

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

3850/3851 Group

User's Manual

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

keep safety first in your circuit designs !

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

Notes regarding these materials

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

REVISION DESCRIPTION LIST

3850/3851 GROUP USER'S MANUAL

Rev. No.	Revision Description	Rev. date
1.0	First Edition	981021

Preface

This user's manual describes Mitsubishi's CMOS 8-bit microcomputers 3851 Group and 3850 Group. After reading this manual, the user should have a thorough knowledge of their functions and features, and should be able to fully utilize the product. The manual starts with specifications and ends with application examples.

The difference between the 3851 Group and 3850 Group is the I²C-BUS built-in or not. The 3850 Group does not have the built-in I²C-BUS. Accordingly, use this user's manual with care, considering the difference between the 3851 Group and 3850 Group. This user's manual mainly explains the 3851 Group. The difference is explained in the section "**FUNCTIONAL DESCRIPTION SUPPLEMENT of Chapter 1**".

For details of software, refer to the "740 SERIES SOFTWARE MANUAL."

For details of development support tools, refer to the data book or the data sheet of "DEVELOPMENT SUPPORT TOOLS FOR 740 FAMILY".

BEFORE USING THIS MANUAL

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

1. Organization

● CHAPTER 1 HARDWARE

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

● CHAPTER 2 APPLICATION

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

● CHAPTER 3 APPENDIX

This chapter includes necessary information for systems development using the microcomputer, such as the electrical characteristics, the list of registers, the Mask ROM confirmation form (for mask ROM version), the ROM programming confirmation form (for One Time PROM version), and the Mark specification form 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 :

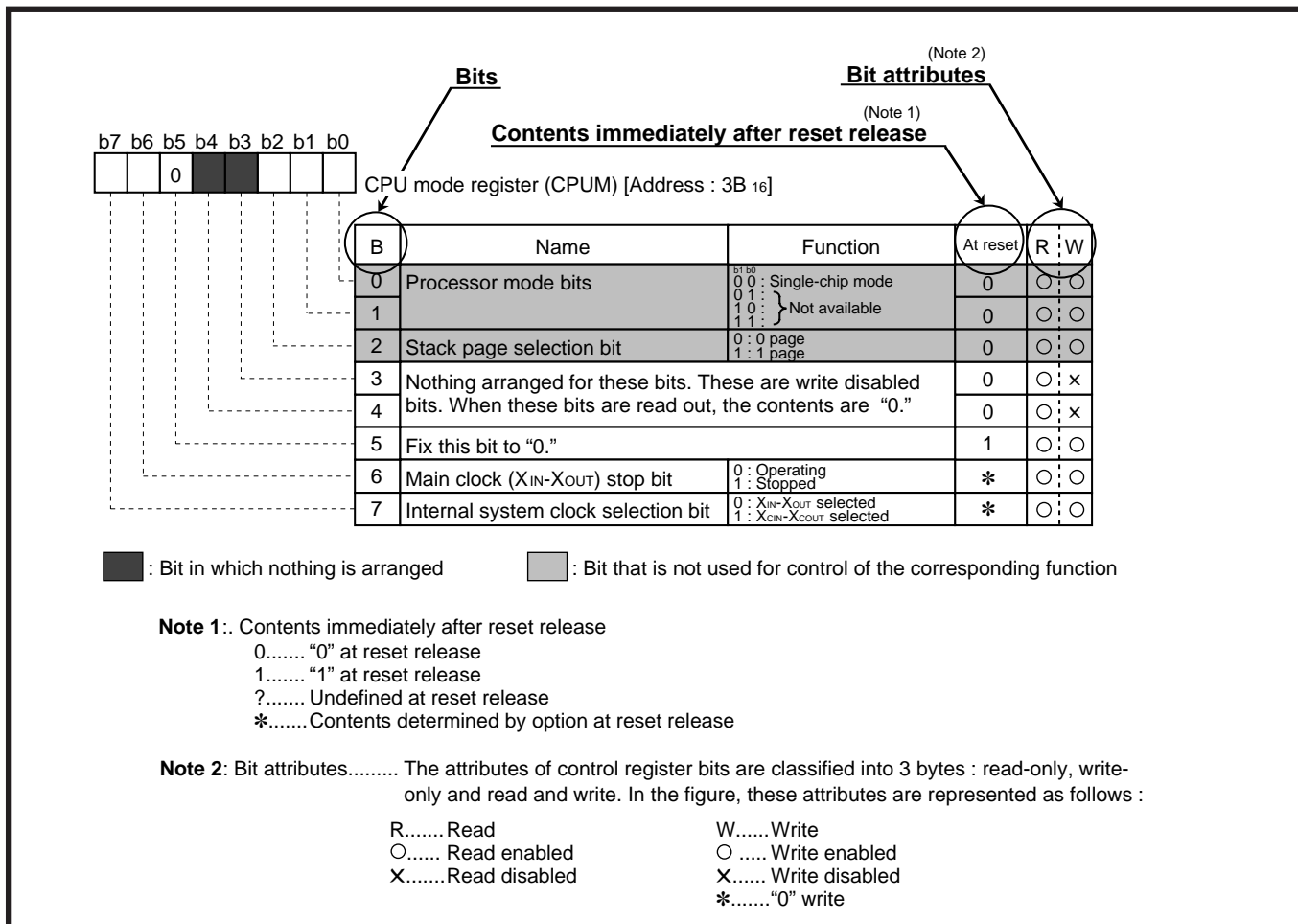


Table of contents

CHAPTER 1 HARDWARE

DESCRIPTION	1-2
FEATURES	1-2
APPLICATION	1-2
PIN CONFIGURATION (TOP VIEW)	1-2
FUNCTIONAL BLOCK	1-3
PIN DESCRIPTION	1-4
PART NUMBERING	1-5
GROUP EXPANSION	1-6
Memory Type	1-6
Memory Size	1-6
Packages	1-6
FUNCTIONAL DESCRIPTION	1-7
Central Processing Unit (CPU)	1-7
MEMORY	1-11
I/O PORTS	1-13
INTERRUPTS	1-16
TIMERS	1-19
SERIAL I/O	1-21
MULTI-MASTER I ² C-BUS INTERFACE	1-25
PULSE WIDTH MODULATION (PWM)	1-36
A-D CONVERTER	1-38
WATCHDOG TIMER	1-39
RESET CIRCUIT	1-40
CLOCK GENERATING CIRCUIT	1-42
NOTES ON PROGRAMMING	1-45
Processor Status Register	1-45
Interrupts	1-45
Decimal Calculations	1-45
Timers	1-45
Multiplication and Division Instructions	1-45
Ports	1-45
Serial I/O	1-45
A-D Converter	1-45
Instruction Execution Time	1-45
DATA REQUIRED FOR MASK ORDERS	1-46
DATA REQUIRED FOR ROM WRITING ORDERS	1-46
ROM PROGRAMMING METHOD	1-46
FUNCTIONAL DESCRIPTION SUPPLEMENT	1-47
Interrupt	1-47
Timing After Interrupt	1-48
A-D Converter	1-49
MISRG	1-51
3850 Group	1-53

CHAPTER 2 APPLICATION

2.1 I/O port	2-2
2.1.1 Memory map	2-2
2.1.2 Relevant registers	2-2
2.1.3 Handling of unused pins	2-3
2.1.4 Notes on input and output pins	2-4
2.1.5 Termination of unused pins	2-5
2.2 Timer	2-6
2.2.1 Memory map	2-6
2.2.2 Relevant registers	2-6
2.2.3 Timer application examples	2-12
2.2.4 Notes on timer	2-25
2.3 Serial I/O	2-26
2.3.1 Memory map	2-26
2.3.2 Relevant registers	2-27
2.3.3 Serial I/O connection examples	2-31
2.3.4 Setting of serial I/O transfer data format	2-33
2.3.5 Serial I/O application examples	2-34
2.3.6 Notes on serial I/O	2-52
2.4 Muti-master I²C-BUS Interface	2-55
2.4.1 Memory map	2-55
2.4.2 Relevant registers	2-55
2.4.3 I ² C-BUS overview	2-61
2.4.4 Communication format	2-62
2.4.5 Synchronization and Arbitration lost	2-63
2.4.6 SMBUS communication usage example	2-65
2.4.7 Notes on muti-master I ² C-BUS Interface	2-81
2.4.8 Notes on programming for SMBUS interface	2-84
2.5 PWM	2-85
2.5.1 Memory map	2-85
2.5.2 Relevant registers	2-85
2.5.3 PWM output circuit application example	2-87
2.5.4 Notes on PWM	2-89
2.6 A-D converter	2-90
2.6.1 Memory map	2-90
2.6.2 Relevant registers	2-90
2.6.3 A-D converter application examples	2-93
2.6.4 Notes on A-D converter	2-95
2.7 Reset	2-96
2.7.1 Connection example of reset IC	2-96
2.7.2 Notes on RESET pin	2-97

CHAPTER 3 APPENDIX

3.1 Electrical characteristics	3-2
3.1.1 Absolute maximum ratings	3-2
3.1.2 Recommended operating conditions	3-3
3.1.3 Electrical characteristics	3-5
3.1.4 A-D converter characteristics	3-6
3.1.5 Timing requirements	3-7
3.1.6 Switching characteristics	3-8
3.1.7 Multi-master I ² C-BUS bus line characteristics	3-11
3.2 Standard characteristics	3-12
3.2.1 Power source current characteristic examples	3-12
3.2.2 Port standard characteristic examples	3-15
3.2.3 A-D conversion standard characteristics	3-17
3.3 Notes on use	3-18
3.3.1 Notes on interrupts	3-18
3.3.2 Notes on timer	3-19
3.3.3 Notes on serial I/O	3-19
3.3.4 Notes on multi-master I ² C-BUS interface	3-21
3.3.5 Notes on A-D converter	3-24
3.3.6 Notes on <u>watchdog</u> timer	3-24
3.3.7 Notes on <u>RESET</u> pin	3-24
3.3.8 Notes on input and output pins	3-25
3.3.9 Notes on low-speed operation mode	3-26
3.3.10 Notes on restarting oscillation	3-26
3.3.11 Notes on programming	3-27
3.3.12 Programming and test of built-in PROM version	3-29
3.3.13 Notes on built-in PROM version	3-29
3.3.14 Termination of unused pins	3-31
3.4 Countermeasures against noise	3-32
3.4.1 Shortest wiring length	3-32
3.4.2 Connection of bypass capacitor across V _{SS} line and V _{CC} line	3-34
3.4.3 Wiring to analog input pins	3-35
3.4.4 Oscillator concerns	3-35
3.4.5 Setup for I/O ports	3-37
3.4.6 Providing of watchdog timer function by software	3-38
3.5 List of registers	3-39
3.6 Mask ROM confirmation form	3-56
3.7 ROM programming confirmation form	3-66
3.8 Mark specification form	3-74
3.9 Package outline	3-76
3.10 Machine instructions	3-78
3.11 List of instruction codes	3-89
3.12 SFR memory map	3-90
3.13 Pin configurations	3-91

List of figures

CHAPTER 1 HARDWARE

Fig. 1 M38513M4-XXXFP/SP pin configuration	1-2
Fig. 2 Functional block diagram.....	1-3
Fig. 3 Part numbering.....	1-5
Fig. 4 Memory expansion plan.....	1-6
Fig. 5 740 Family CPU register structure.....	1-7
Fig. 6 Register push and pop at interrupt generation and subroutine call	1-8
Fig. 7 Structure of CPU mode register	1-10
Fig. 8 Memory map diagram	1-11
Fig. 9 Memory map of special function register (SFR)	1-12
Fig. 10 Port block diagram (1).....	1-14
Fig. 11 Port block diagram (2).....	1-15
Fig. 12 Interrupt control.....	1-18
Fig. 13 Structure of Interrupt-related registers (1)	1-18
Fig. 14 Structure of timer XY mode register.....	1-19
Fig. 15 Structure of timer count source selection register.....	1-19
Fig. 16 Block diagram of timer X, timer 1 and timer 2	1-20
Fig. 17 Block diagram of clock synchronous serial I/O.....	1-21
Fig. 18 Operation of clock synchronous serial I/O function.....	1-21
Fig. 19 Block diagram of UART serial I/O	1-22
Fig. 20 Operation of UART serial I/O function	1-23
Fig. 21 Structure of serial I/O control registers.....	1-24
Fig. 22 Block diagram of multi-master I ² C-BUS interface	1-25
Fig. 23 Structure of I ² C address register.....	1-26
Fig. 24 Structure of I ² C clock control register	1-27
Fig. 25 SDA/SCL pin selection bit.....	1-28
Fig. 26 Structure of I ² C control register.....	1-28
Fig. 27 Structure of I ² C status register	1-30
Fig. 28 Interrupt request signal generating timing	1-30
Fig. 29 START condition generating timing diagram	1-31
Fig. 30 STOP condition generating timing diagram	1-31
Fig. 31 START/STOP condition detecting timing diagram	1-31
Fig. 32 STOP condition detecting timing diagram.....	1-31
Fig. 33 Structure of I ² C START/STOP condition control register	1-33
Fig. 34 Address data communication format.....	1-33
Fig. 35 Timing of PWM period.....	1-36
Fig. 36 Block diagram of PWM function.....	1-36
Fig. 37 Structure of PWM control register	1-37
Fig. 38 PWM output timing when PWM register or PWM prescaler is changed.....	1-37
Fig. 39 Structure of AD control register	1-38
Fig. 40 Structure of A-D conversion registers	1-38
Fig. 41 Block diagram of A-D converter	1-38
Fig. 42 Block diagram of Watchdog timer	1-39
Fig. 43 Structure of Watchdog timer control register	1-39
Fig. 44 Reset circuit example	1-40
Fig. 45 Reset sequence	1-40
Fig. 46 Internal status at reset	1-41
Fig. 47 Ceramic resonator circuit	1-42

List of figures

Fig. 48 External clock input circuit	1-42
Fig. 49 Structure of MISRG	1-43
Fig. 50 System clock generating circuit block diagram (Single-chip mode)	1-43
Fig. 51 State transitions of system clock	1-44
Fig. 52 Programming and testing of One Time PROM version	1-46
Fig. 53 Timing chart after an interrupt occurs	1-48
Fig. 54 Time up to execution of the interrupt processing routine	1-48
Fig. 55 A-D conversion equivalent circuit	1-50
Fig. 56 A-D conversion timing chart	1-50
Fig. 57 Structure of MISRG	1-52
Fig. 58 Structure of I ² C START/STOP condition control register	1-52
Fig. 59 Memory expansion plan of 3850 group	1-53
Fig. 60 Structure of Interrupt request register 1 of 3850 group	1-54
Fig. 61 Structure of Interrupt control register 1 of 3850 group	1-54

CHAPTER 2 APPLICATION

Fig. 2.1.1 Memory map of registers relevant to I/O port	2-2
Fig. 2.1.2 Structure of Port Pi (i = 0, 1, 2, 3, 4)	2-2
Fig. 2.1.3 Structure of Port Pi direction register (i = 0, 1, 2, 3, 4)	2-3
Fig. 2.2.1 Memory map of registers relevant to timers	2-6
Fig. 2.2.2 Structure of Prescaler 12, Prescaler X, Prescaler Y	2-6
Fig. 2.2.3 Structure of Timer 1	2-7
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 Timer count source set register	2-9
Fig. 2.2.7 Structure of Interrupt request register 1	2-10
Fig. 2.2.8 Structure of Interrupt request register 2	2-10
Fig. 2.2.9 Structure of Interrupt control register 1	2-11
Fig. 2.2.10 Structure of Interrupt control register 2	2-11
Fig. 2.2.11 Timers connection and setting of division ratios	2-13
Fig. 2.2.12 Relevant registers setting	2-14
Fig. 2.2.13 Control procedure	2-15
Fig. 2.2.14 Peripheral circuit example	2-16
Fig. 2.2.15 Timers connection and setting of division ratios	2-16
Fig. 2.2.16 Relevant registers setting	2-17
Fig. 2.2.17 Control procedure	2-18
Fig. 2.2.18 Judgment method of valid/invalid of input pulses	2-19
Fig. 2.2.19 Relevant registers setting	2-20
Fig. 2.2.20 Control procedure	2-21
Fig. 2.2.21 Timers connection and setting of division ratios	2-22
Fig. 2.2.22 Relevant registers setting	2-23
Fig. 2.2.23 Control procedure	2-24
Fig. 2.3.1 Memory map of registers relevant to serial I/O	2-26
Fig. 2.3.2 Structure of Transmit/Receive buffer register	2-27
Fig. 2.3.3 Structure of Serial I/O status register	2-27
Fig. 2.3.4 Structure of Serial I/O control register	2-28
Fig. 2.3.5 Structure of UART control register	2-28
Fig. 2.3.6 Structure of Baud rate generator	2-29
Fig. 2.3.7 Structure of Interrupt edge selection register	2-29
Fig. 2.3.8 Structure of Interrupt request register 2	2-29
Fig. 2.3.9 Structure of Interrupt control register 2	2-30

Fig. 2.3.10 Serial I/O connection examples (1)	2-31
Fig. 2.3.11 Serial I/O connection examples (2)	2-32
Fig. 2.3.12 Serial I/O transfer data format	2-33
Fig. 2.3.13 Connection diagram	2-34
Fig. 2.3.14 Timing chart (using clock synchronous serial I/O)	2-34
Fig. 2.3.15 Registers setting relevant to transmitting side.....	2-35
Fig. 2.3.16 Registers setting relevant to receiving side	2-36
Fig. 2.3.17 Control procedure of transmitting side.....	2-37
Fig. 2.3.18 Control procedure of receiving side	2-38
Fig. 2.3.19 Connection diagram	2-39
Fig. 2.3.20 Timing chart	2-39
Fig. 2.3.21 Registers setting relevant to serial I/O	2-40
Fig. 2.3.22 Setting of serial I/O transmission data	2-40
Fig. 2.3.23 Control procedure of serial I/O	2-41
Fig. 2.3.24 Connection diagram	2-42
Fig. 2.3.25 Timing chart	2-43
Fig. 2.3.26 Relevant registers setting	2-43
Fig. 2.3.27 Control procedure of master unit.....	2-44
Fig. 2.3.28 Control procedure of slave unit	2-45
Fig. 2.3.29 Connection diagram (Communication using UART)	2-46
Fig. 2.3.30 Timing chart (using UART)	2-46
Fig. 2.3.31 Registers setting relevant to transmitting side.....	2-48
Fig. 2.3.32 Registers setting relevant to receiving side	2-49
Fig. 2.3.33 Control procedure of transmitting side.....	2-50
Fig. 2.3.34 Control procedure of receiving side	2-51
Fig. 2.3.35 Sequence of setting serial I/O control register again	2-53
Fig. 2.4.1 Memory map of registers relevant to I ² C-BUS interface	2-55
Fig. 2.4.2 Structure of I ² C data shift register.....	2-55
Fig. 2.4.3 Structure of I ² C address register	2-56
Fig. 2.4.4 Structure of I ² C status register.....	2-56
Fig. 2.4.5 Structure of I ² C control register	2-57
Fig. 2.4.6 Structure of I ² C clock control register	2-58
Fig. 2.4.7 Structure of I ² C START/STOP condition control register	2-59
Fig. 2.4.8 Structure of Interrupt request register 1	2-59
Fig. 2.4.9 Structure of Interrupt control register 1	2-60
Fig. 2.4.10 I ² C-BUS connection structure	2-61
Fig. 2.4.11 I ² C-BUS communication format example	2-62
Fig. 2.4.12 RESTART condition of master reception	2-63
Fig. 2.4.13 SCL waveforms when synchronizing clocks.....	2-64
Fig. 2.4.14 Initial setting example using SMBUS communication.....	2-66
Fig. 2.4.15 Read Word protocol communication as SMBUS master device	2-67
Fig. 2.4.16 Transmission process of START condition and slave address	2-68
Fig. 2.4.17 Transmission process of command	2-69
Fig. 2.4.18 Transmission process of RESTART condition and slave address + read bit	2-70
Fig. 2.4.19 Reception process of lower data	2-71
Fig. 2.4.20 Reception process of upper data	2-72
Fig. 2.4.21 Generating of STOP condition	2-73
Fig. 2.4.22 Communication example as SMBUS slave device	2-74
Fig. 2.4.23 Reception process of START condition and slave address	2-75
Fig. 2.4.24 Reception process of command.....	2-76
Fig. 2.4.25 Reception process of RESTART condition and slave address + read bit.....	2-77
Fig. 2.4.26 Transmission process of lower data.....	2-78

List of figures

Fig. 2.4.27 Transmission process of upper data	2-79
Fig. 2.4.28 Reception of STOP condition	2-80
Fig. 2.5.1 Memory map of registers relevant to PWM	2-85
Fig. 2.5.2 Structure of PWM control register	2-85
Fig. 2.5.3 Structure of PWM prescaler	2-86
Fig. 2.5.4 Structure of PWM register	2-86
Fig. 2.5.5 Connection diagram	2-87
Fig. 2.5.6 PWM output timing.....	2-87
Fig. 2.5.7 Setting of related registers	2-88
Fig. 2.5.8 PWM output	2-88
Fig. 2.5.9 Control procedure.....	2-89
Fig. 2.6.1 Memory map of registers relevant to A-D converter	2-90
Fig. 2.6.2 Structure of A-D control register	2-90
Fig. 2.6.3 Structure of A-D conversion register (high-order)	2-91
Fig. 2.6.4 Structure of A-D conversion register (low-order).....	2-91
Fig. 2.6.5 Structure of Interrupt request register 2	2-92
Fig. 2.6.6 Structure of Interrupt control register 2	2-92
Fig. 2.6.7 Connection diagram	2-93
Fig. 2.6.8 Relevant registers setting	2-93
Fig. 2.6.9 Control procedure for 8-bit read	2-94
Fig. 2.6.10 Control procedure for 10-bit read	2-94
Fig. 2.7.1 Example of poweron reset circuit	2-96
Fig. 2.7.2 RAM backup system	2-96

CHAPTER 3 APPENDIX

Fig. 3.1.1 Circuit for measuring output switching characteristics (1)	3-9
Fig. 3.1.2 Circuit for measuring output switching characteristics (2)	3-9
Fig. 3.1.3 Timing chart	3-10
Fig. 3.1.4 Timing diagram of multi-master I ² C-BUS	3-11
Fig. 3.2.1 Power source current characteristic examples ($f(X_{IN}) = 8\text{MHz}$, in high-speed mode)	3-12
Fig. 3.2.2 Power source current characteristic examples ($f(X_{IN}) = 8\text{MHz}$, in middle-speed mode)	3-12
Fig. 3.2.3 Power source current characteristic examples ($f(X_{IN}) = 4\text{MHz}$, in high-speed mode)	3-13
Fig. 3.2.4 Power source current characteristic examples ($f(X_{IN}) = 4\text{MHz}$, in middle-speed mode)	3-13
Fig. 3.2.5 Power source current characteristic examples ($f(X_{CIN}) = 32\text{KHz}$, in low-speed mode)	3-14
Fig. 3.2.6 Standard characteristic examples of CMOS output port at P-channel drive	3-15
Fig. 3.2.7 Standard characteristic examples of CMOS output port at N-channel drive	3-15
Fig. 3.2.8 Standard characteristic examples of N-channel open-drain output port at N-channel drive	3-16
Fig. 3.2.9 Standard characteristic examples of CMOS large current output port at N-channel drive	3-16
Fig. 3.2.10 A-D conversion standard characteristics.....	3-17
Fig. 3.3.1 Sequence of switch the detection edge.....	3-18
Fig. 3.3.2 Sequence of check of interrupt request bit	3-18
Fig. 3.3.3 Sequence of setting serial I/O control register again	3-20
Fig. 3.3.4 Ceramic resonator circuit	3-26
Fig. 3.3.5 Initialization of processor status register	3-27
Fig. 3.3.6 Sequence of PLP instruction execution	3-27

Fig. 3.3.7 Stack memory contents after PHP instruction execution	3-27
Fig. 3.3.8 Interrupt routine	3-28
Fig. 3.3.9 Status flag at decimal calculations	3-28
Fig. 3.3.10 Programming and testing of One Time PROM version	3-29
Fig. 3.4.1 Selection of packages	3-32
Fig. 3.4.2 Wiring for the RESET pin	3-32
Fig. 3.4.3 Wiring for clock I/O pins	3-33
Fig. 3.4.4 Wiring for CNVss pin	3-33
Fig. 3.4.5 Wiring for the V _{PP} pin of the One Time PROM and the EPROM version	3-34
Fig. 3.4.6 Bypass capacitor across the V _{ss} line and the V _{cc} line	3-34
Fig. 3.4.7 Analog signal line and a resistor and a capacitor	3-35
Fig. 3.4.8 Wiring for a large current signal line	3-35
Fig. 3.4.9 Wiring of RESET pin	3-36
Fig. 3.4.10 V _{ss} pattern on the underside of an oscillator	3-36
Fig. 3.4.11 Setup for I/O ports	3-37
Fig. 3.4.12 Watchdog timer by software	3-38
Fig. 3.5.1 Structure of Port Pi (i=0, 1, 2, 3, 4)	3-39
Fig. 3.5.2 Structure of Port Pi direction register(i=0, 1, 2, 3, 4)	3-39
Fig. 3.5.3 Structure of Transmit/Receive buffer register	3-40
Fig. 3.5.4 Structure of Serial I/O status register	3-40
Fig. 3.5.5 Structure of Serial I/O control register	3-41
Fig. 3.5.6 Structure of UART control register	3-41
Fig. 3.5.7 Structure of Baud rate generator	3-42
Fig. 3.5.8 Structure of PWM control register	3-42
Fig. 3.5.9 Structure of PWM prescaler	3-43
Fig. 3.5.10 Structure of PWM register	3-43
Fig. 3.5.11 Structure of Prescaler 12, Prescaler X, Prescaler Y	3-44
Fig. 3.5.12 Structure of Timer 1	3-44
Fig. 3.5.13 Structure of Timer 2, Timer X, Timer Y	3-45
Fig. 3.5.14 Structure of timer count source selection register	3-45
Fig. 3.5.15 Structure of Timer XY mode register	3-46
Fig. 3.5.16 Structure of I ² C data shift register	3-47
Fig. 3.5.17 Structure of I ² C address register	3-47
Fig. 3.5.18 Structure of I ² C status register	3-48
Fig. 3.5.19 Structure of I ² C control register	3-48
Fig. 3.5.20 Structure of I ² C clock control register	3-49
Fig. 3.5.21 Structure of I ² C START/STOP condition control register	3-50
Fig. 3.5.22 Structure of A-D control register	3-50
Fig. 3.5.23 Structure of A-D conversion register(low-order)	3-51
Fig. 3.5.24 Structure of A-D conversion register (high-order)	3-51
Fig. 3.5.25 Structure of MISRG	3-52
Fig. 3.5.26 Structure of Watchdog timer control register	3-52
Fig. 3.5.27 Structure of Interrupt edge selection register	3-53
Fig. 3.5.28 Structure of CPU mode register	3-53
Fig. 3.5.29 Structure of Interrupt request register 1	3-54
Fig. 3.5.30 Structure of Interrupt request register 2	3-54
Fig. 3.5.31 Structure of Interrupt control register 1	3-55
Fig. 3.5.32 Structure of Interrupt control register 2	3-55
Fig. 3.13.1 M38513M4-XXXFP/SP pin configuration	3-91

List of tables

CHAPTER 1 HARDWARE

Table 1 Pin description	1-4
Table 2 Push and pop instructions of accumulator or processor status register	1-8
Table 3 Set and clear instructions of each bit of processor status register	1-9
Table 4 I/O port function table	1-13
Table 5 Interrupt vector address and priority	1-17
Table 6 Multi-master I ² C-BUS interface functions	1-25
Table 7 Set values of I ² C clock control register and SCL frequency	1-27
Table 8 START condition generating timing table	1-31
Table 9 STOP condition generating timing table	1-31
Table 10 START condition/STOP condition detecting conditions	1-31
Table 11 Recommended set value to START/STOP condition set bits (SSC4–SSC0) for each oscillation frequency	1-33
Table 12 Programming adapter	1-46
Table 13 Interrupt sources, vector addresses and interrupt priority	1-47
Table 14 Change of A-D conversion register during A-D conversion	1-49

CHAPTER 2 APPLICATION

Table 2.1.1 Handling of unused pins	2-3
Table 2.2.1 CNTR ₀ / CNTR ₁ active edge switch bit function	2-8
Table 2.3.1 Setting example of Baud rate generator values and transfer bit rate values	2-47
Table 2.4.1 Set value of I ² C clock control register and SCL frequency	2-58

CHAPTER 3 APPENDIX

Table 3.1.1 Absolute maximum ratings	3-2
Table 3.1.2 Recommended operating conditions (1)	3-3
Table 3.1.3 Recommended operating conditions (2)	3-4
Table 3.1.4 Electrical characteristics (1)	3-5
Table 3.1.5 Electrical characteristics (2)	3-6
Table 3.1.6 A-D converter characteristics	3-6
Table 3.1.7 Timing requirements (1)	3-7
Table 3.1.8 Timing requirements (2)	3-7
Table 3.1.9 Switching requirements (1)	3-8
Table 3.1.10 Switching requirements (2)	3-8
Table 3.1.11 Multi-master I ² C-BUS bus line characteristics	3-11
Table 3.3.1 Programming adapters	3-29
Table 3.3.2 PROM programmer address setting	3-30
Table 3.5.1 CNTR ₀ / CNTR ₁ active edge switch bit function	3-46
Table 3.5.2 Set value of I ² C clock control register and SCL frequency	3-49



CHAPTER 1

HARDWARE

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

HARDWARE

DESCRIPTION/FEATURES/APPLICATION/PIN CONFIGURATION

DESCRIPTION

The 3851 group is the 8-bit microcomputer based on the 740 family core technology.

The 3851 group is designed for the household products and office automation equipment and includes serial I/O functions, 8-bit timer, A-D converter, and I²C-bus interface.

FEATURES

- Basic machine-language instructions 71
- Minimum instruction execution time 0.5 μs (at 8 MHz oscillation frequency)
- Memory size
 - ROM 16 K to 24 Kbytes
 - RAM 512 to 640 bytes
- Programmable input/output ports 34
- Interrupts 16 sources, 16 vectors
- Timers 8-bit X 4
- Serial I/O 8-bit X 1 (UART or Clock-synchronized)
- Multi-master I²C-bus interface (option) 1 channel
- PWM 8-bit X 1
- A-D converter 10-bit X 5 channels
- Watchdog timer 16-bit X 1
- Clock generating circuit Built-in 2 circuits (connect to external ceramic resonator or quartz-crystal oscillator)

- Power source voltage
 - In high-speed mode 4.0 to 5.5 V (at 8 MHz oscillation frequency)
 - In high-speed mode 2.7 to 5.5 V (at 4 MHz oscillation frequency)
 - In middle-speed mode 2.7 to 5.5 V (at 8 MHz oscillation frequency)
 - In low-speed mode 2.7 to 5.5 V (at 32 kHz oscillation frequency)
- Power dissipation
 - In high-speed mode 34 mW (at 8 MHz oscillation frequency, at 5 V power source voltage)
 - In low-speed mode 60 μW (at 32 kHz oscillation frequency, at 3 V power source voltage)
- Operating temperature range -20 to 85°C

APPLICATION

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

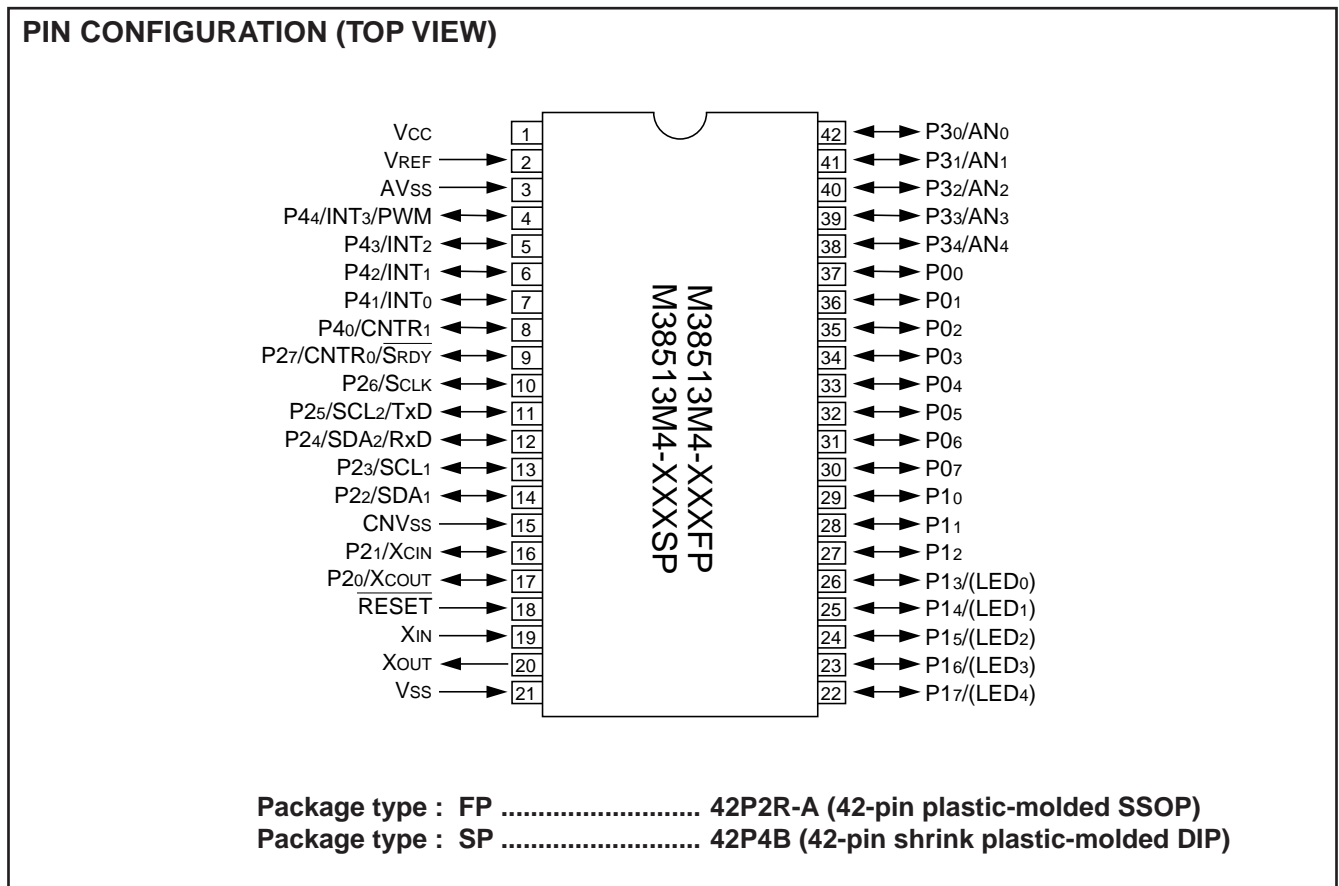


Fig. 1 M38513M4-XXXXFP/SP pin configuration

FUNCTIONAL BLOCK

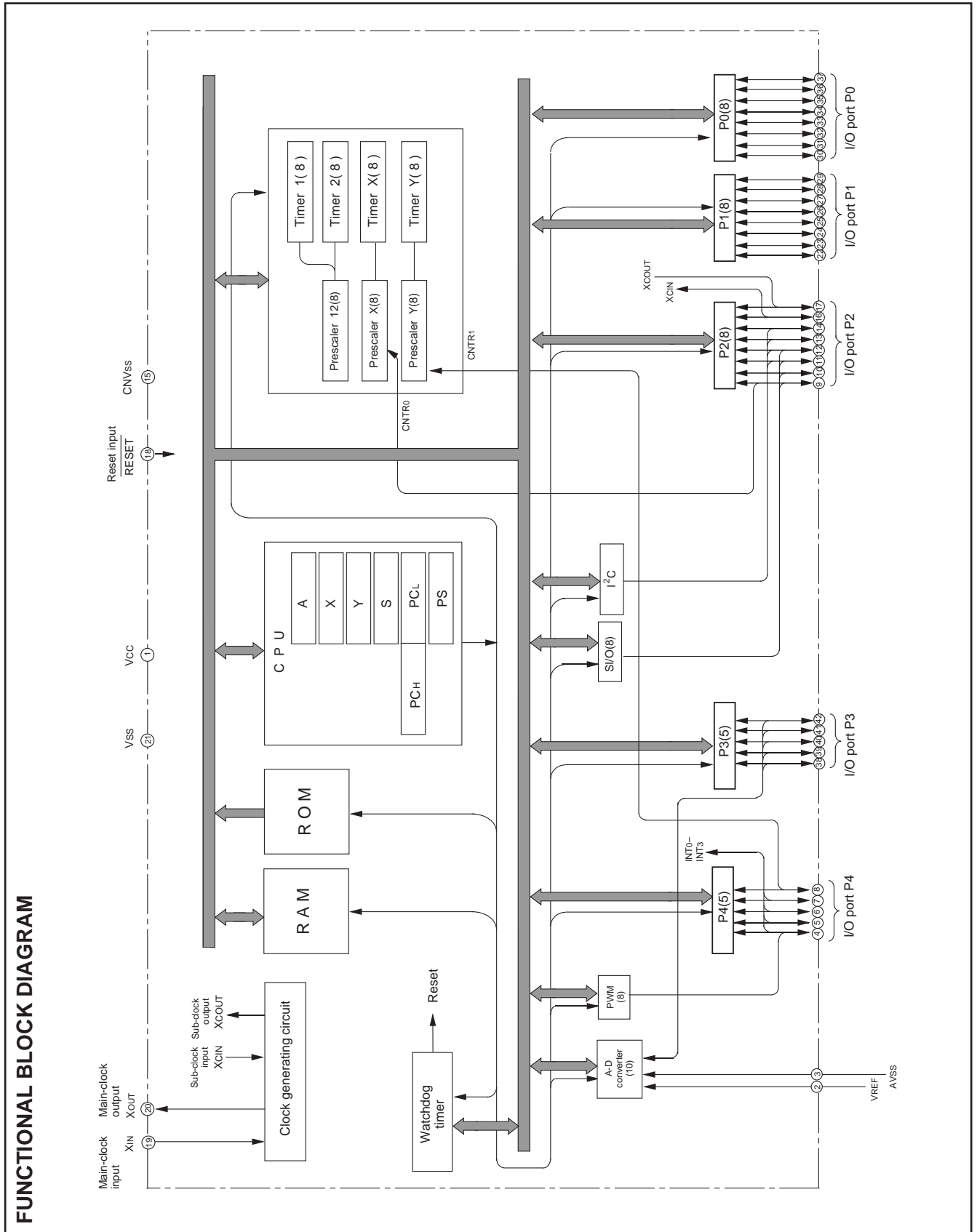


Fig. 2 Functional block diagram

HARDWARE

PIN DESCRIPTION

PIN DESCRIPTION

Table 1 Pin description

Pin	Name	Functions	
			Function except a port function
VCC, VSS	Power source	•Apply voltage of 2.7 V – 5.5 V to Vcc, and 0 V to Vss.	
CNVSS	CNVSS input	•This pin controls the operation mode of the chip. •Normally connected to Vss.	
$\overline{\text{RESET}}$	Reset input	•Reset input pin for active “L.”	
XIN	Clock input	•Input and output pins for the clock generating circuit. •Connect a ceramic resonator or quartz-crystal oscillator between the XIN and XOUT pins to set the oscillation frequency.	
XOUT	Clock output	•When an external clock is used, connect the clock source to the XIN pin and leave the XOUT pin open.	
P00–P07	I/O port P0	•8-bit CMOS I/O port. •I/O direction register allows each pin to be individually programmed as either input or output. •CMOS compatible input level.	
P10–P17	I/O port P1	•CMOS 3-state output structure. •P13 to P17 (5 bits) are enabled to output large current for LED drive (M38513E4/M4). •P10 to P17 (8 bits) are enabled to output large current for LED drive (M38514E6/M6).	
P20/XCOUT P21/XCIN	I/O port P2	•8-bit CMOS I/O port. •I/O direction register allows each pin to be individually programmed as either input or output. •CMOS compatible input level.	
P22/SDA1 P23/SCL1		•P22 to P25 can be switched between CMOS compatible input level or SMBUS input level in the I ² C-BUS interface function.	
P24/SDA2/RxD P25/SCL2/TxD		•P20, P21, P24 to P27: CMOS3-state output structure. •P24, P25: N-channel open-drain structure in the I ² C-BUS interface function. •P22, P23: N-channel open-drain structure.	
P26/SCLK		• Sub-clock generating circuit I/O pins (connect a resonator) • I ² C-BUS interface function pins	
P27/CNTR0/ SRDY		• I ² C-BUS interface function pin/ Serial I/O function pins • Serial I/O function pin • Serial I/O function pin/ Timer X function pin	
P30/AN0– P34/AN4	I/O port P3	•8-bit CMOS I/O port with the same function as port P0. •CMOS compatible input level. •CMOS 3-state output structure.	
P40/CNTR1	I/O port P4	•8-bit CMOS I/O port with the same function as port P0.	
P41/INT0– P43/INT2		•CMOS compatible input level. •CMOS 3-state output structure.	
P44/INT3/PWM		• Timer Y function pin • Interrupt input pins • Interrupt input pin • PWM output pin	

PART NUMBERING

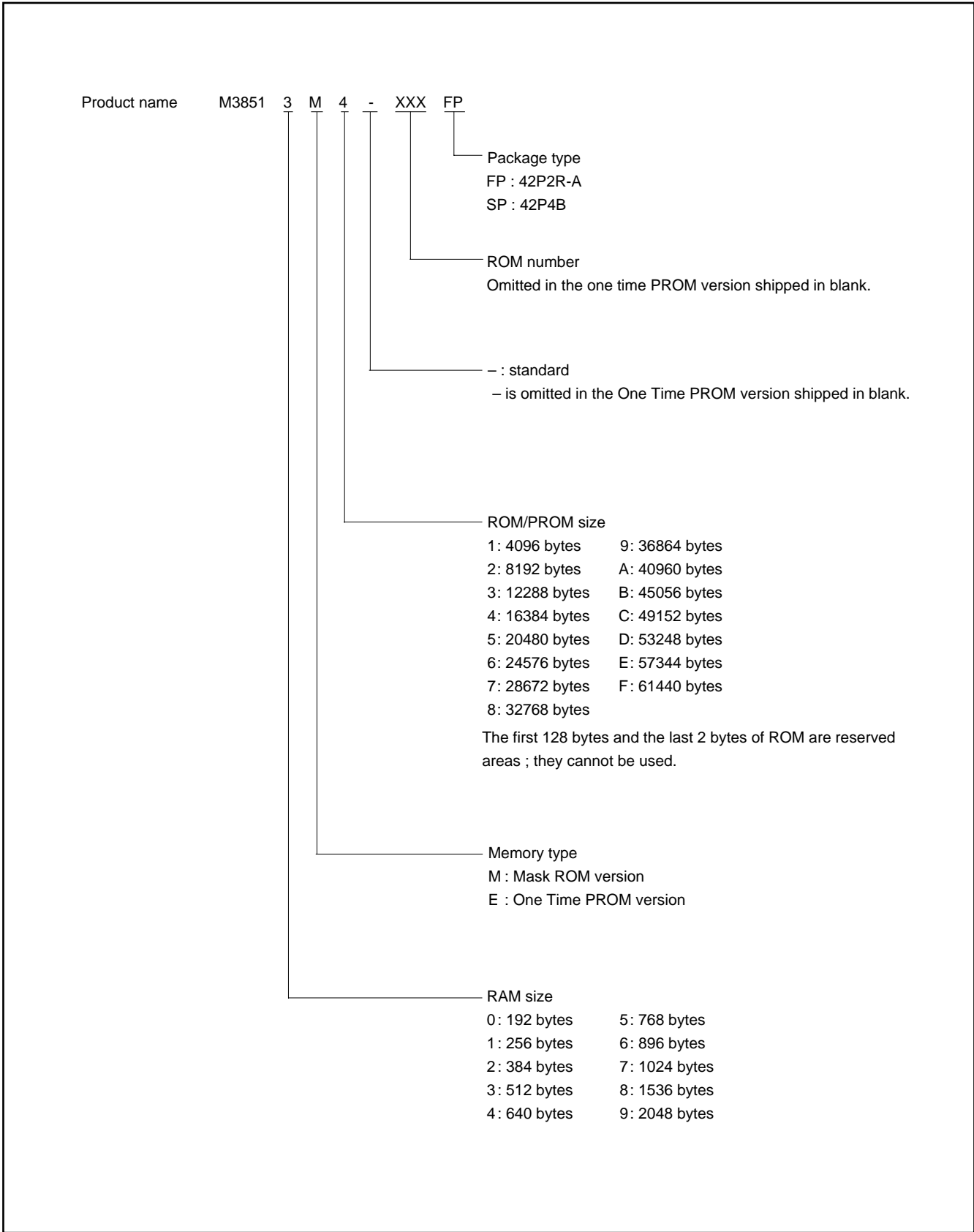


Fig. 3 Part numbering

HARDWARE

GROUP EXPANSION

GROUP EXPANSION

Mitsubishi plans to expand the 3851 group as follows:

Memory Type

Support for mask ROM and One Time PROM versions.

Memory Size

ROM size 16 K to 24 Kbytes

RAM size 512 to 640 bytes

Packages

42P2R-A 42-pin plastic molded SSOP

42P4B 42-pin shrink plastic-molded DIP

Memory Expansion Plan

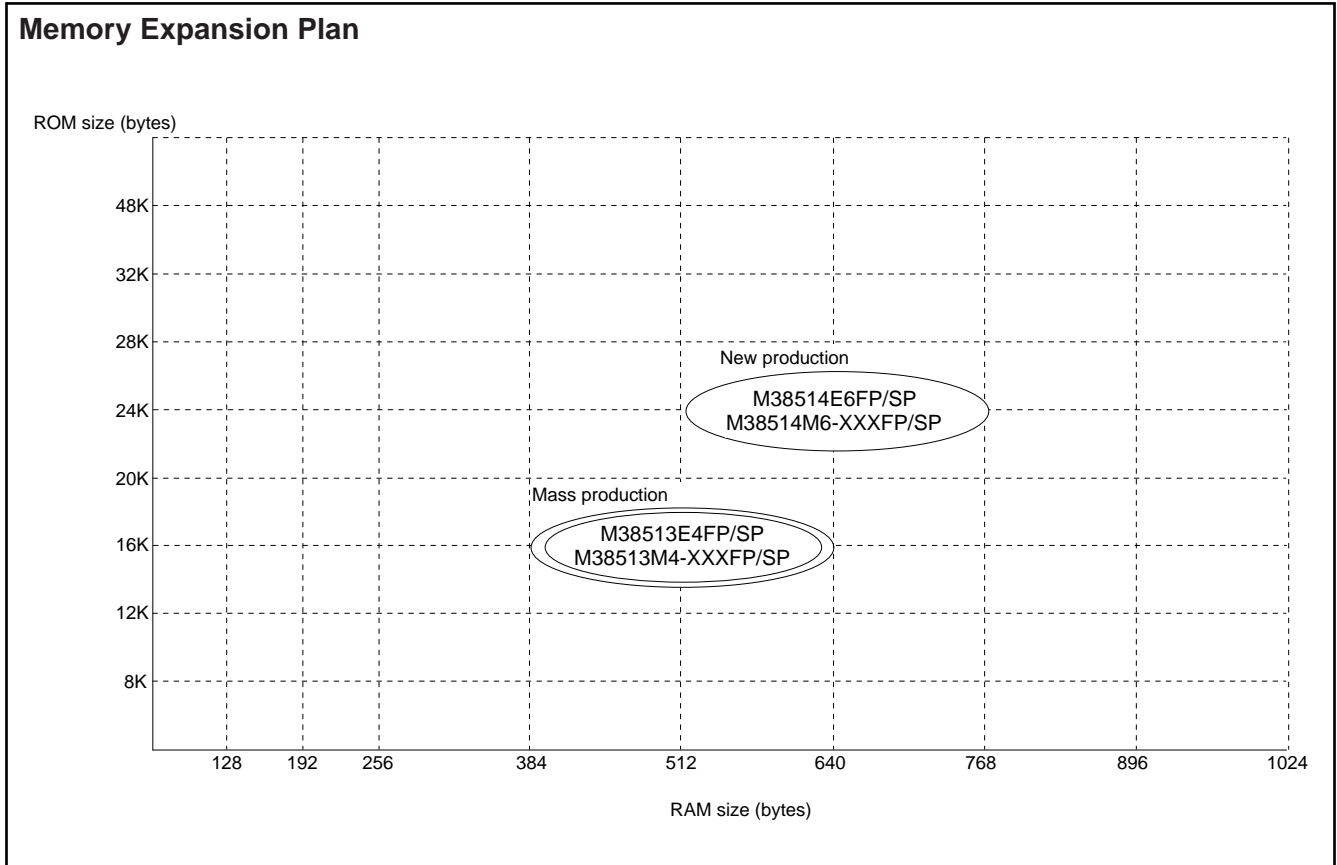


Fig. 4 Memory expansion plan

FUNCTIONAL DESCRIPTION Central Processing Unit (CPU)

The 3851 group uses the standard 740 Family instruction set. Refer to the table of 740 Family addressing modes and machine instructions or the 740 Series Software Manual for details on the instruction set.

Machine-resident 740 Family instructions are as follows:

The FST and SLW instructions cannot be used.

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

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

Accumulator (A)

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

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

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

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

Stack pointer (S)

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

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

The Stack Page Selection Bit is located in the SFR area in the zero page. Note that the initial value of the Stack Page Selection Bit varies with each microcomputer type. Also some microcomputer types have no Stack Page Selection Bit and the upper eight bits of the stack address are fixed. The operations of pushing register contents onto the stack and popping them from the stack are shown in Figure 8.

Program counter (PC)

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

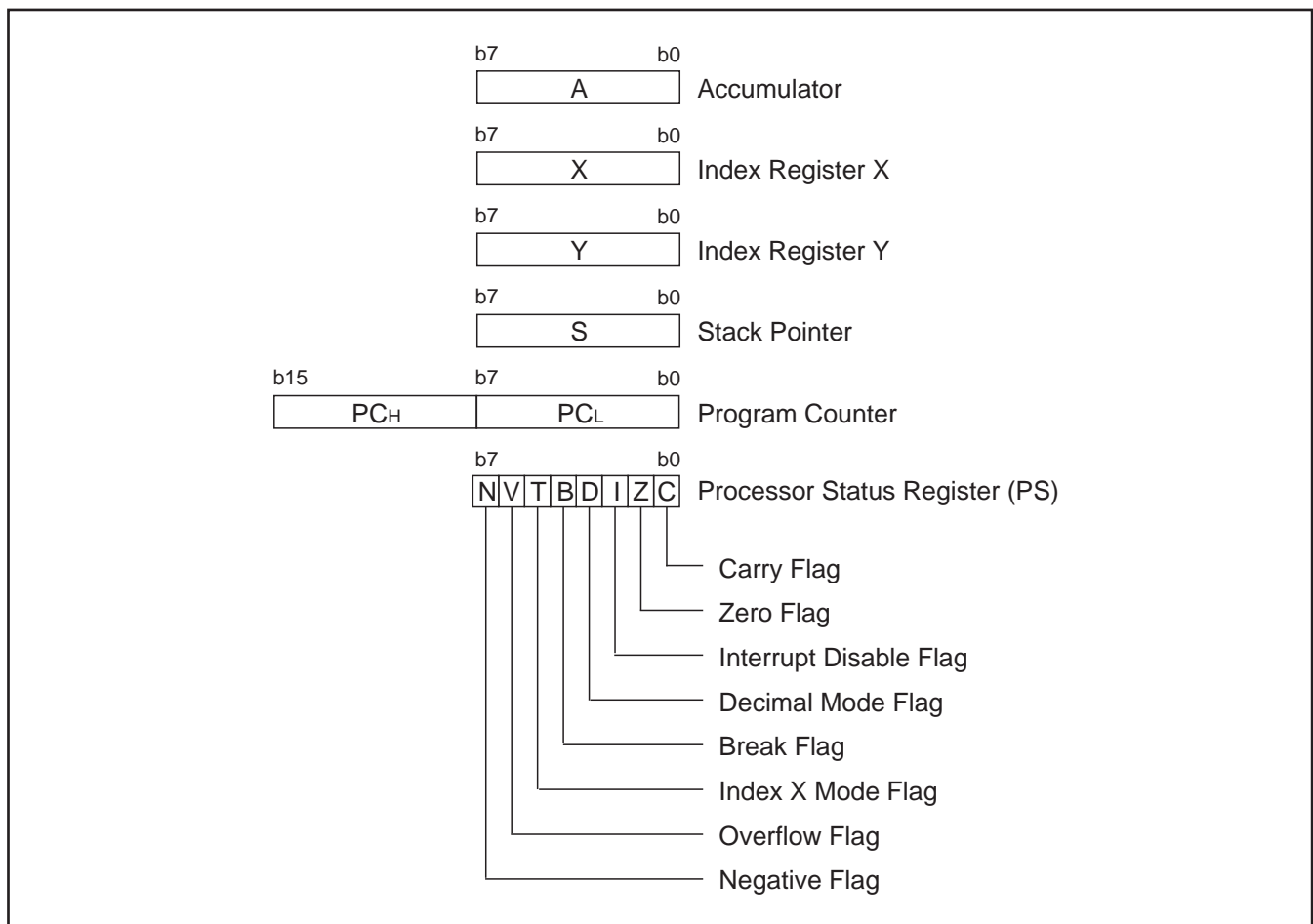


Fig. 5 740 Family CPU register structure

HARDWARE

FUNCTIONAL DESCRIPTION

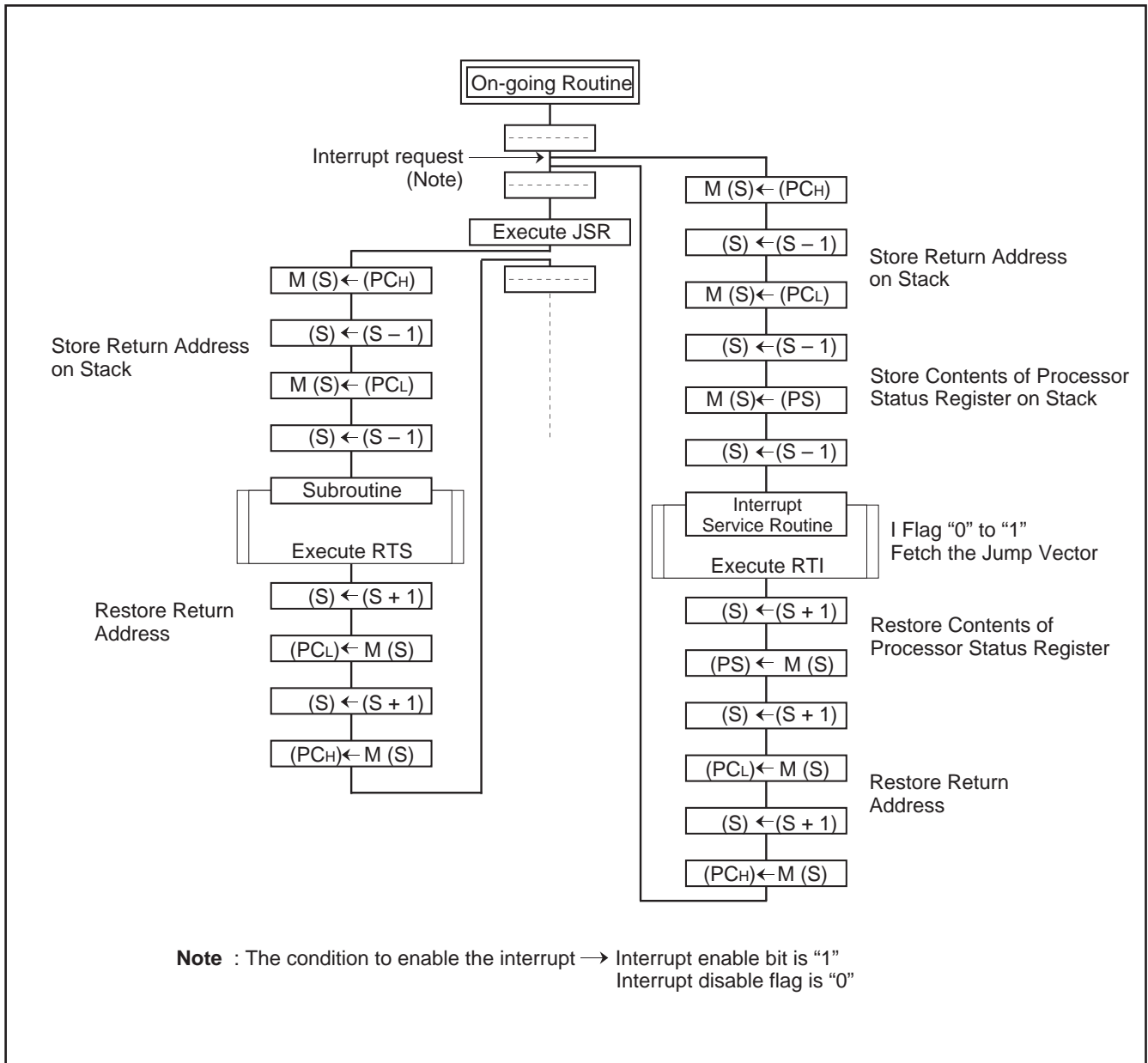


Fig. 6 Register push and pop at interrupt generation and subroutine call

Table 2 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 3 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	–

HARDWARE

FUNCTIONAL DESCRIPTION

[CPU Mode Register (CPUM)] 003B16

The CPU mode register contains the stack page selection bit, etc.

The CPU mode register is allocated at address 003B16.

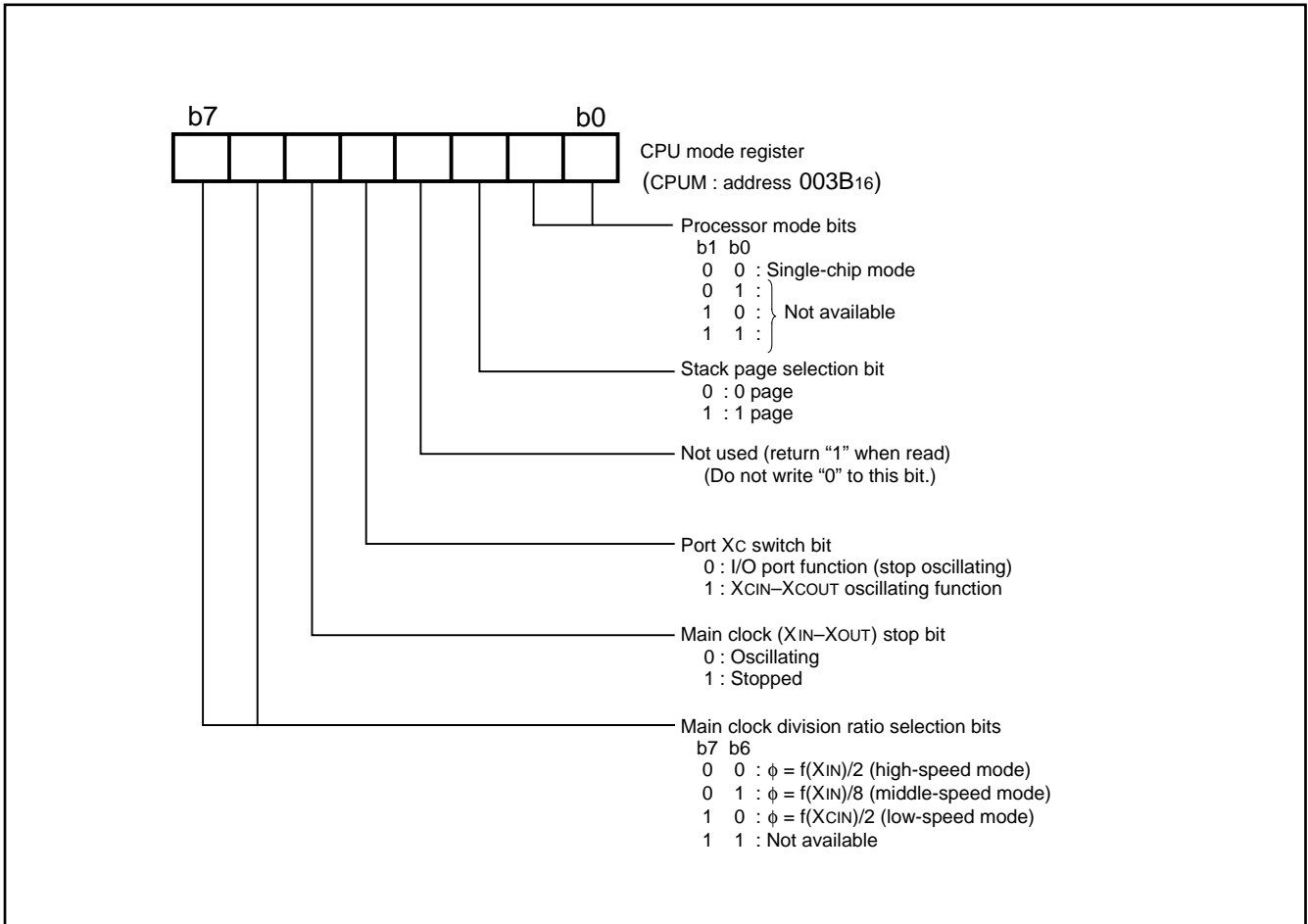


Fig. 7 Structure of CPU mode register

MEMORY

Special Function Register (SFR) Area

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

RAM

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

ROM

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

Interrupt Vector Area

The interrupt vector area contains reset and interrupt vectors.

Zero Page

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

Special Page

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

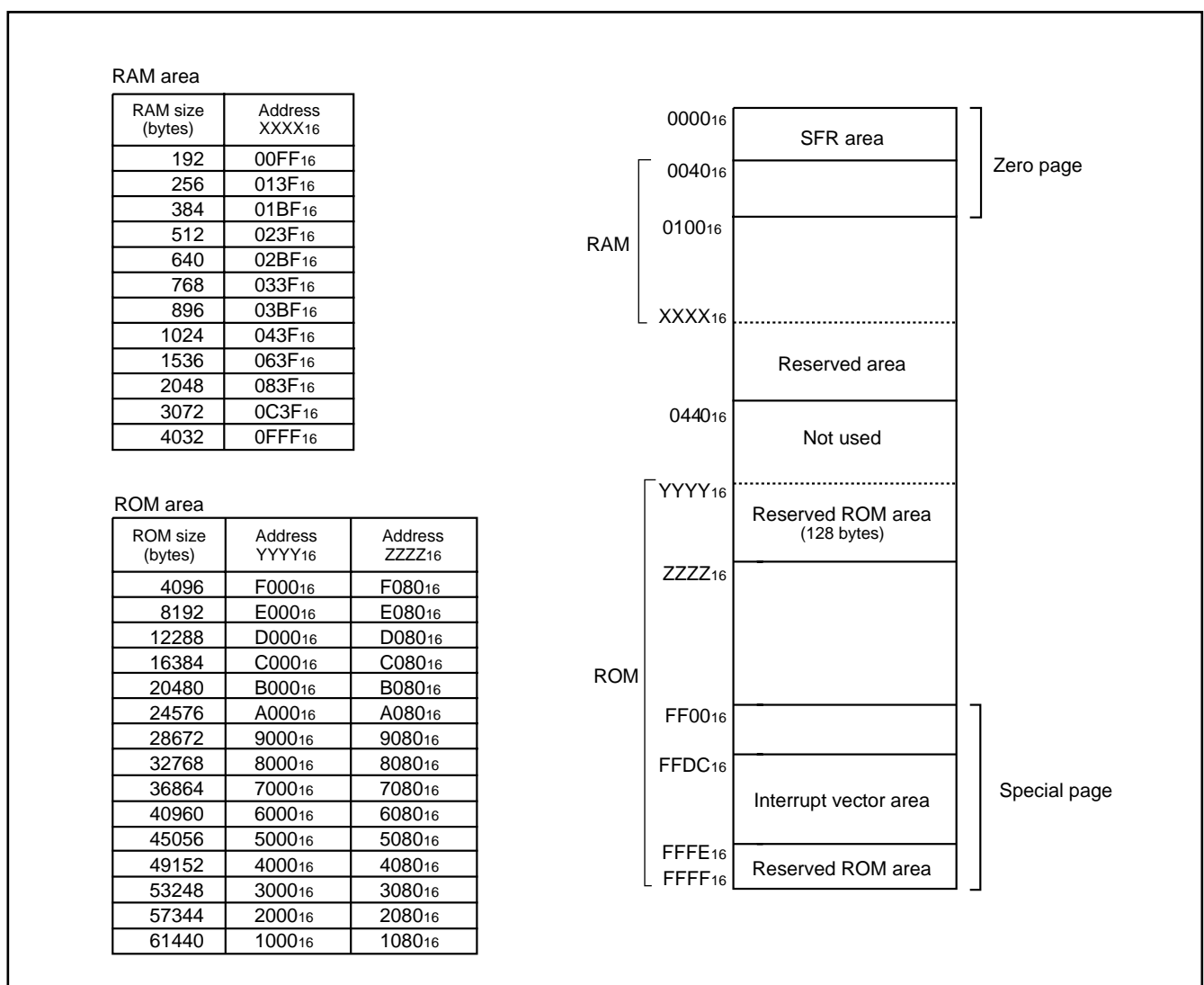


Fig. 8 Memory map diagram

HARDWARE

FUNCTIONAL DESCRIPTION

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	Timer count source selection register (TCSS)
0009 ₁₆	Port P4 direction register (P4D)	0029 ₁₆	
000A ₁₆		002A ₁₆	
000B ₁₆		002B ₁₆	I ² C data shift register (S0)
000C ₁₆		002C ₁₆	I ² C address register (S0D)
000D ₁₆		002D ₁₆	I ² C status register (S1)
000E ₁₆		002E ₁₆	I ² C control register (S1D)
000F ₁₆		002F ₁₆	I ² C clock control register (S2)
0010 ₁₆		0030 ₁₆	I ² C start/stop condition control register (S2D)
0011 ₁₆		0031 ₁₆	Reserved *
0012 ₁₆		0032 ₁₆	
0013 ₁₆		0033 ₁₆	
0014 ₁₆		0034 ₁₆	A-D control register (ADCON)
0015 ₁₆	Reserved *	0035 ₁₆	A-D conversion low-order register (ADL)
0016 ₁₆	Reserved *	0036 ₁₆	A-D conversion high-order register (ADH)
0017 ₁₆	Reserved *	0037 ₁₆	
0018 ₁₆	Transmit/Receive buffer register (TB/RB)	0038 ₁₆	MISRG
0019 ₁₆	Serial I/O status register (SIOSTS)	0039 ₁₆	Watchdog timer control register (WDTCON)
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	PWM control register (PWMCON)	003D ₁₆	Interrupt request register 2 (IREQ2)
001E ₁₆	PWM prescaler (PREPWM)	003E ₁₆	Interrupt control register 1 (ICON1)
001F ₁₆	PWM register (PWM)	003F ₁₆	Interrupt control register 2 (ICON2)

* Reserved : Do not write "1" to this address.

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

I/O PORTS

The I/O ports have direction registers which determine the input/output direction of each individual pin. Each bit in a direction register corresponds to one pin, and each pin can be set to be input port or output port.

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

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

Table 4 I/O port function

Pin	Name	Input/Output	I/O Structure	Non-Port Function	Related SFRs	Ref.No.
P00–P07	Port P0	Input/output, individual bits	CMOS compatible input level CMOS 3-state output			(1)
P10–P17	Port P1			Sub-clock generating circuit	CPU mode register	(2) (3)
P20/XCOUT P21/XCIN	Port P2		CMOS compatible input level CMOS/SMBUS input level (when selecting I ² C-BUS interface function) N-channel open-drain output	I ² C-BUS interface function I/O	I ² C control register	(4) (5)
P22/SDA1 P23/SCL1			CMOS compatible input level CMOS/SMBUS input level (when selecting I ² C-BUS interface function) CMOS 3-state output N-channel open-drain output (when selecting I ² C-BUS interface function)	I ² C-BUS interface function I/O Serial I/O function I/O	I ² C control register Serial I/O control register	(6) (7)
P24/SDA2/RxD P25/SCL2/TxD			Serial I/O function I/O	Serial I/O control register	(8)	
P26/SCLK			Serial I/O function I/O Timer X function I/O	Serial I/O control register Timer XY mode register	(9)	
P27/CNTR0/ $\overline{\text{SRDY}}$			A-D conversion input	A-D control register	(10)	
P30/AN0— P34/AN4			Port P3	Timer Y function I/O	Timer XY mode register	(11)
P40/CNTR1			Port P4	External interrupt input	Interrupt edge selection register	(12)
P41/INT0— P43/INT2	External interrupt input PWM output			Interrupt edge selection register PWM control register	(13)	
P44/INT3/PWM						

HARDWARE

FUNCTIONAL DESCRIPTION

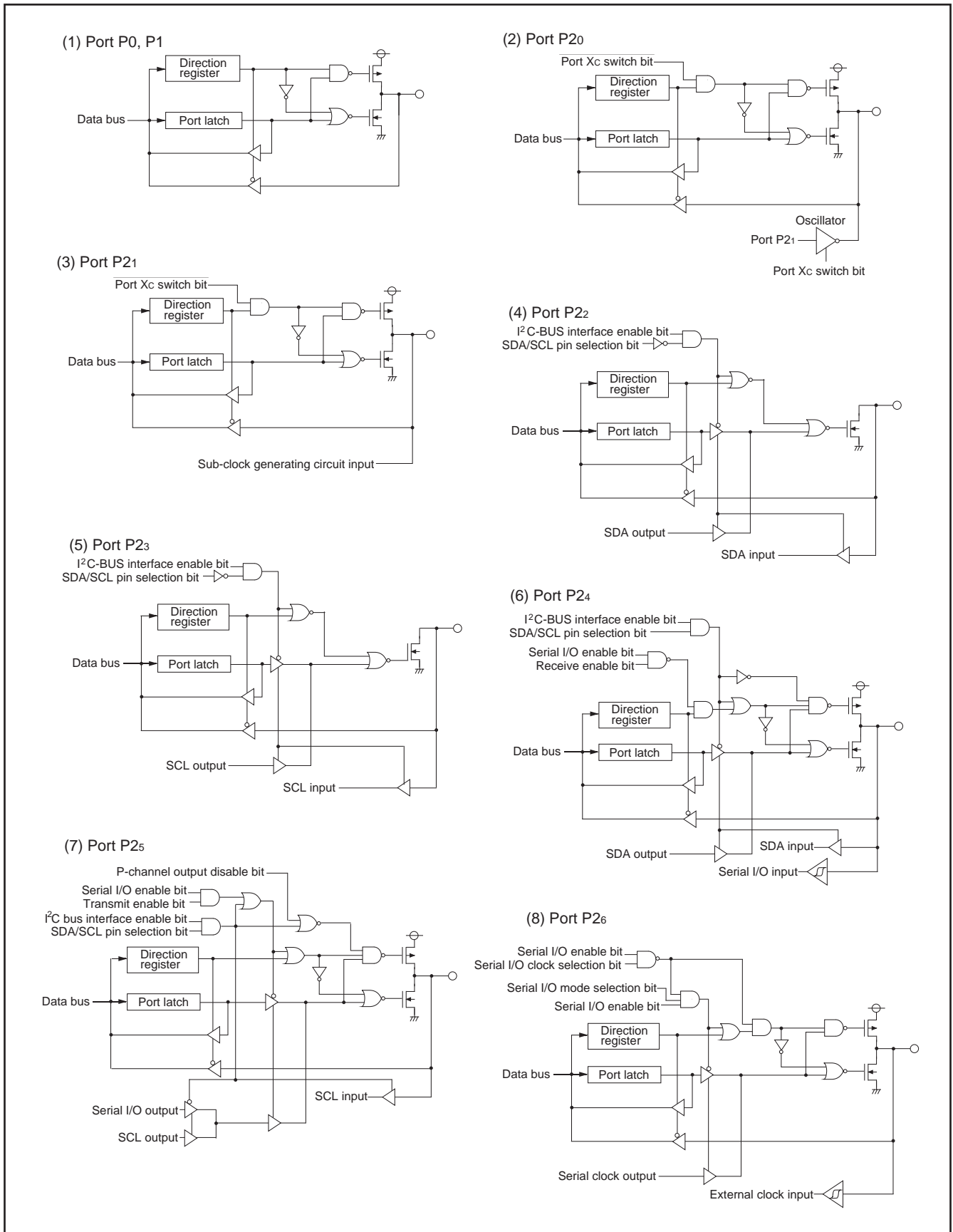


Fig. 10 Port block diagram (1)

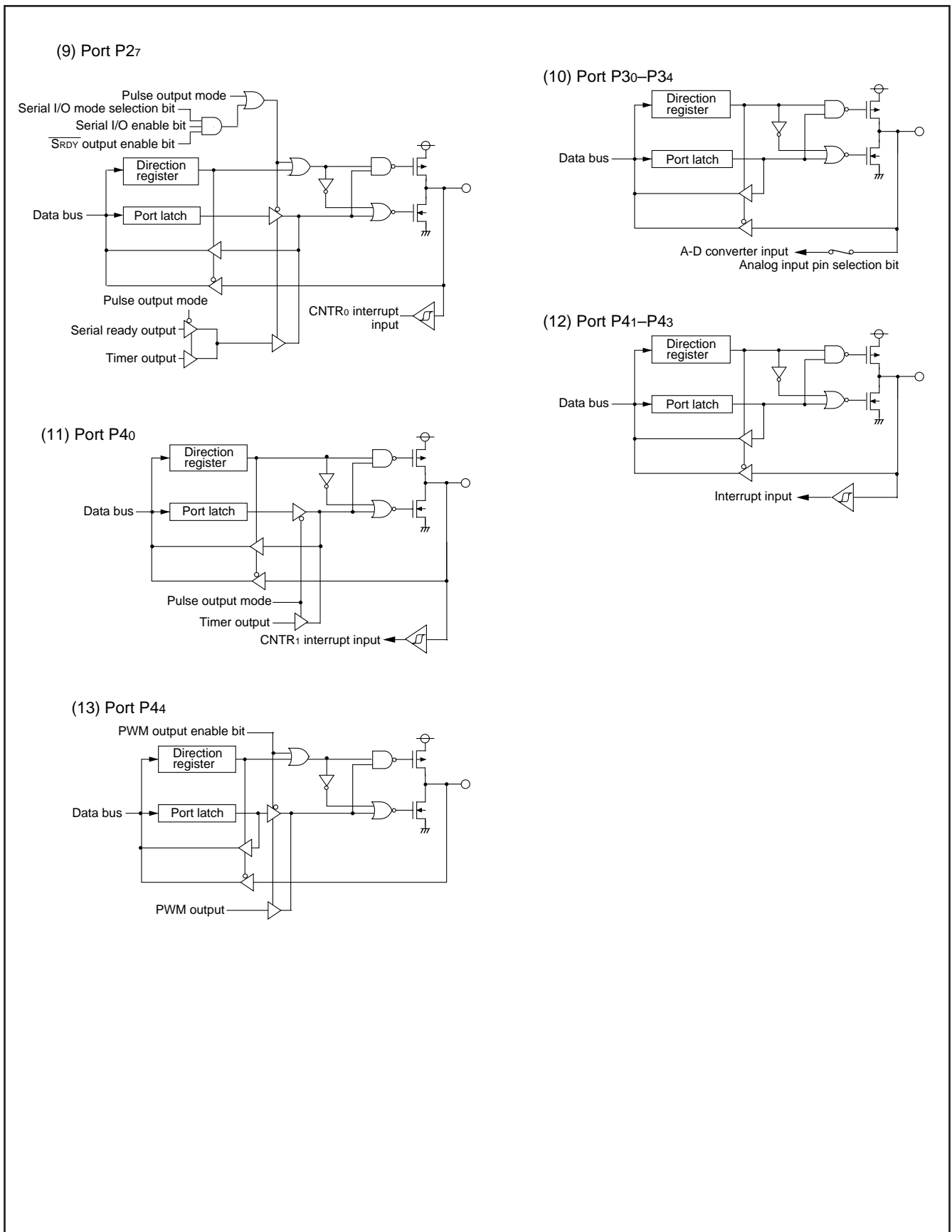


Fig. 11 Port block diagram (2)

HARDWARE

FUNCTIONAL DESCRIPTION

INTERRUPTS

Interrupts occur by 16 sources among 16 sources: seven external, eight internal, and one software.

Interrupt Control

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

Interrupt enable bits can be set or cleared by software.

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

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

When several interrupts occur at the same time, the interrupts are received according to priority.

Interrupt Operation

By acceptance of an interrupt, the following operations are automatically performed:

1. The contents of the program counter and the processor status register are automatically pushed onto the stack.
2. The interrupt disable flag is set and the corresponding interrupt request bit is cleared.
3. The interrupt jump destination address is read from the vector table into the program counter.

■Notes

When the active edge of an external interrupt (INT₀–INT₃, SCL/SDA, CNTR₀, CNTR₁) is set, the corresponding interrupt request bit may also be set. Therefore, take the following sequence:

1. Disable the interrupt
2. Change the interrupt edge selection register (SCL/SDA interrupt pin polarity selection bit for SCL/SDA; the timer XY mode register for CNTR₀ and CNTR₁)
3. Clear the interrupt request bit to "0"
4. Accept the interrupt.

Table 5 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)
SCL, SDA	3	FFF9 ₁₆	FFF8 ₁₆	At detection of either rising or falling edge of SCL or SDA input	External interrupt (active edge selectable)
INT ₁	4	FFF7 ₁₆	FFF6 ₁₆	At detection of either rising or falling edge of INT ₁ input	External interrupt (active edge selectable)
INT ₂	5	FFF5 ₁₆	FFF4 ₁₆	At detection of either rising or falling edge of INT ₂ input	External interrupt (active edge selectable)
INT ₃	6	FFF3 ₁₆	FFF2 ₁₆	At detection of either rising or falling edge of INT ₃ input	External interrupt (active edge selectable)
I ² C	7	FFF1 ₁₆	FFF0 ₁₆	At completion of data transfer	
Timer X	8	FFEF ₁₆	FFEE ₁₆	At timer X underflow	
Timer Y	9	FFED ₁₆	FFEC ₁₆	At timer Y underflow	
Timer 1	10	FFEB ₁₆	FFEA ₁₆	At timer 1 underflow	STP release timer underflow
Timer 2	11	FFE9 ₁₆	FFE8 ₁₆	At timer 2 underflow	
Serial I/O reception	12	FFE7 ₁₆	FFE6 ₁₆	At completion of serial I/O data reception	Valid when serial I/O is selected
Serial I/O Transmission	13	FFE5 ₁₆	FFE4 ₁₆	At completion of serial I/O transfer shift or when transmission buffer is empty	Valid when serial I/O is selected
CNTR ₀	14	FFE3 ₁₆	FFE2 ₁₆	At detection of either rising or falling edge of CNTR ₀ input	External interrupt (active edge selectable)
CNTR ₁	15	FFE1 ₁₆	FFE0 ₁₆	At detection of either rising or falling edge of CNTR ₁ input	External interrupt (active edge selectable)
A-D converter	16	FFDF ₁₆	FFDE ₁₆	At completion of A-D conversion	
BRK instruction	17	FFDD ₁₆	FFDC ₁₆	At BRK instruction execution	Non-maskable software interrupt

Notes 1: Vector addresses contain interrupt jump destination addresses.

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

HARDWARE

FUNCTIONAL DESCRIPTION

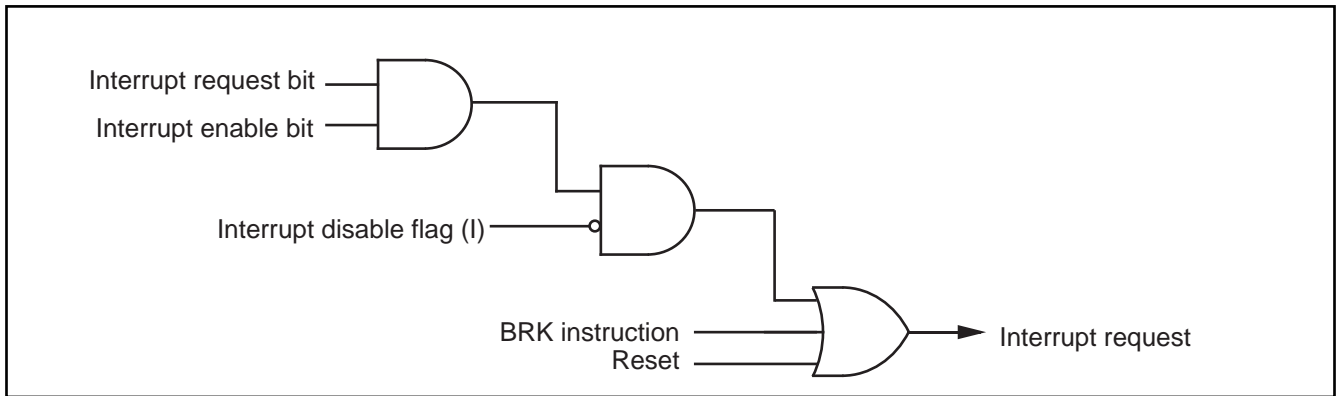


Fig. 12 Interrupt control

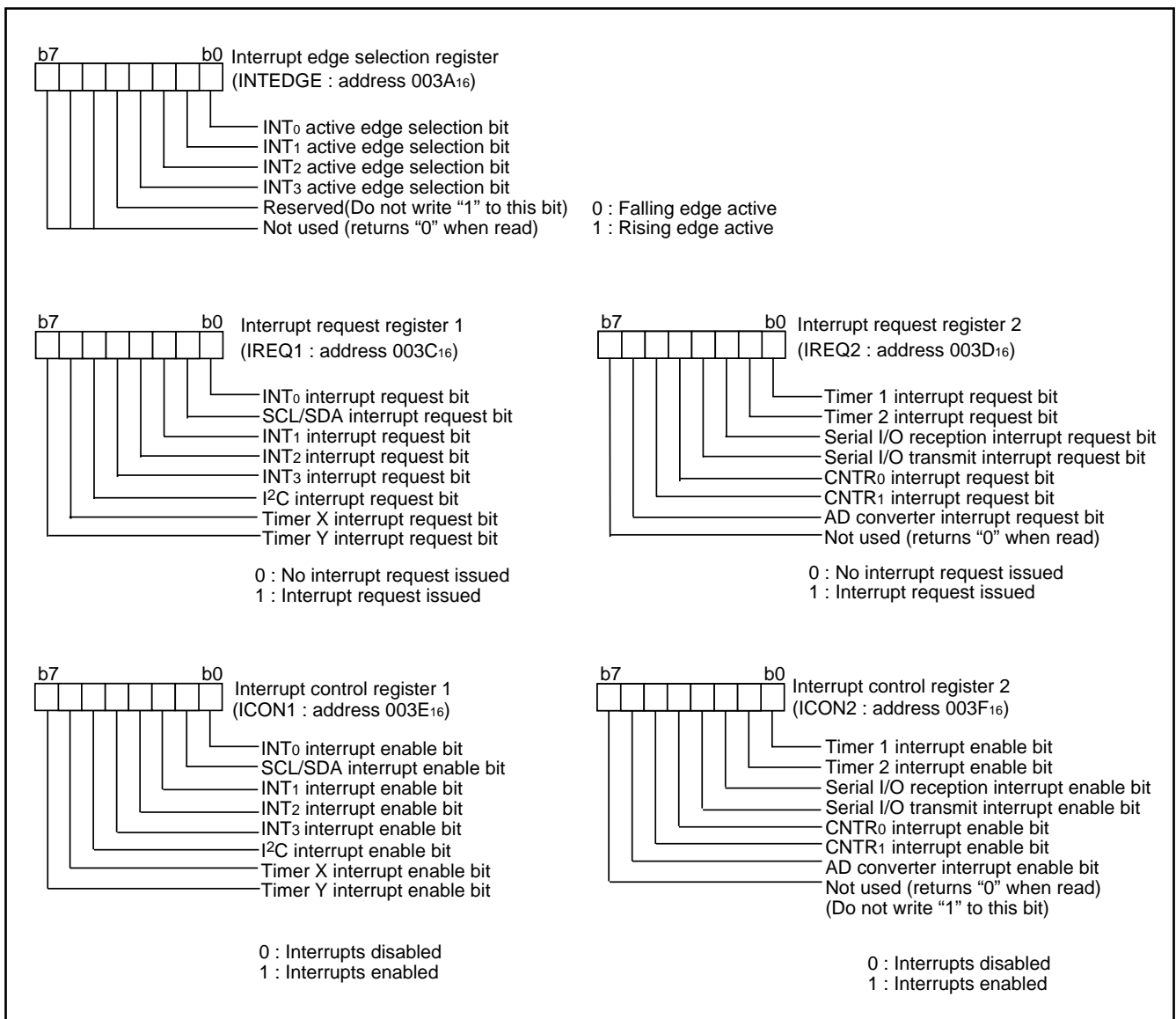


Fig. 13 Structure of interrupt-related registers (1)

TIMERS

The 3851 group has four timers: timer X, timer Y, timer 1, and timer 2.

The division ratio of each timer or prescaler is given by $1/(n + 1)$, where n is the value in the corresponding timer or prescaler latch. All timers are count down. When the timer reaches "0016", an underflow occurs at the next count pulse and the corresponding timer latch is reloaded into the timer and the count is continued. When a timer underflows, the interrupt request bit corresponding to that timer is set to "1".

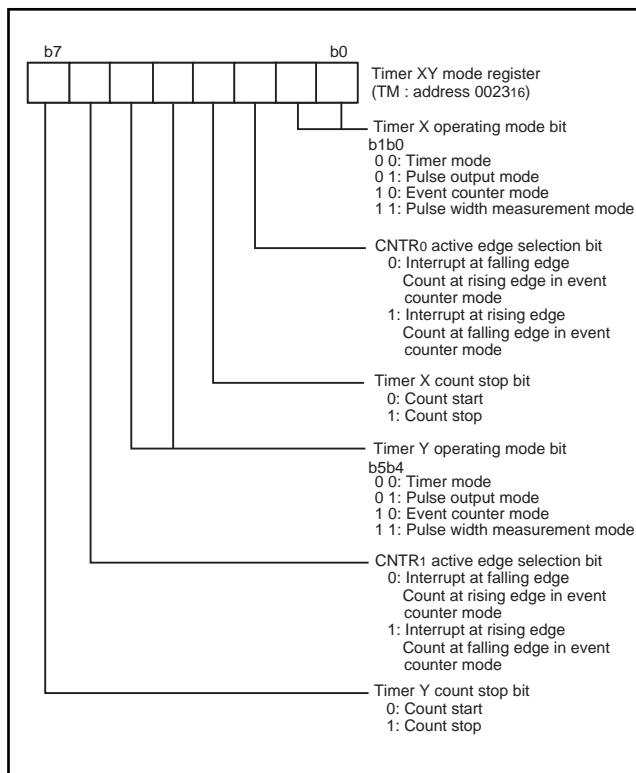


Fig. 14 Structure of timer XY mode register

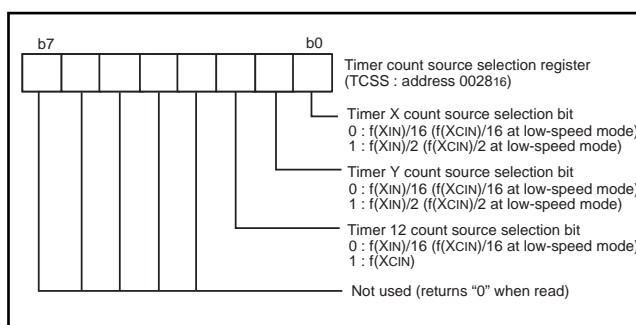


Fig. 15 Structure of timer count source selection register

Timer 1 and Timer 2

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

Timer X and Timer Y

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

(1) Timer Mode

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

(2) Pulse Output Mode

The timer counts the count source selected by Timer count source selection bit. Whenever the contents of the timer reach "0016", the signal output from the CNTR0 (or CNTR1) pin is inverted. If the CNTR0 (or CNTR1) active edge selection bit is "0", output begins at "H".

If it is "1", output starts at "L". When using a timer in this mode, set the corresponding port P27 (or port P40) direction register to output mode.

(3) Event Counter Mode

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

When the CNTR0 (or CNTR1) active edge selection bit is "0", the rising edge of the CNTR0 (or CNTR1) pin is counted.

When the CNTR0 (or CNTR1) active edge selection bit is "1", the falling edge of the CNTR0 (or CNTR1) pin is counted.

(4) Pulse Width Measurement Mode

If the CNTR0 (or CNTR1) active edge selection bit is "0", the timer counts the selected signals by the count source selection bit while the CNTR0 (or CNTR1) pin is at "H". If the CNTR0 (or CNTR1) active edge selection bit is "1", the timer counts it while the CNTR0 (or CNTR1) pin is at "L".

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

■Note

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

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

HARDWARE

FUNCTIONAL DESCRIPTION

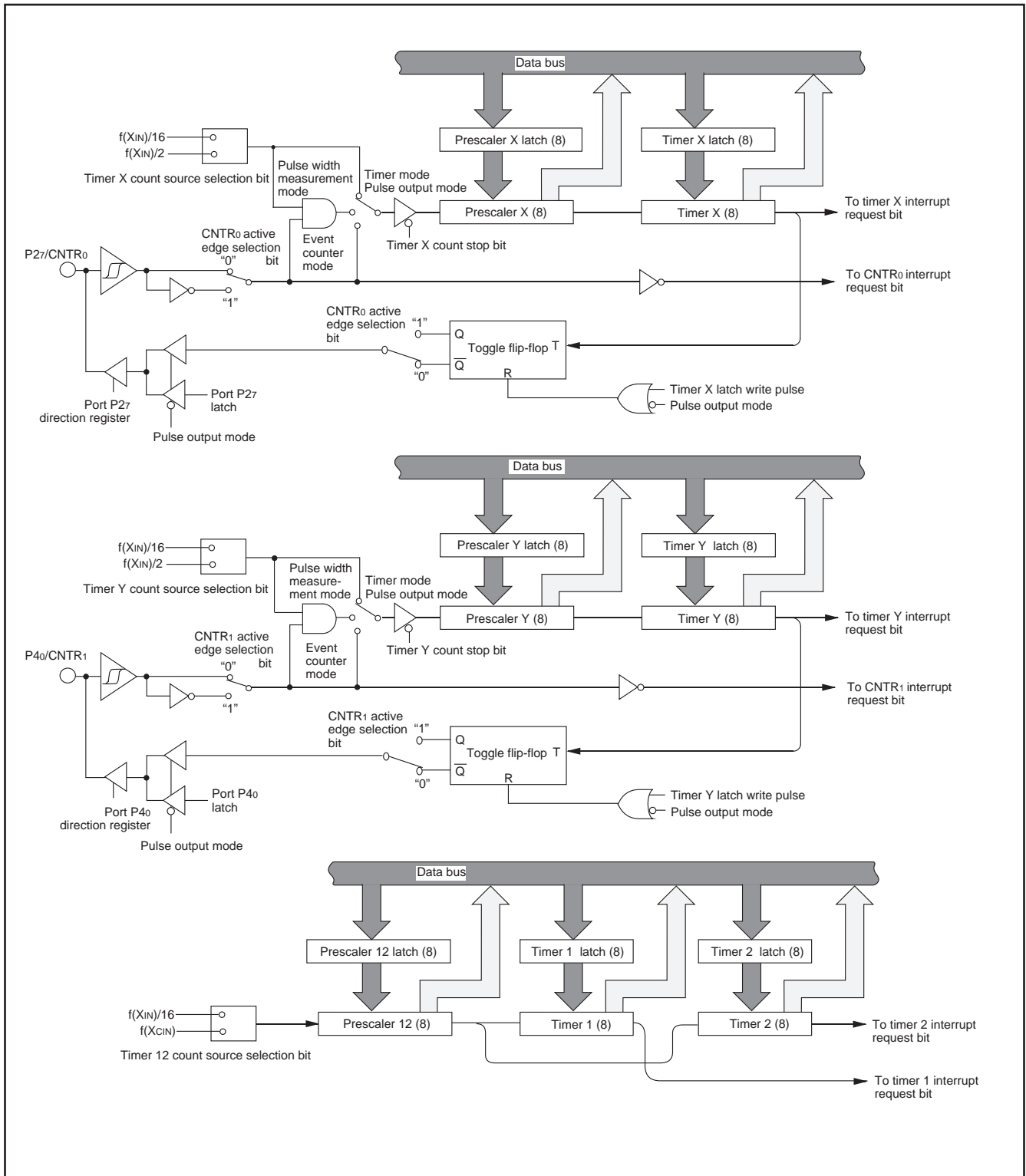


Fig. 16 Block diagram of timer X, timer Y, timer 1, and timer 2

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.

(1) Clock Synchronous Serial I/O Mode

Clock synchronous serial I/O mode can be selected by setting the serial I/O mode selection bit of the serial I/O control register (bit 6 of address 001A16) to "1".

For clock synchronous serial I/O, the transmitter and the receiver must use the same clock. If an internal clock is used, transfer is started by a write signal to the TB/RB.

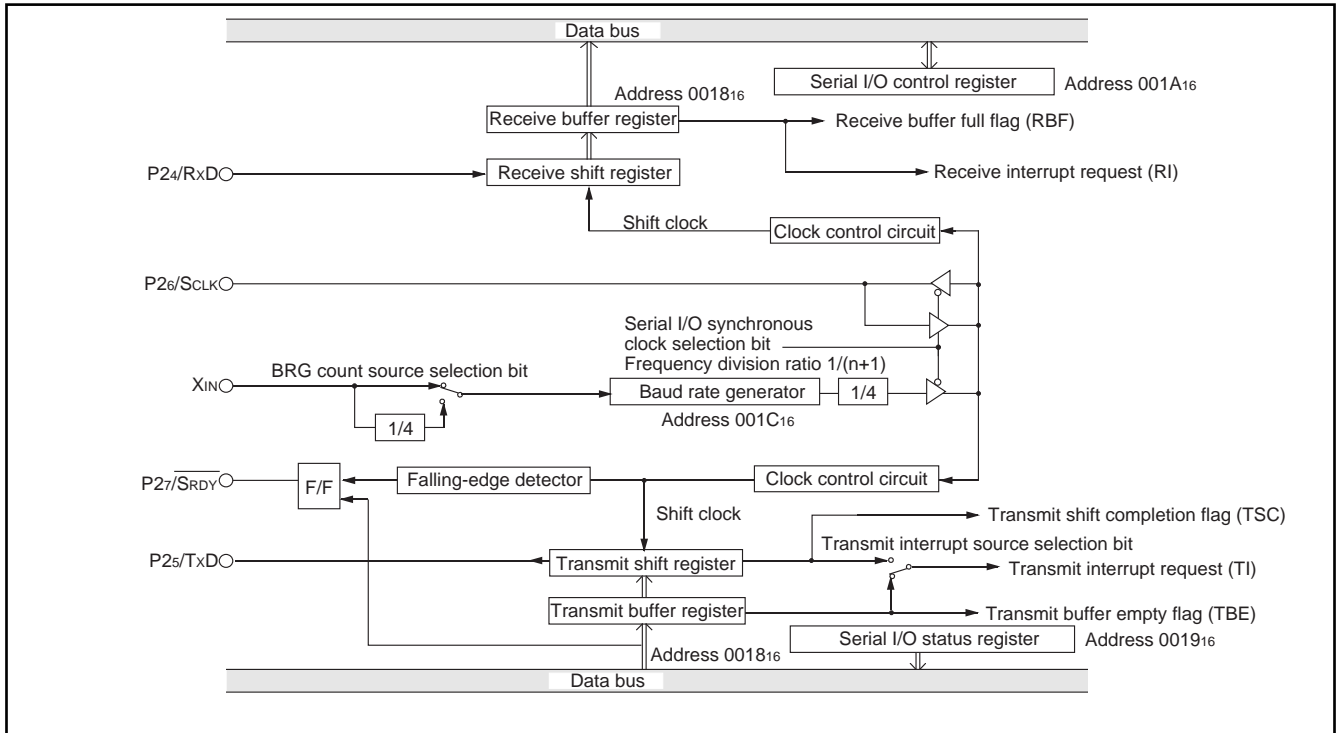


Fig. 17 Block diagram of clock synchronous serial I/O

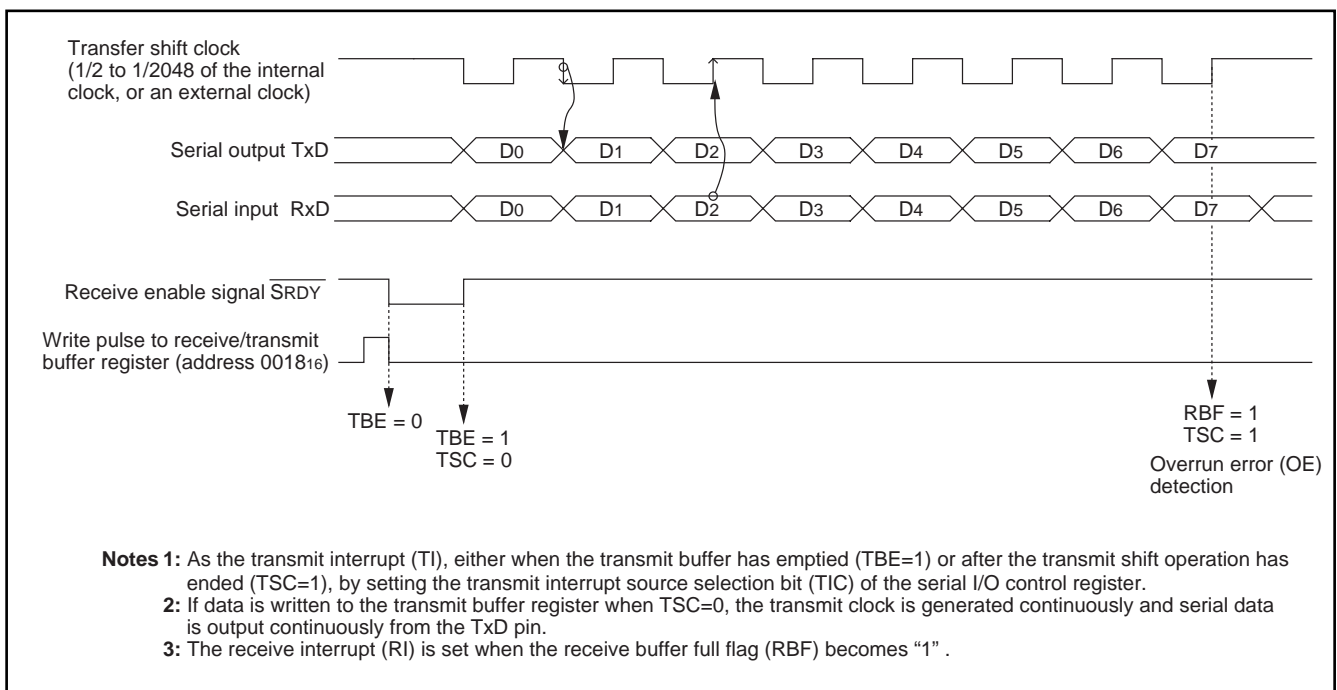


Fig. 18 Operation of clock synchronous serial I/O function

HARDWARE

FUNCTIONAL DESCRIPTION

(2) 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 (b6) of the serial I/O control register to "0".

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

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

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

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

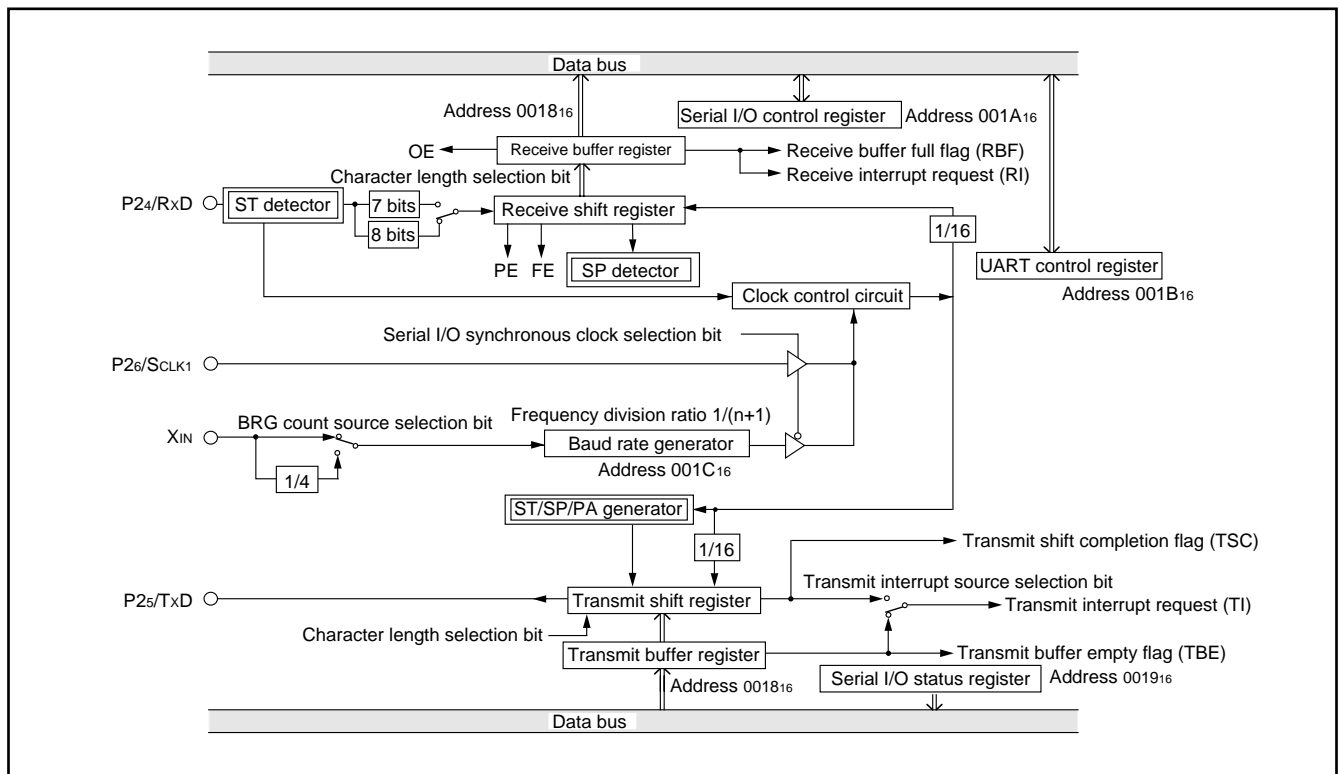


Fig.19 Block diagram of UART serial I/O

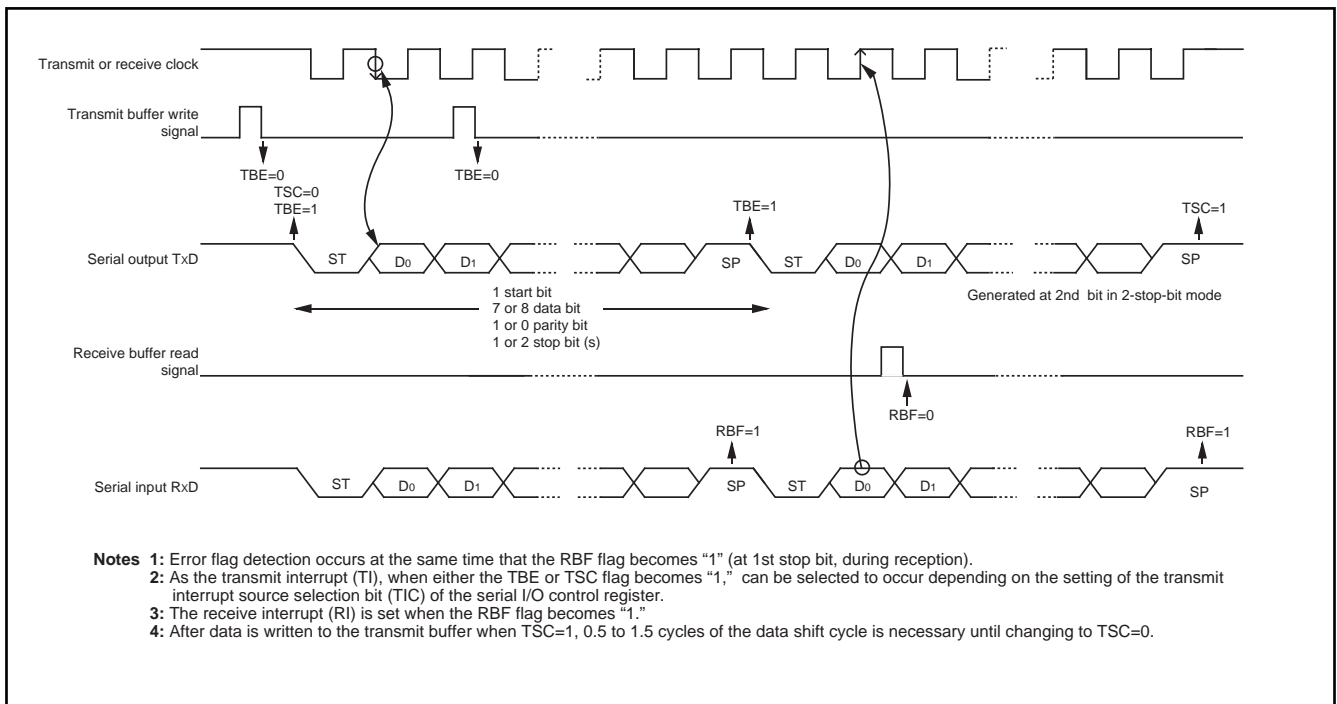


Fig. 20 Operation of UART serial I/O function

[Transmit Buffer Register/Receive Buffer Register (TB/RB)] 001816

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

[Serial I/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 register is read.

If there is an error, it is detected at the same time that data is transferred from the receive shift register to the receive buffer register, and the receive buffer full flag is set. A write to the serial I/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.

Bits 0 to 6 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".

[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 and one bit (bit 4) which is always valid and sets the output structure of the P25/TXD pin.

[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.

■Note

When using the serial I/O, clear the I²C-BUS interface enable bit to "0" or the SDA/SCL pin selection bit to "0".

HARDWARE

FUNCTIONAL DESCRIPTION

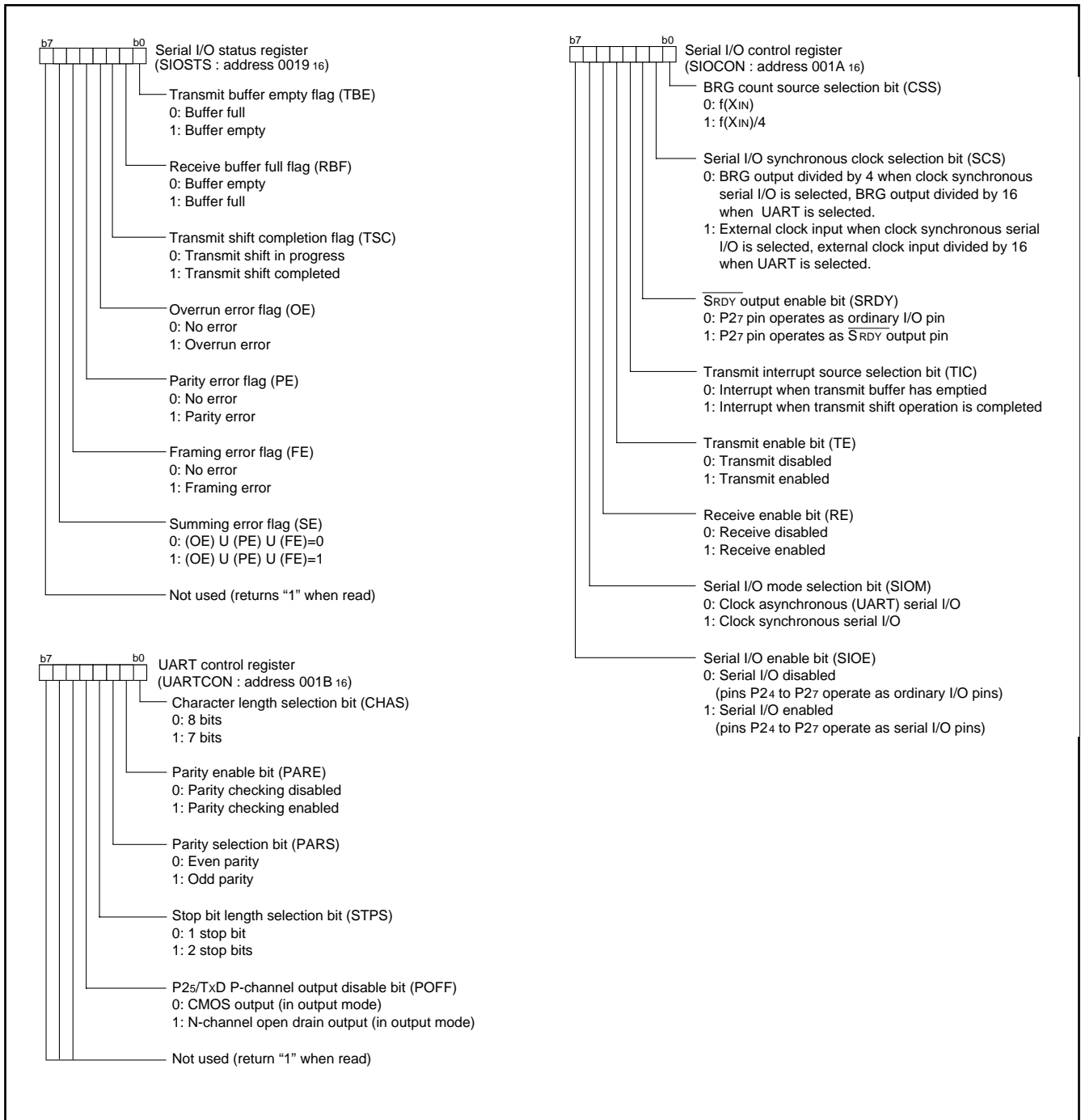


Fig. 21 Structure of serial I/O control registers

MULTI-MASTER I²C-BUS INTERFACE

The multi-master I²C-BUS interface is a serial communications circuit, conforming to the Philips I²C-BUS data transfer format. This interface, offering both arbitration lost detection and a synchronous functions, is useful for the multi-master serial communications.

Figure 19 shows a block diagram of the multi-master I²C-BUS interface and Table 4 lists the multi-master I²C-BUS interface functions.

This multi-master I²C-BUS interface consists of the I²C address register, the I²C data shift register, the I²C clock control register, the I²C control register, the I²C status register, the I²C start/stop condition control register and other control circuits.

When using the multi-master I²C-BUS interface, set 1 MHz or more to ϕ .

Note: Mitsubishi Electric Corporation assumes no responsibility for infringement of any third-party's rights or originating in the use of the connection control function between the I²C-BUS interface and the ports SCL1, SCL2, SDA1 and SDA2 with the bit 6 of I²C control register (002E16).

Table 6 Multi-master I²C-BUS interface functions

Item	Function
Format	In conformity with Philips I ² C-BUS standard: 10-bit addressing format 7-bit addressing format High-speed clock mode Standard clock mode
Communication mode	In conformity with Philips I ² C-BUS standard: Master transmission Master reception Slave transmission Slave reception
SCL clock frequency	16.1 kHz to 400 kHz (at $\phi = 4$ MHz)

System clock $\phi = f(XIN)/2$ (high-speed mode)

$\phi = f(XIN)/8$ (middle-speed mode)

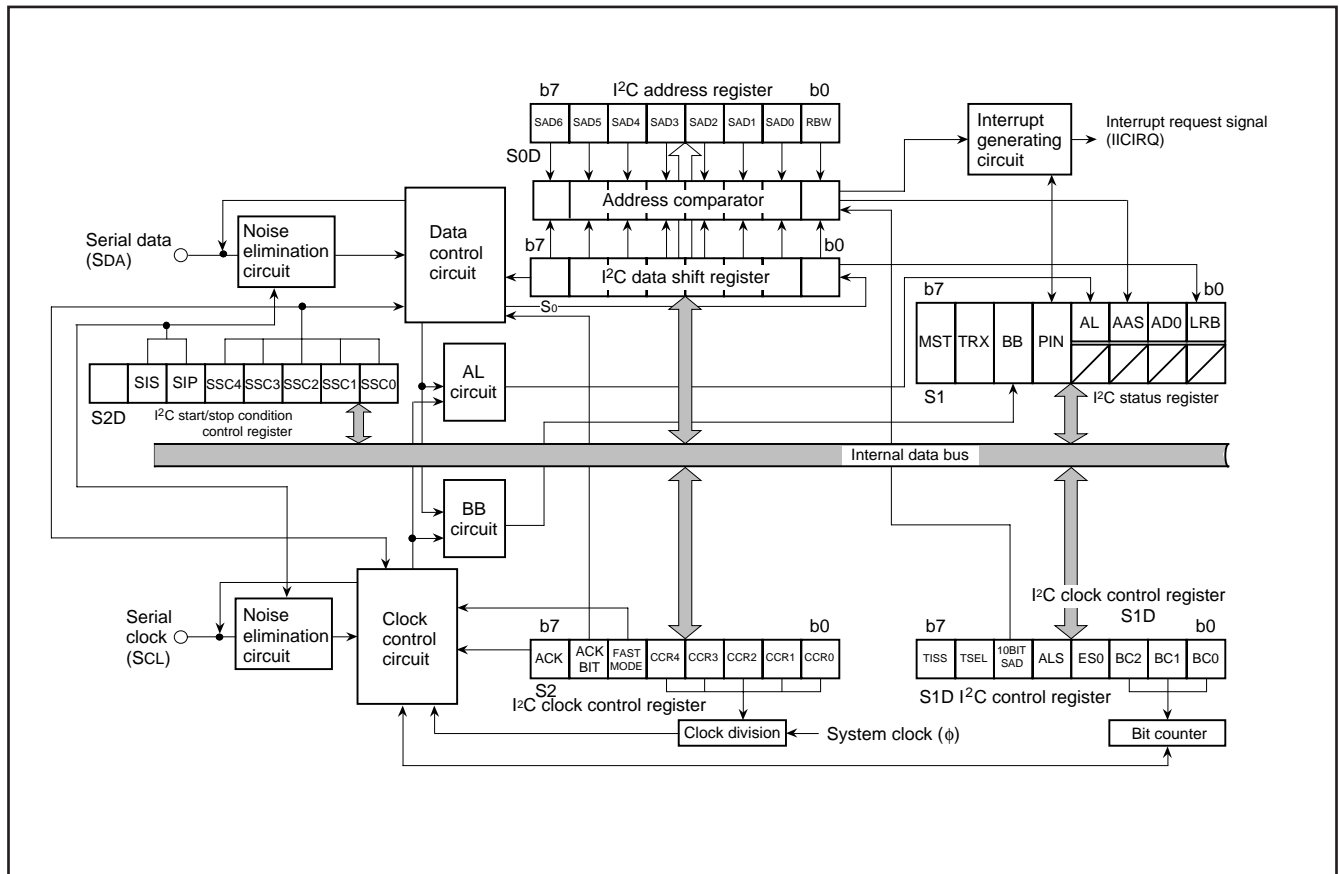


Fig. 22 Block diagram of multi-master I²C-BUS interface

* : Purchase of MITSUBISHI ELECTRIC CORPORATIONS I²C components conveys a license under the Philips I²C Patent Rights to use these components an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

HARDWARE

FUNCTIONAL DESCRIPTION

[I²C Data Shift Register (S0)] 002B16

The I²C data shift register (S0 : address 002B16) is an 8-bit shift register to store receive data and write transmit data.

