



8

7534 Group

User's Manual

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

User's Manual

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
 - (1) artificial life support devices or systems
 - (2) surgical implantations
 - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
 - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable.

When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

- The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

BEFORE USING THIS MANUAL

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

1. Organization

● CHAPTER 1 HARDWARE

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

● CHAPTER 2 APPLICATION

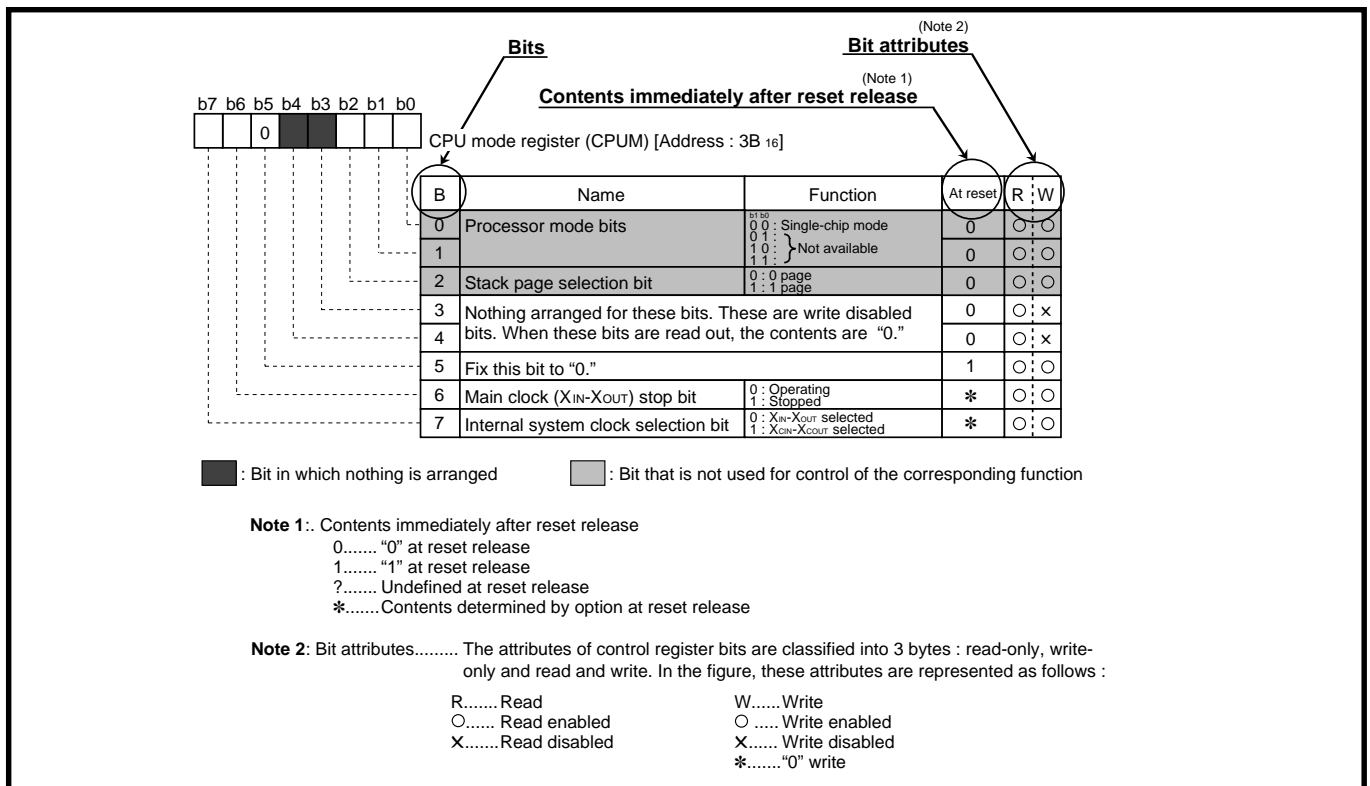
This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of relevant registers.

● CHAPTER 3 APPENDIX

This chapter includes necessary information for systems development using the microcomputer, such as the electrical characteristics, the list of registers.

2. Structure of register

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



3. Supplementation

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

For details of development support tools, refer to the "Renesas Technology" Homepage (<http://www.renesas.com>).

Table of contents

CHAPTER 1 HARDWARE

DESCRIPTION	2
FEATURES	2
APPLICATION	2
PIN CONFIGURATION	2
FUNCTIONAL BLOCK	5
PIN DESCRIPTION	8
GROUP EXPANSION	9
FUNCTIONAL DESCRIPTION	10
Central Processing Unit (CPU)	10
Memory	14
I/O Ports	16
Interrupts	20
Timers	23
Serial Interface	25
A/D Converter	36
Reset Circuit	38
Clock Generating Circuit	40
NOTES ON PROGRAMMING	42
NOTES ON USE	42
DATA REQUIRED FOR MASK ORDERS	43
FUNCTIONAL DESCRIPTION SUPPLEMENT	45
Interrupt	45
Timing After Interrupt	46
A/D Converter	47
Stop mode	49
Wait mode	50
DESCRIPTION OF IMPROVED USB FUNCTION FOR 7534 GROUP	51
DIFFERENCES AMONG 32-PIN, 36-PIN AND 42-PIN	51
DESCRIPTION SUPPLEMENT FOR USE OF USB FUNCTION STABLY	53

CHAPTER 2 APPLICATION

2.1 I/O port	2
2.1.1 Memory map	2
2.1.2 Relevant registers	2
2.1.3 Application example of key-on wake up	6
2.1.4 Handling of unused pins	7
2.1.5 Notes on input and output pins	8
2.1.6 Termination of unused pins	9
2.2 Timer	10
2.2.1 Memory map	10
2.2.2 Relevant registers	10
2.2.3 Timer application examples	16
2.3 Serial I/O	29
2.3.1 Memory map	29
2.3.2 Relevant registers	29
2.3.3 Serial I/O connection examples	35

2.3.4 Serial I/O transfer data format	37
2.3.5 Serial I/O application examples	38
2.3.6 Notes on serial I/O	49
2.4 USB	50
2.4.1 Outline of USB	50
2.4.2 Memory map	56
2.4.3 Relevant registers	57
2.4.4 USB application example	62
2.4.5 Notes concerning USB	69
2.5 A/D converter	71
2.5.1 Memory map	71
2.5.2 Relevant registers	71
2.5.3 A/D converter application examples	75
2.5.4 Notes on A/D converter	77
2.6 Reset.....	78
2.6.1 Connection example of reset IC	78
2.6.2 Notes on RESET pin	78

CHAPTER 3 APPENDIX

3.1 Electrical characteristics	2
3.1.1 Absolute maximum ratings	2
3.1.2 Recommended operating conditions	3
3.1.3 Electrical characteristics	4
3.1.4 A/D converter characteristics	5
3.1.5 Timing requirements	6
3.1.6 Switching characteristics	6
3.2 Typical characteristics	8
3.2.1 Power source current characteristic example ($I_{CC}-V_{CC}$ characteristic)	8
3.2.2 $V_{OH}-I_{OH}$ characteristic example	11
3.2.3 A/D conversion typical characteristics example	15
3.3 Notes on use	17
3.3.1 Notes on interrupts	17
3.3.2 Notes on serial I/O	18
3.3.3 Notes on A/D converter	19
3.3.4 Notes on watchdog timer	20
3.3.5 Notes on \overline{RESET} pin	20
3.3.6 Notes on input and output pins	20
3.3.7 Notes on programming	21
3.3.8 Programming and test of built-in PROM version	22
3.3.9 Notes on built-in PROM version	23
3.3.10 Termination of unused pins	24
3.3.11 Notes on CPU mode register	25
3.3.12 Notes on using 32-pin version	25
3.3.13 Electric characteristic differences among mask ROM and One Time PROM version MCUs ...	25
3.3.14 Note on power source voltage	25
3.3.15 USB communication	26
3.4 Countermeasures against noise	27
3.4.1 Shortest wiring length	27
3.4.2 Connection of bypass capacitor across V_{SS} line and V_{CC} line	29
3.4.3 Wiring to analog input pins	30

3.4.4 Oscillator concerns.....	30
3.4.5 Setup for I/O ports.....	32
3.4.6 Providing of watchdog timer function by software	33
3.5 List of registers	34
3.6 Package outline	53
3.7 List of instruction code	55
3.8 Machine instructions	56
3.9 SFR memory map	67
3.10 Pin configurations	68

List of figures

CHAPTER 1 HARDWARE

Fig. 1 Pin configuration of M37534M4-XXXFP, M37534E8FP	2
Fig. 2 Pin configuration of M37534M4-XXXGP, M37534E4GP	3
Fig. 3 Pin configuration of M37534RSS, M37534M4-XXXSP, M37534E8SP	4
Fig. 4 Functional block diagram (PRSP0036GA-A package type)	5
Fig. 5 Functional block diagram (PLQP0032GB-A package type)	6
Fig. 6 Functional block diagram (PRDP0042BA-A package type)	7
Fig. 7 Memory expansion plan	9
Fig. 8 740 Family CPU register structure	10
Fig. 9 Register push and pop at interrupt generation and subroutine call	11
Fig. 10 Structure of CPU mode register	13
Fig. 11 Switching method of CPU mode register	13
Fig. 12 Memory map diagram	14
Fig. 13 Memory map of special function register (SFR)	15
Fig. 14 Structure of pull-up control register	16
Fig. 15 Structure of port P1P3 control register	16
Fig. 16 Block diagram of ports (1)	18
Fig. 17 Block diagram of ports (2)	19
Fig. 18 Interrupt control	21
Fig. 19 Structure of Interrupt-related registers	21
Fig. 20 Connection example when using key input interrupt and port P0 block diagram ...	22
Fig. 21 Structure of timer X mode register	23
Fig. 22 Timer count source set register	23
Fig. 23 Block diagram of timer X, timer 1 and timer 2	24
Fig. 24 Block diagram of UART serial I/O1	25
Fig. 25 Operation of UART serial I/O1 function	25
Fig. 26 Continuous transmission operation of UART serial I/O	26
Fig. 27 USB mode block diagram	27
Fig. 28 USB transceiver block diagram	27
Fig. 29 Structure of serial I/O-related registers (1)	28
Fig. 30 Structure of serial I/O1-related registers (2)	29
Fig. 31 Structure of serial I/O1-related registers (3)	30
Fig. 32 Structure of serial I/O1-related registers (4)	31
Fig. 33 Structure of serial I/O1-related registers (5)	32
Fig. 34 Structure of serial I/O2 control registers	34
Fig. 35 Block diagram of serial I/O2	34
Fig. 36 Serial I/O2 timing (LSB first)	35
Fig. 37 Structure of A/D control register	36
Fig. 38 Structure of A/D conversion register	36
Fig. 39 Block diagram of A/D converter	36
Fig. 40 Block diagram of watchdog timer	37
Fig. 41 Structure of watchdog timer control register	37
Fig. 42 Example of reset circuit	38
Fig. 43 Timing diagram at reset	38
Fig. 44 Internal status of microcomputer at reset	39
Fig. 45 External circuit of ceramic resonator	40
Fig. 46 External clock input circuit	40
Fig. 47 Structure of MISRG	40

Fig. 48 Block diagram of system clock generating circuit (for ceramic resonator)	41
Fig. 49 Countermeasure (2) by software	43
Fig. 50 Method to stabilize A/D conversion accuracy	43
Fig. 51 Programming and testing of One Time PROM version	44
Fig. 52 Timing chart after an interrupt occurs	46
Fig. 53 Time up to execution of the interrupt processing routine	46
Fig. 54 A/D conversion equivalent circuit	48
Fig. 55 A/D conversion timing chart	48
Fig. 56 Handling of V_{CC} , $USBV_{REFOUT}$ pins of M37534M4-XXXFP, M37534E8FP	53
Fig. 57 Handling of V_{CC} , $USBV_{REFOUT}$ pins of M37534M4-XXXGP, M37534E4GP	54
Fig. 58 Handling of V_{CC} , $USBV_{REFOUT}$ pins of M37534E8SP, M37534M4-XXXSP, M37534RSS	55

CHAPTER 2 APPLICATION

Fig. 2.1.1 Memory map of registers relevant to I/O port	2
Fig. 2.1.2 Structure of Port Pi ($i = 0$ to 4)	2
Fig. 2.1.3 Structure of Port Pi direction register ($i = 0$ to 4)	3
Fig. 2.1.4 Structure of Pull-up control register	3
Fig. 2.1.5 Structure of P1P3 control register	4
Fig. 2.1.6 Structure of Interrupt edge selection register	4
Fig. 2.1.7 Structure of Interrupt request register 1	5
Fig. 2.1.8 Structure of Interrupt control register 1	5
Fig. 2.1.9 Relevant registers setting	6
Fig. 2.1.10 Application circuit example	6
Fig. 2.1.11 Control procedure	7
Fig. 2.2.1 Memory map of registers relevant to timers	10
Fig. 2.2.2 Structure of Prescaler 12, Prescaler X	10
Fig. 2.2.3 Structure of Timer 1	11
Fig. 2.2.4 Structure of Timer 2	11
Fig. 2.2.5 Structure of Timer X	12
Fig. 2.2.6 Structure of Timer X mode register	13
Fig. 2.2.7 Structure of Timer count source set register	14
Fig. 2.2.8 Structure of Interrupt edge selection register	14
Fig. 2.2.9 Structure of Interrupt request register 1	15
Fig. 2.2.10 Structure of Interrupt control register 1	15
Fig. 2.2.11 Timers connection and setting of division ratios	17
Fig. 2.2.12 Relevant registers setting	18
Fig. 2.2.13 Control procedure	19
Fig. 2.2.14 Peripheral circuit example	20
Fig. 2.2.15 Timers connection and setting of division ratios	20
Fig. 2.2.16 Relevant registers setting	21
Fig. 2.2.17 Control procedure	22
Fig. 2.2.18 Judgment method of valid/invalid of input pulses	23
Fig. 2.2.19 Relevant registers setting	24
Fig. 2.2.20 Control procedure	25
Fig. 2.2.21 Timers connection and setting of division ratios	26
Fig. 2.2.22 Relevant registers setting	27
Fig. 2.2.23 Control procedure	28
Fig. 2.3.1 Memory map of registers relevant to serial I/O	29
Fig. 2.3.2 Structure of Transmit/Receive buffer register	29
Fig. 2.3.3 Structure of UART status register	30
Fig. 2.3.4 Structure of Serial I/O1 control register	30

Fig. 2.3.5 Structure of UART control register	31
Fig. 2.3.6 Structure of Baud rate generator	31
Fig. 2.3.7 Structure of Serial I/O2 control register	32
Fig. 2.3.8 Structure of Serial I/O2 register	32
Fig. 2.3.9 Structure of Interrupt edge selection register	33
Fig. 2.3.10 Structure of Interrupt request register 1	33
Fig. 2.3.11 Structure of Interrupt control register 1	34
Fig. 2.3.12 Serial I/O connection examples (1)	35
Fig. 2.3.13 Serial I/O connection examples (2)	36
Fig. 2.3.14 Serial I/O transfer data format	37
Fig. 2.3.15 Connection diagram	38
Fig. 2.3.16 Timing chart	38
Fig. 2.3.17 Registers setting relevant to transmission side	39
Fig. 2.3.18 Transmission data setting of serial I/O2	40
Fig. 2.3.19 Registers setting relevant to reception side	40
Fig. 2.3.20 Control procedure of transmission side	41
Fig. 2.3.21 Control procedure of reception side	42
Fig. 2.3.22 Connection diagram	43
Fig. 2.3.23 Timing chart	43
Fig. 2.3.24 Registers setting relevant to transmission side	45
Fig. 2.3.25 Registers setting relevant to reception side	46
Fig. 2.3.26 Control procedure of transmission side	47
Fig. 2.3.27 Control procedure of reception side	48
Fig. 2.3.28 Sequence of clearing serial I/O	49
Fig. 2.4.1 Communication sequence of USB	51
Fig. 2.4.2 Data structure of USB packet	52
Fig. 2.4.3 USB (L.S.) interface	55
Fig. 2.4.4 USB (L.S.) connection example	55
Fig. 2.4.5 Memory map of registers relevant to USB	56
Fig. 2.4.6 Description of the register structure	57
Fig. 2.4.7 Register structures relevant to USB (1)	58
Fig. 2.4.8 Register structures relevant to USB (2)	59
Fig. 2.4.9 Register structures relevant to USB (3)	60
Fig. 2.4.10 Register structures relevant to USB (4)	61
Fig. 2.4.11 Control method of control sequence	62
Fig. 2.4.12 Timing chart of the transaction according to each token	63
Fig. 2.4.13 USB interrupt processing example (OUT token)	65
Fig. 2.4.14 USB interrupt processing example (IN token)	66
Fig. 2.4.15 Data read timing of SETUP token	67
Fig. 2.4.16 Data read timing of OUT token	67
Fig. 2.4.17 Data read timing of IN token (endpoint 0) and IN token (endpoint 1) token	67
Fig. 2.4.18 Timing chart of each signal	68
Fig. 2.4.19 Example for determination of resume interrupt	69
Fig. 2.4.20 Processing for width of SE0 signal	69
Fig. 2.4.21 Countermeasure (2) by software	70
Fig. 2.5.1 Memory map of registers relevant to A/D converter	71
Fig. 2.5.2 Structure of A/D control register	71
Fig. 2.5.3 Structure of A/D conversion register (high-order)	72
Fig. 2.5.4 Structure of A/D conversion register (low-order)	72
Fig. 2.5.5 Structure of Interrupt edge selection register	73
Fig. 2.5.6 Structure of Interrupt request register 1	73
Fig. 2.5.7 Structure of Interrupt control register 1	74

Fig. 2.5.8 Connection diagram	75
Fig. 2.5.9 Relevant registers setting	75
Fig. 2.5.10 Control procedure for 8-bit read	76
Fig. 2.5.11 Control procedure for 10-bit read	76
Fig. 2.5.12 Method to stabilize A/D conversion accuracy	77
Fig. 2.6.1 Example of poweron reset circuit	78

CHAPTER 3 APPENDIX

Fig. 3.1.1 Power source current measurement circuit in USB mode at oscillation stop	5
Fig. 3.1.2 Output switching characteristics measurement circuit	6
Fig. 3.1.3 Timing chart	7
Fig. 3.2.1 $I_{CC}-V_{CC}$ characteristic example (in double-speed mode)	8
Fig. 3.2.2 $I_{CC}-V_{CC}$ characteristic example (at WIT instruction execution)	8
Fig. 3.2.3 $I_{CC}-V_{CC}$ characteristic example (At STP instruction execution, $T_a = 25\text{ }^\circ\text{C}$)	9
Fig. 3.2.4 $I_{CC}-V_{CC}$ characteristic example (At STP instruction execution, $T_a = 85\text{ }^\circ\text{C}$)	9
Fig. 3.2.5 $I_{CC}-V_{CC}$ characteristic example (at USB suspend, $T_a = 25\text{ }^\circ\text{C}$)	10
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)	10
Fig. 3.2.7 $V_{OH}-I_{OH}$ characteristic example of P-channel ($T_a = 25\text{ }^\circ\text{C}$): normal port	11
Fig. 3.2.8 $V_{OH}-I_{OH}$ characteristic example of P-channel ($T_a = 85\text{ }^\circ\text{C}$): normal port	11
Fig. 3.2.9 $V_{OL}-I_{OL}$ characteristic example of N-channel ($T_a = 25\text{ }^\circ\text{C}$): Normal port	12
Fig. 3.2.10 $V_{OL}-I_{OL}$ characteristic example of N-channel ($T_a = 85\text{ }^\circ\text{C}$): Normal port	12
Fig. 3.2.11 $V_{OL}-I_{OL}$ characteristic example of N-channel ($T_a = 25\text{ }^\circ\text{C}$): LED drive port	13
Fig. 3.2.12 $V_{OL}-I_{OL}$ characteristic example N-channel ($T_a = 85\text{ }^\circ\text{C}$): LED drive port	13
Fig. 3.2.13 "L" input current of port at pull-up transistor connected	14
Fig. 3.2.14 Definition of A/D conversion accuracy	15
Fig. 3.2.15 A/D conversion typical characteristic example	16
Fig. 3.3.1 Sequence of switch the detection edge	17
Fig. 3.3.2 Sequence of check of interrupt request bit	17
Fig. 3.3.3 Structure of interrupt control register 1	18
Fig. 3.3.4 Sequence of clearing serial I/O	18
Fig. 3.3.5 Method to stabilize A/D conversion accuracy	19
Fig. 3.3.6 Initialization of processor status register	21
Fig. 3.3.7 Sequence of PLP instruction execution	21
Fig. 3.3.8 Stack memory contents after PHP instruction execution	21
Fig. 3.3.9 Status flag at decimal calculations	22
Fig. 3.3.10 Programming and testing of One Time PROM version	22
Fig. 3.3.11 Switching method of CPU mode register	25
Fig. 3.3.12 Countermeasure (2) by software	26
Fig. 3.4.1 Selection of packages	27
Fig. 3.4.2 Wiring for the $\overline{\text{RESET}}$ pin	27
Fig. 3.4.3 Wiring for clock I/O pins	28
Fig. 3.4.4 Wiring for CNV_{SS} pin	28
Fig. 3.4.5 Wiring for the V_{PP} pin of the One Time PROM	29
Fig. 3.4.6 Bypass capacitor across the V_{SS} line and the V_{CC} line	29
Fig. 3.4.7 Analog signal line and a resistor and a capacitor	30
Fig. 3.4.8 Wiring for a large current signal line	30
Fig. 3.4.9 Wiring of signal lines where potential levels change frequently	31
Fig. 3.4.10 V_{SS} pattern on the underside of an oscillator	31
Fig. 3.4.11 Setup for I/O ports	32
Fig. 3.4.12 Watchdog timer by software	33

Fig. 3.5.1 Structure of Port Pi (i = 0 to 4).....	34
Fig. 3.5.2 Structure of Port Pi direction register (i = 0 to 4)	34
Fig. 3.5.3 Structure of Pull-up control register	35
Fig. 3.5.4 Structure of Port P1P3 control register	35
Fig. 3.5.5 Structure of Transmit/Receive buffer register	36
Fig. 3.5.6 Structure of UART status register	36
Fig. 3.5.7 Structure of USB status register	37
Fig. 3.5.8 Structure of Serial I/O1 control register	38
Fig. 3.5.9 Structure of UART control register	38
Fig. 3.5.10 Structure of Baud rate generator	39
Fig. 3.5.11 Structure of USB data toggle synchronization register	39
Fig. 3.5.12 Structure of USB interrupt source discrimination register 1	39
Fig. 3.5.13 Structure of USB interrupt source discrimination register 2	40
Fig. 3.5.14 Structure of USB interrupt control register	40
Fig. 3.5.15 Structure of USB transmit data byte number set register 0	41
Fig. 3.5.16 Structure of USB transmit data byte number set register 1	41
Fig. 3.5.17 Structure of USB PID control register 0	41
Fig. 3.5.18 Structure of USB PID control register 1	42
Fig. 3.5.19 Structure of USB address register	42
Fig. 3.5.20 Structure of USB sequence bit initialization register	42
Fig. 3.5.21 Structure of USB control register	42
Fig. 3.5.22 Structure of Prescaler 12, Prescaler X	43
Fig. 3.5.23 Structure of Timer 1	43
Fig. 3.5.24 Structure of Timer 2	44
Fig. 3.5.25 Structure of Timer X mode register	45
Fig. 3.5.26 Structure of Timer X	46
Fig. 3.5.27 Structure of Timer count source set register	46
Fig. 3.5.28 Structure of Serial I/O2 control register	47
Fig. 3.5.29 Structure of Serial I/O2 register	47
Fig. 3.5.30 Structure of A/D control register	48
Fig. 3.5.31 Structure of A/D conversion register (high-order)	49
Fig. 3.5.32 Structure of A/D conversion register (low-order)	49
Fig. 3.5.33 Structure of MISRG	50
Fig. 3.5.34 Structure of Watchdog timer control register	50
Fig. 3.5.35 Structure of Interrupt edge selection register	51
Fig. 3.5.36 Structure of CPU mode register	51
Fig. 3.5.37 Structure of Interrupt request register 1	52
Fig. 3.5.38 Structure of Interrupt control register 1	52
Fig. 3.10.1 M37534M4-XXXFP, M37534E8FP pin configuration	68
Fig. 3.10.2 M37534M4-XXXGP, M37534E4GP pin configuration	69
Fig. 3.10.3 M37534M4-XXXSP, M37534E8SP, M37534RSS pin configuration	70

List of tables

CHAPTER 1 HARDWARE

Table 1 Pin description	8
Table 2 List of supported products	9
Table 3 Push and pop instructions of accumulator or processor status register	11
Table 4 Set and clear instructions of each bit of processor status register	12
Table 5 I/O port function table	17
Table 6 Interrupt vector address and priority	20
Table 7 Relation of the width of SE0 and the state of the device	33
Table 8 Special programming adapter	43
Table 9 Interrupt sources, vector addresses and interrupt priority	44
Table 10 Change of A/D conversion register during A/D conversion	46
Table 11 Stop mode state	48
Table 12 Wait mode state	49
Table 13 Description of improved USB function for 7534 Group	50
Table 14 Differences among 32-pin, 36-pin and 42-pin	50
Table 15 Differences among 32-pin, 36-pin and 42-pin (SFR)	51

CHAPTER 2 APPLICATION

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

CHAPTER 3 APPENDIX

Table 3.1.1 Absolute maximum ratings	2
Table 3.1.2 Recommended operating conditions	3
Table 3.1.3 Electrical characteristics (1)	4
Table 3.1.4 Electrical characteristics (2)	5
Table 3.1.5 A/D Converter characteristics (1)	5
Table 3.1.6 Timing requirements	6
Table 3.1.7 Switching characteristics	6
Table 3.3.1 Programming adapters	23
Table 3.3.2 PROM programmer address setting	24
Table 3.5.1 CNTR ₀ active edge switch bit function	45

CHAPTER 1

HARDWARE

DESCRIPTION

FEATURES

APPLICATION

PIN CONFIGURATION

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

DESCRIPTION OF IMPROVED USB

FUNCTION FOR 7534 GROUP

DIFFERENCES AMONG 32-PIN, 36-PIN

AND 42-PIN

DESCRIPTION SUPPLEMENT FOR

USE OF USB FUNCTION STABLY

DESCRIPTION

The 7534 Group is the 8-bit microcomputer based on the 740 family core technology.

The 7534 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 μs
(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 28 (36-pin type)
..... 24 (32-pin type)
..... 33 (42-pin type)
- Interrupts 14 sources, 8 vectors
- Timers 8-bit X 3

- Serial Interface
 - Serial I/O1 used only for Low Speed in USB
(based on Low-Speed USB2.0 specification)
(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(4.4 to 5.25 V at USB operation)
- Power dissipation 30 mW (standard)
- Operating temperature range -20 to 85 °C
(0 to 70 °C at USB operation)
- Built-in USB 3.3 V Regulator + transceiver based on Low-Speed USB2.0 specification

APPLICATION

Input device for personal computer peripherals

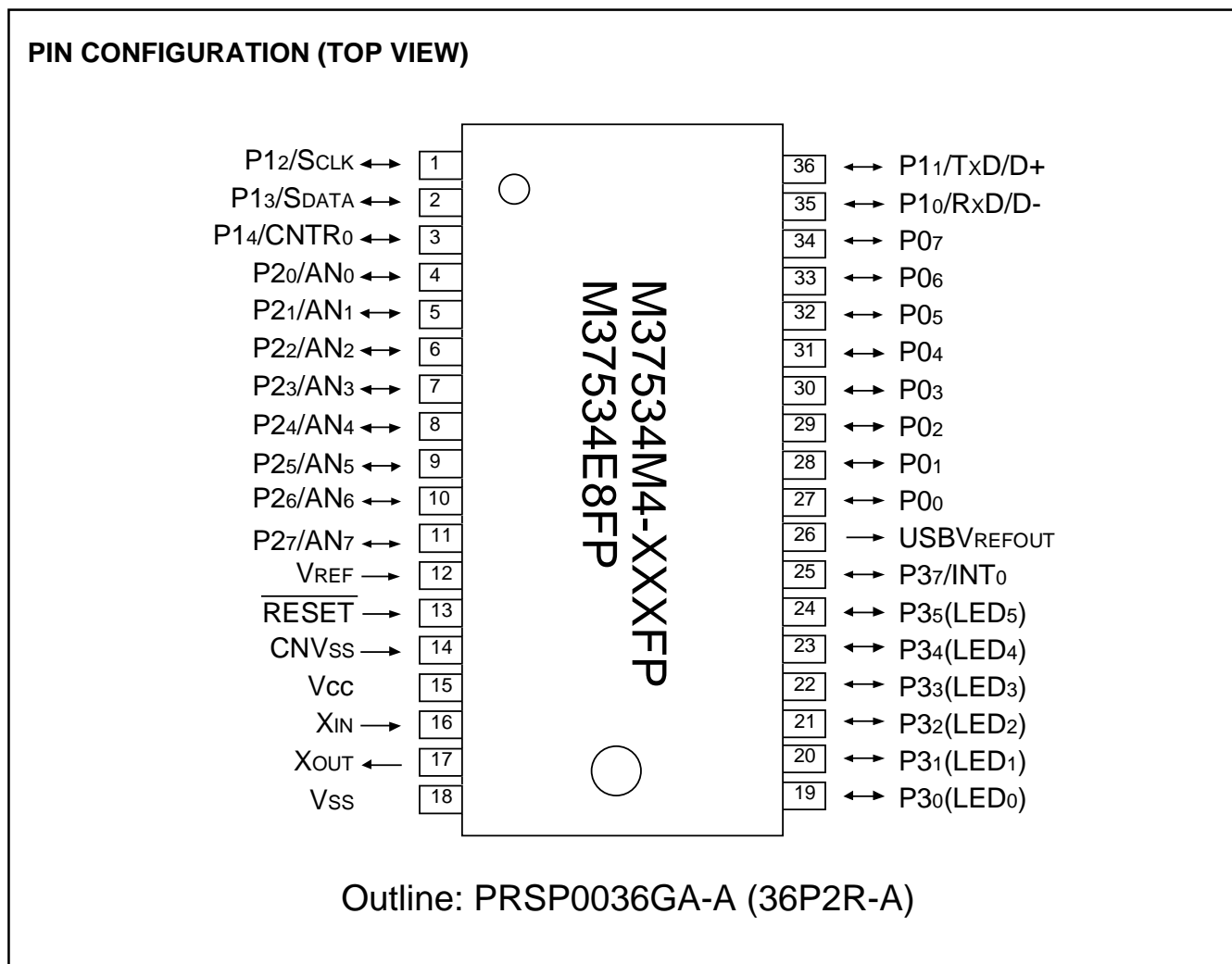


Fig. 1 Pin configuration of M37534M4-XXXXFP, M37534E8FP

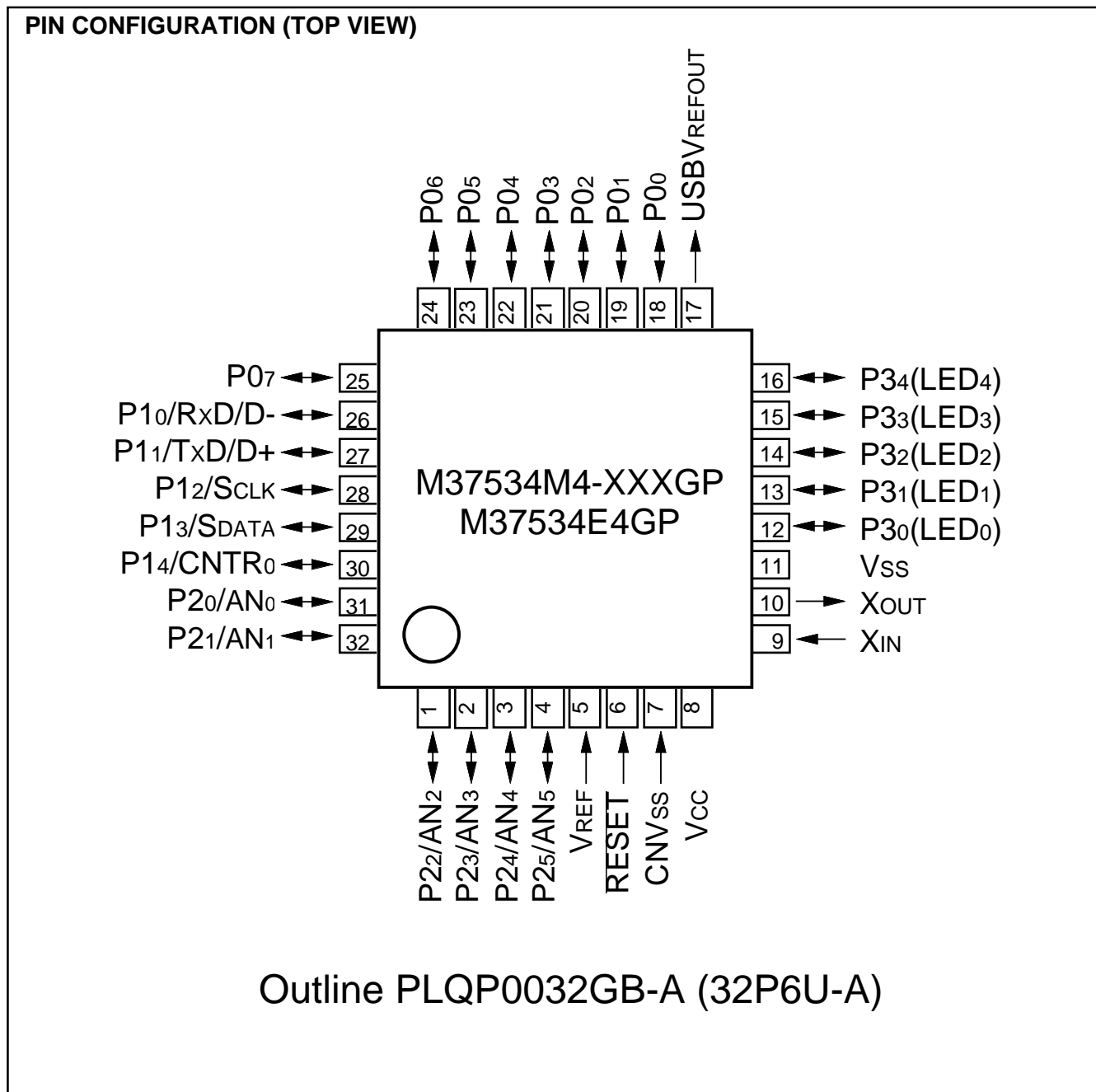


Fig. 2 Pin configuration of M37534M4-XXXGP, M37534E4GP

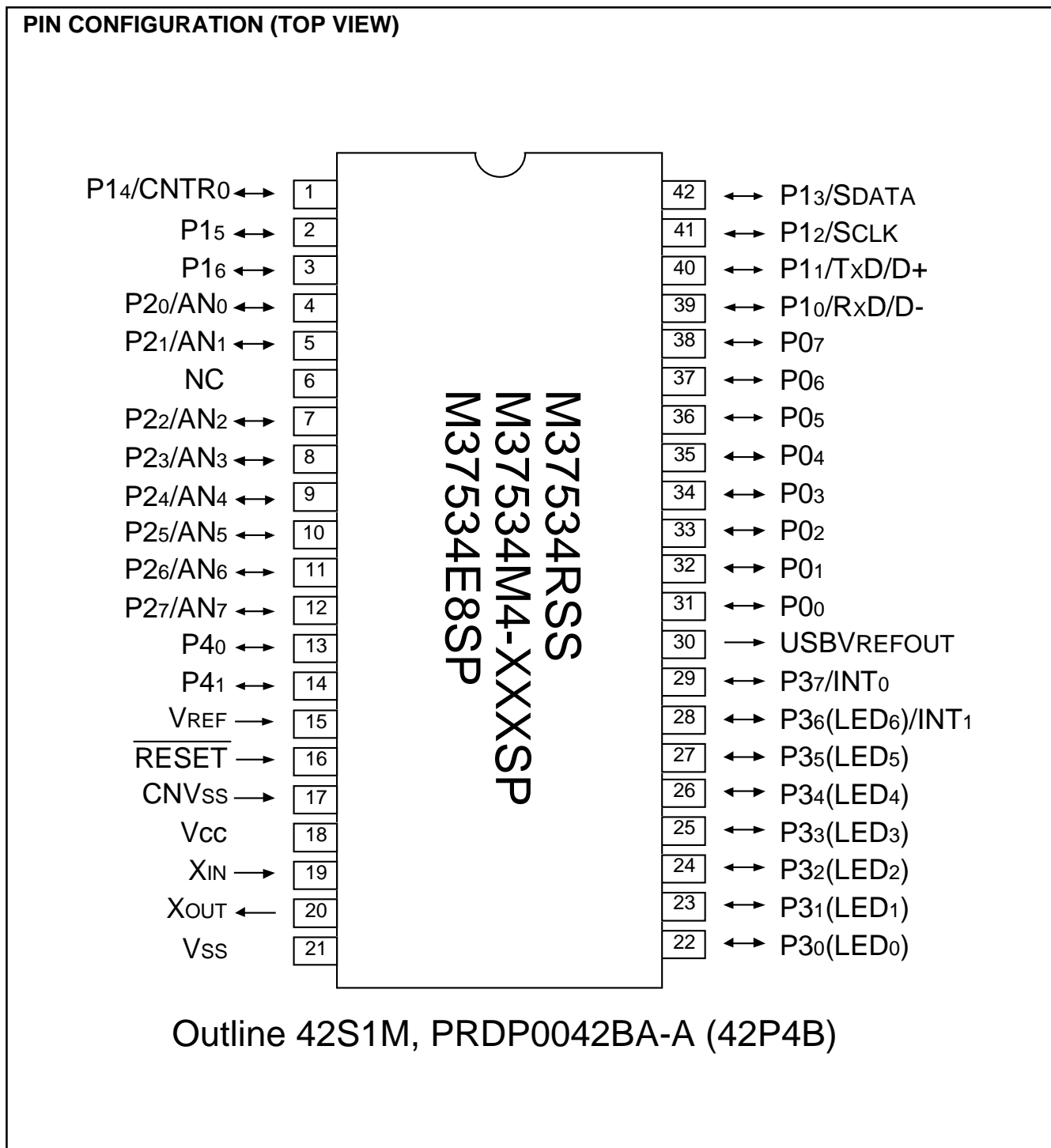


Fig. 3 Pin configuration of M37534RSS, M37534M4-XXXSP, M37534E8SP

FUNCTIONAL BLOCK

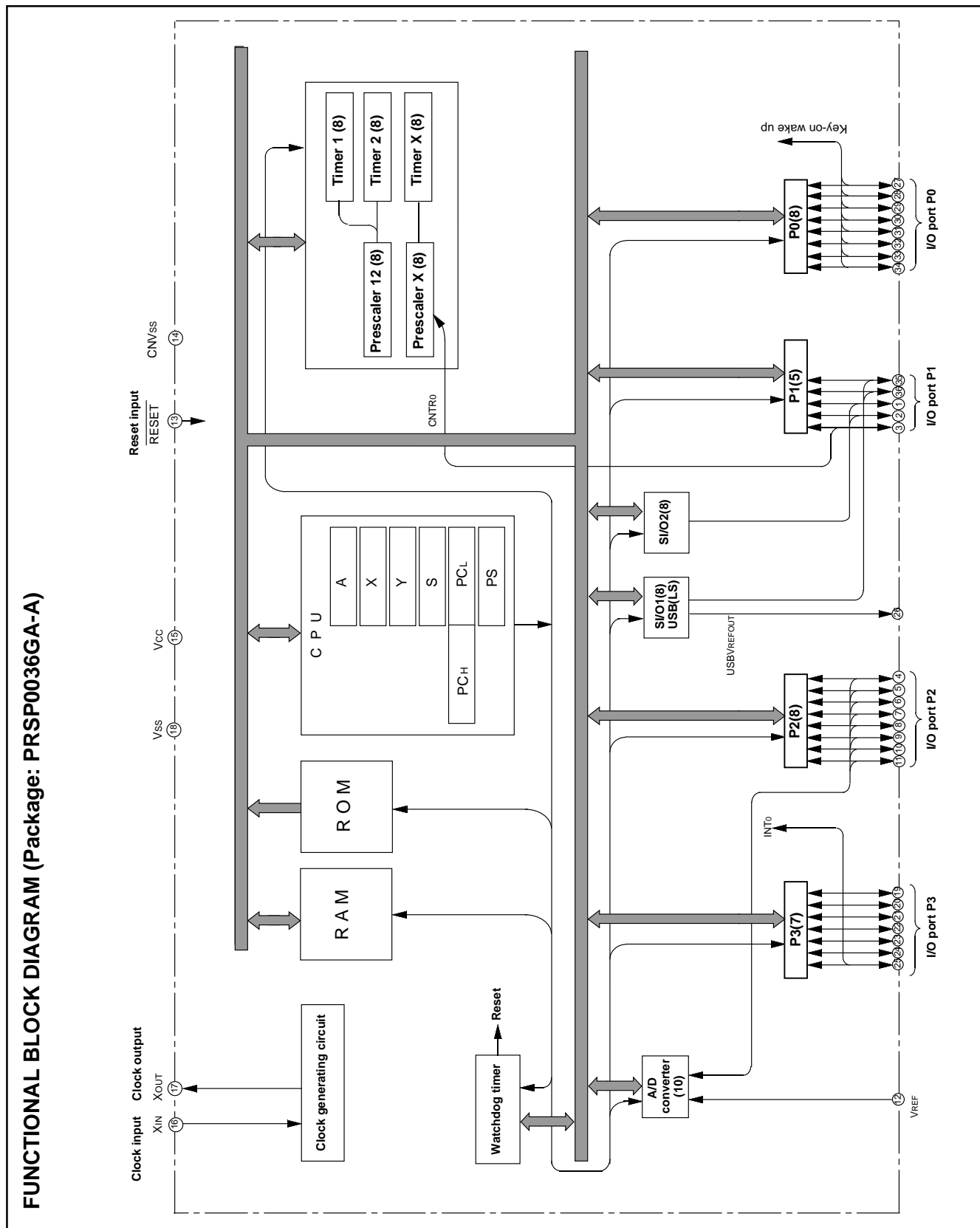
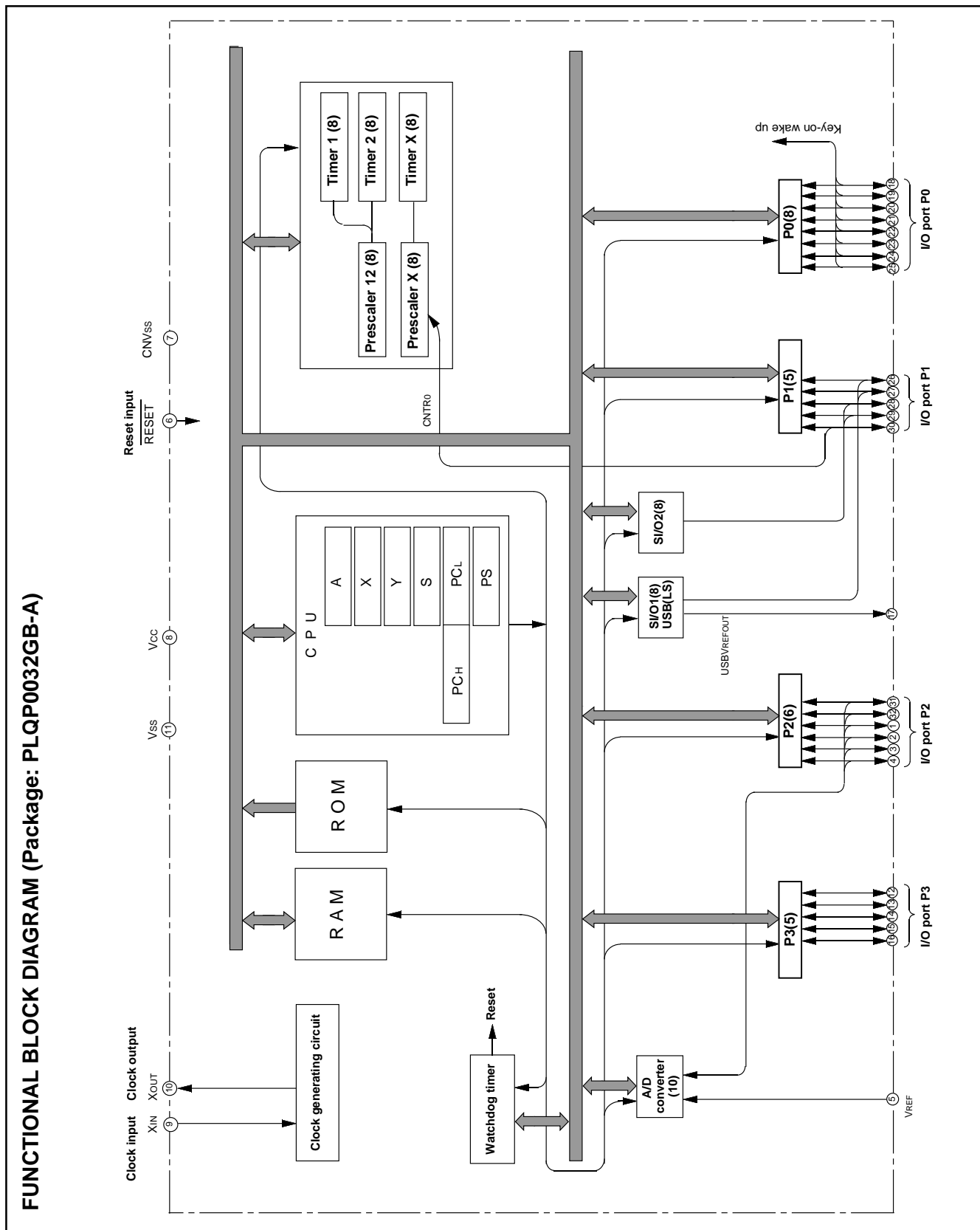


Fig. 4 Functional block diagram (PRSP0036GA-A package type)



FUNCTIONAL BLOCK DIAGRAM (Package: PLQP0032GB-A)

Fig. 5 Functional block diagram (PLQP0032GB-A package type)

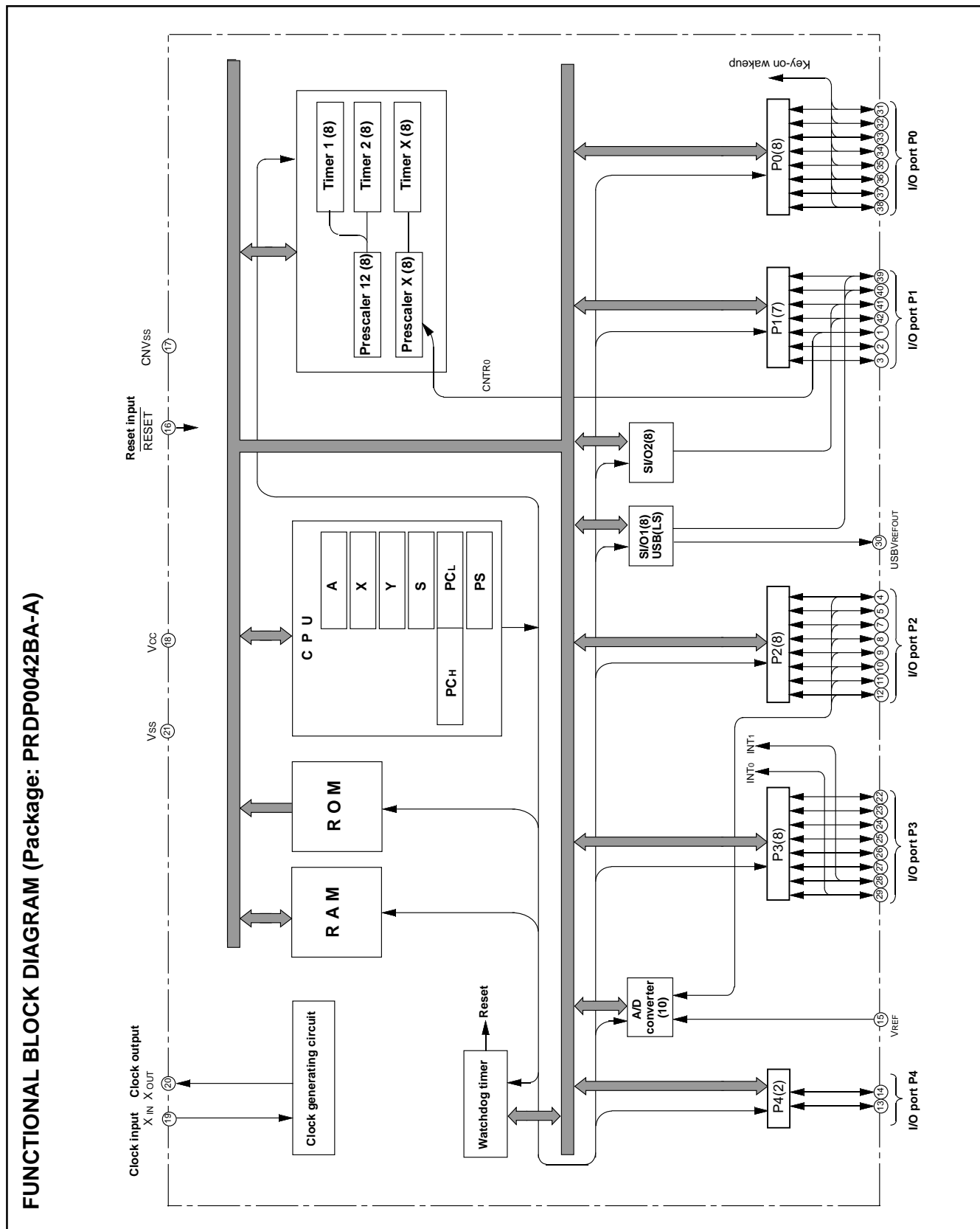


Fig. 6 Functional block diagram (PRDP0042BA-A package type)

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 (4.4 to 5.25 V at USB operating) 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 3-state output structure at CMOS compatible input level •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 3-state output structure at CMOS compatible input level •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			•Serial I/O2 function pin
P13/SDATA			•Timer X function pin
P14/CNTR0			
P15, P16			
P20/AN0– P27/AN7	I/O port P2	•8-bit I/O port having almost the same function as P0 •CMOS 3-state output structure at CMOS compatible input level	•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 3-state output structure at CMOS compatible input level (CMOS/TTL level can be switched for P36, P37). •P30 to P36 can output a large current for driving LED. •Whether a built-in pull-up resistor is to be used or not can be determined by program.	
P36/INT1 P37/INT0			•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.	

GROUP EXPANSION

Renesas plans to expand the 7534 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

PRSP0036GA-A 0.8 mm-pitch plastic molded SOP
 PLQP0032GB-A 0.8 mm-pitch plastic molded LQFP
 PRDP0042BA-A 42 pin plastic molded SDIP
 42SIM 42 pin shrink ceramic PIGGY BACK

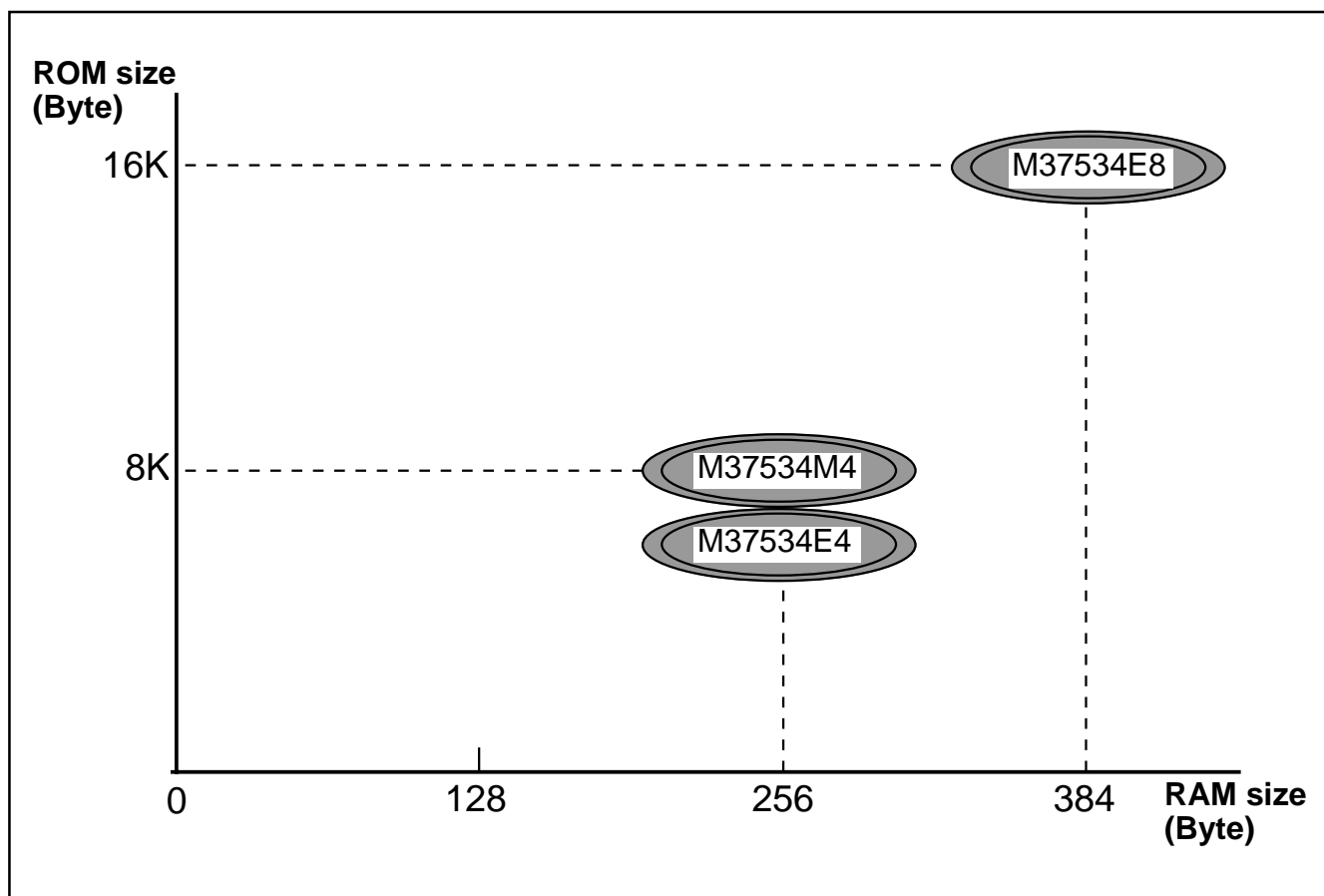


Fig. 7 Memory expansion plan

Currently supported products are listed below.

Table 2 List of supported products

Part number	(P) ROM size (bytes) ROM size for User ()	RAM size (bytes)	Package	Remarks
M37534M4-XXXFP	8192 (8062)	256	PRSP0036GA-A	Mask ROM version
M37534M4-XXXGP	8192 (8062)	256	PLQP0032GB-A	Mask ROM version
M37534M4-XXXSP	8192 (8062)	256	PRDP0042BA-A	Mask ROM version
M37534E4GP	8192 (8062)	256	PLQP0032GB-A	One Time PROM version (blank)
M37534E8FP	16384 (16254)	384	PRSP0036GA-A	One Time PROM version (blank)
M37534E8SP	16384 (16254)	384	PRDP0042BA-A	One Time PROM version (blank)
M37534RSS	—	384	42S1M	Emulator MCU

FUNCTIONAL DESCRIPTION Central Processing Unit (CPU)

The 7534 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 9.

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.

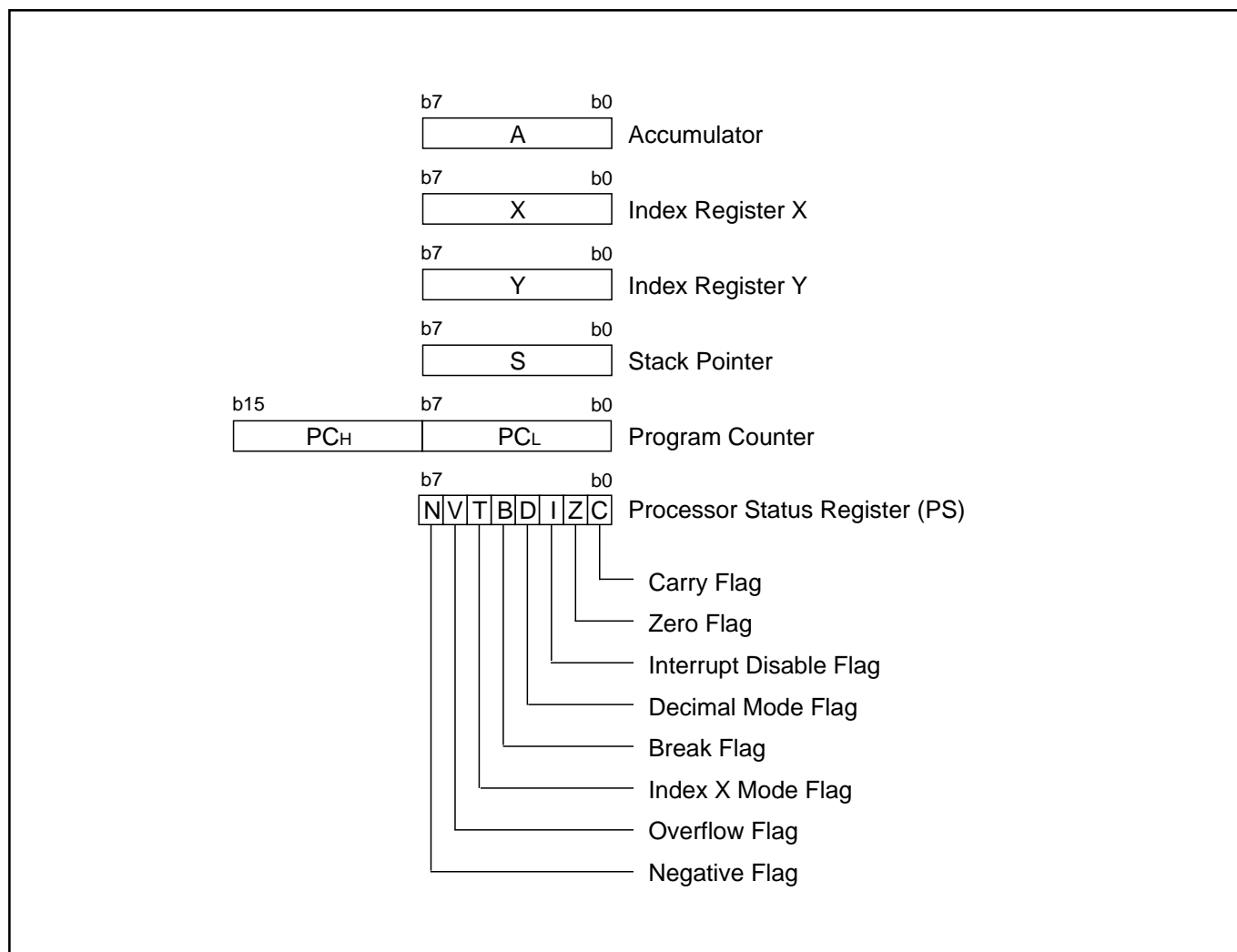


Fig. 8 740 Family CPU register structure

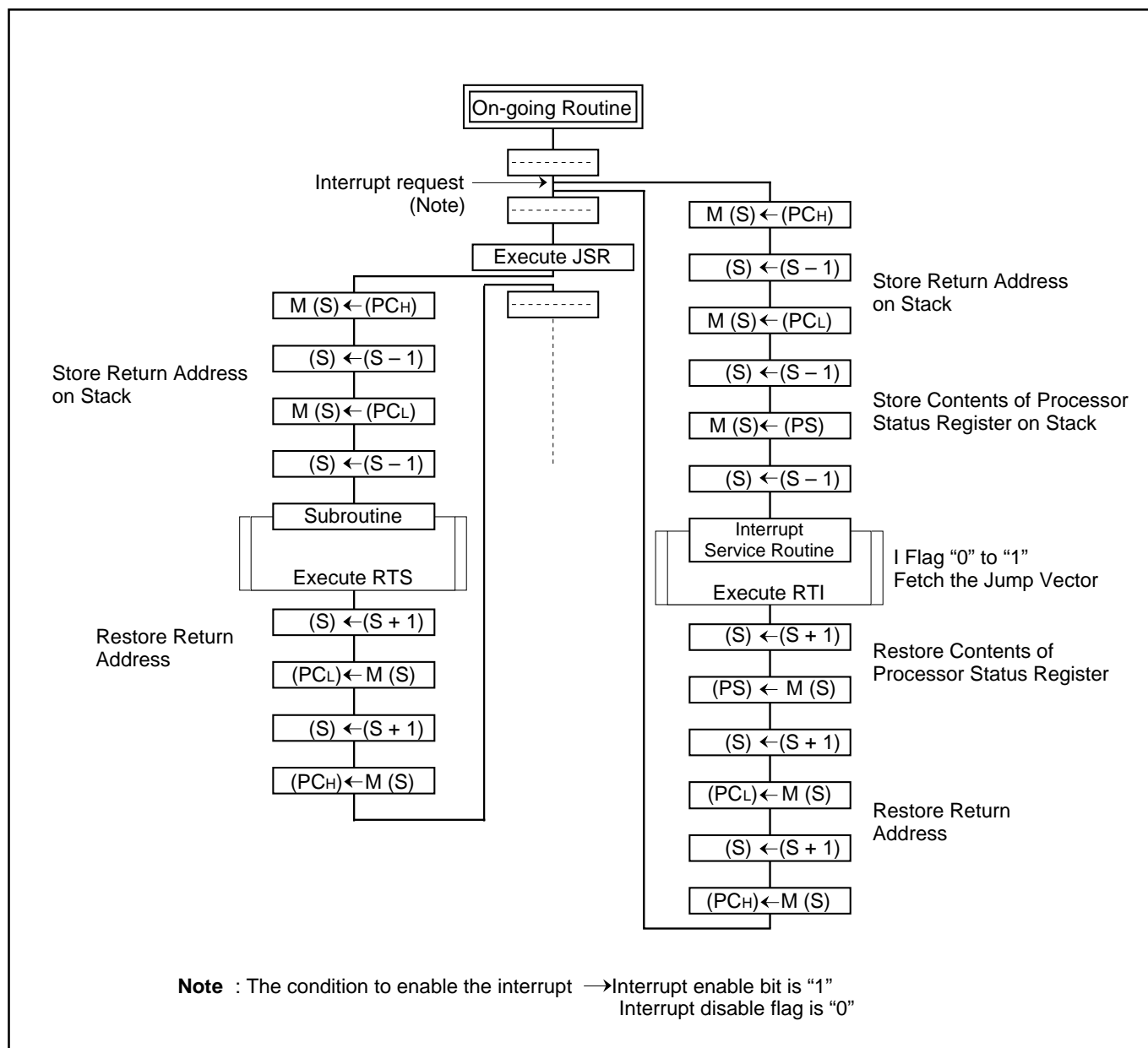


Fig. 9 Register push and pop at interrupt generation and subroutine call

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	–

[CPU Mode Register] CPUM

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

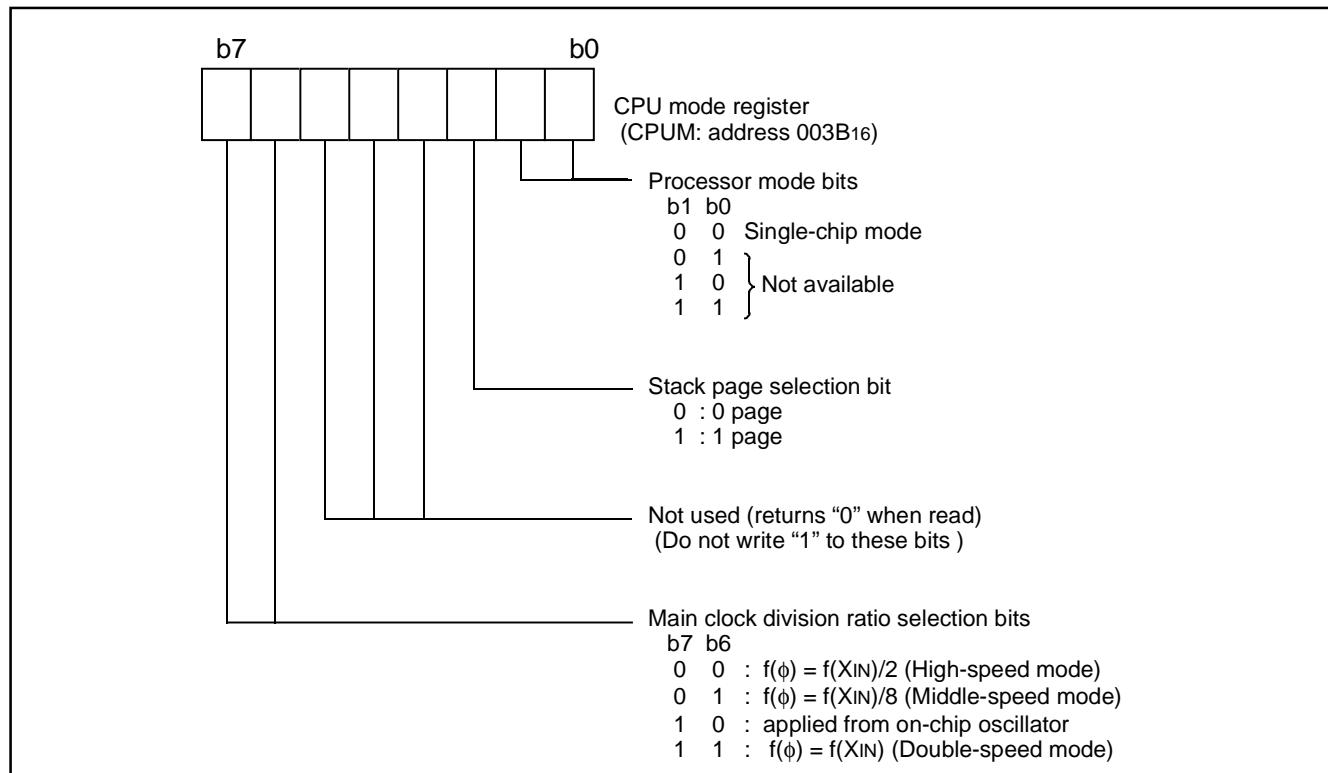


Fig. 10 Structure of CPU mode register

Switching method of CPU mode register

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

Note on stack page

When 1 page is used as stack area by the stack page selection bit, the area which can be used as stack depends on RAM size. Especially, be careful that the RAM area varies in Mask ROM version, One Time PROM version and Emulator MCU.

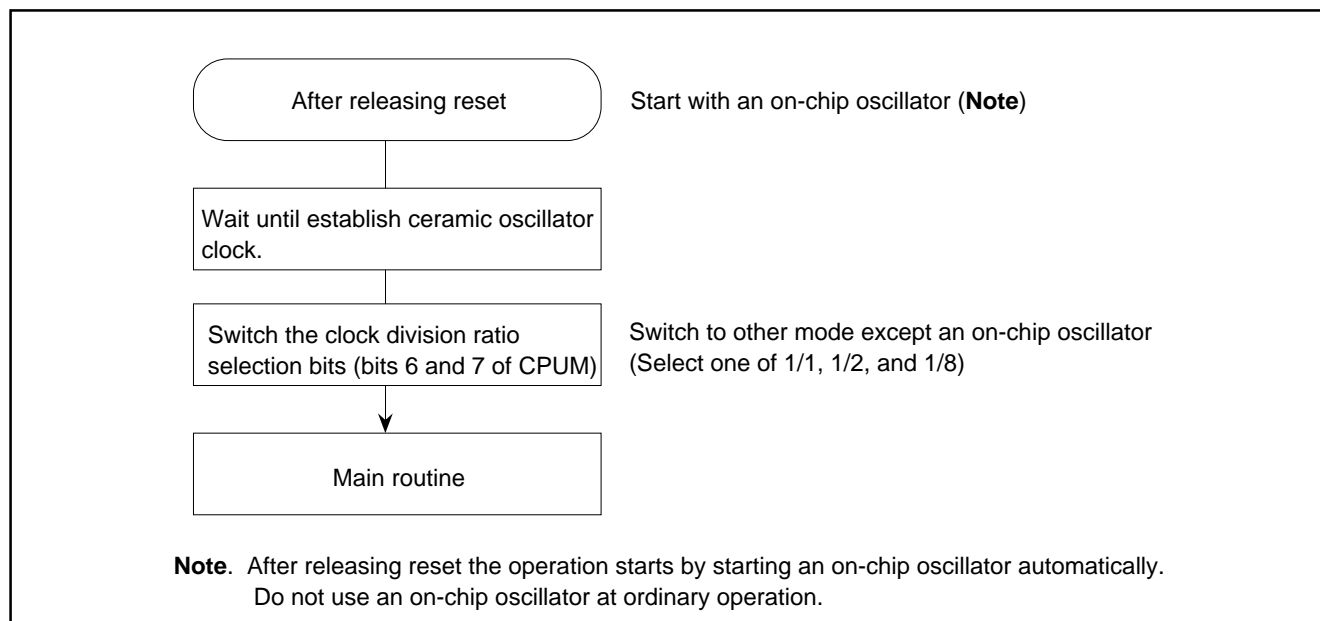


Fig. 11 Switching method of CPU mode register

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.

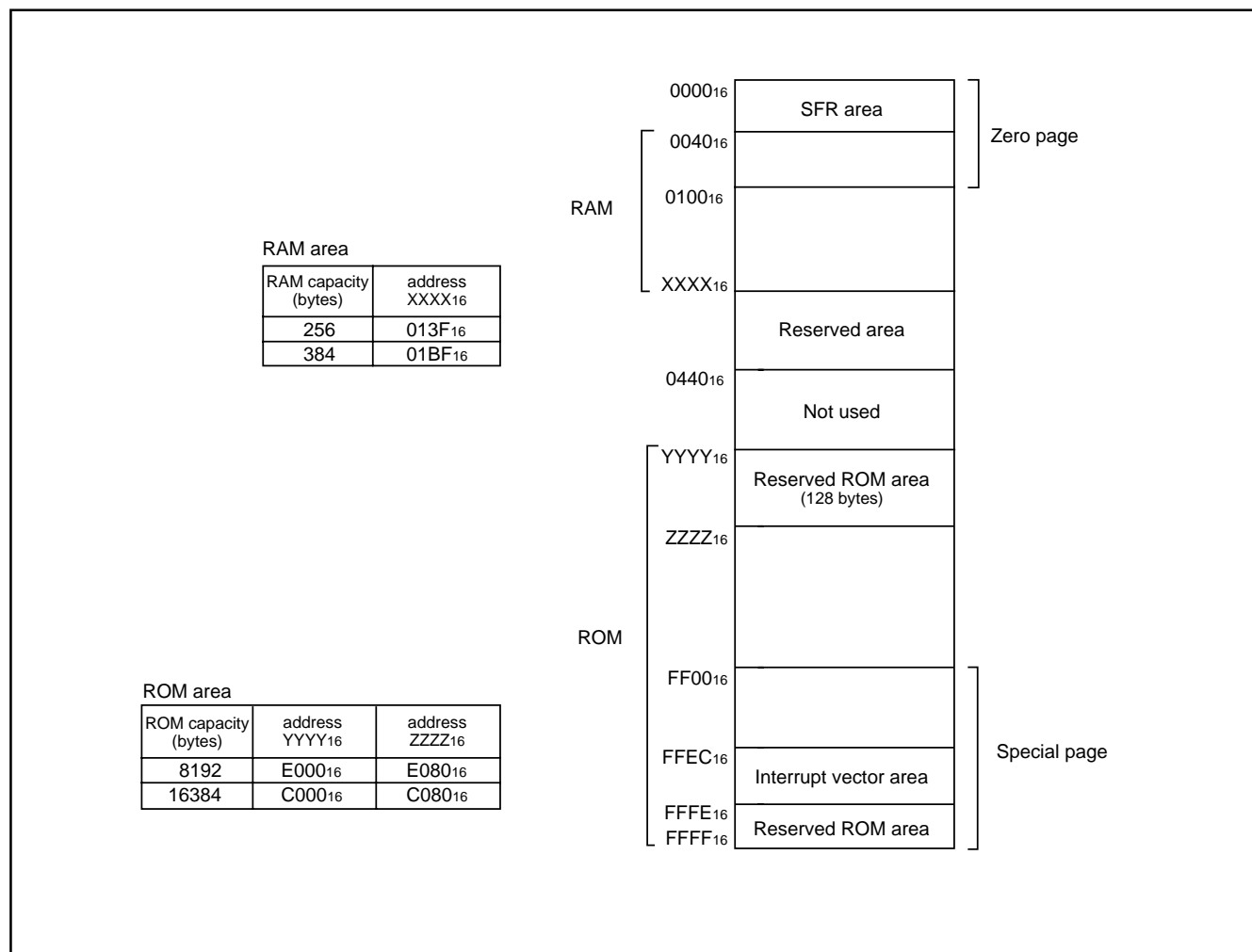


Fig. 12 Memory map diagram

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

Fig. 13 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 individual 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 0016₁₆), 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 0017₁₆), a CMOS input level or a TTL input level can be selected for ports P1₀, P1₂, P1₃, P3₆ and P3₇ by program.

Then, as for the 36-pin version, set "1" to each bit 6 of the port P3 direction register and port P3 register.

As for the 32-pin version, set "1" to respective bits 5, 6, 7 of the port P3 direction register and port P3 register.

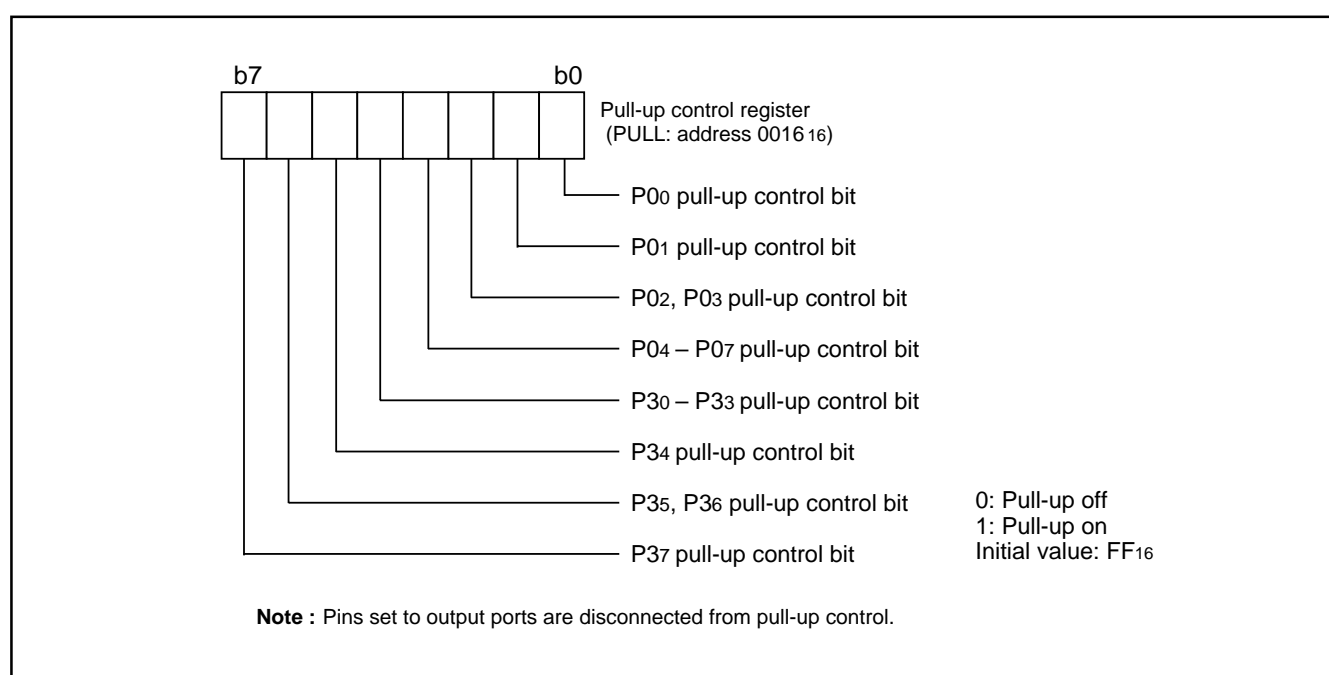


Fig. 14 Structure of pull-up control register

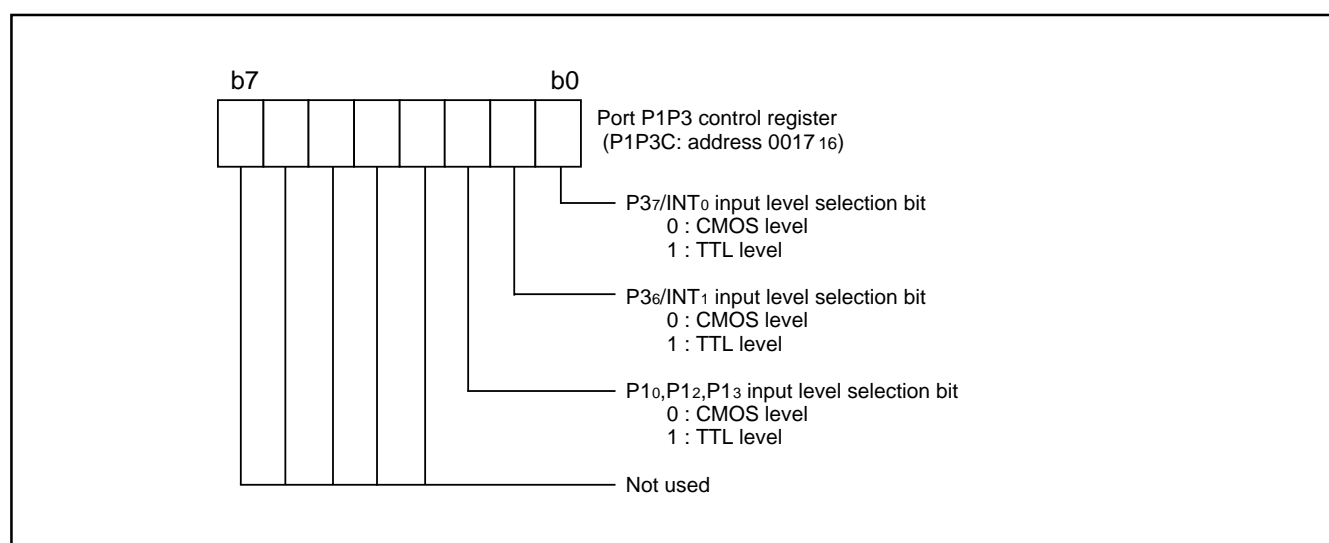


Fig. 15 Structure of port P1P3 control register

Table 5 I/O port function table

Pin	Name	Input/output	I/O format	Non-port function	Related SFRs	Diagram No.
P0 ₀ –P0 ₇	Port P0	I/O individual bits	•CMOS compatible input level •CMOS 3-state output	Key input interrupt	Pull-up control register	(1)
P1 ₀ /RxD/D- P1 ₁ /TxD/D+	Port P1		•USB input/output level when selecting USB function	Serial I/O1 function input/output	Serial I/O1 control register	(2) (3)
P1 ₂ /SCLK P1 ₃ /SDATA P1 ₄ /CNTR ₀ P1 ₅ , P1 ₆			•CMOS compatible input level •CMOS 3-state output (Note)	Serial I/O2 function input/output	Serial I/O2 control register	(4) (5)
				Timer X function input/output	Timer X mode register	(6)
						(10)
P2 ₀ /AN ₀ – P2 ₇ /AN ₇			Port P2		A/D conversion input	A/D control register
P3 ₀ –P3 ₅ P3 ₆ /INT ₁ P3 ₇ /INT ₀	Port P3					(8)
			External interrupt input	Interrupt edge selection register	(9)	
P4 ₀ , P4 ₁	Port P4					(10)

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

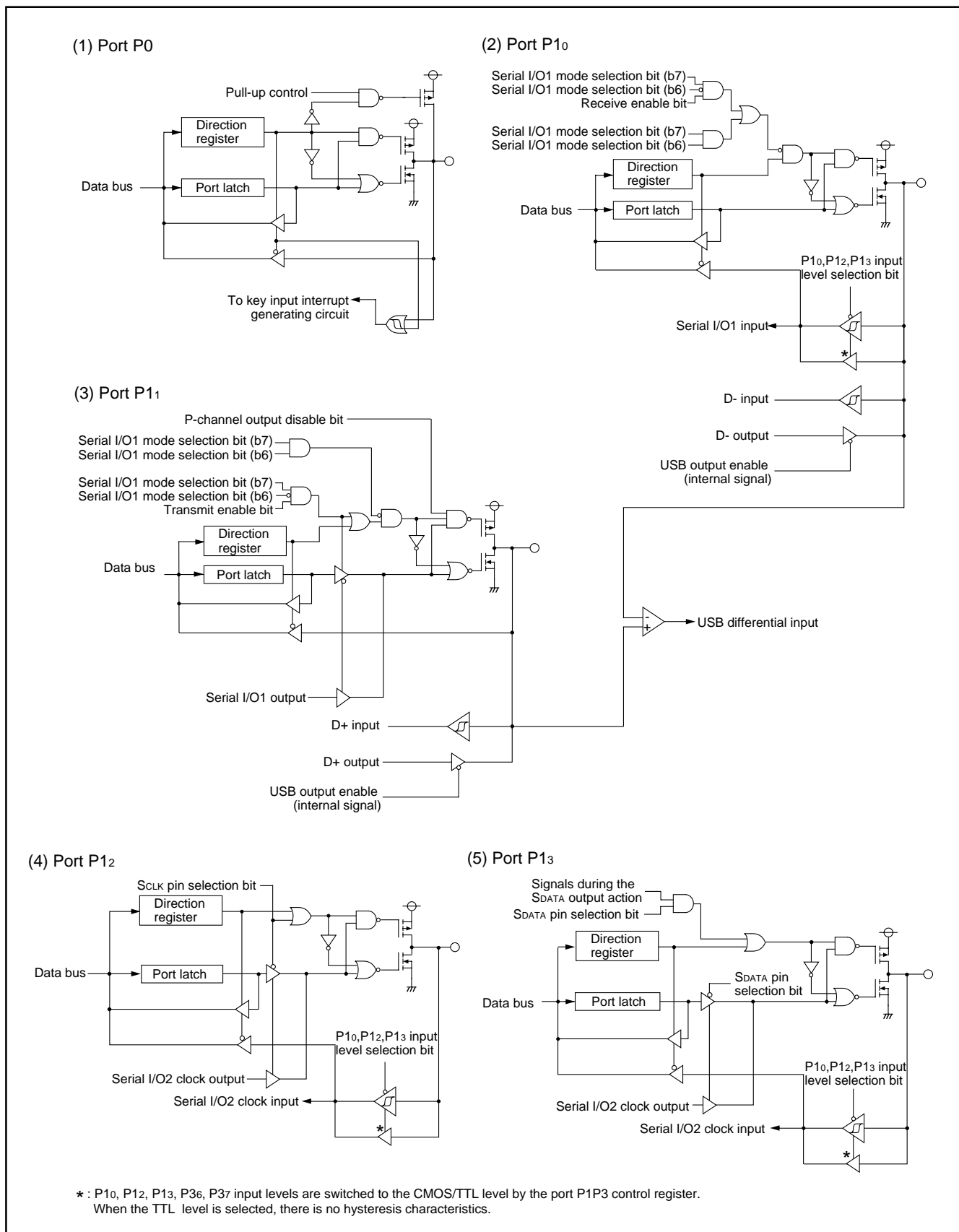


Fig. 16 Block diagram of ports (1)

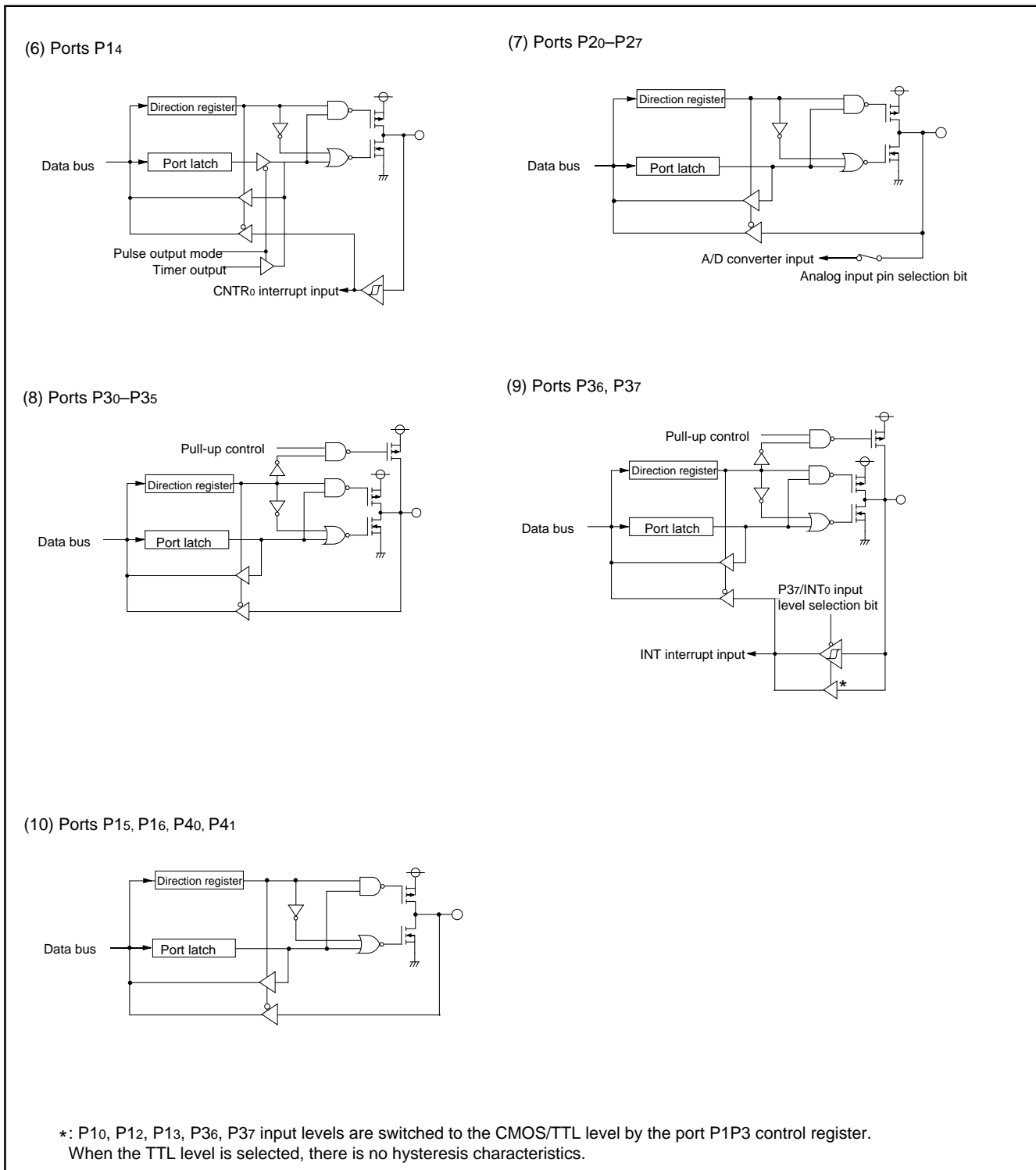


Fig. 17 Block diagram of ports (2)

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 ₁₆	FFFC ₁₆	At reset input	Non-maskable
UART receive	2	FFFB ₁₆	FFFA ₁₆	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 ₁₆	FFF8 ₁₆	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 ₁				At detection of either rising or falling edge of INT ₁ input	External interrupt (active edge selectable)
INT ₀	4	FFF7 ₁₆	FFF6 ₁₆	At detection of either rising or falling edge of INT ₀ input	External interrupt (active edge selectable)
Timer X	5	FFF5 ₁₆	FFF4 ₁₆	At timer X underflow	
Key-on wake-up				At falling of conjunction of input logical level for port P0 (at input)	External interrupt (valid at falling)
Timer 1	6	FFF3 ₁₆	FFF2 ₁₆	At timer 1 underflow	STP release timer underflow
Timer 2	7	FFF1 ₁₆	FFF0 ₁₆	At timer 2 underflow	
Serial I/O ₂				At completion of transmit/receive shift	
CNTR ₀	8	FFEF ₁₆	FFEE ₁₆	At detection of either rising or falling edge of CNTR ₀ input	External interrupt (active edge selectable)
A/D conversion				At completion of A/D conversion	
BRK instruction	9	FFED ₁₆	FFEC ₁₆	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.

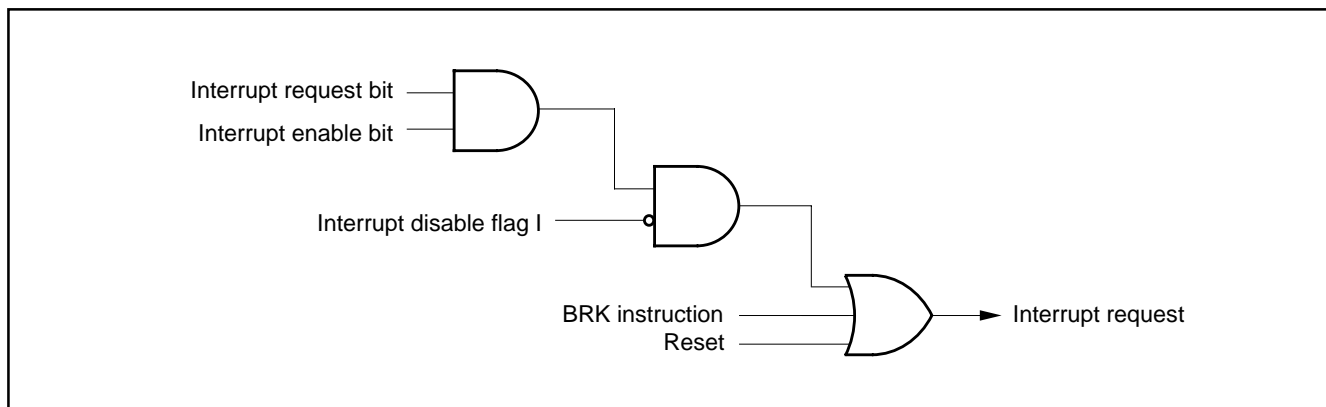


Fig. 18 Interrupt control

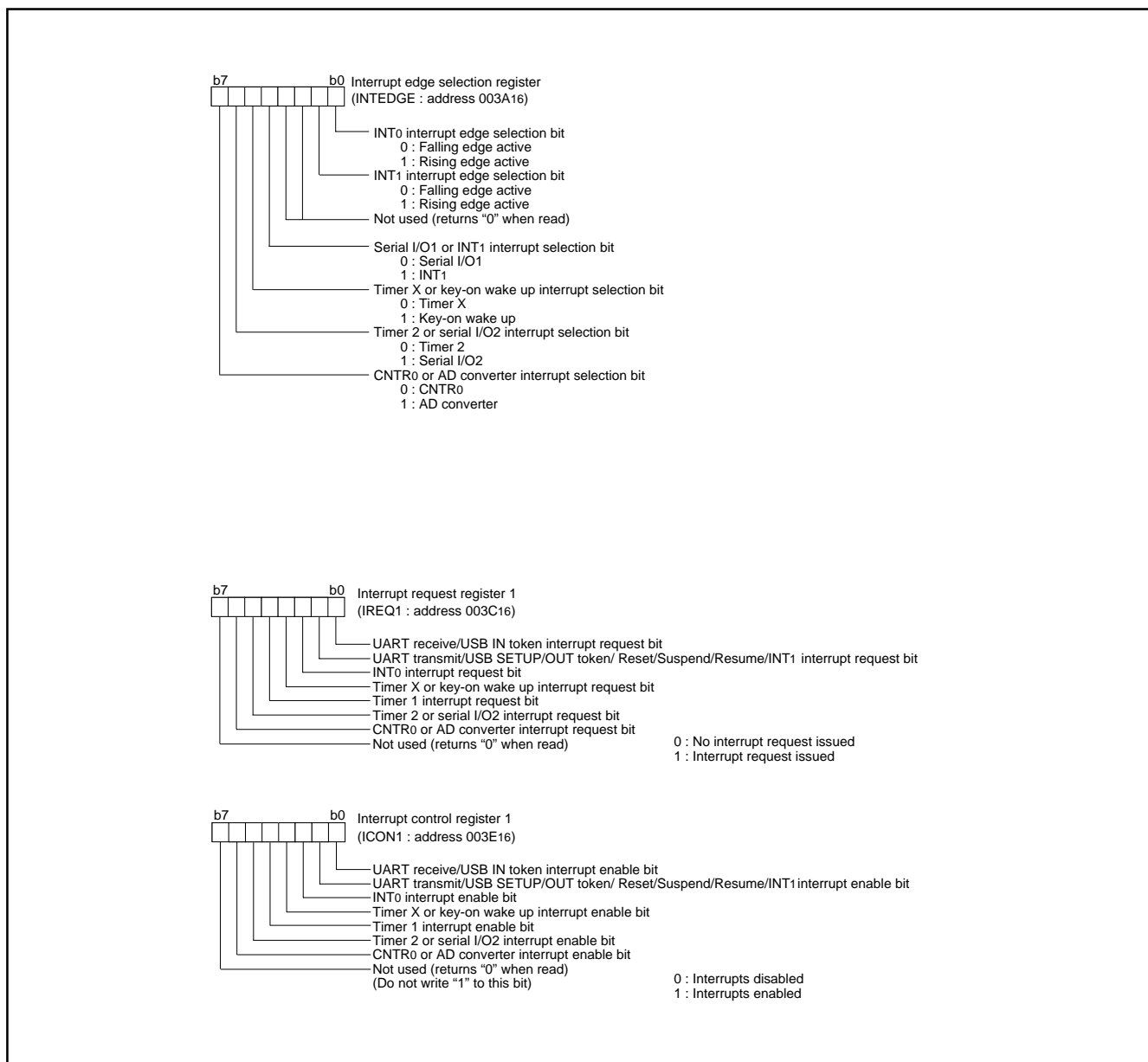


Fig. 19 Structure of Interrupt-related registers

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 20, 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.

