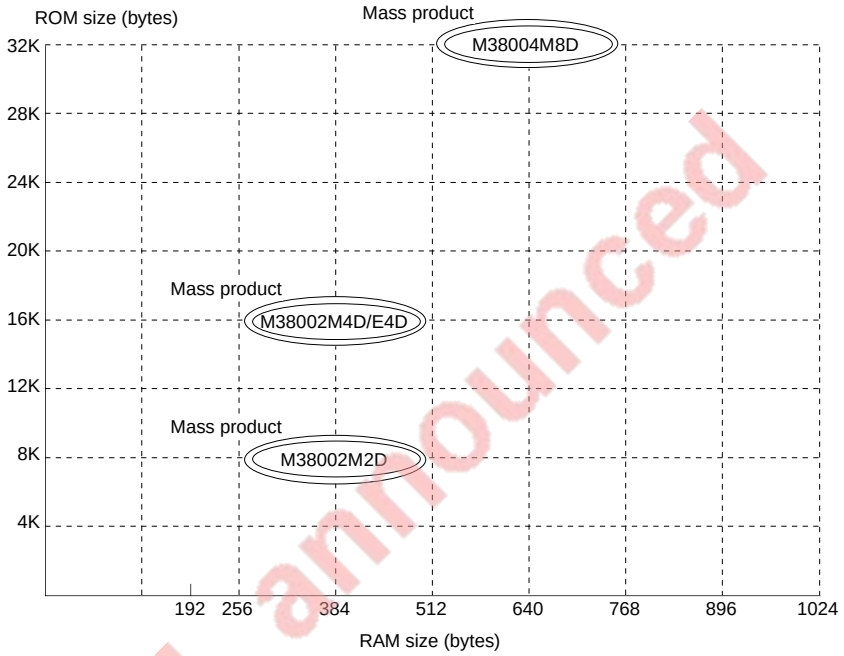


Fig. 6 Memory expansion plan (Extended operating temperature version)

Currently supported products are listed below.

Table 3. List of supported products (Extended operating temperature version) **As of September 1995**

Product name	(P) ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38002M2DXXXSP	8192 (8062)	384	64P4B	Mask ROM version
M38002M2DXXXFP			64P6N-A	Mask ROM version
M38002M4DXXXSP	16384 (16254)	384	64P4B	Mask ROM version
M38002E4DXXXSP				One Time PROM version
M38002E4DSP				One Time PROM version (blank)
M38002M4DXXXFP			64P6N-A	Mask ROM version
M38002E4DXXXFP				One Time PROM version
M38002E4DFP				One Time PROM version (blank)
M38004M8DXXXSP	32768 (32638)	1024	64P4B	Mask ROM version
M38004M8DXXXFP			64P6N-A	Mask ROM version

FUNCTIONAL DESCRIPTION
Central Processing Unit (CPU)

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

Machine-resident 740 family instructions are as follows:

The FST and SLW instruction cannot be used.

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

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

Accumulator (A)

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

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

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

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

Stack pointer (S)

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

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

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

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

Program counter (PC)

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

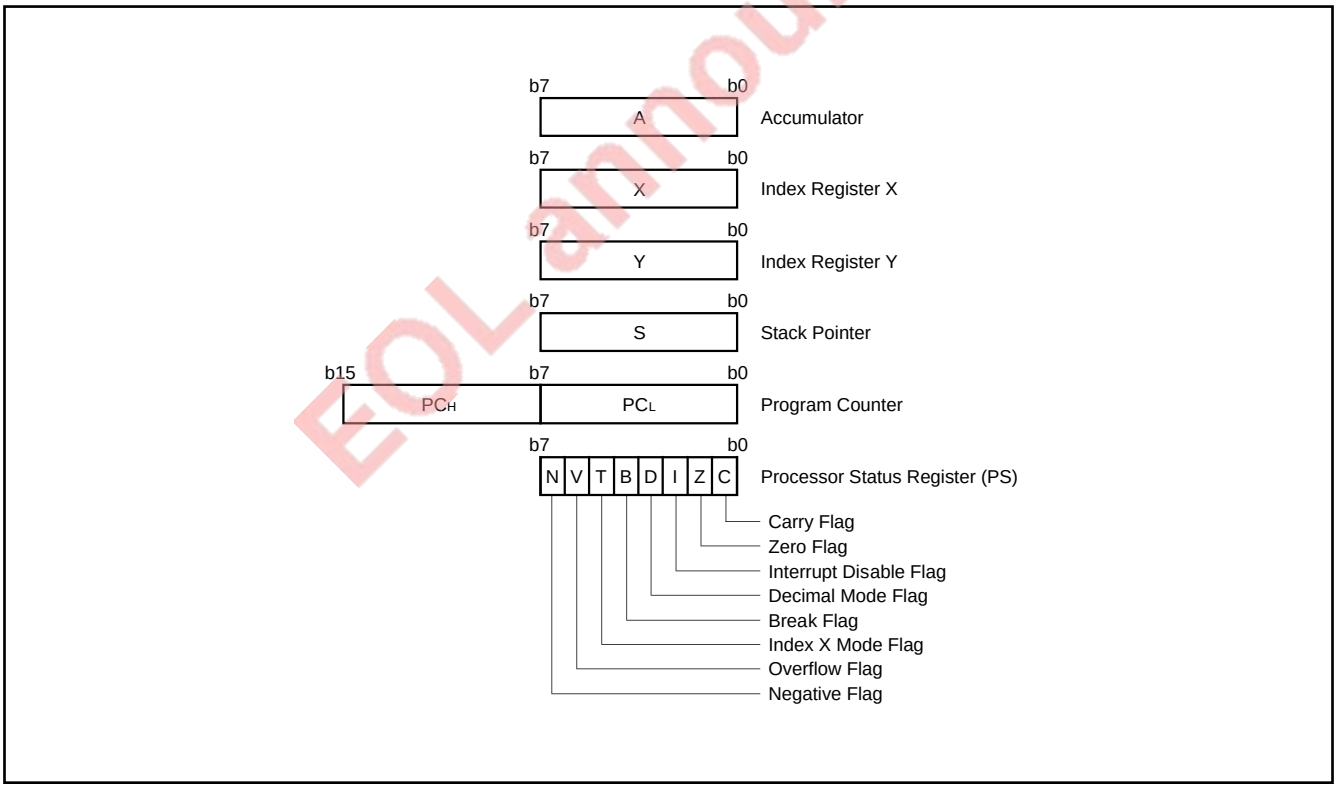
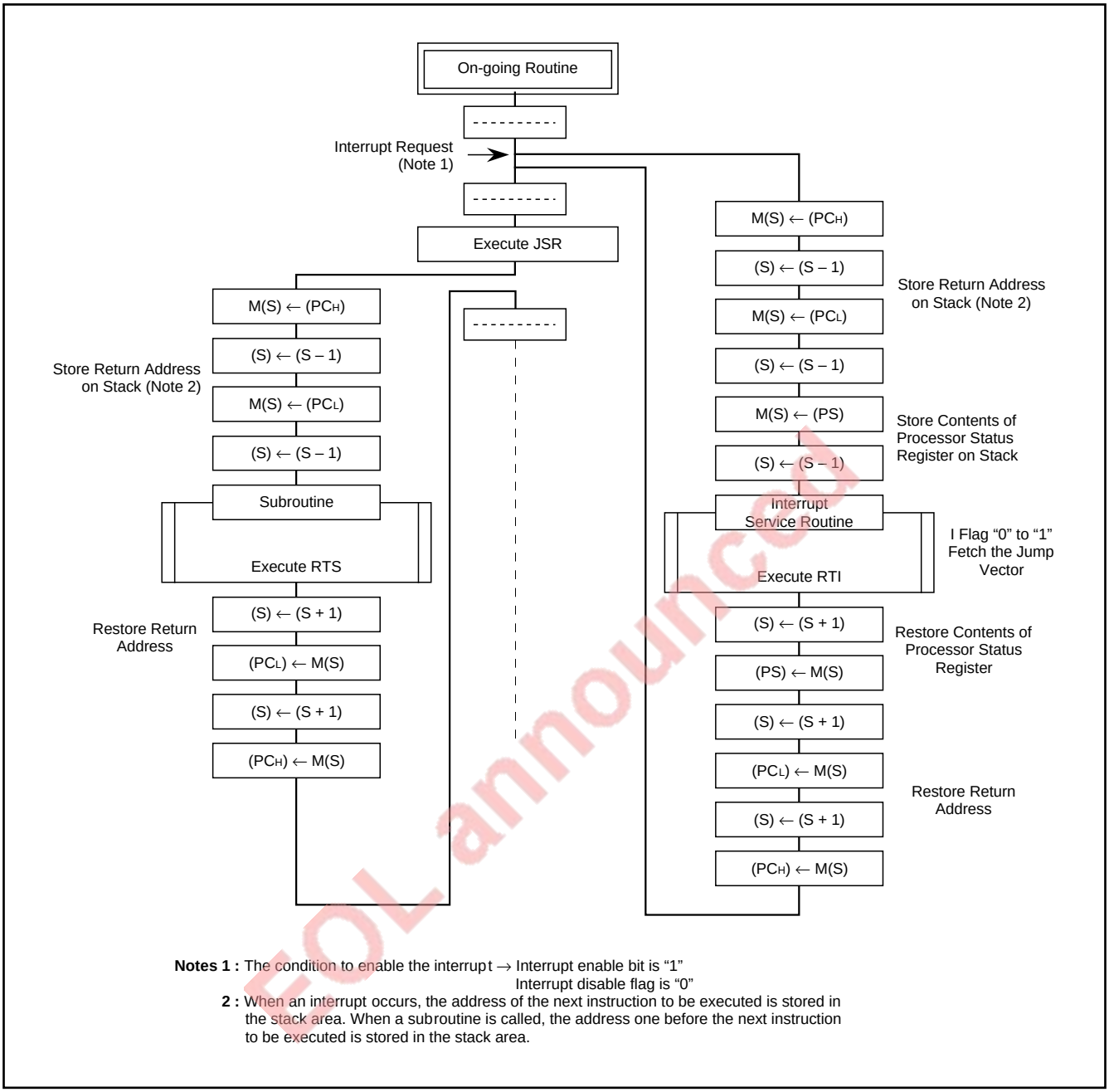


Fig. 7 740 Family CPU register structure



Notes 1 : The condition to enable the interrupt → Interrupt enable bit is “1”
Interrupt disable flag is “0”
2 : When an interrupt occurs, the address of the next instruction to be executed is stored in the stack area. When a subroutine is called, the address one before the next instruction to be executed is stored in the stack area.

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

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

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

Processor status register (PS)

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

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

(1) Carry flag (C)

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

(2) Zero flag (Z)

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

(3) Interrupt disable flag (I)

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

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

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

(4) Decimal mode flag (D)

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

(5) Break flag (B)

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

(6) Index X mode flag (T)

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

(7) Overflow flag (V)

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

(8) Negative flag (N)

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

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

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

CPU mode register

The CPU mode register is allocated at address 003B16.

The CPU mode register contains the stack page selection bit.

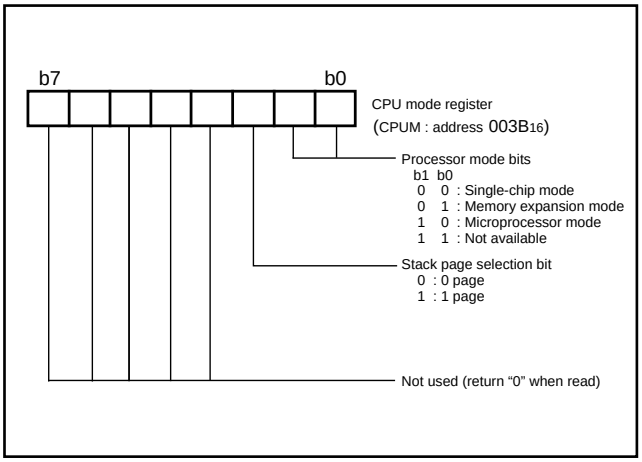


Fig. 9 Structure of CPU mode register

Memory

Special function register (SFR) area

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

RAM

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

ROM

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

Interrupt vector area

The interrupt vector area contains reset and interrupt vectors.

Zero page

The 256 bytes from addresses 0000_{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.

RAM area

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

ROM area

ROM capacity (bytes)	Address YYYY16	Address ZZZZ16
4096	F00016	F08016
8192	E00016	E08016
12288	D00016	D08016
16384	C00016	C08016
20480	B00016	B08016
24576	A00016	A08016
28672	900016	908016
32768	800016	808016

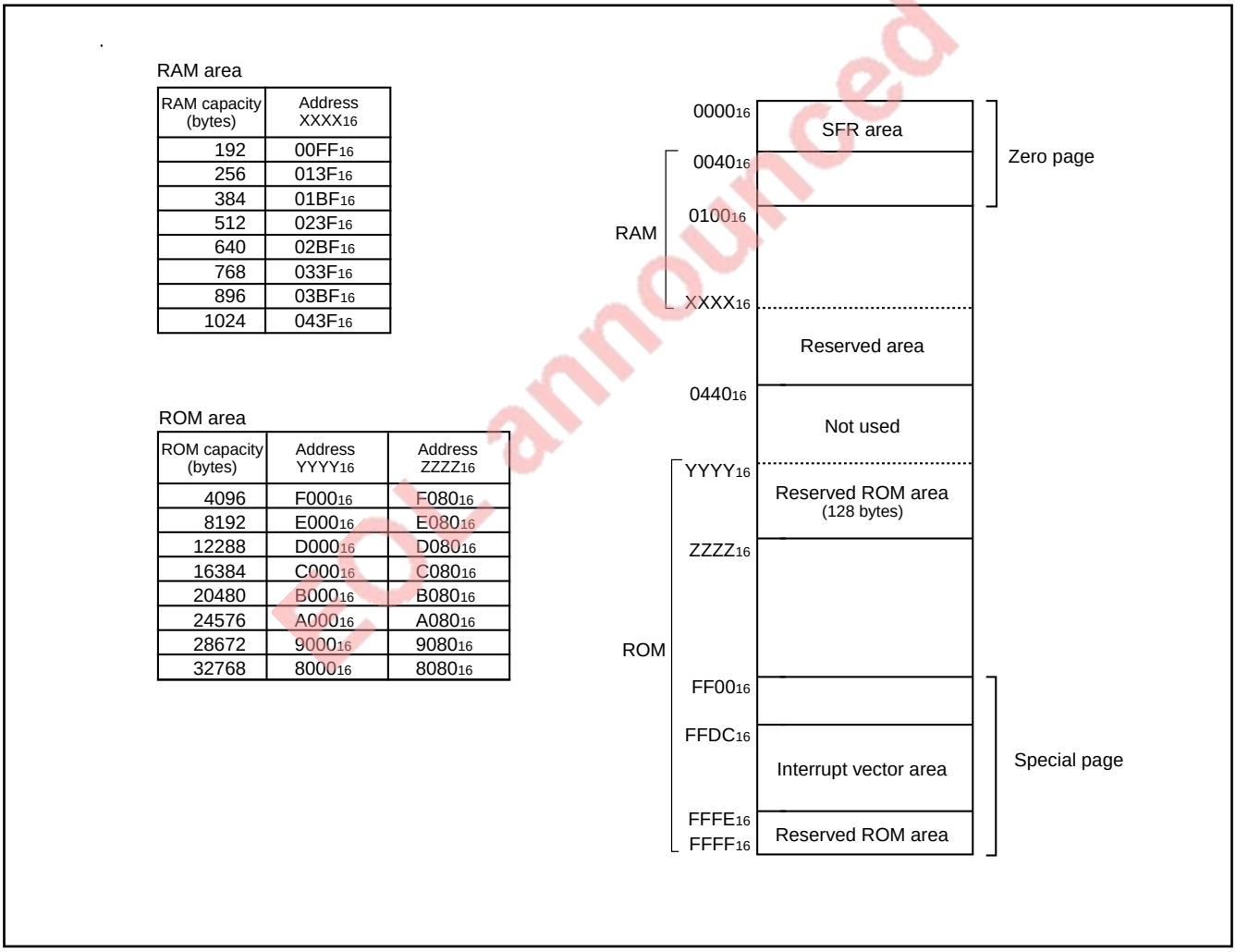


Fig. 10 Memory map diagram

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

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

I/O Ports

Direction registers

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

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

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

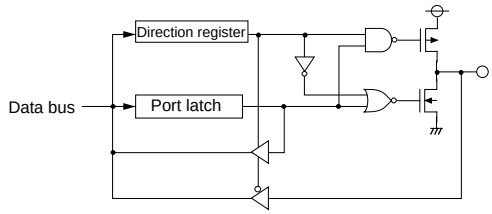
Table 6. List of I/O port functions

Pin	Name	Input/Output	I/O Format	Non-Port Function	Related SFRs	Ref.No.
$P0_0$ – $P0_7$	Port P0	Input/output, individual bits	CMOS 3-state output CMOS compatible input level	Address low-order byte output	CPU mode register	(1)
$P1_0$ – $P1_7$	Port P1	Input/output, individual bits	CMOS 3-state output CMOS compatible input level	Address high-order byte output	CPU mode register	
$P2_0$ – $P2_7$	Port P2	Input/output, individual bits	CMOS 3-state output CMOS compatible input level	Data bus I/O	CPU mode register	
$P3_0$ – $P3_7$	Port P3	Input/output, individual bits	CMOS 3-state output CMOS compatible input level	Control signal I/O	CPU mode register	
$P4_0$,$P4_1$	Port P4	Input/output, individual bits	CMOS 3-state output CMOS compatible input level			
$P4_2$/INT_0, $P4_3$/INT_1				External interrupt input	Interrupt edge selection register	(2)
$P4_4$/RxD, $P4_5$/TxD, $P4_6$/SCLK, $P4_7$/$\overline{SRDY}$				Serial I/O function I/O	Serial I/O control register UART control register	(3) (4) (5) (6)
$P5_0$/INT_2, $P5_1$/INT_3, $P5_2$/INT_4, $P5_3$/INT_5	Port P5	Input/output, individual bits	CMOS 3-state output CMOS compatible input level	External interrupt input	Interrupt edge selection register	(2)
$P5_4$/$CNTR_0$, $P5_5$/$CNTR_1$				Timer X and Timer Y function I/O	Timer XY mode register	(7)
$P5_6$,$P5_7$						(1)
$P6_0$ – $P6_7$	Port P6	Input/output, individual bits	CMOS 3-state output CMOS compatible input level			
$P7_0$, $P7_1$	Port P7	Input/output, individual bits	CMOS 3-state output CMOS compatible input level			

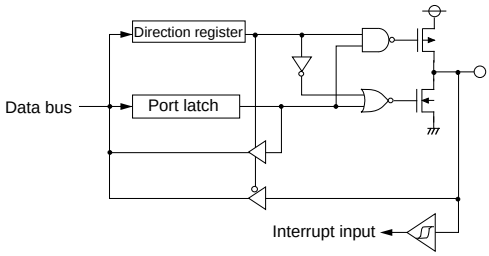
Note 1: For details of the functions of ports P0 to P3 in modes other than single-chip mode, and how to use double-function ports as function I/O ports, refer to the applicable sections.

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

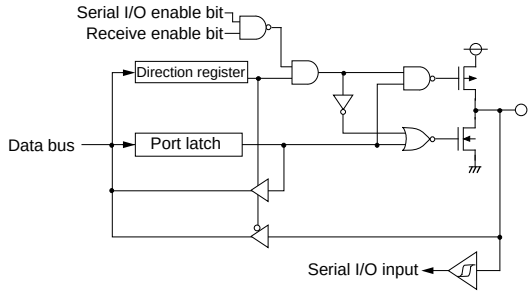
(1) Ports P0, P1, P2, P3, P40, P41, P56, P57, P6, P7



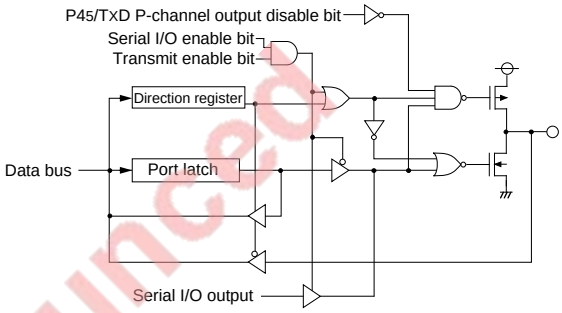
(2) Ports P42, P43, P50 – P53



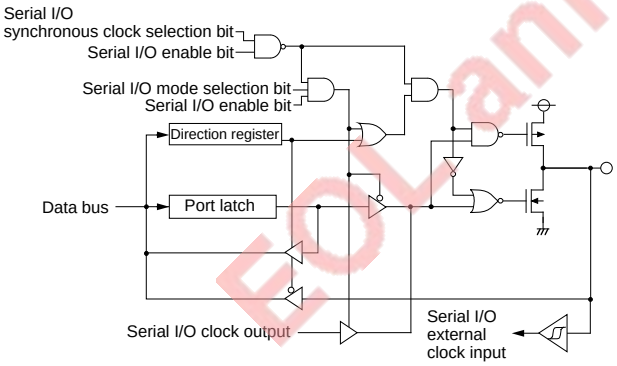
(3) Port P44



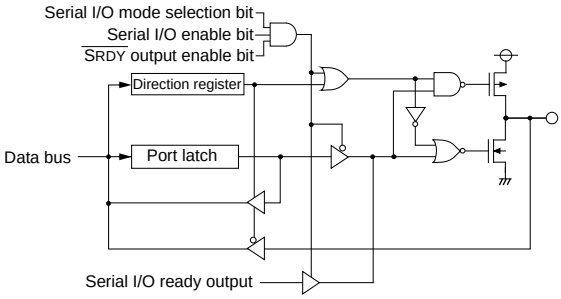
(4) Port P45



(5) Port P46



(6) Port P47



(7) Ports P54, P55

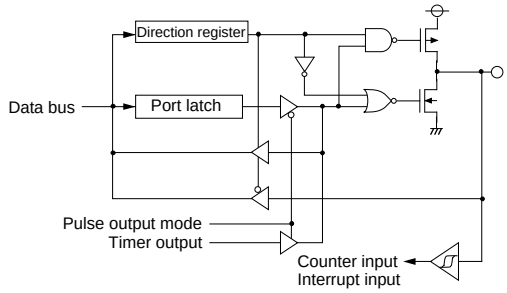


Fig. 12 Port block diagram (single-chip mode)

Interrupts

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

Interrupt control

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

Interrupt enable bits can be set or cleared by software.

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

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

Interrupt operation

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

Notes on use

When the active edge of an external interrupt (INT_0 to INT_5, $CNTR_0$, or $CNTR_1$) is changed, the corresponding interrupt request bit may also be set. Therefore, please take following sequence;

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

Table 7. Interrupt vector addresses and priority

Interrupt Source	Priority	Vector Addresses (Note 1)		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset (Note 2)	1	$FFFD_{16}$	$FFFC_{16}$	At reset	Non-maskable
INT_0	2	$FFFB_{16}$	$FFFA_{16}$	At detection of either rising or falling edge of INT_0 input	External interrupt (active edge selectable)
INT_1	3	$FFF9_{16}$	$FFF8_{16}$	At detection of either rising or falling edge of INT_1 input	External interrupt (active edge selectable)
Serial I/O reception	4	$FFF7_{16}$	$FFF6_{16}$	At completion of serial I/O data reception	Valid when serial I/O is selected
Serial I/O transmission	5	$FFF5_{16}$	$FFF4_{16}$	At completion of serial I/O transfer shift or when transmission buffer is empty	Valid when serial I/O is selected
Timer X	6	$FFF3_{16}$	$FFF2_{16}$	At timer X underflow	
Timer Y	7	$FFF1_{16}$	$FFF0_{16}$	At timer Y underflow	
Timer 1	8	$FFEF_{16}$	$FFEE_{16}$	At timer 1 underflow	STP release timer underflow
Timer 2	9	$FFED_{16}$	$FFEC_{16}$	At timer 2 underflow	
$CNTR_0$	10	$FFEB_{16}$	$FFEA_{16}$	At detection of either rising or falling edge of $CNTR_0$ input	External interrupt (active edge selectable)
$CNTR_1$	11	$FFE9_{16}$	$FFE8_{16}$	At detection of either rising or falling edge of $CNTR_1$ input	External interrupt (active edge selectable)
INT_2	12	$FFE7_{16}$	$FFE6_{16}$	At detection of either rising or falling edge of INT_2 input	External interrupt (active edge selectable)
INT_3	13	$FFE5_{16}$	$FFE4_{16}$	At detection of either rising or falling edge of INT_3 input	External interrupt (active edge selectable)
INT_4	14	$FFE3_{16}$	$FFE2_{16}$	At detection of either rising or falling edge of INT_4 input	External interrupt (active edge selectable)
INT_5	15	$FFE1_{16}$	$FFE0_{16}$	At detection of either rising or falling edge of INT_5 input	External interrupt (active edge selectable)
BRK instruction	16	$FFDD_{16}$	$FFDC_{16}$	At BRK instruction execution	Non-maskable software interrupt

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

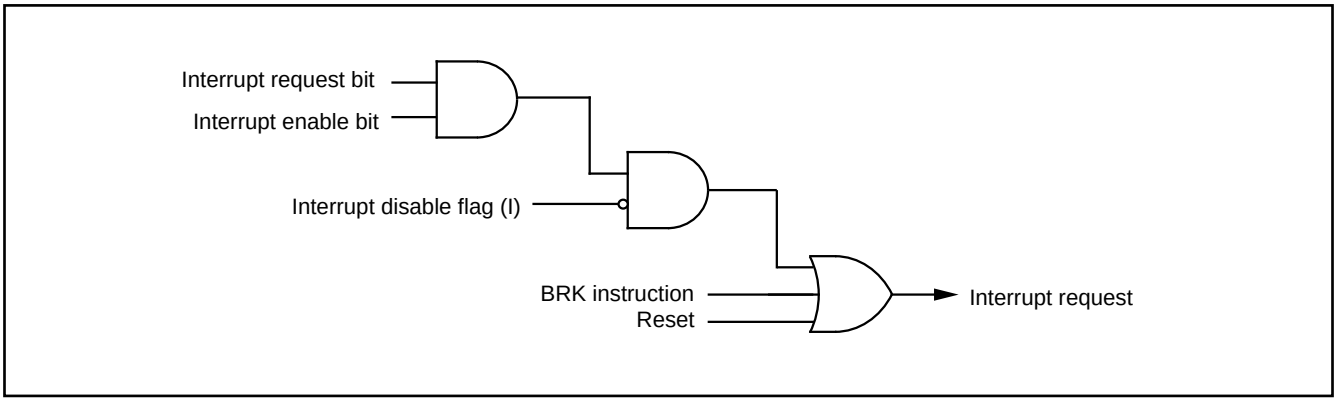


Fig. 13 Interrupt control

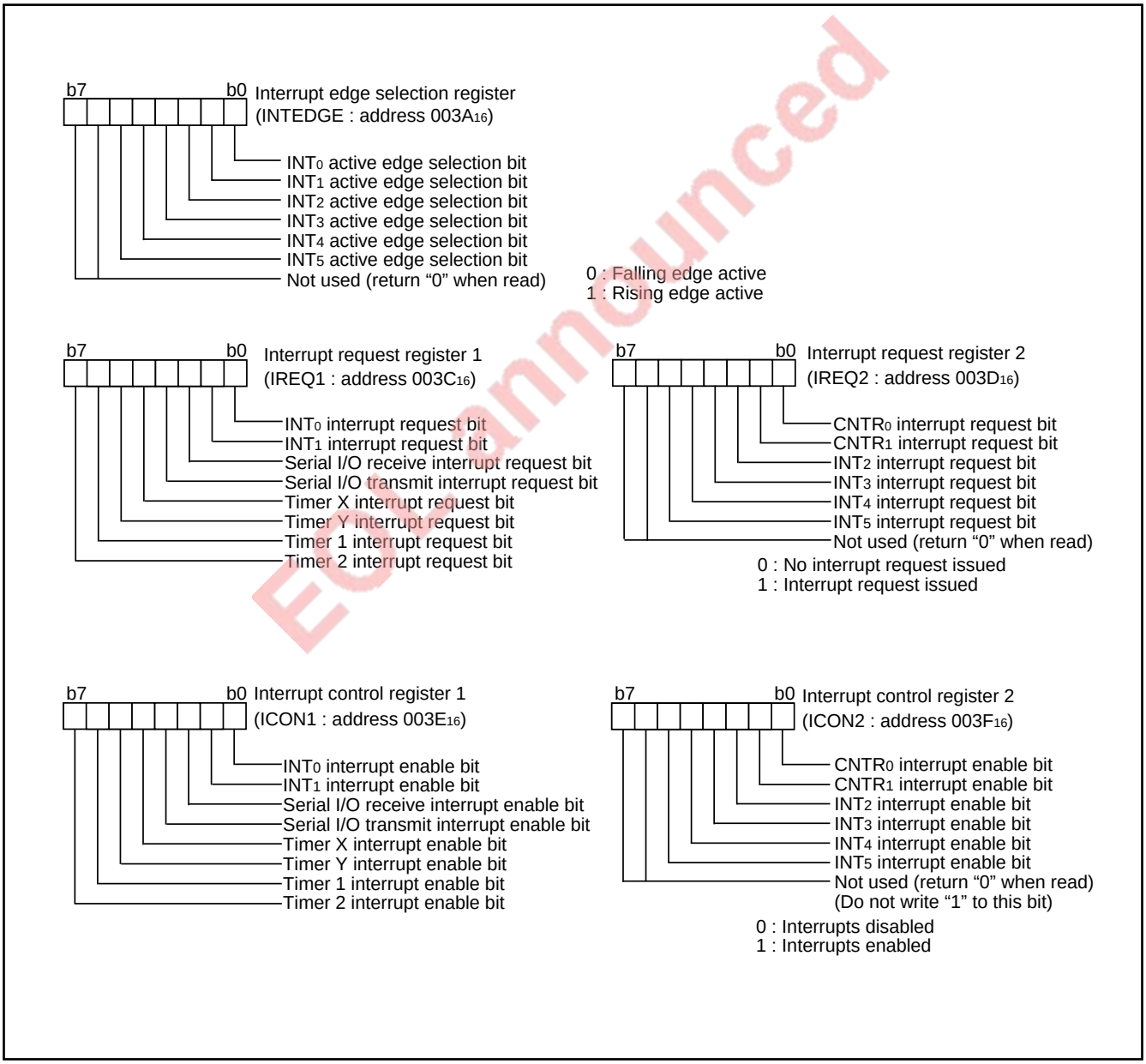


Fig. 14 Structure of interrupt-related registers

Timers

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

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

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

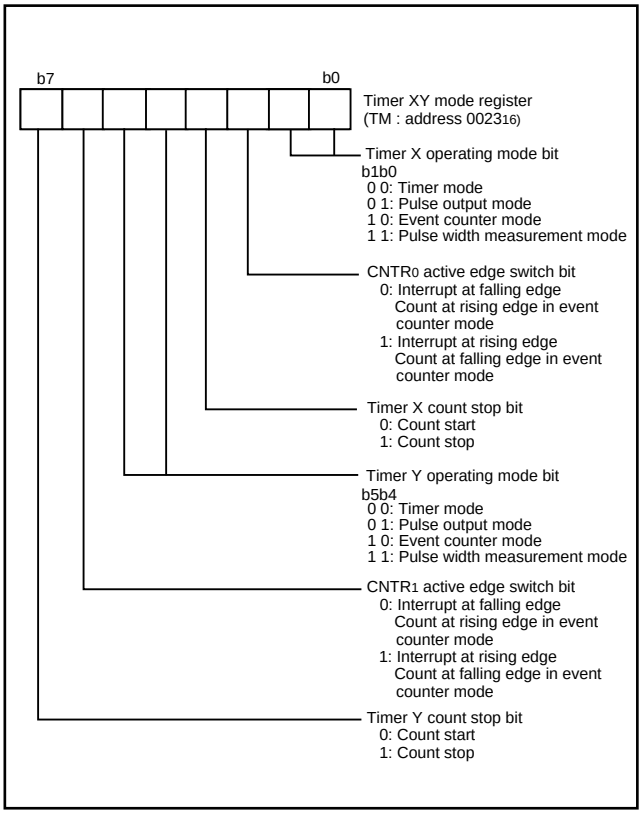


Fig. 15 Structure of timer XY register

Timer 1 and Timer 2

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

Timer X and Timer Y

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

Timer Mode

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

Pulse Output Mode

Timer X (or timer Y) counts f(X_{IN})/16. Whenever the contents of the timer reach "00_{16}", the signal output from the $CNTR_0$ (or $CNTR_1$) pin is inverted. If the $CNTR_0$ (or $CNTR_1$) active edge switch bit is "0", output begins at " H".

If it is "1", output starts at "L". When using a timer in this mode, set the corresponding port $P5_4$ (or port $P5_5$) direction register to output mode.

Event Counter Mode

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

Pulse Width Measurement Mode

If the $CNTR_0$ (or $CNTR_1$) active edge selection bit is "0", the timer counts at the oscillation frequency divided by 16 while the $CNTR_0$ (or $CNTR_1$) pin is at "H". If the $CNTR_0$ (or $CNTR_1$) active edge switch bit is "1", the count continues during the time that the $CNTR_0$ (or $CNTR_1$) pin is at "L".

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

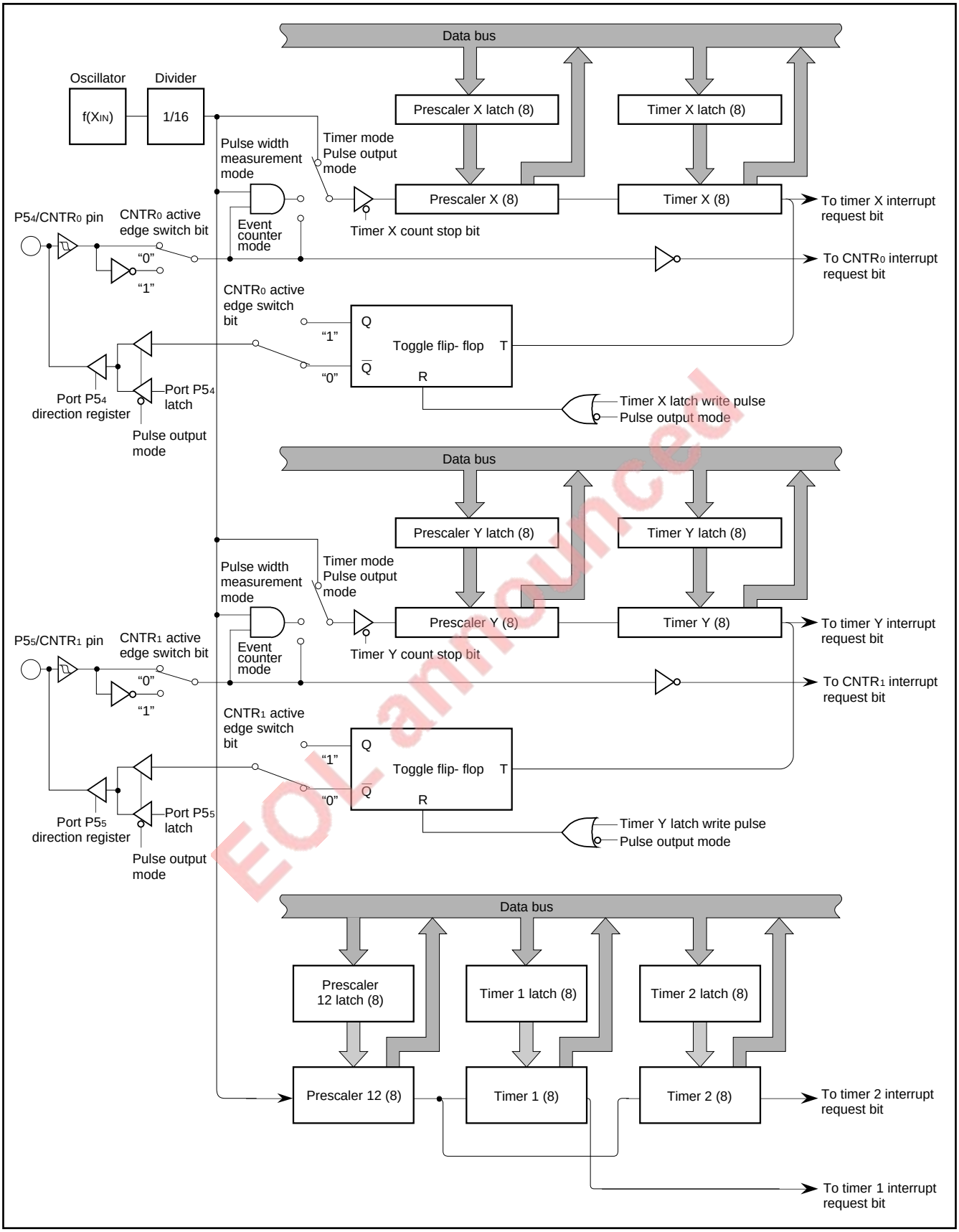


Fig. 16 Block diagram of timer X, timer Y, timer 1, and timer 2

Serial I/O

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

Clock synchronous serial I/O mode

Clock synchronous serial I/O mode can be selected by setting the mode selection bit of the serial I/O control register to "1".

For clock synchronous serial I/O, the transmitter and the receiver must use the same clock. If an internal clock is used, transfer is started by a write signal to the TB/RB (address 0018_{16}).

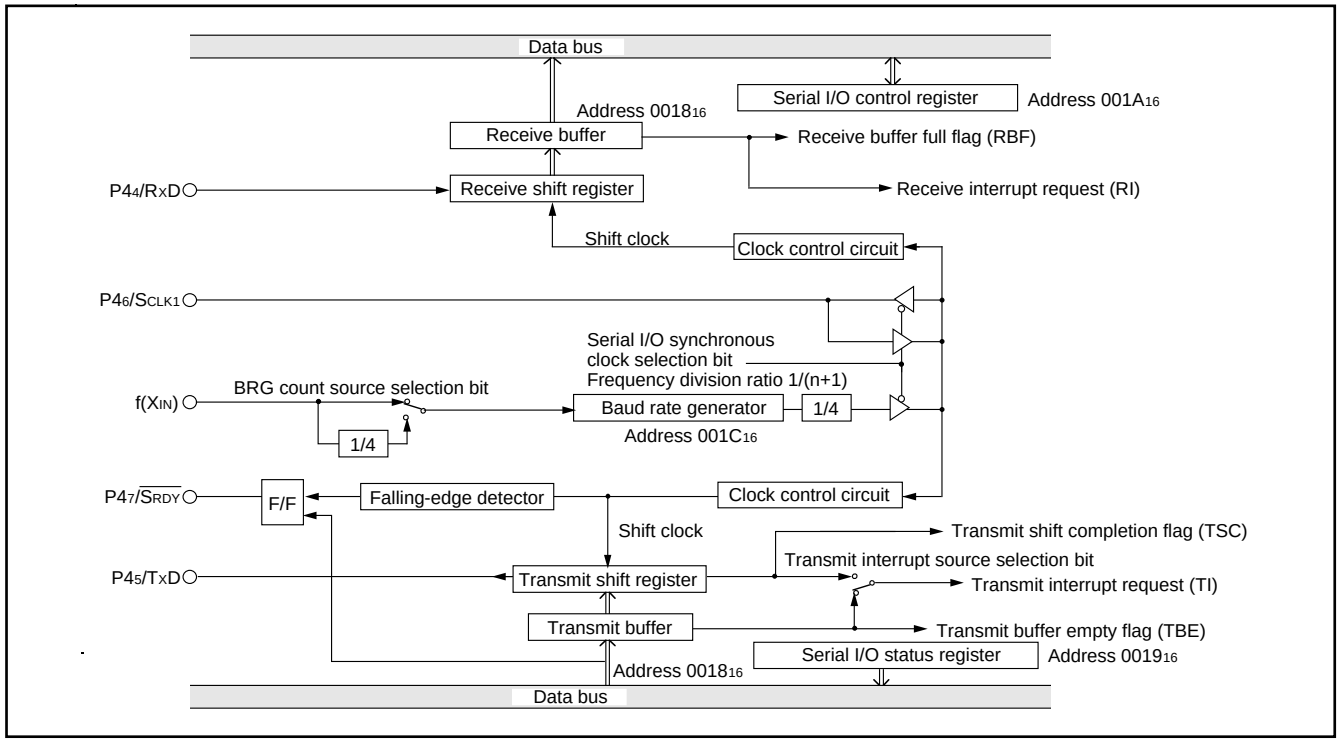
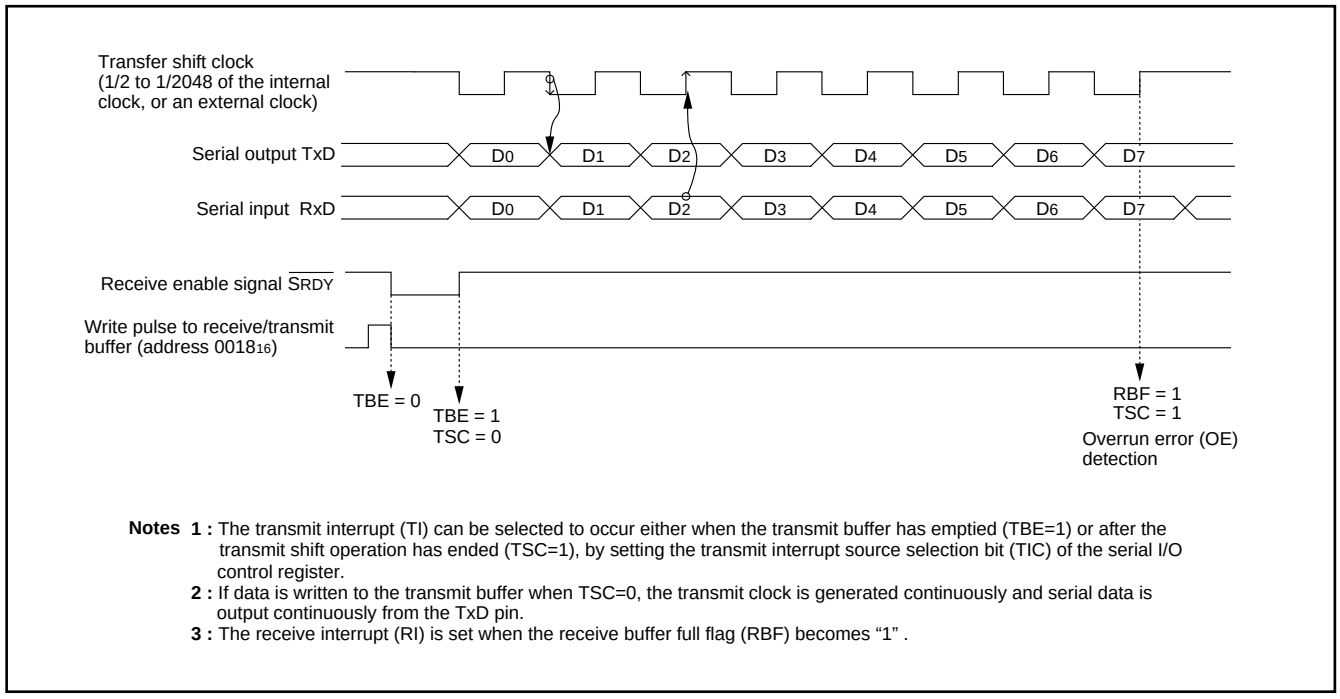


Fig. 17 Block diagram of clock synchronous serial I/O



Notes 1 : The transmit interrupt (TI) can be selected to occur either when the transmit buffer has emptied (TBE=1) or after the transmit shift operation has ended (TSC=1), by setting the transmit interrupt source selection bit (TIC) of the serial I/O control register.

2 : If data is written to the transmit buffer when TSC=0, the transmit clock is generated continuously and serial data is output continuously from the TxD pin.

3 : The receive interrupt (RI) is set when the receive buffer full flag (RBF) becomes "1" .

