

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

7532/7536  
Group

User's Manual

### Keep safety first in your circuit designs!

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

### Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by 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.

The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.

Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (<http://www.mitsubishichips.com>).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.

Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

## Preface

This user's manual describes Mitsubishi's CMOS 8-bit microcomputers 7532/7536 Group.

After reading this manual, the user should have a thorough knowledge of the functions and features of the 7532/7536 Group, and should be able to fully utilize the product. The manual starts with specifications and ends with application examples.

In this user's manual, 7536 Group is mainly described. For the 7532 Group, refer to "DIFFERENCES BETWEEN 7532 GROUP AND 7536 GROUP" in CHAPTER 1.

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

For details of development support tools, refer to the "Mitsubishi Microcomputer Development Support Tools" Homepage ([http://www.tool-spt.mesc.co.jp/index\\_e.htm](http://www.tool-spt.mesc.co.jp/index_e.htm)).

# 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. In this user's manual, 7536 Group is mainly described. For the 7532 Group, refer to "DIFFERENCES BETWEEN 7532 GROUP AND 7536 GROUP" in CHAPTER 1.

### ● CHAPTER 2 APPLICATION

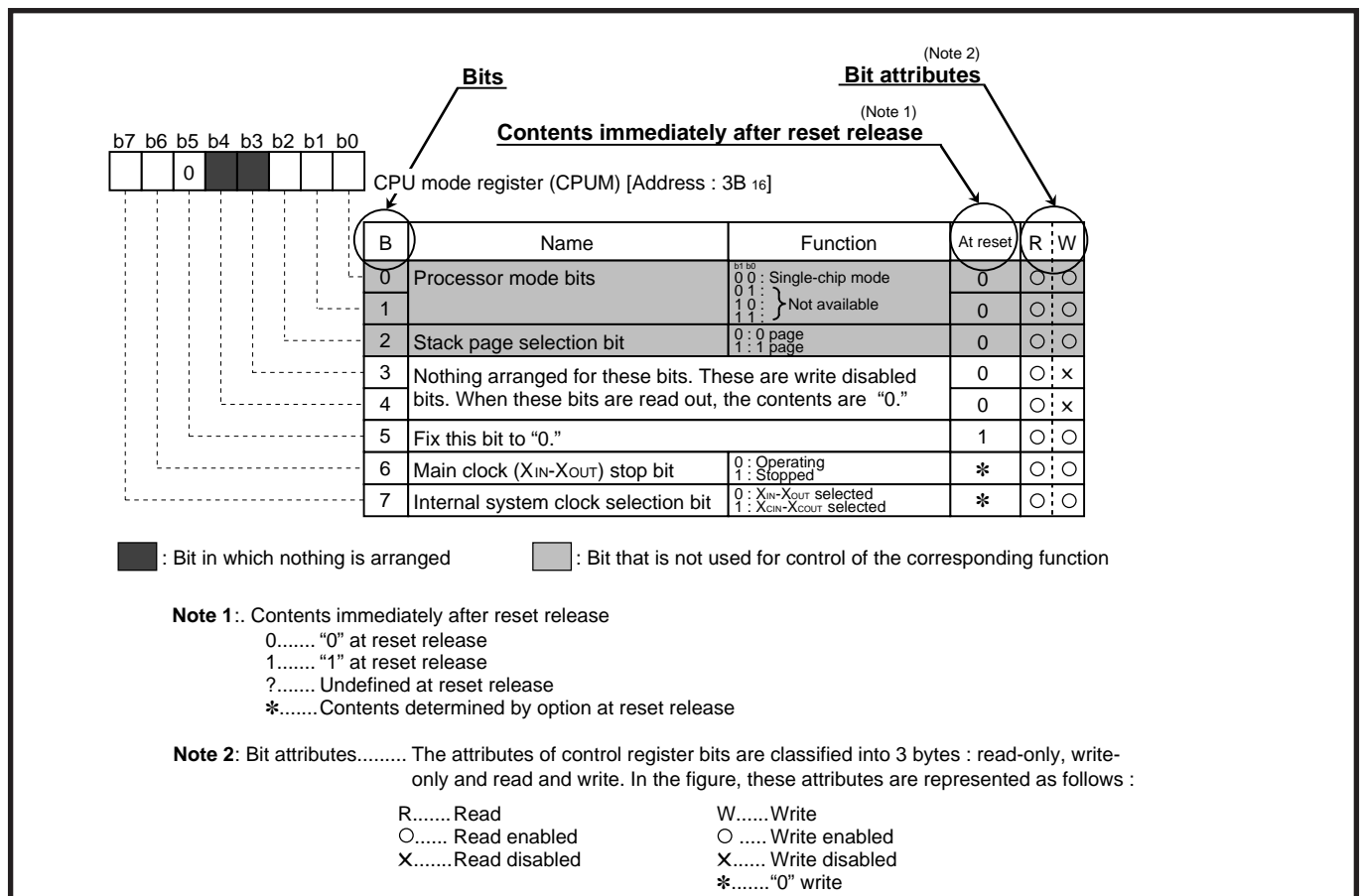
This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of relevant registers.

### ● 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

<b>DESCRIPTION .....</b>	<b>1-2</b>
<b>FEATURES .....</b>	<b>1-2</b>
<b>APPLICATION .....</b>	<b>1-2</b>
<b>PIN CONFIGURATION .....</b>	<b>1-2</b>
<b>FUNCTIONAL BLOCK .....</b>	<b>1-4</b>
<b>PIN DESCRIPTION .....</b>	<b>1-5</b>
<b>GROUP EXPANSION .....</b>	<b>1-6</b>
<b>FUNCTIONAL DESCRIPTION .....</b>	<b>1-7</b>
Central Processing Unit (CPU) .....	1-7
Processor status register (PS) .....	1-9
Switching method of CPU mode register .....	1-10
Memory .....	1-11
I/O Ports .....	1-13
Interrupts .....	1-17
Key Input Interrupt (Key-On Wake-Up) .....	1-19
Timers .....	1-20
Serial I/O .....	1-22
A-D Converter .....	1-33
Reset Circuit .....	1-35
Clock Generating Circuit .....	1-37
<b>NOTES ON PROGRAMMING .....</b>	<b>1-39</b>
<b>NOTES ON USE .....</b>	<b>1-39</b>
<b>DATA REQUIRED FOR MASK ORDERS .....</b>	<b>1-40</b>
<b>ROM PROGRAMMING METHOD .....</b>	<b>1-40</b>
<b>FUNCTIONAL DESCRIPTION SUPPLEMENT .....</b>	<b>1-41</b>
Interrupt .....	1-41
Timing After Interrupt .....	1-42
A-D Converter .....	1-43
Stop mode .....	1-45
Wait mode .....	1-46
<b>DIFFERENCES BETWEEN 7532 GROUP AND 7536 GROUP .....</b>	<b>1-47</b>
<b>Description supplement for use of USB function Stably .....</b>	<b>1-51</b>
Relationship between USB output signal (D+, D- signal) and cable load .....	1-54

## CHAPTER 2 APPLICATION

<b>2.1 I/O port .....</b>	<b>2-2</b>
2.1.1 Memory map .....	2-2
2.1.2 Relevant registers .....	2-2
2.1.3 Application example of key-on wake up .....	2-6
2.1.4 Handling of unused pins .....	2-7
2.1.5 Notes on input and output pins .....	2-8
2.1.6 Termination of unused pins .....	2-9
<b>2.2 Timer .....</b>	<b>2-10</b>
2.2.1 Memory map .....	2-10
2.2.2 Relevant registers .....	2-10
2.2.3 Timer application examples .....	2-16

## Table of contents

---

<b>2.3 Serial I/O .....</b>	<b>2-29</b>
2.3.1 Memory map .....	2-29
2.3.2 Relevant registers .....	2-29
2.3.3 Serial I/O connection examples .....	2-35
2.3.4 Serial I/O transfer data format .....	2-37
2.3.5 Serial I/O application examples .....	2-38
2.3.6 Notes on serial I/O .....	2-49
<b>2.4 USB .....</b>	<b>2-50</b>
2.4.1 Outline of USB .....	2-50
2.4.2 Memory map .....	2-56
2.4.3 Relevant registers .....	2-57
2.4.4 USB application example .....	2-62
2.4.5 Notes concerning USB .....	2-68
<b>2.5 A-D converter .....</b>	<b>2-69</b>
2.5.1 Memory map .....	2-69
2.5.2 Relevant registers .....	2-69
2.5.3 A-D converter application examples .....	2-73
2.5.4 Notes on A-D converter .....	2-75
<b>2.6 Reset.....</b>	<b>2-76</b>
2.6.1 Connection example of reset IC .....	2-76
2.6.2 Notes on RESET pin .....	2-77

## CHAPTER 3 APPENDIX

---

<b>3.1 Electrical characteristics .....</b>	<b>3-2</b>
3.1.1 Absolute Maximum Ratings .....	3-2
3.1.2 Recommended Operating Conditions .....	3-3
3.1.3 Electrical Characteristics .....	3-4
3.1.4 A-D Converter Characteristics .....	3-5
3.1.5 Timing Requirements .....	3-6
3.1.6 Switching Characteristics .....	3-6
<b>3.2 Typical characteristics .....</b>	<b>3-8</b>
3.2.1 Power source current characteristic example (I <sub>CC</sub> -V <sub>CC</sub> characteristic) .....	3-8
3.2.2 V <sub>OH</sub> -I <sub>OH</sub> characteristic example.....	3-11
3.2.3 A-D conversion typical characteristics example.....	3-15
<b>3.3 Notes on use .....</b>	<b>3-17</b>
3.3.1 Notes on interrupts .....	3-17
3.3.2 Notes on serial I/O .....	3-18
3.3.3 Notes on A-D converter .....	3-19
3.3.4 Notes on RESET pin .....	3-19
3.3.5 Notes on input and output pins .....	3-20
3.3.6 Notes on programming .....	3-21
3.3.7 Programming and test of built-in PROM version.....	3-23
3.3.8 Notes on built-in PROM version .....	3-23
3.3.9 Termination of unused pins .....	3-25
3.3.10 Notes on CPU mode register.....	3-26
3.3.11 Notes on using 32-pin version .....	3-26

<b>3.4 Countermeasures against noise .....</b>	<b>3-27</b>
3.4.1 Shortest wiring length .....	3-27
3.4.2 Connection of bypass capacitor across Vss line and Vcc line .....	3-29
3.4.4 Oscillator concerns.....	3-30
3.4.3 Wiring to analog input pins .....	3-30
3.4.5 Setup for I/O ports.....	3-32
3.4.6 Providing of watchdog timer function by software .....	3-33
<b>3.5 List of registers .....</b>	<b>3-34</b>
<b>3.6 Mask ROM confirmation form.....</b>	<b>3-53</b>
<b>3.7 Mark specification form .....</b>	<b>3-59</b>
<b>3.8 Package outline .....</b>	<b>3-62</b>
<b>3.9 Machine instructions .....</b>	<b>3-64</b>
<b>3.10 List of instruction code .....</b>	<b>3-75</b>
<b>3.11 SFR memory map.....</b>	<b>3-76</b>
<b>3.12 Pin configurations .....</b>	<b>3-77</b>

# List of figures

## CHAPTER 1 HARDWARE

Fig. 1 Pin configuration of M37536M4-XXXSP,M37536E8SP .....	1-2
Fig. 2 Pin configuration of M37536RSS .....	1-3
Fig. 3 Functional block diagram.....	1-4
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 Switching method of CPU mode register.....	1-10
Fig. 9 Memory map diagram .....	1-11
Fig. 10 Memory map of special function register (SFR) .....	1-12
Fig. 11 Structure of pull-up control register .....	1-13
Fig. 12 Structure of port P1P3 control register .....	1-13
Fig. 13 Block diagram of ports (1) .....	1-15
Fig. 14 Block diagram of ports (2) .....	1-16
Fig. 15 Interrupt control.....	1-18
Fig. 16 Structure of interrupt-related registers.....	1-18
Fig. 17 Connection example when using key input interrupt and port P0 block diagram .....	1-19
Fig. 18 Structure of timer X mode register .....	1-20
Fig. 19 Timer count source setting register .....	1-20
Fig. 20 Block diagram of timer X, timer 1 and timer 2.....	1-21
Fig. 21 Block diagram of UART serial I/O .....	1-22
Fig. 22 Operation of UART serial I/O function .....	1-22
Fig. 23 Continuous transmission operation of UART serial I/O .....	1-23
Fig. 24 USB mode block diagram .....	1-24
Fig. 25 USB transceiver block diagram .....	1-24
Fig. 26 Structure of serial I/O1-related registers (1) .....	1-25
Fig. 27 Structure of serial I/O1-related registers (2) .....	1-26
Fig. 28 Structure of serial I/O1-related registers (3) .....	1-27
Fig. 29 Structure of serial I/O1-related registers (4) .....	1-28
Fig. 30 Structure of serial I/O1-related registers (5) .....	1-29
Fig. 31 Structure of serial I/O2 control registers.....	1-31
Fig. 32 Block diagram of serial I/O2 .....	1-31
Fig. 33 Serial I/O2 timing (LSB first) .....	1-32
Fig. 34 Structure of A-D control register .....	1-33
Fig. 35 Structure of A-D conversion register .....	1-33
Fig. 36 Block diagram of A-D converter .....	1-33
Fig. 37 Block diagram of watchdog timer.....	1-34
Fig. 38 Structure of watchdog timer control register .....	1-34
Fig. 39 Example of reset circuit.....	1-35
Fig. 40 Timing diagram at reset .....	1-35
Fig. 41 Internal status of microcomputer at reset .....	1-36
Fig. 42 External circuit of ceramic resonator.....	1-37
Fig. 43 External clock input circuit .....	1-37
Fig. 44 Structure of MISRG .....	1-37
Fig. 45 Block diagram of system clock generating circuit (for ceramic resonator) .....	1-38



Fig. 46 Programming and testing of One Time PROM version .....	1-40
Fig. 47 Timing chart after an interrupt occurs .....	1-42
Fig. 48 Time up to execution of the interrupt processing routine .....	1-42
Fig. 49 A-D conversion equivalent circuit .....	1-44
Fig. 50 A-D conversion timing chart .....	1-44
Fig. 51 Pin configuration of M37532M4-XXXFP, M37532E8FP .....	1-49
Fig. 52 Pin configuration of M37532M4-XXXGP .....	1-49
Fig. 53 Pin configuration of M37532RSS .....	1-50
Fig. 54 Handling of $V_{CC}$ , $USBV_{REFOUT}$ pins of M37532M4-XXXFP, M37532E8FP .....	1-51
Fig. 55 Handling of $V_{CC}$ , $USBV_{REFOUT}$ pins of M37532M4-XXXGP .....	1-52
Fig. 56 Handling of $V_{CC}$ , $USBV_{REFOUT}$ pins of M37536E8SP, M37536M4-XXXSP, M37536RSS, M37532RSS .....	1-53

## 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 ( $i = 0$ to 4) .....	2-2
Fig. 2.1.3 Structure of Port Pi direction register ( $i = 0$ to 4) .....	2-3
Fig. 2.1.4 Structure of Pull-up control register .....	2-3
Fig. 2.1.5 Structure of P1P3 control register .....	2-4
Fig. 2.1.6 Structure of Interrupt edge selection register .....	2-4
Fig. 2.1.7 Structure of Interrupt request register 1 .....	2-5
Fig. 2.1.8 Structure of Interrupt control register 1 .....	2-5
Fig. 2.1.9 Relevant registers setting .....	2-6
Fig. 2.1.10 Application circuit example .....	2-6
Fig. 2.1.11 Control procedure .....	2-7
Fig. 2.2.1 Memory map of registers relevant to timers .....	2-10
Fig. 2.2.2 Structure of Prescaler 12, Prescaler X .....	2-10
Fig. 2.2.3 Structure of Timer 1 .....	2-11
Fig. 2.2.4 Structure of Timer 2 .....	2-11
Fig. 2.2.5 Structure of Timer X .....	2-12
Fig. 2.2.6 Structure of Timer X mode register .....	2-13
Fig. 2.2.7 Structure of Timer count source set register .....	2-14
Fig. 2.2.8 Structure of Interrupt edge selection register .....	2-14
Fig. 2.2.9 Structure of Interrupt request register 1 .....	2-15
Fig. 2.2.10 Structure of Interrupt control register 1 .....	2-15
Fig. 2.2.11 Timers connection and setting of division ratios .....	2-17
Fig. 2.2.12 Relevant registers setting .....	2-18
Fig. 2.2.13 Control procedure .....	2-19
Fig. 2.2.14 Peripheral circuit example .....	2-20
Fig. 2.2.15 Timers connection and setting of division ratios .....	2-20
Fig. 2.2.16 Relevant registers setting .....	2-21
Fig. 2.2.17 Control procedure .....	2-22
Fig. 2.2.18 Judgment method of valid/invalid of input pulses .....	2-23
Fig. 2.2.19 Relevant registers setting .....	2-24
Fig. 2.2.20 Control procedure .....	2-25
Fig. 2.2.21 Timers connection and setting of division ratios .....	2-26
Fig. 2.2.22 Relevant registers setting .....	2-27
Fig. 2.2.23 Control procedure .....	2-28
Fig. 2.3.1 Memory map of registers relevant to serial I/O .....	2-29
Fig. 2.3.2 Structure of Transmit/Receive buffer register .....	2-29
Fig. 2.3.3 Structure of UART status register .....	2-30
Fig. 2.3.4 Structure of Serial I/O1 control register .....	2-30
Fig. 2.3.5 Structure of UART control register .....	2-31

## List of figures

---

Fig. 2.3.6 Structure of Baud rate generator .....	2-31
Fig. 2.3.7 Structure of Serial I/O2 control register .....	2-32
Fig. 2.3.8 Structure of Serial I/O2 register .....	2-32
Fig. 2.3.9 Structure of Interrupt edge selection register .....	2-33
Fig. 2.3.10 Structure of Interrupt request register 1 .....	2-33
Fig. 2.3.11 Structure of Interrupt control register 1 .....	2-34
Fig. 2.3.12 Serial I/O connection examples (1) .....	2-35
Fig. 2.3.13 Serial I/O connection examples (2) .....	2-36
Fig. 2.3.14 Serial I/O transfer data format .....	2-37
Fig. 2.3.15 Connection diagram .....	2-38
Fig. 2.3.16 Timing chart .....	2-38
Fig. 2.3.17 Registers setting relevant to transmission side .....	2-39
Fig. 2.3.18 Transmission data setting of serial I/O2 .....	2-40
Fig. 2.3.19 Registers setting relevant to reception side .....	2-40
Fig. 2.3.20 Control procedure of transmission side .....	2-41
Fig. 2.3.21 Control procedure of reception side .....	2-42
Fig. 2.3.22 Connection diagram .....	2-43
Fig. 2.3.23 Timing chart .....	2-43
Fig. 2.3.24 Registers setting relevant to transmission side .....	2-45
Fig. 2.3.25 Registers setting relevant to reception side .....	2-46
Fig. 2.3.26 Control procedure of transmission side .....	2-47
Fig. 2.3.27 Control procedure of reception side .....	2-48
Fig. 2.3.28 Sequence of clearing serial I/O .....	2-49
Fig. 2.4.1 Communication sequence of USB .....	2-51
Fig. 2.4.2 Data structure of USB packet .....	2-52
Fig. 2.4.3 USB (L.S.) interface .....	2-55
Fig. 2.4.4 USB (L.S.) connection example .....	2-55
Fig. 2.4.5 Memory map of registers relevant to USB .....	2-56
Fig. 2.4.6 Description of the register structure .....	2-57
Fig. 2.4.7 Register structures relevant to USB (1) .....	2-58
Fig. 2.4.8 Register structures relevant to USB (2) .....	2-59
Fig. 2.4.9 Register structures relevant to USB (3) .....	2-60
Fig. 2.4.10 Register structures relevant to USB (4) .....	2-61
Fig. 2.4.11 Control method of control sequence .....	2-62
Fig. 2.4.12 Timing chart of the transaction according to each token .....	2-63
Fig. 2.4.13 USB interrupt processing example (OUT token) .....	2-65
Fig. 2.4.14 USB interrupt processing example (IN token) .....	2-66
Fig. 2.4.15 Timing chart of each signal .....	2-67
Fig. 2.4.16 Example for determination of resume interrupt .....	2-68
Fig. 2.4.17 Processing for width of SE0 signal .....	2-68
Fig. 2.5.1 Memory map of registers relevant to A-D converter .....	2-69
Fig. 2.5.2 Structure of A-D control register .....	2-69
Fig. 2.5.3 Structure of A-D conversion register (high-order) .....	2-70
Fig. 2.5.4 Structure of A-D conversion register (low-order) .....	2-70
Fig. 2.5.6 Structure of Interrupt request register 1 .....	2-71
Fig. 2.5.5 Structure of Interrupt edge selection register .....	2-71
Fig. 2.5.7 Structure of Interrupt control register 1 .....	2-72
Fig. 2.5.8 Connection diagram .....	2-73
Fig. 2.5.9 Relevant registers setting .....	2-73
Fig. 2.5.10 Control procedure for 8-bit read .....	2-74
Fig. 2.5.11 Control procedure for 10-bit read .....	2-74
Fig. 2.6.1 Example of poweron reset circuit .....	2-76
Fig. 2.6.2 RAM backup system .....	2-76

## CHAPTER 3 APPENDIX

Fig. 3.1.1 Power source current measurement circuit in USB mode at oscillation stop .....	3-5
Fig. 3.1.2 Switching characteristics measurement circuit .....	3-6
Fig. 3.1.3 Timing chart .....	3-7
Fig. 3.2.1 Icc-Vcc characteristic example (in double-speed mode) .....	3-8
Fig. 3.2.2 Icc-Vcc characteristic example (at WIT instruction execution) .....	3-8
Fig. 3.2.3 Icc-Vcc characteristic example (At STP instruction execution, Ta = 25 °C) .....	3-9
Fig. 3.2.4 Icc-Vcc characteristic example (At STP instruction execution, Ta = 85 °C) .....	3-9
Fig. 3.2.5 Icc-Vcc characteristic example (at USB suspend, Ta = 25 °C) .....	3-10
Fig. 3.2.6 Icc-Vcc characteristic example (A-D conversion executed/not executed, f(XIN) = 6MHz, in double-speed mode) ...	3-10
Fig. 3.2.7 VOH-IOH characteristic example of P-channel (Ta = 25 °C): normal port .....	3-11
Fig. 3.2.8 VOH-IOH characteristic example of P-channel (Ta = 85 °C): normal port .....	3-11
Fig. 3.2.9 VOL-IOL characteristic example of N-channel (Ta = 25 °C): Normal port .....	3-12
Fig. 3.2.10 VOL-IOL characteristic example of N-channel (Ta = 85 °C): Normal port .....	3-12
Fig. 3.2.11 VOL-IOL characteristic example of N-channel (Ta = 25 °C): LED drive port...	3-13
Fig. 3.2.12 VOL-IOL characteristic example N-channel (Ta = 85 °C): LED drive port .....	3-13
Fig. 3.2.13 IIL-VCC characteristic example when connecting pull-up transistor .....	3-14
Fig. 3.2.14 Definition of A-D conversion accuracy .....	3-15
Fig. 3.2.15 A-D conversion typical characteristic example .....	3-16
Fig. 3.3.1 Sequence of switch the detection edge .....	3-17
Fig. 3.3.2 Sequence of check of interrupt request bit .....	3-17
Fig. 3.3.3 Structure of interrupt control register 1 .....	3-18
Fig. 3.3.4 Sequence of clearing serial I/O .....	3-18
Fig. 3.3.5 Initialization of processor status register .....	3-21
Fig. 3.3.6 Sequence of PLP instruction execution .....	3-21
Fig. 3.3.7 Stack memory contents after PHP instruction execution .....	3-21
Fig. 3.3.8 Status flag at decimal calculations .....	3-22
Fig. 3.3.9 Programming and testing of One Time PROM version .....	3-23
Fig. 3.3.10 Switching method of CPU mode register .....	3-26
Fig. 3.4.1 Selection of packages .....	3-27
Fig. 3.4.2 Wiring for the $\overline{\text{RESET}}$ pin .....	3-27
Fig. 3.4.3 Wiring for clock I/O pins .....	3-28
Fig. 3.4.4 Wiring for CNVss pin .....	3-28
Fig. 3.4.5 Wiring for the VPP pin of the One Time PROM .....	3-29
Fig. 3.4.6 Bypass capacitor across the Vss line and the Vcc line .....	3-29
Fig. 3.4.7 Analog signal line and a resistor and a capacitor .....	3-30
Fig. 3.4.8 Wiring for a large current signal line .....	3-30
Fig. 3.4.9 Wiring of signal lines where potential levels change frequently .....	3-31
Fig. 3.4.10 Vss pattern on the underside of an oscillator .....	3-31
Fig. 3.4.11 Setup for I/O ports .....	3-32
Fig. 3.4.12 Watchdog timer by software .....	3-33
Fig. 3.5.1 Structure of Port Pi (i = 0 to 4) .....	3-34
Fig. 3.5.2 Structure of Port Pi direction register (i = 0 to 4) .....	3-34
Fig. 3.5.3 Structure of Pull-up control register .....	3-35
Fig. 3.5.4 Structure of Port P1P3 control register .....	3-35
Fig. 3.5.5 Structure of Transmit/Receive buffer register .....	3-36
Fig. 3.5.6 Structure of UART status register .....	3-36
Fig. 3.5.7 Structure of USB status register .....	3-37
Fig. 3.5.8 Structure of Serial I/O1 control register .....	3-38
Fig. 3.5.9 Structure of UART control register .....	3-38
Fig. 3.5.10 Structure of Baud rate generator .....	3-39

## List of figures

---

Fig. 3.5.11 Structure of USB data toggle synchronization register .....	3-39
Fig. 3.5.12 Structure of USB interrupt source discrimination register 1 .....	3-39
Fig. 3.5.13 Structure of USB interrupt source discrimination register 2 .....	3-40
Fig. 3.5.14 Structure of USB interrupt control register .....	3-40
Fig. 3.5.15 Structure of USB transmit data byte number set register 0 .....	3-41
Fig. 3.5.16 Structure of USB transmit data byte number set register 1 .....	3-41
Fig. 3.5.17 Structure of USB PID control register 0 .....	3-41
Fig. 3.5.18 Structure of USB PID control register 1 .....	3-42
Fig. 3.5.19 Structure of USB address register .....	3-42
Fig. 3.5.20 Structure of USB sequence bit initialization register .....	3-42
Fig. 3.5.21 Structure of USB control register .....	3-42
Fig. 3.5.22 Structure of Prescaler 12, Prescaler X .....	3-43
Fig. 3.5.23 Structure of Timer 1 .....	3-43
Fig. 3.5.24 Structure of Timer 2 .....	3-44
Fig. 3.5.25 Structure of Timer X mode register .....	3-45
Fig. 3.5.26 Structure of Timer X .....	3-46
Fig. 3.5.27 Structure of Timer count source set register .....	3-46
Fig. 3.5.28 Structure of Serial I/O2 control register .....	3-47
Fig. 3.5.29 Structure of Serial I/O2 register .....	3-47
Fig. 3.5.30 Structure of A-D control register .....	3-48
Fig. 3.5.31 Structure of A-D conversion register (high-order) .....	3-49
Fig. 3.5.32 Structure of A-D conversion register (low-order) .....	3-49
Fig. 3.5.33 Structure of MISRG .....	3-50
Fig. 3.5.34 Structure of Watchdog timer control register .....	3-50
Fig. 3.5.35 Structure of Interrupt edge selection register .....	3-51
Fig. 3.5.36 Structure of CPU mode register .....	3-51
Fig. 3.5.37 Structure of Interrupt request register 1 .....	3-52
Fig. 3.5.38 Structure of Interrupt control register 1 .....	3-52
Fig. 3.12.1 M37536M4-XXXSP, M37536E8SP pin configuration .....	3-77
Fig. 3.12.2 M37536RSS pin configuration .....	3-78
Fig. 3.12.3 M37532M4-XXXFP, M37532E8FP pin configuration .....	3-79
Fig. 3.12.4 M37532M4-XXXGP pin configuration .....	3-80
Fig. 3.12.5 M37532RSS pin configuration .....	3-81

# List of tables

## CHAPTER 1 HARDWARE

Table 1 Pin description .....	1-5
Table 2 List of supported products .....	1-6
Table 3 Push and pop instructions of accumulator or processor status register .....	1-8
Table 4 Set and clear instructions of each bit of processor status register .....	1-9
Table 5 I/O port function table .....	1-14
Table 6 Interrupt vector address and priority .....	1-17
Table 7 Relation of the width of SE0 and the state of the device .....	1-30
Table 8 Special programming adapter .....	1-40
Table 9 Interrupt sources, vector addresses and interrupt priority .....	1-41
Table 10 Change of A-D conversion register during A-D conversion .....	1-43
Table 11 Stop mode state .....	1-45
Table 12 Wait mode state .....	1-46
Table 13 Differences between 7532 Group and 7536 Group .....	1-47
Table 14 Differences between 7532 Group and 7536 Group (SFR) .....	1-48
Table 15 Precondition: VCC= 4.4 V to 5.25 V, Ta = 0 °C to 70 °C .....	1-54

## CHAPTER 2 APPLICATION

Table 2.1.1 Handling of unused pins .....	2-7
Table 2.2.1 CNTR0 active edge switch bit function .....	2-13
Table 2.3.1 Setting example of baud rate generator (BRG) and transfer bit rate values .....	2-44
Table 2.4.1 Transfer types of USB .....	2-50
Table 2.4.2 Packet types of USB .....	2-52
Table 2.4.3 Data structure of USB packet .....	2-53
Table 2.4.4 PID .....	2-53
Table 2.4.5 Special signal of USB .....	2-54

## CHAPTER 3 APPENDIX

Table 3.1.1 Absolute maximum ratings .....	3-2
Table 3.1.2 Recommended operating conditions .....	3-3
Table 3.1.3 Electrical characteristics (1) .....	3-4
Table 3.1.4 Electrical characteristics (2) .....	3-5
Table 3.1.5 A-D Converter characteristics (1) .....	3-5
Table 3.1.6 Timing requirements .....	3-6
Table 3.1.7 Switching characteristics .....	3-6
Table 3.3.1 Programming adapters .....	3-23
Table 3.3.2 PROM programmer address setting .....	3-24
Table 3.5.1 CNTR0 active edge switch bit function .....	3-45



# CHAPTER 1

## **HARDWARE**

DESCRIPTION  
FEATURES  
PIN CONFIGURATION  
APPLICATION  
FUNCTIONAL BLOCK  
PIN DESCRIPTION  
GROUP EXPANSION  
FUNCTIONAL DESCRIPTION  
NOTES ON PROGRAMMING  
NOTES ON USE  
DATA REQUIRED FOR MASK ORDERS  
ROM PROGRAMMING METHOD  
FUNCTIONAL DESCRIPTION SUPPLEMENT  
DIFFERENCES BETWEEN 7532  
GROUP AND 7536 GROUP  
DESCRIPTION SUPPLEMENT FOR  
USE OF USB FUNCTION STABLY

# HARDWARE

## DESCRIPTION/FEATURES/APPLICATION/PIN CONFIGURATION

### DESCRIPTION

The 7536 Group is the 8-bit microcomputer based on the 740 family core technology.

The 7536 Group has a USB, 8-bit timers, and an A-D converter, and is useful for an input device for personal computer peripherals.

### FEATURES

- Basic machine-language instructions ..... 69
- The minimum instruction execution time ..... 0.34  $\mu$ s  
(at 6 MHz oscillation frequency for the shortest instruction)
- Memory size
  - ROM ..... 8K to 16K bytes
  - RAM ..... 256 to 384 bytes
- Programmable I/O ports ..... 33
- Interrupts ..... 14 sources, 8 vectors
- Timers ..... 8-bit X 3

- Serial I/O1 ..... used only for Low Speed in USB (USB/UART)
- Serial I/O2 ..... 8-bit X 1  
(Clock-synchronized)
- A-D converter ..... 10-bit X 8 channels
- Clock generating circuit ..... Built-in type  
(connect to external ceramic resonator or quartz-crystal oscillator )
- Watchdog timer ..... 16-bit X 1
- Power source voltage  
At 6 MHz XIN oscillation frequency at ceramic resonator  
..... 4.1 to 5.5 V
- Power dissipation ..... 30 mW (standard)
- Operating temperature range ..... -20 to 85  $^{\circ}$ C

### APPLICATION

Input device for personal computer peripherals

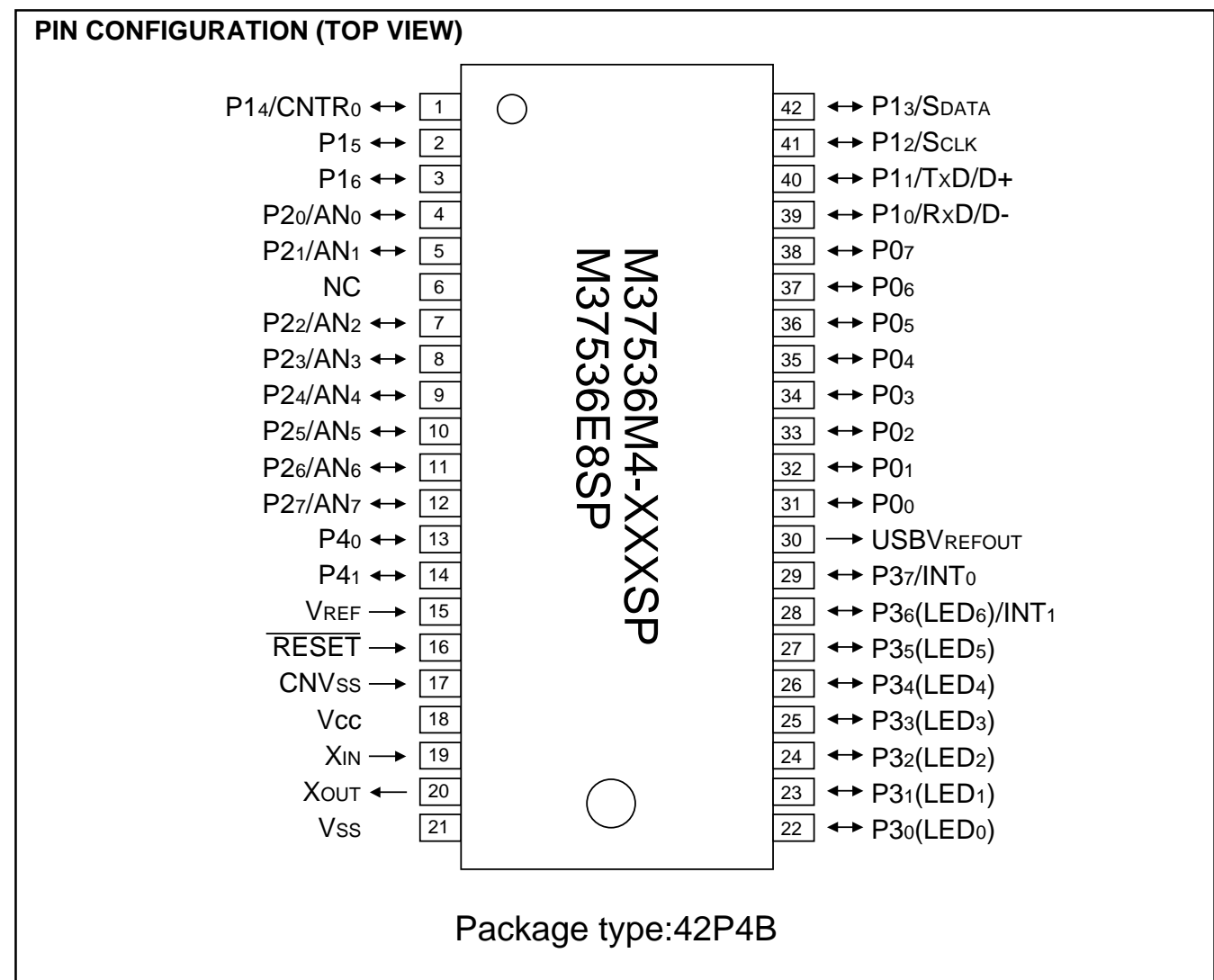


Fig. 1 Pin configuration of M37536M4-XXXSP, M37536E8SP

## PIN CONFIGURATION (TOP VIEW)

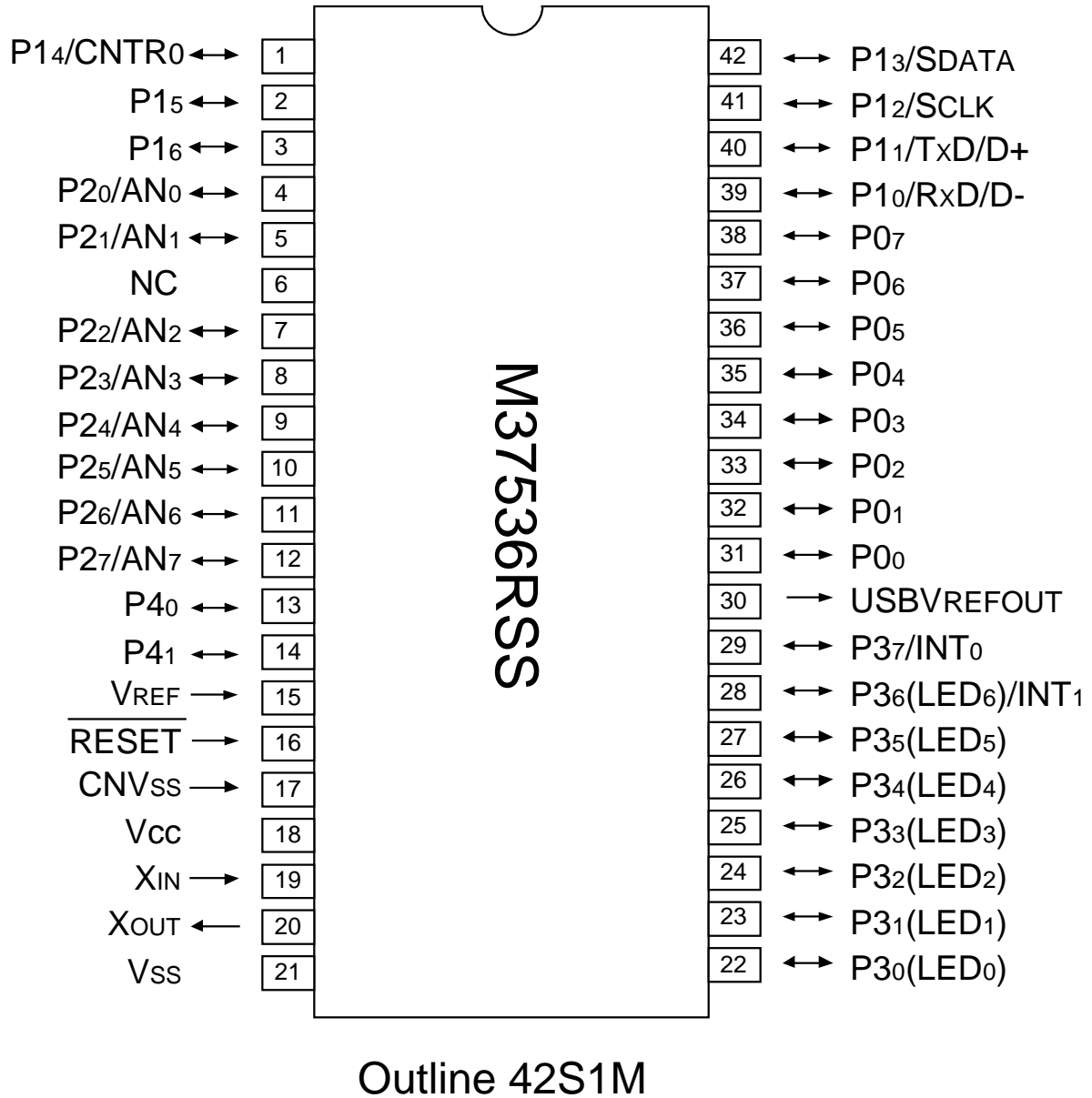


Fig. 2 Pin configuration of M37536RSS



FUNCTIONAL BLOCK

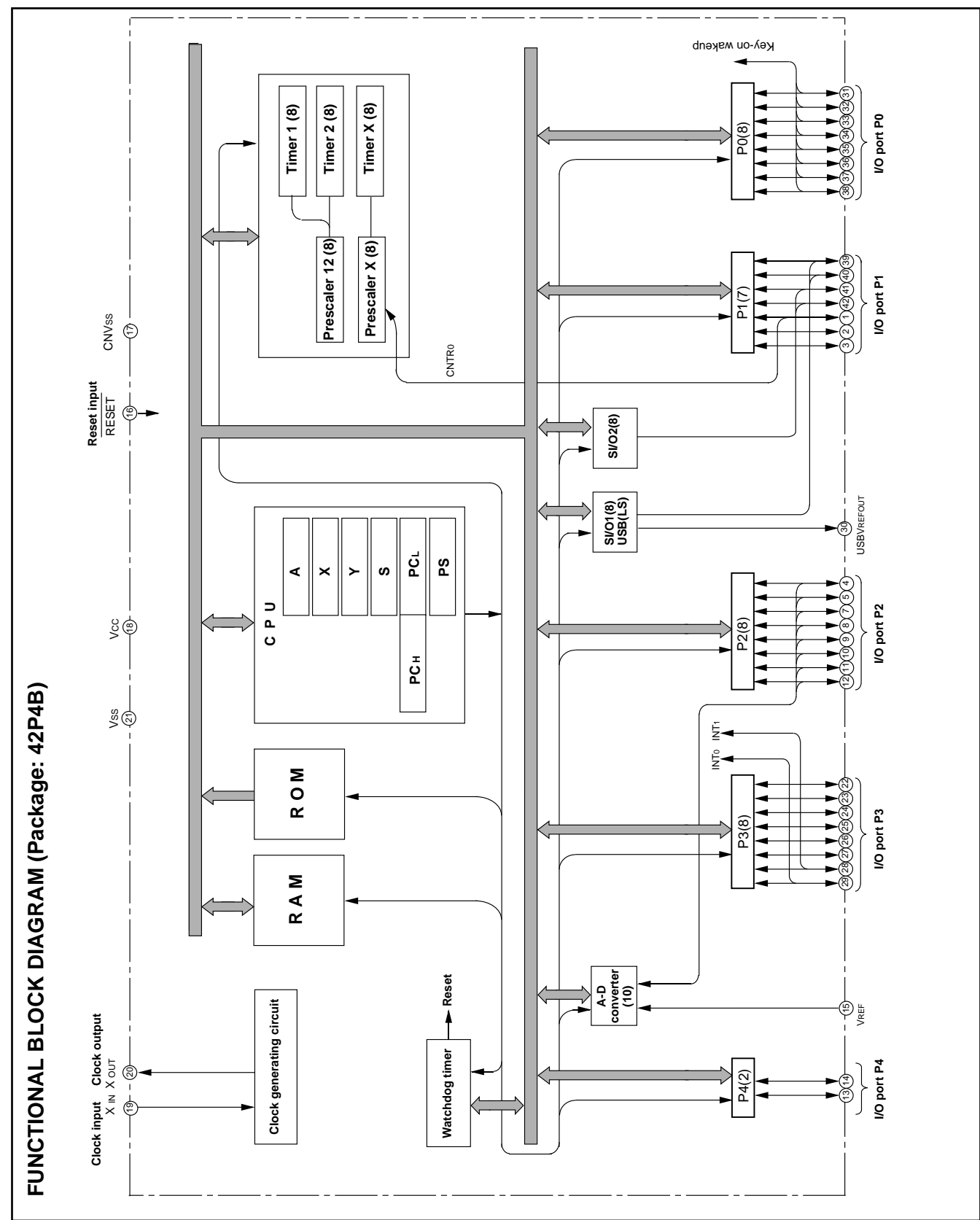


Fig. 3 Functional block diagram

## PIN DESCRIPTION

Table 1 Pin description

Pin	Name	Function	Function expect a port function
Vcc, Vss	Power source	•Apply voltage of 4.1 to 5.5 V to Vcc, and 0 V to Vss.	
VREF	Analog reference voltage	•Reference voltage input pin for A-D converter	
USBVREFOUT	USB reference voltage output	•Output pin for pulling up a D- line with 1.5 kΩ external resistor	
CNVss	CNVss	•Chip operating mode control pin, which is always connected to Vss.	
RESET	Reset input	•Reset input pin for active "L"	
XIN	Clock input	•Input and output pins for main clock generating circuit	
XOUT	Clock output	•Connect a ceramic resonator or quartz crystal oscillator between the XIN and XOUT pins. •If 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 I/O port. •I/O direction register allows each pin to be individually programmed as either input or output. •CMOS compatible input level •CMOS 3-state output structure •Whether a built-in pull-up resistor is to be used or not can be determined by program.	•Key-input (key-on wake up interrupt input) pins
P10/RxD/D- P11/TxD/D+	I/O port P1	•7-bit I/O port •I/O direction register allows each pin to be individually programmed as either input or output. •CMOS compatible input level •CMOS 3-state output structure •CMOS/TTL level can be switched for P10, P12, P13. •When using the USB function, input level of ports P10 and P11 becomes USB input level, and output level of them becomes USB output level.	•Serial I/O1 function pin
P12/SCLK P13/SDATA			•Serial I/O2 function pin
P14/CNTR0			•Timer X function pin
P15, P16			
P20/AN0– P27/AN7	I/O port P2	•8-bit I/O port having almost the same function as P0 •CMOS compatible input level •CMOS 3-state output structure	•Input pins for A-D converter
P30–P35	I/O port P3	•8-bit I/O port •I/O direction register allows each pin to be individually programmed as either input or output. •CMOS compatible input level (CMOS/TTL level can be switched for P36, P37). •CMOS 3-state output structure •P30 to P36 can output a large current for driving LED.	
P36/INT1 P37/INT0		•Whether a built-in pull-up resistor is to be used or not can be determined by program.	•Interrupt input pins
P40, P41	I/O port P4	•2-bit I/O port •I/O direction register allows each pin to be individually programmed as either input or output.	

# HARDWARE

## GROUP EXPANSION

### GROUP EXPANSION

Mitsubishi plans to expand the 7536 group as follow:

#### Memory type

Support for Mask ROM version, One Time PROM version, and Emulator MCU .

#### Memory size

ROM/PROM size ..... 8 K to 16 K bytes  
RAM size ..... 256 to 384 bytes

#### Package

42P4B ..... 42-pin plastic molded SDIP  
42SIM ..... 42-pin shrink ceramic PIGGY BACK

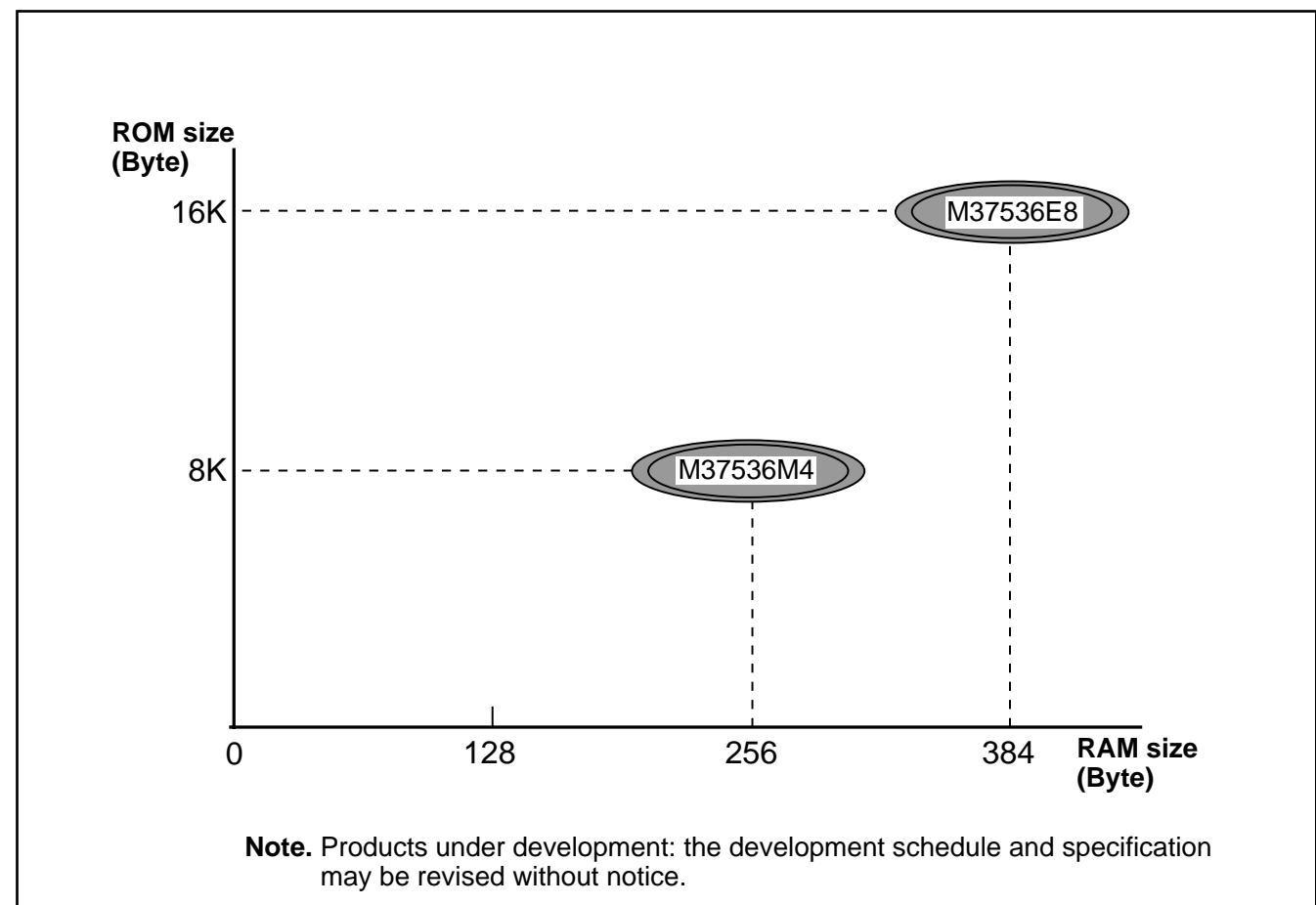


Fig. 4 Memory expansion plan

Currently supported products are listed below.

Table 2 List of supported products

Product	(P) ROM size (bytes) ROM size for User ()	RAM size (bytes)	Package	Remarks
M37536M4-XXXSP	8192 (8062)	256	42P4B	Mask ROM version
M37536E8SP	16384 (16254)	384	42P4B	One Time PROM version (blank)
M37536RSS	—	384	42S1M	Emulator MCU

FUNCTIONAL DESCRIPTION  
Central Processing Unit (CPU)

The 7536 group uses the standard 740 family instruction set. Refer to the table of 740 family addressing modes and machine instructions or the 740 Family 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 MUL and DIV instructions cannot be used.

The WIT and STP 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.

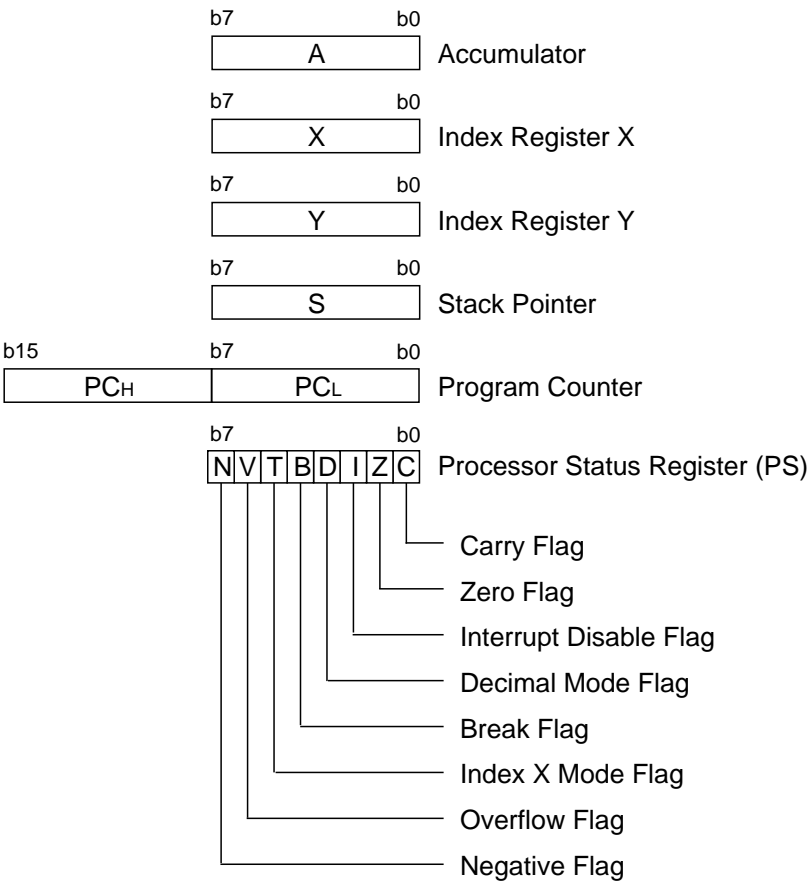


Fig. 5 740 Family CPU register structure

# HARDWARE

## FUNCTIONAL DESCRIPTION

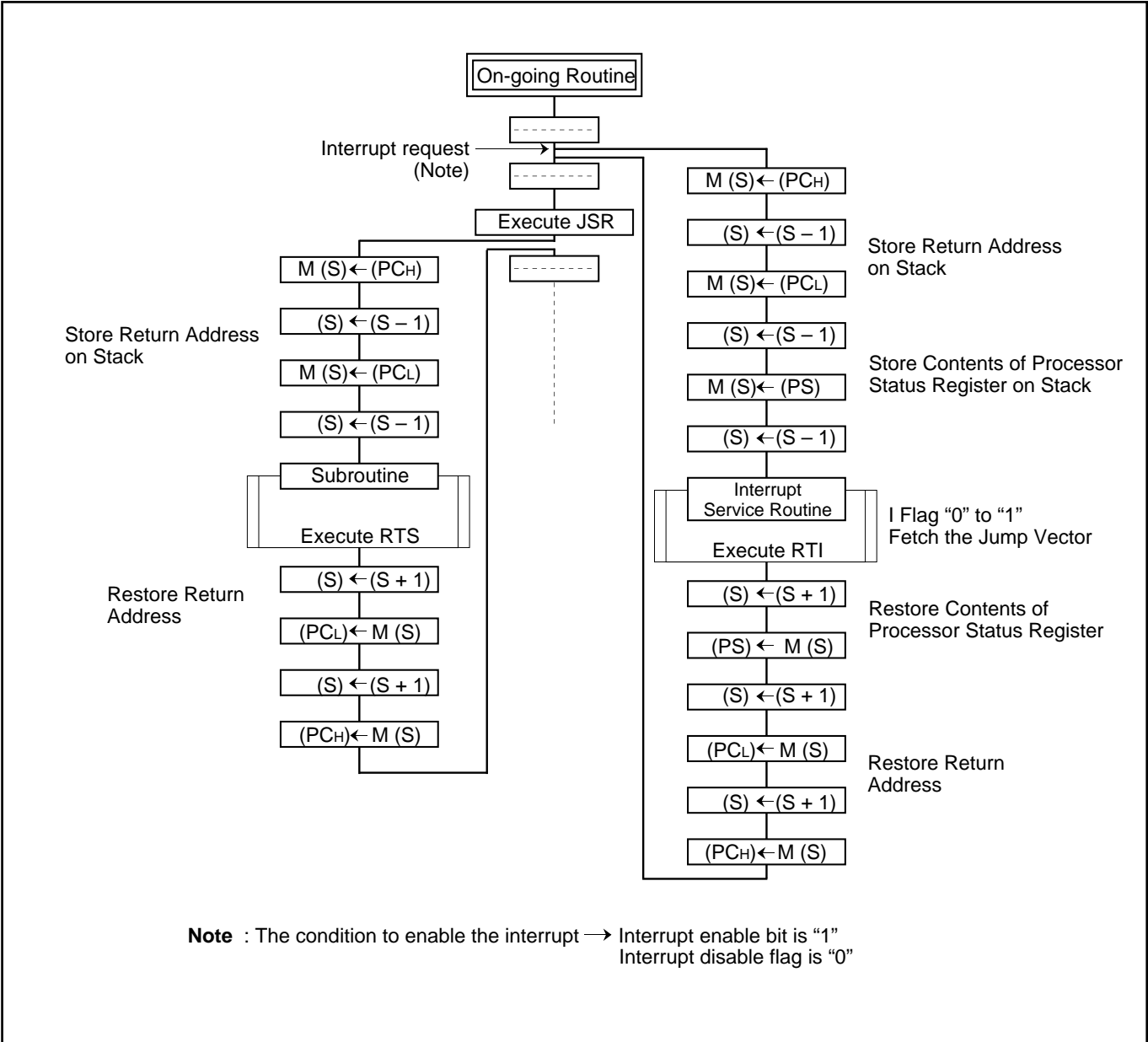


Fig. 6 Register push and pop at interrupt generation and subroutine call

Table 3 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 4 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

The CPU mode register contains the stack page selection bit.  
This register is allocated at address 003B<sub>16</sub>.

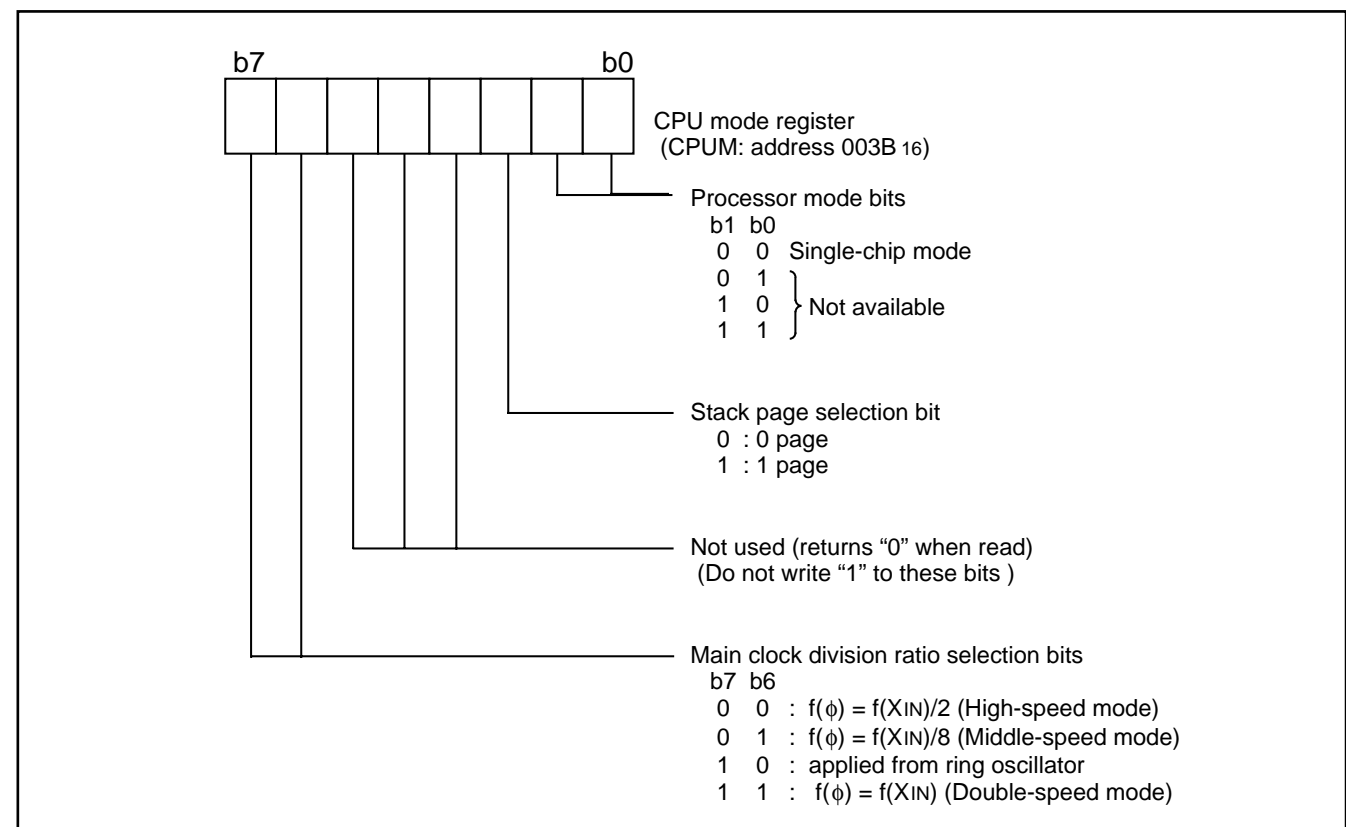


Fig. 7 Structure of CPU mode register

### Switching method of CPU mode register

Switch the CPU mode register (CPUM) at the head of program after releasing Reset in the following method.

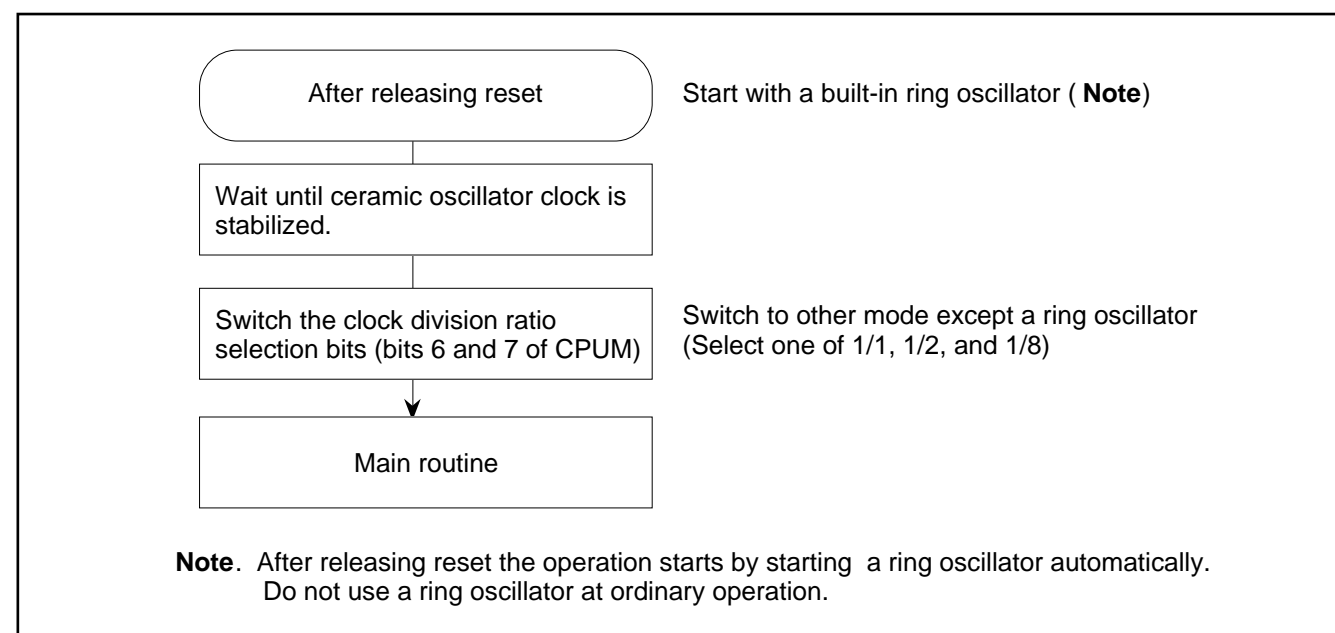


Fig. 8 Switching method of CPU mode register

### Memory

#### Special function register (SFR) area

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

#### RAM

RAM is used for data storage and for a 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 a user area for storing programs.

#### Interrupt vector area

The interrupt vector area contains reset and interrupt vectors.

#### Zero page

The 256 bytes from addresses  $0000_{16}$  to  $00FF_{16}$  are called the zero page area. The internal RAM and the special function registers (SFR) are allocated to this area.

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

#### Special page

The 256 bytes from addresses  $FF00_{16}$  to  $FFFF_{16}$  are called the special page area. The special page addressing mode can be used to specify memory addresses in the special page area. Access to this area with only 2 bytes is possible in the special page addressing mode.

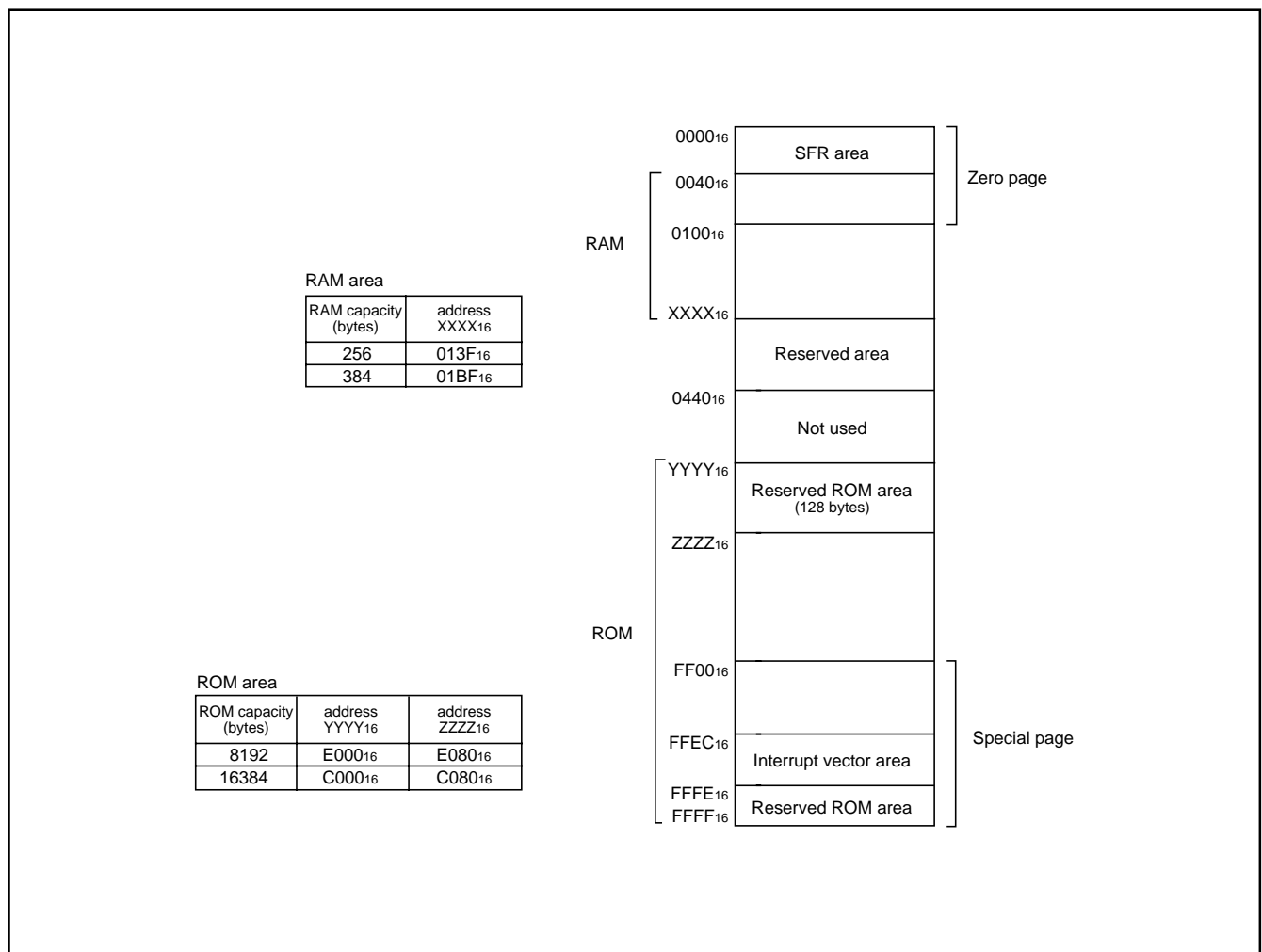


Fig. 9 Memory map diagram



# HARDWARE

## FUNCTIONAL DESCRIPTION

0000 <sub>16</sub>	Port P0 (P0)	0020 <sub>16</sub>	USB interrupt control register (USBICON)
0001 <sub>16</sub>	Port P0 direction register (P0D)	0021 <sub>16</sub>	USB transmit data byte number set register 0 (EP0BYTE)
0002 <sub>16</sub>	Port P1 (P1)	0022 <sub>16</sub>	USB transmit data byte number set register 1 (EP1BYTE)
0003 <sub>16</sub>	Port P1 direction register (P1D)	0023 <sub>16</sub>	USBPID control register 0 (EP0PID)
0004 <sub>16</sub>	Port P2 (P2)	0024 <sub>16</sub>	USBPID control register 1 (EP1PID)
0005 <sub>16</sub>	Port P2 direction register (P2D)	0025 <sub>16</sub>	USB address register (USBA)
0006 <sub>16</sub>	Port P3 (P3)	0026 <sub>16</sub>	USB sequence bit initialization register (INISQ1)
0007 <sub>16</sub>	Port P3 direction register (P3D)	0027 <sub>16</sub>	USB control register (USBCON)
0008 <sub>16</sub>	Port P4 (P4)	0028 <sub>16</sub>	Prescaler 12 (PRE12)
0009 <sub>16</sub>	Port P4 direction register (P4D)	0029 <sub>16</sub>	Timer 1 (T1)
000A <sub>16</sub>		002A <sub>16</sub>	Timer 2 (T2)
000B <sub>16</sub>		002B <sub>16</sub>	Timer X mode register (TM)
000C <sub>16</sub>		002C <sub>16</sub>	Prescaler X (PREX)
000D <sub>16</sub>		002D <sub>16</sub>	Timer X (TX)
000E <sub>16</sub>		002E <sub>16</sub>	Timer count source set register (TCSS)
000F <sub>16</sub>		002F <sub>16</sub>	
0010 <sub>16</sub>		0030 <sub>16</sub>	Serial I/O2 control register (SIO2CON)
0011 <sub>16</sub>		0031 <sub>16</sub>	Serial I/O2 register (SIO2)
0012 <sub>16</sub>		0032 <sub>16</sub>	
0013 <sub>16</sub>		0033 <sub>16</sub>	
0014 <sub>16</sub>		0034 <sub>16</sub>	A-D control register (ADCON)
0015 <sub>16</sub>		0035 <sub>16</sub>	A-D conversion register (low-order) (ADL)
0016 <sub>16</sub>	Pull-up control register (PULL)	0036 <sub>16</sub>	A-D conversion register (high-order) (ADH)
0017 <sub>16</sub>	Port P1P3 control register (P1P3C)	0037 <sub>16</sub>	
0018 <sub>16</sub>	Transmit/Receive buffer register (TB/RB)	0038 <sub>16</sub>	MISRG
0019 <sub>16</sub>	USB status register (USBSTS)/UART status register (UARTSTS)	0039 <sub>16</sub>	Watchdog timer control register (WDTCON)
001A <sub>16</sub>	Serial I/O1 control register (SIO1CON)	003A <sub>16</sub>	Interrupt edge selection register (INTEDGE)
001B <sub>16</sub>	UART control register (UARTCON)	003B <sub>16</sub>	CPU mode register (CPUM)
001C <sub>16</sub>	Baud rate generator (BRG)	003C <sub>16</sub>	Interrupt request register 1 (IREQ1)
001D <sub>16</sub>	USB data toggle synchronization register ( TRSYNC)	003D <sub>16</sub>	
001E <sub>16</sub>	USB interrupt source discrimination register 1 (USBIR1)	003E <sub>16</sub>	Interrupt control register 1 (ICON1)
001F <sub>16</sub>	USB interrupt source discrimination register 2 (USBIR2)	003F <sub>16</sub>	

Fig. 10 Memory map of special function register (SFR)

### I/O Ports

#### [Direction registers] PiD

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

When "1" is set to the bit corresponding to a pin, this pin becomes an output port. When "0" is set to the bit, the pin becomes an input port. When data is read from a pin set to output, not the value of the pin itself but the value of port latch is read. Pins set to input are floating, and permit reading pin values.

If a pin set to input is written to, only the port latch is written to and the pin remains floating.

#### [Pull-up control] PULL

By setting the pull-up control register (address 0016<sub>16</sub>), ports P0 and P3 can exert pull-up control by program. However, pins set to output are disconnected from this control and cannot exert pull-up control.

#### [Port P1P3 control] P1P3C

By setting the port P1P3 control register (address 0017<sub>16</sub>), a CMOS input level or a TTL input level can be selected for ports P10, P12, P13, P35, P36 and P37 by program.

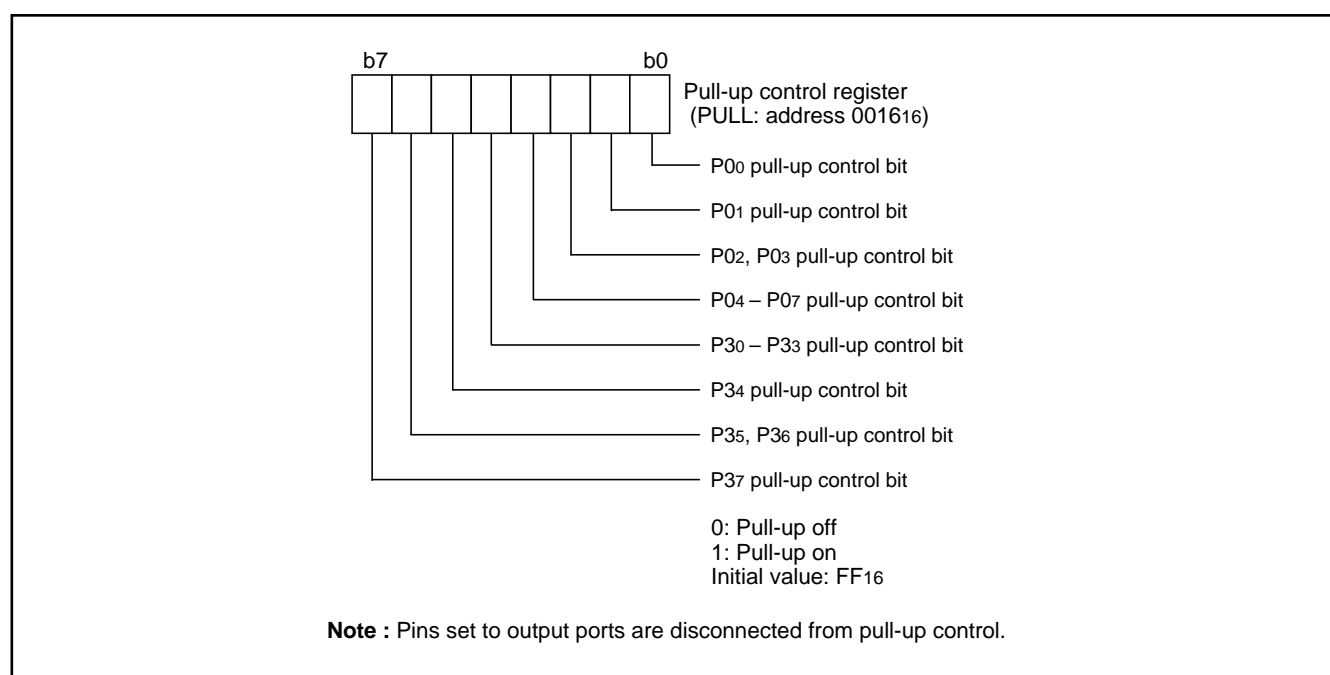


Fig. 11 Structure of pull-up control register

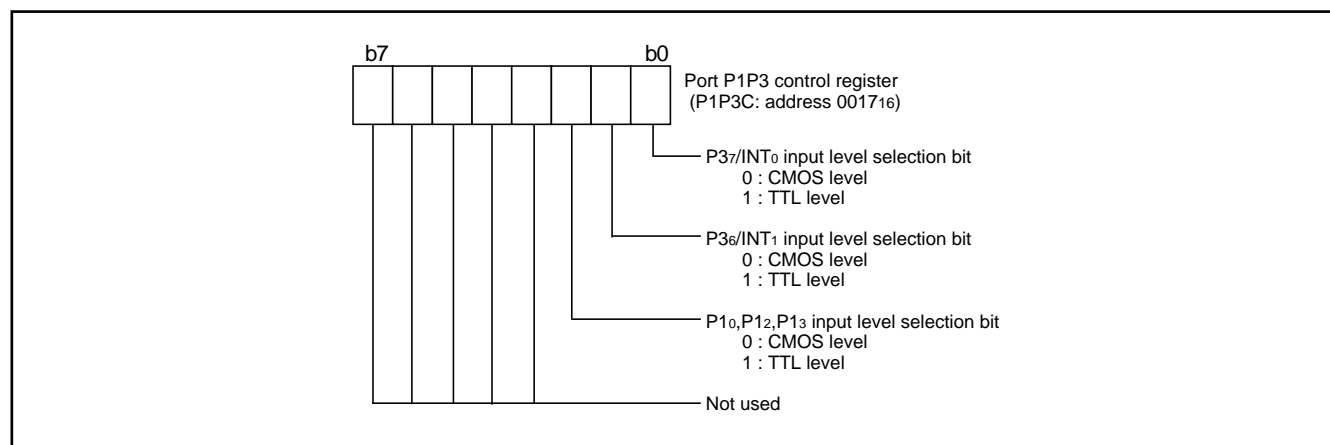


Fig. 12 Structure of port P1P3 control register

# HARDWARE

## FUNCTIONAL DESCRIPTION

Table 5 I/O port function table

Pin	Name	Input/output	I/O format	Non-port function	Related SFRs	Diagram No.
P00–P07	I/O port P0	I/O individual bits	•CMOS compatible input level •CMOS 3-state output	Key input interrupt	Pull-up control register	(1)
P10/RxD/D- P11/TxD/D+	I/O port P1		•USB input/output level when selecting USB function  •CMOS compatible input level •CMOS 3-state output (Note)	Serial I/O1 function input/output	Serial I/O1 control register	(2) (3)
P12/SCLK P13/SDATA				Serial I/O2 function input/output	Serial I/O2 control register	(4) (5)
P14/CNTR0						Timer X function input/output
P15, P16						(10)
P20/AN0– P27/AN7				I/O port P2	A-D conversion input	A-D control register
P30–P35	I/O port P3				(8)	
P36/INT1 P37/INT0			External interrupt input	Interrupt edge selection register	(9)	
P40, P41			I/O port P4			(10)

**Note:** Port P10, P12, P13, P36, P37 is CMOS/TTL level.

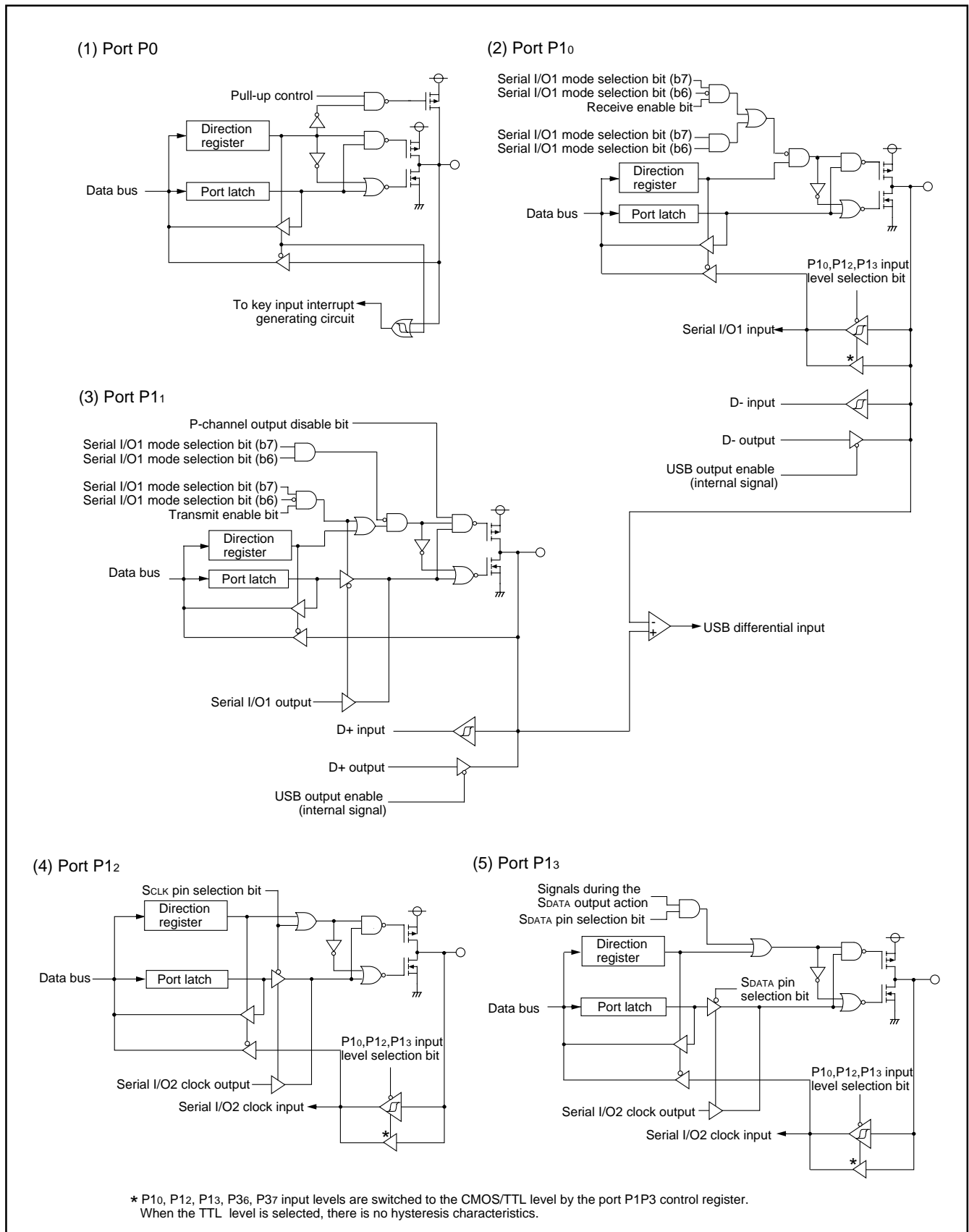
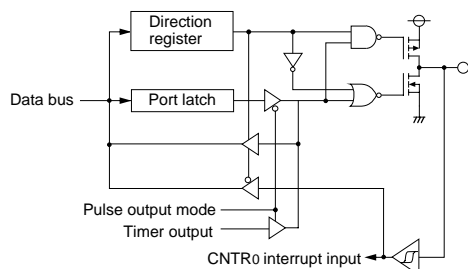


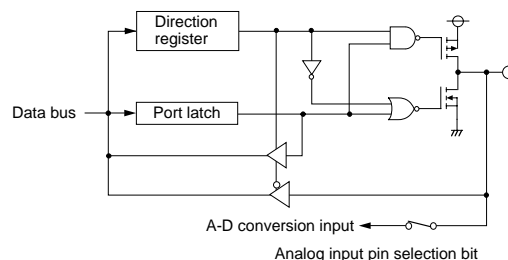
Fig. 13 Block diagram of ports (1)

## FUNCTIONAL DESCRIPTION

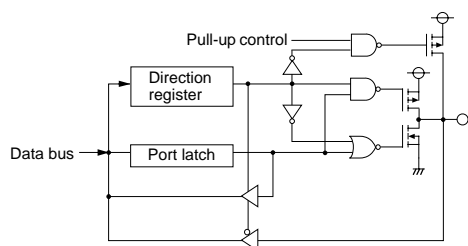
(6) Port P14



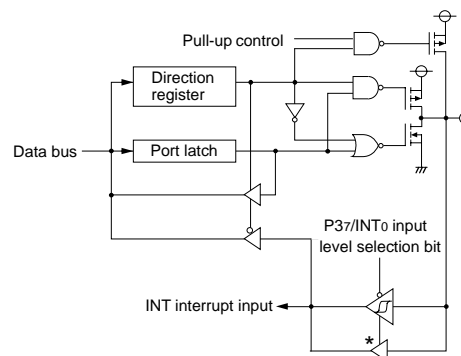
(7) Ports P20 – P27



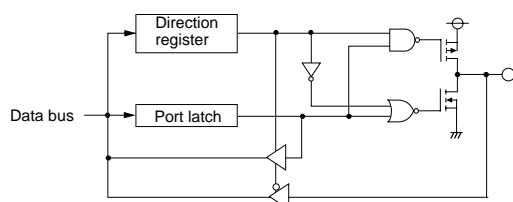
(8) Ports P30 – P35



(9) Port P36, P37



(10) Ports P15, P16, P40, P41



\* P10, P12, P13, P36, P37 input levels are switched to the CMOS/TTL level by the port P1P3 control register. When the TTL level is selected, there is no hysteresis characteristics.

**Fig. 14 Block diagram of ports (2)**

## Interrupts

Interrupts occur by 14 different sources : 4 external sources, 9 internal sources and 1 software source.

### Interrupt control

All interrupts except the BRK instruction interrupt have an interrupt request bit and an interrupt enable bit, and they are controlled by the interrupt disable flag. When the interrupt enable bit and the interrupt request bit are set to "1" and the interrupt disable flag is set to "0", an interrupt is accepted.

The interrupt request bit can be cleared by program but not be set.

The interrupt enable bit can be set and cleared by program.

It becomes usable by switching CNTR0 and A-D interrupt sources with bit 7 of the interrupt edge selection register, timer 2 and serial I/O2 interrupt sources with bit 6, timer X and key-on wake-up interrupt sources with bit 5, and serial I/O transmit and INT1 interrupt sources with bit 4.

The reset and BRK instruction interrupt can never be disabled with any flag or bit. All interrupts except these are disabled when the interrupt disable flag is set.

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

### Interrupt operation

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

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

### Notes on use

When the active edge of an external interrupt (INT0, INT1, CNTR0) is set, the interrupt request bit may be set.

Therefore, please take following sequence:

1. Disable the external interrupt which is selected.
2. Change the active edge in interrupt edge selection register. (in case of CNTR0: Timer X mode register)
3. Clear the set interrupt request bit to "0".
4. Enable the external interrupt which is selected.