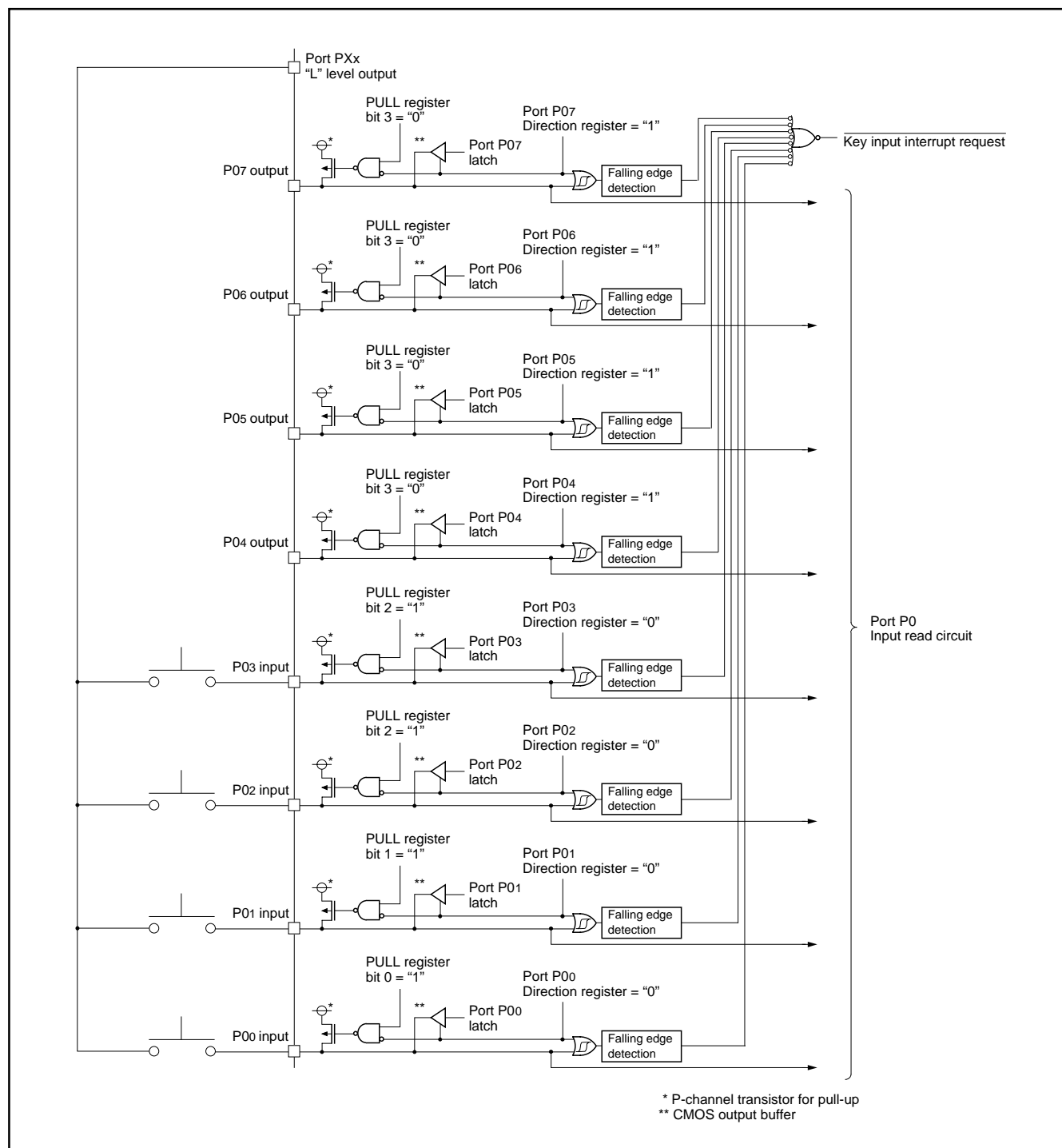


Fig. 20 Connection example when using key input interrupt and port P0 block diagram

Timers

The 7534 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 CNTR₀ pin.

When the CNTR₀ active edge switch bit is "0", the output of the CNTR₀ 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 CNTR₀ pin.

When the CNTR₀ active edge switch bit is "0", the timer counts the rising edge of the CNTR₀ pin. When this bit is "1", the timer counts the falling edge of the CNTR₀ pin.

• Pulse Width Measurement Mode

When the CNTR₀ active edge switch bit is "0", the timer counts the signal selected by the timer X count source selection bit while the CNTR₀ pin is "H". When this bit is "1", the timer counts the signal while the CNTR₀ 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.

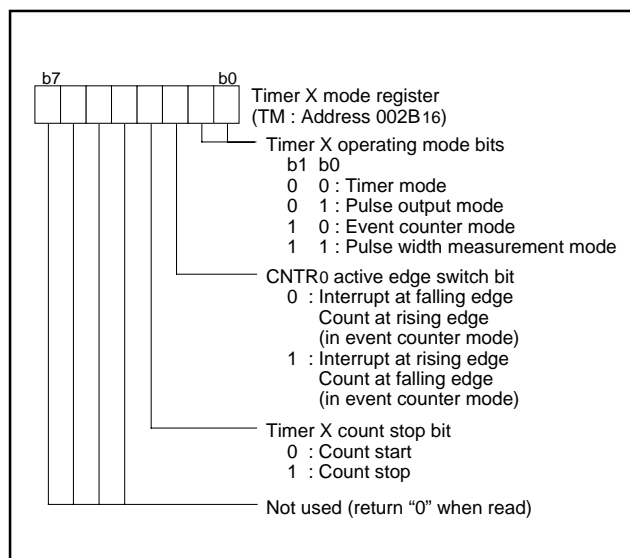


Fig. 21 Structure of timer X mode register

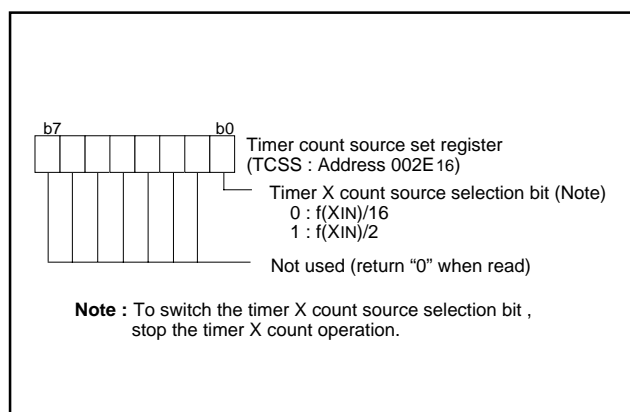


Fig. 22 Timer count source set register

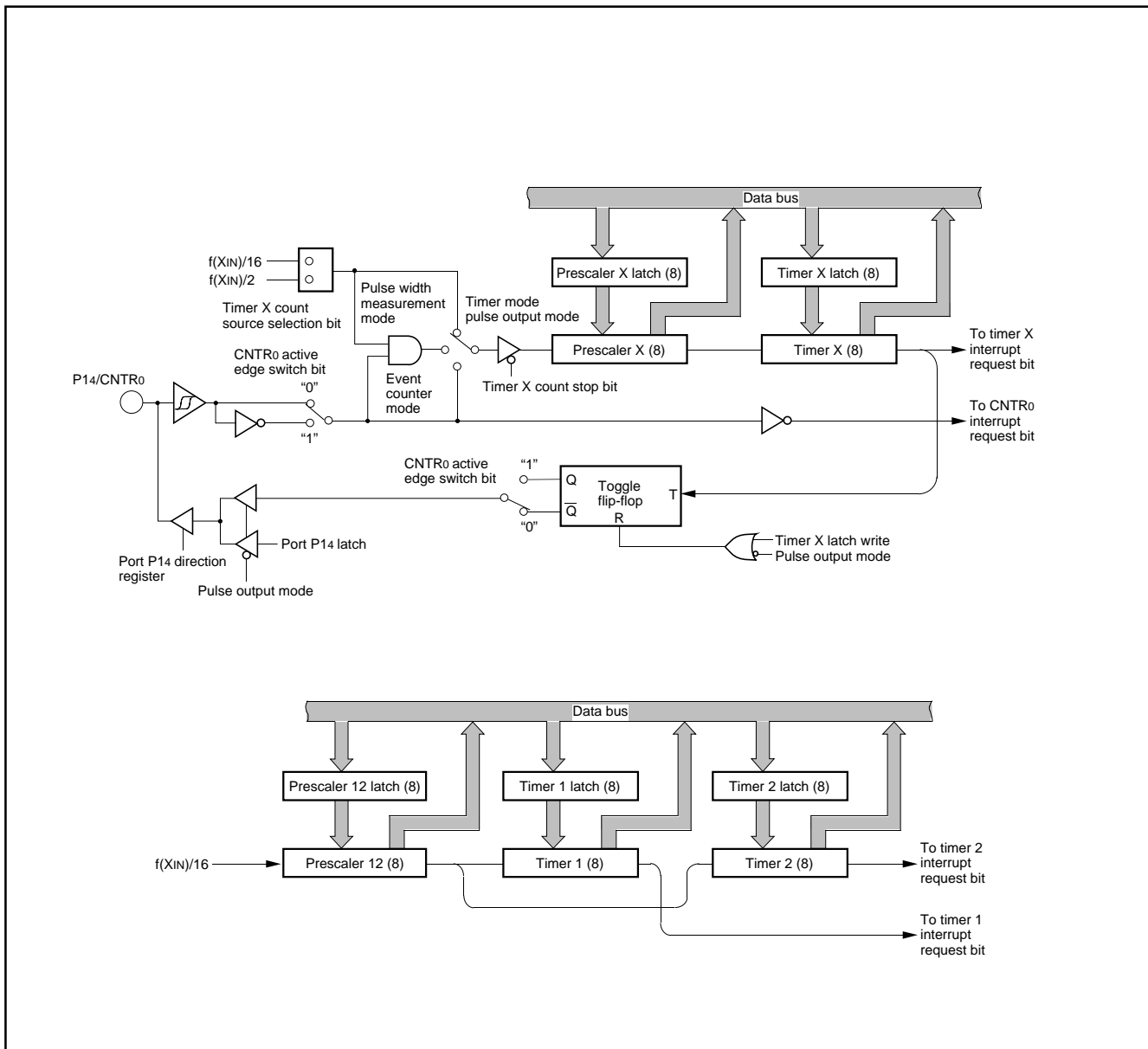


Fig. 23 Block diagram of timer X, timer 1 and timer 2

Serial Interface

●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.

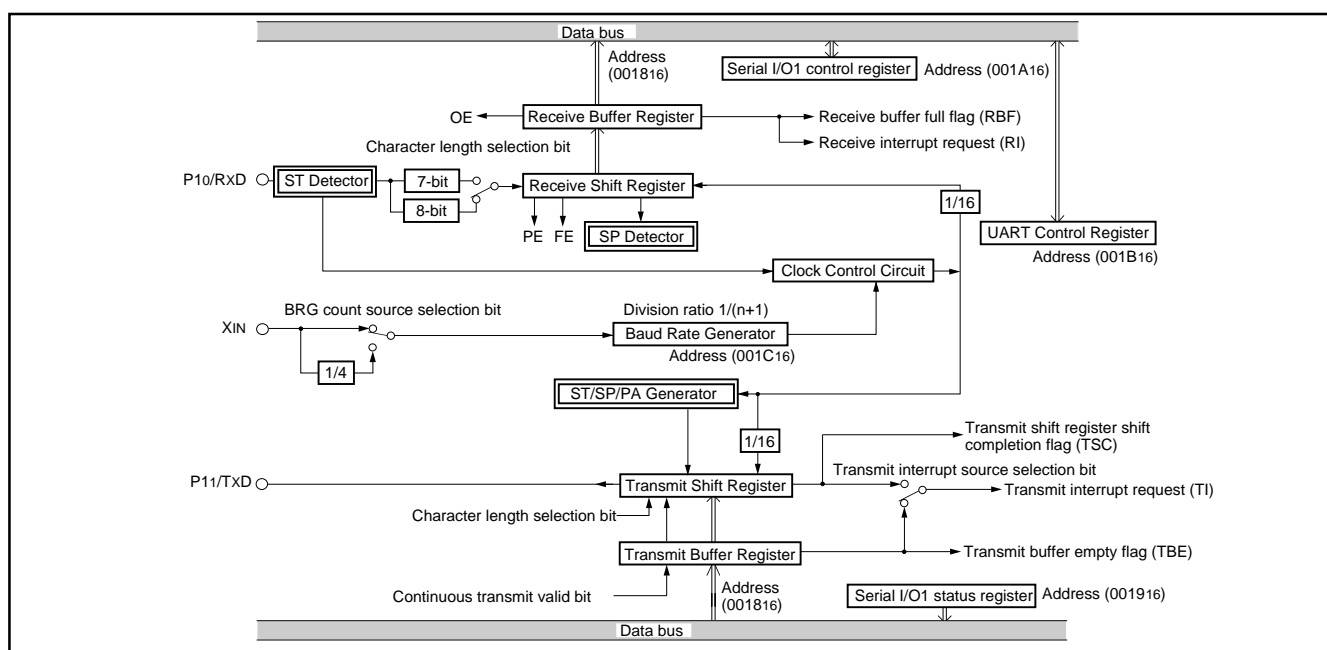


Fig. 24 Block diagram of UART serial I/O1

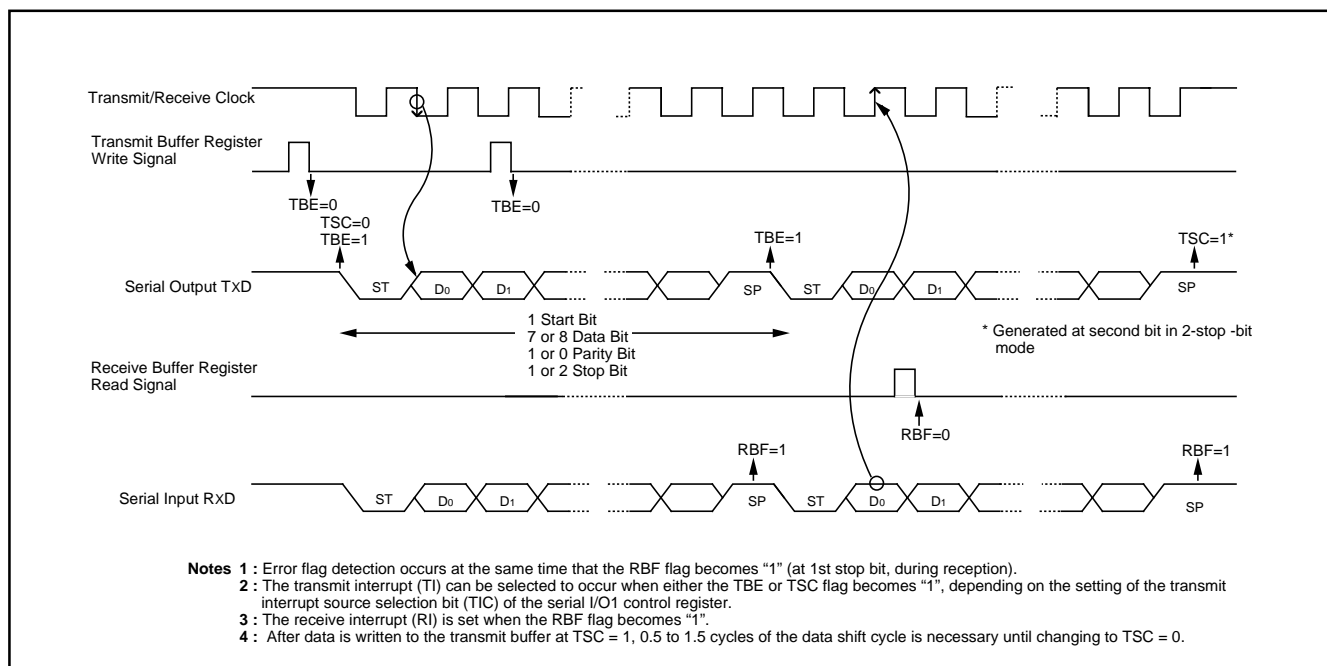


Fig. 25 Operation of UART serial I/O1 function

[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 an data transfer. One bit in this register (bit 4) is always valid and sets the output structure of the P11/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 UART 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 "8116" 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.

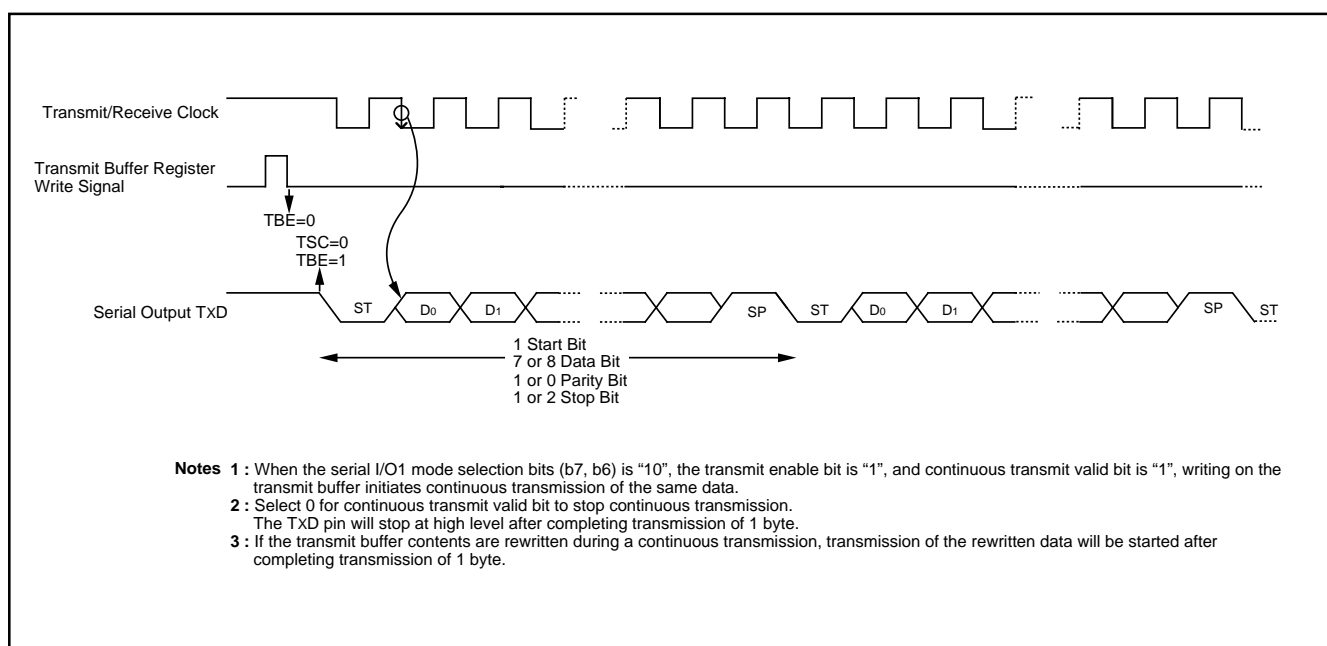


Fig. 26 Continuous transmission operation of UART serial I/O

• **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 USB2.0 specification. In this mode serial I/O1 interrupt have 6 sources; USB in and out token receive, set-up token receive, USB reset, suspend, and resume. The USB status/UART status

register 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 27 and 28.

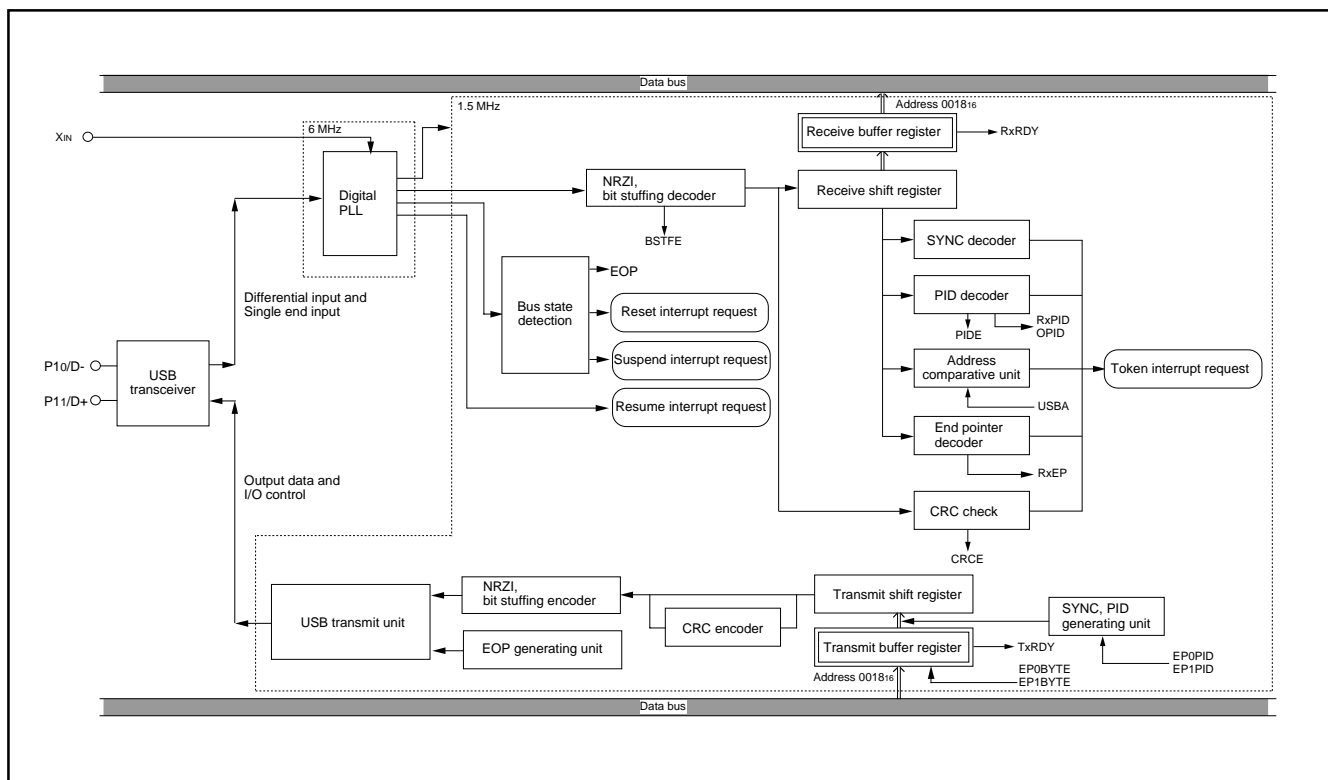


Fig. 27 USB mode block diagram

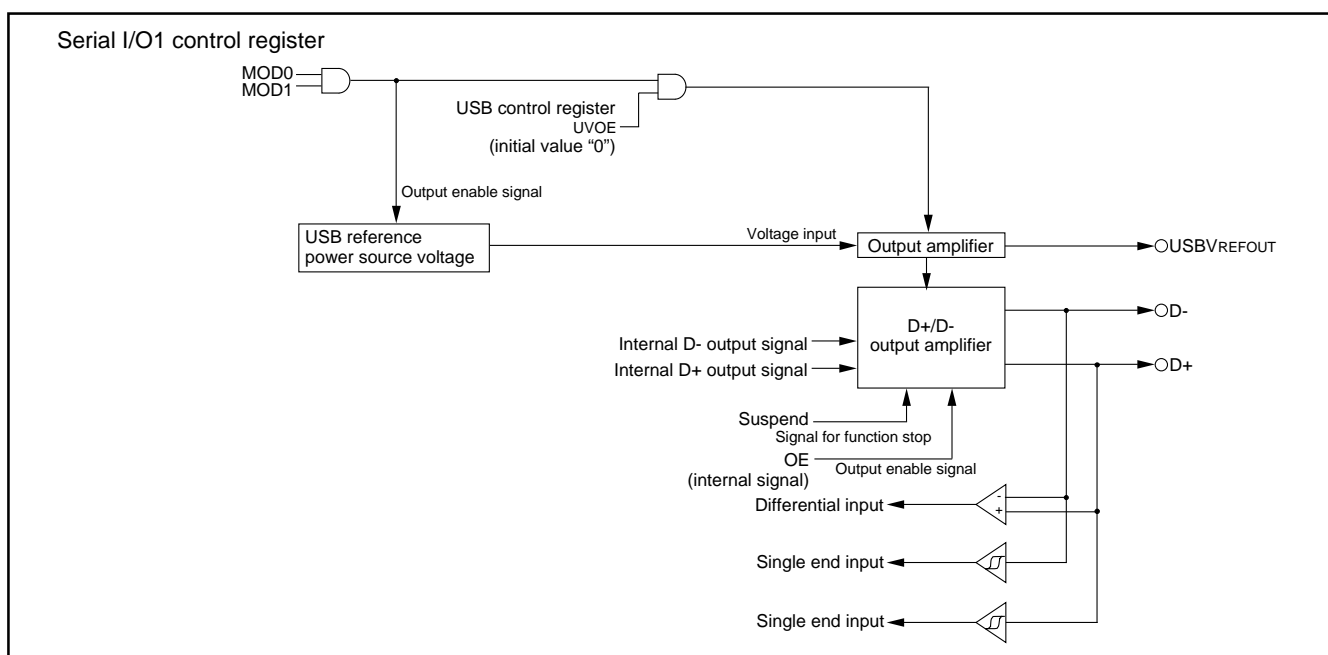


Fig. 28 USB transceiver block diagram

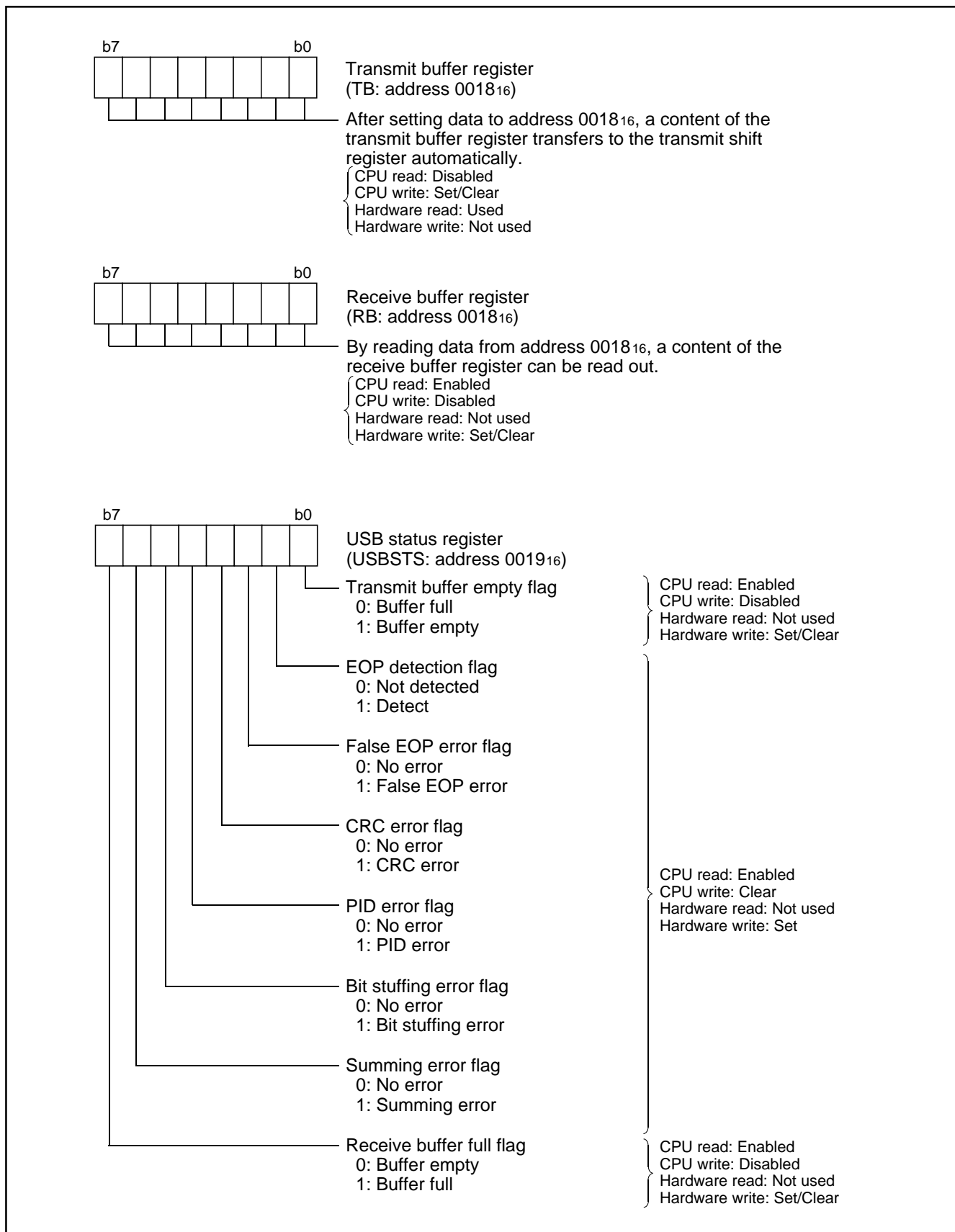


Fig. 29 Structure of serial I/O1-related registers (1)

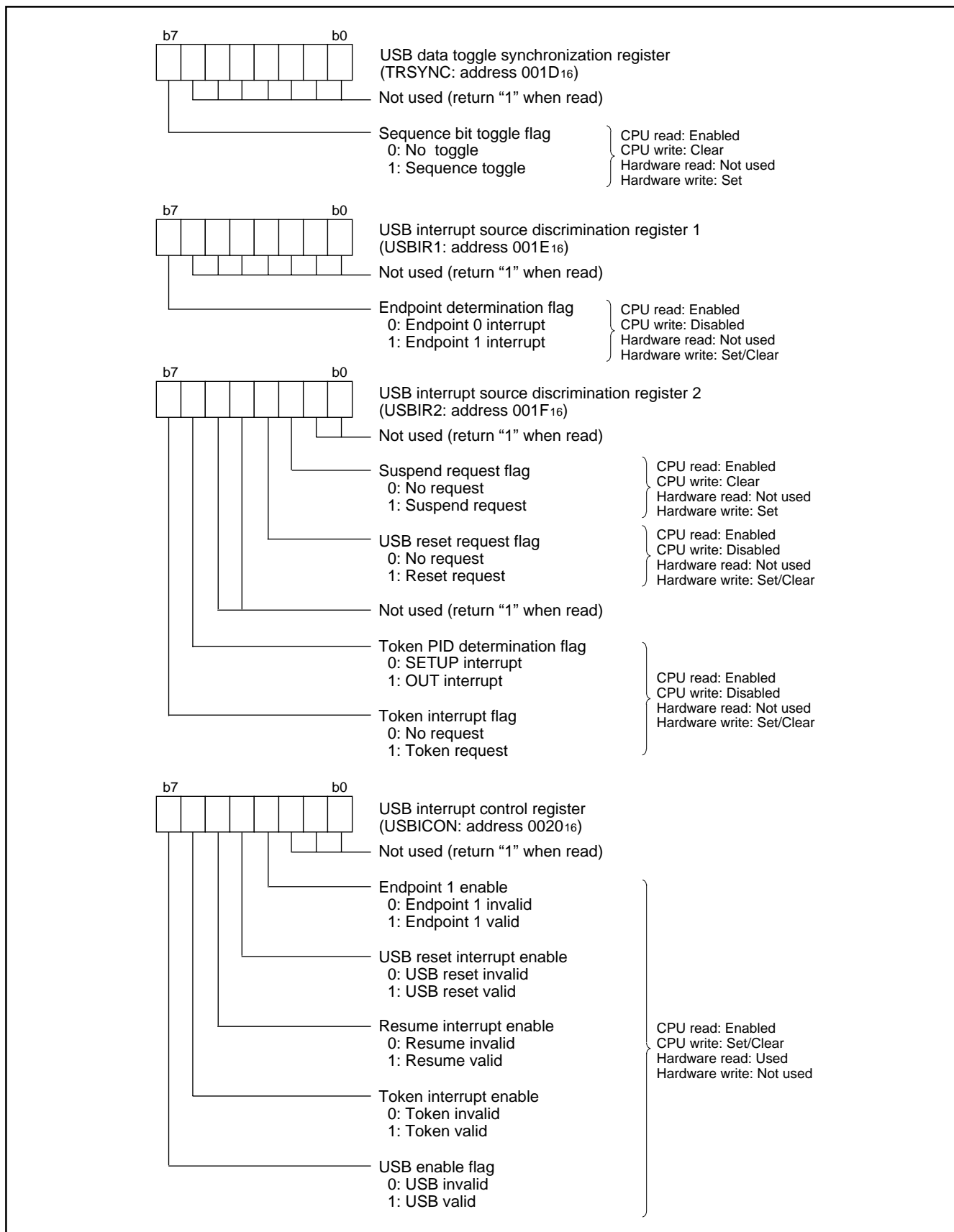


Fig. 30 Structure of serial I/O1-related registers (2)

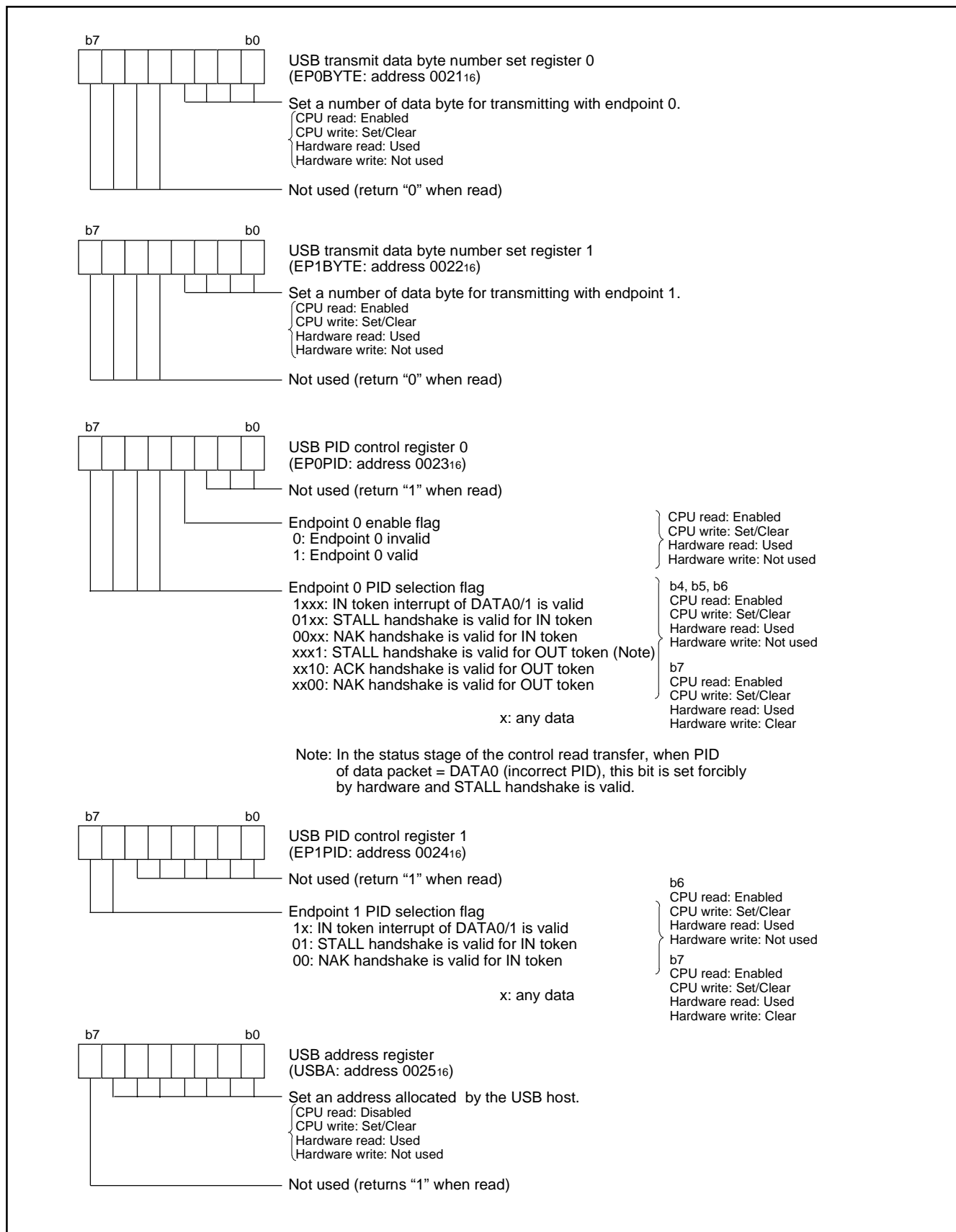


Fig. 31 Structure of serial I/O1-related registers (3)

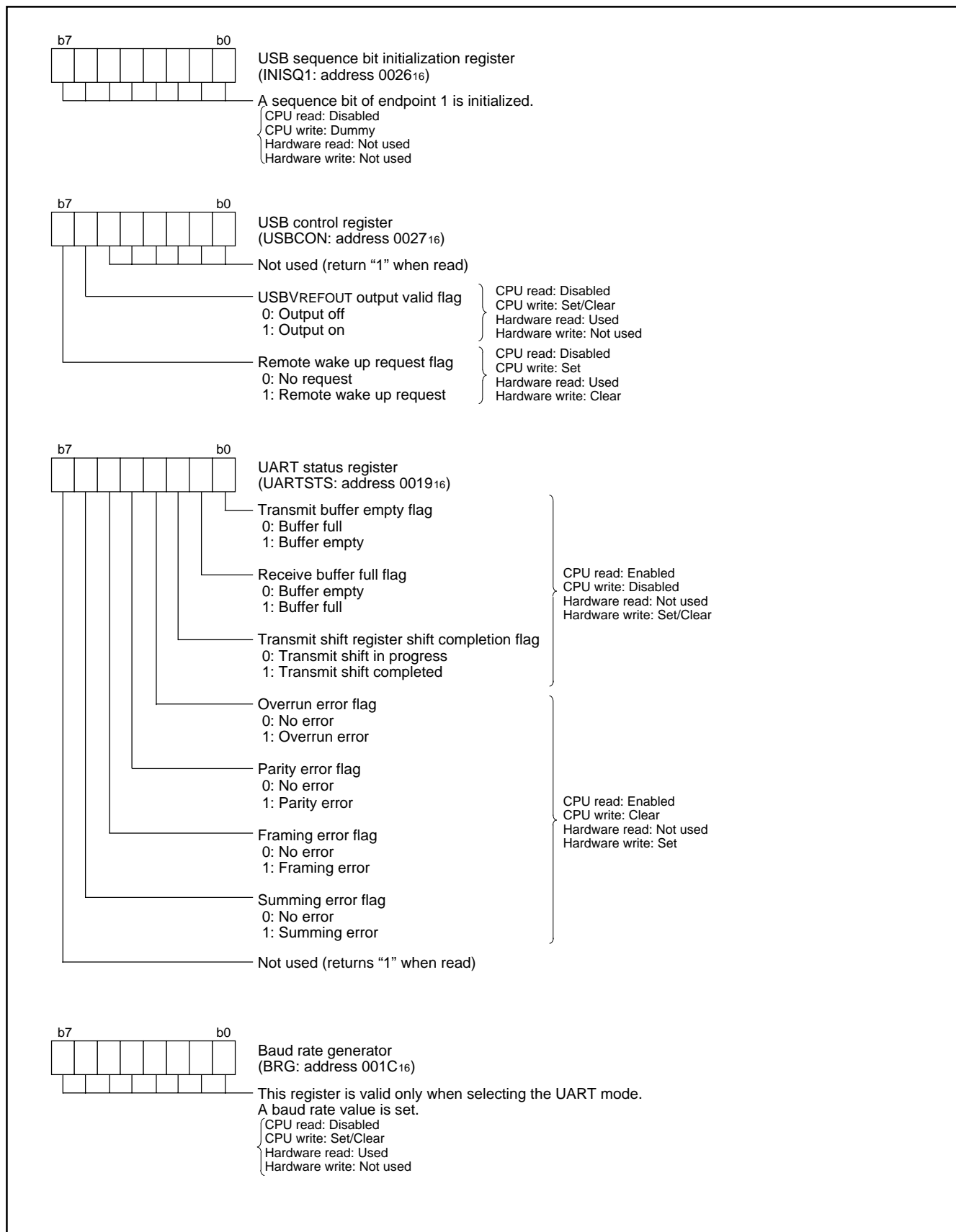


Fig. 32 Structure of serial I/O1-related registers (4)

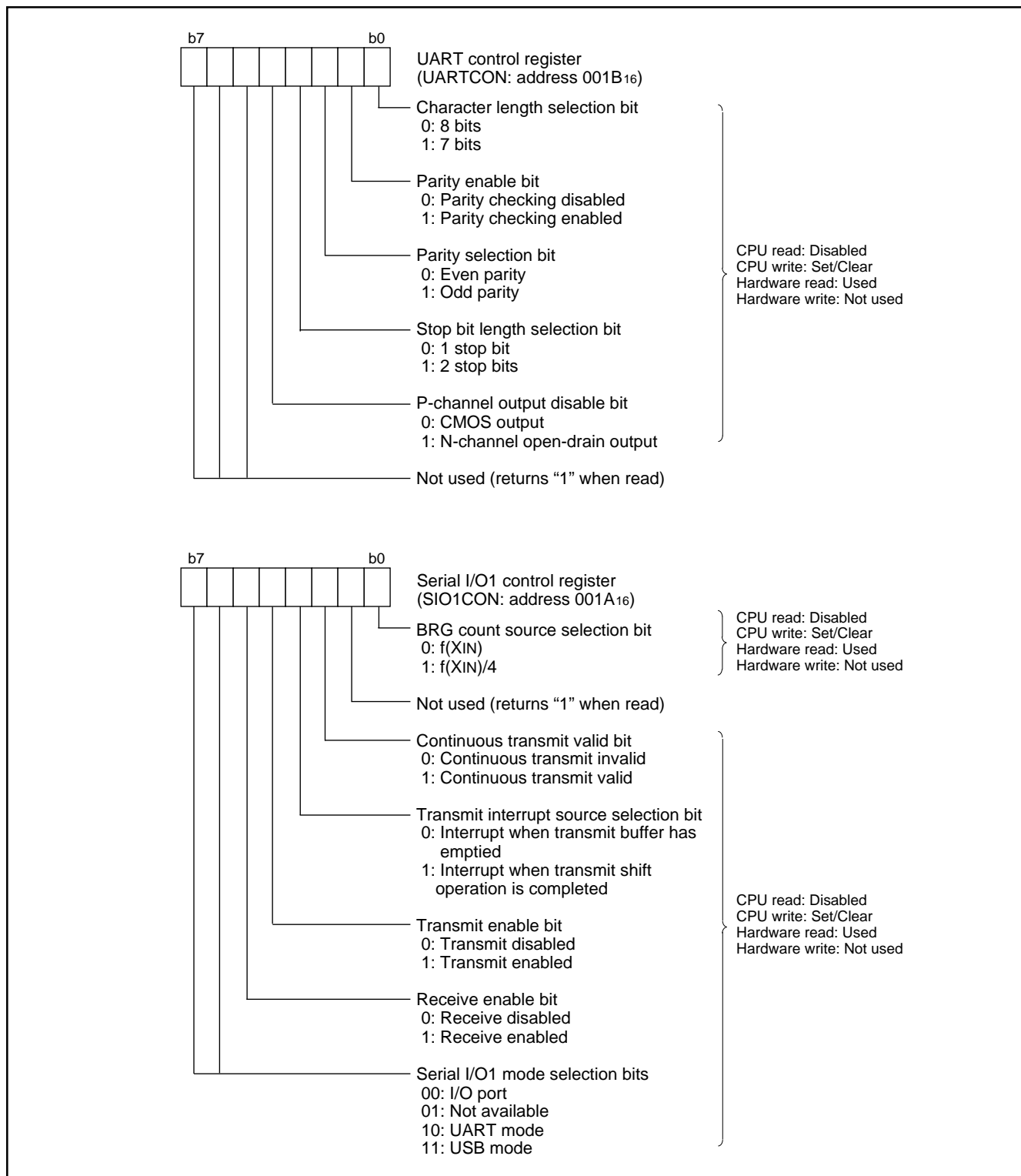


Fig. 33 Structure of serial I/O1-related registers (5)

Note on using USB mode

Handling of SE0 signal in program (at receiving)

7534 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 5 shows as follow.

- TKNE: Token interrupt enable (bit 6 of address 20₁₆)
- RSME: Resume interrupt enable (bit 5 of address 20₁₆)
- RSTE: USB reset interrupt enable (bit 4 of address 20₁₆)
- Spec: A response of the device requested by Low-Speed USB2.0 specification
- SIE: Hardware operation in 7534 group
- F/W: Recommendation process in the program
- FEOPE: False EOP error flag (bit 2 of address 19₁₆)
- RxPID: Token interrupt flag (bit 7 of address 1F₁₆)

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 μs 0.5 μs	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 μs 2.5 μs	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 μs 2.67 μs	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 μs	Spec	Reset	Reset	Reset	F/W	Reset interrupt processing Resume interrupt processing	
	SIE	Reset interrupt request	Reset interrupt request	Reset interrupt request			
	F/W	Reset processing	Reset processing	Reset processing			

- Function of USBPID control register 0 (address 0023₁₆)

Bit 4 (STALL handshake control for OUT token) of this register is forcibly set by SIE under the special condition shown below.

Set condition; when PID of data packet = DATA0 (incorrect PID) in the status stage of the control read transfer.

- SYNC field at reception

Normally, the SYNC field consists of "KJKJKJKK" (8 bits). However, as for SIE of the 7534 Group, when the low-order 6 bits are "KJKJKK", it is determined as SYNC.

●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.

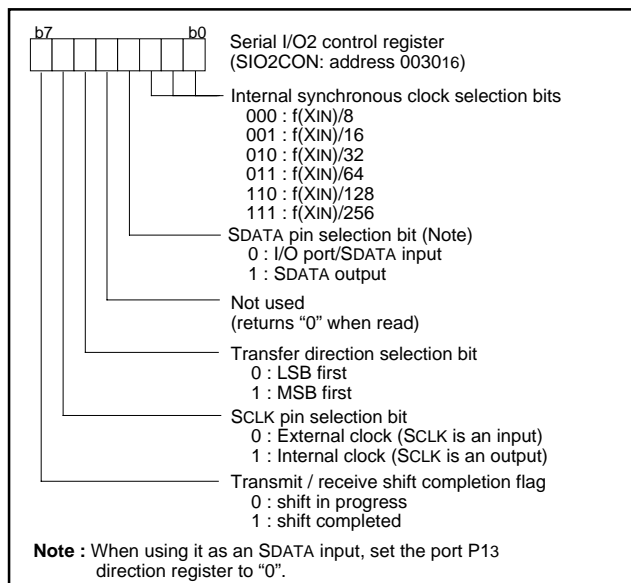


Fig. 34 Structure of serial I/O2 control registers

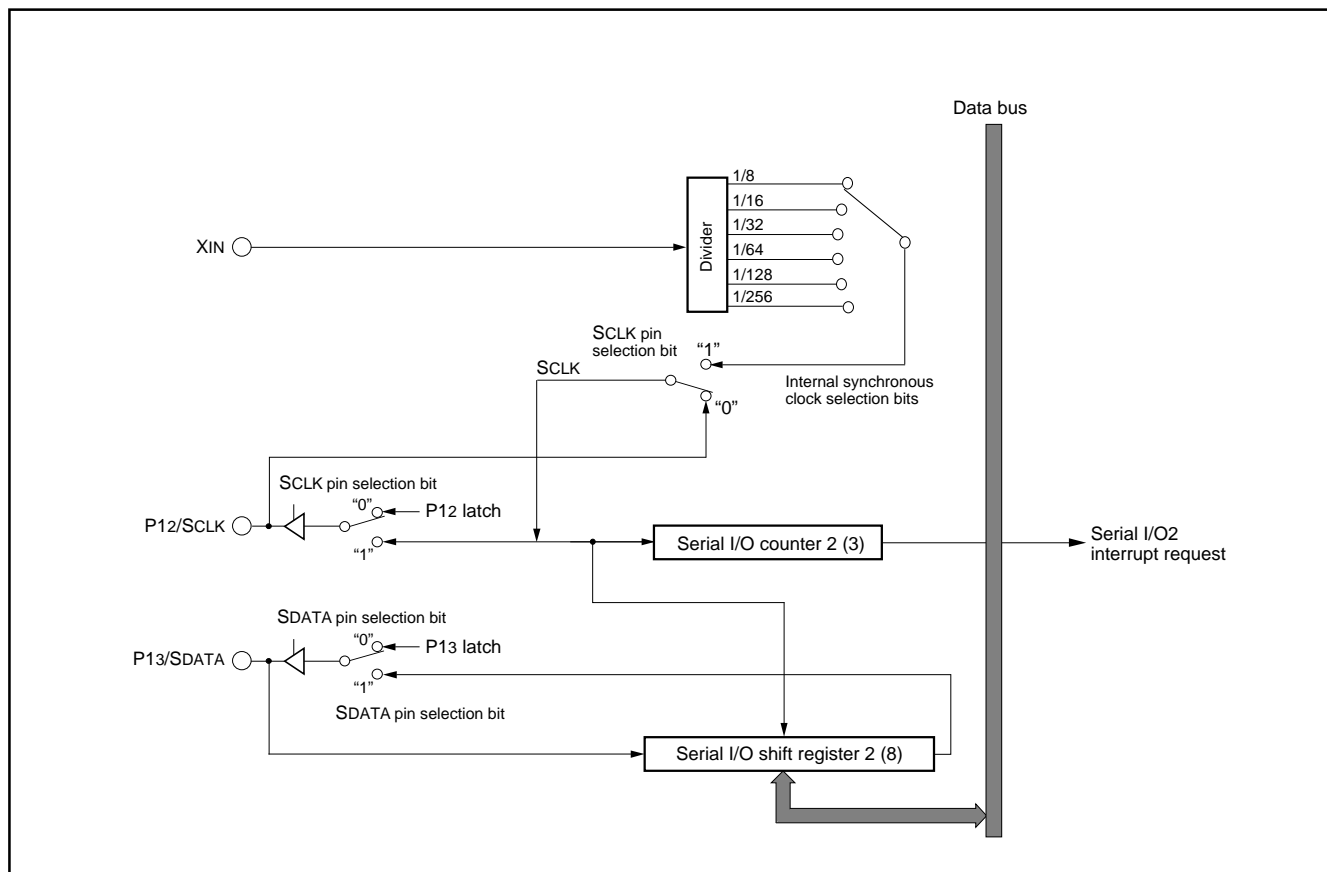


Fig. 35 Block diagram of serial I/O2

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 36.

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.

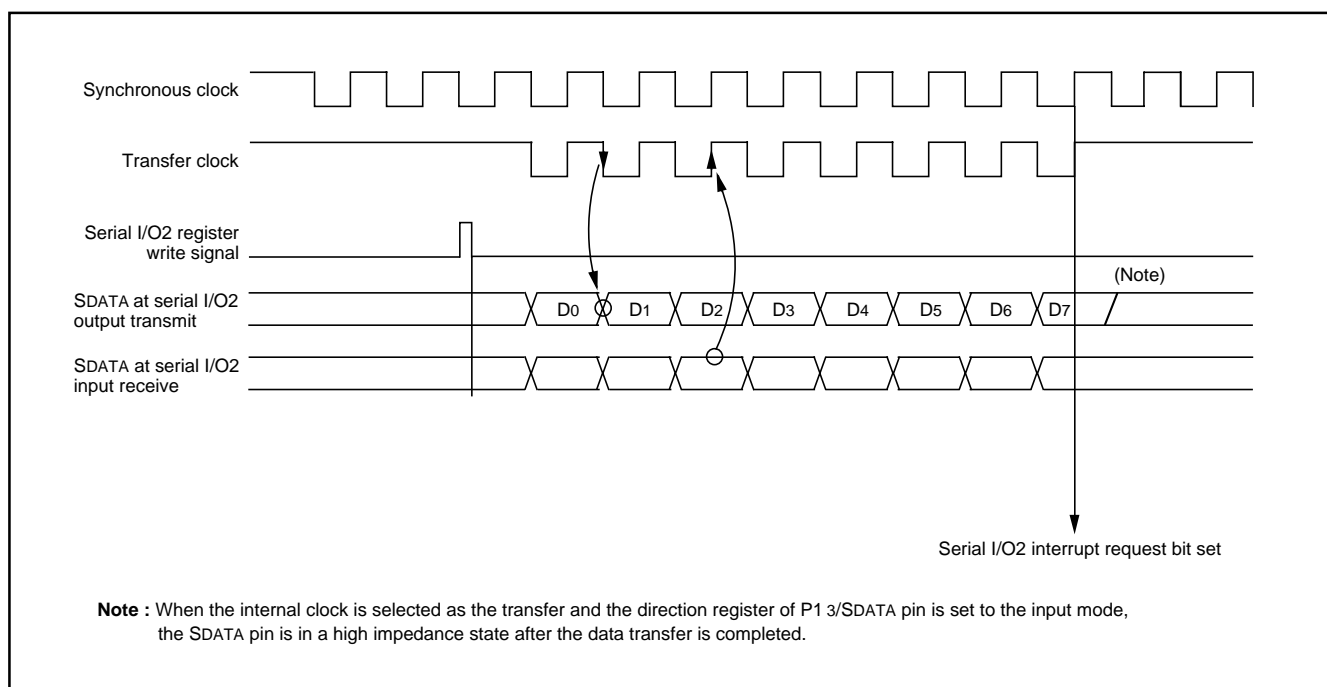


Fig. 36 Serial I/O2 timing (LSB first)

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 out 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" except during an A/D conversion.

[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.

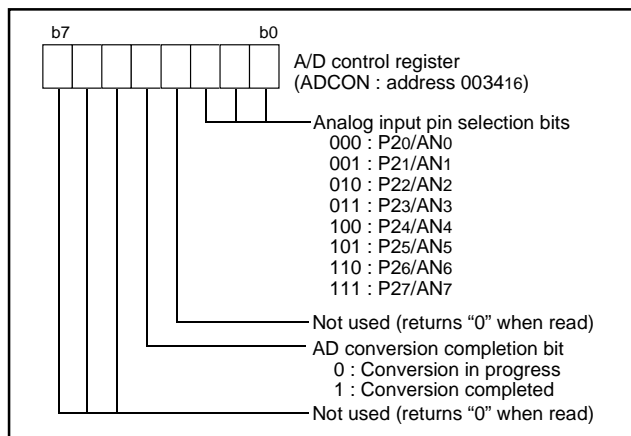


Fig. 37 Structure of A/D control register

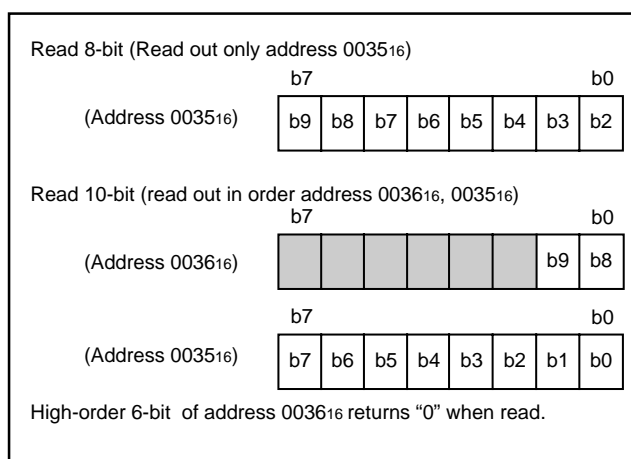


Fig. 38 Structure of A/D conversion register

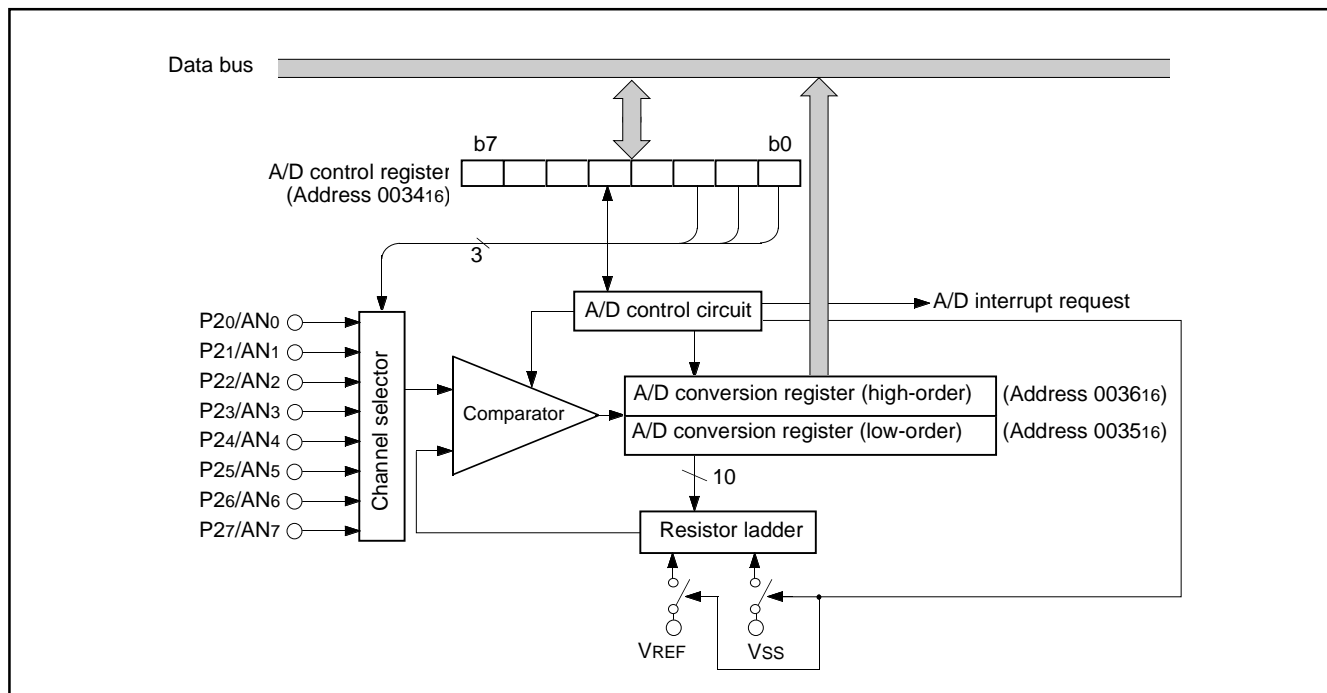


Fig. 39 Block diagram of A/D converter

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 0039₁₆) is not set after reset. Writing an optional value to the watchdog timer control register (address 0039₁₆) 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 0039₁₆) can be set before an underflow occurs.

When the watchdog timer control register (address 0039₁₆) 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 0039₁₆), the watchdog timer H is set to "FF₁₆" and the watchdog timer L is set to "FF₁₆".

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 0039₁₆). When this bit is "0", the count source becomes a watchdog timer L underflow signal. The detection time is 174.763 ms at f(X_{IN})=6 MHz. When this bit is "1", the count source becomes f(X_{IN})/16. In this case, the detection time is 683 μs at f(X_{IN})=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 0039₁₆). 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.

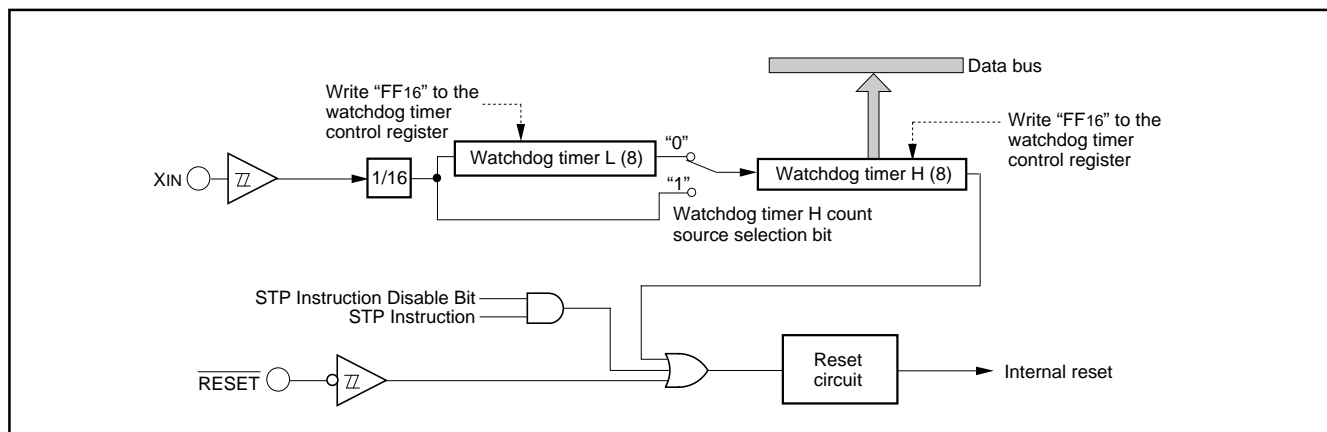


Fig. 40 Block diagram of watchdog timer

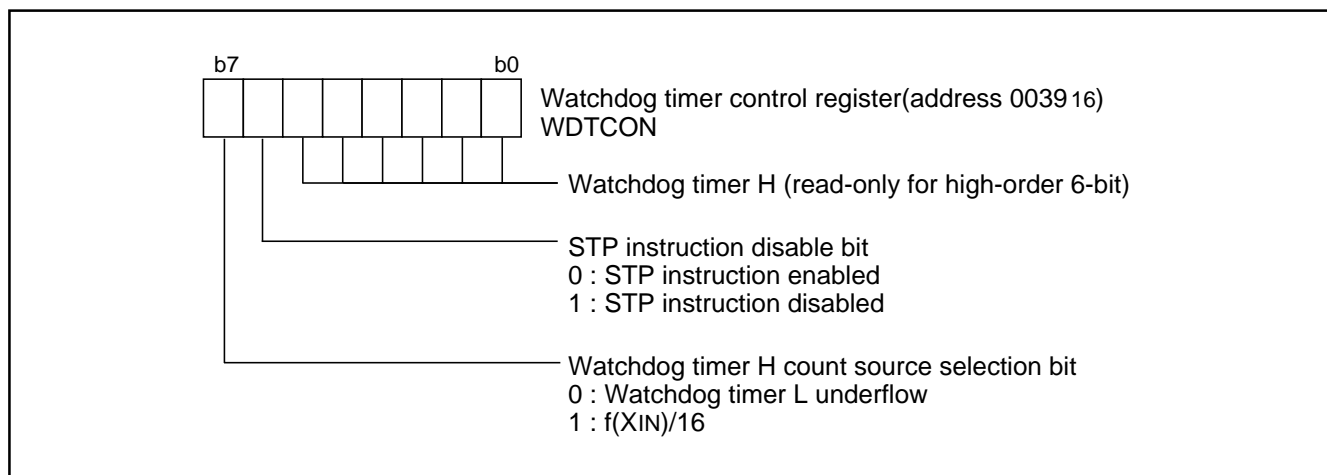


Fig. 41 Structure of watchdog timer control register

Reset Circuit

The microcomputer is put into a reset status by holding the $\overline{\text{RESET}}$ pin at the "L" level for 15 μs or more when the power source voltage is 4.1 to 5.5 V and X_{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 FFFD_{16} as high-order address and the contents of address 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.

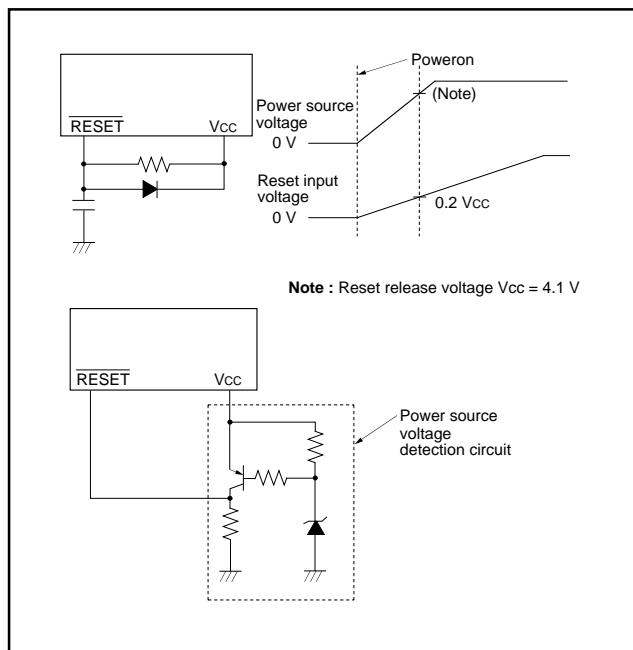


Fig. 42 Example of reset circuit

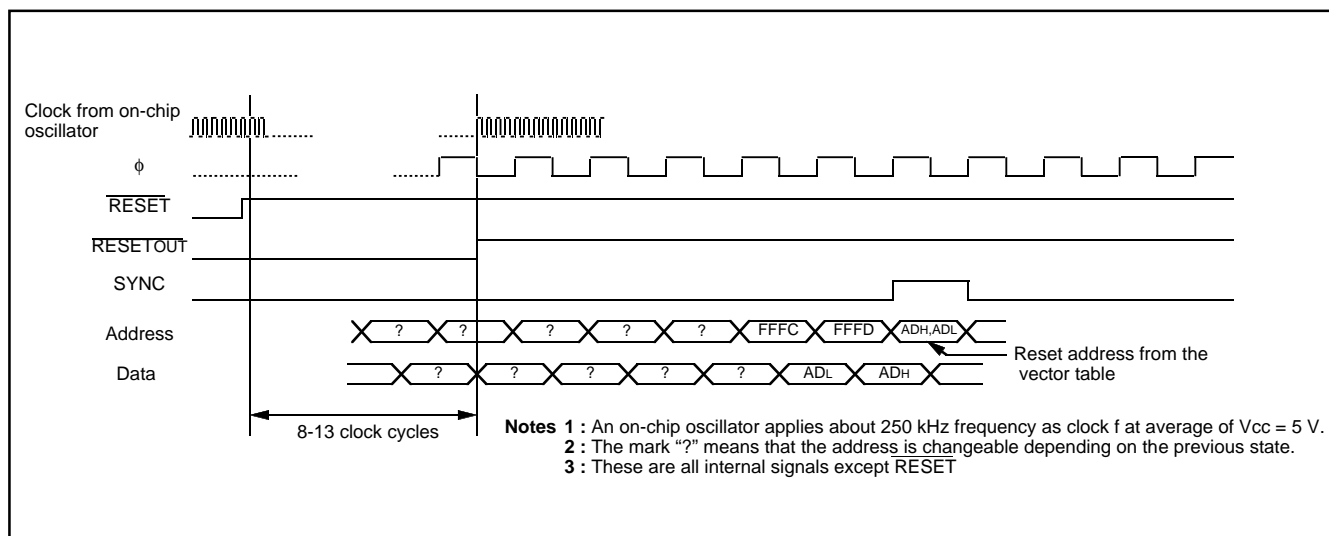


Fig. 43 Timing diagram at reset

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

Note X : Undefined

Fig. 44 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. (An external feed-back resistor may be needed depending on conditions.)

●Oscillation control

• Stop mode

When the STP instruction is executed, the internal clock ϕ 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 ϕ remains at "H" until timer 1 underflows. As soon as timer 1 underflows, the internal clock ϕ 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 \overline{RESET} pin while oscillation becomes stable.

• Wait mode

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

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

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

When the STP status is released, prescaler 12 and timer 1 will start counting clock which is X_{IN} divided by 16, so set the timer 1 interrupt enable bit to "0" before the STP instruction is executed.

Note

For use with the oscillation stabilization 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 an on-chip 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.

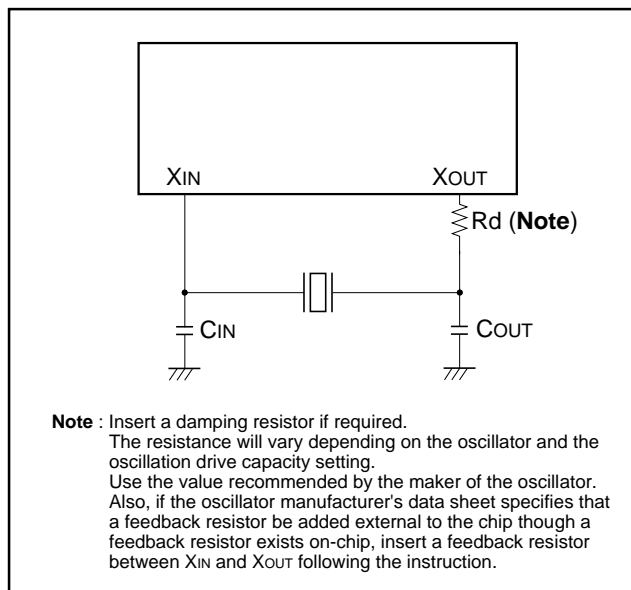


Fig. 45 External circuit of ceramic resonator

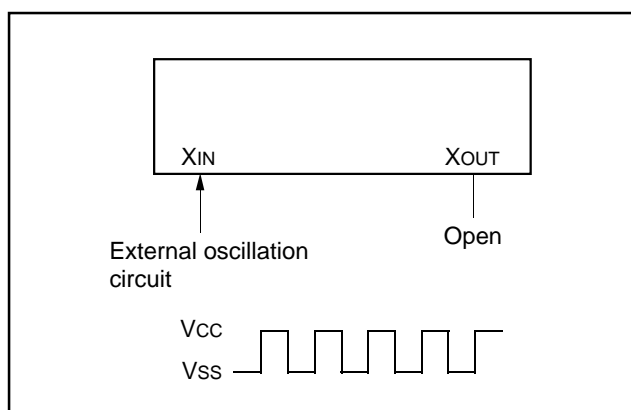


Fig. 46 External clock input circuit

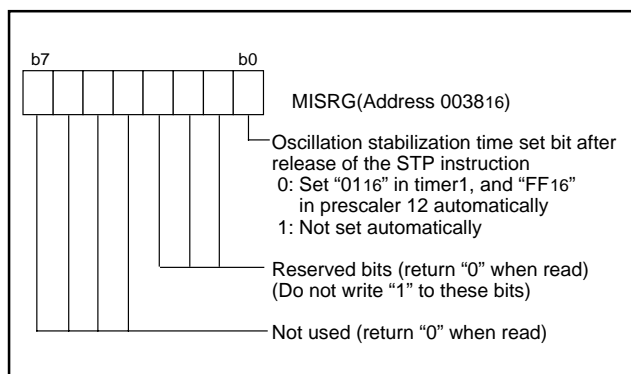


Fig. 47 Structure of MISRG

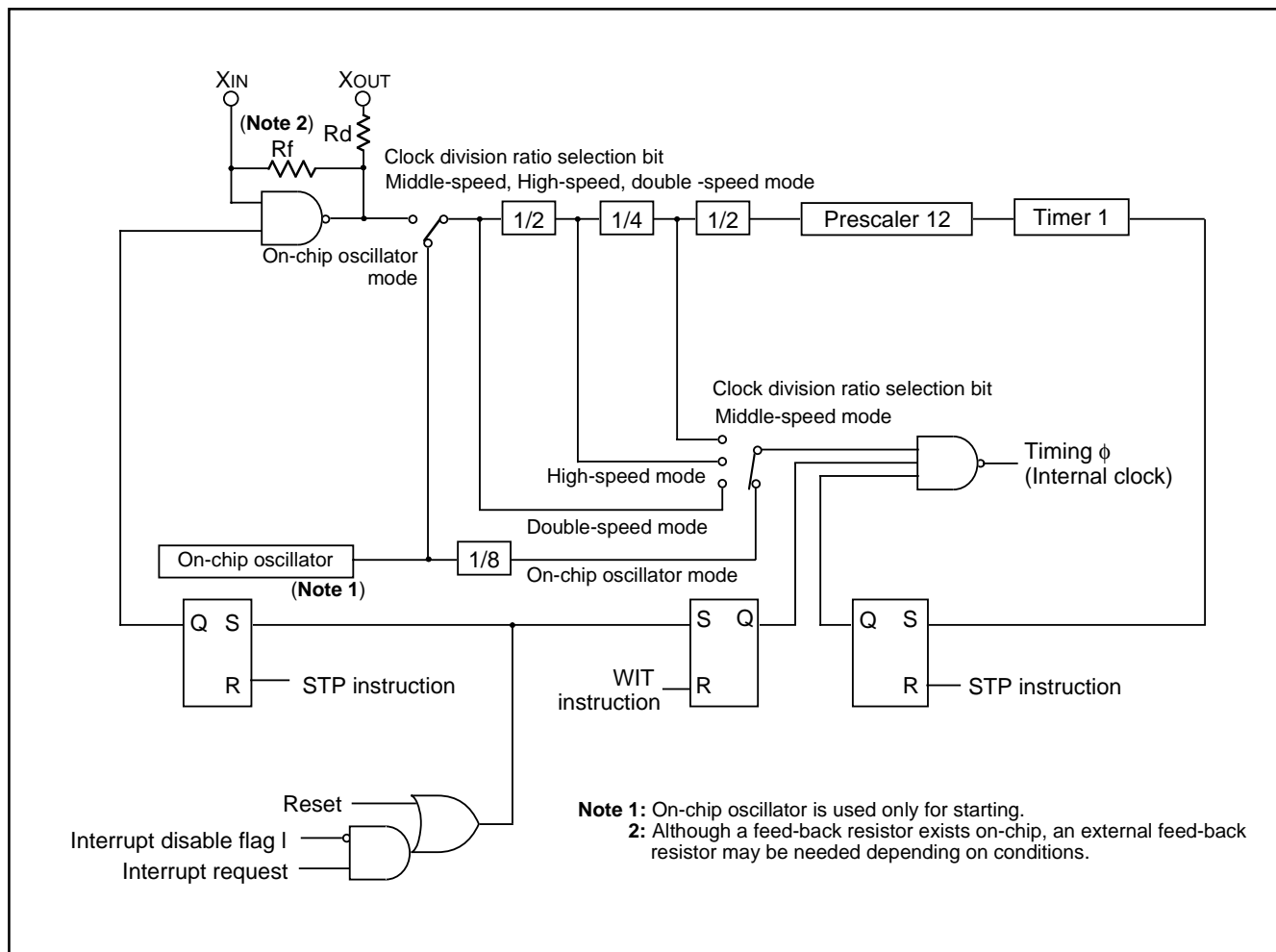


Fig. 48 Block diagram of system clock generating circuit (for ceramic resonator)

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 before 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.
- As for the 36-pin version, set "1" to each bit 6 of the port P3 direction register and the port P3 register.
- As for the 32-pin version, set "1" to respective bits 5, 6, 7 of the port P3 direction register and port P3 register.

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.

Watchdog Timer

The internal reset may not be generated correctly in the middle-speed mode, depending on the underflow timing of the watchdog timer. When using the watchdog timer, operate the MCU in any mode other than the middle-speed mode (i.e., high-speed, low-speed or double-speed mode).

Instruction Execution Timing

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

The frequency of the internal clock ϕ 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.

Note on stack page

When 1 page is used as stack area by the stack page selection bit, the area which can be used as stack depends on RAM size. Especially, be careful that the RAM area varies in Mask ROM version, One Time PROM version and Emulator MCU.

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 or electrolytic capacitor of 1.0 μ 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 0.22 μ F is recommended.

USB Communication

- In applications requiring high-reliability, we recommend providing the system with protective measures such as USB function initialization by software or USB reset by the host to prevent USB communication from being terminated unexpectedly, for example due to external causes such as noise.
- When USB suspend mode with TTL level on P10, P12, P13 input level selection bit (bit 3 of address 1716) set to "1", suspend current as I_{CC} might be greater than 300 μ A as a spec.

[Countermeasure]

There are two countermeasures by software to avoid it as follows.

- (1) Change from TTL input level to CMOS input level for P10, P12, P13 port input.
- (2) Change from TTL input level to CMOS input level before STP instruction in suspend routine; then after RESUME or Remote wake up interrupt, return to TTL input level from CMOS input level. That is shown in Figure 49.

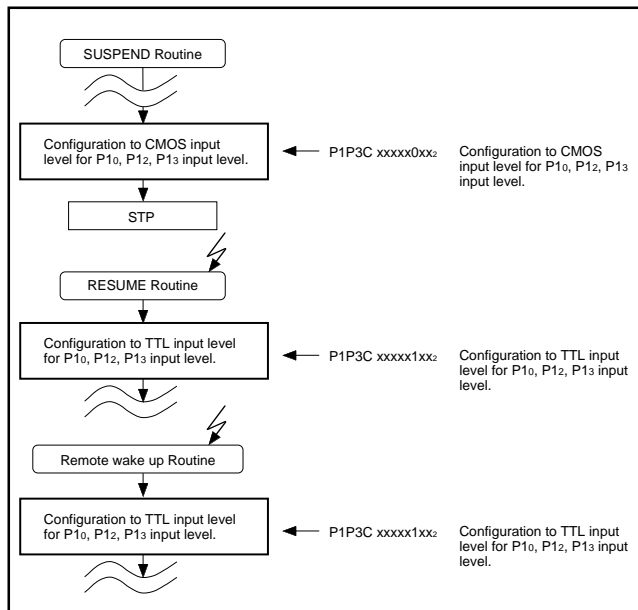


Fig. 49 Countermeasure (2) by software

One Time PROM Version

The CNVss pin is connected to the internal memory circuit block by a low-ohmic resistance, since it has the multiplexed function to be a programmable power source pin (VPP pin) as well. To improve the noise reduction, connect a track between CNVss pin and Vss pin with 1 to 10 kΩ resistance. The mask ROM version track of CNVss pin has no operational interference even if it is connected via a resistor.

Electric Characteristic Differences Among Mask ROM and One Time PROM Version MCUs

There are differences in electric characteristics, operation margin, noise immunity, and noise radiation among mask ROM and One Time PROM version MCUs due to the differences in the manufacturing processes. When manufacturing an application system with One Time PROM version and then switching to use of the mask ROM version, perform sufficient evaluations for the commercial samples of the mask ROM version.

Note on Power Source Voltage

When the power source voltage value of a microcomputer is less than the value which is indicated as the recommended operating conditions, the microcomputer does not operate normally and may perform unstable operation. In a system where the power source voltage drops slowly when the power source voltage drops or the power supply is turned off, reset a microcomputer when the supply voltage is less than the recommended operating conditions and design a system not to cause errors to the system by this unstable operation.

Note on A/D Converter

Method to stabilize A/D Converter is described below.
 (a) A/D conversion accuracy could be affected for Bus Powered*1 USB devices, while the communicating. Figure 50 shows the method to stabilize A/D conversion accuracy, inserting a capacitor between Vref and Vss.
 *1: Power supplied by USB Vcc BUS.

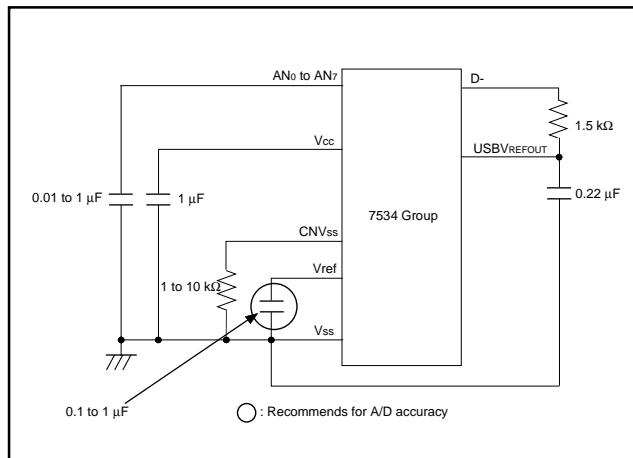


Fig. 50 Method to stabilize A/D conversion accuracy

(b) It is recommended for A/D accuracy to avoid converting while USB communication, and use average value of several converted values.

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)

* For the mask ROM confirmation and the mark specifications, refer to the "Renesas Technology Corp." Homepage (<http://www.renesas.com>).

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
PLQP0032GB-A	PCA7435GPG03
PRSP0036GA-A	PCA7435FP, PCA7435FPG02
PRDP0042BA-A	PCA7435SP, PCA7435SPG02

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 51 is recommended to verify programming.

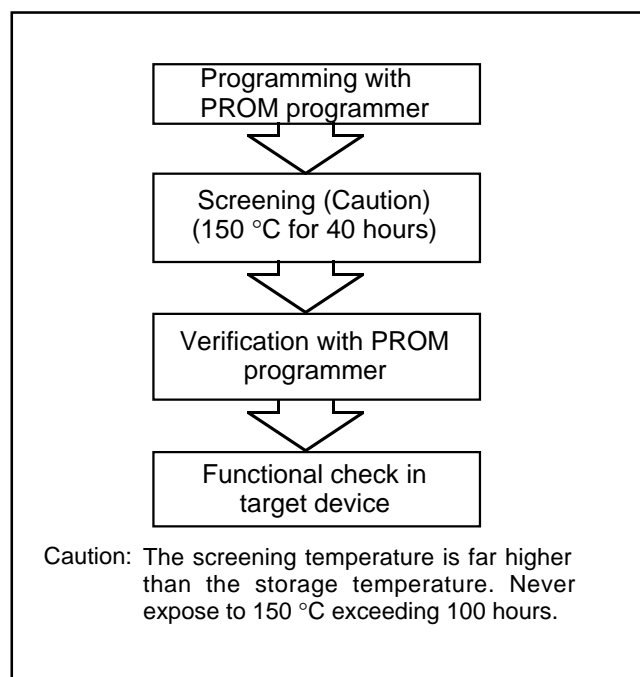


Fig. 51 Programming and testing of One Time PROM version

FUNCTIONAL DESCRIPTION SUPPLEMENT

Interrupt

7534 group permits interrupts on the 14 sources for 42-pin version, 13 sources for 36-pin version and 12 sources for 32-pin version. 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 ₁₆	FFFC ₁₆	At reset input	Non-maskable
UART receive	2	FFFB ₁₆	FFFA ₁₆	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 ₁₆	FFF8 ₁₆	At completion of UART transmit shift or when transmit buffer is empty	Valid in UART mode
USB SETUP/OUT token				At detection of SETUP/OUT token or	Valid in USB mode
Reset/Suspend/Resume				At detection of Reset/ Suspend/ Resume	
INT ₁ (Note 3)	4	FFF7 ₁₆	FFF6 ₁₆	At detection of either rising or falling edge of INT ₁ input	External interrupt (active edge selectable)
INT ₀ (Note 4)				At detection of either rising or falling edge of INT ₀ input	External interrupt (active edge selectable)
Timer X	5	FFF5 ₁₆	FFF4 ₁₆	At timer X underflow	
Key-on wake-up				At falling of conjunction of input logical level for port P0 (at input)	External interrupt (valid at falling)
Timer 1	6	FFF3 ₁₆	FFF2 ₁₆	At timer 1 underflow	STP release timer underflow
Timer 2	7	FFF1 ₁₆	FFF0 ₁₆	At timer 2 underflow	
Serial I/O2				At completion of transmit/receive shift	
CNTR ₀	8	FFEF ₁₆	FFEE ₁₆	At detection of either rising or falling edge of CNTR ₀ input	External interrupt (active edge selectable)
A/D conversion				At completion of A/D conversion	
BRK instruction	9	FFED ₁₆	FFEC ₁₆	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.

3: The INT₁ interrupt does not exist in the 36-pin and 32-pin version.

4: The INT₀ interrupt does not exist in the 32-pin version.

Timing After Interrupt

The interrupt processing routine begins with the machine cycle following the completion of the

instruction that is currently in execution.

Figure 52 shows a timing chart after an interrupt occurs, and Figure 53 shows the time up to execution of the interrupt processing routine.

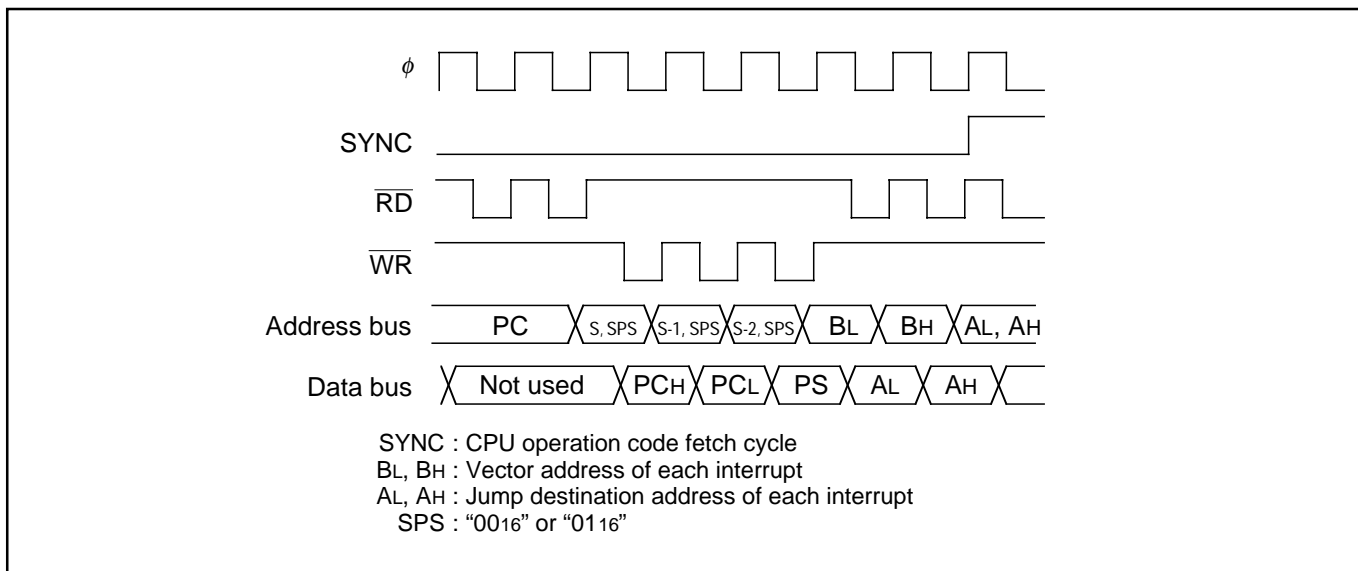


Fig. 52 Timing chart after an interrupt occurs

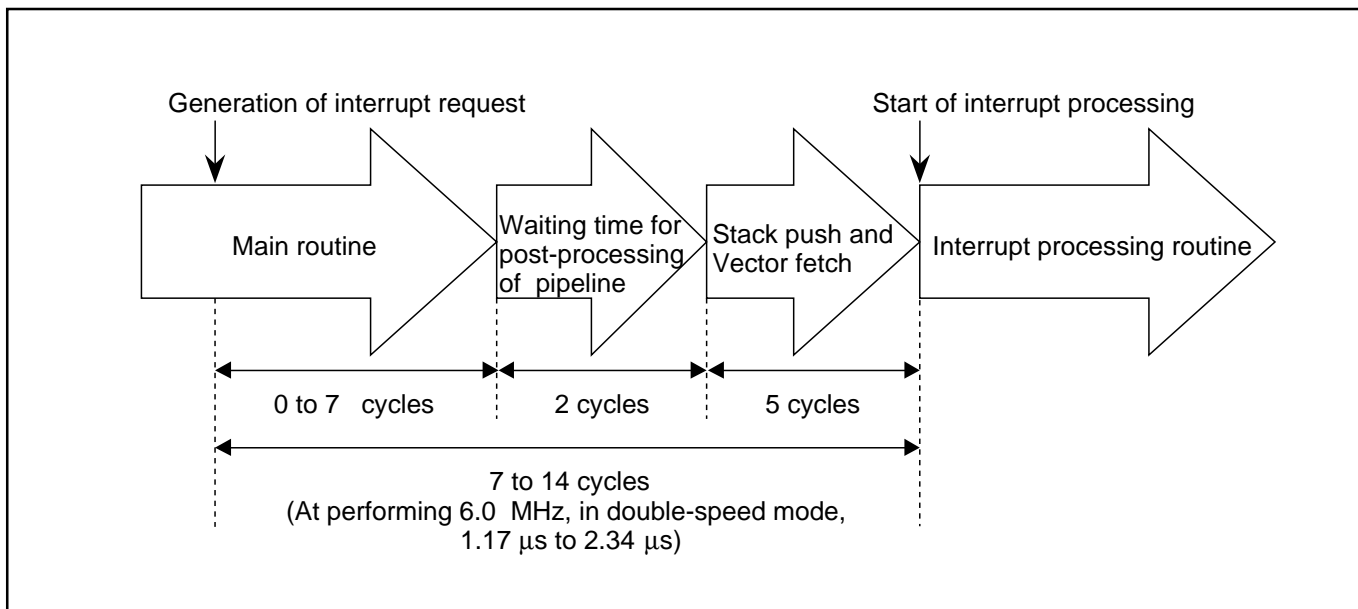


Fig. 53 Time up to execution of the interrupt processing routine

A/D Converter

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

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

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

A/D conversion completes at 122 clock cycles (20.34 μs at f(XIN) = 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 VREF of A/D converter and Vref

When n = 0 Vref = 0

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

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

Table 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	0 0 0 0 0 0 0 0 0 0 0	0
First comparison	1 0 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$
Second comparison	*1 1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4}$
Third comparison	*1 *2 1 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8}$
⋮	⋮	⋮
After completion of tenth comparison	A result of A/D conversion *1 *2 *3 *4 *5 *6 *7 *8 *9 *10	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \dots \pm \frac{V_{REF}}{1024}$

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

Figure 54 shows A/D conversion equivalent circuit, and Figure 55 shows A/D conversion timing chart.

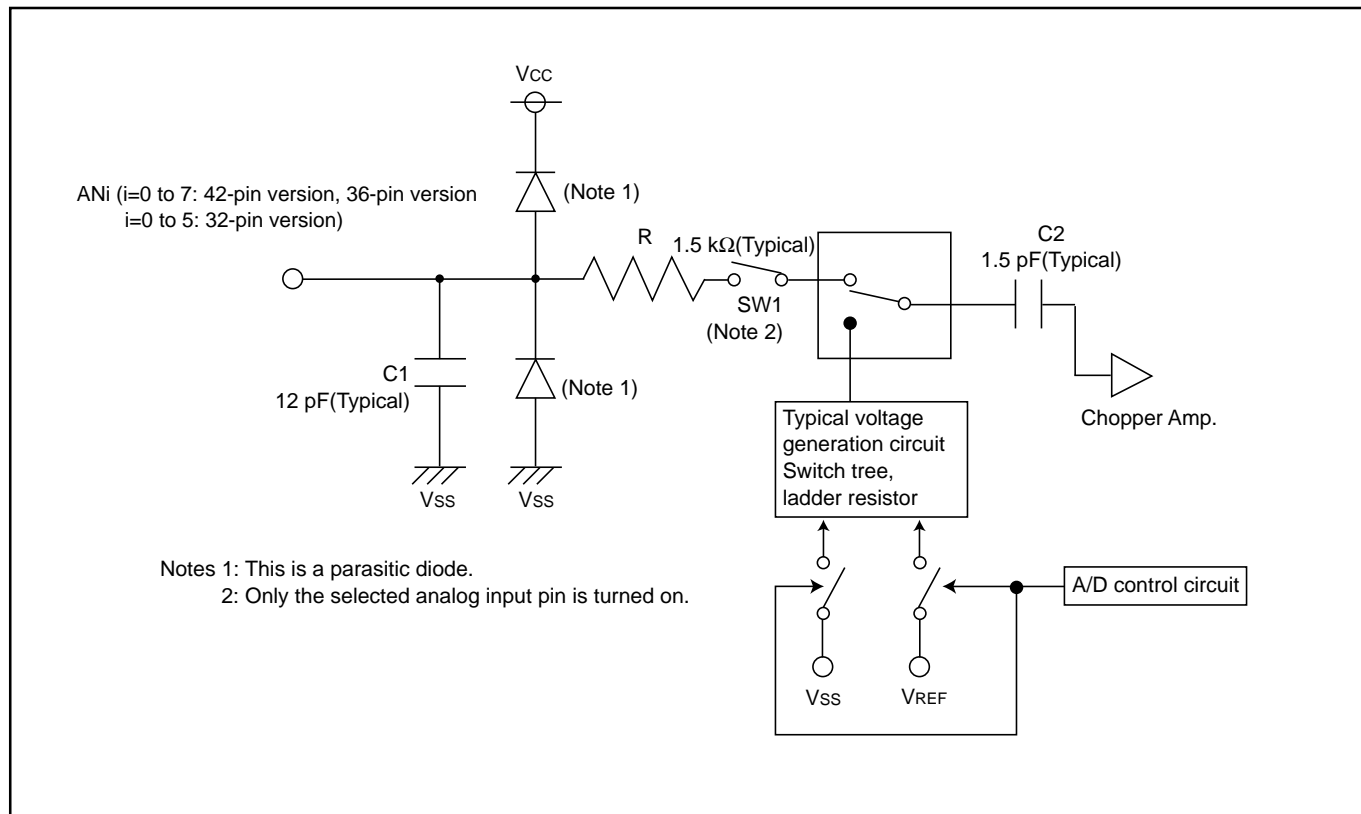


Fig. 54 A/D conversion equivalent circuit

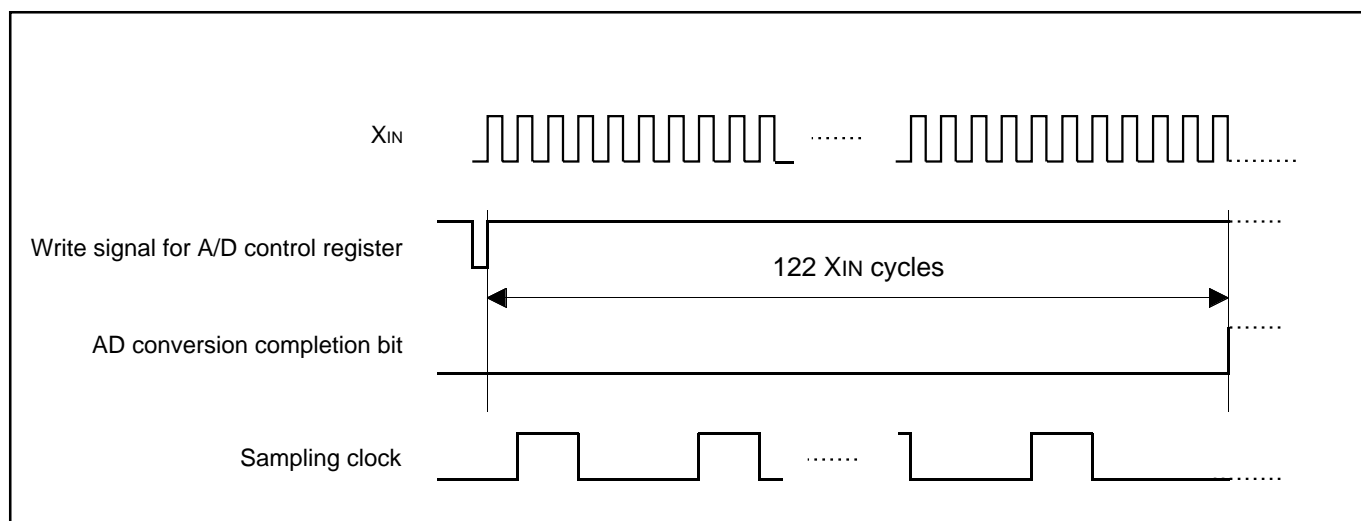


Fig. 55 A/D conversion timing chart

Stop mode

System enters the stop mode by executing the **STP** instruction. In the stop mode, the $f(X_{IN})$ oscillation is stopped, and the internal clock ϕ 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 STP instruction execution	SFR	State retained (Timer 1 and prescaler 12 excepted)
Timer	At selecting internal count source: Stop At selecting external count source: Operating (only Timer X)	CPU register	State retained • Accumulator • Index register X • Index register Y • Stack pointer • Program counter • Processor status register
UART	Stop		
A/D conversion	Stop		
Serial I/O2	At selecting internal synchronous clock: Stop At 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, reset)

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"

Wait mode

System enters the wait mode by executing the **WIT** instruction. In the wait mode, the oscillation is operating, but the internal clock ϕ 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 WIT instruction execution retained
Timer	At selecting internal count source: Operating At selecting external count source: Operating
UART	Operating
A/D conversion	Operating (Conversion is continued if the WIT instruction is executed during conversion)
Serial I/O2	At selecting internal synchronous clock: Operating At 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"> • Accumulator • Index register X • Index register Y • Stack pointer • Program counter • Processor status register

(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"

DESCRIPTION OF IMPROVED USB FUNCTION FOR 7534 GROUP

Table 13 Description of improved USB function for 7534 Group

No.	Parameter	7532/7536 Group	7534 Group
1	Response at Control transfer	Not deal with the host which performs the Control transfer in parallel to plural device.	Connectable to the host which performs the Control transfer in parallel to plural device.
2	D+/D- transceiver circuit	USB function can be used only at the condition of $C_L = 150 \text{ pF}$ to 350 pF .	Deal with the following Low-Speed USB2.0 specification. $C_L = 200 \text{ pF}$ to 450 pF , Trise and Tfall: 75 ns to 300 ns, Tr/Tf: 80 % to 125 %, Cross over Voltage: 1.3 V to 2.0 V.
3	Power dissipation at Suspend	Rating is Max. $300 \mu\text{A}$ not including the output current of USBVREFOUT.	Rating is Max. $300 \mu\text{A}$ including the output current of USBVREFOUT, by low-power dissipation of D+/D- input circuit and 3.3 V-regulator.
4	STALL in Status stage	ACK is returned once to OUT (DATA0) to be valid in Status stage.	STALL is set automatically by hardware when OUT (DATA0) is received in Status stage.
5	6-bit decode of SYNC field	SYNC is detected only when 8-bit full code (8016) is complete.	SYNC is detected only the low-order 6 bits even if the high-order 2 bits are corrupted.