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

Asynchronous serial I/O (UART) mode

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

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

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

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

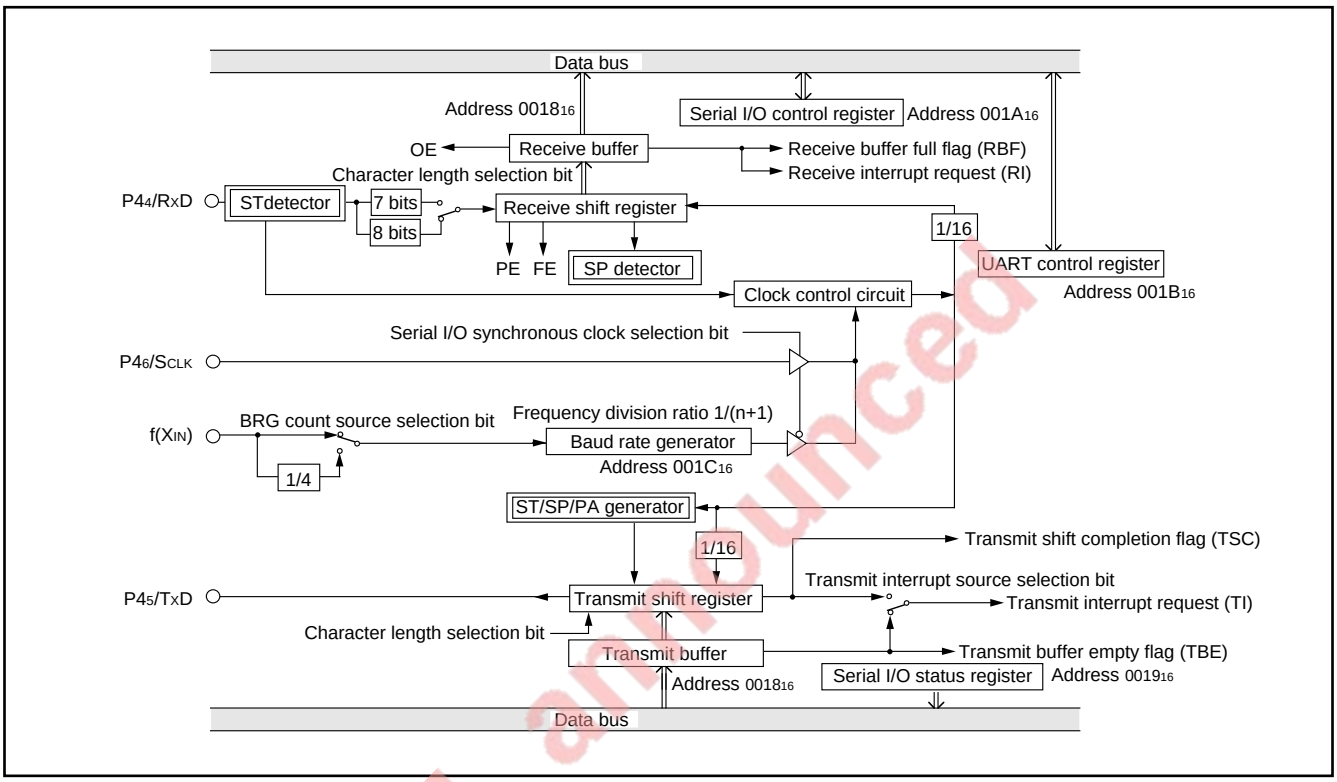
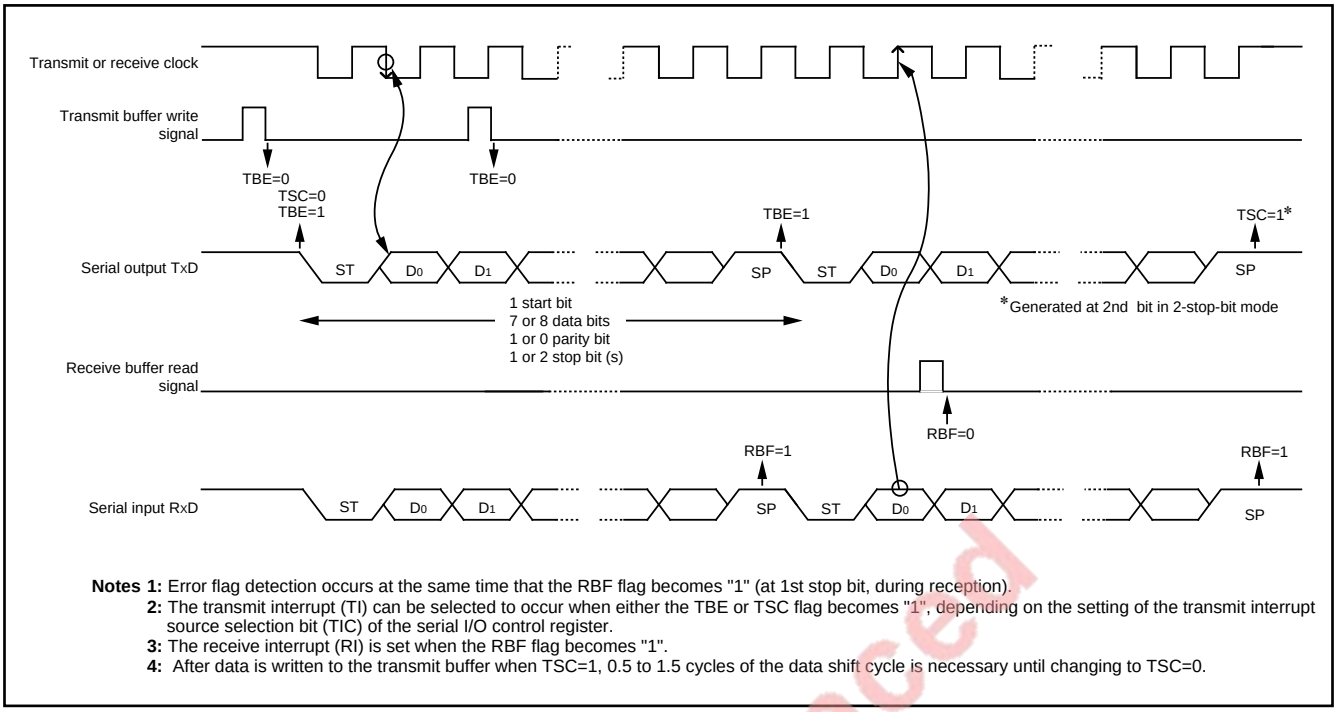


Fig. 19 Block diagram of UART serial I/O



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

Fig. 20 Operation of UART serial I/O function

Serial I/O control register (SIOCON) 001A16

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

UART control register (UARTCON) 001B16

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

Serial I/O status register (SIOSTS) 001916

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

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

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

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

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

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

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

Baud rate generator (BRG) 001C16

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

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

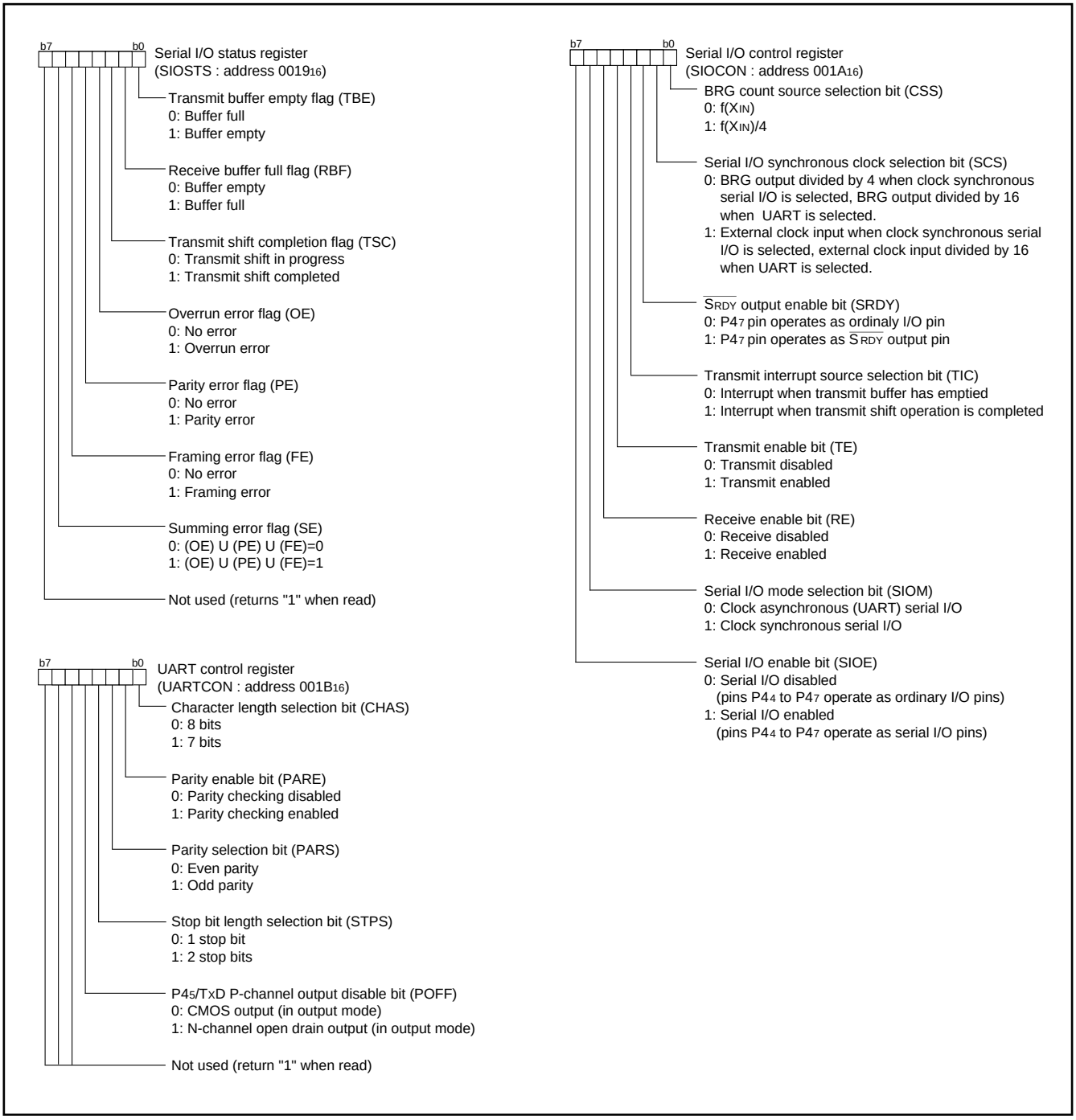


Fig. 21 Structure of serial I/O control registers

Reset Circuit

To reset the microcomputer, the $\overline{\text{RESET}}$ pin should be held at an "L" level for 2 μs or more. Then the $\overline{\text{RESET}}$ pin is returned to an "H" level (the power source voltage should be between 3.0 V and 5.5 V, and between 4.0 V and 5.5 V for extended operating temperature version), reset is released. Internal operation does not begin until after 8 to 13 X_{IN} clock cycles are completed. After the reset is completed, the program starts from the address contained in address FFFD16 (high-order byte) and address FFFC16 (low-order byte).

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

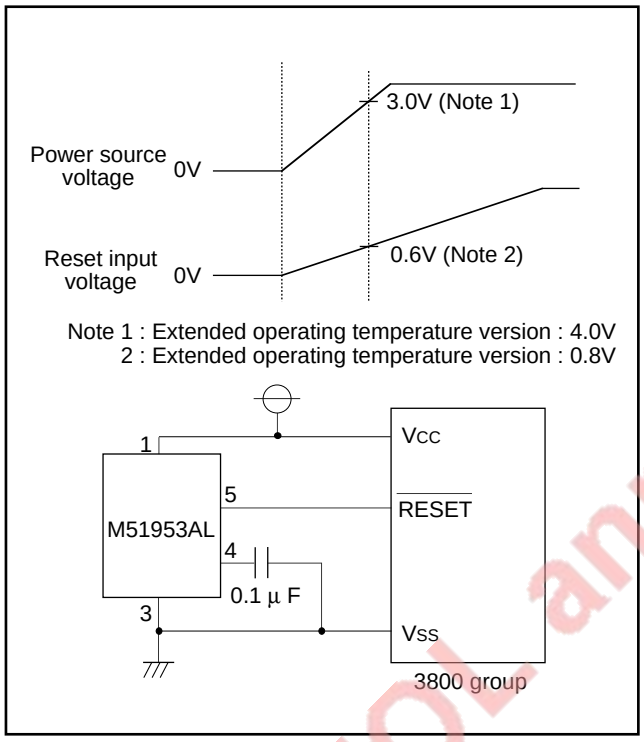


Fig. 22 Example of reset circuit

	Address	Register contents
(1) Port P0 direction register	(0001₁₆)	00₁₆
(2) Port P1 direction register	(0003₁₆)	00₁₆
(3) Port P2 direction register	(0005₁₆)	00₁₆
(4) Port P3 direction register	(0007₁₆)	00₁₆
(5) Port P4 direction register	(0009₁₆)	00₁₆
(6) Port P5 direction register	(000B₁₆)	00₁₆
(7) Port P6 direction register	(000D₁₆)	00₁₆
(8) Port P7 direction register	(000F₁₆)	00₁₆
(9) Serial I/O status register	(0019₁₆)	1 0 0 0 0 0 0 0
(10) Serial I/O control register	(001A₁₆)	00₁₆
(11) UART control register	(001B₁₆)	1 1 1 0 0 0 0 0
(12) Prescaler 12	(0020₁₆)	FF₁₆
(13) Timer 1	(0021₁₆)	01₁₆
(14) Timer 2	(0022₁₆)	FF₁₆
(15) Timer XY mode register	(0023₁₆)	00₁₆
(16) Prescaler X	(0024₁₆)	FF₁₆
(17) Timer X	(0025₁₆)	FF₁₆
(18) Prescaler Y	(0026₁₆)	FF₁₆
(19) Timer Y	(0027₁₆)	FF₁₆
(20) Interrupt edge selection register	(003A₁₆)	00₁₆
(21) CPU mode register	(003B₁₆)	0 0 0 0 0 0 ✻ 0
(22) Interrupt request register 1	(003C₁₆)	00₁₆
(23) Interrupt request register 2	(003D₁₆)	00₁₆
(24) Interrupt control register 1	(003E₁₆)	00₁₆
(25) Interrupt control register 2	(003F₁₆)	00₁₆
(26) Processor status register	(PS)	× × × × × 1 × ×
(27) Program counter	(PC_H)	Contents of address FFFD₁₆
	(PC_L)	Contents of address FFFC₁₆

Note. × : Undefined

✻ : The initial values of CM_1 are determined by the level at the CNV_{SS} pin.

The contents of all other registers and RAM are undefined after a reset, so they must be initialized by software.

Fig. 23 Internal status of microcomputer after reset

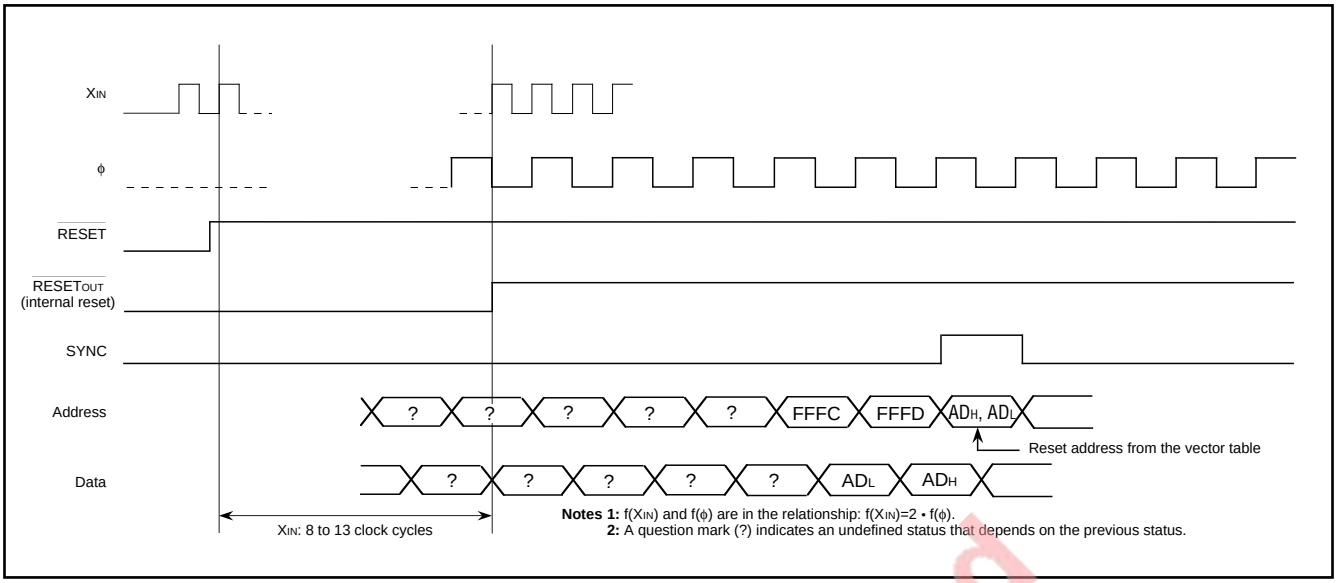


Fig. 24 Timing of reset

Clock Generating Circuit

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

Oscillation control

Stop Mode

If the STP instruction is executed, the internal clock ϕ stops at "H". Timer 1 is set to "01$_{16}$" and prescaler 12 is set to "FF$_{16}$".
Oscillator restarts when an external interrupt is received, but the internal clock ϕ remains at "H" until timer 1 underflows.
This allows time for the clock circuit oscillation to stabilize.
If oscillator is restarted by a reset, no wait time is generated, so keep the $\overline{\text{RESET}}$ pin at "L" level until oscillation has stabilized.

Wait Mode

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

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

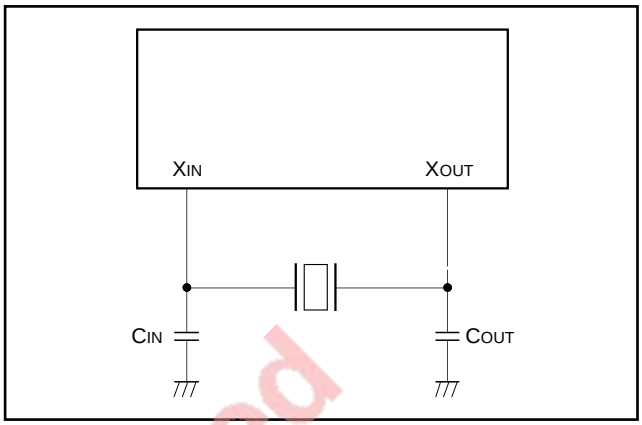


Fig. 25 Ceramic resonator circuit

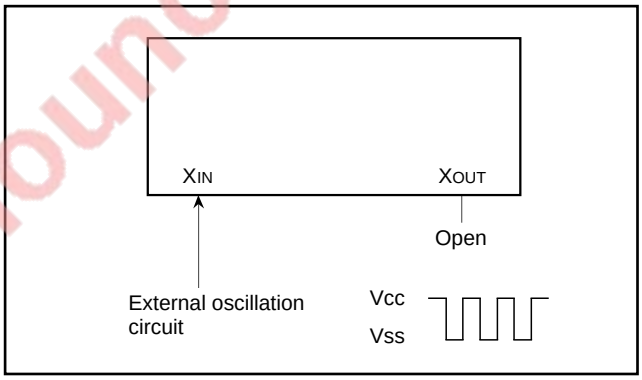


Fig. 26 External clock input circuit

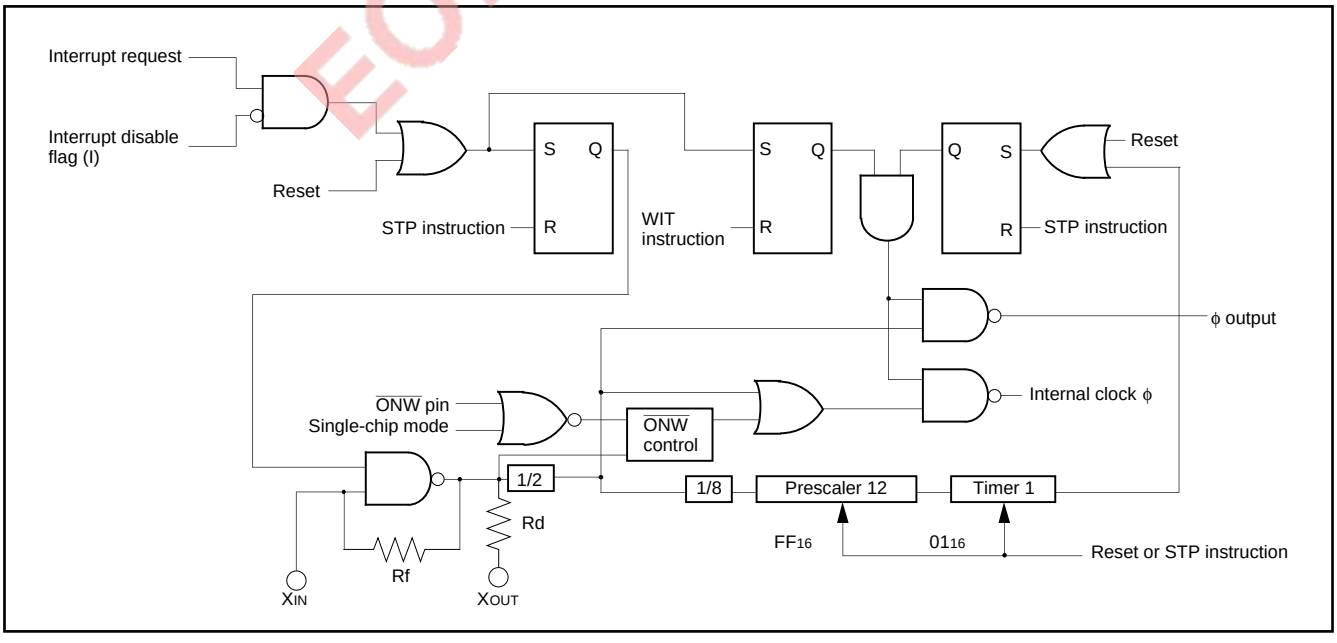


Fig. 27 Block diagram of clock generating circuit

Processor Modes

Single-chip mode, memory expansion mode, and microprocessor mode can be selected by changing the contents of the processor mode bits CM_0 and CM_1 (bits 0 and 1 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 8. Functions of ports in memory expansion mode and microprocessor mode

Port Name	Function
Port P0	Outputs low-order byte of address.
Port P1	Outputs high-order byte of address.
Port P2	Operates as I/O pins for data D_7 to D_0 (including instruction codes).
Port P3	$P3_0$ and $P3_1$ function only as output pins (except that the port latch cannot be read). $P3_2$ is the $\overline{ONW}$ input pin. $P3_3$ is the $\overline{RESET_{OUT}}$ output pin. (Note) $P3_4$ is the ϕ output pin. $P3_5$ is the SYNC output pin. $P3_6$ is the $\overline{WR}$ output pin, and $P3_7$ 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.

Single-Chip Mode

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

Memory Expansion Mode

Select this mode by setting the processor mode bits 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. Internal ROM will take precedence over external memory if addresses conflict.

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.

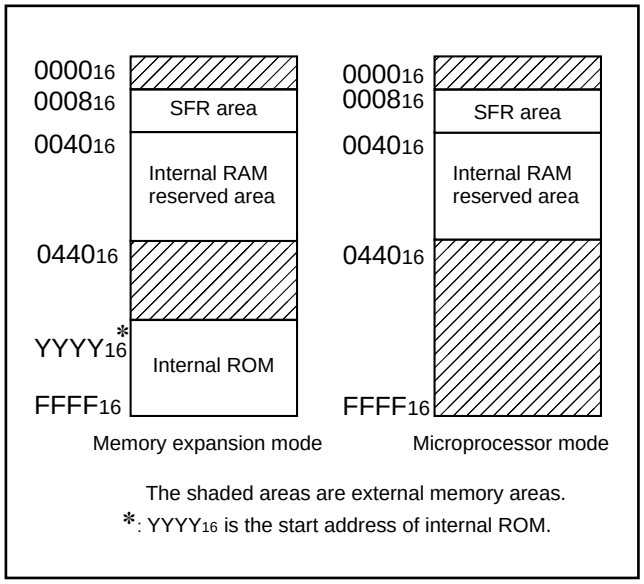


Fig. 28 Memory maps in various processor modes

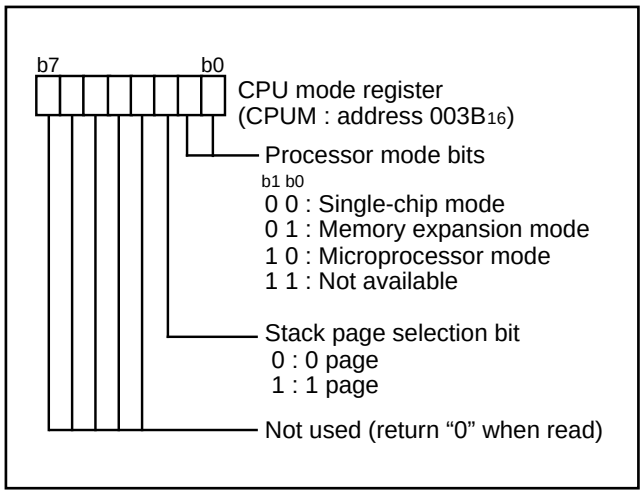
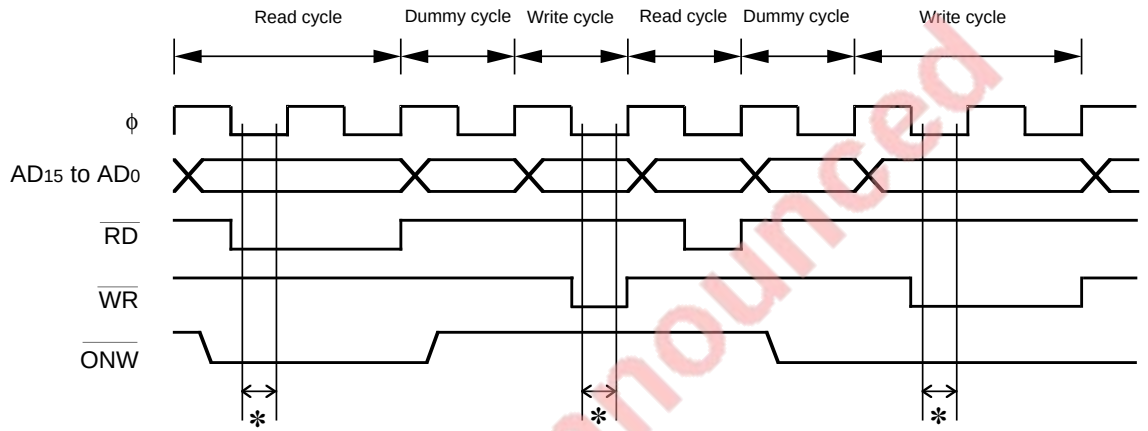


Fig. 29 Structure of CPU mode register

Bus control with memory expansion

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

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



✻ : Period during which $\overline{ONW}$ input signal is received
During this period, the $\overline{ONW}$ signal must be fixed at either "H" or "L". At all other times, the input level of the $\overline{ONW}$ signal has no affect on operations.
The bus cycles is not extended for an address in the area 0008_{16} to $043F_{16}$, regardless of whether the $\overline{ONW}$ signal is received.

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

NOTES ON PROGRAMMING

Processor Status Register

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

Interrupts

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

Decimal Calculations

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

In decimal mode, the values of the negative (N), overflow (V), and zero (Z) flags are invalid.
The carry flag can be used to indicate whether a carry or borrow has occurred. Initialize the carry flag before each calculation. Clear the carry flag before an ADC and set the flag before an SBC.

Timers

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

Multiplication and Division Instructions

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

Ports

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

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

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

Serial I/O

In clock synchronous serial I/O, if the receive side is using an external clock and it is to output the $\overline{\text{SRDY}}$ signal, set the transmit enable bit, the receive enable bit, and the $\overline{\text{SRDY}}$ output enable bit to "1".
Serial I/O continues to output the final bit from the TxD pin after transmission is completed.

Instruction Execution Time

The instruction execution time is obtained by multiplying the frequency of the internal clock ϕ by the number of cycles needed to execute an instruction.
The number of cycles required to execute an instruction is shown in the list of machine instructions.
The frequency of the internal clock ϕ is half of the XIN frequency.
When the $\overline{\text{ONW}}$ function is used in modes other than single-chip mode, the frequency of the internal clock ϕ may be one fourth the XIN frequency.

Memory Expansion Mode and Microprocessor Mode

Execute the LDM or STA instruction for writing to port P3 (address 0006_{16}) in memory expansion mode and microprocessor mode.
Set areas which can be read out and write to port P3 (address 0006_{16}) in a memory, using the read-modify-write instruction (SEB, CLB).

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. Mark Specification Form
3. Data to be written to ROM, in EPROM form (three identical copies)

ROM PROGRAMMING METHOD

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

Table 9. Programming adapter

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

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

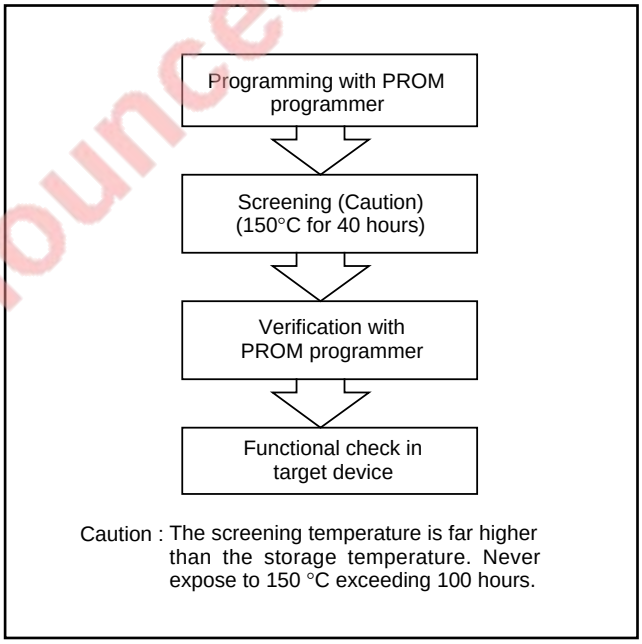


Fig. 31 Programming and testing of One Time PROM version

FUNCTIONAL DESCRIPTION SUPPLEMENT

Interrupt

3800 group permits interrupts on the basis of 15 sources. It is vector interrupts with a fixed priority system. Accordingly, when two or more interrupt requests occur during the same sampling, the higher-priority interrupt is accepted first. This priority is determined by hardware, but variety of priority processing can be performed by software, using an interrupt enable bit and an interrupt disable flag.

For interrupt sources, vector addresses and interrupt priority, refer to **"Table 10."**

Table 10. Interrupt sources, vector addresses and interrupt priority

Priority	Interrupt sources	Vector addresses		Remarks
		High-order	Low-order	
1	Reset (Note)	$FFFD_{16}$	$FFFC_{16}$	Non-maskable
2	INT_0 interrupt	$FFFB_{16}$	$FFFA_{16}$	External interrupt (active edge selectable)
3	INT_1 interrupt	$FFF9_{16}$	$FFF8_{16}$	External interrupt (active edge selectable)
4	Serial I/O receive interrupt	$FFF7_{16}$	$FFF6_{16}$	Valid when serial I/O is selected
5	Serial I/O transmit interrupt	$FFF5_{16}$	$FFF4_{16}$	Valid when serial I/O is selected
6	Timer X interrupt	$FFF3_{16}$	$FFF2_{16}$	
7	Timer Y interrupt	$FFF1_{16}$	$FFF0_{16}$	
8	Timer 1 interrupt	$FFEF_{16}$	$FFEE_{16}$	STP release timer underflow
9	Timer 2 interrupt	$FFED_{16}$	$FFEC_{16}$	
10	$CNTR_0$ interrupt	$FFEB_{16}$	$FFEA_{16}$	External interrupt (active edge selectable)
11	$CNTR_1$ interrupt	$FFE9_{16}$	$FFE8_{16}$	External interrupt (active edge selectable)
12	INT_2 interrupt	$FFE7_{16}$	$FFE6_{16}$	External interrupt (active edge selectable)
13	INT_3 interrupt	$FFE5_{16}$	$FFE4_{16}$	External interrupt (active edge selectable)
14	INT_4 interrupt	$FFE3_{16}$	$FFE2_{16}$	External interrupt (active edge selectable)
15	INT_5 interrupt	$FFE1_{16}$	$FFE0_{16}$	External interrupt (active edge selectable)
16	BRK instruction interrupt	$FFDD_{16}$	$FFDC_{16}$	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 32 shows a timing chart after an interrupt occurs, and Figure 33 shows the time up to execution of the interrupt processing routine.

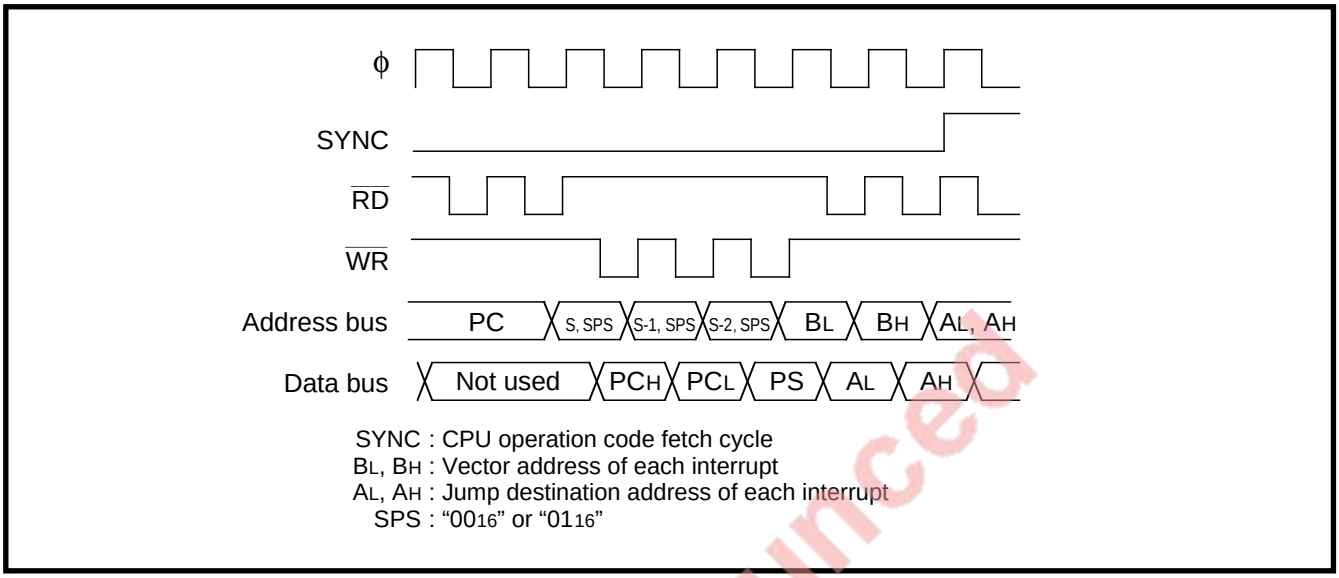


Fig. 32 Timing chart after an interrupt occurs

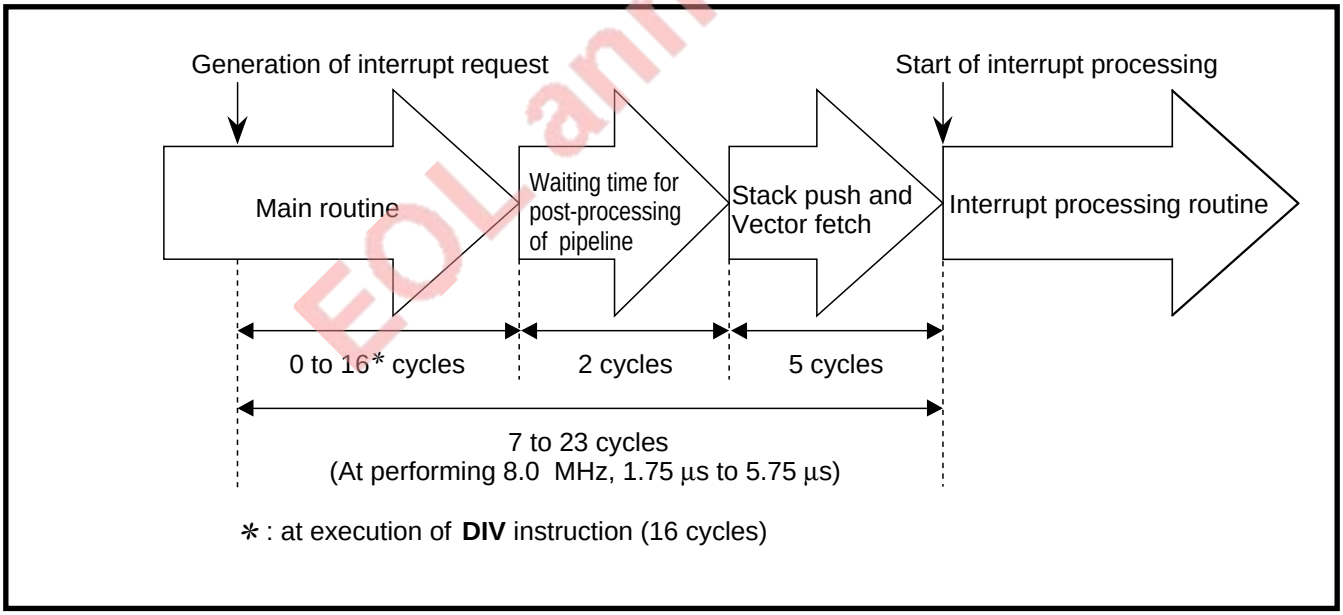


Fig. 33 Time up to execution of the interrupt processing routine

CHAPTER 2
APPLICATION

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

2.1 I/O port

2.1.1 Memory map of I/O port

0000_{16}	Port P0 (P0)
0001_{16}	Port P0 direction register (P0D)
0002_{16}	Port P1 (P1)
0003_{16}	Port P1 direction register (P1D)
0004_{16}	Port P2 (P2)
0005_{16}	Port P2 direction register (P2D)
0006_{16}	Port P3 (P3)
0007_{16}	Port P3 direction register (P3D)
0008_{16}	Port P4 (P4)
0009_{16}	Port P4 direction register (P4D)
$000A_{16}$	Port P5 (P5)
$000B_{16}$	Port P5 direction register (P5D)
$000C_{16}$	Port P6 (P6)
$000D_{16}$	Port P6 direction register (P6D)
$000E_{16}$	Port P7 (P7)
$000F_{16}$	Port P7 direction register (P7D)

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

2.1.2 Related registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (Pi) (i = 0, 1, 2, 3, 4, 5, 6, 7)
[Address : 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{16}, $0A_{16}$, $0C_{16}$, $0E_{16}$]

B	Name	Function	At reset	R	W
0	Port Pi_0	• In output mode Write } Port latch Read } • In input mode Write : Port latch Read : Value of pins (**Note**)	?	○	○
1	Port Pi_1		?	○	○
2	Port Pi_2		?	○	○
3	Port Pi_3		?	○	○
4	Port Pi_4		?	○	○
5	Port Pi_5		?	○	○
6	Port Pi_6		?	○	○
7	Port Pi_7		?	○	○

Note : Port P7 register [Address : $0E_{16}$]
Port P7 is a 2-bit port ($P7_0$, $P7_1$). Accordingly, when bits 2 to 7 are read out, the contents are "0."

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

Port Pi direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi direction register (PiD) (i =0, 1, 2, 3, 4, 5, 6, 7)
[Address : 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{16}, $0B_{16}$, $0D_{16}$, $0F_{16}$]

B	Name	Function	At reset	R	W
0	Port Pi direction register	0 : Port Pi_0 input mode 1 : Port Pi_0 output mode	0	×	○
1		0 : Port Pi_1 input mode 1 : Port Pi_1 output mode	0	×	○
2		0 : Port Pi_2 input mode 1 : Port Pi_2 output mode (**Note**)	0	×	○
3		0 : Port Pi_3 input mode 1 : Port Pi_3 output mode (**Note**)	0	×	○
4		0 : Port Pi_4 input mode 1 : Port Pi_4 output mode (**Note**)	0	×	○
5		0 : Port Pi_5 input mode 1 : Port Pi_5 output mode (**Note**)	0	×	○
6		0 : Port Pi_6 input mode 1 : Port Pi_6 output mode (**Note**)	0	×	○
7		0 : Port Pi_7 input mode 1 : Port Pi_7 output mode (**Note**)	0	×	○

Note : Port P7 direction register [Address : $0F_{16}$]
Port P7 is a 2-bit port ($P7_0$, $P7_1$). Accordingly, these bits do not have a direction register function.

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

2.1 I/O port

2.1.3 Handling of unused pins

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

Name of Pins/Ports	Handling
P0, P1, P2, P3, P4, P5, P6, P7	• Set to the input mode and connect to V_{CC} or V_{SS} through a resistor of 1 kΩ to 10 kΩ. • Set to the output mode and open at "L" or "H."
X_{OUT}	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
$P3_0$, $P3_1$	Open
P4, P5, P6, P7	• Set to the input mode and connect to V_{CC} or V_{SS} through a resistor of 1 kΩ to 10 kΩ. • Set to the output mode and open at "L" or "H."
$\overline{ONW}$	Connect to V_{CC} through a resistor of 1 kΩ to 10 kΩ.
$\overline{RESET_{OUT}}$	Open
ϕ	Open
SYNC	Open
X_{OUT}	Open (only when using external clock).

2.2 Timer

2.2.1 Memory map of timer

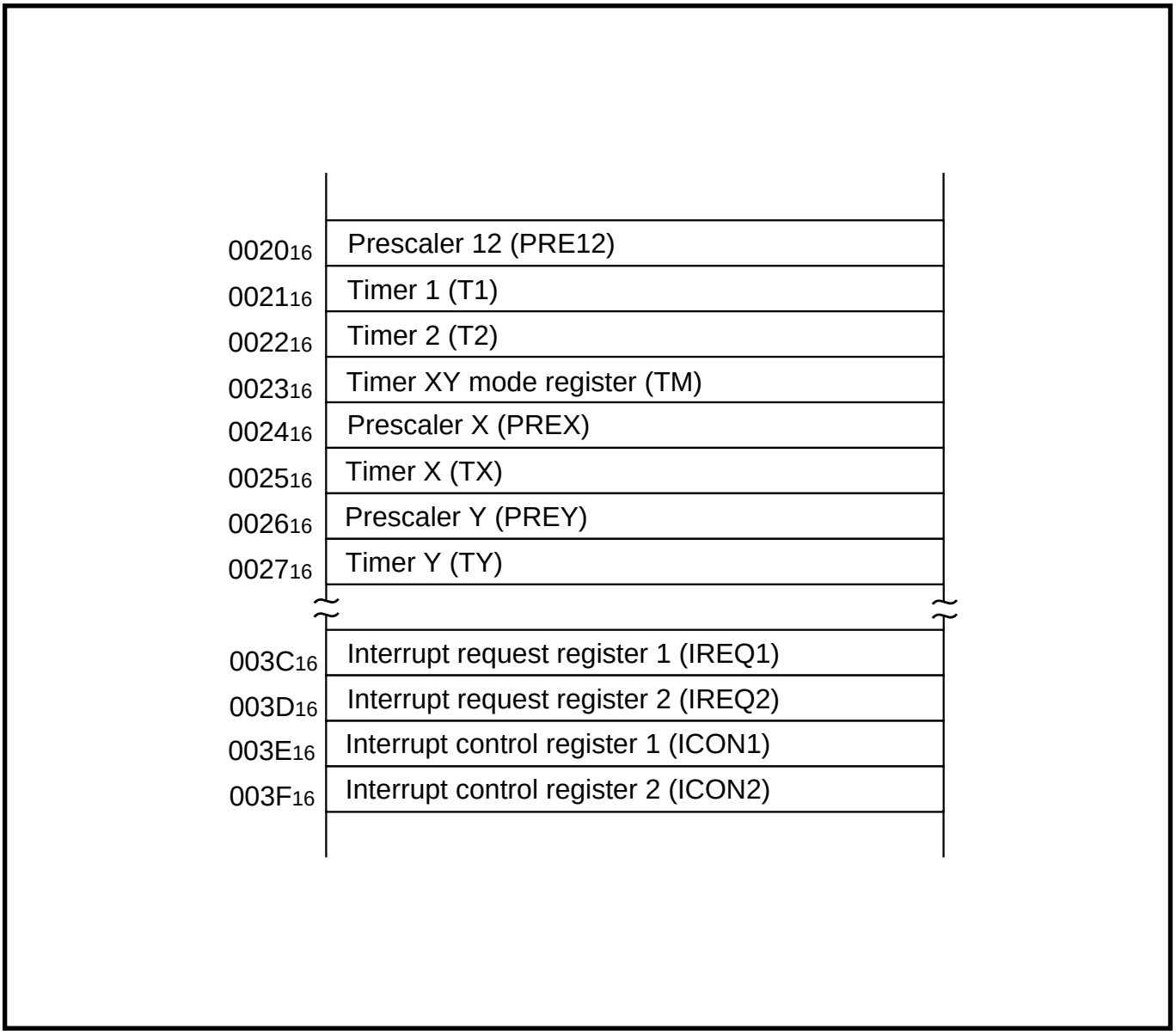


Fig. 2.2.1 Memory map of timer related registers

2.2.2 Related registers

Prescaler 12, Prescaler X, Prescaler Y

b7 b6 b5 b4 b3 b2 b1 b0

Prescaler 12 (PRE12), Prescaler X (PREX), Prescaler Y (PREY)
[Address : 20_{16}, 24_{16}, 26_{16}]

B	Function	At reset	R	W
0	● The count value of each prescaler is set. ● The value set in this register is written to both the prescaler and the prescaler latch at the same time. ● When the prescaler is read out, the value (count value) of the prescaler is read out.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

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

Timer 1

b7 b6 b5 b4 b3 b2 b1 b0

Timer 1 (T1) [Address : 21_{16}]

B	Function	At reset	R	W
0	● The count value of the Timer 1 is set. ● The value set in this register is written to both the Timer 1 and the Timer 1 latch at the same time. ● When the Timer 1 is read out, the value (count value) of the Timer 1 is read out.	1	○	○
1		0	○	○
2		0	○	○
3		0	○	○
4		0	○	○
5		0	○	○
6		0	○	○
7		0	○	○

Fig. 2.2.3 Structure of Timer 1

Timer 2, Timer X, Timer Y

b7	b6	b5	b4	b3	b2	b1	b0

Timer 2 (T2), Timer X (TX), Timer Y (TY)
[Address : 22_{16}, 25_{16}, 27_{16}]

B	Function	At reset	R	W
0	● The count value of each timer is set. ● The value set in this register is written to both the Timer and the Timer latch at the same time. ● When the Timer is read out, the value (count value) of the Timer is read out.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

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

2.2 Timer

Timer XY mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer XY mode register (TM) [Address : 23_{16}]

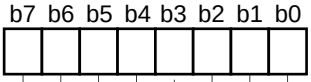
B	Name	Function	At reset	R	W
0	Timer X operating mode bit	b1 b0 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
1			0	○	○
2	$CNTR_0$ active edge switch bit	It depends on the operating mode of the Timer X (refer to Table 2.2.1).	0	○	○
3	Timer X count stop bit	0 : Count start 1 : Count stop	0	○	○
4	Timer Y operating mode bit	b5 b4 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
5			0	○	○
6	$CNTR_1$ active edge switch bit	It depends on the operating mode of the Timer Y (refer to Table 2.2.1).	0	○	○
7	Timer Y count stop bit	0 : Count start 1 : Count stop	0	○	○

Fig. 2.2.5 Structure of Timer XY mode register

Table. 2.2.1 Function of $CNTR_0$/$CNTR_1$ edge switch bit

Operating mode of Timer X/Timer Y	Function of $CNTR_0$/$CNTR_1$ edge switch bit (bits 2 and 6)	
Timer mode	"0"	• Generation of $CNTR_0$/$CNTR_1$ interrupt request : Falling edge (No effect on timer count)
	"1"	• Generation of $CNTR_0$/$CNTR_1$ interrupt request : Rising edge (No effect on timer count)
Pulse output mode	"0"	• Start of pulse output : From "H" level • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Falling edge
	"1"	• Start of pulse output : From "L" level • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Rising edge
Event counter mode	"0"	• Timer X/Timer Y : Count of rising edge • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Falling edge
	"1"	• Timer X/Timer Y : Count of falling edge • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Rising edge
Pulse width measurement mode	"0"	• Timer X/Timer Y : Measurement of "H" level width • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Falling edge
	"1"	• Timer X/Timer Y : Measurement of "L" level width • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Rising edge

Interrupt request register 1



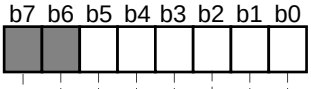
Interrupt request reigster 1 (IREQ1) [Address : $3C_{16}$]

B	Name	Function	At reset	R	W
0	INT_0 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
1	INT_1 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
2	Serial I/O receive interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
3	Serial I/O transmit interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
4	Timer X interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
5	Timer Y interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
6	Timer 1 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
7	Timer 2 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*

* "0" is set by software, but not "1."