When transmit data is written into this register, it is transferred to the outside from bit 7 in synchronization with the SCL clock, and each time one-bit data is output, the data of this register are shifted by one bit to the left. When data is received, it is input to this register from bit 0 in synchronization with the SCL clock, and each time one-bit data is input, the data of this register are shifted by one bit to the left. The minimum 2 machine cycles are required from the rising of the SCL clock until input to this register.

The I²C data shift register is in a write enable status only when the I²C-BUS interface enable bit (ES0 bit : bit 3 of address 002E16) of the I²C control register is "1". The bit counter is reset by a write instruction to the I²C data shift register. When both the ES0 bit and the MST bit of the I²C status register (address 002D16) are "1," the SCL is output by a write instruction to the I²C data shift register. Reading data from the I²C data shift register is always enabled regardless of the ES0 bit value.

[I²C Address Register (S0D)] 002C16

The I²C address register (address 002C16) consists of a 7-bit slave address and a read/write bit. In the addressing mode, the slave address written in this register is compared with the address data to be received immediately after the START condition is detected.

•Bit 0: Read/write bit (RWB)

This is not used in the 7-bit addressing mode. In the 10-bit addressing mode, the first address data to be received is compared with the contents (SAD6 to SAD0 + RBW) of the I²C address register.

The RWB bit is cleared to "0" automatically when the stop condition is detected.

•Bits 1 to 7: Slave address (SAD0–SAD6)

These bits store slave addresses. Regardless of the 7-bit address

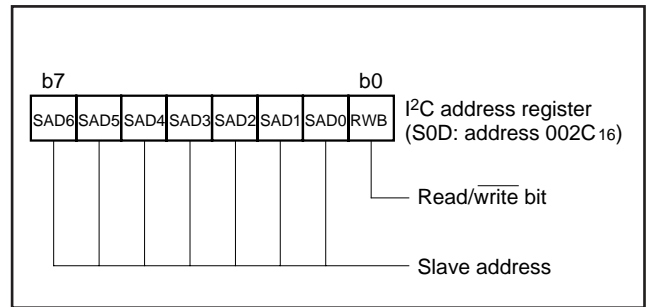


Fig. 23 Structure of I²C address register

[I²C Clock Control Register (S2)] 002F16

The I²C clock control register (address 002F16) is used to set ACK control, SCL mode and SCL frequency.

•**Bits 0 to 4: SCL frequency control bits (CCR0–CCR4)**

These bits control the SCL frequency. Refer to Table 5.

•**Bit 5: SCL mode specification bit (FAST MODE)**

This bit specifies the SCL mode. When this bit is set to “0,” the standard clock mode is selected. When the bit is set to “1,” the high-speed clock mode is selected.

When connecting the bus of the high-speed mode I²C bus standard (maximum 400 kbits/s), use 8 MHz or more oscillation frequency f(XIN) and 2 division clock.

•**Bit 6: ACK bit (ACK BIT)**

This bit sets the SDA status when an ACK clock* is generated. When this bit is set to “0,” the ACK return mode is selected and SDA goes to “L” at the occurrence of an ACK clock. When the bit is set to “1,” the ACK non-return mode is selected. The SDA is held in the “H” status at the occurrence of an ACK clock.

However, when the slave address agree with the address data in the reception of address data at ACK BIT = “0,” the SDA is automatically made “L” (ACK is returned). If there is a disagreement between the slave address and the address data, the SDA is automatically made “H” (ACK is not returned).

*ACK clock: Clock for acknowledgment

•**Bit 7: ACK clock bit (ACK)**

This bit specifies the mode of acknowledgment which is an acknowledgment response of data transfer. When this bit is set to “0,” the no ACK clock mode is selected. In this case, no ACK clock occurs after data transmission. When the bit is set to “1,” the ACK clock mode is selected and the master generates an ACK clock each completion of each 1-byte data transfer. The device for transmitting address data and control data releases the SDA at the occurrence of an ACK clock (makes SDA “H”) and receives the ACK bit generated by the data receiving device.

Note: Do not write data into the I²C clock control register during transfer. If data is written during transfer, the I²C clock generator is reset, so that data cannot be transferred normally.

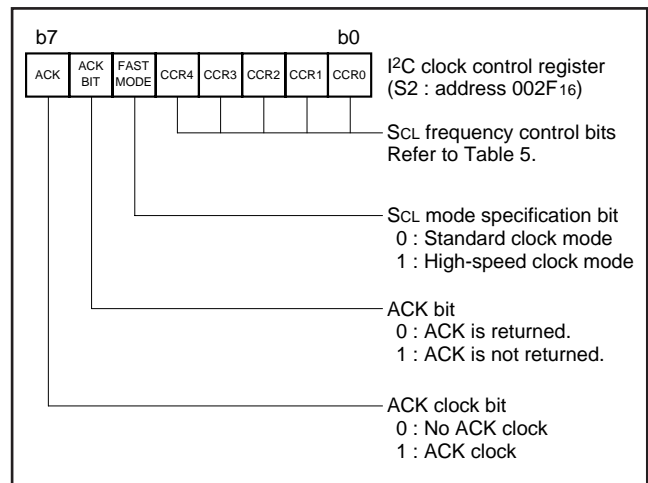


Fig. 24 Structure of I²C clock control register

Table 7 Set values of I²C clock control register and SCL frequency

Setting value of CCR4–CCR0					SCL frequency (at $\phi = 4$ MHz, unit : kHz)	
CCR4	CCR3	CCR2	CCR1	CCR0	Standard clock mode	High-speed clock mode
0	0	0	0	0	Setting disabled	Setting disabled
0	0	0	0	1	Setting disabled	Setting disabled
0	0	0	1	0	Setting disabled	Setting disabled
0	0	0	1	1	– (Note 2)	333
0	0	1	0	0	– (Note 2)	250
0	0	1	0	1	100	400 (Note 3)
0	0	1	1	0	83.3	166
⋮	⋮	⋮	⋮	⋮	500/CCR value (Note 3)	1000/CCR value (Note 3)
1	1	1	0	1	17.2	34.5
1	1	1	1	0	16.6	33.3
1	1	1	1	1	16.1	32.3

Notes 1: Duty of SCL clock output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at $\phi = 4$ MHz). “H” duration of the clock fluctuates from –4 to +2 machine cycles in the standard clock mode, and fluctuates from –2 to +2 machine cycles in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because “L” duration is extended instead of “H” duration reduction.

These are value when SCL clock synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

2: Each value of SCL frequency exceeds the limit at $\phi = 4$ MHz or more. When using these setting value, use ϕ of 4 MHz or less.

3: The data formula of SCL frequency is described below:

$\phi / (8 \times \text{CCR value})$ Standard clock mode

$\phi / (4 \times \text{CCR value})$ High-speed clock mode (CCR value $\neq 5$)

$\phi / (2 \times \text{CCR value})$ High-speed clock mode (CCR value = 5)

Do not set 0 to 2 as CCR value regardless of ϕ frequency.

Set 100 kHz (max.) in the standard clock mode and 400 kHz (max.) in the high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

HARDWARE

FUNCTIONAL DESCRIPTION

[I²C Control Register (S1D)] 002E16

The I²C control register (address 002E16) controls data communication format.

•Bit 0 to 2: Bit counter (BC0–BC2)

These bits decide the number of bits for the next 1-byte data to be transmitted. The I²C interrupt request signal occurs immediately after the number of count specified with these bits (ACK clock is added to the number of count when ACK clock is selected by ACK clock bit (bit 7 of address 002F16)) have been transferred, and BC0 to BC2 are returned to "0002".

Also when a START condition is received, these bits become "0002" and the address data is always transmitted and received in 8 bits.

•Bit 3: I²C interface enable bit (ES0)

This bit enables to use the multi-master I²C-BUS interface. When this bit is set to "0," the use disable status is provided, so that the SDA and the SCL become high-impedance. When the bit is set to "1," use of the interface is enabled.

When ES0 = "0," the following is performed.

- PIN = "1," BB = "0" and AL = "0" are set (which are bits of the I²C status register at address 002D16).
- Writing data to the I²C data shift register (address 002B16) is disabled.

•Bit 4: Data format selection bit (ALS)

This bit decides whether or not to recognize slave addresses. When this bit is set to "0," the addressing format is selected, so that address data is recognized. When a match is found between a slave address and address data as a result of comparison or when a general call (refer to "I²C Status Register," bit 1) is received, transfer processing can be performed. When this bit is set to "1," the free data format is selected, so that slave addresses are not recognized.

•Bit 5: Addressing format selection bit (10BIT SAD)

This bit selects a slave address specification format. When this bit is set to "0," the 7-bit addressing format is selected. In this case, only the high-order 7 bits (slave address) of the I²C address register (address 002C16) are compared with address data. When this bit is set to "1," the 10-bit addressing format is selected, and all the bits of the I²C address register are compared with address data.

•Bit 6: SDA/SCL pin selection bit

This bit selects the input/output pins of SCL and SDA of the multi-master I²C-BUS interface.

•Bit 7: I²C-BUS interface pin input level selection bit

This bit selects the input level of the SCL and SDA pins of the multi-master I²C-BUS interface.

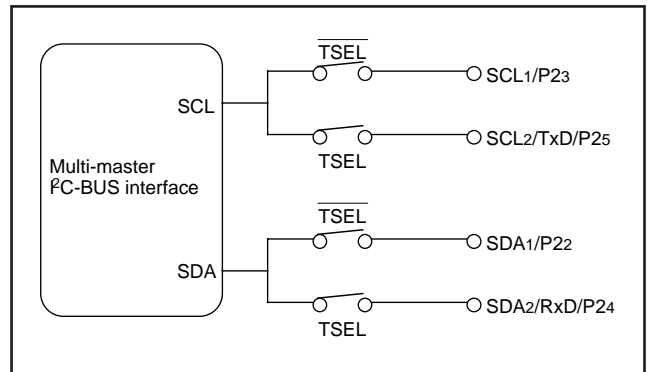


Fig. 25 SDA/SCL pin selection bit

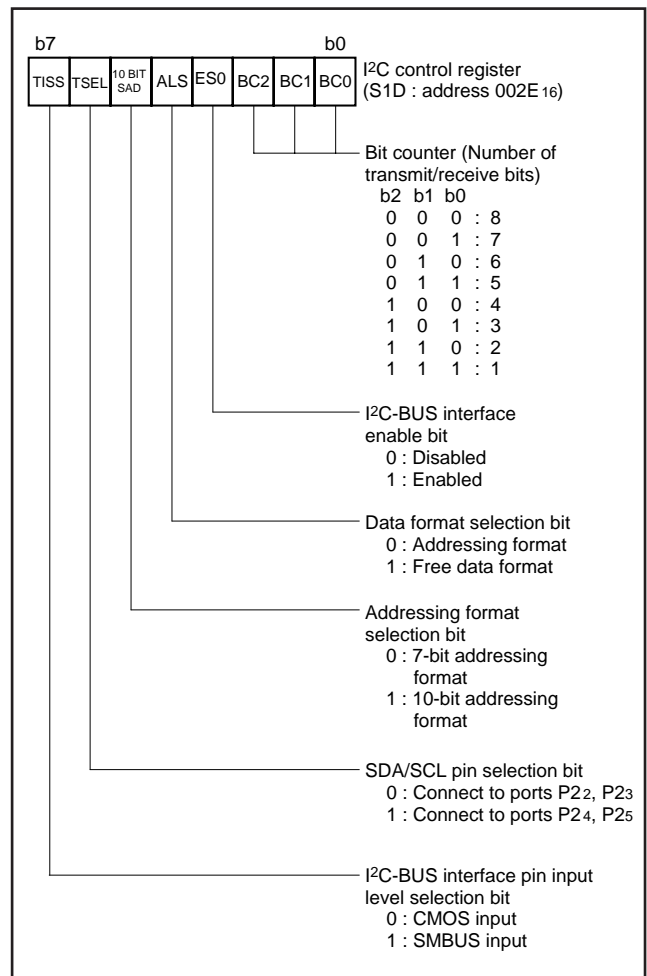


Fig. 26 Structure of I²C control register

[I²C Status Register (S1)] 002D16

The I²C status register (address 002D16) controls the I²C-BUS interface status. The low-order 4 bits are read-only bits and the high-order 4 bits can be read out and written to.

Set "00002" to the low-order 4 bits, because these bits become the reserved bits at writing.

•Bit 0: Last receive bit (LRB)

This bit stores the last bit value of received data and can also be used for ACK receive confirmation. If ACK is returned when an ACK clock occurs, the LRB bit is set to "0." If ACK is not returned, this bit is set to "1." Except in the ACK mode, the last bit value of received data is input. The state of this bit is changed from "1" to "0" by executing a write instruction to the I²C data shift register (address 002B16).

•Bit 1: General call detecting flag (AD0)

When the ALS bit is "0", this bit is set to "1" when a general call* whose address data is all "0" is received in the slave mode. By a general call of the master device, every slave device receives control data after the general call. The AD0 bit is set to "0" by detecting the STOP condition or START condition, or reset.

*General call: The master transmits the general call address "0016" to all slaves.

•Bit 2: Slave address comparison flag (AAS)

This flag indicates a comparison result of address data when the ALS bit is "0".

- ① In the slave receive mode, when the 7-bit addressing format is selected, this bit is set to "1" in one of the following conditions:
 - The address data immediately after occurrence of a START condition agrees with the slave address stored in the high-order 7 bits of the I²C address register (address 002C16).
 - A general call is received.
- ② In the slave receive mode, when the 10-bit addressing format is selected, this bit is set to "1" with the following condition:
 - When the address data is compared with the I²C address register (8 bits consisting of slave address and RBW bit), the first bytes agree.
- ③ This bit is set to "0" by executing a write instruction to the I²C data shift register (address 002B16) when ES0 is set to "1" or reset.

•Bit 3: Arbitration lost* detecting flag (AL)

In the master transmission mode, when the SDA is made "L" by any other device, arbitration is judged to have been lost, so that this bit is set to "1." At the same time, the TRX bit is set to "0," so that immediately after transmission of the byte whose arbitration was lost is completed, the MST bit is set to "0." The arbitration lost can be detected only in the master transmission mode. When arbitration is lost during slave address transmission, the TRX bit is set to "0" and the reception mode is set. Consequently, it becomes possible to detect the agreement of its own slave address and address data transmitted by another master device.

*Arbitration lost: The status in which communication as a master is disabled.

•Bit 4: SCL pin low hold bit (PIN)

This bit generates an interrupt request signal. Each time 1-byte data is transmitted, the PIN bit changes from "1" to "0." At the same time, an interrupt request signal occurs to the CPU. The PIN bit is set to "0" in synchronization with a falling of the last clock (including the ACK clock) of an internal clock and an interrupt request signal occurs in synchronization with a falling of the PIN bit. When the PIN bit is "0," the SCL is kept in the "0" state and clock generation is disabled. Figure 25 shows an interrupt request signal generating timing chart.

The PIN bit is set to "1" in one of the following conditions:

- Executing a write instruction to the I²C data shift register (address 002B16). (This is the only condition which the prohibition of the internal clock is released and data can be communicated except for the start condition detection.)
- When the ES0 bit is "0"
- At reset
- When writing "1" to the PIN bit by software

The conditions in which the PIN bit is set to "0" are shown below:

- Immediately after completion of 1-byte data transmission (including when arbitration lost is detected)
- Immediately after completion of 1-byte data reception
- In the slave reception mode, with ALS = "0" and immediately after completion of slave address agreement or general call address reception
- In the slave reception mode, with ALS = "1" and immediately after completion of address data reception

•Bit 5: Bus busy flag (BB)

This bit indicates the status of use of the bus system. When this bit is set to "0," this bus system is not busy and a START condition can be generated. The BB flag is set/reset by the SCL, SDA pins input signal regardless of master/slave. This flag is set to "1" by detecting the start condition, and is set to "0" by detecting the stop condition. The condition of these detecting is set by the start/stop condition setting bits (SSC4–SSC0) of the I²C start/stop condition control register (address 003016). When the ES0 bit of the I²C control register (address 002E16) is "0" or reset, the BB flag is set to "0."

For the writing function to the BB flag, refer to the sections "START Condition Generating Method" and "STOP Condition Generating Method" described later.

HARDWARE

FUNCTIONAL DESCRIPTION

•Bit 6: Communication mode specification bit (transfer direction specification bit: TRX)

This bit decides a direction of transfer for data communication. When this bit is “0,” the reception mode is selected and the data of a transmitting device is received. When the bit is “1,” the transmission mode is selected and address data and control data are output onto the SDA in synchronization with the clock generated on the SCL.

This bit is set/reset by software and hardware. About set/reset by hardware is described below. This bit is set to “1” by hardware when all the following conditions are satisfied:

- When ALS is “0”
- In the slave reception mode or the slave transmission mode
- When the $\overline{R/W}$ bit reception is “1”

This bit is set to “0” in one of the following conditions:

- When arbitration lost is detected.
- When a STOP condition is detected.
- When writing “1” to this bit by software is invalid by the START condition duplication preventing function (**Note**).
- With MST = “0” and when a START condition is detected.
- With MST = “0” and when ACK non-return is detected.
- At reset

•Bit 7: Communication mode specification bit (master/slave specification bit: MST)

This bit is used for master/slave specification for data communication. When this bit is “0,” the slave is specified, so that a START condition and a STOP condition generated by the master are received, and data communication is performed in synchronization with the clock generated by the master. When this bit is “1,” the master is specified and a START condition and a STOP condition are generated. Additionally, the clocks required for data communication are generated on the SCL.

This bit is set to “0” in one of the following conditions.

- Immediately after completion of 1-byte data transfer when arbitration lost is detected
- When a STOP condition is detected.
- Writing “1” to this bit by software is invalid by the START condition duplication preventing function (**Note**).
- At reset

Note: START condition duplication preventing function

The MST, TRX, and BB bits is set to “1” at the same time after confirming that the BB flag is “0” in the procedure of a START condition occurrence. However, when a START condition by another master device occurs and the BB flag is set to “1” immediately after the contents of the BB flag is confirmed, the START condition duplication preventing function makes the writing to the MST and TRX bits invalid. The duplication preventing function becomes valid from the rising of the BB flag to reception completion of slave address.

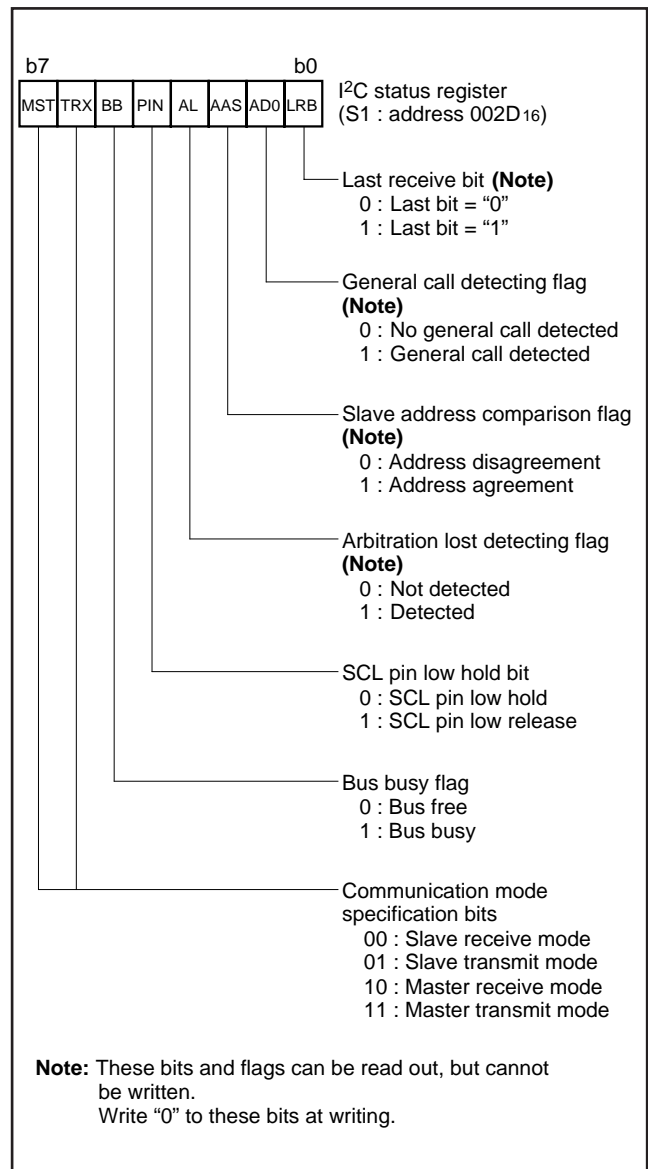


Fig. 27 Structure of I²C status register

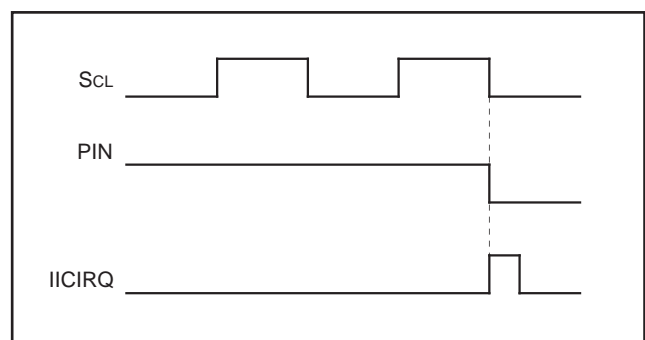


Fig. 28 Interrupt request signal generating timing

START Condition Generating Method

When writing "1" to the MST, TRX, and BB bits of the I²C status register (address 002D16) at the same time after writing the slave address to the I²C data shift register (address 002B16) with the condition in which the ES0 bit of the I²C control register (address 002E16) is "1" and the BB flag is "0", a START condition occurs. After that, the bit counter becomes "0002" and an SCL for 1 byte is output. The START condition generating timing is different in the standard clock mode and the high-speed clock mode. Refer to Figure 26, the START condition generating timing diagram, and Table 6, the START condition generating timing table.

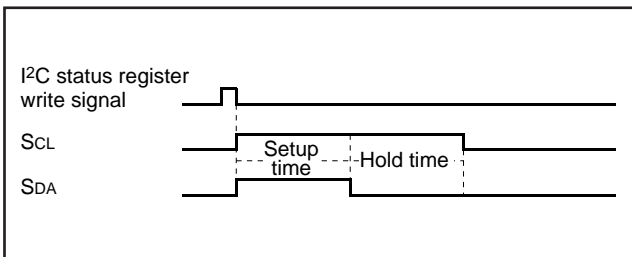


Fig. 29 START condition generating timing diagram

Table 8 START condition generating timing table

Item	Standard clock mode	High-speed clock mode
Setup time	5.0 μs (20 cycles)	2.5 μs (10 cycles)
Hold time	5.0 μs (20 cycles)	2.5 μs (10 cycles)

Note: Absolute time at $\phi = 4$ MHz. The value in parentheses denotes the number of ϕ cycles.

STOP Condition Generating Method

When the ES0 bit of the I²C control register (address 002E16) is "1," write "1" to the MST and TRX bits, and write "0" to the BB bit of the I²C status register (address 002D16) simultaneously. Then a STOP condition occurs. The STOP condition generating timing is different in the standard clock mode and the high-speed clock mode. Refer to Figure 27, the STOP condition generating timing diagram, and Table 7, the STOP condition generating timing table.

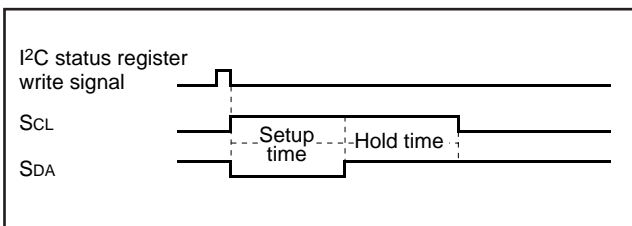


Fig. 30 STOP condition generating timing diagram

Table 9 STOP condition generating timing table

Item	Standard clock mode	High-speed clock mode
Setup time	5.0 μs (20 cycles)	3.0 μs (12 cycles)
Hold time	4.5 μs (18 cycles)	2.5 μs (10 cycles)

Note: Absolute time at $\phi = 4$ MHz. The value in parentheses denotes the number of ϕ cycles.

START/STOP Condition Detecting Operation

The START/STOP condition detection operations are shown in Figures 28, 29, and Table 8. The START/STOP condition is set by the START/STOP condition set bit.

The START/STOP condition can be detected only when the input signal of the SCL and SDA pins satisfy three conditions: SCL release time, setup time, and hold time (see Table 8).

The BB flag is set to "1" by detecting the START condition and is reset to "0" by detecting the STOP condition.

The BB flag set/reset timing is different in the standard clock mode and the high-speed clock mode. Refer to Table 8, the BB flag set/reset time.

Note: When a STOP condition is detected in the slave mode (MST = 0), an interrupt request signal "IICIRQ" occurs to the CPU.

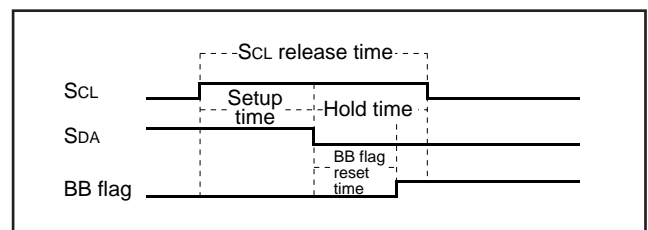


Fig. 31 START/STOP condition detecting timing diagram

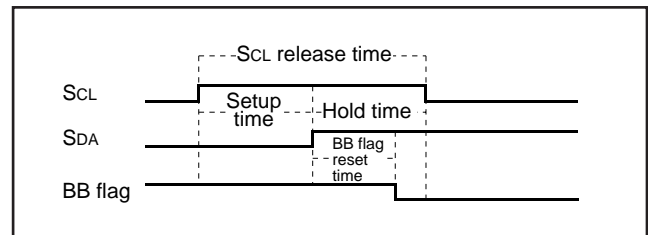


Fig. 32 STOP condition detecting timing diagram

Table 10 START condition/STOP condition detecting conditions

	Standard clock mode	High-speed clock mode
SCL release time	SCC value + 1 cycle (6.25 μs)	4 cycles (1.0 μs)
Setup time	$\frac{\text{SCC value} + 1}{2}$ cycle < 4.0 μs (3.125 μs)	2 cycles (1.0 μs)
Hold time	$\frac{\text{SCC value} + 1}{2}$ cycle < 4.0 μs (3.125 μs)	2 cycles (0.5 μs)
BB flag set/reset time	$\frac{\text{SCC value} - 1}{2} + 2$ cycles (3.375 μs)	3.5 cycles (0.875 μs)

Note: Unit : Cycle number of system clock ϕ

SSC value is the decimal notation value of the START/STOP condition set bits SSC4 to SSC0. Do not set "0" or an odd number to SSC value. The value in parentheses is an example when the I²C START/STOP condition control register is set to "1816" at $\phi = 4$ MHz.

HARDWARE

FUNCTIONAL DESCRIPTION

[I²C START/STOP Condition Control Register (S2D)] 0030₁₆

The I²C START/STOP condition control register (address 0030₁₆) controls START/STOP condition detection.

•**Bits 0 to 4: START/STOP condition set bit (SSC4–SSC0)**

SCL release time, setup time, and hold time change the detection condition by value of the main clock divide ratio selection bit and the oscillation frequency $f(X_{IN})$ because these time are measured by the internal system clock. Accordingly, set the proper value to the START/STOP condition set bits (SSC4 to SSC0) in considered of the system clock frequency. Refer to Table 8.

Do not set "00000₂" or an odd number to the START/STOP condition set bit (SSC4 to SSC0).

Refer to Table 9, the recommended set value to START/STOP condition set bits (SSC4–SSC0) for each oscillation frequency.

•**Bit 5: SCL/SDA interrupt pin polarity selection bit (SIP)**

An interrupt can occur when detecting the falling or rising edge of the SCL or SDA pin. This bit selects the polarity of the SCL or SDA pin interrupt pin.

•**Bit 6: SCL/SDA interrupt pin selection bit (SIS)**

This bit selects the pin of which interrupt becomes valid between the SCL pin and the SDA pin.

Note: When changing the setting of the SCL/SDA interrupt pin polarity selection bit, the SCL/SDA interrupt pin selection bit, or the I²C-BUS interface enable bit ES0, the SCL/SDA interrupt request bit may be set. When selecting the SCL/SDA interrupt source, disable the interrupt before the SCL/SDA interrupt pin polarity selection bit, the SCL/SDA interrupt pin selection bit, or the I²C-BUS interface enable bit ES0 is set. Reset the request bit to "0" after setting these bits, and enable the interrupt.

Address Data Communication

There are two address data communication formats, namely, 7-bit addressing format and 10-bit addressing format. The respective address communication formats are described below.

① 7-bit addressing format

To adapt the 7-bit addressing format, set the 10BIT SAD bit of the I²C control register (address 002E₁₆) to "0." The first 7-bit address data transmitted from the master is compared with the high-order 7-bit slave address stored in the I²C address register (address 002C₁₆). At the time of this comparison, address comparison of the RWB bit of the I²C address register (address 002C₁₆) is not performed. For the data transmission format when the 7-bit addressing format is selected, refer to Figure 31, (1) and (2).

② 10-bit addressing format

To adapt the 10-bit addressing format, set the 10BIT SAD bit of the I²C control register (address 002E₁₆) to "1." An address comparison is performed between the first-byte address data transmitted from the master and the 8-bit slave address stored in the I²C address register (address 002C₁₆). At the time of this comparison, an address comparison between the RWB bit of the I²C address register (address 002C₁₆) and the R/\bar{W} bit which is the last bit of the address data transmitted from the master is made. In the 10-bit addressing mode, the RWB bit which is the last bit of the address data not only specifies the direction of communication for control data, but also is processed as an address data bit.

When the first-byte address data agree with the slave address, the AAS bit of the I²C status register (address 002D₁₆) is set to "1." After the second-byte address data is stored into the I²C data shift register (address 002B₁₆), perform an address comparison between the second-byte data and the slave address by software. When the address data of the 2 bytes agree with the slave address, set the RBW bit of the I²C address register (address 002C₁₆) to "1" by software. This processing can make the 7-bit slave address and R/\bar{W} data agree, which are received after a RESTART condition is detected, with the value of the I²C address register (address 002C₁₆). For the data transmission format when the 10-bit addressing format is selected, refer to Figure 31, (3) and (4).

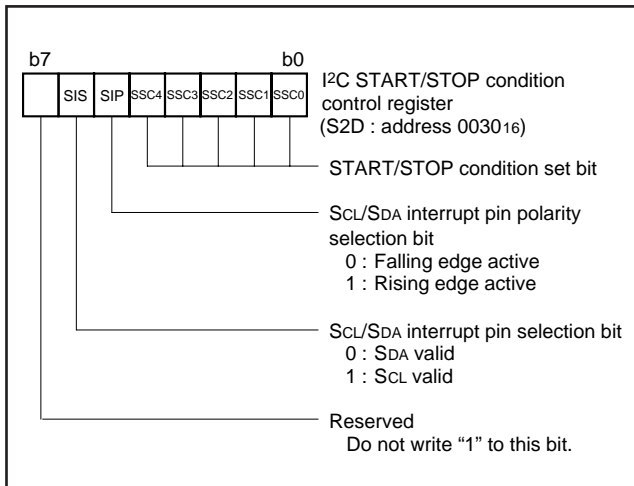


Fig. 33 Structure of I²C START/STOP condition control register

Table 11 Recommended set value to START/STOP condition set bits (SSC4–SSC0) for each oscillation frequency

Oscillation frequency f(XIN) (MHz)	Main clock divide ratio	System clock ϕ (MHz)	START/STOP condition control register	SCL release time (μ s)	Setup time (μ s)	Hold time (μ s)
8	2	4	XXX11010	6.75 μ s (27 cycles)	3.375 μ s (13.5 cycles)	3.375 μ s (13.5 cycles)
			XXX11000	6.25 μ s (25 cycles)	3.125 μ s (12.5 cycles)	3.125 μ s (12.5 cycles)
8	8	1	XXX00100	5.0 μ s (5 cycles)	2.5 μ s (2.5 cycles)	2.5 μ s (2.5 cycles)
4	2	2	XXX01100	6.5 μ s (13 cycles)	3.25 μ s (6.5 cycles)	3.25 μ s (6.5 cycles)
			XXX01010	5.5 μ s (11 cycles)	2.75 μ s (5.5 cycles)	2.75 μ s (5.5 cycles)
2	2	1	XXX00100	5.0 μ s (5 cycles)	2.5 μ s (2.5 cycles)	2.5 μ s (2.5 cycles)

Note: Do not set an odd number to the START/STOP condition set bit (SSC4 to SSC0).

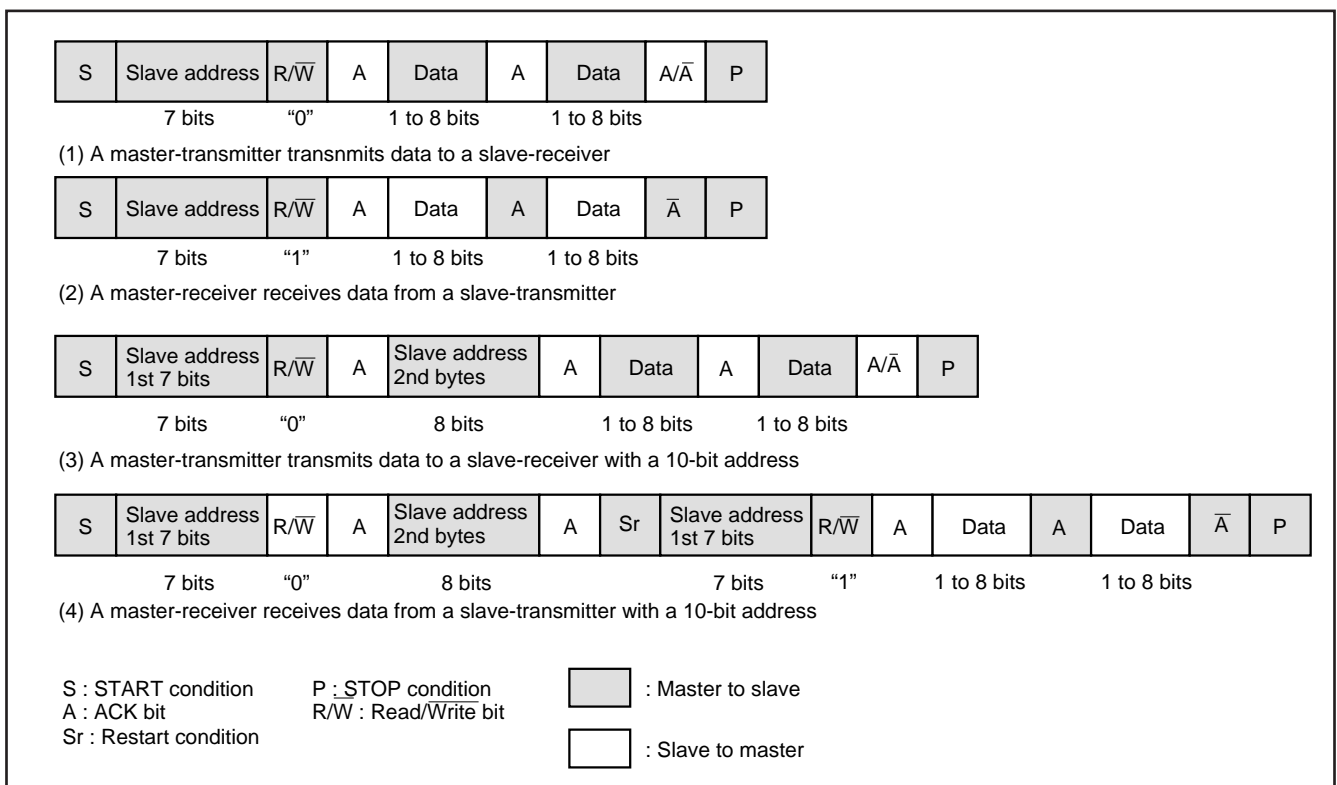


Fig. 34 Address data communication format

HARDWARE

FUNCTIONAL DESCRIPTION

Example of Master Transmission

An example of master transmission in the standard clock mode, at the SCL frequency of 100 kHz and in the ACK return mode is shown below.

- ① Set a slave address in the high-order 7 bits of the I²C address register (address 002C₁₆) and “0” into the RWB bit.
- ② Set the ACK return mode and SCL = 100 kHz by setting “85₁₆” in the I²C clock control register (address 002F₁₆).
- ③ Set “00₁₆” in the I²C status register (address 002D₁₆) so that transmission/reception mode can become initializing condition.
- ④ Set a communication enable status by setting “08₁₆” in the I²C control register (address 002E₁₆).
- ⑤ Confirm the bus free condition by the BB flag of the I²C status register (address 002D₁₆).
- ⑥ Set the address data of the destination of transmission in the high-order 7 bits of the I²C data shift register (address 002B₁₆) and set “0” in the least significant bit.
- ⑦ Set “F0₁₆” in the I²C status register (address 002D₁₆) to generate a START condition. At this time, an SCL for 1 byte and an ACK clock automatically occur.
- ⑧ Set transmit data in the I²C data shift register (address 002B₁₆). At this time, an SCL and an ACK clock automatically occur.
- ⑨ When transmitting control data of more than 1 byte, repeat step ⑧.
- ⑩ Set “D0₁₆” in the I²C status register (address 002D₁₆) to generate a STOP condition if ACK is not returned from slave reception side or transmission ends.

Example of Slave Reception

An example of slave reception in the high-speed clock mode, at the SCL frequency of 400 kHz, in the ACK non-return mode and using the addressing format is shown below.

- ① Set a slave address in the high-order 7 bits of the I²C address register (address 002C₁₆) and “0” in the RWB bit.
- ② Set the no ACK clock mode and SCL = 400 kHz by setting “65₁₆” in the I²C clock control register (address 002F₁₆).
- ③ Set “00₁₆” in the I²C status register (address 002D₁₆) so that transmission/reception mode can become initializing condition.
- ④ Set a communication enable status by setting “08₁₆” in the I²C control register (address 002E₁₆).
- ⑤ When a START condition is received, an address comparison is performed.
- ⑥ •When all transmitted addresses are “0” (general call):
AD0 of the I²C status register (address 002D₁₆) is set to “1” and an interrupt request signal occurs.
• When the transmitted addresses agree with the address set in ①:
ASS of the I²C status register (address 002D₁₆) is set to “1” and an interrupt request signal occurs.
• In the cases other than the above AD0 and AAS of the I²C status register (address 002D₁₆) are set to “0” and no interrupt request signal occurs.
- ⑦ Set dummy data in the I²C data shift register (address 002B₁₆).
- ⑧ When receiving control data of more than 1 byte, repeat step ⑦.
- ⑨ When a STOP condition is detected, the communication ends.

■Precautions when using multi-master I²C-BUS interface

(1) Read-modify-write instruction

The precautions when the read-modify-write instruction such as SEB, CLB etc. is executed for each register of the multi-master I²C-BUS interface are described below.

- I²C data shift register (S0: address 002B₁₆)
When executing the read-modify-write instruction for this register during transfer, data may become a value not intended.
- I²C address register (S0D: address 002C₁₆)
When the read-modify-write instruction is executed for this register at detecting the STOP condition, data may become a value not intended. It is because H/W changes the read/write bit (RBW) at the above timing.
- I²C status register (S1: address 002D₁₆)
Do not execute the read-modify-write instruction for this register because all bits of this register are changed by H/W.
- I²C control register (S1D: address 002E₁₆)
When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become a value not intended. Because H/W changes the bit counter (BC0-BC2) at the above timing.
- I²C clock control register (S2: address 002F₁₆)
The read-modify-write instruction can be executed for this register.
- I²C START/STOP condition control register (S2D: address 0030₁₆)
The read-modify-write instruction can be executed for this register.

(2) START condition generating procedure using multi-master

1. Procedure example (The necessary conditions of the generating procedure are described as the following 2 to 5.

```

:
LDA —          (Taking out of slave address value)
SEI             (Interrupt disabled)
BBS 5, S1, BUSBUSY (BB flag confirming and branch process)

BUSFREE:
STA S0         (Writing of slave address value)
LDM #$F0, S1  (Trigger of START condition generating)
CLI           (Interrupt enabled)
:
BUSBUSY:
CLI           (Interrupt enabled)
:

```

2. Use "Branch on Bit Set" of "BBS 5, \$002D, -" for the BB flag confirming and branch process.
3. Use "STA \$2B, STX \$2B" or "STY \$2B" of the zero page addressing instruction for writing the slave address value to the I²C data shift register.
4. Execute the branch instruction of above 2 and the store instruction of above 3 continuously shown the above procedure example.

5. Disable interrupts during the following three process steps:

- BB flag confirming
- Writing of slave address value
- Trigger of START condition generating

When the condition of the BB flag is bus busy, enable interrupts immediately.

(3) RESTART condition generating procedure

1. Procedure example (The necessary conditions of the generating procedure are described as the following 2 to 4.)

Execute the following procedure when the PIN bit is "0."

```

:
LDM #$00, S1      (Select slave receive mode)
LDA —            (Taking out of slave address value)
SEI               (Interrupt disabled)
STA S0            (Writing of slave address value)
LDM #$F0, S1      (Trigger of RESTART condition generating)
CLI               (Interrupt enabled)
:

```

2. Select the slave receive mode when the PIN bit is "0." Do not write "1" to the PIN bit. Neither "0" nor "1" is specified for the writing to the BB bit.
The TRX bit becomes "0" and the SDA pin is released.
3. The SCL pin is released by writing the slave address value to the I²C data shift register.
4. Disable interrupts during the following two process steps:
 - Writing of slave address value
 - Trigger of RESTART condition generating

(4) Writing to I²C status register

Do not execute an instruction to set the PIN bit to "1" from "0" and an instruction to set the MST and TRX bits to "0" from "1" simultaneously. It is because it may enter the state that the SCL pin is released and the SDA pin is released after about one machine cycle. Do not execute an instruction to set the MST and TRX bits to "0" from "1" simultaneously when the PIN bit is "1." It is because it may become the same as above.

(5) Process of after STOP condition generating

Do not write data in the I²C data shift register S0 and the I²C status register S1 until the bus busy flag BB becomes "0" after generating the STOP condition in the master mode. It is because the STOP condition waveform might not be normally generated. Reading to the above registers do not have the problem.

HARDWARE

FUNCTIONAL DESCRIPTION

PULSE WIDTH MODULATION (PWM)

The 3851 group has a PWM function with an 8-bit resolution, based on a signal that is the clock input X_{IN} or that clock input divided by 2.

Data Setting

The PWM output pin also functions as port P44. Set the PWM period by the PWM prescaler, and set the "H" term of output pulse by the PWM register.

If the value in the PWM prescaler is n and the value in the PWM register is m (where n = 0 to 255 and m = 0 to 255) :

$$\begin{aligned} \text{PWM period} &= 255 \times (n+1) / f(X_{IN}) \\ &= 31.875 \times (n+1) \mu\text{s} \\ &\quad (\text{when } f(X_{IN}) = 8 \text{ MHz, count source is } f(X_{IN})) \end{aligned}$$

$$\begin{aligned} \text{Output pulse "H" term} &= \text{PWM period} \times m / 255 \\ &= 0.125 \times (n+1) \times m \mu\text{s} \\ &\quad (\text{when } f(X_{IN}) = 8 \text{ MHz, count source is } f(X_{IN})) \end{aligned}$$

PWM Operation

When bit 0 (PWM enable bit) of the PWM control register is set to "1", operation starts by initializing the PWM output circuit, and pulses are output starting at an "H".

If the PWM register or PWM prescaler is updated during PWM output, the pulses will change in the cycle after the one in which the change was made.

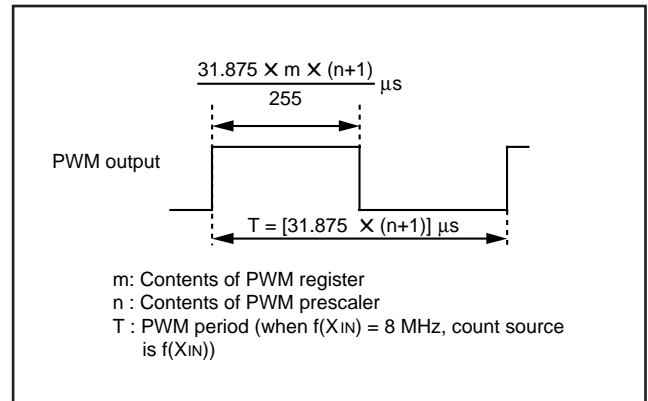


Fig. 35 Timing of PWM period

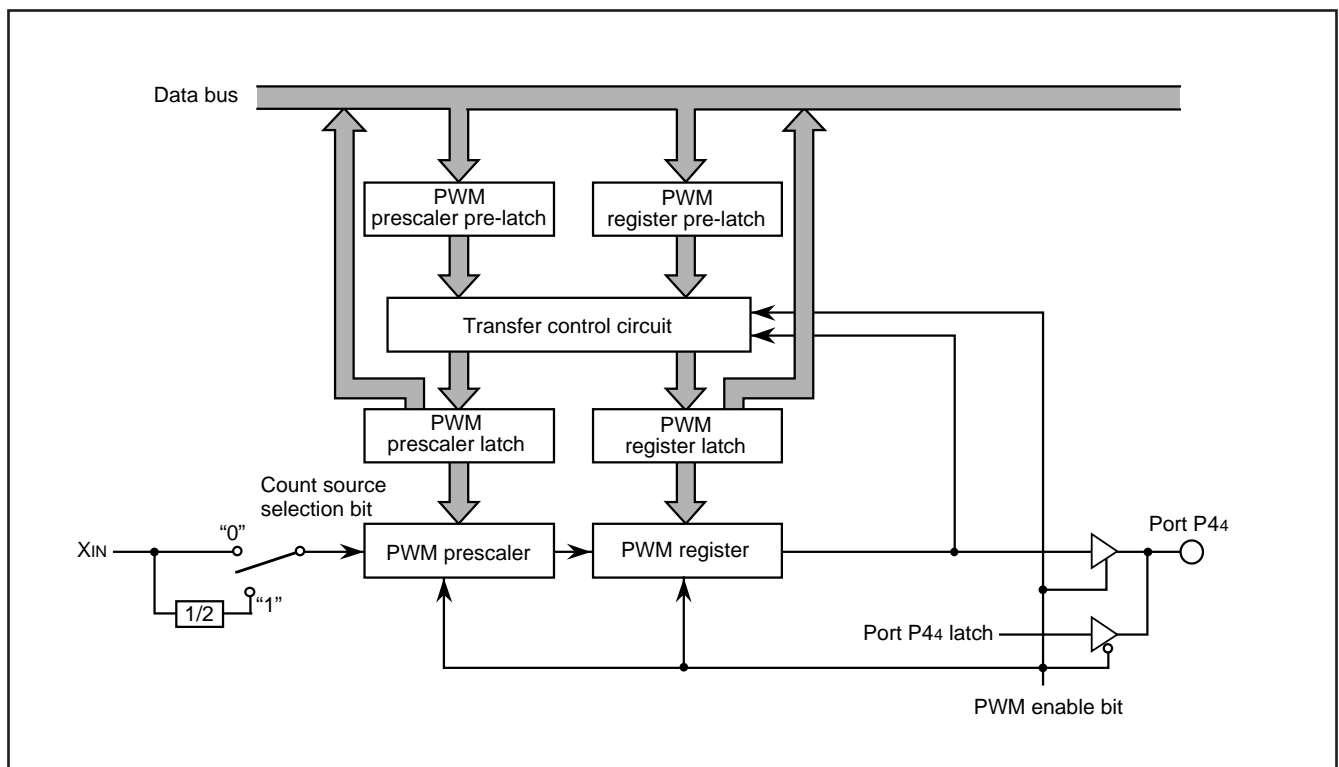


Fig. 36 Block diagram of PWM function

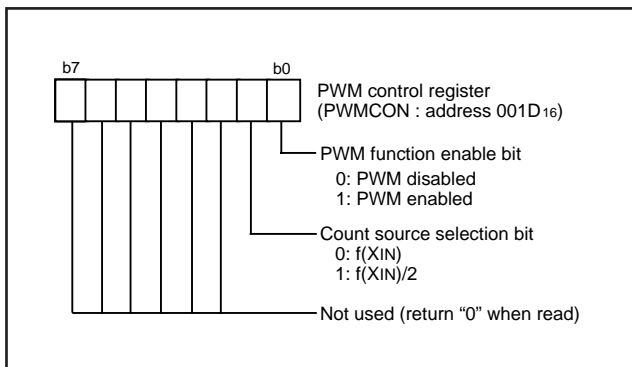


Fig. 37 Structure of PWM control register

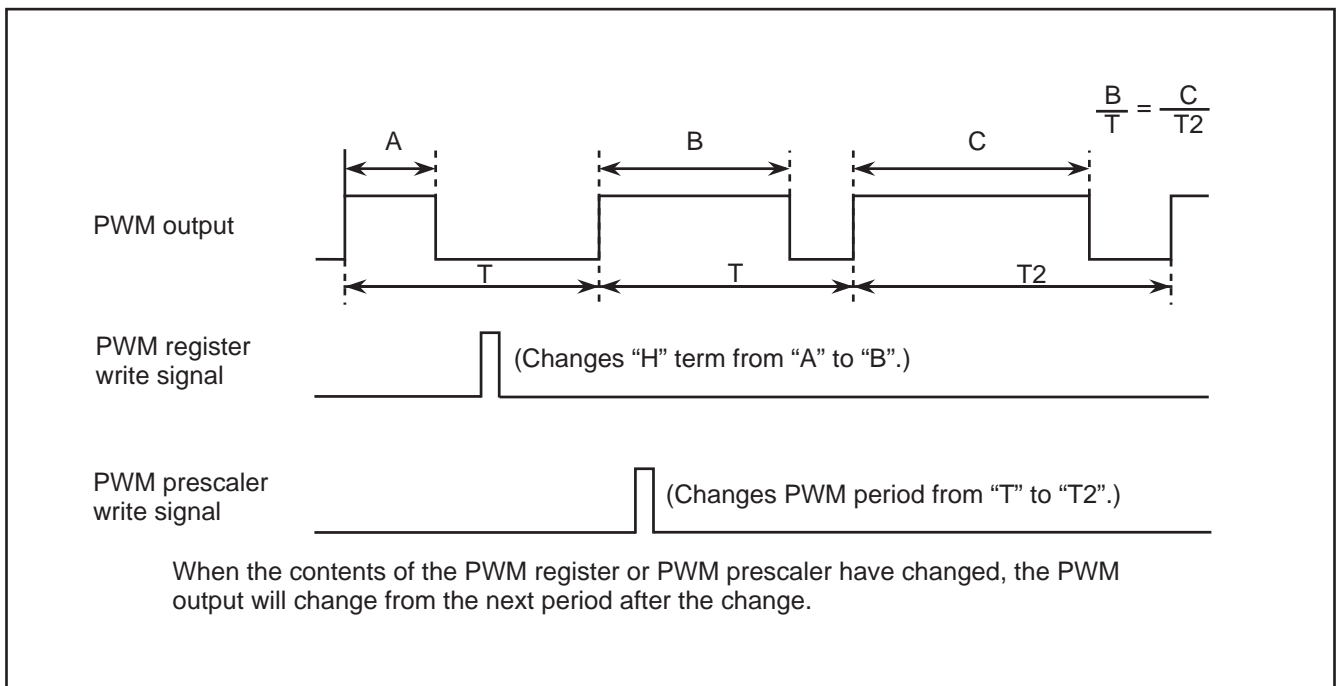


Fig. 38 PWM output timing when PWM register or PWM prescaler is changed

■Note

The PWM starts after the PWM enable bit is set to enable and "L" level is output from the PWM pin.

The length of this "L" level output is as follows:

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

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

HARDWARE

FUNCTIONAL DESCRIPTION

A-D CONVERTER

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

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

[AD Control Register (ADCON)] 003416

The AD control register controls the A-D conversion process. Bits 0 to 2 select a specific analog input pin. Bit 4 indicates the completion of an A-D conversion. The value of this bit remains at "0" during an A-D conversion and changes to "1" when an A-D conversion ends. Writing "0" to this bit starts the A-D conversion.

Comparison Voltage Generator

The comparison voltage generator divides the voltage between AVSS and VREF into 1024 and outputs the divided voltages.

Channel Selector

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

Comparator and Control Circuit

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

Note that because the comparator consists of a capacitor coupling, set $f(X_{IN})$ to 500 kHz or more during an A-D conversion. The M38514E6/M6 can operate at even low-speed mode, because of the A-D converter of the M38514E6/M6 has a built-in self-oscillation circuit.

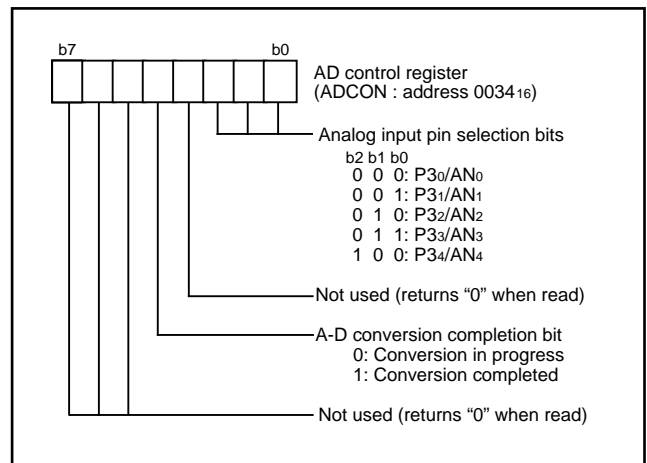


Fig. 39 Structure of AD control register

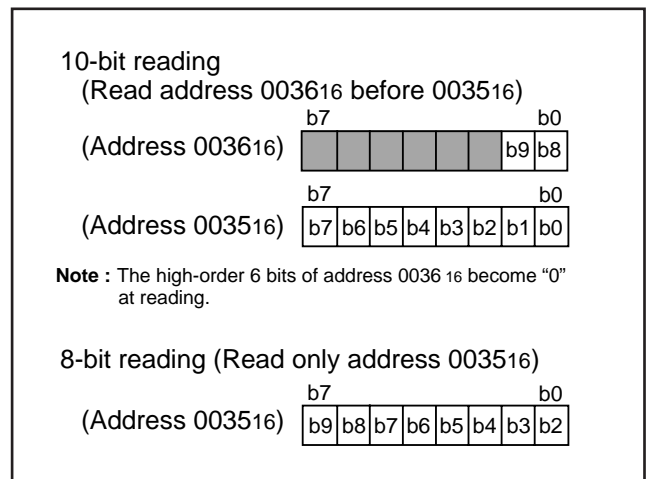


Fig. 40 Structure of A-D conversion registers

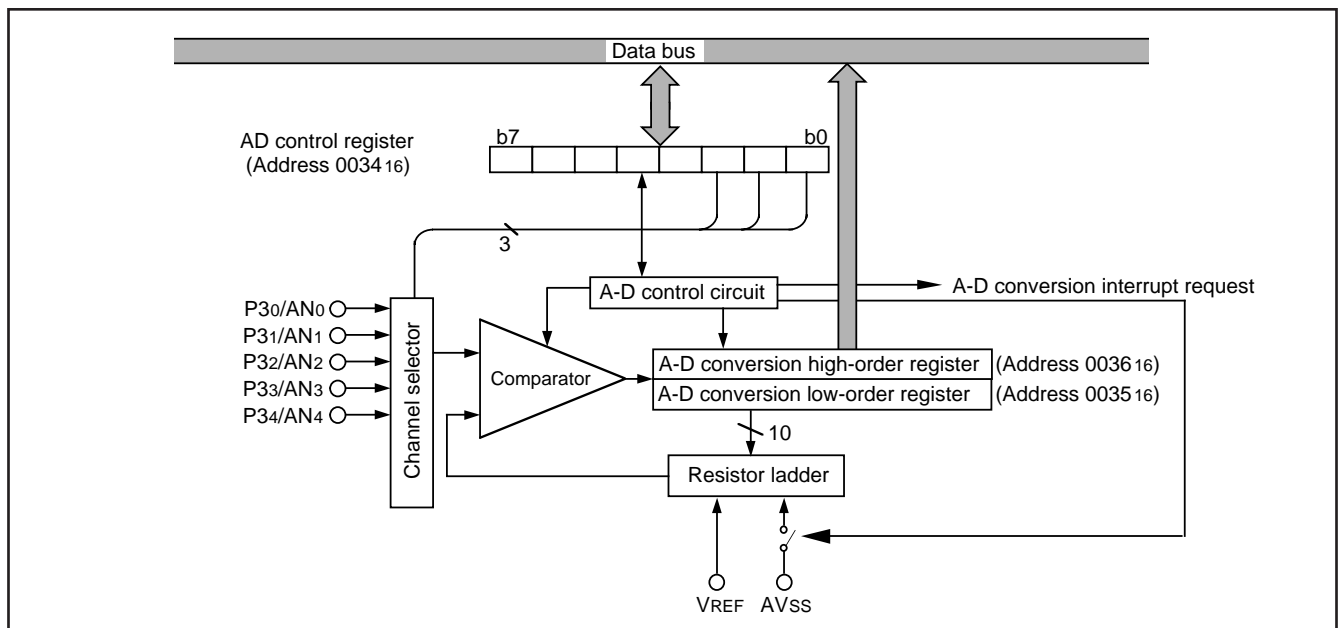


Fig. 41 Block diagram of A-D converter

WATCHDOG TIMER

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

Standard Operation of Watchdog Timer

When any data is not written into the watchdog timer control register (address 0039₁₆) after resetting, the watchdog timer is in the stop state. The watchdog timer starts to count down by writing an optional value into the watchdog timer control register (address 0039₁₆) and an internal reset occurs at an underflow of the watchdog timer H.

Accordingly, programming is usually performed so that writing to the watchdog timer control register (address 0039₁₆) may be started before an underflow. When the watchdog timer control register (address 0039₁₆) is read, the values of the high-order 6 bits of the watchdog timer H, STP instruction disable bit, and watchdog timer H count source selection bit are read.

Initial value of watchdog timer

At reset or writing to the watchdog timer control register (address 0039₁₆), each watchdog timer H and L is set to "FF₁₆".

Watchdog timer H count source selection bit operation

Bit 7 of the watchdog timer control register (address 0039₁₆) permits selecting a watchdog timer H count source. When this bit is set to "0", the count source becomes the underflow signal of watchdog timer L. The detection time is set to 131.072 ms at $f(X_{IN}) = 8$ MHz frequency and 32.768 s at $f(X_{CIN}) = 32$ kHz frequency. When this bit is set to "1", the count source becomes the signal divided by 16 for $f(X_{IN})$ (or $f(X_{CIN})$). The detection time in this case is set to 512 μ s at $f(X_{IN}) = 8$ MHz frequency and 128 ms at $f(X_{CIN}) = 32$ kHz frequency. This bit is cleared to "0" after resetting.

Operation of STP instruction disable bit

Bit 6 of the watchdog timer control register (address 0039₁₆) permits disabling the STP instruction when the watchdog timer is in operation.

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

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

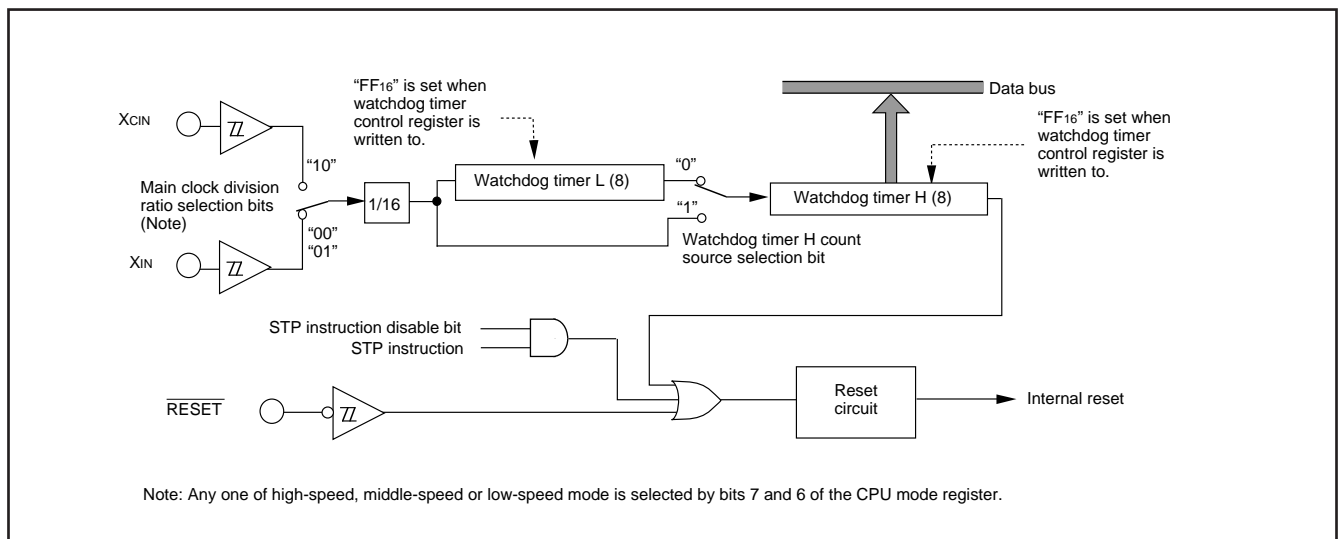


Fig. 42 Block diagram of Watchdog timer

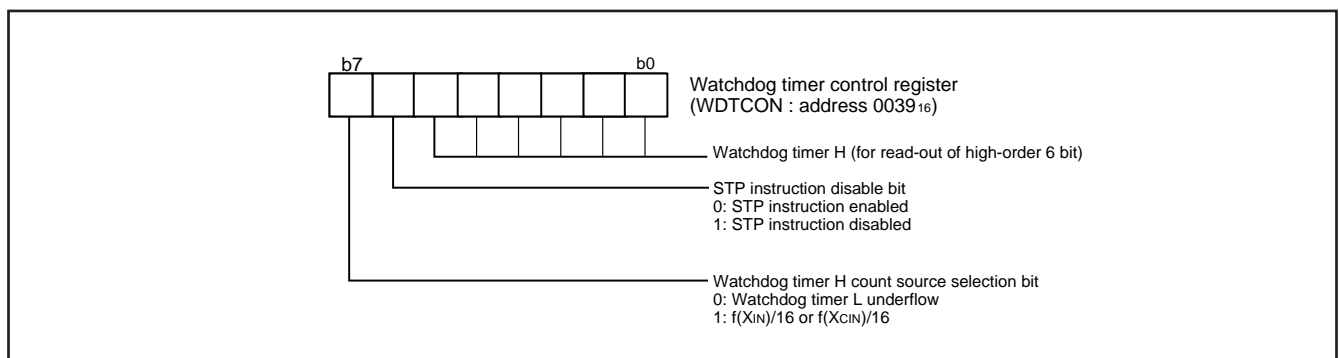


Fig. 43 Structure of Watchdog timer control register

HARDWARE

FUNCTIONAL DESCRIPTION

RESET CIRCUIT

To reset the microcomputer, $\overline{\text{RESET}}$ pin must 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 must be between 2.7 V and 5.5 V, and the oscillation must be stable), reset is released. After the reset is completed, the program starts from the address FFFD₁₆ (high-order byte) and address FFFC₁₆ (low-order byte). Make sure that the reset input voltage is less than 0.54 V for Vcc of 2.7 V.

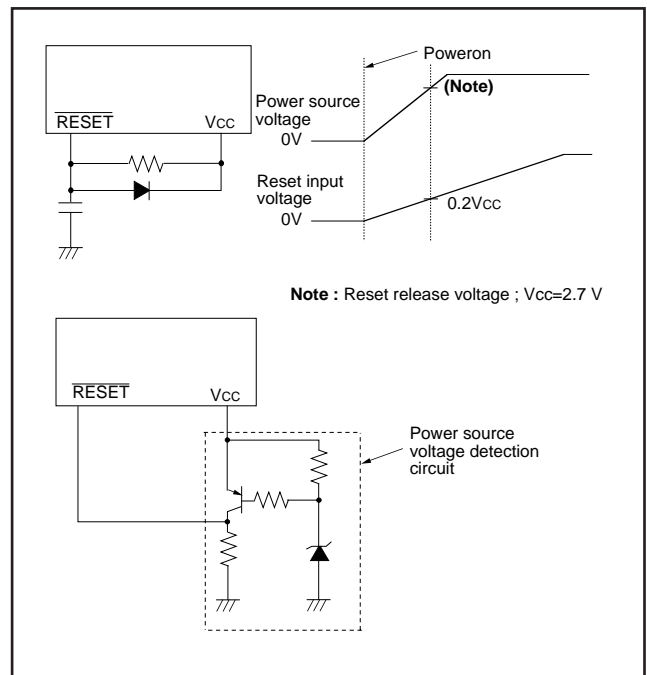


Fig. 44 Reset circuit example

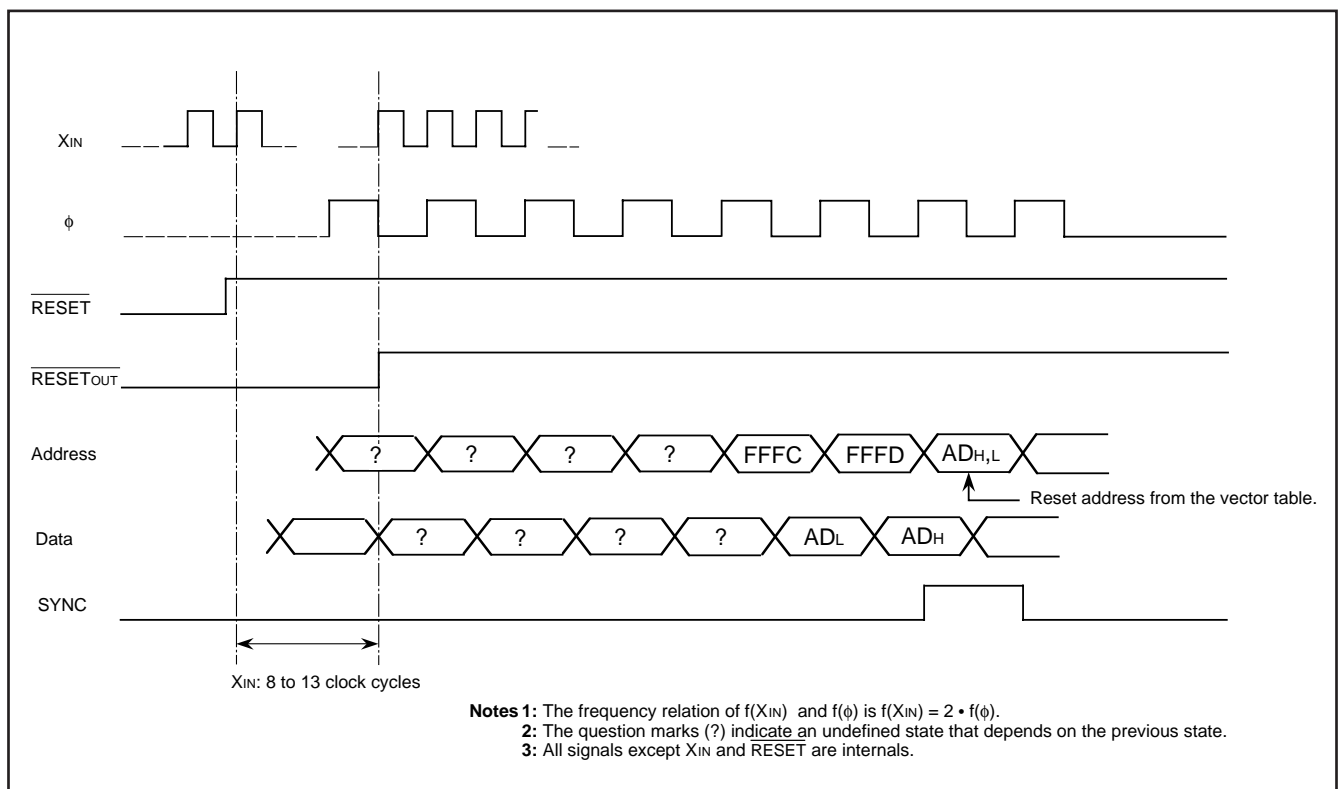


Fig. 45 Reset sequence

	Address	Register contents
(1) Port P0 direction register (P0D)	0001 ₁₆	00 ₁₆
(2) Port P1 direction register (P1D)	0003 ₁₆	00 ₁₆
(3) Port P2 direction register (P2D)	0005 ₁₆	00 ₁₆
(4) Port P3 direction register (P3D)	0007 ₁₆	00 ₁₆
(5) Port P4 direction register (P4D)	0009 ₁₆	00 ₁₆
(6) Serial I/O status register (SIOSTS)	0019 ₁₆	1 0 0 0 0 0 0 0 0 0
(7) Serial I/O control register (SIOCON)	001A ₁₆	00 ₁₆
(8) UART control register (UARTCON)	001B ₁₆	1 1 1 0 0 0 0 0 0 0
(9) PWM control register (PWMCON)	001D ₁₆	00 ₁₆
(10) Prescaler 12 (PRE12)	0020 ₁₆	FF ₁₆
(11) Timer 1 (T1)	0021 ₁₆	01 ₁₆
(12) Timer 2 (T2)	0022 ₁₆	00 ₁₆
(13) Timer XY mode register (TM)	0023 ₁₆	00 ₁₆
(14) Prescaler X (PREX)	0024 ₁₆	FF ₁₆
(15) Timer X (TX)	0025 ₁₆	FF ₁₆
(16) Prescaler Y (PREY)	0026 ₁₆	FF ₁₆
(17) Timer Y (TY)	0027 ₁₆	FF ₁₆
(18) Timer count source select register	0028 ₁₆	00 ₁₆
(19) I ² C address register (S0D)	002C ₁₆	00 ₁₆
(20) I ² C status register (S1)	002D ₁₆	0 0 0 1 0 0 0 X
(21) I ² C control register (S1D)	002E ₁₆	00 ₁₆
(22) I ² C clock control register (S2)	002F ₁₆	00 ₁₆
(23) I ² C start/stop condition control register (S2D)	0030 ₁₆	0 0 0 X X X X X
(24) AD control register (ADCON)	0034 ₁₆	0 0 0 1 0 0 0 0
(25) MISRG	0038 ₁₆	00 ₁₆
(26) Watchdog timer control register (WDTCON)	0039 ₁₆	0 0 1 1 1 1 1 1
(27) Interrupt edge selection register (INTEDGE)	003A ₁₆	00 ₁₆
(28) CPU mode register (CPUM)	003B ₁₆	0 1 0 0 1 0 0 0
(29) Interrupt request register 1 (IREQ1)	003C ₁₆	00 ₁₆
(30) Interrupt request register 2 (IREQ2)	003D ₁₆	00 ₁₆
(31) Interrupt control register 1 (ICON1)	003E ₁₆	00 ₁₆
(32) Interrupt control register 2 (ICON2)	003F ₁₆	00 ₁₆
(33) Processor status register	(PS)	X X X X X 1 X X
(34) Program counter	(PC _H)	FFF ₁₆ contents
	(PC _L)	FFC ₁₆ contents

Note : X indicates Not fixed .

Fig. 46 Internal status at reset

HARDWARE

FUNCTIONAL DESCRIPTION

CLOCK GENERATING CIRCUIT

The 3851 group has two built-in oscillation circuits. An oscillation circuit can be formed by connecting a resonator between XIN and XOUT (XCIN and XCOUT). Use the circuit constants in accordance with the resonator manufacturer's recommended values. No external resistor is needed between XIN and XOUT since a feed-back resistor exists on-chip. However, an external feed-back resistor is needed between XCIN and XCOUT.

Immediately after power on, only the XIN oscillation circuit starts oscillating, and XCIN and XCOUT pins function as I/O ports.

Frequency Control

(1) Middle-speed mode

The internal clock ϕ is the frequency of XIN divided by 8. After reset, this mode is selected.

(2) High-speed mode

The internal clock ϕ is half the frequency of XIN.

(3) Low-speed mode

The internal clock ϕ is half the frequency of XCIN.

■Note

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

(4) Low power dissipation mode

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

The sub-clock XCIN-XCOUT oscillating circuit can not directly input clocks that are generated externally. Accordingly, make sure to cause an external resonator to oscillate.

Oscillation Control

(1) Stop mode

If the STP instruction is executed, the internal clock ϕ stops at an "H" level, and XIN and XCIN oscillation stops. When the oscillation stabilizing time set after STP instruction released bit is "0," the prescaler 12 is set to "FF16" and timer 1 is set to "0116." When the oscillation stabilizing time set after STP instruction released bit is "1," set the sufficient time for oscillation of used oscillator to stabilize since nothing is set to the prescaler 12 and timer 1.

Either XIN or XCIN divided by 16 is input to the prescaler 12 as count source. Oscillator restarts when an external interrupt is received, but the internal clock ϕ is not supplied to the CPU (remains at "H") until timer 1 underflows. The internal clock ϕ is supplied for the first time, when timer 1 underflows. This ensures time for the clock oscillation using the ceramic resonators to be stabilized. When the oscillator is restarted by reset, apply "L" level to the $\overline{\text{RESET}}$ pin until the oscillation is stable since a wait time will not

be generated.

(2) Wait mode

If the WIT instruction is executed, the internal clock ϕ stops at an "H" level, but the oscillator does not stop. The internal clock ϕ restarts at reset or when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

To ensure that the interrupts will be received to release the STP or WIT state, their interrupt enable bits must be set to "1" before executing of the STP or WIT instruction.

When releasing the STP state, the prescaler 12 and timer 1 will start counting the clock XIN divided by 16. Accordingly, set the timer 1 interrupt enable bit to "0" before executing the STP instruction.

■Note

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

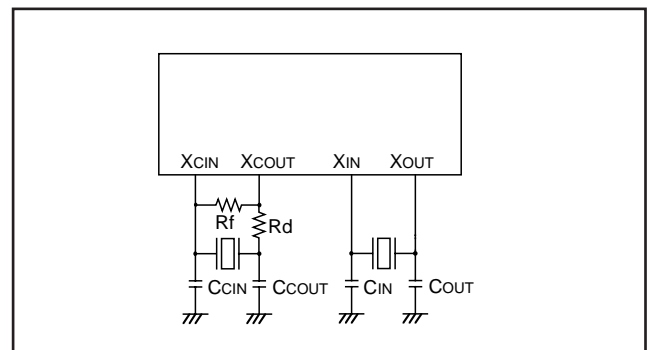


Fig. 47 Ceramic resonator circuit

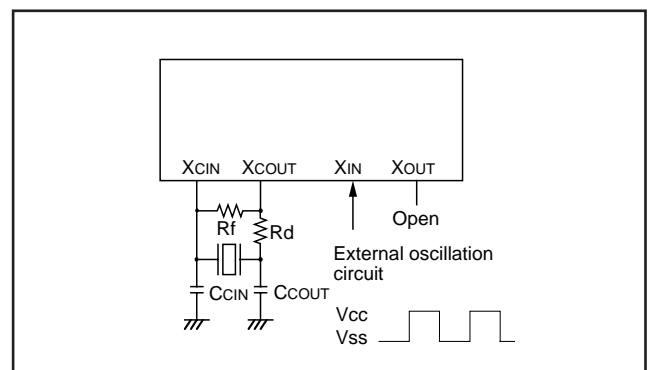


Fig. 48 External clock input circuit

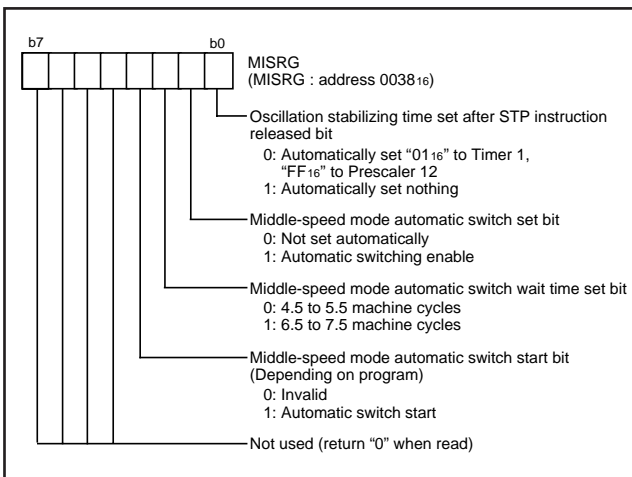


Fig. 49 Structure of MISRG

Middle-speed mode automatic switch set bit

By setting the middle-speed mode automatic switch set bit to "1" while operating in the low-speed mode, XIN oscillation automatically starts and the mode is automatically switched to the middle-speed mode when detecting a rising/falling edge of the SCL or SDA pin. The middle-speed automatic switch wait time set bit can select the switch timing from the low-speed to the middle-speed mode; either 4.5 to 5.5 machine cycles or 6.5 to 7.5 machine cycles in the low-speed mode. Select it according to oscillation start characteristics of used XIN oscillator.

The middle-speed mode automatic switch start bit is used to automatically make to XIN oscillation start and switch to the middle-speed mode by setting this bit to "1" while operating in the low-speed mode.

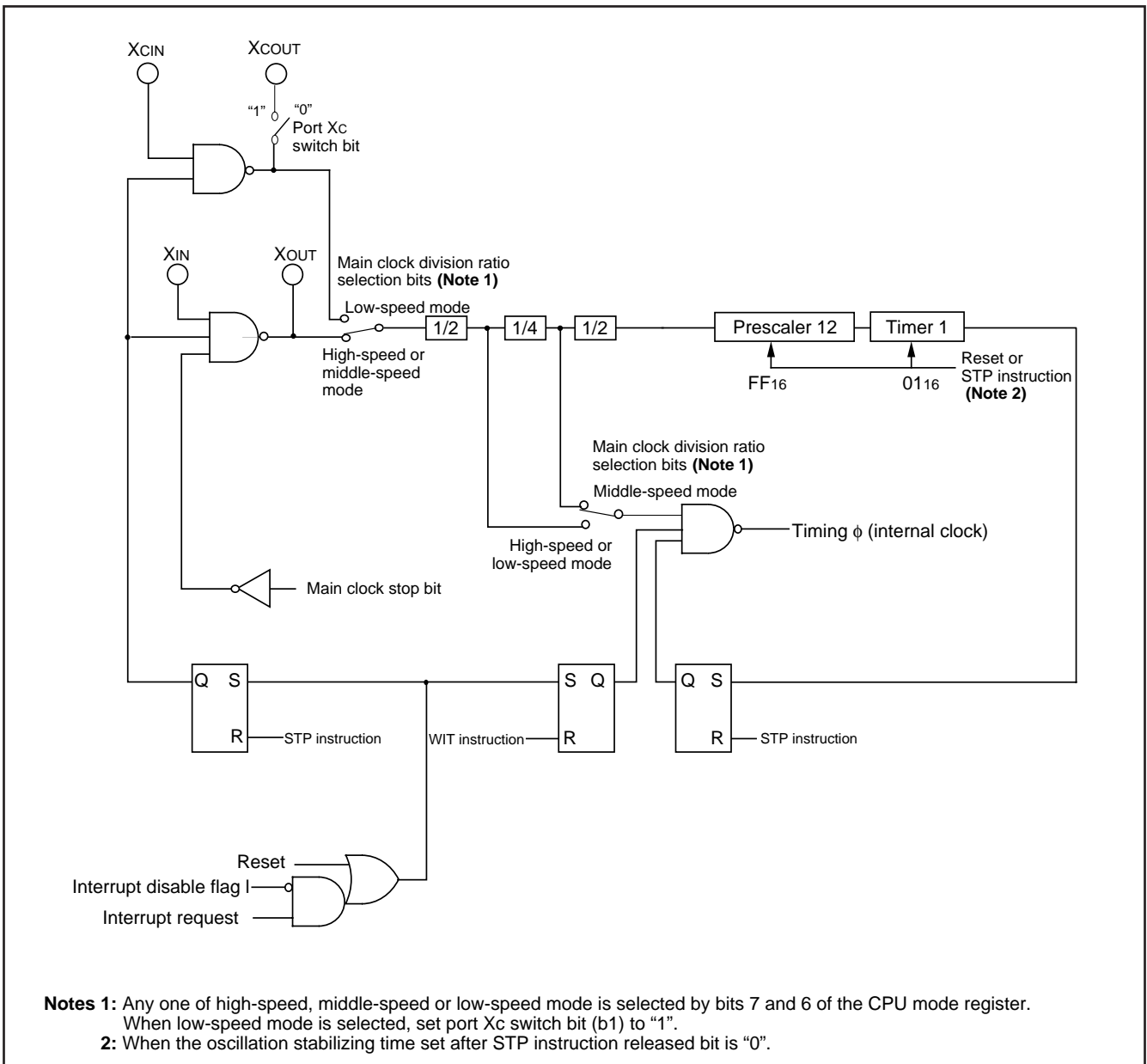


Fig. 50 System clock generating circuit block diagram (Single-chip mode)

HARDWARE

FUNCTIONAL DESCRIPTION

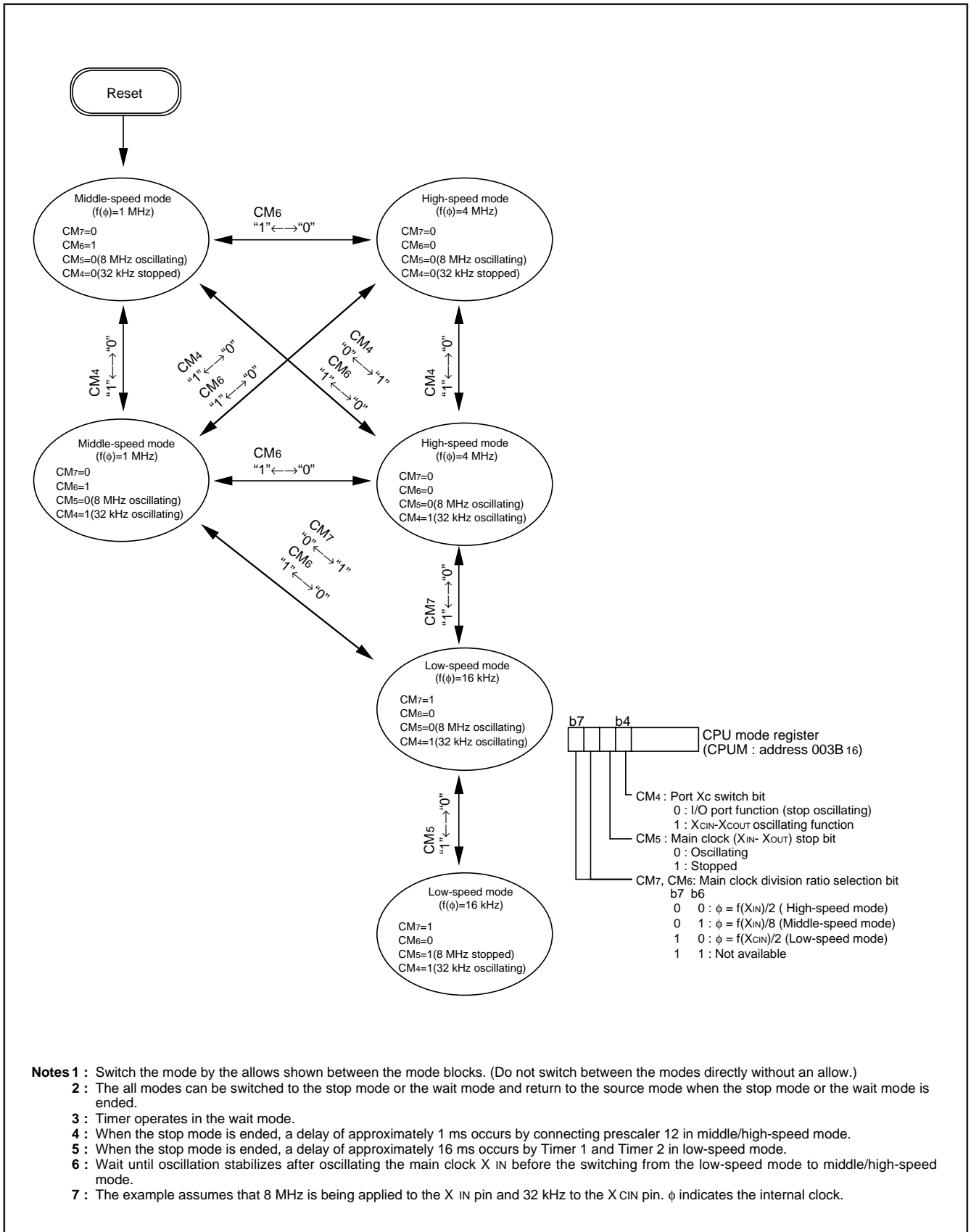


Fig. 51 State transitions of system clock

NOTES ON PROGRAMMING

Processor Status Register

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

Interrupts

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

Decimal Calculations

- To calculate in decimal notation, set the decimal mode flag (D) to "1", then execute an ADC or SBC instruction. After executing an ADC or SBC instruction, execute at least one instruction before executing a SEC, CLC, or CLD instruction.
- In decimal mode, the values of the negative (N), overflow (V), and zero (Z) flags are invalid.

Timers

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

Multiplication and Division Instructions

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

Ports

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

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

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

Serial I/O

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

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

When an external clock is used as synchronous clock in serial I/O, write transmission data to the transmit buffer register while the transfer clock is "H."

A-D Converter

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

Therefore, make sure that $f(XIN)$ is at least on 500 kHz during an A-D conversion.

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

Instruction Execution Time

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

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

The frequency of the internal clock ϕ is half of the XIN frequency in high-speed mode.

HARDWARE

DATA REQUIRED FOR MASK ORDERS/ROM PROGRAMMING METHOD

DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

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

DATA REQUIRED FOR ROM WRITING ORDERS

The following are necessary when ordering a ROM writing:

- 1.ROM Writing 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 12 Programming adapter

Package	Name of Programming Adapter
42P2R-A	PCA4738F-42A
42P4B	PCA4738S-42A

The PROM of the blank One Time PROM version is not tested or screened in the assembly process and following processes. To ensure proper operation after programming, the procedure shown in Figure 52 is recommended to verify programming.

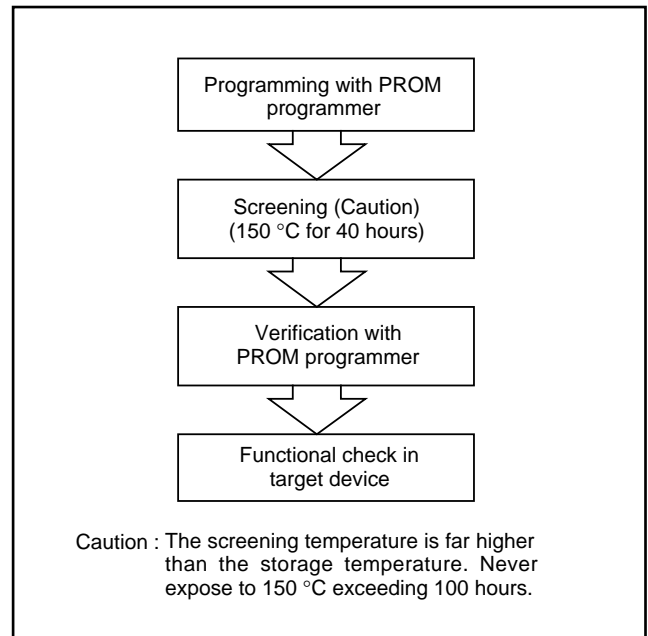


Fig. 52 Programming and testing of One Time PROM version

FUNCTIONAL DESCRIPTION SUPPLEMENT

Interrupt

3851 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 13".

Table 13 Interrupt sources, vector addresses and interrupt priority

Priority	Interrupt sources	Vector addresses		Remarks
		High-order	Low-order	
1	Reset (Note 1)	FFFD ₁₆	FFFC ₁₆	Non-maskable
2	INT0 interrupt	FFFB ₁₆	FFFA ₁₆	External interrupt (active edge selectable)
3	SCL, SDA	FFF9 ₁₆	FFF8 ₁₆	External interrupt (active edge selectable)
4	INT1 interrupt	FFF7 ₁₆	FFF6 ₁₆	External interrupt (active edge selectable)
5	INT2 interrupt	FFF5 ₁₆	FFF4 ₁₆	External interrupt (active edge selectable)
6	INT3 interrupt	FFF3 ₁₆	FFF2 ₁₆	External interrupt (active edge selectable)
7	I ² C interrupt	FFF1 ₁₆	FFF0 ₁₆	
8	Timer X interrupt	FFEF ₁₆	FFEE ₁₆	
9	Timer Y interrupt	FFED ₁₆	FFEC ₁₆	
10	Timer 1 interrupt	FFEB ₁₆	FFEA ₁₆	STP instruction release timer underflow
11	Timer 2 interrupt	FFE9 ₁₆	FFE8 ₁₆	
12	Serial I/O receive interrupt	FFE7 ₁₆	FFE6 ₁₆	Valid when serial I/O is selected
13	Serial I/O transmit interrupt	FFE5 ₁₆	FFE4 ₁₆	Valid when serial I/O is selected
14	CNTR0 interrupt	FFE3 ₁₆	FFE2 ₁₆	External interrupt (active edge selectable)
15	CNTR1 interrupt	FFE1 ₁₆	FFE0 ₁₆	External interrupt (active edge selectable)
16	A-D conversion interrupt	FFDF ₁₆	FFDE ₁₆	
17	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.

HARDWARE

FUNCTIONAL DESCRIPTION SUPPLEMENT

Timing After Interrupt

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

Figure 53 shows a timing chart after an interrupt occurs, and Figure 54 shows the time up to execution of the interrupt processing routine.

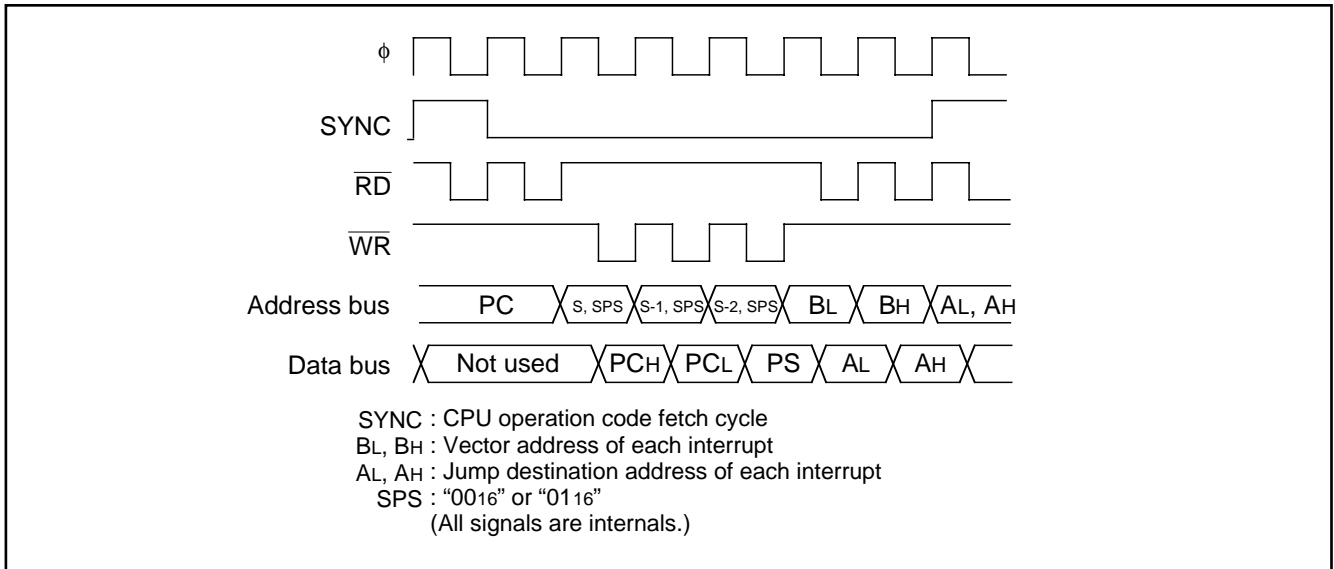


Fig. 53 Timing chart after an interrupt occurs

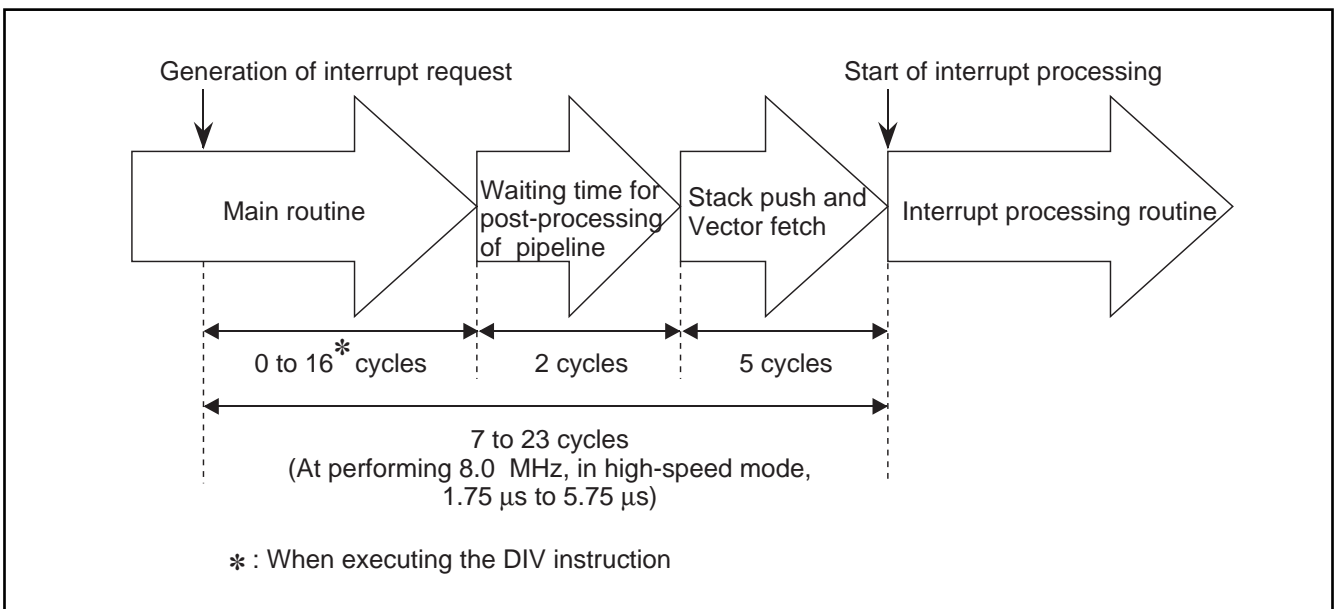


Fig. 54 Time up to execution of the interrupt processing routine

A-D Converter

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

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

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

A-D conversion completes at 61 clock cycles (15.25μs at f(XIN) = 8.0 MHz) after it is started, and the result of the conversion is stored into the A-D conversion register.

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

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

When n = 0 Vref = 0

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

n : the value of A-D converter (decimal numeral)

Table 14 Change of A-D conversion register during A-D conversion

	Change of A-D conversion register	Value of comparison voltage (Vref)
At start of conversion	0 0 0 0 0 0 0 0 0 0 0	0
First comparison	1 0 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$
Second comparison	*1 1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4}$
Third comparison	*1 *2 1 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8}$
⋮	⋮	⋮
After completion of tenth comparison	A result of A-D conversion *1 *2 *3 *4 *5 *6 *7 *8 *9 *10	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \dots \pm \frac{V_{REF}}{1024}$

*1—*10: A result of the first to tenth comparison

HARDWARE

FUNCTIONAL DESCRIPTION SUPPLEMENT

Figure 55 shows A-D conversion equivalent circuit, and Figure 56 shows A-D conversion timing chart.

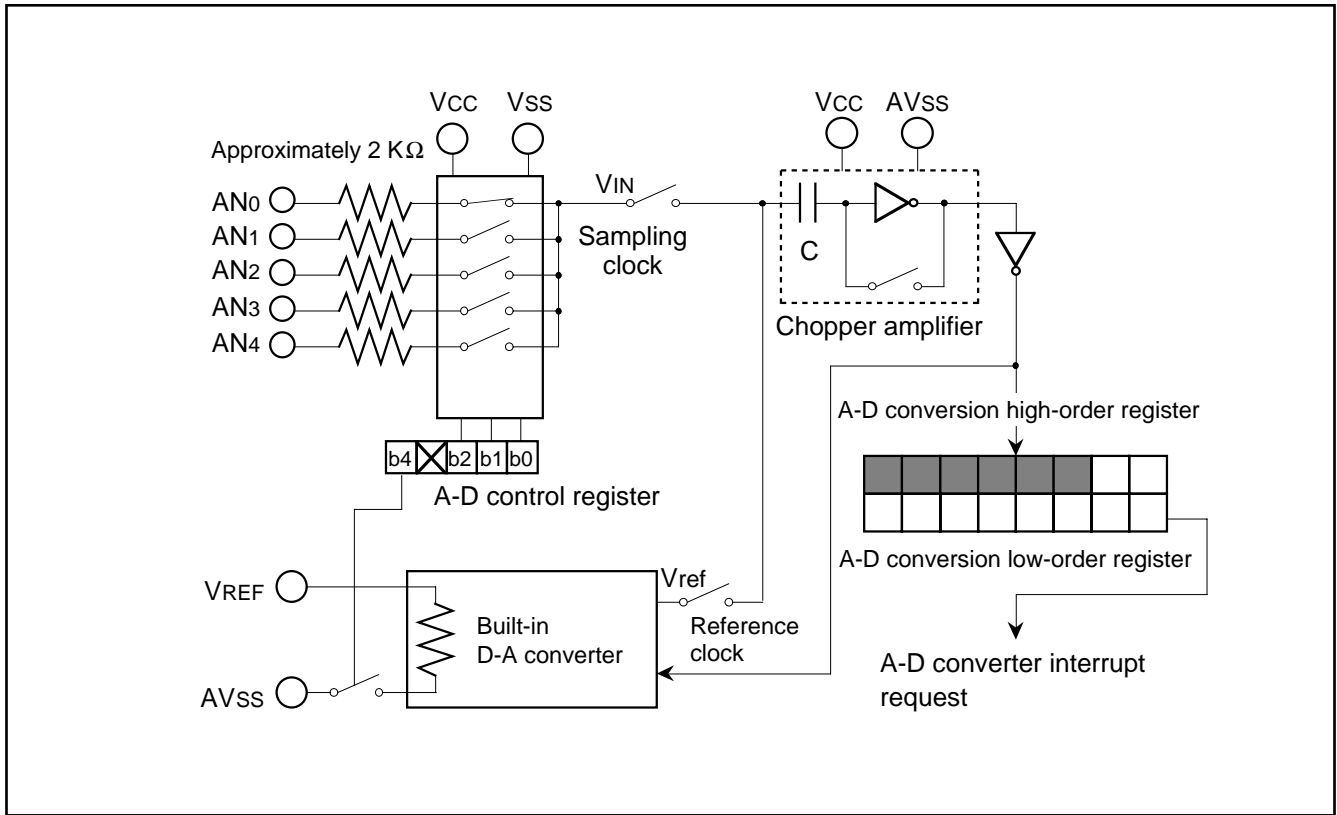


Fig. 55 A-D conversion equivalent circuit

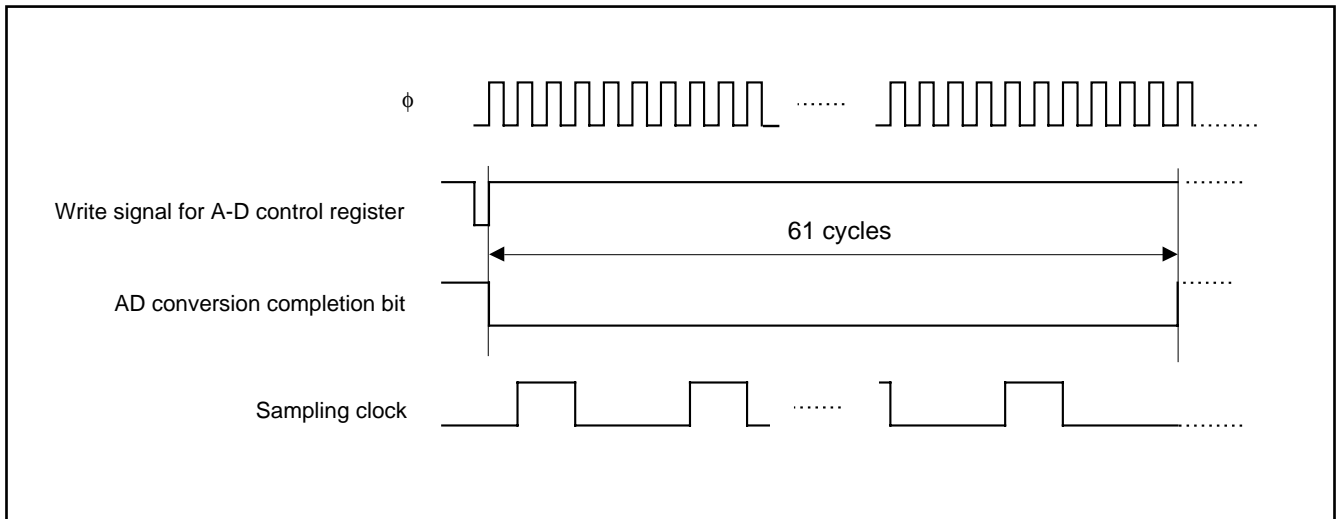


Fig. 56 A-D conversion timing chart

MISRG

(1) Oscillation stabilizing time set after STP instruction released bit (bit 0 of address 003816)

Usually, when the MCU stops the clock oscillation by the STP instruction and the STP instruction has been released by an external interrupt source, the fixed values of Timer 1 and Prescaler 12 (Timer 1 = 0116, Prescaler 12 = FF16) are automatically reloaded in order for the oscillation to stabilize. The user can inhibit the automatic setting by writing "1" to bit 0 of MISRG (address 003816).

However, by setting this bit to "1", the previous values, set just before the STP instruction was executed, will remain in Timer 1 and Prescaler 12. Therefore, you will need to set an appropriate value to each register, in accordance with the oscillation stabilizing time, before executing the STP instruction.

Figure 57 shows the structure of MISRG.

(2) Middle-speed mode automatic switch function

In order to switch the clock mode of an MCU which has a subclock, the following procedure is necessary:

set CPU mode register (003B16) --> start main clock oscillation --> wait for oscillation stabilization --> switch to middle-speed mode (or high-speed mode).

However, the 3851 group has the built-in function which automatically switches from low to middle-speed mode either by the SCL/SDA interrupt or by program.

Figure 58 shows the structure of the I²C start/stop condition control register.

- Middle-speed mode automatic switch by SCL/SDA Interrupt
The SCL/SDA interrupt source enables an automatic switch when the middle-speed mode automatic switch set bit (bit 1) of MISRG (address 003816) is set to "1". The conditions for an automatic switch execution depend on the settings of bits 5 and 6 of the I²C start/stop condition control register (address 003016). Bit 5 is the SCL/SDA interrupt pin polarity selection bit and bit 6 is the SCL/SDA interrupt pin selection bit. The main clock oscillation stabilizing time can also be selected by middle-speed mode automatic switch wait time set bit (bit 2) of the MISRG.
- Middle-speed mode automatic switch by program
The middle-speed mode can also be automatically switched by program while operating in low-speed mode. By setting the middle-speed automatic switch start bit (bit 3) of MISRG (address 003816) to "1" while operating in low-speed mode, the MCU will automatically switch to middle-speed mode. In this case, the oscillation stabilizing time of the main clock can be selected by the middle-speed automatic switch wait set bit (bit 2) of MISRG (address 003816).

HARDWARE

FUNCTIONAL DESCRIPTION SUPPLEMENT

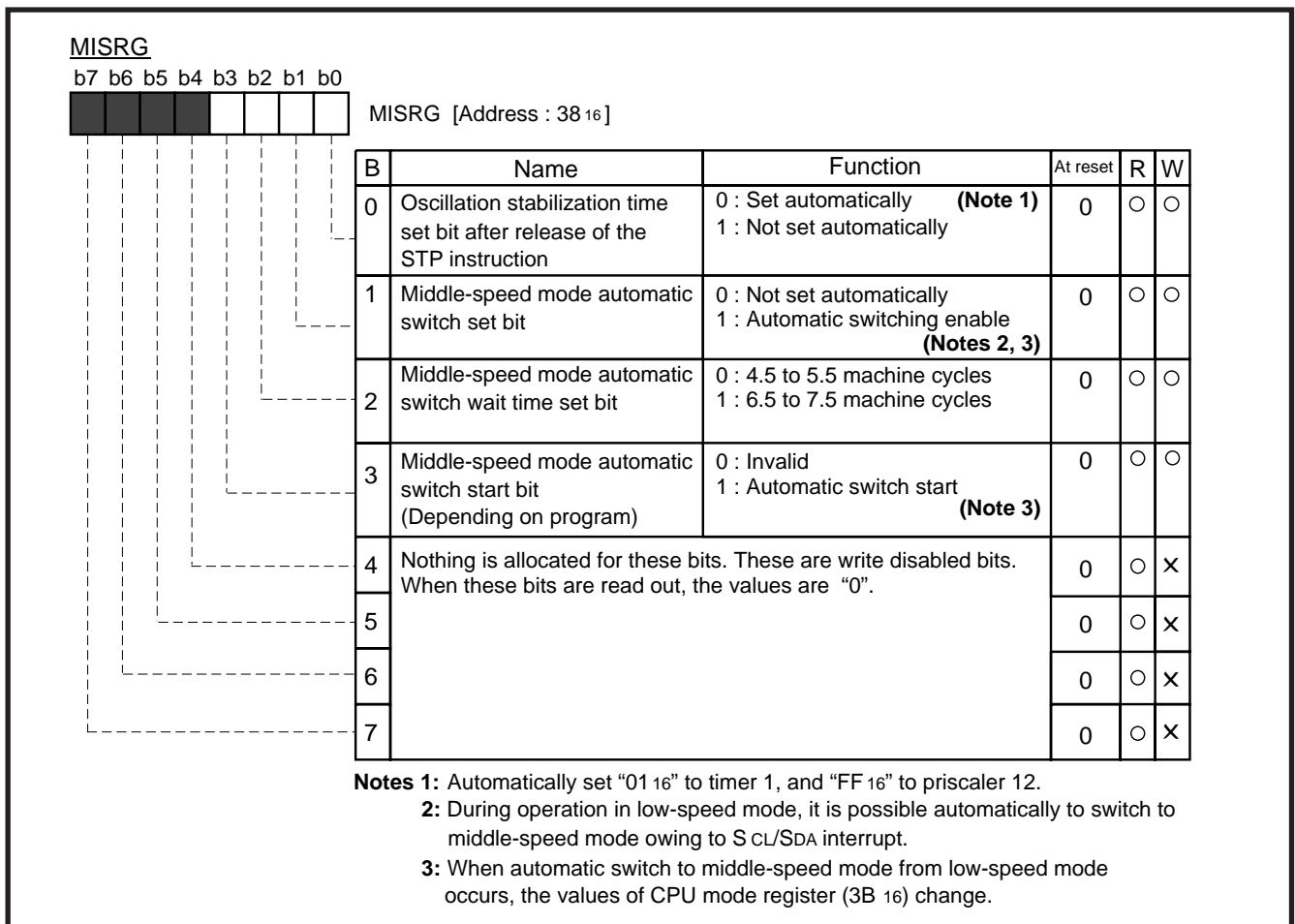


Fig. 57 Structure of MISRG

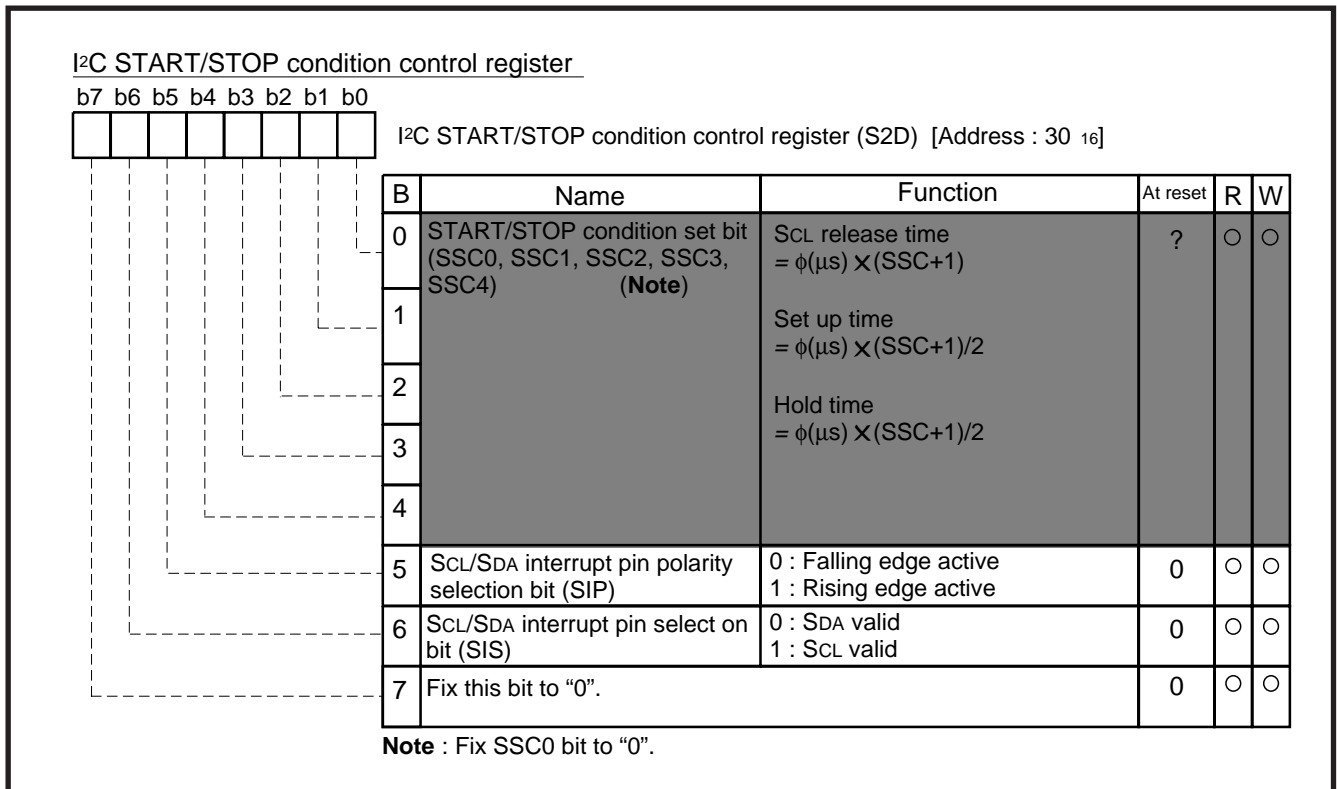


Fig. 58 Structure of I²C START/STOP condition control register

3850 group

Differences between 3850 and 3851 groups

3850 group MCUs do not have the built-in I²C-bus as in the 3851 group. Accordingly, the 3850 group does not have registers relevant to I²C-bus interface for the SFR area. The structure of the interrupt control registers also differs. The following is a list of registers which are not included in the 3850 Group.

- (1) I²C data shift register (address 002B₁₆)
- (2) I²C address register (address 002C₁₆)
- (3) I²C status register (address 002D₁₆)
- (4) I²C control register (address 002E₁₆)
Fix ES0 bit (bit3) to "0".
- (5) I²C clock control register (address 002F₁₆)
- (6) I²C START/STOP condition control register (address 0030₁₆)
- (7) SCL/SDA interrupt request bit (bit1) of Interrupt request register 1 (address 003C₁₆)
- (8) I²C interrupt request bit (bit5) of Interrupt request register 1 (address 003C₁₆)
- (9) SCL/SDA interrupt enable bit (bit1) of Interrupt control register 1 (address 003E₁₆)
Fix this bit to "0".
- (10) I²C interrupt enable bit (bit5) of Interrupt control register 1 (address 003E₁₆)
Fix this bit to "0".