**Table 6 Interrupt vector address and priority**

Interrupt source	Priority	Vector addresses (Note 1)		Interrupt request generating conditions	Remarks
		High-order	Low-order		
Reset (Note 2)	1	FFFD <sub>16</sub>	FFFC <sub>16</sub>	At reset input	Non-maskable
UART receive	2	FFFB <sub>16</sub>	FFFA <sub>16</sub>	At completion of UART data receive	Valid in UART mode
USB IN token				At detection of IN token	Valid in USB mode
UART transmit	3	FFF9 <sub>16</sub>	FFF8 <sub>16</sub>	At completion of UART transmit shift or when transmit buffer is empty	Valid in UART mode
USB SETUP/OUT token Reset/Suspend/Resume				At detection of SETUP/OUT token or At detection of Reset/ Suspend/ Resume	Valid in USB mode
INT1				At detection of either rising or falling edge of INT1 input	External interrupt (active edge selectable)
INT0	4	FFF7 <sub>16</sub>	FFF6 <sub>16</sub>	At detection of either rising or falling edge of INT0 input	External interrupt (active edge selectable)
Timer X	5	FFF5 <sub>16</sub>	FFF4 <sub>16</sub>	At timer X underflow	External interrupt (valid at falling)
Key-on wake-up				At falling of conjunction of input logical level for port P0 (at input)	
Timer 1	6	FFF3 <sub>16</sub>	FFF2 <sub>16</sub>	At timer 1 underflow	STP release timer underflow
Timer 2	7	FFF1 <sub>16</sub>	FFF0 <sub>16</sub>	At timer 2 underflow	
Serial I/O2				At completion of transmit/receive shift	
CNTR0	8	FFEF <sub>16</sub>	FFEE <sub>16</sub>	At detection of either rising or falling edge of CNTR0 input	External interrupt (active edge selectable)
A-D conversion				At completion of A-D conversion	
BRK instruction	9	FFED <sub>16</sub>	FFEC <sub>16</sub>	At BRK instruction execution	Non-maskable software interrupt

**Note 1:** Vector addressed contain internal jump destination addresses.

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

# HARDWARE

## FUNCTIONAL DESCRIPTION

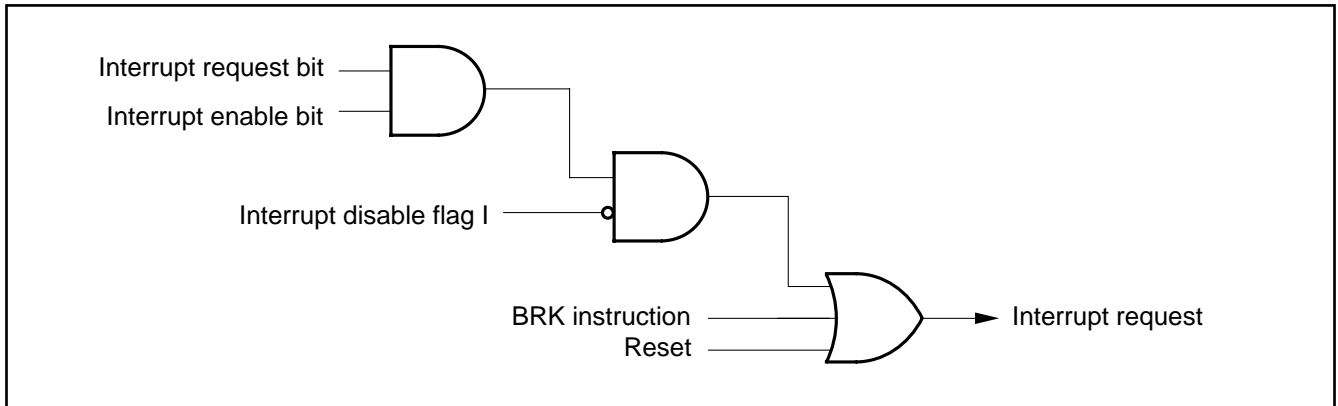


Fig. 15 Interrupt control

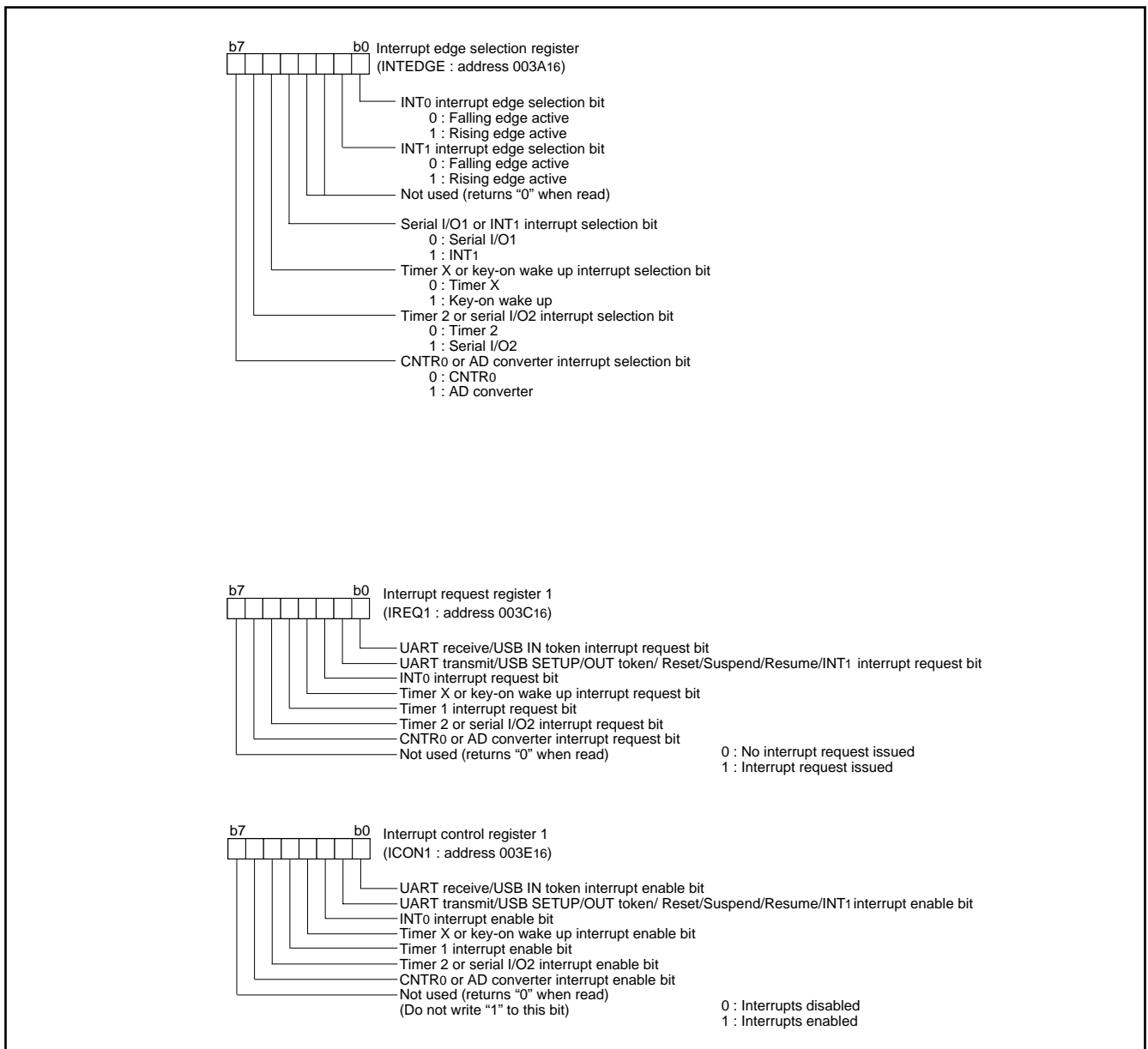


Fig. 16 Structure of interrupt-related registers

### Key Input Interrupt (Key-On Wake-Up)

A key-on wake-up interrupt request is generated by applying "L" level to any pin of port P0 that has been set to input mode.

In other words, it is generated when the AND of input level goes from "1" to "0". An example of using a key input interrupt is shown in Figure 14, where an interrupt request is generated by pressing one of the keys provided as an active-low key matrix which uses ports P00 to P03 as input ports.

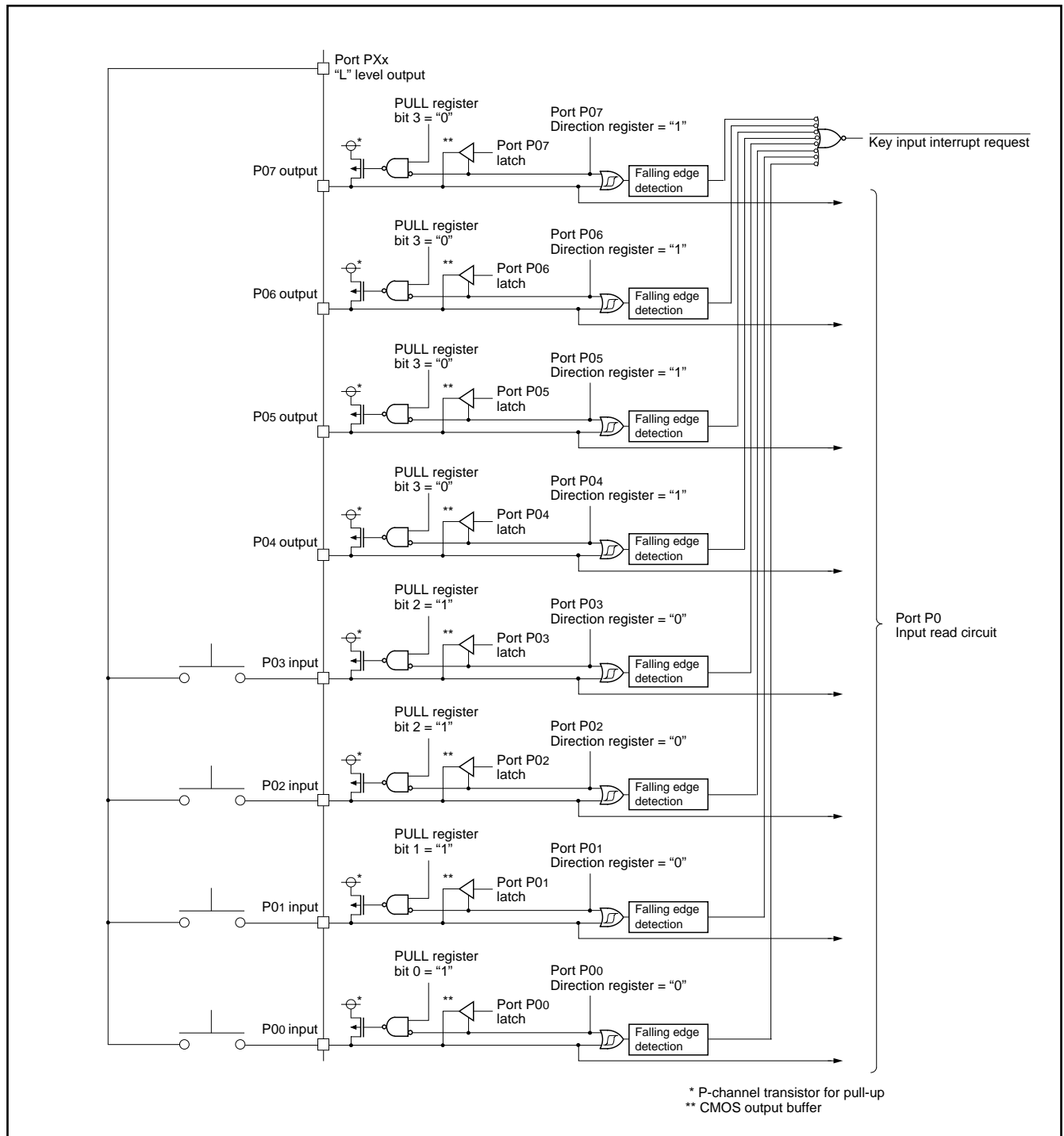


Fig. 17 Connection example when using key input interrupt and port P0 block diagram



# HARDWARE

## FUNCTIONAL DESCRIPTION

### Timers

The 7536 Group has 3 timers: timer X, timer 1 and timer 2.

The division ratio of every timer and prescaler is  $1/(n+1)$  provided that the value of the timer latch or prescaler is  $n$ .

All the timers are down count timers. When a timer reaches "0", an underflow occurs at the next count pulse, and the corresponding timer latch is reloaded into the timer. When a timer underflows, the interrupt request bit corresponding to each timer is set to "1".

#### ●Timer 1, Timer 2

Prescaler 12 always counts  $f(X_{IN})/16$ . Timer 1 and timer 2 always count the prescaler output and periodically sets the interrupt request bit.

#### ●Timer X

Timer X can be selected in one of 4 operating modes by setting the timer X mode register.

##### • Timer Mode

The timer counts the signal selected by the timer X count source selection bit.

##### • Pulse Output Mode

The timer counts the signal selected by the timer X count source selection bit, and outputs a signal whose polarity is inverted each time the timer value reaches "0", from the CNTR<sub>0</sub> pin.

When the CNTR<sub>0</sub> active edge switch bit is "0", the output of the CNTR<sub>0</sub> pin is started with an "H" output.

At "1", this output is started with an "L" output. When using a timer in this mode, set the port P14 direction register to output mode.

##### • Event Counter Mode

The operation in the event counter mode is the same as that in the timer mode except that the timer counts the input signal from the CNTR<sub>0</sub> pin.

When the CNTR<sub>0</sub> active edge switch bit is "0", the timer counts the rising edge of the CNTR<sub>0</sub> pin. When this bit is "1", the timer counts the falling edge of the CNTR<sub>0</sub> pin.

##### • Pulse Width Measurement Mode

When the CNTR<sub>0</sub> active edge switch bit is "0", the timer counts the signal selected by the timer X count source selection bit while the CNTR<sub>0</sub> pin is "H". When this bit is "1", the timer counts the signal while the CNTR<sub>0</sub> pin is "L".

In any mode, the timer count can be stopped by setting the timer X count stop bit to "1". Each time the timer overflows, the interrupt request bit is set.

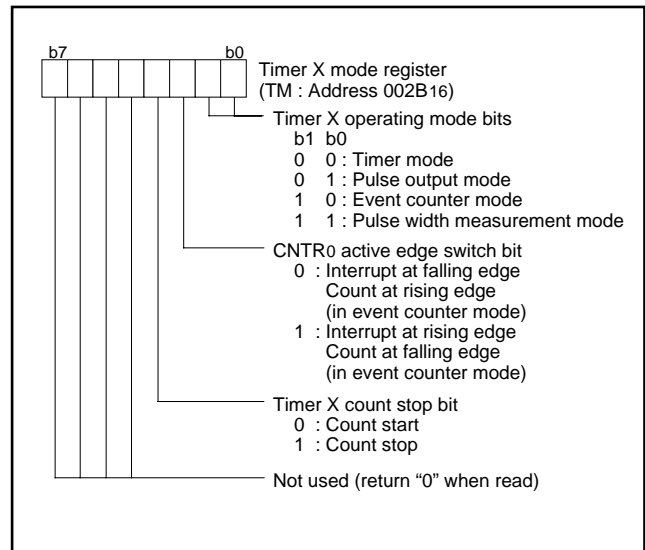


Fig. 18 Structure of timer X mode register

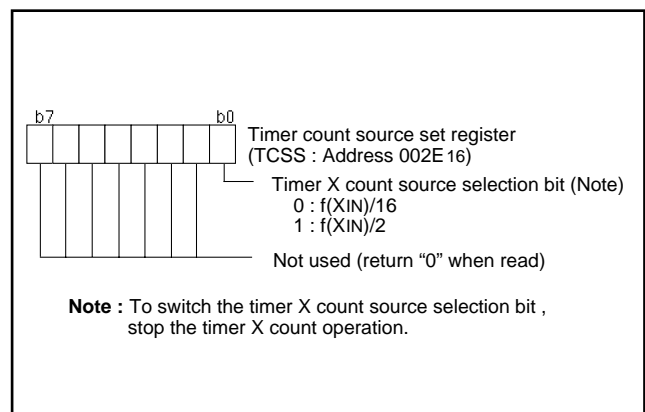


Fig. 19 Timer count source setting register

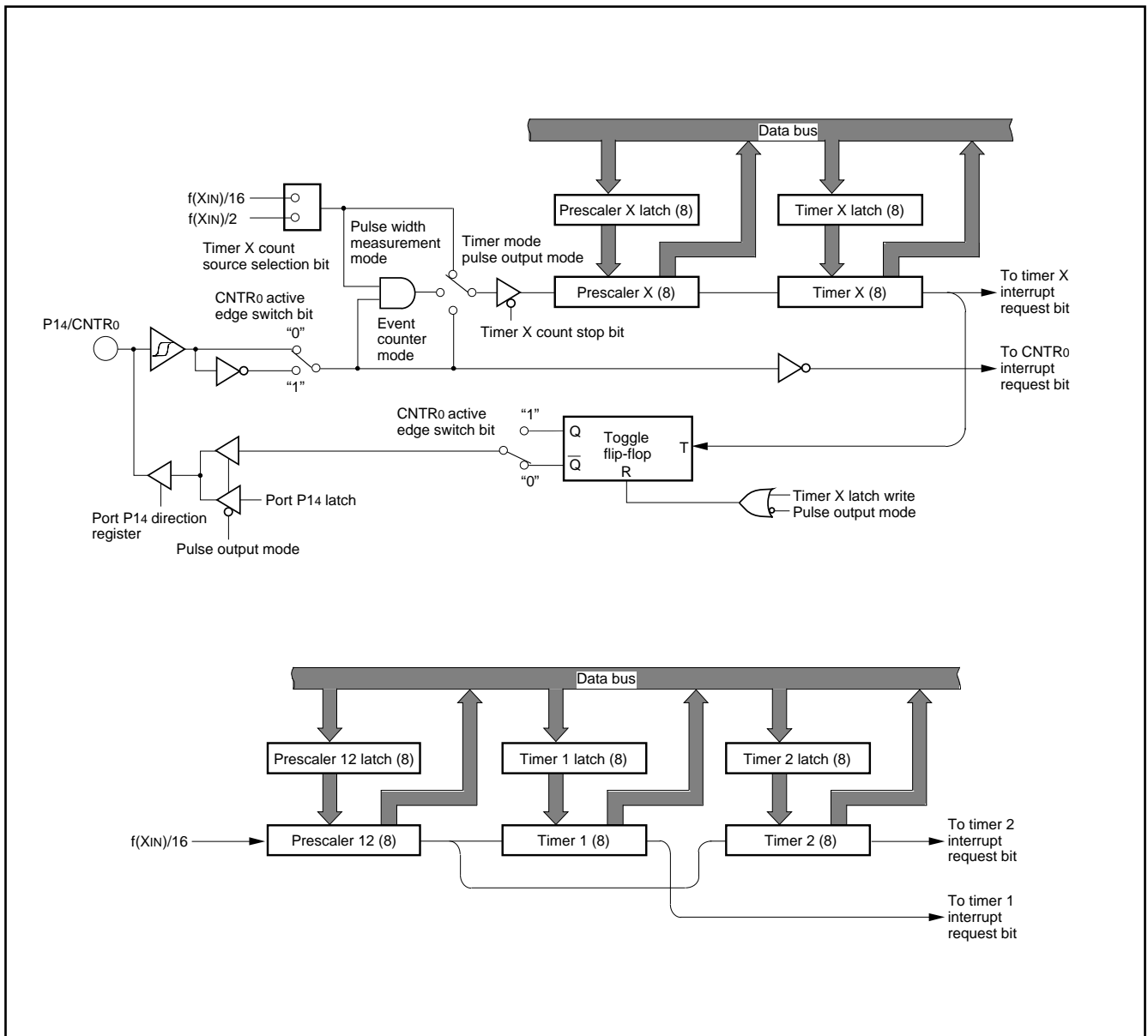


Fig. 20 Block diagram of timer X, timer 1 and timer 2

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Serial I/O

#### ●Serial I/O1

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

Serial I/O1 can be used as an asynchronous (UART) serial I/O. A dedicated timer (baud rate generator) is also provided for baud rate generation when serial I/O1 is in operation.

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

Each of the transmit and receive shift registers has a buffer register

(the same address on memory). Since the shift register cannot be written to or read from directly, transmit data is written to the transmit buffer, and receive data is read from the respective buffer registers. These buffer registers can also hold the next data to be transmitted and receive 2-byte receive data in succession.

By selecting "1" for continuous transmit valid bit (bit 2 of SIO1CON), continuous transmission of the same data is made possible. This can be used as a simplified PWM.

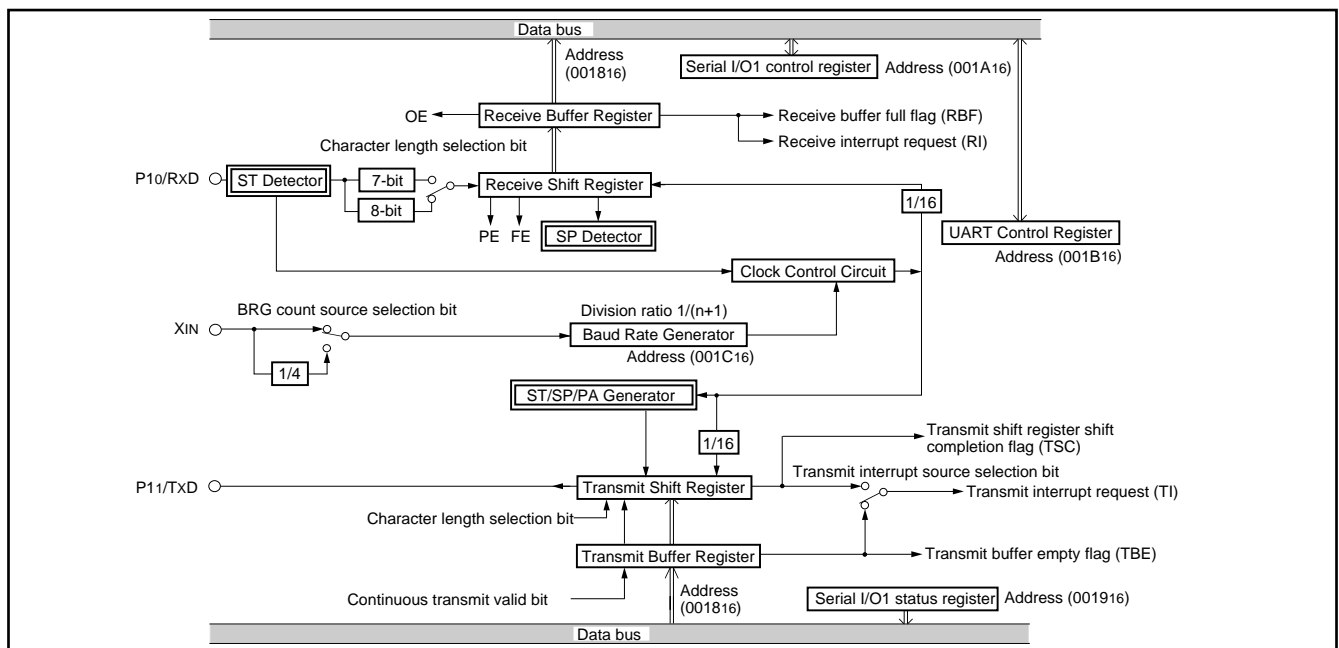


Fig. 21 Block diagram of UART serial I/O

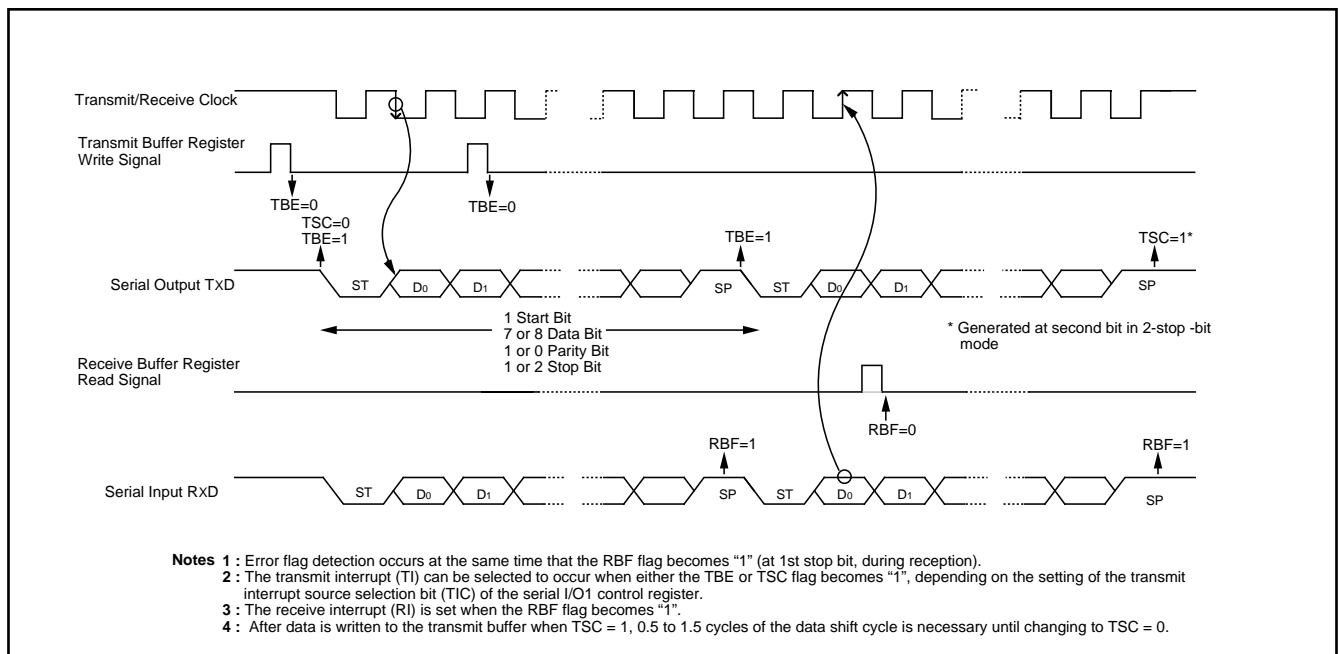


Fig. 22 Operation of UART serial I/O function

### [Serial I/O1 control register] SIO1CON

The serial I/O1 control register consists of eight control bits for the serial I/O1 function.

### [UART control register] UARTCON

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 a data transfer. One bit in this register (bit 4) is always valid and sets the input/output structure of the P1<sub>1</sub>/TxD pin.

### [UART status register] UARTSTS

The read-only UART status register consists of seven flags (bits 0 to 6) which indicate the operating status of the UART function and various errors. This register functions as the UART status register (UARTSTS) when selecting the UART.

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

If there is an error, it is detected at the same time that data is transferred from the receive shift register to the receive buffer, and the receive buffer full flag is set. A write to the serial I/O1 status register clears all the error flags OE, PE, FE, and SE (bit 3 to bit 6, respectively). Writing "0" to the serial I/O1 mode selection bits MOD1 and MOD0 (bit 7 and 6 of the Serial I/O1 control register) also clears all the status flags, including the error flags.

All bits of the serial I/O1 status register are initialized to "81<sub>16</sub>" at reset, but if the transmit enable bit (bit 4) of the serial I/O1 control register has been set to "1", the continuous transmit valid bit (bit 2) becomes "1".

### [Transmit/Receive buffer register] TB/RB

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

### [Baud Rate Generator] BRG

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

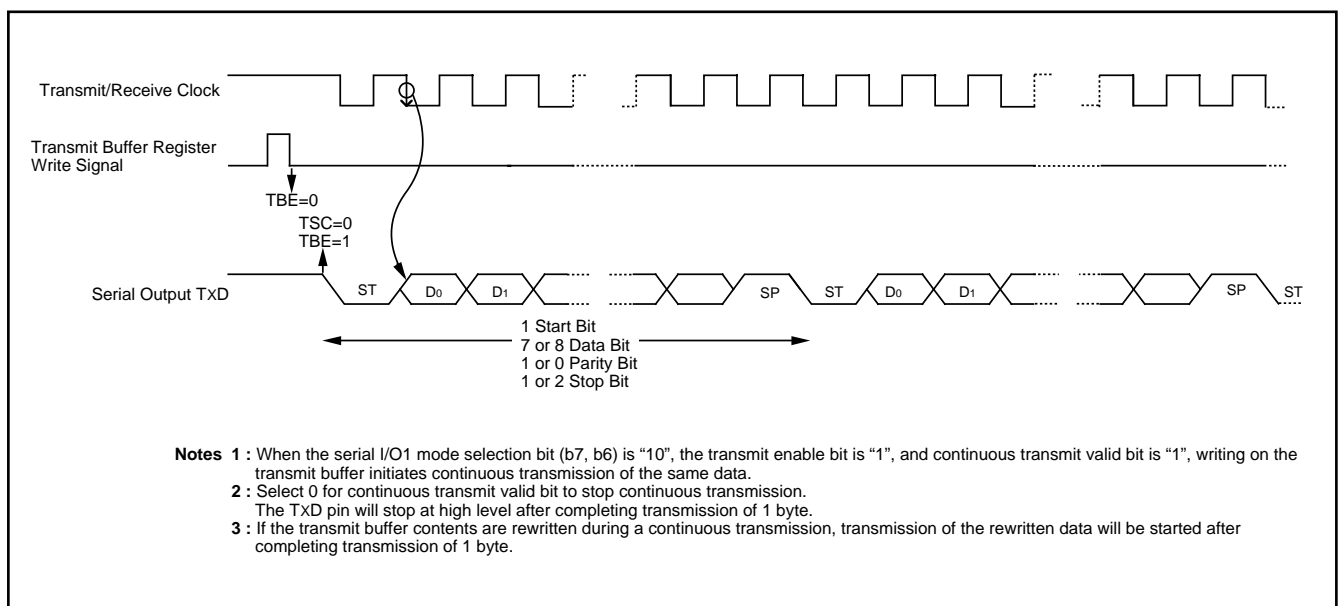


Fig. 23 Continuous transmission operation of UART serial I/O

# HARDWARE

## FUNCTIONAL DESCRIPTION

### • Universal serial bus (USB) mode

By setting bits 7 and 6 of the serial I/O1 control register (address 001A16) to “11”, the USB mode is selected.

This mode conforms to “Low Speed device” of USB Specification 1.0. In this mode serial I/O1 interrupt have 6 sources; USB in and out token receive, USB reset, suspend, and resume. The USB/UART

status register (address 001916) functions as the USB status register (USBSTS). There is the USBVREFOUT pin for the USB reference voltage output, and a D-line with 1.5 kΩ external resistor can be pull up.

USB mode block and USB transceiver block show in figures 21 and 22.

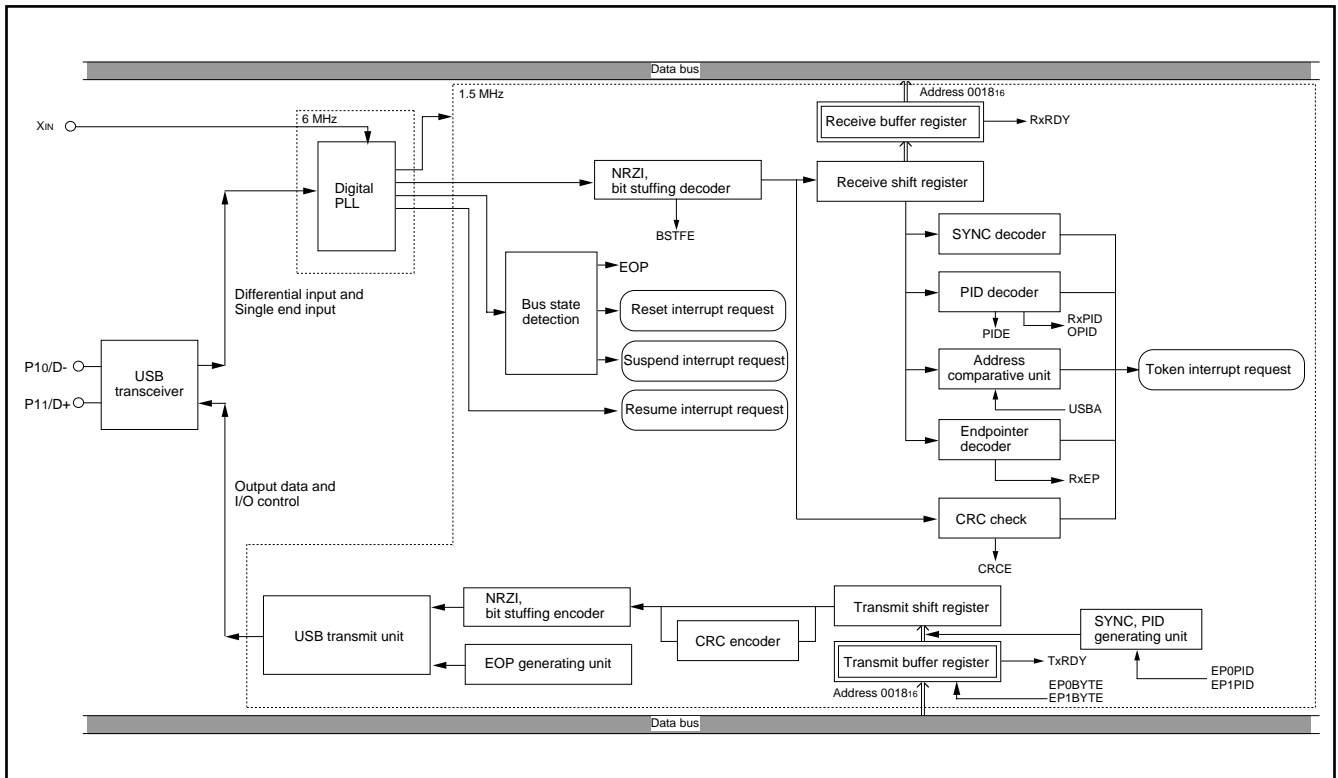


Fig. 24 USB mode block diagram

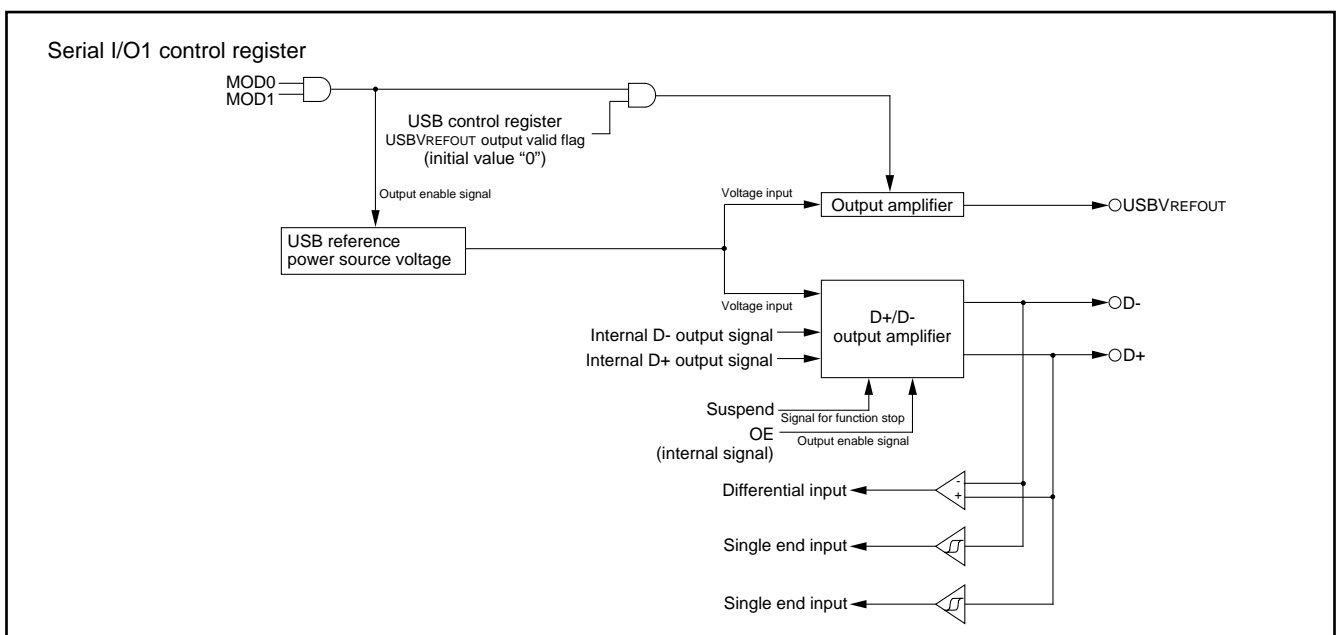


Fig. 25 USB transceiver block diagram

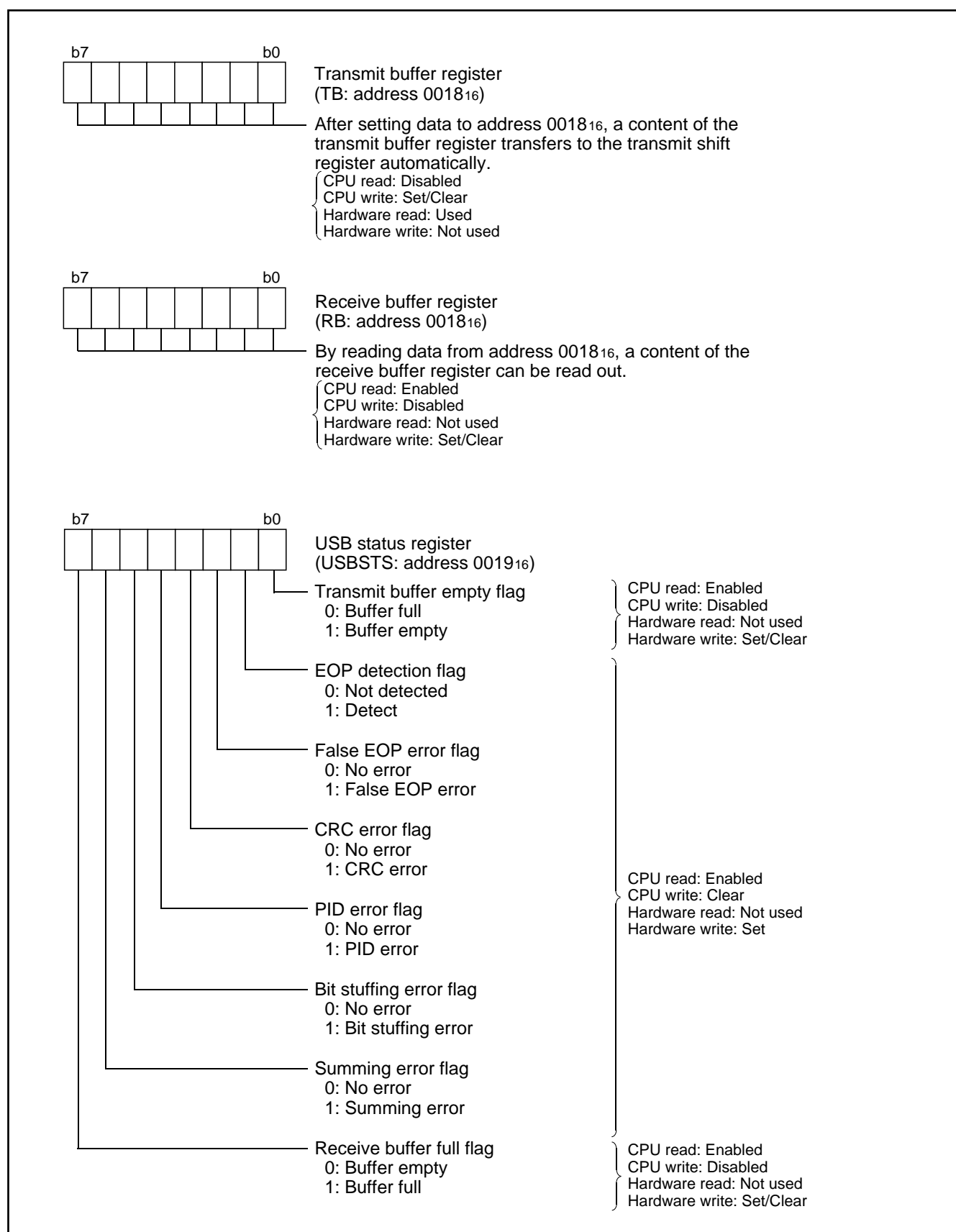


Fig. 26 Structure of serial I/O1-related registers (1)

# HARDWARE

## FUNCTIONAL DESCRIPTION

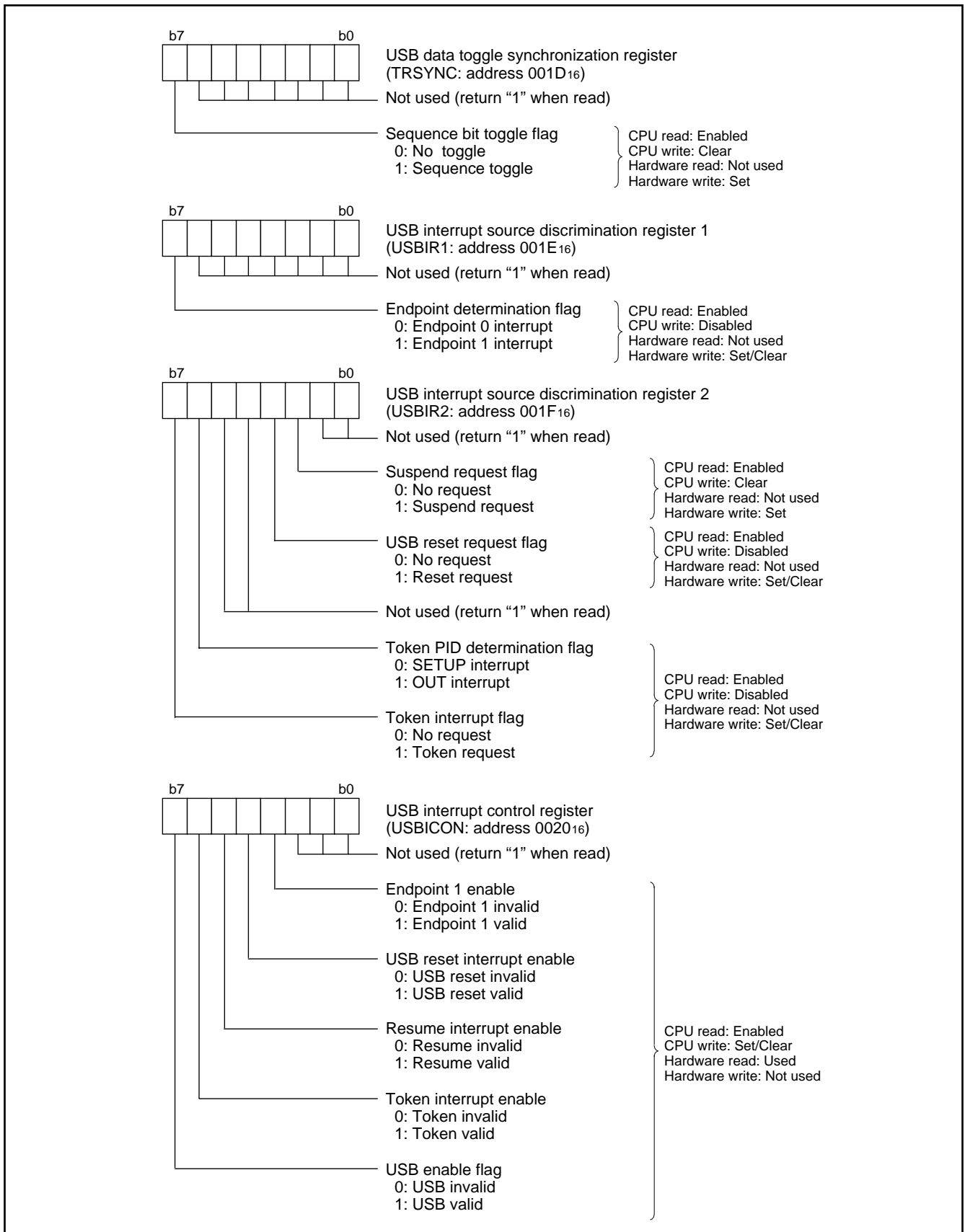


Fig. 27 Structure of serial I/O1-related registers (2)

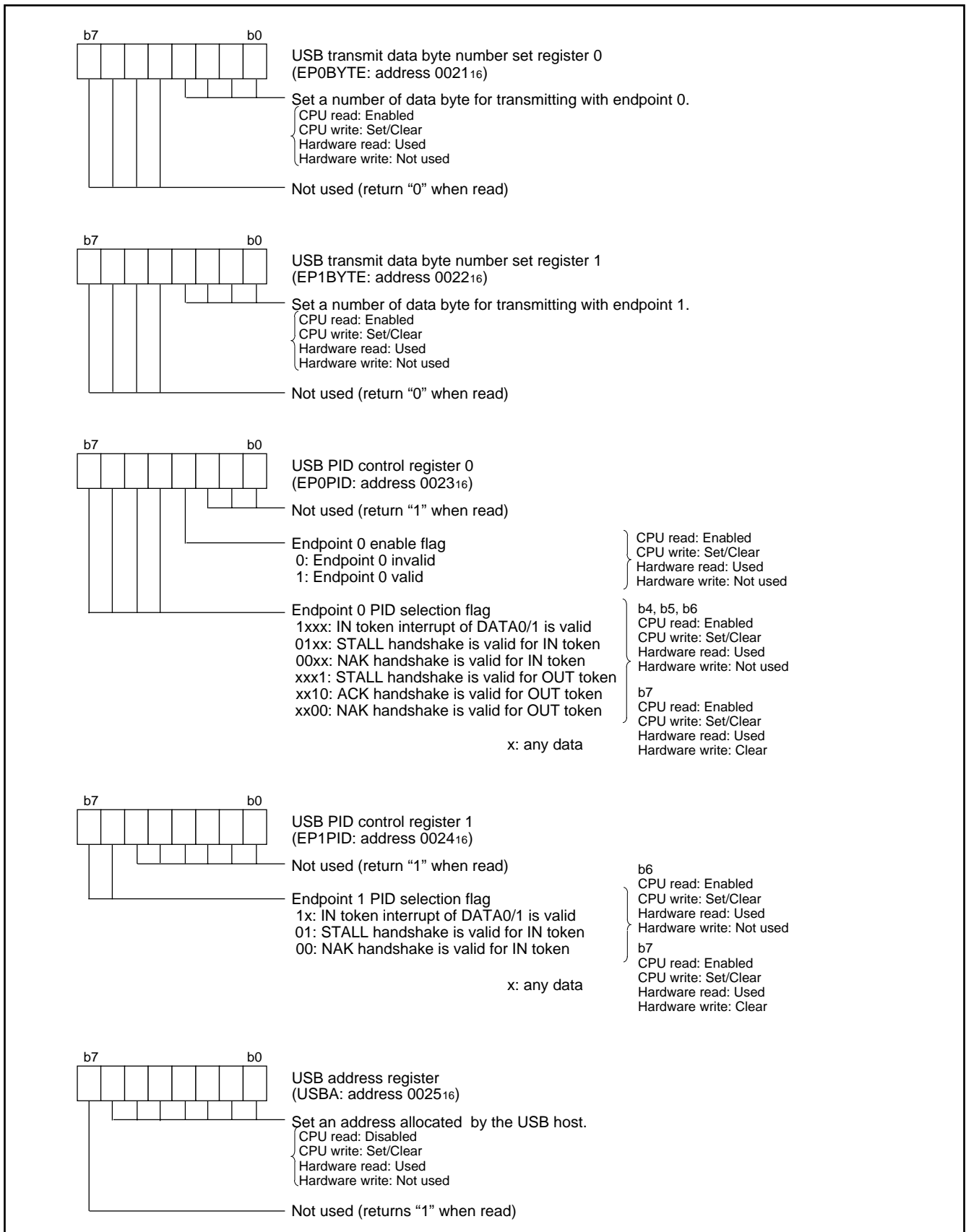


Fig. 28 Structure of serial I/O1-related registers (3)



# HARDWARE

## FUNCTIONAL DESCRIPTION

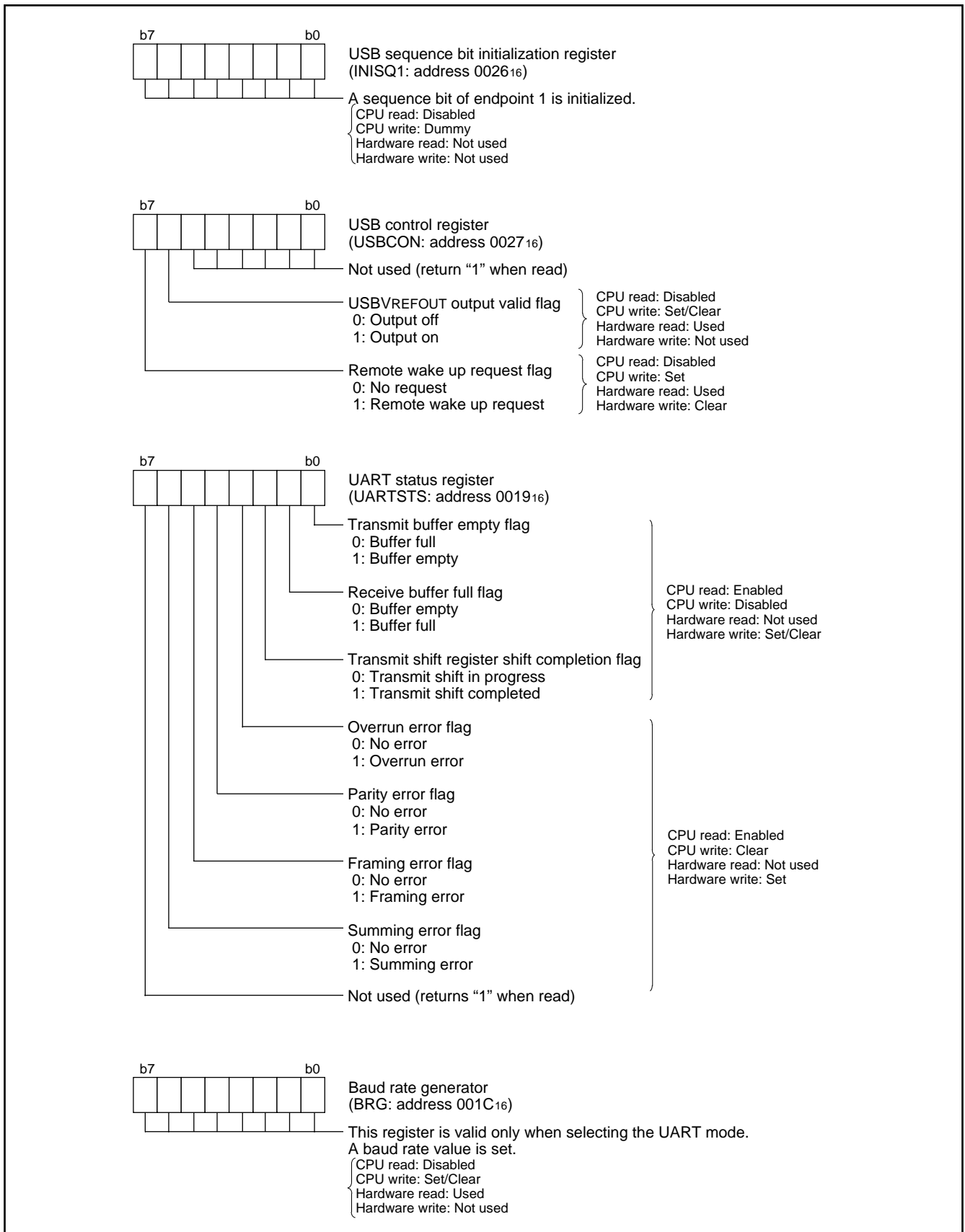


Fig. 29 Structure of serial I/O1-related registers (4)

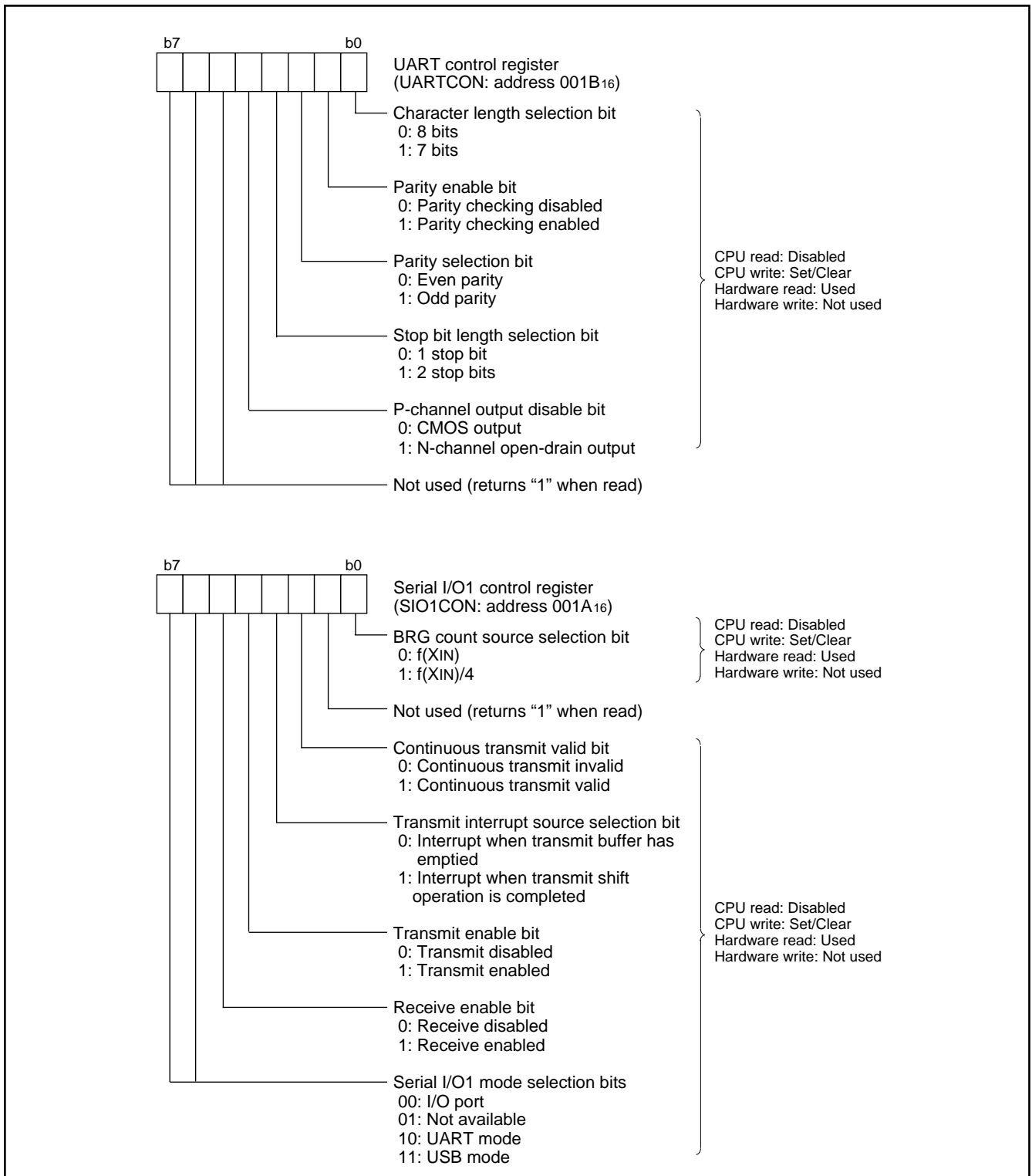


Fig. 30 Structure of serial I/O1-related registers (5)

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Note on using USB mode

#### Handling of SE0 signal in program (at receiving)

7536 group has the border line to detect as USB RESET or EOP (End of Packet) on the width of SE0 (Single Ended 0).

A response apposite to a state of the device is expected.

The name of the following short words which is used in table 7 shows as follow.

- TKNE: Token interrupt enable (bit 6 of address 20<sub>16</sub>)
- RSME: Resume interrupt enable (bit 5 of address 20<sub>16</sub>)
- RSTE: USB reset interrupt enable (bit 4 of address 20<sub>16</sub>)
- Spec: A response of the device requested by USB Specification 1.0
- SIE: Hardware operation in 7536 group
- F/W: Recommendation process in the program
- FEOPE: False EOP error flag (bit 2 of address 19<sub>16</sub>)
- RxPID: Token interrupt flag (bit 7 of address 1F<sub>16</sub>)

**Table 7 Relation of the width of SE0 and the state of the device**

		State of device			
Width of SE0		Idle state TKNE = X RSME = 0 RSTE =1	End of Token in transaction TKNE = 1 RSME = 0 RSTE =1	End of data or handshake in transaction TKNE = 0 RSME = 0 RSTE = 0 or 1	Suspend state TKNE = 0 RSME = 1 RSTE = 0
0 μ sec. 0.5 μ sec.	Spec	Ignore	Ignore	Ignore	Spec Reset or resume
	SIE	Keep counting suspend timer	Not detected as EOP(in case of no detection EOP, SIE returns idle state as time out. FEOPE flag is set.)	Not detected as EOP(in case of no detection EOP, SIE returns idle state as timeup. FEOPE flag is set.)	
	F/W	Not acknowledge	Not acknowledge	Wait for the next EOP flag	
0.5 μ sec. 2.5 μ sec.	Spec	Keep alive	EOP	EOP	SIE Reset interrupt request
	SIE	Initialize suspend timer count value	Token interrupt request	Set EOP flag	
	F/W	Not acknowledge	Token interrupt processing execute	After checking the set of EOP flag, go to the next processing	
2.5 μ sec. 2.67 μ sec.	Spec	Keep alive or Reset	EOP or Reset	EOP or Reset	F/W Reset interrupt processing Resume interrupt processing
	SIE	may determine as keep alive and Reset interrupt	may determine as EOP and Reset interrupt	may determine as EOP and Reset interrupt	
	F/W	Keep alive in case of no interrupt request Reset processing in case of interrupt request	RxPID = 1> Token interrupt processing RxPID = 0> Reset interrupt processing	Continue the processing in case of no interrupt request Reset processing in case of interrupt request	
2.67 μ sec.	Spec	Reset	Reset	Reset	
	SIE	Reset interrupt request	Reset interrupt request	Reset interrupt request	
	F/W	Reset processing	Reset processing	Reset processing	

### ●Serial I/O2

The serial I/O2 function can be used only for clock synchronous serial I/O.

For clock synchronous serial I/O2 the transmitter and the receiver must use the same clock. When the internal clock is used, transfer is started by a write signal to the serial I/O2 register.

#### [Serial I/O2 control register] SIO2CON

The serial I/O2 control register contains 8 bits which control various serial I/O functions.

- For receiving, set "0" to bit 3.
- When receiving, bit 7 is cleared by writing dummy data to serial I/O2 register after shift is completed.
- Bit 7 is set earlier a half cycle of shift clock than completion of shift operation. Accordingly, when checking shift completion by using this bit, the setting is as follows:
  - (1) check that this bit is set to "1",
  - (2) wait a half cycle of shift clock,
  - (3) read/write to serial I/O2 register.

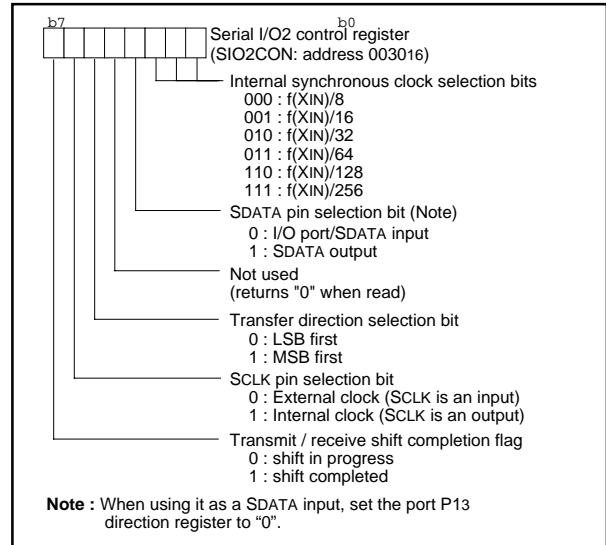


Fig. 31 Structure of serial I/O2 control registers

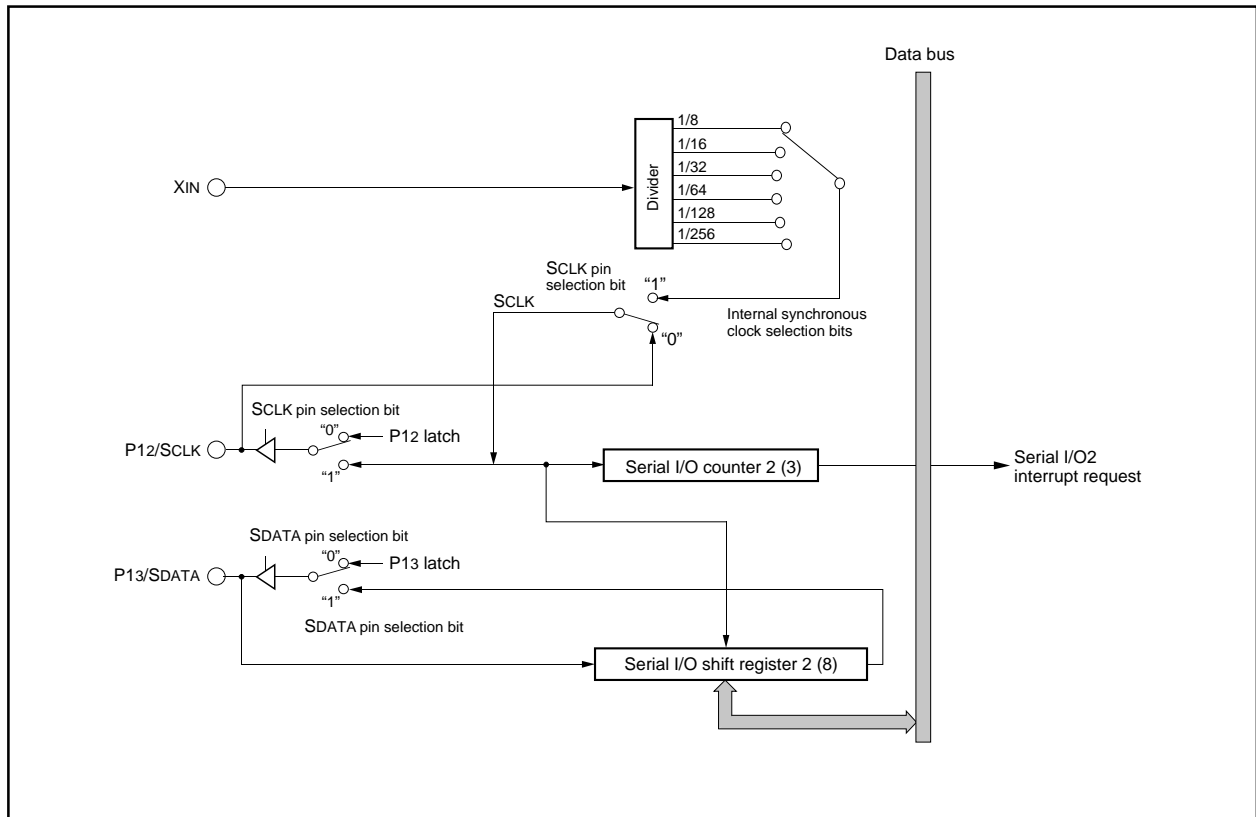


Fig. 32 Block diagram of serial I/O2

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Serial I/O2 operation

By writing to the serial I/O2 register(address 003116) the serial I/O2 counter is set to "7".

After writing, the SDATA pin outputs data every time the transfer clock shifts from a high to a low level. And, as the transfer clock shifts from a low to a high, the SDATA pin reads data, and at the same time the contents of the serial I/O2 register are shifted by 1 bit.

When the internal clock is selected as the transfer clock source, the following operations execute as the transfer clock counts up to 8.

- Serial I/O2 counter is cleared to "0".
- Transfer clock stops at an "H" level.
- Interrupt request bit is set.
- Shift completion flag is set.

Also, the SDATA pin is in a high impedance state after the data transfer is complete (refer to figure 33).

When the external clock is selected as the transfer clock source, the interrupt request bit is set as the transfer clock counts up to 8, but external control of the clock is required since it does not stop. Notice that the SDATA pin is not in a high impedance state on the completion of data transfer.

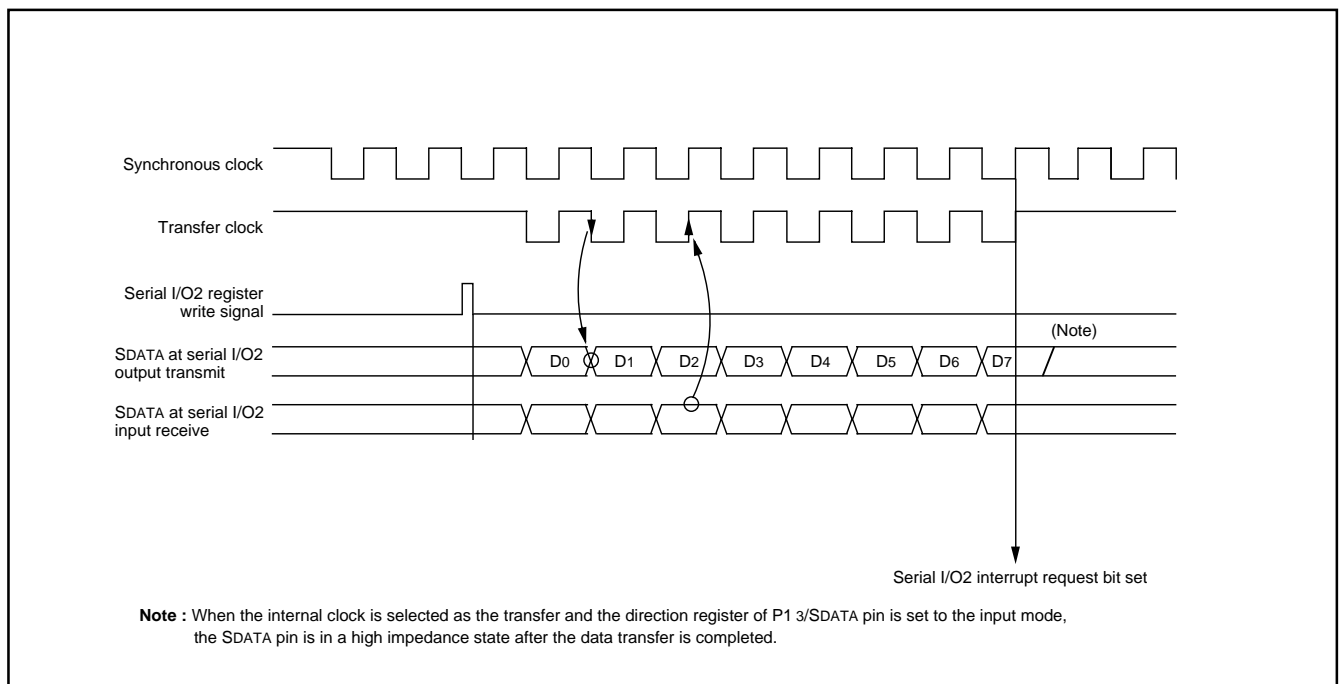


Fig. 33 Serial I/O2 timing (LSB first)

### A-D Converter

The functional blocks of the A-D converter are described below.

#### [A-D conversion register] AD

The A-D conversion register is a read-only register that stores the result of A-D conversion. Do not read this register during an A-D conversion.

#### [A-D control register] ADCON

The A-D control register controls the A-D converter. Bit 2 to 0 are analog input pin selection bits. Bit 4 is the AD conversion completion bit. The value of this bit remains at "0" during A-D conversion, and changes to "1" at completion of A-D conversion.

A-D conversion is started by setting this bit to "0".

#### [Comparison voltage generator]

The comparison voltage generator divides the voltage between VSS and VREF by 1024 by a resistor ladder, and outputs the divided voltages. Since the generator is disconnected from VREF pin and VSS pin, current is not flowing into the resistor ladder.

#### [Channel Selector]

The channel selector selects one of ports P27/AN7 to P20/AN0, and inputs the voltage to the comparator.

#### [Comparator and control circuit]

The comparator and control circuit compares an analog input voltage with the comparison voltage and stores its result into the A-D conversion register. When A-D conversion is completed, the control circuit sets the AD conversion completion bit and the AD interrupt request bit to "1". Because the comparator is constructed linked to a capacitor, set f(XIN) to 500 kHz or more during A-D conversion.

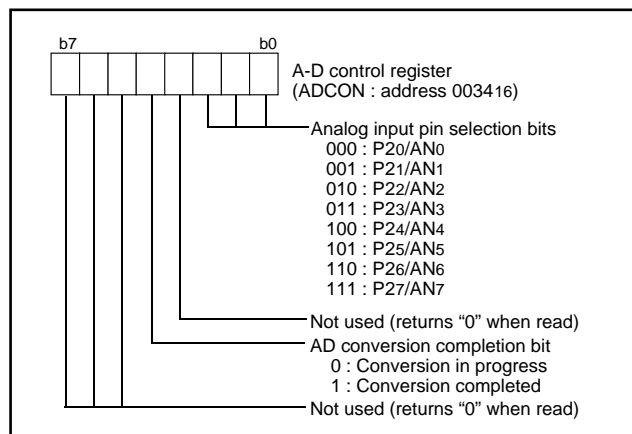


Fig. 34 Structure of A-D control register

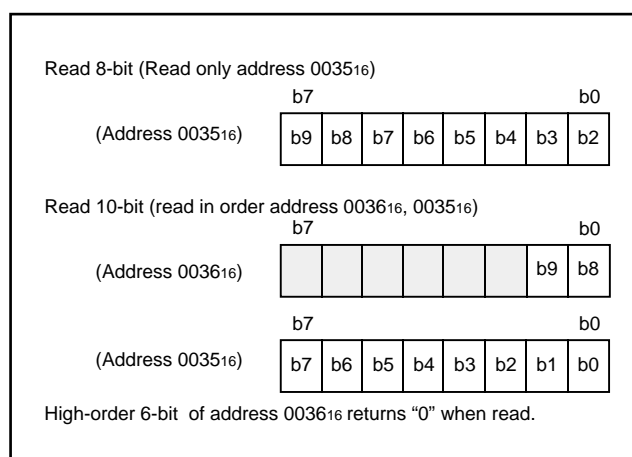


Fig. 35 Structure of A-D conversion register

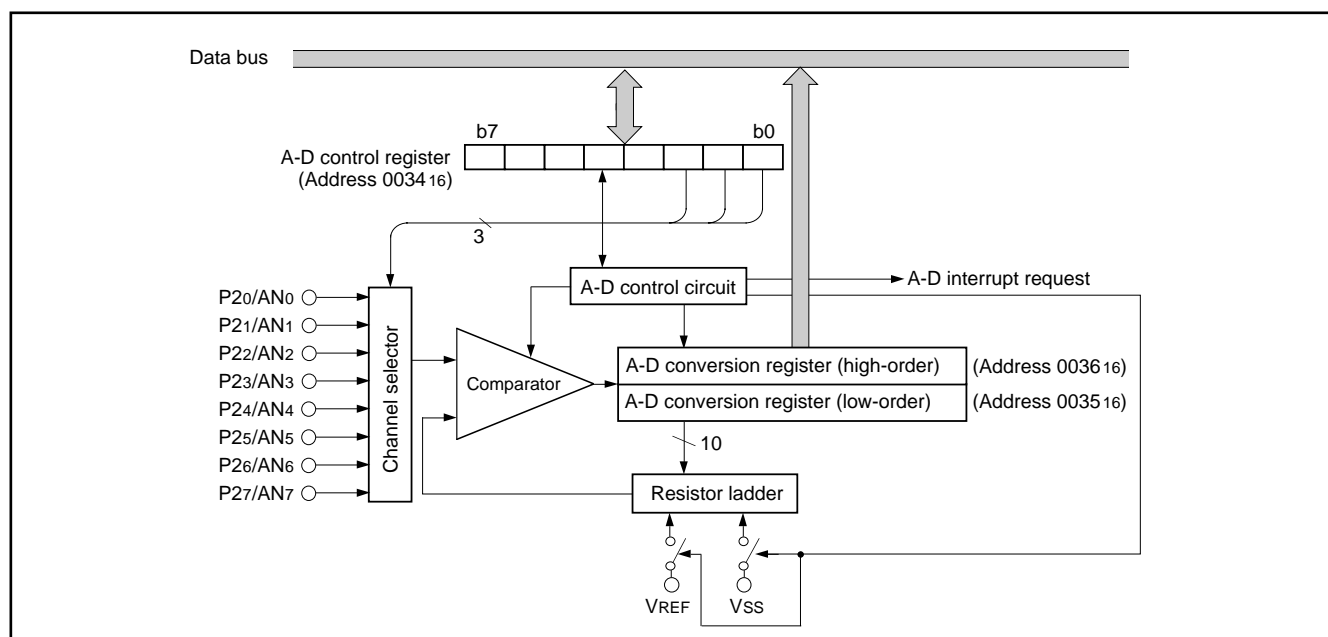


Fig. 36 Block diagram of A-D converter

# HARDWARE

## FUNCTIONAL DESCRIPTION

### Watchdog Timer

The watchdog timer gives a means for returning to a reset status when the program fails to run on its normal loop due to a runaway. The watchdog timer consists of an 8-bit watchdog timer H and an 8-bit watchdog timer L, being a 16-bit counter.

#### Standard operation of watchdog timer

The watchdog timer stops when the watchdog timer control register (address 0039<sub>16</sub>) is not set after reset. Writing an optional value to the watchdog timer control register (address 0039<sub>16</sub>) causes the watchdog timer to start to count down. When the watchdog timer H underflows, an internal reset occurs. Accordingly, it is programmed that the watchdog timer control register (address 0039<sub>16</sub>) can be set before an underflow occurs.

When the watchdog timer control register (address 0039<sub>16</sub>) is read, the values of the high-order 6-bit of the watchdog timer H, STP instruction disable bit and watchdog timer H count source selection bit are read.

#### Initial value of watchdog timer

By a reset or writing to the watchdog timer control register (address 0039<sub>16</sub>), the watchdog timer H is set to "FF<sub>16</sub>" and the watchdog timer L is set to "FF<sub>16</sub>".

#### Operation of watchdog timer H count source selection bit

A watchdog timer H count source can be selected by bit 7 of the watchdog timer control register (address 0039<sub>16</sub>). When this bit is "0", the count source becomes a watchdog timer L underflow signal. The detection time is 174.763 ms at  $f(XIN)=6$  MHz.

When this bit is "1", the count source becomes  $f(XIN)/16$ . In this case, the detection time is 683  $\mu$ s at  $f(XIN)=6$  MHz.

This bit is cleared to "0" after reset.

#### Operation of STP instruction disable bit

When the watchdog timer is in operation, the STP instruction can be disabled by bit 6 of the watchdog timer control register (address 0039<sub>16</sub>).

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

When this bit is "1", the STP instruction is disabled, and an internal reset occurs if the STP instruction is executed.

Once this bit is set to "1", it cannot be changed to "0" by program.

This bit is cleared to "0" after reset.

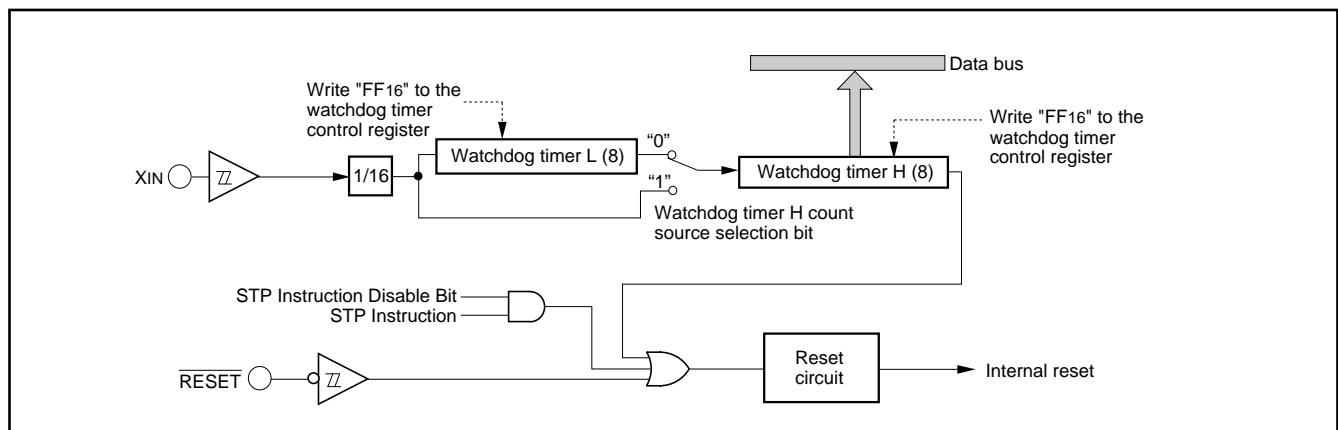


Fig. 37 Block diagram of watchdog timer

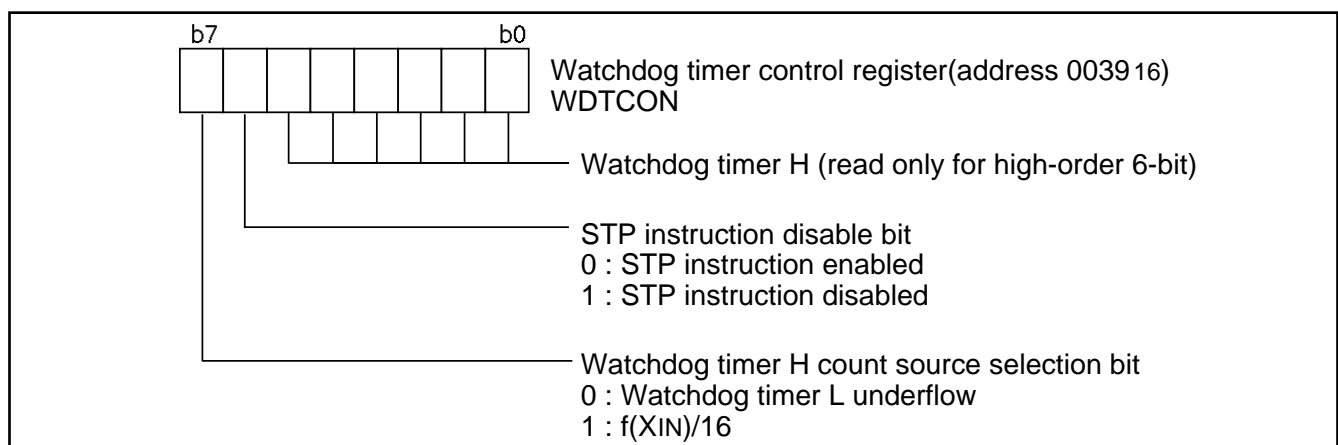


Fig. 38 Structure of watchdog timer control register

### Reset Circuit

The microcomputer is put into a reset status by holding the  $\overline{\text{RESET}}$  pin at the "L" level for 15  $\mu\text{s}$  or more when the power source voltage is 4.1 to 5.5 V and  $X_{\text{IN}}$  is in stable oscillation.

After that, this reset status is released by returning the  $\overline{\text{RESET}}$  pin to the "H" level. The program starts from the address having the contents of address  $\text{FFFD}_{16}$  as high-order address and the contents of address  $\text{FFFC}_{16}$  as low-order address.

Note that the reset input voltage should be 0.82 V or less when the power source voltage passes 4.1 V.

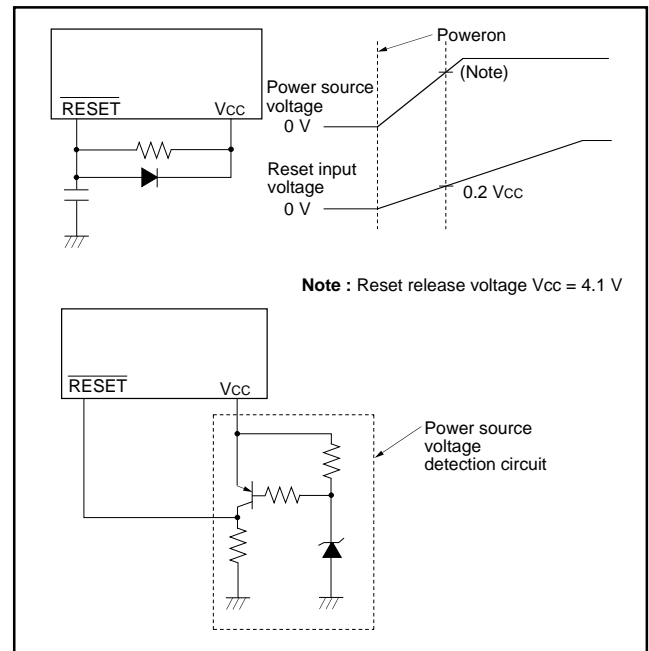


Fig. 39 Example of reset circuit

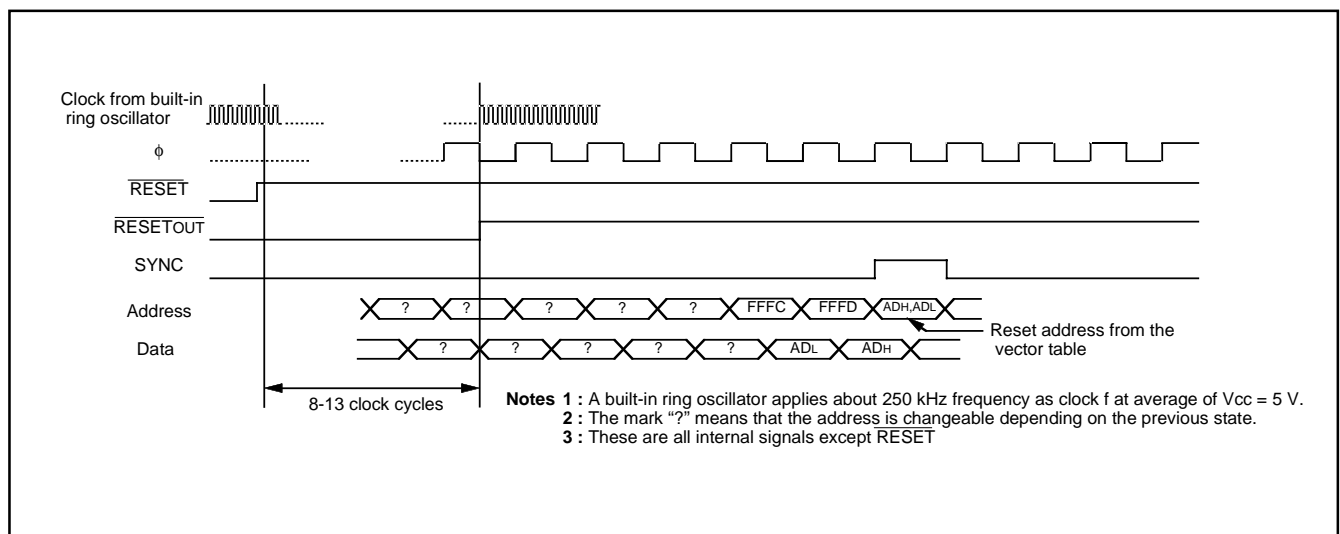


Fig. 40 Timing diagram at reset



# HARDWARE

## FUNCTIONAL DESCRIPTION

	Address	Register contents
(1) Port P0 direction register	0001 <sub>16</sub>	00 <sub>16</sub>
(2) Port P1 direction register	0003 <sub>16</sub>	X 0 0 0 0 0 0 0
(3) Port P2 direction register	0005 <sub>16</sub>	00 <sub>16</sub>
(4) Port P3 direction register	0007 <sub>16</sub>	00 <sub>16</sub>
(5) Port P4 direction register	0009 <sub>16</sub>	X X X X X X 0 0
(6) Pull-up control register	0016 <sub>16</sub>	FF <sub>16</sub>
(7) USB/UART status register	0019 <sub>16</sub>	1 0 0 0 0 0 0 1
(8) Serial I/O1 control register	001A <sub>16</sub>	02 <sub>16</sub>
(9) UART control register	001B <sub>16</sub>	1 1 1 0 0 0 0 0
(10) USB data toggle synchronization register	001D <sub>16</sub>	0 1 1 1 1 1 1 1
(11) USB interrupt source discrimination register 1	001E <sub>16</sub>	0 1 1 1 1 1 1 1
(12) USB interrupt source discrimination register 2	001F <sub>16</sub>	0 1 1 1 0 0 1 1
(13) USB interrupt control register	0020 <sub>16</sub>	0 0 0 0 0 1 1 1
(14) USB transmit data byte number set register 0	0021 <sub>16</sub>	00 <sub>16</sub>
(15) USB transmit data byte number set register 1	0022 <sub>16</sub>	00 <sub>16</sub>
(16) USBPID control register 0	0023 <sub>16</sub>	0 0 0 0 0 1 1 1
(17) USBPID control register 1	0024 <sub>16</sub>	0 0 1 1 1 1 1 1
(18) USB address register	0025 <sub>16</sub>	1 0 0 0 0 0 0 0
(19) USB sequence bit initialization register	0026 <sub>16</sub>	1 1 1 1 1 1 1 1
(20) USB control register	0027 <sub>16</sub>	0 0 1 1 1 1 1 1
(21) Prescaler 12	0028 <sub>16</sub>	FF <sub>16</sub>
(22) Timer 1	0029 <sub>16</sub>	01 <sub>16</sub>
(23) Timer 2	002A <sub>16</sub>	00 <sub>16</sub>
(24) Timer X mode register	002B <sub>16</sub>	00 <sub>16</sub>
(25) Prescaler X	002C <sub>16</sub>	FF <sub>16</sub>
(26) Timer X	002D <sub>16</sub>	FF <sub>16</sub>
(27) Timer count source set register	002E <sub>16</sub>	00 <sub>16</sub>
(28) Serial I/O2 control register	0030 <sub>16</sub>	00 <sub>16</sub>
(29) A-D control register	0034 <sub>16</sub>	10 <sub>16</sub>
(30) MISRG	0038 <sub>16</sub>	00 <sub>16</sub>
(31) Watchdog timer control register	0039 <sub>16</sub>	0 0 1 1 1 1 1 1
(32) Interrupt edge selection register	003A <sub>16</sub>	00 <sub>16</sub>
(33) CPU mode register	003B <sub>16</sub>	1 0 0 0 0 0 0 0
(34) Interrupt request register 1	003C <sub>16</sub>	00 <sub>16</sub>
(35) Interrupt control register 1	003E <sub>16</sub>	00 <sub>16</sub>
(36) Processor status register	(PS)	X X X X X 1 X X
(37) Program counter	(PCH)	Contents of address FFFD <sub>16</sub>
	(PCL)	Contents of address FFFC <sub>16</sub>

**Note X** : Undefined

Fig. 41 Internal status of microcomputer at reset

### Clock Generating Circuit

An oscillation circuit can be formed by connecting a resonator between  $X_{IN}$  and  $X_{OUT}$ .

Use the circuit constants in accordance with the resonator manufacturer's recommended values. No external resistor is needed between  $X_{IN}$  and  $X_{OUT}$  since a feed-back resistor exists on-chip.

### ●Oscillation control

For the details of the state and release of stop mode and wait mode, refer to "Stop mode" and "Wait mode" of "FUNCTIONAL DESCRIPTION SUPPLEMENT".

#### • Stop mode

When the STP instruction is executed, the internal clock  $\phi$  stops at an "H" level and the  $X_{IN}$  oscillator stops. At this time, timer 1 is set to "0116" and prescaler 12 is set to "FF16" when the oscillation stabilization time set bit after release of the STP instruction is "0". On the other hand, timer 1 and prescaler 12 are not set when the above bit is "1". Accordingly, set the wait time fit for the oscillation stabilization time of the oscillator to be used.

$f(X_{IN})/16$  is forcibly connected to the input of prescaler 12.

When an external interrupt is accepted, oscillation is restarted but the internal clock  $\phi$  remains at "H" until timer 1 underflows. As soon as timer 1 underflows, the internal clock  $\phi$  is supplied. This is because when a ceramic oscillator is used, some time is required until a start of oscillation.

In case oscillation is restarted by reset, no wait time is generated. So apply an "L" level to the RESET pin while oscillation becomes stable.

#### • Wait mode

If the WIT instruction is executed, the internal clock  $\phi$  stops at an "H" level, but the oscillator does not stop. The internal clock  $\phi$  re starts if a reset occurs or when an interrupt is received.

Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

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

When the STP status is released, prescaler 12 and timer 1 will start counting with the  $f(X_{IN})/16$ , so set the timer 1 interrupt enable bit to "0" before the STP instruction is executed.

#### Note

For use with the oscillation stabilization time set bit after release of the STP instruction set to "1", set values in timer 1 and prescaler 12 after fully appreciating the oscillation stabilization time of the oscillator to be used.

#### • Clock mode

Operation is started by a built-in ring oscillator after releasing reset. A division ratio (1/1, 1/2, 1/8) is selected by setting bits 7 and 6 of the CPU mode register after releasing it.