Fig. 2.2.6 Structure of Interrupt request register 1

Interrupt request register 2



Interrupt request reigster 2 (IREQ2) [Address : $3D_{16}$]

B	Name	Function	At reset	R	W
0	$CNTR_0$ interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
1	$CNTR_1$ interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
2	INT_2 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
3	INT_3 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
4	INT_4 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
5	INT_5 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
6	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0."		0	○	×
7			0	○	×

* "0" is set by software, but not "1."

Fig. 2.2.7 Structure of Interrupt request register 2

Interrupt control register 1

b7	b6	b5	b4	b3	b2	b1	b0

Interrupt control register 1 (ICON1) [Address : $3E_{16}$]

B	Name	Function	At reset	R	W
0	INT_0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT_1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.2.8 Structure of Interrupt control register 1

Interrupt control register 2

b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Interrupt control reigster 2 (ICON2) [Address : $3F_{16}$]

B	Name	Function	At reset	R	W
0	$CNTR_0$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	$CNTR_1$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT_2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	INT_5 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Fix these bits to “0.”		0	○	○
7			0	○	○

Fig. 2.2.9 Structure of Interrupt control register 2

2.2.3 Timer application examples

(1) Basic functions and uses

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

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

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

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

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

[Use]
- Generation of cyclic interrupts
- Clock function (measurement of 250m second) → Application example 1
- Control of a main routine cycle

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

The output level of the CNTR pin is inverted every time a timer underflows (Pulse output mode).

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

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

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

[Use]
- Measurement of frequency → Application example 3
- Division of external pulses.
- Generation of interrupts in a cycle based on an external pulse. (count of a reel pulse)

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

The "H" or "L" level width of external pulses input to CNTR pin is measured (Pulse width measurement mode).

[Use]
- Measurement of external pulse frequency (Measurement of pulse width of FG pulse* generated by motor) → Application example 4
- Measurement of external pulse duty (when the frequency is fixed)

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

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

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

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

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

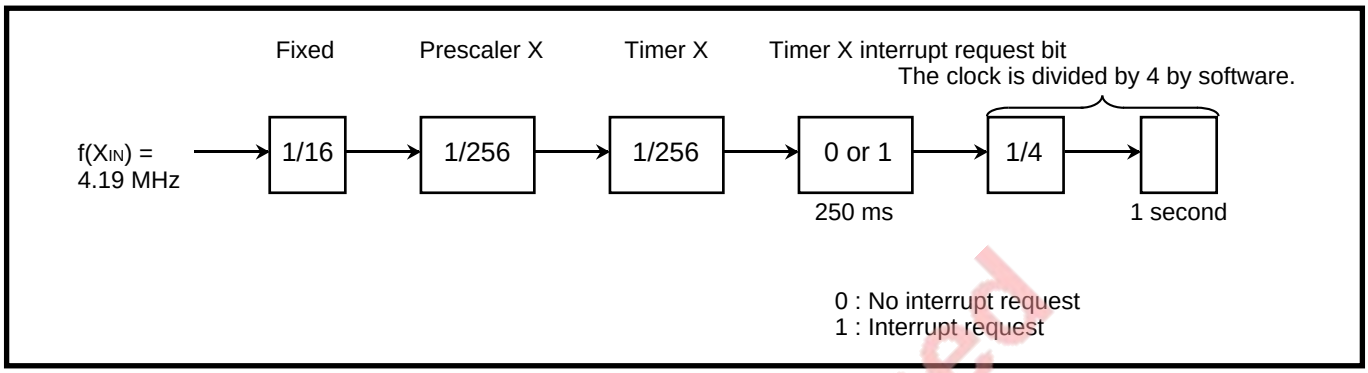


Fig. 2.2.10 Connection of timers and setting of division ratios [Clock function]

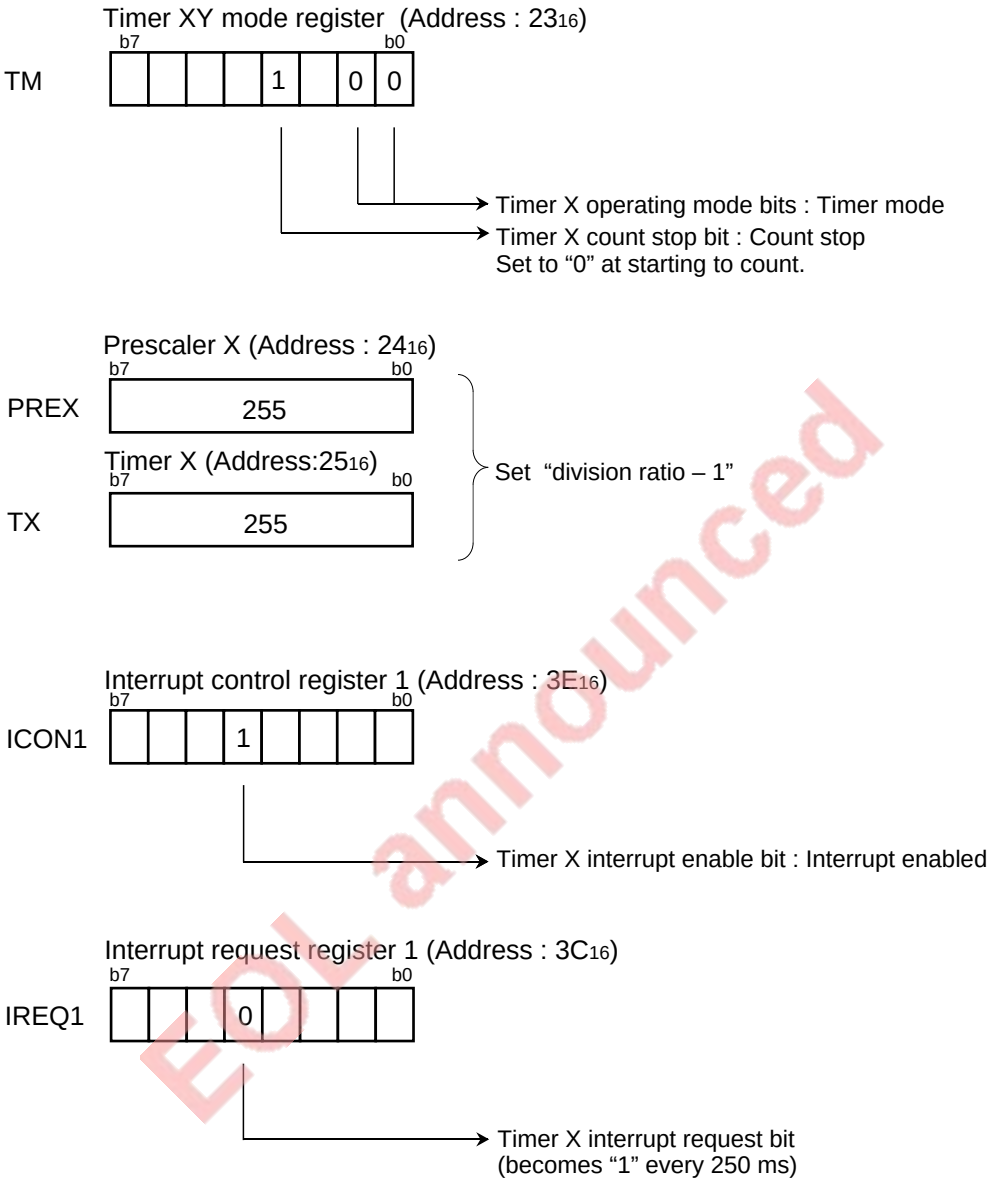


Fig. 2.2.11 Setting of related registers [Clock function]

2.2 Timer

Control procedure :

Figure 2.2.12 shows a control procedure.

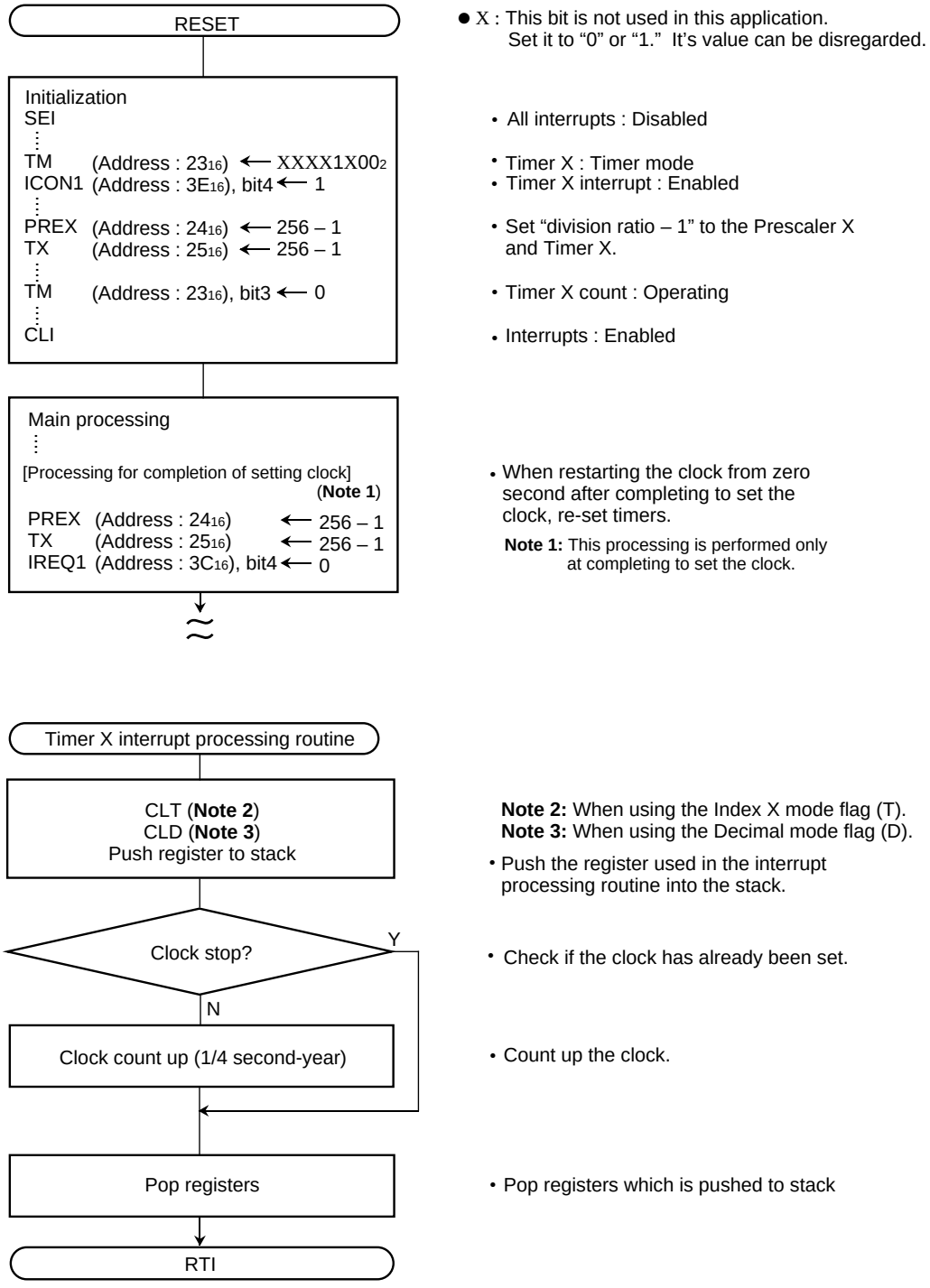


Fig. 2.2.12 Control procedure [Clock function]

(3) Timer application example 2 : Piezoelectric buzzer output

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

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

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

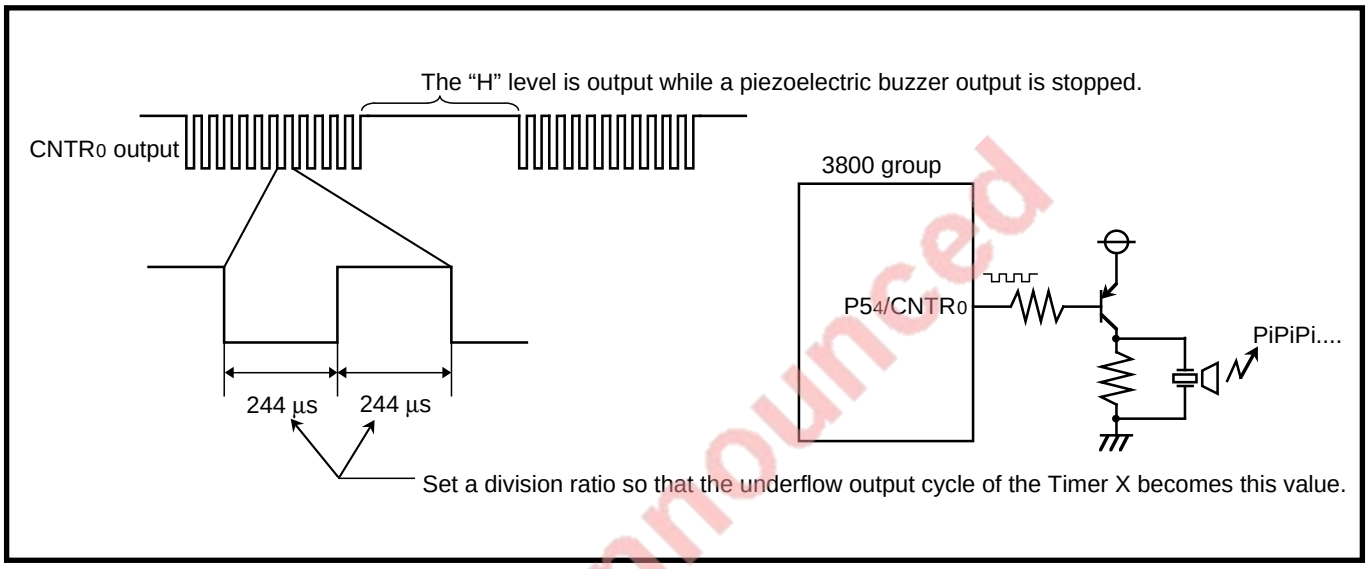


Fig. 2.2.13 Example of a peripheral circuit

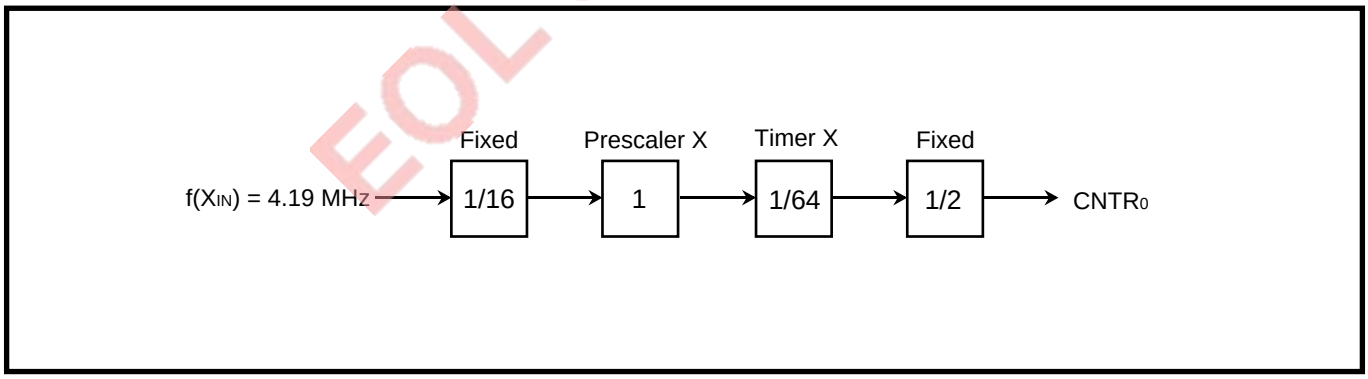


Fig. 2.2.14 Connection of the timer and setting of the division ratio [Piezoelectric buzzer output]

2.2 Timer

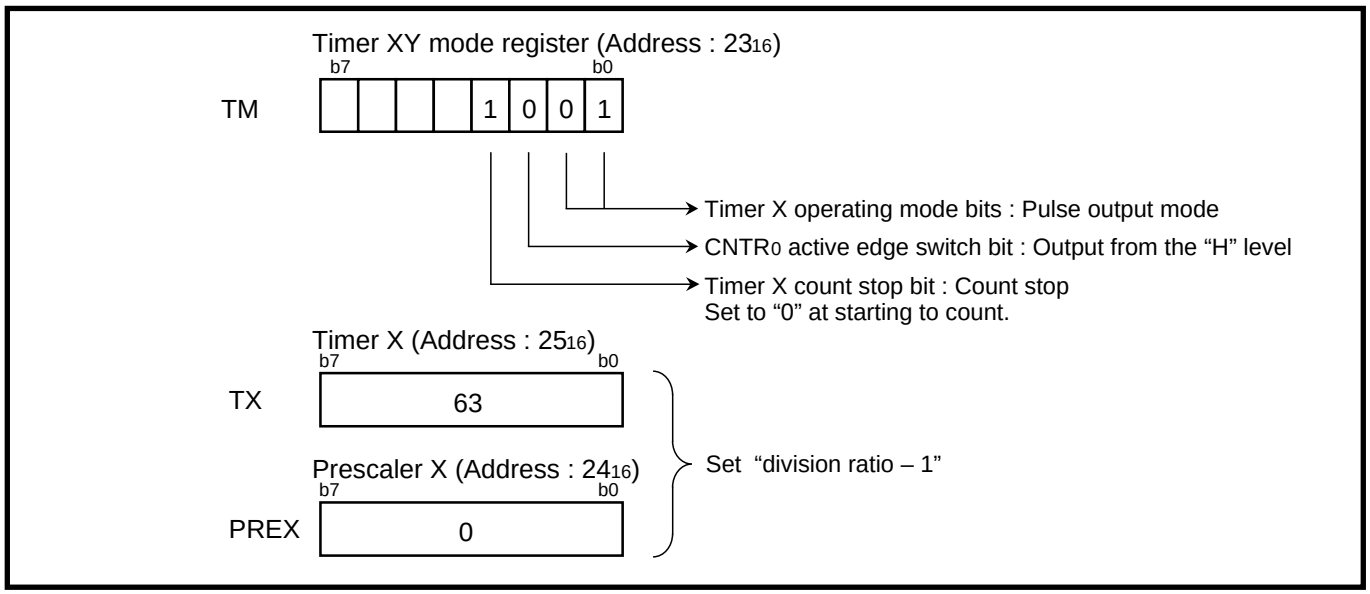


Fig. 2.2.15 Setting of related registers [Piezoelectric buzzer output]

Control procedure :

Figure 2.2.16 shows a control procedure.

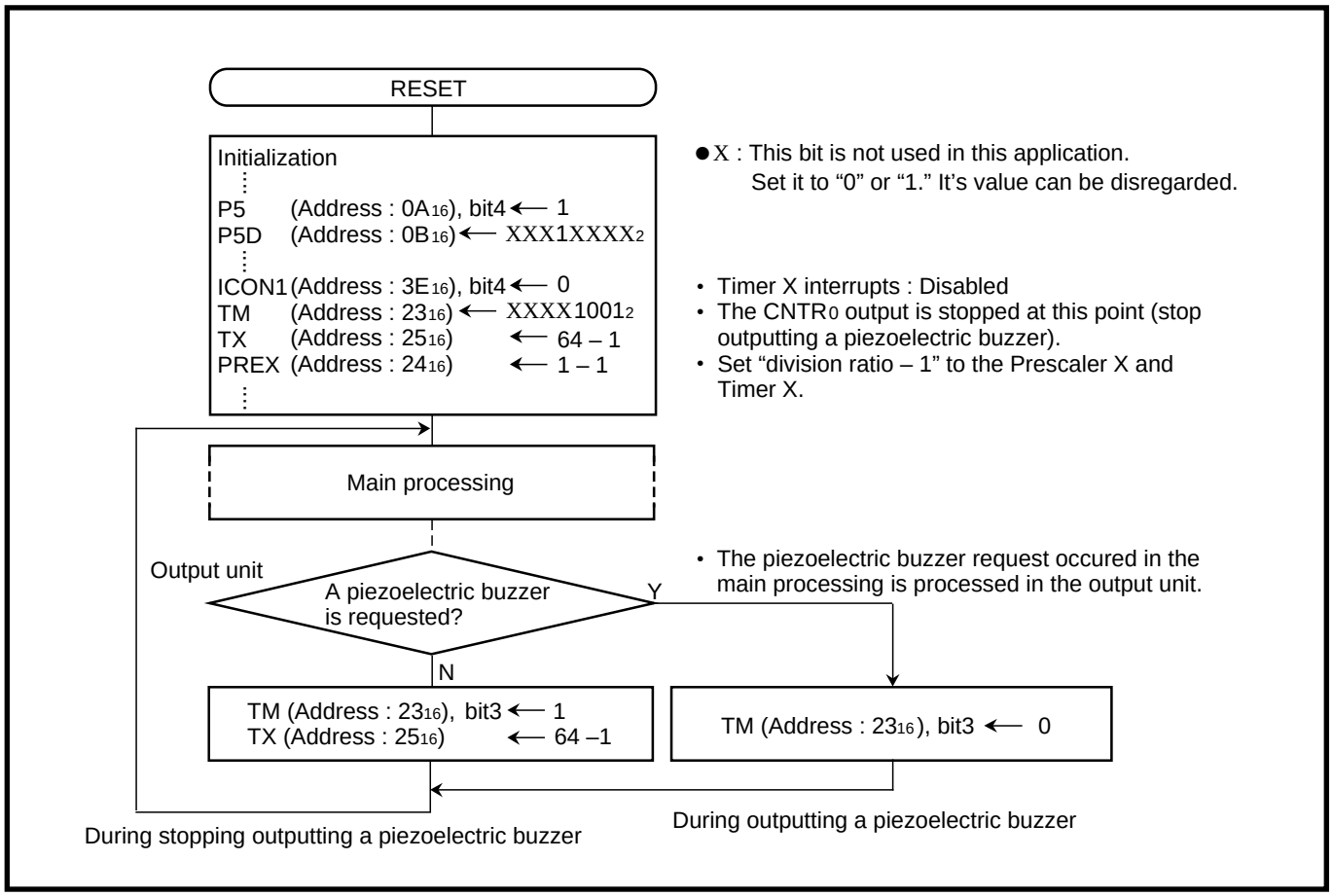


Fig. 2.2.16 Control procedure [Piezoelectric buzzer output]

(4) Timer application example 3 : Measurement of frequency

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

- A value counted a pulse which is input to P5$_5$/CNTR$_1$ pin by a timer.
- A referance value

Specifications :

- The pulse is input to the P5$_5$/CNTR$_1$ pin and counted by the Timer Y.
- A count value is read out at the interval of about 2 ms (Timer 1 interrupt interval : 244 μs × 8). When the count value is 28 to 40, it is regarded the input pulse as a valid.
- Because the timer is a down-counter, the count value is compared with 227 to 215* .

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

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

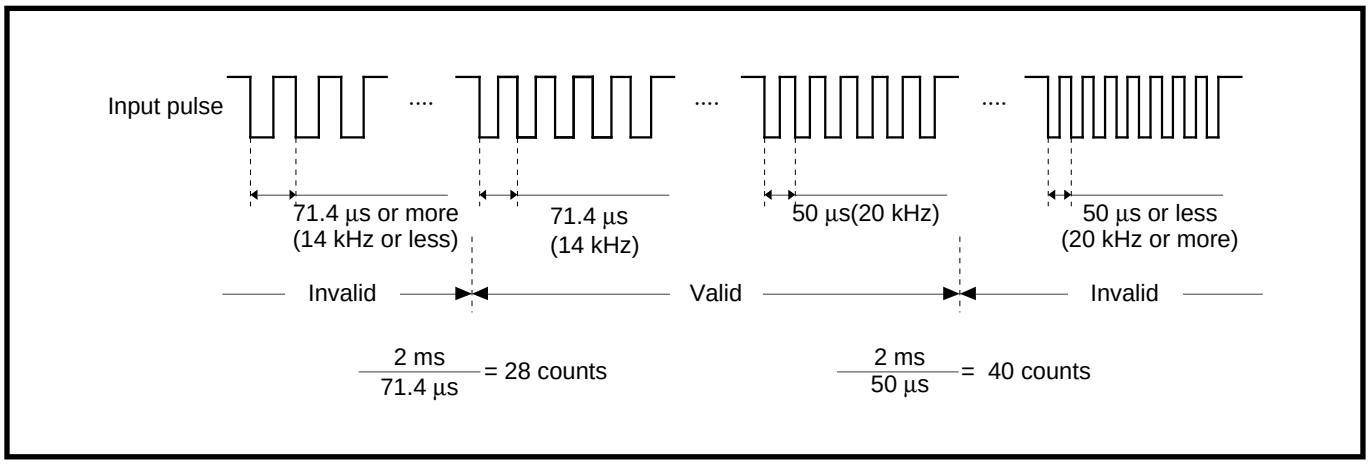


Fig 2.2.17 A method for judging if input pulse exists

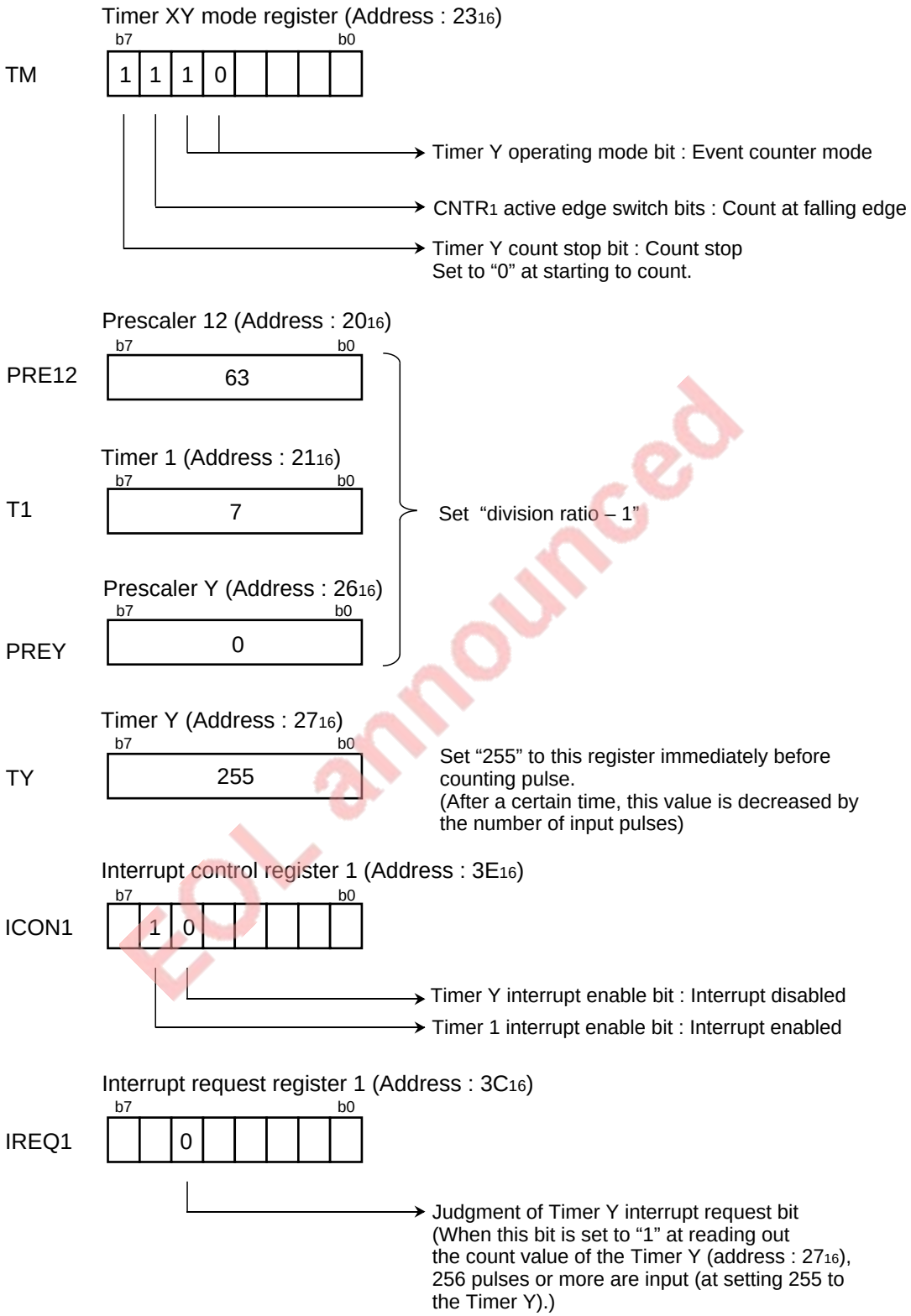


Fig. 2.2.18 Setting of related registers [Measurement of frequency]

Control procedure :

Figure 2.2.19 shows a control procedure.

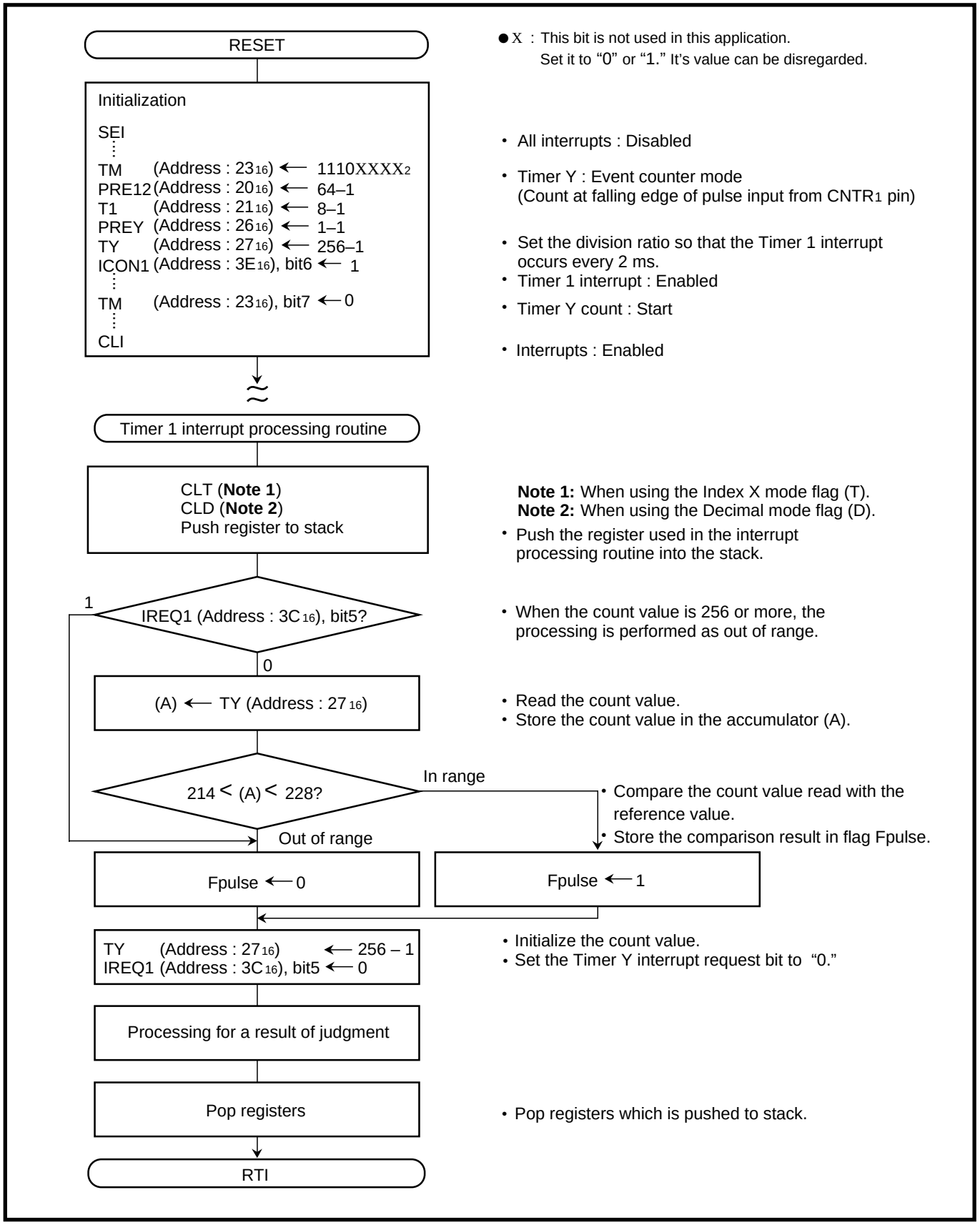


Fig. 2.2.19 Control procedure [Measurement of frequency]

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

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

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

(Example : When the clock frequency is 4.19 MHz, the count source would be 3.8 μs that is obtained by dividing the clock frequency by 16. Measurement can be made up to 250 ms in the range of FFFF16 to 000016.)

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

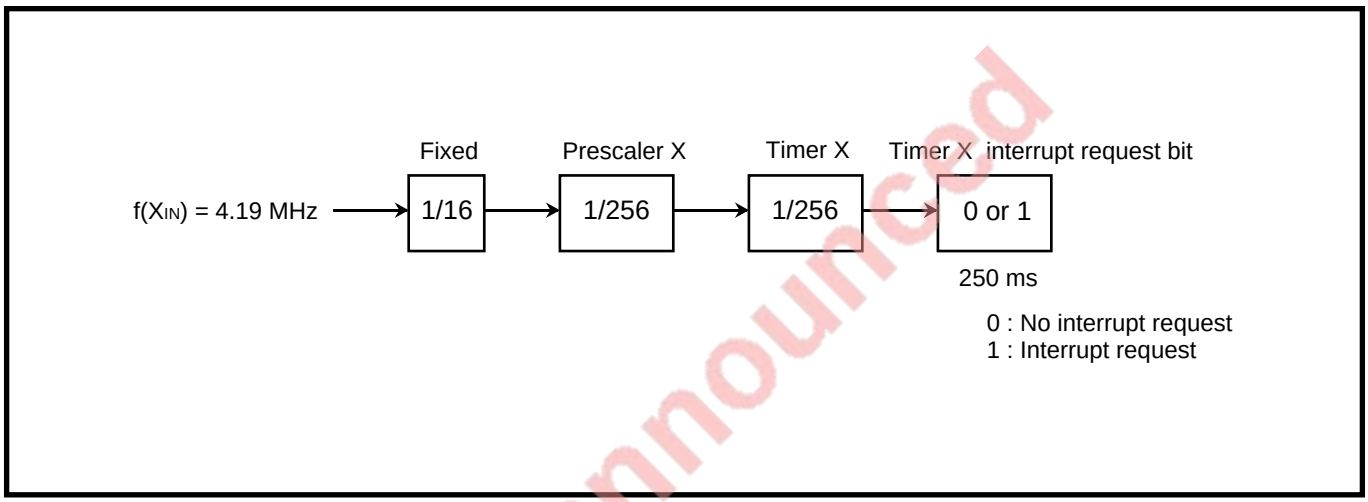


Fig. 2.2.20 Connection of the timer and setting of the division ratio [Measurement of pulse width]

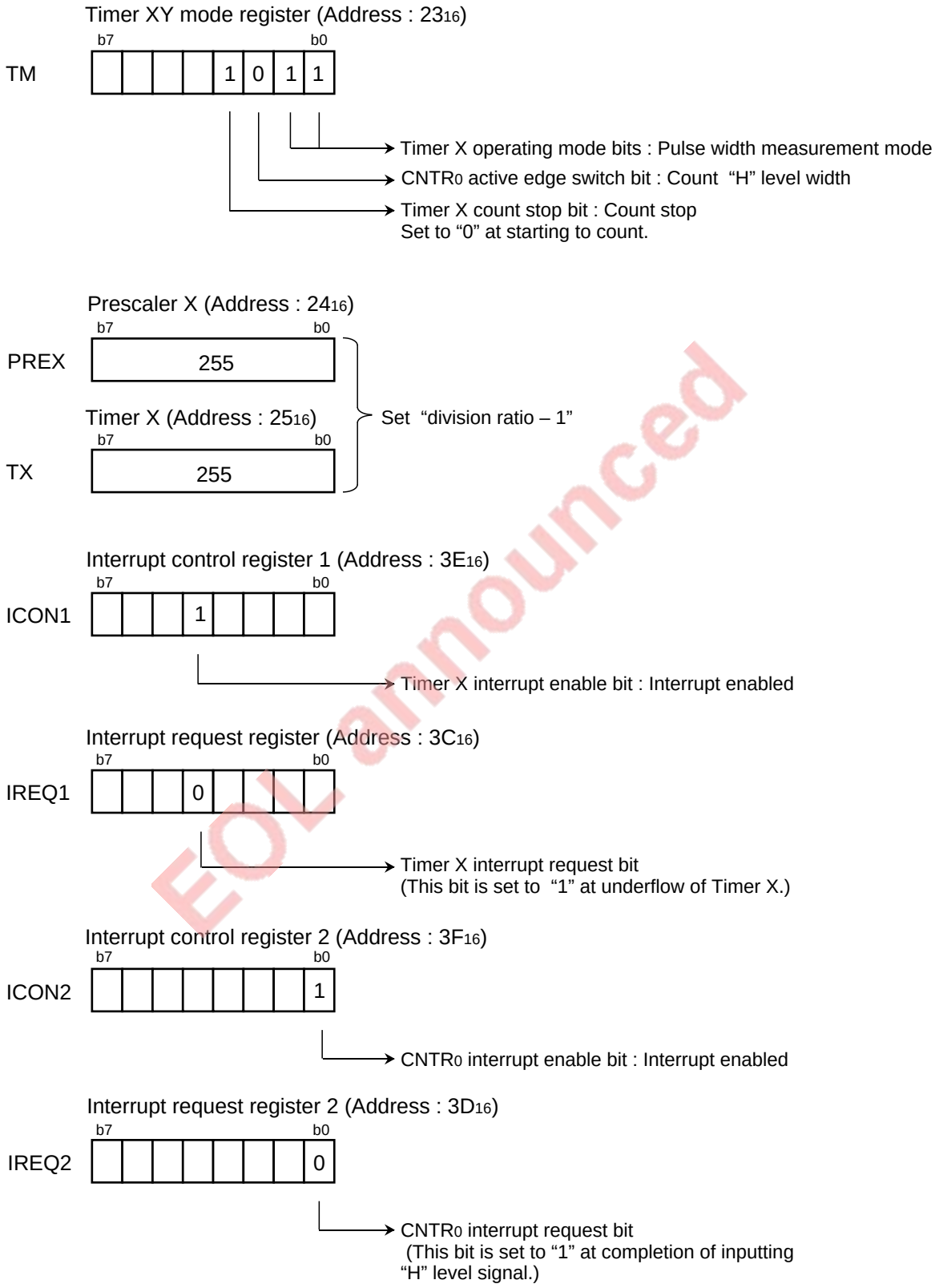


Fig. 2.2.21 Setting of related registers [Measurement of pulse width]

2.2 Timer

Figure 2.2.22 shows a control procedure.

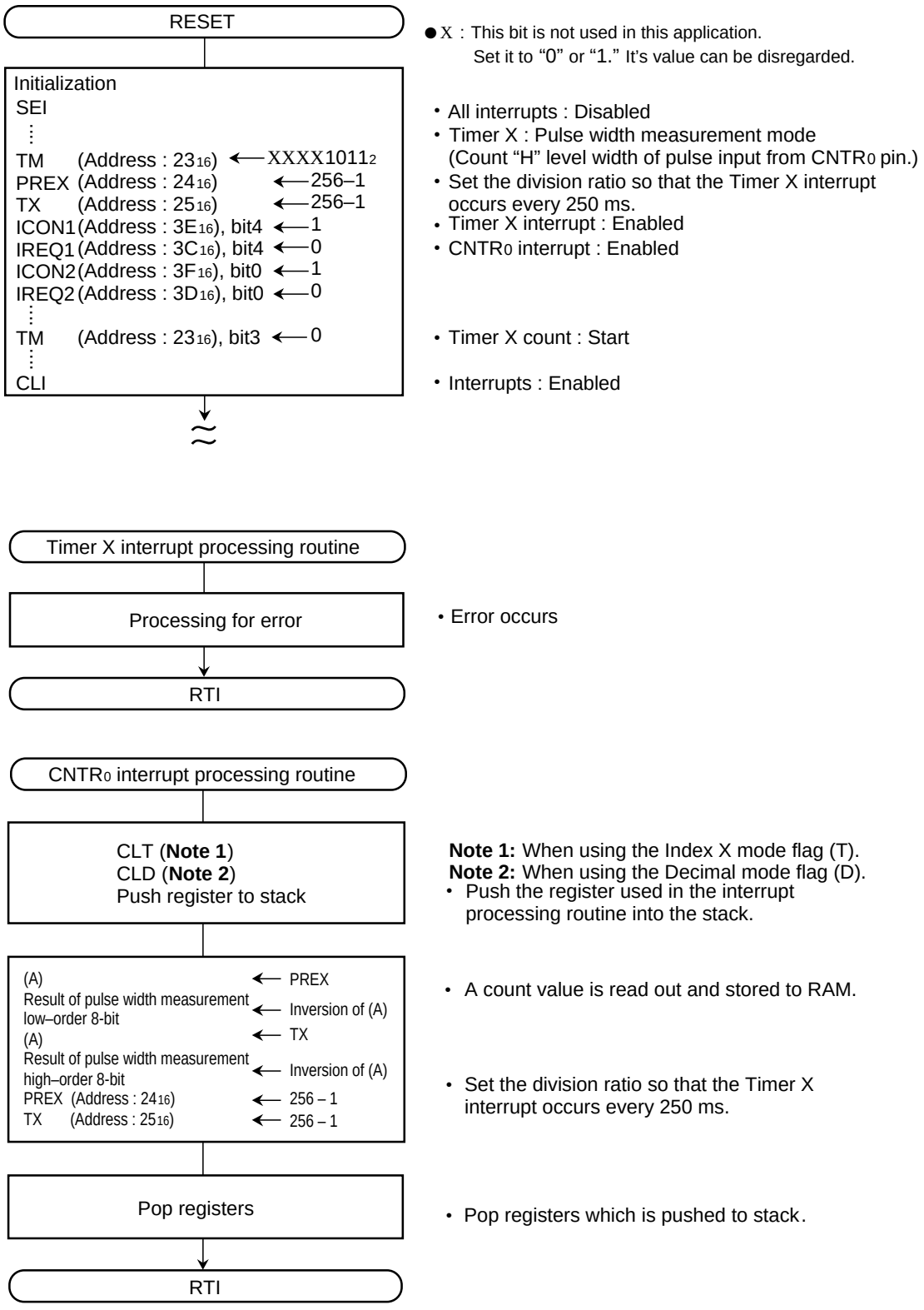


Fig. 2.2.22 Control procedure [Measurement of pulse width]

2.3 Serial I/O

2.3.1 Memory map of serial I/O

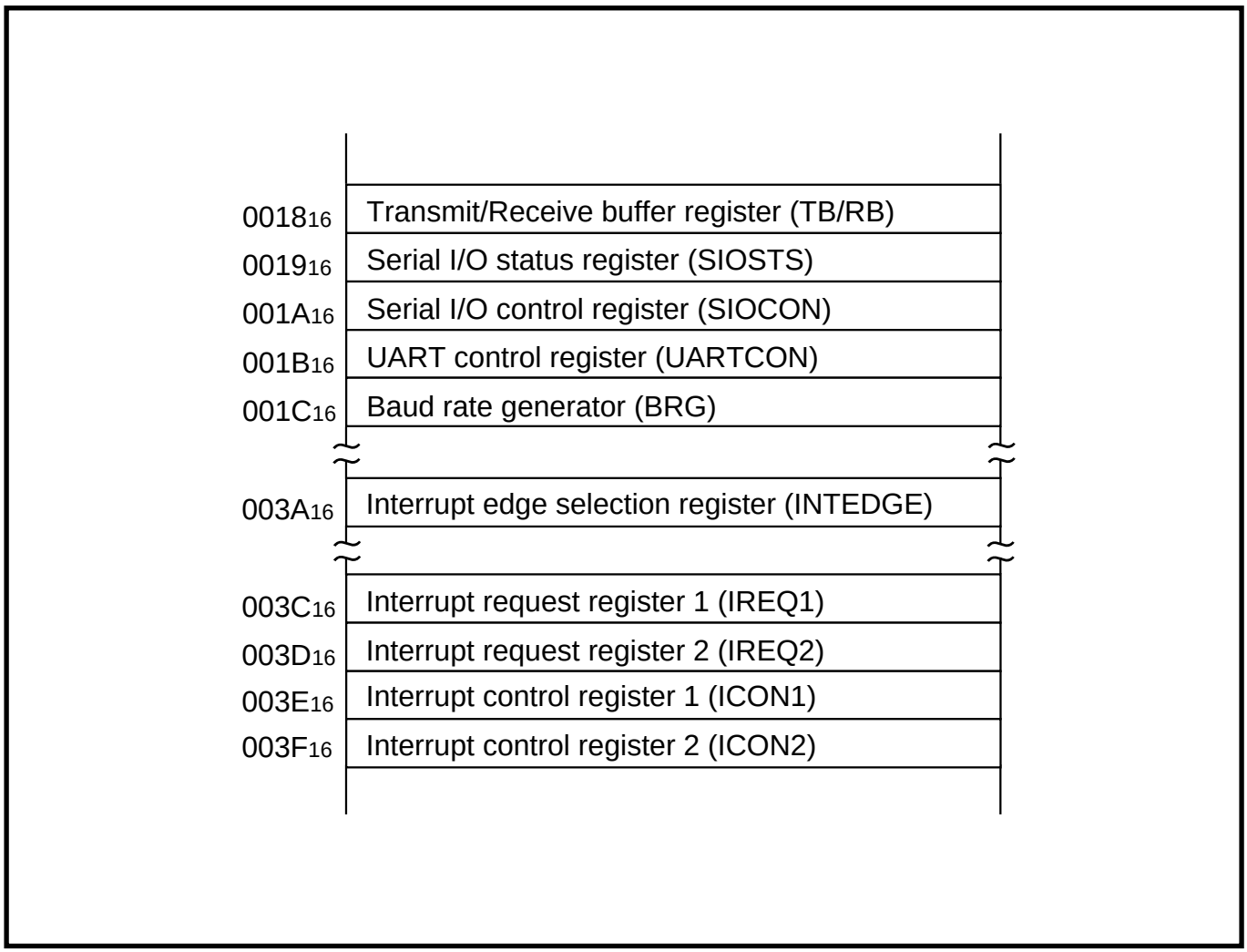


Fig. 2.3.1 Memory map of serial I/O related registers

2.3 Serial I/O

2.3.2 Related registers

Transmit/Receive buffer register

b7 b6 b5 b4 b3 b2 b1 b0

Transmit/Receive buffer register (TB/RB) [Address : 18_{16}]

B	Function	At reset	R	W
0	A transmission data is written to or a receive data is read out from this buffer register. • At writing : a data is written to the Transmit buffer register. • At reading : a content of the Receive buffer register is read out.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Note : A content of the Transmit buffer register cannot be read out.
A data cannot be written to the Receive buffer register.