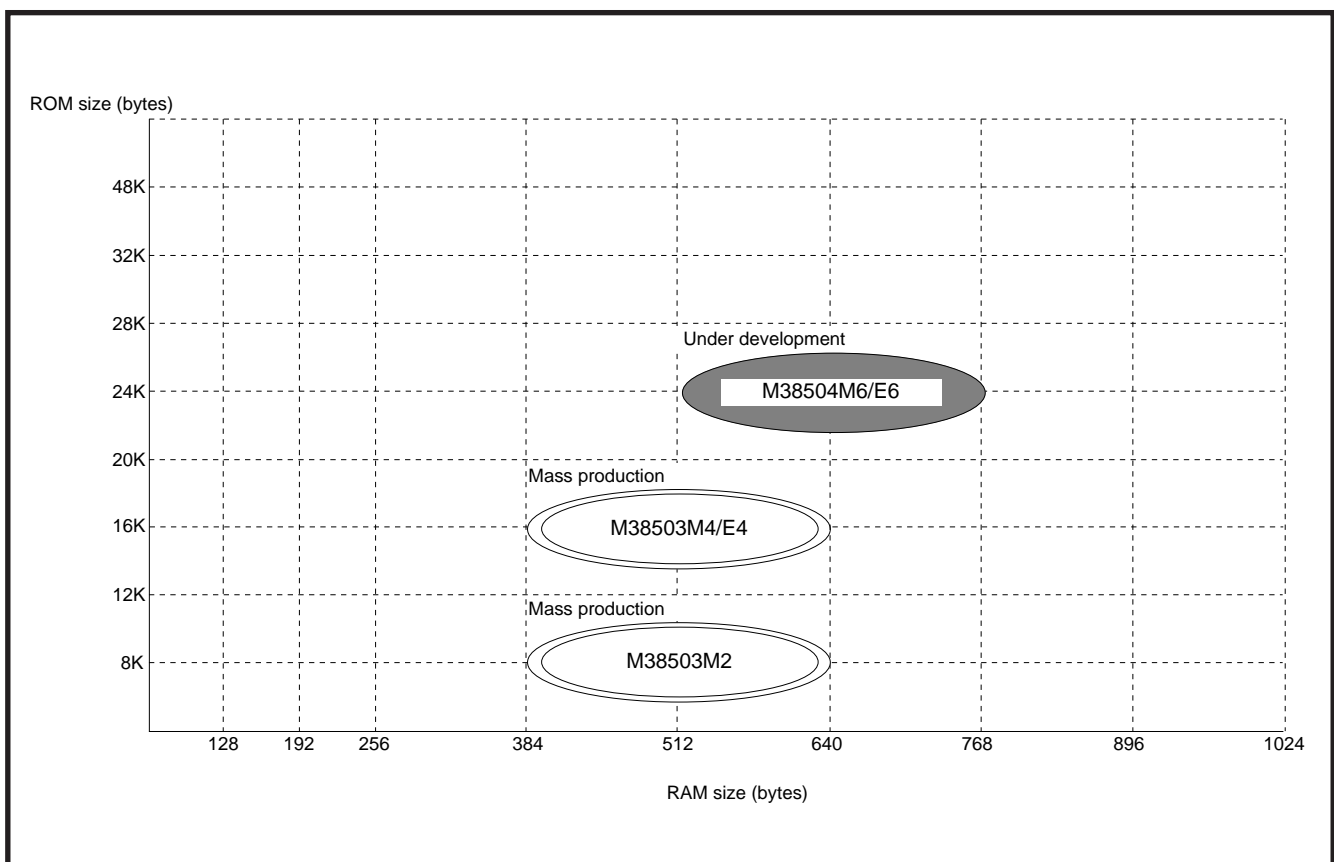


Fig. 59 Memory expansion plan of 3850 group

HARDWARE

FUNCTIONAL DESCRIPTION SUPPLEMENT

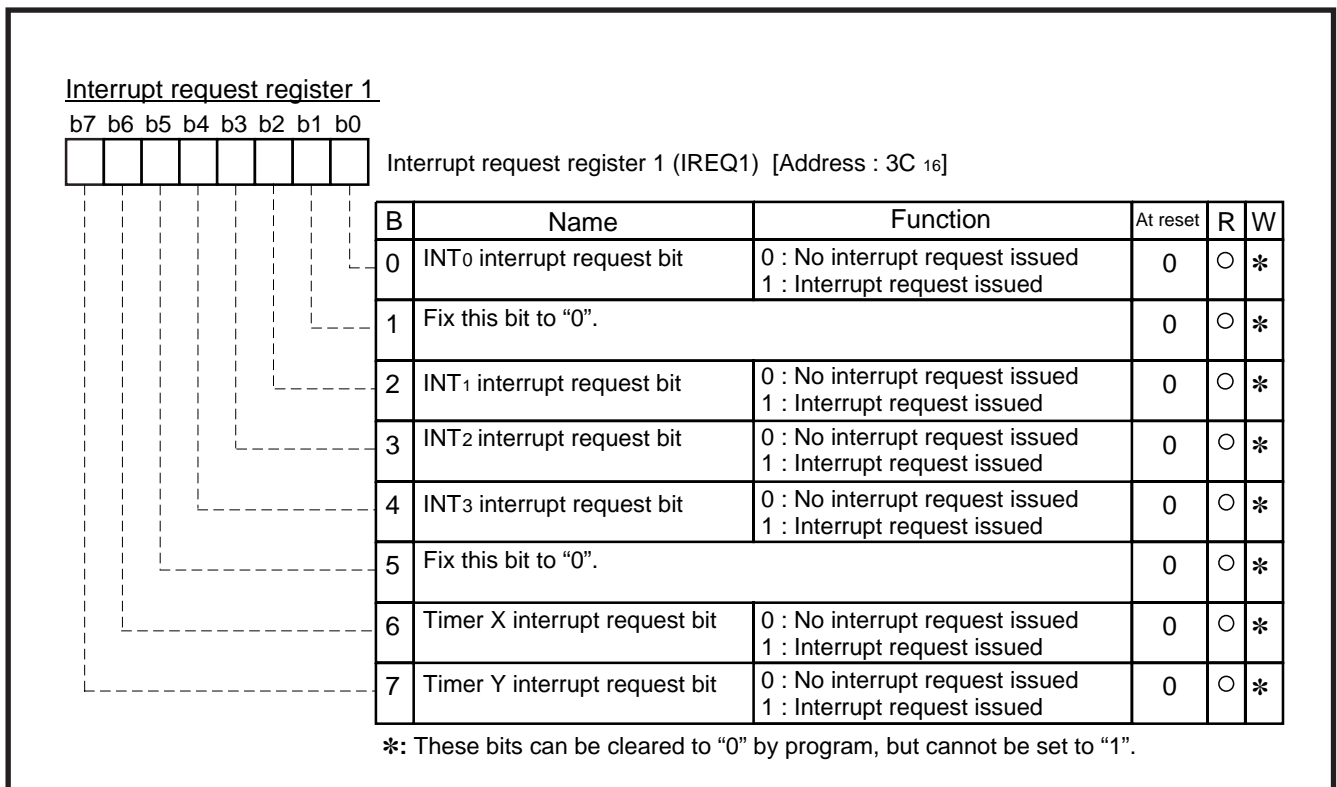


Fig. 60 Structure of Interrupt request register 1 of 3850 group

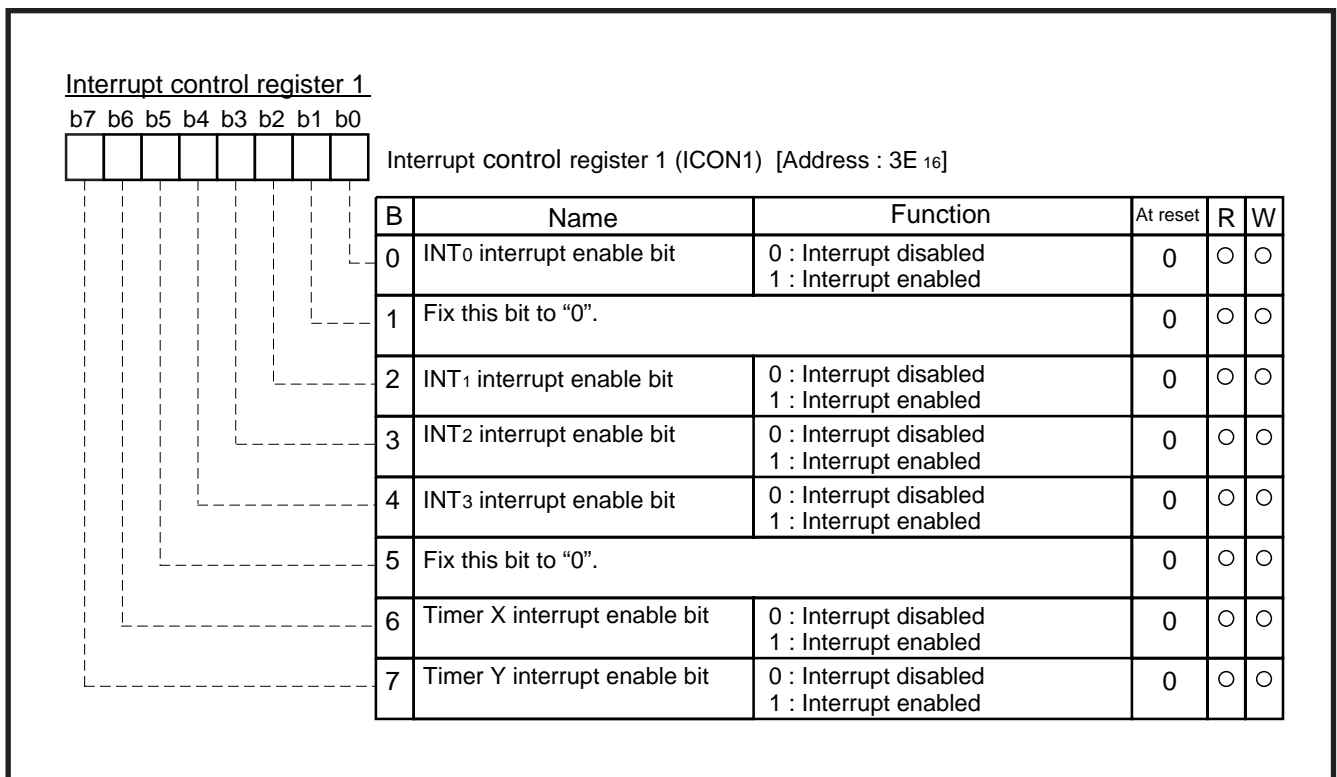


Fig. 61 Structure of Interrupt control register 1 of 3850 group



CHAPTER 2

APPLICATION

- 2.1 I/O port
- 2.2 Timer
- 2.3 Serial I/O
- 2.4 Multi-master I²C-BUS interface
- 2.5 PWM
- 2.6 A-D converter
- 2.7 Reset

APPLICATION

2.1 I/O port

2.1 I/O port

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

2.1.1 Memory map

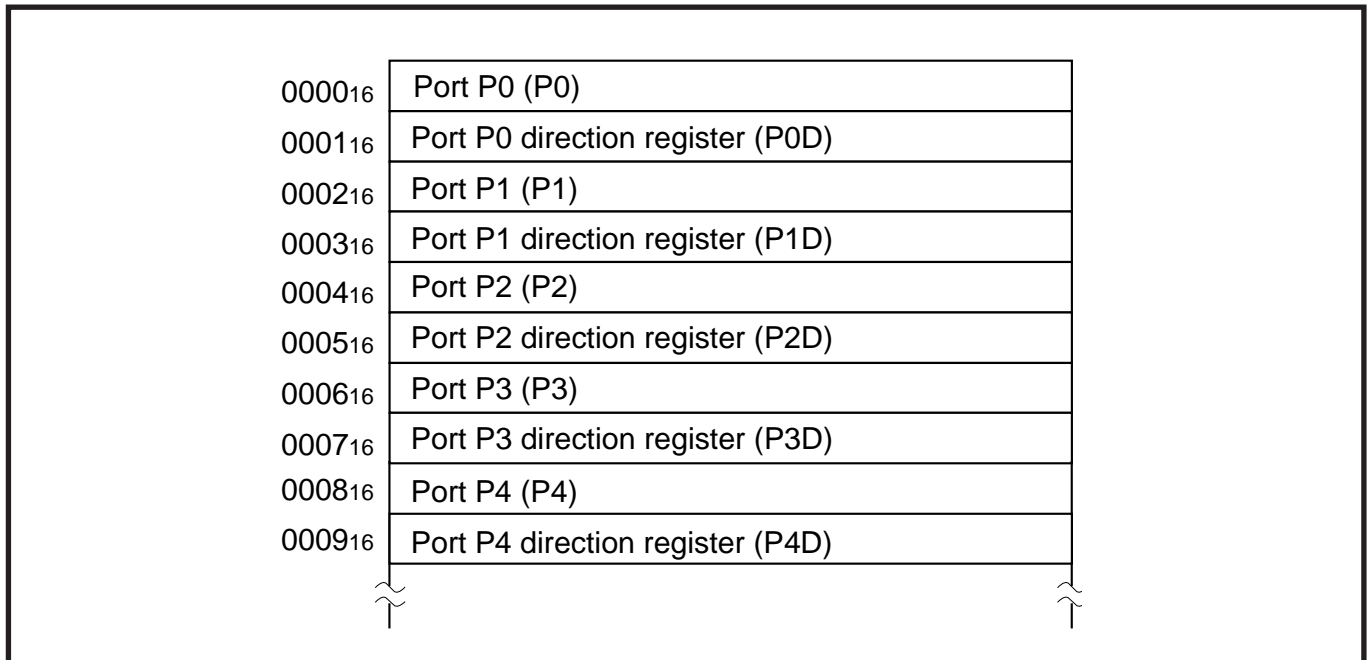


Fig. 2.1.1 Memory map of registers relevant to I/O port

2.1.2 Relevant registers

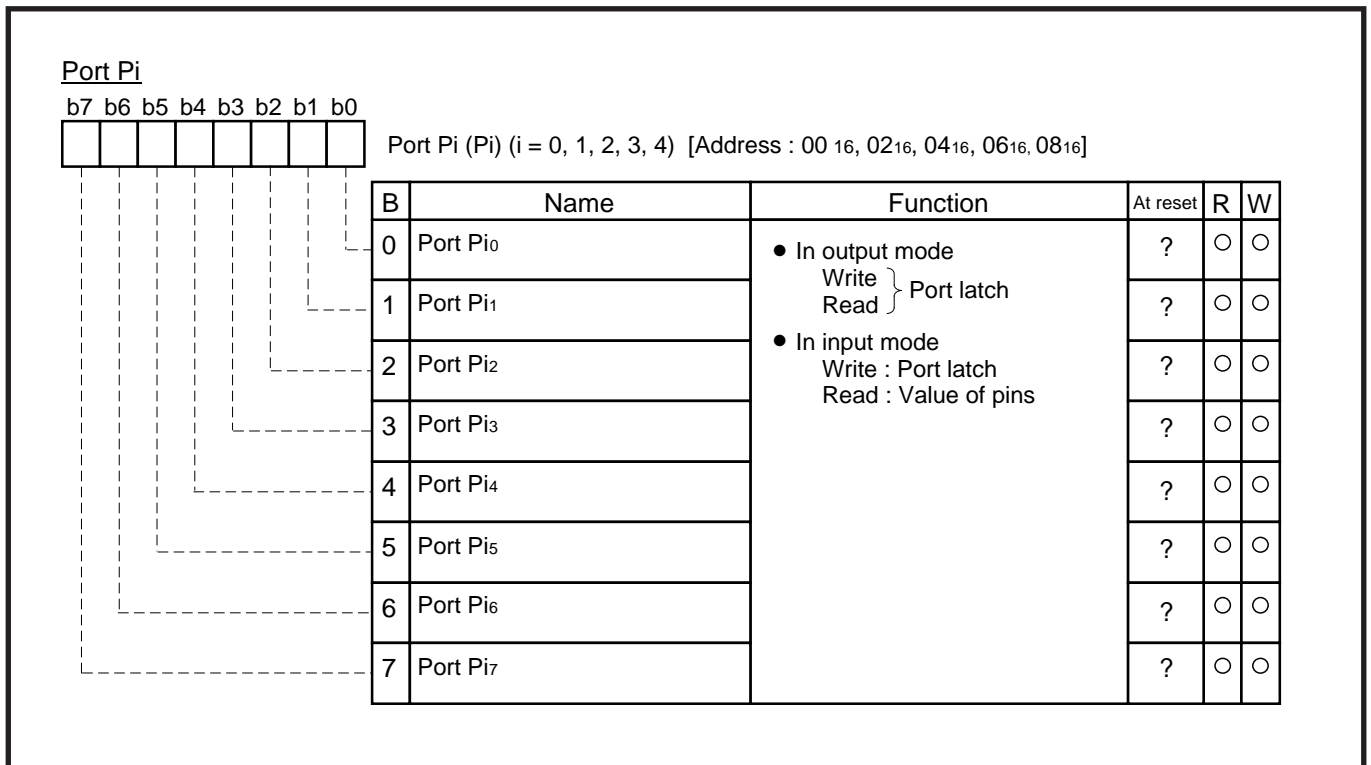


Fig. 2.1.2 Structure of Port Pi (i = 0, 1, 2, 3, 4)

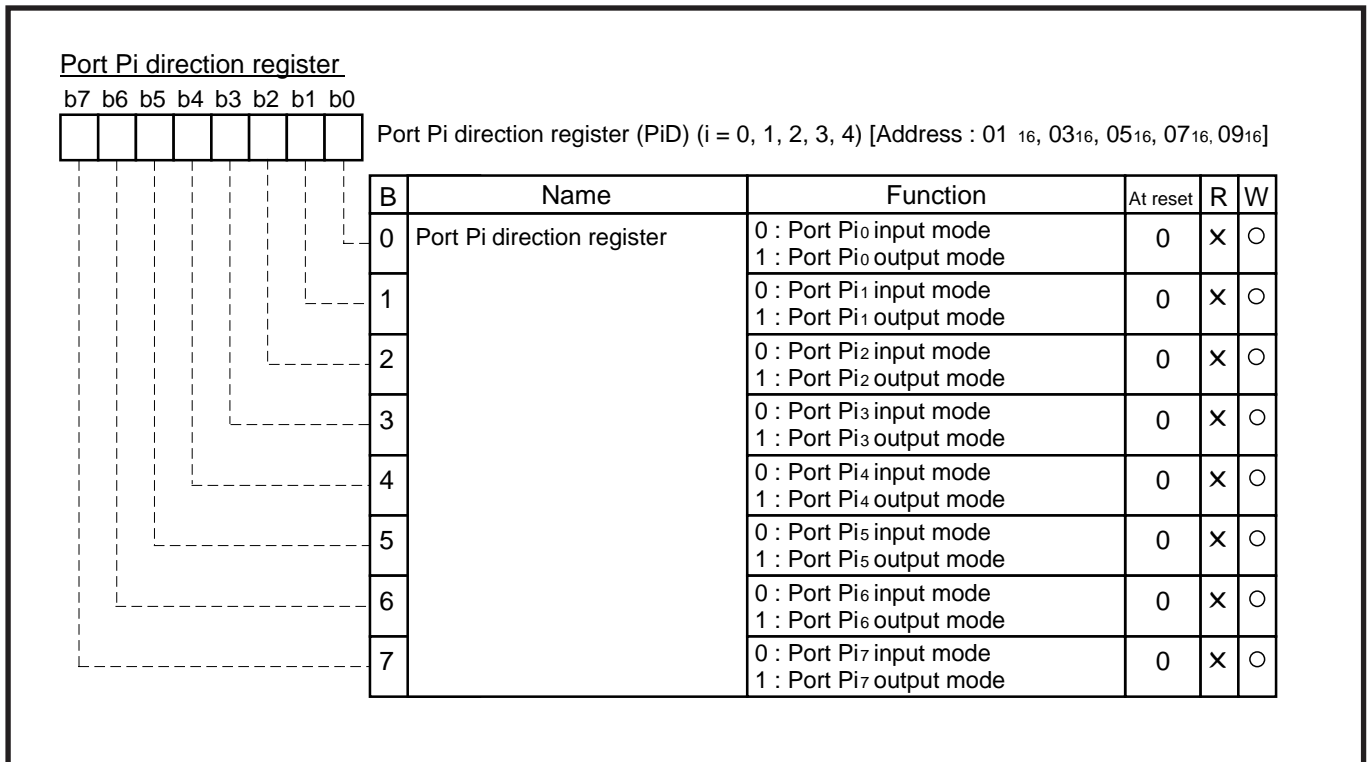


Fig. 2.1.3 Structure of Port Pi direction register (i=0, 1, 2, 3, 4)

2.1.3 Handling of unused pins

Table 2.1.1 Handling of unused pins

Pins/Ports name	Handling
P0, P1, P2, P3, P4	<ul style="list-style-type: none"> •Set to the input mode and connect each to Vcc or Vss through a resistor of 1 kΩ to 10 kΩ. •Set to the output mode and open at “L” or “H” level.
V _{REF}	•Connect to Vss (GND).
AV _{SS}	•Connect to Vss (GND).
X _{OUT}	•Open, only when using an external clock

APPLICATION

2.1 I/O port

2.1.4 Notes on input and output pins

(1) Notes in stand-by state

In stand-by state*¹ for low-power dissipation, do not make input levels of an input port and an I/O port “undefined”, especially for I/O ports of the P-channel and the N-channel open-drain.

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

When determining a resistance value, note the following points:

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

● Reason

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

- P-channel when the content of the port latch is “0”
- 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.

Note 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 a input port and an I/O port are “undefined”. This may cause power source current.

*¹ stand-by state : the stop mode by executing the **STP** instruction
the wait mode by executing the **WIT** instruction

(2) Modifying output data with bit managing instruction

When the port latch of an I/O port is modified with the bit managing instruction*², the value of the unspecified bit may be changed.

● Reason

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

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

Note the following :

- Even when a port which is set as an output port is changed for an input port, its port latch holds the output data.
- As for a bit of the port latch which is set for an input port, its value may be changed even when not specified with a bit managing instruction in case where the pin state differs from its port latch contents.

*² bit managing instructions : **SEB**, and **CLB** instructions

2.1.5 Termination of unused pins

(1) Terminate unused pins

- ① Output ports : Open
- ② Input ports :
Connect each pin to VCC or VSS through each resistor of 1 k Ω to 10 k Ω .
As for pins whose potential affects to operation modes such as pins CNVSS, INT or others, select the VCC pin or the VSS pin according to their operation mode.
- ③ I/O ports :
 - Set the I/O ports for the input mode and connect them to VCC or VSS through each resistor of 1 k Ω to 10 k Ω .
Set the I/O ports for the output mode and open them at "L" or "H".
 - When opening them in the output mode, the input mode of the initial status remains until the mode of the ports is switched over to the output mode by the program after reset. Thus, the potential at these pins is undefined and the power source current may increase in the input mode. With regard to an effects on the system, thoroughly perform system evaluation on the user side.
 - Since the direction register setup may be changed because of a program runaway or noise, set direction registers by program periodically to increase the reliability of program.
- ④ The AVss pin when not using the A-D converter :
 - When not using the A-D converter, handle a power source pin for the A-D converter, AVss pin as follows:
 - AVss:Connect to the Vss pin

(2) Termination remarks

- ① Input ports and I/O ports :
Do not open in the input mode.
 - **Reason**
 - The power source current may increase depending on the first-stage circuit.
 - An effect due to noise may be easily produced as compared with proper termination ② and ③ shown on the above.
 - ② I/O ports :
When setting for the input mode, do not connect to VCC or VSS directly.
 - **Reason**
If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between a port and VCC (or VSS).
 - ③ I/O ports :
When setting for the input mode, do not connect multiple ports in a lump to VCC or VSS through a resistor.
 - **Reason**
If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between ports.
- At the termination of unused pins, perform wiring at the shortest possible distance (20 mm or less) from microcomputer pins.

APPLICATION

2.2 Timer

2.2 Timer

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

2.2.1 Memory map

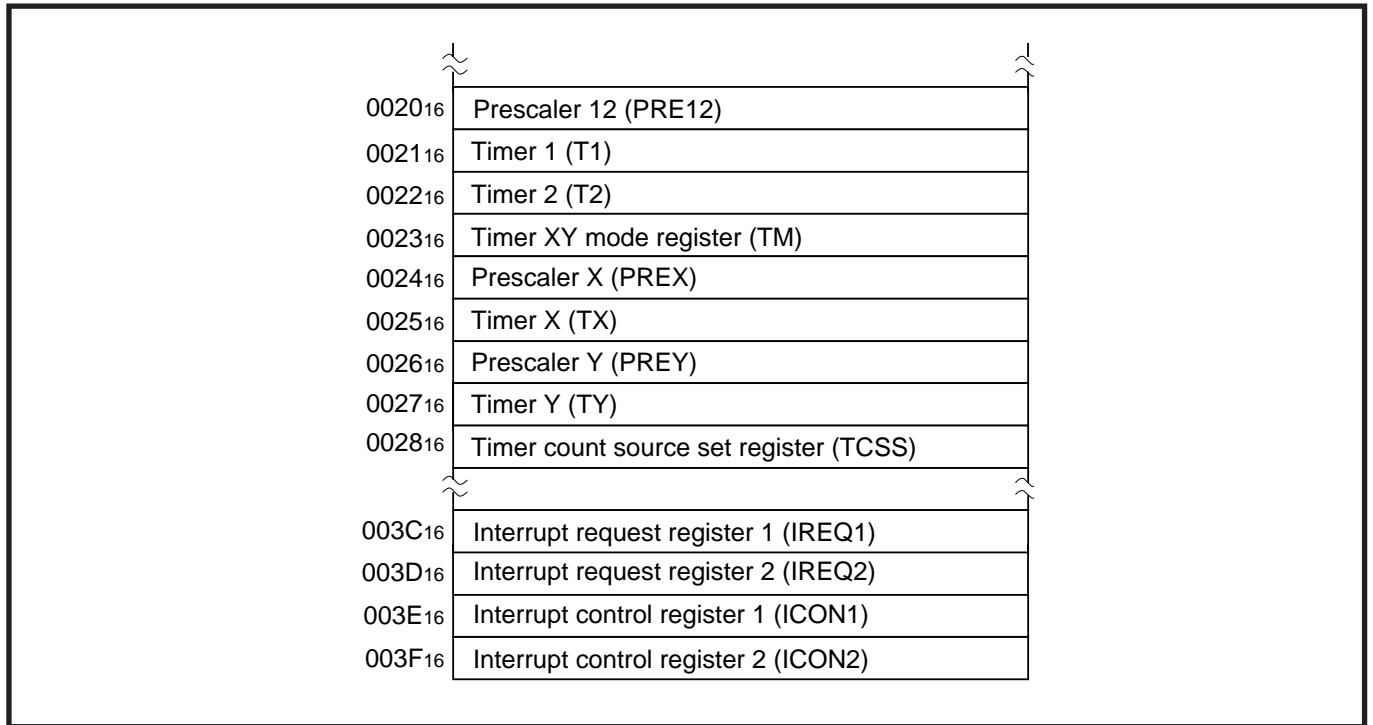


Fig. 2.2.1 Memory map of registers relevant to timers

2.2.2 Relevant registers

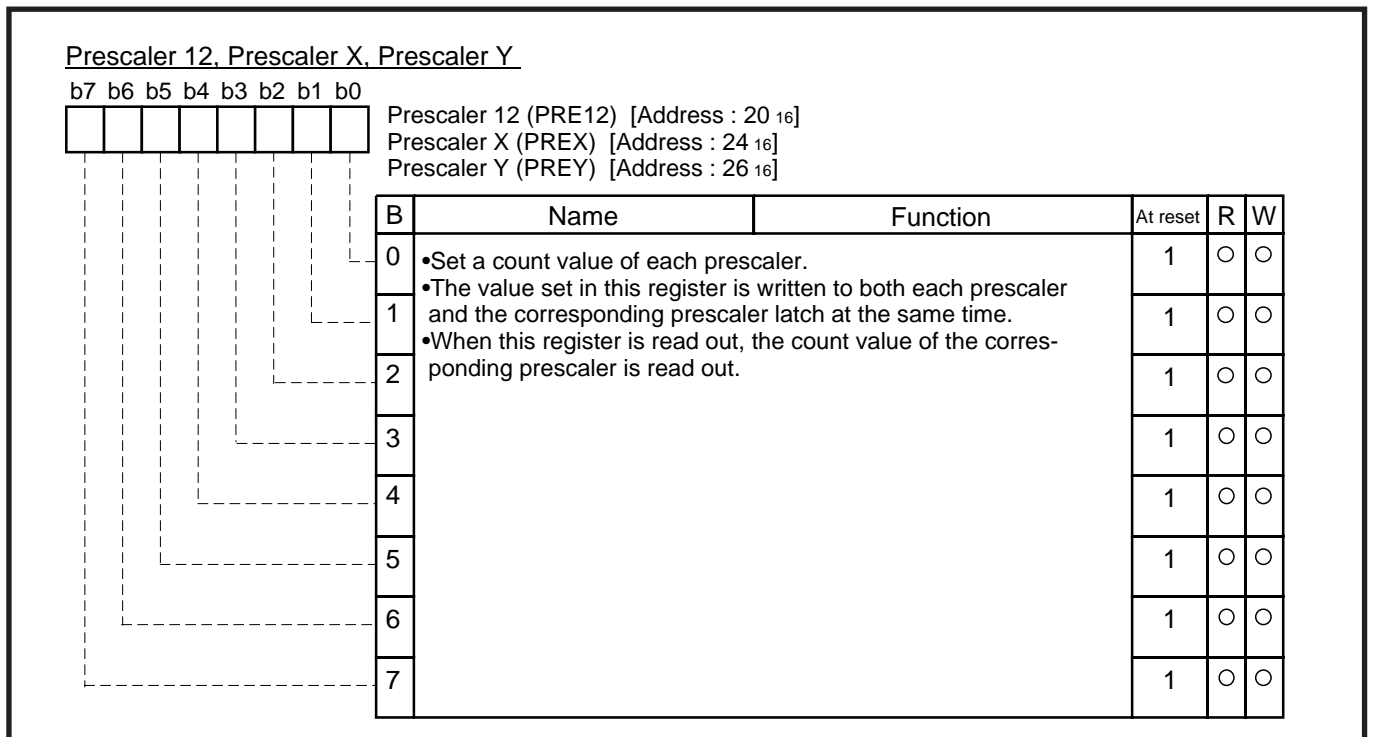


Fig. 2.2.2 Structure of Prescaler 12, Prescaler X, Prescaler Y

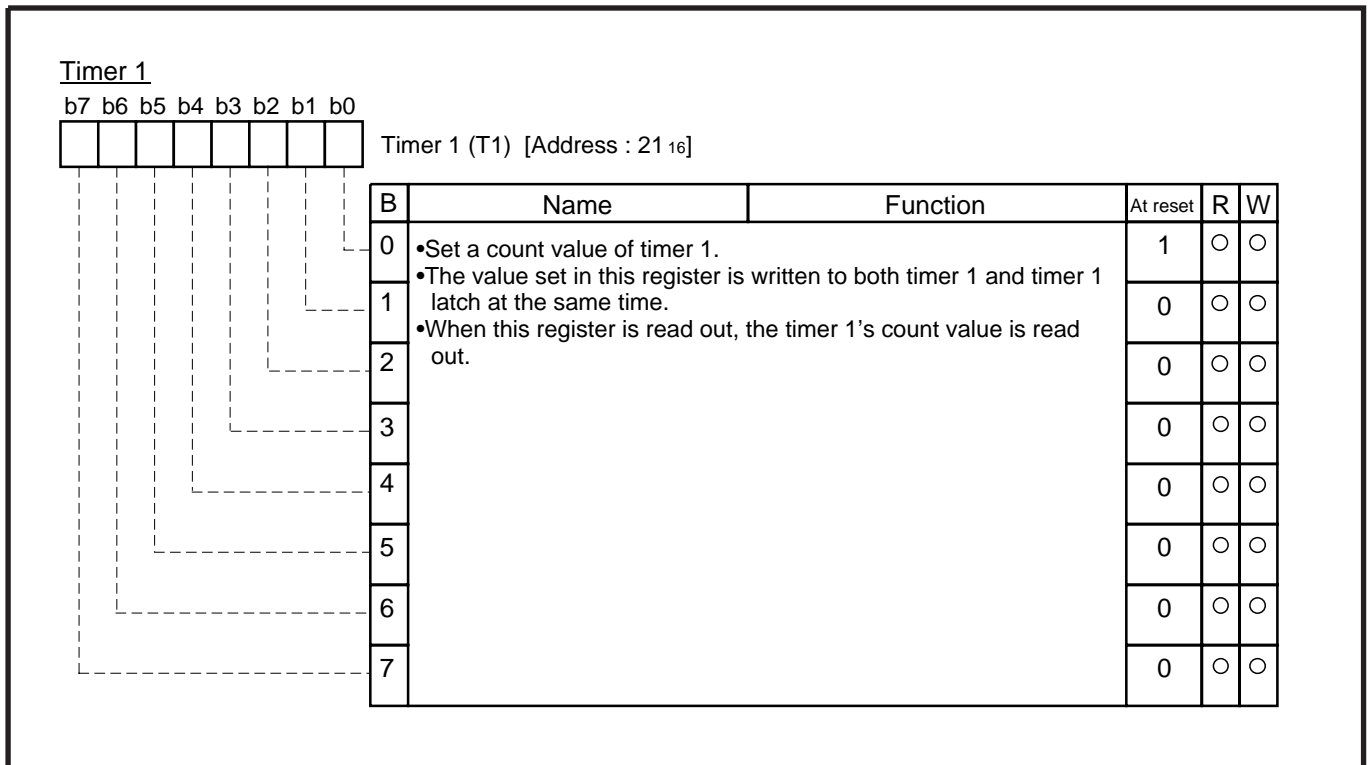


Fig. 2.2.3 Structure of Timer 1

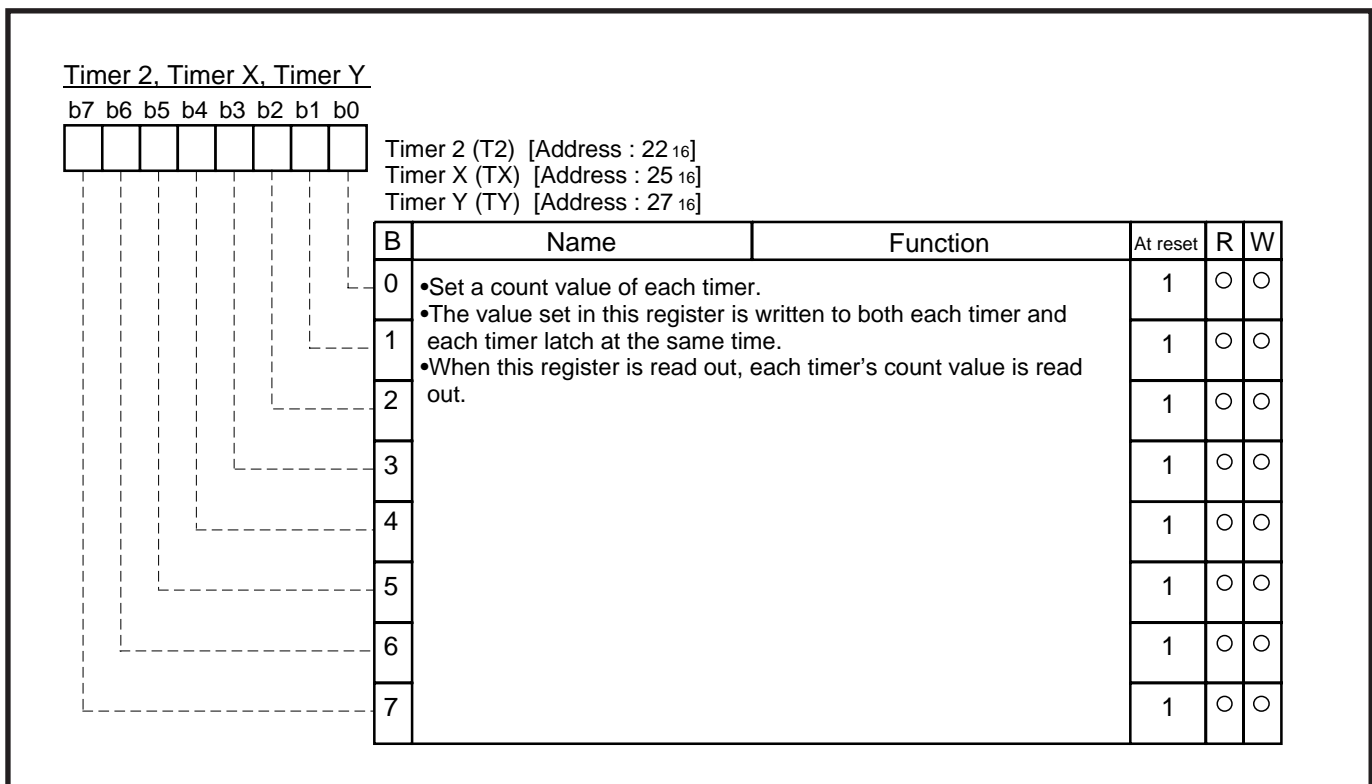


Fig. 2.2.4 Structure of Timer 2, Timer X, Timer Y

APPLICATION

2.2 Timer

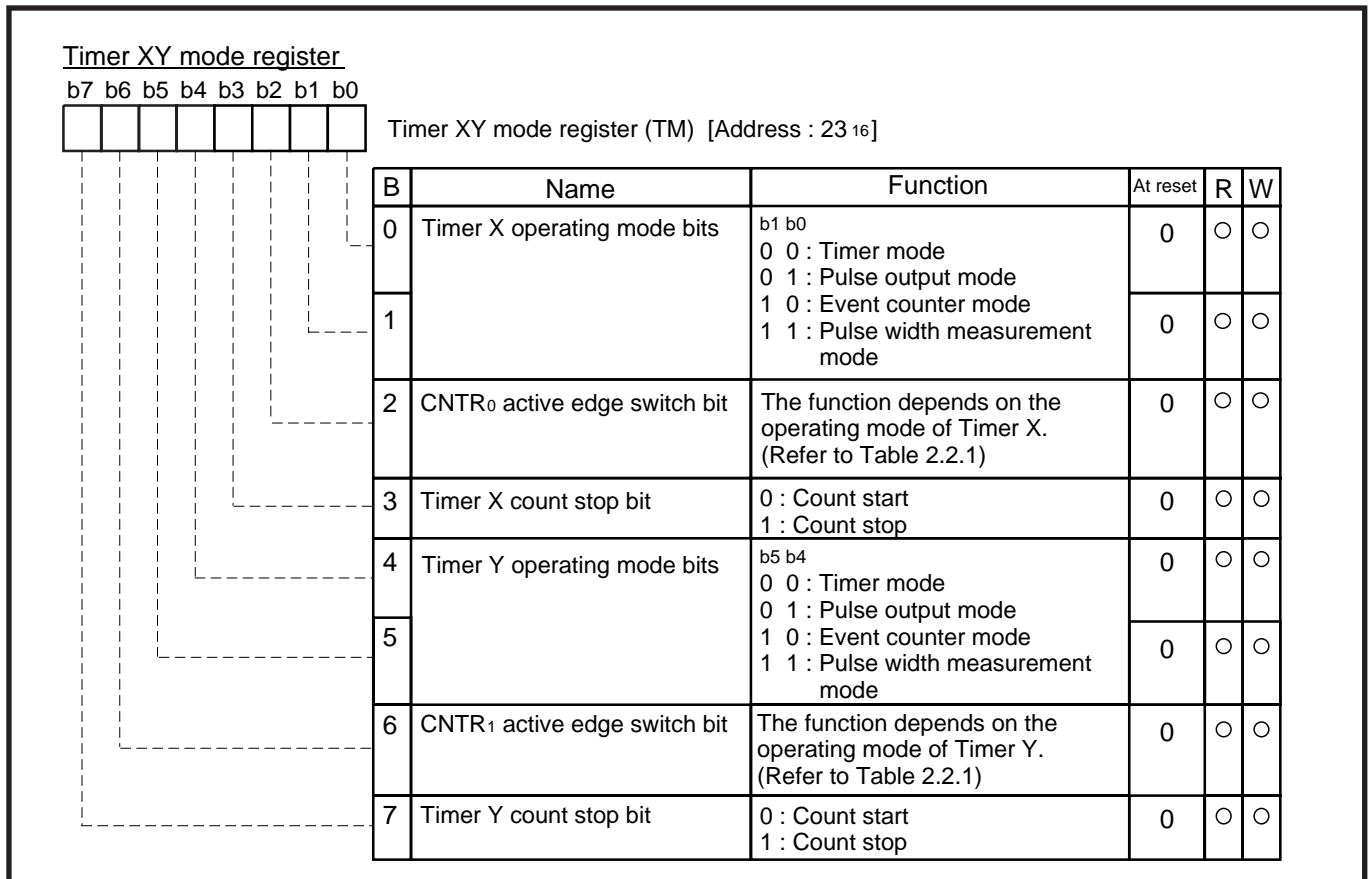


Fig. 2.2.5 Structure of Timer XY mode register

Table 2.2.1 CNTR₀ /CNTR₁ active edge switch bit function

Timer X /Timer Y operation modes	CNTR ₀ / CNTR ₁ active edge switch bit (bits 2, 6 of address 23 ₁₆) contents	
Timer mode	"0"	CNTR ₀ / CNTR ₁ interrupt request occurrence: Falling edge ; No influence to timer count
	"1"	CNTR ₀ / CNTR ₁ interrupt request occurrence: Rising edge ; No influence to timer count
Pulse output mode	"0"	Pulse output start: Beginning at "H" level CNTR ₀ / CNTR ₁ interrupt request occurrence: Falling edge
	"1"	Pulse output start: Beginning at "L" level CNTR ₀ / CNTR ₁ interrupt request occurrence: Rising edge
Event counter mode	"0"	Timer X / Timer Y: Rising edge count CNTR ₀ / CNTR ₁ interrupt request occurrence: Falling edge
	"1"	Timer X / Timer Y: Falling edge count CNTR ₀ / CNTR ₁ interrupt request occurrence: Rising edge
Pulse width measurement mode	"0"	Timer X / Timer Y: "H" level width measurement CNTR ₀ / CNTR ₁ interrupt request occurrence: Falling edge
	"1"	Timer X / Timer Y: "L" level width measurement CNTR ₀ / CNTR ₁ interrupt request occurrence: Rising edge

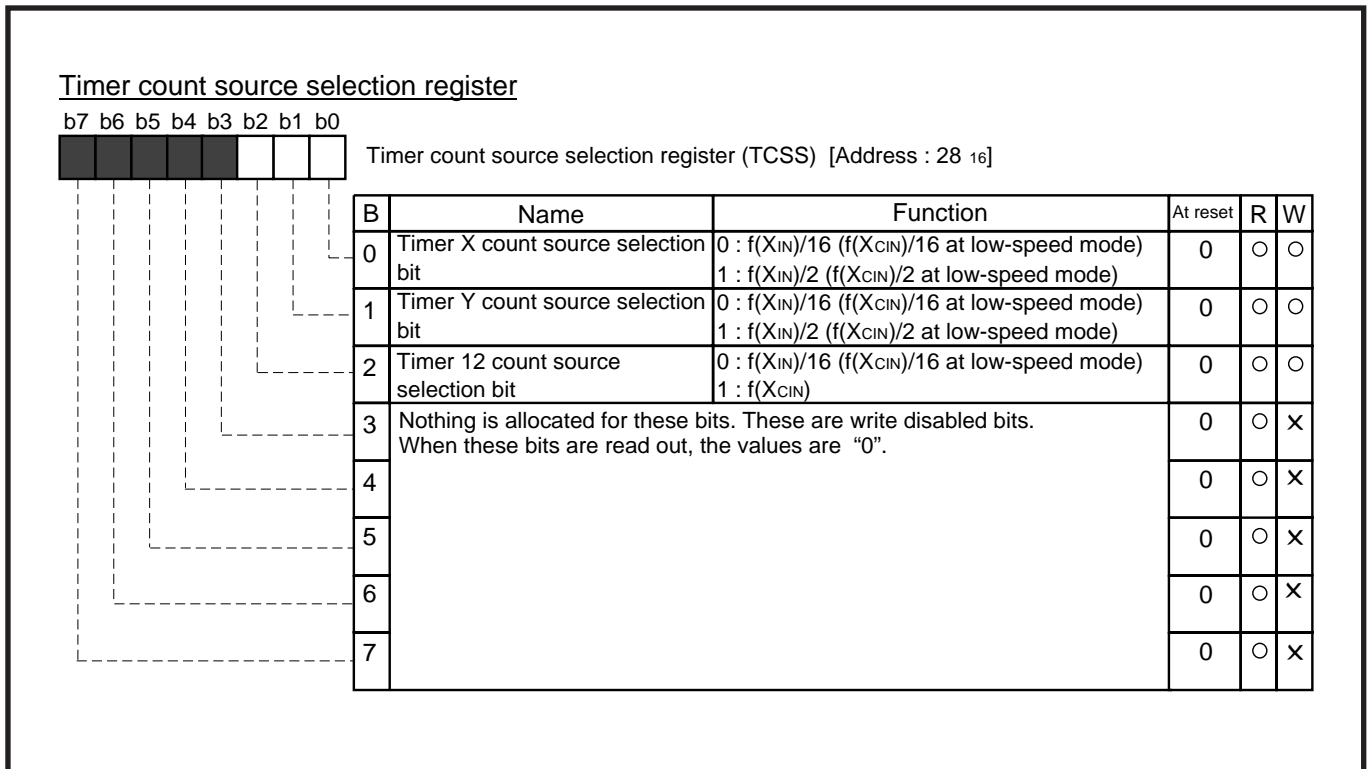


Fig. 2.2.6 Structure of Timer count source set register

APPLICATION

2.2 Timer

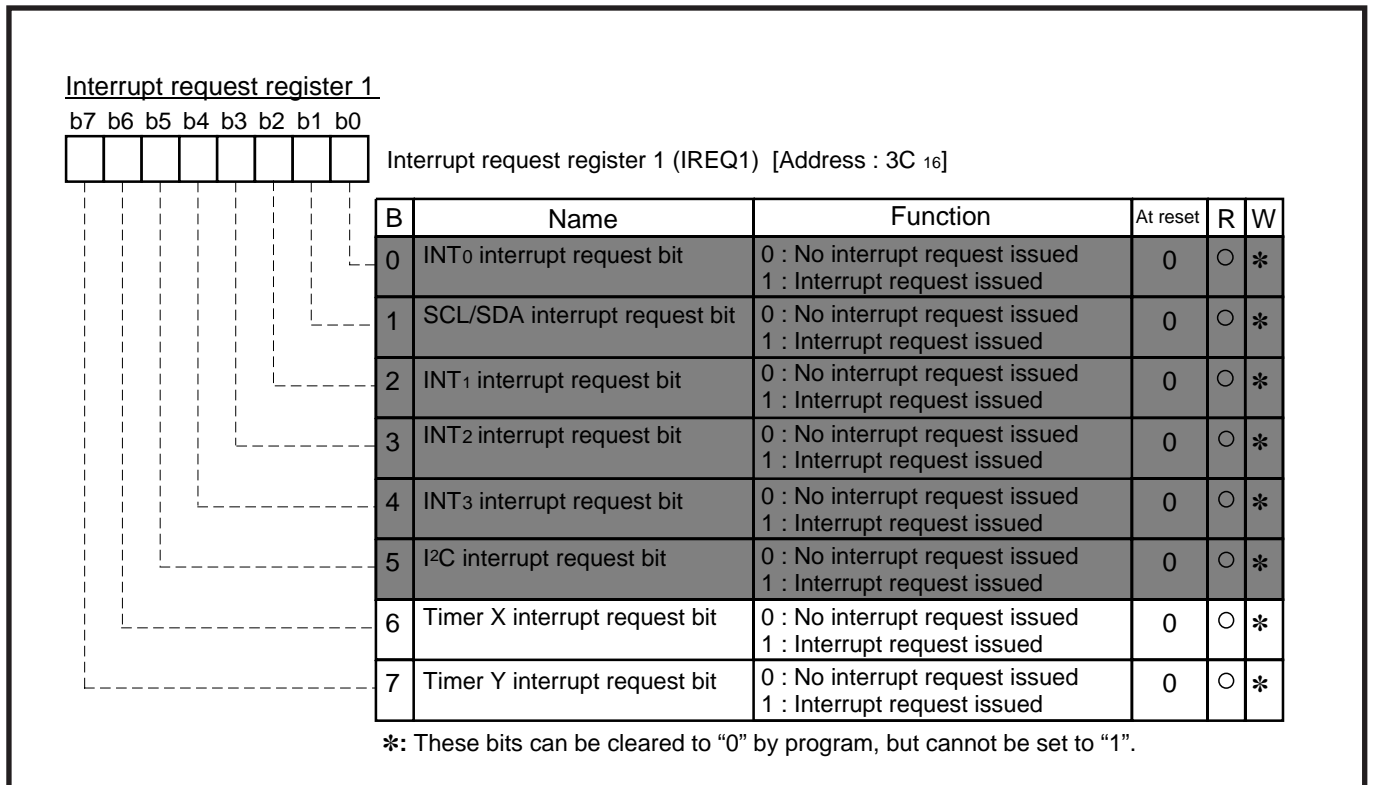


Fig. 2.2.7 Structure of Interrupt request register 1

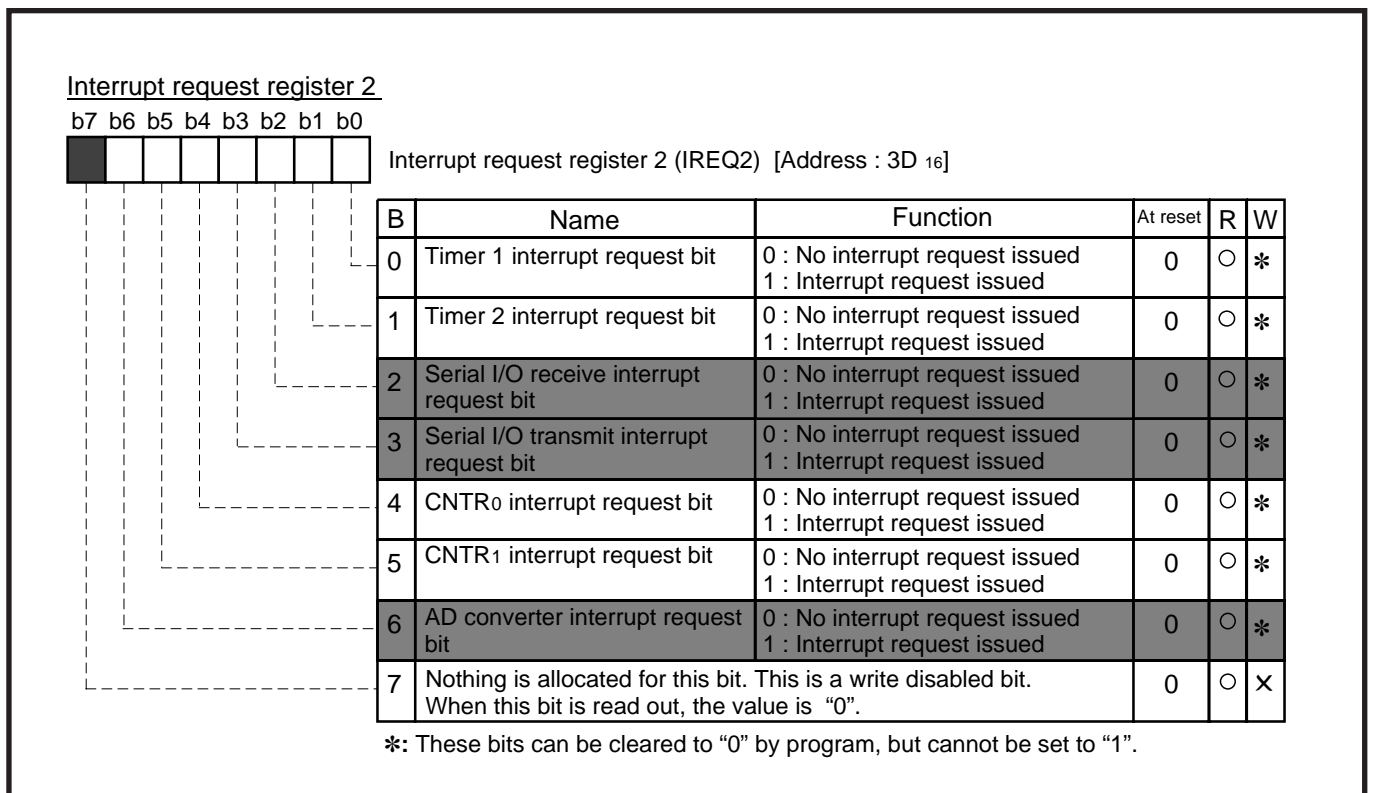


Fig. 2.2.8 Structure of Interrupt request register 2

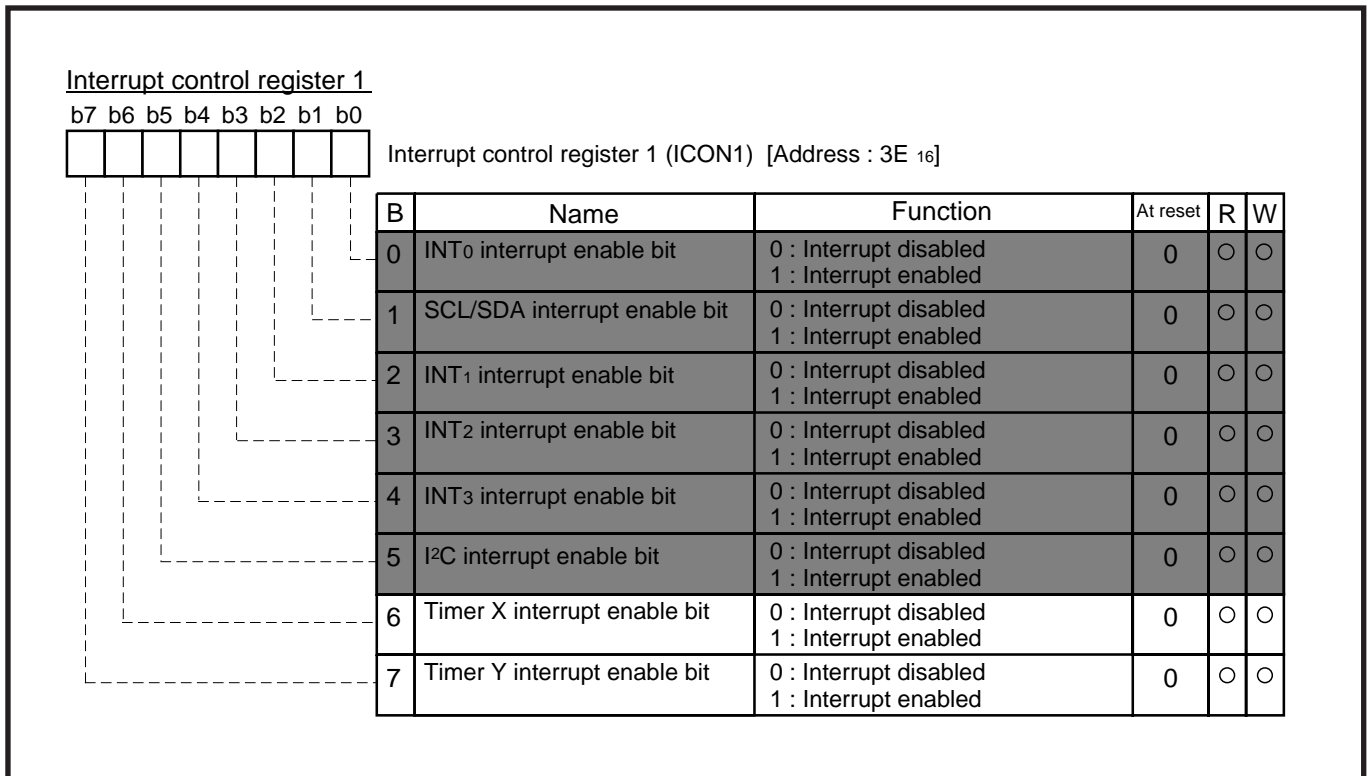


Fig. 2.2.9 Structure of Interrupt control register 1

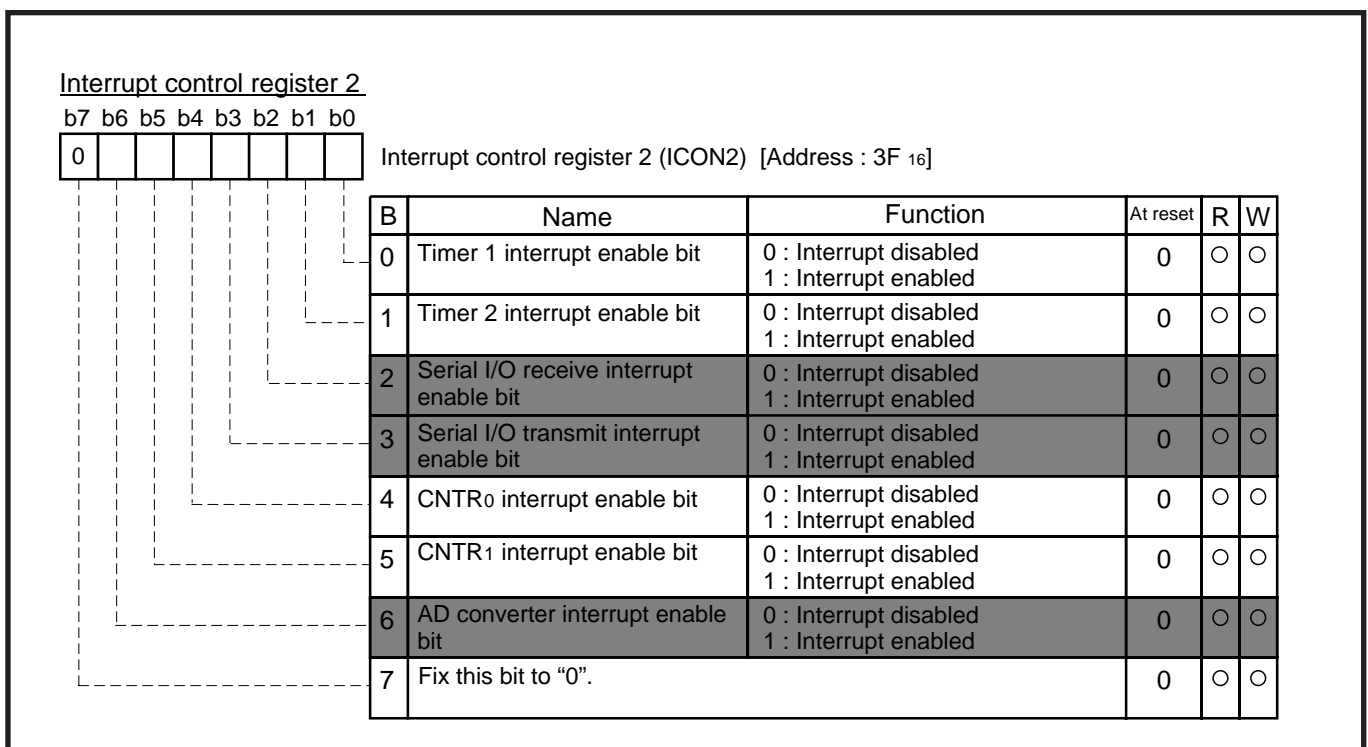


Fig. 2.2.10 Structure of Interrupt control register 2

APPLICATION

2.2 Timer

2.2.3 Timer application examples

(1) Basic functions and uses

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

When a certain time, by setting a count value to each timer, has passed, the timer interrupt request occurs.

<Use>

- Generation of an output signal timing
- Generation of a wait time

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

The value of the timer latch is automatically written to the corresponding timer each time the timer underflows, and each timer interrupt request occurs in cycles.

<Use>

- Generation of cyclic interrupts
- Clock function (measurement of 250 ms); see Application example 1
- Control of a main routine cycle

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

The output level of the CNTR₀ pin or CNTR₁ pin is inverted each time the timer underflows (in the pulse output mode).

<Use>

- Piezoelectric buzzer output; see Application example 2
- Generation of the remote control carrier waveforms

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

External pulses input to the CNTR₀ pin or CNTR₁ pin are counted as the timer count source (in the event counter mode).

<Use>

- Frequency measurement; see Application example 3
- Division of external pulses
- Generation of interrupts due to a cycle using external pulses as the count source; count of a reel pulse

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

The "H" or "L" level width of external pulses input to CNTR₀ pin or CNTR₁ pin is measured (in the pulse width measurement mode).

<Use>

- Measurement of external pulse frequency (measurement of pulse width of FG pulse* for a motor); see Application example 4
- Measurement of external pulse duty (when the frequency is fixed)

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

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

Outline: The input clock is divided by the timer so that the clock can count up at 250 ms intervals.

Specifications:

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

Figure 2.2.11 shows the timers connection and setting of division ratios; Figure 2.2.12 shows the relevant registers setting; Figure 2.2.13 shows the control procedure.

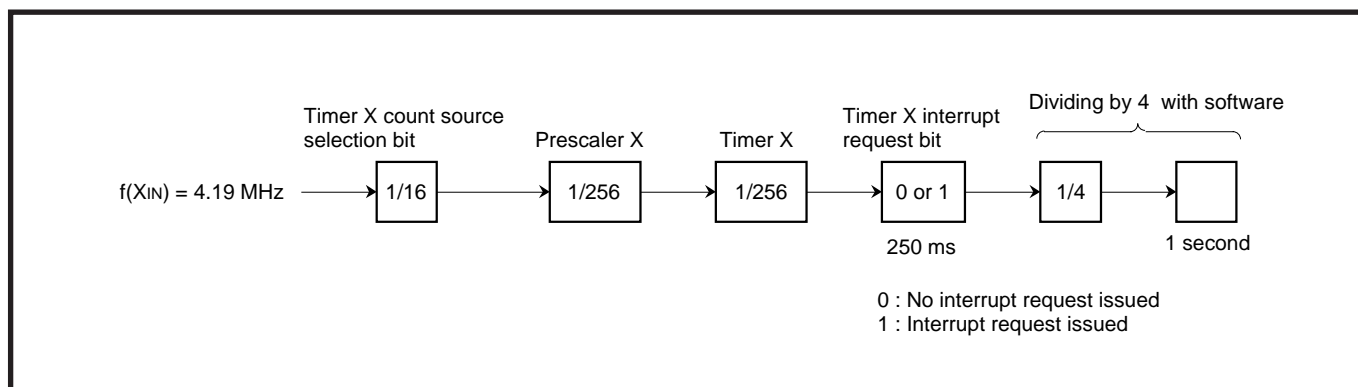


Fig. 2.2.11 Timers connection and setting of division ratios

APPLICATION

2.2 Timer

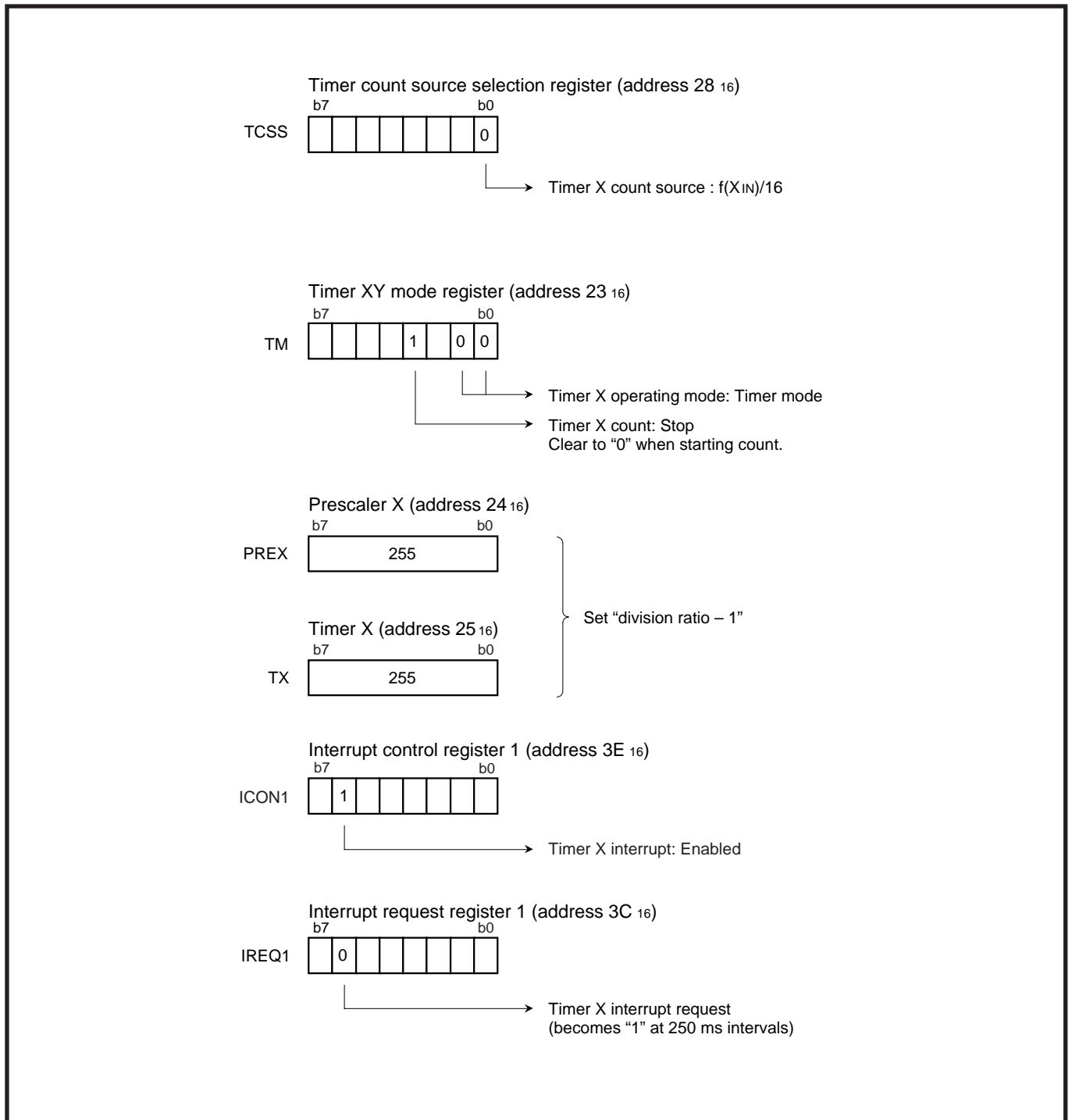


Fig. 2.2.12 Relevant registers setting

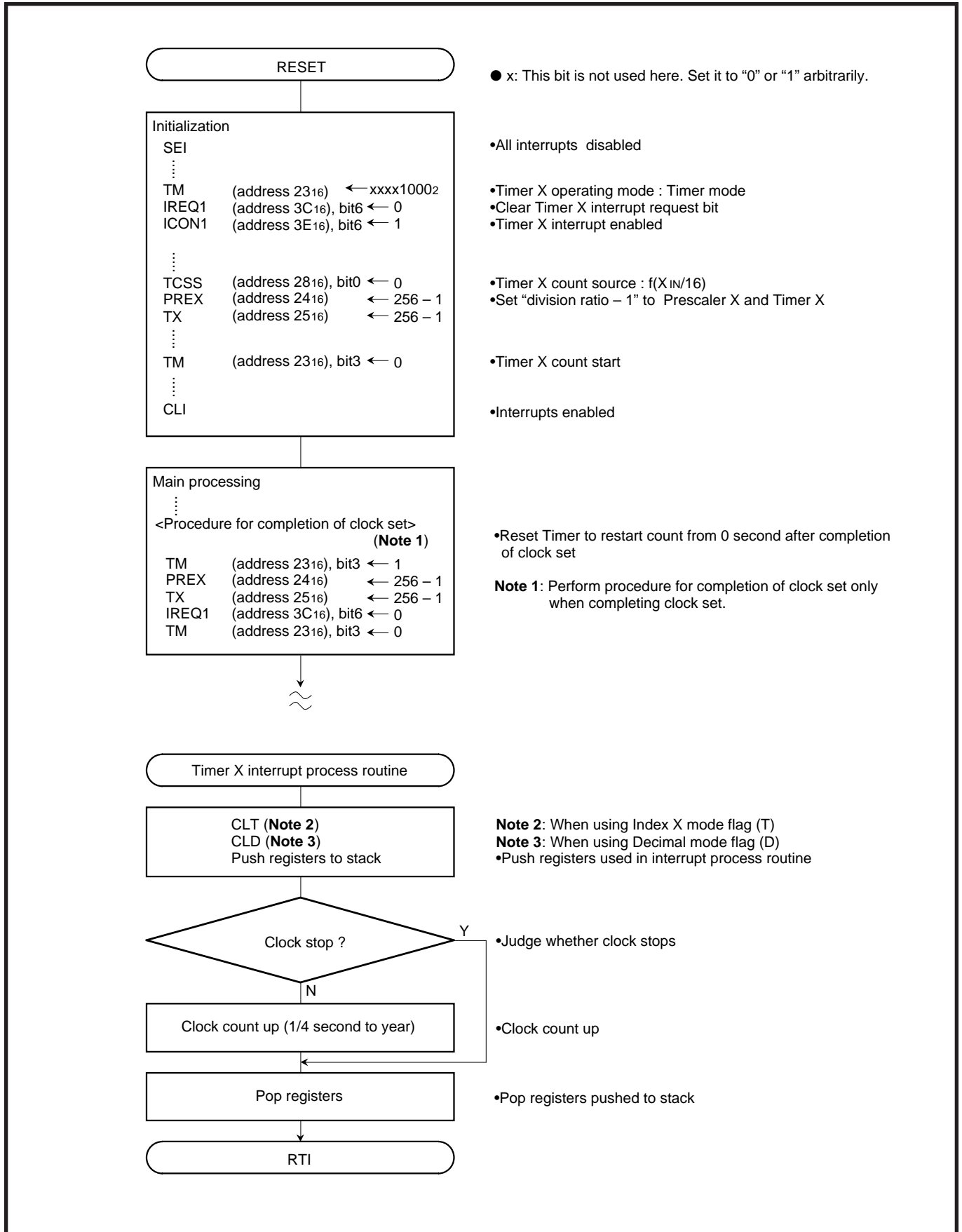


Fig. 2.2.13 Control procedure

APPLICATION

2.2 Timer

(3) Timer application example 2: Piezoelectric buzzer output

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

- Specifications:**
- The rectangular waveform, dividing the clock $f(X_{IN}) = 4.19 \text{ MHz}$ (2^{22} Hz) into about 2 kHz (2048 Hz), is output from the P27/CNTR₀ pin.
 - The level of the P27/CNTR₀ pin is fixed to “H” while a piezoelectric buzzer output stops.

Figure 2.2.14 shows a peripheral circuit example, and Figure 2.2.15 shows the timers connection and setting of division ratios. Figures 2.2.16 shows the relevant registers setting, and Figure 2.2.17 shows the control procedure.

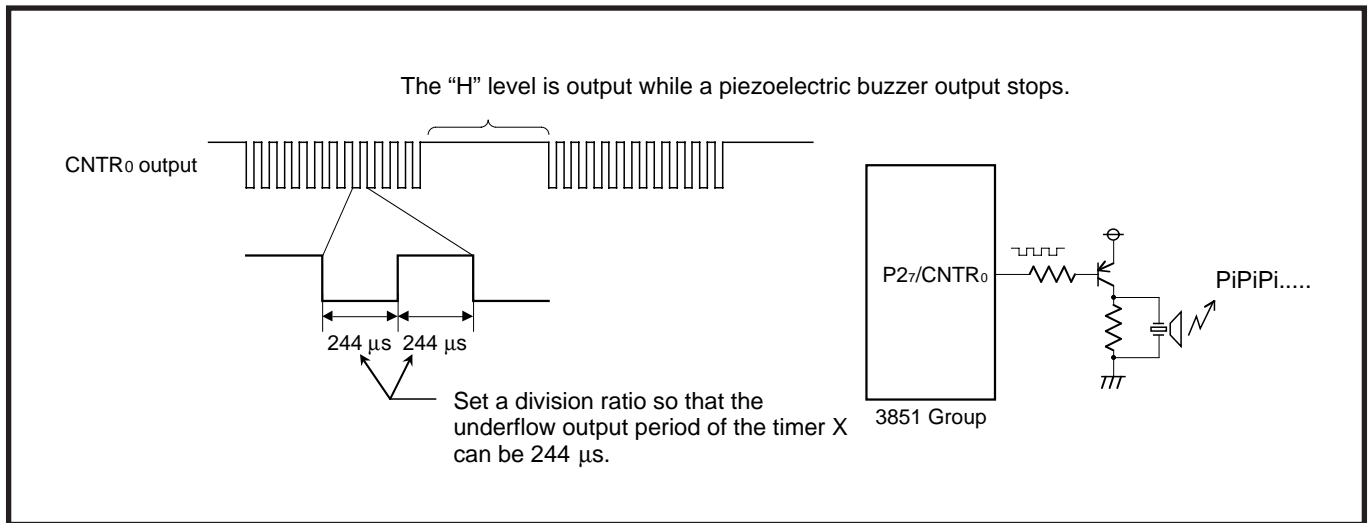


Fig. 2.2.14 Peripheral circuit example

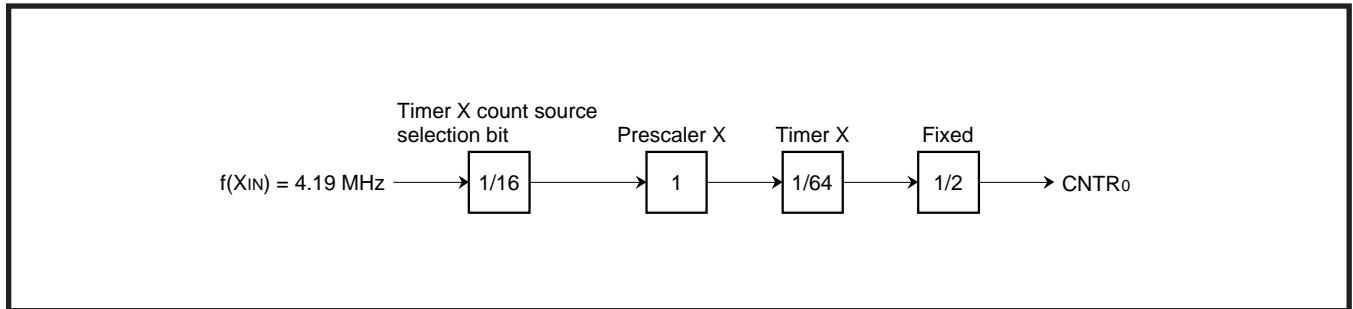


Fig. 2.2.15 Timers connection and setting of division ratios

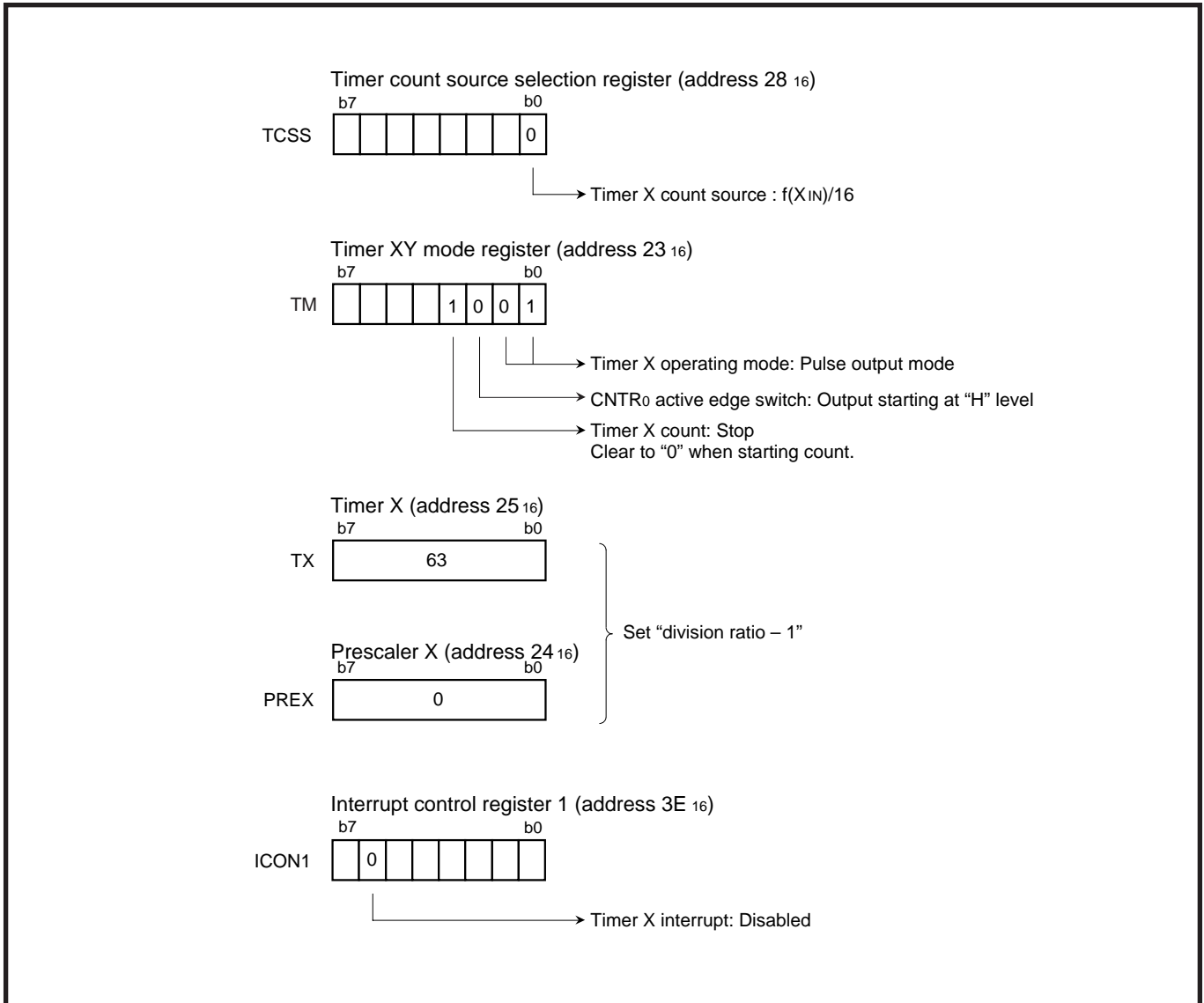


Fig. 2.2.16 Relevant registers setting

APPLICATION

2.2 Timer

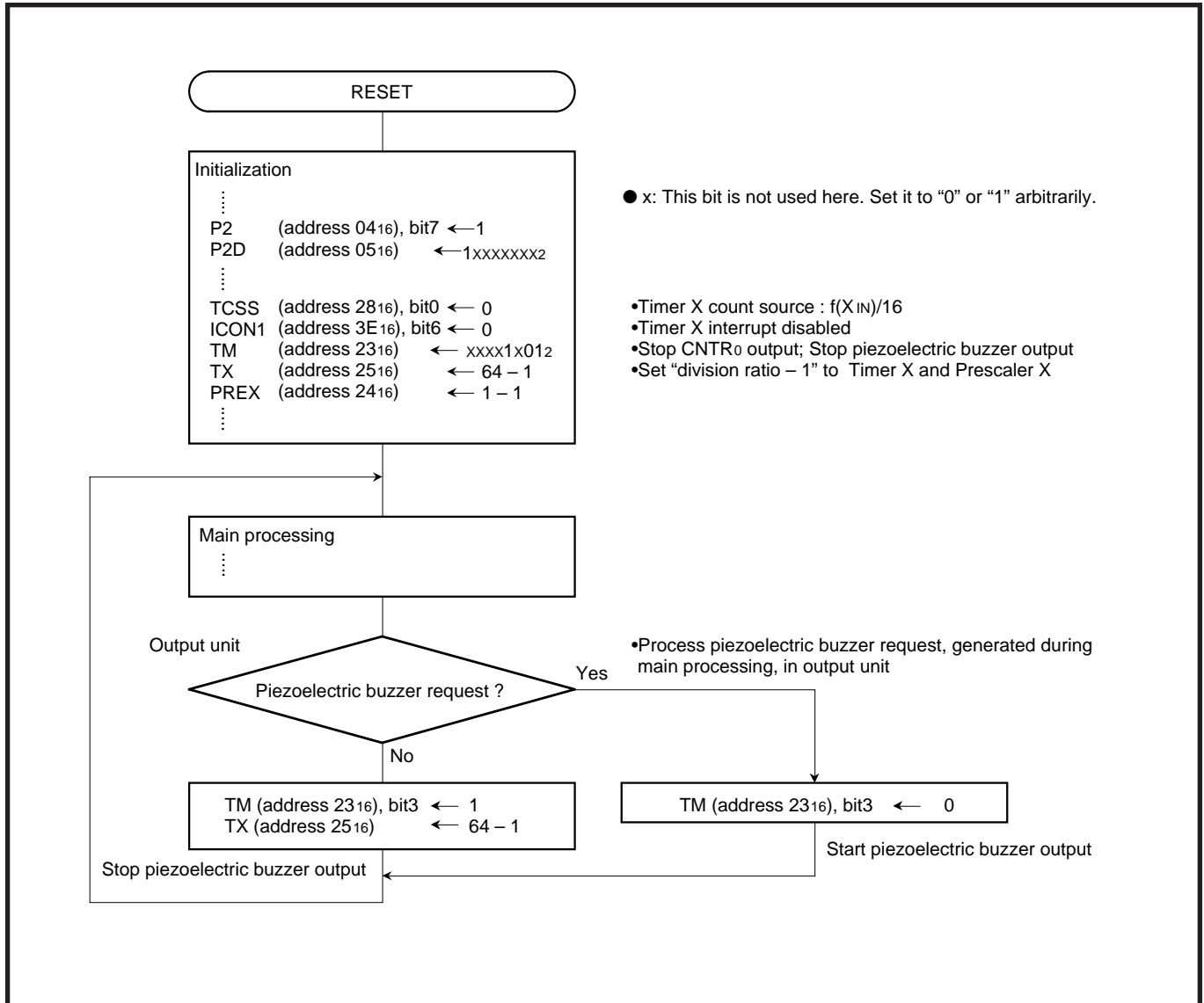


Fig. 2.2.17 Control procedure

(4) Timer application example 3: Frequency measurement

Outline: The following two values are compared to judge whether the frequency is within a valid range.

- A value by counting pulses input to P4₀/CNTR₁ pin with the timer.
- A reference value

Specifications:

- The pulse is input to the P4₀/CNTR₁ pin and counted by the timer Y.
- A count value is read out at about 2 ms intervals, the timer 1 interrupt interval. When the count value is 28 to 40, it is judged that the input pulse is valid.
- Because the timer is a down-counter, the count value is compared with 227 to 215 (Note).

Note: 227 to 215 = {255 (initial value of counter) – 28} to {255 – 40}; 28 to 40 means the number of valid value.

Figure 2.2.18 shows the judgment method of valid/invalid of input pulses; Figure 2.2.19 shows the relevant registers setting; Figure 2.2.20 shows the control procedure.

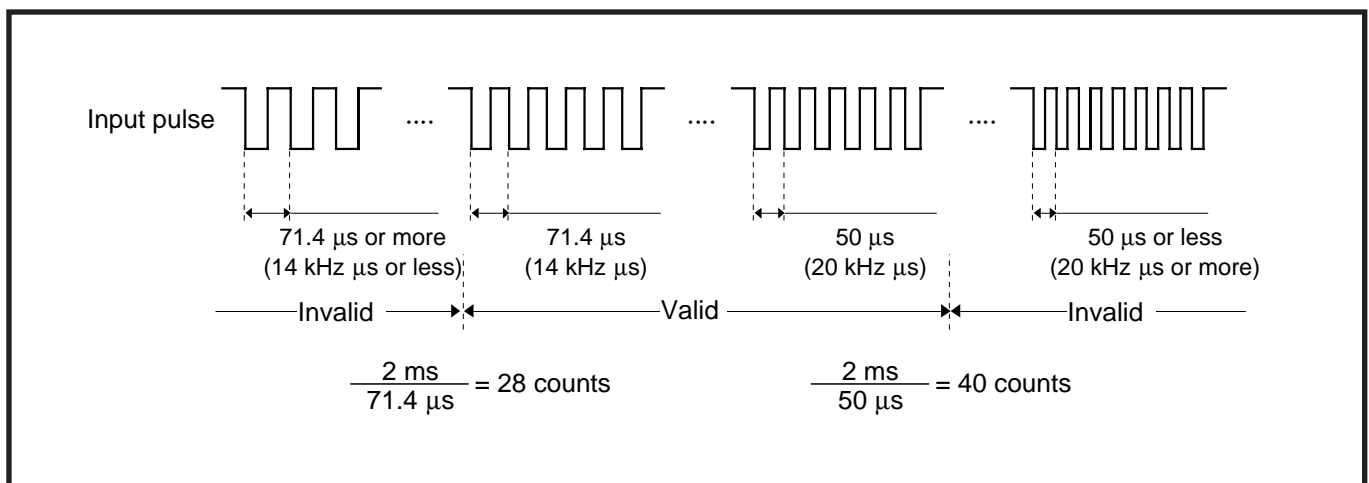


Fig 2.2.18 Judgment method of valid/invalid of input pulses

APPLICATION

2.2 Timer

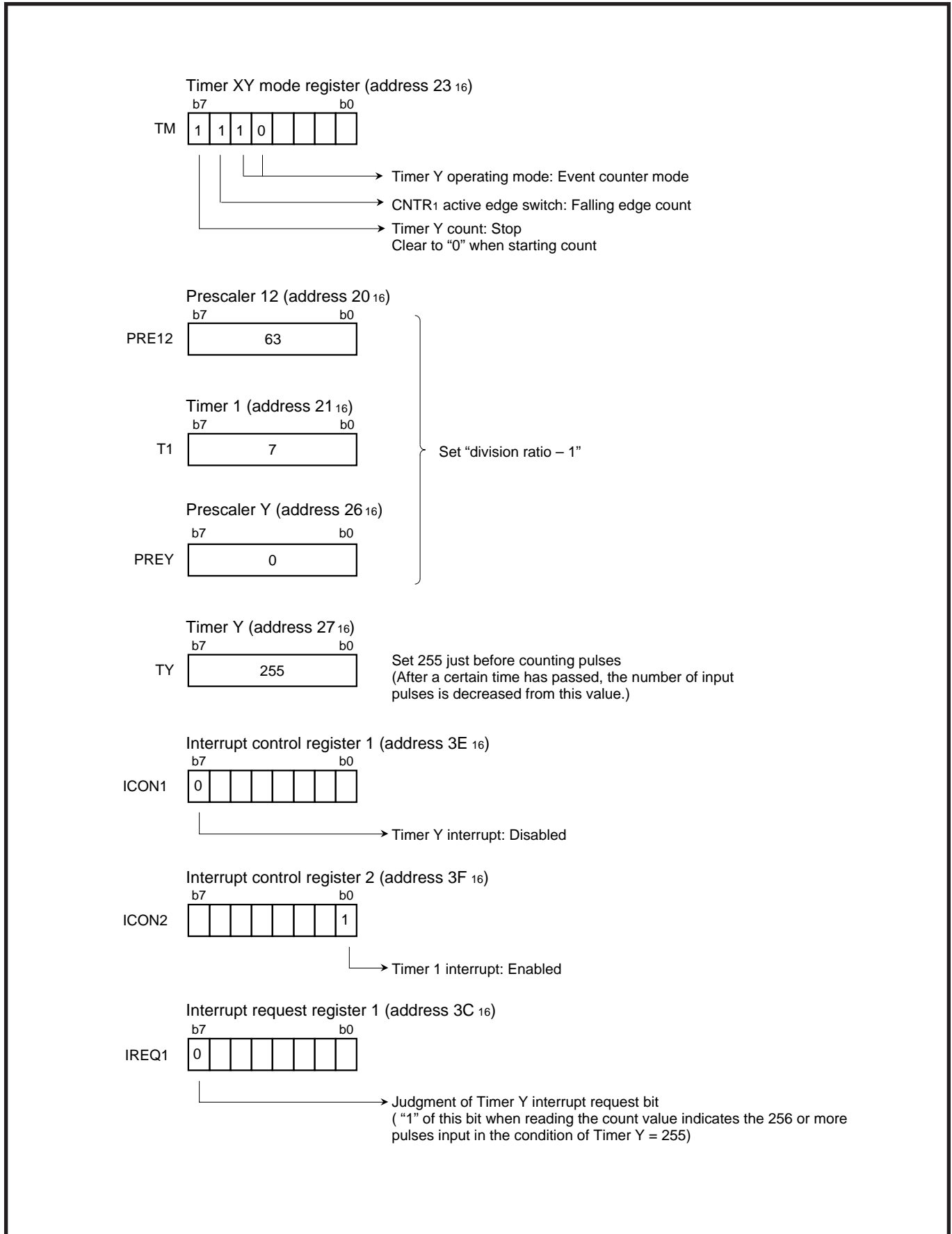


Fig. 2.2.19 Relevant registers setting

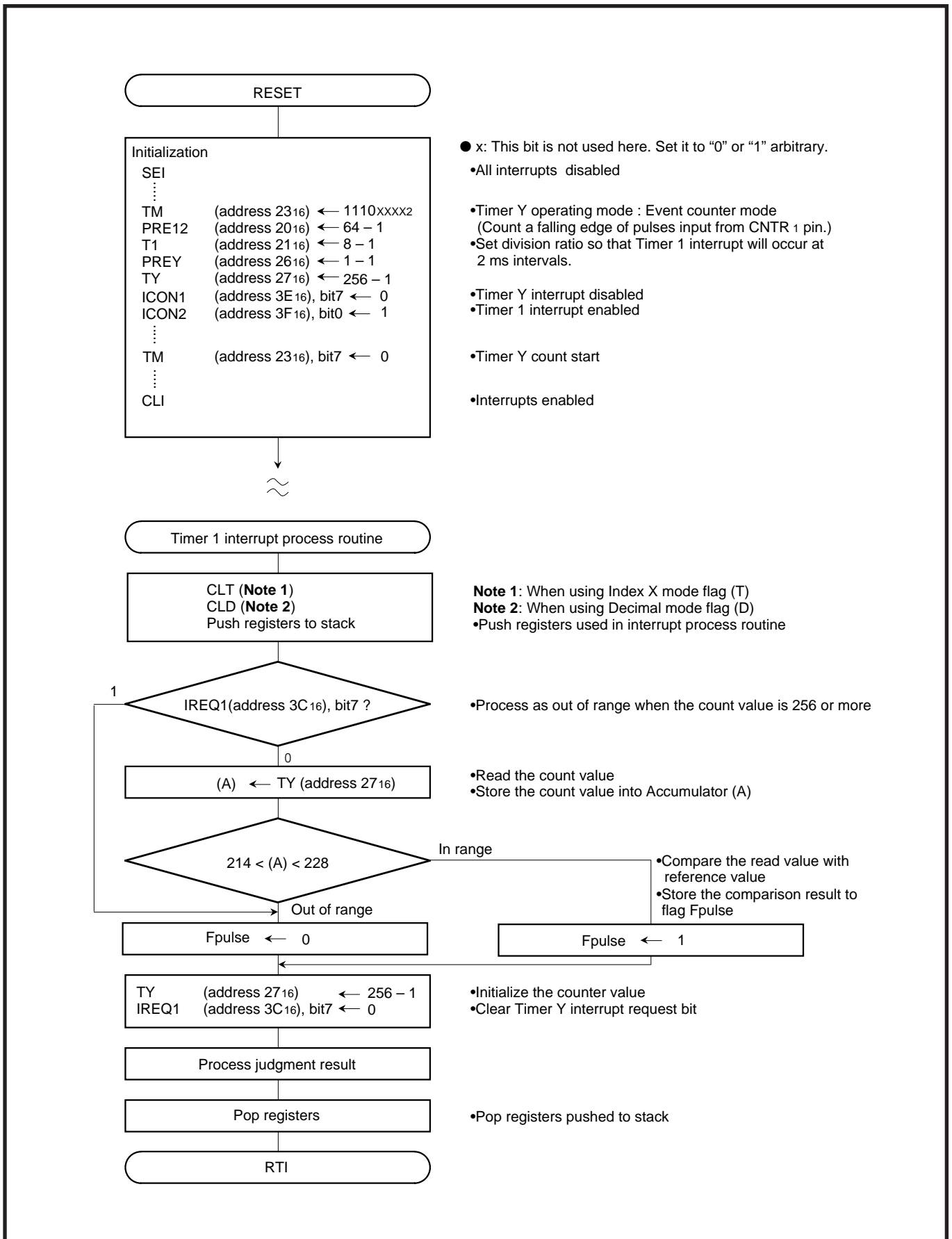


Fig. 2.2.20 Control procedure

APPLICATION

2.2 Timer

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

Outline: The timer X counts the “H” level width of the pulses input to the P27/CNTR₀ pin. An underflow is detected by the timer X interrupt and an end of the input pulse “H” level is detected by the CNTR₀ interrupt.

Specifications: •The timer X counts the “H” level width of the FG pulse input to the P27/CNTR₀ pin.

<Example>

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

Figure 2.2.21 shows the timers connection and setting of division ratio; Figure 2.2.22 shows the relevant registers setting; Figure 2.2.23 shows the control procedure.

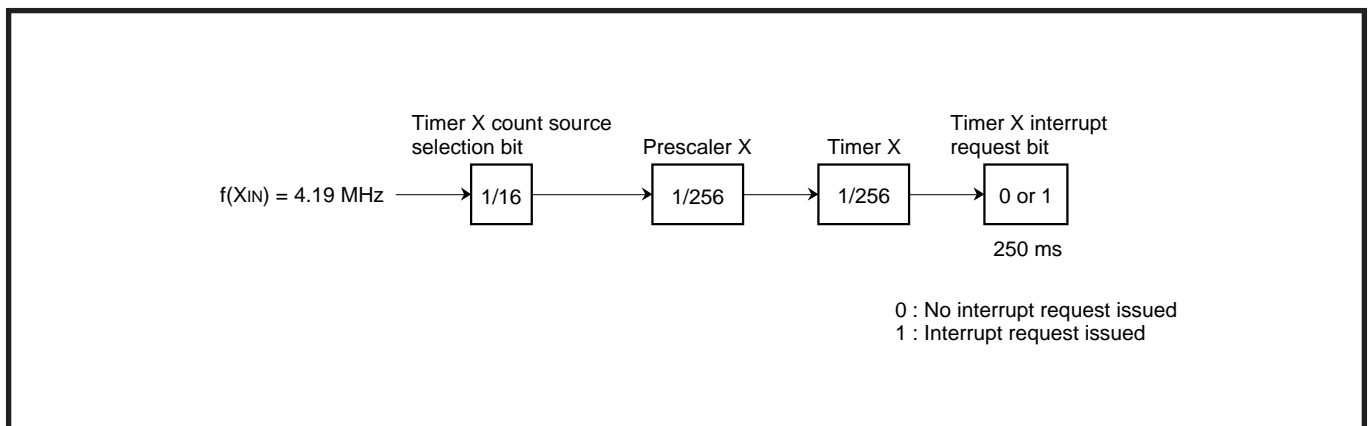


Fig. 2.2.21 Timers connection and setting of division ratios

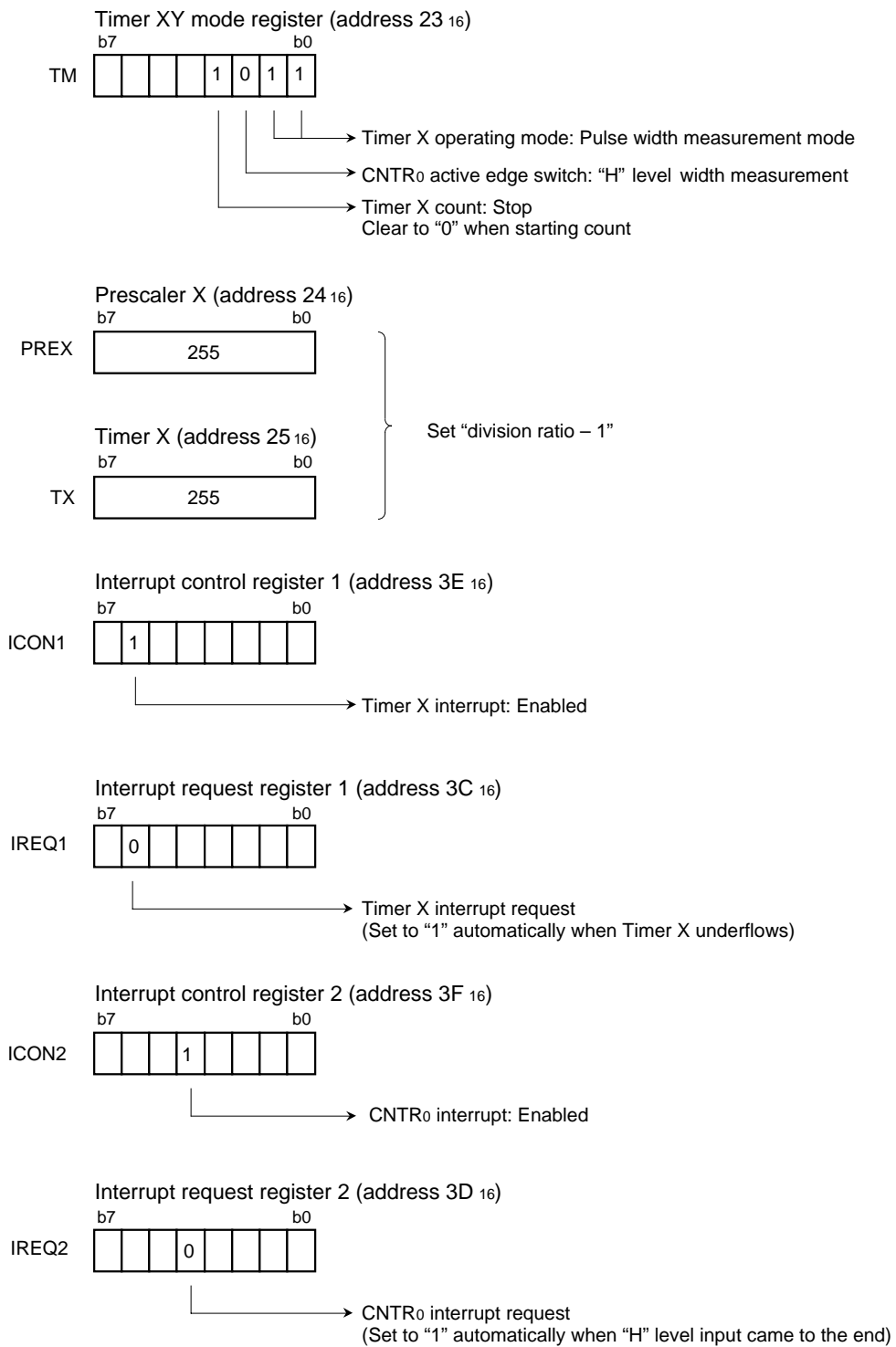


Fig. 2.2.22 Relevant registers setting

APPLICATION

2.2 Timer

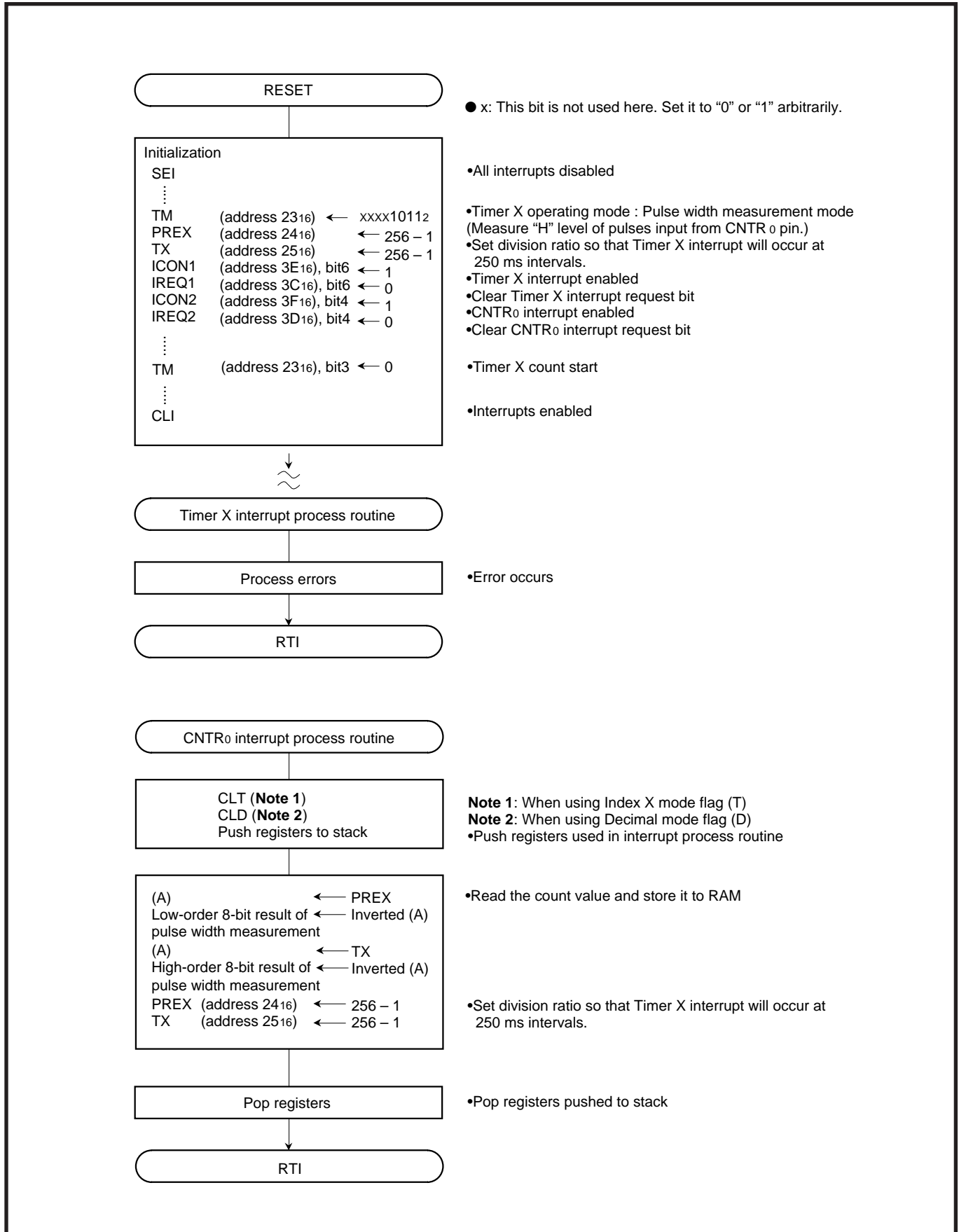


Fig. 2.2.23 Control procedure

2.2.4 Notes on the timer

- If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is $1/(n+1)$.
- When switching the count source by the timer 12, X and Y count source selection bit, the value of timer count is altered in unconsiderable amount owing to generating of a thin pulses in the count input signals.