DIFFERENCES AMONG 32-PIN, 36-PIN AND 42-PIN

The 7534 Group has three package types, and each of the number of I/O ports are different. Accordingly, when the pins which have the function except a port function are eliminated, be careful that the functions are also eliminated.

Table 14 Differences among 32-pin, 36-pin and 42-pin

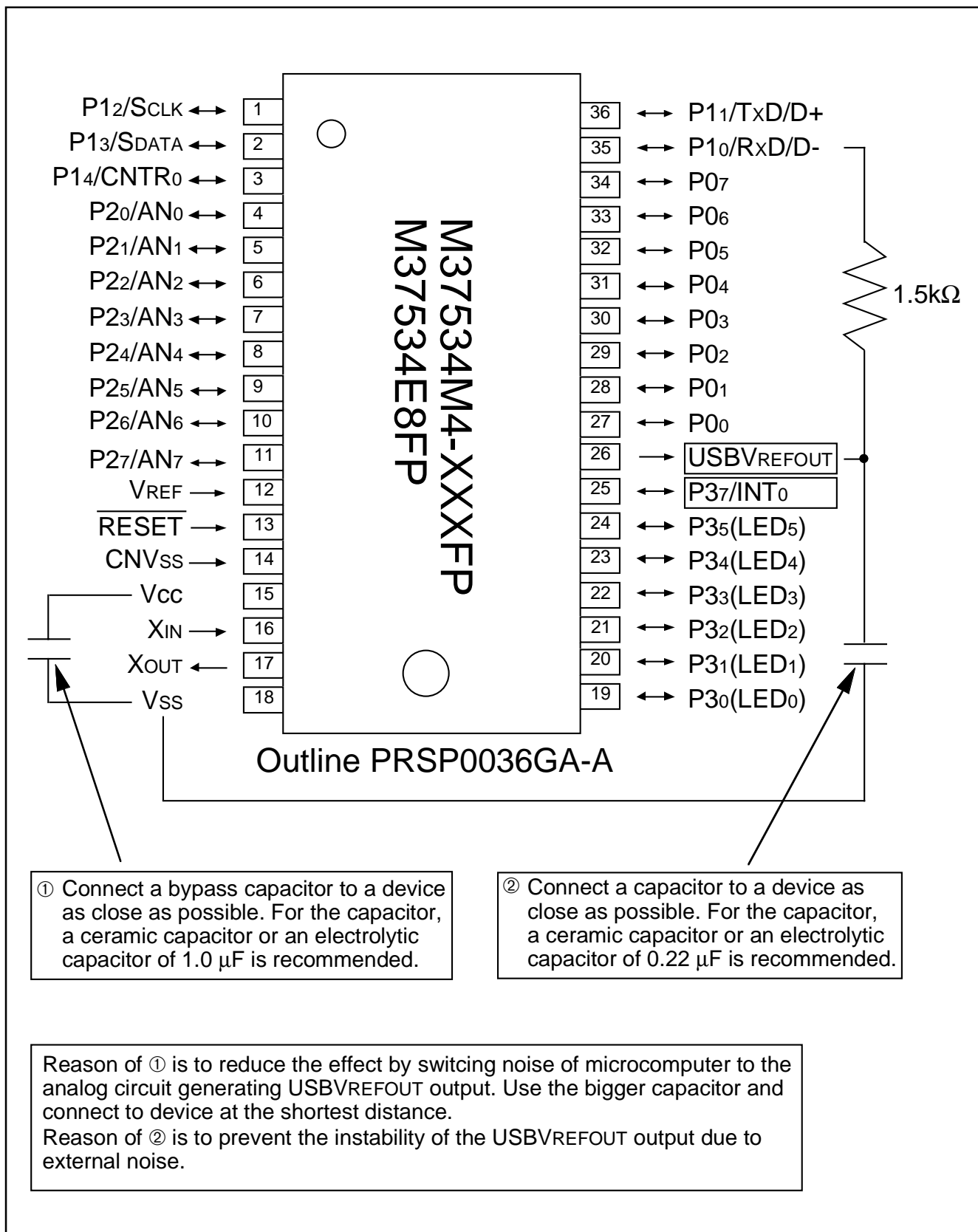
I/O port	42-pin SDIP	36-pin SSOP	32-pin LQFP
Port P1	P10–P16 (7-bit structure)	P10–P14 (5-bit structure)	P10–P14 (5-bit structure)
Port P2	P20–P27 (8-bit structure) (A/D converter 8-channel)	P20–P27 (8-bit structure) (A/D converter 8-channel)	P20–P25 (6-bit structure) (A/D converter 6-channel)
Port P3	P30–P37 (8-bit structure) (INT0, INT1 available)	P30–P35, P37 (7-bit structure) (INT0 available)	P30–P34 (5-bit structure) (INT function not available)
Port P4	P40, P41 (2-bit structure)	No port	No port

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

Table 15 Differences among 32-pin, 36-pin and 42-pin (SFR)

Register (Address)	42-pin SDIP	36-pin SSOP	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

DESCRIPTION SUPPLEMENT FOR USE OF USB FUNCTION STABLY



① Connect a bypass capacitor to a device as close as possible. For the capacitor, a ceramic capacitor or an electrolytic capacitor of 1.0 μF is recommended.

② Connect a capacitor to a device as close as possible. For the capacitor, a ceramic capacitor or an electrolytic capacitor of 0.22 μF 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.

Fig. 56 Handling of Vcc, USBVREFOUT pins of M37534M4-XXXXFP, M37534E8FP

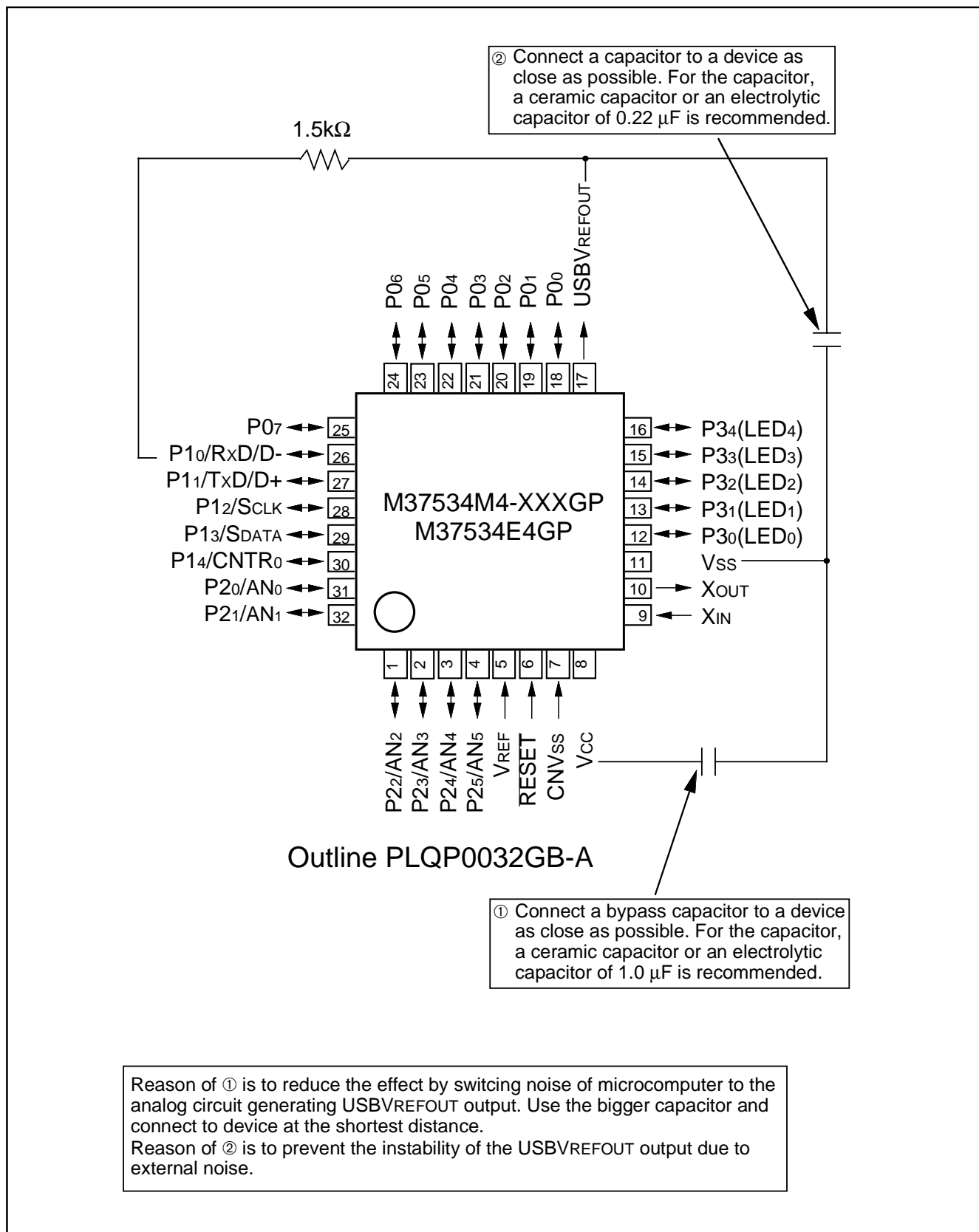


Fig. 57 Handling of Vcc, USBVREFOUT pins of M37534M4-XXXGP, M37534E4GP

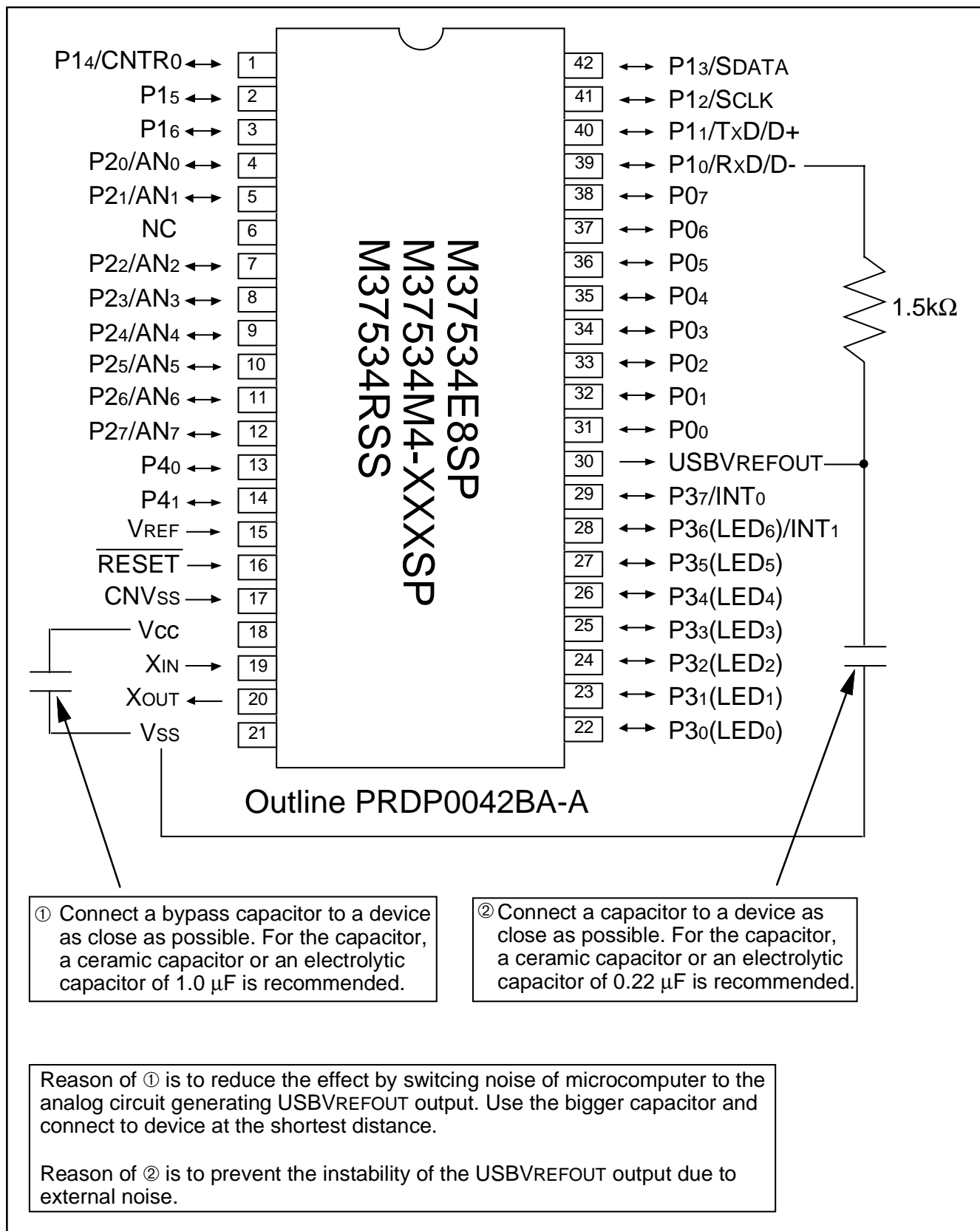


Fig. 58 Handling of Vcc, USBVREFOUT pins of M37534E8SP, M37534M4-XXXSP, M37534RSS

THIS PAGE IS BLANK FOR REASONS OF LAYOUT.

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

2.1 I/O port

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

2.1.1 Memory map

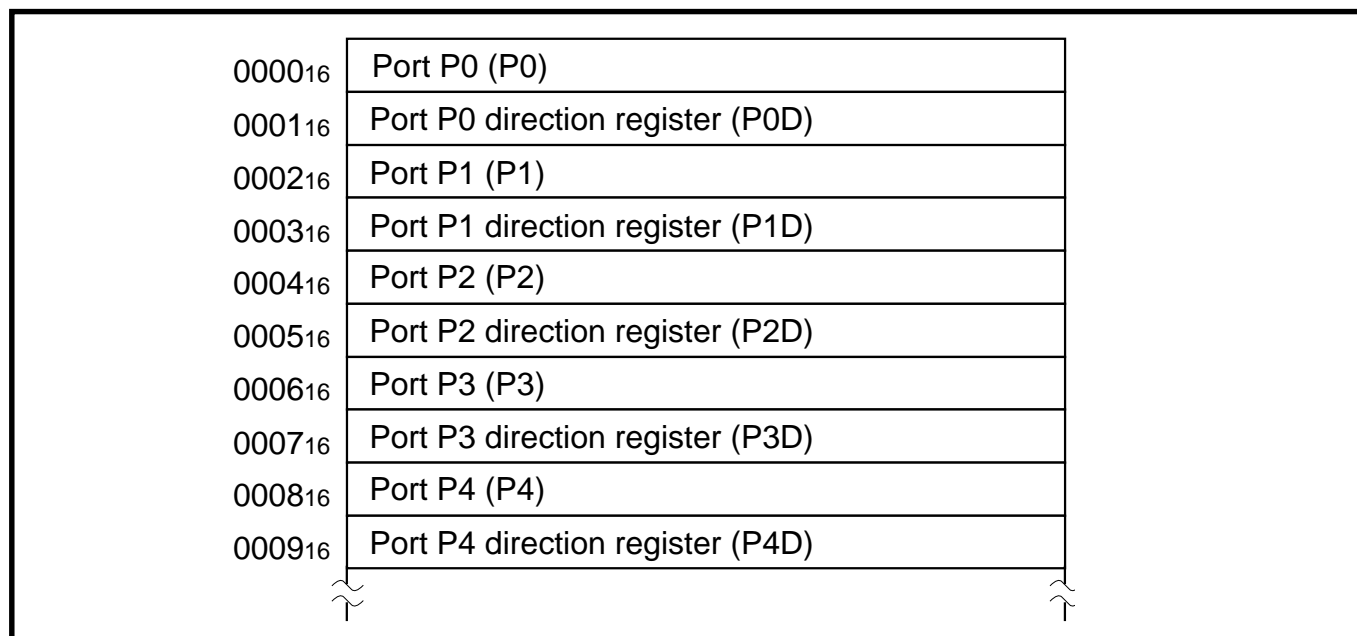


Fig. 2.1.1 Memory map of registers relevant to I/O port

2.1.2 Relevant registers

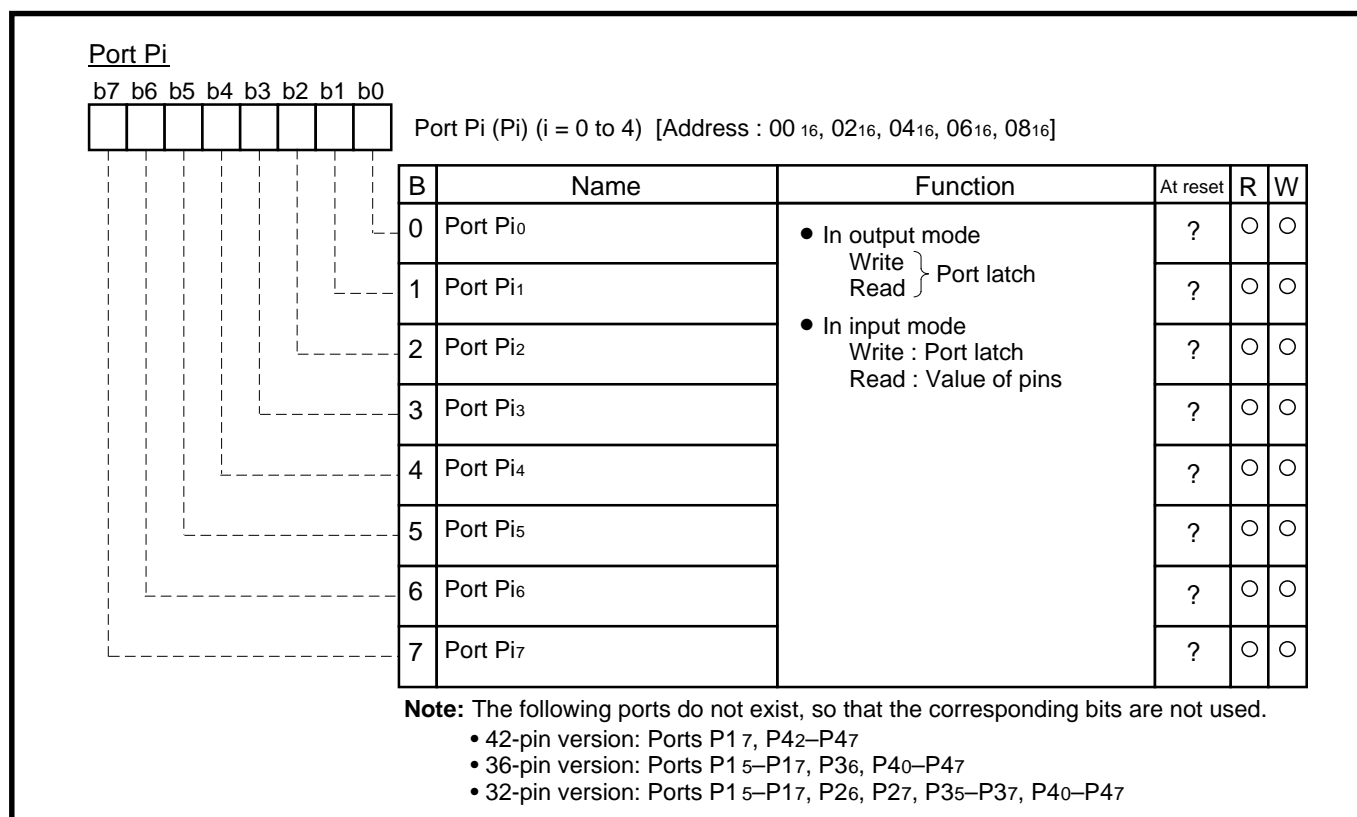


Fig. 2.1.2 Structure of Port P_i (i = 0 to 4)

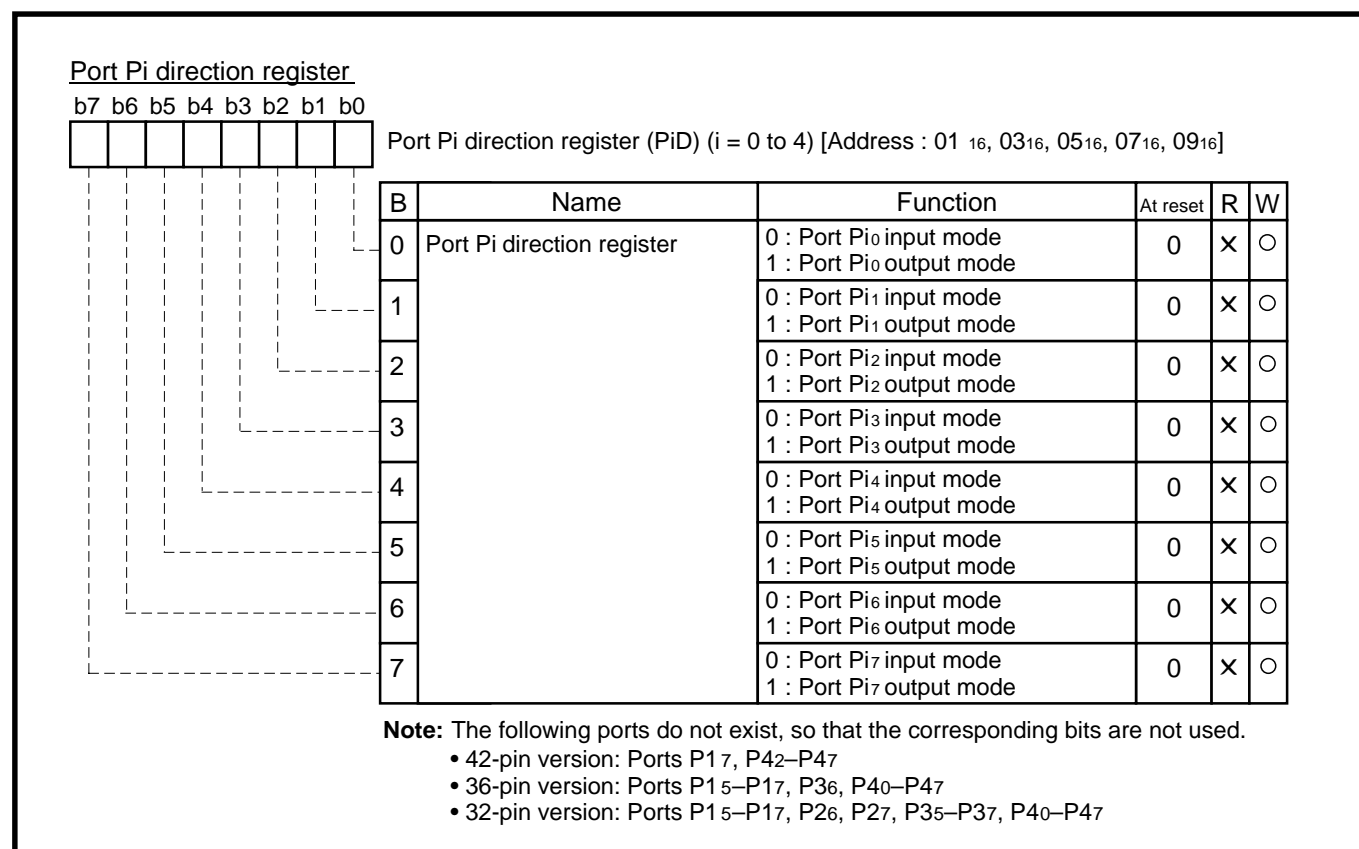


Fig. 2.1.3 Structure of Port Pi direction register (i = 0 to 4)

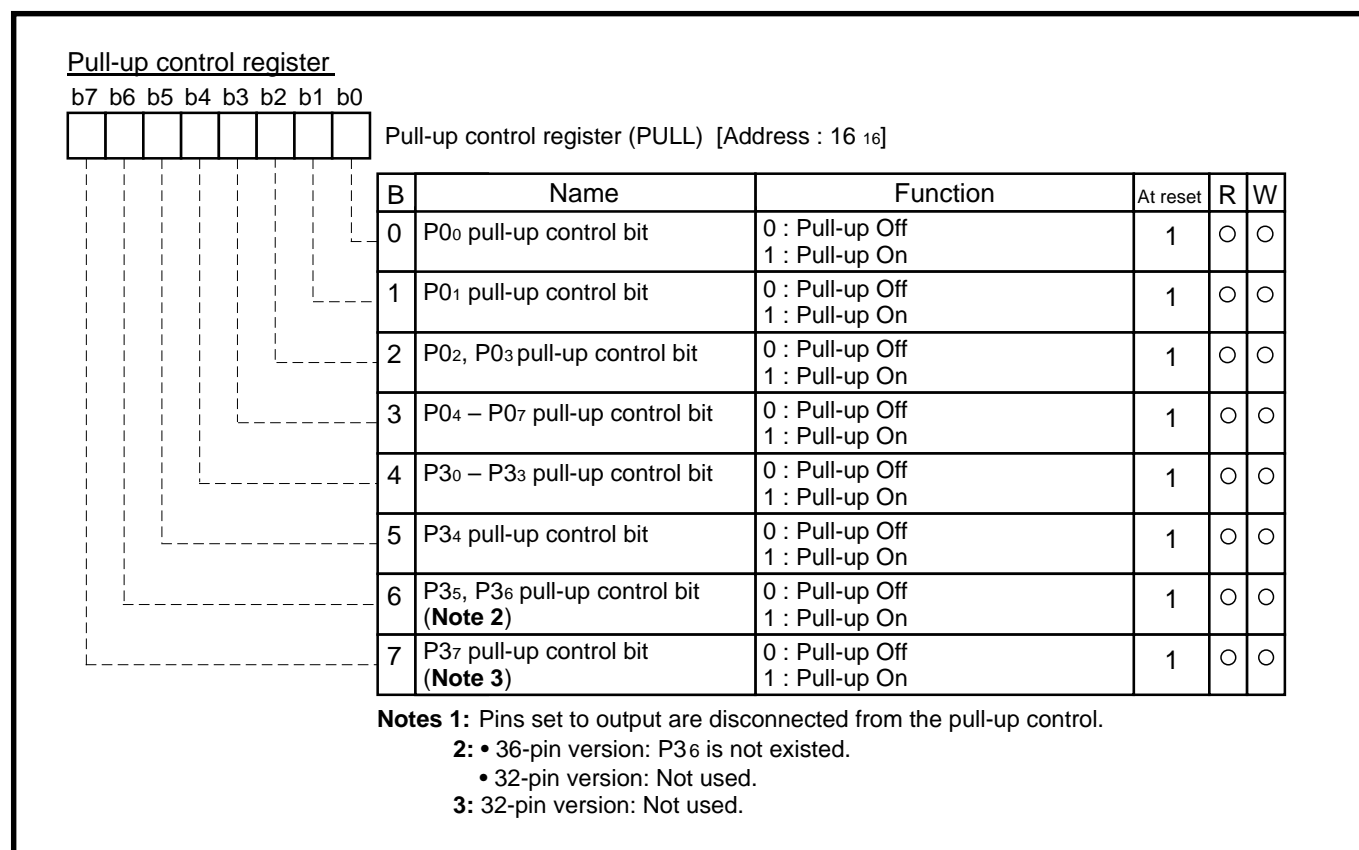


Fig. 2.1.4 Structure of Pull-up control register

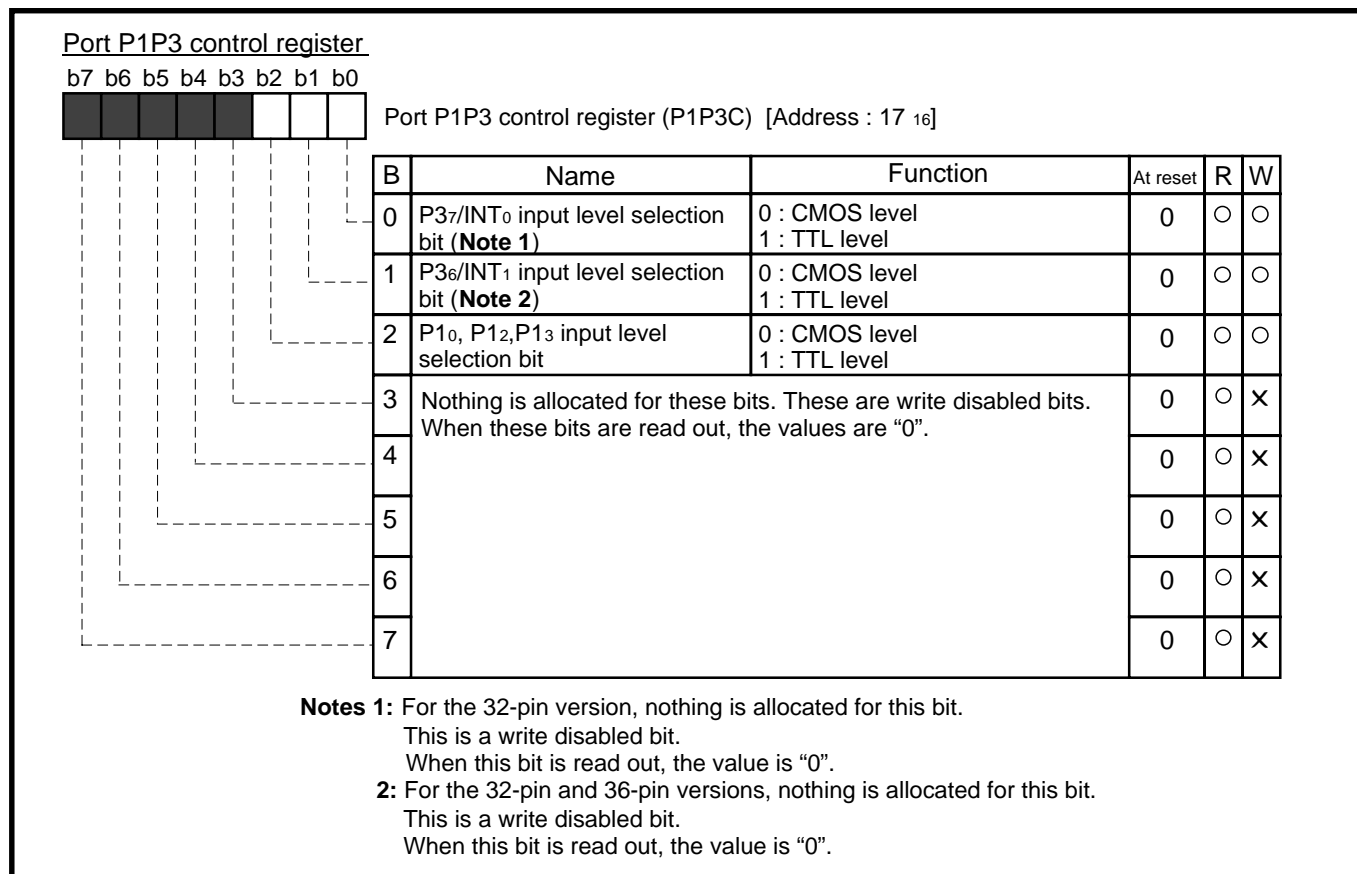


Fig. 2.1.5 Structure of P1P3 control register

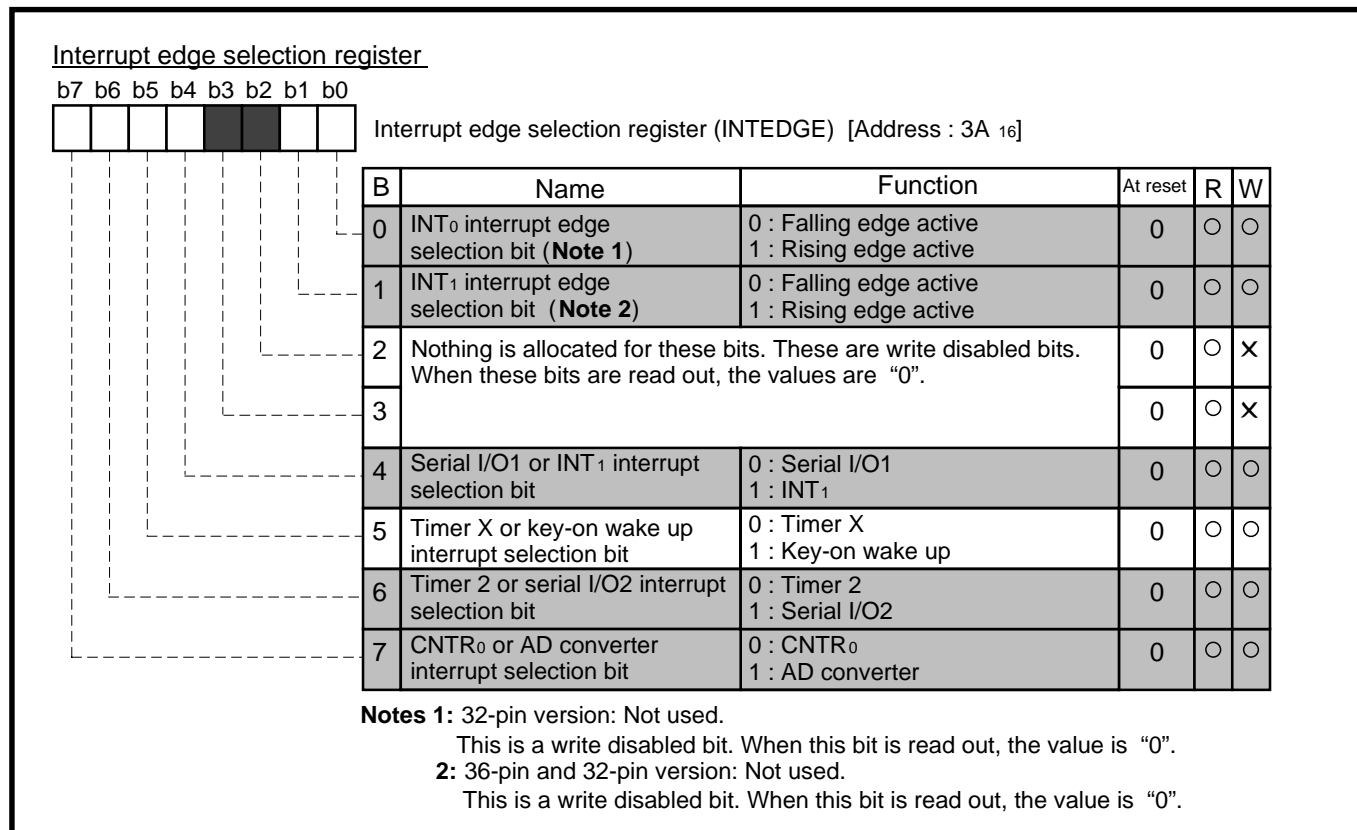


Fig. 2.1.6 Structure of Interrupt edge selection register

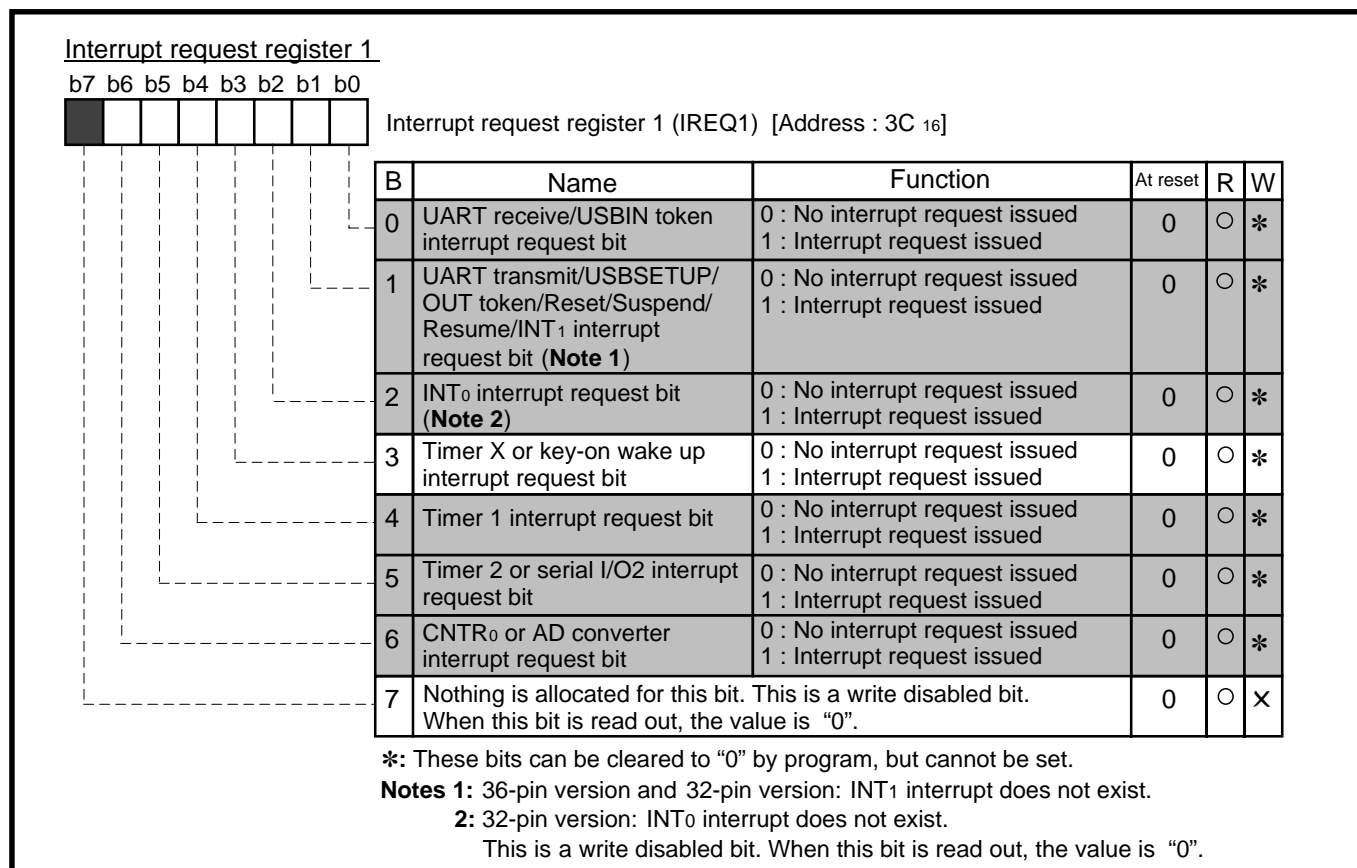


Fig. 2.1.7 Structure of Interrupt request register 1

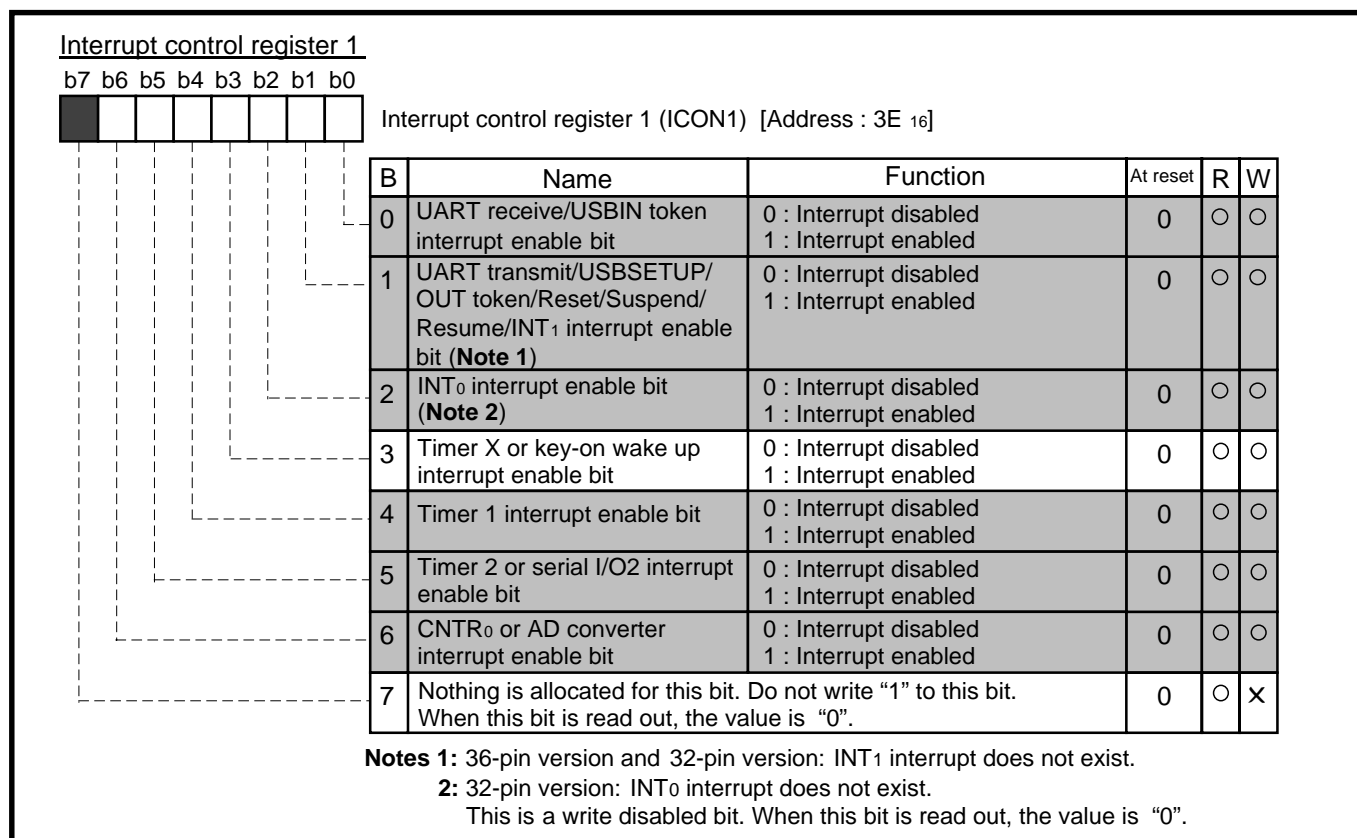


Fig. 2.1.8 Structure of Interrupt control register 1

2.1.3 Application example of key-on wake up

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

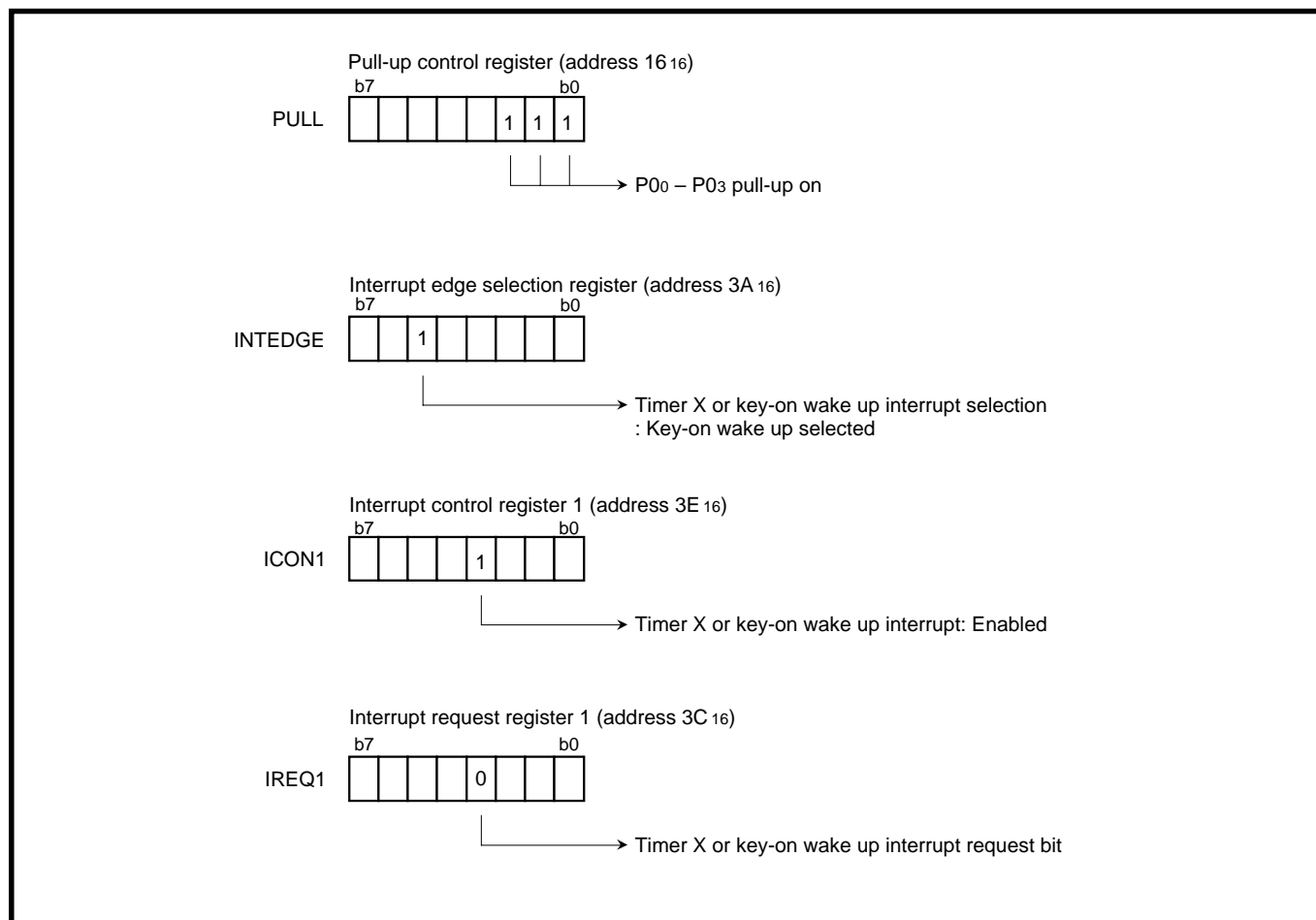


Fig. 2.1.9 Relevant registers setting

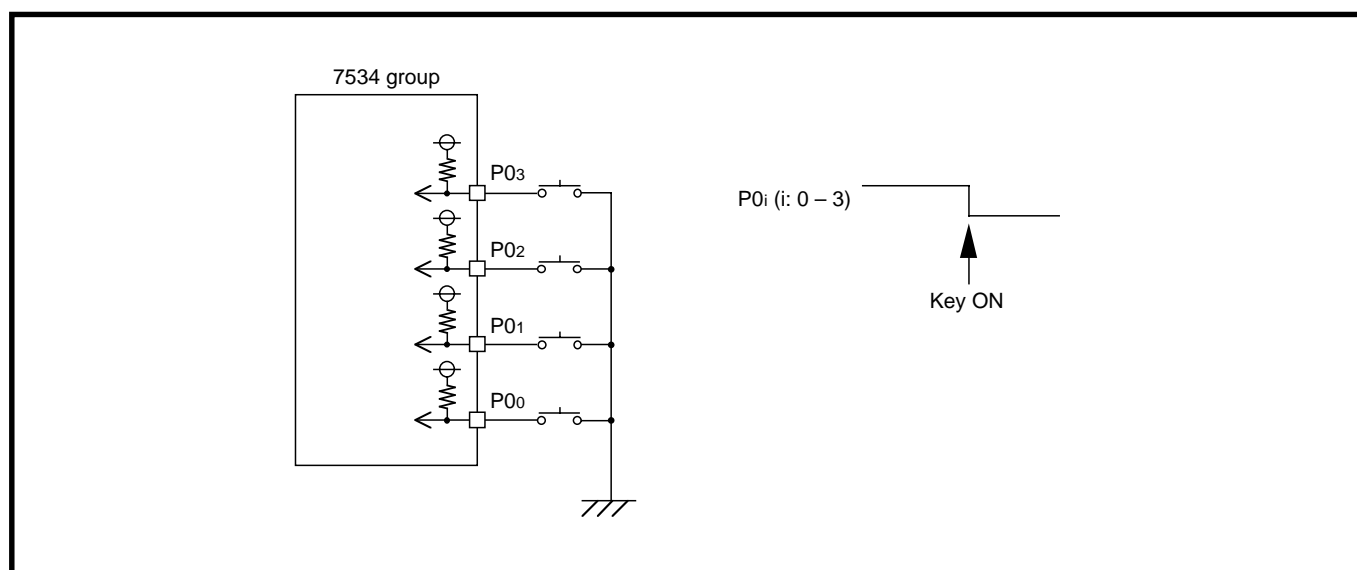


Fig. 2.1.10 Application circuit example

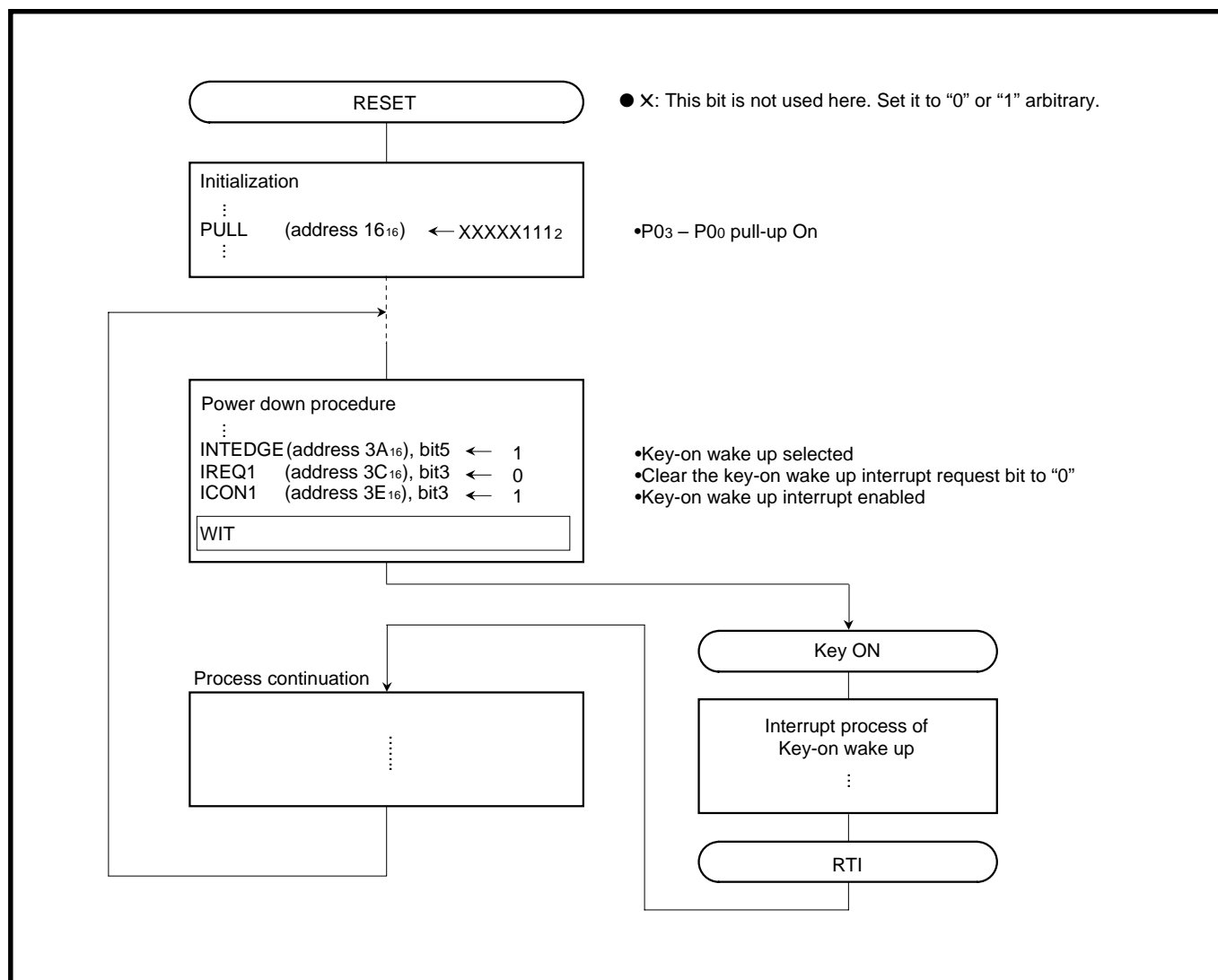


Fig. 2.1.11 Control procedure

2.1.4 Handling of unused pins

Table 2.1.1 Handling of unused pins

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

2.1.5 Notes on input and output pins

(1) Notes in stand-by state

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

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.

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

(2) Modifying output data with bit managing instruction

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

● Reason

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

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

Note the following :

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

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

2.1.6 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 kΩ to 10 kΩ.

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 kΩ to 10 kΩ.

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.

2.2 Timer

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

2.2.1 Memory map

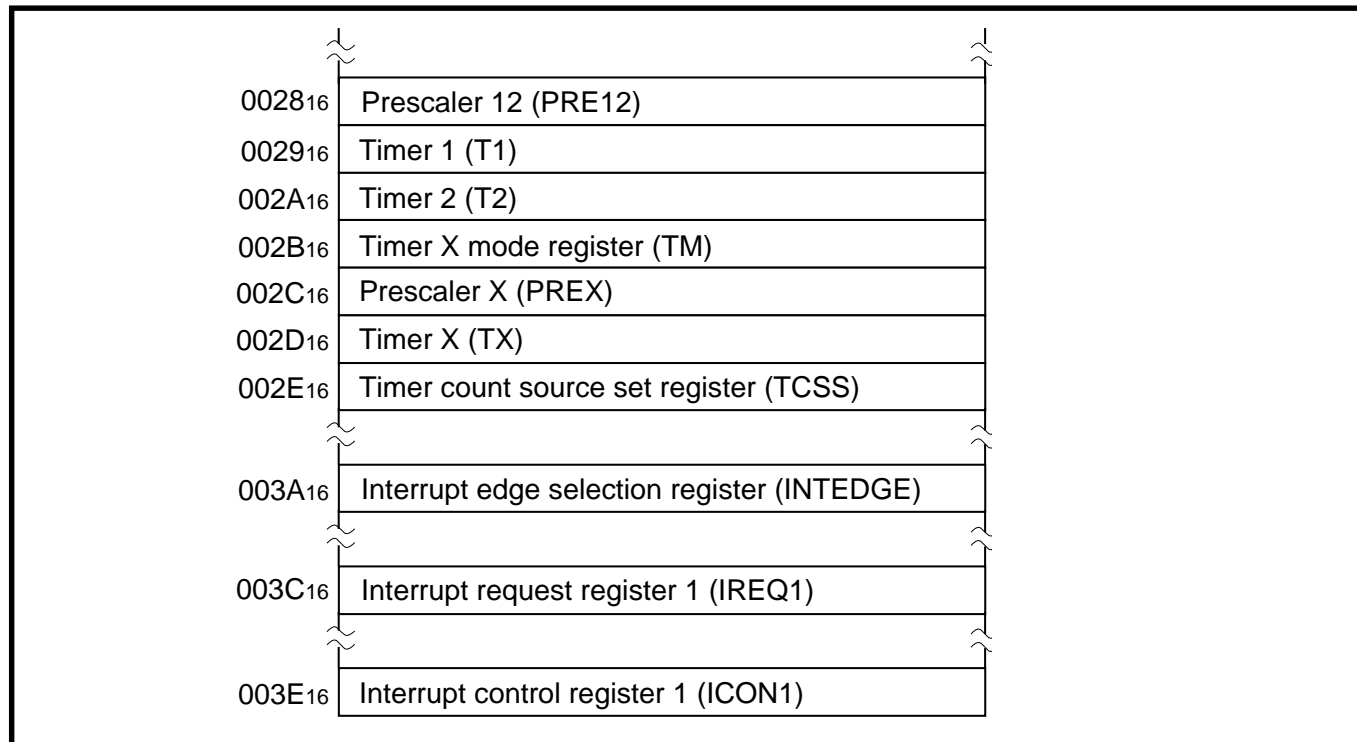


Fig. 2.2.1 Memory map of registers relevant to timers

2.2.2 Relevant registers

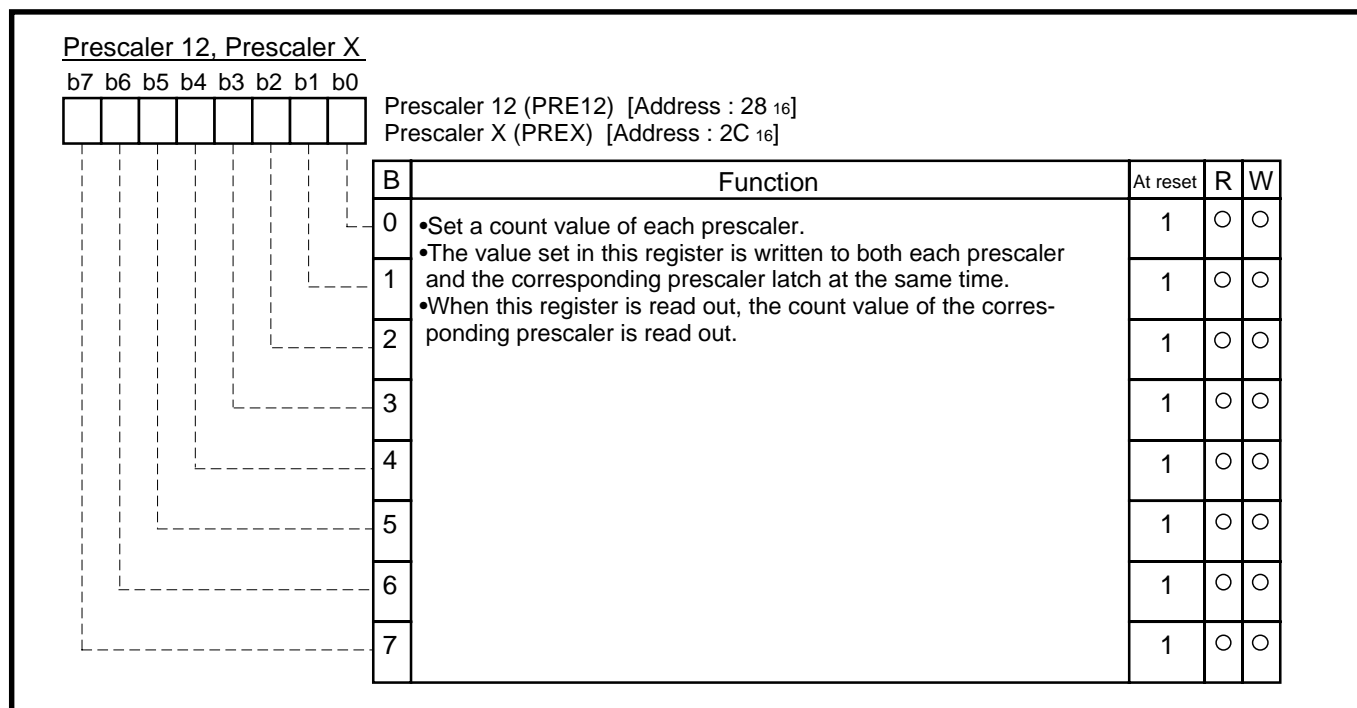


Fig. 2.2.2 Structure of Prescaler 12, Prescaler X

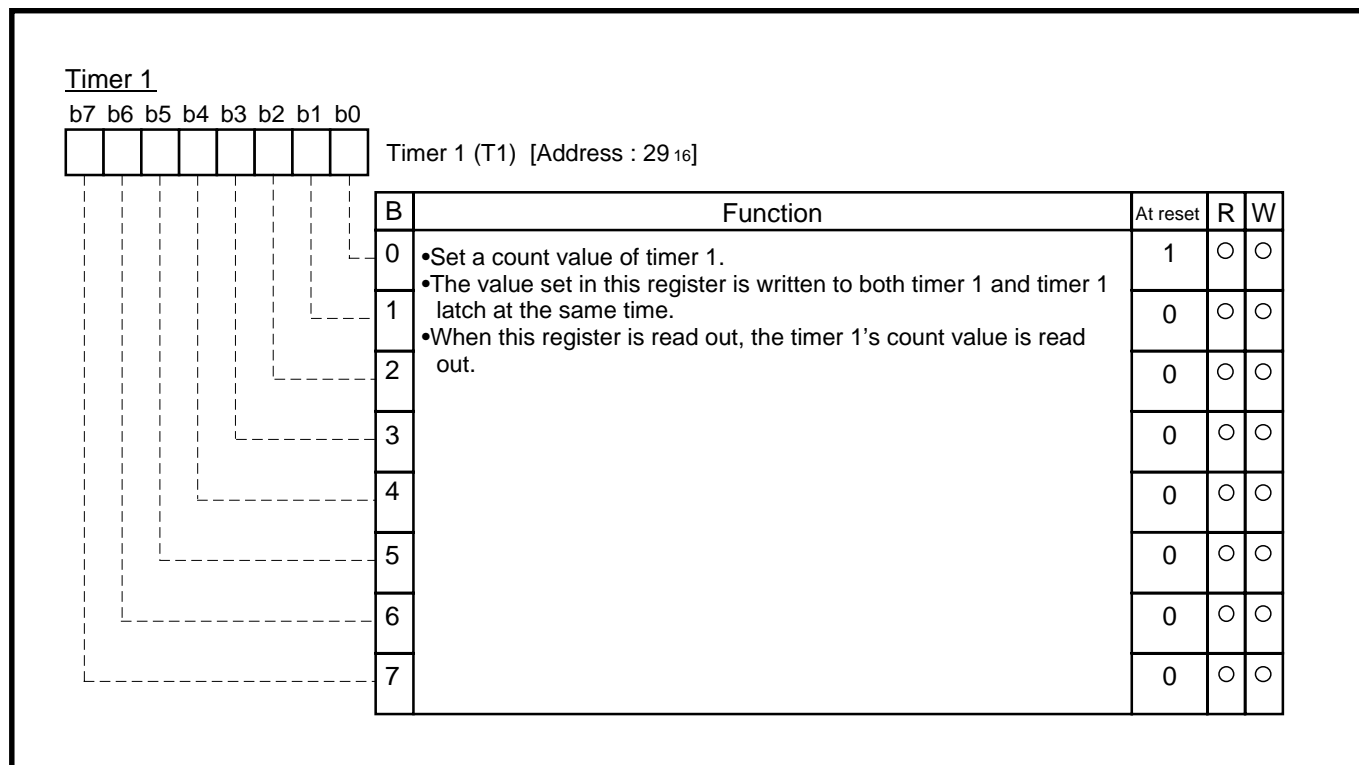


Fig. 2.2.3 Structure of Timer 1

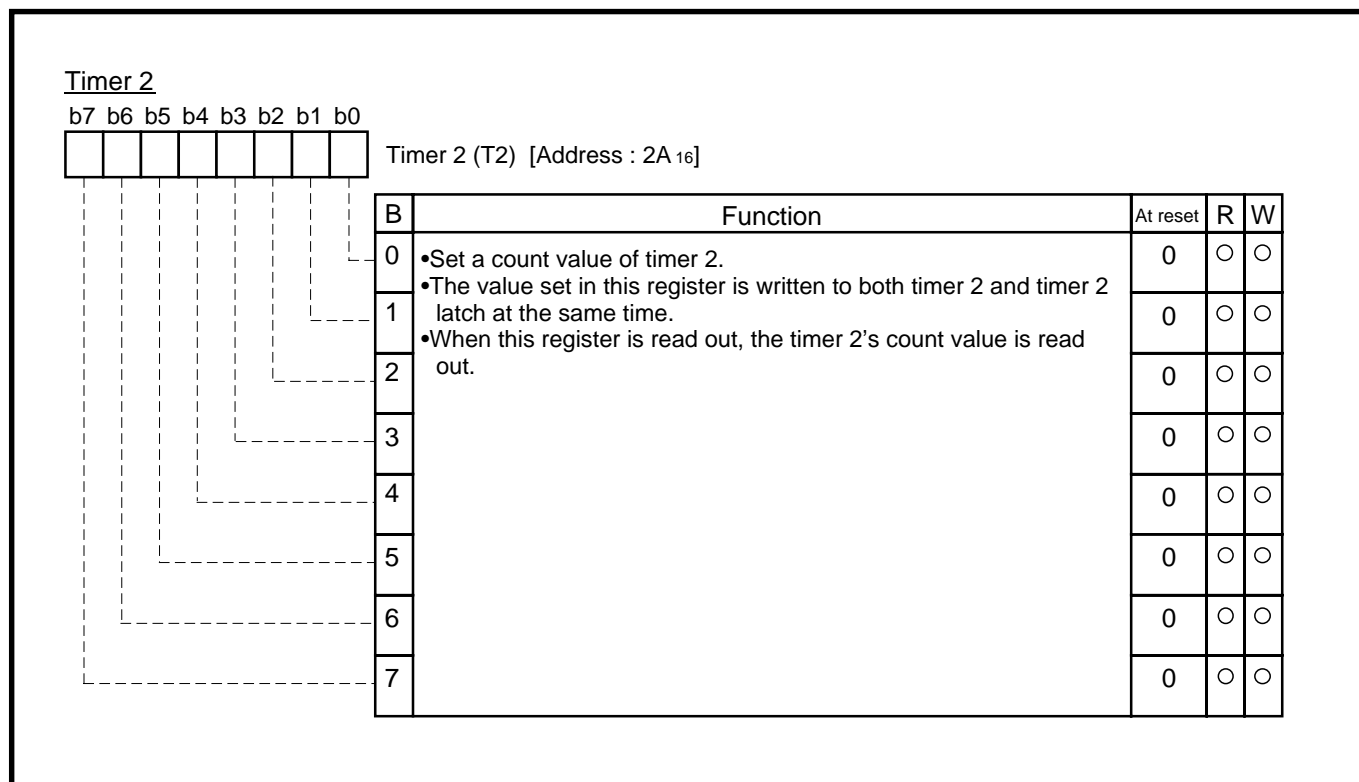


Fig. 2.2.4 Structure of Timer 2

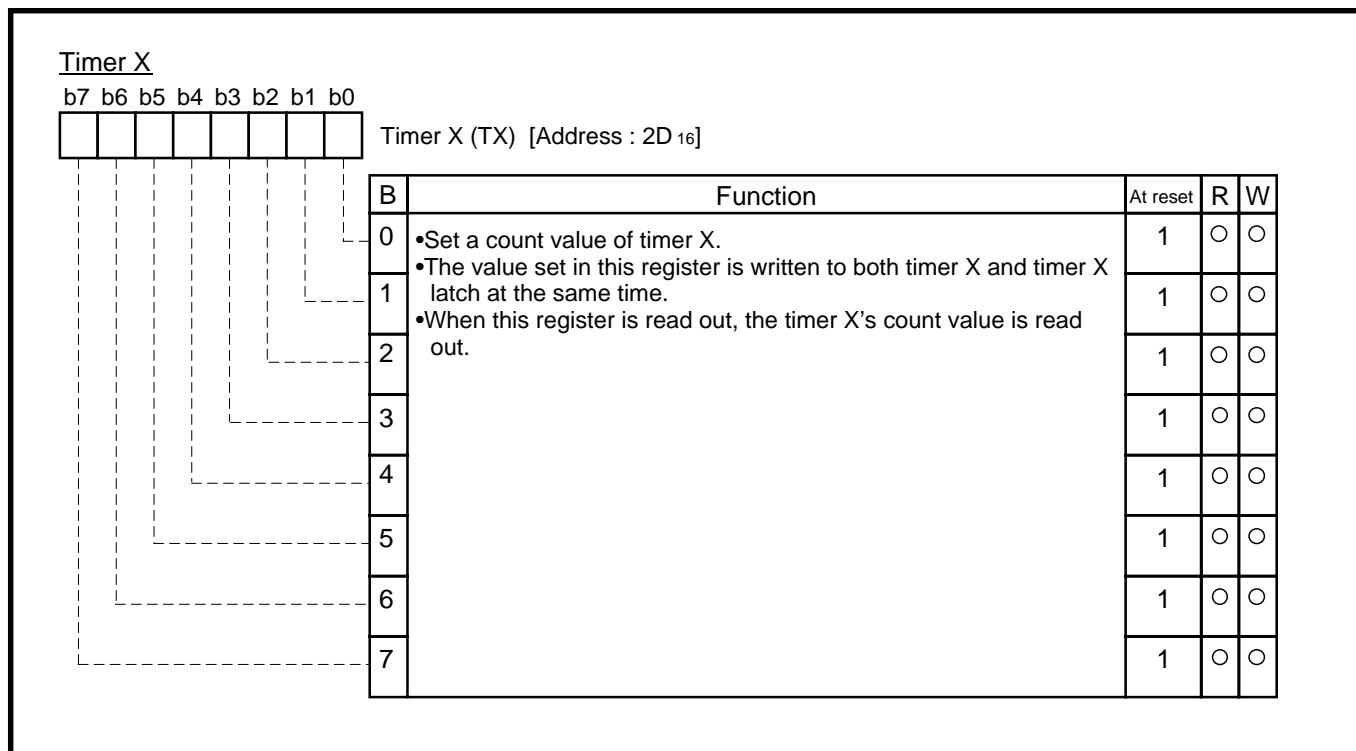


Fig. 2.2.5 Structure of Timer X

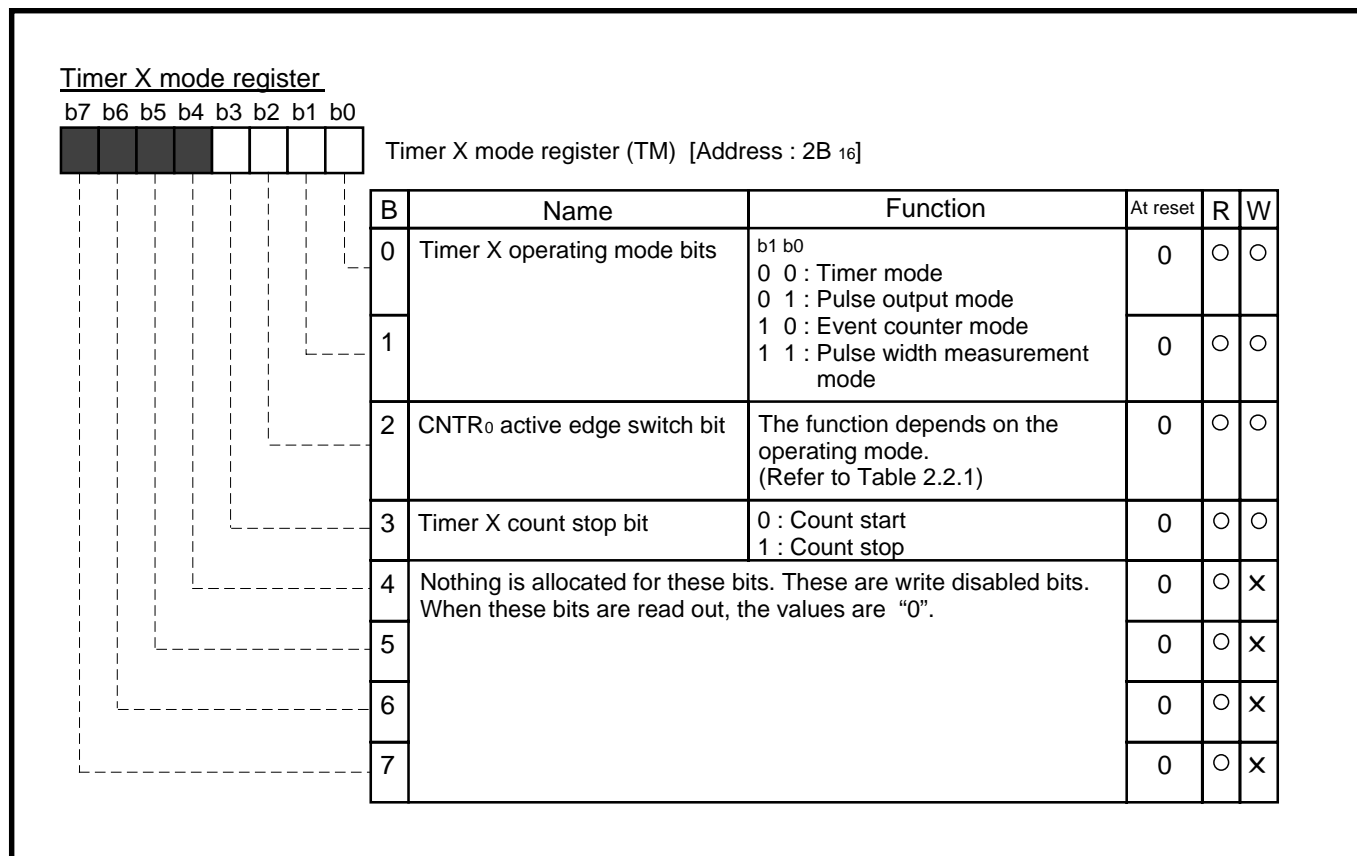


Fig. 2.2.6 Structure of Timer X mode register

Table 2.2.1 CNTR₀ active edge switch bit function

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

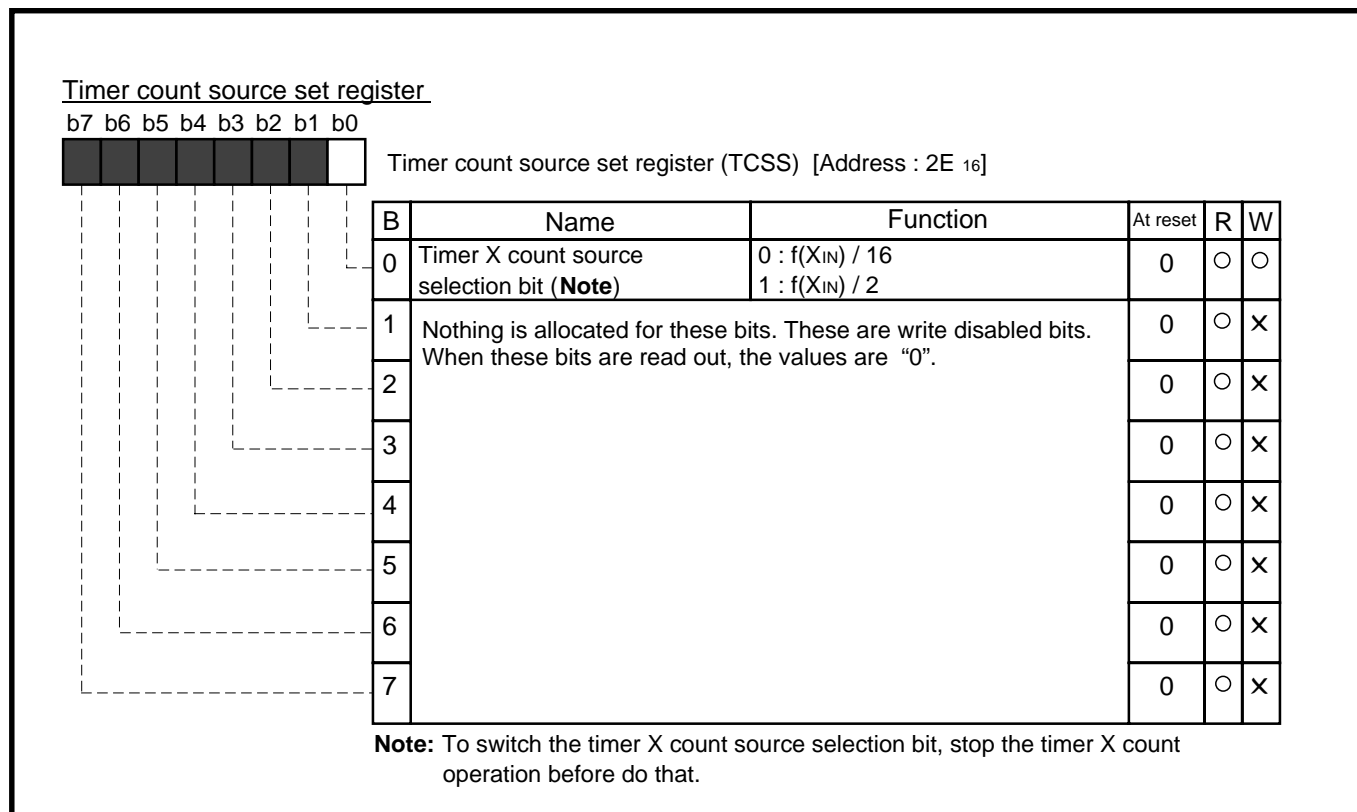


Fig. 2.2.7 Structure of Timer count source set register

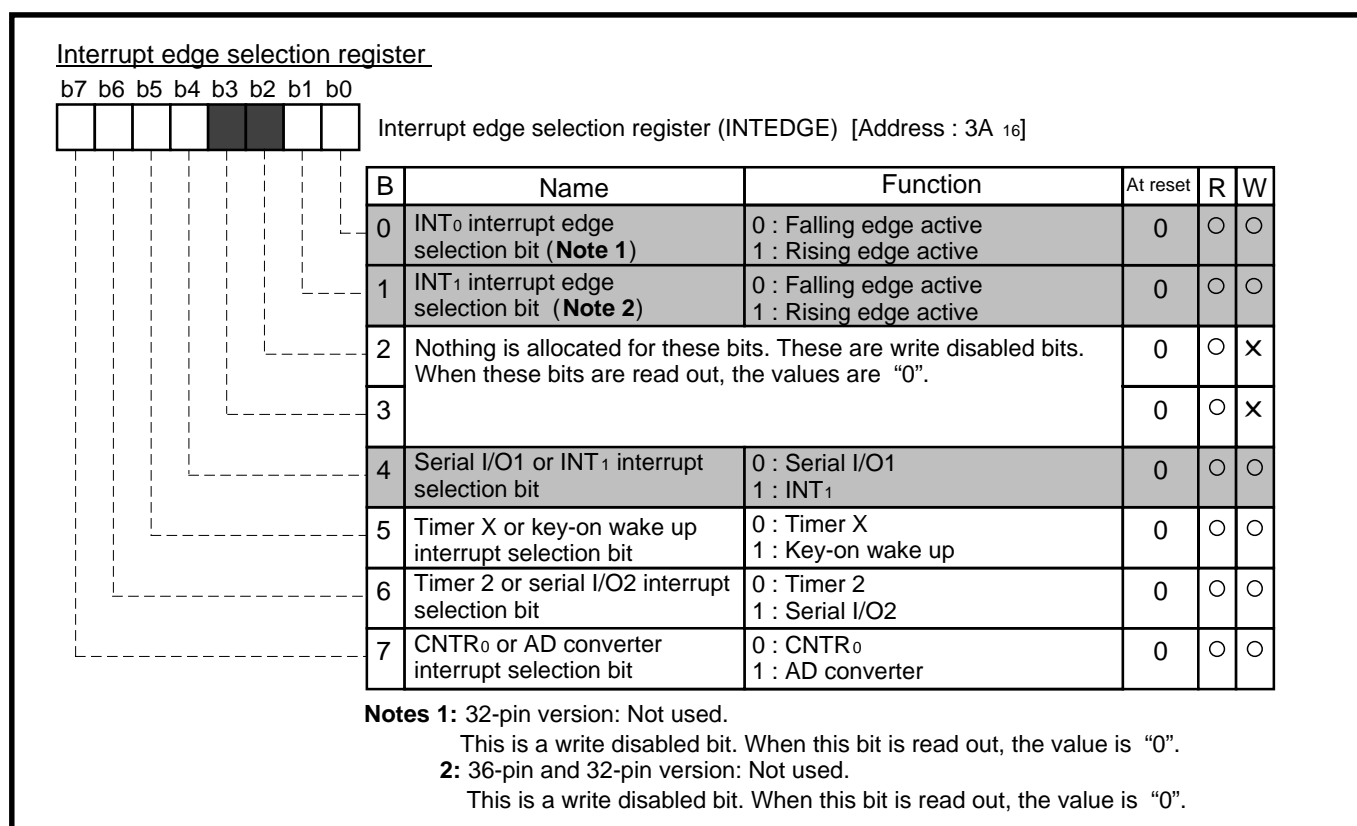


Fig. 2.2.8 Structure of Interrupt edge selection register

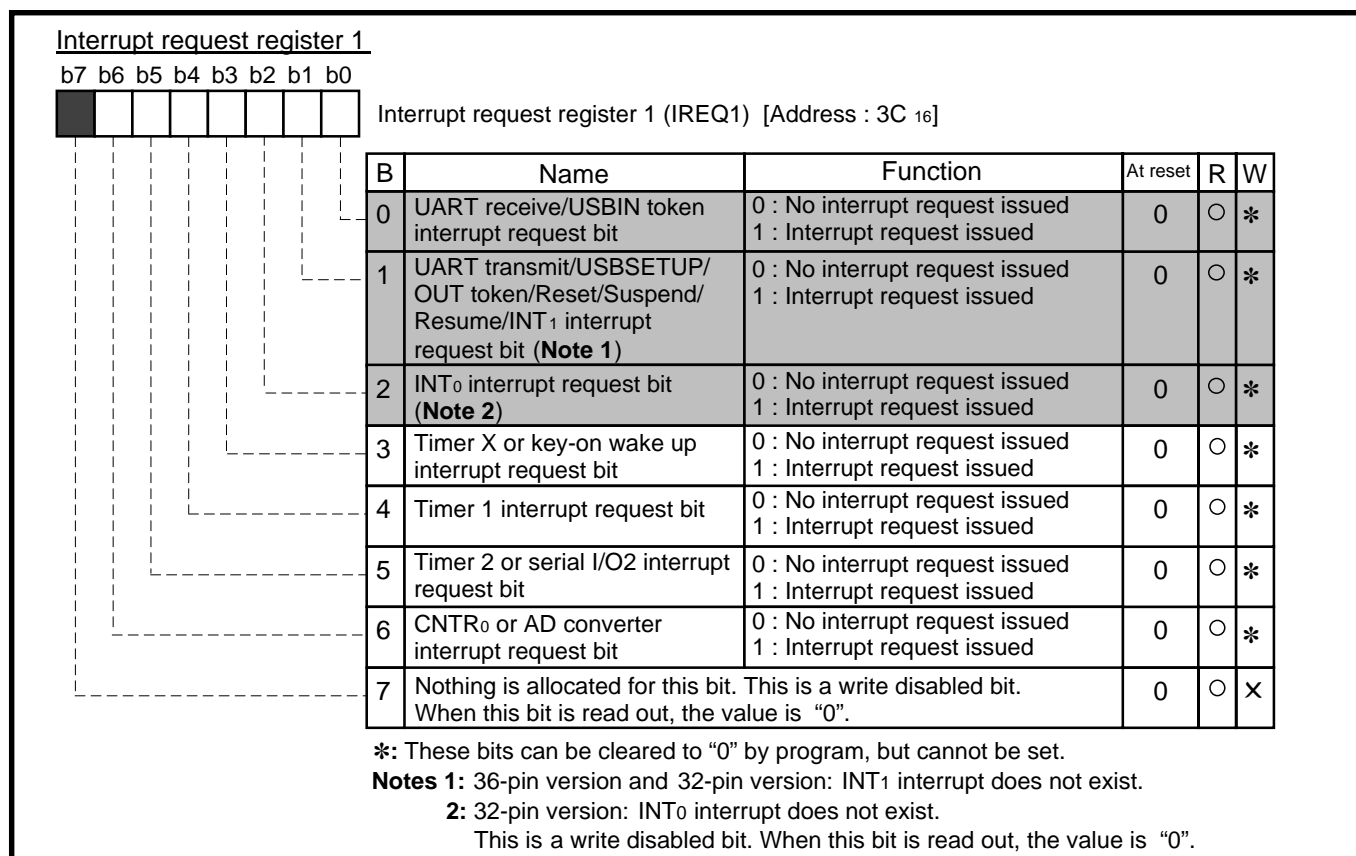


Fig. 2.2.9 Structure of Interrupt request register 1

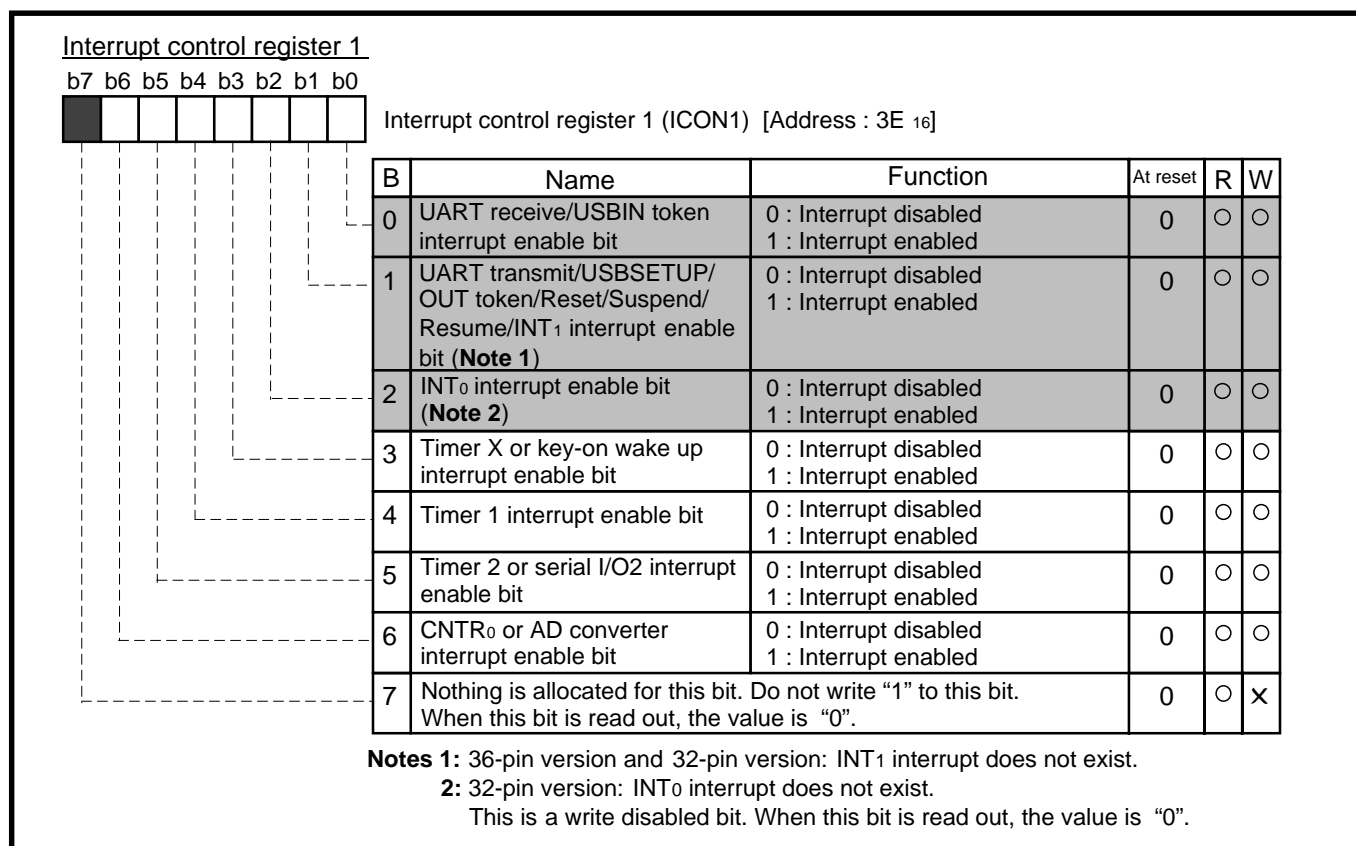


Fig. 2.2.10 Structure of Interrupt control register 1

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 CNTR₀ 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 CNTR₀ 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 CNTR₀ 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.

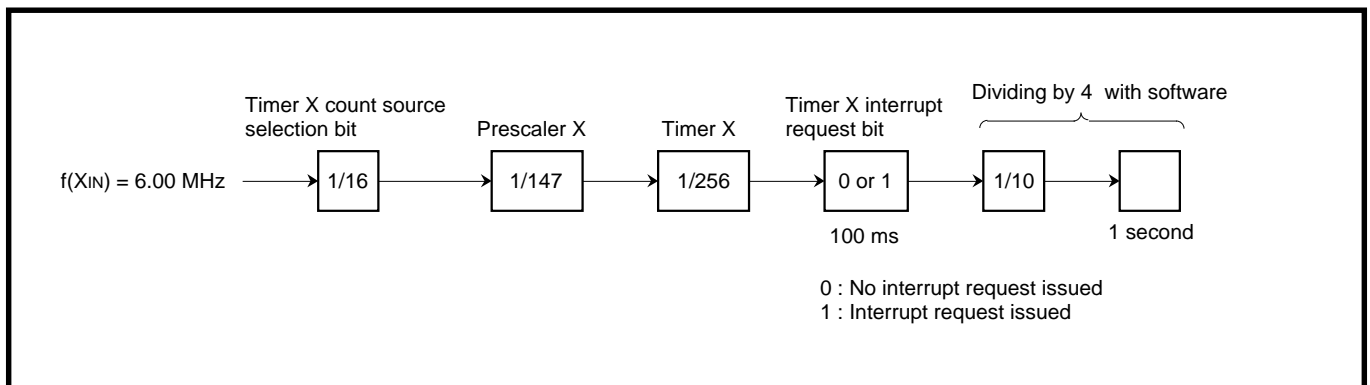


Fig. 2.2.11 Timers connection and setting of division ratios

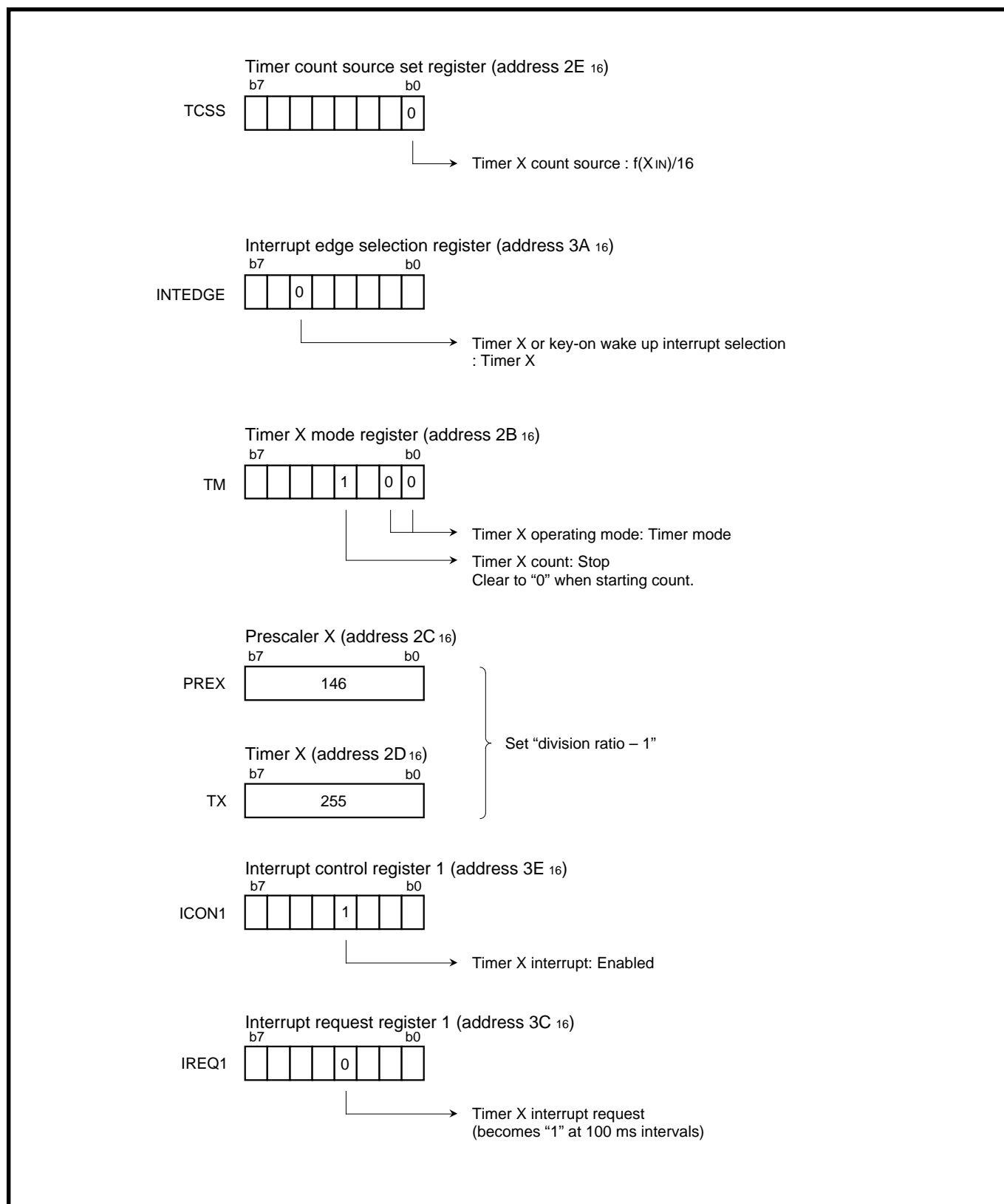


Fig. 2.2.12 Relevant registers setting

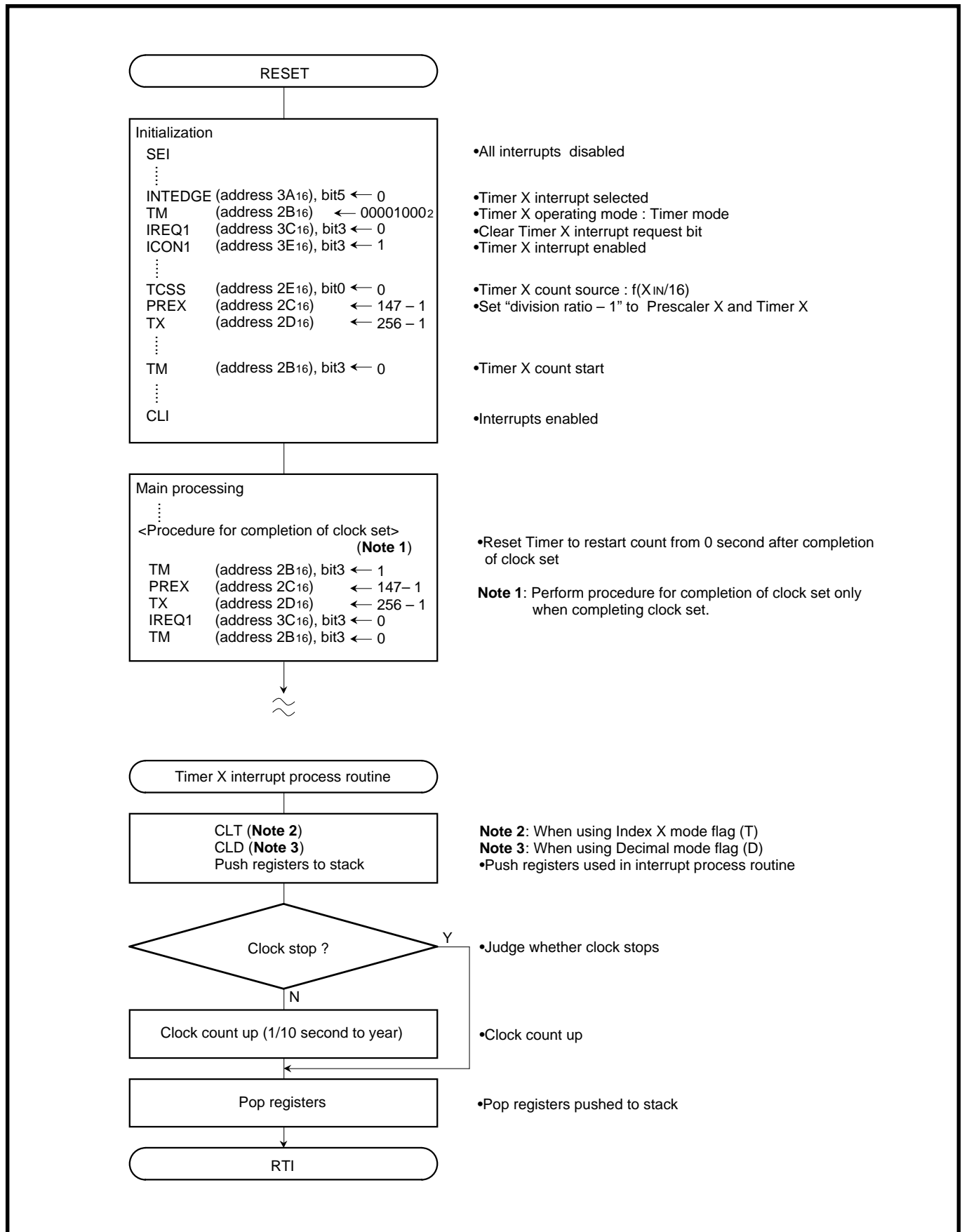


Fig. 2.2.13 Control procedure

(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 $f(X_{IN}) = 6.00$ MHz into about 2 kHz (1995 Hz), is output from the P14/CNTR₀ pin.