Fig. 2.3.2 Structure of Transmit/Receive buffer register

Serial I/O status register

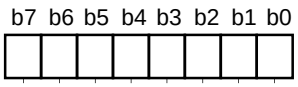
b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O status reigster (SIOSTS) [Address : 19_{16}]

B	Name	Function	At reset	R	W
0	Transmit buffer empty flag (TBE)	0 : Buffer full 1 : Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0 : Buffer empty 1 : Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0 : Transmit shift in progress 1 : Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0 : No error 1 : Overrun error	0	○	×
4	Parity error flag (PE)	0 : No error 1 : Parity error	0	○	×
5	Framing error flag (FE)	0 : No error 1 : Framing error	0	○	×
6	Summing error flag (SE)	0 : (OE) ∪ (PE) ∪ (FE) = 0 1 : (OE) ∪ (PE) ∪ (FE) = 1	0	○	×
7	Nothing is allocated for this bit. It is a write disabled bit. When this bit is read out, the value is “0.”		1	○	×

Fig. 2.3.3 Structure of Serial I/O status register

Serial I/O control register

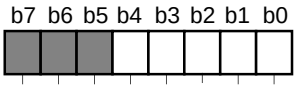


Serial I/O control register (SIOCON) [Address : $1A_{16}$]

B	Name	Function	At reset	R	W
0	BRG count source selection bit (CSS)	0 : $f(X_{IN})$ 1 : $f(X_{IN})/4$	0	○	○
1	Serial I/O 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	$\overline{S_{RDY}}$ output enable bit (SRDY)	0 : I/O port (P47) 1 : $\overline{S_{RDY}}$ 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/O mode selection bit (SIOM)	0 : UART 1 : Clock synchronous serial I/O	0	○	○
7	Serial I/O enable bit (SIOE)	0 : Serial I/O disabled ($P4_4$–$P4_7$: I/O port) 1 : Serial I/O enabled ($P4_4$–$P4_7$: Serial I/O function pin)	0	○	○

Fig. 2.3.4 Structure of Serial I/O control register

UART control register



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

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	$P4_5$/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	○	×
6			1	○	×
7			1	○	×

Fig. 2.3.5 Structure of UART control register

2.3 Serial I/O

Baud rate generator

b7 b6 b5 b4 b3 b2 b1 b0

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

B	Function	At reset	R	W
0	A count value of Baud rate generator is set.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 2.3.6 Structure of Baud rate generator

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0

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

B	Name	Function	At reset	R	W
0	INT_0 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT_1 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
2	INT_2 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
3	INT_3 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
4	INT_4 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
5	INT_5 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
6	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are “0.”		0	○	×
7			0	○	×

Fig. 2.3.7 Structure of Interrupt edge selection register

Interrupt request register 1

b7	b6	b5	b4	b3	b2	b1	b0

Interrupt request reigster 1 (IREQ1) [Address : $3C_{16}$]

B	Name	Function	At reset	R	W
0	INT_0 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
1	INT_1 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
2	Serial I/O receive interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
3	Serial I/O transmit interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
4	Timer X interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
5	Timer Y interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
6	Timer 1 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
7	Timer 2 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*

* "0" is set by software, but not "1."

Fig. 2.3.8 Structure of Interrupt request register 1

Interrupt control register 1

b7	b6	b5	b4	b3	b2	b1	b0

Interrupt control register 1 (ICON1) [Address : $3E_{16}$]

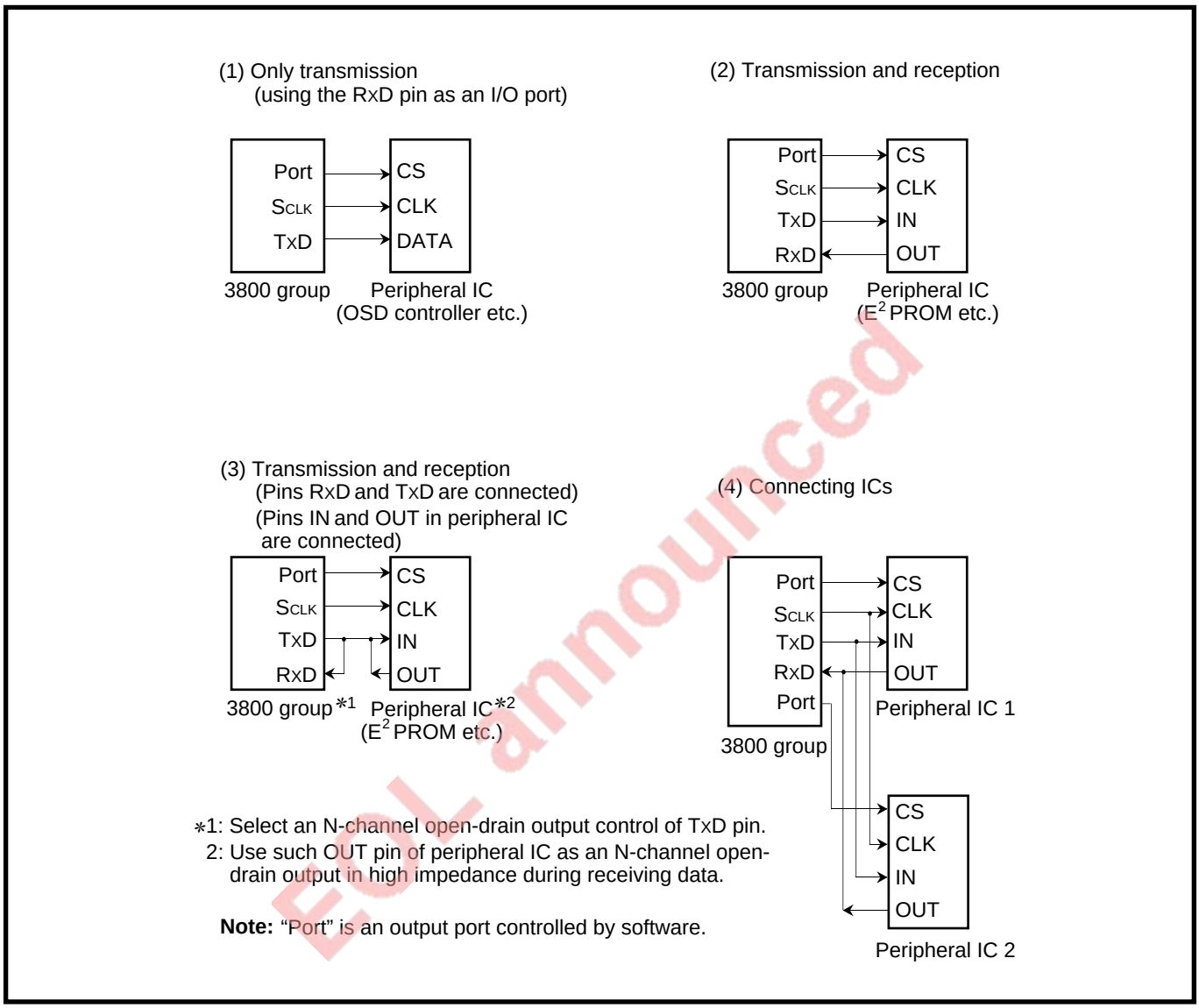
B	Name	Function	At reset	R	W
0	INT_0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT_1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.3.9 Structure of Interrupt control register 1

2.3.3 Serial I/O connection examples

(1) Control of peripheral IC equipped with CS pin

There are connection examples using a clock synchronous serial I/O mode.
Figure 2.3.10 shows connection examples of a peripheral IC equipped with the CS pin.



∗1: Select an N-channel open-drain output control of TxD pin.
2: Use such OUT pin of peripheral IC as an N-channel open-drain output in high impedance during receiving data.

Note: "Port" is an output port controlled by software.

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

(2) Connection with microcomputer

Figure 2.3.11 shows connection examples of the other microcomputers.

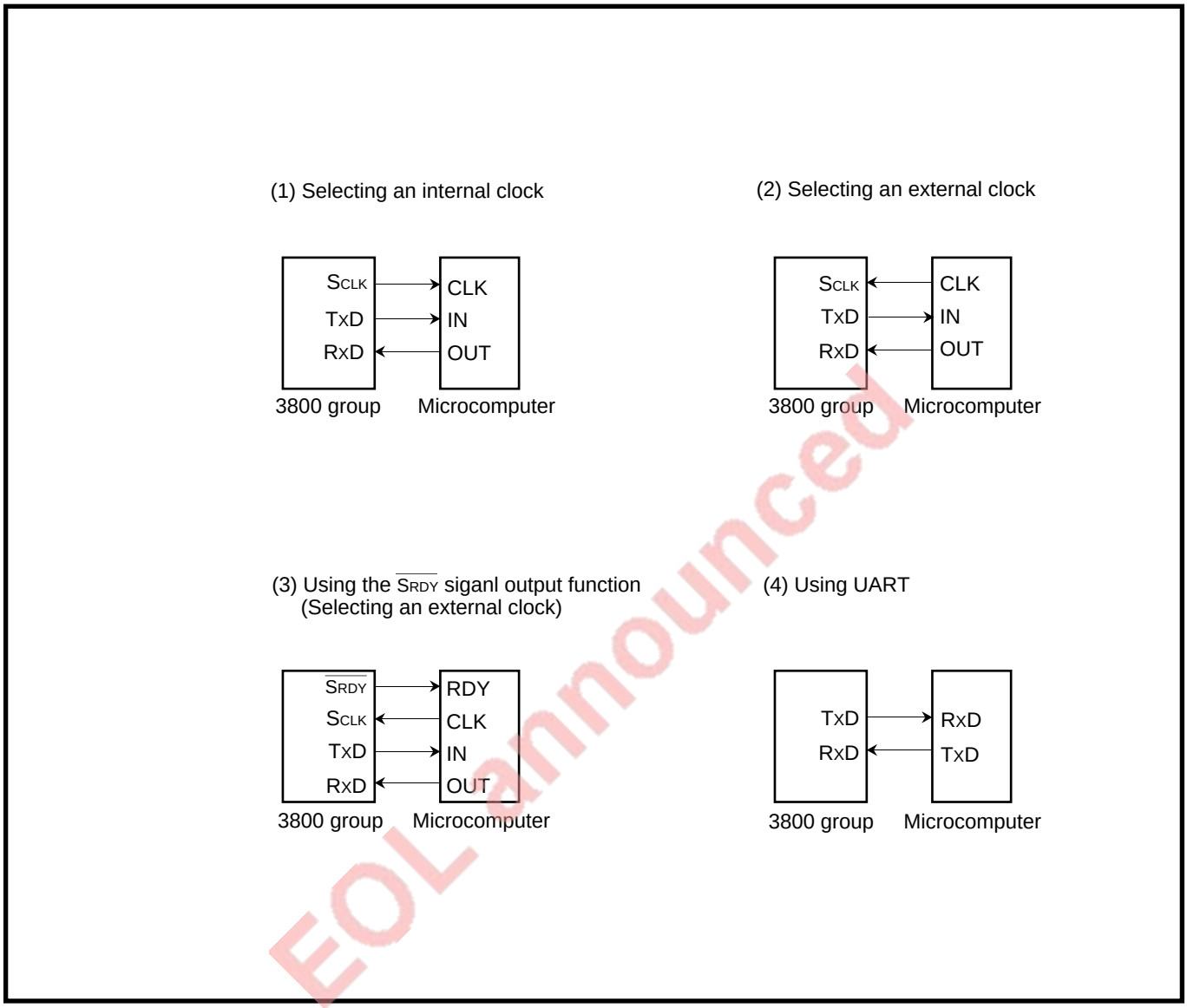


Fig. 2.3.11 Serial I/O connection examples (2)

2.3 Serial I/O

2.3.4 Setting of serial I/O transfer data format

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

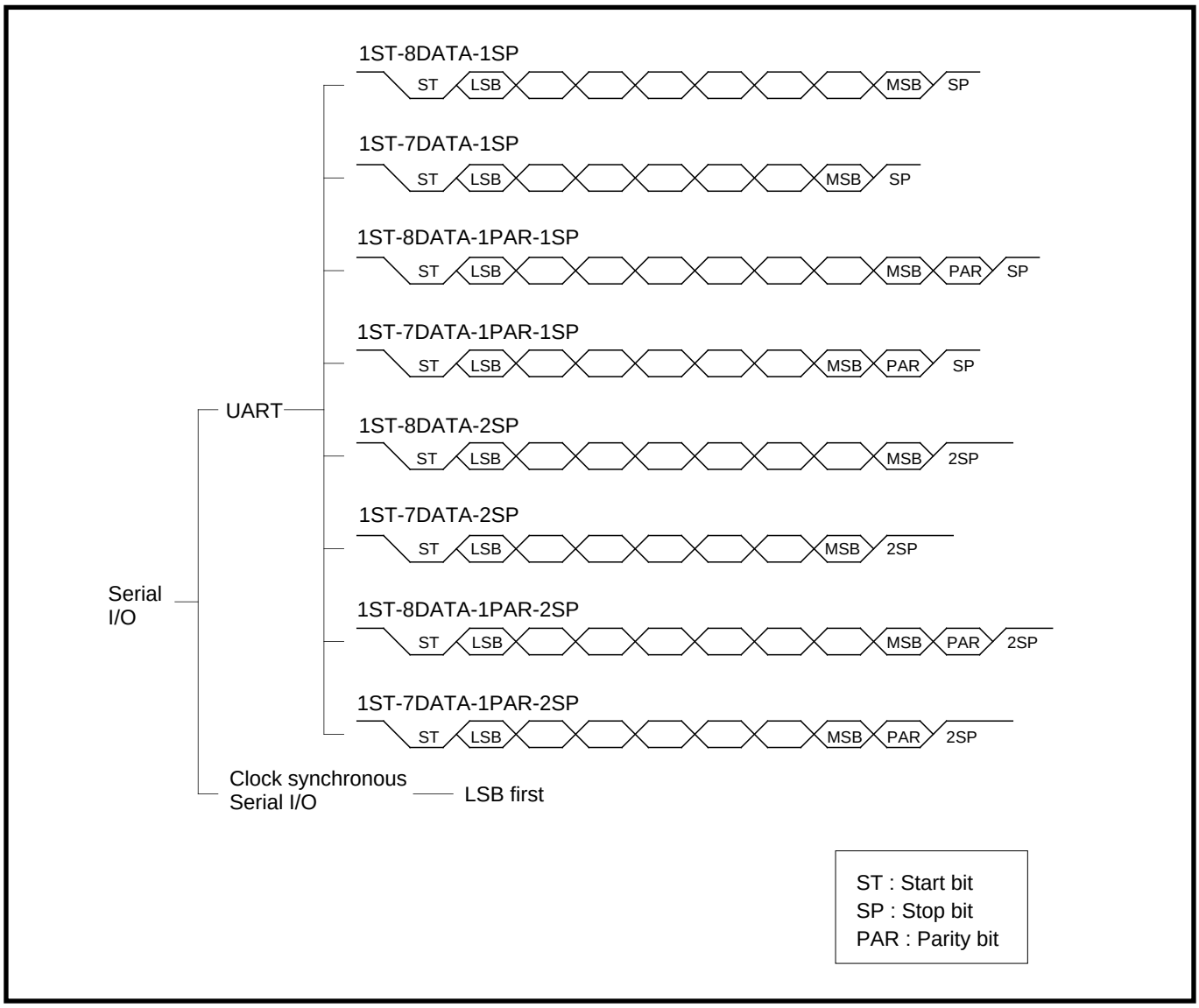


Fig. 2.3.12 Setting of Serial I/O transfer data format

2.3.5 Serial I/O application examples

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

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

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

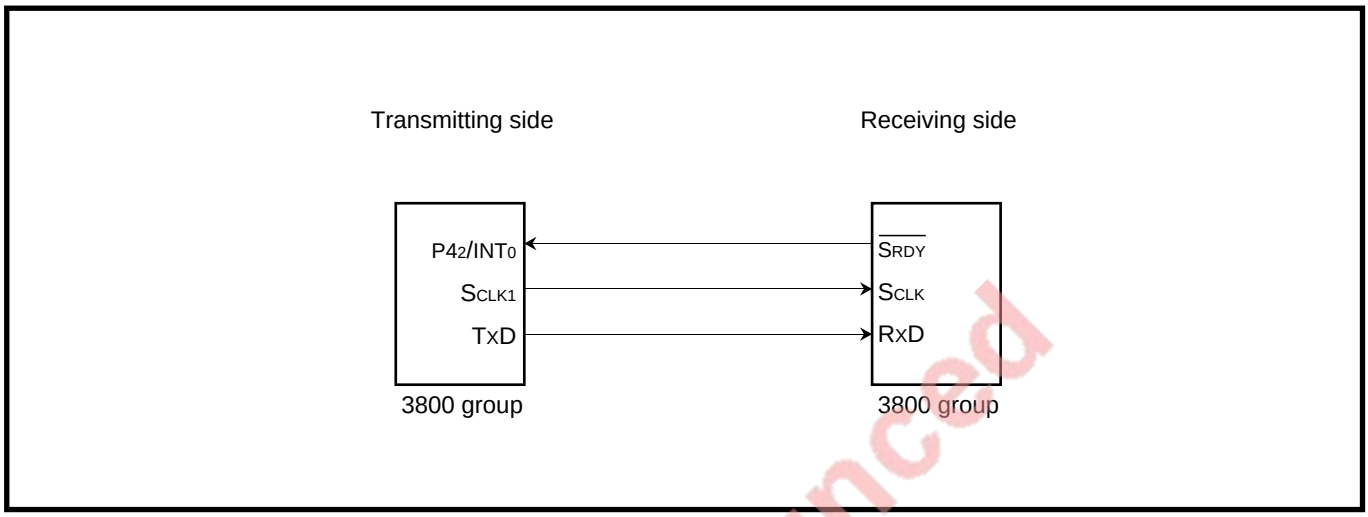


Fig. 2.3.13 Connection diagram [Communication using a clock synchronous serial I/O]

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

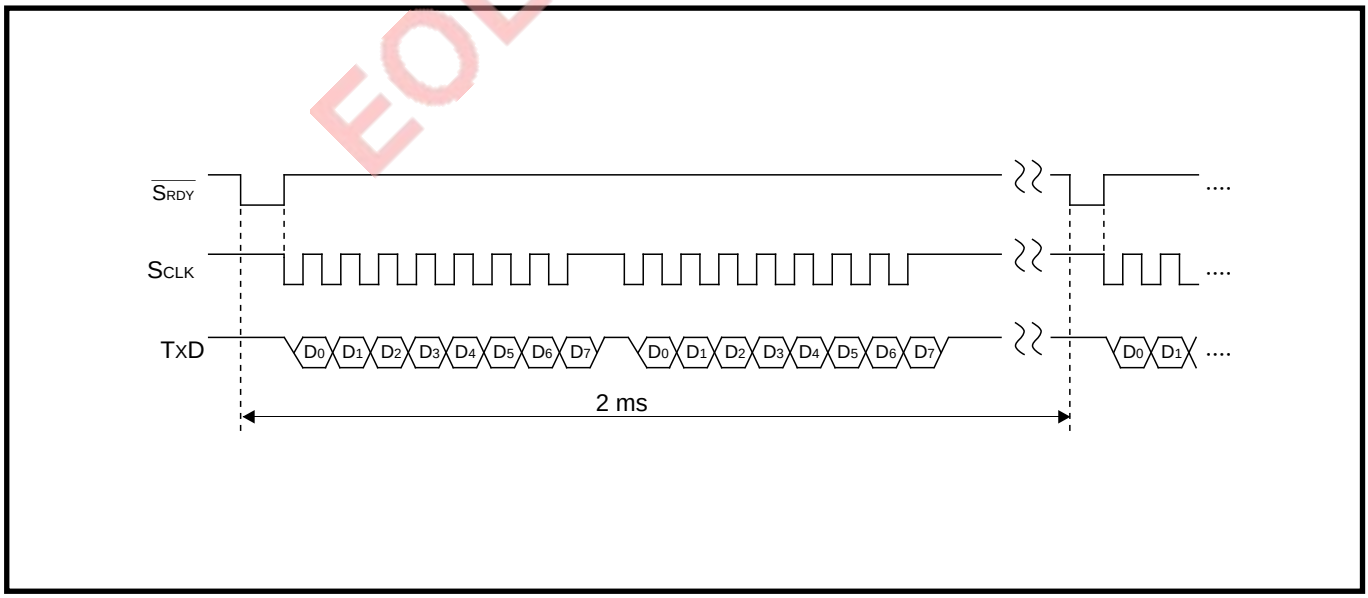


Fig. 2.3.14 Timing chart [Communication using a clock synchronous serial I/O]

2.3 Serial I/O

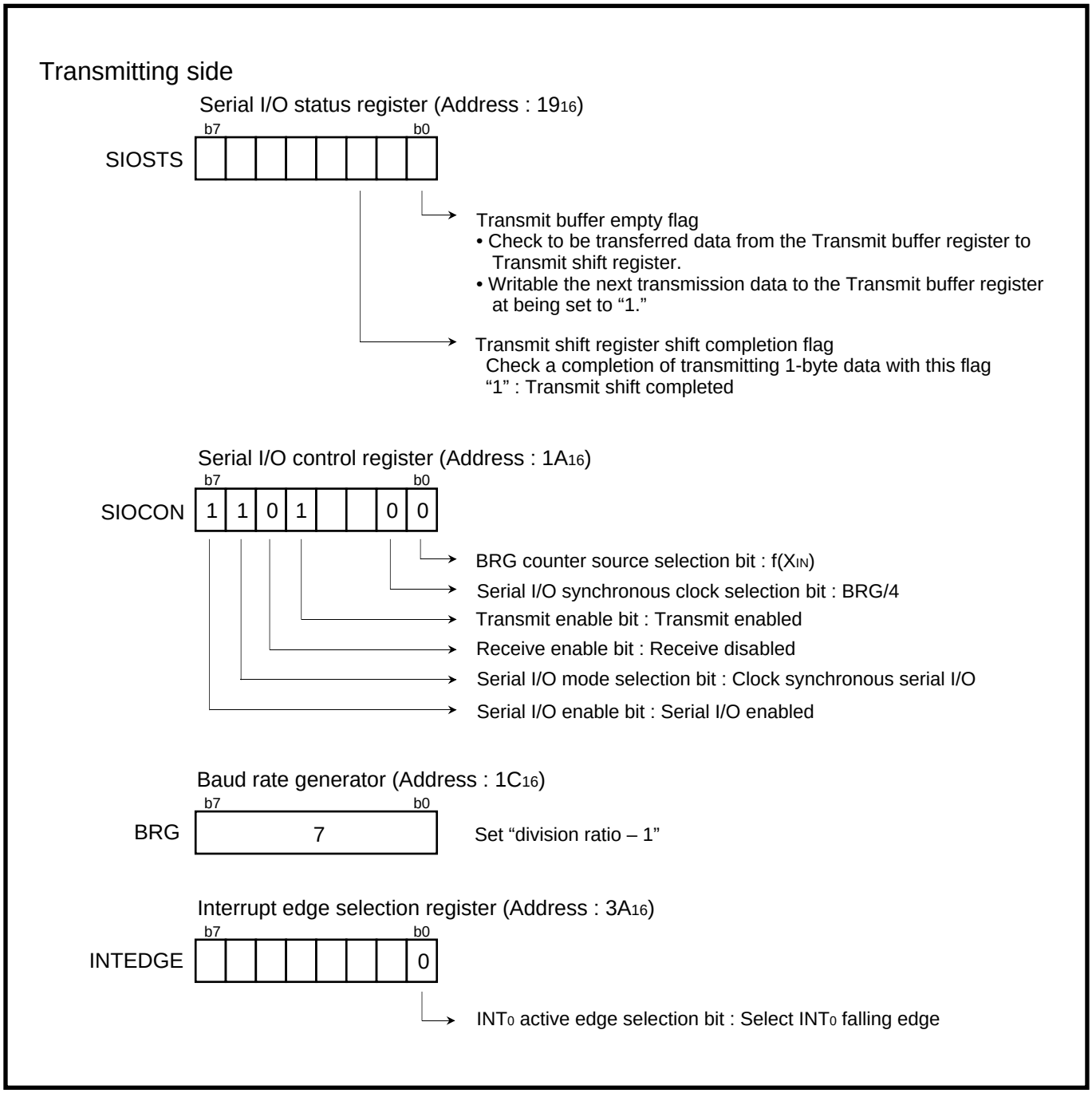


Fig. 2.3.15 Setting of related registers at a transmitting side [Communication using a clock synchronous serial I/O]

Receiving side

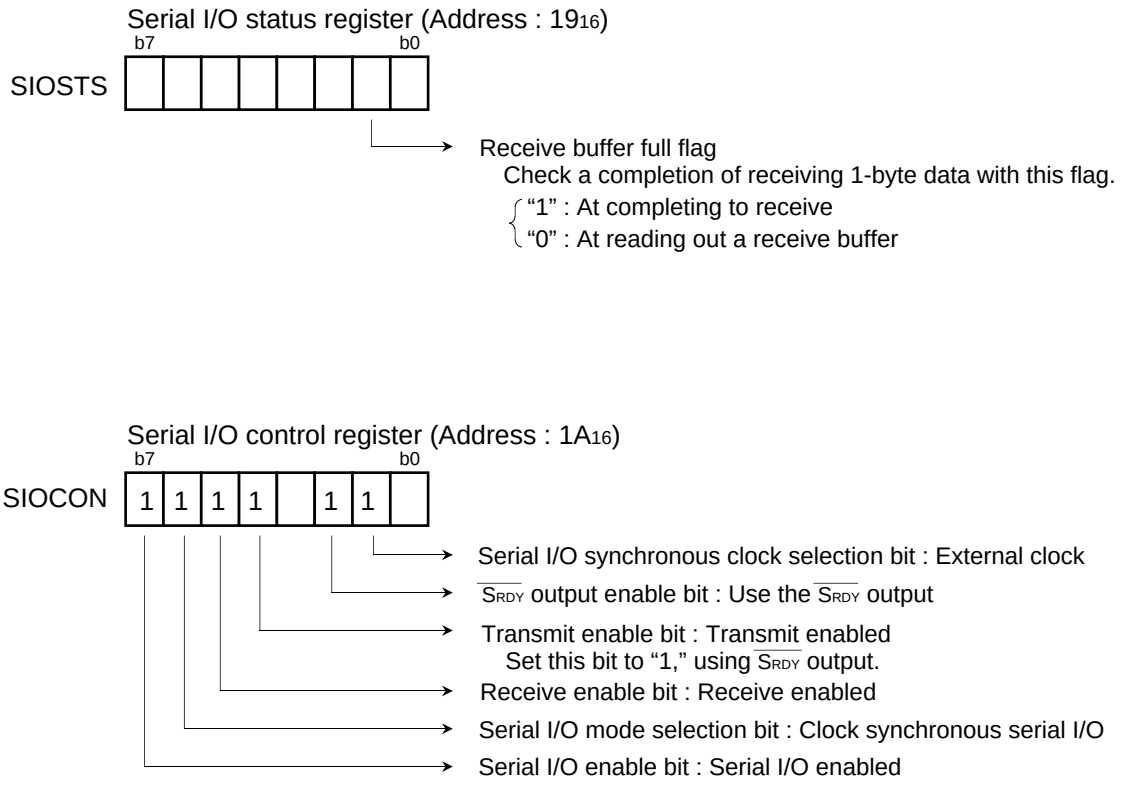


Fig. 2.3.16 Setting of related registers at a receiving side [Communication using a clock synchronous serial I/O]

2.3 Serial I/O

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

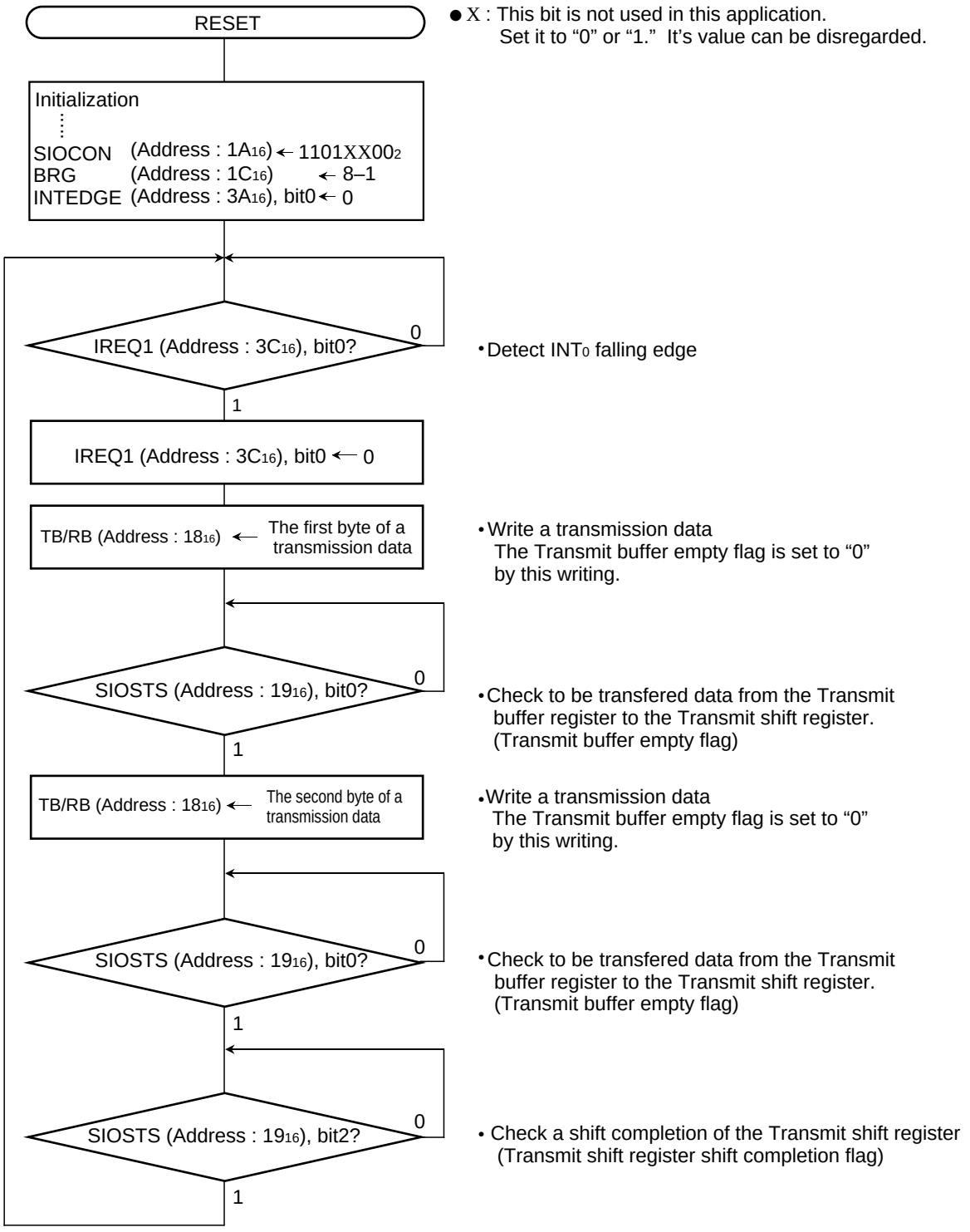


Fig. 2.3.17 Control procedure at a transmitting side [Communication using a clock synchronous serial I/O]

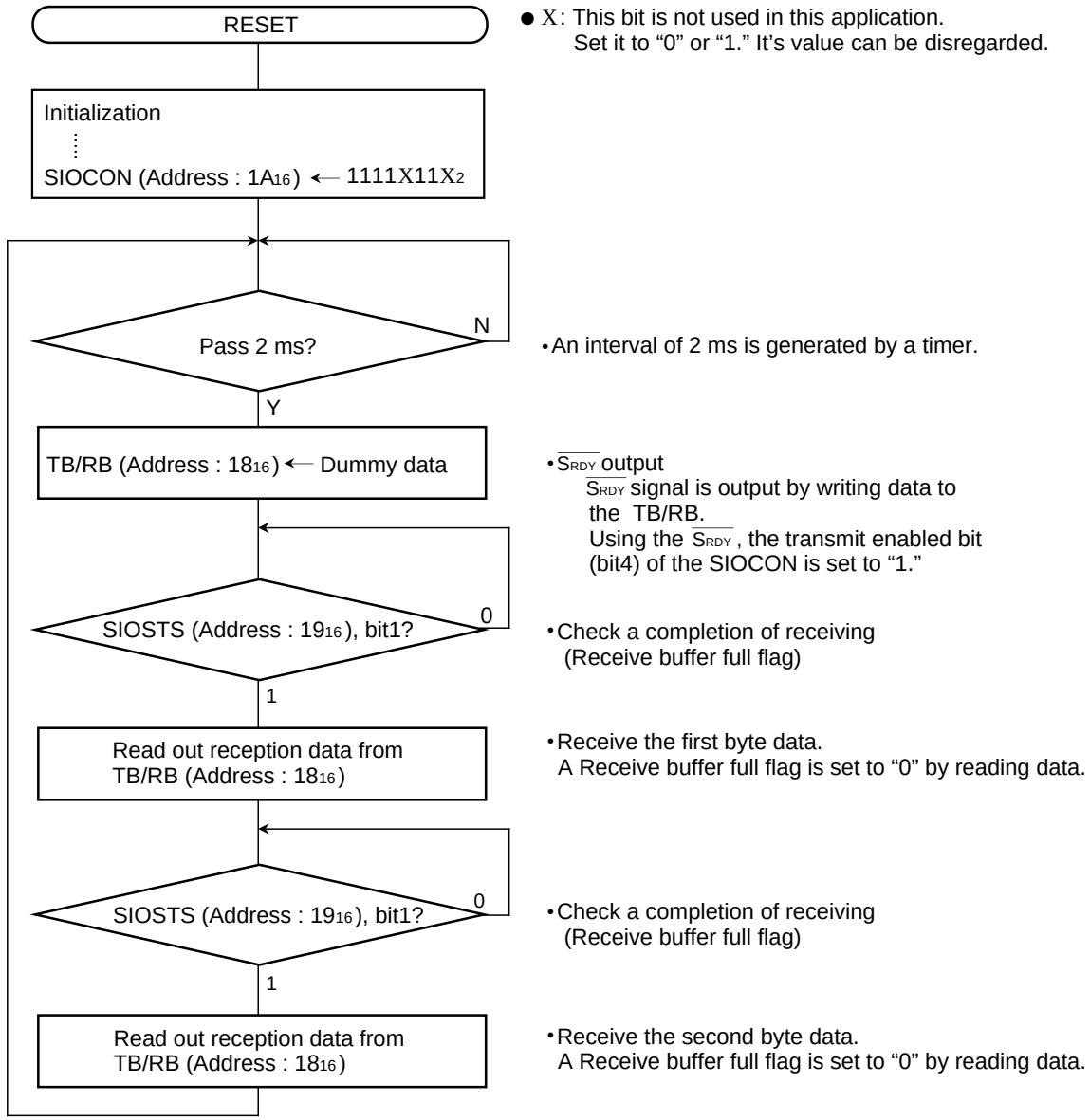


Fig. 2.3.18 Control procedure at a receiving side [Communication using a clock synchronous serial I/O]

2.3 Serial I/O

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

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

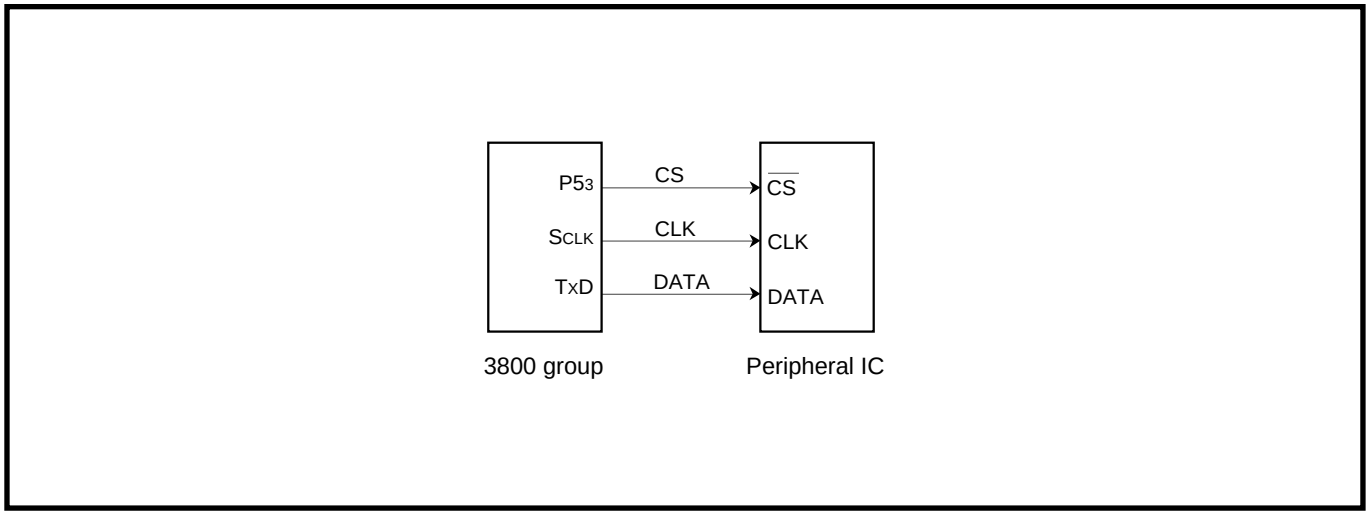


Fig. 2.3.19 Connection diagram [Output of serial data]

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

Figre 2.3.20 shows an output timing chart of serial data.

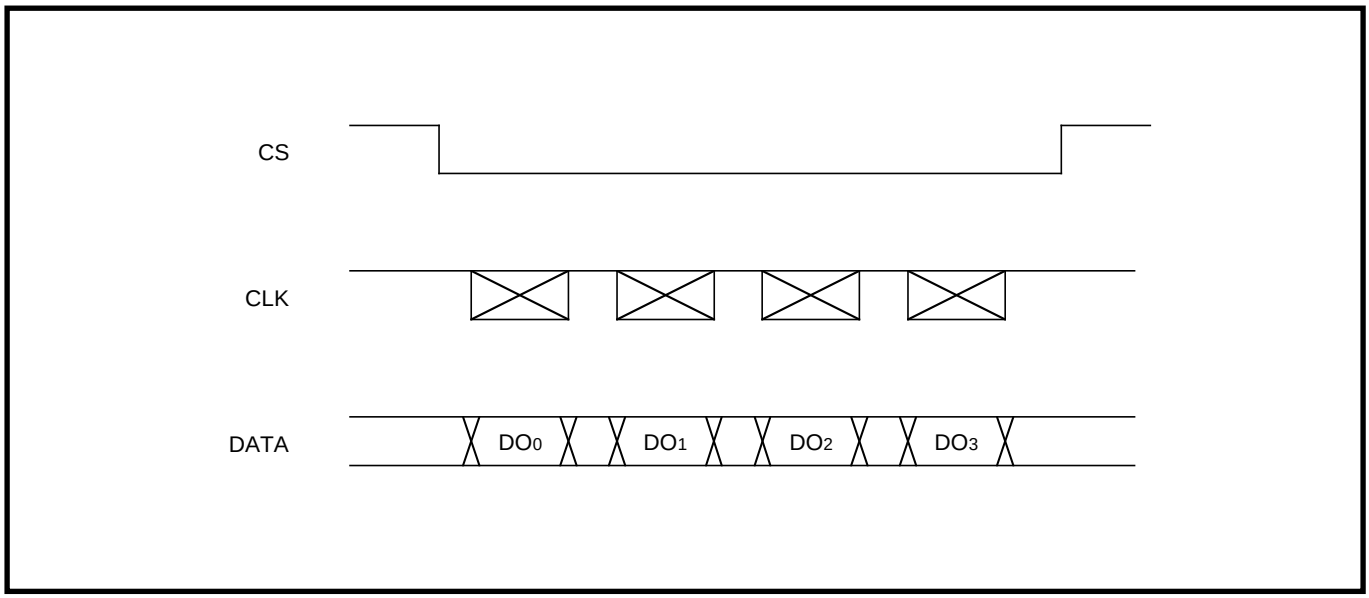


Fig. 2.3.20 Timing chart [Output of serial data]

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

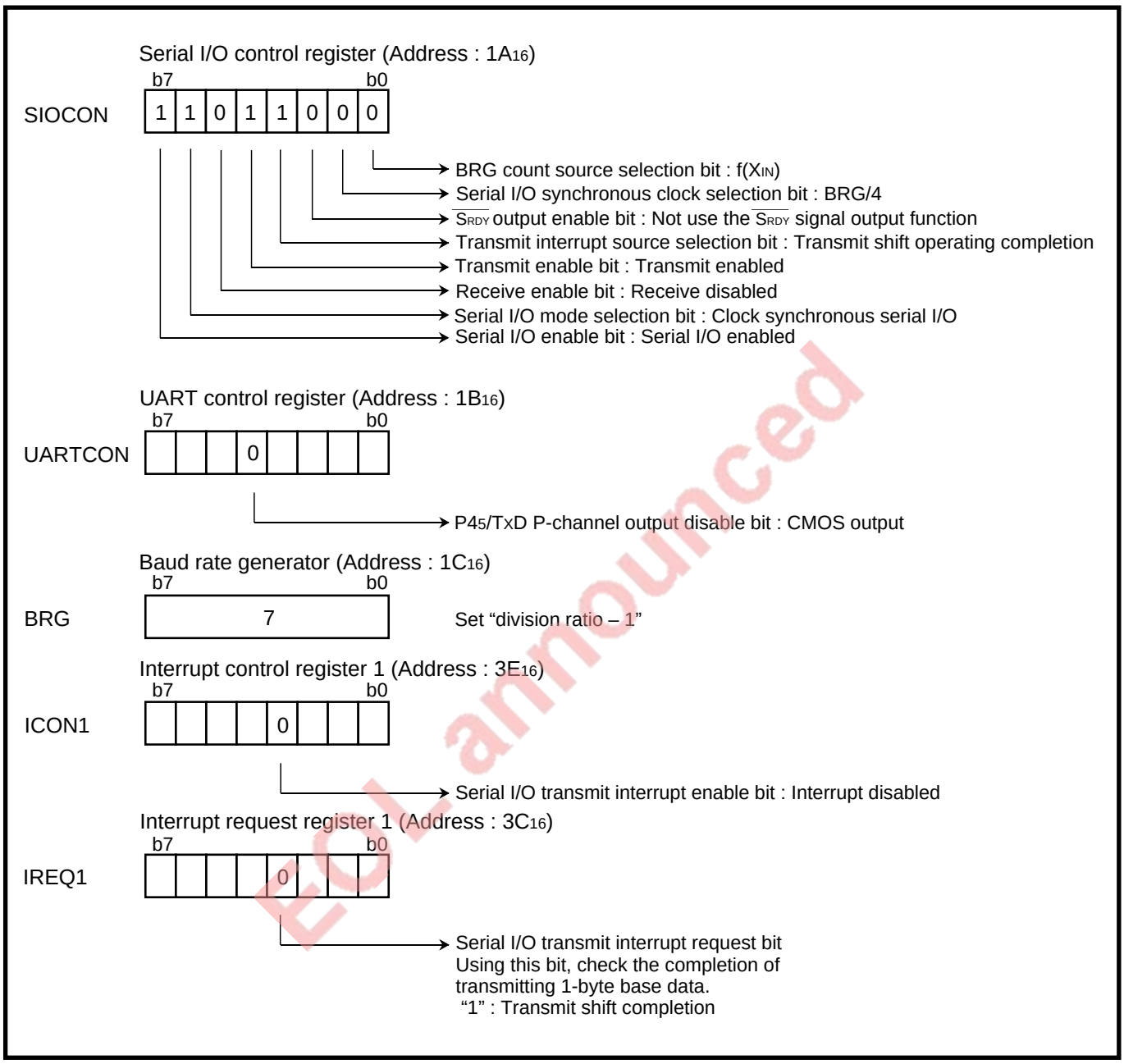


Fig. 2.3.21 Setting of serial I/O related registers [Output of serial data]

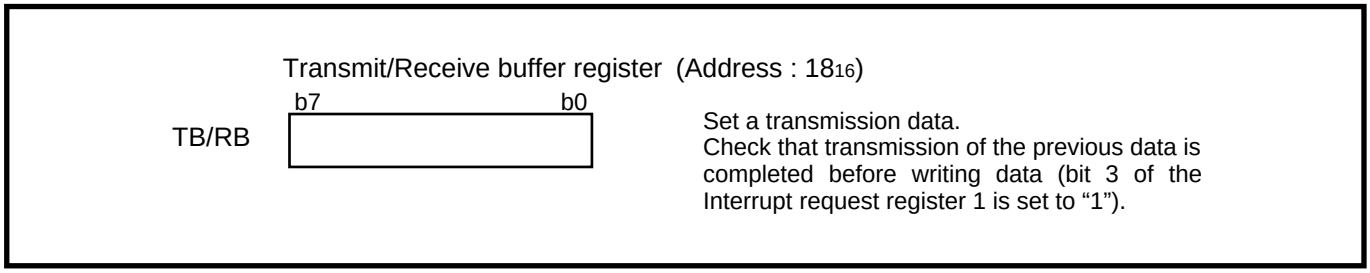


Fig. 2.3.22 Setting of serial I/O transmission data [Output of serial data]

2.3 Serial I/O

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

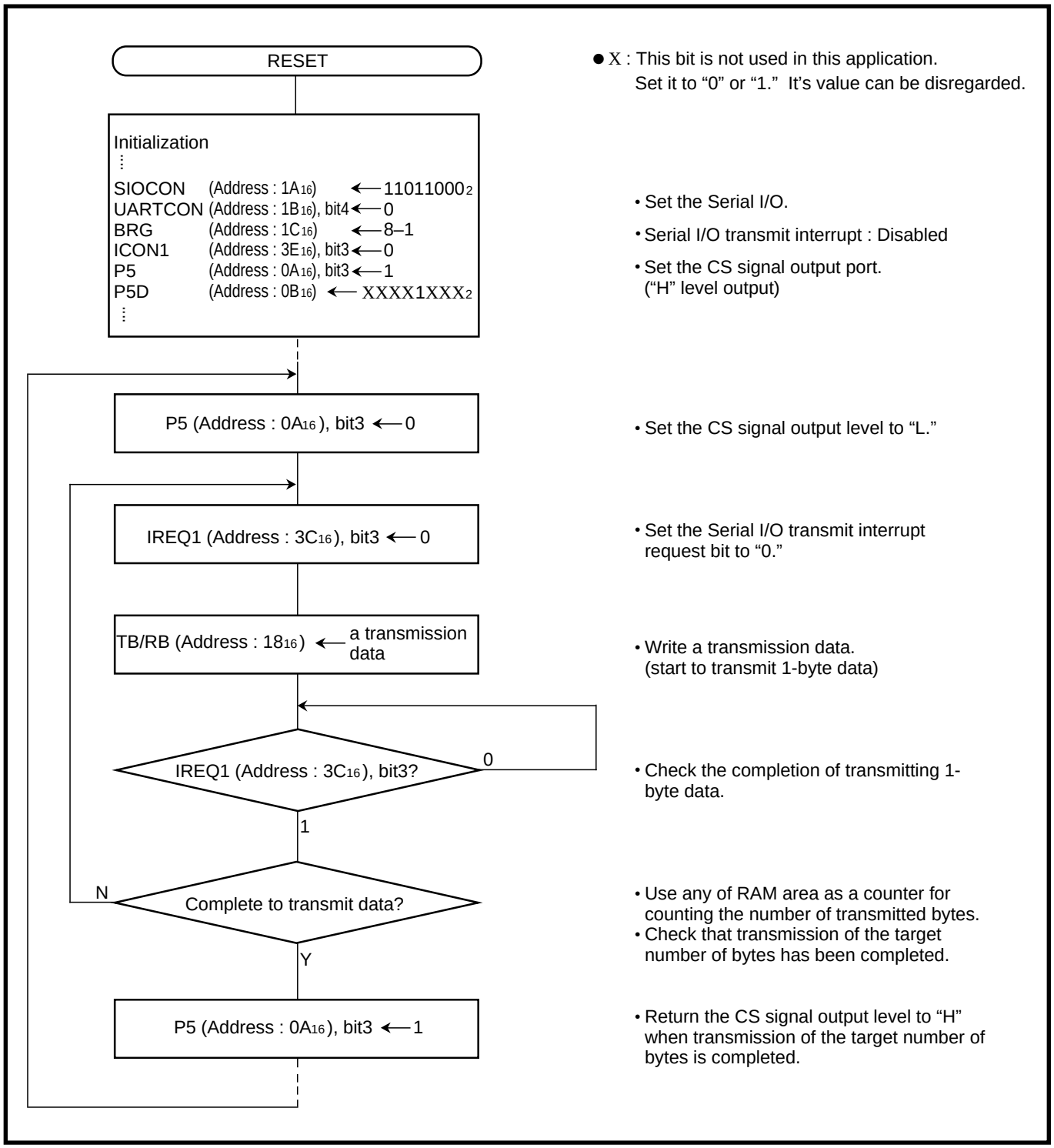


Fig. 2.3.23 Control procedure of serial I/O [Output of serial data]

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

Outline : When a clock synchronous serial I/O is used for communication, synchronization of the clock and the data between the transmitting and receiving sides may be lost because of noise included in the synchronizing clock. Thus, it is necessary to be corrected constantly. This "heading adjustment" is carried out by using the interval between blocks in this example.