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

APPLICATION

2.3 Serial I/O

2.3 Serial I/O

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

2.3.1 Memory map

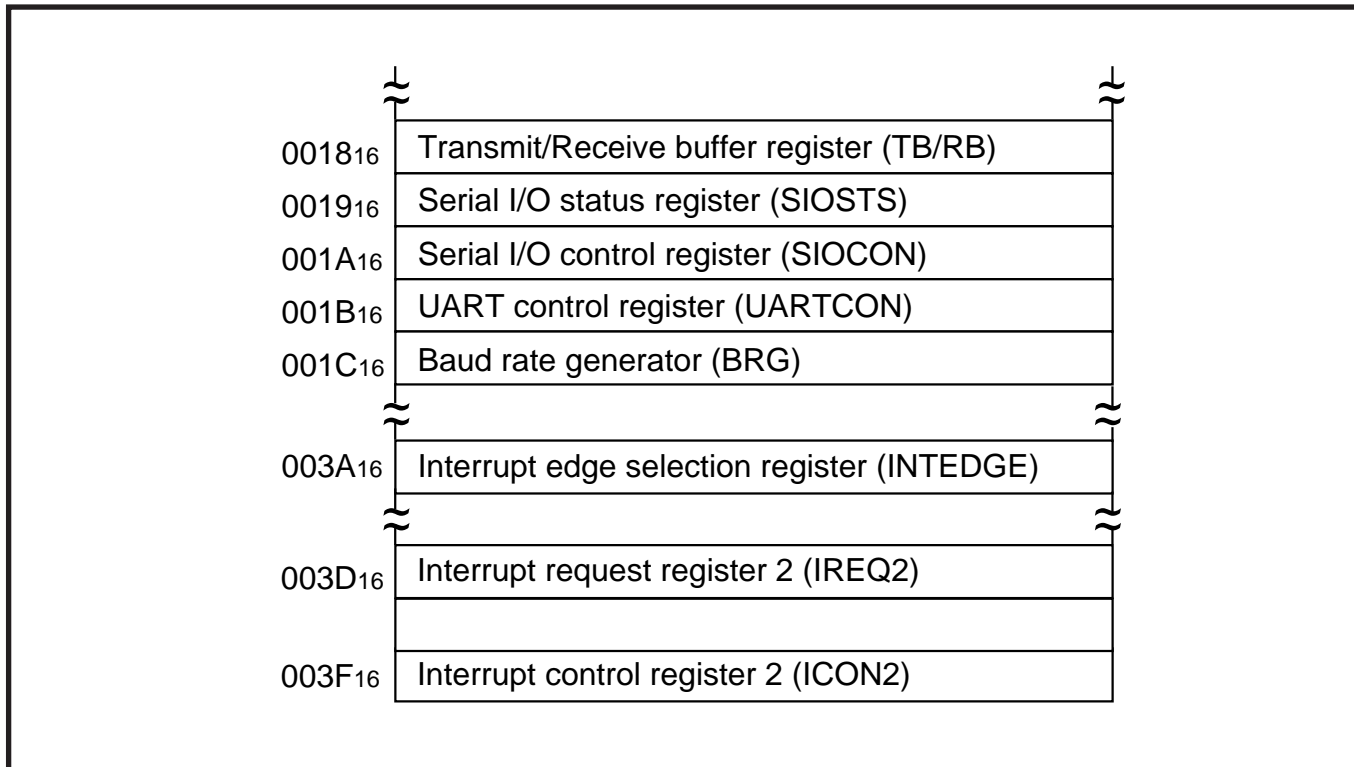


Fig. 2.3.1 Memory map of registers relevant to Serial I/O

2.3.2 Relevant registers

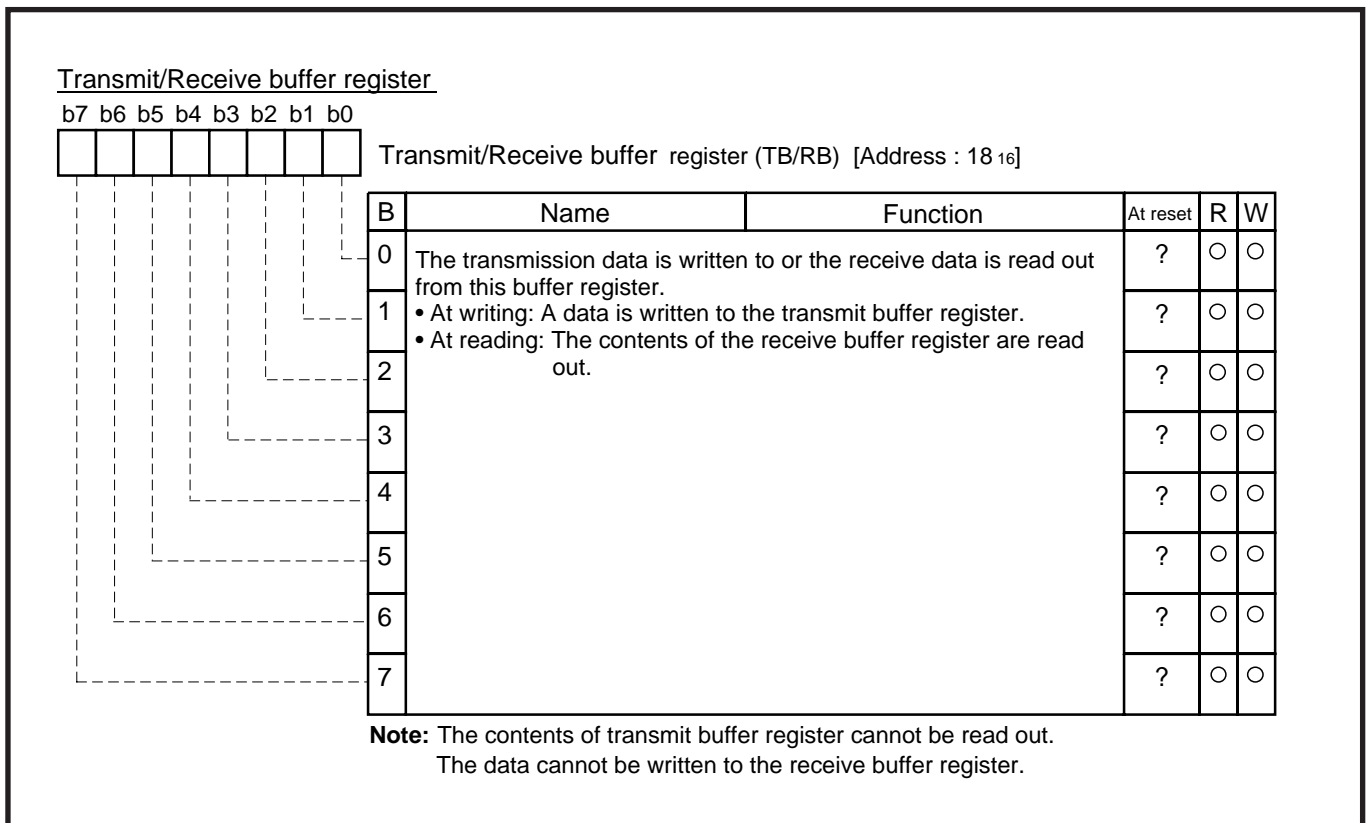


Fig. 2.3.2 Structure of Transmit/Receive buffer register

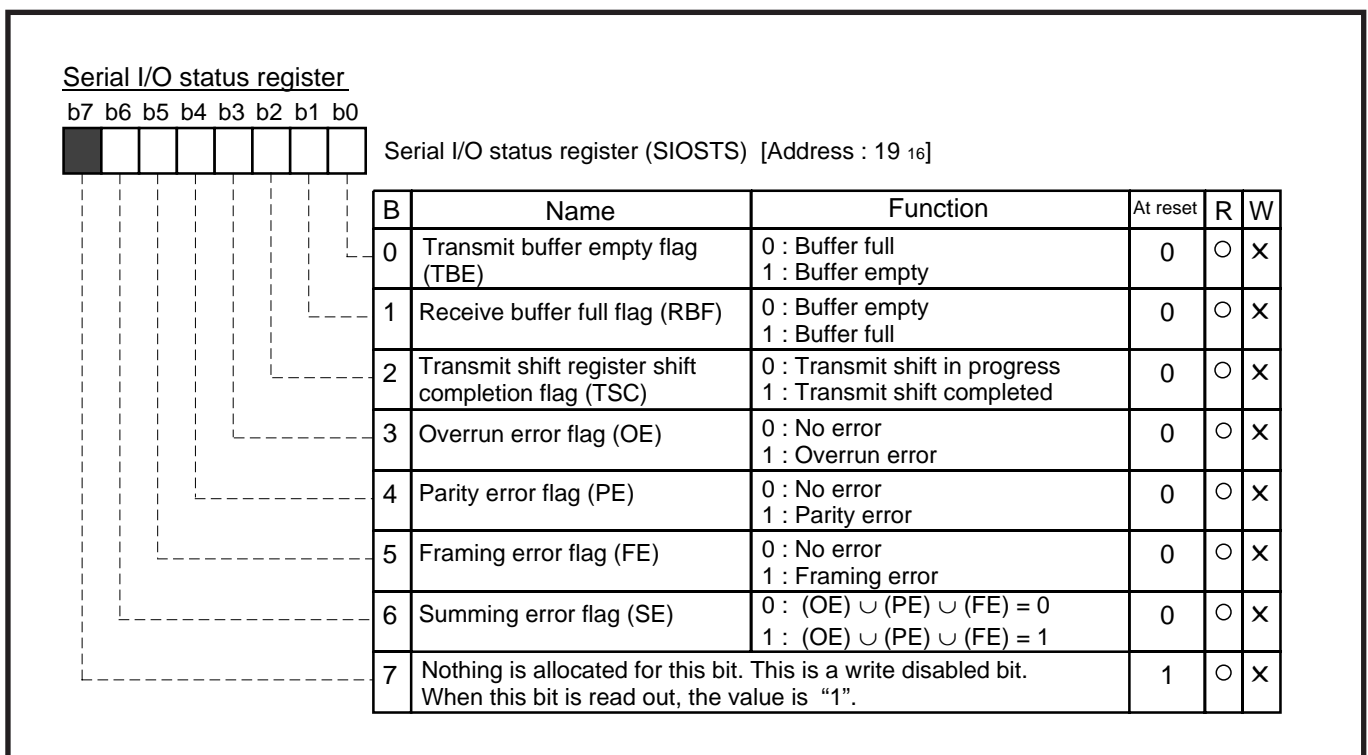


Fig. 2.3.3 Structure of Serial I/O status register

APPLICATION

2.3 Serial I/O

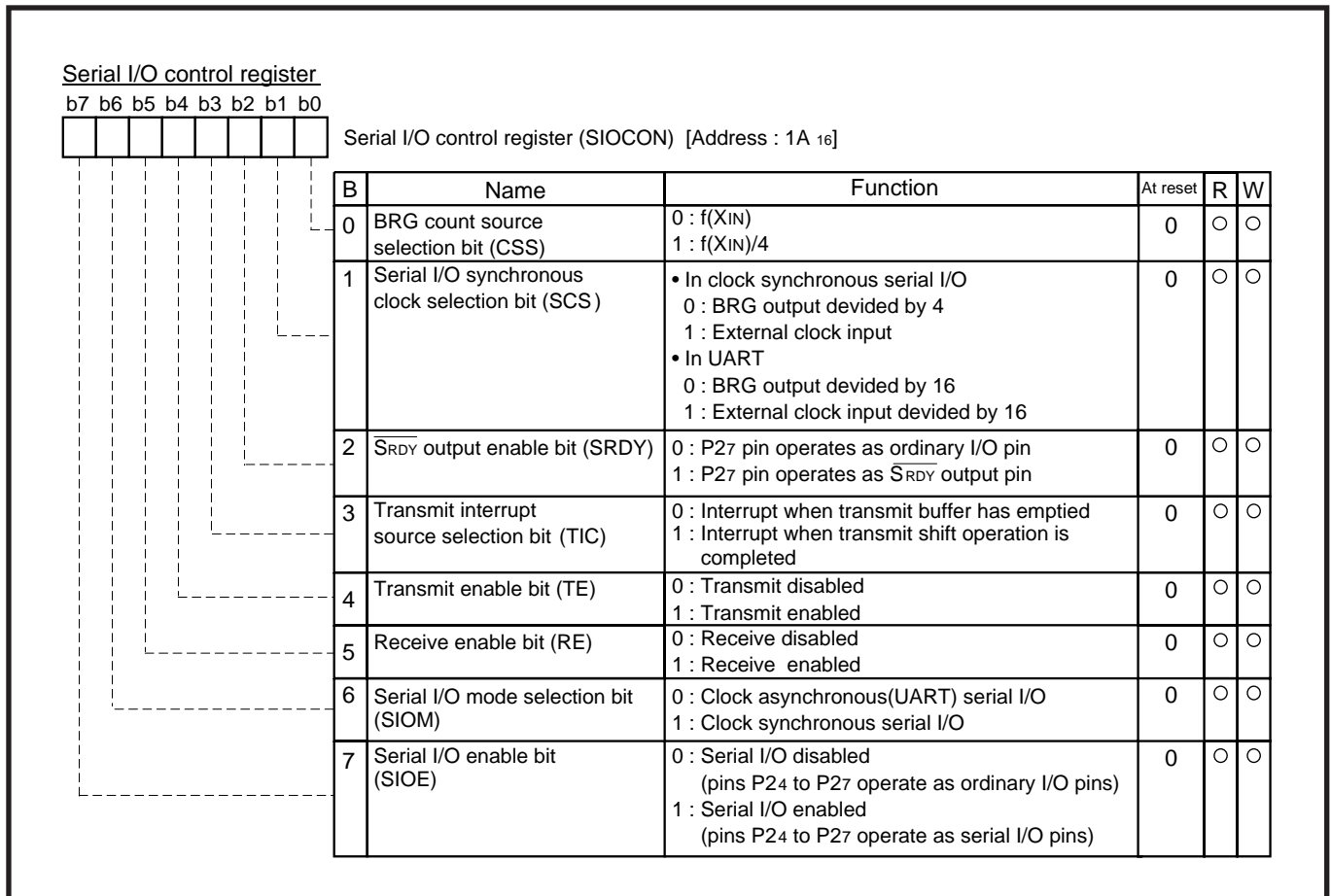


Fig. 2.3.4 Structure of Serial I/O control register

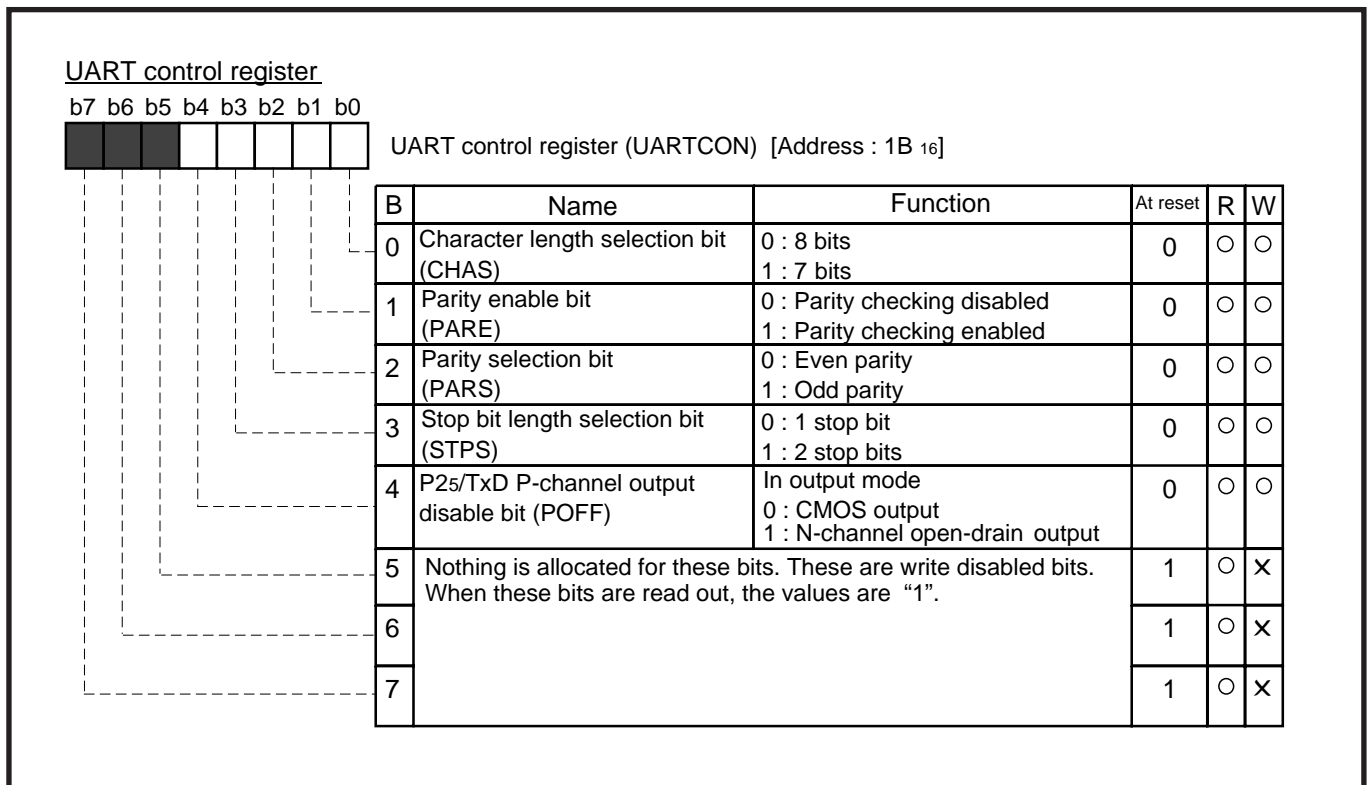


Fig. 2.3.5 Structure of UART control register

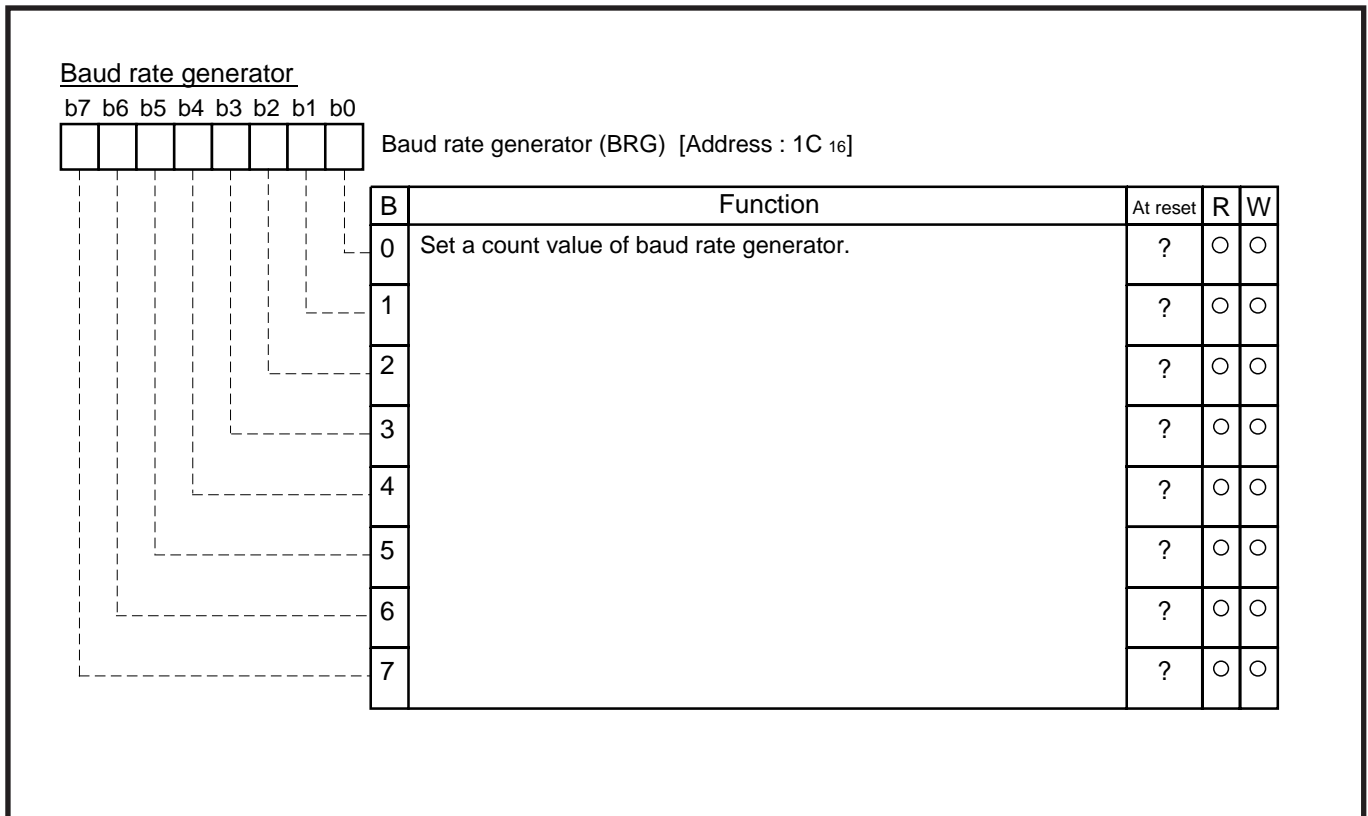


Fig. 2.3.6 Structure of Baud rate generator

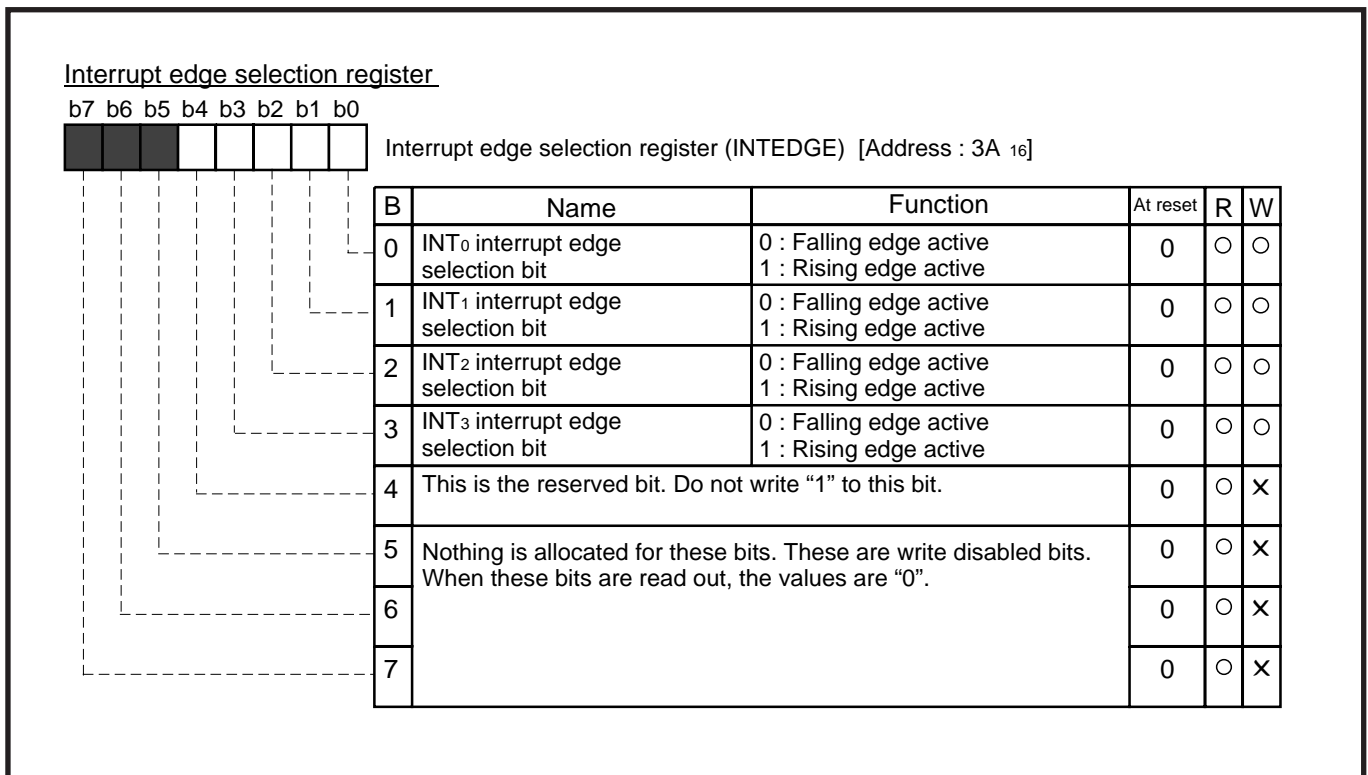


Fig. 2.3.7 Structure of Interrupt edge selection register

APPLICATION

2.3 Serial I/O

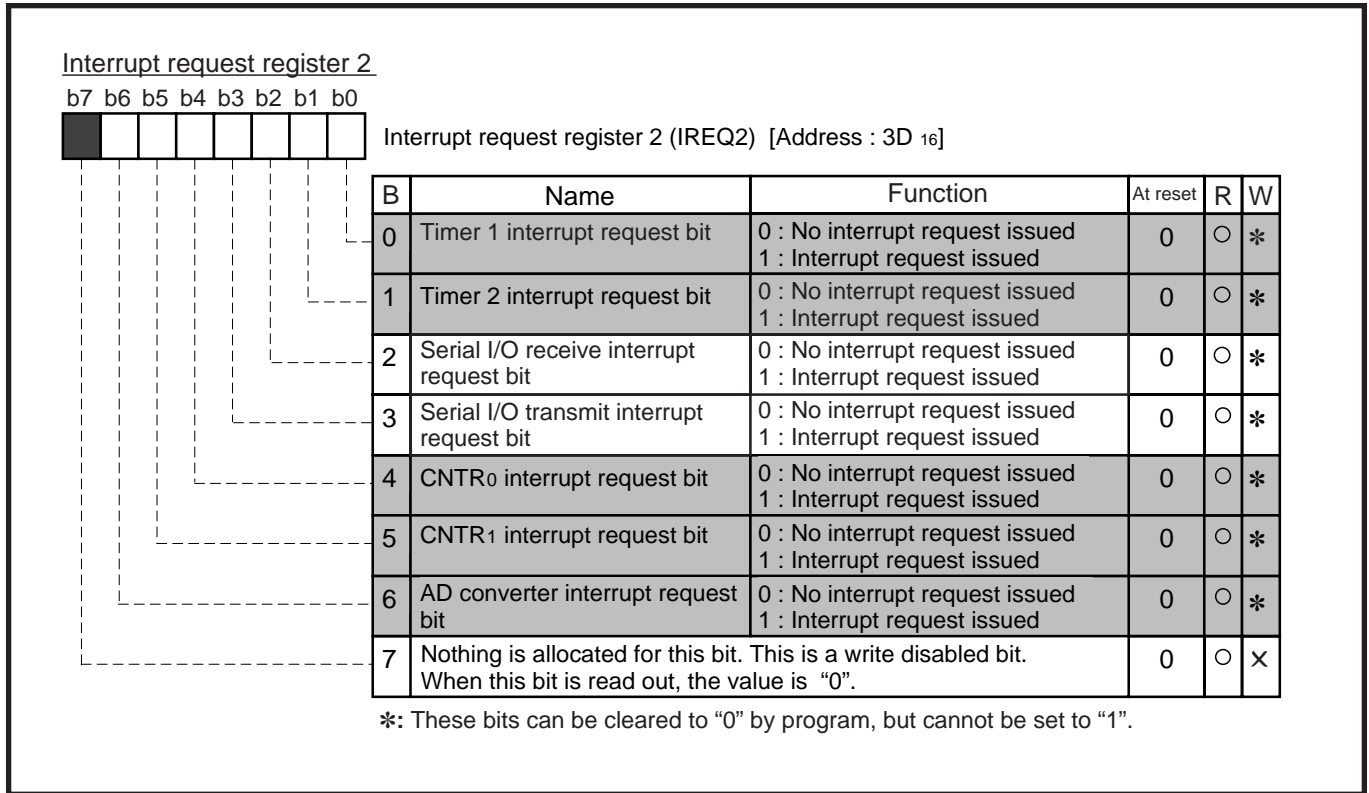


Fig. 2.3.8 Structure of Interrupt request register 2

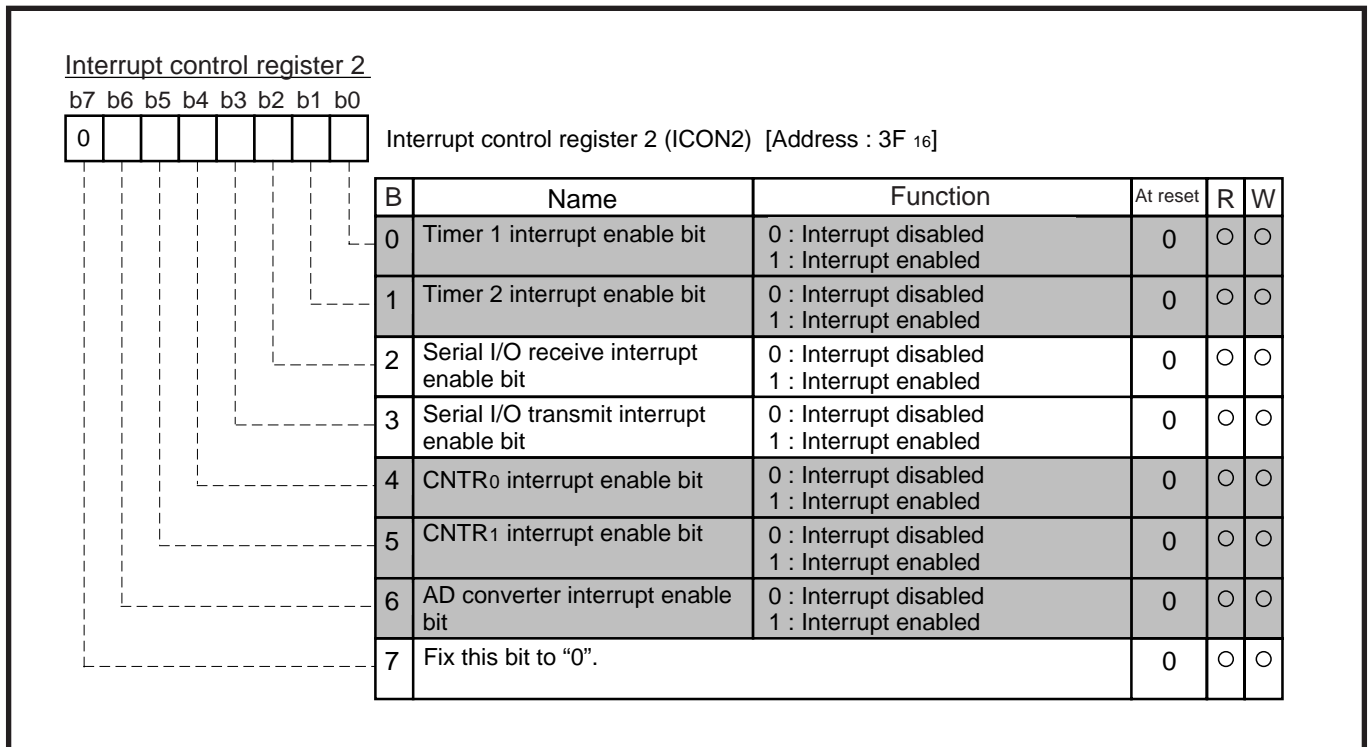


Fig. 2.3.9 Structure of Interrupt control register 2

2.3.3 Serial I/O connection examples

(1) Control of peripheral IC equipped with CS pin

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

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

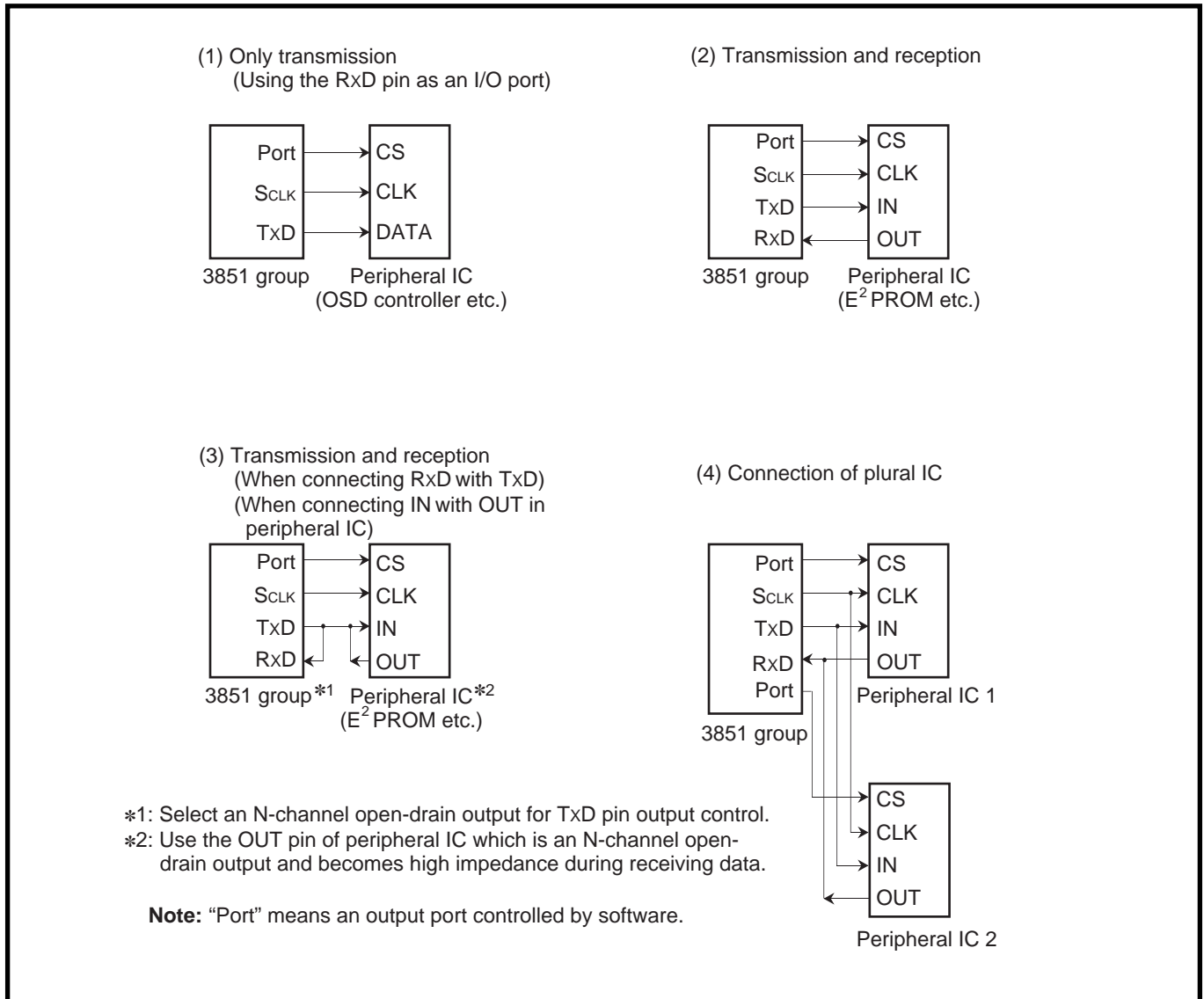


Fig. 2.3.10 Serial I/O connection examples (1)

APPLICATION

2.3 Serial I/O

(2) Connection with microcomputer

Figure 2.3.11 shows connection examples with another microcomputer.

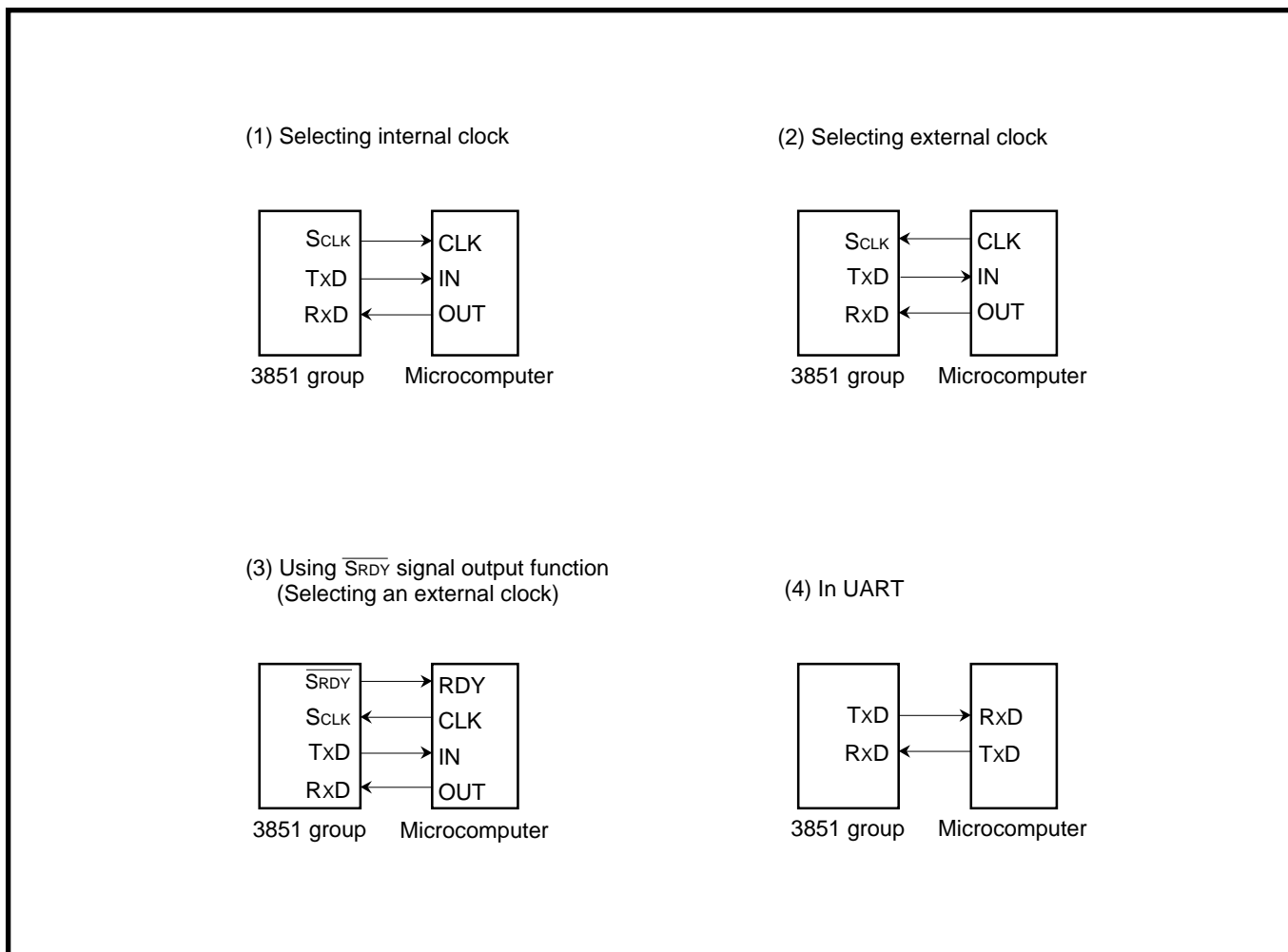


Fig. 2.3.11 Serial I/O connection examples (2)

2.3.4 Setting of serial I/O transfer data format

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

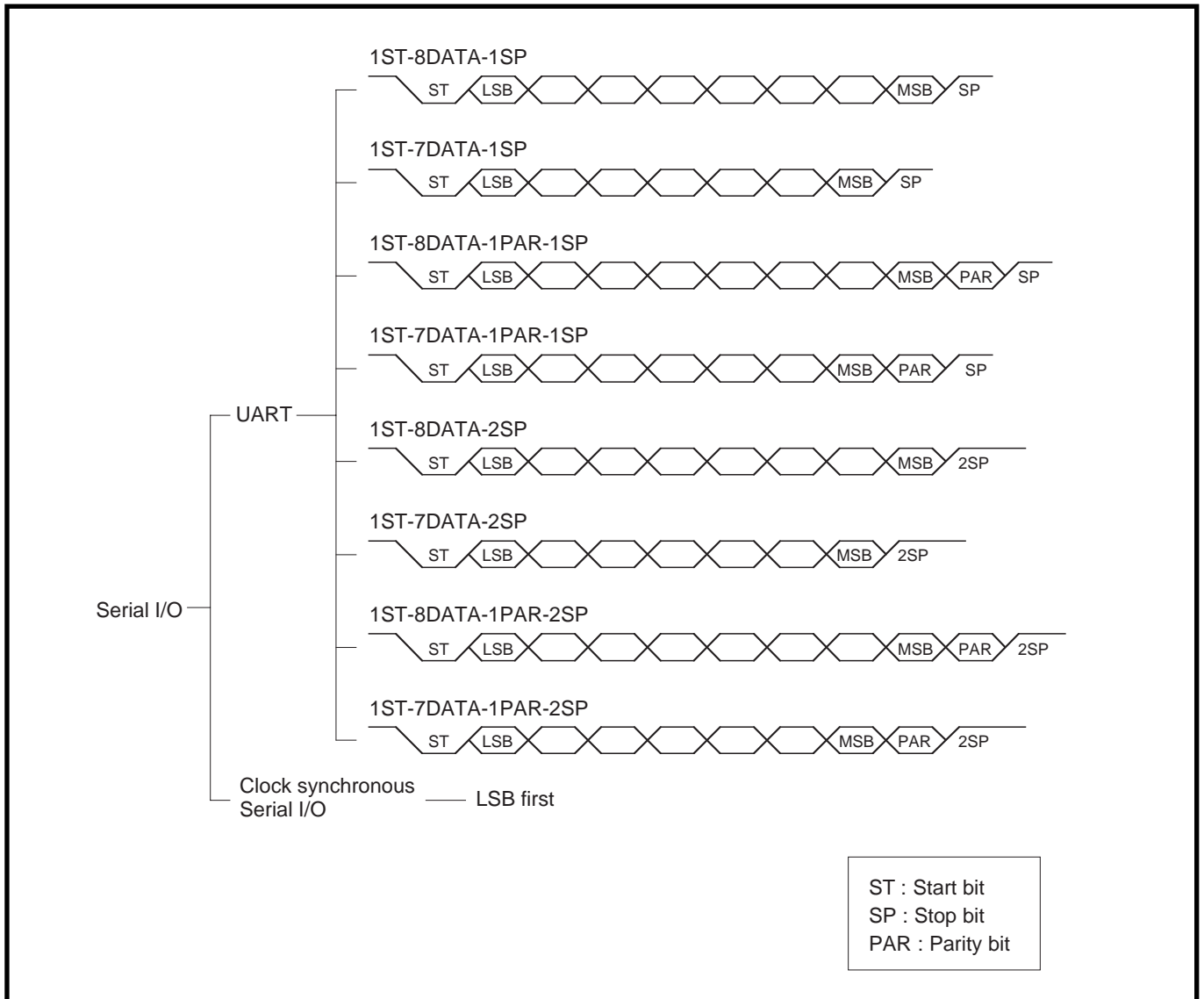


Fig. 2.3.12 Serial I/O transfer data format

APPLICATION

2.3 Serial I/O

2.3.5 Serial I/O application examples

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

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

Figure 2.3.13 shows a connection diagram, and Figure 2.3.14 shows a timing chart.
Figure 2.3.15 shows a registers setting relevant to the transmitting side, and Figure 2.3.16 shows registers setting relevant to the receiving side.

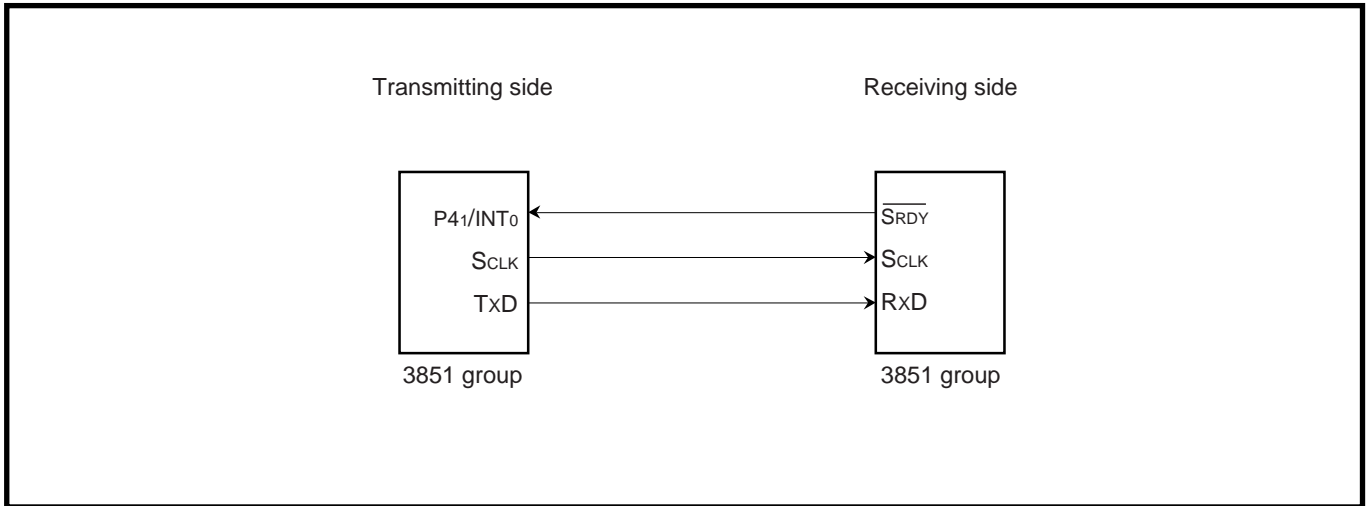


Fig. 2.3.13 Connection diagram

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

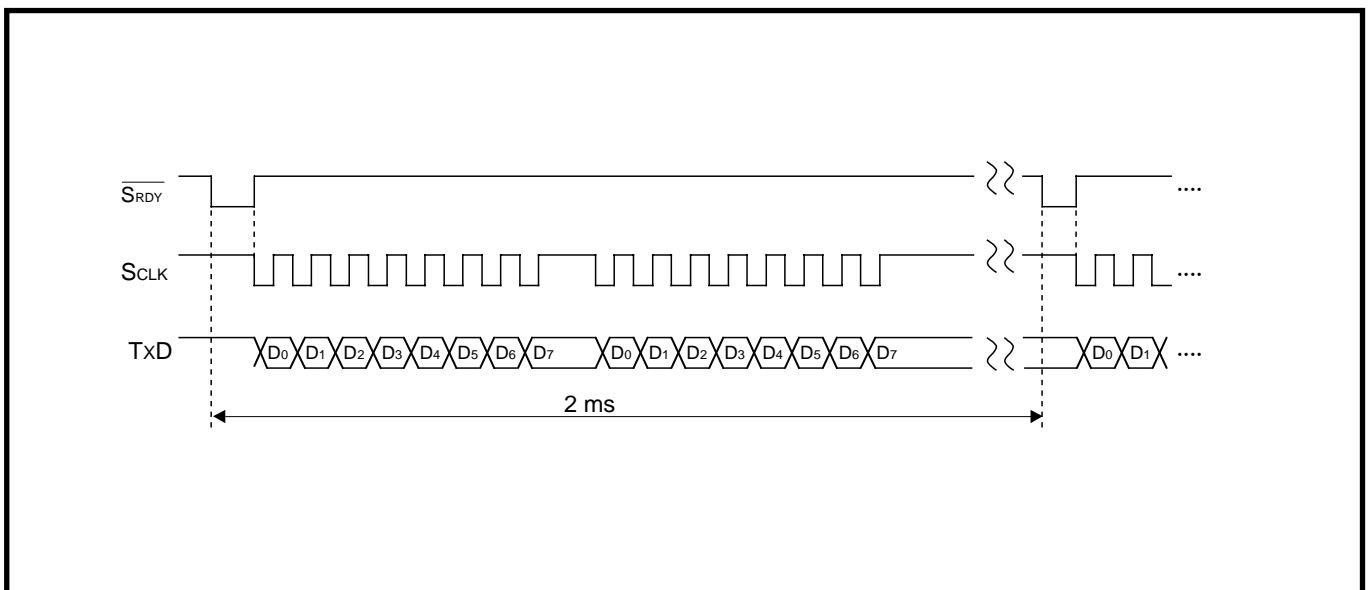


Fig. 2.3.14 Timing chart (using clock synchronous serial I/O)

Transmitting side

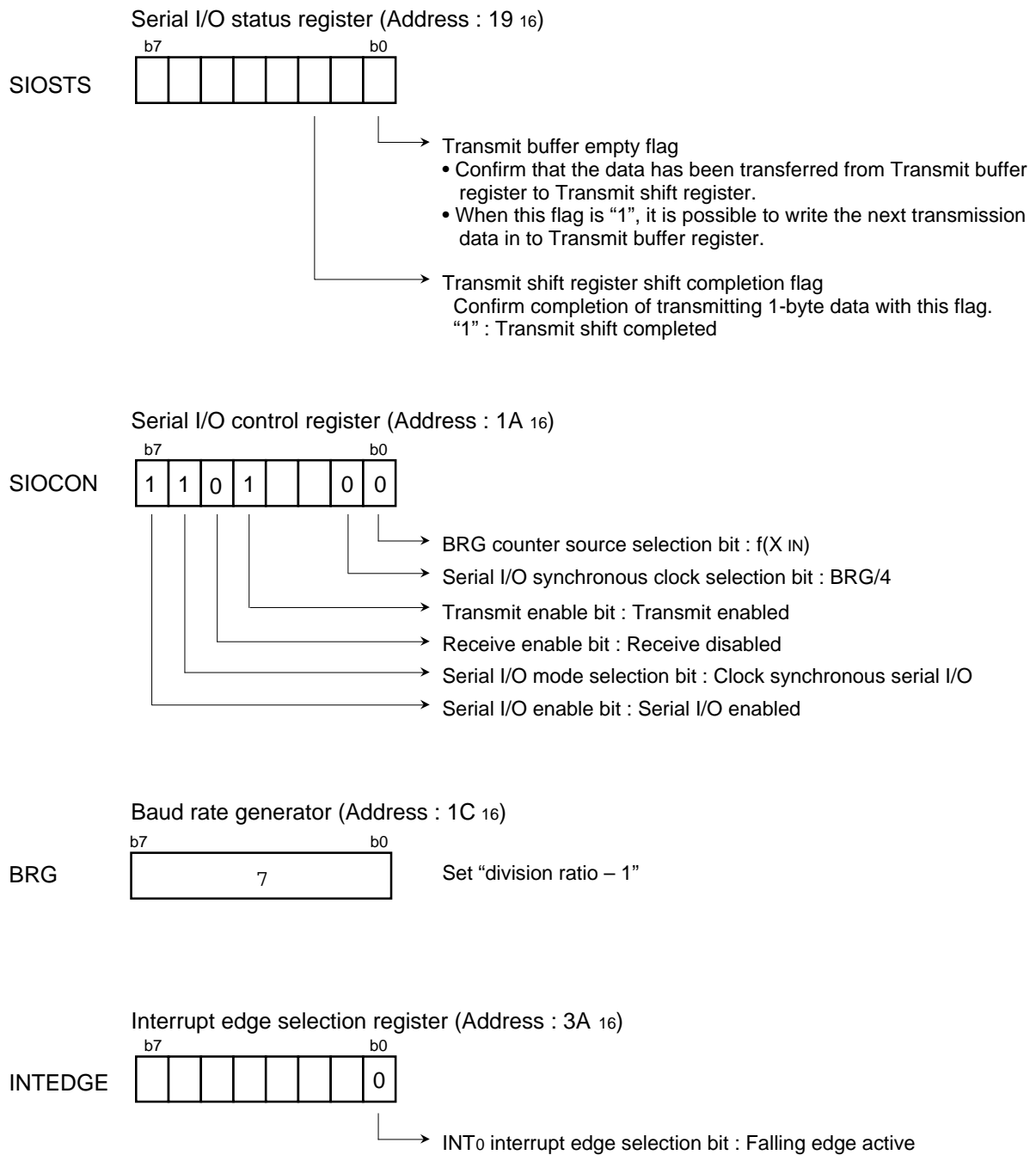


Fig. 2.3.15 Registers setting relevant to transmitting side

APPLICATION

2.3 Serial I/O

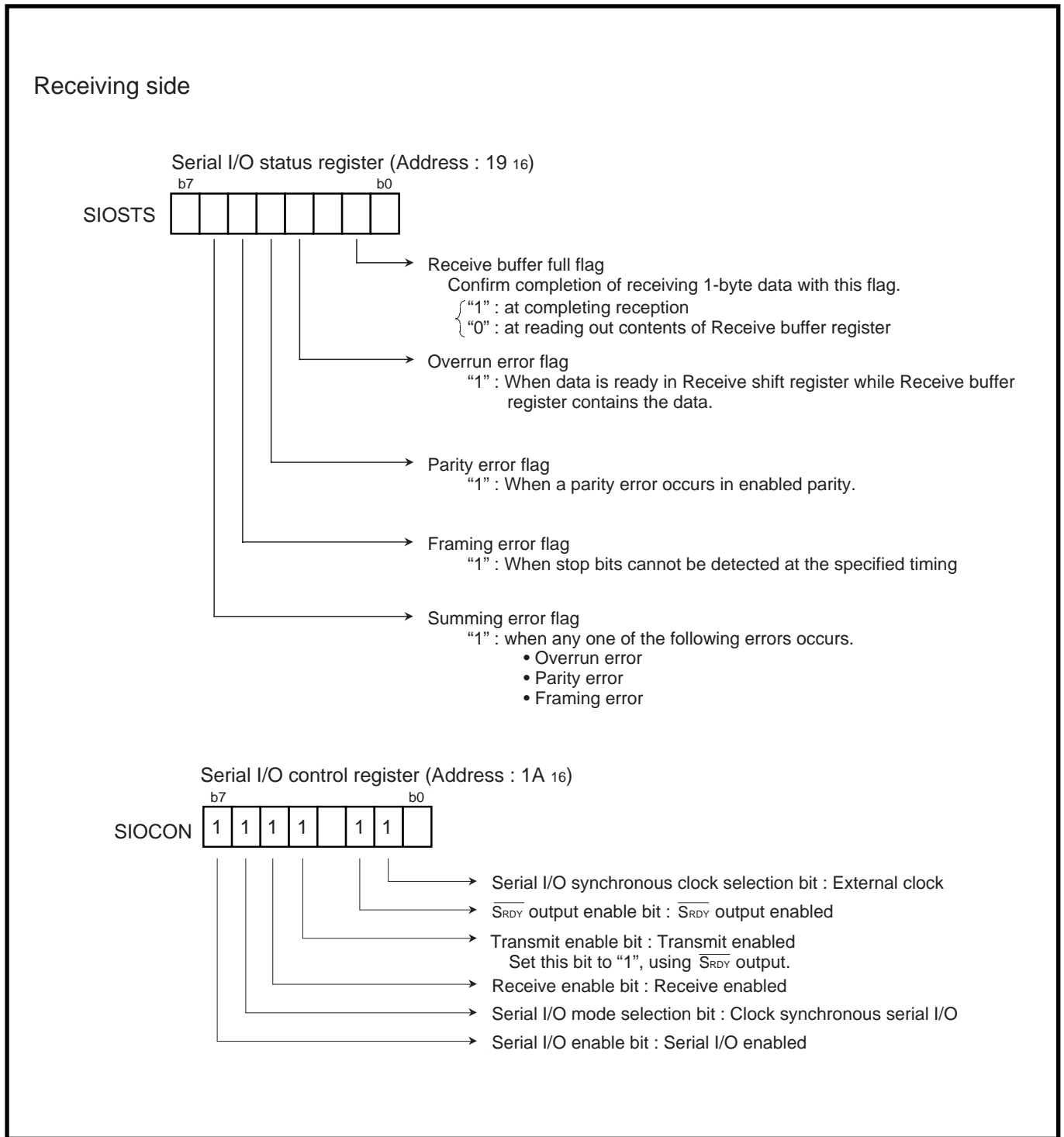


Fig. 2.3.16 Registers setting relevant to receiving side

Figure 2.3.17 shows a control procedure of the transmitting side, and Figure 2.3.18 shows a control procedure of the receiving side.

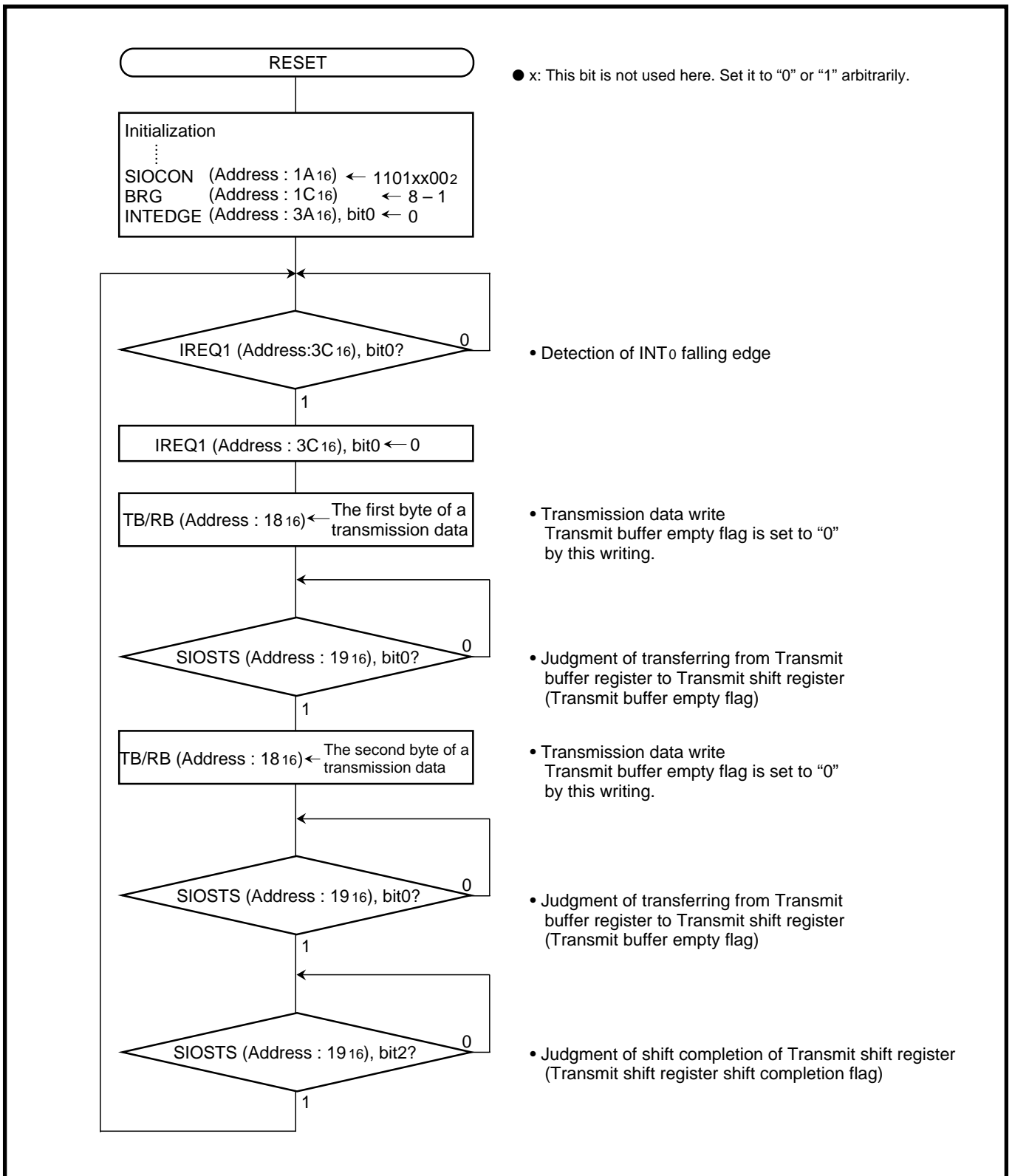


Fig. 2.3.17 Control procedure of transmitting side

APPLICATION

2.3 Serial I/O

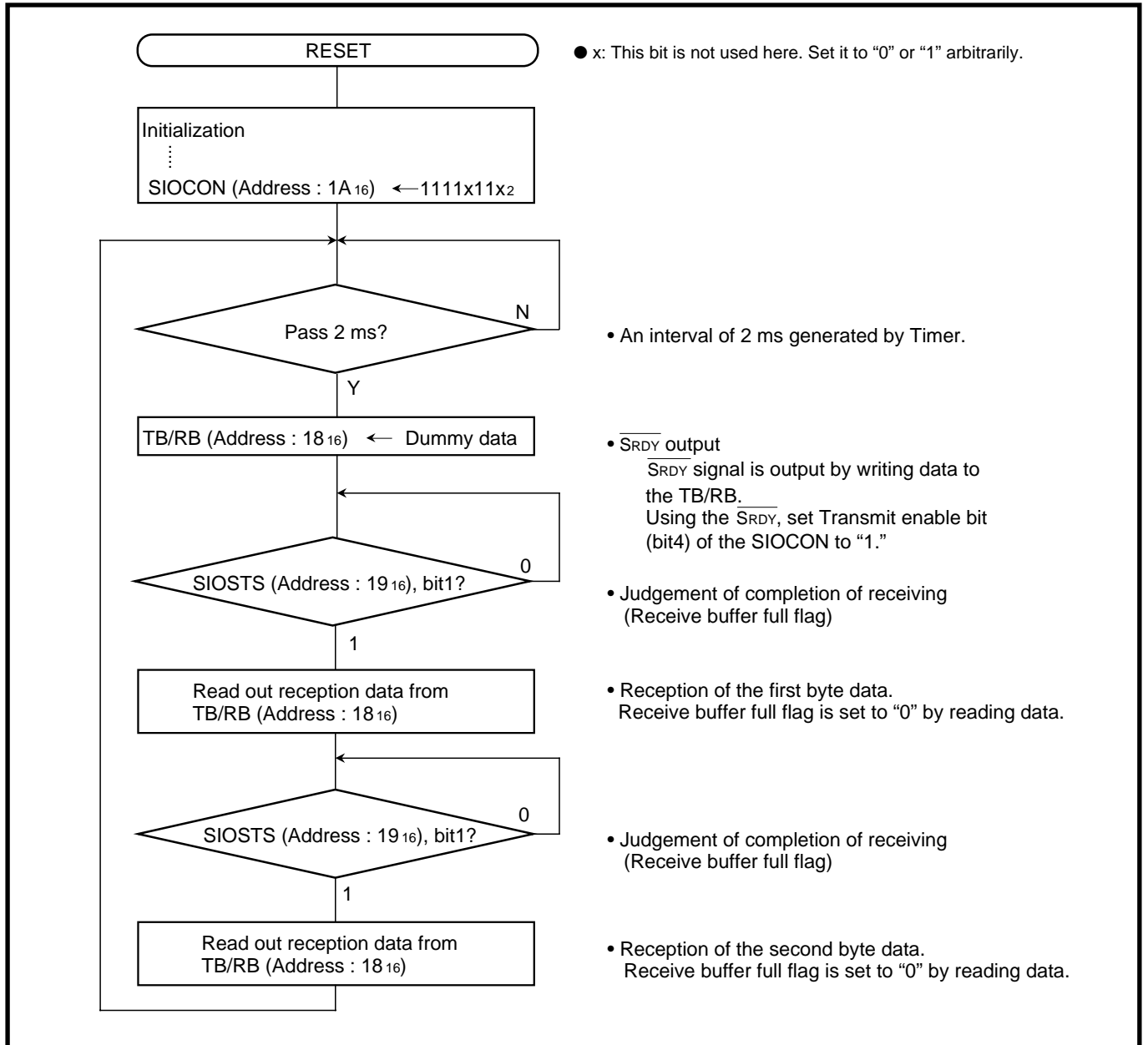


Fig. 2.3.18 Control procedure of receiving side

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

Outline : 4-byte data is transmitted and received, using the clock synchronous serial I/O.
The CS signal is output to a peripheral IC through port P4₃.

Figure 2.3.19 shows a connection diagram, and Figure 2.3.20 shows a timing chart.

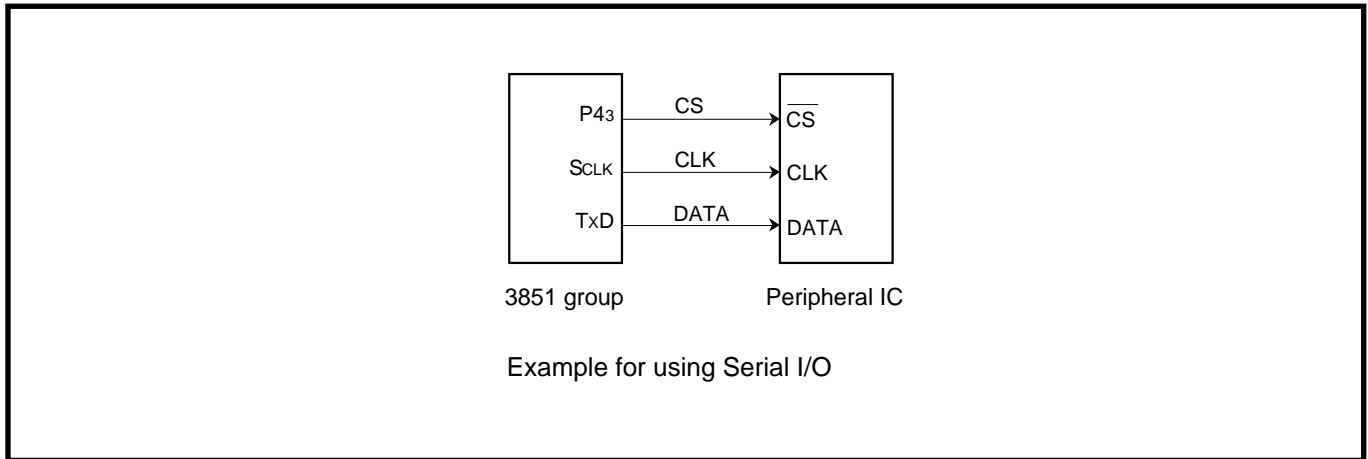


Fig. 2.3.19 Connection diagram

- Specifications :**
- The Serial I/O is used (clock synchronous serial I/O is selected.)
 - Synchronous clock frequency : 125 kHz ($f(X_{IN}) = 4 \text{ MHz}$ is divided by 32)
 - Transfer direction : LSB first
 - The Serial I/O interrupt is not used.
 - Port P4₃ is connected to the \overline{CS} pin ("L" active) of the peripheral IC for transmission control; the output level of port P4₃ is controlled by software.

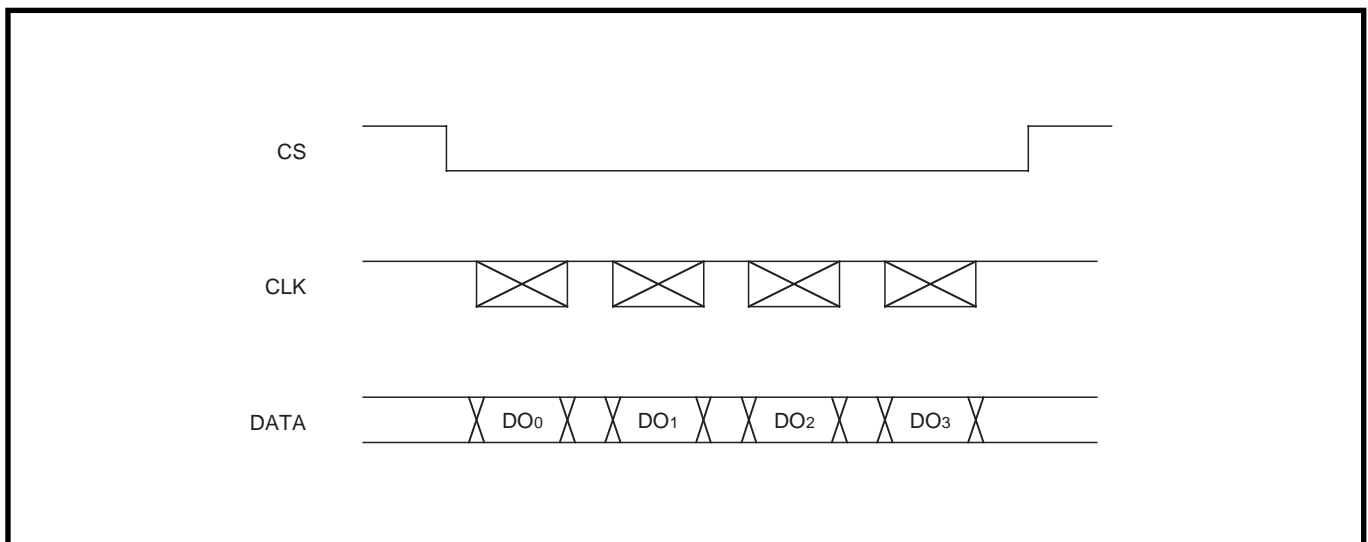


Fig. 2.3.20 Timing chart

APPLICATION

2.3 Serial I/O

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

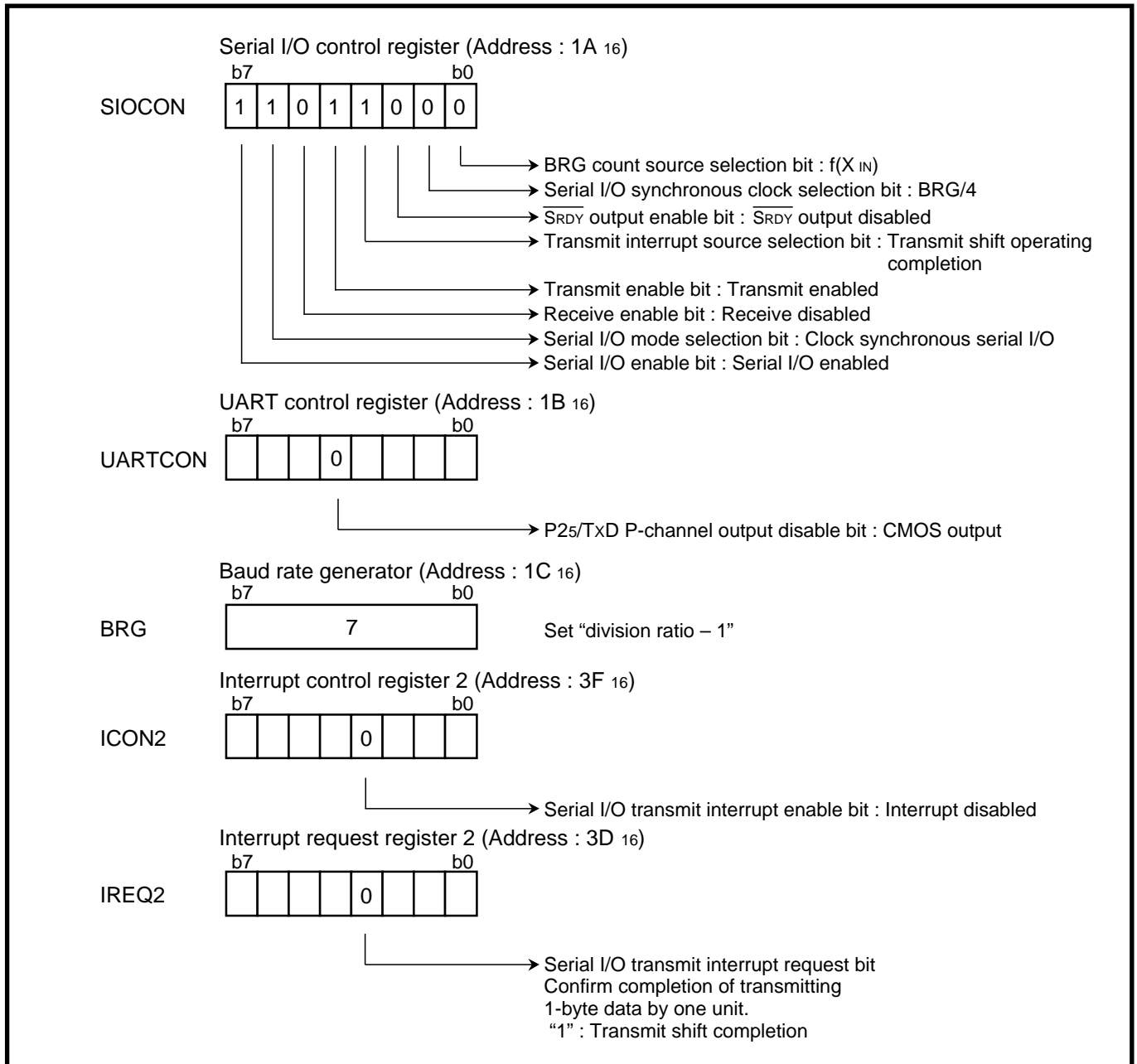


Fig. 2.3.21 Registers setting relevant to serial I/O

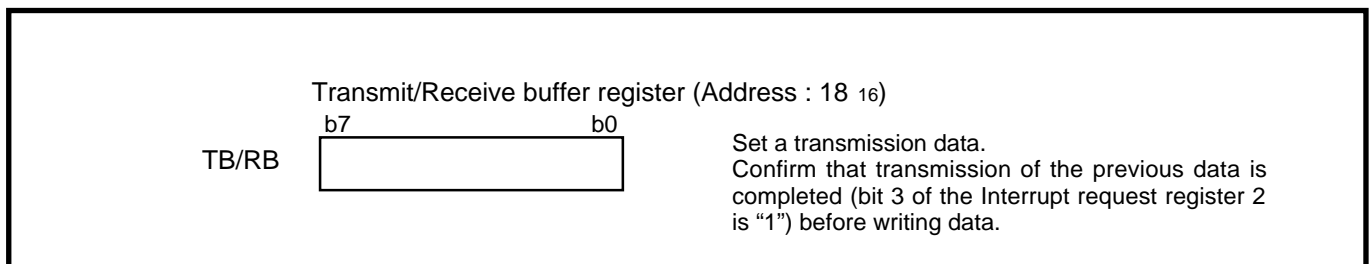


Fig. 2.3.22 Setting of serial I/O transmission data

When the registers are set as shown in Fig. 2.3.21, the Serial I/O can transmit 1-byte data by writing data to the transmit buffer register.

Thus, after setting the CS signal to "L", write the transmission data to the transmit buffer register by each 1 byte, and return the CS signal to "H" when the target number of bytes has been transmitted. Figure 2.3.23 shows a control procedure of serial I/O.

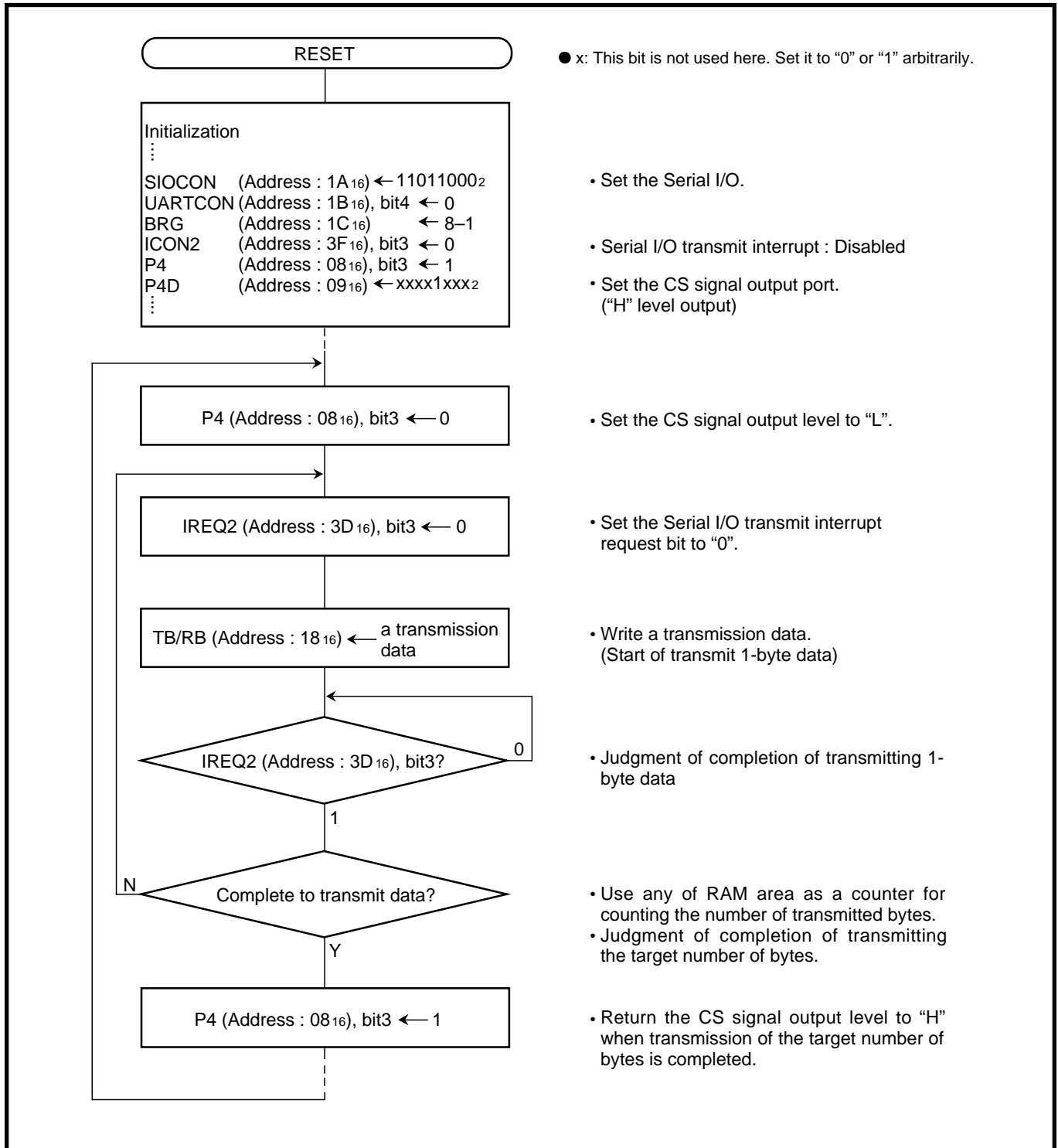


Fig. 2.3.23 Control procedure of serial I/O

APPLICATION

2.3 Serial I/O

(3) Cyclic transmission or reception of block data (data of specified number of bytes) between two microcomputers

Outline : When the clock synchronous serial I/O is used for communication, synchronization of the clock and the data between the transmitting and receiving sides may be lost because of noise included in the synchronous clock. It is necessary to correct that constantly, using “heading adjustment”.

This “heading adjustment” is carried out by using the interval between blocks in this example.

Figure 2.3.24 shows a connection diagram.

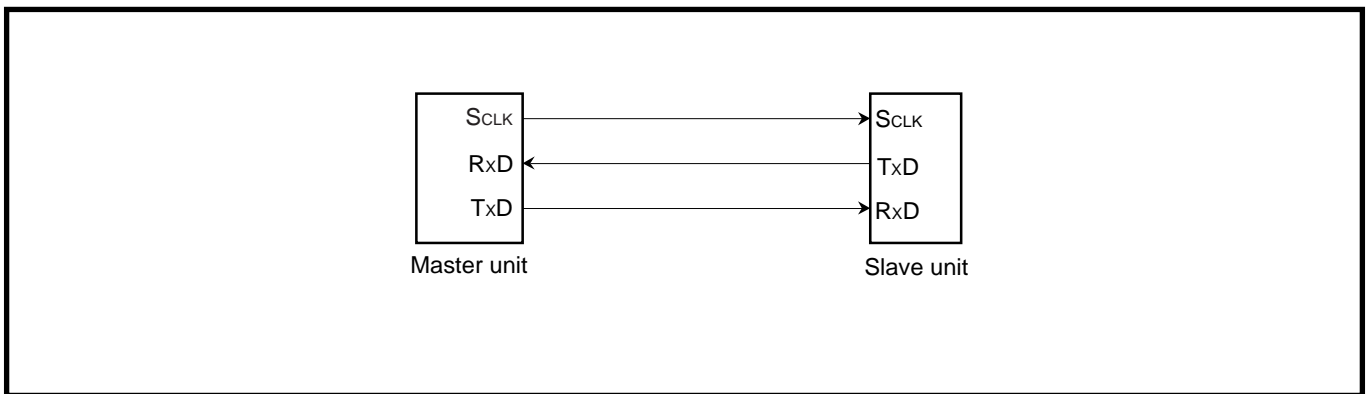


Fig. 2.3.24 Connection diagram

Specifications :

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

Limitations of the specifications :

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

The communication is performed according to the timing shown in Figure 2.3.25. In the slave unit, when a synchronous clock is not input within a certain time (heading adjustment time), the next clock input is processed as the beginning (heading) of a block.

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

Figure 2.3.26 shows relevant registers setting.

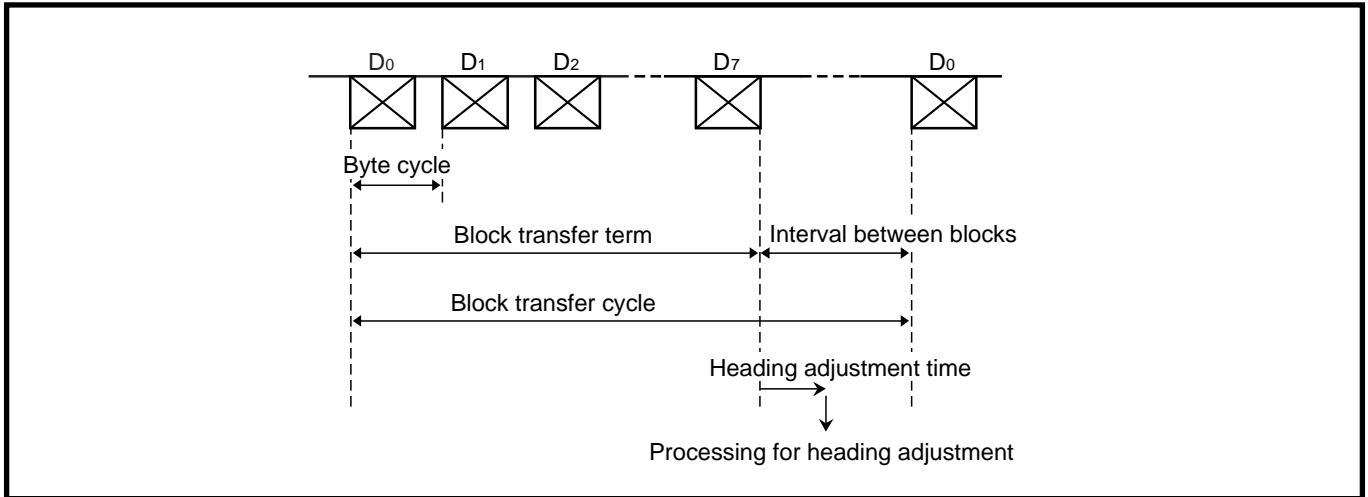


Fig. 2.3.25 Timing chart

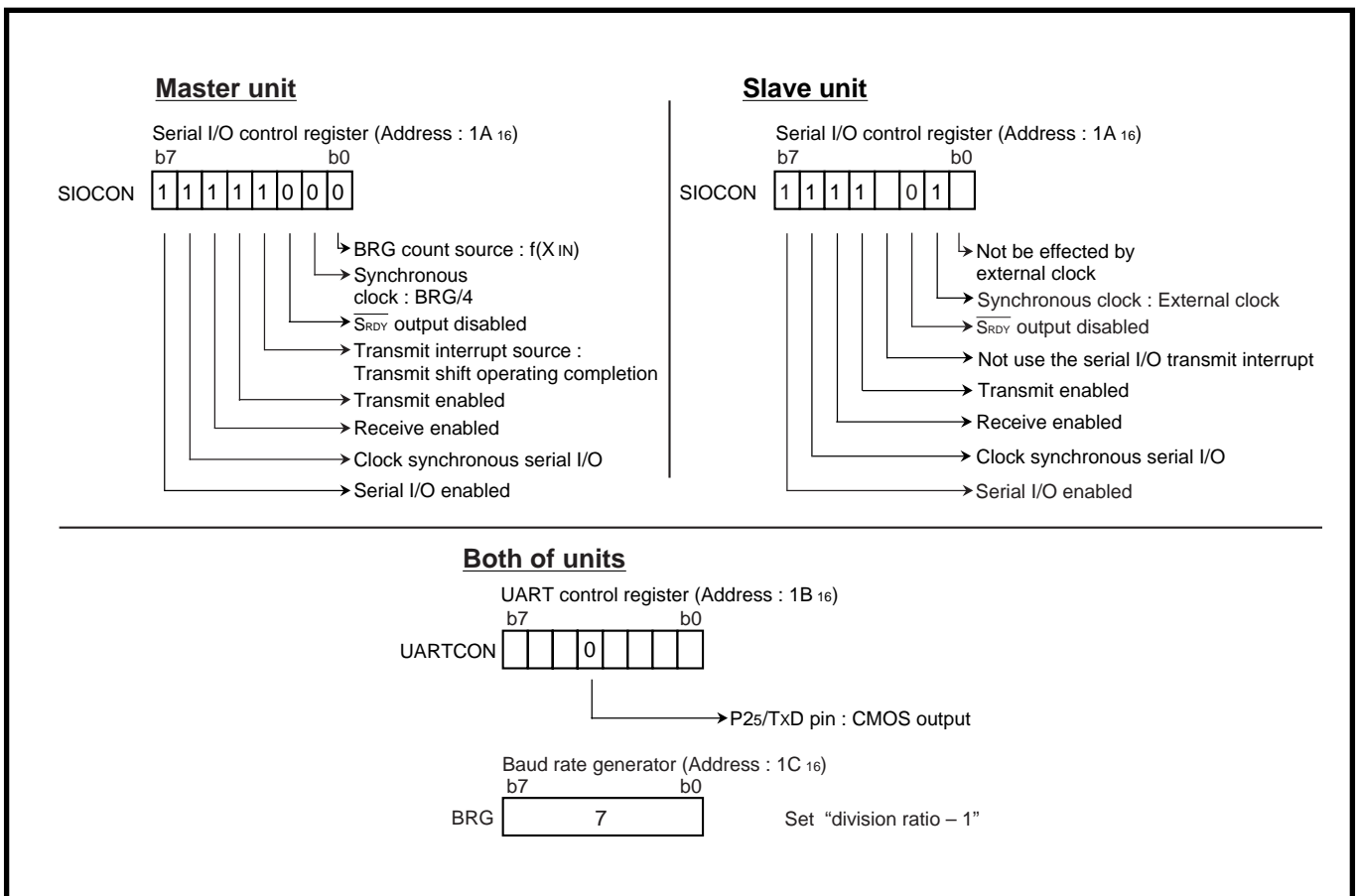


Fig. 2.3.26 Relevant registers setting

APPLICATION

2.3 Serial I/O

Control procedure :

- Control in the master unit

After setting the relevant registers shown in Figure 2.3.26, the master unit starts transmission or reception of 1-byte data by writing transmission data to the transmit buffer register.

To perform the communication in the timing shown in Figure 2.3.25, take the timing into account and write transmission data. Additionally, 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.

Figure 2.3.27 shows a control procedure of the master unit using timer interrupts.

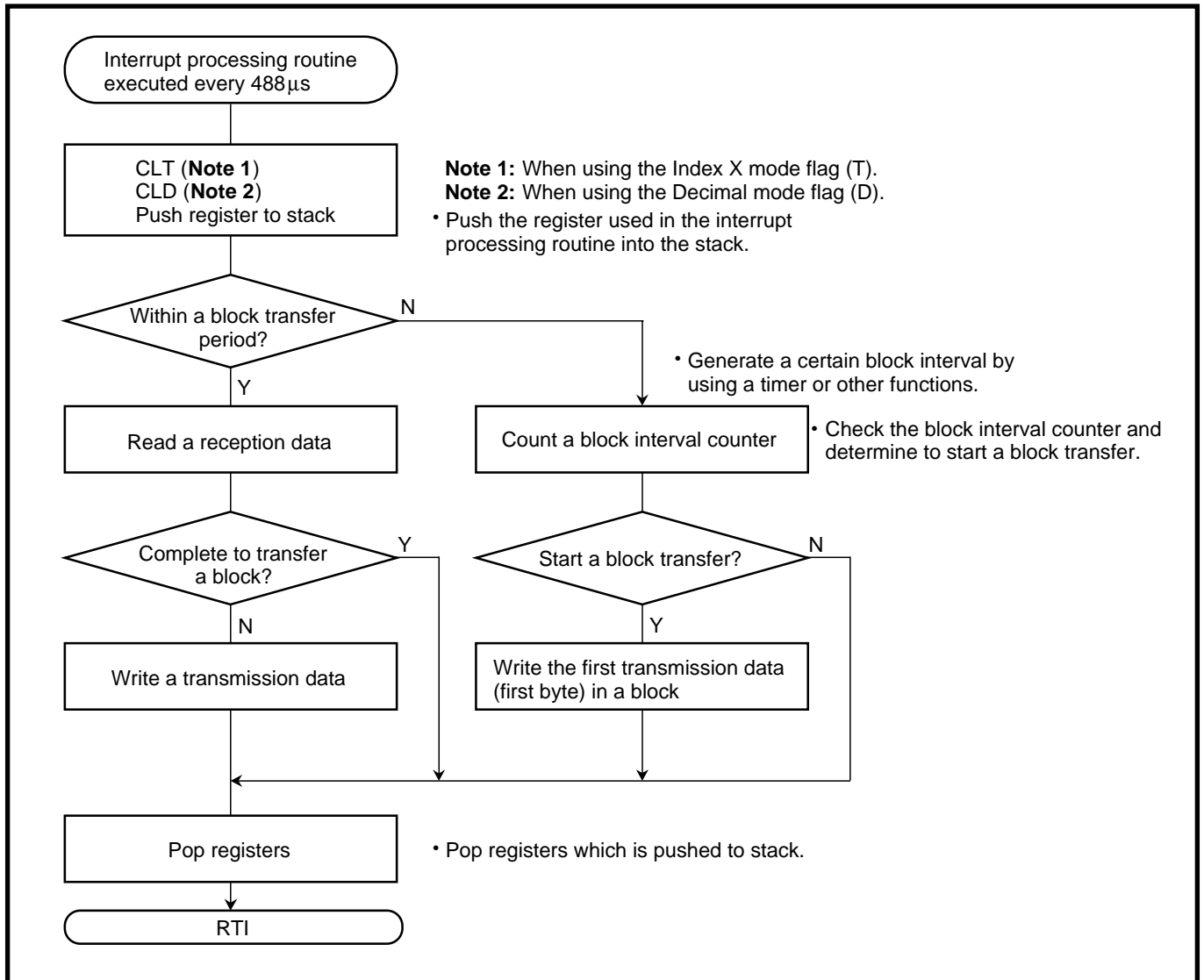


Fig. 2.3.27 Control procedure of master unit

● Control in the slave unit

After setting the relevant registers as shown in Figure 2.3.26, the slave unit becomes the state where a synchronous clock can be received at any time, and the serial I/O receive interrupt request bit is set to "1" each time an 8-bit synchronous clock is received.

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

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

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

Figure 2.3.28 shows a control procedure of the slave unit using the serial I/O receive interrupt and any timer interrupt (for heading adjustment).

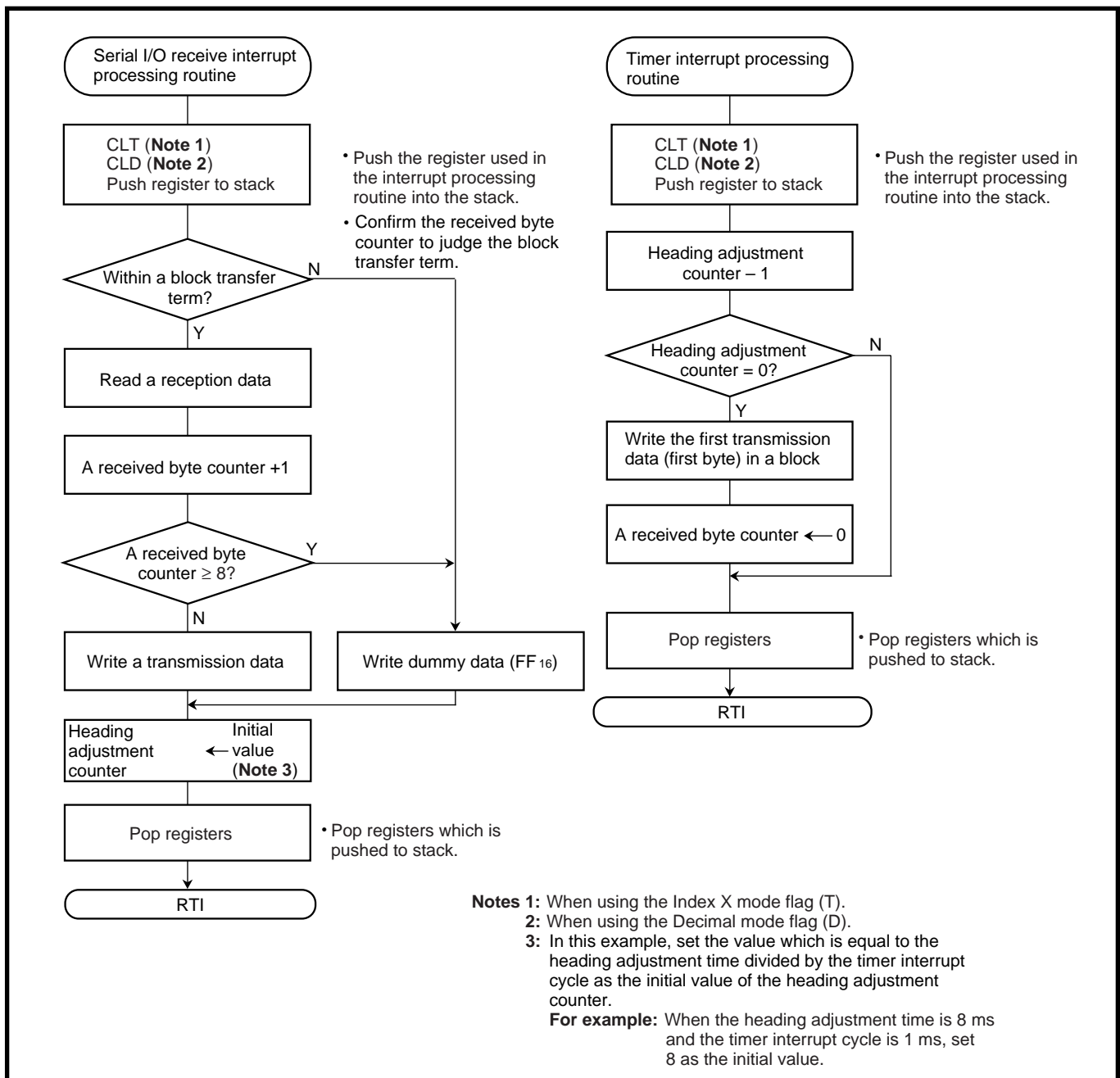


Fig. 2.3.28 Control procedure of slave unit

APPLICATION

2.3 Serial I/O

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

Outline : 2-byte data is transmitted and received, using the asynchronous serial I/O.
Port P4₀ is used for communication control.

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

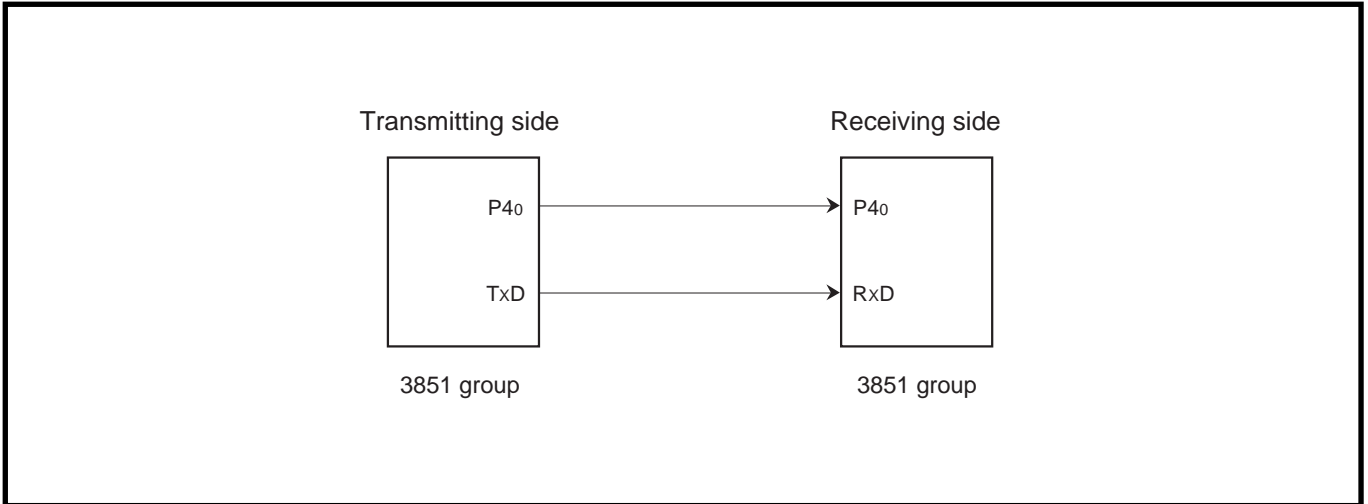


Fig. 2.3.29 Connection diagram (Communication using UART)

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

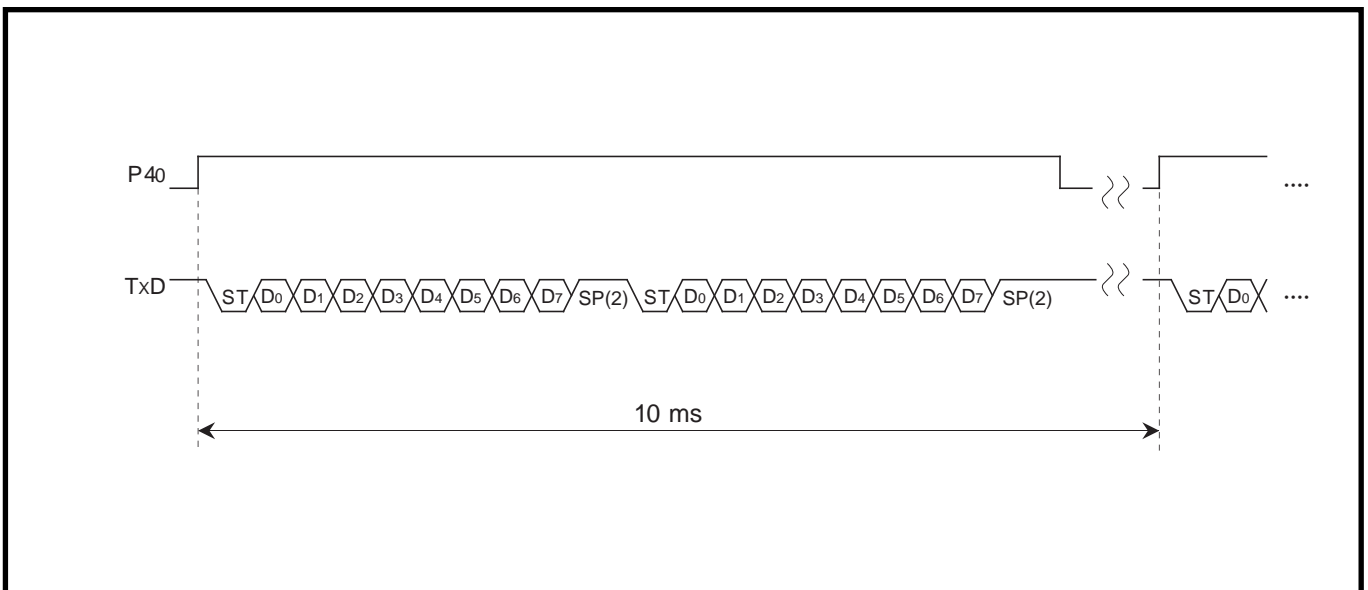


Fig. 2.3.30 Timing chart (using UART)

Table 2.3.1 shows setting examples of the baud rate generator (BRG) values and transfer bit rate values; Figure 2.3.31 shows registers setting relevant to the transmitting side; Figure 2.3.32 shows registers setting relevant to the receiving side.

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

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

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

2: Equation of transfer bit rate:

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

*m: When bit 0 of the Serial I/O control register (Address : 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.

APPLICATION

2.3 Serial I/O

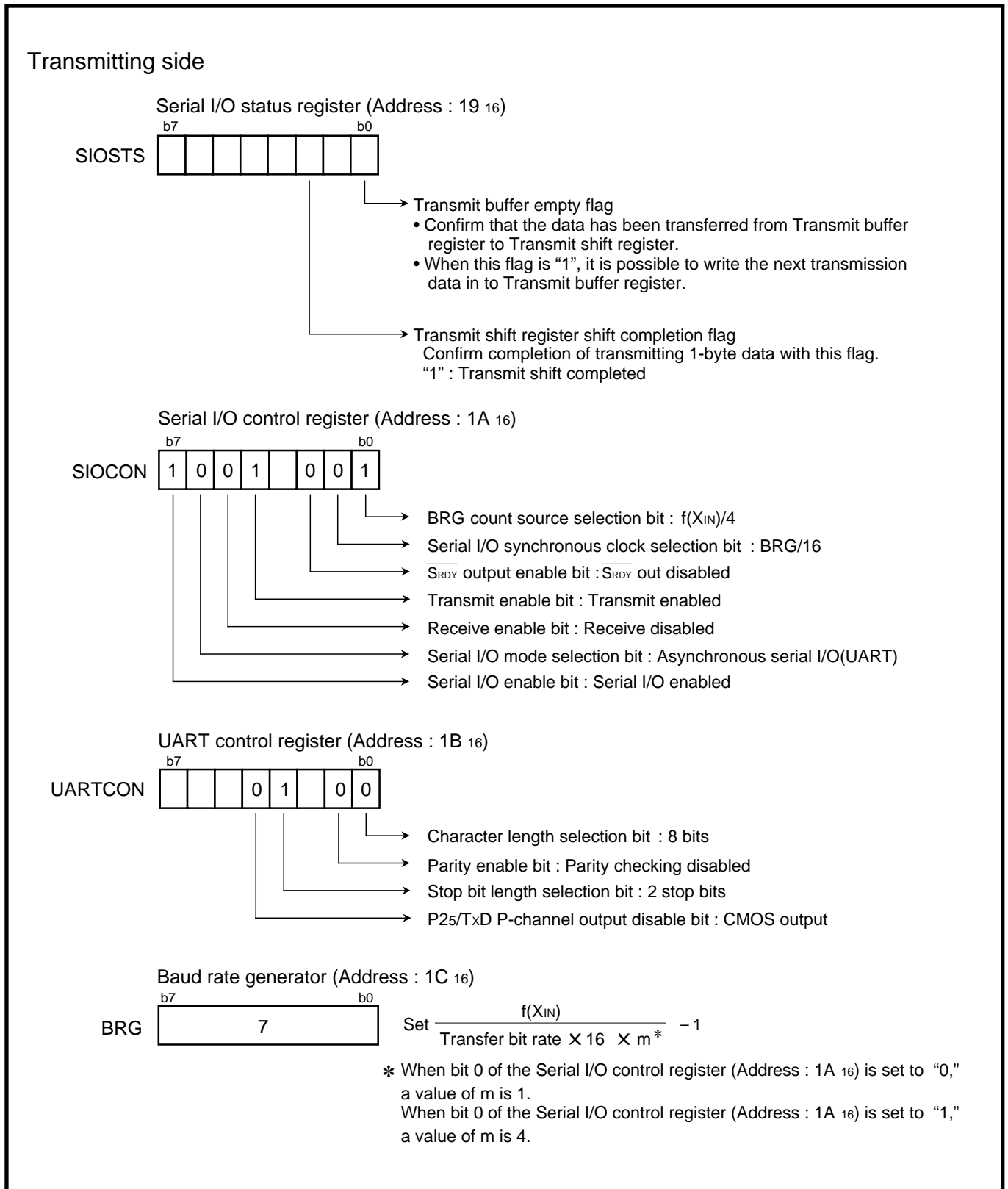


Fig. 2.3.31 Registers setting relevant to transmitting side

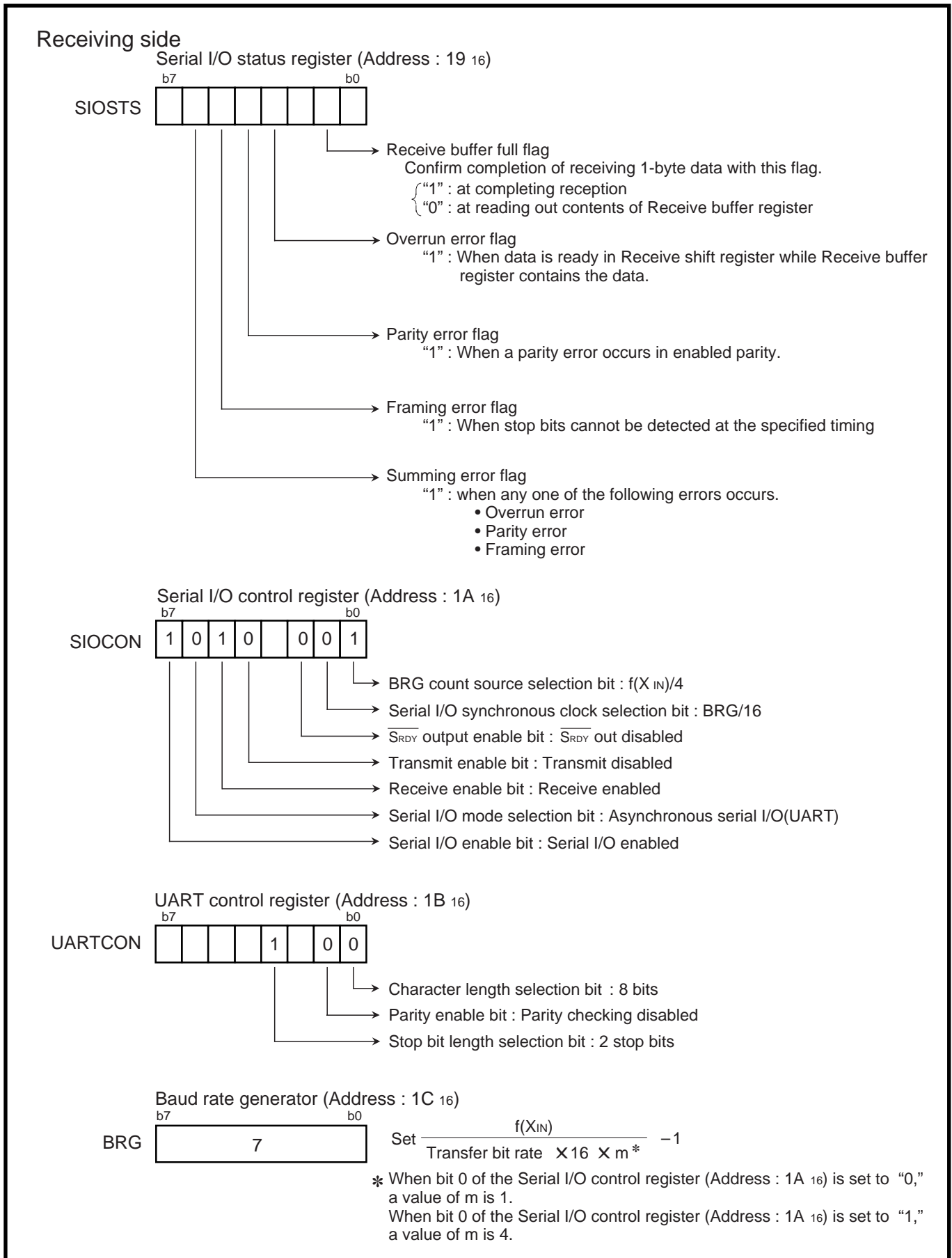


Fig. 2.3.32 Registers setting relevant to receiving side

APPLICATION

2.3 Serial I/O

Figure 2.3.33 shows a control procedure of the transmitting side, and Figure 2.3.34 shows a control procedure of the receiving side.

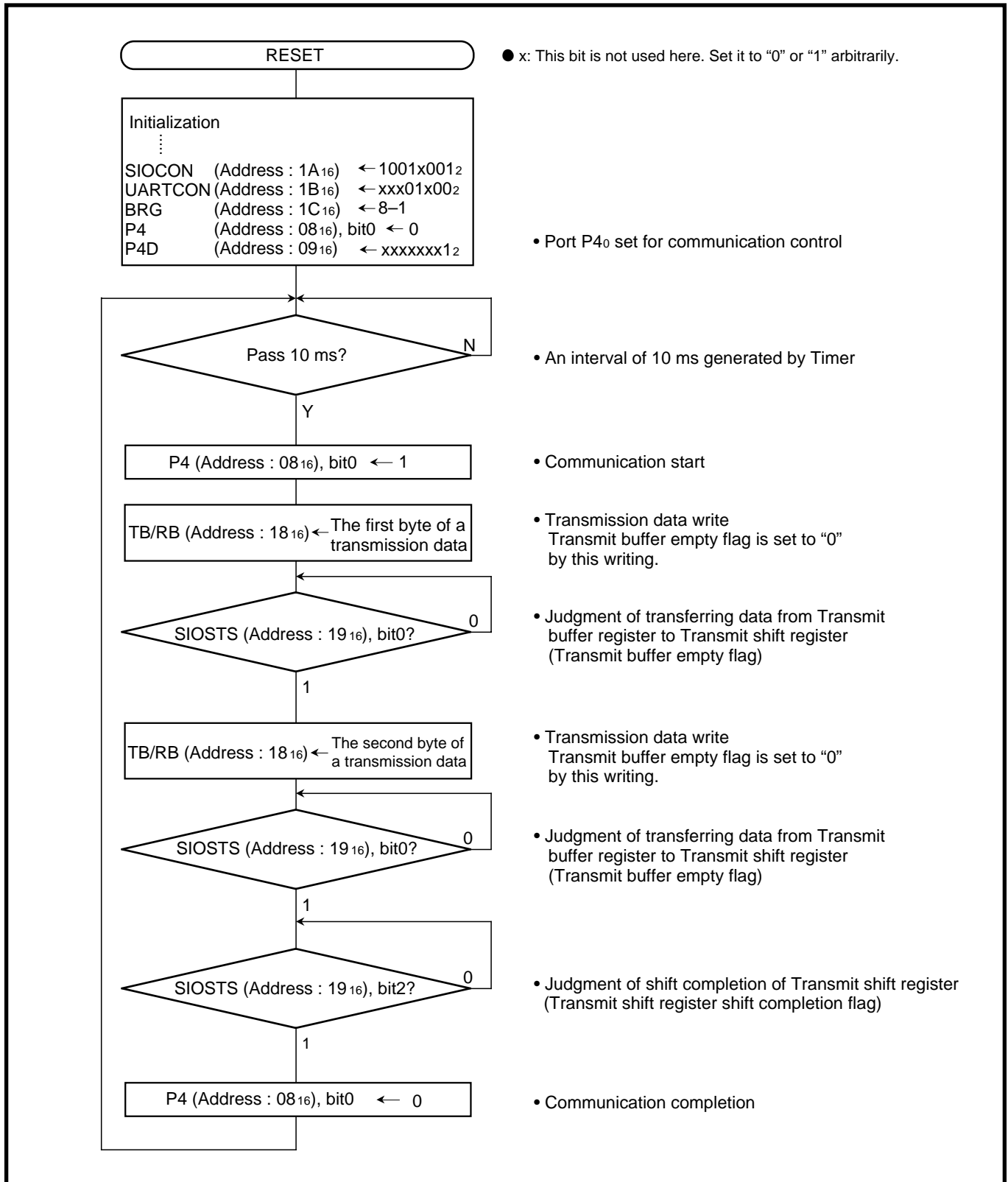


Fig. 2.3.33 Control procedure of transmitting side

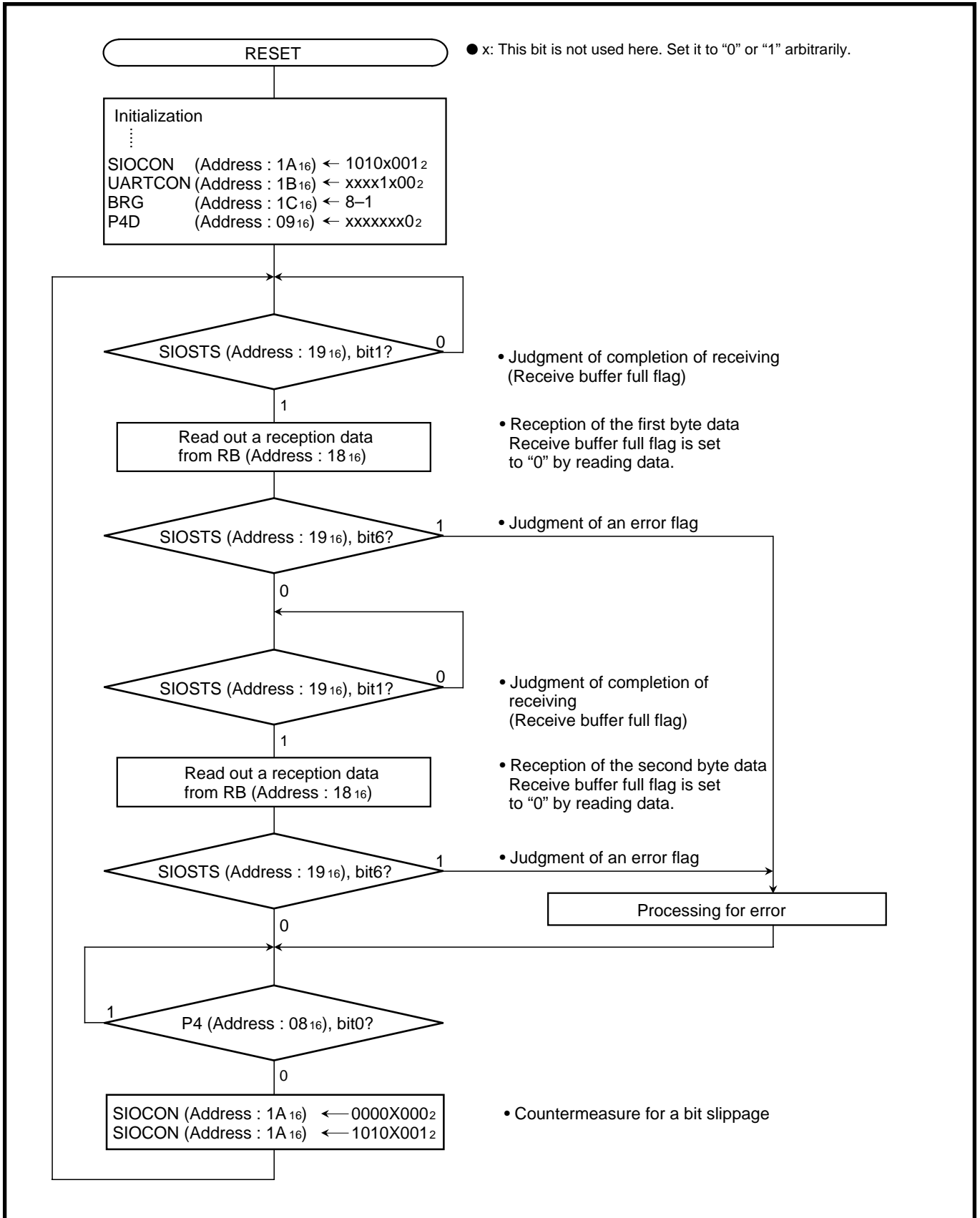


Fig. 2.3.34 Control procedure of receiving side

APPLICATION

2.3 Serial I/O

2.3.6 Notes on serial I/O

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

① Stop of transmission operation

Clear the serial I/O enable bit and the transmit enable bit to "0" (serial I/O and transmit disabled).

● Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only 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 SRDY function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O enable bit is set to "1" at this time, the data during internally shifting is output to the TxD pin and an operation failure occurs.

② Stop of receive operation

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

③ Stop of transmit/receive operation

Clear both the transmit enable bit and receive enable bit to "0" (transmit and receive disabled). (when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

● Reason

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

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

(2) Notes when selecting clock asynchronous serial I/O**① Stop of transmission operation**

Clear the transmit enable bit to “0” (transmit disabled).

● Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O enable bit is cleared to “0” (serial I/O disabled), the internal transmission is running (in this case, since pins Tx_D, Rx_D, SCLK, and SRDY function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O enable bit is set to “1” at this time, the data during internally shifting is output to the Tx_D pin and an operation failure occurs.

② Stop of receive operation

Clear the receive enable bit to “0” (receive disabled).

③ Stop of transmit/receive operation**Only transmission operation is stopped.**

Clear the transmit enable bit to “0” (transmit disabled).

● Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O enable bit is cleared to “0” (serial I/O disabled), the internal transmission is running (in this case, since pins Tx_D, Rx_D, SCLK, and SRDY function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O enable bit is set to “1” at this time, the data during internally shifting is output to the Tx_D pin and an operation failure occurs.

Only receive operation is stopped.

Clear the receive enable bit to “0” (receive disabled).

(3) SRDY output of reception side

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

(4) Setting serial I/O control register again

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.”

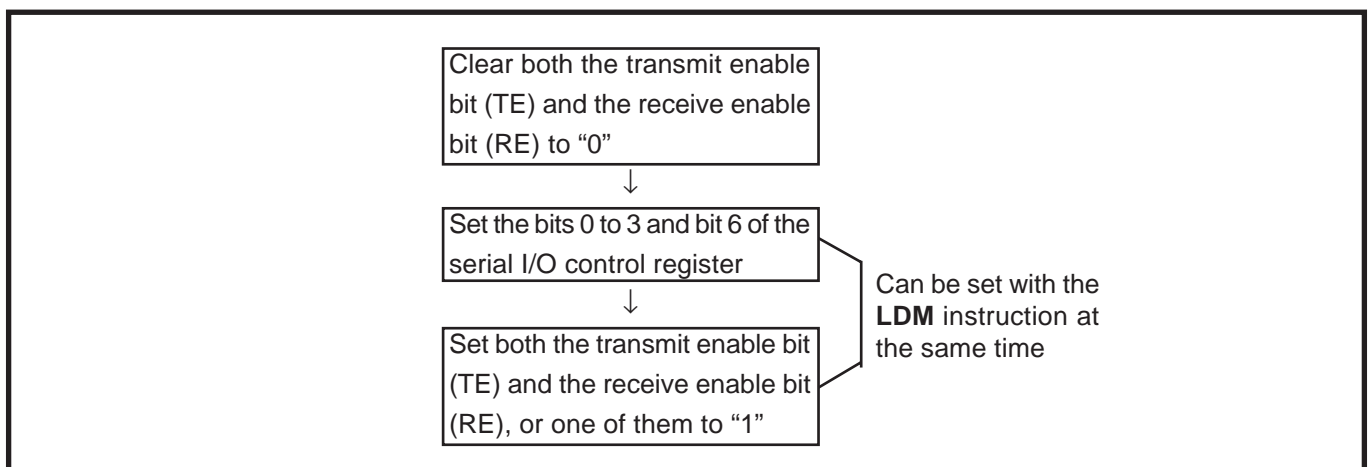


Fig. 2.3.35 Sequence of setting serial I/O control register again

APPLICATION

2.3 Serial I/O

(5) Data transmission control with referring to transmit shift register completion flag

The transmit shift register completion flag changes from “1” to “0” with a delay of 0.5 to 1.5 shift clocks. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

(6) Transmission control when external clock is selected

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

(7) Transmit interrupt request when transmit enable bit is set

The transmission interrupt request bit is set and the interruption request is generated even when selecting timing that either of the following flags is set to “1” as timing where the transmission interruption is generated.

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

Therefore, when the transmit interrupt is used, set the transmit interrupt enable bit to transmit enabled as the following sequence.

- ① Transmit enable bit is set to “1”
- ② Transmit interrupt request bit is set to “0”

● Reason

When the transmission enable bit is set to “1”, the transmit buffer empty flag and transmit shift register completion flag are set to “1”.

2.4 Multi-master I²C-BUS interface

The multi-master I²C-BUS interface is a serial communication circuit, conforming to the Philips I²C-BUS data transfer format.