•The level of the P14/CNTR₀ pin is fixed to “H” while a piezoelectric buzzer output stops.

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

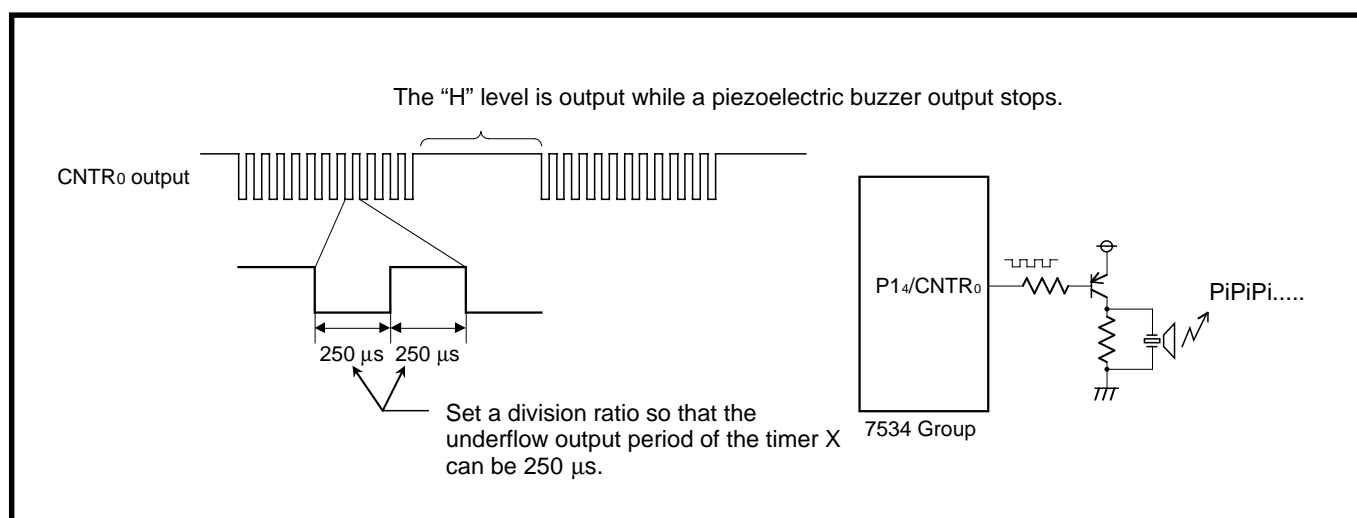


Fig. 2.2.14 Peripheral circuit example

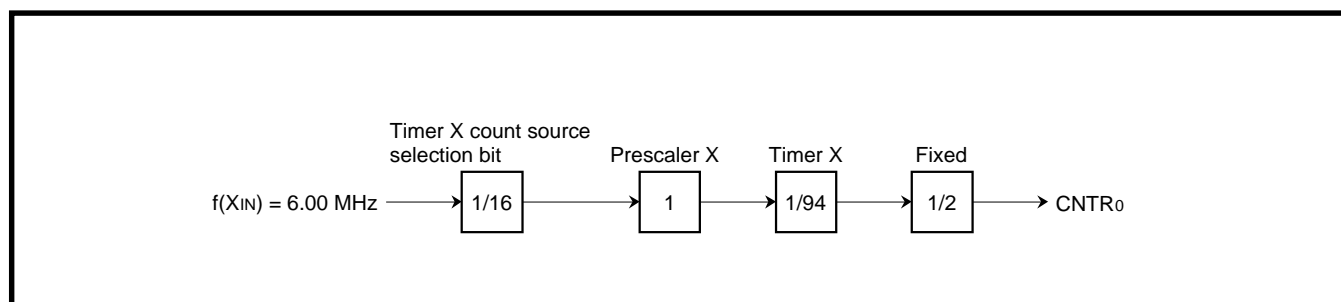


Fig. 2.2.15 Timers connection and setting of division ratios

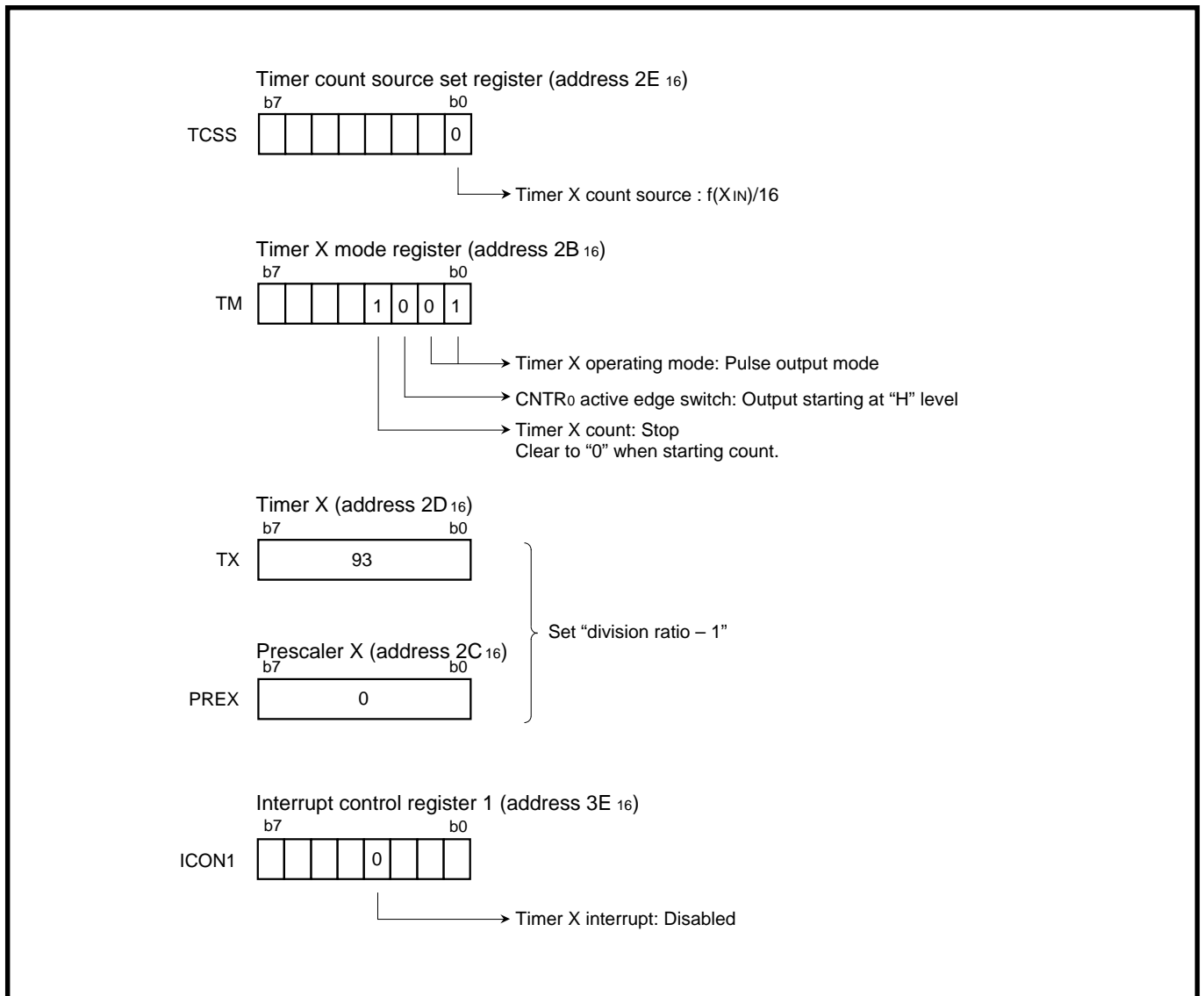


Fig. 2.2.16 Relevant registers setting

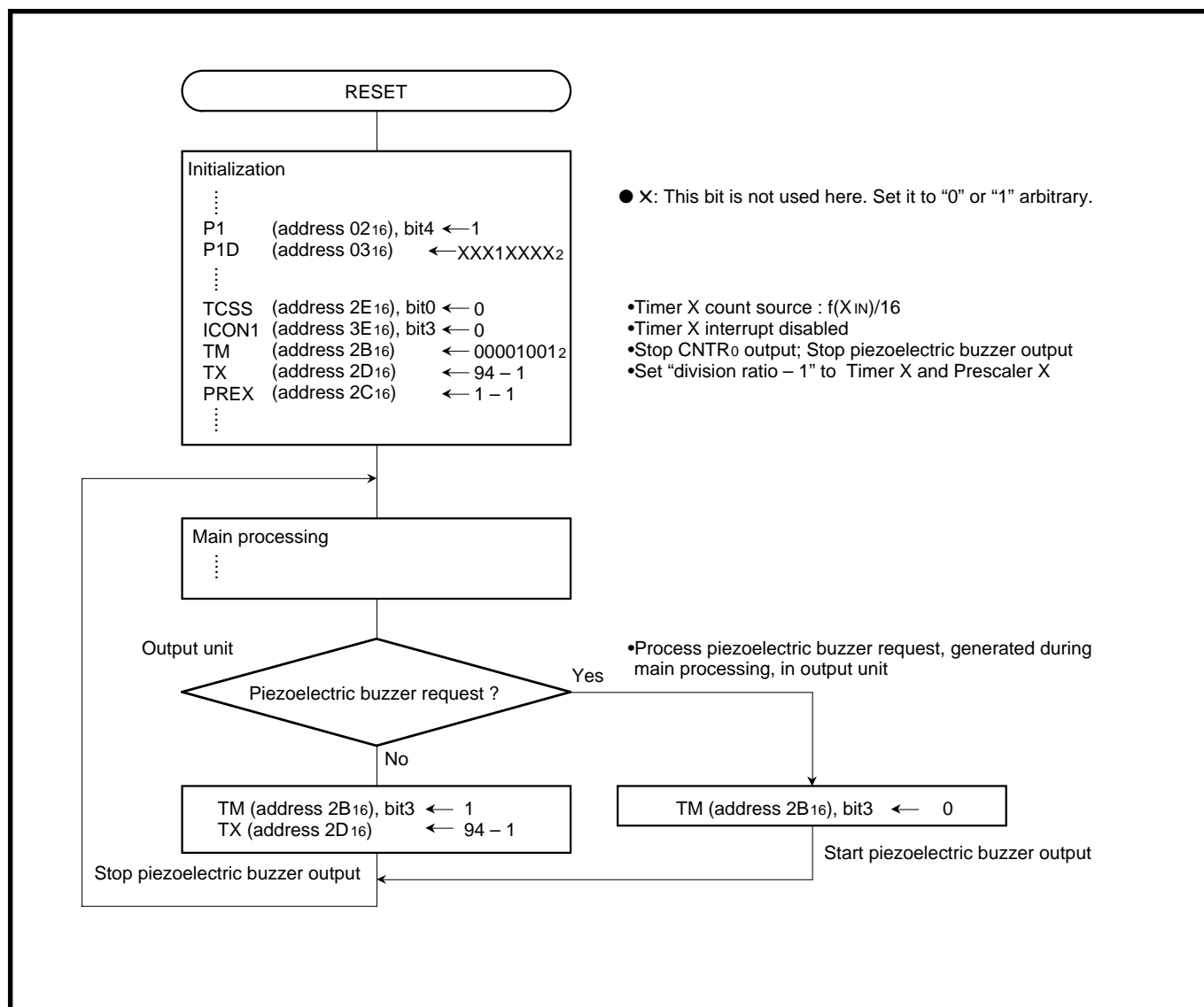


Fig. 2.2.17 Control procedure

(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/CNTR₀ pin with the timer.
- A reference value

Specifications:

- The pulse is input to the P1₄/CNTR₀ 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 – 215 = 255 (initial value of counter) – 28 to 40 (the number of valid count)

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.

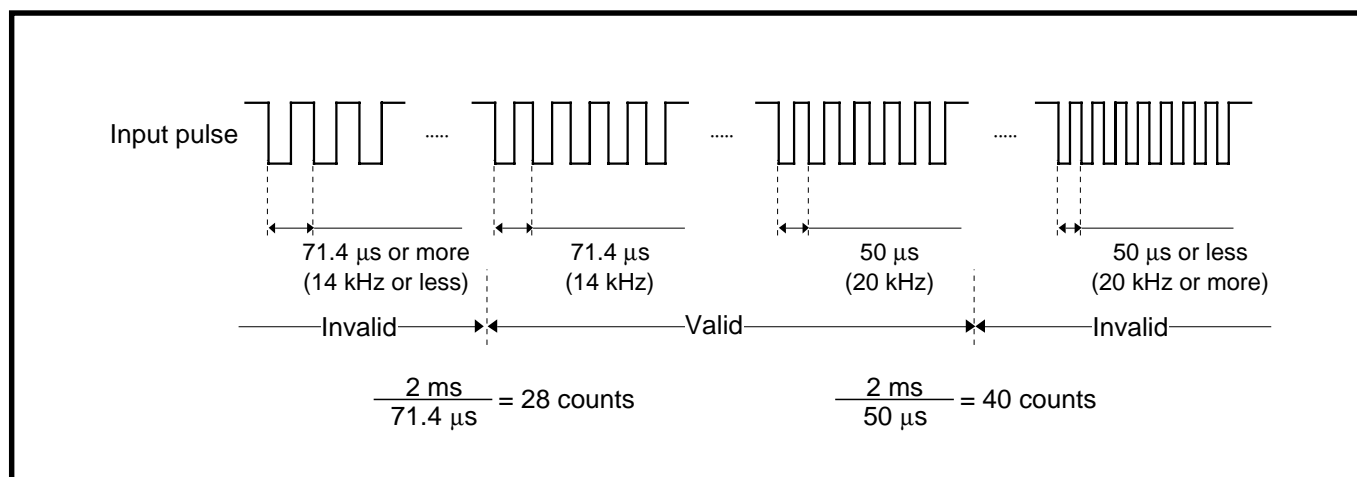


Fig. 2.2.18 Judgment method of valid/invalid of input pulses

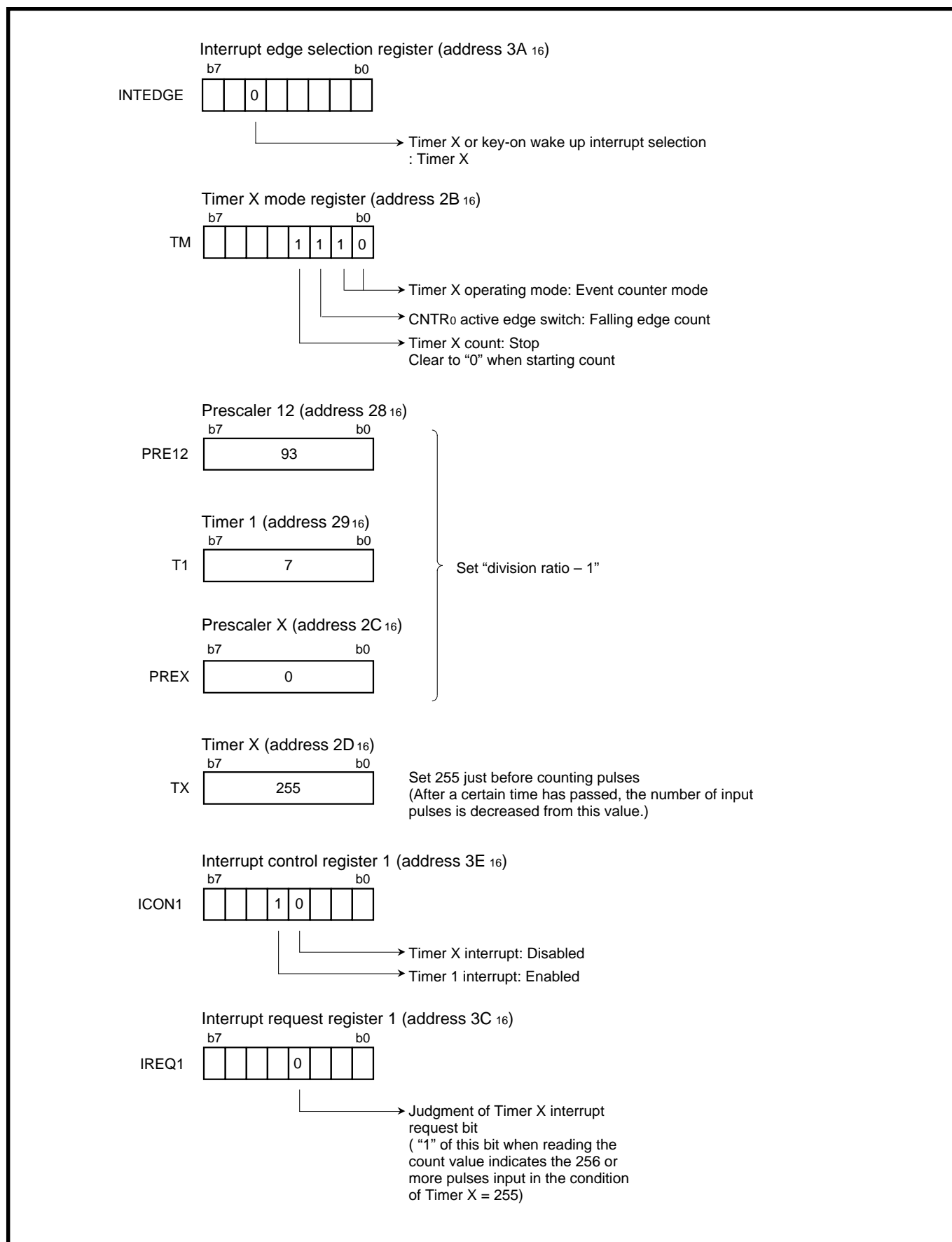


Fig. 2.2.19 Relevant registers setting

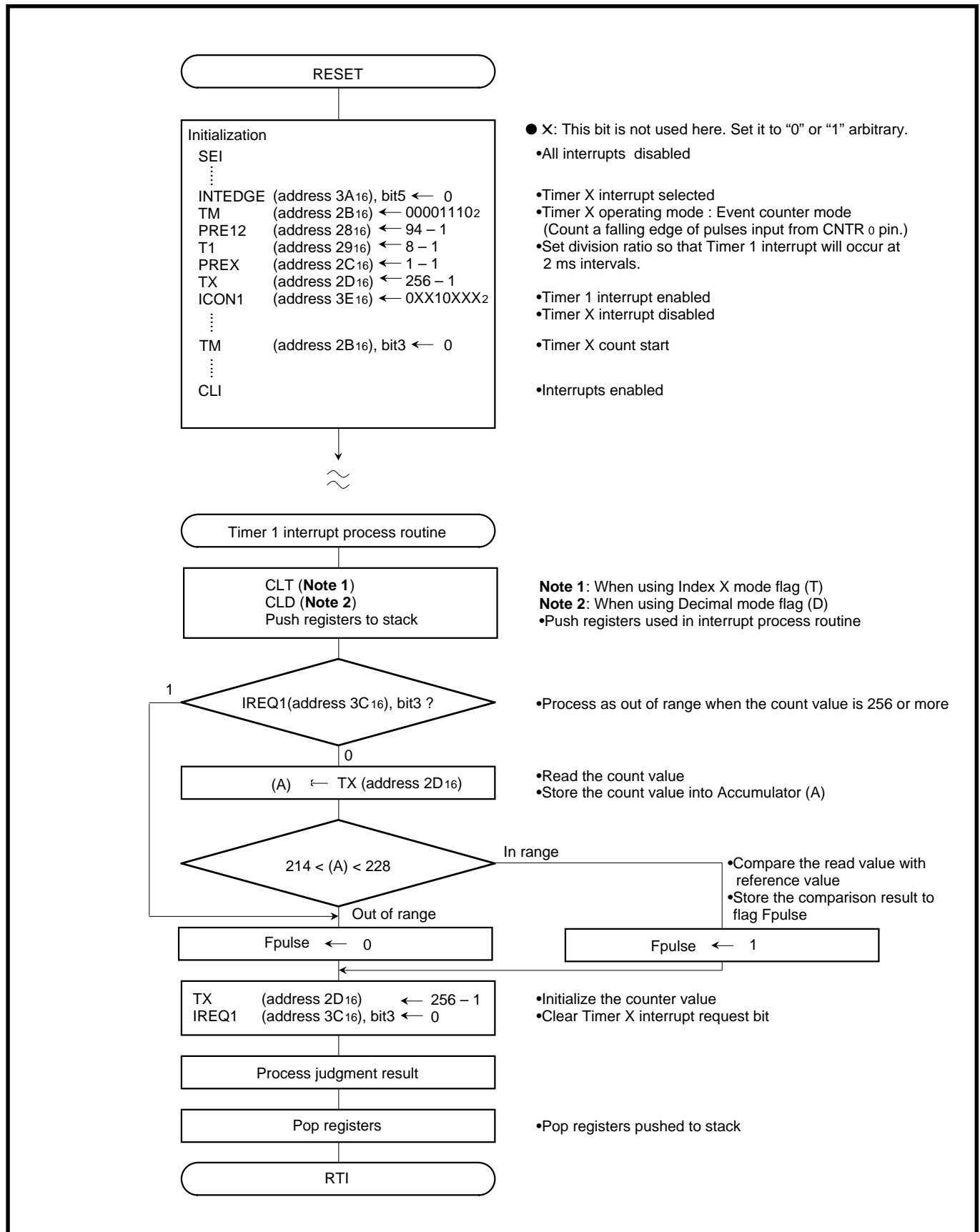


Fig. 2.2.20 Control procedure

(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/CNTR₀ pin. An underflow is detected by the timer X interrupt and an end of the input pulse “H” level is detected by the CNTR₀ interrupt.

Specifications: •The timer X counts the “H” level width of the FG pulse input to the P1₄/CNTR₀ pin.

<Example>

When the clock frequency is 6.00 MHz, the count source is 2.7 μs, which is obtained by dividing the clock frequency by 16. Measurement can be made up to 174 ms in the range of FFFF₁₆ to 0000₁₆.

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

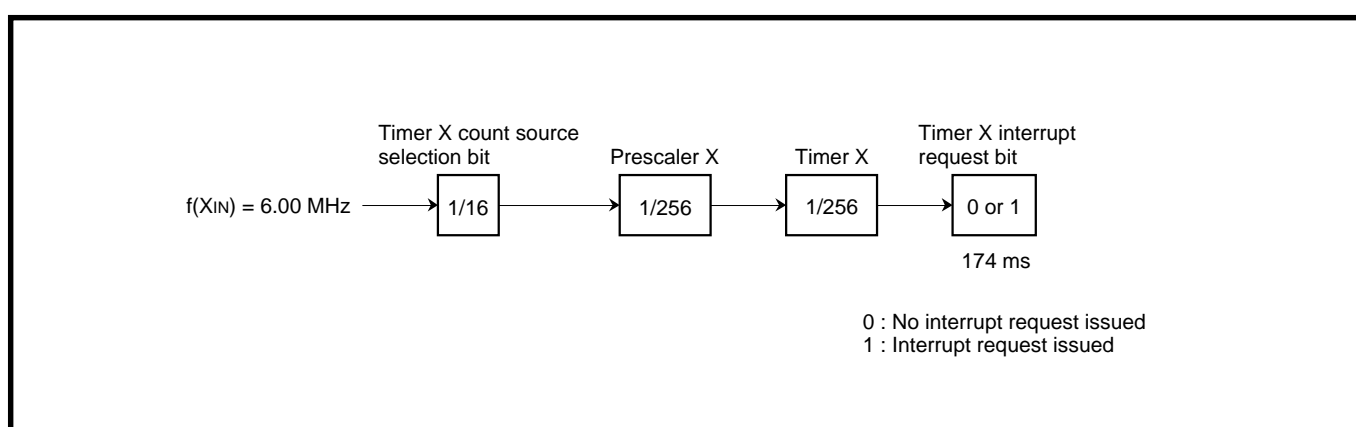


Fig. 2.2.21 Timers connection and setting of division ratios

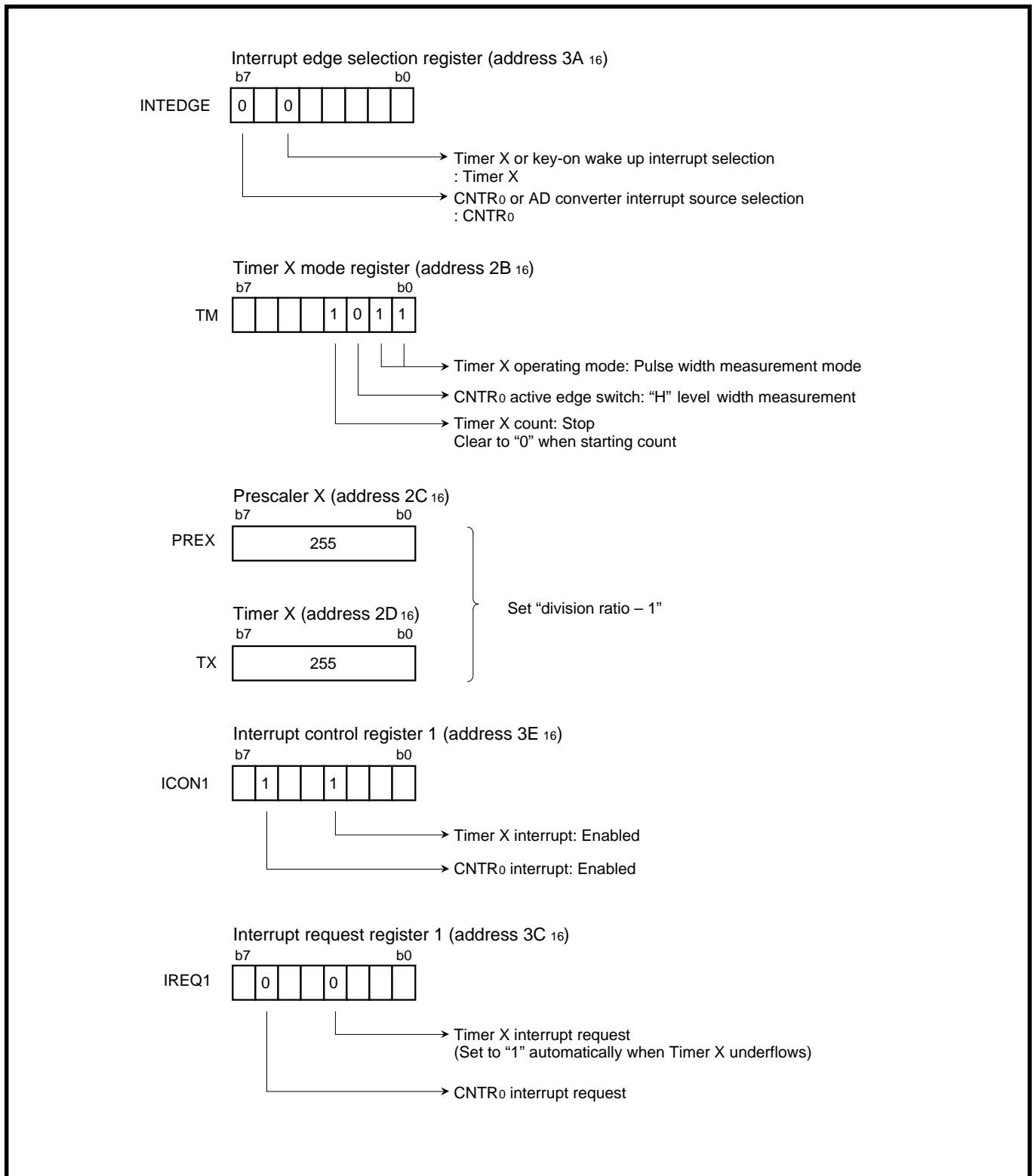


Fig. 2.2.22 Relevant registers setting

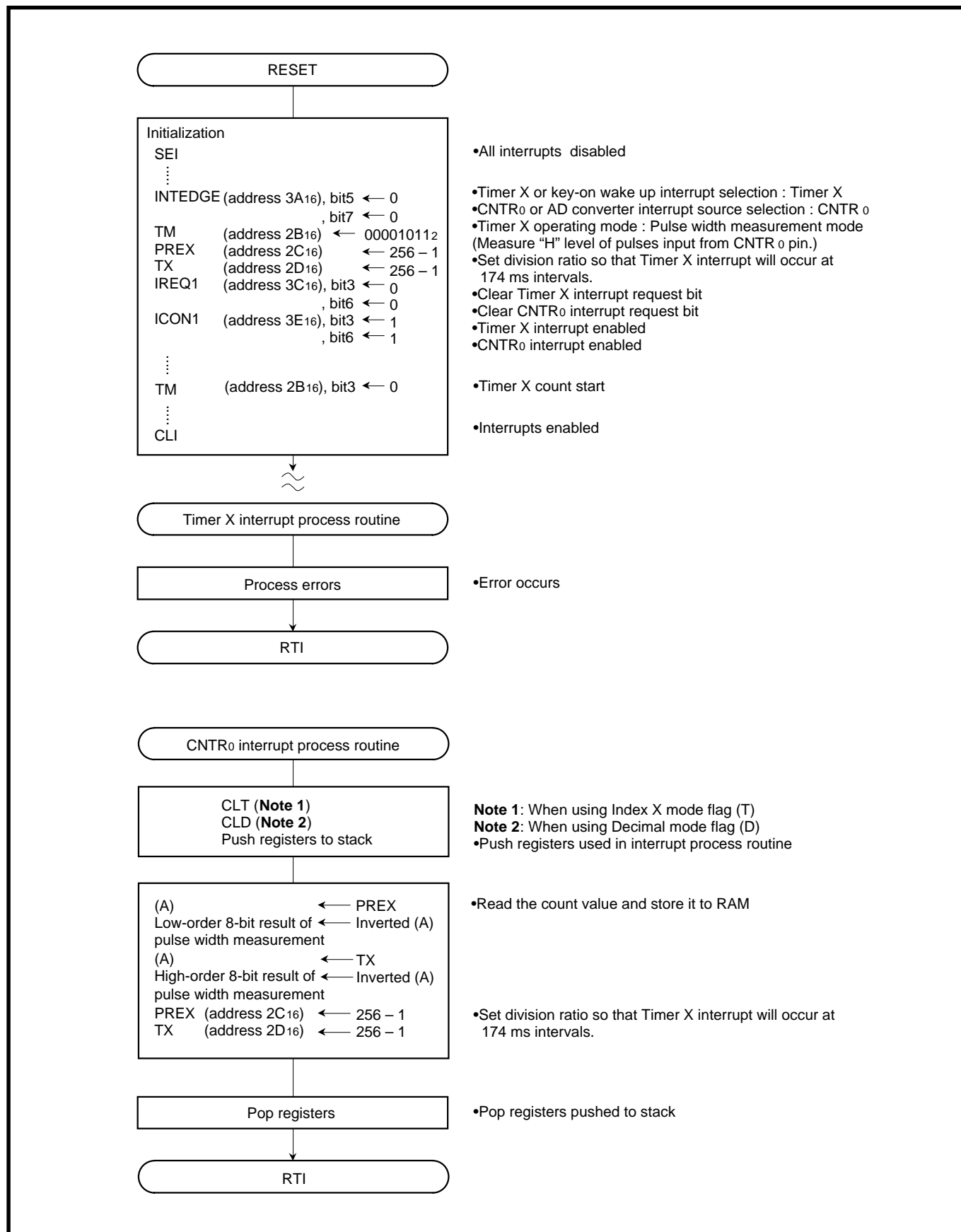


Fig. 2.2.23 Control procedure

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

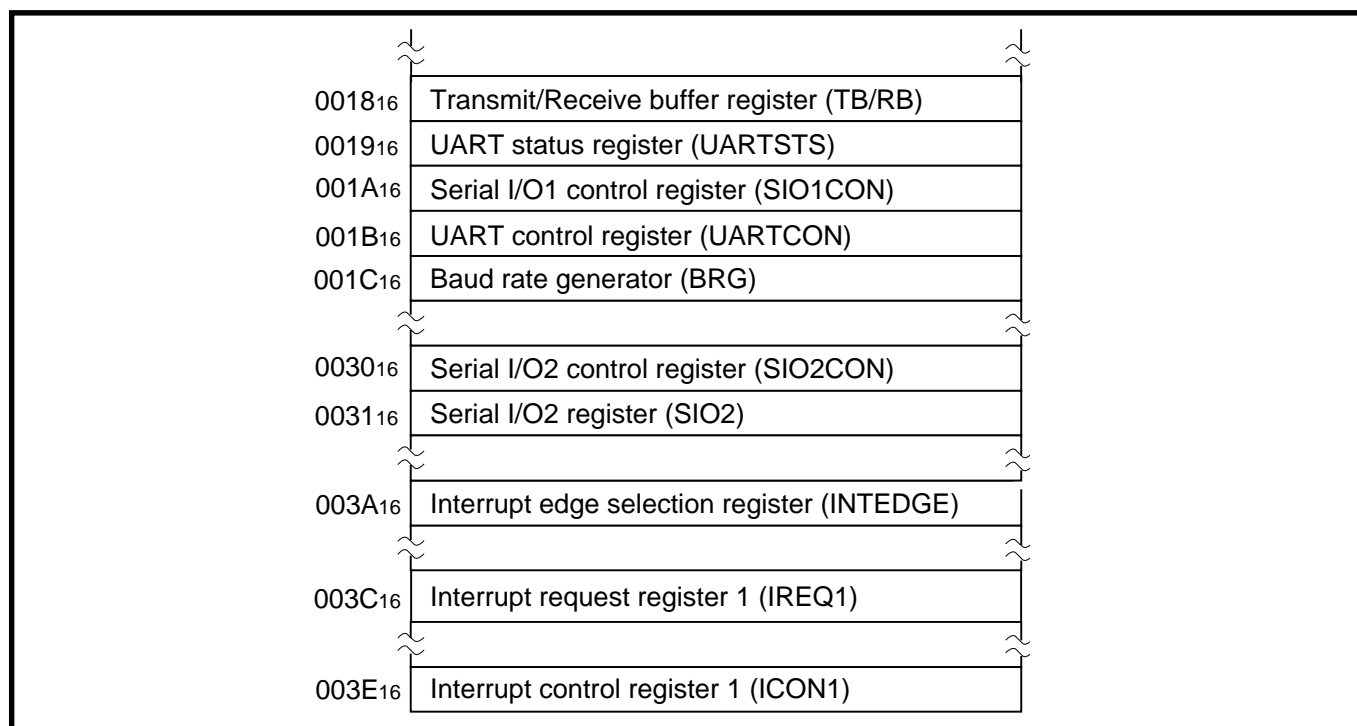


Fig. 2.3.1 Memory map of registers relevant to serial I/O

2.3.2 Relevant registers

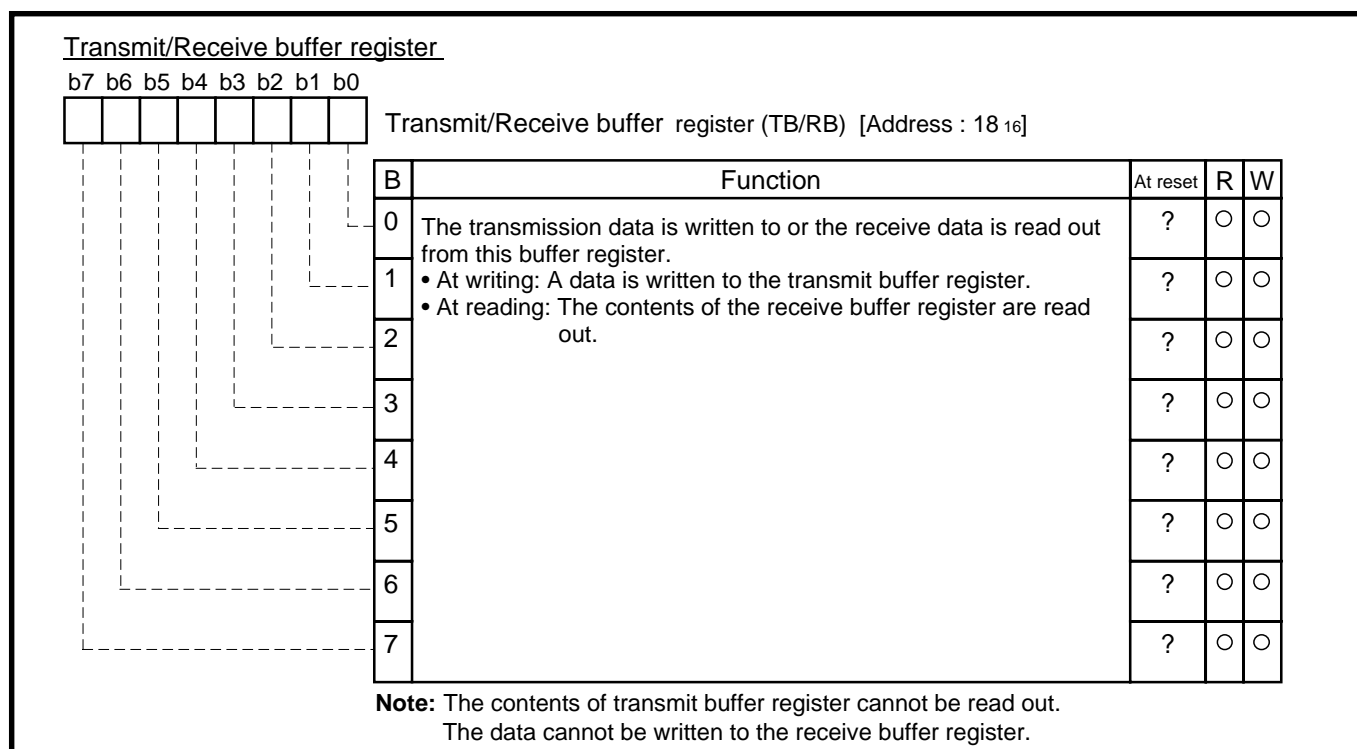


Fig. 2.3.2 Structure of Transmit/Receive buffer register

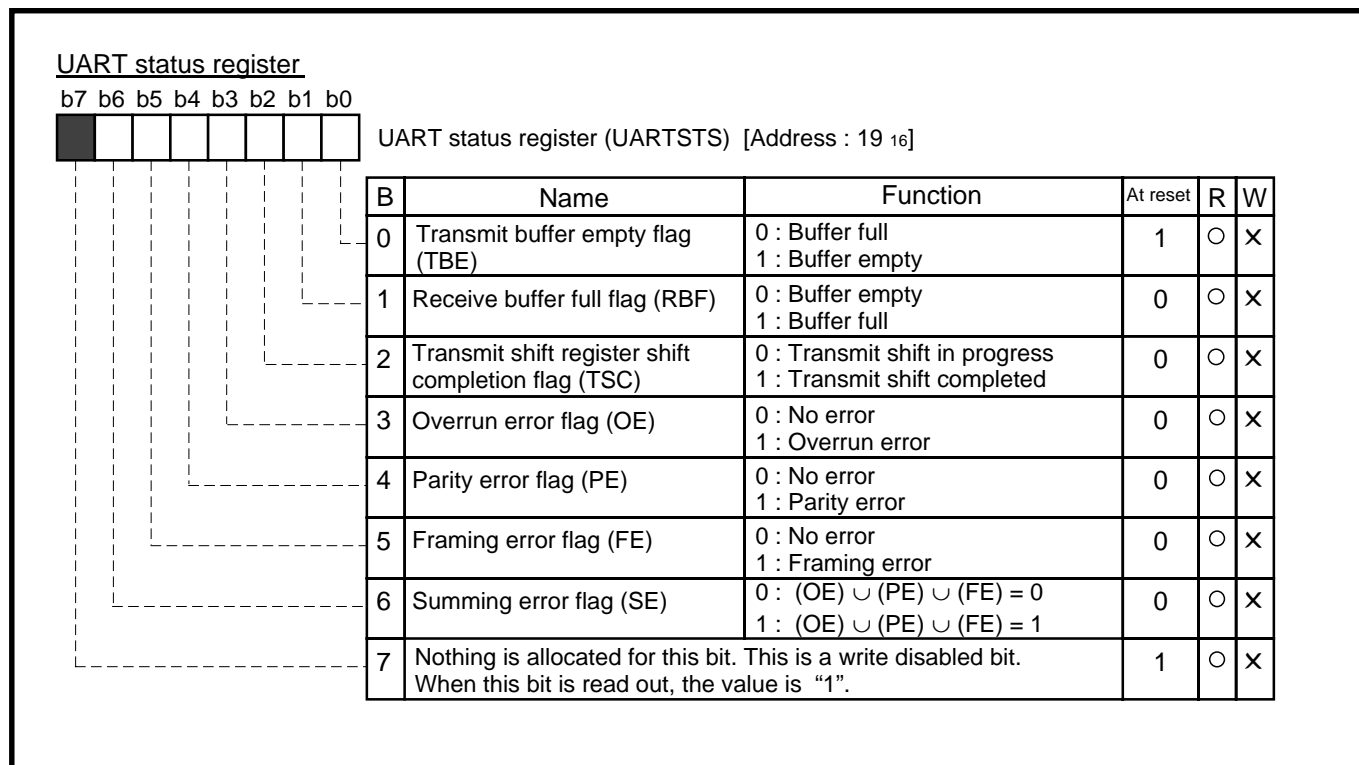


Fig. 2.3.3 Structure of UART status register

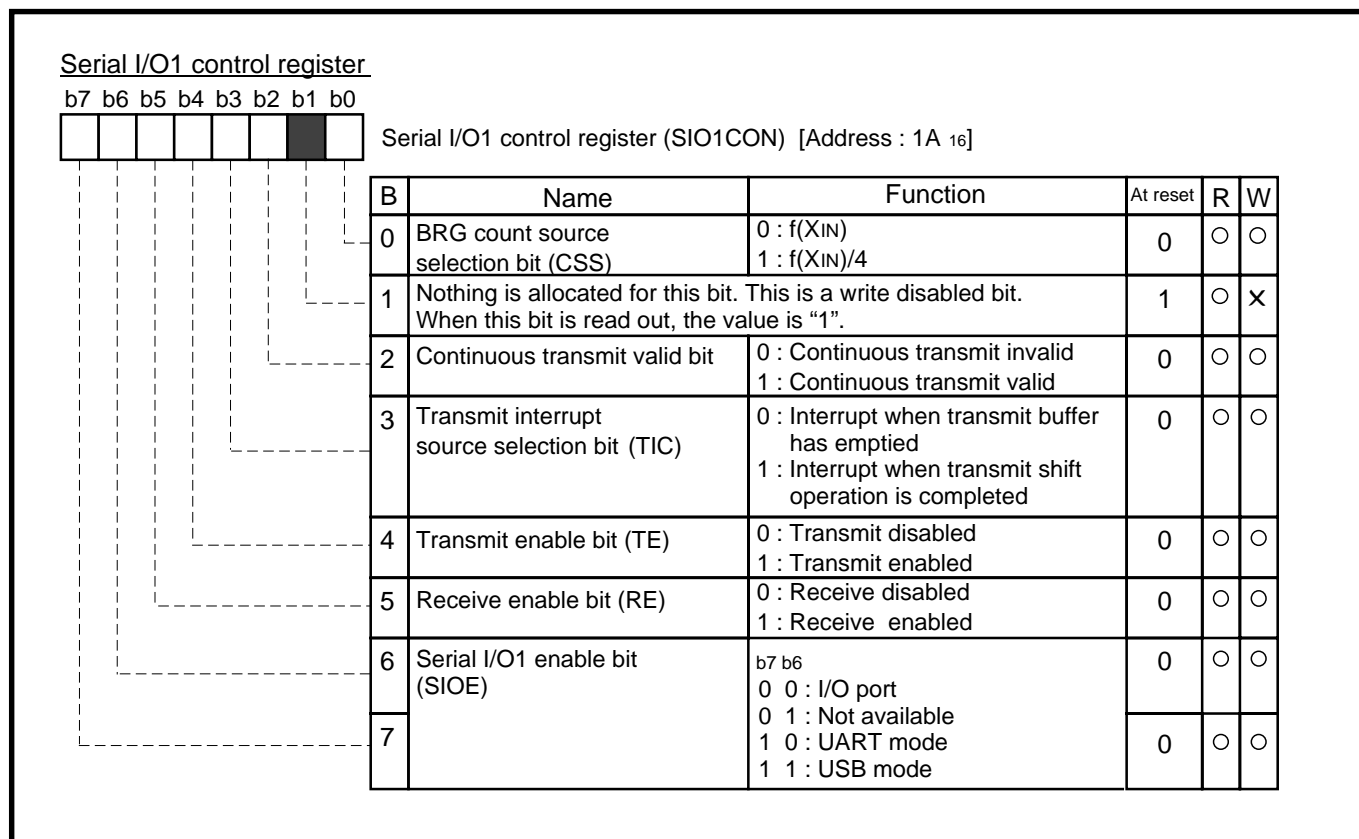


Fig. 2.3.4 Structure of Serial I/O1 control register

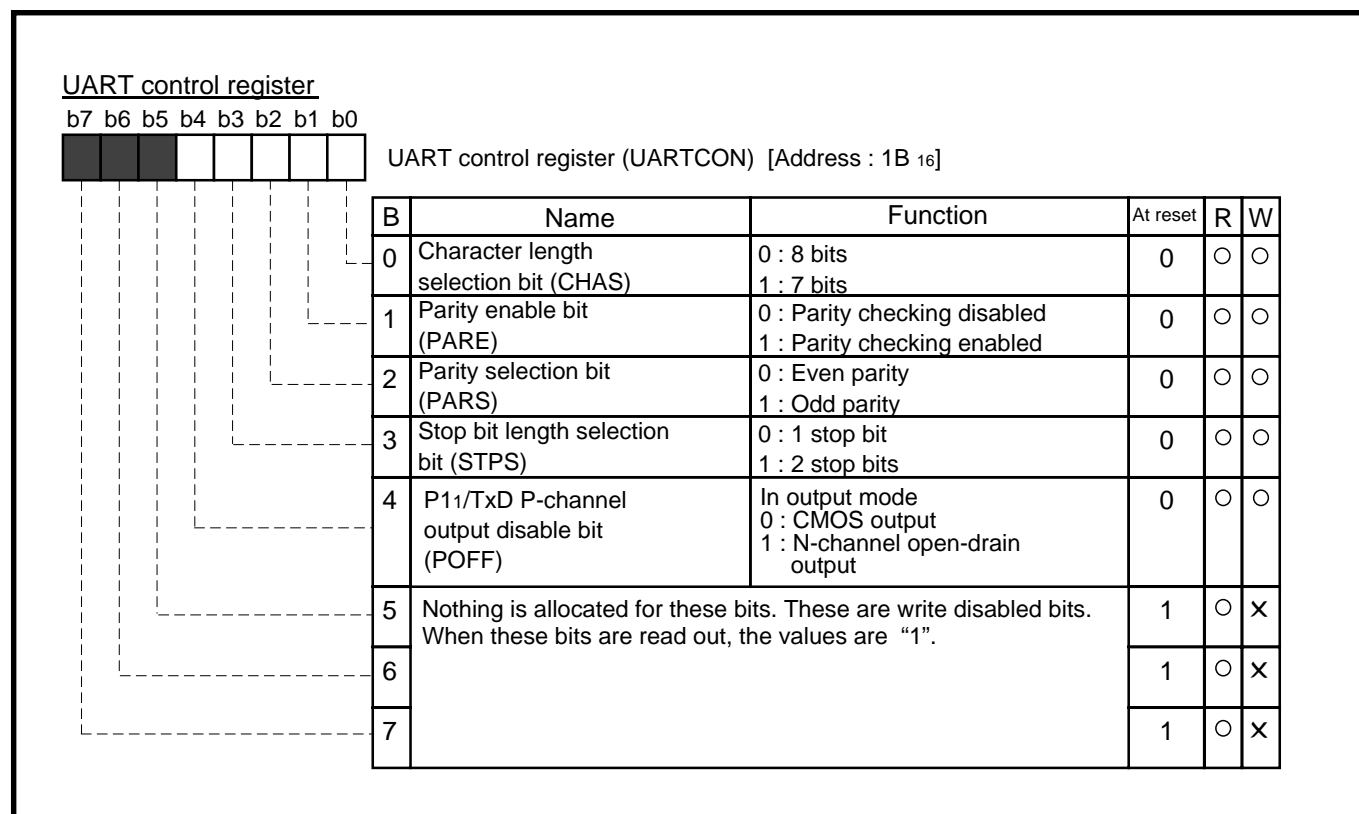


Fig. 2.3.5 Structure of UART control register

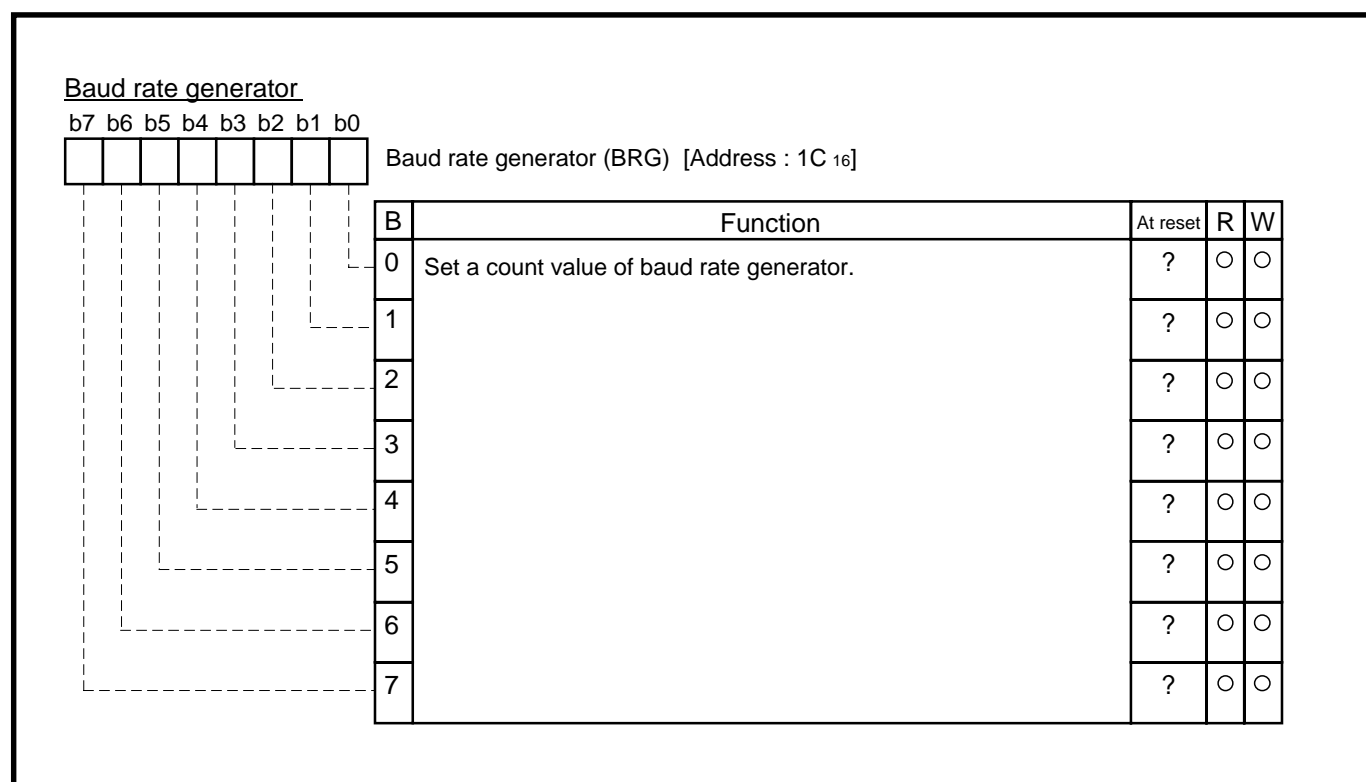


Fig. 2.3.6 Structure of Baud rate generator

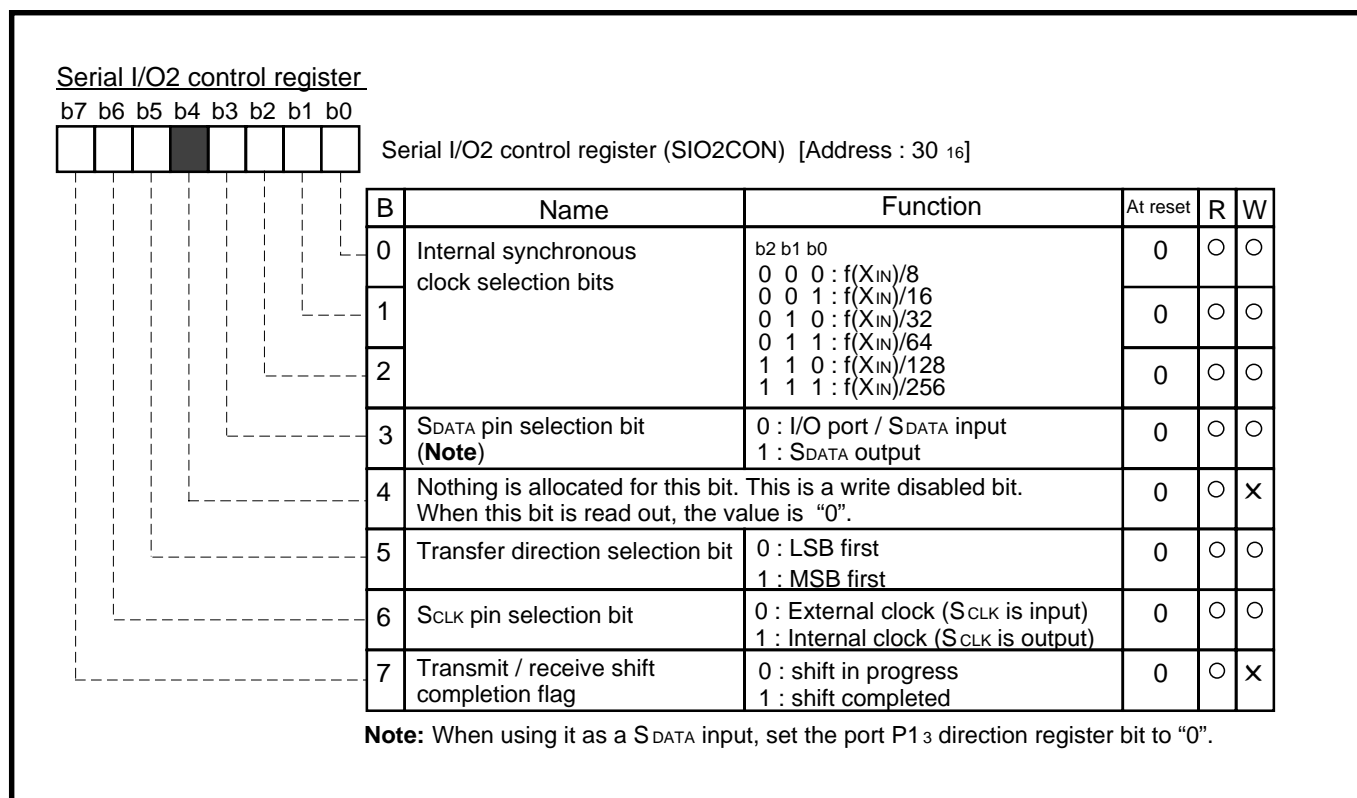


Fig. 2.3.7 Structure of Serial I/O2 control register

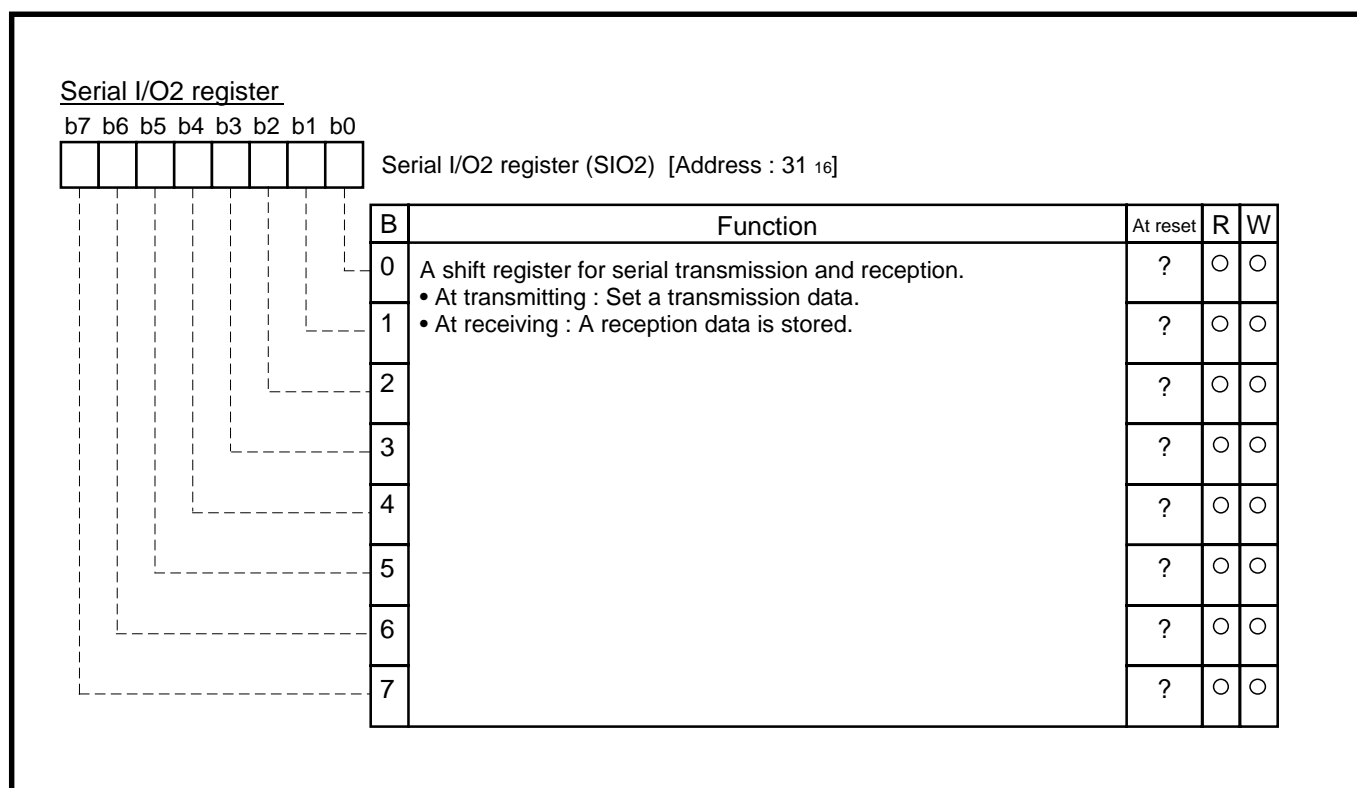


Fig. 2.3.8 Structure of Serial I/O2 register

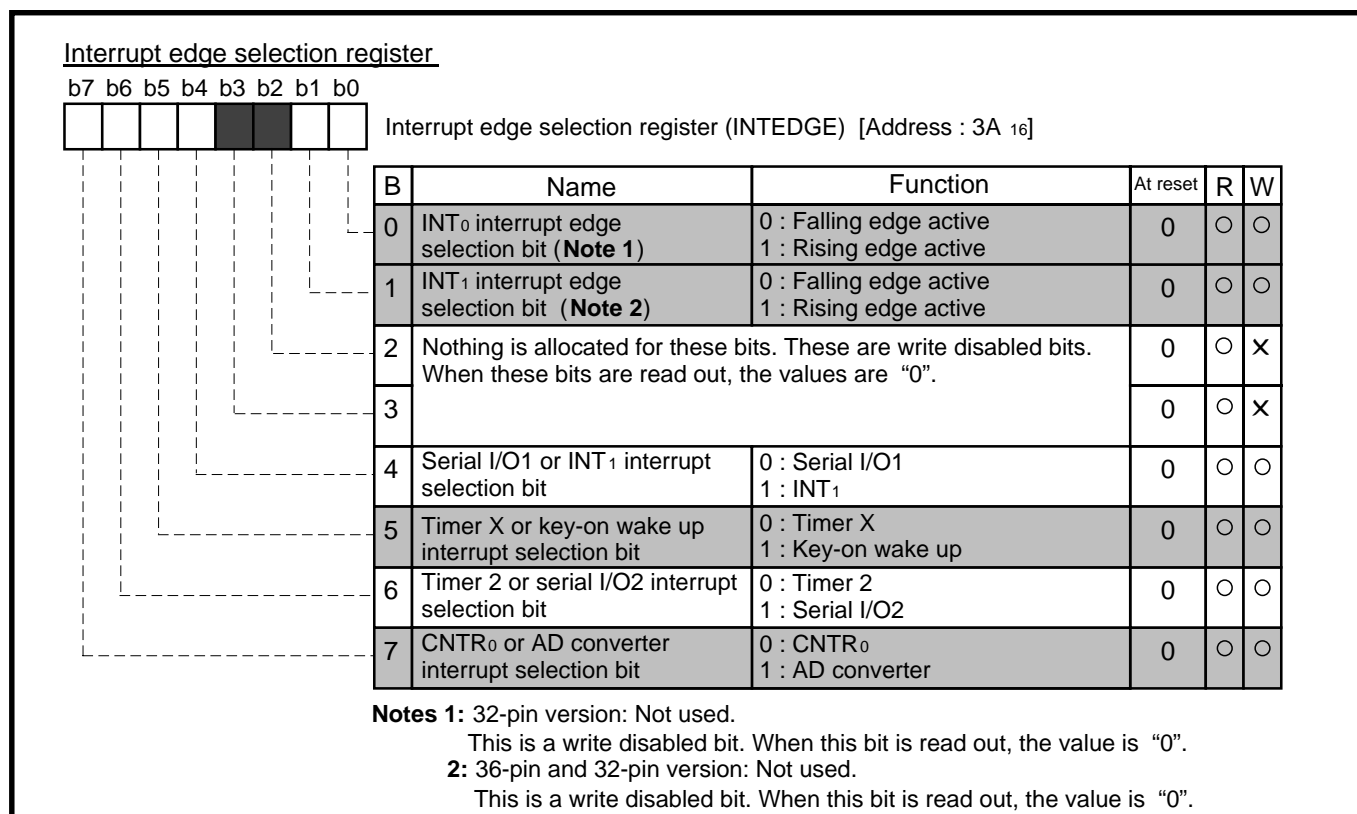


Fig. 2.3.9 Structure of Interrupt edge selection register

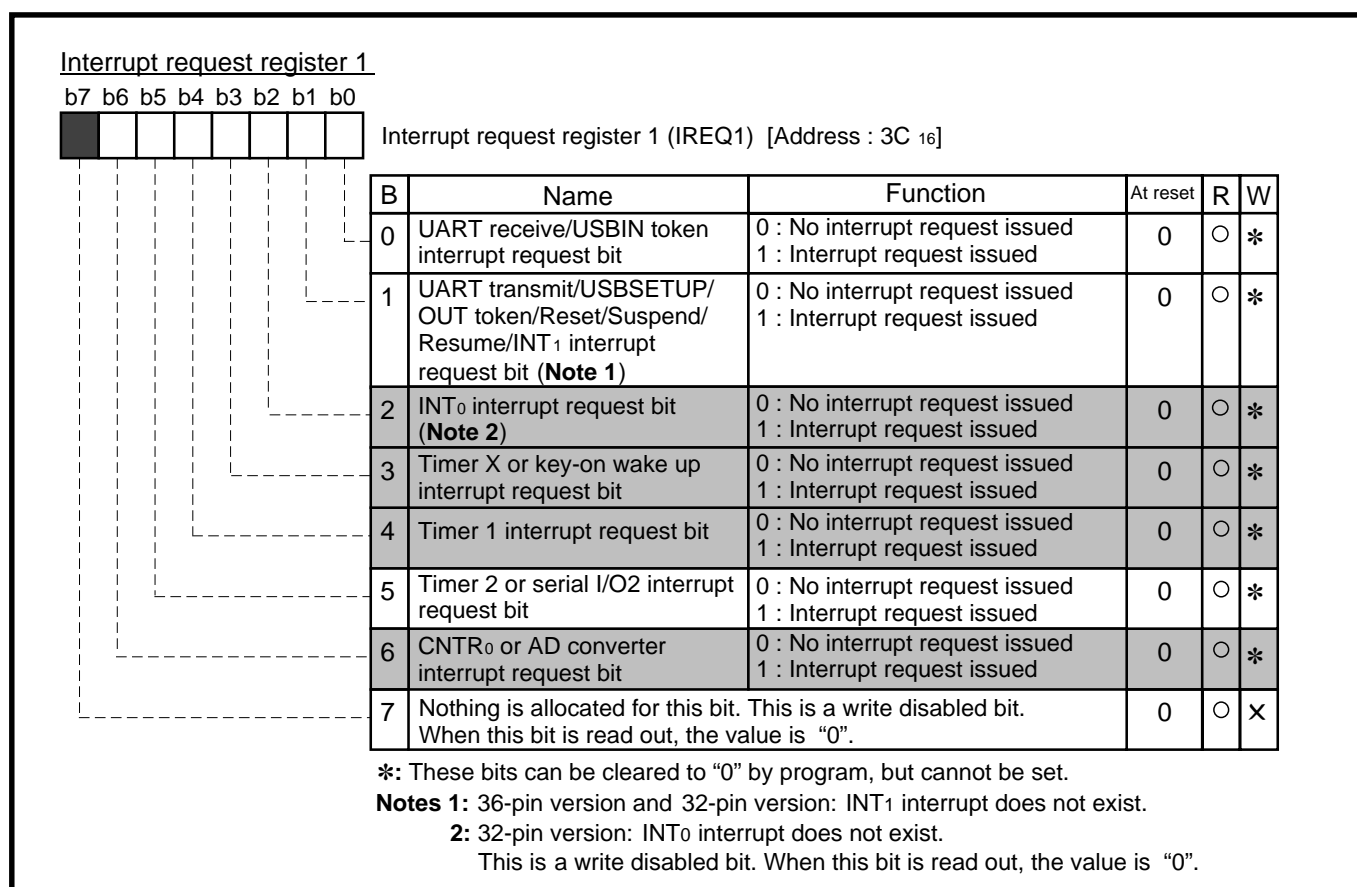


Fig. 2.3.10 Structure of Interrupt request register 1

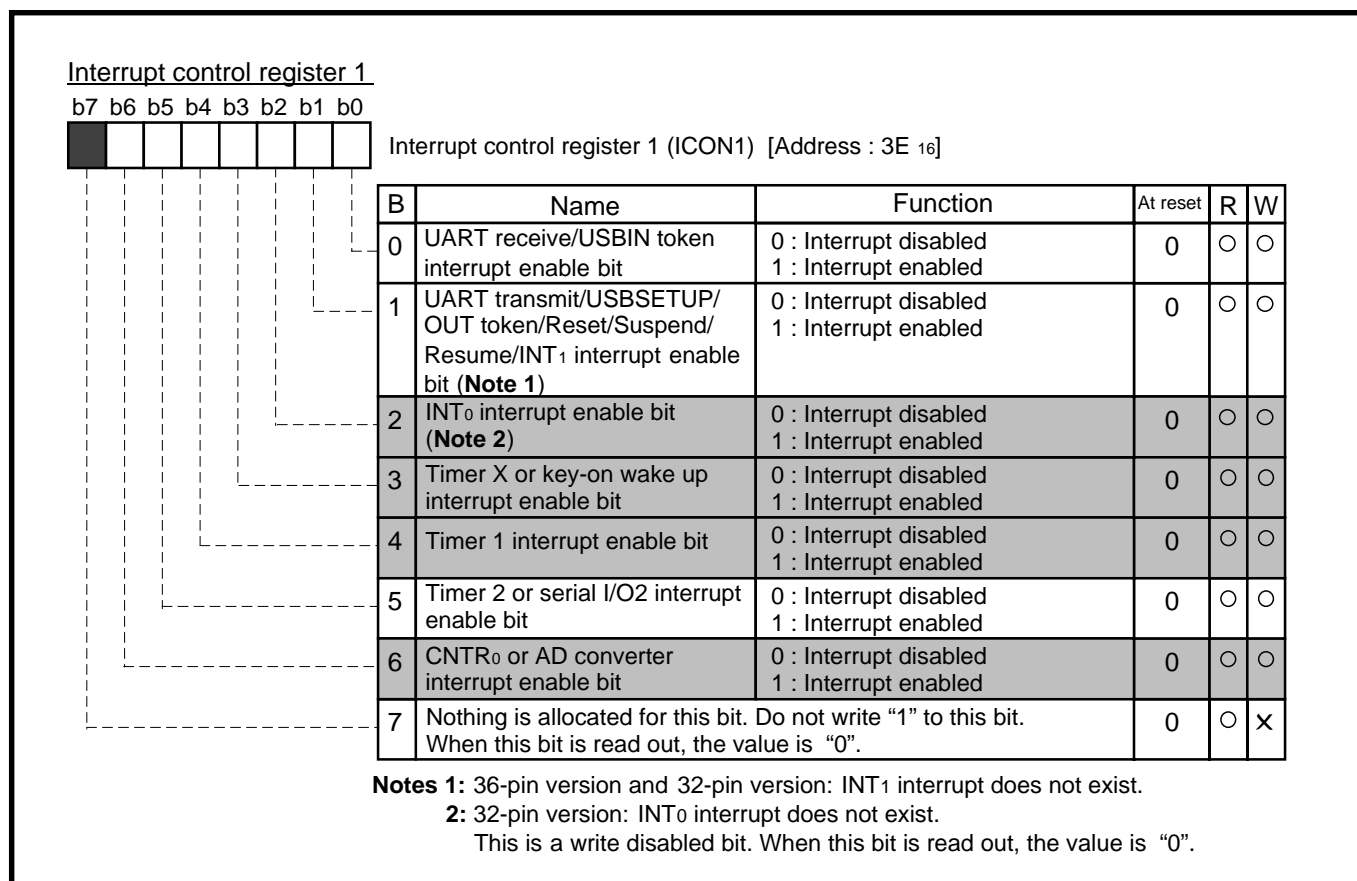


Fig. 2.3.11 Structure of Interrupt control register 1

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.

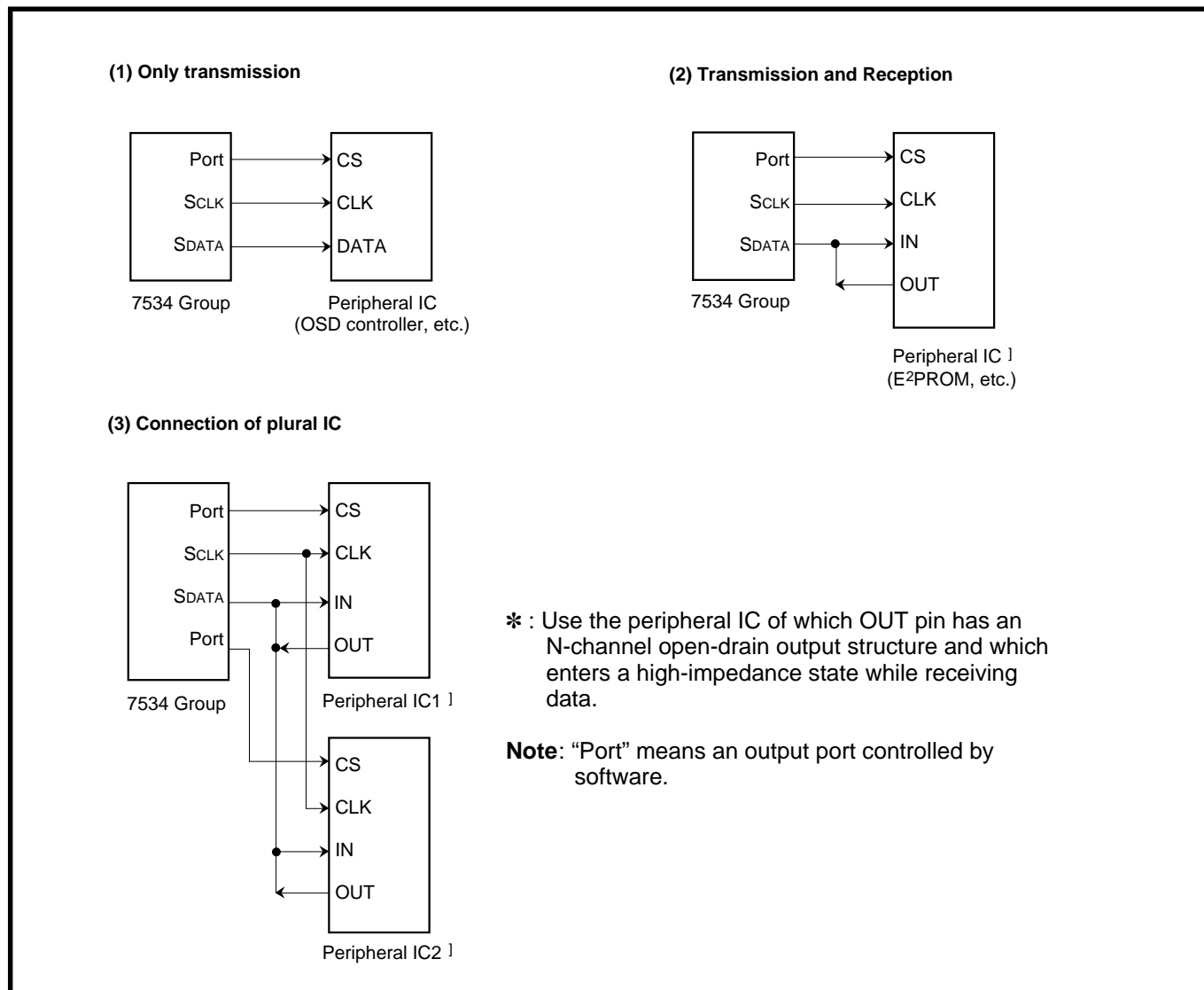


Fig. 2.3.12 Serial I/O connection examples (1)

(2) Connection with microcomputer

Figure 2.3.13 shows connection examples with another microcomputer.

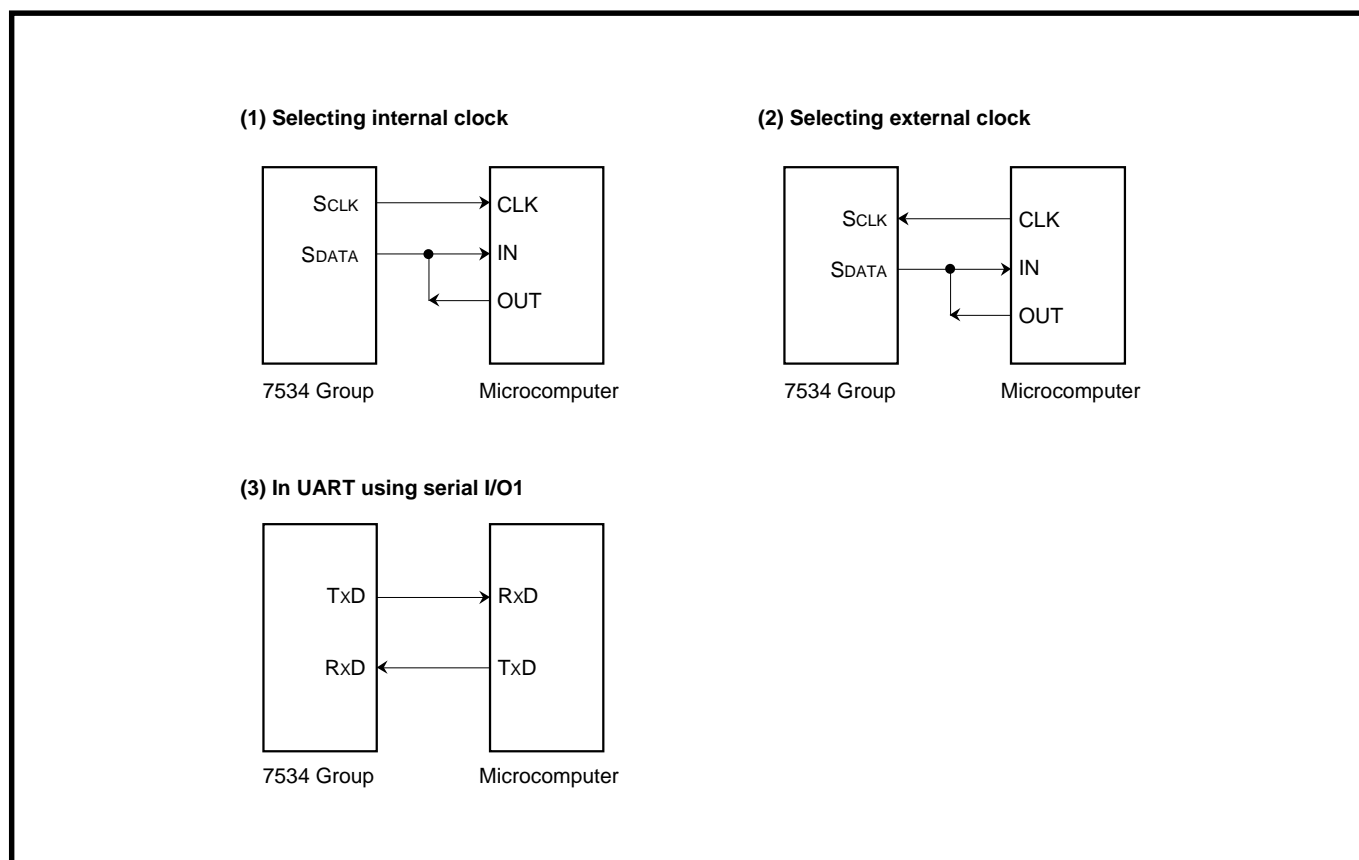


Fig. 2.3.13 Serial I/O connection examples (2)

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.

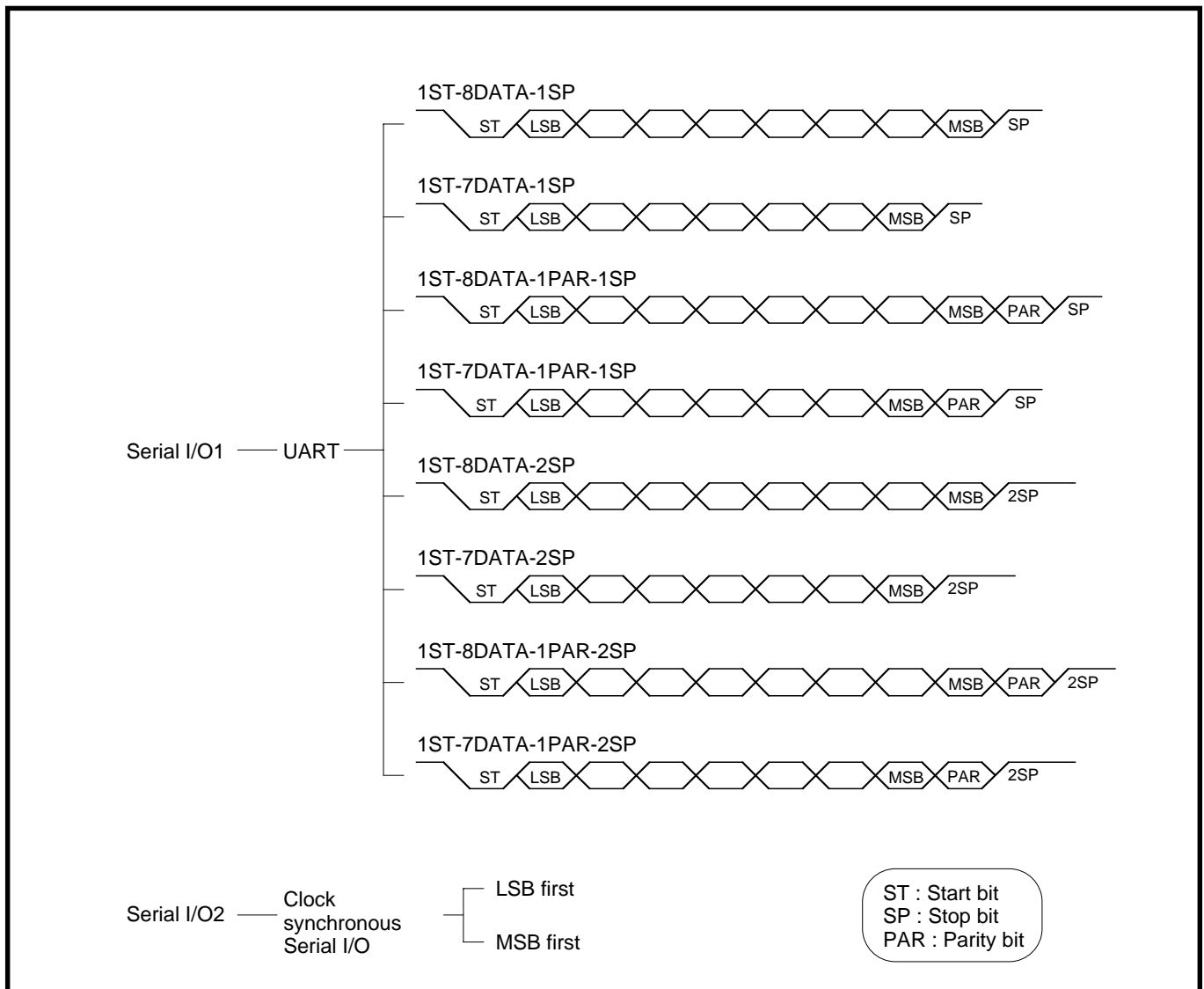


Fig. 2.3.14 Serial I/O transfer data format

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 P0₀ is used for communication control and outputs the quasi- $\overline{\text{SRDY}}$ signal.

The following explain an example using the serial I/O₂. Figure 2.3.15 shows a connection diagram, and Figure 2.3.16 shows a timing chart.

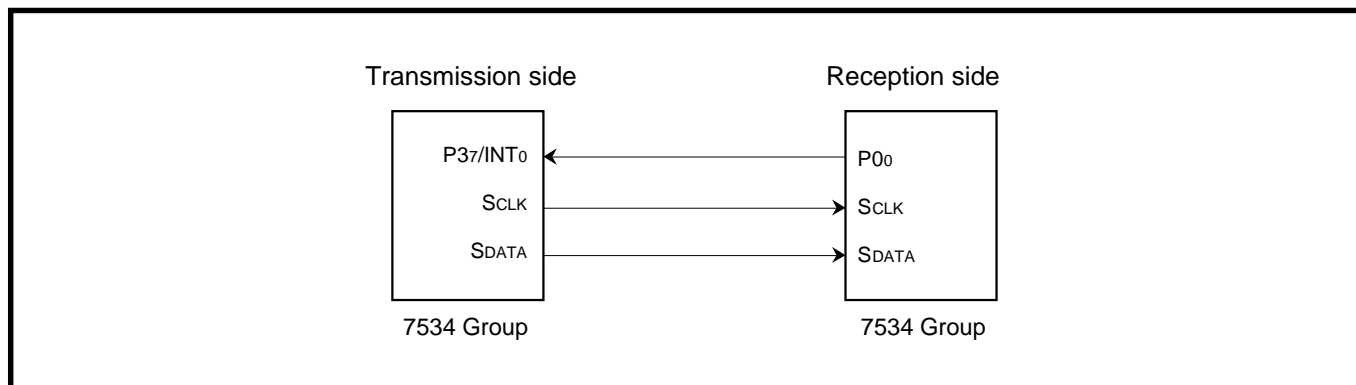


Fig. 2.3.15 Connection diagram

Specifications : •The Serial I/O₂, 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.

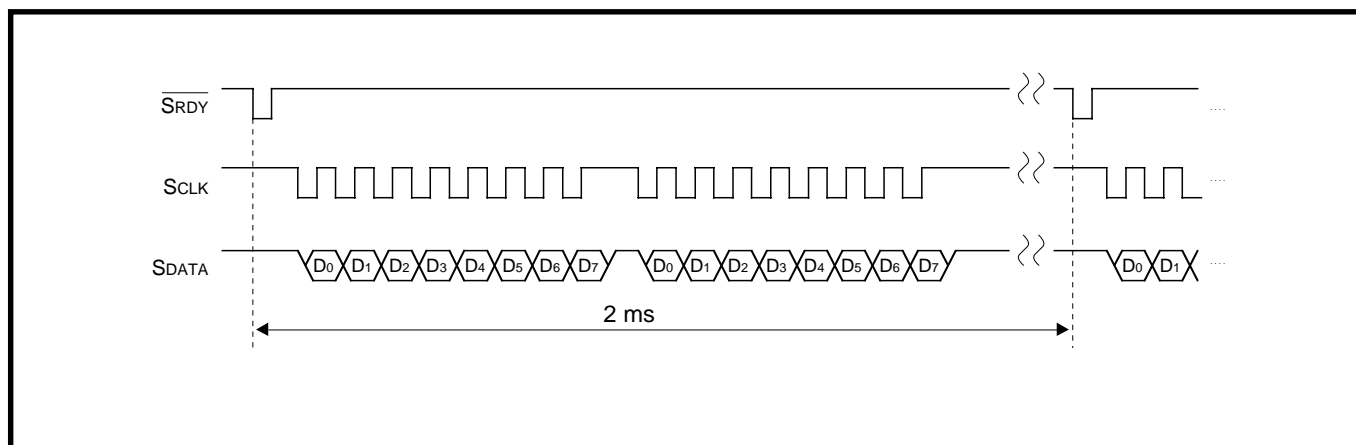


Fig. 2.3.16 Timing chart

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.

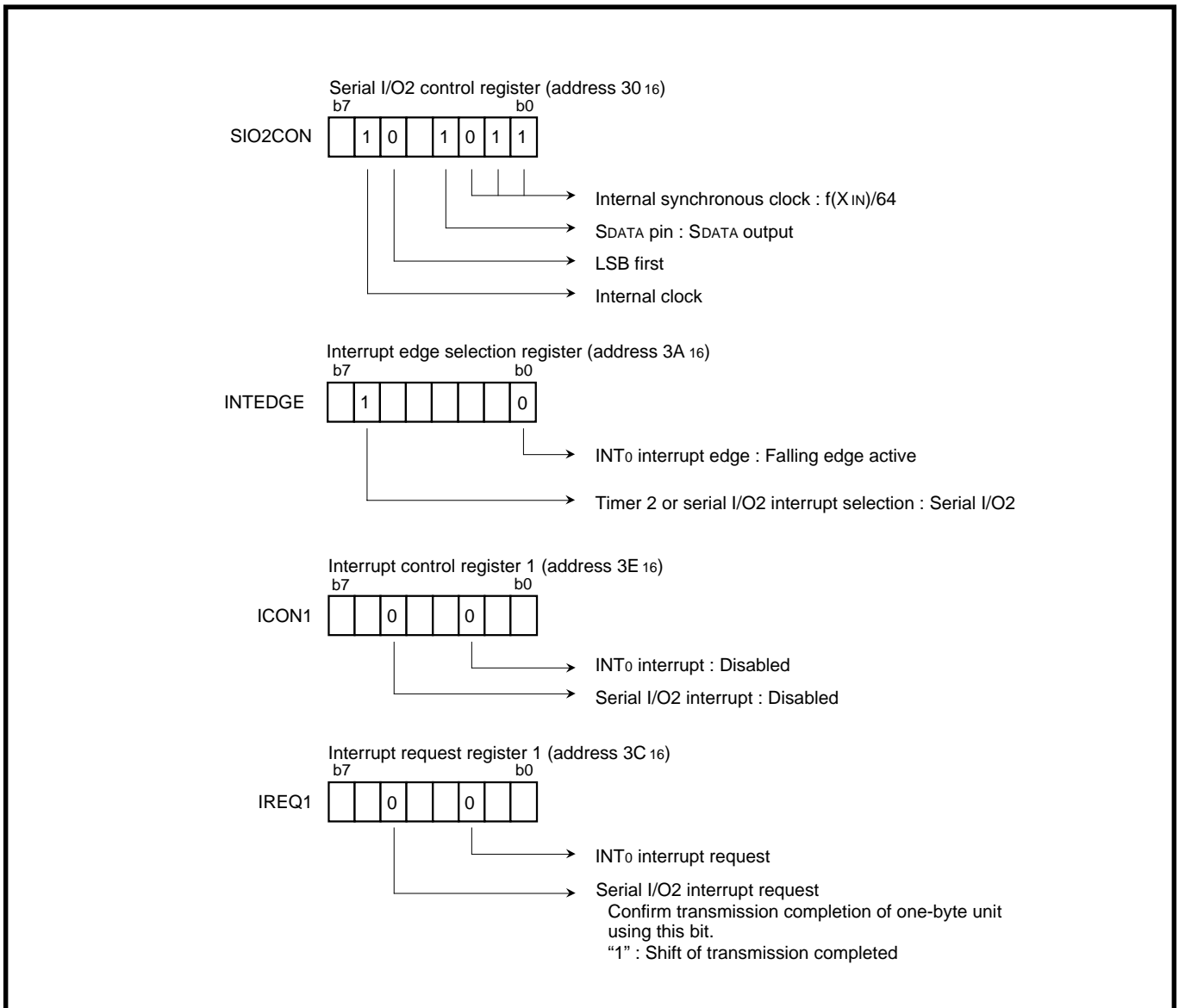


Fig. 2.3.17 Registers setting relevant to transmission side

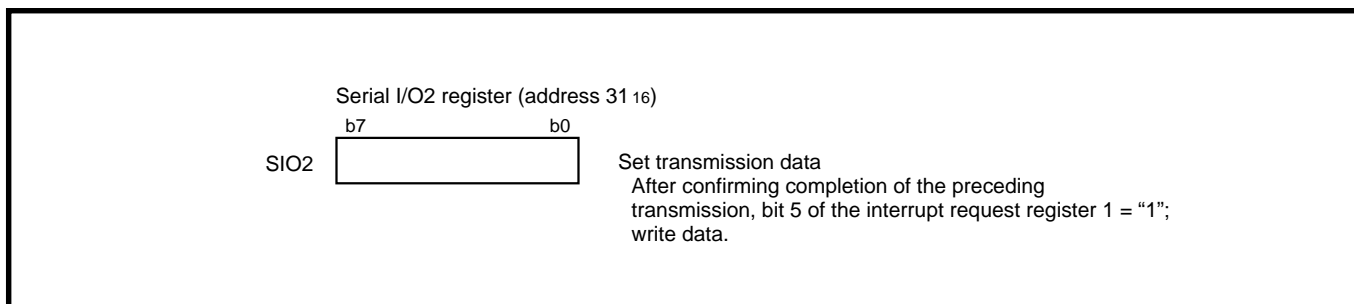


Fig. 2.3.18 Transmission data setting of serial I/O2

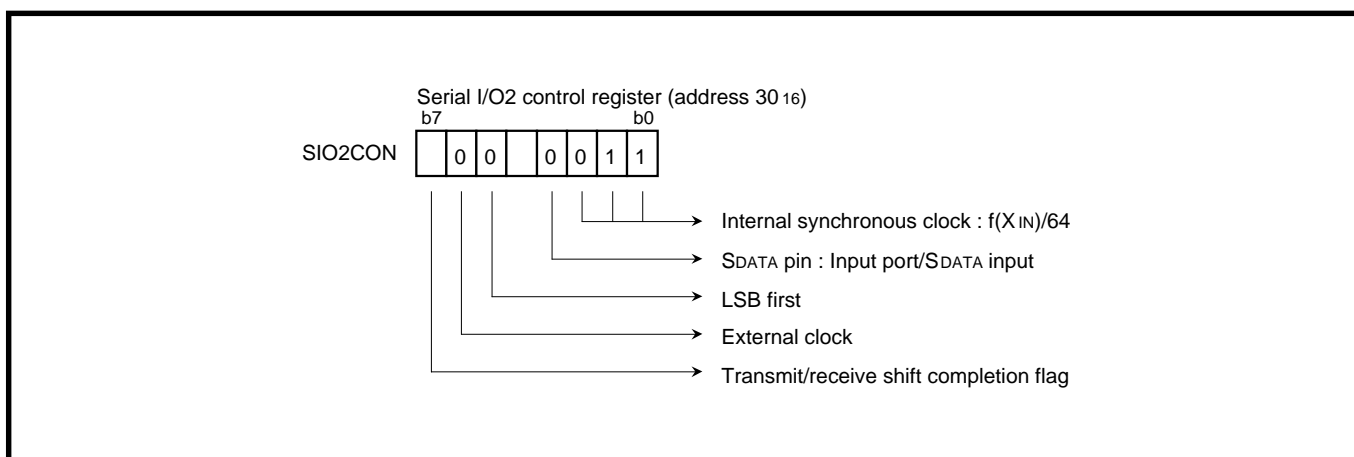


Fig. 2.3.19 Registers setting relevant to reception side

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

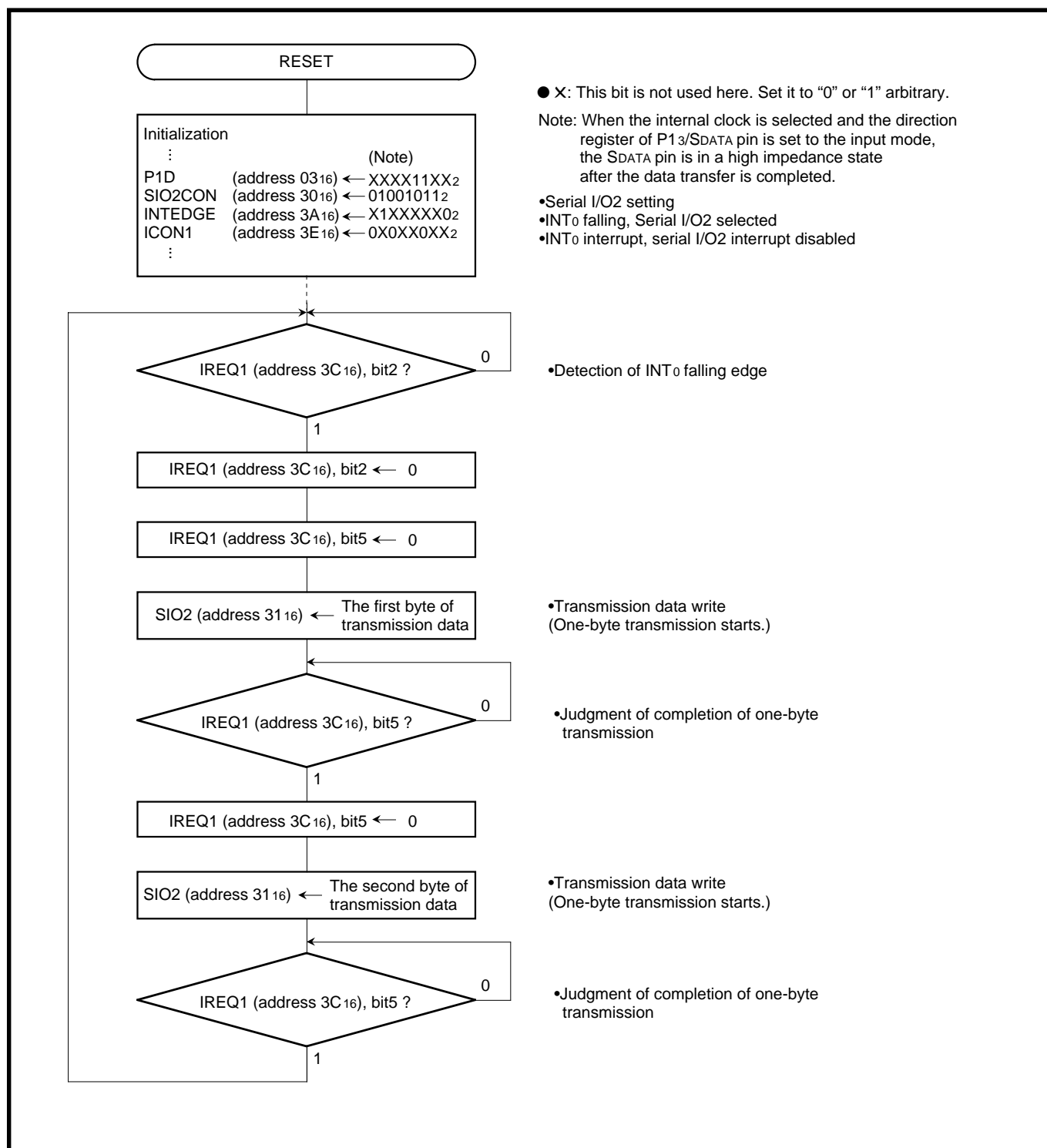


Fig. 2.3.20 Control procedure of transmission side

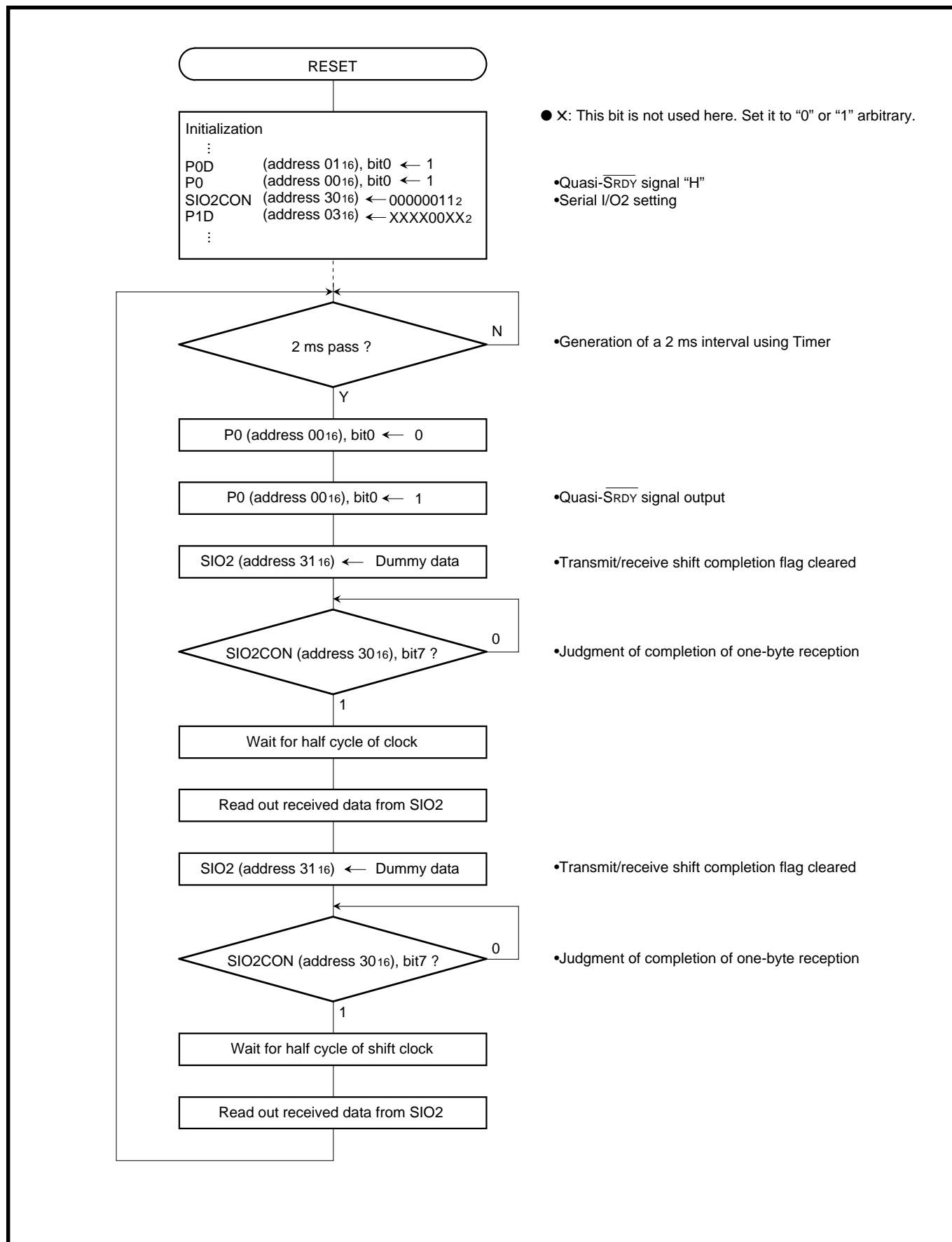


Fig. 2.3.21 Control procedure of reception side

(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 P0₀ is used for communication control.