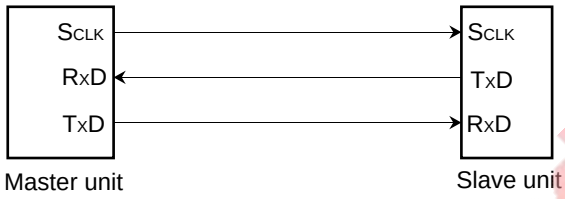


Fig. 2.3.24 Connection diagram [Cyclic transmission or reception of block data between microcomputers]

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

Limitations of the specifications

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

2.3 Serial I/O

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 2.3.26 shows a setting of related registers.

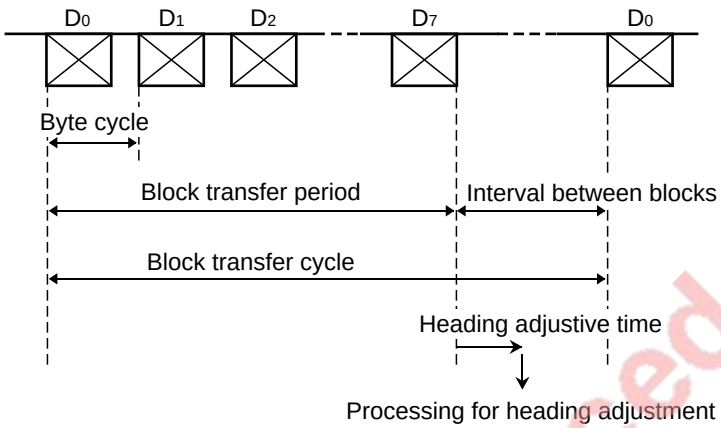


Fig. 2.3.25 Timing chart [Cyclic transmission or reception of block data between microcomputers]

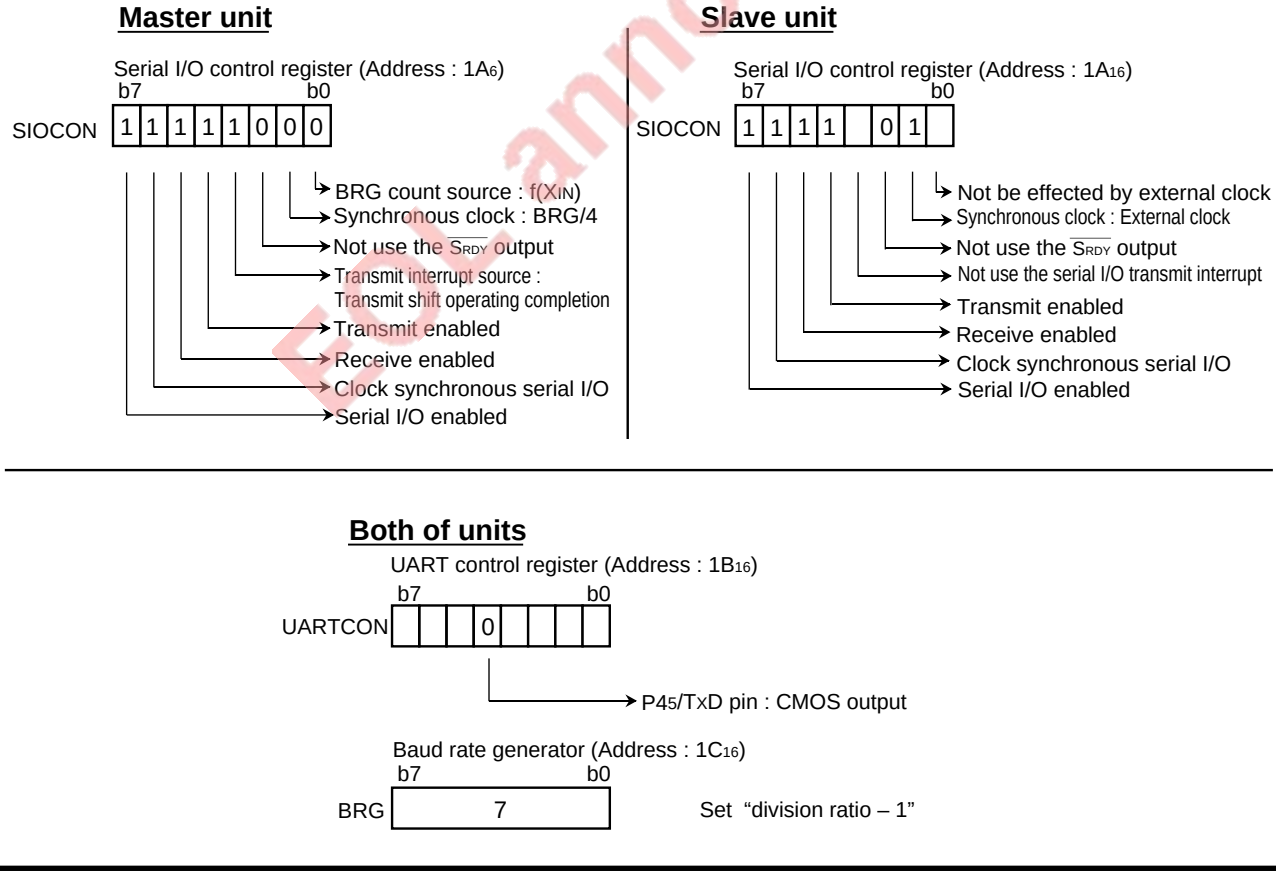


Fig. 2.3.26 Setting of related registers [Cyclic transmission or reception of block data between microcomputers]

Control procedure :

① **Control in the master unit**

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

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

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

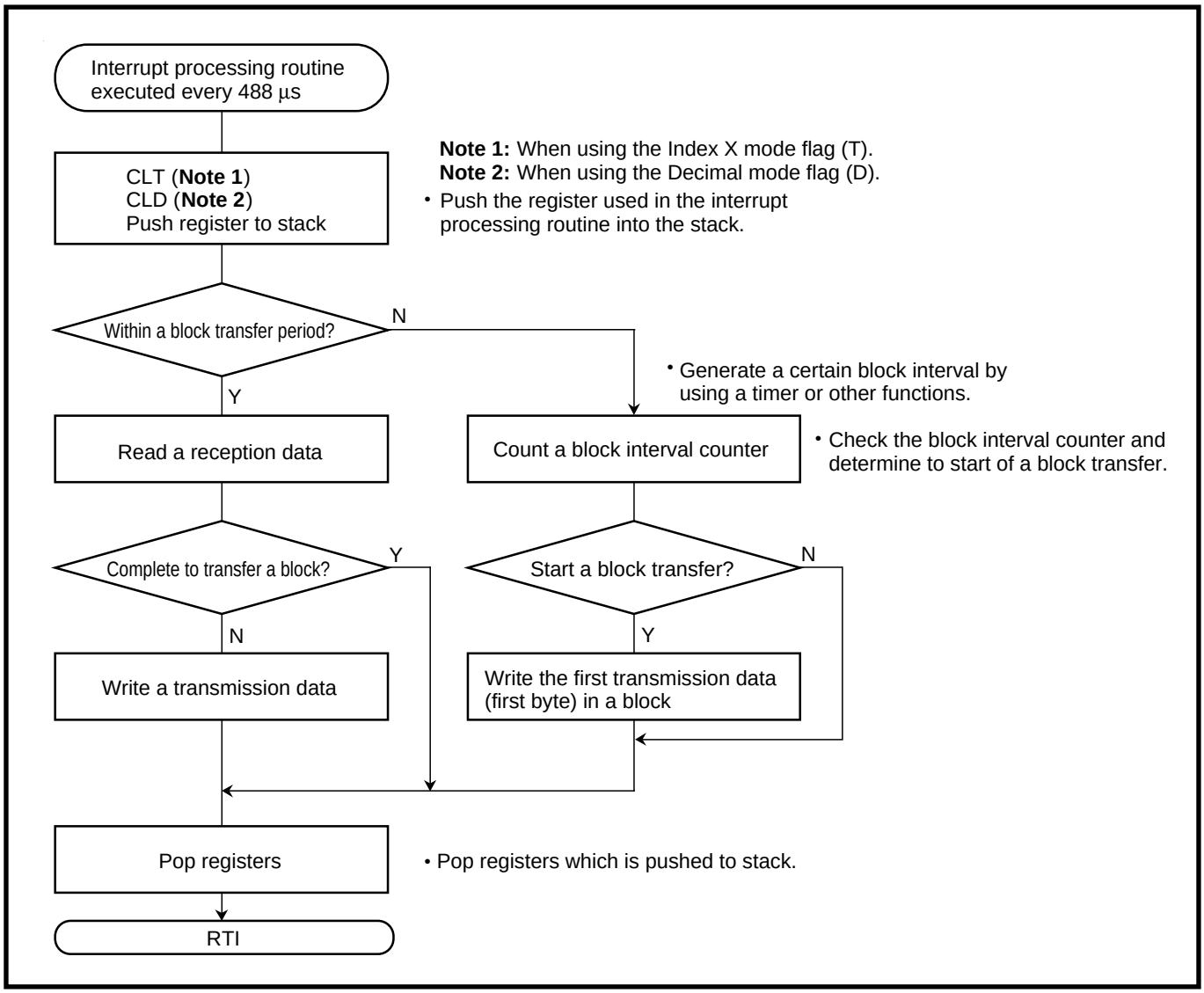


Fig. 2.3.27 Control in the master unit

2.3 Serial I/O

② **Control in the slave unit**

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

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

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

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

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

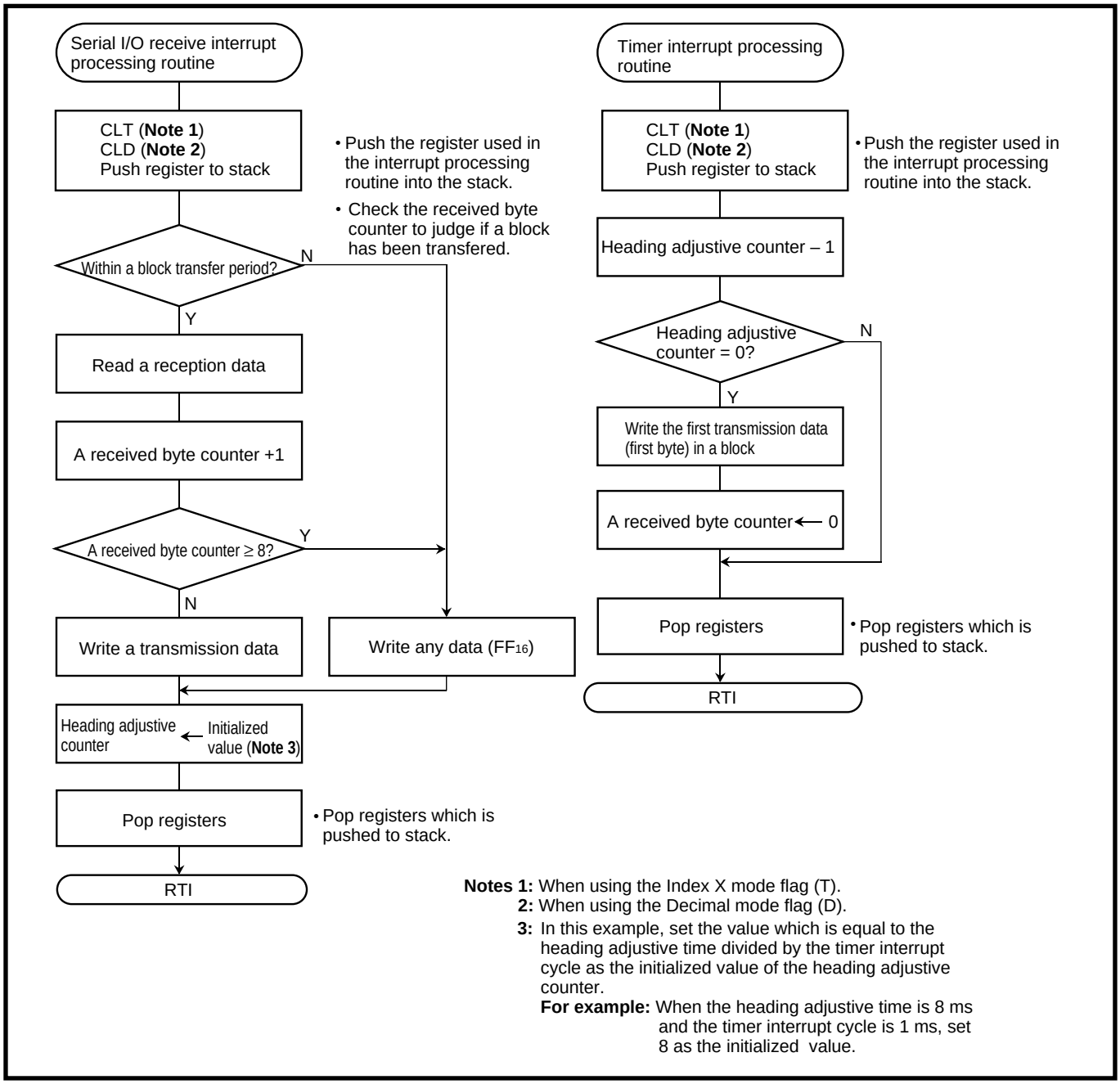


Fig. 2.3.28 Control in the slave unit

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

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

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

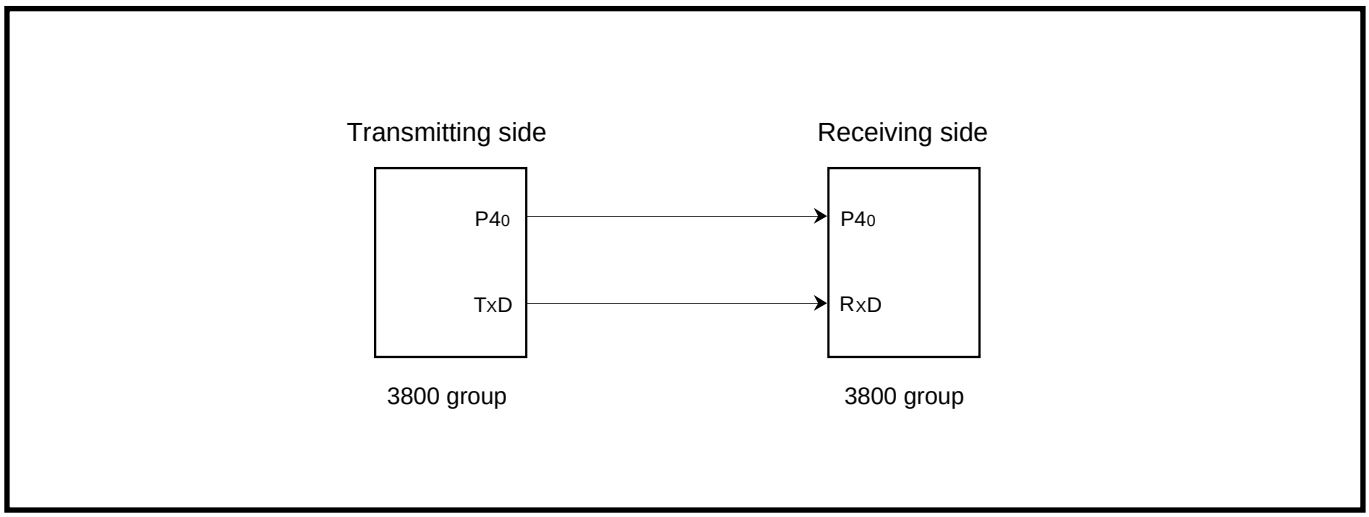


Fig. 2.3.29 Connection diagram [Communication using UART]

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

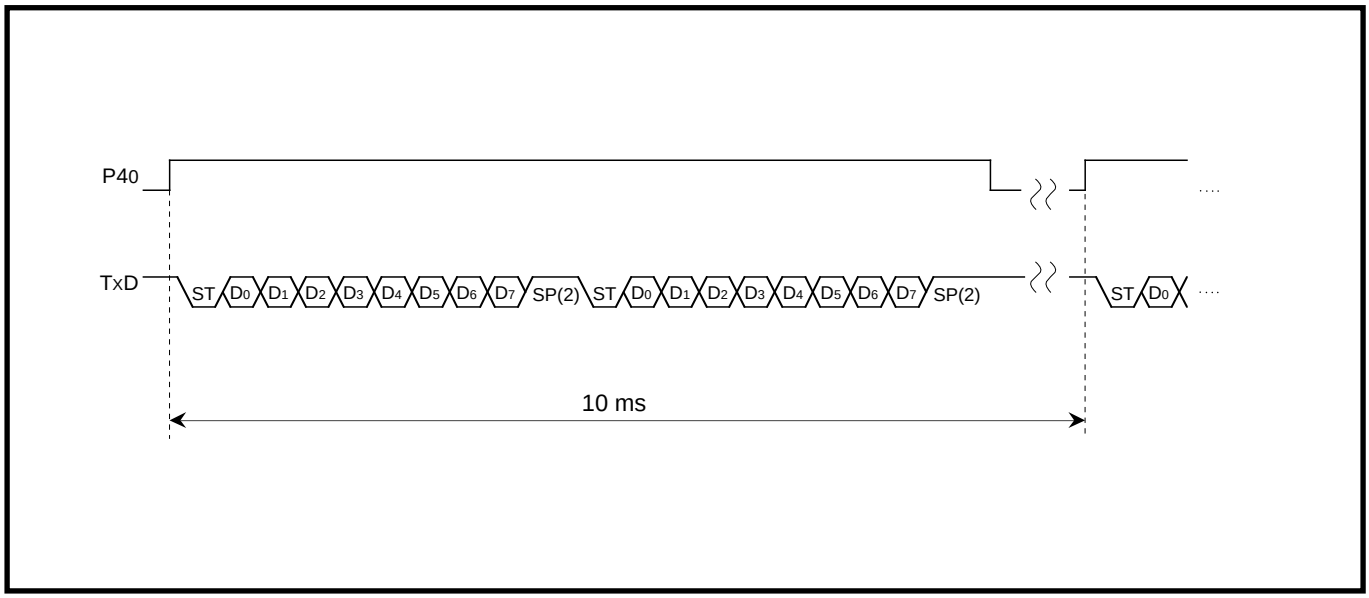


Fig. 2.3.30 Timing chart [Communication using UART]

2.3 Serial I/O

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

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

Transfer bit rate (bps) (Note 1)	BRG count source (Note 2)	at f(XIN) = 4.9152 MHz		at f(XIN) = 7.3728 MHz		at f(XIN) = 8 MHz	
		BRG setting value	Actual time (bps)	BRG setting value	Actual time (bps)	BRG setting value	Actual time (bps)
600	f(XIN)/4	127(7F16)	600.00	191(BF16)	600.00	207(CF16)	600.96
1200	f(XIN)/4	63(3F16)	1200.00	95(5F16)	1200.00	103(6716)	1201.92
2400	f(XIN)/4	31(1F16)	2400.00	47(2F16)	2400.00	51(3316)	2403.85
4800	f(XIN)/4	15(0F16)	4800.00	23(1716)	4800.00	25(1916)	4807.69
9600	f(XIN)/4	7(0716)	9600.00	11(0B16)	9600.00	12(0C16)	9615.38
19200	f(XIN)/4	3(0316)	19200.00	5(0516)	19200.00	5(0516)	20833.33
38400	f(XIN)/4	1(0116)	38400.00	2(0216)	38400.00	2(0216)	41666.67
76800	f(XIN)	3(0316)	76800.00	5(0516)	76800.00	5(0516)	83333.33
31250	f(XIN)	———	———	———	———	15(0F16)	31250.00
62500	f(XIN)	———	———	———	———	7(0716)	62500.00

Notes 1: Equation of transfer bit rate

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

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

2: A BRG count source is selected by bit 0 of the Serial I/O control register (Address : 1A16).

Transmitting side

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

SIOSTS (b7 … b0)

- Transmit buffer empty flag
  - Check to be transferred data from the Transmit buffer register to the Transmit shift register.
  - Writable the next transmission data to the Transmit buffer register at being set to "1."
- Transmit shift register shift completion flag
  Check a completion of transmitting 1-byte data with this flag.
  "1" : Transmit shift completed

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

SIOCON (b7 … b0) = 1 0 0 1 _ 0 0 1

- BRG count source selection bit : $f(X_{IN})/4$
- Serial I/O synchronous clock selection bit : BRG/16
- $\overline{S_{RDY}}$ output enable bit : Not use $\overline{S_{RDY}}$ out
- Transmit enable bit : Transmit enabled
- Receive enable bit : Receive disabled
- Serial I/O mode selection bit : Asynchronous serial I/O(UART)
- Serial I/O enable bit : Serial I/O enabled

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

UARTCON (b7 … b0) = _ _ _ 0 1 _ 0 0

- Character length selection bit : 8 bits
- Parity enable bit : Parity checking disabled
- Stop bit length selection bit : 2 stop bits
- $P4_5/T_XD$ P-channel output disable bit : CMOS output

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

BRG (b7 … b0) = 7

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

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

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

Receiving side

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

b7 b0

SIOSTS

- Receive buffer full flag
  Check a completion of receiving 1-byte data with this flag.
  - "1" : at completing to receive
  - "0" : at reading out a content of the Receive buffer register
- Overrun error flag
  "1" : when data are ready to be transferred to the Receive shift register in the state of storing data into the Receive buffer register.
- Parity error flag
  "1" : when parity error occurs at enabled parity.
- Framing error flag
  "1" : when data can not be received at the timing of setting a stop bit.
- Summing error flag
  "1" : when even one of the following errors occurs.
  - Overrun error
  - Parity error
  - Framing error

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

b7 b0

SIOCON 1 0 1 0 _ 0 0 1

- BRG count source selection bit : $f(X_{IN})/4$
- Serial I/O synchronous clock selection bit : BRG/16
- $\overline{S_{RDY}}$ output enable bit : Not use $\overline{S_{RDY}}$ out
- Transmit enable bit : Transmit disabled
- Receive enable bit : Receive enabled
- Serial I/O mode selection bit : Asynchronous serial I/O(UART)
- Serial I/O enable bit : Serial I/O enabled

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

b7 b0

UARTCON _ _ _ 1 _ _ 0 0

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

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

b7 b0

BRG 7

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

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

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

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

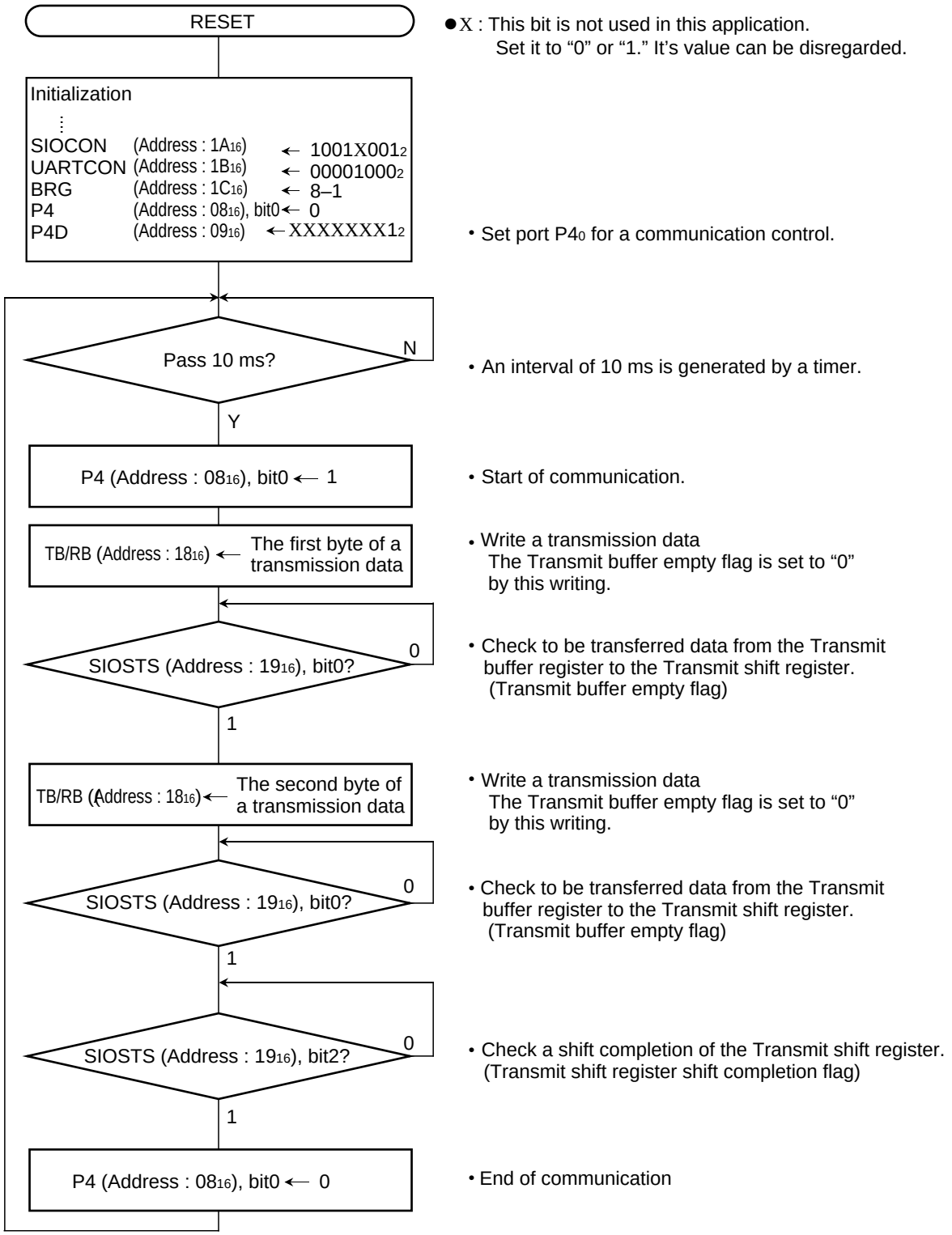


Fig. 2.3.33 Control procedure at a transmitting side [Communication using UART]

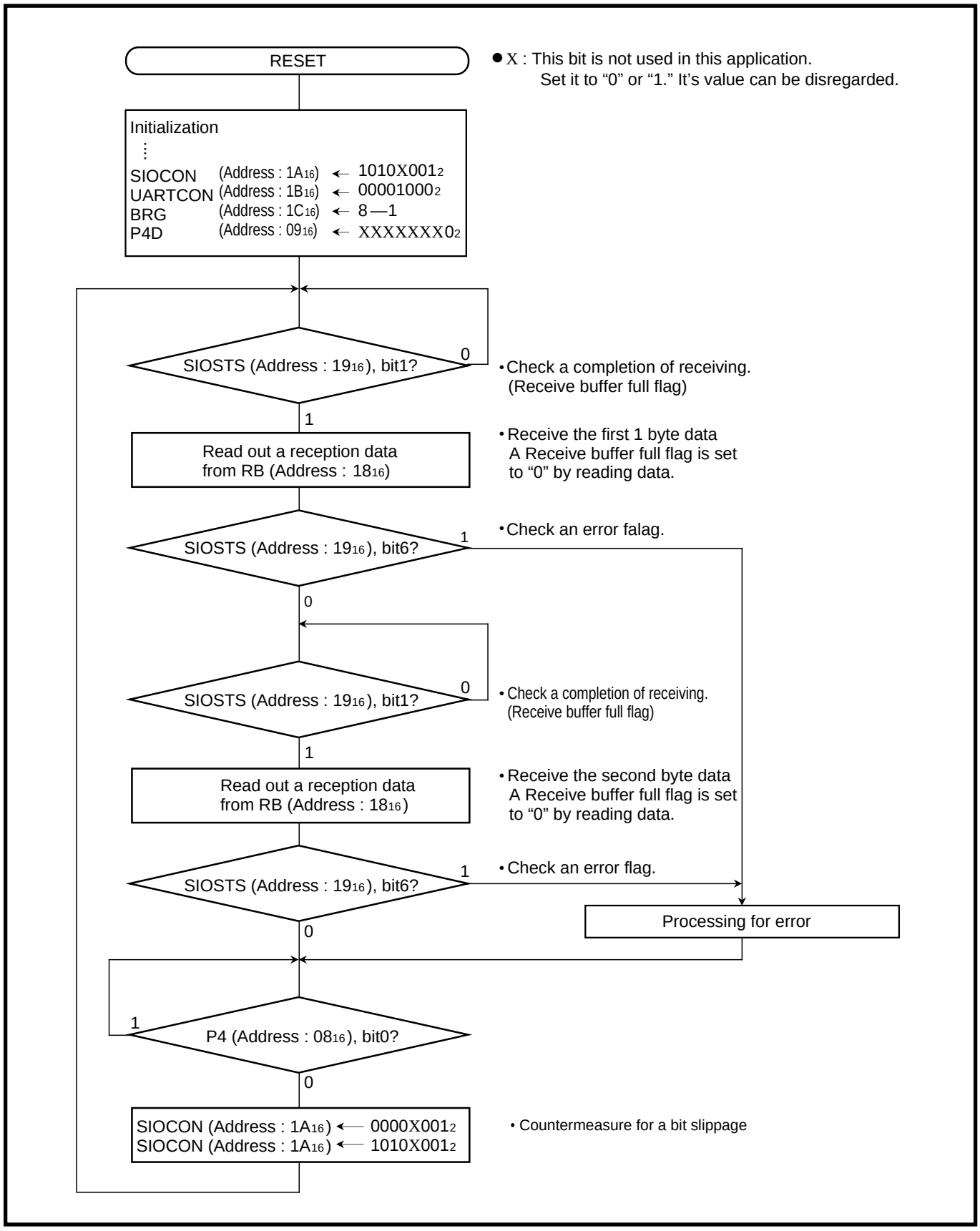


Fig. 2.3.34 Control procedure at a receiving side [Communication using UART]

2.4 Processor mode

2.4.1 Memory map of processor mode

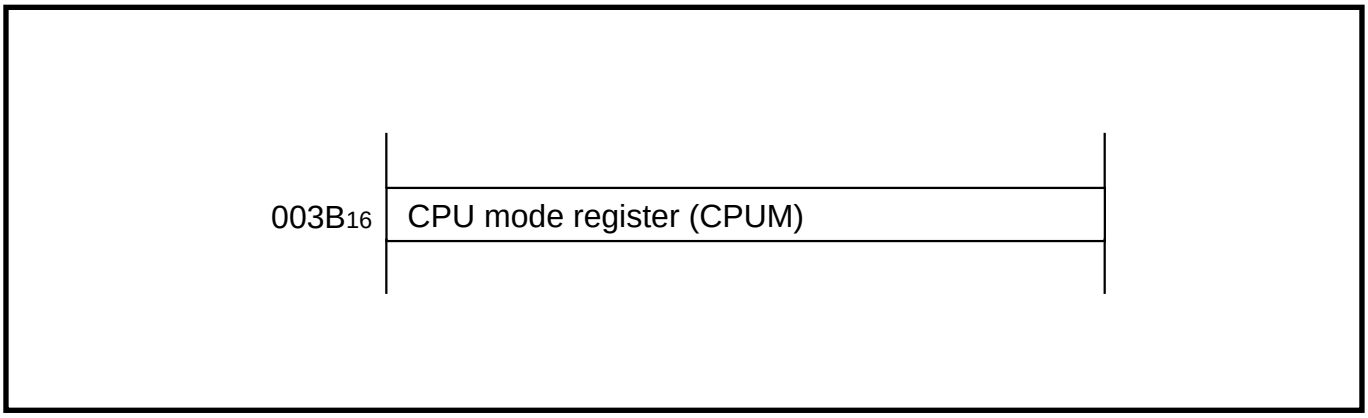


Fig. 2.4.1 Memory map of processor mode related register

2.4.2 Related register

CPU mode register

b7 b6 b5 b4 b3 b2 b1 b0

CPU mode register (CPUM) [Adress : $3B_{16}$]

B	Name	Function	At reset	R	W
0	Processor mode bits	00 : Single-chip mode 01 : Memory expansion mode	0	○	○
1		10 : Microprocessor mode 11 : Not available	*	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	×
3	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are “0.”		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

* An initial value of bit 1 is determined by a level of the CNV_{SS} pin.

Fig. 2.4.2 Structure of CPU mode register

2.4 Processor mode

2.4.3 Processor mode application examples

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

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

At f(X_{IN}) = 8 MHz, an available RAM is given by the following :

- $\overline{OE}$ access time : ta (OE) ≤ 50 ns
- Setup time for writing data : tsu (D) ≤ 65 ns

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

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

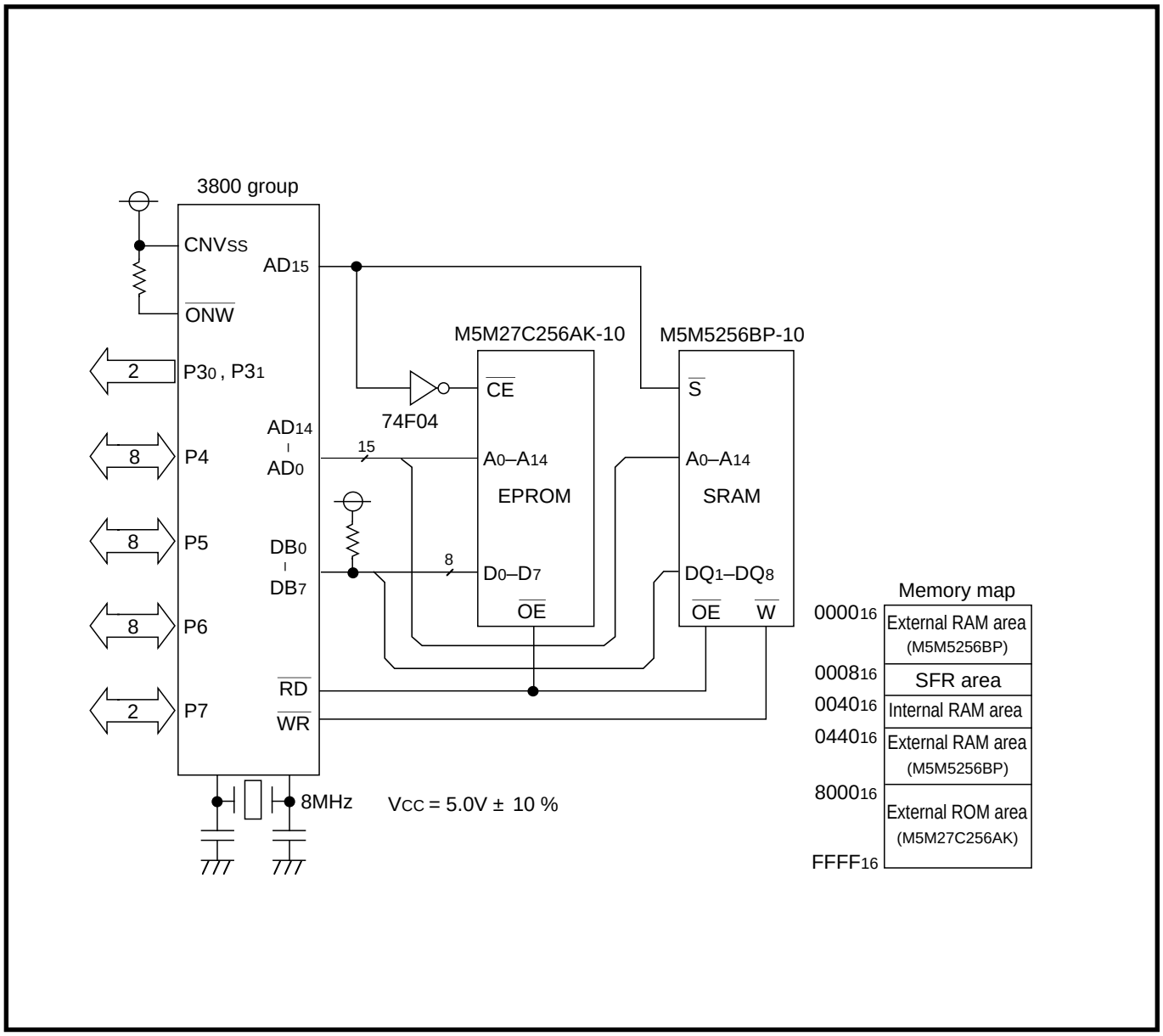


Fig. 2.4.3 Expansion example of ROM and RAM

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

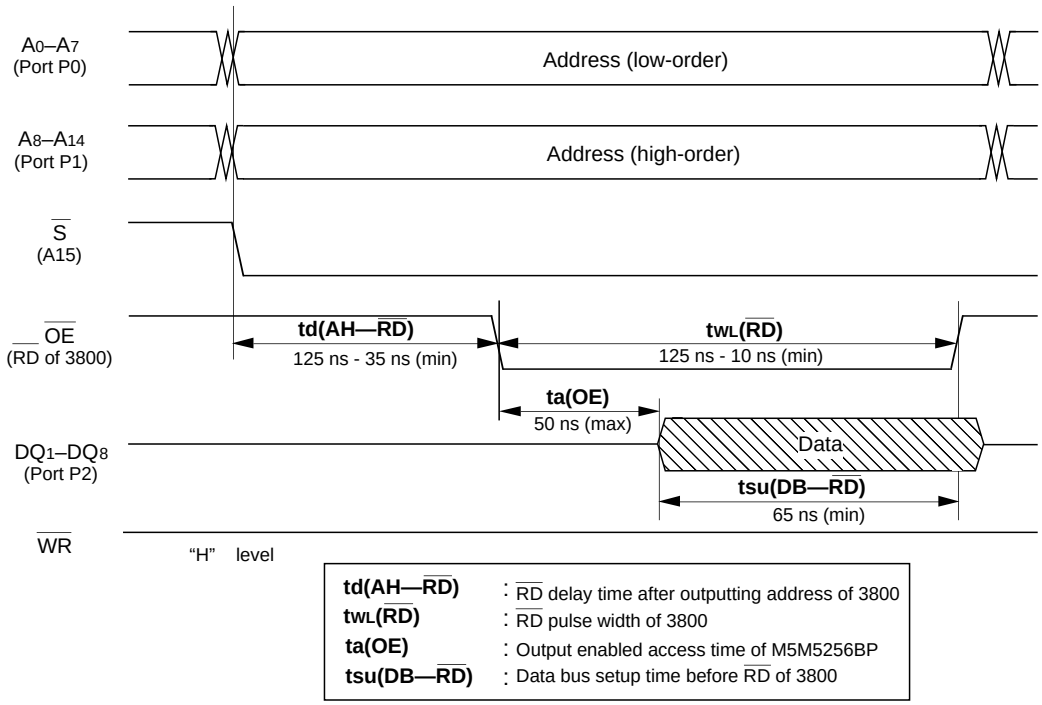


Fig. 2.4.4 Read-cycle (OE access, SRAM)

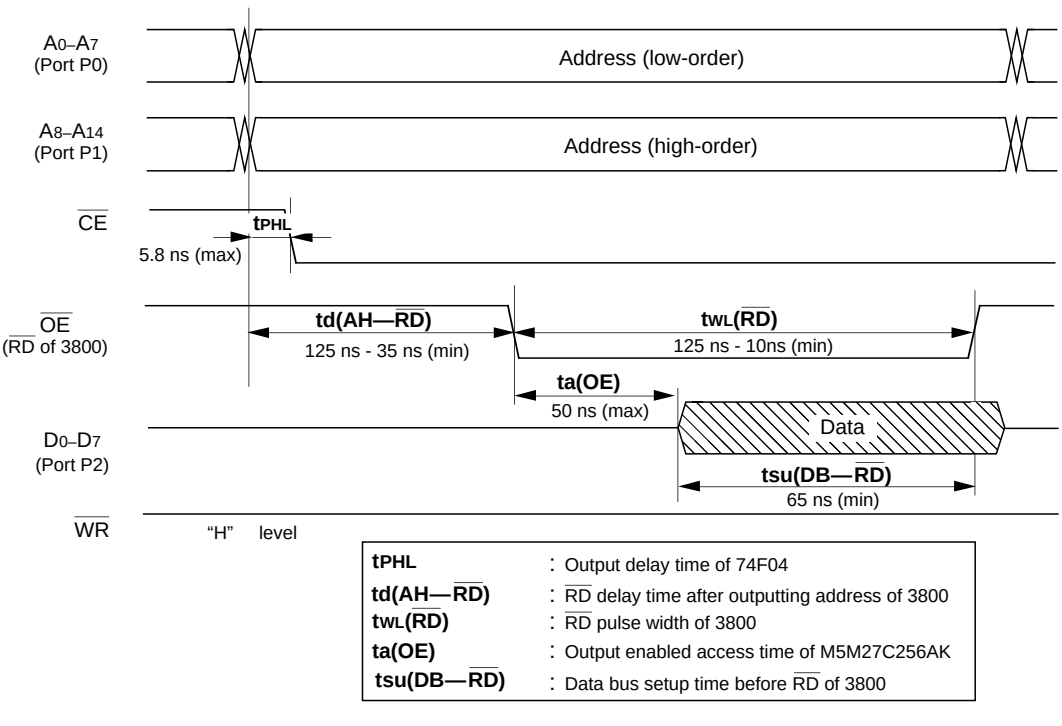
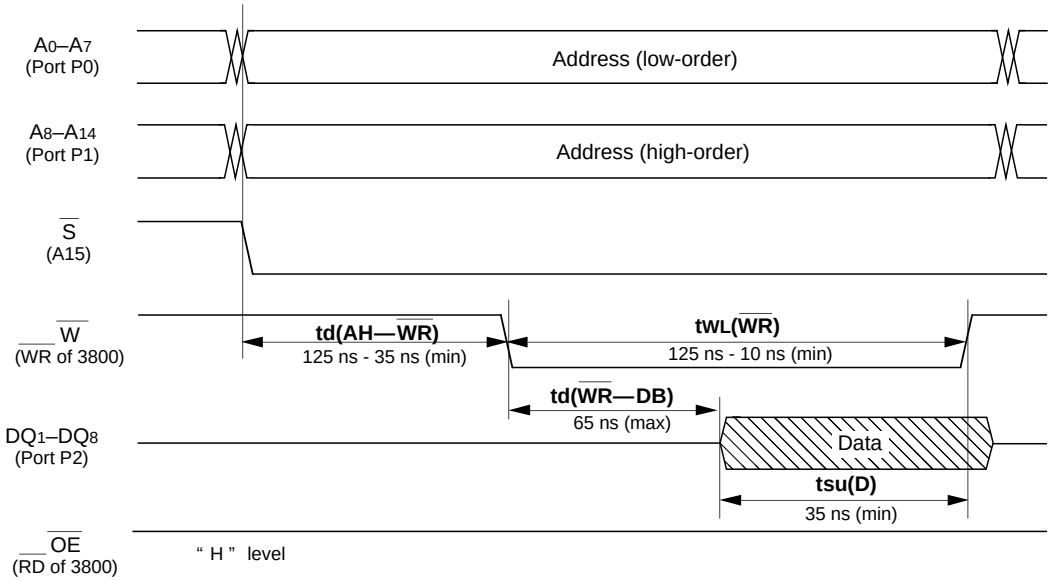


Fig. 2.4.5 Read-cycle (OE access, EPROM)

2.4 Processor mode



td(AH—WR)	: WR delay time after outputting address of 3800
tWL(WR)	: WR pulse width of 3800
td(WR—DB)	: Data bus delay time after WR of 3800
tsu(D)	: Data setup time of M5M5256BP

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

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

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

If "L" level signal is input to the P3$_2$/$\overline{ONW}$ pin while the CPU is in the read or write status, the read or write cycle corresponding to 1 cycle of ϕ is extended. In the extended period, the $\overline{RD}$ or $\overline{WR}$ signal is kept at the "L" level. The $\overline{ONW}$ function operates only when data is read from or written into addresses 0000$_{16}$ to 0007$_{16}$ and addresses 0440$_{16}$ to FFFF$_{16}$.