2.4.1 Memory map

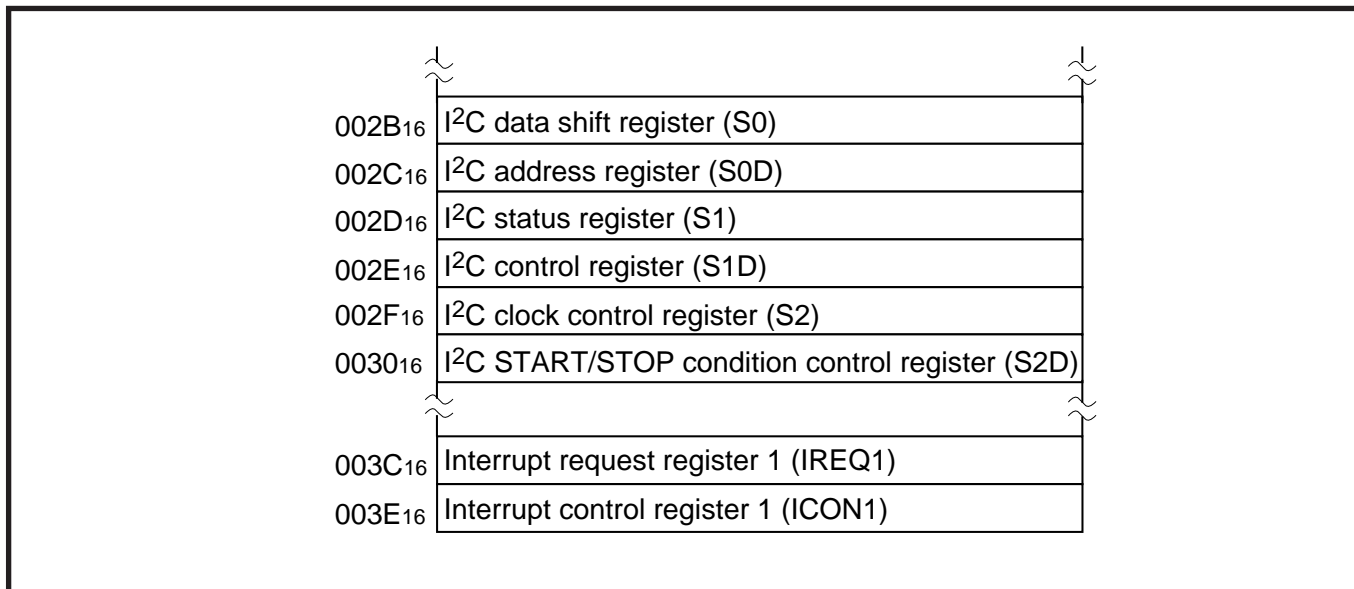


Fig. 2.4.1 Memory map of registers relevant to I²C-BUS interface

2.4.2 Relevant registers

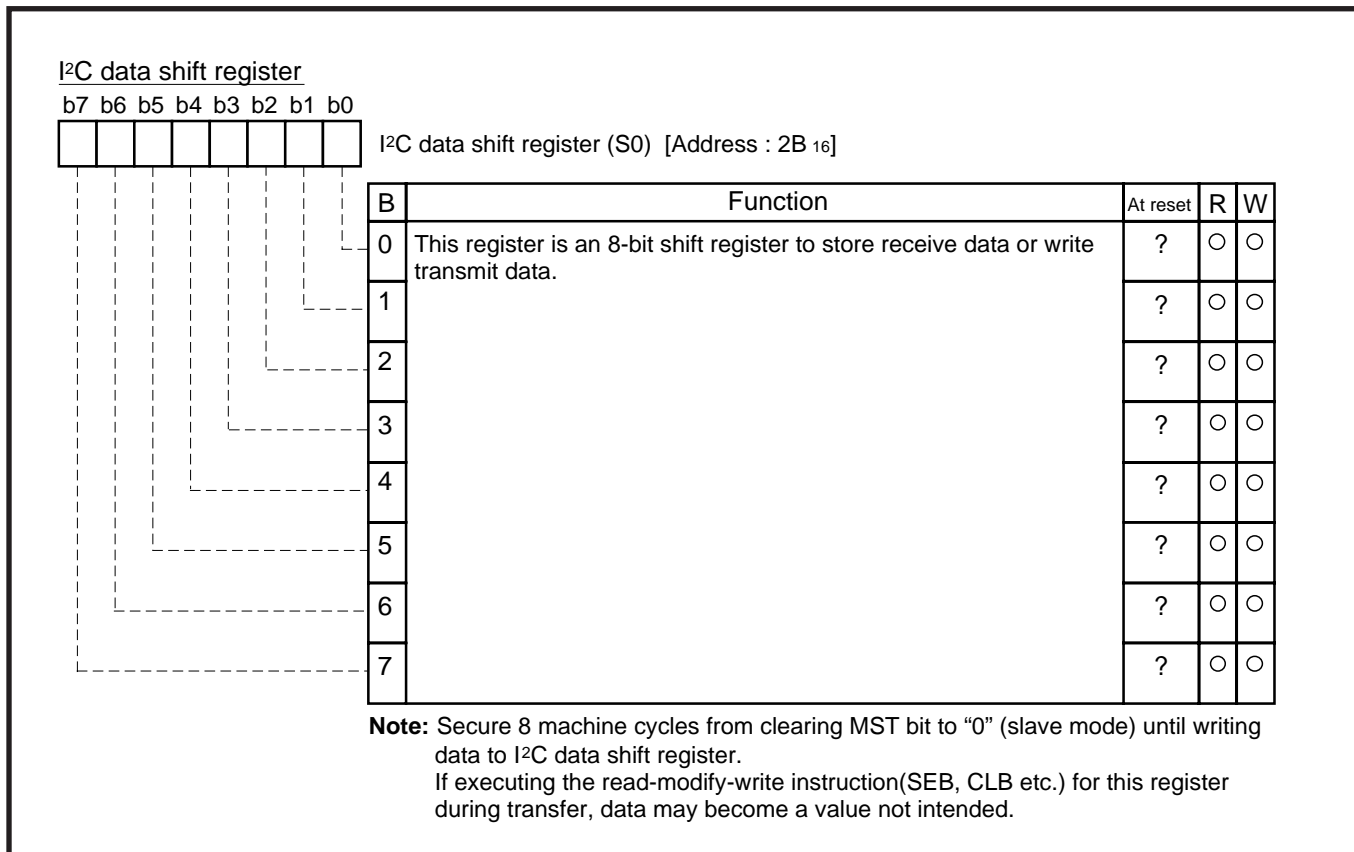


Fig. 2.4.2 Structure of I²C data shift register

APPLICATION

2.4 Multi-master I²C-BUS interface

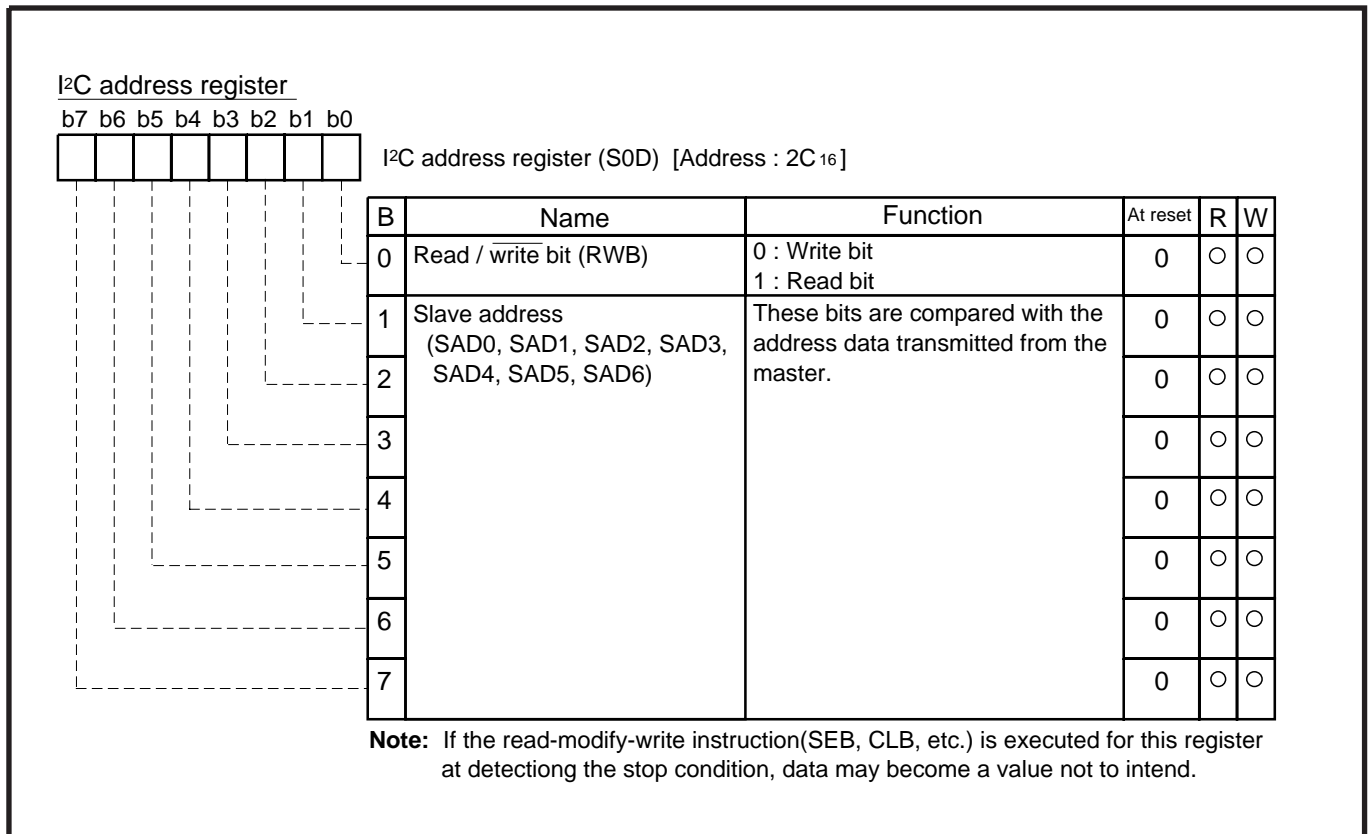


Fig. 2.4.3 Structure of I²C address register

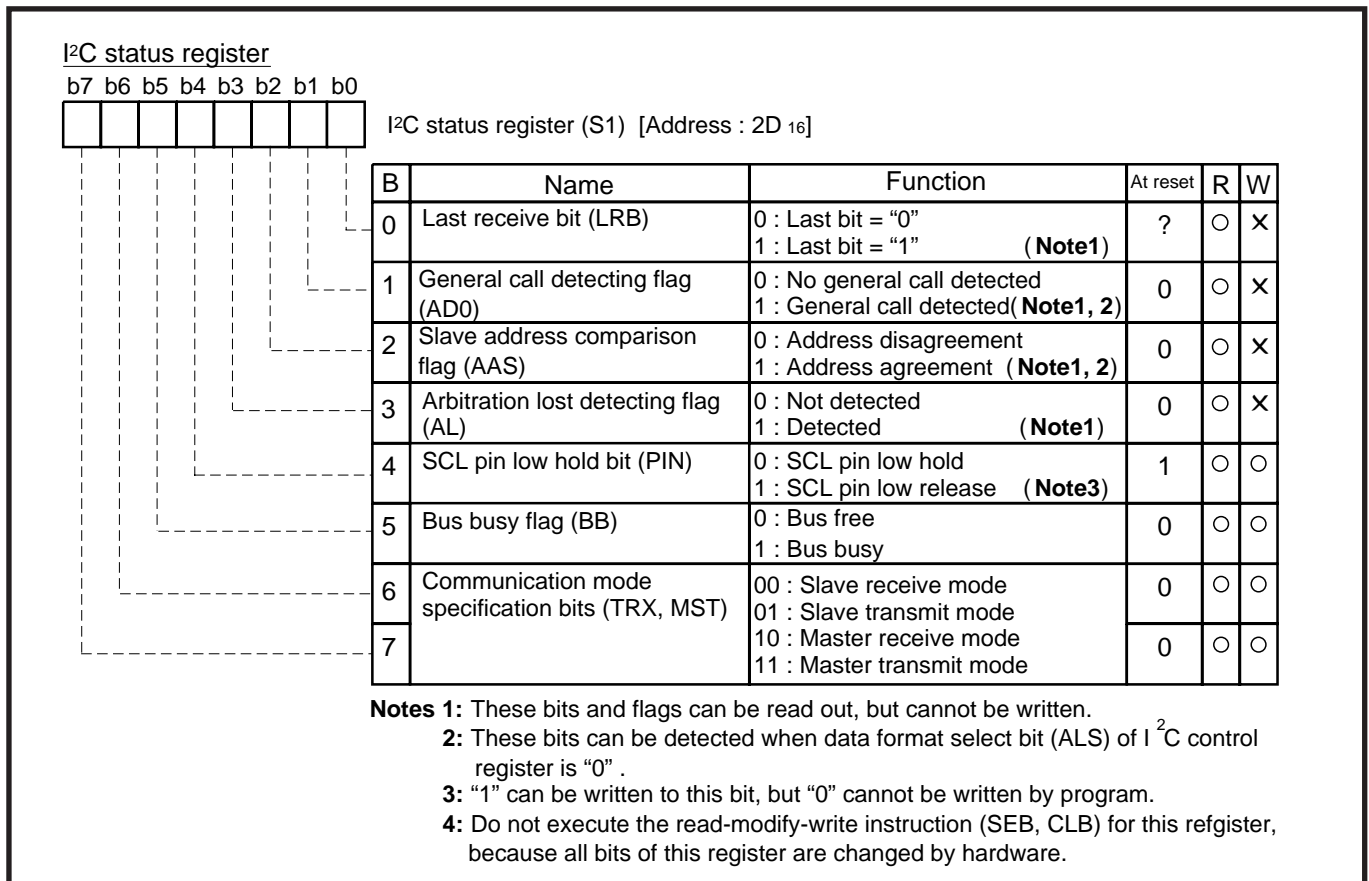


Fig. 2.4.4 Structure of I²C status register

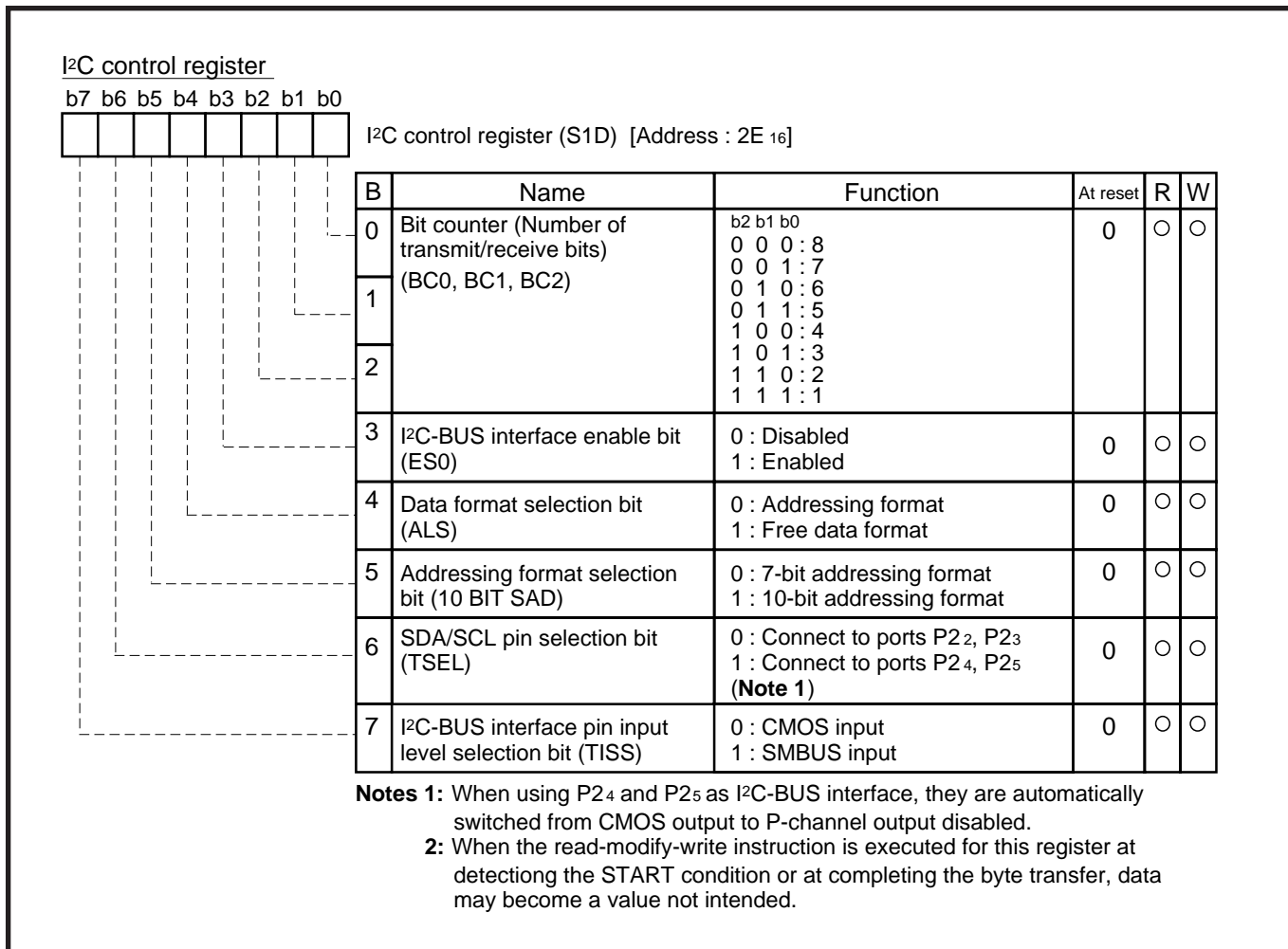


Fig. 2.4.5 Structure of I²C control register

APPLICATION

2.4 Multi-master I²C-BUS interface

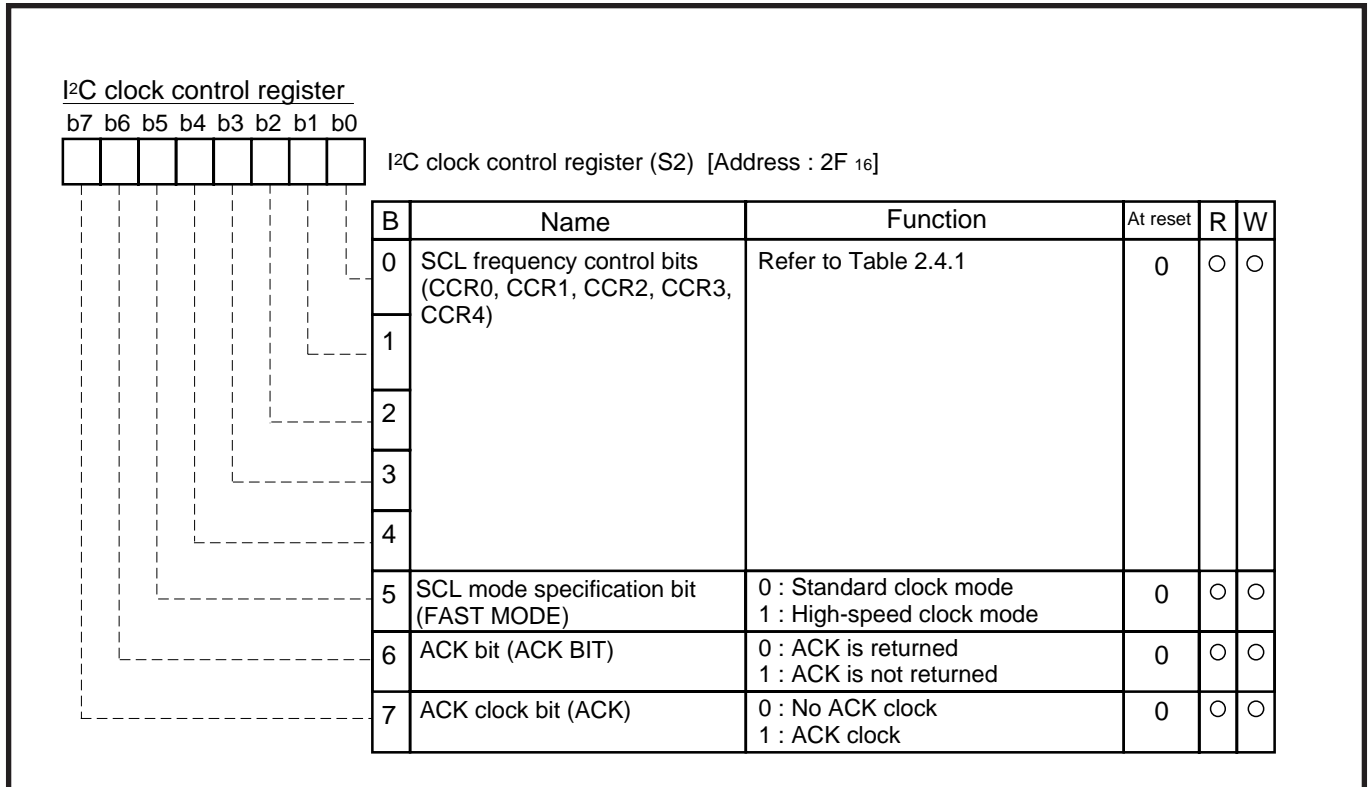


Fig. 2.4.6 Structure of I²C clock control register

Table 2.4.1 Set value of I²C clock control register and SCL frequency

Setting value of CCR4-CCR0					SCL frequency (Note) (at $\phi = 4$ MHz, unit : kHz)	
CCR4	CCR3	CCR2	CCR1	CCR0	Standard clock mode	High-speed clock mode
0	0	0	0	0	Setting disabled	Setting disabled
0	0	0	0	1	Setting disabled	Setting disabled
0	0	0	1	0	Setting disabled	Setting disabled
0	0	0	1	1	– (Note 2)	333
0	0	1	0	0	– (Note 2)	250
0	0	1	0	1	100	400 (Note 3)
0	0	1	1	0	83.3	166
⋮	⋮	⋮	⋮	⋮	500/CCR value (Note 3)	1000/CCR value (Note 3)
1	1	1	0	1	17.2	34.5
1	1	1	1	0	16.6	33.3
1	1	1	1	1	16.1	32.3

Notes 1: Duty of SCL clock output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at $\phi = 4$ MHz). "H" duration of the clock fluctuates from -4 to +2 machine cycles in the standard clock mode, and fluctuates from -2 to +2 machine cycles in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because "L" duration is extended instead of "H" duration reduction.

These are value when SCL clock synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

2: Each value of SCL frequency exceeds the limit at $\phi = 4$ MHz or more. When using these setting value, use ϕ of 4 MHz or less.

3: The data formula of SCL frequency is described below:

$\phi / (8 \times \text{CCR value})$ Standard clock mode

$\phi / (4 \times \text{CCR value})$ High-speed clock mode (CCR value $\neq 5$)

$\phi / (2 \times \text{CCR value})$ High-speed clock mode (CCR value = 5)

Do not set 0 to 2 as CCR value regardless of ϕ frequency.

Set 100 kHz (max.) in the standard clock mode and 400 kHz (max.) in the high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

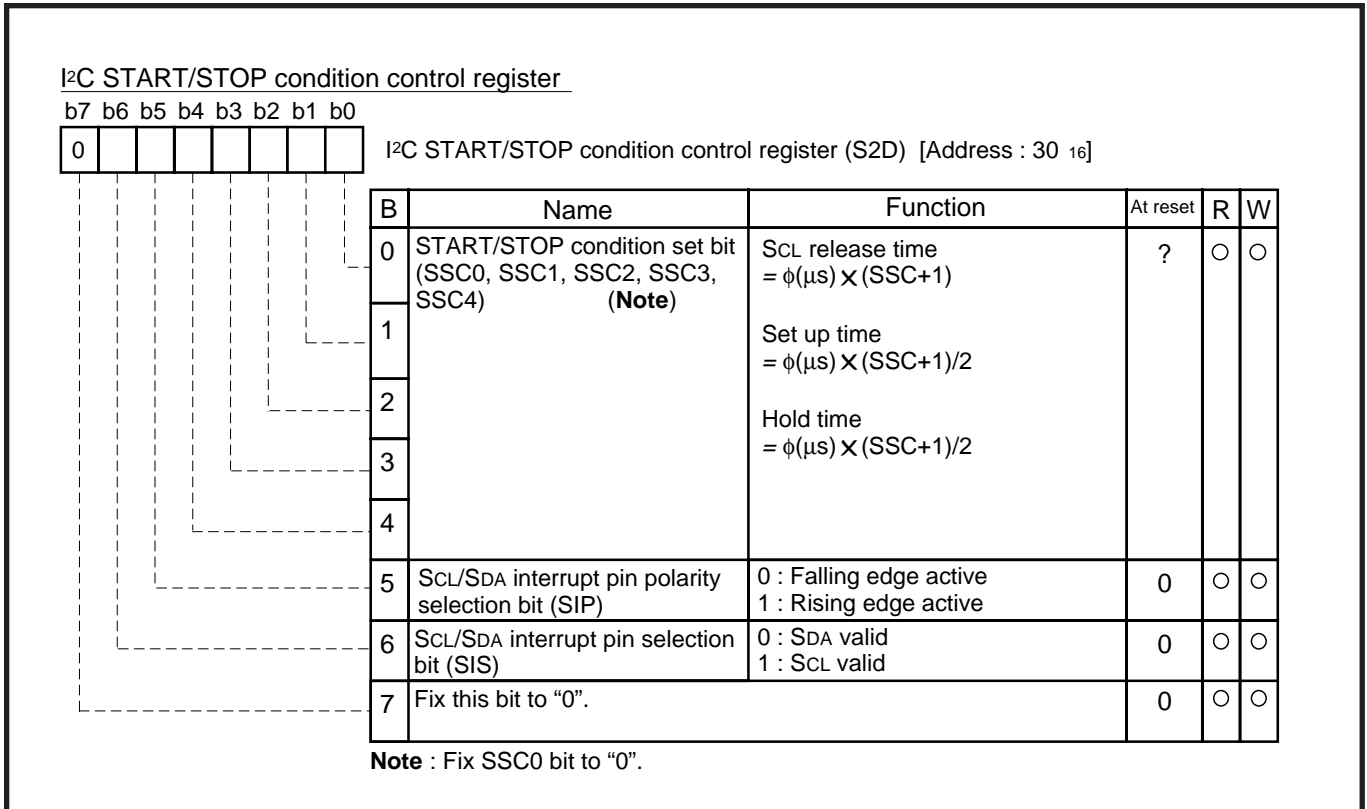


Fig. 2.4.7 Structure of I²C START/STOP condition control register

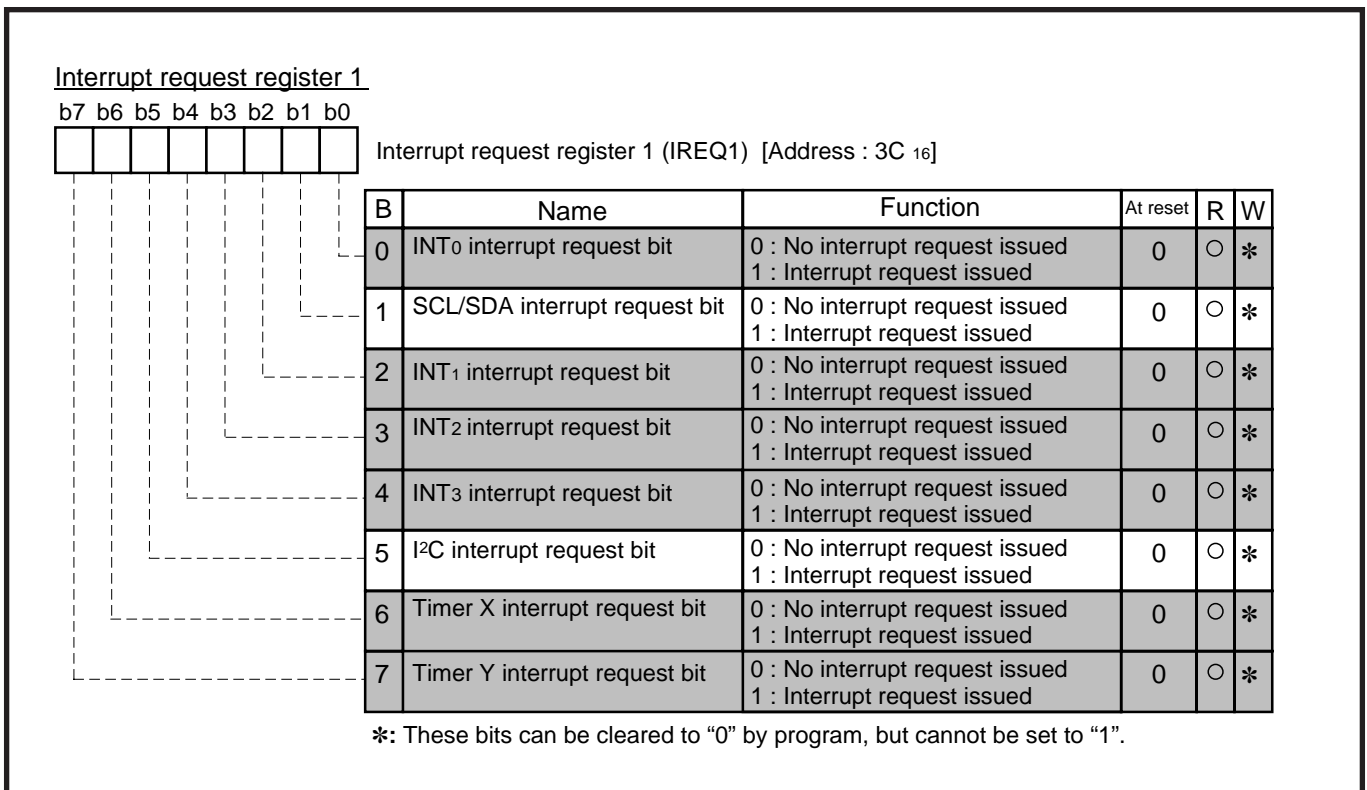


Fig. 2.4.8 Structure of Interrupt request register 1

APPLICATION

2.4 Multi-master I²C-BUS interface

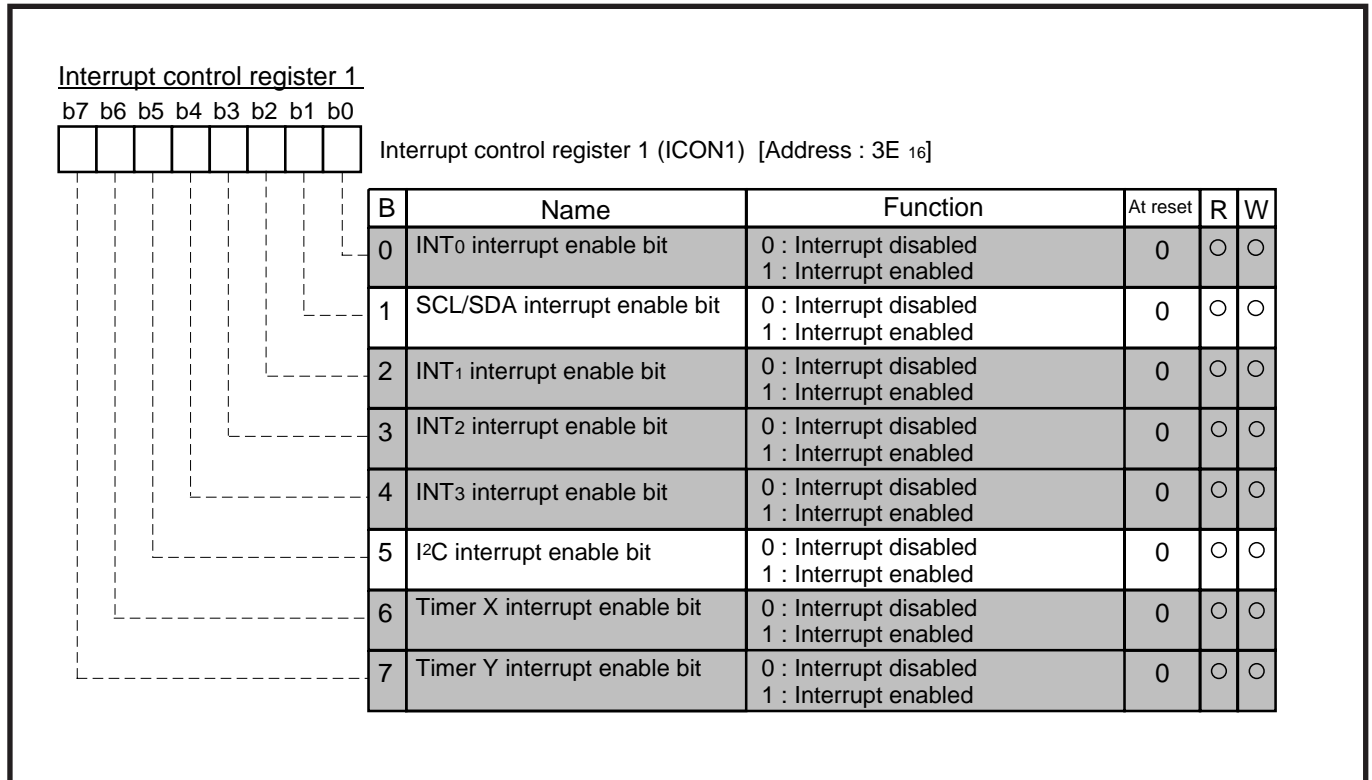


Fig. 2.4.9 Structure of Interrupt control register 1

2.4.3 I²C-BUS overview

The I²C-BUS is a both directions serial bus connected with two signal lines; the SCL which transmits a clock and the SDA which transmits a data.

Each port of the 3851 group has an N-channel open-drain structure for output and a CMOS structure for input. The devices connected with the I²C-BUS interface use an open drain, so that external pull-up resistors are required. Accordingly, while any one of devices always outputs "L", other devices cannot output "H".

Figure 2.4.10 shows the I²C-BUS connection structure.

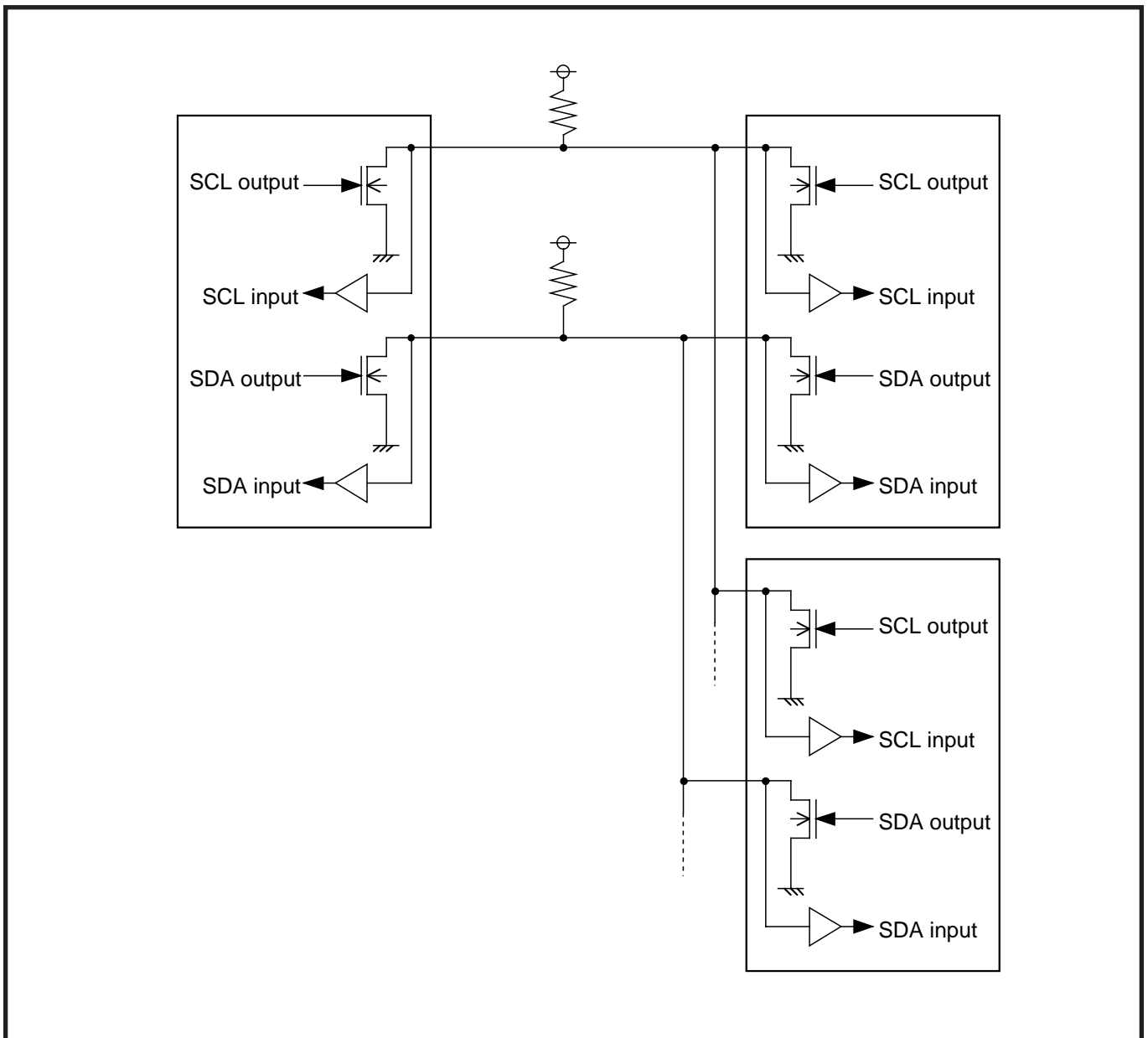


Fig. 2.4.10 I²C-BUS connection structure

APPLICATION

2.4 Multi-master I²C-BUS interface

2.4.4 Communication format

Figure 2.4.11 shows an I²C-BUS communication format example.

The I²C-BUS consists of the following:

- START condition to indicate communication start
- Slave address and data to specify each device
- ACK to indicate acknowledgment of address and data
- STOP condition to indicate communication completion.

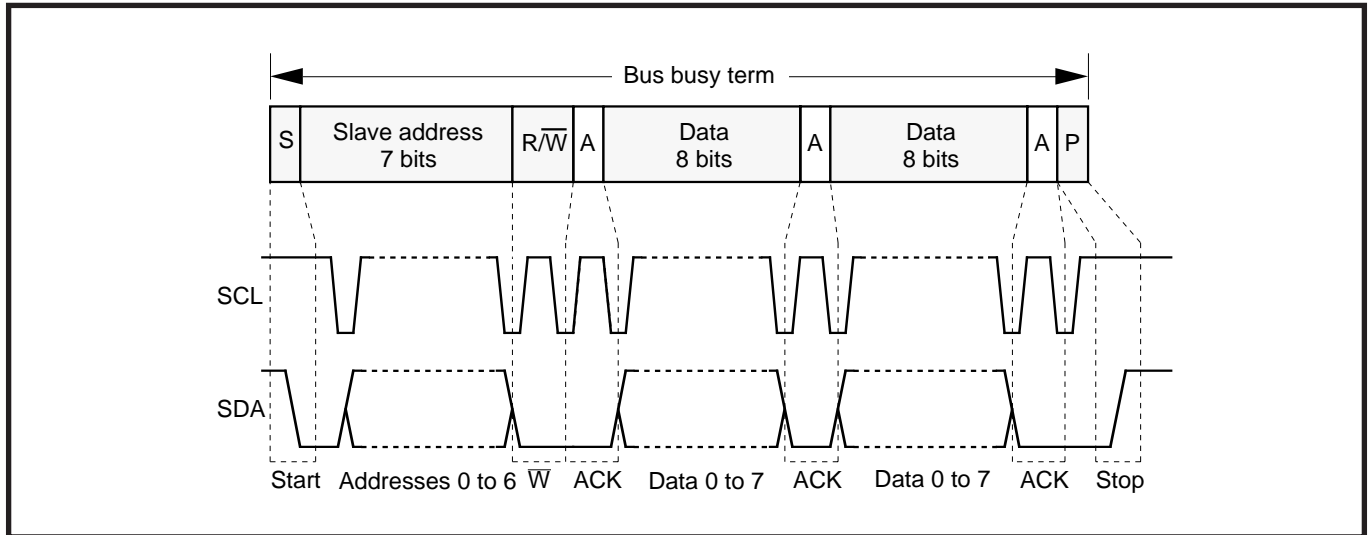


Fig. 2.4.11 I²C-BUS communication format example

(1) START condition

When communication starts, the master device outputs the START condition to the slave device. The I²C-BUS defines that a data can be changed when a clock line is “L”. Accordingly, data change when a clock line is “H” is treated as STOP or START condition.

The data line change from “H” to “L” when a clock line is “H” is START condition.

(2) STOP condition

Just as in START condition, the data line change from “L” to “H” when a clock line is “H” is STOP condition.

The term from START condition to STOP condition is called “Bus busy”. The master device is inhibited from starting data transfer during that term.

The Bus busy status can be judged by using the BB flag of I²C status register (bit 5 of address 002D₁₆).

(3) Slave address

The slave address is transmitted after START condition. This address consists of 7 bits and the 7-th bit functions as the read/write (R/W) bit which indicates a data transmission method. The slave devices connected with the same I²C-BUS must have their addresses, individually. It is because that address is defined for the master to specify the transmitted/received slave device.

The read/write (R/W) bit indicates a data transmission direction; “L” means write from the master to the slave, and “H” means read in.

(4) Data

The data has an 8-bit length. There are two cases depending on the read/write (R/W) bit of a slave address; one is from the master to the slave and the other is from the slave to the master.

(5) ACK bit

The ACK bit clock is generated by the master. This is used for indication of acknowledgment on the SDA line, the slave's busy and the data end.

For example, the slave device makes the SDA line "L" for acknowledgment when confirming the slave address following the START condition. The built-in I²C-BUS interface has the slave address automatic judgment function and the ACK acknowledgment function. "L" is automatically output when the ACK bit of I²C clock control register (bit 6 of address 002F₁₆) is "0" and an address data is received. When the slave address and the address data do not correspond, "H" (NACK) is automatically output.

In case the slave device cannot receive owing to an interrupt process, performing operation or others, the master can output STOP condition and complete data transfer by making the ACK data of the slave address "H" for acknowledgment. Even in case the slave device cannot receive a data during data transferring, the communication can be interrupted by performing NACK acknowledgment to the following data.

When the master is receiving the data from the slave, the master can notify the slave of completion of data reception by performing NACK acknowledgment to the last data received from the slave.

(5) RESTART condition

The master can receive or transmit data without transmission of STOP condition while the master is transmitting or receiving a data.

For example, after the master transmitted a data to the slave, transmitting a slave address + R (Read) following RESTART condition can make the following data treat as a reception data.

Additionally, transmitting a slave address + \overline{W} (Write) following RESTART condition can make the following data treat as a transmission data.

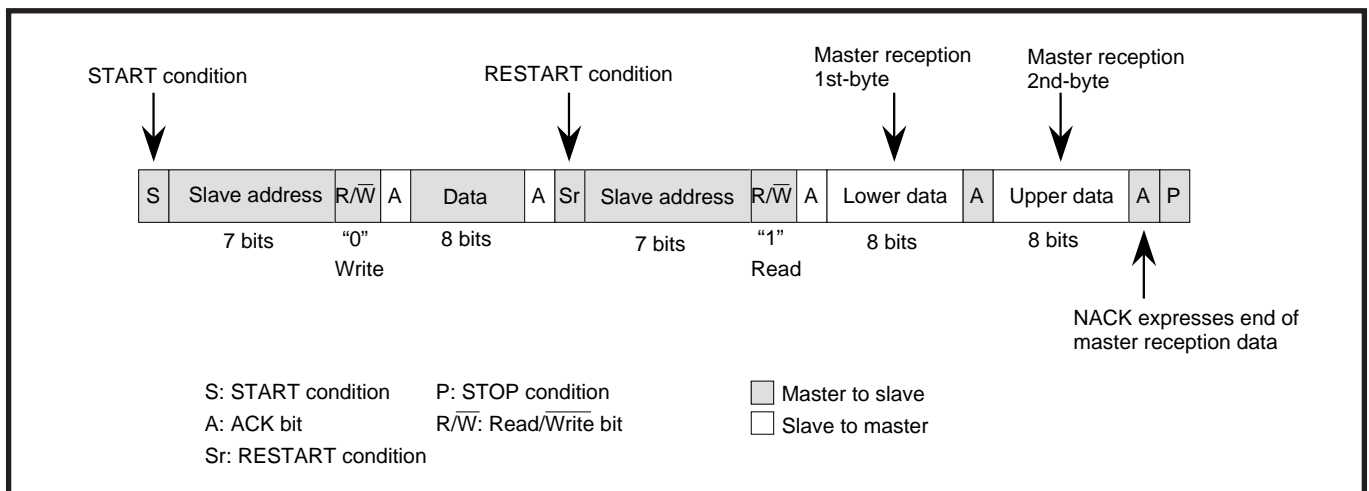


Fig. 2.4.12 RESTART condition of master reception

2.4.5 Synchronization and Arbitration lost

(1) Synchronization

When a plural master exists on the I²C-BUS and the masters, which have different speed, are going to simultaneously communicate; there is a rule to unify clocks so that a clock of each bit can be output correctly.

Figure 2.4.13 shows a synchronized SCL line example. The SCL (A) and the SCL (B) are the master devices having a different speed. The SCL is synchronized waveforms.

As shown by Figure 2.4.13, the SCL lines can be synchronized by the following method; the device which first finishes "H" term makes the SCL line "L" and the device which last remains "L" makes the SCL line "H".

APPLICATION

2.4 Multi-master I²C-BUS interface

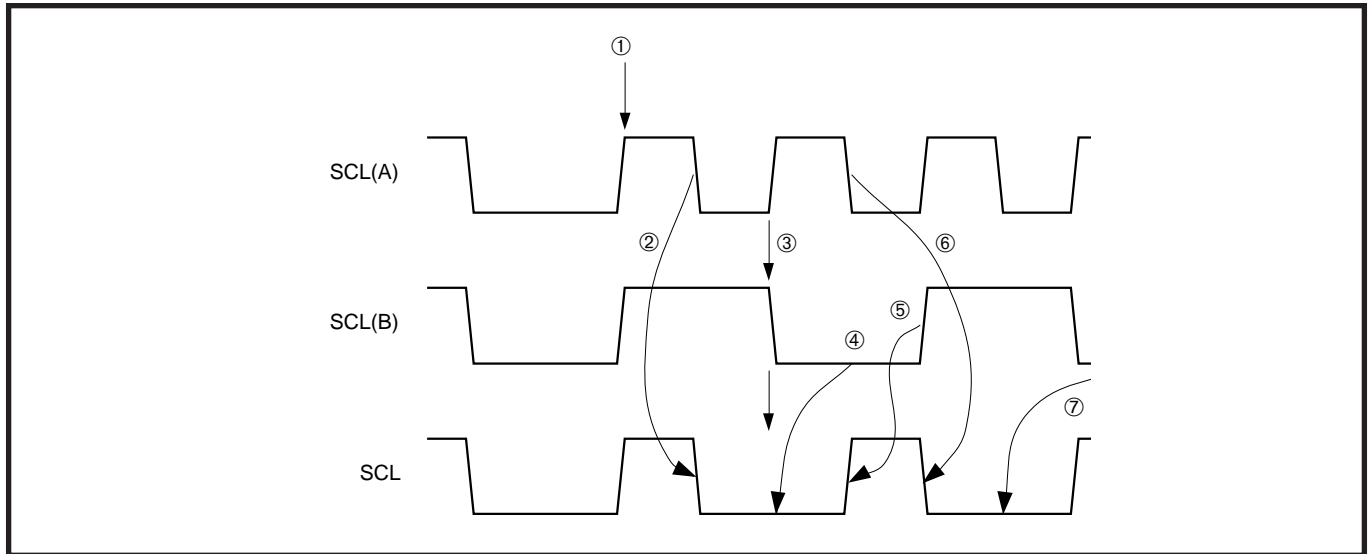


Fig. 2.4.13 SCL waveforms when synchronizing clocks

- ① After START condition, the masters, which have different speed, simultaneously start clock transmission.
- ② The SCL outputs “L” because (A) finished counting “H” output; then (B)’s “H” output counting is interrupted and (B) starts counting “L” output.
- ③ The (A) outputs “H” because (A) finished counting “L” term; the SCL level does not become “H” because (B) outputs “L”, and counting “H” term does not start but stop.
- ④ (B) outputs “L” term.
- ⑤ The SCL outputs “H” because (B) finished counting “L” term; then (B)’s “H” output counting is started at the same time as (A).
- ⑥ The SCL outputs “L” because (A) first finished counting “H” output; then (B)’s “H” output counting is interrupted and (B) starts counting “L” output.
- ⑦ The above are repeatedly performed.

(2) Clock synchronization during communication

In the I²C-BUS, the slave device is permitted to retain the SCL line “L” and become waiting status for transmission from the master. By byte unit, for the reception preparation of the slave device, the master can become waiting status by making the SCL line “L”, which is after completion of byte reception or the ACK.

By bit unit, it is possible to slow down a clock speed by retaining the SCL line “L” for slave devices having limited hardware.

The 3851 group can transmit data correctly without reduction of data bits toward waiting status request from the slave device. It is because the synchronization circuit is included for the case when retaining the SCL line “L” as an internal hardware.

After the last bit, including the ACK bit, of a transmission/reception data byte, the SCL line automatically remains “L” and waiting status is generated until completion of an interrupt process or reception preparation.

(3) Arbitration lost

A plural master exists on the same bus in the I²C-BUS and there are possibility to start communication simultaneously. Even when the master devices having the same transmission frequency start communication simultaneously, which device must transmit data correctly. Accordingly, there is the definition to detect a communication confliction on the SDA line in the I²C-BUS.

The SDA line is output at the timing synchronized by the SCL, however, the synchronization among the SDA signals is not performed.

2.4.6 SMBUS communication usage example

This clause explains a SMBUS communication control example using the I²C-BUS. This is a control example as the master device and the slave device in the Read Word protocol of SMBUS protocol.

The following is a communication example of the "Voltage () command" of the Smart battery data.

Communication specifications:

- Communication frequency = 100 kHz
- Slave address of itself, battery, = "0001011X₂" (X means the read/ $\overline{\text{write}}$ bit)
- Slave address of communication destination, host, = "0001000X₂" (X means the read/ $\overline{\text{write}}$ bit)
- Voltage () command = "09₁₆"
- Voltage value of acknowledgment = "2EE0₁₆"; 12000 mV)
- The communication process is performed in the interrupt process. However, the main process performs an occurrence of the first START condition and a slave address set.
- A communication buffer is established. Data transfer between the main process and the interrupt process is performed through the communication buffer.

(1) Initial setting

Figure 2.4.14 shows an initial setting example using SMBUS communication.

APPLICATION

2.4 Multi-master I²C-BUS interface

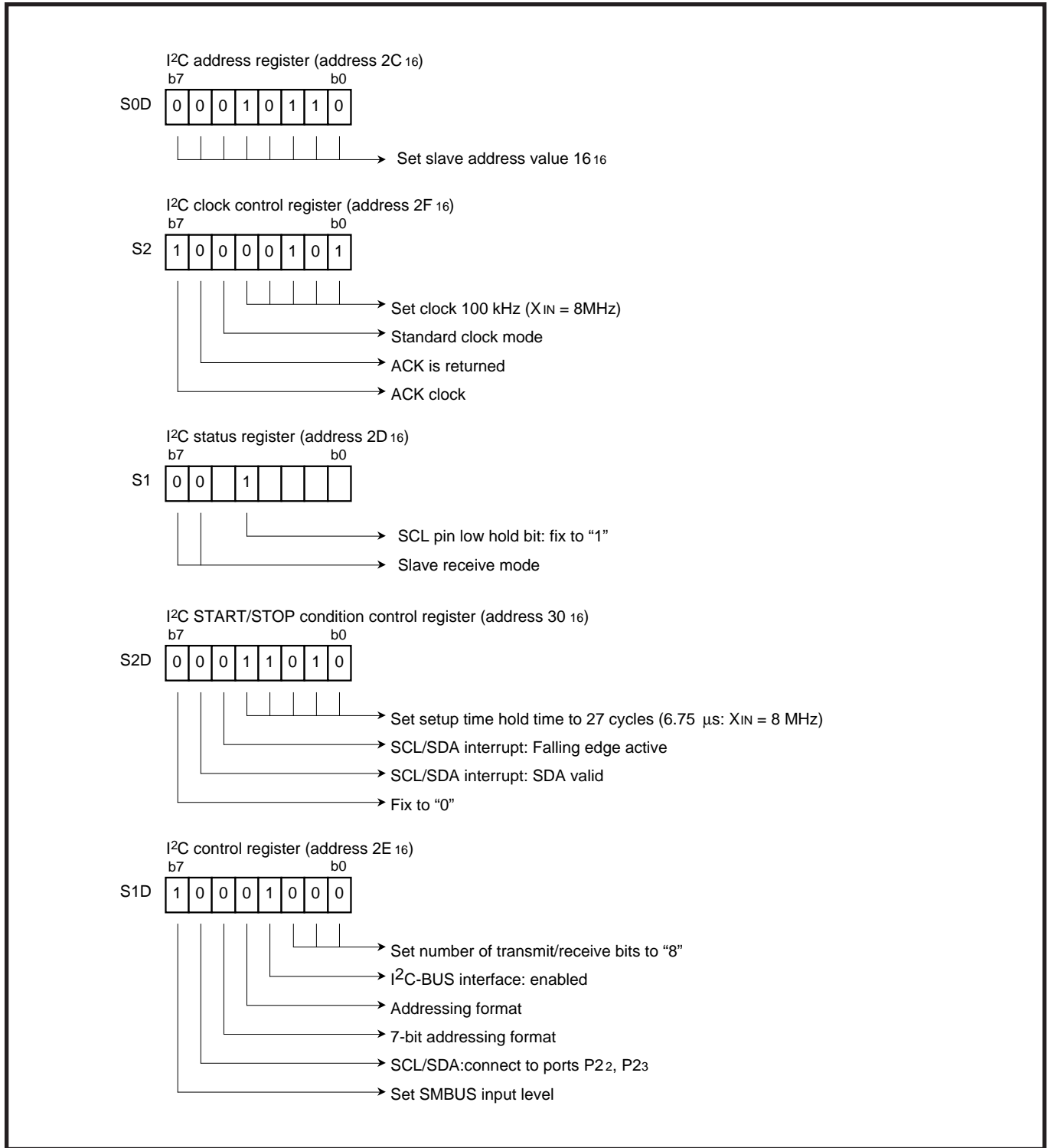


Fig. 2.4.14 Initial setting example using SMBUS communication

(2) Communication example in master device

The master device follows the procedures ① to ⑥ shown by Figure 2.4.15.

Additionally, the shaded area in the figure is a transmission data from the master device and the white area is a transmission data from the slave device.

- ① Generating of START condition; Transmission of slave address + write bit
- ② Transmission of command
- ③ Generating of RESTART condition; Transmission of slave address + read bit
- ④ Reception of lower data
- ⑤ Reception of upper data
- ⑥ Generating of STOP condition

Figures 2.4.16 to 2.4.21 show the procedures ① to ⑥.

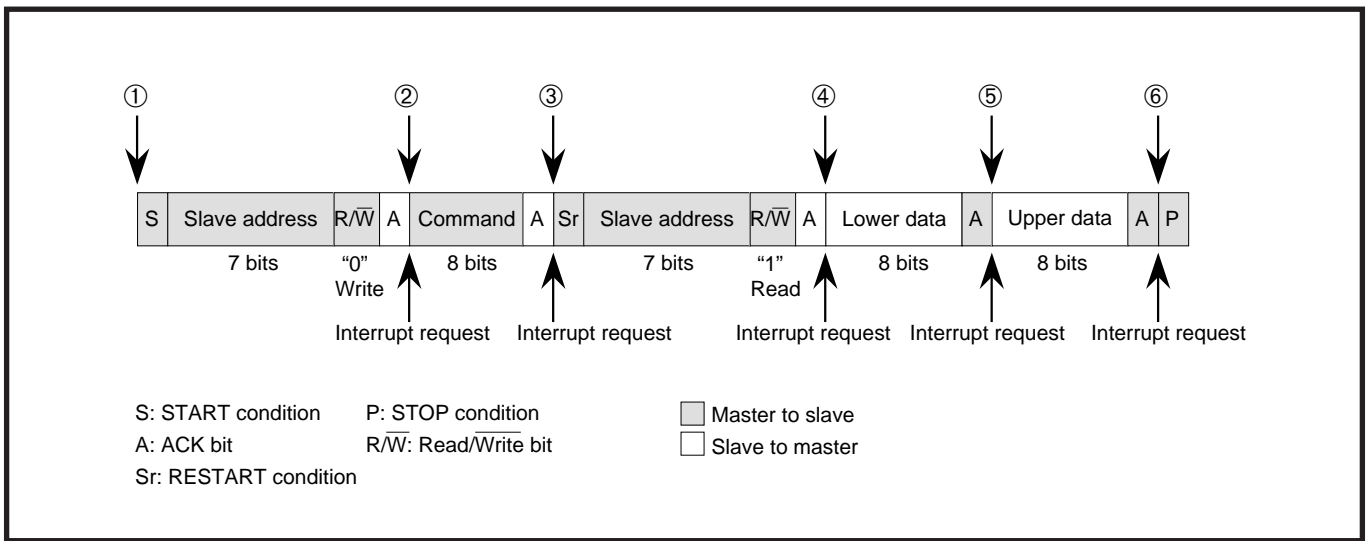


Fig. 2.4.15 Read Word protocol communication as SMBUS master device

APPLICATION

2.4 Multi-master I²C-BUS interface

① Generating of START condition; Transmission of slave address + write bit

After confirming that other master devices do not use the bus, generate the START condition, because the SMBUS is a multi-master.

Write "slave address + write bit" to the I²C data shift register (address 002B₁₆) before performing to make the START condition generate. It is because the SCL of 1-byte unit is output, following occurrence of the START condition.

If other master devices start communication until an occurrence of the START condition after confirming the bus use, it cannot communicate correctly. However in this case, that situation does not affect other master devices owing to detection of an arbitration lost or the START condition duplication preventing function.

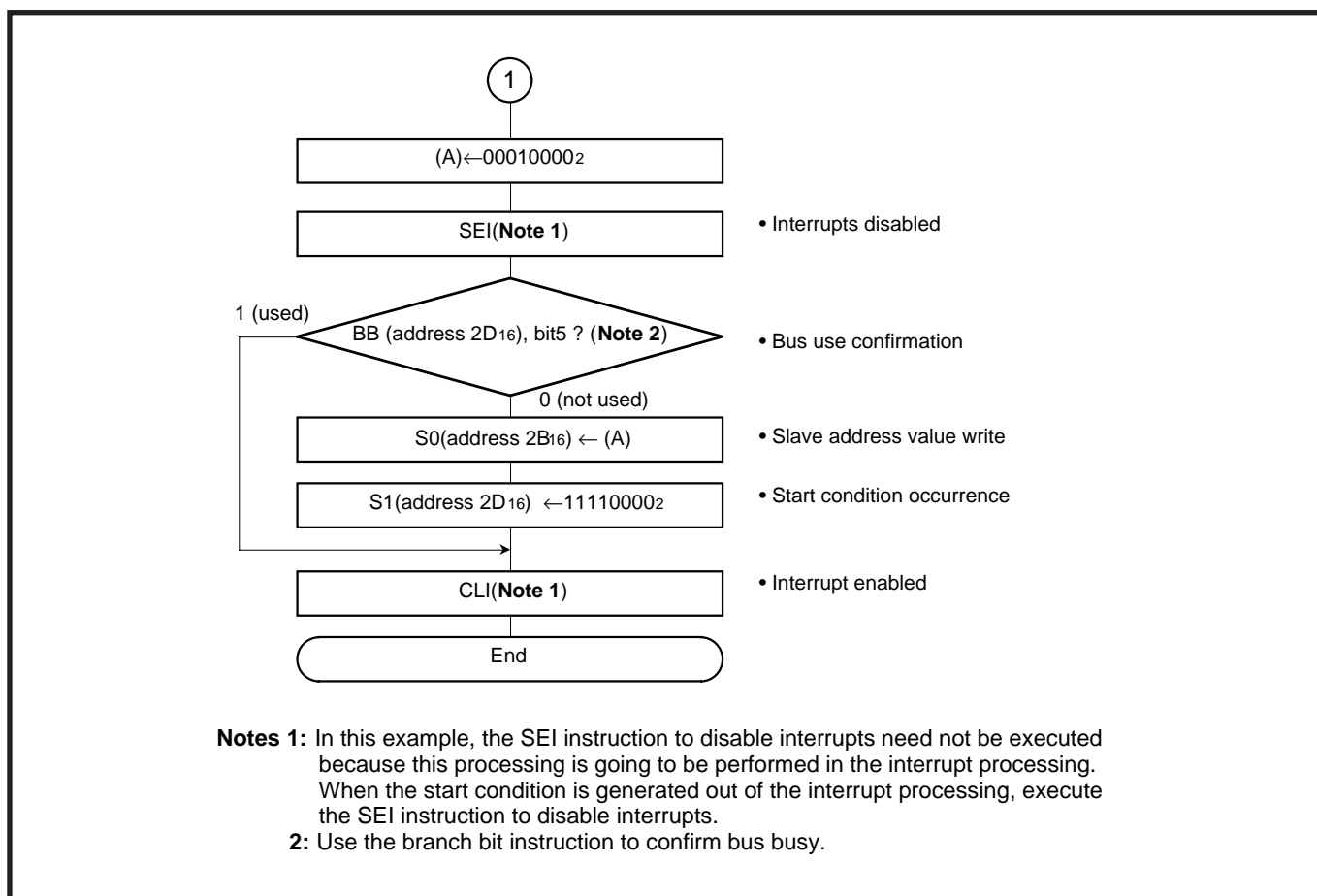


Fig. 2.4.16 Transmission process of START condition and slave address

② Transmission of command

Confirm correct completion of communication at ① before command transmission. When receiving the STOP condition, a process not to transmit a command is required, because the internal I²C-BUS generates an interrupt request also owing to the STOP condition transmitted to other devices.

After confirming correct completion of communication, write a command to the I²C data shift register (address 002B₁₆).

In case the AL bit (bit 3 of address 002D₁₆) is "1", check the slave address comparison flag (ASS bit; bit 2 of address 002D₁₆) to judge whether the device given a right of master transmission owing to an arbitration specifies itself as a slave address. When it is "1", perform the slave reception; when "0", wait for a STOP condition occurrence caused by other devices and the communication completion.

In case the AL bit is "0", check the last received bit (LRB bit; bit 0 of address 002D₁₆). When it is "1", make the STOP condition generate and release the bus use, because the specified slave device does not exist on the SMBUS.

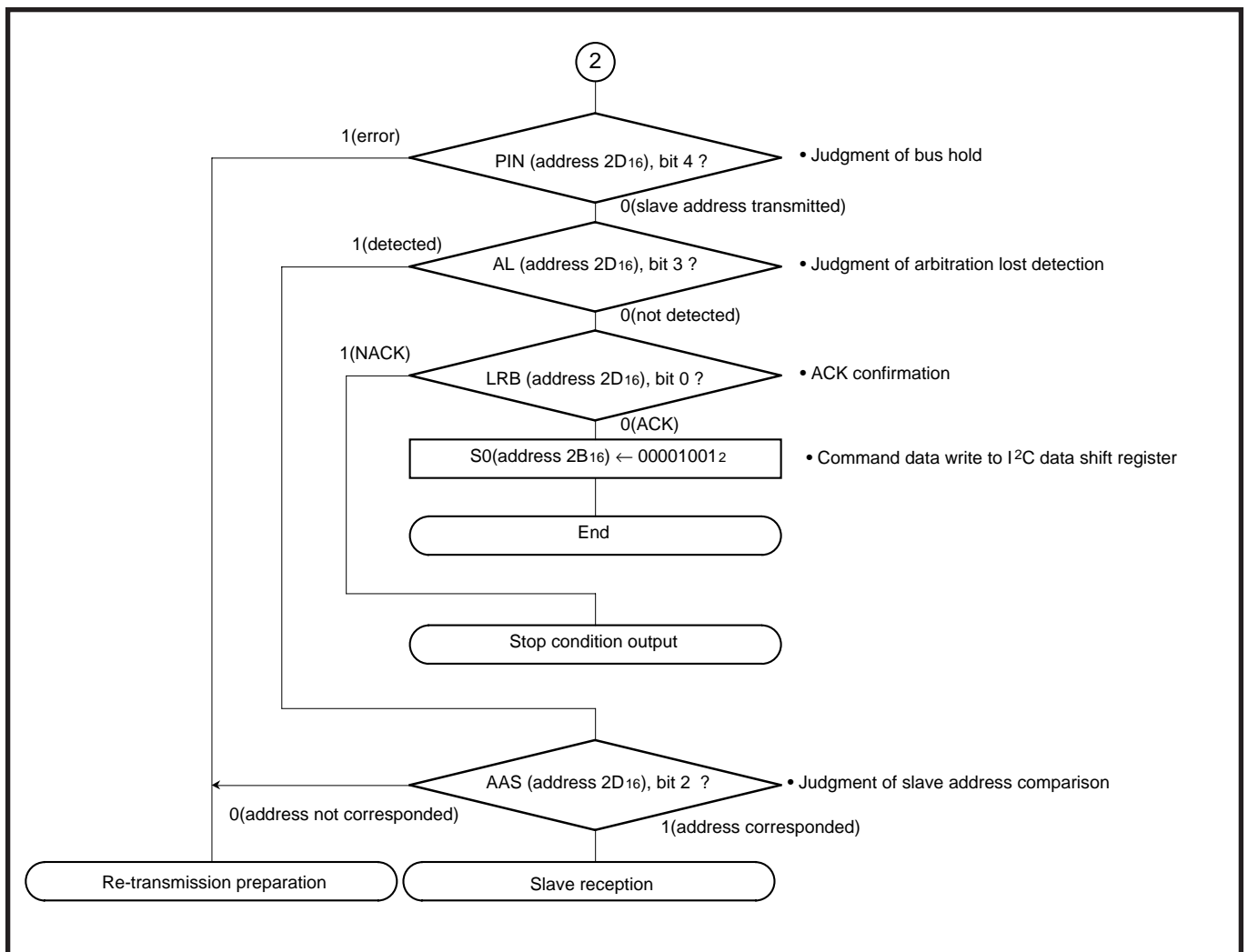


Fig. 2.4.17 Transmission process of command

APPLICATION

2.4 Multi-master I²C-BUS interface

③ Generating of RESTART condition; Transmission of slave address + read bit

Confirm correct completion of communication at ② before generating the RESTART condition. After confirming correct completion, generate the RESTART condition and perform the transmission process of “slave address + read bit”. Note that procedure because that is different from ①’s process.

As the same reason as ①, write “slave address + read bit” to the I²C data shift register (address 002B₁₆) before performing to make the START condition generate. However, when writing a slave address to the I²C data shift register in this condition, a slave address is output at that time. Consequently, the RESTART condition cannot be generated. Therefore, follow the slave reception procedure before those processes.

In case the arbitration lost detecting flag (AL bit, bit 3 of address 002D₁₆) is “1”, return to the process ①, because other master devices will have priority to communicate.

When the last received bit (LRB bit; bit 0 of address 002D₁₆) is “1”, generate the STOP condition and make the bus release, because acknowledgment cannot be done owing to BUSY status of the slave device specified on the SMBUS or other reasons.

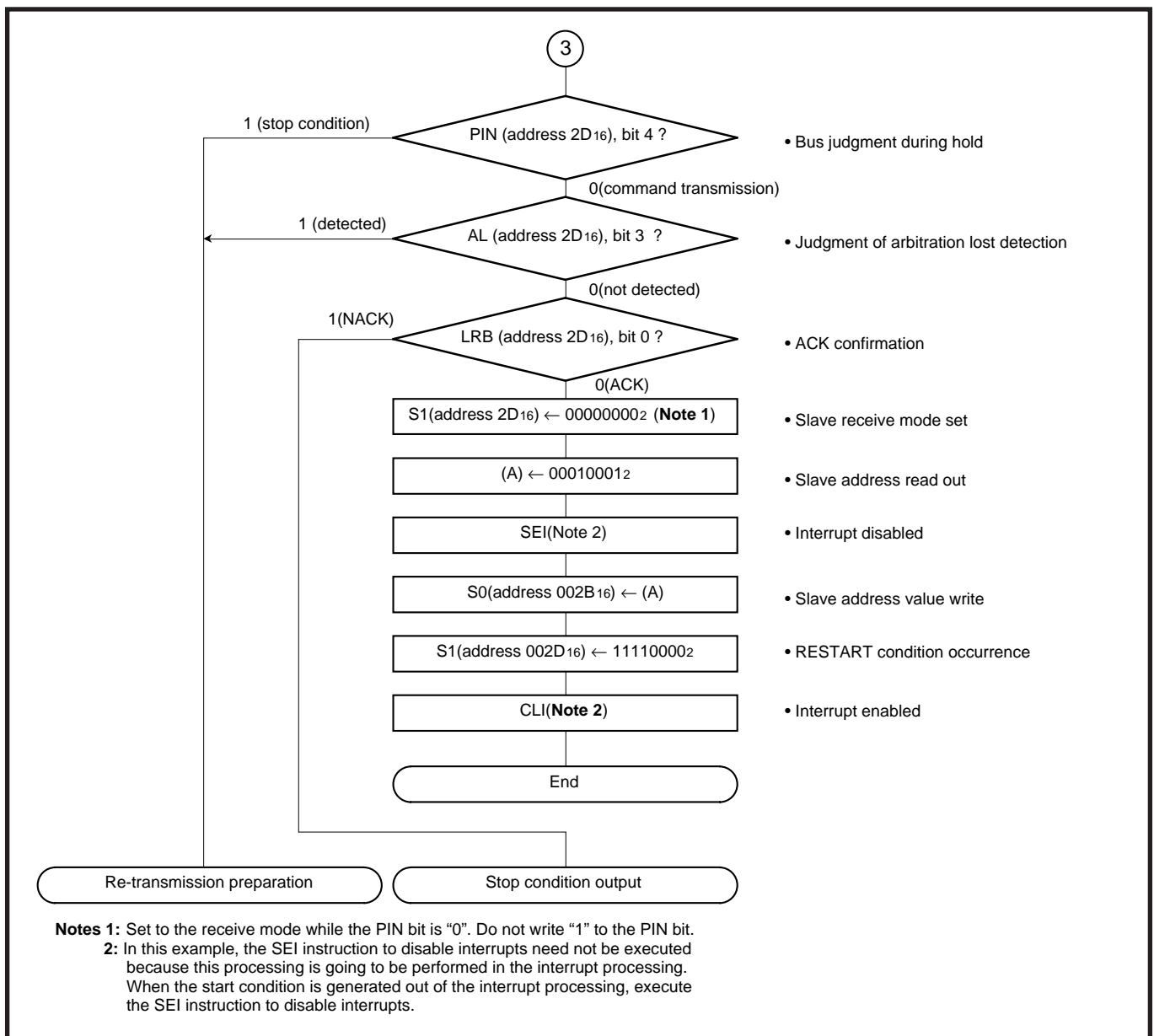


Fig. 2.4.18 Transmission process of RESTART condition and slave address + read bit

④ Reception of lower data

Confirm correct completion of communication at ③ before receiving the lower data. After confirming correct completion, clear the ACK bit (bit 6 of address 002F₁₆) to "0", in which ACK is returned and set to the master receive mode. After that, write a dummy data to the I²C data shift register (address 002B₁₆).

When the MST bit (bit 7 of address 002D₁₆) is "0", perform the error process explained as follows and return to the process ①.

When the last received bit (LRB bit; bit 0 of address 002D₁₆) is "1", generate the STOP condition and make the bus release, because the slave device specified on the SMBUS does not exist.

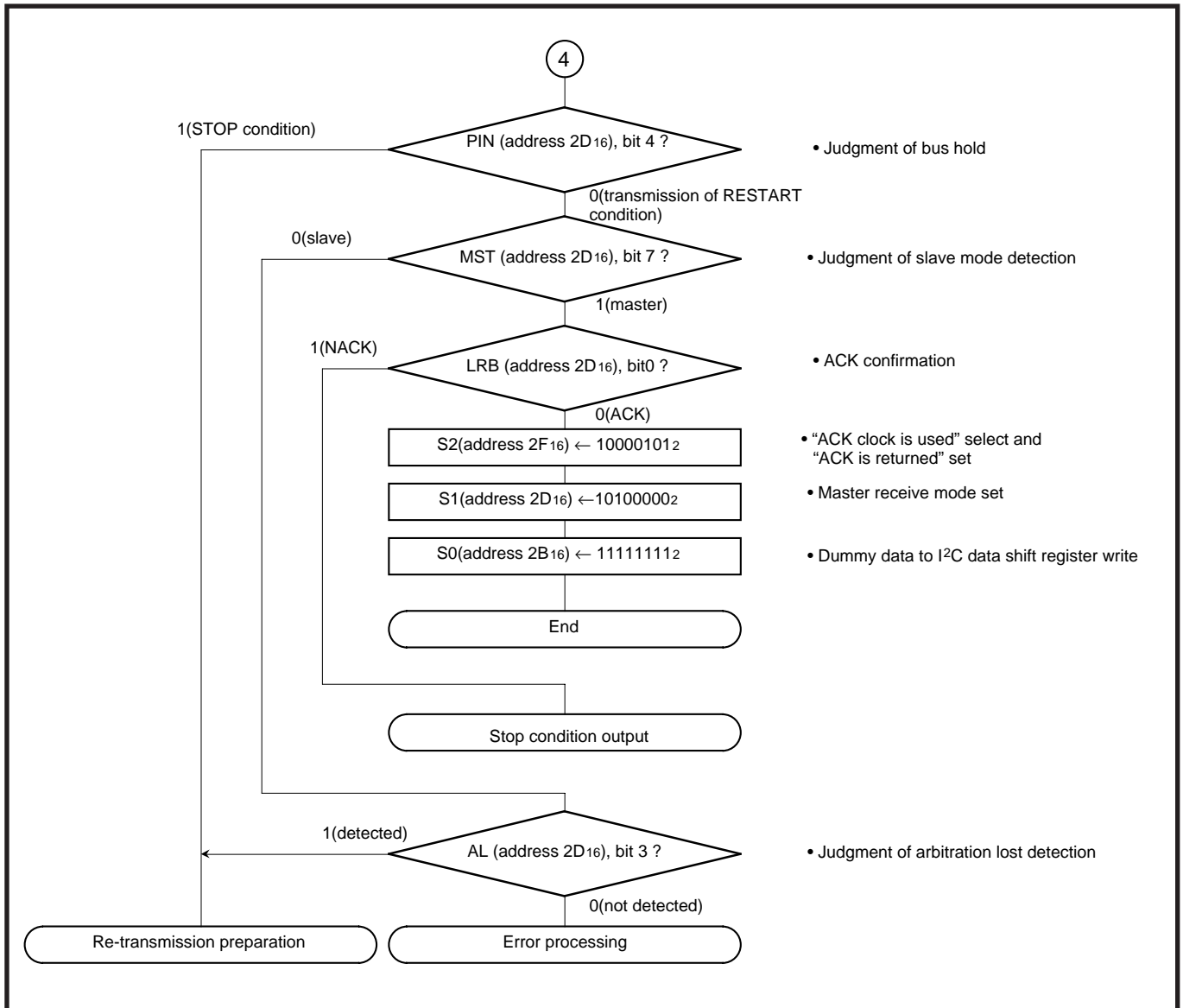


Fig. 2.4.19 Reception process of lower data

APPLICATION

2.4 Multi-master I²C-BUS interface

⑤ Transmission of upper data

Confirm correct completion of communication at ④ before receiving the upper data. After confirming correct completion, store the received data (lower data).

Set the ACK bit (bit 6 of address 002F₁₆) to "1", in which ACK is not returned, write a dummy data to the I²C data shift register (address 002B₁₆).

When the MST bit (bit 7 of address 002D₁₆) is "0", return to the process ①, because other devices have priority to communicate.

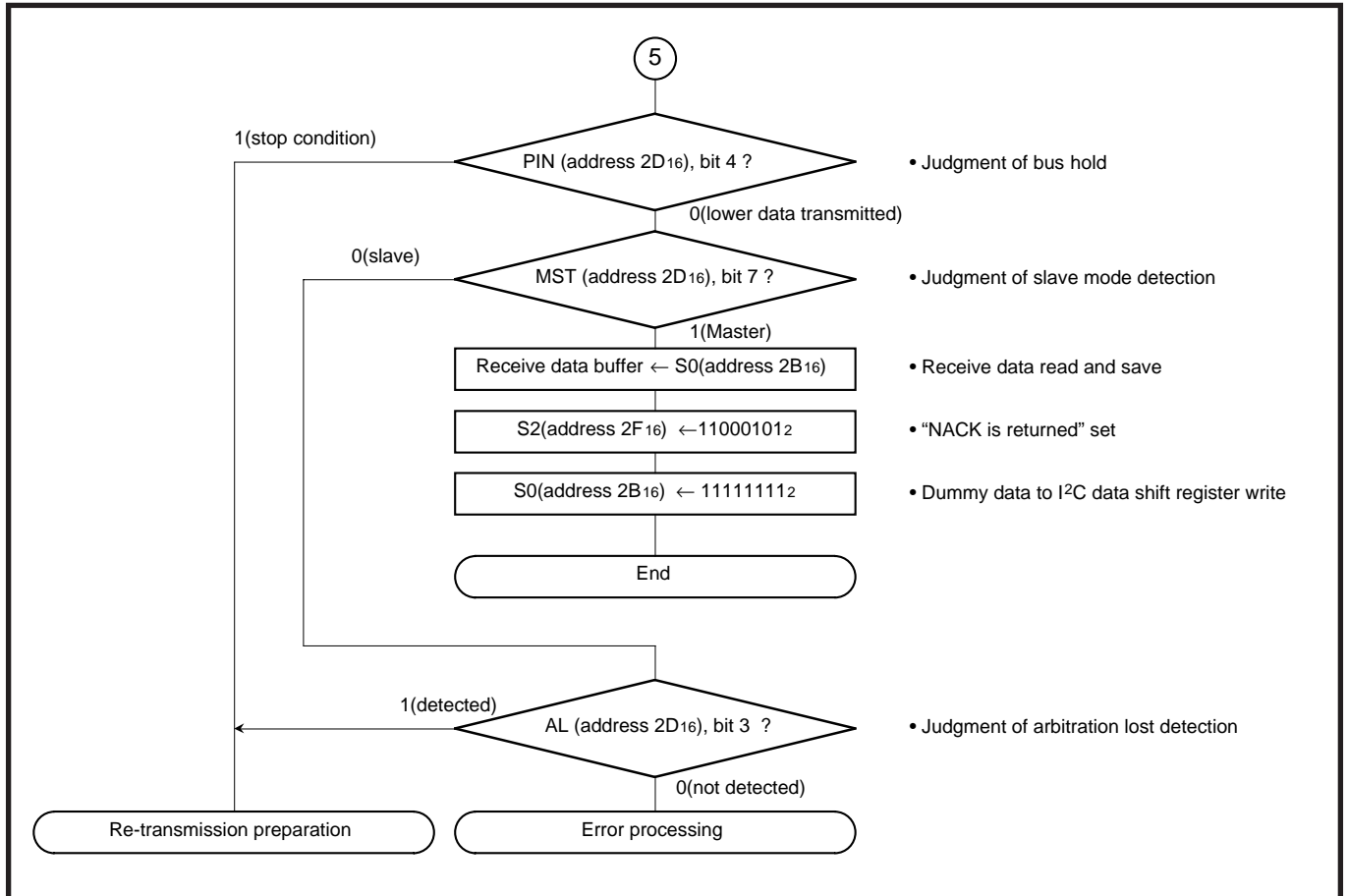


Fig. 2.4.20 Reception process of upper data

⑥ Generating of STOP condition

Confirm correct completion of communication at ⑤ before generating the STOP condition. After confirming correct completion, store the received data (upper data).

Clear the ACK bit (bit 6 of address 002F₁₆) to "0", in which ACK is returned, generate the STOP condition. The communication mode is set to the slave receive mode by the occurrence of STOP condition.

When the MST bit (bit 7 of address 002D₁₆) is "0", return to the process ①, because other devices have priority to communicate.

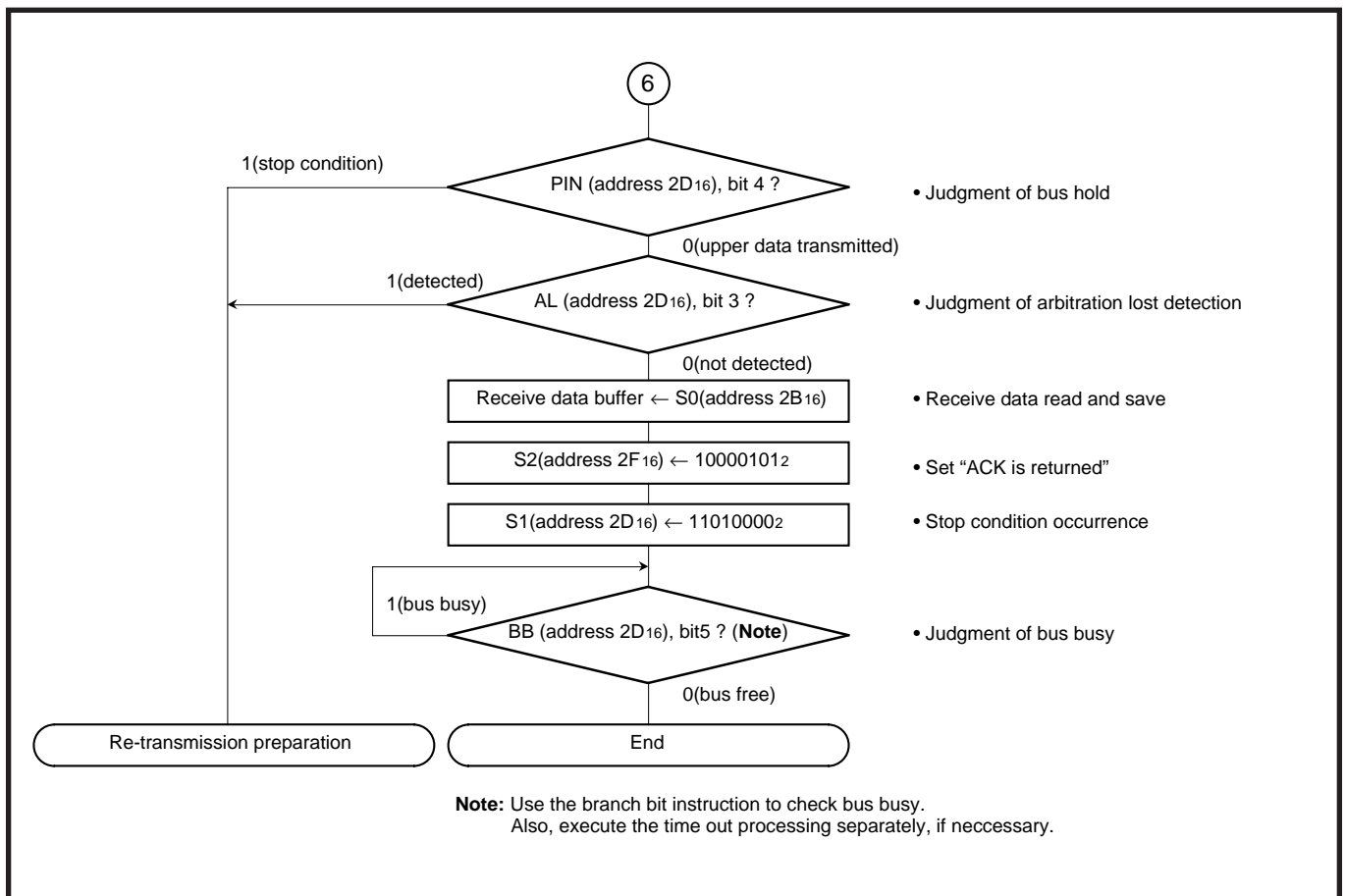


Fig. 2.4.21 Generating of STOP condition

APPLICATION

2.4 Multi-master I²C-BUS interface

(3) Communication example in slave device

The slave device follows the procedures ① to ⑥ shown by Figure 2.4.22.

The only difference from the master device's communication is an occurrence of interrupt request after detection of STOP condition.

- ① Reception of START condition; Transmission of ACK bit due to slave address correspondence
- ② Reception of command
- ③ Reception of RESTART condition; Reception of slave address + read bit
- ④ Transmission of lower data
- ⑤ Transmission of upper data
- ⑥ Reception of STOP condition

Figures 2.4.23 to 2.4.28 show the procedures ① to ⑥.

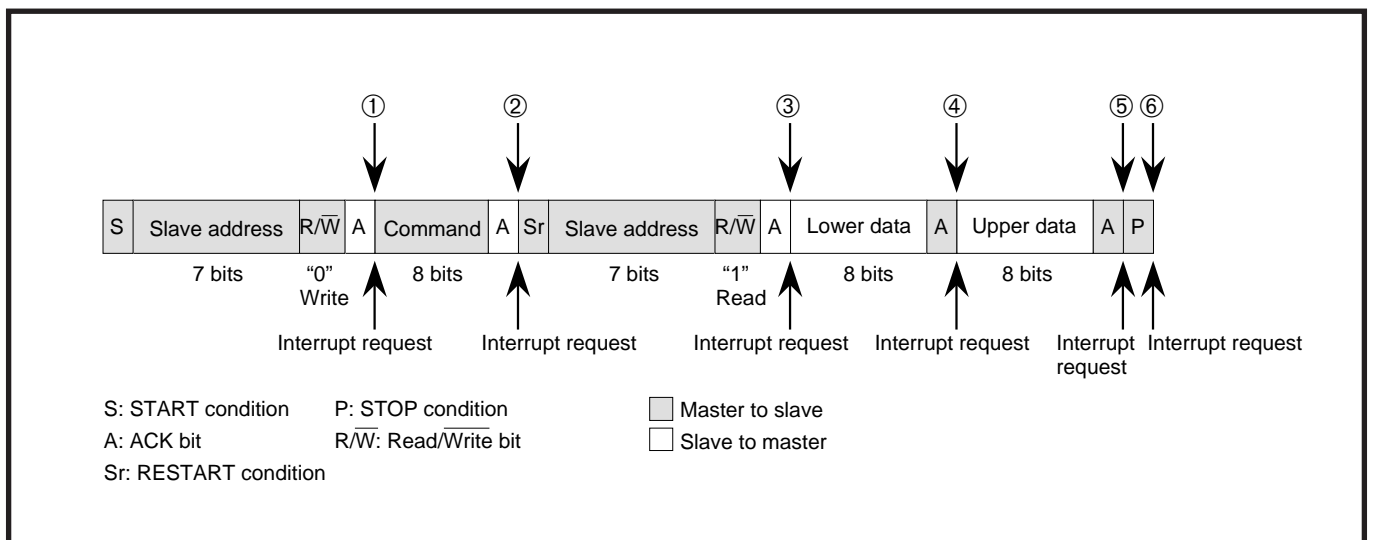


Fig. 2.4.22 Communication example as SMBUS slave device

① Reception of START condition; Transmission of ACK bit due to slave address correspondence

In the case of operation as the slave, all processes are performed in the interrupt after setting of the slave reception in the main process, because an interrupt request does not occur until correspondence of a slave address.

In the first interrupt, after confirming correspondence of the slave address, write a dummy data to receive a command into the I²C data shift register.

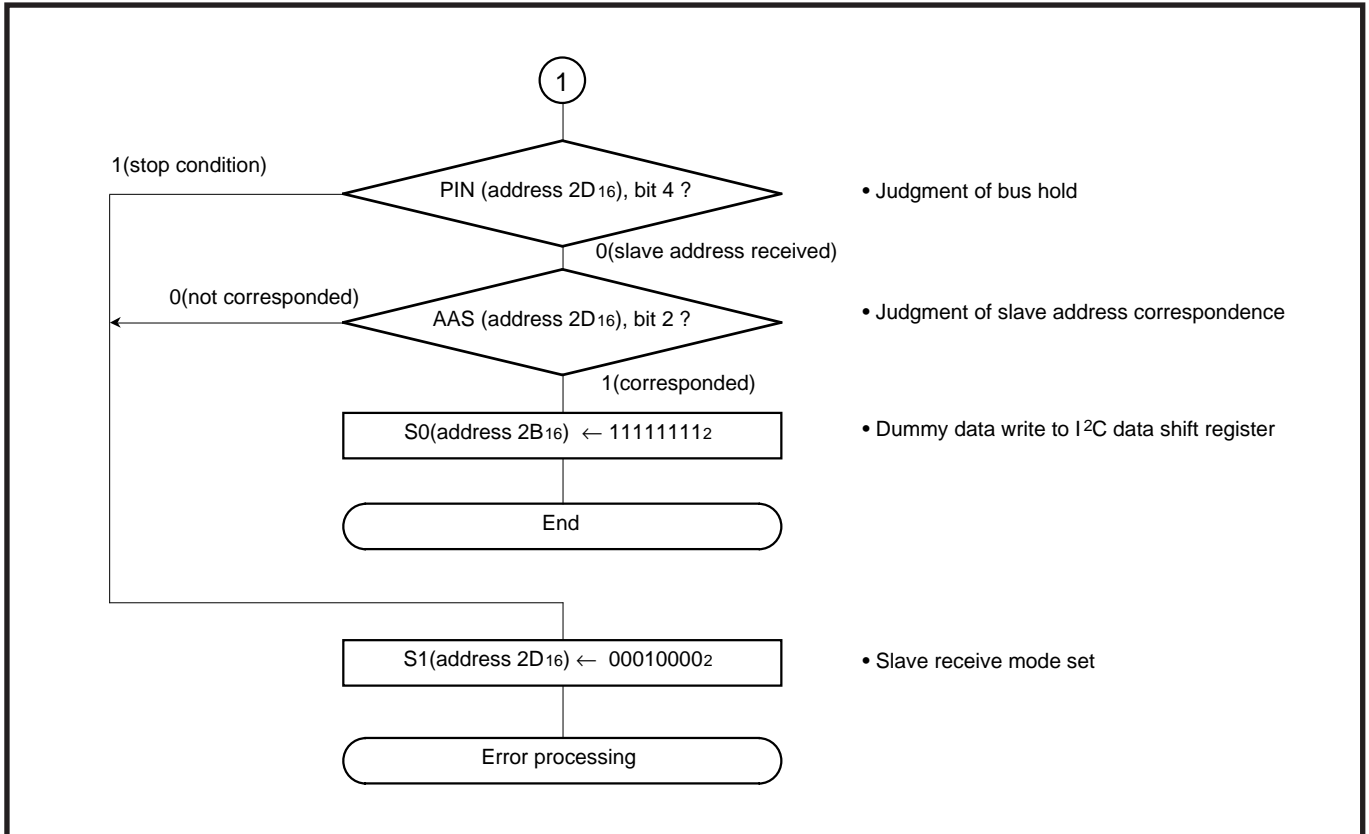


Fig. 2.4.23 Reception process of START condition and slave address

APPLICATION

2.4 Multi-master I²C-BUS interface

② Reception of command

Confirm correct completion of the command reception in the interrupt after receiving the command. After confirming correct command from the host, write a dummy data to the I²C data shift register to wait for reception of the next slave address.

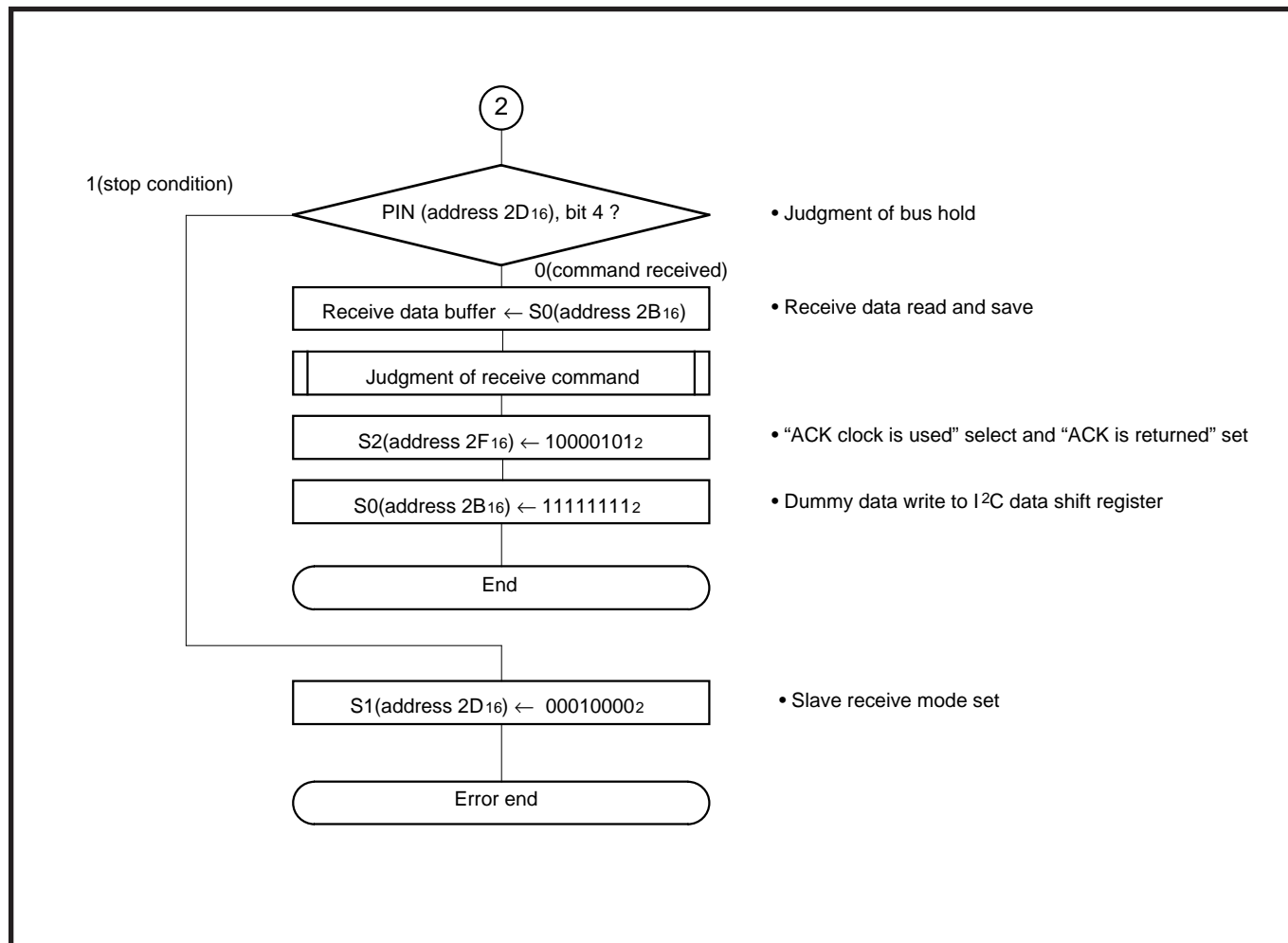


Fig. 2.4.24 Reception process of command

③ Reception of RESTART condition and slave address + read bit

After receiving a slave address, prepare a transmission data.

Judgment whether receiving a data or transmitting is required, because the mode is automatically switched between the receive mode and the transmit mode depending on the R/W bit of the received slave address. Accordingly, judge whether read or write referring the slave address comparison flag (AAS bit; bit 2 of address 002D₁₆).

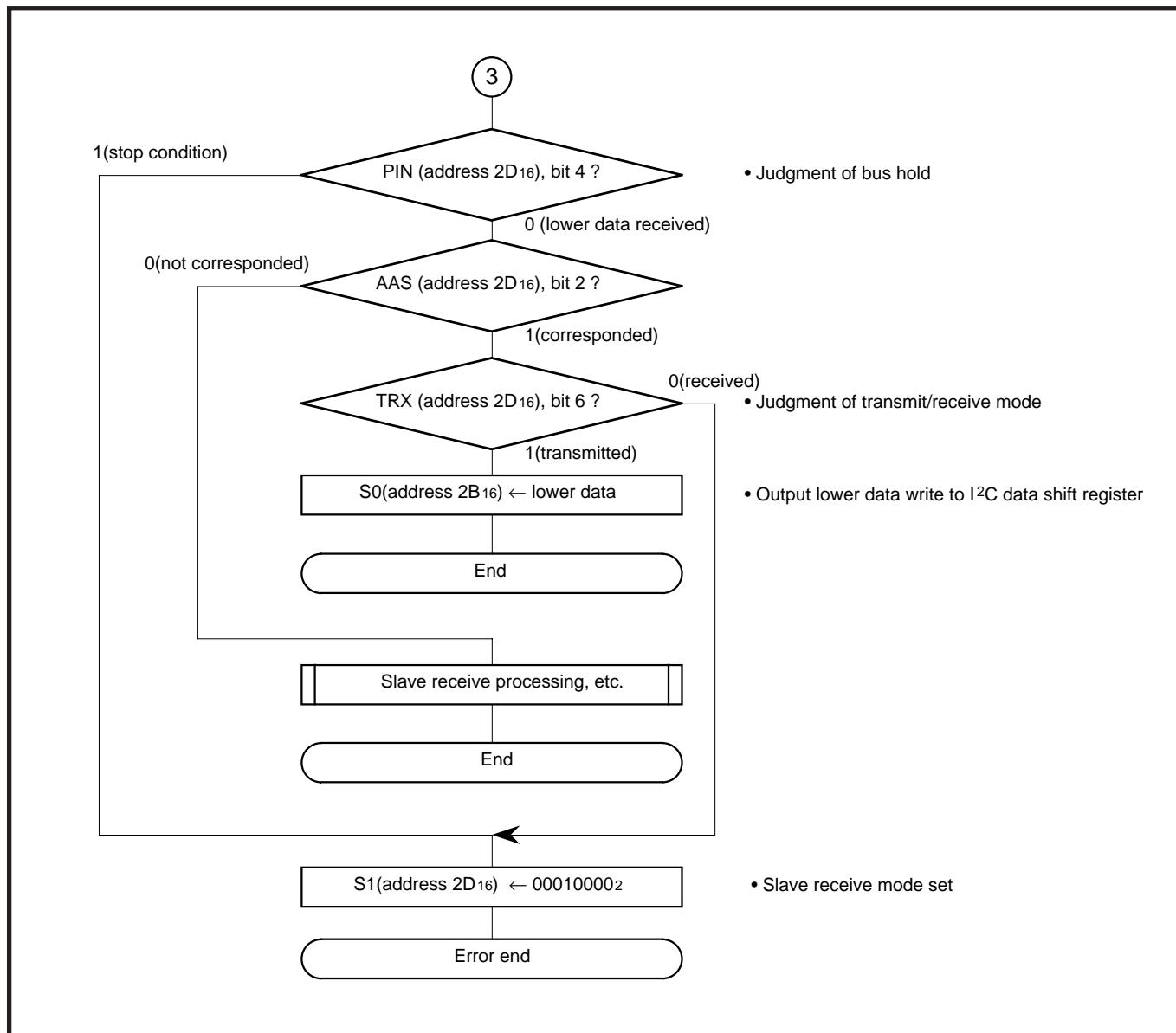


Fig. 2.4.25 Reception process of RESTART condition and slave address + read bit

APPLICATION

2.4 Multi-master I²C-BUS interface

④ Transmission of lower data

Before transmitting the upper data, restart to transmit the data at ④ and confirm correct completion of transmission of the lower data set in the slave address reception interrupt.

After that, transmit the upper data.

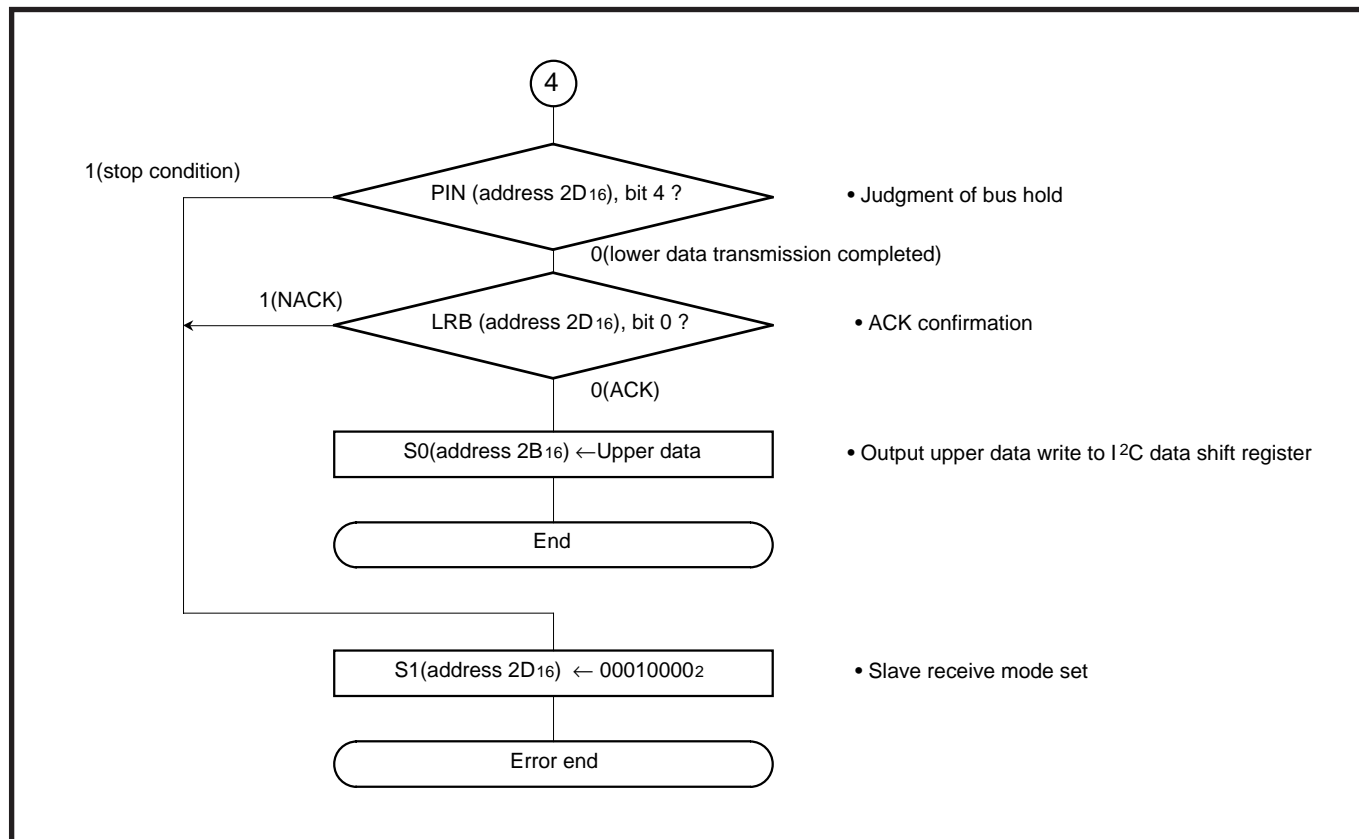


Fig. 2.4.26 Transmission process of lower data

⑤ Transmission of upper data

Confirm correct completion of the upper data transmission. The master returns the NACK toward the transmitted second-byte data, the upper data. Accordingly, confirm that the last received bit (LRB bit; bit 0 of address 002D₁₆) is "1".

After that, write a dummy data to the I²C data shift register (address 002B₁₆) and wait for the interrupt of STOP condition.

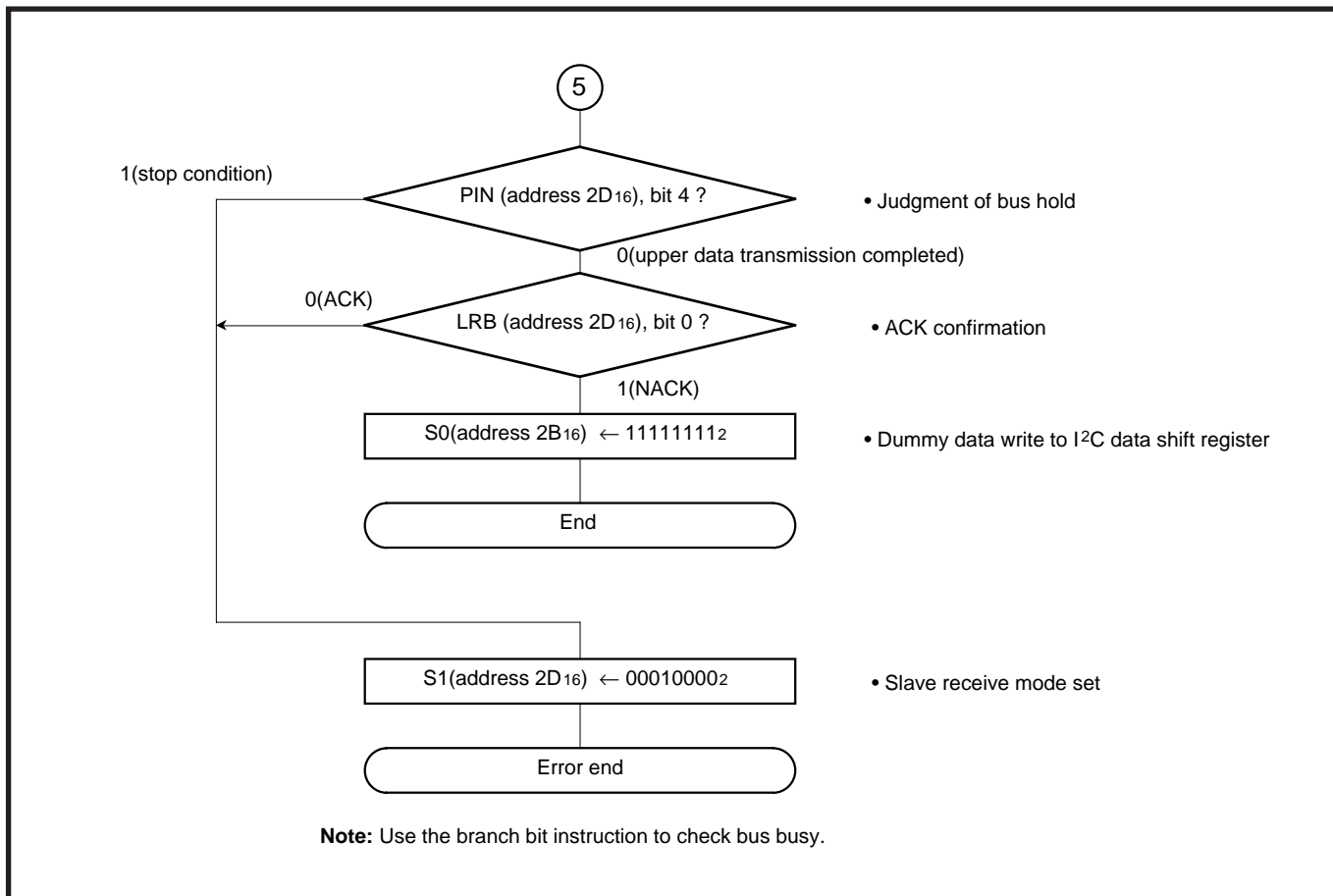


Fig. 2.4.27 Transmission process of upper data

APPLICATION

2.4 Multi-master I²C-BUS interface

⑥ Reception of STOP condition

Confirm that the STOP condition is correctly output and the bus is released.

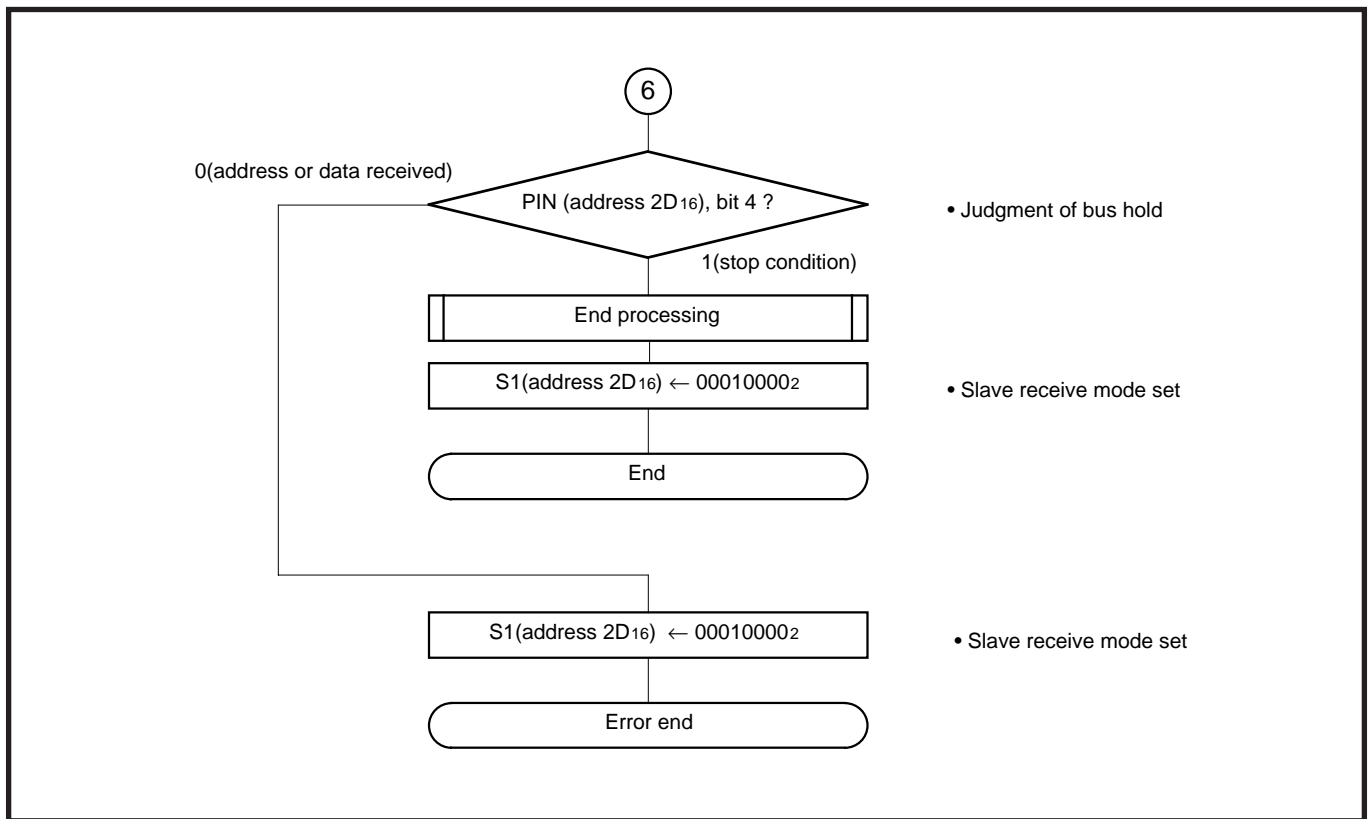


Fig. 2.4.28 Reception of STOP condition

2.4.7 Notes on multi-master I²C-BUS interface

(1) Read-modify-write instruction

Each register of the multi-master I²C-BUS interface has bits to change by hardware. The precautions when the read-modify-write instruction such as SEB, CLB etc. is executed for each register of the multi-master I²C-BUS interface are described below.

① **I²C data shift register (S0: address 002B₁₆)**

When executing the read-modify-write instruction for this register during transfer, data may become a value not intended.

② **I²C address register (S0D: address 002C₁₆)**

When the read-modify-write instruction is executed for this register at detecting the STOP condition, data may become a value not intended.

● **Reason**

It is because hardware changes the read/write bit (RBW) at detecting the STOP condition.

③ **I²C status register (S1: address 002D₁₆)**

Do not execute the read-modify-write instruction for this register because all bits of this register are changed by hardware.

④ **I²C control register (S1D: address 002E₁₆)**

When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become a value not intended.

● **Reason**

Because hardware changes the bit counter (BC0 to BC2).

⑤ **I²C clock control register (S2: address 002F₁₆)**

The read-modify-write instruction can be executed for this register.

⑥ **I²C START/STOP condition control register (S2D: address 0030₁₆)**

The read-modify-write instruction can be executed for this register.

APPLICATION

2.4 Multi-master I²C-BUS interface

(2) START condition generating procedure using multi-master

① Procedure example (The necessary conditions of the generating procedure are described as the following ② to ⑤).

```
LDA #SLADR          (Taking out of slave address value)
SEI                 (Interrupt disabled)
BBS 5, S1, BUSBUSY (BB flag confirming and branch process)
BUSFREE:
STA S0              (Writing of slave address value)
LDM #$F0, S1       (Trigger of START condition generating)
CLI                 (Interrupt enabled)
:
:
BUSBUSY:
CLI                 (Interrupt enabled)
:
:
```

- ② Use “Branch on Bit Set” of “BBS 5, S1, –” for the BB flag confirming and branch process.
- ③ Use “STA, STX” or “STY” of the zero page addressing instruction for writing the slave address value to the I²C data shift register (S0: address 002B₁₆).
- ④ Execute the branch instruction of above 2 and the store instruction of above 3 continuously shown the above procedure example.
- ⑤ Disable interrupts during the following three process steps:
 - BB flag confirming
 - Writing of slave address value
 - Trigger of START condition generating

(3) RESTART condition generating procedure in master

① Procedure example (The necessary conditions of the generating procedure are described as the following ② to ④). Execute the following procedure when the PIN bit is “0”.

```
LDM #$00, S1       (Select slave receive mode)
LDA #SLADR          (Taking out of slave address value)
SEI                 (Interrupt disabled)
STA S0              (Writing of slave address value)
LDM #$F0, S1       (Trigger of RESTART condition generating)
CLI                 (Interrupt enabled)
:
:
```

- ② Select the slave receive mode when the PIN bit is “0”. Do not write “1” to the PIN bit. Neither “0” nor “1” is specified for the writing to the BB bit. The TRX bit becomes “0” and the SDA pin is released.
- ③ The SCL pin is released by writing the slave address value to the I²C data shift register.
- ④ Disable interrupts during the following two process steps:
 - Writing of slave address value
 - Trigger of RESTART condition generating

(4) Writing to I²C status register

Do not execute an instruction to set the PIN bit to “1” from “0” and an instruction to set the MST and TRX bits to “0” from “1” simultaneously. It is because it may enter the state that the SCL pin is released and the SDA pin is released after about one machine cycle. Do not execute an instruction to set the MST and TRX bits to “0” from “1” simultaneously when the PIN bit is “1”. It is because it may become the same as above.

(5) STOP condition generating procedure in master

① Procedure example (The necessary conditions of the generating procedure are described as the following ② to ⑤).

```

SEI                (Interrupt disabled)
LDM #$C0, S1      (Select master transmit mode)
NOP               (Set NOP)
LDM #$D0, S1      (Trigger of STOP condition generating)
CLI               (Interrupt enabled)
:
:

```

② When setting the master transmit mode, write "0" to the PIN bit.

③ Execute the NOP instruction after the master transmit mode is set. In addition, set the STOP condition to be triggered within 10 machine cycles after the master transmit mode has been set.

④ Make sure all interrupts are disabled during the term from when the master transmit mode is set until the triggering process, which generates the STOP condition, is complete.

⑤ The above procedure is only applicable to the M38513E4.

(6) Process of after STOP condition generating

Do not write data in the I²C data shift register S0 and the I²C status register S1 until the bus busy flag BB becomes "0" after generating the STOP condition in the master mode. It is because the STOP condition waveform might not be normally generated. Reading to the above registers do not have the problem.

(7) STOP condition input at 7th clock pulse

In the slave mode, the STOP condition is input at the 7th clock pulse while receiving a slave address or data. As the clock pulse is continuously input, the SDA line may be held at LOW even if flag BB is set to "0".

Countermeasure:

Write dummy data to the I²C shift register or reset the ES0 bit in the S1D register (ES0 = "L" → ES0 = "H") during a stop condition interrupt routine with flag PIN = "1".

Note: Do not use the read-modify-write instruction at this time. Furthermore, when the ES0 bit is set to "0", it becomes a general-purpose port ; so that the port must be set to input mode or "H".

Note: The M38514E6/M6 does not have this problem which is the SDA line remaining "L".

(8) ES0 bit switch

In standard clock mode when SSC = "00010₂" or in high-speed clock mode, flag BB may switch to "1" if ES0 bit is set to "1" when SDA is "L".

Countermeasure:

Set ES0 to "1" when SDA is "H".

APPLICATION

2.4 Multi-master I²C-BUS interface

2.4.8 Notes on programming for SMBUS interface

(1) Time out process

For a smart battery system, the time out process with a program is required so that the communication can be completed even when communication is interrupted. It is because there is possibility of extracting a battery from a PC.

The specifications are defined so that communication has been able to be completed within 25 ms from START condition to STOP condition and within 10 ms from the ACK pulse from the ACK pulse of each byte. Accordingly, the following two should be considered as count start conditions.

① SDA falling edge caused by SCL/SDA interrupt

This is the countermeasure for a communication interrupt in the middle of from START condition to a slave address. However, the detection condition must be considered because a interrupt is also generated by communication from other masters to other slaves.

② SMBUS interrupt after receiving slave address

This is the countermeasure for when communication is interrupted from receiving a slave address until receiving a command.

(2) Low hold of communication line

The I²C-BUS interface conforms to the I²C-BUS Standard Specifications. However, because the use condition of SMBUS differs from the I²C-BUS's, there is possibility of occurrences of the following two problems.

① Low hold of SDA line caused by ACK pulse at voltage drop of communication line

When the SMBUS voltage slowly drops, that is caused by extracting a battery from equipment or turning off a PC's power or etc., it might be incorrectly treated as the SCL pulse near the threshold level voltage.

When the SDA is judged "L" in that condition, it becomes the general call and the ACK is transmitted. However, when the SCL remains "L" at the ACK pulse, the SDA continuously remains "L" until input of the next SCL pulse.

Countermeasure:

As explained before, start the time out count at the falling of SDA line of START condition and reset ES0 bit of the S1D register when the time out is satisfied (**Note**).

② STOP condition input at 7th clock pulse

In the slave mode, the STOP condition is input at the 7th clock pulse while receiving a slave address or data. As the clock pulse is continuously input, the SDA line may be held at "L" even if flag BB is set to "0".