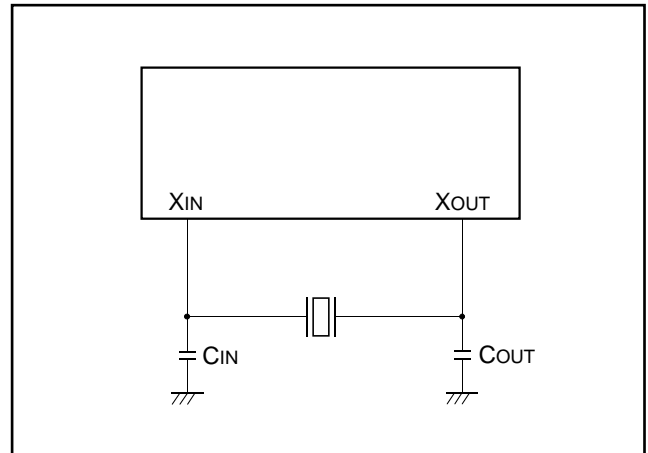


Fig. 42 External circuit of ceramic resonator

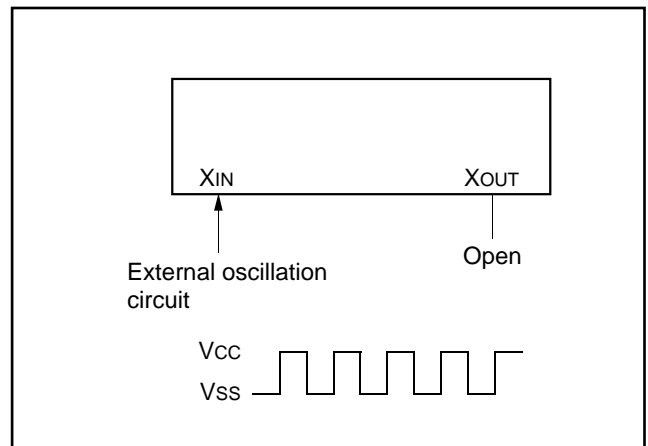


Fig. 43 External clock input circuit

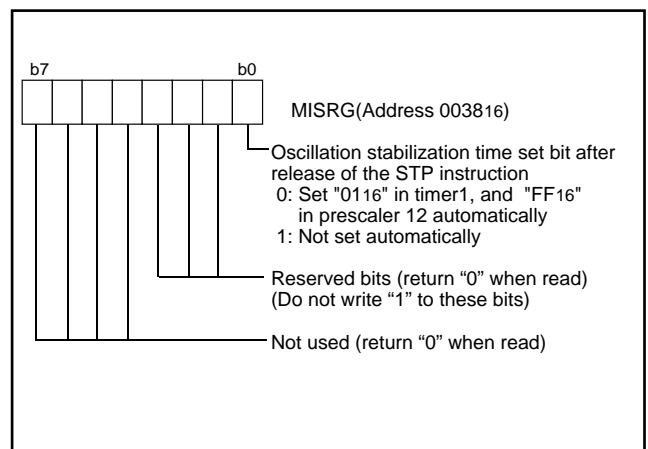


Fig. 44 Structure of MISRG

# HARDWARE

## FUNCTIONAL DESCRIPTION

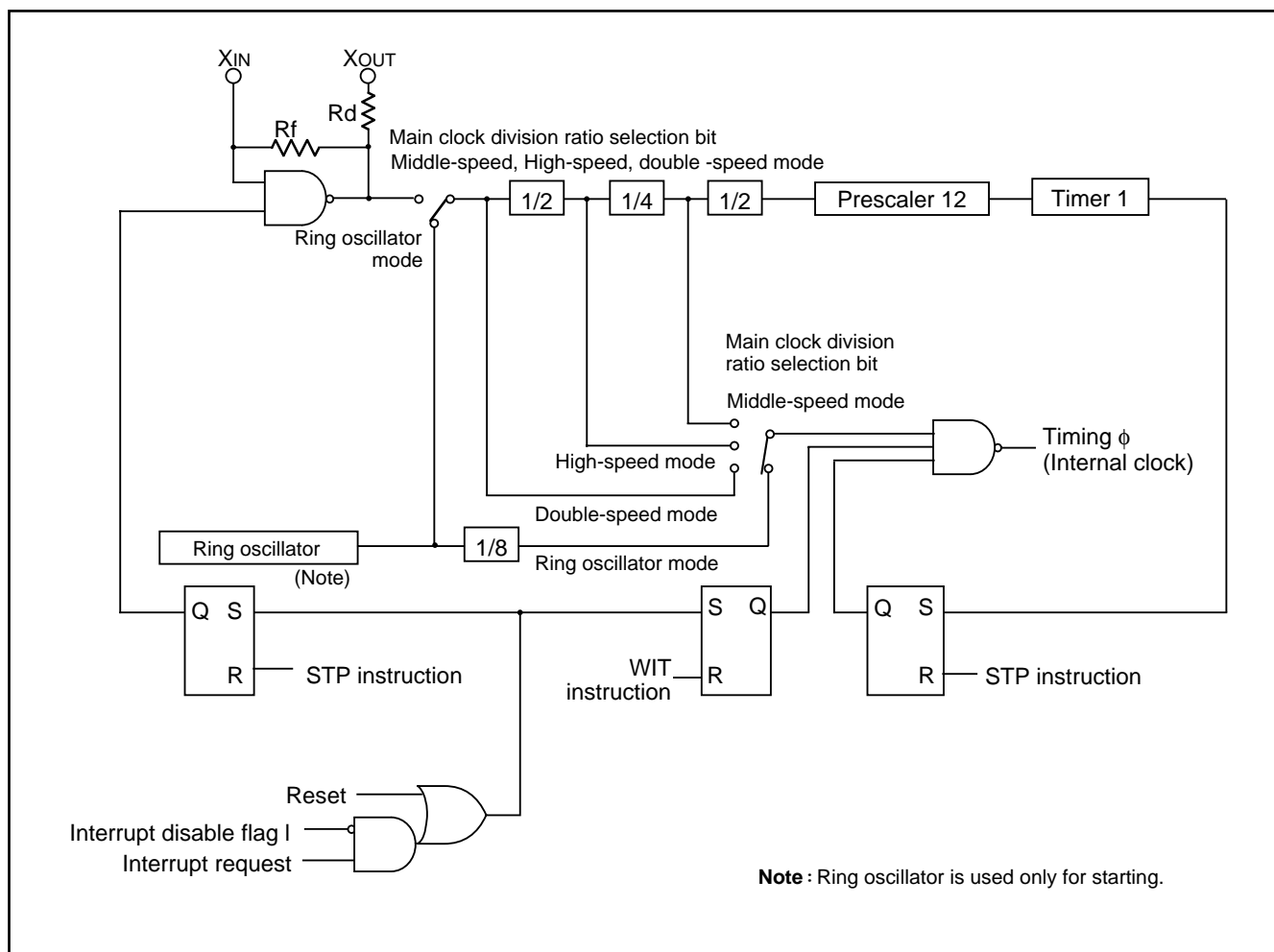


Fig. 45 Block diagram of system clock generating circuit (for ceramic resonator)

## NOTES ON PROGRAMMING

**Processor Status Register**

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

**Interrupts**

The contents of the interrupt request bit do not change even if the BBC or BBS instruction is executed immediately after they are changed by program because this instruction is executed for the previous contents. For executing the instruction for the changed contents, execute one instruction before executing the BBC or BBS instruction.

**Decimal Calculations**

- For calculations in decimal notation, set the decimal mode flag D to "1", then execute the ADC instruction or SBC instruction. In this case, execute SEC instruction, CLC instruction or CLD instruction after executing one instruction following the ADC instruction or SBC instruction.
- In the decimal mode, the values of the N (negative), V (overflow) and Z (zero) flags are invalid.

**Timers**

- When n (0 to 255) is written to a timer latch, the frequency division ratio is  $1/(n+1)$ .
- When a count source of timer X is switched, stop a count of timer X.

**Ports**

- The values of the port direction registers cannot be read.  
That is, it is impossible to use the LDA instruction, memory operation instruction when the T flag is "1", addressing mode using direction register values as qualifiers, and bit test instructions such as BBC and BBS.  
It is also impossible to use bit operation instructions such as CLB and SEB and read/modify/write instructions of direction registers for calculations such as ROR.  
For setting direction registers, use the LDM instruction, STA instruction, etc.
- Set "1" to each bit 6 of the port P3 direction register and the port P3 register (for only 7532 Group).

**A-D Converter**

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

Make sure that  $f(X_{IN})$  is 500kHz or more during A-D conversion.

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

**Instruction Execution Timing**

The instruction execution time can be obtained by multiplying the frequency of the internal clock  $\phi$  by the number of cycles mentioned in the machine-language instruction table.

The frequency of the internal clock  $\phi$  is the same as that of the  $X_{IN}$  in double-speed mode, twice the  $X_{IN}$  cycle in high-speed mode and 8 times the  $X_{IN}$  cycle in middle-speed mode.

## NOTES ON USE

**Handling of Power Source Pin**

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

**Handling of USBVREFOUT Pin**

In order to prevent the instability of the USBVREFOUT output due to external noise, connect a capacitor as bypass capacitor between USBVREFOUT pin and GND pin ( $V_{SS}$  pin). Besides, connect the capacitor to as close as possible. For bypass capacitor, a ceramic or electrolytic capacitor of 0.1  $\mu F$  is recommended.

**One Time PROM Version**

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

To improve the noise reduction, connect a track between CNVss pin and  $V_{SS}$  pin with 1 to 10 k $\Omega$  resistance.

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

# 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)

### ROM PROGRAMMING METHOD

The built-in PROM of the blank One Time PROM 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 8 Special programming adapter

Package	Name of Programming Adapter
42P4B	PCA7435SP

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 46 is recommended to verify programming.

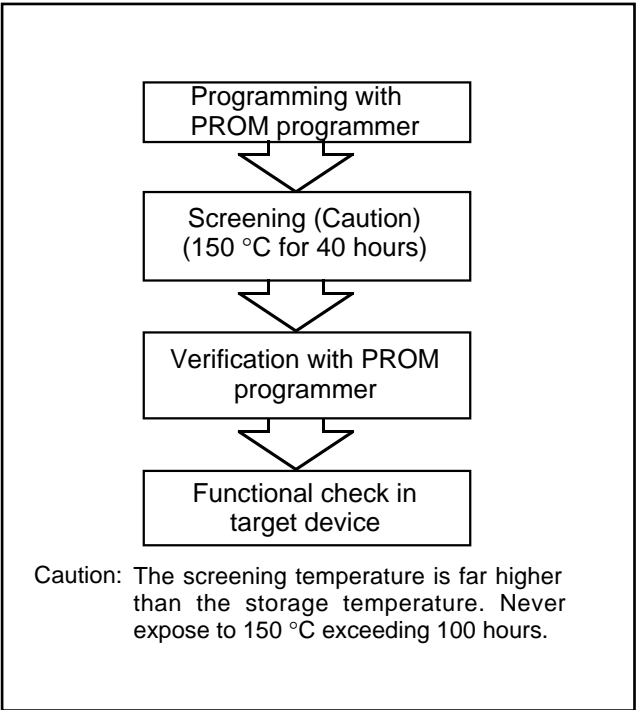


Fig. 46 Programming and testing of One Time PROM version

## FUNCTIONAL DESCRIPTION SUPPLEMENT

## Interrupt

7536 group permits interrupts on the basis of 14 sources (13 sources for 36-pin version of 7532 Group and 12 sources for 32-pin version of 7532 Group). 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 9.”

Table 9 Interrupt sources, vector addresses and interrupt priority

Interrupt source	Priority	Vector addresses (Note 1)		Interrupt request generating conditions	Remarks
		High-order	Low-order		
Reset (Note 2)	1	FFFD <sub>16</sub>	FFFC <sub>16</sub>	At reset input	Non-maskable
UART receive	2	FFFB <sub>16</sub>	FFFA <sub>16</sub>	At completion of UART data receive	Valid in UART mode
USB IN token				At detection of IN token	Valid in USB mode
UART transmit	3	FFF9 <sub>16</sub>	FFF8 <sub>16</sub>	At completion of UART transmit shift or when transmit buffer is empty	Valid in UART mode
USB SETUP/OUT token Reset/Suspend/Resume				At detection of SETUP/OUT token or At detection of Reset/ Suspend/ Resume	Valid in USB mode
INT <sub>1</sub>				At detection of either rising or falling edge of INT <sub>1</sub> input	External interrupt (active edge selectable)
INT <sub>0</sub>	4	FFF7 <sub>16</sub>	FFF6 <sub>16</sub>	At detection of either rising or falling edge of INT <sub>0</sub> input	External interrupt (active edge selectable)
Timer X	5	FFF5 <sub>16</sub>	FFF4 <sub>16</sub>	At timer X underflow	External interrupt (valid at falling)
Key-on wake-up				At falling of conjunction of input logical level for port P0 (at input)	
Timer 1	6	FFF3 <sub>16</sub>	FFF2 <sub>16</sub>	At timer 1 underflow	STP release timer underflow
Timer 2	7	FFF1 <sub>16</sub>	FFF0 <sub>16</sub>	At timer 2 underflow	
Serial I/O2				At completion of transmit/receive shift	
CNTR <sub>0</sub>	8	FFEF <sub>16</sub>	FFEE <sub>16</sub>	At detection of either rising or falling edge of CNTR <sub>0</sub> input	External interrupt (active edge selectable)
A-D conversion				At completion of A-D conversion	
BRK instruction	9	FFED <sub>16</sub>	FFEC <sub>16</sub>	At BRK instruction execution	Non-maskable software interrupt

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

**2:** Reset function 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 47 shows a timing chart after an interrupt occurs, and Figure 48 shows the time up to execution of the interrupt processing routine.

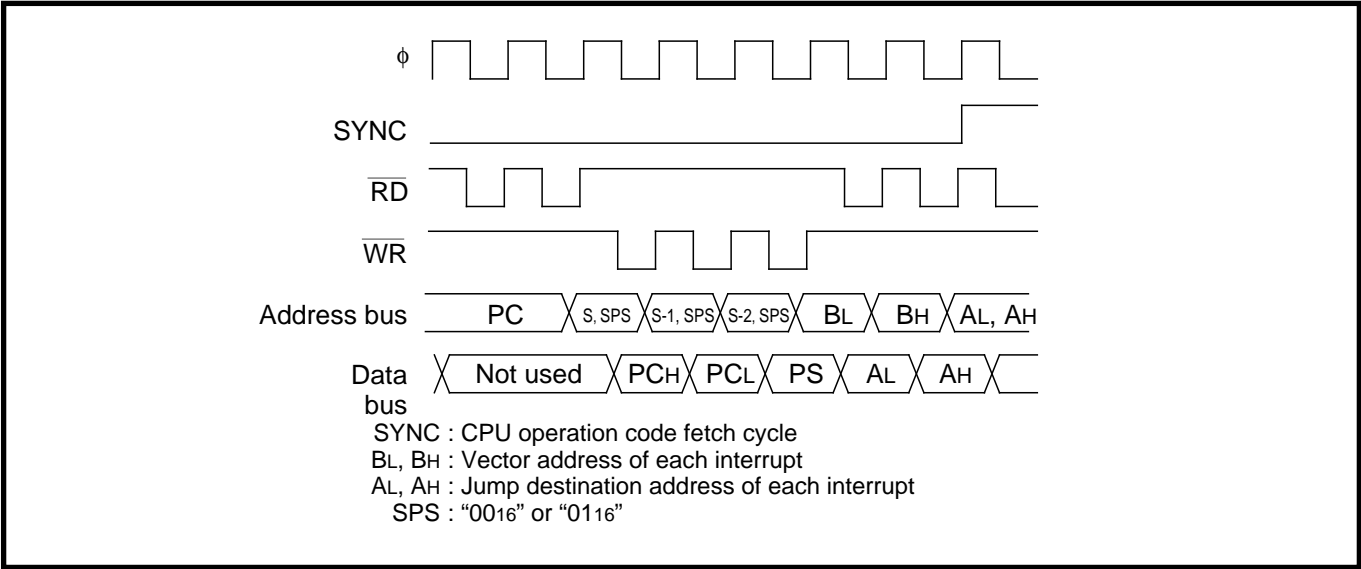


Fig. 47 Timing chart after an interrupt occurs

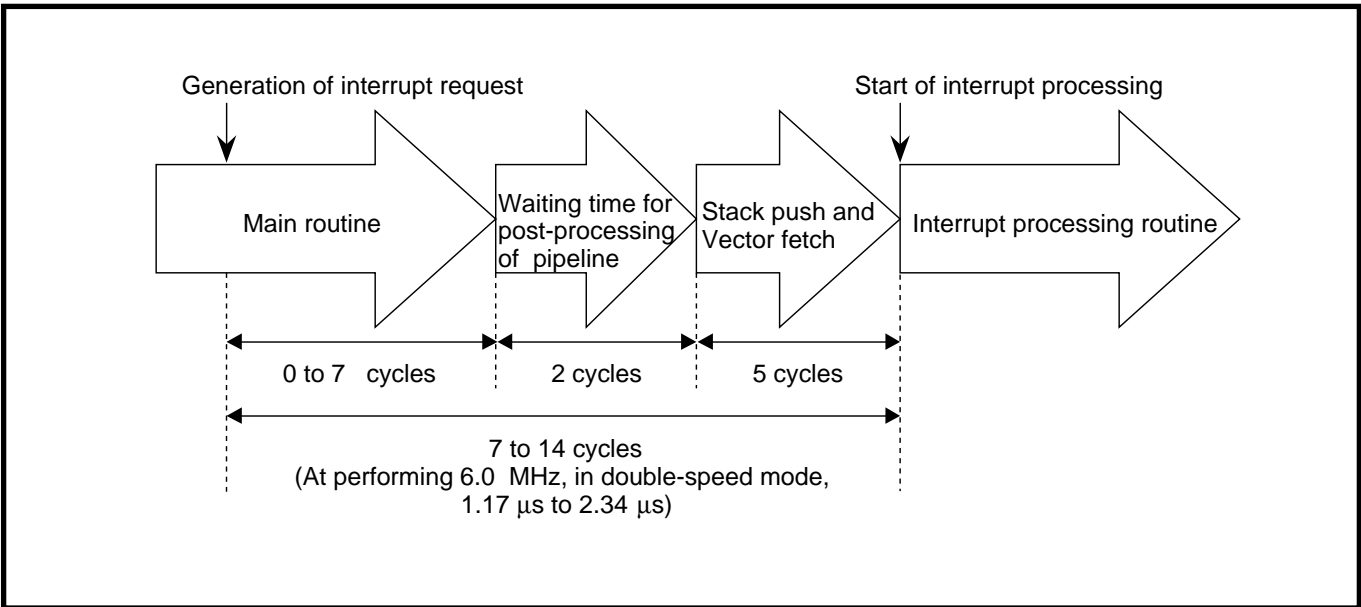


Fig. 48 Time up to execution of the interrupt processing routine

### 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  $V_{ref}$  is input to the comparator. Then,  $V_{ref}$  is compared with analog input voltage  $V_{IN}$ .
3. As a result of comparison, when  $V_{ref} < V_{IN}$ , the highest-order bit of A-D conversion register becomes “1.” When  $V_{ref} > V_{IN}$ , 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 122 clock cycles (20.34  $\mu$ s at  $f(X_{IN}) = 6.0$  MHz) after it is started, and the result of the conversion is stored into the A-D conversion register.

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

#### Relative formula for a reference voltage $V_{REF}$ of A-D converter and $V_{ref}$

When  $n = 0$

$V_{ref} = 0$

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

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

**Table 10 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	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0			
First comparison	<table><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	1	0	0	0	0	0	0	0	0	0	$\frac{V_{REF}}{2}$
1	0	0	0	0	0	0	0	0	0			
Second comparison	<table><tr><td>* 1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	* 1	1	0	0	0	0	0	0	0	0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4}$
* 1	1	0	0	0	0	0	0	0	0			
Third comparison	<table><tr><td>* 1</td><td>* 2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	* 1	* 2	1	0	0	0	0	0	0	0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8}$
* 1	* 2	1	0	0	0	0	0	0	0			
⋮	⋮	⋮										
After completion of tenth comparison	<div>A result of A-D conversion</div> <table><tr><td>* 1</td><td>* 2</td><td>* 3</td><td>* 4</td><td>* 5</td><td>* 6</td><td>* 7</td><td>* 8</td><td>* 9</td><td>* 10</td></tr></table>	* 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	* 2	* 3	* 4	* 5	* 6	* 7	* 8	* 9	* 10			

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



# HARDWARE

## FUNCTIONAL DESCRIPTION SUPPLEMENT

Figures 49 shows A-D conversion equivalent circuit, and Figure 50 shows A-D conversion timing chart.

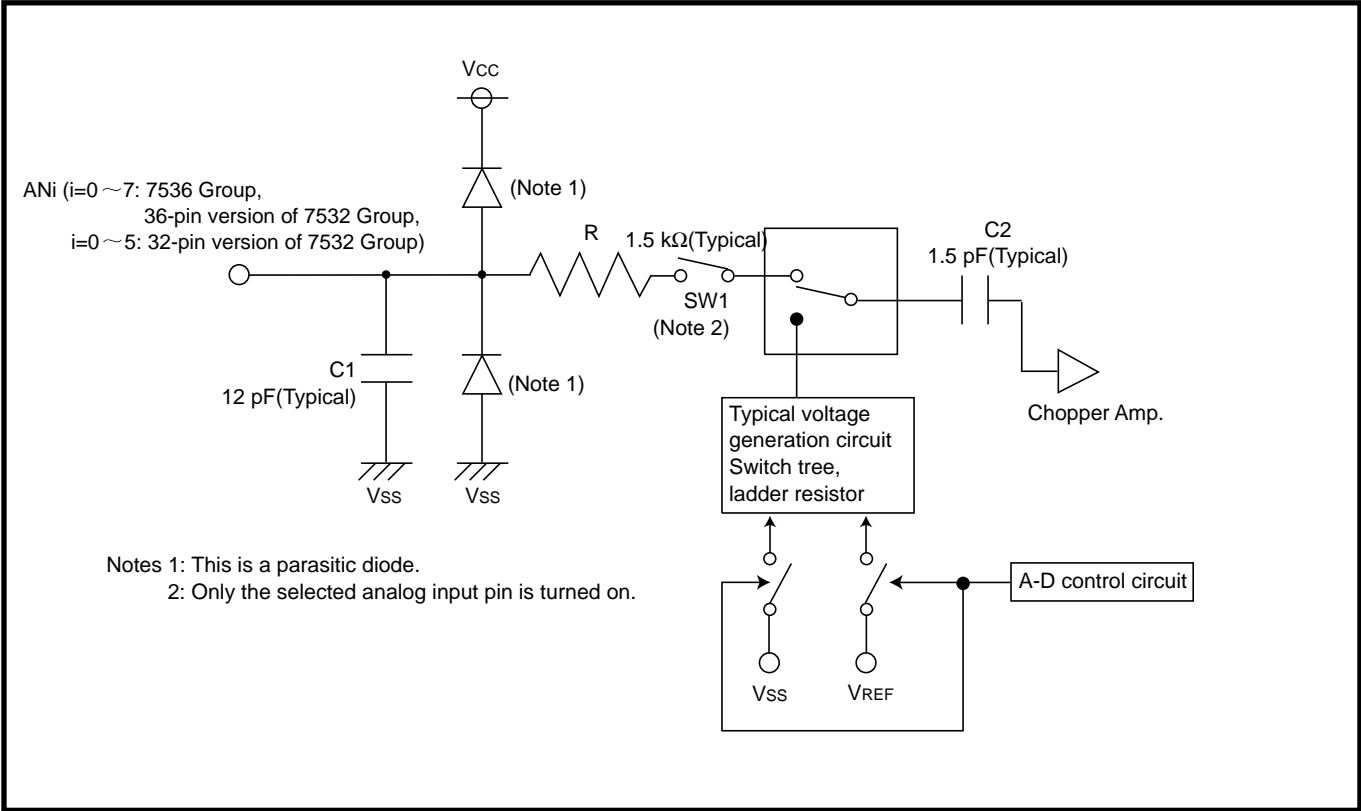


Fig. 49 A-D conversion equivalent circuit

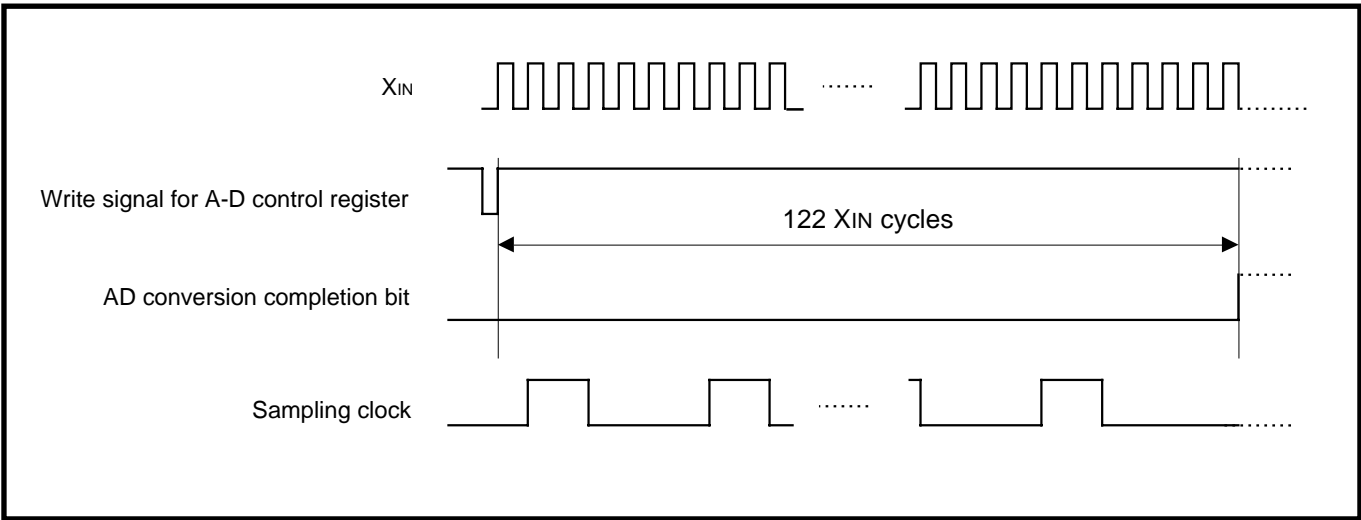


Fig. 50 A-D conversion timing chart

### Stop mode

System enters the stop mode by executing the **STP** instruction. In the stop mode, the  $f(XIN)$  oscillation is stopped, and the internal clock  $\phi$  is stopped. Accordingly, CPU and the peripheral devices are stopped.

#### (1) Stop mode state

Table 11 shows the state at stop mode.

**Table 11 Stop mode state**

Parameter	State	Parameter	State
Oscillation	Stop	Watchdog timer	Stop
CPU	Stop	RAM	State retained
I/O port	State retained at <b>STP</b> instruction execution	SFR	State retained (Timer 1 and prescaler 12 excepted)
Timer	When selecting internal count source: Stop When selecting external count source: Operating (only Timer X)	CPU register	State retained <ul style="list-style-type: none"> <li>• Accumulator</li> <li>• Index register X</li> <li>• Index register Y</li> <li>• Stack pointer</li> <li>• Program counter</li> <li>• Processor status register</li> </ul>
UART	Stop		
A-D conversion	Stop		
Serial I/O2	When selecting internal synchronous clock: Stop When selecting external synchronous clock: Operating		
USB	Stop (suspend state)		

#### (2) Stop mode release

Stop mode is released by reset input or interrupt occurrence. The interrupt sources which can be used for return from stop mode are shown below.

- INT0
- INT1
- CNTR0
- Timer (Timer X) when using external clock
- Serial I/O2 when using external clock
- Key-on wakeup
- USB function (resume)

When the above interrupt sources are used for return from stop mode, execute the **STP** instruction after the following are set in order to enable the using interrupts.

- ① Clear the timer 1 interrupt enable bit to "0" (ICON1, bit 4)
- ② Clear the timer 2 interrupt enable bit to "0" (ICON1, bit 5)
- ③ Clear the timer 1 interrupt request bit to "0" (IREQ1, bit 4)
- ④ Clear the timer 2 interrupt request bit to "0" (IREQ1, bit 5)
- ⑤ Clear the interrupt request bit of the interrupt using for return to "0"
- ⑥ Set the interrupt enable bit of the interrupt using for return to "1"
- ⑦ Clear the interrupt disable flag (I) to "0"

# HARDWARE

## FUNCTIONAL DESCRIPTION SUPPLEMENT

### Wait mode

System enters the wait mode by executing the **WIT** instruction. In the wait mode, the oscillation is operating, but the internal clock  $\phi$  is stopped. Accordingly, CPU is stopped, but the peripheral devices are operating.

#### (1) Wait mode state

Table 12 shows the state at wait mode.

**Table 12 Wait mode state**

Parameter	State
Oscillation	Stop
CPU	Stop
I/O port	State at <b>WIT</b> instruction execution retained
Timer	When selecting internal count source: Operating When selecting external count source: Operating
UART	Operating
A-D conversion	Operating (Conversion is continued if the <b>WIT</b> instruction is executed during conversion)
Serial I/O2	When selecting internal synchronous clock: Operating When selecting external synchronous clock: Operating
USB	Operating

Parameter	State
Watchdog timer	Operating
RAM	State retained
SFR	State retained (Timer 1, timer 2 and prescaler 12 excepted)
CPU register	State retained <ul style="list-style-type: none"><li>• Accumulator</li><li>• Index register X</li><li>• Index register Y</li><li>• Stack pointer</li><li>• Program counter</li><li>• Processor status register</li></ul>

#### (2) Wait mode release

Wait mode is released by reset input or interrupt occurrence. In the wait mode, since the oscillation is continued, the instruction is executed after the system is released from the wait mode. The interrupt sources which can be used for return from wait mode are shown below.

- INT0
- INT1
- CNTR0
- Timer
- Serial I/O2
- A-D conversion
- Key-on wakeup
- USB function
- UART

When the above interrupt sources are used for return from wait mode, execute the **WIT** instruction after the following are set in order to enable the using interrupts.

- ① Clear the interrupt request bit of the interrupt using for return to "0"
- ② Set the interrupt enable bit of the interrupt using for return to "1"
- ③ Clear the interrupt disable flag (I) to "0"

## DIFFERENCES BETWEEN 7532 GROUP AND 7536 GROUP

The 7532 Group has less I/O ports than 7536 Group. Accordingly, when the pins which have the function except a port function are eliminated, be careful that the functions are also eliminated. Also, in the 7532 Group, be careful that the 36-pin SSOP package version differs from the 32-pin LQFP package version for the eliminated functions.

**Table 13 Differences between 7532 Group and 7536 Group**

I/O port	7536 Group (42-pin SDIP)	7532 Group (36-pin SSOP)	7532 Group (32-pin LQFP)
Port P1	P10–P16 (7-bit structure)	P10–P14 (5-bit structure)	P10–P14 (5-bit structure)
Port P2	P20–P27 (8-bit structure) (A-D converter 8-channel)	P20–P27 (8-bit structure) (A-D converter 8-channel)	P20–P25 (6-bit structure) (A-D converter 6-channel)
Port P3	P30–P37 (8-bit structure) (INT <sub>0</sub> , INT <sub>1</sub> available)	P30–P35, P37 (7-bit structure) (INT <sub>0</sub> available)	P30–P34 (5-bit structure) (INT function not available)
Port P4	P40, P41 (2-bit structure)	No port	No port

# HARDWARE

## DIFFERENCES BETWEEN 7532 GROUP AND 7536 GROUP

Additionally, there are differences of SFR usage and functional definitions.

**Table 14 Differences between 7532 Group and 7536 Group (SFR)**

Register (Address)	7536 Group (42-pin SDIP)	7532 Group (36-pin SSOP)	7532 Group (32-pin LQFP)
Port P1/Direction (0216/0316)	Bit 7 not available	Bits 5 to 7 not available	Bits 5 to 7 not available
Port P2/Direction (0416/0516)	All bits available	All bits available	Bits 6 and 7 not available
Port P3/Direction (0616/0716)	All bits available	Bit 6 not available	Bits 5 to 7 not available
Port P4/Direction (0816/0916)	Bits 2 to 7 not available	All bits not available	All bits not available
Pull-up control (1616)	Bit 6 definition: "P35, P36 pull-up control" Bit 7 definition: "P37 pull-up control"	Bit 6 definition: "P35 pull-up control" Bit 7 definition: "P37 pull-up control"	Bits 6 and 7 not available
Port P1P3 control (1716)	Bit 0 definition: "P37/INT0 input level selection" Bit 1 definition: "P36/INT1 input level selection"	Bit 0 definition: "P37/INT0 input level selection" Bit 1 not available	Bits 0 and 1 not available
A-DControl (3416)	Bits 0 to 2 "Input pins selected by setting these bits to 000 to 111"	Bits 0 to 2 "Input pins selected by setting these bits to 000 to 111"	Bits 0 to 2 "Input pins selected by setting these bits to 000 to 101"
Interrupt edge selection (3A16)	Bit 0 definition "INT0 interrupt edge selection" Bit 1 definition "INT1 interrupt edge selection" Bit 4 definition "Serial I/O1, INT1 interrupt selection"	Bit 0 definition "INT0 interrupt edge selection" Bits 1 and 4 not available	Bits 0, 1 and 4 not available
Interrupt request (3C16)	Bit 1 definition "UART transmission, USB (except IN), INT1" Bit 2 definition "INT0"	Bit 1 definition "UART transmission, USB (except IN)" Bit 2 definition "INT0"	Bit 1 definition "UART transmission, USB (except IN)" Bit 2 not available
Interrupt control (3E16)	Bit 1 definition "UART transmission, USB (except IN), INT1" Bit 2 definition "INT0"	Bit 1 definition "UART transmission, USB (except IN)" Bit 2 definition "INT0"	Bit 1 definition "UART transmission, USB (except IN)" Bit 2 not available

## DIFFERENCES BETWEEN 7532 GROUP AND 7536 GROUP

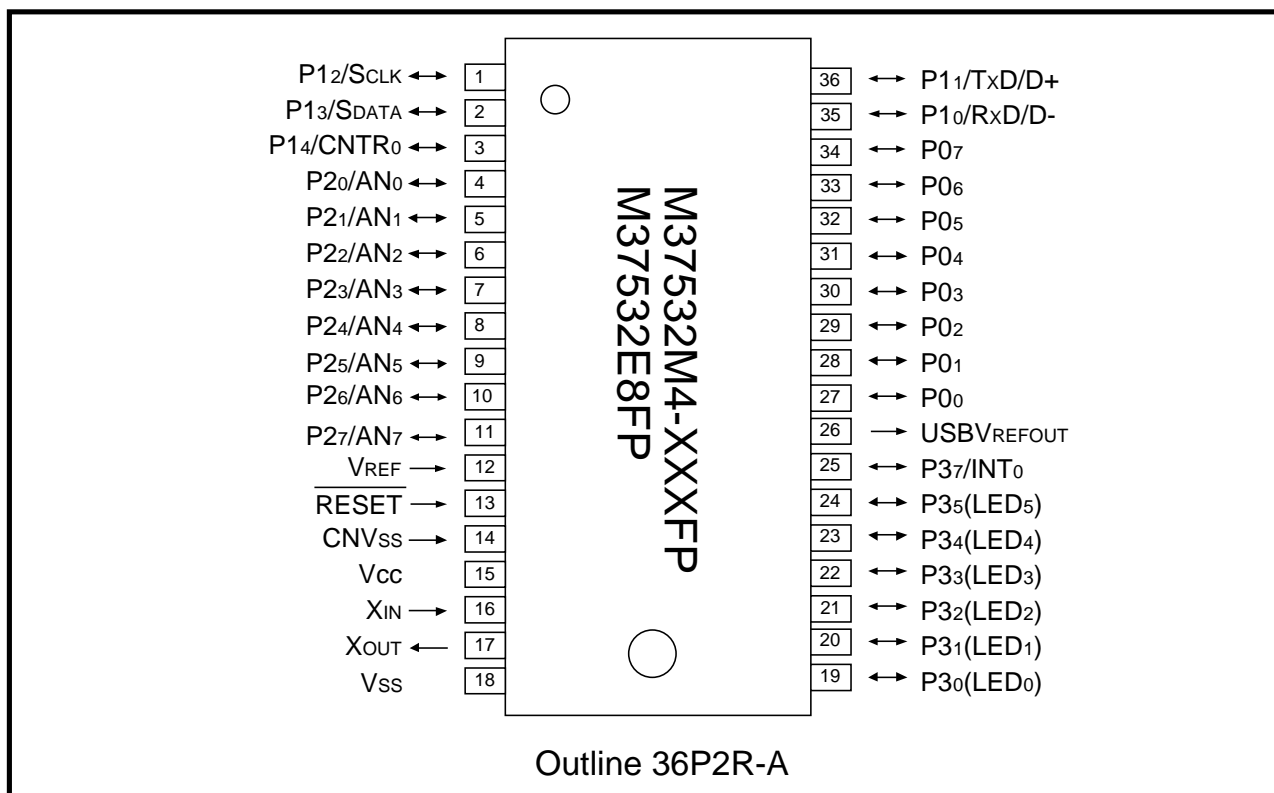


Fig. 51 Pin configuration of M37532M4-XXXFP, M37532E8FP

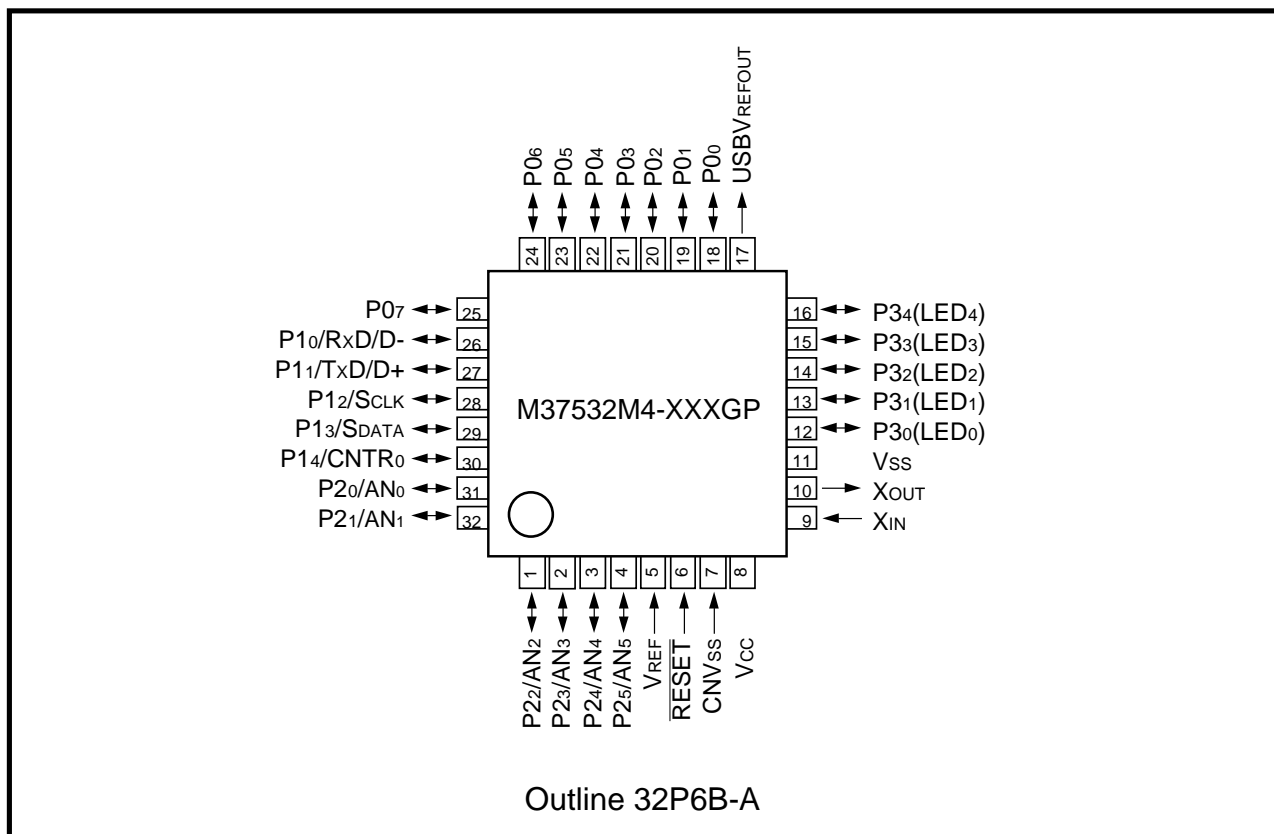


Fig. 52 Pin configuration of M37532M4-XXXGP

# HARDWARE

## DIFFERENCES BETWEEN 7532 GROUP AND 7536 GROUP

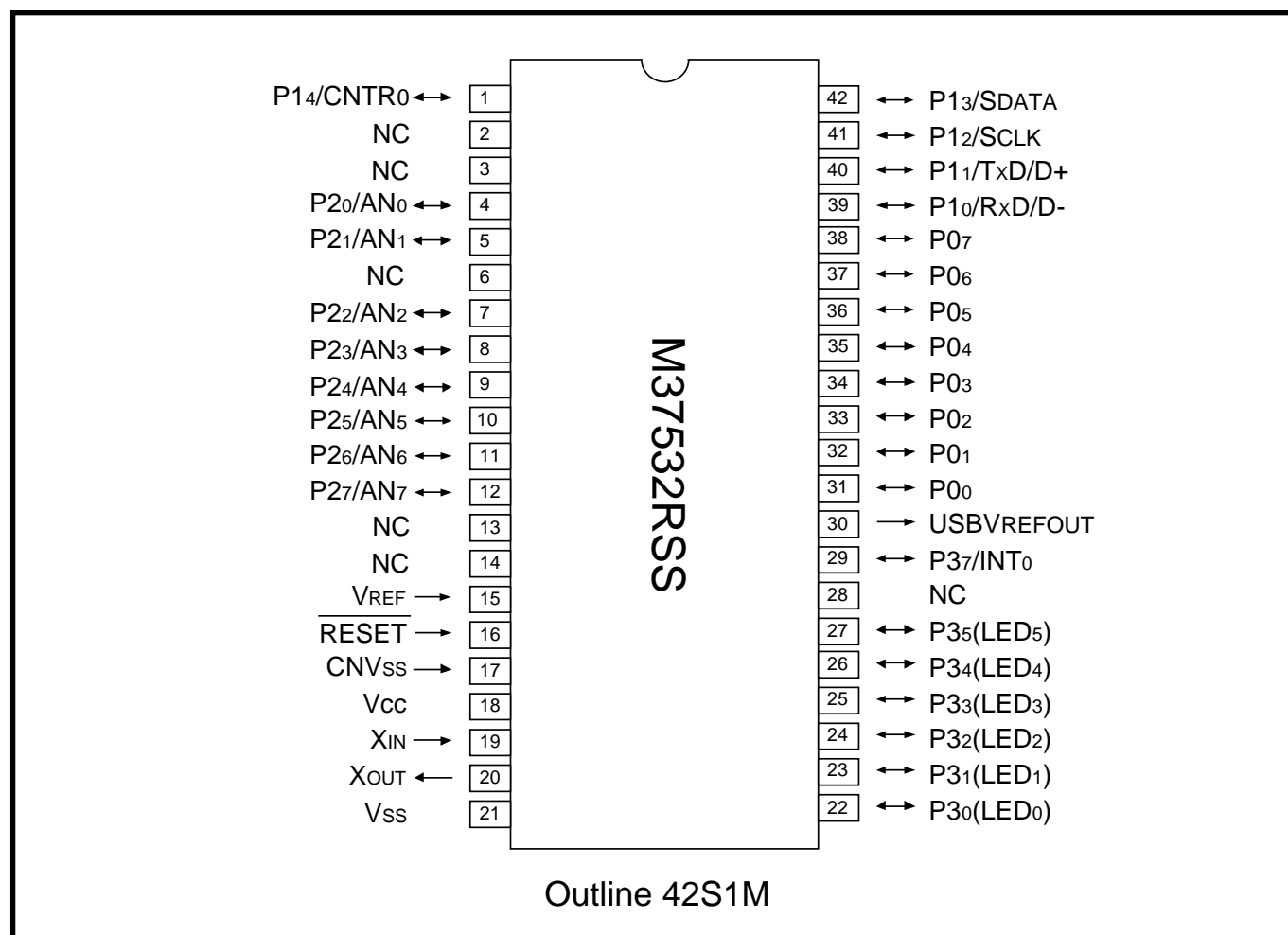
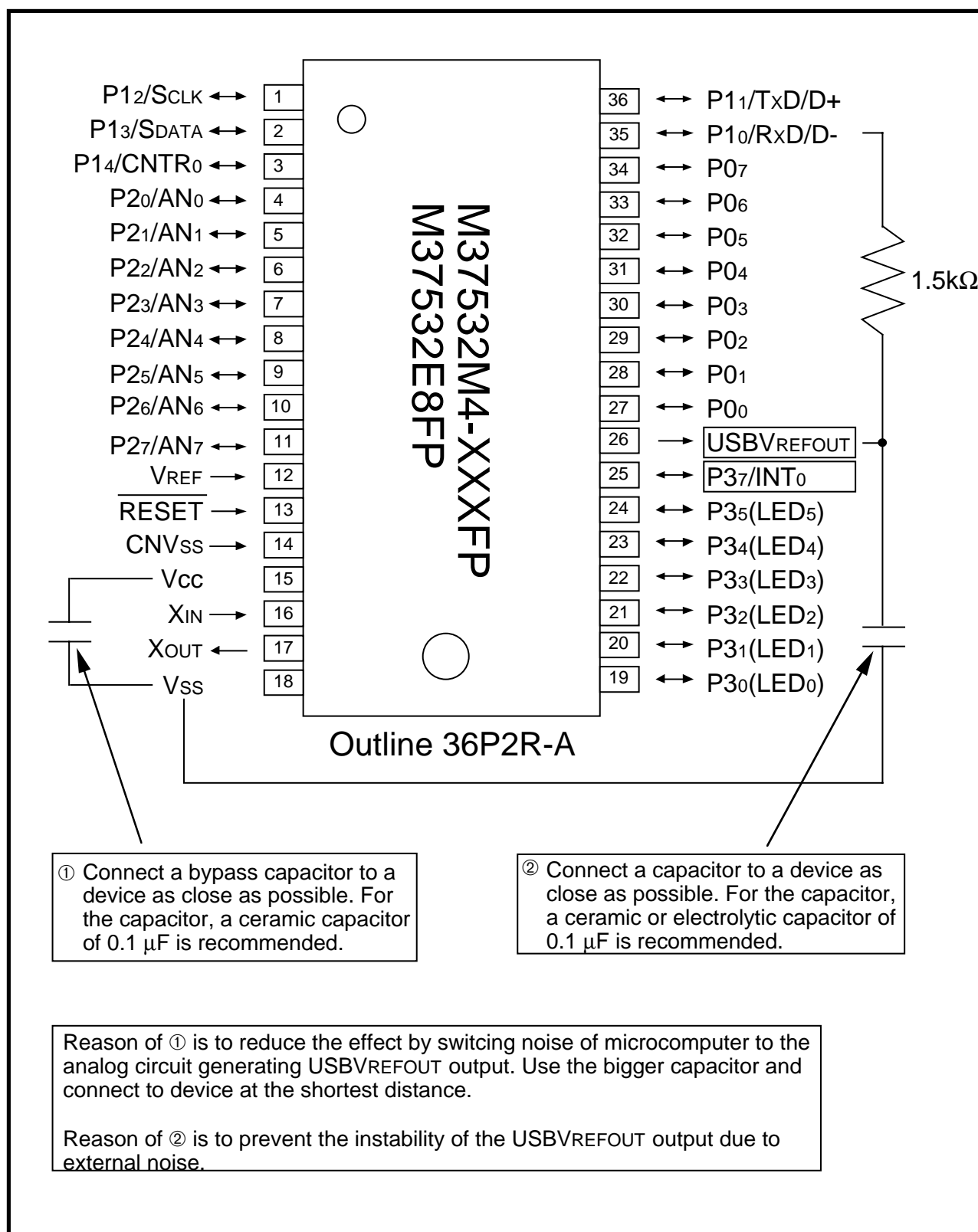


Fig. 53 Pin configuration of M37532RSS

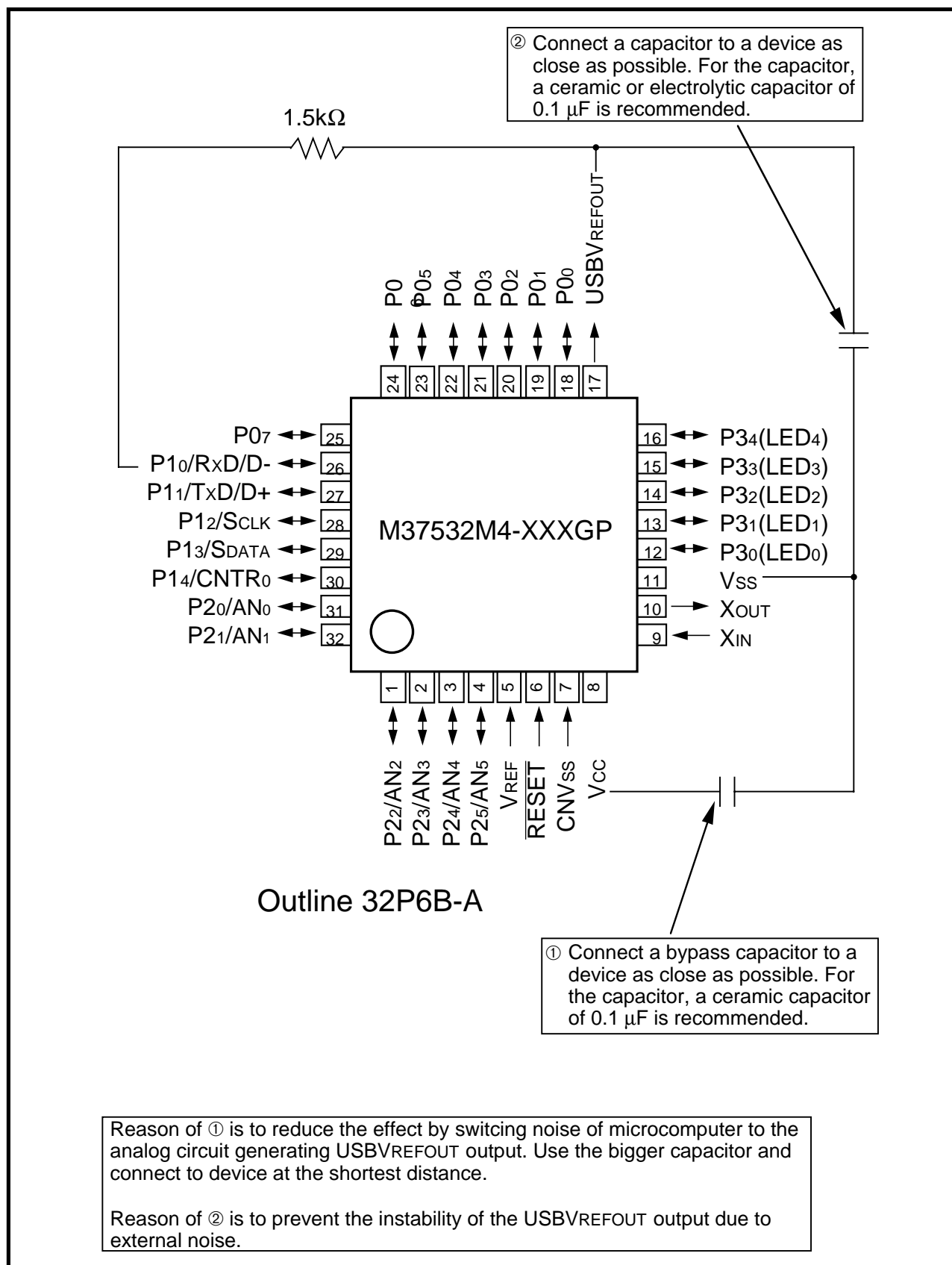
## DESCRIPTION SUPPLEMENT FOR USE OF USB FUNCTION STABLY

Fig. 54 Handling of V<sub>CC</sub>, USBV<sub>REFOUT</sub> pins of M37532M4-XXXXFP, M37532E8FP

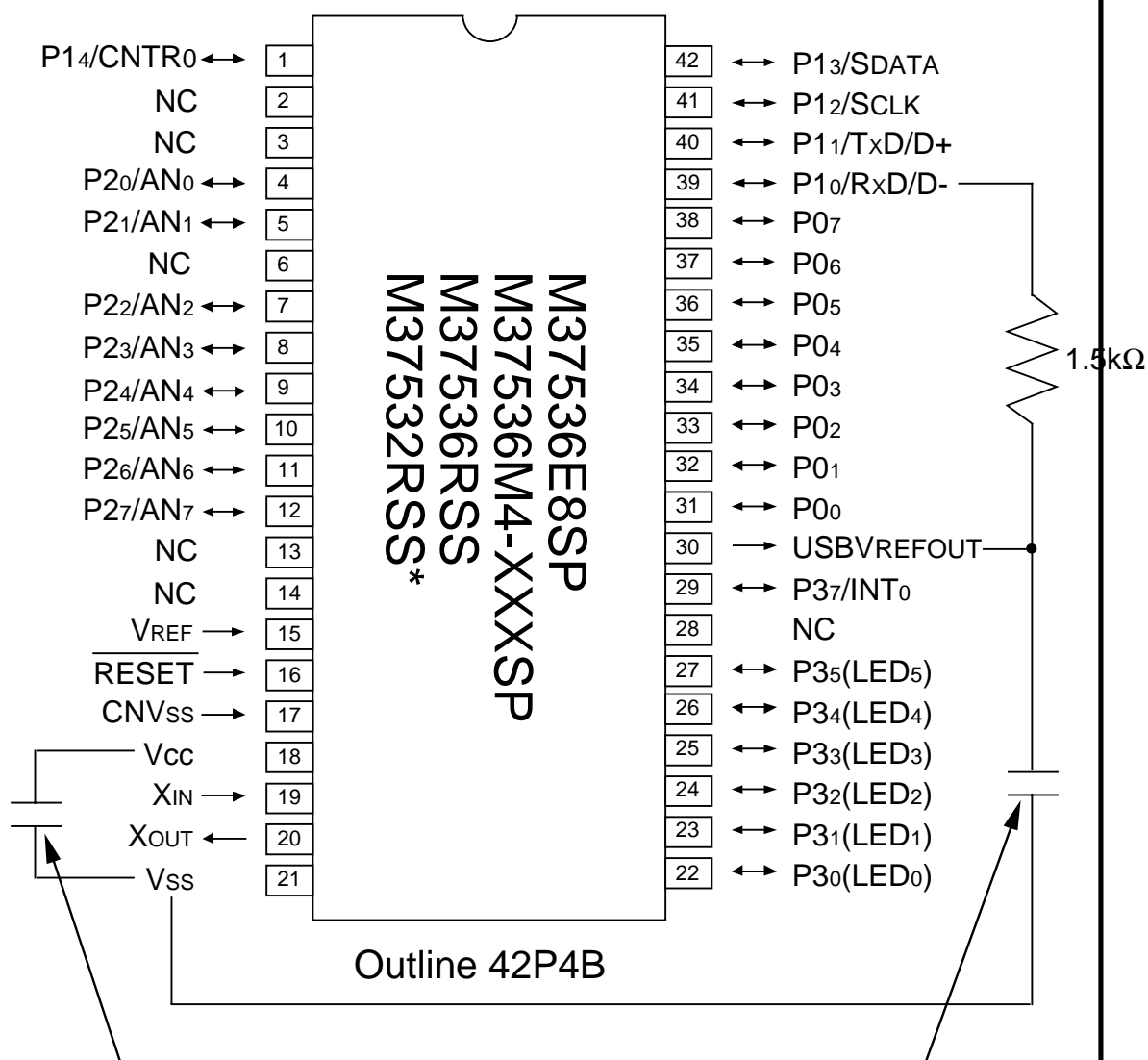


# HARDWARE

## DESCRIPTION SUPPLEMENT FOR USE OF USB FUNCTION STABLY



**Fig. 55 Handling of VCC, USBVREFOUT pins of M37532M4-XXXGP**



① Connect a bypass capacitor to a device as close as possible. For the capacitor, a ceramic capacitor of 0.1  $\mu\text{F}$  is recommended.

② Connect a capacitor to a device as close as possible. For the capacitor, a ceramic or electrolytic capacitor of 0.1  $\mu\text{F}$  is recommended.

Reason of ① is to reduce the effect by switching noise of microcomputer to the analog circuit generating USBVREFOUT output. Use the bigger capacitor and connect to device at the shortest distance.

Reason of ② is to prevent the instability of the USBVREFOUT output due to external noise.

\*: For the M37532RSS, the following pins are the Not Connected pins.  
Pin number: 2, 3, 13, 14, 28

Fig. 56 Handling of VCC, USBVREFOUT pins of M37536E8SP, M37536M4-XXXSP, M37536RSS, M37532RSS

# HARDWARE

## DESCRIPTION SUPPLEMENT FOR USE OF USB FUNCTION STABLY

### Relationship between USB output singal (D+, D- signal) and cable load

Description supplement is added to the table shown below for the UNIVERSAL SERIAL BUS SPECIFICATION VERSION 1.0\* Chapter 7 (Table 7-6).

For detail, the limits of cable load condition CL in the table shown below is set,

Though "CL = concentrated constant" is the precondition for the AC specification in the table below, the followings are checked as the standard.

- "CL = 150 pF" corresponds the untwisted 50 cm USB cable (general thin cable) **(Note)**
- "CL = 350 pF" corresponds the untwisted 150 cm USB cable (general thin cable) **(Note)**

**Note:** They are valiable according to the material and structure of the cable.

\*: Proposal for Universal Serial Bus Specification

Copyright (C) 1996, Compaq Computer Corporation, Digital Equipment Corporation,  
IBM PC Company, Intel Corporation, Microsoft Corporation, NEC, Northern Telecom.  
All rights reserved

**Table 15 Precondition:  $V_{CC}= 4.4\text{ V to }5.25\text{ V}$ ,  $T_a = 0\text{ }^{\circ}\text{C to }70\text{ }^{\circ}\text{C}$**

Parameter	Symbol	USB specification			7532/7536			Unit
		Condition	Min.	Max.	Condition	Min.	Max.	
Driver characteristics	TR	CL = 50 pF	75		CL = 50 pF	75		ns
		CL = 350 pF		300	CL = 350 pF		300	ns
	TF	CL = 50 pF	75		CL = 50 pF	75		ns
		CL = 350 pF		300	CL = 350 pF		300	ns
Rising/Falling time match	TRFM	TR/TF	80	120	TR/TF, CL = 150 to 350 pF	80	120	%
Output singal crossover voltage	VCRS		1.3	2.0	CL = 150 to 350 pF	1.3	2.0	V



# CHAPTER 2

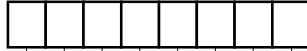
## **APPLICATION**

- 2.1 I/O port
- 2.2 Timer
- 2.3 Serial I/O
- 2.4 USB
- 2.5 A-D converter
- 2.6 Reset



Port Pi direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi direction register (PiD) (i = 0 to 4) [Address : 01<sub>16</sub>, 03<sub>16</sub>, 05<sub>16</sub>, 07<sub>16</sub>, 09<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Port Pi direction register	0 : Port Pi <sub>0</sub> input mode 1 : Port Pi <sub>0</sub> output mode	0	X	○
1		0 : Port Pi <sub>1</sub> input mode 1 : Port Pi <sub>1</sub> output mode	0	X	○
2		0 : Port Pi <sub>2</sub> input mode 1 : Port Pi <sub>2</sub> output mode	0	X	○
3		0 : Port Pi <sub>3</sub> input mode 1 : Port Pi <sub>3</sub> output mode	0	X	○
4		0 : Port Pi <sub>4</sub> input mode 1 : Port Pi <sub>4</sub> output mode	0	X	○
5		0 : Port Pi <sub>5</sub> input mode 1 : Port Pi <sub>5</sub> output mode	0	X	○
6		0 : Port Pi <sub>6</sub> input mode 1 : Port Pi <sub>6</sub> output mode	0	X	○
7		0 : Port Pi <sub>7</sub> input mode 1 : Port Pi <sub>7</sub> output mode	0	X	○

**Note:** The following ports do not exist, so that the corresponding bits are not used.

- 7536 Group: Ports P1<sub>7</sub>, P4<sub>2</sub>–P4<sub>7</sub>
- 7532 Group, 36-pin version: Ports P1<sub>5</sub>–P1<sub>7</sub>, P3<sub>6</sub>, P4<sub>0</sub>–P4<sub>7</sub>
- 7532 Group, 32-pin version: Ports P1<sub>5</sub>–P1<sub>7</sub>, P2<sub>6</sub>, P3<sub>5</sub>–P3<sub>7</sub>, P4<sub>0</sub>–P4<sub>7</sub>

**Fig. 2.1.3 Structure of Port Pi direction register (i = 0 to 4)**Pull-up control register

b7 b6 b5 b4 b3 b2 b1 b0

Pull-up control register (PULL) [Address : 16<sub>16</sub>]

B	Name	Function	At reset	R	W
0	P0 <sub>0</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
1	P0 <sub>1</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
2	P0 <sub>2</sub> , P0 <sub>3</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
3	P0 <sub>4</sub> – P0 <sub>7</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
4	P3 <sub>0</sub> – P3 <sub>3</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
5	P3 <sub>4</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
6	P3 <sub>5</sub> , P3 <sub>6</sub> pull-up control bit (Note 2)	0 : Pull-up Off 1 : Pull-up On	1	○	○
7	P3 <sub>7</sub> pull-up control bit (Note 3)	0 : Pull-up Off 1 : Pull-up On	1	○	○

**Notes 1:** Pins set to output are disconnected from the pull-up control.**2:** • 7532 Group, 36-pin version: P3<sub>6</sub> is not existed.

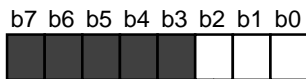
• 7532 Group, 32-pin version: Not used.

**3:** 7532 Group, 32-pin version: Not used.**Fig. 2.1.4 Structure of Pull-up control register**

# APPLICATION

## 2.1 I/O port

### Port P1P3 control register



Port P1P3 control register (P1P3C) [Address : 17<sub>16</sub>]

B	Name	Function	At reset	R	W
0	P37/INT <sub>0</sub> input level selection bit ( <b>Note 1</b> )	0 : CMOS level 1 : TTL level	0	○	○
1	P36/INT <sub>1</sub> input level selection bit ( <b>Note 2</b> )	0 : CMOS level 1 : TTL level	0	○	○
2	P1 <sub>0</sub> , P1 <sub>2</sub> , P1 <sub>3</sub> input level selection bit	0 : CMOS level 1 : TTL level	0	○	○
3	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

**Notes 1:** For the 32-pin version of 7532 Group, nothing is allocated for this bit.

This is a write disabled bit.

When this bit is read out, the value is "0".

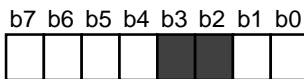
**2:** For the 32-pin and 36-pin versions of 7532 Group, nothing is allocated for this bit.

This is a write disabled bit.

When this bit is read out, the value is "0".

Fig. 2.1.5 Structure of P1P3 control register

### Interrupt edge selection register



Interrupt edge selection register (INTEDGE) [Address : 3A<sub>16</sub>]

B	Name	Function	At reset	R	W
0	INT <sub>0</sub> interrupt edge selection bit ( <b>Note 1</b> )	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT <sub>1</sub> interrupt edge selection bit ( <b>Note 2</b> )	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
3			0	○	×
4	Serial I/O1 or INT <sub>1</sub> interrupt selection bit	0 : Serial I/O1 1 : INT <sub>1</sub>	0	○	○
5	Timer X or key-on wake up interrupt selection bit	0 : Timer X 1 : Key-on wake up	0	○	○
6	Timer 2 or serial I/O2 interrupt selection bit	0 : Timer 2 1 : Serial I/O2	0	○	○
7	CNTR <sub>0</sub> or AD converter interrupt selection bit	0 : CNTR <sub>0</sub> 1 : AD converter	0	○	○

**Notes 1:** 7532 Group, 32-pin version: Not used.

This is a write disabled bit. When this bit is read out, the value is "0".

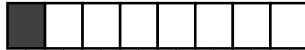
**2:** 7532 Group, 36-pin and 32-pin version: Not used.

This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.1.6 Structure of Interrupt edge selection register

## Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : 3C<sub>16</sub>]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
1	UART transmit/USBSETUP/OUT token/Reset/Suspend/Resume/INT <sub>1</sub> interrupt request bit ( <b>Note 1</b> )	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
2	INT <sub>0</sub> interrupt request bit ( <b>Note 2</b> )	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
3	Timer X or key-on wake up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
4	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
5	Timer 2 or serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
6	CNTR <sub>0</sub> or AD converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

\*: These bits can be cleared to "0" by program, but cannot be set.

**Notes 1:** 7532 Group, 36-pin version and 32-pin version: INT<sub>1</sub> interrupt does not exist.**2:** 7532 Group, 32-pin version: INT<sub>0</sub> interrupt does not exist.

This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.1.7 Structure of Interrupt request register 1

## Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1 (ICON1) [Address : 3E<sub>16</sub>]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	UART transmit/USBSETUP/OUT token/Reset/Suspend/Resume/INT <sub>1</sub> interrupt enable bit ( <b>Note 1</b> )	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT <sub>0</sub> interrupt enable bit ( <b>Note 2</b> )	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Timer X or key-on wake up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer 2 or serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	CNTR <sub>0</sub> or AD converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Nothing is allocated for this bit. Do not write "1" to this bit. When this bit is read out, the value is "0".		0	○	×

**Notes 1:** 7532 Group, 36-pin version and 32-pin version: INT<sub>1</sub> interrupt does not exist.**2:** 7532 Group, 32-pin version: INT<sub>0</sub> interrupt does not exist.

This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.1.8 Structure of Interrupt control register 1



# APPLICATION

## 2.1 I/O port

### 2.1.3 Application example of key-on wake up

**Outline:** The built-in pull-up resistor is used.

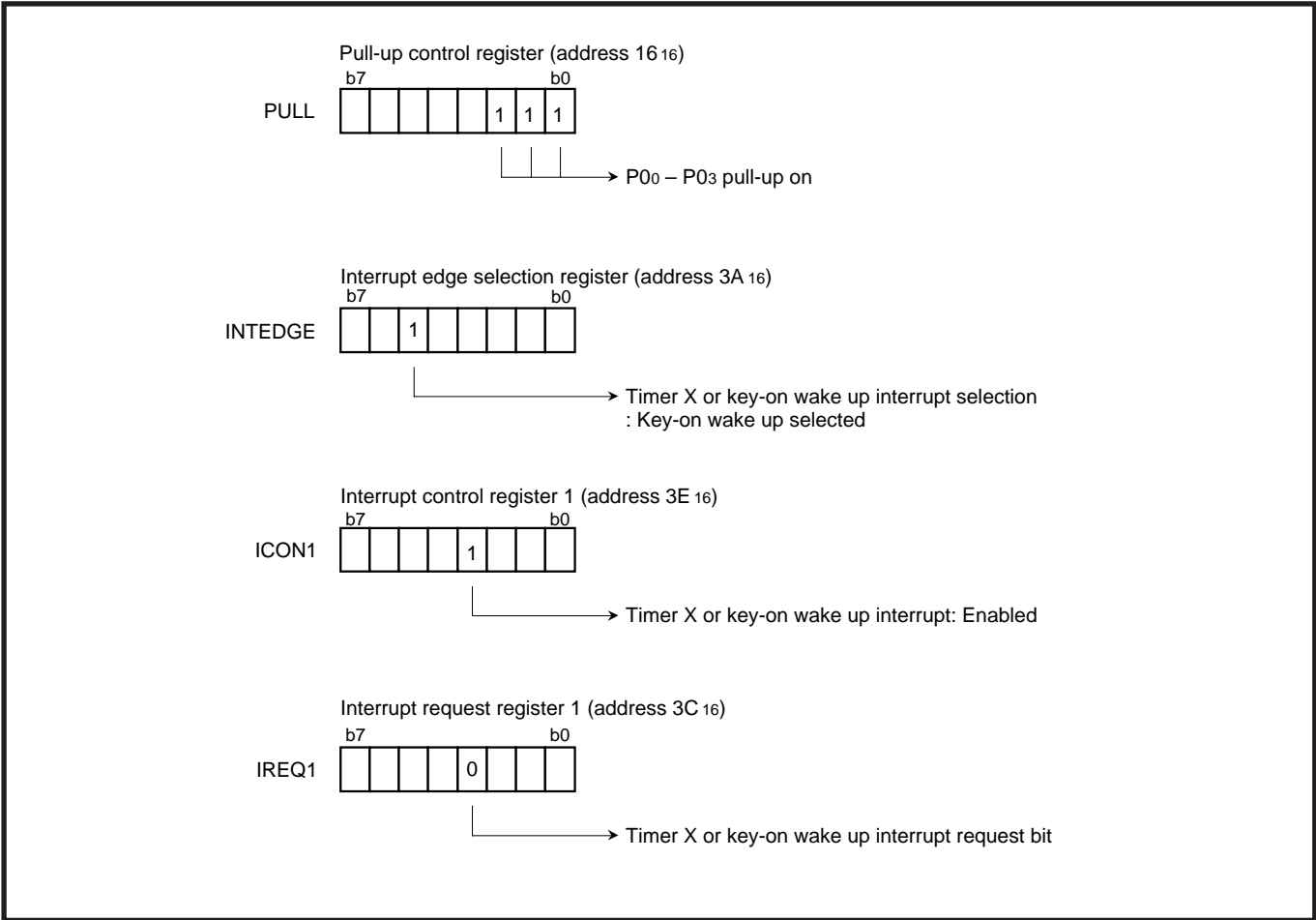


Fig. 2.1.9 Relevant registers setting

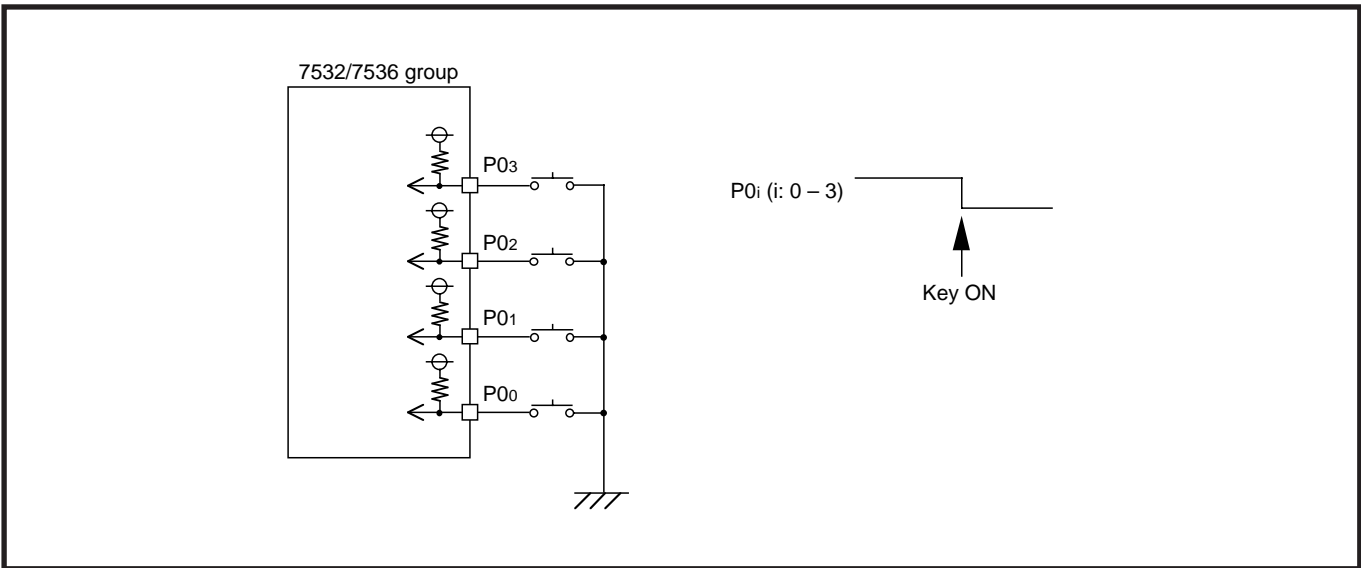


Fig. 2.1.10 Application circuit example

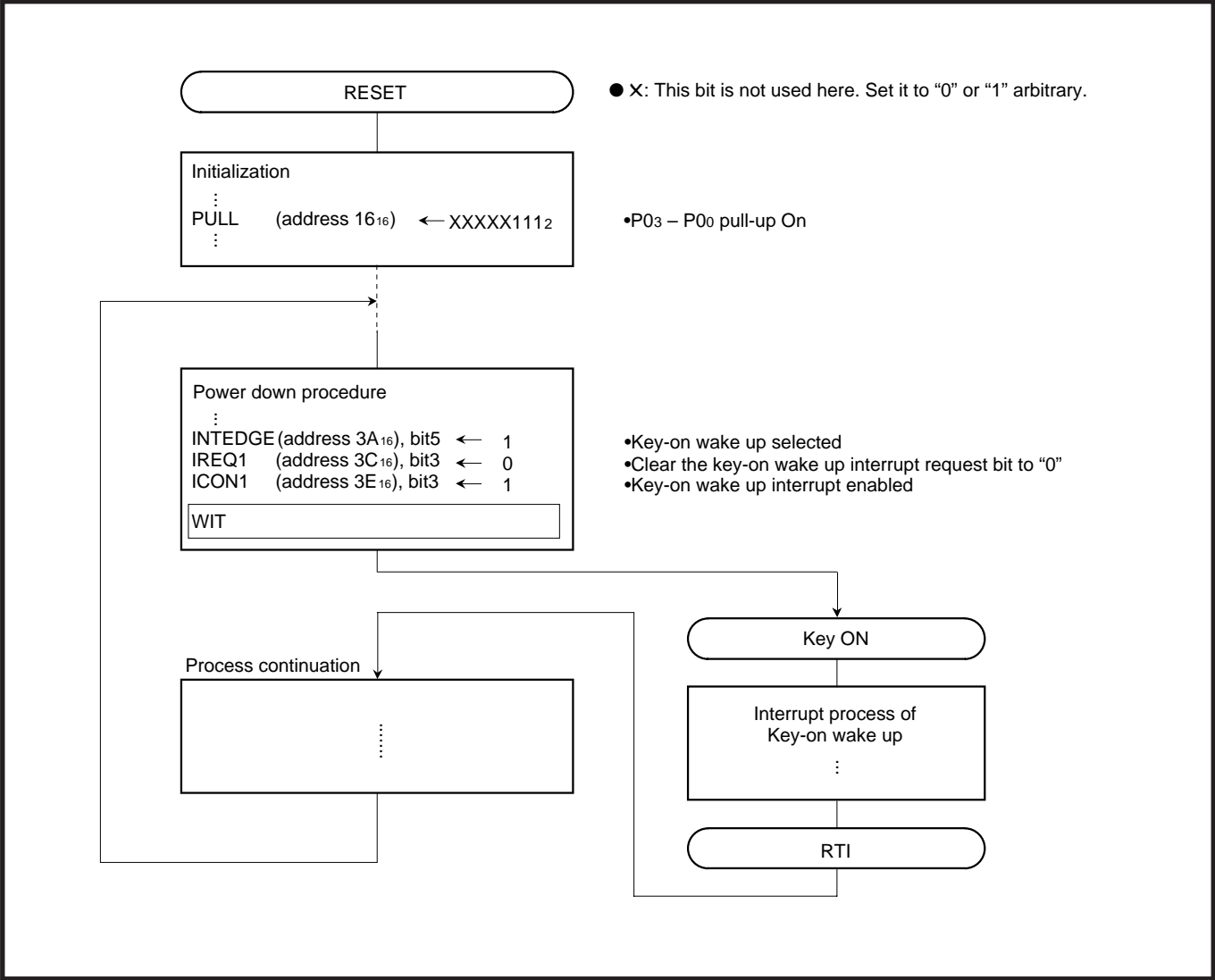


Fig. 2.1.11 Control procedure

2.1.4 Handling of unused pins

Table 2.1.1 Handling of unused pins

Pins/Ports name	Handling
P0, P1, P2, P3, P4	●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 <sub>REF</sub>	●Connect to Vss (GND).
X <sub>OUT</sub>	●Open, only when using an external clock

# APPLICATION

## 2.1 I/O port

---

### 2.1.5 Notes on input and output pins

#### (1) Notes in stand-by state

In stand-by state\*<sup>1</sup> for low-power dissipation, do not make input levels of an input port and an I/O port “undefined”.

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 a built-in pull-up 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

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.

\*<sup>1</sup> 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\*<sup>2</sup>, 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.

\*<sup>2</sup> bit managing instructions : **SEB**, and **CLB** instructions

### 2.1.6 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 $\Omega$  to 10 k $\Omega$ .

Ports that permit the selecting of a built-in pull-up resistor can also use this resistor. 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 $\Omega$  to 10 k $\Omega$ .

Ports that permit the selecting of a built-in pull-up resistor can also use this resistor. 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.

# 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

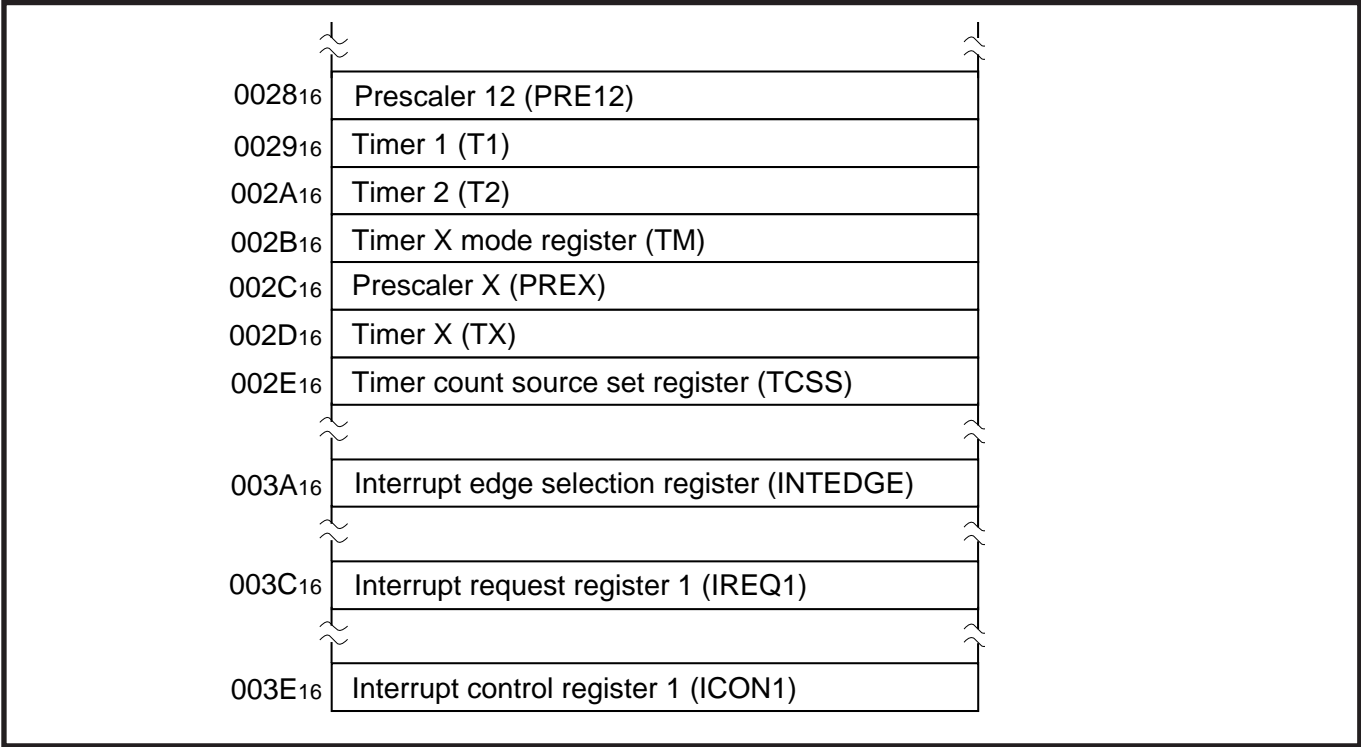


Fig. 2.2.1 Memory map of registers relevant to timers

#### 2.2.2 Relevant registers

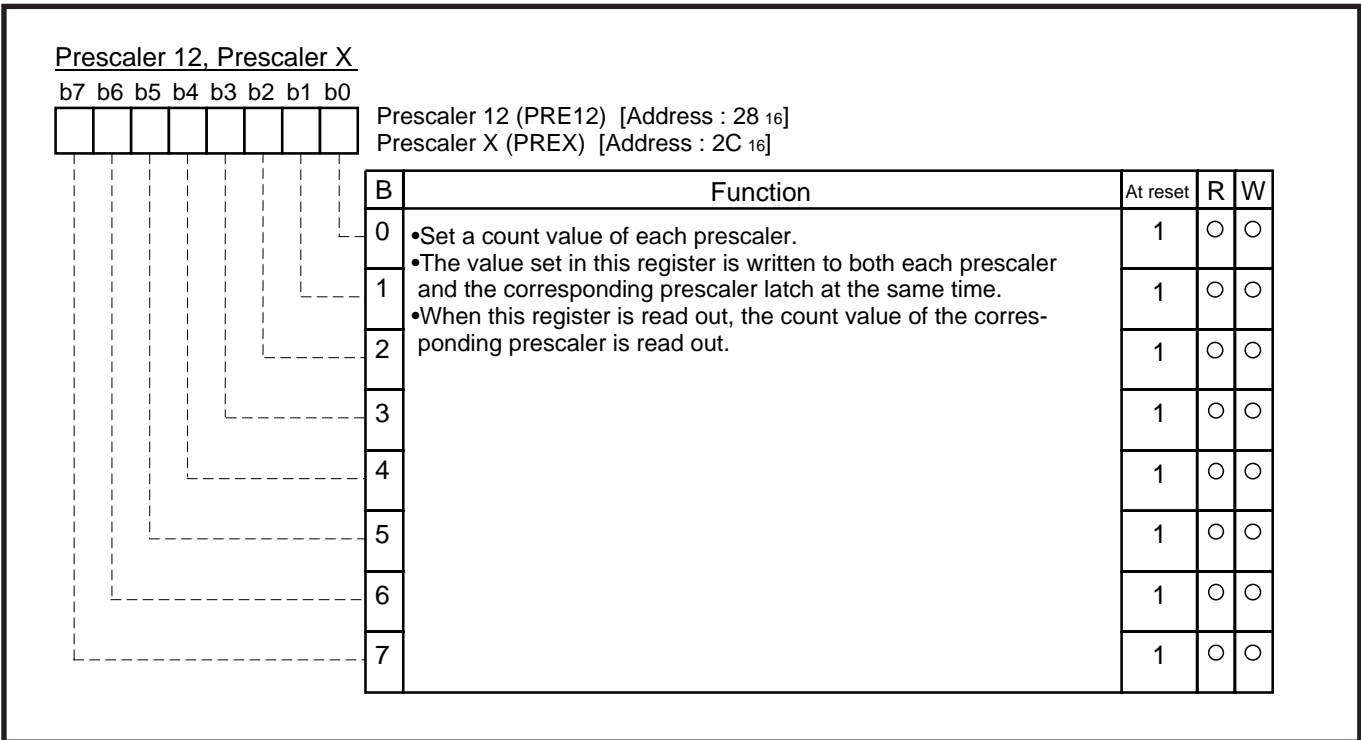
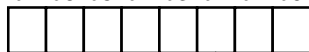


Fig. 2.2.2 Structure of Prescaler 12, Prescaler X

Timer 1

b7 b6 b5 b4 b3 b2 b1 b0

Timer 1 (T1) [Address : 29<sub>16</sub>]

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

Fig. 2.2.3 Structure of Timer 1

Timer 2

b7 b6 b5 b4 b3 b2 b1 b0

Timer 2 (T2) [Address : 2A<sub>16</sub>]

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

Fig. 2.2.4 Structure of Timer 2

# APPLICATION

## 2.2 Timer

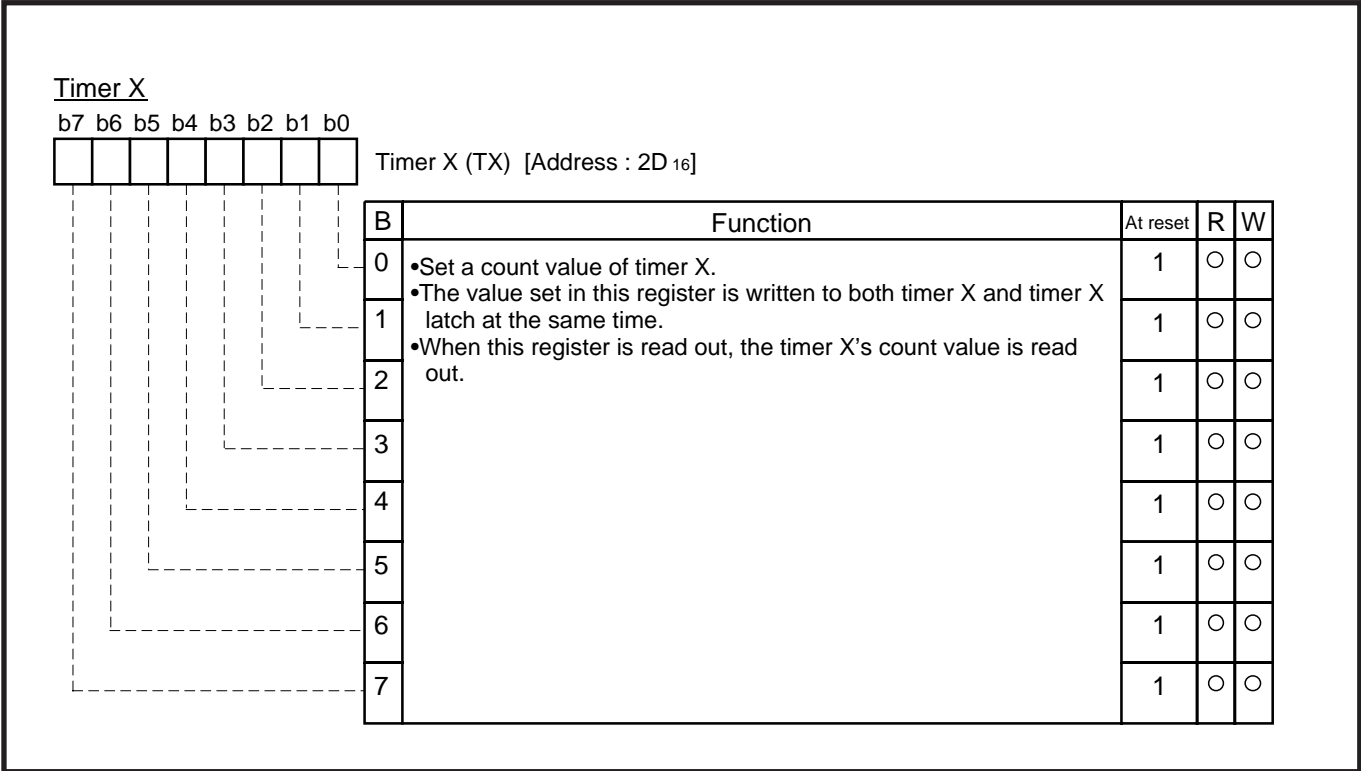


Fig. 2.2.5 Structure of Timer X

Timer X mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer X mode register (TM) [Address : 2B<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
1			0	○	○
2	CNTR <sub>0</sub> active edge switch bit	The function depends on the operating mode. (Refer to Table 2.2.1)	0	○	○
3	Timer X count stop bit	0 : Count start 1 : Count stop	0	○	○
4	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 2.2.6 Structure of Timer X mode register

Table 2.2.1 CNTR<sub>0</sub> active edge switch bit function

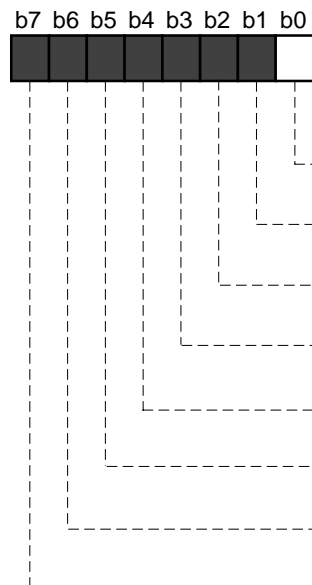
Timer X operation modes	CNTR <sub>0</sub> active edge switch bit (bit 2 of address 2B <sub>16</sub> ) contents	
Timer mode	"0"	CNTR <sub>0</sub> interrupt request occurrence: Falling edge ; No influence to timer count
	"1"	CNTR <sub>0</sub> interrupt request occurrence: Rising edge ; No influence to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



# APPLICATION

## 2.2 Timer

### Timer count source set register

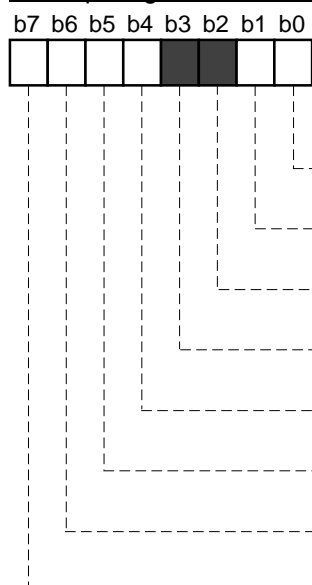
b7 b6 b5 b4 b3 b2 b1 b0  

 Timer count source set register (TCSS) [Address : 2E<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Timer X count source selection bit ( <b>Note</b> )	0 : f(X <sub>IN</sub> ) / 16 1 : f(X <sub>IN</sub> ) / 2	0	○	○
1	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
2			0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

**Note:** To switch the timer X count source selection bit, stop the timer X count operation before do that.

Fig. 2.2.7 Structure of Timer count source set register

### Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0  

 Interrupt edge selection register (INTEDGE) [Address : 3A<sub>16</sub>]

B	Name	Function	At reset	R	W
0	INT <sub>0</sub> interrupt edge selection bit ( <b>Note 1</b> )	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT <sub>1</sub> interrupt edge selection bit ( <b>Note 2</b> )	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
3			0	○	×
4	Serial I/O1 or INT <sub>1</sub> interrupt selection bit	0 : Serial I/O1 1 : INT <sub>1</sub>	0	○	○
5	Timer X or key-on wake up interrupt selection bit	0 : Timer X 1 : Key-on wake up	0	○	○
6	Timer 2 or serial I/O2 interrupt selection bit	0 : Timer 2 1 : Serial I/O2	0	○	○
7	CNTR <sub>0</sub> or AD converter interrupt selection bit	0 : CNTR <sub>0</sub> 1 : AD converter	0	○	○

**Notes 1:** 7532 Group, 32-pin version: Not used.

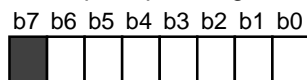
This is a write disabled bit. When this bit is read out, the value is "0".

**2:** 7532 Group, 36-pin and 32-pin version: Not used.

This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.2.8 Structure of Interrupt edge selection register

Interrupt request register 1

Interrupt request register 1 (IREQ1) [Address : 3C<sub>16</sub>]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
1	UART transmit/USBSETUP/OUT token/Reset/Suspend/Resume/INT <sub>1</sub> interrupt request bit ( <b>Note 1</b> )	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
2	INT <sub>0</sub> interrupt request bit ( <b>Note 2</b> )	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
3	Timer X or key-on wake up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
4	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
5	Timer 2 or serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
6	CNTR <sub>0</sub> or AD converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

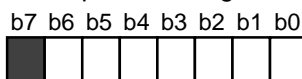
\*: These bits can be cleared to "0" by program, but cannot be set.

**Notes 1:** 7532 Group, 36-pin version and 32-pin version: INT<sub>1</sub> interrupt does not exist.**2:** 7532 Group, 32-pin version: INT<sub>0</sub> interrupt does not exist.

This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.2.9 Structure of Interrupt request register 1

Interrupt control register 1

Interrupt control register 1 (ICON1) [Address : 3E<sub>16</sub>]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	UART transmit/USBSETUP/OUT token/Reset/Suspend/Resume/INT <sub>1</sub> interrupt enable bit ( <b>Note 1</b> )	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT <sub>0</sub> interrupt enable bit ( <b>Note 2</b> )	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Timer X or key-on wake up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer 2 or serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	CNTR <sub>0</sub> or AD converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Nothing is allocated for this bit. Do not write "1" to this bit. When this bit is read out, the value is "0".		0	○	×

**Notes 1:** 7532 Group, 36-pin version and 32-pin version: INT<sub>1</sub> interrupt does not exist.**2:** 7532 Group, 32-pin version: INT<sub>0</sub> interrupt does not exist.

This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.2.10 Structure of Interrupt control register 1

# APPLICATION

## 2.2 Timer

---

### 2.2.3 Timer application examples

#### (1) Basic functions and uses

##### [Function 1] Control of Event interval (Timer X, 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 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 100 ms); see Application example 1
- Control of a main routine cycle

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

The output level of the CNTR<sub>0</sub> 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)

External pulses input to the CNTR<sub>0</sub> 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)

The “H” or “L” level width of external pulses input to CNTR<sub>0</sub> 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.

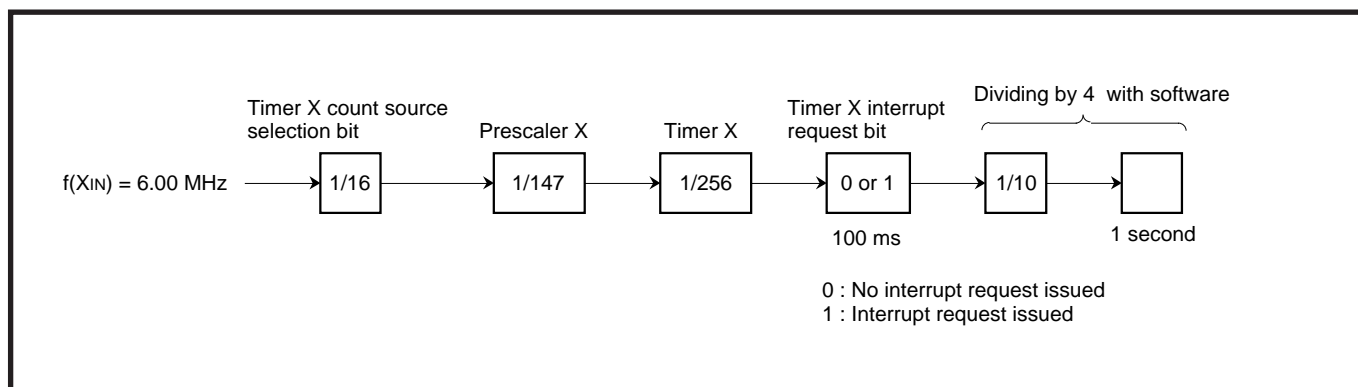
**(2) Timer application example 1: Clock function (measurement of 100 ms)**

**Outline:** The input clock is divided by the timer so that the clock can count up at 100 ms intervals.

**Specifications:** •The clock  $f(X_{IN}) = 6.00$  MHz is divided by the timer.

•The clock is counted up in the process routine of the timer X interrupt which occurs at 100 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

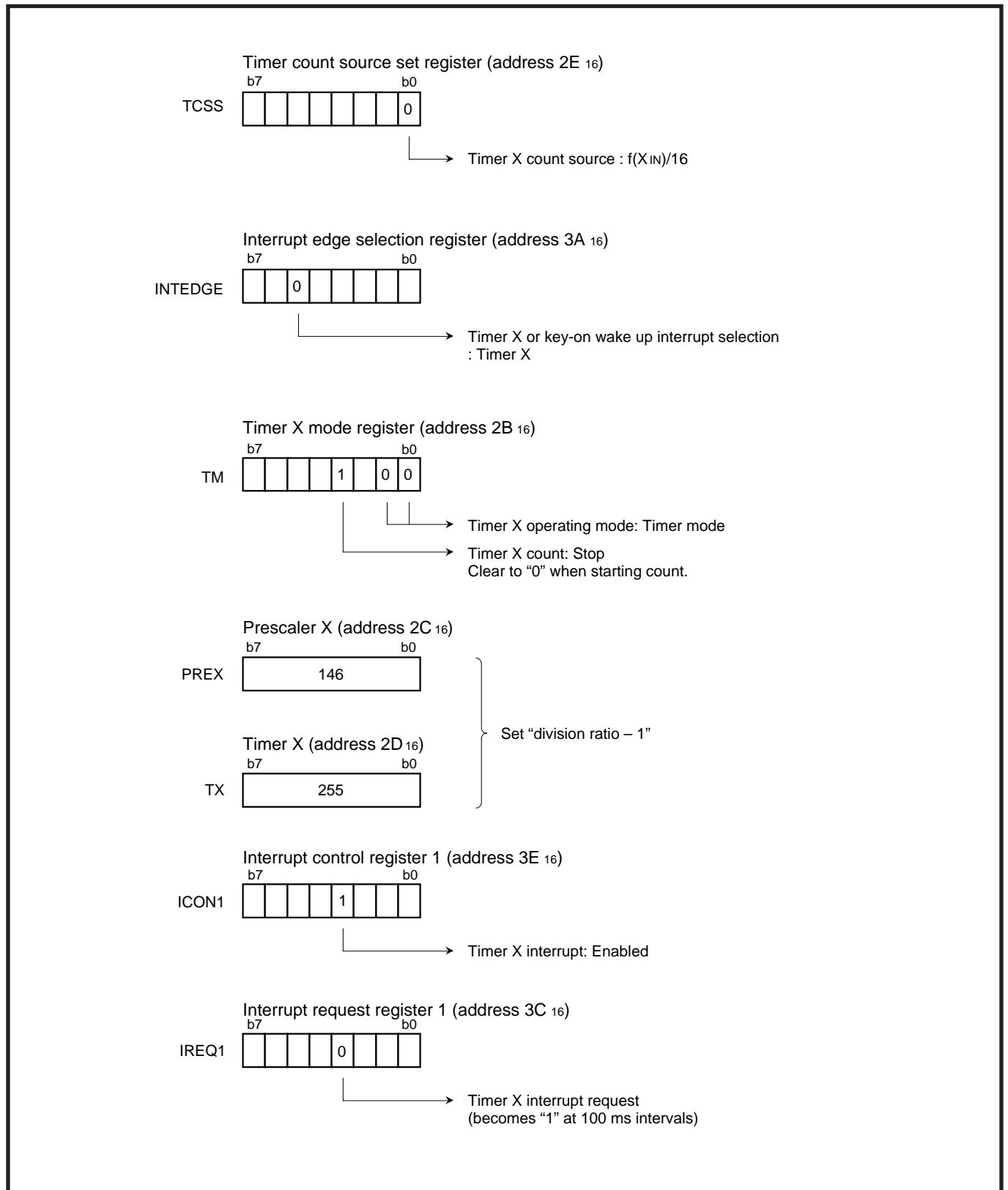


Fig. 2.2.12 Relevant registers setting

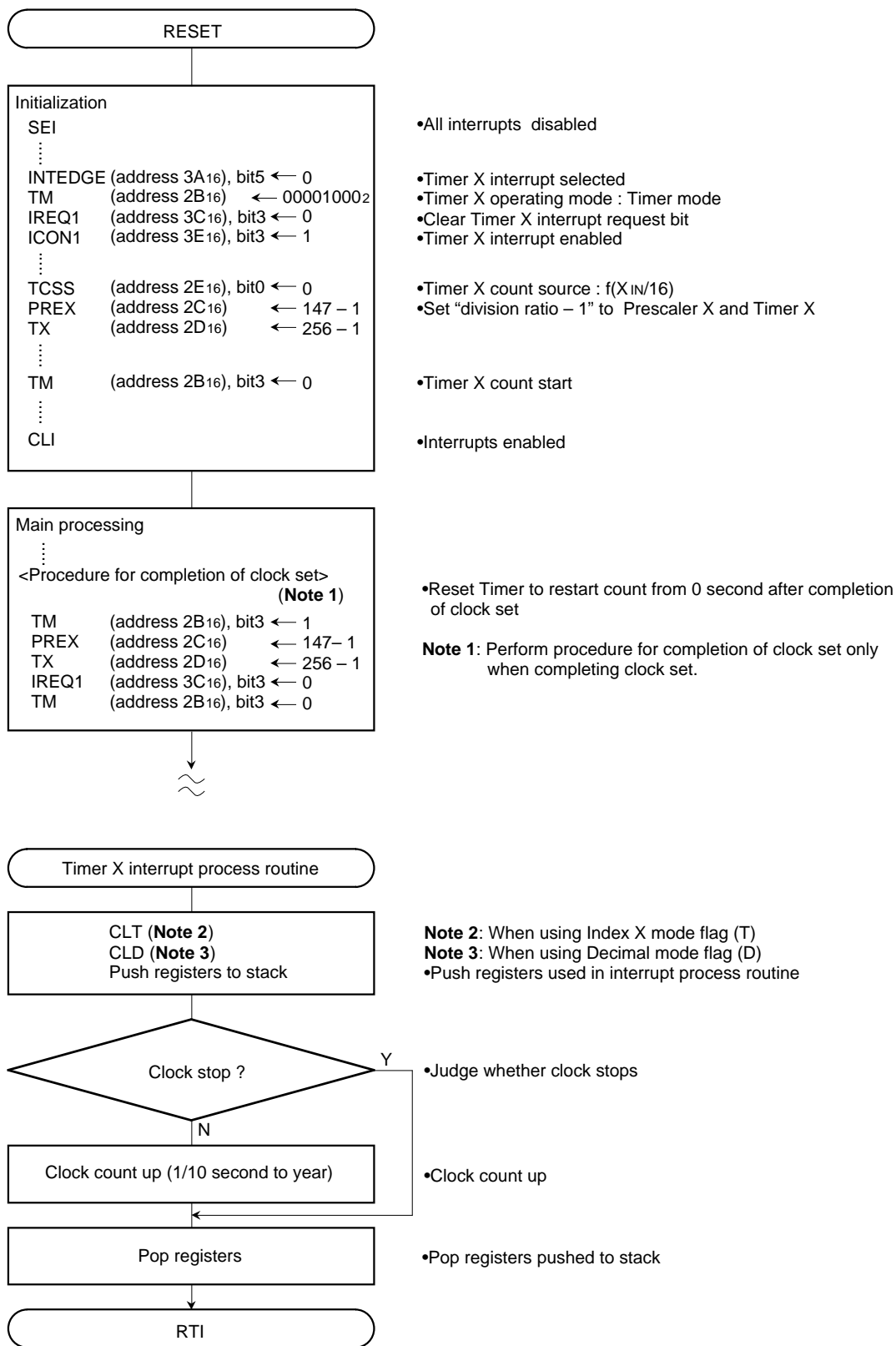


Fig. 2.2.13 Control procedure

# 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  $f(X_{IN}) = 6.00 \text{ MHz}$  into about 2 kHz (1995 Hz), is output from the P14/CNTR<sub>0</sub> pin.
- The level of the P14/CNTR<sub>0</sub> 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.

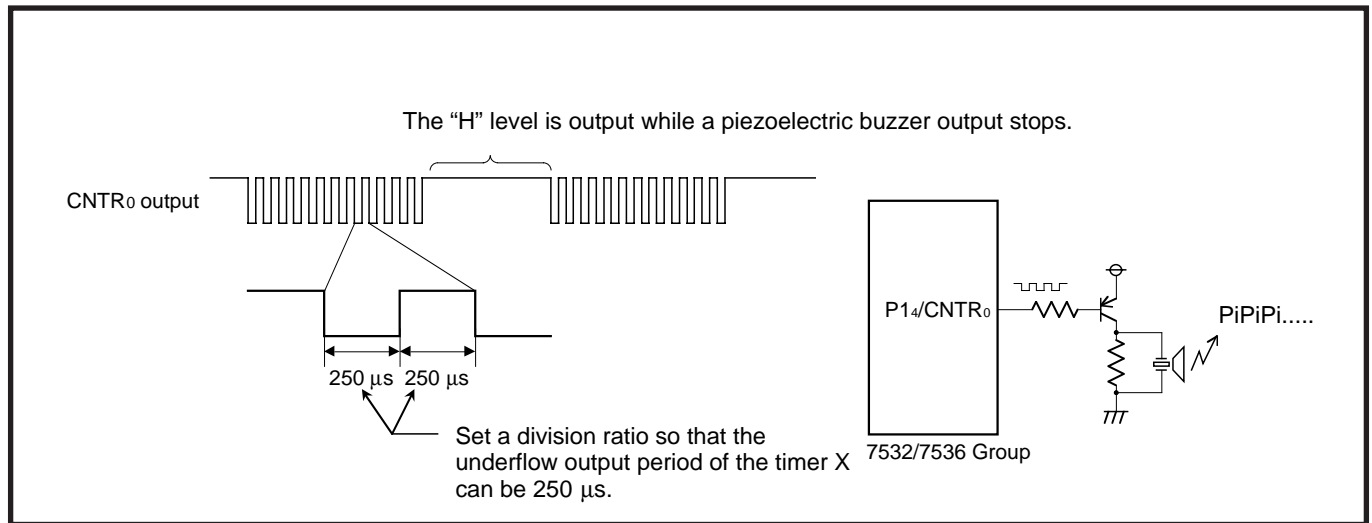


Fig. 2.2.14 Peripheral circuit example

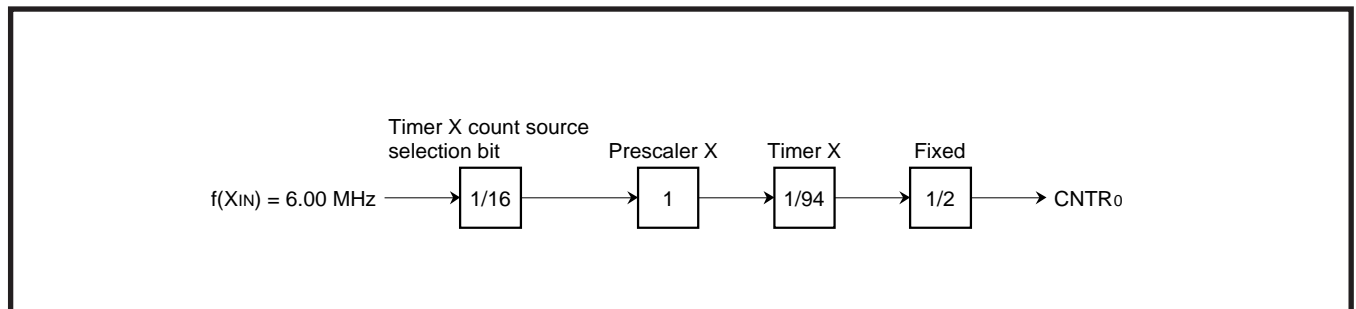


Fig. 2.2.15 Timers connection and setting of division ratios

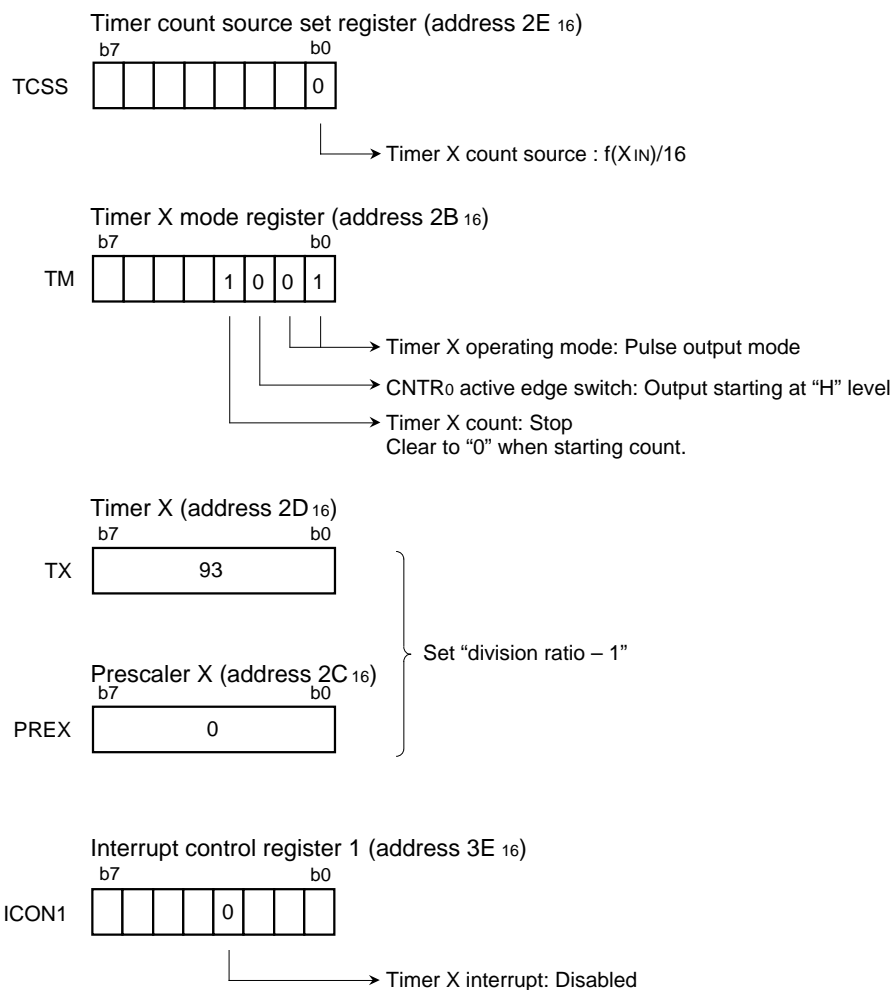


Fig. 2.2.16 Relevant registers setting



# APPLICATION

## 2.2 Timer

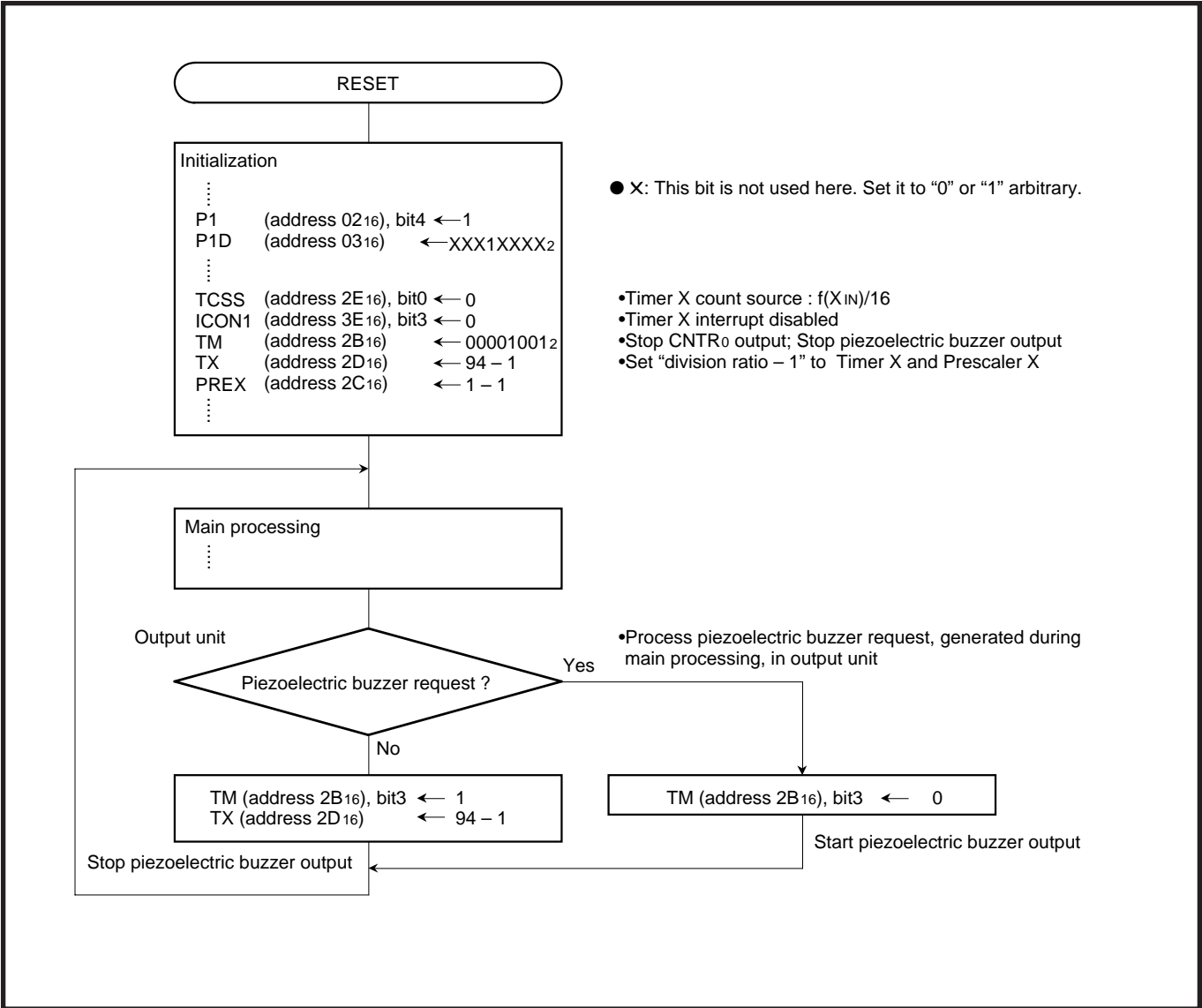


Fig. 2.2.17 Control procedure

**(4) Timer application example 3: Frequency measurement**

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

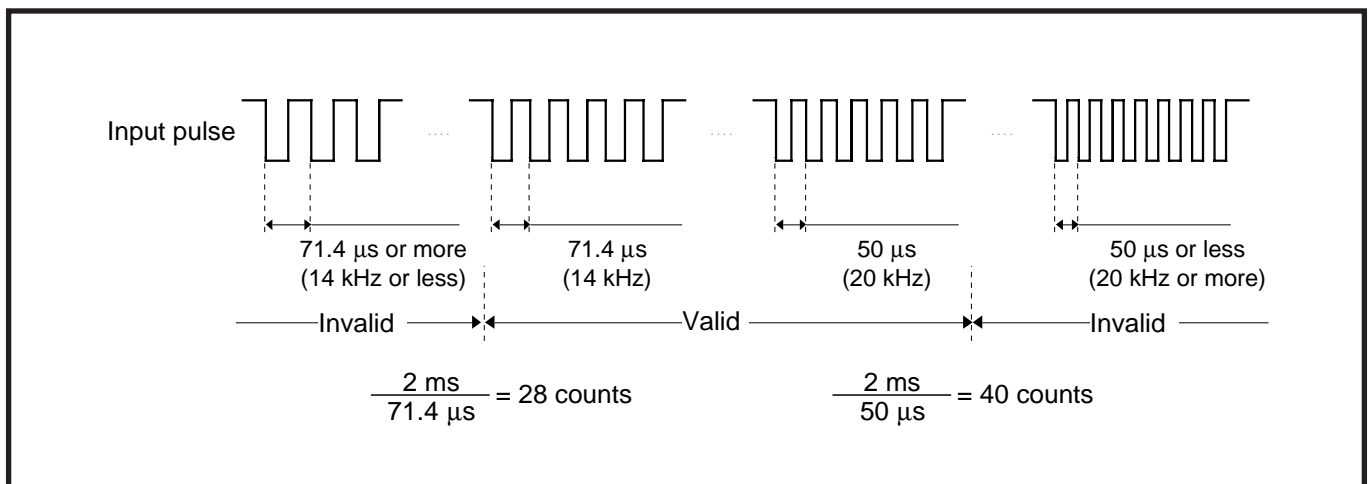
- A value by counting pulses input to P14/CNTR<sub>0</sub> pin with the timer.
- A reference value

**Specifications:** •The pulse is input to the P14/CNTR<sub>0</sub> pin and counted by the timer X.

- 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

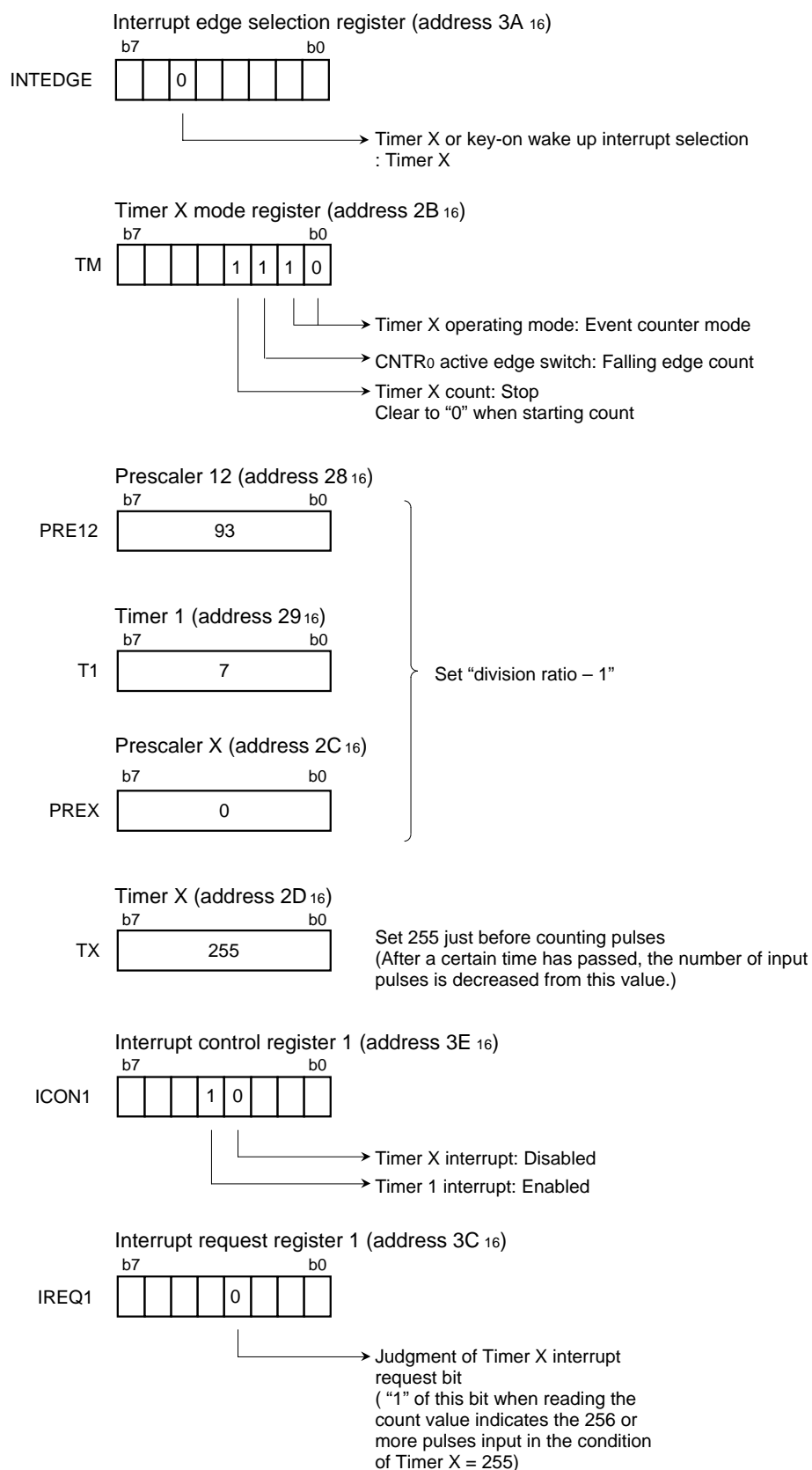


Fig. 2.2.19 Relevant registers setting

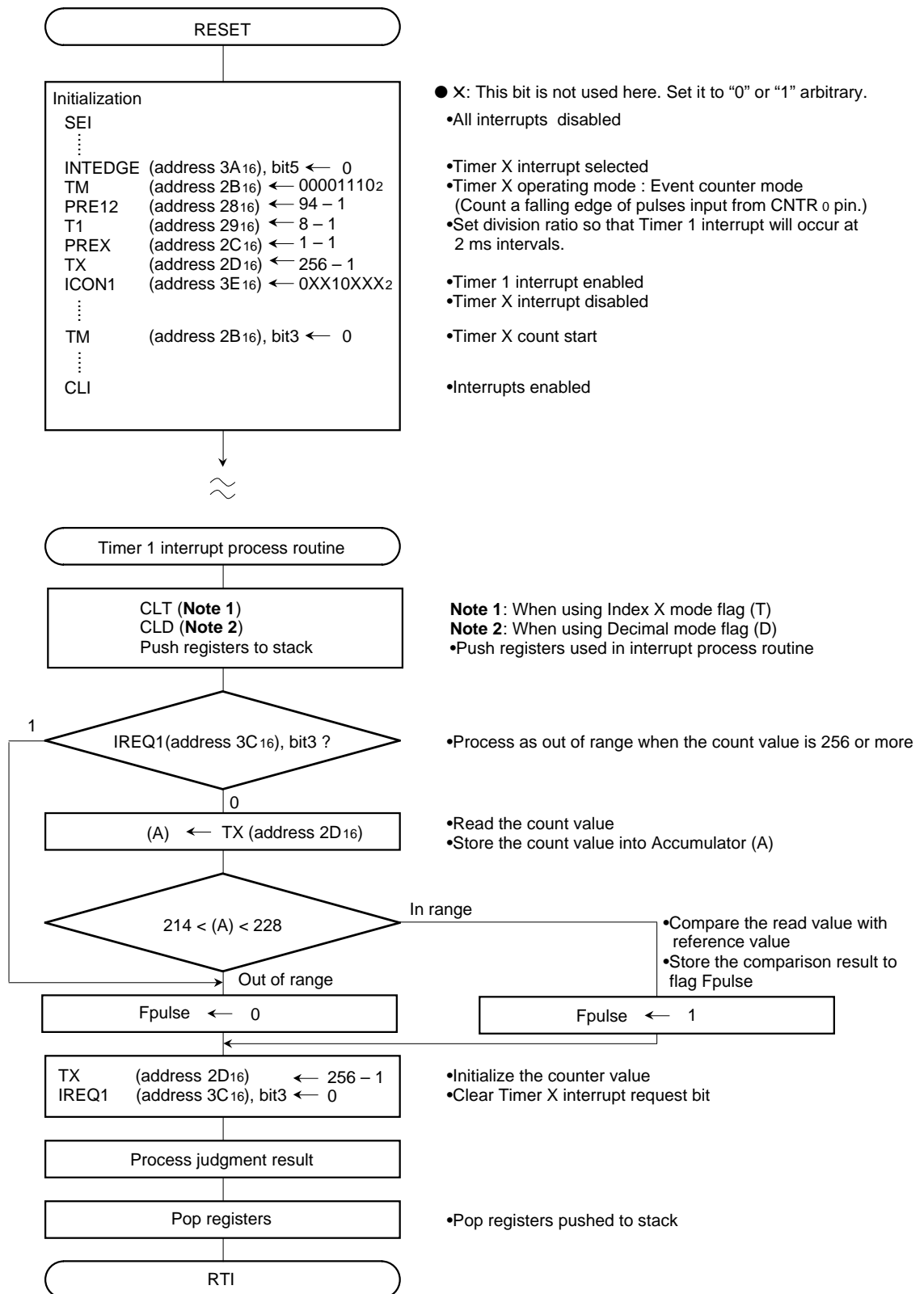


Fig. 2.2.20 Control procedure

# 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 P14/CNTR<sub>0</sub> pin. An underflow is detected by the timer X interrupt and an end of the input pulse “H” level is detected by the CNTR<sub>0</sub> interrupt.

**Specifications:** •The timer X counts the “H” level width of the FG pulse input to the P14/CNTR<sub>0</sub> pin.

<Example>

When the clock frequency is 6.00 MHz, the count source is 2.7  $\mu$ s, which is obtained by dividing the clock frequency by 16. Measurement can be made up to 174 ms in the range of FFFF<sub>16</sub> to 0000<sub>16</sub>.

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.

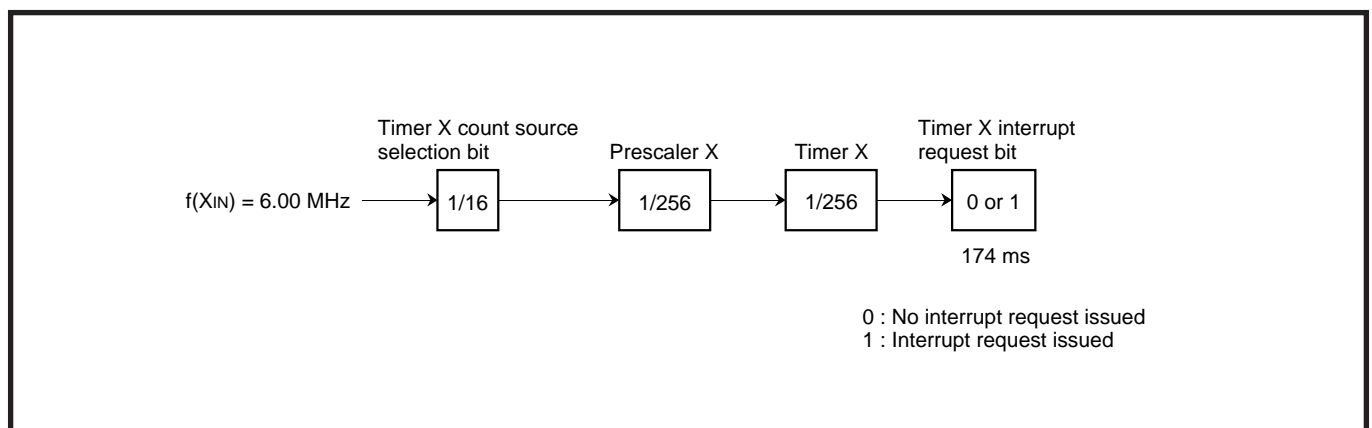


Fig. 2.2.21 Timers connection and setting of division ratios

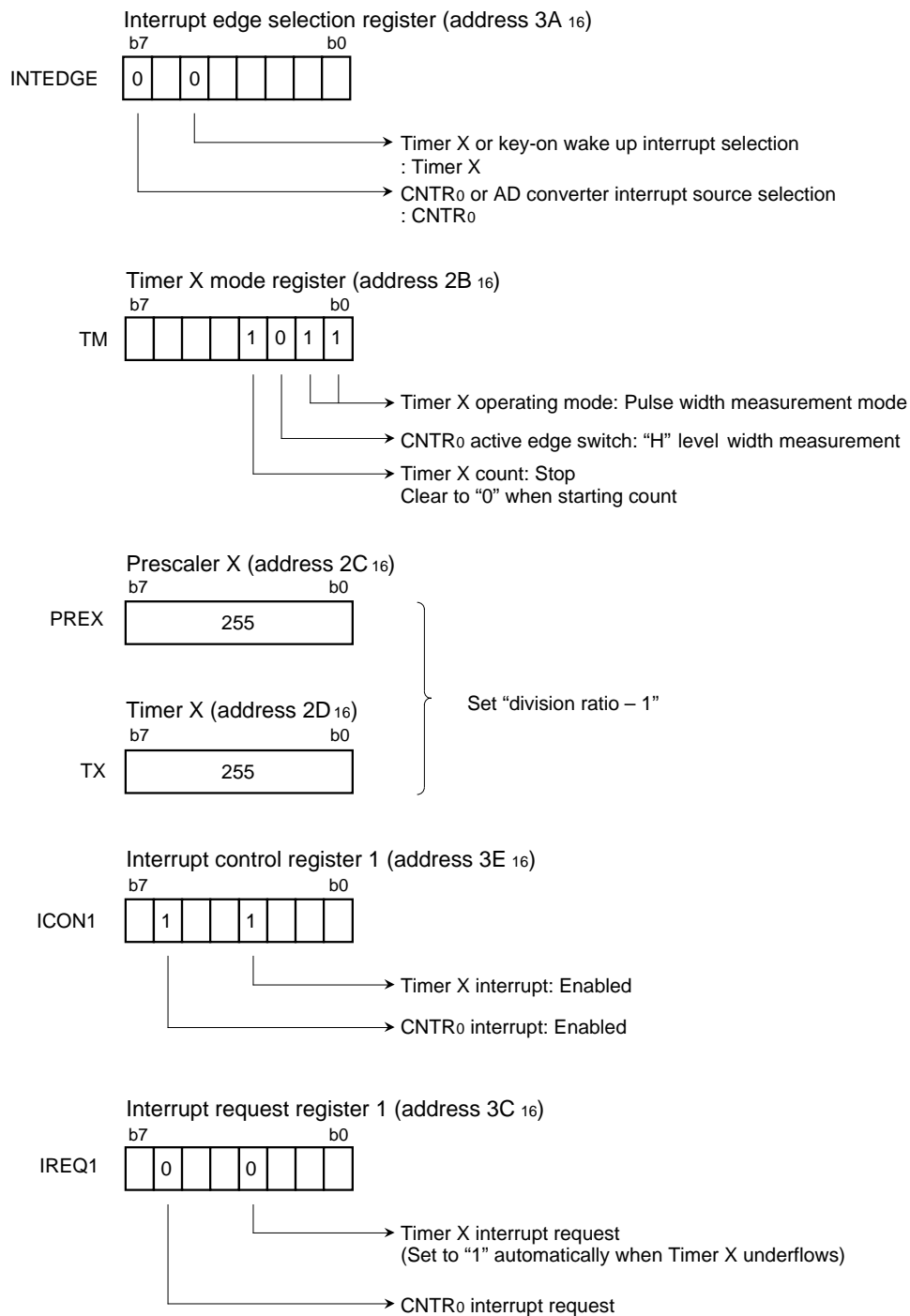


Fig. 2.2.22 Relevant registers setting

# APPLICATION

## 2.2 Timer

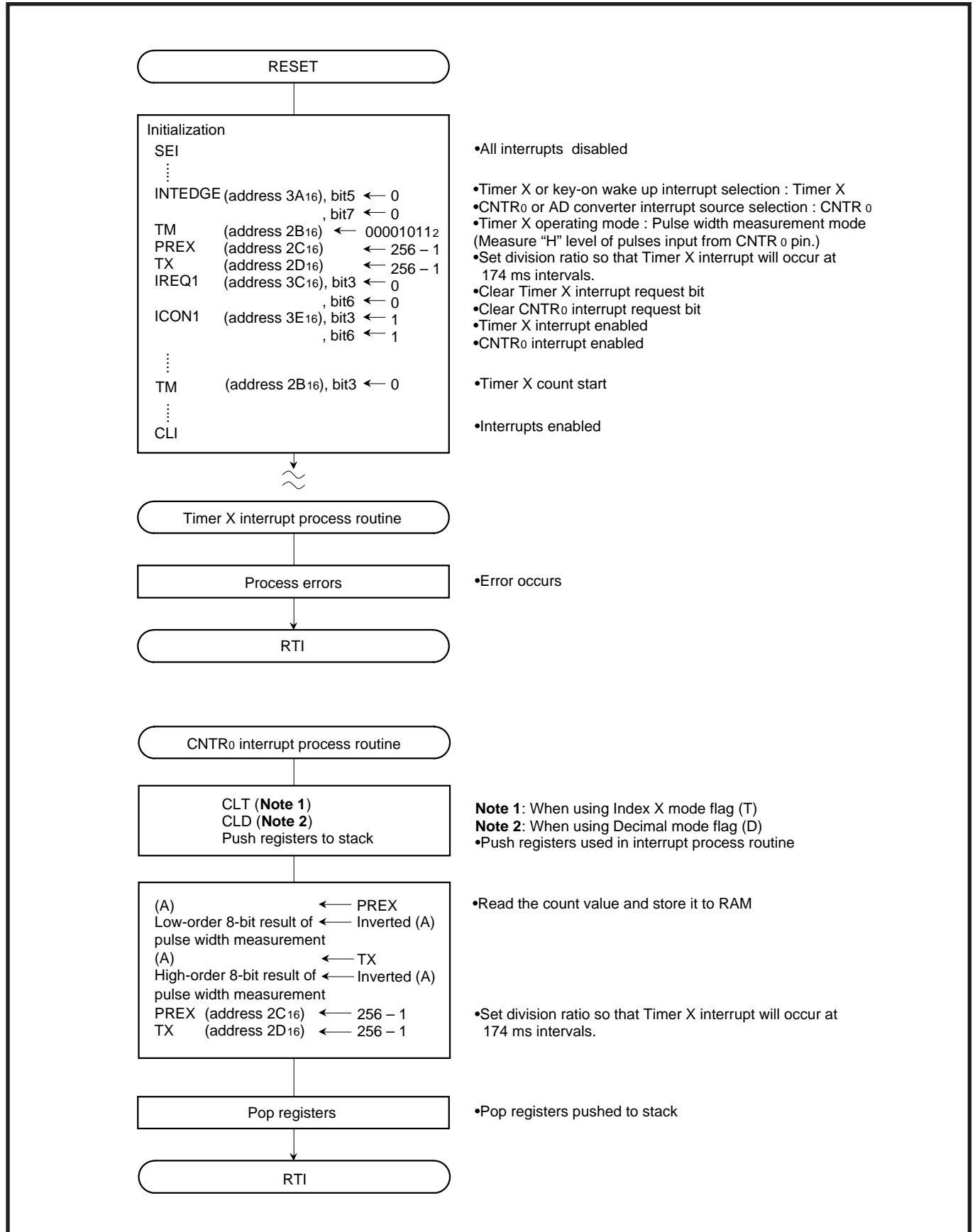


Fig. 2.2.23 Control procedure

## 2.3 Serial I/O

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

### 2.3.1 Memory map

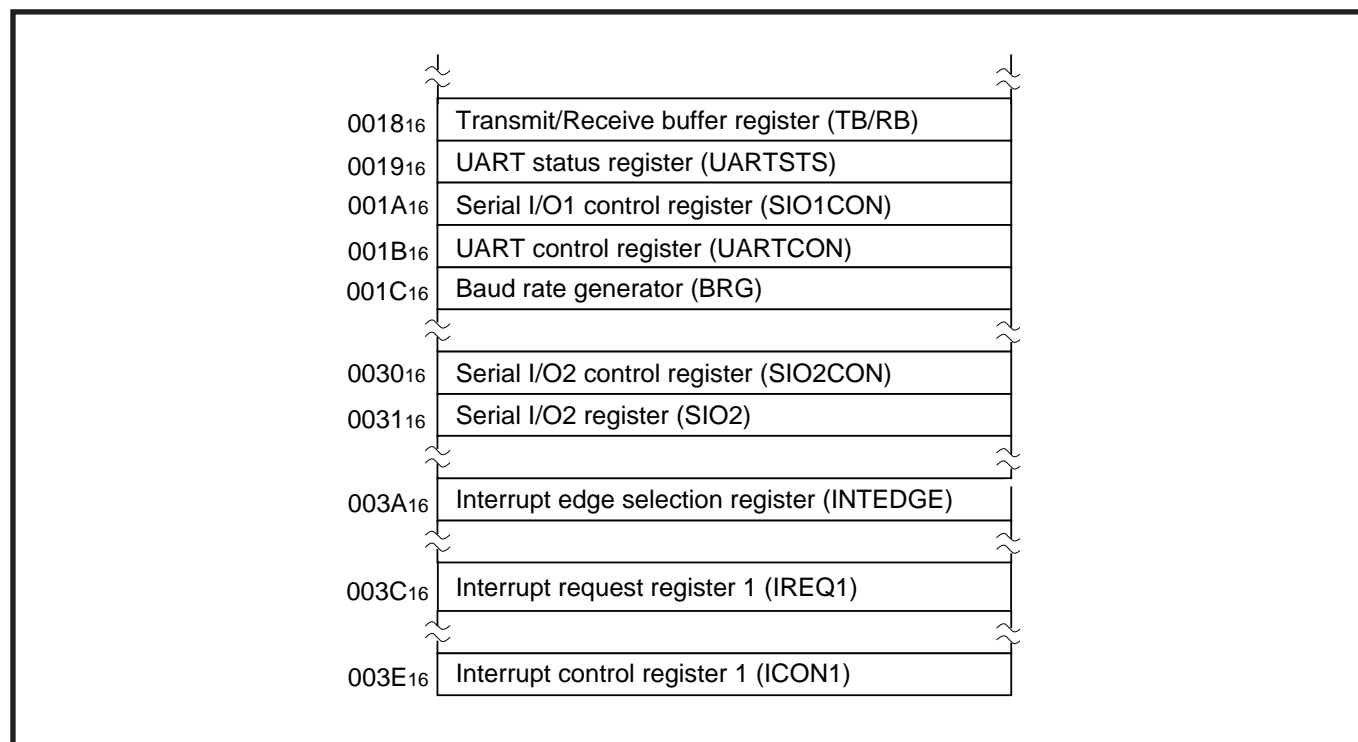


Fig. 2.3.1 Memory map of registers relevant to serial I/O

### 2.3.2 Relevant registers

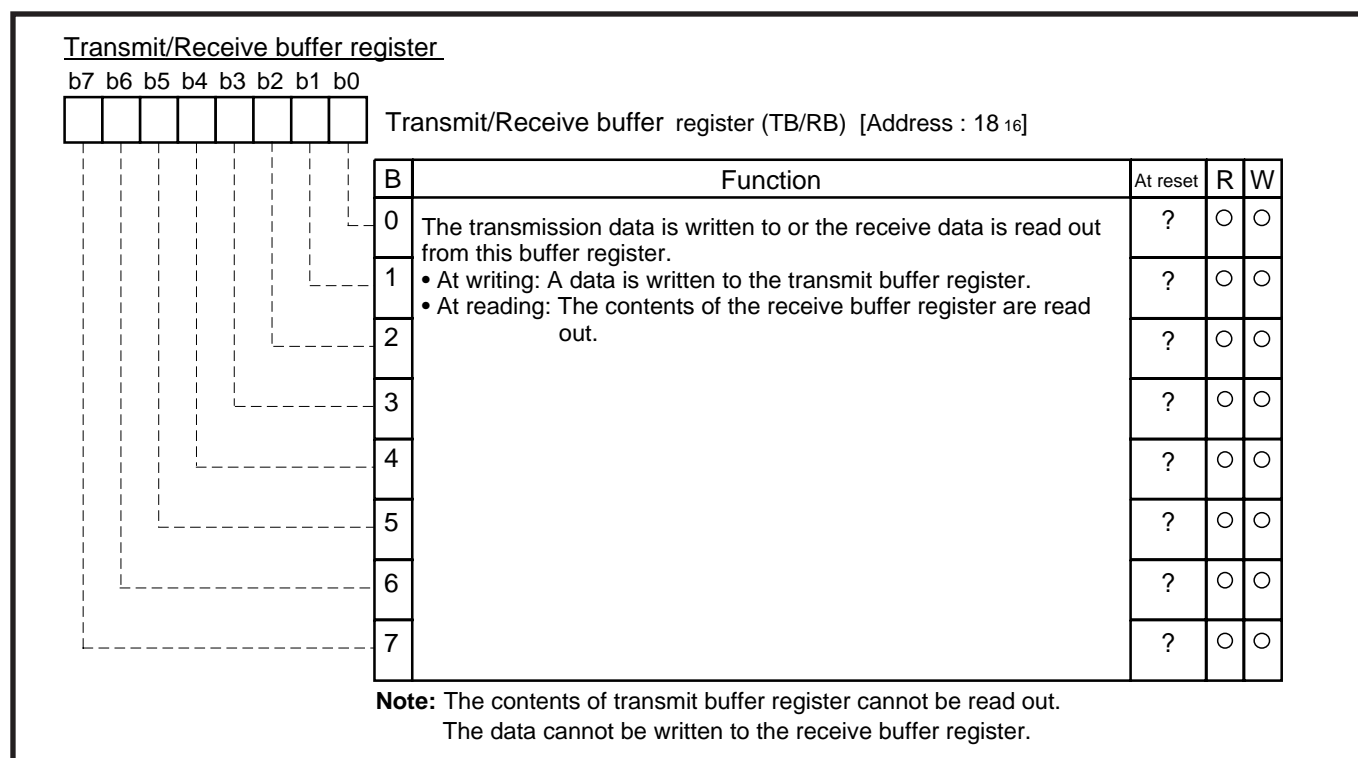


Fig. 2.3.2 Structure of Transmit/Receive buffer register



# APPLICATION

## 2.3 Serial I/O

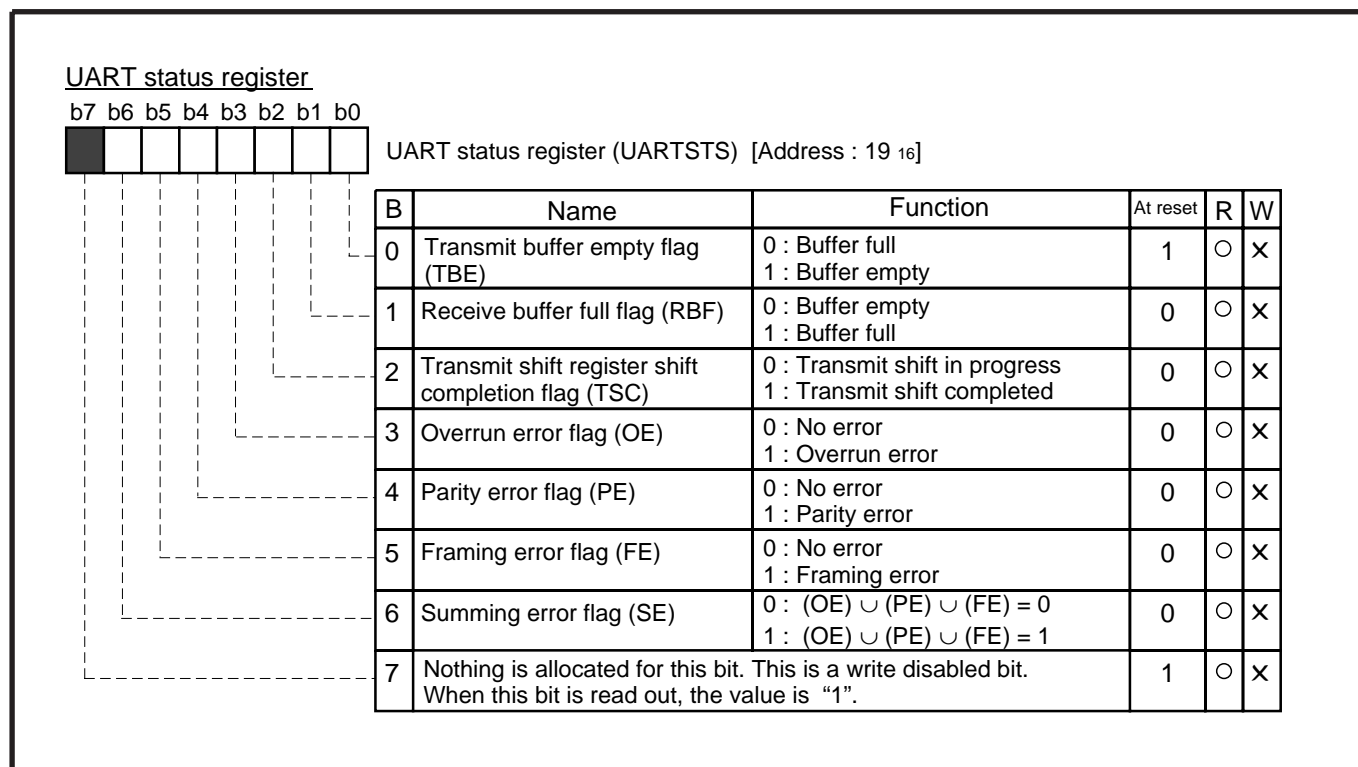


Fig. 2.3.3 Structure of UART status register

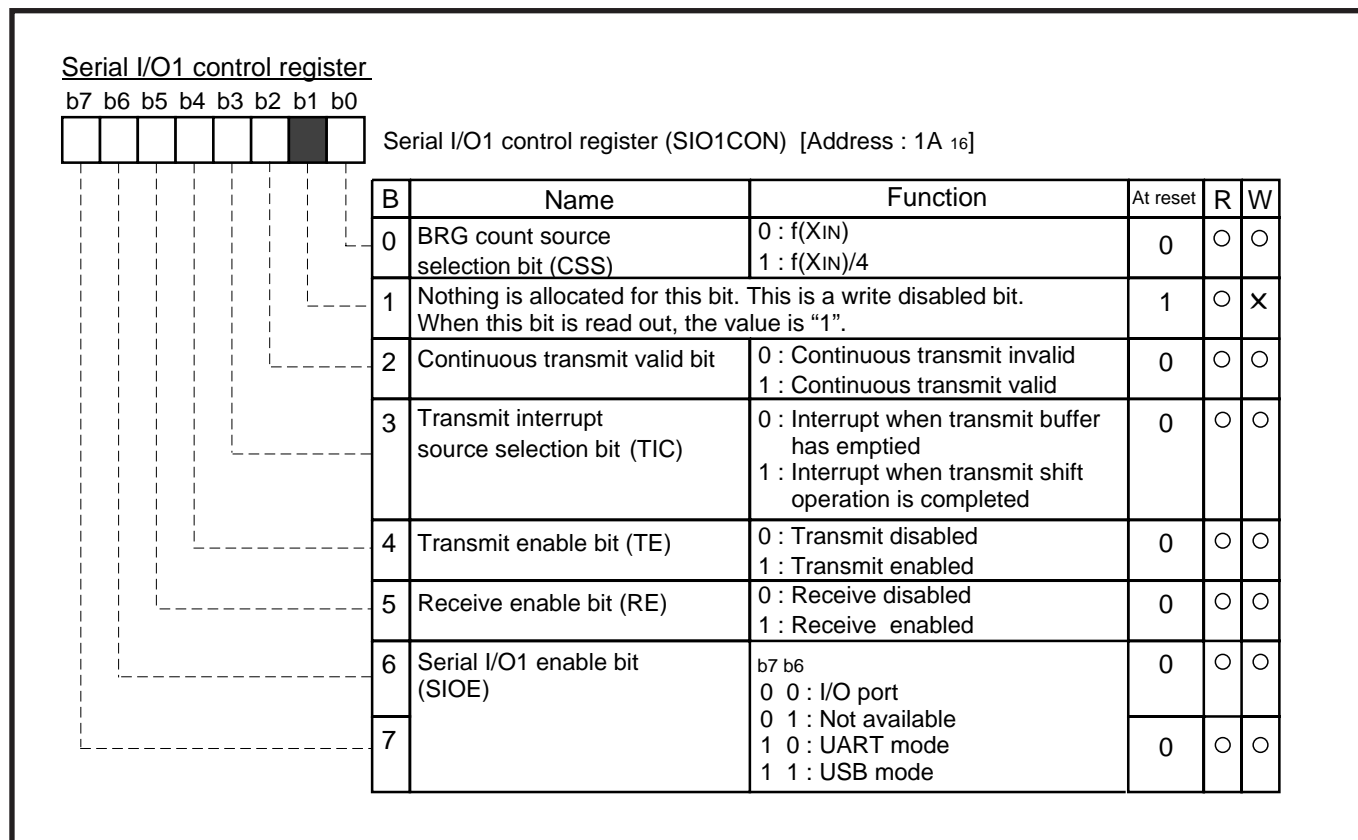
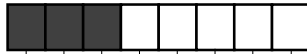


Fig. 2.3.4 Structure of Serial I/O1 control register

UART control register

b7 b6 b5 b4 b3 b2 b1 b0

UART control register (UARTCON) [Address : 1B<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Character length selection bit (CHAS)	0 : 8 bits 1 : 7 bits	0	○	○
1	Parity enable bit (PARE)	0 : Parity checking disabled 1 : Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0 : Even parity 1 : Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0 : 1 stop bit 1 : 2 stop bits	0	○	○
4	P11/TxD P-channel output disable bit (POFF)	In output mode 0 : CMOS output 1 : N-channel open-drain output	0	○	○
5	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "1".		1	○	×
6			1	○	×
7			1	○	×

Fig. 2.3.5 Structure of UART control register

Baud rate generator

b7 b6 b5 b4 b3 b2 b1 b0

Baud rate generator (BRG) [Address : 1C<sub>16</sub>]

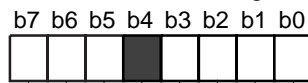
B	Function	At reset	R	W
0	Set a count value of baud rate generator.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 2.3.6 Structure of Baud rate generator

# APPLICATION

## 2.3 Serial I/O

Serial I/O2 control register



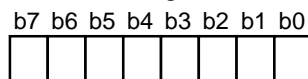
Serial I/O2 control register (SIO2CON) [Address : 30<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Internal synchronous clock selection bits	b2 b1 b0	0	○	○
1		0 0 0 : $f(X_{IN})/8$	0	○	○
		0 0 1 : $f(X_{IN})/16$			
		0 1 0 : $f(X_{IN})/32$			
		0 1 1 : $f(X_{IN})/64$			
2		1 1 0 : $f(X_{IN})/128$	0	○	○
		1 1 1 : $f(X_{IN})/256$			
3	S <sub>DATA</sub> pin selection bit (Note)	0 : I/O port / S <sub>DATA</sub> input 1 : S <sub>DATA</sub> output	0	○	○
4	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×
5	Transfer direction selection bit	0 : LSB first 1 : MSB first	0	○	○
6	S <sub>CLK</sub> pin selection bit	0 : External clock (S <sub>CLK</sub> is input) 1 : Internal clock (S <sub>CLK</sub> is output)	0	○	○
7	Transmit / receive shift completion flag	0 : shift in progress 1 : shift completed	0	○	×

**Note:** When using it as a S<sub>DATA</sub> input, set the port P1<sub>3</sub> direction register bit to "0".

Fig. 2.3.7 Structure of Serial I/O2 control register

Serial I/O2 register



Serial I/O2 register (SIO2) [Address : 31<sub>16</sub>]

B	Function	At reset	R	W
0	A shift register for serial transmission and reception. • At transmitting : Set a transmission data. • At receiving : A reception data is stored.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 2.3.8 Structure of Serial I/O2 register

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt edge selection register (INTEDGE) [Address : 3A<sub>16</sub>]

B	Name	Function	At reset	R	W
0	INT <sub>0</sub> interrupt edge selection bit ( <b>Note 1</b> )	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT <sub>1</sub> interrupt edge selection bit ( <b>Note 2</b> )	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
3			0	○	×
4	Serial I/O1 or INT <sub>1</sub> interrupt selection bit	0 : Serial I/O1 1 : INT <sub>1</sub>	0	○	○
5	Timer X or key-on wake up interrupt selection bit	0 : Timer X 1 : Key-on wake up	0	○	○
6	Timer 2 or serial I/O2 interrupt selection bit	0 : Timer 2 1 : Serial I/O2	0	○	○
7	CNTR <sub>0</sub> or AD converter interrupt selection bit	0 : CNTR <sub>0</sub> 1 : AD converter	0	○	○

**Notes 1:** 7532 Group, 32-pin version: Not used.

This is a write disabled bit. When this bit is read out, the value is "0".

**2:** 7532 Group, 36-pin and 32-pin version: Not used.

This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.3.9 Structure of Interrupt edge selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : 3C<sub>16</sub>]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
1	UART transmit/USBSETUP/OUT token/Reset/Suspend/Resume/INT <sub>1</sub> interrupt request bit ( <b>Note 1</b> )	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
2	INT <sub>0</sub> interrupt request bit ( <b>Note 2</b> )	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
3	Timer X or key-on wake up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
4	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
5	Timer 2 or serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
6	CNTR <sub>0</sub> or AD converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

\*: These bits can be cleared to "0" by program, but cannot be set.

**Notes 1:** 7532 Group, 36-pin version and 32-pin version: INT<sub>1</sub> interrupt does not exist.**2:** 7532 Group, 32-pin version: INT<sub>0</sub> interrupt does not exist.

This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.3.10 Structure of Interrupt request register 1

# APPLICATION

## 2.3 Serial I/O

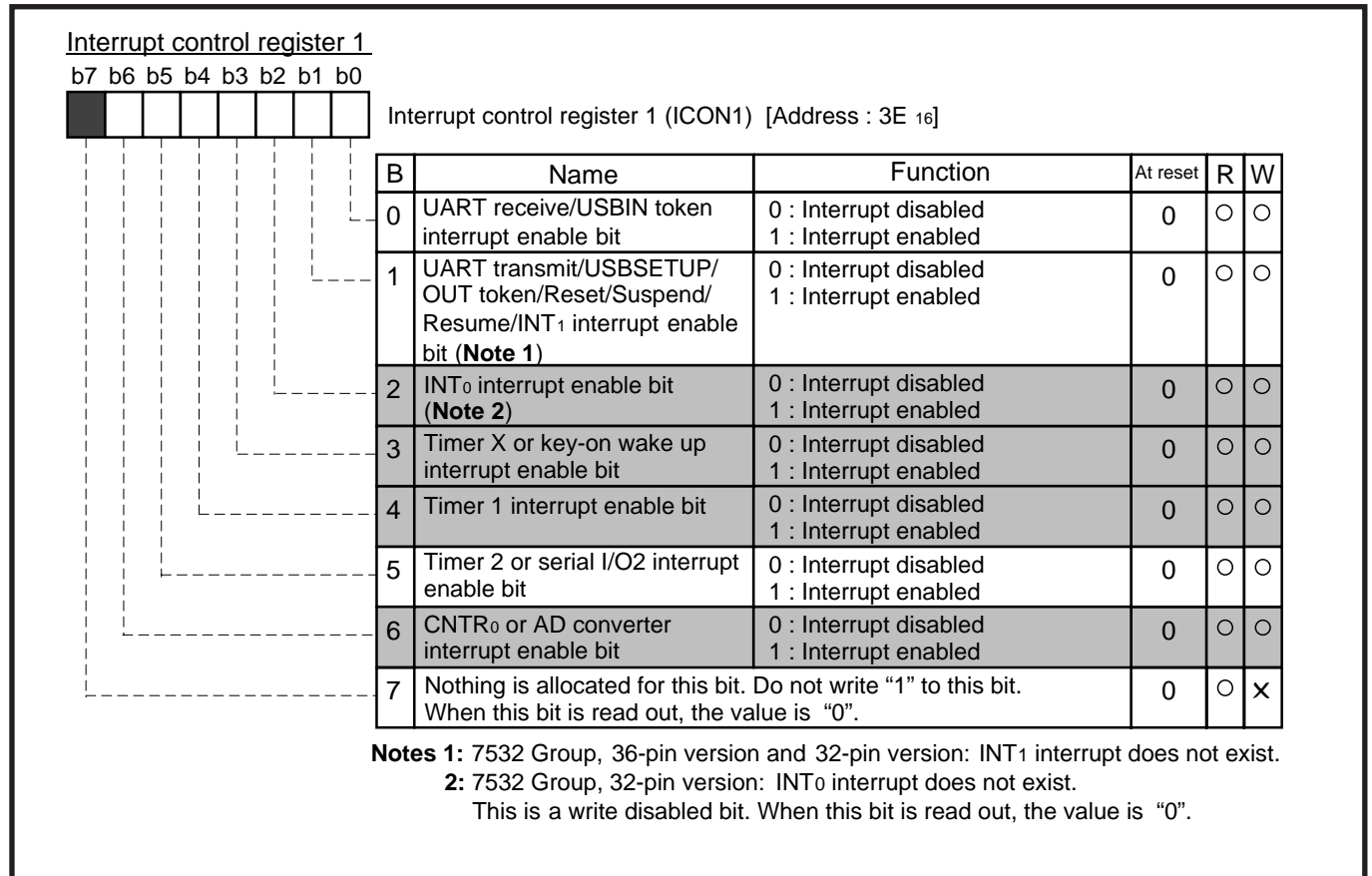


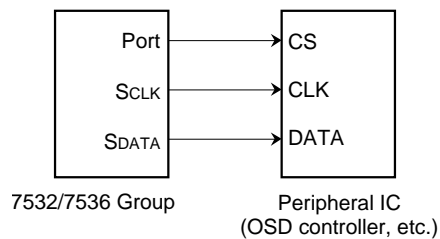
Fig. 2.3.11 Structure of Interrupt control register 1

## 2.3.3 Serial I/O connection examples

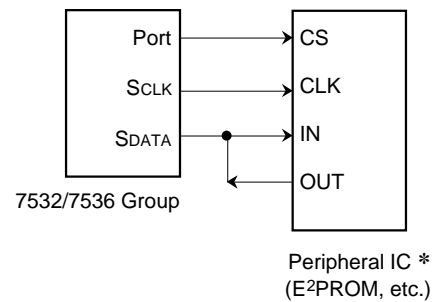
## (1) Control of peripheral IC equipped with CS pin

Figure 2.3.12 shows connection examples with a peripheral IC equipped with the CS pin. Each case uses the clock synchronous serial I/O mode.

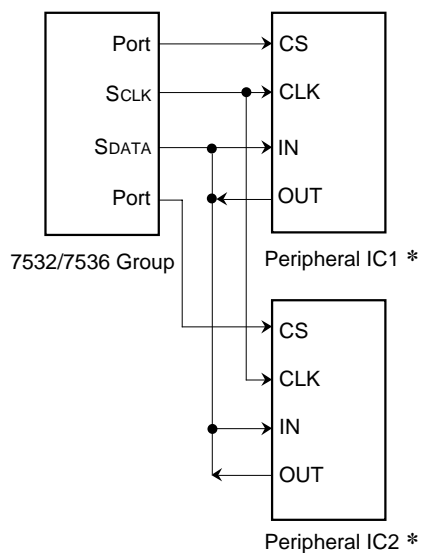
## (1) Only transmission



## (2) Transmission and Reception



## (3) Connection of plural IC



\* : Use the peripheral IC of which OUT pin has an N-channel open-drain output structure and which enters a high-impedance state while receiving data.

**Note:** "Port" means an output port controlled by software.

Fig. 2.3.12 Serial I/O connection examples (1)

# APPLICATION

## 2.3 Serial I/O

### (2) Connection with microcomputer

Figure 2.3.13 shows connection examples with another microcomputer.

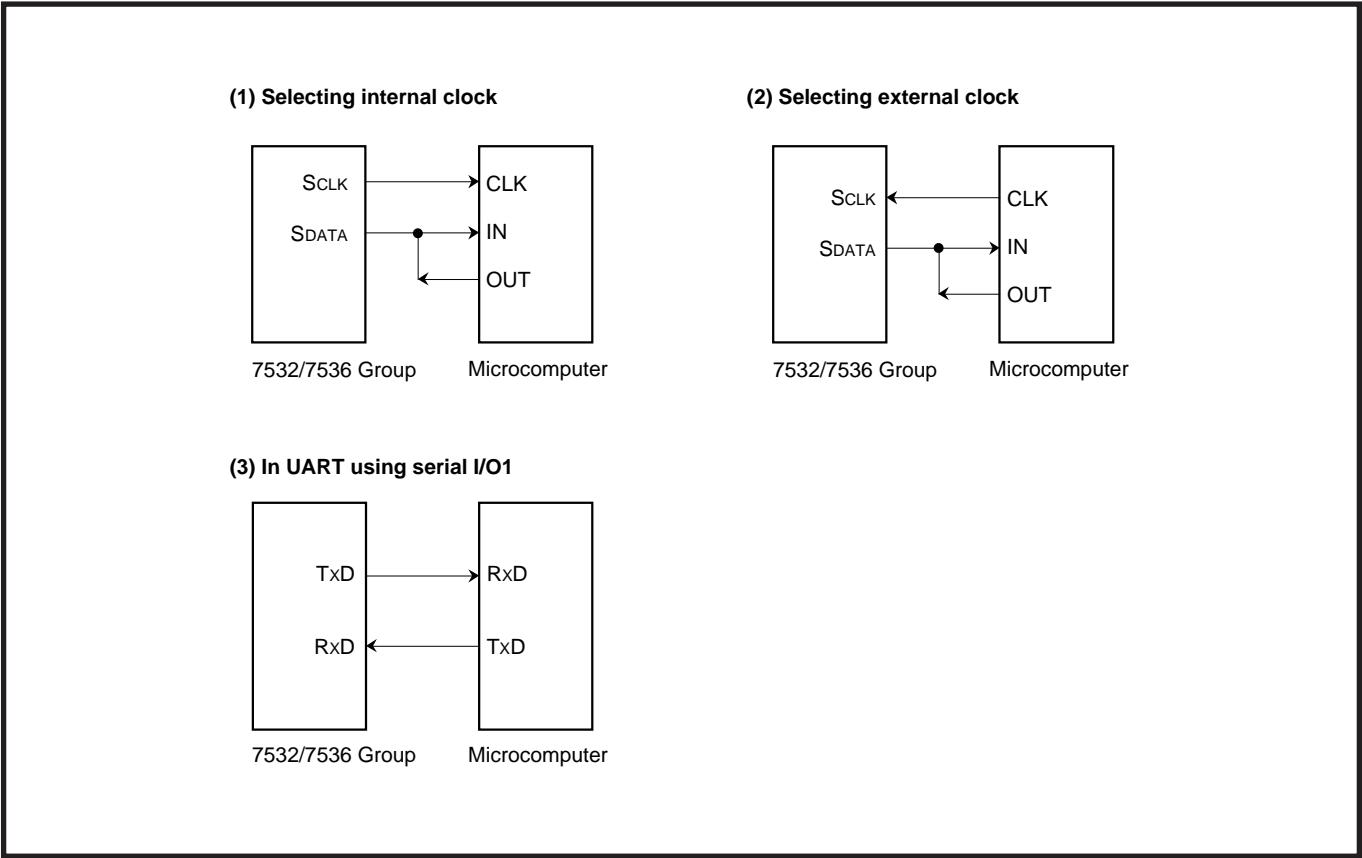


Fig. 2.3.13 Serial I/O connection examples (2)

### 2.3.4 Serial I/O transfer data format

The clock synchronous or the clock asynchronous (UART) can be selected as the serial I/O. Figure 2.3.14 shows the serial I/O transfer data format.

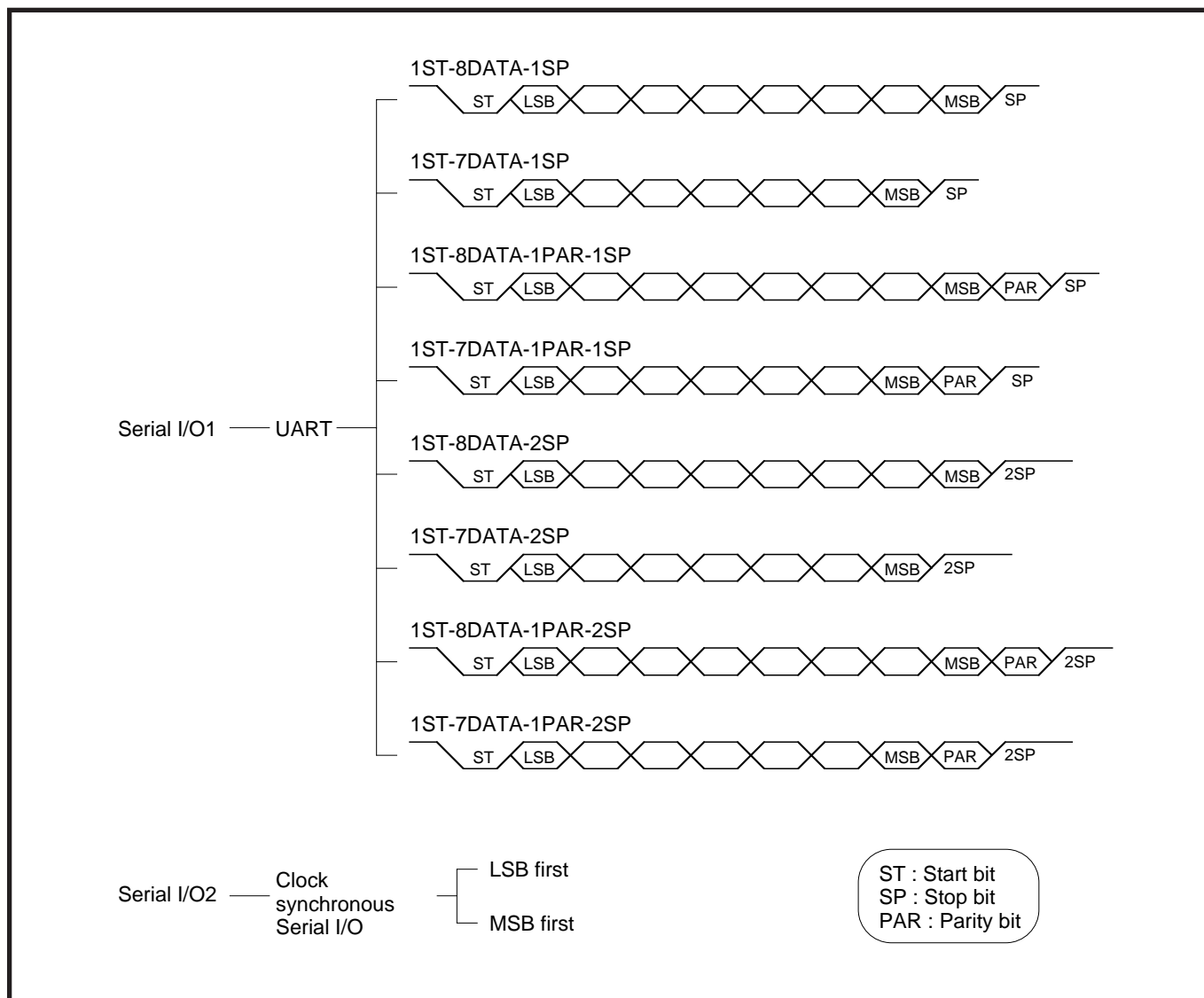


Fig. 2.3.14 Serial I/O transfer data format



# 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. Port P0<sub>0</sub> is used for communication control and outputs the quasi- $\overline{\text{SRDY}}$  signal.

The following explain an example using the serial I/O<sub>2</sub>. Figure 2.3.15 shows a connection diagram, and Figure 2.3.16 shows a timing chart.

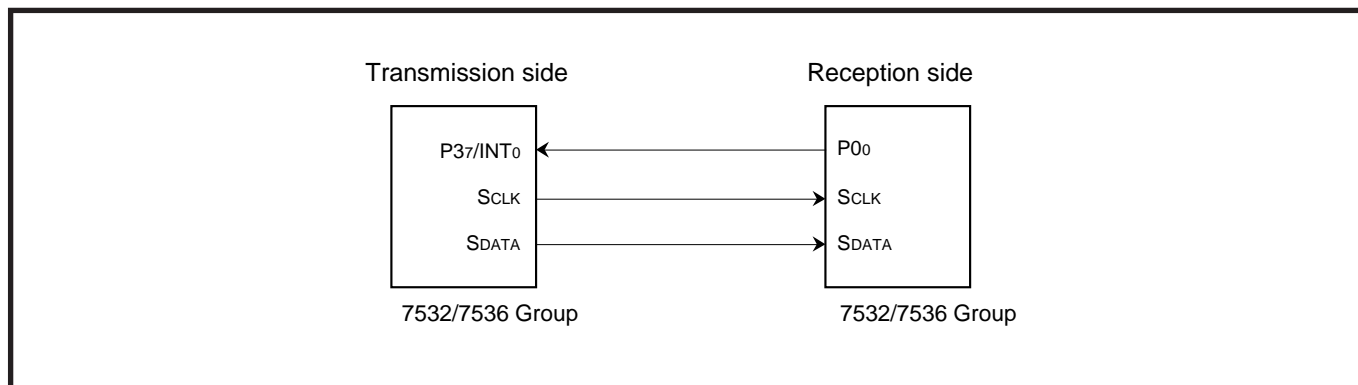


Fig. 2.3.15 Connection diagram

- Specifications :**
- The Serial I/O<sub>2</sub>, clock synchronous serial I/O, is used.
  - Synchronous clock frequency : 94 kHz;  $f(X_{IN}) = 6 \text{ MHz}$  divided by 64
  - Transfer direction : LSB first
  - The reception side outputs the quasi- $\overline{\text{SRDY}}$  signal at 2 ms intervals which the timer generates, and 2-byte data is transferred from the transmission side to the reception side.

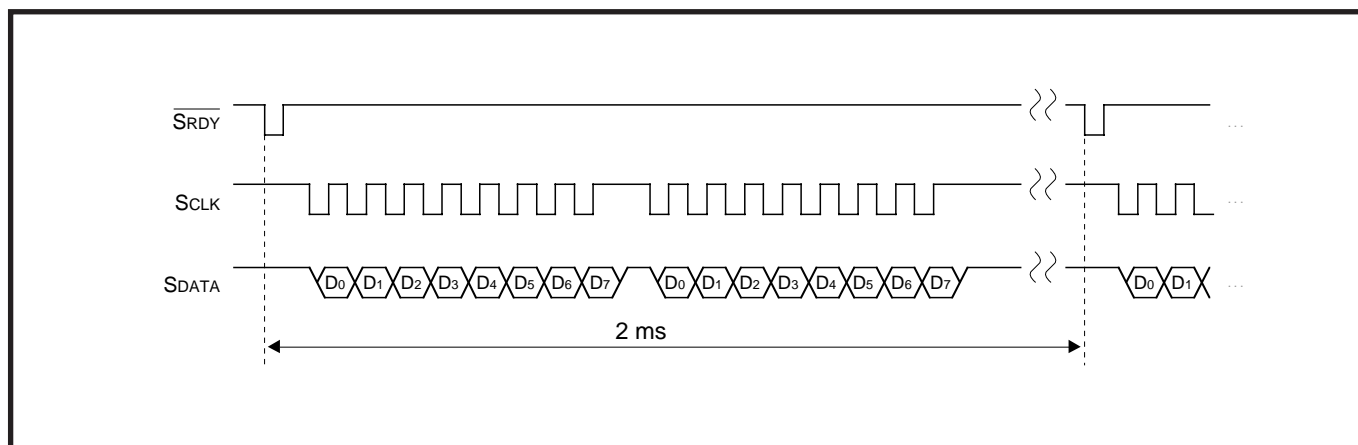


Fig. 2.3.16 Timing chart

Figures 2.3.17 and 2.3.19 show the registers setting relevant to the serial I/O2 and Figure 2.3.18 shows the transmission data setting of the serial I/O2.

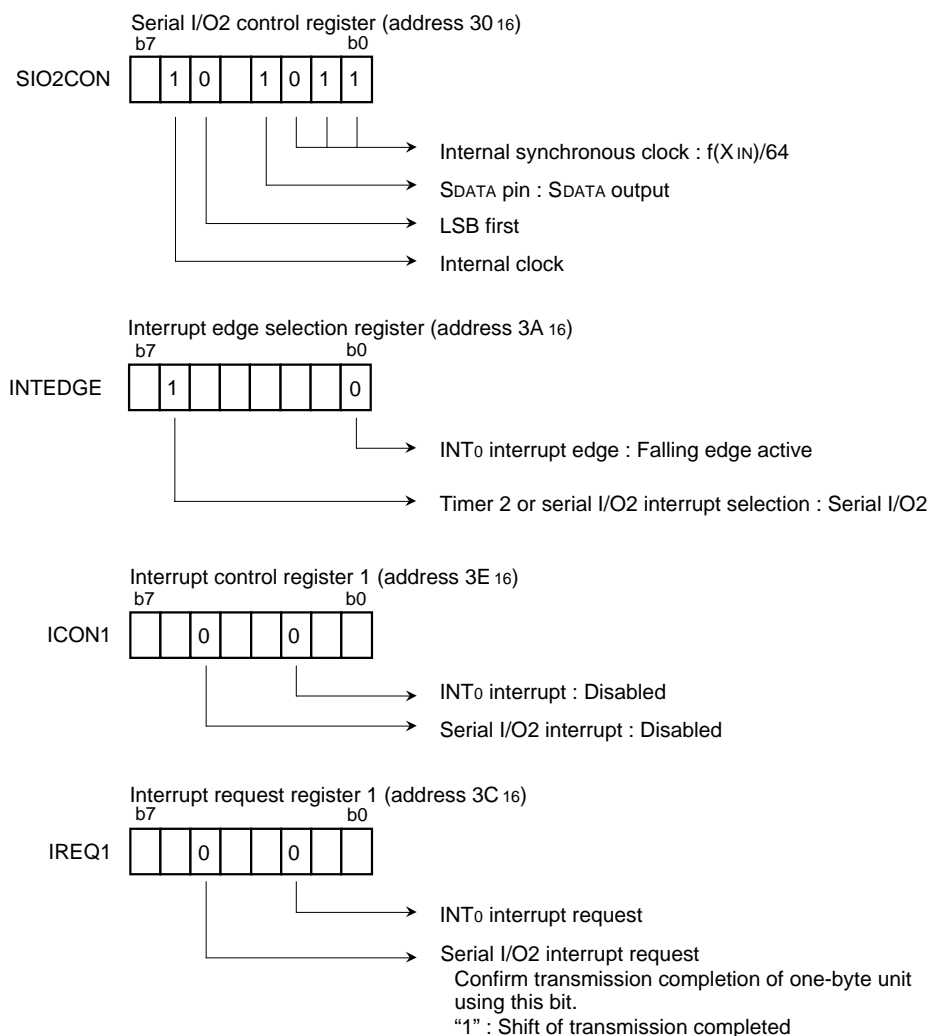


Fig. 2.3.17 Registers setting relevant to transmission side

# APPLICATION

## 2.3 Serial I/O

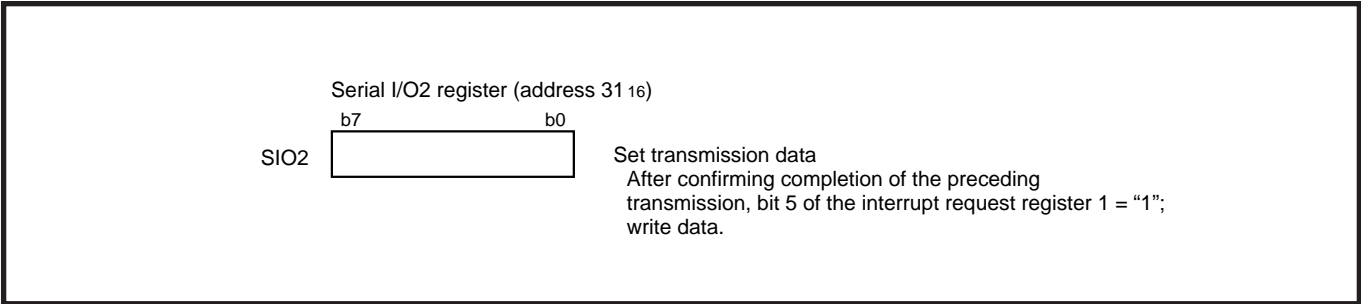


Fig. 2.3.18 Transmission data setting of serial I/O2

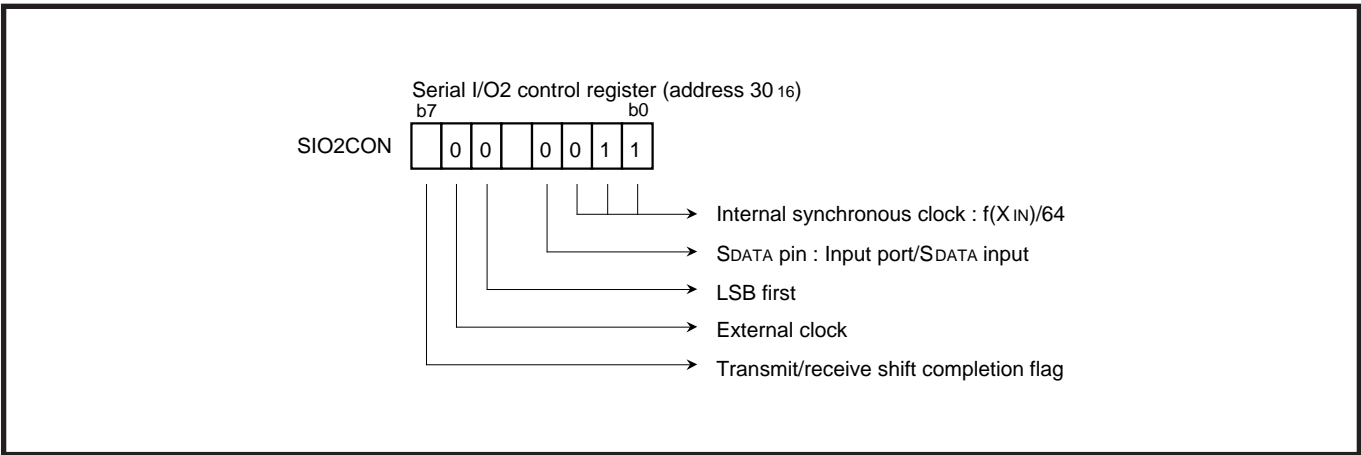


Fig. 2.3.19 Registers setting relevant to reception side

Figure 2.3.20 shows a control procedure of transmission side, and Figure 2.3.21 shows a control procedure of reception side.

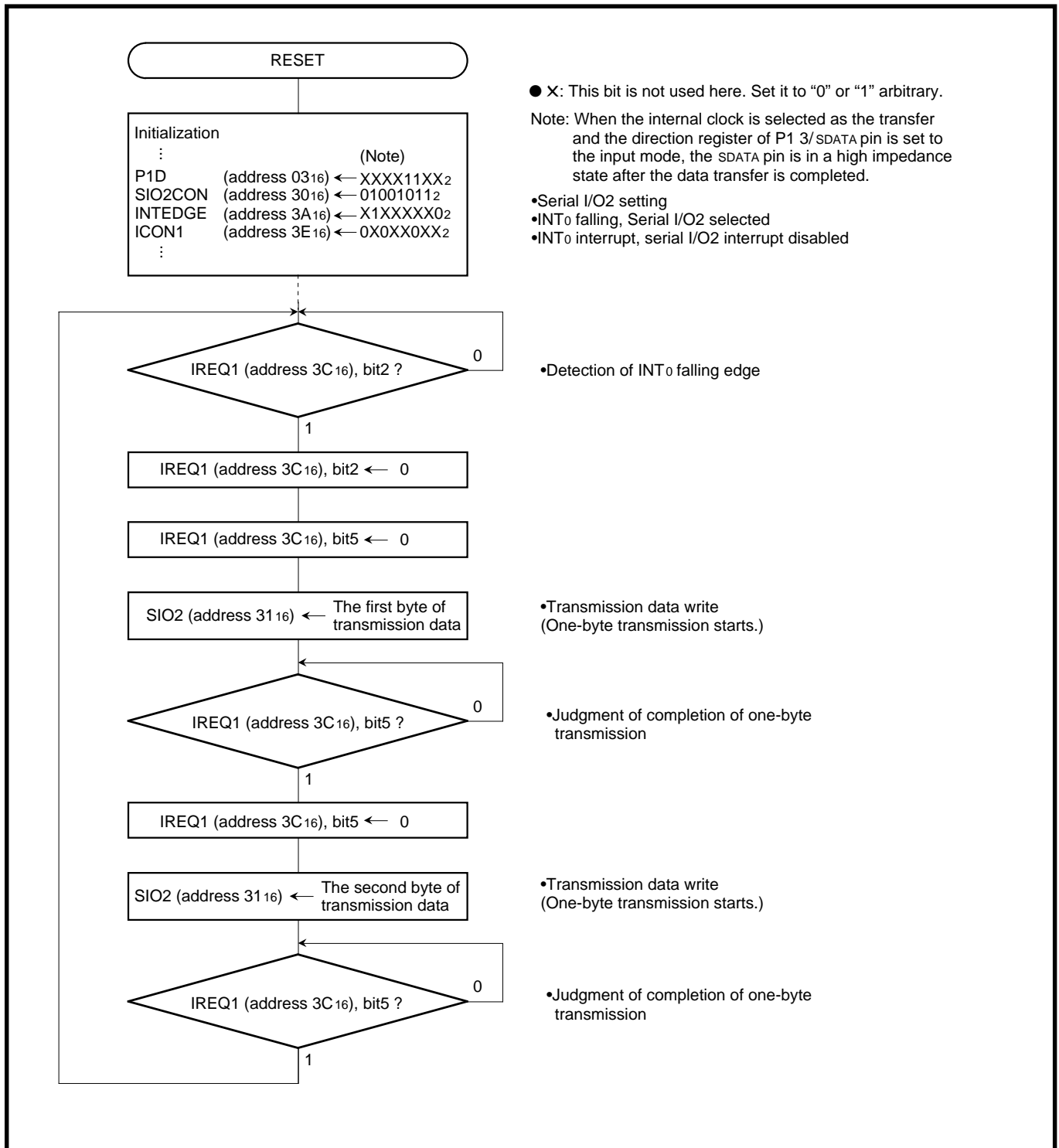


Fig. 2.3.20 Control procedure of transmission side

# APPLICATION

## 2.3 Serial I/O

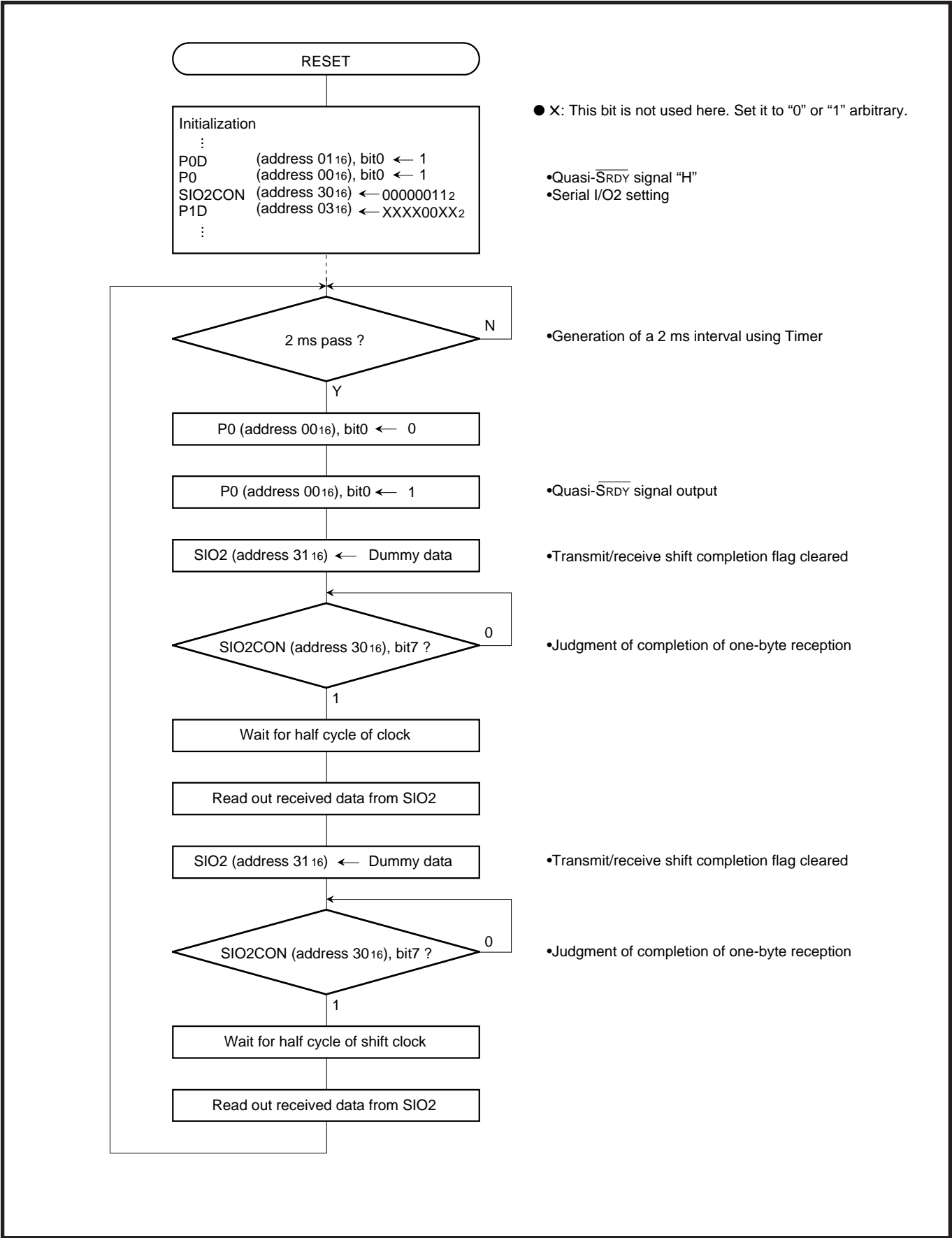
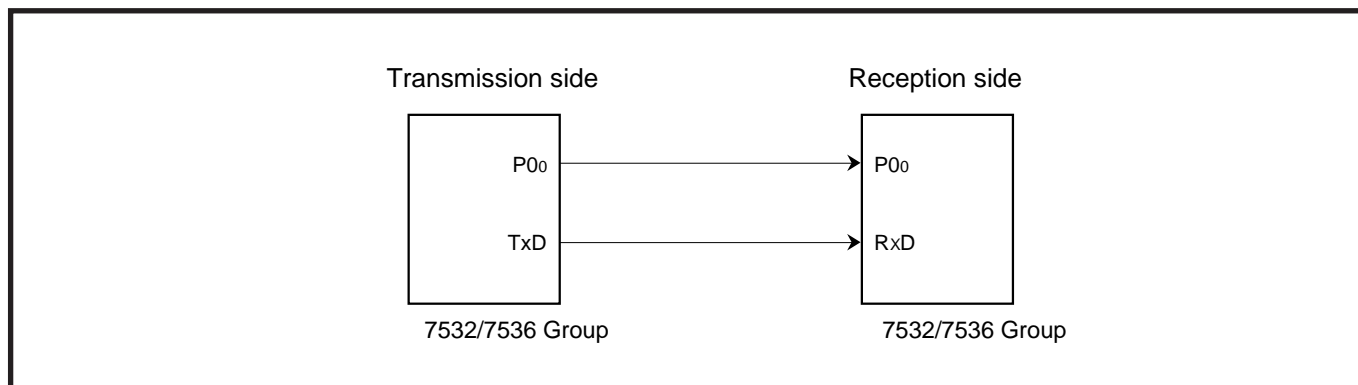


Fig. 2.3.21 Control procedure of reception side

### (2) Communication using asynchronous serial I/O, UART (transmit/receive)

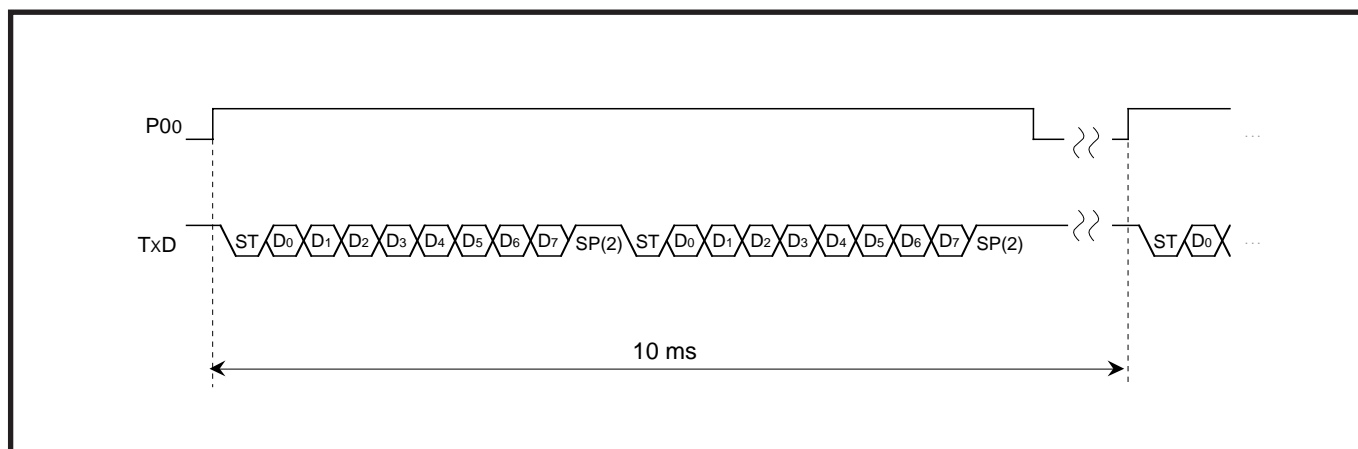
**Outline** : 2-byte data is transmitted and received, using the clock asynchronous serial I/O. Port P0<sub>0</sub> is used for communication control.

Figure 2.3.22 shows a connection diagram, and Figure 2.3.23 shows a timing chart.



**Fig. 2.3.22 Connection diagram**

- Specifications** :
- The Serial I/O1, asynchronous serial I/O, is used.
  - Transfer bit rate : 9600 bps;  $f(X_{IN}) = 4.9152 \text{ MHz}$  divided by 512
  - Communication control using port P0<sub>0</sub>; Port P0<sub>0</sub> output level is controlled by software.
  - 2-byte data is transferred from the transmission side to the reception side at 10 ms intervals which the timer generates



**Fig. 2.3.23 Timing chart**

# APPLICATION

## 2.3 Serial I/O

Table 2.3.1 shows a setting example of the baud rate generator (BRG) and transfer bit rate values; Figure 2.3.23 shows the registers setting relevant to transmission side; Figure 2.3.24 shows the registers setting relevant to reception side

**Table 2.3.1 Setting example of baud rate generator (BRG) and transfer bit rate values**

BRG count source (Note 1)	BRG set value	Transfer bit rate (bps) (Note 2)	
		At $f(X_{IN}) = 4.9152 \text{ MHz}$	At $f(X_{IN}) = 6 \text{ MHz}$
$f(X_{IN}) / 4$	255 ( $FF_{16}$ )	300	366.2109375
$f(X_{IN}) / 4$	127 ( $7F_{16}$ )	600	732.421875
$f(X_{IN}) / 4$	63 ( $3F_{16}$ )	1200	1464.84375
$f(X_{IN}) / 4$	31 ( $1F_{16}$ )	2400	2929.6875
$f(X_{IN}) / 4$	15 ( $0F_{16}$ )	4800	5859.375
$f(X_{IN}) / 4$	7 ( $07_{16}$ )	9600	11718.75
$f(X_{IN}) / 4$	3 ( $03_{16}$ )	19200	23437.5
$f(X_{IN}) / 4$	1 ( $01_{16}$ )	38400	46875
$f(X_{IN})$	3 ( $03_{16}$ )	76800	93750
$f(X_{IN})$	1 ( $01_{16}$ )	153600	187500
$f(X_{IN})$	0 ( $00_{16}$ )	307200	375000

**Notes** 1: Select the BRG count source with bit 0 of the serial I/O1 control register (address  $1A_{16}$ ).

2: Equation of transfer bit rate:

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

$m^*$ :  $m = 1$  in the case of bit 0 of the serial I/O1 control register (address  $001A_{16}$ ) = "0"

$m = 4$  in the case of bit 0 of the serial I/O1 control register (address  $001A_{16}$ ) = "1"

## Transmission side

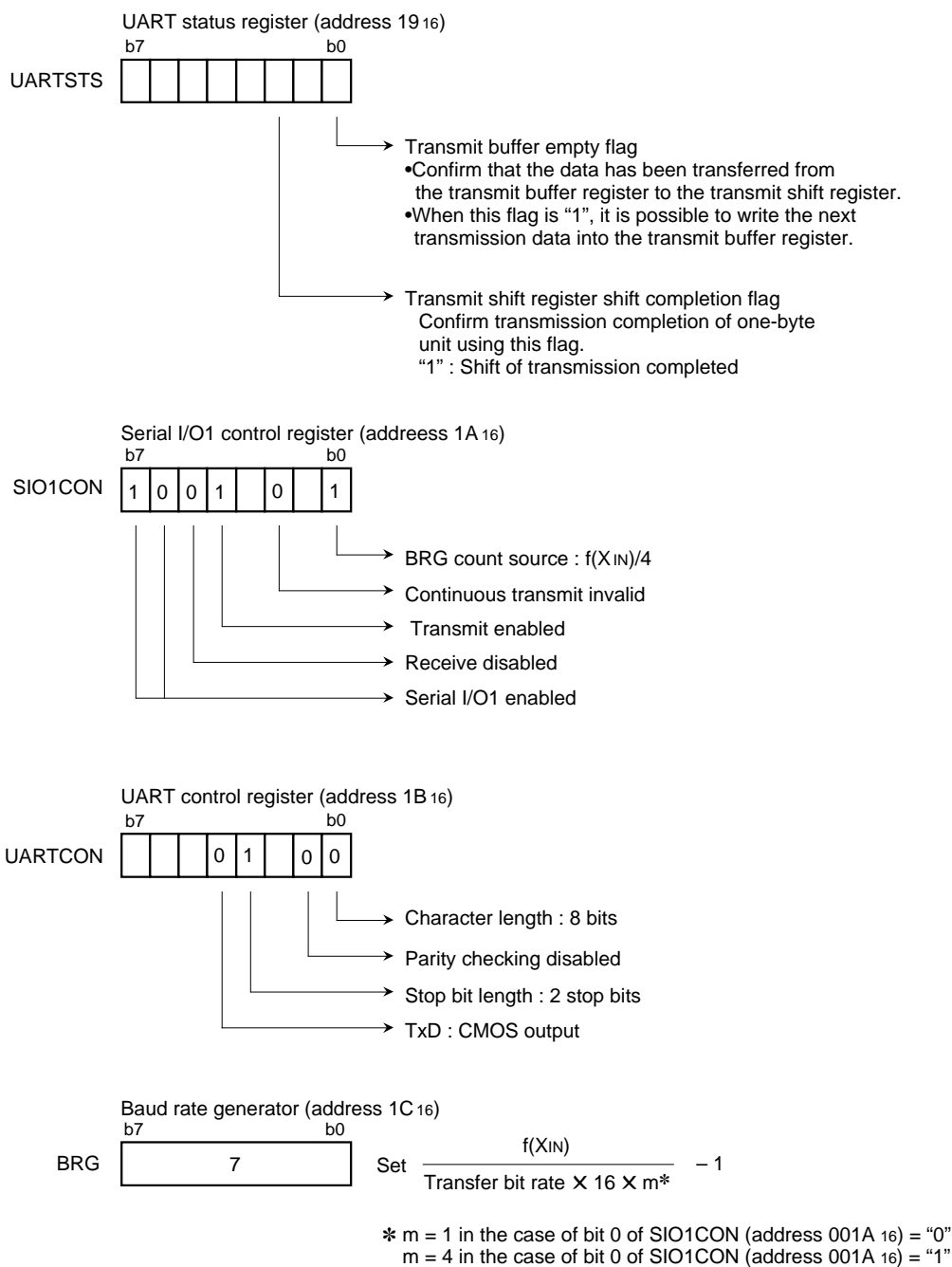


Fig. 2.3.24 Registers setting relevant to transmission side



# APPLICATION

## 2.3 Serial I/O

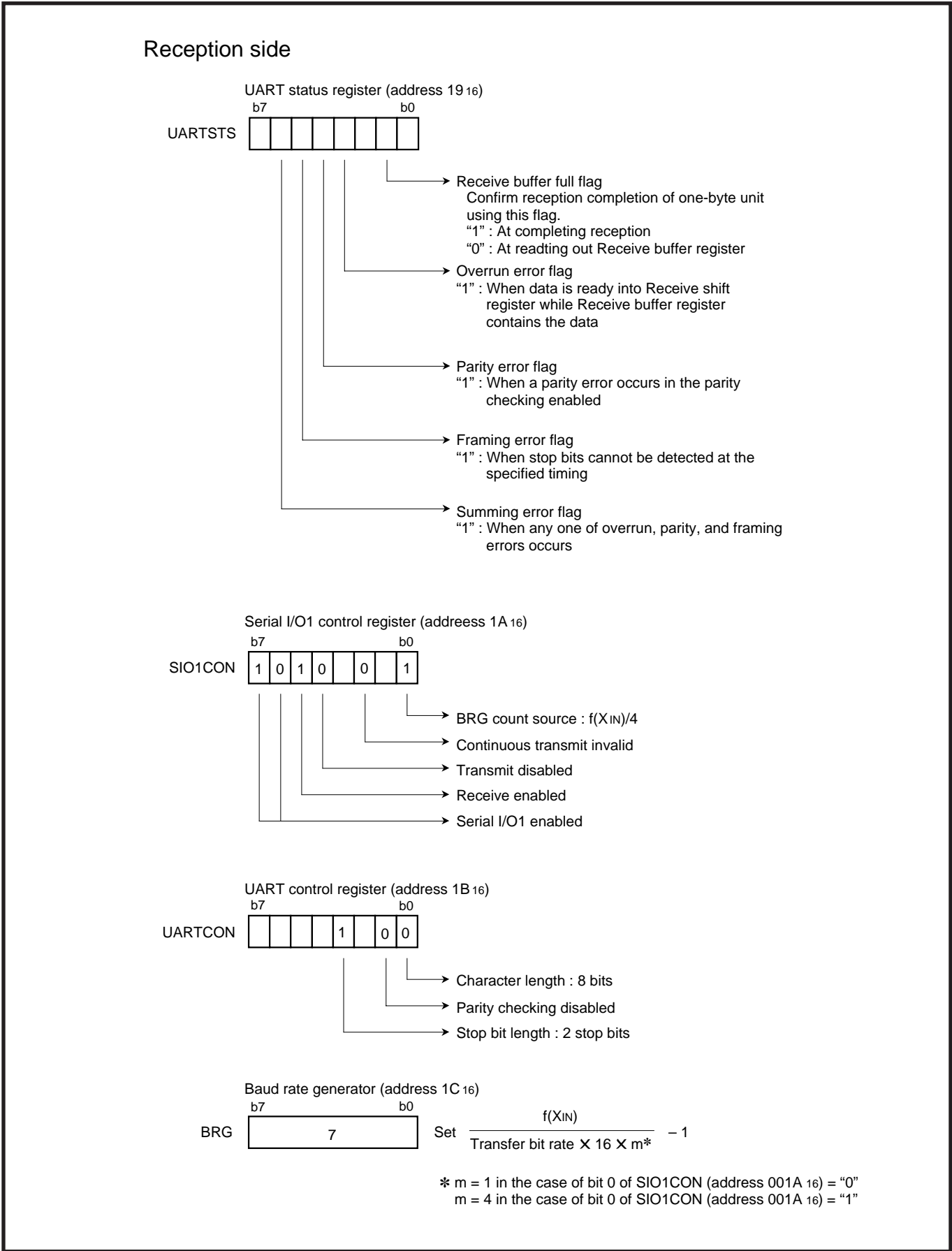


Fig. 2.3.25 Registers setting relevant to reception side

Figure 2.3.26 shows a control procedure of transmission side, and Figure 2.3.27 shows a control procedure of reception side.

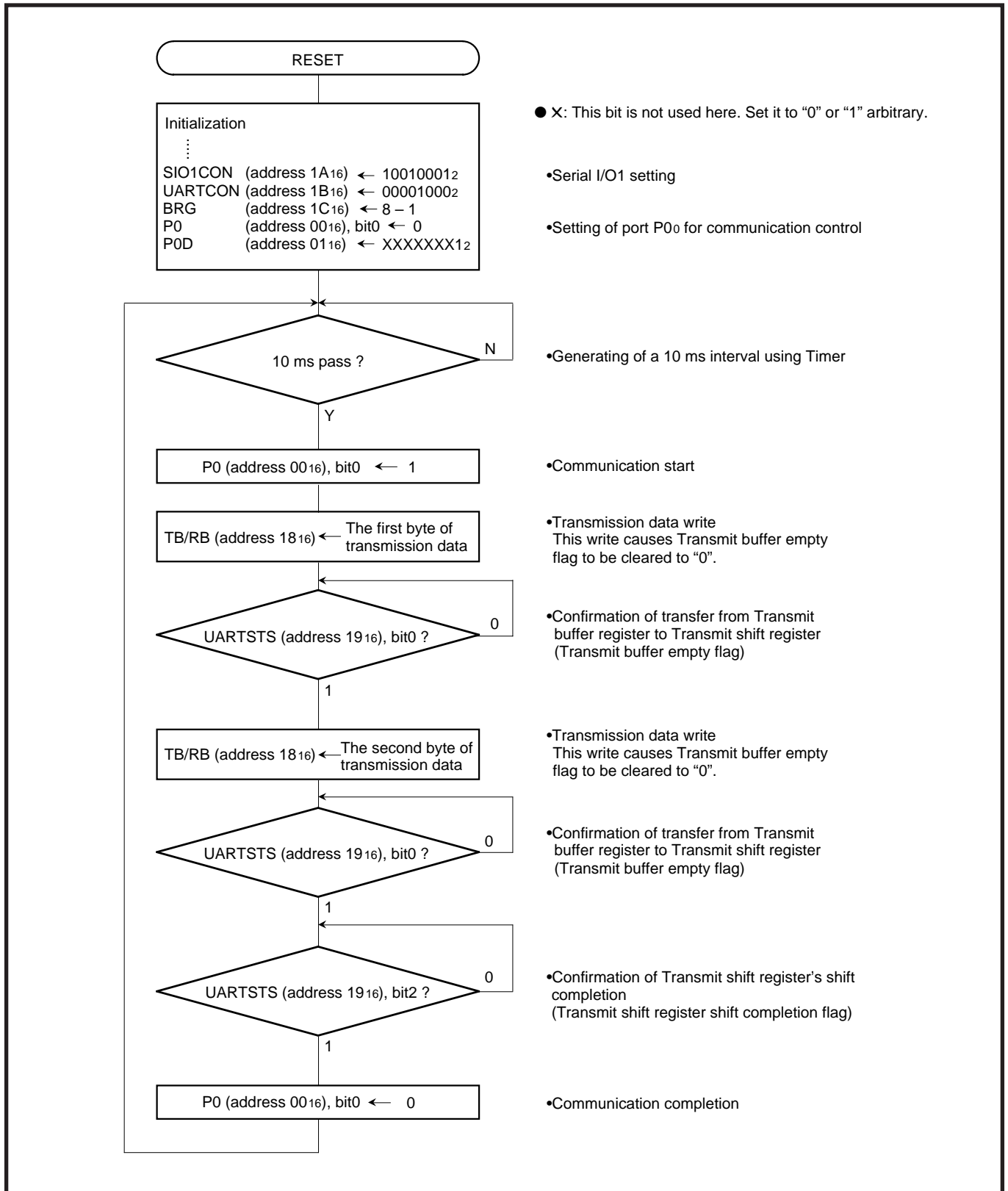


Fig. 2.3.26 Control procedure of transmission side

# APPLICATION

## 2.3 Serial I/O

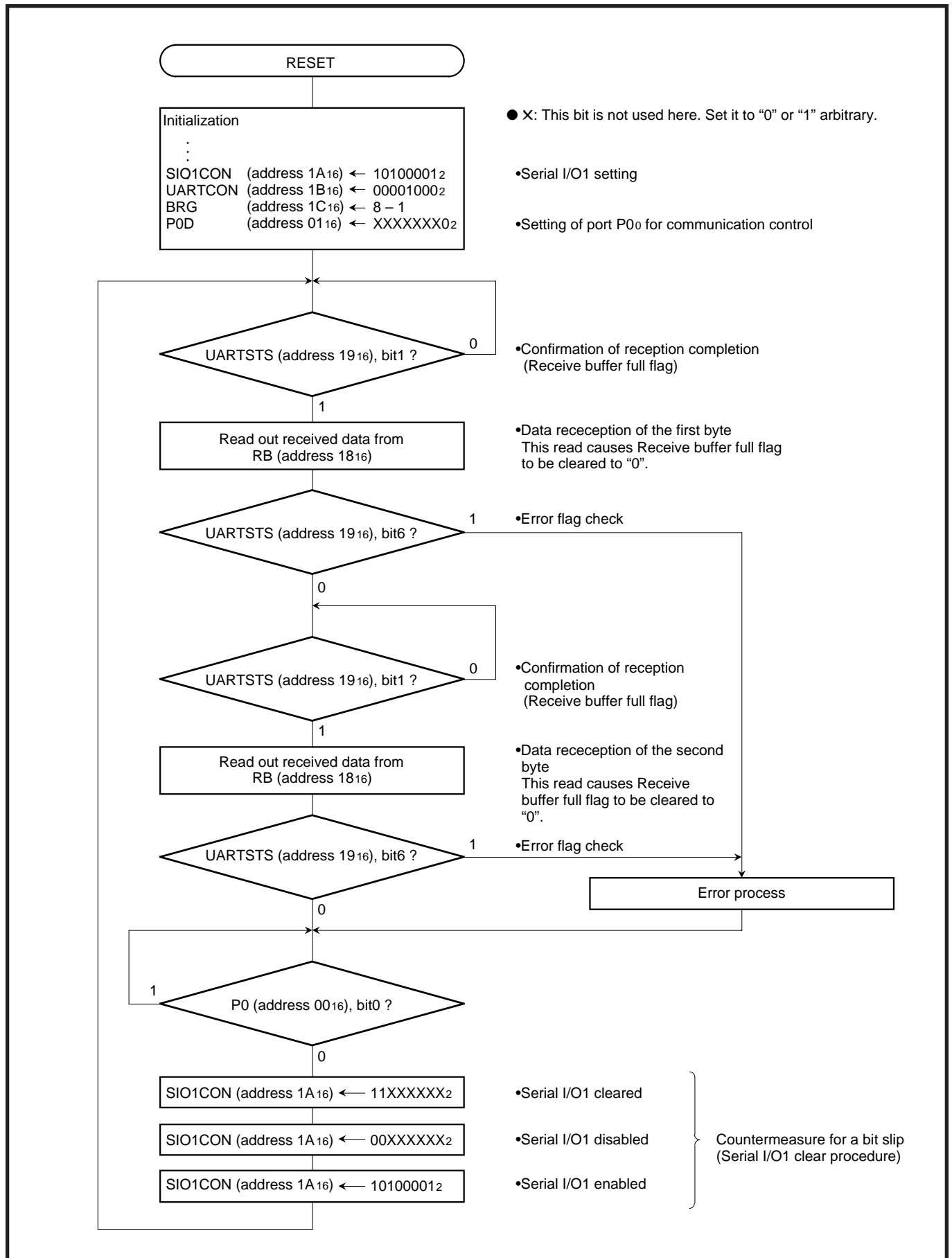


Fig. 2.3.27 Control procedure of reception side

## 2.3.6 Notes on serial I/O

## (1) Handling of clear the serial I/O1

When serial I/O1 is set again or the transmit/receive operation is stopped/restarted while serial I/O1 is operating, clear the serial I/O1 as shown in Figure 2.3.28.

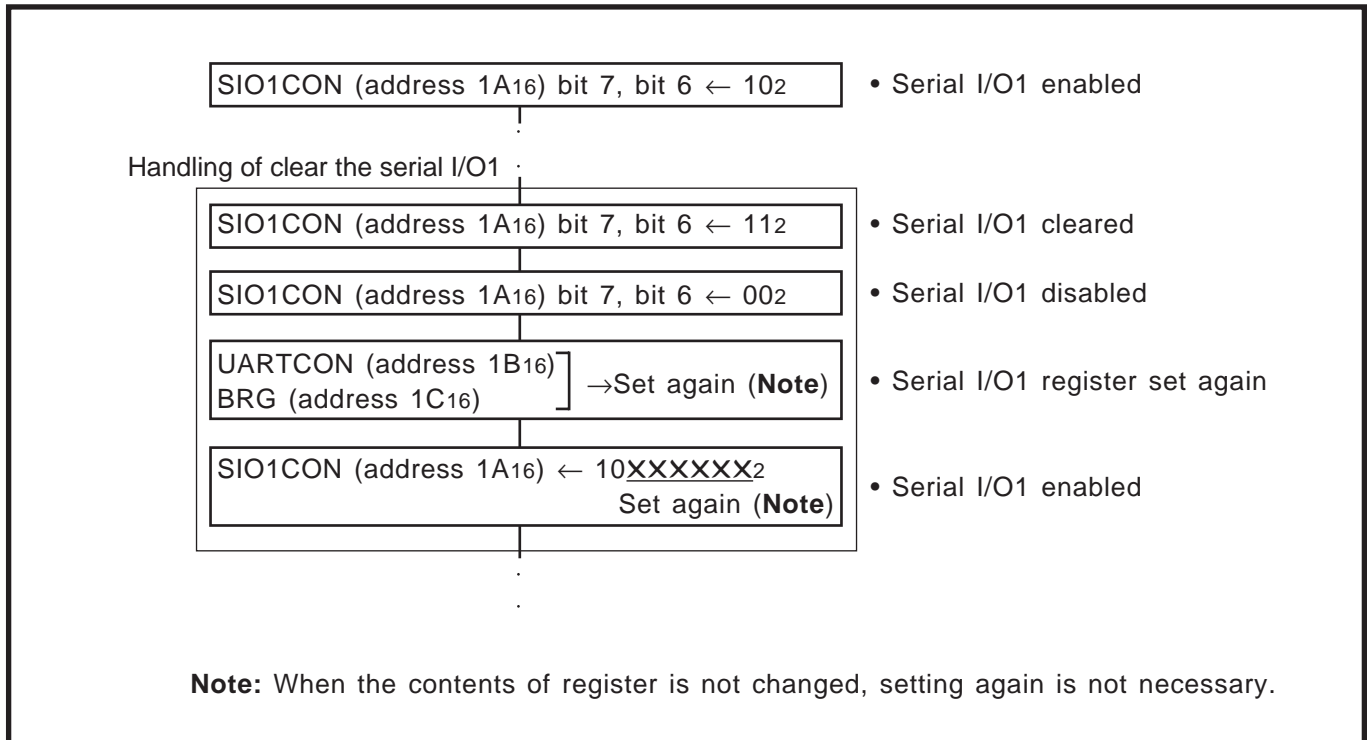


Fig. 2.3.28 Sequence of clearing serial I/O

## (2) 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.

## (3) Writing transmit data

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

## (4) Serial I/O2 transmit/receive shift completion flag

- The transmit/receive shift completion flag of the serial I/O2 control register is set to “1” after completing transmit/receive shift. In order to set this flag to “0”, write data (dummy data at reception) to the serial I/O2 register by program.
- Bit 7 of the serial I/O2 control register is set to “1” a half cycle (of the shift clock) earlier than completion of shift operation. Accordingly, when using this bit to confirm shift completion, a half cycle or more of the shift clock must pass after confirming that this bit is set to “1”, before performing read/write to the serial I/O2 register.

# APPLICATION

## 2.4 USB

### 2.4 USB

This paragraph explains operational outline, the registers relevant to the USB, the registers setting method, the application example for communication and notes on use.

#### 2.4.1 Outline of USB

7532/7536 Group have low-speed communication function (L.S.:Low Speed function) in accordance with USB Specification V.1.0 among USB (Universal Serial Bus) communication functions which are connection standard with PC peripherals.

In this section, the outline of USB communication function and the USB function of 7532/7536 Group are described.

#### (1) Transfer type

In present PC, 2 or more standards used for the connection with peripherals exist (RS-232C and Centronics, etc.).

USB tries to unite all those communication standards.

The standard of USB has the host side (PC,Hub) which controls connected peripherals and the connected peripherals side (device).

The following 2 types of communication standards exist depending on the data amount treated on the peripherals side.

- **High-speed communication (F.S.: Full Speed function):**

Communication standard with fast transfer rate (12 Mbps)

This standard is used for peripherals that a large amount of data transfer is required for one time (image and voice, etc.).

- **Low-speed communication (L.S.: Low Speed function):**

Communication standard with slow transfer rate (1.5 Mbps)

This standard is used for peripherals with a small amount of data (keyboard and mouse, etc.).

This communication standard depends on the kind of peripherals.

The transfer type for each peripheral is decided.

Table 2.4.1 shows the transfer types of USB.

**Table 2.4.1 Transfer types of USB**

Transfer type	L.S./F.S	Operation
Control	L.S./F.S	This is used when setting up and for all devices common.
Interrupt	L.S./F.S	This is used when transferring a small amount of data in real time.
Bulk	F.S	This is used when transferring a large amount of data in no real time.
Isochronous	F.S	This is used when transferring a large amount of data in real time.

L.S.: Low-Speed function, F.S.: Full-Speed function

The 7532/7536 Group has one low-speed communication function, and the control transfer and interrupt transfer can be used in Table 2.4.1

**(2) Communication sequence**

The control transfer and the interrupt transfer have a different communication sequence, respectively. The control transfer is used when setting up all devices common and communicates combining 3 types of stages in one processing.

The communication starts first in Setup Stage, and Data Stage of the content is executed, and then, one processing of the communication sequence is completed by executing Status Stage.

Data can be set from host to device through this sequence (=Control Write), and the result can be read out to host from device (=Control Read).

Use endpoint (ENDP) "0" in the control transfer.

The interrupt transfer is used when transferring a small amount of data in real time.

There is no stage unlike the control transfer.

Only when the host requests data (Token=IN), the device can transmit data.

Use the endpoint in the interrupt transfer excluding "0" ("1" is set for the 7532/7536 Group).

Figure 2.4.1 shows the communication sequence of USB.

Control Transfer: ENDP (endpoint) = 0

● Control Read

Stage

Setup Token = SETUP → Data → Handshake 1 transaction

Data Token = IN → Data → Handshake 1 or more transaction

Status Token = OUT → Data → Handshake 1 transaction

● Control Write

Stage

Setup Token = SETUP → Data → Handshake 1 transaction

Data Token = OUT → Data → Handshake 1 or more transaction

Status Token = IN → Data → Handshake 1 transaction

● No-data Control

Stage

Setup Token = SETUP → Data → Handshake 1 transaction

Status Token = IN → Data → Handshake 1 transaction

Interrupt Transfer: ENDP = 1

Token = IN → Data → Handshake

**Note:** The shaded parts show the case when the device transmits data to the host.

**Fig. 2.4.1 Communication sequence of USB**

# APPLICATION

## 2.4 USB

### (3) Packet type

The host side controls all the communications in USB.

Basically, the host gives the device the instruction of the communication processing which the host side executes (= Token), executes data transmit/receive (= Data), and indicates the completion of the communication is shown at the end (= Handshake).

These communication processings are executed in order by each one unit which has the data structure (= Packet format), and one processing (= Transaction) is completed.

The content of processing can be identified according to PID (Packet Identifier field) which is one unit of data (Field) which composes each packet.

Not only the content of processing, but also the data structure in the packet can be identified by this PID.

Table 2.4.2 shows the packet type of USB.

**Table 2.4.2 Packet types of USB**

Packet type	Transmitter	Operation
SOF (Start of Frame)	Host	Packet indicating the top of frame (1 ms) including all transfer types
Token	Host	Packet indicating the processing to execute
Data	Host/Device	Packet indicating the transmit/receive data for processing shown by token
Handshake	Host/Device	Packet indicating the result of communication processing

In 7532/7536 Group, the token, data, and the handshake of packet types in Table 2.4.2 can be controlled by software.

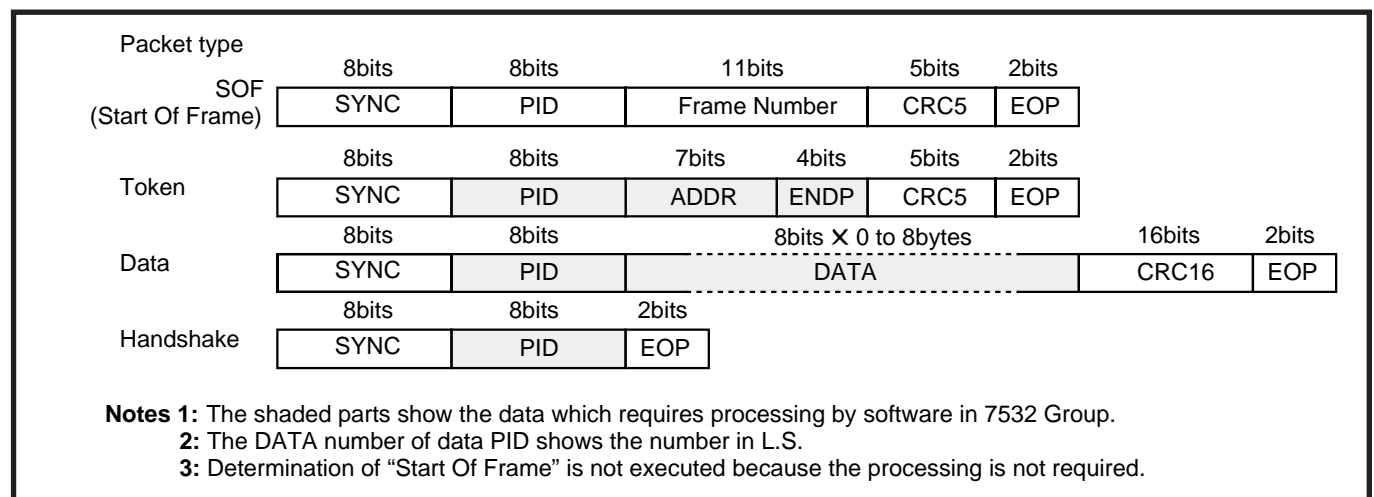
### (4) Packet structure

As for the packet type of USB, the data and the structure are different by PID.

Each packet includes the following data shown with PID for which the control is required by 7532/7536 Group;

- token: the receiver to communicate (ADDR) and transfer type (ENDP)
- data: execution (DATA) of the processing ordered by the token
- handshake: the completion of the communication

Figure 2.4.2 shows the data structure of the packet.



**Fig. 2.4.2 Data structure of USB packet**

In 7532/7536 Group, the PID, ADDR, ENDP and DATA of packet structure data in Figure 2.4.2 can be controlled by software.

**(5) Data structure**

Data which composes the communication of USB transmits and receives data that the structure number of bits is different as shown in Figure 2.4.2 continuously by the LSB first.

Basically, the contents except SYNC which is synchronizing signal and EOP which is the completion signal are treated as data.

Accordingly, the difference of the number of bits can be detected by forecasting and determining the following data from the content of PID.

Table 2.4.3 shows the content of data which composes the packet.

**Table 2.4.3 Data structure of USB packet**

Data name	Symbol	Structure	Operation
Synchronize	Sync	8 bits	Synchronized signal to communicate
Packet Identifier	PID	8 bits	Data indicating processing of packet
Frame Number	–	11 bits	Data to control the frame during communication by time
Address	ADDR	7 bits	Data to confirm the transmit destination of packet and notify from transmitter
Endpoint	ENDP	4 bits	Data indicating transfer type used by device
Data	DATA	8 bits X 0 to 8 bytes	Data to be used when the processing specified by PID is executed
Token CRCs	CRC5	5 bits	Data to check error when PID is token and SOF
Data CRCs	CRC16	16 bits	Data to check error when PID is DATA
End Of Packet	EOP	2 bits	Data indicating the completion of packet

**Note:** The DATA numbers in L.S. are shown in this table.

PID is classified into 3 kinds in the structure data of Table 2.4.3.

Token is used to give the device the instruction of communication processing, and report on the processing of the following stages. It is only a host to be able to issue the token.

Data is used to transmit and receive data which is the content of the instruction of the token, and execute the processing in the stage. It is a host to be able to issue the data when the token is SETUP and OUT, and it is a device when the token is IN.

Handshake indicates the completion of the communication at the end.

It is a host to be able to issue the handshake when the token is IN, and it is a host when the token is SETUP and OUT.

Table 2.4.4 shows PID.

**Table 2.4.4 PID**

Packet type	PID name	Bit structure (bits 3 to 0)	Processing
Token	SETUP	1101 <sub>2</sub>	The processing is reported by host to device
	IN	1001 <sub>2</sub>	Data transmit is requested from host to device
	OUT	0001 <sub>2</sub>	Data receive is requested from host to device
	SOF	0101 <sub>2</sub>	Top of frame is indicated by host to device
Data	DATA0	0011 <sub>2</sub>	The state that sequence bit of transmit/receive data is even is indicated
	DATA1	1011 <sub>2</sub>	The state that sequence bit of transmit/receive data is odd is indicated
Handshake	ACK	0010 <sub>2</sub>	Normal completion of communication is reported
	NAK	1010 <sub>2</sub>	The state that device is waiting for communication is reported
	STALL	1110 <sub>2</sub>	Completion error of communication is reported
Special	PRE	1100 <sub>2</sub>	Communication to the L.S. device which has low-priority is enabled

In 7532/7536 Group, data except SOF and PRE of PID can be controlled by software.



# APPLICATION

## 2.4 USB

### (6) USB special signal

The host side has function to control the state of the device side, and the device side has the function to transmit the state to the host side and other devices on USB communications line by the signal besides the above-mentioned data transfer.

Usually, the transfer confirmation [Keep alive, Idle: only for L.S.] is performed on the USB communication line regardless of data transmit/receive. The contents of transfer confirmation are as follows;

- host confirms the connection of device on the communication line
- device confirms the normal data transfer of host.

The confirmation transfer is transmitted by host side for each frame (1ms/frame) which is the basic time unit of the USB transfer.

When the host stops all the device, the host informs device of the stop of functions by suspending the confirmation transfer for 3 ms or more.

The signal to stop the function is called "Suspend".

The stopped device can be returned by 2 methods.

The return is performed basically when there is a change in the device in the stopped state.

One is a method (change on communications line) to return to a normal state by restarting the data communication which is suspended before suspend.

The signal to return is called "Resume".

The other is a method (change on the device) to return to a normal state by the change in an external input of the device.

The host can inform all the other connected devices of the return of the device by the change in an external input by outputting K state signal of 10 to 15 ms.

The signal to activate for other devices is called "Remote wake up".

When the SE0 signal of 2.5  $\mu$ s or more is input on communications line regardless of the state of the stop/start of the function of the device side, the packet, and the stage processing, the device makes all states concerning the USB function initial state.

The signal to initialize is called "Reset".

Table 2.4.5 shows a special signal of USB.

**Table 2.4.5 Special signal of USB**

Signal type	Operation	Signal format
Suspend	Stop of all device function	No data transfer for 3 ms
Resume	Return of device function	K state input/reset input in the suspend state
Reset	Initialization of USB setting	SE0 input for 2.5 $\mu$ s or more
Remote wake up	Report of return to other devices	K state output for 10 to 15 ms

**(7) USB interface**

In 7532/7536 Group, the USB interface divides one communications line by 2 depending on the contents of data.

Figure 2.4.3 shows the interface of USB.

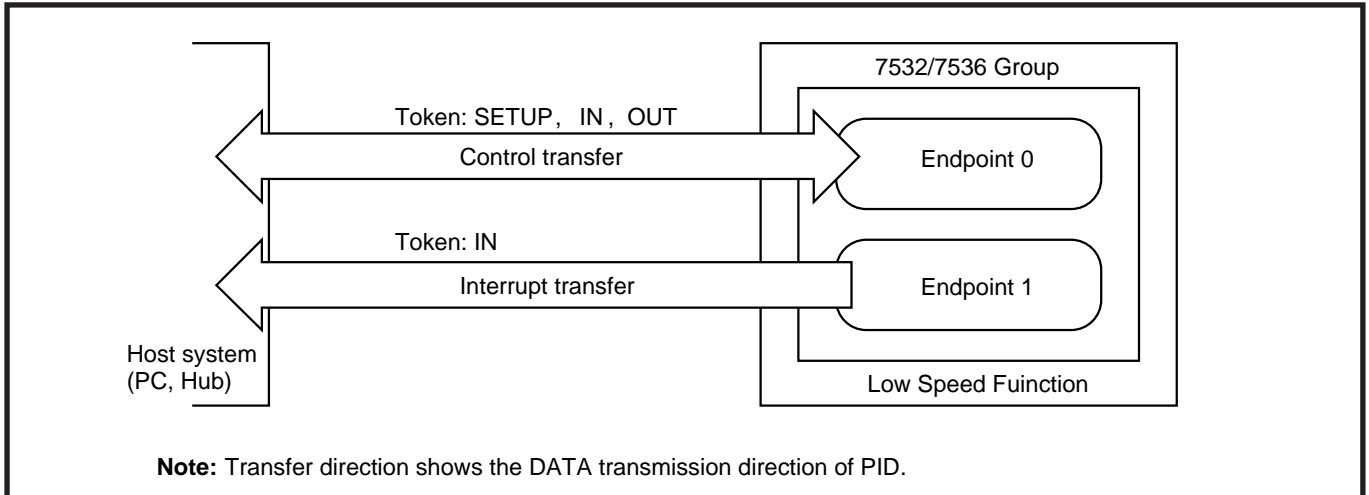


Fig. 2.4.3 USB (L.S.) interface

**(8) System configuration of USB**

In the system configuration used with the low-speed communication device of USB, the host side can recognize the connection of the low-speed communication device by pull-up the D-pin with the voltage of 3.0 to 3.6 V and the resistor of 1.5 k $\Omega$  of communications lines.

Figure 2.4.4 shows the example of connecting USB [L.S.].

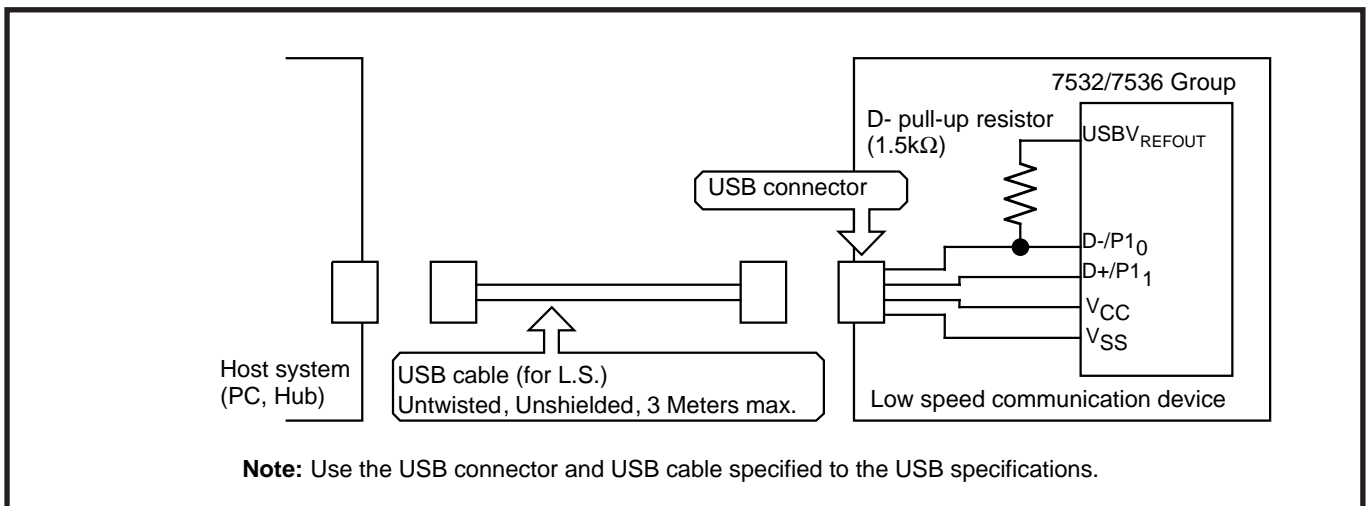


Fig. 2.4.4 USB (L.S.) connection example

# APPLICATION

## 2.4 USB

### 2.4.2 Memory map

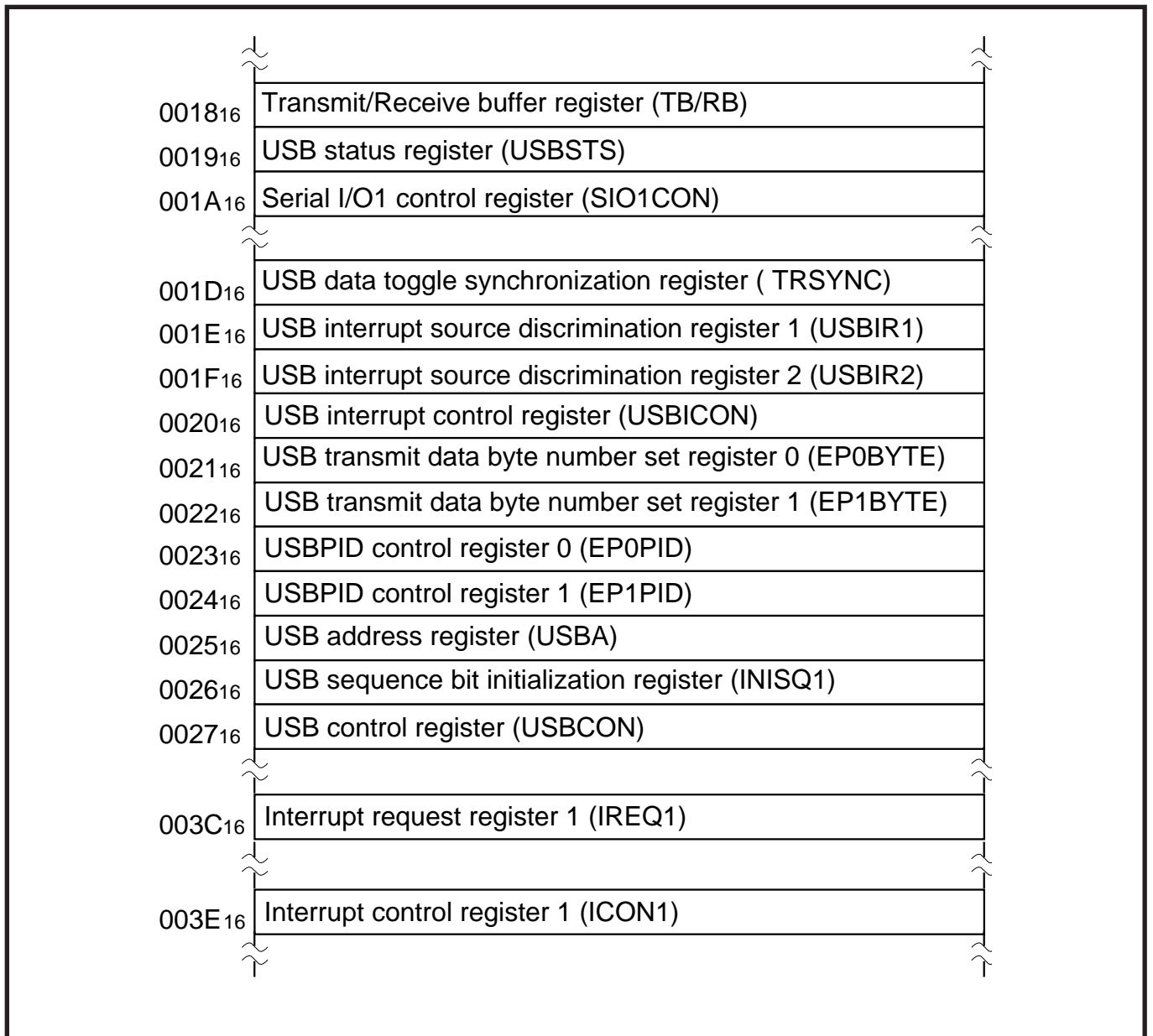


Fig. 2.4.5 Memory map of registers relevant to USB

### 2.4.3 Relevant registers

In this section, the contents of control are described as follows;

- Transmit/Receive buffer
- Setting to enable/disable of each function including USB communication and interrupts
- Determination of USB communication error occurrence
- Setting to stage and handshake by unit of endpoint
- Auto-determination of self-address of USB device

Figure 2.4.6 shows the description of the register structure, and Figure 2.4.7 to Figure 2.4.10 show the register structures relevant to USB.

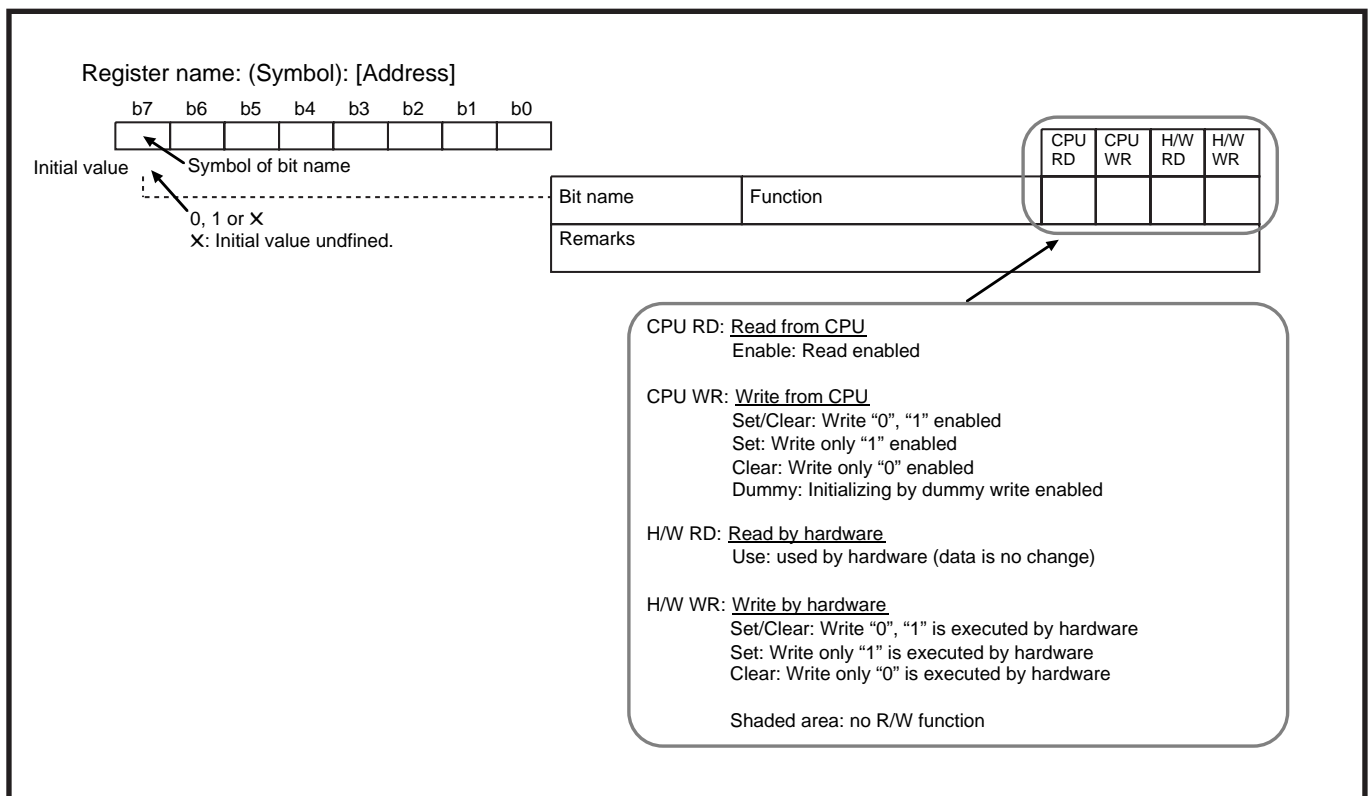


Fig. 2.4.6 Description of the register structure

# APPLICATION

## 2.4 USB

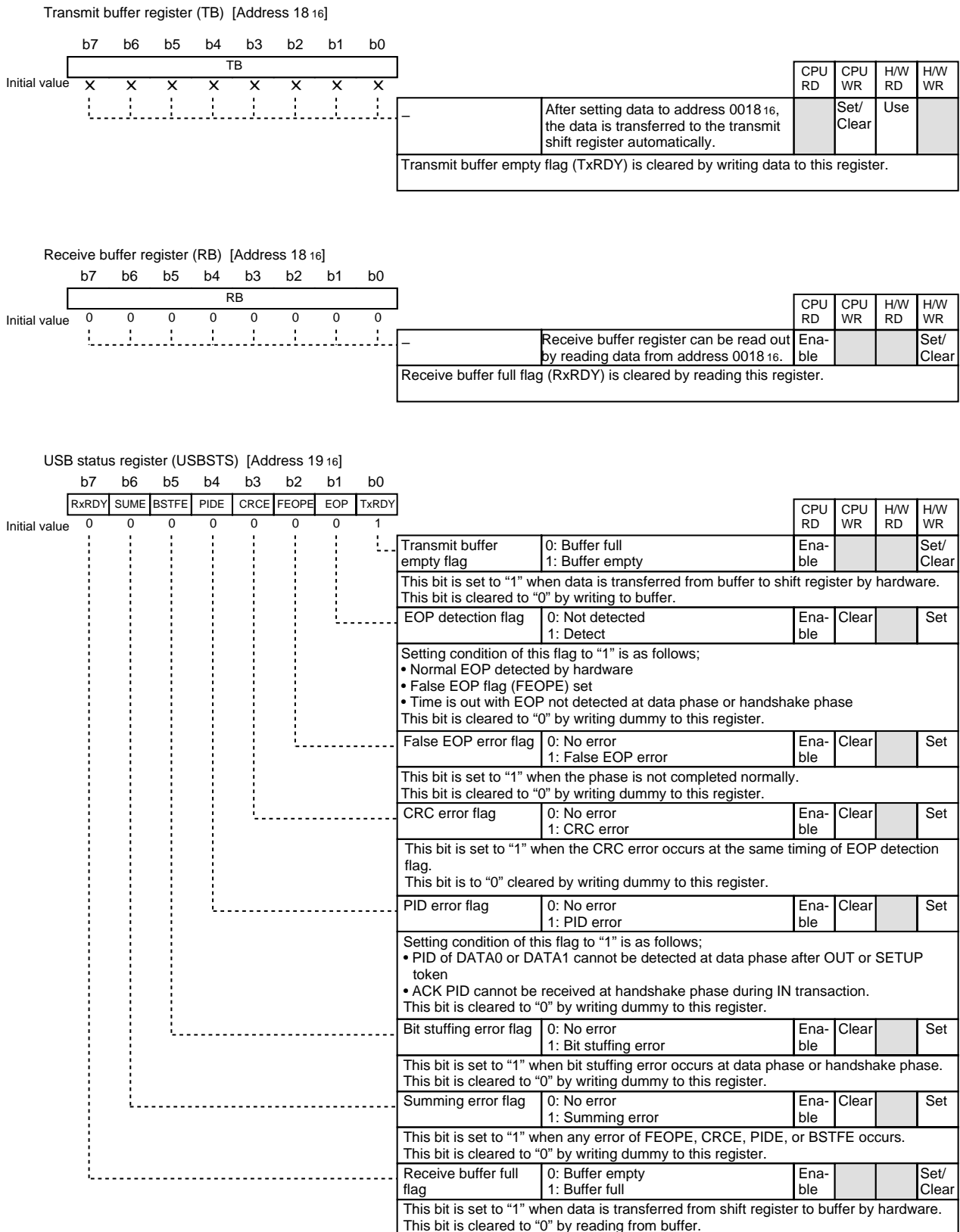


Fig. 2.4.7 Register structures relevant to USB (1)

The diagram shows the SQTGL register with bits b7 to b0. Bit b7 is labeled 'SQTGL'. Below the register, the 'Initial value' is shown as 0. A dashed line connects the 'Initial value' to the 'Sequence bit toggle flag' field in the control logic table.

CPU RD	CPU WR	H/W RD	H/W WR
Enable	Clear		Set

Sequence bit toggle flag: 0: No toggle, 1: Sequence toggle

Setting condition of this flag to "1" is as follows:

- Setting of handshake for OUT token in EP0PID is ACK, toggle of data PID is performed normally, and errors do not occur at data phase during OUT and SETUP transaction.
- When ACK is received during IN transaction.

This bit is cleared to "0" by writing dummy to this register.

b7	b6	b5	b4	b3	b2	b1	b0				
RxEP								CPU RD	CPU WR	H/W RD	H/W WR
Initial value 0											
								Endpoint determination flag			
								0: Endpoint 0 interrupt 1: Endpoint 1 interrupt			
								Enable			Set/ Clear
								This flag is set to "1" when IN token interrupt of endpoint 1 occurs. This flag is cleared to "0" when IN token interrupt of endpoint 0 occurs. Writing to this bit is invalid. Do not write "1" to bits 0 to 6.			

Initial value

b7	b6	b5	b4	b3	b2	b1	b0
RxPID	OPID			RSTRQ	SPRQ		

0 1 0 0

Suspend request flag  
0: No request  
1: Suspend request

Suspend request is set to "1" when system enters to state J for 3 ms or more.  
Suspend request is cleared to "0" by writing dummy to this register.

USB reset request flag  
0: No request  
1: Reset request

USB reset request is set to "1" when the SE0 signal is input for 2.5  $\mu$ s or more.  
USB reset request is cleared to "0" when the SE0 signal is stopped.

Token PID determination flag  
0: SETUP interrupt  
1: OUT interrupt

This flag is set to "1" during no SETUP transaction.  
This flag is cleared to "0" when PID of SETUP is detected.

Token interrupt flag  
0: No request  
1: OUT/SETUP token request

This flag is set to "1" when OUT or SETUP interrupt occurs.  
This flag is cleared to "0" after the end of transaction.

	b7	b6	b5	b4	b3	b2	b1	b0						
	USBE	TKNE	RSME	RSTE	EP1E									
Initial value	0	0	0	0	0									
									Endpoint 1 enable	0: Endpoint 1 invalid 1: Endpoint 1 valid	Enable	Set/Clear	Use	
									USB reset interrupt enable	0: USB reset invalid 1: USB reset valid	Enable	Set/Clear	Use	
This flag is invalid in suspend mode (USB reset is always valid in suspend mode).														
									Resume interrupt enable	0: Resume invalid 1: Resume valid	Enable	Set/Clear	Use	
									Token interrupt enable	0: Token invalid 1: Token valid	Enable	Set/Clear	Use	
									USB enable flag	0: USB invalid 1: USB valid	Enable	Set/Clear	Use	
<p>The internal state except the following registers can be cleared by clearing this flag to "0".</p> <p>The following registers and bits are initialized.</p> <ul style="list-style-type: none"> <li>• USB status register [address 19 16]</li> <li>• USB data toggle synchronization register [address 1D 16]</li> <li>• USB interrupt source discrimination register 1 [address 1E 16]</li> <li>• Bits 7, 6 and 2 of USB interrupt source discrimination register 2 [address 1F 16]</li> </ul>														

2-59

## 2.4 USB







# APPLICATION

## 2.4 USB

### 2.4.4 USB application example

In this section, the application examples when using the USB communication are described using examples of the timing chart and register setting.

#### (1) Example of processing each control sequence

In 7532/7536 Group, the control and the determination of the control sequence are executed by software.

The content of processing is executed in the setup stage, transmitting and receiving the data stage of the content are executed, and the completion of the sequence is shown in the status stage.

Note that the contents of the control and the determination are different even in the software of 7532/7536 Group, because the following processing is different respectively depending on the content of the setup stage.

In the control transfer, the processing since the data stage is determined when receiving the setup stage, the execution of the transmit and receive processing and the number of data bytes are controlled, the status stage at the end is executed, and the sequence is completed.

Only the control of each packet is performed because there is no stage in the interrupt transfer.

Figure 2.4.11 shows the control method of control sequence.

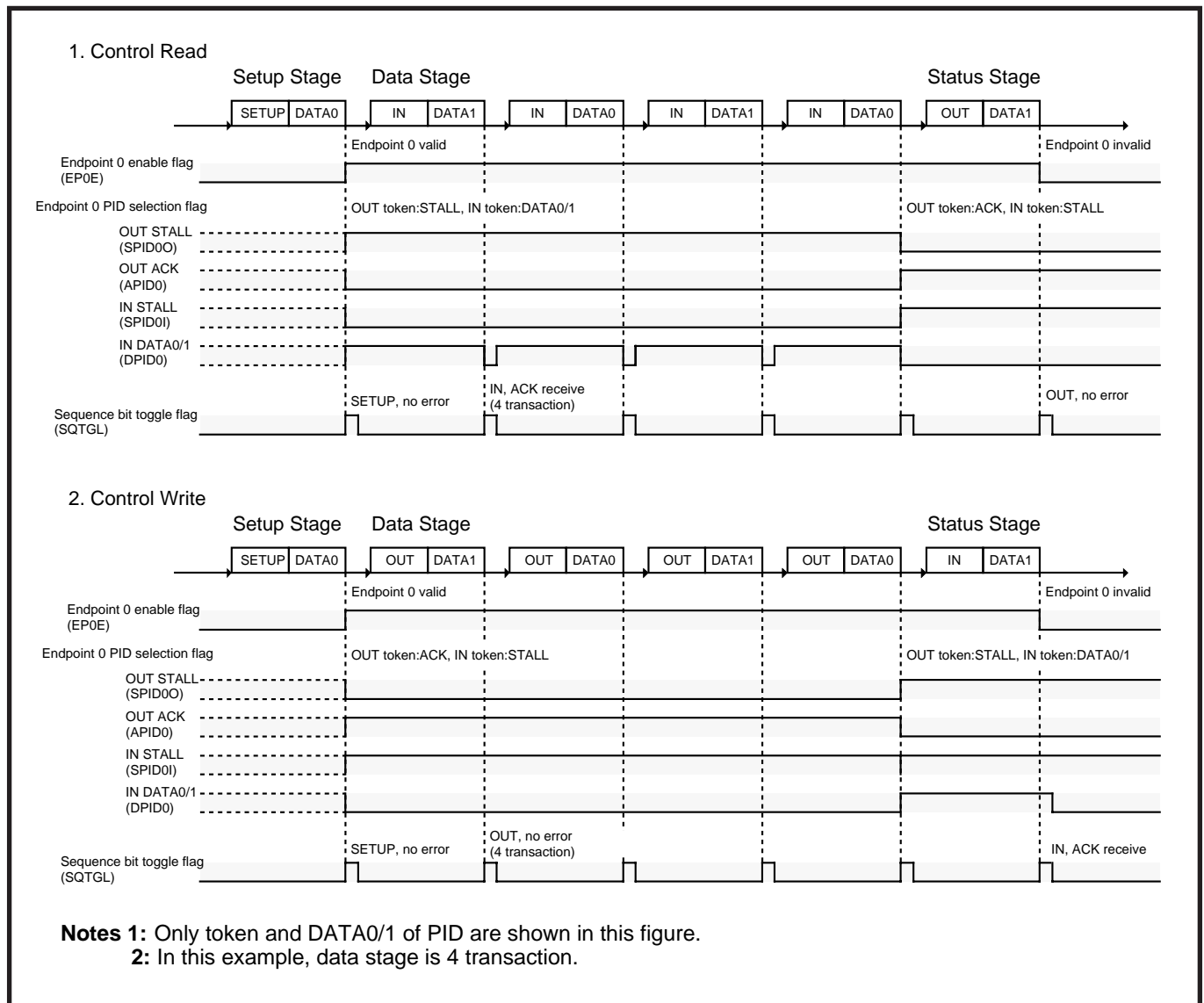


Fig. 2.4.11 Control method of control sequence

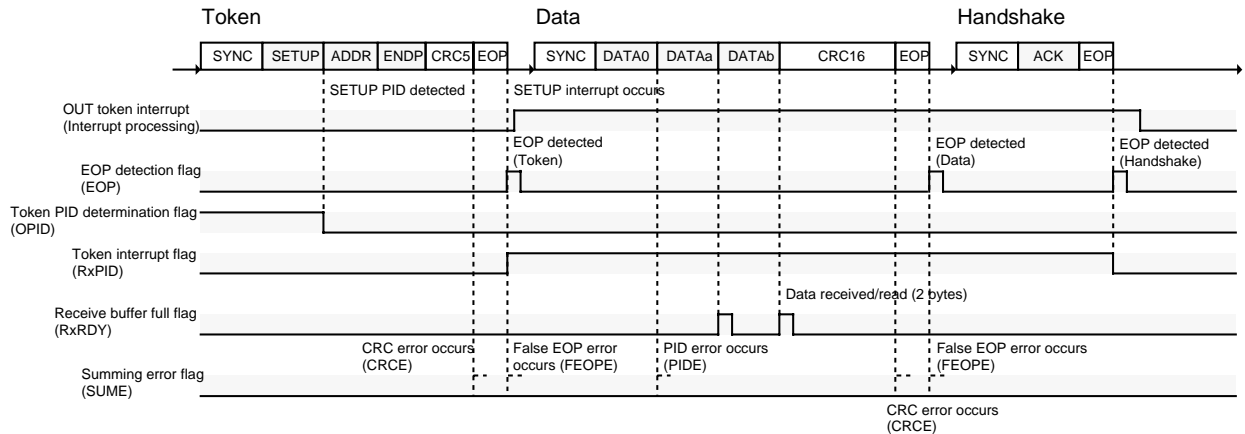
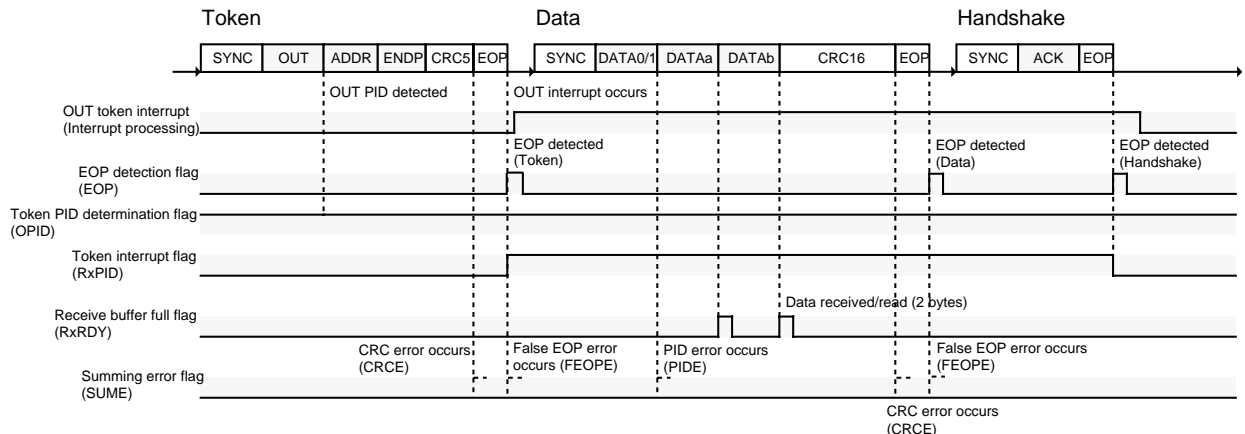
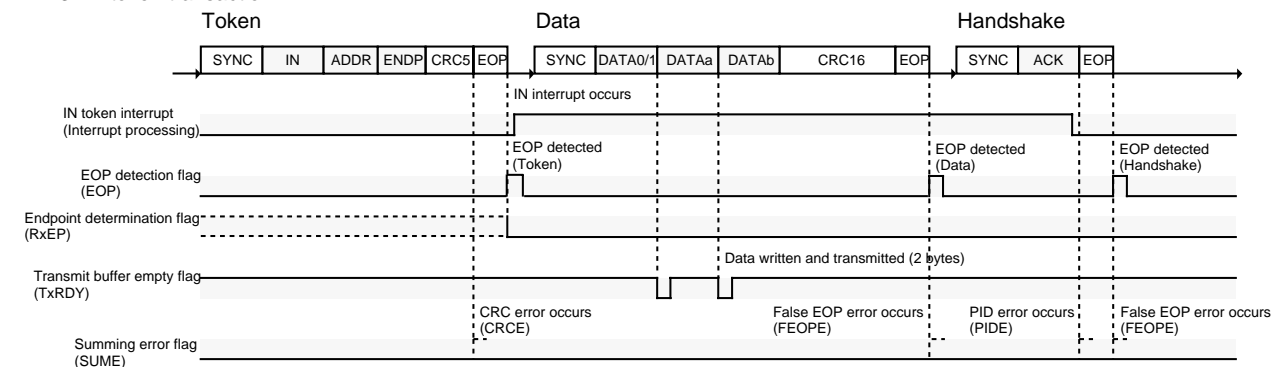
**(2) Example of processing each transaction**

In 7532/7536 Group, the control and the determination of the packet are executed by software.

First, the content of the received token is determined, transmit and receive data according to it are executed, the completion of the transaction is shown by the handshake.

Note that the contents of the control and the determination are different even of the software so that processing is different depending on the content of the token.

Figure 2.4.12 shows timing chart of the transaction according to each token.

**1. SETUP token transaction****2. OUT token transaction****3. IN token transaction**

**Notes 1:** The data number of PID = DATA0/1 is 2 bytes in this example.

**2:** Endpoint (ENDP) is 0 in this example.

**3:** The dot line on SUME shows the timing of each error occur.

**Fig. 2.4.12 Timing chart of the transaction according to each token**

# APPLICATION

## 2.4 USB

---

### (3) Interrupt processing

In 7532/7536 Group, the interrupt related to the USB communication processes 7 sources 2 jump destination.

Accordingly, determine the interrupt source and execute the processing after executing the interrupt processing.

Moreover, control the USB function and the interrupt source which has been enabled before interrupt is enabled.

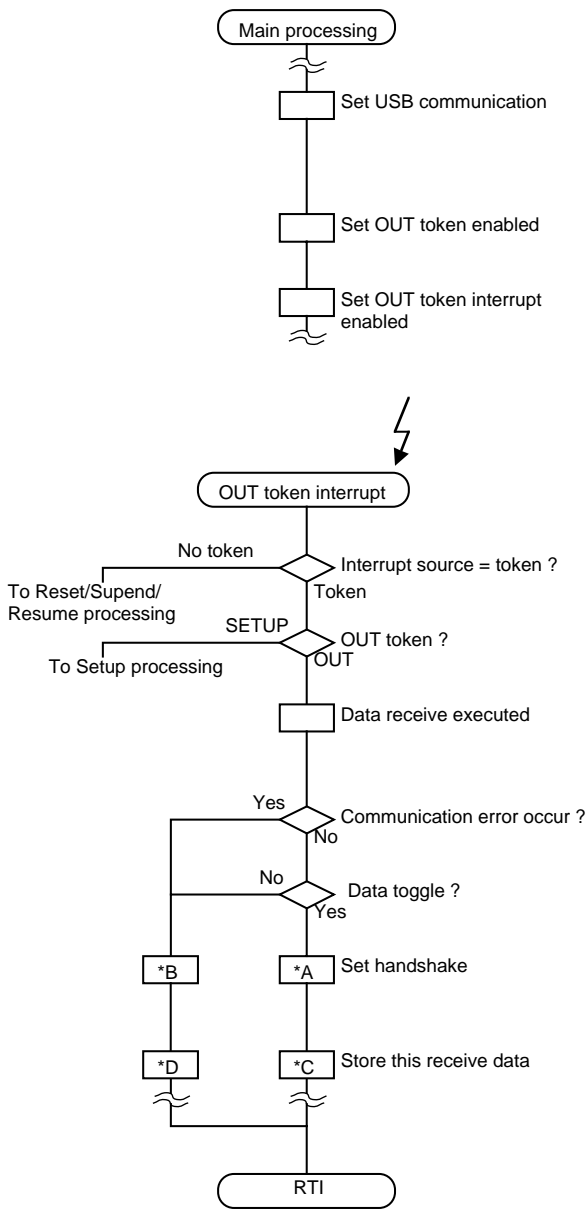
The interrupt jump destination and the source are shown as follows.

IN token interrupt: IN token (endpoint 0) and IN token (endpoint 1)

OUT token interrupt: OUT token, SETUP token, Reset, Suspend, and Resume

The determination (processing at the OUT token in figure) by the interrupt and the setting of a related register when using the interrupt of the OUT token is shown in Figure 2.4.13. The determination (processing at the IN token (endpoint 0) in figure) by the interrupt and the setting a related register when using the interrupt of the IN token in Figure 2.4.14.

[Flow chart]



●X: Not used here.  
Set it to "0" or "1" arbitrary.

[Program description]

```
( USBA      X0000000 2 Initial USB address = "0"      )

SIO1CON  11XXXXXX 2 Set hardware (I/O port) to USB mode
USBICON  1XXXXXXX 2 Set internal state to USB enabled

USBCON   X1XXXXXX 2 Use USBVREFOUT pin

EP0PID   XXXX1XXX 2 Set endpoint 0 to be valid

USBICON  X1XXXXXX 2 Set token interrupt to be valid
( IREQ1   XXXXXX0X 2 Initialize IN token interrupt request )
( ICON1   XXXXXX1X 2 Set IN token interrupt enable      )

USBIR2   ?XXXXXXX 2 Interrupt by token if ? = "1"

USBIR2   X?XXXXXX 2 Interrupt by OUT token if ? = "1"

RB       ???????? 2 Read receive data
                        (Max. 8 bytes in 1 interrupt
                        processing)

USBSTS   X?????XX 2 No error if all ? = "0"
                        Error/No error can be determined by SUME of bit 6

TRSYNC   ?XXXXXXX 2 Toggle normally if ? = "1"

In normal receive, handshake is set to ACK (*A).
When communication error or data doggle error occur, handshake is
set to NAK or STALL according to the contents (*B).

In normal receive, data is stored (*C).
When communication error or data doggle error occur, received data
in this time is canceled (*D).
```

**Notes 1:** In this figure, only USB in interrupt processing is described.  
Note that the storing stack, etc. is not described.  
**2:** Data shown by ? in program description expresses determination  
or data setting.

Fig. 2.4.13 USB interrupt processing example (OUT token)

# APPLICATION

## 2.4 USB

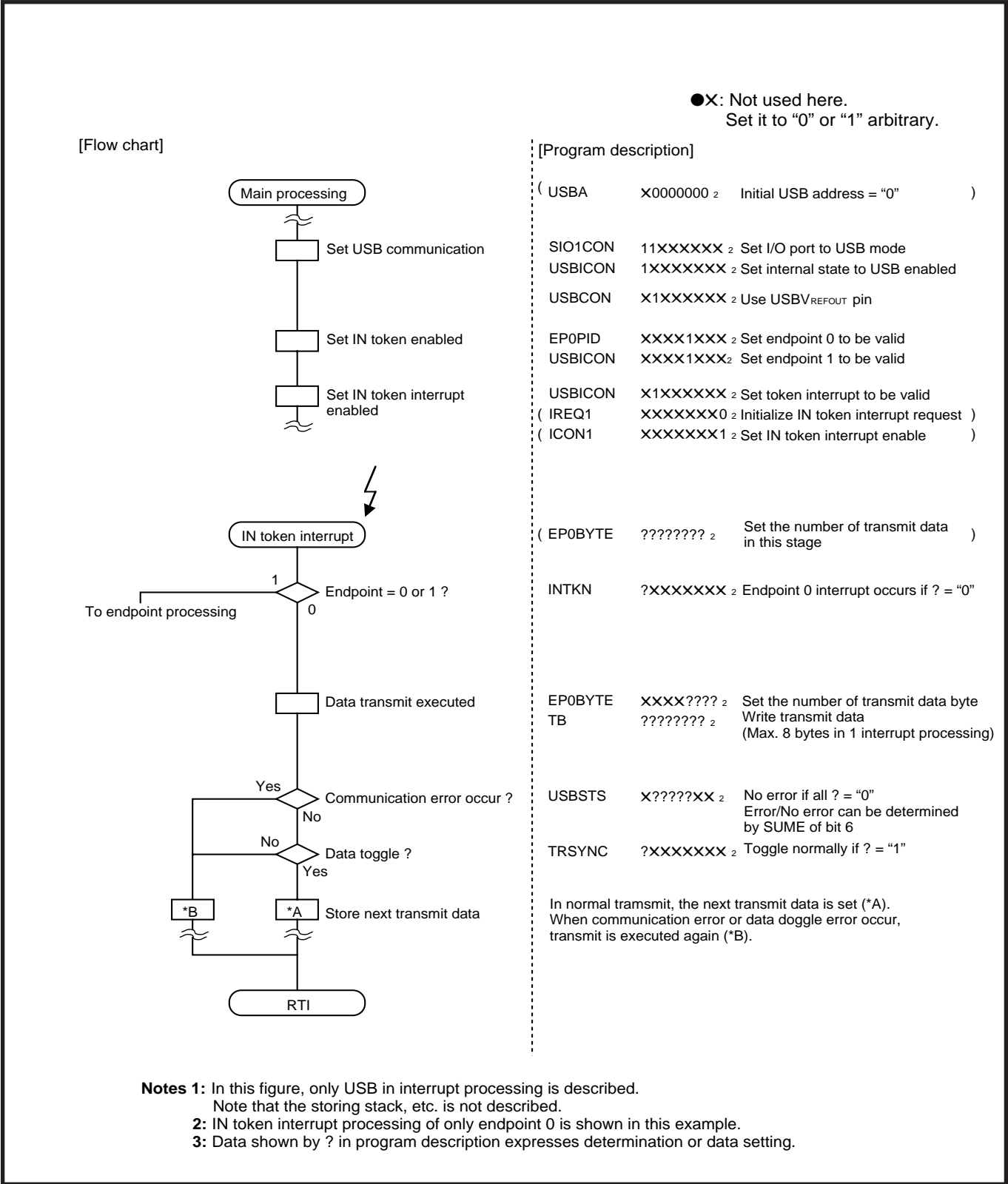


Fig. 2.4.14 USB interrupt processing example (IN token)

**(4) Processing to special signal**

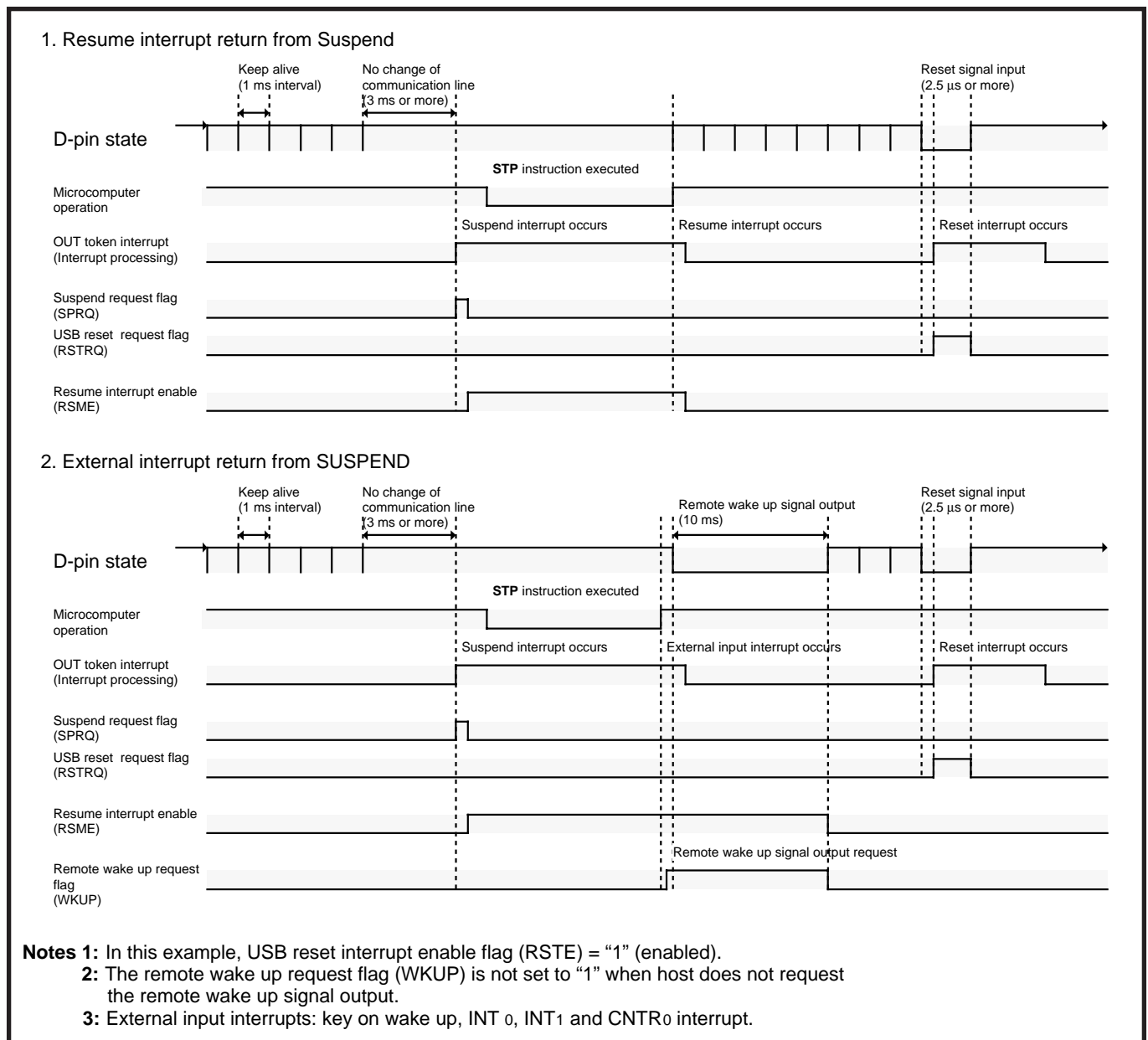
In 7532/7536 Group, control the USB function to the signal shown in Table 2.4.5 by software. Execute the **STP** instruction at reset which initializes the internal reset, and at suspend for stop. Set an external interrupt which is the return condition from the stopped state before **STP** instruction. The enable of resume is included in this condition. The generation of these signals can be recognized by the interrupt request. The interrupt to a special signal is all included in the OUT token interrupt.

IN token interrupt: IN token (endpoint 0) and IN token (endpoint 1)

OUT token interrupt: OUT token, SETUP token, Reset, Suspend, and Resume

The remote wake up function is used to output signals when to output as USB function is required for the return from the stopped state by an external input.

Figure 2.4.15 shows timing chart of each signal.



**Fig. 2.4.15 Timing chart of each signal**

# APPLICATION

## 2.4 USB

### 2.4.5 Notes concerning USB

#### (1) Determination of resume interrupt

The occurrence of the resume interrupt cannot be determined directly because there is no corresponding interrupt request flag when the resume interrupt occurs in the OUT token interrupt processing. Accordingly, determine the occurrence of the resume interrupt from other interrupt conditions in the order shown in Figure 2.4.16.

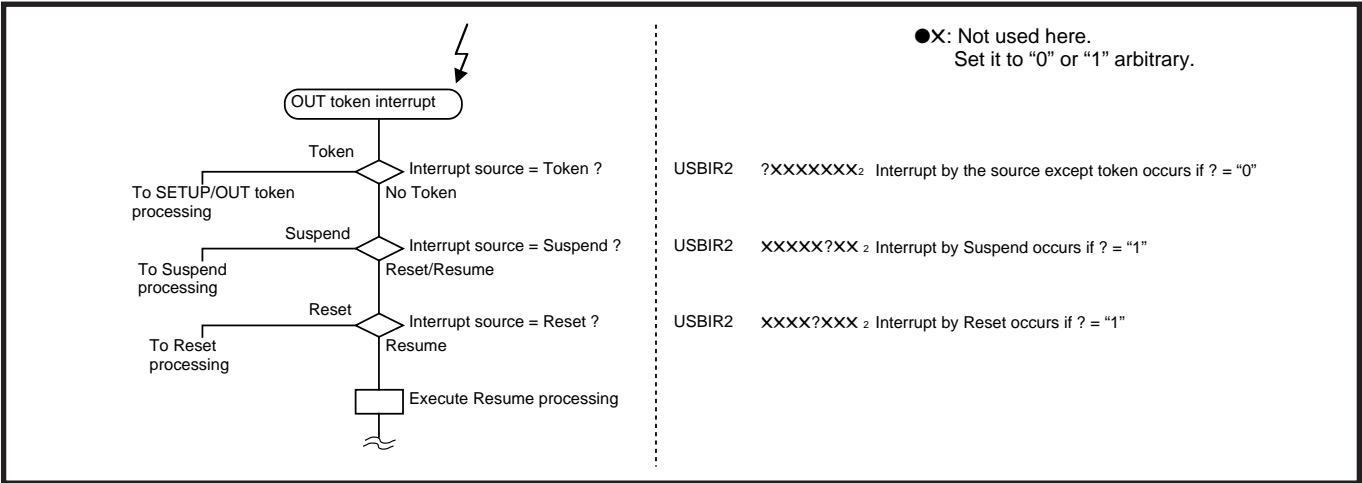


Fig. 2.4.16 Example for determination of resume interrupt

#### (2) Clear of suspend request flag

When the request of the suspend interrupt occurs, the suspend request flag is set to "1". After the suspend state is fixed, the state of this flag is retained during fixed time (13  $\mu$ s). The purpose of this is to retain the internal state until the count source to measure the time (3 ms) until suspend is fixed is updated. Accordingly, the state might not change even if this flag is cleared to "0" immediately after the suspend request flag is "1" is determined. Clear this flag to "0" after the wait of 13  $\mu$ s or 79 machine cycle ( $f(XIN) = 6$  MHz time) after this flag is "1".

#### (3) Determination of SE0 signal

In 7532/7536 Group, USB reset and EOP can be distinguished according to the width of the SE0 signal. However, there is the time zone which corresponds any on the dividing line of the time of the width of the signal. Moreover, the control in a present state is required because there is a difference in processing by the state of the device. Accordingly, select the processing method in software by the state of the device. Figure 2.4.17 shows processing to the width of the signal according to the situation.

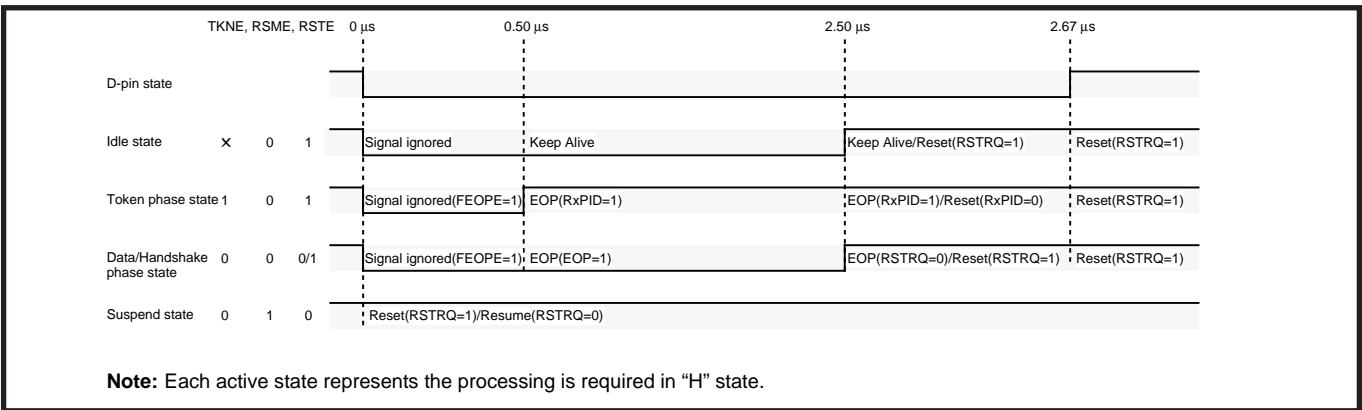


Fig. 2.4.17 Processing for width of SE0 signal

## 2.5 A-D converter

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

### 2.5.1 Memory map

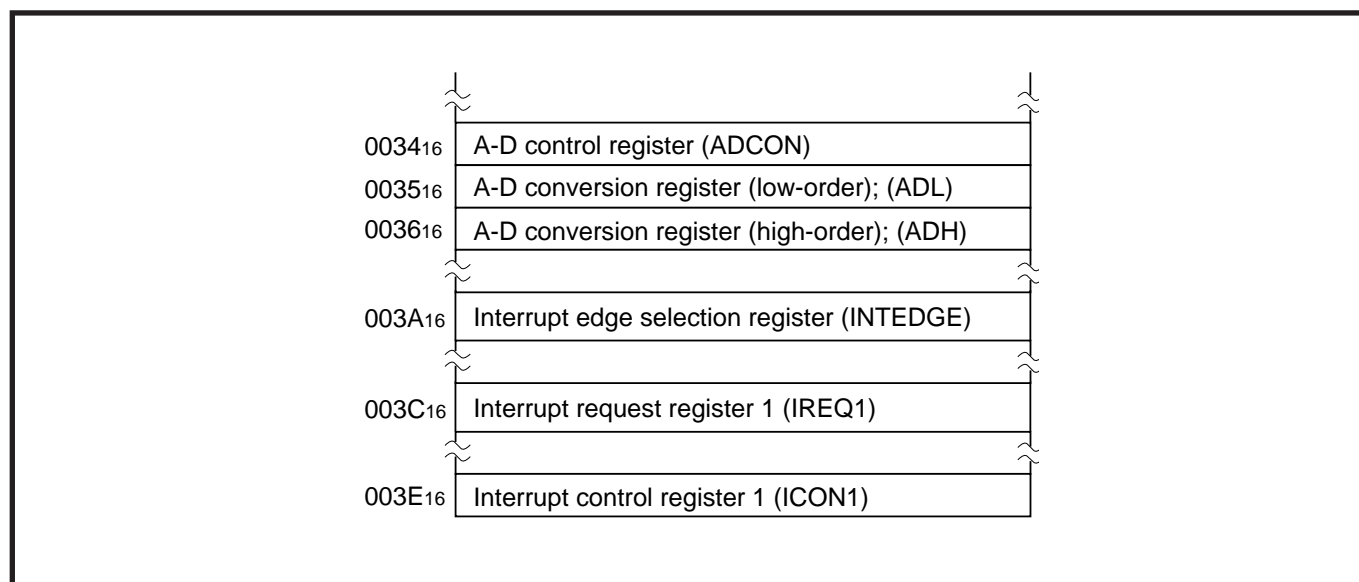


Fig. 2.5.1 Memory map of registers relevant to A-D converter

### 2.5.2 Relevant registers

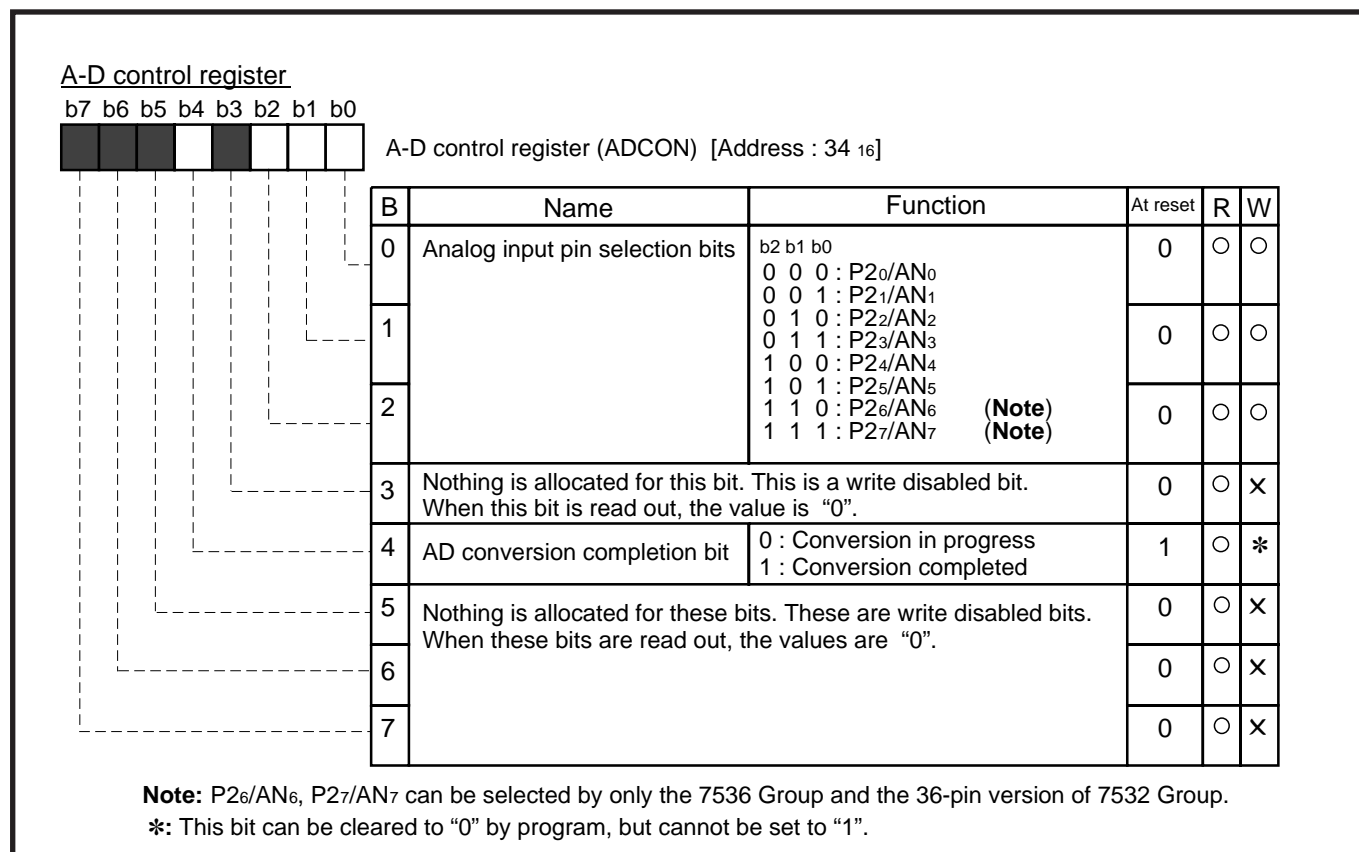


Fig. 2.5.2 Structure of A-D control register



# APPLICATION

## 2.5 A-D converter

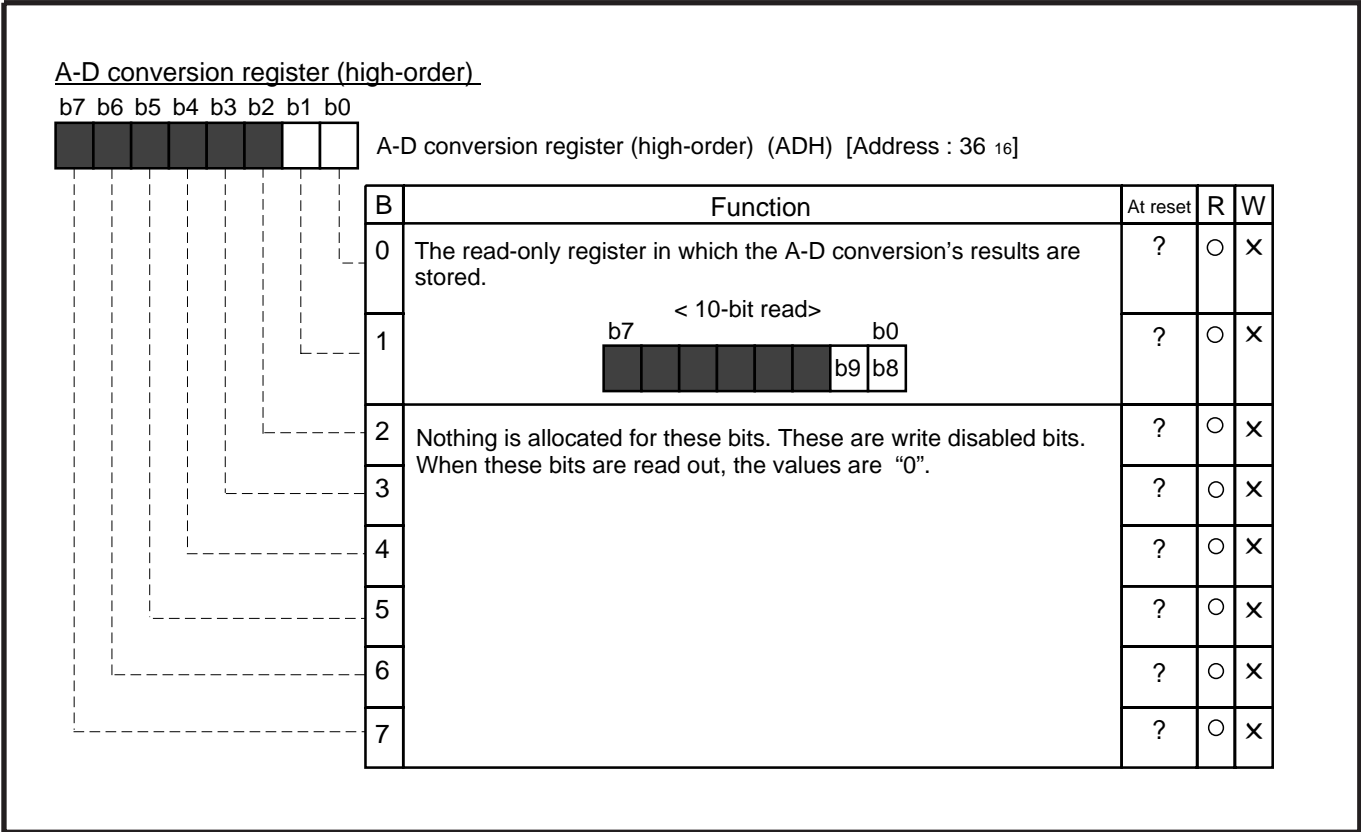


Fig. 2.5.3 Structure of A-D conversion register (high-order)

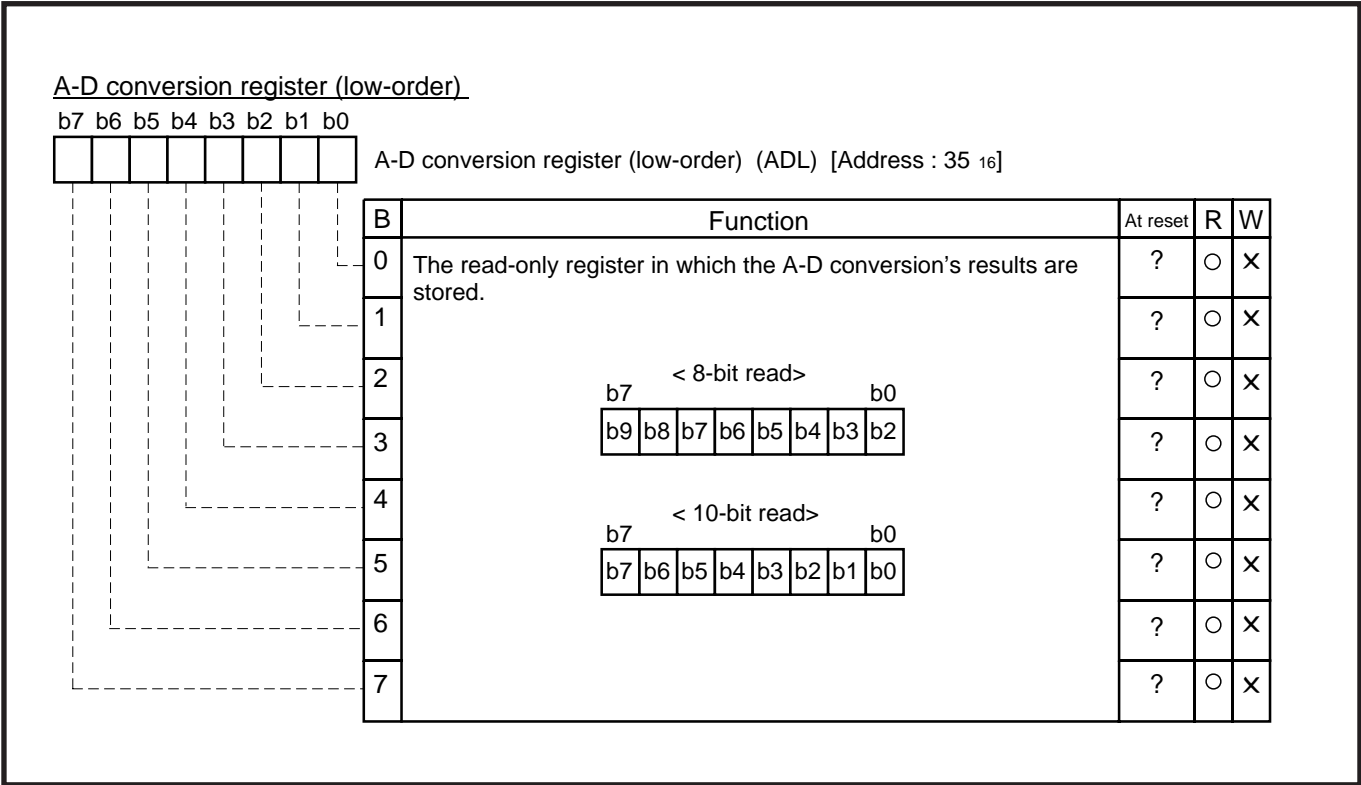


Fig. 2.5.4 Structure of A-D conversion register (low-order)

Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt edge selection register (INTEDGE) [Address : 3A<sub>16</sub>]

B	Name	Function	At reset	R	W
0	INT <sub>0</sub> interrupt edge selection bit ( <b>Note 1</b> )	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT <sub>1</sub> interrupt edge selection bit ( <b>Note 2</b> )	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
3			0	○	×
4	Serial I/O1 or INT <sub>1</sub> interrupt selection bit	0 : Serial I/O1 1 : INT <sub>1</sub>	0	○	○
5	Timer X or key-on wake up interrupt selection bit	0 : Timer X 1 : Key-on wake up	0	○	○
6	Timer 2 or serial I/O2 interrupt selection bit	0 : Timer 2 1 : Serial I/O2	0	○	○
7	CNTR <sub>0</sub> or AD converter interrupt selection bit	0 : CNTR <sub>0</sub> 1 : AD converter	0	○	○

**Notes 1:** 7532 Group, 32-pin version: Not used.

This is a write disabled bit. When this bit is read out, the value is "0".

**2:** 7532 Group, 36-pin and 32-pin version: Not used.

This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.5.5 Structure of Interrupt edge selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : 3C<sub>16</sub>]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
1	UART transmit/USBSETUP/OUT token/Reset/Suspend/Resume/INT <sub>1</sub> interrupt request bit ( <b>Note 1</b> )	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
2	INT <sub>0</sub> interrupt request bit ( <b>Note 2</b> )	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
3	Timer X or key-on wake up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
4	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
5	Timer 2 or serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
6	CNTR <sub>0</sub> or AD converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

\*: These bits can be cleared to "0" by program, but cannot be set.

**Notes 1:** 7532 Group, 36-pin version and 32-pin version: INT<sub>1</sub> interrupt does not exist.**2:** 7532 Group, 32-pin version: INT<sub>0</sub> interrupt does not exist.

This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.5.6 Structure of Interrupt request register 1

# APPLICATION

## 2.5 A-D converter

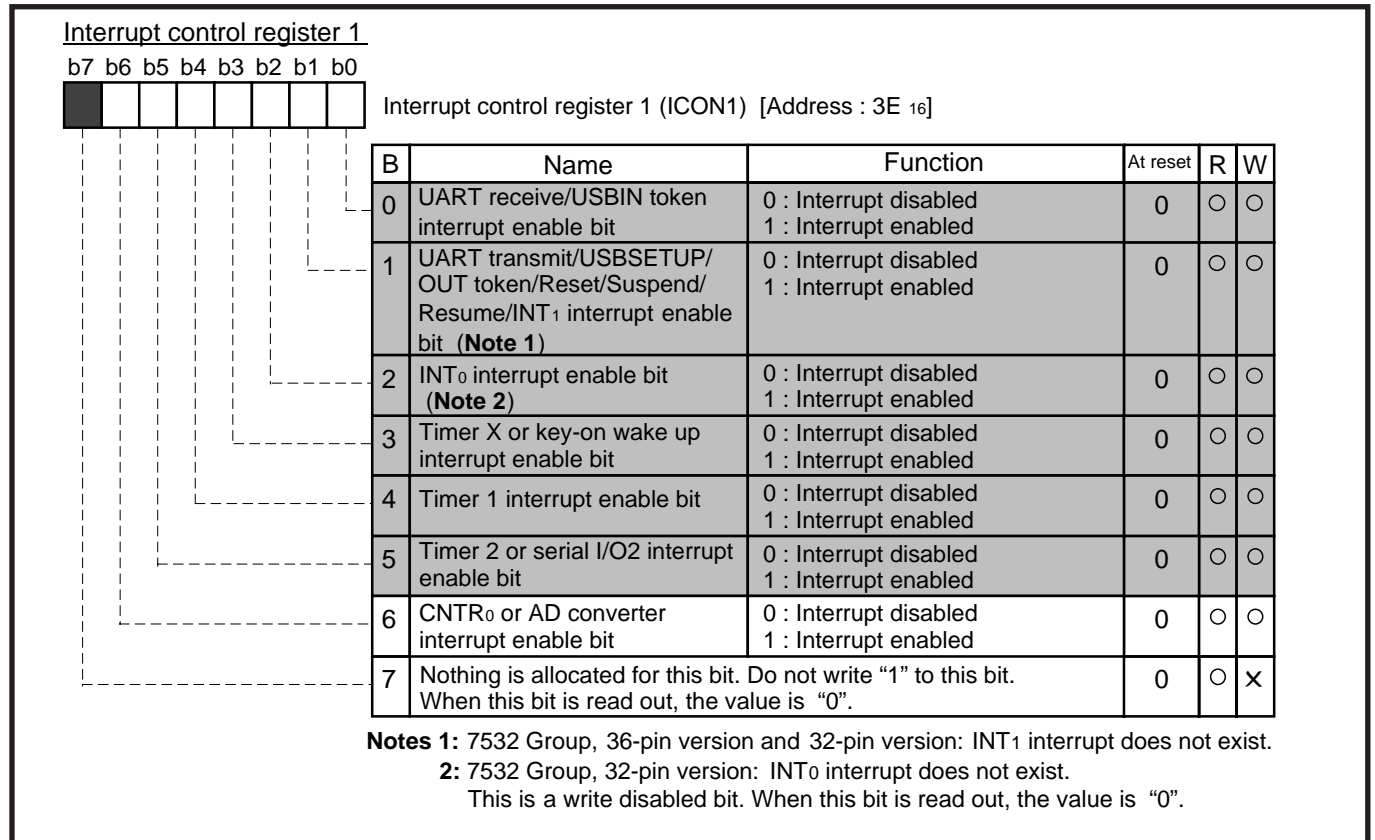


Fig. 2.5.7 Structure of Interrupt control register 1

## 2.5.3 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 2.5.8 shows a connection diagram, and Figure 2.5.9 shows the relevant registers setting.

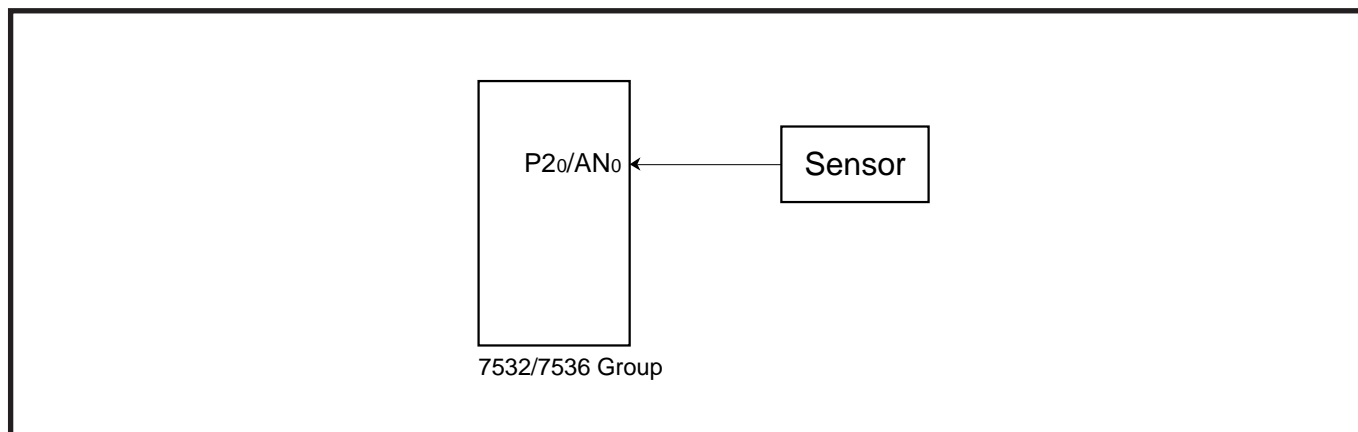


Fig. 2.5.8 Connection diagram

**Specifications :**

- The analog input voltage input from a sensor is converted to digital values.
- P20/AN0 pin is used as an analog input pin.

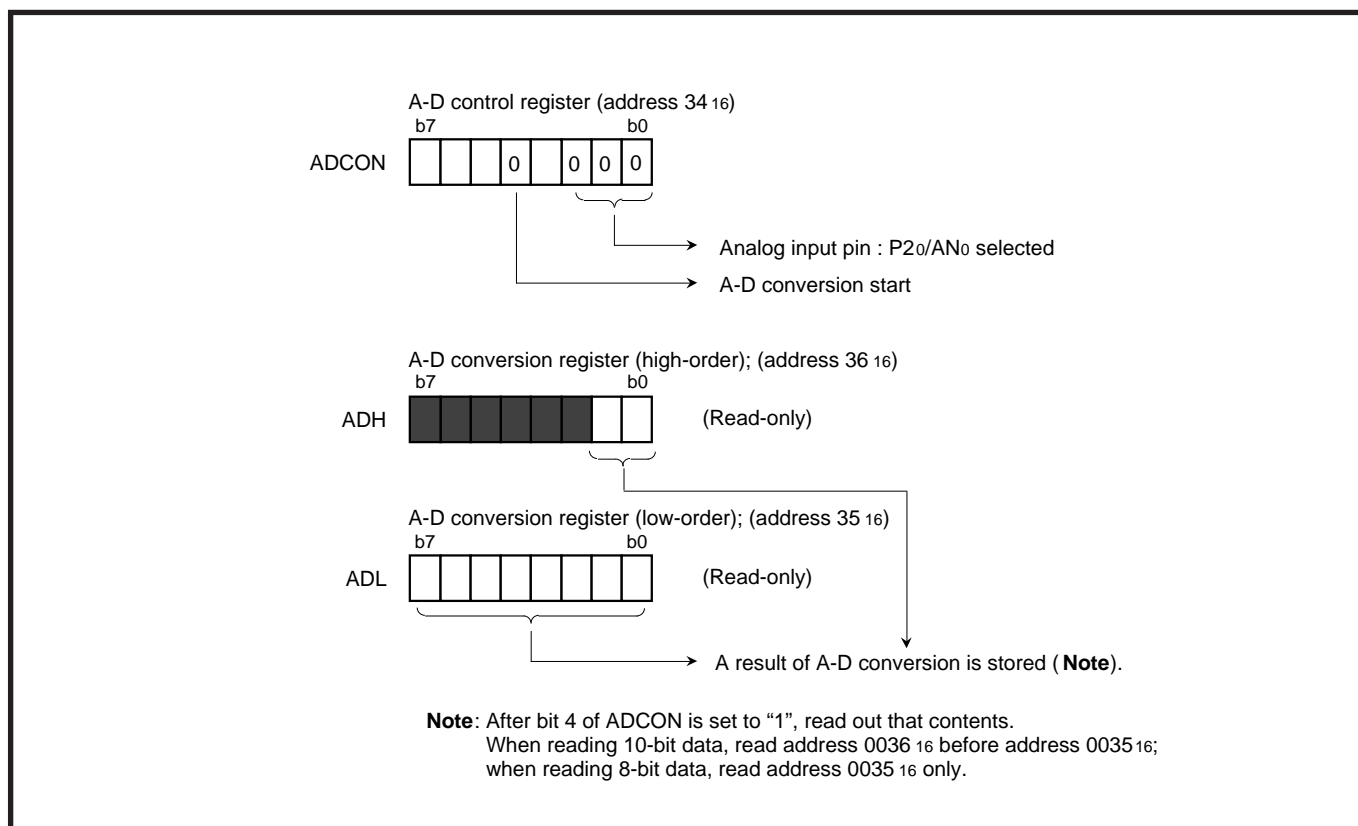


Fig. 2.5.9 Relevant registers setting

# APPLICATION

## 2.5 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 2.5.9. Figure 2.5.10 shows the control procedure for 8-bit read, and Figure 2.5.11 shows the control procedure for 10-bit read.

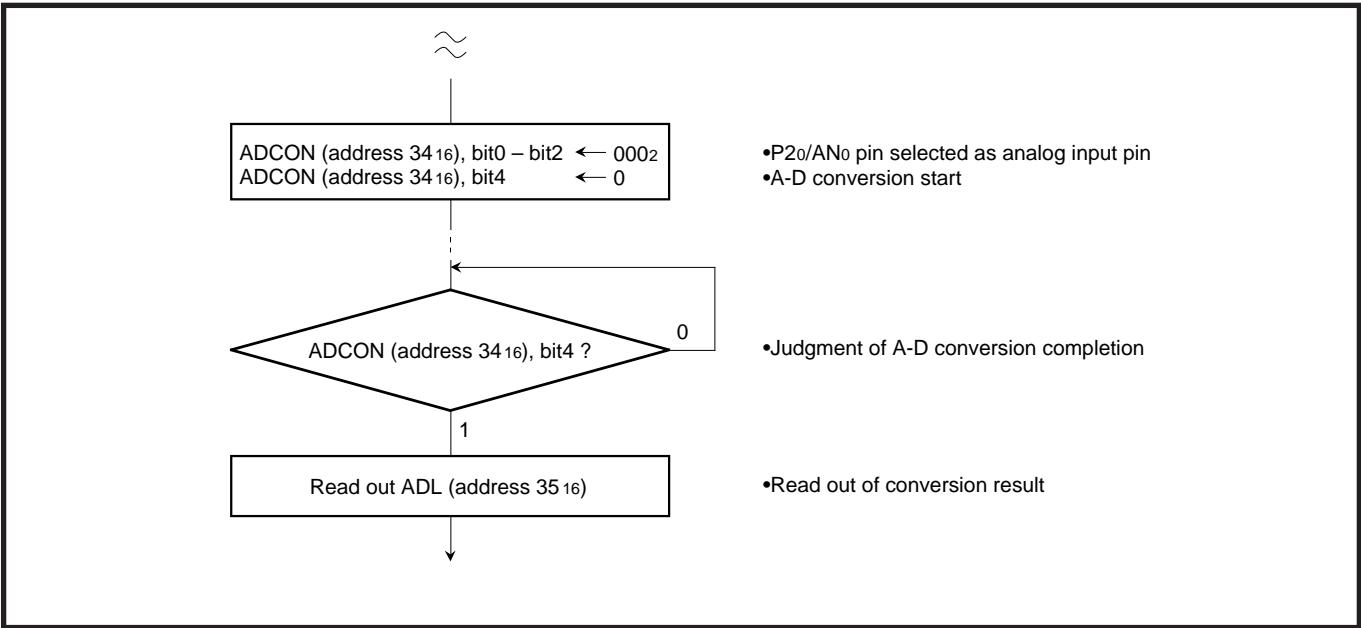


Fig. 2.5.10 Control procedure for 8-bit read

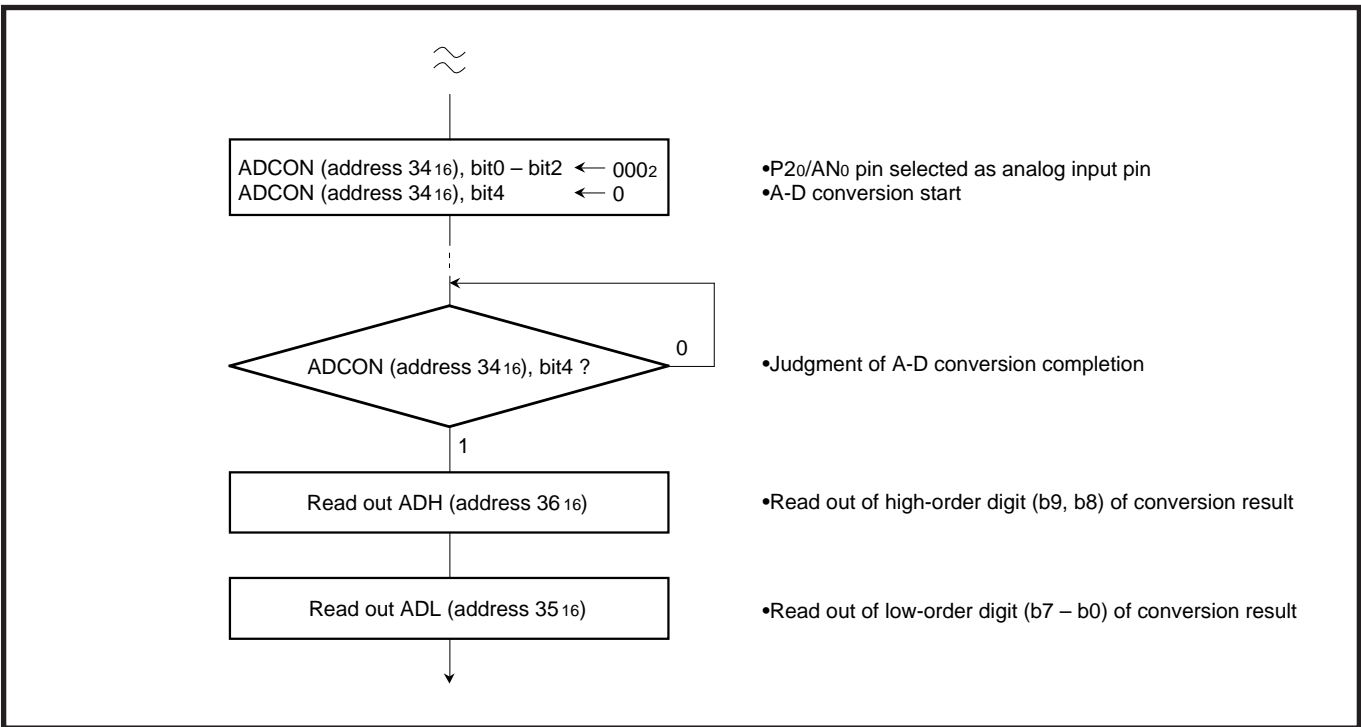


Fig. 2.5.11 Control procedure for 10-bit read

### 2.5.4 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 $\mu$ F to 1 $\mu$ 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/comparison precision to be worse.

#### (2) 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.

- $f(X_{IN})$  is 500 kHz or more
- Do not execute the **STP** instruction

# APPLICATION

## 2.6 Reset

## 2.6 Reset

### 2.6.1 Connection example of reset IC

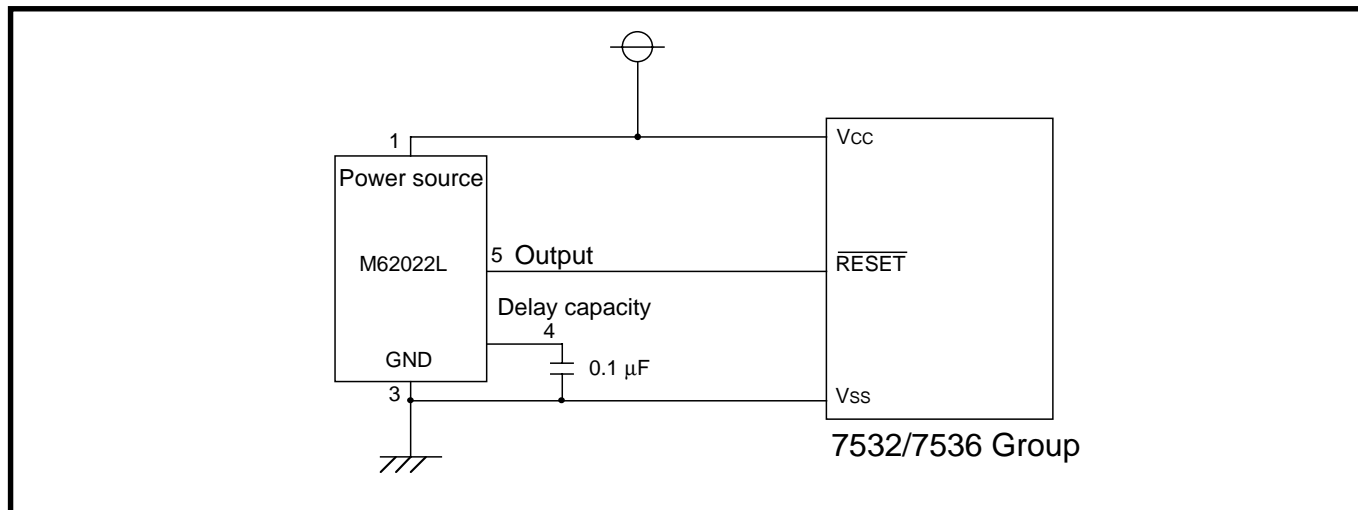


Fig. 2.6.1 Example of poweron reset circuit

Figure 2.6.2 shows the system example which switches to the RAM backup mode by detecting a drop of the system power source voltage with the INT interrupt.

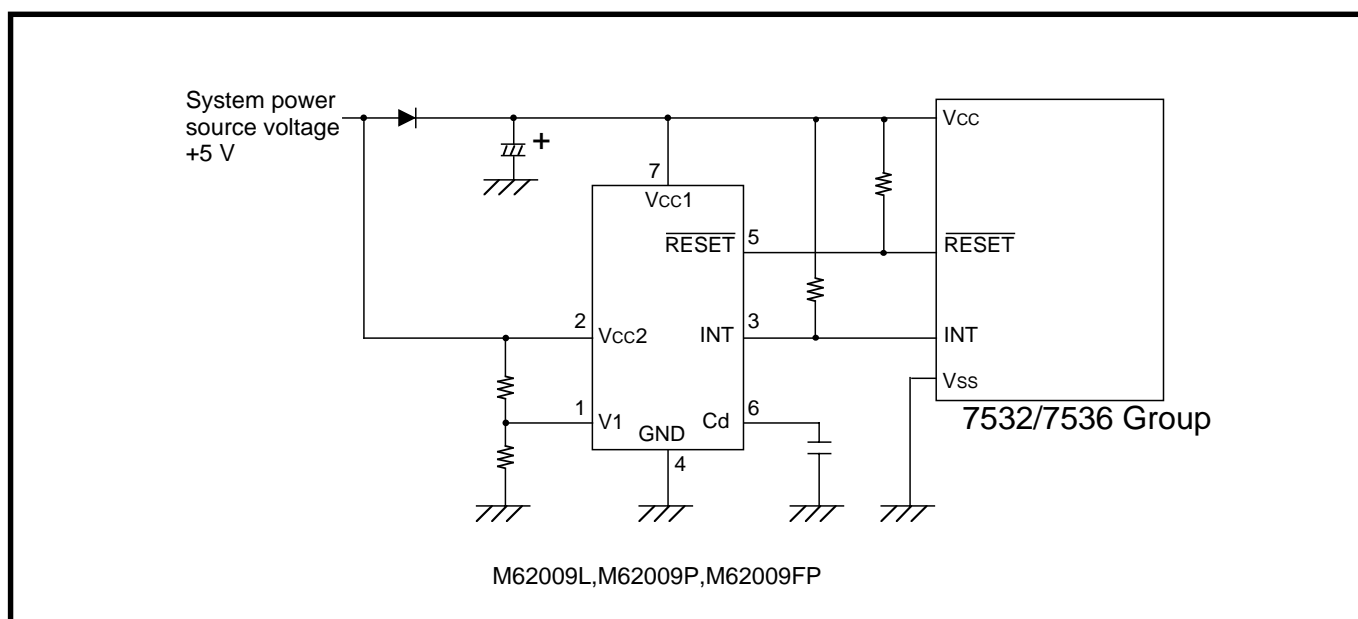


Fig. 2.6.2 RAM backup system

### 2.6.2 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

## 2.6 Reset

---

### *MEMO*



## CHAPTER 3

# APPENDIX

- 3.1 Electrical characteristics
- 3.2 Typical characteristics
- 3.3 Notes on use
- 3.4 Countermeasures against noise
- 3.5 List of registers
- 3.6 Mask ROM confirmation form
- 3.7 Mark specification form
- 3.8 Package outline
- 3.9 Machine instructions
- 3.10 List of instruction code
- 3.11 SFR memory map
- 3.12 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 <sub>CC</sub>	Power source voltage	All voltages are based on V <sub>SS</sub> . Output transistors are cut off.	−0.3 to 7.0	V
V <sub>I</sub>	Input voltage P00–P07, P10–P16, P20–P27, P30–P37, V <sub>REF</sub> , P40, P41		−0.3 to V <sub>CC</sub> + 0.3	V
V <sub>I</sub>	Input voltage RESET, X <sub>IN</sub>		−0.3 to V <sub>CC</sub> + 0.3	V
V <sub>I</sub>	Input voltage CNV <sub>SS</sub> (Note 1)		−0.3 to 13	V
V <sub>O</sub>	Output voltage P00–P07, P10–P16, P20–P27, P30–P37, X <sub>OUT</sub> , USBV <sub>REFOUT</sub> , P40, P41		−0.3 to V <sub>CC</sub> + 0.3	V
P <sub>d</sub>	Power dissipation (Note 2)	T <sub>a</sub> = 25°C	1000 (Note 3)	mW
T <sub>opr</sub>	Operating temperature		−20 to 85	°C
T <sub>stg</sub>	Storage temperature		−40 to 125	°C

**Notes 1:** It is a rating only for the One Time PROM version. Connect to V<sub>SS</sub> for mask ROM version.

**2:** The rating value depends on packages.

**3:** This rating value is for the 7536 Group.

The value of the 36-pin version of the 7532 Group is 300 mW.

The value of the 32-pin version of the 7532 Group is 200 mW.

# APPENDIX

## 3.1 Electrical characteristics

### 3.1.2 Recommended Operating Conditions

**Table 3.1.2 Recommended operating conditions**  
(V<sub>CC</sub> = 4.1 to 5.5 V, T<sub>a</sub> = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V <sub>CC</sub>	Power source voltage	f(XIN) = 6 MHz	4.1	5.0	5.5	V
V <sub>SS</sub>	Power source voltage			0		V
V <sub>REF</sub>	Analog reference voltage		2.0		V <sub>CC</sub>	V
V <sub>IH</sub>	"H" input voltage	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41	0.8 V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IH</sub>	"H" input voltage (TTL input level selected)	P10, P12, P13, P36, P37	2.0		V <sub>CC</sub>	V
V <sub>IH</sub>	"H" input voltage	RESET, XIN	0.8 V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IH</sub>	"H" input voltage	D+, D-	2.0		3.6	V
V <sub>IL</sub>	"L" input voltage	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41	0		0.3 V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage (TTL input level selected)	P10, P12, P13, P36, P37	0		0.8	V
V <sub>IL</sub>	"L" input voltage	RESET, CNVss	0		0.2 V <sub>CC</sub>	V
V <sub>IL</sub>	"L" input voltage	D+, D-	0		0.8	V
V <sub>IL</sub>	"L" input voltage	XIN	0		0.16V <sub>CC</sub>	V
ΣIOH(peak)	"H" total peak output current (Note 1)	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41			–80	mA
ΣIOL(peak)	"L" total peak output current (Note 1)	P00–P07, P10–P16, P20–P27, P37, P40, P41			80	mA
ΣIOL(peak)	"L" total peak output current (Note 1)	P30–P36			60	mA
ΣIOH(avg)	"H" total average output current (Note 1)	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41			–40	mA
ΣIOL(avg)	"L" total average output current (Note 1)	P00–P07, P10–P16, P20–P27, P37, P40, P41			40	mA
ΣIOL(avg)	"L" total average output current (Note 1)	P30–P36			30	mA
IOH(peak)	"H" peak output current (Note 2)	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41			–10	mA
IOL(peak)	"L" peak output current (Note 2)	P00–P07, P10–P16, P20–P27, P37, P40, P41			10	mA
IOL(peak)	"L" peak output current (Note 2)	P30–P36			30	mA
IOH(avg)	"H" average output current (Note 3)	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41			–5	mA
IOL(avg)	"L" average output current (Note 3)	P00–P07, P10–P16, P20–P27, P37, P40, P41			5	mA
IOL(avg)	"L" average output current (Note 3)	P30–P36			15	mA
f(XIN)	Oscillation frequency (Note 4)	V <sub>CC</sub> = 4.1 to 5.5 V at ceramic oscillation or external clock input Double-speed mode			6	MHz

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

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

**3:** The average output current IOL (avg), IOH (avg) in an average value measured over 100 ms.

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

# APPENDIX

## 3.1 Electrical characteristics

### 3.1.3 Electrical Characteristics

**Table 3.1.3 Electrical characteristics (1)**

(V<sub>CC</sub> = 4.1 to 5.5 V, V<sub>SS</sub> = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
VOH	“H” output voltage P00–P07, P10–P16, P20–P27, P30–P37, P40, P41 (Note 1)	IOH = –5 mA VCC = 4.1 to 5.5 V	VCC–1.5			V
		IOH = –1.0 mA VCC = 4.1 to 5.5 V	VCC–1.0			V
VOH	“H” output voltage D+, D-	VCC = 4.4 to 5.25 V Pull-down through 15kΩ ±5 % for D+, D- Pull-up through 1.5kΩ ±5 % by USBVREFOUT for D- (Ta = 0 to 70 °C)	2.8		3.6	V
VOL	“L” output voltage P00–P07, P10–P16, P20–P27, P37, P40, P41	IOL = 5 mA VCC = 4.1 to 5.5 V			1.5	V
		IOL = 1.5 mA VCC = 4.1 to 5.5 V			0.3	V
VOL	“L” output voltage D+, D-	VCC = 4.4 to 5.25 V Pull-down through 15kΩ ±5 % for D+, D- Pull-up through 1.5kΩ ±5 % by USBVREFOUT for D- (Ta = 0 to 70 °C)			0.3	V
VOL	“L” output voltage P30–P36	IOL = 15 mA VCC = 4.1 to 5.5 V			2.0	V
		IOL = 1.5 mA VCC = 4.1 to 5.5 V			0.3	V
VT+–VT–	Hysteresis D+, D-			0.15		V
VT+–VT–	Hysteresis CNTR0, INT0, INT1 (Note 2), P00–P07 (Note 3)			0.4		V
VT+–VT–	Hysteresis RXD, SCLK, SDATA (Note 2)			0.5		V
VT+–VT–	Hysteresis RESET			0.5		V
IiH	“H” input current P00–P07, P10–P16, P20–P27, P30–P37, P40, P41	Vi = VCC (Pin floating. Pull-up transistors “off”)			5.0	μA
IiH	“H” input current RESET	Vi = VCC			5.0	μA
IiH	“H” input current XIN	Vi = VCC		4		μA
IiL	“L” input current P00–P07, P10–P16, P20–P27, P30–P37, P40, P41	Vi = VSS (Pin floating. Pull-up transistors “off”)			–5.0	μA
IiL	“L” input current RESET, CNVSS	Vi = VSS			–5.0	μA
IiL	“L” input current XIN	Vi = VSS		–4		μA
IiL	“L” input current P00–P07, P30–P37	Vi = VSS (Pull-up transistors “on”)		–0.2	–0.5	mA
VRAM	RAM hold voltage	When clock stopped	2.0		5.5	V

**Note 1:** P11 is measured when the P-channel output disable bit of the UART control register (bit 4 of address 001B16) is “0”.

**2:** RXD, SCLK, SDATA, INT0 and INT1 have hystereses only when bits 0, 1 and 2 of the port P1P3 control register are set to “0” (CMOS level).

**3:** It is available only when operating key-on wake-up.

# APPENDIX

## 3.1 Electrical characteristics

**Table 3.1.4 Electrical characteristics (2)**

( $V_{CC} = 4.1$  to  $5.5$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85$  °C, unless otherwise noted)

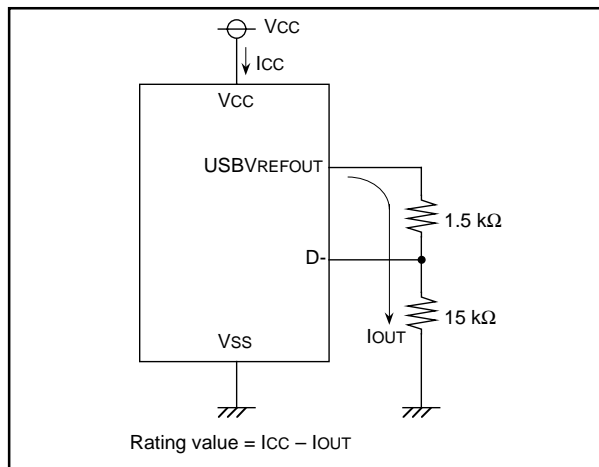
Symbol	Parameter		Test conditions	Limits			Unit
				Min.	Typ.	Max.	
I <sub>CC</sub>	Power source current	Double-speed mode, f(X <sub>IN</sub> ) = 6 MHz, Output transistors "off"			6	10	mA
		f(X <sub>IN</sub> ) = 6 MHz, (in WIT state) Output transistors "off"			1.6	3.2	mA
		Increment when A-D conversion is executed f(X <sub>IN</sub> ) = 6 MHz, V <sub>CC</sub> = 5 V			0.8		mA
		All oscillation stopped (in STP state) Output transistors "off"	T <sub>a</sub> = 25 °C		0.1	1.0	μA
			T <sub>a</sub> = 85 °C			10	μA
		V <sub>CC</sub> = 4.4 V to 5.25 V Oscillation stopped in USB mode USB (SUSPEND), (pull-up resistor output not included) (Fig. 3.1.1)	T <sub>a</sub> = 0 to 70 °C			300	μA

### 3.1.4 A-D Converter Characteristics

**Table 3.1.5 A-D Converter characteristics (1)**

( $V_{CC} = 4.1$  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.	
—	Resolution				10	Bits
—	Linearity error	V <sub>CC</sub> = 4.1 to 5.5 V T <sub>a</sub> = 25 °C			±3	LSB
—	Differential nonlinear error	V <sub>CC</sub> = 4.1 to 5.5 V T <sub>a</sub> = 25 °C			±0.9	LSB
V <sub>OT</sub>	Zero transition voltage	V <sub>CC</sub> = V <sub>REF</sub> = 5.12 V	0	5	20	mV
V <sub>FST</sub>	Full scale transition voltage	V <sub>CC</sub> = V <sub>REF</sub> = 5.12 V	5105	5115	5125	mV
t <sub>CONV</sub>	Conversion time				122	t <sub>c</sub> (X <sub>IN</sub> )
R <sub>LADDER</sub>	Ladder resistor			55		kΩ
I <sub>VREF</sub>	Reference power source input current	V <sub>REF</sub> = 5.0 V	50	150	200	μA
		V <sub>REF</sub> = 3.0 V	30	70	120	
I <sub>I(AD)</sub>	A-D port input current				5.0	μA



**Fig. 3.1.1 Power source current measurement circuit in USB mode at oscillation stop**

# APPENDIX

## 3.1 Electrical characteristics

### 3.1.5 Timing Requirements

**Table 3.1.6 Timing requirements**

( $V_{CC} = 4.1$  to  $5.5$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85$  °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
tw(RESET)	Reset input "L" pulse width	15			μs
tc(XIN)	External clock input cycle time	166			ns
tWH(XIN)	External clock input "H" pulse width	70			ns
tWL(XIN)	External clock input "L" pulse width	70			ns
tc(CNTR)	CNTR0 input cycle time	200			ns
tWH(CNTR)	CNTR0, INT0, INT1 input "H" pulse width	80			ns
tWL(CNTR)	CNTR0, INT0, INT1 input "L" pulse width	80			ns
tc(SCLK)	Serial I/O2 clock input cycle time	1000			ns
tWH(SCLK)	Serial I/O2 clock input "H" pulse width	400			ns
tWL(SCLK)	Serial I/O2 clock input "L" pulse width	400			ns
tsu(SCLK–SDATA)	Serial I/O2 input set up time	200			ns
th(SCLK–SDATA)	Serial I/O2 input hold time	200			ns

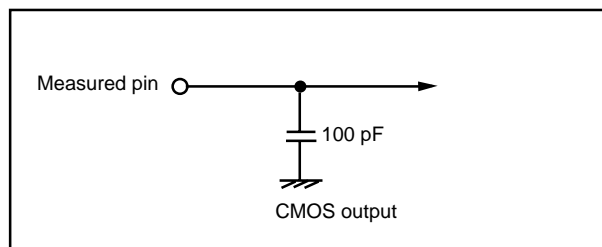
### 3.1.6 Switching Characteristics

**Table 3.1.7 Switching characteristics**

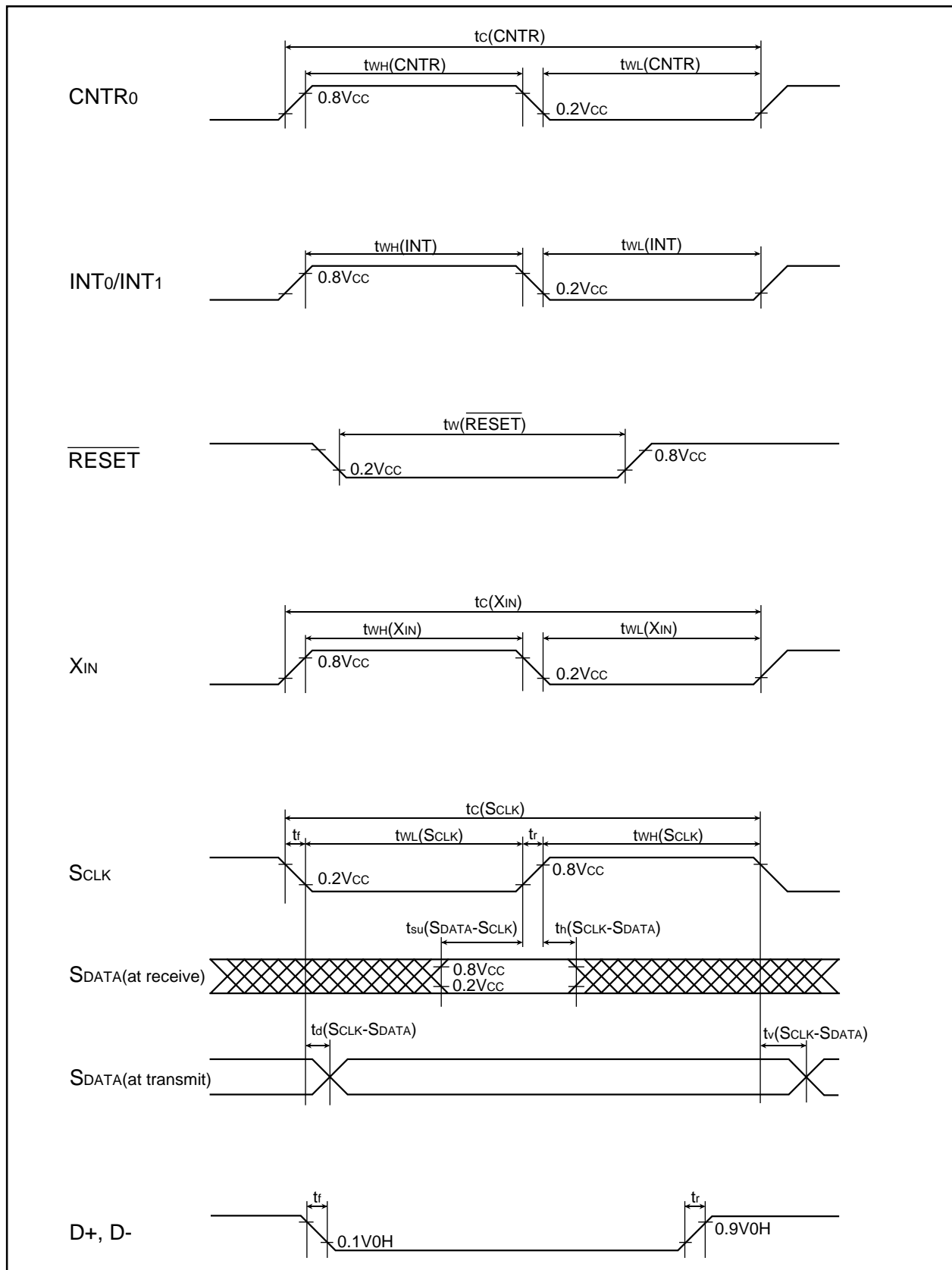
( $V_{CC} = 4.1$  to  $5.5$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85$  °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
tWH(SCLK)	Serial I/O2 clock output "H" pulse width	tc(SCLK)/2–30			ns
tWL(SCLK)	Serial I/O2 clock output "L" pulse width	tc(SCLK)/2–30			ns
td(SCLK–SDATA)	Serial I/O2 output delay time			140	ns
tv(SCLK–SDATA)	Serial I/O2 output valid time	0			ns
tr(SCLK)	Serial I/O2 clock output rising time			30	ns
tf(SCLK)	Serial I/O2 clock output falling time			30	ns
tr(CMOS)	CMOS output rising time (Note)		10	30	ns
tf(CMOS)	CMOS output falling time (Note)		10	30	ns
tr(D+), tr(D-)	USB output rising time, $C_L = 350$ pF, $T_a = 0$ to $70$ °C	100	200	300	ns
tf(D+), tf(D-)	USB output falling time, $C_L = 350$ pF, $T_a = 0$ to $70$ °C	100	200	300	ns

**Note:** XOUT pin is excluded.



**Fig. 3.1.2 Switching characteristics measurement circuit**



**Fig. 3.1.3 Timing chart**



# APPENDIX

## 3.2 Typical characteristics

## 3.2 Typical characteristics

### 3.2.1 Power source current characteristic example (Icc-Vcc characteristic)

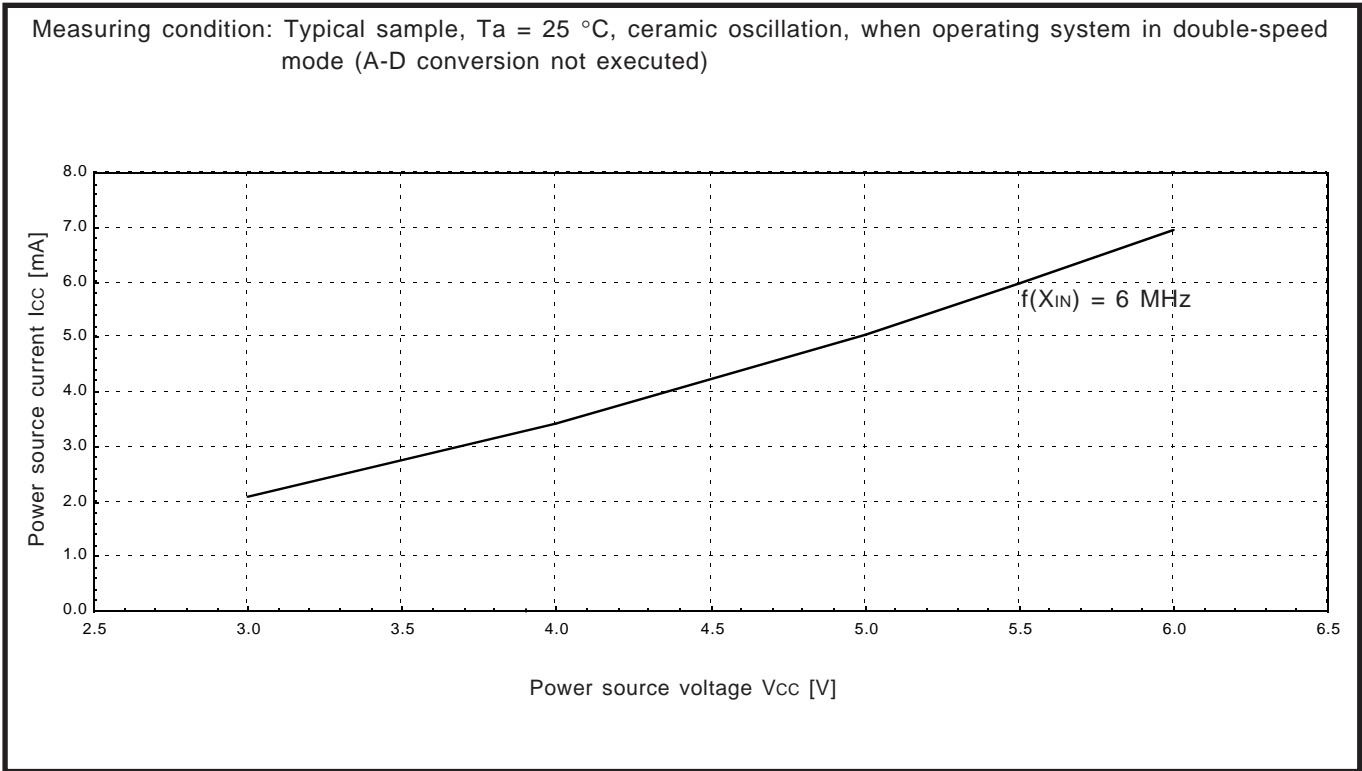


Fig. 3.2.1 Icc-Vcc characteristic example (in double-speed mode)

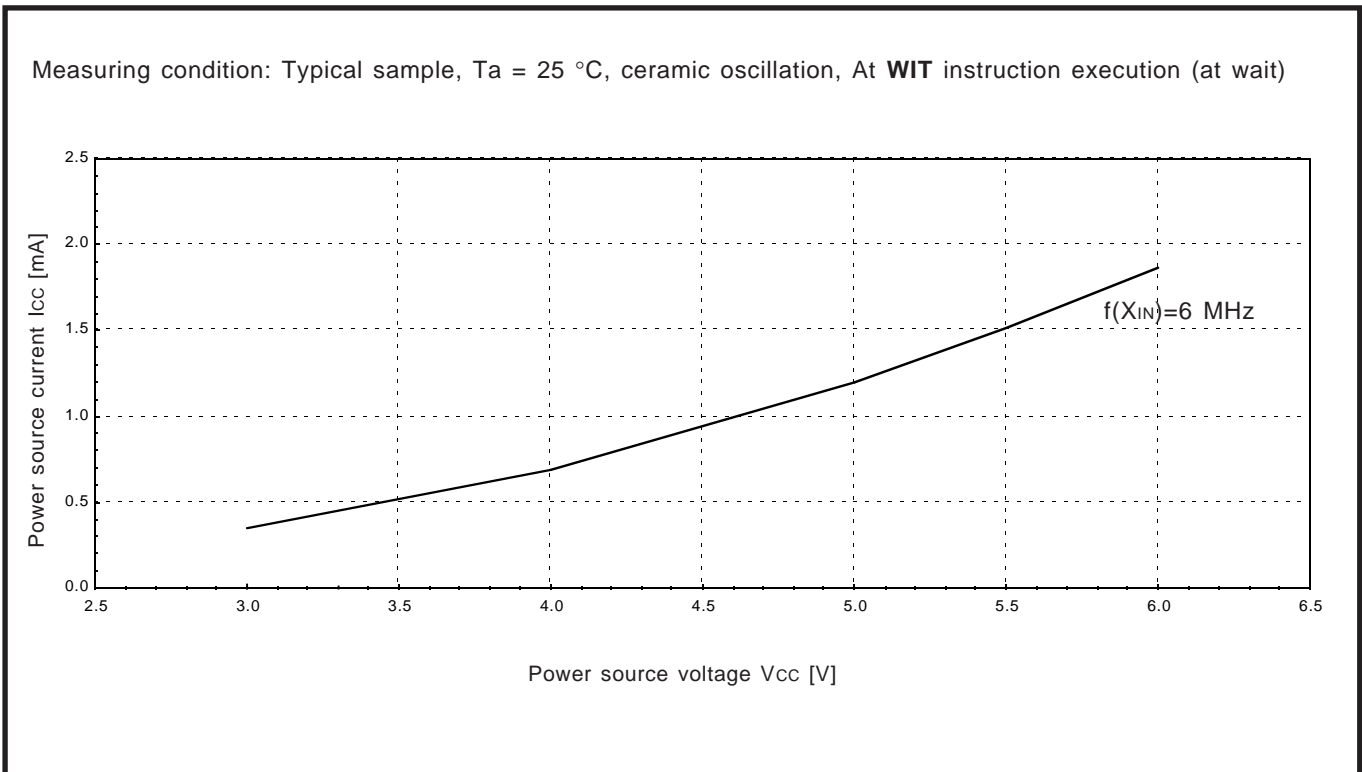


Fig. 3.2.2 Icc-Vcc characteristic example (at WIT instruction execution)

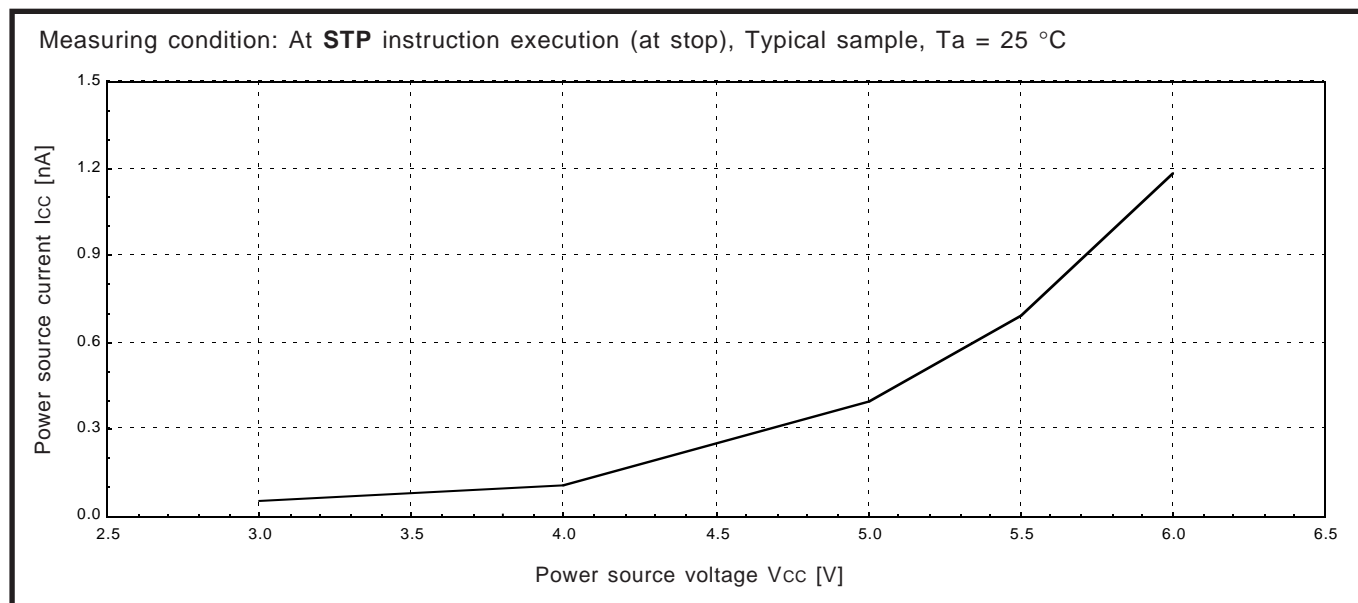


Fig. 3.2.3  $I_{cc}$ - $V_{cc}$  characteristic example (At **STP** instruction execution,  $T_a = 25\text{ }^{\circ}\text{C}$ )

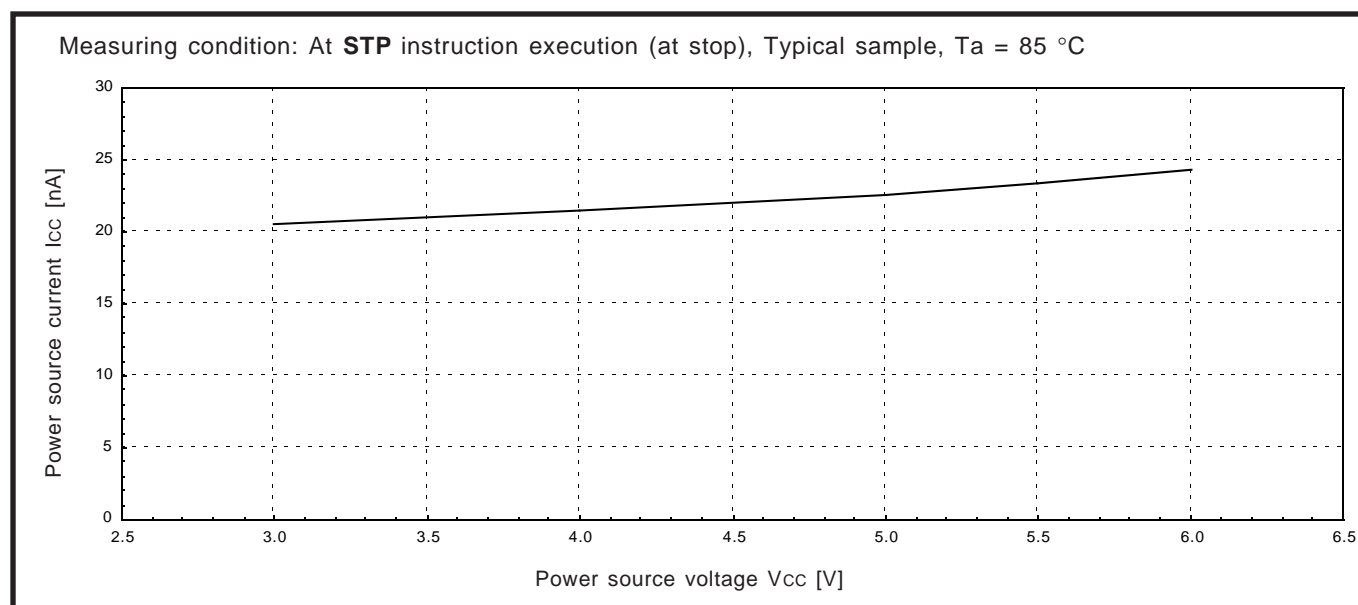


Fig. 3.2.4  $I_{cc}$ - $V_{cc}$  characteristic example (At **STP** instruction execution,  $T_a = 85\text{ }^{\circ}\text{C}$ )

# APPENDIX

## 3.2 Typical characteristics

Measuring condition: At **STP** instruction execution, (at USB suspend), Typical sample,  $T_a = 25\text{ }^{\circ}\text{C}$   
Currents flowing to  $1.5\text{ k}\Omega$  pull-up resistor and  $15\text{ k}\Omega$  pull-down resistor connecting D- pin externally are not included

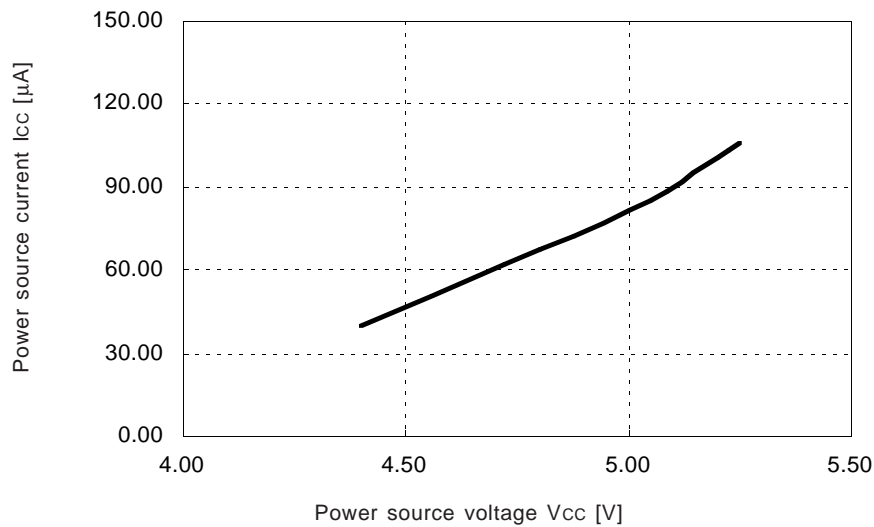


Fig. 3.2.5  $I_{cc}$ - $V_{cc}$  characteristic example (at USB suspend,  $T_a = 25\text{ }^{\circ}\text{C}$ )

Measuring condition: A-D conversion executed/not executed ( $f(X_{IN}) = 6\text{ MHz}$ , in double-speed mode),  
Typical sample,  $T_a = 25\text{ }^{\circ}\text{C}$ , At ceramic oscillation

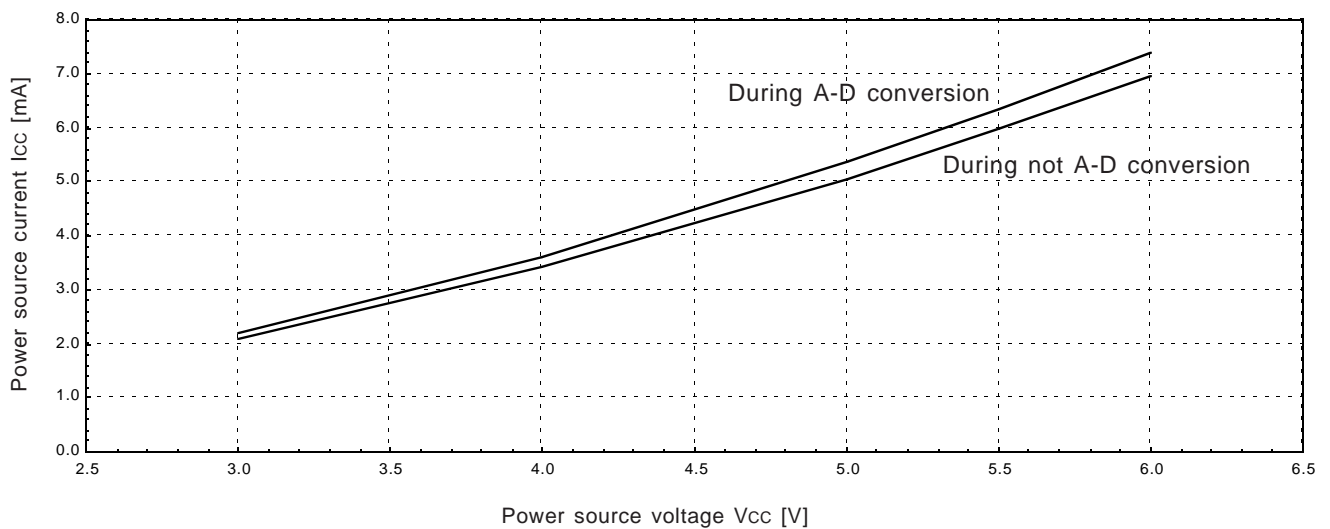


Fig. 3.2.6  $I_{cc}$ - $V_{cc}$  characteristic example (A-D conversion executed/not executed,  $f(X_{IN}) = 6\text{ MHz}$ , in double-speed mode)

### 3.2.2 $V_{OH}$ - $I_{OH}$ characteristic example

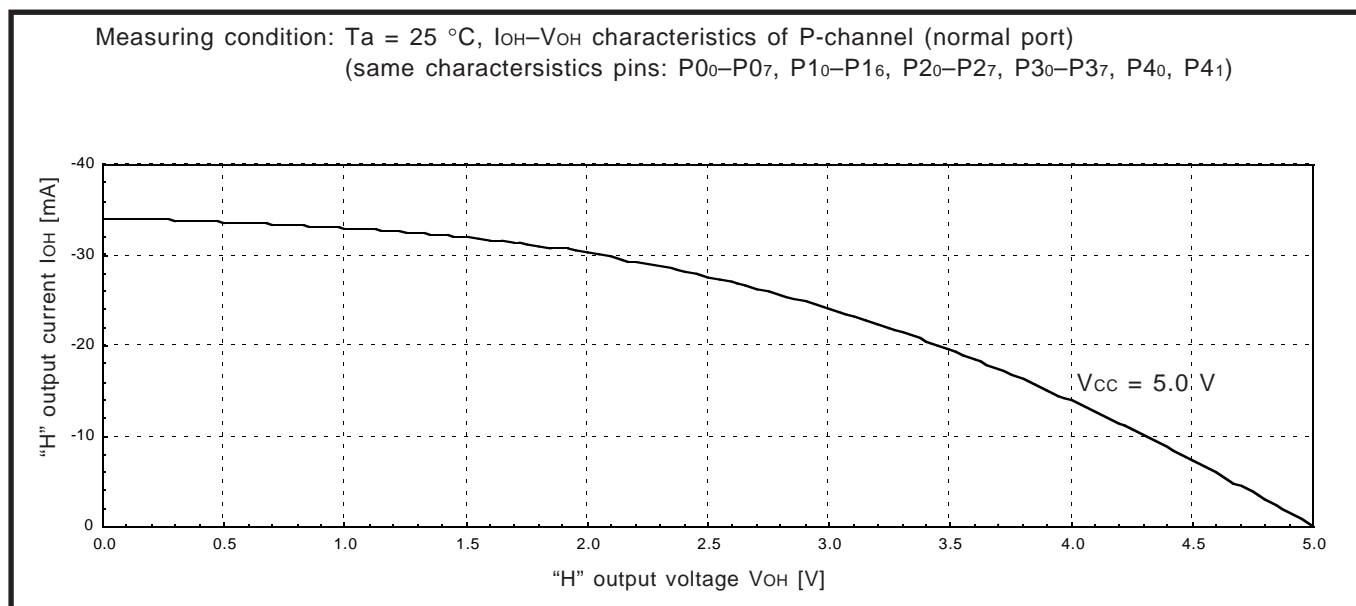


Fig. 3.2.7  $V_{OH}$ - $I_{OH}$  characteristic example of P-channel ( $T_a = 25\text{ }^{\circ}\text{C}$ ): normal port

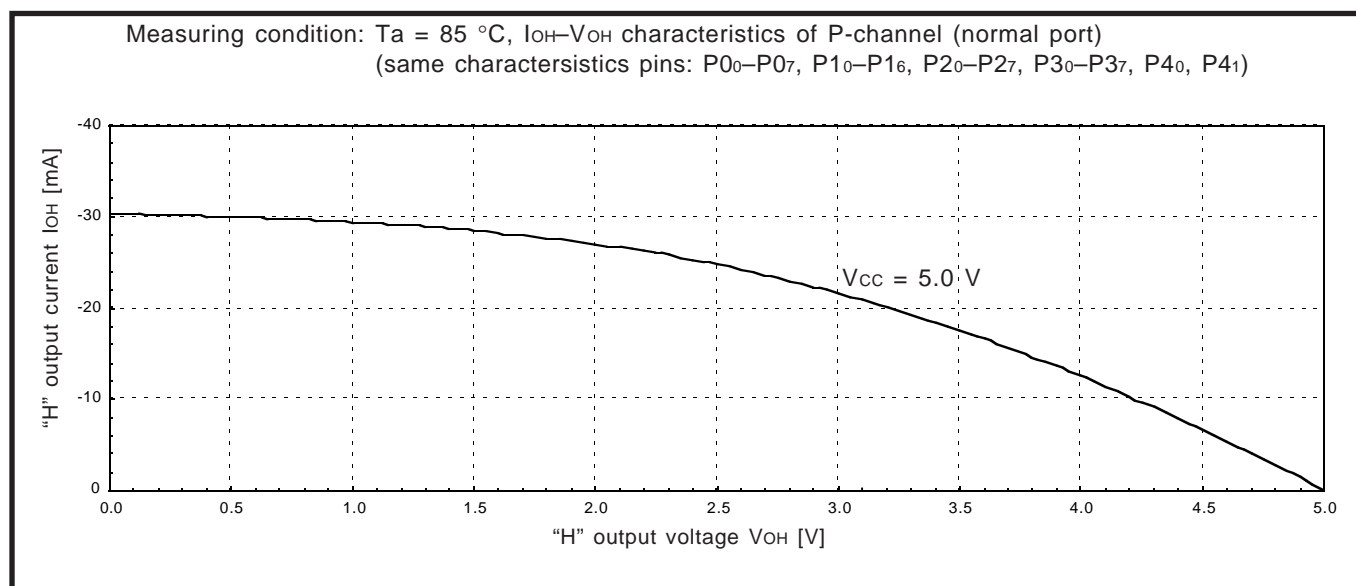


Fig. 3.2.8  $V_{OH}$ - $I_{OH}$  characteristic example of P-channel ( $T_a = 85\text{ }^{\circ}\text{C}$ ): normal port

# APPENDIX

## 3.2 Typical characteristics

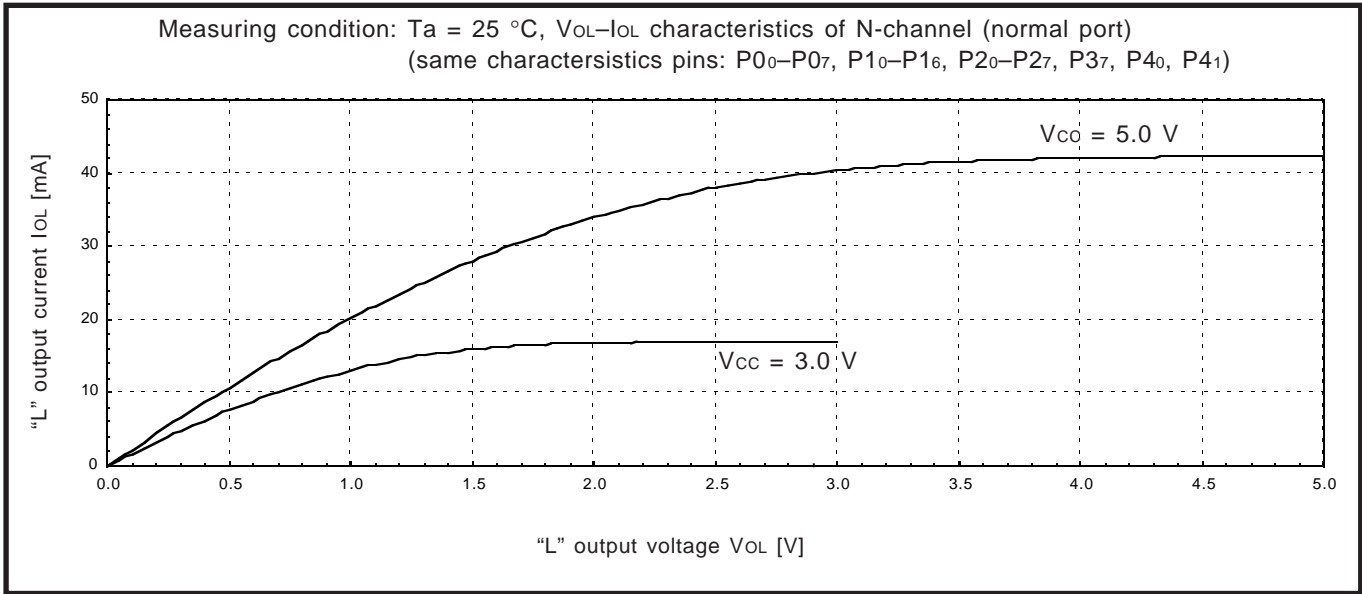


Fig. 3.2.9  $V_{OL}$ - $I_{OL}$  characteristic example of N-channel ( $T_a = 25\text{ }^{\circ}\text{C}$ ): Normal port

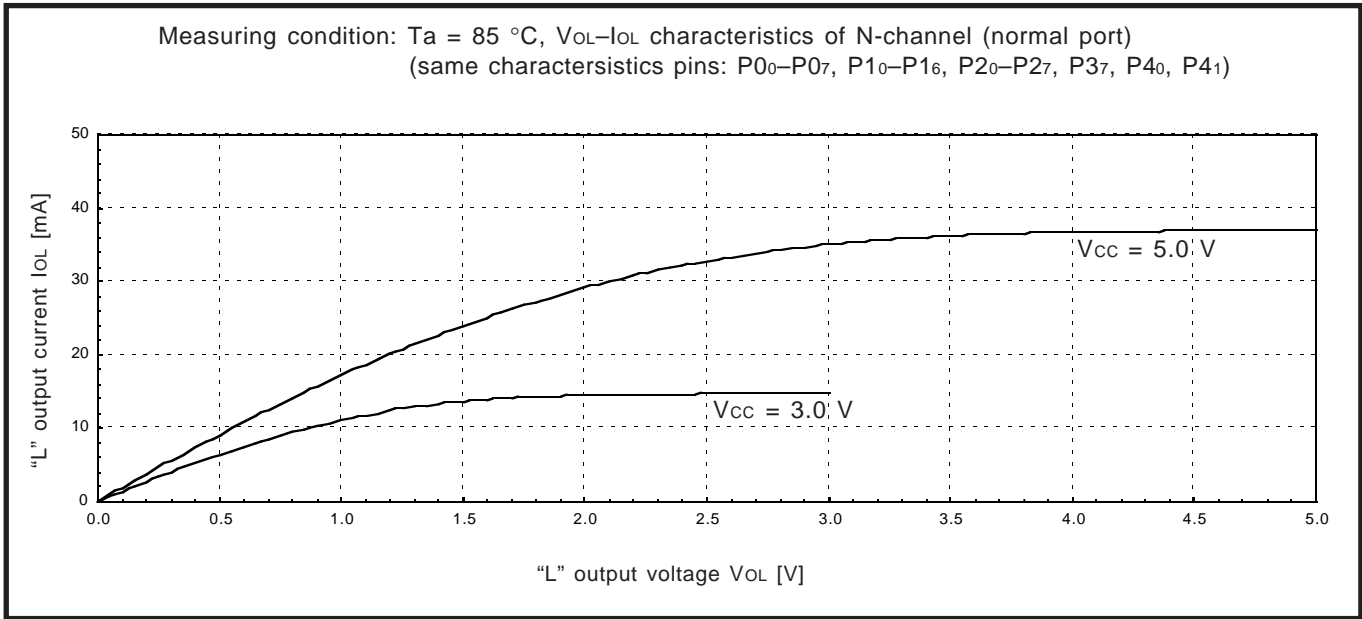
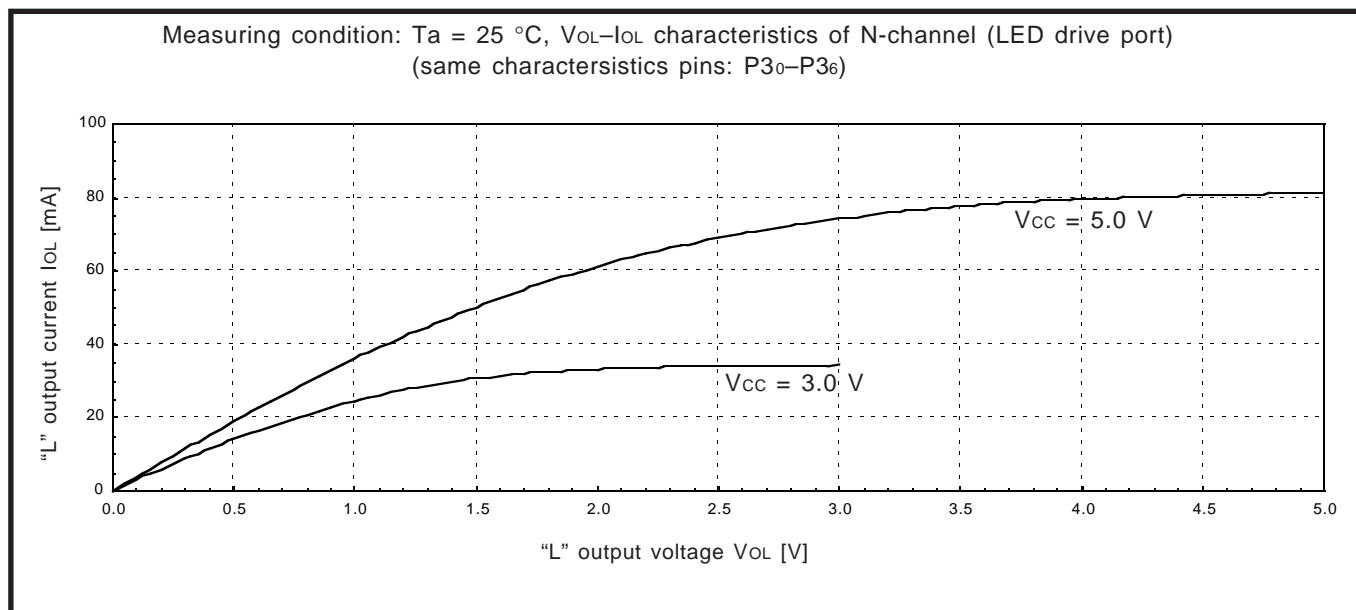
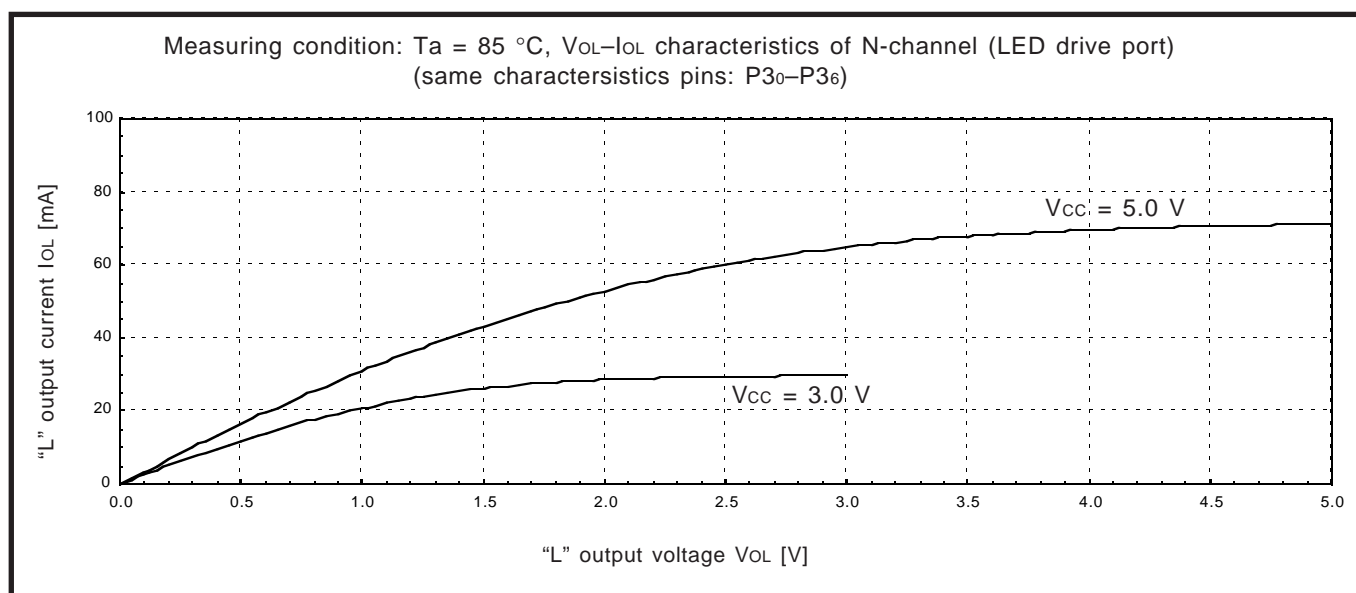


Fig. 3.2.10  $V_{OL}$ - $I_{OL}$  characteristic example of N-channel ( $T_a = 85\text{ }^{\circ}\text{C}$ ): Normal port



**Fig. 3.2.11  $V_{OL}$ - $I_{OL}$  characteristic example of N-channel ( $T_a = 25\text{ }^{\circ}\text{C}$ ): LED drive port**



**Fig. 3.2.12  $V_{OL}$ - $I_{OL}$  characteristic example N-channel ( $T_a = 85\text{ }^{\circ}\text{C}$ ): LED drive port**

# APPENDIX

## 3.2 Typical characteristics

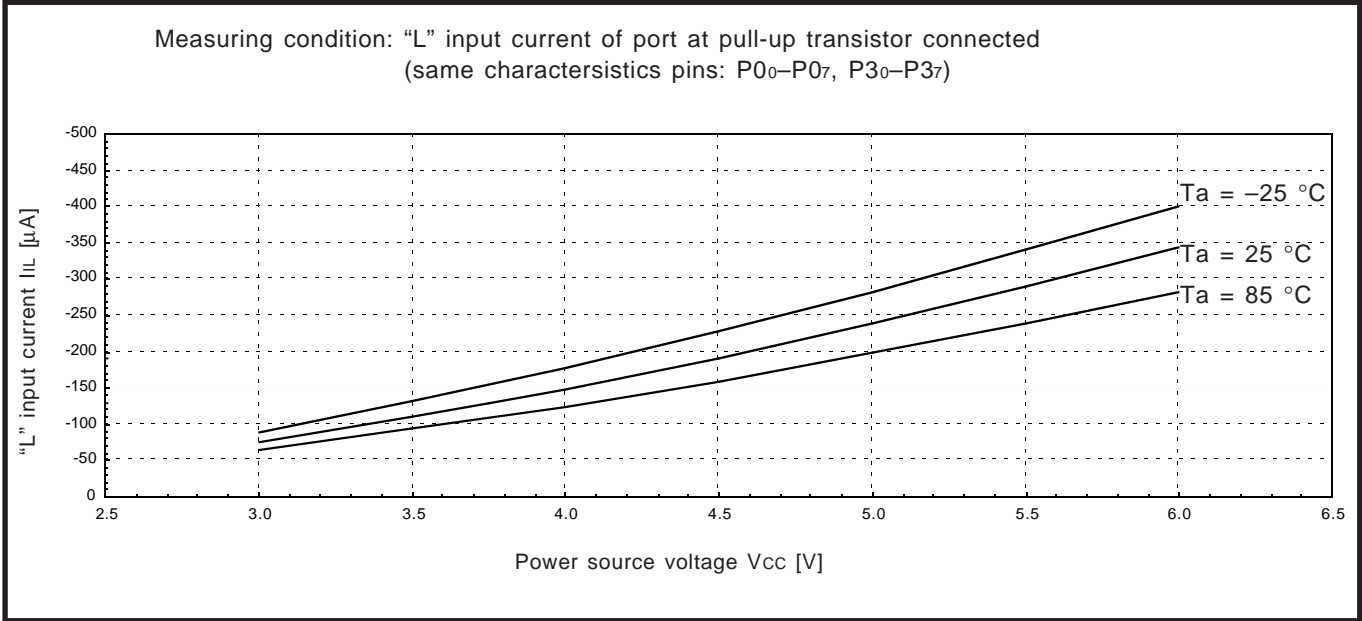


Fig. 3.2.13  $I_{IL}$ - $V_{CC}$  characteristic example when connecting pull-up transistor

### 3.2.3 A-D conversion typical characteristics example

#### (1) Definition of A-D conversion accuracy

The A-D conversion accuracy is defined below (refer to Fig. 3.2.14).