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

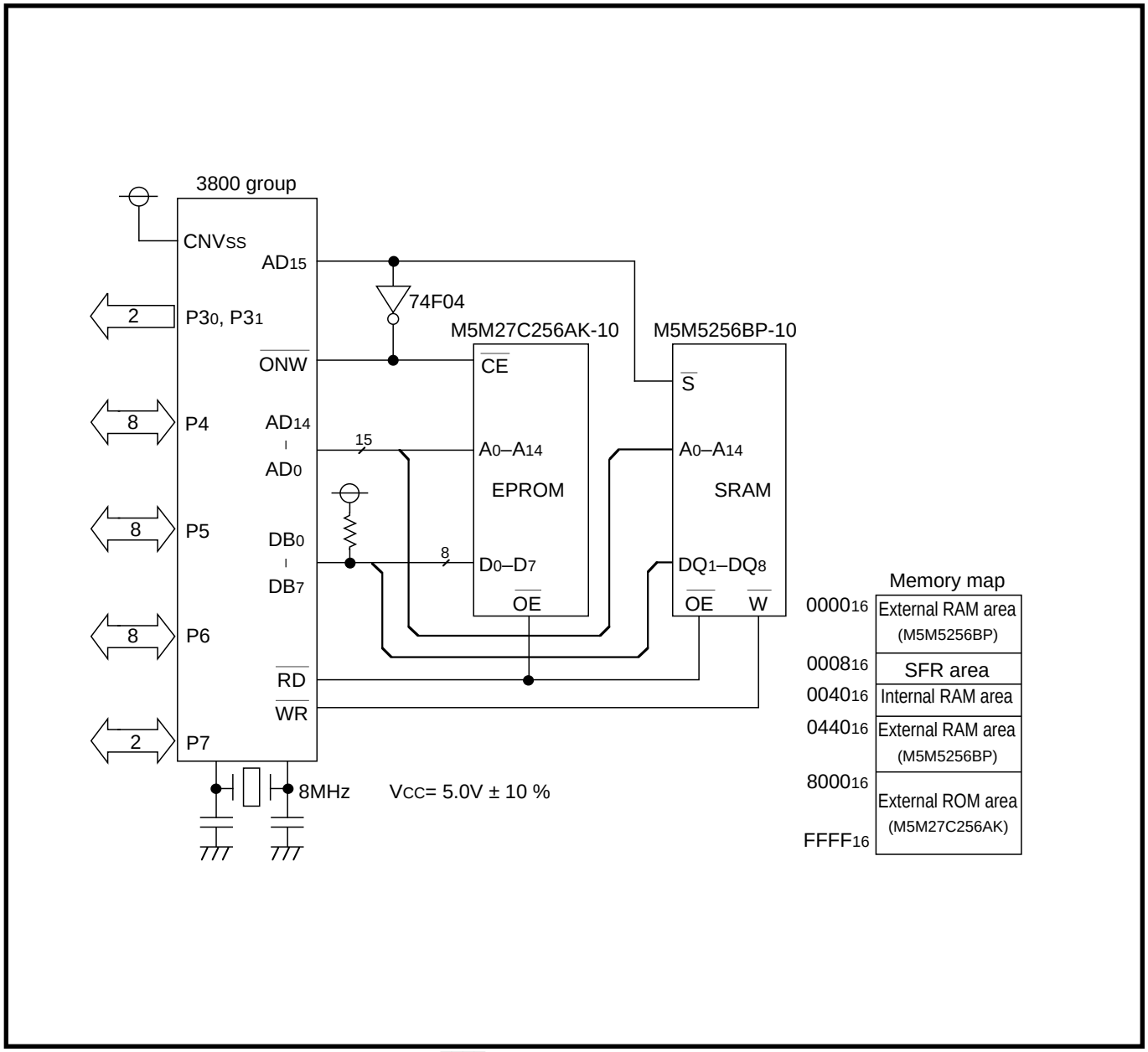


Fig. 2.4.7 Application example of the $\overline{ONW}$ function

2.5 Reset

2.5.1 Connection example of reset IC

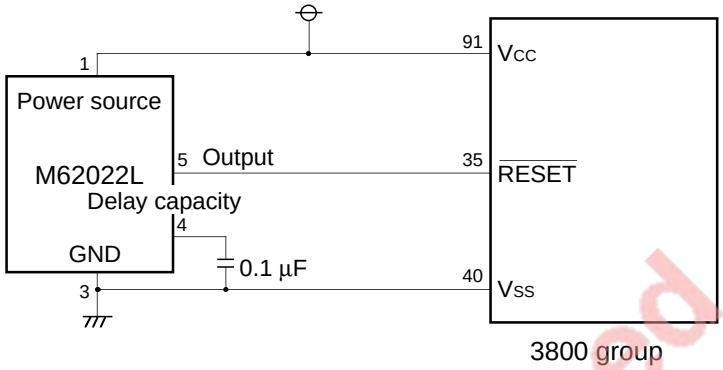


Fig. 2.5.1 Example of Poweron reset circuit

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

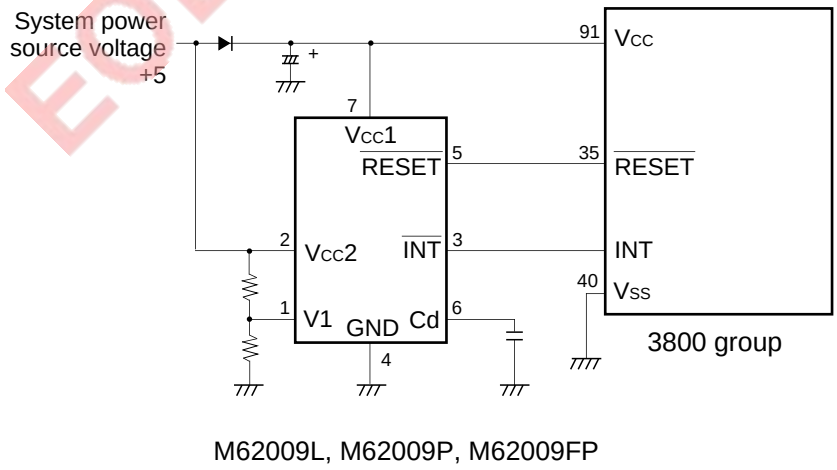


Fig. 2.5.2 RAM back-up system

CHAPTER 3
APPENDIX

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

3.1 Electrical characteristics

3.1.1 Absolute maximum ratings

Table 3.1.1 Absolute maximum ratings

Symbol	Parameter		Conditions	Ratings	Unit
V_{CC}	Power source voltage		All voltages are based on V_{SS}. Output transistors are cut off.	–0.3 to 7.0	V
V_I	Input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$		−0.3 to V_{CC} +0.3	V
V_I	Input voltage	$\overline{RESET}$, X_{IN}		–0.3 to V_{CC} +0.3	V
V_I	Input voltage	CNV_{SS}		–0.3 to 13	V
V_O	Output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$, X_{OUT}		–0.3 to V_{CC} +0.3	V
P_d	Power dissipation		T_a = 25 °C	1000(Note)	mW
T_{opr}	Operating temperature			–20 to 85	°C
T_{stg}	Storage temperature			–40 to 125	°C

Note : 300 mW in case of the flat package.

3.1.2 Recommended operating conditions

Table 3.1.2 Recommended operating conditions (V_{CC} = 3.0 to 5.5 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage (Note 1)	($f(X_{IN}) \leq$ 2 MHz)	3.0	5.0	5.5	V
		($f(X_{IN})$ = 8 MHz)	4.0	5.0	5.5	
V_{SS}	Power source voltage			0		V
V_{IH}	“H” input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$	0.8 V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage	$\overline{RESET}$, X_{IN}, CNV_{SS}	0.8 V_{CC}		V_{CC}	V
V_{IL}	“L” input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$	0		0.2 V_{CC}	V
V_{IL}	“L” input voltage	$\overline{RESET}$, CNV_{SS}	0		0.2 V_{CC}	V
V_{IL}	“L” input voltage	X_{IN}	0		0.16 V_{CC}	V
$\Sigma I_{OH(peak)}$	“H” total peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$(Note 2)			–80	mA
$\Sigma I_{OH(peak)}$	“H” total peak output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$(Note 2)			–80	mA
$\Sigma I_{OL(peak)}$	“L” total peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$(Note 2)			80	mA
$\Sigma I_{OL(peak)}$	“L” total peak output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$(Note 2)			80	mA
$\Sigma I_{OH(avg)}$	“H” total average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$(Note 2)			–40	mA
$\Sigma I_{OH(avg)}$	“H” total average output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$(Note 2)			–40	mA
$\Sigma I_{OL(avg)}$	“L” total average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$(Note 2)			40	mA
$\Sigma I_{OL(avg)}$	“L” total average output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$(Note 2)			40	mA
$I_{OH(peak)}$	“H” peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$(Note 3)			–10	mA
$I_{OL(peak)}$	“L” peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$(Note 3)			10	mA
$I_{OH(avg)}$	“H” average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$(Note 4)			–5	mA
$I_{OL(avg)}$	“L” average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$(Note 4)			5	mA
$f(X_{IN})$	Internal clock oscillation frequency (4.0 V≤Vcc≤5.5 V)				8	MHz
	Internal clock oscillation frequency (3.0 V≤Vcc≤4.0 V)				6 V_{CC}–16	

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

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

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

4: The average output current $I_{OL(avg)}$, $I_{OH(avg)}$ in an average value measured over 100 ms.

3.1.3 Electrical characteristics

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

Symbol	Parameter		Test conditions		Limits			Unit
					Min.	Typ.	Max.	
V_{OH}	"H" output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note)	I_{OH} = –10 mA V_{CC} = 4.0 to 5.5 V		V_{CC}–2.0			V
			I_{OH} = –1.0 mA V_{CC} = 3.0 to 5.5 V		V_{CC}–1.0			
V_{OL}	"L" output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$	I_{OL} = 10 mA V_{CC} = 4.0 to 5.5 V				2.0	V
			I_{OL} = 1.0 mA V_{CC} = 3.0 to 5.5 V				1.0	
$V_{T+} - V_{T-}$	Hysteresis	$CNTR_0$, $CNTR_1$, INT_0–INT_5				0.4		V
$V_{T+} - V_{T-}$	Hysteresis	RxD, SCLK				0.5		V
$V_{T+} - V_{T-}$	Hysteresis	$\overline{RESET}$				0.5		V
I_{IH}	"H" input current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$	V_I = V_{CC}				5.0	µA
I_{IH}	"H" input current	$\overline{RESET}$, CNV_{SS}	V_I = V_{CC}				5.0	µA
I_{IH}	"H" input current	X_{IN}	V_I = V_{CC}			4		µA
I_{IL}	"L" input current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ $\overline{RESET}$, CNV_{SS}	V_I = V_{SS}				–5.0	µA
I_{IL}	"L" input current	X_{IN}	V_I = V_{SS}			–4		µA
V_{RAM}	RAM hold voltage		When clock stopped		2.0		5.5	V
I_{CC}	Power source current		f(X_{IN}) = 8 MHz, V_{CC} = 5 V			6.4	13	mA
			f(X_{IN}) = 5 MHz, V_{CC} = 5 V			4	8	
			f(X_{IN}) = 2 MHz, V_{CC} = 3 V			0.8	2.0	
			When WIT instruction is executed with f(X_{IN}) = 8 MHz, V_{CC} = 5 V			1.5		
			When WIT instruction is executed with f(X_{IN}) = 5 MHz, V_{CC} = 5 V			1		
			When WIT instruction is executed with f(X_{IN}) = 2 MHz, V_{CC} = 3 V			0.2		
			When STP instruction is executed with clock stopped, output transistors isolated.	T_a = 25 °C		0.1	1	µA
				T_a = 85 °C			10	

Note : $P4_5$ is measured when the $P4_5$/TxD P-channel output disable bit of the UART control register (bit 4 of address $001B_{16}$) is "0".

APPENDIX

3.1 Electrical characteristics

3.1.4 Timing requirements and Switching characteristics

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

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input “L” pulse width	2			µs
$t_{c(X_{IN})}$	External clock input cycle time	125			ns
$t_{WH(X_{IN})}$	External clock input “H” pulse width	50			ns
$t_{WL(X_{IN})}$	External clock input “L” pulse width	50			ns
$t_{c(CNTR)}$	$CNTR_0$, $CNTR_1$ input cycle time	200			ns
$t_{WH(CNTR)}$	$CNTR_0$, $CNTR_1$ input “H” pulse width	80			ns
$t_{WL(CNTR)}$	$CNTR_0$, $CNTR_1$ input “L” pulse width	80			ns
$t_{WH(INT)}$	INT_0 to INT_5 input “H” pulse width	80			ns
$t_{WL(INT)}$	INT_0 to INT_5 input “L” pulse width	80			ns
$t_{c(S_{CLK})}$	Serial I/O clock input cycle time (Note)	800			ns
$t_{WH(S_{CLK})}$	Serial I/O clock input “H” pulse width (Note)	370			ns
$t_{WL(S_{CLK})}$	Serial I/O clock input “L” pulse width (Note)	370			ns
$t_{su(R_xD–S_{CLK})}$	Serial I/O input set up time	220			ns
$t_{h(S_{CLK}–R_xD)}$	Serial I/O input hold time	100			ns

Note: When bit 6 of address $001A_{16}$ is “1”. Divide this value by four when bit 6 of address $001A_{16}$ is “0”.

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

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input “L” pulse width	2			µs
$t_{c(X_{IN})}$	External clock input cycle time	500/ (3 V_{CC}–8)			ns
$t_{WH(X_{IN})}$	External clock input “H” pulse width	200/ (3 V_{CC}–8)			ns
$t_{WL(X_{IN})}$	External clock input “L” pulse width	200/ (3 V_{CC}–8)			ns
$t_{c(CNTR)}$	$CNTR_0$, $CNTR_1$ input cycle time	500			ns
$t_{WH(CNTR)}$	$CNTR_0$, $CNTR_1$ input “H” pulse width	230			ns
$t_{WL(CNTR)}$	$CNTR_0$, $CNTR_1$ input “L” pulse width	230			ns
$t_{WH(INT)}$	INT_0 to INT_5 input “H” pulse width	230			ns
$t_{WL(INT)}$	INT_0 to INT_5 input “L” pulse width	230			ns
$t_{c(S_{CLK})}$	Serial I/O clock input cycle time (Note)	2000			ns
$t_{WH(S_{CLK})}$	Serial I/O clock input “H” pulse width (Note)	950			ns
$t_{WL(S_{CLK})}$	Serial I/O clock input “L” pulse width (Note)	950			ns
$t_{su(R_xD–S_{CLK})}$	Serial I/O input set up time	400			ns
$t_{h(S_{CLK}–R_xD)}$	Serial I/O input hold time	200			ns

Note: When bit 6 of address $001A_{16}$ is “1” (clock synchronous mode). Divide this value by four when bit 6 of address $001A_{16}$ is “0” (UART mode).

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

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

Note1: When the P4$_5$/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B$_{16}$) is "0".
2: X_{OUT} pin is excluded.

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

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

Note1: When the P4$_5$/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B$_{16}$) is "0".
2: X_{OUT} pin is excluded.

3.1 Electrical characteristics

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

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

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

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

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_{c(\phi)}$	ϕ clock cycle time	Fig. 3.1.1		$2t_{c(X_{IN})}$		ns
$t_{wH(\phi)}$	ϕ clock “H” pulse width		$t_{c(X_{IN})}$–10			ns
$t_{wL(\phi)}$	ϕ clock “L” pulse width		$t_{c(X_{IN})}$–10			ns
$t_{d(\phi-AH)}$	After ϕ AD_{15}–AD_8 delay time			20	40	ns
$t_{v(\phi-AH)}$	After ϕ AD_{15}–AD_8 valid time		6	10		ns
$t_{d(\phi-AL)}$	After ϕ AD_7–AD_0 delay time			25	45	ns
$t_{v(\phi-AL)}$	After ϕ AD_7–AD_0 valid time		6	10		ns
$t_{d(\phi-SYNC)}$	SYNC delay time			20		ns
$t_{v(\phi-SYNC)}$	SYNC valid time			10		ns
$t_{d(\phi-\overline{WR})}$	$\overline{RD}$ and $\overline{WR}$ delay time			10	20	ns
$t_{v(\phi-\overline{WR})}$	$\overline{RD}$ and $\overline{WR}$ valid time		3	5	10	ns
$t_{d(\phi-DB)}$	After ϕ data bus delay time			20	70	ns
$t_{v(\phi-DB)}$	After ϕ data bus valid time		15			ns
$t_{wL(\overline{RD})}$ $t_{wL(\overline{WR})}$	$\overline{RD}$ pulse width, $\overline{WR}$ pulse width		$t_{c(X_{IN})}$–10			ns
	$\overline{RD}$ pulse width, $\overline{WR}$ pulse width (When one-wait is valid)		$3t_{c(X_{IN})}$–10			ns
$t_{d(AH-\overline{RD})}$ $t_{d(AH-\overline{WR})}$	After AD_{15}–AD_8 $\overline{RD}$ delay time After AD_{15}–AD_8 $\overline{WR}$ delay time		$t_{c(X_{IN})}$–35	$t_{c(X_{IN})}$–15		ns
$t_{d(AL-\overline{RD})}$ $t_{d(AL-\overline{WR})}$	After AD_7–AD_0 $\overline{RD}$ delay time After AD_7–AD_0 $\overline{WR}$ delay time		$t_{c(X_{IN})}$–40	$t_{c(X_{IN})}$–20		ns
$t_{v(\overline{RD}-AH)}$ $t_{v(\overline{WR}-AH)}$	After $\overline{RD}$ AD_{15}–AD_8 valid time After $\overline{WR}$ AD_{15}–AD_8 valid time		0	5		ns
$t_{v(\overline{RD}-AL)}$ $t_{v(\overline{WR}-AL)}$	After $\overline{RD}$ AD_7–AD_0 valid time After $\overline{WR}$ AD_7–AD_0 valid time		0	5		ns
$t_{d(\overline{WR}-DB)}$	After $\overline{WR}$ data bus delay time			15	65	ns
$t_{v(\overline{WR}-DB)}$	After $\overline{WR}$ data bus valid time		10			ns
$t_{d(\overline{RESET}-\overline{RESET_{OUT}})}$	$\overline{RESET_{OUT}}$ output delay time				200	ns
$t_{v(\phi-\overline{RESET})}$	$\overline{RESET_{OUT}}$ output valid time (Note)		0		200	ns

Note : The $\overline{RESET_{OUT}}$ goes “H” in sync with the fall of the ϕ clock that is anywhere between about 8 cycle and 13 cycles after the $\overline{RESET}$ input goes “H”.

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

(V_{CC} = 3.0 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{su(\overline{ONW}-\phi)}$	Before ϕ $\overline{ONW}$ input set up time	–20			ns
$t_{h(\phi-\overline{ONW})}$	After ϕ $\overline{ONW}$ input hold time	–20			ns
$t_{su(DB-\phi)}$	Before ϕ data bus set up time	180			ns
$t_{h(\phi-DB)}$	After ϕ data bus hold time	0			ns
$t_{su(\overline{ONW}-\overline{RD})}$ $t_{su(\overline{ONW}-\overline{WR})}$	Before $\overline{RD}$ $\overline{ONW}$ input set up time Before $\overline{WR}$ $\overline{ONW}$ input set up time	–20			ns
$t_{h(\overline{RD}-\overline{ONW})}$ $t_{h(\overline{WR}-\overline{ONW})}$	After $\overline{RD}$ $\overline{ONW}$ input hold time After $\overline{WR}$ $\overline{ONW}$ input hold time	–20			ns
$t_{su(DB-\overline{RD})}$	Before $\overline{RD}$ data bus set up time	185			ns
$t_{h(\overline{RD}-DB)}$	After $\overline{RD}$ data bus hold time	0			ns

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

(V_{CC} = 3.0 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_{c(\phi)}$	ϕ clock cycle time	Fig. 3.1.1		$2t_{c(X_{IN})}$		ns
$t_{wH(\phi)}$	ϕ clock "H" pulse width		$t_{c(X_{IN})}$–20			ns
$t_{wL(\phi)}$	ϕ clock "L" pulse width		$t_{c(X_{IN})}$–20			ns
$t_{d(\phi-AH)}$	After ϕ AD_{15}–AD_8 delay time				150	ns
$t_{v(\phi-AH)}$	After ϕ AD_{15}–AD_8 valid time		10	15		ns
$t_{d(\phi-AL)}$	After ϕ AD_7–AD_0 delay time				150	ns
$t_{v(\phi-AL)}$	After ϕ AD_7–AD_0 valid time		10	15		ns
$t_{d(\phi-SYNC)}$	SYNC delay time			40		ns
$t_{v(\phi-SYNC)}$	SYNC valid time			20		ns
$t_{d(\phi-\overline{WR})}$	$\overline{RD}$ and $\overline{WR}$ delay time			15	25	ns
$t_{v(\phi-\overline{WR})}$	$\overline{RD}$ and $\overline{WR}$ valid time		3	7	15	ns
$t_{d(\phi-DB)}$	After ϕ data bus delay time				200	ns
$t_{v(\phi-DB)}$	After ϕ data bus valid time		15			ns
$t_{wL(\overline{RD})}$ $t_{wL(\overline{WR})}$	$\overline{RD}$ pulse width, $\overline{WR}$ pulse width		$t_{c(X_{IN})}$–20			ns
	$\overline{RD}$ pulse width, $\overline{WR}$ pulse width (When one-wait is valid)		$3t_{c(X_{IN})}$–20			ns
$t_{d(AH-\overline{RD})}$ $t_{d(AH-\overline{WR})}$	After AD_{15}–AD_8 $\overline{RD}$ delay time After AD_{15}–AD_8 $\overline{WR}$ delay time		$t_{c(X_{IN})}$–145			ns
$t_{d(AL-\overline{RD})}$ $t_{d(AL-\overline{WR})}$	After AD_7–AD_0 $\overline{RD}$ delay time After AD_7–AD_0 $\overline{WR}$ delay time		$t_{c(X_{IN})}$–145			ns
$t_{v(\overline{RD}-AH)}$ $t_{v(\overline{WR}-AH)}$	After $\overline{RD}$ AD_{15}–AD_8 valid time After $\overline{WR}$ AD_{15}–AD_8 valid time		5	10		ns
$t_{v(\overline{RD}-AL)}$ $t_{v(\overline{WR}-AL)}$	After $\overline{RD}$ AD_7–AD_0 valid time After $\overline{WR}$ AD_7–AD_0 valid time		5	10		ns
$t_{d(\overline{WR}-DB)}$	After $\overline{WR}$ data bus delay time				195	ns
$t_{v(\overline{WR}-DB)}$	After $\overline{WR}$ data bus valid time		10			ns
$t_{d(\overline{RESET}-\overline{RESET_{OUT}})}$	$\overline{RESET_{OUT}}$ output delay time				300	ns
$t_{v(\phi-\overline{RESET})}$	$\overline{RESET_{OUT}}$ output valid time (Note)		0		300	ns

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

3.1 Electrical characteristics

3.1.5 Absolute maximum ratings (Extended operating temperature version)

Table 3.1.12 Absolute maximum ratings (Extended operating temperature version)

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Power source voltage	All voltages are based on V_{SS}. Output transistors are cut off.	–0.3 to 7.0	V
V_I	Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$		–0.3 to V_{CC} +0.3	V
V_I	Input voltage $\overline{RESET}$, X_{IN}		–0.3 to V_{CC} +0.3	V
V_I	Input voltage CNV$_{SS}$		–0.3 to 13	V
V_O	Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$, X_{OUT}		–0.3 to V_{CC} +0.3	V
P_d	Power dissipation	T_a = 25 °C	1000(Note)	mW
T_{opr}	Operating temperature		–40 to 85	°C
T_{stg}	Storage temperature		–65 to 150	°C

Note : 300 mW in case of the flat package.

3.1.6 Recommended operating conditions (Extended operating temperature version)

Table 3.1.13 Recommended operating conditions (Extended operating temperature version)

(V_{CC} = 4.0 to 5.5 V, T_a = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage		4.0	5.0	5.5	V
V_{SS}	Power source voltage			0		V
V_{IH}	“H” input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$	0.8 V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage	$\overline{RESET}$, X_{IN}, CNV$_{SS}$	0.8 V_{CC}		V_{CC}	V
V_{IL}	“L” input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$	0		0.2 V_{CC}	V
V_{IL}	“L” input voltage	$\overline{RESET}$, CNV$_{SS}$	0		0.2 V_{CC}	V
V_{IL}	“L” input voltage	X_{IN}	0		0.16 V_{CC}	V
$\Sigma I_{OH(peak)}$	“H” total peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ (Note 1)			–80	mA
$\Sigma I_{OH(peak)}$	“H” total peak output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 1)			–80	mA
$\Sigma I_{OL(peak)}$	“L” total peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ (Note 1)			80	mA
$\Sigma I_{OL(peak)}$	“L” total peak output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 1)			80	mA
$\Sigma I_{OH(avg)}$	“H” total average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ (Note 1)			–40	mA
$\Sigma I_{OH(avg)}$	“H” total average output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 1)			–40	mA
$\Sigma I_{OL(avg)}$	“L” total average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ (Note 1)			40	mA
$\Sigma I_{OL(avg)}$	“L” total average output current	$P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 1)			40	mA
$I_{OH(peak)}$	“H” peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 2)			–10	mA
$I_{OL(peak)}$	“L” peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 2)			10	mA
$I_{OH(avg)}$	“H” average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 3)			–5	mA
$I_{OL(avg)}$	“L” average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note 3)			5	mA
f(X_{IN})	Internal clock oscillation frequency				8	MHz

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

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

3: The average output current $I_{OL(avg)}$, $I_{OH(avg)}$ in an average value measured over 100 ms.

3.1.7 Electrical characteristics (Extended operating temperature version)

Table 3.1.14 Electrical characteristics (Extended operating temperature version)

(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter		Test conditions		Limits			Unit
					Min.	Typ.	Max.	
V_{OH}	"H" output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$ (Note)	I_{OH} = –10 mA		V_{CC}–2.0			V
V_{OL}	"L" output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$,$P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$	I_{OL} = 10 mA				2.0	V
V_{T+} – V_{T-}	Hysteresis	$CNTR_0$, $CNTR_1$, INT_0–INT_5				0.4		V
V_{T+} – V_{T-}	Hysteresis	R_XD, S_{CLK}				0.5		V
V_{T+} – V_{T-}	Hysteresis	$\overline{RESET}$				0.5		V
I_{IH}	"H" input current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$	V_I = V_{CC}				5.0	µA
I_{IH}	"H" input current	$\overline{RESET}$, CNV_{SS}	V_I = V_{CC}				5.0	µA
I_{IH}	"H" input current	X_{IN}	V_I = V_{CC}			4		µA
I_{IL}	"L" input current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$, $P7_1$, $\overline{RESET}$, CNV_{SS}	V_I = V_{SS}				–5.0	µA
I_{IL}	"L" input current	X_{IN}	V_I = V_{SS}			–4		µA
V_{RAM}	RAM hold voltage		When clock stopped		2.0		5.5	V
I_{CC}	Power source current		$f(X_{IN})$ = 8 MHz			6.4	13	mA
			$f(X_{IN})$ = 5 MHz			4	8	
			When WIT instruction is executed with $f(X_{IN})$ = 8 MHz			1.5		
			When WIT instruction is executed with $f(X_{IN})$ = 5 MHz			1		
			When STP instruction is executed with clock stopped, output transistors isolated.	T_a = 25 °C		0.1	1	µA
				T_a = 85 °C			10	

Note : $P4_5$ is measured when the $P4_5$/T_XD P-channel output disable bit of the UART control register (bit 4 of address $001B_{16}$) is "0".

APPENDIX

3.1 Electrical characteristics

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

Table 3.1.15 Timing requirements (Extended operating temperature version)

(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input “L” pulse width	2			µs
$t_{C(X_{IN})}$	External clock input cycle time	125			ns
$t_{WH(X_{IN})}$	External clock input “H” pulse width	50			ns
$t_{WL(X_{IN})}$	External clock input “L” pulse width	50			ns
$t_{C(CNTR)}$	$CNTR_0$, $CNTR_1$ input cycle time	200			ns
$t_{WH(CNTR)}$	$CNTR_0$, $CNTR_1$ input “H” pulse width	80			ns
$t_{WL(CNTR)}$	$CNTR_0$, $CNTR_1$ input “L” pulse width	80			ns
$t_{WH(INT)}$	INT_0 to INT_5 input “H” pulse width	80			ns
$t_{WL(INT)}$	INT_0 to INT_5 input “L” pulse width	80			ns
$t_{C(SCLK)}$	Serial I/O clock input cycle time (Note)	800			ns
$t_{WH(SCLK)}$	Serial I/O clock input “H” pulse width (Note)	370			ns
$t_{WL(SCLK)}$	Serial I/O clock input “L” pulse width (Note)	370			ns
$t_{su(RxD-SCLK)}$	Serial I/O input set up time	220			ns
$t_{h(SCLK-RxD)}$	Serial I/O input hold time	100			ns

Note: Bit 6 of address $001A_{16}$ is “1”. Divide this value by four bit 6 of address $001A_{16}$ is “0”.

Table 3.1.16 Switching characteristics (Extended operating temperature version)

(V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, T_a = –40 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_{WH(SCLK)}$	Serial I/O clock output “H” pulse width	Fig. 3.1.1	$t_{C(SCLK)}/2-30$			ns
$t_{WL(SCLK)}$	Serial I/O clock output “L” pulse width		$t_{C(SCLK)}/2-30$			ns
$t_{d(SCLK-TxD)}$	Serial I/O output delay time (Note 1)				140	ns
$t_{v(SCLK-TxD)}$	Serial I/O output valid time (Note 1)		–30			ns
$t_{r(SCLK)}$	Serial I/O clock output rise time				30	ns
$t_{f(SCLK)}$	Serial I/O clock output fall time				30	ns
$t_{r(CMOS)}$	CMOS output rise time (Note 2)			10	30	ns
$t_{f(CMOS)}$	CMOS output fall time (Note 2)			10	30	ns

Note1: When the $P4_5$/TxD P-channel output disable bit of the UART control register (bit 4 of address $001B_{16}$) is “0”.

2: X_{OUT} pin is excluded.

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

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

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
$t_{c(\phi)}$	ϕ clock cycle time	Fig. 3.1.1		$2t_{c(X_{IN})}$		ns
$t_{wH(\phi)}$	ϕ clock "H" pulse width		$t_{c(X_{IN})}$–10			ns
$t_{wL(\phi)}$	ϕ clock "L" pulse width		$t_{c(X_{IN})}$–10			ns
$t_{d(\phi-AH)}$	After ϕ AD_{15}–AD_8 delay time			20	40	ns
$t_{v(\phi-AH)}$	After ϕ AD_{15}–AD_8 valid time		6	10		ns
$t_{d(\phi-AL)}$	After ϕ AD_7–AD_0 delay time			25	45	ns
$t_{v(\phi-AL)}$	After ϕ AD_7–AD_0 valid time		6	10		ns
$t_{d(\phi-SYNC)}$	SYNC delay time			20		ns
$t_{v(\phi-SYNC)}$	SYNC valid time			10		ns
$t_{d(\phi-\overline{WR})}$	$\overline{RD}$ and $\overline{WR}$ delay time			10	20	ns
$t_{v(\phi-\overline{WR})}$	$\overline{RD}$ and $\overline{WR}$ valid time		3	5	10	ns
$t_{d(\phi-DB)}$	After ϕ data bus delay time			20	70	ns
$t_{v(\phi-DB)}$	After ϕ data bus valid time		15			ns
$t_{wL(\overline{RD})}$ $t_{wL(\overline{WR})}$	$\overline{RD}$ pulse width, $\overline{WR}$ pulse width		$t_{c(X_{IN})}$–10			ns
	$\overline{RD}$ pulse width, $\overline{WR}$ pulse width (When one-wait is valid)		$3t_{c(X_{IN})}$–10			ns
$t_{d(AH-\overline{RD})}$ $t_{d(AH-\overline{WR})}$	After AD_{15}–AD_8 $\overline{RD}$ delay time After AD_{15}–AD_8 $\overline{WR}$ delay time		$t_{c(X_{IN})}$–35	$t_{c(X_{IN})}$–15		ns
$t_{d(AL-\overline{RD})}$ $t_{d(AL-\overline{WR})}$	After AD_7–AD_0 $\overline{RD}$ delay time After AD_7–AD_0 $\overline{WR}$ delay time		$t_{c(X_{IN})}$–40	$t_{c(X_{IN})}$–20		ns
$t_{v(\overline{RD}-AH)}$ $t_{v(\overline{WR}-AH)}$	After $\overline{RD}$ AD_{15}–AD_8 valid time After $\overline{WR}$ AD_{15}–AD_8 valid time		0	5		ns
$t_{v(\overline{RD}-AL)}$ $t_{v(\overline{WR}-AL)}$	After $\overline{RD}$ AD_7–AD_0 valid time After $\overline{WR}$ AD_7–AD_0 valid time		0	5		ns
$t_{d(\overline{WR}-DB)}$	After $\overline{WR}$ data bus delay time			15	65	ns
$t_{v(\overline{WR}-DB)}$	After $\overline{WR}$ data bus valid time		10			ns
$t_{d(\overline{RESET}-\overline{RESET_{OUT}})}$	$\overline{RESET_{OUT}}$ output delay time				200	ns
$t_{v(\phi-\overline{RESET})}$	$\overline{RESET_{OUT}}$ output valid time (Note)		0		200	ns

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

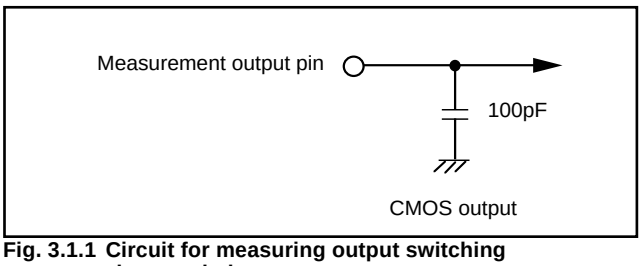


Fig. 3.1.1 Circuit for measuring output switching characteristics

APPENDIX

3.1 Electrical characteristics

3.1.9 Timing diagram

Timing Diagram

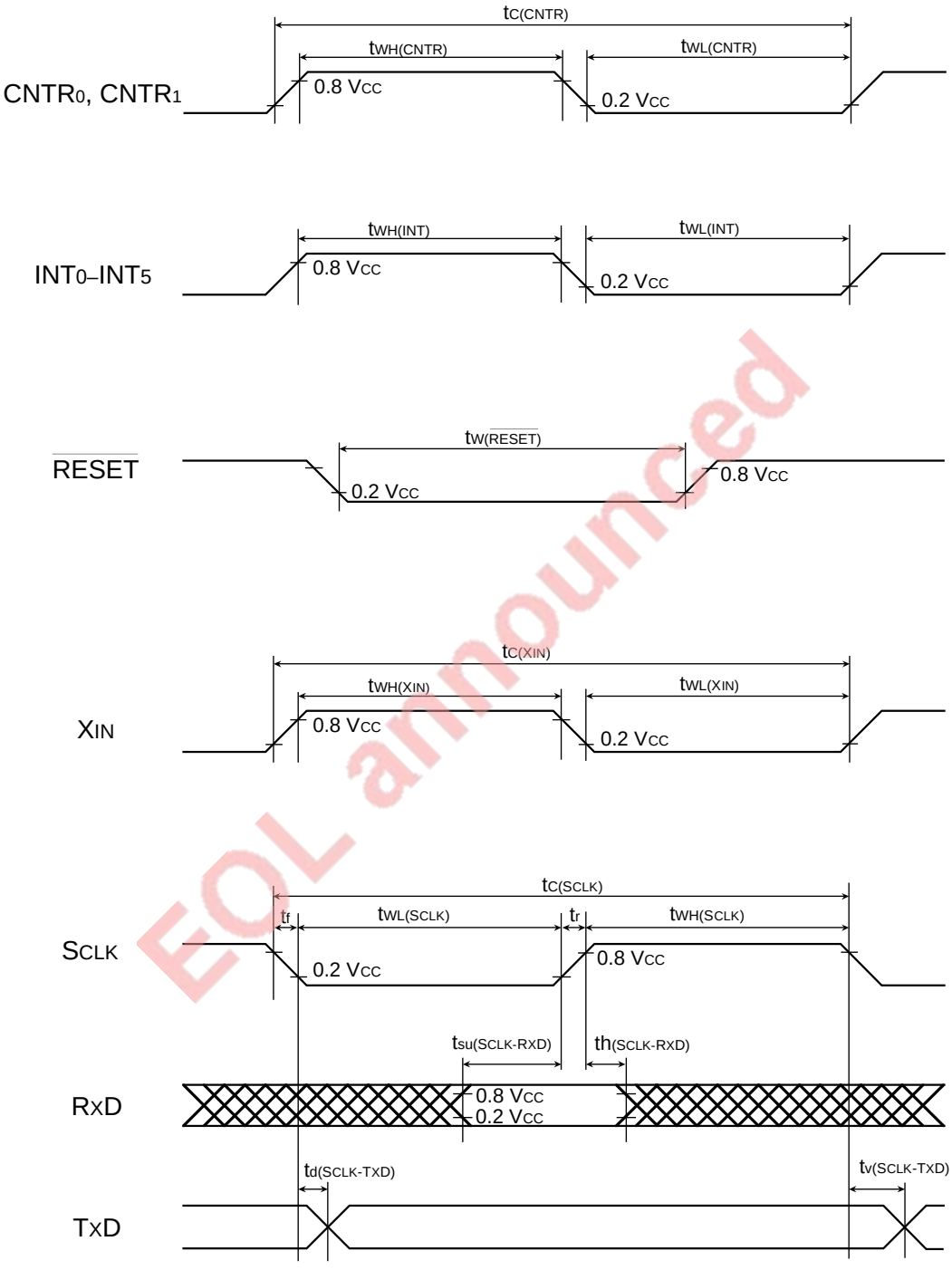
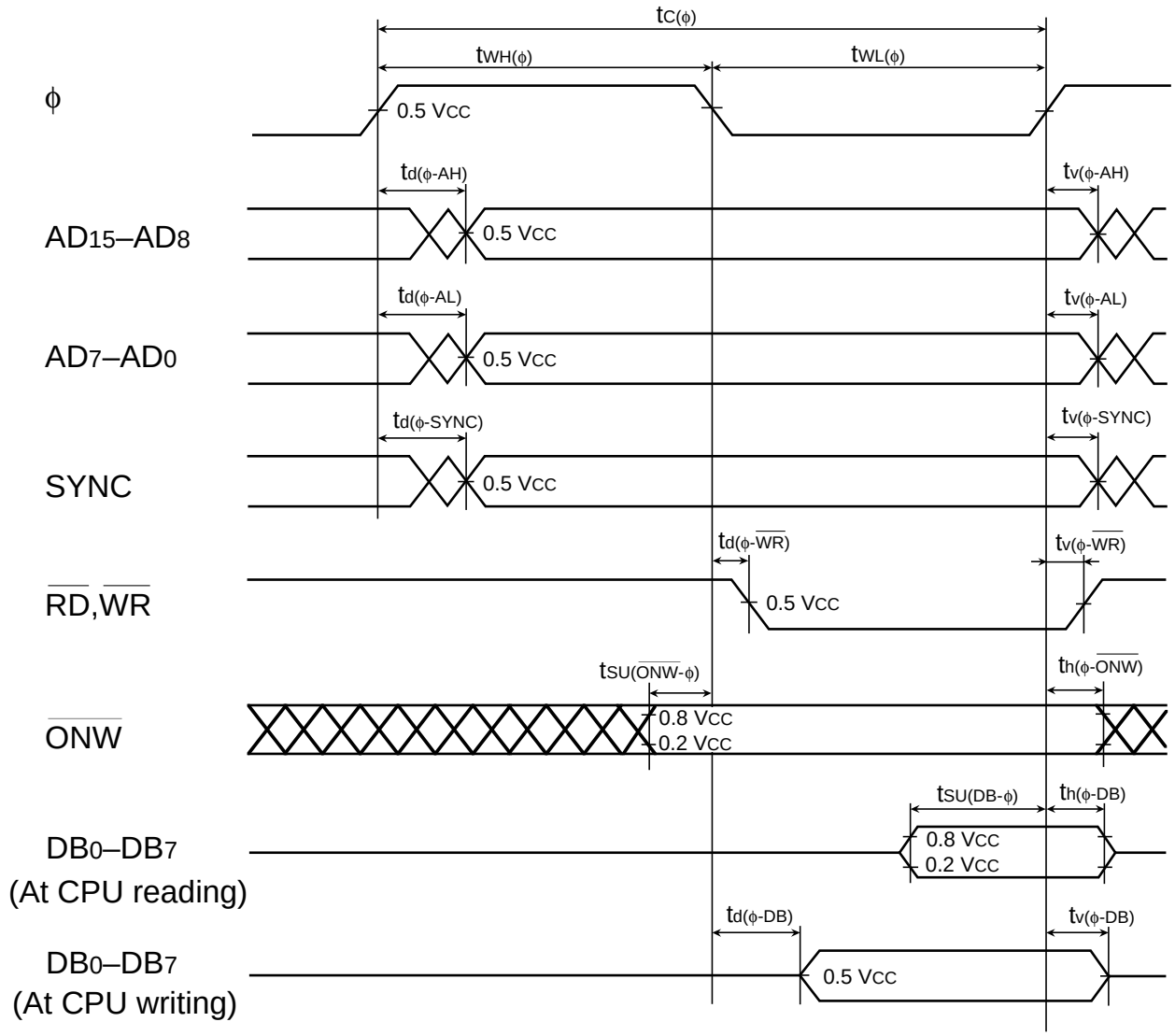


Fig. 3.1.2 Timing diagram (in single-chip mode)

Timing Diagram in Memory Expansion Mode and Microprocessor Mode (1)



Timing Diagram in Microprocessor Mode

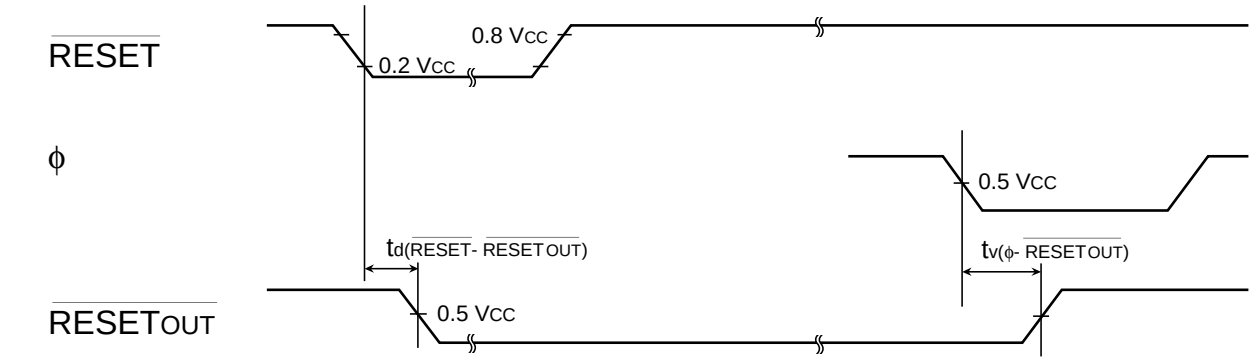


Fig. 3.1.3 Timing diagram (in memory expansion mode and microprocessor mode) (1)

3.1 Electrical characteristics

Timing Diagram in Memory Expansion Mode and Microprocessor Mode (2)

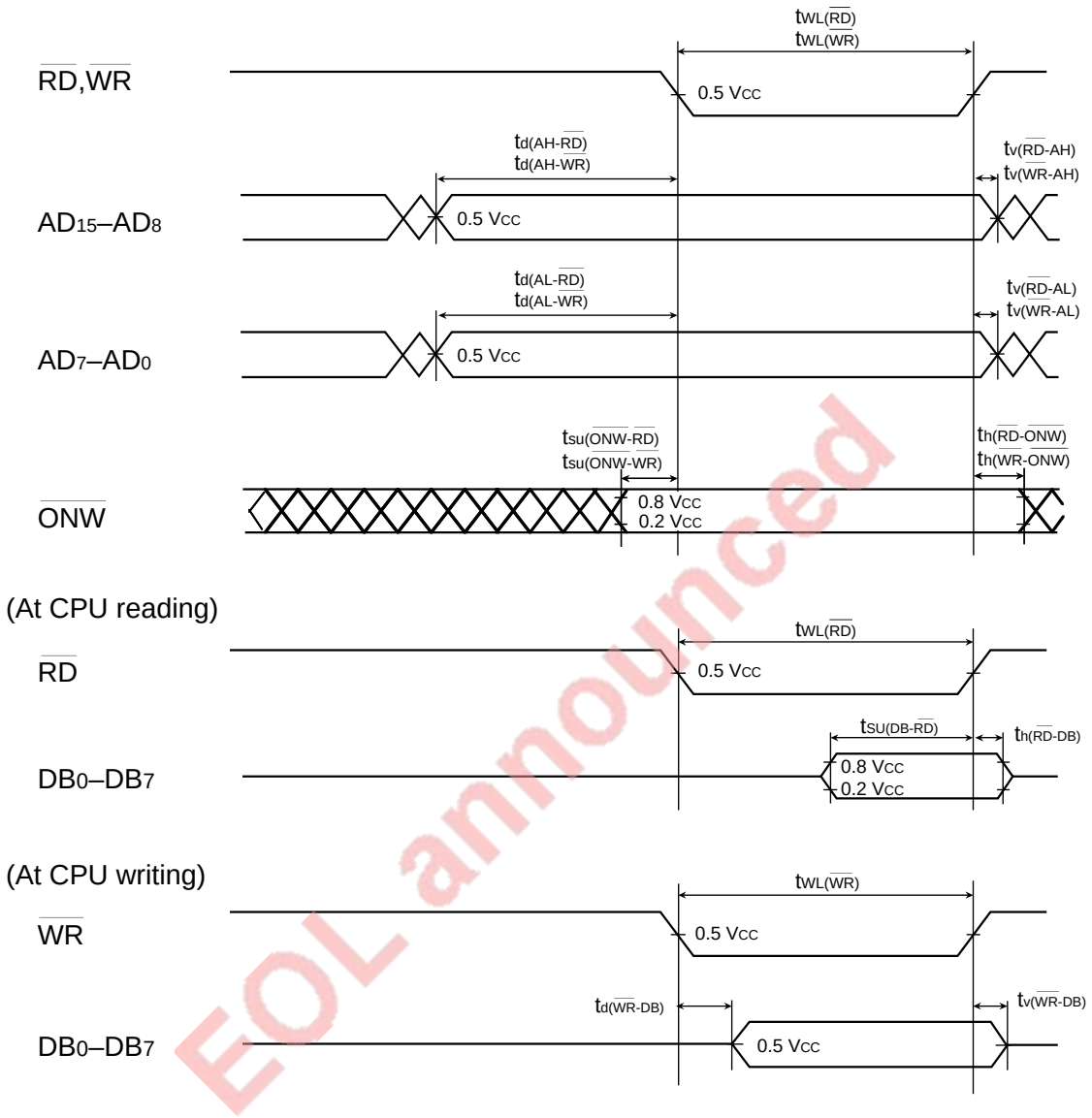


Fig. 3.1.4 Timing diagram (in memory expansion mode and microprocessor mode) (2)