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

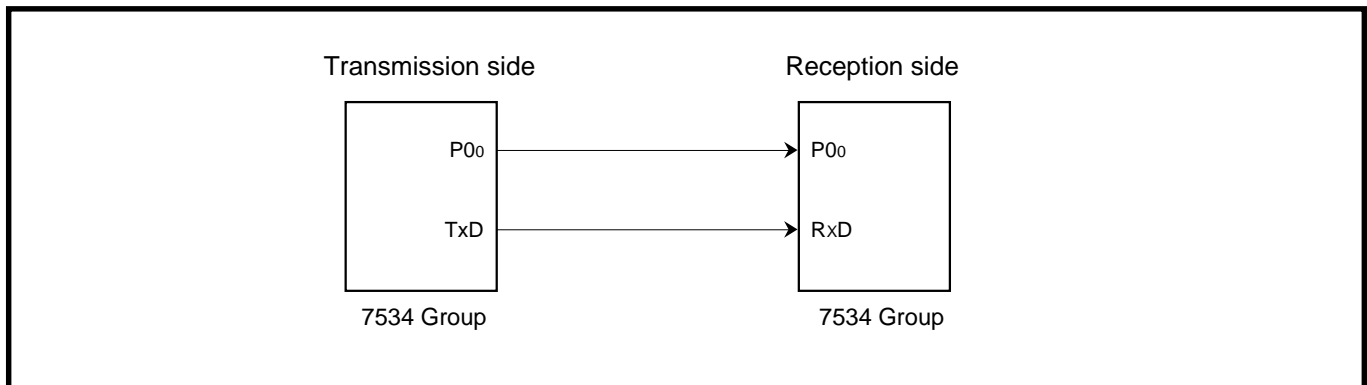


Fig. 2.3.22 Connection diagram

- 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 P0₀; Port P0₀ 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

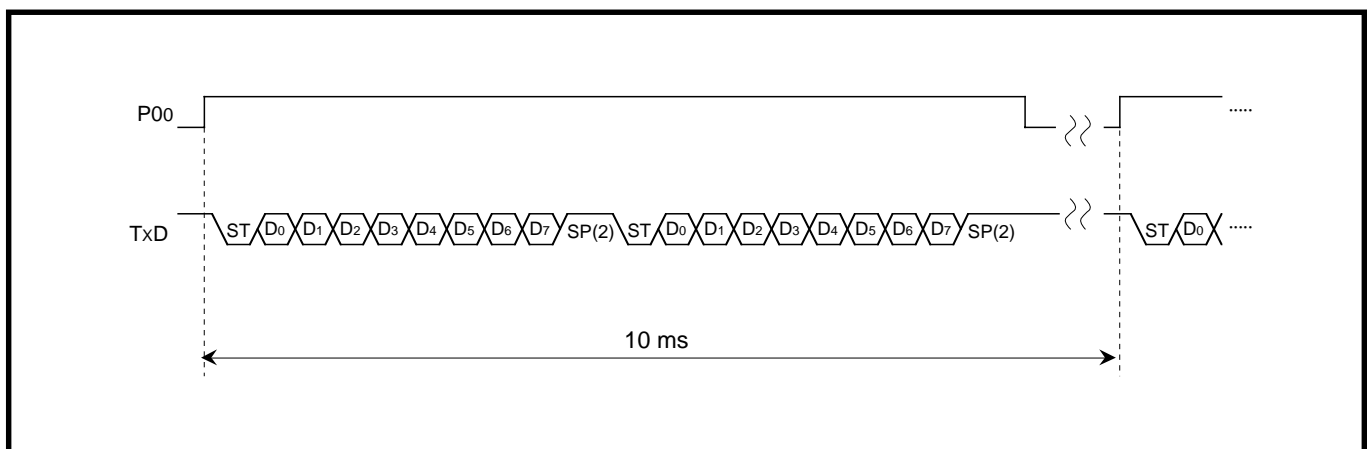


Fig. 2.3.23 Timing chart

Table 2.3.1 shows a setting example of the baud rate generator (BRG) and transfer bit rate values; Figure 2.3.24 shows the registers setting relevant to transmission side; Figure 2.3.25 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 $f(X_{IN}) = 4.9152$ MHz	At $f(X_{IN}) = 6$ 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"

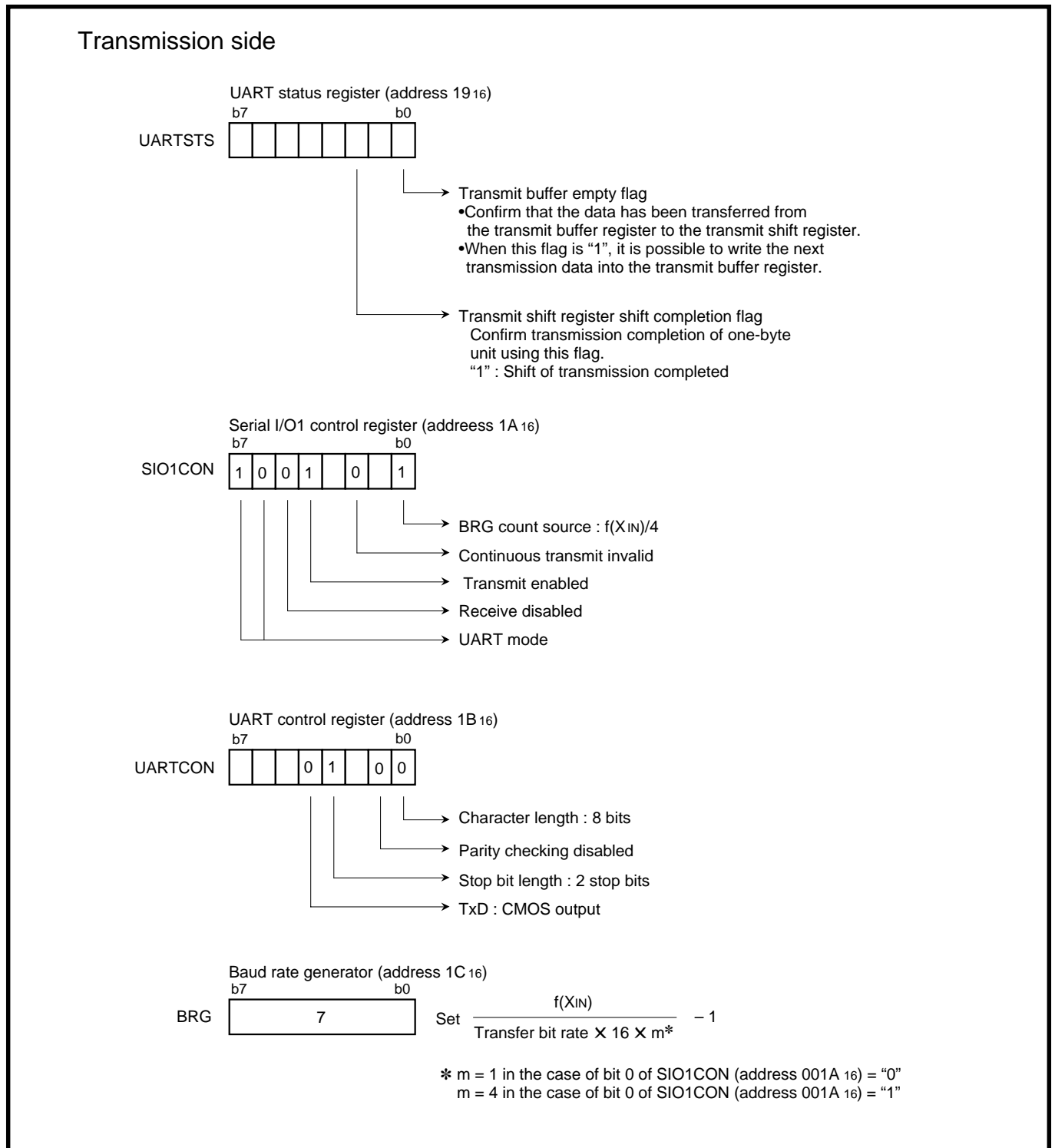


Fig. 2.3.24 Registers setting relevant to transmission side

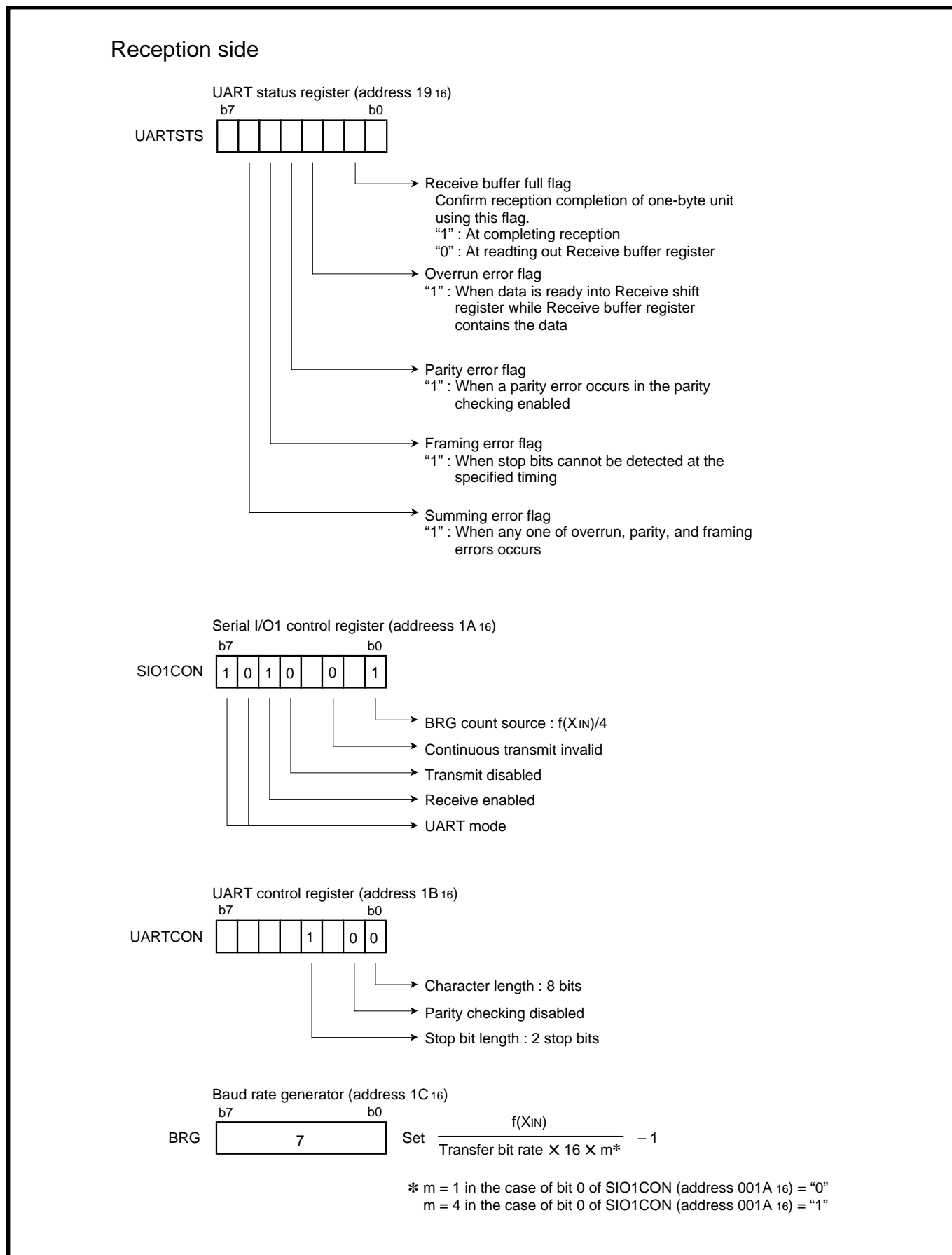


Fig. 2.3.25 Registers setting relevant to reception side

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

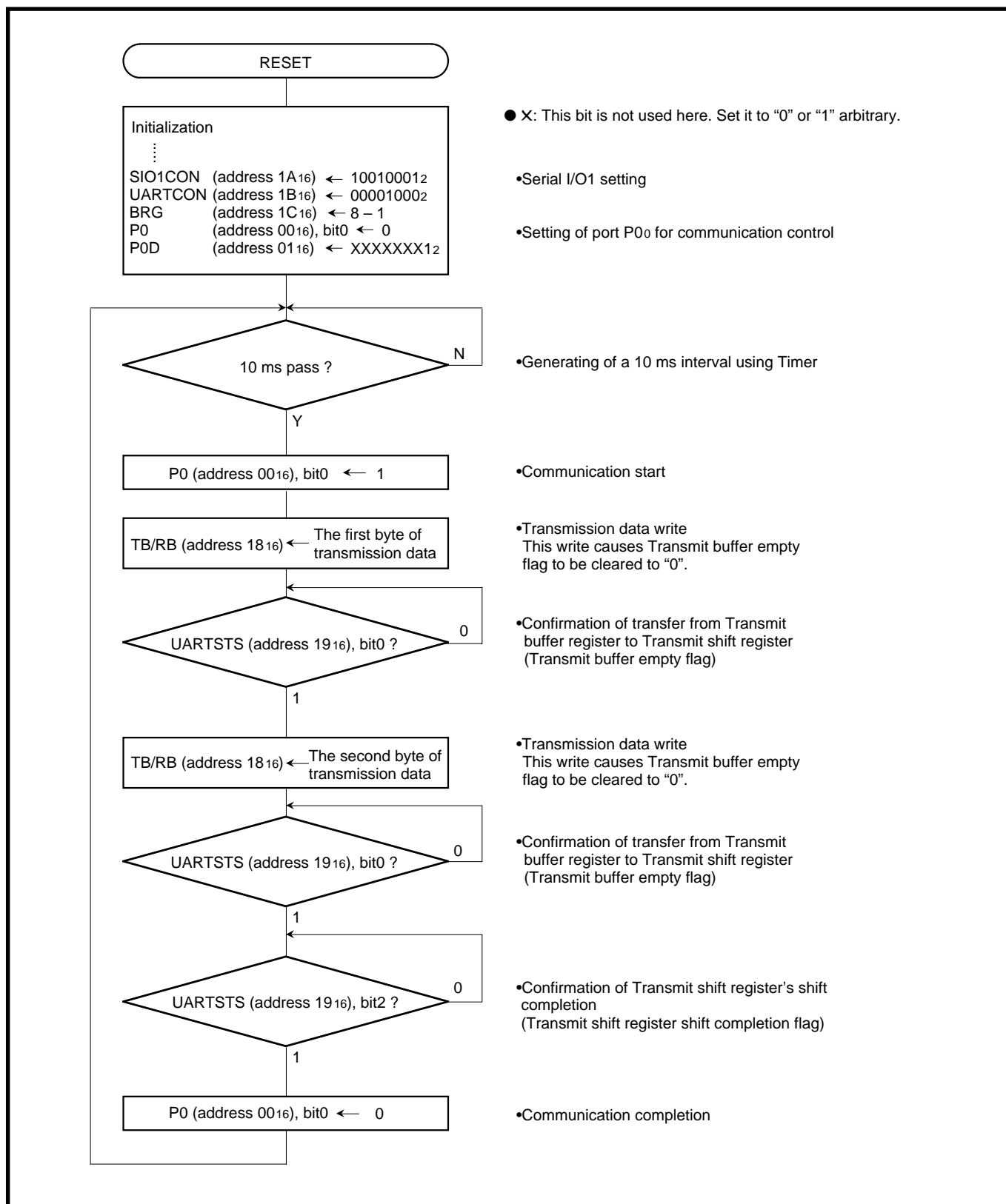


Fig. 2.3.26 Control procedure of transmission side

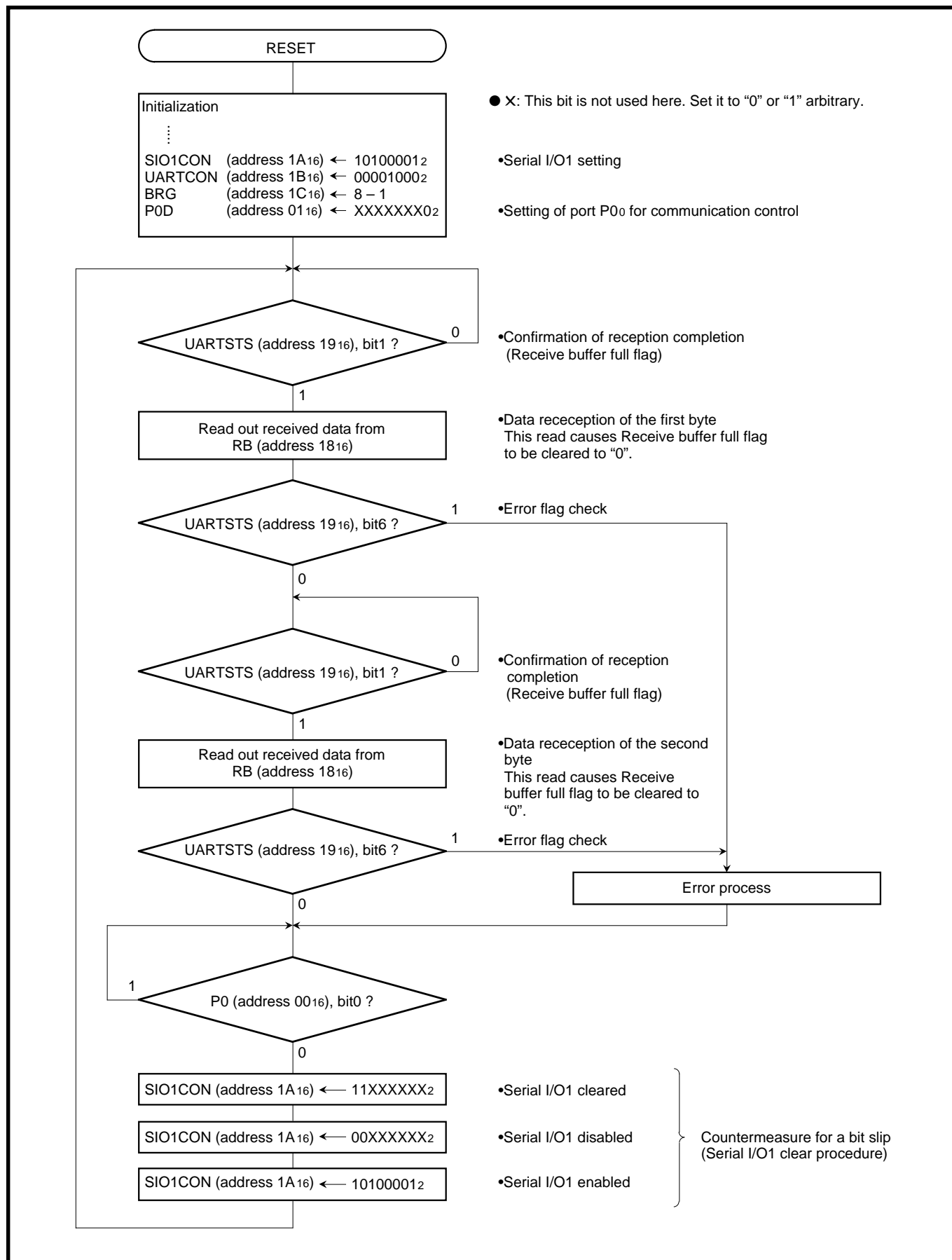


Fig. 2.3.27 Control procedure of reception side

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.

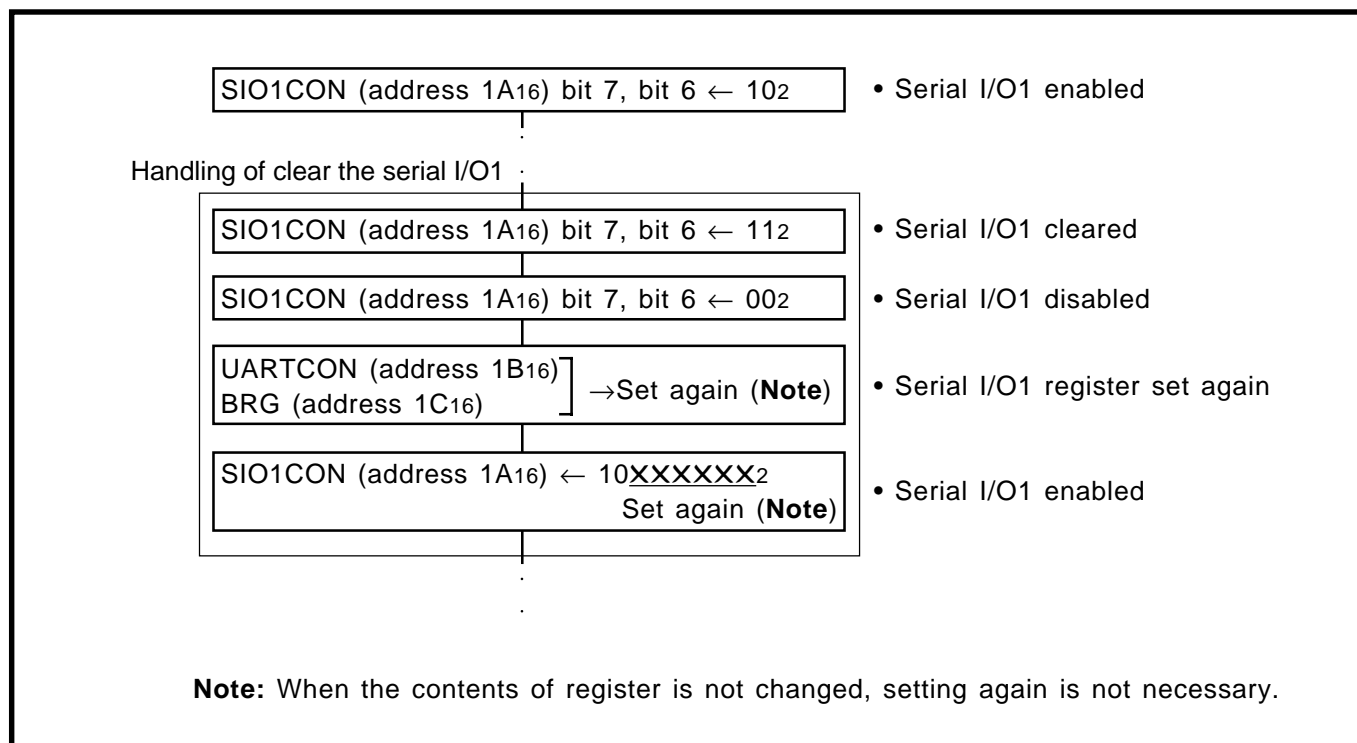


Fig. 2.3.28 Sequence of clearing serial I/O

(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 (bit 7) 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.

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

7534 Group has USB functions in compliance with the Low-Speed USB2.0 specification which are connection standard with PC peripherals.

In this section, the outline of USB communication function and the USB function of 7534 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 3 types of communication standards exist depending on the data amount treated on the peripherals side.

- **Hi-Speed function: H.S.**

USB operation at 480Mbps.

- **Full-Speed function: F.S.**

USB operation at 12Mbps.

- **Low-Speed function: L.S.**

USB operation at 1.5Mbps.

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

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

The 7534 Group has USB Low-Speed 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 7534 Group).

Figure 2.4.1 shows the communication sequence of USB.

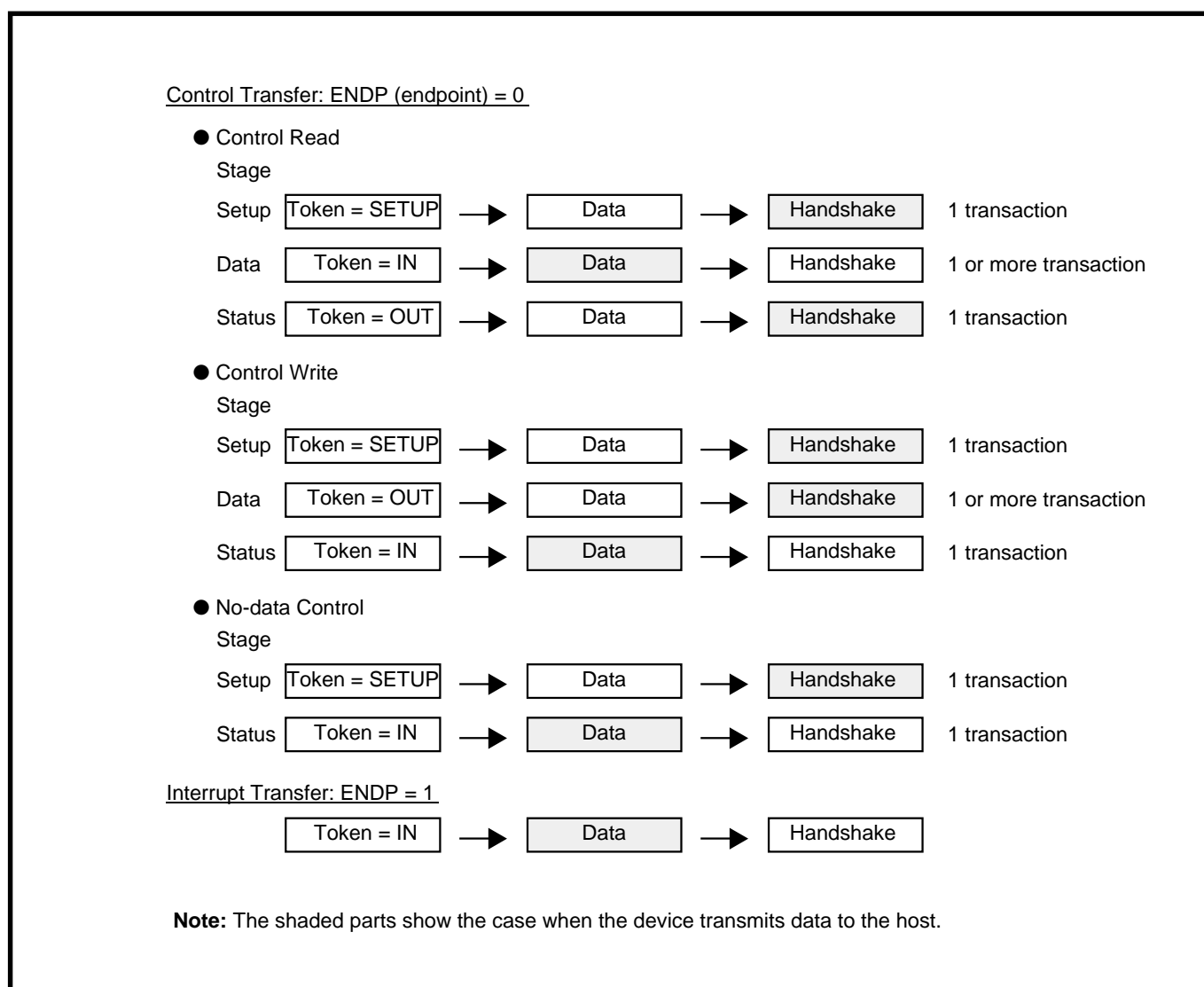


Fig. 2.4.1 Communication sequence of USB

(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 7534 Group, the token, data, and the handshake of packet types in Table 2.4.2 can be controlled by software.

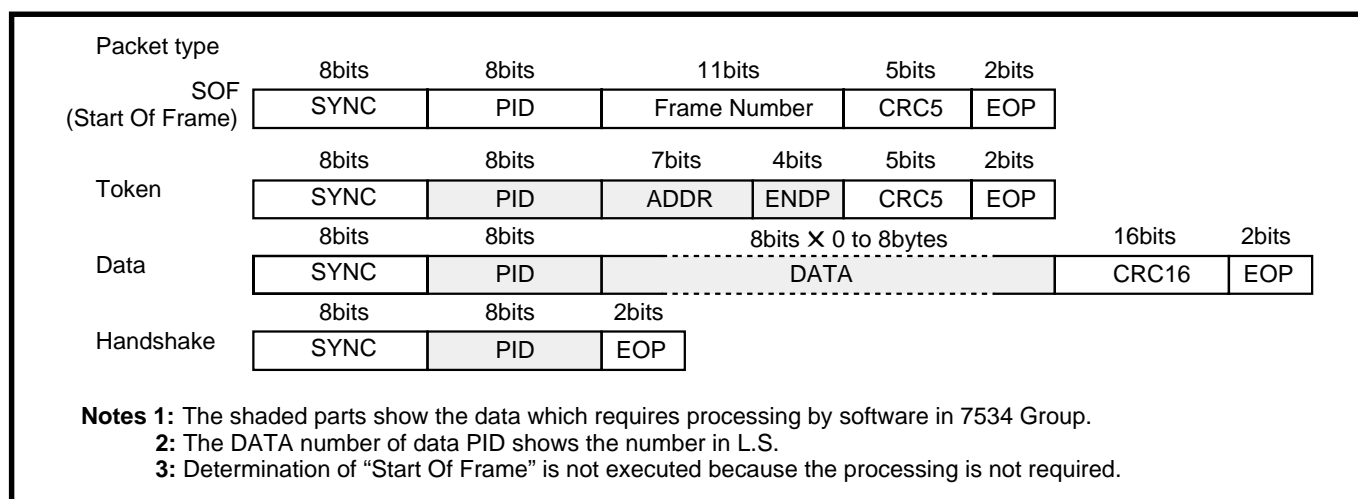
(4) Packet structure

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 7534 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 7534 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 ₂	The processing is reported by host to device
	IN	1001 ₂	Data transmit is requested from host to device
	OUT	0001 ₂	Data receive is requested from host to device
	SOF	0101 ₂	Top of frame is indicated by host to device
Data	DATA0	0011 ₂	The state that sequence bit of transmit/receive data is even is indicated
	DATA1	1011 ₂	The state that sequence bit of transmit/receive data is odd is indicated
Handshake	ACK	0010 ₂	Normal completion of communication is reported
	NAK	1010 ₂	The state that device is waiting for communication is reported
	STALL	1110 ₂	Completion error of communication is reported
Special	PRE	1100 ₂	Communication to the L.S. device which has low-priority is enabled

In 7534 Group, data except SOF and PRE of PID can be controlled by software.

(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 1 to 15 ms.

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

When the SE0 signal of 2.5 μ 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 μ s or more
Remote wake up	Report of return to other devices	K state output for 1 to 15 ms

(7) USB interface

In 7534 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.

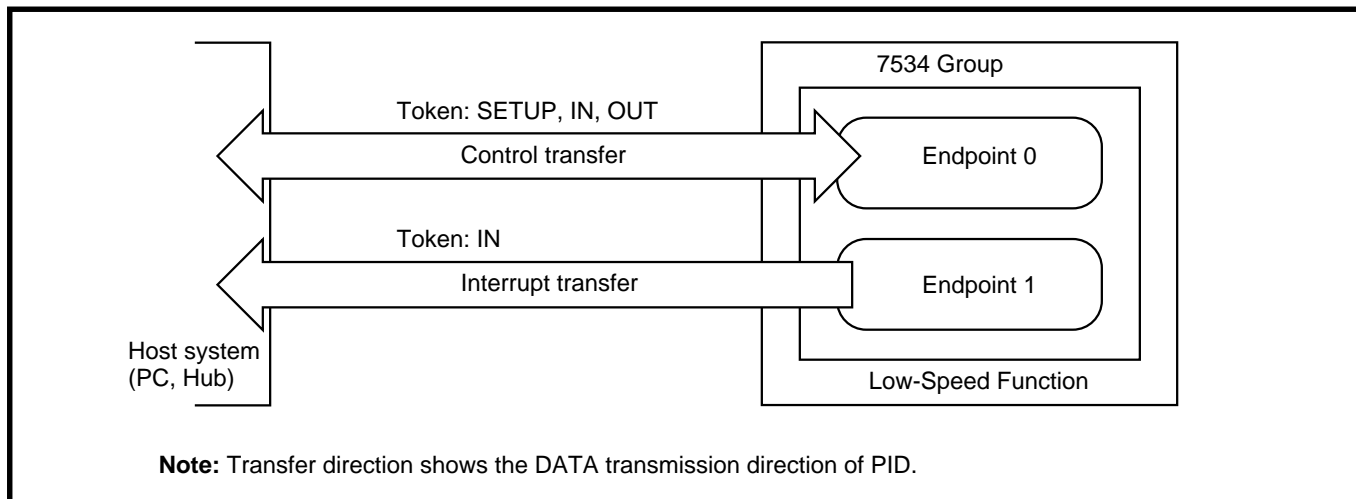


Fig. 2.4.3 USB (L.S.) interface

(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 Ω of communications lines.

Figure 2.4.4 shows the example of connecting USB (L.S.).

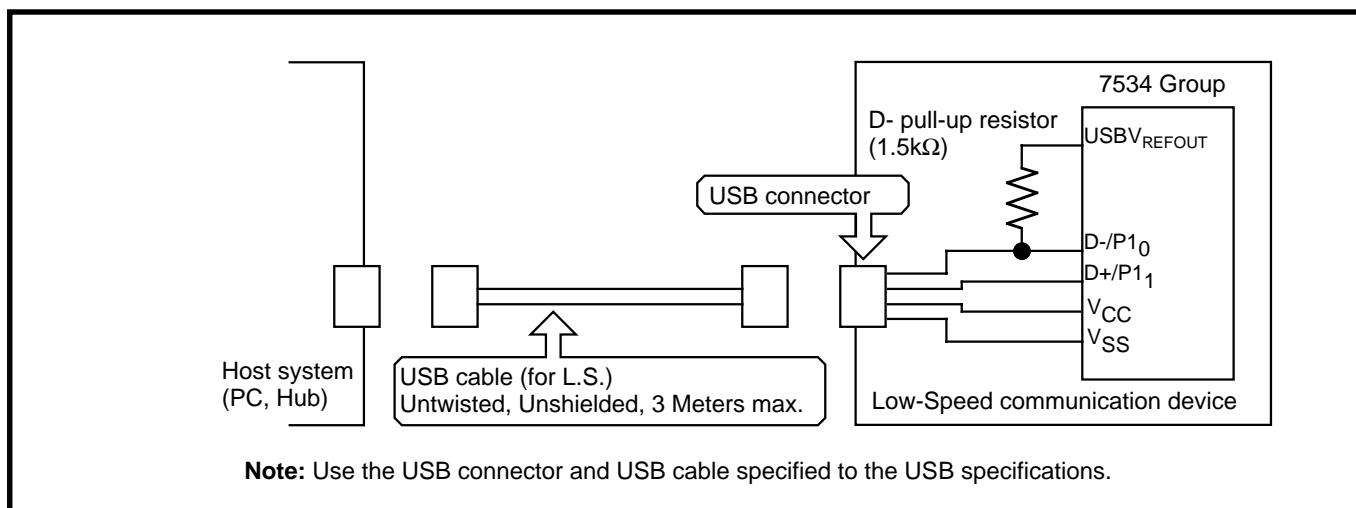


Fig. 2.4.4 USB (L.S.) connection example

2.4.2 Memory map

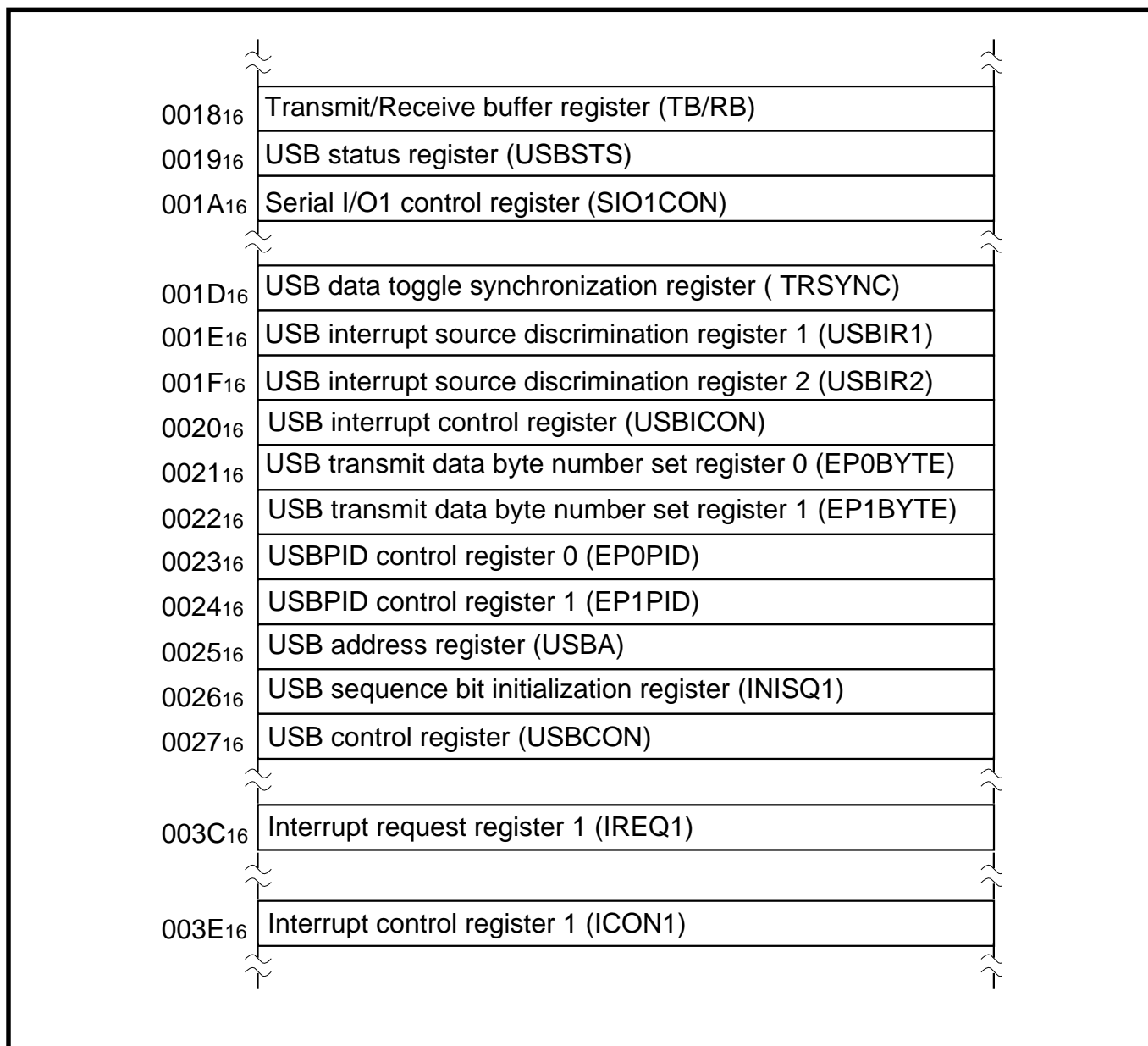


Fig. 2.4.5 Memory map of registers relevant to USB

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.

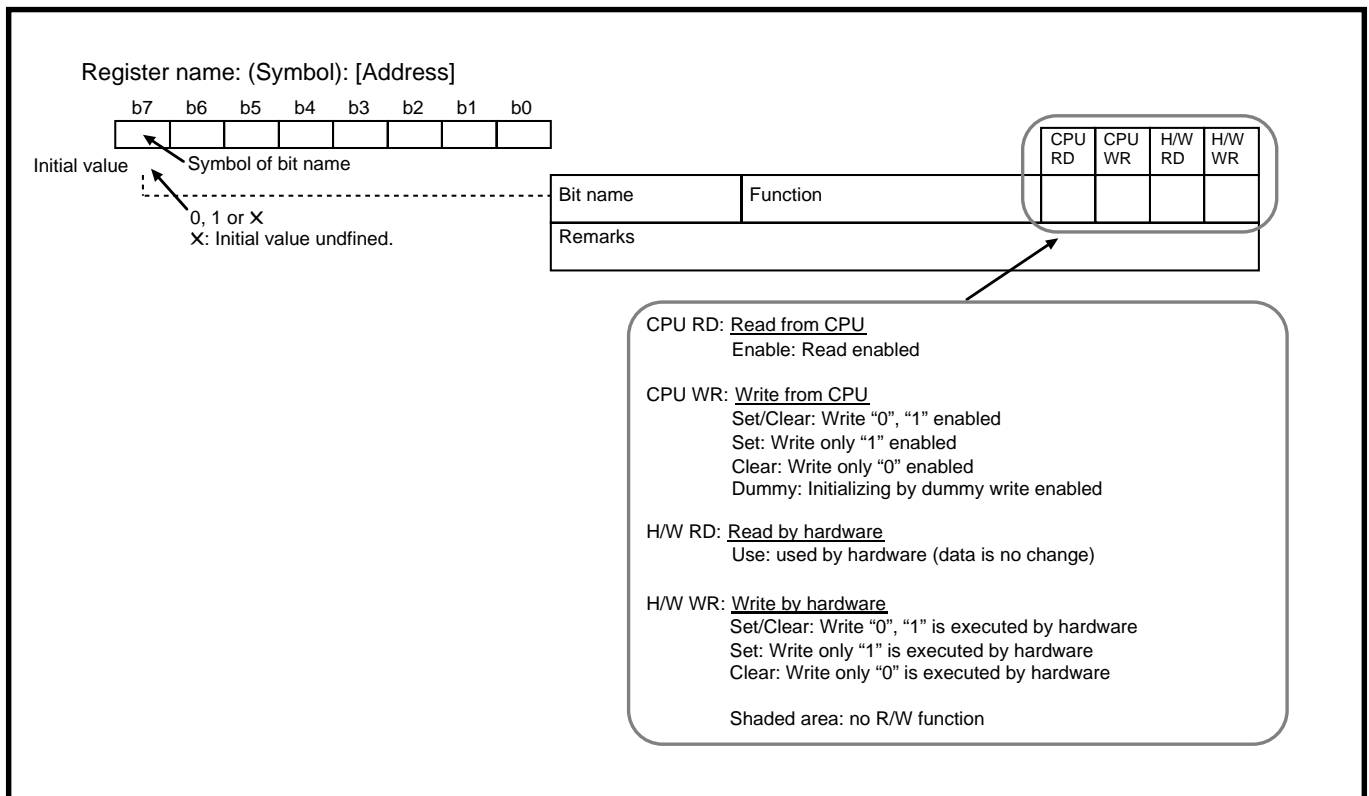


Fig. 2.4.6 Description of the register structure

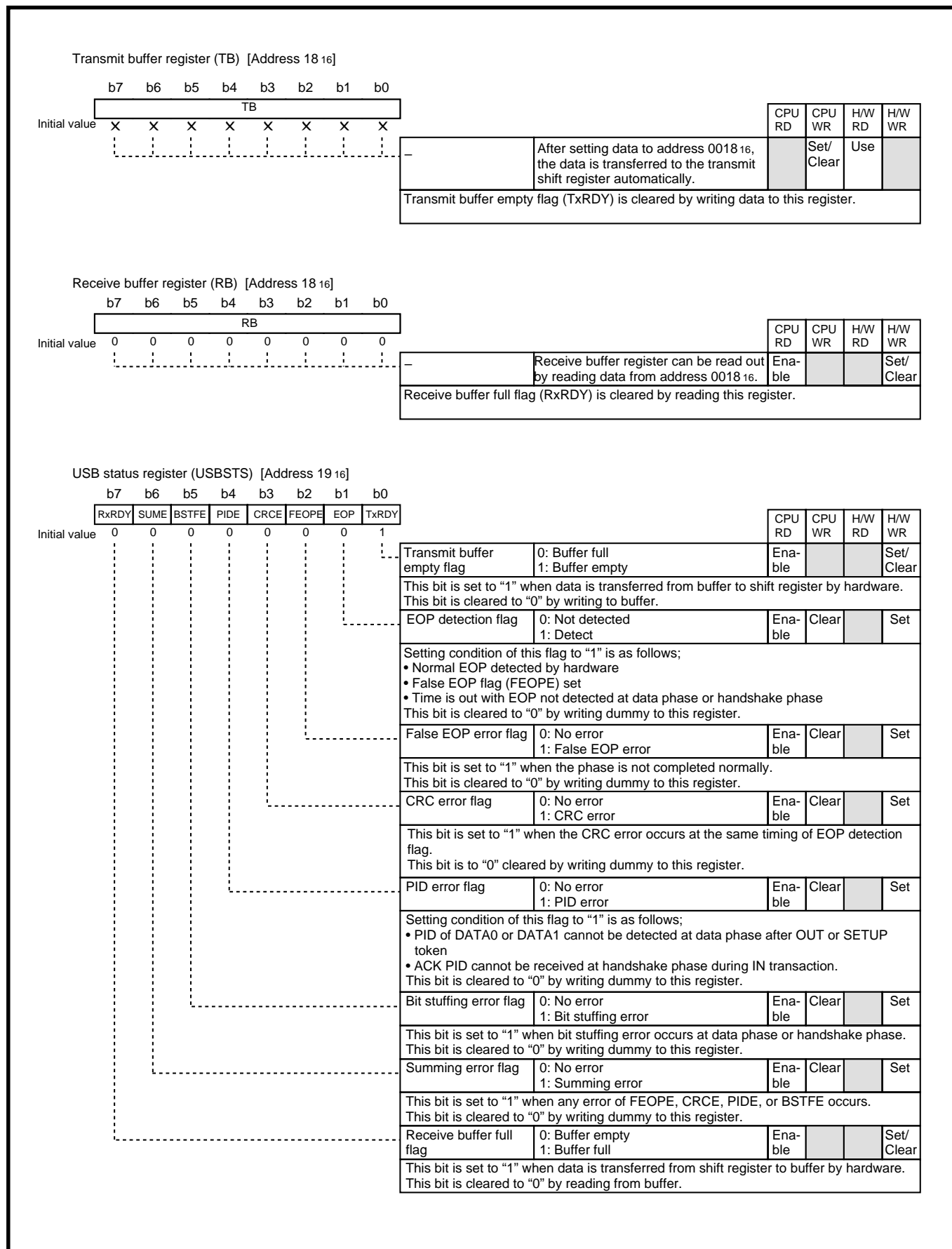


Fig. 2.4.7 Register structures relevant to USB (1)

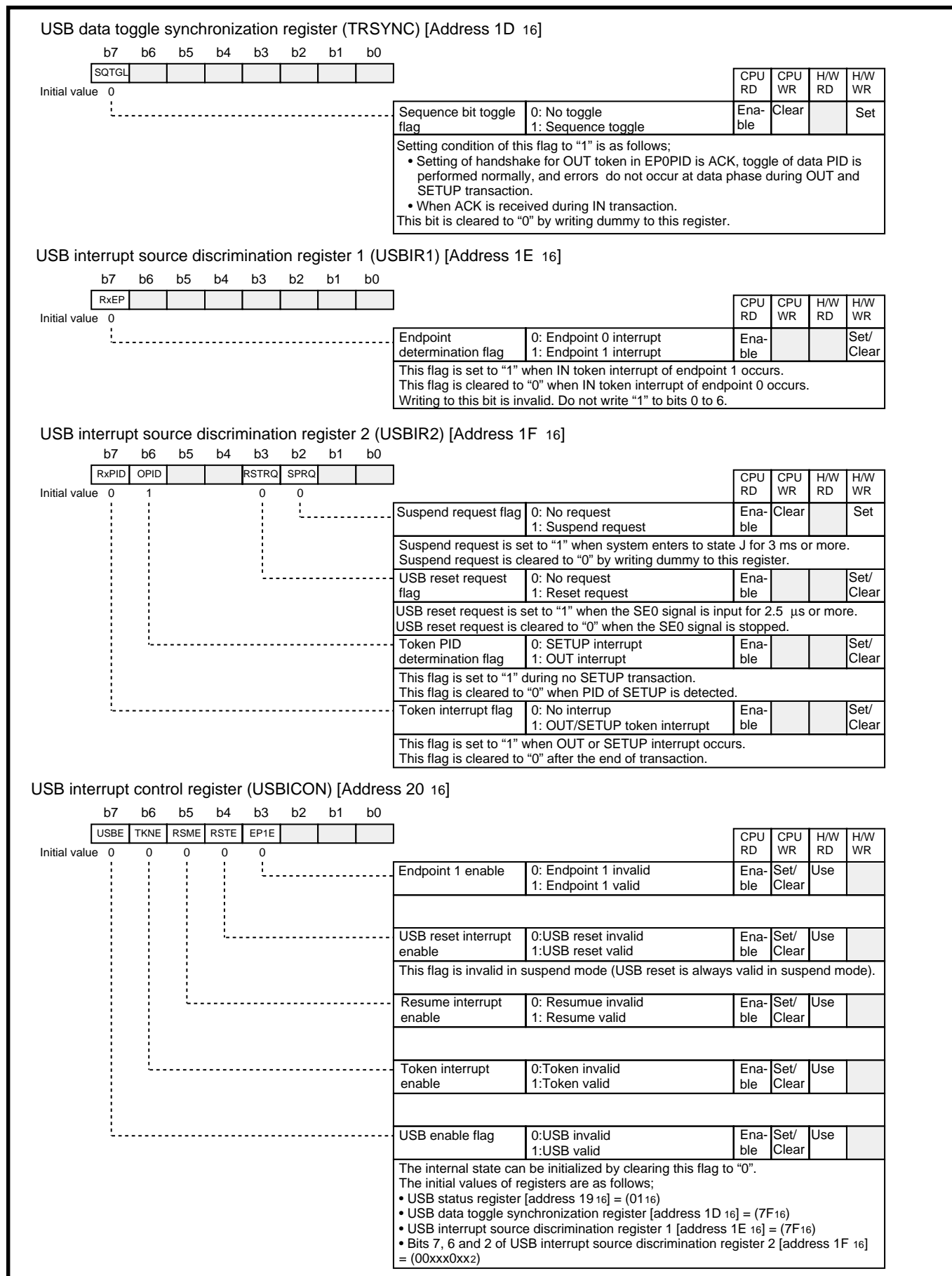


Fig. 2.4.8 Register structures relevant to USB (2)

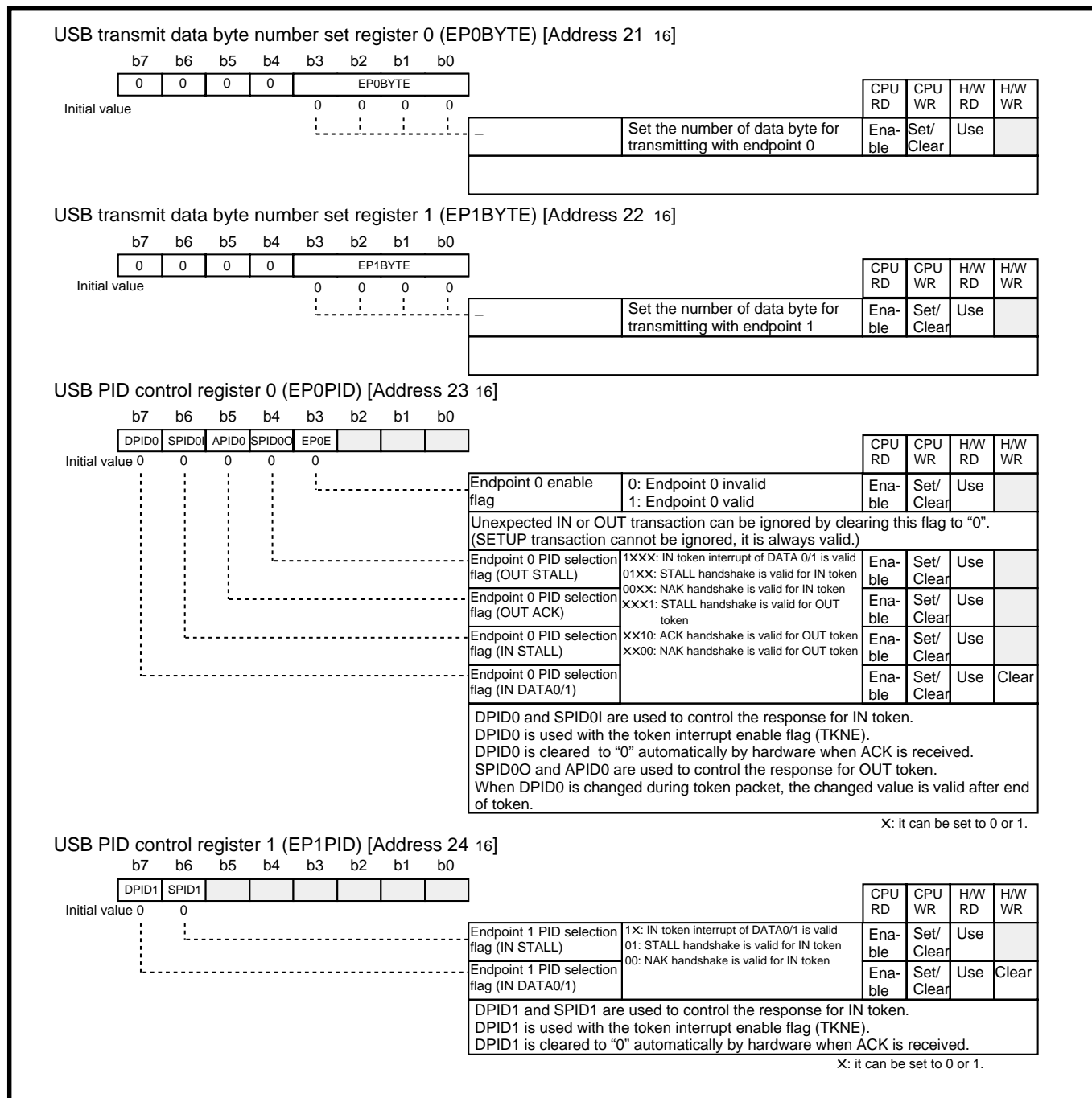


Fig. 2.4.9 Register structures relevant to USB (3)

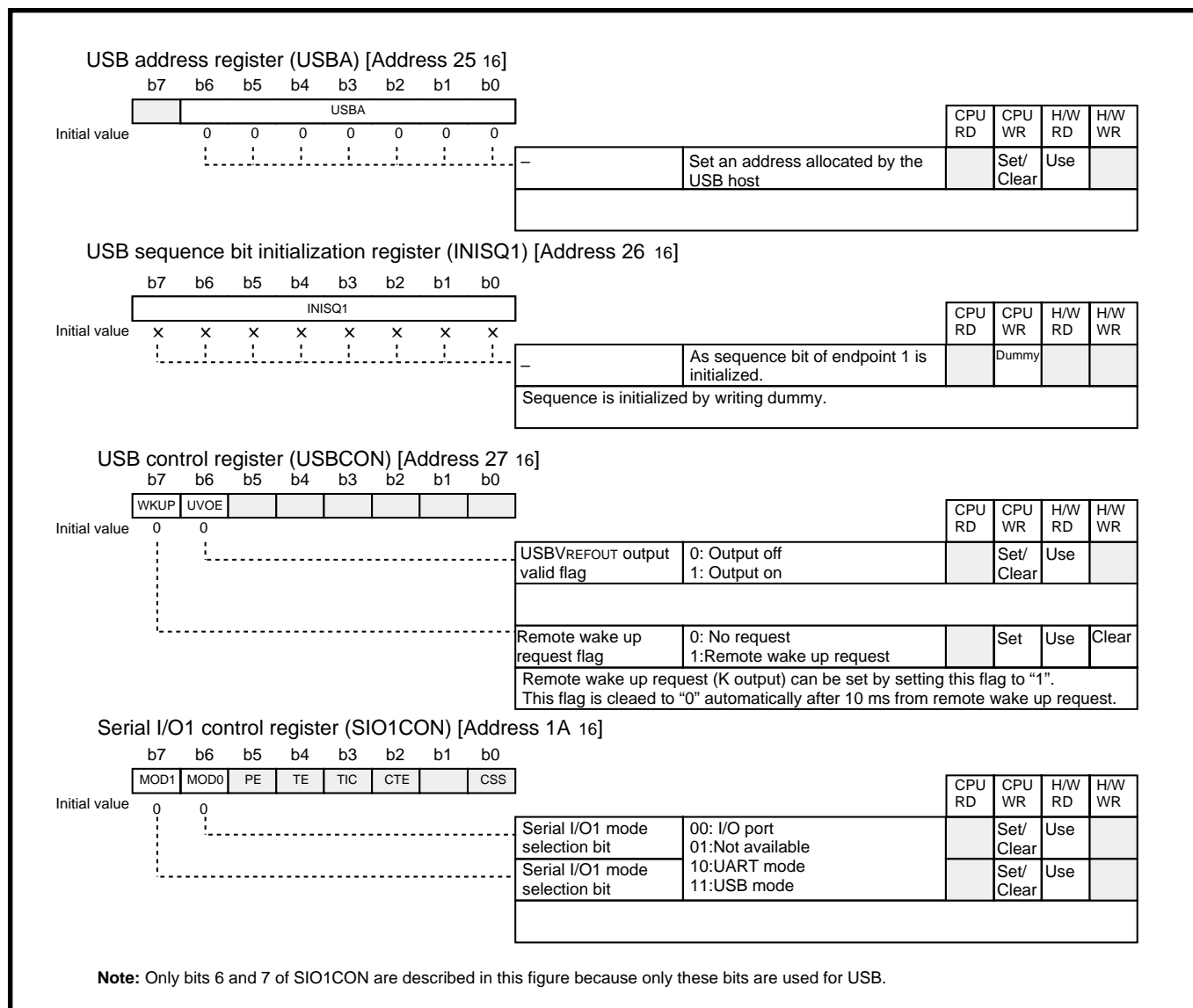


Fig. 2.4.10 Register structures relevant to USB (4)

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 7534 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 7534 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.

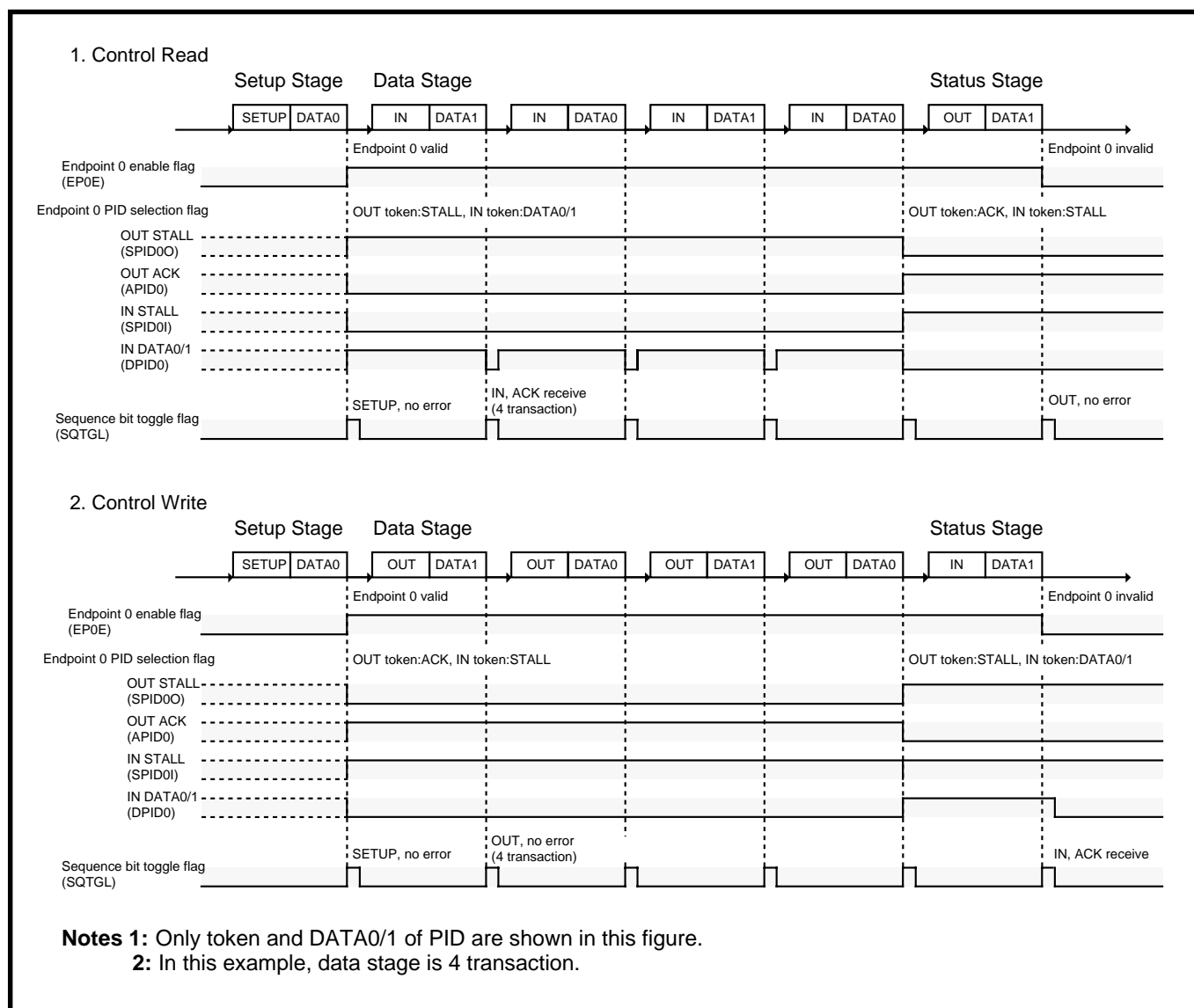


Fig. 2.4.11 Control method of control sequence

(2) Example of processing each transaction

In 7534 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.

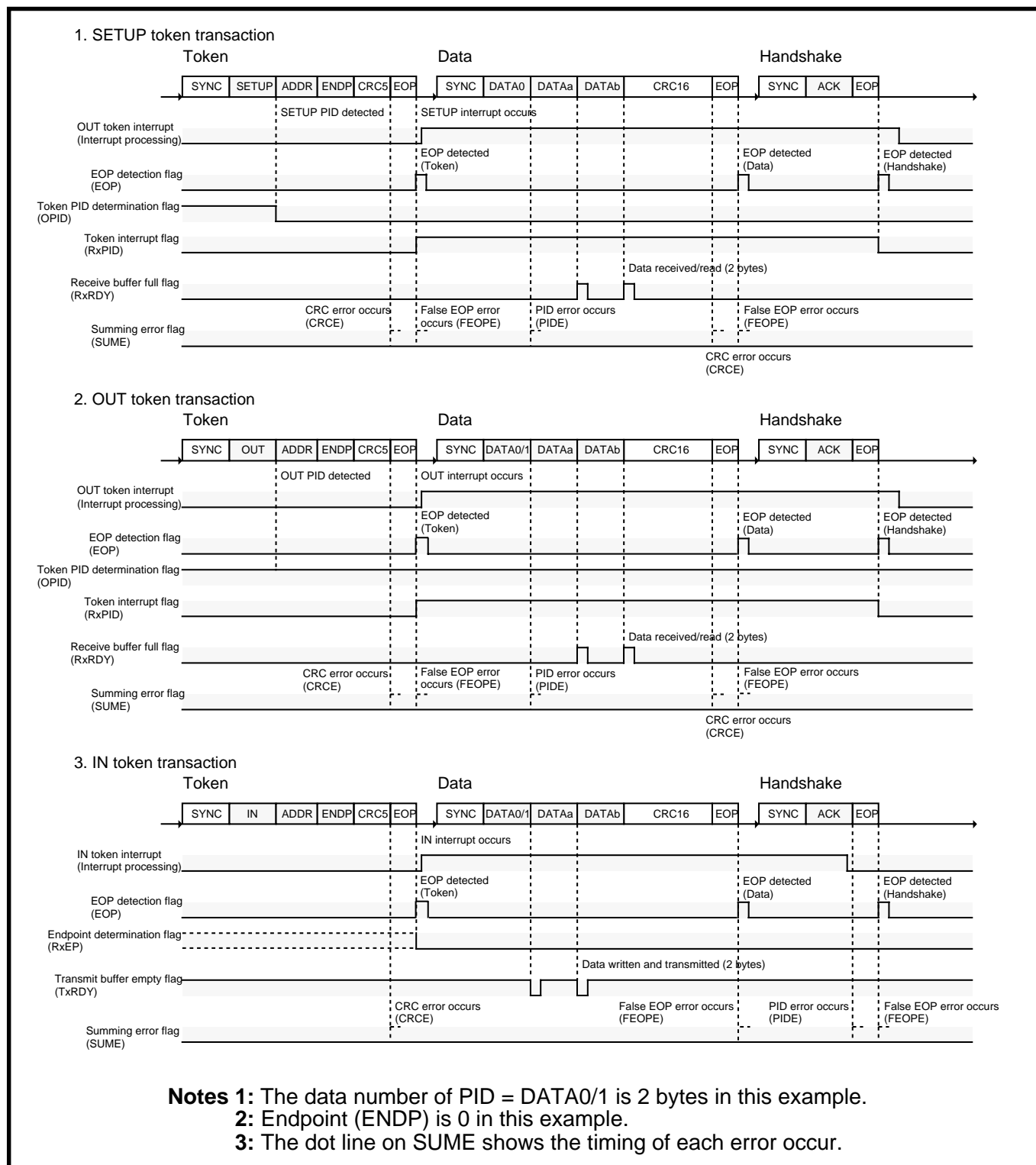


Fig. 2.4.12 Timing chart of the transaction according to each token

(3) Interrupt processing

In 7534 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.

In the OUT token interrupt, read data of OUT token and SETUP token at the timing shown in Figures 2.4.15 and 2.4.16. Also, in the IN token interrupt, write data to IN token (endpoint 0) and IN token (endpoint 1) at the timing shown in Figure 2.4.17.

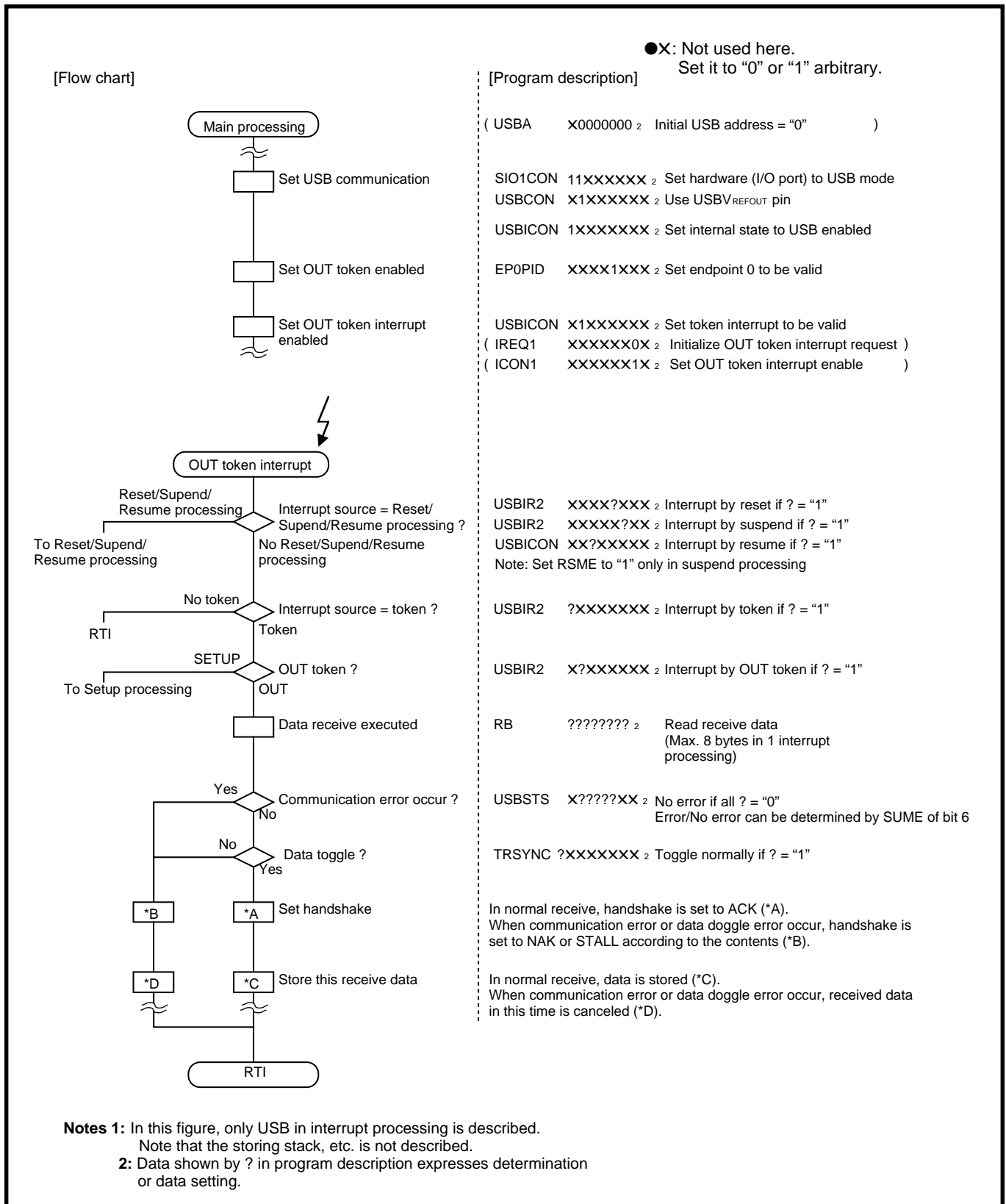


Fig. 2.4.13 USB interrupt processing example (OUT token)

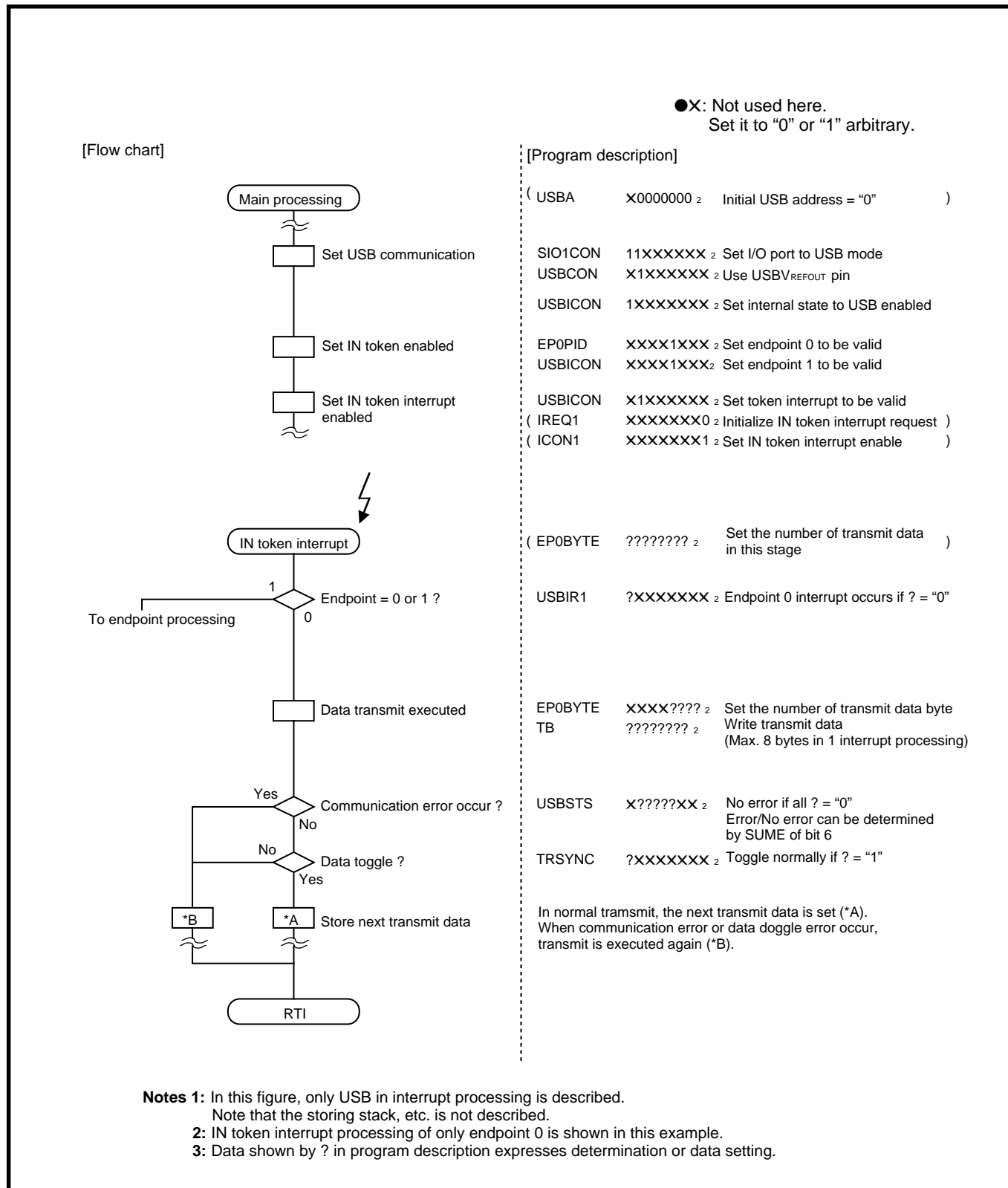


Fig. 2.4.14 USB interrupt processing example (IN token)

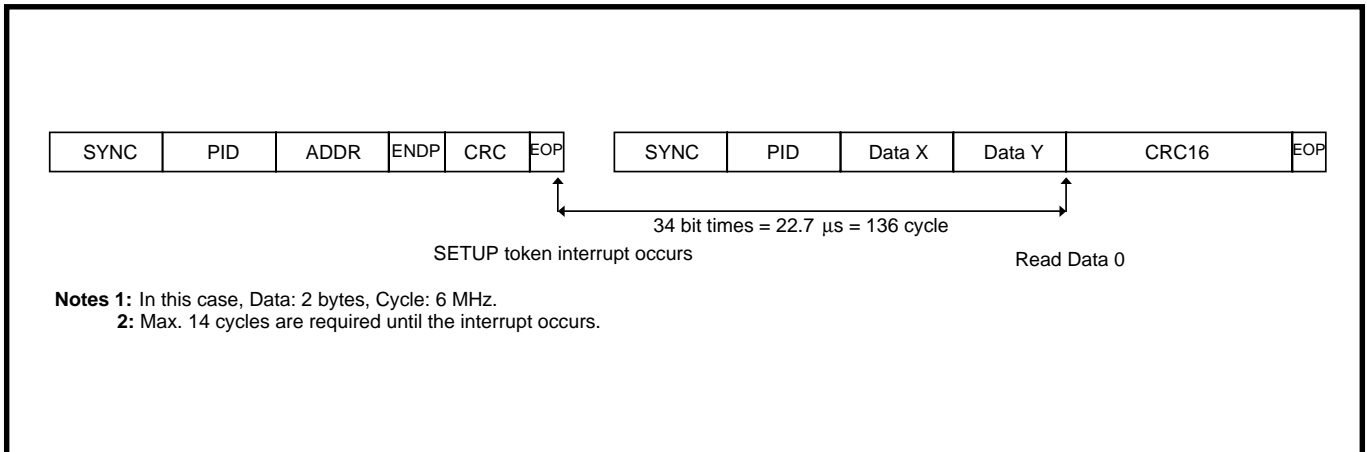


Fig. 2.4.15 Data read timing of SETUP token

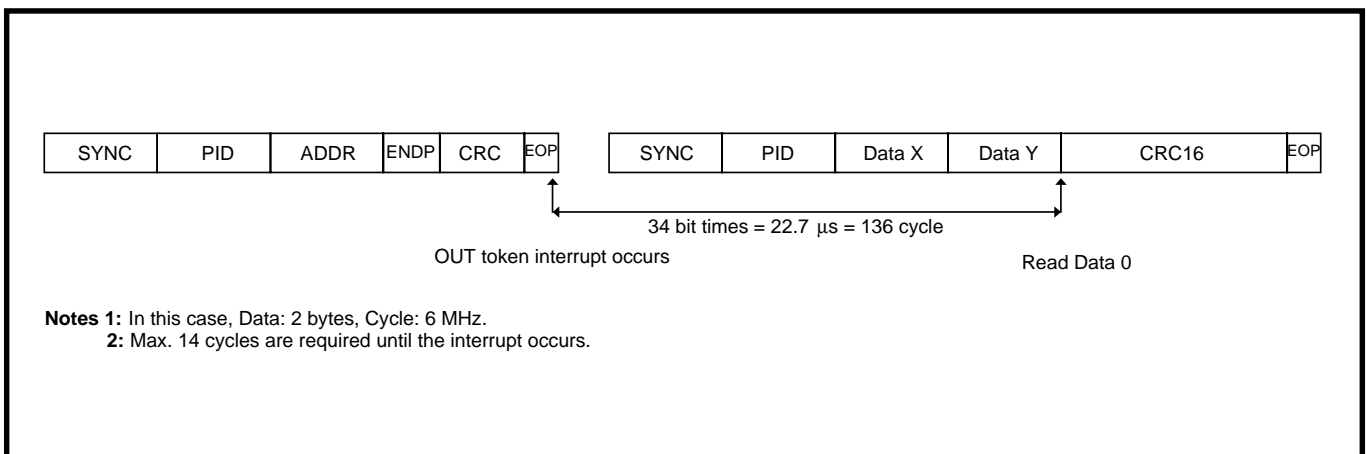


Fig. 2.4.16 Data read timing of OUT token

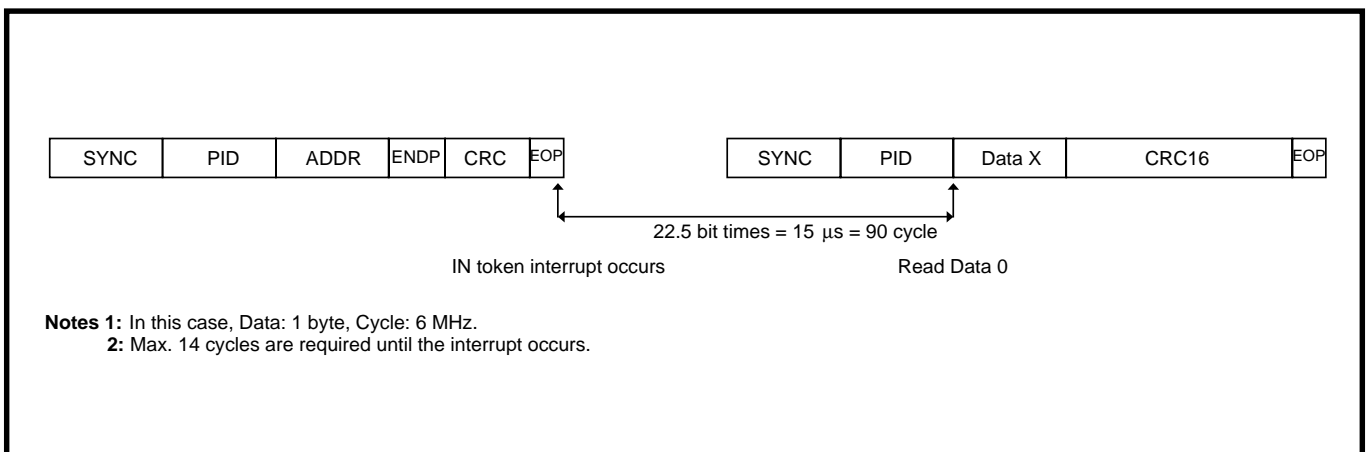


Fig. 2.4.17 Data read timing of IN token (endpoint 0) and IN token (endpoint 1) token

(4) Processing to special signal

In 7534 Group, control the USB function to the signal shown in Table 2.4.5 by software.

At USB reset, initialize the registers relevant to USB described in the above section 2.4.3.

Also, at Suspend, execute the **STP** instruction after the interrupt is enabled.

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 2.4.18 shows timing chart of each signal.

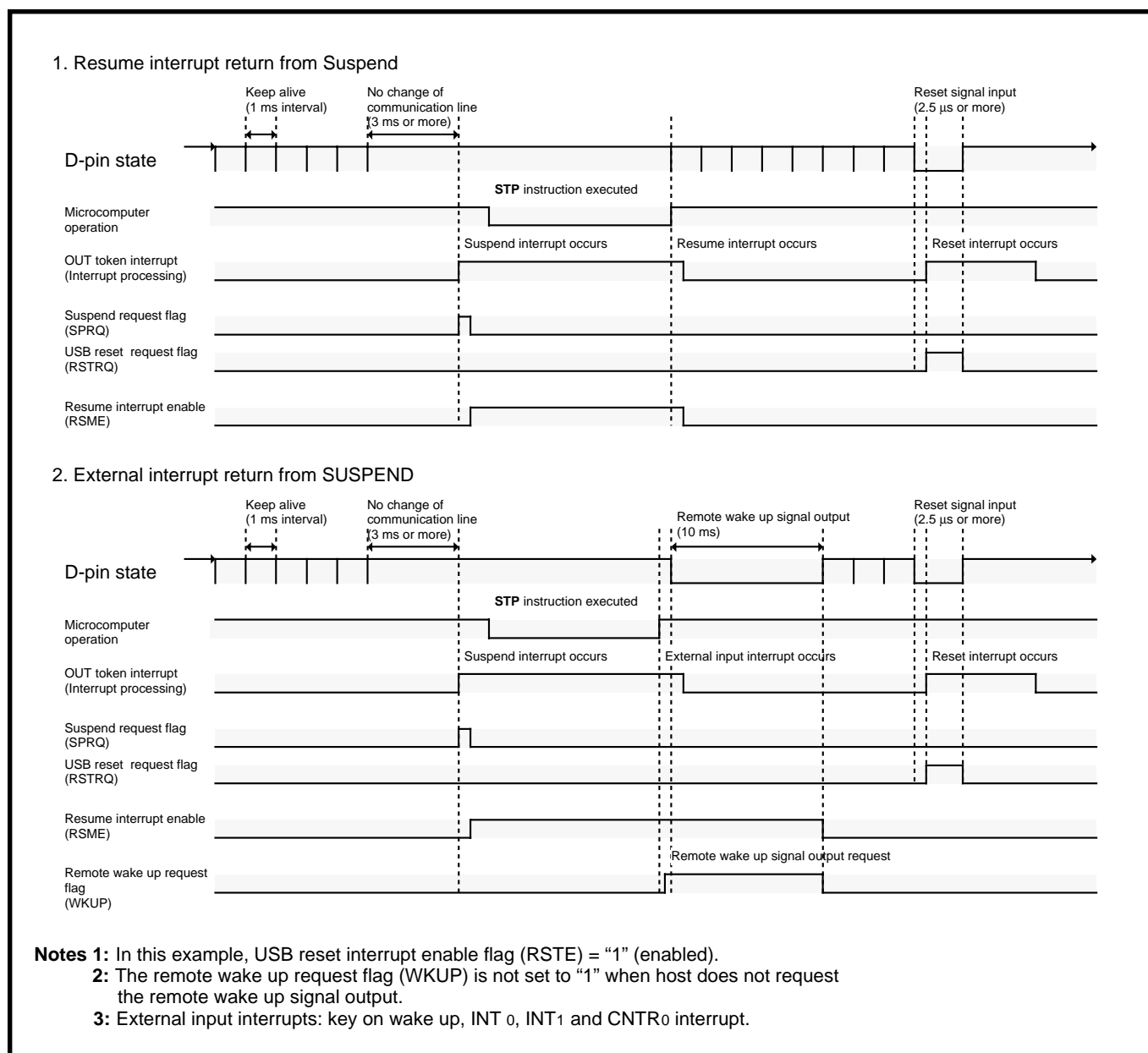


Fig. 2.4.18 Timing chart of each signal

2.4.5 Notes concerning USB

(1) Determination of interrupt condition in OUT token interrupt processing

Determine the occurrence of the reset/suspend/resume interrupt from other interrupt conditions in the order as shown in Figure 2.4.19 when they occur in the OUT token interrupt processing.

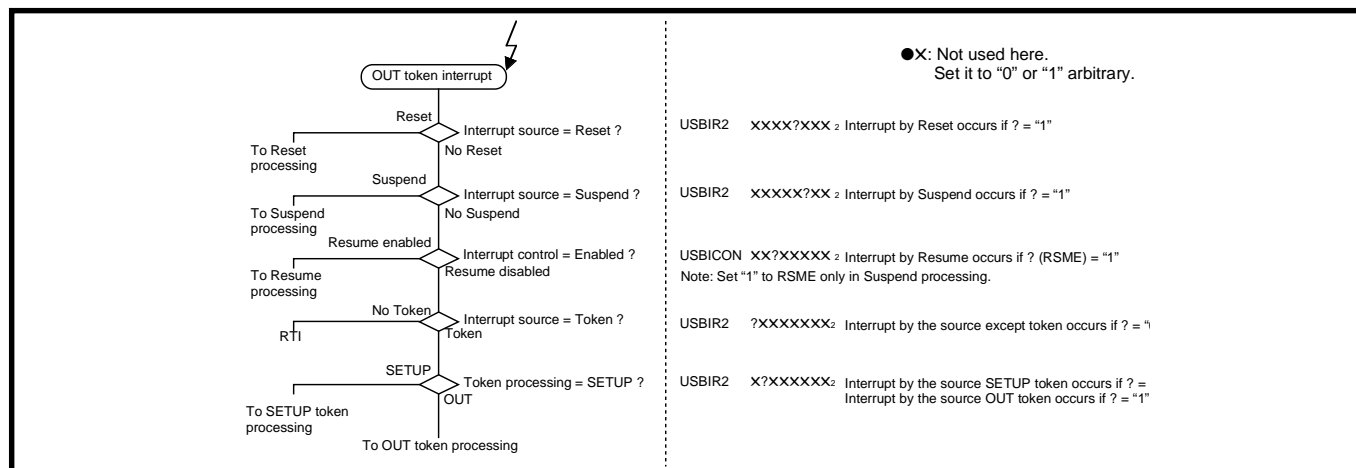


Fig. 2.4.19 Example for determination of resume interrupt

(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 μ 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 μ s or 79 machine cycle ($f(XIN) = 6$ MHz time) after this flag is "1".

(3) Determination of SE0 signal

In 7534 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 2.4.20 shows processing to the width of the signal according to the situation.

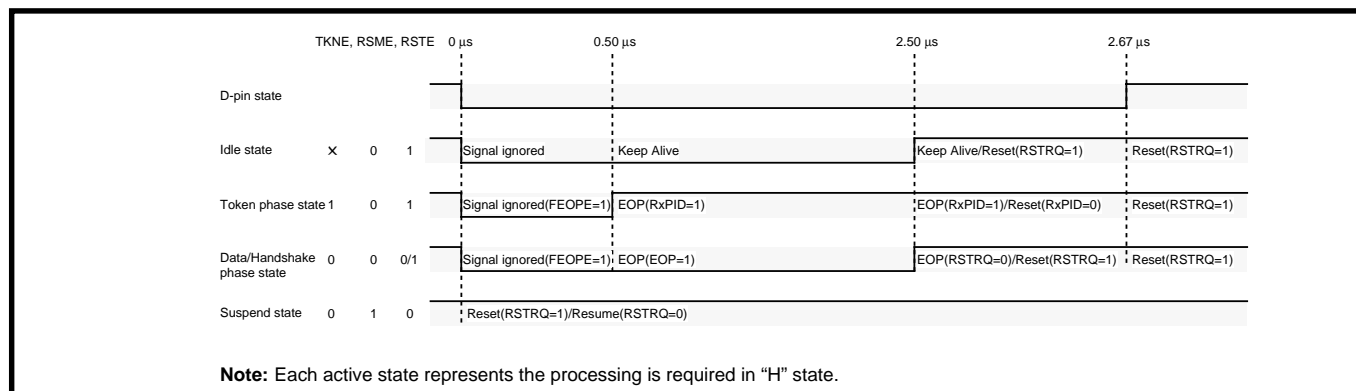


Fig. 2.4.20 Processing for width of SE0 signal

(4) USB communication

In applications requiring high-reliability, we recommend providing the system with protective measures such as USB function initialization by software or USB reset by the host to prevent USB communication from being terminated unexpectedly, for example due to external causes such as noise.

(5) USB suspend current

When USB suspend mode with TTL level on P10, P12, P13 input level selection bit (bit 3 of address 1716) set to "1", suspend current as I_{CC} might be greater than 300 μA as a spec.

[Countermeasure]

There are two countermeasures by software to avoid it as follows.

- (1) Change from TTL input level to CMOS input level for P10, P12, P13 port input.
- (2) Change from TTL input level to CMOS input level before STP instruction in suspend routine; then after RESUME or Remote wake up interrupt, return to TTL input level from CMOS input level. That is shown in Figure 2.4.21.