##### ●Relative accuracy

###### ① Zero transition voltage ( $V_{OT}$ )

This means an analog input voltage when the actual A-D conversion output data changes from “0” to “1.”

###### ② Full-scale transition voltage ( $V_{FST}$ )

This means an analog input voltage when the actual A-D conversion output data changes from “1023” to “1022.”

###### ③ Non-linearity error

This means a deviation from the line between  $V_{OT}$  and  $V_{FST}$  of a converted value between  $V_{OT}$  and  $V_{FST}$ .

###### ④ Differential non-linearity error

This means a deviation from the input potential difference required to change a converted value between  $V_{OT}$  and  $V_{FST}$  by 1 LSB of the 1 LSB at the relative accuracy.

##### ●Absolute accuracy

This means a deviation from the ideal characteristics between 0 to  $V_{REF}$  of actual A-D conversion characteristics.

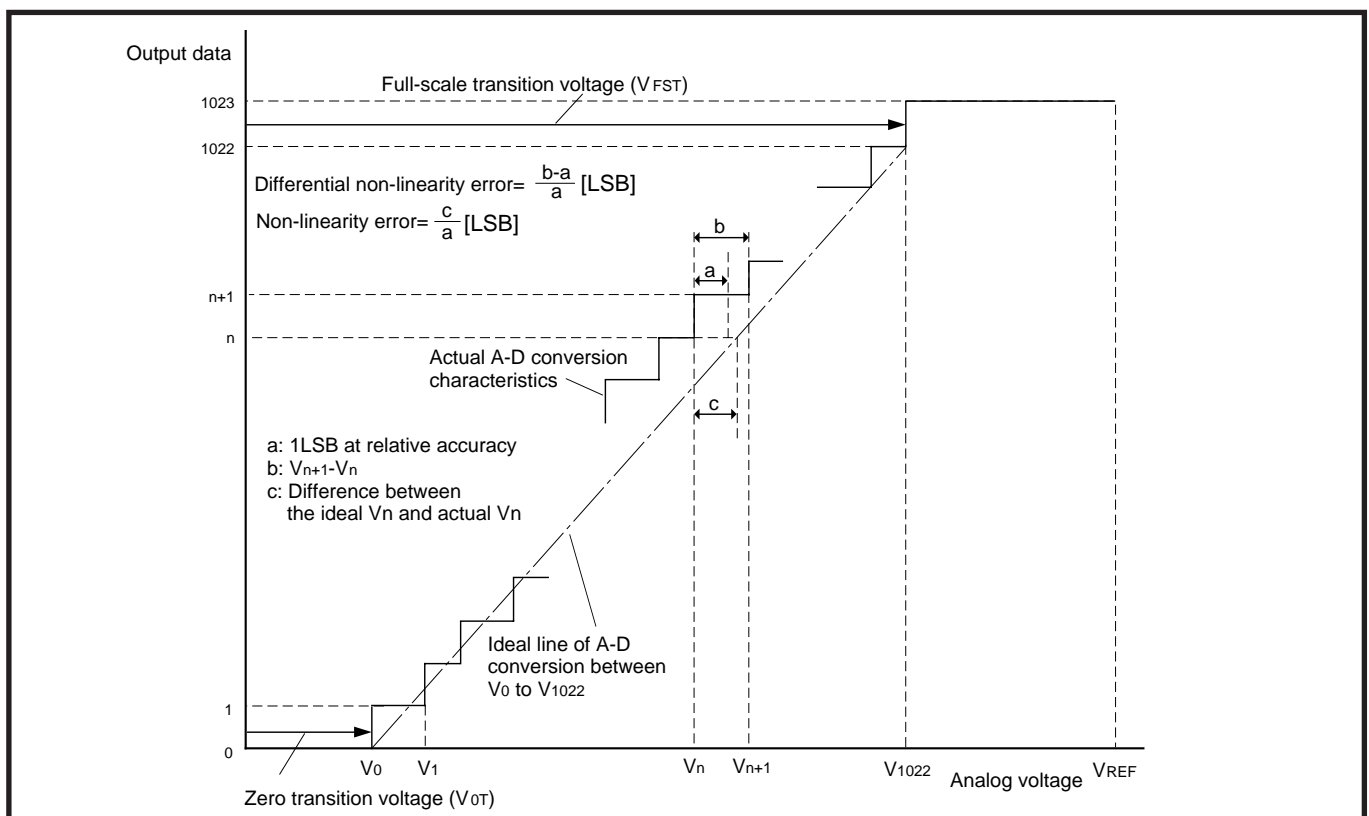


Fig. 3.2.14 Definition of A-D conversion accuracy

$V_n$ : Analog input voltage when the output data changes from “ $n$ ” to “ $n + 1$ ” ( $n = 0$  to 1022)

- 1 LSB at relative accuracy  $\rightarrow \frac{V_{FST} - V_{OT}}{1022}$  (V)
- 1 LSB at absolute accuracy  $\rightarrow \frac{V_{REF}}{1024}$  (V)



# APPENDIX

## 3.2 Typical characteristics

### M37532M4-XXXFP A-D CONVERTER STEP WIDTH MEASUREMENT

- VCC = 5 [V]
- VREF = 5 [V]
- XIN = 6 [MHz]
- Temp. = 25 [°C]
- CPU mode = double-speed mode

- Zero transition voltage: 6.714 mV
- Full-scale transition voltage: 4994.812 mV
- Differential non-linearity error: 1.373 mV (0.281 LSB)
- Non-linearity error: -5.201 mV (-1.066 LSB)

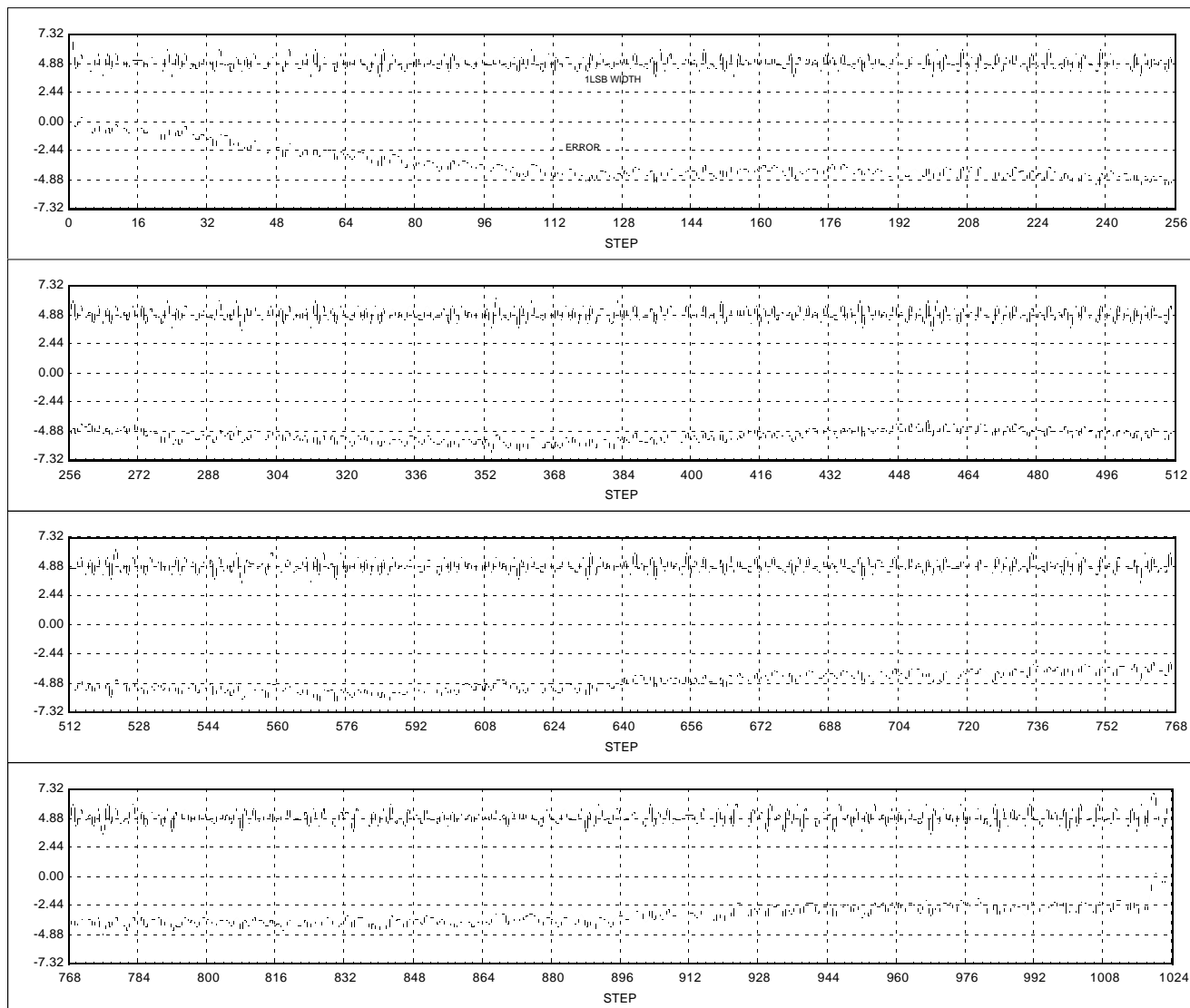


Fig. 3.2.15 A-D conversion typical characteristic example

### 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

For the products able to switch the external interrupt detection edge, switch it as the following sequence.

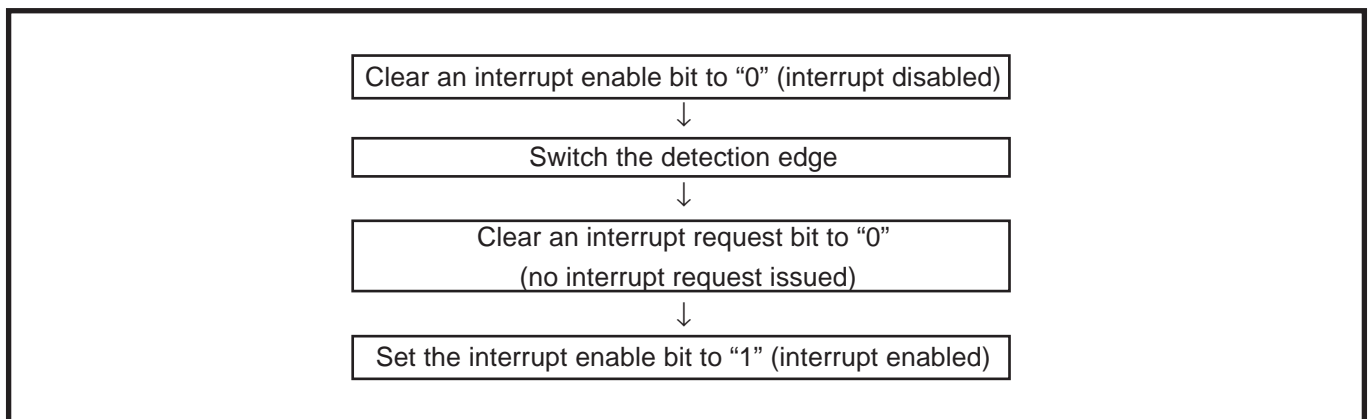


Fig. 3.3.1 Sequence of switch the detection edge

##### ● 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.

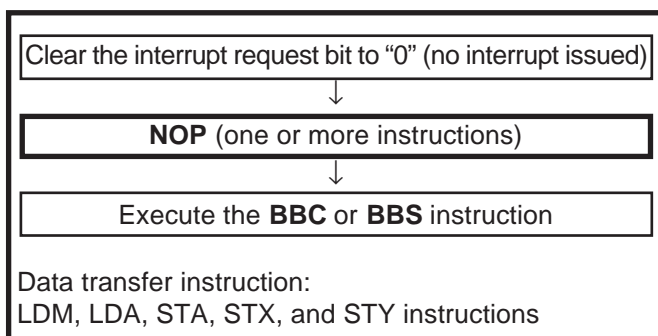


Fig. 3.3.2 Sequence of check of interrupt request bit

##### ● 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.

# APPENDIX

## 3.3 Notes on use

### (4) Structure of interrupt control register 1

Fix the bit 7 of the interrupt control register 1 to “0”. Figure 3.3.3 shows the structure of the interrupt control register 1.

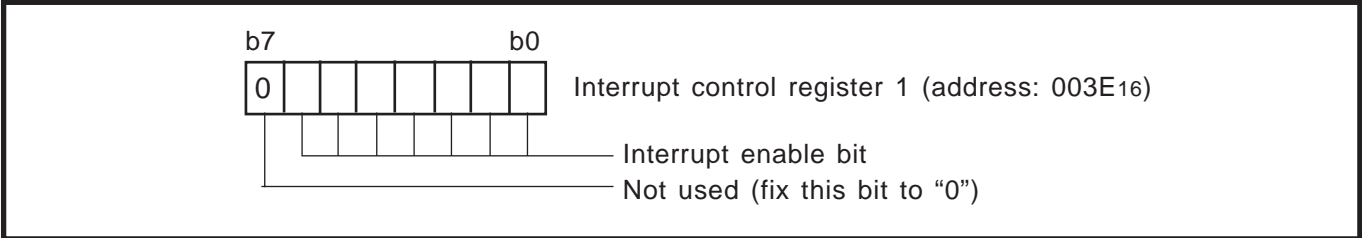


Fig. 3.3.3 Structure of interrupt control register 1

### 3.3.2 Notes on serial I/O

#### (1) Handling of serial I/O1 clear

When serial I/O1 is set again or the transmit/receive operation is stopped/restarted while serial I/O1 is operating, clear the serial I/O1 as shown in Figure 3.3.4.

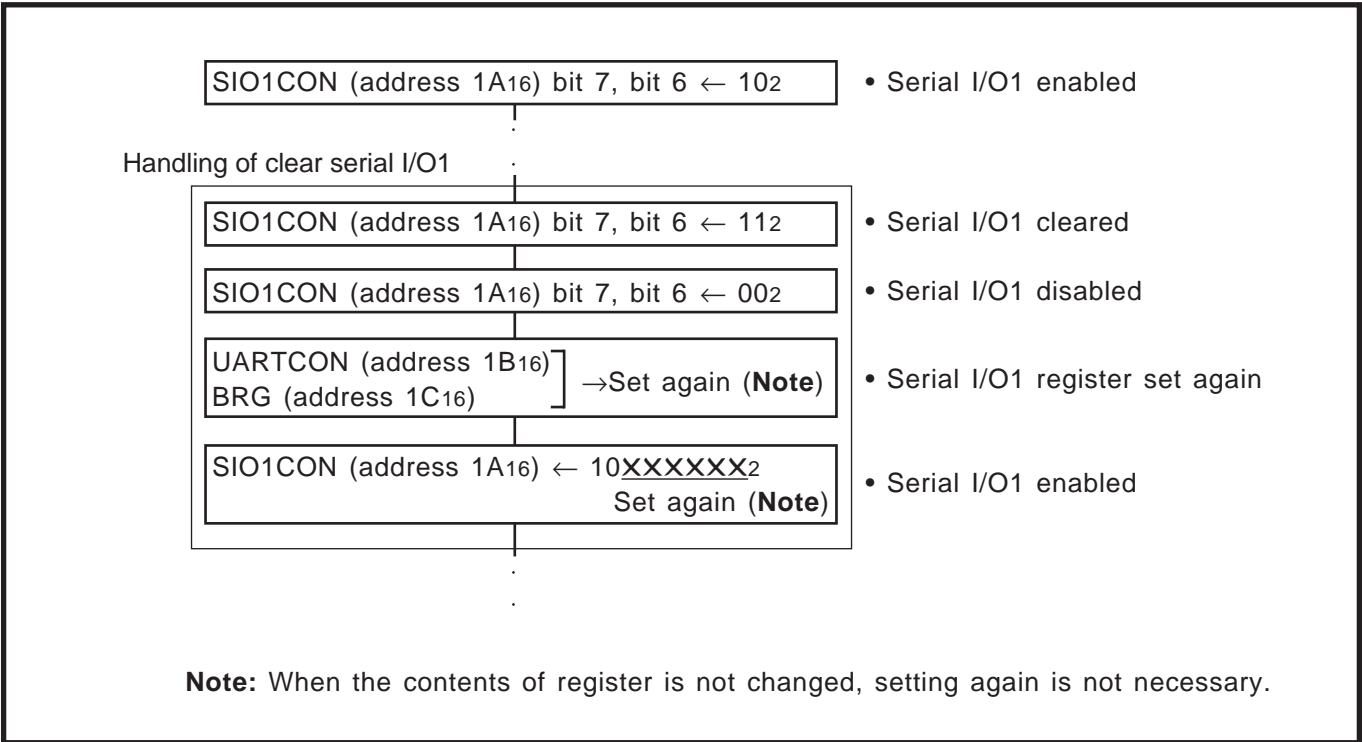


Fig. 3.3.4 Sequence of clearing serial I/O

#### (2) 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.

#### (3) Writing transmit data

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

### (4) Serial I/O2 transmit/receive shift completion flag

- The transmit/receive shift completion flag of the serial I/O2 control register is set to “1” after completing transmit/receive shift. In order to set this flag to “0”, write data (dummy data at reception) to the serial I/O2 register by program.
- Bit 7 of the serial I/O2 control register is set to “1” a half cycle (of the shift clock) earlier than completion of shift operation. Accordingly, when using this bit to confirm shift completion, a half cycle or more of the shift clock must pass after confirming that this bit is set to “1”, before performing read/write to the serial I/O2 register.

### 3.3.3 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 $\mu$ F to 1 $\mu$ 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) 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

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

#### (1) Connecting capacitor

In case where the  $\overline{RESET}$  signal rise time is long, connect a ceramic capacitor or others across the  $\overline{RESET}$  pin and the  $V_{SS}$  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  $\overline{RESET}$  pin, it may cause a microcomputer failure.

# APPENDIX

## 3.3 Notes on use

---

### 3.3.5 Notes on input and output pins

#### (1) Notes in stand-by state

In stand-by state\*<sup>1</sup> for low-power dissipation, do not make input levels of an input port and an I/O port “undefined”.

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

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.

\*<sup>1</sup> 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\*<sup>2</sup>, 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.

\*<sup>2</sup> bit managing instructions : **SEB**, and **CLB** instructions

### 3.3.6 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".

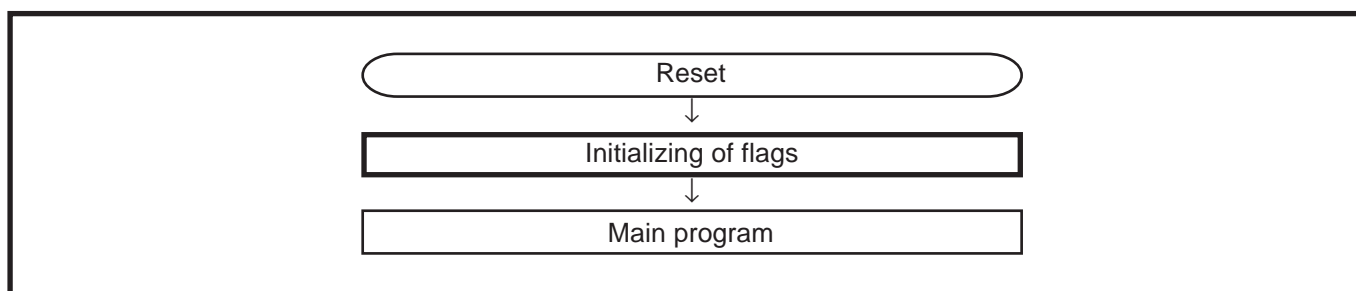


Fig. 3.3.5 Initialization of processor status register

##### 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.

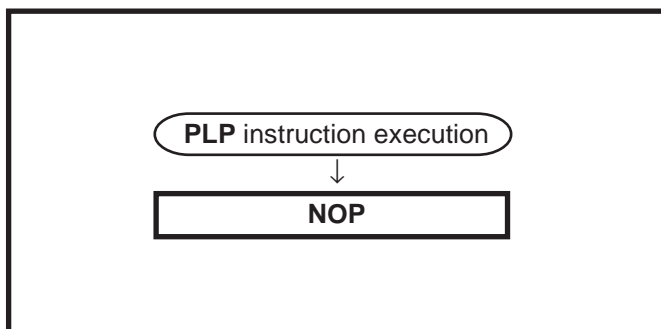


Fig. 3.3.6 Sequence of PLP instruction execution

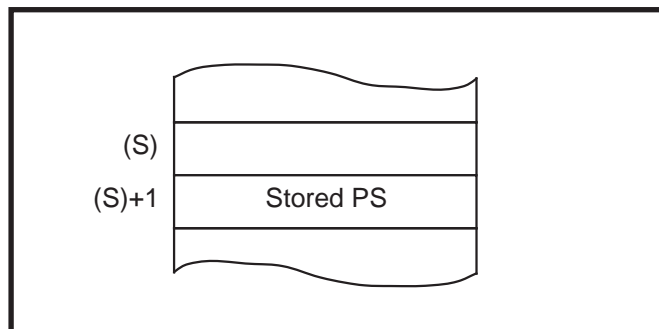


Fig. 3.3.7 Stack memory contents after PHP instruction execution

# APPENDIX

## 3.3 Notes on use

---

### (2) 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.

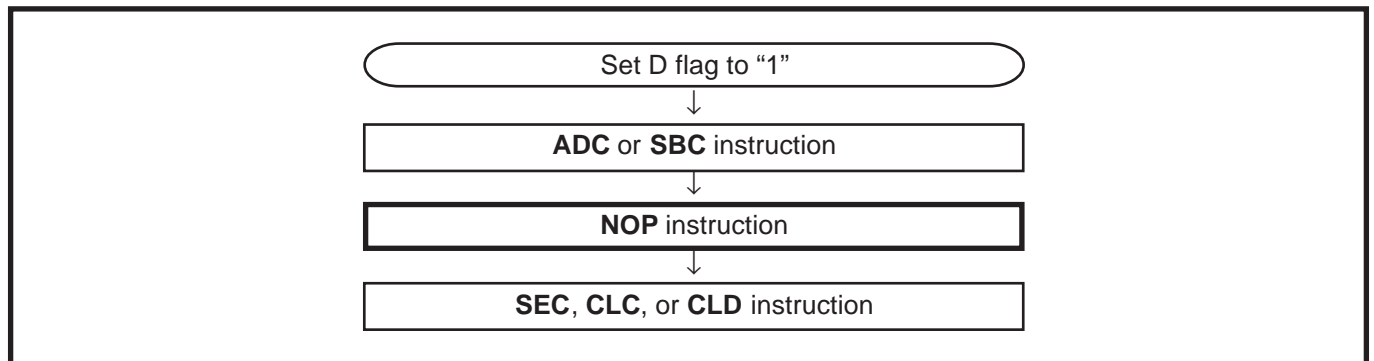


Fig. 3.3.8 Status flag at decimal calculations

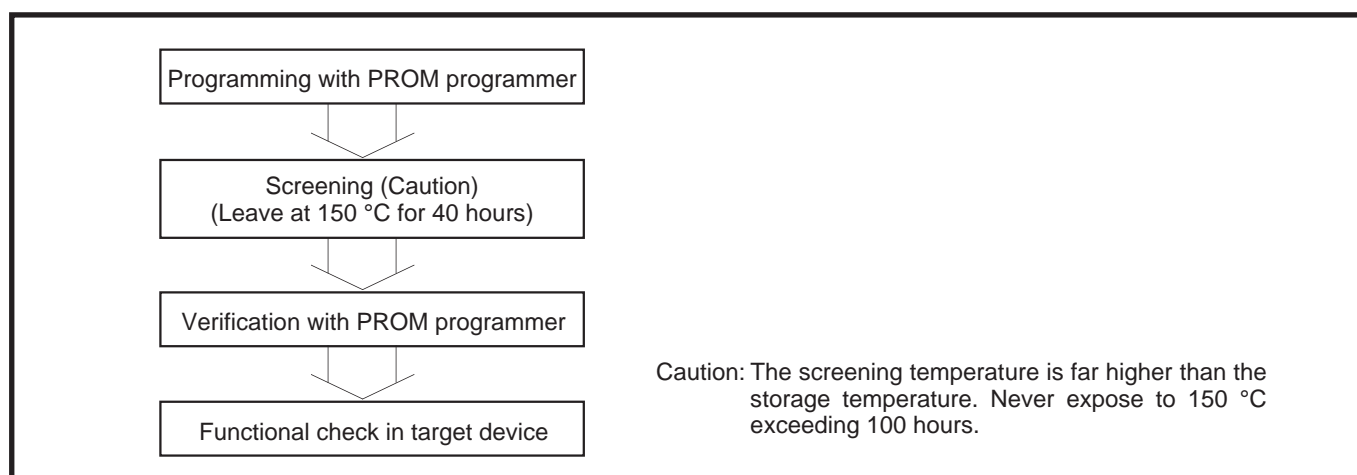
### (3) JMP instruction

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

### 3.3.7 Programming and test of built-in PROM version

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

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 3.3.9 before actual use are recommended.



**Fig. 3.3.9 Programming and testing of One Time PROM version**

### 3.3.8 Notes on built-in PROM version

#### (1) Programming adapter

Use a special programming adapter shown in Table 3.3.1 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
M37536E8SP (One Time PROM version shipped in blank)	PCA7435SP
M37532E8FP (One Time PROM version shipped in blank)	PCA7435FP



# 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 VPP pin (is also used as the CNVss pin), or the product may be permanently damaged.

- Programming voltage: 12.5 V
- Setting of PROM programmer switch: refer to Table 3.3.2.

**Table 3.3.2 PROM programmer address setting**

Product name format	PROM programmer start address	PROM programmer end address
M37536E8SP	Address 0C080 <sub>16</sub> ( <b>Note</b> )	Address 0FFFD <sub>16</sub> ( <b>Note</b> )
M37532E8FP		

**Note:** Addersses C080<sub>16</sub> to FFFD<sub>16</sub> in the built-in PROM corresponds to addresses 0C080<sub>16</sub> to 0FFFD<sub>16</sub> in the PROM programmer.

### 3.3.9 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 $\Omega$  to 10 k $\Omega$ .

Ports that permit the selecting of a built-in pull-up resistor can also use this resistor. 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 $\Omega$  to 10 k $\Omega$ .

Ports that permit the selecting of a built-in pull-up resistor can also use this resistor. 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

## 3.3 Notes on use

---

### 3.3.10 Notes on CPU mode register

#### (1) Switching method of CPU mode register after releasing reset

Switch the CPU mode register (CPUM) at the head of program after releasing reset in the following method.

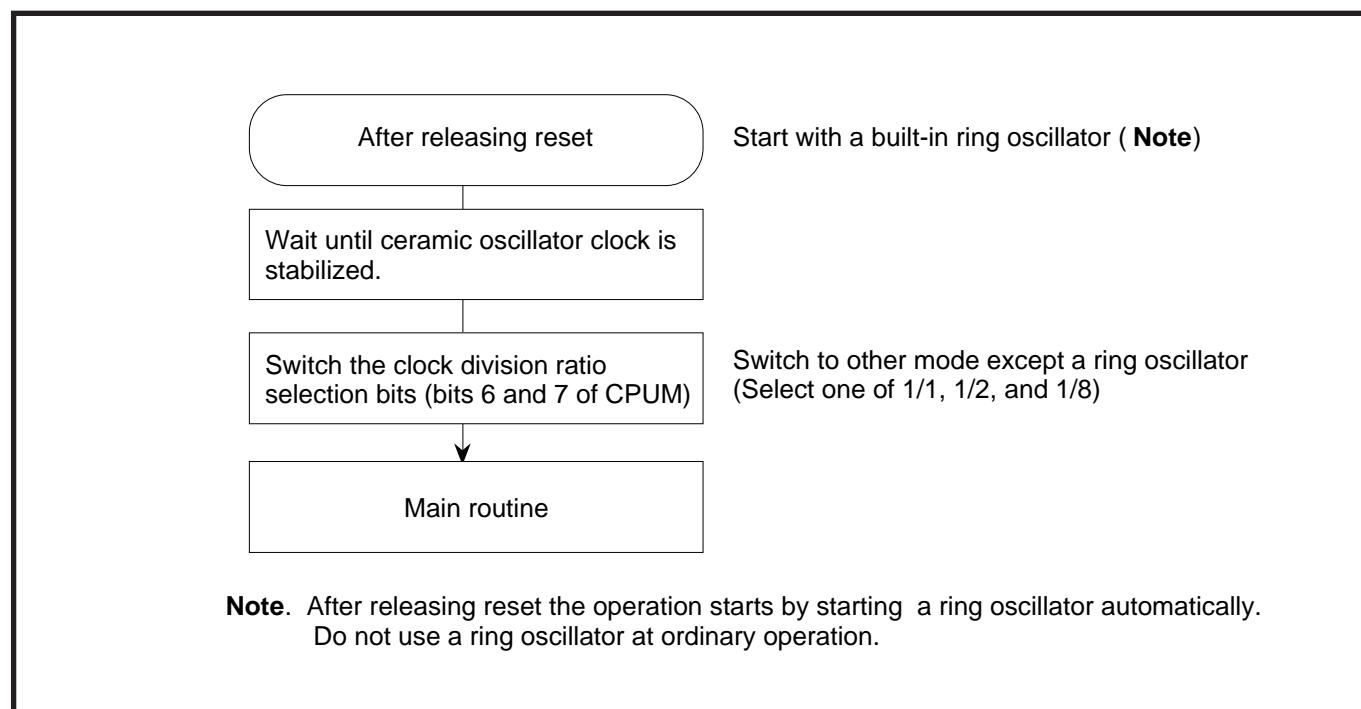


Fig. 3.3.10 Switching method of CPU mode register

#### 3.3.11 Notes on using 32-pin version

- Do not change the P35, P36 pull-up control bit of the pull-up control register from the initial value "1".
- Do not write to "1" to the serial I/O1 or INT1 interrupt selection bit of the interrupt edge selection register.

### 3.4 Countermeasures against noise

#### 3.4.1 Shortest wiring length

##### (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.

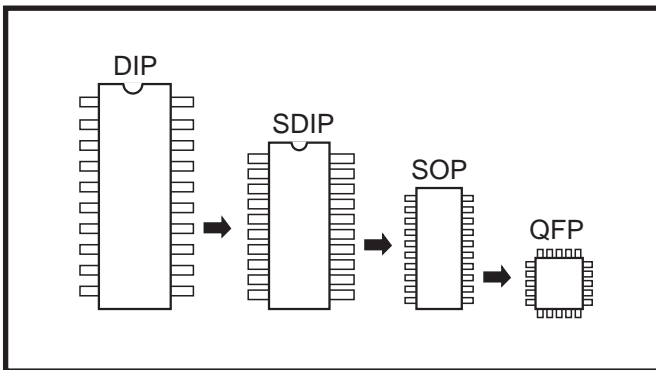


Fig. 3.4.1 Selection of packages

##### (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  $V_{SS}$  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.

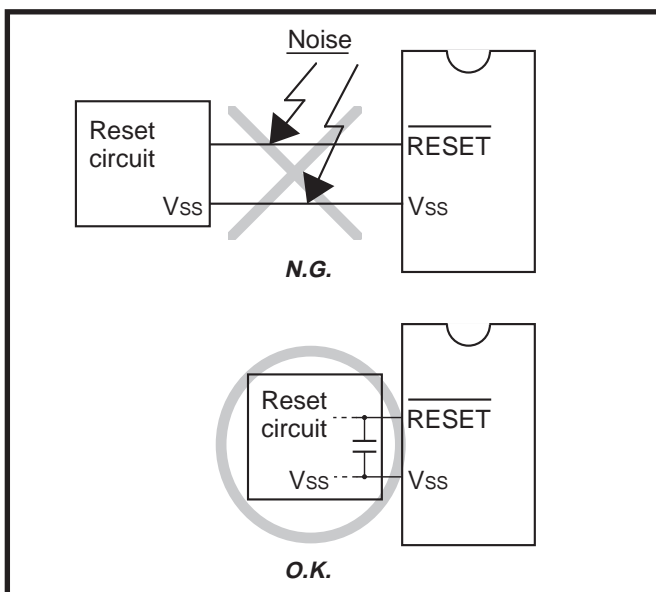


Fig. 3.4.2 Wiring for the  $\overline{\text{RESET}}$  pin

# APPENDIX

## 3.4 Countermeasures against noise

### (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.

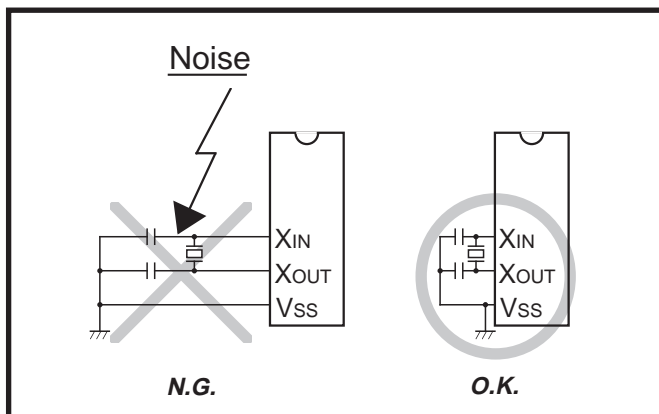


Fig. 3.4.3 Wiring for clock I/O pins

### (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.

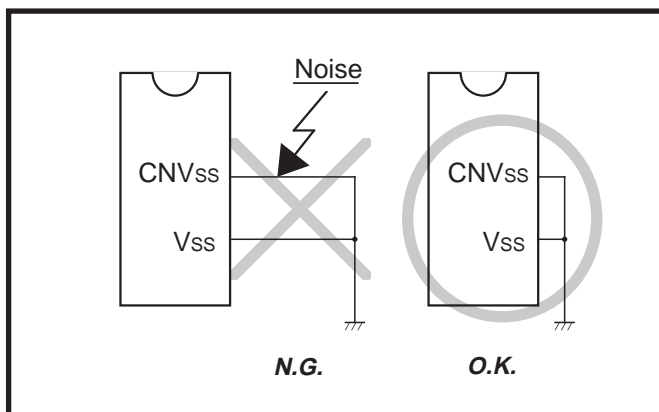


Fig. 3.4.4 Wiring for CNVss pin

### (5) Wiring to VPP pin of One Time PROM 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 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.

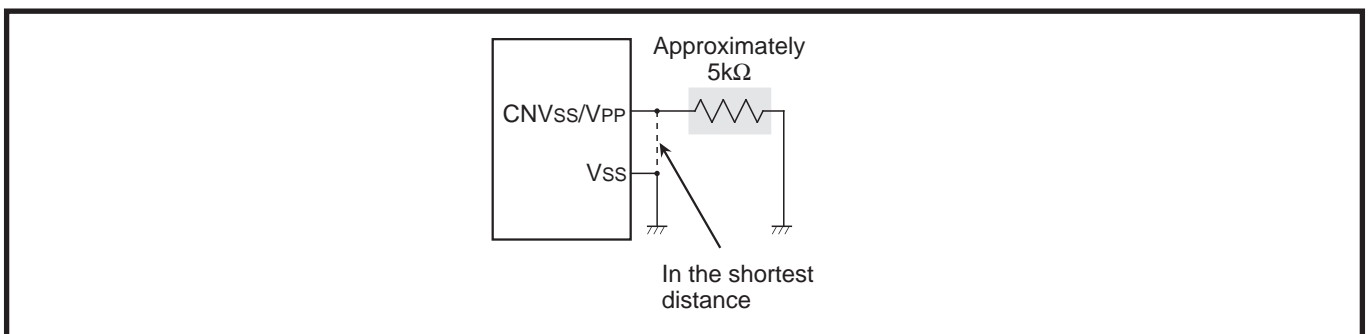


Fig. 3.4.5 Wiring for the VPP pin of the One Time PROM

### 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.

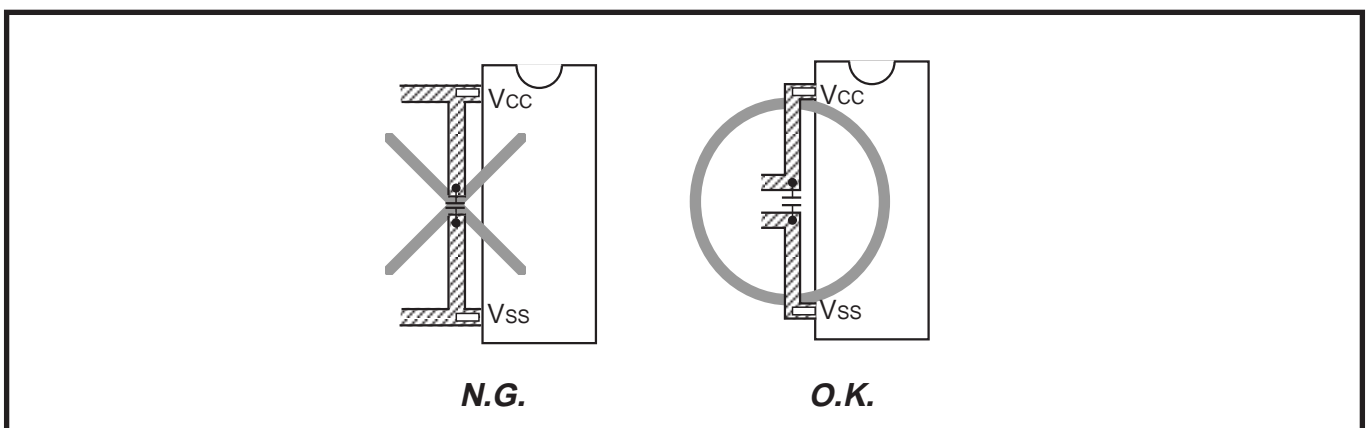


Fig. 3.4.6 Bypass capacitor across the Vss line and the Vcc line

# APPENDIX

## 3.4 Countermeasures against noise

### 3.4.3 Wiring to analog input pins

- Connect an approximately  $100\ \Omega$  to  $1\ \text{k}\Omega$  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  $1000\ \text{pF}$  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.

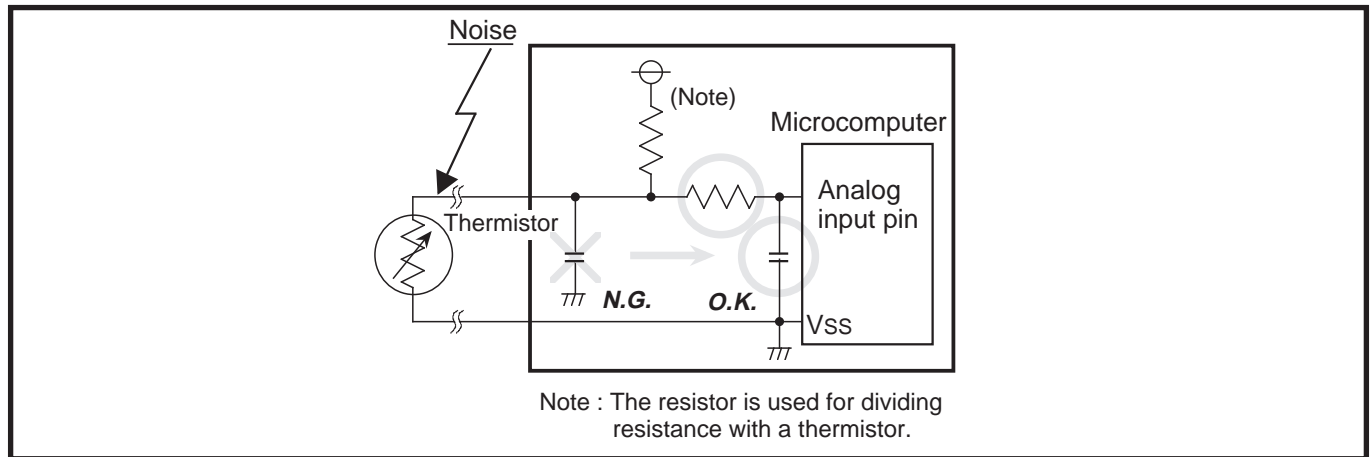


Fig. 3.4.7 Analog signal line and a resistor and a capacitor

### 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.

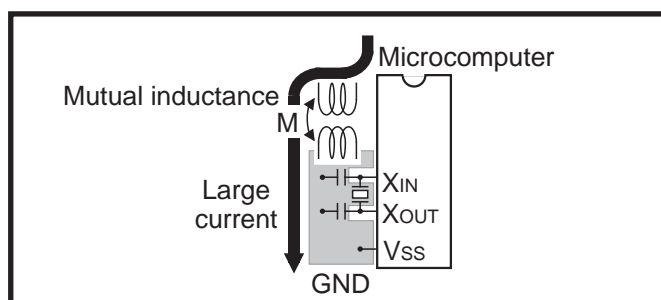


Fig. 3.4.8 Wiring for a large current signal line

### (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.

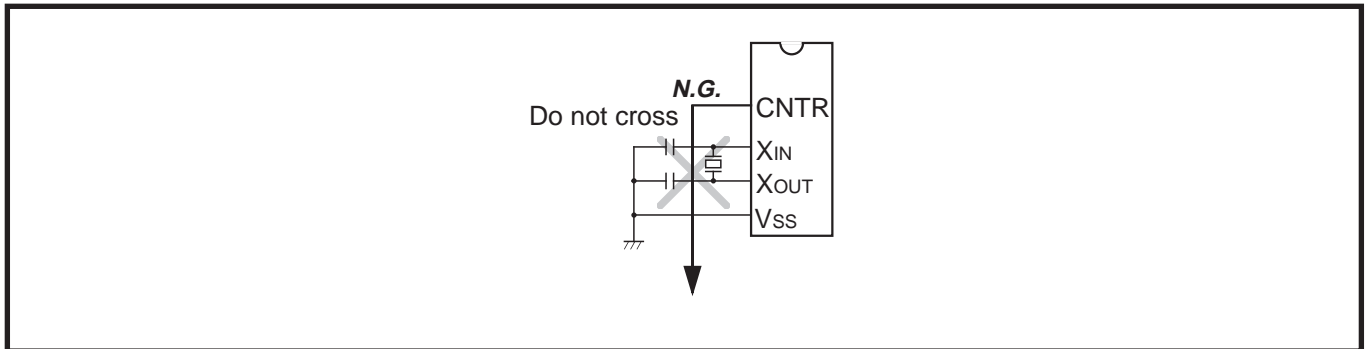


Fig. 3.4.9 Wiring of signal lines where potential levels change frequently

### (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.

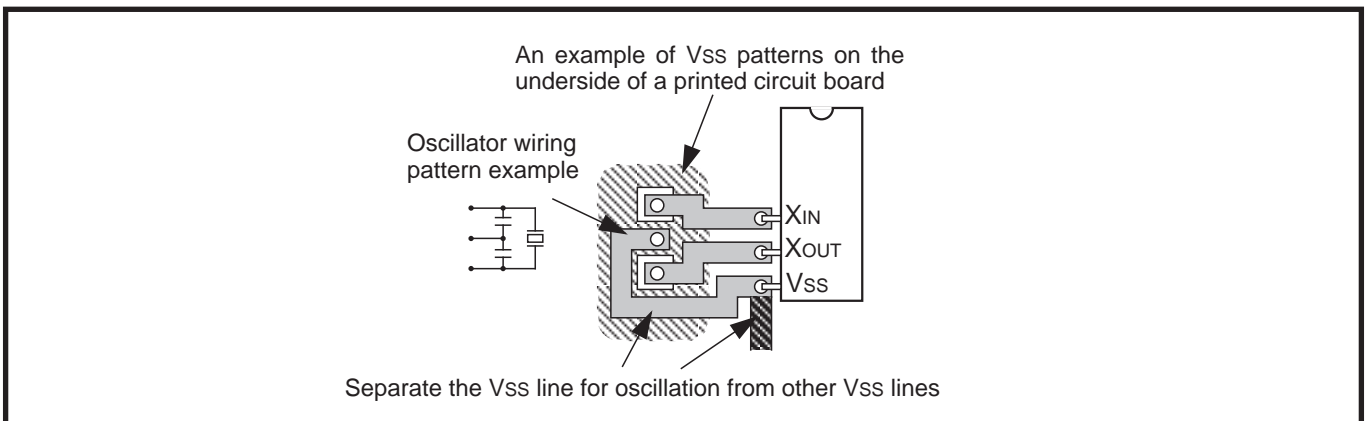


Fig. 3.4.10 Vss pattern on the underside of an oscillator



# APPENDIX

## 3.4 Countermeasures against noise

### 3.4.5 Setup for I/O ports

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

#### <Hardware>

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

#### <Software>

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

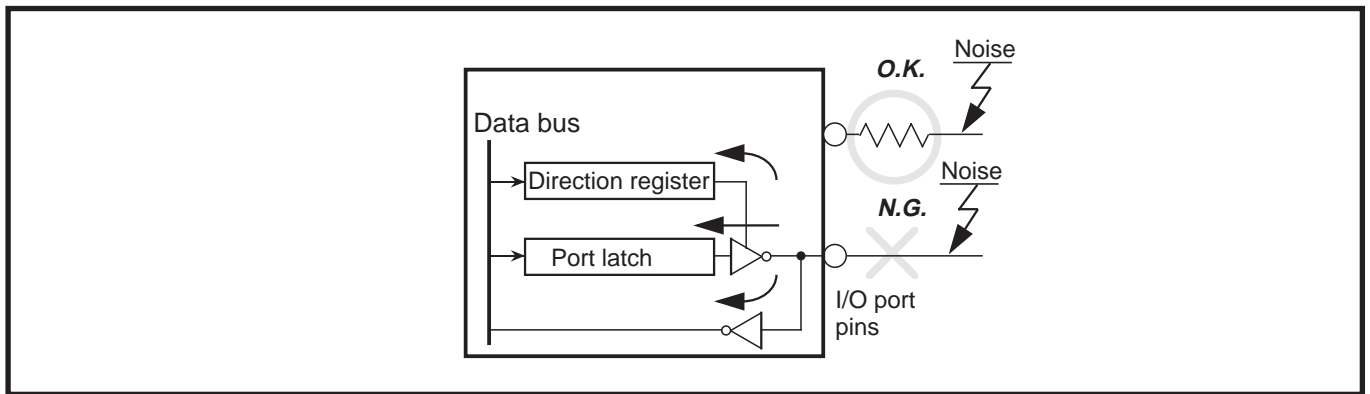


Fig. 3.4.11 Setup for I/O ports

### 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.

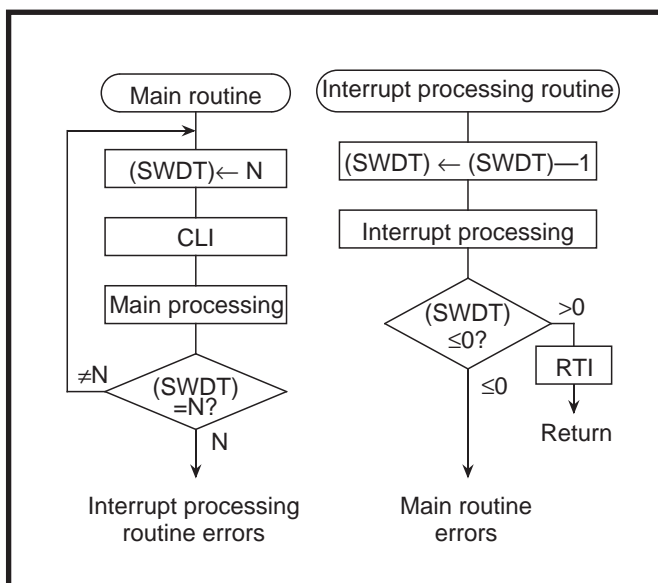


Fig. 3.4.12 Watchdog timer by software

# APPENDIX

## 3.5 List of registers

## 3.5 List of registers

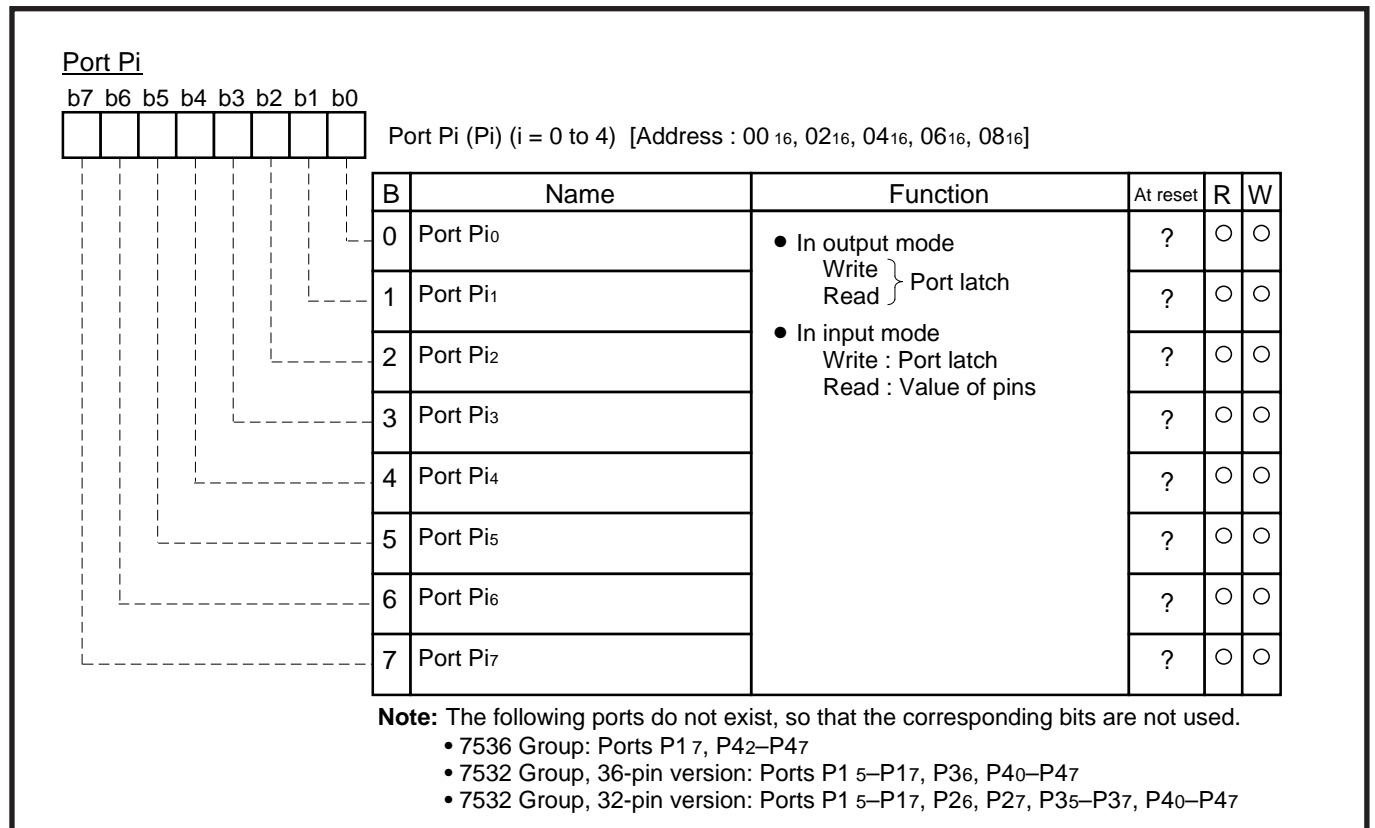


Fig. 3.5.1 Structure of Port Pi (i = 0 to 4)

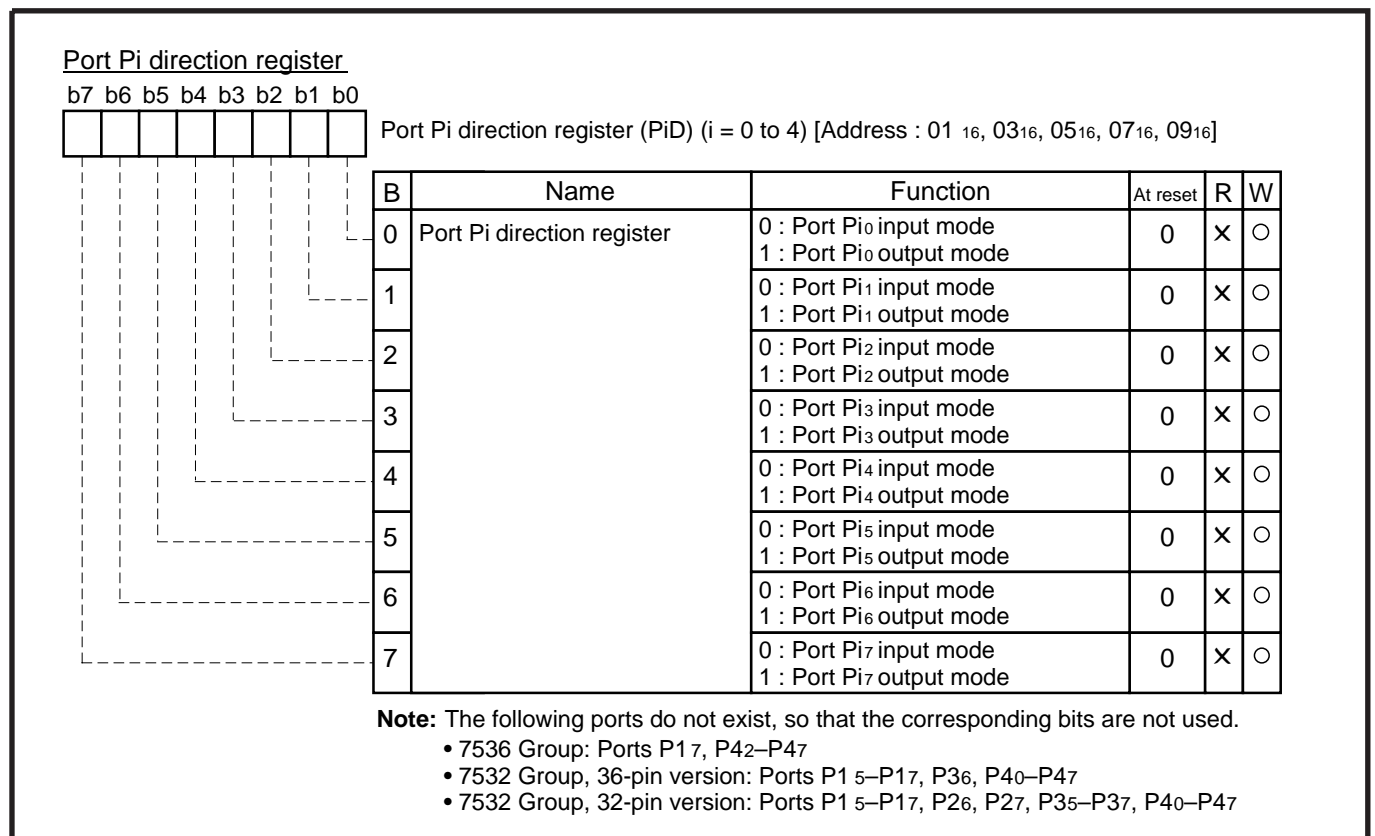


Fig. 3.5.2 Structure of Port Pi direction register (i = 0 to 4)

### Pull-up control register

b7 b6 b5 b4 b3 b2 b1 b0



Pull-up control register (PULL) [Address : 16<sub>16</sub>]

B	Name	Function	At reset	R	W
0	P0 <sub>0</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
1	P0 <sub>1</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
2	P0 <sub>2</sub> , P0 <sub>3</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
3	P0 <sub>4</sub> – P0 <sub>7</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
4	P3 <sub>0</sub> – P3 <sub>3</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
5	P3 <sub>4</sub> pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
6	P3 <sub>5</sub> , P3 <sub>6</sub> pull-up control bit (Note 2)	0 : Pull-up Off 1 : Pull-up On	1	○	○
7	P3 <sub>7</sub> pull-up control bit (Note 3)	0 : Pull-up Off 1 : Pull-up On	1	○	○

**Notes 1:** Pins set to output are disconnected from the pull-up control.

**2:** • 7532 Group, 36-pin version: P3<sub>6</sub> is not existed.

• 7532 Group, 32-pin version: Not used.

**3:** 7532 Group, 32-pin version: Not used.

**Fig. 3.5.3 Structure of Pull-up control register**

### Port P1P3 control register

b7 b6 b5 b4 b3 b2 b1 b0



Port P1P3 control register (P1P3C) [Address : 17<sub>16</sub>]

B	Name	Function	At reset	R	W
0	P3 <sub>7</sub> /INT <sub>0</sub> input level selection bit (Note 1)	0 : CMOS level 1 : TTL level	0	○	○
1	P3 <sub>6</sub> /INT <sub>1</sub> input level selection bit (Note 2)	0 : CMOS level 1 : TTL level	0	○	○
2	P1 <sub>0</sub> , P1 <sub>2</sub> , P1 <sub>3</sub> input level selection bit	0 : CMOS level 1 : TTL level	0	○	○
3	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

**Notes 1:** For the 32-pin version of 7532 Group, nothing is allocated for this bit.

This is a write disabled bit.

When this bit is read out, the value is "0".

**2:** For the 32-pin and 36-pin versions of 7532 Group, nothing is allocated for this bit.

This is a write disabled bit.

When this bit is read out, the value is "0".

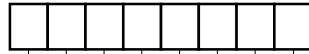
**Fig. 3.5.4 Structure of Port P1P3 control register**

# APPENDIX

## 3.5 List of registers

### Transmit/Receive buffer register

b7 b6 b5 b4 b3 b2 b1 b0



Transmit/Receive buffer register (TB/RB) [Address : 18<sub>16</sub>]

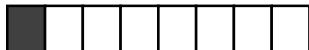
B	Function	At reset	R	W
0	The transmission data is written to or the receive data is read out from this buffer register.	?	○	○
1	• At writing: A data is written to the transmit buffer register. • At reading: The contents of the receive buffer register are read out.	?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5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

**Note:** The contents of transmit buffer register cannot be read out.  
The data cannot be written to the receive buffer register.

Fig. 3.5.5 Structure of Transmit/Receive buffer register

### UART status register

b7 b6 b5 b4 b3 b2 b1 b0



UART status register (UARTSTS) [Address : 19<sub>16</sub>]

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

Fig. 3.5.6 Structure of UART status register

Refer to “Figure 2.4.6 Description of register structure” for registers relevant to USB.

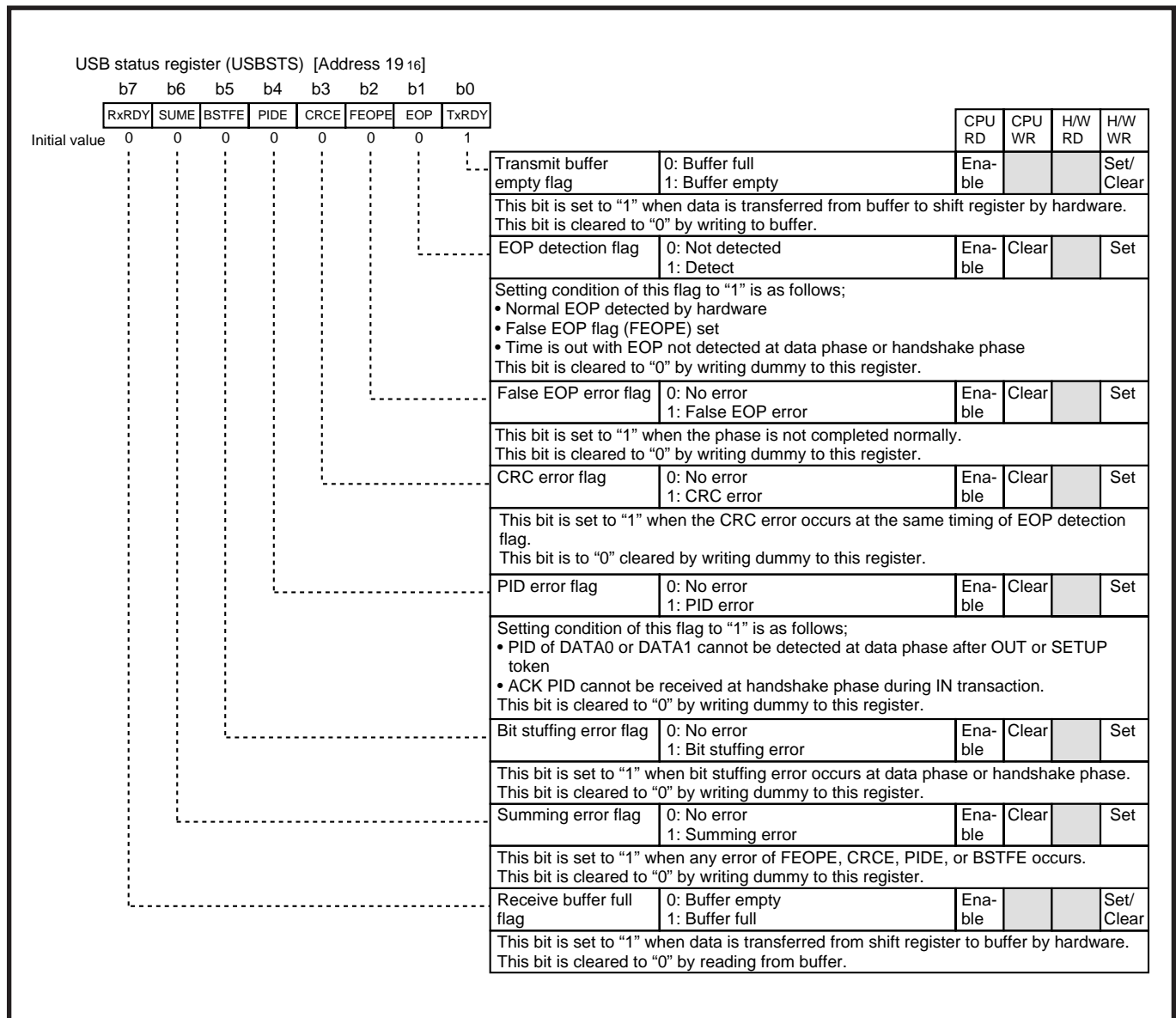


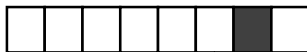
Fig. 3.5.7 Structure of USB status register

# APPENDIX

## 3.5 List of registers

### Serial I/O1 control register

b7 b6 b5 b4 b3 b2 b1 b0



Serial I/O1 control register (SIO1CON) [Address : 1A<sub>16</sub>]

B	Name	Function	At reset	R	W
0	BRG count source selection bit (CSS)	0 : f(XIN) 1 : f(XIN)/4	0	○	○
1	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "1".		1	○	×
2	Continuous transmit valid bit	0 : Continuous transmit invalid 1 : Continuous transmit valid	0	○	○
3	Transmit interrupt source selection bit (TIC)	0 : Interrupt when transmit buffer has emptied 1 : Interrupt when transmit shift operation is completed	0	○	○
4	Transmit enable bit (TE)	0 : Transmit disabled 1 : Transmit enabled	0	○	○
5	Receive enable bit (RE)	0 : Receive disabled 1 : Receive enabled	0	○	○
6	Serial I/O1 enable bit (SIOE)	b7 b6 0 0 : I/O port 0 1 : Not available 1 0 : UART mode 1 1 : USB mode	0	○	○
7			0	○	○

Fig. 3.5.8 Structure of Serial I/O1 control register

### UART control register

b7 b6 b5 b4 b3 b2 b1 b0



UART control register (UARTCON) [Address : 1B<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Character length selection bit (CHAS)	0 : 8 bits 1 : 7 bits	0	○	○
1	Parity enable bit (PARE)	0 : Parity checking disabled 1 : Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0 : Even parity 1 : Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0 : 1 stop bit 1 : 2 stop bits	0	○	○
4	P11/TxD P-channel output disable bit (POFF)	In output mode 0 : CMOS output 1 : N-channel open-drain output	0	○	○
5	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "1".		1	○	×
6			1	○	×
7			1	○	×

Fig. 3.5.9 Structure of UART control register

### Baud rate generator

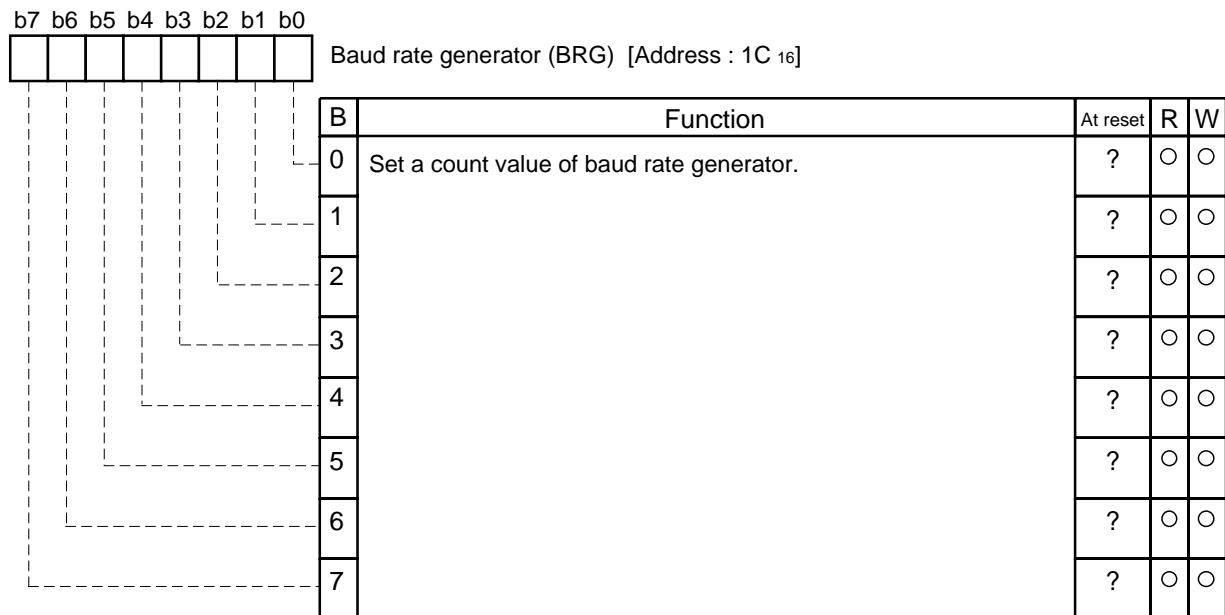


Fig. 3.5.10 Structure of Baud rate generator

Refer to “Figure 2.4.6 Description of register structure” for registers relevant to USB.

### USB data toggle synchronization register (TRSYNC) [Address 1D 16]

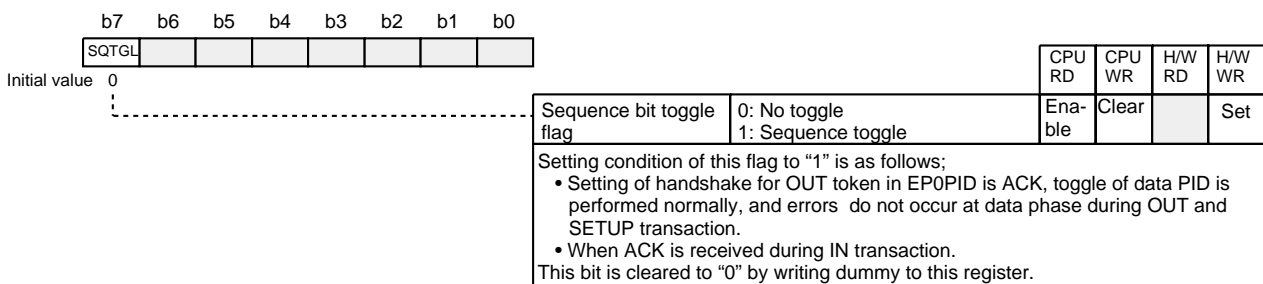


Fig. 3.5.11 Structure of USB data toggle synchronization register

### USB interrupt source discrimination register 1 (USBIR1) [Address 1E 16]

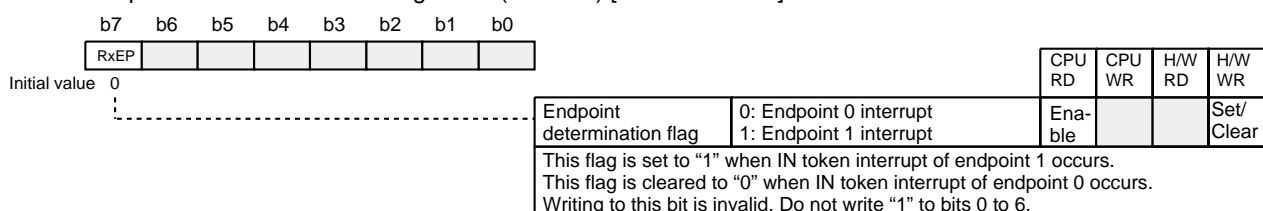


Fig. 3.5.12 Structure of USB interrupt source discrimination register 1



# APPENDIX

## 3.5 List of registers

Refer to “Figure 2.4.6 Description of register structure” for registers relevant to USB.

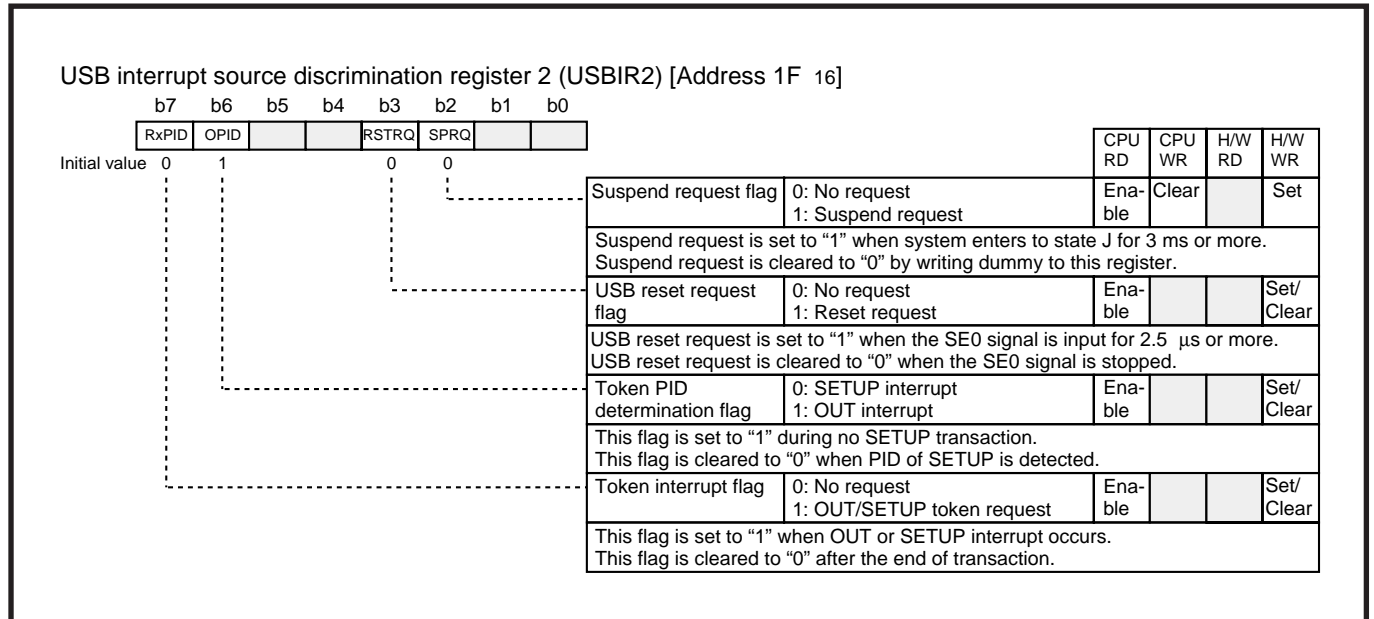


Fig. 3.5.13 Structure of USB interrupt source discrimination register 2

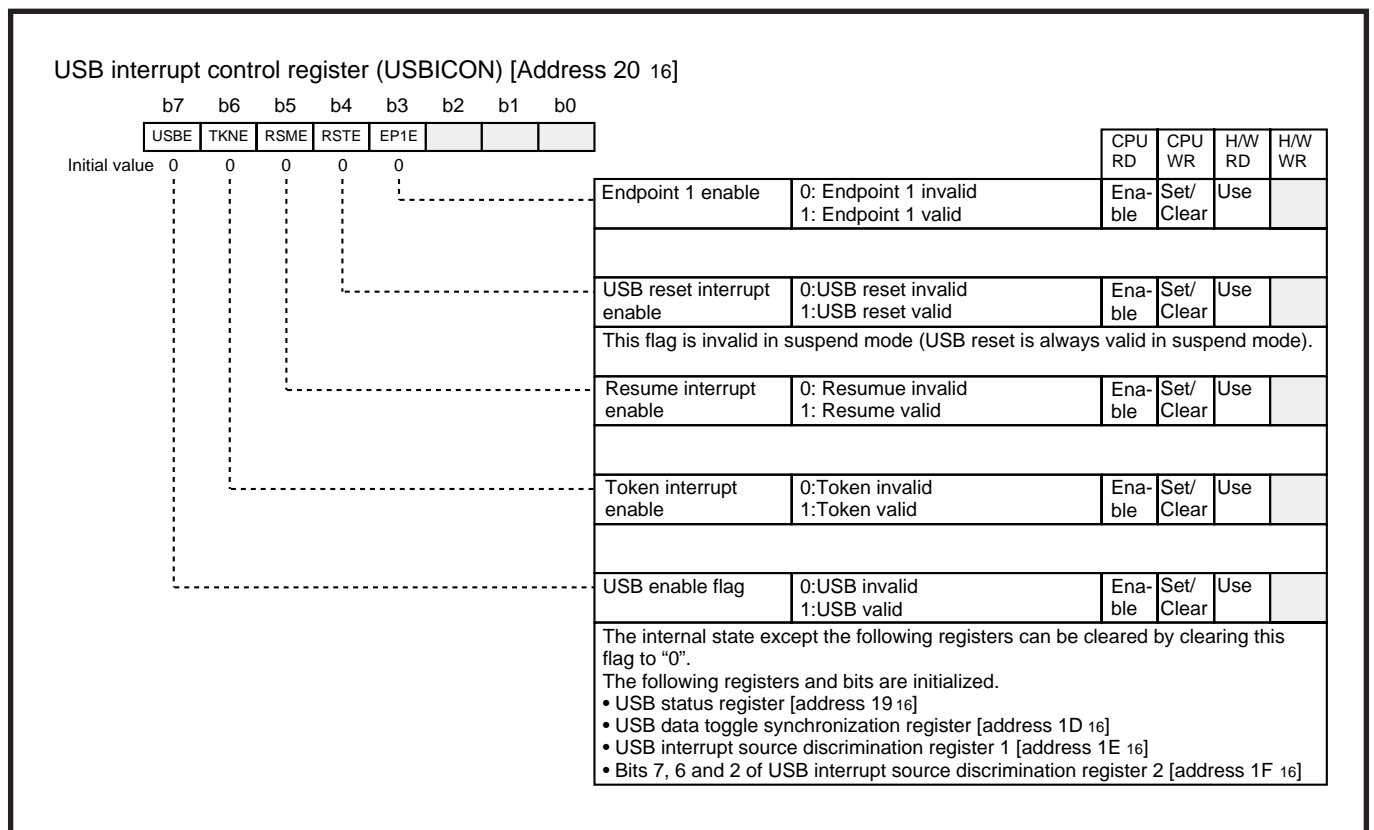


Fig. 3.5.14 Structure of USB interrupt control register

Refer to “Figure 2.4.6 Description of register structure” for registers relevant to USB.

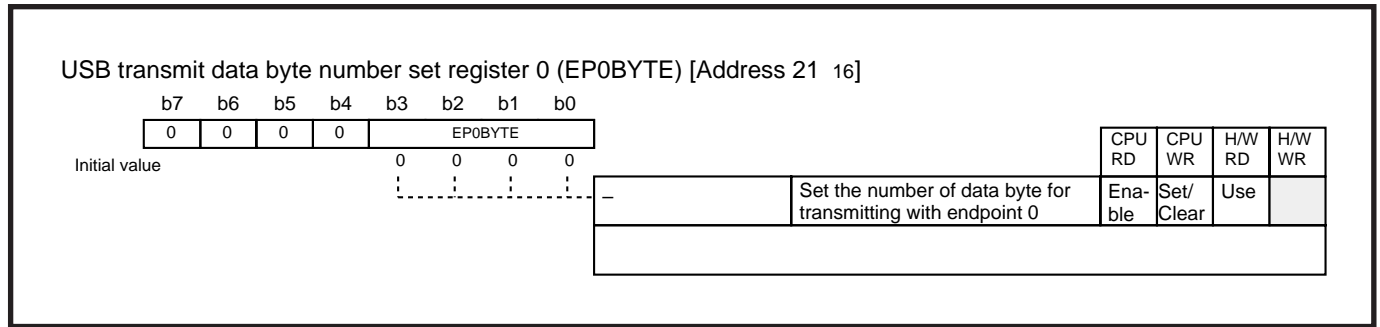


Fig. 3.5.15 Structure of USB transmit data byte number set register 0

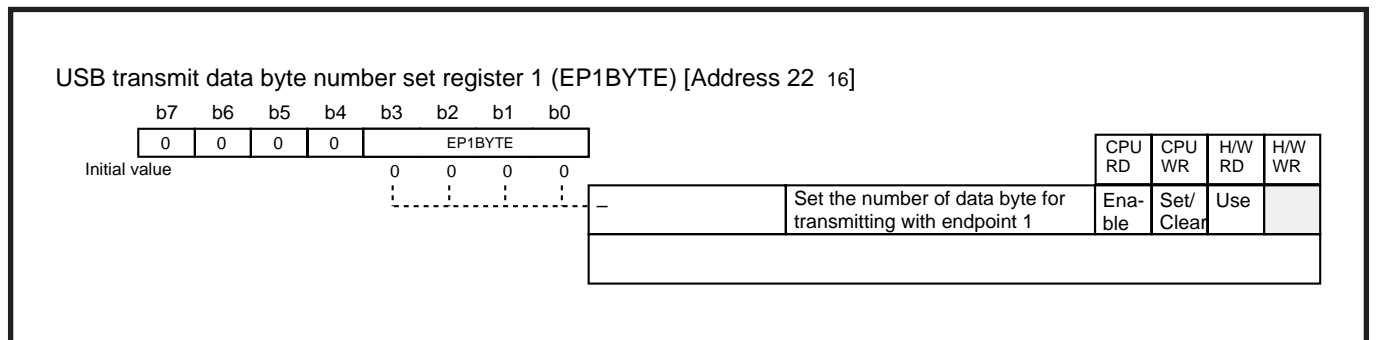


Fig. 3.5.16 Structure of USB transmit data byte number set register 1

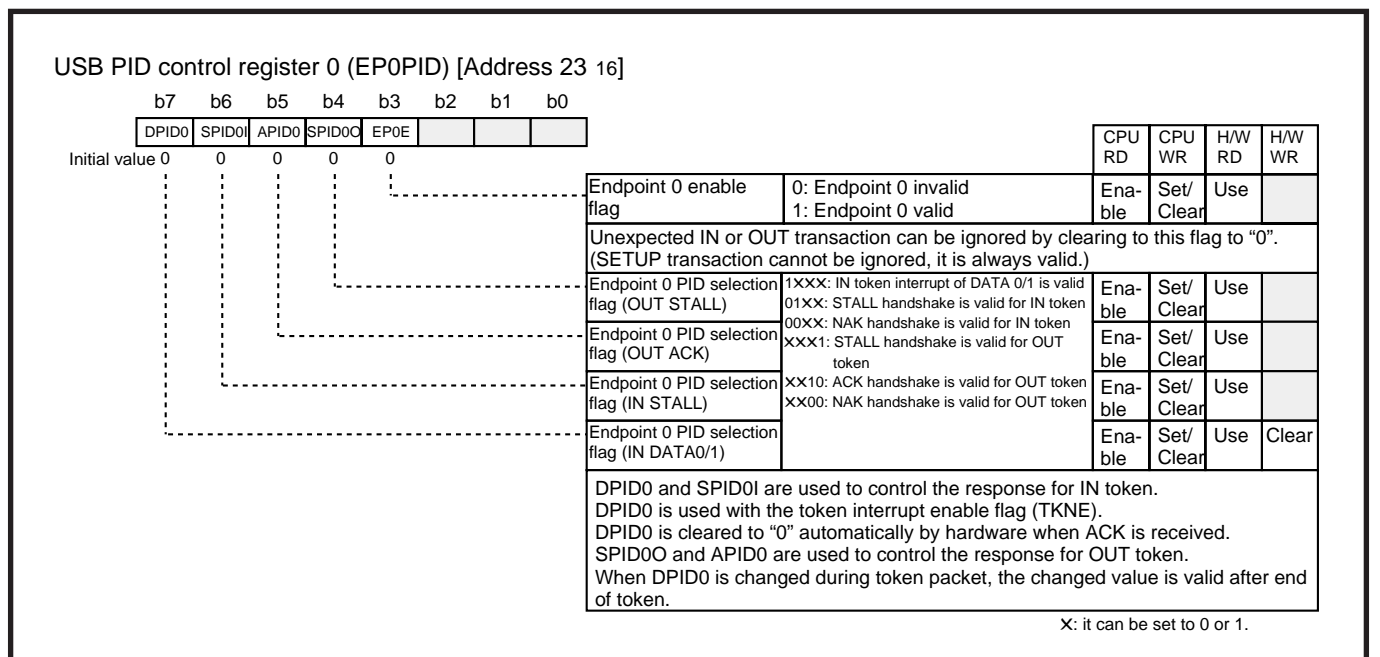


Fig. 3.5.17 Structure of USB PID control register 0

# APPENDIX

## 3.5 List of registers

Refer to “Figure 2.4.6 Description of register structure” for registers relevant to USB.

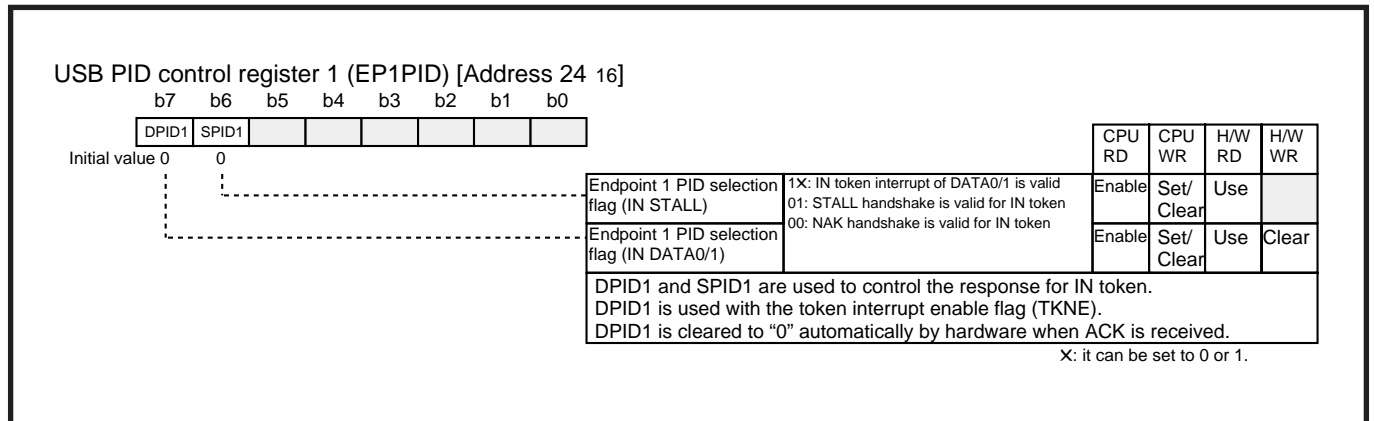


Fig. 3.5.18 Structure of USB PID control register 1

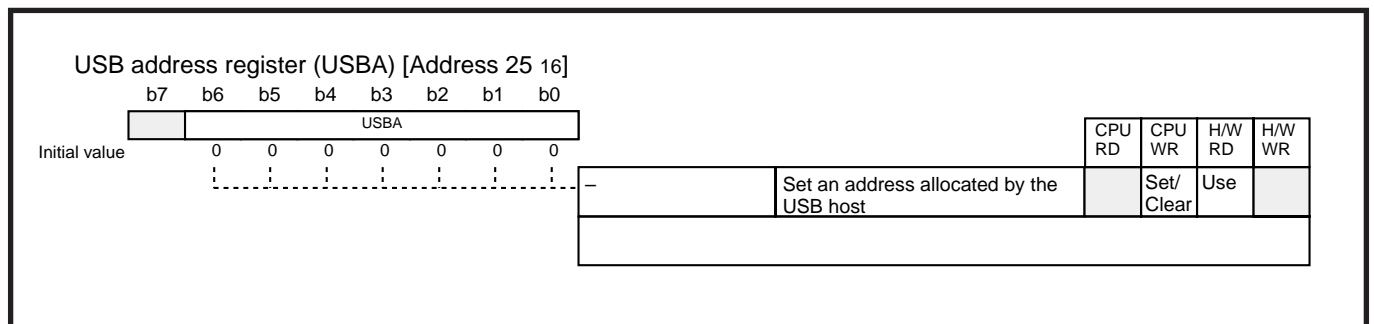


Fig. 3.5.19 Structure of USB address register

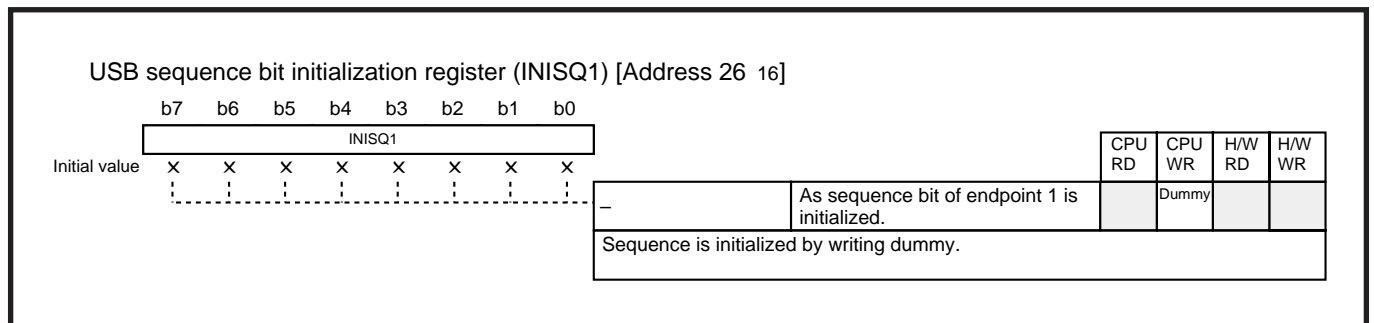


Fig. 3.5.20 Structure of USB sequence bit initialization register

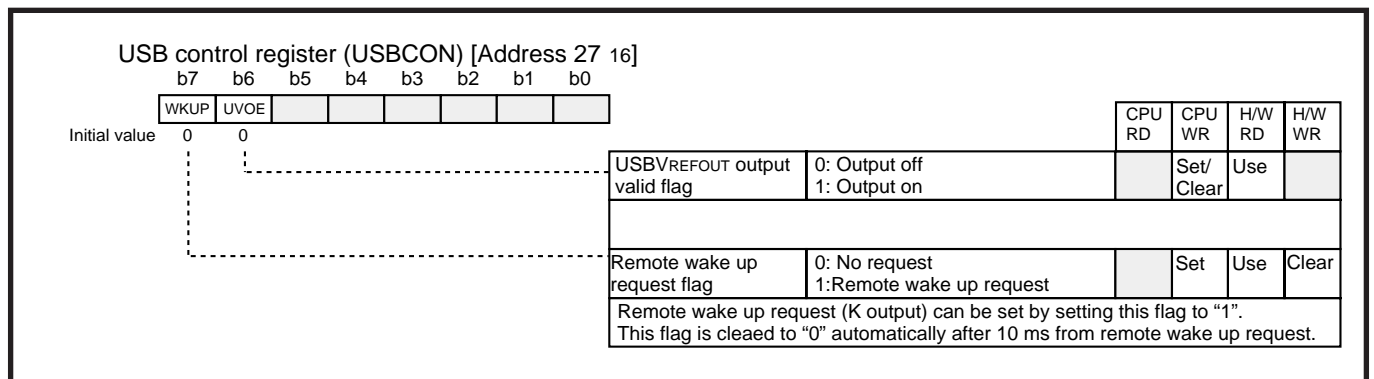
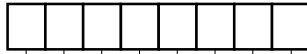


Fig. 3.5.21 Structure of USB control register

### Prescaler 12, Prescaler X

b7 b6 b5 b4 b3 b2 b1 b0



Prescaler 12 (PRE12) [Address : 28<sub>16</sub>]

Prescaler X (PREX) [Address : 2C<sub>16</sub>]

B	Function	At reset	R	W
0	<ul style="list-style-type: none"> <li>Set a count value of each prescaler.</li> <li>The value set in this register is written to both each prescaler and the corresponding prescaler latch at the same time.</li> <li>When this register is read out, the count value of the corresponding prescaler is read out.</li> </ul>	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Fig. 3.5.22 Structure of Prescaler 12, Prescaler X

### Timer 1

b7 b6 b5 b4 b3 b2 b1 b0



Timer 1 (T1) [Address : 29<sub>16</sub>]

B	Function	At reset	R	W
0	<ul style="list-style-type: none"> <li>Set a count value of timer 1.</li> <li>The value set in this register is written to both timer 1 and timer 1 latch at the same time.</li> <li>When this register is read out, the timer 1's count value is read out.</li> </ul>	1	○	○
1		0	○	○
2		0	○	○
3		0	○	○
4		0	○	○
5		0	○	○
6		0	○	○
7		0	○	○

Fig. 3.5.23 Structure of Timer 1

# APPENDIX

## 3.5 List of registers

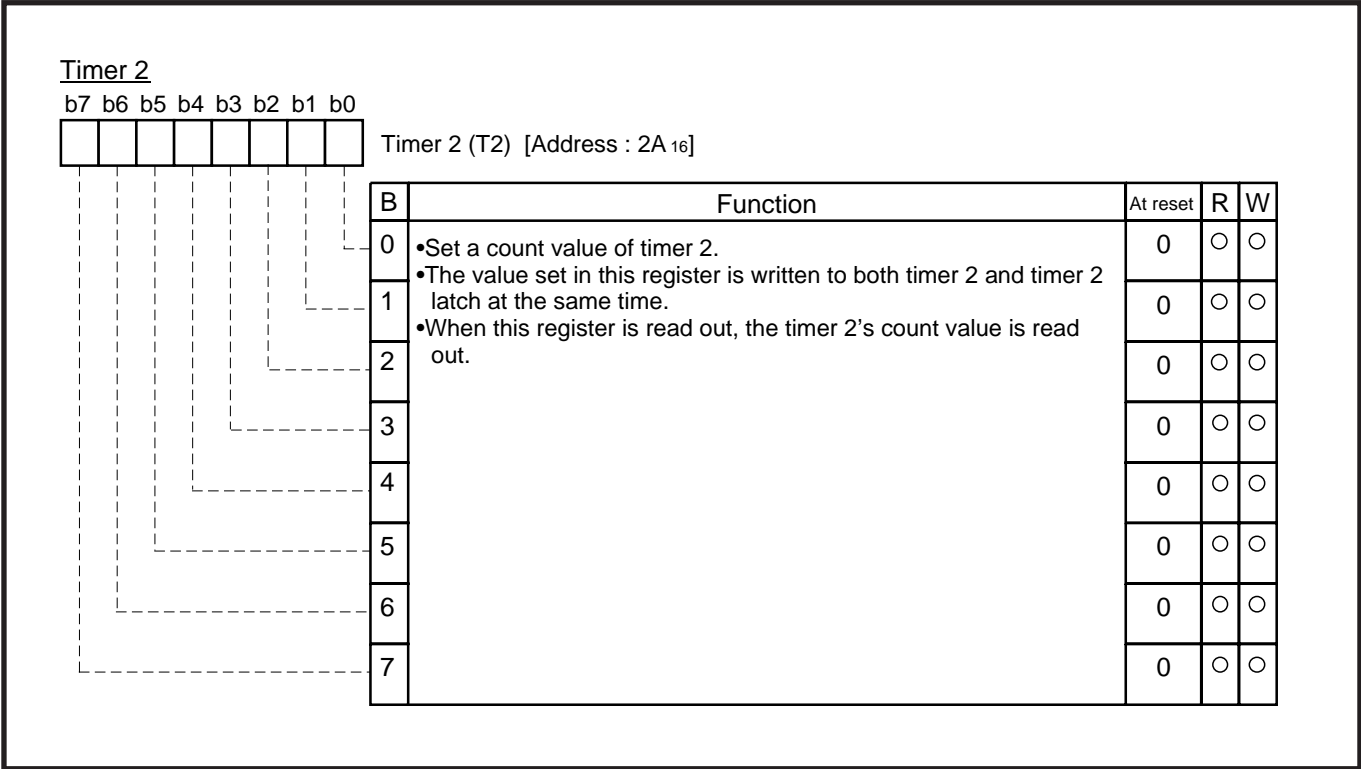


Fig. 3.5.24 Structure of Timer 2

Timer X mode register

b7 b6 b5 b4 b3 b2 b1 b0



Timer X mode register (TM) [Address : 2B<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
1			0	○	○
2	CNTR <sub>0</sub> active edge switch bit	The function depends on the operating mode. (Refer to Table 3.5.1)	0	○	○
3	Timer X count stop bit	0 : Count start 1 : Count stop	0	○	○
4	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.25 Structure of Timer X mode register

Table 3.5.1 CNTR<sub>0</sub> active edge switch bit function

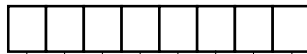
Timer X operation modes	CNTR <sub>0</sub> active edge switch bit (bit 2 of address 2B <sub>16</sub> ) contents	
Timer mode	"0"	CNTR <sub>0</sub> interrupt request occurrence: Falling edge ; No influence to timer count
	"1"	CNTR <sub>0</sub> interrupt request occurrence: Rising edge ; No influence to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

# APPENDIX

## 3.5 List of registers

### Timer X

b7 b6 b5 b4 b3 b2 b1 b0



Timer X (TX) [Address : 2D<sub>16</sub>]

B	Function	At reset	R	W
0	<ul style="list-style-type: none"> <li>Set a count value of timer X.</li> <li>The value set in this register is written to both timer X and timer X latch at the same time.</li> <li>When this register is read out, the timer X's count value is read out.</li> </ul>	1	○	○
1		1	○	○
2		1	○	○
3		1	○	○
4		1	○	○
5		1	○	○
6		1	○	○
7		1	○	○

Fig. 3.5.26 Structure of Timer X

### Timer count source set register

b7 b6 b5 b4 b3 b2 b1 b0



Timer count source set register (TCSS) [Address : 2E<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Timer X count source selection bit ( <b>Note</b> )	0 : $f(X_{IN}) / 16$ 1 : $f(X_{IN}) / 2$	0	○	○
1	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
2			0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

**Note:** To switch the timer X count source selection bit, stop the timer X count operation before do that.

Fig. 3.5.27 Structure of Timer count source set register

### Serial I/O2 control register

b7 b6 b5 b4 b3 b2 b1 b0



Serial I/O2 control register (SIO2CON) [Address : 30<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Internal synchronous clock selection bits	b2 b1 b0 0 0 0 : $f(X_{IN})/8$	0	○	○
1		0 0 1 : $f(X_{IN})/16$	0	○	○
2		0 1 0 : $f(X_{IN})/32$	0	○	○
		0 1 1 : $f(X_{IN})/64$			
		1 1 0 : $f(X_{IN})/128$	0	○	○
		1 1 1 : $f(X_{IN})/256$			
3	S <sub>DATA</sub> pin selection bit (Note)	0 : I/O port / S <sub>DATA</sub> input 1 : S <sub>DATA</sub> output	0	○	○
4	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×
5	Transfer direction selection bit	0 : LSB first 1 : MSB first	0	○	○
6	S <sub>CLK</sub> pin selection bit	0 : External clock (S <sub>CLK</sub> is input) 1 : Internal clock (S <sub>CLK</sub> is output)	0	○	○
7	Transmit / receive shift completion flag	0 : shift in progress 1 : shift completed	0	○	×

**Note:** When using it as a S<sub>DATA</sub> input, set the port P1<sub>3</sub> direction register bit to "0".

Fig. 3.5.28 Structure of Serial I/O2 control register

### Serial I/O2 register

b7 b6 b5 b4 b3 b2 b1 b0



Serial I/O2 register (SIO2) [Address : 31<sub>16</sub>]

B	Function	At reset	R	W
0	A shift register for serial transmission and reception. • At transmitting : Set a transmission data. • At receiving : A reception data is stored.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 3.5.29 Structure of Serial I/O2 register



# APPENDIX

## 3.5 List of registers

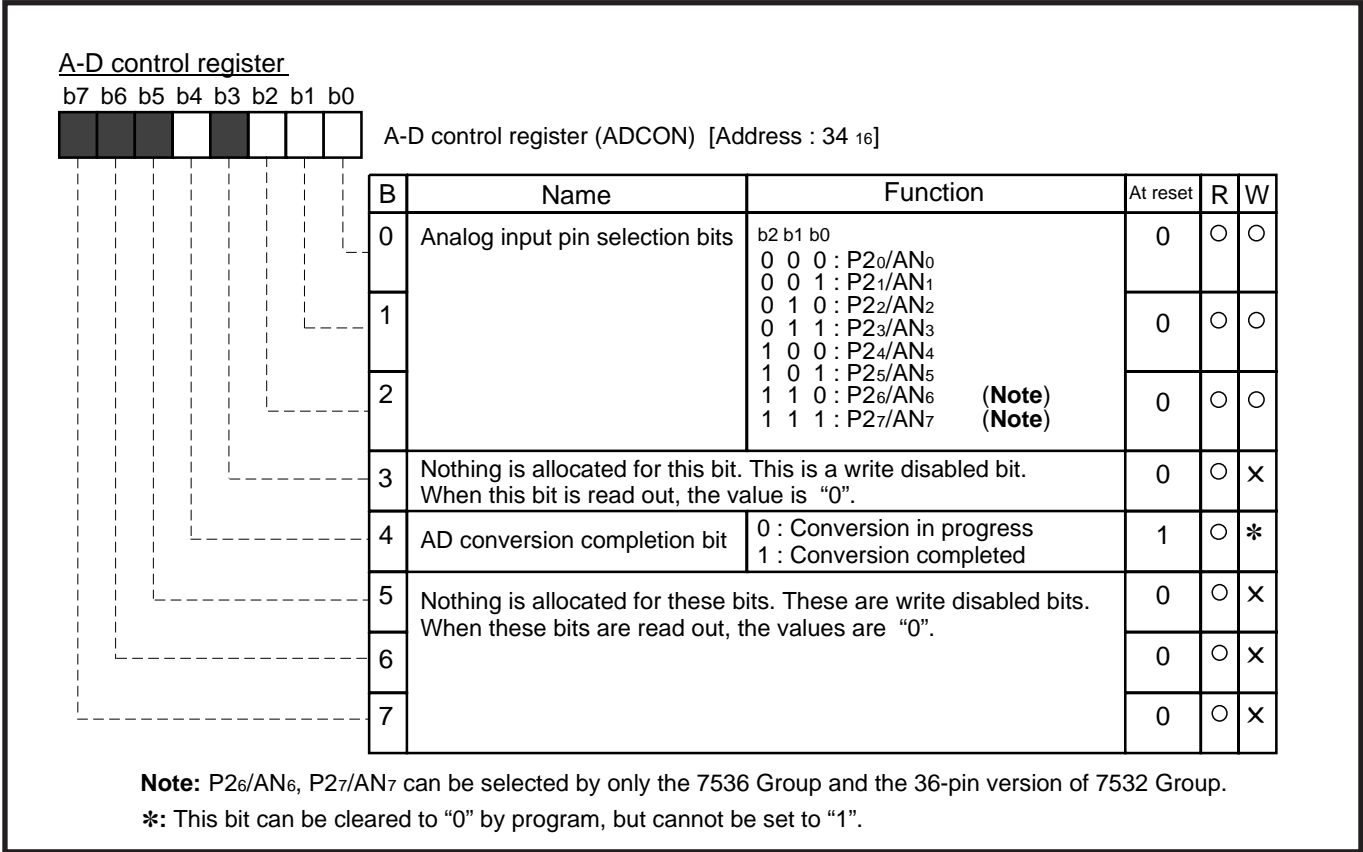
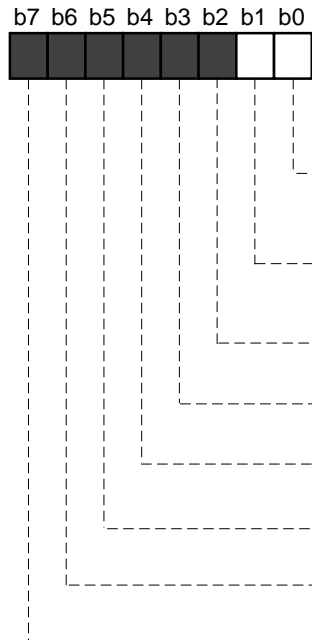


Fig. 3.5.30 Structure of A-D control register

### A-D conversion register (high-order)

b7 b6 b5 b4 b3 b2 b1 b0  

A-D conversion register (high-order) (ADH) [Address : 36<sub>16</sub>]

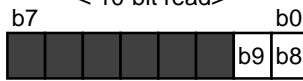
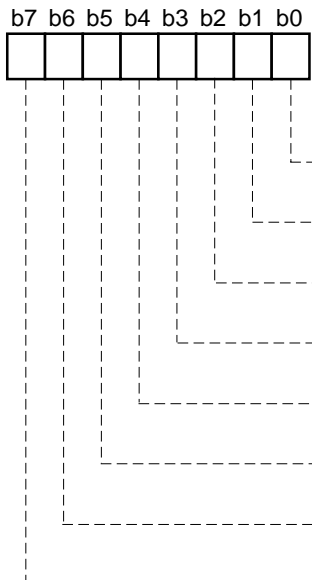
B	Function	At reset	R	W
0	The read-only register in which the A-D conversion's results are stored.	?	○	×
1	<div style="text-align: center;">                     &lt; 10-bit read &gt;                      b7                      b0   </div>	?	○	×
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".	?	○	×
3		?	○	×
4		?	○	×
5		?	○	×
6		?	○	×
7		?	○	×

Fig. 3.5.31 Structure of A-D conversion register (high-order)

### A-D conversion register (low-order)

b7 b6 b5 b4 b3 b2 b1 b0  

A-D conversion register (low-order) (ADL) [Address : 35<sub>16</sub>]

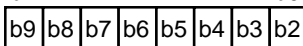
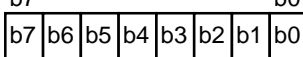
B	Function	At reset	R	W
0	The read-only register in which the A-D conversion's results are stored.	?	○	×
1		?	○	×
2	<div style="text-align: center;">                     &lt; 8-bit read &gt;                      b7                      b0   </div>	?	○	×
3		?	○	×
4	<div style="text-align: center;">                     &lt; 10-bit read &gt;                      b7                      b0   </div>	?	○	×
5		?	○	×
6		?	○	×
7		?	○	×

Fig. 3.5.32 Structure of A-D conversion register (low-order)

# APPENDIX

## 3.5 List of registers

MISRG

b7 b6 b5 b4 b3 b2 b1 b0



MISRG [Address : 38<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Oscillation stabilization time set bit after release of the STP instruction	0 : Set "01 <sub>16</sub> " in timer 1, and "FF <sub>16</sub> " in prescaler 12 automatically 1 : Not set automatically	0	○	○
1	These are reserved bits . Do not write "1" to these bits.		0	○	×
2			0	○	×
3			0	○	×
4	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.33 Structure of MISRG

Watchdog timer control register

b7 b6 b5 b4 b3 b2 b1 b0



Watchdog timer control register (WDTCON) [Address : 39<sub>16</sub>]

B	Name	Function	At reset	R	W
0	Watchdog timer H (The high-order 6 bits are read-only bits.)		1	○	×
1			1	○	×
2			1	○	×
3			1	○	×
4			1	○	×
5			1	○	×
6	STP instruction disable bit	0 : STP instruction enabled 1 : STP instruction disabled	0	○	○
7	Watchdog timer H count source selection bit	0 : Watchdog timer L underflow 1 : f(X <sub>IN</sub> )/16	0	○	○

Fig. 3.5.34 Structure of Watchdog timer control register

### Interrupt edge selection register

b7 b6 b5 b4 b3 b2 b1 b0



Interrupt edge selection register (INTEDGE) [Address : 3A<sub>16</sub>]

B	Name	Function	At reset	R	W
0	INT <sub>0</sub> interrupt edge selection bit ( <b>Note 1</b> )	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT <sub>1</sub> interrupt edge selection bit ( <b>Note 2</b> )	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
3			0	○	×
4	Serial I/O1 or INT <sub>1</sub> interrupt selection bit	0 : Serial I/O1 1 : INT <sub>1</sub>	0	○	○
5	Timer X or key-on wake up interrupt selection bit	0 : Timer X 1 : Key-on wake up	0	○	○
6	Timer 2 or serial I/O2 interrupt selection bit	0 : Timer 2 1 : Serial I/O2	0	○	○
7	CNTR <sub>0</sub> or AD converter interrupt selection bit	0 : CNTR <sub>0</sub> 1 : AD converter	0	○	○

**Notes 1:** 7532 Group, 32-pin version: Not used.

This is a write disabled bit. When this bit is read out, the value is "0".

**2:** 7532 Group, 36-pin and 32-pin version: Not used.

This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 3.5.35 Structure of Interrupt edge selection register

### CPU mode register

b7 b6 b5 b4 b3 b2 b1 b0



CPU mode register (CPUM) [Address : 3B<sub>16</sub>]

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	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0". (Do not write "1".)		0	○	×
4			0	○	×
5			0	○	×
6	Clock division ratio selection bits	b7 b6 0 0 : $\phi = f(X_{IN})/2$ (high-speedmode) 0 1 : $\phi = f(X_{IN})/8$ (middle-speed mode) 1 0 : Applied from ring oscillator 1 1 : $\phi = f(X_{IN})$ (double-speed mode)	0	○	○
7			1	○	○

Fig. 3.5.36 Structure of CPU mode register

# APPENDIX

## 3.5 List of registers

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0



Interrupt request register 1 (IREQ1) [Address : 3C<sub>16</sub>]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
1	UART transmit/USBSETUP/OUT token/Reset/Suspend/Resume/INT <sub>1</sub> interrupt request bit ( <b>Note 1</b> )	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
2	INT <sub>0</sub> interrupt request bit ( <b>Note 2</b> )	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
3	Timer X or key-on wake up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
4	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
5	Timer 2 or serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
6	CNTR <sub>0</sub> or AD converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	*
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	X

\*: These bits can be cleared to "0" by program, but cannot be set.

**Notes 1:** 7532 Group, 36-pin version and 32-pin version: INT<sub>1</sub> interrupt does not exist.

**2:** 7532 Group, 32-pin version: INT<sub>0</sub> interrupt does not exist.

This is a write disabled bit. When this bit is read out, the value is "0".

**Fig. 3.5.37 Structure of Interrupt request register 1**

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0



Interrupt control register 1 (ICON1) [Address : 3E<sub>16</sub>]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	UART transmit/USBSETUP/OUT token/Reset/Suspend/Resume/INT <sub>1</sub> interrupt enable bit ( <b>Note 1</b> )	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT <sub>0</sub> interrupt enable bit ( <b>Note 2</b> )	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Timer X or key-on wake up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer 2 or serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	CNTR <sub>0</sub> or AD converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Nothing is allocated for this bit. Do not write "1" to this bit. When this bit is read out, the value is "0".		0	○	X

**Notes 1:** 7532 Group, 36-pin version and 32-pin version: INT<sub>1</sub> interrupt does not exist.

**2:** 7532 Group, 32-pin version: INT<sub>0</sub> interrupt does not exist.

This is a write disabled bit. When this bit is read out, the value is "0".

**Fig. 3.5.38 Structure of Interrupt control register 1**

## 3.6 Mask ROM confirmation form

GZZ-SH51-96B&lt;7ZB1&gt;

Mask ROM number

**740 FAMILY MASK ROM CONFIRMATION FORM**  
**SINGLE-CHIP MICROCOMPUTER**  
**M37532M4-XXXFP**  
**MITSUBISHI ELECTRIC**

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked \*.

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

## \* 1. Confirmation

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

Three EPROMs are required for each pattern (Check @ in the appropriate box).

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

Checksum code for entire EPROM

--	--	--	--

(hexadecimal notation)

EPROM type (indicate the type used)

<input type="checkbox"/> <b>27C256</b>	<input type="checkbox"/> <b>27C512</b>
<b>EPROM address</b> 0000 <sub>16</sub> Area for ASCII codes of the name of the product 'M37532M4-' 000F <sub>16</sub> 0010 <sub>16</sub> 607F <sub>16</sub> 6080 <sub>16</sub> 7FFD <sub>16</sub> Data ROM (8K-130) bytes 7FFE <sub>16</sub> 7FFF <sub>16</sub> 	<b>EPROM address</b> 0000 <sub>16</sub> Area for ASCII codes of the name of the product 'M37532M4-' 000F <sub>16</sub> 0010 <sub>16</sub> E07F <sub>16</sub> E080 <sub>16</sub> FFFD <sub>16</sub> Data ROM (8K-130) bytes FFFE <sub>16</sub> FFFF <sub>16</sub> 

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

(1) Set "FF<sub>16</sub>" in the shaded area.

(2) Write the ASCII codes that indicates the name of the product 'M37532M4-' to addresses 0000<sub>16</sub> to 000F<sub>16</sub>. ASCII codes 'M37532M4-' are listed on the right. The addresses and data are in hexadecimal notation.

Address

0000 <sub>16</sub>	'M' = 4D <sub>16</sub>
0001 <sub>16</sub>	'3' = 33 <sub>16</sub>
0002 <sub>16</sub>	'7' = 37 <sub>16</sub>
0003 <sub>16</sub>	'5' = 35 <sub>16</sub>
0004 <sub>16</sub>	'3' = 33 <sub>16</sub>
0005 <sub>16</sub>	'2' = 32 <sub>16</sub>
0006 <sub>16</sub>	'M' = 4D <sub>16</sub>
0007 <sub>16</sub>	'4' = 34 <sub>16</sub>

Address

0008 <sub>16</sub>	'-' = 2D <sub>16</sub>
0009 <sub>16</sub>	FF <sub>16</sub>
000A <sub>16</sub>	FF <sub>16</sub>
000B <sub>16</sub>	FF <sub>16</sub>
000C <sub>16</sub>	FF <sub>16</sub>
000D <sub>16</sub>	FF <sub>16</sub>
000E <sub>16</sub>	FF <sub>16</sub>
000F <sub>16</sub>	FF <sub>16</sub>

# APPENDIX

## 3.6 Mask ROM confirmation form

GZZ-SH51-96B<7ZB1>

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

### 740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M37532M4-XXXFP MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the assembler source file :

EPROM type	27C256	27C512
The pseudo-command	$\ast=\triangle \$8000$ .BYTE $\triangle$ 'M37532M4-'	$\ast=\triangle \$0000$ .BYTE $\triangle$ 'M37532M4-'

ASCII codes, that indicates the name of the product, are written in addresses 0000<sub>16</sub> to 000F<sub>16</sub> of the EPROM by programming the above pseudo-command, which depends on a type of EPROM to be written, at beginning of the source program.

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

#### ✱ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (36P2R-A for M37532M4-XXXFP) and attach to the mask ROM confirmation form.

#### ✱ 3. Usage conditions

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

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

- ☐ Ceramic resonator  
☐ External clock input      ☐ Other (            )

At what frequency?

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

#### ✱ 4. Comments

GZZ-SH54-47B&lt;8YA0&gt;

Mask ROM number

**740 FAMILY MASK ROM CONFIRMATION FORM**  
**SINGLE-CHIP MICROCOMPUTER**  
**M37532M4-XXXGP**  
**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 (Check @ in the appropriate box).

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

Checksum code for entire EPROM

--	--	--	--

(hexadecimal notation)

EPROM type (indicate the type used)

<input type="checkbox"/> <b>27C256</b>	<input type="checkbox"/> <b>27C512</b>																						
<p>EPROM address</p> <table border="1"> <tr> <td>0000<sub>16</sub></td> <td rowspan="3">Area for ASCII codes of the name of the product 'M37532M4-'</td> </tr> <tr> <td>000F<sub>16</sub></td> </tr> <tr> <td>0010<sub>16</sub></td> </tr> <tr> <td>607F<sub>16</sub></td> <td rowspan="2">Data ROM (8K-130) bytes</td> </tr> <tr> <td>6080<sub>16</sub></td> </tr> <tr> <td>7FFD<sub>16</sub></td> <td rowspan="3">Data ROM (8K-130) bytes</td> </tr> <tr> <td>7FFE<sub>16</sub></td> </tr> <tr> <td>7FFF<sub>16</sub></td> </tr> </table>	0000 <sub>16</sub>	Area for ASCII codes of the name of the product 'M37532M4-'	000F <sub>16</sub>	0010 <sub>16</sub>	607F <sub>16</sub>	Data ROM (8K-130) bytes	6080 <sub>16</sub>	7FFD <sub>16</sub>	Data ROM (8K-130) bytes	7FFE <sub>16</sub>	7FFF <sub>16</sub>	<p>EPROM address</p> <table border="1"> <tr> <td>0000<sub>16</sub></td> <td rowspan="3">Area for ASCII codes of the name of the product 'M37532M4-'</td> </tr> <tr> <td>000F<sub>16</sub></td> </tr> <tr> <td>0010<sub>16</sub></td> </tr> <tr> <td>E07F<sub>16</sub></td> <td rowspan="2">Data ROM (8K-130) bytes</td> </tr> <tr> <td>E080<sub>16</sub></td> </tr> <tr> <td>FFFD<sub>16</sub></td> <td rowspan="3">Data ROM (8K-130) bytes</td> </tr> <tr> <td>FFFE<sub>16</sub></td> </tr> <tr> <td>FFFF<sub>16</sub></td> </tr> </table>	0000 <sub>16</sub>	Area for ASCII codes of the name of the product 'M37532M4-'	000F <sub>16</sub>	0010 <sub>16</sub>	E07F <sub>16</sub>	Data ROM (8K-130) bytes	E080 <sub>16</sub>	FFFD <sub>16</sub>	Data ROM (8K-130) bytes	FFFE <sub>16</sub>	FFFF <sub>16</sub>
0000 <sub>16</sub>	Area for ASCII codes of the name of the product 'M37532M4-'																						
000F <sub>16</sub>																							
0010 <sub>16</sub>																							
607F <sub>16</sub>	Data ROM (8K-130) bytes																						
6080 <sub>16</sub>																							
7FFD <sub>16</sub>	Data ROM (8K-130) bytes																						
7FFE <sub>16</sub>																							
7FFF <sub>16</sub>																							
0000 <sub>16</sub>	Area for ASCII codes of the name of the product 'M37532M4-'																						
000F <sub>16</sub>																							
0010 <sub>16</sub>																							
E07F <sub>16</sub>	Data ROM (8K-130) bytes																						
E080 <sub>16</sub>																							
FFFD <sub>16</sub>	Data ROM (8K-130) bytes																						
FFFE <sub>16</sub>																							
FFFF <sub>16</sub>																							

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

(1) Set "FF<sub>16</sub>" in the shaded area.

(2) Write the ASCII codes that indicates the name of the product 'M37532M4-' to addresses 0000<sub>16</sub> to 000F<sub>16</sub>. ASCII codes 'M37532M4-' are listed on the right. The addresses and data are in hexadecimal notation.

Address

0000 <sub>16</sub>	'M' = 4D <sub>16</sub>
0001 <sub>16</sub>	'3' = 33 <sub>16</sub>
0002 <sub>16</sub>	'7' = 37 <sub>16</sub>
0003 <sub>16</sub>	'5' = 35 <sub>16</sub>
0004 <sub>16</sub>	'3' = 33 <sub>16</sub>
0005 <sub>16</sub>	'2' = 32 <sub>16</sub>
0006 <sub>16</sub>	'M' = 4D <sub>16</sub>
0007 <sub>16</sub>	'4' = 34 <sub>16</sub>

Address

0008 <sub>16</sub>	'-' = 2D <sub>16</sub>
0009 <sub>16</sub>	FF <sub>16</sub>
000A <sub>16</sub>	FF <sub>16</sub>
000B <sub>16</sub>	FF <sub>16</sub>
000C <sub>16</sub>	FF <sub>16</sub>
000D <sub>16</sub>	FF <sub>16</sub>
000E <sub>16</sub>	FF <sub>16</sub>
000F <sub>16</sub>	FF <sub>16</sub>



# APPENDIX

## 3.6 Mask ROM confirmation form

GZZ-SH54-47B<8YA0>

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

### 740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M37532M4-XXXGP MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the assembler source file :

EPROM type	27C256	27C512
The pseudo-command	*=△\$8000 .BYTE △'M37532M4-'	*=△\$0000 .BYTE △'M37532M4-'

ASCII codes, that indicates the name of the product, are written in addresses 0000<sub>16</sub> to 000F<sub>16</sub> of the EPROM by programming the above pseudo-command, which depends on a type of EPROM to be written, at beginning of the source program.

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

#### ※ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered fill out the appropriate mark specification form (32P6B-A for M37532M4-XXXGP) and attach to the mask ROM confirmation form.

#### ※ 3. Usage conditions

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

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

☐ Ceramic resonator

☐ External clock input

☐ Other (            )

At what frequency?

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

#### ※ 4. Comments

GZZ-SH51-97B&lt;7ZA1&gt;

Mask ROM number

**740 FAMILY MASK ROM CONFIRMATION FORM**  
**SINGLE-CHIP MICROCOMPUTER**  
**M37536M4-XXXSP**  
**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 (Check @ in the appropriate box).

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

Checksum code for entire EPROM

--	--	--	--

(hexadecimal notation)

EPROM type (indicate the type used)

<input type="checkbox"/> <b>27C256</b>	<input type="checkbox"/> <b>27C512</b>																						
<p>EPROM address</p> <table border="1"> <tr> <td>0000<sub>16</sub></td> <td rowspan="3">Area for ASCII codes of the name of the product 'M37536M4-'</td> </tr> <tr> <td>000F<sub>16</sub></td> </tr> <tr> <td>0010<sub>16</sub></td> </tr> <tr> <td>607F<sub>16</sub></td> <td rowspan="2">Data ROM (8K-130) bytes</td> </tr> <tr> <td>6080<sub>16</sub></td> </tr> <tr> <td>7FFD<sub>16</sub></td> <td rowspan="3">Data ROM (8K-130) bytes</td> </tr> <tr> <td>7FFE<sub>16</sub></td> </tr> <tr> <td>7FFF<sub>16</sub></td> </tr> </table>	0000 <sub>16</sub>	Area for ASCII codes of the name of the product 'M37536M4-'	000F <sub>16</sub>	0010 <sub>16</sub>	607F <sub>16</sub>	Data ROM (8K-130) bytes	6080 <sub>16</sub>	7FFD <sub>16</sub>	Data ROM (8K-130) bytes	7FFE <sub>16</sub>	7FFF <sub>16</sub>	<p>EPROM address</p> <table border="1"> <tr> <td>0000<sub>16</sub></td> <td rowspan="3">Area for ASCII codes of the name of the product 'M37536M4-'</td> </tr> <tr> <td>000F<sub>16</sub></td> </tr> <tr> <td>0010<sub>16</sub></td> </tr> <tr> <td>E07F<sub>16</sub></td> <td rowspan="2">Data ROM (8K-130) bytes</td> </tr> <tr> <td>E080<sub>16</sub></td> </tr> <tr> <td>FFFD<sub>16</sub></td> <td rowspan="3">Data ROM (8K-130) bytes</td> </tr> <tr> <td>FFFE<sub>16</sub></td> </tr> <tr> <td>FFFF<sub>16</sub></td> </tr> </table>	0000 <sub>16</sub>	Area for ASCII codes of the name of the product 'M37536M4-'	000F <sub>16</sub>	0010 <sub>16</sub>	E07F <sub>16</sub>	Data ROM (8K-130) bytes	E080 <sub>16</sub>	FFFD <sub>16</sub>	Data ROM (8K-130) bytes	FFFE <sub>16</sub>	FFFF <sub>16</sub>
0000 <sub>16</sub>	Area for ASCII codes of the name of the product 'M37536M4-'																						
000F <sub>16</sub>																							
0010 <sub>16</sub>																							
607F <sub>16</sub>	Data ROM (8K-130) bytes																						
6080 <sub>16</sub>																							
7FFD <sub>16</sub>	Data ROM (8K-130) bytes																						
7FFE <sub>16</sub>																							
7FFF <sub>16</sub>																							
0000 <sub>16</sub>	Area for ASCII codes of the name of the product 'M37536M4-'																						
000F <sub>16</sub>																							
0010 <sub>16</sub>																							
E07F <sub>16</sub>	Data ROM (8K-130) bytes																						
E080 <sub>16</sub>																							
FFFD <sub>16</sub>	Data ROM (8K-130) bytes																						
FFFE <sub>16</sub>																							
FFFF <sub>16</sub>																							

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

(1) Set "FF<sub>16</sub>" in the shaded area.

(2) Write the ASCII codes that indicates the name of the product 'M37536M4-' to addresses 0000<sub>16</sub> to 000F<sub>16</sub>. ASCII codes 'M37536M4-' are listed on the right. The addresses and data are in hexadecimal notation.

Address

0000 <sub>16</sub>	'M' = 4D <sub>16</sub>
0001 <sub>16</sub>	'3' = 33 <sub>16</sub>
0002 <sub>16</sub>	'7' = 37 <sub>16</sub>
0003 <sub>16</sub>	'5' = 35 <sub>16</sub>
0004 <sub>16</sub>	'3' = 33 <sub>16</sub>
0005 <sub>16</sub>	'6' = 36 <sub>16</sub>
0006 <sub>16</sub>	'M' = 4D <sub>16</sub>
0007 <sub>16</sub>	'4' = 34 <sub>16</sub>

Address

0008 <sub>16</sub>	'-' = 2D <sub>16</sub>
0009 <sub>16</sub>	FF <sub>16</sub>
000A <sub>16</sub>	FF <sub>16</sub>
000B <sub>16</sub>	FF <sub>16</sub>
000C <sub>16</sub>	FF <sub>16</sub>
000D <sub>16</sub>	FF <sub>16</sub>
000E <sub>16</sub>	FF <sub>16</sub>
000F <sub>16</sub>	FF <sub>16</sub>

# APPENDIX

## 3.6 Mask ROM confirmation form

GZZ-SH51-97B<7ZA1>

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

### 740 FAMILY MASK ROM CONFIRMATION FORM

#### SINGLE-CHIP MICROCOMPUTER

#### M37536M4-XXXSP

#### MITSUBISHI ELECTRIC

Recommend to writing the following pseudo-command to the assembler source file :

EPROM type	27C256	27C512
The pseudo-command	*=△\$8000 .BYTE △'M37536M4-'	*=△\$0000 .BYTE △'M37536M4-'

ASCII codes, that indicates the name of the product, are written in addresses 0000<sub>16</sub> to 000F<sub>16</sub> of the EPROM by programming the above pseudo-command, which depends on a type of EPROM to be written, at beginning of the source program.

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation, the ROM processing is disabled. Write the data correctly.

#### ※ 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 M37536M4-XXXSP) and attach to the mask ROM confirmation form.

#### ※ 3. Usage conditions

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

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

☐ Ceramic resonator

☐ External clock input

☐ Other (            )

At what frequency?

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

#### ※ 4. Comments

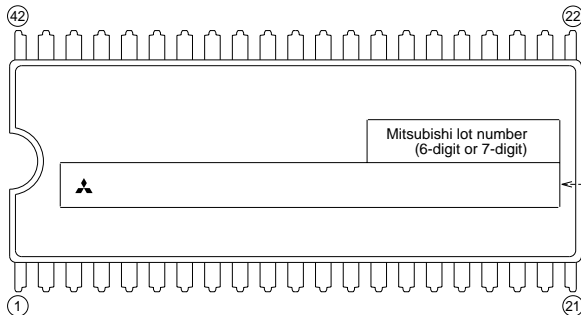
### 3.7 Mark specification form

#### 42P4B (42-PIN SHRINK DIP) MARK SPECIFICATION FORM

Mitsubishi IC catalog name

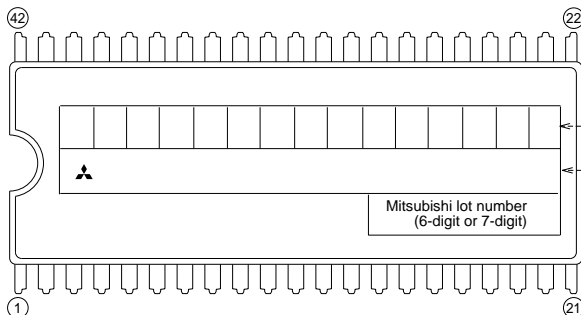
Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

##### A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

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



Customer's Parts Number

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

Mitsubishi IC catalog name

Note1 : The mark field should be written right aligned.

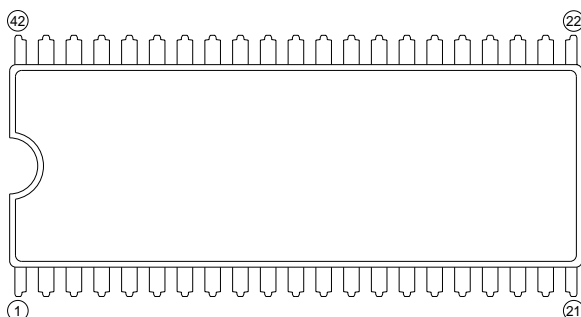
2 : The fonts and size of characters are standard Mitsubishi type.

3 : Customer's Parts Number can be up to 15 characters : Only 0 ~ 9, A ~ Z, +, -, /, (, ), &, @, . (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.

# APPENDIX

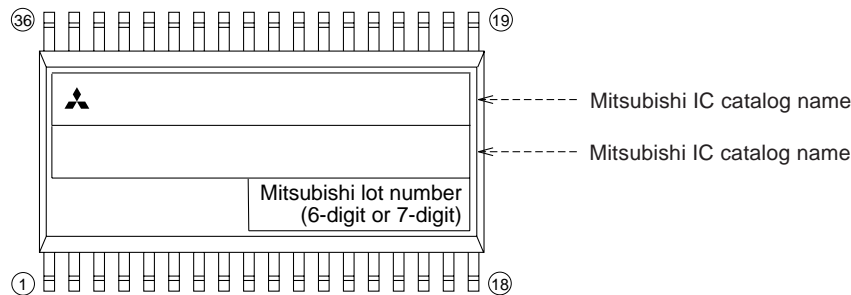
## 3.7 Mark specification form

### 36P2R-A (36-PIN SHRINK SOP) MARK SPECIFICATION FORM

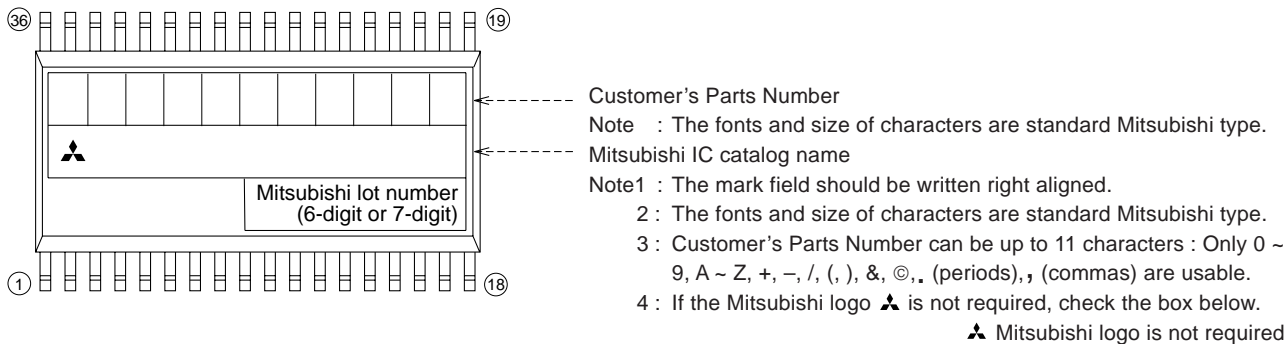
Mitsubishi IC catalog name

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

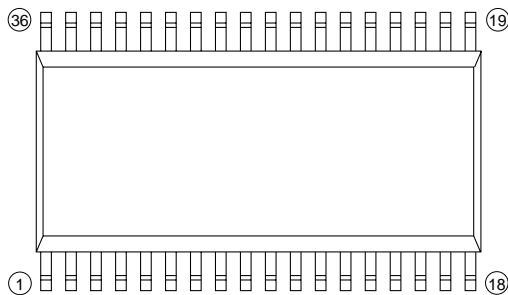


B. Customer's Parts Number + Mitsubishi catalog name



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

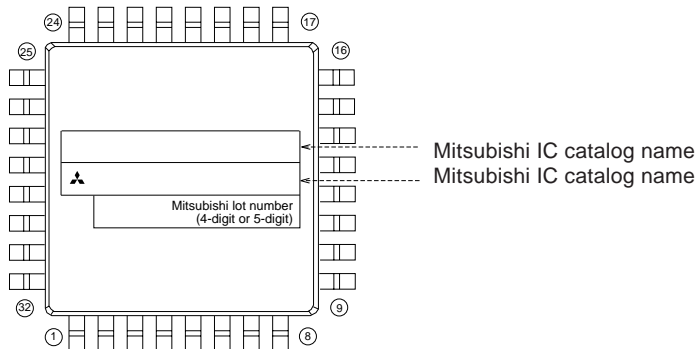
3 : The standard Mitsubishi font is used for all characters except for a logo.

### 32P6B-A (32-PIN LQFP) MARK SPECIFICATION FORM

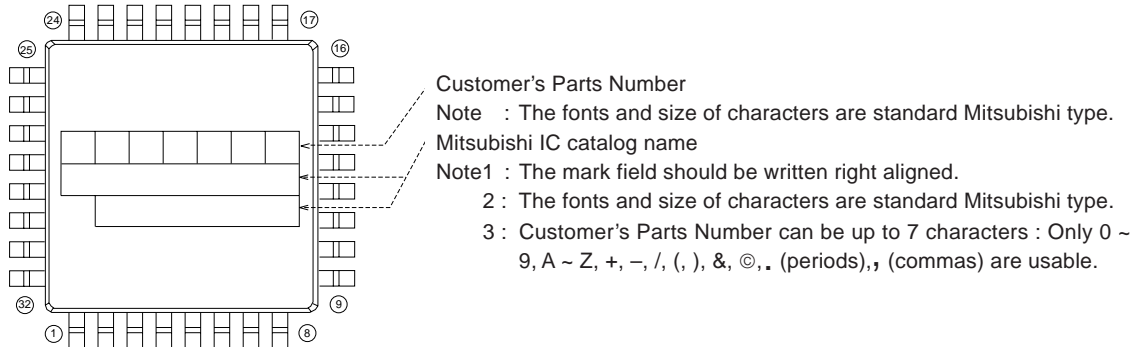
Mitsubishi IC catalog name

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

#### A. Standard Mitsubishi Mark



#### B. Customer's Parts Number + Mitsubishi catalog name



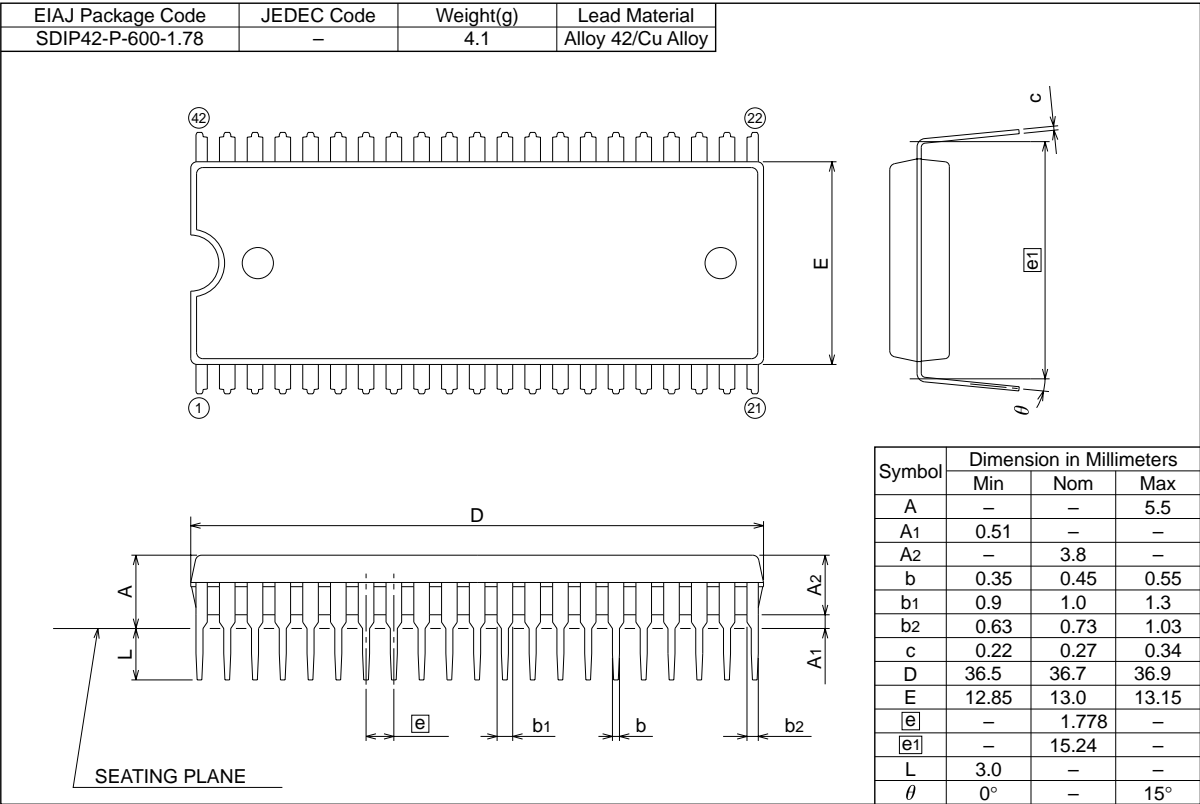
# APPENDIX

## 3.8 Package outline

### 3.8 Package outline

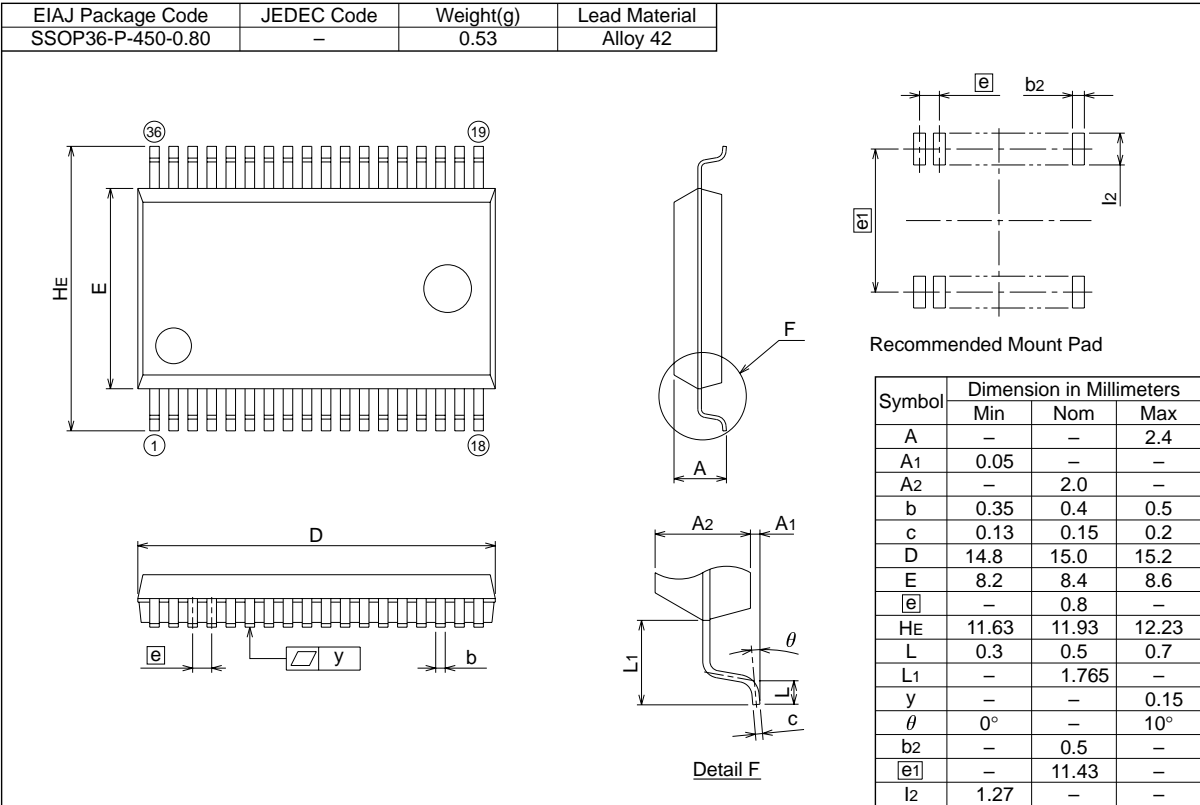
#### 42P4B

Plastic 42pin 600mil SDIP



#### 36P2R-A

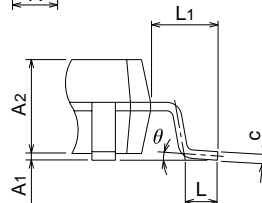
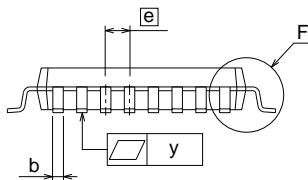
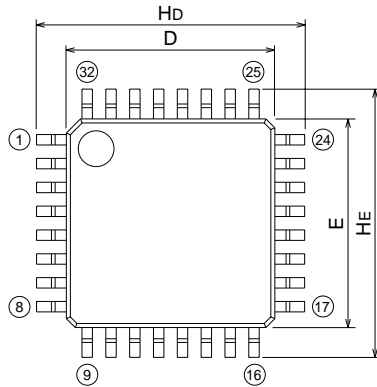
Plastic 36pin 450mil SSOP



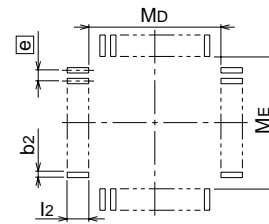
### 32P6B-A

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP32-P-77-0.80	—		Alloy 42

### Plastic 32pin 7X7mm body LQFP



Detail F

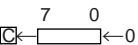


Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	1.7
A1	0	0.1	0.2
A2	—	1.4	—
b	0.3	0.35	0.45
c	0.105	0.125	0.175
D	6.9	7.0	7.1
E	6.9	7.0	7.1
e	—	0.8	—
Hd	8.8	9.0	9.2
HE	8.8	9.0	9.2
L	0.3	0.5	0.7
L1	—	1.0	—
y	—	—	0.1
θ	0°	—	10°
b2	—	0.5	—
l2	1.0	—	—
MD	—	7.4	—
ME	—	7.4	—



3.9 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)	Ai or Mi = 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)	Ai or Mi = 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.																		

Addressing mode																								Processor status register																	
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0	
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C				
75	4	2				6D	4	3	7D	5	3	79	5	3							61	6	2	71	6	2				N	V	•	•	•	•	Z	C				
35	4	2				2D	4	3	3D	5	3	39	5	3							21	6	2	31	6	2				N	•	•	•	•	•	Z	•				
16	6	2				0E	6	3	1E	7	3																		N	•	•	•	•	•	Z	C					
																											90	2	2				•	•	•	•	•	•	•	•	
																											B0	2	2				•	•	•	•	•	•	•	•	
																												F0	2	2				•	•	•	•	•	•	•	•
						2C	4	3																						M7	M6	•	•	•	•	Z	•				
																											30	2	2				•	•	•	•	•	•	•	•	
																												D0	2	2				•	•	•	•	•	•	•	•

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	#
BPL (Note 4)	N = 0?	This instruction takes a branch to the appointed address if N is 0. The branch address is specified by a relative address. If N is 1, the next instruction is executed.																		
BRA	PC ← PC ± offset	This instruction branches to the appointed address. The branch address is specified by a relative address.																		
BRK	B ← 1 (PC) ← (PC) + 2 M(S) ← PCH S ← S − 1 M(S) ← PCL S ← S − 1 M(S) ← PS S ← S − 1 I ← 1 PCL ← ADL PCH ← ADH	When the BRK instruction is executed, the CPU pushes the current PC contents onto the stack. The BADRS designated in the interrupt vector table is stored into the PC.	00	7	1															
BVC (Note 4)	V = 0?	This instruction takes a branch to the appointed address if V is 0. The branch address is specified by a relative address. If V is 1, the next instruction is executed.																		
BVS (Note 4)	V = 1?	This instruction takes a branch to the appointed address when V is 1. The branch address is specified by a relative address. When V is 0, the next instruction is executed.																		
CLB	Ai or Mi ← 0	This instruction clears the designated bit i of A or M.										1B 20i	2	1				1F 20i	5	2
CLC	C ← 0	This instruction clears C.	18	2	1															
CLD	D ← 0	This instruction clears D.	D8	2	1															
CLI	I ← 0	This instruction clears I.	58	2	1															
CLT	T ← 0	This instruction clears T.	12	2	1															
CLV	V ← 0	This instruction clears V.	B8	2	1															
CMP (Note 3)	When T = 0 A − M When T = 1 M(X) − M	When T = 0, this instruction subtracts the contents of M from the contents of A. The result is not stored and the contents of A or M are not modified. When T = 1, the CMP subtracts the contents of M from the contents of M(X). The result is not stored and the contents of X, M, and A are not modified. M(X) represents the contents of memory where is indicated by X.				C9	2	2							C5	3	2			
COM	M ← M	This instruction takes the one's complement of the contents of M and stores the result in M.													44	5	2			
CPX	X − M	This instruction subtracts the contents of M from the contents of X. The result is not stored and the contents of X and M are not modified.				E0	2	2							E4	3	2			
CPY	Y − M	This instruction subtracts the contents of M from the contents of Y. The result is not stored and the contents of Y and M are not modified.				C0	2	2							C4	3	2			
DEC	A ← A − 1 or M ← M − 1	This instruction subtracts 1 from the contents of A or M.							1A	2	1				C6	5	2			

Addressing mode																											Processor status register													
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C			
																														•	•	•	•	•	•	•	•			
																														•	•	•	•	•	•	•	•			
																														•	•	•	1	•	1	•	•			
																														•	•	•	•	•	•	•	•			
																														•	•	•	•	•	•	•	•			
																														•	•	•	•	•	•	•	•			
																														•	•	•	•	•	•	•	•			
																														•	•	•	•	0	•	•	•			
																														•	•	•	•	0	•	•	•			
																														•	•	•	•	•	0	•	•			
																														•	•	0	•	•	•	•	•			
D5	4	2							CD	4	3	DD	5	3	D9	5	3													N	•	•	•	•	•	•	Z	C		

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															
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			

Addressing mode																											Processor status register													
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C			
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
55	4	2				4D	4	3	5D	5	3	59	5	3							41	6	2	51	6	2				N	•	•	•	•	•	•	Z	•		
F6	6	2				EE	6	3	FE	7	3																		N	•	•	•	•	•	•	Z	•			
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
						4C	3	3							6C	5	3	B2	4	2										•	•	•	•	•	•	•	•	•		
						20	6	3										02	7	2								22	5	2	•	•	•	•	•	•	•	•	•	
B5	4	2				AD	4	3	BD	5	3	B9	5	3							A1	6	2	B1	6	2				N	•	•	•	•	•	•	Z	•		
																															•	•	•	•	•	•	•	•	•	
						B6	4	2	AE	4	3				BE	5	3													N	•	•	•	•	•	•	Z	•		
B4	4	2				AC	4	3	BC	5	3																			N	•	•	•	•	•	•	Z	•		

### 3.9 Machine instructions

[illegible]

### 3.9 Machine instructions

Addressing mode																		Processor status register																						
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C			
56	6	2				4E	6	3	5E	7	3																		0	•	•	•	•	•	•	Z	C			
																														•	•	•	•	•	•	•	•	•		
15	4	2				0D	4	3	1D	5	3	19	5	3							01	6	2	11	6	2				N	•	•	•	•	•	•	Z	•		
																														•	•	•	•	•	•	•	•	•		
																														•	•	•	•	•	•	•	•	•		
																														N	•	•	•	•	•	•	Z	•		
																														(Value saved in stack)										
36	6	2				2E	6	3	3E	7	3																		N	•	•	•	•	•	•	Z	C			
76	6	2				6E	6	3	7E	7	3																		N	•	•	•	•	•	•	Z	C			
																														•	•	•	•	•	•	•	•	•		
																														(Value saved in stack)										
																														•	•	•	•	•	•	•	•	•		

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 $A \leftarrow A - M - \overline{C}$  When T = 1 $M(X) \leftarrow M(X) - M - \overline{C}$	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.

Addressing mode																				Processor status register																				
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C			
F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2				N	V	•	•	•	•	Z	C			
																														•	•	•	•	•	•	•	•			
																														•	•	•	•	•	•	•	•	1		
																														•	•	•	•	•	1	•	•	•		
																														•	•	•	•	•	1	•	•	•		
																														•	•	1	•	•	•	•	•	•		
95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	2				•	•	•	•	•	•	•	•	•		
																														•	•	•	•	•	•	•	•	•		
																														•	•	•	•	•	•	•	•	•		
																														•	•	•	•	•	•	•	•	•		
94	5	2				8C	5	3																					•	•	•	•	•	•	•	•	•			
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		
																														N	•	•	•	•	•	•	Z	•		

# APPENDIX

## 3.9 Machine instructions

Symbol	Contents	Symbol	Contents
IMP	Implied addressing mode	+	Addition
IMM	Immediate addressing mode	−	Subtraction
A	Accumulator or Accumulator addressing mode	∧	Logical OR
BIT, A	Accumulator bit addressing mode	∨	Logical AND
BIT, A, R	Accumulator bit relative addressing mode	⊕	Logical exclusive OR
ZP	Zero page addressing mode	—	Negation
BIT, ZP	Zero page bit addressing mode	←	Shows direction of data flow
BIT, ZP, R	Zero page bit relative addressing mode	X	Index register X
ZP, X	Zero page X addressing mode	Y	Index register Y
ZP, Y	Zero page Y addressing mode	S	Stack pointer
ABS	Absolute addressing mode	PC	Program counter
ABS, X	Absolute X addressing mode	PS	Processor status register
ABS, Y	Absolute Y addressing mode	PCH	8 high-order bits of program counter
IND	Indirect absolute addressing mode	PCL	8 low-order bits of program counter
		ADH	8 high-order bits of address
ZP, IND	Zero page indirect absolute addressing mode	ADL	8 low-order bits of address
		FF	FF in Hexadecimal notation
IND, X	Indirect X addressing mode	nn	Immediate value
IND, Y	Indirect Y addressing mode	zz	Zero page address
REL	Relative addressing mode	M	Memory specified by address designation of any addressing mode
SP	Special page addressing mode	M(X)	Memory of address indicated by contents of index register X
C	Carry flag	M(S)	Memory of address indicated by contents of stack pointer
Z	Zero 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.
I	Interrupt disable flag	M(00, ADL)	Contents of address indicated by zero page ADL
D	Decimal mode flag	Ai	Bit i (i = 0 to 7) of accumulator
B	Break flag	Mi	Bit i (i = 0 to 7) of memory
T	X-modified arithmetic mode flag	OP	Opcode
V	Overflow flag	n	Number of cycles
N	Negative flag	#	Number of bytes

## 3.10 List of instruction code

D3 – D0 D7 – D4	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND, X	JSR ZP, IND	BBS 0, A	—	ORA ZP	ASL ZP	BBS 0, ZP	PHP	ORA IMM	ASL A	SEB 0, A	—	ORA ABS	ASL ABS	SEB 0, ZP
0001	1	BPL	ORA IND, Y	CLT	BBC 0, A	—	ORA ZP, X	ASL ZP, X	BBC 0, ZP	CLC	ORA ABS, Y	DEC A	CLB 0, A	—	ORA ABS, X	ASL ABS, X	CLB 0, ZP
0010	2	JSR ABS	AND IND, X	JSR SP	BBS 1, A	BIT ZP	AND ZP	ROL ZP	BBS 1, ZP	PLP	AND IMM	ROL A	SEB 1, A	BIT ABS	AND ABS	ROL ABS	SEB 1, ZP
0011	3	BMI	AND IND, Y	SET	BBC 1, A	—	AND ZP, X	ROL ZP, X	BBC 1, ZP	SEC	AND ABS, Y	INC A	CLB 1, A	LDM ZP	AND ABS, X	ROL ABS, X	CLB 1, ZP
0100	4	RTI	EOR IND, X	STP	BBS 2, A	COM ZP	EOR ZP	LSR ZP	BBS 2, ZP	PHA	EOR IMM	LSR A	SEB 2, A	JMP ABS	EOR ABS	LSR ABS	SEB 2, ZP
0101	5	BVC	EOR IND, Y	—	BBC 2, A	—	EOR ZP, X	LSR ZP, X	BBC 2, ZP	CLI	EOR ABS, Y	—	CLB 2, A	—	EOR ABS, X	LSR ABS, X	CLB 2, ZP
0110	6	RTS	ADC IND, X	—	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	—	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

## 3.11 SFR memory map

### 3.11 SFR memory map

0000 <sub>16</sub>	Port P0 (P0)	0020 <sub>16</sub>	USB interrupt control register (USBICON)
0001 <sub>16</sub>	Port P0 direction register (P0D)	0021 <sub>16</sub>	USB transmit data byte number set register 0 (EP0BYTE)
0002 <sub>16</sub>	Port P1 (P1)	0022 <sub>16</sub>	USB transmit data byte number set register 1 (EP1BYTE)
0003 <sub>16</sub>	Port P1 direction register (P1D)	0023 <sub>16</sub>	USBPID control register 0 (EP0PID)
0004 <sub>16</sub>	Port P2 (P2)	0024 <sub>16</sub>	USBPID control register 1 (EP1PID)
0005 <sub>16</sub>	Port P2 direction register (P2D)	0025 <sub>16</sub>	USB address register (USBA)
0006 <sub>16</sub>	Port P3 (P3)	0026 <sub>16</sub>	USB sequence bit initialization register (INISQ1)
0007 <sub>16</sub>	Port P3 direction register (P3D)	0027 <sub>16</sub>	USB control register (USBCON)
0008 <sub>16</sub>	Port P4 (P4)	0028 <sub>16</sub>	Prescaler 12 (PRE12)
0009 <sub>16</sub>	Port P4 direction register (P4D)	0029 <sub>16</sub>	Timer 1 (T1)
000A <sub>16</sub>		002A <sub>16</sub>	Timer 2 (T2)
000B <sub>16</sub>		002B <sub>16</sub>	Timer X mode register (TM)
000C <sub>16</sub>		002C <sub>16</sub>	Prescaler X (PREX)
000D <sub>16</sub>		002D <sub>16</sub>	Timer X (TX)
000E <sub>16</sub>		002E <sub>16</sub>	Timer count source set register (TCSS)
000F <sub>16</sub>		002F <sub>16</sub>	
0010 <sub>16</sub>		0030 <sub>16</sub>	Serial I/O2 control register (SIO2CON)
0011 <sub>16</sub>		0031 <sub>16</sub>	Serial I/O2 register (SIO2)
0012 <sub>16</sub>		0032 <sub>16</sub>	
0013 <sub>16</sub>		0033 <sub>16</sub>	
0014 <sub>16</sub>		0034 <sub>16</sub>	A-D control register (ADCON)
0015 <sub>16</sub>		0035 <sub>16</sub>	A-D conversion register (low-order) (ADL)
0016 <sub>16</sub>	Pull-up control register (PULL)	0036 <sub>16</sub>	A-D conversion register (high-order) (ADH)
0017 <sub>16</sub>	Port P1P3 control register (P1P3C)	0037 <sub>16</sub>	
0018 <sub>16</sub>	Transmit/Receive buffer register (TB/RB)	0038 <sub>16</sub>	MISRG
0019 <sub>16</sub>	USB status register (USBSTS)/UART status register (UARTSTS)	0039 <sub>16</sub>	Watchdog timer control register (WDTCON)
001A <sub>16</sub>	Serial I/O1 control register (SIO1CON)	003A <sub>16</sub>	Interrupt edge selection register (INTEDGE)
001B <sub>16</sub>	UART control register (UARTCON)	003B <sub>16</sub>	CPU mode register (CPUM)
001C <sub>16</sub>	Baud rate generator (BRG)	003C <sub>16</sub>	Interrupt request register 1 (IREQ1)
001D <sub>16</sub>	USB data toggle synchronization register ( TRSYNC)	003D <sub>16</sub>	
001E <sub>16</sub>	USB interrupt source discrimination register 1 (USBIR1)	003E <sub>16</sub>	Interrupt control register 1 (ICON1)
001F <sub>16</sub>	USB interrupt source discrimination register 2 (USBIR2)	003F <sub>16</sub>	



## 3.12 Pin configurations

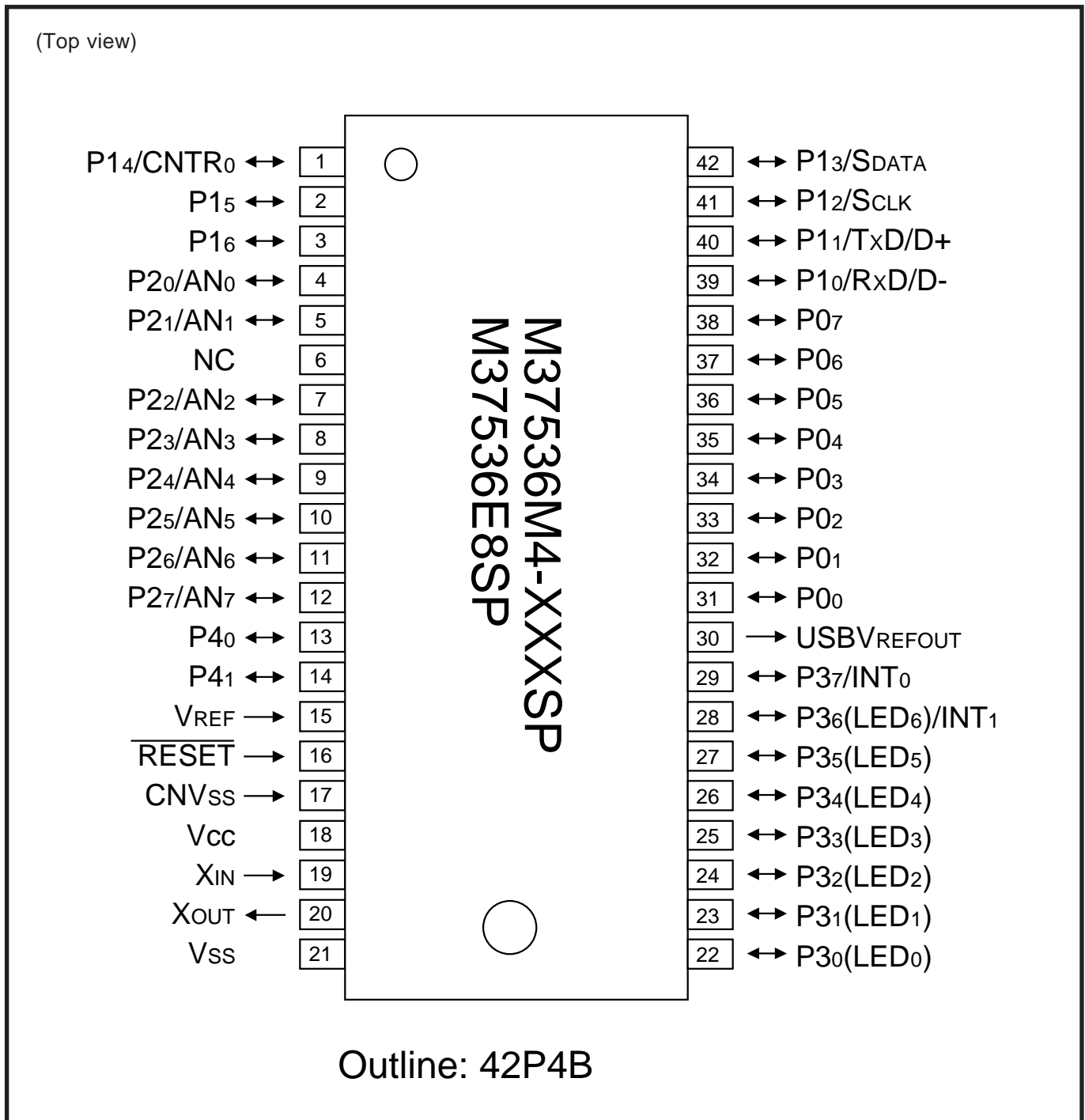


Fig. 3.12.1 M37536M4-XXXSP, M37536E8SP pin configuration

# APPENDIX

## 3.12 Pin configurations

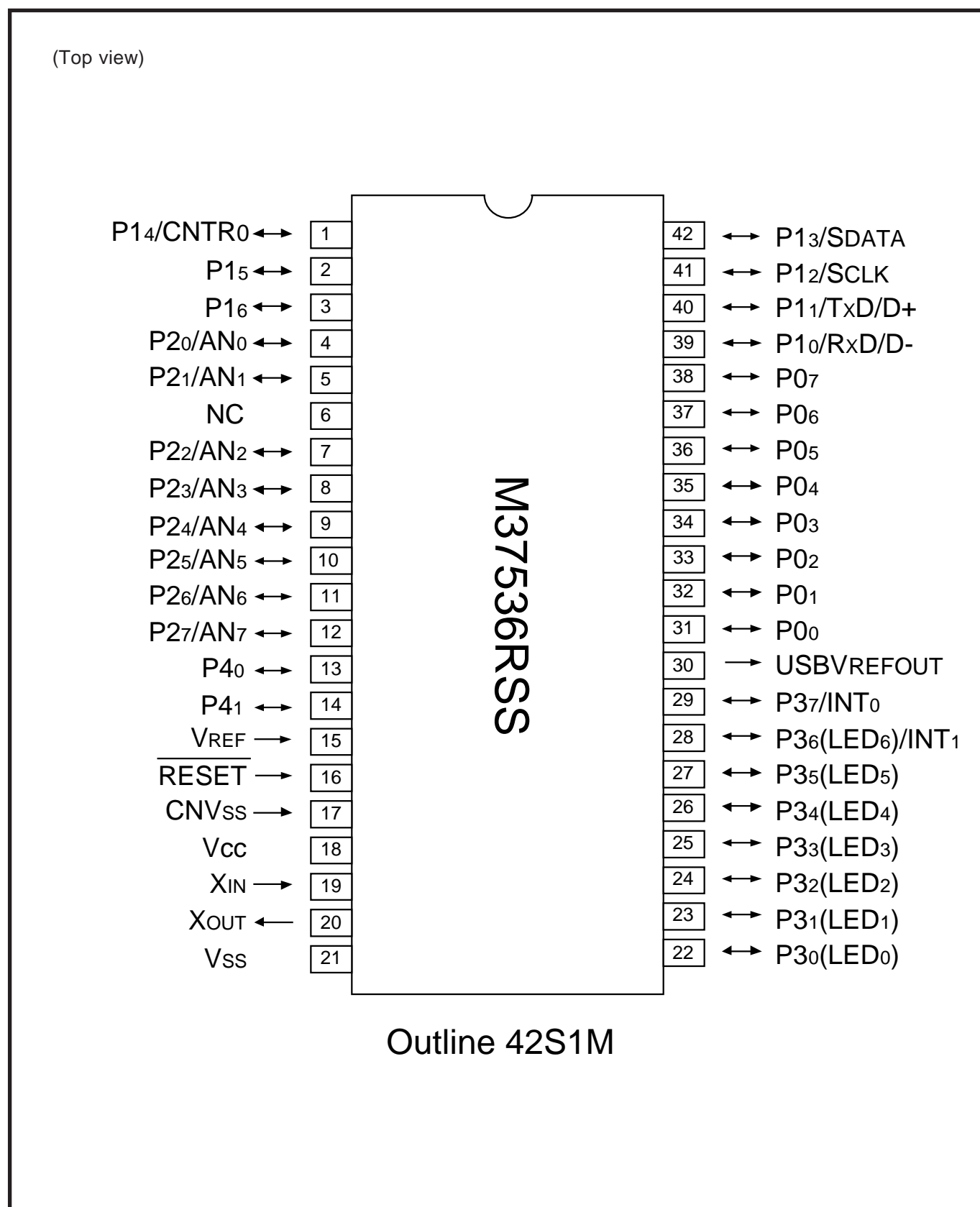


Fig. 3.12.2 M37536RSS pin configuration

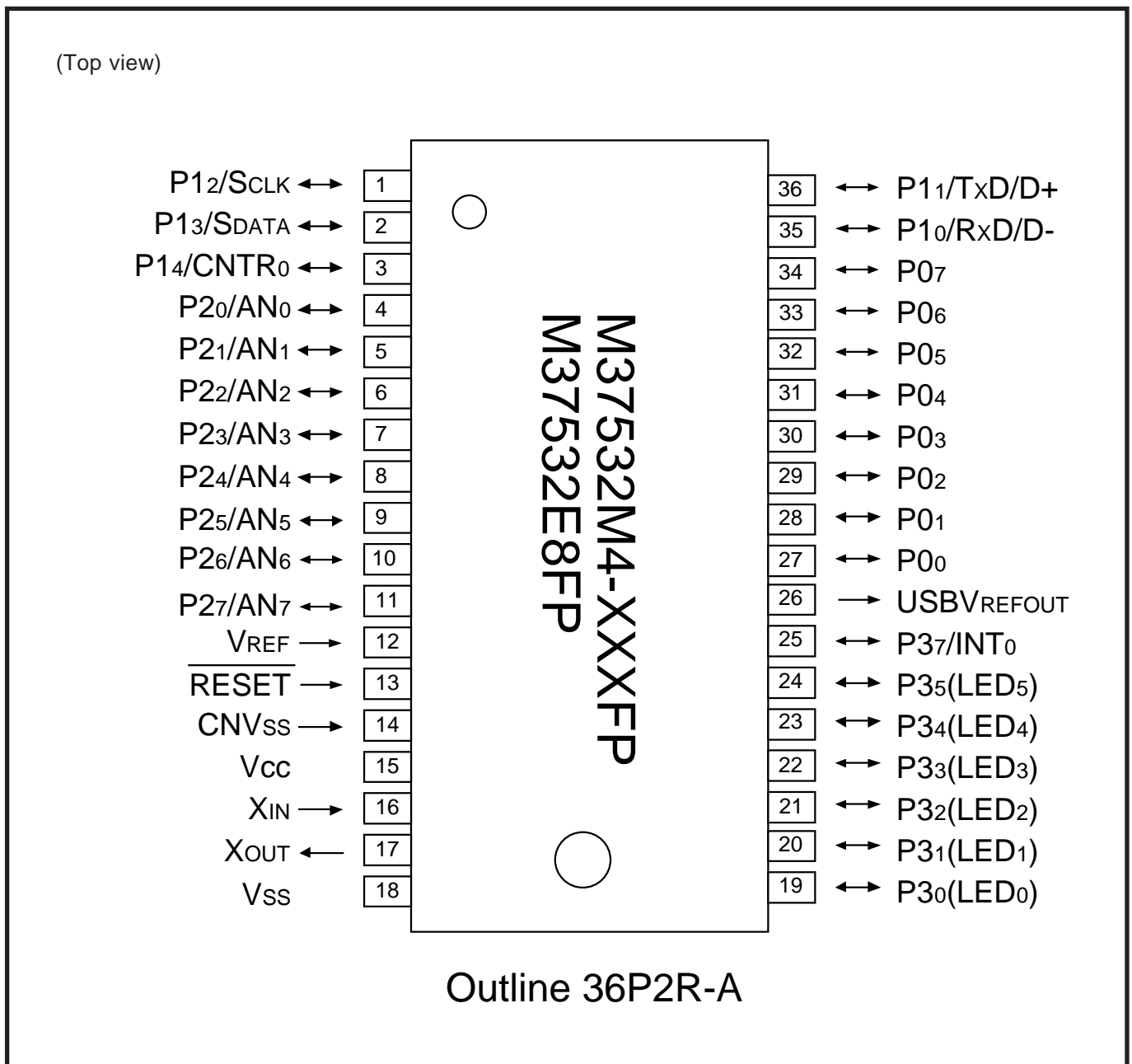


Fig. 3.12.3 M37532M4-XXXXFP, M37532E8FP pin configuration

# APPENDIX

## 3.12 Pin configurations

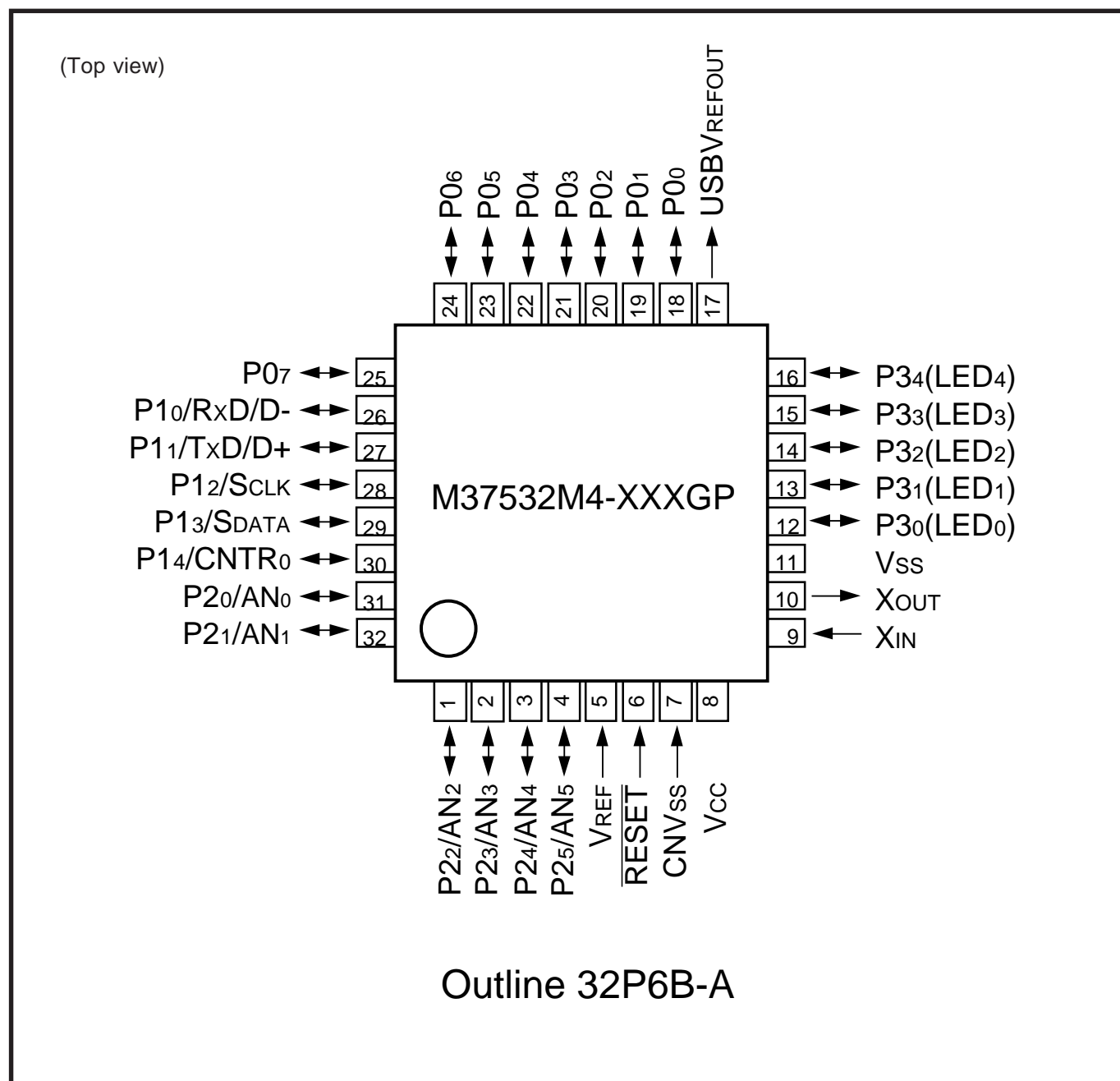
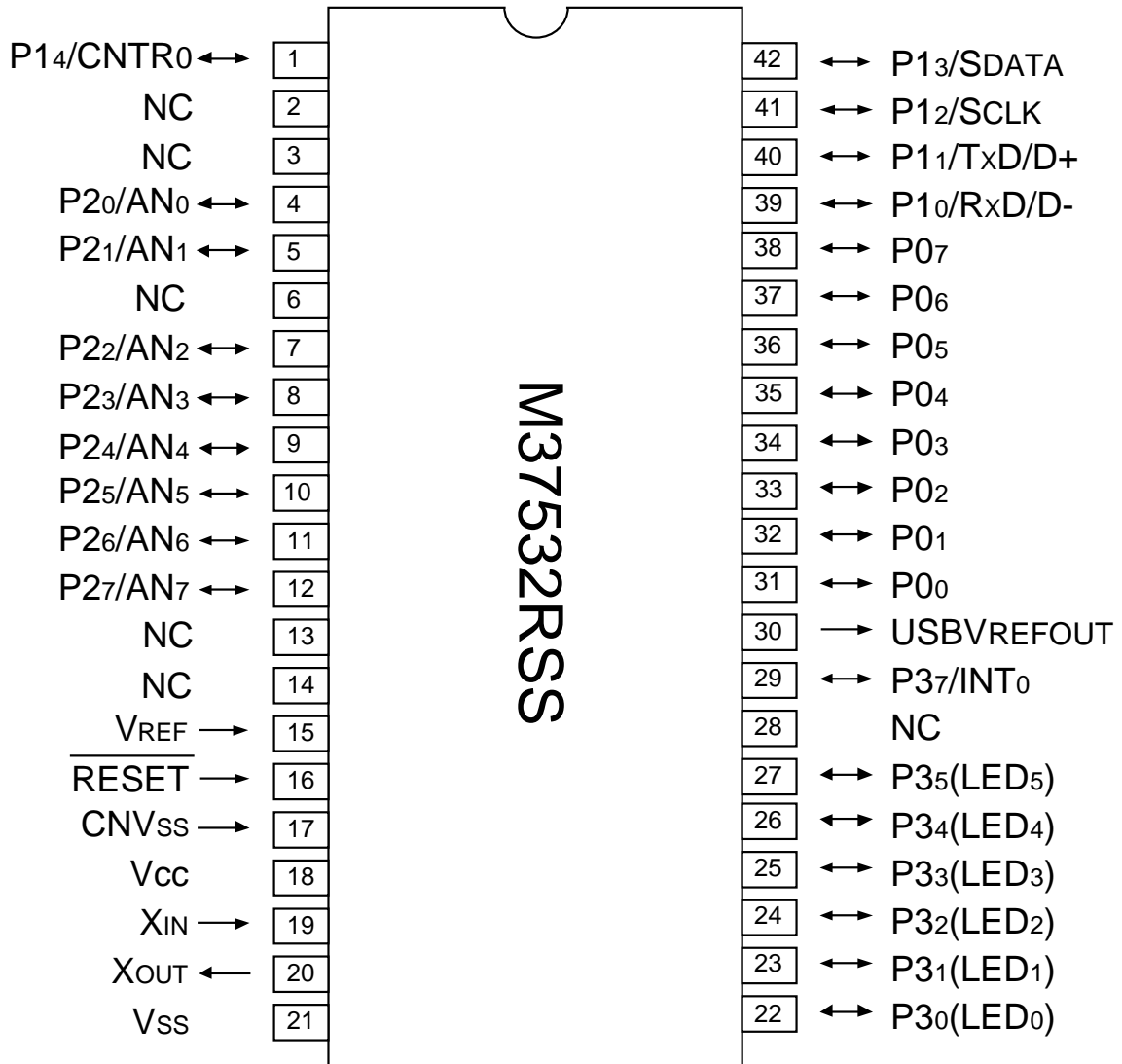


Fig. 3.12.4 M37532M4-XXXGP pin configuration

(Top view)



Outline 42S1M

Fig. 3.12.5 M37532RSS pin configuration

# APPENDIX

## 3.12 Pin configurations

---

### *MEMO*

**MITSUBISHI SEMICONDUCTORS  
USER'S MANUAL  
7532/7536 Group**

---

Jun. First Edition 1999

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.

**©1999 MITSUBISHI ELECTRIC CORPORATION**

# User's Manual

## 7532/7536 Group



**mitsubishi electric corporation**

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



# REVISION DESCRIPTION LIST

# 7532/7536 GROUP USER'S MANUAL

Rev. No.	Revision Description	Rev. date
1.0	First Edition	990625
2.0	<ul style="list-style-type: none"> <li>• The contents of Table of Contents/List of Figures/List of Tables are revised.</li> <li>• 1-1: A title is added.</li> <li>• 1-35: Fig. 40 <math>\phi</math> is added.</li> <li>• 1-39: A note is added.</li> <li>• 1-50 to 1-54: New contents are added.</li> <li>• 2-54: The value and the contents of the table is revised.</li> <li>• 3-3, 3-5: Values are revised.</li> <li>• 3-60: Package A is revised.</li> </ul>	990721
2.1	<ul style="list-style-type: none"> <li>• 1-49 Fig. 52, 1-52 Fig. 55, 3-80 Fig. 3.12.4: Pin functions revised. Pin 26: P10/RxD/D- (correct) Pin 27: P11/TxD/D+ (correct)</li> <li>• 2-65 Fig. 2.4.13: Description about EP0BYTE setting eliminated.</li> </ul>	991015
2.2	<p>Updated as follows:</p> <ul style="list-style-type: none"> <li>•Page1-2; Power dissipation to 30 mW</li> <li>•Page 1-11, Fig. 9; Start address of Interrupt vector area to FFEC16</li> <li>•Pages 2-8, 3-20 (1) Notes in stand-by state •Reason; The following explanation is deleted : The output 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,</li> <li>•Page 3-5, Table 3.1.5; Parameter to Linearity error from former Linear error</li> </ul>	991110
2.3	<ul style="list-style-type: none"> <li>•Page 1-15: Fig.13 Note revised</li> <li>•Page 1-32: Fig.33 Note revised</li> <li>•Page 1-35: Description revised; <math>\overline{\text{RESET}}</math> “L” pulse width 2 <math>\mu\text{s}</math> <math>\rightarrow</math> 15 <math>\mu\text{s}</math></li> <li>•Page 2-41: Fig.2.3.20 Note added</li> <li>•Page 2-76: Fig.2.6.2 revised</li> <li>•Page 3-4: Table 3.1.3 Hysteresis “<math>\overline{\text{RESET}}</math>” added</li> <li>•Page 3-6: Table 3.1.6 <math>t_w(\overline{\text{RESET}})</math> revised; 2 <math>\rightarrow</math> 15</li> </ul>	000614
2.4	•Pages 1-15, 1-16, 1-35, 2-41: Character fonts errors revised	000905