3.2 Standard characteristics

3.2.1 Power source current characteristic examples

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

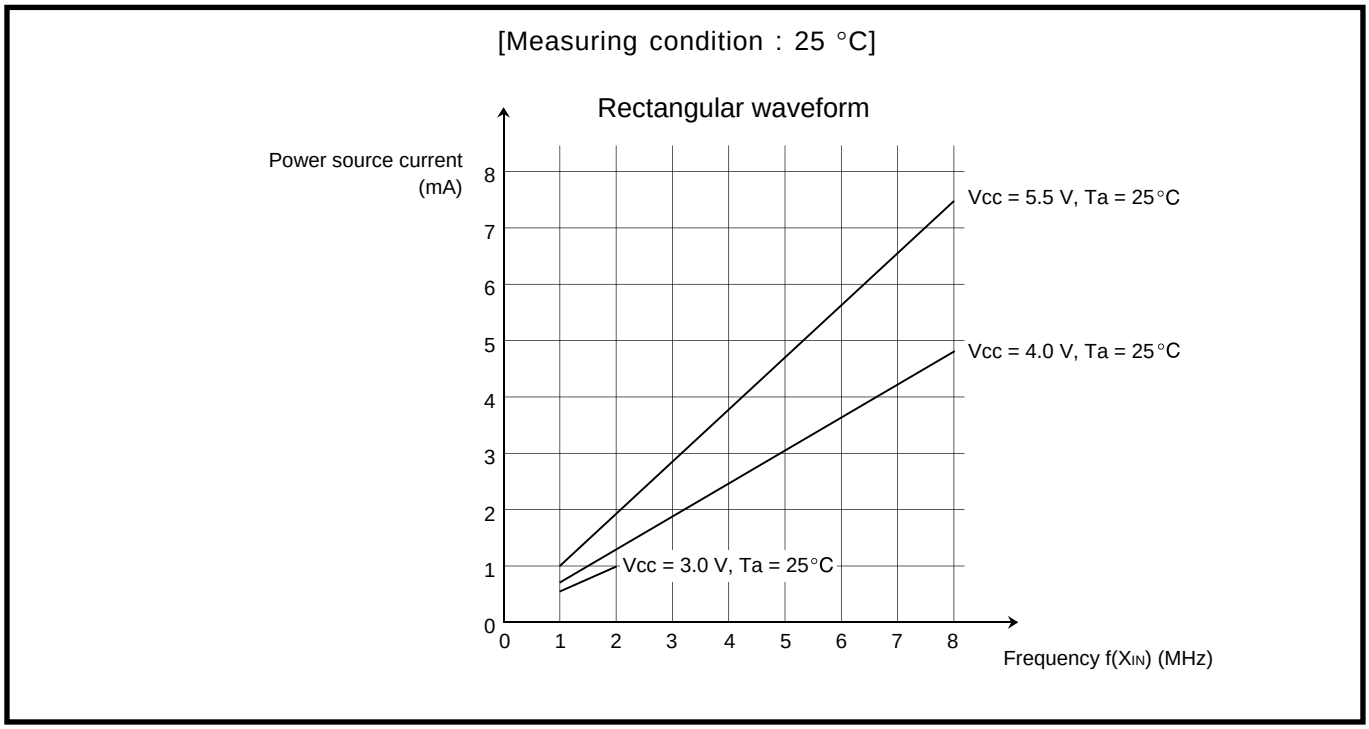


Fig. 3.2.1 Power source current characteristic example

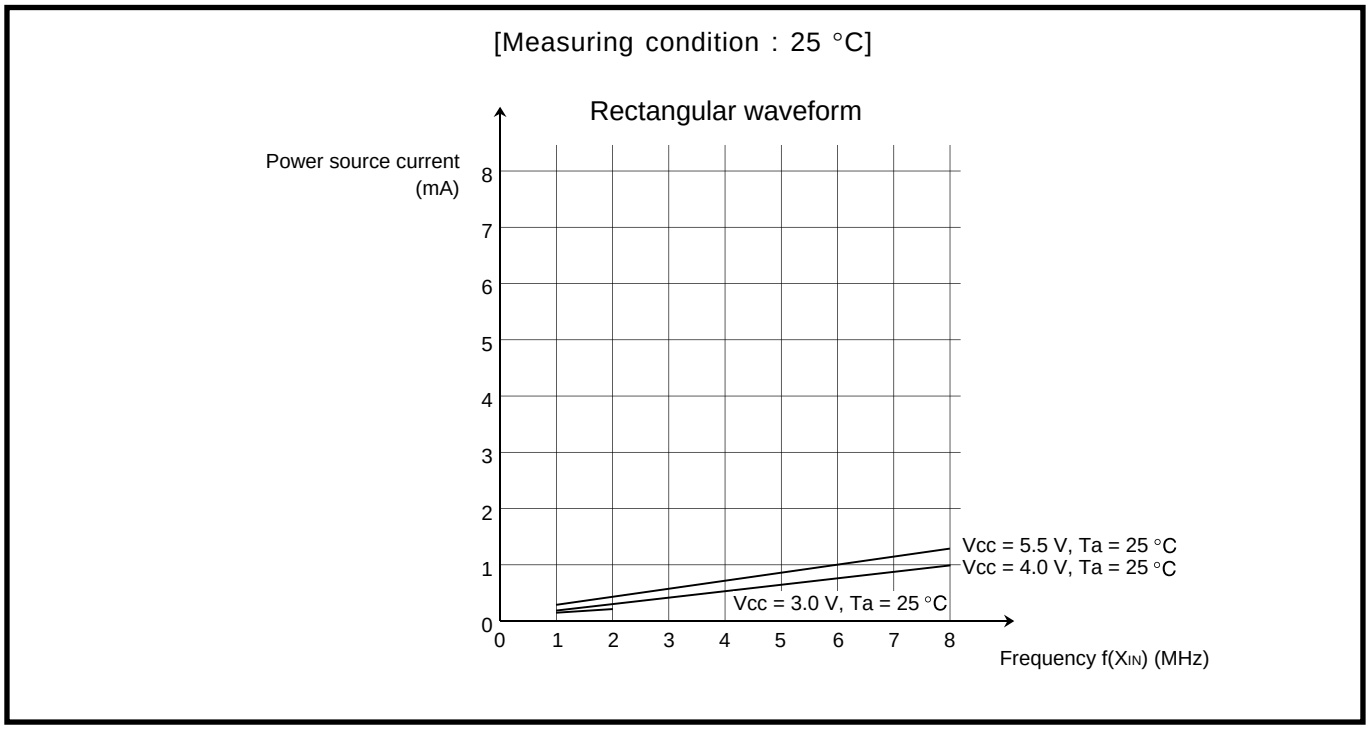


Fig. 3.2.2 Power source current characteristic example (in wait mode)

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.

[Port P0$_0$ I_{OH}–V_{OH} characteristic (P-channel drive)]
(Pins with same characteristic : P0, P1, P2, P3, P4, P5, P6, P7)

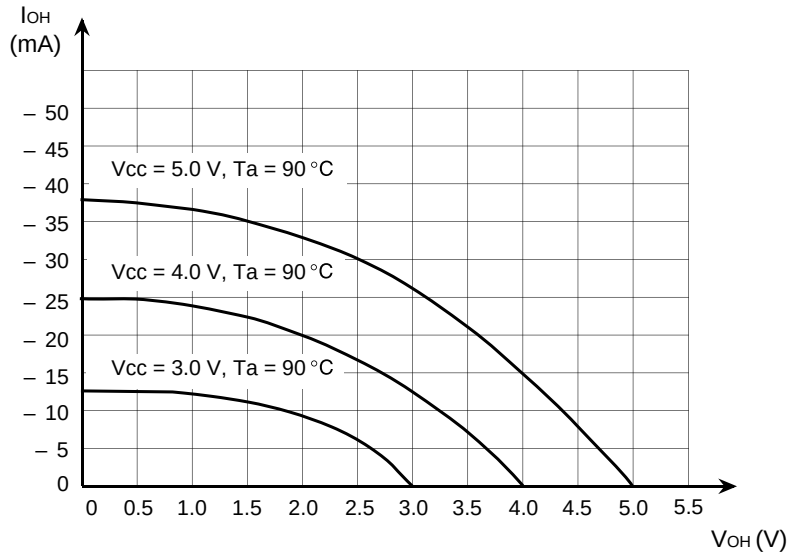


Fig. 3.2.3 Standard characteristic example of CMOS output port at P-channel drive (1)

[Port P0$_0$ I_{OH}–V_{OH} characteristic (P-channel drive)]
(Pins with same characteristic : P0, P1, P2, P3, P4, P5, P6, P7)

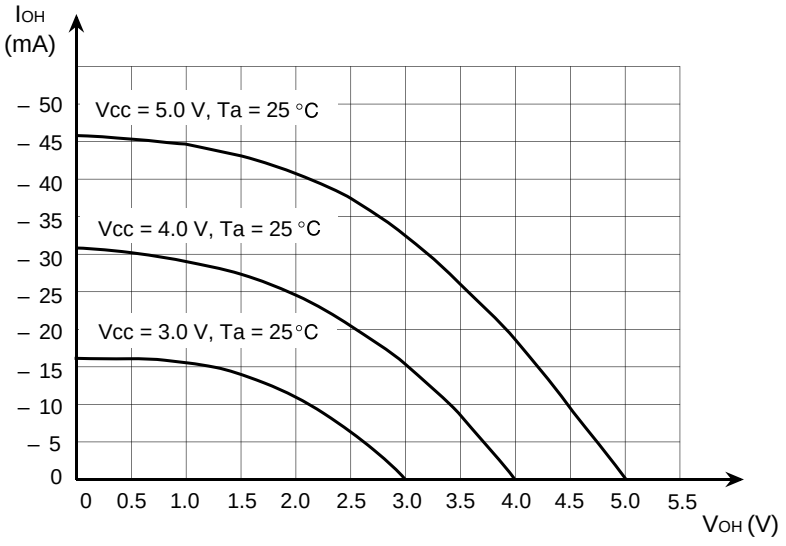


Fig. 3.2.4 Standard characteristic example of CMOS output port at P-channel drive (2)

[Port P00 IOL–VOL characteristic (N-channel drive)]
(Pins with same characteristic : P0, P1, P2, P3, P4, P5, P6, P7)

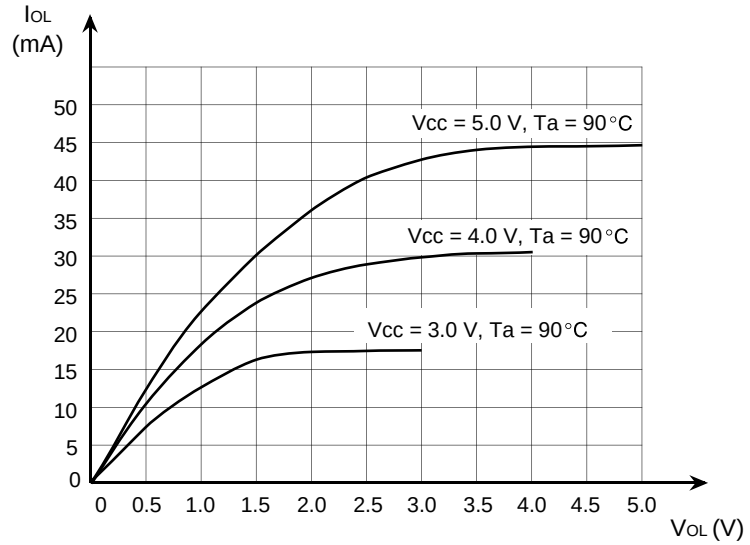


Fig. 3.2.5 Standard characteristic example of CMOS output port at N-channel drive (1)

[Port P00 IOL–VOL characteristic (N-channel drive)]
(Pins with same characteristic : P0, P1, P2, P3, P4, P5, P6, P7)

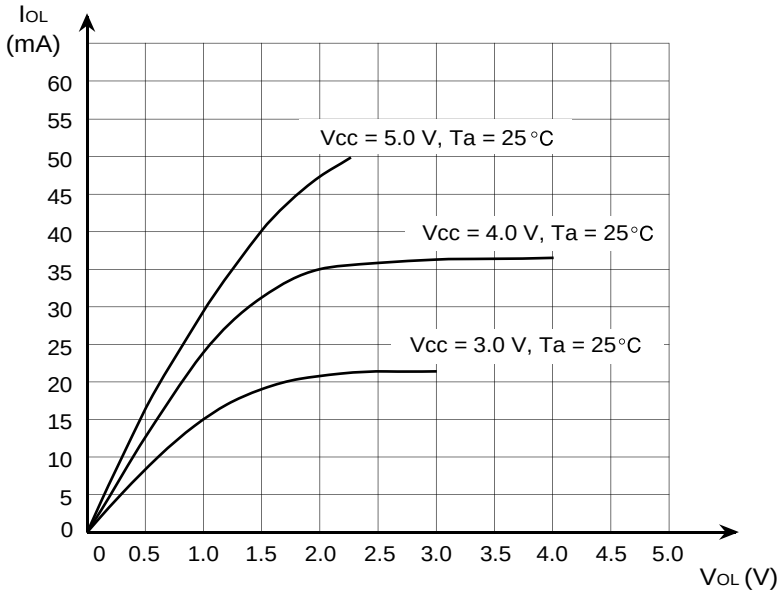


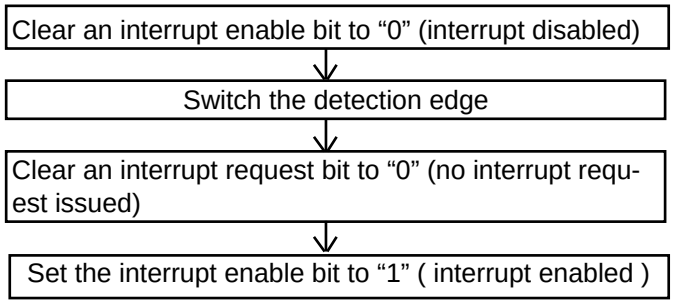
Fig. 3.2.6 Standard characteristic example of CMOS output port at N-channel drive (2)

3.3 Notes on use

3.3.1 Notes on interrupts

(1) Sequence for switching an external interrupt detection edge

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



Reason

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

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

Fix the bits 7 and 6 of the interrupt control register 2 (Address:$003F_{16}$) to “0”.

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

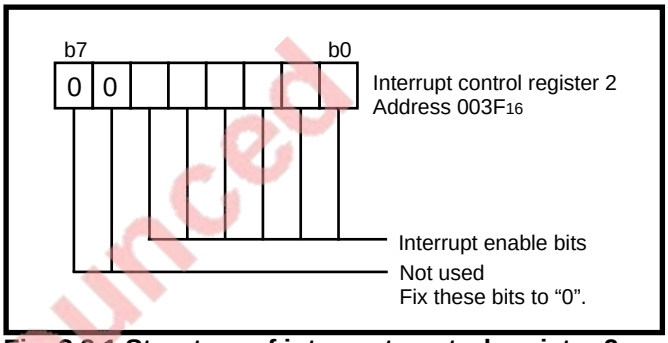


Fig. 3.3.1 Structure of interrupt control register 2

3.3.2 Notes on the serial I/O

(1) Stop of data transmission

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

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

Reason

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

(2) Stop of data reception

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

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

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

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

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

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

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

Reason

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

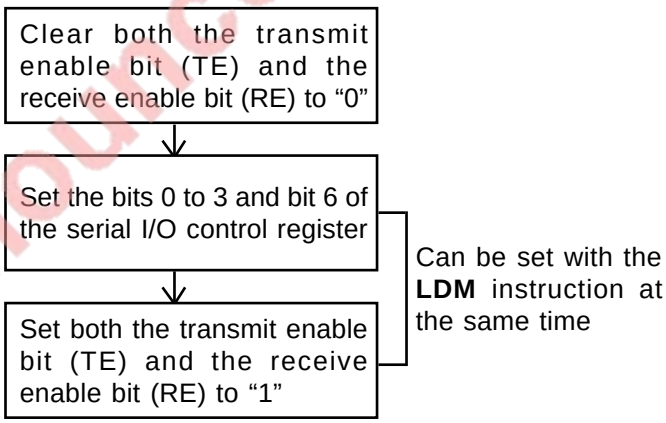
In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to "0" (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/O enable bit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

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

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

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

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



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

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

(7) Control of data transmission using an external clock

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

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

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

●Make the length of the wiring which is connected to a capacitor the shortest possible.

●Make sure to check the operation of application products on the user side.

Reason

If the several nanosecond or several ten nanosecond impulse noise enters the $\overline{\text{RESET}}$ pin, a microcomputer may malfunction.

3.3 Notes on use

3.3.4 Notes on input and output pins

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

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

Pull-up (connect the port to V_{CC}) or pull-down (connect the port to V_{SS}) 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 speccified with a bit managing instruction, its value may be changed in case where content of the pin differs from a content of the port latch.

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

3.3.5 Notes on memory expansion mode and microprocessor mode

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

In the memory expansion or the microprocessor mode, ports $P3_0$ and $P3_1$ can be used as the output port. Use the **LDM** or **STA** instruction for writing data to the port latch (address 0006_{16}) 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_{16}.

Reason

In the memory expansion or microprocessor mode, address 0006_{16} 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_{16}, 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_{16}, 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.

3.3 Notes on use

3.3.6 Notes on built-in PROM

(1) Programming adapter

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

Table 3.3.1 Programming adapter

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

(2) Write and read

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

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

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

Table 3.3.2 Setting of programming adapter switch

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

Table 3.3.3 Setting of PROM programmer address

Microcomputer	PROM programmer start address	PROM programmer completion address
M38002E2SP	Address : 6080_{16} (Note 1)	Address : $7FFD_{16}$ (Note 1)
M38002E2FP		
M38002E4SS	Address : 4080_{16} (Note 2)	Address : $7FFD_{16}$ (Note 2)
M38002E4SP		
M38002E4FS		
M38002E4FP		
M38002E4DSP		
M38002E4DFP		
M38004E8SS	Address : 0080_{16} (Note 3)	Address : $7FFD_{16}$ (Note 3)
M38004E8SP		
M38004E8FS		
M38004E8FP		

Note1 : Addresses $E080_{16}$ to $FFFD_{16}$ in the internal PROM correspond to addresses 6080_{16} to $7FFD_{16}$ in the ROM programmer.

2 : Addresses $C080_{16}$ to $FFFD_{16}$ in the internal PROM correspond to addresses 4080_{16} to $7FFD_{16}$ in the ROM programmer.

3 : Addresses 8080_{16} to $FFFD_{16}$ in the internal PROM correspond to addresses 0080_{16} to $7FFD_{16}$ 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^2 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 V_{SS} 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.

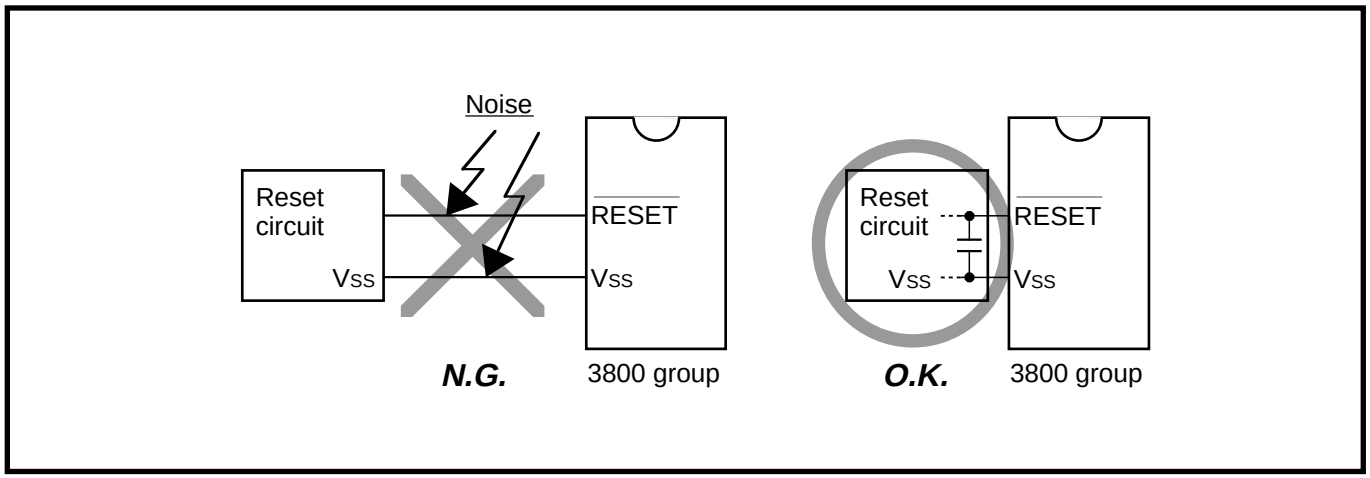


Fig. 3.4.1 Wiring for the $\overline{\text{RESET}}$ pin

(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 oscillatorand the V_{SS} pin of a microcomputer as short as possible.
- Separate the V_{SS} pattern only for oscillation from other V_{SS} 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 V_{SS} level of a microcomputer and the V_{SS} level of an oscillator, the correct clock will not be input in the microcomputer.

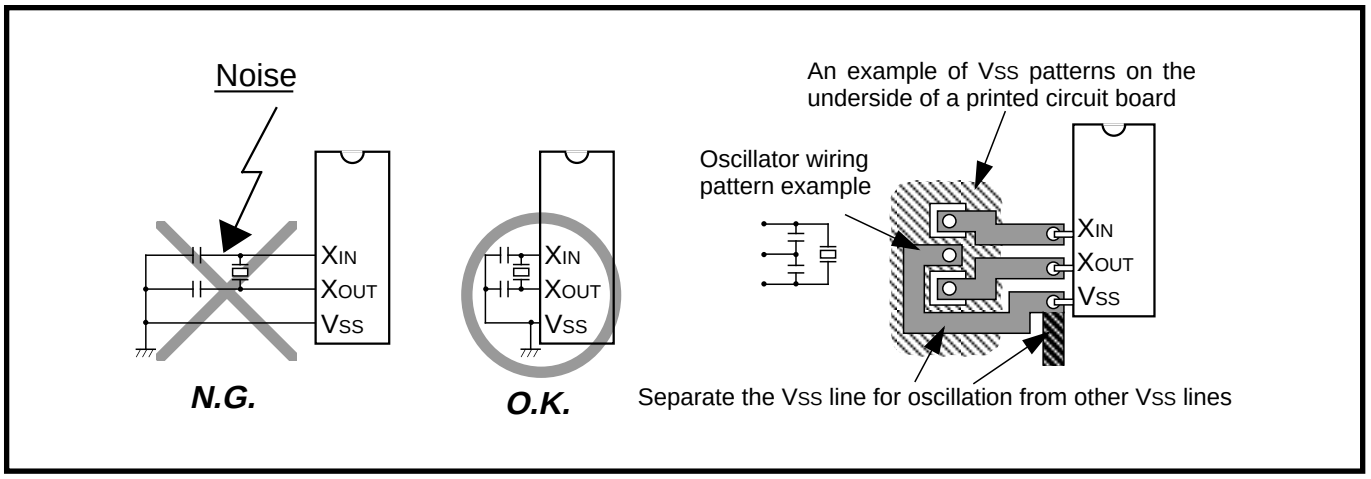


Fig. 3.4.2 Wiring for clock I/O pins

(3) Wiring 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 inclued an approximately 5 kΩ resistor is used in the Mask ROM version, the maicrocomputer operates correctly.

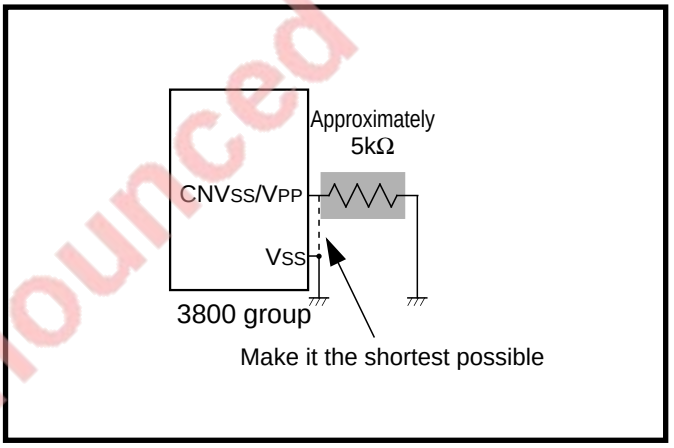


Fig. 3.4.3 Wiring for the VPP pin of the One Time PROM and the EPROM version

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 in struction codes or data are read from the built-in PROM, which may cause a program runaway.

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.

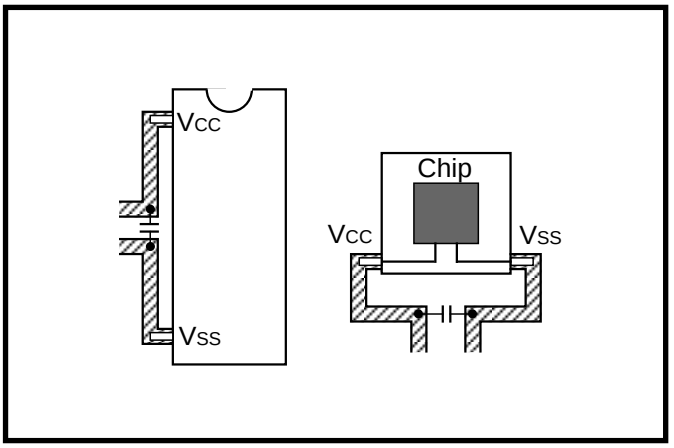


Fig. 3.4.4 Bypass capacitor across the Vss line and the Vcc line

3.4 Countermeasures against noise

3.4.3. Consideration for oscillator

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

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

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

Reason

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

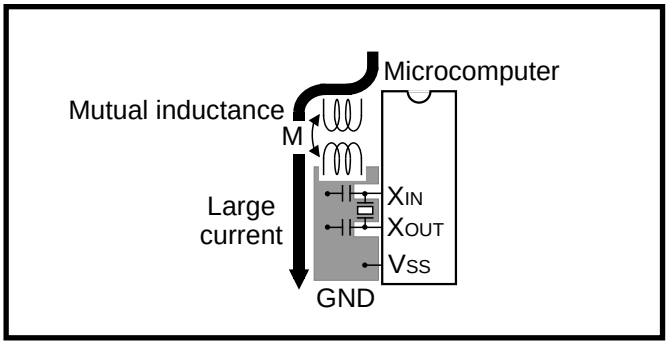


Fig.3.4.5 Wiring for a large current signal line

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

Install an oscillator and a connecting pattern of an osillator 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.

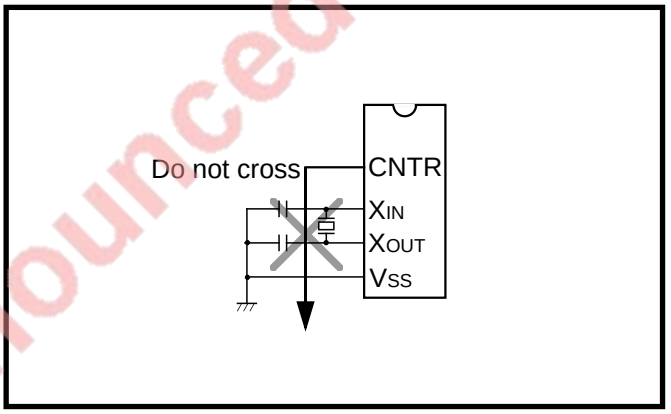


Fig.3.4.6 Wiring to a signal line where potential levels change frequently

3.4.4 Setup for I/O ports

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

<Hardware>

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

<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.
- Rewirte data to direction registers and pull-up control registers (only the product having it) at fixed periods.

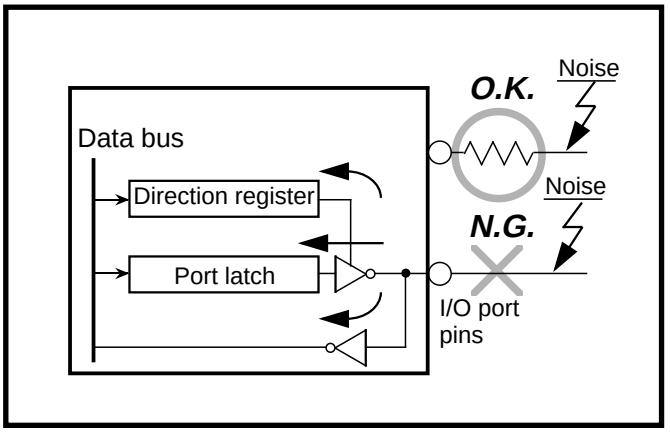


Fig. 3.4.7 Setup for I/O ports

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.

3.4.5 Providing of watchdog timer function by software

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

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

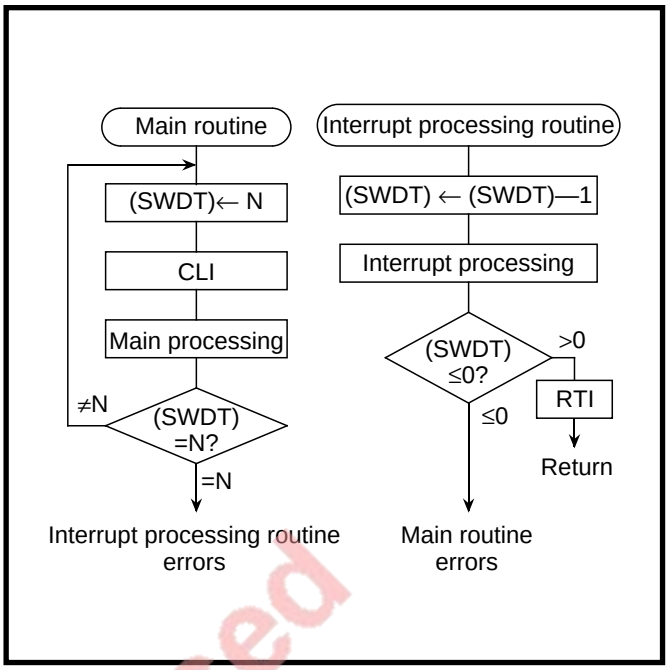


Fig. 3.4.8 Watchdog timer by software

<The main routine>

- Assigns a single byte of RAM to a software watchdog timer (SWDT) and writes the initial value N in the SWDT once at each execution of the main routine. The initial value N should satisfy the following condition:
  N+1 ≥ (Counts of interrupt processing executed in each main routine)

  As the main routine execution cycle may change because of an interrupt processing or others, the initial value N should have a margin.
- Watches the operation of the interrupt processing routine by comparing the SWDT contents with counts of interrupt processing count after the initial value N has been set.
- Detects that the interrupt processing routine has failed and determines to branch to the program initialization routine for recovery processing in the following cases:
  If the SWDT contents do not change after interrupt processing

<The interrupt processing routine>

- Decrements the SWDT contents by 1 at each interrupt processing.
- Determins 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 .

3.5 List of registers

Port Pi

b7	b6	b5	b4	b3	b2	b1	b0

Port Pi (Pi) (i = 0, 1, 2, 3, 4, 5, 6, 7)
[Address : 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{16}, $0A_{16}$, $0C_{16}$, $0E_{16}$]

B	Name	Function	At reset	R	W
0	Port Pi_0	• In output mode Write / Read } Port latch • In input mode Write : Port latch Read : Value of pins (**Note**)	?	○	○
1	Port Pi_1		?	○	○
2	Port Pi_2		?	○	○
3	Port Pi_3		?	○	○
4	Port Pi_4		?	○	○
5	Port Pi_5		?	○	○
6	Port Pi_6		?	○	○
7	Port Pi_7		?	○	○

Note : Port P7 register [Address : $0E_{16}$]
Port P7 is a 2-bit port ($P7_0$, $P7_1$). Accordingly, when bits 2 to 7 are read out, the contents are "0."

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

Port Pi direction register

b7	b6	b5	b4	b3	b2	b1	b0

Port Pi direction register (PiD) (i =0, 1, 2, 3, 4, 5, 6, 7)
[Address : 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{16}, $0B_{16}$, $0D_{16}$, $0F_{16}$]

B	Name	Function	At reset	R	W
0	Port Pi direction register	0 : Port Pi_0 input mode 1 : Port Pi_0 output mode	0	×	○
1		0 : Port Pi_1 input mode 1 : Port Pi_1 output mode	0	×	○
2		0 : Port Pi_2 input mode 1 : Port Pi_2 output mode (**Note**)	0	×	○
3		0 : Port Pi_3 input mode 1 : Port Pi_3 output mode (**Note**)	0	×	○
4		0 : Port Pi_4 input mode 1 : Port Pi_4 output mode (**Note**)	0	×	○
5		0 : Port Pi_5 input mode 1 : Port Pi_5 output mode (**Note**)	0	×	○
6		0 : Port Pi_6 input mode 1 : Port Pi_6 output mode (**Note**)	0	×	○
7		0 : Port Pi_7 input mode 1 : Port Pi_7 output mode (**Note**)	0	×	○

Note : Port P7 direction register [Address : $0F_{16}$]
Port P7 is a 2-bit port ($P7_0$, $P7_1$). Accordingly, these bits do not have a direction register function.

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

Transmit/Receive buffer register

b7 b6 b5 b4 b3 b2 b1 b0

Transmit/Receive buffer register (TB/RB) [Address : 18_{16}]

B	Function	At reset	R	W
0	A transmission data is written to or a receive data is read out from this buffer register. • At writing : a data is written to the Transmit buffer register. • At reading : a content of the Receive buffer register is read out.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Note : A content of the Transmit buffer register cannot be read out.
A data cannot be written to the Receive buffer register.

Fig. 3.5.3 Structure of Transmit/Receive buffer register

Serial I/O status register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O status reigster (SIOSTS) [Address : 19_{16}]

B	Name	Function	At reset	R	W
0	Transmit buffer empty flag (TBE)	0 : Buffer full 1 : Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0 : Buffer empty 1 : Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0 : Transmit shift in progress 1 : Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0 : No error 1 : Overrun error	0	○	×
4	Parity error flag (PE)	0 : No error 1 : Parity error	0	○	×
5	Framing error flag (FE)	0 : No error 1 : Framing error	0	○	×
6	Summing error flag (SE)	0 : (OE) ∪ (PE) ∪ (FE) = 0 1 : (OE) ∪ (PE) ∪ (FE) = 1	0	○	×
7	Nothing is allocated for this bit. It is a write disabled bit. When this bit is read out, the value is “0.”		1	○	×

Fig. 3.5.4 Structure of Serial I/O status register

3.5 List of registers

Serial I/O control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O control register (SIOCON) [Address : $1A_{16}$]

B	Name	Function	At reset	R	W
0	BRG count source selection bit (CSS)	0 : $f(X_{IN})$ 1 : $f(X_{IN})/4$	0	○	○
1	Serial I/O 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	$\overline{S_{RDY}}$ output enable bit (SRDY)	0 : I/O port (P47) 1 : $\overline{S_{RDY}}$ 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/O mode selection bit (SIOM)	0 : UART 1 : Clock synchronous serial I/O	0	○	○
7	Serial I/O enable bit (SIOE)	0 : Serial I/O disabled ($P4_4$–$P4_7$: I/O port) 1 : Serial I/O enabled ($P4_4$–$P4_7$: Serial I/O function pin)	0	○	○

Fig. 3.5.5 Structure of Serial I/O control register

UART control register

b7 b6 b5 b4 b3 b2 b1 b0

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

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	$P4_5$/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	○	×
6			1	○	×
7			1	○	×

Fig. 3.5.6 Structure of UART control register

Baud rate generator

b7 b6 b5 b4 b3 b2 b1 b0

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

B	Function	At reset	R	W
0	A count value of Baud rate generator is set.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 3.5.7 Structure of Baud rate generator

Prescaler 12, Prescaler X, Prescaler Y

b7 b6 b5 b4 b3 b2 b1 b0

Prescaler 12 (PRE12), Prescaler X (PREX), Prescaler Y (PREY)
[Address : 20_{16}, 24_{16}, 26_{16}]

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

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

3.5 List of registers

Timer 1

b7 b6 b5 b4 b3 b2 b1 b0

Timer 1 (T1) [Address : 21_{16}]

B	Function	At reset	R	W
0	● The count value of the Timer 1 is set. ● The value set in this register is written to both the Timer 1 and the Timer 1 latch at the same time. ● When the Timer 1 is read out, the value (count value) of the Timer 1 is read out.	1	○	○
1		0	○	○
2		0	○	○
3		0	○	○
4		0	○	○
5		0	○	○
6		0	○	○
7		0	○	○

Fig. 3.5.9 Structure of Timer 1

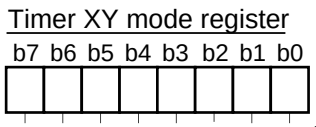
Timer 2, Timer X, Timer Y

b7 b6 b5 b4 b3 b2 b1 b0

Timer 2 (T2), Timer X (TX), Timer Y (TY)
[Address : 22_{16}, 25_{16}, 27_{16}]

B	Function	At reset	R	W
0	● The count value of each timer is set. ● The value set in this register is written to both the Timer and the Timer latch at the same time. ● When the Timer is read out, the value (count value) of the Timer is read out.	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

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



Timer XY mode register (TM) [Address : 23_{16}]

B	Name	Function	At reset	R	W
0	Timer X operating mode bit	b1 b0 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
1			0	○	○
2	$CNTR_0$ active edge switch bit	It depends on the operating mode of the Timer X (refer to Table 3.5.1).	0	○	○
3	Timer X count stop bit	0 : Count start 1 : Count stop	0	○	○
4	Timer Y operating mode bit	b5 b4 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
5			0	○	○
6	$CNTR_1$ active edge switch bit	It depends on the operating mode of the Timer Y (refer to Table 3.5.1).	0	○	○
7	Timer Y count stop bit	0 : Count start 1 : Count stop	0	○	○

Fig. 3.5.11 Structure of Timer XY mode register

Table. 3.5.1 Function of $CNTR_0$/$CNTR_1$ edge switch bit

Operating mode of Timer X/Timer Y	Function of $CNTR_0$/$CNTR_1$ edge switch bit (bits 2 and 6)	
Timer mode	"0"	• Generation of $CNTR_0$/$CNTR_1$ interrupt request : Falling edge (No effect on timer count)
	"1"	• Generation of $CNTR_0$/$CNTR_1$ interrupt request : Rising edge (No effect on timer count)
Pulse output mode	"0"	• Start of pulse output : From "H" level • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Falling edge
	"1"	• Start of pulse output : From "L" level • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Rising edge
Event counter mode	"0"	• Timer X/Timer Y : Count of rising edge • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Falling edge
	"1"	• Timer X/Timer Y : Count of falling edge • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Rising edge
Pulse width measurement mode	"0"	• Timer X/Timer Y : Measurement of "H" level width • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Falling edge
	"1"	• Timer X/Timer Y : Measurement of "L" level width • Generation of $CNTR_0$/$CNTR_1$ interrupt request : Rising edge

3.5 List of registers

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0

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

B	Name	Function	At reset	R	W
0	INT_0 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT_1 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
2	INT_2 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
3	INT_3 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
4	INT_4 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
5	INT_5 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
6	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are “0.”		0	○	×
7			0	○	×

Fig. 3.5.12 Structure of Interrupt edge selection register

CPU mode register

b7 b6 b5 b4 b3 b2 b1 b0

CPU mode register (CPUM) [Adress : $3B_{16}$]

B	Name	Function	At reset	R	W
0	Processor mode bits	00 : Single-chip mode 01 : Memory expansion mode 10 : Microprocessor mode 11 : Not available	0	○	○
1			*	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	×
3	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are “0.”		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

* An initial value of bit 1 is determined by a level of the CNV_{SS} pin.

Fig. 3.5.13 Structure of CPU mode register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request reigster 1 (IREQ1) [Address : $3C_{16}$]

B	Name	Function	At reset	R	W
0	INT_0 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
1	INT_1 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
2	Serial I/O receive interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
3	Serial I/O transmit interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
4	Timer X interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
5	Timer Y interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
6	Timer 1 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
7	Timer 2 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*

* "0" is set by software, but not "1."

Fig. 3.5.14 Structure of Interrupt request register 1

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request reigster 2 (IREQ2) [Address : $3D_{16}$]

B	Name	Function	At reset	R	W
0	$CNTR_0$ interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
1	$CNTR_1$ interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
2	INT_2 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
3	INT_3 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
4	INT_4 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
5	INT_5 interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	*
6	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0."		0	○	X
7			0	○	X

* "0" is set by software, but not "1."

Fig. 3.5.15 Structure of Interrupt request register 2

3.5 List of registers

Interrupt control register 1

b7	b6	b5	b4	b3	b2	b1	b0

Interrupt control register 1 (ICON1) [Address : $3E_{16}$]

B	Name	Function	At reset	R	W
0	INT_0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	INT_1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer Y interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 3.5.16 Structure of Interrupt control register 1

Interrupt control register 2

b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Interrupt control reigster 2 (ICON2) [Address : $3F_{16}$]

B	Name	Function	At reset	R	W
0	$CNTR_0$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	$CNTR_1$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT_2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	INT_3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT_4 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	INT_5 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Fix these bits to “0.”		0	○	○
7			0	○	○

Fig. 3.5.17 Structure of Interrupt control register 2

3.6 Mask ROM ordering method

GZZ-SH04-34B<13B0>

Mask ROM number	

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

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

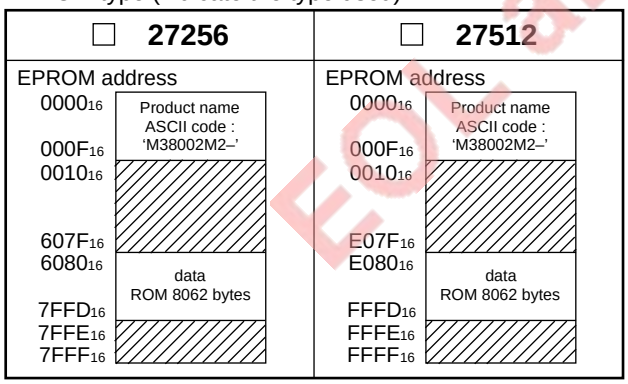
✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M38002M2-XXXSP ☐ M38002M2-XXXFP

Checksum code for entire EPROM | | | | | (hexadecimal notation)

EPROM type (indicate the type used)



In the address space of the microcomputer, the internal ROM area is from address $E080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".
(2) The ASCII codes of the product name "M38002M2–" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' – ' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'8' = 38_{16}	$000A_{16}$	FF_{16}
0003_{16}	'0' = 30_{16}	$000B_{16}$	FF_{16}
0004_{16}	'0' = 30_{16}	$000C_{16}$	FF_{16}
0005_{16}	'2' = 32_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'2' = 32_{16}	$000F_{16}$	FF_{16}

(1/2)

3.6 Mask ROM ordering method

GZZ-SH04-34B<13B0>

Mask ROM number	

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

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	*=△$8000 .BYTE△ 'M38002M2–'	*=△$0000 .BYTE△ 'M38002M2–'

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

✵ 2. Mark specification

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

✵ 3. Usage conditions

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

(1) How will you use the X_{IN}-X_{OUT} oscillator?

☐ Ceramic resonator ☐ Quartz crystal

☐ External clock input ☐ Other ( )

At what frequency? $f(X_{IN})$ = ☐ MHz

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

☐ Single-chip mode ☐ Memory expansion mode

☐ Microprocessor mode

✵ 4. Comments

(2/2)

GZZ-SH04-79B<16A0>

Mask ROM number	

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

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M38002M2DXXXSP ☐ M38002M2DXXXFP

Checksum code for entire EPROM | | | | | (hexadecimal notation)

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Product name ASCII code : 'M38002M2D'	0000_{16} Product name ASCII code : 'M38002M2D'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$607F_{16}$	$E07F_{16}$
6080_{16} data ROM 8062 bytes	$E080_{16}$ data ROM 8062 bytes
$7FFD_{16}$	$FFFD_{16}$
$7FFE_{16}$	$FFFE_{16}$
$7FFF_{16}$	$FFFF_{16}$

In the address space of the microcomputer, the internal ROM area is from address $E080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".

(2) The ASCII codes of the product name "M38002M2D" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	'D' = 44_{16}
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'8' = 38_{16}	$000A_{16}$	FF_{16}
0003_{16}	'0' = 30_{16}	$000B_{16}$	FF_{16}
0004_{16}	'0' = 30_{16}	$000C_{16}$	FF_{16}
0005_{16}	'2' = 32_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'2' = 32_{16}	$000F_{16}$	FF_{16}

(1/2)

APPENDIX

3.6 Mask ROM ordering method

GZZ-SH04-79B<16A0>

Mask ROM number	

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

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

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

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

✵ 2. Mark specification

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

✵ 3. Usage conditions

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

(1) How will you use the X_{IN}-X_{OUT} oscillator?

☐ Ceramic resonator ☐ Quartz crystal

☐ External clock input ☐ Other ( )

At what frequency? $f(X_{IN})$ = ☐ MHz

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

☐ Single-chip mode ☐ Memory expansion mode

☐ Microprocessor mode

✵ 4. Comments

(2/2)

GZZ-SH03-22B<9YB0>

Mask ROM number	

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

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M38002M4-XXXSP ☐ M38002M4-XXXFP

Checksum code for entire EPROM [ ][ ][ ][ ] (hexadecimal notation)

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Product name ASCII code : 'M38002M4–'	0000_{16} Product name ASCII code : 'M38002M4–'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$407F_{16}$	$C07F_{16}$
4080_{16} data ROM 16254 bytes	$C080_{16}$ data ROM 16254 bytes
$7FFD_{16}$	$FFFD_{16}$
$7FFE_{16}$	$FFFE_{16}$
$7FFF_{16}$	$FFFF_{16}$

In the address space of the microcomputer, the internal ROM area is from address $C080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".
(2) The ASCII codes of the product name "M38002M4–" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' – ' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'8' = 38_{16}	$000A_{16}$	FF_{16}
0003_{16}	'0' = 30_{16}	$000B_{16}$	FF_{16}
0004_{16}	'0' = 30_{16}	$000C_{16}$	FF_{16}
0005_{16}	'2' = 32_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'4' = 34_{16}	$000F_{16}$	FF_{16}

(1/2)

3.6 Mask ROM ordering method

GZZ-SH03-22B<9YB0>

Mask ROM number	

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

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

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

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

✵ 2. Mark specification

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

✵ 3. Usage conditions

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

(1) How will you use the X_{IN}-X_{OUT} oscillator?