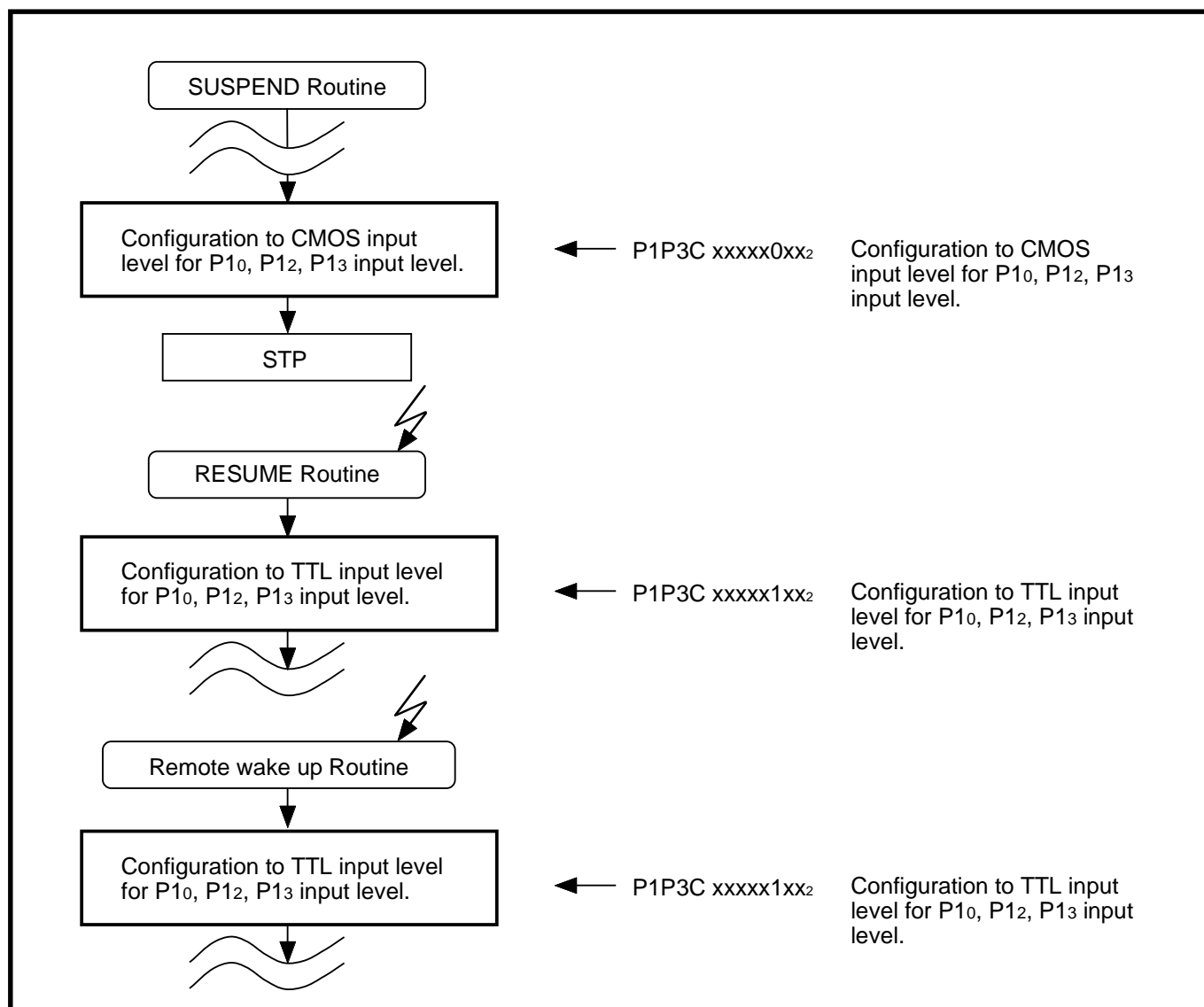


Fig. 2.4.21 Countermeasure (2) by software

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

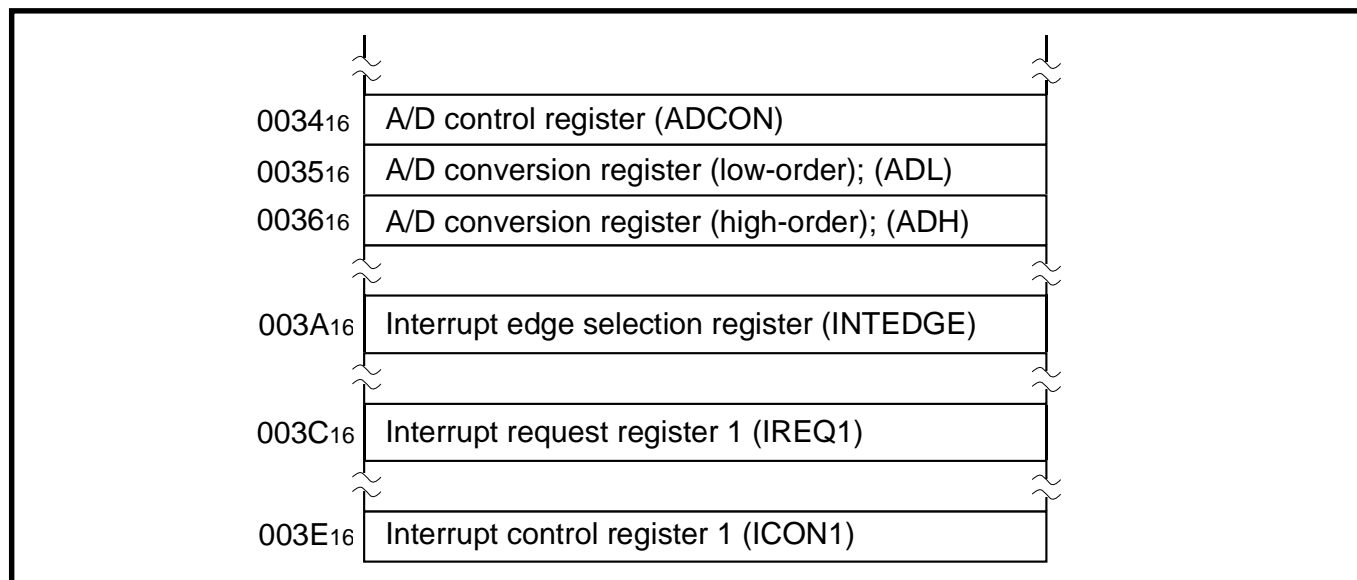


Fig. 2.5.1 Memory map of registers relevant to A/D converter

2.5.2 Relevant registers

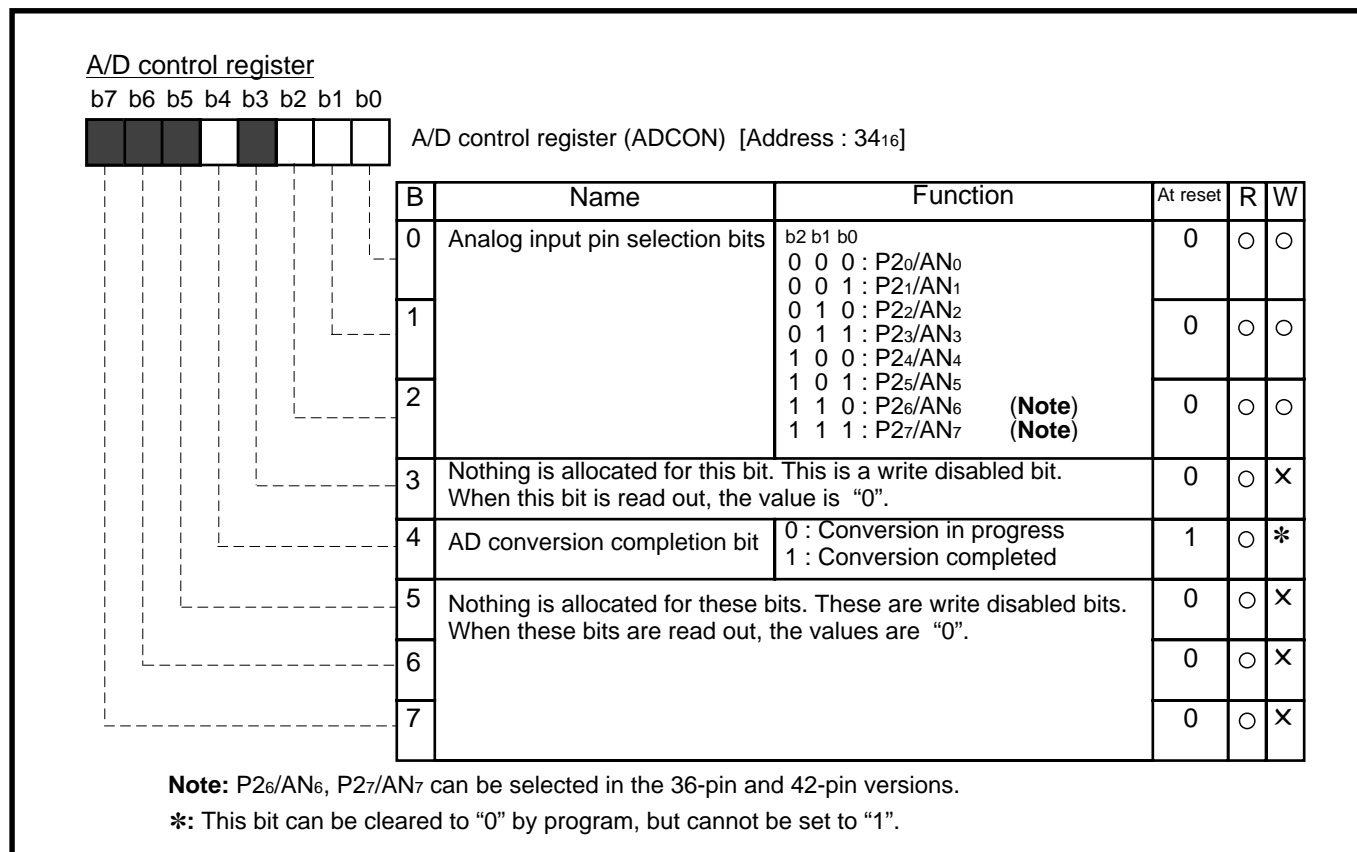


Fig. 2.5.2 Structure of A/D control register

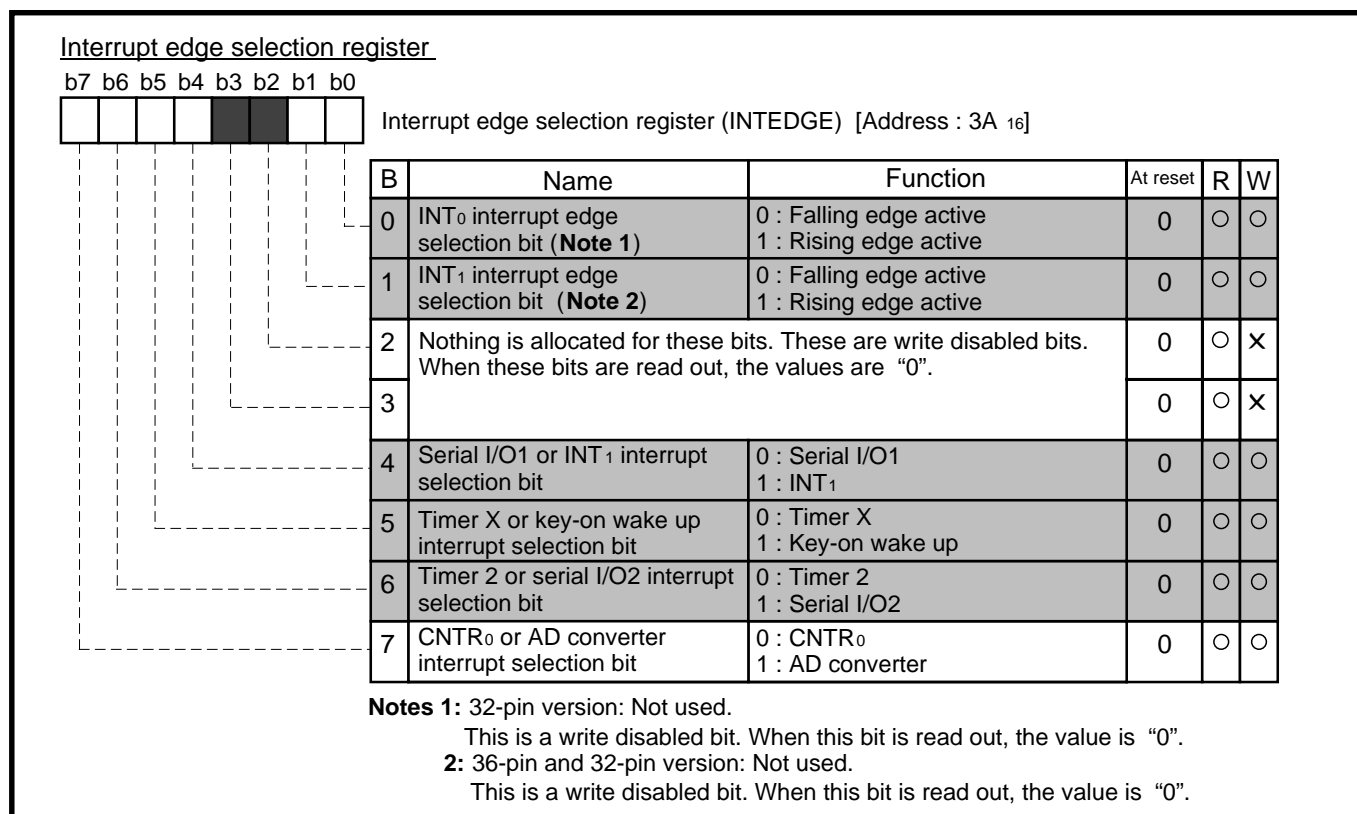


Fig. 2.5.5 Structure of Interrupt edge selection register

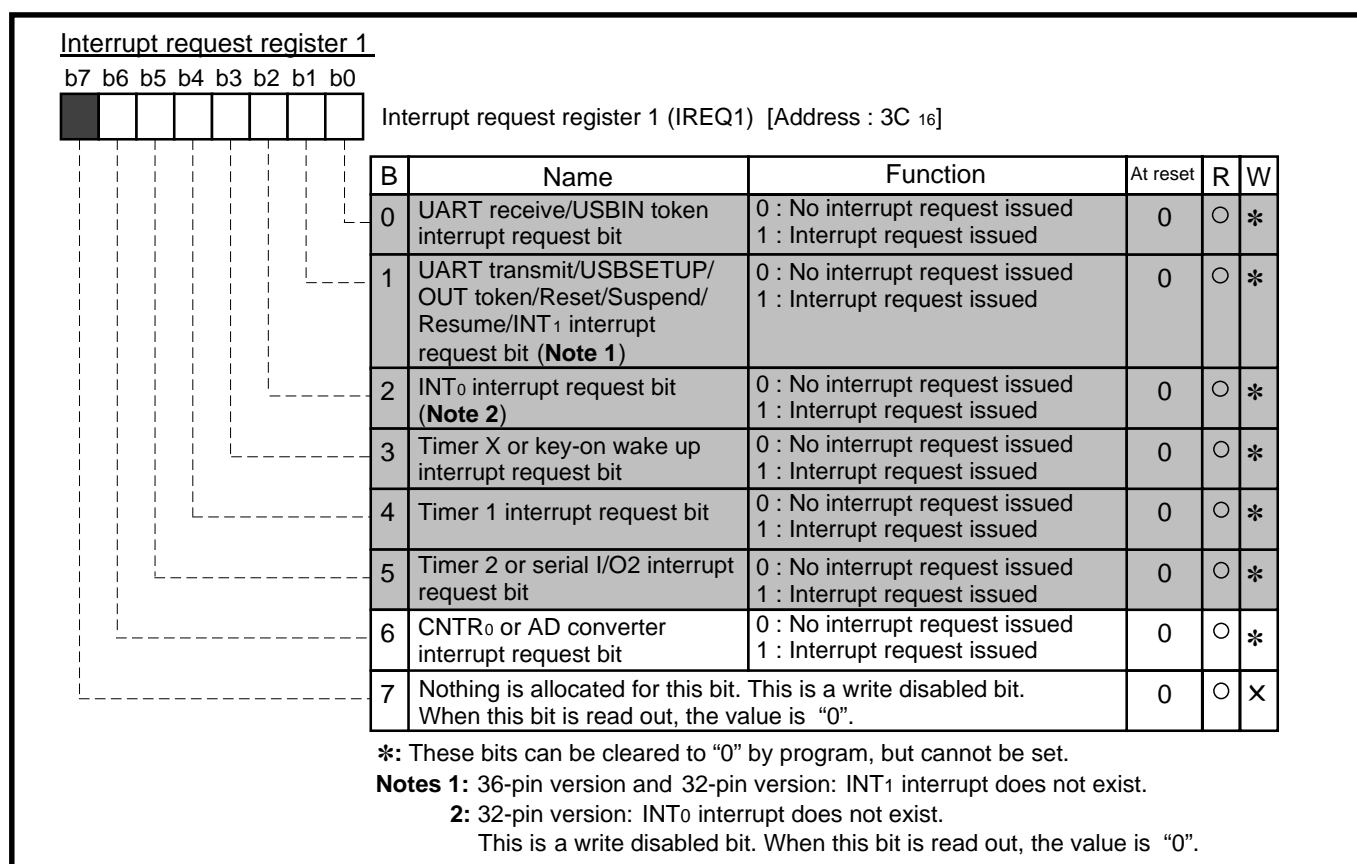


Fig. 2.5.6 Structure of Interrupt request register 1

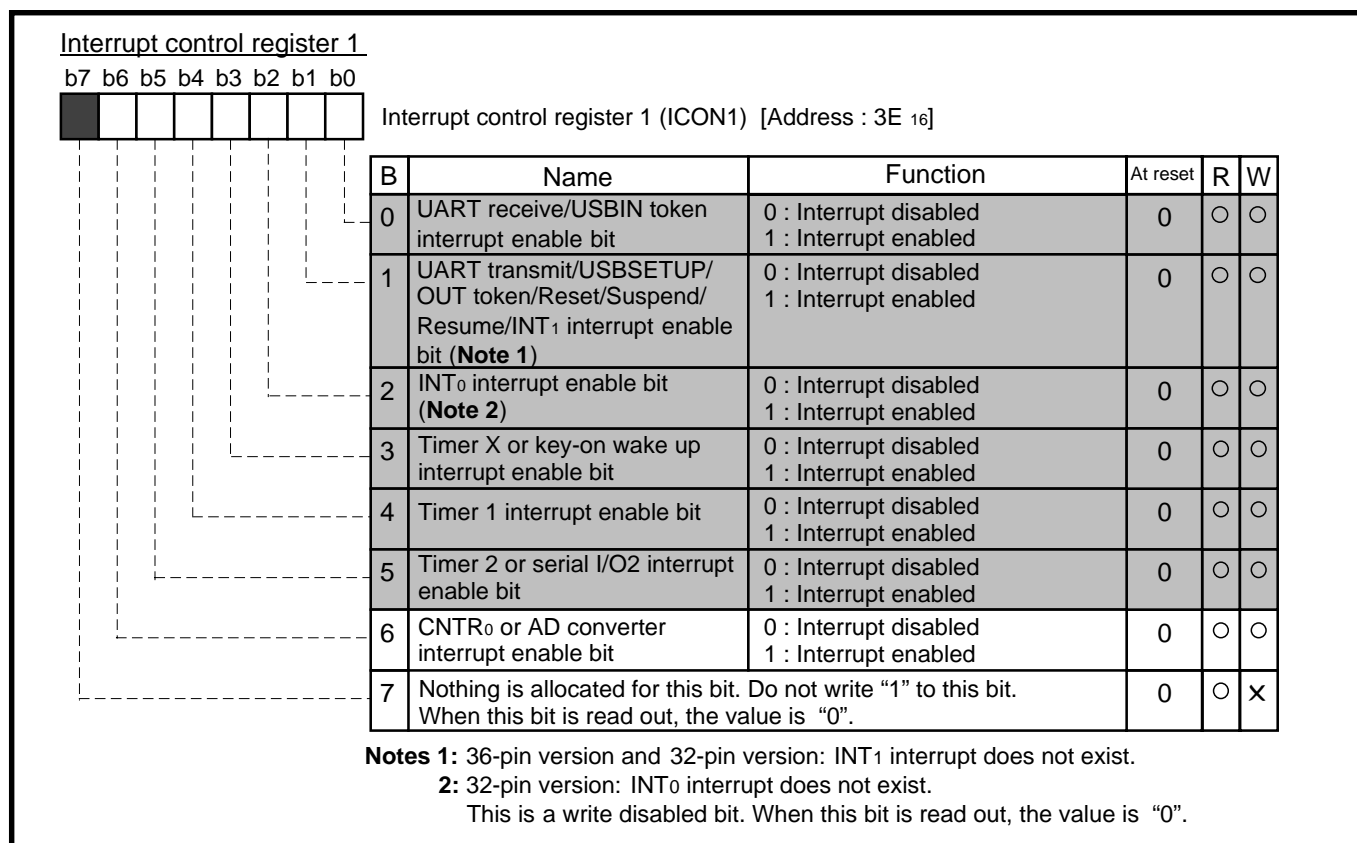


Fig. 2.5.7 Structure of Interrupt control register 1

2.5.3 A/D converter application examples

(1) Conversion of analog input voltage

Outline : The analog input voltage 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.

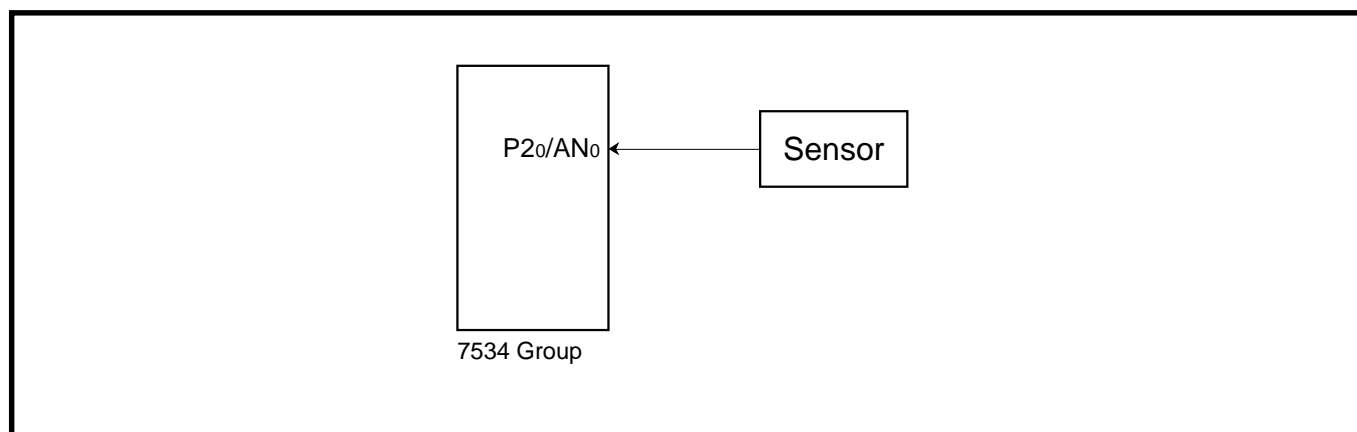


Fig. 2.5.8 Connection diagram

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

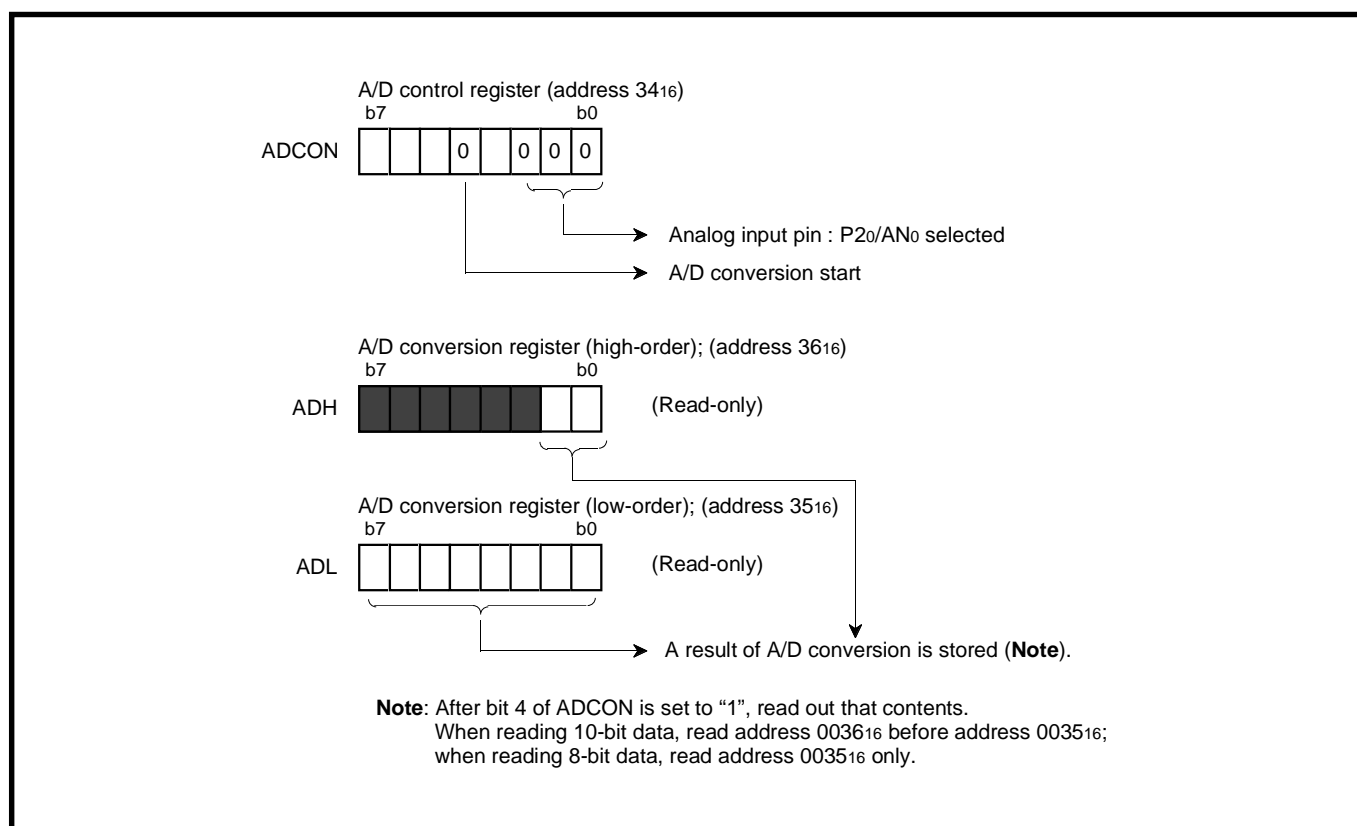


Fig. 2.5.9 Relevant registers setting

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.

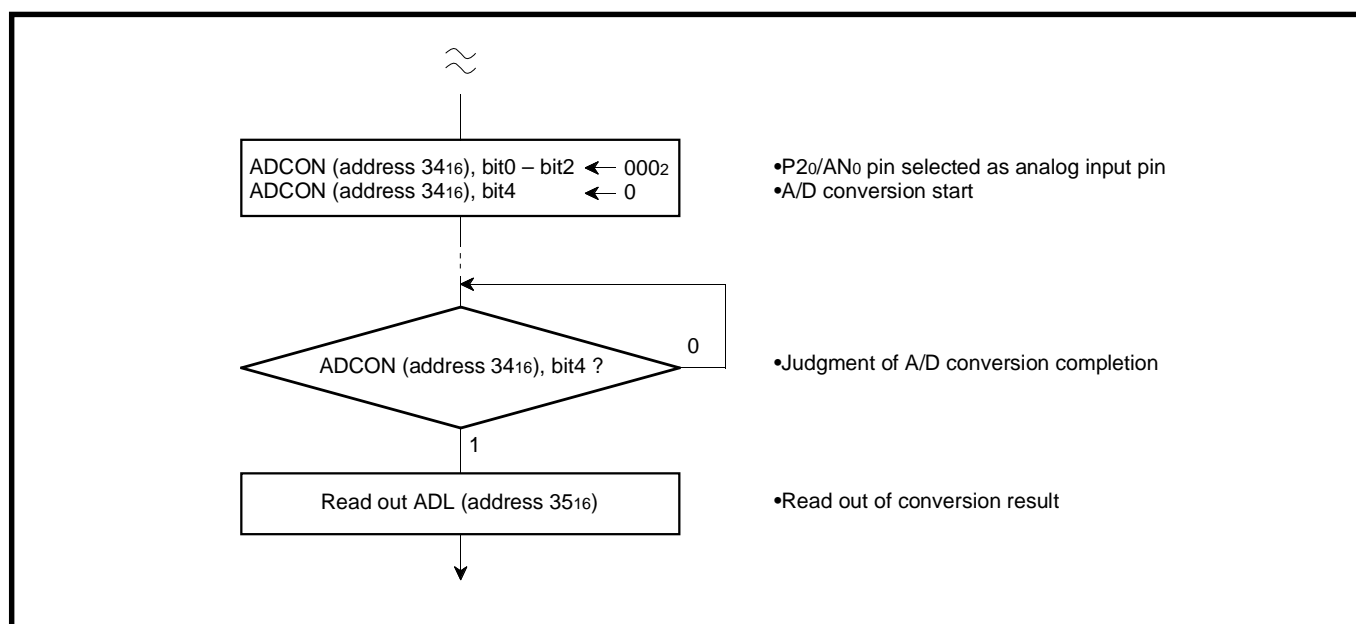


Fig. 2.5.10 Control procedure for 8-bit read

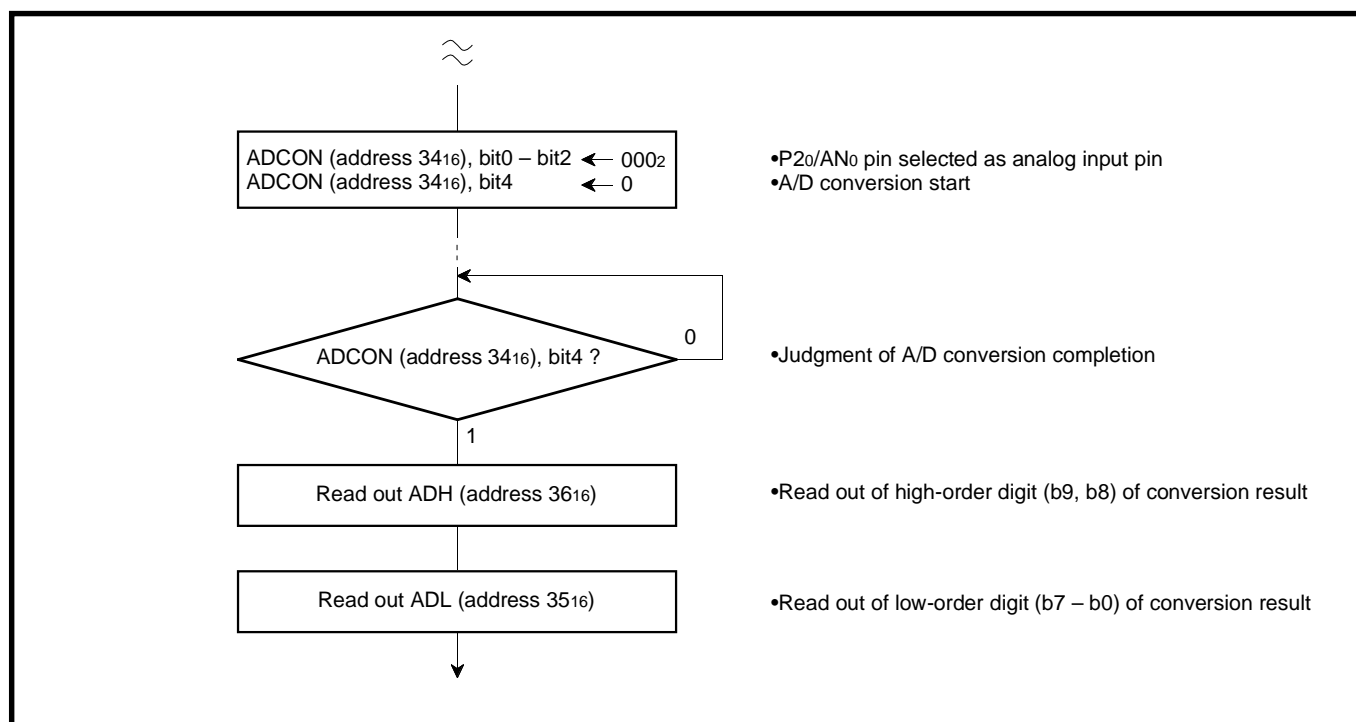


Fig. 2.5.11 Control procedure for 10-bit read

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 μ F to 1 μ F. Further, be sure to verify the operation of application products on the user side.

● Reason

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

(3) Method to stabilize A/D Converter

Method to stabilize A/D Converter is described below.

- (a) A/D conversion accuracy could be affected for Bus Powered^{*1} USB devices, while the communicating. Figure 2.5.12 shows the method to stabilize A/D conversion accuracy, inserting a capacitor between Vref and Vss.

*1: Power supplied by USB V_{CC} BUS.

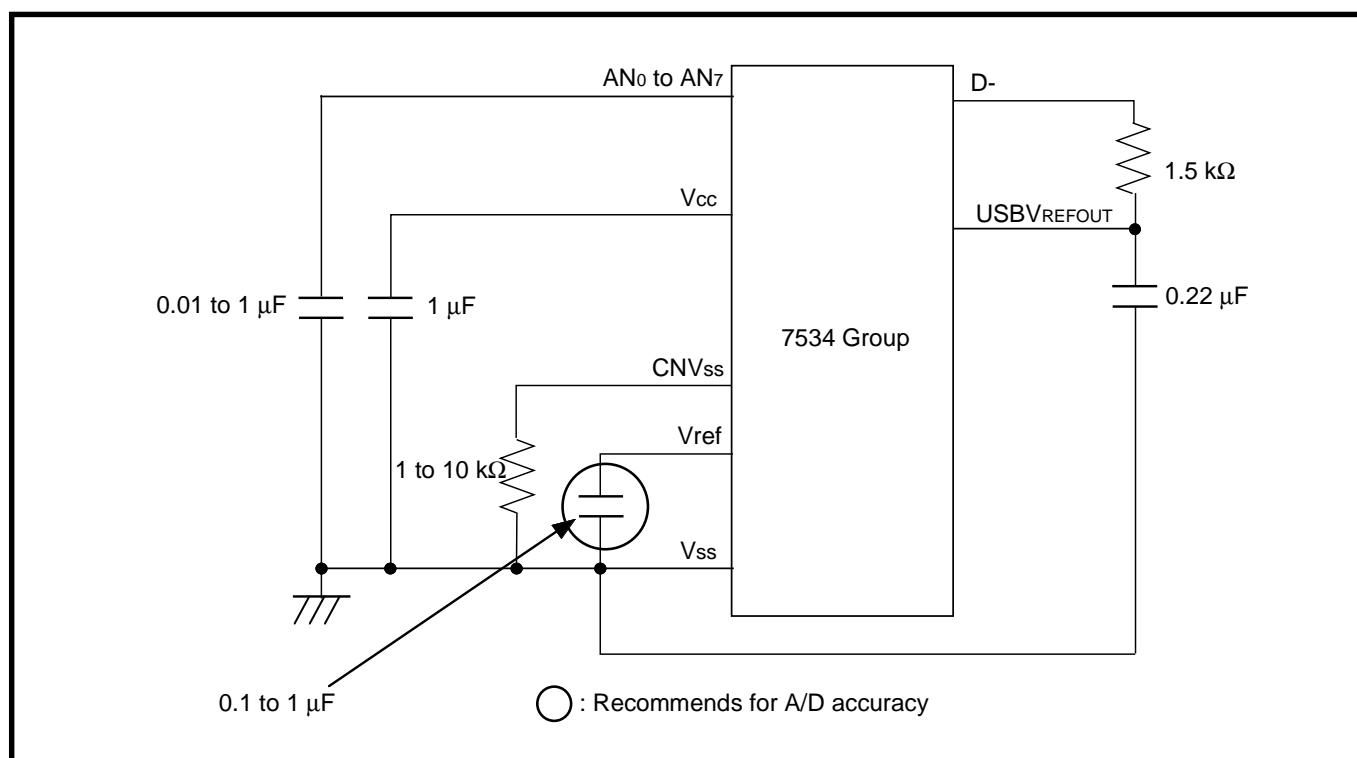


Fig. 2.5.12 Method to stabilize A/D conversion accuracy

- (b) It is recommended for A/D accuracy (b) to avoid converting while USB communication, and use average value of several converted values.

2.6 Reset

2.6.1 Connection example of reset IC

Figure 2.6.1 shows the system example to switch to the RAM back-up mode when detecting the falling of system power source by the INT interrupt.

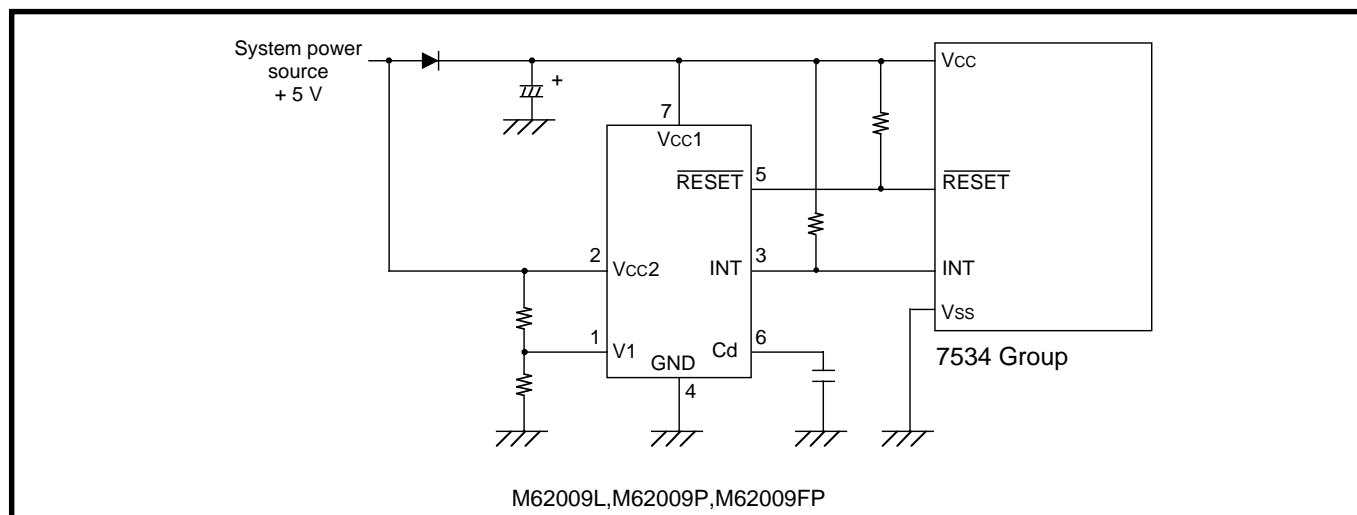


Fig. 2.6.1 Example of power on reset circuit

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.

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 Package outline**
- 3.7 List of instruction code**
- 3.8 Machine instructions**
- 3.9 SFR memory map**
- 3.10 Pin configurations**

3.1 Electrical characteristics

3.1.1 Absolute maximum ratings

Table 3.1.1 Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
VCC	Power source voltage	All voltages are based on VSS. Output transistors are cut off.	-0.3 to 7.0	V
Vi	Input voltage P00-P07, P10-P16, P20-P27, P30-P37, VREF, P40, P41		-0.3 to VCC + 0.3	V
Vi	Input voltage RESET, XIN		-0.3 to VCC + 0.3	V
Vi	Input voltage CNVSS (Note 1)		-0.3 to 13	V
VO	Output voltage P00-P07, P10-P16, P20-P27, P30-P37, XOUT, USBVREFOUT, P40, P41		-0.3 to VCC + 0.3	V
Pd	Power dissipation (Note 2)	Ta = 25°C	1000 (Note 3)	mW
Topr	Operating temperature		-20 to 85	°C
Tstg	Storage temperature		-40 to 125	°C

Notes 1: It is a rating only for the One Time PROM version. Connect to VSS for mask ROM version.

2: The rating value depends on packages.

3: This is the value for the 42-pin version.

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

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

3.1.2 Recommended operating conditions

Table 3.1.2 Recommended operating conditions
(V_{CC} = 4.1 to 5.5 V, T_a = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V _{CC}	Power source voltage	f(XIN) = 6 MHz	4.1	5.0	5.5	V
V _{SS}	Power source voltage			0		V
V _{REF}	Analog reference voltage		2.0		V _{CC}	V
V _{IH}	"H" input voltage	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41	0.8 V _{CC}		V _{CC}	V
V _{IH}	"H" input voltage (TTL input level selected)	P10, P12, P13, P36, P37	2.0		V _{CC}	V
V _{IH}	"H" input voltage	$\overline{\text{RESET}}$, XIN	0.8 V _{CC}		V _{CC}	V
V _{IH}	"H" input voltage	D+, D-	2.0		3.6	V
V _{IL}	"L" input voltage	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41	0		0.3 V _{CC}	V
V _{IL}	"L" input voltage (TTL input level selected)	P10, P12, P13, P36, P37	0		0.8	V
V _{IL}	"L" input voltage	$\overline{\text{RESET}}$, CNV _{SS}	0		0.2 V _{CC}	V
V _{IL}	"L" input voltage	D+, D-	0		0.8	V
V _{IL}	"L" input voltage	XIN	0		0.16V _{CC}	V
I _{OH(peak)}	"H" total peak output current (Note 1)	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41			-80	mA
I _{OL(peak)}	"L" total peak output current (Note 1)	P00–P07, P10–P16, P20–P27, P37, P40, P41			80	mA
I _{OL(peak)}	"L" total peak output current (Note 1)	P30–P36			60	mA
I _{OH(avg)}	"H" total average output current (Note 1)	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41			-40	mA
I _{OL(avg)}	"L" total average output current (Note 1)	P00–P07, P10–P16, P20–P27, P37, P40, P41			40	mA
I _{OL(avg)}	"L" total average output current (Note 1)	P30–P36			30	mA
I _{OH(peak)}	"H" peak output current (Note 2)	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41			-10	mA
I _{OL(peak)}	"L" peak output current (Note 2)	P00–P07, P10–P16, P20–P27, P37, P40, P41			10	mA
I _{OL(peak)}	"L" peak output current (Note 2)	P30–P36			30	mA
I _{OH(avg)}	"H" average output current (Note 3)	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41			-5	mA
I _{OL(avg)}	"L" average output current (Note 3)	P00–P07, P10–P16, P20–P27, P37, P40, P41			5	mA
I _{OL(avg)}	"L" average output current (Note 3)	P30–P36			15	mA
f(XIN)	Oscillation frequency (Note 4) at ceramic oscillation or external clock input	V _{CC} = 4.1 to 5.5 V 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 a term of 100 ms. The total peak current is the peak value of all the currents.

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

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

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

3.1.3 Electrical characteristics

Table 3.1.3 Electrical characteristics (1) (VCC = 4.1 to 5.5 V, VSS = 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.

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 _{CC}	Power source current	Double-speed mode, $f(X_{IN}) = 6$ MHz, Output transistors "off"		6	10	mA	
		$f(X_{IN}) = 6$ MHz, (in WIT state) Output transistors "off"		1.6	3.2	mA	
		Increment when A/D conversion is executed $f(X_{IN}) = 6$ MHz, $V_{CC} = 5$ V		0.8		mA	
		All oscillation stopped (in STP state) Output transistors "off"	$T_a = 25$ °C		0.1	1.0	μA
			$T_a = 85$ °C			10	μA
$V_{CC} = 4.4$ V to 5.25 V Oscillation stopped in USB mode USB (SUSPEND), (pull-up resistor output included) (Fig. 3.1.1)	$T_a = 0$ 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	$V_{CC} = 4.1$ to 5.5 V $T_a = 25$ °C			±3	LSB
—	Differential nonlinear error	$V_{CC} = 4.1$ to 5.5 V $T_a = 25$ °C			±0.9	LSB
VOT	Zero transition voltage	$V_{CC} = V_{REF} = 5.12$ V	0	5	20	mV
VFST	Full scale transition voltage	$V_{CC} = V_{REF} = 5.12$ V	5105	5115	5125	mV
tCONV	Conversion time				122	tc(X _{IN})
RLADDER	Ladder resistor			55		kΩ
I _{VREF}	Reference power source input current	$V_{REF} = 5.0$ V	50	150	200	μA
		$V_{REF} = 3.0$ V	30	70	120	
I _{I(AD)}	A/D port input current				5.0	μA

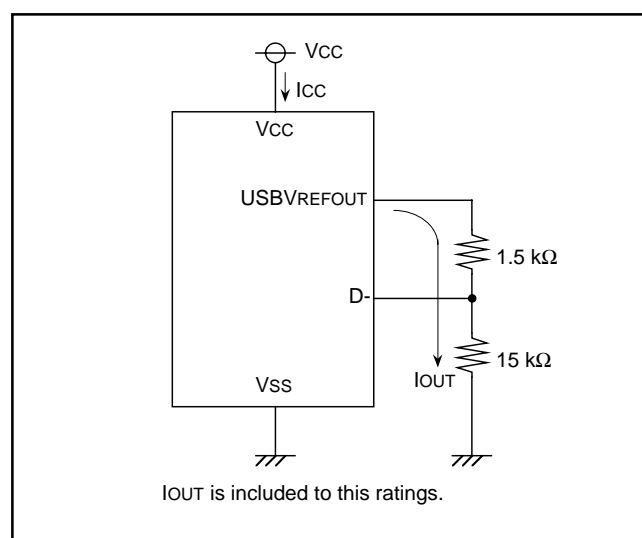


Fig. 3.1.1 Power source current measurement circuit in USB mode at oscillation stop

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.	
t _w (RESET)	Reset input "L" pulse width	15			μs
t _c (XIN)	External clock input cycle time	166			ns
t _{WH} (XIN)	External clock input "H" pulse width	70			ns
t _{WL} (XIN)	External clock input "L" pulse width	70			ns
t _c (CNTR)	CNTR ₀ input cycle time	200			ns
t _{WH} (CNTR)	CNTR ₀ , INT ₀ , INT ₁ input "H" pulse width	80			ns
t _{WL} (CNTR)	CNTR ₀ , INT ₀ , INT ₁ input "L" pulse width	80			ns
t _c (SCLK)	Serial I/O2 clock input cycle time	1000			ns
t _{WH} (SCLK)	Serial I/O2 clock input "H" pulse width	400			ns
t _{WL} (SCLK)	Serial I/O2 clock input "L" pulse width	400			ns
t _{su} (SDATA-SCLK)	Serial I/O2 input set up time	200			ns
t _h (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.	
t _{WH} (SCLK)	Serial I/O2 clock output "H" pulse width	t _c (SCLK)/2-30			ns
t _{WL} (SCLK)	Serial I/O2 clock output "L" pulse width	t _c (SCLK)/2-30			ns
t _d (SCLK-SDATA)	Serial I/O2 output delay time			140	ns
t _v (SCLK-SDATA)	Serial I/O2 output valid time	0			ns
t _r (SCLK)	Serial I/O2 clock output rising time			30	ns
t _f (SCLK)	Serial I/O2 clock output falling time			30	ns
t _r (CMOS)	CMOS output rising time (Note)		10	30	ns
t _f (CMOS)	CMOS output falling time (Note)		10	30	ns
t _r (D+), t _r (D-)	USB output rising time, C _L = 200 to 450 pF, T _a = 0 to 70 °C, V _{CC} = 4.4 to 5.25 V	75	150	300	ns
t _f (D+), t _f (D-)	USB output falling time, C _L = 200 to 450 pF, T _a = 0 to 70 °C, V _{CC} = 4.4 to 5.25 V	75	150	300	ns

Notes: XOUT pin is excluded.

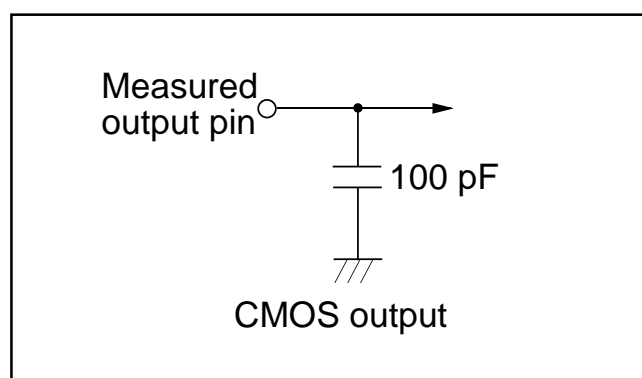


Fig. 3.1.2 Output switching characteristics measurement circuit

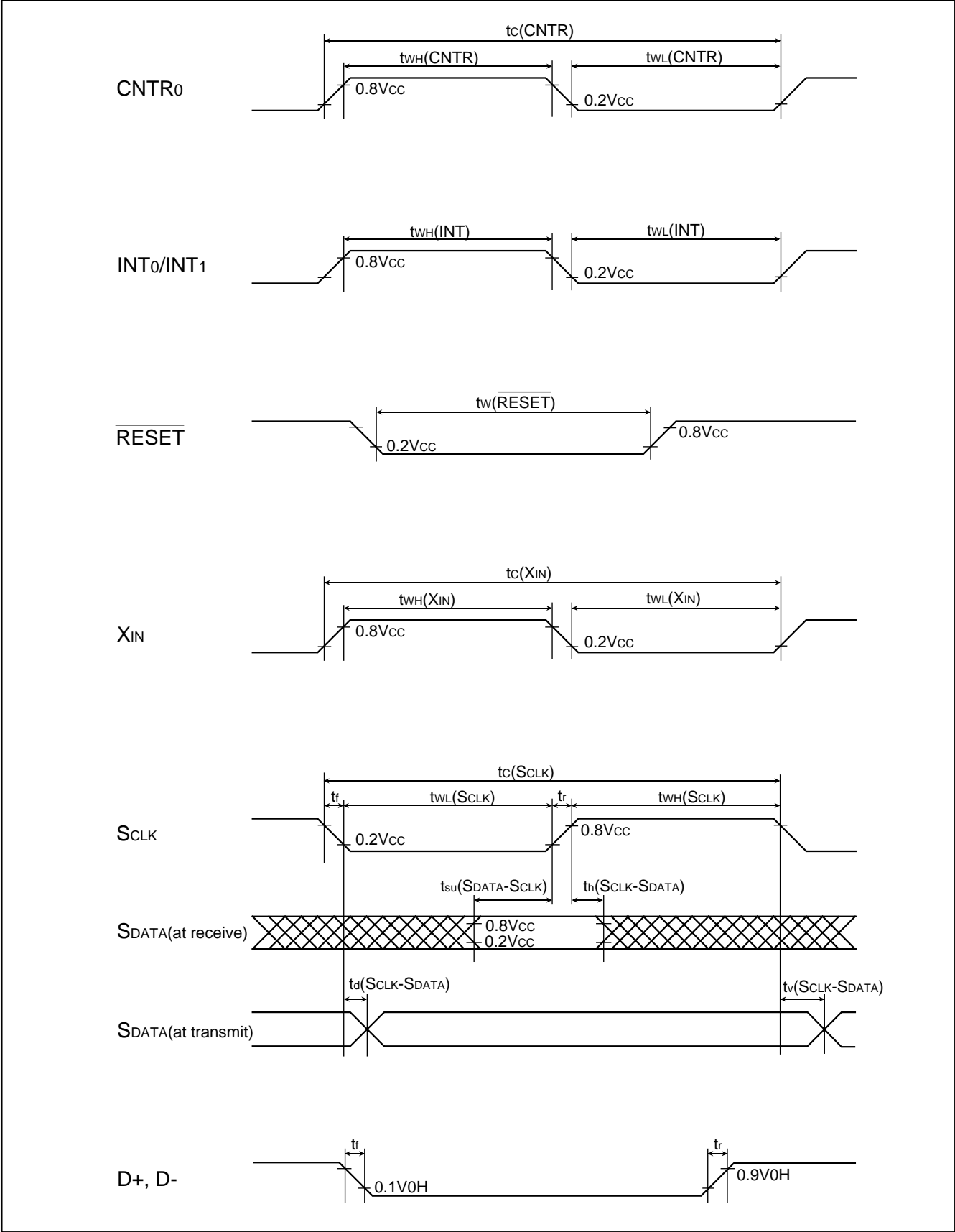


Fig. 3.1.3 Timing chart

3.2 Typical characteristics

3.2.1 Power source current characteristic example (I_{CC}-V_{CC} characteristic)

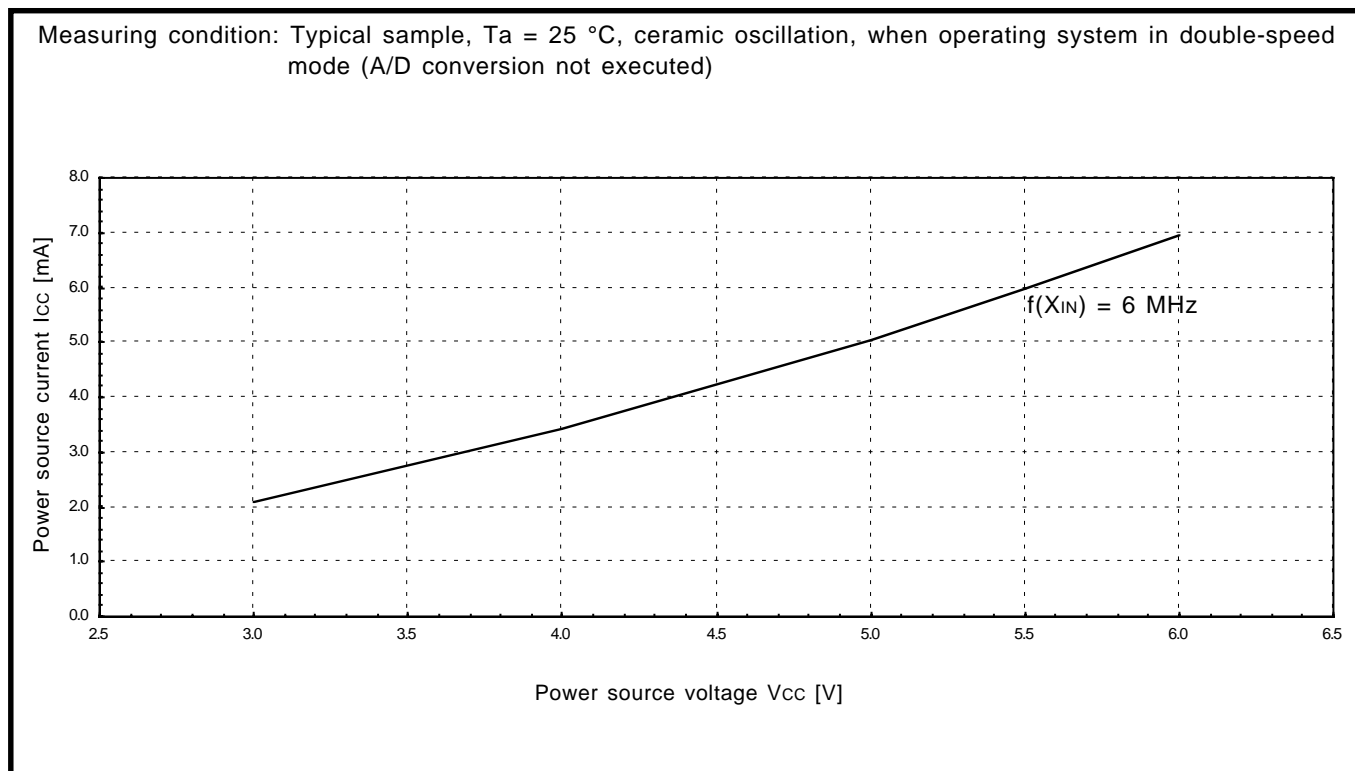


Fig. 3.2.1 I_{CC}-V_{CC} characteristic example (in double-speed mode)

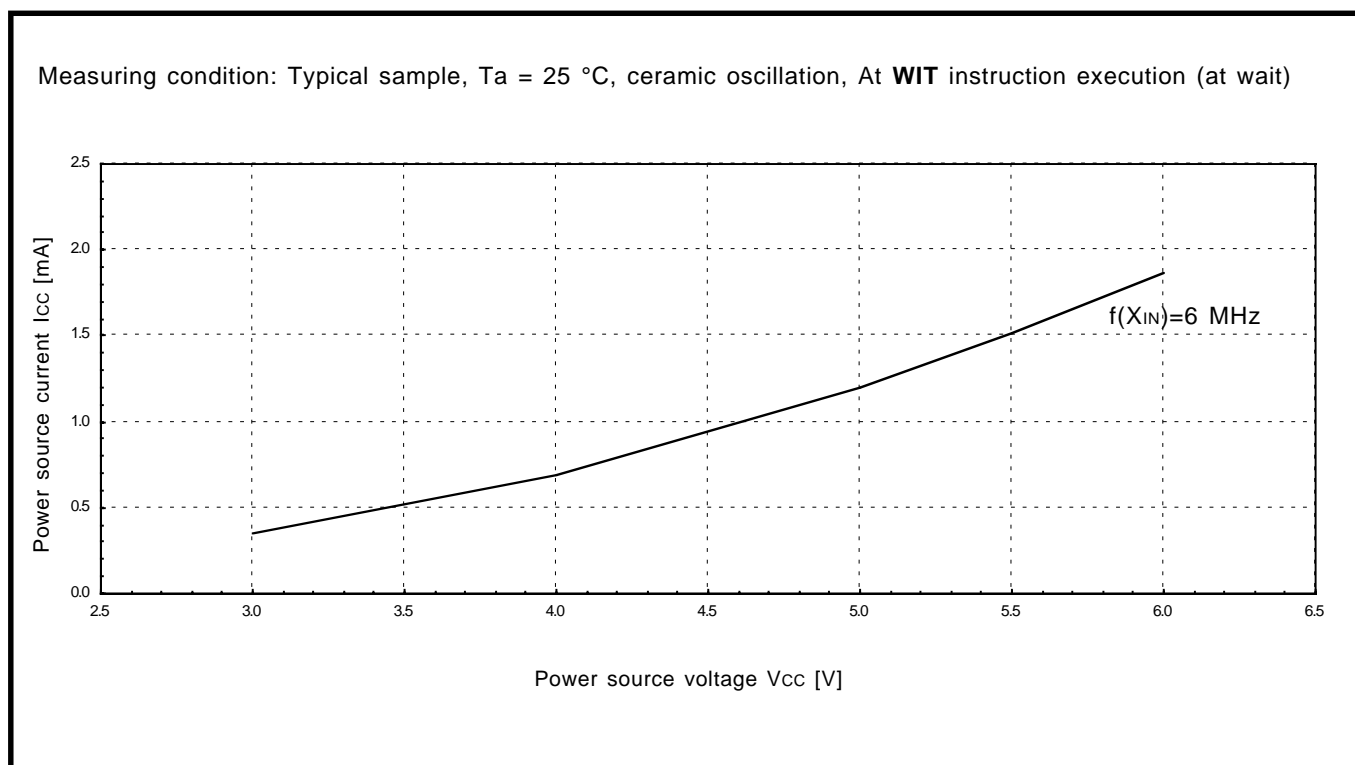


Fig. 3.2.2 I_{CC}-V_{CC} characteristic example (at WIT instruction execution)

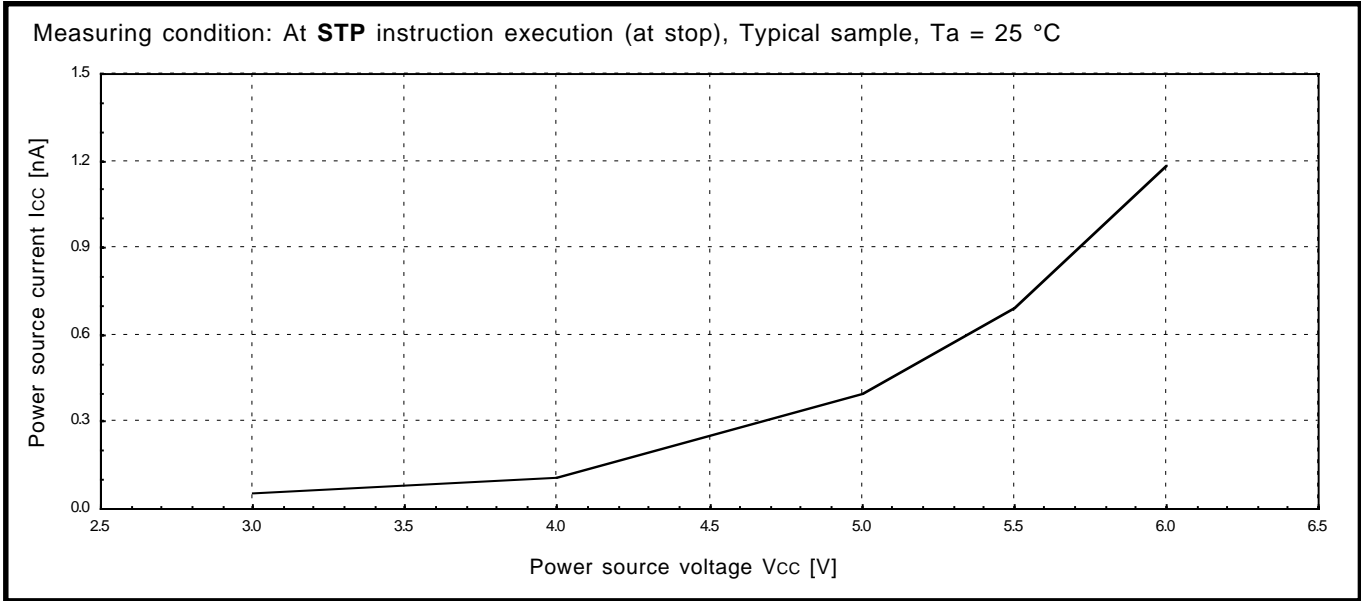


Fig. 3.2.3 Icc-Vcc characteristic example (At STP instruction execution, Ta = 25 °C)

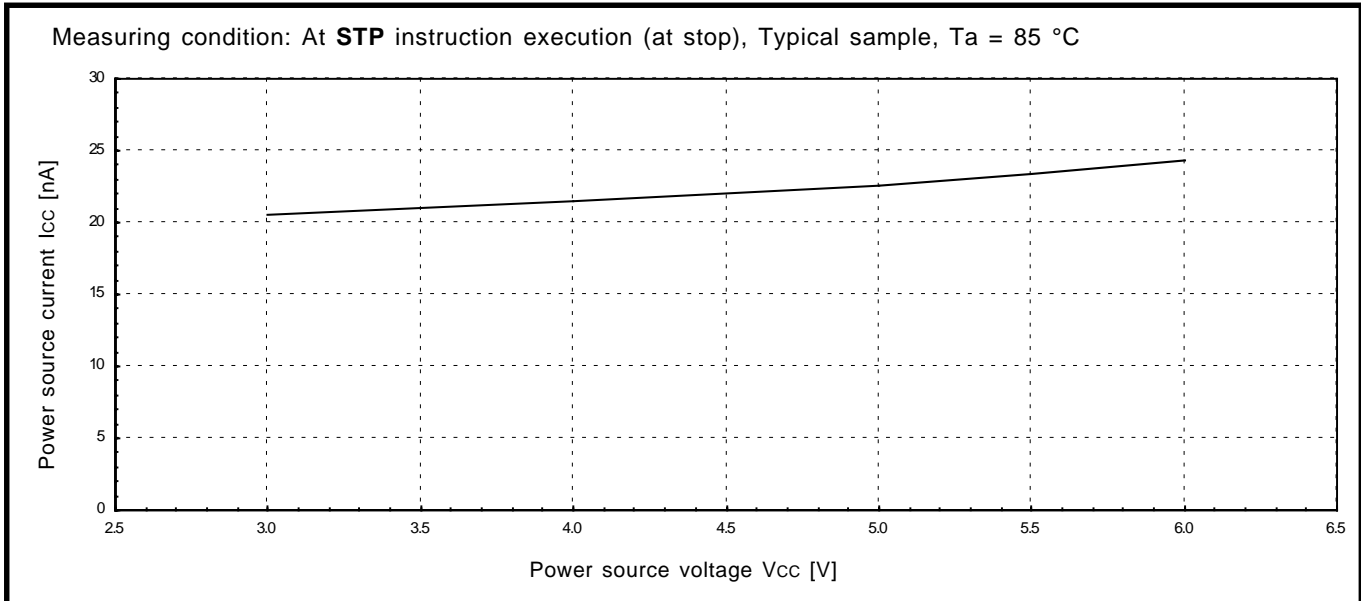


Fig. 3.2.4 Icc-Vcc characteristic example (At STP instruction execution, Ta = 85 °C)

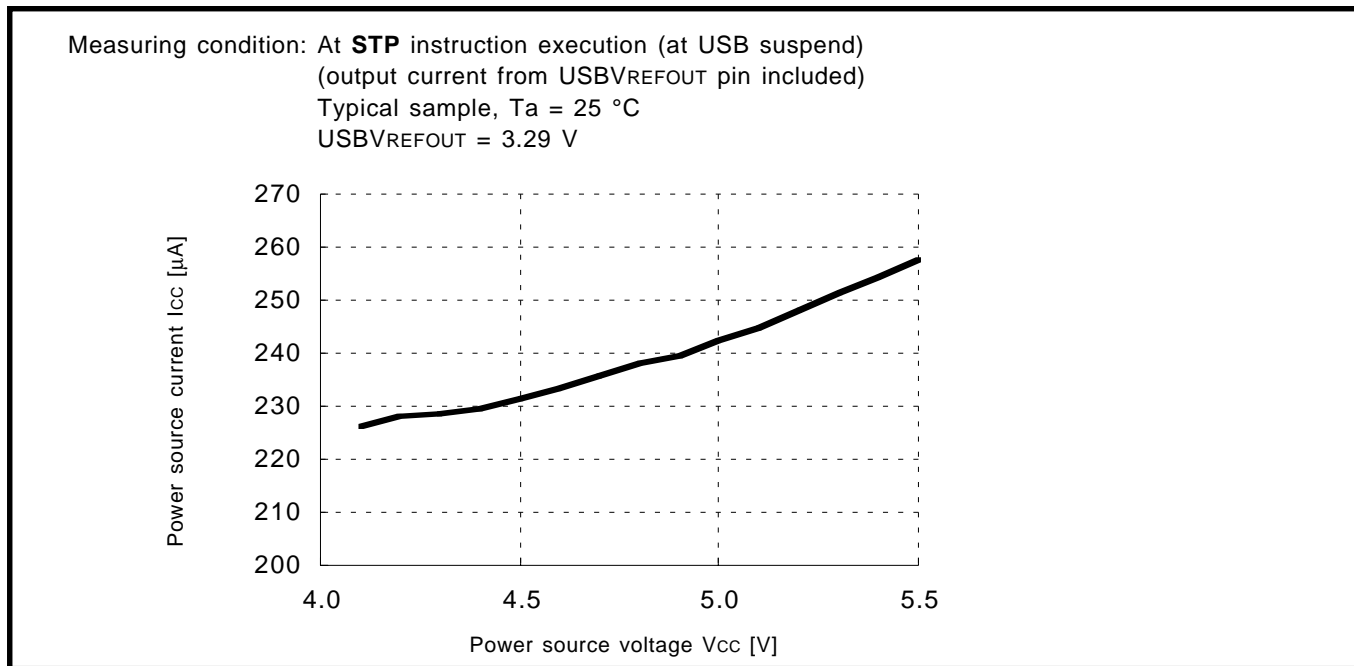


Fig. 3.2.5 Icc-Vcc characteristic example (at USB suspend, Ta = 25 °C)

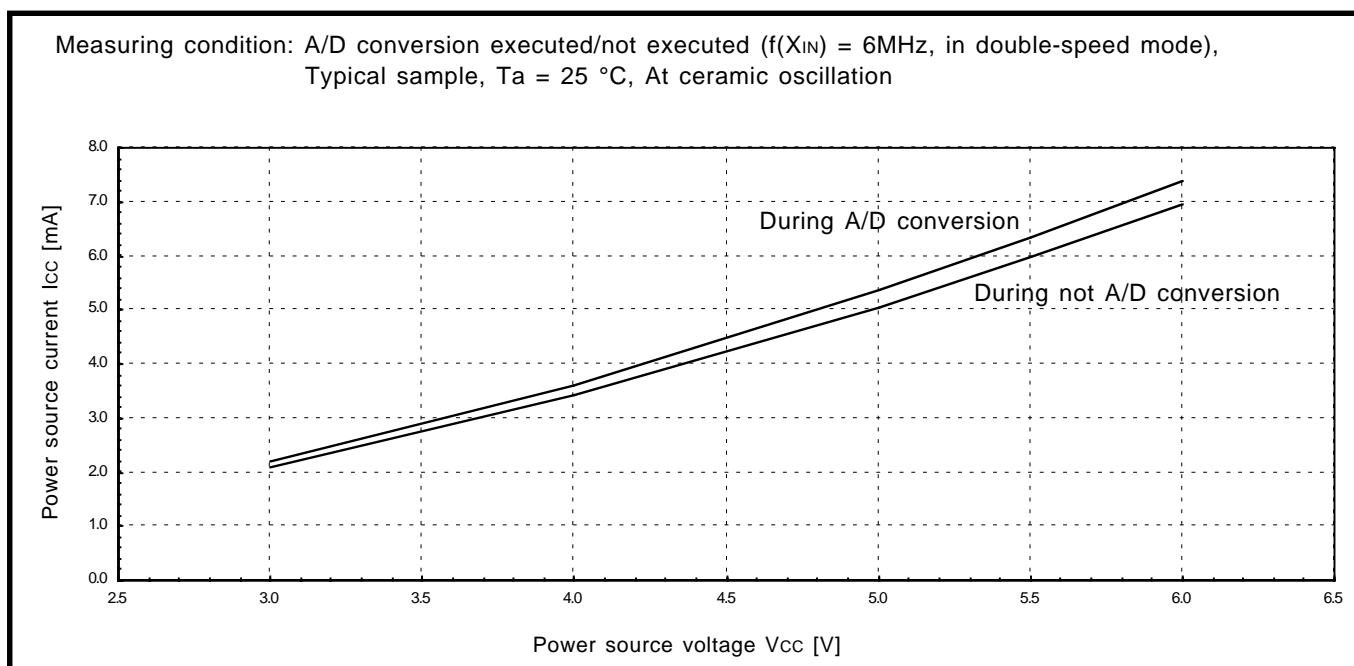


Fig. 3.2.6 Icc-Vcc characteristic example (A/D conversion executed/not executed, f(X_{IN}) = 6MHz, in double-speed mode)

3.2.2 VOH-IOH characteristic example

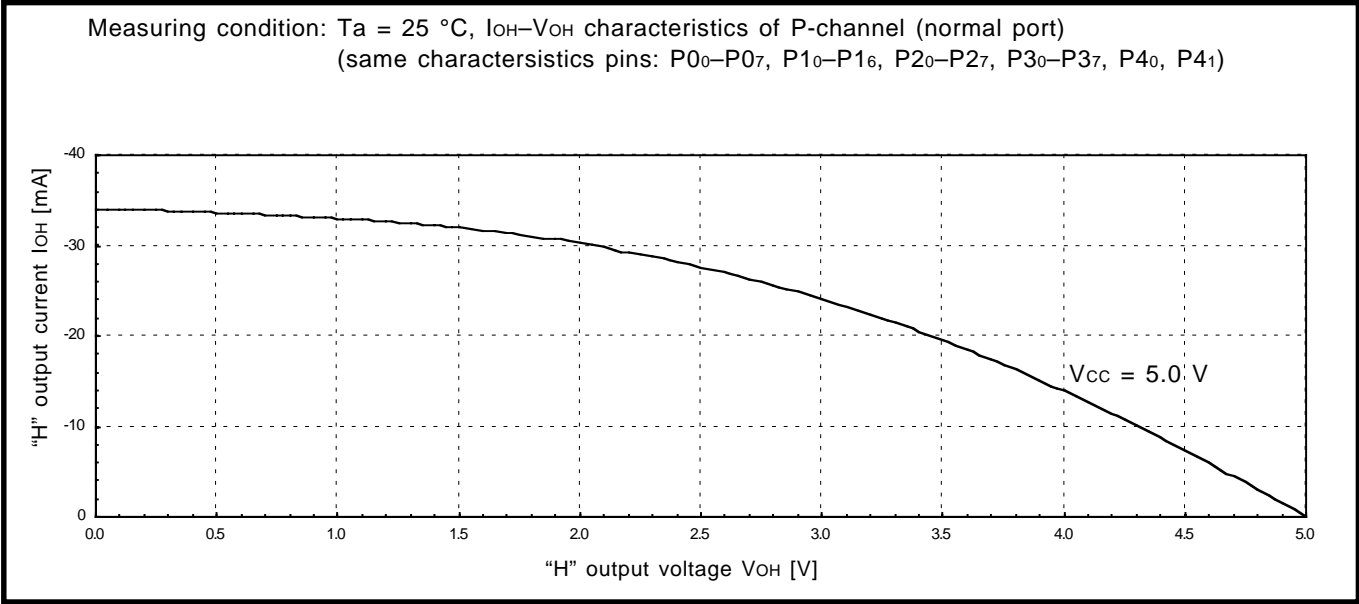


Fig. 3.2.7 VOH-IOH characteristic example of P-channel (Ta = 25 °C): normal port

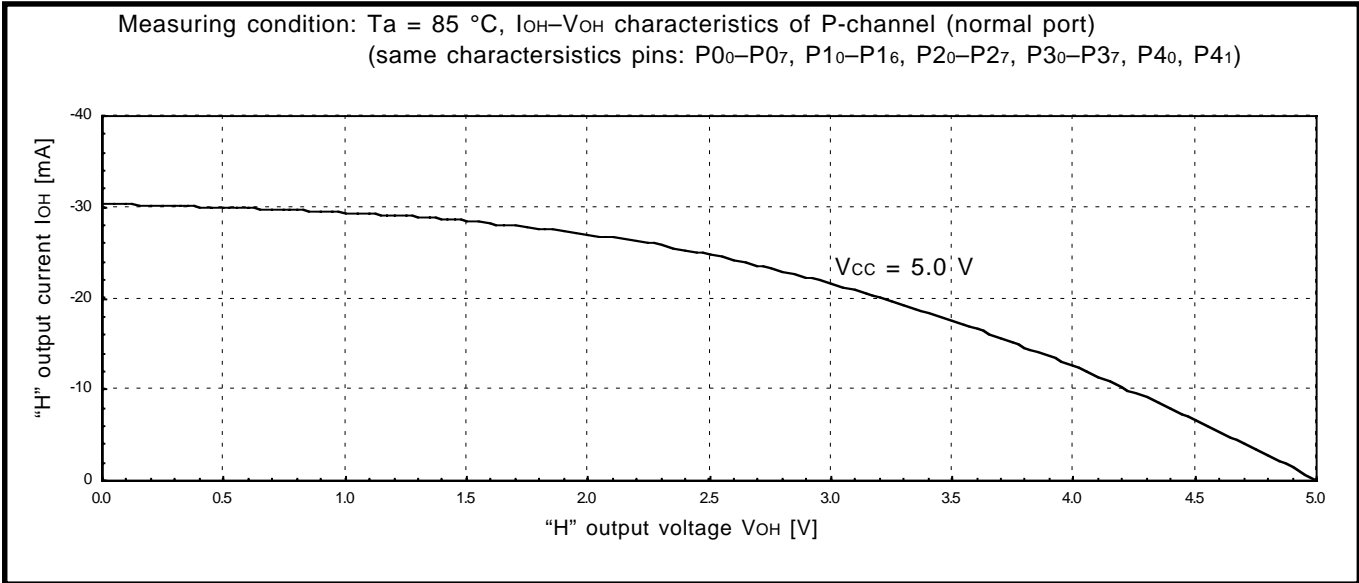


Fig. 3.2.8 VOH-IOH characteristic example of P-channel (Ta = 85 °C): normal port

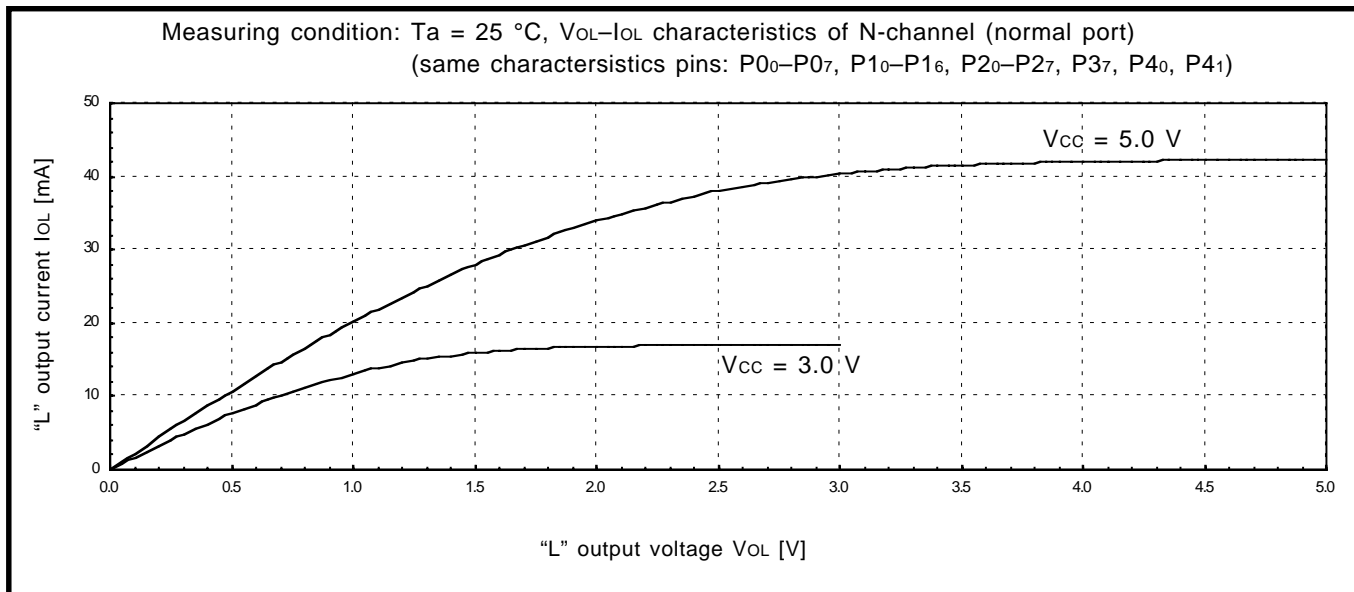


Fig. 3.2.9 V_{OL} - I_{OL} characteristic example of N-channel ($T_a = 25\text{ }^\circ\text{C}$): Normal port

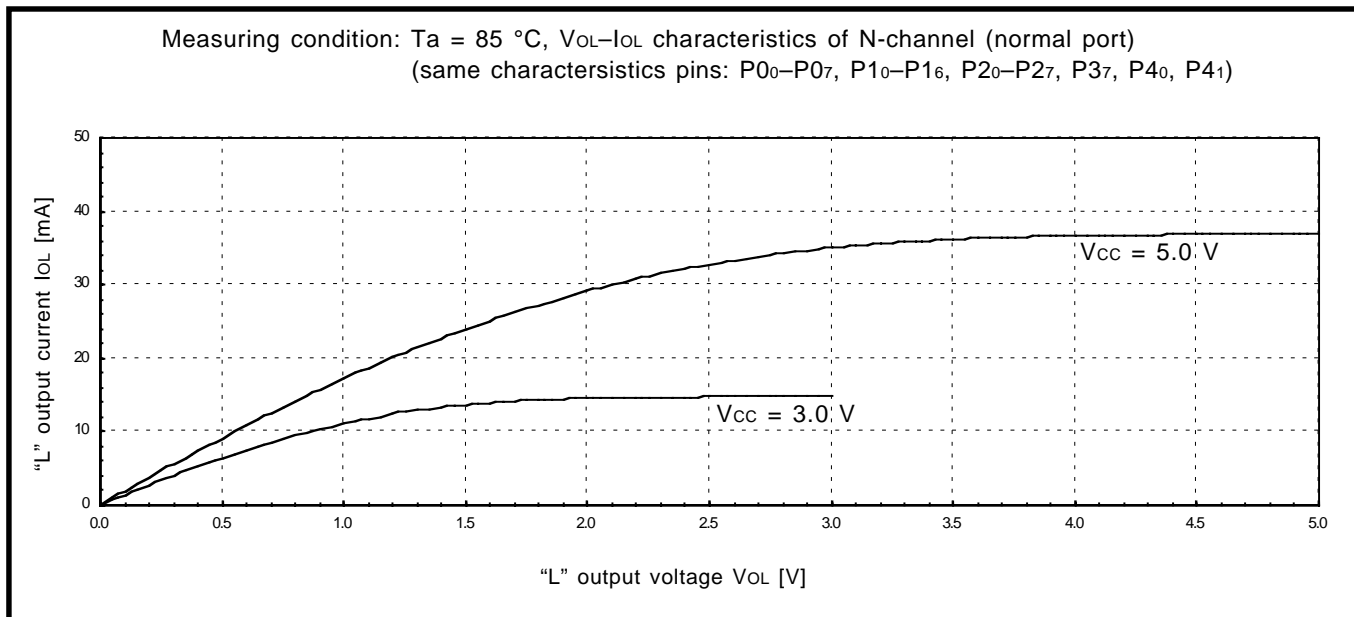


Fig. 3.2.10 V_{OL} - I_{OL} characteristic example of N-channel ($T_a = 85\text{ }^\circ\text{C}$): Normal port

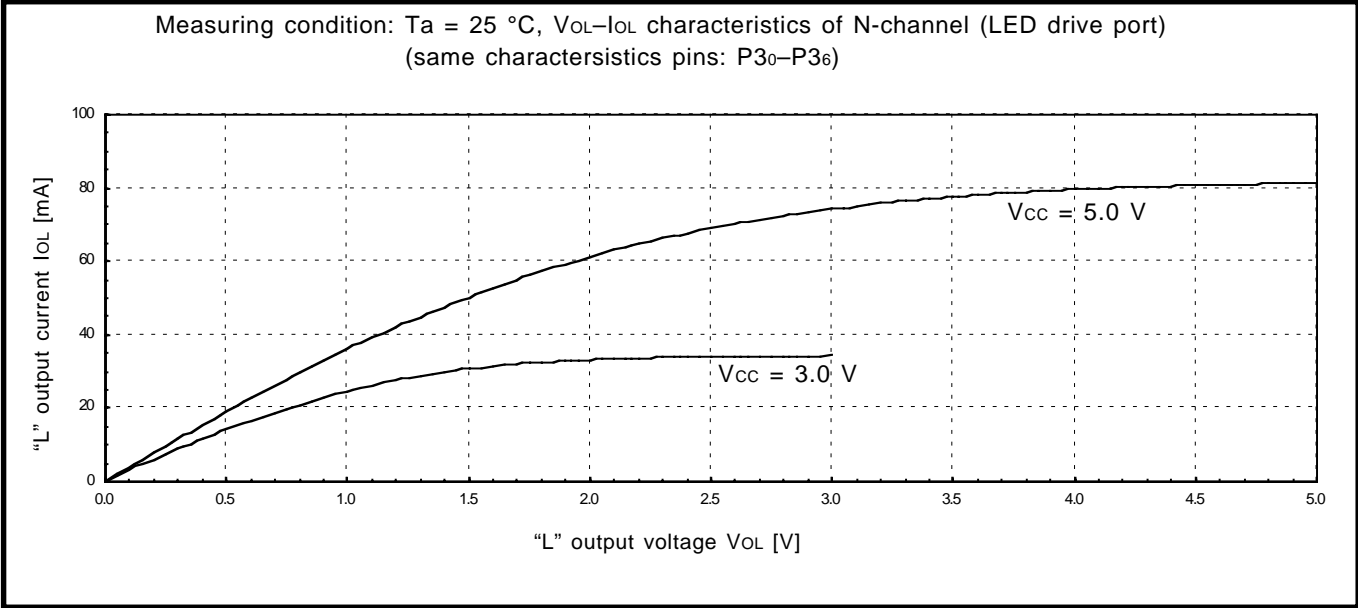


Fig. 3.2.11 V_{OL} - I_{OL} characteristic example of N-channel ($T_a = 25\text{ }^\circ\text{C}$): LED drive port

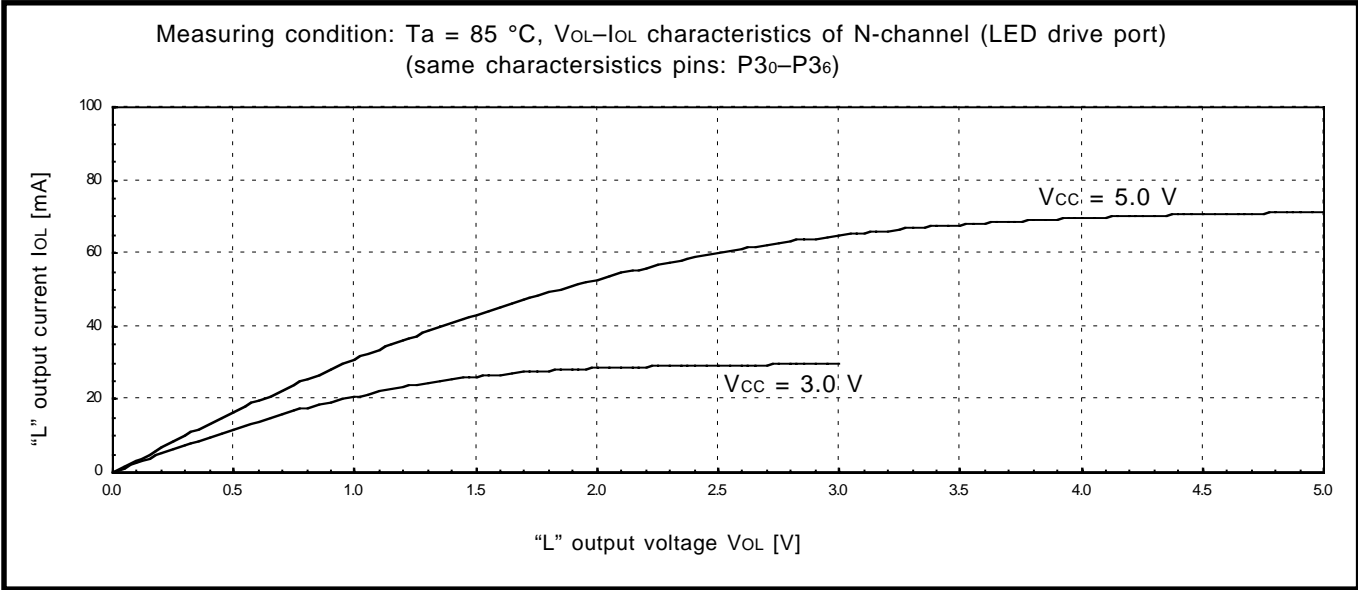


Fig. 3.2.12 V_{OL} - I_{OL} characteristic example N-channel ($T_a = 85\text{ }^\circ\text{C}$): LED drive port

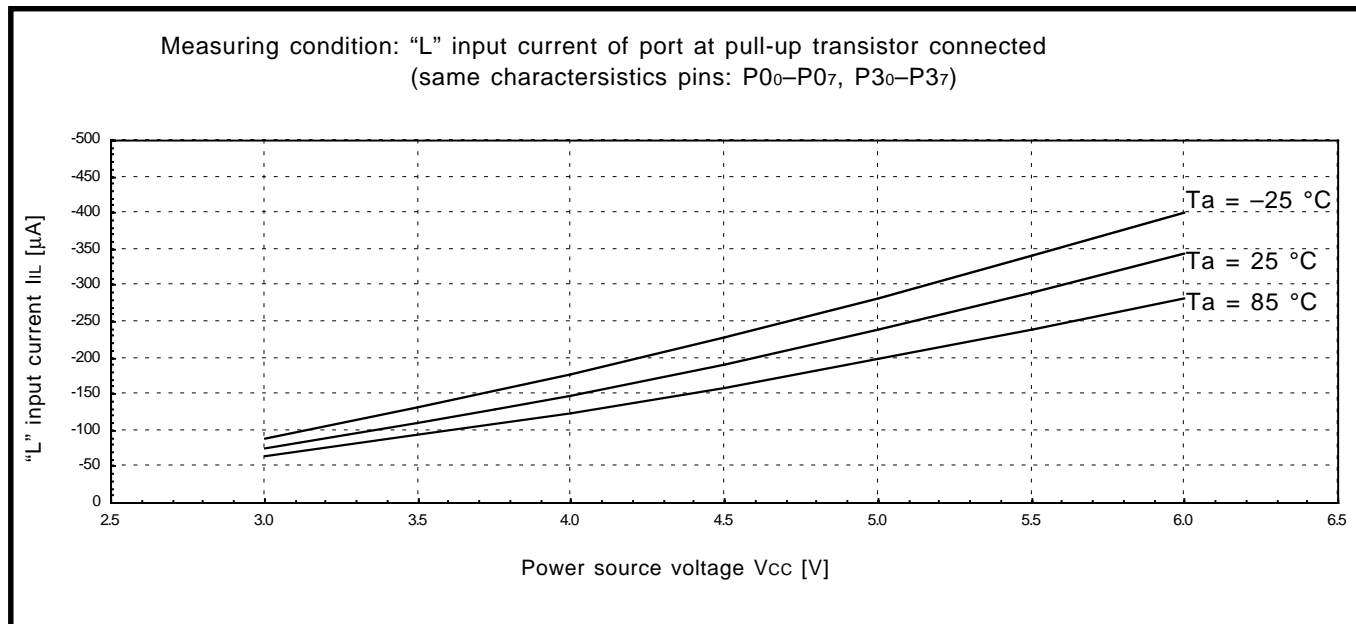


Fig. 3.2.13 "L" input current of port at pull-up transistor connected

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.

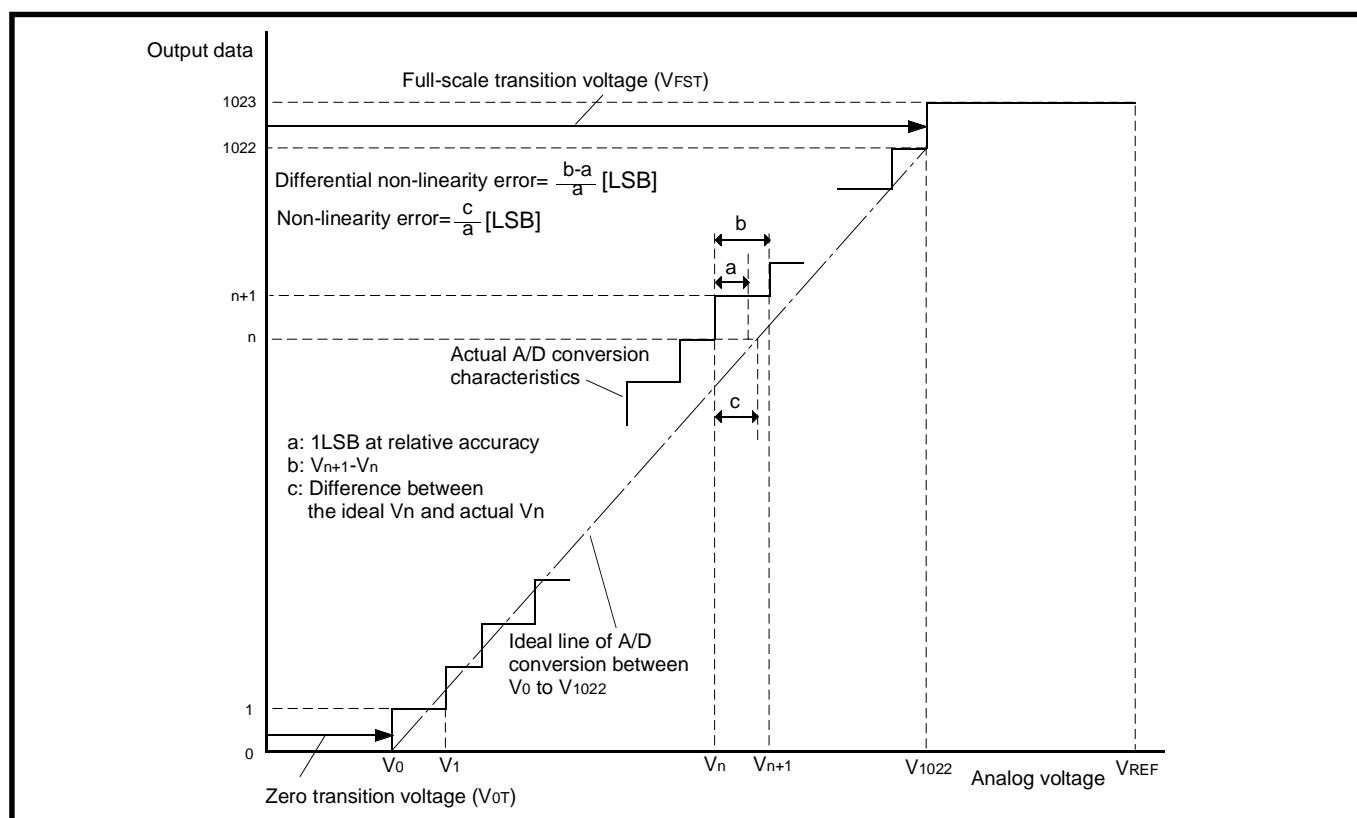


Fig. 3.2.14 Definition of A/D conversion accuracy

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)

M37534M4-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)

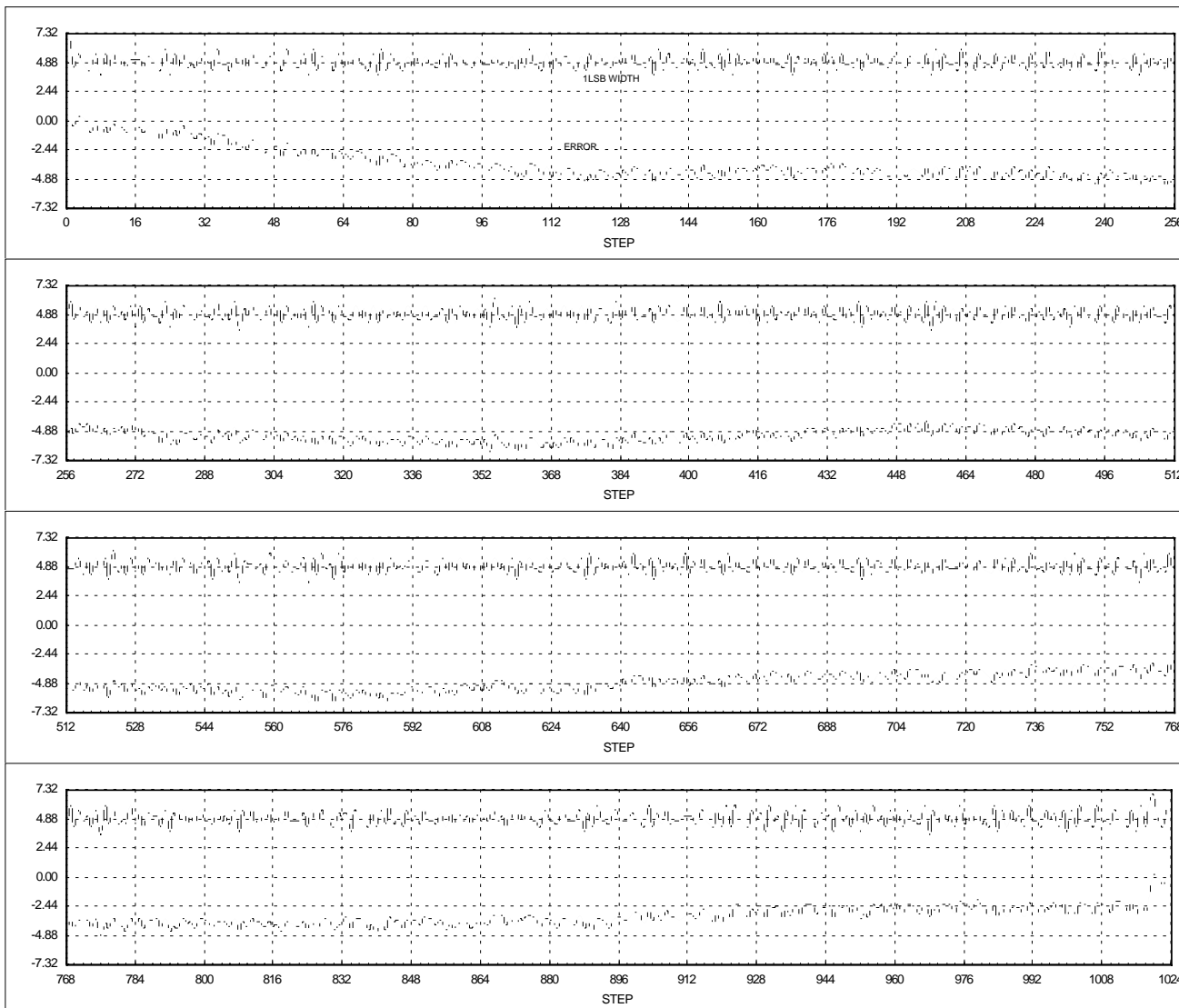


Fig. 3.2.15 A/D conversion typical characteristic example

3.3 Notes on use

3.3.1 Notes on interrupts

(1) Switching external interrupt detection edge

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

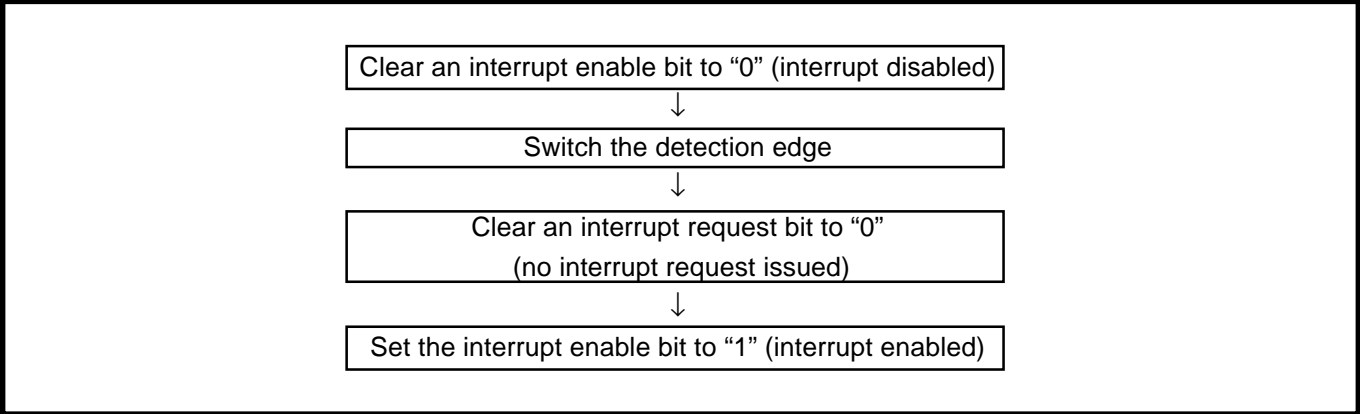


Fig. 3.3.1 Sequence of switch the detection edge

● Reason

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

(2) Check of interrupt request bit

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

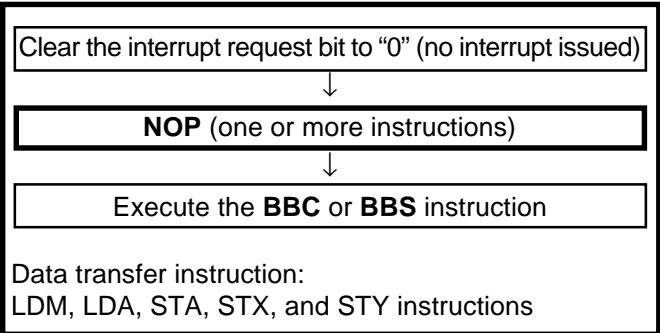


Fig. 3.3.2 Sequence of check of interrupt request bit

● Reason

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

(3) 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.