Countermeasure:

Write dummy data to the I²C shift register or reset the ES0 bit in the S1D register (ES0 = "L" → ES0 = "H") during a stop condition interrupt routine with flag PIN = "1".

Note: Do not use the read-modify-write instruction at this time. Furthermore, when the ES0 bit is set to "0", it becomes a general-purpose port ; so that the port must be set to input mode or "H".

Note: The M38514E6/M6 does not have this problem which is the SDA line remaining "L".

2.5 PWM

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

2.5.1 Memory map

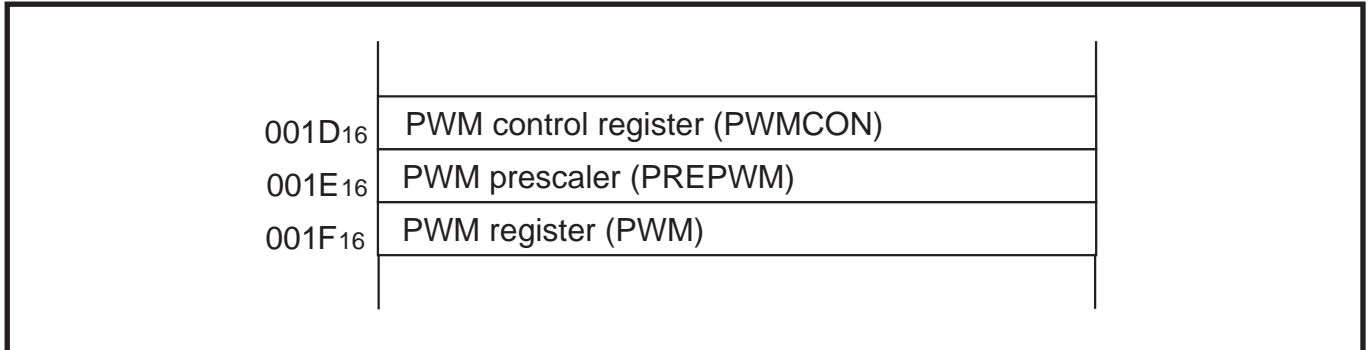


Fig. 2.5.1 Memory map of registers relevant to PWM

2.5.2 Related registers

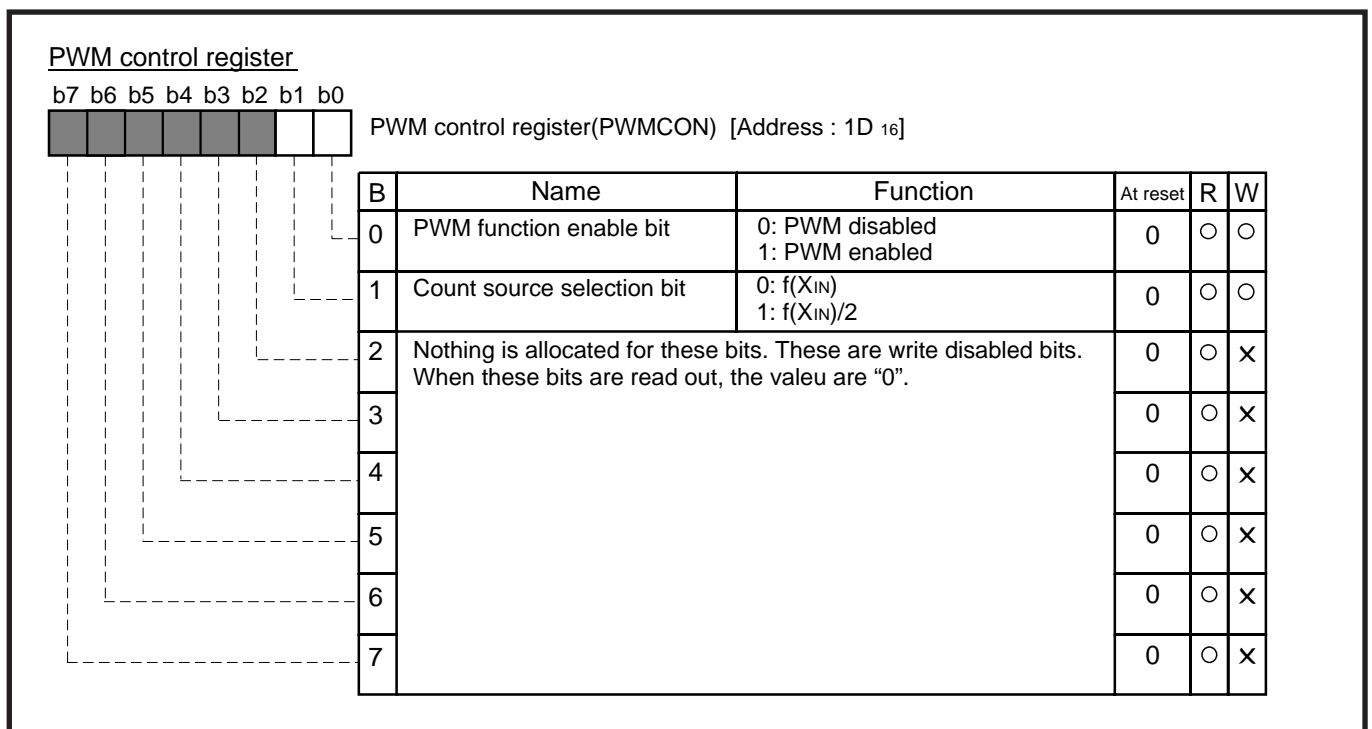


Fig. 2.5.2 Structure of PWM control register

APPLICATION

2.5 PWM

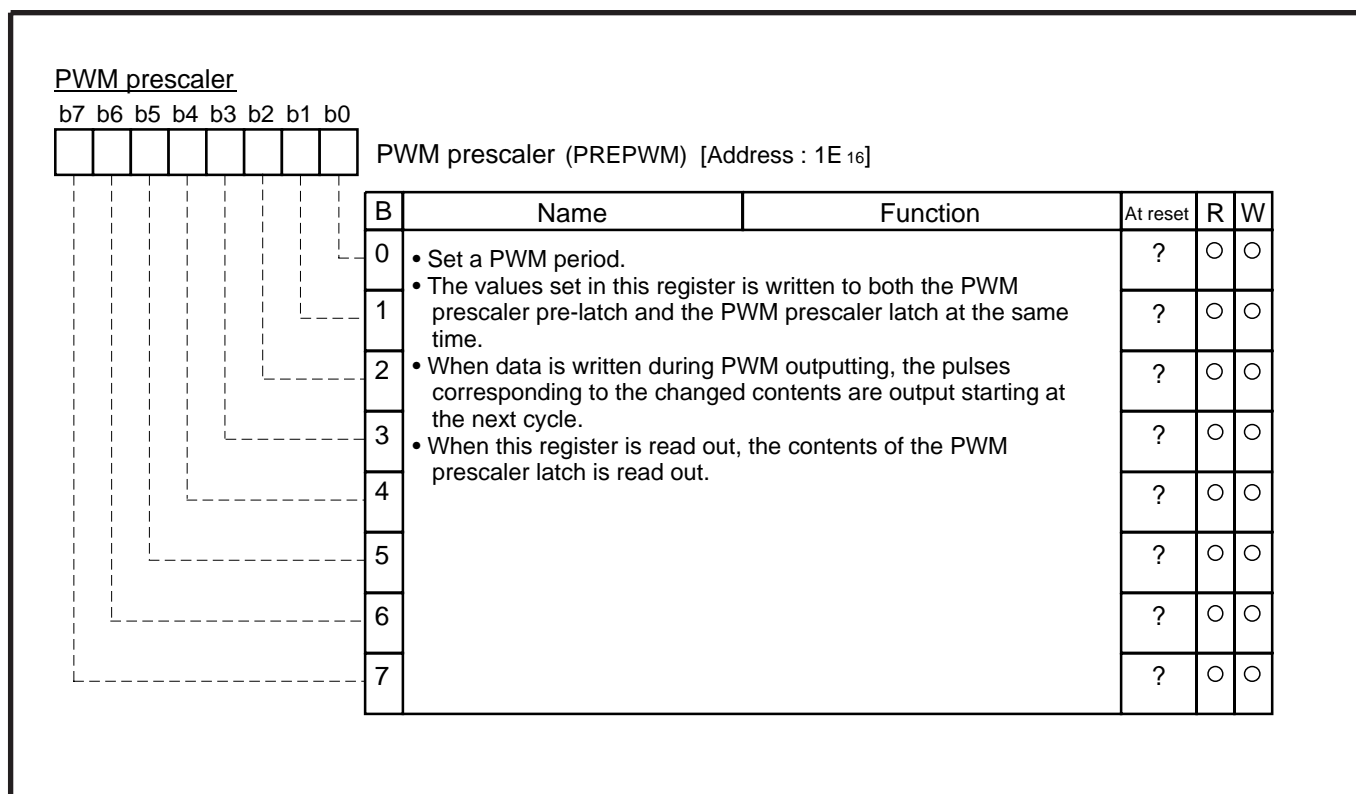


Fig. 2.5.3 Structure of PWM prescaler

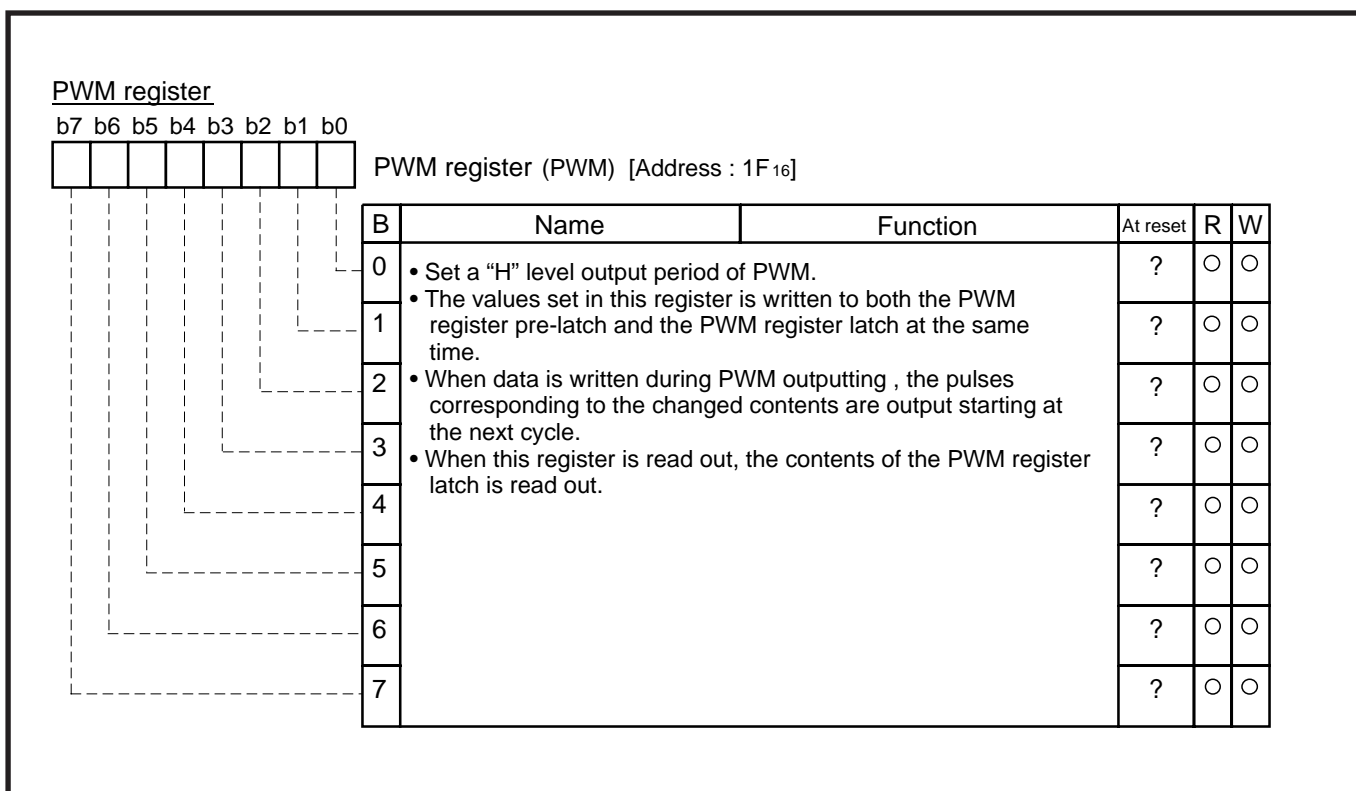


Fig. 2.5.4 Structure of PWM register

2.5.3 PWM output circuit application example

<Motor control>

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

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

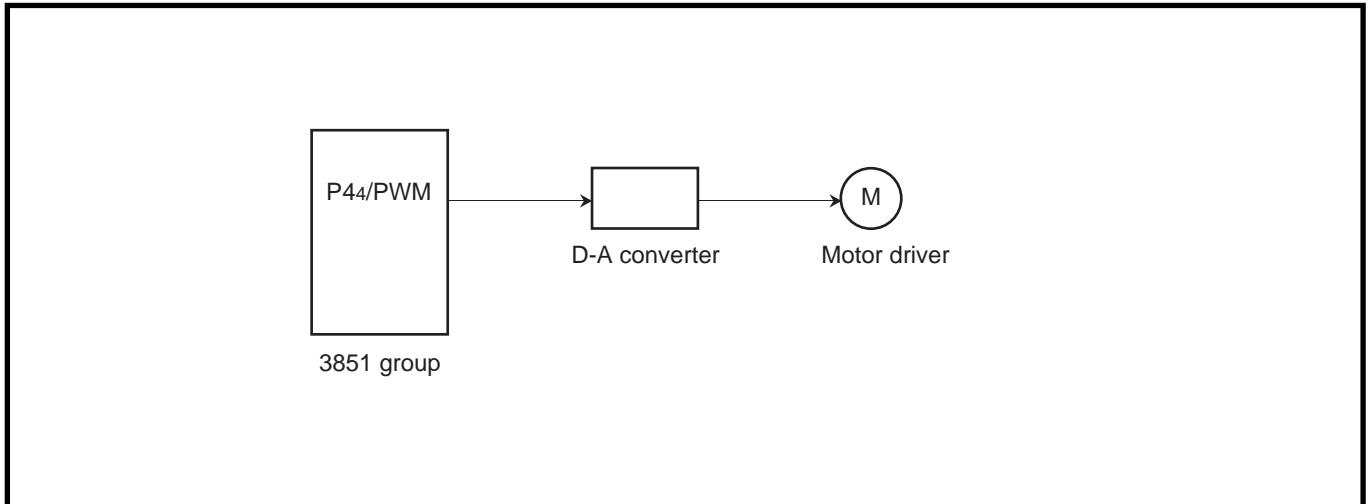


Fig. 2.5.5 Connection diagram

- Specifications :**
- Motor is controlled by using the PWM output function of 8-bit resolution.
 - Clock $f(XIN) = 5.0 \text{ MHz}$
 - "T", PWM cycle : $102 \mu\text{s}$
 - "t", "H" level width of output pulse : $40 \mu\text{s}$ (Fixed speed)
- * A motor speed can be changed by modifying the "H" level width of output pulse.

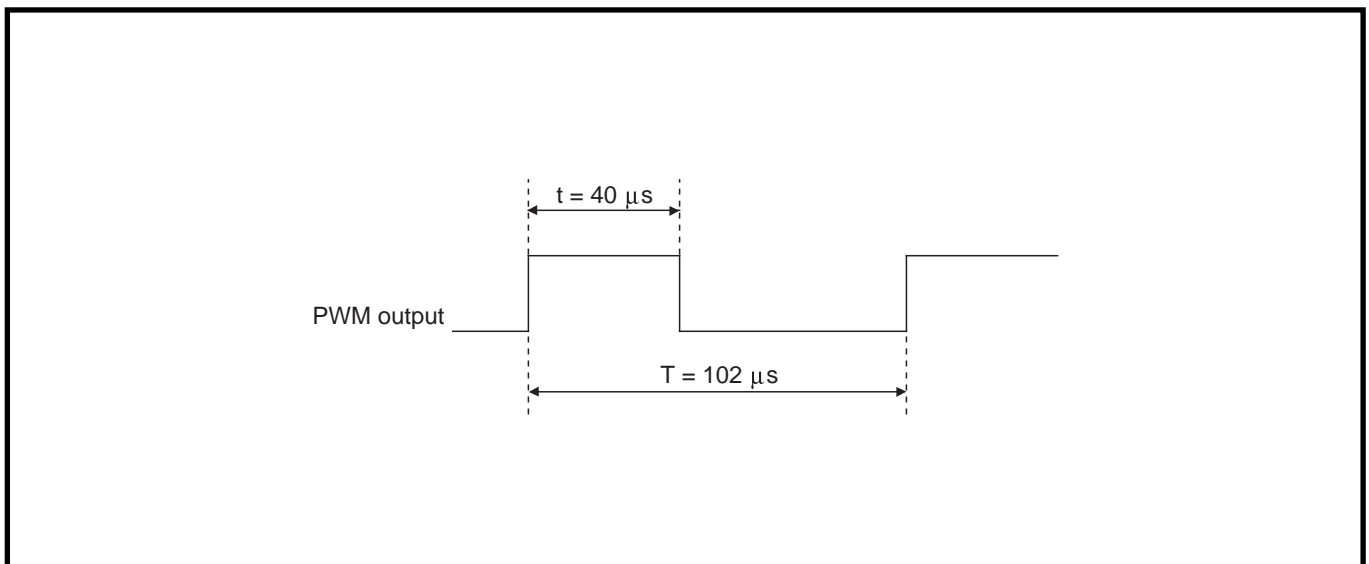


Fig. 2.5.6 PWM output timing

APPLICATION

2.5 PWM

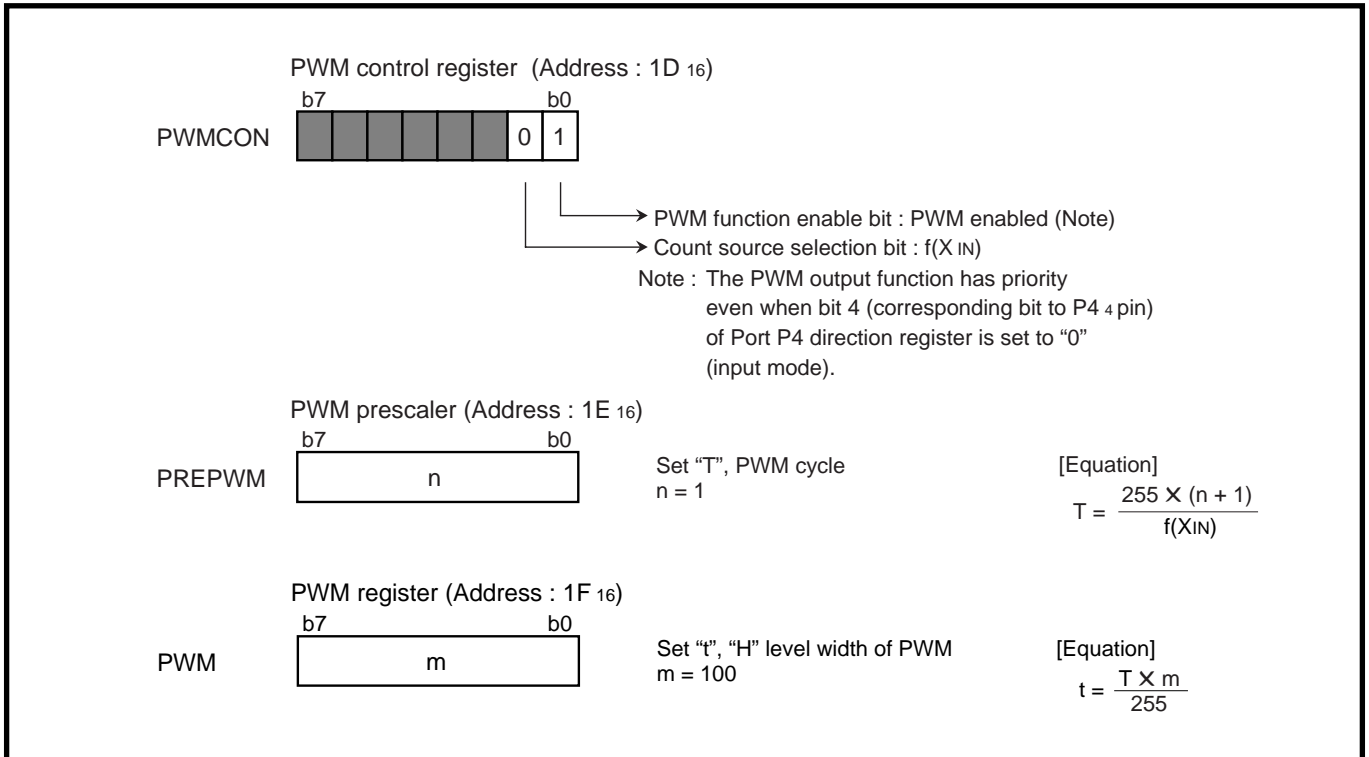


Fig. 2.5.7 Setting of related registers

<About PWM output>

- Set the PWM function enable bit to "1" : The P4₄/PWM pin is used as the PWM pin. The pulse beginning with "H" level pulse is output.
- Set the PWM function enable bit to "0" : The P4₄/PWM pin is used as the port P4₄. Thus, when fixing the output level, take the following procedure:
 - Write an output value to bit 4 of the port P4 register.
 - Write "0001000₂" to the port P4 direction register.
- After data is set to the PWM prescaler and the PWM register, the PWM waveforms corresponding to updated data will be output from the next repetitive cycle.

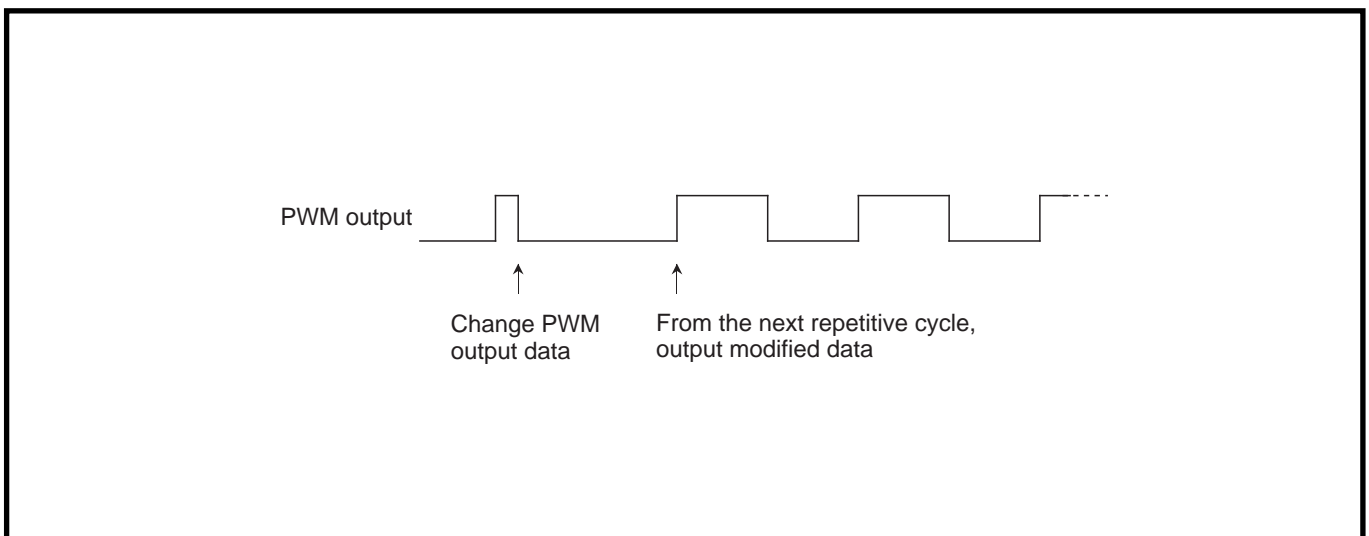


Fig. 2.5.8 PWM output

Control procedure : By setting the related registers as shown by Figure 2.5.7, PWM waveforms are output to the externals. This PWM output is integrated through the low pass filter, and that converted into DC signals is used for control of the motor.

Figure 2.5.9 shows control procedure.

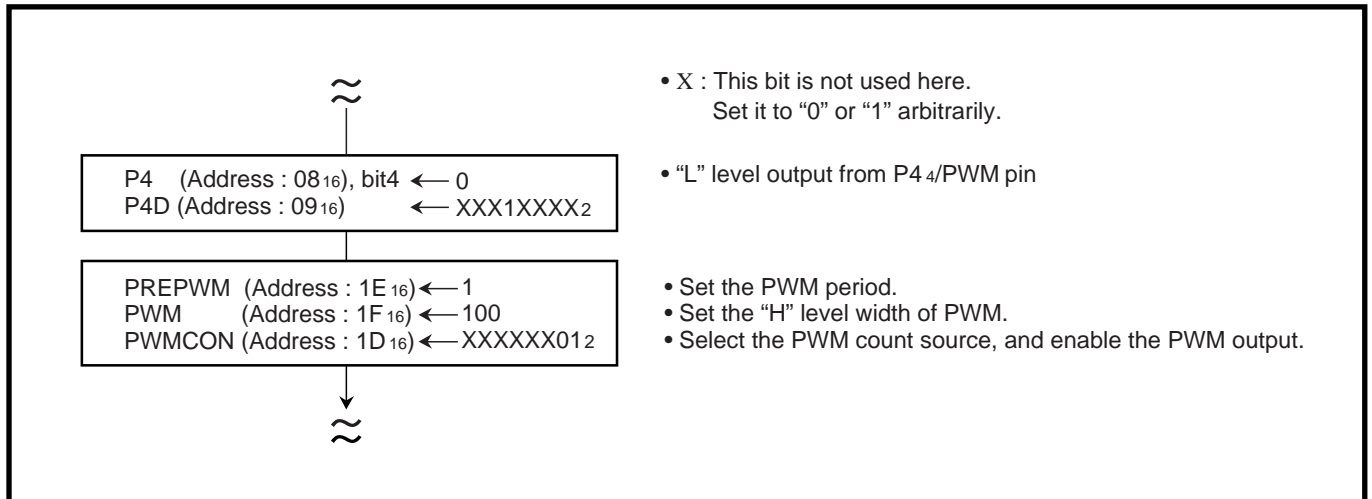


Fig. 2.5.9 Control procedure

2.5.4 Notes on PWM

The PWM starts after the PWM enable bit is set to enable and "L" level is output from the PWM pin. The length of this "L" level output is as follows:

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

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

APPLICATION

2.6 A-D converter

2.6 A-D converter

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

2.6.1 Memory map

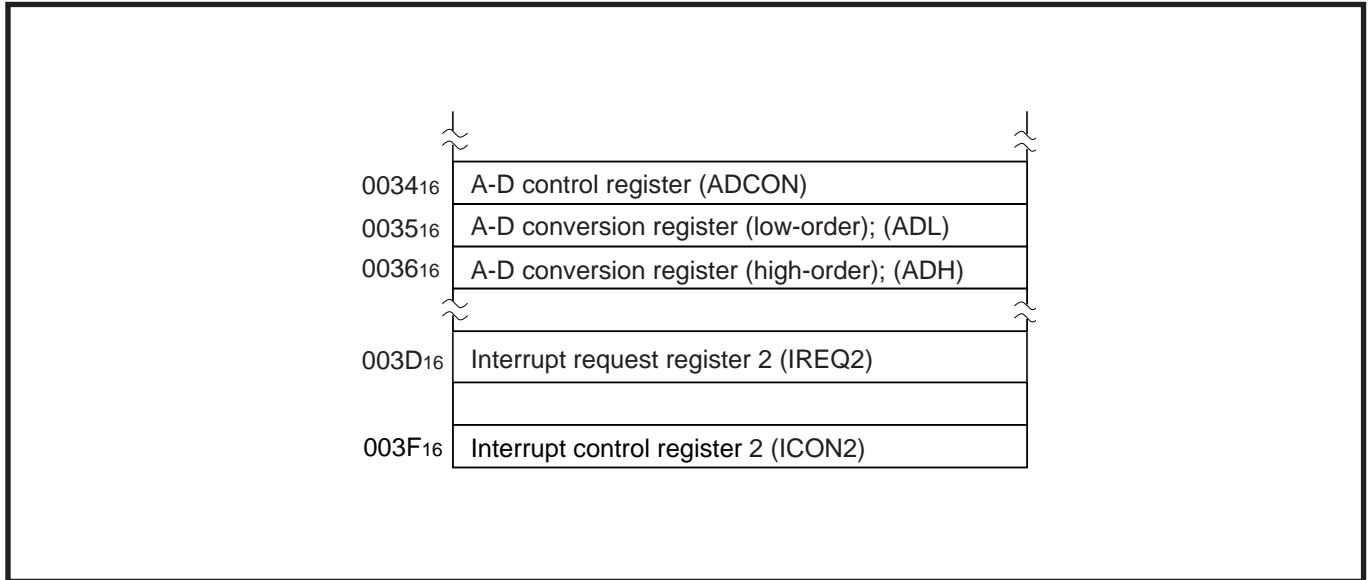


Fig. 2.6.1 Memory map of registers relevant to A-D converter

2.6.2 Relevant registers

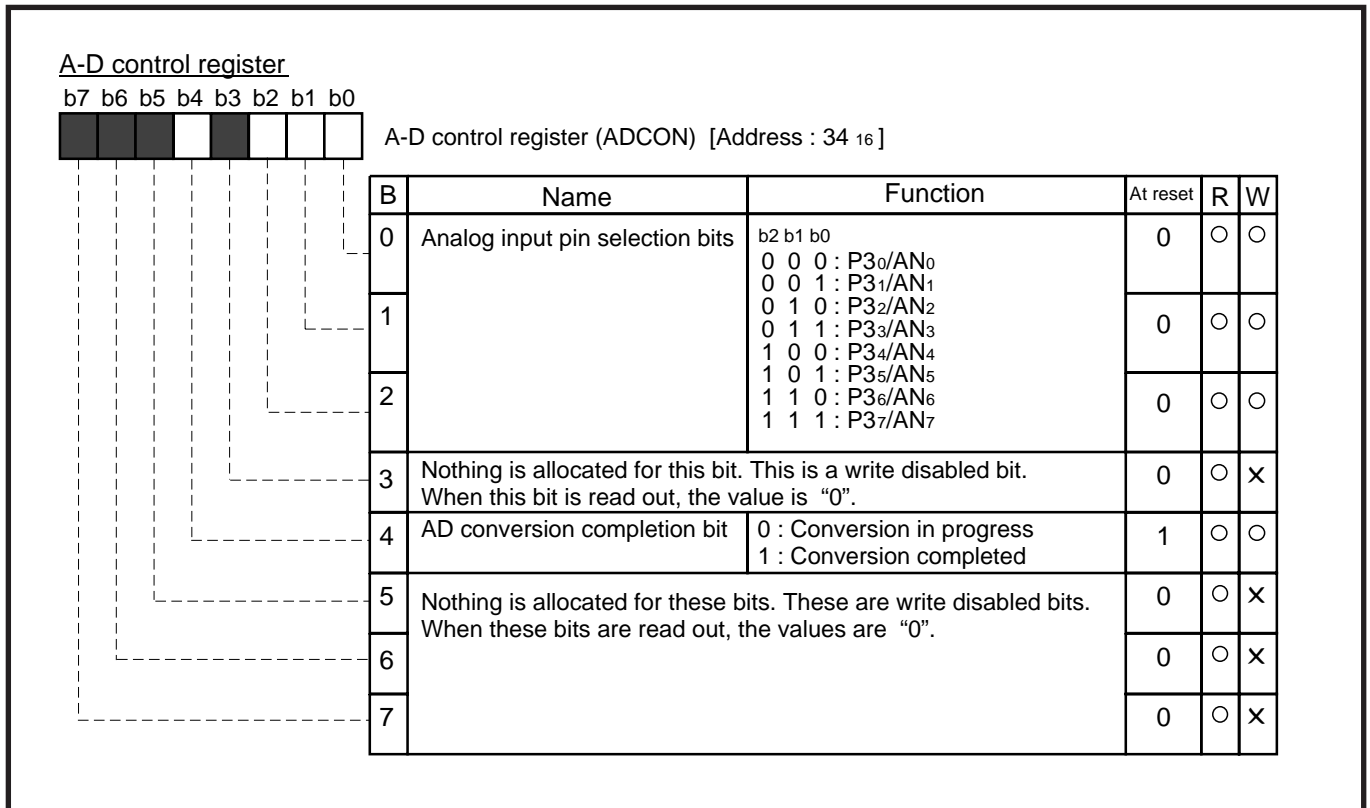


Fig. 2.6.2 Structure of A-D control register

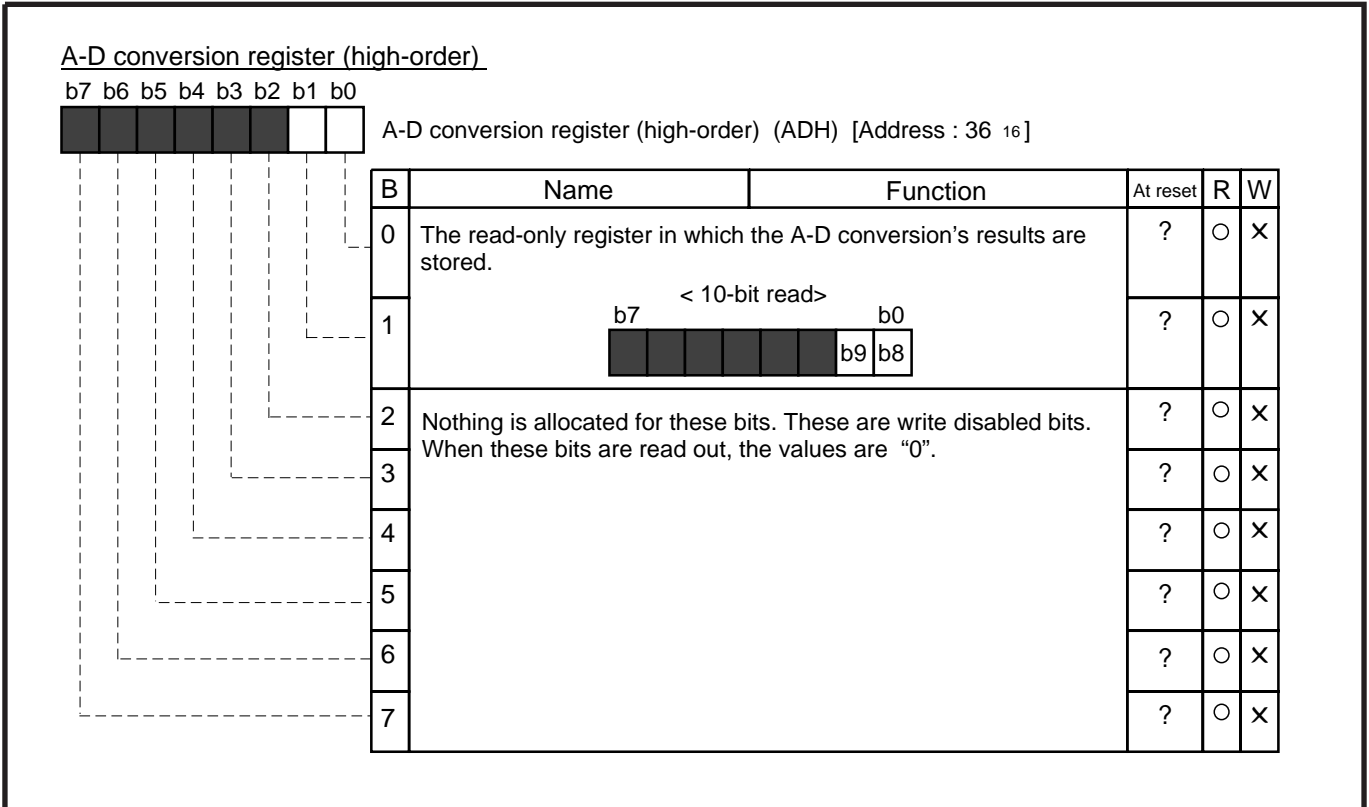


Fig. 2.6.3 Structure of A-D conversion register (high-order)

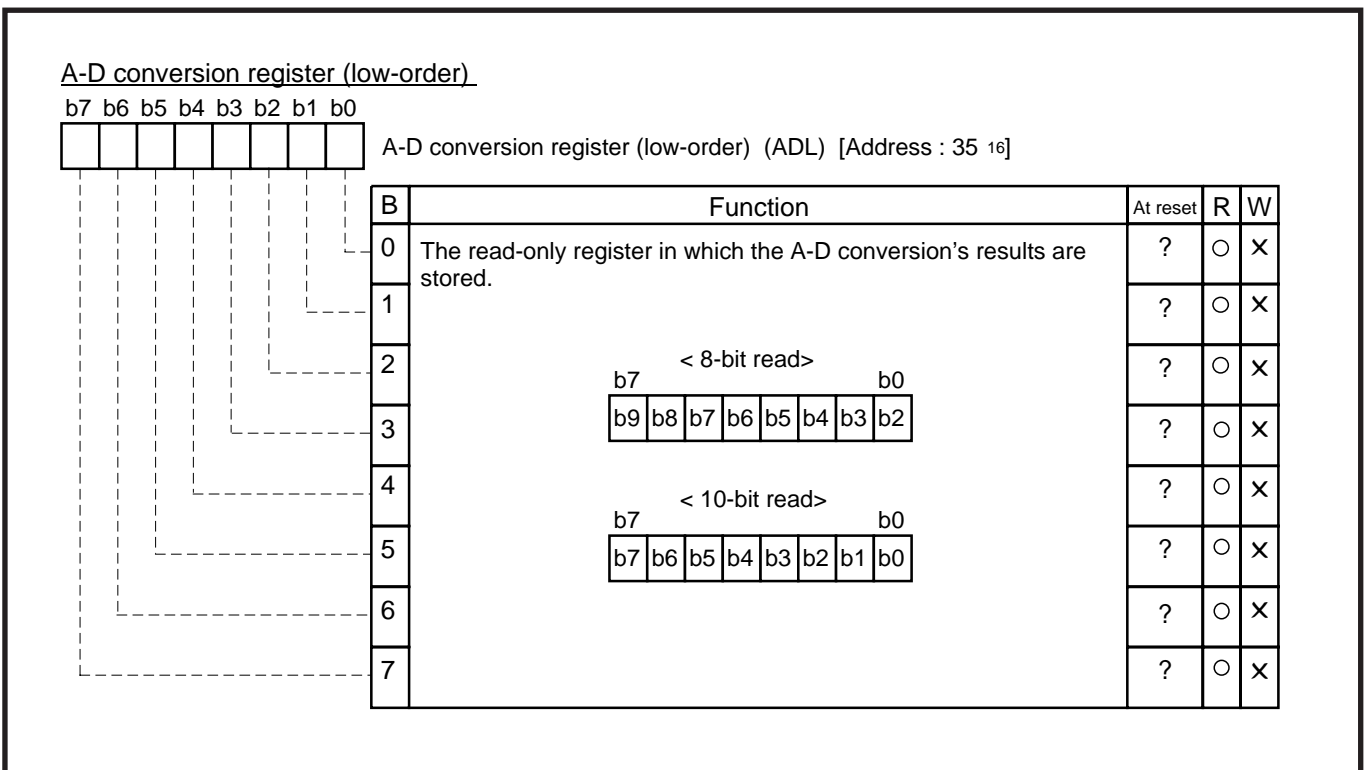


Fig. 2.6.4 Structure of A-D conversion register(low-order)

APPLICATION

2.6 A-D converter

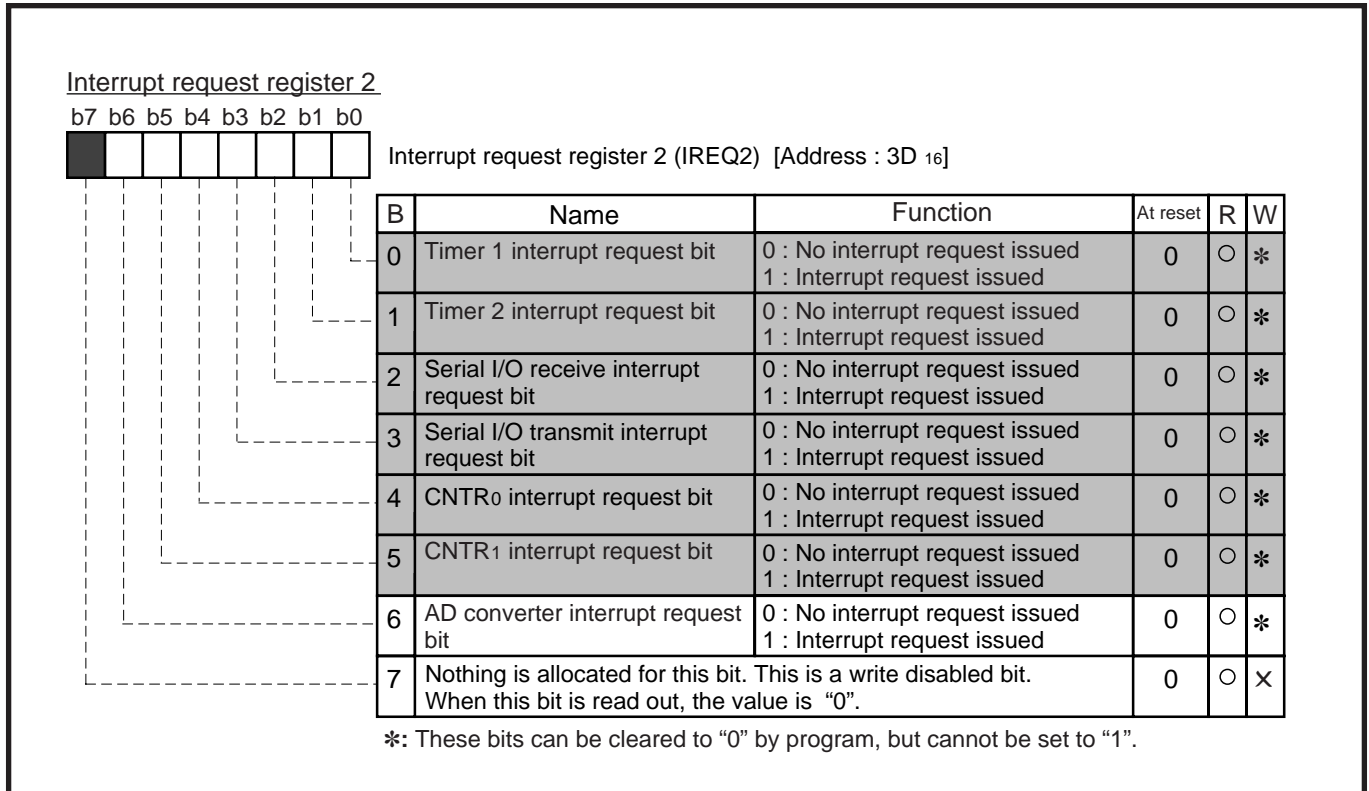


Fig. 2.6.5 Structure of Interrupt request register 2

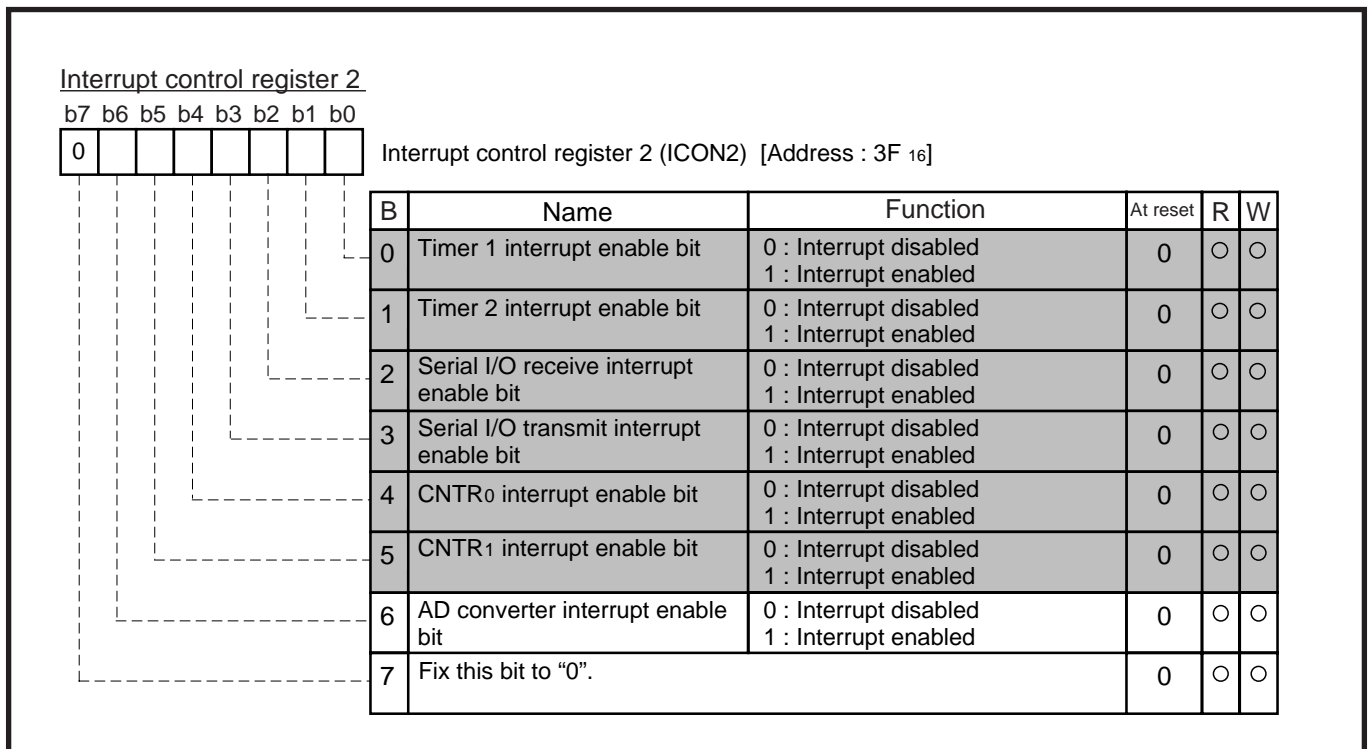


Fig. 2.6.6 Structure of Interrupt control register 2

2.6.3 A-D converter application examples

(1) Conversion of analog input voltage

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

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

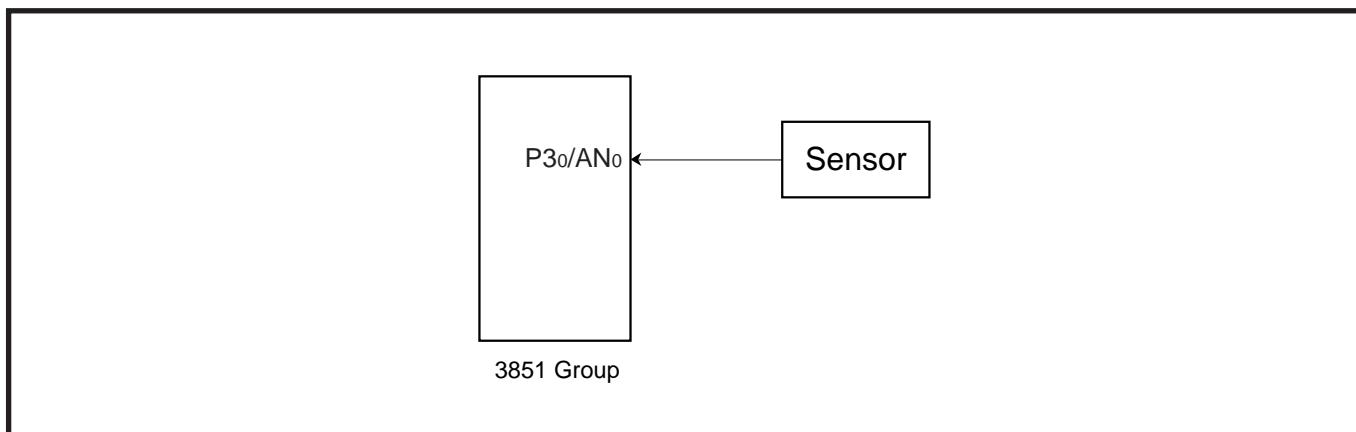


Fig. 2.6.7 Connection diagram

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

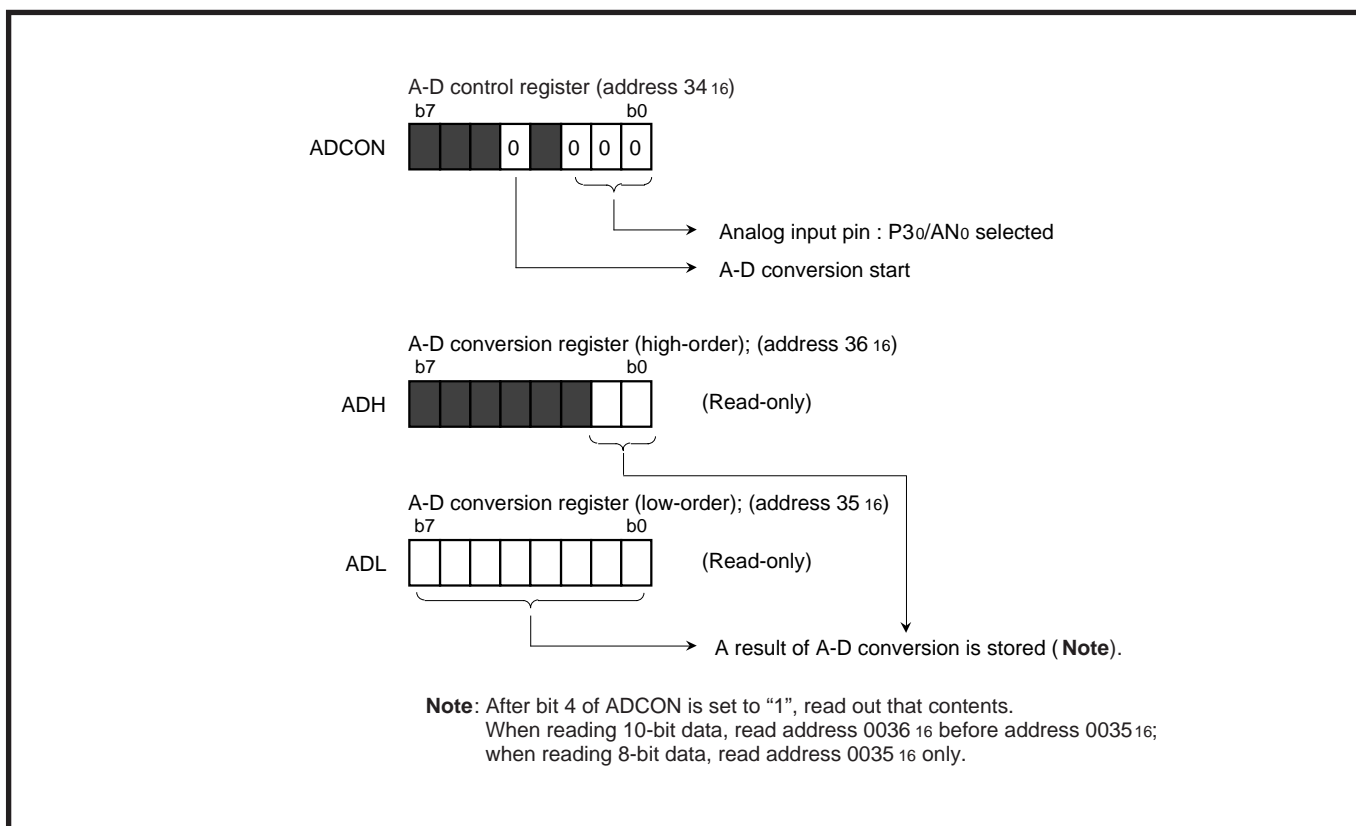


Fig. 2.6.8 Relevant registers setting

APPLICATION

2.6 A-D converter

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

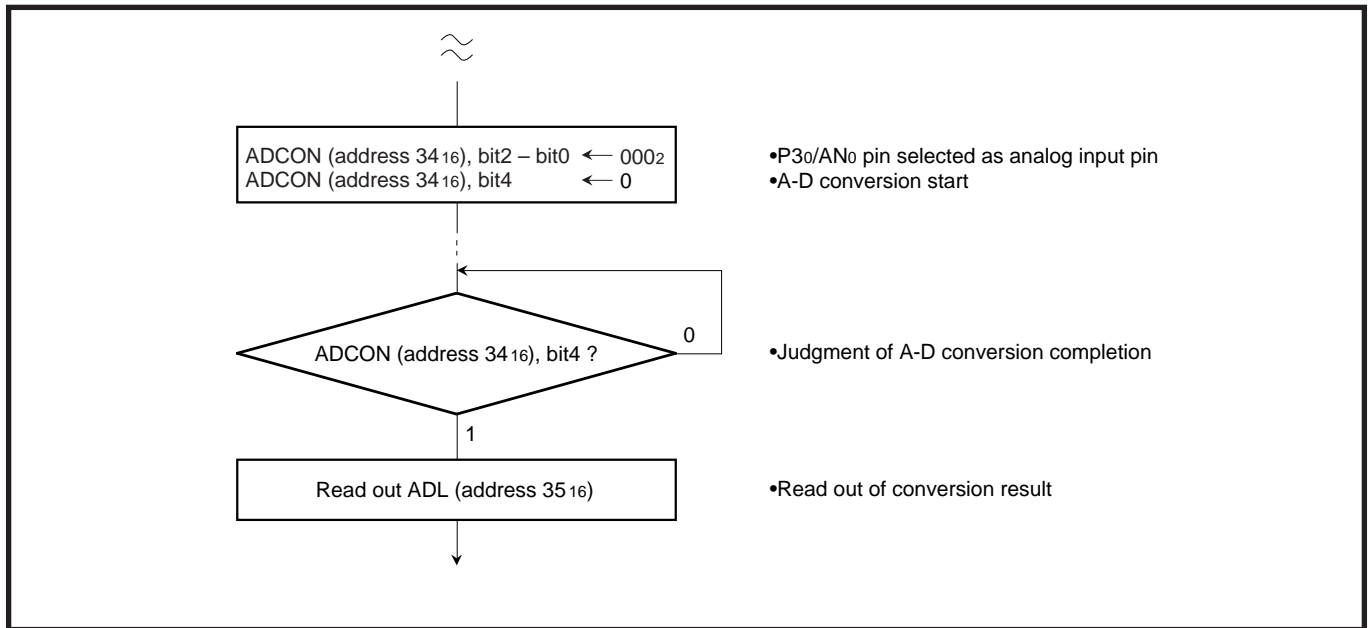


Fig. 2.6.9 Control procedure for 8-bit read

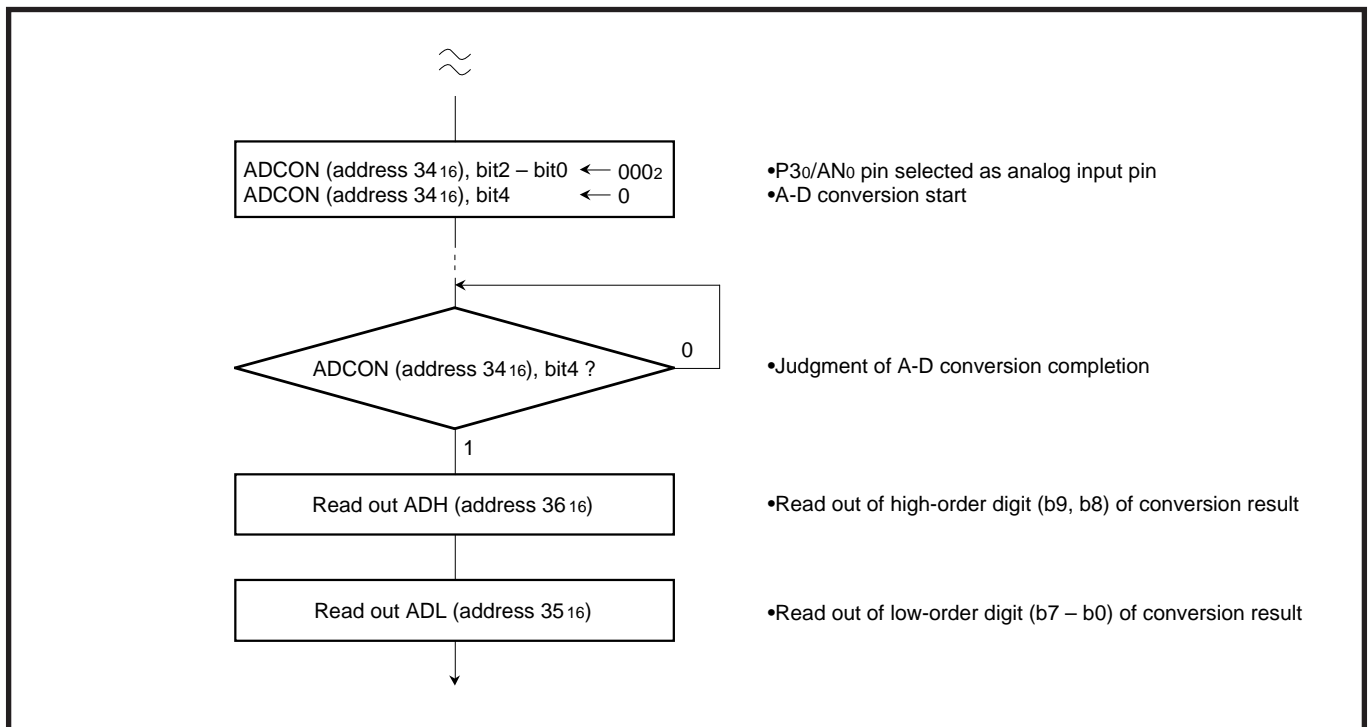


Fig. 2.6.10 Control procedure for 10-bit read

2.6.4 Notes on A-D converter

(1) Analog input pin

Make the signal source impedance for analog input low, or equip an analog input pin with an external capacitor of 0.01 μ F to 1 μ F. Further, be sure to verify the operation of application products on the user side.

● Reason

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

(2) A-D converter power source pin

Pins AVCC and AVSS are A-D converter power source pins. Regardless of using the A-D conversion function or not, connect them as following :

- AVCC : Connect to the VCC line
- AVSS : Connect to the VSS line

● Reason

If the AVCC and the AVSS pin are opened, the microcomputer may have a failure because of noise or others. Also, if the AVCC pin is connected to the VSS pin, current flows from AVCC to VSS.

(3) Clock frequency during A-D conversion

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

- $f(XIN)$ is 500 kHz or more
- Do not execute the **STP** instruction and **WIT** instruction

APPLICATION

2.7 Reset

2.7 Reset

2.7.1 Connection example of reset IC

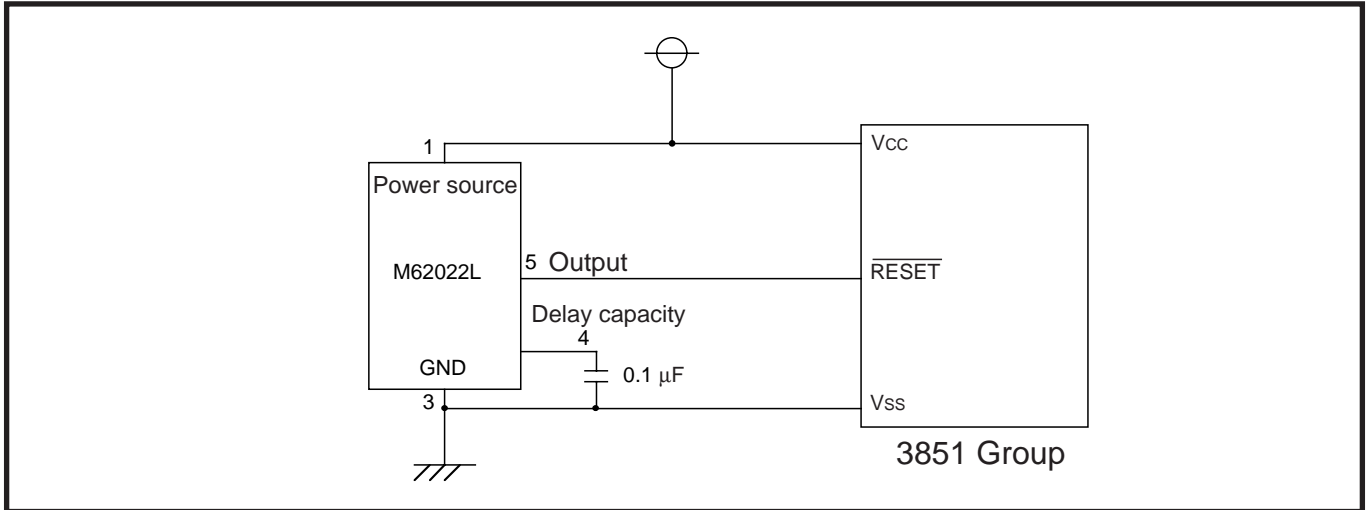


Fig. 2.7.1 Example of poweron reset circuit

Figure 2.7.2 shows the system example which switches to the RAM backup mode by detecting a drop of the system power source voltage with the INT interrupt.

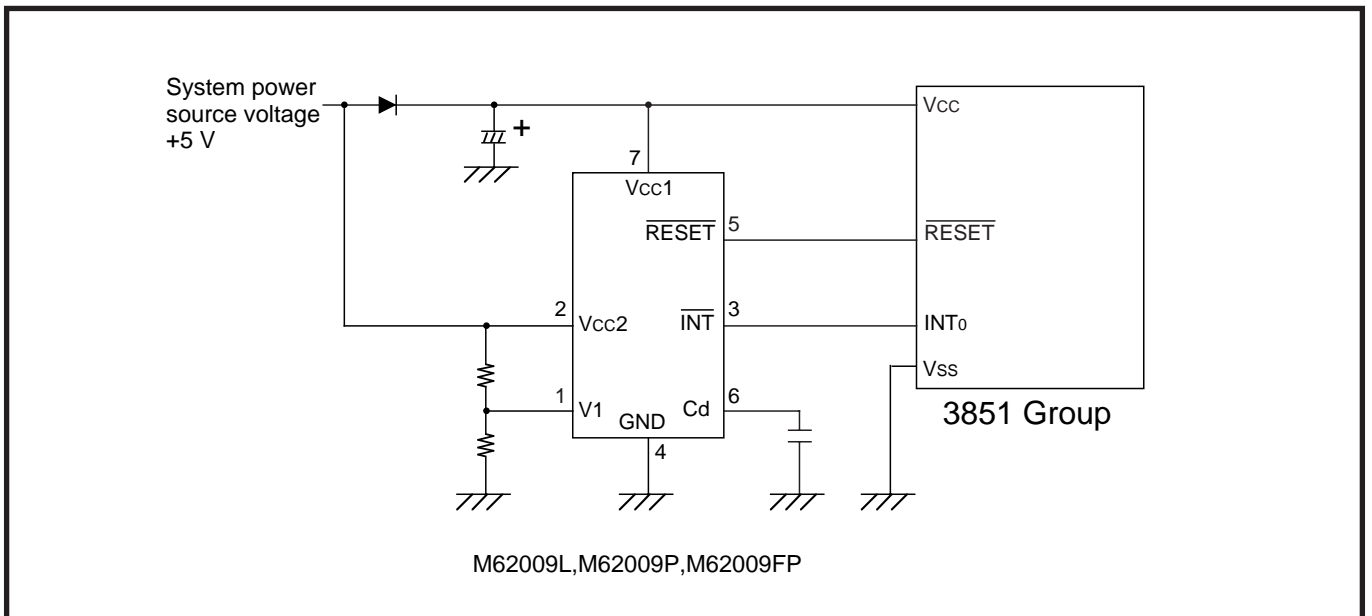


Fig. 2.7.2 RAM backup system

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

Connecting capacitor

In case where the $\overline{\text{RESET}}$ signal rise time is long, connect a ceramic capacitor or others across the $\overline{\text{RESET}}$ pin and the VSS pin. Use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, note the following :

- Make the length of the wiring which is connected to a capacitor as short as possible.
- Be sure to verify the operation of application products on the user side.

● Reason

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

APPLICATION

2.7 Reset

MEMORANDUM



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 confirmation form
- 3.7 ROM programming confirmation form
- 3.8 Mark specification form
- 3.9 Package outline
- 3.10 Machine instructions
- 3.11 List of instruction codes
- 3.12 SFR memory map
- 3.13 Pin configurations

APPENDIX

3.1 Electrical characteristics

3.1 Electrical characteristics

3.1.1 Absolute maximum ratings

Table 3.1.1 Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V _{CC}	Power source voltage	M38513E4/M4 M38514E6	-0.3 to 7.0	V
		M38514M6	-0.3 to 6.5	
V _I	Input voltage P00–P07, P10–P17, P20, P21, P24–P27, P30–P34, P40–P44, VREF	All voltages are based on V _{SS} . Output transistors are cut off.	-0.3 to V _{CC} +0.3	V
V _I	Input voltage P22, P23		-0.3 to 5.8	V
V _I	Input voltage RESET, XIN		-0.3 to V _{CC} +0.3	V
V _I	Input voltage CNVSS		-0.3 to 13	V
V _O	Output voltage P00–P07, P10–P17, P20, P21, P24–P27, P30–P34, P40–P44, XOUT		-0.3 to V _{CC} +0.3	V
V _O	Output voltage P22, P23		-0.3 to 5.8	V
P _d	Power dissipation	T _a = 25 °C	300	mW
T _{opr}	Operating temperature		-20 to 85	°C
T _{stg}	Storage temperature		-40 to 125	°C

3.1.2 Recommended operating conditions

Table 3.1.2 Recommended operating conditions (1)

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

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V _{CC}	Power source voltage (At 8 MHz)	4.0	5.0	5.5	V
	Power source voltage (At 4 MHz)	2.7	5.0	5.5	
V _{SS}	Power source voltage		0		V
V _{REF}	A-D convert reference voltage	2.0		V _{CC}	V
AV _{SS}	Analog power source voltage		0		V
V _{IA}	Analog input voltage AN ₀ –AN ₄		AV _{SS}	V _{CC}	V
V _{IH}	“H” input voltage P0 ₀ –P0 ₇ , P1 ₀ –P1 ₇ , P2 ₀ –P2 ₇ , P3 ₀ –P3 ₄ , P4 ₀ –P4 ₄		0.8V _{CC}	V _{CC}	V
V _{IH}	“H” input voltage (when I ² C-BUS input level is selected) SDA ₁ , SCL ₁		0.7V _{CC}	5.8	V
V _{IH}	“H” input voltage (when I ² C-BUS input level is selected) SDA ₂ , SCL ₂		0.7V _{CC}	V _{CC}	V
V _{IH}	“H” input voltage (when SMBUS input level is selected) SDA ₁ , SCL ₁		1.4	5.8	V
V _{IH}	“H” input voltage (when SMBUS input level is selected) SDA ₂ , SCL ₂		1.4	V _{CC}	V
V _{IH}	“H” input voltage RESET, XIN, CNV _{SS}		0.8V _{CC}	V _{CC}	V
V _{IL}	“L” input voltage P0 ₀ –P0 ₇ , P1 ₀ –P1 ₇ , P2 ₀ –P2 ₇ , P3 ₀ –P3 ₄ , P4 ₀ –P4 ₄		0	0.2V _{CC}	V
V _{IL}	“L” input voltage (when I ² C-BUS input level is selected) SDA ₁ , SDA ₂ , SCL ₁ , SCL ₂		0	0.3V _{CC}	V
V _{IL}	“L” input voltage (when SMBUS input level is selected) SDA ₁ , SDA ₂ , SCL ₁ , SCL ₂		0	0.6	V
V _{IL}	“L” input voltage RESET, CNV _{SS}		0	0.2V _{CC}	V
V _{IL}	“L” input voltage XIN		0	0.16V _{CC}	V
ΣIOH(peak)	“H” total peak output current P0 ₀ –P0 ₇ , P1 ₀ –P1 ₇ , P3 ₀ –P3 ₄ (Note)			-80	mA
ΣIOH(peak)	“H” total peak output current P2 ₀ , P2 ₁ , P2 ₄ –P2 ₇ , P4 ₀ –P4 ₄ (Note)			-80	mA
ΣIOL(peak)	“L” total peak output current (Note)	P0 ₀ –P0 ₇ , P1 ₀ –P1 ₂ , P3 ₀ –P3 ₄ (M38513E4/M4)		80	mA
		P0 ₀ –P0 ₇ , P3 ₀ –P3 ₄ (M38514E6/M6)			
ΣIOL(peak)	“L” total peak output current (Note)	P1 ₃ –P1 ₇ (M38513E4/M4)		80	mA
		P1 ₀ –P1 ₇ (M38514E6/M6)		120	mA
ΣIOL(peak)	“L” total peak output current P2 ₀ –P2 ₇ , P4 ₀ –P4 ₄ (Note)			80	mA
ΣIOH(avg)	“H” total average output current P0 ₀ –P0 ₇ , P1 ₀ –P1 ₇ , P3 ₀ –P3 ₄ (Note)			-40	mA
ΣIOH(avg)	“H” total average output current P2 ₀ , P2 ₁ , P2 ₄ –P2 ₇ , P4 ₀ –P4 ₄ (Note)			-40	mA
ΣIOL(avg)	“L” total average output current (Note)	P0 ₀ –P0 ₇ , P1 ₀ –P1 ₂ , P3 ₀ –P3 ₄ (M38513E4/M4)		40	mA
		P0 ₀ –P0 ₇ , P3 ₀ –P3 ₄ (M38514E6/M6)			
ΣIOL(avg)	“L” total average output current (Note)	P1 ₃ –P1 ₇ (M38513E4/M4)		40	mA
		P1 ₀ –P1 ₇ (M38514E6/M6)		60	mA
ΣIOL(avg)	“L” total average output current P2 ₀ –P2 ₇ , P4 ₀ –P4 ₄ (Note)			40	mA

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

APPENDIX

3.1 Electrical characteristics

Table 3.1.3 Recommended operating conditions (2)

(VCC = 2.7 to 5.5 V, Ta = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit	
		Min.	Typ.	Max.		
IOH(peak)	"H" peak output current P00-P07, P10-P17, P20, P21, P24-P27, P30-P34, P40-P44 (Note 1)			-10	mA	
IOL(peak)	"L" peak output current (Note 1)	P00-P07, P10-P12, P20-P27, P30-P34, P40-P44 (M38513E4/M4)			10	mA
		P00-P07, P20-P27, P30-P34, P40-P44 (M38514E6/M6)				
IOL(peak)	"L" peak output current (Note 1)	P13-P17 (M38513E4/M4)			20	mA
		P10-P17 (M38514E6/M6)				
IOH(avg)	"H" average output current P00-P07, P10-P17, P20, P21, P24-P27, P30-P34, P40-P44 (Note 2)			-5	mA	
IOL(avg)	"L" average output current (Note 2)	P00-P07, P10-P12, P20-P27, P30-P34, P40-P44 (M38513E4/M4)			5	mA
		P00-P07, P20-P27, P30-P34, P40-P44 (M38514E6/M6)				
IOL(avg)	"L" peak output current (Note 2)	P13-P17 (M38513E4/M4)			15	mA
		P10-P17 (M38514E6/M6)				
f(XIN)	Internal clock oscillation frequency (VCC = 4.0 to 5.5V) (Note 3)			8	MHz	
f(XIN)	Internal clock oscillation frequency (VCC = 2.7 to 5.5V) (Note 3)			4	MHz	

Notes 1: The peak output current is the peak current flowing in each port.

2: The average output current IOL(avg), IOH(avg) are average value measured over 100 ms.

3: When the oscillation frequency has a duty cycle of 50%.

3.1.3 Electrical characteristics

Table 3.1.4 Electrical characteristics (1)

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V _{OH}	“H” output voltage P00–P07, P10–P17, P20, P21, P24–P27, P30–P34, P40–P44 (Note)	I _{OH} = -10 mA V _{CC} = 4.0–5.5 V	V _{CC} -2.0			V
		I _{OH} = -1.0 mA V _{CC} = 2.7–5.5 V	V _{CC} -1.0			V
V _{OL}	“L” output voltage P00–P07, P10–P12, P20–P27 P30–P34, P40–P44 (M38513E4/M4)	I _{OL} = 10 mA V _{CC} = 4.0–5.5 V			2.0	V
		I _{OL} = 1.0 mA V _{CC} = 2.7–5.5 V			1.0	V
V _{OL}	“L” output voltage P00–P07, P20–P27, P30–P34, P40–P44 (M38514E6/M6)	I _{OL} = 10 mA V _{CC} = 4.0–5.5 V			2.0	V
		I _{OL} = 1.0 mA V _{CC} = 2.7–5.5 V			1.0	V
V _{OL}	“L” output voltage P13–P17 (M38513E4/M4)	I _{OL} = 20 mA V _{CC} = 4.0–5.5 V			2.0	V
		I _{OL} = 10 mA V _{CC} = 2.7–5.5 V			1.0	V
V _{OL}	“L” output voltage P10–P17 (M38514E6/M6)	I _{OL} = 20 mA V _{CC} = 4.0–5.5 V			2.0	V
		I _{OL} = 10 mA V _{CC} = 2.7–5.5 V			1.0	V
V _{T+} –V _{T-}	Hysteresis CNTR0, CNTR1, INT0–INT3			0.4		V
V _{T+} –V _{T-}	Hysteresis RxD, SCLK, SDA1, SDA2, SCL1, SCL2			0.5		V
V _{T+} –V _{T-}	Hysteresis $\overline{\text{RESET}}$			0.5		V
I _{IH}	“H” input current P00–P07, P10–P17, P20, P21, P24–P27, P30–P34, P40–P44	V _I = V _{CC}			5.0	μA
I _{IH}	“H” input current $\overline{\text{RESET}}$, 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 P00–P07, P10–P17, P20–P27 P30–P34, P40–P44	V _I = V _{SS}			-5.0	μA
I _{IL}	“L” input current $\overline{\text{RESET}}$, CNV _{SS}	V _I = V _{SS}			-5.0	μA
I _{IL}	“L” input current X _{IN}	V _I = V _{SS}		-4		μA
V _{RAM}	RAM hold voltage	When clock stopped	2.0		5.5	V

Note: P25 is measured when the P25/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is “0”.

APPENDIX

3.1 Electrical characteristics

Table 3.1.5 Electrical characteristics (2)

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

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

3.1.4 A-D converter characteristics

Table 3.1.6 A-D converter characteristics

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
-	Resolution				10	bit
-	Absolute accuracy (excluding quantization error)				±4	LSB
t _{CONV}	Conversion time	High-speed mode, Middle-speed mode			61	tc(φ)
		Low-speed mode (Note)		40		μs
RLADDER	Ladder resistor			35		kΩ
I _{VREF}	Reference power source input current	V _{REF} = 5.0 V	50	150	200	μA
I _{I(AD)}	A-D port input current			0.5	5.0	μA

Note: Only M38514E6/M6 can operate the A-D conversion at low-speed mode.

3.1.5 Timing requirements

Table 3.1.7 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 (RESET)	Reset input “L” pulse width	2			μs
t _c (XIN)	External clock input cycle time	125			ns
t _{WH} (XIN)	External clock input “H” pulse width	50			ns
t _{WL} (XIN)	External clock input “L” pulse width	50			ns
t _c (CNTR)	CNTR0, CNTR1 input cycle time	200			ns
t _{WH} (CNTR)	CNTR0, CNTR1, INT0–INT3 input “H” pulse width	80			ns
t _{WL} (CNTR)	CNTR0, CNTR1, INT0–INT3 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 setup time	220			ns
t _h (SCLK-RxD)	Serial I/O input hold time	100			ns

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

Table 3.1.8 Timing requirements (2)

(V_{CC} = 2.7 to 5.0 V, V_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t _w (RESET)	Reset input “L” pulse width	2			μs
t _c (XIN)	External clock input cycle time	250			ns
t _{WH} (XIN)	External clock input “H” pulse width	100			ns
t _{WL} (XIN)	External clock input “L” pulse width	100			ns
t _c (CNTR)	CNTR0, CNTR1 input cycle time	500			ns
t _{WH} (CNTR)	CNTR0, CNTR1, INT0–INT3 input “H” pulse width	230			ns
t _{WL} (CNTR)	CNTR0, CNTR1, INT0–INT3 input “L” pulse width	230			ns
t _c (SCLK)	Serial I/O clock input cycle time (Note)	2000			ns
t _{WH} (SCLK)	Serial I/O clock input “H” pulse width (Note)	950			ns
t _{WL} (SCLK)	Serial I/O clock input “L” pulse width (Note)	950			ns
t _{su} (RxD-SCLK)	Serial I/O input setup time	400			ns
t _h (SCLK-RxD)	Serial I/O input hold time	200			ns

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

APPENDIX

3.1 Electrical characteristics

3.1.6 Switching characteristics

Table 3.1.9 Switching characteristics (1)

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t _{WH} (SCLK)	Serial I/O clock output “H” pulse width	Fig. 3.1.1	tc(SCLK)/2–30			ns
t _{WL} (SCLK)	Serial I/O clock output “L” pulse width		tc(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

Notes 1: For t_{WH}(SCLK), t_{WL}(SCLK), when the P25/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is “0”.

2: The XOUT pin is excluded.

Table 3.1.10 Switching characteristics (2)

(V_{CC} = 2.7 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	tc(SCLK)/2–50			ns
t _{WL} (SCLK)	Serial I/O clock output “L” pulse width		tc(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

Notes 1: For t_{WH}(SCLK), t_{WL}(SCLK), when the P25/TxD P-channel output disable bit of the UART control register (bit 4 of address 001B16) is “0”.

2: The XOUT pin is excluded.

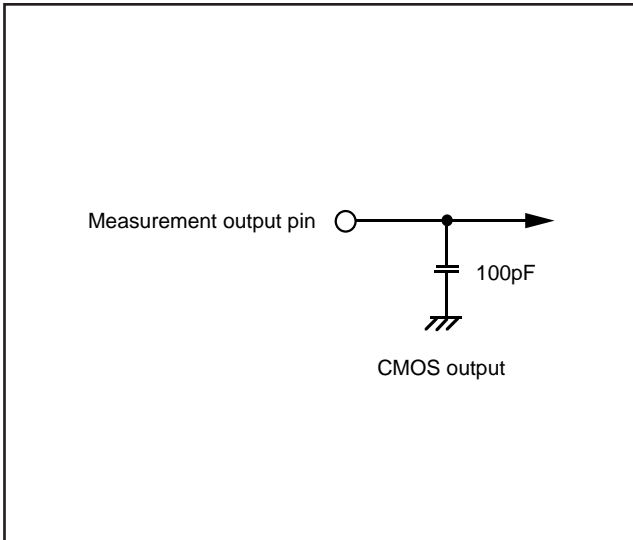


Fig. 3.1.1 Circuit for measuring output switching characteristics (1)

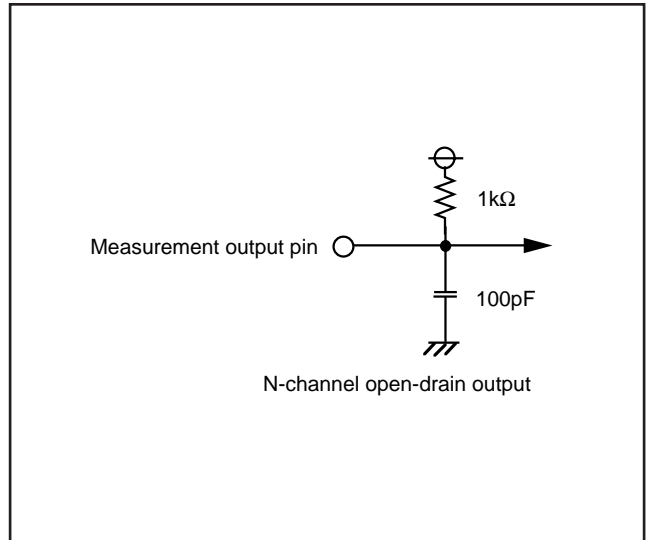


Fig. 3.1.2 Circuit for measuring output switching characteristics (2)

APPENDIX

3.1 Electrical characteristics

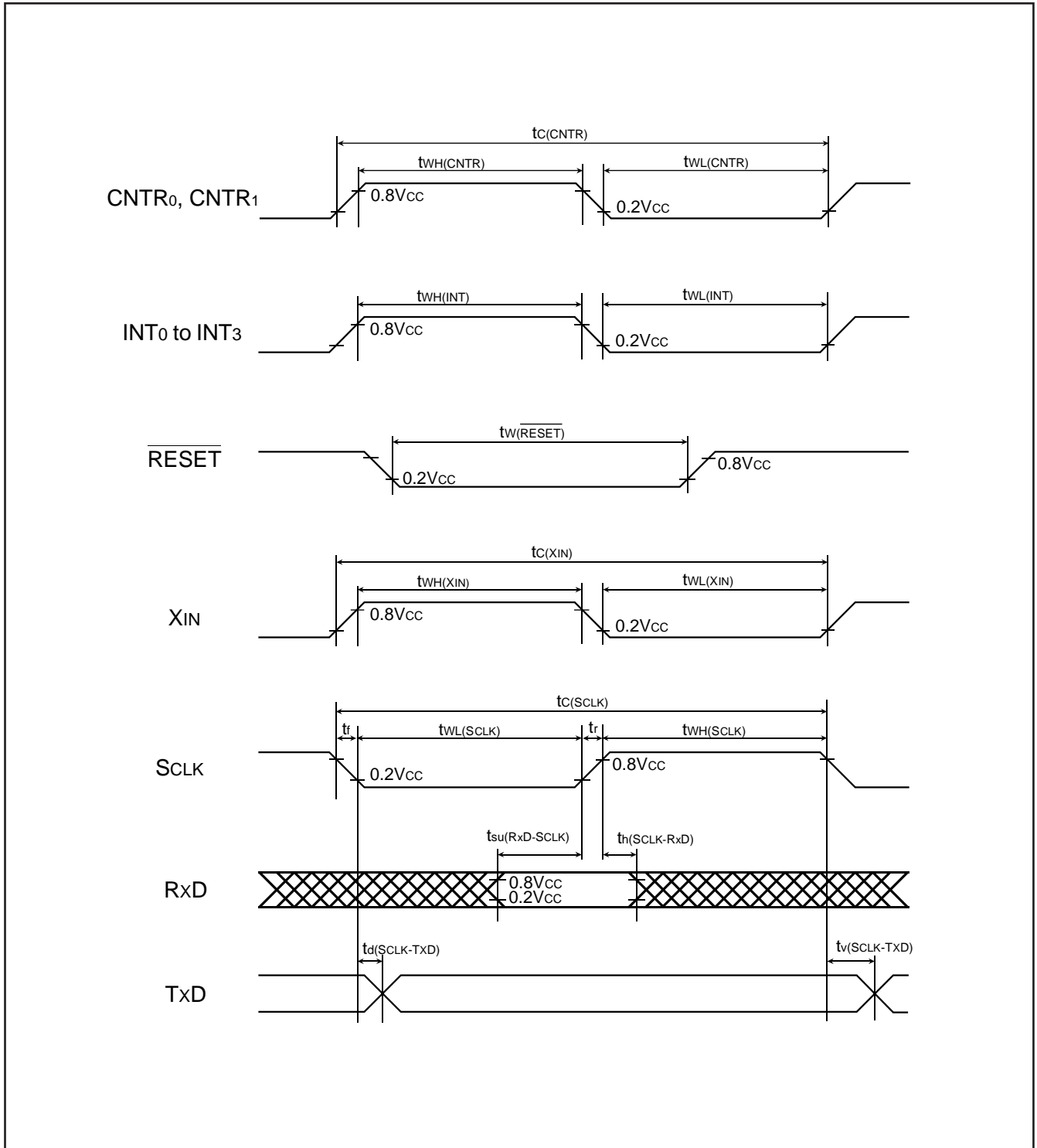


Fig. 3.1.3 Timing diagram

3.1.7 Multi-master I²C-BUS bus line characteristics

Table 3.1.11 Multi-master I²C-BUS bus line characteristics

Symbol	Parameter	Standard clock mode		High-speed clock mode		Unit
		Min.	Max.	Min.	Max.	
t _{BUF}	Bus free time	4.7		1.3		μs
t _{HD;STA}	Hold time for START condition	4.0		0.6		μs
t _{LOW}	Hold time for SCL clock = "0"	4.7		1.3		μs
t _R	Rising time of both SCL and SDA signals		1000	20+0.1C _b	300	ns
t _{HD;DAT}	Data hold time	0		0	0.9	μs
t _{HIGH}	Hold time for SCL clock = "1"	4.0		0.6		μs
t _F	Falling time of both SCL and SDA signals		300	20+0.1C _b	300	ns
t _{SU;DAT}	Data setup time	250		100		ns
t _{SU;STA}	Setup time for repeated START condition	4.7		0.6		μs
t _{SU;STO}	Setup time for STOP condition	4.0		0.6		μs

Note: C_b = total capacitance of 1 bus line

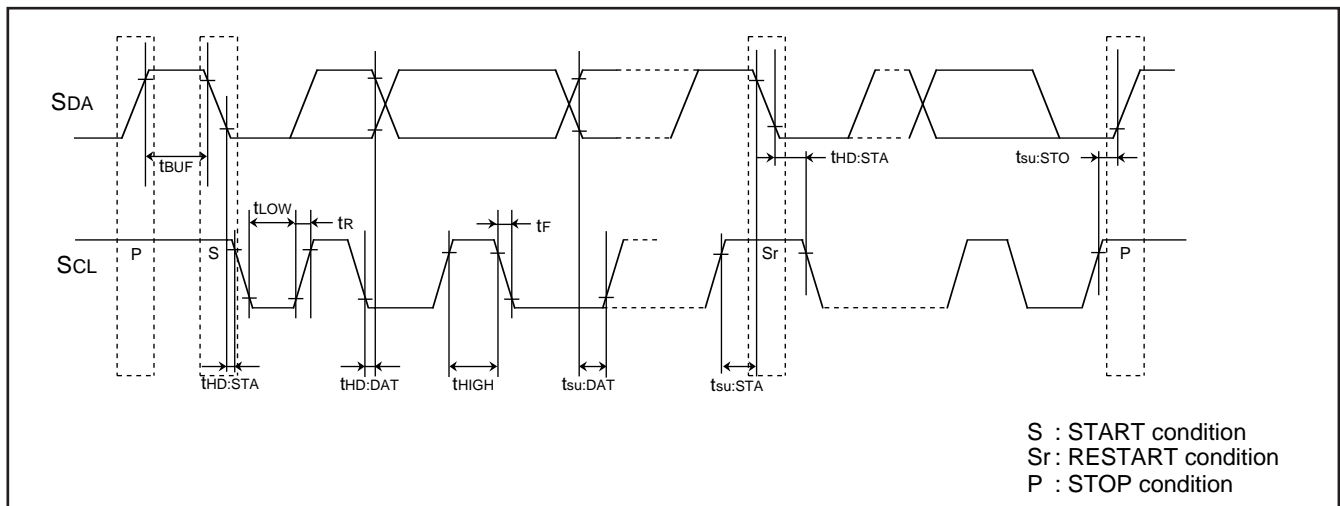


Fig. 3.1.4 Timing diagram of multi-master I²C-BUS

APPENDIX

3.2 Standard characteristics

3.2 Standard characteristics

3.2.1 Power source current characteristic examples

Figures 3.2.1, Figures 3.2.2, Figures 3.2.3, Figures 3.2.3, Figures 3.2.4 and Figures 3.2.5 show power source current characteristic examples.

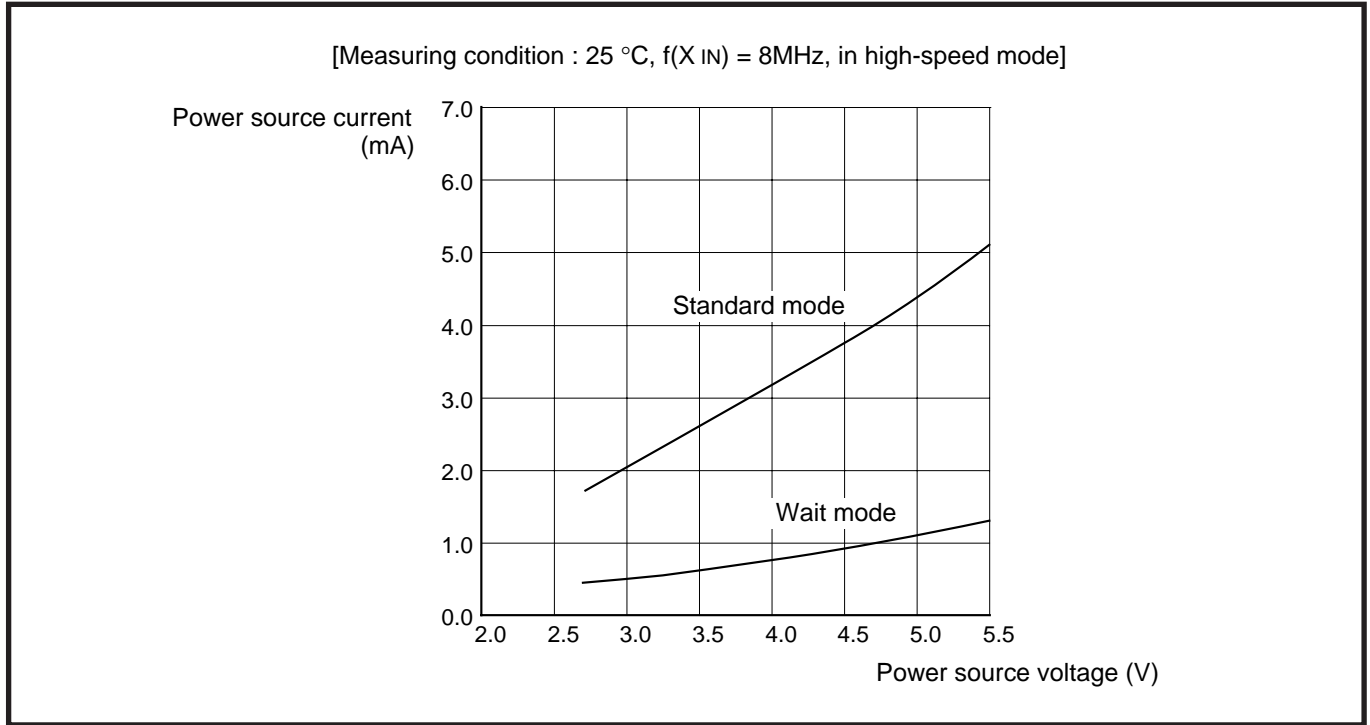


Fig. 3.2.1 Power source current characteristic examples ($f(X_{IN}) = 8\text{MHz}$, in high-speed mode)

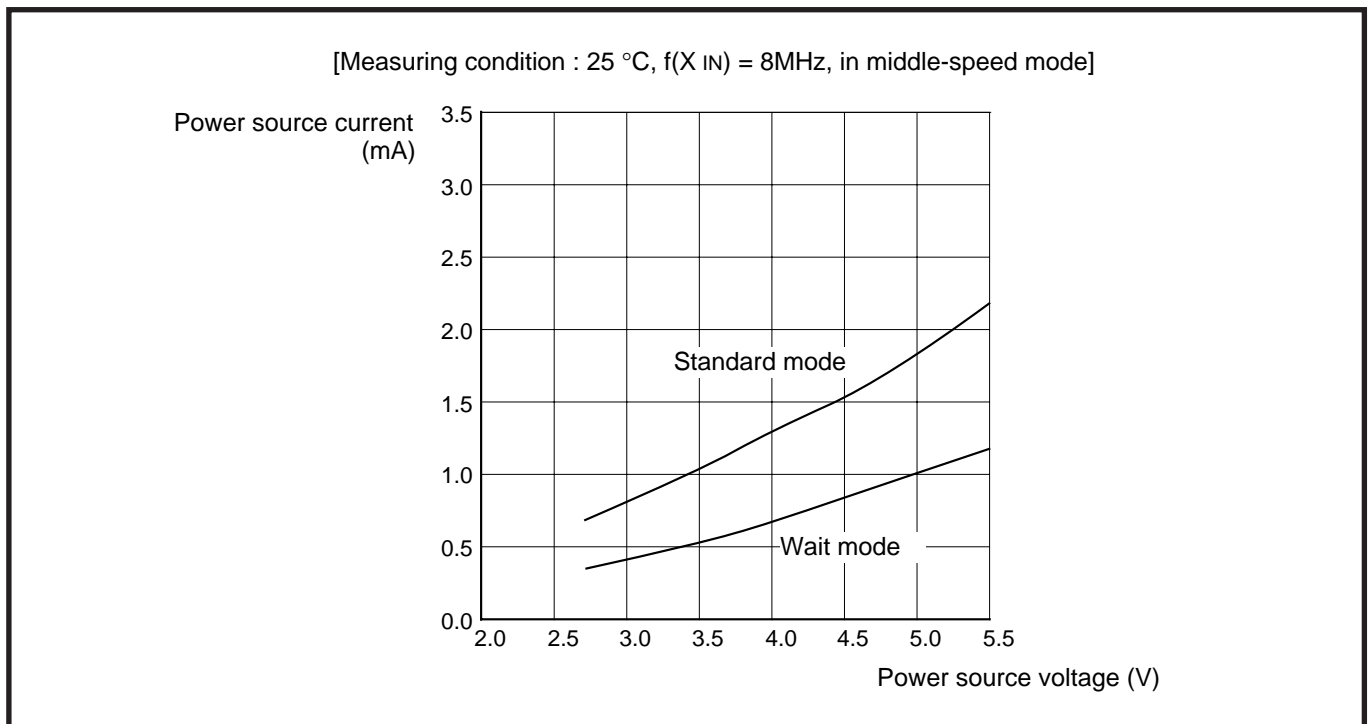


Fig. 3.2.2 Power source current characteristic examples ($f(X_{IN}) = 8\text{MHz}$, in middle-speed mode)

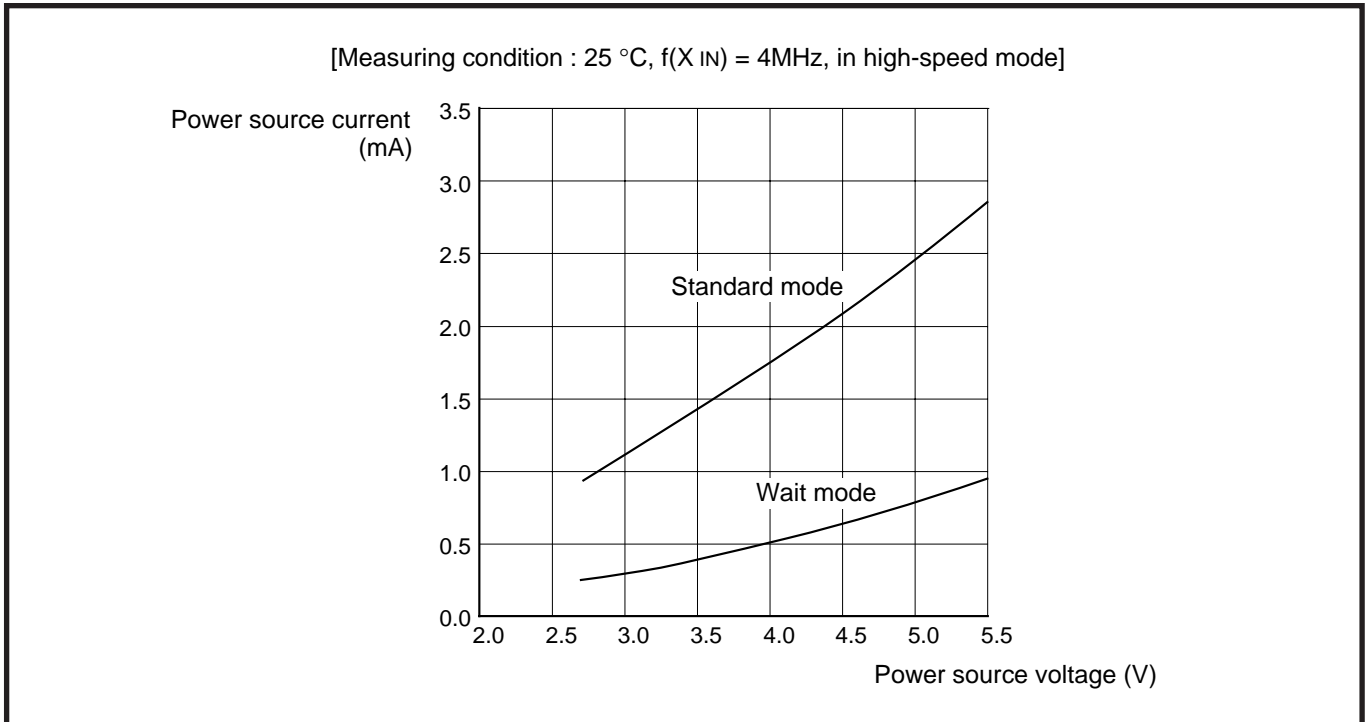


Fig. 3.2.3 Power source current characteristic examples ($f(X_{IN}) = 4\text{MHz}$, in high-speed mode)

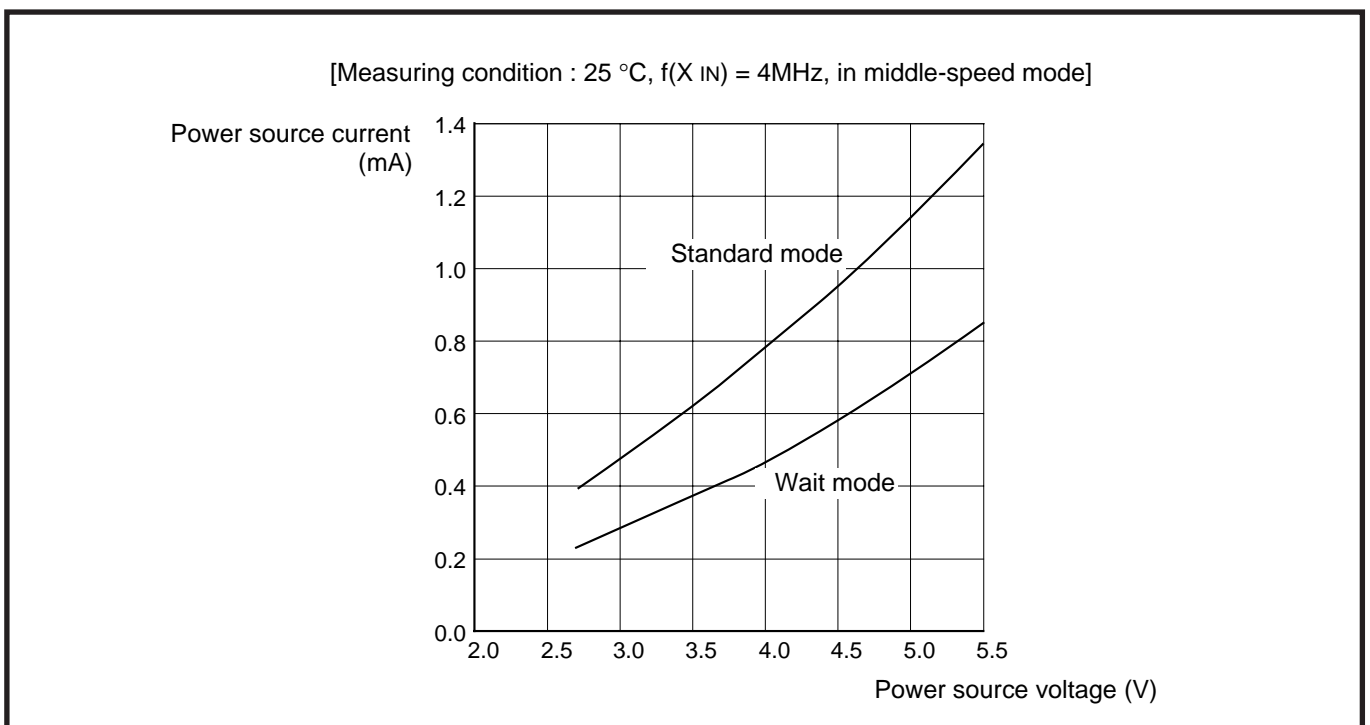


Fig. 3.2.4 Power source current characteristic examples ($f(X_{IN}) = 4\text{MHz}$, in middle-speed mode)

APPENDIX

3.2 Standard characteristics

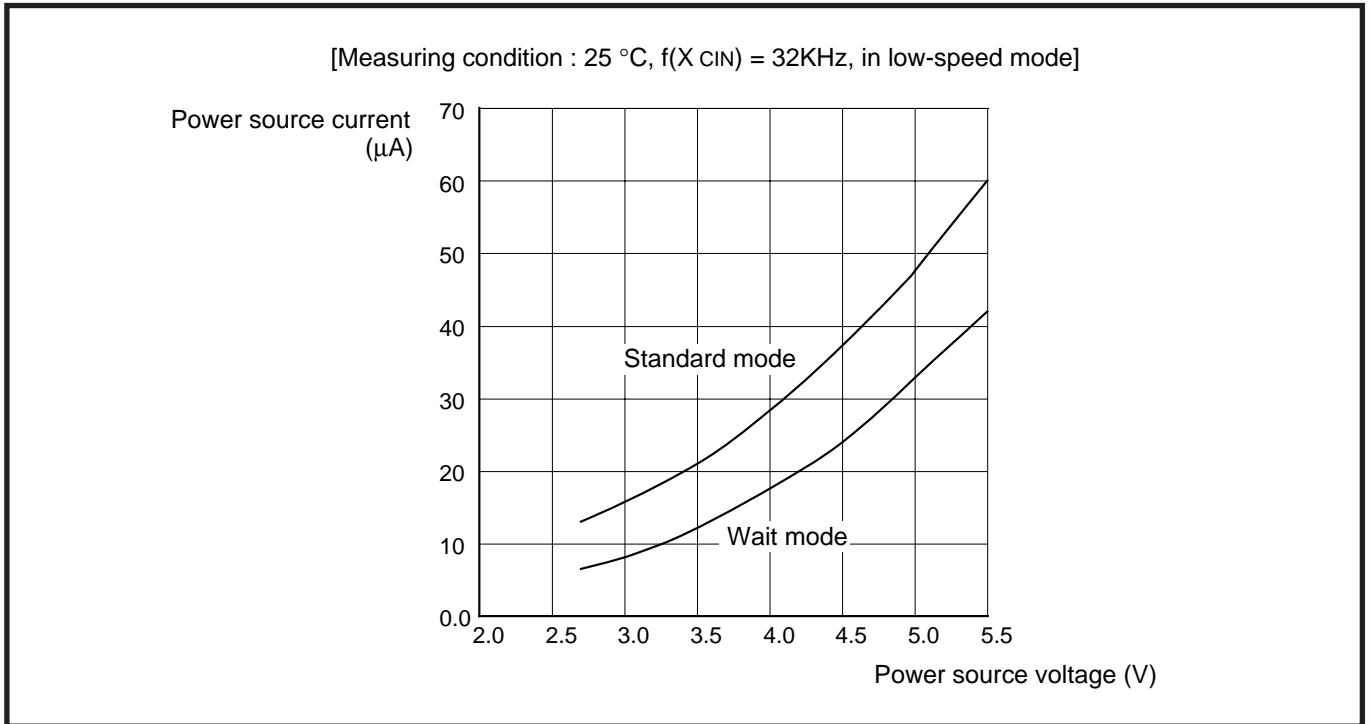


Fig. 3.2.5 Power source current characteristic examples ($f(X_{CIN}) = 32\text{KHz}$, in low-speed mode)

3.2.2 Port standard characteristic examples

Figures 3.2.6, Figures 3.2.7, Figures 3.2.8 and Figures 3.2.9 show port standard characteristic examples.

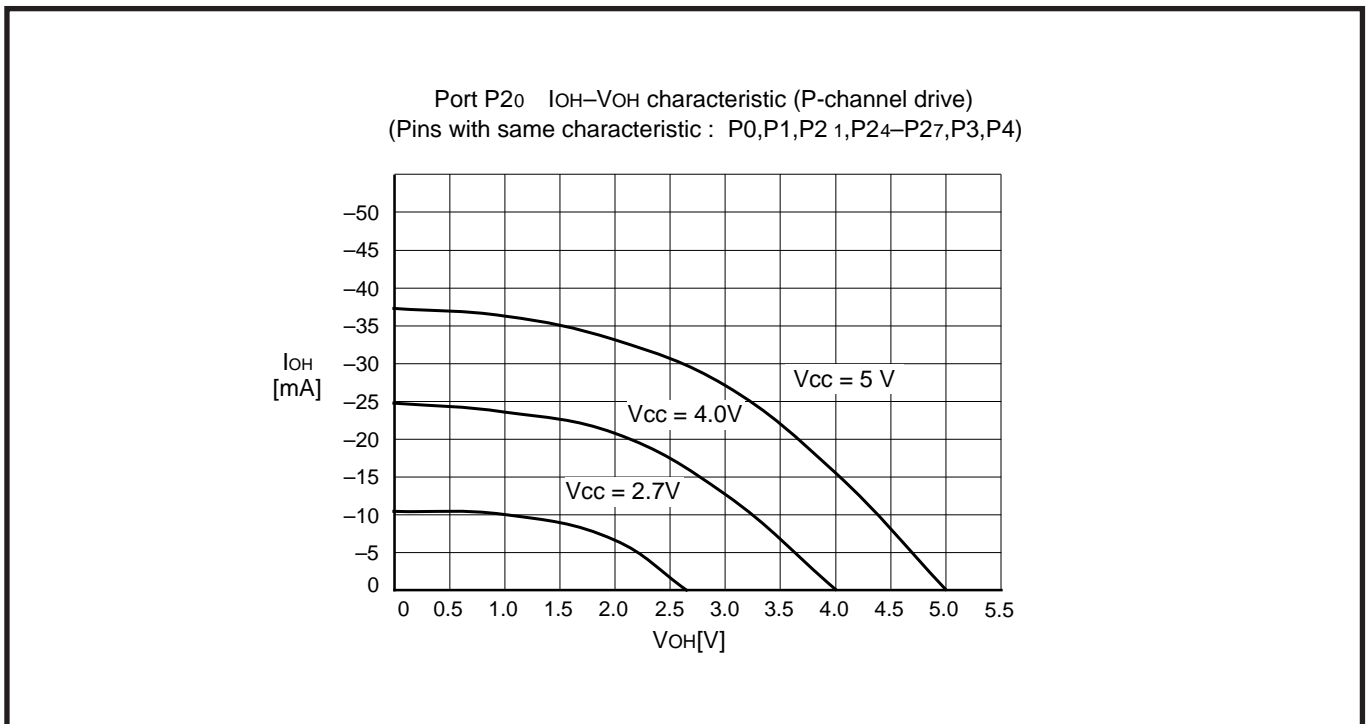


Fig. 3.2.6 Standard characteristic examples of CMOS output port at P-channel drive

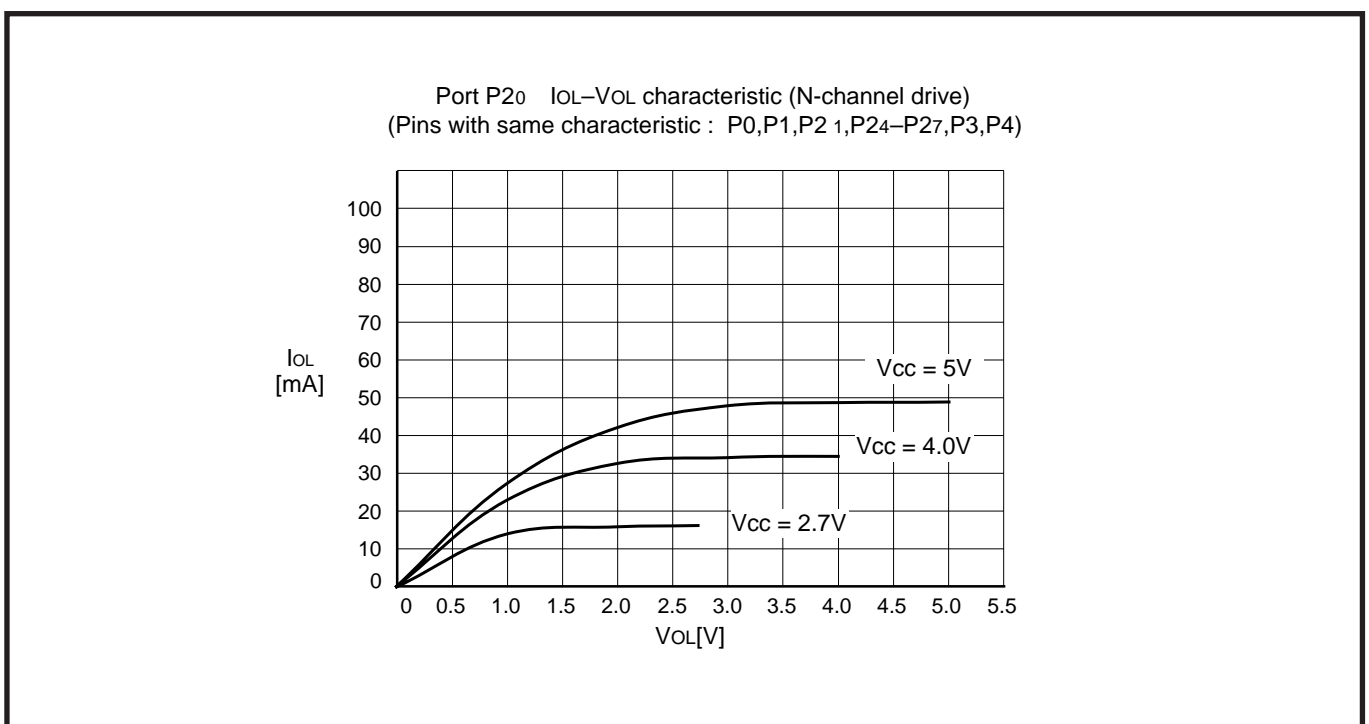


Fig. 3.2.7 Standard characteristic examples of CMOS output port at N-channel drive

APPENDIX

3.2 Standard characteristics

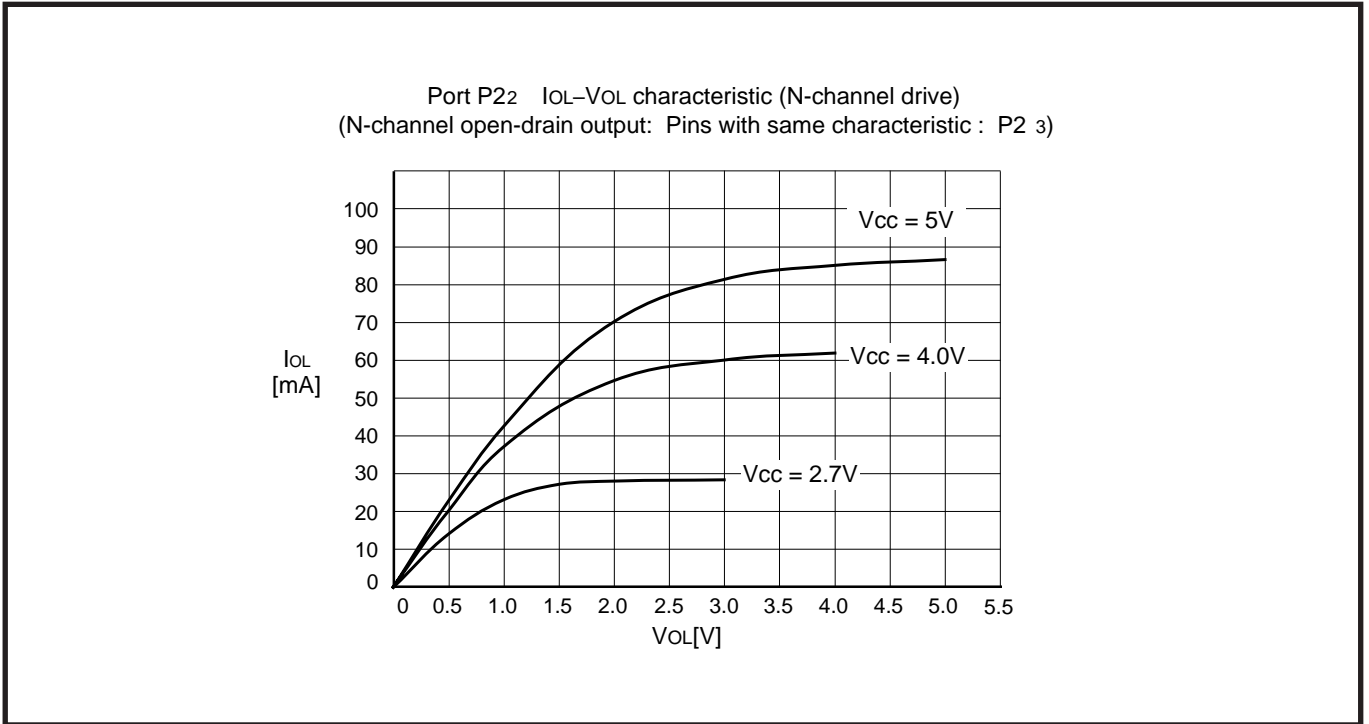


Fig. 3.2.8 Standard characteristic examples of N-channel open-drain output port at N-channel drive

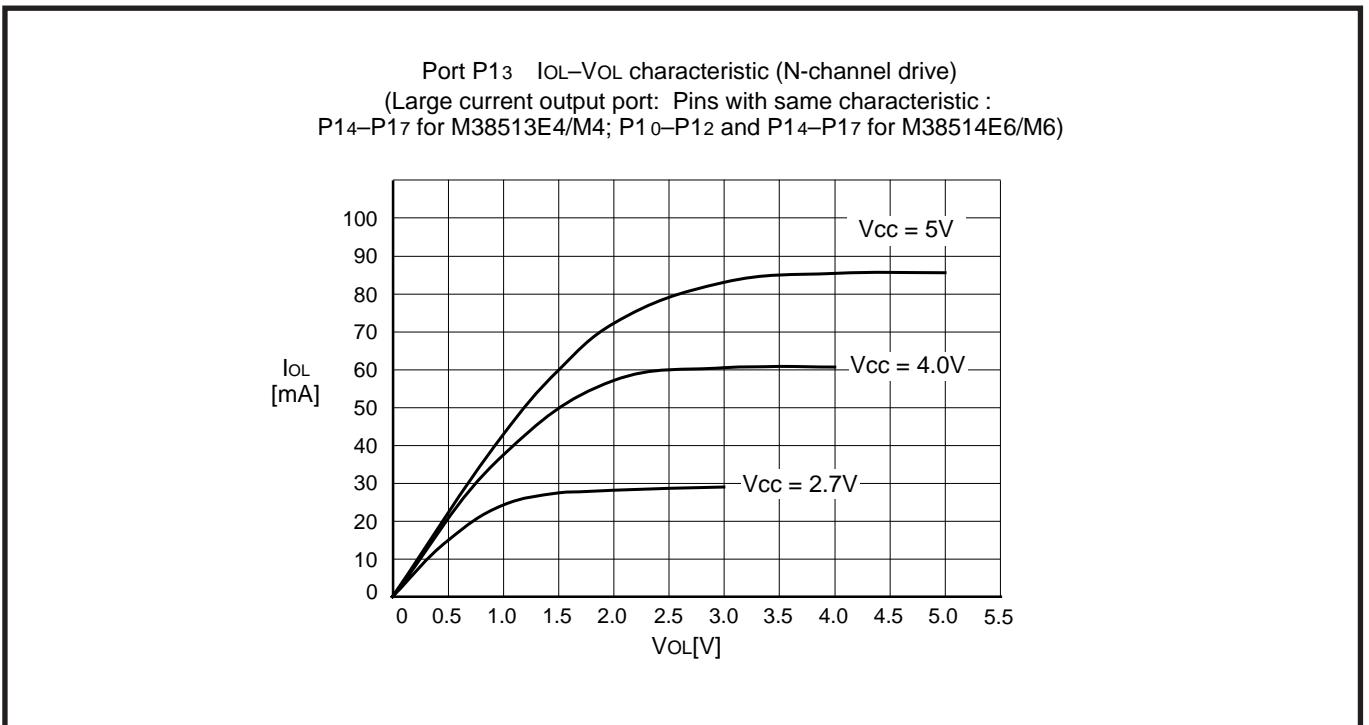


Fig. 3.2.9 Standard characteristic examples of CMOS large current output port at N-channel drive

3.2.3 A-D conversion standard characteristics

Figure 3.2.10 shows the A-D conversion standard characteristics.

The lower-side line on the graph indicates the absolute precision error. It represents the deviation from the ideal value. For example, the conversion of output code from 0 to 1 occurs ideally at the point of $AN_0 = 2.5 \text{ mV}$, but the measured value is -4 mV . Accordingly, the measured point of conversion is represented as " $2.5 - 4 = -1.5 \text{ mV}$."

The upper-side line on the graph indicates the width of input voltages equivalent to output codes. For example, the measured width of the input voltage for output code 96 is 5 mV , so the differential nonlinear error is represented as " $5 - 5 = 0 \text{ mV}$ " (0 LSB).

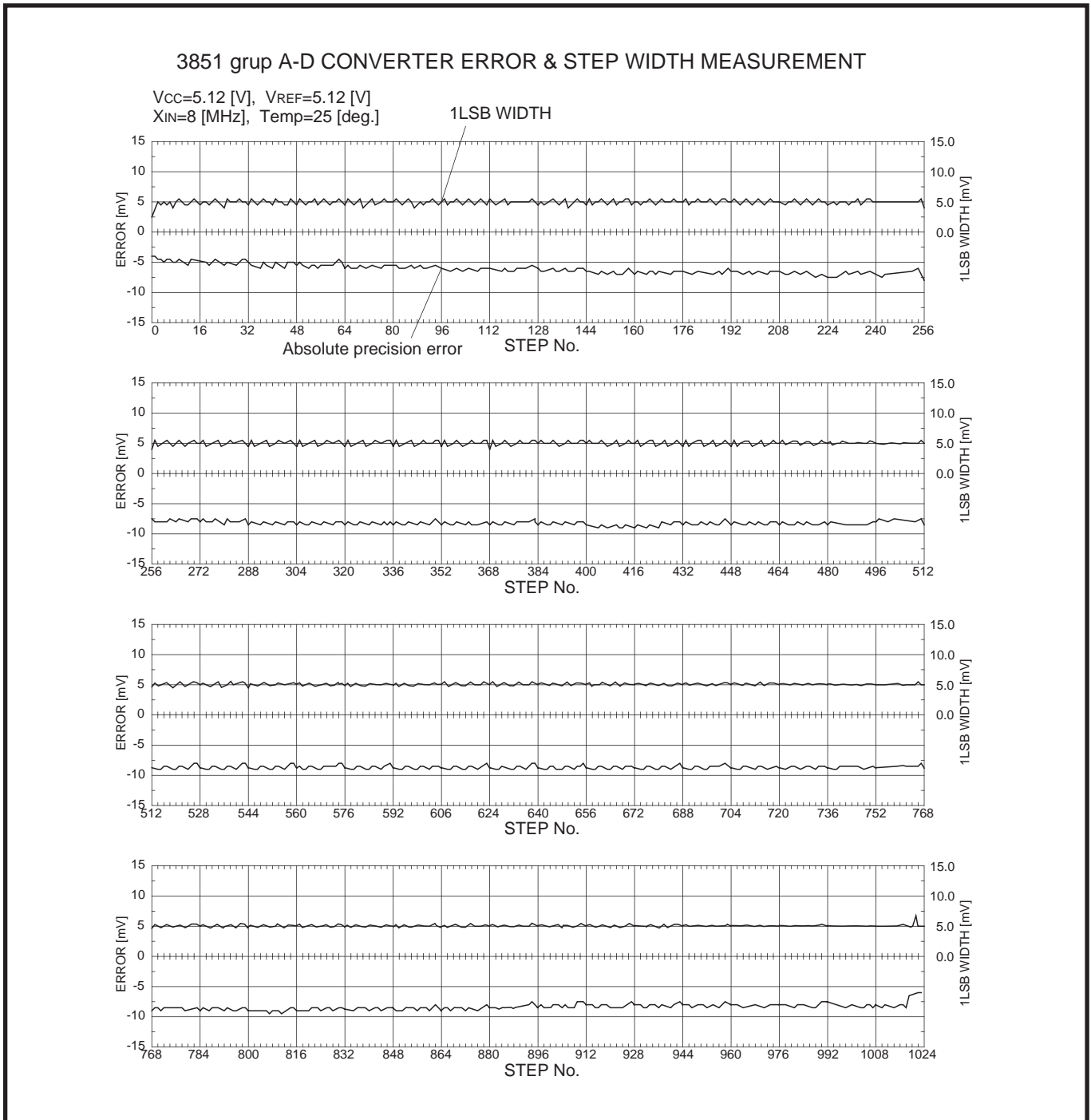


Fig. 3.2.10 A-D conversion standard characteristics

APPENDIX

3.3 Notes on use

3.3 Notes on use

3.3.1 Notes on interrupts

(1) Setting of interrupt request bit and interrupt enable bit

To set an interrupt request bit and an interrupt enable bit for interrupts, execute as the following sequence :

- ① Clear an interrupt request bit to "0" (no interrupt request issued).
- ② Set an interrupt enable bit to "1" (interrupts enabled).

● Reason

If the above setting ①, ② are performed simultaneously with one instruction, an unnecessary interrupt processing routine is executed. Because an interrupt enable bit is set to "1" (interrupts enabled) before an interrupt request bit is cleared to "0".

(2) Switching external interrupt detection edge

When switching the external interrupt detection edge, switch it as the following sequence.

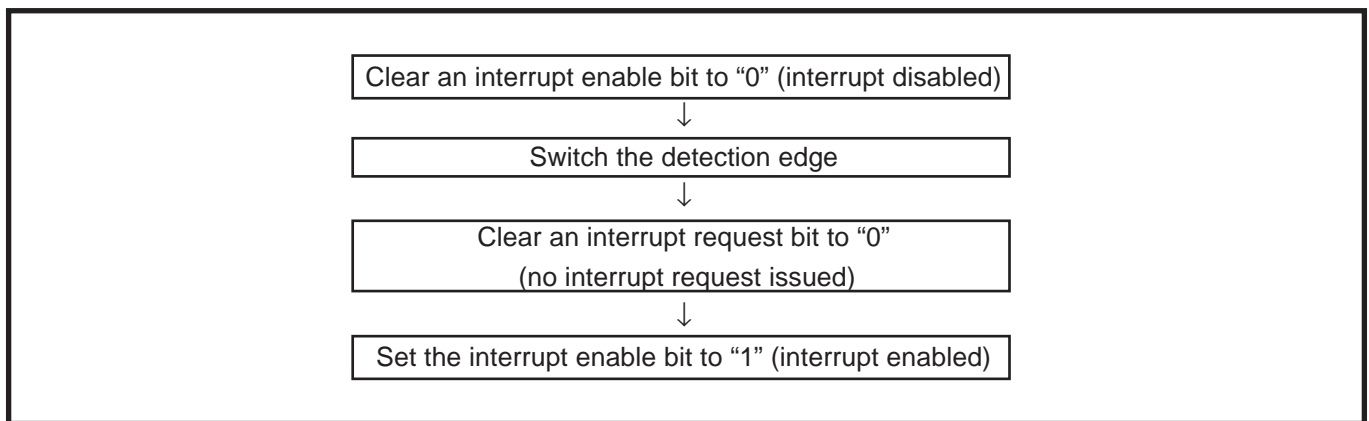


Fig. 3.3.1 Sequence of switch the detection edge

● Reason

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

(3) Check of interrupt request bit

When executing the **BBC** or **BBS** instruction to an interrupt request bit of an interrupt request register immediately after this bit is set to "0" by using a data transfer instruction, execute one or more instructions before executing the **BBC** or **BBS** instruction.

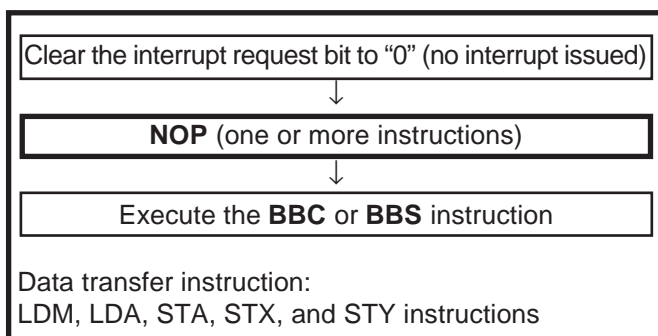


Fig. 3.3.2 Sequence of check of interrupt request bit

● Reason

If the **BBC** or **BBS** instruction is executed immediately after an interrupt request bit of an interrupt request register is cleared to "0", the value of the interrupt request bit before being cleared to "0" is read.

3.3.2 Notes on timer

- If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is $1/(n+1)$.
- When switching the count source by the timer 12, X and Y count source selection bit, the value of timer count is altered in unconsiderable amount owing to generating of a thin pulses in the count input signals.
Therefore, select the timer count source before set the value to the prescaler and the timer.

3.3.3 Notes on serial I/O

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

① Stop of transmission operation

Clear the serial I/O enable bit and the transmit enable bit to "0" (serial I/O and transmit disabled).

● Reason

Since transmission is not stopped and the transmission circuit is not initialized even if only 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, data starts to be shifted to the transmit shift register. 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 an operation failure occurs.

② Stop of receive operation

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

③ Stop of transmit/receive operation

Clear both the transmit enable bit and receive enable bit to "0" (transmit and receive disabled) at the same time.

(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) ①).

APPENDIX

3.3 Notes on use

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

① **Stop of transmission operation**

Clear the transmit enable bit to “0” (transmit disabled).

● **Reason**

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/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, data starts to be shifted to the transmit shift register. 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 an operation failure occurs.

② **Stop of receive operation**

Clear the receive enable bit to “0” (receive disabled).

③ **Stop of transmit/receive operation**

Only transmission operation is stopped.

Clear the transmit enable bit to “0” (transmit disabled).

● **Reason**

Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/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, data starts to be shifted to the transmit shift register. 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 an operation failure occurs.

Only receive operation is stopped.

Clear the receive enable bit to “0” (receive disabled).

(3) $\overline{\text{SRDY}}$ output of reception 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).

(4) Setting serial I/O control register again

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.”

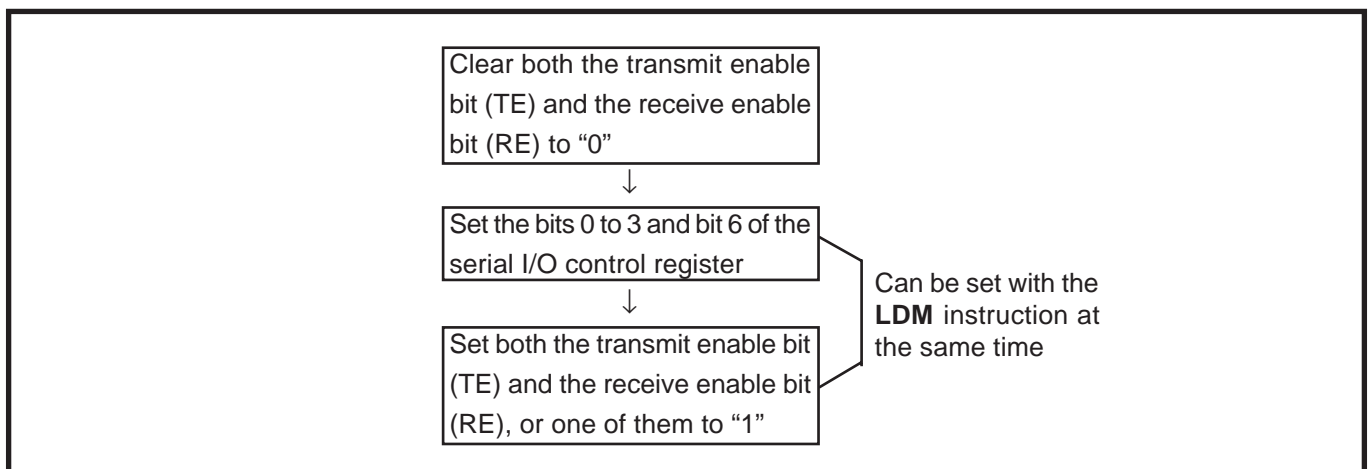


Fig. 3.3.3 Sequence of setting serial I/O control register again

(5) Data transmission control with referring to transmit shift register completion flag

The transmit shift register completion flag changes from “1” to “0” with a delay of 0.5 to 1.5 shift clocks. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

(6) Transmission control when external clock is selected

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

(7) Transmit interrupt request when transmit enable bit is set

The transmission interrupt request bit is set and the interruption request is generated even when selecting timing that either of the following flags is set to “1” as timing where the transmission interruption is generated.

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

Therefore, when the transmit interrupt is used, set the transmit interrupt enable bit to transmit enabled as the following sequence.

- ① Transmit enable bit is set to “1”
- ② Transmit interrupt request bit is set to “0”

● Reason

When the transmission enable bit is set to “1”, the transmit buffer empty flag and transmit shift register completion flag are set to “1”.

3.3.4 Notes on multi-master I²C-BUS interface**(1) Read-modify-write instruction**

Each register of the multi-master I²C-BUS interface has bits to change by hardware. The precautions when the read-modify-write instruction such as SEB, CLB etc. is executed for each register of the multi-master I²C-BUS interface are described below.

① I²C data shift register (S0: address 002B₁₆)

When executing the read-modify-write instruction for this register during transfer, data may become a value not intended.

② I²C address register (S0D: address 002C₁₆)

When the read-modify-write instruction is executed for this register at detecting the STOP condition, data may become a value not intended.

● Reason

It is because hardware changes the read/write bit (RBW) at detecting the STOP condition.

③ I²C status register (S1: address 002D₁₆)

Do not execute the read-modify-write instruction for this register because all bits of this register are changed by hardware.

④ I²C control register (S1D: address 002E₁₆)

When the read-modify-write instruction is executed for this register at detecting the START condition or at completing the byte transfer, data may become a value not intended.

● Reason

Because hardware changes the bit counter (BC0 to BC2).

⑤ I²C clock control register (S2: address 002F₁₆)

The read-modify-write instruction can be executed for this register.

⑥ I²C START/STOP condition control register (S2D: address 0030₁₆)

The read-modify-write instruction can be executed for this register.

APPENDIX

3.3 Notes on use

(2) START condition generating procedure using multi-master

① Procedure example (The necessary conditions of the generating procedure are described as the following ② to ⑤).

```
LDA #SLADR          (Taking out of slave address value)
SEI                 (Interrupt disabled)
BBS 5, S1, BUSBUSY (BB flag confirming and branch process)
BUSFREE:
STA S0              (Writing of slave address value)
LDM #$F0, S1       (Trigger of START condition generating)
CLI                 (Interrupt enabled)
:
:
BUSBUSY:
CLI                 (Interrupt enabled)
:
:
```

- ② Use “Branch on Bit Set” of “BBS 5, S1, –” for the BB flag confirming and branch process.
- ③ Use “STA, STX” or “STY” of the zero page addressing instruction for writing the slave address value to the I²C data shift register (S0: address 002B₁₆).
- ④ Execute the branch instruction of above 2 and the store instruction of above 3 continuously shown the above procedure example.
- ⑤ Disable interrupts during the following three process steps:
 - BB flag confirming
 - Writing of slave address value
 - Trigger of START condition generating

(3) RESTART condition generating procedure in master

① Procedure example (The necessary conditions of the generating procedure are described as the following ② to ④). Execute the following procedure when the PIN bit is “0”.

```
LDM #$00, S1       (Select slave receive mode)
LDA #SLADR          (Taking out of slave address value)
SEI                 (Interrupt disabled)
STA S0              (Writing of slave address value)
LDM #$F0, S1       (Trigger of RESTART condition generating)
CLI                 (Interrupt enabled)
:
:
```

- ② Select the slave receive mode when the PIN bit is “0”. Do not write “1” to the PIN bit. Neither “0” nor “1” is specified for the writing to the BB bit. The TRX bit becomes “0” and the SDA pin is released.
- ③ The SCL pin is released by writing the slave address value to the I²C data shift register.
- ④ Disable interrupts during the following two process steps:
 - Writing of slave address value
 - Trigger of RESTART condition generating

(4) Writing to I²C status register

Do not execute an instruction to set the PIN bit to “1” from “0” and an instruction to set the MST and TRX bits to “0” from “1” simultaneously. It is because it may enter the state that the SCL pin is released and the SDA pin is released after about one machine cycle. Do not execute an instruction to set the MST and TRX bits to “0” from “1” simultaneously when the PIN bit is “1”. It is because it may become the same as above.

(5) STOP condition generating procedure in master

① Procedure example (The necessary conditions of the generating procedure are described as the following ② to ⑤).

SEI	(Interrupt disabled)
LDM #\$C0, S1	(Select master transmit mode)
NOP	(Set NOP)
LDM #\$D0, S1	(Trigger of STOP condition generating)
CLI	(Interrupt enabled)
⋮	
⋮	

② When setting the master transmit mode, write "0" to the PIN bit.

③ Execute the NOP instruction after the master transmit mode is set. In addition, set the STOP condition to be triggered within 10 machine cycles after the master transmit mode has been set.

④ Make sure all interrupts are disabled during the term from when the master transmit mode is set until the triggering process, which generates the STOP condition, is complete.

⑤ The above procedure is only applicable to the M38513E4.

(6) Process of after STOP condition generating

Do not write data in the I²C data shift register S0 and the I²C status register S1 until the bus busy flag BB becomes "0" after generating the STOP condition in the master mode. It is because the STOP condition waveform might not be normally generated. Reading to the above registers do not have the problem.

(7) STOP condition input at 7th clock pulse

In the slave mode, the STOP condition is input at the 7th clock pulse while receiving a slave address or data. As the clock pulse is continuously input, the SDA line may be held at LOW even if flag BB is set to "0".

Countermeasure:

Write dummy data to the I²C shift register or reset the ES0 bit in the S1D register (ES0 = "L" → ES0 = "H") during a stop condition interrupt routine with flag PIN = "1".

Notes 1: Do not use the read-modify-write instruction at this time. Furthermore, when the ES0 bit is set to "0", it becomes a general-purpose port ; so that the port must be set to input mode or "H".

2: The M38514E6/M6 does not have this problem which is the SDA line remaining "L".

(8) ES0 bit switch

In standard clock mode when SSC = "00010₂" or in high-speed clock mode, flag BB may switch to "1" if ES0 bit is set to "1" when SDA is "L".

Countermeasure:

Set ES0 to "1" when SDA is "H".

APPENDIX

3.3 Notes on use

3.3.5 Notes on A-D converter

(1) Analog input pin

Make the signal source impedance for analog input low, or equip an analog input pin with an external capacitor of 0.01 μ F to 1 μ F. Further, be sure to verify the operation of application products on the user side.

● Reason

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

(2) A-D converter power source pin

Pins AVCC and AVSS are A-D converter power source pins. Regardless of using the A-D conversion function or not, connect them as following :

- AVCC : Connect to the VCC line
- AVSS : Connect to the VSS line

● Reason

If the AVCC and the AVSS pin are opened, the microcomputer may have a failure because of noise or others. Also, if the AVCC pin is connected to the VSS pin, current flows from AVCC to VSS.

(3) Clock frequency during A-D conversion

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

- f(XIN) is 500 kHz or more
- Do not execute the **STP** instruction and **WIT** instruction

3.3.6 Notes on watchdog timer

- The watchdog timer continues counting even while waiting for the stop release. Make sure the watchdog timer does not underflow during this term.
- Once the **STP** instruction inhibit bit of the watchdog timer control register is set to "1", the bit can not be reprogrammed to "0".

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

(1) Connecting capacitor

In case where the $\overline{\text{RESET}}$ signal rise time is long, connect a ceramic capacitor or others across the $\overline{\text{RESET}}$ pin and the VSS pin. And use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, note the following :

- Make the length of the wiring which is connected to a capacitor as short as possible.
- Be sure to verify the operation of application products on the user side.

● Reason

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

3.3.8 Notes on input and output pins

(1) Notes in stand-by state

In stand-by state*¹ for low-power dissipation, do not make input levels of an input port and an I/O port “undefined”, especially for I/O ports of the P-channel and the N-channel open-drain.

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

When determining a resistance value, note the following points:

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

When using built-in pull-up or pull-down resistor, note on varied current values:

- When setting as an input port : Fix its input level
- When setting as an output port : Prevent current from flowing out to external

● Reason

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

- P-channel.....when the content of the port latch is “0”
- 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.

Note 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 a input port and an I/O port are “undefined”. This may cause power source current.

*¹ stand-by state : the stop mode by executing the **STP** instruction
the wait mode by executing the **WIT** instruction

(2) Modifying output data with bit managing instruction

When the port latch of an I/O port is modified with the bit managing instruction*², the value of the unspecified bit may be changed.

● Reason

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

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

Note the following :

- Even when a port which is set as an output port is changed for an input port, its port latch holds the output data.
- As for a bit of the port latch which is set for an input port, its value may be changed even when not specified with a bit managing instruction in case where the pin state differs from its port latch contents.

*² bit managing instructions : **SEB**, and **CLB** instructions

APPENDIX

3.3 Notes on use

3.3.9 Notes on low-speed operation mode

(1) Using sub-clock

To use a sub-clock, fix the bit 3 of the CPU mode register to “1” (XCOOUT drive capacity is “High”) and control the R_d (refer to Figure 3.3.4) resistance value to a certain level to stabilize an oscillation. For resistance value of R_d , consult the oscillator manufacturer.

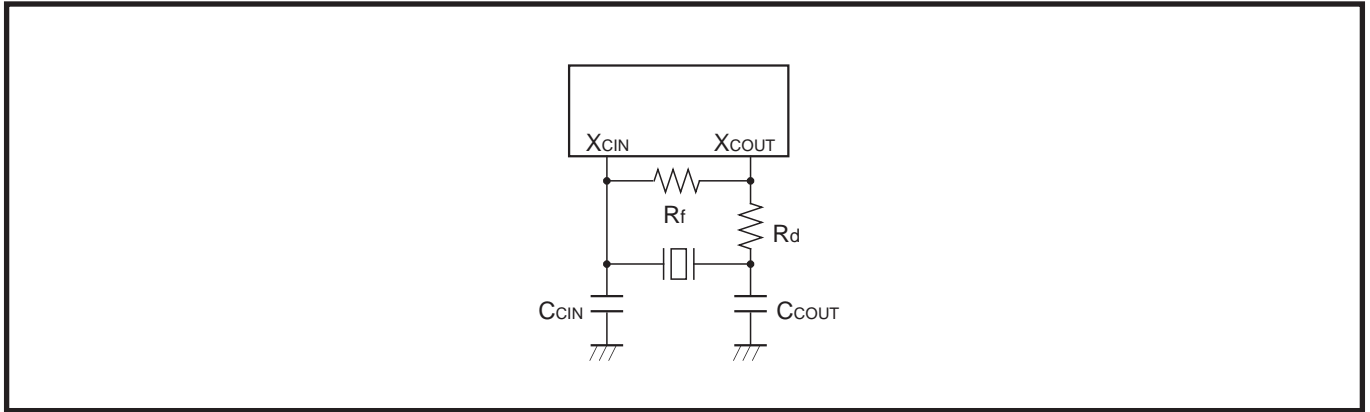


Fig. 3.3.4 Ceramic resonator circuit

● Reason

When the bit 3 of CPU mode register is set to “0”, the sub-clock oscillation may stop.

3.3.10 Notes on restarting oscillation

(1) Restarting oscillation

Usually, when the MCU stops the clock oscillation by STP instruction and the STP instruction has been released by an external interrupt source, the fixed values of Timer 1 and Prescaler 12 (Timer 1 = 01_{16} , Prescaler 12 = FF_{16}) are automatically reloaded in order for the oscillation to stabilize. The user can inhibit the automatic setting by writing “1” to bit 0 of MISRG (address 0038_{16}).

However, by setting this bit to “1”, the previous values, set just before the STP instruction was executed, will remain in Timer 1 and Prescaler 12. Therefore, you will need to set an appropriate value to each register, in accordance with the oscillation stabilizing time, before executing the STP instruction.

● Reason

Oscillation will restart when an external interrupt is received. However, internal clock phi is supplied to the CPU only when Timer 1 starts to underflow. This ensures time for the clock oscillation using the ceramic resonators to be stabilized.

3.3.11 Notes on programming

(1) Processor status register

① Initializing of processor status register

Flags which affect program execution must be initialized after a reset.

In particular, it is essential to initialize the T and D flags because they have an important effect on calculations.

● Reason

After a reset, the contents of the processor status register (PS) are undefined except for the I flag which is "1".

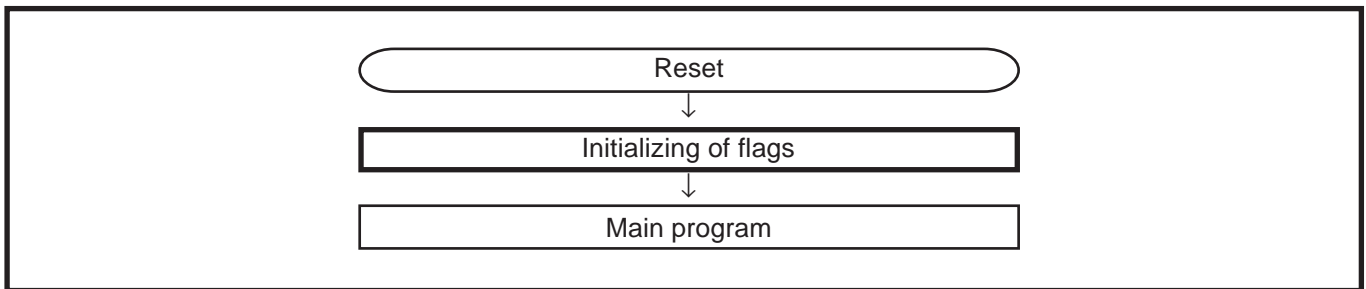


Fig. 3.3.5 Initialization of processor status register

② How to reference the processor status register

To reference the contents of the processor status register (PS), execute the **PHP** instruction once then read the contents of (S+1). If necessary, execute the **PLP** instruction to return the PS to its original status.

A **NOP** instruction should be executed after every **PLP** instruction.

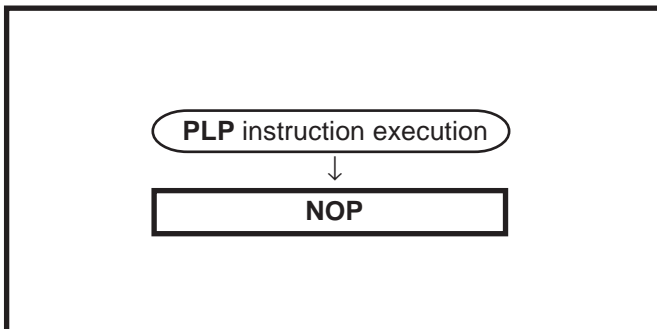


Fig. 3.3.6 Sequence of PLP instruction execution

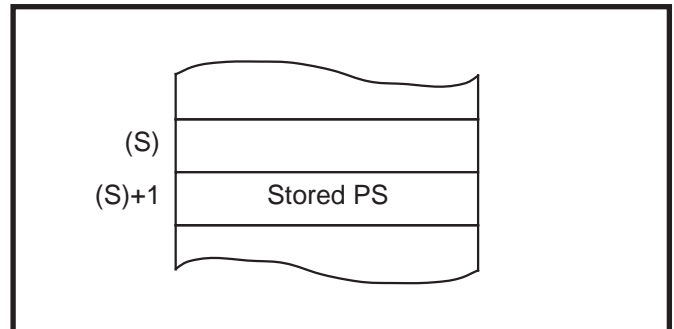


Fig. 3.3.7 Stack memory contents after PHP instruction execution

APPENDIX

3.3 Notes on use

(2) BRK instruction

① Detection of interrupt source

It can be detected that the **BRK** instruction interrupt event or the least priority interrupt event by referring the stored B flag state. Refer to the stored B flag state in the interrupt routine.

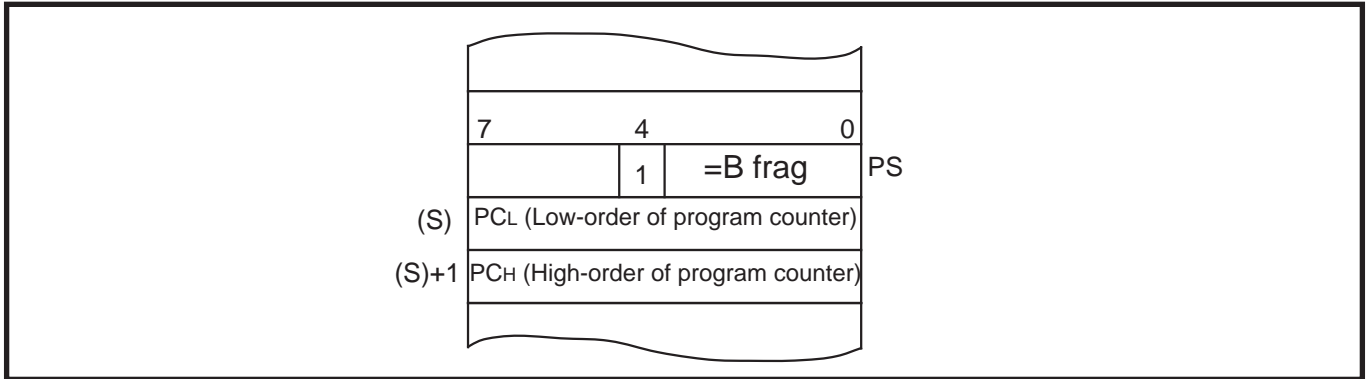


Fig. 3.3.8 Interrupt routine

② Interrupt priority level

When the **BRK** instruction is executed with the following conditions satisfied, the interrupt execution is started from the address of interrupt vector which has the highest priority.

- Interrupt request bit and interrupt enable bit are set to "1".
- Interrupt disable flag (I) is set to "1" to disable interrupt.

(3) Decimal calculations

① Execution of decimal calculations

The **ADC** and **SBC** are the only instructions which will yield proper decimal notation, set the decimal mode flag (D) to "1" with the **SED** instruction. After executing the **ADC** or **SBC** instruction, execute another instruction before executing the **SEC**, **CLC**, or **CLD** instruction.

② Notes on status flag in decimal mode

When decimal mode is selected, the values of three of the flags in the status register (the N, V, and Z flags) are invalid after a **ADC** or **SBC** instruction is executed.

The carry flag (C) is set to "1" if a carry is generated as a result of the calculation, or is cleared to "0" if a borrow is generated. To determine whether a calculation has generated a carry, the C flag must be initialized to "0" before each calculation. To check for a borrow, the C flag must be initialized to "1" before each calculation.

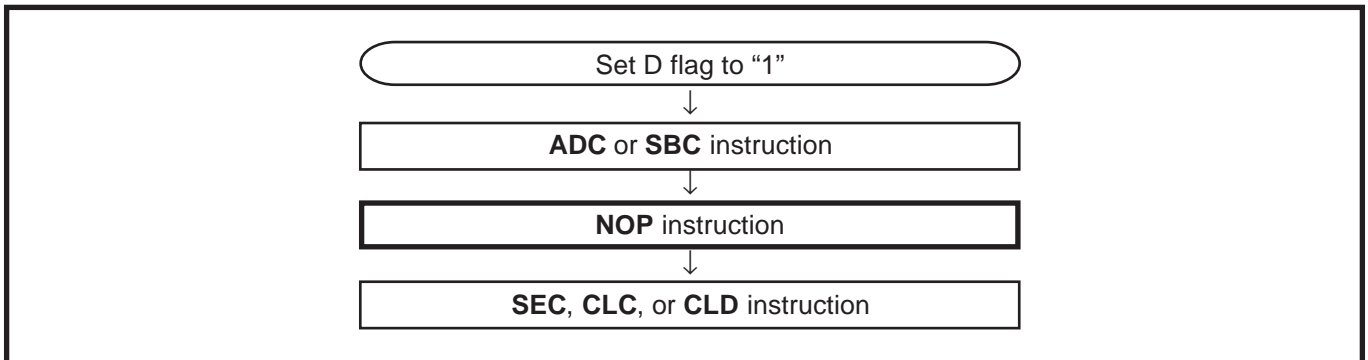


Fig. 3.3.9 Status flag at decimal calculations

(4) JMP instruction

When using the **JMP** instruction in indirect addressing mode, do not specify the last address on a page as an indirect address.

3.3.12 Programming and test of built-in PROM version

As for in the One Time PROM version (shipped in blank) and the built-in EPROM version, their built-in PROM can be read or programmed with a general-purpose PROM programmer using a special programming adapter.

The built-in EPROM version is available only for program development and on-chip program evaluation. The programming test and screening for PROM of the One Time PROM version (shipped in blank) are not performed in the assembly process and the following processes. To ensure reliability after programming, performing programming and test according to the Figure 3.3.10 before actual use are recommended.

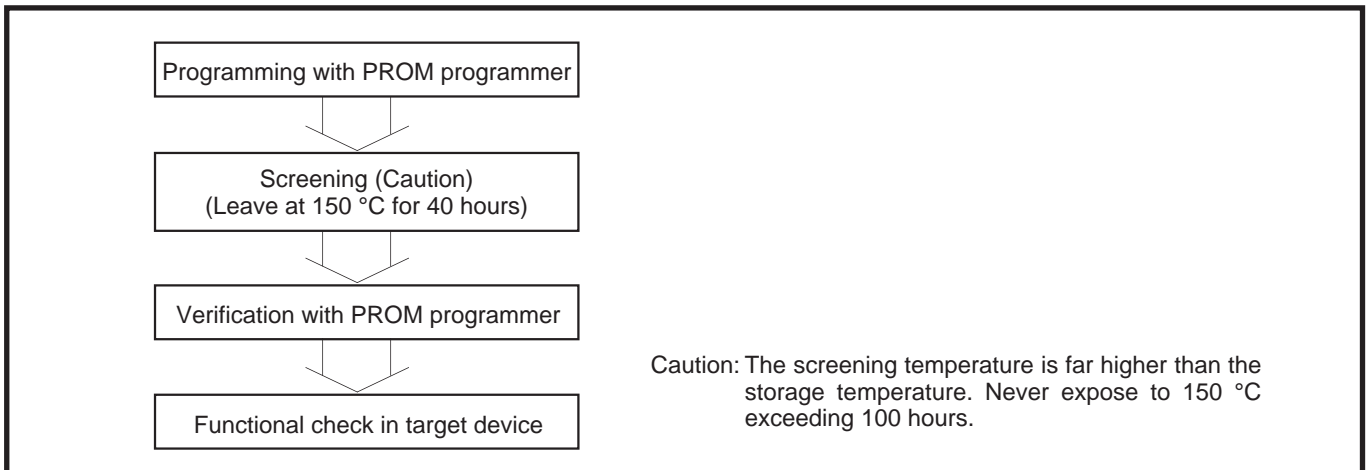


Fig. 3.3.10 Programming and testing of One Time PROM version

3.3.13 Notes on built-in PROM version

(1) Programming adapter

Use a special programming adapter shown in Table 3.3.2 and a general-purpose PROM programmer when reading from or programming to the built-in PROM in the built-in PROM version.

Table 3.3.1 Programming adapters

Microcomputer	Programming adapter
M38513E4SS	PCA4738S-42A
M38514E6SS	
M38513E4SP (One Time PROM version shipped in blank)	
M38514E6SP (One Time PROM version shipped in blank)	
M38513E4FP (One Time PROM version shipped in blank)	PCA4738F-42A
M38514E6FP (One Time PROM version shipped in blank)	

APPENDIX

3.3 Notes on use

(2) Programming/reading

In PROM mode, operation is the same as that of the M5M27C101AK, 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 programming /reading. Take care not to apply 21 V to V_{PP} pin (is also used as the CNV_{SS} pin), or the product may be permanently damaged.

- Programming voltage: 12.5 V
- Setting of PROM programmer switch: refer to Table 3.3.3.

Table 3.3.2 PROM programmer address setting

Product name format	PROM programmer start address	PROM programmer end address
M38513E4SS	Address 0C080 ₁₆ (Note 1)	Address 0FFFD ₁₆ (Note 1)
M38513E4SP		
M38513E4FP		
M38514E6SS	Address 0A080 ₁₆ (Note 2)	Address 0FFFD ₁₆ (Note 2)
M38514E6SP		
M38514E6FP		

Notes 1: Addresses C080₁₆ to FFFD₁₆ in the built-in PROM corresponds to addresses 0C080₁₆ to 0FFFD₁₆ in the PROM programmer.

2: Addresses A080₁₆ to FFFD₁₆ in the built-in PROM corresponds to addresses 0A080₁₆ to 0FFFD₁₆ in the PROM 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 are required to erase EPROM contents.

3.3.14 Termination of unused pins

(1) Terminate unused pins

① Output ports : Open

② Input ports :

Connect each pin to VCC or VSS through each resistor of 1 k Ω to 10 k Ω .

As for pins whose potential affects to operation modes such as pins CNVSS, INT or others, select the VCC pin or the VSS pin according to their operation mode.

③ I/O ports :

- Set the I/O ports for the input mode and connect them to VCC or VSS through each resistor of 1 k Ω to 10 k Ω .

Set the I/O ports for the output mode and open them at “L” or “H”.

- When opening them in the output mode, the input mode of the initial status remains until the mode of the ports is switched over to the output mode by the program after reset. Thus, the potential at these pins is undefined and the power source current may increase in the input mode. With regard to an effects on the system, thoroughly perform system evaluation on the user side.
- Since the direction register setup may be changed because of a program runaway or noise, set direction registers by program periodically to increase the reliability of program.

(2) Termination remarks

① Input ports and I/O ports :

Do not open in the input mode.

● Reason

- The power source current may increase depending on the first-stage circuit.
- An effect due to noise may be easily produced as compared with proper termination ② and ③ shown on the above.

② I/O ports :

When setting for the input mode, do not connect to VCC or VSS directly.

● Reason

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

③ I/O ports :

When setting for the input mode, do not connect multiple ports in a lump to VCC or VSS through a resistor.

● Reason

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

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

APPENDIX

3.4 Countermeasures against noise

3.4 Countermeasures against noise

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

3.4.1 Shortest wiring length

The wiring on a printed circuit board can function as an antenna which feeds noise into the microcomputer. The shorter the total wiring length (by mm unit), the less the possibility of noise insertion into a microcomputer.

(1) Package

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

● Reason

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

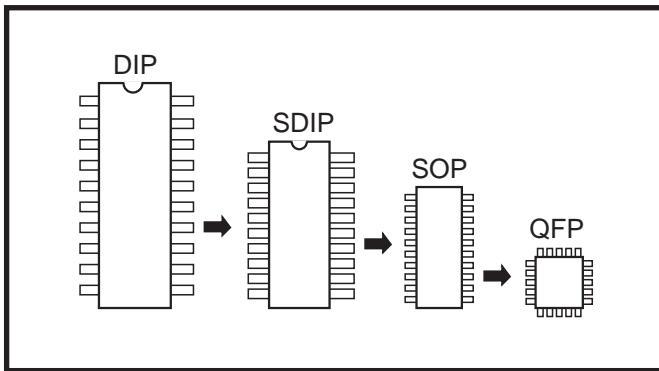


Fig. 3.4.1 Selection of packages

(2) Wiring for $\overline{\text{RESET}}$ pin

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

● Reason

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

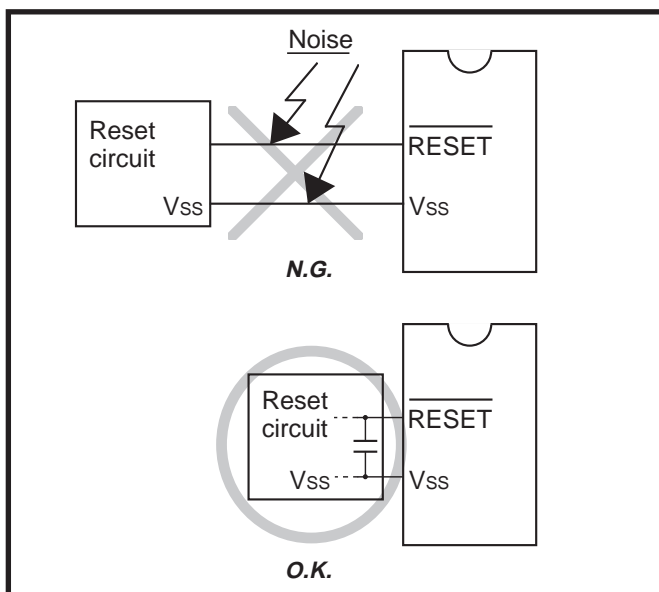


Fig. 3.4.2 Wiring for the $\overline{\text{RESET}}$ pin

(3) Wiring for clock input/output pins

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

● Reason

If noise enters clock I/O pins, clock waveforms may be deformed. This may cause a program failure or program runaway. Also, if a potential difference is caused by the noise between the Vss level of a microcomputer and the Vss level of an oscillator, the correct clock will not be input in the microcomputer.

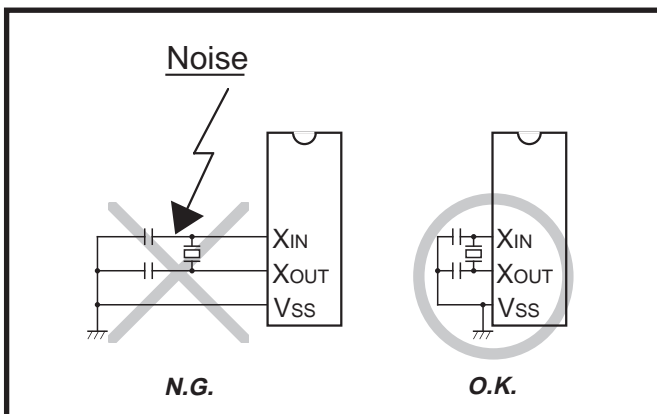


Fig. 3.4.3 Wiring for clock I/O pins

(4) Wiring to CNVss pin

Connect the CNVss pin to the Vss pin with the shortest possible wiring.

● Reason

The processor mode of a microcomputer is influenced by a potential at the CNVss pin. If a potential difference is caused by the noise between pins CNVss and Vss, the processor mode may become unstable. This may cause a microcomputer malfunction or a program runaway.

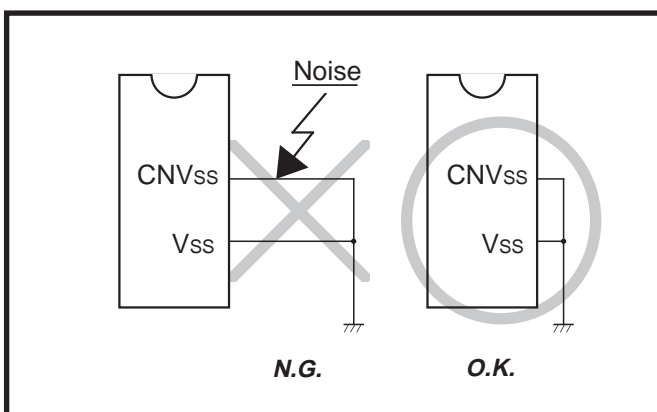


Fig. 3.4.4 Wiring for CNVss pin

APPENDIX

3.4 Countermeasures against noise

(5) Wiring to VPP pin of One Time PROM version and EPROM version

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

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

● Reason

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

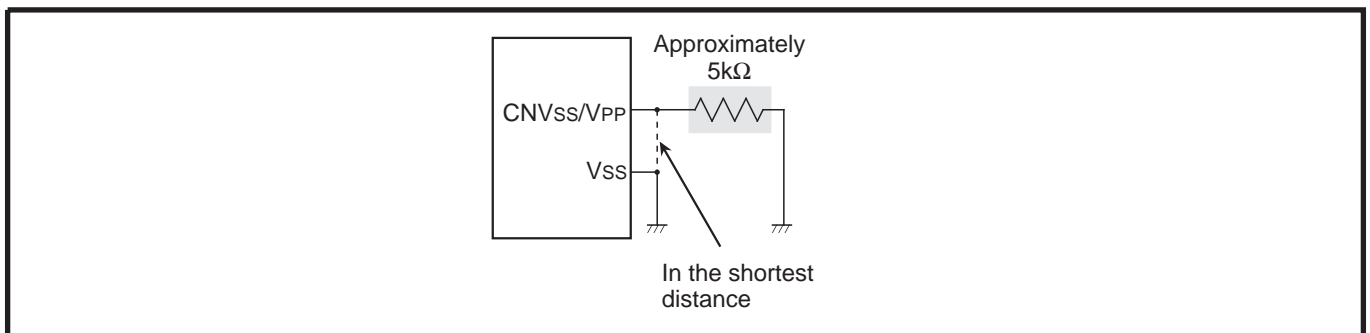


Fig. 3.4.5 Wiring for the VPP pin of the One Time PROM and the EPROM version

3.4.2 Connection of bypass capacitor across Vss line and 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.
- Connect the power source wiring via a bypass capacitor to the VSS pin and the VCC pin.

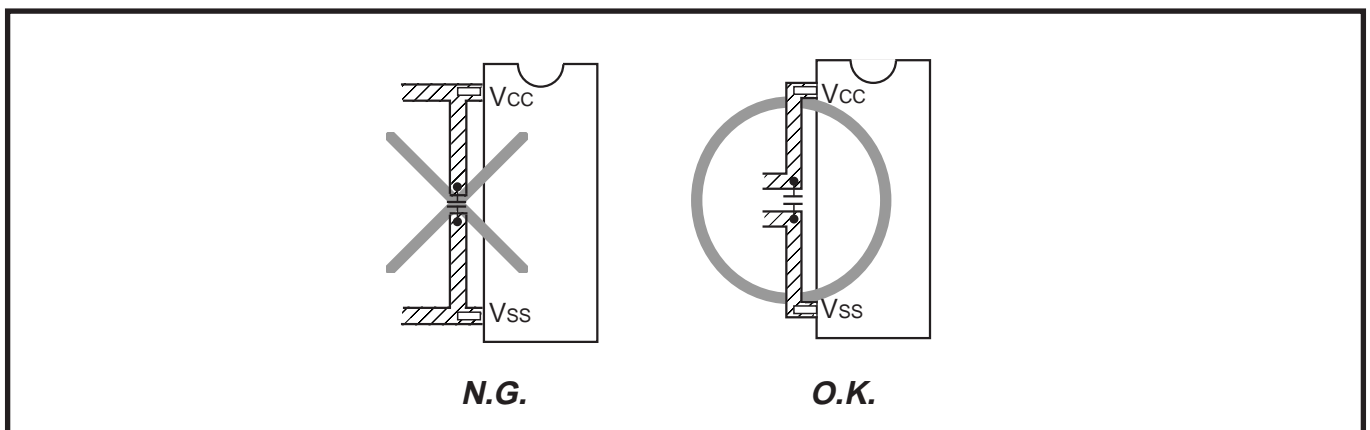


Fig. 3.4.6 Bypass capacitor across the VSS line and the VCC line

3.4.3 Wiring to analog input pins

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

● Reason

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

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

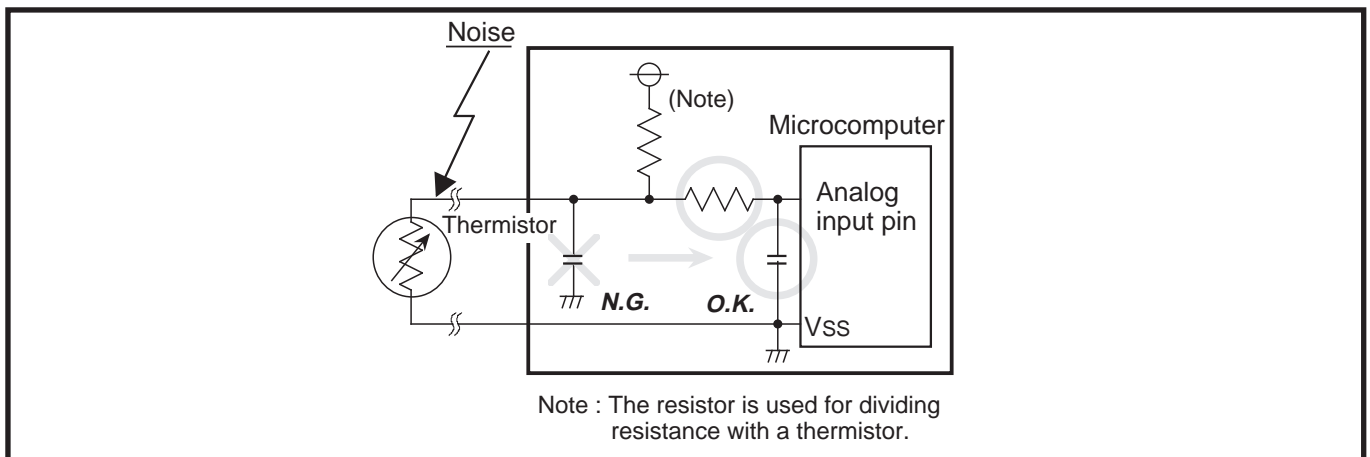


Fig. 3.4.7 Analog signal line and a resistor and a capacitor

3.4.4 Oscillator concerns

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

(1) Keeping oscillator away from large current signal lines

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

● Reason

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

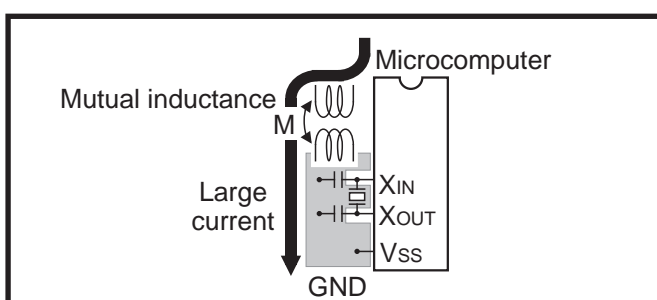


Fig. 3.4.8 Wiring for a large current signal line

APPENDIX

3.4 Countermeasures against noise

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

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

● Reason

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

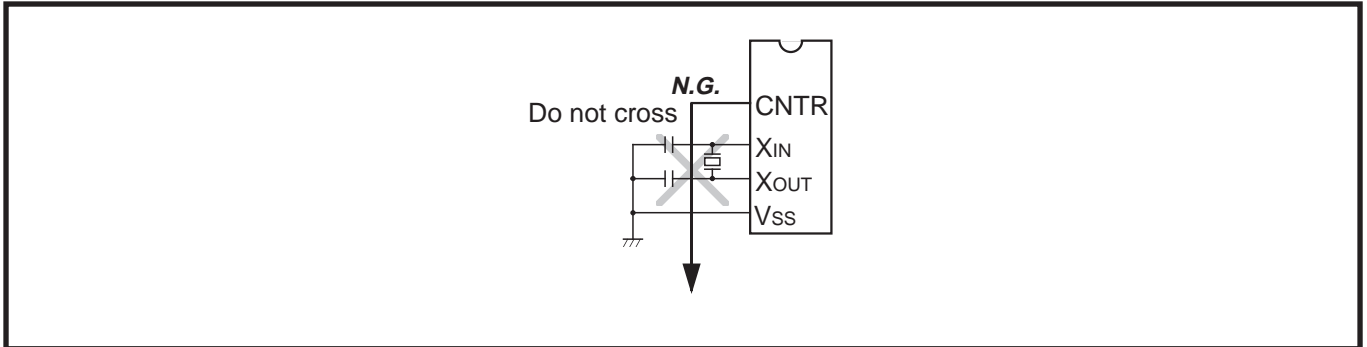


Fig. 3.4.9 Wiring of RESET pin

(3) Oscillator protection using Vss pattern

As for a two-sided printed circuit board, print a Vss pattern on the underside (soldering side) of the position (on the component side) where an oscillator is mounted.

Connect the Vss pattern to the microcomputer Vss pin with the shortest possible wiring. Besides, separate this Vss pattern from other Vss patterns.

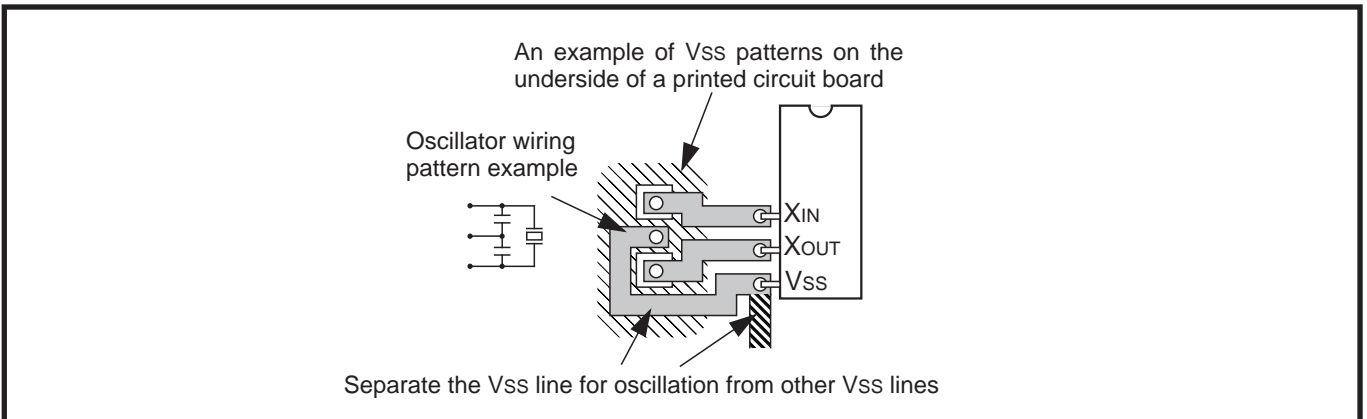


Fig. 3.4.10 Vss pattern on the underside of an oscillator

3.4.5 Setup for I/O ports

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

<Hardware>

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

<Software>

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

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

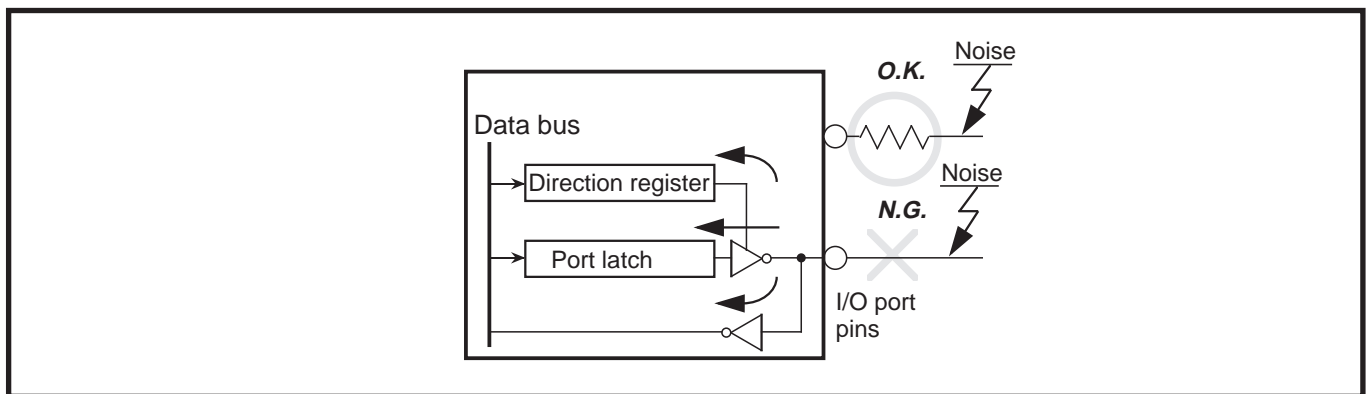


Fig. 3.4.11 Setup for I/O ports

APPENDIX

3.4 Countermeasures against noise

3.4.6 Providing of watchdog timer function by software

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

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

<The main routine>

- Assigns a single byte of RAM to a software watchdog timer (SWDT) and writes the initial value N in the SWDT once at each execution of the main routine. The initial value N should satisfy the following condition:
 $N+1 \geq (\text{Counts of interrupt processing executed in each main routine})$
As the main routine execution cycle may change because of an interrupt processing or others, the initial value N should have a margin.
- Watches the operation of the interrupt processing routine by comparing the SWDT contents with counts of interrupt processing after the initial value N has been set.
- Detects that the interrupt processing routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
If the SWDT contents do not change after interrupt processing.

<The interrupt processing routine>

- Decrements the SWDT contents by 1 at each interrupt processing.
- Determines that the main routine operates normally when the SWDT contents are reset to the initial value N at almost fixed cycles (at the fixed interrupt processing count).
- Detects that the main routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
If the SWDT contents are not initialized to the initial value N but continued to decrement and if they reach 0 or less.

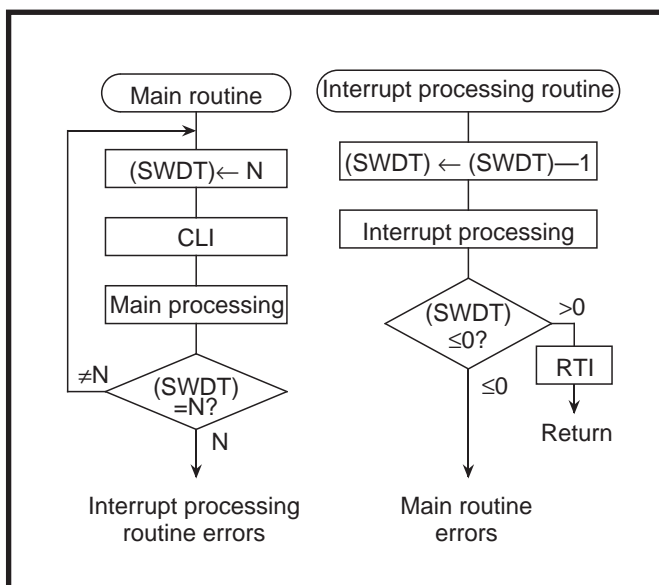


Fig. 3.4.12 Watchdog timer by software

3.5 List of registers

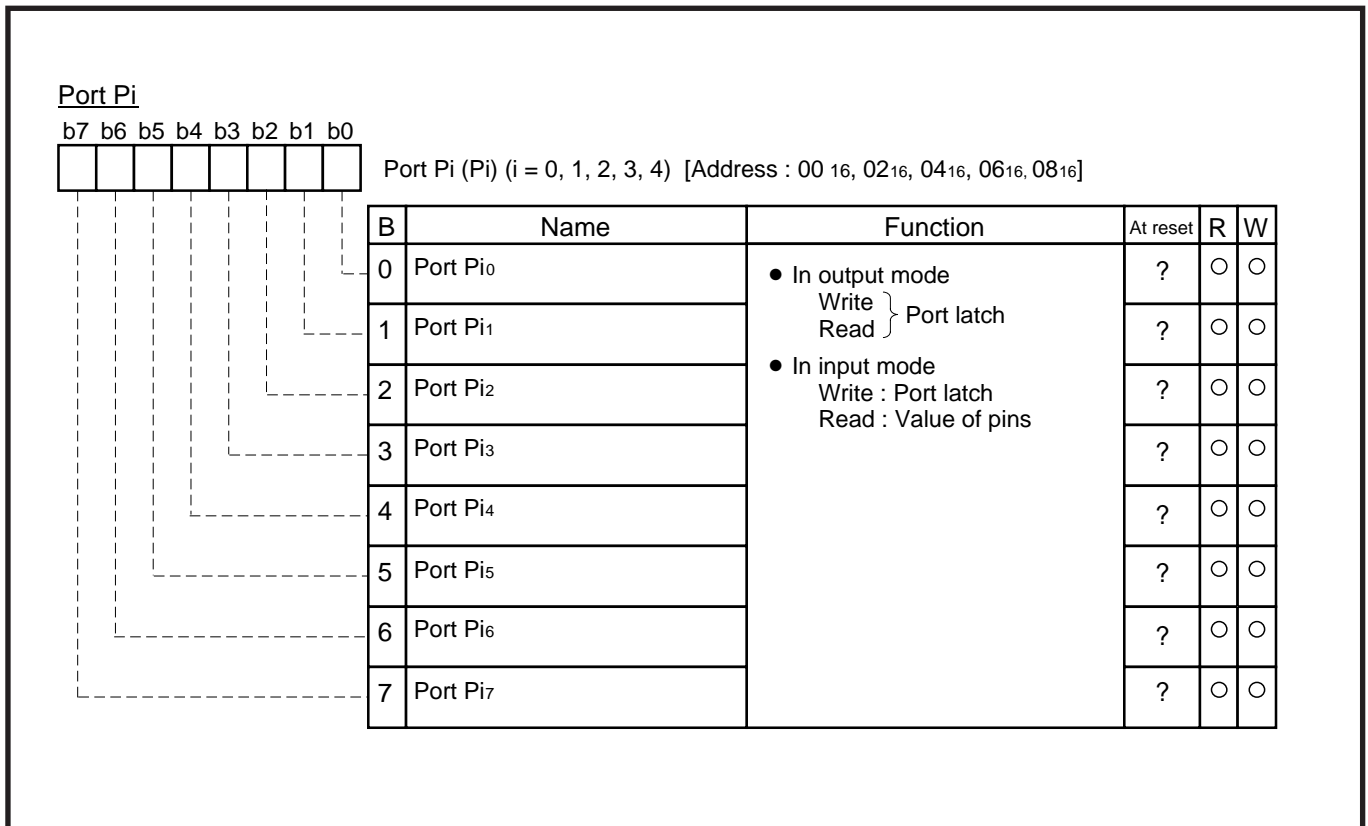


Fig. 3.5.1 Structure of Port Pi (i=0, 1, 2, 3, 4)

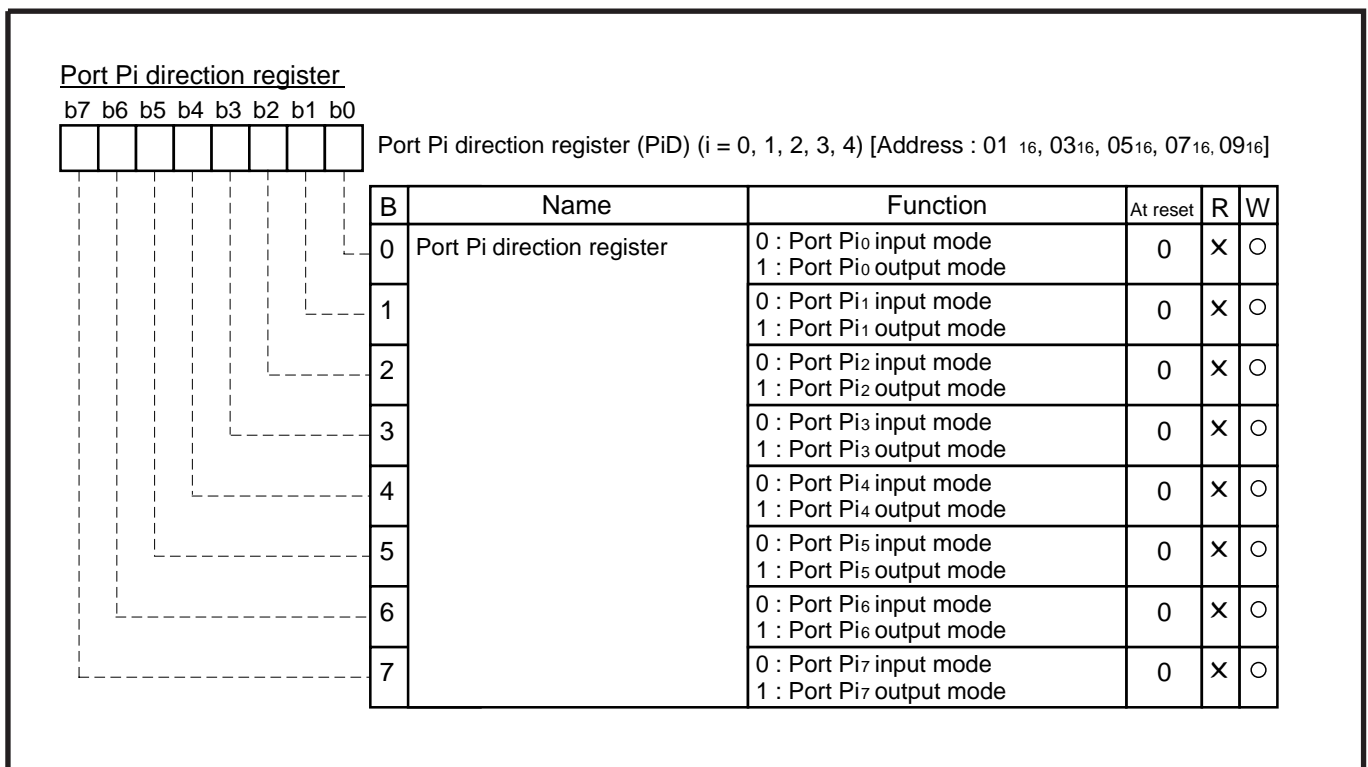


Fig. 3.5.2 Structure of Port Pi direction register(i=0, 1, 2, 3, 4)

APPENDIX

3.5 List of registers

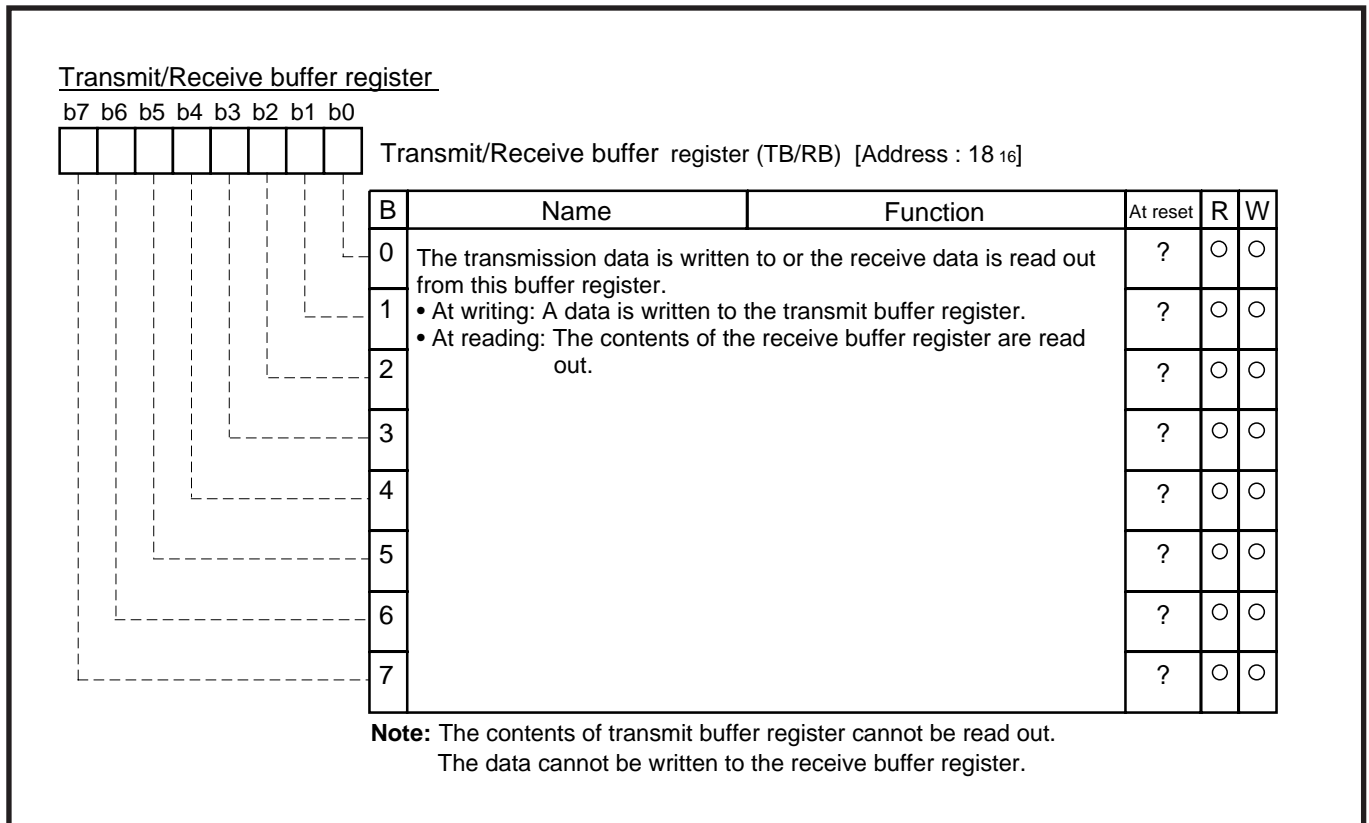


Fig. 3.5.3 Structure of Transmit/Receive buffer register

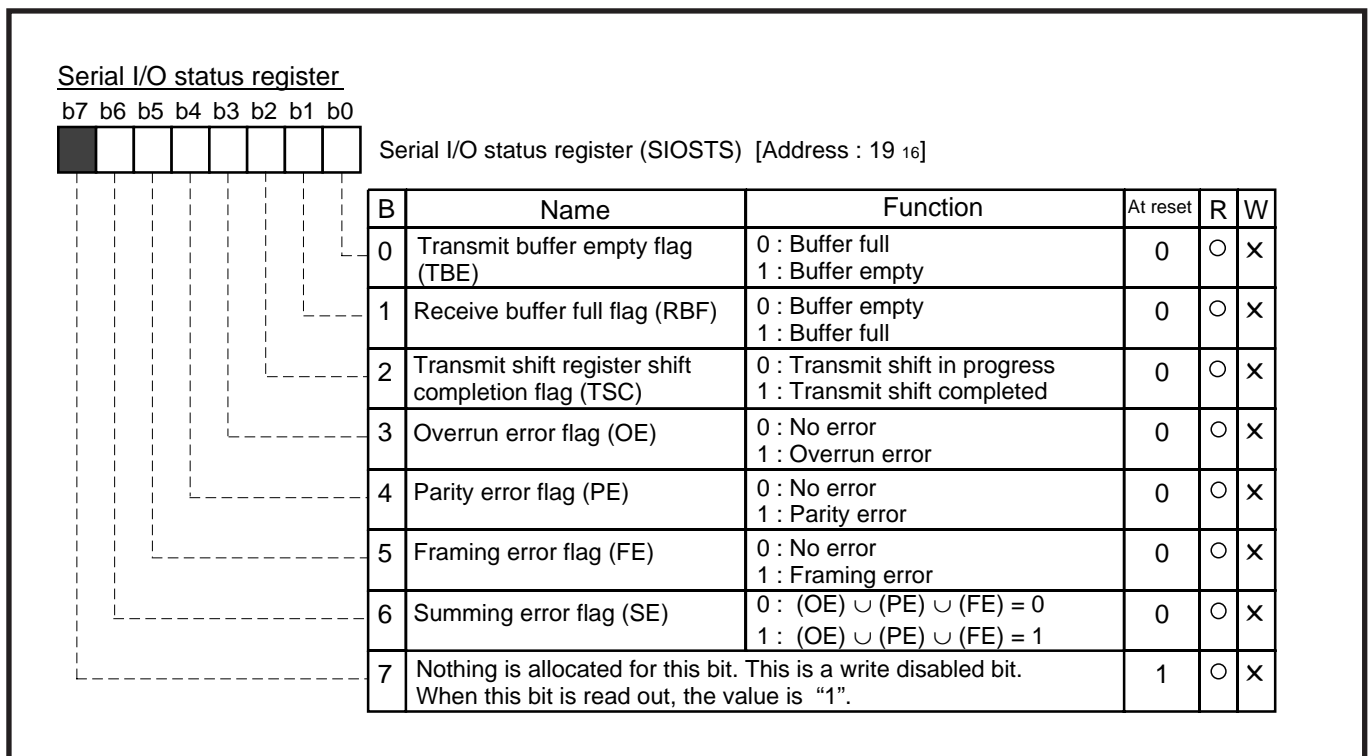


Fig. 3.5.4 Structure of Serial I/O status register

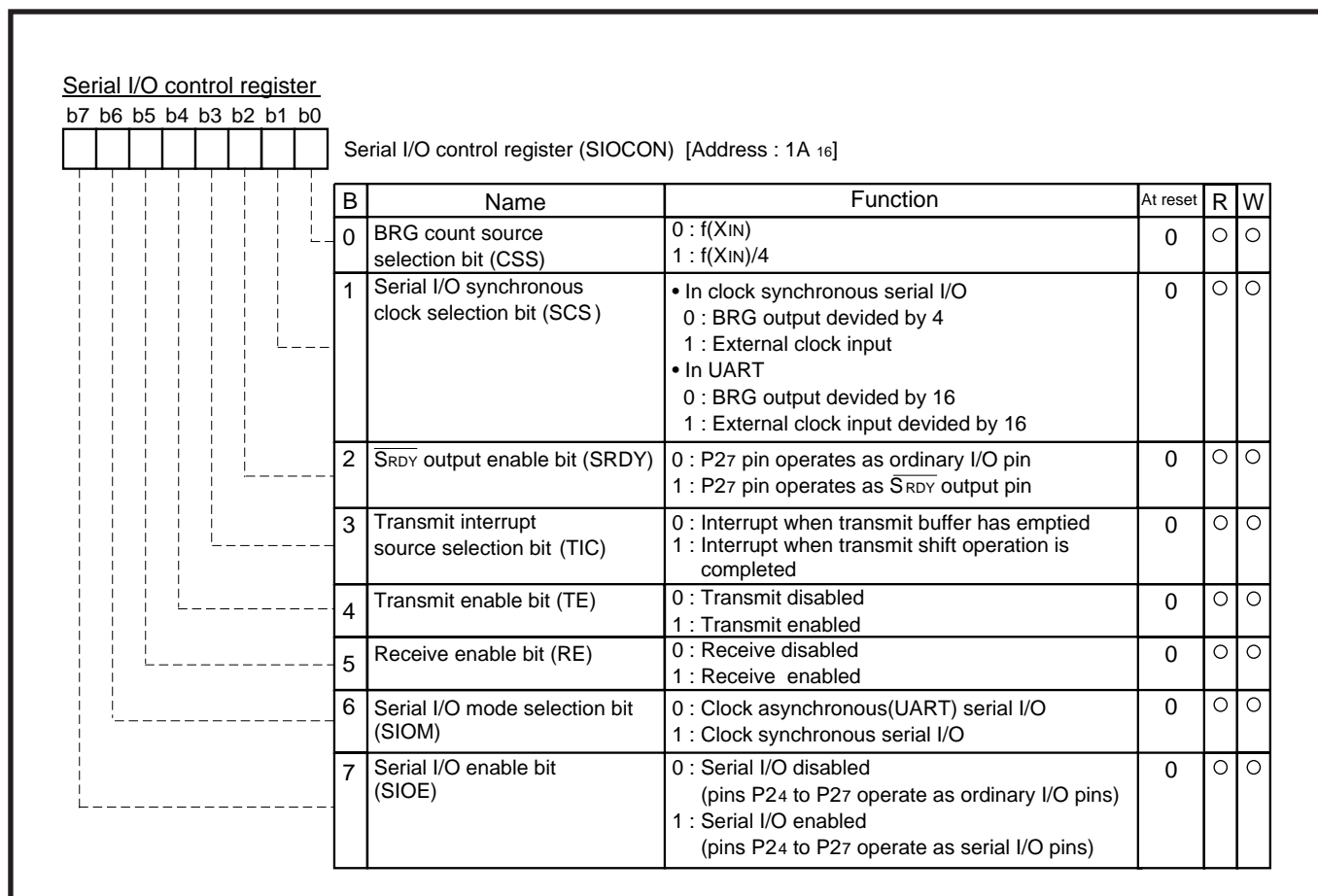


Fig. 3.5.5 Structure of Serial I/O control register

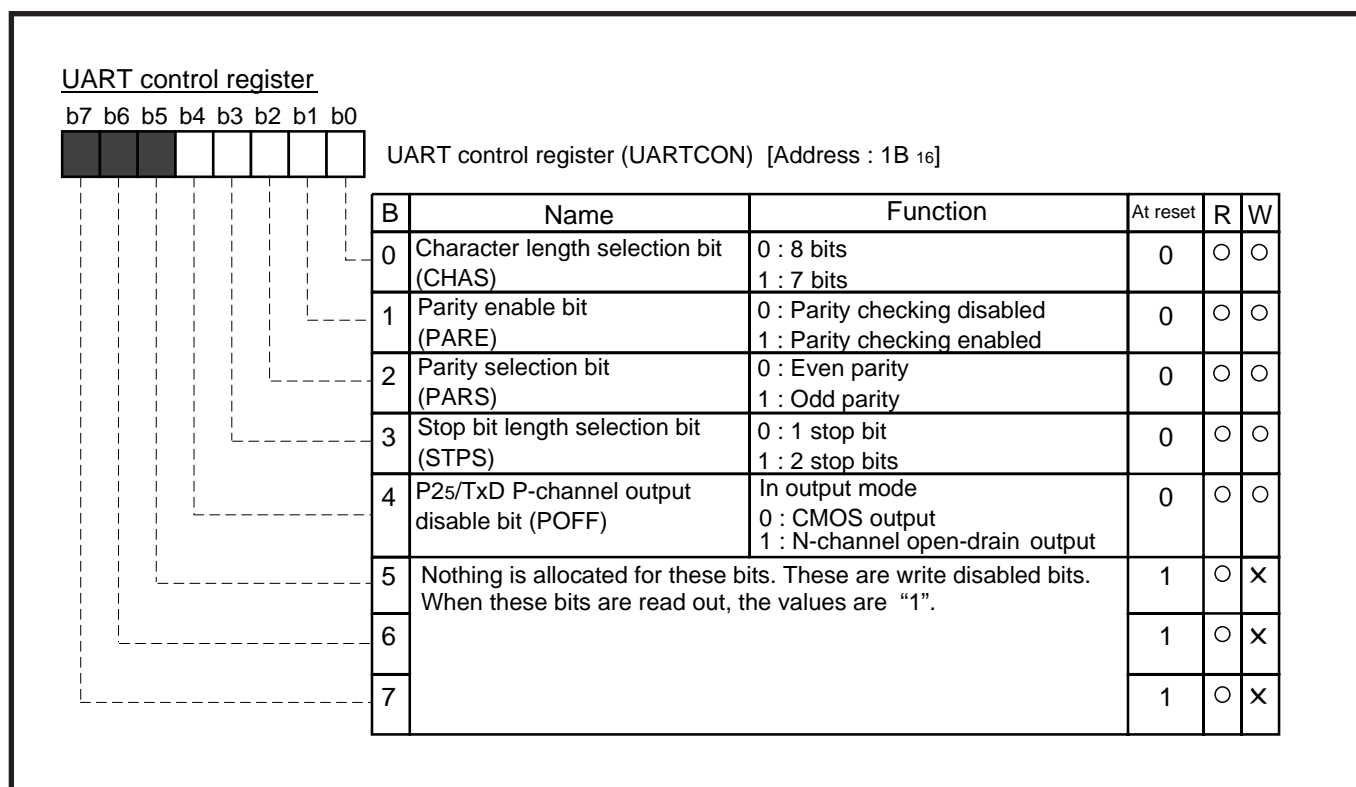


Fig. 3.5.6 Structure of UART control register

APPENDIX

3.5 List of registers

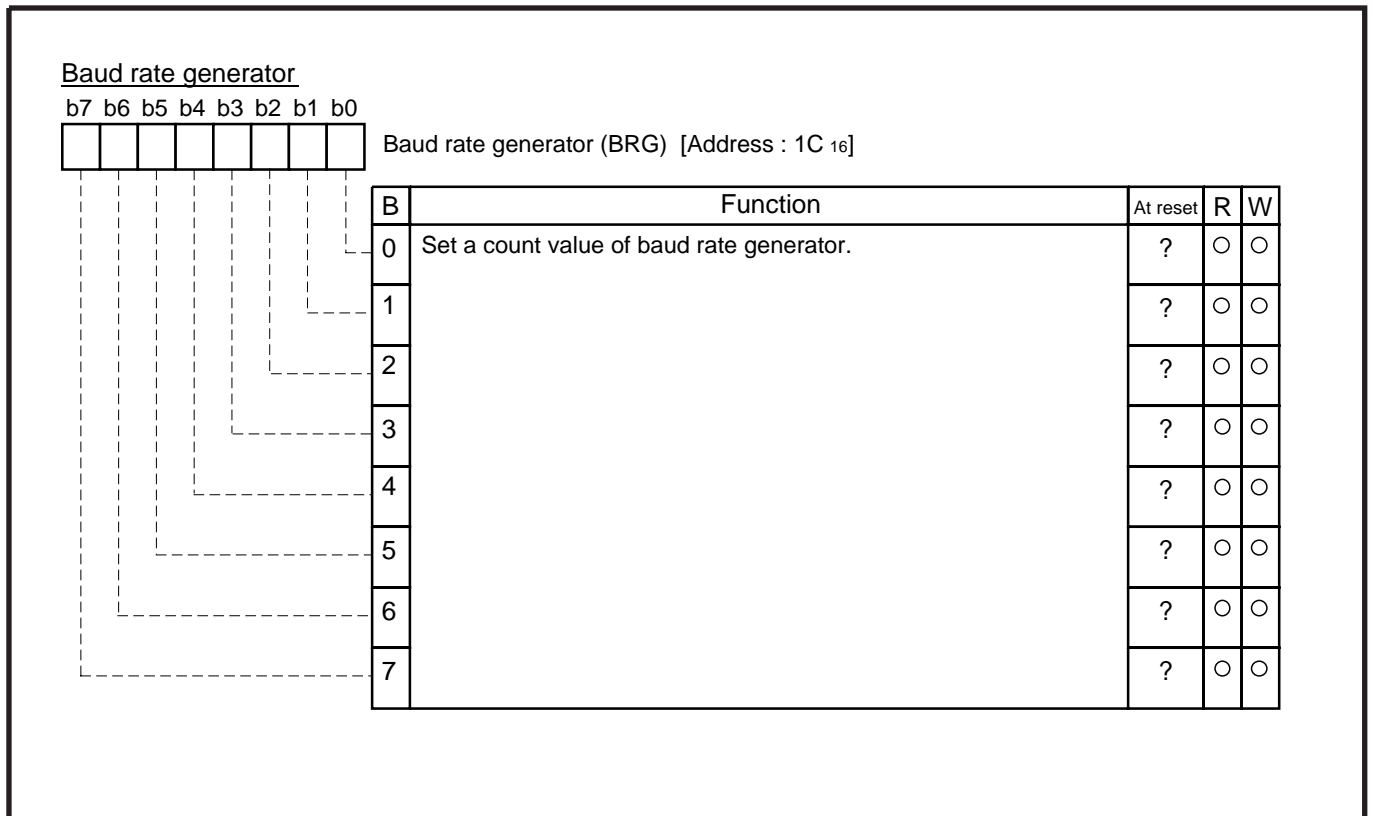


Fig. 3.5.7 Structure of Baud rate generator

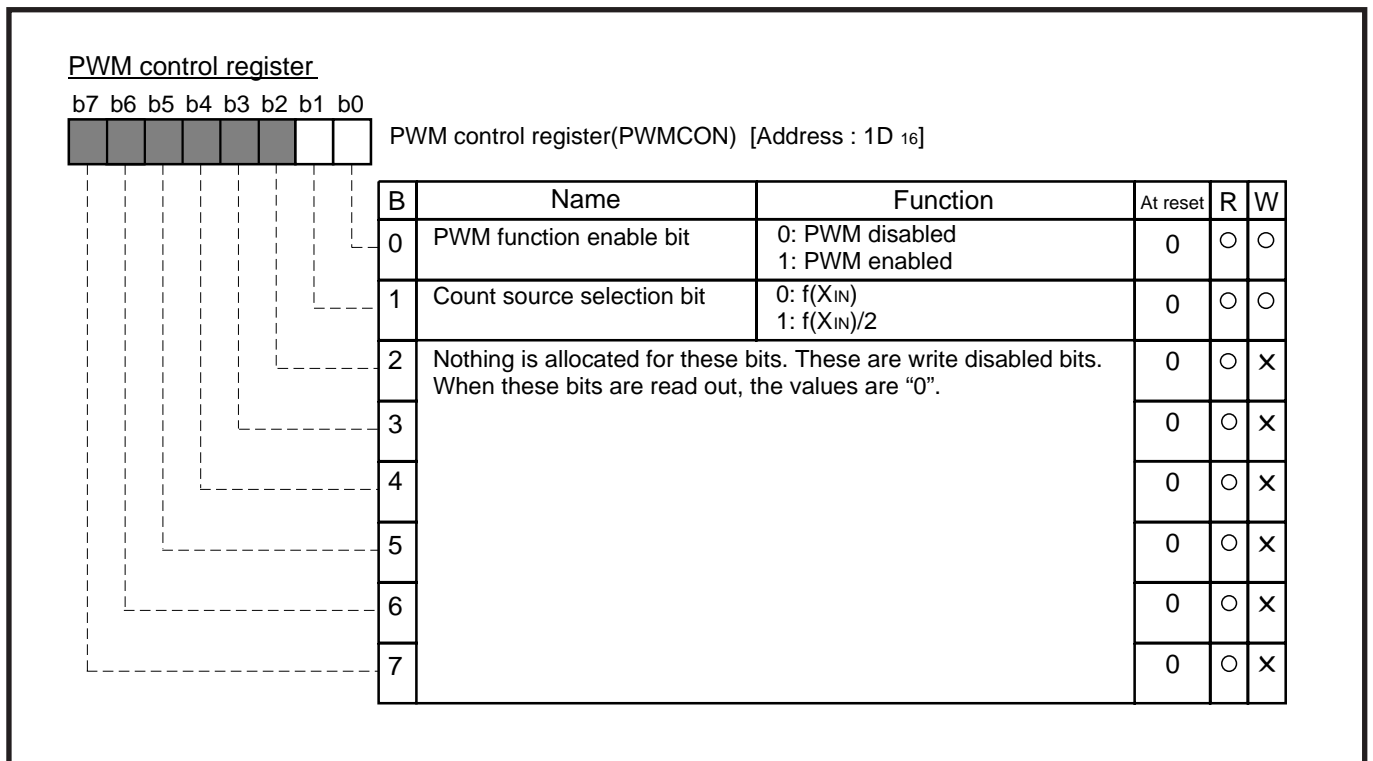


Fig. 3.5.8 Structure of PWM control register

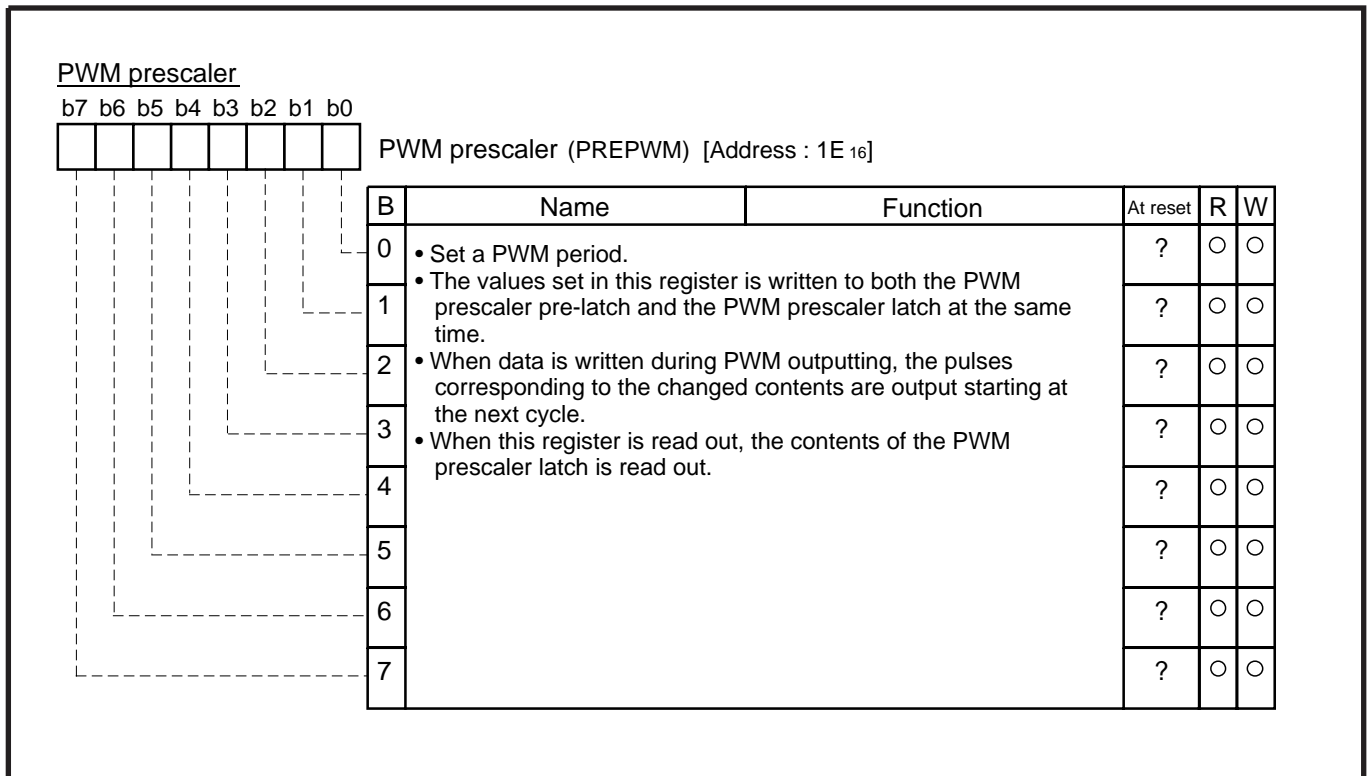


Fig. 3.5.9 Structure of PWM prescaler

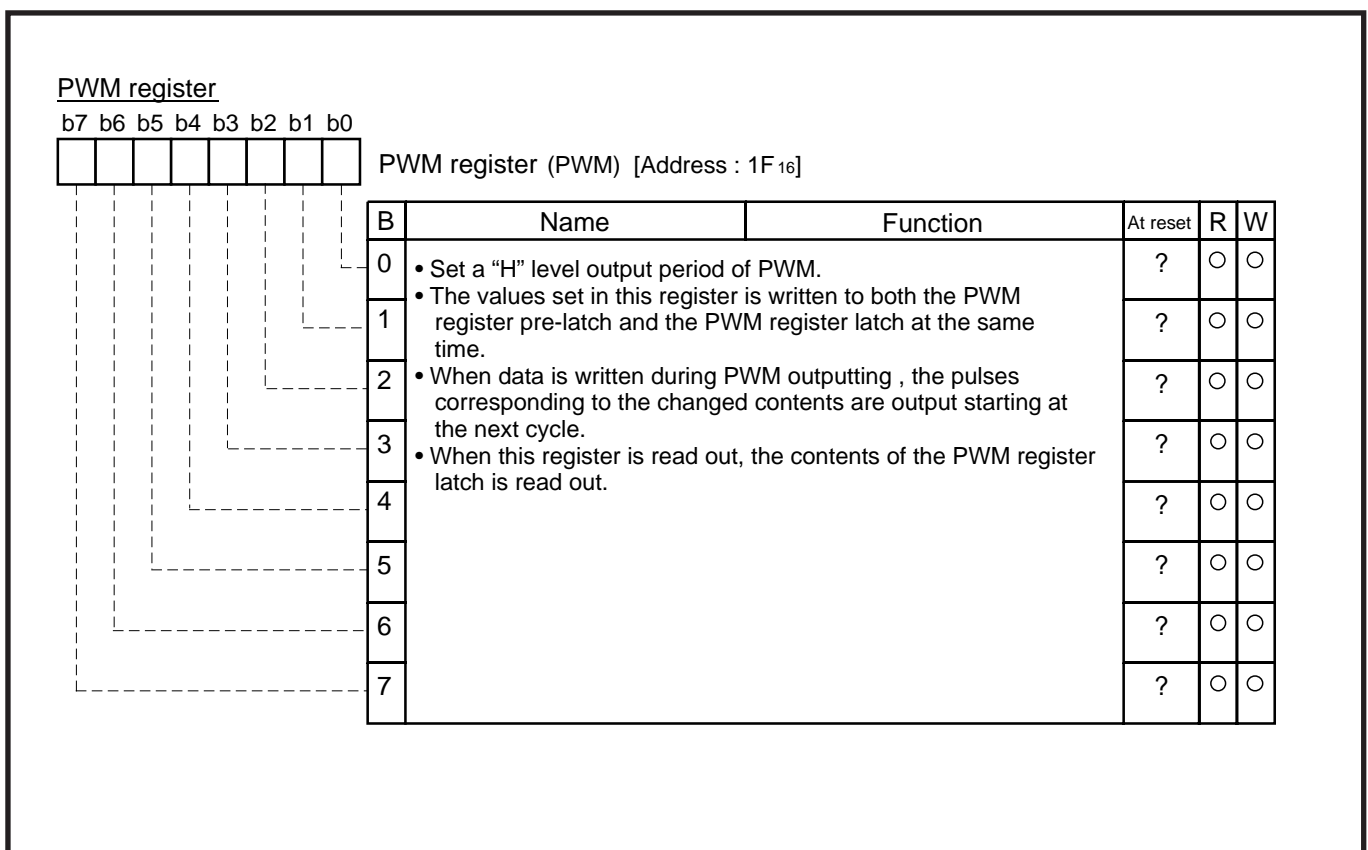


Fig. 3.5.10 Structure of PWM register

APPENDIX

3.5 List of registers

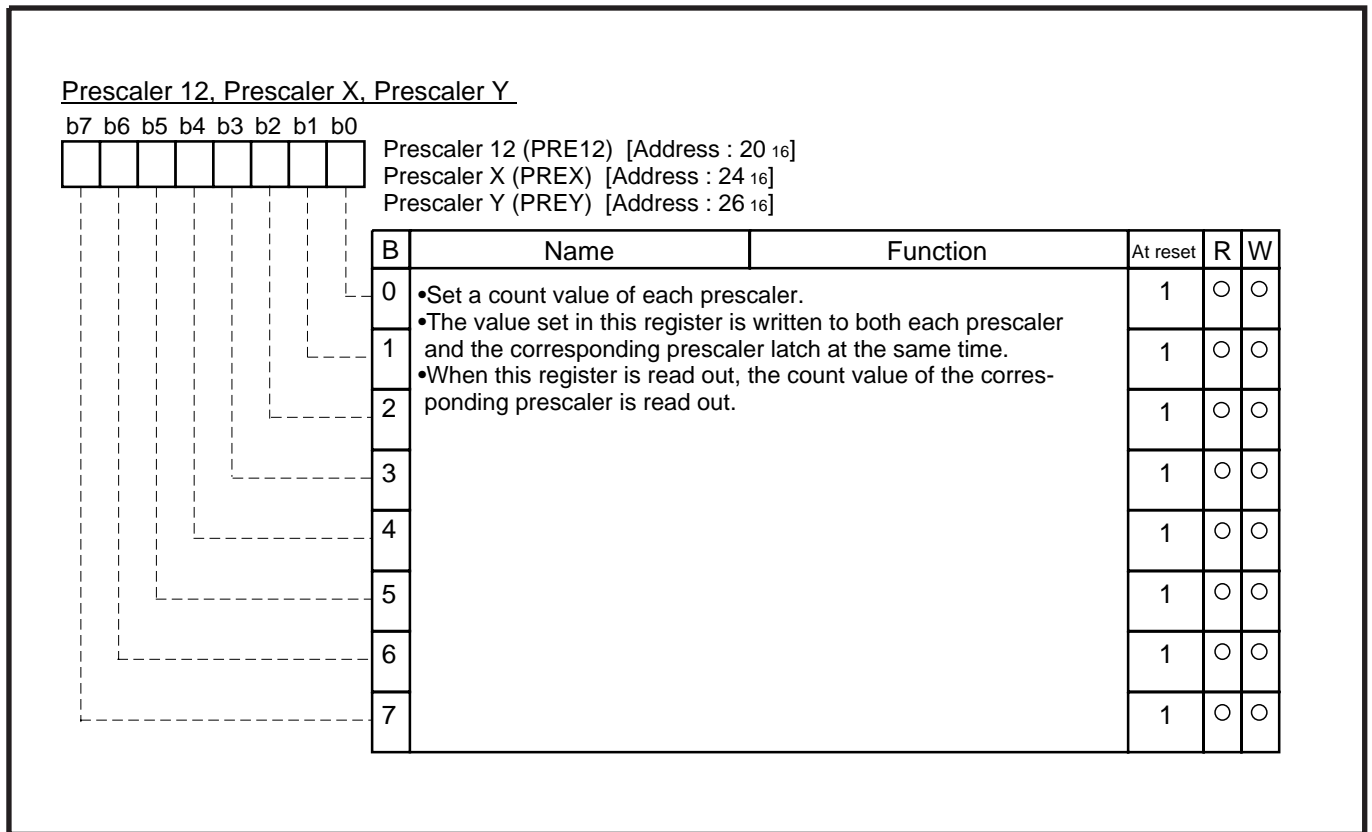


Fig. 3.5.11 Structure of Prescaler 12, Prescaler X, Prescaler Y

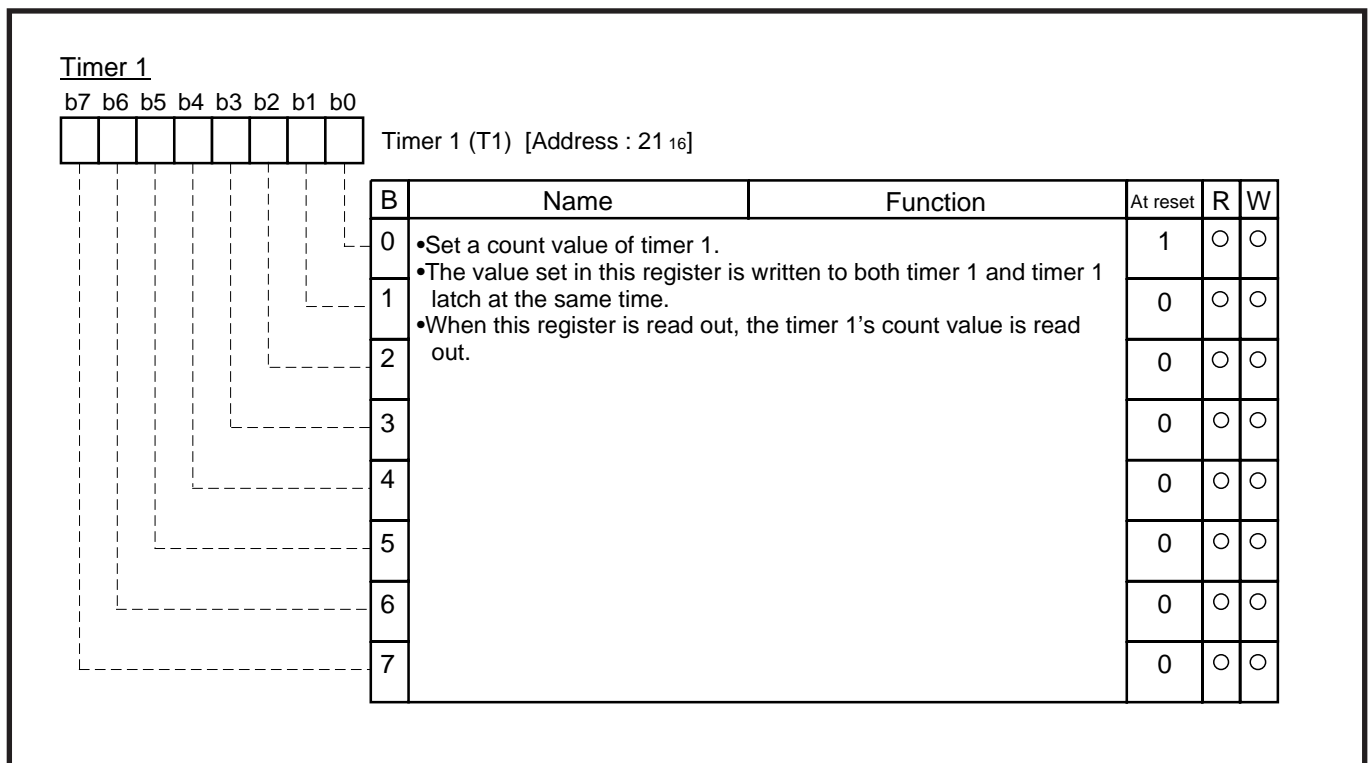


Fig. 3.5.12 Structure of Timer 1

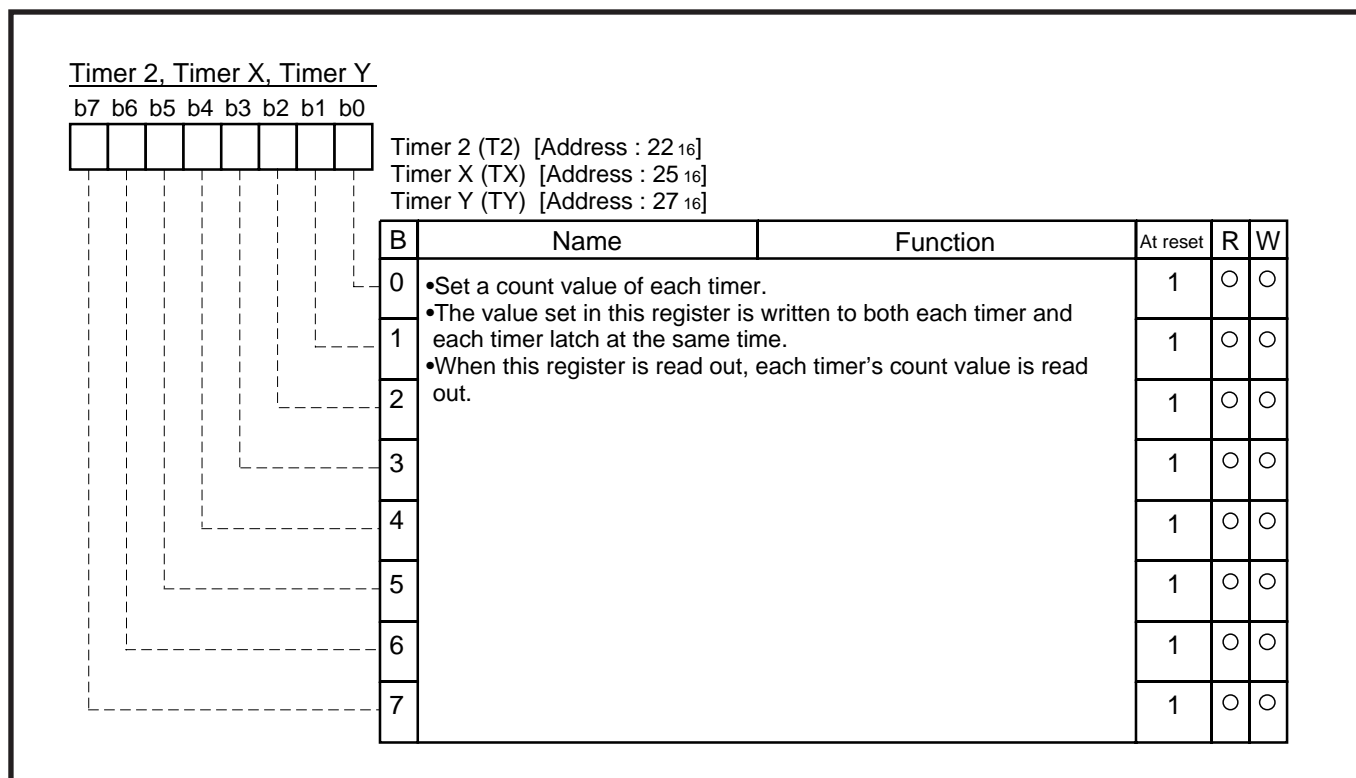


Fig. 3.5.13 Structure of Timer 2, Timer X, Timer Y

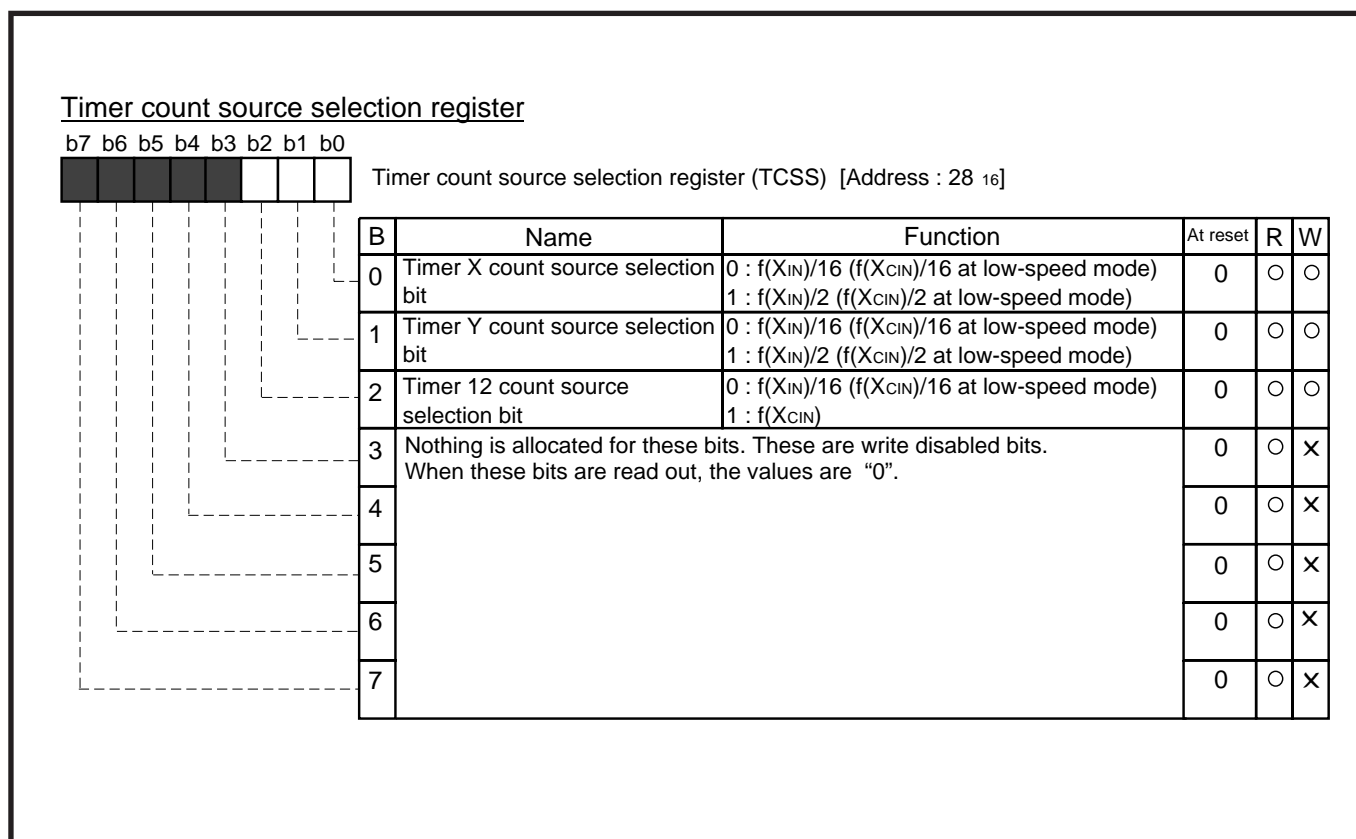


Fig. 3.5.14 Structure of timer count source selection register

APPENDIX

3.5 List of registers

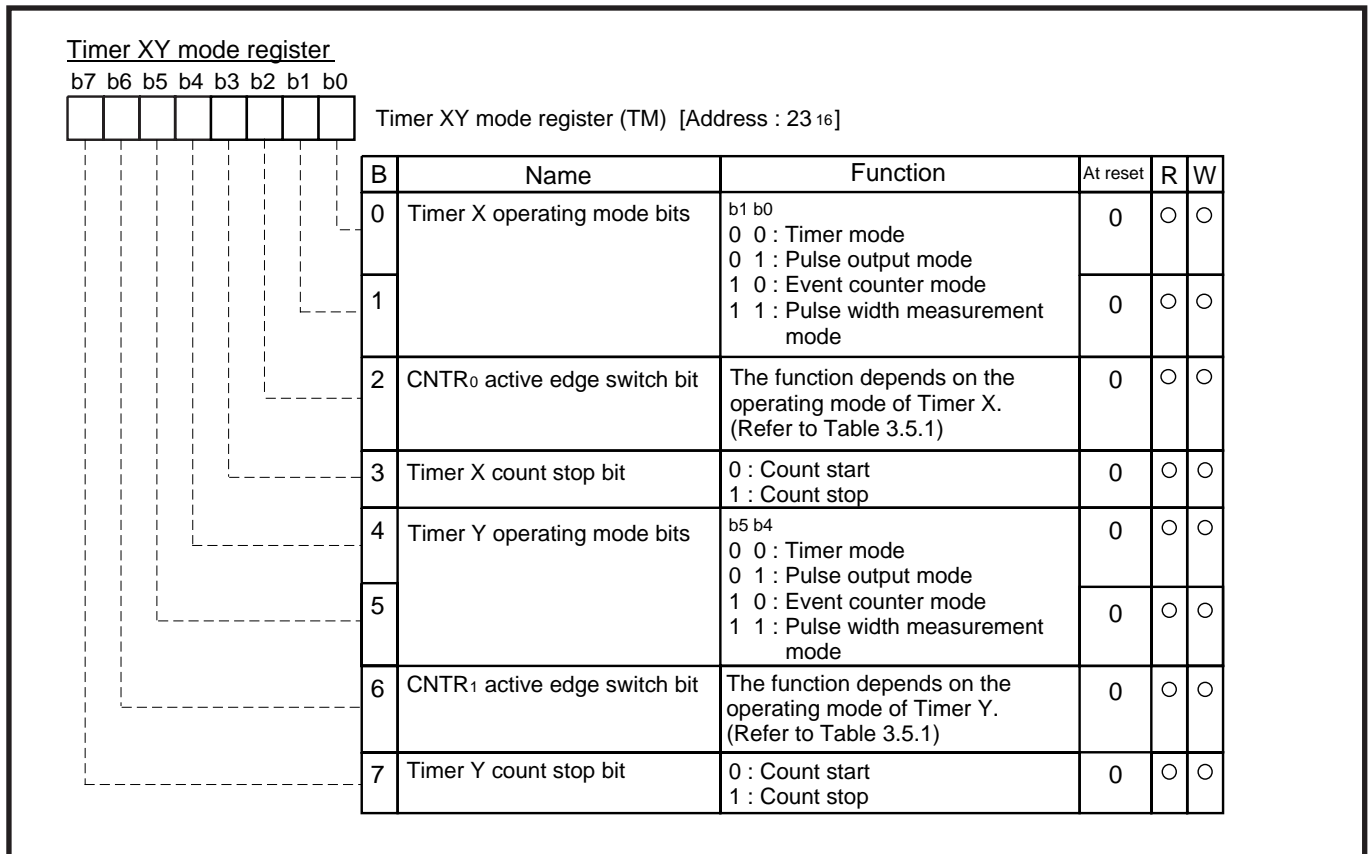


Fig. 3.5.15 Structure of Timer XY mode register

Table 3.5.1 CNTR₀ /CNTR₁ active edge switch bit function

Timer X /Timer Y operation modes	CNTR ₀ / CNTR ₁ active edge switch bit (bits 2, 6 of address 23 ₁₆) contents	
Timer mode	"0"	CNTR ₀ / CNTR ₁ interrupt request occurrence: Falling edge ; No influence to timer count
	"1"	CNTR ₀ / CNTR ₁ interrupt request occurrence: Rising edge ; No influence to timer count
Pulse output mode	"0"	Pulse output start: Beginning at "H" level CNTR ₀ / CNTR ₁ interrupt request occurrence: Falling edge
	"1"	Pulse output start: Beginning at "L" level CNTR ₀ / CNTR ₁ interrupt request occurrence: Rising edge
Event counter mode	"0"	Timer X / Timer Y: Rising edge count CNTR ₀ / CNTR ₁ interrupt request occurrence: Falling edge
	"1"	Timer X / Timer Y: Falling edge count CNTR ₀ / CNTR ₁ interrupt request occurrence: Rising edge
Pulse width measurement mode	"0"	Timer X / Timer Y: "H" level width measurement CNTR ₀ / CNTR ₁ interrupt request occurrence: Falling edge
	"1"	Timer X / Timer Y: "L" level width measurement CNTR ₀ / CNTR ₁ interrupt request occurrence: Rising edge

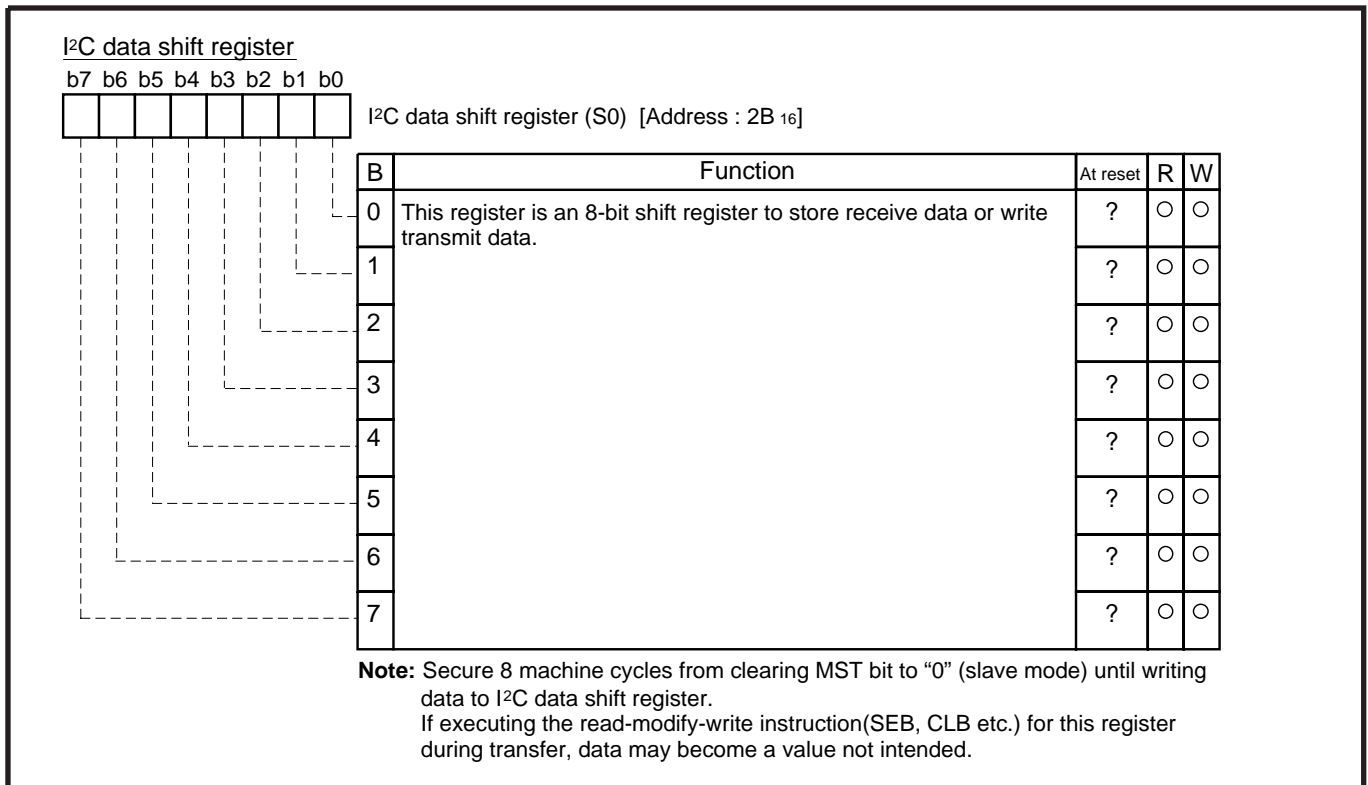


Fig. 3.5.16 Structure of I²C data shift register

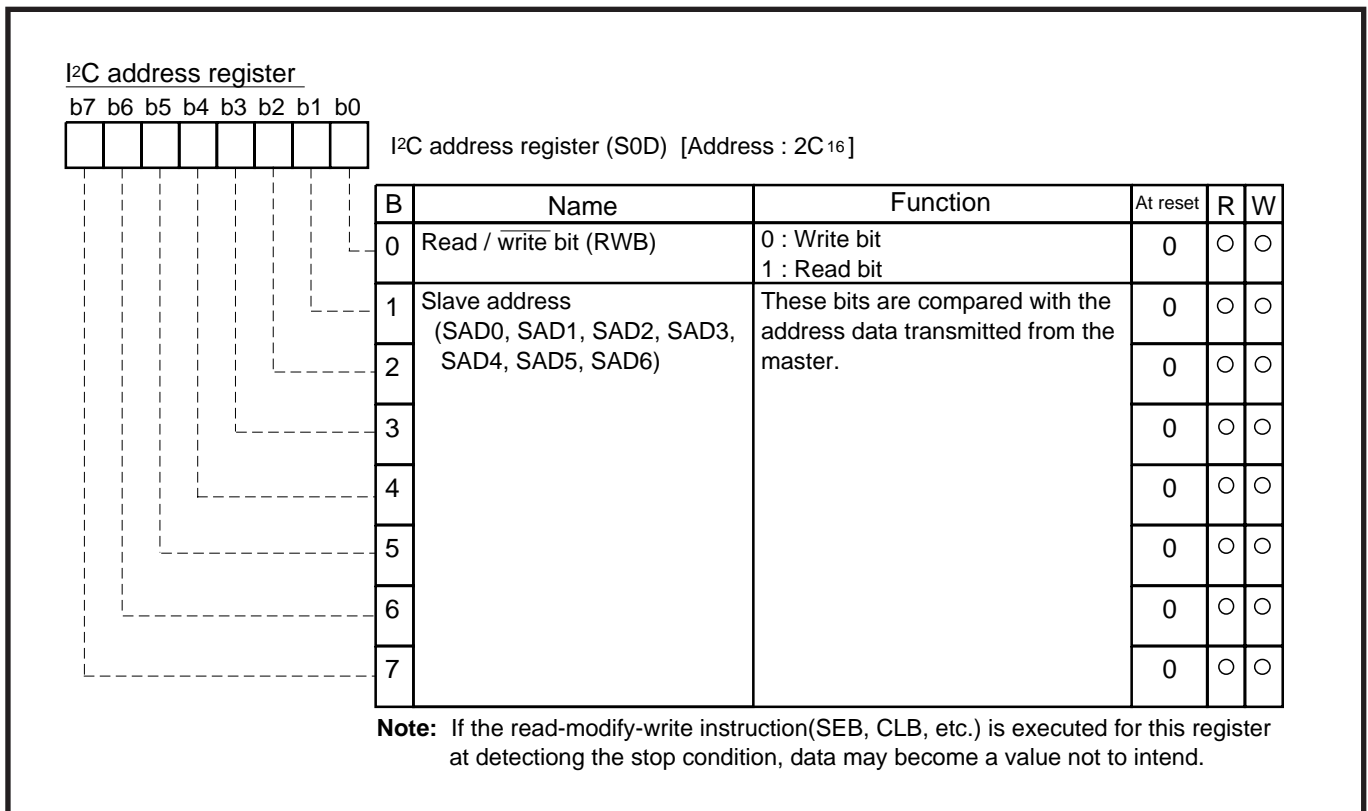


Fig. 3.5.17 Structure of I²C address register

APPENDIX

3.5 List of registers

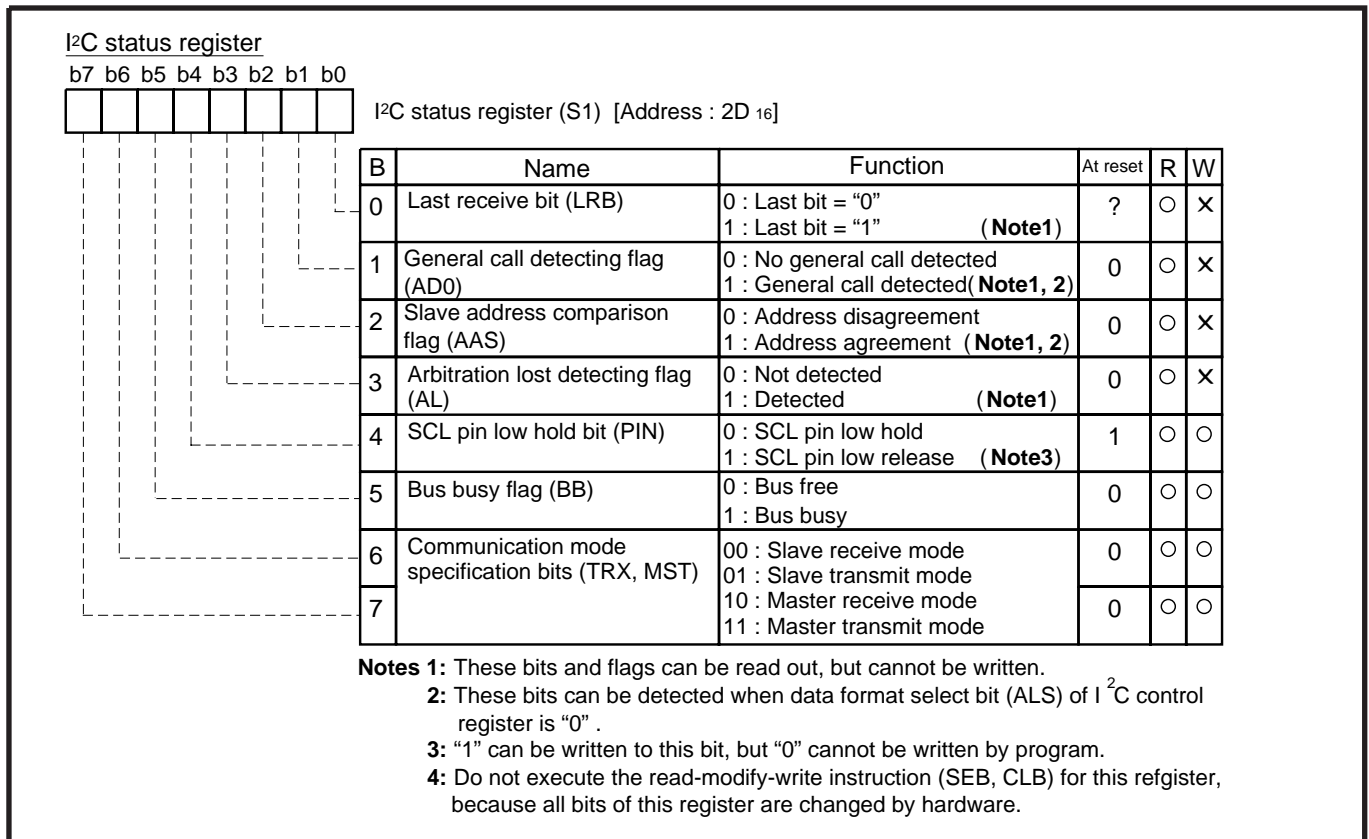


Fig. 3.5.18 Structure of I²C status register

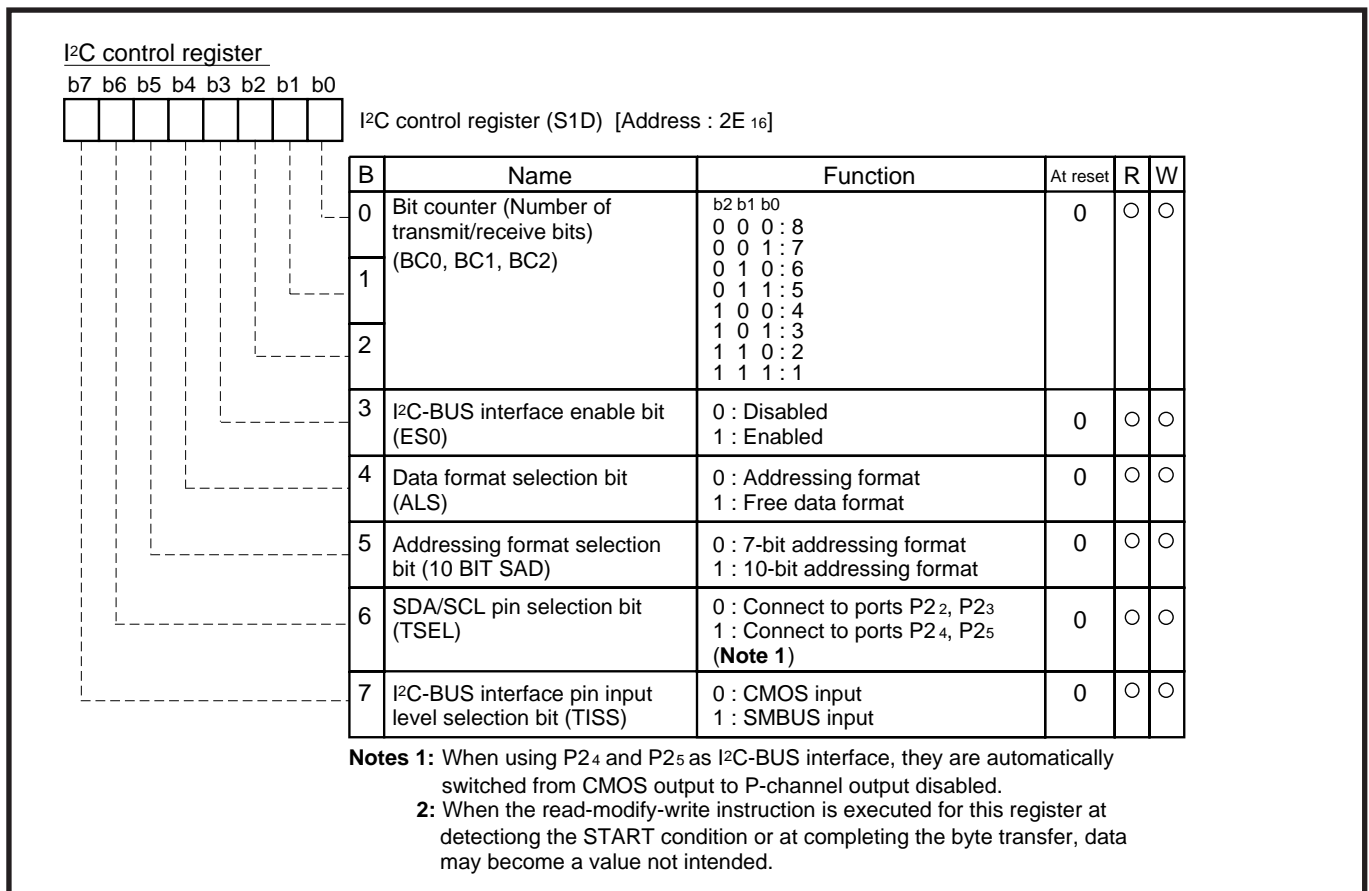


Fig. 3.5.19 Structure of I²C control register

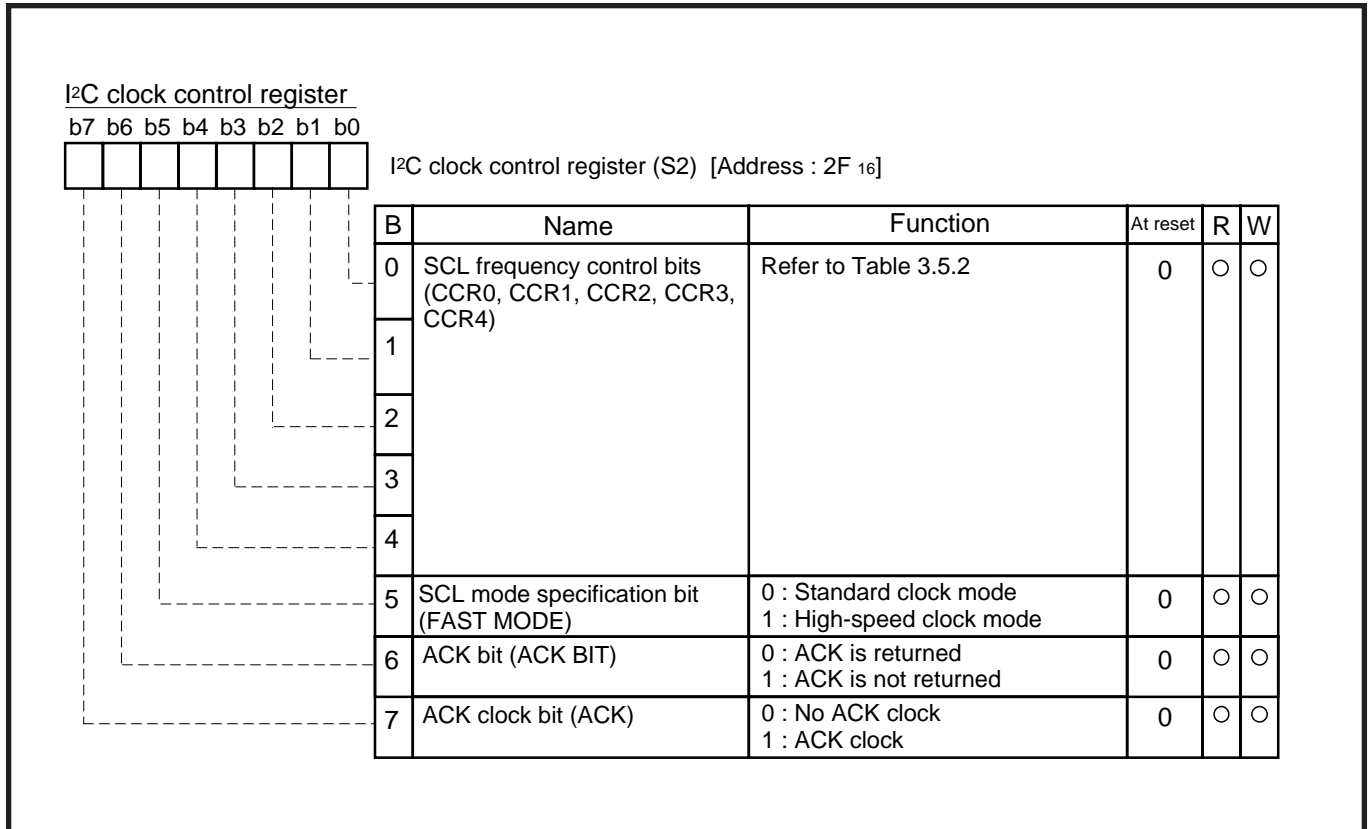


Fig. 3.5.20 Structure of I²C clock control register

Table 3.5.2 Set value of I²C clock control register and SCL frequency

Setting value of CCR4–CCR0					SCL frequency (at $\phi = 4$ MHz, unit : kHz) (Note 1)	
CCR4	CCR3	CCR2	CCR1	CCR0	Standard clk mode	High-speed clock mode
0	0	0	0	0	Setting disabled	Setting disabled
0	0	0	0	1	Setting disabled	Setting disabled
0	0	0	1	0	Setting disabled	Setting disabled
0	0	0	1	1	– (Note 2)	333
0	0	1	0	0	– (Note 2)	250
0	0	1	0	1	100	400 (Note 3)
0	0	1	1	0	83.3	166
⋮	⋮	⋮	⋮	⋮	500/CCR value (Note 3)	1000/CCR value (Note 3)
1	1	1	0	1	17.2	34.5
1	1	1	1	0	16.6	33.3
1	1	1	1	1	16.1	32.3

Notes 1: Duty of SCL clock output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at $\phi = 4$ MHz). "H" duration of the clock fluctuates from -4 to $+2$ machine cycles in the standard clock mode, and fluctuates from -2 to $+2$ machine cycles in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because "L" duration is extended instead of "H" duration reduction.

These are value when SCL clock synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

2: Each value of SCL frequency exceeds the limit at $\phi = 4$ MHz or more. When using these setting value, use ϕ of 4 MHz or less.

3: The data formula of SCL frequency is described below:

$\phi / (8 \times \text{CCR value})$ Standard clock mode

$\phi / (4 \times \text{CCR value})$ High-speed clock mode (CCR value $\neq 5$)

$\phi / (2 \times \text{CCR value})$ High-speed clock mode (CCR value = 5)

Do not set 0 to 2 as CCR value regardless of ϕ frequency.

Set 100 kHz (max.) in the standard clock mode and 400 kHz (max.) in the high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

APPENDIX

3.5 List of registers

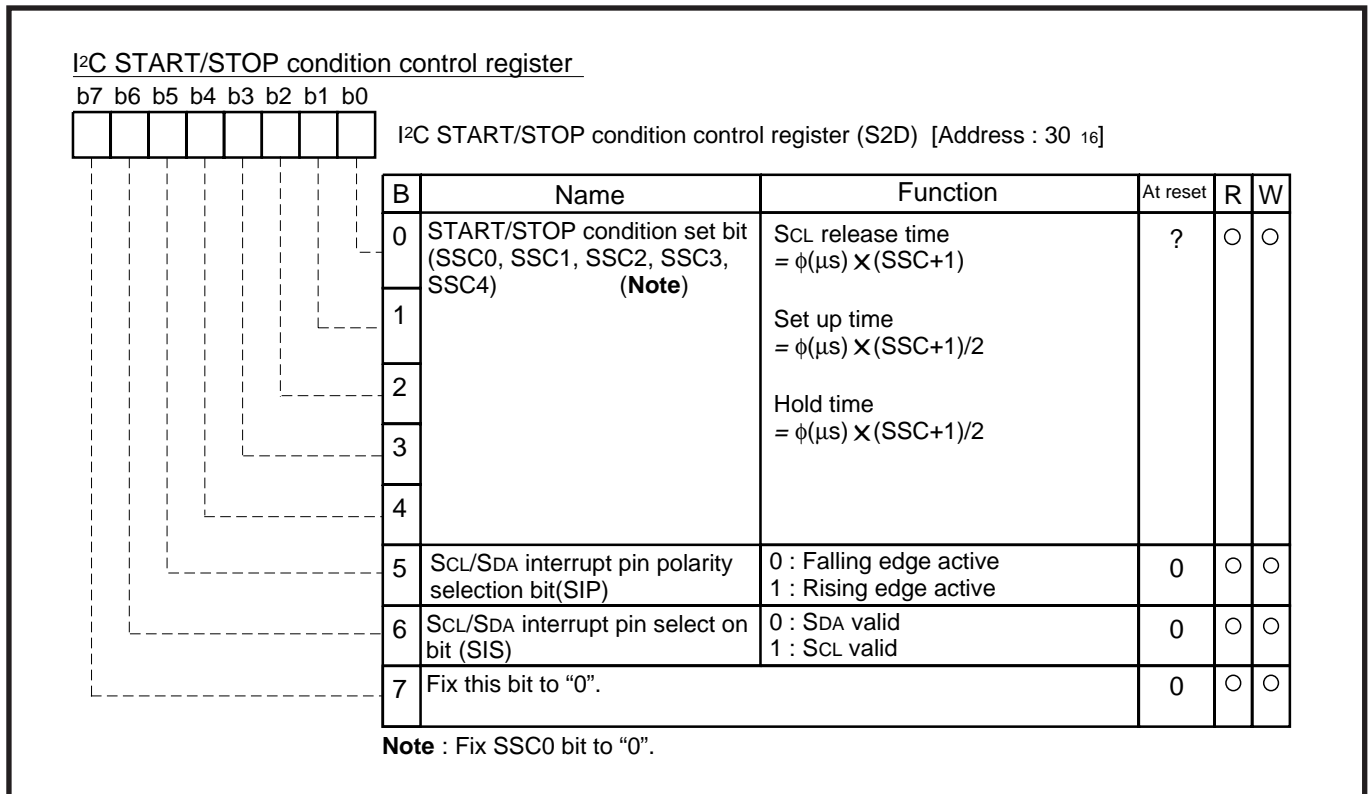


Fig. 3.5.21 Structure of I²C START/STOP condition control register

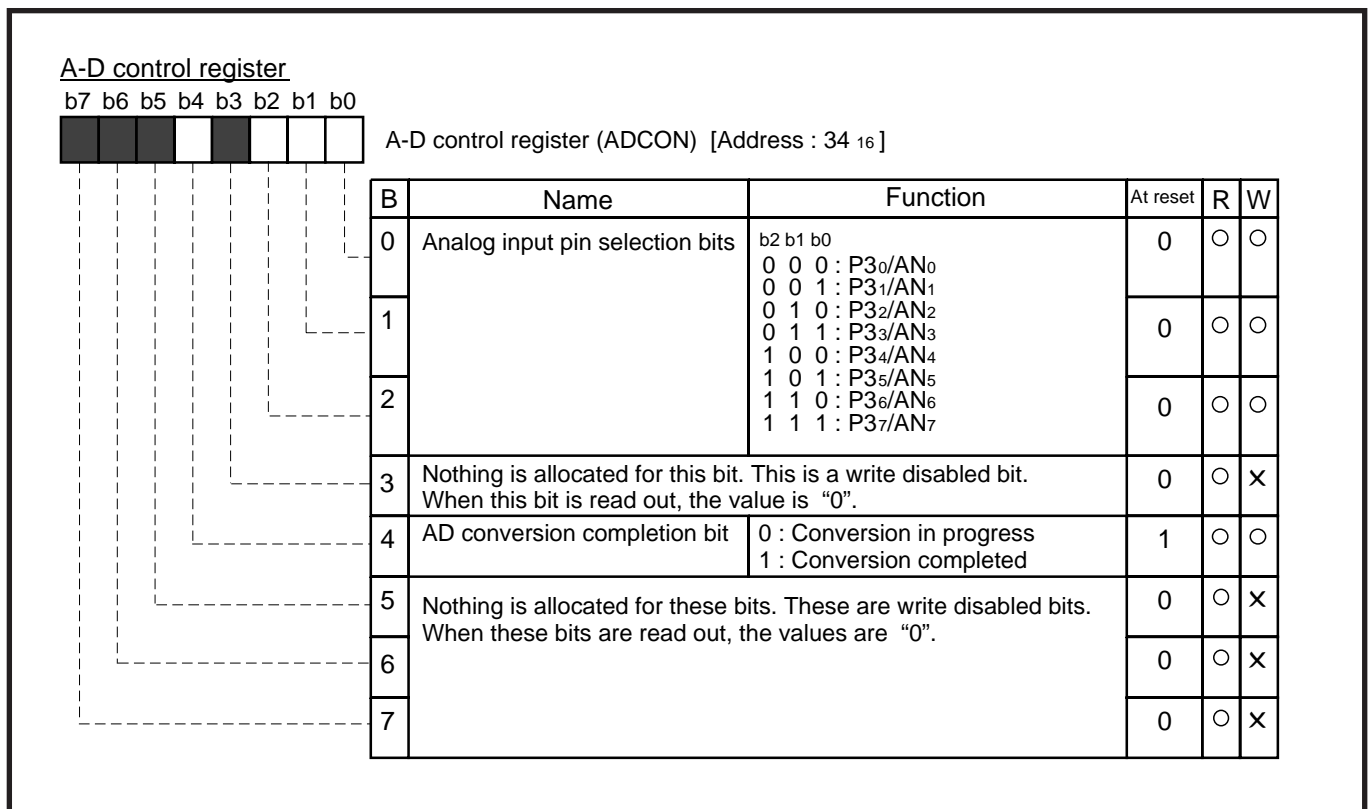


Fig. 3.5.22 Structure of A-D control register

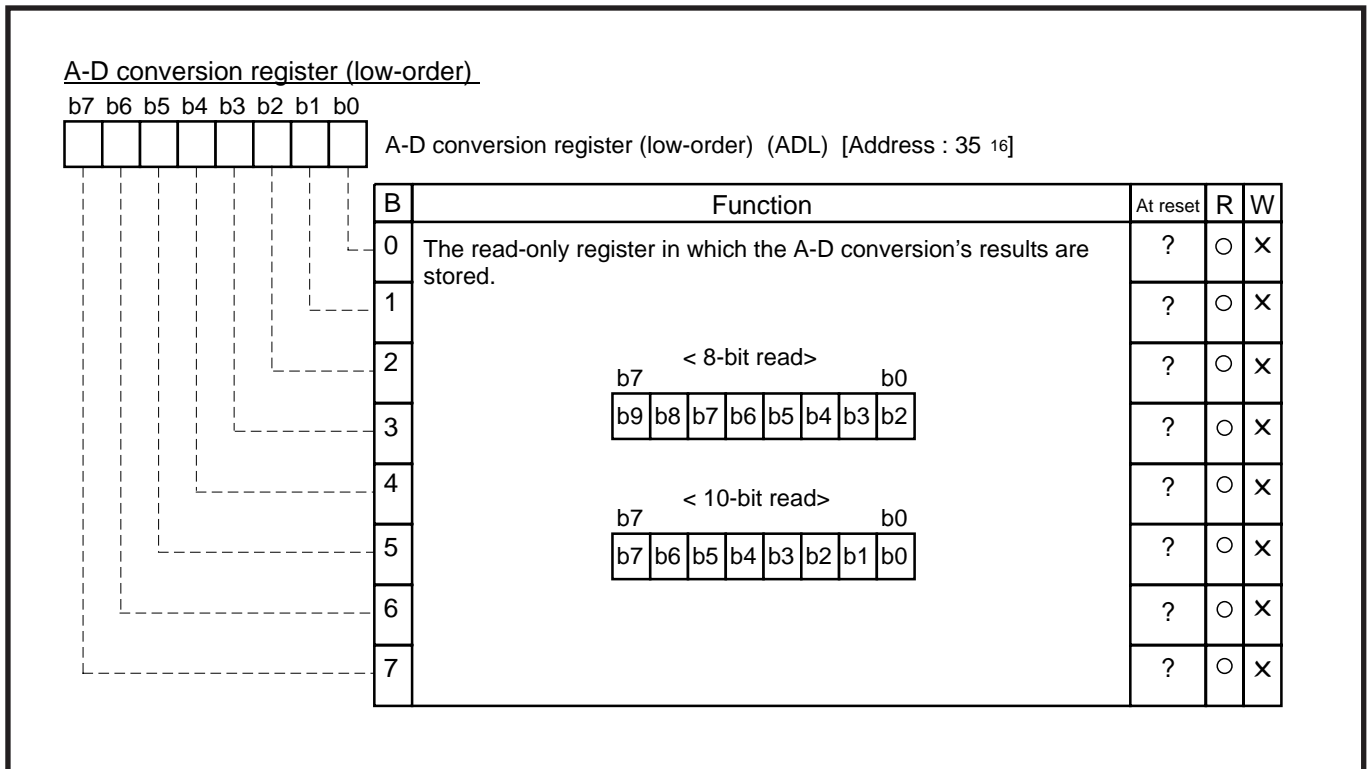


Fig. 3.5.23 Structure of A-D conversion register(low-order)

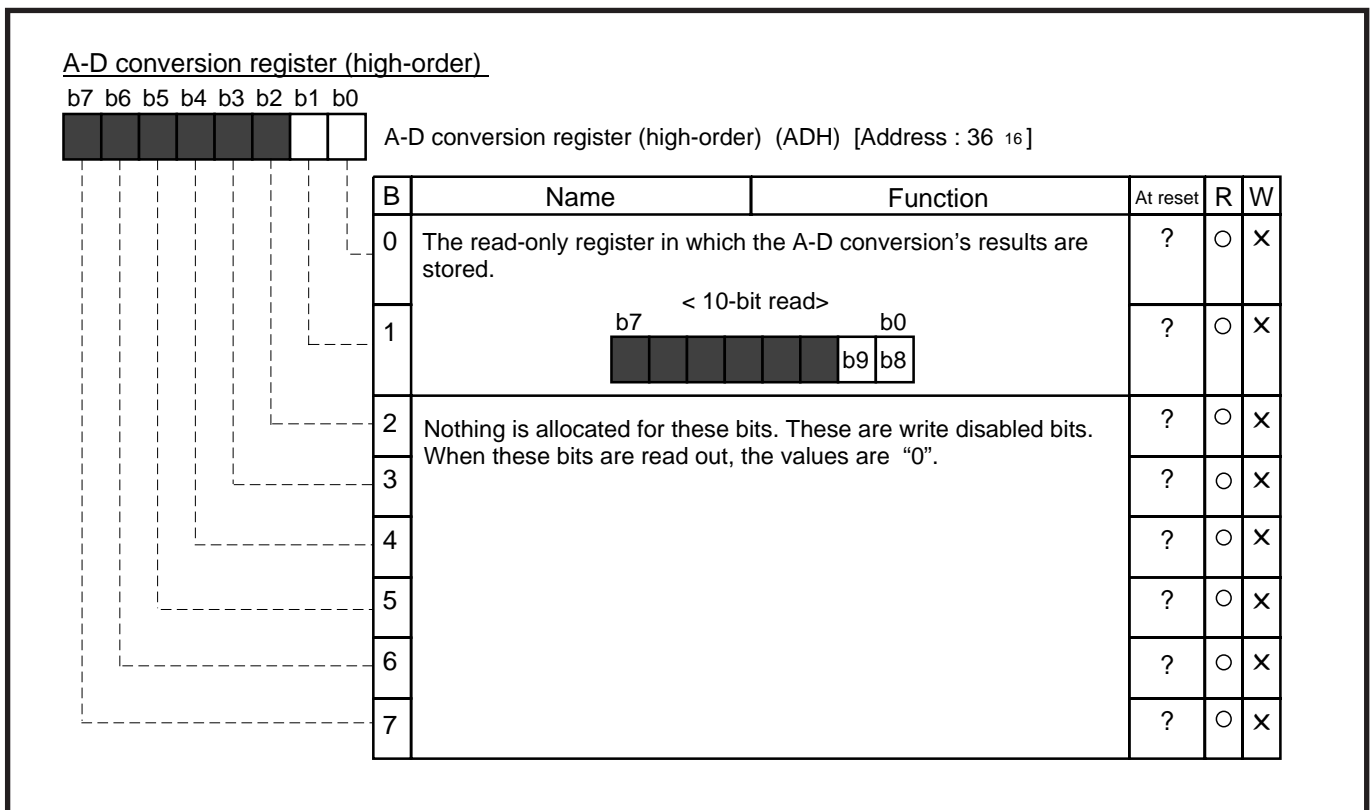


Fig. 3.5.24 Structure of A-D conversion register (high-order)

APPENDIX

3.5 List of registers

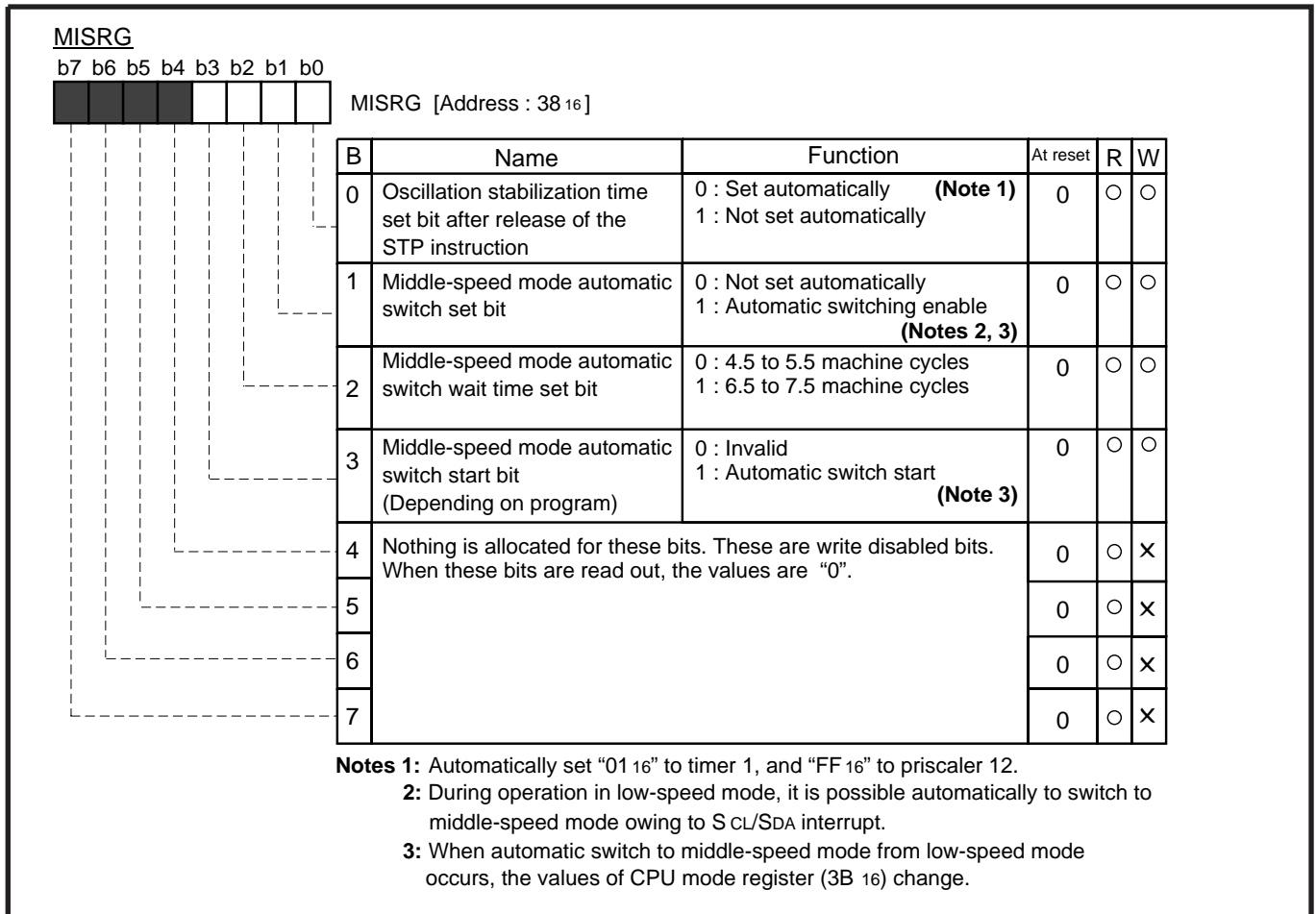


Fig. 3.5.25 Structure of MISRG

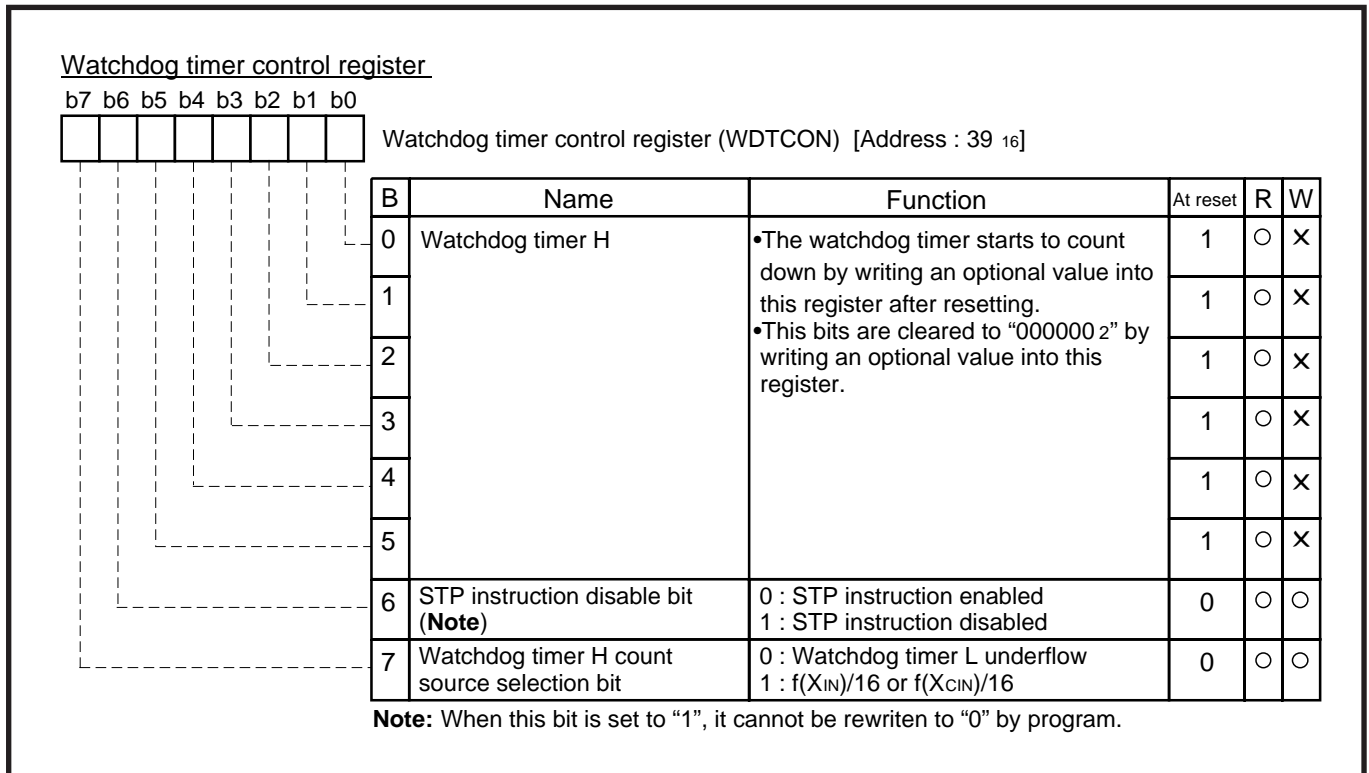
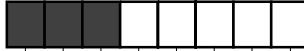


Fig. 3.5.26 Structure of Watchdog timer control register

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0



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

B	Name	Function	At reset	R	W
0	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	This is the reserved bit. Do not write "1" to this bit.		0	○	×
5	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.27 Structure of Interrupt edge selection register

CPU mode register

b7 b6 b5 b4 b3 b2 b1 b0



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

B	Name	Function	At reset	R	W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Not available 1 0 : Not available 1 1 : Not available	0	○	○
1			0 (Note)	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to "1".		1	○	○
4	Port Xc switch bit	0 : I/O port function 1 : Xcin-Xcout operating function	0	○	○
5	Main clock (X _{IN} -X _{OUT}) stop bit	0 : Operating 1 : Stopped	0	○	○
6	Main clock division ratio selection bits	b7 b6 0 0 : $\phi = f(X_{IN})/2$ (high-speed mode) 0 1 : $\phi = f(X_{IN})/8$ (middle-speed mode) 1 0 : $\phi = f(X_{CIN})/2$ (low-speed mode) 1 1 : $\phi =$ Not available	1	○	○
7			0	○	○

Note: An initial value of bit 1 depends on the CNV_{SS} pin level.

Fig. 3.5.28 Structure of CPU mode register

APPENDIX

3.5 List of registers

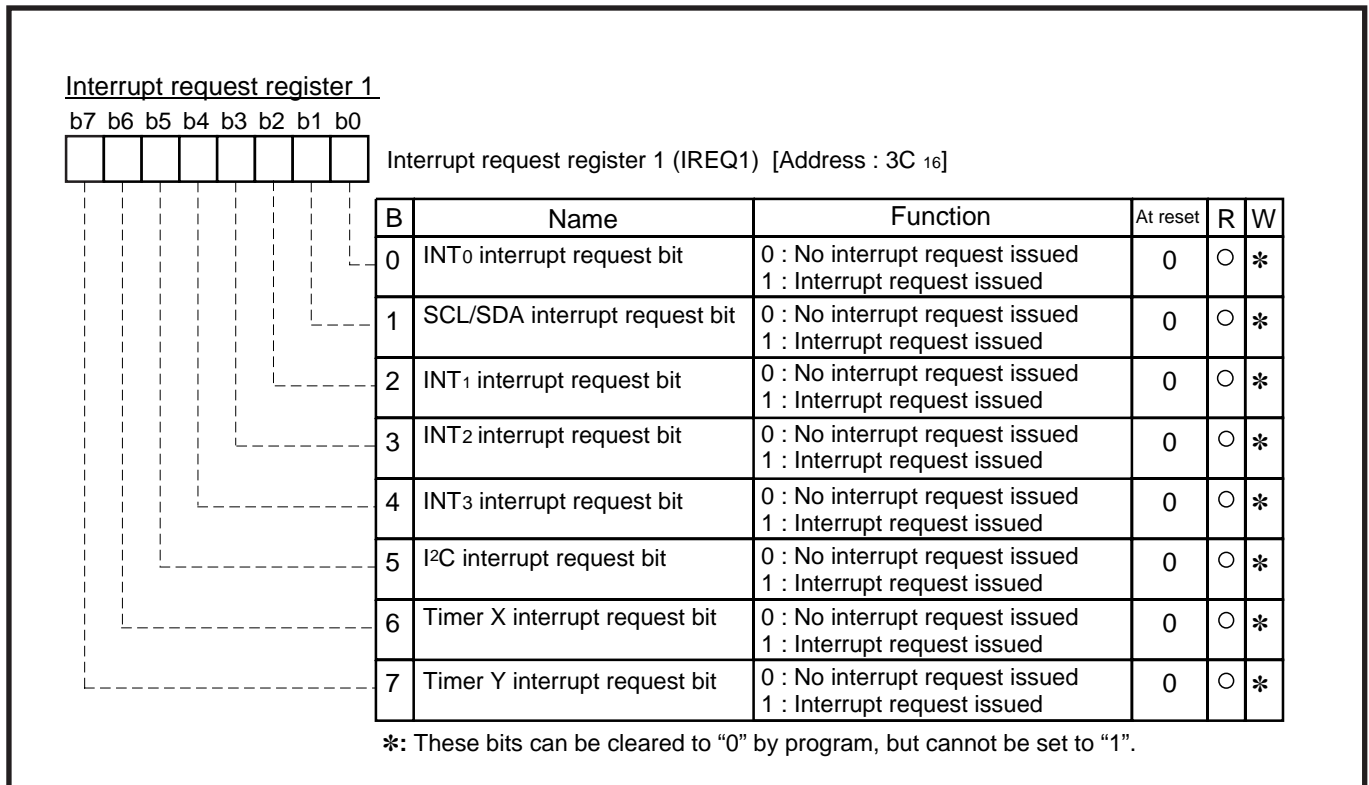


Fig. 3.5.29 Structure of Interrupt request register 1

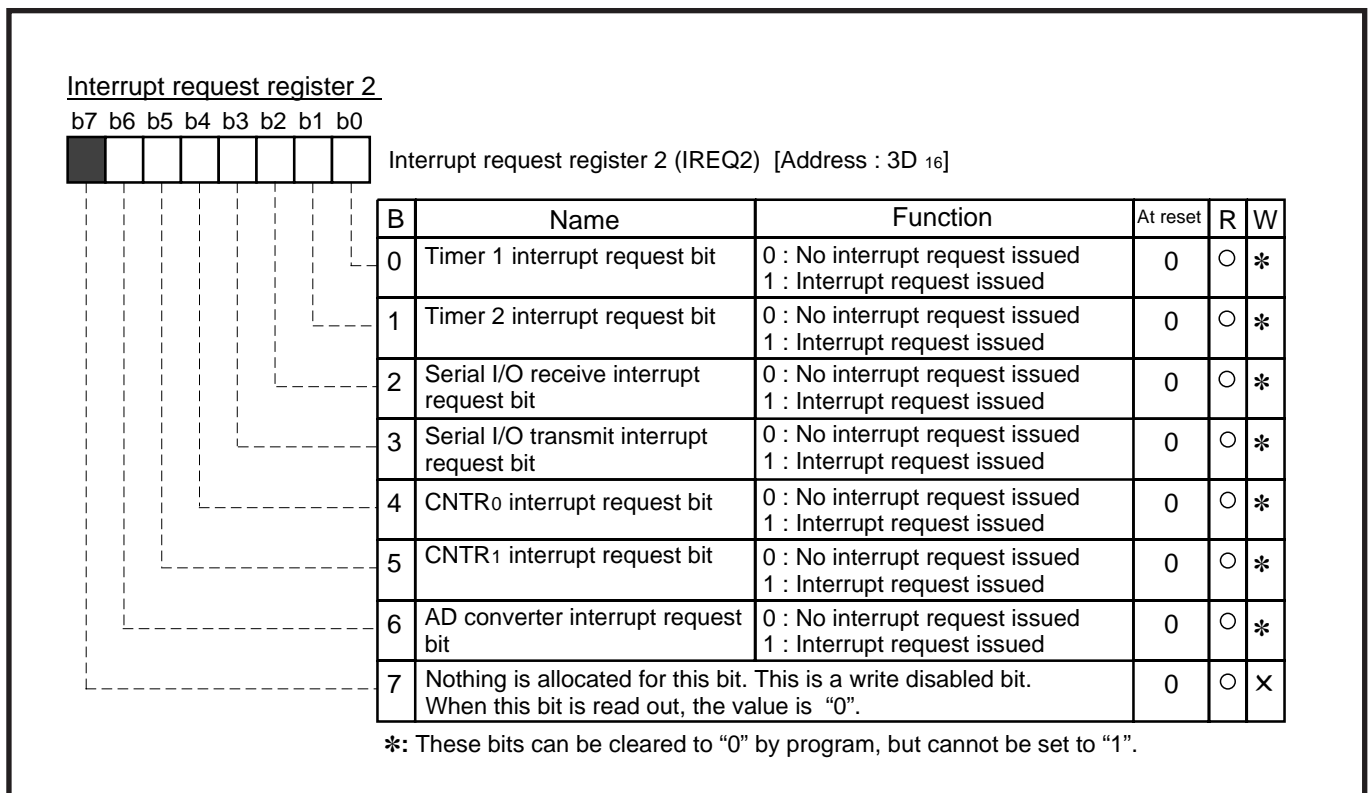


Fig. 3.5.30 Structure of Interrupt request register 2

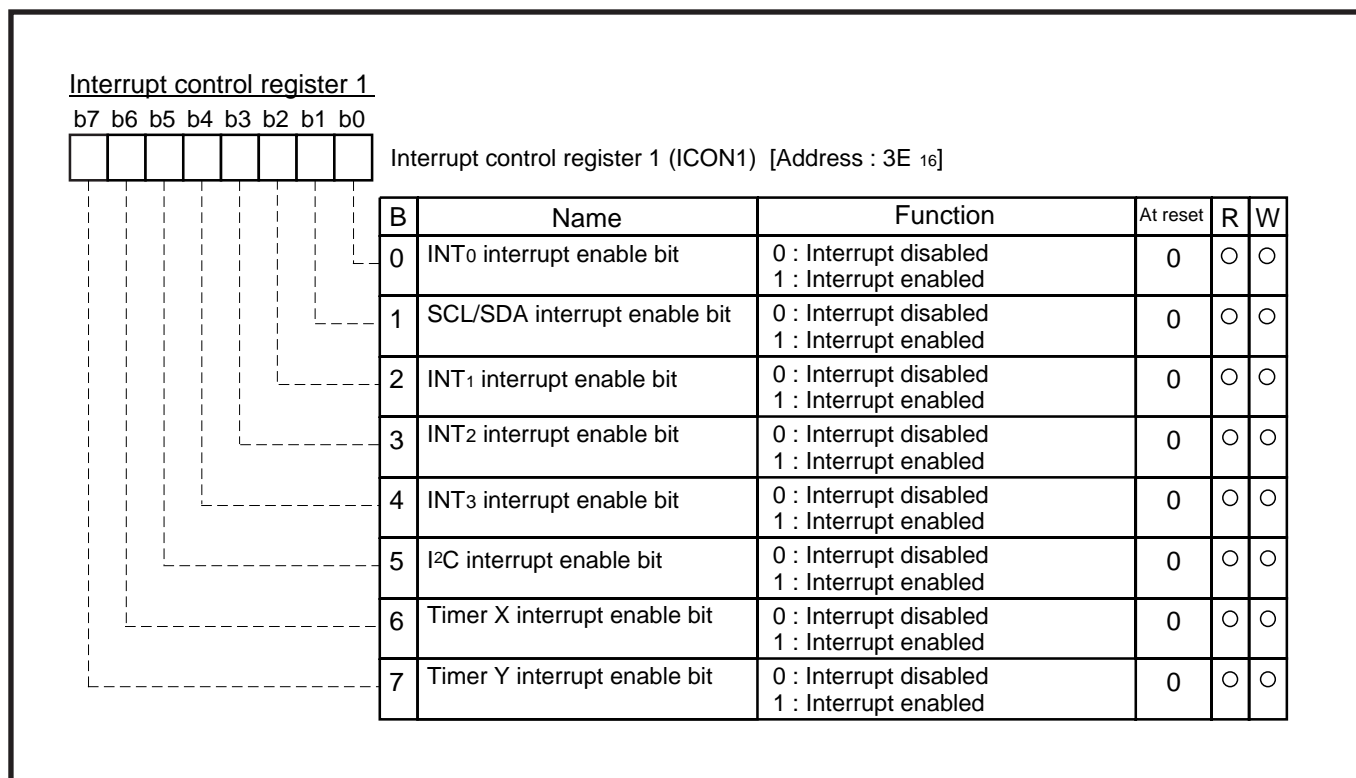


Fig. 3.5.31 Structure of Interrupt control register 1

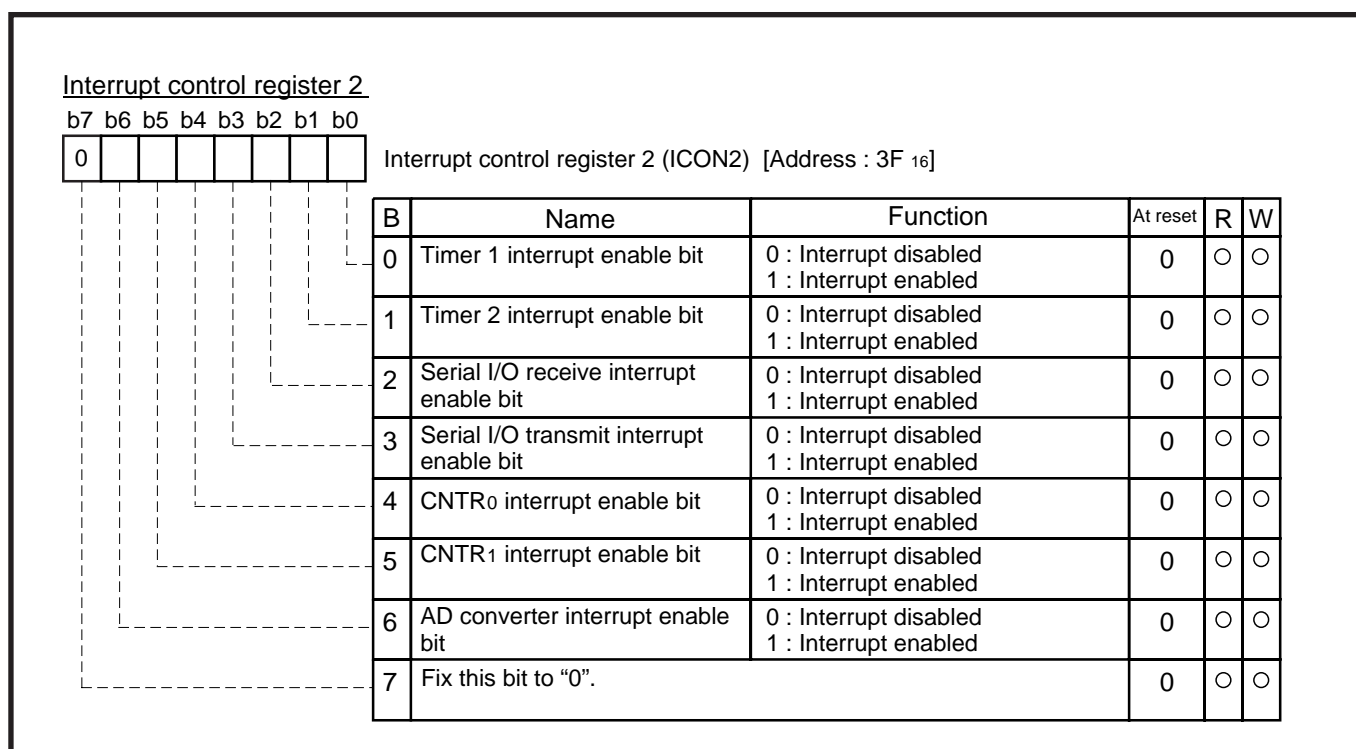


Fig. 3.5.32 Structure of Interrupt control register 2

APPENDIX

3.6 Mask ROM confirmation form

3.6 Mask ROM ordering method

GZZ-SH53-11B<86A0>

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

740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M38503M2-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.

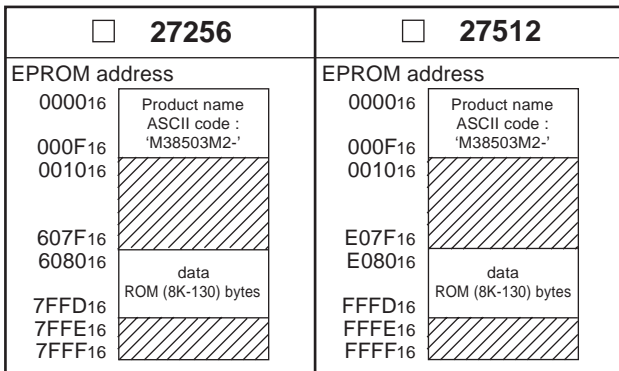
Microcomputer name: M38503M2-XXXSP M38503M2-XXXFP

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 E080₁₆ to FFFD₁₆. The reset vector is stored in addresses FFFC₁₆ and FFFD₁₆.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF₁₆".
- (2) The ASCII codes of the product name "M38503M2-" must be entered in addresses 0000₁₆ to 0008₁₆. And set the data "FF₁₆" in addresses 0009₁₆ to 000F₁₆. 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	'5' = 35 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	'2' = 32 ₁₆	000F ₁₆	FF ₁₆

APPENDIX

3.6 Mask ROM confirmation form

GZZ-SH11-40A<6YA0>

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

**740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38503M4-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.

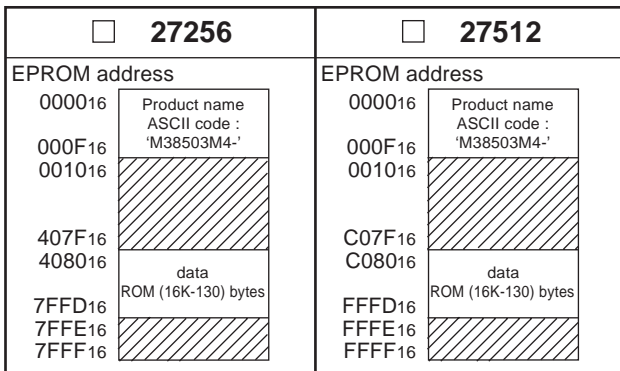
Microcomputer name: M38503M4-XXXSP M38503M4-XXXFP

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 C080₁₆ to FFFD₁₆. The reset vector is stored in addresses FFFC₁₆ and FFFD₁₆.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF₁₆".
- (2) The ASCII codes of the product name "M38503M4-" must be entered in addresses 0000₁₆ to 0008₁₆. And set the data "FF₁₆" in addresses 0009₁₆ to 000F₁₆. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address
0000 ₁₆	'M' = 4D ₁₆	0008 ₁₆
0001 ₁₆	'3' = 33 ₁₆	0009 ₁₆
0002 ₁₆	'8' = 38 ₁₆	000A ₁₆
0003 ₁₆	'5' = 35 ₁₆	000B ₁₆
0004 ₁₆	'0' = 30 ₁₆	000C ₁₆
0005 ₁₆	'3' = 33 ₁₆	000D ₁₆
0006 ₁₆	'M' = 4D ₁₆	000E ₁₆
0007 ₁₆	'4' = 34 ₁₆	000F ₁₆
		'_' = 2D ₁₆

GZZ-SH11-40A<6YA0>

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

**740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38503M4-XXXSP/FP
MITSUBISHI ELECTRIC**

We recommend the use of the following pseudo-command to set the start address of the assembler source program because ASCII codes of the product name are written to addresses 0000₁₆ to 0008₁₆ of EPROM.

EPROM type	27256	27512
The pseudo-command	*= △\$8000 .BYTE △'M38503M4-'	*= △\$0000 .BYTE △'M38503M4-'

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 (42P4B for M38503M4-XXXSP, 42P2R-A for M38503M4-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 XIN-XOUT oscillator?

- Ceramic resonator Quartz crystal
 External clock input Other ()

At what frequency? f(XIN) = MHz

(2) Which function will you use the pins P21/XCIN and P20/XCOUT as P21 and P20, or XCIN and XCOUT ?

- Ports P21 and P20 function XCIN and XCOUT function (external resonator)

※ 4. Comments

APPENDIX

3.6 Mask ROM confirmation form

GZZ-SH54-31B<89A0>

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

**740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38504M6-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 : M38504M6-XXXSP M38504M6-XXXFP

Checksum code for entire EPROM

--	--	--	--

 (hexadecimal notation)

EPROM type (indicate the type used)

<input type="checkbox"/> 27256	<input type="checkbox"/> 27512																						
<p>EPROM address</p> <table border="1" style="width: 100%;"> <tr><td>0000₁₆</td><td rowspan="3">Product name ASCII code : 'M38504M6-'</td></tr> <tr><td>000F₁₆</td></tr> <tr><td>0010₁₆</td></tr> <tr><td>207F₁₆</td><td rowspan="2">data ROM (24K-130) bytes</td></tr> <tr><td>2080₁₆</td></tr> <tr><td>7FFD₁₆</td><td rowspan="3">data ROM (24K-130) bytes</td></tr> <tr><td>7FFE₁₆</td></tr> <tr><td>7FFF₁₆</td></tr> </table>	0000 ₁₆	Product name ASCII code : 'M38504M6-'	000F ₁₆	0010 ₁₆	207F ₁₆	data ROM (24K-130) bytes	2080 ₁₆	7FFD ₁₆	data ROM (24K-130) bytes	7FFE ₁₆	7FFF ₁₆	<p>EPROM address</p> <table border="1" style="width: 100%;"> <tr><td>0000₁₆</td><td rowspan="3">Product name ASCII code : 'M38504M6-'</td></tr> <tr><td>000F₁₆</td></tr> <tr><td>0010₁₆</td></tr> <tr><td>A07F₁₆</td><td rowspan="2">data ROM (24K-130) bytes</td></tr> <tr><td>A080₁₆</td></tr> <tr><td>FFFD₁₆</td><td rowspan="3">data ROM (24K-130) bytes</td></tr> <tr><td>FFFE₁₆</td></tr> <tr><td>FFFF₁₆</td></tr> </table>	0000 ₁₆	Product name ASCII code : 'M38504M6-'	000F ₁₆	0010 ₁₆	A07F ₁₆	data ROM (24K-130) bytes	A080 ₁₆	FFFD ₁₆	data ROM (24K-130) bytes	FFFE ₁₆	FFFF ₁₆
0000 ₁₆	Product name ASCII code : 'M38504M6-'																						
000F ₁₆																							
0010 ₁₆																							
207F ₁₆	data ROM (24K-130) bytes																						
2080 ₁₆																							
7FFD ₁₆	data ROM (24K-130) bytes																						
7FFE ₁₆																							
7FFF ₁₆																							
0000 ₁₆	Product name ASCII code : 'M38504M6-'																						
000F ₁₆																							
0010 ₁₆																							
A07F ₁₆	data ROM (24K-130) bytes																						
A080 ₁₆																							
FFFD ₁₆	data ROM (24K-130) bytes																						
FFFE ₁₆																							
FFFF ₁₆																							

In the address space of the microcomputer, the internal ROM area is from address A080₁₆ to FFFD₁₆. The reset vector is stored in addresses FFFC₁₆ and FFFD₁₆.

- Set the data in the unused area (the shaded area of the diagram) to "FF₁₆".
- The ASCII codes of the product name "M38504M6—" must be entered in addresses 0000₁₆ to 0008₁₆. And set the data "FF₁₆" in addresses 0009₁₆ to 000F₁₆. 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	'5' = 35 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	'6' = 36 ₁₆	000F ₁₆	FF ₁₆

GZZ-SH54-31B<89A0>

Mask ROM number	<input type="text"/>
-----------------	----------------------

**740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38504M6-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	*= Δ \$8000 .BYTE Δ 'M38504M6-'	*= Δ \$0000 .BYTE Δ 'M38504M6-'

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 (42P4B for M38504M6-XXXSP , 42P2R for M38504M6-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 XIN-XOUT oscillator?

- Ceramic resonator Quartz crystal
 External clock input Other ()

At what frequency? f(XIN) = MHz

(2) Which function will you use the pins P21/XCIN and P20/XCOUT as P21 and P20, or XCIN and XCOUT ?

- Ports P21 and P20 function XCIN and XCOUT function (external resonator)

※ 4. Comments

APPENDIX

3.6 Mask ROM confirmation form

GZZ-SH52-61B<83A0>

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

**740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38513M4-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.

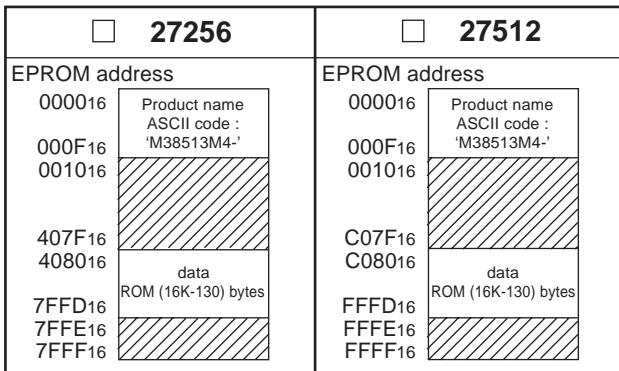
Microcomputer name: M38513M4-XXXSP M38513M4-XXXFP

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 C080₁₆ to FFFD₁₆. The reset vector is stored in addresses FFFC₁₆ and FFFD₁₆.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF₁₆".
- (2) The ASCII codes of the product name "M38513M4-" must be entered in addresses 0000₁₆ to 0008₁₆. And set the data "FF₁₆" in addresses 0009₁₆ to 000F₁₆. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address	Address	Address
0000 ₁₆	'M' = 4D ₁₆	0008 ₁₆
0001 ₁₆	'3' = 33 ₁₆	'_' = 2D ₁₆
0002 ₁₆	'8' = 38 ₁₆	0009 ₁₆
0003 ₁₆	'5' = 35 ₁₆	000A ₁₆
0004 ₁₆	'1' = 31 ₁₆	000B ₁₆
0005 ₁₆	'3' = 33 ₁₆	000C ₁₆
0006 ₁₆	'M' = 4D ₁₆	000D ₁₆
0007 ₁₆	'4' = 34 ₁₆	000E ₁₆
		000F ₁₆

GZZ-SH52-61B<83A0>

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

**740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38513M4-XXXSP/FP
MITSUBISHI ELECTRIC**

We recommend the use of the following pseudo-command to set the start address of the assembler source program because ASCII codes of the product name are written to addresses 0000₁₆ to 0008₁₆ of EPROM.

EPROM type	27256	27512
The pseudo-command	*=△\$8000 .BYTE △'M38513M4-'	*=△\$0000 .BYTE △'M38513M4-'

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 (42P4B for M38513M4-XXXSP, 42P2R for M38513M4-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 XIN-XOUT oscillator?

- Ceramic resonator Quartz crystal
 External clock input Other ()

At what frequency? f(XIN) = MHz

(2) Which function will you use the pins P21/XCIN and P20/XCOUT as P21 and P20, or XCIN and XCOUT ?

- Ports P21 and P20 function XCIN and XCOUT function (external resonator)

(3) Will you use the I²C-BUS function or the SM-BUS function ?

- I²C-BUS function used SM-BUS function used
 Not used

※ 4. Comments

APPENDIX

3.6 Mask ROM confirmation form

GZZ-SH54-32B<89A0>

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

**740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38514M6-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 : M38514M6-XXXSP M38514M6-XXXFP

Checksum code for entire EPROM

--	--	--	--

 (hexadecimal notation)

EPROM type (indicate the type used)

<input type="checkbox"/> 27256	<input type="checkbox"/> 27512																										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">EPROM address</td> <td style="width: 80%;"></td> </tr> <tr> <td>0000₁₆</td> <td rowspan="3" style="text-align: center; vertical-align: middle;">Product name ASCII code : 'M38514M6-'</td> </tr> <tr> <td>000F₁₆</td> </tr> <tr> <td>0010₁₆</td> </tr> <tr> <td>207F₁₆</td> <td rowspan="2" style="text-align: center; vertical-align: middle;">data ROM (24K-130) bytes</td> </tr> <tr> <td>2080₁₆</td> </tr> <tr> <td>7FFD₁₆</td> <td rowspan="3" style="text-align: center; vertical-align: middle;">data ROM (24K-130) bytes</td> </tr> <tr> <td>7FFE₁₆</td> </tr> <tr> <td>7FFF₁₆</td> </tr> </table>	EPROM address		0000 ₁₆	Product name ASCII code : 'M38514M6-'	000F ₁₆	0010 ₁₆	207F ₁₆	data ROM (24K-130) bytes	2080 ₁₆	7FFD ₁₆	data ROM (24K-130) bytes	7FFE ₁₆	7FFF ₁₆	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">EPROM address</td> <td style="width: 80%;"></td> </tr> <tr> <td>0000₁₆</td> <td rowspan="3" style="text-align: center; vertical-align: middle;">Product name ASCII code : 'M38514M6-'</td> </tr> <tr> <td>000F₁₆</td> </tr> <tr> <td>0010₁₆</td> </tr> <tr> <td>A07F₁₆</td> <td rowspan="2" style="text-align: center; vertical-align: middle;">data ROM (24K-130) bytes</td> </tr> <tr> <td>A080₁₆</td> </tr> <tr> <td>FFFD₁₆</td> <td rowspan="3" style="text-align: center; vertical-align: middle;">data ROM (24K-130) bytes</td> </tr> <tr> <td>FFFE₁₆</td> </tr> <tr> <td>FFFF₁₆</td> </tr> </table>	EPROM address		0000 ₁₆	Product name ASCII code : 'M38514M6-'	000F ₁₆	0010 ₁₆	A07F ₁₆	data ROM (24K-130) bytes	A080 ₁₆	FFFD ₁₆	data ROM (24K-130) bytes	FFFE ₁₆	FFFF ₁₆
EPROM address																											
0000 ₁₆	Product name ASCII code : 'M38514M6-'																										
000F ₁₆																											
0010 ₁₆																											
207F ₁₆	data ROM (24K-130) bytes																										
2080 ₁₆																											
7FFD ₁₆	data ROM (24K-130) bytes																										
7FFE ₁₆																											
7FFF ₁₆																											
EPROM address																											
0000 ₁₆	Product name ASCII code : 'M38514M6-'																										
000F ₁₆																											
0010 ₁₆																											
A07F ₁₆	data ROM (24K-130) bytes																										
A080 ₁₆																											
FFFD ₁₆	data ROM (24K-130) bytes																										
FFFE ₁₆																											
FFFF ₁₆																											

In the address space of the microcomputer, the internal ROM area is from address A080₁₆ to FFFD₁₆. The reset vector is stored in addresses FFFC₁₆ and FFFD₁₆.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF₁₆".
- (2) The ASCII codes of the product name "M38514M6—" must be entered in addresses 0000₁₆ to 0008₁₆. And set the data "FF₁₆" in addresses 0009₁₆ to 000F₁₆. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address
0000 ₁₆	'M' = 4D ₁₆	0008 ₁₆
0001 ₁₆	'3' = 33 ₁₆	0009 ₁₆
0002 ₁₆	'8' = 38 ₁₆	000A ₁₆
0003 ₁₆	'5' = 35 ₁₆	000B ₁₆
0004 ₁₆	'1' = 31 ₁₆	000C ₁₆
0005 ₁₆	'4' = 34 ₁₆	000D ₁₆
0006 ₁₆	'M' = 4D ₁₆	000E ₁₆
0007 ₁₆	'6' = 36 ₁₆	000F ₁₆
		'-' = 2D ₁₆
		FF ₁₆
		FF ₁₆
		FF ₁₆
		FF ₁₆
		FF ₁₆
		FF ₁₆
		FF ₁₆

GZZ-SH54-32B<89A0>

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

**740 FAMILY MASK ROM CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38514M6-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	*= Δ \$8000 .BYTE Δ 'M38514M6-'	*= Δ \$0000 .BYTE Δ 'M38514M6-'

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 (42P4B for M38514M6-XXXSP , 42P2R for M38514M6-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 XIN-XOUT oscillator?

- Ceramic resonator Quartz crystal
 External clock input Other ()

At what frequency? $f(XIN) =$ MHz

(2) Which function will you use the pins P21/XCIN and P20/XCOUT as P21 and P20, or XCIN and XCOUT ?

- Ports P21 and P20 function XCIN and XCOUT function (external resonator)

(3) Will you use the I²C-BUS function or the SM-BUS function ?

- I²C-BUS function used SM-BUS function used
 Not used

※ 4. Comments

APPENDIX

3.7 ROM programming confirmation form

3.7 ROM programming confirmation form

GZZ-SH11-41A<6YA0>

ROM number	
------------	--

**740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38503E4-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 ROM programming based on this data. We shall assume the responsibility for errors only if the programming data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

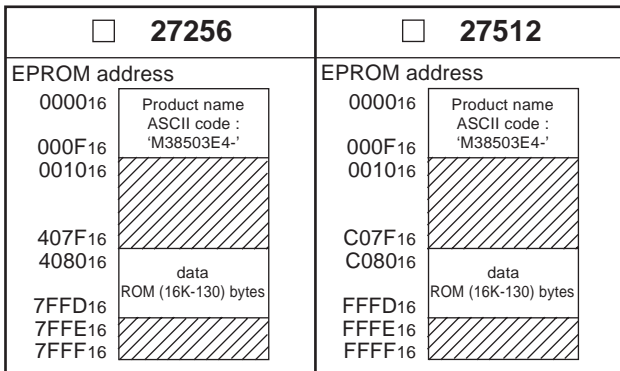
Microcomputer name: M38503E4-XXXSP M38503E4-XXXFP

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 C080₁₆ to FFFD₁₆. The reset vector is stored in addresses FFFC₁₆ and FFFD₁₆.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF₁₆".
- (2) The ASCII codes of the product name "M38503E4-" must be entered in addresses 0000₁₆ to 0008₁₆. And set the data "FF₁₆" in addresses 0009₁₆ to 000F₁₆. 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	'5' = 35 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	'E' = 45 ₁₆	000E ₁₆	FF ₁₆
0007 ₁₆	'4' = 34 ₁₆	000F ₁₆	FF ₁₆

(1/2)

APPENDIX

3.7 ROM programming confirmation form

GZZ-SH53-16B<86A0>

ROM number	
------------	--

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38504E6-XXXFP/SP
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 ROM programming based on this data. We shall assume the responsibility for errors only if the programming data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

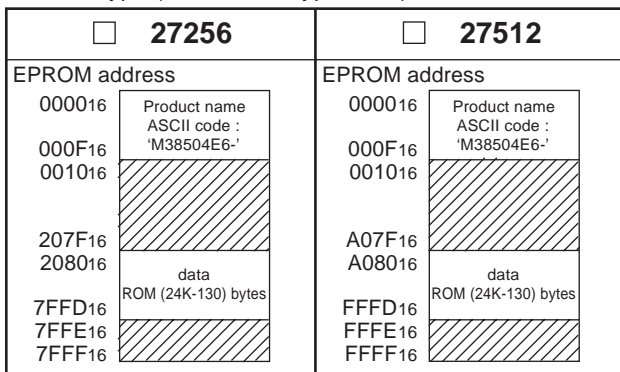
Product name: M38504E6-XXXFP M38504E6-XXXSP

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 A080₁₆ to FFFD₁₆. The reset vector is stored in addresses FFFC₁₆ and FFFD₁₆.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF₁₆".
- (2) The ASCII codes of the product name "M38504E6-" must be entered in addresses 0000₁₆ to 0008₁₆. And set the data "FF₁₆" in addresses 0009₁₆ to 000F₁₆. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address	Address	Address
0000 ₁₆	'M' = 4D ₁₆	0008 ₁₆
0001 ₁₆	'3' = 33 ₁₆	'-' = 2D ₁₆
0002 ₁₆	'8' = 38 ₁₆	0009 ₁₆
0003 ₁₆	'5' = 35 ₁₆	FF ₁₆
0004 ₁₆	'0' = 30 ₁₆	000A ₁₆
0005 ₁₆	'4' = 34 ₁₆	FF ₁₆
0006 ₁₆	'E' = 45 ₁₆	000B ₁₆
0007 ₁₆	'6' = 36 ₁₆	FF ₁₆
		000C ₁₆
		FF ₁₆
		000D ₁₆
		FF ₁₆
		000E ₁₆
		FF ₁₆
		000F ₁₆
		FF ₁₆

(1/2)

GZZ-SH53-16B<86A0>

ROM number	
------------	--

**740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38504E6-XXXFP/SP
MITSUBISHI ELECTRIC**

We recommend the use of the following pseudo-command to set the start address of the assembler source program because ASCII codes of the product name are written to addresses 0000₁₆ to 0008₁₆ of EPROM.

EPROM type	27256	27512
The pseudo-command	*= △ \$8000 .BYTE △ 'M38504E6-'	*= △ \$0000 .BYTE △ 'M38504E6-'

Note : If the name of the product written to the EPROMs does not match the name of the ROM programming 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; 42P2R for the M38504E6-XXXFP, the shrink DIP package Mark Specification Form (only for built-in One Time PROM microcomputer) for the M38504E6-XXXSP; and attach it to the ROM programming confirmation form.

※ 3. Usage conditions

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

(1) How will you use the XIN-XOUT oscillator?

- Ceramic resonator Quartz crystal
 External clock input Other ()

At what frequency? f(XIN) = MHz

(2) Which function will you use the pins P21/XCIN and P20/XCOUT as P21 and P20, or XCIN and XCOUT ?

- Ports P21 and P20 function XCIN and XCOUT function (external resonator)

※ 4. Comments

APPENDIX

3.7 ROM programming confirmation form

GZZ-SH11-43A<6YB0>

ROM number	
------------	--

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M38513E4-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 ROM programming based on this data. We shall assume the responsibility for errors only if the programming data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

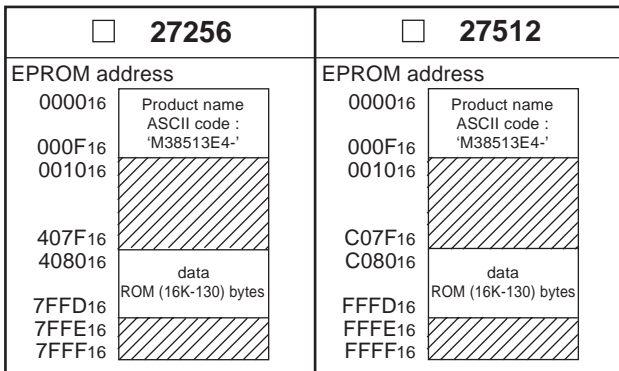
Microcomputer name: M38513E4-XXXSP M38513E4-XXXFP

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 C080₁₆ to FFFD₁₆. The reset vector is stored in addresses FFFC₁₆ and FFFD₁₆.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF₁₆".
- (2) The ASCII codes of the product name "M38513E4-" must be entered in addresses 0000₁₆ to 0008₁₆. And set the data "FF₁₆" in addresses 0009₁₆ to 000F₁₆. 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	'5' = 35 ₁₆	000B ₁₆	FF ₁₆
0004 ₁₆	'1' = 31 ₁₆	000C ₁₆	FF ₁₆
0005 ₁₆	'3' = 33 ₁₆	000D ₁₆	FF ₁₆
0006 ₁₆	'E' = 45 ₁₆	000E ₁₆	FF ₁₆
0007 ₁₆	'4' = 34 ₁₆	000F ₁₆	FF ₁₆

GZZ-SH11-43A<6YB0>

ROM number	
------------	--

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M38513E4-XXXSP/FP MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program because ASCII codes of the product name are written to addresses 0000₁₆ to 0008₁₆ of EPROM.

EPROM type	27256	27512
The pseudo-command	*=△\$8000 .BYTE △'M38513E4-'	*=△\$0000 .BYTE △'M38513E4-'

Note : If the name of the product written to the EPROMs does not match the name of the ROM programming 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; 42P2R for the M38513E4-XXXFP, the shrink DIP package Mark Specification Form (only for built-in One Time PROM microcomputer) for the M38513E4-XXXSP; and attach it to the ROM programming confirmation form.

* 3. Usage conditions

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

(1) How will you use the XIN-XOUT oscillator?

- Ceramic resonator Quartz crystal
 External clock input Other ()

At what frequency? $f(XIN) =$ MHz

(2) Which function will you use the pins P21/XCIN and P20/XCOUT as P21 and P20, or XCIN and XCOUT ?

- Ports P21 and P20 function XCIN and XCOUT function (external resonator)

(3) Will you use the I²C-BUS function or the SM-BUS function ?

- I²C-BUS function used SM-BUS function used
 Not used

* 4. Comments

(2/2)

APPENDIX

3.7 ROM programming confirmation form

GZZ-SH53-16B<86A0>

ROM number	
------------	--

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M38514E6-XXXFP/SP
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 ROM programming based on this data. We shall assume the responsibility for errors only if the programming data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

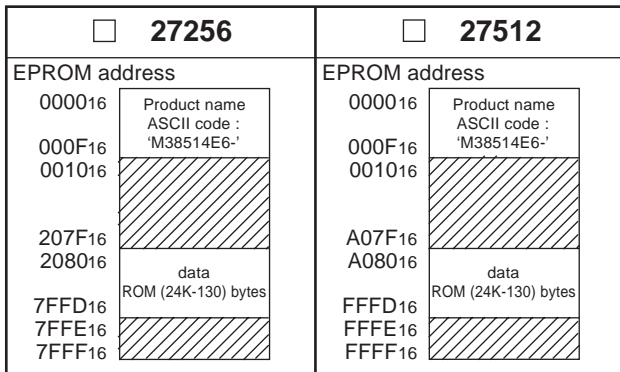
Product name: M38514E6-XXXFP M38514E6-XXXSP

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 A080₁₆ to FFFD₁₆. The reset vector is stored in addresses FFFC₁₆ and FFFD₁₆.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF₁₆".
- (2) The ASCII codes of the product name "M38514E6-" must be entered in addresses 0000₁₆ to 0008₁₆. And set the data "FF₁₆" in addresses 0009₁₆ to 000F₁₆. The ASCII codes and addresses are listed to the right in hexadecimal notation.

Address		Address
0000 ₁₆	'M' = 4D ₁₆	0008 ₁₆
0001 ₁₆	'3' = 33 ₁₆	0009 ₁₆
0002 ₁₆	'8' = 38 ₁₆	000A ₁₆
0003 ₁₆	'5' = 35 ₁₆	000B ₁₆
0004 ₁₆	'1' = 31 ₁₆	000C ₁₆
0005 ₁₆	'4' = 34 ₁₆	000D ₁₆
0006 ₁₆	'E' = 45 ₁₆	000E ₁₆
0007 ₁₆	'6' = 36 ₁₆	000F ₁₆
		'-' = 2D ₁₆
		FF ₁₆
		FF ₁₆
		FF ₁₆
		FF ₁₆
		FF ₁₆
		FF ₁₆

(1/2)

GZZ-SH53-16B<86A0>

ROM number	
------------	--

740 FAMILY ROM PROGRAMMING CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M38514E6-XXXFP/SP MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program because ASCII codes of the product name are written to addresses 0000₁₆ to 0008₁₆ of EPROM.

EPROM type	27256	27512
The pseudo-command	*= Δ \$8000 .BYTE Δ 'M38514E6-'	*= Δ \$0000 .BYTE Δ 'M38514E6-'

Note : If the name of the product written to the EPROMs does not match the name of the ROM programming 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; 42P2R for the M38514E6-XXXFP, the shrink DIP package Mark Specification Form (only for built-in One Time PROM microcomputer) for the M38514E6-XXXSP; and attach it to the ROM programming confirmation form.

* 3. Usage conditions

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

(1) How will you use the XIN-XOUT oscillator?

- Ceramic resonator Quartz crystal
 External clock input Other ()

At what frequency? $f(XIN) =$ MHz

(2) Which function will you use the pins P21/XCIN and P20/XCOUT as P21 and P20, or XCIN and XCOUT ?

- Ports P21 and P20 function XCIN and XCOUT function (external resonator)

(3) Will you use the I²C-BUS function or the SM-BUS function ?

- I²C-BUS function used SM-BUS function used
 Not used

* 4. Comments

(2/2)

APPENDIX

3.8 Mark specification form

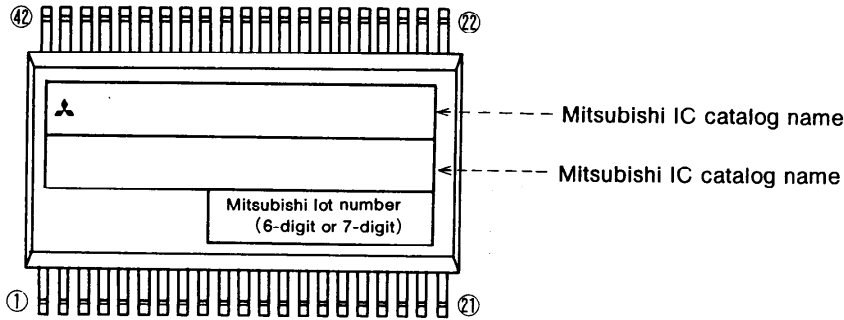
3.8 Mark specification form

42P2R-A (42-PIN SHRINK SOP) MARK SPECIFICATION FORM

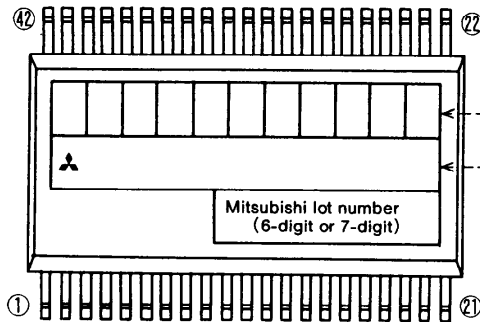
Mitsubishi IC catalog name

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

A. Standard Mitsubishi Mark



B. Customer's Parts Number + Mitsubishi catalog name



Customer's Parts Number

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

Mitsubishi IC catalog name

Note1: The mark field should be written right aligned.

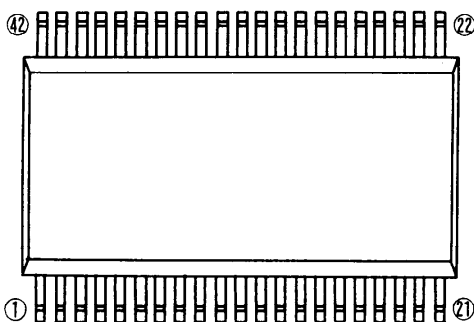
2: The fonts and size of characters are standard Mitsubishi type.

3: Customer's Parts Number can be up to 11 characters: Only 0~9, A~Z, +, -, /, (,), &, ©, · (periods), , (commas) are usable.

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

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 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

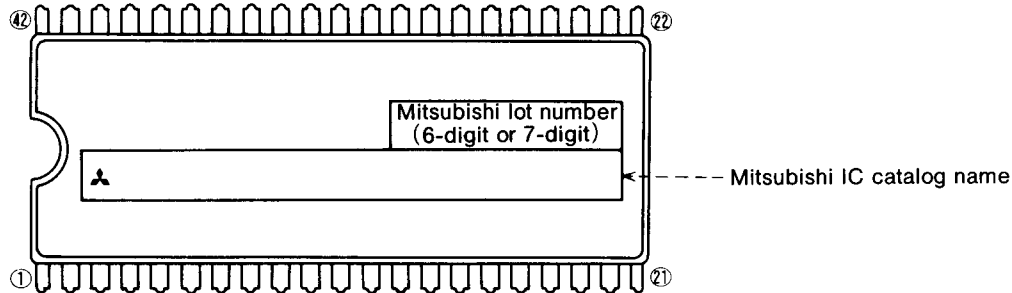
3: The standard Mitsubishi font is used for all characters except for a logo.

42P4B (42-PIN SHRINK DIP) MARK SPECIFICATION FORM

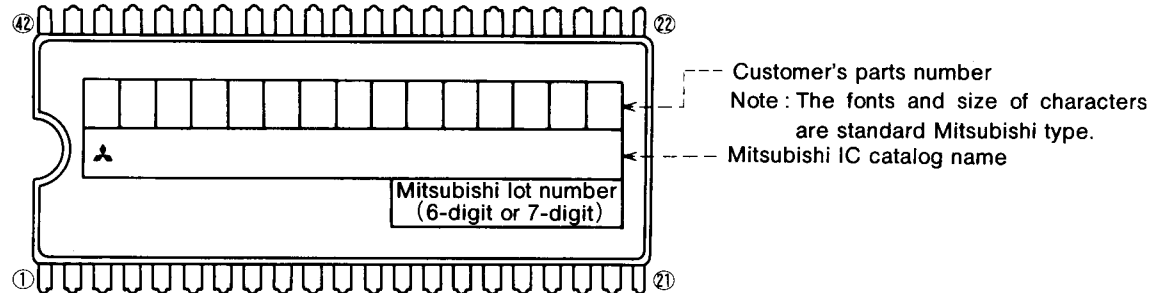
Mitsubishi IC catalog name

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.

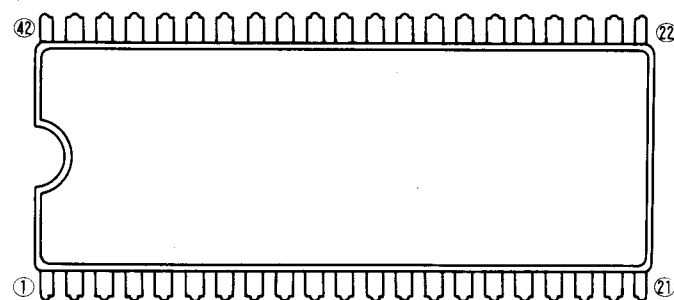
3 : Customer's parts number can be up to 15 characters :

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.

APPENDIX

3.9 Package outline

3.9 Package outline

42P2R-A

Plastic 42pin 450mil SSOP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SSOP42-P-450-0.80	-	0.63	Alloy 42/Cu Alloy

Caution !
TBD

42P2R-A package outline is to be updated.

L1	-	1.765	-
y	-	-	0.15
θ	0°	-	10°
b2	-	0.5	-
e1	-	11.43	-
l2	1.27	-	-

42P4B

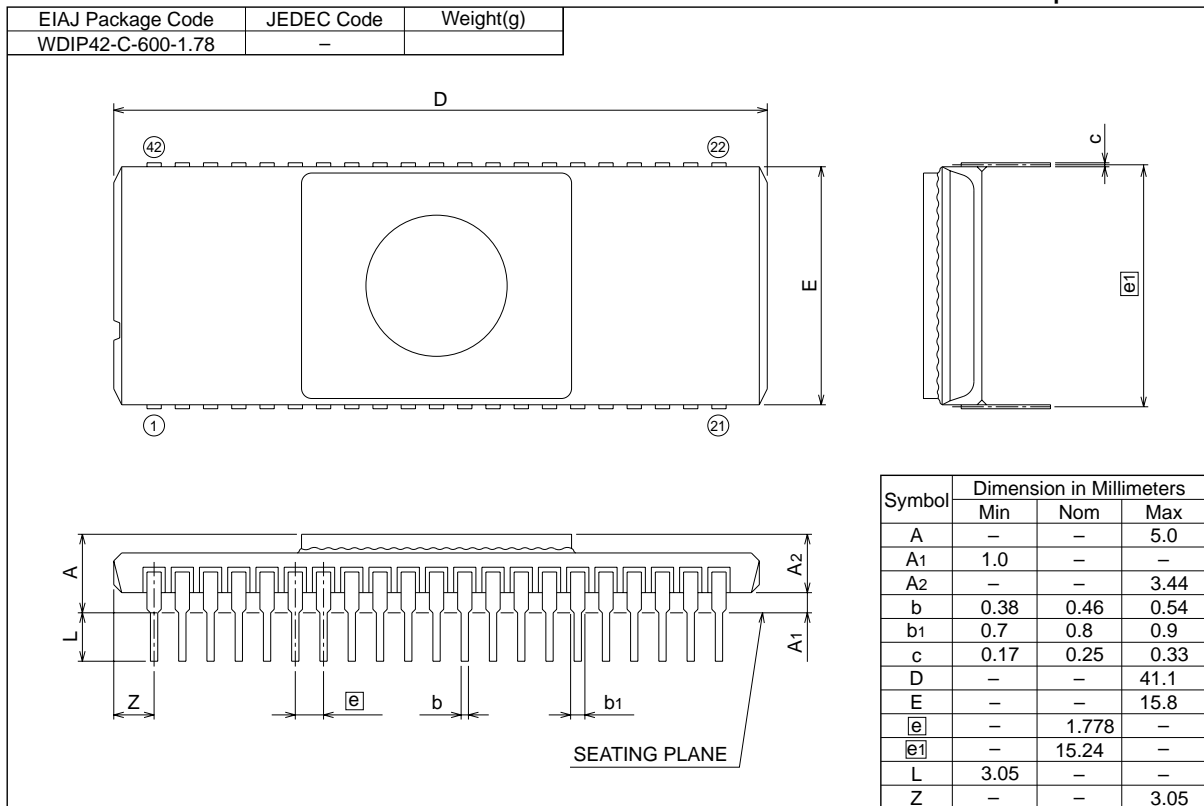
Plastic 42pin 600mil SDIP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SDIP42-P-600-1.78	-	4.1	Alloy 42/Cu Alloy

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	-	-	5.5
A1	0.51	-	-
A2	-	3.8	-
b	0.35	0.45	0.55
b1	0.9	1.0	1.3
b2	0.63	0.73	1.03
c	0.22	0.27	0.34
D	36.5	36.7	36.9
E	12.85	13.0	13.15
e	-	1.778	-
e1	-	15.24	-
L	3.0	-	-
θ	0°	-	15°

42S1B-A

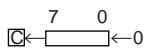
Metal seal 42pin 600mil DIP



APPENDIX

3.10 Machine instructions

3.10 Machine instructions

Symbol	Function	Details	Addressing mode																				
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R					
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#			
ADC (Note 1) (Note 5)	When T = 0 $A \leftarrow A + M + C$ When T = 1 $M(X) \leftarrow M(X) + M + C$	When T = 0, this instruction adds the contents M, C, and A; and stores the results in A and C. When T = 1, this instruction adds the contents of M(X), M and C; and stores the results in M(X) and C. When T=1, the contents of A remain unchanged, but the contents of status flags are changed. M(X) represents the contents of memory where is indicated by X.				69	2	2										65	3	2			
AND (Note 1)	When T = 0 $A \leftarrow A \wedge M$ When T = 1 $M(X) \leftarrow M(X) \wedge M$	When T = 0, this instruction transfers the contents of A and M to the ALU which performs a bit-wise AND operation and stores the result back in A. When T = 1, this instruction transfers the contents M(X) and M to the ALU which performs a bit-wise AND operation and stores the results back in M(X). When T = 1 the contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				29	2	2										25	3	2			
ASL		This instruction shifts the content of A or M by one bit to the left, with bit 0 always being set to 0 and bit 7 of A or M always being contained in C.							0A	2	1							06	5	2			
BBC (Note 4)	A_i or $M_i = 0?$	This instruction tests the designated bit i of M or A and takes a branch if the bit is 0. The branch address is specified by a relative address. If the bit is 1, next instruction is executed.										13 $20i$	4	2				17 $20i$	5	3			
BBS (Note 4)	A_i or $M_i = 1?$	This instruction tests the designated bit i of the M or A and takes a branch if the bit is 1. The branch address is specified by a relative address. If the bit is 0, next instruction is executed.										03 $20i$	4	2				07 $20i$	5	3			
BCC (Note 4)	$C = 0?$	This instruction takes a branch to the appointed address if C is 0. The branch address is specified by a relative address. If C is 1, the next instruction is executed.																					
BCS (Note 4)	$C = 1?$	This instruction takes a branch to the appointed address if C is 1. The branch address is specified by a relative address. If C is 0, the next instruction is executed.																					
BEQ (Note 4)	$Z = 1?$	This instruction takes a branch to the appointed address when Z is 1. The branch address is specified by a relative address. If Z is 0, the next instruction is executed.																					
BIT	$A \wedge M$	This instruction takes a bit-wise logical AND of A and M contents; however, the contents of A and M are not modified. The contents of N, V, Z are changed, but the contents of A, M remain unchanged.																24	3	2			
BMI (Note 4)	$N = 1?$	This instruction takes a branch to the appointed address when N is 1. The branch address is specified by a relative address. If N is 0, the next instruction is executed.																					
BNE (Note 4)	$Z = 0?$	This instruction takes a branch to the appointed address if Z is 0. The branch address is specified by a relative address. If Z is 1, the next instruction is executed.																					

APPENDIX

3.10 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	#
DEX	$X \leftarrow X - 1$	This instruction subtracts one from the current contents of X.	CA	2	1															
DEY	$Y \leftarrow Y - 1$	This instruction subtracts one from the current contents of Y.	88	2	1															
DIV	$A \leftarrow (M(zz + X + 1), M(zz + X)) / A$ $M(S) \leftarrow \text{one's complement of Remainder}$ $S \leftarrow S - 1$	Divides the 16-bit data in $M(zz+(X))$ (low-order byte) and $M(zz+(X)+1)$ (high-order byte) by the contents of A. The quotient is stored in A and the one's complement of the remainder is pushed onto the stack.																		
EOR (Note 1)	When $T = 0$ $A \leftarrow A \vee M$ When $T = 1$ $M(X) \leftarrow M(X) \vee M$	When $T = 0$, this instruction transfers the contents of the M and A to the ALU which performs a bit-wise Exclusive OR, and stores the result in A. When $T = 1$, the contents of $M(X)$ and M are transferred to the ALU, which performs a bit-wise Exclusive OR and stores the results in $M(X)$. The contents of A remain unchanged, but status flags are changed. $M(X)$ represents the contents of memory where is indicated by X.				49	2	2							45	3	2			
INC	$A \leftarrow A + 1$ or $M \leftarrow M + 1$	This instruction adds one to the contents of A or M.							3A	2	1				E6	5	2			
INX	$X \leftarrow X + 1$	This instruction adds one to the contents of X.	E8	2	1															
INY	$Y \leftarrow Y + 1$	This instruction adds one to the contents of Y.	C8	2	1															
JMP	If addressing mode is ABS $PCL \leftarrow ADL$ $PCH \leftarrow ADH$ If addressing mode is IND $PCL \leftarrow M(ADH, ADL)$ $PCH \leftarrow M(ADH, ADL + 1)$ If addressing mode is ZP, IND $PCL \leftarrow M(00, ADL)$ $PCH \leftarrow M(00, ADL + 1)$	This instruction jumps to the address designated by the following three addressing modes: Absolute Indirect Absolute Zero Page Indirect Absolute																		
JSR	$M(S) \leftarrow PCH$ $S \leftarrow S - 1$ $M(S) \leftarrow PCL$ $S \leftarrow S - 1$ After executing the above, if addressing mode is ABS, $PCL \leftarrow ADL$ $PCH \leftarrow ADH$ if addressing mode is SP, $PCL \leftarrow ADL$ $PCH \leftarrow FF$ If addressing mode is ZP, IND, $PCL \leftarrow M(00, ADL)$ $PCH \leftarrow M(00, ADL + 1)$	This instruction stores the contents of the PC in the stack, then jumps to the address designated by the following addressing modes: Absolute Special Page Zero Page Indirect Absolute																		
LDA (Note 2)	When $T = 0$ $A \leftarrow M$ When $T = 1$ $M(X) \leftarrow M$	When $T = 0$, this instruction transfers the contents of M to A. When $T = 1$, this instruction transfers the contents of M to $M(X)$. The contents of A remain unchanged, but status flags are changed. $M(X)$ represents the contents of memory where is indicated by X.				A9	2	2							A5	3	2			
LDM	$M \leftarrow nn$	This instruction loads the immediate value in M.													3C	4	3			
LDX	$X \leftarrow M$	This instruction loads the contents of M in X.				A2	2	2							A6	3	2			
LDY	$Y \leftarrow M$	This instruction loads the contents of M in Y.				A0	2	2							A4	3	2			

APPENDIX

3.10 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	#		
LSR	$0 \rightarrow \begin{array}{ c } \hline 7 \quad 0 \\ \hline \end{array} \rightarrow \text{C}$	This instruction shifts either A or M one bit to the right such that bit 7 of the result always is set to 0, and the bit 0 is stored in C.							4A	2	1					46	5	2				
MUL	$M(S) \cdot A \leftarrow A * M(zz + X)$ $S \leftarrow S - 1$	Multiplies Accumulator with the memory specified by the Zero Page X address mode and stores the high-order byte of the result on the Stack and the low-order byte in A.																				
NOP	$PC \leftarrow PC + 1$	This instruction adds one to the PC but does no other operation.	EA	2	1																	
ORA (Note 1)	When T = 0 $A \leftarrow A \vee M$ When T = 1 $M(X) \leftarrow M(X) \vee M$	When T = 0, this instruction transfers the contents of A and M to the ALU which performs a bit-wise "OR", and stores the result in A. When T = 1, this instruction transfers the contents of M(X) and the M to the ALU which performs a bit-wise OR, and stores the result in M(X). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				09	2	2							05	3	2					
PHA	$S \leftarrow S - 1$	This instruction pushes the contents of A to the memory location designated by S, and decrements the contents of S by one.	48	3	1																	
PHP	$M(S) \leftarrow PS$ $S \leftarrow S - 1$	This instruction pushes the contents of PS to the memory location designated by S and decrements the contents of S by one.	08	3	1																	
PLA	$S \leftarrow S + 1$ $A \leftarrow M(S)$	This instruction increments S by one and stores the contents of the memory designated by S in A.	68	4	1																	
PLP	$S \leftarrow S + 1$ $PS \leftarrow M(S)$	This instruction increments S by one and stores the contents of the memory location designated by S in PS.	28	4	1																	
ROL	$\begin{array}{ c } \hline 7 \quad 0 \\ \hline \end{array} \leftarrow \text{C}$	This instruction shifts either A or M one bit left through C. C is stored in bit 0 and bit 7 is stored in C.							2A	2	1				26	5	2					
ROR	$\text{C} \rightarrow \begin{array}{ c } \hline 7 \quad 0 \\ \hline \end{array}$	This instruction shifts either A or M one bit right through C. C is stored in bit 7 and bit 0 is stored in C.							6A	2	1				66	5	2					
RRF	$\begin{array}{ c } \hline 7 \quad 0 \\ \hline \end{array} \rightarrow$	This instruction rotates 4 bits of the M content to the right.													82	8	2					
RTI	$S \leftarrow S + 1$ $PS \leftarrow M(S)$ $S \leftarrow S + 1$ $PCL \leftarrow M(S)$ $S \leftarrow S + 1$ $PCH \leftarrow M(S)$	This instruction increments S by one, and stores the contents of the memory location designated by S in PS. S is again incremented by one and stores the contents of the memory location designated by S in PCL. S is again incremented by one and stores the contents of memory location designated by S in PCH.	40	6	1																	
RTS	$S \leftarrow S + 1$ $PCL \leftarrow M(S)$ $S \leftarrow S + 1$ $PCH \leftarrow M(S)$ $(PC) \leftarrow (PC) + 1$	This instruction increments S by one and stores the contents of the memory location designated by S in PCL. S is again incremented by one and the contents of the memory location is stored in PCH. PC is incremented by 1.	60	6	1																	

APPENDIX

3.10 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	#			
SBC (Note 1) (Note 5)	When T = 0 $A \leftarrow A - M - \bar{C}$ When T = 1 $M(X) \leftarrow M(X) - M - \bar{C}$	When T = 0, this instruction subtracts the value of M and the complement of C from A, and stores the results in A and C. When T = 1, the instruction subtracts the contents of M and the complement of C from the contents of M(X), and stores the results in M(X) and C. A remain unchanged, but status flag are changed. M(X) represents the contents of memory where is indicated by X.				E9	2	2										E5	3	2			
SEB	A_i or $M_i \leftarrow 1$	This instruction sets the designated bit i of A or M.										0B 20i	2	1				0F 20i	5	2			
SEC	$C \leftarrow 1$	This instruction sets C.	38	2	1																		
SED	$D \leftarrow 1$	This instruction set D.	F8	2	1																		
SEI	$I \leftarrow 1$	This instruction set I.	78	2	1																		
SET	$T \leftarrow 1$	This instruction set T.	32	2	1																		
STA	$M \leftarrow A$	This instruction stores the contents of A in M. The contents of A does not change.																85	4	2			
STP		This instruction resets the oscillation control F/F and the oscillation stops. Reset or interrupt input is needed to wake up from this mode.	42	2	1																		
STX	$M \leftarrow X$	This instruction stores the contents of X in M. The contents of X does not change.																86	4	2			
STY	$M \leftarrow Y$	This instruction stores the contents of Y in M. The contents of Y does not change.																84	4	2			
TAX	$X \leftarrow A$	This instruction stores the contents of A in X. The contents of A does not change.	AA	2	1																		
TAY	$Y \leftarrow A$	This instruction stores the contents of A in Y. The contents of A does not change.	A8	2	1																		
TST	$M = 0?$	This instruction tests whether the contents of M are "0" or not and modifies the N and Z.																64	3	2			
TSX	$X \leftarrow S$	This instruction transfers the contents of S in X.	BA	2	1																		
TXA	$A \leftarrow X$	This instruction stores the contents of X in A.	8A	2	1																		
TXS	$S \leftarrow X$	This instruction stores the contents of X in S.	9A	2	1																		
TYA	$A \leftarrow Y$	This instruction stores the contents of Y in A.	98	2	1																		
WIT		The WIT instruction stops the internal clock but not the oscillation of the oscillation circuit is not stopped. CPU starts its function after the Timer X over flows (comes to the terminal count). All registers or internal memory contents except Timer X will not change during this mode. (Of course needs VDD).	C2	2	1																		

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

APPENDIX

3.10 Machine instructions

Symbol	Contents	Symbol	Contents
IMP	Implied addressing mode	+	Addition
IMM	Immediate addressing mode	-	Subtraction
A	Accumulator or Accumulator addressing mode	*	Multiplication
BIT, A	Accumulator bit addressing mode	/	Division
BIT, A, R	Accumulator bit relative addressing mode	∧	Logical OR
ZP	Zero page addressing mode	∨	Logical AND
BIT, ZP	Zero page bit addressing mode	⊕	Logical exclusive OR
BIT, ZP, R	Zero page bit relative addressing mode	—	Negation
ZP, X	Zero page X addressing mode	←	Shows direction of data flow
ZP, Y	Zero page Y addressing mode	X	Index register X
ABS	Absolute addressing mode	Y	Index register Y
ABS, X	Absolute X addressing mode	S	Stack pointer
ABS, Y	Absolute Y addressing mode	PC	Program counter
IND	Indirect absolute addressing mode	PS	Processor status register
ZP, IND	Zero page indirect absolute addressing mode	PCH	8 high-order bits of program counter
IND, X	Indirect X addressing mode	PCL	8 low-order bits of program counter
IND, Y	Indirect Y addressing mode	ADH	8 high-order bits of address
REL	Relative addressing mode	ADL	8 low-order bits of address
SP	Special page addressing mode	FF	FF in Hexadecimal notation
C	Carry flag	nn	Immediate value
Z	Zero flag	zz	Zero page address
I	Interrupt disable flag	M	Memory specified by address designation of any addressing mode
D	Decimal mode flag	M(X)	Memory of address indicated by contents of index register X
B	Break flag	M(S)	Memory of address indicated by contents of stack pointer
T	X-modified arithmetic mode flag	M(ADH, ADL)	Contents of memory at address indicated by ADH and ADL, in ADH is 8 high-order bits and ADL is 8 low-order bits.
V	Overflow flag	M(00, ADL)	Contents of address indicated by zero page ADL
N	Negative flag	Ai	Bit i (i = 0 to 7) of accumulator
		Mi	Bit i (i = 0 to 7) of memory
		OP	Opcode
		n	Number of cycles
		#	Number of bytes

3.11 List of instruction code

D7 – D4	D3 – D0	Hexadecimal notation															
		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

APPENDIX

3.12 SFR memory map

3.12 SFR memory map

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	Timer count source selection register (TCSS)
0009 ₁₆	Port P4 direction register (P4D)	0029 ₁₆	
000A ₁₆		002A ₁₆	
000B ₁₆		002B ₁₆	I ² C data shift register (S0)
000C ₁₆		002C ₁₆	I ² C address register (S0D)
000D ₁₆		002D ₁₆	I ² C status register (S1)
000E ₁₆		002E ₁₆	I ² C control register (S1D)
000F ₁₆		002F ₁₆	I ² C clock control register (S2)
0010 ₁₆		0030 ₁₆	I ² C start/stop condition control register (S2D)
0011 ₁₆		0031 ₁₆	Reserved *
0012 ₁₆		0032 ₁₆	
0013 ₁₆		0033 ₁₆	
0014 ₁₆		0034 ₁₆	A-D control register (ADCON)
0015 ₁₆	Reserved *	0035 ₁₆	A-D conversion low-order register (ADL)
0016 ₁₆	Reserved *	0036 ₁₆	A-D conversion high-order register (ADH)
0017 ₁₆	Reserved *	0037 ₁₆	
0018 ₁₆	Transmit/Receive buffer register (TB/RB)	0038 ₁₆	MISRG
0019 ₁₆	Serial I/O status register (SIOSTS)	0039 ₁₆	Watchdog timer control register (WDTCN)
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	PWM control register (PWMCN)	003D ₁₆	Interrupt request register 2 (IREQ2)
001E ₁₆	PWM prescaler (PREPWM)	003E ₁₆	Interrupt control register 1 (ICON1)
001F ₁₆	PWM register (PWM)	003F ₁₆	Interrupt control register 2 (ICON2)

* Reserved : Do not write "1" to this address.

3.13 Pin configurations

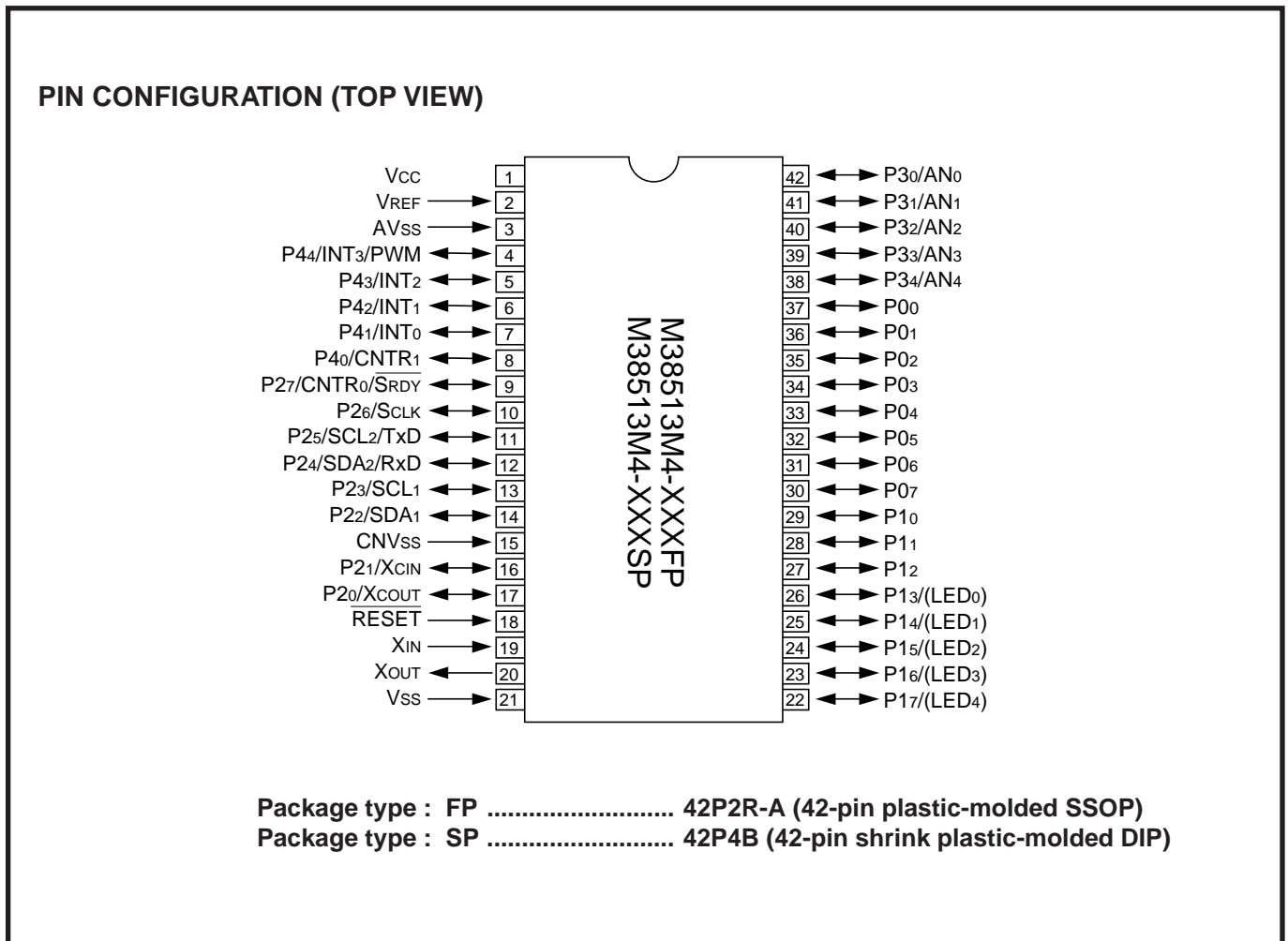


Fig. 3.13.1 M38513M4-XXXXFP/SP pin configuration

APPENDIX

3.13 Pin configurations

MEMORANDUM

**MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
3850/3851 Group**

Oct. First Edition 1998

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

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

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

©1998 MITSUBISHI ELECTRIC CORPORATION

**3850/3851 Group
User's Manual**



Renesas Electronics Corporation

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