☐ Ceramic resonator ☐ Quartz crystal

☐ External clock input ☐ Other ( )

At what frequency? $f(X_{IN})$ = ☐ MHz

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

☐ Single-chip mode ☐ Memory expansion mode

☐ Microprocessor mode

✵ 4. Comments

(2/2)

GZZ-SH05-12B<21A0>

Mask ROM number	

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

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M38002M4DXXXSP ☐ M38002M4DXXXFP

Checksum code for entire EPROM | | | | | (hexadecimal notation)

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Product name ASCII code : 'M38002M4D'	0000_{16} Product name ASCII code : 'M38002M4D'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$407F_{16}$	$C07F_{16}$
4080_{16} data ROM 16254 bytes	$C080_{16}$ data ROM 16254 bytes
$7FFD_{16}$	$FFFD_{16}$
$7FFE_{16}$	$FFFE_{16}$
$7FFF_{16}$	$FFFF_{16}$

In the address space of the microcomputer, the internal ROM area is from address $C080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".

(2) The ASCII codes of the product name "M38002M4D" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	'D' = 44_{16}
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'8' = 38_{16}	$000A_{16}$	FF_{16}
0003_{16}	'0' = 30_{16}	$000B_{16}$	FF_{16}
0004_{16}	'0' = 30_{16}	$000C_{16}$	FF_{16}
0005_{16}	'2' = 32_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'4' = 34_{16}	$000F_{16}$	FF_{16}

(1/2)

APPENDIX

3.6 Mask ROM ordering method

GZZ-SH05-12B<21A0>

Mask ROM number	

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

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

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

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

✵ 2. Mark specification

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

✵ 3. Usage conditions

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

(1) How will you use the X_{IN}-X_{OUT} oscillator?

☐ Ceramic resonator ☐ Quartz crystal

☐ External clock input ☐ Other ( )

At what frequency? $f(X_{IN})$ = ☐ MHz

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

☐ Single-chip mode ☐ Memory expansion mode

☐ Microprocessor mode

✵ 4. Comments

(2/2)

GZZ-SH04-62B<14B0>

Mask ROM number	

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

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

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

Checksum code for entire EPROM | | | | | (hexadecimal notation)

EPROM type (indicate the type used)

☐ 27256	☐ 27512
EPROM address	EPROM address
0000_{16} Product name ASCII code : 'M38003M6–'	0000_{16} Product name ASCII code : 'M38003M6–'
$000F_{16}$	$000F_{16}$
0010_{16}	0010_{16}
$207F_{16}$	$A07F_{16}$
2080_{16} data ROM 24446 bytes	$A080_{16}$ data ROM 24446 bytes
$7FFD_{16}$	$FFFD_{16}$
$7FFE_{16}$	$FFFE_{16}$
$7FFF_{16}$	$FFFF_{16}$

In the address space of the microcomputer, the internal ROM area is from address $A080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".
(2) The ASCII codes of the product name "M38003M6–" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' – ' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'8' = 38_{16}	$000A_{16}$	FF_{16}
0003_{16}	'0' = 30_{16}	$000B_{16}$	FF_{16}
0004_{16}	'0' = 30_{16}	$000C_{16}$	FF_{16}
0005_{16}	'3' = 33_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'6' = 36_{16}	$000F_{16}$	FF_{16}

(1/2)

APPENDIX

3.6 Mask ROM ordering method

GZZ-SH04-62B<14B0>

Mask ROM number	

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

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	*=△$8000 .BYTE△ 'M38003M6–'	*=△$0000 .BYTE△ 'M38003M6–'

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

✵ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (64P4B for M38003M6-XXXSP, 64P6N for M38003M6-XXXFP) and attach it to the mask ROM confirmation form.

M38003M6-XXXHP is specified to the standard mark.

✵ 3. Usage conditions

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

(1) How will you use the X_{IN}-X_{OUT} oscillator?

☐ Ceramic resonator ☐ Quartz crystal

☐ External clock input ☐ Other ( )

At what frequency? $f(X_{IN})$ = ☐ MHz

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

☐ Single-chip mode ☐ Memory expansion mode

☐ Microprocessor mode

✵ 4. Comments

GZZ-SH04-30B<13B0>

Mask ROM number	

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

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

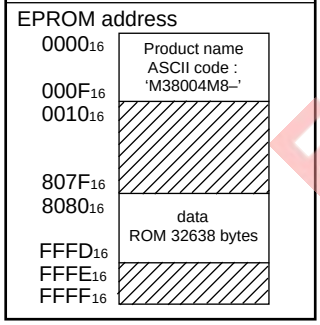
Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M38004M8-XXXSP ☐ M38004M8-XXXFP

Checksum code for entire EPROM | | | | | (hexadecimal notation)

EPROM type (indicate the type used)

☐ **27512**



In the address space of the microcomputer, the internal ROM area is from address 8080_{16} to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".
(2) The ASCII codes of the product name "M38004M8–" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	' – ' = $2D_{16}$
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'8' = 38_{16}	$000A_{16}$	FF_{16}
0003_{16}	'0' = 30_{16}	$000B_{16}$	FF_{16}
0004_{16}	'0' = 30_{16}	$000C_{16}$	FF_{16}
0005_{16}	'4' = 34_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'8' = 38_{16}	$000F_{16}$	FF_{16}

(1/2)

3.6 Mask ROM ordering method

GZZ-SH04-30B<13B0>

Mask ROM number	

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

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

EPROM type	27512
The pseudo-command	*=△$0000 .BYTE△ 'M38004M8–'

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

✵ 2. Mark specification

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

✵ 3. Usage conditions

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

(1) How will you use the X_{IN}-X_{OUT} oscillator?

☐ Ceramic resonator ☐ Quartz crystal

☐ External clock input ☐ Other ( )

At what frequency? $f(X_{IN})$ = ☐ MHz

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

☐ Single-chip mode ☐ Memory expansion mode

☐ Microprocessor mode

✵ 4. Comments

(2/2)

GZZ-SH07-23B<33A0>

Mask ROM number	

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

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ✵.

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

✵ 1. Confirmation

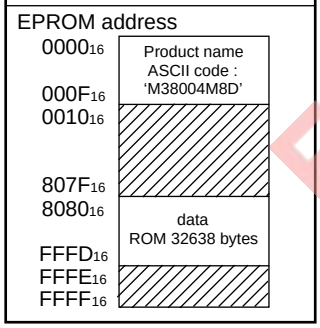
Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Microcomputer name : ☐ M38004M8DXXXSP ☐ M38004M8DXXXFP

Checksum code for entire EPROM | | | | | (hexadecimal notation)

EPROM type (indicate the type used)

☐ **27512**



In the address space of the microcomputer, the internal ROM area is from address 8080_{16} to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "FF_{16}".
(2) The ASCII codes of the product name "M38004M8D" must be entered in addresses 0000_{16} to 0008_{16}. And set the data "FF_{16}" in addresses 0009_{16} to $000F_{16}$. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address	
0000_{16}	'M' = $4D_{16}$	0008_{16}	'D' = 44_{16}
0001_{16}	'3' = 33_{16}	0009_{16}	FF_{16}
0002_{16}	'8' = 38_{16}	$000A_{16}$	FF_{16}
0003_{16}	'0' = 30_{16}	$000B_{16}$	FF_{16}
0004_{16}	'0' = 30_{16}	$000C_{16}$	FF_{16}
0005_{16}	'4' = 34_{16}	$000D_{16}$	FF_{16}
0006_{16}	'M' = $4D_{16}$	$000E_{16}$	FF_{16}
0007_{16}	'8' = 38_{16}	$000F_{16}$	FF_{16}

(1/2)

GZZ-SH07-23B<33A0>

Mask ROM number	

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

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

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

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

✵ 2. Mark specification

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

✵ 3. Usage conditions

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

(1) How will you use the X_{IN}-X_{OUT} oscillator?

☐ Ceramic resonator ☐ Quartz crystal

☐ External clock input ☐ Other ( )

At what frequency? $f(X_{IN})$ = ☐ MHz

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

☐ Single-chip mode ☐ Memory expansion mode

☐ Microprocessor mode

✵ 4. Comments

(2/2)

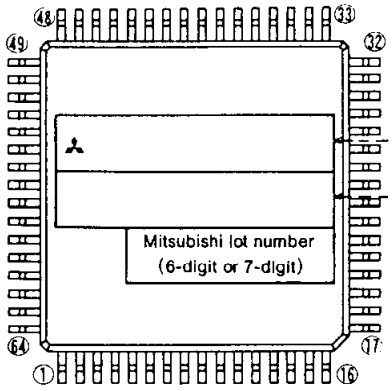
3.7 Mark specification form

64P6N (64-PIN QFP) MARK SPECIFICATION FORM

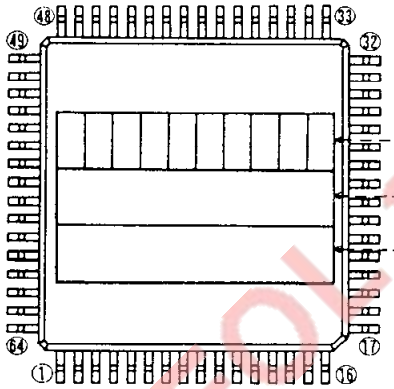
Mitsubishi IC catalog name

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

A. Standard Mitsubishi Mark



B. Customer's Parts Number + Mitsubishi Catalog Name



Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type. (The character size became smaller than A (standard Mitsubishi mark) type)

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

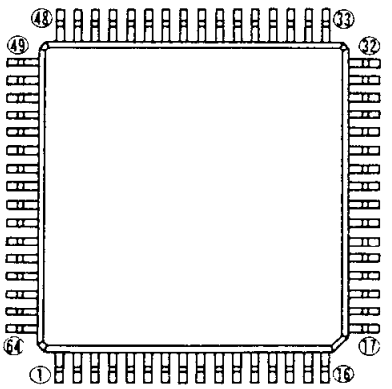
Note3 : Customer's parts number can be up to 10 characters :
Only 0~9, A~Z, +, −, /, (,), &, ©, . (period), and , (comma) are usable.

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

Mitsubishi logo is not required ☐

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

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 ☐

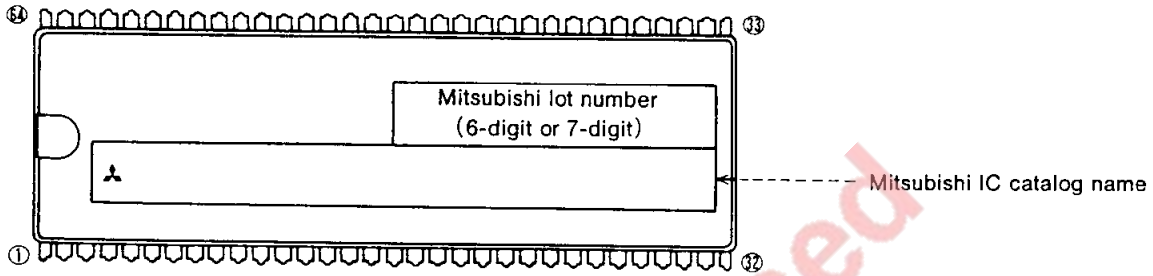
The standard Mitsubishi font is used for all characters except for a logo.

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

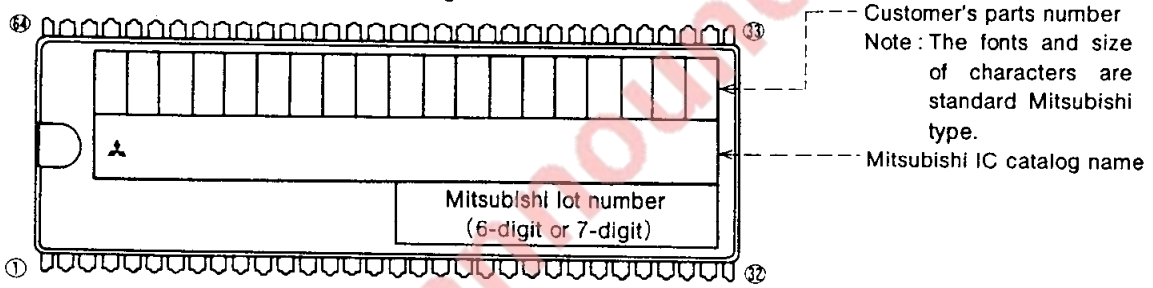
Mitsubishi IC catalog name ☐

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

A. Standard Mitsubishi Mark

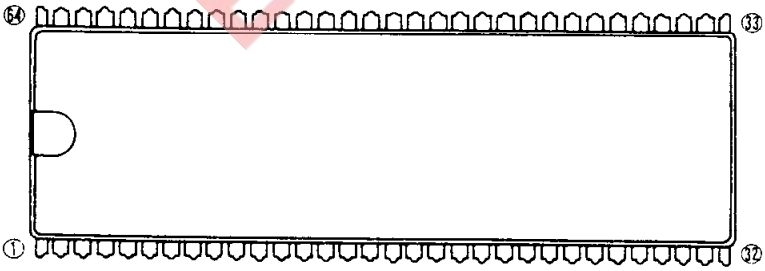


B. Customer's Parts Number + Mitsubishi Catalog Name



Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 19 characters :
Only 0~9, A~Z, +, −, /, (,), &, ©, . (period), and , (comma) are usable.
4 : If the Mitsubishi logo ⅄ is not required, check the box on the right. ⅄ Mitsubishi logo is not required ☐

C. Special Mark Required



Note1 : If the special mark is to be printed, indicate the desired layout of the mark in the upper figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit or 7-digit) and mask ROM number (3-digit) are always marked.
2 : If the customer's trade mark logo must be used in the special mark, check the box below. Please submit a clean original of the logo.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.
Special logo required ☐
The standard Mitsubishi font is used for all characters except for a logo.

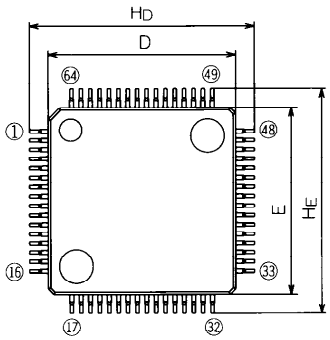
3.8 Package outline

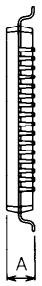
64P6D–A

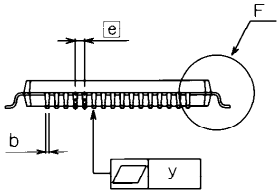
Plastic 64pin 10×10mm body LQFP

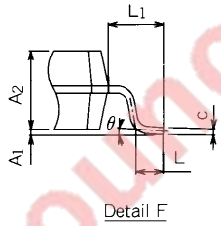
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP64-P-1010-0.50	—	0.32	Alloy 42

Scale : 2.5/1

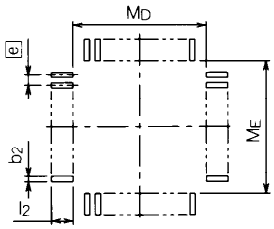








Detail F



Recommended Mount Pad

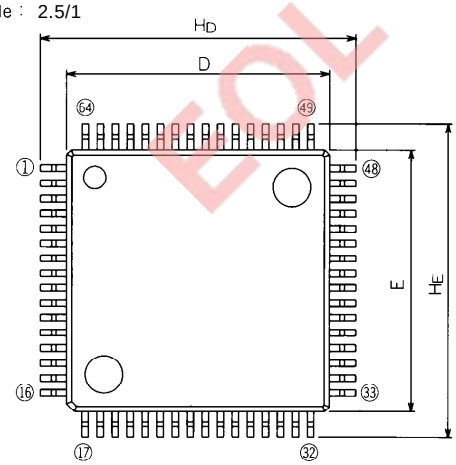
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	1.7
A_1	0	0.1	0.2
A_2	—	1.4	—
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	9.9	10.0	10.1
E	9.9	10.0	10.1
e	—	0.5	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
L	0.3	0.5	0.7
L_1	—	1.0	—
y	—	—	0.1
θ	0°	—	10°
b_2	—	0.225	—
l_2	1.0	—	—
M_D	—	10.4	—
M_E	—	10.4	—

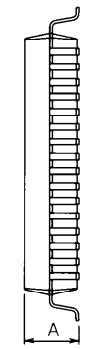
64P6N–A

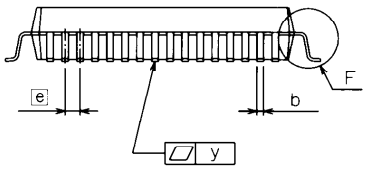
Plastic 64pin 14×14mm body QFP

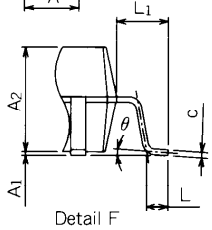
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP64-P-1414-0.80	—	1.11	Alloy 42

Scale : 2.5/1

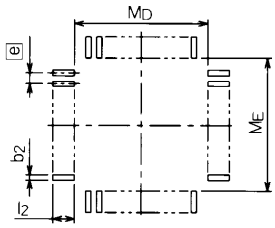








Detail F



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	3.05
A_1	0	0.1	0.2
A_2	—	2.8	—
b	0.3	0.35	0.45
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	13.8	14.0	14.2
e	—	0.8	—
H_D	16.5	16.8	17.1
H_E	16.5	16.8	17.1
L	0.4	0.6	0.8
L_1	—	1.4	—
y	—	—	0.1
θ	0°	—	10°
b_2	—	0.5	—
l_2	1.3	—	—
M_D	—	14.6	—
M_E	—	14.6	—

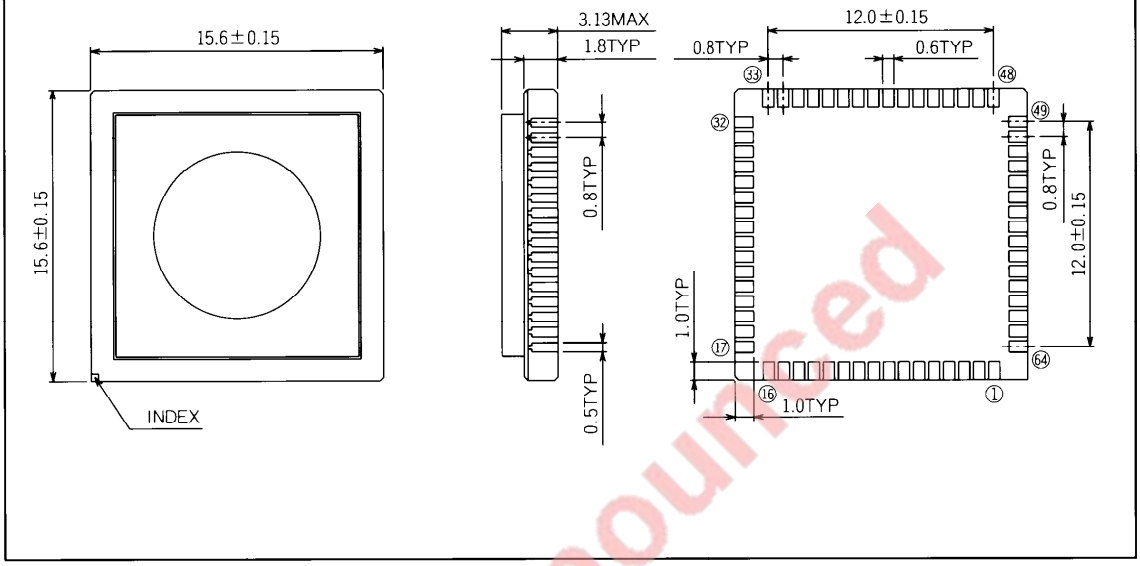
3.8 Package outline

64D0

Metal seal 64pin QFN

EIAJ Package Code	JEDEC Code	Weight(g)
—	—	

Scale : 2.5/1

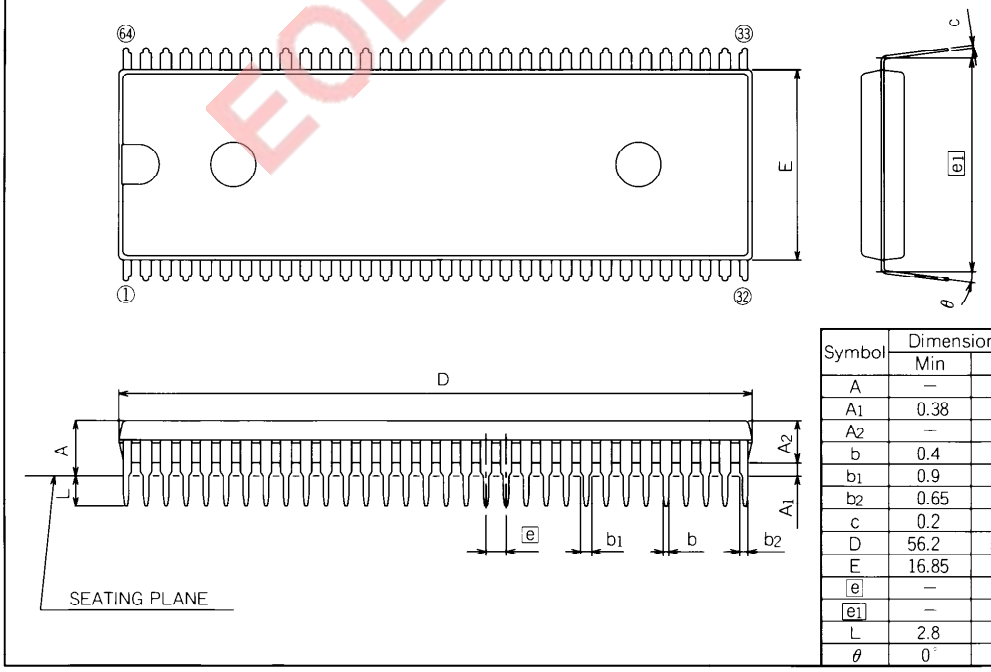


64P4B

Plastic 64pin 750mil SDIP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SDIP64-P-750-1.78	—	7.9	Alloy 42

Scale : 1.5/1



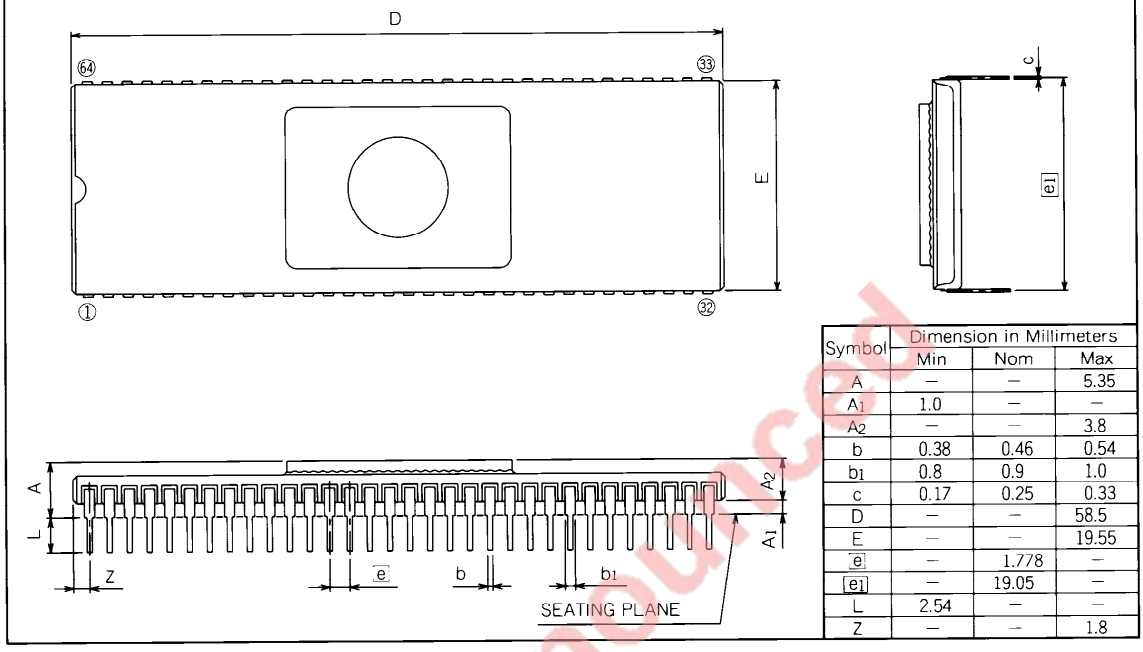
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	5.08
A_1	0.38	—	—
A_2	—	3.8	—
b	0.4	0.5	0.6
b_1	0.9	1.0	1.3
b_2	0.65	0.75	1.05
c	0.2	0.25	0.32
D	56.2	56.4	56.6
E	16.85	17.0	17.15
e	—	1.778	—
e_1	—	19.05	—
L	2.8	—	—
θ	0°	—	15°

64S1B–E

Metal seal 64pin 750mil DIP

EIAJ Package Code	JEDEC Code	Weight(g)
WDIP64-C-750-1.78	–	

Scale : 1.5/1



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	5.35
A_1	1.0	–	–
A_2	–	–	3.8
b	0.38	0.46	0.54
b_1	0.8	0.9	1.0
c	0.17	0.25	0.33
D	–	–	58.5
E	–	–	19.55
e	–	1.778	–
e1	–	19.05	–
L	2.54	–	–
Z	–	–	1.8

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 ← A + M + C When T = 1 M(X) ← 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 ← A ∧ M When T = 1 M(X) ← M(X) ∧ 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	7 0 C←□←0	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+2i	4	2				17+2i	5	3
BBS (Note 4)	Ab or Mb = 1?	Branches when the contents of the bit specified in the accumulator or memory is "1".										03+2i	4	2				07+2i	5	3
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 ∧ 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 ← PC ± offset	Jumps to address specified by adding offset to the program counter.																		
BRK	B ← 1 M(S) ← PC_H S ← S – 1 M(S) ← PC_L S ← S – 1 M(S) ← PS S ← S – 1 PC_L ← AD_L PC_H ← AD_H	Executes a software interrupt.	00	7	1															

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
75	4	2				6D	4	3	7D	5	3	79	5	3							61	6	2	71	6	2							N	V	•	•	•	•	Z	C
35	4	2				2D	4	3	3D	5	3	39	5	3							21	6	2	31	6	2							N	•	•	•	•	•	Z	•
16	6	2				0E	6	3	1E	7	3																						N	•	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																											90	2	2				•	•	•	•	•	•	•	•
																											B0	2	2				•	•	•	•	•	•	•	•
																											F0	2	2				•	•	•	•	•	•	•	•
						2C	4	3																										M_7	M_6	•	•	•	•	Z	•
																											30	2	2				•	•	•	•	•	•	•	•
																											D0	2	2				•	•	•	•	•	•	•	•
																											10	2	2				•	•	•	•	•	•	•	•
																											80	4	2				•	•	•	•	•	•	•	•
																																	•	•	•	1	•	1	•	•

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 or Mb ← 0	Clears the contents of the bit specified in the accumulator or memory to "0".										1B + 2i	2	1				1F + 2i	5	2
CLC	C ← 0	Clears the contents of the carry flag to "0".	18	2	1															
CLD	D ← 0	Clears the contents of decimal mode flag to "0".	D8	2	1															
CLI	I ← 0	Clears the contents of interrupt disable flag to "0".	58	2	1															
CLT	T ← 0	Clears the contents of index X mode flag to "0".	12	2	1															
CLV	V ← 0	Clears the contents of 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 ← $\overline{M}$	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 ← A – 1 or M ← M – 1	Decrements the contents of the accumulator or memory by 1.							1A	2	1				C6	5	2			
DEX	X ← X – 1	Decrements the contents of index register X by 1.	CA	2	1															
DEY	Y ← Y – 1	Decrements the contents of index register Y by 1.	88	2	1															
DIV	A ← (M(zz + X + 1), M(zz + X)) / A M(S) ← 1's complememt of Remainder S ← 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 ← A ∀ M When T = 1 M(X) ← M(X) ∀ 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 ← A + 1 or M ← M + 1	Increments the contents of accumulator or memory by 1.							3A	2	1				E6	5	2			
INX	X ← X + 1	Increments the contents of index register X by 1.	E8	2	1															
INY	Y ← Y + 1	Increments the contents of index register Y by 1.	C8	2	1															

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
																											50	2	2				•	•	•	•	•	•	•	•
																											70	2	2				•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	0
																																	•	•	•	•	0	•	•	•
																																	•	•	•	•	•	0	•	•
																																	•	•	0	•	•	•	•	•
																																	•	0	•	•	•	•	•	•
D5	4	2				CD	4	3	DD	5	3	D9	5	3							C1	6	2	D1	6	2							N	•	•	•	•	•	Z	C
																																	N	•	•	•	•	•	Z	•
						EC	4	3																									N	•	•	•	•	•	Z	C
						CC	4	3																									N	•	•	•	•	•	Z	C
D6	6	2				CE	6	3	DE	7	3																						N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
E2	16	2																															•	•	•	•	•	•	•	•
55	4	2				4D	4	3	5D	5	3	59	5	3							41	6	2	51	6	2							N	•	•	•	•	•	Z	•
F6	6	2				EE	6	3	FE	7	3																						N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•

APPENDIX

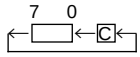
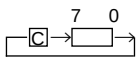
3.9 Machine instructions

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	#
JMP	If addressing mode is ABS $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ If addressing mode is IND $PC_L \leftarrow M(AD_H, AD_L)$ $PC_H \leftarrow M(AD_H, AD_L + 1)$ If addressing mode is ZP, IND $PC_L \leftarrow M(00, AD_L)$ $PC_H \leftarrow M(00, AD_L + 1)$	Jumps to the specified address.																		
JSR	$M(S) \leftarrow PC_H$ $S \leftarrow S - 1$ $M(S) \leftarrow PC_L$ $S \leftarrow S - 1$ After executing the above, if addressing mode is ABS, $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ if addressing mode is SP, $PC_L \leftarrow AD_L$ $PC_H \leftarrow FF$ If addressing mode is ZP, IND, $PC_L \leftarrow M(00, AD_L)$ $PC_H \leftarrow M(00, AD_L + 1)$	After storing contents of program counter in stack, and jumps to the specified address.																		
LDA (Note 2)	When T = 0 $A \leftarrow M$ When T = 1 $M(X) \leftarrow M$	Load accumulator with contents of memory. Load memory indicated by index register X with contents of memory specified by the addressing mode.				A9	2	2							A5	3	2			
LDM	$M \leftarrow nn$	Load memory with immediate value.													3C	4	3			
LDX	$X \leftarrow M$	Load index register X with contents of memory.				A2	2	2							A6	3	2			
LDY	$Y \leftarrow M$	Load index register Y with contents of memory.				A0	2	2							A4	3	2			
LSR	7 0 0→□→C	Shift the contents of accumulator or memory to the right by one bit. The low order bit of accumulator or memory is stored in carry, 7th bit is cleared.							4A	2	1				46	5	2			
MUL	$M(S) \cdot A \leftarrow A \times M(zz + X)$ $S \leftarrow S - 1$	Multiplies the accumulator with the contents of memory specified by the zero page X addressing mode and stores the high byte of the result on the stack and the low byte in the accumulator.																		
NOP	$PC \leftarrow PC + 1$	No operation.	EA	2	1															
ORA (Note 1)	When T = 0 $A \leftarrow A \vee M$ When T = 1 $M(X) \leftarrow M(X) \vee M$	"Logical OR's" the contents of memory and accumulator. The result is stored in the accumulator. "Logical OR's" the contents of memory indicated by index register X and contents of memory specified by the addressing mode. The result is stored in the memory specified by index register X.				09	2	2							05	3	2			

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
						4C	3	3							6C	5	3	B2	4	2													•	•	•	•	•	•	•	•
						20	6	3										02	7	2										22	5	2	•	•	•	•	•	•	•	•
B5	4	2				AD	4	3	BD	5	3	B9	5	3							A1	6	2	B1	6	2							N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
			B6	4	2	AE	4	3				BE	5	3																			N	•	•	•	•	•	Z	•
B4	4	2				AC	4	3	BC	5	3																						N	•	•	•	•	•	Z	•
56	6	2				4E	6	3	5E	7	3																						0	•	•	•	•	•	Z	C
62	15	2																															•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
15	4	2				0D	4	3	1D	5	3	19	5	3							01	6	2	11	6	2							N	•	•	•	•	•	Z	•

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) ← A S ← 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) ← PS S ← 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 ← S + 1 A ← 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 ← S + 1 PS ← 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 ← S + 1 PS ← M(S) S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S)	Returns from an interrupt routine to the main routine.	40	6	1															
RTS	S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S)	Returns from a subroutine to the main routine.	60	6	1															
SBC (Note 1) (Note 5)	When T = 0 A ← A – M – $\overline{C}$ When T = 1 M(X) ← M(X) – M – $\overline{C}$	Subtracts the contents of memory and complement of carry flag from the contents of accumulator. The results are stored into the accumulator. Subtracts contents of complement of carry flag and contents of the memory indicated by the addressing mode from the memory at the address indicated by index register X. The results are stored into the memory of the address indicated by index register X.				E9	2	2							E5	3	2			
SEB	Ab or Mb ← 1	Sets the specified bit in the accumulator or memory to “1”.										0B+2i	2	1				0F+2i	5	2
SEC	C ← 1	Sets the contents of the carry flag to “1”.	38	2	1															
SED	D ← 1	Sets the contents of the decimal mode flag to “1”.	F8	2	1															
SEI	I ← 1	Sets the contents of the interrupt disable flag to “1”.	78	2	1															
SET	T ← 1	Sets the contents of the index X mode flag to “1”.	32	2	1															

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	(Value saved in stack)							
36	6	2				2E	6	3	3E	7	3																						N	•	•	•	•	•	Z	C
76	6	2				6E	6	3	7E	7	3																						N	•	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	(Value saved in stack)							
																																	•	•	•	•	•	•	•	•
F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2							N	V	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	1
																																	•	•	•	•	1	•	•	•
																																	•	•	•	•	•	1	•	•
																																	•	•	1	•	•	•	•	•

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 ← A	Stores the contents of accumulator in memory.													85	4	2			
STP		Stops the oscillator.	42	2	1															
STX	M ← X	Stores the contents of index register X in memory.													86	4	2			
STY	M ← Y	Stores the contents of index register Y in memory.													84	4	2			
TAX	X ← A	Transfers the contents of the accumulator to index register X.	AA	2	1															
TAY	Y ← 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 ← S	Transfers the contents of the stack pointer to index register X.	BA	2	1															
TXA	A ← X	Transfers the contents of index register X to the accumulator.	8A	2	1															
TXS	S ← X	Transfers the contents of index register X to the stack pointer.	9A	2	1															
TYA	A ← 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.

Addressing mode																												Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	2							•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
			96	5	2	8E	5	3																									•	•	•	•	•	•	•	•
94	5	2				8C	5	3																									•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•

Symbol	Contents	Symbol	Contents
IMP	Implied addressing mode	+	Addition
IMM	Immediate addressing mode	–	Subtraction
A	Accumulator or Accumulator addressing mode	∧	Logical OR
		∨	Logical AND
BIT, A	Accumulator bit relative addressing mode	∀	Logical exclusive OR
		—	Negation
ZP	Zero page addressing mode	←	Shows direction of data flow
BIT, ZP	Zero page bit relative addressing mode	X	Index register X
		Y	Index register Y
ZP, X	Zero page X addressing mode	S	Stack pointer
ZP, Y	Zero page Y addressing mode	PC	Program counter
ABS	Absolute addressing mode	PS	Processor status register
ABS, X	Absolute X addressing mode	PC_H	8 high-order bits of program counter
ABS, Y	Absolute Y addressing mode	PC_L	8 low-order bits of program counter
IND	Indirect absolute addressing mode	AD_H	8 high-order bits of address
		AD_L	8 low-order bits of address
ZP, IND	Zero page indirect absolute addressing mode	FF	FF in Hexadecimal notation
		nn	Immediate value
IND, X	Indirect X addressing mode	M	Memory specified by address designation of any addressing mode
IND, Y	Indirect Y addressing mode		
REL	Relative addressing mode	M(X)	Memory of address indicated by contents of index register X
SP	Special page addressing mode		
C	Carry flag	M(S)	Memory of address indicated by contents of stack pointer
Z	Zero flag		
I	Interrupt disable flag	$M(AD_H, AD_L)$	Contents of memory at address indicated by AD_H and AD_L, in AD_H is 8 high-order bits and AD_L is 8 low-order bits.
D	Decimal mode flag		
B	Break flag		
T	X-modified arithmetic mode flag	$M(00, AD_L)$	Contents of address indicated by zero page AD_L
V	Overflow flag	Ab	1 bit of accumulator
N	Negative flag	Mb	1 bit of memory
		OP	Opcode
		n	Number of cycles
		#	Number of bytes

3.10 List of instruction codes

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

3-byte instruction

2-byte instruction

1-byte instruction

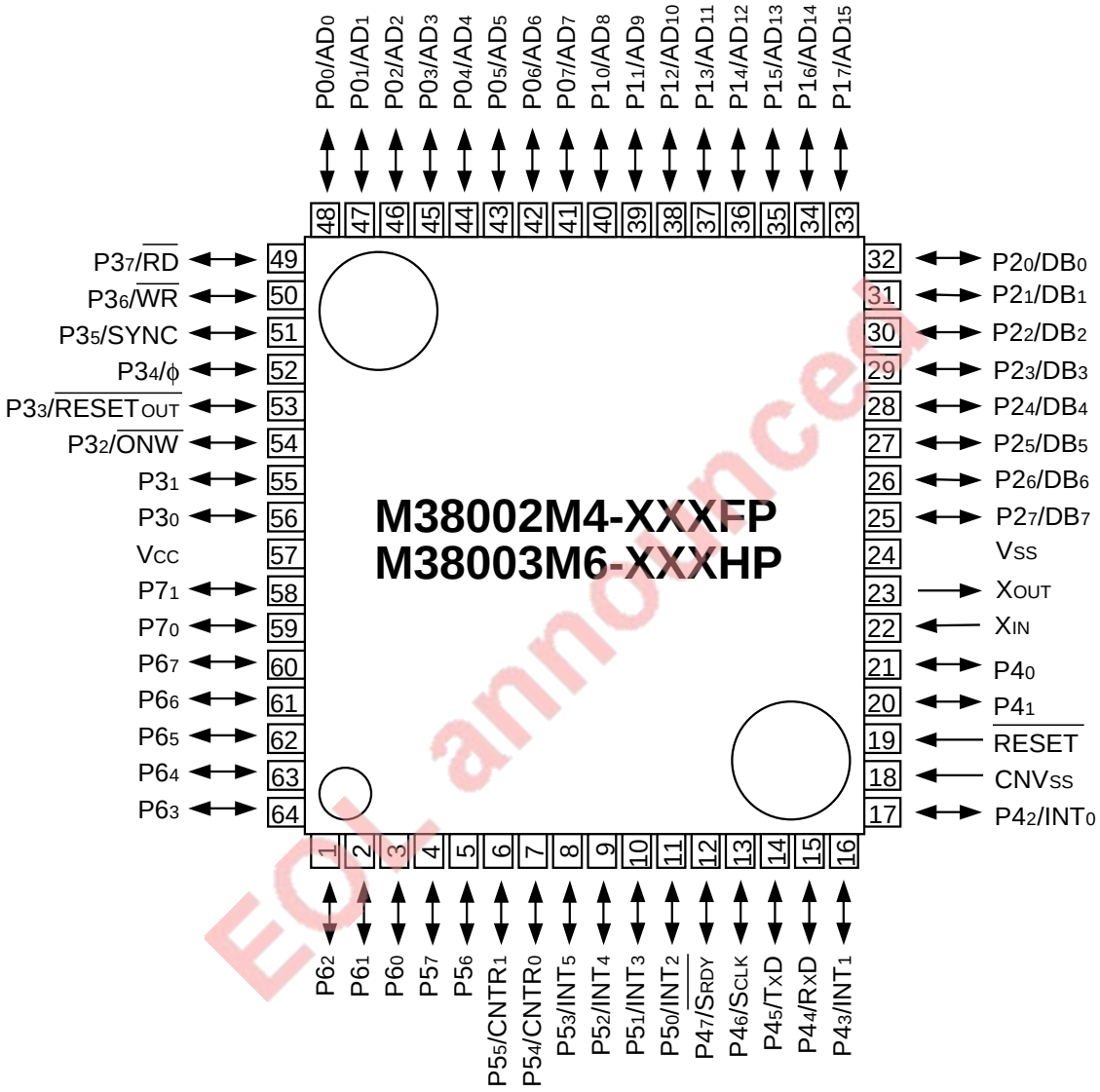
3.11 SFR memory map

0000_{16}	Port P0 (P0)
0001_{16}	Port P0 direction register (P0D)
0002_{16}	Port P1 (P1)
0003_{16}	Port P1 direction register (P1D)
0004_{16}	Port P2 (P2)
0005_{16}	Port P2 direction register (P2D)
0006_{16}	Port P3 (P3)
0007_{16}	Port P3 direction register (P3D)
0008_{16}	Port P4 (P4)
0009_{16}	Port P4 direction register (P4D)
$000A_{16}$	Port P5 (P5)
$000B_{16}$	Port P5 direction register (P5D)
$000C_{16}$	Port P6 (P6)
$000D_{16}$	Port P6 direction register (P6D)
$000E_{16}$	Port P7 (P7)
$000F_{16}$	Port P7 direction register (P7D)
0010_{16}	
0011_{16}	
0012_{16}	
0013_{16}	
0014_{16}	
0015_{16}	
0016_{16}	
0017_{16}	
0018_{16}	Transmit/Receive buffer register (TB/RB)
0019_{16}	Serial I/O status register (SIOSTS)
$001A_{16}$	Serial I/O control register (SIOCON)
$001B_{16}$	UART control register (UARTCON)
$001C_{16}$	Baud rate generator (BRG)
$001D_{16}$	
$001E_{16}$	
$001F_{16}$	

0020_{16}	Prescaler 12 (PRE12)
0021_{16}	Timer 1 (T1)
0022_{16}	Timer 2 (T2)
0023_{16}	Timer XY mode register (TM)
0024_{16}	Prescaler X (PREX)
0025_{16}	Timer X (TX)
0026_{16}	Prescaler Y (PREY)
0027_{16}	Timer Y (TY)
0028_{16}	
0029_{16}	
$002A_{16}$	
$002B_{16}$	
$002C_{16}$	
$002D_{16}$	
$002E_{16}$	
$002F_{16}$	
0030_{16}	
0031_{16}	
0032_{16}	
0033_{16}	
0034_{16}	
0035_{16}	
0036_{16}	
0037_{16}	
0038_{16}	
0039_{16}	
$003A_{16}$	Interrupt edge selection register (INTEDGE)
$003B_{16}$	CPU mode register (CPUM)
$003C_{16}$	Interrupt request register 1 (IREQ1)
$003D_{16}$	Interrupt request register 2 (IREQ2)
$003E_{16}$	Interrupt control register 1 (ICON1)
$003F_{16}$	Interrupt control register 2 (ICON2)

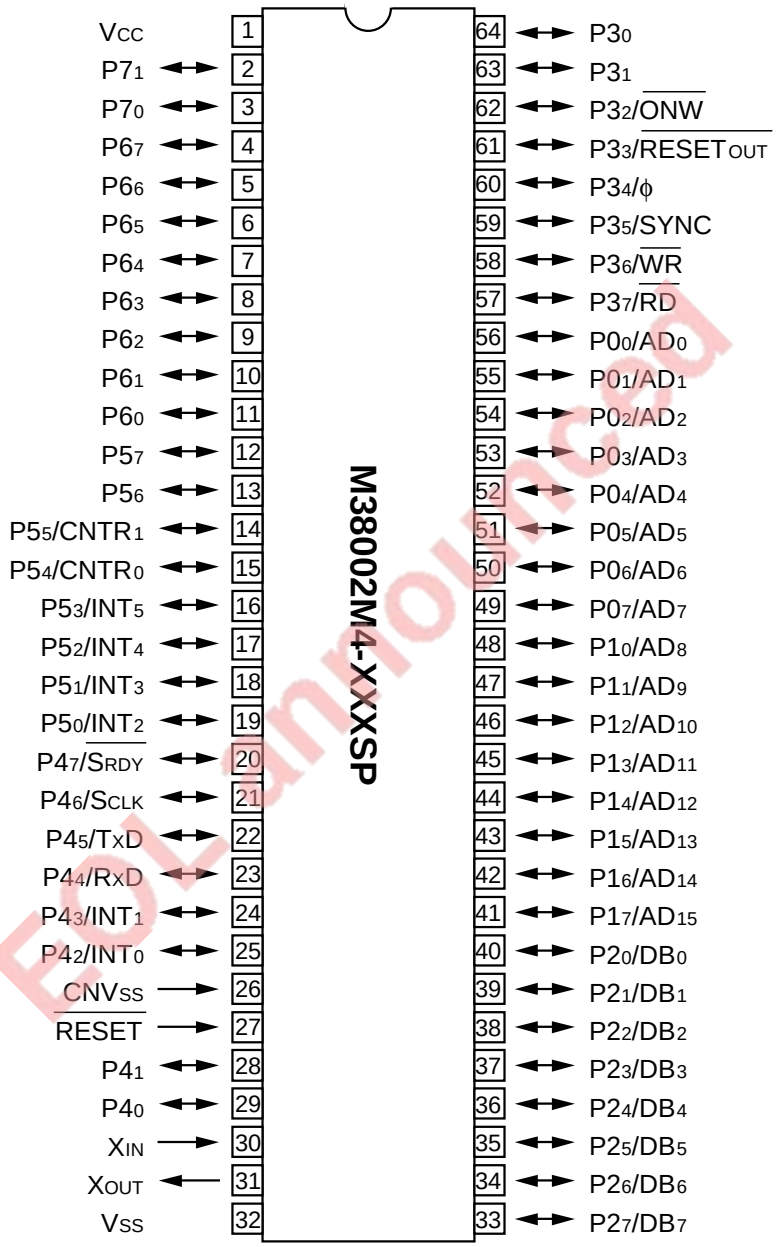
3.12 Pin configuration

PIN CONFIGURATION (TOP VIEW)



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

PIN CONFIGURATION (TOP VIEW)



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

3800 Group
User's Manual

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

H-EE418-A KI-9603