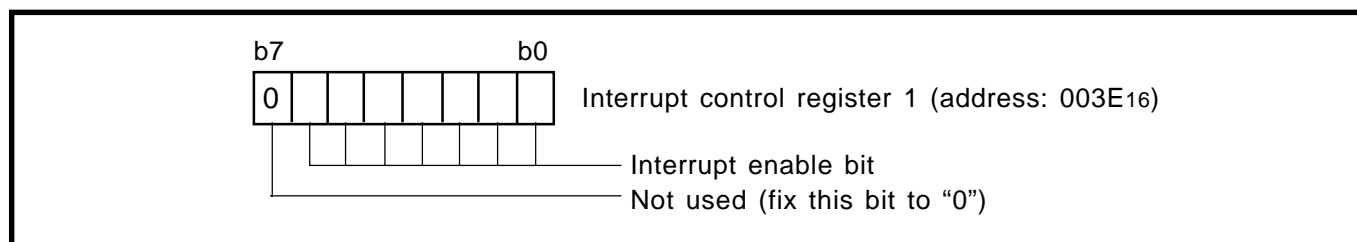


Fig. 3.3.3 Structure of interrupt control register 1

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.

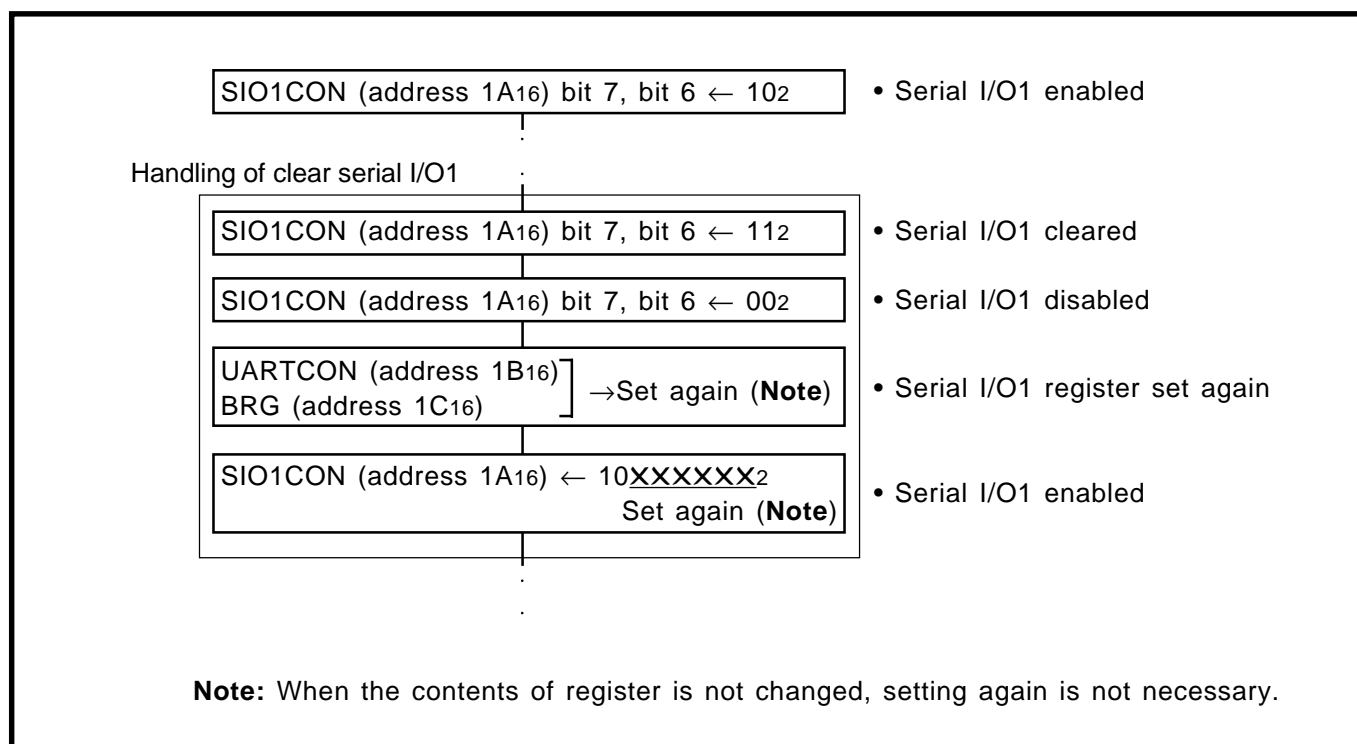


Fig. 3.3.4 Sequence of clearing serial I/O

(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 $0.01\mu\text{F}$ to $1\mu\text{F}$. 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.

- $f(\text{XIN})$ is 500 kHz or more
- Do not execute the **STP** instruction

(3) Method to stabilize A/D Converter

Method to stabilize A/D Converter is described below.

(a) A/D conversion accuracy could be affected for Bus Powered*1 USB devices, while the communicating.

Figure 3.3.5 shows the method to stabilize A/D conversion accuracy, inserting a capacitor between Vref and Vss.

*1: Power supplied by USB Vcc BUS.

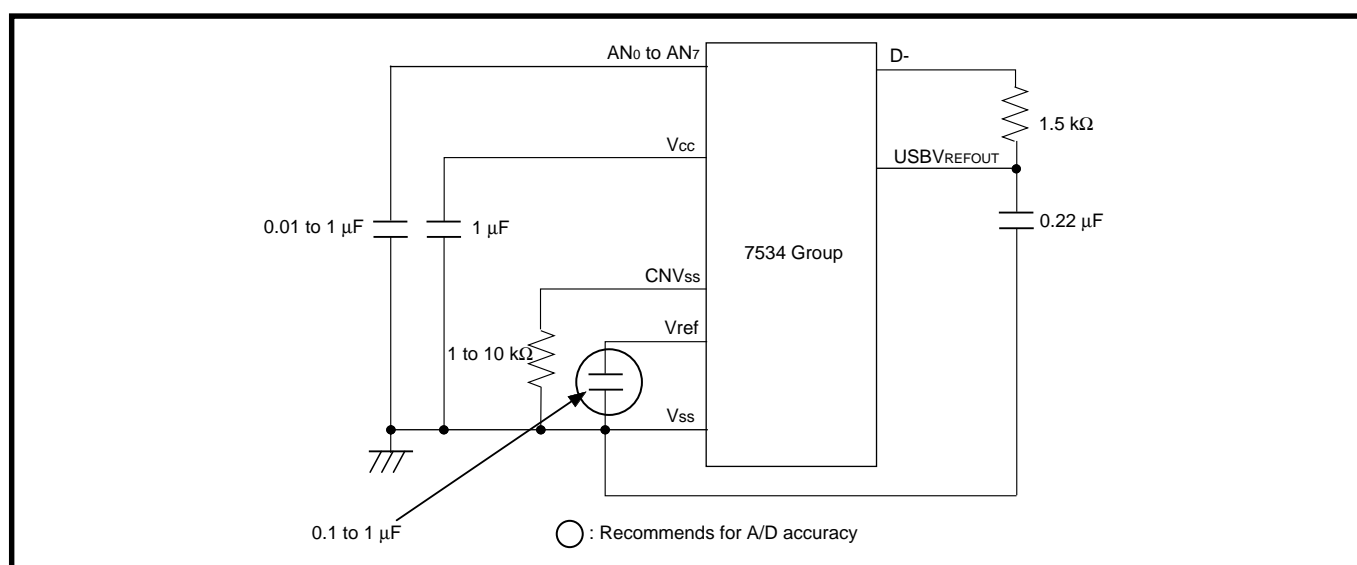


Fig. 3.3.5 Method to stabilize A/D conversion accuracy

(b) It is recommended for A/D accuracy to avoid converting while USB communication, and use average value of several converted values.

3.3.4 Notes on watchdog timer

The internal reset may not be generated correctly in the middle-speed mode, depending on the underflow timing of the watchdog timer.

When using the watchdog timer, operate the MCU in any mode other than the middle-speed mode (i.e., high-speed, low-speed or double-speed mode).

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

(1) Connecting capacitor

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

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

● Reason

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

3.3.6 Notes on input and output pins

(1) Notes in stand-by state

In stand-by state*1 for low-power dissipation, do not make input levels of an input port and an I/O port "undefined".

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.

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

(2) Modifying output data with bit managing instruction

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

● Reason

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

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

Note the following :

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

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

3.3.7 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".

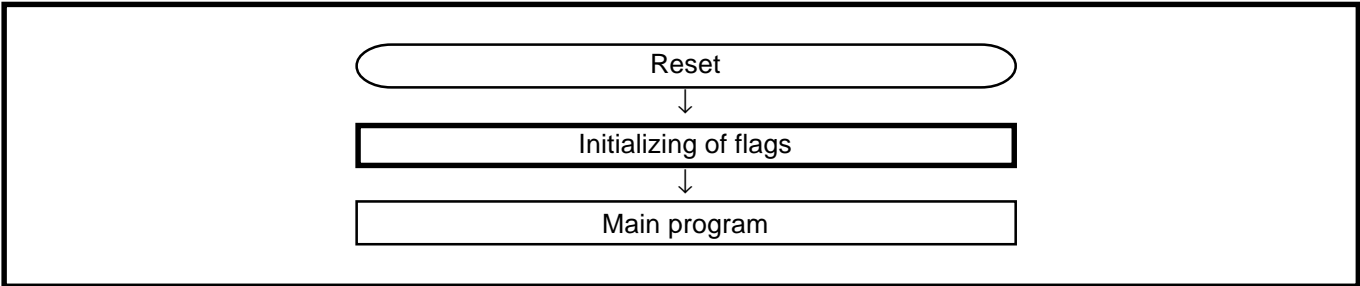


Fig. 3.3.6 Initialization of processor status register

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.

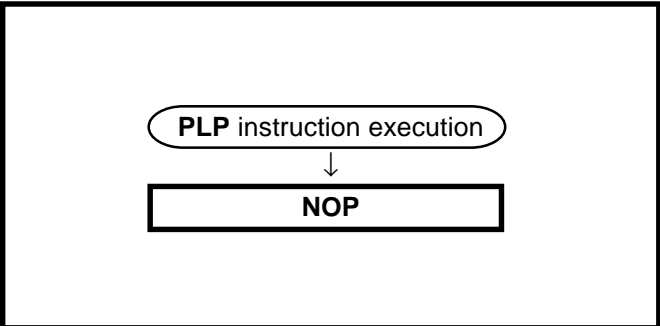


Fig. 3.3.7 Sequence of PLP instruction execution

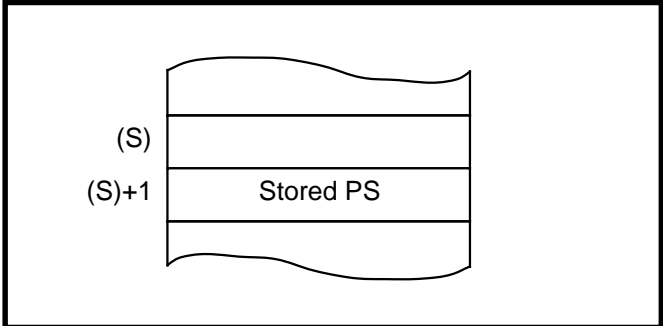


Fig. 3.3.8 Stack memory contents after PHP instruction execution

(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.

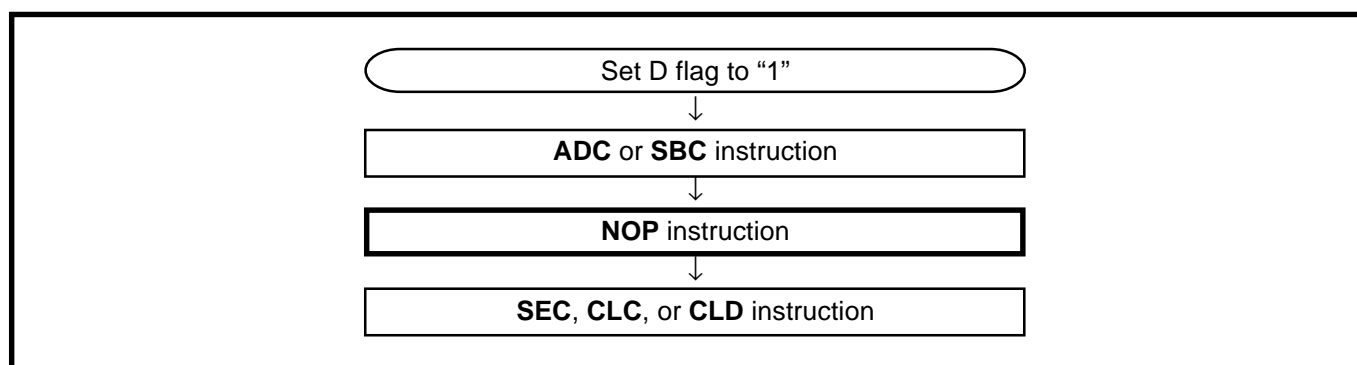


Fig. 3.3.9 Status flag at decimal calculations

(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.

3.3.8 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 3.3.10 before actual use are recommended.

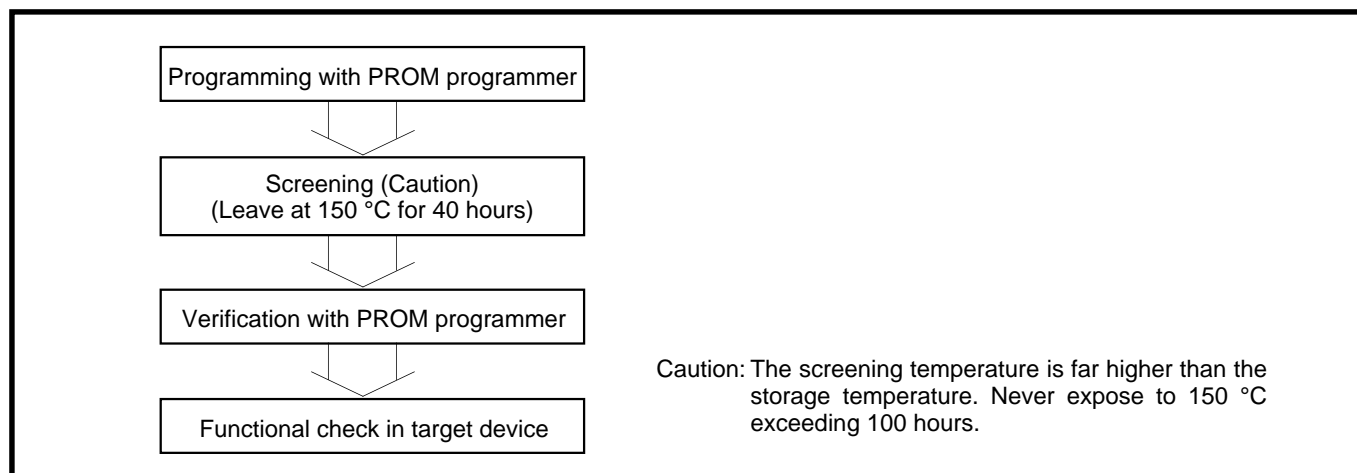


Fig. 3.3.10 Programming and testing of One Time PROM version

3.3.9 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
M37534E4GP (One Time PROM version shipped in blank)	PCA7435GPG03
M37534E8SP (One Time PROM version shipped in blank)	PCA7435SP, PCA7435SPG02
M37534E8FP (One Time PROM version shipped in blank)	PCA7435FP, PCA7435FPG02

(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
M37534E4GP	Address 0E080 ₁₆ (Note 1)	Address 0FFFD ₁₆ (Note 1)
M37534E8SP	Address 0C080 ₁₆ (Note 2)	Address 0FFFD ₁₆ (Note 2)
M37534E8FP		

Notes 1: Addersses E080₁₆ to FFFD₁₆ in the built-in PROM corresponds to addresses 0E080₁₆ to 0FFFD₁₆ in the PROM programmer.

2: Addersses C080₁₆ to FFFD₁₆ in the built-in PROM corresponds to addresses 0C080₁₆ to 0FFFD₁₆ in the PROM programmer.

3.3.10 Termination of unused pins

(1) Terminate unused pins

① Output ports : Open

② Input ports :

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

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 Ω to 10 k Ω .

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.

3.3.11 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.

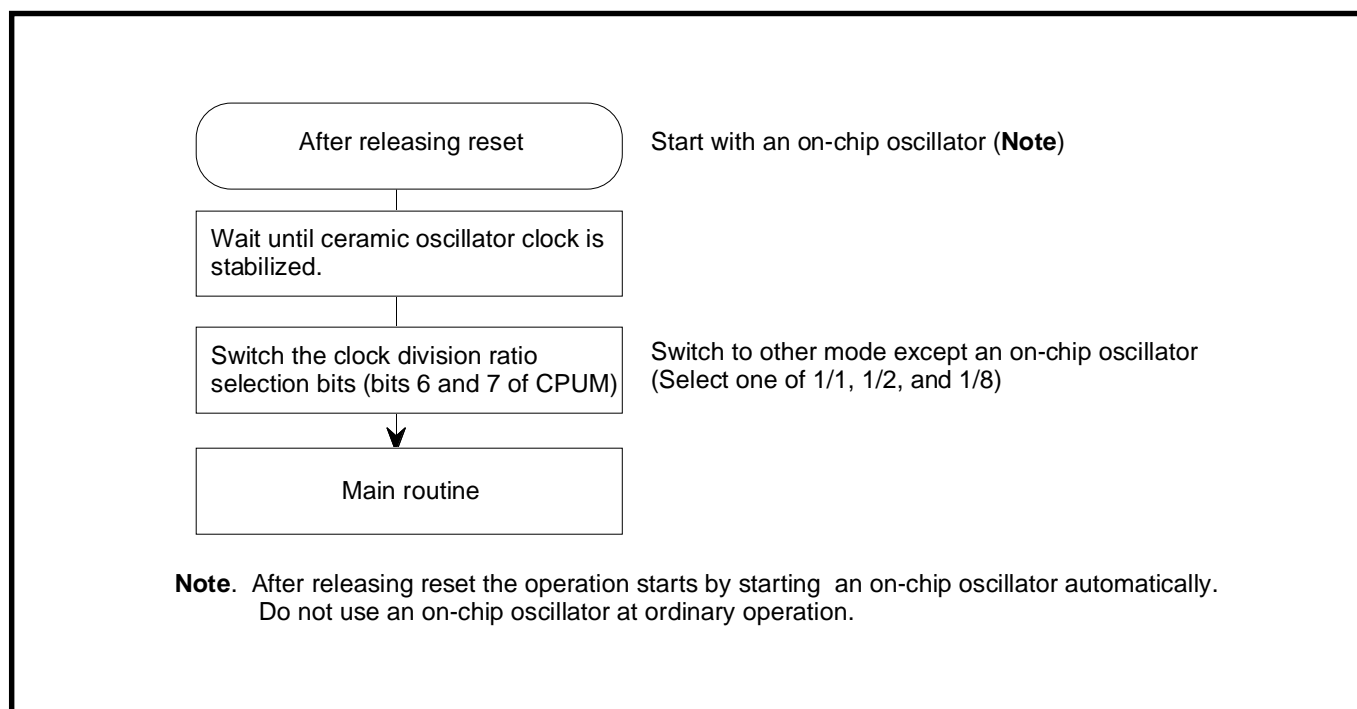


Fig. 3.3.11 Switching method of CPU mode register

3.3.12 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.

3.3.13 Electric characteristic differences among mask ROM and One Time PROM version MCUs

There are differences in electric characteristics, operation margin, noise immunity, and noise radiation among mask ROM and One Time PROM version MCUs due to the differences in the manufacturing processes.

When manufacturing an application system with One Time PROM version and then switching to use of the mask ROM version, perform sufficient evaluations for the commercial samples of the mask ROM version.

3.3.14 Note on power source voltage

When the power source voltage value of a microcomputer is less than the value which is indicated as the recommended operating conditions, the microcomputer does not operate normally and may perform unstable operation.

In a system where the power source voltage drops slowly when the power source voltage drops or the power supply is turned off, reset a microcomputer when the supply voltage is less than the recommended operating conditions and design a system not to cause errors to the system by this unstable operation.

3.3.15 USB communication

- In applications requiring high-reliability, we recommend providing the system with protective measures such as USB function initialization by software or USB reset by the host to prevent USB communication from being terminated unexpectedly, for example due to external causes such as noise.
- When USB suspend mode with TTL level on P10, P12, P13 input level selection bit (bit 3 of address 1716) set to "1", suspend current as I_{CC} might be greater than 300 μA as a spec.

[Countermeasure]

There are two countermeasures by software to avoid it as follows.

- (1) Change from TTL input level to CMOS input level for P10, P12, P13 port input.
- (2) Change from TTL input level to CMOS input level before STP instruction in suspend routine; then after RESUME or Remote wake up interrupt, return to TTL input level from CMOS input level. That is shown in Figure 3.3.12.

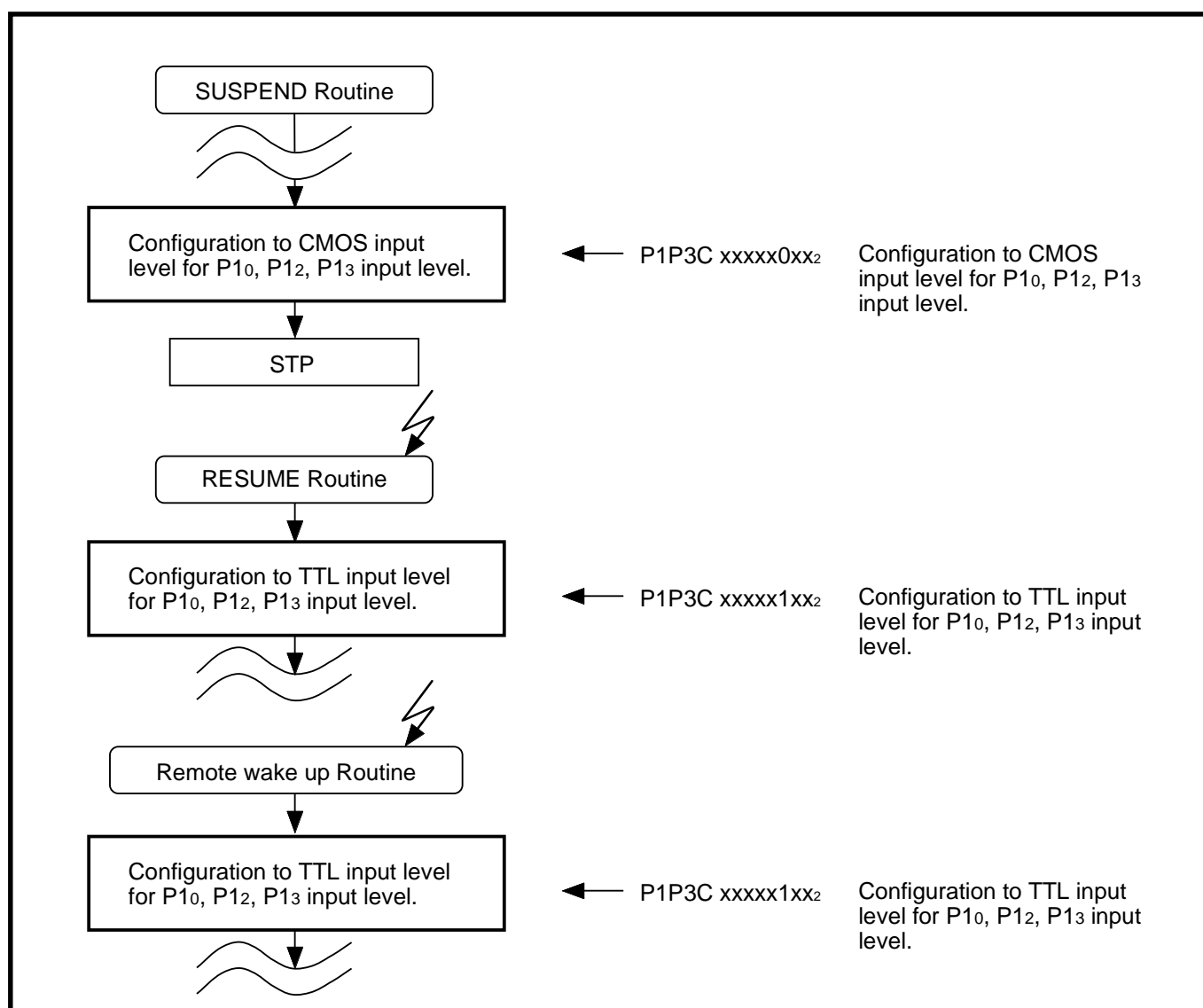


Fig. 3.3.12 Countermeasure (2) by software

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.

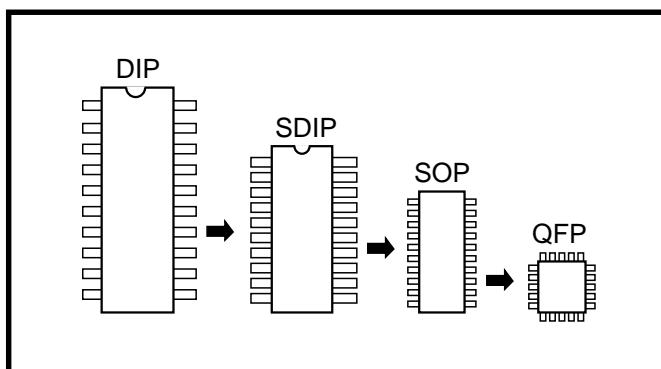


Fig. 3.4.1 Selection of packages

(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.

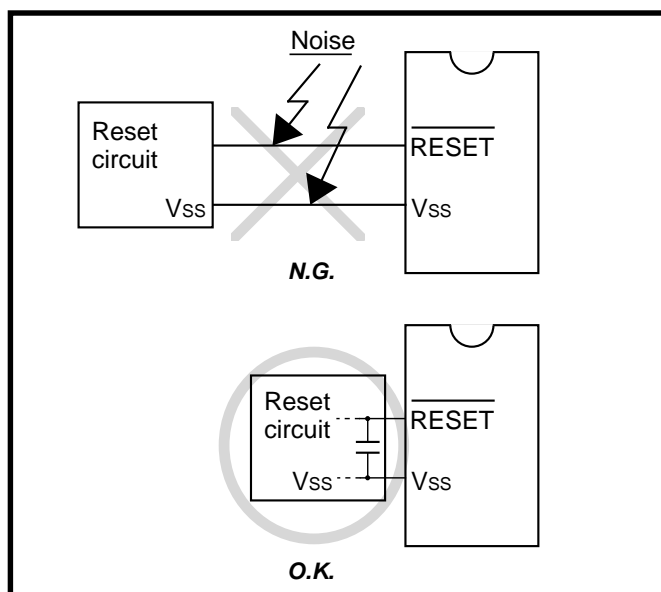


Fig. 3.4.2 Wiring for the $\overline{\text{RESET}}$ pin

(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.

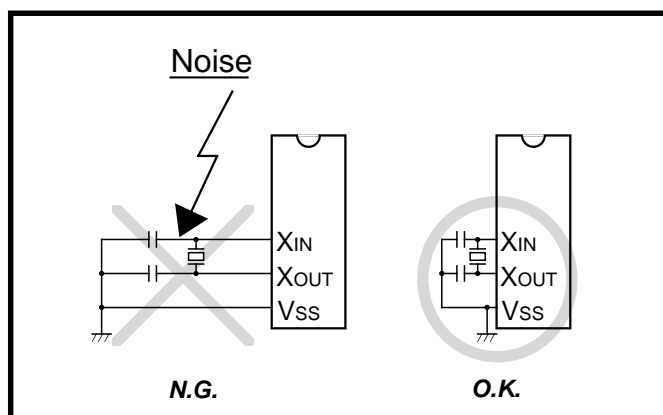


Fig. 3.4.3 Wiring for clock I/O pins

(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.

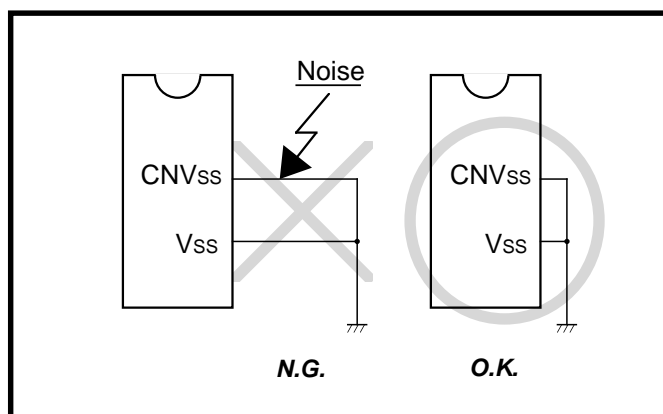


Fig. 3.4.4 Wiring for CNVss pin

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

Connect an approximately 5 k Ω 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 k Ω 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.

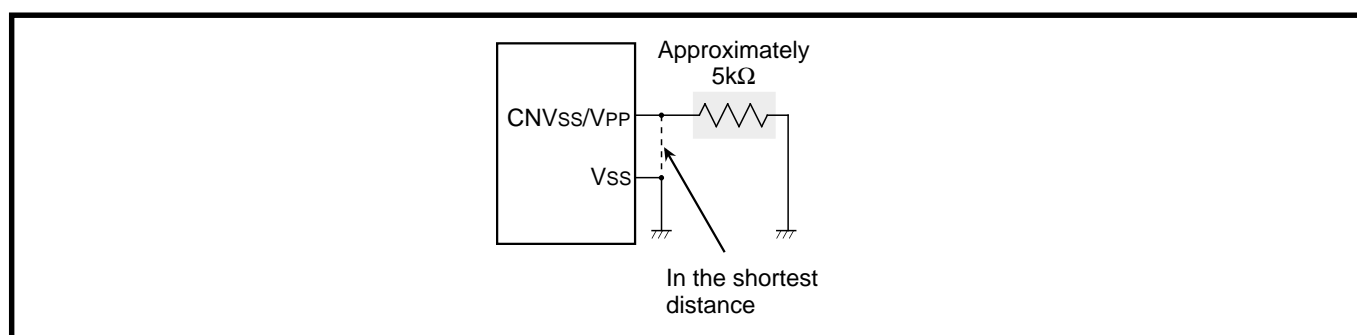


Fig. 3.4.5 Wiring for the VPP pin of the One Time PROM

3.4.2 Connection of bypass capacitor across Vss line and Vcc line

Connect an approximately 1.0 μ F bypass capacitor across the VSS line and the VCC line as follows:

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

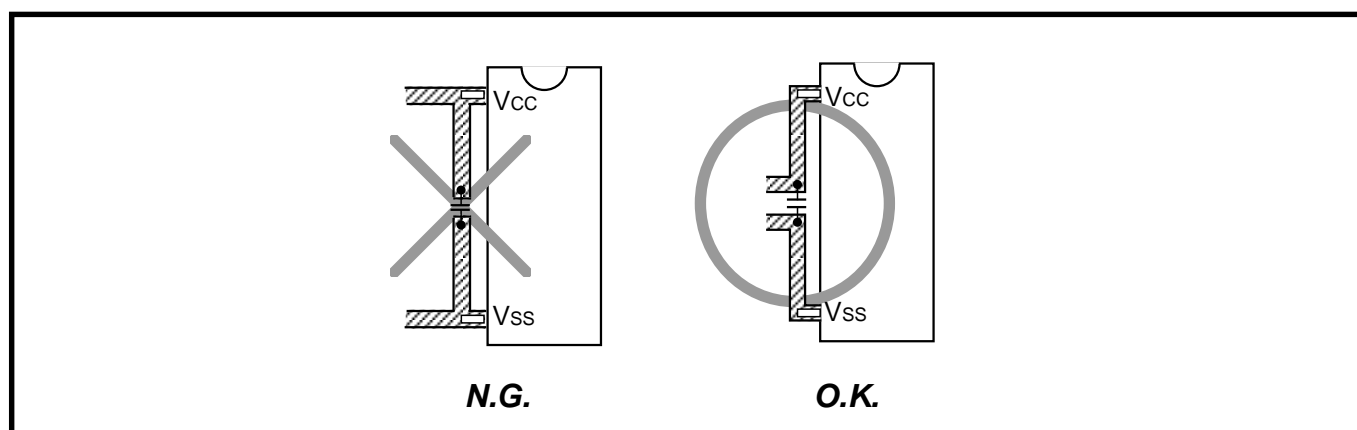


Fig. 3.4.6 Bypass capacitor across the Vss line and the Vcc line

3.4.3 Wiring to analog input pins

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

● Reason

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

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

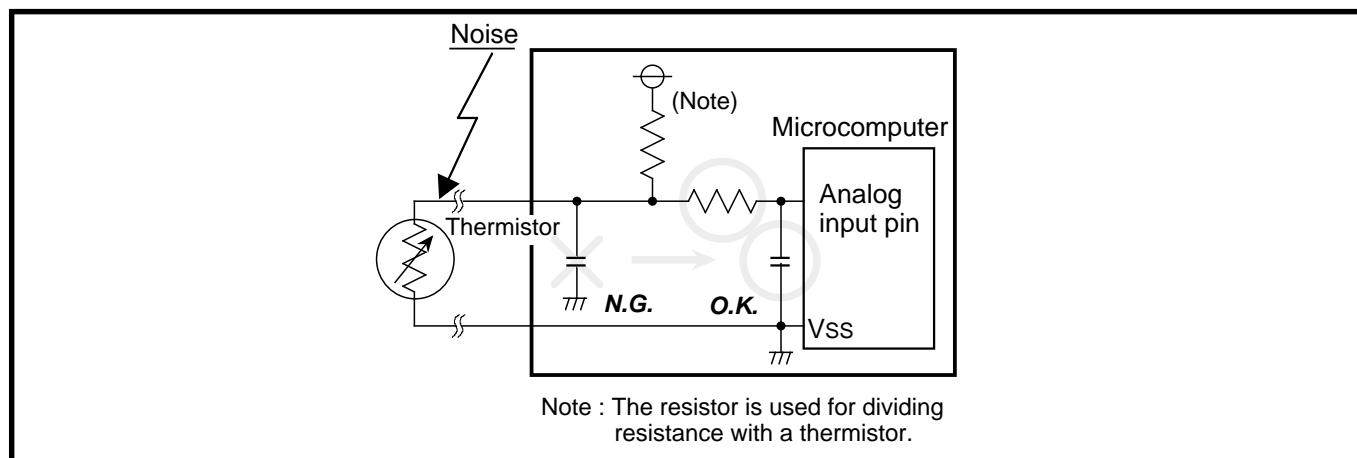


Fig. 3.4.7 Analog signal line and a resistor and a capacitor

3.4.4 Oscillator concerns

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

(1) Keeping oscillator away from large current signal lines

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

● Reason

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

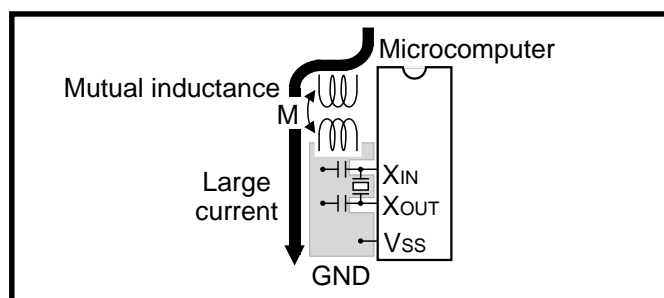


Fig. 3.4.8 Wiring for a large current signal line

(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.

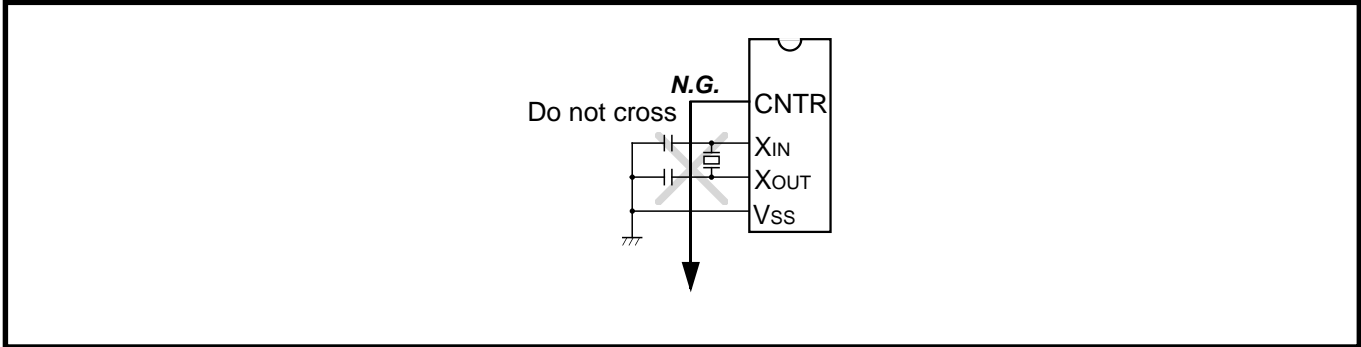


Fig. 3.4.9 Wiring of signal lines where potential levels change frequently

(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.

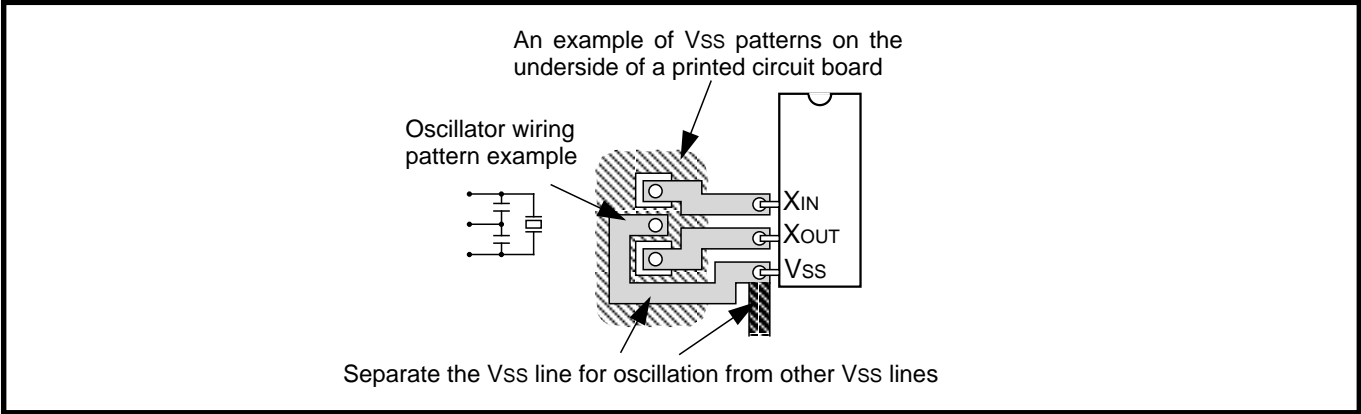


Fig. 3.4.10 Vss pattern on the underside of an oscillator

3.4.5 Setup for I/O ports

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

<Hardware>

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

<Software>

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

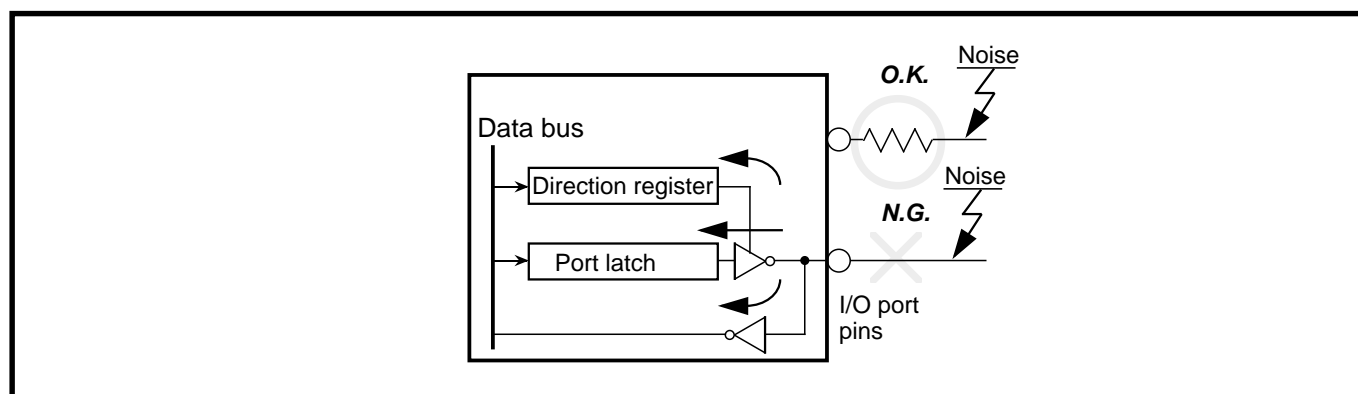


Fig. 3.4.11 Setup for I/O ports

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.

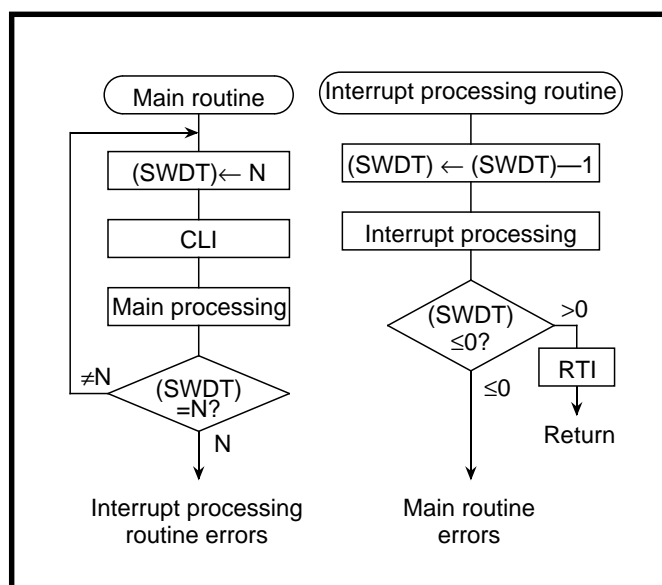


Fig. 3.4.12 Watchdog timer by software

3.5 List of registers

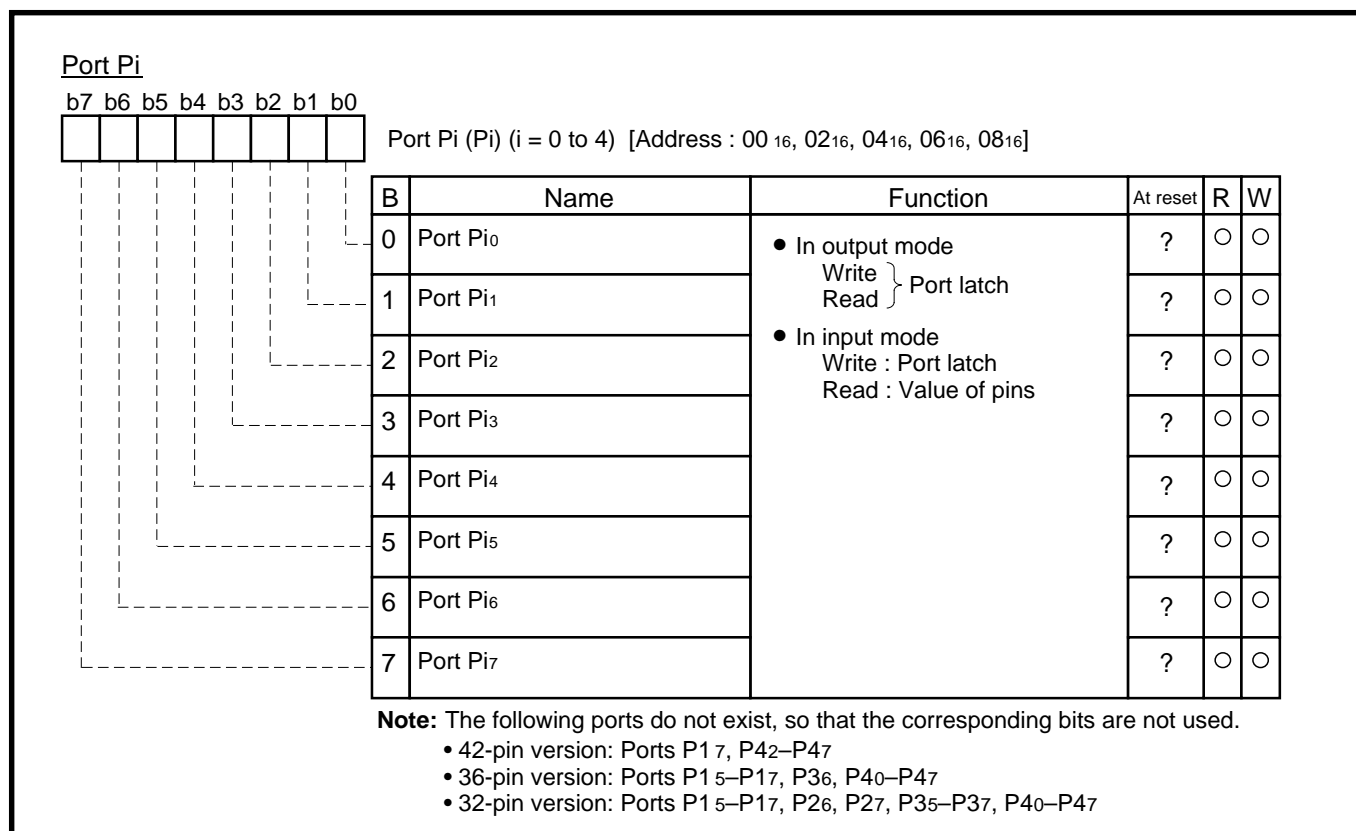


Fig. 3.5.1 Structure of Port Pi (i = 0 to 4)

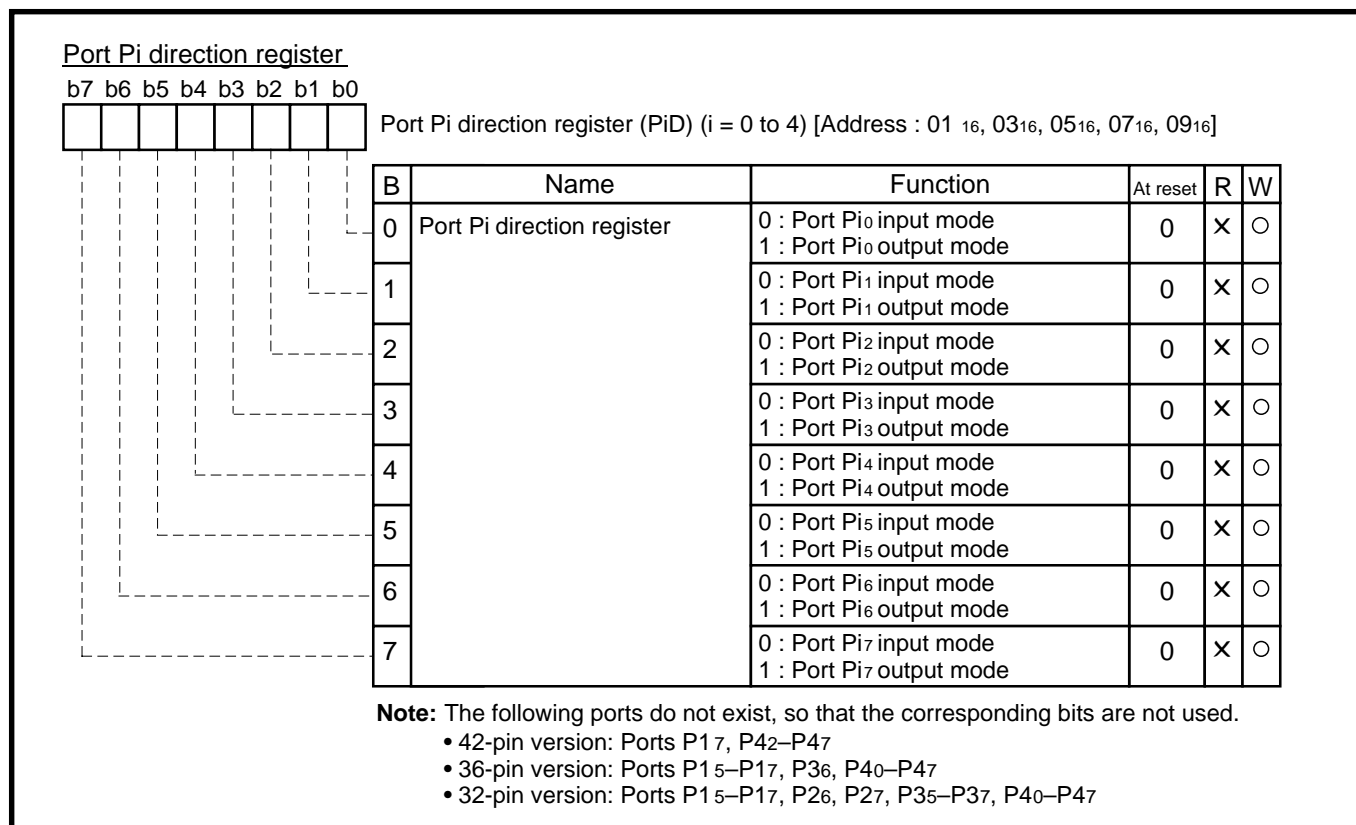


Fig. 3.5.2 Structure of Port Pi direction register (i = 0 to 4)

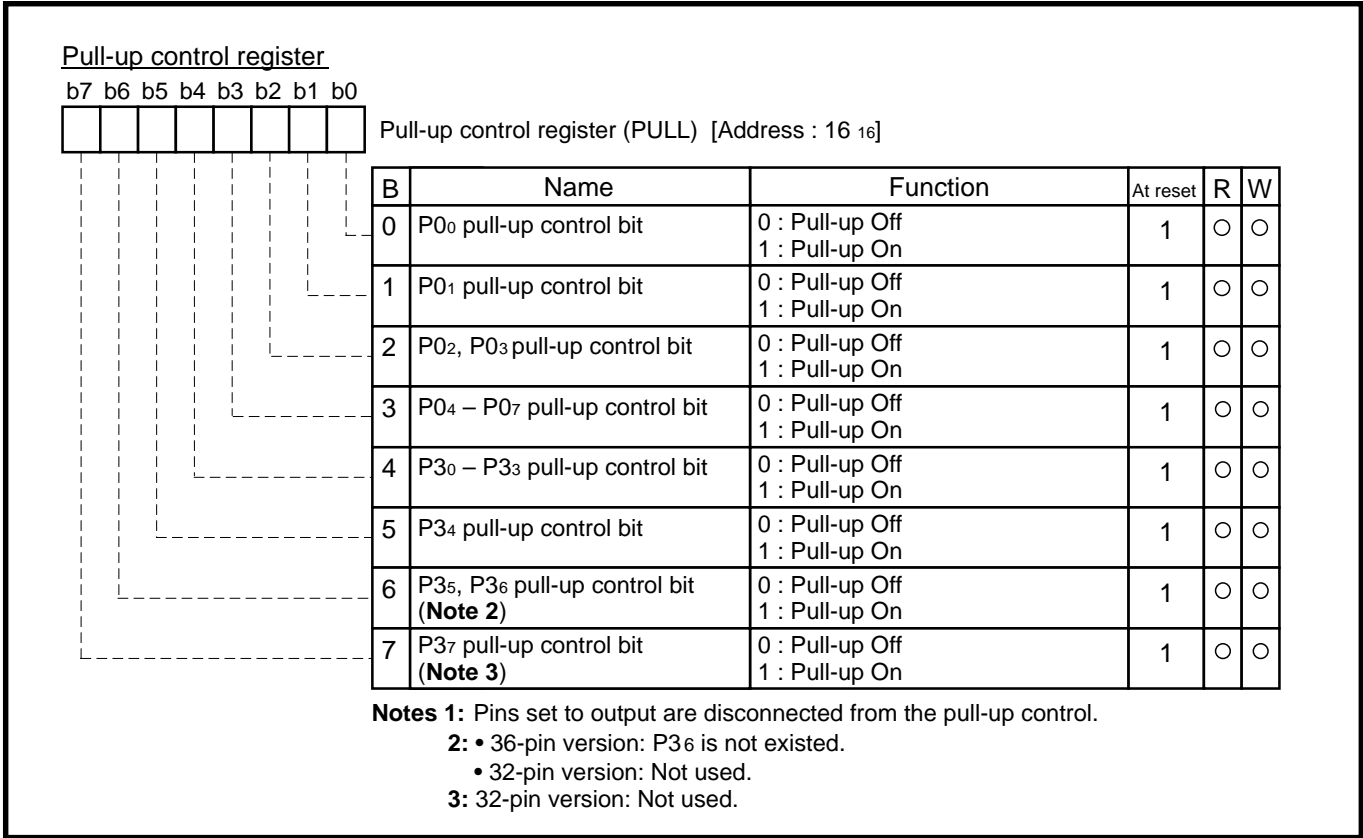


Fig. 3.5.3 Structure of Pull-up control register

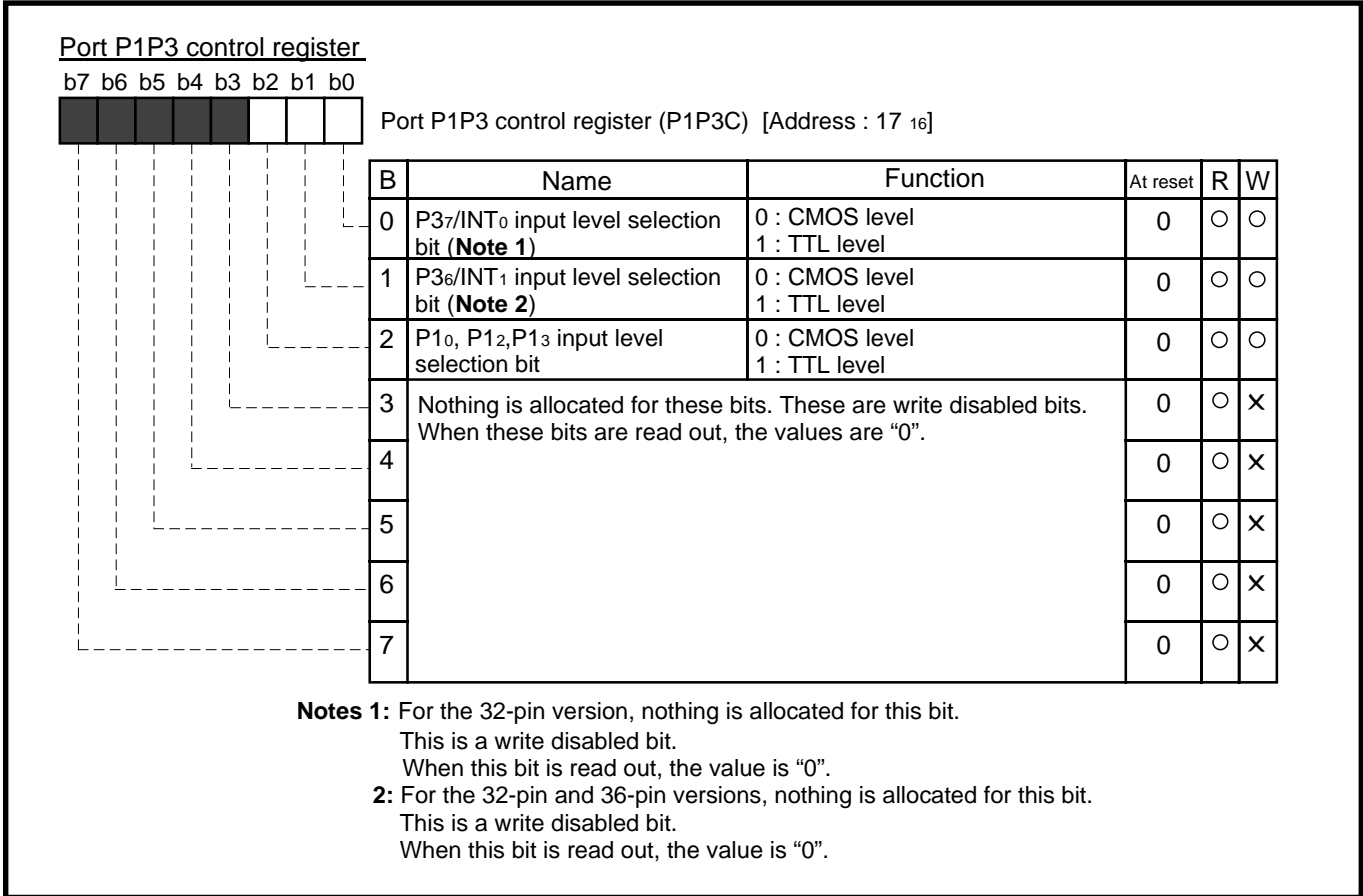


Fig. 3.5.4 Structure of Port P1P3 control register

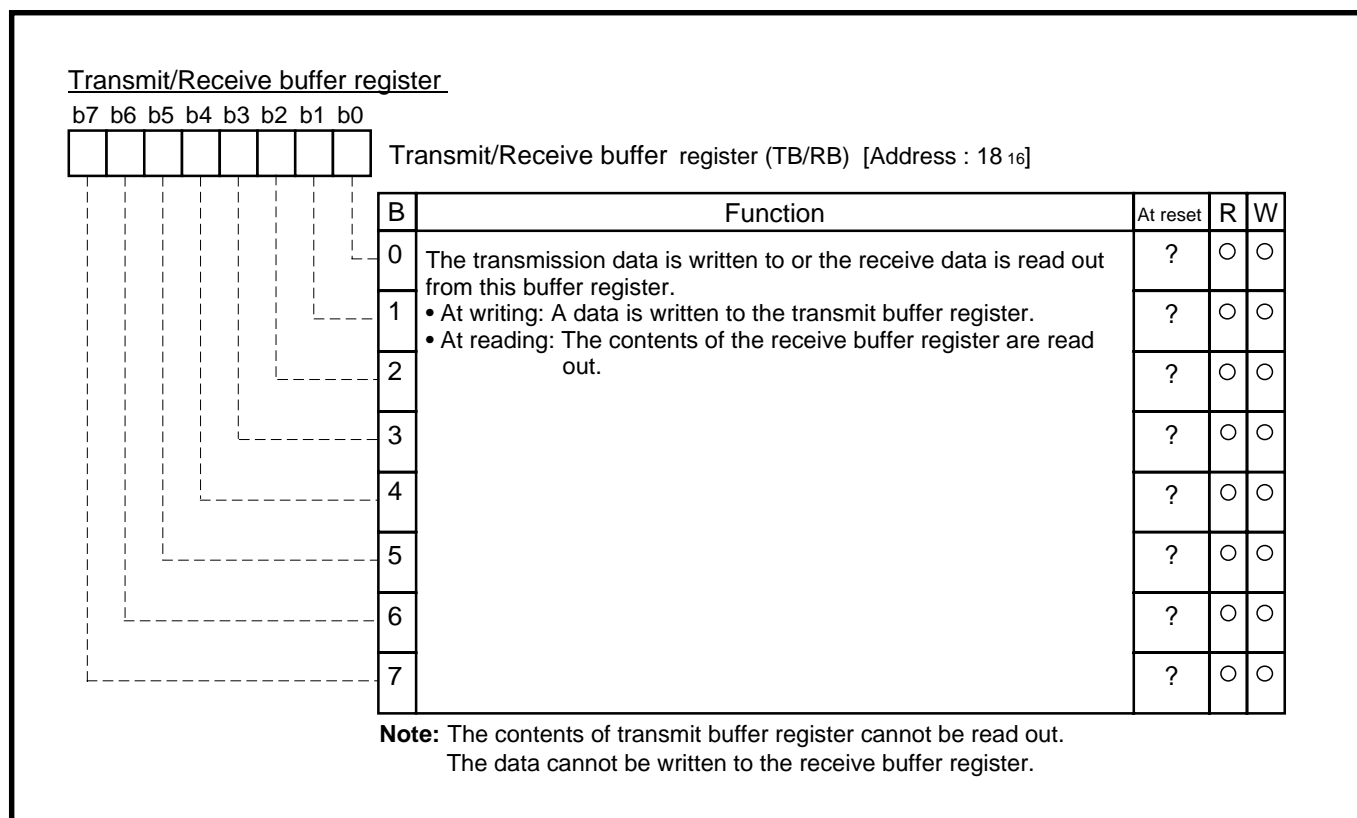


Fig. 3.5.5 Structure of Transmit/Receive buffer register

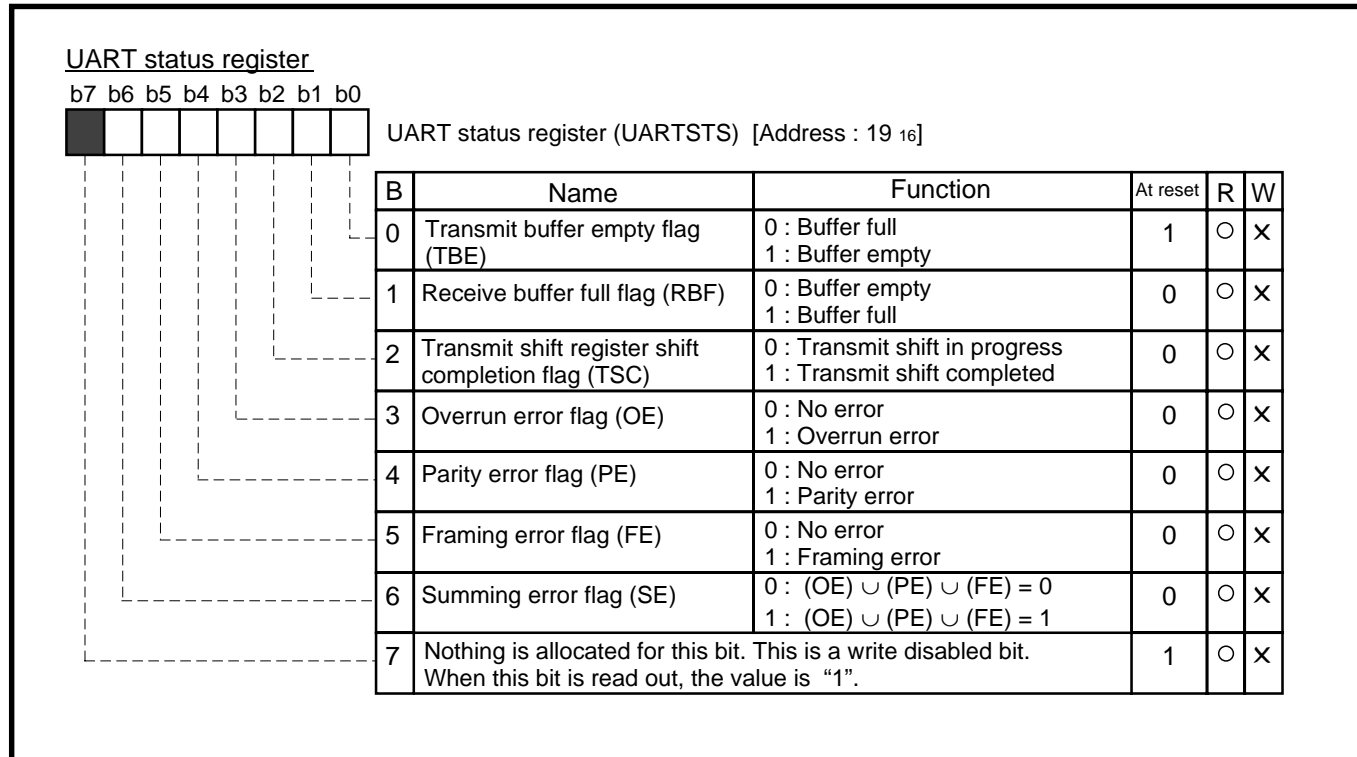


Fig. 3.5.6 Structure of UART status register

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

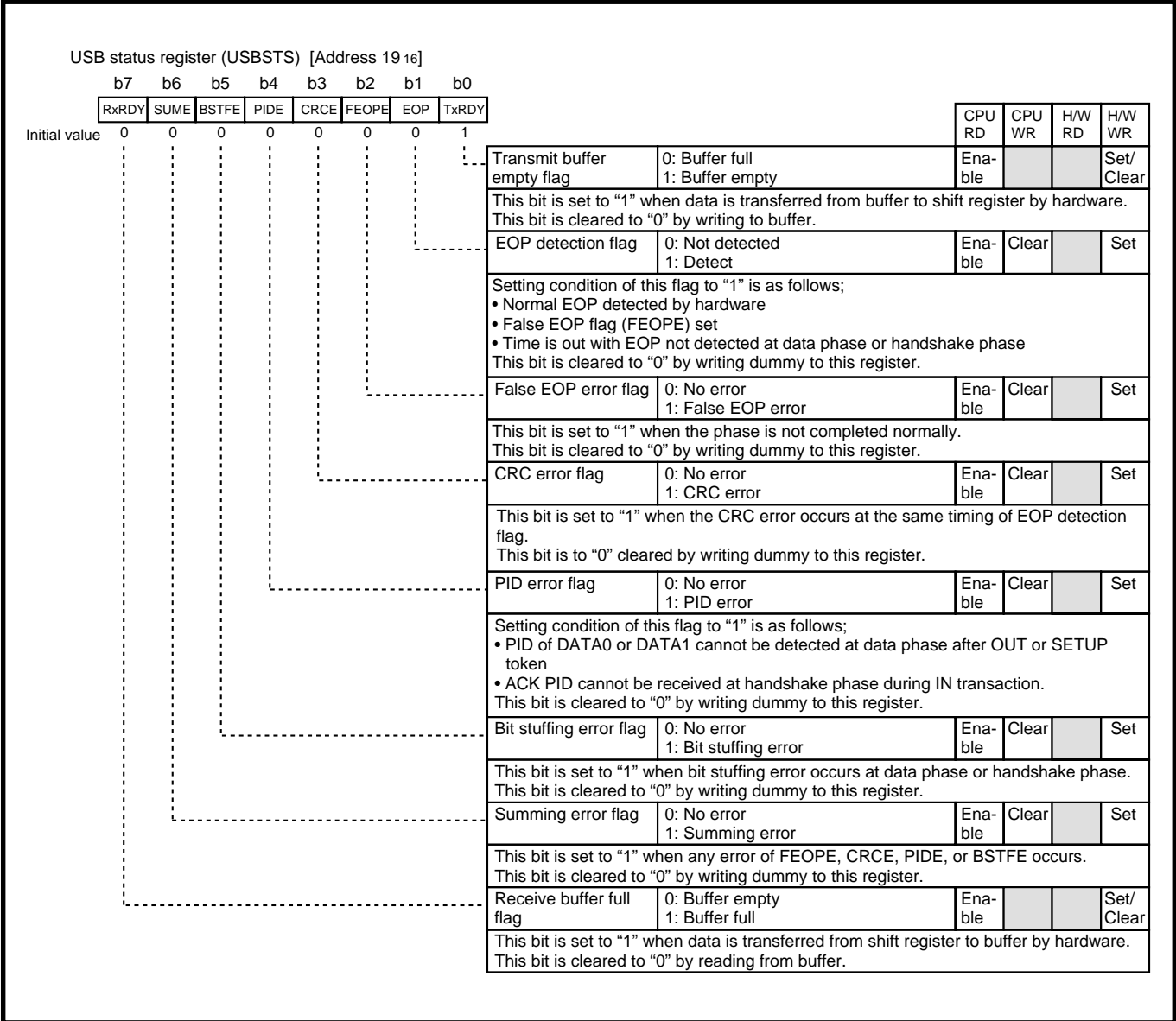


Fig. 3.5.7 Structure of USB status register

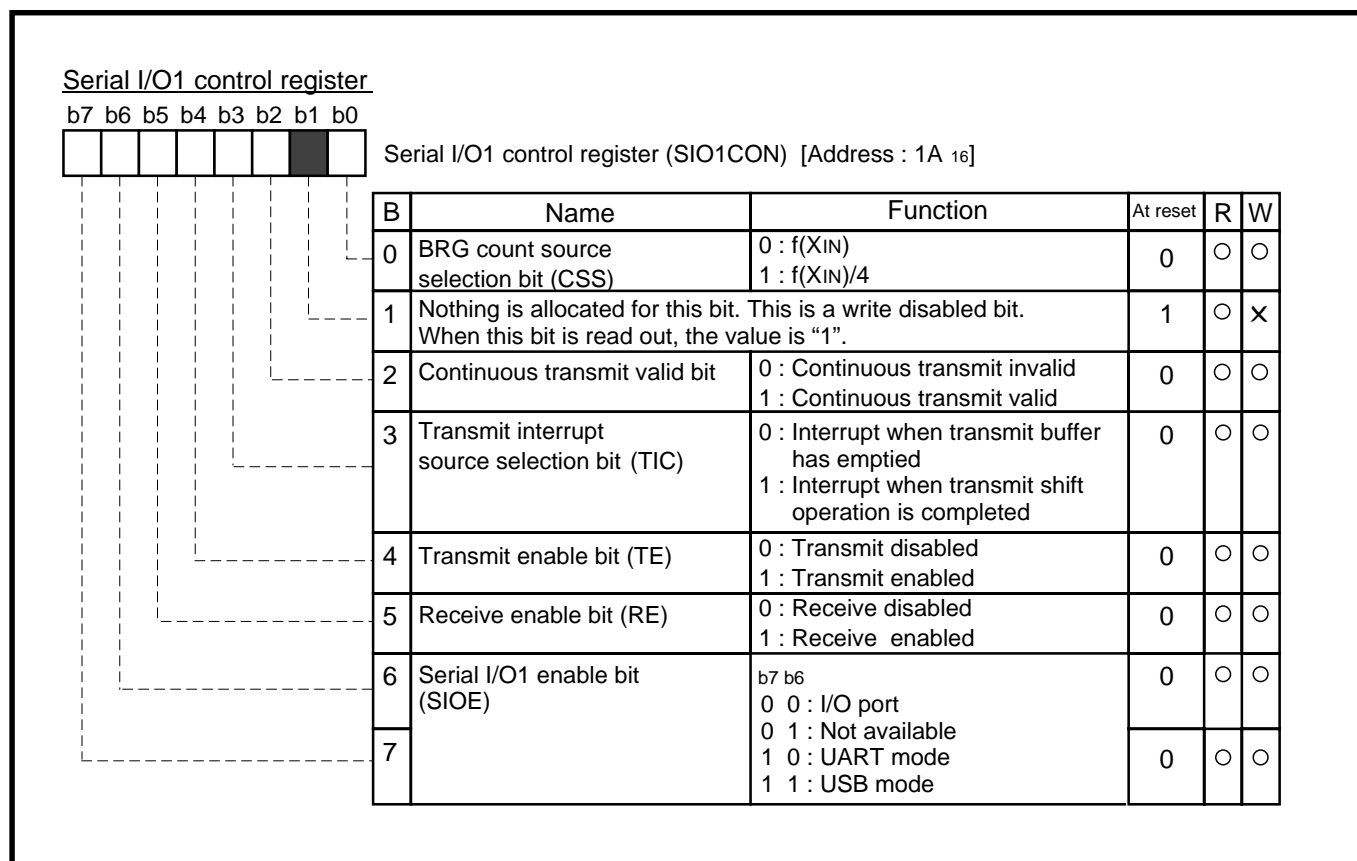


Fig. 3.5.8 Structure of Serial I/O1 control register

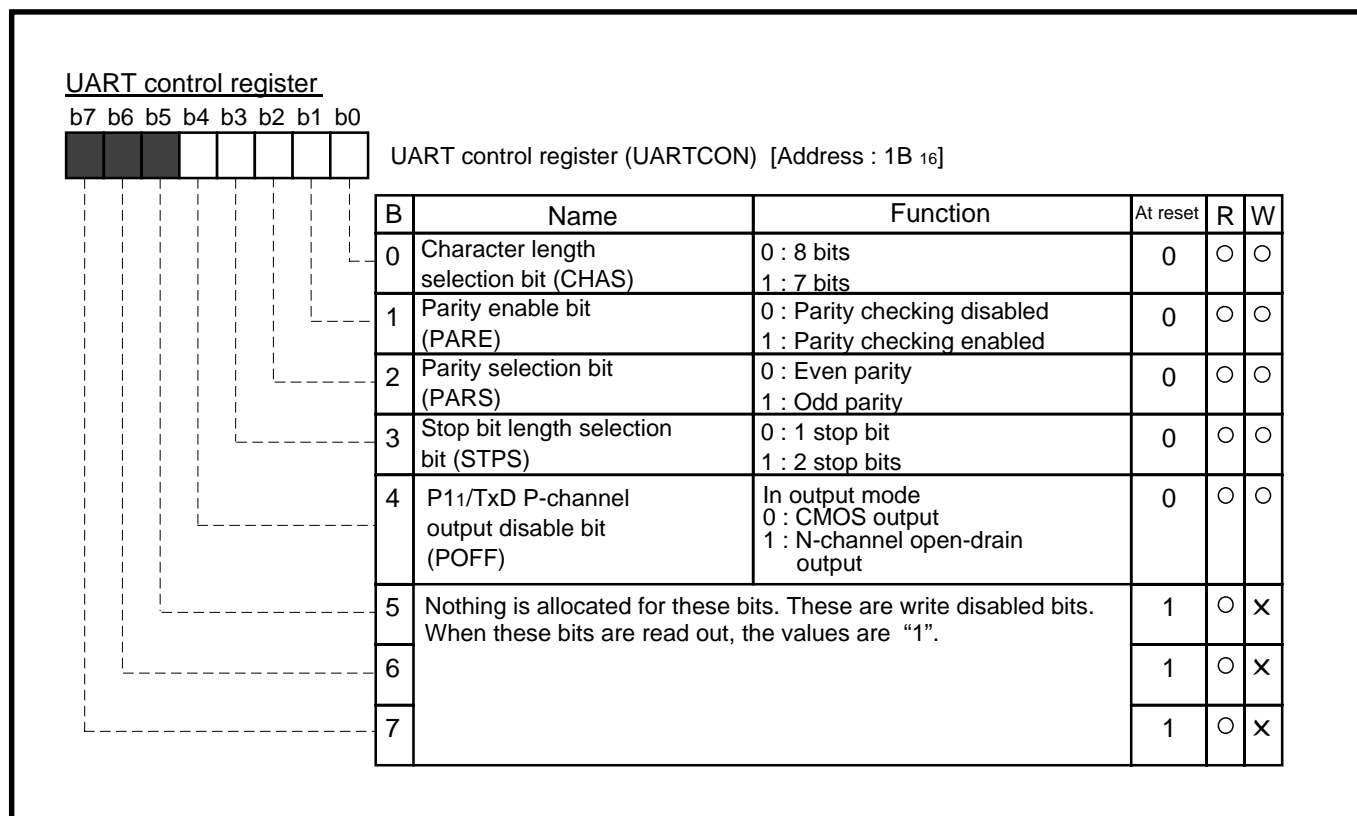


Fig. 3.5.9 Structure of UART control register

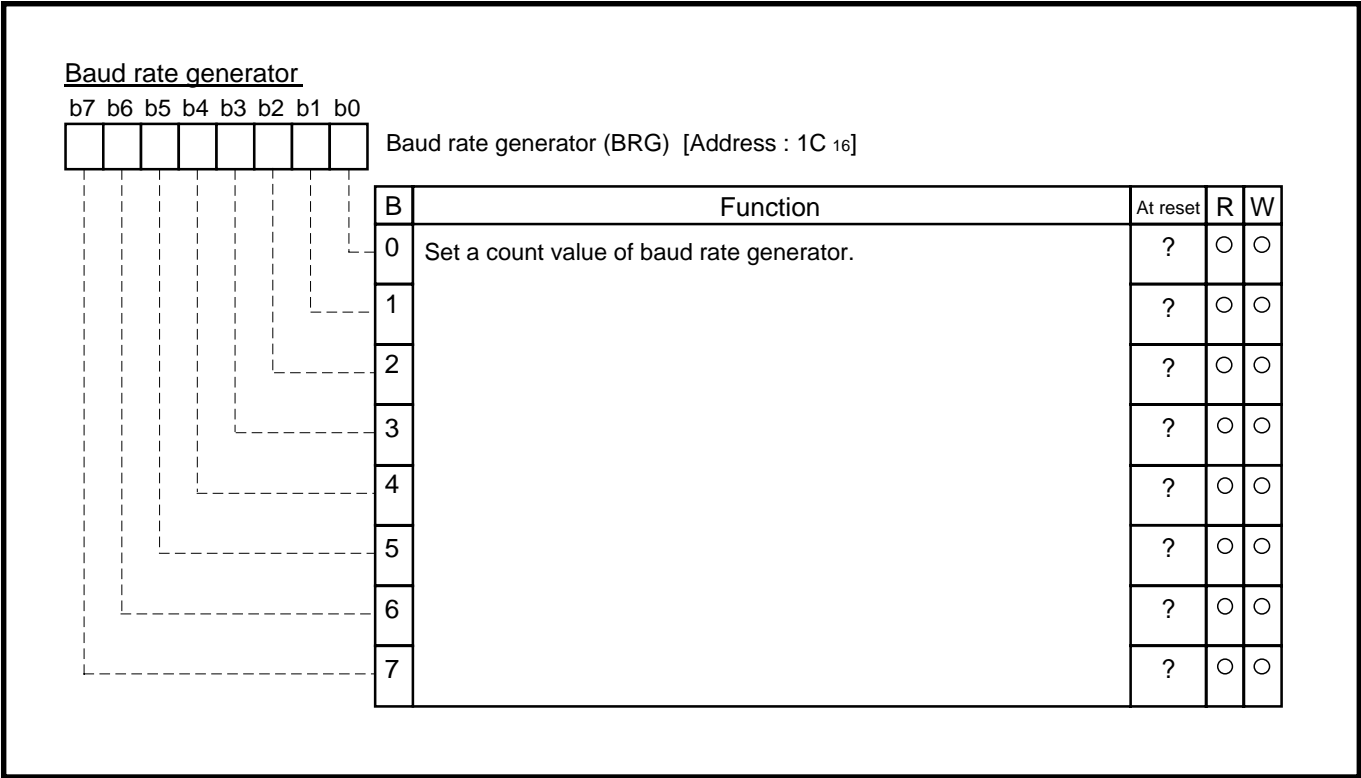


Fig. 3.5.10 Structure of Baud rate generator

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

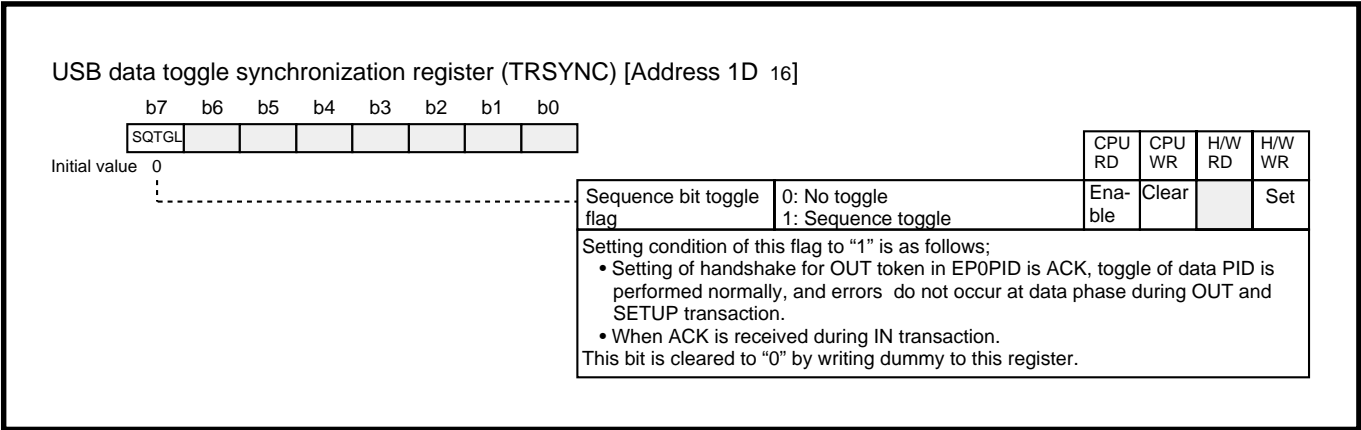


Fig. 3.5.11 Structure of USB data toggle synchronization register

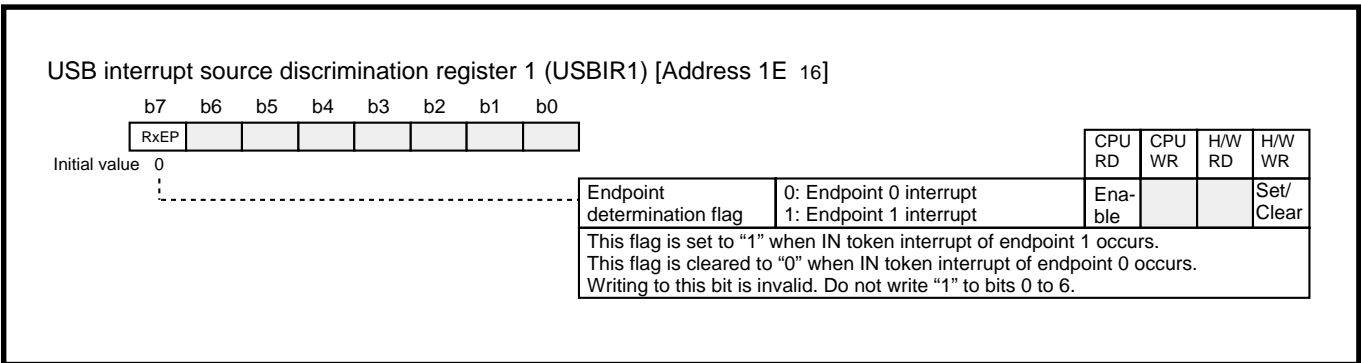


Fig. 3.5.12 Structure of USB interrupt source discrimination register 1

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

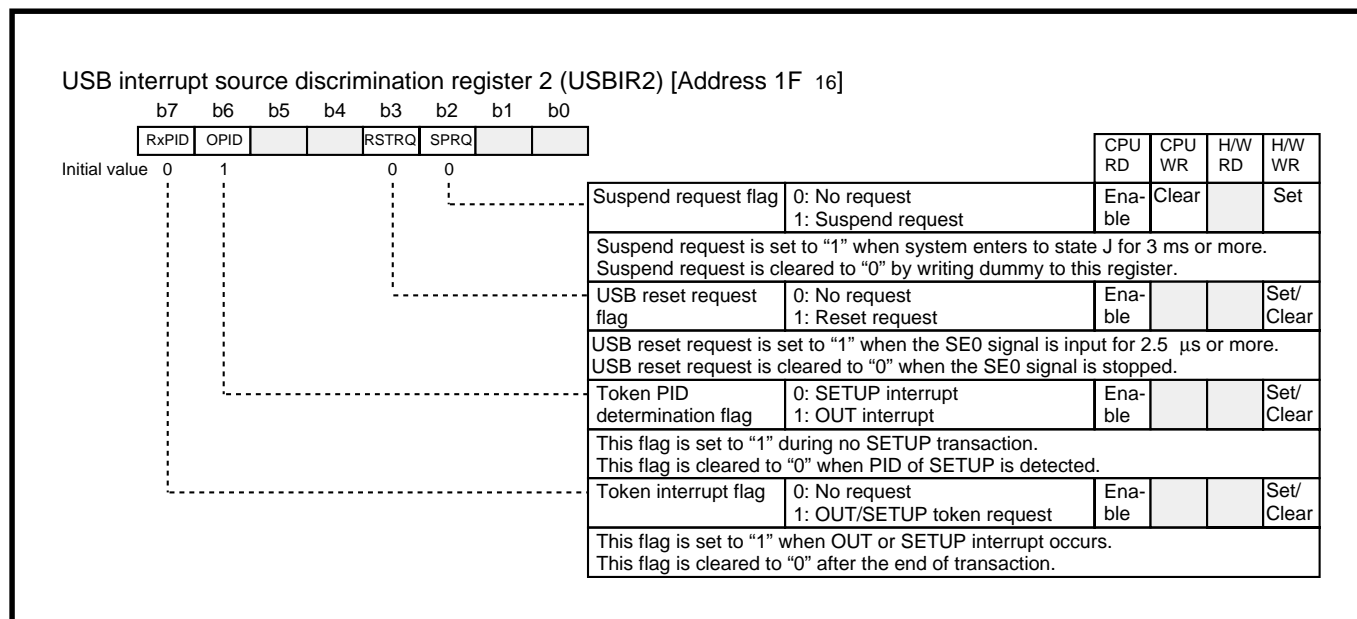


Fig. 3.5.13 Structure of USB interrupt source discrimination register 2

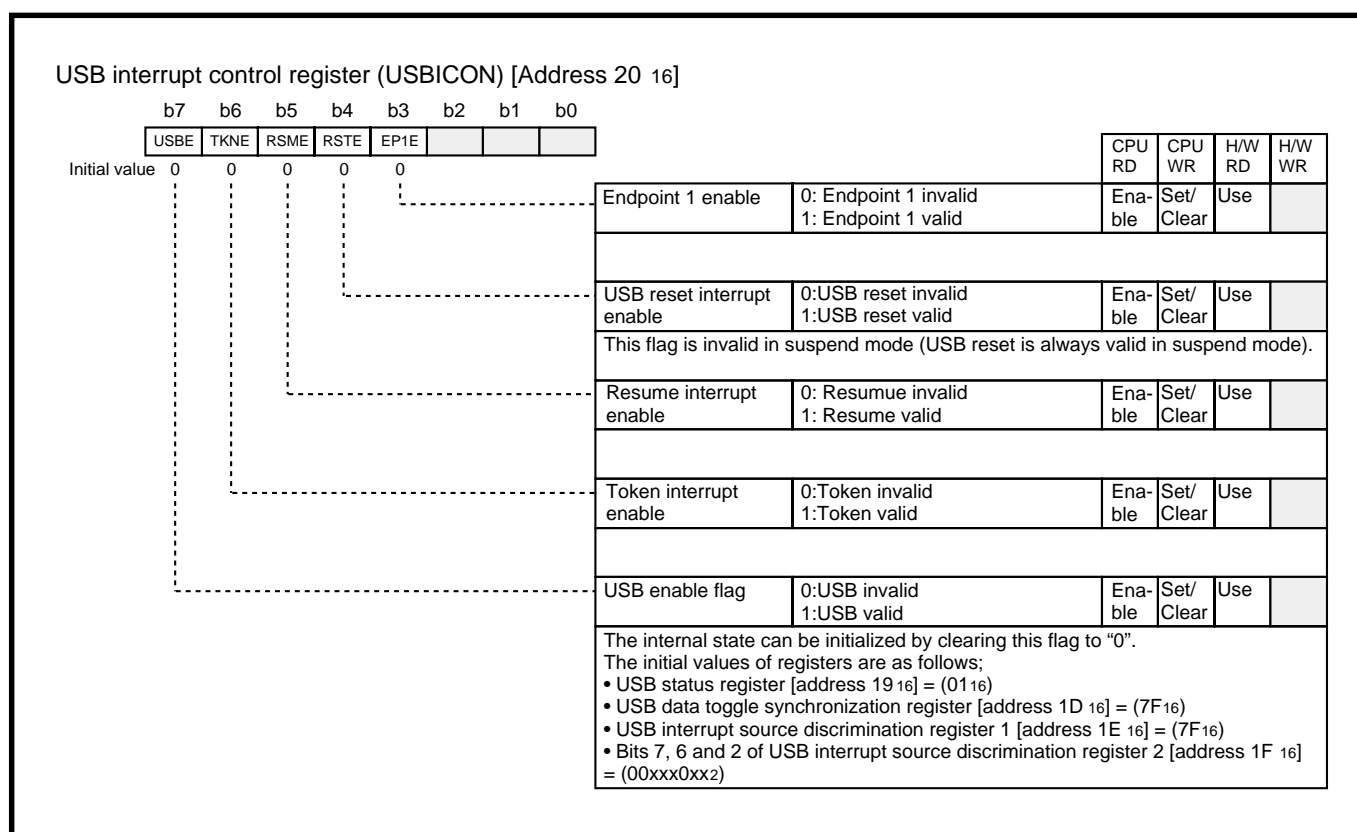


Fig. 3.5.14 Structure of USB interrupt control register

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

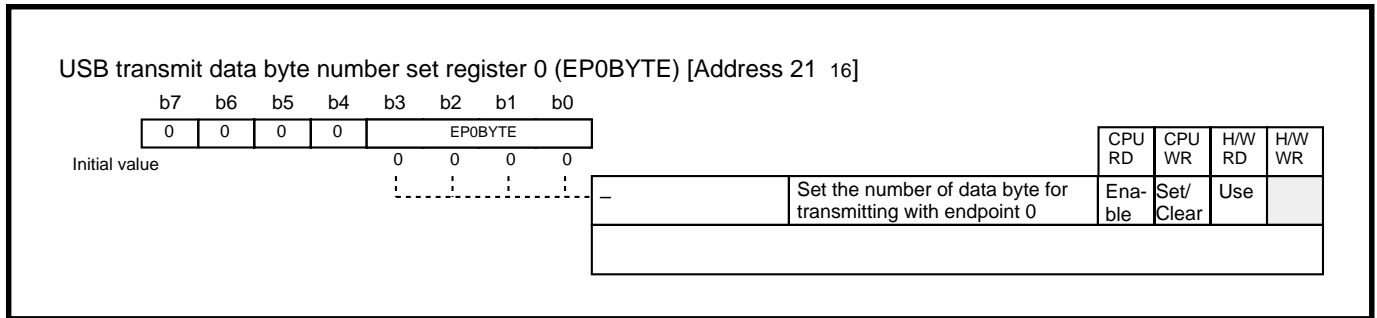


Fig. 3.5.15 Structure of USB transmit data byte number set register 0

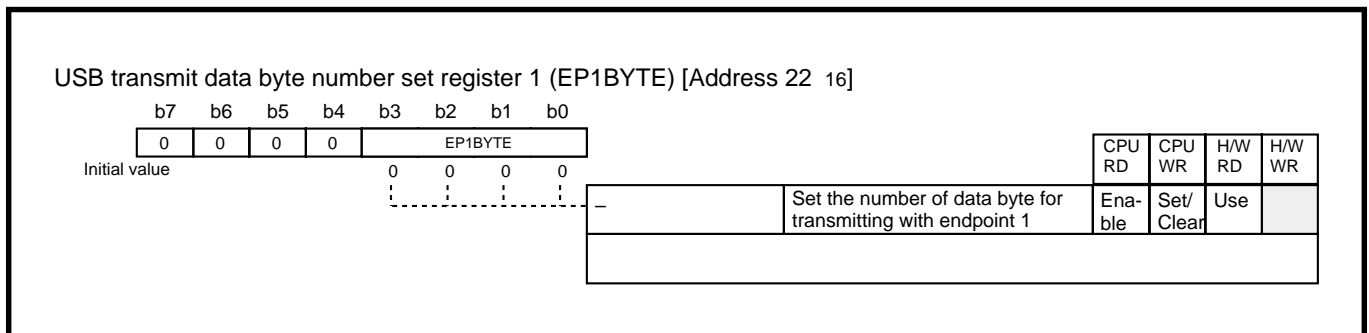


Fig. 3.5.16 Structure of USB transmit data byte number set register 1

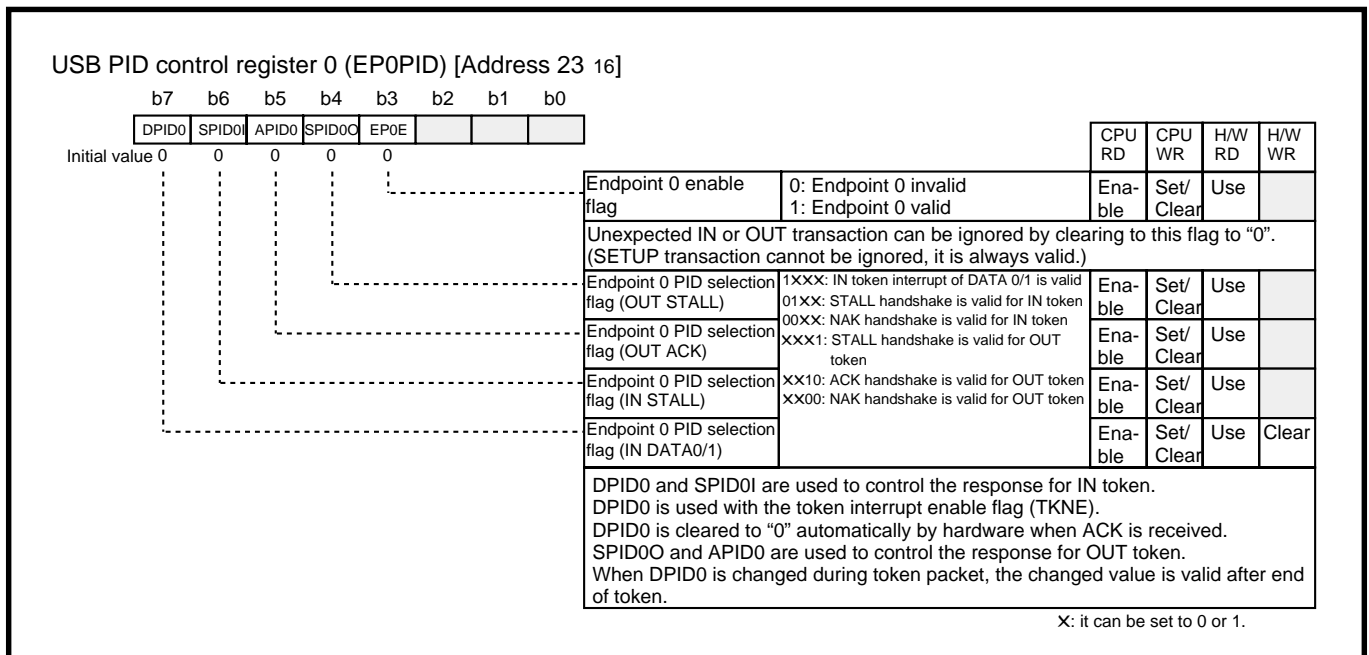


Fig. 3.5.17 Structure of USB PID control register 0

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

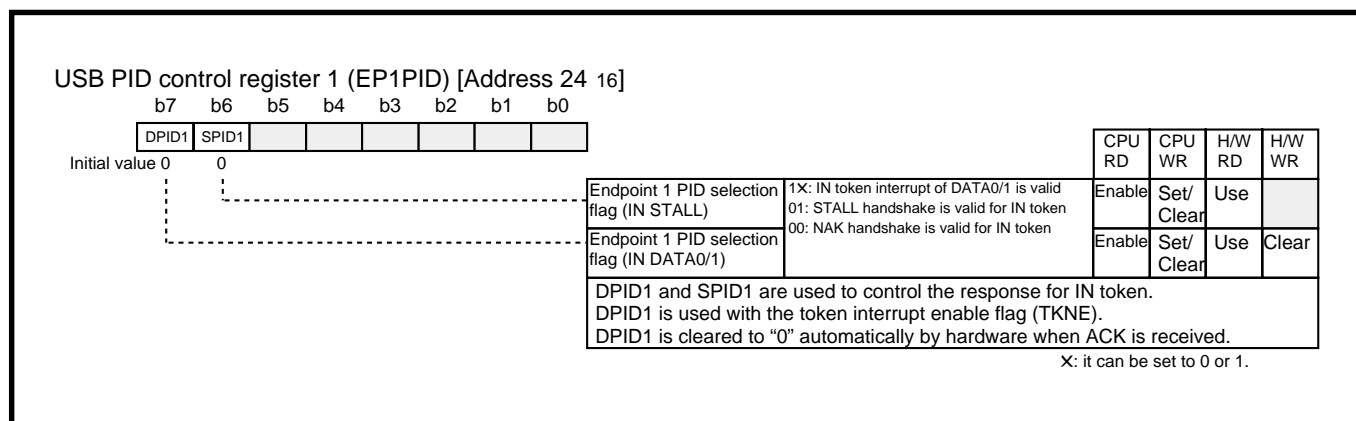


Fig. 3.5.18 Structure of USB PID control register 1

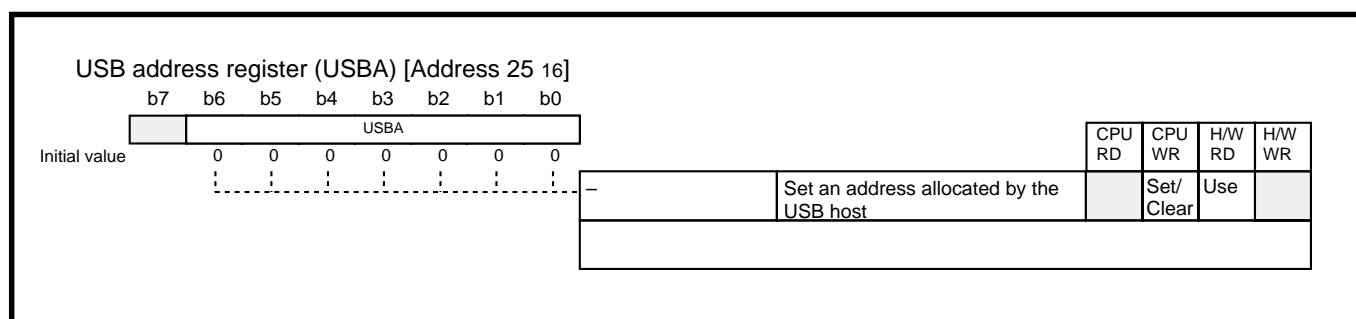


Fig. 3.5.19 Structure of USB address register

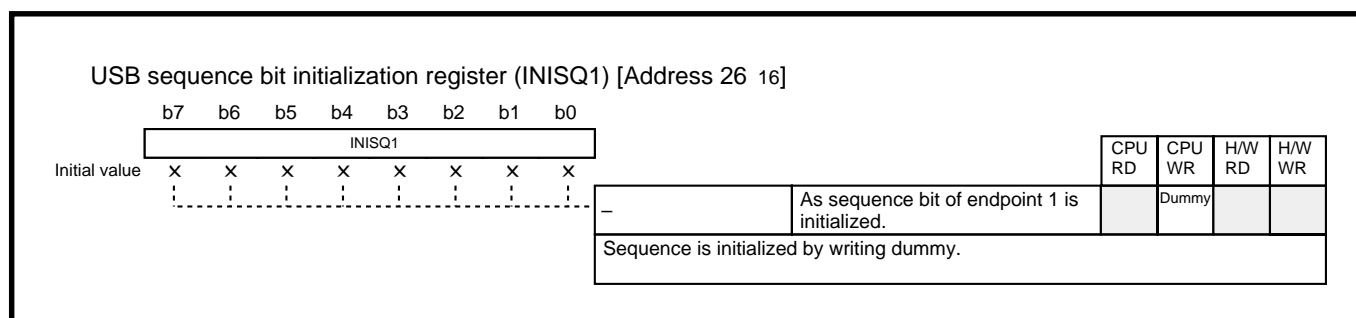


Fig. 3.5.20 Structure of USB sequence bit initialization register

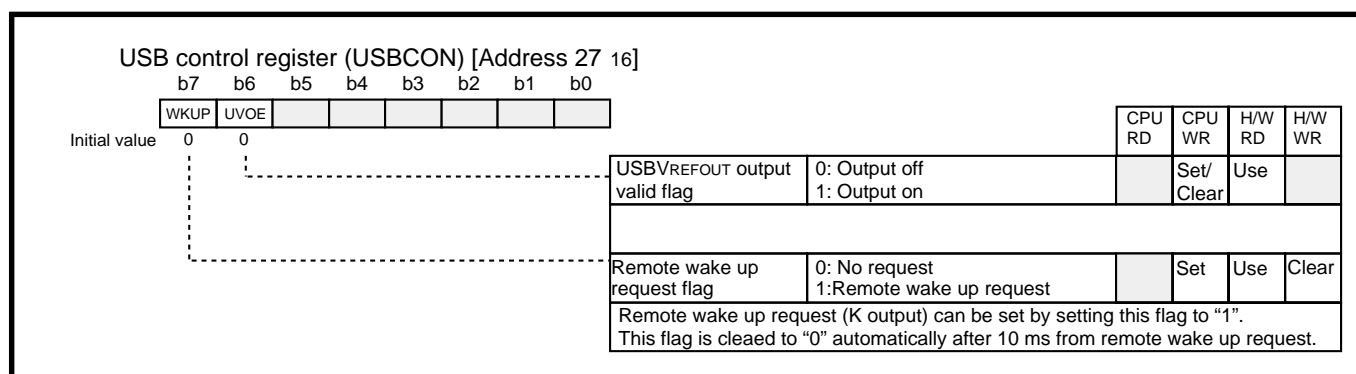


Fig. 3.5.21 Structure of USB control register

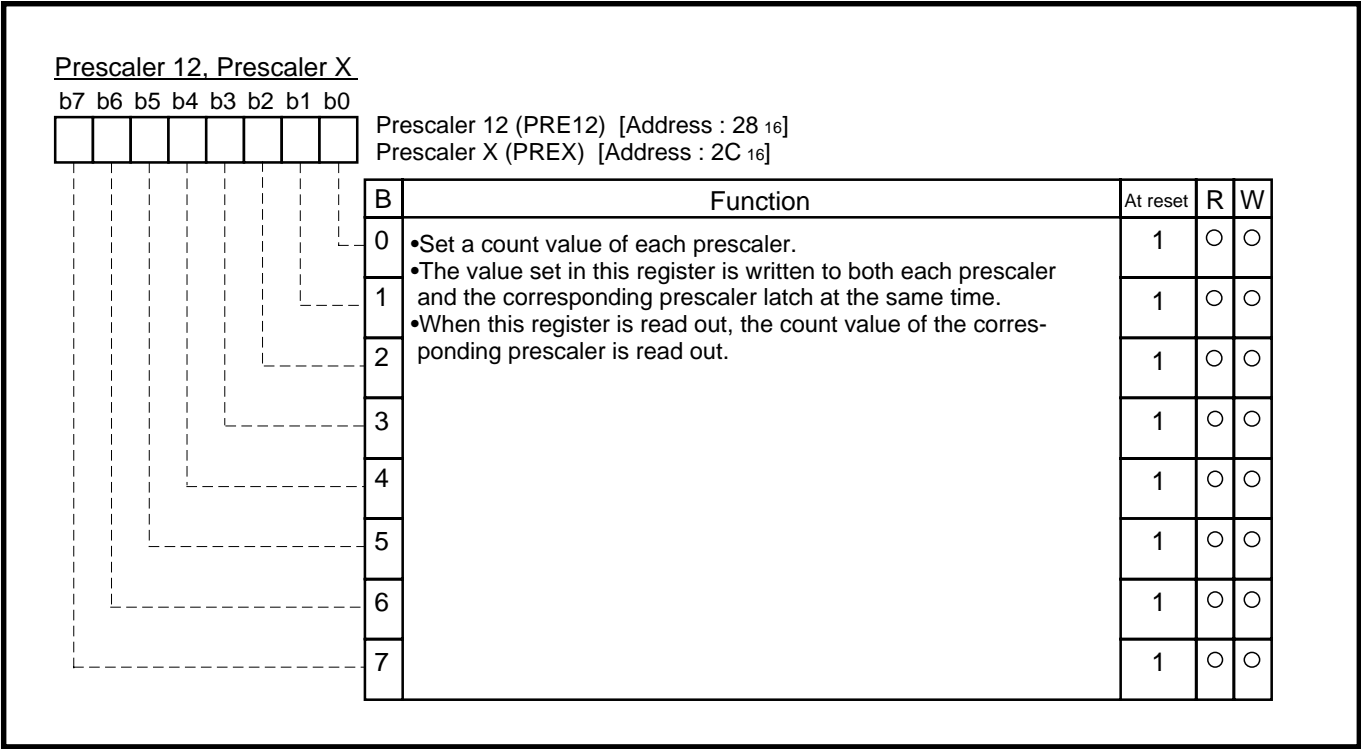


Fig. 3.5.22 Structure of Prescaler 12, Prescaler X

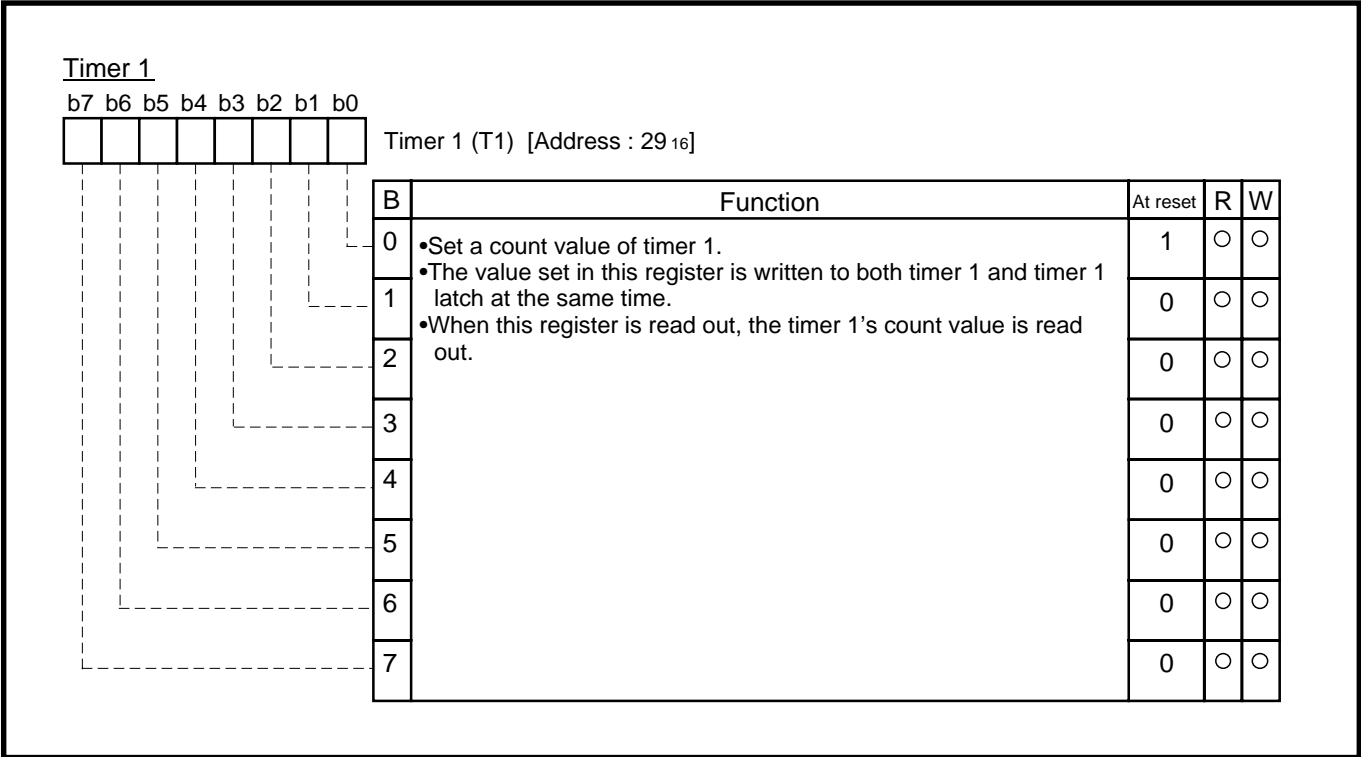


Fig. 3.5.23 Structure of Timer 1

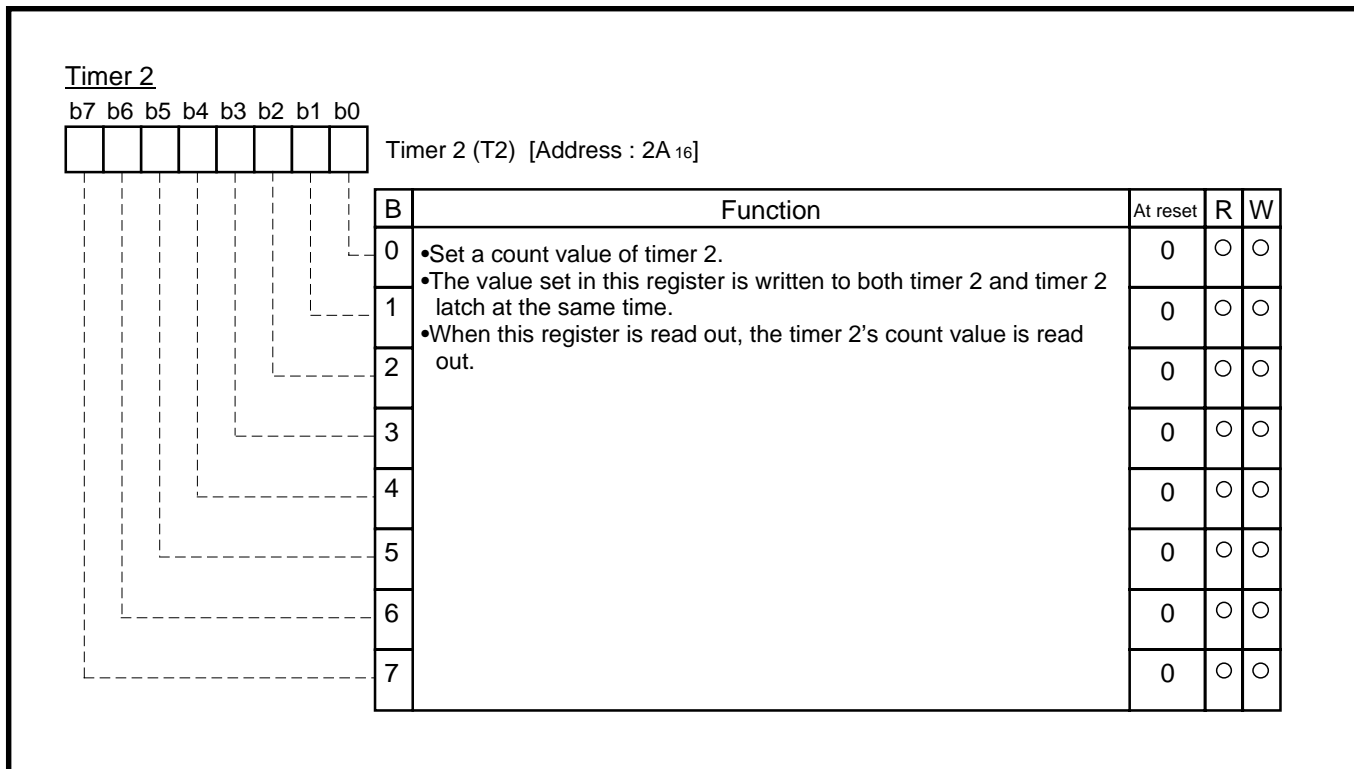


Fig. 3.5.24 Structure of Timer 2

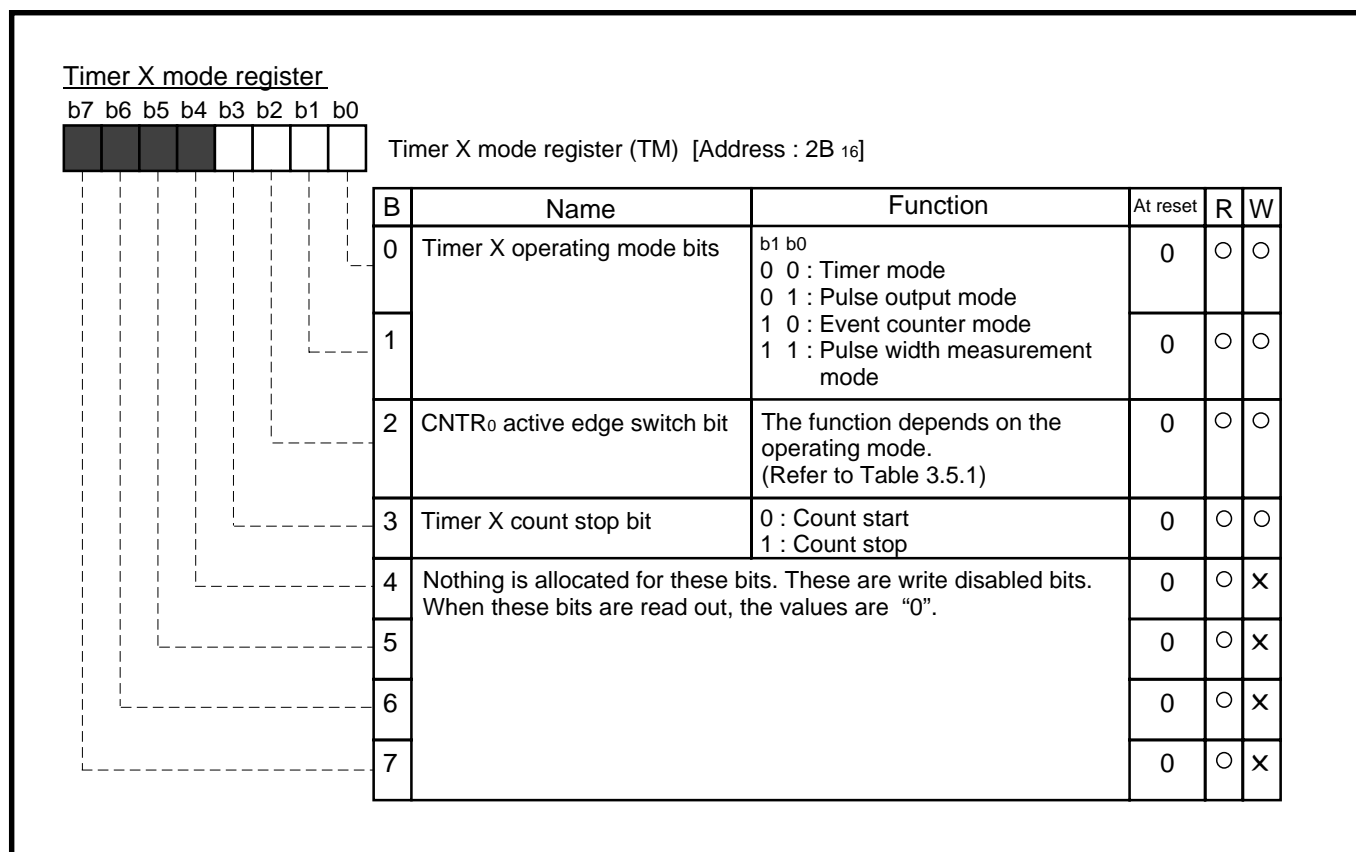


Fig. 3.5.25 Structure of Timer X mode register

Table 3.5.1 CNTR₀ active edge switch bit function

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

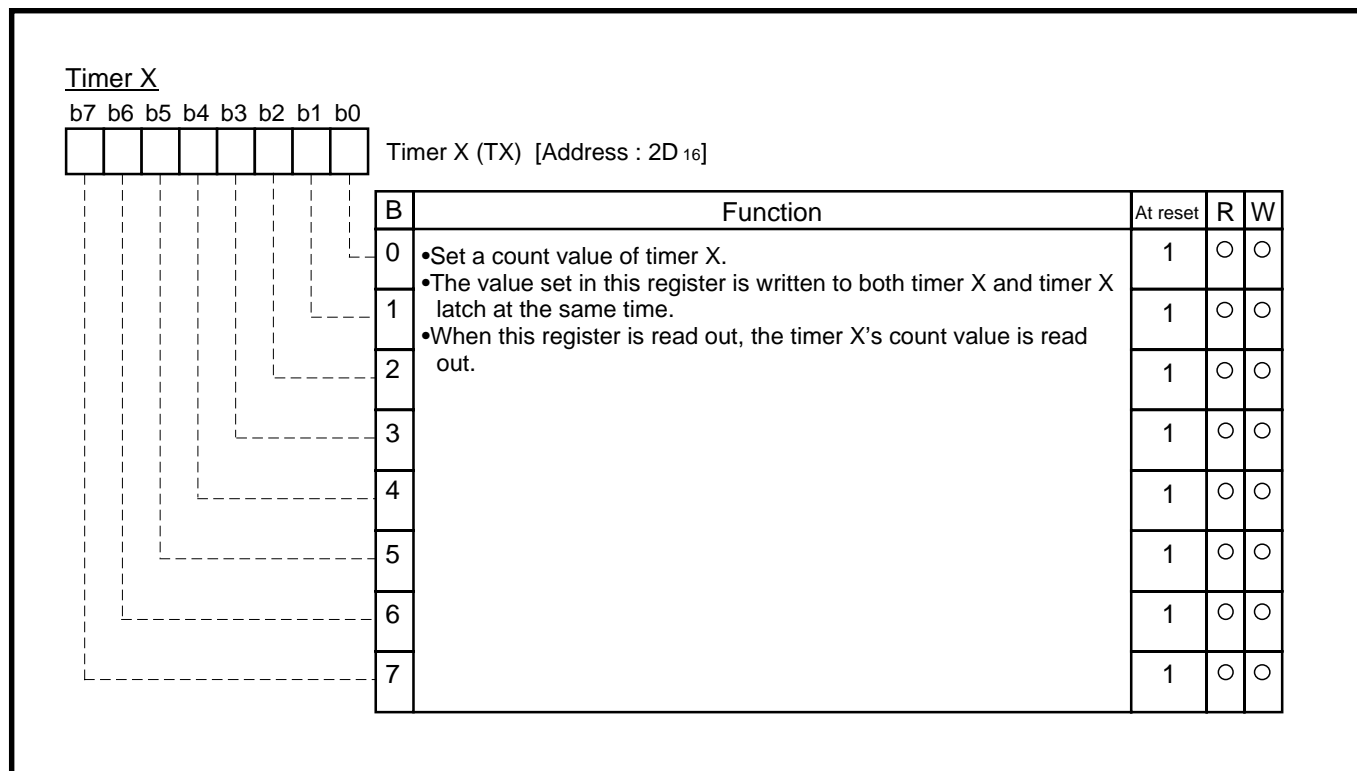


Fig. 3.5.26 Structure of Timer X

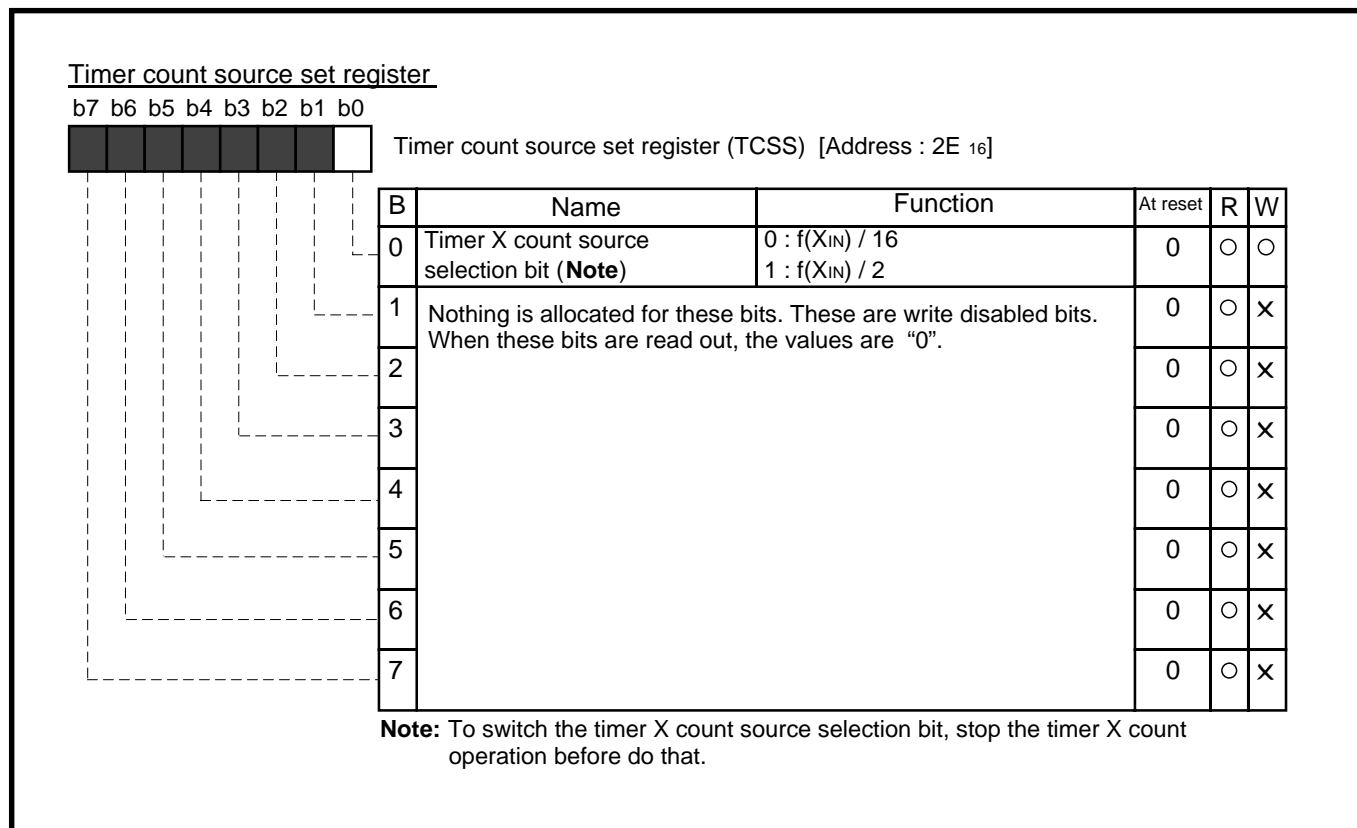


Fig. 3.5.27 Structure of Timer count source set register

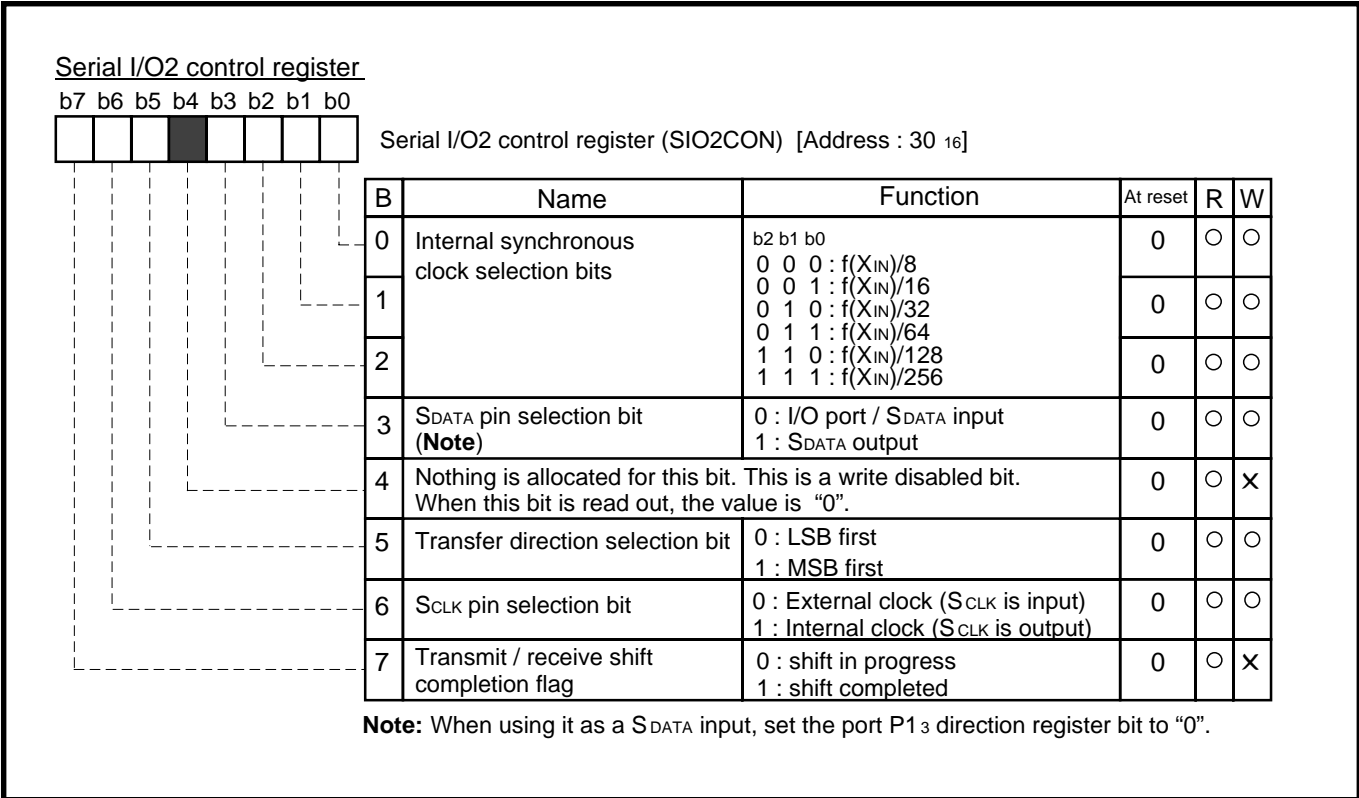


Fig. 3.5.28 Structure of Serial I/O2 control register

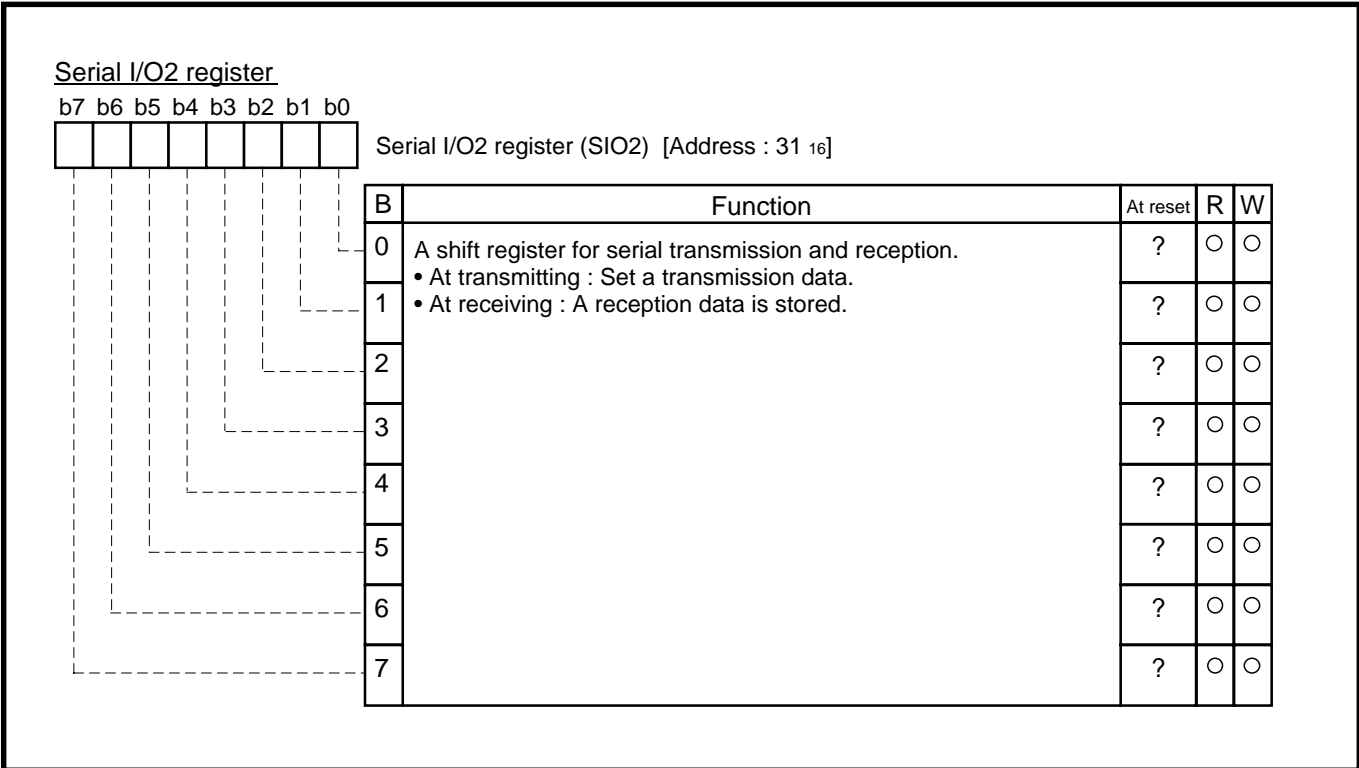


Fig. 3.5.29 Structure of Serial I/O2 register

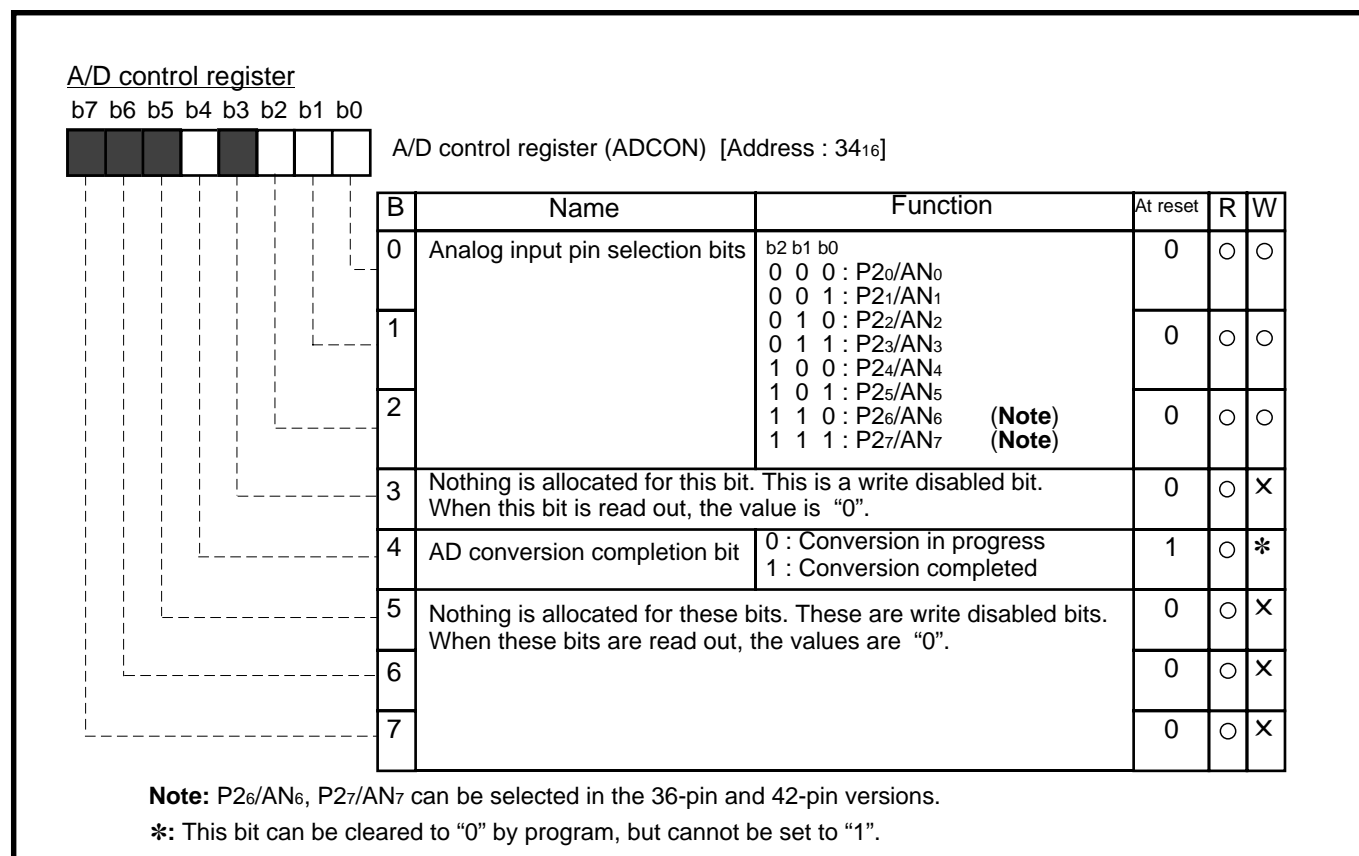


Fig. 3.5.30 Structure of A/D control register

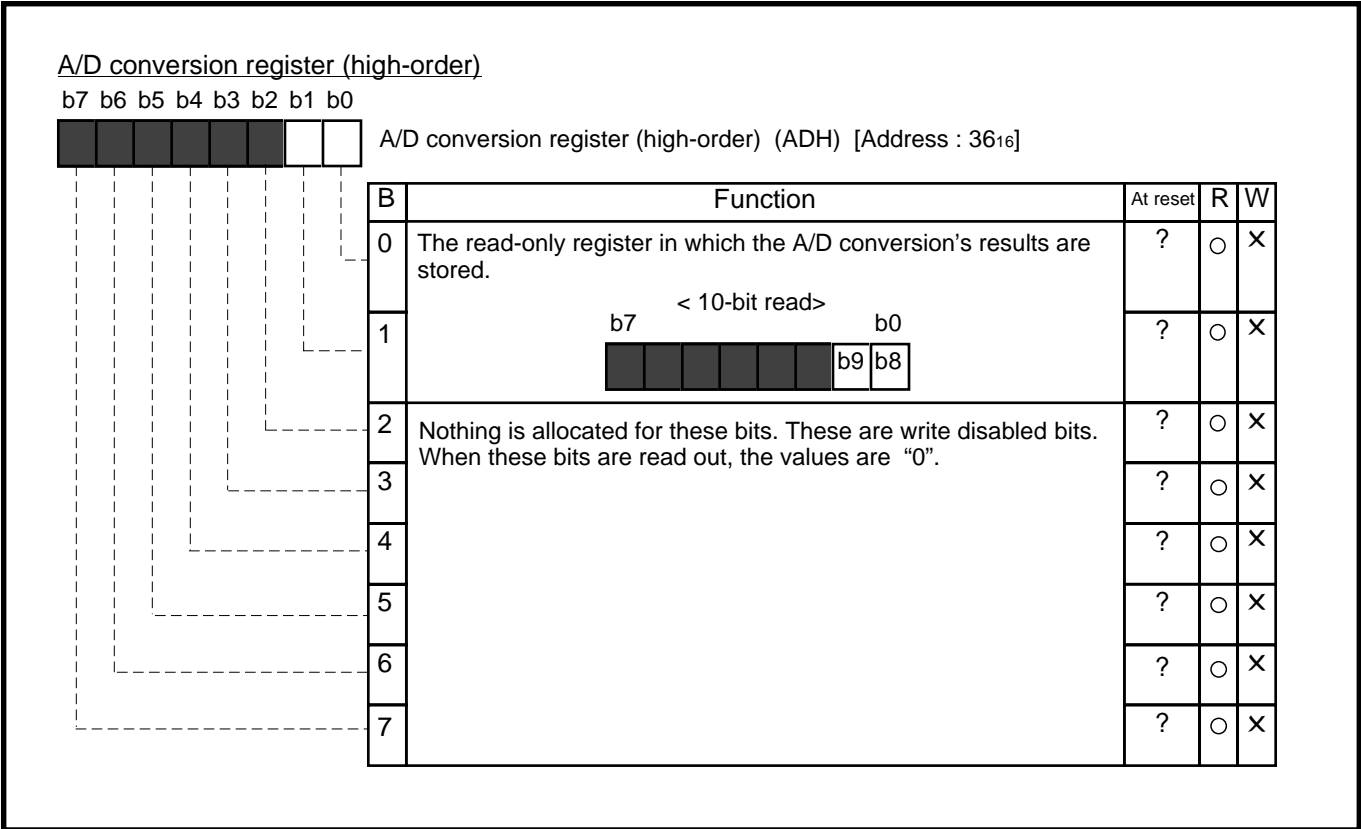


Fig. 3.5.31 Structure of A/D conversion register (high-order)

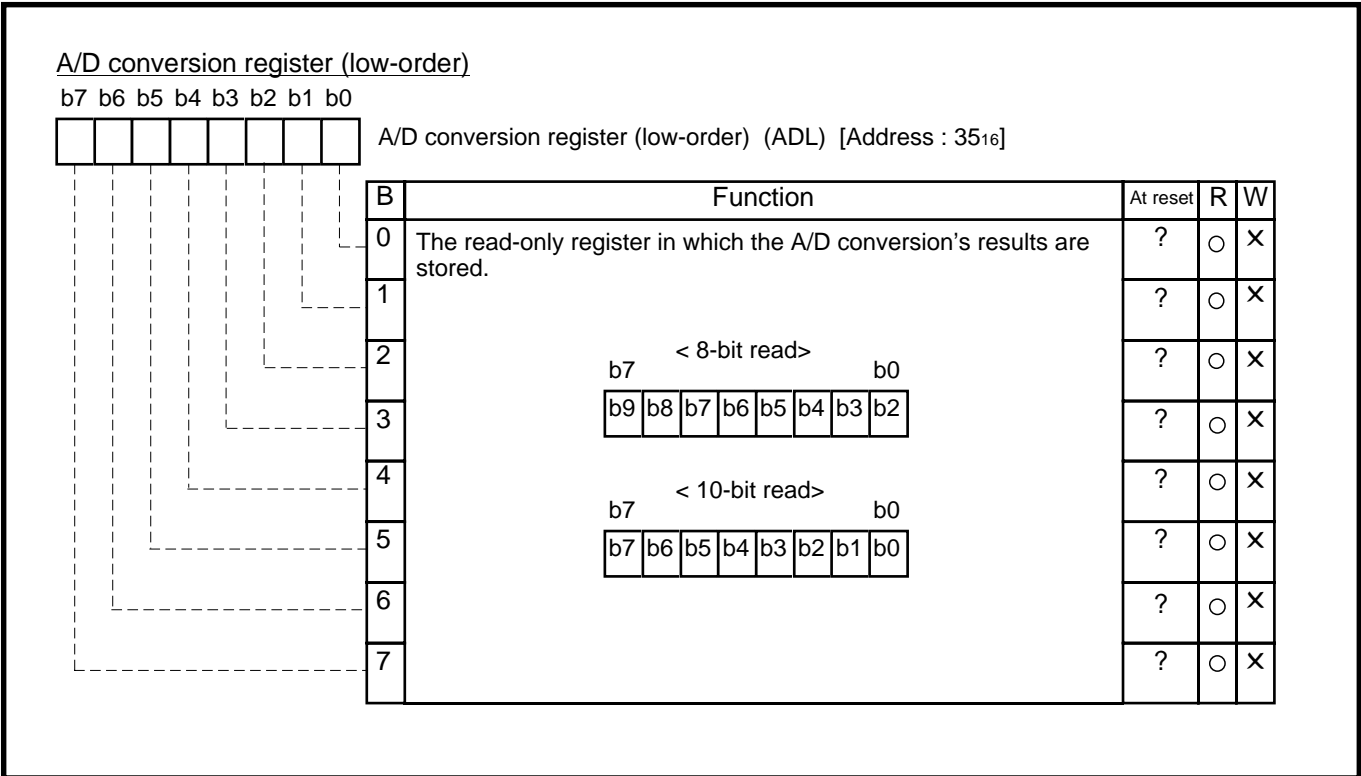


Fig. 3.5.32 Structure of A/D conversion register (low-order)

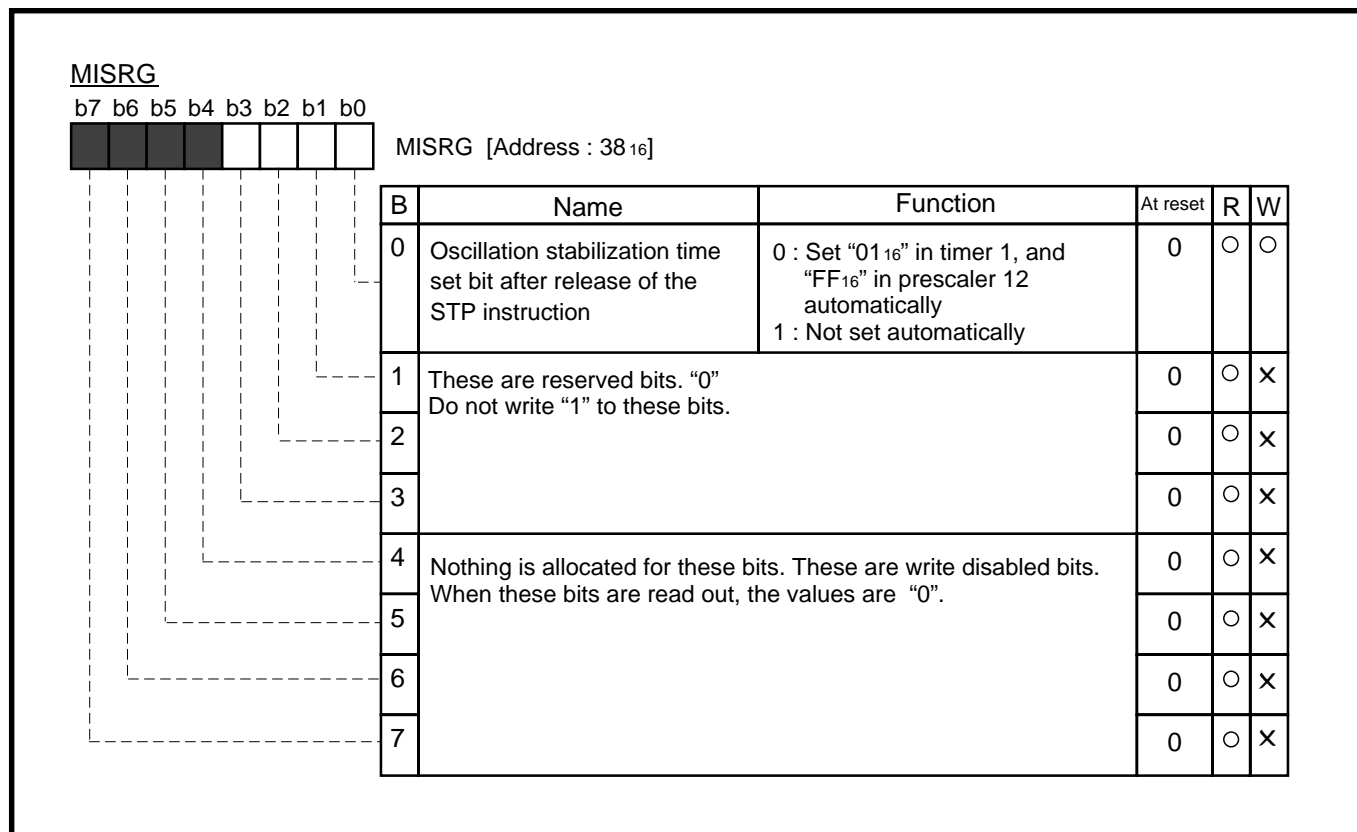


Fig. 3.5.33 Structure of MISRG

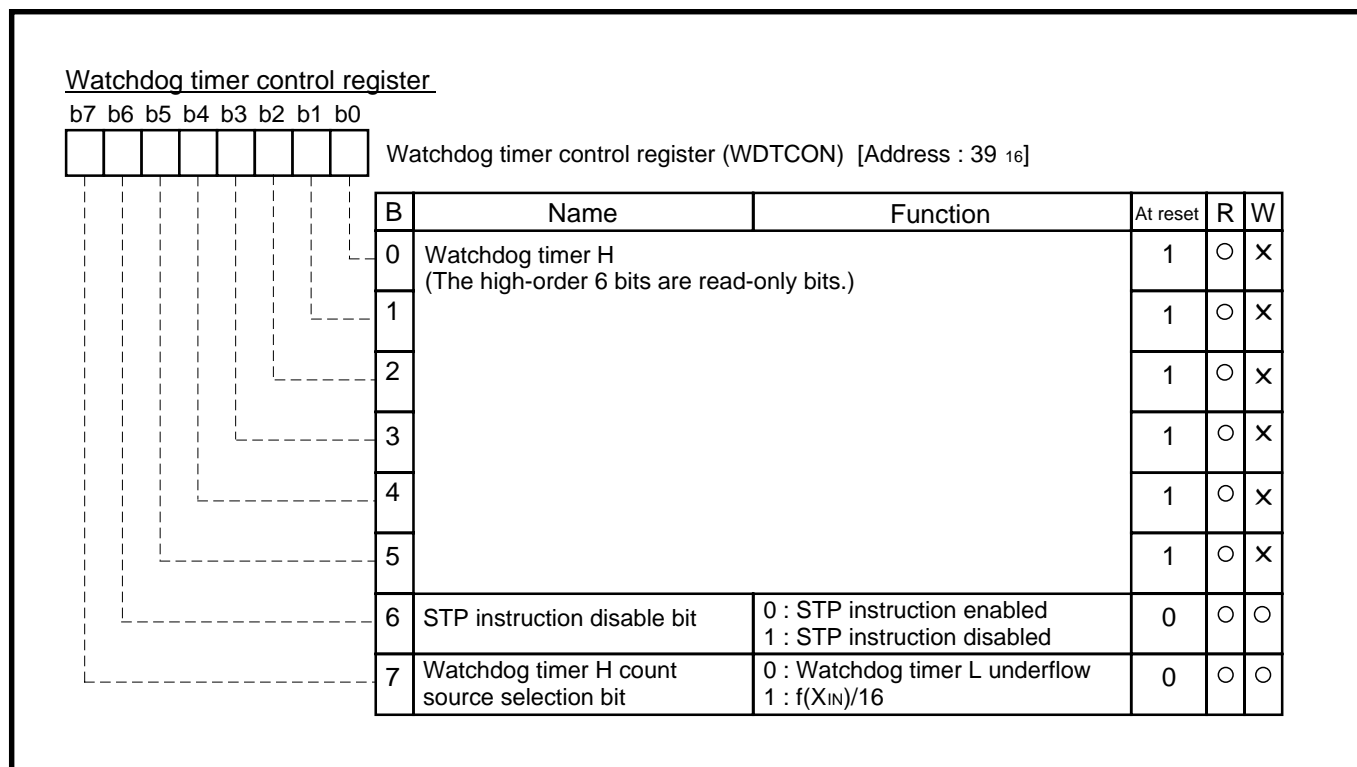


Fig. 3.5.34 Structure of Watchdog timer control register

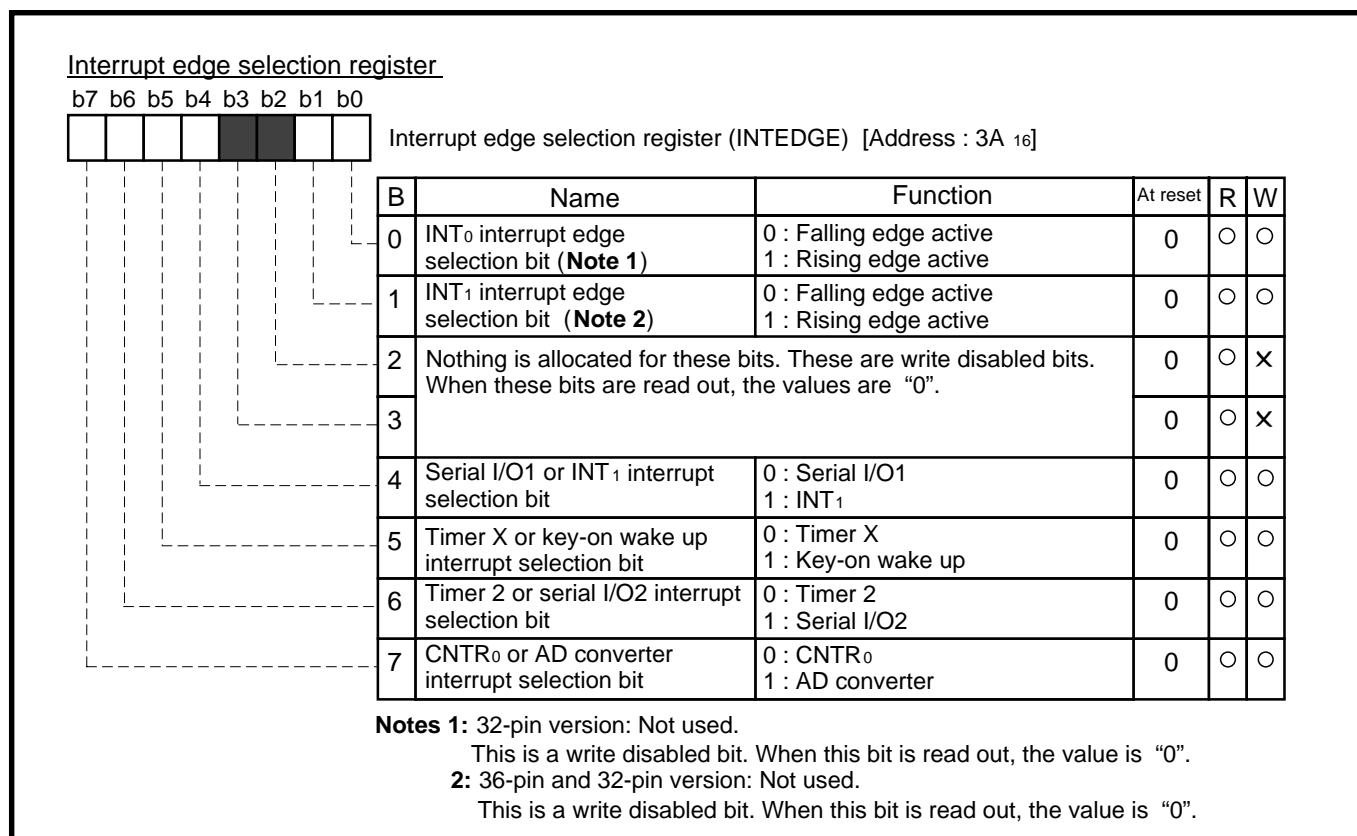


Fig. 3.5.35 Structure of Interrupt edge selection register

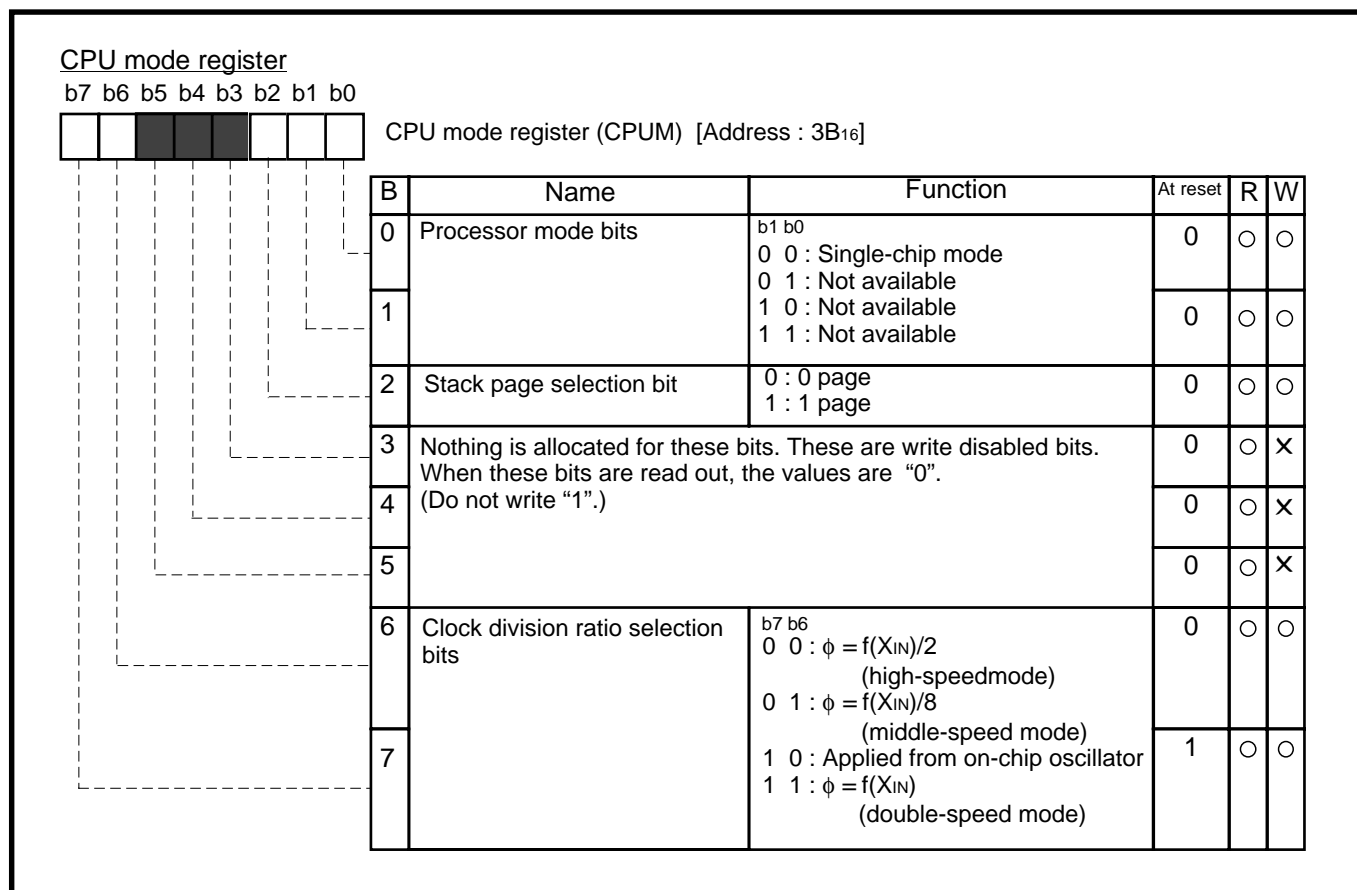


Fig. 3.5.36 Structure of CPU mode register

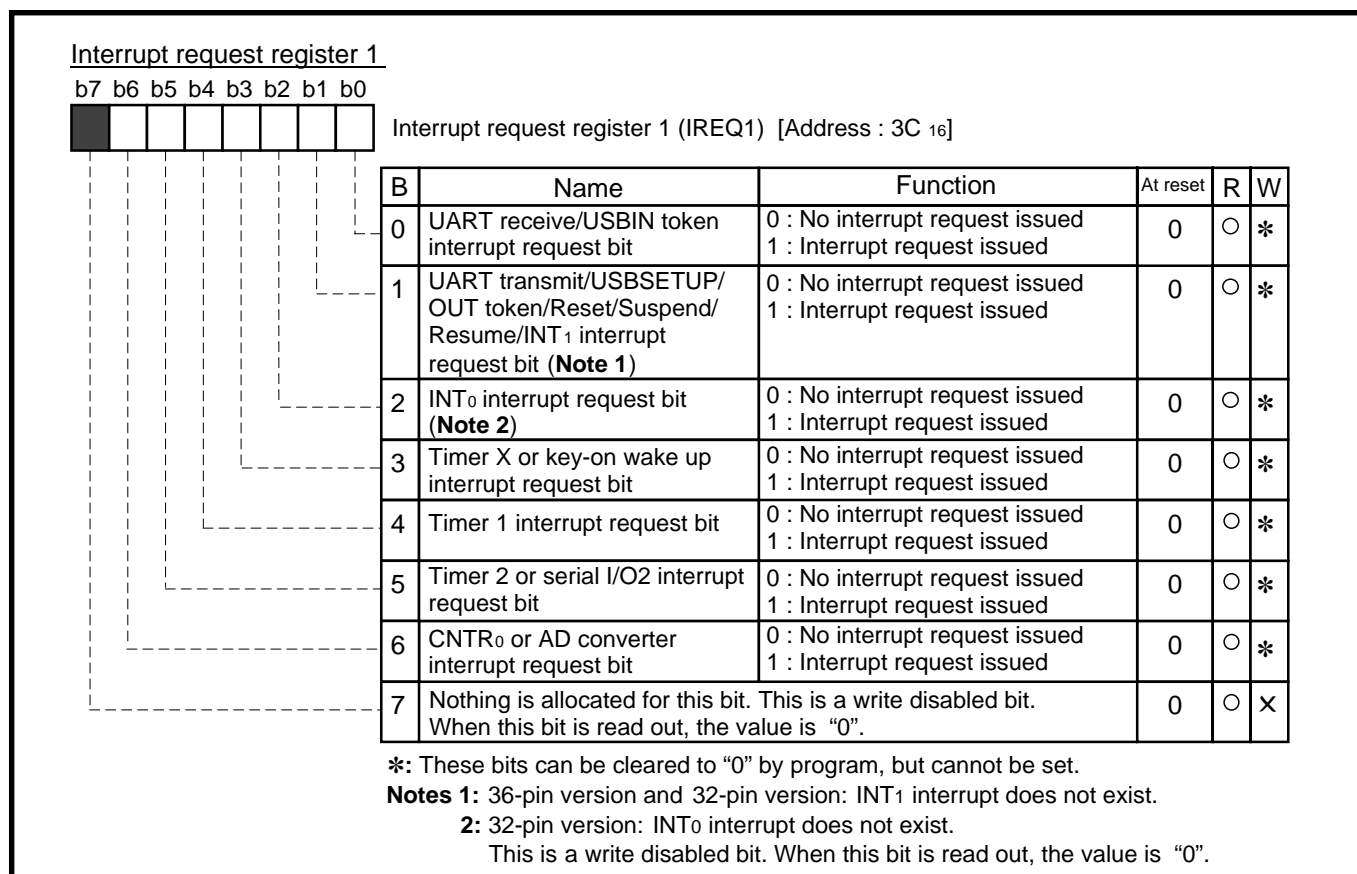


Fig. 3.5.37 Structure of Interrupt request register 1

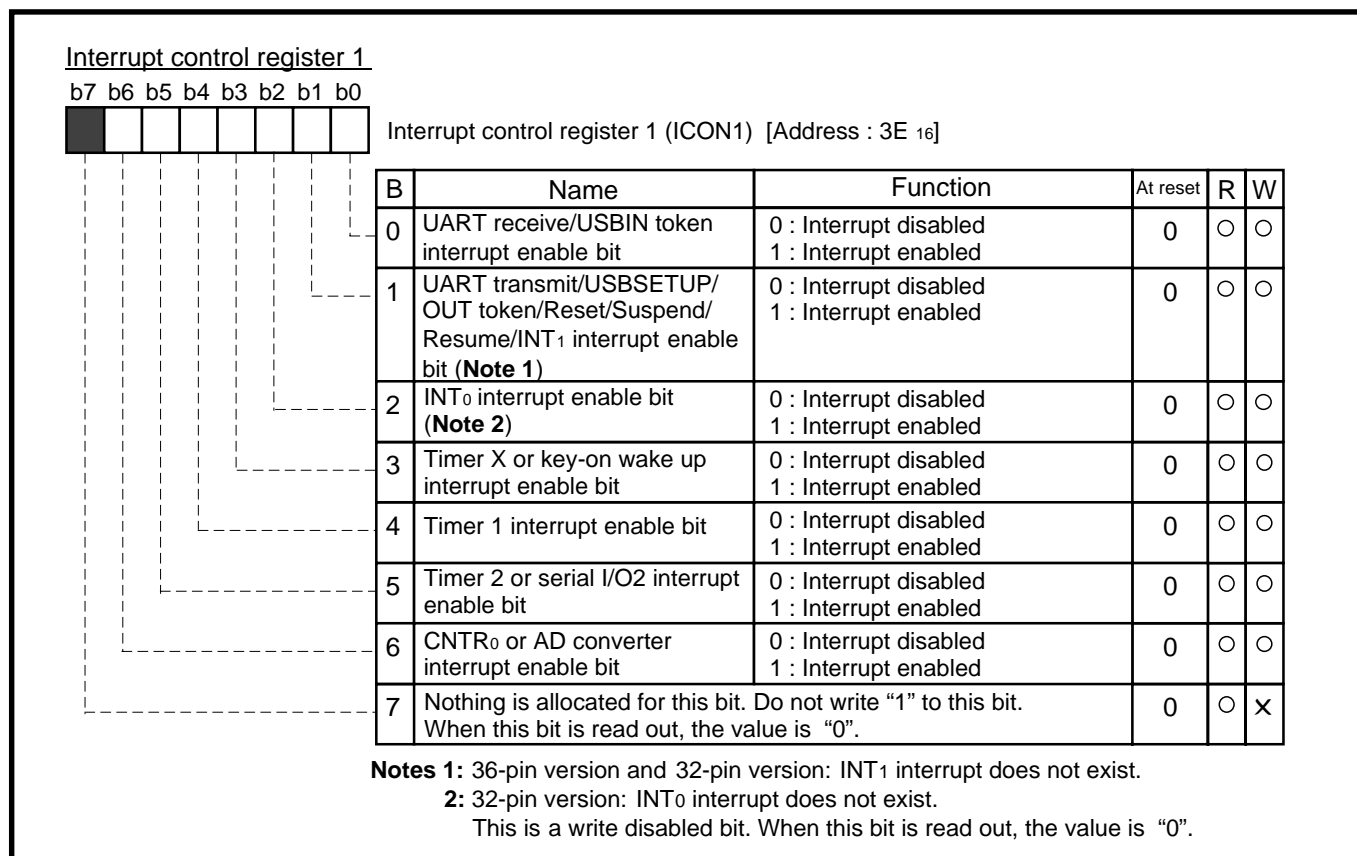
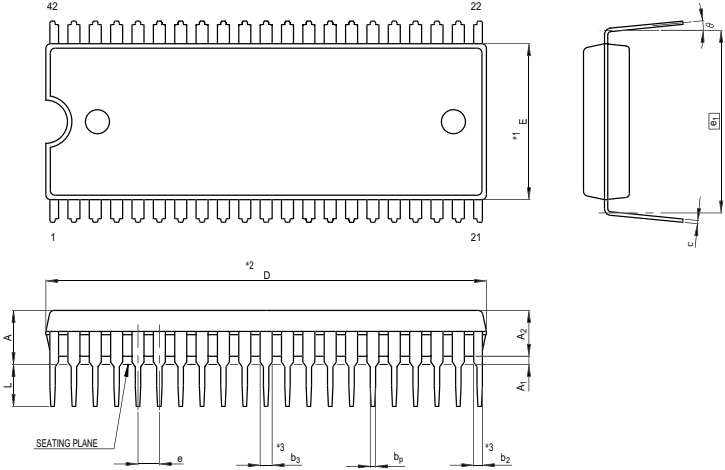


Fig. 3.5.38 Structure of Interrupt control register 1

3.6 Package outline

PRDP0042BA-A

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-SDIP42-13x36.72-1.78	PRDP0042BA-A	42P4B	4.1g

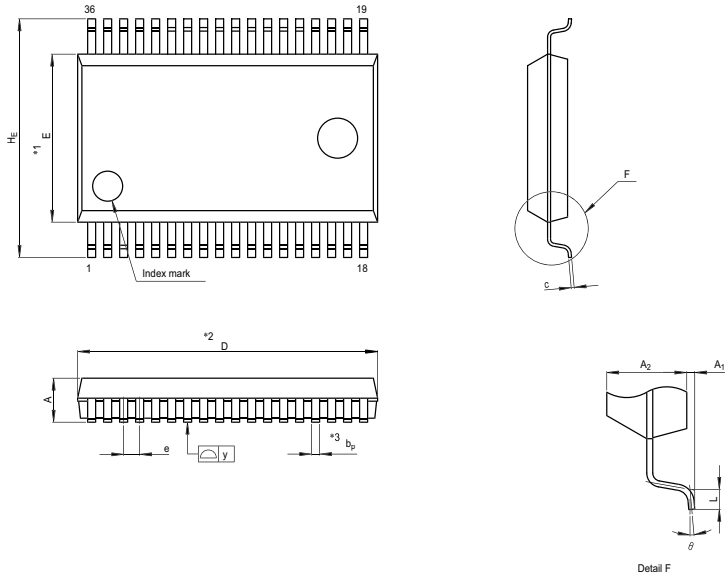


NOTE)
 1. DIMENSIONS 1A AND 2D
 DO NOT INCLUDE MOLD FLASH.
 2. DIMENSION 3D DOES NOT
 INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
e_1	14.94	15.24	15.54
D	36.5	36.7	36.9
E	12.85	13.0	13.15
A	—	—	5.5
A_1	0.51	—	—
A_2	—	3.8	—
b_p	0.35	0.45	0.55
b_2	0.63	0.73	1.03
b_3	0.9	1.0	1.3
c	0.22	0.27	0.34
θ	0°	—	15°
e	1.528	1.778	2.028
L	3.0	—	—

PRSP0036GA-A

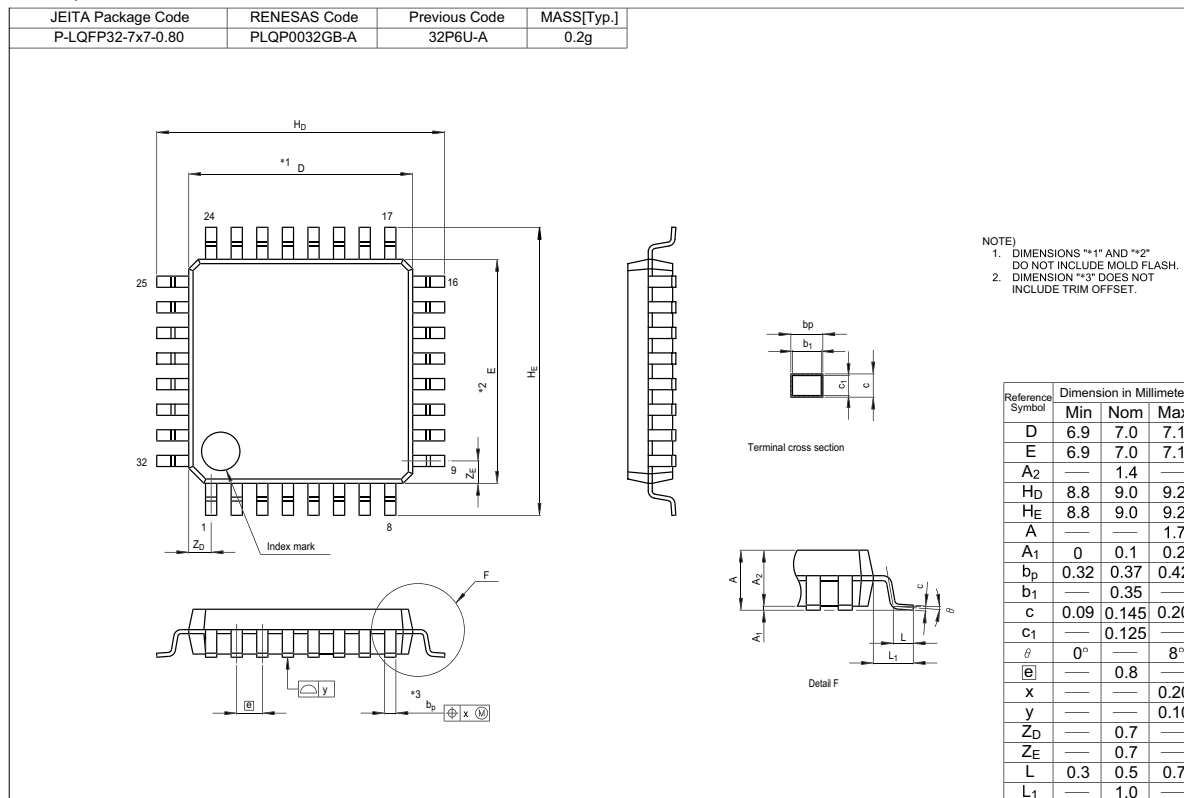
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-SSOP36-8.4x15-0.80	PRSP0036GA-A	36P2R-A	0.5g



NOTE)
 1. DIMENSIONS 1A AND 2D
 DO NOT INCLUDE MOLD FLASH.
 2. DIMENSION 3D DOES NOT
 INCLUDE TRIM OFFSET.


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	14.8	15.0	15.2
E	8.2	8.4	8.6
A_2	—	2.0	—
A	—	—	2.4
A_1	0.05	—	—
b_p	0.35	0.4	0.5
c	0.13	0.15	0.2
θ	0°	—	10°
H_E	11.63	11.93	12.23
e	0.65	0.8	0.95
y	—	—	0.15
L	0.3	0.5	0.7


PLQP0032GB-A




3.7 List of instruction code

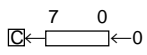
D3 – D0 D7 – D4	Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND, X	JSR ZP, IND	BBS 0, A	—	ORA ZP	ASL ZP	BBS 0, ZP	PHP	ORA IMM	ASL A	SEB 0, A	—	ORA ABS	ASL ABS	SEB 0, ZP
0001	1	BPL	ORA IND, Y	CLT	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

3.8 Machine instructions

Symbol	Function	Details	Addressing mode																				
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R					
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#			
ADC (Note 1) (Note 5)	When T = 0 $A \leftarrow A + M + C$ When T = 1 $M(X) \leftarrow M(X) + M + C$	When T = 0, this instruction adds the contents M, C, and A; and stores the results in A and C. When T = 1, this instruction adds the contents of M(X), M and C; and stores the results in M(X) and C. When T=1, the contents of A remain unchanged, but the contents of status flags are changed. M(X) represents the contents of memory where is indicated by X.				69	2	2										65	3	2			
AND (Note 1)	When T = 0 $A \leftarrow A \wedge M$ When T = 1 $M(X) \leftarrow M(X) \wedge M$	When T = 0, this instruction transfers the contents of A and M to the ALU which performs a bit-wise AND operation and stores the result back in A. When T = 1, this instruction transfers the contents M(X) and M to the ALU which performs a bit-wise AND operation and stores the results back in M(X). When T = 1, the contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				29	2	2										25	3	2			
ASL		This instruction shifts the content of A or M by one bit to the left, with bit 0 always being set to 0 and bit 7 of A or M always being contained in C.							0A	2	1							06	5	2			
BBC (Note 4)	A_i or $M_i = 0?$	This instruction tests the designated bit i of M or A and takes a branch if the bit is 0. The branch address is specified by a relative address. If the bit is 1, next instruction is executed.										$\frac{13}{20i}$	4	2				$\frac{17}{20i}$	5	3			
BBS (Note 4)	A_i or $M_i = 1?$	This instruction tests the designated bit i of the M or A and takes a branch if the bit is 1. The branch address is specified by a relative address. If the bit is 0, next instruction is executed.										$\frac{03}{20i}$	4	2				$\frac{07}{20i}$	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.																					

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					
55	4	2				4D	4	3	5D	5	3	59	5	3																												
F6	6	2				EE	6	3	FE	7	3																															
						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																												
			B6	4	2	AE	4	3				BE	5	3																												
B4	4	2				AC	4	3	BC	5	3																															

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

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	V	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
ZP, IND	Zero page indirect absolute addressing mode	ADH	8 high-order bits of address
IND, X	Indirect X addressing mode	ADL	8 low-order bits of address
IND, Y	Indirect Y addressing mode	FF	FF in Hexadecimal notation
REL	Relative addressing mode	nn	Immediate value
SP	Special page addressing mode	zz	Zero page address
C	Carry flag	M	Memory specified by address designation of any addressing mode
Z	Zero flag	M(X)	Memory of address indicated by contents of index register X
I	Interrupt disable flag	M(S)	Memory of address indicated by contents of stack pointer
D	Decimal mode flag	M(ADH, ADL)	Contents of memory at address indicated by ADH and ADL, in ADH is 8 high-order bits and ADL is 8 low-order bits.
B	Break flag	M(00, ADL)	Contents of address indicated by zero page ADL
T	X-modified arithmetic mode flag	Ai	Bit i (i = 0 to 7) of accumulator
V	Overflow flag	Mi	Bit i (i = 0 to 7) of memory
N	Negative flag	OP	Opcode
		n	Number of cycles
		#	Number of bytes

3.9 SFR memory map

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

3.10 Pin configurations

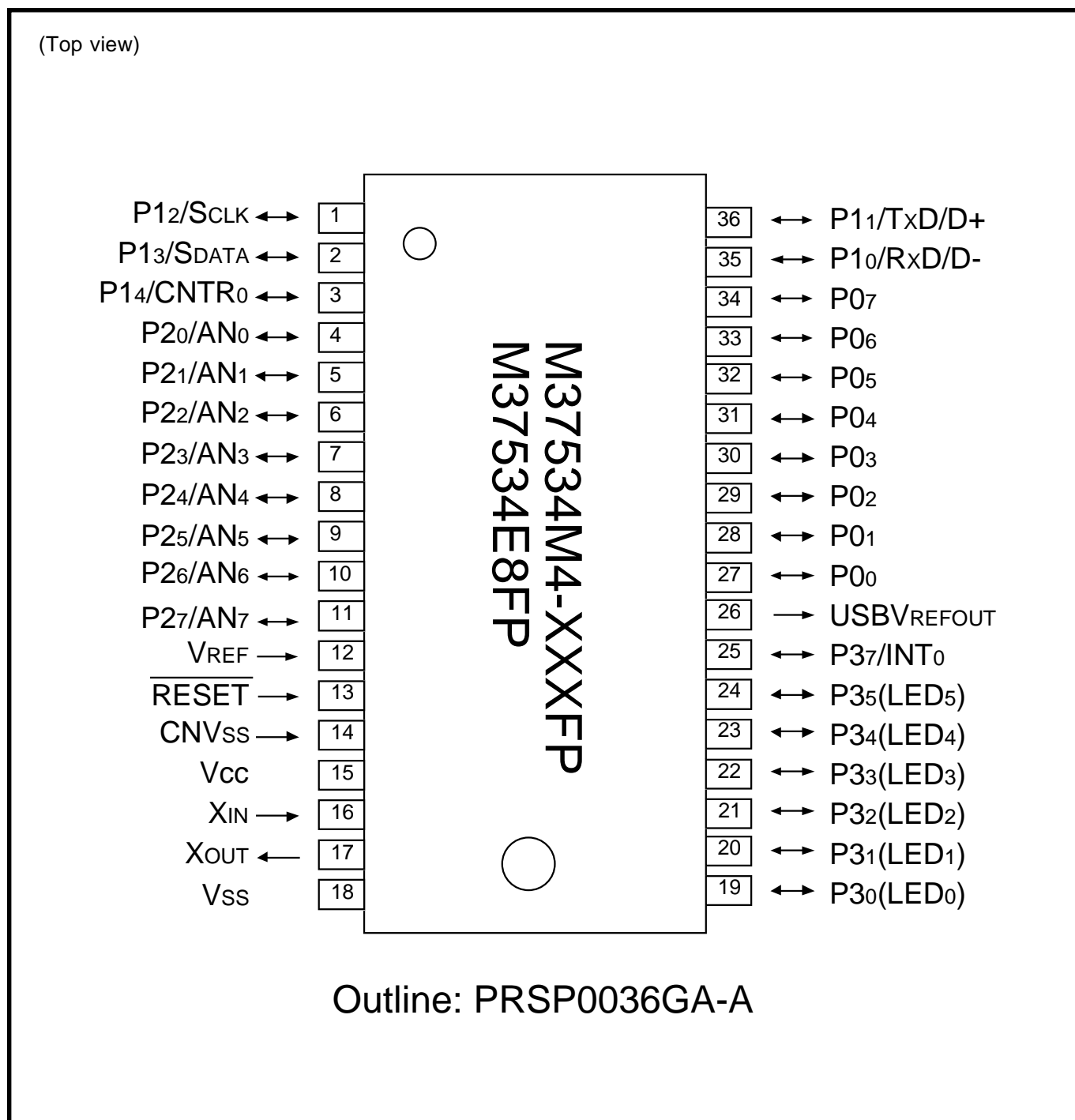


Fig. 3.10.1 M37534M4-XXXXFP, M37534E8FP pin configuration

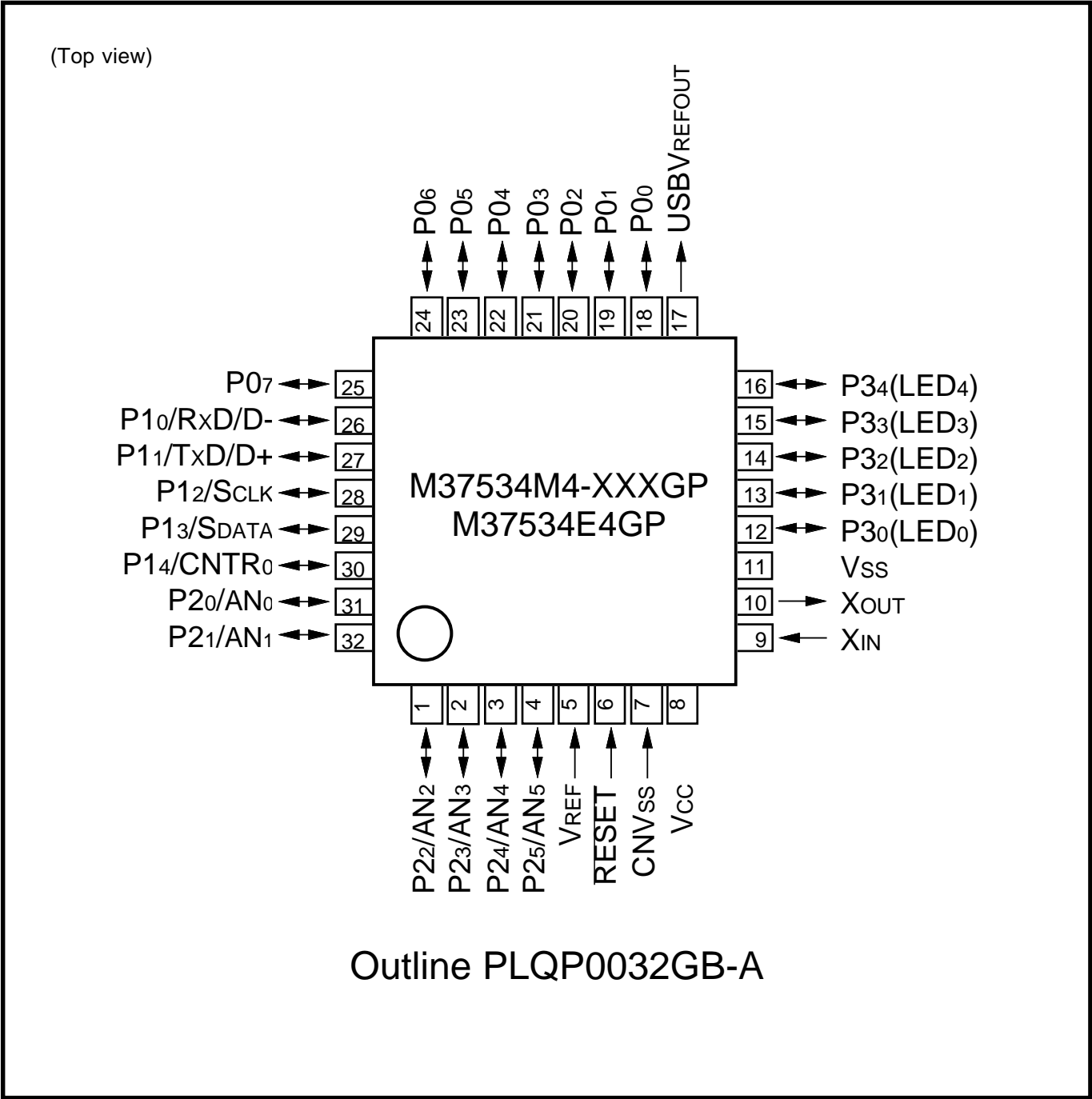


Fig. 3.10.2 M37534M4-XXXGP, M37534E4GP pin configuration

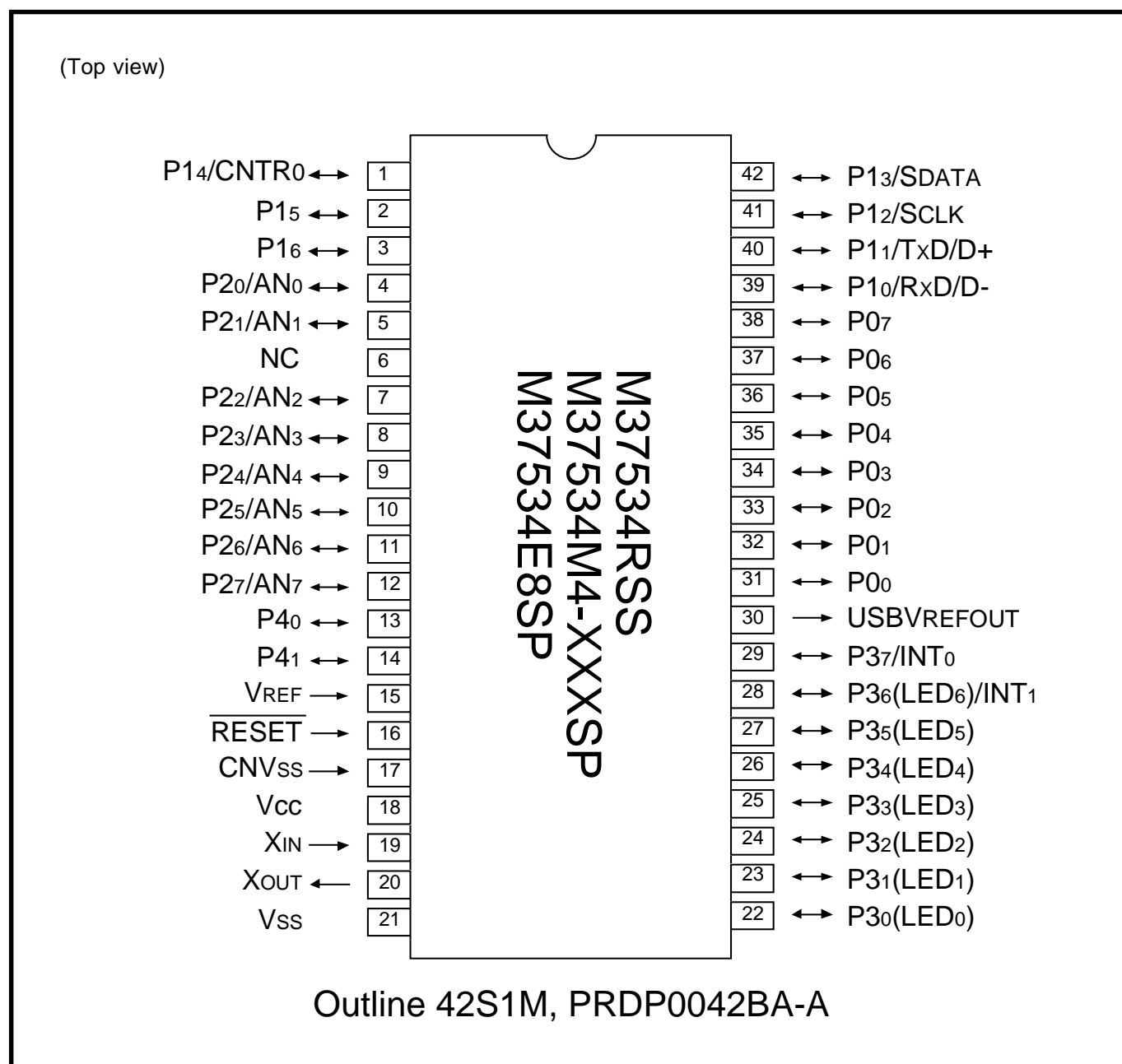


Fig. 3.10.3 M37534M4-XXXSP, M37534E8SP, M37534RSS pin configuration

REVISION HISTORY

7534 Group User's Manual

Rev.	Date	Description	
		Page	Summary
1.00	Nov 24, 2000	–	First edition
1.10	Sep 15, 2001	4 5 1-8 1-9 1-13 1-19 1-42 1-43	Preface: Home page address revised. BEFORE USING THIS MANUAL: Home page address revised. Table 1 Function description of Vss, Vcc revised. Fig. 7 "Under development" eliminated. "Note on stack page" added. Fig. 17 Ports P36, P37 revised. NOTES ON PROGRAMMING "Note on Stack Page" added. Table 8 32P6U-A added. Name of Programming Adapter revised.
2.00	Jun. 21, 2004	All pages 1-42 1-43 3-26 3-53	Words standardized: On-chip oscillator, A/D converter Electric Characteristic Difference Among Mask ROM and One Time PROM Version MCUs added. Note on Power Source Voltage added. DATA REQUIRED FOR MASK ORDERS revised. 3.3.12 Electric Characteristic Difference Among Mask ROM and One Time PROM Version MCUs, 3.3.13 Note on Power Source Voltage added. 32P6U-A revised.
3.00	Oct. 23, 2006	All pages Chapter 1 40 41 42 43 Chapter 2 50 69 77	Package names "36P2R-A" → "PRSP0036GA-A" revised Package names "32P6U-A" → "PLQP0032GB-A" revised Package names "42P4B" → "PRDP0042BA-A" revised "USB Spec. Rev.1.1" → "Low-Speed USB2.0 specification" revised Clock Generating Circuit; "No external resistor is needed resistor exists on- chip." → "No external resistor is needed depending on conditions.) Fig. 45; Pulled up added, NOTE added Fig. 48; NOTE 2 added NOTES ON PROGRAMMING; Watchdog Timer added NOTES ON USE; USB Communication added NOTES ON USE; Note on A/D Converter added 2.4.1 Transfer type: "Hi-Speed function: H.S." added 2.4.5; USB Communication added 2.5.4 (3) Method to stabilize A/D Converter added

REVISION HISTORY

7534 Group User's Manual

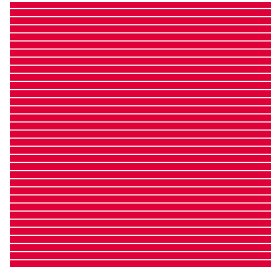
Rev.	Date	Description	
		Page	Summary
3.00	Oct. 23, 2006	Chapter 3	
		19	3.3.3 (3) Method to stabilize A/D Converter
		20	3.3.4 Notes on watchdog timer
		26	3.3.15 USB Communication added
		53	3.6 Package outline revised

**RENESAS 8-BIT SINGLE-CHIP MICROCOMPUTER
USER'S MANUAL
7534 Group**

**Publication Data : Rev.1.00 Nov 24, 2000
Rev.3.00 Oct 23, 2006**

**Published by : Sales Strategic Planning Div.
Renesas Technology Corp.**

7534 Group User's Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan