To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

8

RENESAS

7534 Group

User's Manual

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

Renesas Electronics
www.renesas.com

Rev.3.00 2006.10

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 life
   Renesas 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

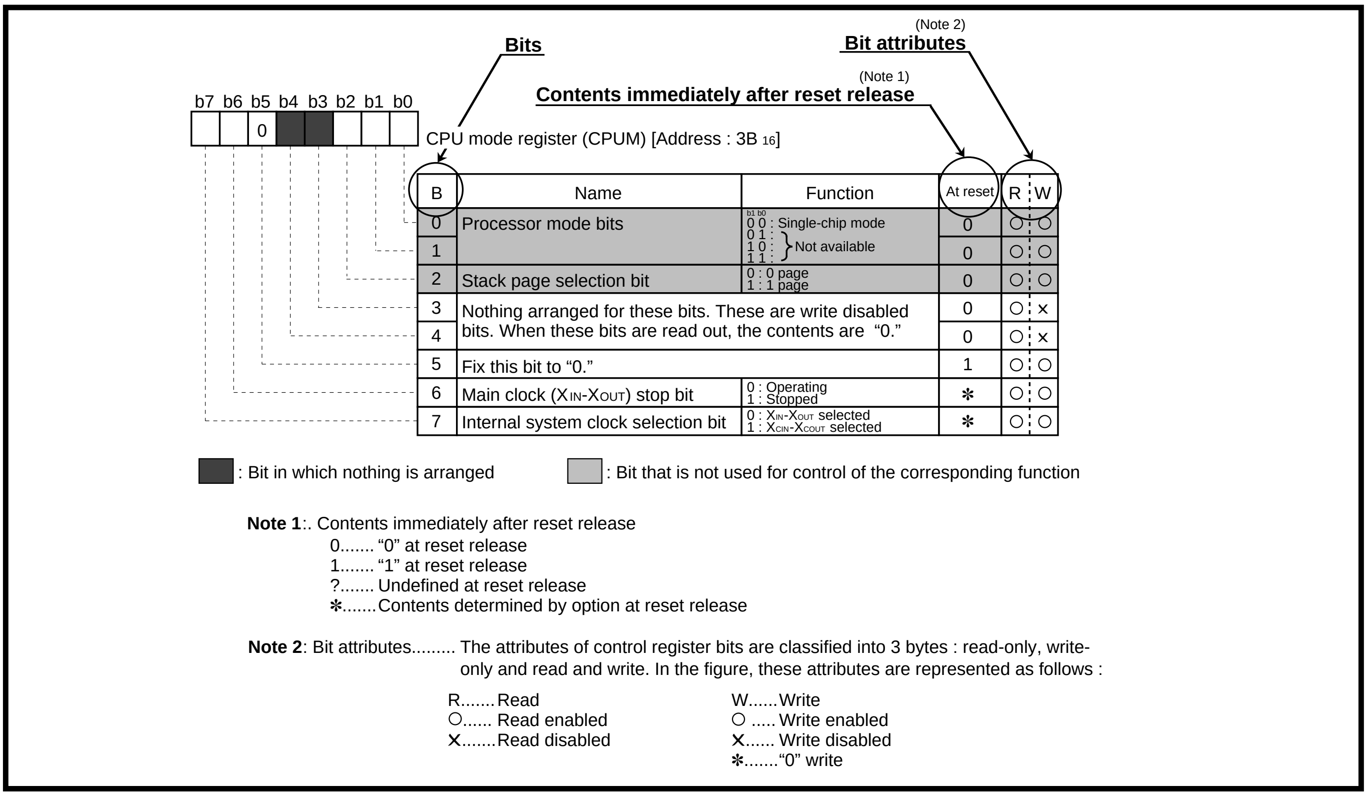
- **CHAPTER 1 HARDWARE**
  This chapter describes features of the microcomputer and operation of each peripheral function.

- **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 :



CPU mode register (CPUM) [Address : 3B $_{16}$]

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

■ : Bit in which nothing is arranged

▨ : Bit that is not used for control of the corresponding function

Note 1:. Contents immediately after reset release
0....... "0" at reset release
1....... "1" at reset release
?....... Undefined at reset release
✻.......Contents determined by option at reset release

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

R.......Read	W......Write
○...... Read enabled	○ .....Write enabled
×.......Read disabled	×...... Write disabled
	✻......."0" write

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

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

List of figures

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

List of tables

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

CHAPTER 1

HARDWARE

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 × 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 × 1
  (Clock-synchronized)
- A/D converter .... 10-bit × 8 channels
- Clock generating circuit .... Built-in type
  (connect to external ceramic resonator or quartz-crystal oscillator)
- Watchdog timer .... 16-bit × 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

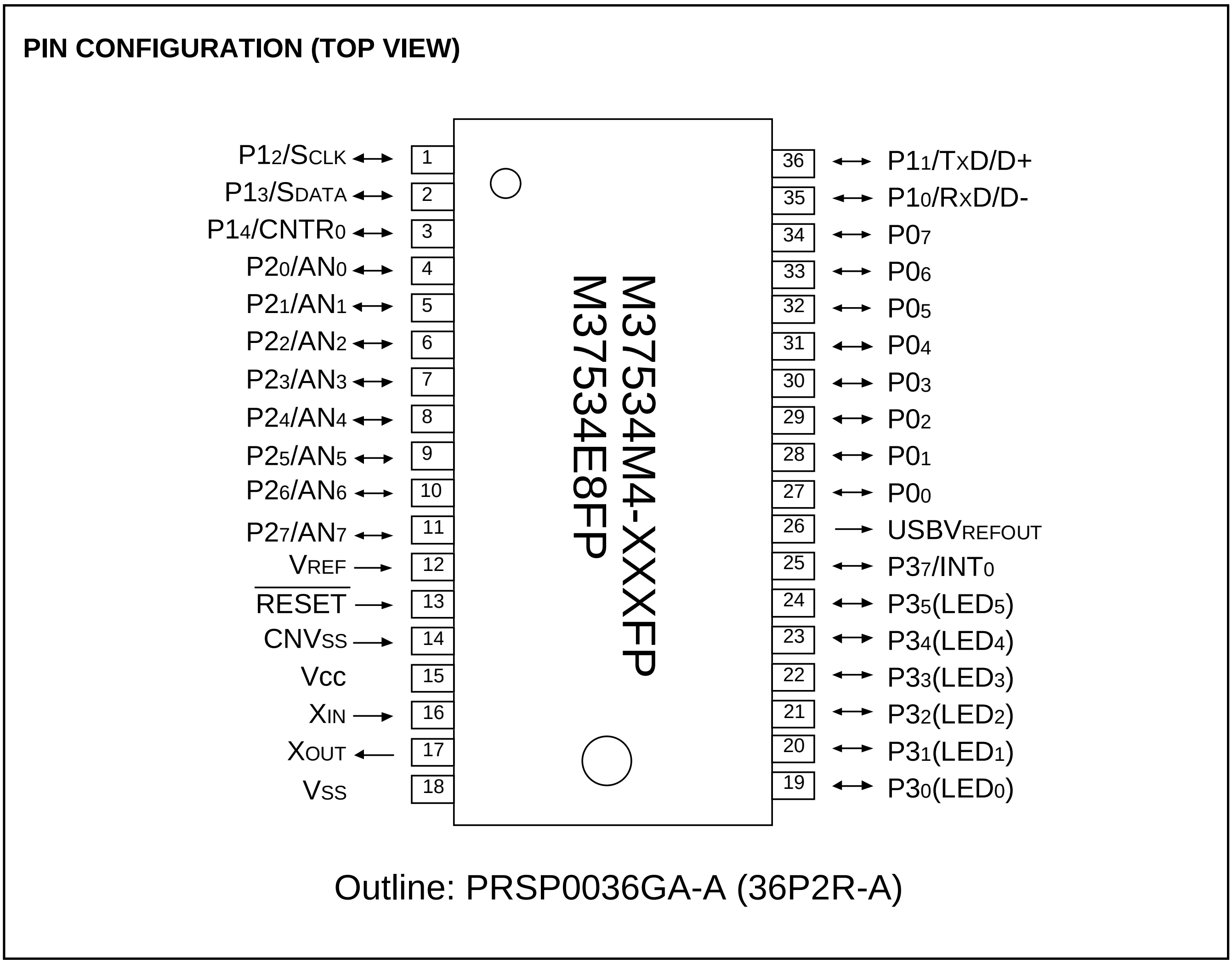


Fig. 1 Pin configuration of M37534M4-XXXFP, M37534E8FP

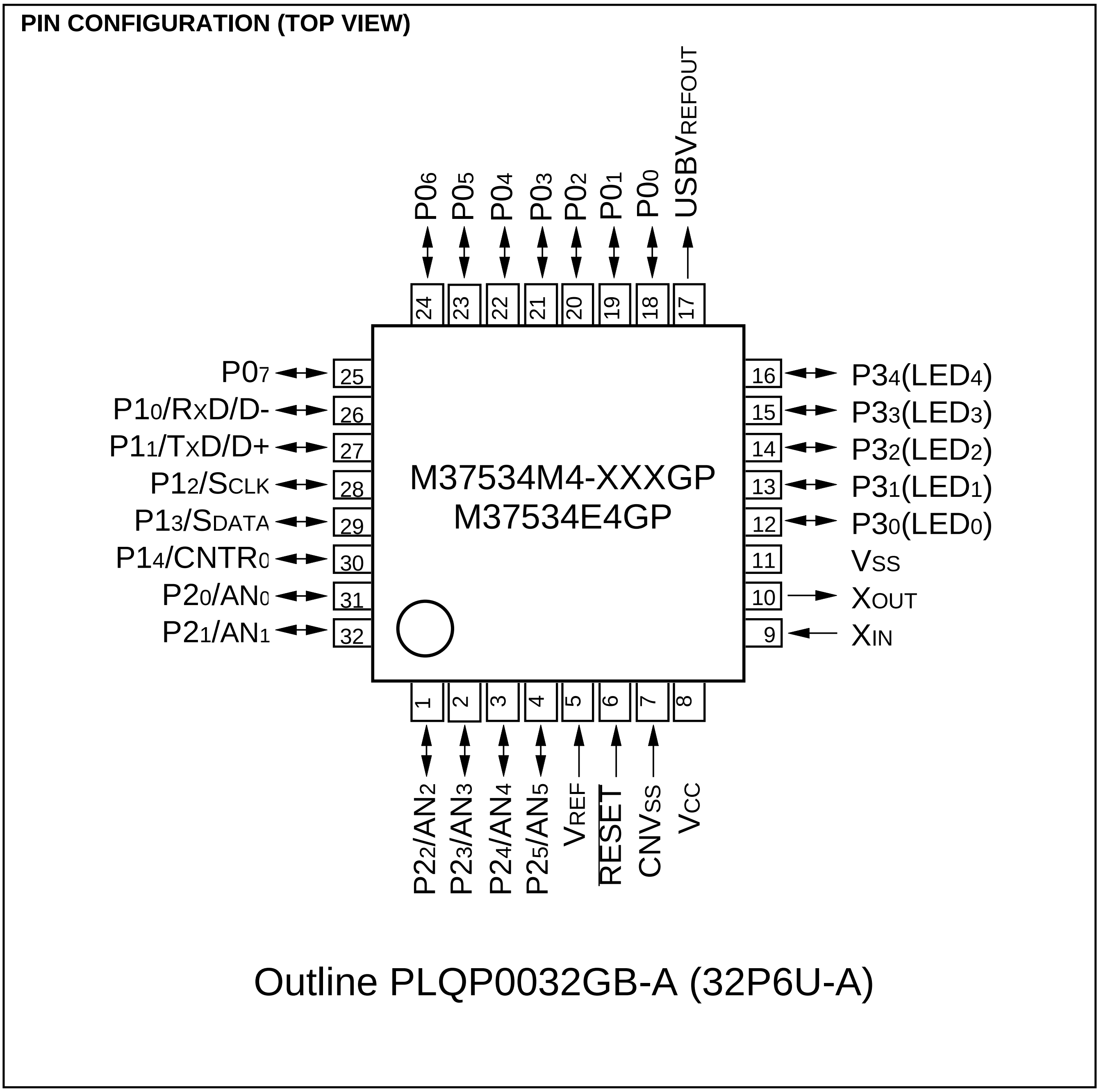


Fig. 2 Pin configuration of M37534M4-XXXGP, M37534E4GP

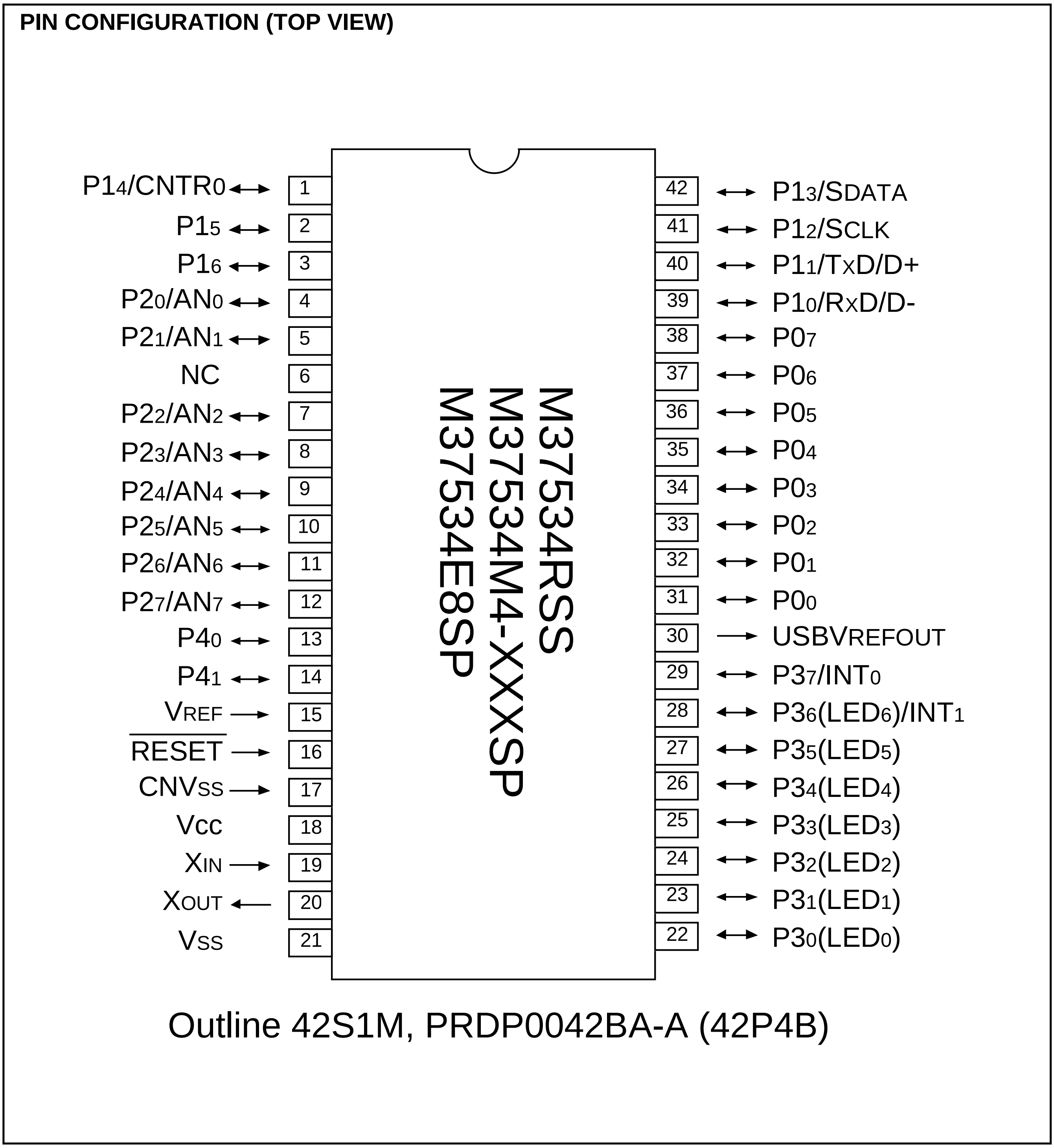


Fig. 3 Pin configuration of M37534RSS, M37534M4-XXXSP, M37534E8SP

FUNCTIONAL BLOCK

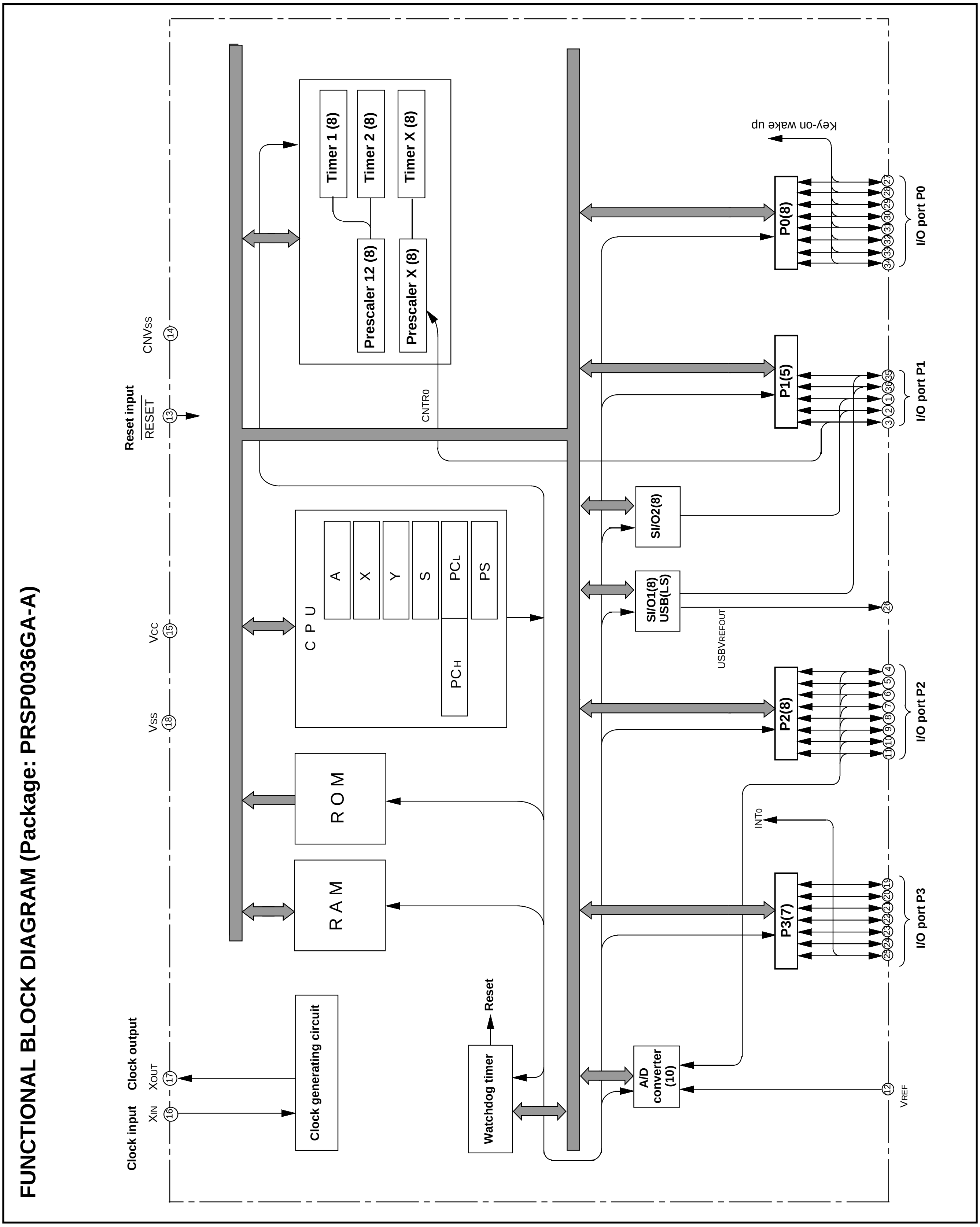


Fig. 4 Functional block diagram (PRSP0036GA-A package type)

FUNCTIONAL BLOCK DIAGRAM (Package: PLQP0032GB-A)

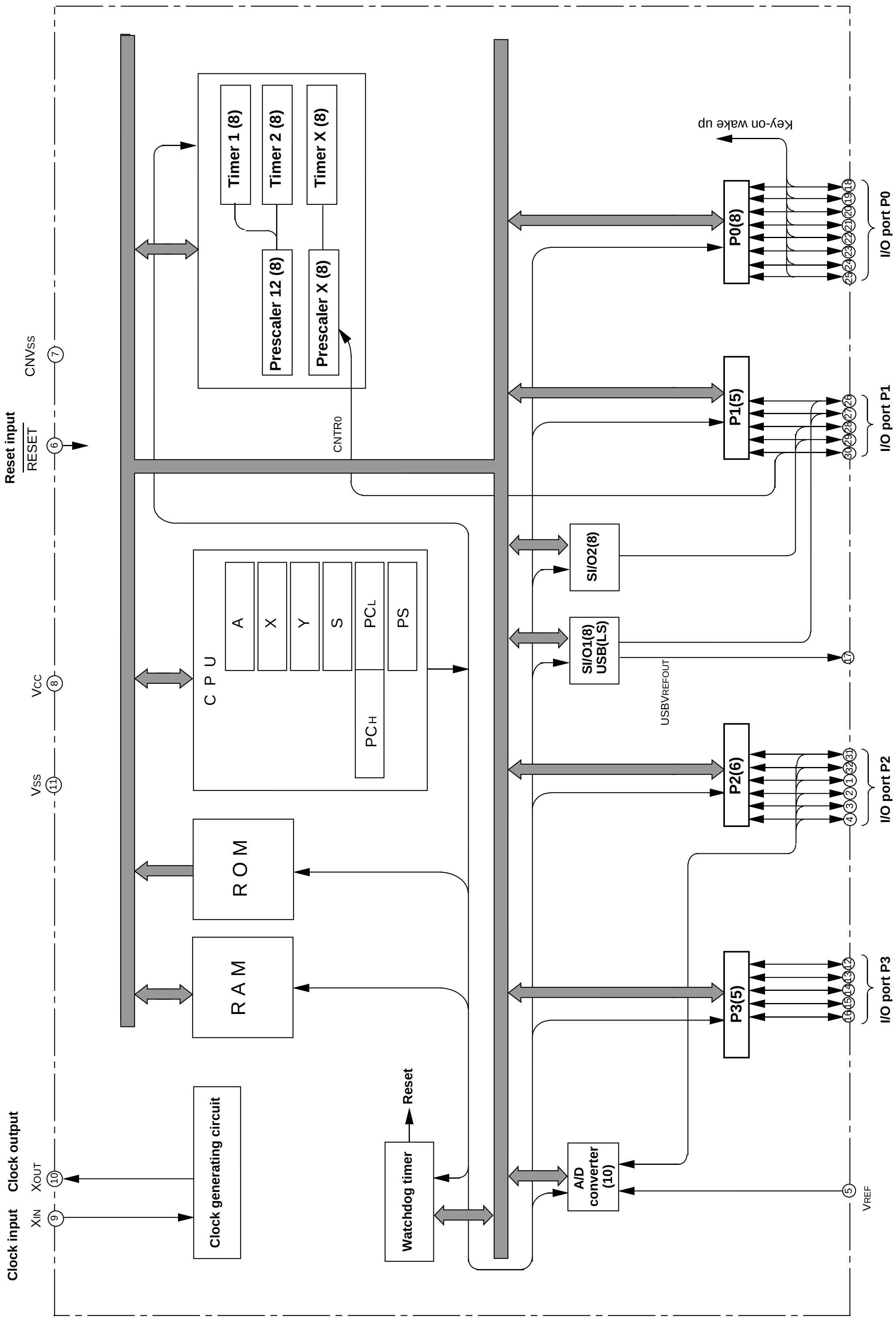


Fig. 5 Functional block diagram (PLQP0032GB-A package type)

FUNCTIONAL BLOCK DIAGRAM (Package: PRDP0042BA-A)

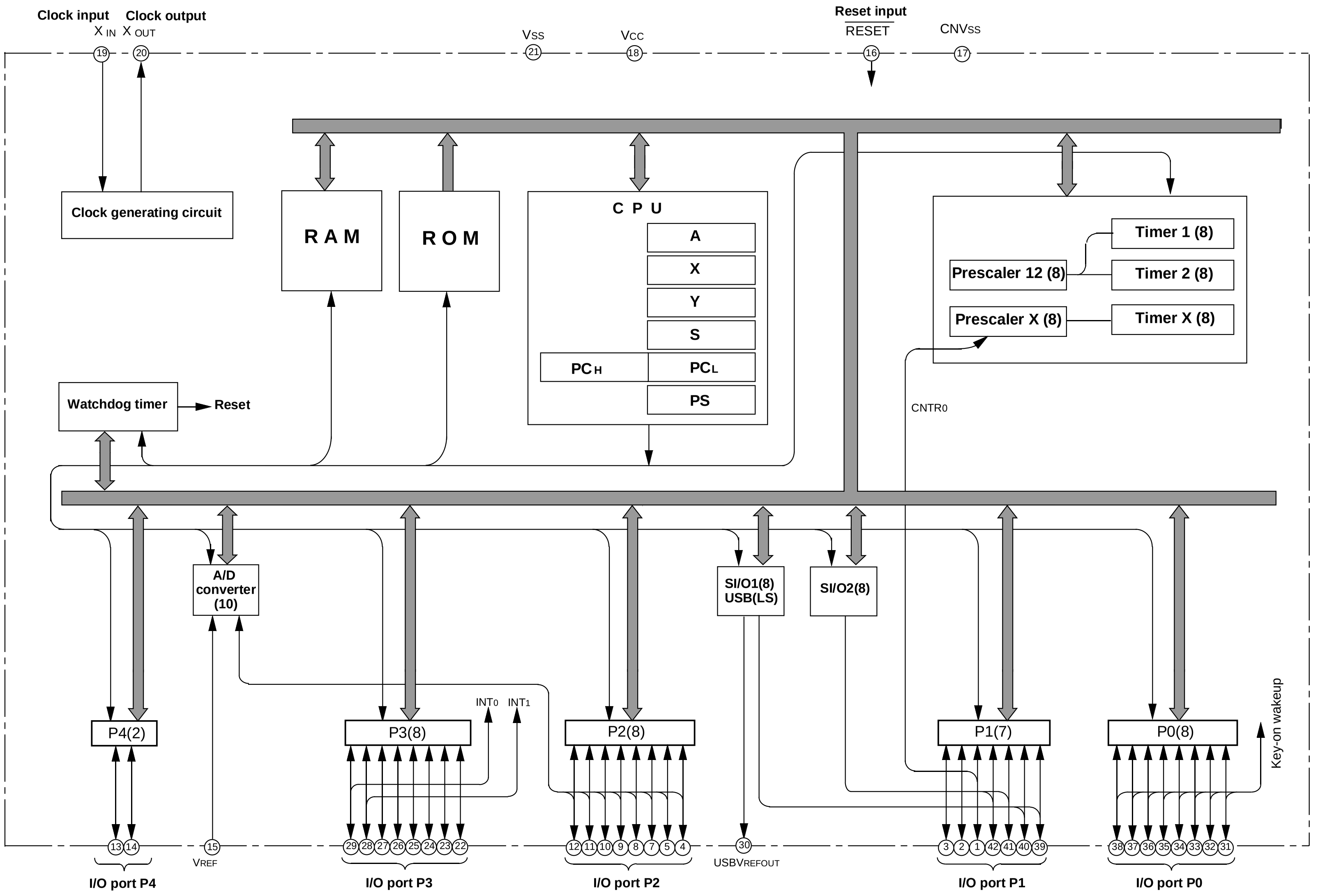


Fig. 6 Functional block diagram (PRDP0042BA-A package type)

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.	
$\overline{\text{RESET}}$	Reset input	•Reset input pin for active “L”	
XIN	Clock input	•Input and output pins for main clock generating circuit •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.	
XOUT	Clock output		
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 P13/SDATA			•Serial I/O2 function pin
P14/CNTR0			•Timer X function pin
P15, P16			
P20/AN0– P27/AN7	I/O port P2	•8-bit I/O port having almost the same function as P0 •CMOS 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.	
P36/INT1 P37/INT0		•Whether a built-in pull-up resistor is to be used or not can be determined by program.	•Interrupt input pins
P40, P41	I/O port P4	•2-bit I/O port •I/O direction register allows each pin to be individually programmed as either input or output.	

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

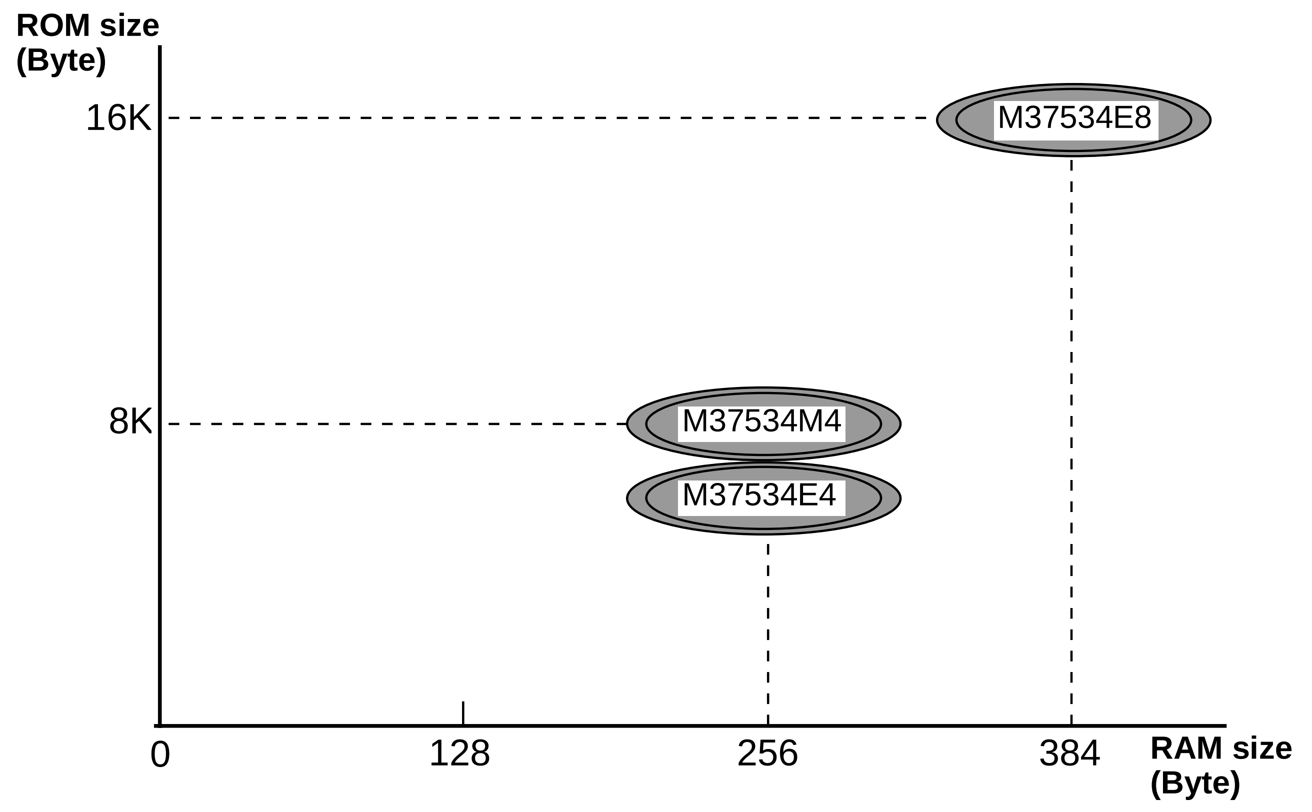


Fig. 7 Memory expansion plan

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.

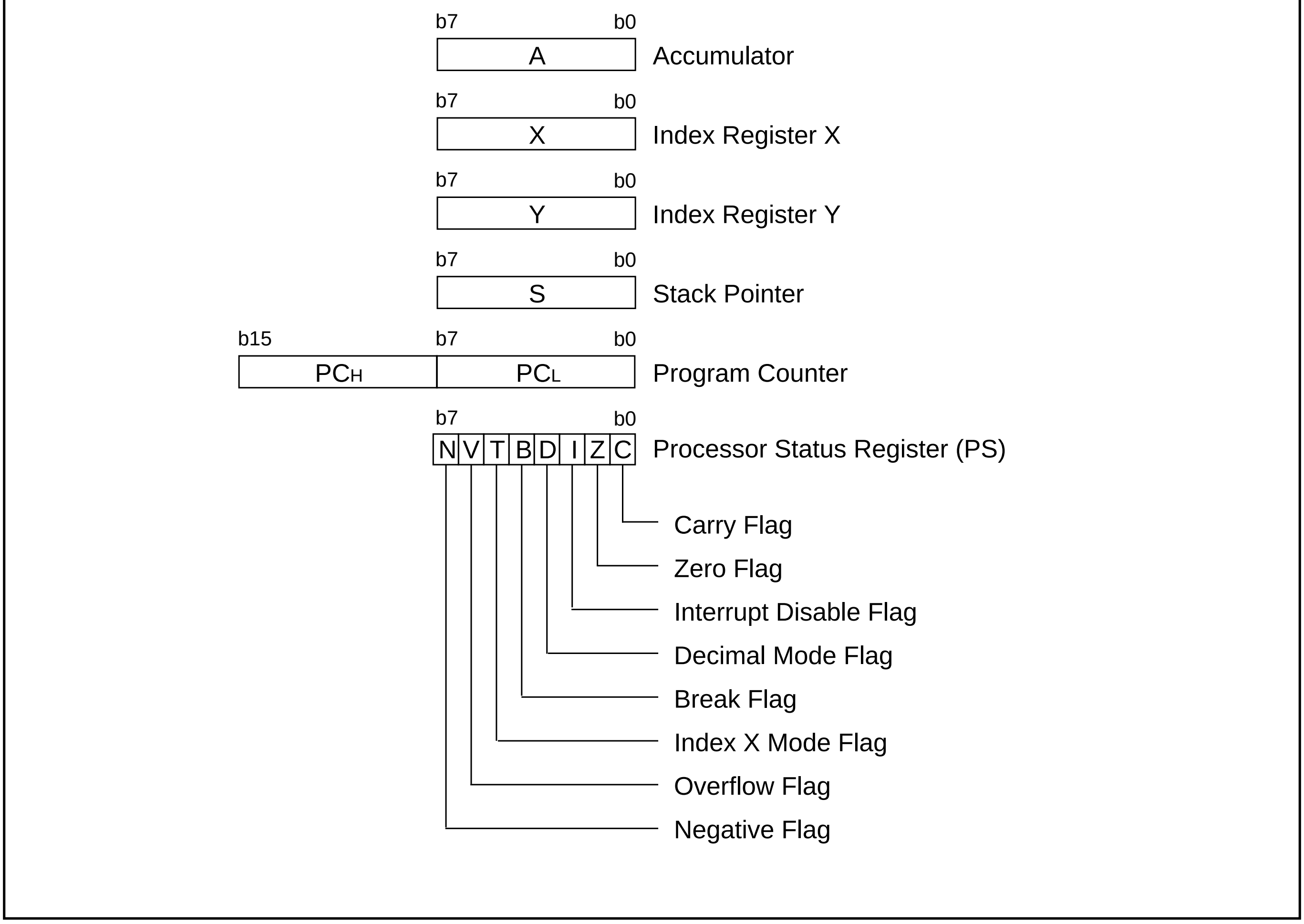
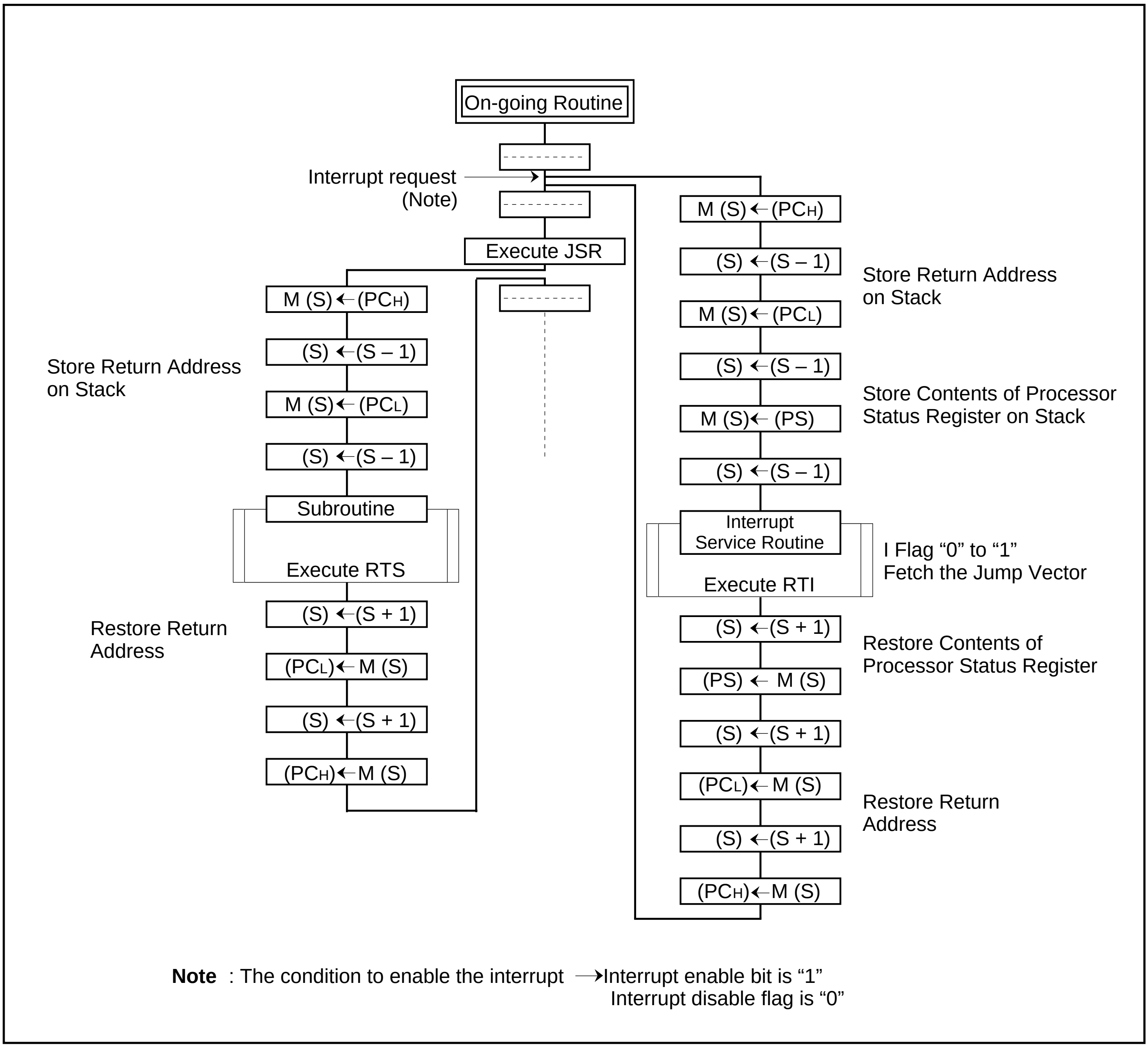


Fig. 8 740 Family CPU register structure



Note : The condition to enable the interrupt →Interrupt enable bit is "1"
Interrupt disable flag is "0"

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 $003B_{16}$.

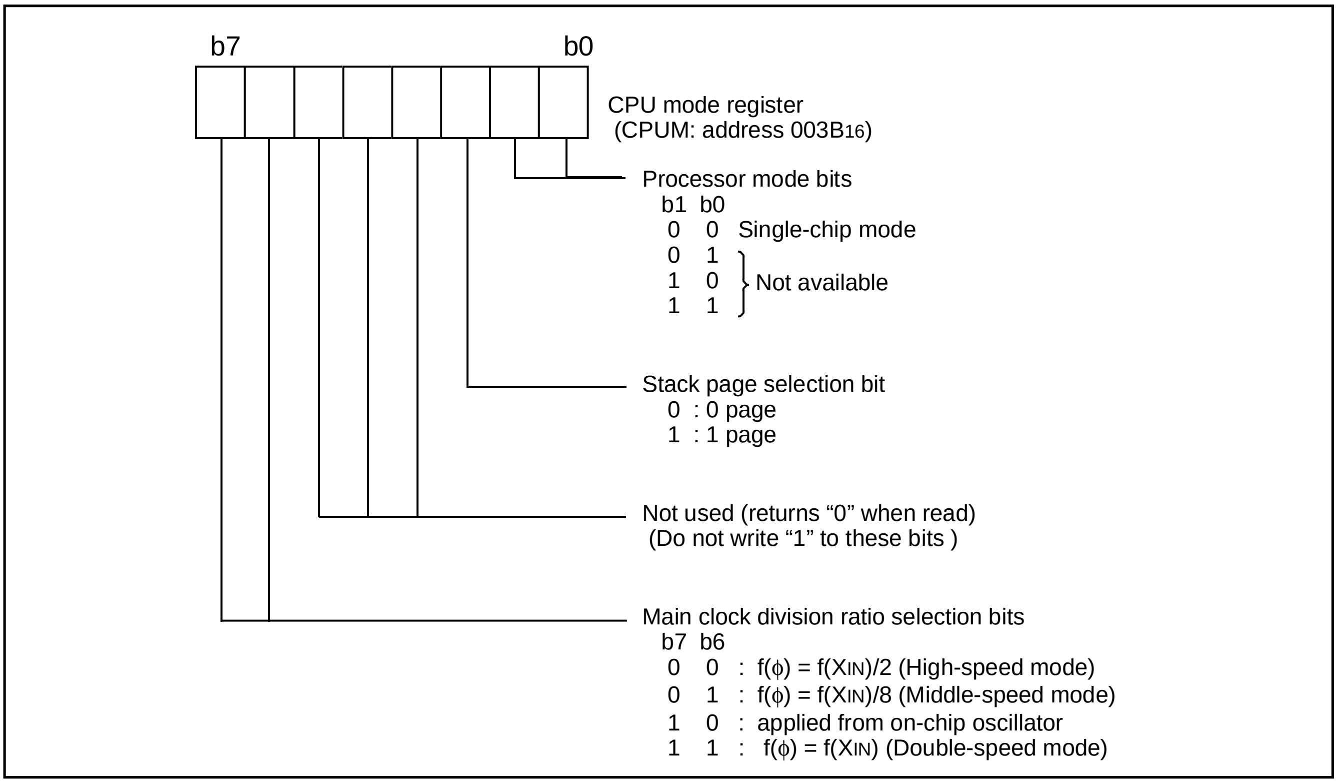

b7
b0
CPU mode register
(CPUM: address 003B16)
Processor mode bits
b1 b0
0 0 Single-chip mode
0 1
1 0
1 1
Not available
Stack page selection bit
0 : 0 page
1 : 1 page
Not used (returns "0" when read)
(Do not write "1" to these bits)
Main clock division ratio selection bits
b7 b6
0 0 : f(φ) = f(XIN)/2 (High-speed mode)
0 1 : f(φ) = f(XIN)/8 (Middle-speed mode)
1 0 : applied from on-chip oscillator
1 1 : f(φ) = f(XIN) (Double-speed mode)

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.

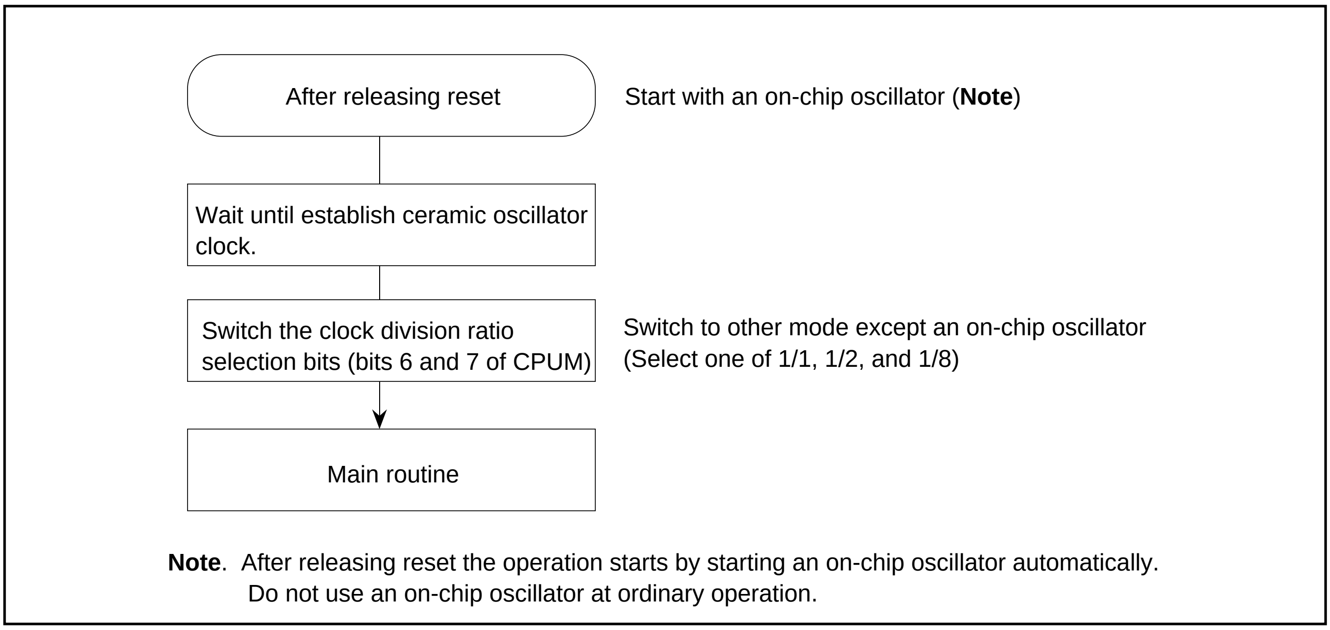

After releasing reset
Start with an on-chip oscillator (Note)
Wait until establish ceramic oscillator clock.
Switch the clock division ratio selection bits (bits 6 and 7 of CPUM)
Switch to other mode except an on-chip oscillator
(Select one of 1/1, 1/2, and 1/8)
Main routine
Note. After releasing reset the operation starts by starting an on-chip oscillator automatically.
Do not use an on-chip oscillator at ordinary operation.

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

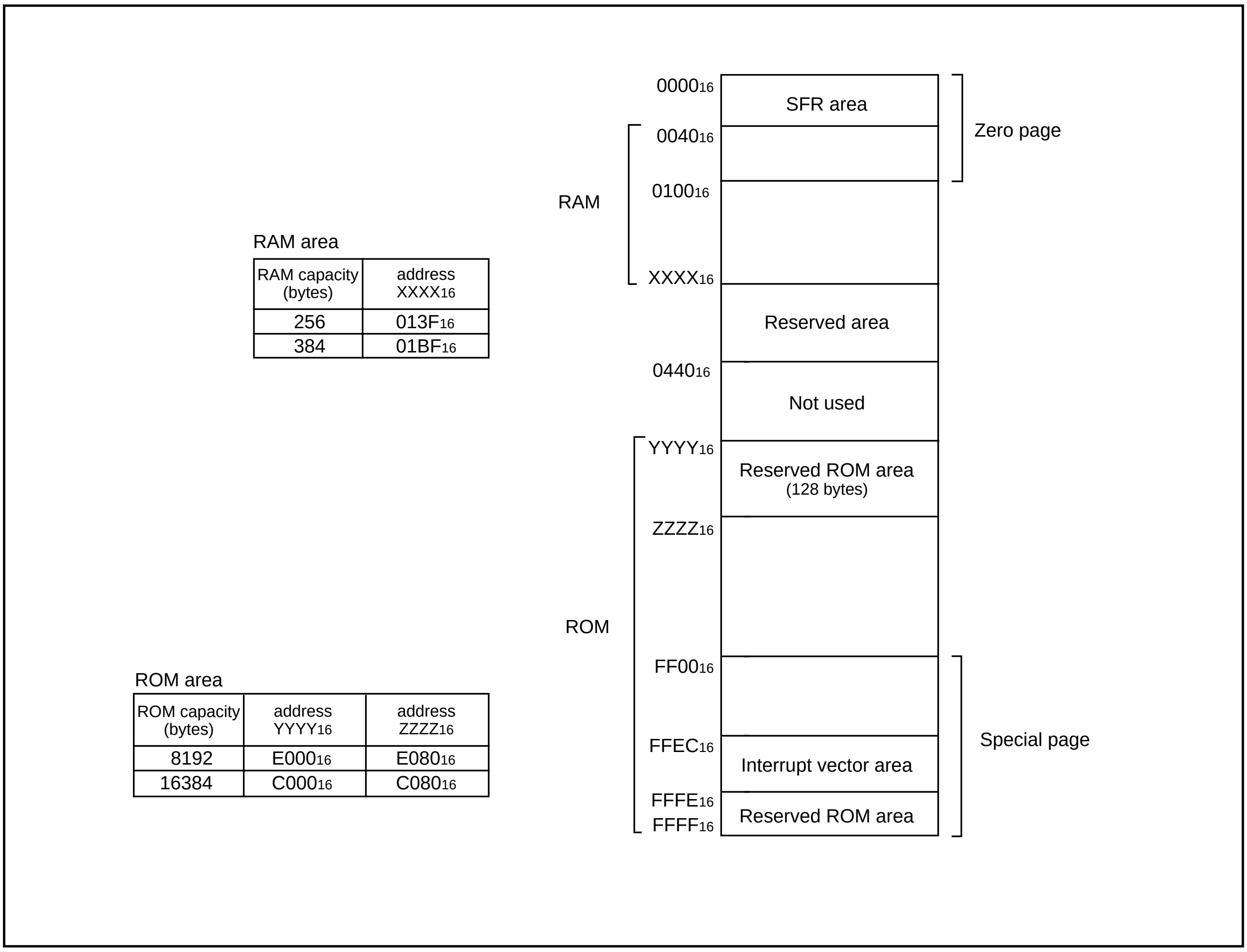
The interrupt vector area contains reset and interrupt vectors.

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.



RAM area

RAM capacity (bytes)	address $XXXX_{16}$
256	$013F_{16}$
384	$01BF_{16}$

ROM area

ROM capacity (bytes)	address $YYYY_{16}$	address $ZZZZ_{16}$
8192	$E000_{16}$	$E080_{16}$
16384	$C000_{16}$	$C080_{16}$

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 (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)	$003F_{16}$	

Fig. 13 Memory map of special function register (SFR)

I/O Ports

[Direction registers] PiD

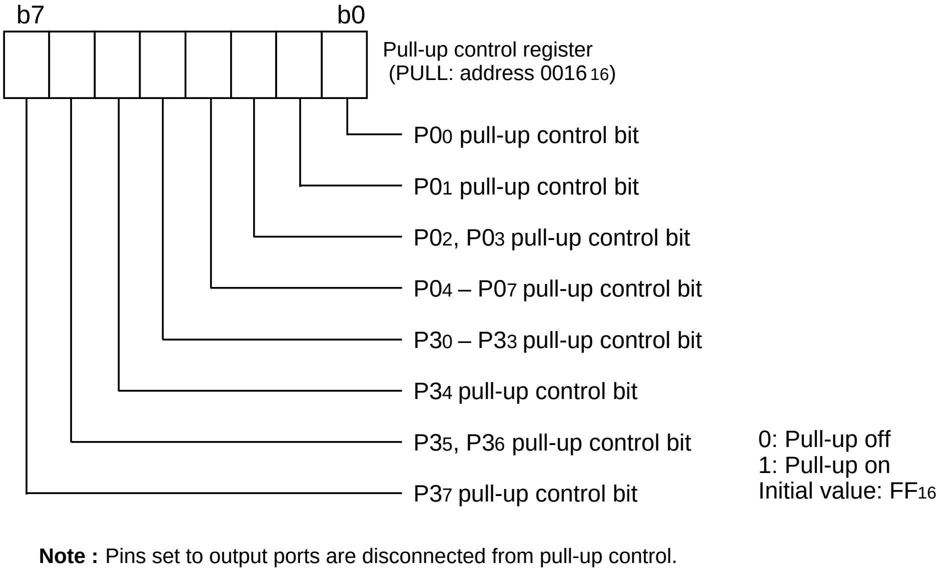
The I/O ports have direction registers which determine the input/output direction of each pin. Each bit in a direction register corresponds to individual pin, and each pin can be set to be input or output.
When "1" is set to the bit corresponding to a pin, this pin becomes an output port. When "0" is 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 0016_{16}), ports P0 and P3 can exert pull-up control by program. However, pins set to output are disconnected from this control and cannot exert pull-up control.

[Port P1P3 control] P1P3C

By setting the port P1P3 control register (address 0017_{16}), a CMOS input level or a TTL input level can be selected for ports $P1_0$, $P1_2$, $P1_3$, $P3_6$ and $P3_7$ 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.



Note : Pins set to output ports are disconnected from pull-up control.

Fig. 14 Structure of pull-up control register

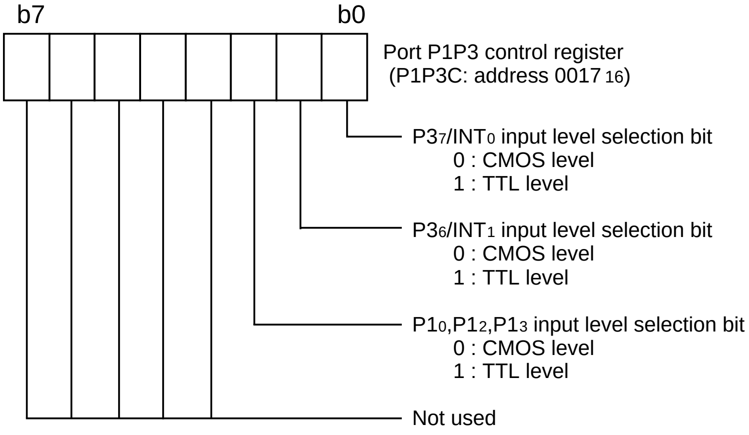
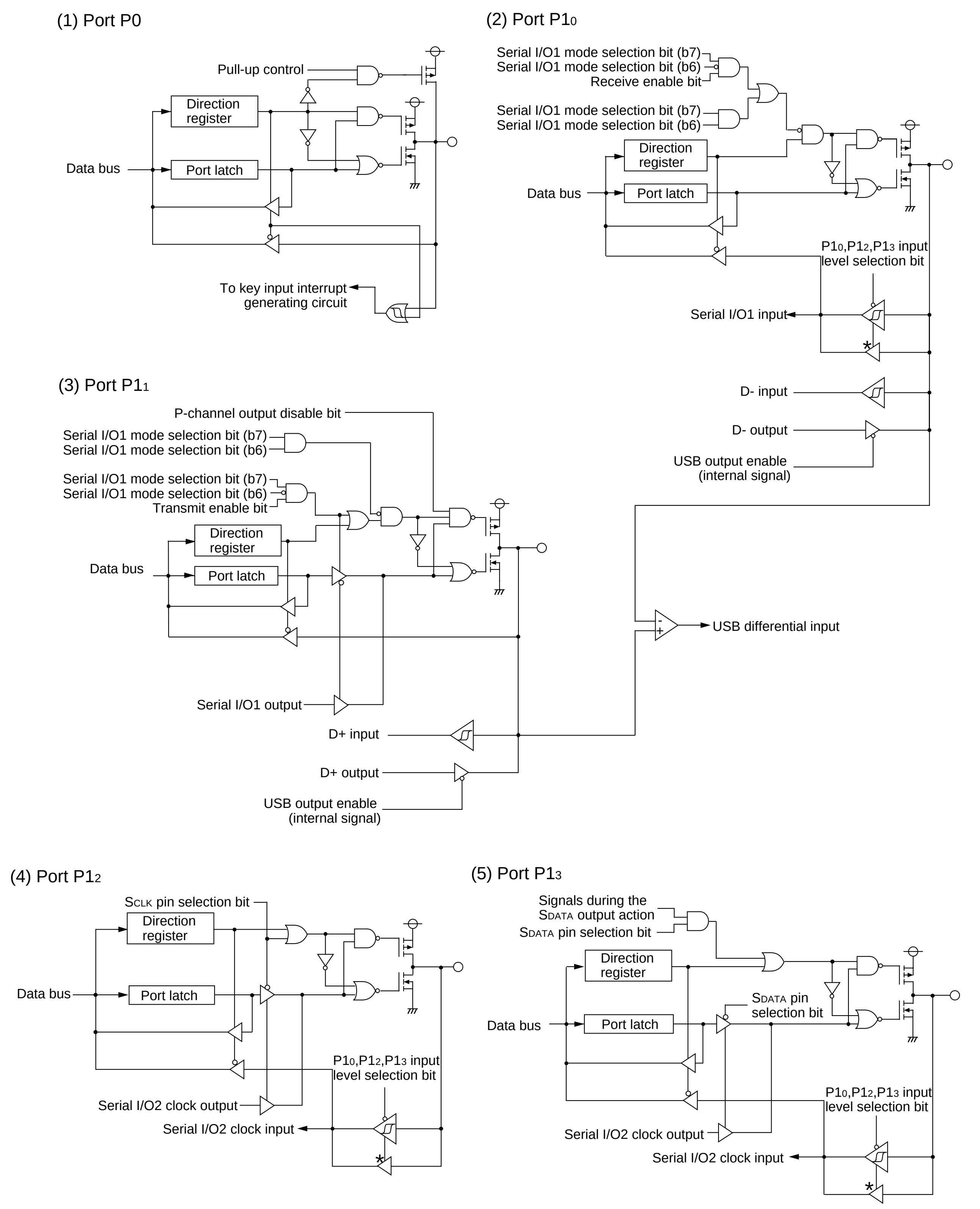

b7
b0
Port P1P3 control register
(P1P3C: address 0017 16)
P37/INT0 input level selection bit
0 : CMOS level
1 : TTL level
P36/INT1 input level selection bit
0 : CMOS level
1 : TTL level
P10,P12,P13 input level selection bit
0 : CMOS level
1 : TTL level
Not used

Fig. 15 Structure of port P1P3 control register

Table 5 I/O port function table

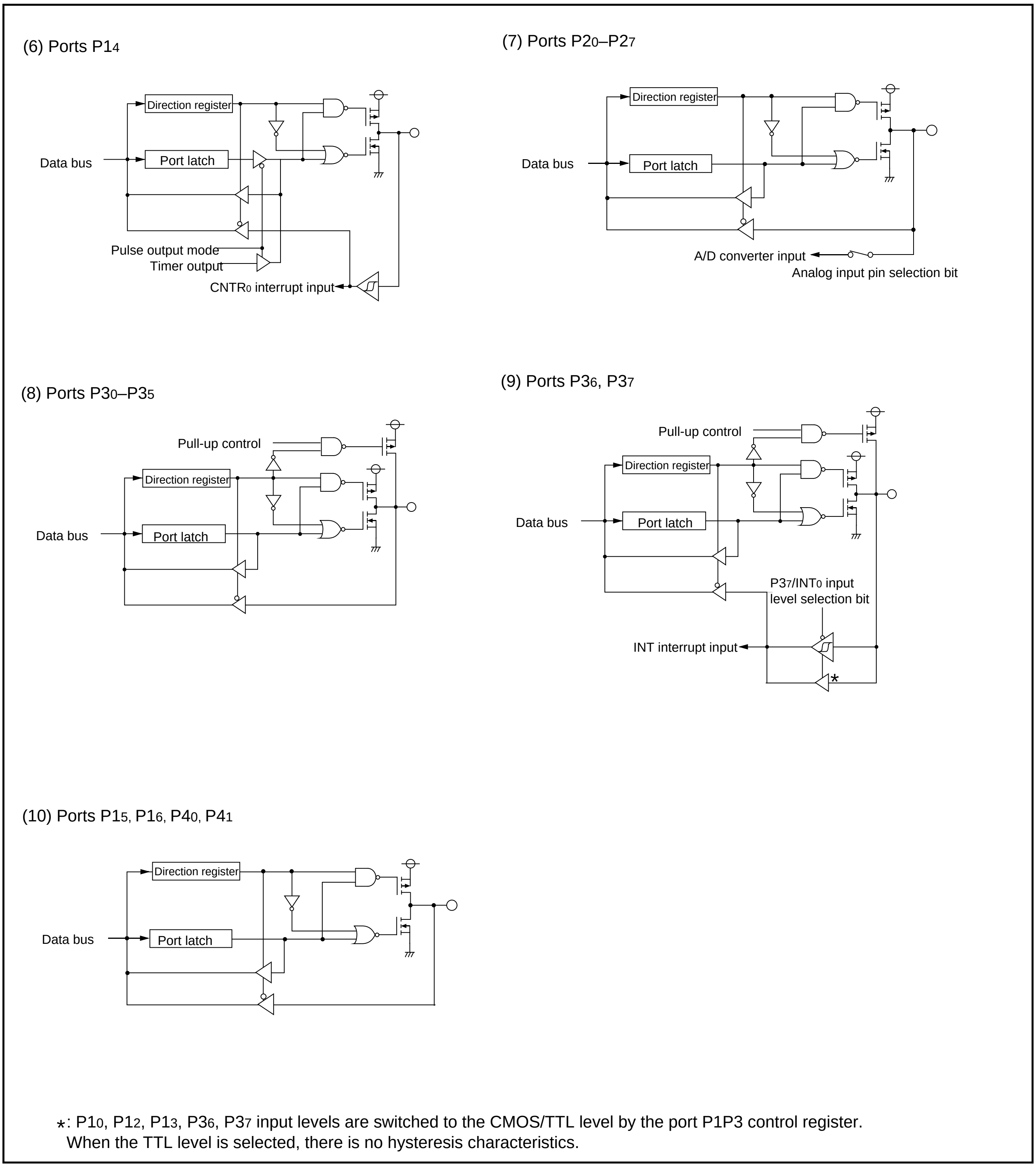
Pin	Name	Input/output	I/O format	Non-port function	Related SFRs	Diagram No.
$P0_0$–$P0_7$	Port P0	I/O individual bits	•CMOS compatible input level •CMOS 3-state output	Key input interrupt	Pull-up control register	(1)
$P1_0$/RxD/D- $P1_1$/TxD/D+	Port P1		•USB input/output level when selecting USB function	Serial I/O1 function input/output	Serial I/O1 control register	(2) (3)
$P1_2$/SCLK $P1_3$/SDATA			•CMOS compatible input level •CMOS 3-state output (Note)	Serial I/O2 function input/output	Serial I/O2 control register	(4) (5)
$P1_4$/$CNTR_0$				Timer X function input/output	Timer X mode register	(6)
$P1_5$, $P1_6$						(10)
$P2_0$/AN_0– $P2_7$/AN_7	Port P2			A/D conversion input	A/D control register	(7)
$P3_0$–$P3_5$	Port P3					(8)
$P3_6$/INT_1 $P3_7$/INT_0				External interrupt input	Interrupt edge selection register	(9)
$P4_0$, $P4_1$	Port P4					(10)

Note: Port $P1_0$, $P1_2$, $P1_3$, $P3_6$, $P3_7$ is CMOS/TTL input level.


(1) Port P0
Pull-up control
Direction register
Data bus
Port latch
To key input interrupt generating circuit
(2) Port P1$_0$
Serial I/O1 mode selection bit (b7)
Serial I/O1 mode selection bit (b6)
Receive enable bit
Serial I/O1 mode selection bit (b7)
Serial I/O1 mode selection bit (b6)
Direction register
Data bus
Port latch
P1$_0$,P1$_2$,P1$_3$ input level selection bit
Serial I/O1 input
*
D- input
D- output
USB output enable (internal signal)
USB differential input
-
+
(3) Port P1$_1$
P-channel output disable bit
Serial I/O1 mode selection bit (b7)
Serial I/O1 mode selection bit (b6)
Serial I/O1 mode selection bit (b7)
Serial I/O1 mode selection bit (b6)
Transmit enable bit
Direction register
Data bus
Port latch
Serial I/O1 output
D+ input
D+ output
USB output enable (internal signal)
(4) Port P1$_2$
SCLK pin selection bit
Direction register
Data bus
Port latch
P1$_0$,P1$_2$,P1$_3$ input level selection bit
Serial I/O2 clock output
Serial I/O2 clock input
*
(5) Port P1$_3$
Signals during the SDATA output action
SDATA pin selection bit
Direction register
Data bus
Port latch
SDATA pin selection bit
P1$_0$,P1$_2$,P1$_3$ input level selection bit
Serial I/O2 clock output
Serial I/O2 clock input
*

* : P1$_0$, P1$_2$, P1$_3$, P3$_6$, P3$_7$ input levels are switched to the CMOS/TTL level by the port P1P3 control register.
When the TTL level is selected, there is no hysteresis characteristics.

Fig. 16 Block diagram of ports (1)



(6) Ports $P1_4$

(7) Ports $P2_0$–$P2_7$

(8) Ports $P3_0$–$P3_5$

(9) Ports $P3_6$, $P3_7$

(10) Ports $P1_5$, $P1_6$, $P4_0$, $P4_1$

*: $P1_0$, $P1_2$, $P1_3$, $P3_6$, $P3_7$ input levels are switched to the CMOS/TTL level by the port P1P3 control register.
When the TTL level is selected, there is no hysteresis characteristics.

Fig. 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 CNTR$_0$ 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 INT$_1$ 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 (INT$_0$, INT$_1$, CNTR$_0$) 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 CNTR$_0$: 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/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.

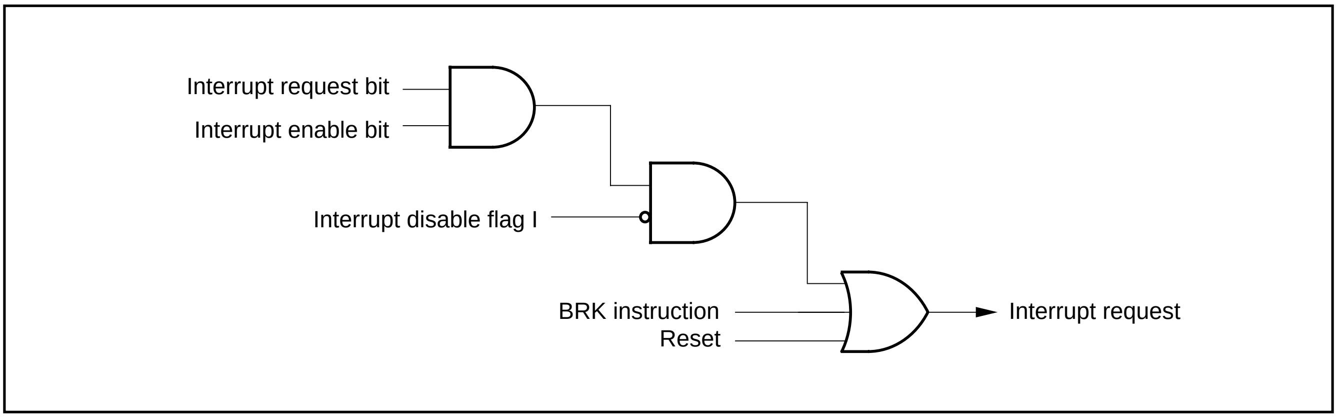
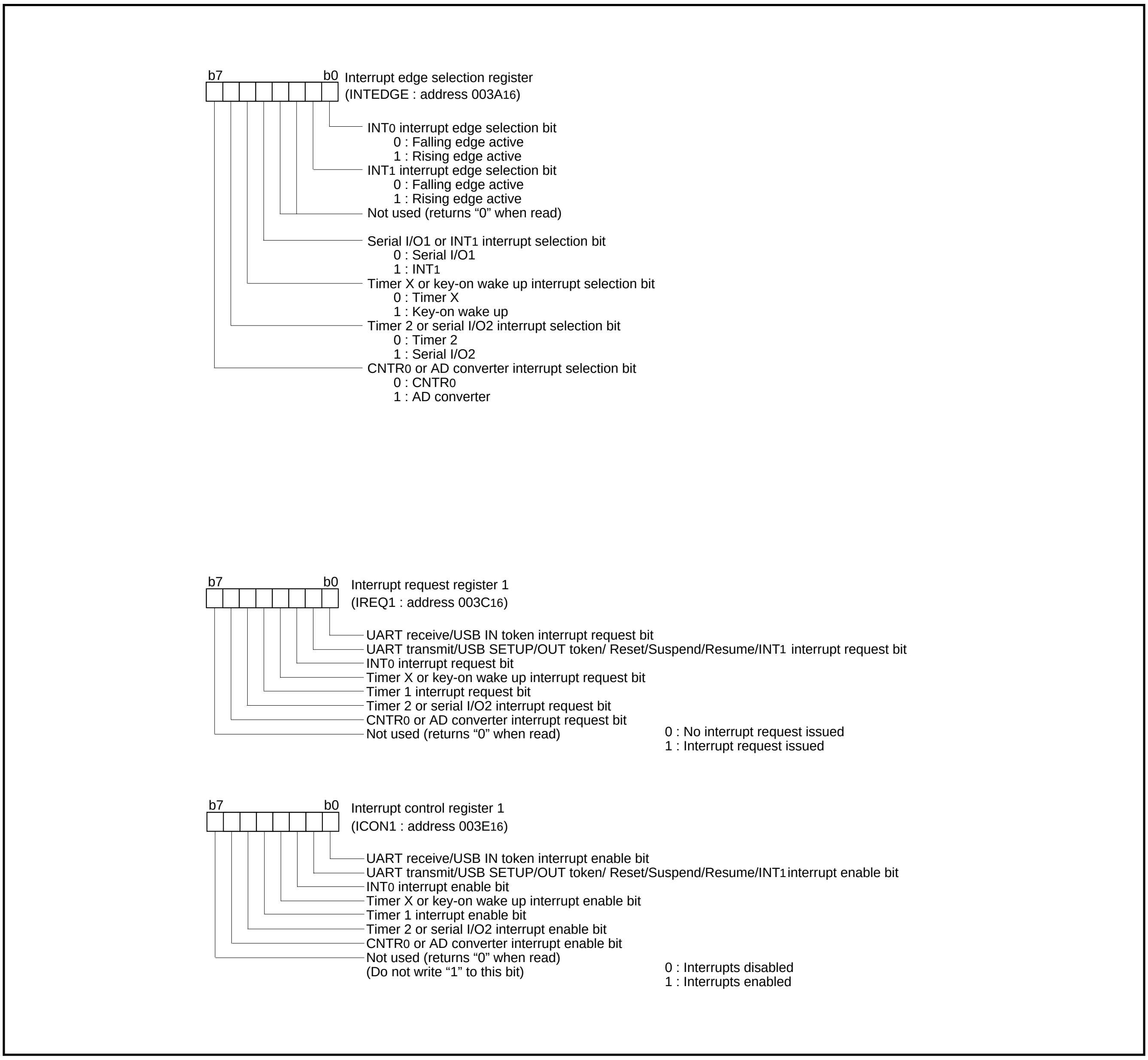


Fig. 18 Interrupt control



Interrupt edge selection register
(INTEDGE : address 003A$_{16}$)

INT$_0$ interrupt edge selection bit
0 : Falling edge active
1 : Rising edge active
INT$_1$ interrupt edge selection bit
0 : Falling edge active
1 : Rising edge active
Not used (returns “0” when read)
Serial I/O1 or INT$_1$ interrupt selection bit
0 : Serial I/O1
1 : INT$_1$
Timer X or key-on wake up interrupt selection bit
0 : Timer X
1 : Key-on wake up
Timer 2 or serial I/O2 interrupt selection bit
0 : Timer 2
1 : Serial I/O2
CNTR$_0$ or AD converter interrupt selection bit
0 : CNTR$_0$
1 : AD converter

Interrupt request register 1
(IREQ1 : address 003C$_{16}$)

UART receive/USB IN token interrupt request bit
UART transmit/USB SETUP/OUT token/ Reset/Suspend/Resume/INT$_1$ interrupt request bit
INT$_0$ interrupt request bit
Timer X or key-on wake up interrupt request bit
Timer 1 interrupt request bit
Timer 2 or serial I/O2 interrupt request bit
CNTR$_0$ or AD converter interrupt request bit
Not used (returns “0” when read)

0 : No interrupt request issued
1 : Interrupt request issued

Interrupt control register 1
(ICON1 : address 003E$_{16}$)

UART receive/USB IN token interrupt enable bit
UART transmit/USB SETUP/OUT token/ Reset/Suspend/Resume/INT$_1$interrupt enable bit
INT$_0$ interrupt enable bit
Timer X or key-on wake up interrupt enable bit
Timer 1 interrupt enable bit
Timer 2 or serial I/O2 interrupt enable bit
CNTR$_0$ or AD converter interrupt enable bit
Not used (returns “0” when read)
(Do not write “1” to this bit)

0 : Interrupts disabled
1 : Interrupts enabled

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.

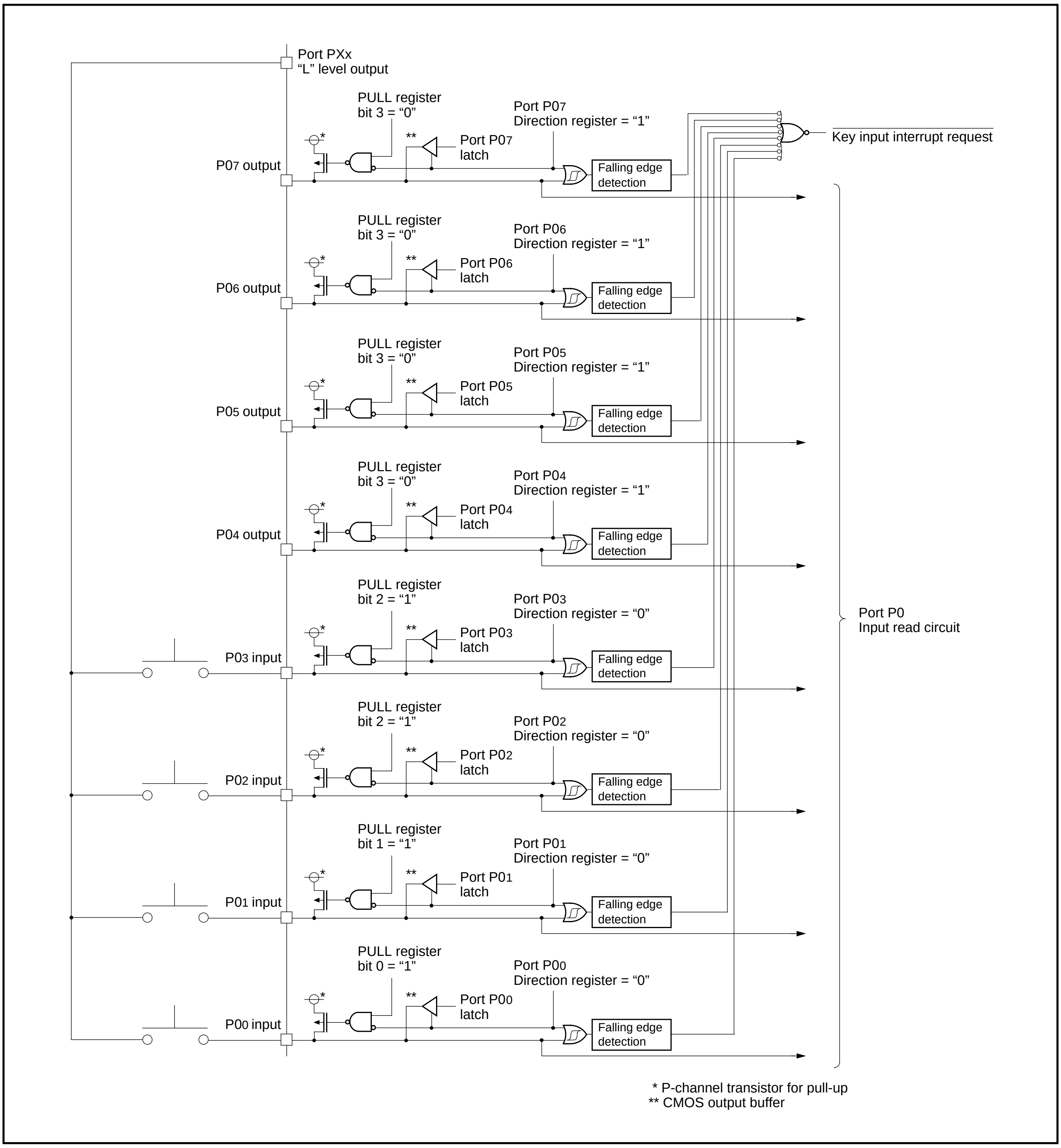

Port PXx
“L” level output
PULL register bit 3 = “0”
Port P07 Direction register = “1”
Port P07 latch
P07 output
Falling edge detection
Key input interrupt request
PULL register bit 3 = “0”
Port P06 Direction register = “1”
Port P06 latch
P06 output
PULL register bit 3 = “0”
Port P05 Direction register = “1”
Port P05 latch
P05 output
PULL register bit 3 = “0”
Port P04 Direction register = “1”
Port P04 latch
P04 output
PULL register bit 2 = “1”
Port P03 Direction register = “0”
Port P03 latch
P03 input
Port P0 Input read circuit
PULL register bit 2 = “1”
Port P02 Direction register = “0”
Port P02 latch
P02 input
PULL register bit 1 = “1”
Port P01 Direction register = “0”
Port P01 latch
P01 input
PULL register bit 0 = “1”
Port P00 Direction register = “0”
Port P00 latch
P00 input
* P-channel transistor for pull-up
** CMOS output buffer

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 CNTR$_0$ pin.

When the CNTR$_0$ active edge switch bit is “0”, the output of the CNTR$_0$ 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 P1$_4$ direction register to output mode.

• Event Counter Mode

The operation in the event counter mode is the same as that in the timer mode except that the timer counts the input signal from the CNTR$_0$ pin.

When the CNTR$_0$ active edge switch bit is “0”, the timer counts the rising edge of the CNTR$_0$ pin. When this bit is “1”, the timer counts the falling edge of the CNTR$_0$ pin.

• Pulse Width Measurement Mode

When the CNTR$_0$ active edge switch bit is “0”, the timer counts the signal selected by the timer X count source selection bit while the CNTR$_0$ pin is “H”. When this bit is “1”, the timer counts the signal while the CNTR$_0$ 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.

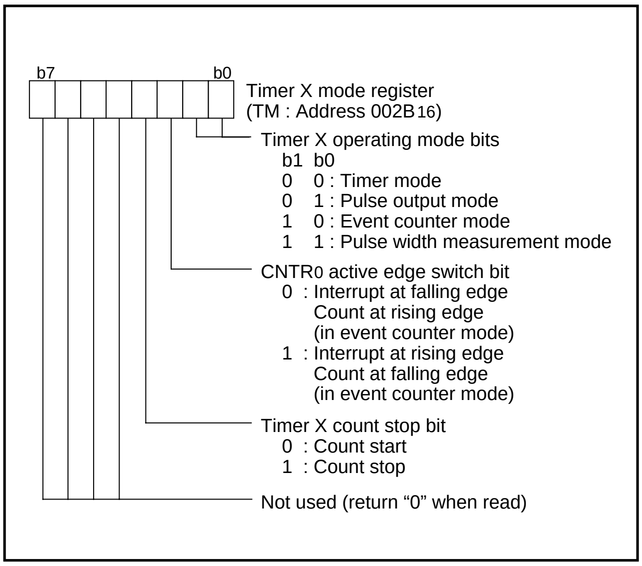


Fig. 21 Structure of timer X mode register

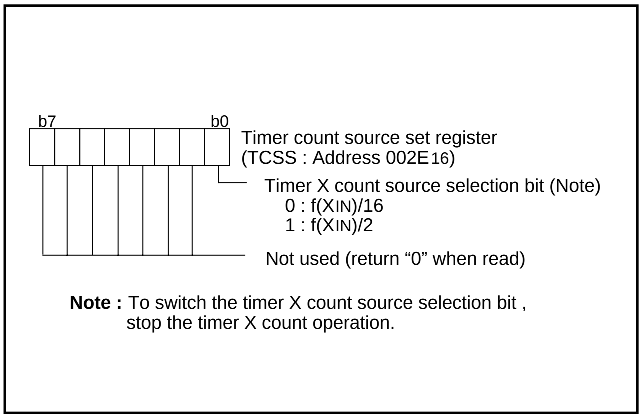

b7
b0
Timer count source set register
(TCSS : Address 002E16)
Timer X count source selection bit (Note)
0 : f(XIN)/16
1 : f(XIN)/2
Not used (return “0” when read)
Note : To switch the timer X count source selection bit ,
stop the timer X count operation.

Fig. 22 Timer count source set register

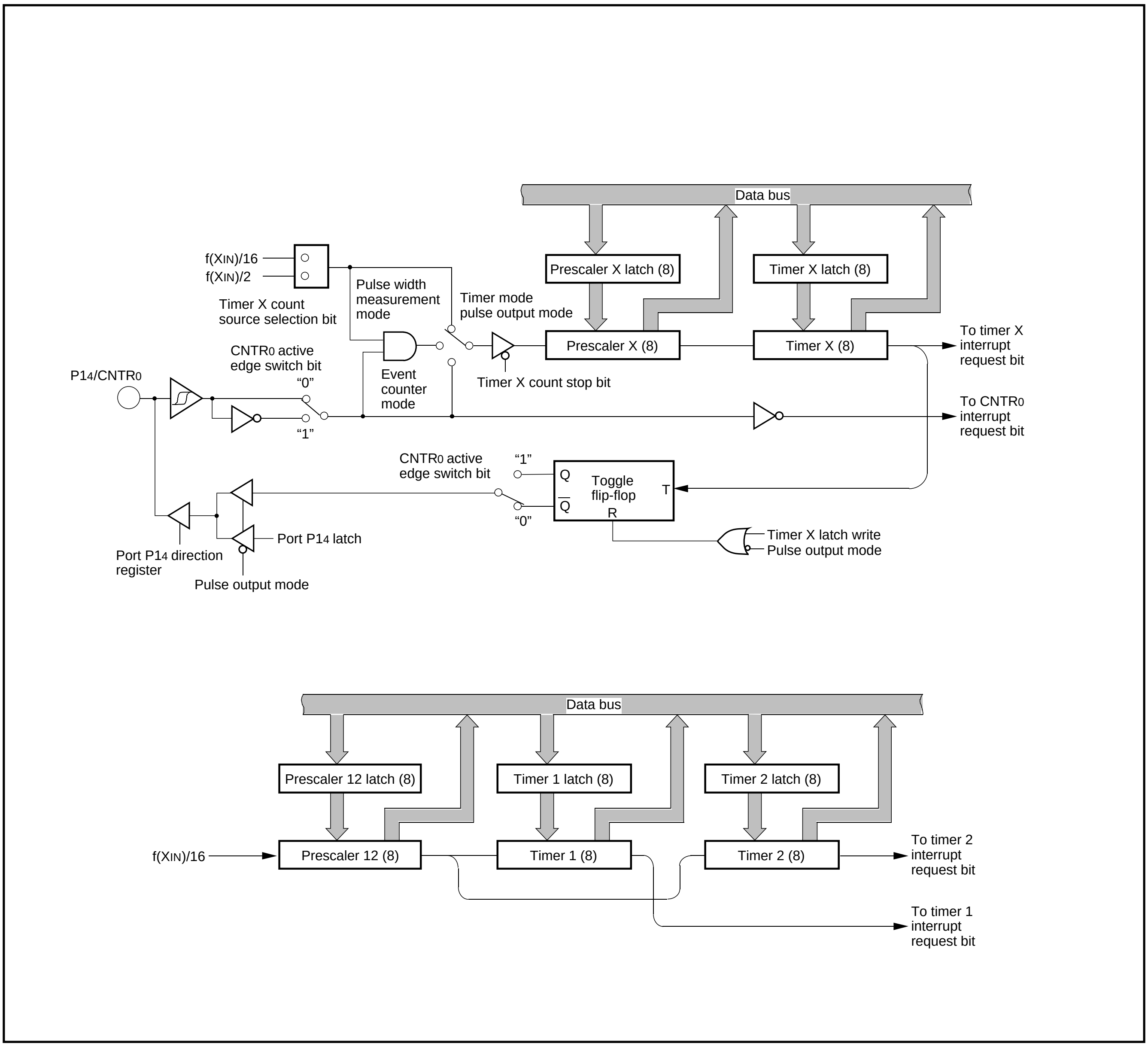


Fig. 23 Block diagram of timer X, timer 1 and timer 2

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.

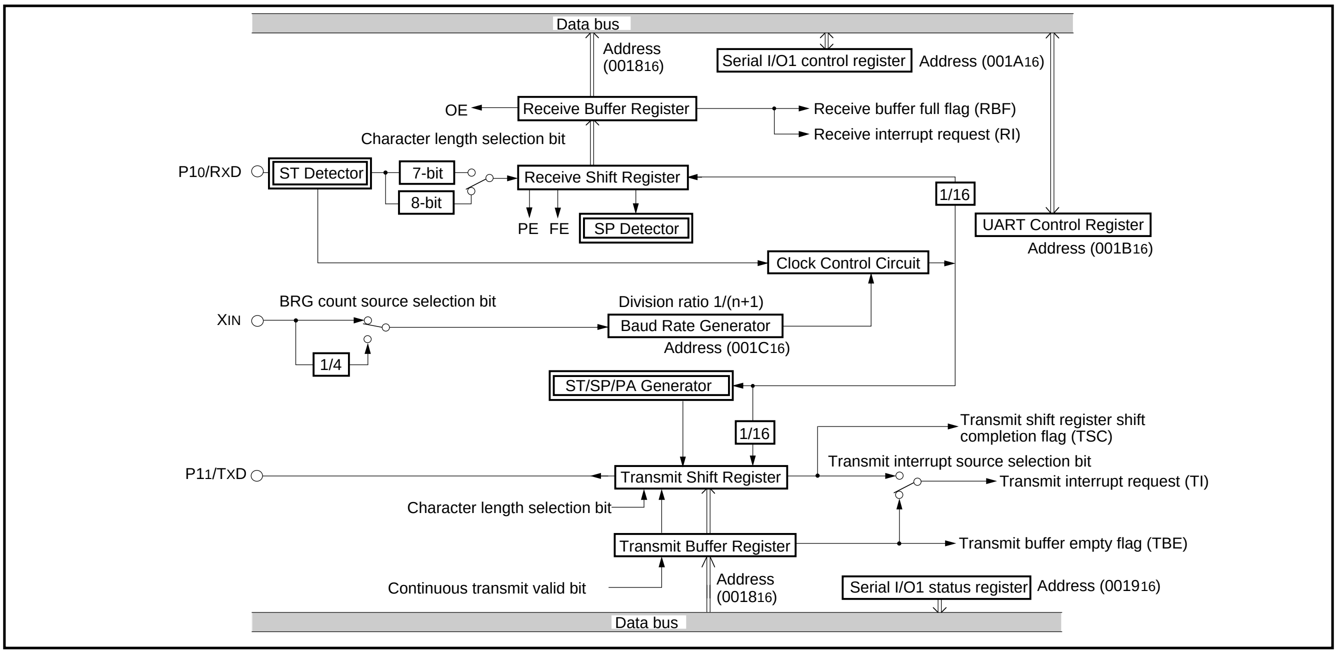
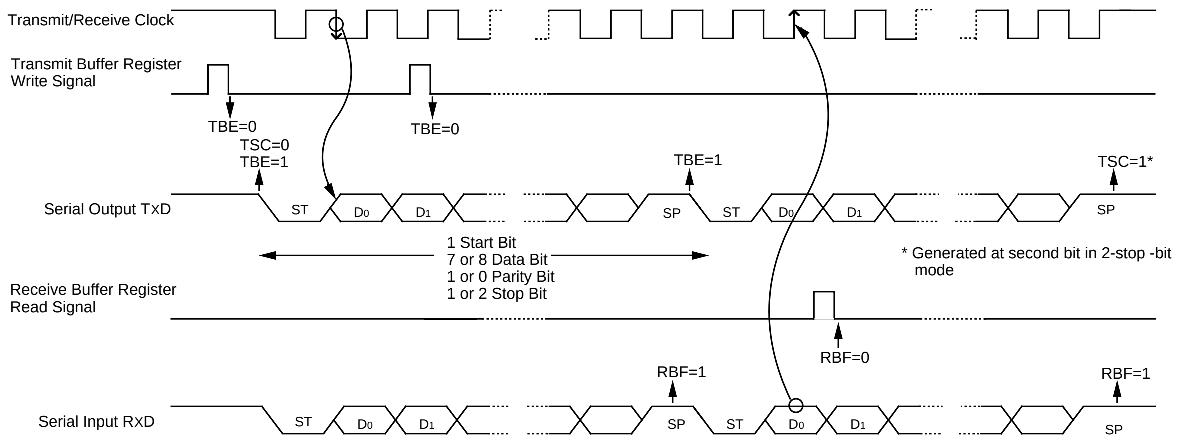

Data bus
Address (0018₁₆)
Serial I/O1 control register
Address (001A₁₆)
OE
Receive Buffer Register
Receive buffer full flag (RBF)
Receive interrupt request (RI)
Character length selection bit
P1₀/RXD
ST Detector
7-bit
8-bit
Receive Shift Register
1/16
UART Control Register
Address (001B₁₆)
PE FE
SP Detector
Clock Control Circuit
BRG count source selection bit
Division ratio 1/(n+1)
XIN
Baud Rate Generator
Address (001C₁₆)
1/4
ST/SP/PA Generator
1/16
Transmit shift register shift completion flag (TSC)
Transmit interrupt source selection bit
P1₁/TXD
Transmit Shift Register
Transmit interrupt request (TI)
Character length selection bit
Transmit Buffer Register
Transmit buffer empty flag (TBE)
Continuous transmit valid bit
Address (0018₁₆)
Serial I/O1 status register
Address (0019₁₆)
Data bus

Fig. 24 Block diagram of UART serial I/O1



Notes
1 : Error flag detection occurs at the same time that the RBF flag becomes “1” (at 1st stop bit, during reception).
2 : The transmit interrupt (TI) can be selected to occur when either the TBE or TSC flag becomes “1”, depending on the setting of the transmit interrupt source selection bit (TIC) of the serial I/O1 control register.
3 : The receive interrupt (RI) is set when the RBF flag becomes “1”.
4 : After data is written to the transmit buffer at TSC = 1, 0.5 to 1.5 cycles of the data shift cycle is necessary until changing to TSC = 0.

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 P1$_1$/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.

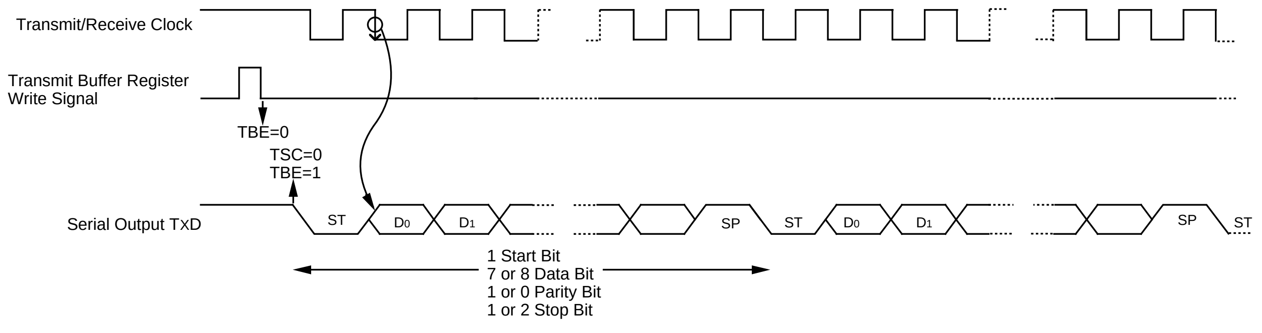
All bits of the serial I/O1 status register are initialized to “81$_{16}$” at reset, but if the transmit enable bit (bit 4) of the serial I/O1 control register has been set to “1”, the continuous transmit valid bit (bit 2) becomes “1”.

[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.



Notes 1 : When the serial I/O1 mode selection bits (b7, b6) is “10”, the transmit enable bit is “1”, and continuous transmit valid bit is “1”, writing on the transmit buffer initiates continuous transmission of the same data.
2 : Select 0 for continuous transmit valid bit to stop continuous transmission.
The TxD pin will stop at high level after completing transmission of 1 byte.
3 : If the transmit buffer contents are rewritten during a continuous transmission, transmission of the rewritten data will be started after completing transmission of 1 byte.

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 $001A_{16}$) 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.

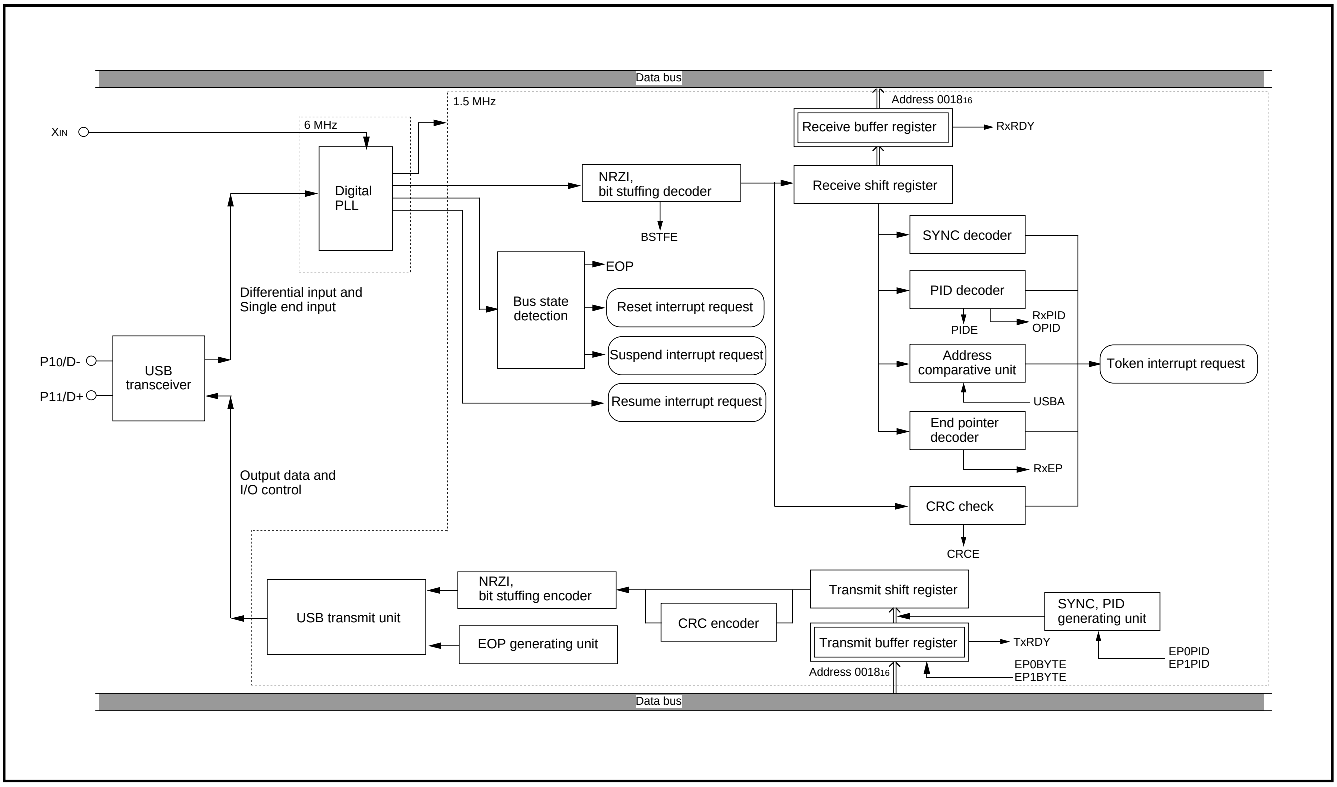


Fig. 27 USB mode block diagram

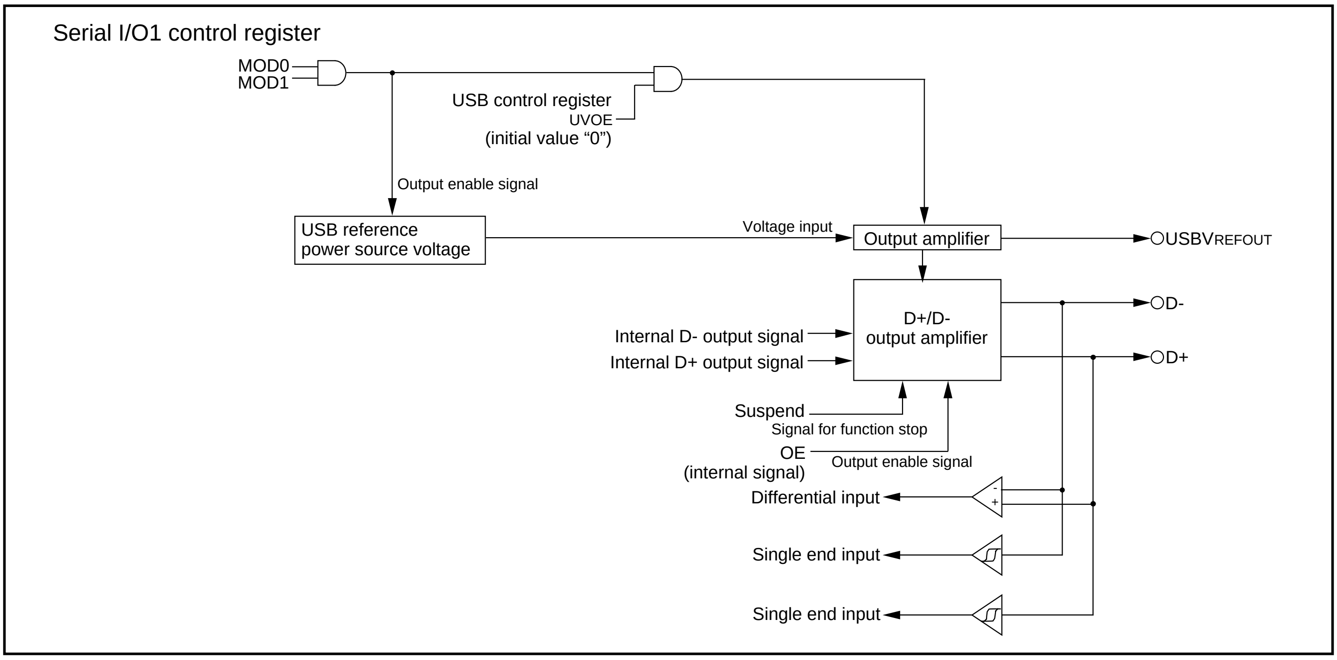
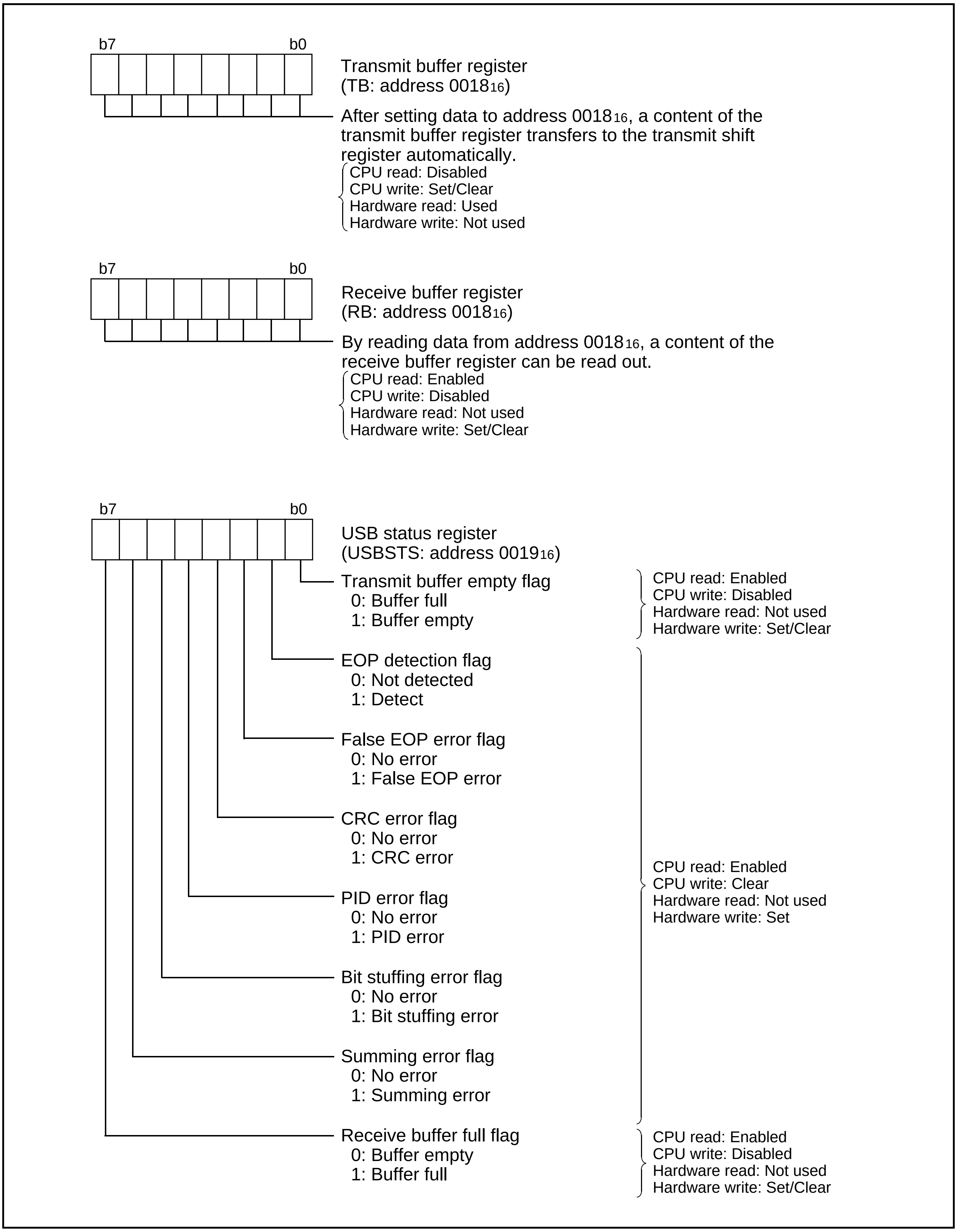


Fig. 28 USB transceiver block diagram



Transmit buffer register
(TB: address 0018_{16})

After setting data to address 0018_{16}, a content of the transmit buffer register transfers to the transmit shift register automatically.

CPU read: Disabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Not used

Receive buffer register
(RB: address 0018_{16})

By reading data from address 0018_{16}, a content of the receive buffer register can be read out.

CPU read: Enabled
CPU write: Disabled
Hardware read: Not used
Hardware write: Set/Clear

USB status register
(USBSTS: address 0019_{16})

Transmit buffer empty flag
0: Buffer full
1: Buffer empty

CPU read: Enabled
CPU write: Disabled
Hardware read: Not used
Hardware write: Set/Clear

EOP detection flag
0: Not detected
1: Detect

False EOP error flag
0: No error
1: False EOP error

CRC error flag
0: No error
1: CRC error

PID error flag
0: No error
1: PID error

Bit stuffing error flag
0: No error
1: Bit stuffing error

Summing error flag
0: No error
1: Summing error

CPU read: Enabled
CPU write: Clear
Hardware read: Not used
Hardware write: Set

Receive buffer full flag
0: Buffer empty
1: Buffer full

CPU read: Enabled
CPU write: Disabled
Hardware read: Not used
Hardware write: Set/Clear

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

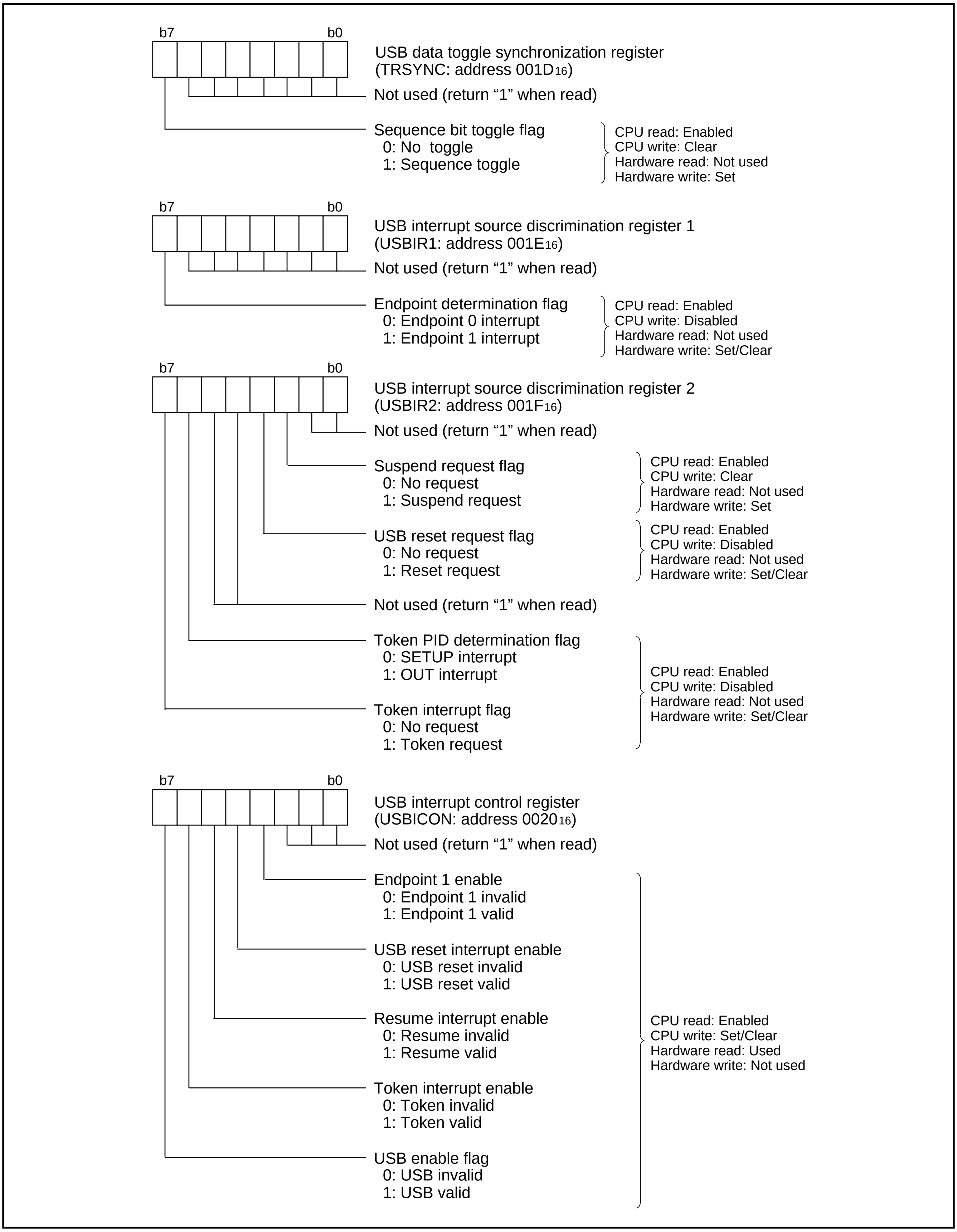
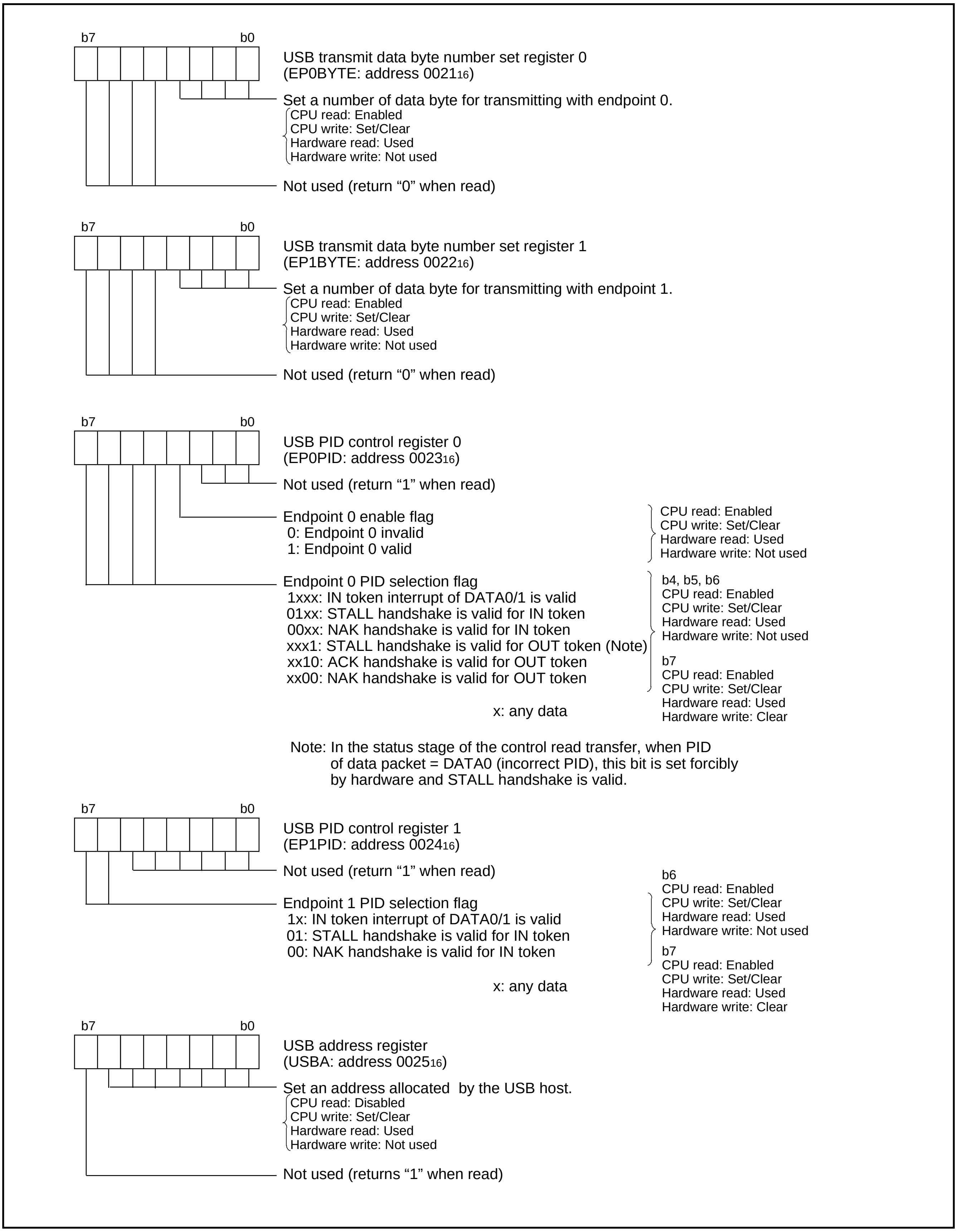


Fig. 30 Structure of serial I/O1-related registers (2)



USB transmit data byte number set register 0
(EP0BYTE: address 0021_{16})

Set a number of data byte for transmitting with endpoint 0.
CPU read: Enabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Not used

Not used (return “0” when read)

USB transmit data byte number set register 1
(EP1BYTE: address 0022_{16})

Set a number of data byte for transmitting with endpoint 1.
CPU read: Enabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Not used

Not used (return “0” when read)

USB PID control register 0
(EP0PID: address 0023_{16})

Not used (return “1” when read)

Endpoint 0 enable flag
0: Endpoint 0 invalid
1: Endpoint 0 valid

CPU read: Enabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Not used

Endpoint 0 PID selection flag
1xxx: IN token interrupt of DATA0/1 is valid
01xx: STALL handshake is valid for IN token
00xx: NAK handshake is valid for IN token
xxx1: STALL handshake is valid for OUT token (Note)
xx10: ACK handshake is valid for OUT token
xx00: NAK handshake is valid for OUT token

x: any data

b4, b5, b6
CPU read: Enabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Not used

b7
CPU read: Enabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Clear

Note: In the status stage of the control read transfer, when PID of data packet = DATA0 (incorrect PID), this bit is set forcibly by hardware and STALL handshake is valid.

USB PID control register 1
(EP1PID: address 0024_{16})

Not used (return “1” when read)

Endpoint 1 PID selection flag
1x: IN token interrupt of DATA0/1 is valid
01: STALL handshake is valid for IN token
00: NAK handshake is valid for IN token

x: any data

b6
CPU read: Enabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Not used

b7
CPU read: Enabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Clear

USB address register
(USBA: address 0025_{16})

Set an address allocated by the USB host.
CPU read: Disabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Not used

Not used (returns “1” when read)

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

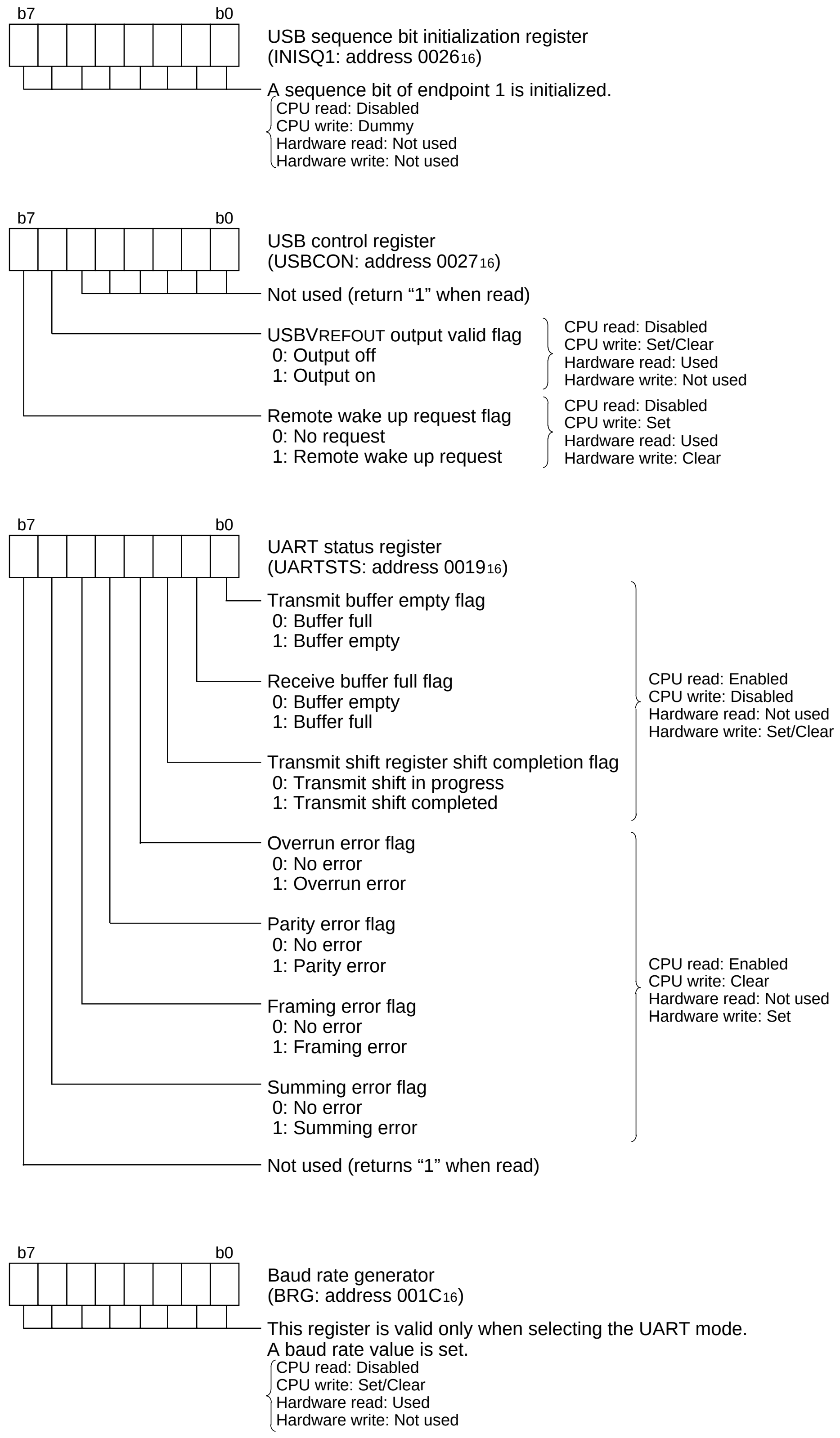


Fig. 32 Structure of serial I/O1-related registers (4)

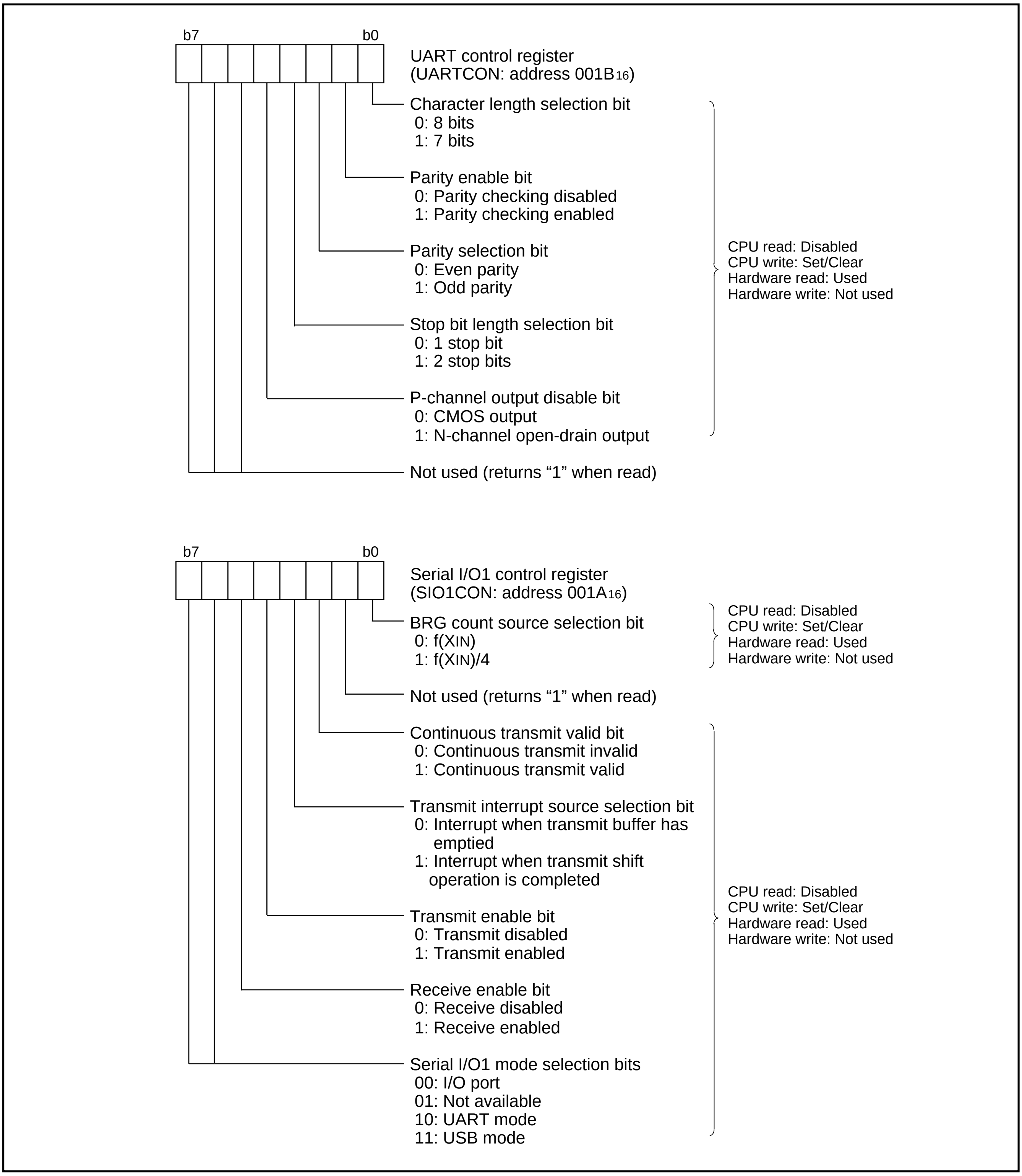

b7
b0
UART control register
(UARTCON: address 001B16)
Character length selection bit
0: 8 bits
1: 7 bits
Parity enable bit
0: Parity checking disabled
1: Parity checking enabled
Parity selection bit
0: Even parity
1: Odd parity
Stop bit length selection bit
0: 1 stop bit
1: 2 stop bits
P-channel output disable bit
0: CMOS output
1: N-channel open-drain output
Not used (returns “1” when read)
CPU read: Disabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Not used
b7
b0
Serial I/O1 control register
(SIO1CON: address 001A16)
BRG count source selection bit
0: f(XIN)
1: f(XIN)/4
CPU read: Disabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Not used
Not used (returns “1” when read)
Continuous transmit valid bit
0: Continuous transmit invalid
1: Continuous transmit valid
Transmit interrupt source selection bit
0: Interrupt when transmit buffer has emptied
1: Interrupt when transmit shift operation is completed
Transmit enable bit
0: Transmit disabled
1: Transmit enabled
Receive enable bit
0: Receive disabled
1: Receive enabled
Serial I/O1 mode selection bits
00: I/O port
01: Not available
10: UART mode
11: USB mode
CPU read: Disabled
CPU write: Set/Clear
Hardware read: Used
Hardware write: Not used

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 20_{16})
•RSME: Resume interrupt enable (bit 5 of address 20_{16})
•RSTE: USB reset interrupt enable (bit 4 of address 20_{16})
•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 19_{16})
•RxPID: Token interrupt flag (bit 7 of address $1F_{16}$)

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	Initialize suspend timer count value	Token interrupt request	Set EOP flag	SIE	Reset interrupt request
	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		
	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	F/W	Reset interrupt processing Resume interrupt processing
2.67 µs	Spec	Reset	Reset	Reset		
	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 0023_{16})

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.

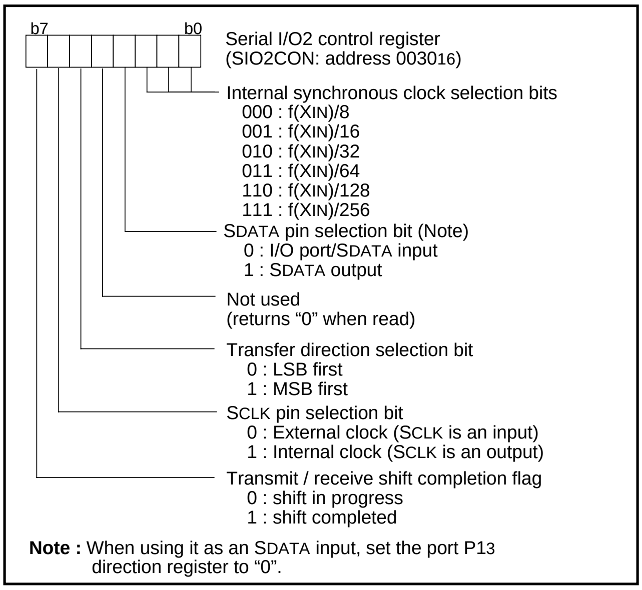

b7
b0
Serial I/O2 control register
(SIO2CON: address 003016)
Internal synchronous clock selection bits
000 : f(XIN)/8
001 : f(XIN)/16
010 : f(XIN)/32
011 : f(XIN)/64
110 : f(XIN)/128
111 : f(XIN)/256
SDATA pin selection bit (Note)
0 : I/O port/SDATA input
1 : SDATA output
Not used
(returns "0" when read)
Transfer direction selection bit
0 : LSB first
1 : MSB first
SCLK pin selection bit
0 : External clock (SCLK is an input)
1 : Internal clock (SCLK is an output)
Transmit / receive shift completion flag
0 : shift in progress
1 : shift completed
Note : When using it as an SDATA input, set the port P13 direction register to "0".

Fig. 34 Structure of serial I/O2 control registers

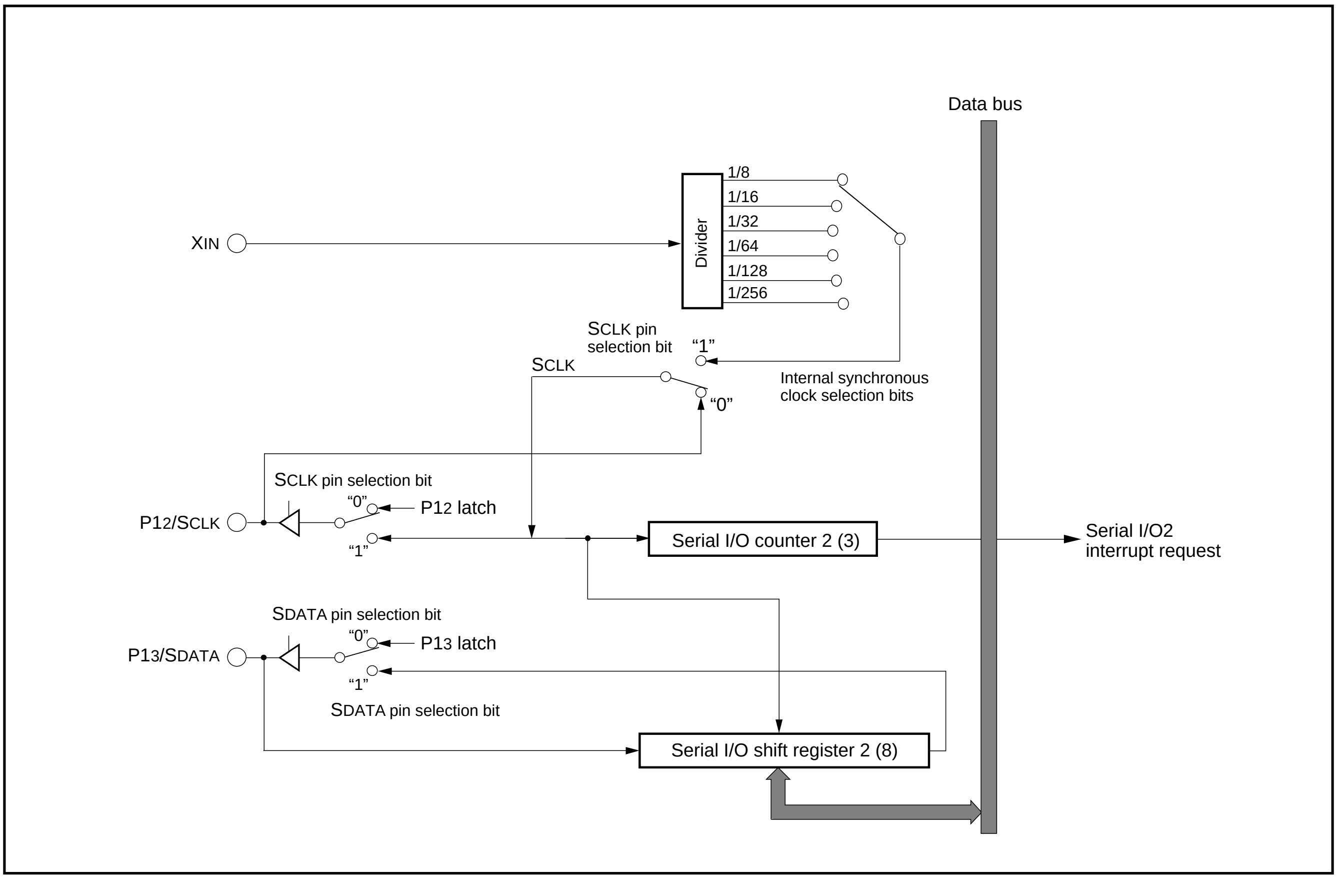


Fig. 35 Block diagram of serial I/O2

Serial I/O2 operation

By writing to the serial I/O2 register(address 0031_{16}) the serial I/O2 counter is set to “7”.

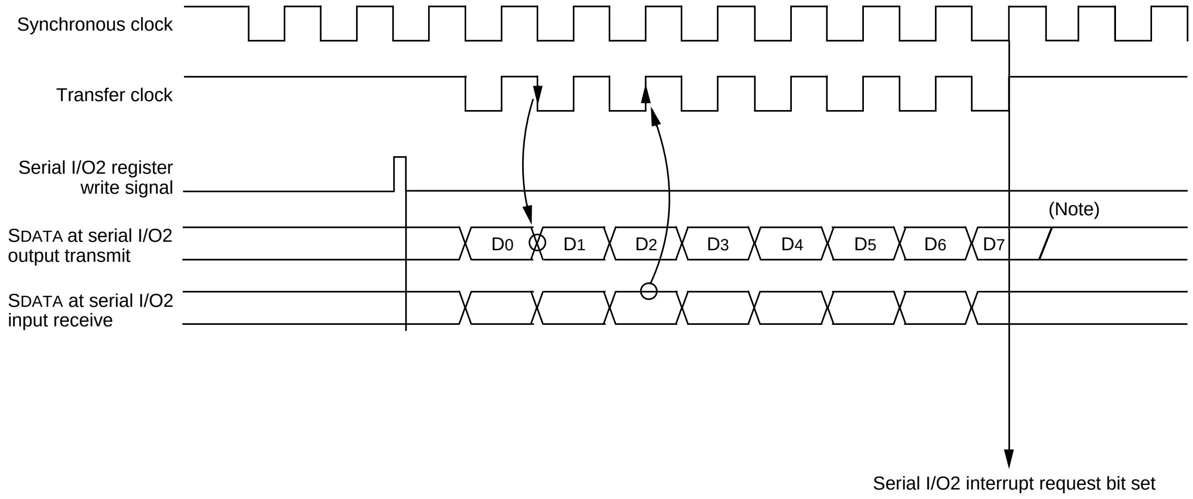
After writing, the SDATA pin outputs data every time the transfer clock shifts from a high to a low level. And, as the transfer clock shifts from a low to a high, the SDATA pin reads data, and at the same time the contents of the serial I/O2 register are shifted by 1 bit.

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.


Synchronous clock
Transfer clock
Serial I/O2 register write signal
SDATA at serial I/O2 output transmit
SDATA at serial I/O2 input receive
D0
D1
D2
D3
D4
D5
D6
D7
(Note)
Serial I/O2 interrupt request bit set

Note : When the internal clock is selected as the transfer and the direction register of P1 3/SDATA pin is set to the input mode, the SDATA pin is in a high impedance state after the data transfer is completed.

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 P2$_7$/AN$_7$ to P2$_0$/AN$_0$, 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.

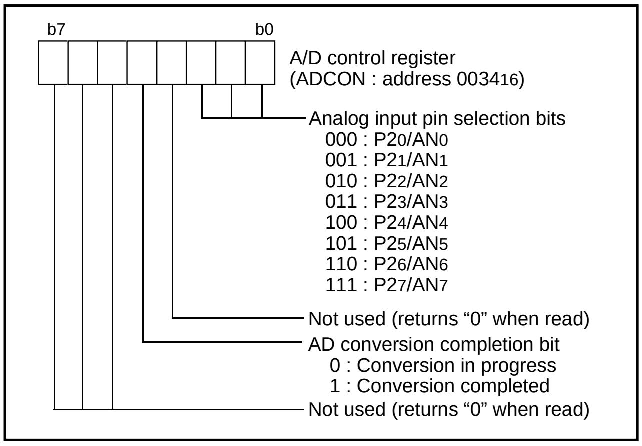


Fig. 37 Structure of A/D control register

Read 8-bit (Read out only address 0035$_{16}$)

	b7							b0
(Address 0035$_{16}$)	b9	b8	b7	b6	b5	b4	b3	b2

Read 10-bit (read out in order address 0036$_{16}$, 0035$_{16}$)

	b7							b0
(Address 0036$_{16}$)	▨	▨	▨	▨	▨	▨	b9	b8

	b7							b0
(Address 0035$_{16}$)	b7	b6	b5	b4	b3	b2	b1	b0

High-order 6-bit of address 0036$_{16}$ returns “0” when read.

Fig. 38 Structure of A/D conversion register

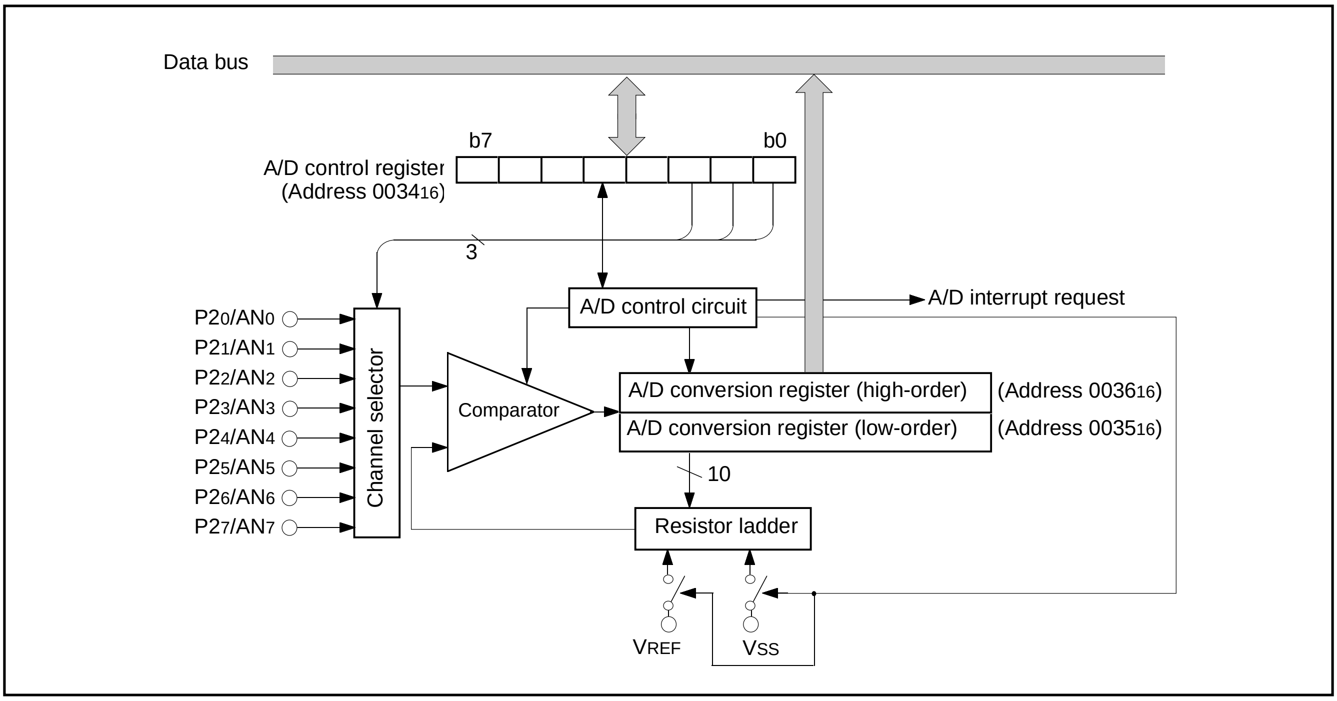


Fig. 39 Block diagram of A/D converter

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 0039_{16}) is not set after reset. Writing an optional value to the watchdog timer control register (address 0039_{16}) causes the watchdog timer to start to count down. When the watchdog timer H underflows, an internal reset occurs. Accordingly, it is programmed that the watchdog timer control register (address 0039_{16}) can be set before an underflow occurs.

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

Initial value of watchdog timer

By a reset or writing to the watchdog timer control register (address 0039_{16}), the watchdog timer H is set to "FF_{16}" and the watchdog timer L is set to "FF_{16}".

Operation of watchdog timer H count source selection bit

A watchdog timer H count source can be selected by bit 7 of the watchdog timer control register (address 0039_{16}). 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 0039_{16}).

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.

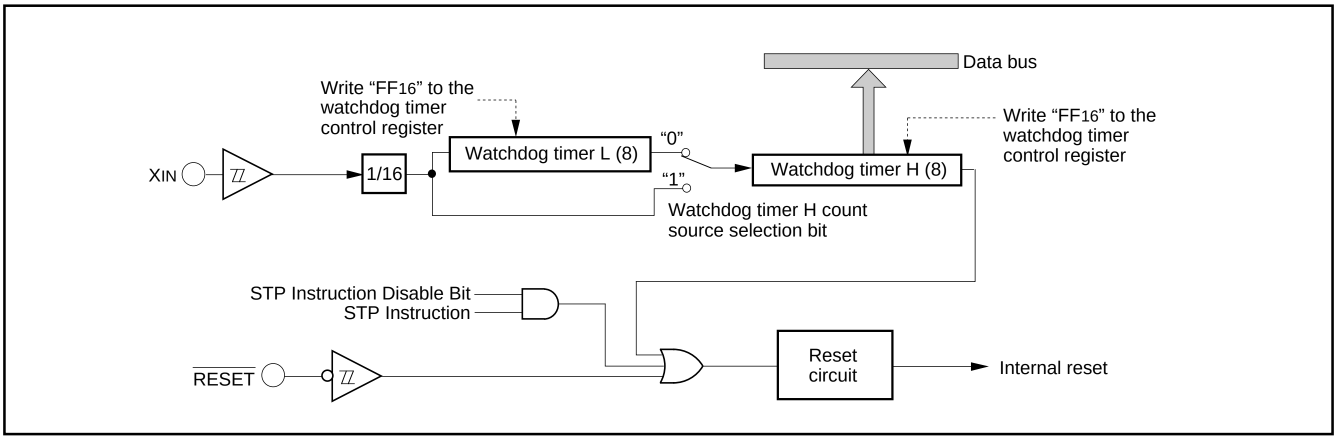

Data bus
Write "FF16" to the watchdog timer control register
Write "FF16" to the watchdog timer control register
XIN
1/16
Watchdog timer L (8)
"0"
"1"
Watchdog timer H (8)
Watchdog timer H count source selection bit
STP Instruction Disable Bit
STP Instruction
RESET
Reset circuit
Internal reset

Fig. 40 Block diagram of watchdog timer

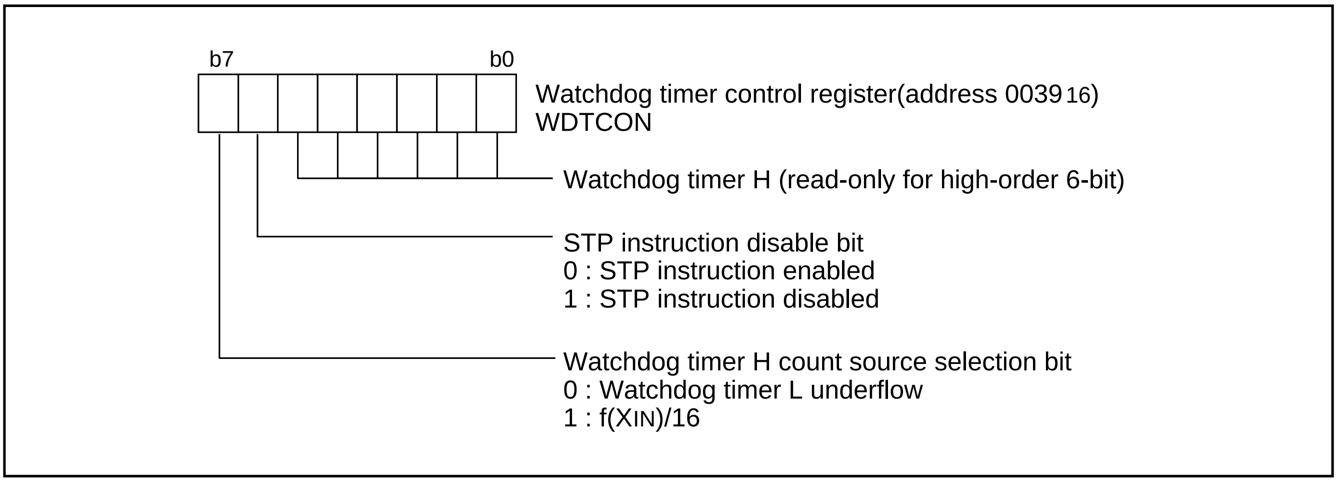

b7
b0
Watchdog timer control register(address 0039 16)
WDTCON
Watchdog timer H (read-only for high-order 6-bit)
STP instruction disable bit
0 : STP instruction enabled
1 : STP instruction disabled
Watchdog timer H count source selection bit
0 : Watchdog timer L underflow
1 : f(XIN)/16

Fig. 41 Structure of watchdog timer control register

Reset Circuit

The microcomputer is put into a reset status by holding the 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 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.

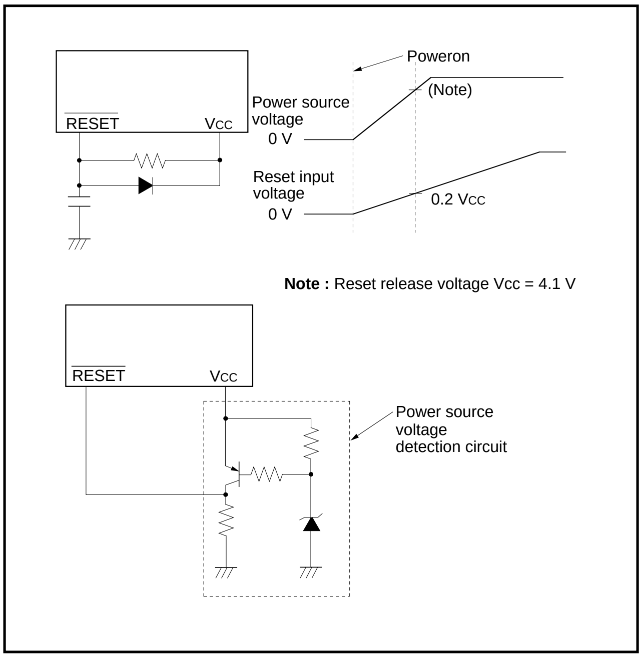
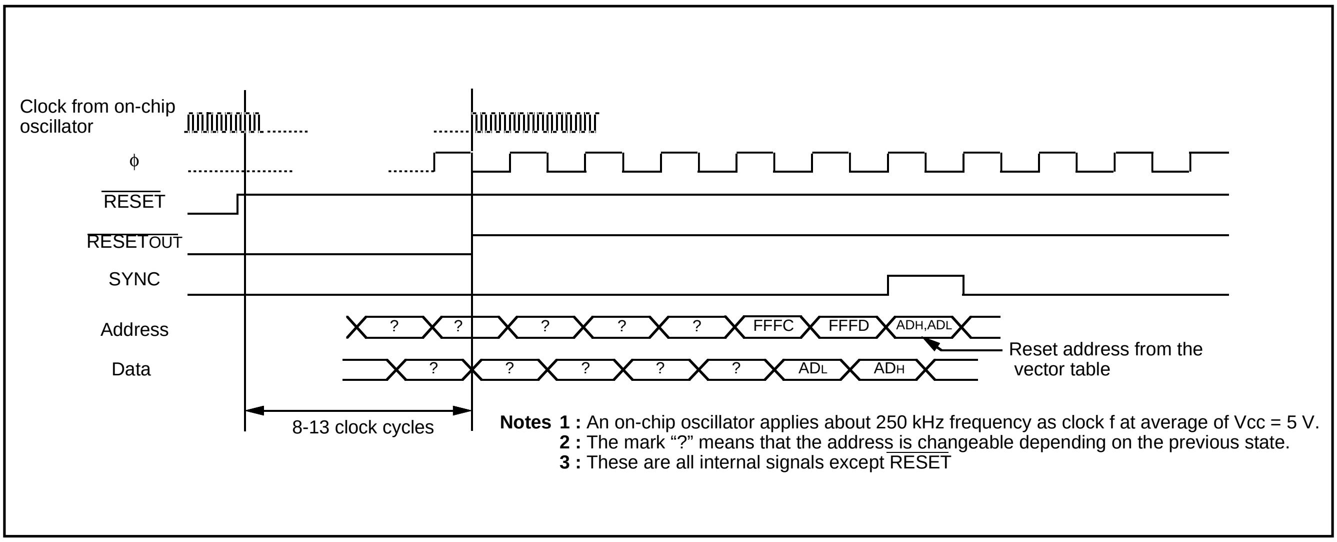


Fig. 42 Example of reset circuit



Notes 1 : An on-chip oscillator applies about 250 kHz frequency as clock f at average of Vcc = 5 V.
2 : The mark "?" means that the address is changeable depending on the previous state.
3 : These are all internal signals except RESET

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	(PC_H)	Contents of address $FFFD_{16}$
		(PC_L)	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 XIN and XOUT.

Use the circuit constants in accordance with the resonator manufacturer's recommended values. No external resistor is needed between XIN and XOUT since a feed-back resistor exists on-chip. (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 XIN oscillator stops. At this time, timer 1 is set to "01_{16}" and prescaler 12 is set to "FF_{16}" 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(XIN)/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 $\overline{\text{RESET}}$ 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 XIN divided by 16, so set the timer 1 interrupt enable bit to "0" before the STP instruction is executed.

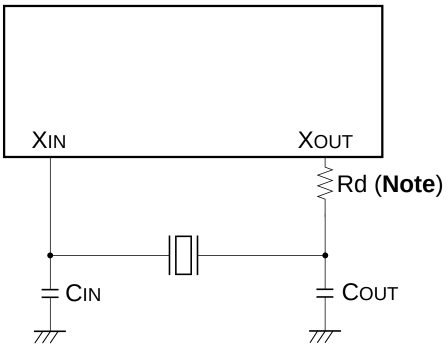
Note

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.



Note : Insert a damping resistor if required.
The resistance will vary depending on the oscillator and the oscillation drive capacity setting.
Use the value recommended by the maker of the oscillator.
Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip though a feedback resistor exists on-chip, insert a feedback resistor between XIN and XOUT following the instruction.

Fig. 45 External circuit of ceramic resonator

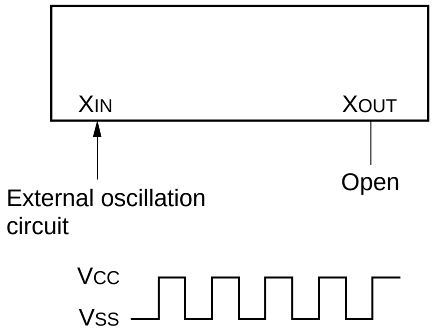


Fig. 46 External clock input circuit

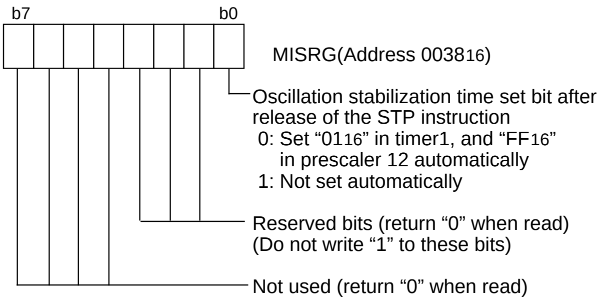
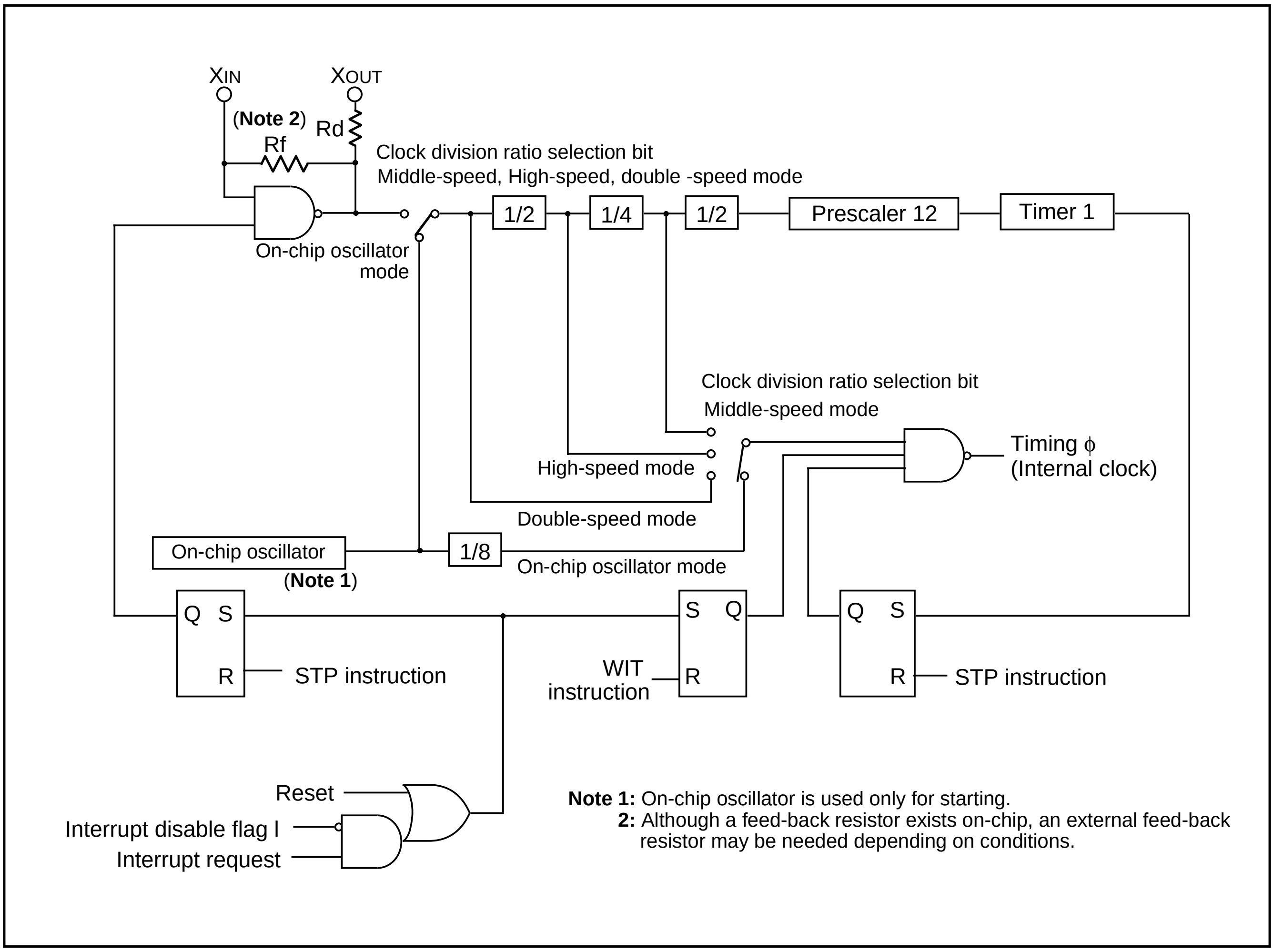

b7
b0
MISRG(Address 003816)
Oscillation stabilization time set bit after release of the STP instruction
0: Set "0116" in timer1, and "FF16" in prescaler 12 automatically
1: Not set automatically
Reserved bits (return "0" when read)
(Do not write "1" to these bits)
Not used (return "0" when read)

Fig. 47 Structure of MISRG



Note 1: On-chip oscillator is used only for starting.
2: Although a feed-back resistor exists on-chip, an external feed-back resistor may be needed depending on conditions.

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 (Vcc pin) and GND pin (Vss 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 µF 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 (VSS pin). Besides, connect the capacitor to as close as possible. For bypass capacitor, a ceramic or electrolytic capacitor of 0.22 µF 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 $P1_0$, $P1_2$, $P1_3$ input level selection bit (bit 3 of address 17_{16}) 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 $P1_0$, $P1_2$, $P1_3$ 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.

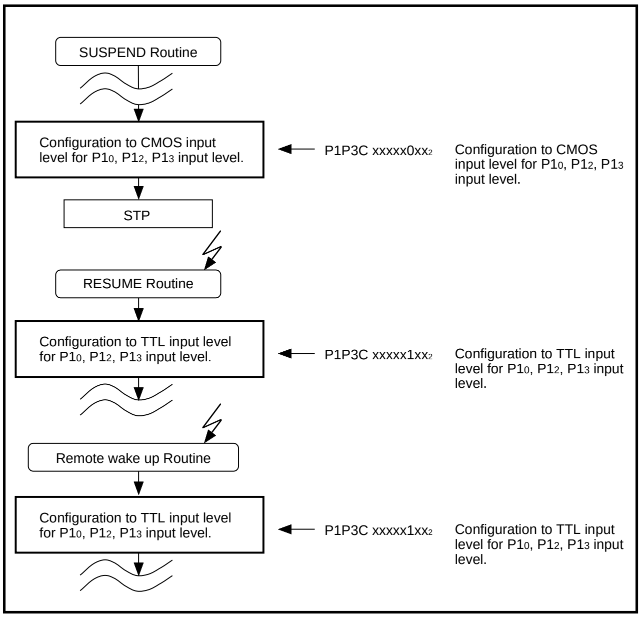

SUSPEND Routine
Configuration to CMOS input level for P1$_0$, P1$_2$, P1$_3$ input level.
P1P3C xxxxx0xx$_2$
Configuration to CMOS input level for P1$_0$, P1$_2$, P1$_3$ input level.
STP
RESUME Routine
Configuration to TTL input level for P1$_0$, P1$_2$, P1$_3$ input level.
P1P3C xxxxx1xx$_2$
Configuration to TTL input level for P1$_0$, P1$_2$, P1$_3$ input level.
Remote wake up Routine
Configuration to TTL input level for P1$_0$, P1$_2$, P1$_3$ input level.
P1P3C xxxxx1xx$_2$
Configuration to TTL input level for P1$_0$, P1$_2$, P1$_3$ input level.

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.

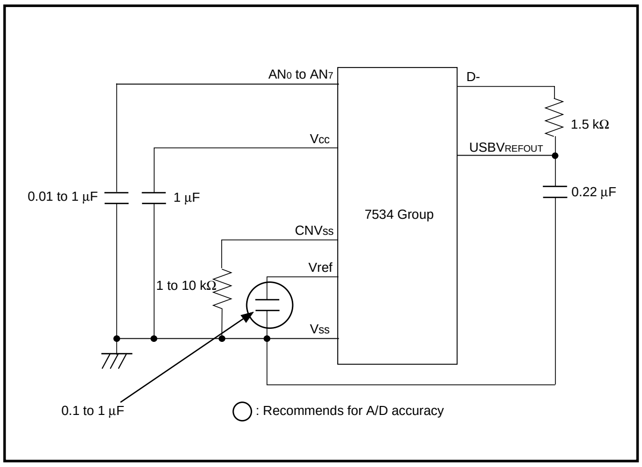


Fig. 50 Method to stabilize A/D conversion accuracy

(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).

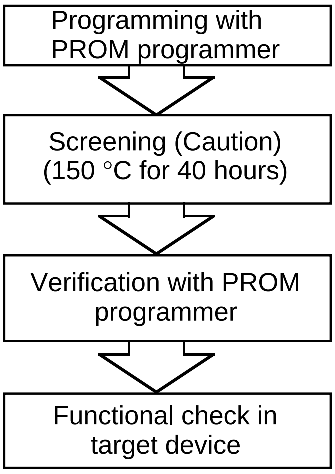
ROM PROGRAMMING METHOD

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.


Programming with PROM programmer
Screening (Caution) (150 °C for 40 hours)
Verification with PROM programmer
Functional check in target device

Caution: The screening temperature is far higher than the storage temperature. Never expose to 150 °C exceeding 100 hours.

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

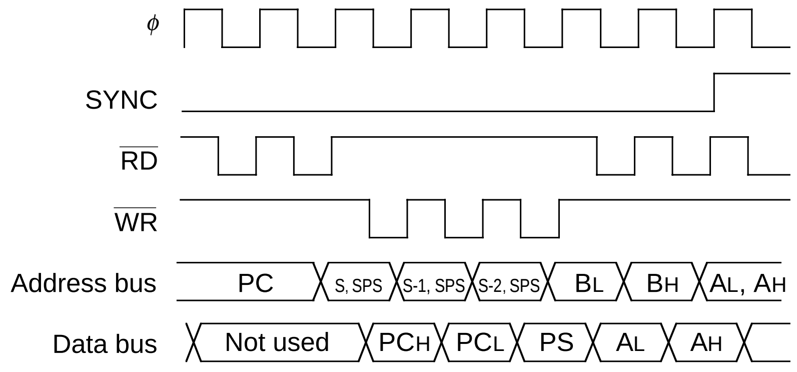
Interrupt source	Priority	Vector addresses (Note 1)		Interrupt request generating conditions	Remarks
		High-order	Low-order		
Reset (Note 2)	1	$FFFD_{16}$	$FFFC_{16}$	At reset input	Non-maskable
UART receive	2	$FFFB_{16}$	$FFFA_{16}$	At completion of UART data receive	Valid in UART mode
USB IN token				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 Reset/Suspend/Resume				At detection of SETUP/OUT token or At detection of Reset/ Suspend/ Resume	Valid in USB mode
INT$_1$ (Note 3)				At detection of either rising or falling edge of INT$_1$ input	External interrupt (active edge selectable)
INT$_0$ (Note 4)	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/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 INT$_1$ interrupt does not exist in the 36-pin and 32-pin version.
4: The INT$_0$ 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.



SYNC : CPU operation code fetch cycle
BL, BH : Vector address of each interrupt
AL, AH : Jump destination address of each interrupt
SPS : “0016” or “0116”

Fig. 52 Timing chart after an interrupt occurs

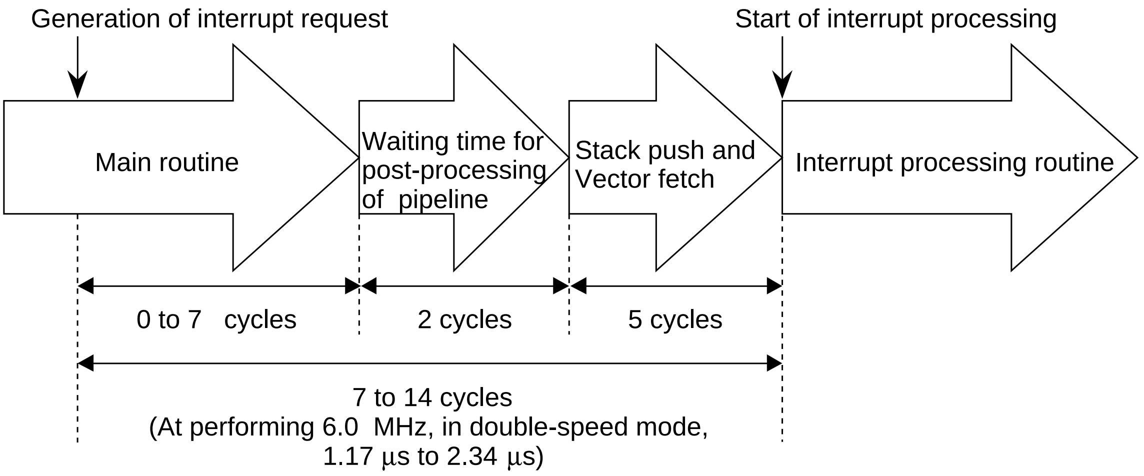

Generation of interrupt request
Start of interrupt processing
Main routine
Waiting time for post-processing of pipeline
Stack push and Vector fetch
Interrupt processing routine
0 to 7 cycles
2 cycles
5 cycles
7 to 14 cycles
(At performing 6.0 MHz, in double-speed mode, 1.17 µs to 2.34 µs)

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 “00_{16}.”
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 V_{IN}.
3. As a result of comparison, when Vref < V_{IN}, the highest-order bit of A/D conversion register becomes “1.” When Vref > V_{IN}, the highest-order bit becomes “0.”

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

A/D conversion completes at 122 clock cycles (20.34 µs at $f(X_{IN})$ = 6.0 MHz) after it is started, and the result of the conversion is stored into the A/D conversion register.

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

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

When n = 0 Vref = 0

When n = 1 to 1023 Vref = $\frac{\text{VREF}}{1024} \times n$

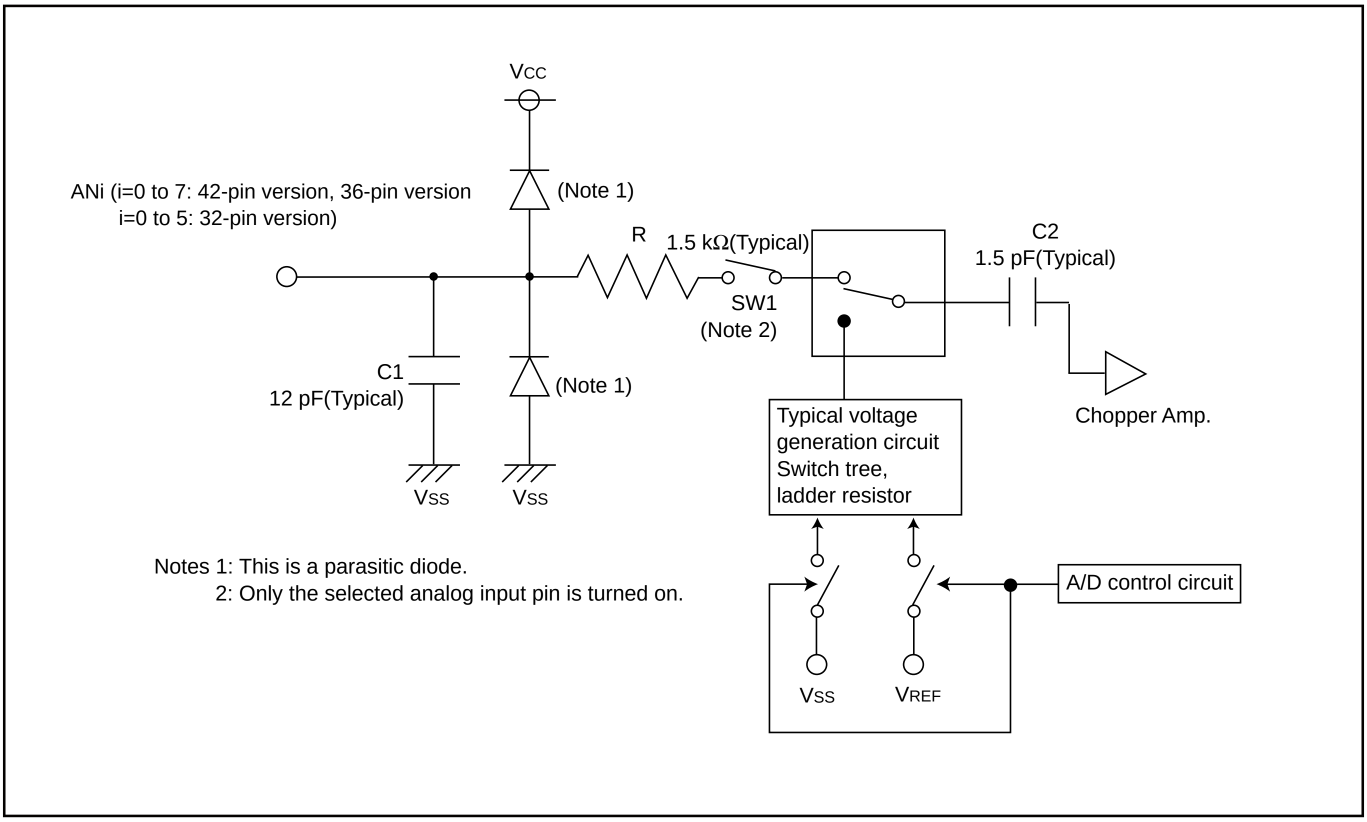
n : the value of A/D converter (decimal numeral)

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

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

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


VCC
ANi (i=0 to 7: 42-pin version, 36-pin version
i=0 to 5: 32-pin version)
(Note 1)
R
1.5 kΩ(Typical)
SW1
(Note 2)
C2
1.5 pF(Typical)
C1
12 pF(Typical)
(Note 1)
VSS
VSS
Typical voltage generation circuit Switch tree, ladder resistor
Chopper Amp.
A/D control circuit
VSS
VREF

Notes 1: This is a parasitic diode.
2: Only the selected analog input pin is turned on.

Fig. 54 A/D conversion equivalent circuit

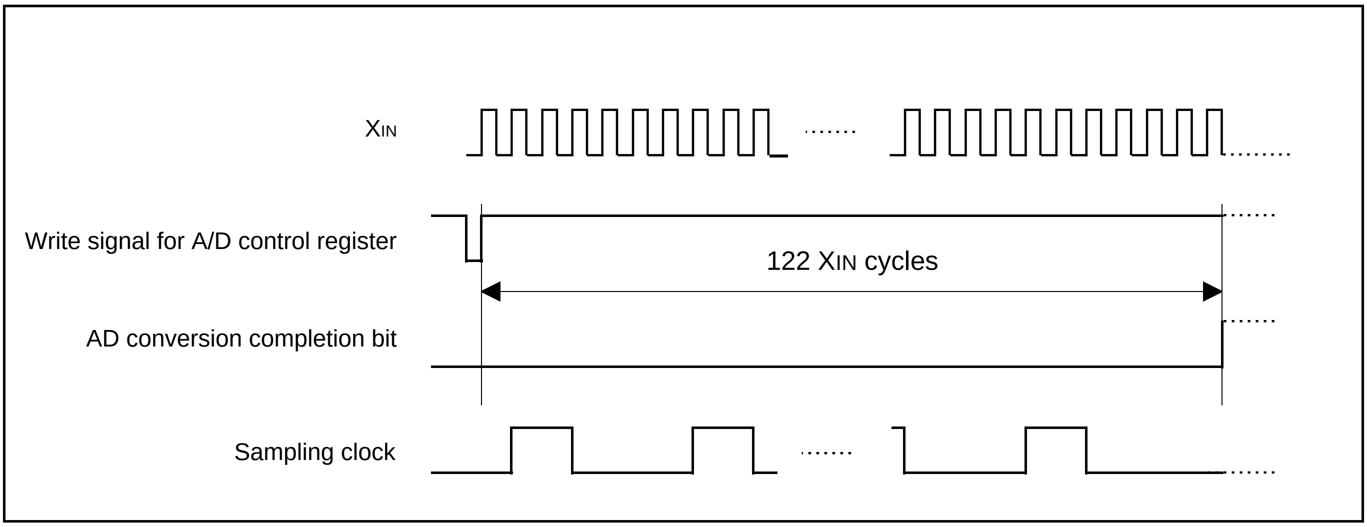

XIN
Write signal for A/D control register
122 XIN cycles
AD conversion completion bit
Sampling clock

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(XIN) oscillation is stopped, and the internal clock ϕ is stopped. Accordlingly, 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
Oscillation	Stop
CPU	Stop
I/O port	State retained at **STP** instruction execution
Timer	At selecting internal count source: Stop At selecting external count source: Operating (only Timer X)
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)
Watchdog timer	Stop
RAM	State retained
SFR	State retained (Timer 1 and prescaler 12 excepted)
CPU register	State retained • Accumulator • Index register X • Index register Y • Stack pointer • Program counter • Processor status register

(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 interupt request bit of the interrupt using for return to “0”
⑥ Set the interupt 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. Accordlingly, 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 • 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.

- $\mathrm{INT_0}$
- $\mathrm{INT_1}$
- $\mathrm{CNTR_0}$
- 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 interupt request bit of the interrupt using for return to “0”
② Set the interupt 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 pF to 350 pF.	Deal with the following Low-Speed USB2.0 specification. C_L = 200 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 µA not including the output current of $USBV_{REFOUT}$.	Rating is Max. 300 µA including the output current of $USBV_{REFOUT}$, 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 automaticcally 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 (80_{16}) 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

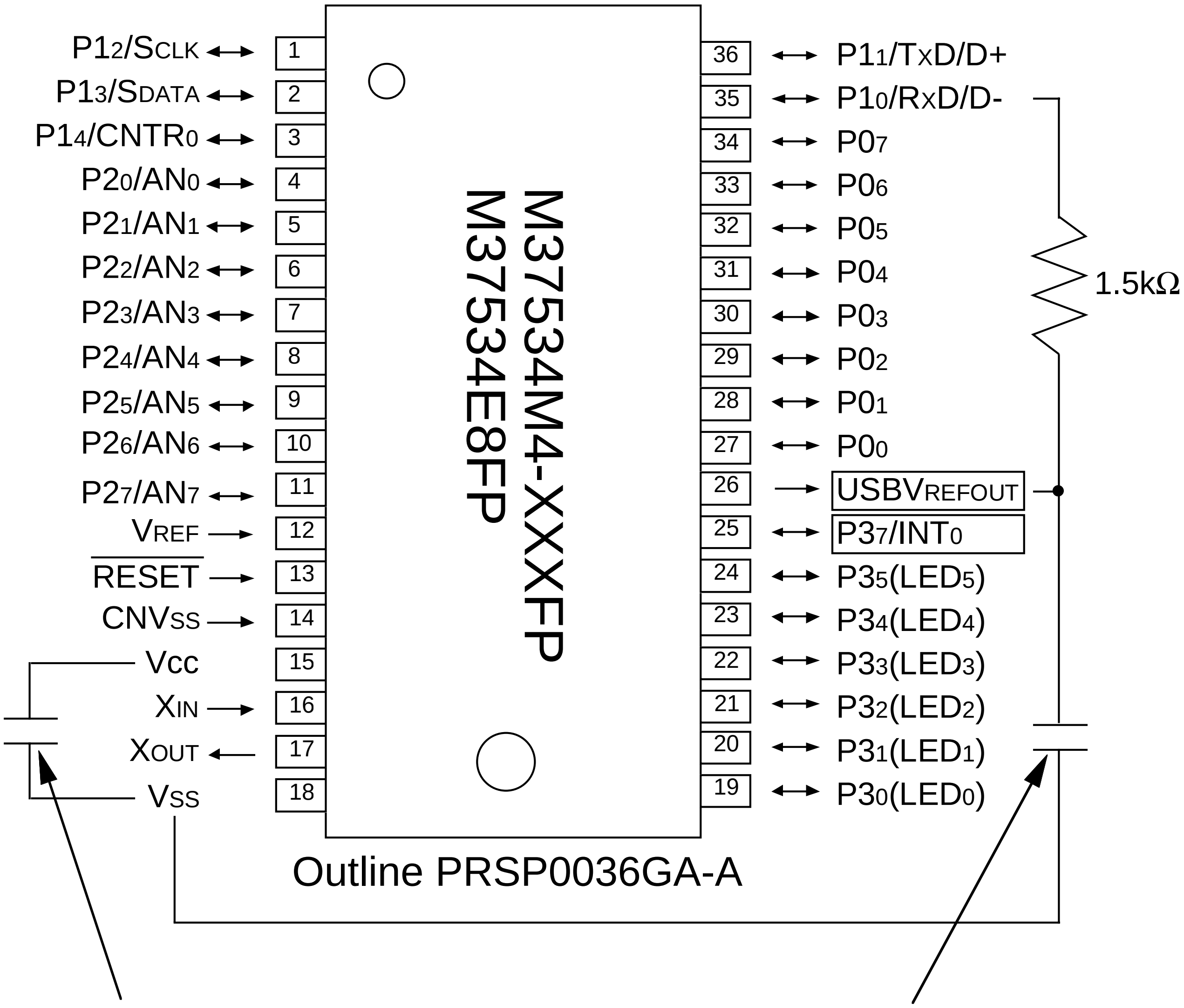
I/O port	42-pin SDIP	36-pin SSOP	32-pin LQFP
Port P1	$P1_0$–$P1_6$ (7-bit structure)	$P1_0$–$P1_4$ (5-bit structure)	$P1_0$–$P1_4$ (5-bit structure)
Port P2	$P2_0$–$P2_7$ (8-bit structure) (A/D converter 8-channel)	$P2_0$–$P2_7$ (8-bit structure) (A/D converter 8-channel)	$P2_0$–$P2_5$ (6-bit structure) (A/D converter 6-channel)
Port P3	$P3_0$–$P3_7$ (8-bit structure) (INT_0, INT_1 available)	$P3_0$–$P3_5$, $P3_7$ (7-bit structure) (INT_0 available)	$P3_0$–$P3_4$ (5-bit structure) (INT function not available)
Port P4	$P4_0$, $P4_1$ (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 (02_{16}/03_{16})	Bit 7 not available	Bits 5 to 7 not available	Bits 5 to 7 not available
Port P2/Direction (04_{16}/05_{16})	All bits available	All bits available	Bits 6 and 7 not available
Port P3/Direction (06_{16}/07_{16})	All bits available	Bit 6 not available	Bits 5 to 7 not available
Port P4/Direction (08_{16}/09_{16})	Bits 2 to 7 not available	All bits not available	All bits not available
Pull-up control (16_{16})	Bit 6 definition: “P3$_5$, P3$_6$ pull-up control” Bit 7 definition: “P3$_7$ pull-up control”	Bit 6 definition: “P3$_5$ pull-up control” Bit 7 definition: “P3$_7$ pull-up control”	Bits 6 and 7 not available
Port P1P3 control (17_{16})	Bit 0 definition: “P3$_7$/INT$_0$ input level selection” Bit 1 definition: “P3$_6$/INT$_1$ input level selection”	Bit 0 definition: “P3$_7$/INT$_0$ input level selection” Bit 1 not available	Bits 0 and 1 not available
A/DControl (34_{16})	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 ($3A_{16}$)	Bit 0 definition “INT$_0$ interrupt edge selection” Bit 1 definition “INT$_1$ interrupt edge selection” Bit 4 definition “Serial I/O1, INT$_1$ interrupt selection”	Bit 0 definition “INT$_0$ interrupt edge selection” Bits 1 and 4 not available	Bits 0, 1 and 4 not available
Interrupt request ($3C_{16}$)	Bit 1 definition “UART transmission, USB (except IN), INT$_1$” Bit 2 definition “INT$_0$”	Bit 1 definition “UART transmission, USB (except IN)” Bit 2 definition “INT$_0$”	Bit 1 definition “UART transmission, USB (except IN)” Bit 2 not available
Interrupt control ($3E_{16}$)	Bit 1 definition “UART transmission, USB (except IN), INT$_1$” Bit 2 definition “INT$_0$”	Bit 1 definition “UART transmission, USB (except IN)” Bit 2 definition “INT$_0$”	Bit 1 definition “UART transmission, USB (except IN)” Bit 2 not available

DESCRIPTION SUPPLEMENT FOR USE OF USB FUNCTION STABLY

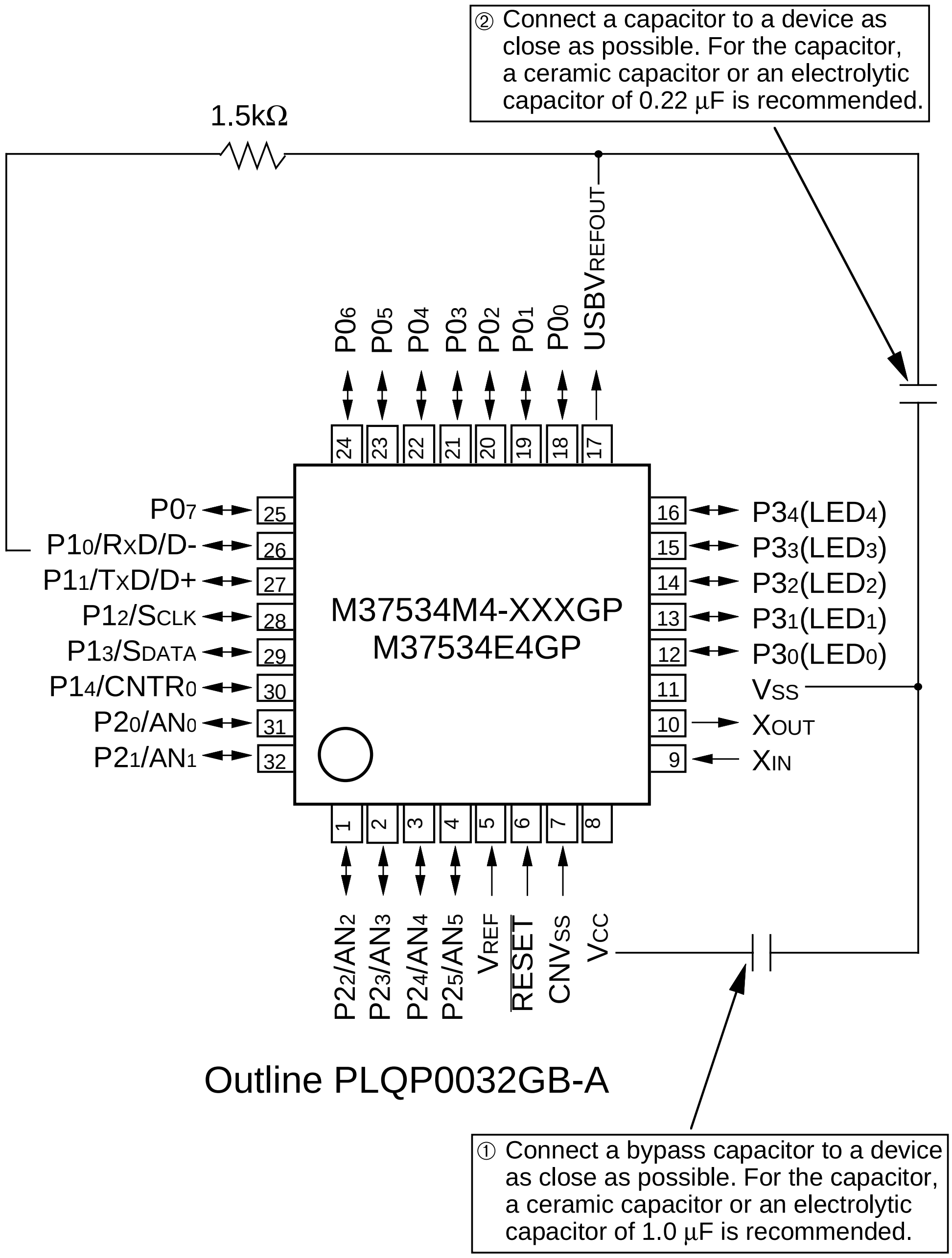


① Connect a bypass capacitor to a device as close as possible. For the capacitor, a ceramic capacitor or an electrolytic capacitor of 1.0 µF is recommended.

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 switcing 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-XXXFP, M37534E8FP

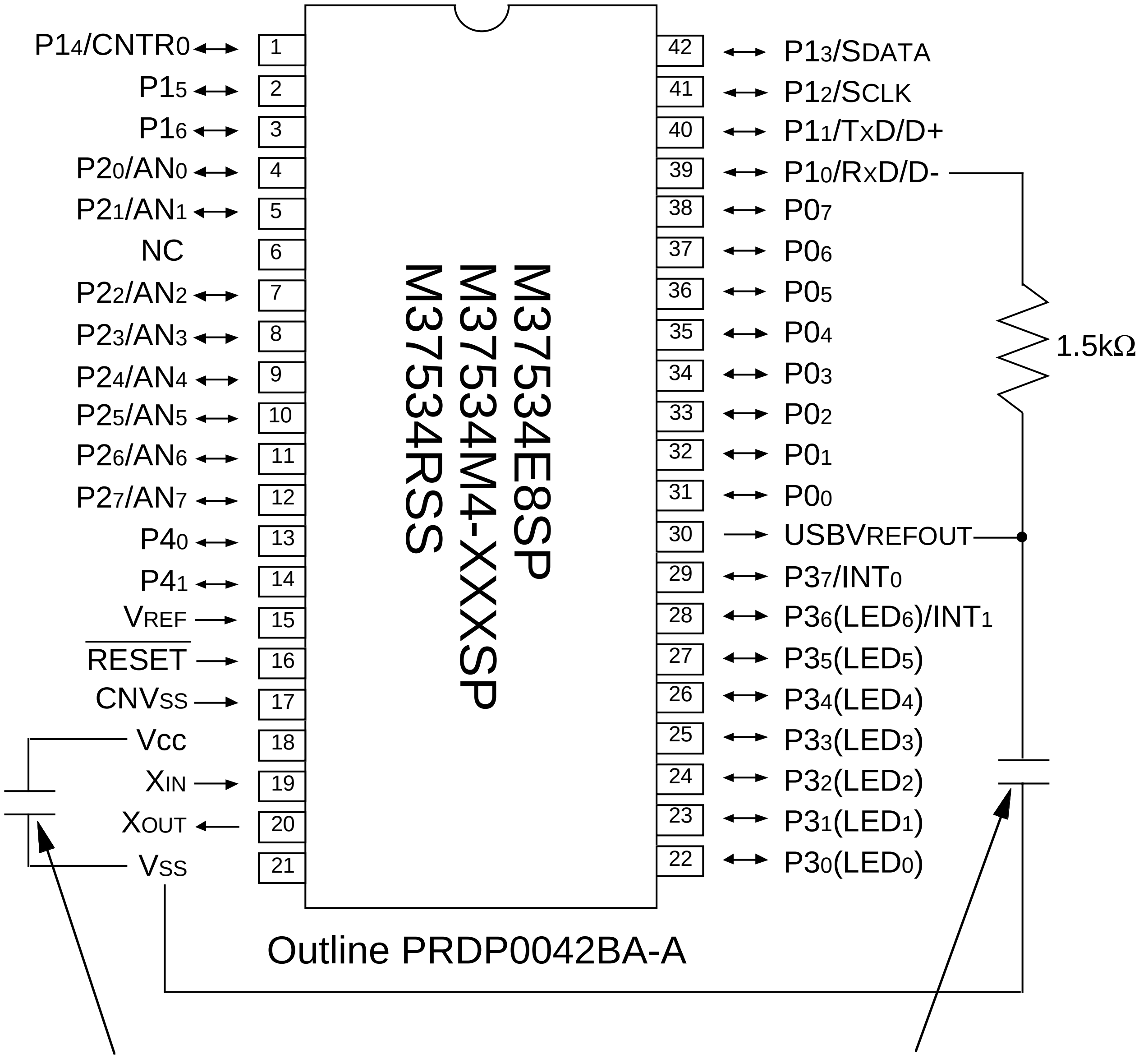


② Connect a capacitor to a device as close as possible. For the capacitor, a ceramic capacitor or an electrolytic capacitor of 0.22 μF is recommended.

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.

Reason of ① is to reduce the effect by switcing 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. 57 Handling of VCC, USBVREFOUT pins of M37534M4-XXXGP, M37534E4GP



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 switcing 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. 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

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

2.1.1 Memory map

0000_{16}	Port P0 (P0)
0001_{16}	Port P0 direction register (P0D)
0002_{16}	Port P1 (P1)
0003_{16}	Port P1 direction register (P1D)
0004_{16}	Port P2 (P2)
0005_{16}	Port P2 direction register (P2D)
0006_{16}	Port P3 (P3)
0007_{16}	Port P3 direction register (P3D)
0008_{16}	Port P4 (P4)
0009_{16}	Port P4 direction register (P4D)

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

2.1.2 Relevant registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (Pi) (i = 0 to 4) [Address : 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{16}]

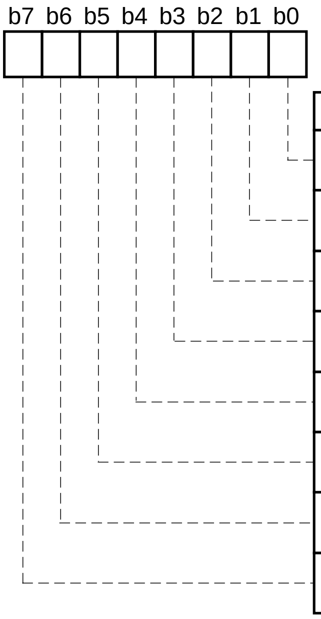
B	Name	Function	At reset	R	W
0	Port Pi_0	● In output mode Write } Port latch Read } ● In input mode Write : Port latch Read : Value of pins	?	○	○
1	Port Pi_1		?	○	○
2	Port Pi_2		?	○	○
3	Port Pi_3		?	○	○
4	Port Pi_4		?	○	○
5	Port Pi_5		?	○	○
6	Port Pi_6		?	○	○
7	Port Pi_7		?	○	○

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

- 42-pin version: Ports $P1_7$, $P4_2$–$P4_7$
- 36-pin version: Ports $P1_5$–$P1_7$, $P3_6$, $P4_0$–$P4_7$
- 32-pin version: Ports $P1_5$–$P1_7$, $P2_6$, $P2_7$, $P3_5$–$P3_7$, $P4_0$–$P4_7$

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

Port Pi direction register



Port Pi direction register (PiD) (i = 0 to 4) [Address : 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{16}]

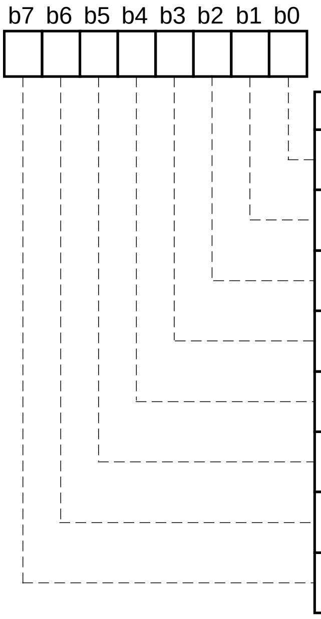
B	Name	Function	At reset	R	W
0	Port Pi direction register	0 : Port Pi_0 input mode 1 : Port Pi_0 output mode	0	×	○
1		0 : Port Pi_1 input mode 1 : Port Pi_1 output mode	0	×	○
2		0 : Port Pi_2 input mode 1 : Port Pi_2 output mode	0	×	○
3		0 : Port Pi_3 input mode 1 : Port Pi_3 output mode	0	×	○
4		0 :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

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

- 42-pin version: Ports $P1_7$, $P4_2$–$P4_7$
- 36-pin version: Ports $P1_5$–$P1_7$, $P3_6$, $P4_0$–$P4_7$
- 32-pin version: Ports $P1_5$–$P1_7$, $P2_6$, $P2_7$, $P3_5$–$P3_7$, $P4_0$–$P4_7$

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

Pull-up control register



Pull-up control register (PULL) [Address : 16_{16}]

B	Name	Function	At reset	R	W
0	$P0_0$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
1	$P0_1$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
2	$P0_2$, $P0_3$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
3	$P0_4$ – $P0_7$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
4	$P3_0$ – $P3_3$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
5	$P3_4$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
6	$P3_5$, $P3_6$ pull-up control bit (**Note 2**)	0 : Pull-up Off 1 : Pull-up On	1	○	○
7	$P3_7$ pull-up control bit (**Note 3**)	0 : Pull-up Off 1 : Pull-up On	1	○	○

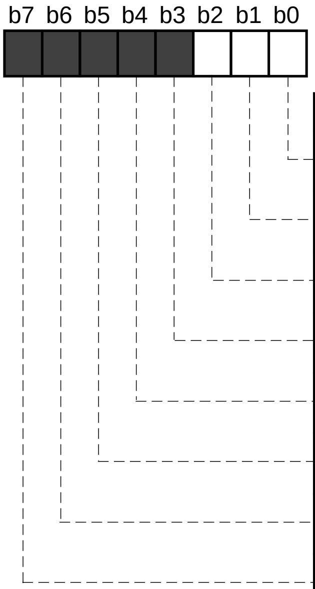
Notes 1: Pins set to output are disconnected from the pull-up control.

2: • 36-pin version: $P3_6$ is not existed.
• 32-pin version: Not used.

3: 32-pin version: Not used.

Fig. 2.1.4 Structure of Pull-up control register

Port P1P3 control register



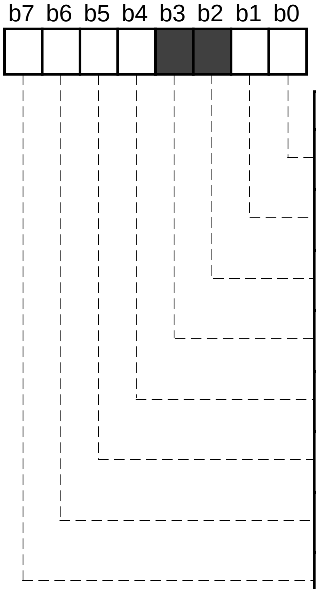
Port P1P3 control register (P1P3C) [Address : 17_{16}]

B	Name	Function	At reset	R	W
0	$\mathrm{P3_7/INT_0}$ input level selection bit (**Note 1**)	0 : CMOS level 1 : TTL level	0	○	○
1	$\mathrm{P3_6/INT_1}$ input level selection bit (**Note 2**)	0 : CMOS level 1 : TTL level	0	○	○
2	$\mathrm{P1_0}$, $\mathrm{P1_2}$,$\mathrm{P1_3}$ input level selection bit	0 : CMOS level 1 : TTL level	0	○	○
3	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are “0”.		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Notes 1: For the 32-pin version, nothing is allocated for this bit.
This is a write disabled bit.
When this bit is read out, the value is “0”.
2: For the 32-pin and 36-pin versions, nothing is allocated for this bit.
This is a write disabled bit.
When this bit is read out, the value is “0”.

Fig. 2.1.5 Structure of P1P3 control register

Interrupt edge selection register



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

B	Name	Function	At reset	R	W
0	$\mathrm{INT_0}$ interrupt edge selection bit (**Note 1**)	0 : Falling edge active 1 : Rising edge active	0	○	○
1	$\mathrm{INT_1}$ interrupt edge selection bit (**Note 2**)	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are “0”.		0	○	×
3			0	○	×
4	Serial I/O1 or $\mathrm{INT_1}$ interrupt selection bit	0 : Serial I/O1 1 : $\mathrm{INT_1}$	0	○	○
5	Timer X or key-on wake up interrupt selection bit	0 : Timer X 1 : Key-on wake up	0	○	○
6	Timer 2 or serial I/O2 interrupt selection bit	0 : Timer 2 1 : Serial I/O2	0	○	○
7	$\mathrm{CNTR_0}$ or AD converter interrupt selection bit	0 : $\mathrm{CNTR_0}$ 1 : AD converter	0	○	○

Notes 1: 32-pin version: Not used.
This is a write disabled bit. When this bit is read out, the value is “0”.
2: 36-pin and 32-pin version: Not used.
This is a write disabled bit. When this bit is read out, the value is “0”.

Fig. 2.1.6 Structure of Interrupt edge selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : 3C 16]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	UART transmit/USBSETUP/ OUT token/Reset/Suspend/ Resume/INT1 interrupt request bit (**Note 1**)	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	INT0 interrupt request bit (**Note 2**)	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Timer X or key-on wake up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer 2 or serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	CNTR0 or AD converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	✕

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

Notes 1: 36-pin version and 32-pin version: INT1 interrupt does not exist.

2: 32-pin version: INT0 interrupt does not exist.
This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.1.7 Structure of Interrupt request register 1

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1 (ICON1) [Address : 3E 16]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	UART transmit/USBSETUP/ OUT token/Reset/Suspend/ Resume/INT1 interrupt enable bit (**Note 1**)	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT0 interrupt enable bit (**Note 2**)	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Timer X or key-on wake up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer 2 or serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	CNTR0 or AD converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Nothing is allocated for this bit. Do not write "1" to this bit. When this bit is read out, the value is "0".		0	○	✕

Notes 1: 36-pin version and 32-pin version: INT1 interrupt does not exist.

2: 32-pin version: INT0 interrupt does not exist.
This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.1.8 Structure of Interrupt control register 1

2.1.3 Application example of key-on wake up

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

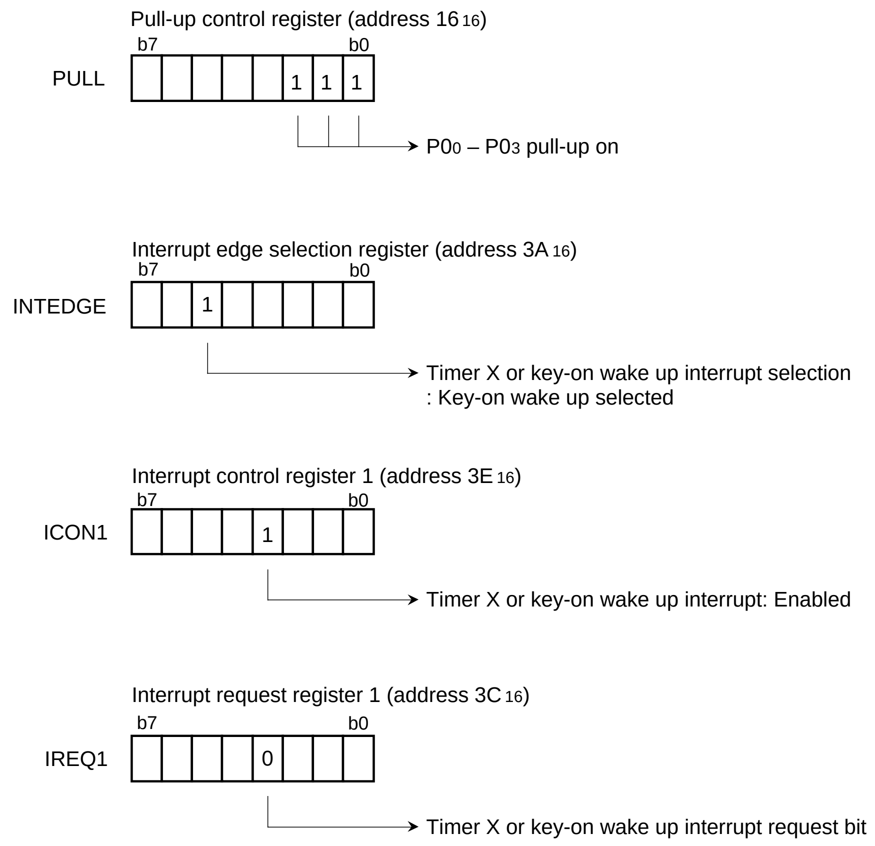

Pull-up control register (address 16 16)
b7
b0
PULL
1 1 1
P00 – P03 pull-up on
Interrupt edge selection register (address 3A 16)
b7
b0
INTEDGE
1
Timer X or key-on wake up interrupt selection
: Key-on wake up selected
Interrupt control register 1 (address 3E 16)
b7
b0
ICON1
1
Timer X or key-on wake up interrupt: Enabled
Interrupt request register 1 (address 3C 16)
b7
b0
IREQ1
0
Timer X or key-on wake up interrupt request bit

Fig. 2.1.9 Relevant registers setting

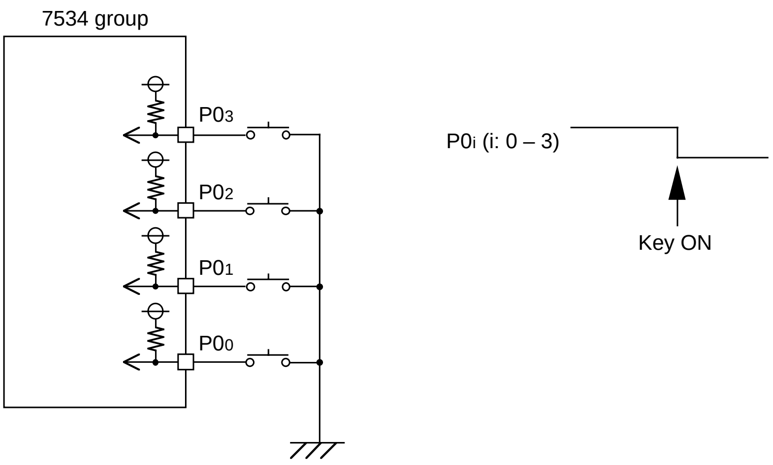


Fig. 2.1.10 Application circuit example

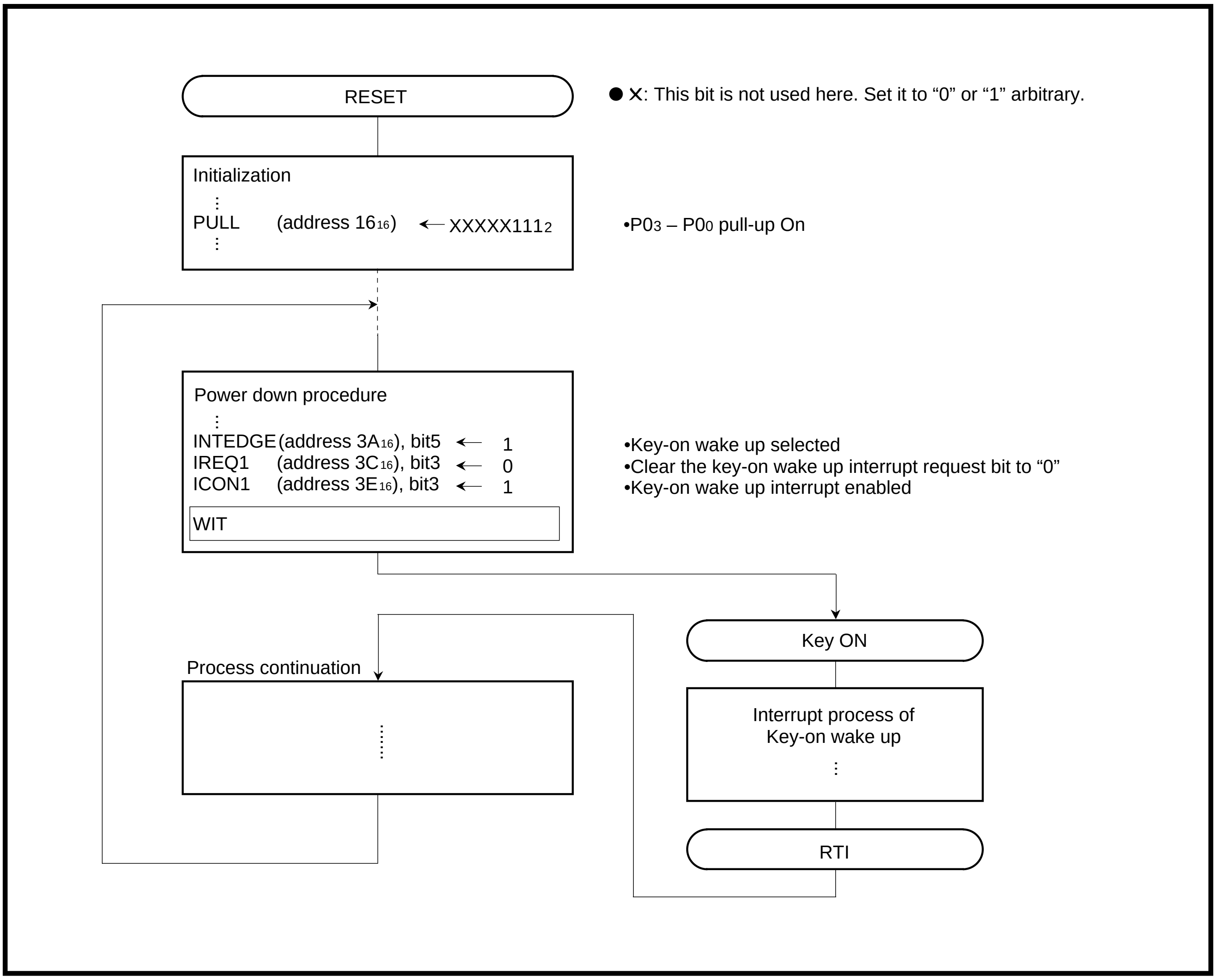

RESET
● X: This bit is not used here. Set it to “0” or “1” arbitrary.
Initialization
PULL (address 16₁₆) ← XXXXX111₂
•P0₃ – P0₀ pull-up On
Power down procedure
INTEDGE (address 3A₁₆), bit5 ← 1
IREQ1 (address 3C₁₆), bit3 ← 0
ICON1 (address 3E₁₆), bit3 ← 1
WIT
•Key-on wake up selected
•Clear the key-on wake up interrupt request bit to “0”
•Key-on wake up interrupt enabled
Key ON
Interrupt process of Key-on wake up
RTI
Process continuation

Fig. 2.1.11 Control procedure

2.1.4 Handling of unused pins

Table 2.1.1 Handling of unused pins

Pins/Ports name	Handling
P0, P1, P2, P3, P4	•Set to the input mode and connect each to Vcc or Vss through a resistor of 1 kΩ to 10 kΩ. •Set to the output mode and open at “L” or “H” level.
V_{REF}	•Connect to Vss (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 V_{CC}) or pull-down (connect the port to V_{SS}) 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.

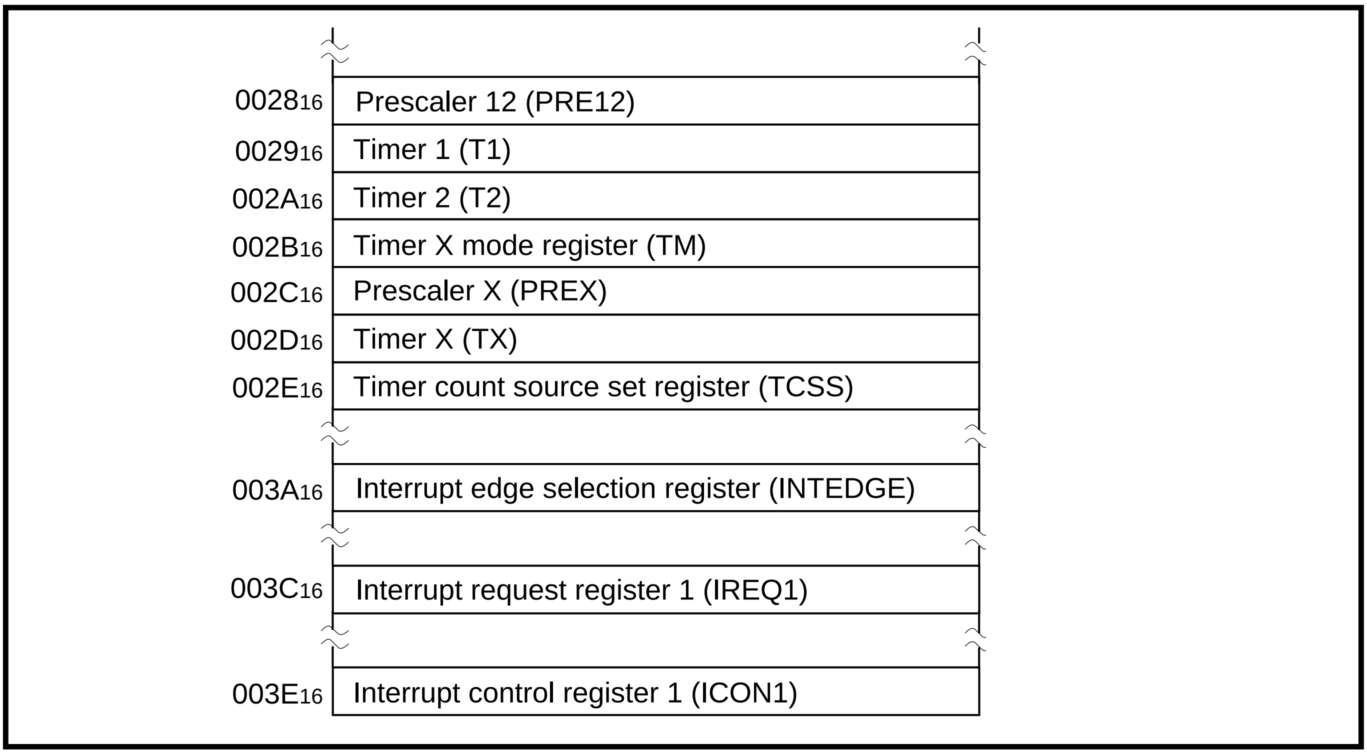
● **Reason**
If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between ports.

- 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.

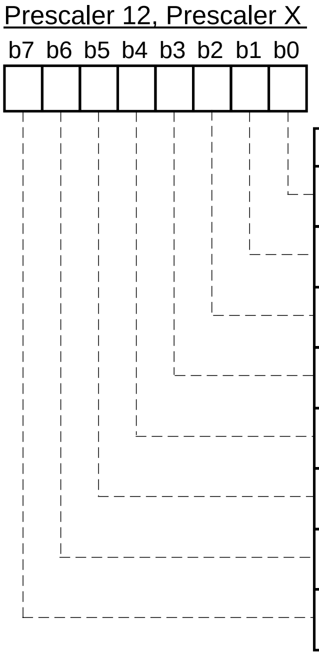
2.2.1 Memory map



0028_{16}	Prescaler 12 (PRE12)
0029_{16}	Timer 1 (T1)
$002A_{16}$	Timer 2 (T2)
$002B_{16}$	Timer X mode register (TM)
$002C_{16}$	Prescaler X (PREX)
$002D_{16}$	Timer X (TX)
$002E_{16}$	Timer count source set register (TCSS)
$003A_{16}$	Interrupt edge selection register (INTEDGE)
$003C_{16}$	Interrupt request register 1 (IREQ1)
$003E_{16}$	Interrupt control register 1 (ICON1)

Fig. 2.2.1 Memory map of registers relevant to timers

2.2.2 Relevant registers

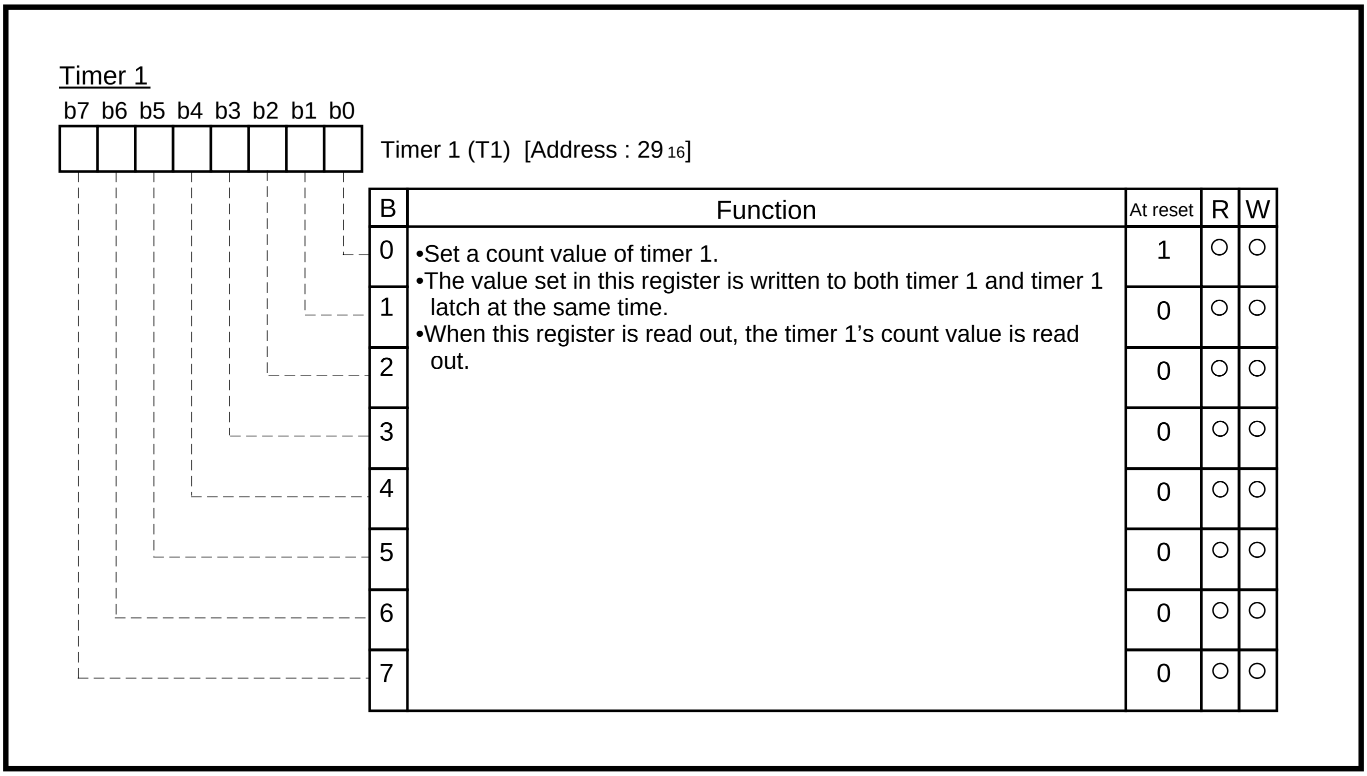


Prescaler 12, Prescaler X

Prescaler 12 (PRE12) [Address : 28_{16}]
Prescaler X (PREX) [Address : $2C_{16}$]

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

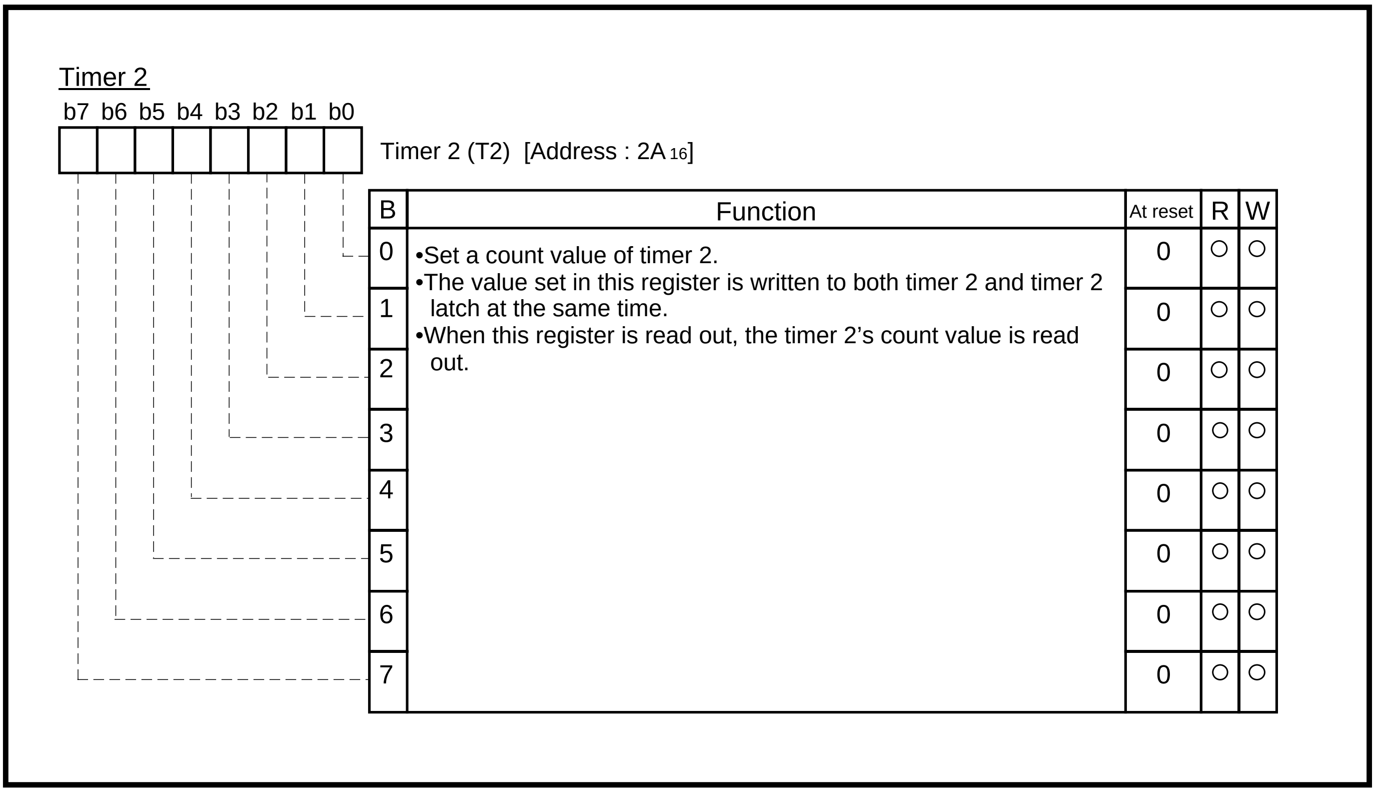
Fig. 2.2.2 Structure of Prescaler 12, Prescaler X



Timer 1 (T1) [Address : 29_{16}]

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

Fig. 2.2.3 Structure of Timer 1

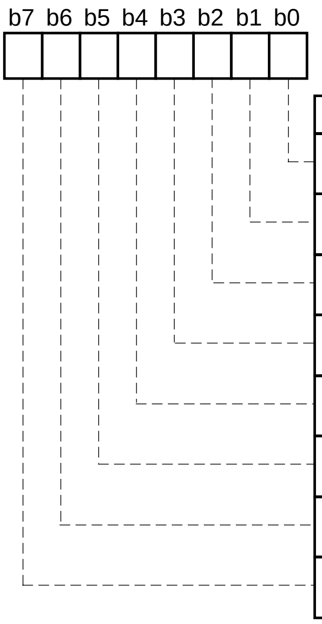


Timer 2 (T2) [Address : $2A_{16}$]

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

Fig. 2.2.4 Structure of Timer 2

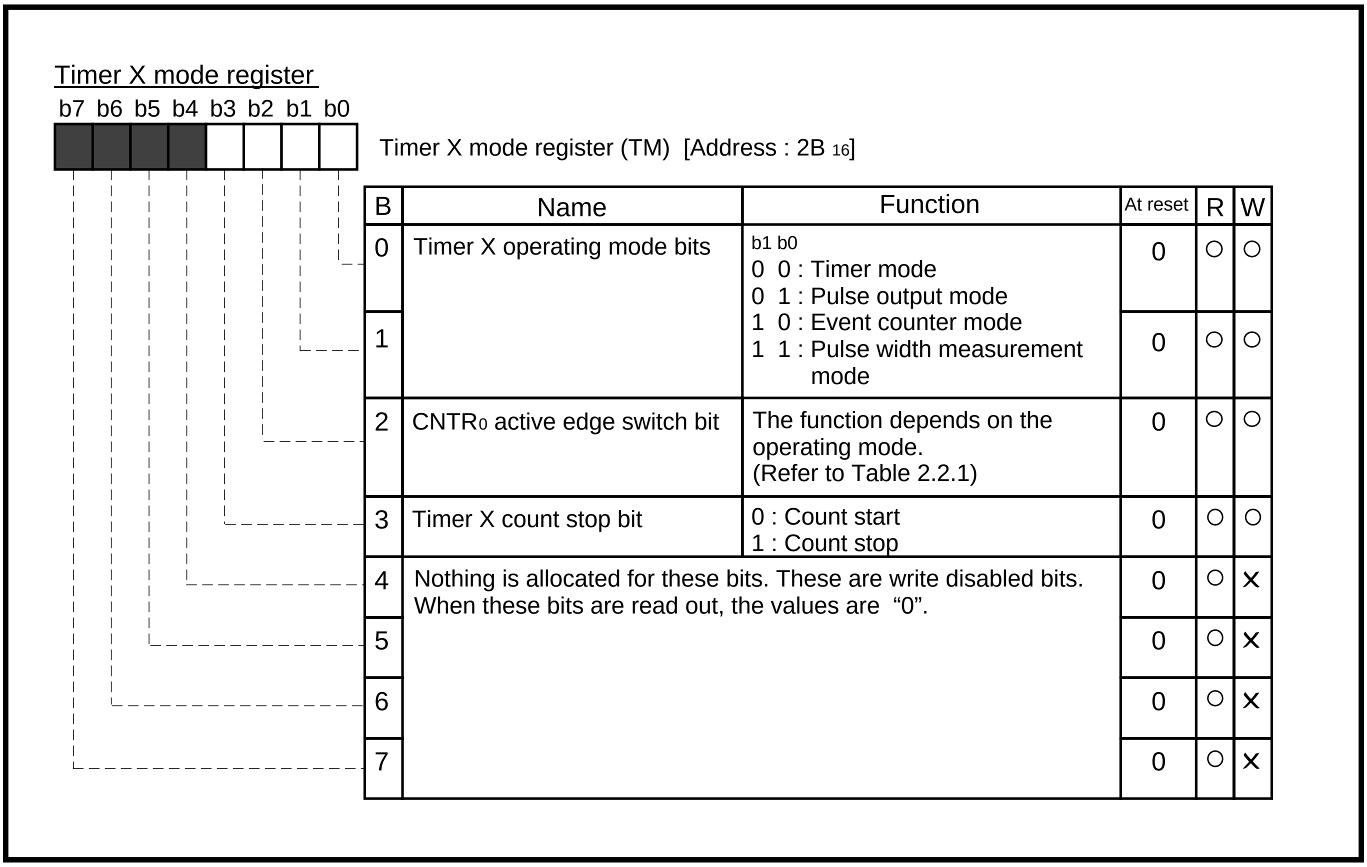
Timer X



Timer X (TX) [Address : $2D_{16}$]

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

Fig. 2.2.5 Structure of Timer X



Timer X mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer X mode register (TM) [Address : 2B $_{16}$]

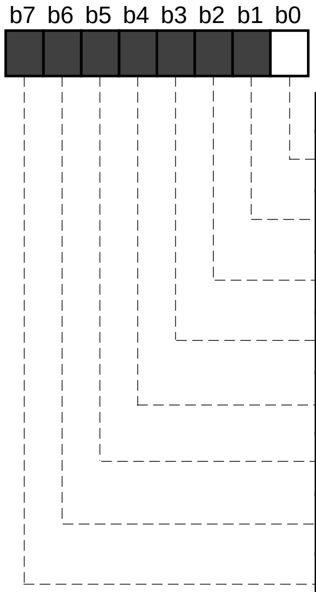
B	Name	Function	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
1			0	○	○
2	CNTR $_0$ active edge switch bit	The function depends on the operating mode. (Refer to Table 2.2.1)	0	○	○
3	Timer X count stop bit	0 : Count start 1 : Count stop	0	○	○
4	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 2.2.6 Structure of Timer X mode register

Table 2.2.1 CNTR $_0$ 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

Timer count source set register



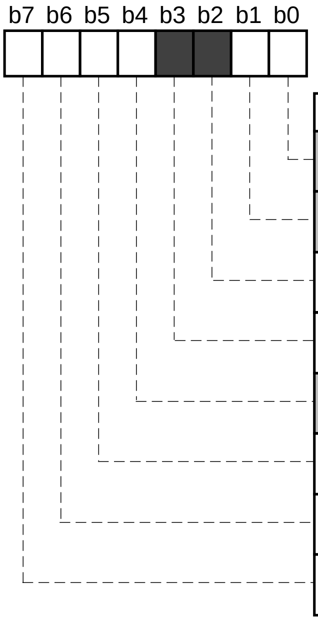
Timer count source set register (TCSS) [Address : 2E 16]

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

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

Fig. 2.2.7 Structure of Timer count source set register

Interrupt edge selection register



Interrupt edge selection register (INTEDGE) [Address : 3A 16]

B	Name	Function	At reset	R	W
0	INT0 interrupt edge selection bit (**Note 1**)	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT1 interrupt edge selection bit (**Note 2**)	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
3			0	○	×
4	Serial I/O1 or INT1 interrupt selection bit	0 : Serial I/O1 1 : INT1	0	○	○
5	Timer X or key-on wake up interrupt selection bit	0 : Timer X 1 : Key-on wake up	0	○	○
6	Timer 2 or serial I/O2 interrupt selection bit	0 : Timer 2 1 : Serial I/O2	0	○	○
7	CNTR0 or AD converter interrupt selection bit	0 : CNTR0 1 : AD converter	0	○	○

Notes 1: 32-pin version: Not used.
This is a write disabled bit. When this bit is read out, the value is "0".
2: 36-pin and 32-pin version: Not used.
This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.2.8 Structure of Interrupt edge selection register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : 3C$_{16}$]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	UART transmit/USBSETUP/ OUT token/Reset/Suspend/ Resume/INT$_1$ interrupt request bit (**Note 1**)	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	INT$_0$ interrupt request bit (**Note 2**)	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Timer X or key-on wake up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer 2 or serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	CNTR$_0$ or AD converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is “0”.		0	○	×

✻: These bits can be cleared to “0” by program, but cannot be set.

Notes 1: 36-pin version and 32-pin version: INT$_1$ interrupt does not exist.

2: 32-pin version: INT$_0$ interrupt does not exist.
This is a write disabled bit. When this bit is read out, the value is “0”.

Fig. 2.2.9 Structure of Interrupt request register 1

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

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

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	UART transmit/USBSETUP/ OUT token/Reset/Suspend/ Resume/INT$_1$ interrupt enable bit (**Note 1**)	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT$_0$ interrupt enable bit (**Note 2**)	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Timer X or key-on wake up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer 2 or serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	CNTR$_0$ or AD converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Nothing is allocated for this bit. Do not write “1” to this bit. When this bit is read out, the value is “0”.		0	○	×

Notes 1: 36-pin version and 32-pin version: INT$_1$ interrupt does not exist.

2: 32-pin version: INT$_0$ interrupt does not exist.
This is a write disabled bit. When this bit is read out, the value is “0”.

Fig. 2.2.10 Structure of Interrupt control register 1

2.2.3 Timer application examples

(1) Basic functions and uses

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

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

<Use>

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

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

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

<Use>

- •Generation of cyclic interrupts
- •Clock function (measurement of 100 ms); see Application example 1
- •Control of a main routine cycle

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

The output level of the $CNTR_0$ pin is inverted each time the timer underflows (in the pulse output mode).

<Use>

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

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

External pulses input to the $CNTR_0$ pin are counted as the timer count source (in the event counter mode).

<Use>

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

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

The “H” or “L” level width of external pulses input to $CNTR_0$ 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.

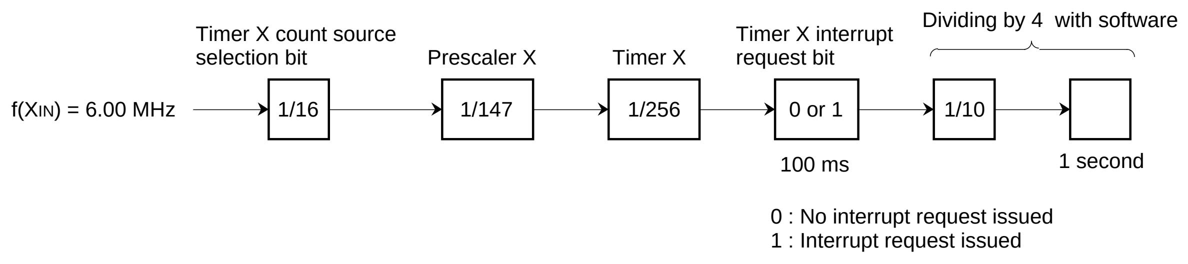

Timer X count source selection bit
Prescaler X
Timer X
Timer X interrupt request bit
Dividing by 4 with software
f(XIN) = 6.00 MHz
1/16
1/147
1/256
0 or 1
1/10
100 ms
1 second
0 : No interrupt request issued
1 : Interrupt request issued

Fig. 2.2.11 Timers connection and setting of division ratios

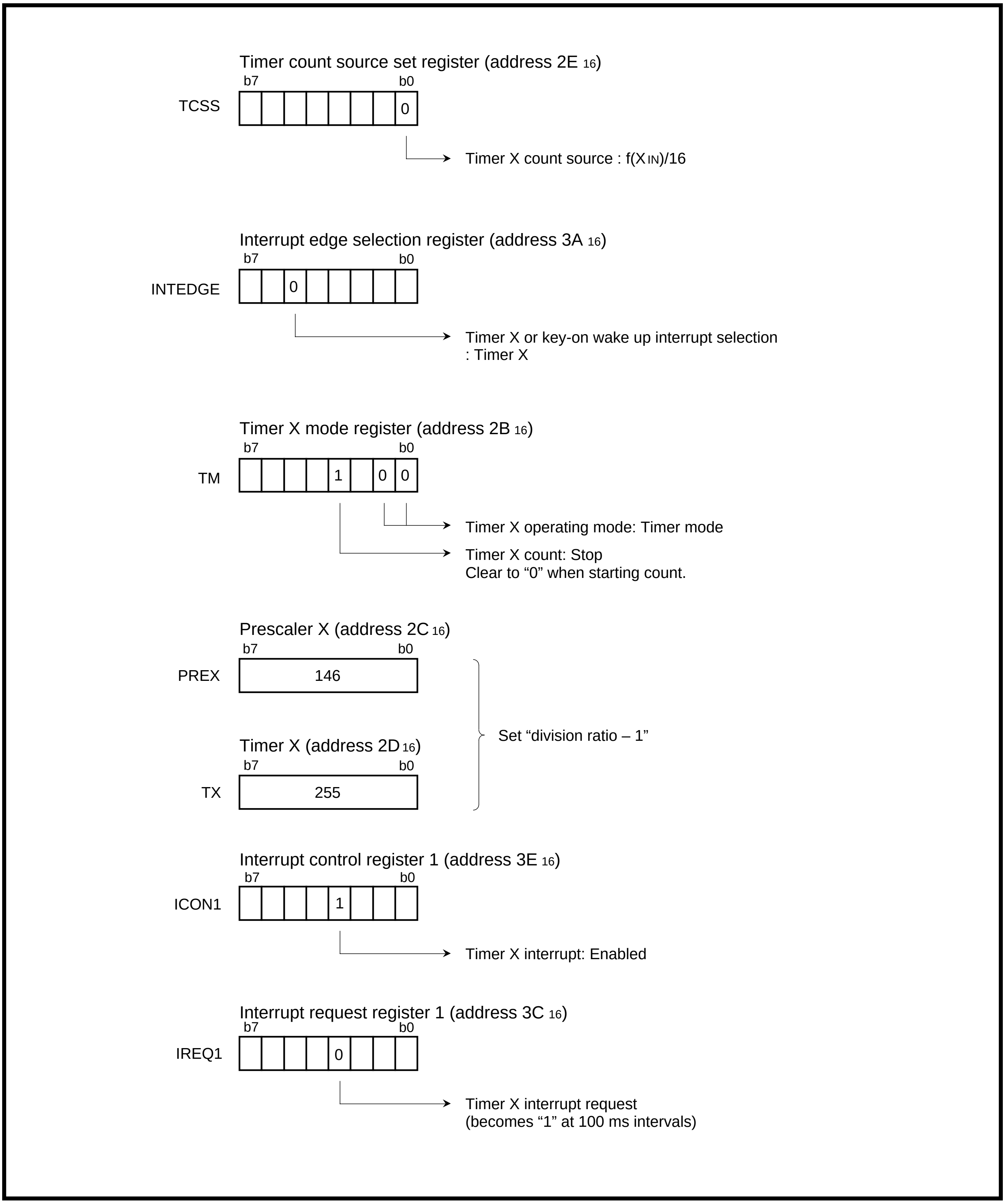

Timer count source set register (address 2E 16)
b7
b0
TCSS
0
Timer X count source : f(X IN)/16
Interrupt edge selection register (address 3A 16)
b7
b0
INTEDGE
0
Timer X or key-on wake up interrupt selection
: Timer X
Timer X mode register (address 2B 16)
b7
b0
TM
1
0
0
Timer X operating mode: Timer mode
Timer X count: Stop
Clear to “0” when starting count.
Prescaler X (address 2C 16)
b7
b0
PREX
146
Set “division ratio – 1”
Timer X (address 2D 16)
b7
b0
TX
255
Interrupt control register 1 (address 3E 16)
b7
b0
ICON1
1
Timer X interrupt: Enabled
Interrupt request register 1 (address 3C 16)
b7
b0
IREQ1
0
Timer X interrupt request
(becomes “1” at 100 ms intervals)

Fig. 2.2.12 Relevant registers setting

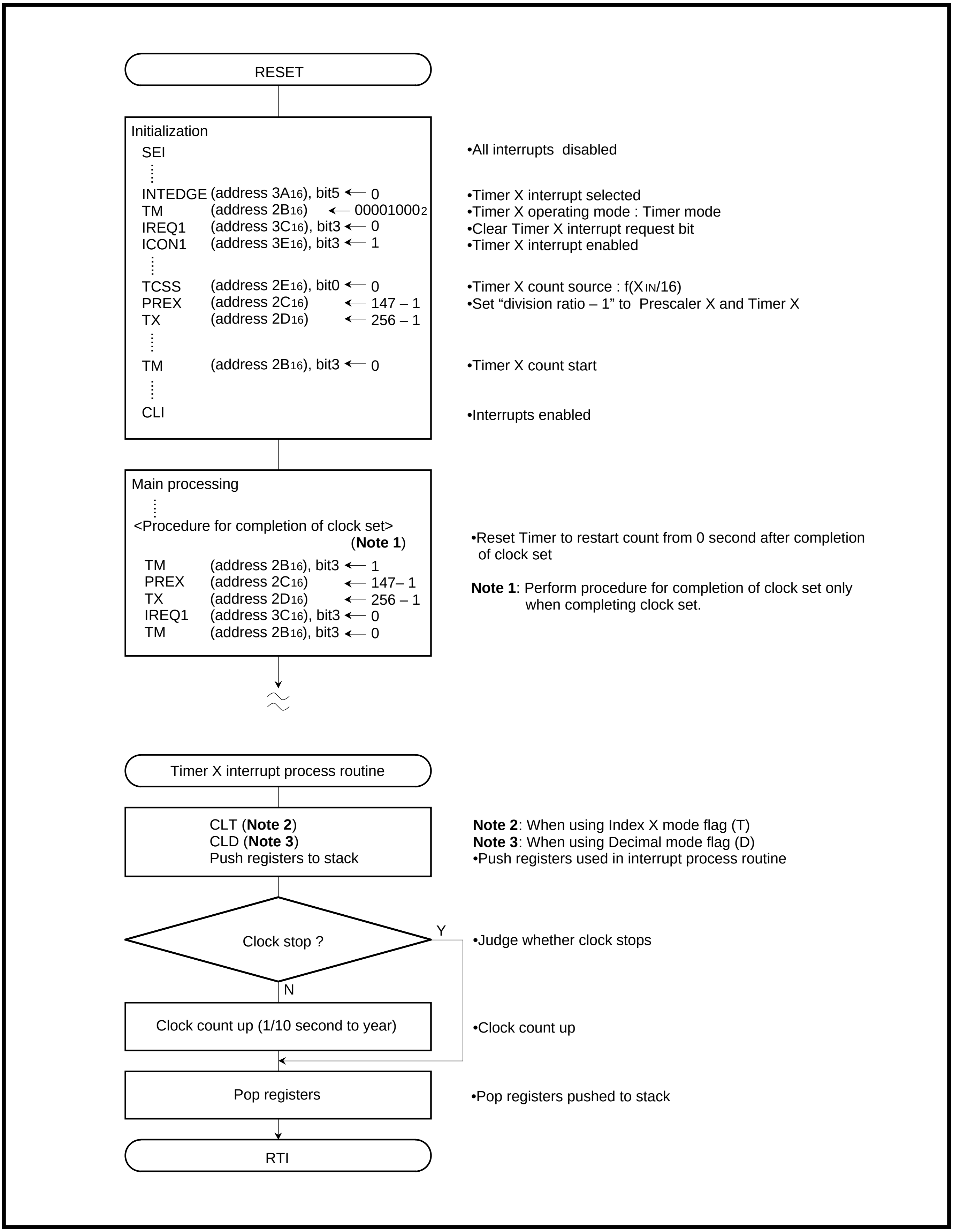

RESET
Initialization
SEI
INTEDGE (address 3A16), bit5 ← 0
TM (address 2B16) ← 000010002
IREQ1 (address 3C16), bit3 ← 0
ICON1 (address 3E16), bit3 ← 1
TCSS (address 2E16), bit0 ← 0
PREX (address 2C16) ← 147 – 1
TX (address 2D16) ← 256 – 1
TM (address 2B16), bit3 ← 0
CLI
•All interrupts disabled
•Timer X interrupt selected
•Timer X operating mode : Timer mode
•Clear Timer X interrupt request bit
•Timer X interrupt enabled
•Timer X count source : f(XIN/16)
•Set "division ratio – 1" to Prescaler X and Timer X
•Timer X count start
•Interrupts enabled
Main processing
<Procedure for completion of clock set>
(Note 1)
TM (address 2B16), bit3 ← 1
PREX (address 2C16) ← 147– 1
TX (address 2D16) ← 256 – 1
IREQ1 (address 3C16), bit3 ← 0
TM (address 2B16), bit3 ← 0
•Reset Timer to restart count from 0 second after completion of clock set
Note 1: Perform procedure for completion of clock set only when completing clock set.
Timer X interrupt process routine
CLT (Note 2)
CLD (Note 3)
Push registers to stack
Note 2: When using Index X mode flag (T)
Note 3: When using Decimal mode flag (D)
•Push registers used in interrupt process routine
Clock stop ?
Y
N
•Judge whether clock stops
Clock count up (1/10 second to year)
•Clock count up
Pop registers
•Pop registers pushed to stack
RTI

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 $P1_4/CNTR_0$ pin.
•The level of the $P1_4/CNTR_0$ 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.

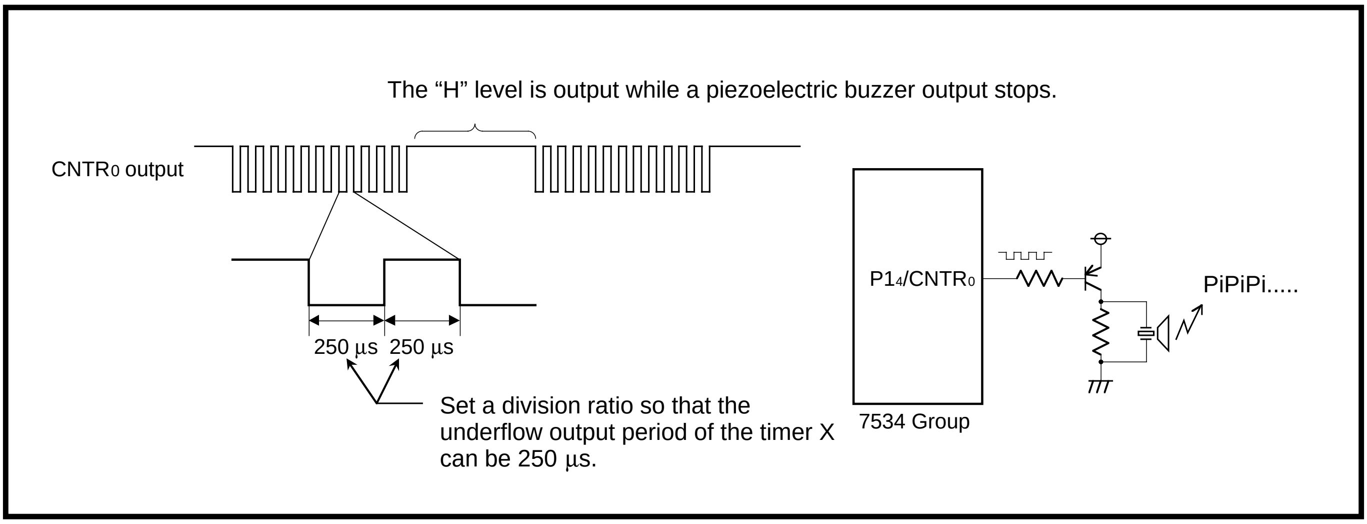

The "H" level is output while a piezoelectric buzzer output stops.
CNTR0 output
250 µs
250 µs
Set a division ratio so that the underflow output period of the timer X can be 250 µs.
P14/CNTR0
7534 Group
PiPiPi.....

Fig. 2.2.14 Peripheral circuit example

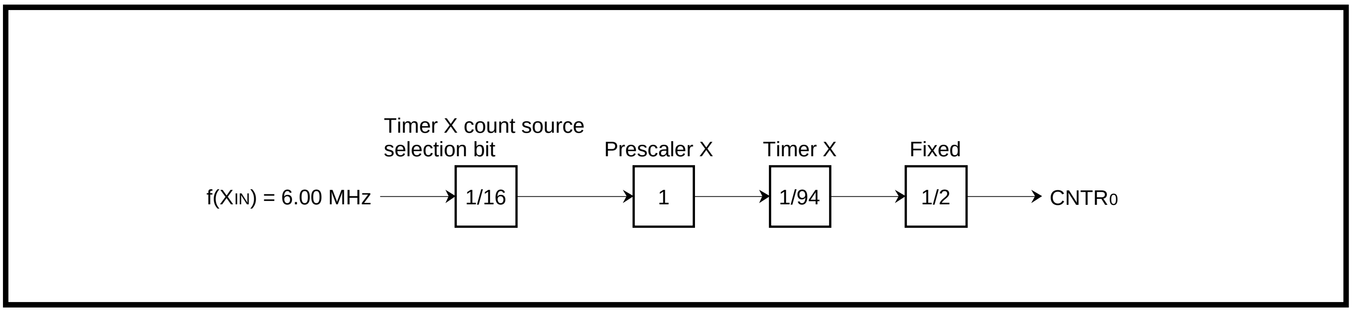


Fig. 2.2.15 Timers connection and setting of division ratios

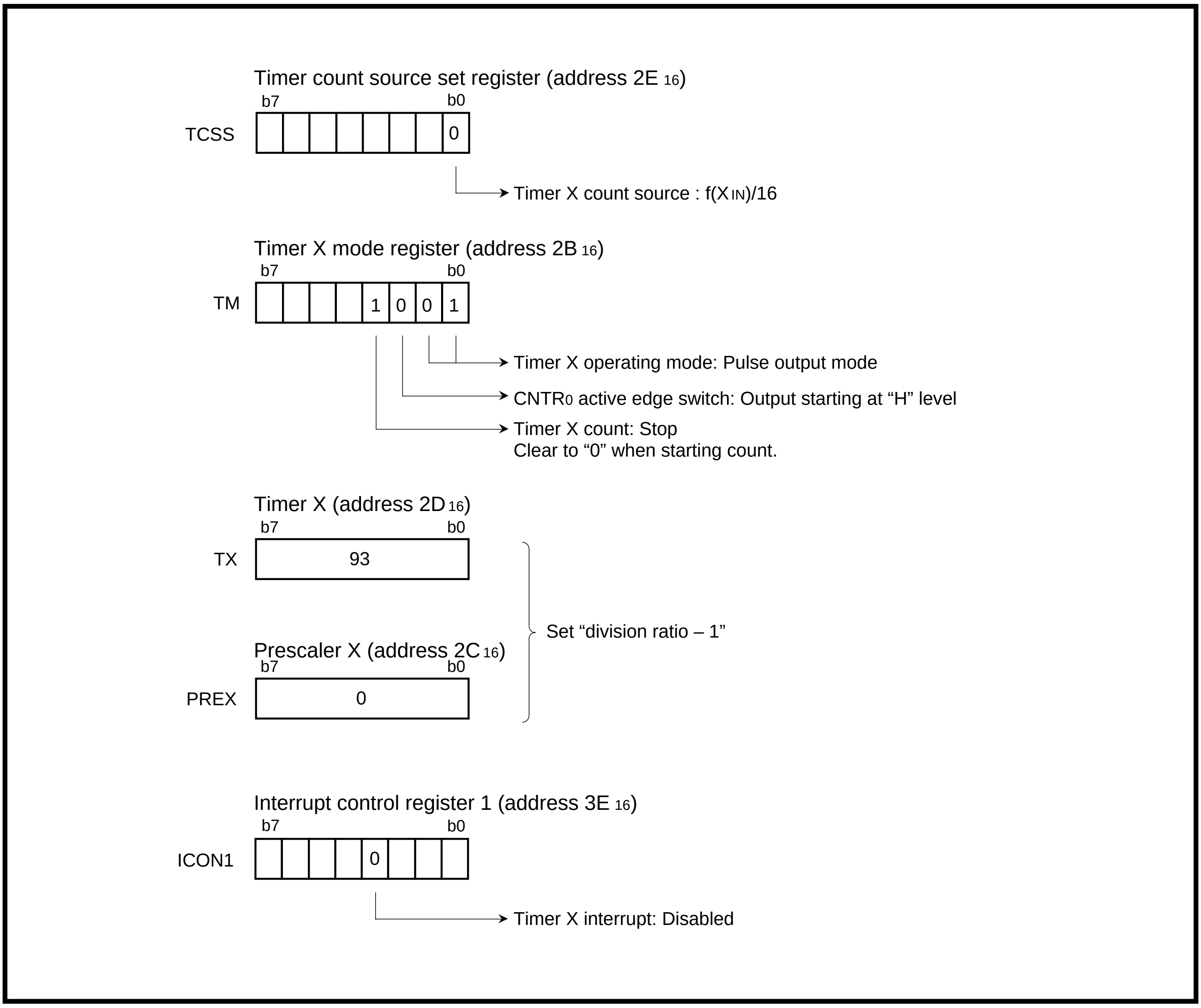

Timer count source set register (address 2E16)
b7
b0
TCSS
0
Timer X count source : f(XIN)/16
Timer X mode register (address 2B16)
b7
b0
TM
1
0
0
1
Timer X operating mode: Pulse output mode
CNTR0 active edge switch: Output starting at “H” level
Timer X count: Stop
Clear to “0” when starting count.
Timer X (address 2D16)
b7
b0
TX
93
Set “division ratio – 1”
Prescaler X (address 2C16)
b7
b0
PREX
0
Interrupt control register 1 (address 3E16)
b7
b0
ICON1
0
Timer X interrupt: Disabled

Fig. 2.2.16 Relevant registers setting

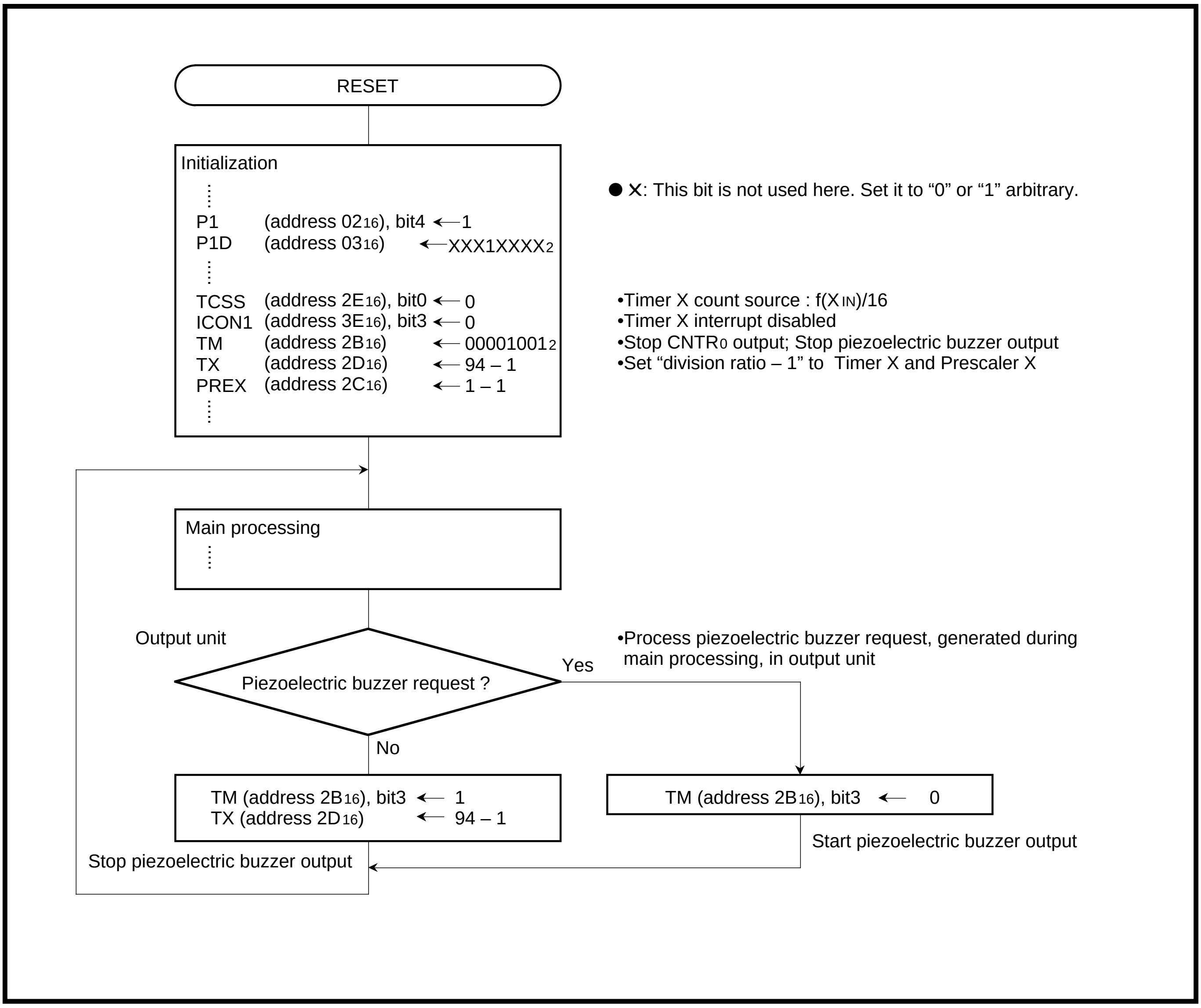

RESET
Initialization
P1 (address 02$_{16}$), bit4 ← 1
P1D (address 03$_{16}$) ← XXX1XXXX$_2$
TCSS (address 2E$_{16}$), bit0 ← 0
ICON1 (address 3E$_{16}$), bit3 ← 0
TM (address 2B$_{16}$) ← 00001001$_2$
TX (address 2D$_{16}$) ← 94 – 1
PREX (address 2C$_{16}$) ← 1 – 1
● X: This bit is not used here. Set it to “0” or “1” arbitrary.
•Timer X count source : f(X$_{IN}$)/16
•Timer X interrupt disabled
•Stop CNTR$_0$ output; Stop piezoelectric buzzer output
•Set “division ratio – 1” to Timer X and Prescaler X
Main processing
Output unit
Piezoelectric buzzer request ?
Yes
No
•Process piezoelectric buzzer request, generated during main processing, in output unit
TM (address 2B$_{16}$), bit3 ← 1
TX (address 2D$_{16}$) ← 94 – 1
Stop piezoelectric buzzer output
TM (address 2B$_{16}$), bit3 ← 0
Start piezoelectric buzzer output

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 $P1_4/CNTR_0$ pin with the timer.
- A reference value

Specifications:

- The pulse is input to the $P1_4/CNTR_0$ 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.

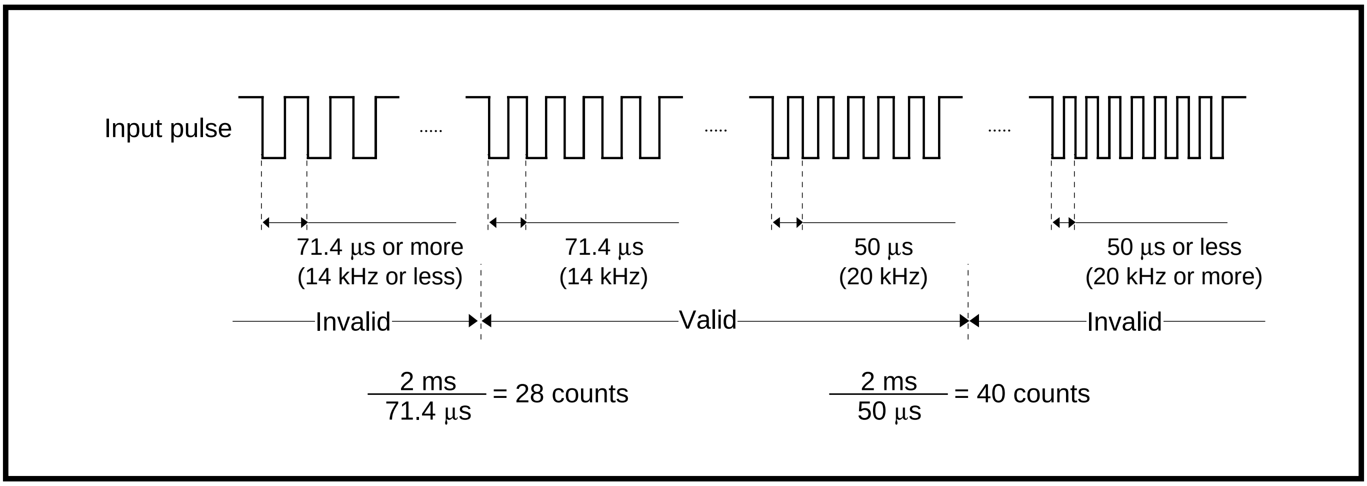


Fig. 2.2.18 Judgment method of valid/invalid of input pulses

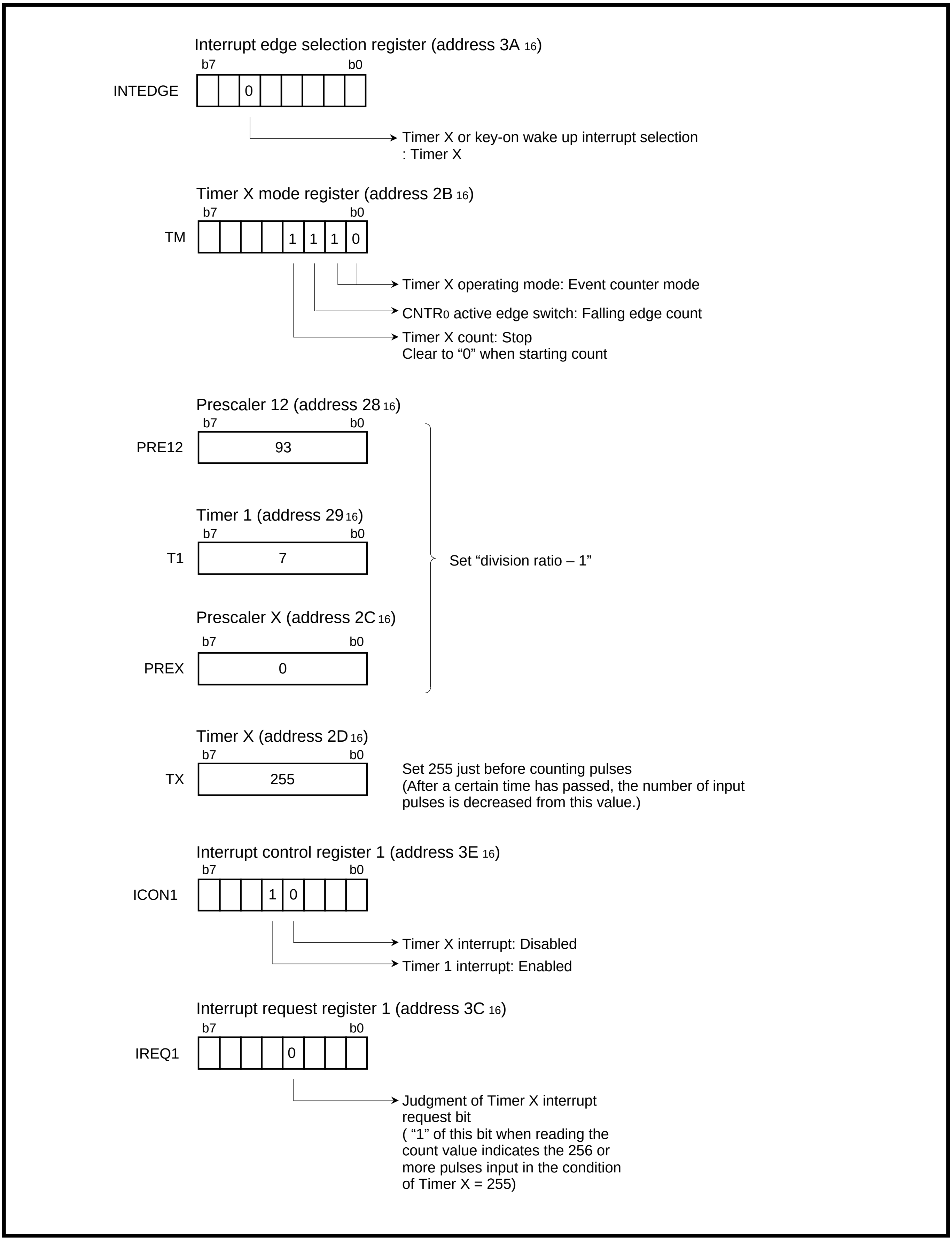

Interrupt edge selection register (address 3A 16)
b7
b0
INTEDGE
0
Timer X or key-on wake up interrupt selection
: Timer X
Timer X mode register (address 2B 16)
b7
b0
TM
1 1 1 0
Timer X operating mode: Event counter mode
CNTR0 active edge switch: Falling edge count
Timer X count: Stop
Clear to "0" when starting count
Prescaler 12 (address 28 16)
PRE12
93
Timer 1 (address 29 16)
T1
7
Prescaler X (address 2C 16)
PREX
0
Set "division ratio – 1"
Timer X (address 2D 16)
TX
255
Set 255 just before counting pulses
(After a certain time has passed, the number of input
pulses is decreased from this value.)
Interrupt control register 1 (address 3E 16)
ICON1
1 0
Timer X interrupt: Disabled
Timer 1 interrupt: Enabled
Interrupt request register 1 (address 3C 16)
IREQ1
0
Judgment of Timer X interrupt
request bit
("1" of this bit when reading the
count value indicates the 256 or
more pulses input in the condition
of Timer X = 255)

Fig. 2.2.19 Relevant registers setting

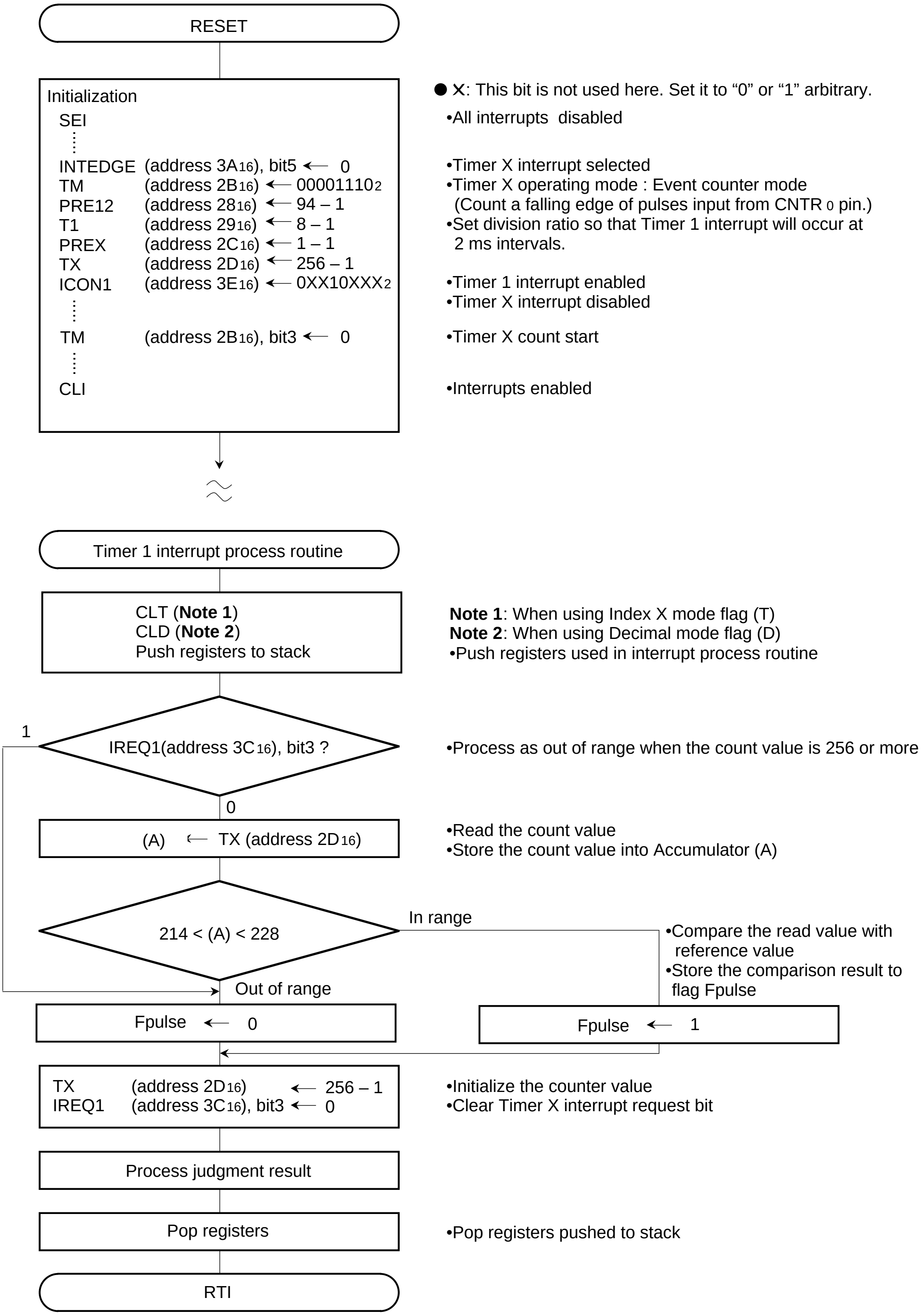

RESET
Initialization
SEI
INTEDGE (address 3A16), bit5 ← 0
TM (address 2B16) ← 000011102
PRE12 (address 2816) ← 94 – 1
T1 (address 2916) ← 8 – 1
PREX (address 2C16) ← 1 – 1
TX (address 2D16) ← 256 – 1
ICON1 (address 3E16) ← 0XX10XXX2
TM (address 2B16), bit3 ← 0
CLI
● X: This bit is not used here. Set it to “0” or “1” arbitrary.
•All interrupts disabled
•Timer X interrupt selected
•Timer X operating mode : Event counter mode
(Count a falling edge of pulses input from CNTR 0 pin.)
•Set division ratio so that Timer 1 interrupt will occur at 2 ms intervals.
•Timer 1 interrupt enabled
•Timer X interrupt disabled
•Timer X count start
•Interrupts enabled
Timer 1 interrupt process routine
CLT (Note 1)
CLD (Note 2)
Push registers to stack
Note 1: When using Index X mode flag (T)
Note 2: When using Decimal mode flag (D)
•Push registers used in interrupt process routine
IREQ1(address 3C16), bit3 ?
1
0
•Process as out of range when the count value is 256 or more
(A) ← TX (address 2D16)
•Read the count value
•Store the count value into Accumulator (A)
214 < (A) < 228
In range
Out of range
•Compare the read value with reference value
•Store the comparison result to flag Fpulse
Fpulse ← 0
Fpulse ← 1
TX (address 2D16) ← 256 – 1
IREQ1 (address 3C16), bit3 ← 0
•Initialize the counter value
•Clear Timer X interrupt request bit
Process judgment result
Pop registers
•Pop registers pushed to stack
RTI

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 $P1_4/CNTR_0$ pin. An underflow is detected by the timer X interrupt and an end of the input pulse "H" level is detected by the $CNTR_0$ interrupt.

Specifications: •The timer X counts the "H" level width of the FG pulse input to the $P1_4/CNTR_0$ 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 $FFFF_{16}$ to 0000_{16}.

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.

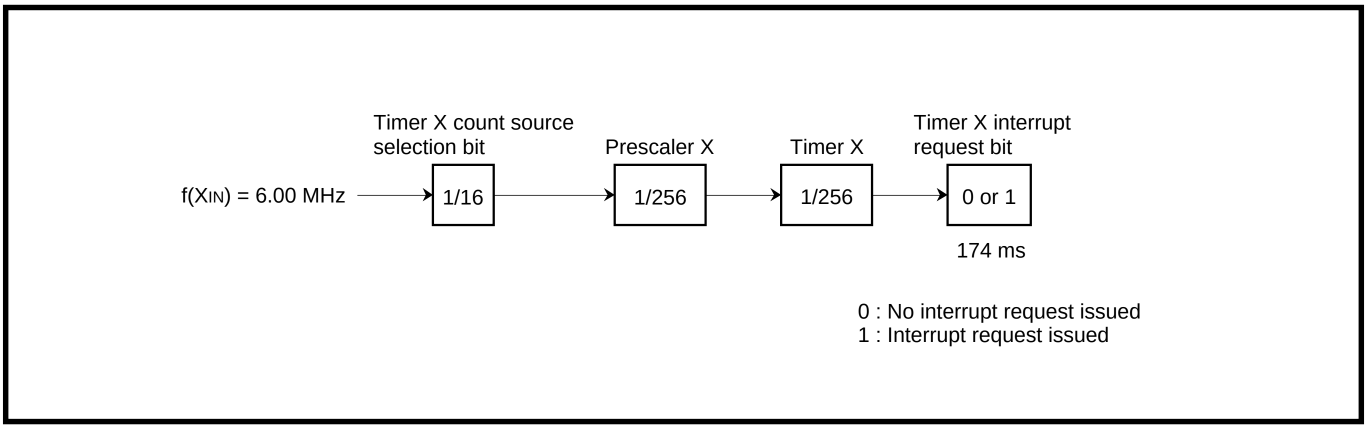

Timer X count source selection bit
Prescaler X
Timer X
Timer X interrupt request bit
f(XIN) = 6.00 MHz
1/16
1/256
1/256
0 or 1
174 ms
0 : No interrupt request issued
1 : Interrupt request issued

Fig. 2.2.21 Timers connection and setting of division ratios

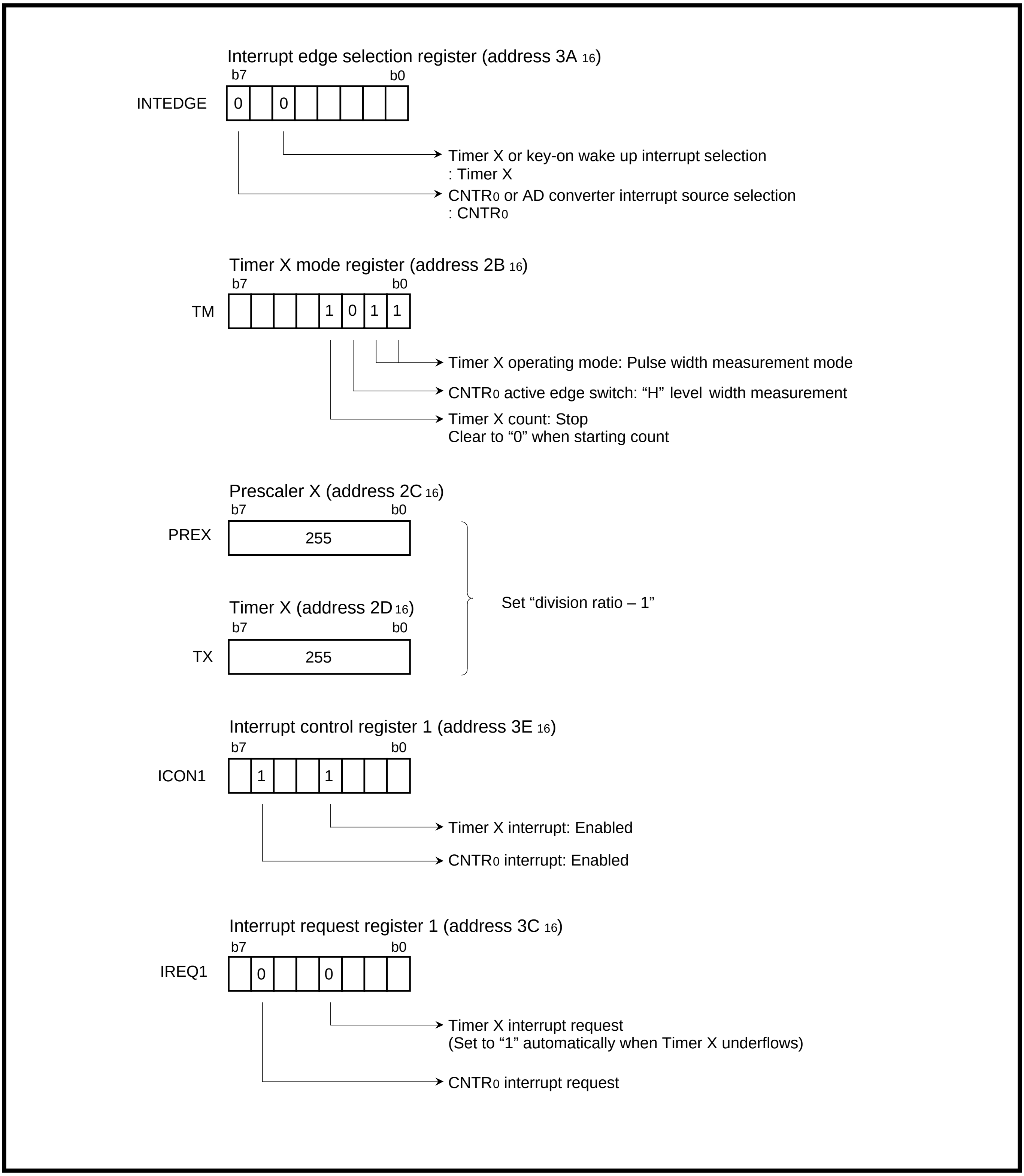

Interrupt edge selection register (address 3A16)
b7
b0
INTEDGE
0
0
Timer X or key-on wake up interrupt selection
: Timer X
CNTR0 or AD converter interrupt source selection
: CNTR0
Timer X mode register (address 2B16)
b7
b0
TM
1
0
1
1
Timer X operating mode: Pulse width measurement mode
CNTR0 active edge switch: "H" level width measurement
Timer X count: Stop
Clear to "0" when starting count
Prescaler X (address 2C16)
b7
b0
PREX
255
Timer X (address 2D16)
b7
b0
TX
255
Set "division ratio – 1"
Interrupt control register 1 (address 3E16)
b7
b0
ICON1
1
1
Timer X interrupt: Enabled
CNTR0 interrupt: Enabled
Interrupt request register 1 (address 3C16)
b7
b0
IREQ1
0
0
Timer X interrupt request
(Set to "1" automatically when Timer X underflows)
CNTR0 interrupt request

Fig. 2.2.22 Relevant registers setting

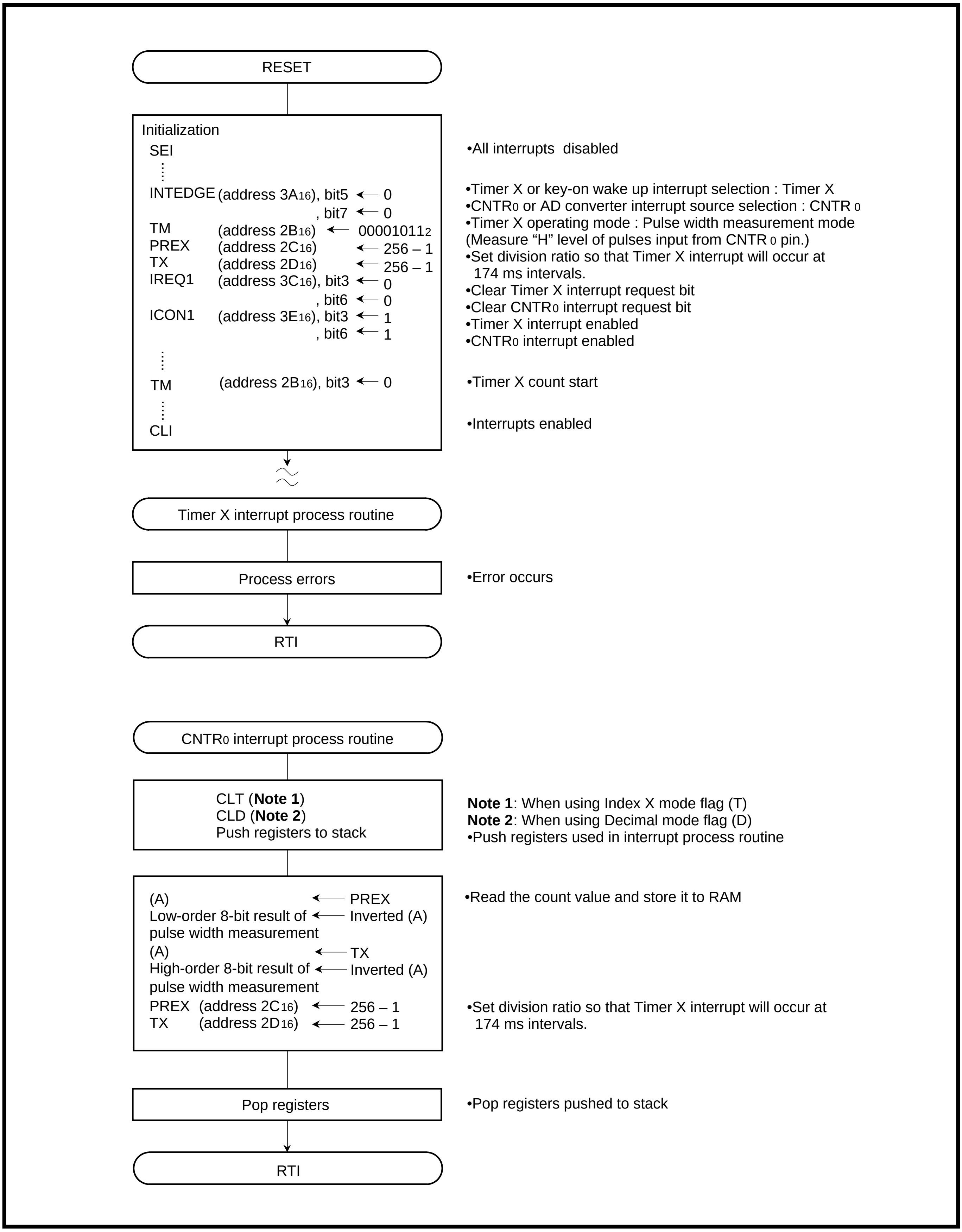

RESET
Initialization
SEI
INTEDGE (address 3A16), bit5 ← 0
, bit7 ← 0
TM (address 2B16) ← 000101112
PREX (address 2C16) ← 256 – 1
TX (address 2D16) ← 256 – 1
IREQ1 (address 3C16), bit3 ← 0
, bit6 ← 0
ICON1 (address 3E16), bit3 ← 1
, bit6 ← 1
TM (address 2B16), bit3 ← 0
CLI
•All interrupts disabled
•Timer X or key-on wake up interrupt selection : Timer X
•CNTR0 or AD converter interrupt source selection : CNTR 0
•Timer X operating mode : Pulse width measurement mode
(Measure “H” level of pulses input from CNTR 0 pin.)
•Set division ratio so that Timer X interrupt will occur at 174 ms intervals.
•Clear Timer X interrupt request bit
•Clear CNTR0 interrupt request bit
•Timer X interrupt enabled
•CNTR0 interrupt enabled
•Timer X count start
•Interrupts enabled
Timer X interrupt process routine
Process errors
•Error occurs
RTI
CNTR0 interrupt process routine
CLT (Note 1)
CLD (Note 2)
Push registers to stack
Note 1: When using Index X mode flag (T)
Note 2: When using Decimal mode flag (D)
•Push registers used in interrupt process routine
(A) ← PREX
Low-order 8-bit result of pulse width measurement ← Inverted (A)
(A) ← TX
High-order 8-bit result of pulse width measurement ← Inverted (A)
PREX (address 2C16) ← 256 – 1
TX (address 2D16) ← 256 – 1
•Read the count value and store it to RAM
•Set division ratio so that Timer X interrupt will occur at 174 ms intervals.
Pop registers
•Pop registers pushed to stack
RTI

Fig. 2.2.23 Control procedure

2.3 Serial I/O

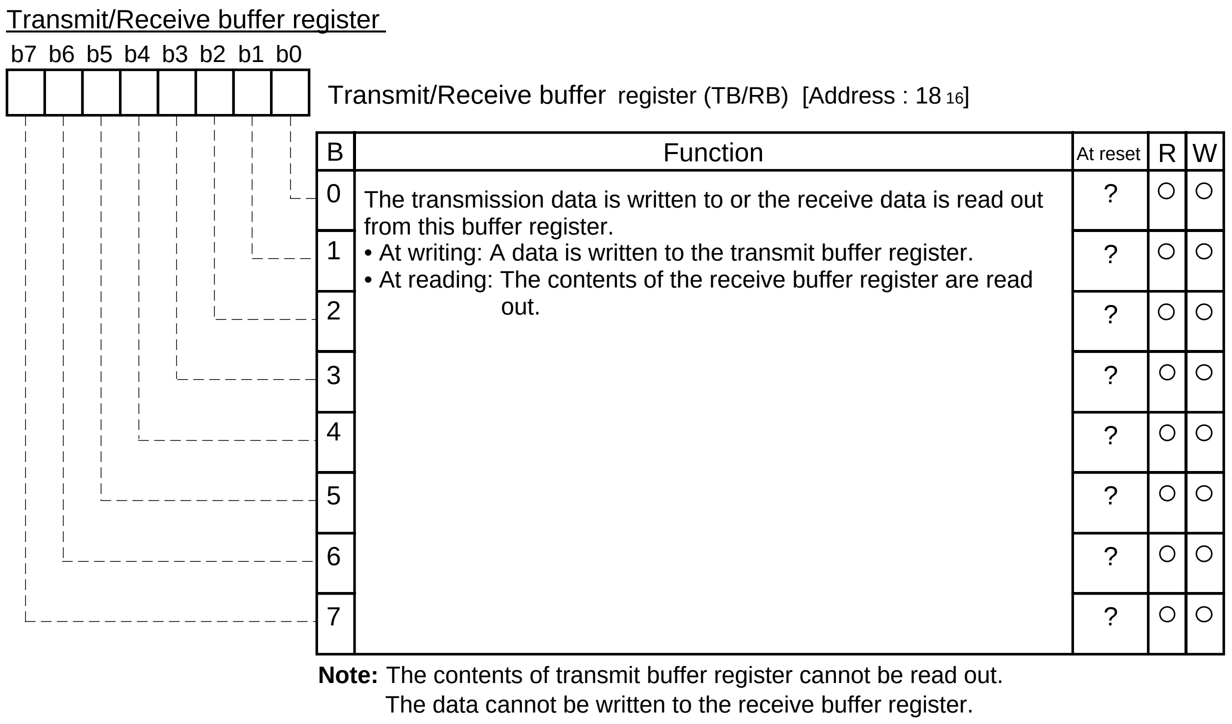
This paragraph explains the registers setting method and the notes relevant to the serial I/O.

2.3.1 Memory map

0018_{16}	Transmit/Receive buffer register (TB/RB)
0019_{16}	UART status register (UARTSTS)
$001A_{16}$	Serial I/O1 control register (SIO1CON)
$001B_{16}$	UART control register (UARTCON)
$001C_{16}$	Baud rate generator (BRG)
0030_{16}	Serial I/O2 control register (SIO2CON)
0031_{16}	Serial I/O2 register (SIO2)
$003A_{16}$	Interrupt edge selection register (INTEDGE)
$003C_{16}$	Interrupt request register 1 (IREQ1)
$003E_{16}$	Interrupt control register 1 (ICON1)

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

2.3.2 Relevant registers



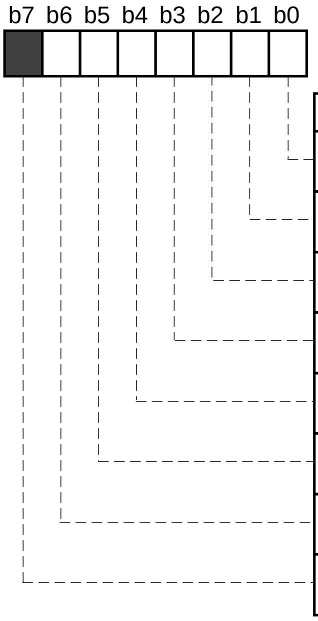
Transmit/Receive buffer register (TB/RB) [Address : 18_{16}]

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

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

Fig. 2.3.2 Structure of Transmit/Receive buffer register

UART status register

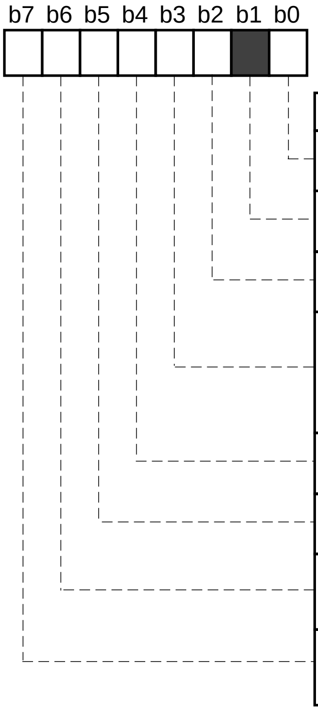


UART status register (UARTSTS) [Address : 19 16]

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

Fig. 2.3.3 Structure of UART status register

Serial I/O1 control register

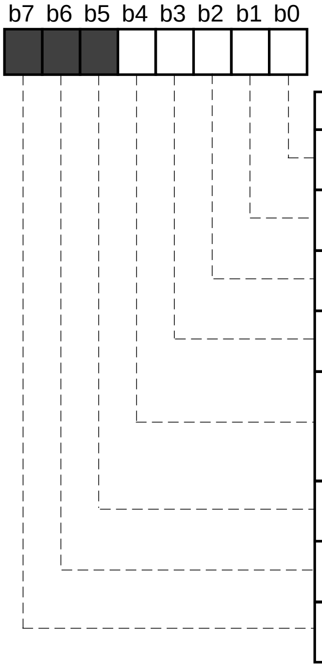


Serial I/O1 control register (SIO1CON) [Address : 1A 16]

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

Fig. 2.3.4 Structure of Serial I/O1 control register

UART control register

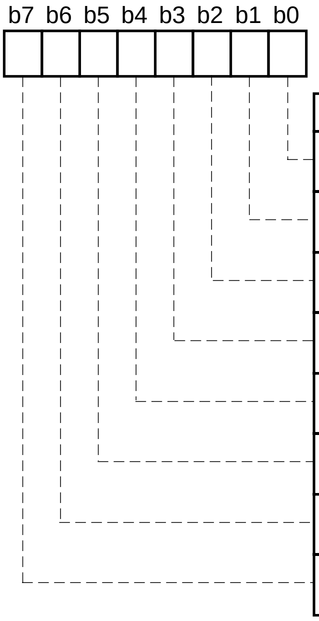


UART control register (UARTCON) [Address : 1B16]

B	Name	Function	At reset	R	W
0	Character length selection bit (CHAS)	0 : 8 bits 1 : 7 bits	0	○	○
1	Parity enable bit (PARE)	0 : Parity checking disabled 1 : Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0 : Even parity 1 : Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0 : 1 stop bit 1 : 2 stop bits	0	○	○
4	P1$_1$/TxD P-channel output disable bit (POFF)	In output mode 0 : CMOS output 1 : N-channel open-drain output	0	○	○
5	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "1".		1	○	×
6			1	○	×
7			1	○	×

Fig. 2.3.5 Structure of UART control register

Baud rate generator



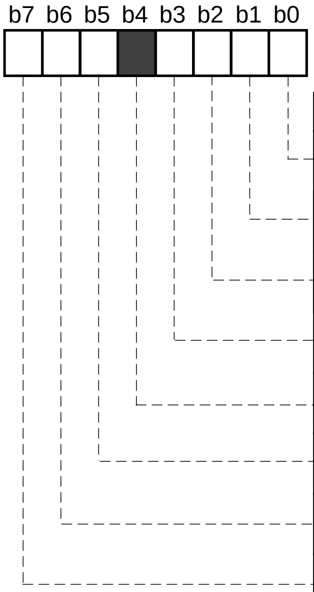
Baud rate generator (BRG) [Address : 1C16]

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

Fig. 2.3.6 Structure of Baud rate generator

Serial I/O2 control register

b7 b6 b5 b4 b3 b2 b1 b0



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

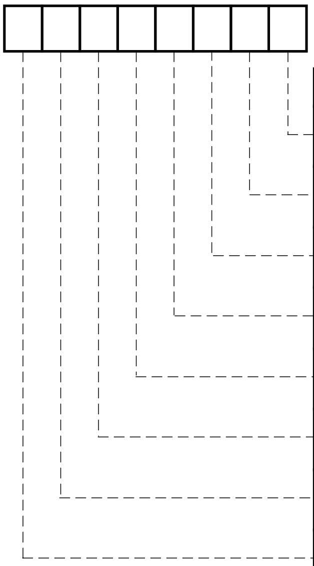
B	Name	Function	At reset	R	W
0	Internal synchronous clock selection bits	b2 b1 b0 0 0 0 : f(XIN)/8 0 0 1 : f(XIN)/16 0 1 0 : f(XIN)/32 0 1 1 : f(XIN)/64 1 1 0 : f(XIN)/128 1 1 1 : f(XIN)/256	0	○	○
1			0	○	○
2			0	○	○
3	SDATA pin selection bit (**Note**)	0 : I/O port / SDATA input 1 : SDATA output	0	○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

Note: When using it as a SDATA input, set the port P13 direction register bit to "0".

Fig. 2.3.7 Structure of Serial I/O2 control register

Serial I/O2 register

b7 b6 b5 b4 b3 b2 b1 b0

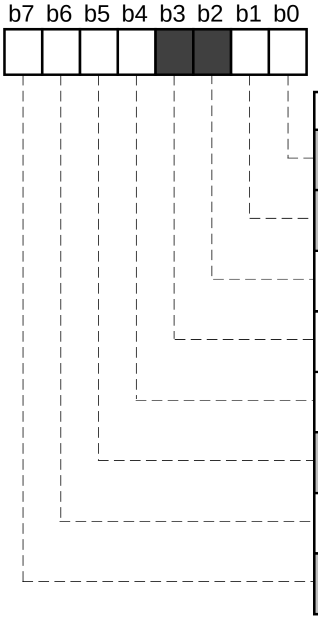


Serial I/O2 register (SIO2) [Address : 31 16]

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

Fig. 2.3.8 Structure of Serial I/O2 register

Interrupt edge selection register



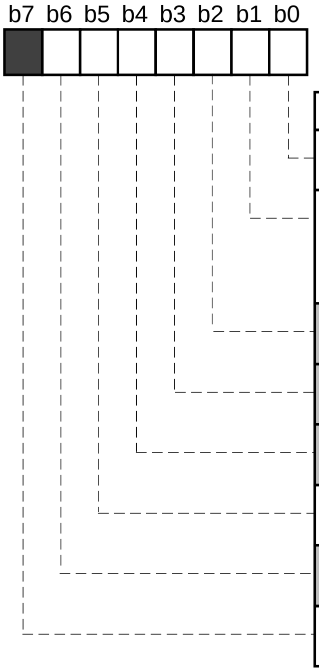
Interrupt edge selection register (INTEDGE) [Address : 3A$_{16}$]

B	Name	Function	At reset	R	W
0	INT$_0$ interrupt edge selection bit (**Note 1**)	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT$_1$ interrupt edge selection bit (**Note 2**)	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
3			0	○	×
4	Serial I/O1 or INT$_1$ interrupt selection bit	0 : Serial I/O1 1 : INT$_1$	0	○	○
5	Timer X or key-on wake up interrupt selection bit	0 : Timer X 1 : Key-on wake up	0	○	○
6	Timer 2 or serial I/O2 interrupt selection bit	0 : Timer 2 1 : Serial I/O2	0	○	○
7	CNTR$_0$ or AD converter interrupt selection bit	0 : CNTR$_0$ 1 : AD converter	0	○	○

Notes 1: 32-pin version: Not used.
This is a write disabled bit. When this bit is read out, the value is "0".
2: 36-pin and 32-pin version: Not used.
This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.3.9 Structure of Interrupt edge selection register

Interrupt request register 1



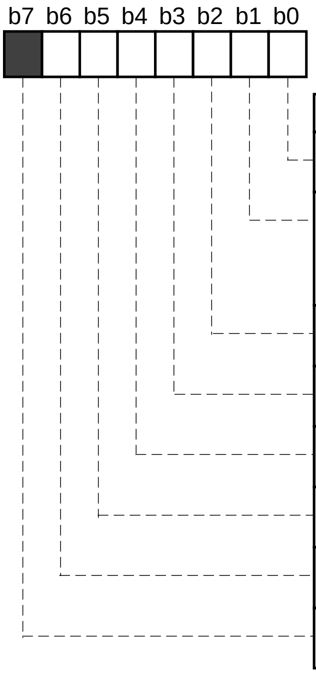
Interrupt request register 1 (IREQ1) [Address : 3C$_{16}$]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	UART transmit/USBSETUP/ OUT token/Reset/Suspend/ Resume/INT$_1$ interrupt request bit (**Note 1**)	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	INT$_0$ interrupt request bit (**Note 2**)	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Timer X or key-on wake up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer 2 or serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	CNTR$_0$ or AD converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

✻: These bits can be cleared to "0" by program, but cannot be set.
Notes 1: 36-pin version and 32-pin version: INT$_1$ interrupt does not exist.
2: 32-pin version: INT$_0$ interrupt does not exist.
This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.3.10 Structure of Interrupt request register 1

Interrupt control register 1



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

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	UART transmit/USBSETUP/ OUT token/Reset/Suspend/ Resume/INT$_1$ interrupt enable bit (**Note 1**)	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT$_0$ interrupt enable bit (**Note 2**)	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Timer X or key-on wake up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer 2 or serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	CNTR$_0$ or AD converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Nothing is allocated for this bit. Do not write “1” to this bit. When this bit is read out, the value is “0”.		0	○	✕

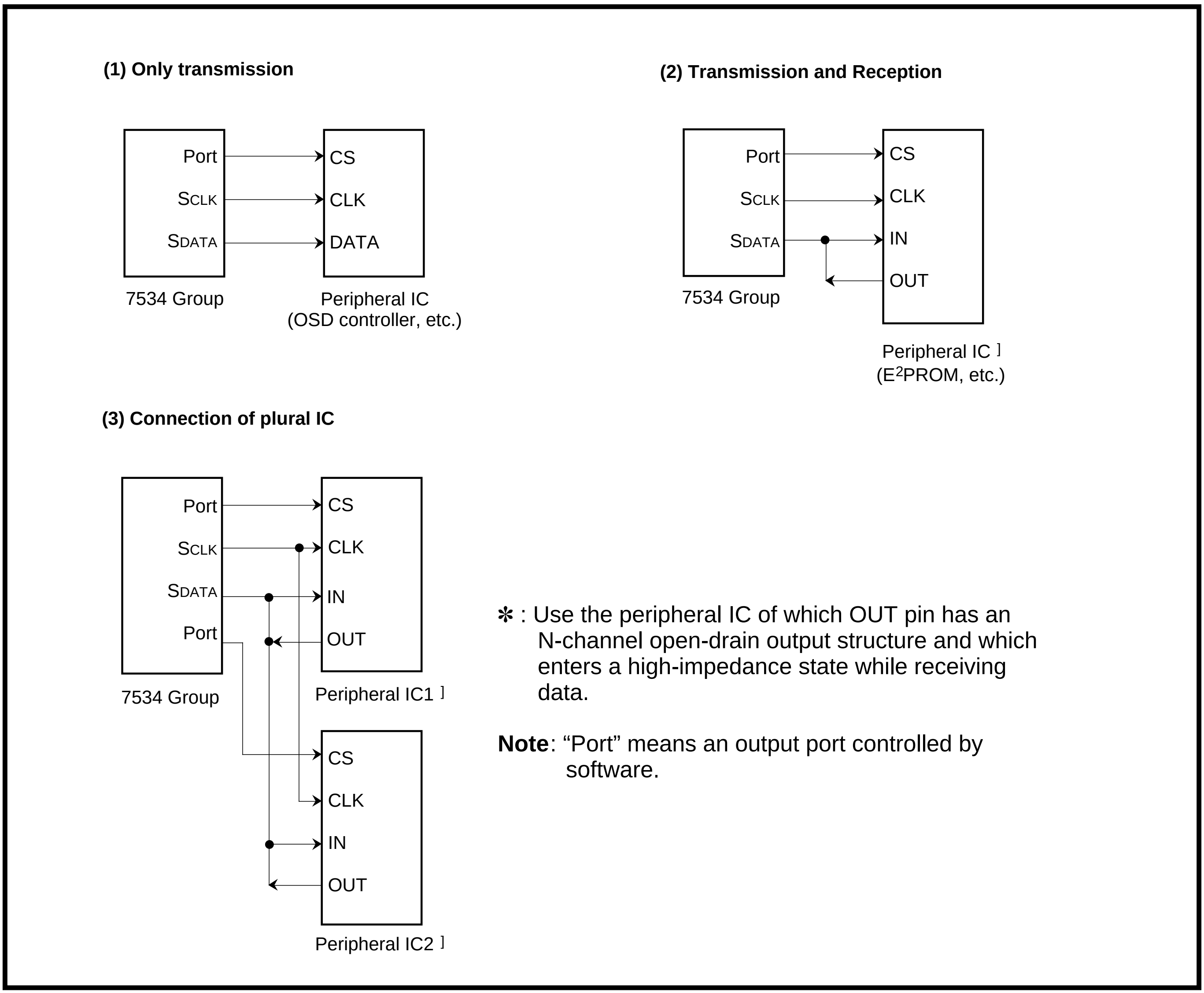
Notes 1: 36-pin version and 32-pin version: INT$_1$ interrupt does not exist.
2: 32-pin version: INT$_0$ interrupt does not exist.
This is a write disabled bit. When this bit is read out, the value is “0”.

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.



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

Note: “Port” means an output port controlled by software.

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

(2) Connection with microcomputer

Figure 2.3.13 shows connection examples with another microcomputer.

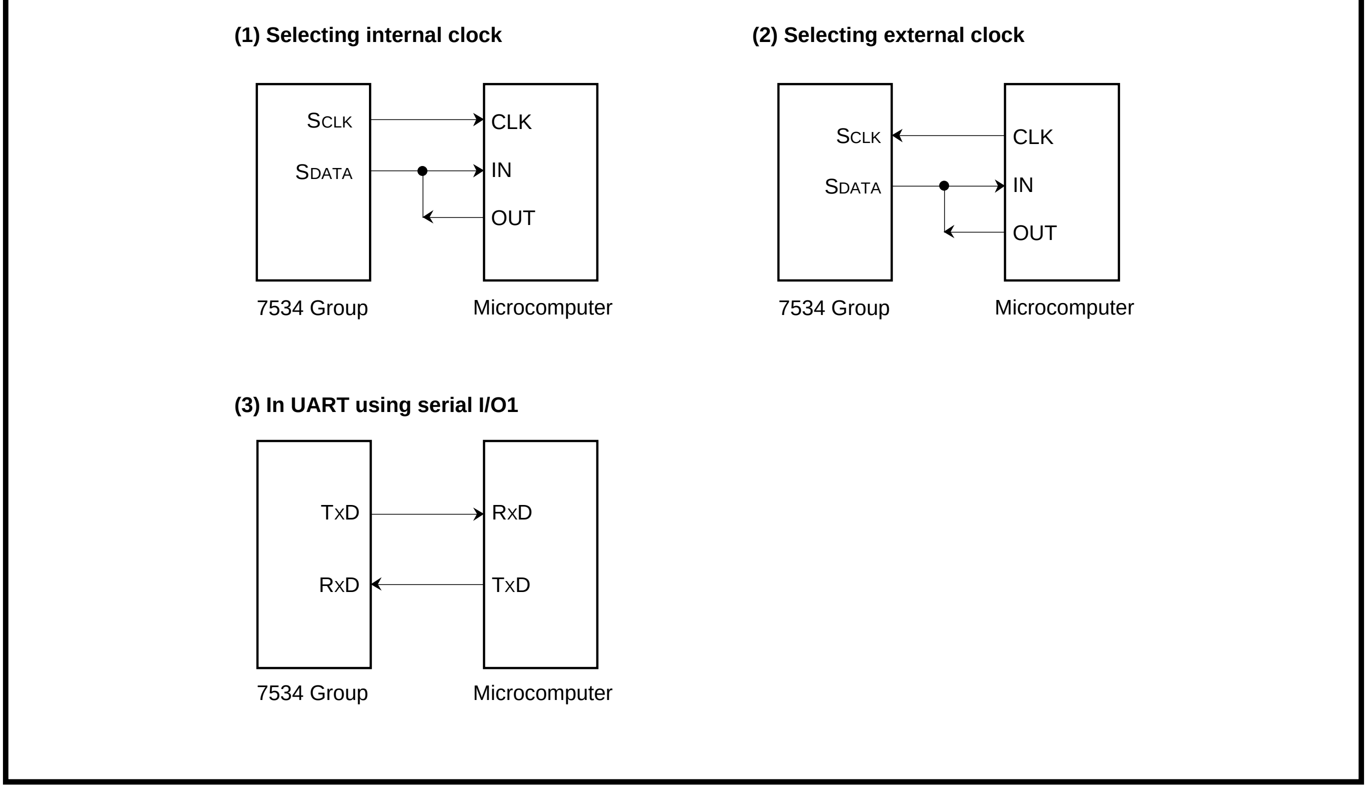


Fig. 2.3.13 Serial I/O connection examples (2)

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.

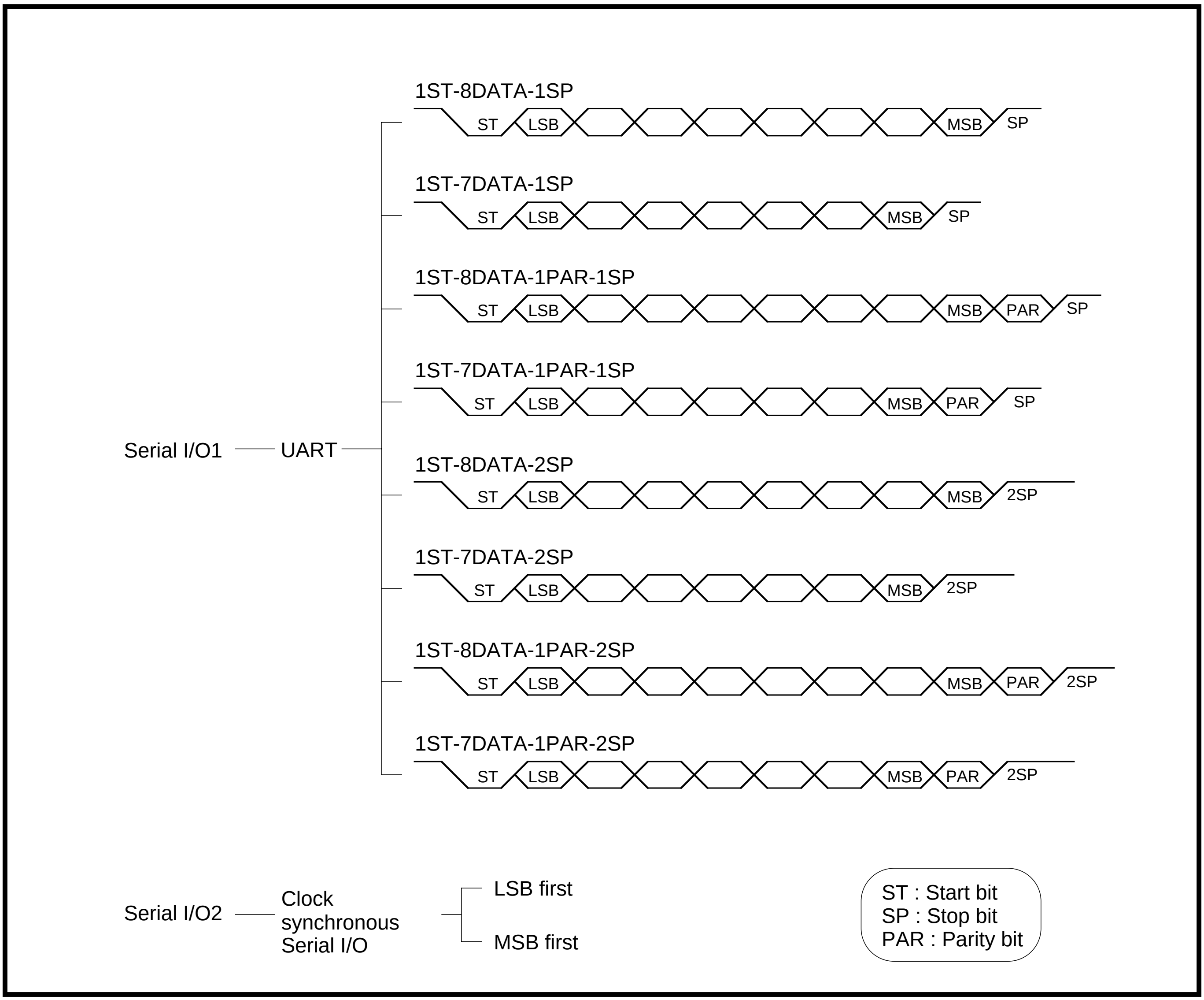


Fig. 2.3.14 Serial I/O transfer data format

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 $P0_0$ is used for communication control and outputs the quasi-$\overline{S_{RDY}}$ 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.

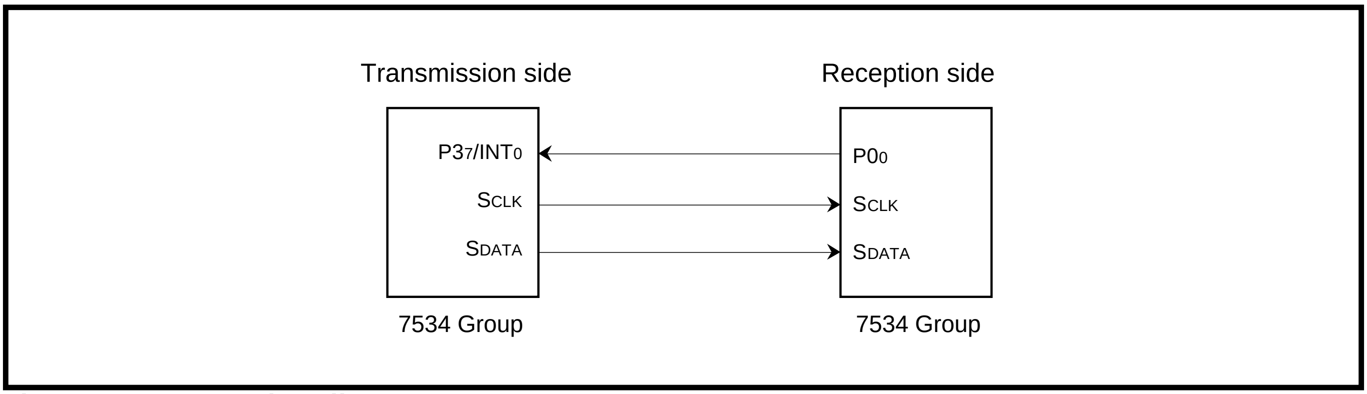


Fig. 2.3.15 Connection diagram

Specifications : •The Serial I/O2, clock synchronous serial I/O, is used.
•Synchronous clock frequency : 94 kHz; $f(X_{IN})$ = 6 MHz divided by 64
•Transfer direction : LSB first
•The reception side outputs the quasi-$\overline{S_{RDY}}$ signal at 2 ms intervals which the timer generates, and 2-byte data is transferred from the transmission side to the reception side.

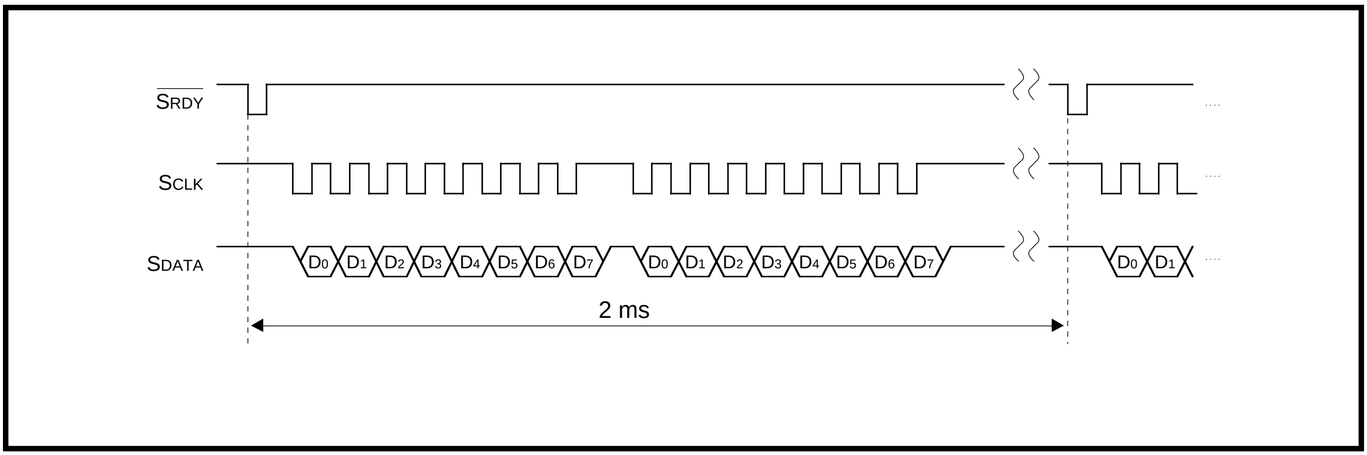


Fig. 2.3.16 Timing chart

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.

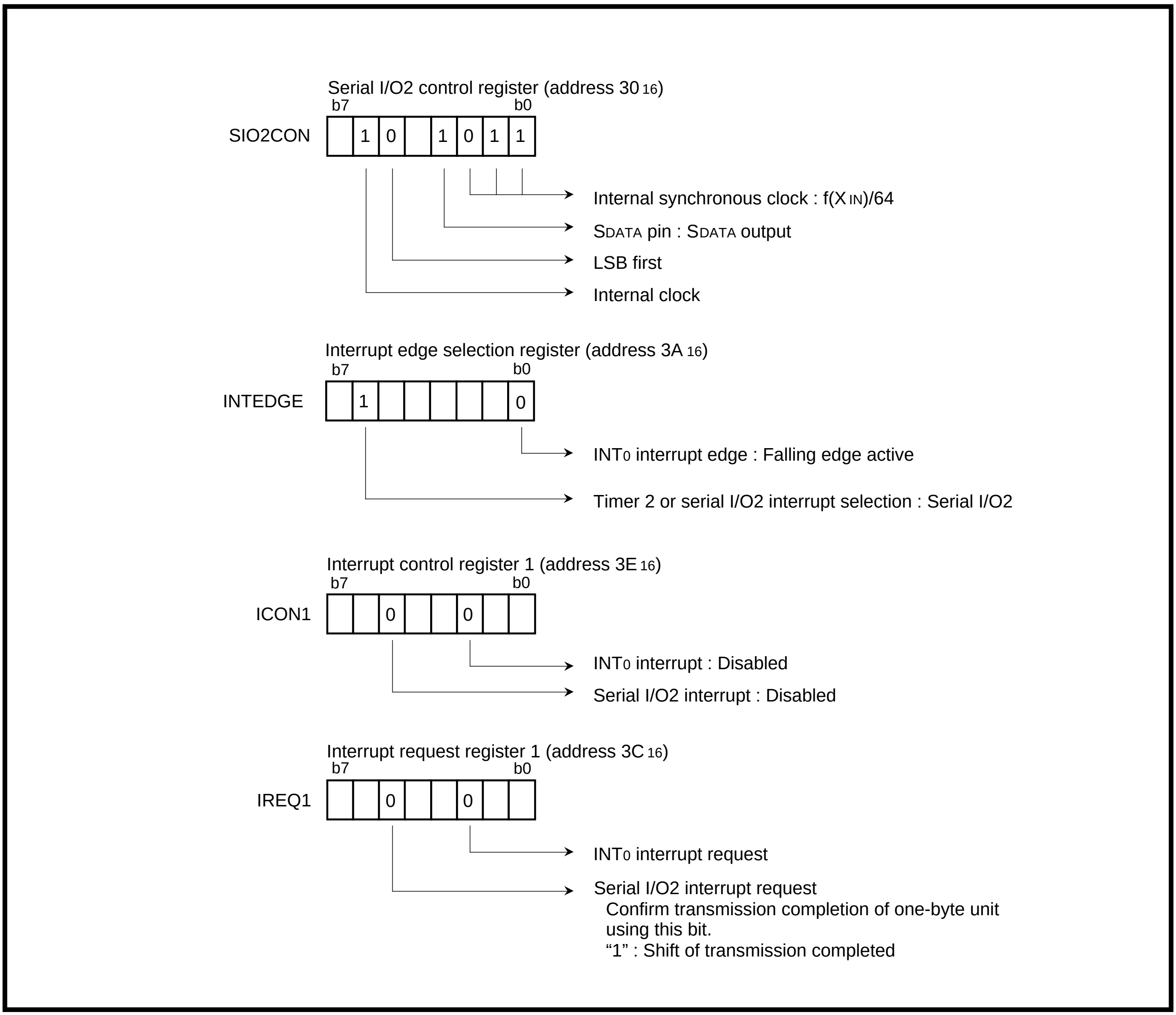

Serial I/O2 control register (address 30 16)
b7
b0
SIO2CON
1
0
1
0
1
1
Internal synchronous clock : f(X IN)/64
SDATA pin : SDATA output
LSB first
Internal clock
Interrupt edge selection register (address 3A 16)
b7
b0
INTEDGE
1
0
INT0 interrupt edge : Falling edge active
Timer 2 or serial I/O2 interrupt selection : Serial I/O2
Interrupt control register 1 (address 3E 16)
b7
b0
ICON1
0
0
INT0 interrupt : Disabled
Serial I/O2 interrupt : Disabled
Interrupt request register 1 (address 3C 16)
b7
b0
IREQ1
0
0
INT0 interrupt request
Serial I/O2 interrupt request
Confirm transmission completion of one-byte unit
using this bit.
"1" : Shift of transmission completed

Fig. 2.3.17 Registers setting relevant to transmission side

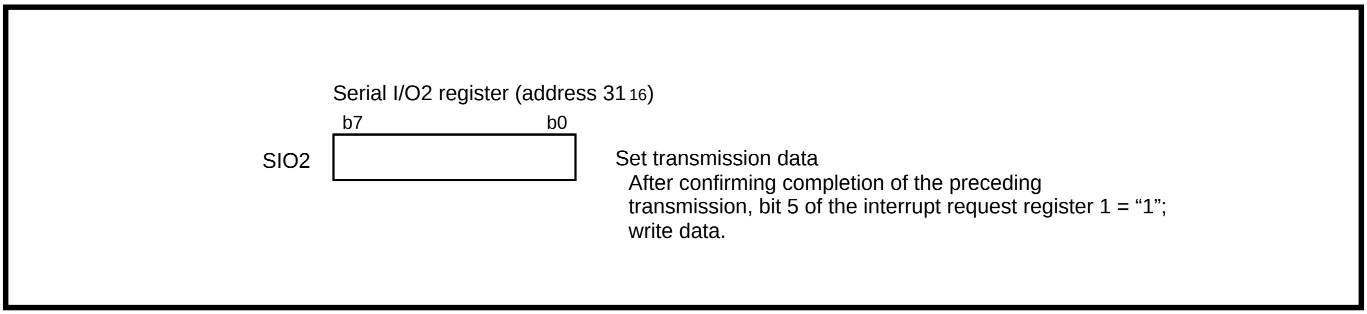

Serial I/O2 register (address 31₁₆)
b7
b0
SIO2
Set transmission data
After confirming completion of the preceding transmission, bit 5 of the interrupt request register 1 = "1"; write data.

Fig. 2.3.18 Transmission data setting of serial I/O2

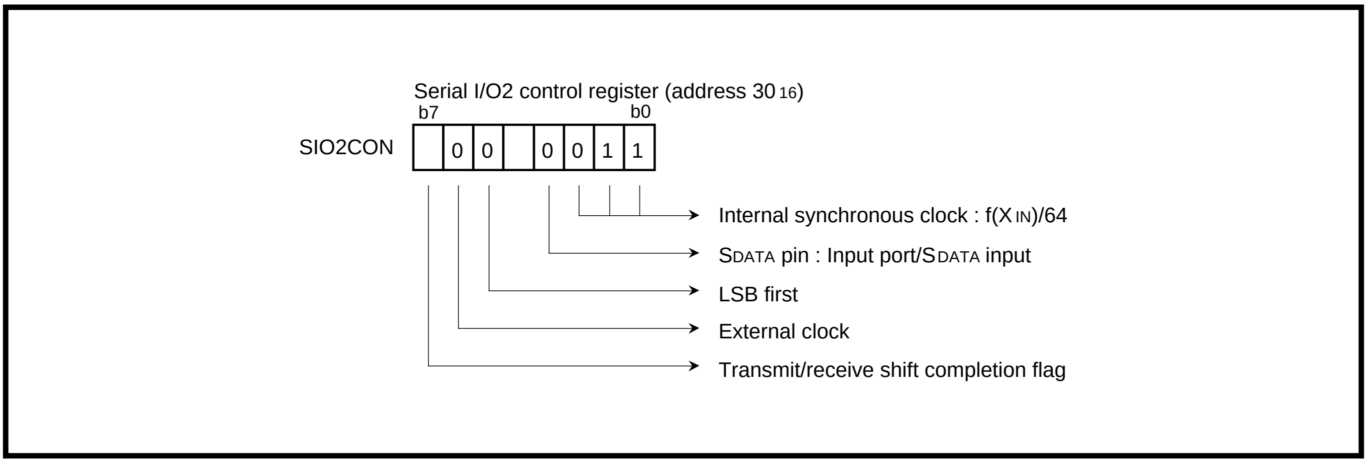
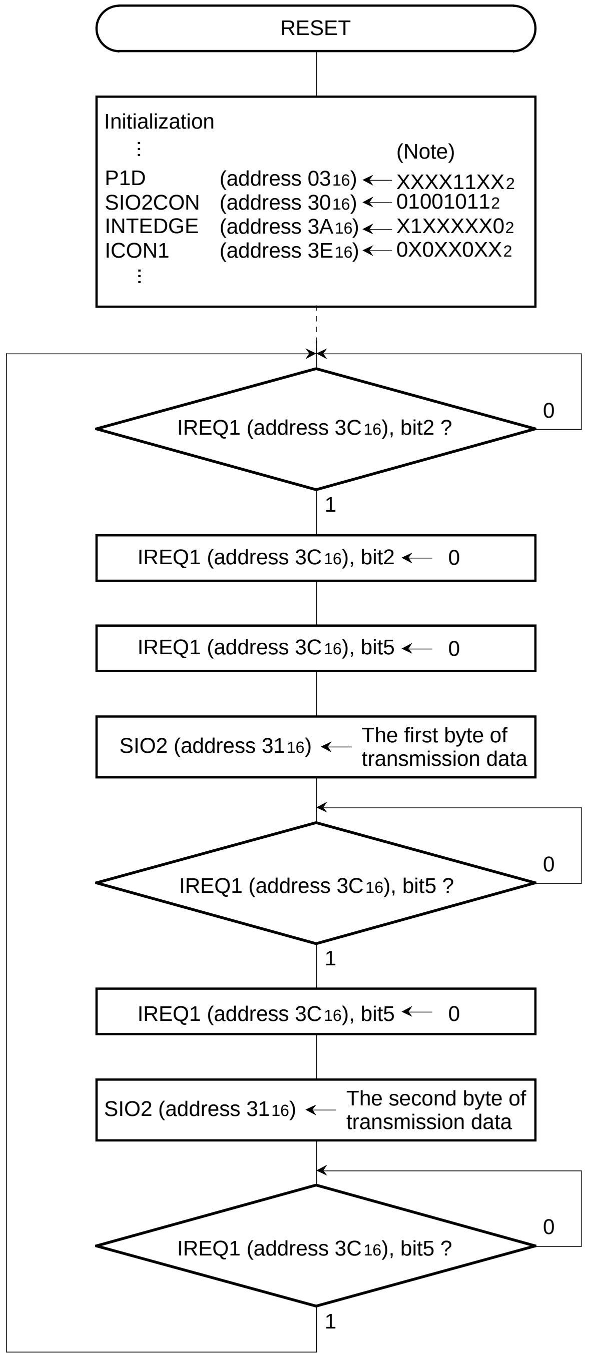

Serial I/O2 control register (address 30₁₆)
b7
b0
SIO2CON
0 0 0 0 1 1
Internal synchronous clock : f(XIN)/64
SDATA pin : Input port/SDATA input
LSB first
External clock
Transmit/receive shift completion flag

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.


RESET
Initialization
(Note)
P1D (address 03_{16}) ← $XXXX11XX_2$
SIO2CON (address 30_{16}) ← 01001011_2
INTEDGE (address $3A_{16}$) ← $X1XXXXX0_2$
ICON1 (address $3E_{16}$) ← $0X0XX0XX_2$
IREQ1 (address $3C_{16}$), bit2 ?
0
1
IREQ1 (address $3C_{16}$), bit2 ← 0
IREQ1 (address $3C_{16}$), bit5 ← 0
SIO2 (address 31_{16}) ← The first byte of transmission data
IREQ1 (address $3C_{16}$), bit5 ?
0
1
IREQ1 (address $3C_{16}$), bit5 ← 0
SIO2 (address 31_{16}) ← The second byte of transmission data
IREQ1 (address $3C_{16}$), bit5 ?
0
1

● X: This bit is not used here. Set it to “0” or “1” arbitrary.

Note: When the internal clock is selected and the direction register of P1$_3$/SDATA pin is set to the input mode, the SDATA pin is in a high impedance state after the data transfer is completed.

•Serial I/O2 setting
•INT$_0$ falling, Serial I/O2 selected
•INT$_0$ interrupt, serial I/O2 interrupt disabled

•Detection of INT$_0$ falling edge

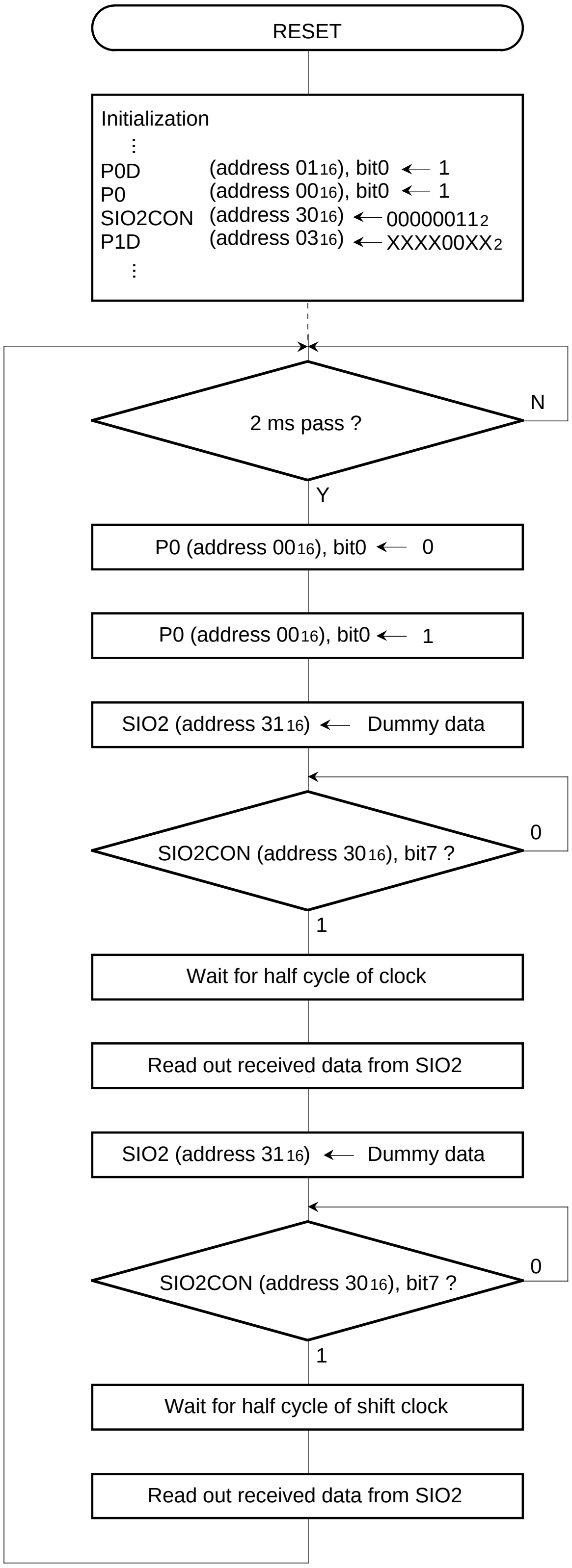
•Transmission data write
(One-byte transmission starts.)

•Judgment of completion of one-byte transmission

•Transmission data write
(One-byte transmission starts.)

•Judgment of completion of one-byte transmission

Fig. 2.3.20 Control procedure of transmission side


RESET
Initialization
⋮
P0D (address 01₁₆), bit0 ← 1
P0 (address 00₁₆), bit0 ← 1
SIO2CON (address 30₁₆) ← 00000011₂
P1D (address 03₁₆) ← XXXX00XX₂
⋮
2 ms pass ?
N
Y
P0 (address 00₁₆), bit0 ← 0
P0 (address 00₁₆), bit0 ← 1
SIO2 (address 31₁₆) ← Dummy data
SIO2CON (address 30₁₆), bit7 ?
0
1
Wait for half cycle of clock
Read out received data from SIO2
SIO2 (address 31₁₆) ← Dummy data
SIO2CON (address 30₁₆), bit7 ?
0
1
Wait for half cycle of shift clock
Read out received data from SIO2

● X: This bit is not used here. Set it to “0” or “1” arbitrary.

•Quasi-$\overline{S_{RDY}}$ signal “H”
•Serial I/O2 setting

•Generation of a 2 ms interval using Timer

•Quasi-$\overline{S_{RDY}}$ signal output

•Transmit/receive shift completion flag cleared

•Judgment of completion of one-byte reception

•Transmit/receive shift completion flag cleared

•Judgment of completion of one-byte reception

Fig. 2.3.21 Control procedure of reception side

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

Outline : 2-byte data is transmitted and received, using the clock asynchronous serial I/O. Port $P0_0$ is used for communication control.

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

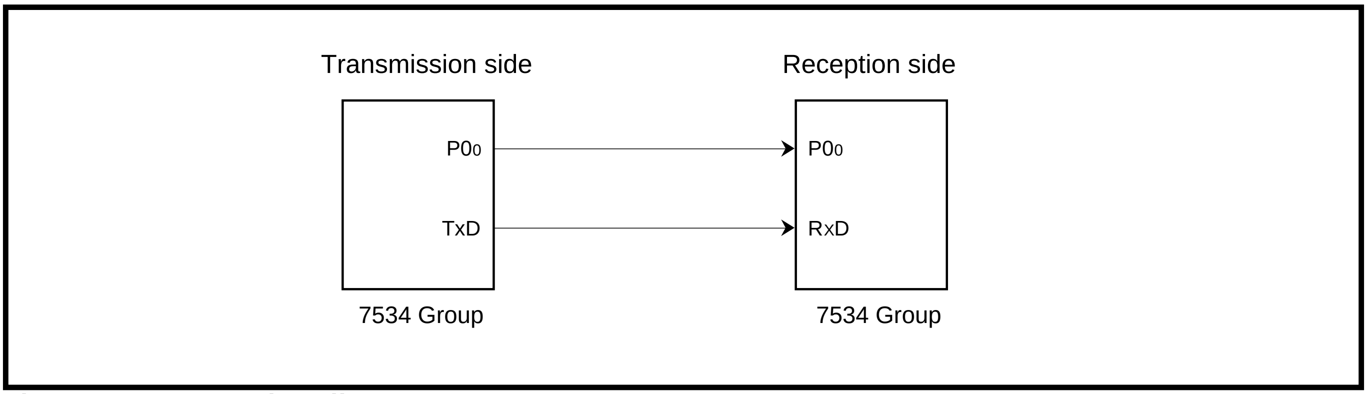


Fig. 2.3.22 Connection diagram

Specifications : •The Serial I/O1, asynchronous serial I/O, is used.
•Transfer bit rate : 9600 bps; $f(X_{IN})$ = 4.9152 MHz divided by 512
•Communication control using port $P0_0$; Port $P0_0$ 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

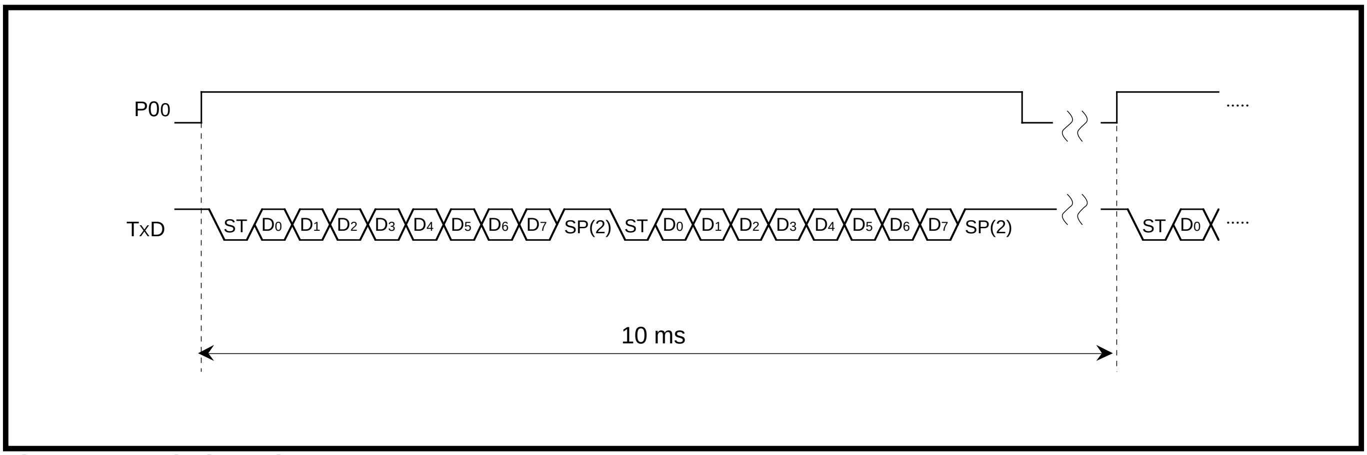


Fig. 2.3.23 Timing chart

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”

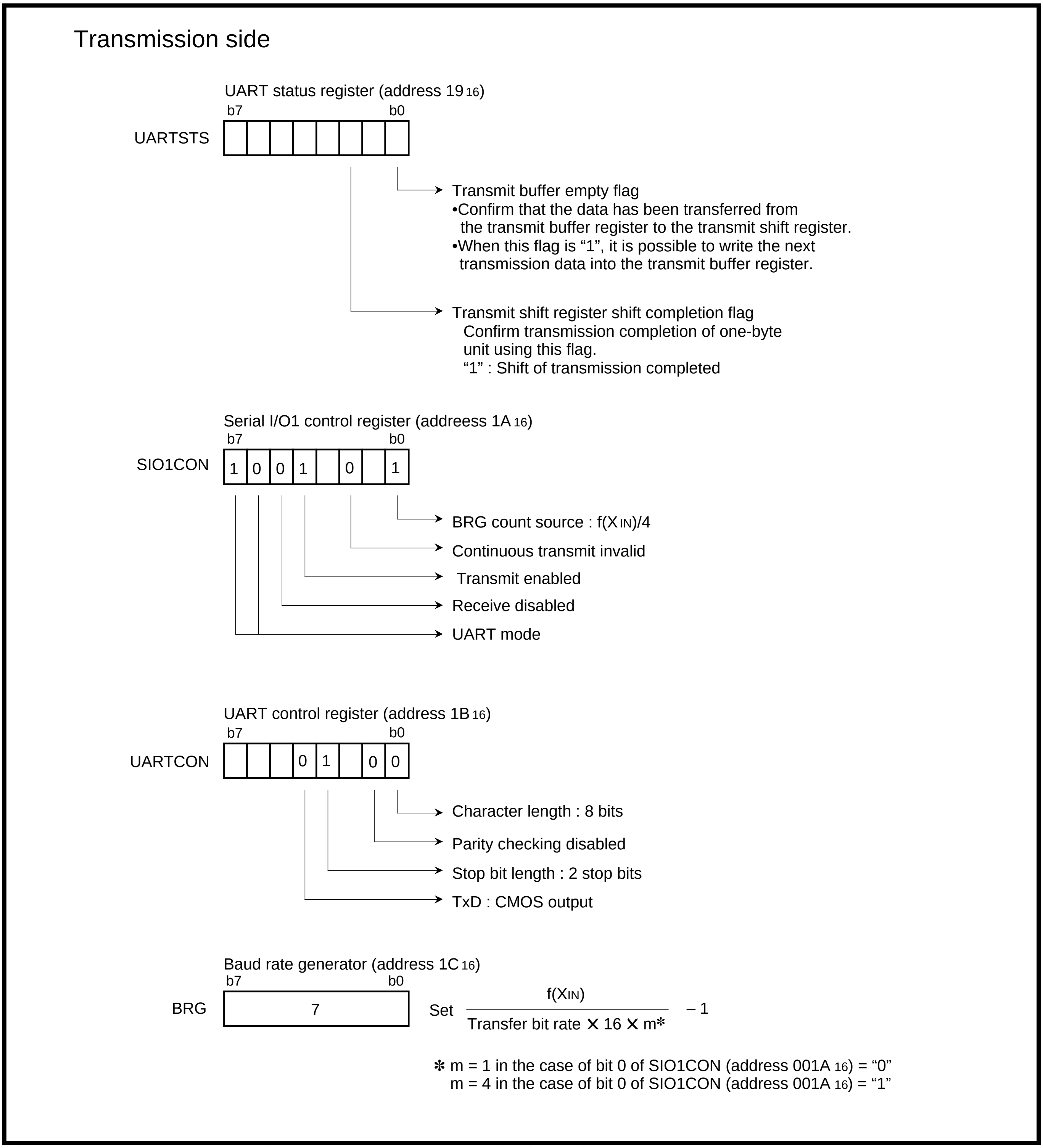
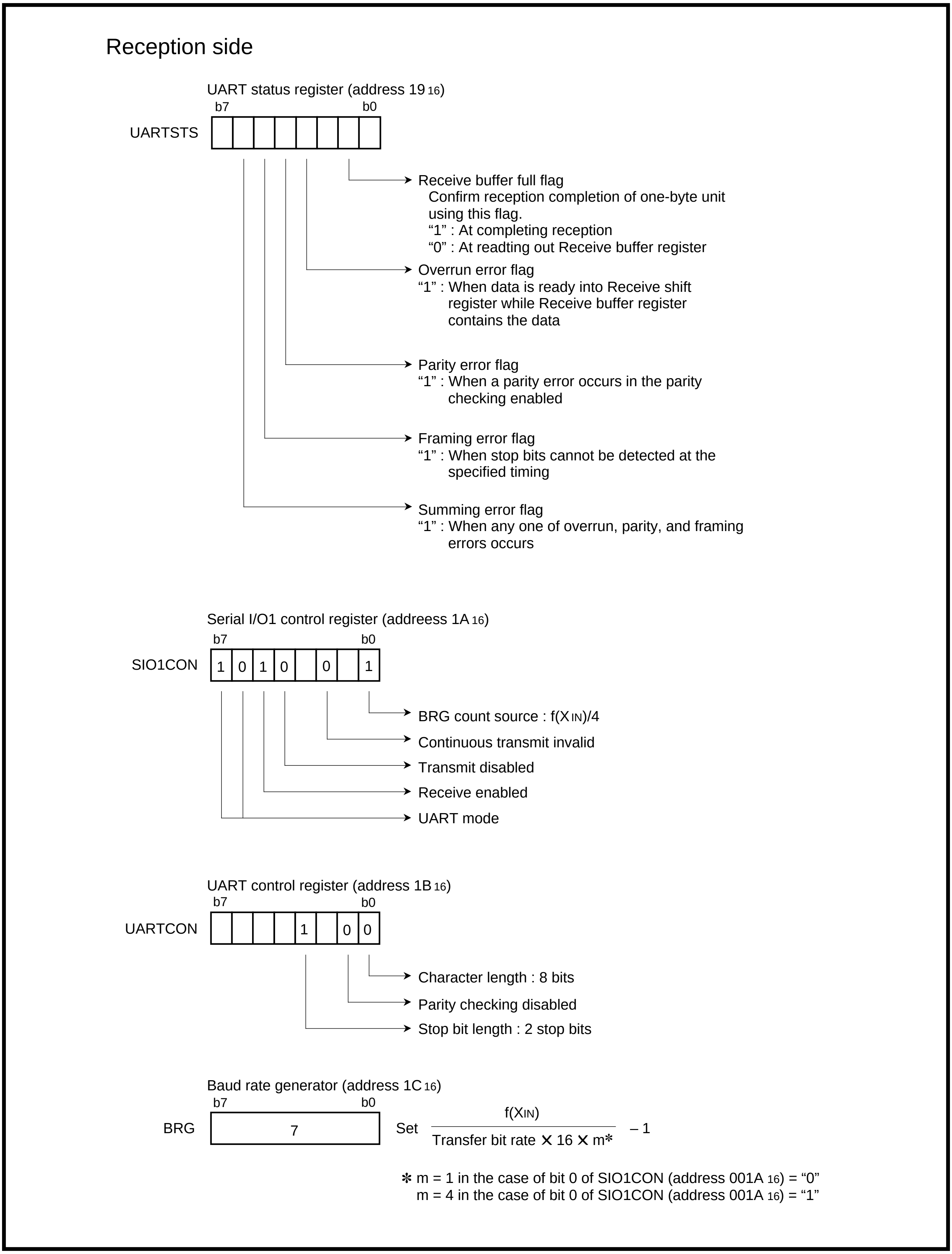

Transmission side
UART status register (address 19 16)
b7
b0
UARTSTS
Transmit buffer empty flag
•Confirm that the data has been transferred from the transmit buffer register to the transmit shift register.
•When this flag is “1”, it is possible to write the next transmission data into the transmit buffer register.
Transmit shift register shift completion flag
Confirm transmission completion of one-byte unit using this flag.
“1” : Shift of transmission completed
Serial I/O1 control register (addreess 1A 16)
b7
b0
SIO1CON
1 0 0 1 0 1
BRG count source : f(X IN)/4
Continuous transmit invalid
Transmit enabled
Receive disabled
UART mode
UART control register (address 1B 16)
b7
b0
UARTCON
0 1 0 0
Character length : 8 bits
Parity checking disabled
Stop bit length : 2 stop bits
TxD : CMOS output
Baud rate generator (address 1C 16)
b7
b0
BRG
7
Set f(XIN) / Transfer bit rate ✕ 16 ✕ m✻ – 1
✻ m = 1 in the case of bit 0 of SIO1CON (address 001A 16) = “0”
m = 4 in the case of bit 0 of SIO1CON (address 001A 16) = “1”

Fig. 2.3.24 Registers setting relevant to transmission side



Reception side

UART status register (address 19_{16})

Receive buffer full flag
Confirm reception completion of one-byte unit using this flag.
"1" : At completing reception
"0" : At readting out Receive buffer register

Overrun error flag
"1" : When data is ready into Receive shift register while Receive buffer register contains the data

Parity error flag
"1" : When a parity error occurs in the parity checking enabled

Framing error flag
"1" : When stop bits cannot be detected at the specified timing

Summing error flag
"1" : When any one of overrun, parity, and framing errors occurs

Serial I/O1 control register (addeess $1A_{16}$)

BRG count source : $f(X_{IN})/4$

Continuous transmit invalid

Transmit disabled

Receive enabled

UART mode

UART control register (address $1B_{16}$)

Character length : 8 bits

Parity checking disabled

Stop bit length : 2 stop bits

Baud rate generator (address $1C_{16}$)

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

* m = 1 in the case of bit 0 of SIO1CON (address $001A_{16}$) = "0"
m = 4 in the case of bit 0 of SIO1CON (address $001A_{16}$) = "1"

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.

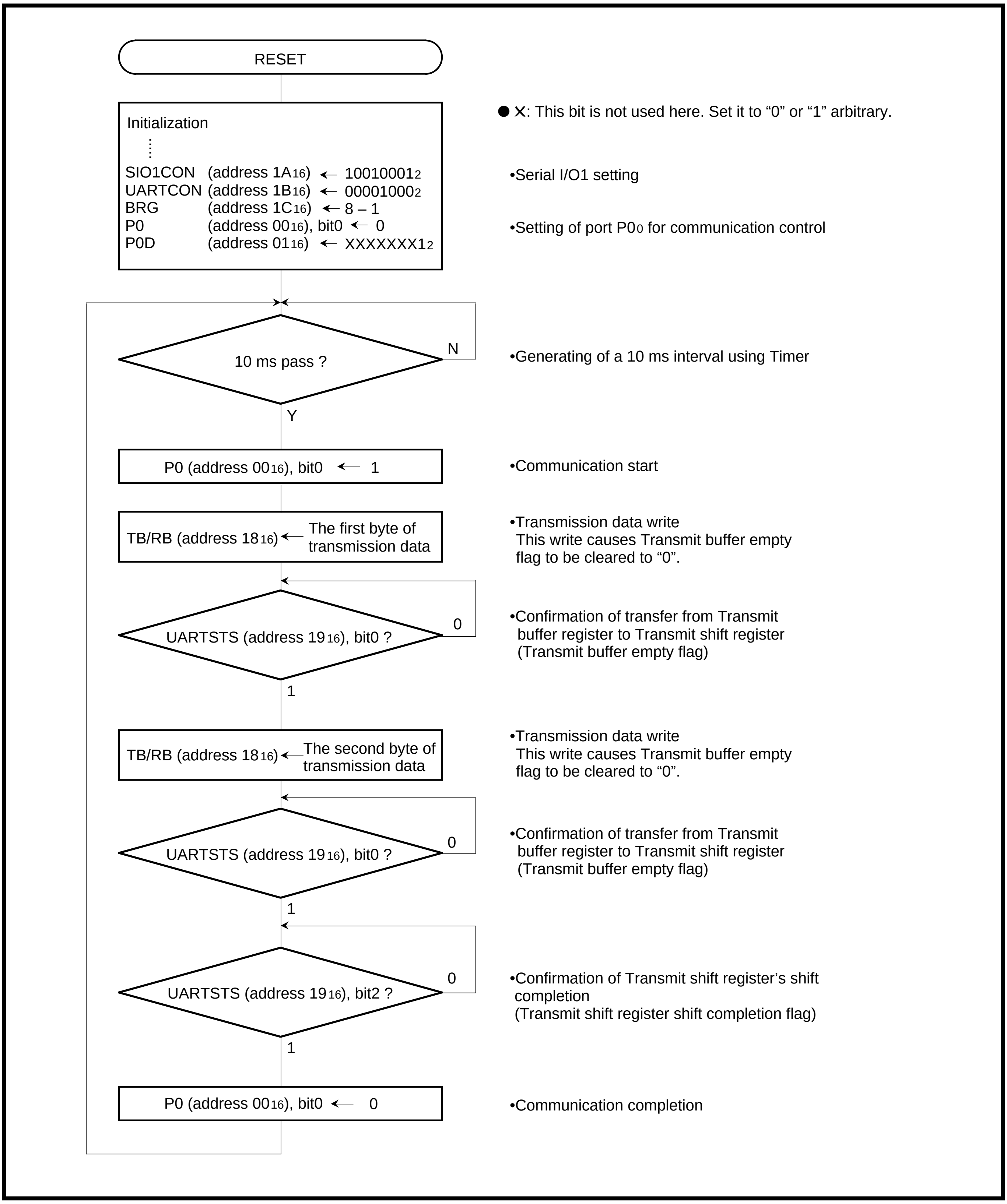

RESET
Initialization
SIO1CON (address 1A16) ← 100100012
UARTCON (address 1B16) ← 000010002
BRG (address 1C16) ← 8 – 1
P0 (address 0016), bit0 ← 0
P0D (address 0116) ← XXXXXXX12
● X: This bit is not used here. Set it to "0" or "1" arbitrary.
•Serial I/O1 setting
•Setting of port P00 for communication control
10 ms pass ?
N
Y
•Generating of a 10 ms interval using Timer
P0 (address 0016), bit0 ← 1
•Communication start
TB/RB (address 1816) ← The first byte of transmission data
•Transmission data write This write causes Transmit buffer empty flag to be cleared to "0".
UARTSTS (address 1916), bit0 ?
0
1
•Confirmation of transfer from Transmit buffer register to Transmit shift register (Transmit buffer empty flag)
TB/RB (address 1816) ← The second byte of transmission data
•Transmission data write This write causes Transmit buffer empty flag to be cleared to "0".
UARTSTS (address 1916), bit0 ?
0
1
•Confirmation of transfer from Transmit buffer register to Transmit shift register (Transmit buffer empty flag)
UARTSTS (address 1916), bit2 ?
0
1
•Confirmation of Transmit shift register's shift completion (Transmit shift register shift completion flag)
P0 (address 0016), bit0 ← 0
•Communication completion

Fig. 2.3.26 Control procedure of transmission side

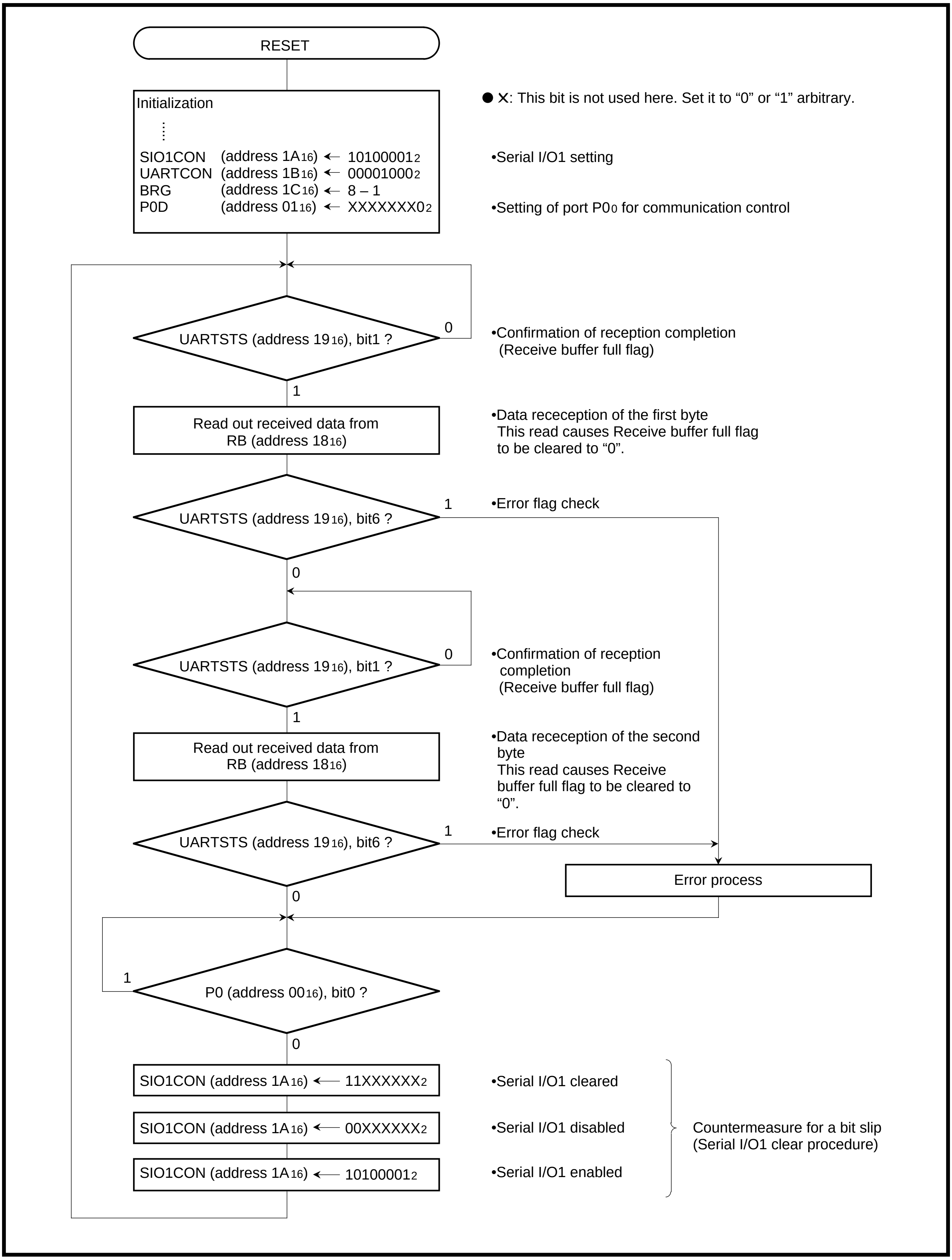
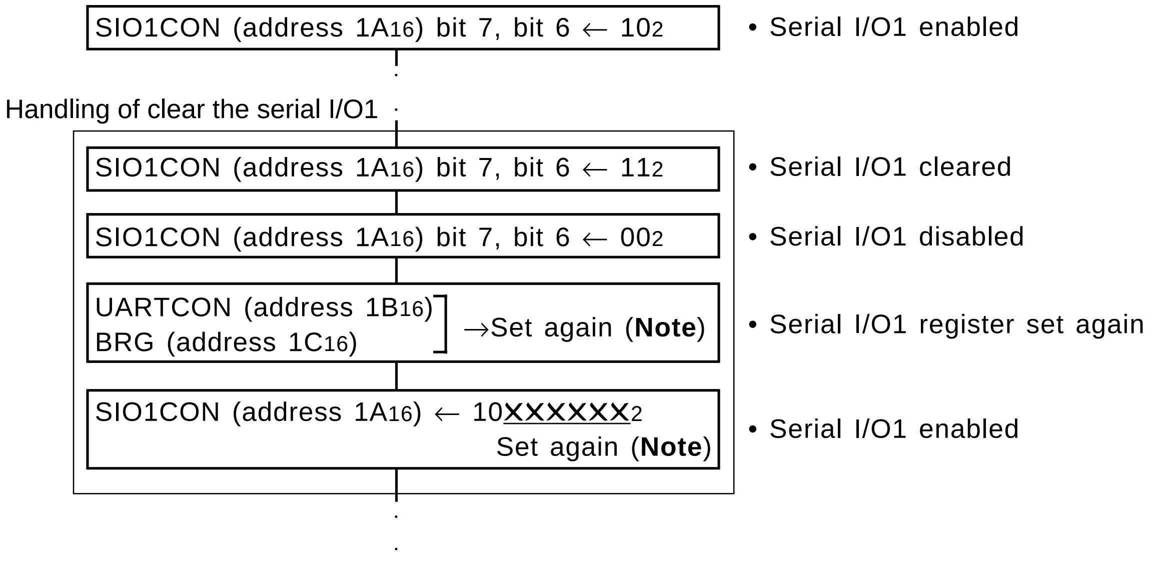

RESET
Initialization
SIO1CON (address $1A_{16}$) ← 10100001_2
UARTCON (address $1B_{16}$) ← 00001000_2
BRG (address $1C_{16}$) ← 8 – 1
P0D (address 01_{16}) ← $XXXXXXX0_2$
● X: This bit is not used here. Set it to “0” or “1” arbitrary.
•Serial I/O1 setting
•Setting of port $P0_0$ for communication control
UARTSTS (address 19_{16}), bit1 ?
0
1
•Confirmation of reception completion (Receive buffer full flag)
Read out received data from RB (address 18_{16})
•Data receception of the first byte This read causes Receive buffer full flag to be cleared to “0”.
UARTSTS (address 19_{16}), bit6 ?
1
0
•Error flag check
UARTSTS (address 19_{16}), bit1 ?
0
1
•Confirmation of reception completion (Receive buffer full flag)
Read out received data from RB (address 18_{16})
•Data receception of the second byte This read causes Receive buffer full flag to be cleared to “0”.
UARTSTS (address 19_{16}), bit6 ?
1
0
•Error flag check
Error process
P0 (address 00_{16}), bit0 ?
1
0
SIO1CON (address $1A_{16}$) ← $11XXXXXX_2$
•Serial I/O1 cleared
SIO1CON (address $1A_{16}$) ← $00XXXXXX_2$
•Serial I/O1 disabled
SIO1CON (address $1A_{16}$) ← 10100001_2
•Serial I/O1 enabled
Countermeasure for a bit slip (Serial I/O1 clear procedure)

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.


SIO1CON (address 1A16) bit 7, bit 6 ← 102
• Serial I/O1 enabled
Handling of clear the serial I/O1
SIO1CON (address 1A16) bit 7, bit 6 ← 112
• Serial I/O1 cleared
SIO1CON (address 1A16) bit 7, bit 6 ← 002
• Serial I/O1 disabled
UARTCON (address 1B16)
BRG (address 1C16)
→Set again (Note)
• Serial I/O1 register set again
SIO1CON (address 1A16) ← 10XXXXXX2
Set again (Note)
• Serial I/O1 enabled

Note: When the contents of register is not changed, setting again is not necessary.

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

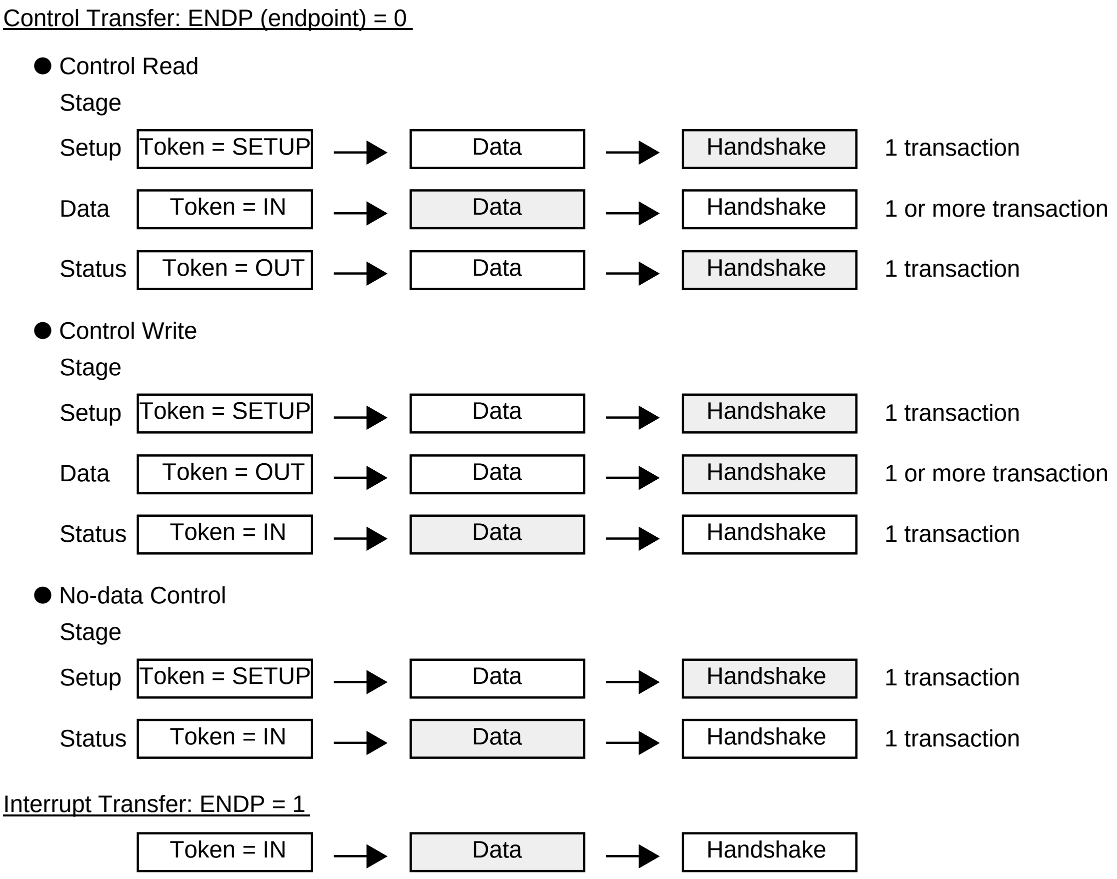
Transfer type	L.S./F.S./H.S.	Operation
Control	L.S./F.S./H.S.	This is used when setting up and for all devices common.
Interrupt	L.S./F.S./H.S.	This is used when transferring a small amount of data in real time.
Bulk	F.S./H.S.	This is used when transferring a large amount of data in no real time.
Isochronous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



Control Transfer: ENDP (endpoint) = 0

● Control Read

Stage				
Setup	Token = SETUP →	Data →	Handshake	1 transaction
Data	Token = IN →	Data →	Handshake	1 or more transaction
Status	Token = OUT →	Data →	Handshake	1 transaction

● Control Write

Stage				
Setup	Token = SETUP →	Data →	Handshake	1 transaction
Data	Token = OUT →	Data →	Handshake	1 or more transaction
Status	Token = IN →	Data →	Handshake	1 transaction

● No-data Control

Stage				
Setup	Token = SETUP →	Data →	Handshake	1 transaction
Status	Token = IN →	Data →	Handshake	1 transaction

Interrupt Transfer: ENDP = 1

Token = IN → Data → Handshake

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

Fig. 2.4.1 Communication sequence of USB

(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.

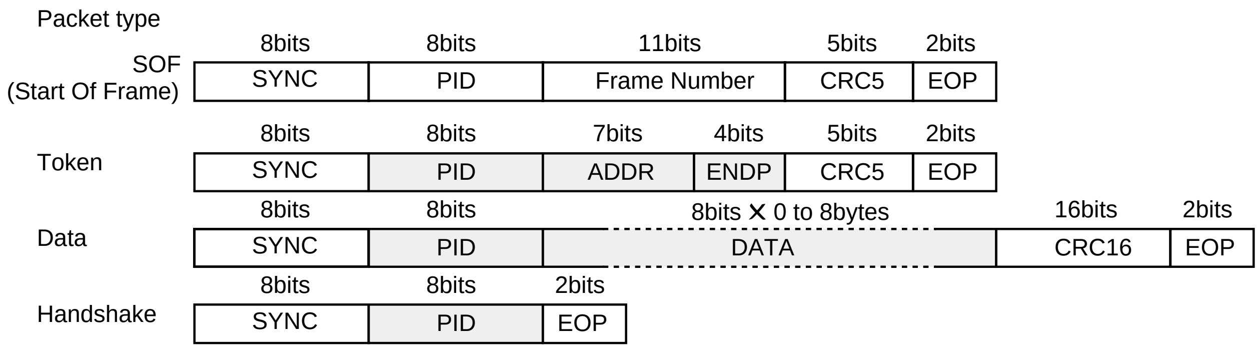
(4) Packet structure

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.



Notes 1: The shaded parts show the data which requires processing by software in 7534 Group.
2: The DATA number of data PID shows the number in L.S.
3: Determination of "Start Of Frame" is not executed because the processing is not required.

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 × 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 µs 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.

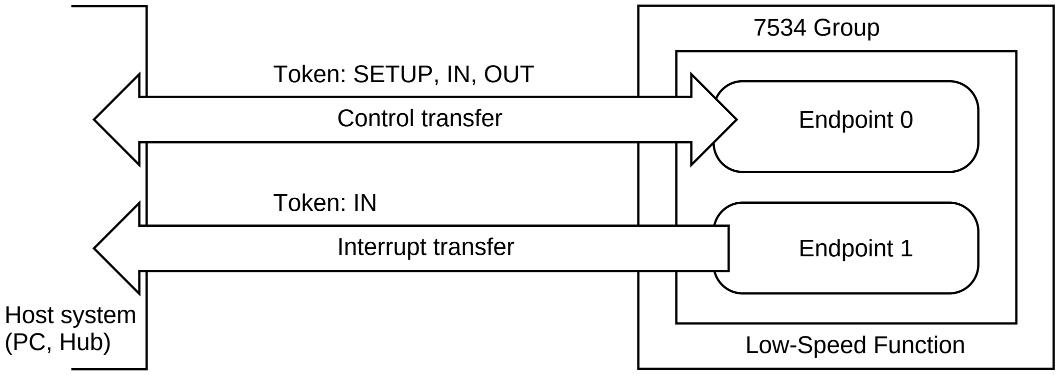
Table 2.4.5 Special signal of USB

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 µs 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.



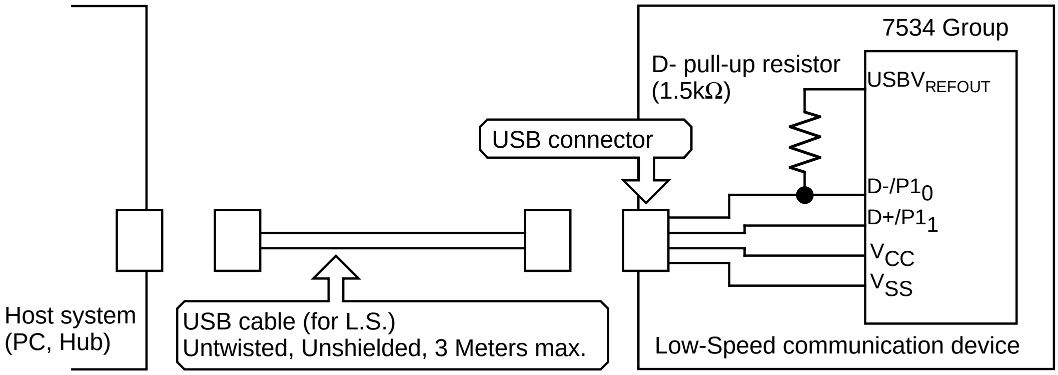
Note: Transfer direction shows the DATA transmission direction of PID.

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 kΩ of communications lines.

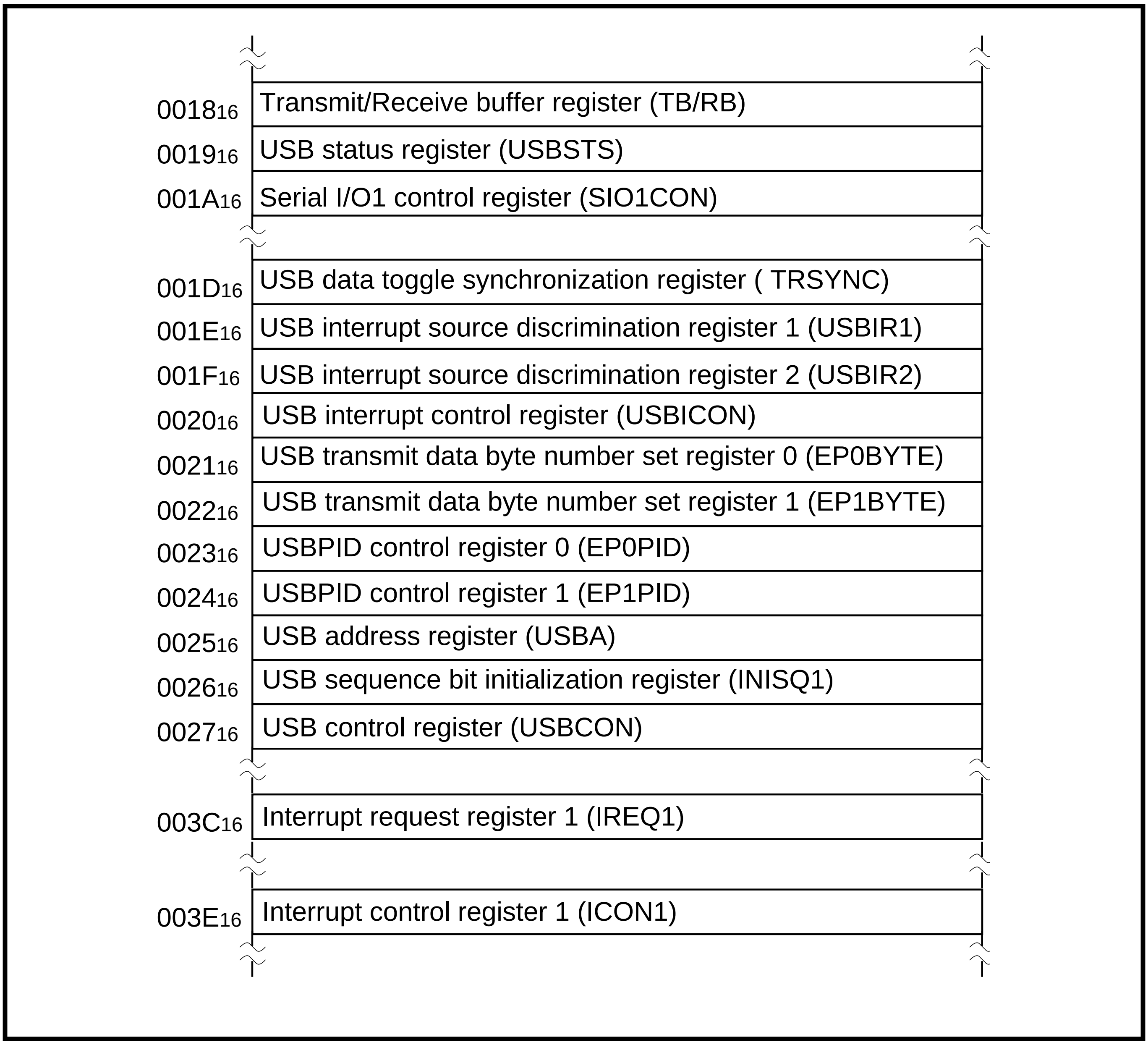
Figure 2.4.4 shows the example of connecting USB (L.S.).



Note: Use the USB connector and USB cable specified to the USB specifications.

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

2.4.2 Memory map



0018_{16}	Transmit/Receive buffer register (TB/RB)
0019_{16}	USB status register (USBSTS)
$001A_{16}$	Serial I/O1 control register (SIO1CON)
$001D_{16}$	USB data toggle synchronization register (TRSYNC)
$001E_{16}$	USB interrupt source discrimination register 1 (USBIR1)
$001F_{16}$	USB interrupt source discrimination register 2 (USBIR2)
0020_{16}	USB interrupt control register (USBICON)
0021_{16}	USB transmit data byte number set register 0 (EP0BYTE)
0022_{16}	USB transmit data byte number set register 1 (EP1BYTE)
0023_{16}	USBPID control register 0 (EP0PID)
0024_{16}	USBPID control register 1 (EP1PID)
0025_{16}	USB address register (USBA)
0026_{16}	USB sequence bit initialization register (INISQ1)
0027_{16}	USB control register (USBCON)
$003C_{16}$	Interrupt request register 1 (IREQ1)
$003E_{16}$	Interrupt control register 1 (ICON1)

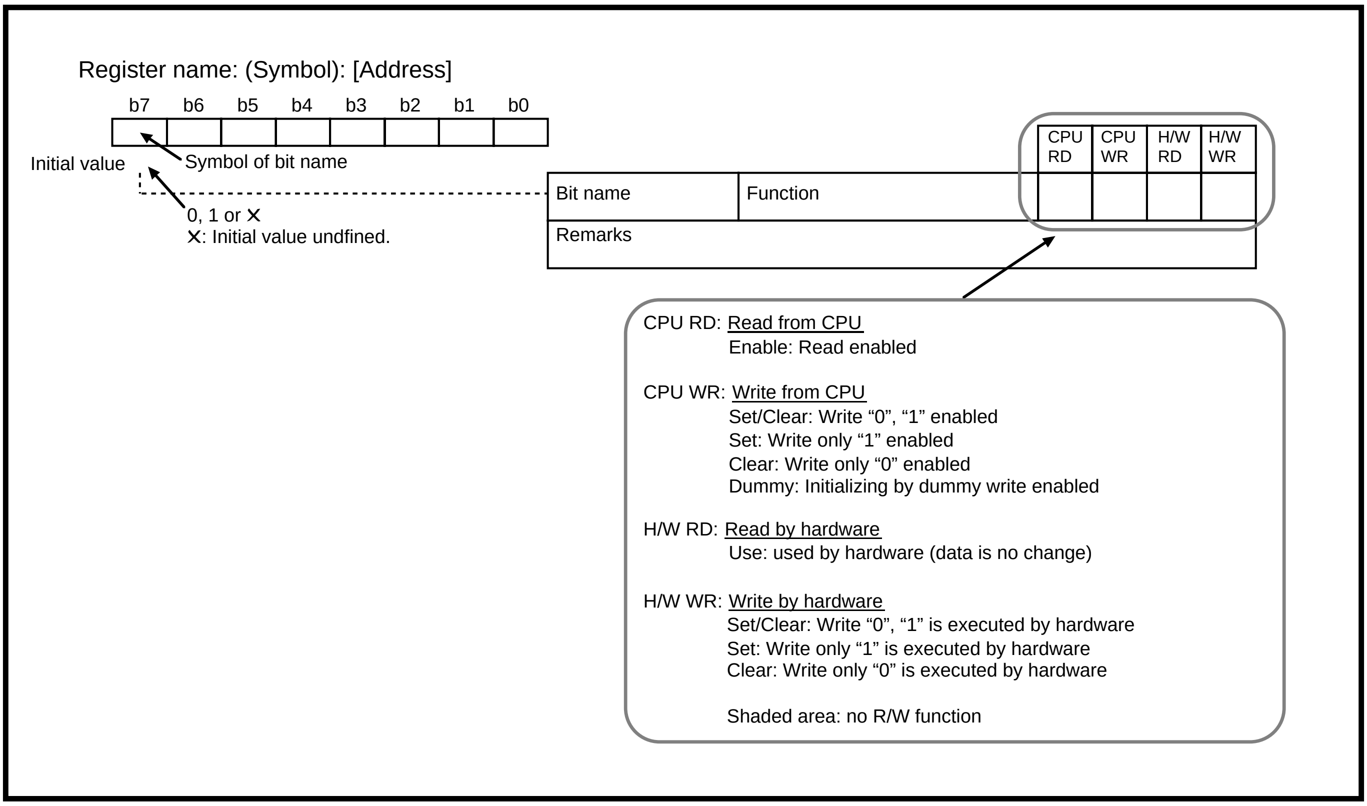
Fig. 2.4.5 Memory map of registers relevant to USB

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.



CPU RD: Read from CPU
Enable: Read enabled

CPU WR: Write from CPU
Set/Clear: Write “0”, “1” enabled
Set: Write only “1” enabled
Clear: Write only “0” enabled
Dummy: Initializing by dummy write enabled

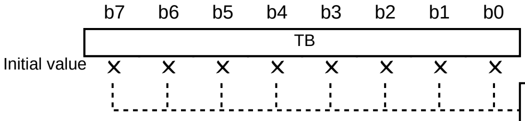
H/W RD: Read by hardware
Use: used by hardware (data is no change)

H/W WR: Write by hardware
Set/Clear: Write “0”, “1” is executed by hardware
Set: Write only “1” is executed by hardware
Clear: Write only “0” is executed by hardware

Shaded area: no R/W function

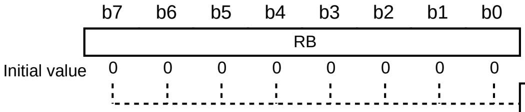
Fig. 2.4.6 Description of the register structure

Transmit buffer register (TB) [Address 18_{16}]



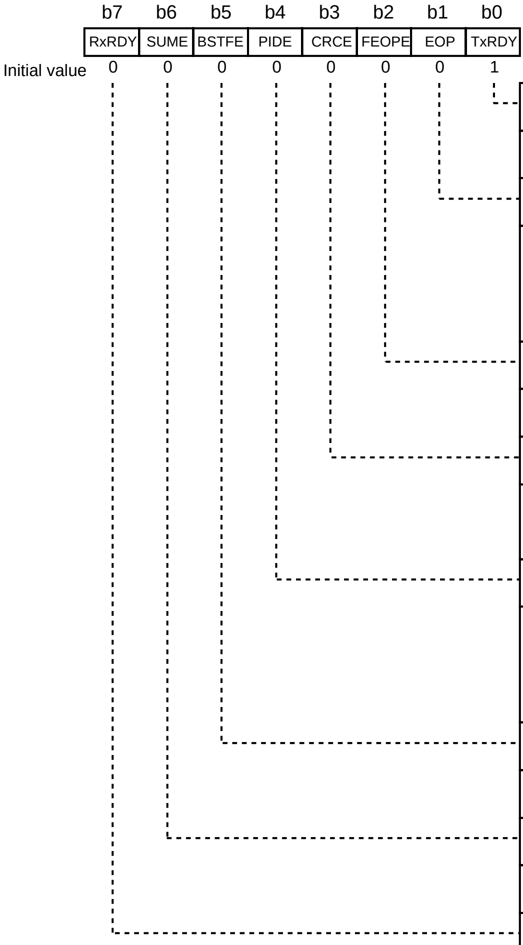
		CPU RD	CPU WR	H/W RD	H/W WR
–	After setting data to address 0018_{16}, the data is transferred to the transmit shift register automatically.		Set/ Clear	Use	
Transmit buffer empty flag (TxRDY) is cleared by writing data to this register.					

Receive buffer register (RB) [Address 18_{16}]



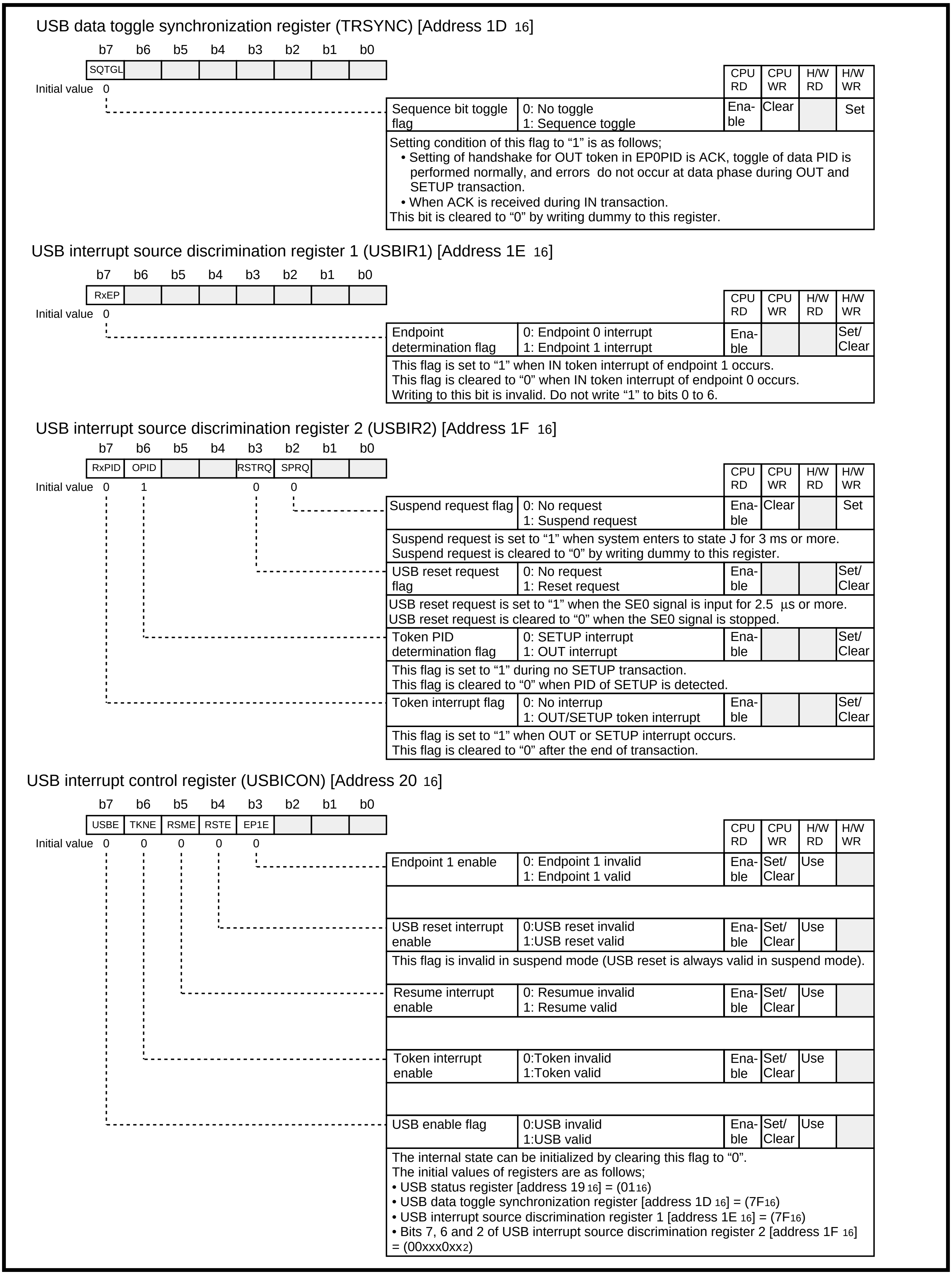
		CPU RD	CPU WR	H/W RD	H/W WR
–	Receive buffer register can be read out by reading data from address 0018_{16}.	Ena- ble			Set/ Clear
Receive buffer full flag (RxRDY) is cleared by reading this register.					

USB status register (USBSTS) [Address 19_{16}]



		CPU RD	CPU WR	H/W RD	H/W WR
Transmit buffer empty flag	0: Buffer full 1: Buffer empty	Ena- ble			Set/ Clear
This bit is set to “1” when data is transferred from buffer to shift register by hardware. This bit is cleared to “0” by writing to buffer.					
EOP detection flag	0: Not detected 1: Detect	Ena- ble	Clear		Set
Setting condition of this flag to “1” is as follows; • Normal EOP detected by hardware • False EOP flag (FEOPE) set • Time is out with EOP not detected at data phase or handshake phase This bit is cleared to “0” by writing dummy to this register.					
False EOP error flag	0: No error 1: False EOP error	Ena- ble	Clear		Set
This bit is set to “1” when the phase is not completed normally. This bit is cleared to “0” by writing dummy to this register.					
CRC error flag	0: No error 1: CRC error	Ena- ble	Clear		Set
This bit is set to “1” when the CRC error occurs at the same timing of EOP detection flag. This bit is to “0” cleared by writing dummy to this register.					
PID error flag	0: No error 1: PID error	Ena- ble	Clear		Set
Setting condition of this flag to “1” is as follows; • PID of DATA0 or DATA1 cannot be detected at data phase after OUT or SETUP token • ACK PID cannot be received at handshake phase during IN transaction. This bit is cleared to “0” by writing dummy to this register.					
Bit stuffing error flag	0: No error 1: Bit stuffing error	Ena- ble	Clear		Set
This bit is set to “1” when bit stuffing error occurs at data phase or handshake phase. This bit is cleared to “0” by writing dummy to this register.					
Summing error flag	0: No error 1: Summing error	Ena- ble	Clear		Set
This bit is set to “1” when any error of FEOPE, CRCE, PIDE, or BSTFE occurs. This bit is cleared to “0” by writing dummy to this register.					
Receive buffer full flag	0: Buffer empty 1: Buffer full	Ena- ble			Set/ Clear
This bit is set to “1” when data is transferred from shift register to buffer by hardware. This bit is cleared to “0” by reading from buffer.					

Fig. 2.4.7 Register structures relevant to USB (1)



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

	b7	b6	b5	b4	b3	b2	b1	b0
	SQTGL							
Initial value	0							

		CPU RD	CPU WR	H/W RD	H/W WR
Sequence bit toggle flag	0: No toggle 1: Sequence toggle	Enable	Clear		Set

Setting condition of this flag to “1” is as follows;
- Setting of handshake for OUT token in EP0PID is ACK, toggle of data PID is performed normally, and errors do not occur at data phase during OUT and SETUP transaction.
- When ACK is received during IN transaction.

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

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

	b7	b6	b5	b4	b3	b2	b1	b0
	RxEP							
Initial value	0							

		CPU RD	CPU WR	H/W RD	H/W WR
Endpoint determination flag	0: Endpoint 0 interrupt 1: Endpoint 1 interrupt	Enable			Set/ Clear

This flag is set to “1” when IN token interrupt of endpoint 1 occurs.
This flag is cleared to “0” when IN token interrupt of endpoint 0 occurs.
Writing to this bit is invalid. Do not write “1” to bits 0 to 6.

USB interrupt source discrimination register 2 (USBIR2) [Address 1F 16]

	b7	b6	b5	b4	b3	b2	b1	b0
	RxPID	OPID			RSTRQ	SPRQ		
Initial value	0	1			0	0		

		CPU RD	CPU WR	H/W RD	H/W WR
Suspend request flag	0: No request 1: Suspend request	Enable	Clear		Set
Suspend request is set to “1” when system enters to state J for 3 ms or more. Suspend request is cleared to “0” by writing dummy to this register.					
USB reset request flag	0: No request 1: Reset request	Enable			Set/ Clear
USB reset request is set to “1” when the SE0 signal is input for 2.5 µs or more. USB reset request is cleared to “0” when the SE0 signal is stopped.					
Token PID determination flag	0: SETUP interrupt 1: OUT interrupt	Enable			Set/ Clear
This flag is set to “1” during no SETUP transaction. This flag is cleared to “0” when PID of SETUP is detected.					
Token interrupt flag	0: No interrup 1: OUT/SETUP token interrupt	Enable			Set/ Clear
This flag is set to “1” when OUT or SETUP interrupt occurs. This flag is cleared to “0” after the end of transaction.					

USB interrupt control register (USBICON) [Address 20 16]

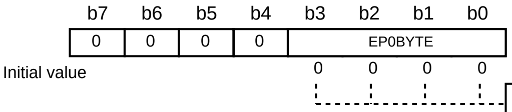
	b7	b6	b5	b4	b3	b2	b1	b0
	USBE	TKNE	RSME	RSTE	EP1E			
Initial value	0	0	0	0	0			

		CPU RD	CPU WR	H/W RD	H/W WR
Endpoint 1 enable	0: Endpoint 1 invalid 1: Endpoint 1 valid	Enable	Set/ Clear	Use	
USB reset interrupt enable	0:USB reset invalid 1:USB reset valid	Enable	Set/ Clear	Use	
This flag is invalid in suspend mode (USB reset is always valid in suspend mode).					
Resume interrupt enable	0: Resumue invalid 1: Resume valid	Enable	Set/ Clear	Use	
Token interrupt enable	0:Token invalid 1:Token valid	Enable	Set/ Clear	Use	
USB enable flag	0:USB invalid 1:USB valid	Enable	Set/ Clear	Use	

The internal state can be initialized by clearing this flag to “0”.
The initial values of registers are as follows;
- USB status register [address 19 16] = (01 16)
- USB data toggle synchronization register [address 1D 16] = (7F 16)
- USB interrupt source discrimination register 1 [address 1E 16] = (7F 16)
- Bits 7, 6 and 2 of USB interrupt source discrimination register 2 [address 1F 16] = (00xxx0xx 2)

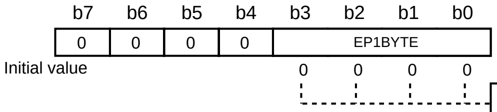
Fig. 2.4.8 Register structures relevant to USB (2)

USB transmit data byte number set register 0 (EP0BYTE) [Address 21 16]



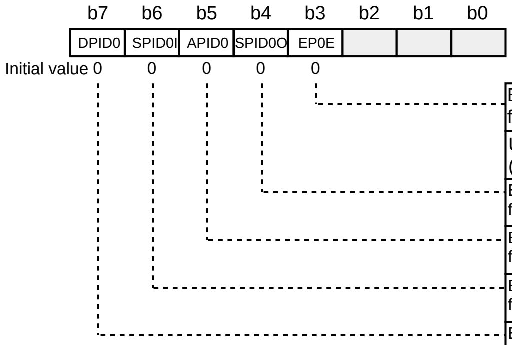
		CPU RD	CPU WR	H/W RD	H/W WR
–	Set the number of data byte for transmitting with endpoint 0	Ena-ble	Set/ Clear	Use	

USB transmit data byte number set register 1 (EP1BYTE) [Address 22 16]



		CPU RD	CPU WR	H/W RD	H/W WR
–	Set the number of data byte for transmitting with endpoint 1	Ena-ble	Set/ Clear	Use	

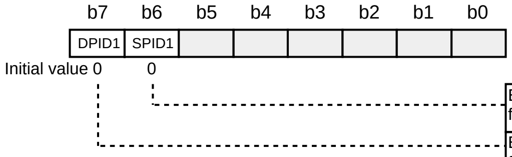
USB PID control register 0 (EP0PID) [Address 23 16]



		CPU RD	CPU WR	H/W RD	H/W WR
Endpoint 0 enable flag	0: Endpoint 0 invalid 1: Endpoint 0 valid	Ena-ble	Set/ Clear	Use	
Unexpected IN or OUT transaction can be ignored by clearing this flag to “0”. (SETUP transaction cannot be ignored, it is always valid.)					
Endpoint 0 PID selection flag (OUT STALL)	1××× : IN token interrupt of DATA 0/1 is valid 01××: STALL handshake is valid for IN token 00××: NAK handshake is valid for IN token ××× 1: STALL handshake is valid for OUT token ××10: ACK handshake is valid for OUT token ××00: NAK handshake is valid for OUT token	Ena-ble	Set/ Clear	Use	
Endpoint 0 PID selection flag (OUT ACK)		Ena-ble	Set/ Clear	Use	
Endpoint 0 PID selection flag (IN STALL)		Ena-ble	Set/ Clear	Use	
Endpoint 0 PID selection flag (IN DATA0/1)		Ena-ble	Set/ Clear	Use	Clear
DPID0 and SPID0I are used to control the response for IN token. DPID0 is used with the token interrupt enable flag (TKNE). DPID0 is cleared to “0” automatically by hardware when ACK is received. SPID0O and APID0 are used to control the response for OUT token. When DPID0 is changed during token packet, the changed value is valid after end of token.					

×: it can be set to 0 or 1.

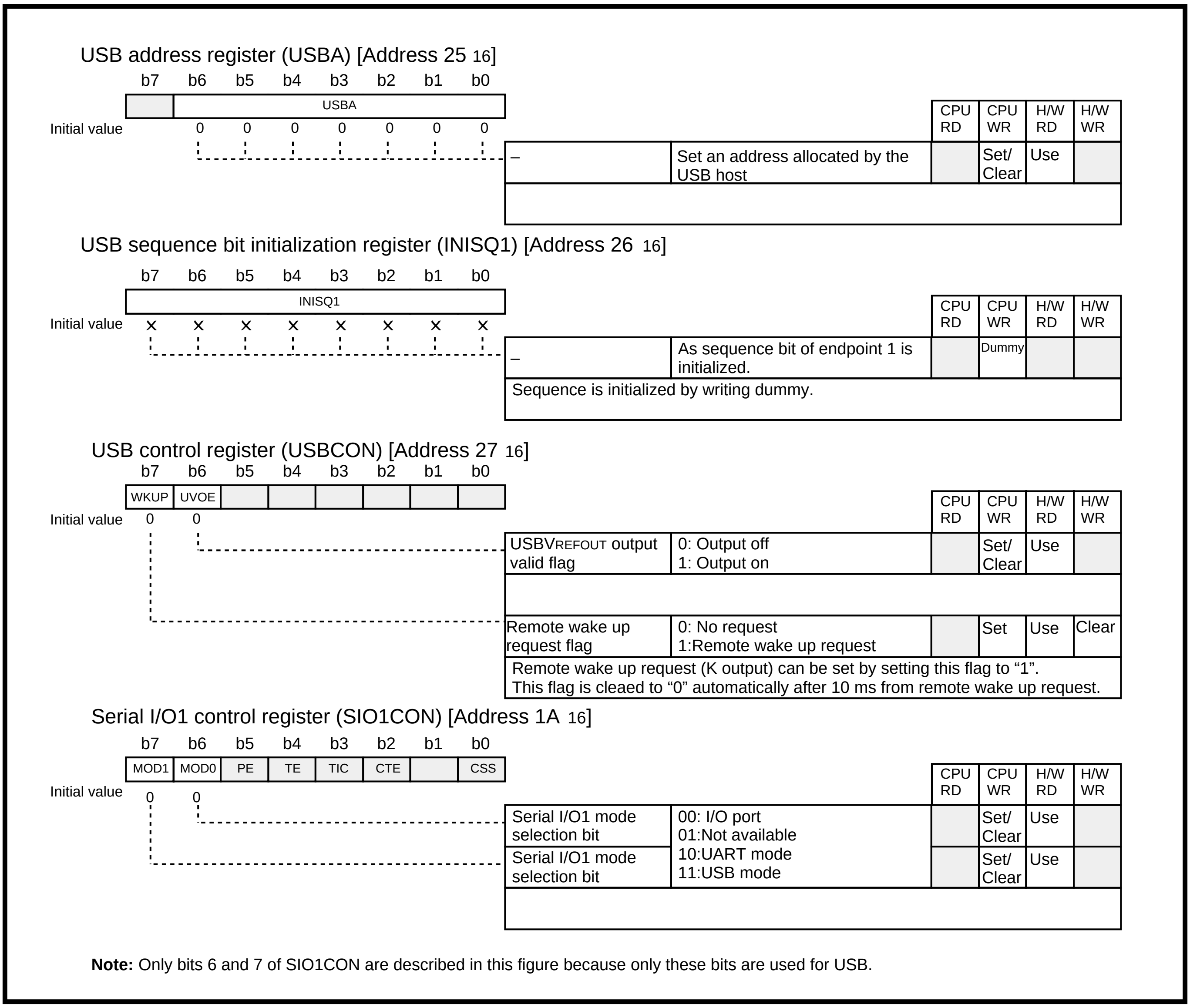
USB PID control register 1 (EP1PID) [Address 24 16]



		CPU RD	CPU WR	H/W RD	H/W WR
Endpoint 1 PID selection flag (IN STALL)	1×: IN token interrupt of DATA0/1 is valid 01: STALL handshake is valid for IN token 00: NAK handshake is valid for IN token	Ena-ble	Set/ Clear	Use	
Endpoint 1 PID selection flag (IN DATA0/1)		Ena-ble	Set/ Clear	Use	Clear
DPID1 and SPID1 are used to control the response for IN token. DPID1 is used with the token interrupt enable flag (TKNE). DPID1 is cleared to “0” automatically by hardware when ACK is received.					

×: it can be set to 0 or 1.

Fig. 2.4.9 Register structures relevant to USB (3)



USB address register (USBA) [Address 25 16]

	b7	b6	b5	b4	b3	b2	b1	b0
		USBA						
Initial value		0	0	0	0	0	0	0

		CPU RD	CPU WR	H/W RD	H/W WR
–	Set an address allocated by the USB host		Set/ Clear	Use	

USB sequence bit initialization register (INISQ1) [Address 26 16]

	b7	b6	b5	b4	b3	b2	b1	b0
	INISQ1							
Initial value	×	×	×	×	×	×	×	×

		CPU RD	CPU WR	H/W RD	H/W WR
–	As sequence bit of endpoint 1 is initialized.		Dummy		
Sequence is initialized by writing dummy.					

USB control register (USBCON) [Address 27 16]

	b7	b6	b5	b4	b3	b2	b1	b0
	WKUP	UVOE						
Initial value	0	0						

		CPU RD	CPU WR	H/W RD	H/W WR
USBVREFOUT output valid flag	0: Output off 1: Output on		Set/ Clear	Use	
Remote wake up request flag	0: No request 1:Remote wake up request		Set	Use	Clear
Remote wake up request (K output) can be set by setting this flag to “1”. This flag is cleaed to “0” automatically after 10 ms from remote wake up request.					

Serial I/O1 control register (SIO1CON) [Address 1A 16]

	b7	b6	b5	b4	b3	b2	b1	b0
	MOD1	MOD0	PE	TE	TIC	CTE		CSS
Initial value	0	0						

		CPU RD	CPU WR	H/W RD	H/W WR
Serial I/O1 mode selection bit	00: I/O port 01:Not available 10:UART mode 11:USB mode		Set/ Clear	Use	
Serial I/O1 mode selection bit			Set/ Clear	Use	

Note: Only bits 6 and 7 of SIO1CON are described in this figure because only these bits are used for USB.

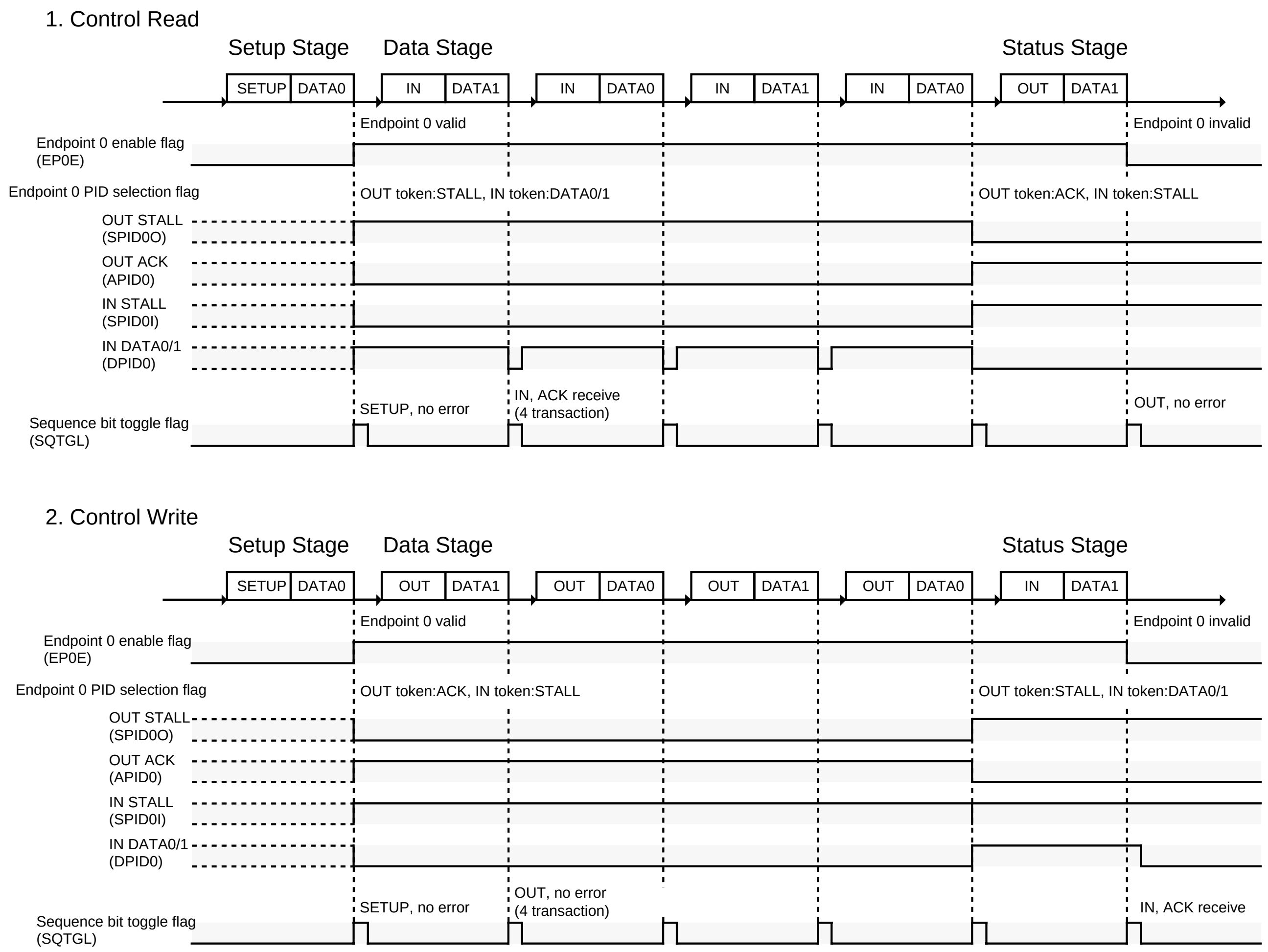
Fig. 2.4.10 Register structures relevant to USB (4)

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.



Notes 1: Only token and DATA0/1 of PID are shown in this figure.
2: In this example, data stage is 4 transaction.

Fig. 2.4.11 Control method of control sequence

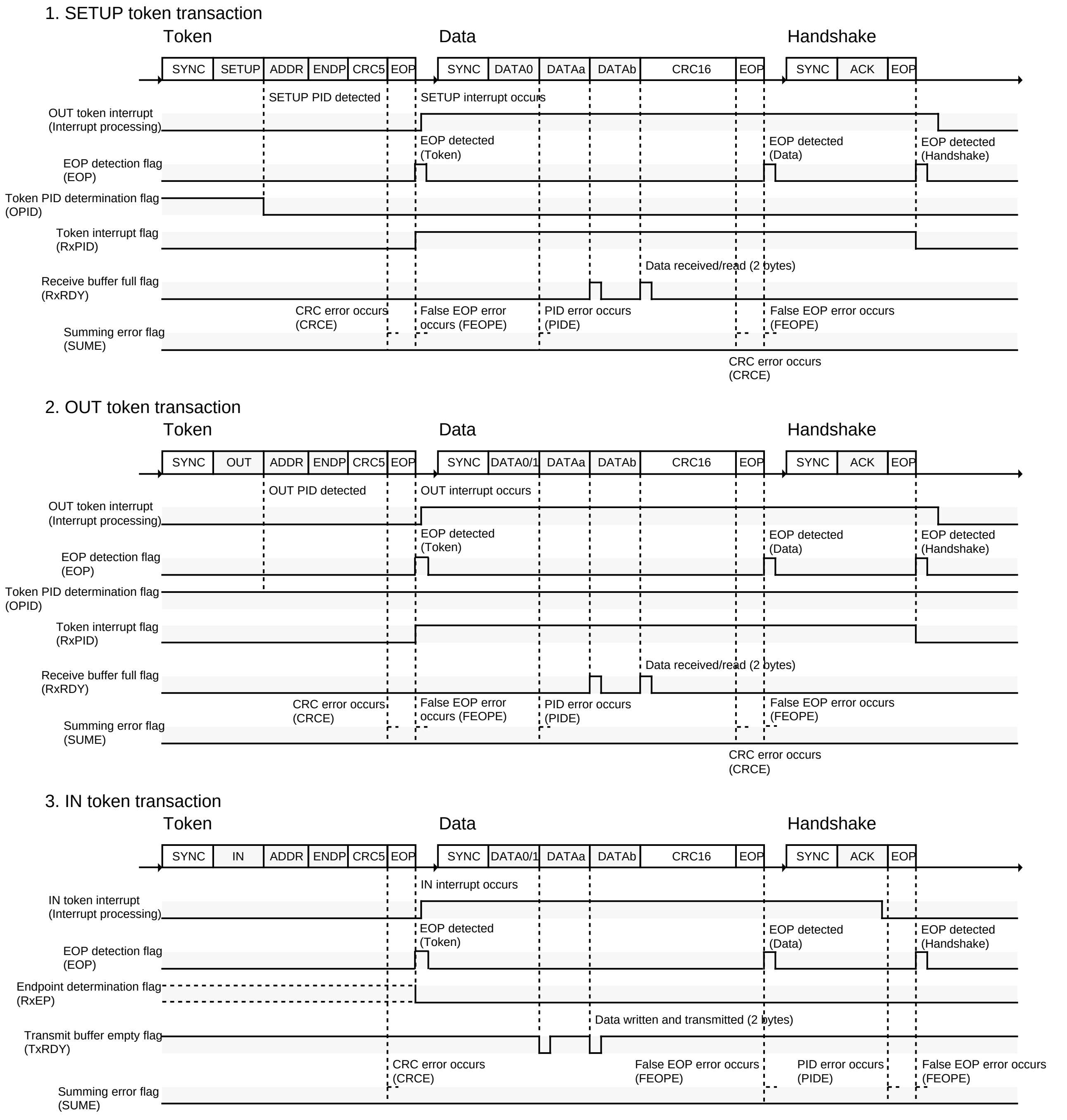
(2) Example of processing each transaction

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.



Notes 1: The data number of PID = DATA0/1 is 2 bytes in this example.
2: Endpoint (ENDP) is 0 in this example.
3: The dot line on SUME shows the timing of each error occur.

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

(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.

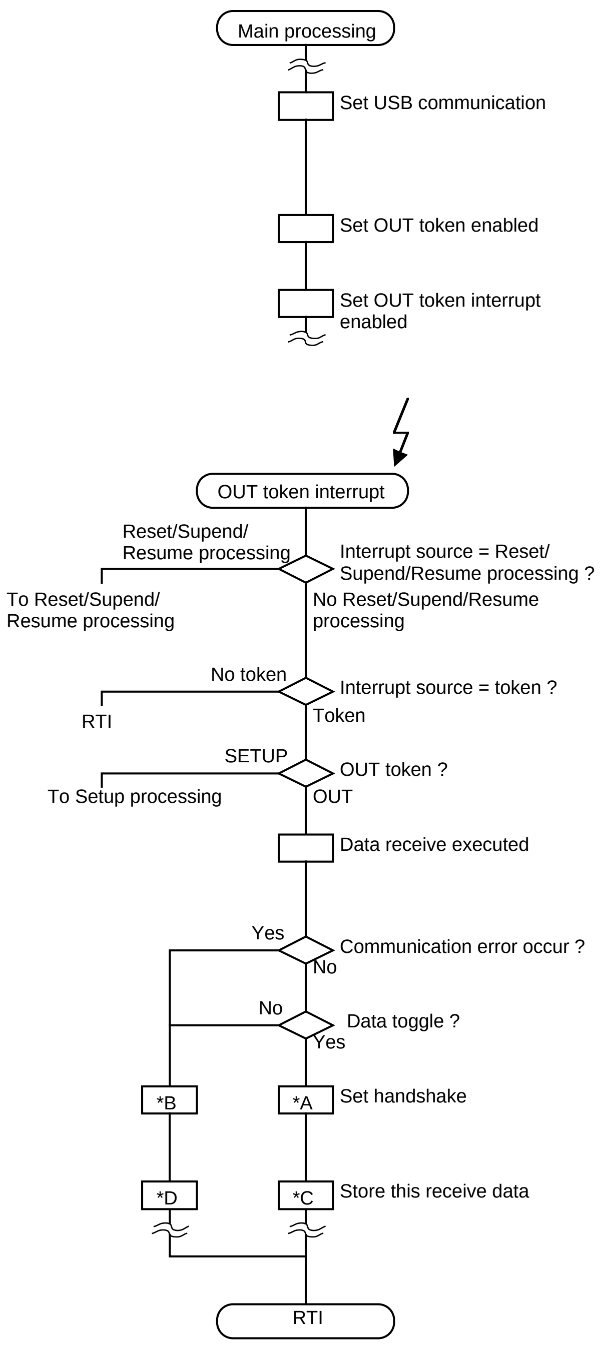
IN token interrupt: IN token (endpoint 0) and IN token (endpoint 1)
OUT token interrupt: OUT token, SETUP token, Reset, Suspend, and Resume

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.

●×: Not used here.
Set it to “0” or “1” arbitrary.

[Flow chart]


Main processing
Set USB communication
Set OUT token enabled
Set OUT token interrupt enabled
OUT token interrupt
Reset/Supend/ Resume processing
Interrupt source = Reset/ Supend/Resume processing ?
To Reset/Supend/ Resume processing
No Reset/Supend/Resume processing
No token
Interrupt source = token ?
RTI
Token
SETUP
OUT token ?
To Setup processing
OUT
Data receive executed
Yes
Communication error occur ?
No
No
Data toggle ?
Yes
*B
*A
Set handshake
*D
*C
Store this receive data
RTI

[Program description]

(USBA	×0000000 ₂	Initial USB address = “0”)
SIO1CON	11×××××× ₂	Set hardware (I/O port) to USB mode
USBCON	×1×××××× ₂	Use USBV$_{REFOUT}$ pin
USBICON	1××××××× ₂	Set internal state to USB enabled
EP0PID	××××1××× ₂	Set endpoint 0 to be valid
USBICON	×1×××××× ₂	Set token interrupt to be valid
(IREQ1	××××××0× ₂	Initialize OUT token interrupt request)
(ICON1	××××××1× ₂	Set OUT token interrupt enable)
USBIR2	××××?××× ₂	Interrupt by reset if ? = “1”
USBIR2	×××××?×× ₂	Interrupt by suspend if ? = “1”
USBICON	××?××××× ₂	Interrupt by resume if ? = “1”
USBIR2	?××××××× ₂	Interrupt by token if ? = “1”
USBIR2	×?×××××× ₂	Interrupt by OUT token if ? = “1”
RB	???????? ₂	Read receive data (Max. 8 bytes in 1 interrupt processing)
USBSTS	×?????×× ₂	No error if all ? = “0” Error/No error can be determined by SUME of bit 6
TRSYNC	?××××××× ₂	Toggle normally if ? = “1”

Note: Set RSME to “1” only in suspend processing

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

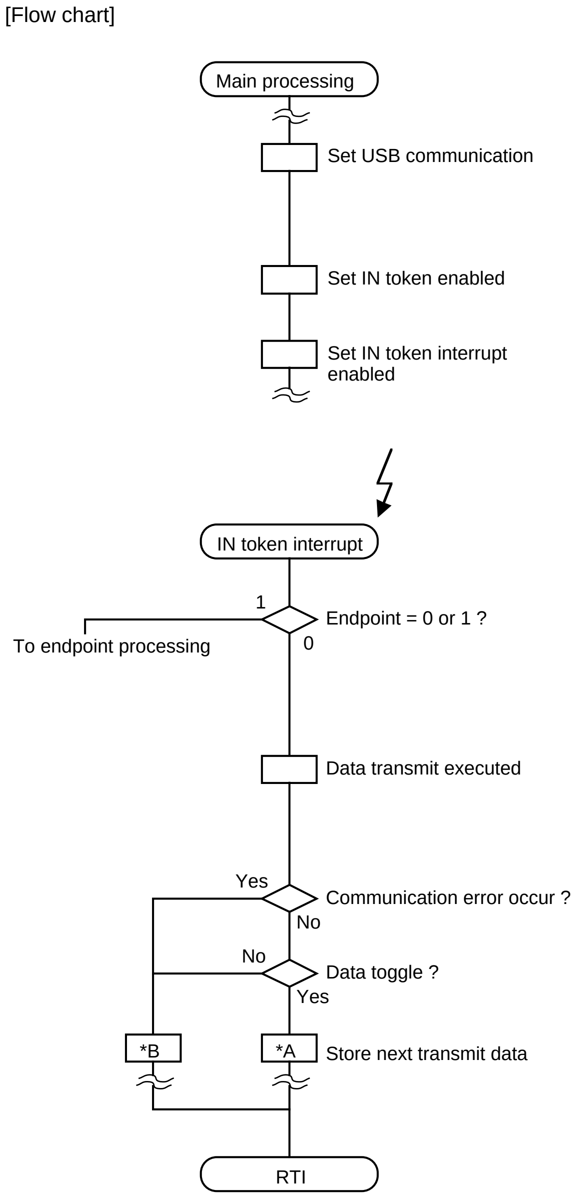
In normal receive, data is stored (*C).
When communication error or data doggle error occur, received data in this time is canceled (*D).

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

Fig. 2.4.13 USB interrupt processing example (OUT token)

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

[Flow chart]



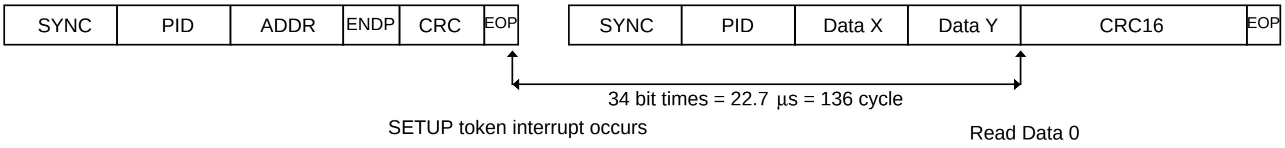
[Program description]

(USBA	X0000000 $_2$	Initial USB address = “0”	)
SIO1CON	11XXXXXX $_2$	Set I/O port to USB mode	
USBCON	X1XXXXXX $_2$	Use USBV$_{REFOUT}$ pin	
USBICON	1XXXXXXX $_2$	Set internal state to USB enabled	
EP0PID	XXXX1XXX $_2$	Set endpoint 0 to be valid	
USBICON	XXXX1XXX$_2$	Set endpoint 1 to be valid	
USBICON	X1XXXXXX $_2$	Set token interrupt to be valid	
(IREQ1	XXXXXXX0 $_2$	Initialize IN token interrupt request	)
(ICON1	XXXXXXX1 $_2$	Set IN token interrupt enable	)
(EP0BYTE	???????? $_2$	Set the number of transmit data in this stage	)
USBIR1	?XXXXXXX $_2$	Endpoint 0 interrupt occurs if ? = “0”	
EP0BYTE	XXXX???? $_2$	Set the number of transmit data byte	
TB	???????? $_2$	Write transmit data (Max. 8 bytes in 1 interrupt processing)	
USBSTS	X?????XX $_2$	No error if all ? = “0” Error/No error can be determined by SUME of bit 6	
TRSYNC	?XXXXXXX $_2$	Toggle normally if ? = “1”	

In normal tramsmit, the next transmit data is set (*A).
When communication error or data doggle error occur, transmit is executed again (*B).

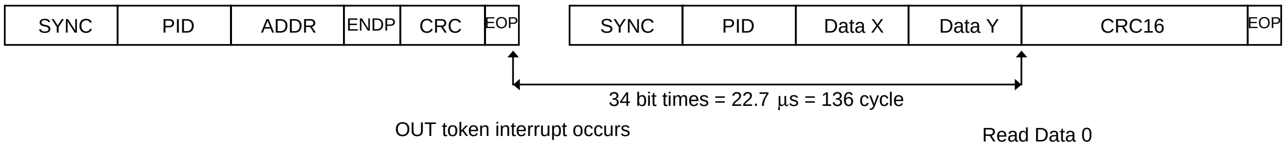
Notes 1: In this figure, only USB in interrupt processing is described.
Note that the storing stack, etc. is not described.
2: IN token interrupt processing of only endpoint 0 is shown in this example.
3: Data shown by ? in program description expresses determination or data setting.

Fig. 2.4.14 USB interrupt processing example (IN token)


SYNC
PID
ADDR
ENDP
CRC
EOP
SYNC
PID
Data X
Data Y
CRC16
EOP
34 bit times = 22.7 µs = 136 cycle
SETUP token interrupt occurs
Read Data 0

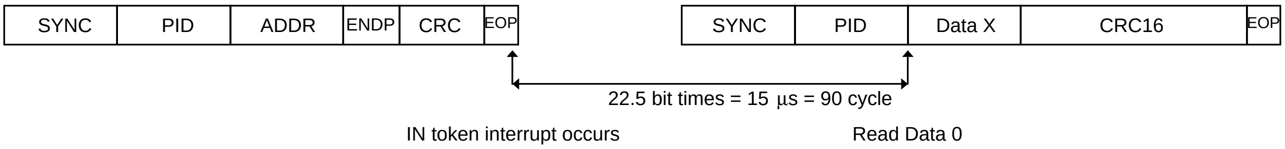
Notes 1: In this case, Data: 2 bytes, Cycle: 6 MHz.
2: Max. 14 cycles are required until the interrupt occurs.

Fig. 2.4.15 Data read timing of SETUP token


SYNC
PID
ADDR
ENDP
CRC
EOP
SYNC
PID
Data X
Data Y
CRC16
EOP
34 bit times = 22.7 µs = 136 cycle
OUT token interrupt occurs
Read Data 0

Notes 1: In this case, Data: 2 bytes, Cycle: 6 MHz.
2: Max. 14 cycles are required until the interrupt occurs.

Fig. 2.4.16 Data read timing of OUT token



Notes 1: In this case, Data: 1 byte, Cycle: 6 MHz.
2: Max. 14 cycles are required until the interrupt occurs.

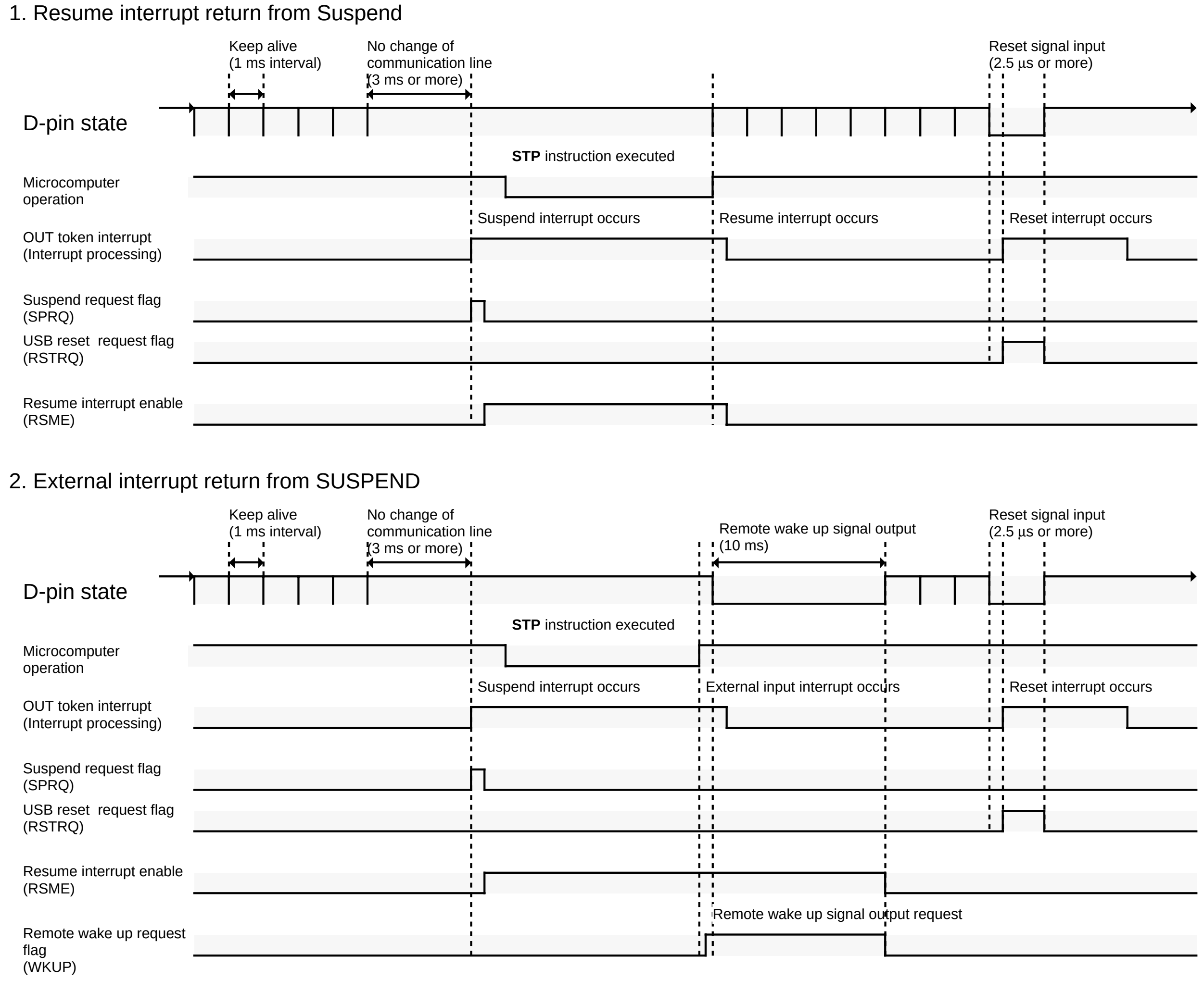
Fig. 2.4.17 Data read timing of IN token (endpoint 0) and IN token (endpoint 1) token

(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.



Notes 1: In this example, USB reset interrupt enable flag (RSTE) = “1” (enabled).
2: The remote wake up request flag (WKUP) is not set to “1” when host does not request the remote wake up signal output.
3: External input interrupts: key on wake up, INT $_0$, INT$_1$ and CNTR$_0$ interrupt.

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.

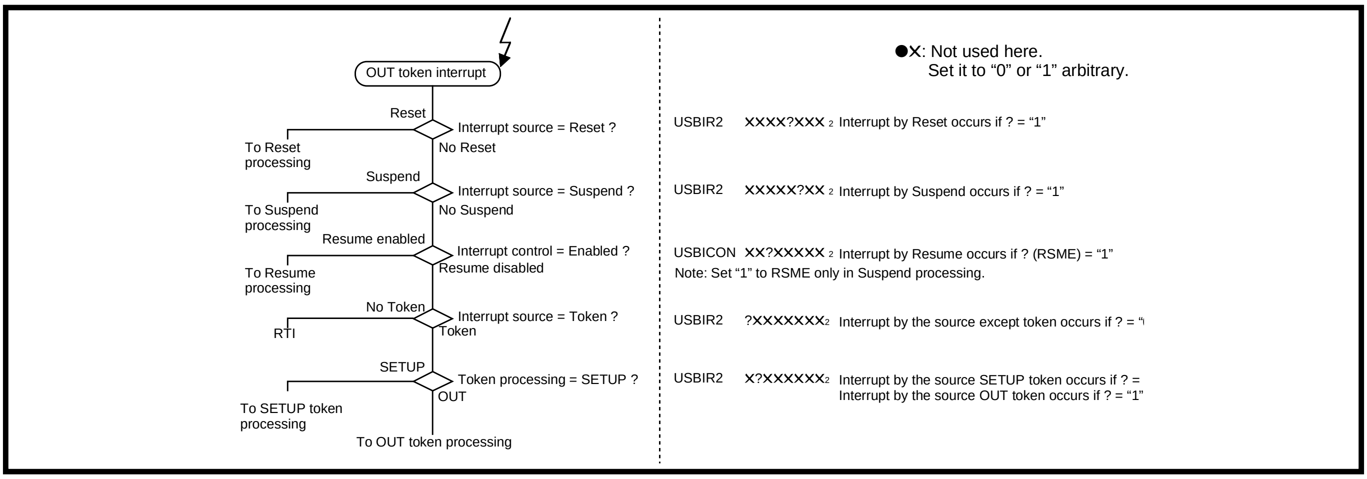

OUT token interrupt
Reset
Interrupt source = Reset ?
No Reset
To Reset processing
Suspend
Interrupt source = Suspend ?
No Suspend
To Suspend processing
Resume enabled
Interrupt control = Enabled ?
Resume disabled
To Resume processing
No Token
Interrupt source = Token ?
Token
RTI
SETUP
Token processing = SETUP ?
OUT
To SETUP token processing
To OUT token processing
●X: Not used here.
Set it to "0" or "1" arbitrary.
USBIR2 XXXX?XXX 2 Interrupt by Reset occurs if ? = "1"
USBIR2 XXXXX?XX 2 Interrupt by Suspend occurs if ? = "1"
USBICON XX?XXXXX 2 Interrupt by Resume occurs if ? (RSME) = "1"
Note: Set "1" to RSME only in Suspend processing.
USBIR2 ?XXXXXXX2 Interrupt by the source except token occurs if ? = "
USBIR2 X?XXXXXX2 Interrupt by the source SETUP token occurs if ? =
Interrupt by the source OUT token occurs if ? = "1"

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 µs).

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

(3) Determination of SE0 signal

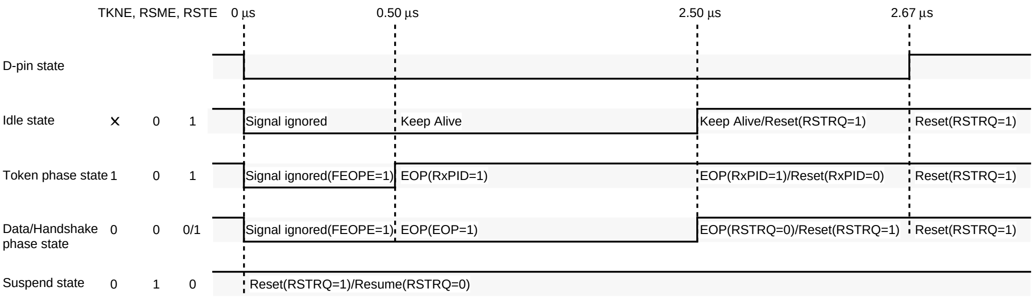
In 7534 Group, USB reset and EOP can be distinguished according to the width of the SE0 signal.

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.



Note: Each active state represents the processing is required in "H" state.

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 $P1_0$, $P1_2$, $P1_3$ input level selection bit (bit 3 of address 17_{16}) 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 $P1_0$, $P1_2$, $P1_3$ 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.

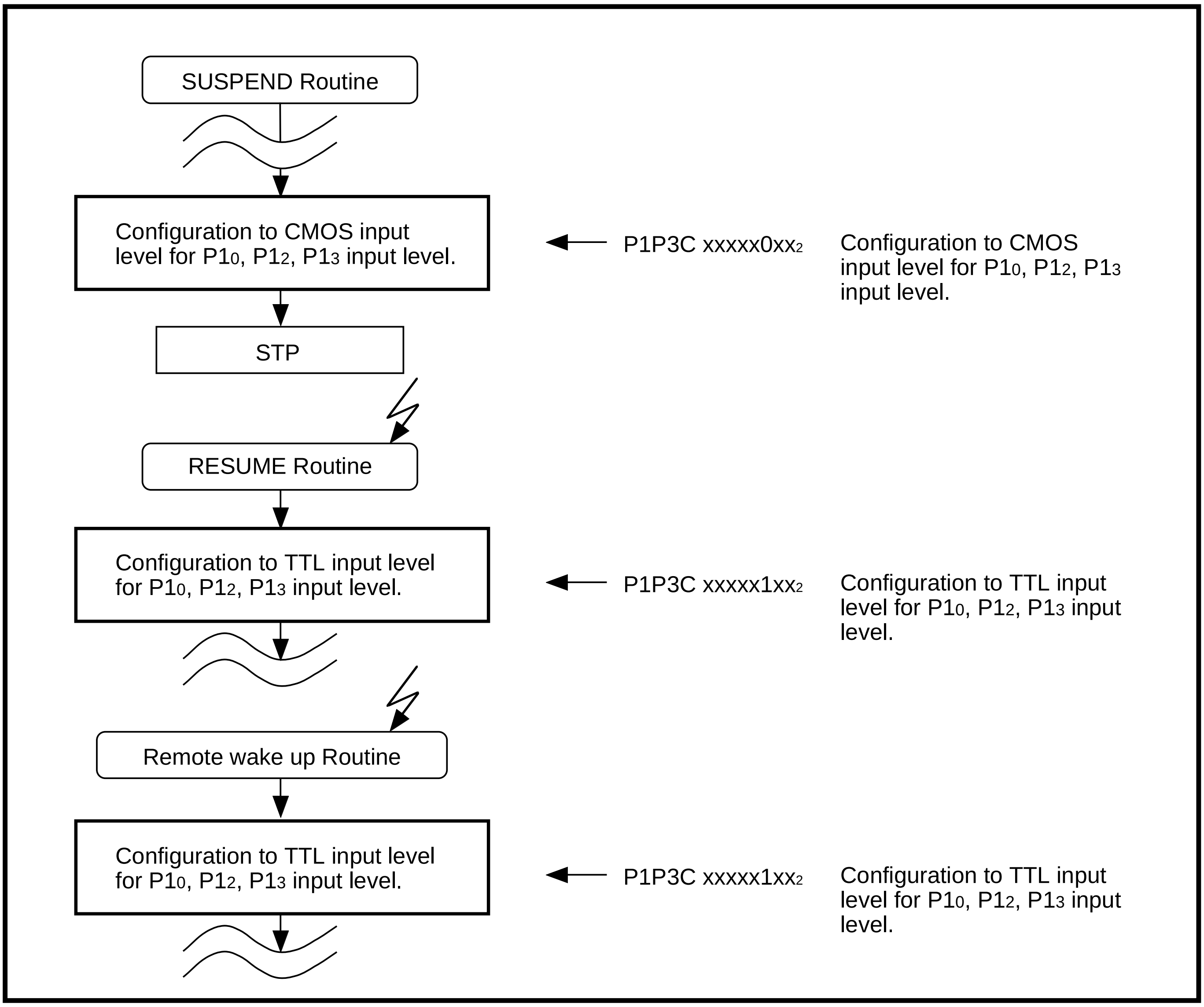

SUSPEND Routine
Configuration to CMOS input level for P10, P12, P13 input level.
P1P3C xxxxx0xx2
Configuration to CMOS input level for P10, P12, P13 input level.
STP
RESUME Routine
Configuration to TTL input level for P10, P12, P13 input level.
P1P3C xxxxx1xx2
Configuration to TTL input level for P10, P12, P13 input level.
Remote wake up Routine
Configuration to TTL input level for P10, P12, P13 input level.
P1P3C xxxxx1xx2
Configuration to TTL input level for P10, P12, P13 input level.

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

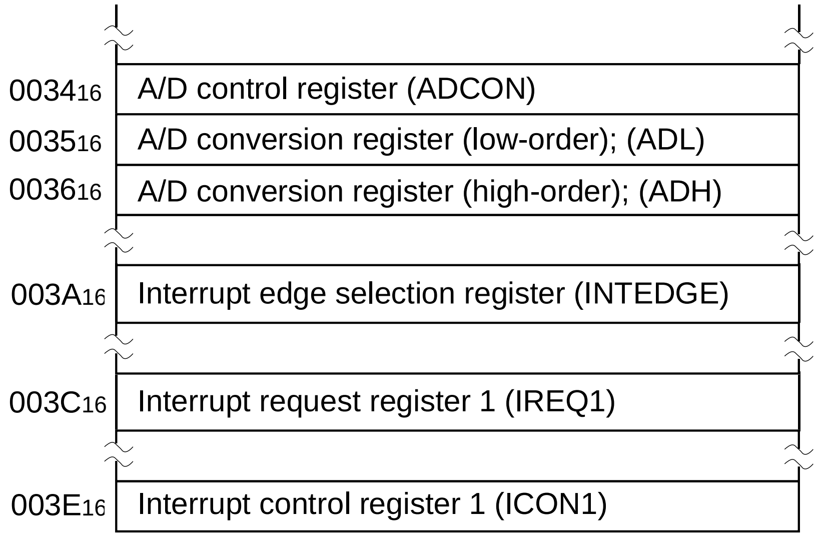
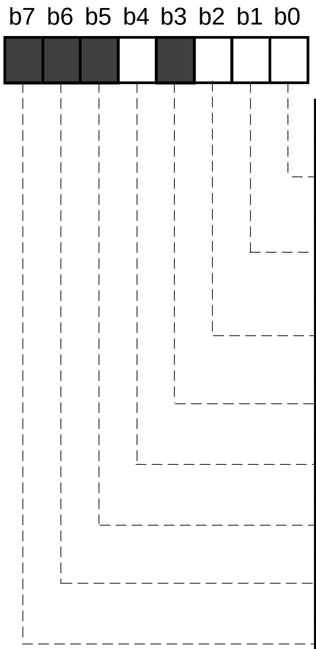

003416 A/D control register (ADCON)
003516 A/D conversion register (low-order); (ADL)
003616 A/D conversion register (high-order); (ADH)
003A16 Interrupt edge selection register (INTEDGE)
003C16 Interrupt request register 1 (IREQ1)
003E16 Interrupt control register 1 (ICON1)

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

2.5.2 Relevant registers

A/D control register



A/D control register (ADCON) [Address : 34_{16}]

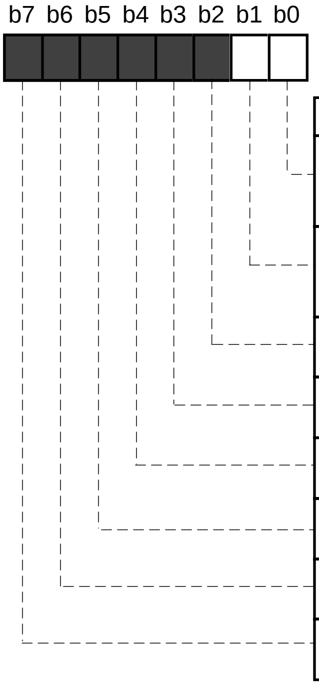
B	Name	Function	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0 : $P2_0/AN_0$ 0 0 1 : $P2_1/AN_1$ 0 1 0 : $P2_2/AN_2$ 0 1 1 : $P2_3/AN_3$ 1 0 0 : $P2_4/AN_4$ 1 0 1 : $P2_5/AN_5$ 1 1 0 : $P2_6/AN_6$ (**Note**) 1 1 1 : $P2_7/AN_7$ (**Note**)	0	○	○
1			0	○	○
2			0	○	○
3	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×
4	AD conversion completion bit	0 : Conversion in progress 1 : Conversion completed	1	○	✻
5	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
6			0	○	×
7			0	○	×

Note: $P2_6/AN_6$, $P2_7/AN_7$ can be selected in the 36-pin and 42-pin versions.

✻: This bit can be cleared to "0" by program, but cannot be set to "1".

Fig. 2.5.2 Structure of A/D control register

A/D conversion register (high-order)

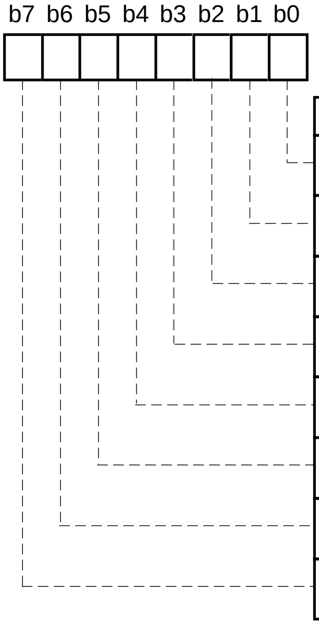


A/D conversion register (high-order) (ADH) [Address : 36_{16}]

B	Function	At reset	R	W
0	The read-only register in which the A/D conversion's results are stored. < 10-bit read> b7 b0 ▨ ▨ ▨ ▨ ▨ ▨ b9 b8	?	○	×
1		?	○	×
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".	?	○	×
3		?	○	×
4		?	○	×
5		?	○	×
6		?	○	×
7		?	○	×

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

A/D conversion register (low-order)

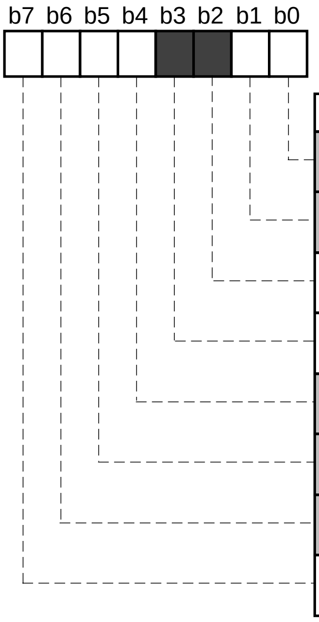


A/D conversion register (low-order) (ADL) [Address : 35_{16}]

B	Function	At reset	R	W
0	The read-only register in which the A/D conversion's results are stored. < 8-bit read> b7 b0 b9 b8 b7 b6 b5 b4 b3 b2 < 10-bit read> b7 b0 b7 b6 b5 b4 b3 b2 b1 b0	?	○	×
1		?	○	×
2		?	○	×
3		?	○	×
4		?	○	×
5		?	○	×
6		?	○	×
7		?	○	×

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

Interrupt edge selection register



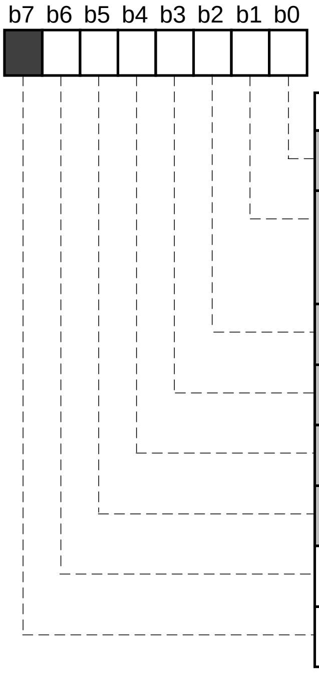
Interrupt edge selection register (INTEDGE) [Address : 3A$_{16}$]

B	Name	Function	At reset	R	W
0	INT$_0$ interrupt edge selection bit (**Note 1**)	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT$_1$ interrupt edge selection bit (**Note 2**)	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
3			0	○	×
4	Serial I/O1 or INT$_1$ interrupt selection bit	0 : Serial I/O1 1 : INT$_1$	0	○	○
5	Timer X or key-on wake up interrupt selection bit	0 : Timer X 1 : Key-on wake up	0	○	○
6	Timer 2 or serial I/O2 interrupt selection bit	0 : Timer 2 1 : Serial I/O2	0	○	○
7	CNTR$_0$ or AD converter interrupt selection bit	0 : CNTR$_0$ 1 : AD converter	0	○	○

Notes 1: 32-pin version: Not used.
This is a write disabled bit. When this bit is read out, the value is "0".
2: 36-pin and 32-pin version: Not used.
This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.5.5 Structure of Interrupt edge selection register

Interrupt request register 1

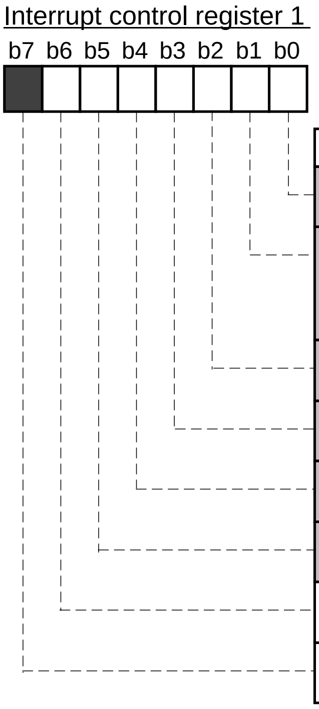


Interrupt request register 1 (IREQ1) [Address : 3C$_{16}$]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	UART transmit/USBSETUP/ OUT token/Reset/Suspend/ Resume/INT$_1$ interrupt request bit (**Note 1**)	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	INT$_0$ interrupt request bit (**Note 2**)	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Timer X or key-on wake up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer 2 or serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	CNTR$_0$ or AD converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

✻: These bits can be cleared to "0" by program, but cannot be set.
Notes 1: 36-pin version and 32-pin version: INT$_1$ interrupt does not exist.
2: 32-pin version: INT$_0$ interrupt does not exist.
This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 2.5.6 Structure of Interrupt request register 1



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

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	UART transmit/USBSETUP/ OUT token/Reset/Suspend/ Resume/INT_1 interrupt enable bit (**Note 1**)	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT_0 interrupt enable bit (**Note 2**)	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Timer X or key-on wake up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer 2 or serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	$CNTR_0$ or AD converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Nothing is allocated for this bit. Do not write “1” to this bit. When this bit is read out, the value is “0”.		0	○	×

Notes 1: 36-pin version and 32-pin version: INT_1 interrupt does not exist.
2: 32-pin version: INT_0 interrupt does not exist.
This is a write disabled bit. When this bit is read out, the value is “0”.

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.

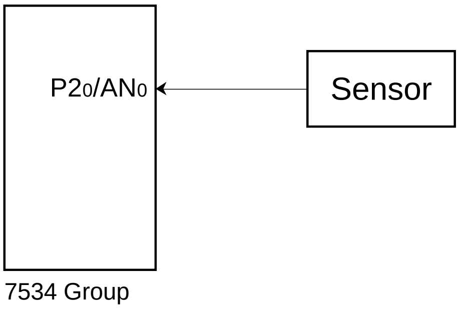
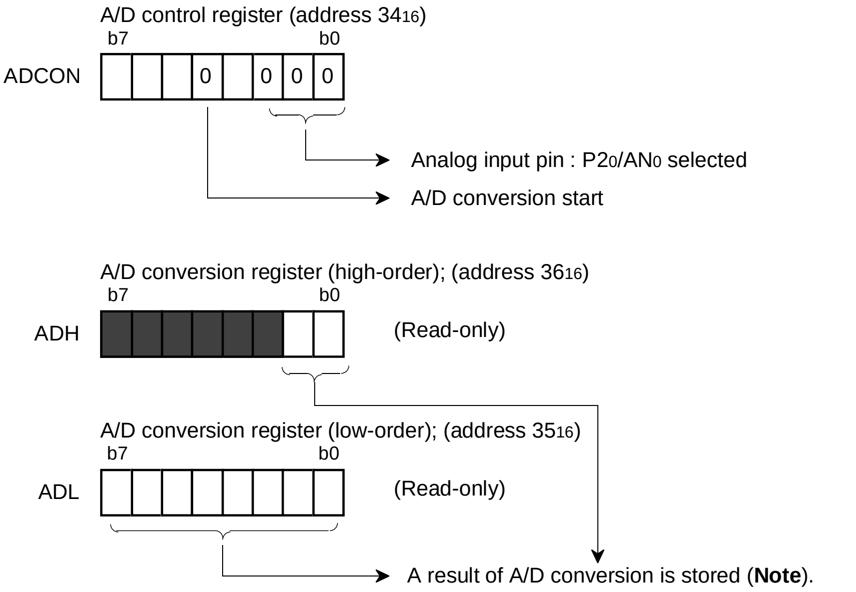


Fig. 2.5.8 Connection diagram

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


A/D control register (address 34₁₆)
b7
b0
ADCON
0
0
0
0
Analog input pin : P2₀/AN₀ selected
A/D conversion start
A/D conversion register (high-order); (address 36₁₆)
b7
b0
ADH
(Read-only)
A/D conversion register (low-order); (address 35₁₆)
b7
b0
ADL
(Read-only)
A result of A/D conversion is stored (Note).

Note: After bit 4 of ADCON is set to “1”, read out that contents.
When reading 10-bit data, read address 0036₁₆ before address 0035₁₆;
when reading 8-bit data, read address 0035₁₆ only.

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.

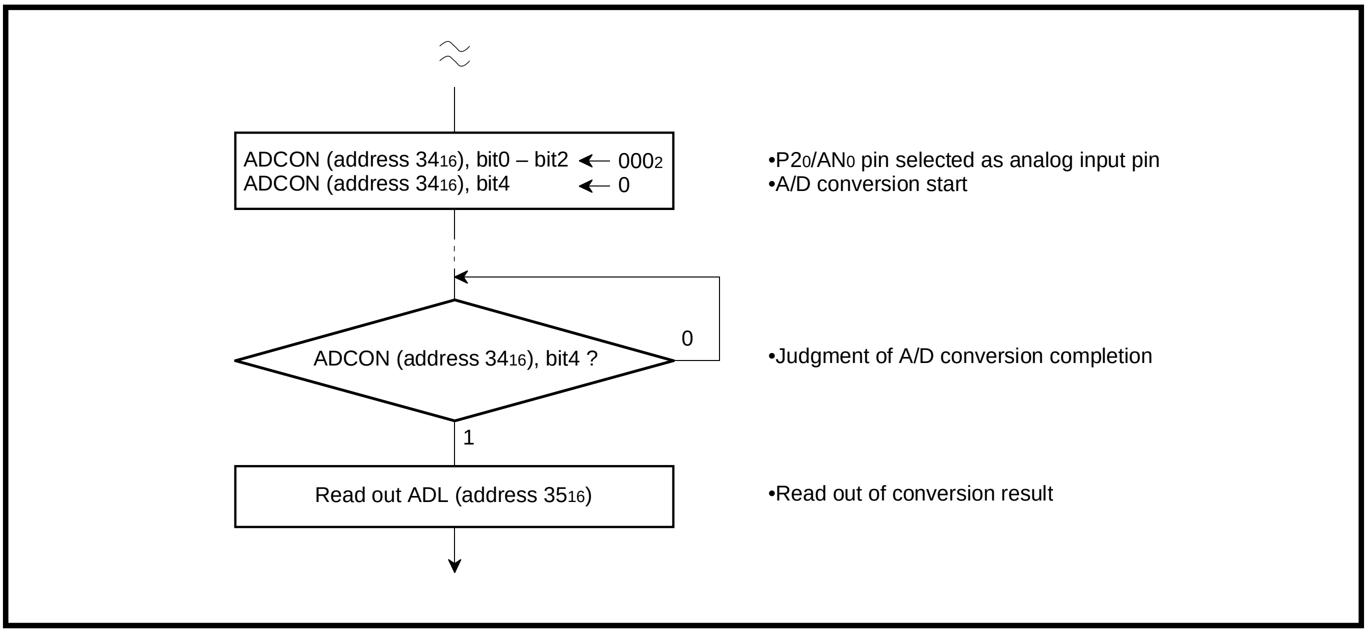

ADCON (address 3416), bit0 – bit2 ← 0002
ADCON (address 3416), bit4 ← 0
•P20/AN0 pin selected as analog input pin
•A/D conversion start
ADCON (address 3416), bit4 ?
0
•Judgment of A/D conversion completion
1
Read out ADL (address 3516)
•Read out of conversion result

Fig. 2.5.10 Control procedure for 8-bit read

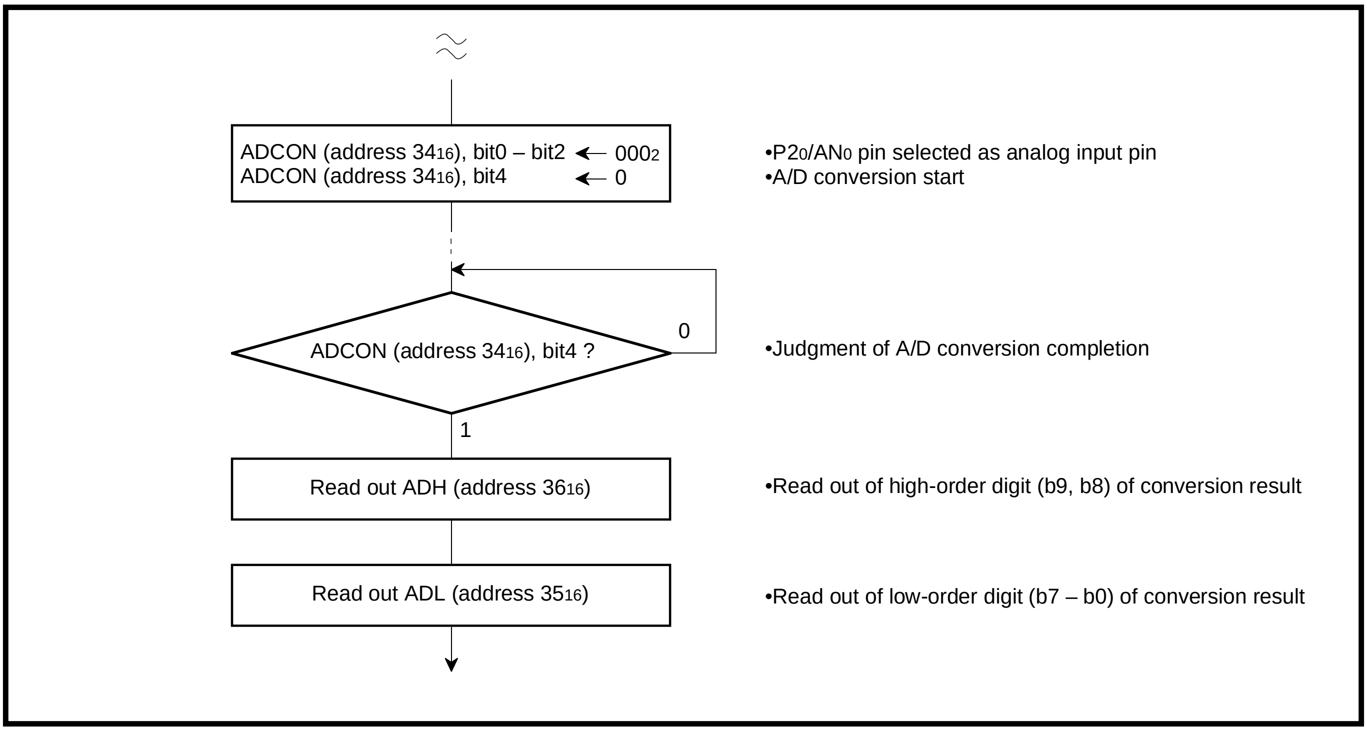

ADCON (address 3416), bit0 – bit2 ← 0002
ADCON (address 3416), bit4 ← 0
•P20/AN0 pin selected as analog input pin
•A/D conversion start
ADCON (address 3416), bit4 ?
0
•Judgment of A/D conversion completion
1
Read out ADH (address 3616)
•Read out of high-order digit (b9, b8) of conversion result
Read out ADL (address 3516)
•Read out of low-order digit (b7 – b0) of conversion result

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 0.01µF to 1µF. 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 V_{SS}.
*1: Power supplied by USB V_{CC} BUS.

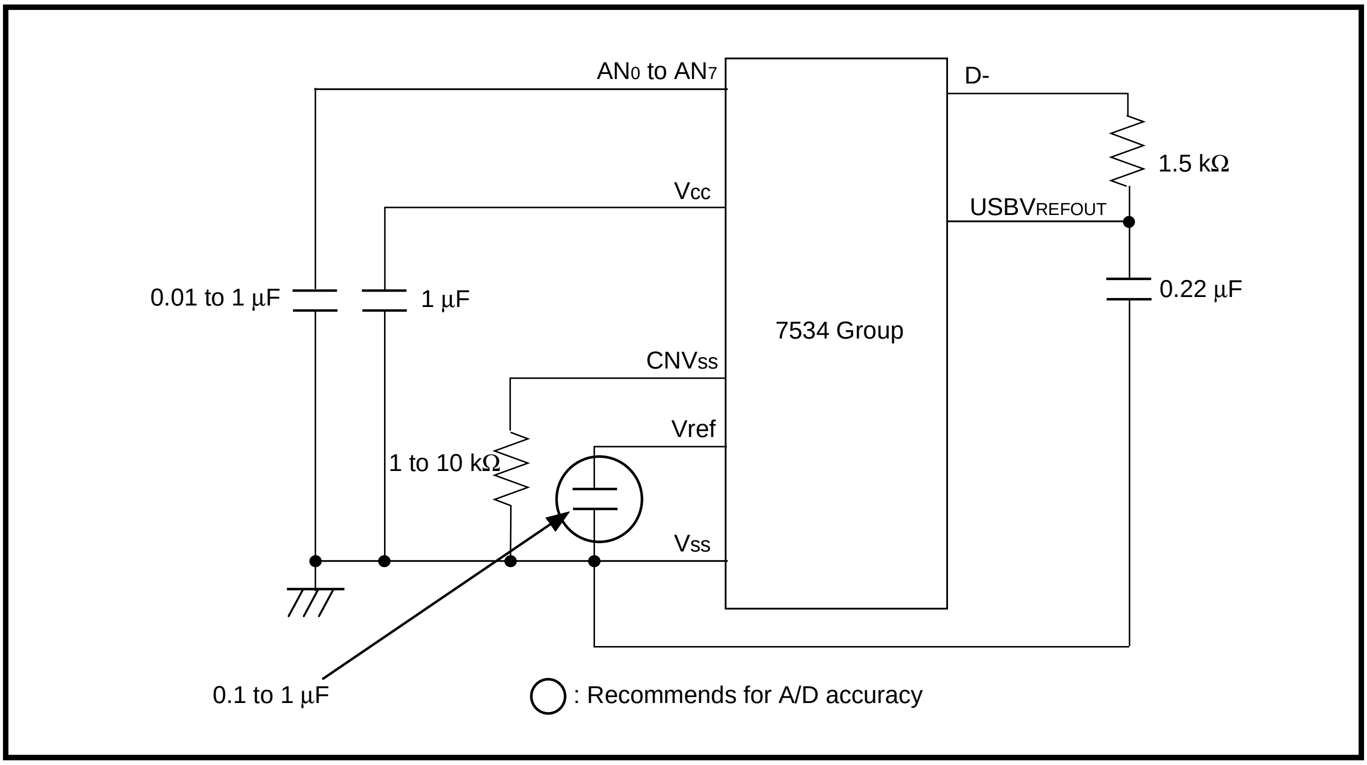


Fig. 2.5.12 Method to stabilize A/D conversion accuracy

(b) It is recommended for A/D accuracy 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.

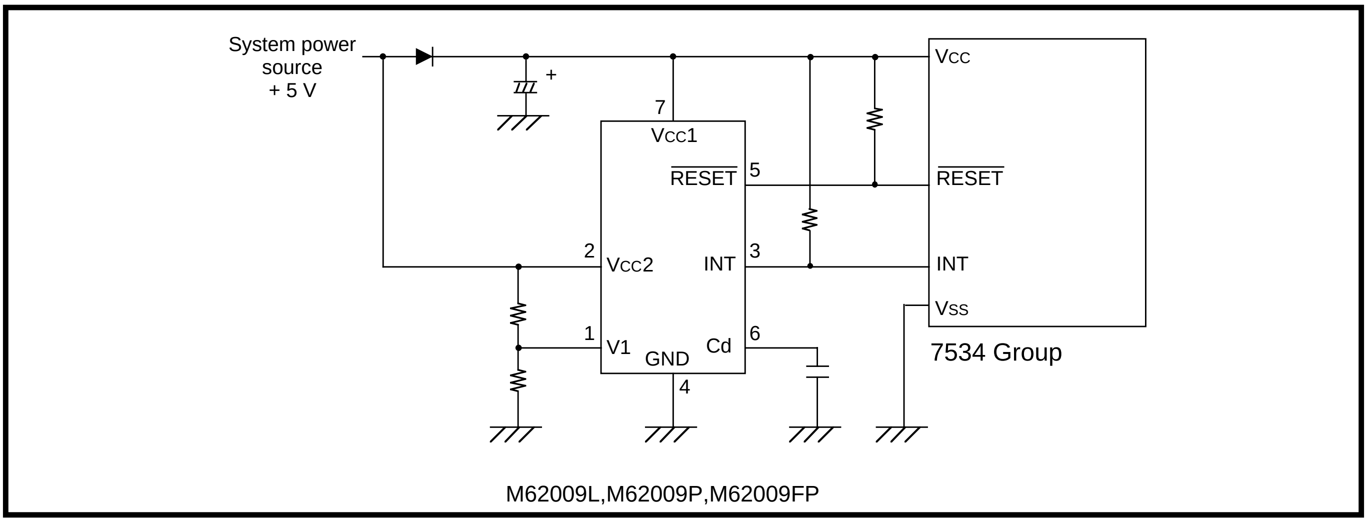


Fig. 2.6.1 Example of poweron reset circuit

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

3.1.1 Absolute maximum ratings

Table 3.1.1 Absolute maximum ratings

Symbol	Parameter		Conditions	Ratings	Unit
V_{CC}	Power source voltage		All voltages are based on V_{SS}. Output transistors are cut off.	–0.3 to 7.0	V
V_I	Input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_6$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, V_{REF}, $P4_0$, $P4_1$		–0.3 to V_{CC} + 0.3	V
V_I	Input voltage	$\overline{RESET}$, X_{IN}		–0.3 to V_{CC} + 0.3	V
V_I	Input voltage	CNV_{SS} (Note 1)		–0.3 to 13	V
V_O	Output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_6$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, X_{OUT}, $USBV_{REFOUT}$, $P4_0$, $P4_1$		–0.3 to V_{CC} + 0.3	V
P_d	Power dissipation	(Note 2)	Ta = 25°C	1000 (Note 3)	mW
T_{opr}	Operating temperature			–20 to 85	°C
T_{stg}	Storage temperature			–40 to 125	°C

Notes 1: It is a rating only for the One Time PROM version. Connect to V_{SS} 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, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Power source voltage	f(X_{IN}) = 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	$\overline{\text{RESET}}$, X_{IN}	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	X_{IN}	0		0.16V_{CC}	V
$\Sigma I_{OH(peak)}$	“H” total peak output current (Note 1)	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41			–80	mA
$\Sigma I_{OL(peak)}$	“L” total peak output current (Note 1)	P00–P07, P10–P16, P20–P27, P37, P40, P41			80	mA
$\Sigma I_{OL(peak)}$	“L” total peak output current (Note 1)	P30–P36			60	mA
$\Sigma I_{OH(avg)}$	“H” total average output current (Note 1)	P00–P07, P10–P16, P20–P27, P30–P37, P40, P41			–40	mA
$\Sigma I_{OL(avg)}$	“L” total average output current (Note 1)	P00–P07, P10–P16, P20–P27, P37, P40, P41			40	mA
$\Sigma I_{OL(avg)}$	“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(X_{IN})	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	$\overline{\text{RESET}}$			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	$\overline{\text{RESET}}$	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	$\overline{\text{RESET}}$, 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.

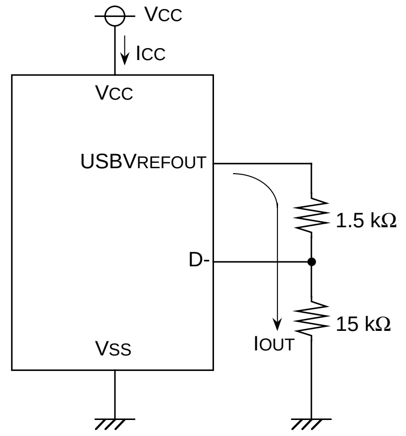
Table 3.1.4 Electrical characteristics (2) (V_{CC} = 4.1 to 5.5 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

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”	Ta = 25 °C		0.1	1.0	µA
			Ta = 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)	Ta = 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, Ta = –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 Ta = 25 °C			±3	LSB
—	Differential nonlinear error	V_{CC} = 4.1 to 5.5 V Ta = 25 °C			±0.9	LSB
V_{OT}	Zero transition voltage	V_{CC} = V_{REF} = 5.12 V	0	5	20	mV
V_{FST}	Full scale transition voltage	V_{CC} = V_{REF} = 5.12 V	5105	5115	5125	mV
t_{CONV}	Conversion time				122	tc(X_{IN})
R_{LADDER}	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



I_{OUT} is included to this ratings.

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, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W}(\overline{RESET})$	Reset input “L” pulse width	15			µs
$t_{C}(X_{IN})$	External clock input cycle time	166			ns
$t_{WH}(X_{IN})$	External clock input “H” pulse width	70			ns
$t_{WL}(X_{IN})$	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
$t_{C}(S_{CLK})$	Serial I/O2 clock input cycle time	1000			ns
$t_{WH}(S_{CLK})$	Serial I/O2 clock input “H” pulse width	400			ns
$t_{WL}(S_{CLK})$	Serial I/O2 clock input “L” pulse width	400			ns
$t_{su}(S_{DATA}–S_{CLK})$	Serial I/O2 input set up time	200			ns
$t_{h}(S_{CLK}–S_{DATA})$	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, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{WH}(S_{CLK})$	Serial I/O2 clock output “H” pulse width	$t_{C}(S_{CLK})$/2–30			ns
$t_{WL}(S_{CLK})$	Serial I/O2 clock output “L” pulse width	$t_{C}(S_{CLK})$/2–30			ns
$t_{d}(S_{CLK}–S_{DATA})$	Serial I/O2 output delay time			140	ns
$t_{v}(S_{CLK}–S_{DATA})$	Serial I/O2 output valid time	0			ns
$t_{r}(S_{CLK})$	Serial I/O2 clock output rising time			30	ns
$t_{f}(S_{CLK})$	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, Ta = 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, Ta = 0 to 70 °C, V$_{CC}$ = 4.4 to 5.25 V	75	150	300	ns

Notes: X$_{OUT}$ pin is excluded.

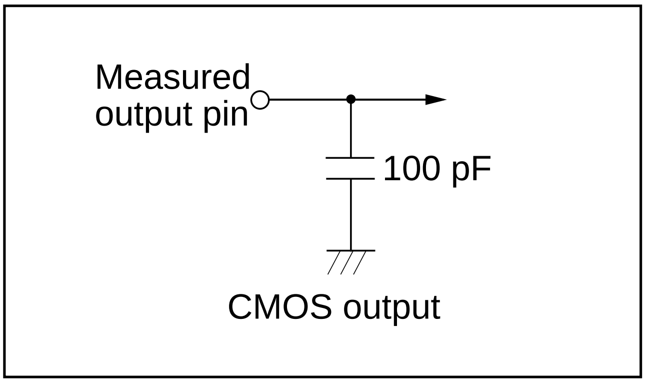


Fig. 3.1.2 Output switching characteristics measurement circuit

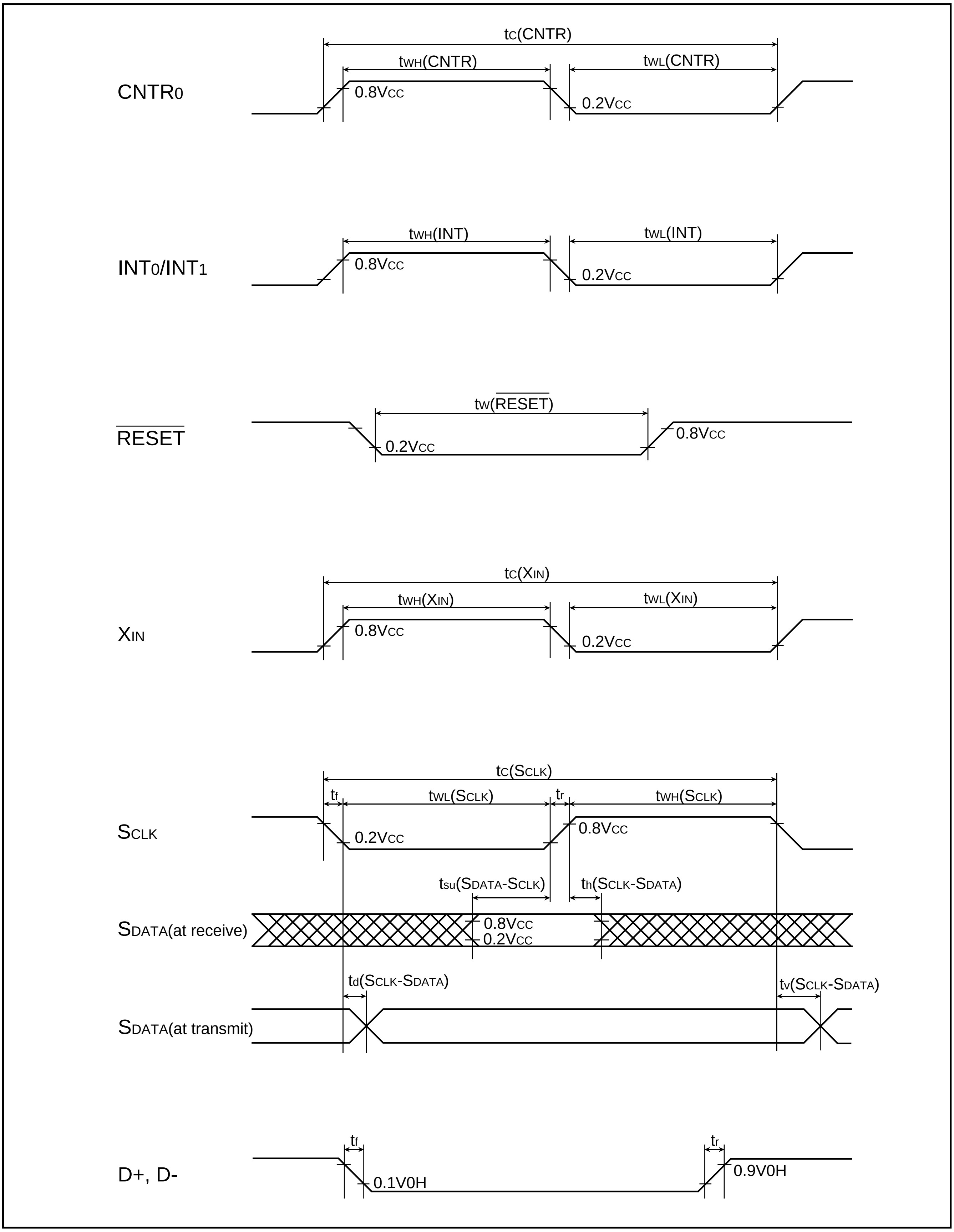


Fig. 3.1.3 Timing chart

3.2 Typical characteristics

3.2.1 Power source current characteristic example (ICC-VCC characteristic)

Measuring condition: Typical sample, Ta = 25 °C, ceramic oscillation, when operating system in double-speed mode (A/D conversion not executed)

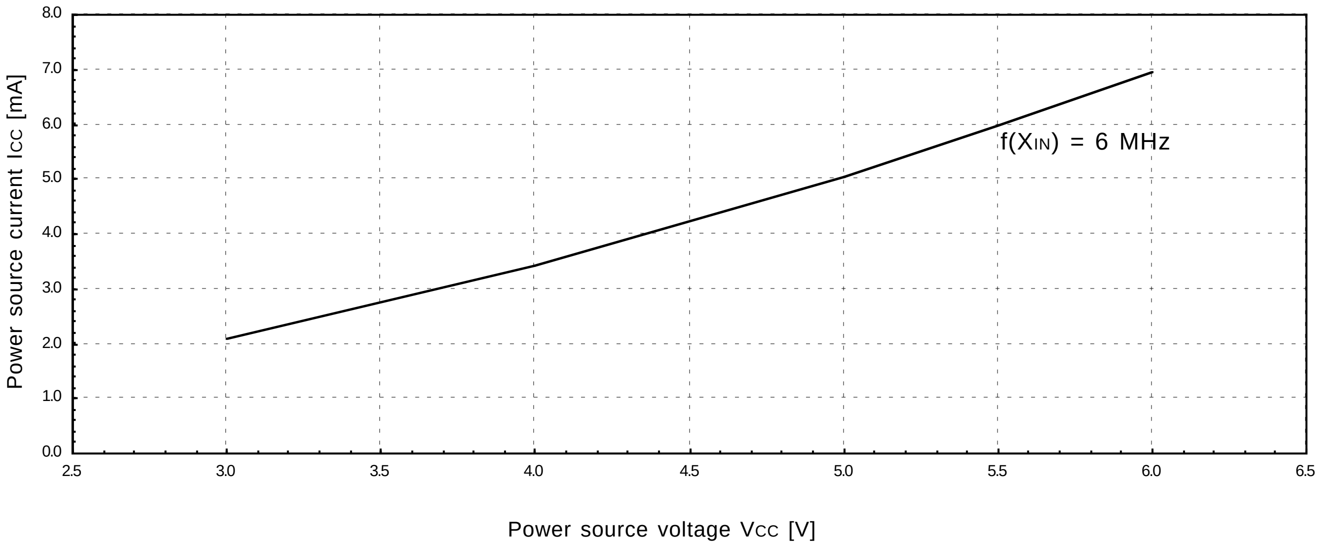


Fig. 3.2.1 ICC-VCC characteristic example (in double-speed mode)

Measuring condition: Typical sample, Ta = 25 °C, ceramic oscillation, At **WIT** instruction execution (at wait)

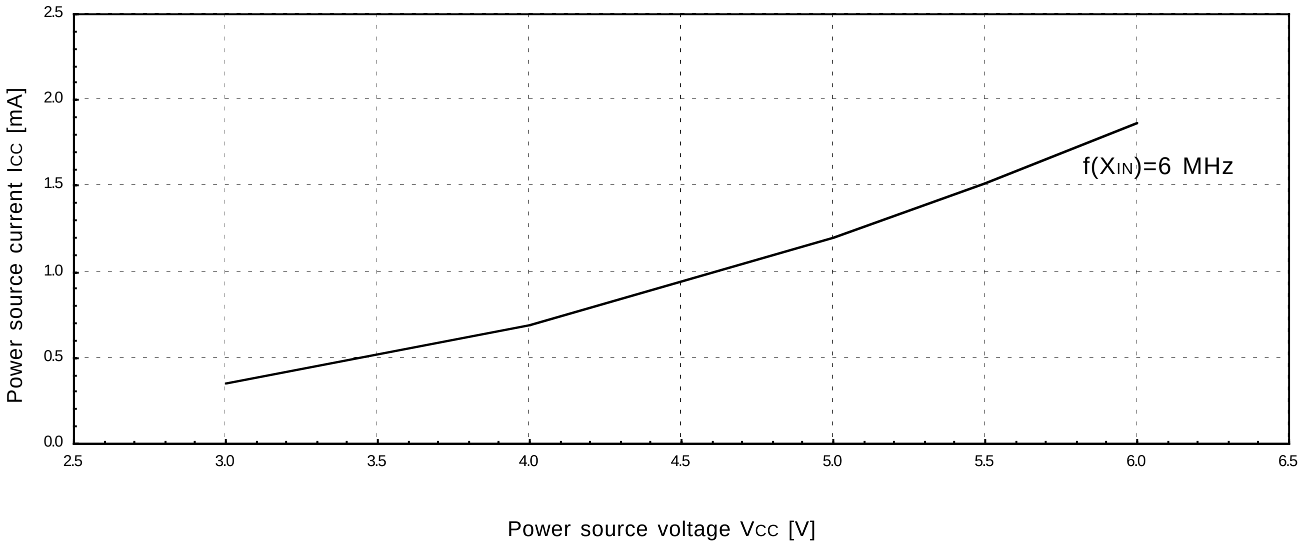


Fig. 3.2.2 ICC-VCC characteristic example (at WIT instruction execution)

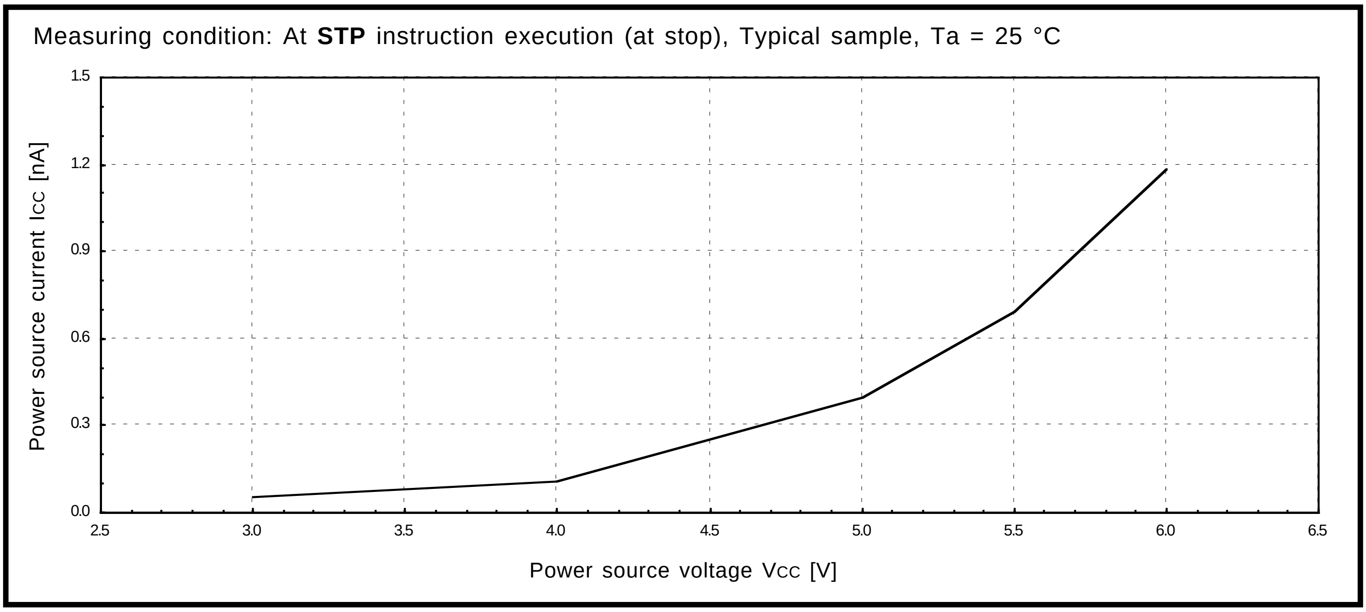


Fig. 3.2.3 ICC-VCC characteristic example (At STP instruction execution, Ta = 25 °C)

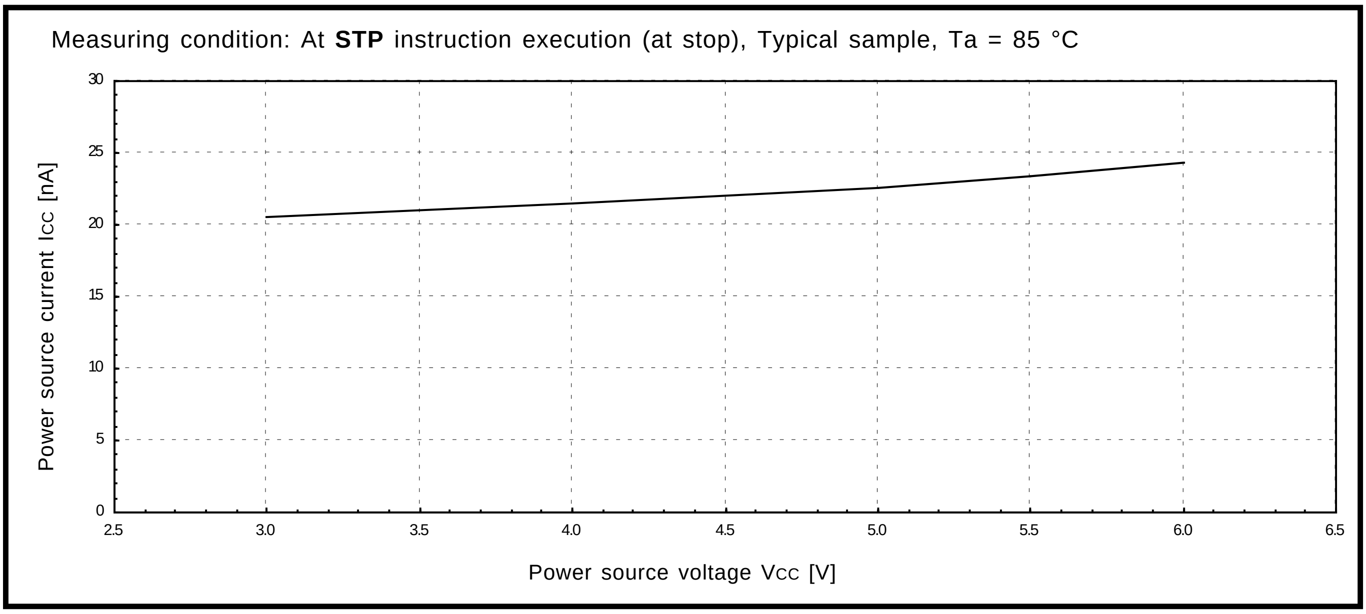


Fig. 3.2.4 ICC-VCC characteristic example (At STP instruction execution, Ta = 85 °C)

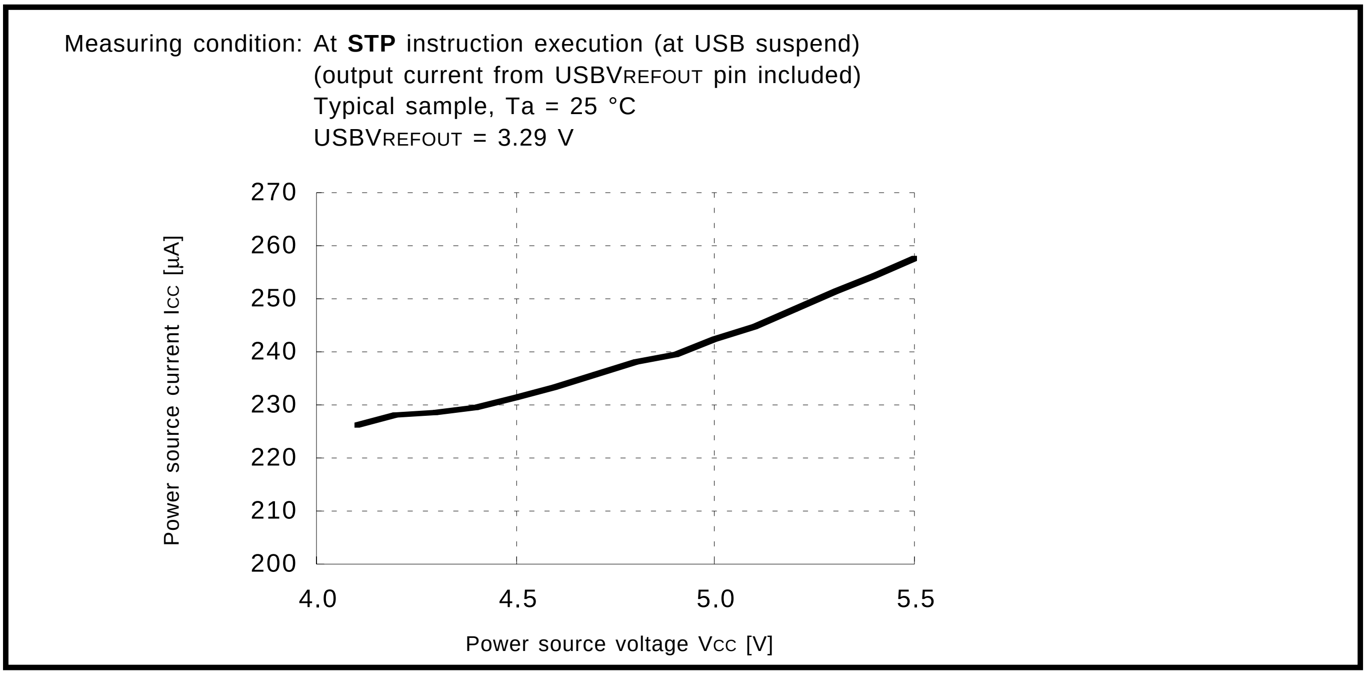


Fig. 3.2.5 ICC-VCC characteristic example (at USB suspend, Ta = 25 °C)

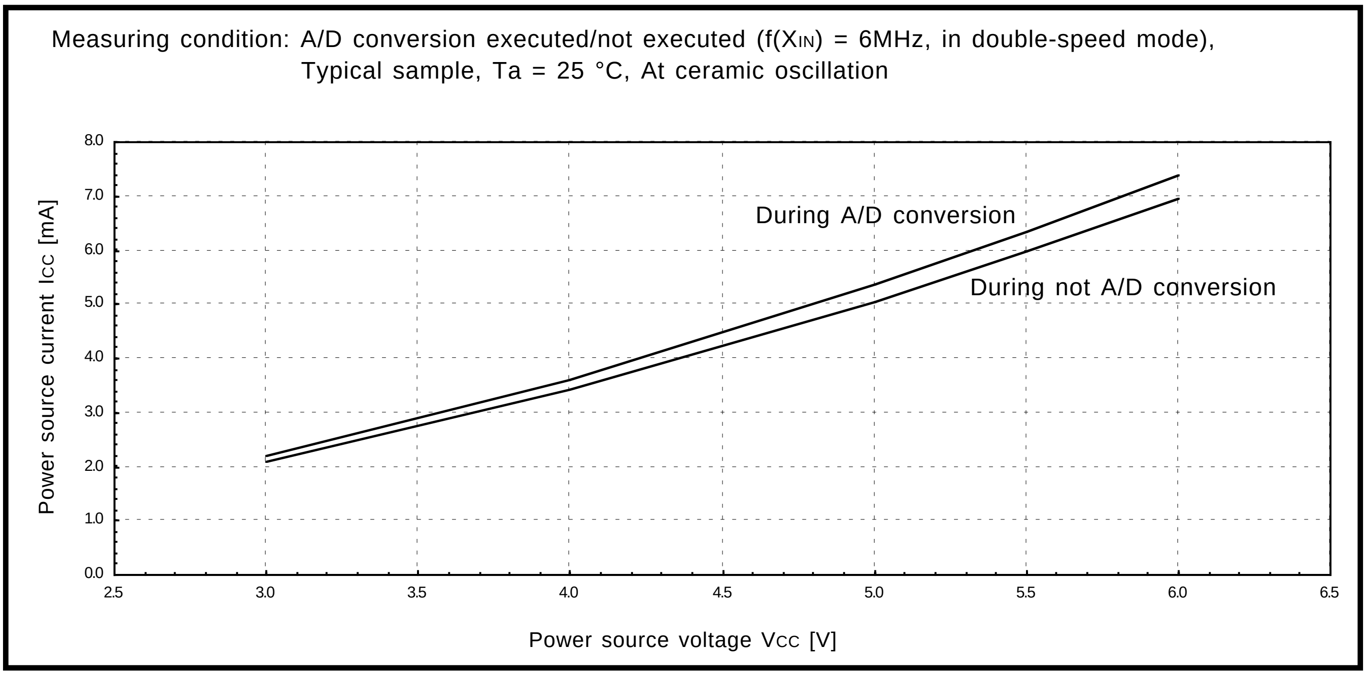
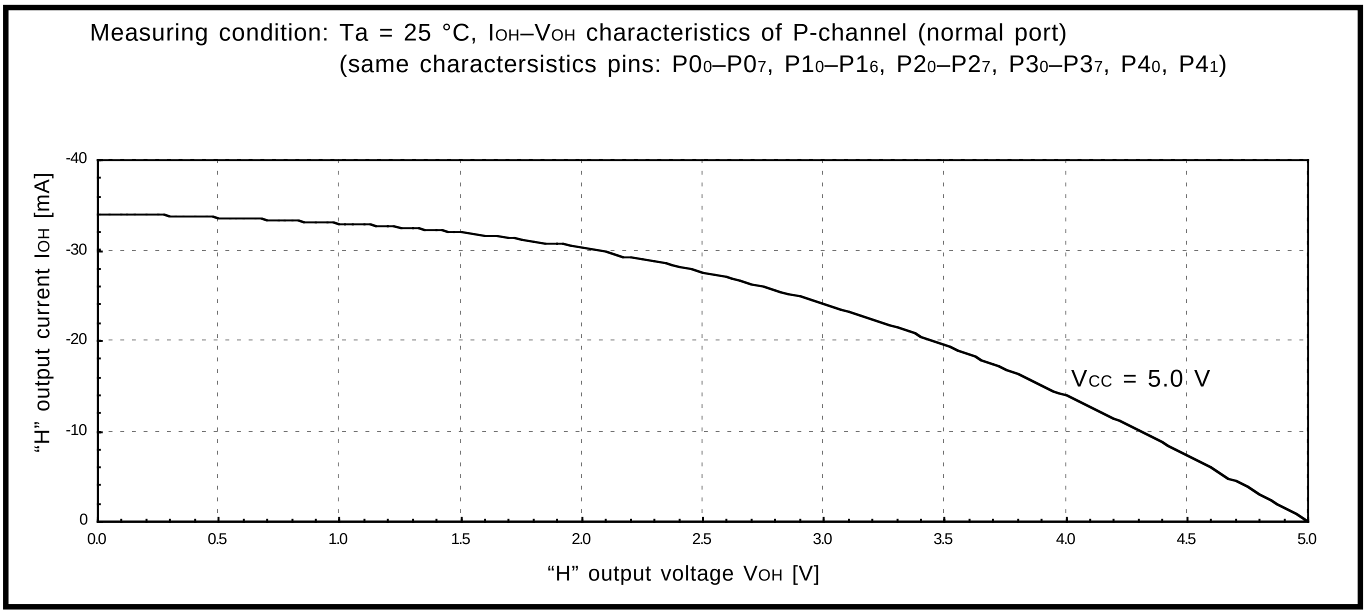


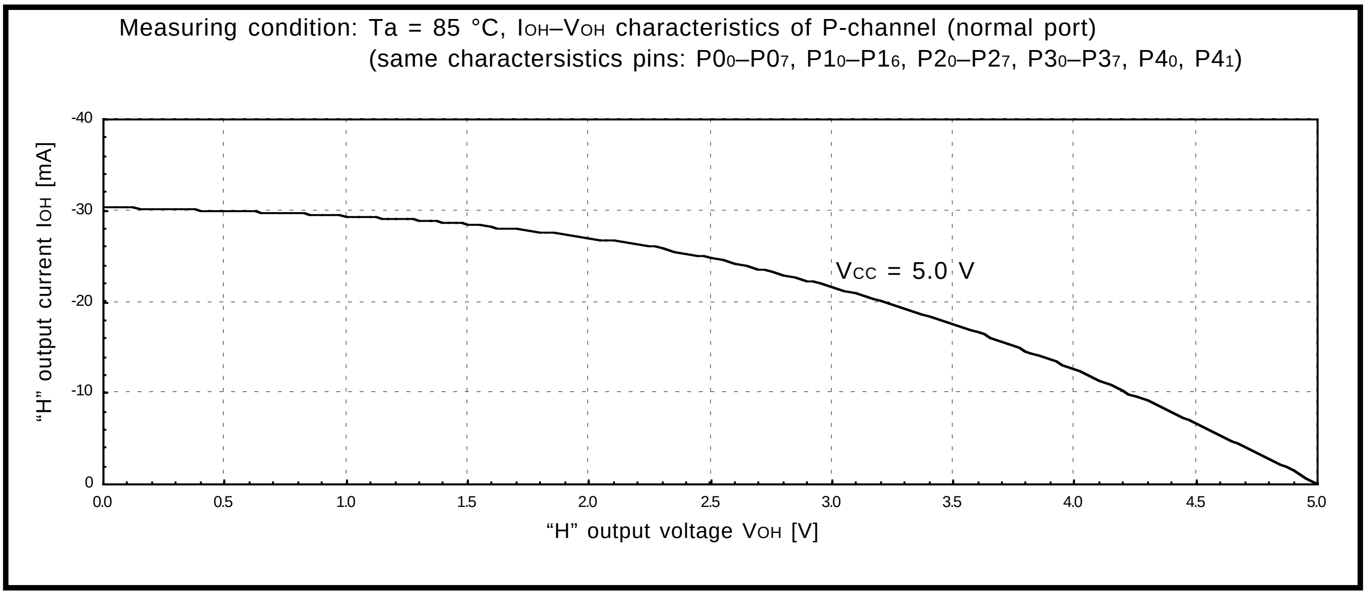
Fig. 3.2.6 ICC-VCC characteristic example (A/D conversion executed/not executed, f(XIN) = 6MHz, in double-speed mode)

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



Measuring condition: Ta = 25 °C, I_{OH}–V_{OH} characteristics of P-channel (normal port)
(same charactersistics pins: $P0_0$–$P0_7$, $P1_0$–$P1_6$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$, $P4_1$)

Fig. 3.2.7 V_{OH}-I_{OH} characteristic example of P-channel (Ta = 25 °C): normal port



Measuring condition: Ta = 85 °C, I_{OH}–V_{OH} characteristics of P-channel (normal port)
(same charactersistics pins: $P0_0$–$P0_7$, $P1_0$–$P1_6$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$, $P4_1$)

Fig. 3.2.8 V_{OH}-I_{OH} characteristic example of P-channel (Ta = 85 °C): normal port

Measuring condition: Ta = 25 °C, V_{OL}–I_{OL} characteristics of N-channel (normal port)
(same charactersisitics pins: P0_0–P0_7, P1_0–P1_6, P2_0–P2_7, P3_7, P4_0, P4_1)

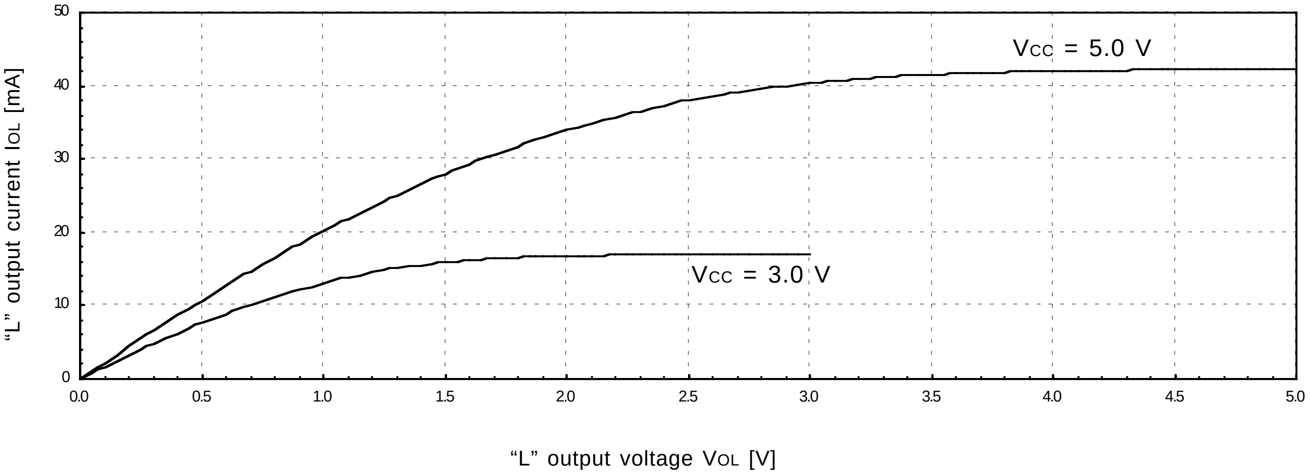


Fig. 3.2.9 V_{OL}-I_{OL} characteristic example of N-channel (Ta = 25 °C): Normal port

Measuring condition: Ta = 85 °C, V_{OL}–I_{OL} characteristics of N-channel (normal port)
(same charactersisitics pins: P0_0–P0_7, P1_0–P1_6, P2_0–P2_7, P3_7, P4_0, P4_1)

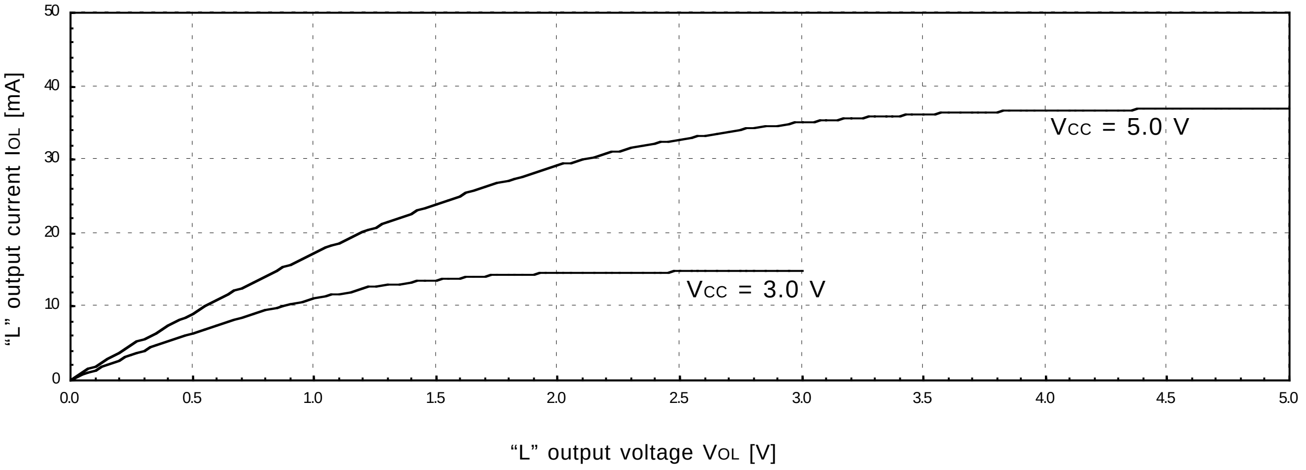


Fig. 3.2.10 V_{OL}-I_{OL} characteristic example of N-channel (Ta = 85 °C): Normal port

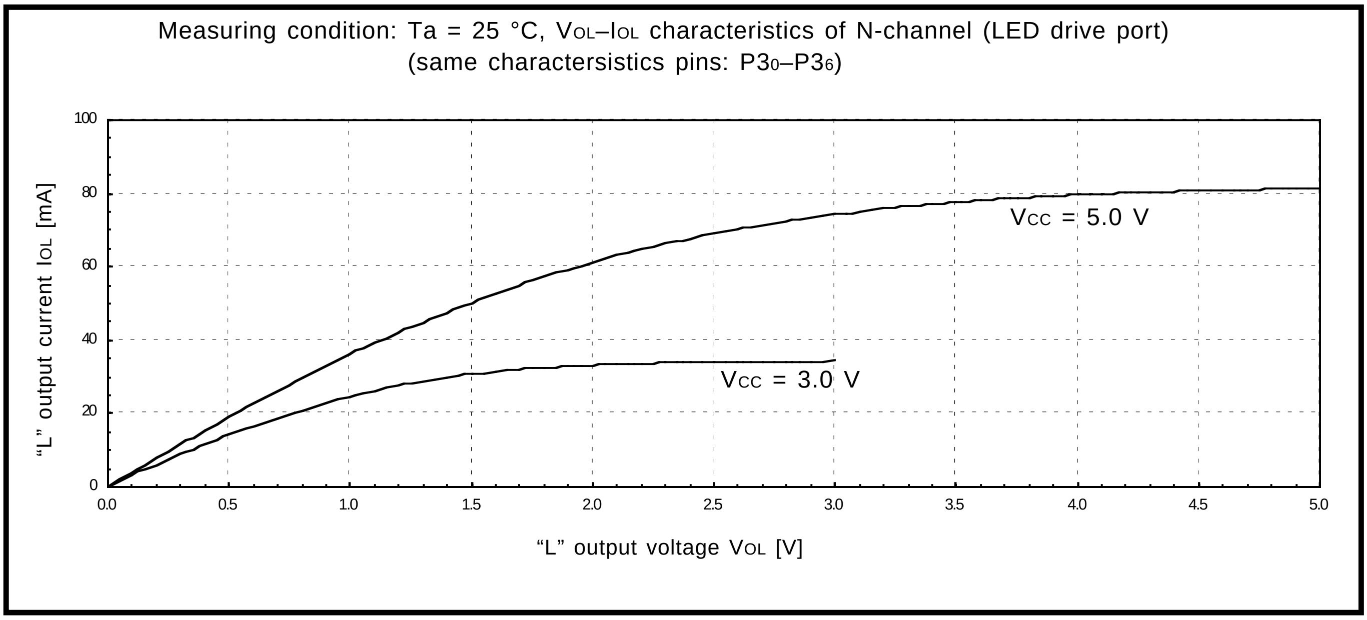


Fig. 3.2.11 VOL-IOL characteristic example of N-channel (Ta = 25 °C): LED drive port

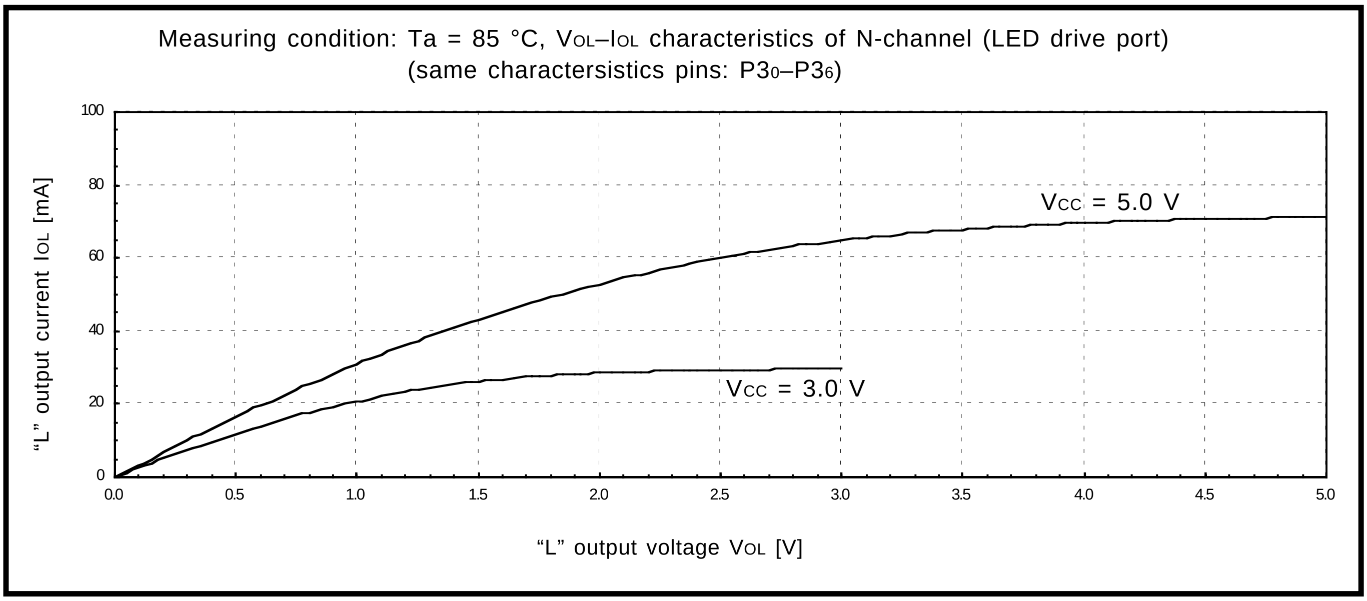


Fig. 3.2.12 VOL-IOL characteristic example N-channel (Ta = 85 °C): LED drive port

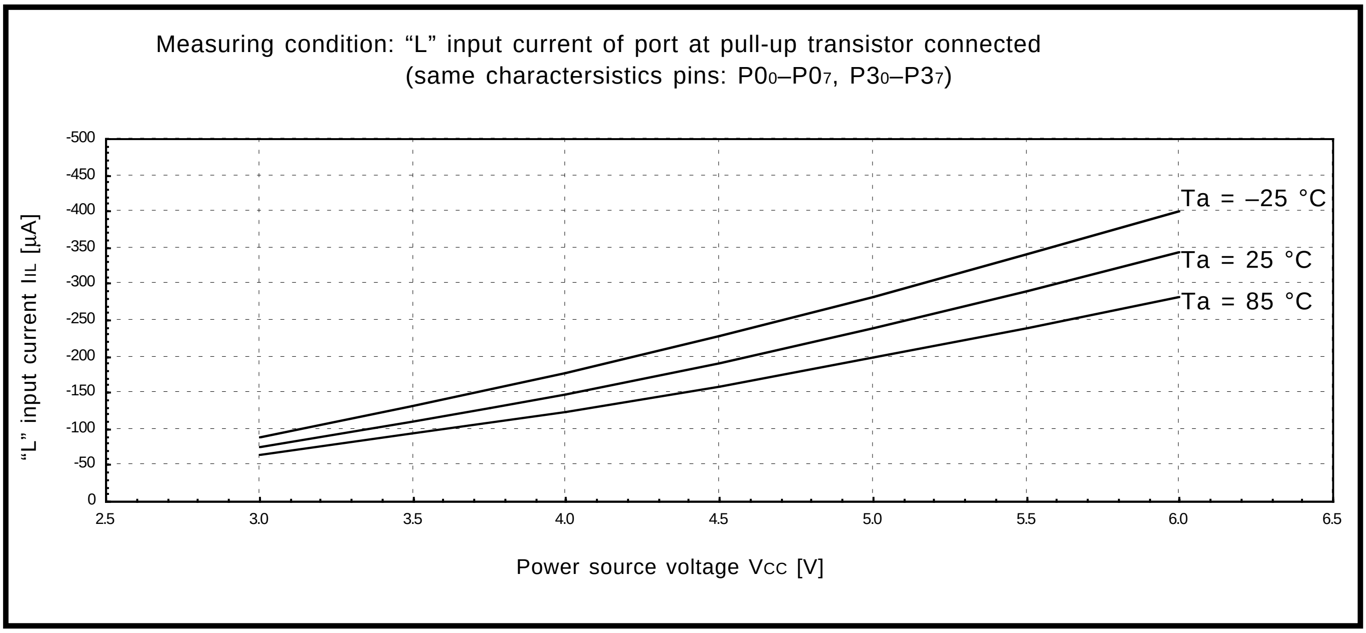


Fig. 3.2.13 “L” input current of port at pull-up transistor connected

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.

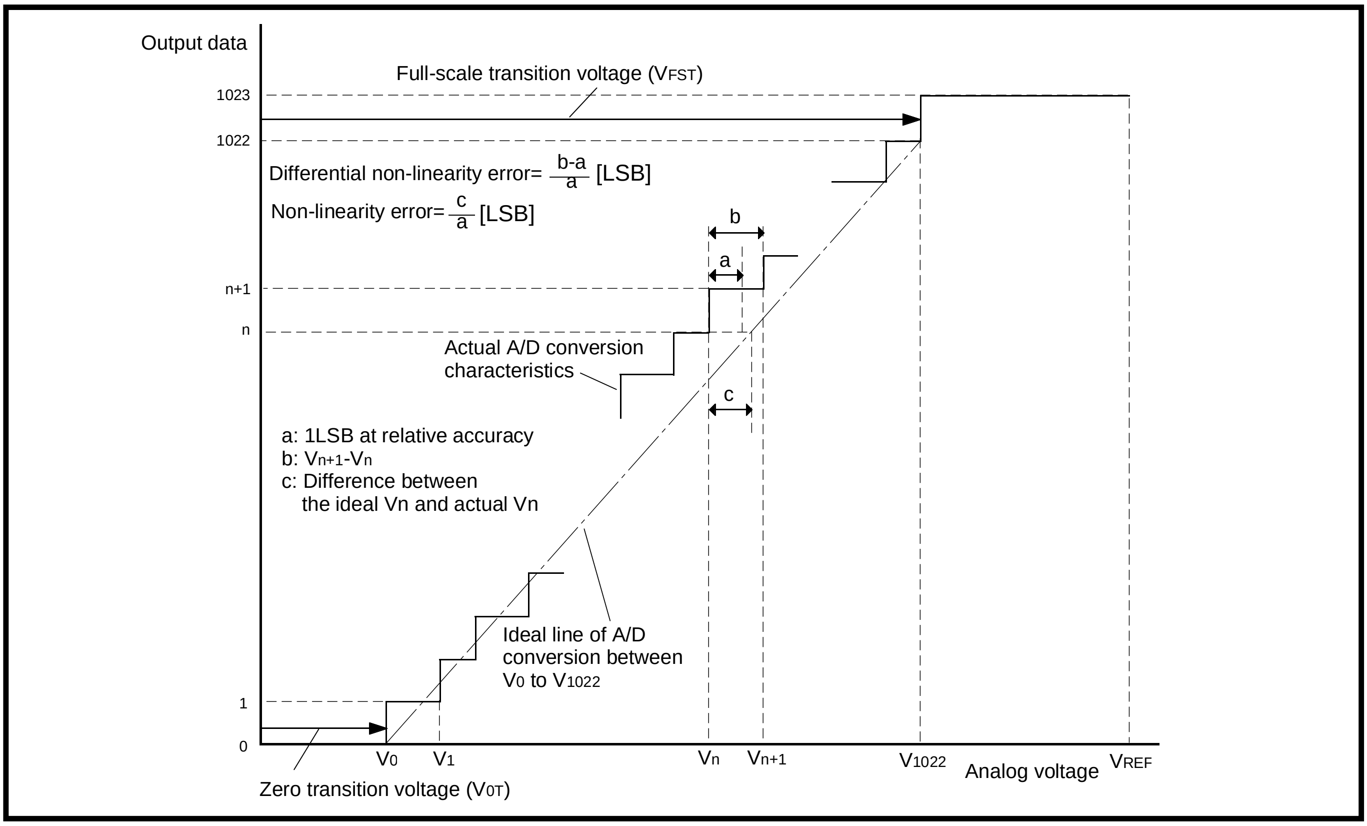

Output data
Full-scale transition voltage (VFST)
1023
1022
Differential non-linearity error= (b-a)/a [LSB]
Non-linearity error= c/a [LSB]
b
a
n+1
n
Actual A/D conversion characteristics
c
a: 1LSB at relative accuracy
b: Vn+1-Vn
c: Difference between the ideal Vn and actual Vn
Ideal line of A/D conversion between V0 to V1022
1
0
V0
V1
Vn
Vn+1
V1022
Analog voltage
VREF
Zero transition voltage (VOT)

Fig. 3.2.14 Definition of A/D conversion accuracy

Vn: Analog input voltage when the output data changes from "n" to "n + 1" (n = 0 to 1022)

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

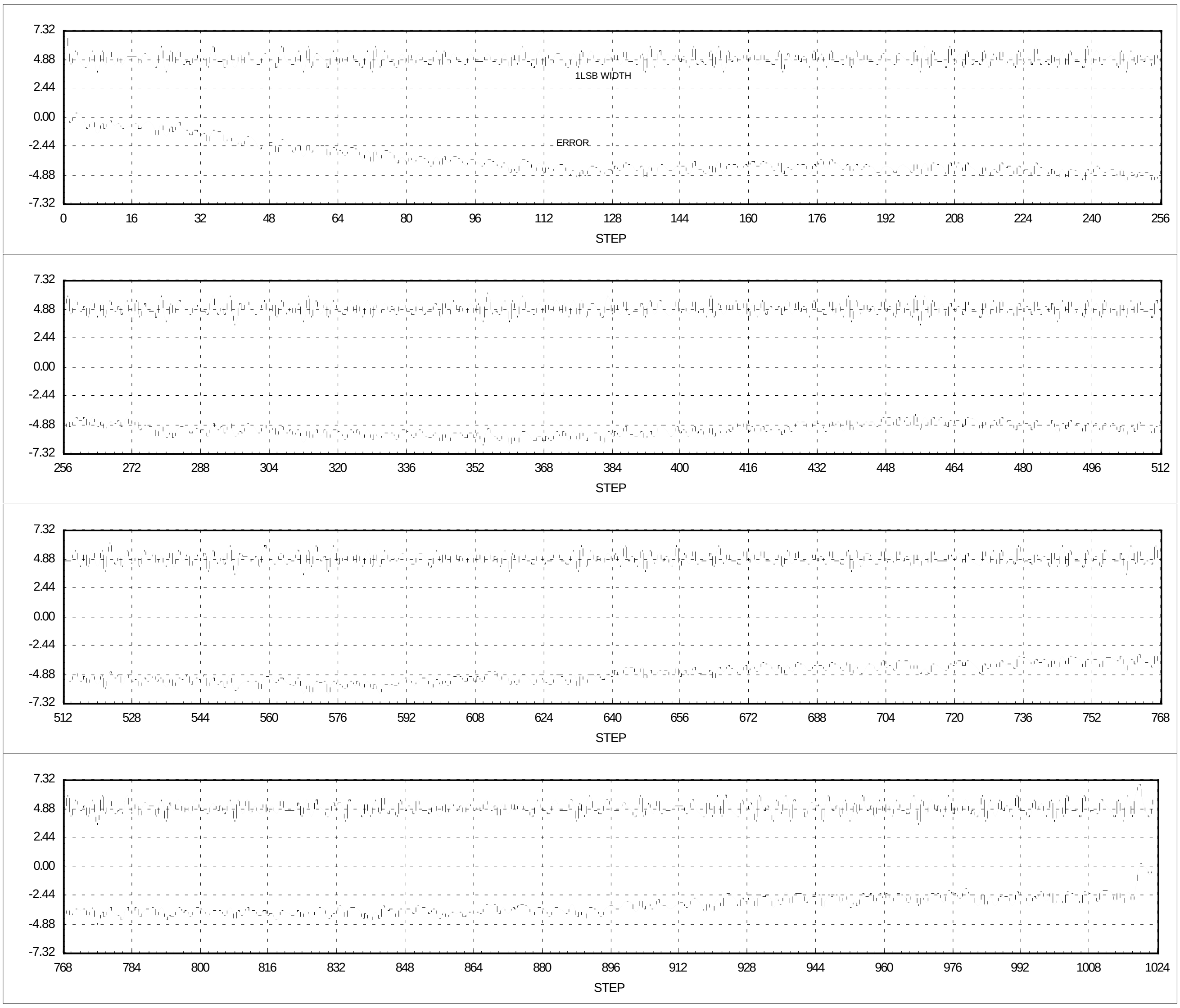


Fig. 3.2.15 A/D conversion typical characteristic example

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.

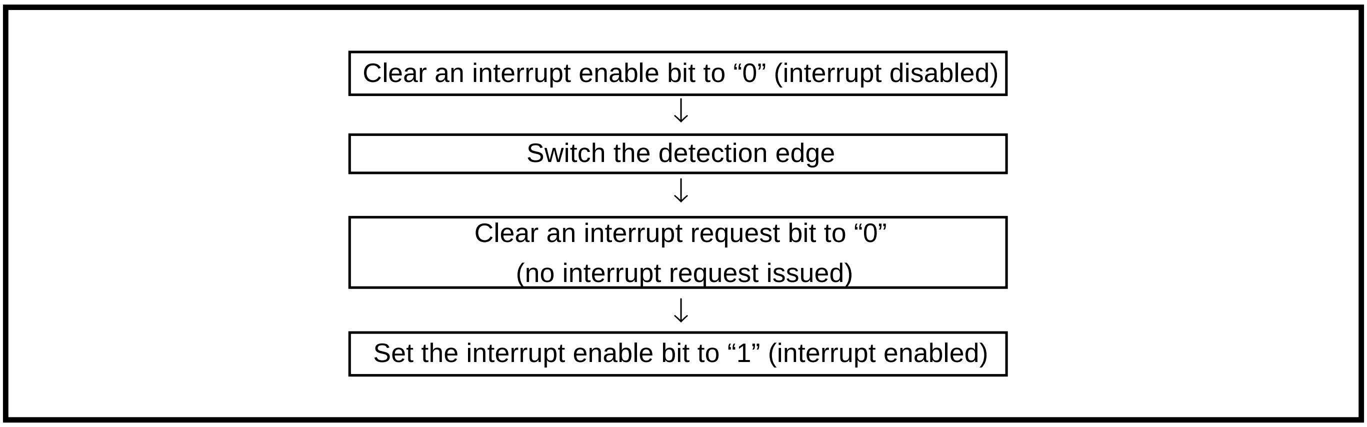

Clear an interrupt enable bit to "0" (interrupt disabled)
↓
Switch the detection edge
↓
Clear an interrupt request bit to "0" (no interrupt request issued)
↓
Set the interrupt enable bit to "1" (interrupt enabled)

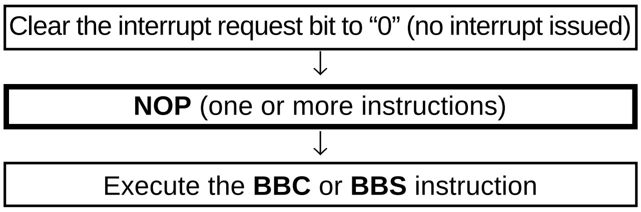
Fig. 3.3.1 Sequence of switch the detection edge

● **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.


Clear the interrupt request bit to "0" (no interrupt issued)
↓
NOP (one or more instructions)
↓
Execute the BBC or BBS instruction

Data transfer instruction:
LDM, LDA, STA, STX, and STY instructions

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.

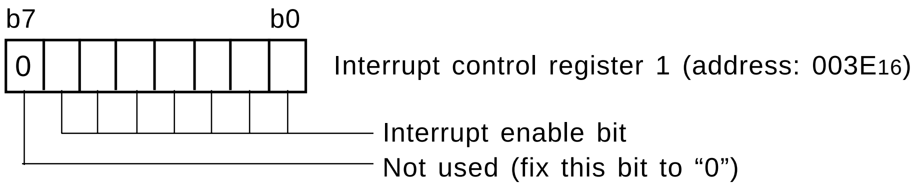
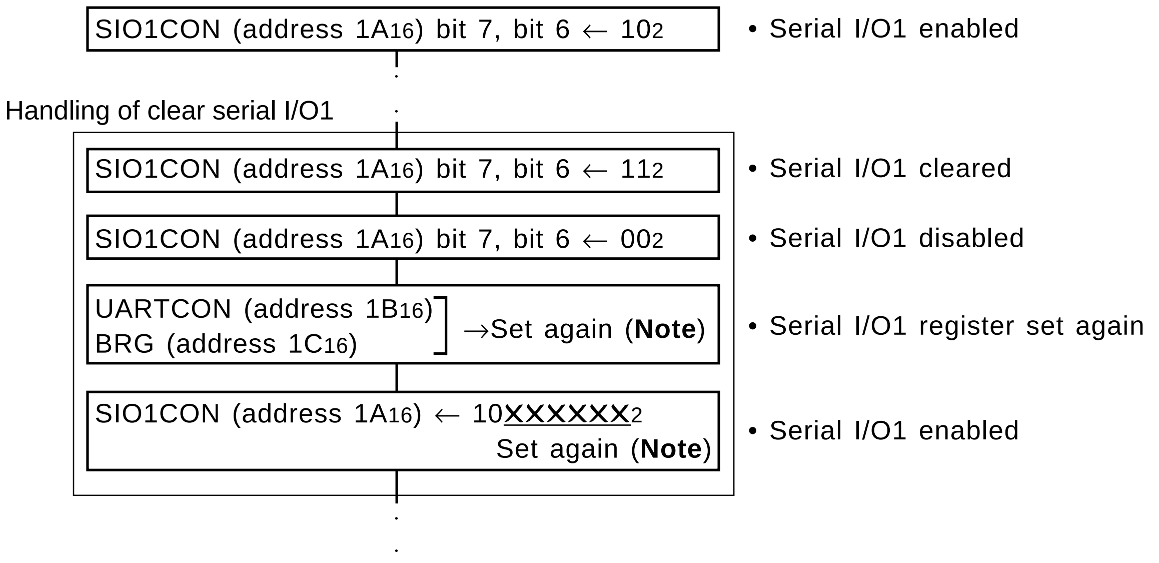

b7
b0
0
Interrupt control register 1 (address: 003E16)
Interrupt enable bit
Not used (fix this bit to “0”)

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.


SIO1CON (address 1A16) bit 7, bit 6 ← 102
• Serial I/O1 enabled
Handling of clear serial I/O1
SIO1CON (address 1A16) bit 7, bit 6 ← 112
• Serial I/O1 cleared
SIO1CON (address 1A16) bit 7, bit 6 ← 002
• Serial I/O1 disabled
UARTCON (address 1B16)
BRG (address 1C16)
→Set again (Note)
• Serial I/O1 register set again
SIO1CON (address 1A16) ← 10XXXXXX2
Set again (Note)
• Serial I/O1 enabled

Note: When the contents of register is not changed, setting again is not necessary.

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 0.01μF to 1μF. Further, be sure to verify the operation of application products on the user side.

● **Reason**

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

(2) Clock frequency during A/D conversion

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

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

(3) 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.

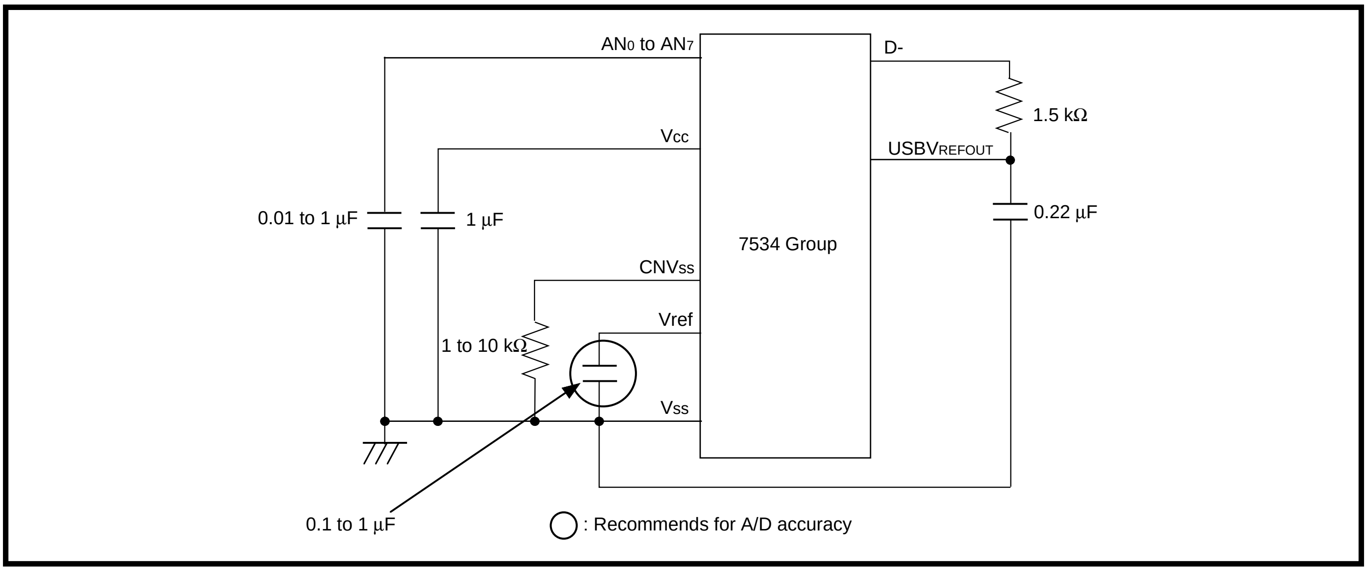


Fig. 3.3.5 Method to stabilize A/D conversion accuracy

(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".

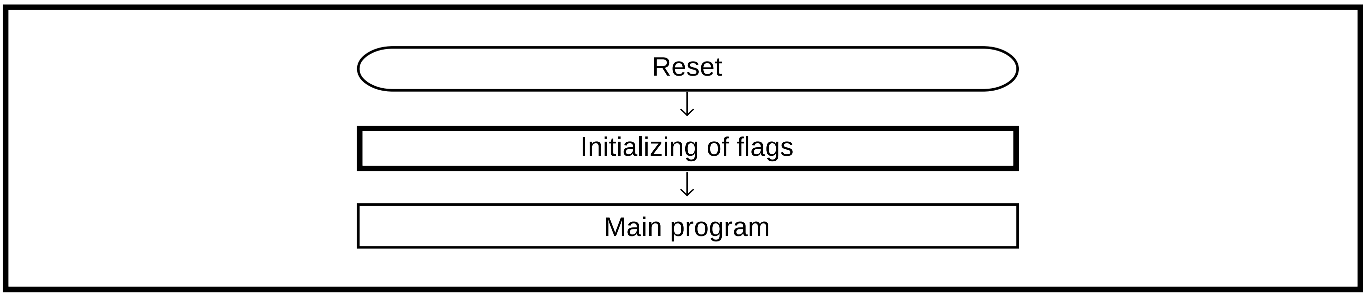


Fig. 3.3.6 Initialization of processor status register

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.

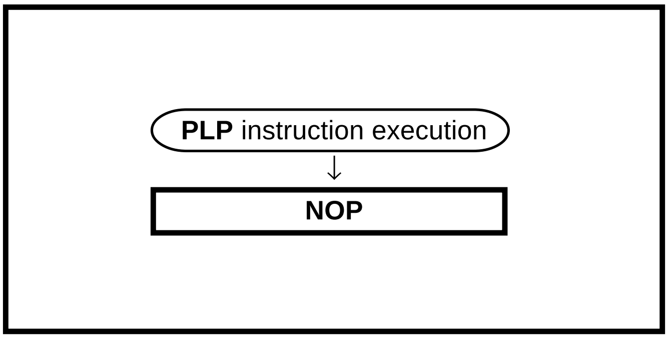


Fig. 3.3.7 Sequence of PLP instruction execution

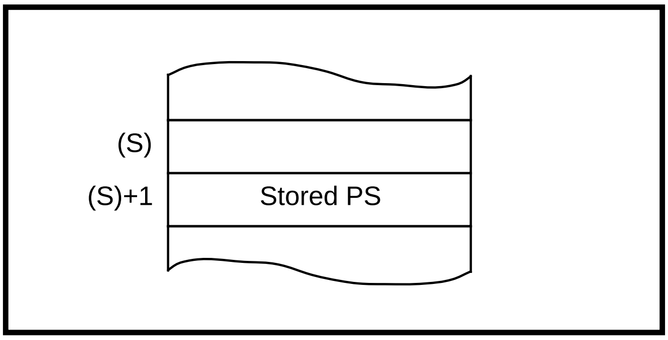


Fig. 3.3.8 Stack memory contents after PHP instruction execution

(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.

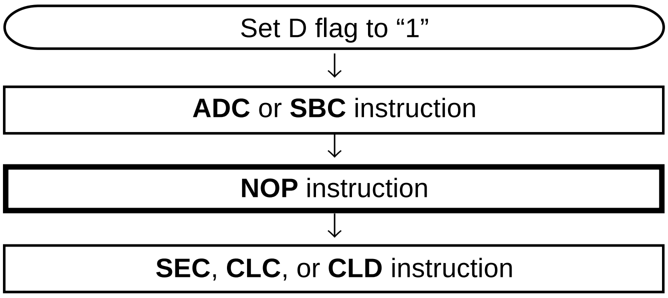


Fig. 3.3.9 Status flag at decimal calculations

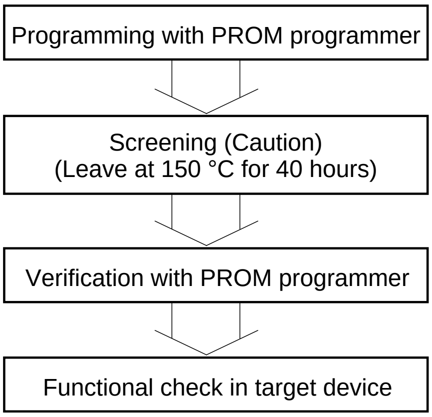
(3) JMP instruction

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 3.3.9 before actual use are recommended.


Programming with PROM programmer
Screening (Caution)
(Leave at 150 °C for 40 hours)
Verification with PROM programmer
Functional check in target device

Caution: The screening temperature is far higher than the storage temperature. Never expose to 150 °C exceeding 100 hours.

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 V_{PP} pin (is also used as the CNV_{SS} pin), or the product may be permanently damaged.

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

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: Adersses $E080_{16}$ to $FFFD_{16}$ in the built-in PROM corresponds to addresses $0E080_{16}$ to $0FFFD_{16}$ in the PROM programmer.

2: Adersses $C080_{16}$ to $FFFD_{16}$ in the built-in PROM corresponds to addresses $0C080_{16}$ to $0FFFD_{16}$ 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 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.

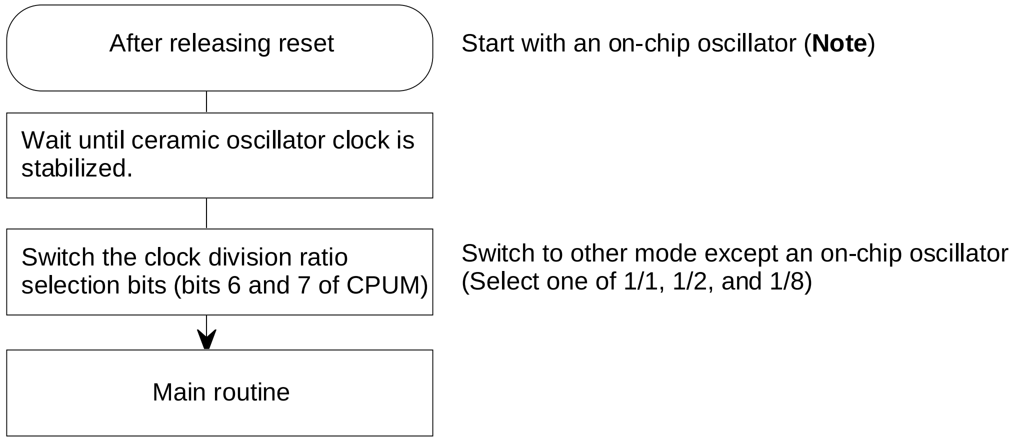
● **Reason**
If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between ports.

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

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.


After releasing reset
Start with an on-chip oscillator (**Note**)
Wait until ceramic oscillator clock is stabilized.
Switch the clock division ratio selection bits (bits 6 and 7 of CPUM)
Switch to other mode except an on-chip oscillator (Select one of 1/1, 1/2, and 1/8)
Main routine

Note. After releasing reset the operation starts by starting an on-chip oscillator automatically. Do not use an on-chip oscillator at ordinary operation.

Fig. 3.3.11 Switching method of CPU mode register

3.3.12 Notes on using 32-pin version

- Do not change the P3$_5$, P3$_6$ 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 INT$_1$ 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 $P1_0$, $P1_2$, $P1_3$ input level selection bit (bit 3 of address 17_{16}) 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 $P1_0$, $P1_2$, $P1_3$ 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.

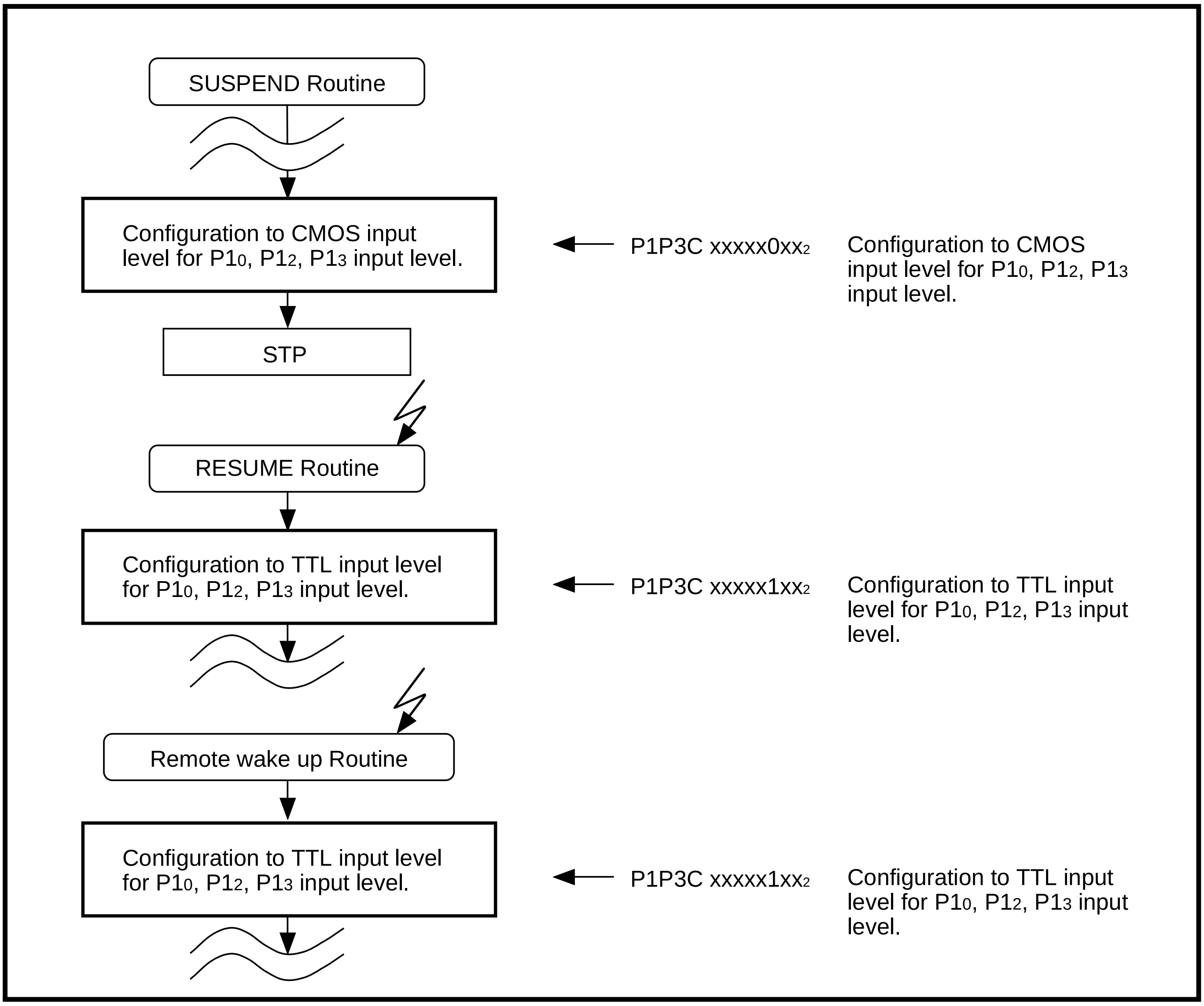

SUSPEND Routine
Configuration to CMOS input level for P1₀, P1₂, P1₃ input level.
P1P3C xxxxx0xx₂
Configuration to CMOS input level for P1₀, P1₂, P1₃ input level.
STP
RESUME Routine
Configuration to TTL input level for P1₀, P1₂, P1₃ input level.
P1P3C xxxxx1xx₂
Configuration to TTL input level for P1₀, P1₂, P1₃ input level.
Remote wake up Routine
Configuration to TTL input level for P1₀, P1₂, P1₃ input level.
P1P3C xxxxx1xx₂
Configuration to TTL input level for P1₀, P1₂, P1₃ input level.

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.

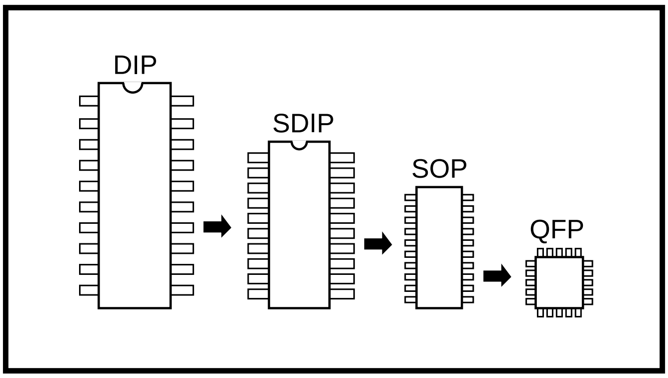


Fig. 3.4.1 Selection of packages

(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 Vss pin with the shortest possible wiring (within 20mm).

● **Reason**

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

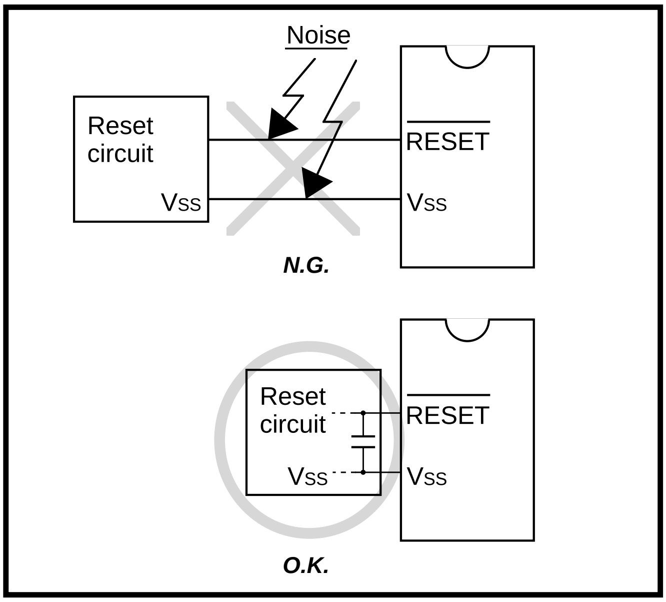


Fig. 3.4.2 Wiring for the $\overline{\text{RESET}}$ pin

(3) Wiring for clock input/output pins

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

● **Reason**

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

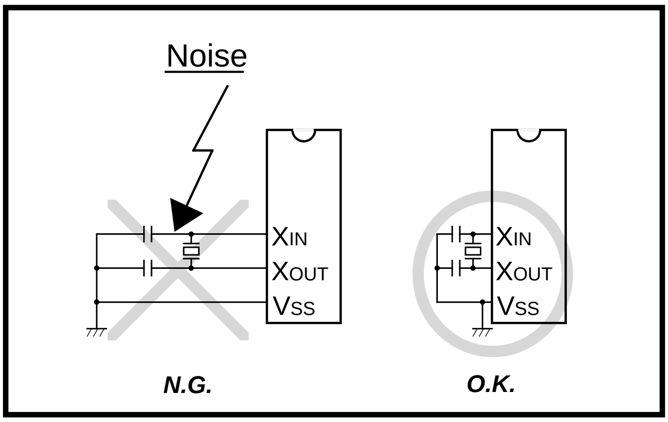


Fig. 3.4.3 Wiring for clock I/O pins

(4) Wiring to CNVss pin

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

● **Reason**

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

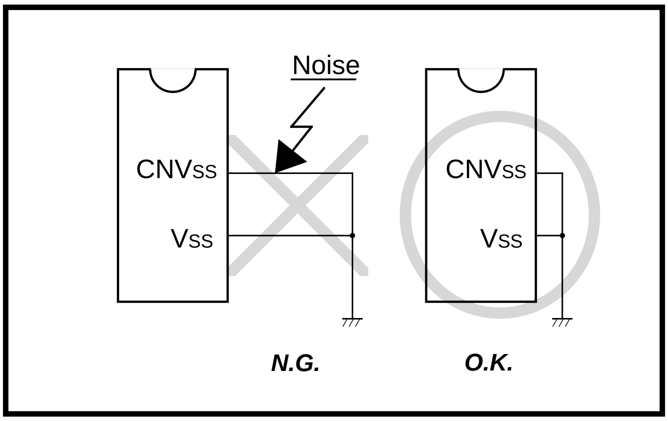


Fig. 3.4.4 Wiring for CNVss pin

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

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

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

● **Reason**

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

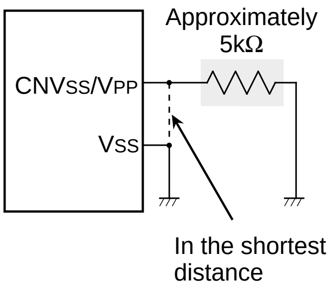


Fig. 3.4.5 Wiring for the VPP pin of the One Time PROM

3.4.2 Connection of bypass capacitor across Vss line and Vcc line

Connect an approximately 1.0 μF bypass capacitor across the Vss line and the Vcc line as follows:

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

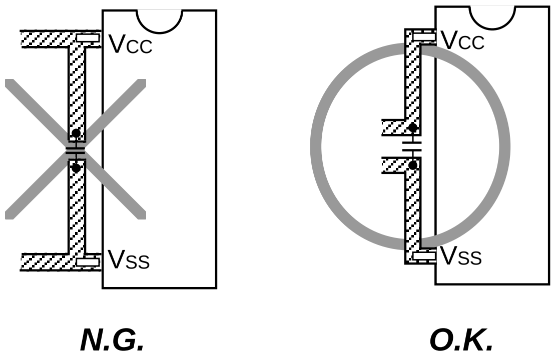


Fig. 3.4.6 Bypass capacitor across the Vss line and the Vcc line

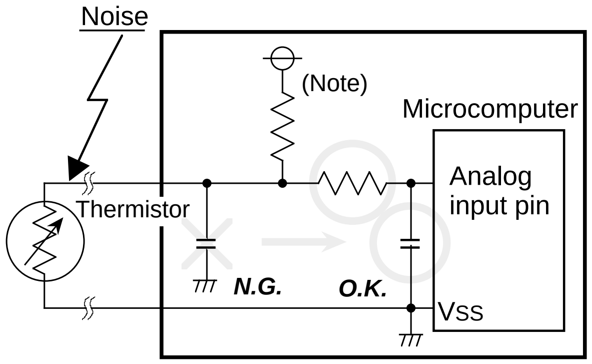
3.4.3 Wiring to analog input pins

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

● **Reason**

Signals which is input in an analog input pin (such as an A/D converter/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.



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

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.

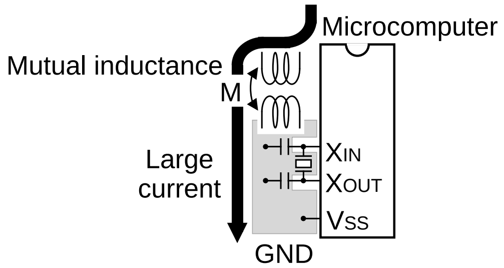


Fig. 3.4.8 Wiring for a large current signal line

(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.

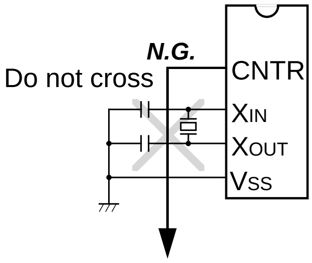


Fig. 3.4.9 Wiring of signal lines where potential levels change frequently

(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.

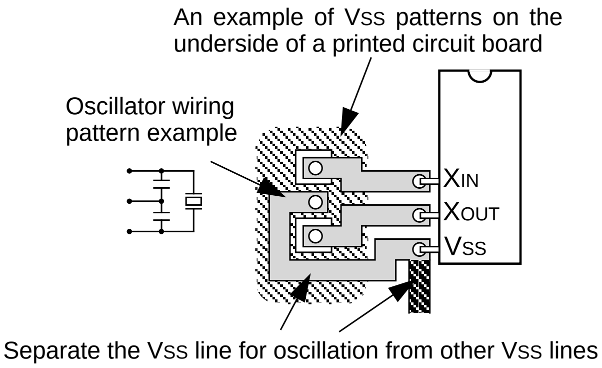

An example of Vss patterns on the underside of a printed circuit board
Oscillator wiring pattern example
XIN
XOUT
VSS
Separate the Vss line for oscillation from other Vss lines

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.

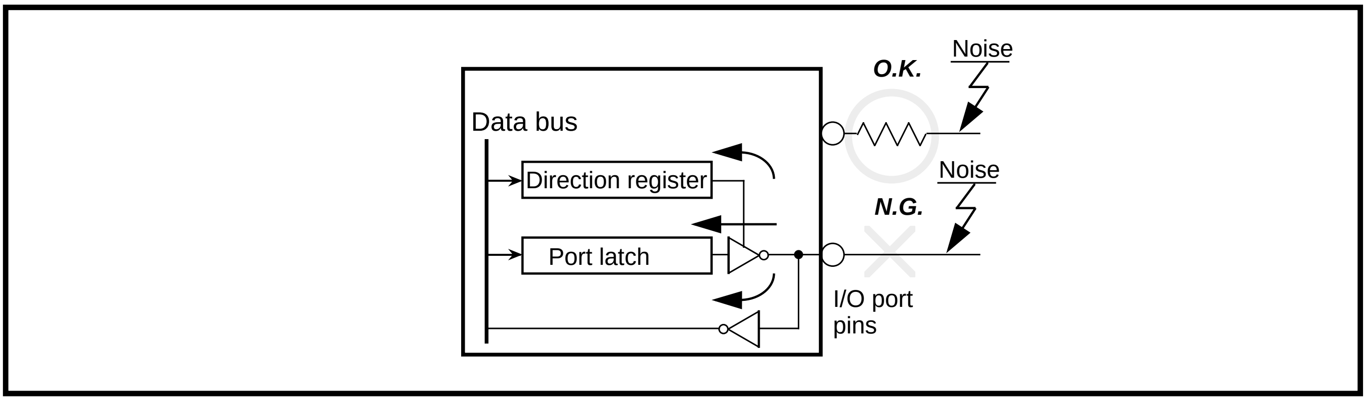


Fig. 3.4.11 Setup for I/O ports

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 ≥ (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.

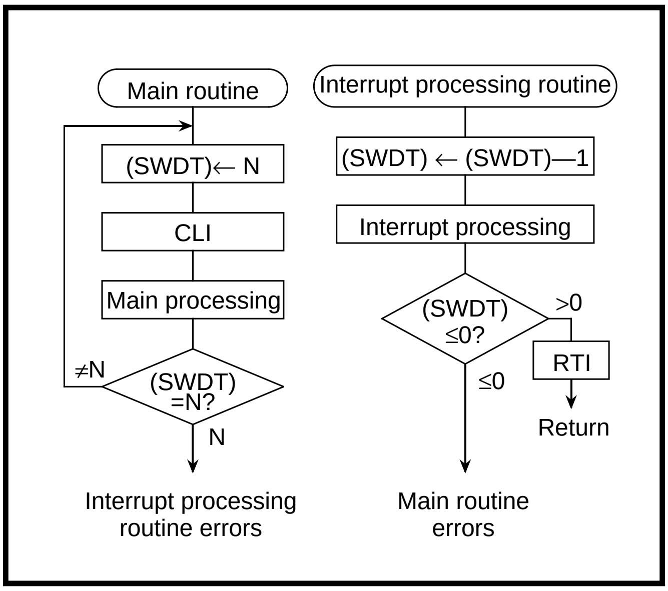
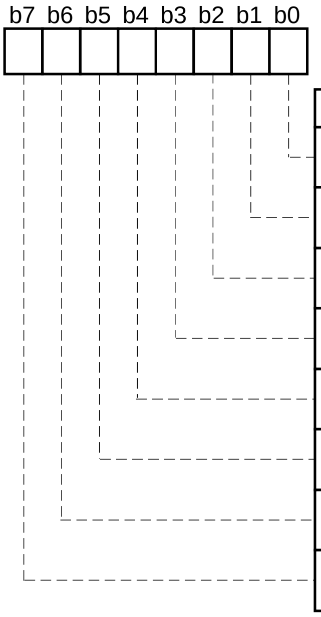


Fig. 3.4.12 Watchdog timer by software

3.5 List of registers

Port Pi



Port Pi (Pi) (i = 0 to 4) [Address : 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{16}]

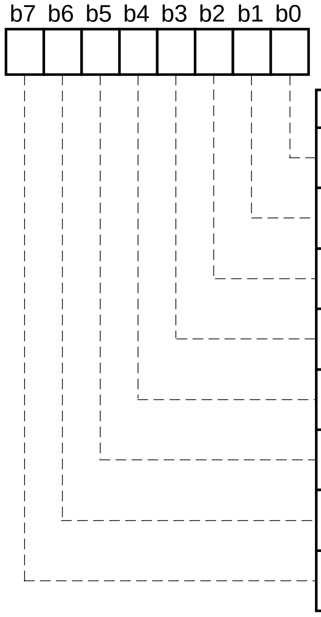
B	Name	Function	At reset	R	W
0	Port Pi$_0$	● In output mode Write } Port latch Read ● In input mode Write : Port latch Read : Value of pins	?	○	○
1	Port Pi$_1$		?	○	○
2	Port Pi$_2$		?	○	○
3	Port Pi$_3$		?	○	○
4	Port Pi$_4$		?	○	○
5	Port Pi$_5$		?	○	○
6	Port Pi$_6$		?	○	○
7	Port Pi$_7$		?	○	○

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

- 42-pin version: Ports P1$_7$, P4$_2$–P4$_7$
- 36-pin version: Ports P1$_5$–P1$_7$, P3$_6$, P4$_0$–P4$_7$
- 32-pin version: Ports P1$_5$–P1$_7$, P2$_6$, P2$_7$, P3$_5$–P3$_7$, P4$_0$–P4$_7$

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

Port Pi direction register



Port Pi direction register (PiD) (i = 0 to 4) [Address : 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{16}]

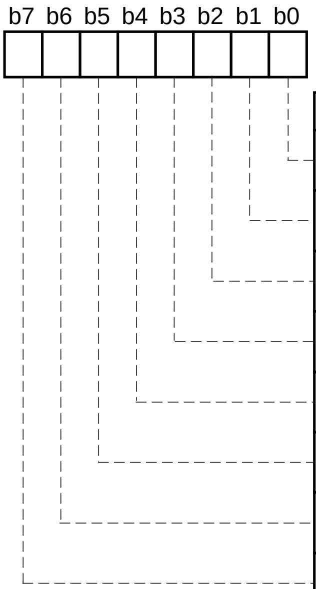
B	Name	Function	At reset	R	W
0	Port Pi direction register	0 : Port Pi$_0$ input mode 1 : Port Pi$_0$ output mode	0	×	○
1		0 : Port Pi$_1$ input mode 1 : Port Pi$_1$ output mode	0	×	○
2		0 : Port Pi$_2$ input mode 1 : Port Pi$_2$ output mode	0	×	○
3		0 : Port Pi$_3$ input mode 1 : Port Pi$_3$ output mode	0	×	○
4		0 : Port Pi$_4$ input mode 1 : Port Pi$_4$ output mode	0	×	○
5		0 : Port Pi$_5$ input mode 1 : Port Pi$_5$ output mode	0	×	○
6		0 : Port Pi$_6$ input mode 1 : Port Pi$_6$ output mode	0	×	○
7		0 : Port Pi$_7$ input mode 1 : Port Pi$_7$ output mode	0	×	○

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

- 42-pin version: Ports P1$_7$, P4$_2$–P4$_7$
- 36-pin version: Ports P1$_5$–P1$_7$, P3$_6$, P4$_0$–P4$_7$
- 32-pin version: Ports P1$_5$–P1$_7$, P2$_6$, P2$_7$, P3$_5$–P3$_7$, P4$_0$–P4$_7$

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

Pull-up control register



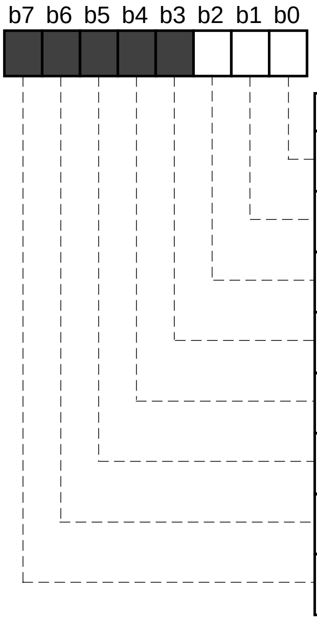
Pull-up control register (PULL) [Address : 16_{16}]

B	Name	Function	At reset	R	W
0	$P0_0$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
1	$P0_1$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
2	$P0_2$, $P0_3$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
3	$P0_4$ – $P0_7$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
4	$P3_0$ – $P3_3$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
5	$P3_4$ pull-up control bit	0 : Pull-up Off 1 : Pull-up On	1	○	○
6	$P3_5$, $P3_6$ pull-up control bit (**Note 2**)	0 : Pull-up Off 1 : Pull-up On	1	○	○
7	$P3_7$ pull-up control bit (**Note 3**)	0 : Pull-up Off 1 : Pull-up On	1	○	○

Notes 1: Pins set to output are disconnected from the pull-up control.
2: • 36-pin version: $P3_6$ is not existed.
• 32-pin version: Not used.
3: 32-pin version: Not used.

Fig. 3.5.3 Structure of Pull-up control register

Port P1P3 control register



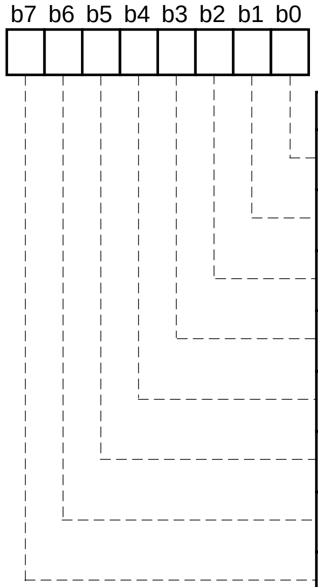
Port P1P3 control register (P1P3C) [Address : 17_{16}]

B	Name	Function	At reset	R	W
0	$P3_7$/INT_0 input level selection bit (**Note 1**)	0 : CMOS level 1 : TTL level	0	○	○
1	$P3_6$/INT_1 input level selection bit (**Note 2**)	0 : CMOS level 1 : TTL level	0	○	○
2	$P1_0$, $P1_2$,$P1_3$ input level selection bit	0 : CMOS level 1 : TTL level	0	○	○
3	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are “0”.		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Notes 1: For the 32-pin version, nothing is allocated for this bit.
This is a write disabled bit.
When this bit is read out, the value is “0”.
2: For the 32-pin and 36-pin versions, nothing is allocated for this bit.
This is a write disabled bit.
When this bit is read out, the value is “0”.

Fig. 3.5.4 Structure of Port P1P3 control register

Transmit/Receive buffer register



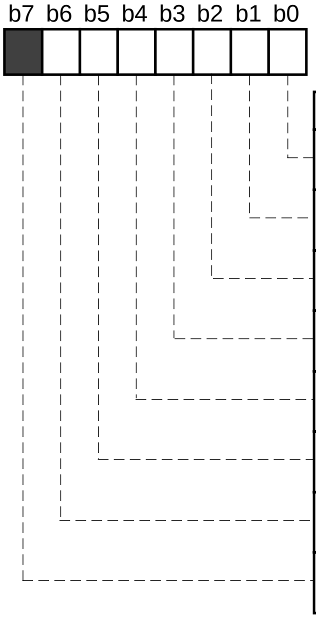
Transmit/Receive buffer register (TB/RB) [Address : 18_{16}]

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

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

Fig. 3.5.5 Structure of Transmit/Receive buffer register

UART status register

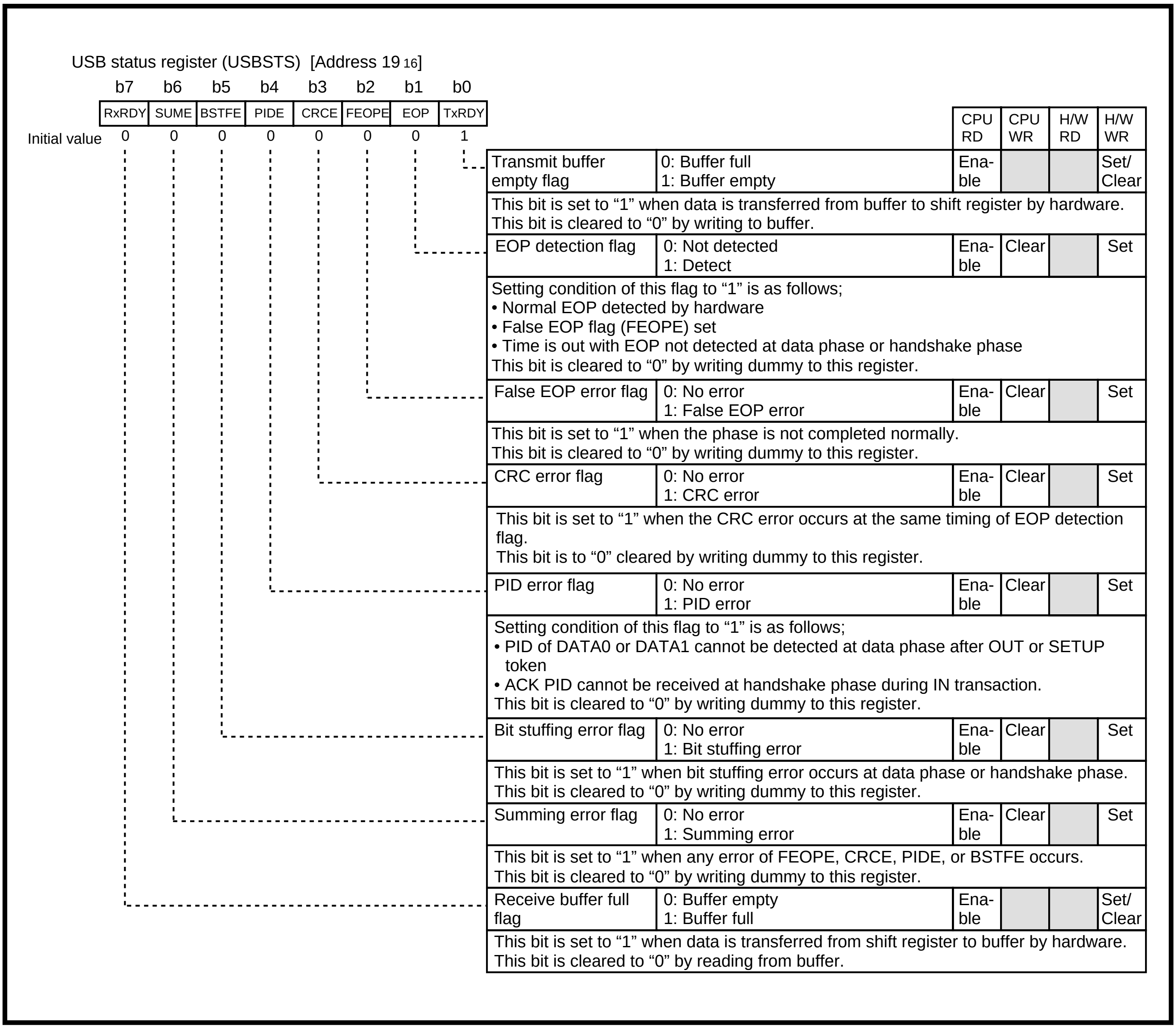


UART status register (UARTSTS) [Address : 19_{16}]

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

Fig. 3.5.6 Structure of UART status register

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



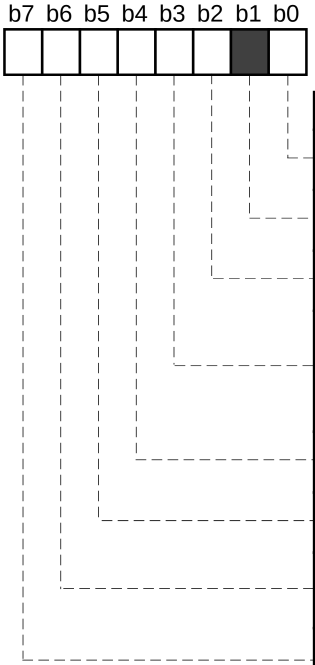
USB status register (USBSTS) [Address 19 16]

	b7	b6	b5	b4	b3	b2	b1	b0
	RxRDY	SUME	BSTFE	PIDE	CRCE	FEOPE	EOP	TxRDY
Initial value	0	0	0	0	0	0	0	1

		CPU RD	CPU WR	H/W RD	H/W WR
Transmit buffer empty flag	0: Buffer full 1: Buffer empty	Ena-ble	▨	▨	Set/ Clear
This bit is set to “1” when data is transferred from buffer to shift register by hardware. This bit is cleared to “0” by writing to buffer.					
EOP detection flag	0: Not detected 1: Detect	Ena-ble	Clear	▨	Set
Setting condition of this flag to “1” is as follows; • Normal EOP detected by hardware • False EOP flag (FEOPE) set • Time is out with EOP not detected at data phase or handshake phase This bit is cleared to “0” by writing dummy to this register.					
False EOP error flag	0: No error 1: False EOP error	Ena-ble	Clear	▨	Set
This bit is set to “1” when the phase is not completed normally. This bit is cleared to “0” by writing dummy to this register.					
CRC error flag	0: No error 1: CRC error	Ena-ble	Clear	▨	Set
This bit is set to “1” when the CRC error occurs at the same timing of EOP detection flag. This bit is to “0” cleared by writing dummy to this register.					
PID error flag	0: No error 1: PID error	Ena-ble	Clear	▨	Set
Setting condition of this flag to “1” is as follows; • PID of DATA0 or DATA1 cannot be detected at data phase after OUT or SETUP token • ACK PID cannot be received at handshake phase during IN transaction. This bit is cleared to “0” by writing dummy to this register.					
Bit stuffing error flag	0: No error 1: Bit stuffing error	Ena-ble	Clear	▨	Set
This bit is set to “1” when bit stuffing error occurs at data phase or handshake phase. This bit is cleared to “0” by writing dummy to this register.					
Summing error flag	0: No error 1: Summing error	Ena-ble	Clear	▨	Set
This bit is set to “1” when any error of FEOPE, CRCE, PIDE, or BSTFE occurs. This bit is cleared to “0” by writing dummy to this register.					
Receive buffer full flag	0: Buffer empty 1: Buffer full	Ena-ble	▨	▨	Set/ Clear
This bit is set to “1” when data is transferred from shift register to buffer by hardware. This bit is cleared to “0” by reading from buffer.					

Fig. 3.5.7 Structure of USB status register

Serial I/O1 control register

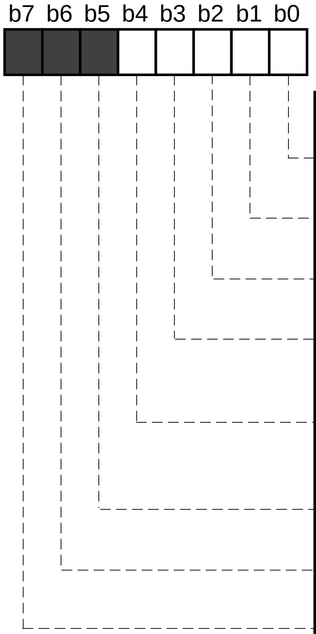


Serial I/O1 control register (SIO1CON) [Address : 1A 16]

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

Fig. 3.5.8 Structure of Serial I/O1 control register

UART control register



UART control register (UARTCON) [Address : 1B 16]

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

Fig. 3.5.9 Structure of UART control register

Baud rate generator

b7	b6	b5	b4	b3	b2	b1	b0

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

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

Fig. 3.5.10 Structure of Baud rate generator

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

USB data toggle synchronization register (TRSYNC) [Address 1D $_{16}$]

	b7	b6	b5	b4	b3	b2	b1	b0
	SQTGL							
Initial value	0							

		CPU RD	CPU WR	H/W RD	H/W WR
Sequence bit toggle flag	0: No toggle 1: Sequence toggle	Ena-ble	Clear		Set

Setting condition of this flag to "1" is as follows;
- Setting of handshake for OUT token in EP0PID is ACK, toggle of data PID is performed normally, and errors do not occur at data phase during OUT and SETUP transaction.
- When ACK is received during IN transaction.

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

Fig. 3.5.11 Structure of USB data toggle synchronization register

USB interrupt source discrimination register 1 (USBIR1) [Address 1E $_{16}$]

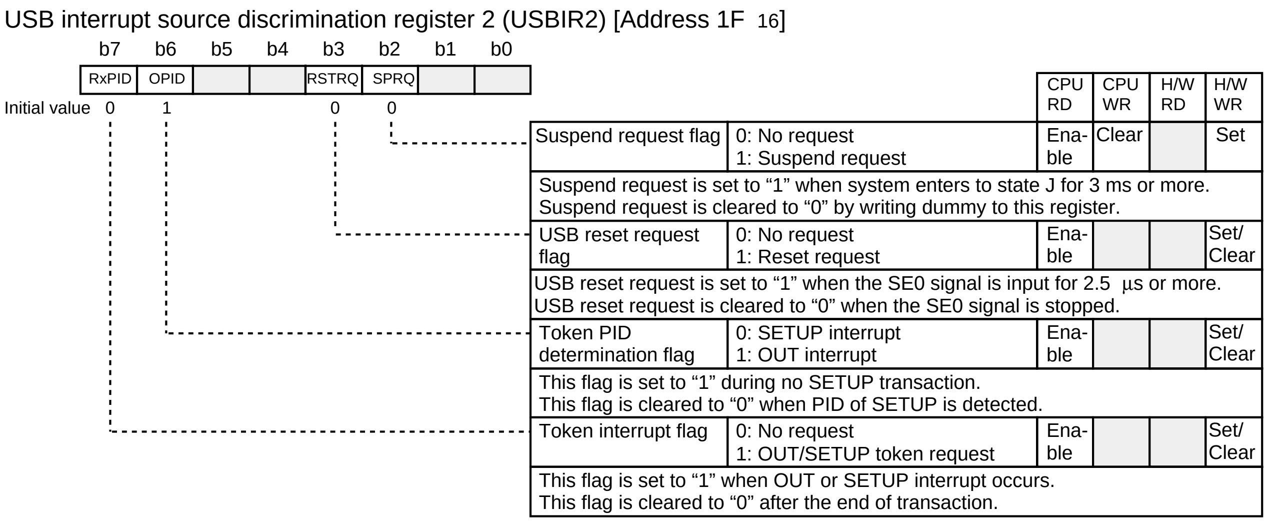
	b7	b6	b5	b4	b3	b2	b1	b0
	RxEP							
Initial value	0							

		CPU RD	CPU WR	H/W RD	H/W WR
Endpoint determination flag	0: Endpoint 0 interrupt 1: Endpoint 1 interrupt	Ena-ble			Set/ Clear

This flag is set to "1" when IN token interrupt of endpoint 1 occurs.
This flag is cleared to "0" when IN token interrupt of endpoint 0 occurs.
Writing to this bit is invalid. Do not write "1" to bits 0 to 6.

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.

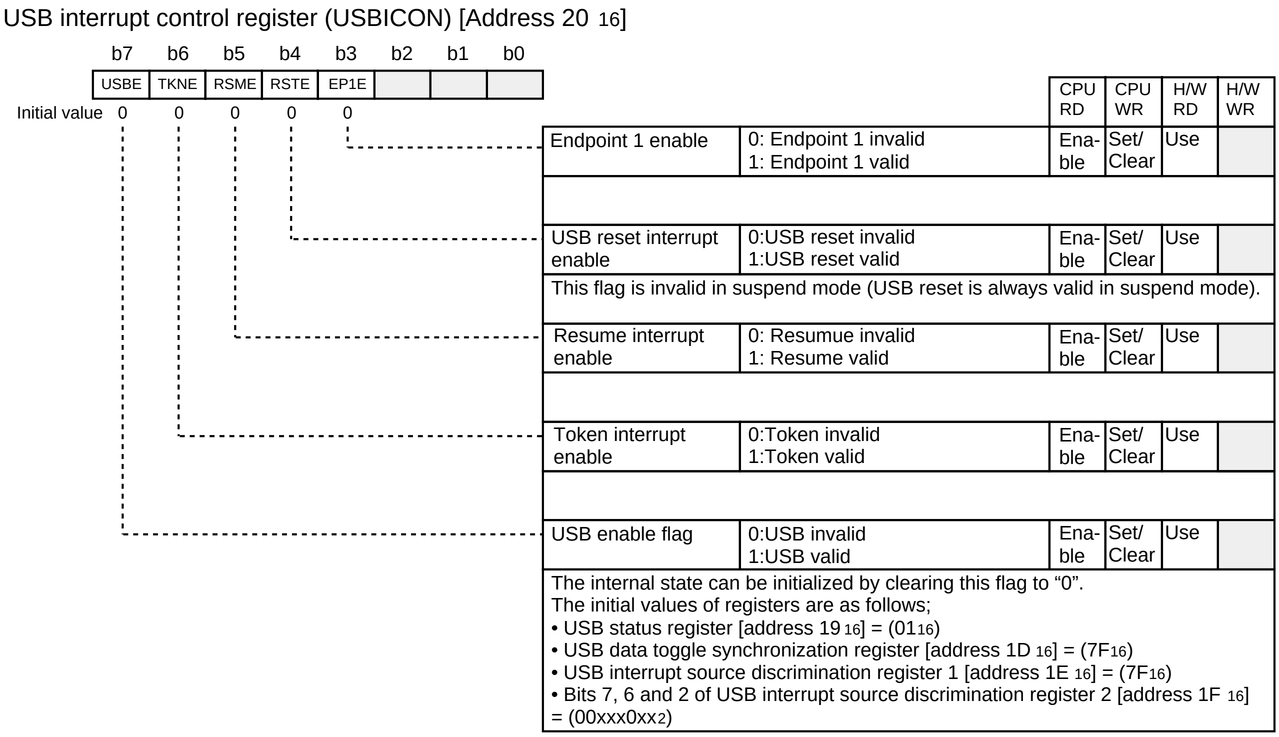


USB interrupt source discrimination register 2 (USBIR2) [Address 1F$_{16}$]

	b7	b6	b5	b4	b3	b2	b1	b0
	RxPID	OPID			RSTRQ	SPRQ		
Initial value	0	1			0	0		

		CPU RD	CPU WR	H/W RD	H/W WR
Suspend request flag	0: No request 1: Suspend request	Ena-ble	Clear		Set
Suspend request is set to “1” when system enters to state J for 3 ms or more. Suspend request is cleared to “0” by writing dummy to this register.					
USB reset request flag	0: No request 1: Reset request	Ena-ble			Set/ Clear
USB reset request is set to “1” when the SE0 signal is input for 2.5 µs or more. USB reset request is cleared to “0” when the SE0 signal is stopped.					
Token PID determination flag	0: SETUP interrupt 1: OUT interrupt	Ena-ble			Set/ Clear
This flag is set to “1” during no SETUP transaction. This flag is cleared to “0” when PID of SETUP is detected.					
Token interrupt flag	0: No request 1: OUT/SETUP token request	Ena-ble			Set/ Clear
This flag is set to “1” when OUT or SETUP interrupt occurs. This flag is cleared to “0” after the end of transaction.					

Fig. 3.5.13 Structure of USB interrupt source discrimination register 2



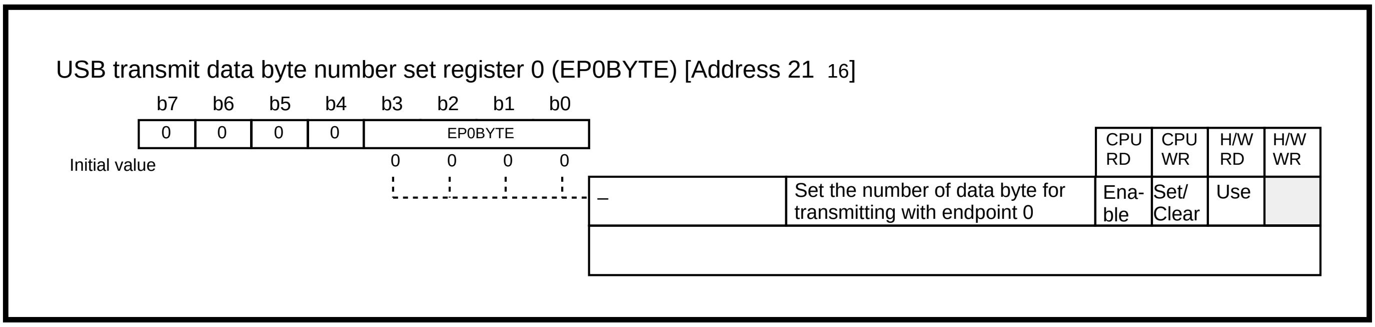
USB interrupt control register (USBICON) [Address 20$_{16}$]

	b7	b6	b5	b4	b3	b2	b1	b0
	USBE	TKNE	RSME	RSTE	EP1E			
Initial value	0	0	0	0	0			

		CPU RD	CPU WR	H/W RD	H/W WR
Endpoint 1 enable	0: Endpoint 1 invalid 1: Endpoint 1 valid	Ena-ble	Set/ Clear	Use	
USB reset interrupt enable	0:USB reset invalid 1:USB reset valid	Ena-ble	Set/ Clear	Use	
This flag is invalid in suspend mode (USB reset is always valid in suspend mode).					
Resume interrupt enable	0: Resumue invalid 1: Resume valid	Ena-ble	Set/ Clear	Use	
Token interrupt enable	0:Token invalid 1:Token valid	Ena-ble	Set/ Clear	Use	
USB enable flag	0:USB invalid 1:USB valid	Ena-ble	Set/ Clear	Use	
The internal state can be initialized by clearing this flag to “0”. The initial values of registers are as follows; • USB status register [address 19$_{16}$] = (01$_{16}$) • USB data toggle synchronization register [address 1D$_{16}$] = (7F$_{16}$) • USB interrupt source discrimination register 1 [address 1E$_{16}$] = (7F$_{16}$) • Bits 7, 6 and 2 of USB interrupt source discrimination register 2 [address 1F$_{16}$] = (00xxx0xx$_2$)					

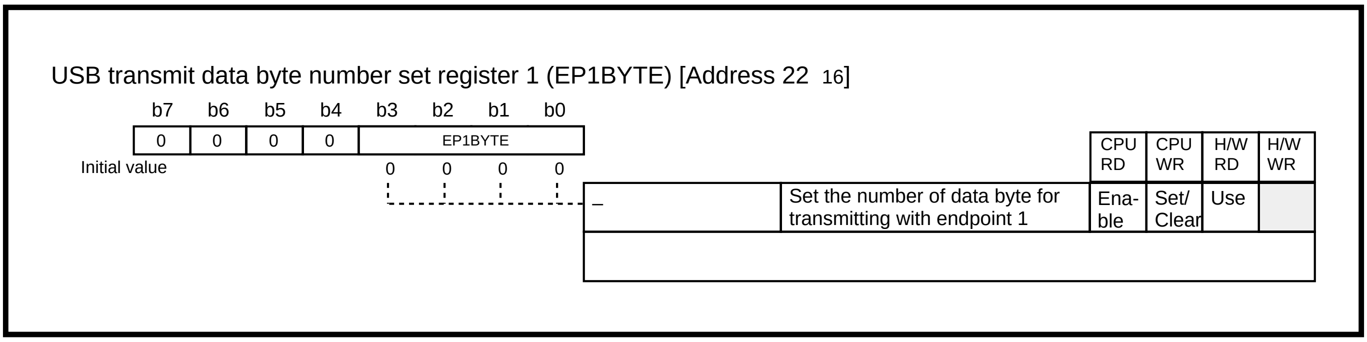
Fig. 3.5.14 Structure of USB interrupt control register

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


USB transmit data byte number set register 0 (EP0BYTE) [Address 21 16]
b7 b6 b5 b4 b3 b2 b1 b0
0 0 0 0 EP0BYTE
Initial value 0 0 0 0

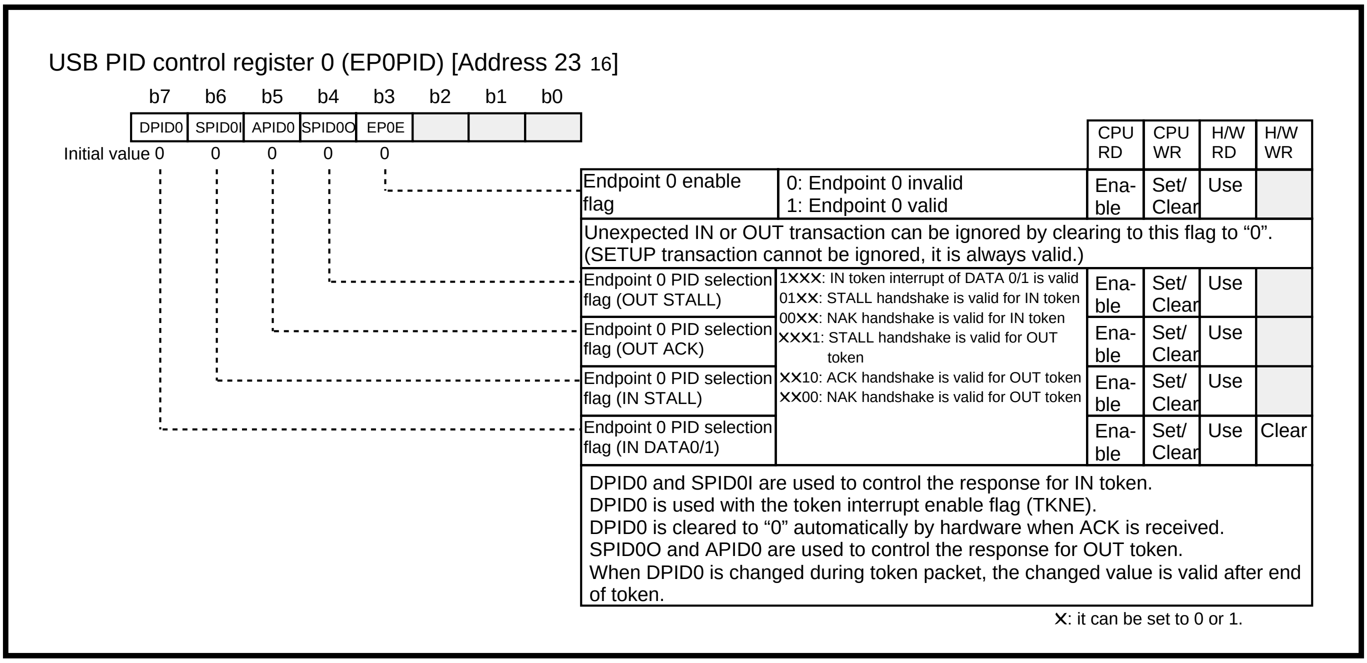
		CPU RD	CPU WR	H/W RD	H/W WR
–	Set the number of data byte for transmitting with endpoint 0	Ena-ble	Set/ Clear	Use	

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


USB transmit data byte number set register 1 (EP1BYTE) [Address 22 16]
b7 b6 b5 b4 b3 b2 b1 b0
0 0 0 0 EP1BYTE
Initial value 0 0 0 0

		CPU RD	CPU WR	H/W RD	H/W WR
–	Set the number of data byte for transmitting with endpoint 1	Ena-ble	Set/ Clear	Use	

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


USB PID control register 0 (EP0PID) [Address 23 16]
b7 b6 b5 b4 b3 b2 b1 b0
DPID0 SPID0I APID0 SPID0O EP0E
Initial value 0 0 0 0 0

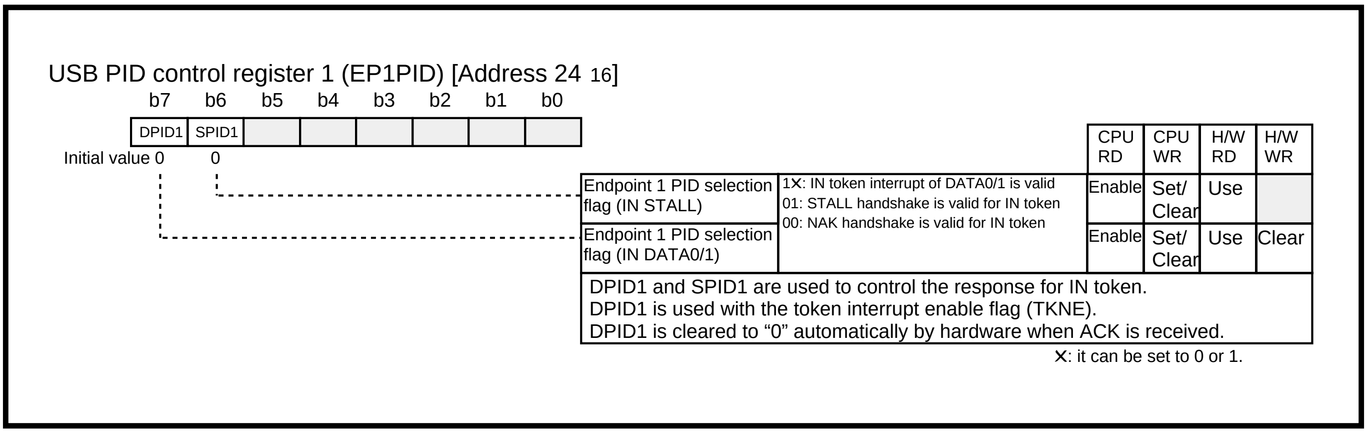
		CPU RD	CPU WR	H/W RD	H/W WR
Endpoint 0 enable flag	0: Endpoint 0 invalid 1: Endpoint 0 valid	Ena-ble	Set/ Clear	Use	
Unexpected IN or OUT transaction can be ignored by clearing to this flag to “0”. (SETUP transaction cannot be ignored, it is always valid.)					
Endpoint 0 PID selection flag (OUT STALL)	1××××: IN token interrupt of DATA 0/1 is valid 01××: STALL handshake is valid for IN token 00××: NAK handshake is valid for IN token ×××1: STALL handshake is valid for OUT token ××10: ACK handshake is valid for OUT token ××00: NAK handshake is valid for OUT token	Ena-ble	Set/ Clear	Use	
Endpoint 0 PID selection flag (OUT ACK)		Ena-ble	Set/ Clear	Use	
Endpoint 0 PID selection flag (IN STALL)		Ena-ble	Set/ Clear	Use	
Endpoint 0 PID selection flag (IN DATA0/1)		Ena-ble	Set/ Clear	Use	Clear

DPID0 and SPID0I are used to control the response for IN token.
DPID0 is used with the token interrupt enable flag (TKNE).
DPID0 is cleared to “0” automatically by hardware when ACK is received.
SPID0O and APID0 are used to control the response for OUT token.
When DPID0 is changed during token packet, the changed value is valid after end of token.

×: it can be set to 0 or 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.

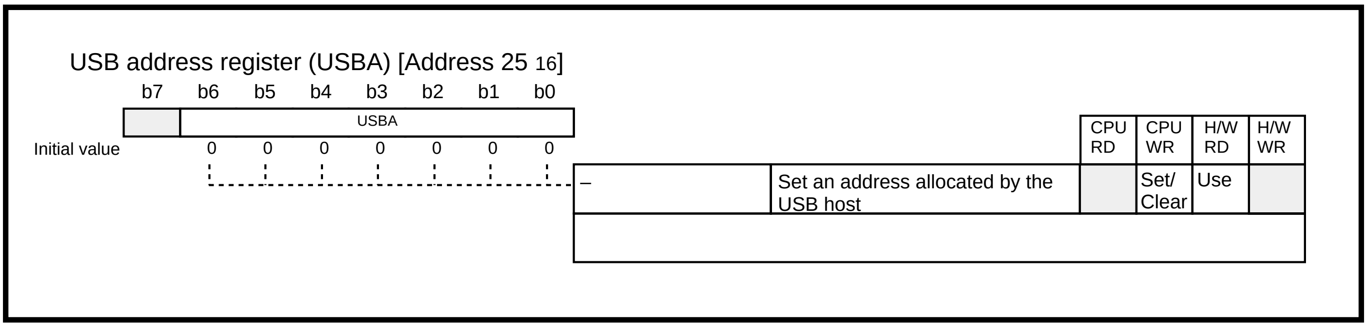


		CPU RD	CPU WR	H/W RD	H/W WR
Endpoint 1 PID selection flag (IN STALL)	1X: IN token interrupt of DATA0/1 is valid 01: STALL handshake is valid for IN token 00: NAK handshake is valid for IN token	Enable	Set/ Clear	Use	
Endpoint 1 PID selection flag (IN DATA0/1)		Enable	Set/ Clear	Use	Clear

DPID1 and SPID1 are used to control the response for IN token.
DPID1 is used with the token interrupt enable flag (TKNE).
DPID1 is cleared to “0” automatically by hardware when ACK is received.

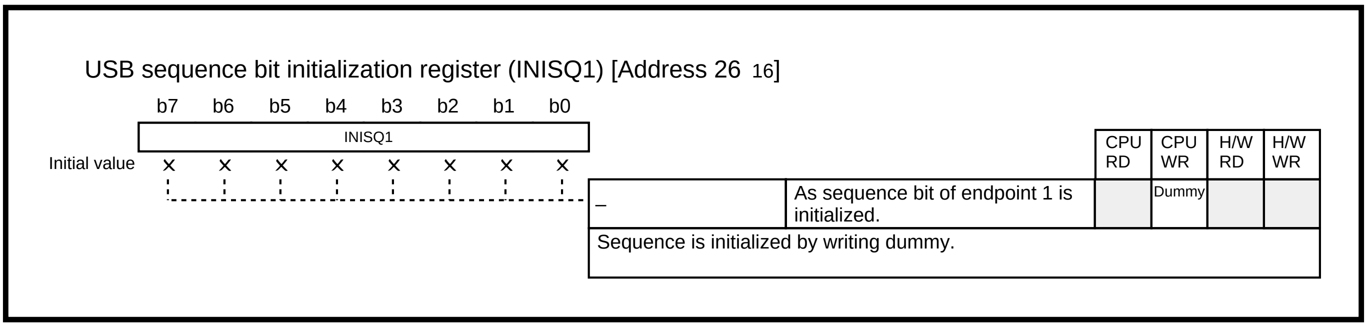
X: it can be set to 0 or 1.

Fig. 3.5.18 Structure of USB PID control register 1


USB address register (USBA) [Address 25 16]
b7 b6 b5 b4 b3 b2 b1 b0
USBA
Initial value 0 0 0 0 0 0 0

		CPU RD	CPU WR	H/W RD	H/W WR
–	Set an address allocated by the USB host		Set/ Clear	Use	

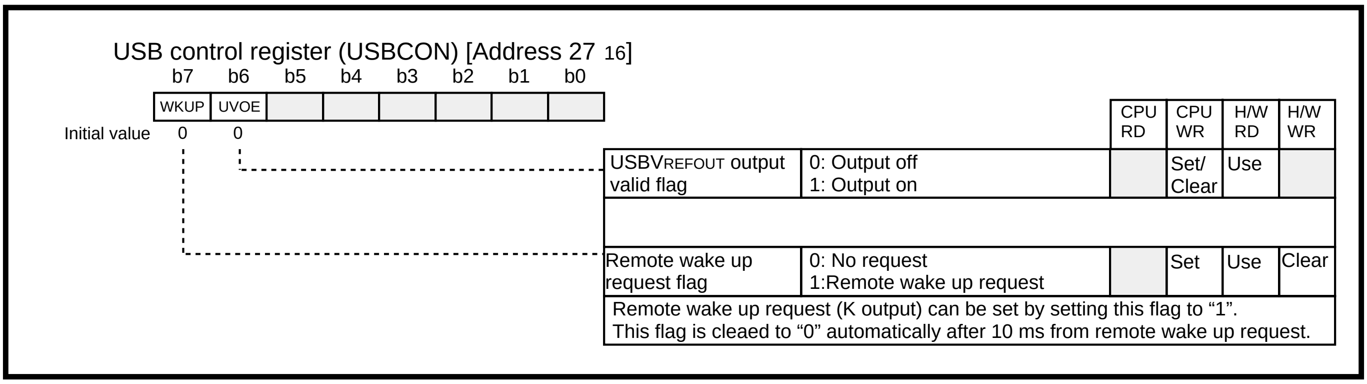
Fig. 3.5.19 Structure of USB address register


USB sequence bit initialization register (INISQ1) [Address 26 16]
b7 b6 b5 b4 b3 b2 b1 b0
INISQ1
Initial value X X X X X X X X

		CPU RD	CPU WR	H/W RD	H/W WR
–	As sequence bit of endpoint 1 is initialized.		Dummy		

Sequence is initialized by writing dummy.

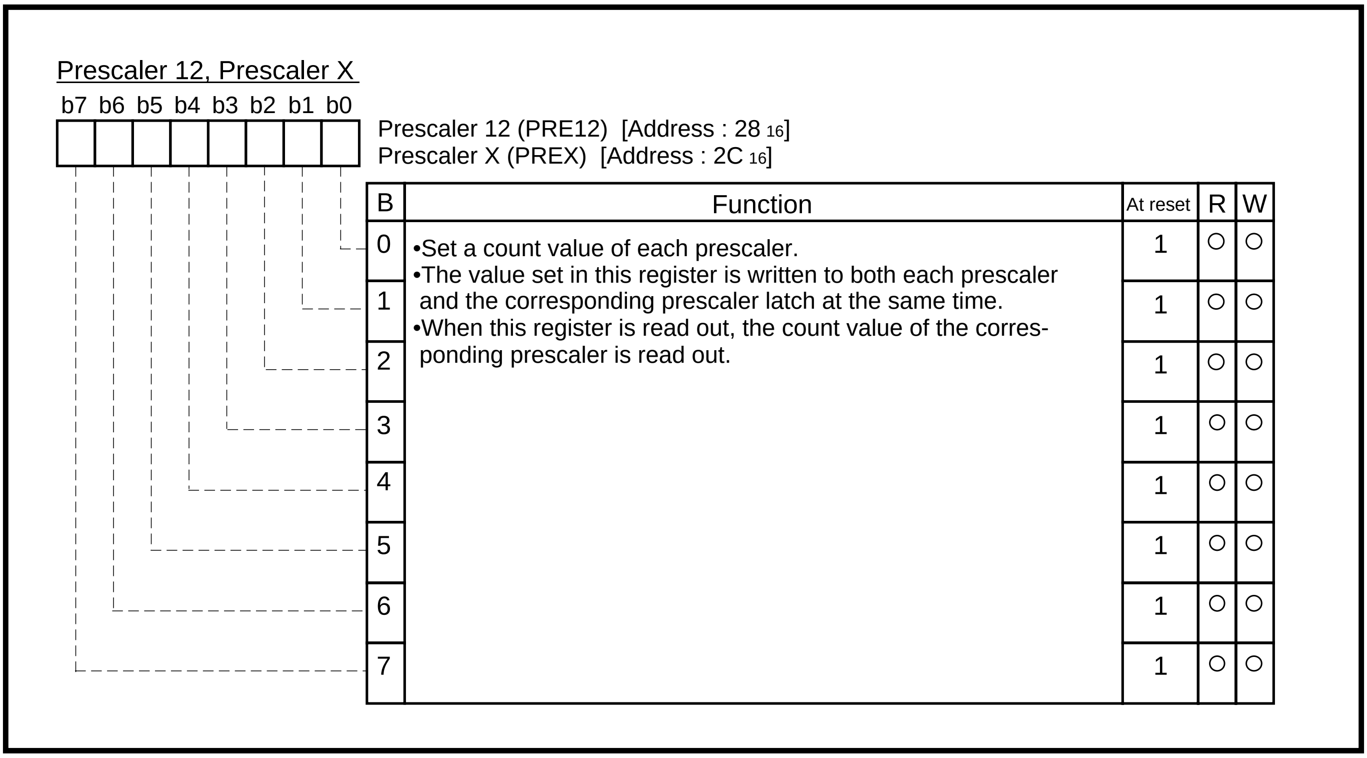
Fig. 3.5.20 Structure of USB sequence bit initialization register



		CPU RD	CPU WR	H/W RD	H/W WR
USBVREFOUT output valid flag	0: Output off 1: Output on		Set/ Clear	Use	
Remote wake up request flag	0: No request 1:Remote wake up request		Set	Use	Clear

Remote wake up request (K output) can be set by setting this flag to “1”.
This flag is cleaed to “0” automatically after 10 ms from remote wake up request.

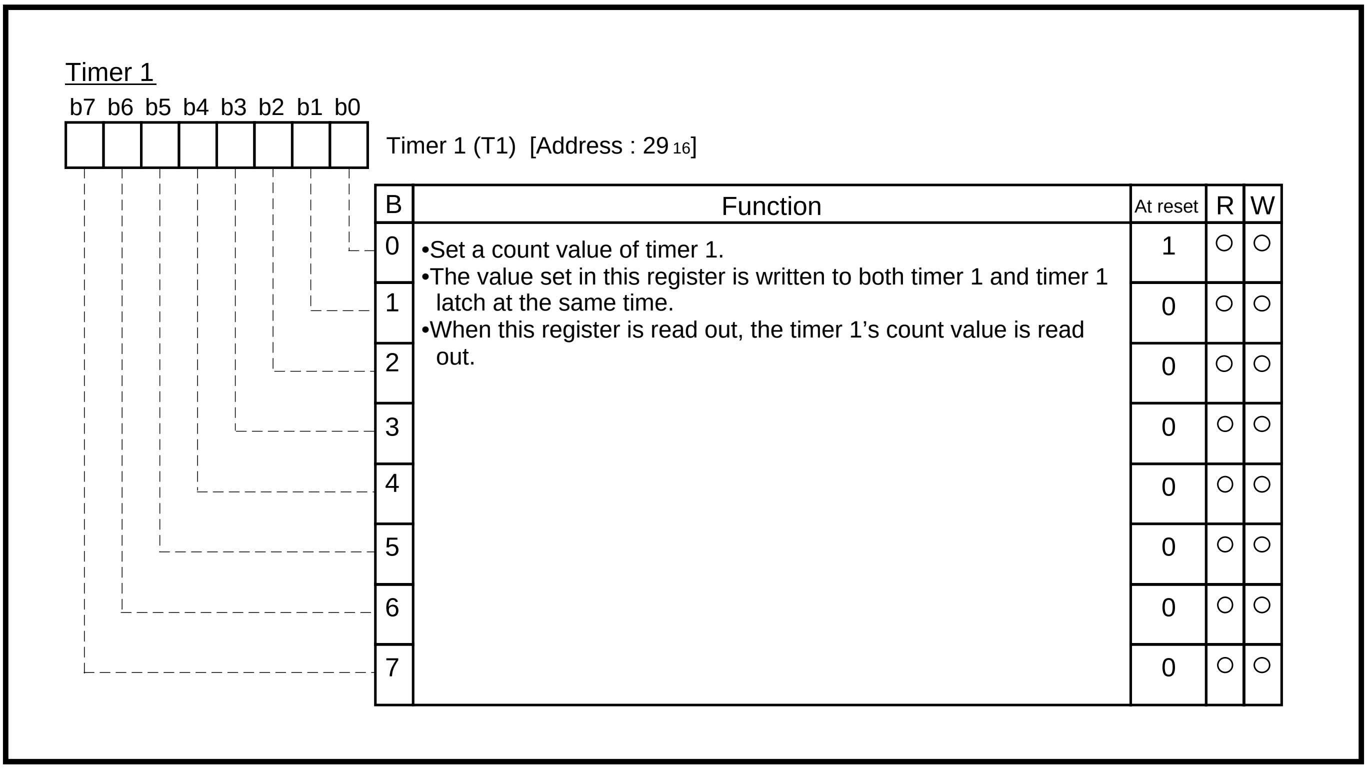
Fig. 3.5.21 Structure of USB control register



Prescaler 12 (PRE12) [Address : 28_{16}]
Prescaler X (PREX) [Address : $2C_{16}$]

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

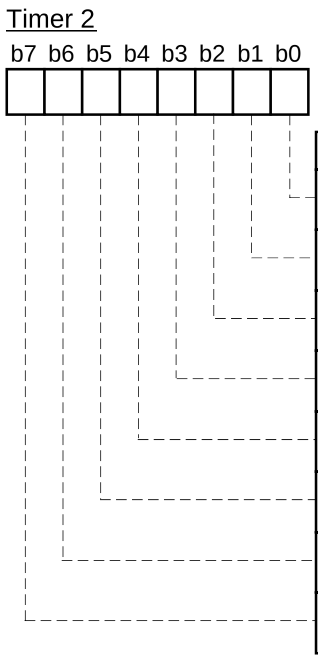
Fig. 3.5.22 Structure of Prescaler 12, Prescaler X



Timer 1 (T1) [Address : 29_{16}]

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

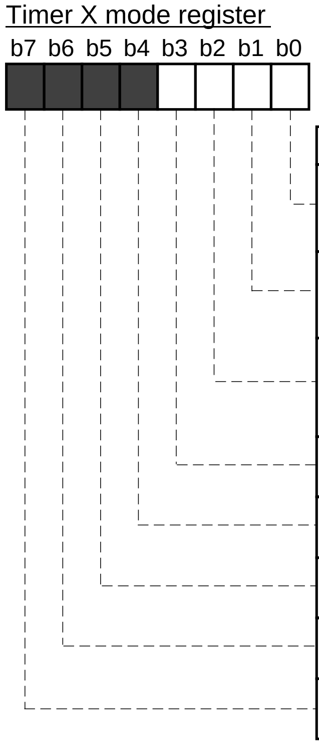
Fig. 3.5.23 Structure of Timer 1



Timer 2 (T2) [Address : 2A 16]

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

Fig. 3.5.24 Structure of Timer 2



Timer X mode register (TM) [Address : 2B $_{16}$]

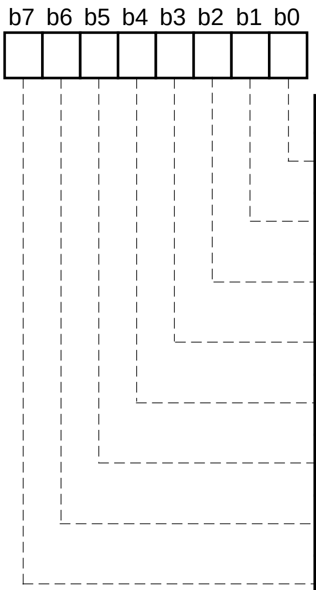
B	Name	Function	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
1			0	○	○
2	CNTR $_0$ active edge switch bit	The function depends on the operating mode. (Refer to Table 3.5.1)	0	○	○
3	Timer X count stop bit	0 : Count start 1 : Count stop	0	○	○
4	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 3.5.25 Structure of Timer X mode register

Table 3.5.1 CNTR $_0$ 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

Timer X

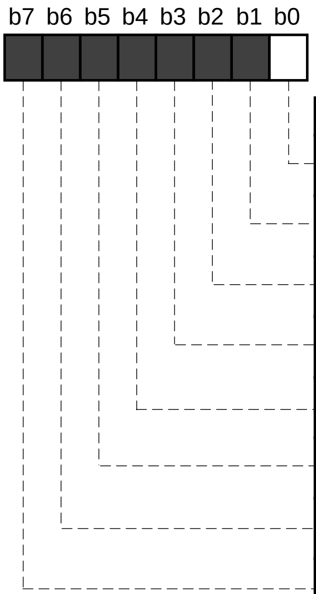


Timer X (TX) [Address : 2D $_{16}$]

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

Fig. 3.5.26 Structure of Timer X

Timer count source set register



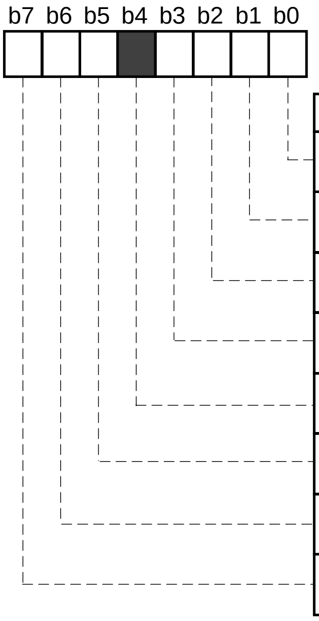
Timer count source set register (TCSS) [Address : 2E $_{16}$]

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

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

Fig. 3.5.27 Structure of Timer count source set register

Serial I/O2 control register



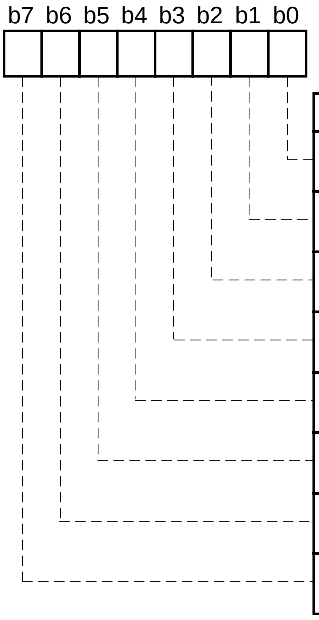
Serial I/O2 control register (SIO2CON) [Address : 30 16]

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

Note: When using it as a SDATA input, set the port P13 direction register bit to "0".

Fig. 3.5.28 Structure of Serial I/O2 control register

Serial I/O2 register

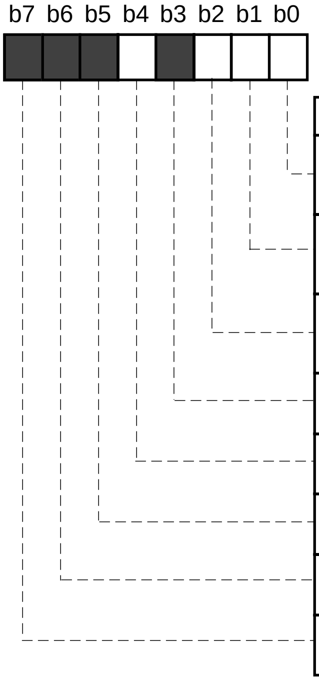


Serial I/O2 register (SIO2) [Address : 31 16]

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

Fig. 3.5.29 Structure of Serial I/O2 register

A/D control register



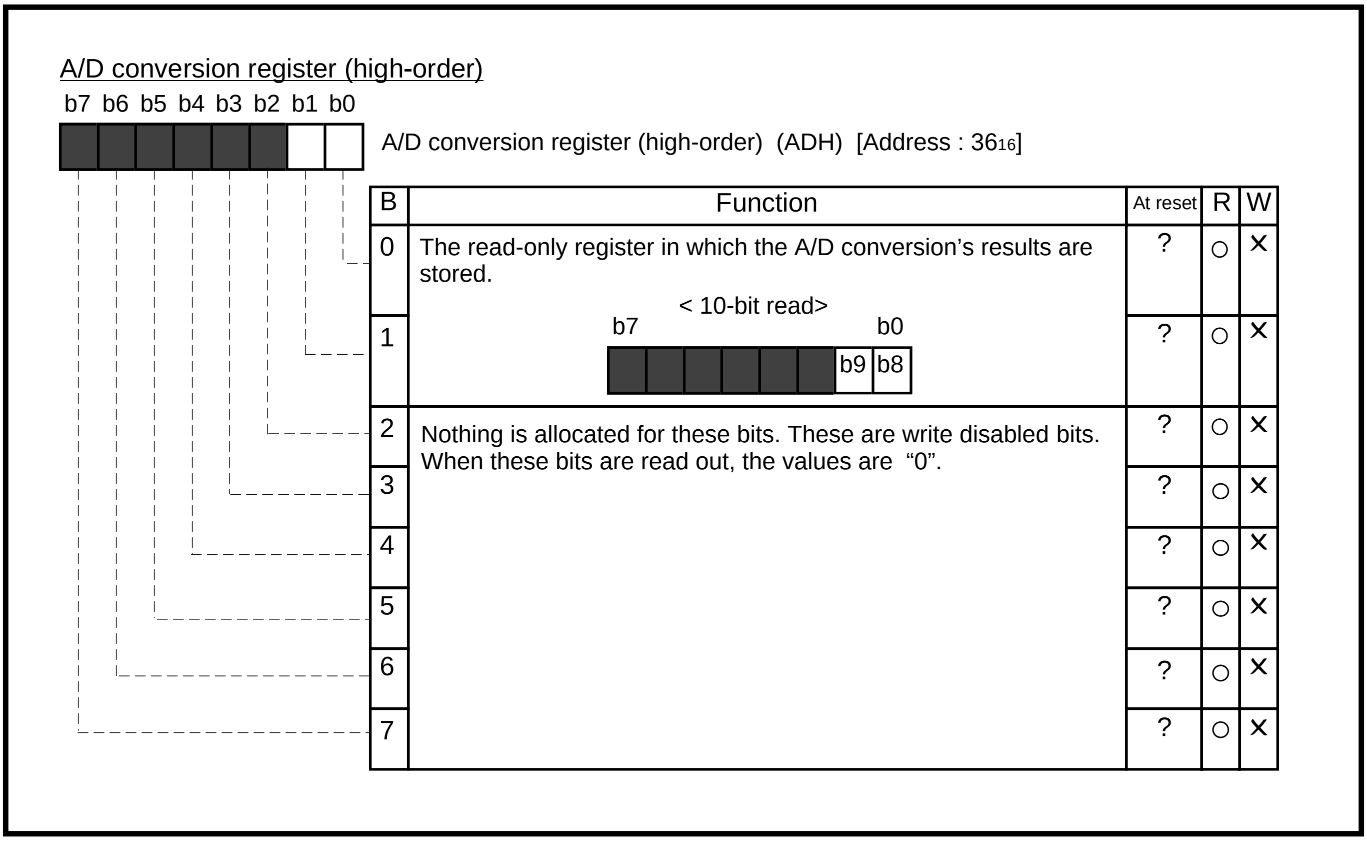
A/D control register (ADCON) [Address : 34_{16}]

B	Name	Function	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0 : $P2_0/AN_0$ 0 0 1 : $P2_1/AN_1$ 0 1 0 : $P2_2/AN_2$ 0 1 1 : $P2_3/AN_3$ 1 0 0 : $P2_4/AN_4$ 1 0 1 : $P2_5/AN_5$ 1 1 0 : $P2_6/AN_6$ (**Note**) 1 1 1 : $P2_7/AN_7$ (**Note**)	0	○	○
1			0	○	○
2			0	○	○
3	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×
4	AD conversion completion bit	0 : Conversion in progress 1 : Conversion completed	1	○	✻
5	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
6			0	○	×
7			0	○	×

Note: $P2_6/AN_6$, $P2_7/AN_7$ can be selected in the 36-pin and 42-pin versions.

✻: This bit can be cleared to "0" by program, but cannot be set to "1".

Fig. 3.5.30 Structure of A/D control register



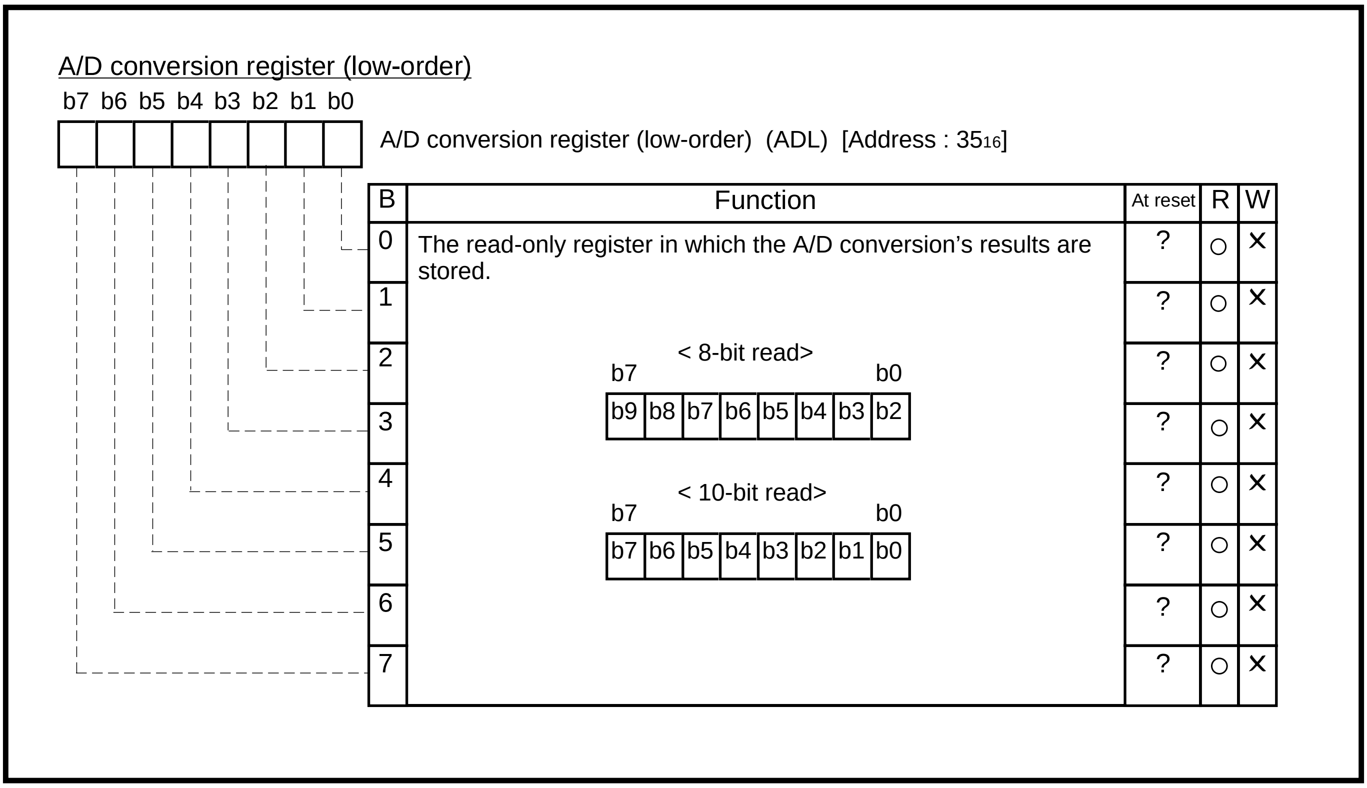
A/D conversion register (high-order)

b7 b6 b5 b4 b3 b2 b1 b0

A/D conversion register (high-order) (ADH) [Address : 36_{16}]

B	Function	At reset	R	W
0	The read-only register in which the A/D conversion's results are stored. < 10-bit read> b7 b0 ▨ ▨ ▨ ▨ ▨ ▨ b9 b8	?	○	×
1		?	○	×
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".	?	○	×
3		?	○	×
4		?	○	×
5		?	○	×
6		?	○	×
7		?	○	×

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



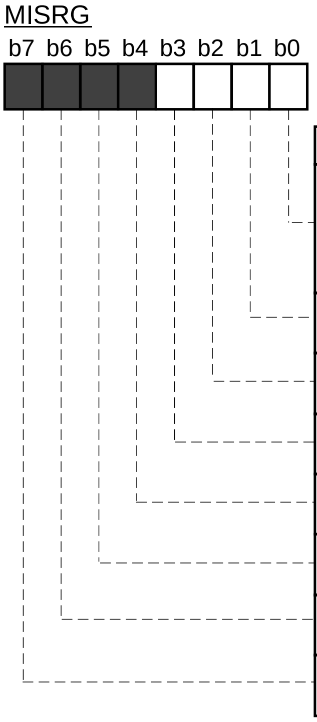
A/D conversion register (low-order)

b7 b6 b5 b4 b3 b2 b1 b0

A/D conversion register (low-order) (ADL) [Address : 35_{16}]

B	Function	At reset	R	W
0	The read-only register in which the A/D conversion's results are stored. < 8-bit read> b7 b0 b9 b8 b7 b6 b5 b4 b3 b2 < 10-bit read> b7 b0 b7 b6 b5 b4 b3 b2 b1 b0	?	○	×
1		?	○	×
2		?	○	×
3		?	○	×
4		?	○	×
5		?	○	×
6		?	○	×
7		?	○	×

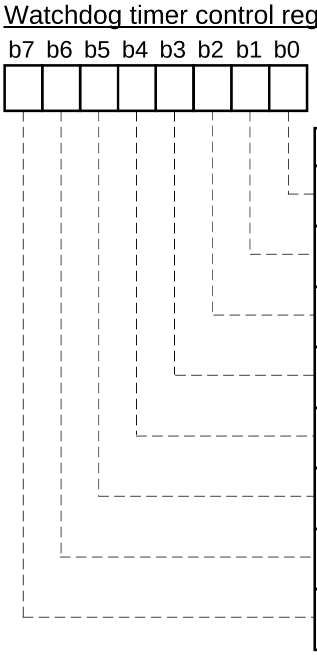
Fig. 3.5.32 Structure of A/D conversion register (low-order)



MISRG [Address : 38_{16}]

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

Fig. 3.5.33 Structure of MISRG

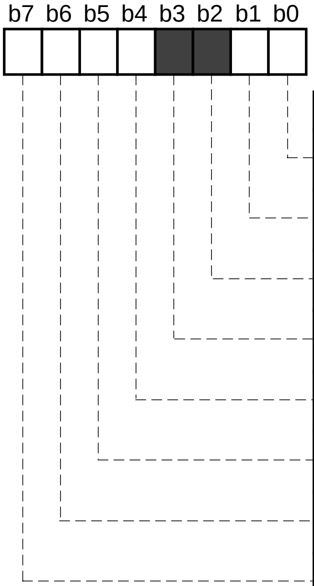


Watchdog timer control register (WDTCON) [Address : 39_{16}]

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

Fig. 3.5.34 Structure of Watchdog timer control register

Interrupt edge selection register



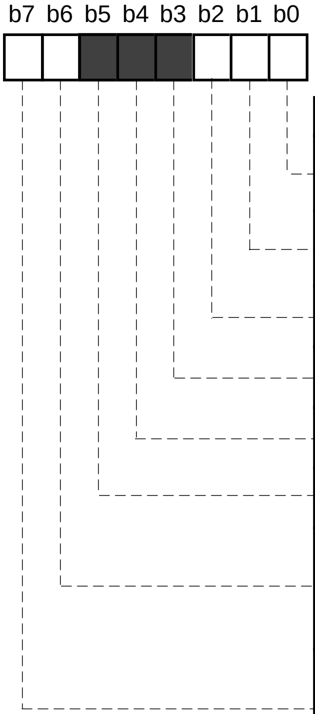
Interrupt edge selection register (INTEDGE) [Address : 3A $_{16}$]

B	Name	Function	At reset	R	W
0	INT$_0$ interrupt edge selection bit (**Note 1**)	0 : Falling edge active 1 : Rising edge active	0	○	○
1	INT$_1$ interrupt edge selection bit (**Note 2**)	0 : Falling edge active 1 : Rising edge active	0	○	○
2	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0".		0	○	×
3			0	○	×
4	Serial I/O1 or INT$_1$ interrupt selection bit	0 : Serial I/O1 1 : INT$_1$	0	○	○
5	Timer X or key-on wake up interrupt selection bit	0 : Timer X 1 : Key-on wake up	0	○	○
6	Timer 2 or serial I/O2 interrupt selection bit	0 : Timer 2 1 : Serial I/O2	0	○	○
7	CNTR$_0$ or AD converter interrupt selection bit	0 : CNTR$_0$ 1 : AD converter	0	○	○

Notes 1: 32-pin version: Not used.
This is a write disabled bit. When this bit is read out, the value is "0".
2: 36-pin and 32-pin version: Not used.
This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 3.5.35 Structure of Interrupt edge selection register

CPU mode register



CPU mode register (CPUM) [Address : 3B$_{16}$]

B	Name	Function	At reset	R	W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Not available 1 0 : Not available 1 1 : Not available	0	○	○
1			0	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the values are "0". (Do not write "1".)		0	○	×
4			0	○	×
5			0	○	×
6	Clock division ratio selection bits	b7 b6 0 0 : ϕ = f(X$_{IN}$)/2 (high-speedmode) 0 1 : ϕ = f(X$_{IN}$)/8 (middle-speed mode) 1 0 : Applied from on-chip oscillator 1 1 : ϕ = f(X$_{IN}$) (double-speed mode)	0	○	○
7			1	○	○

Fig. 3.5.36 Structure of CPU mode register

Interrupt request register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1) [Address : 3C 16]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	UART transmit/USBSETUP/ OUT token/Reset/Suspend/ Resume/INT1 interrupt request bit (**Note 1**)	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	INT0 interrupt request bit (**Note 2**)	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Timer X or key-on wake up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer 2 or serial I/O2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	CNTR0 or AD converter interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the value is "0".		0	○	×

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

Notes 1: 36-pin version and 32-pin version: INT1 interrupt does not exist.
2: 32-pin version: INT0 interrupt does not exist.
This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 3.5.37 Structure of Interrupt request register 1

Interrupt control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt control register 1 (ICON1) [Address : 3E 16]

B	Name	Function	At reset	R	W
0	UART receive/USBIN token interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	UART transmit/USBSETUP/ OUT token/Reset/Suspend/ Resume/INT1 interrupt enable bit (**Note 1**)	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	INT0 interrupt enable bit (**Note 2**)	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Timer X or key-on wake up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer 2 or serial I/O2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	CNTR0 or AD converter interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Nothing is allocated for this bit. Do not write "1" to this bit. When this bit is read out, the value is "0".		0	○	×

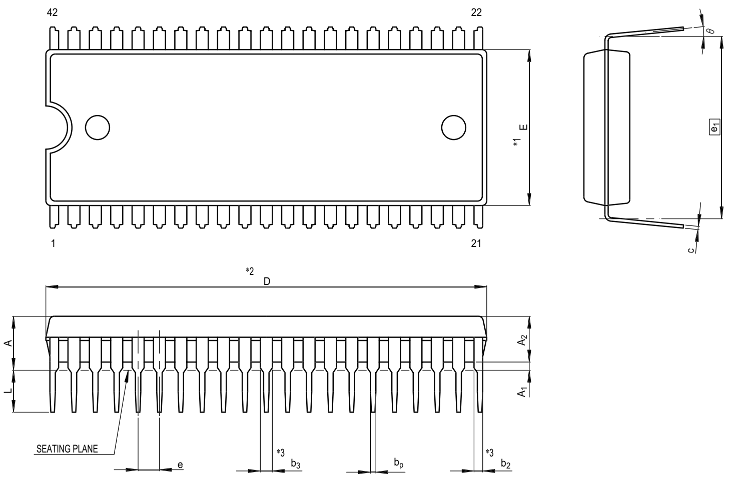
Notes 1: 36-pin version and 32-pin version: INT1 interrupt does not exist.
2: 32-pin version: INT0 interrupt does not exist.
This is a write disabled bit. When this bit is read out, the value is "0".

Fig. 3.5.38 Structure of Interrupt control register 1

3.6 Package outline

PRDP0042BA-A

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

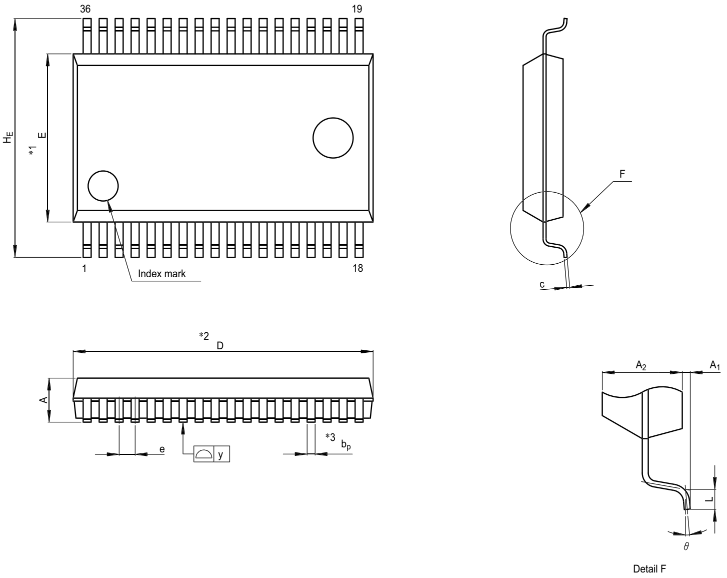


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
$\boxed{e_1}$	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

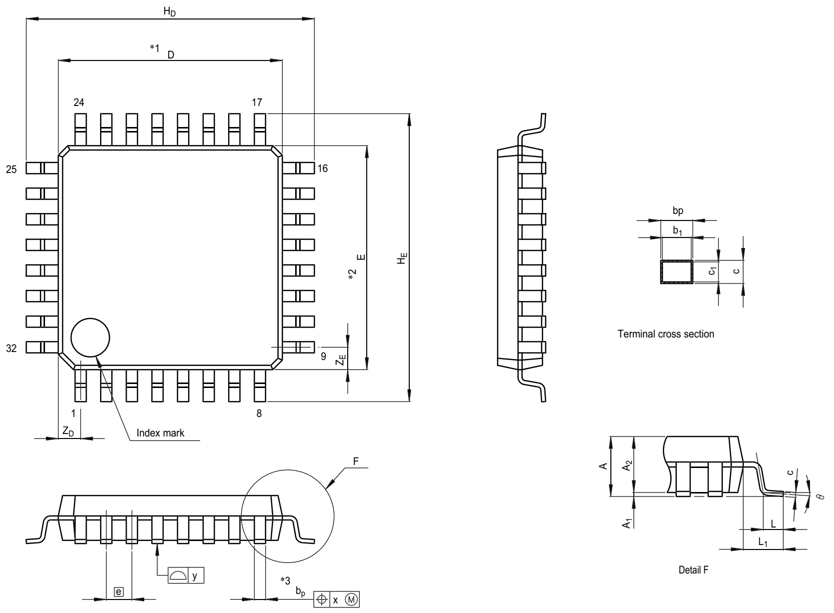


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

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

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP32-7x7-0.80	PLQP0032GB-A	32P6U-A	0.2g



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

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0	0.1	0.2
b_p	0.32	0.37	0.42
b_1	—	0.35	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
$\boxed{e}$	—	0.8	—
x	—	—	0.20
y	—	—	0.10
Z_D	—	0.7	—
Z_E	—	0.7	—
L	0.3	0.5	0.7
L_1	—	1.0	—

3.7 List of instruction code

D_7 – D_4	D_3 – D_0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	Hexadecimal notation	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND, X	JSR ZP, IND	BBS 0, A	—	ORA ZP	ASL ZP	BBS 0, ZP	PHP	ORA IMM	ASL A	SEB 0, A	—	ORA ABS	ASL ABS	SEB 0, ZP
0001	1	BPL	ORA IND, Y	CLT	BBC 0, A	—	ORA ZP, X	ASL ZP, X	BBC 0, ZP	CLC	ORA ABS, Y	DEC A	CLB 0, A	—	ORA ABS, X	ASL ABS, X	CLB 0, ZP
0010	2	JSR ABS	AND IND, X	JSR SP	BBS 1, A	BIT ZP	AND ZP	ROL ZP	BBS 1, ZP	PLP	AND IMM	ROL A	SEB 1, A	BIT ABS	AND ABS	ROL ABS	SEB 1, ZP
0011	3	BMI	AND IND, Y	SET	BBC 1, A	—	AND ZP, X	ROL ZP, X	BBC 1, ZP	SEC	AND ABS, Y	INC A	CLB 1, A	LDM ZP	AND ABS, X	ROL ABS, X	CLB 1, ZP
0100	4	RTI	EOR IND, X	STP	BBS 2, A	COM ZP	EOR ZP	LSR ZP	BBS 2, ZP	PHA	EOR IMM	LSR A	SEB 2, A	JMP ABS	EOR ABS	LSR ABS	SEB 2, ZP
0101	5	BVC	EOR IND, Y	—	BBC 2, A	—	EOR ZP, X	LSR ZP, X	BBC 2, ZP	CLI	EOR ABS, Y	—	CLB 2, A	—	EOR ABS, X	LSR ABS, X	CLB 2, ZP
0110	6	RTS	ADC IND, X	—	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 ← A + M + C When T = 1 M(X) ← 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 ← A ∧ M When T = 1 M(X) ← M(X) ∧ 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	7 0 C ← ▭ ← 0	This instruction shifts the content of A or M by one bit to the left, with bit 0 always being set to 0 and bit 7 of A or M always being contained in C.							0A	2	1				06	5	2			
BBC (Note 4)	Ai or Mi = 0?	This instruction tests the designated bit i of M or A and takes a branch if the bit is 0. The branch address is specified by a relative address. If the bit is 1, next instruction is executed.										13 + 20i	4	2				17 + 20i	5	3
BBS (Note 4)	Ai or Mi = 1?	This instruction tests the designated bit i of the M or A and takes a branch if the bit is 1. The branch address is specified by a relative address. If the bit is 0, next instruction is executed.										03 + 20i	4	2				07 + 20i	5	3
BCC (Note 4)	C = 0?	This instruction takes a branch to the appointed address if C is 0. The branch address is specified by a relative address. If C is 1, the next instruction is executed.																		
BCS (Note 4)	C = 1?	This instruction takes a branch to the appointed address if C is 1. The branch address is specified by a relative address. If C is 0, the next instruction is executed.																		
BEQ (Note 4)	Z = 1?	This instruction takes a branch to the appointed address when Z is 1. The branch address is specified by a relative address. If Z is 0, the next instruction is executed.																		
BIT	A ∧ M	This instruction takes a bit-wise logical AND of A and M contents; however, the contents of A and M are not modified. The contents of N, V, Z are changed, but the contents of A, M remain unchanged.													24	3	2			
BMI (Note 4)	N = 1?	This instruction takes a branch to the appointed address when N is 1. The branch address is specified by a relative address. If N is 0, the next instruction is executed.																		
BNE (Note 4)	Z = 0?	This instruction takes a branch to the appointed address if Z is 0. The branch address is specified by a relative address. If Z is 1, the next instruction is executed.																		

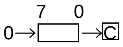
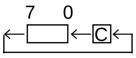
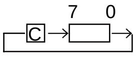
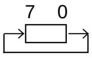
Addressing mode																																	Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
75	4	2				6D	4	3	7D	5	3	79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

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

Addressing mode																																	Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
																											10	2	2				•	•	•	•	•	•	•	•
																											80	4	2				•	•	•	•	•	•	•	•
																																	•	•	•	1	•	1	•	•
																											50	2	2				•	•	•	•	•	•	•	•
																											70	2	2				•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	0
																																	•	•	•	•	0	•	•	•
																																	•	•	•	•	•	0	•	•
																																	•	•	0	•	•	•	•	•
																																	•	0	•	•	•	•	•	•
D5	4	2				CD	4	3	DD	5	3	D9	5	3							C1	6	2	D1	6	2							N	•	•	•	•	•	Z	C
																																	N	•	•	•	•	•	Z	•
						EC	4	3																									N	•	•	•	•	•	Z	C
						CC	4	3																									N	•	•	•	•	•	Z	C
D6	6	2				CE	6	3	DE	7	3																						N	•	•	•	•	•	Z	•

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 ← X – 1	This instruction subtracts one from the current contents of X.	CA	2	1															
DEY	Y ← Y – 1	This instruction subtracts one from the current contents of Y.	88	2	1															
EOR (Note 1)	When T = 0 A ← A ∀ M When T = 1 M(X) ← M(X) ∀ 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 ← A + 1 or M ← M + 1	This instruction adds one to the contents of A or M.							3A	2	1				E6	5	2			
INX	X ← X + 1	This instruction adds one to the contents of X.	E8	2	1															
INY	Y ← Y + 1	This instruction adds one to the contents of Y.	C8	2	1															
JMP	If addressing mode is ABS PCL ← ADL PCH ← ADH If addressing mode is IND PCL ← M (ADH, ADL) PCH ← M (ADH, ADL + 1) If addressing mode is ZP, IND PCL ← M(00, ADL) PCH ← 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) ← PCH S ← S – 1 M(S) ← PCL S ← S – 1 After executing the above, if addressing mode is ABS, PCL ← ADL PCH ← ADH if addressing mode is SP, PCL ← ADL PCH ← FF If addressing mode is ZP, IND, PCL ← M(00, ADL) PCH ← 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 ← M When T = 1 M(X) ← 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 ← nn	This instruction loads the immediate value in M.													3C	4	3			
LDX	X ← M	This instruction loads the contents of M in X.				A2	2	2							A6	3	2			
LDY	Y ← 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	#	OP	n	#	N	V	T	B	D	I	Z	C
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
55	4	2				4D	4	3	5D	5	3	59	5	3							41	6	2	51	6	2							N	•	•	•	•	•	Z	•
F6	6	2				EE	6	3	FE	7	3																						N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
						4C	3	3							6C	5	3	B2	4	2													•	•	•	•	•	•	•	•
						20	6	3										02	7	2										22	5	2	•	•	•	•	•	•	•	•
B5	4	2				AD	4	3	BD	5	3	B9	5	3							A1	6	2	B1	6	2							N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
			B6	4	2	AE	4	3				BE	5	3																			N	•	•	•	•	•	Z	•
B4	4	2				AC	4	3	BC	5	3																						N	•	•	•	•	•	Z	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
LSR		This instruction shifts either A or M one bit to the right such that bit 7 of the result always is set to 0, and the bit 0 is stored in C.							4A	2	1				46	5	2			
NOP	PC ← PC + 1	This instruction adds one to the PC but does no otheroperation.	EA	2	1															
ORA (Note 1)	When T = 0 A ← A V M When T = 1 M(X) ← M(X) V M	When T = 0, this instruction transfers the contents of A and M to the ALU which performs a bit-wise “OR”, and stores the result in A. When T = 1, this instruction transfers the contents of M(X) and the M to the ALU which performs a bit-wise OR, and stores the result in M(X). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				09	2	2							05	3	2			
PHA	S ← S – 1	This instruction pushes the contents of A to the memory location designated by S, and decrements the contents of S by one.	48	3	1															
PHP	M(S) ← PS S ← S – 1	This instruction pushes the contents of PS to the memory location designated by S and decrements the contents of S by one.	08	3	1															
PLA	S ← S + 1 A ← M(S)	This instruction increments S by one and stores the contents of the memory designated by S in A.	68	4	1															
PLP	S ← S + 1 PS ← M(S)	This instruction increments S by one and stores the contents of the memory location designated by S in PS.	28	4	1															
ROL		This instruction shifts either A or M one bit left through C. C is stored in bit 0 and bit 7 is stored in C.							2A	2	1				26	5	2			
ROR		This instruction shifts either A or M one bit right through C. C is stored in bit 7 and bit 0 is stored in C.							6A	2	1				66	5	2			
RRF		This instruction rotates 4 bits of the M content to the right.													82	8	2			
RTI	S ← S + 1 PS ← M(S) S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S)	This instruction increments S by one, and stores the contents of the memory location designated by S in PS. S is again incremented by one and stores the contents of the memory location designated by S in PCL. S is again incremented by one and stores the contents of memory location designated by S in PCH.	40	6	1															
RTS	S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S) (PC) ← (PC) + 1	This instruction increments S by one and stores the contents of the memory location designated by S in PCL. S is again incremented by one and the contents of the memory location is stored in PCH. PC is incremented by 1.	60	6	1															

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

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
SBC (Note 1) (Note 5)	When T = 0 A ← A – M – $\overline{C}$ When T = 1 M(X) ← M(X) – M – $\overline{C}$	When T = 0, this instruction subtracts the value of M and the complement of C from A, and stores the results in A and C. When T = 1, the instruction subtracts the contents of M and the complement of C from the contents of M(X), and stores the results in M(X) and C. A remain unchanged, but status flag are changed. M(X) represents the contents of memory where is indicated by X.				E9	2	2							E5	3	2			
SEB	Ai or Mi ← 1	This instruction sets the designated bit i of A or M.										0B + 20i	2	1				0F + 20i	5	2
SEC	C ← 1	This instruction sets C.	38	2	1															
SED	D ← 1	This instruction set D.	F8	2	1															
SEI	I ← 1	This instruction set I.	78	2	1															
SET	T ← 1	This instruction set T.	32	2	1															
STA	M ← 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 ← X	This instruction stores the contents of X in M. The contents of X does not change.													86	4	2			
STY	M ← Y	This instruction stores the contents of Y in M. The contents of Y does not change.													84	4	2			
TAX	X ← A	This instruction stores the contents of A in X. The contents of A does not change.	AA	2	1															
TAY	Y ← 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 ← S	This instruction transfers the contents of S in X.	BA	2	1															
TXA	A ← X	This instruction stores the contents of X in A.	8A	2	1															
TXS	S ← X	This instruction stores the contents of X in S.	9A	2	1															
TYA	A ← Y	This instruction stores the contents of Y in A.	98	2	1															
WIT		The WIT instruction stops the internal clock but not the oscillation of the oscillation circuit is not stopped. CPU starts its function after the Timer X over flows (comes to the terminal count). All registers or internal memory contents except Timer X will not change during this mode. (Of course needs VDD).	C2	2	1															

Notes 1 : The number of cycles “n” is increased by 3 when T is 1.
2 : The number of cycles “n” is increased by 2 when T is 1.
3 : The number of cycles “n” is increased by 1 when T is 1.
4 : The number of cycles “n” is increased by 2 when branching has occurred.
5 : N, V, and Z flags are invalid in decimal operation mode.

Addressing mode																																	Processor status register							
ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2							N	V	•	•	•	•	Z	C
																																	•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	1
																																	•	•	•	•	1	•	•	•
																																	•	•	•	•	•	1	•	•
																																	•	•	1	•	•	•	•	•
95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	2							•	•	•	•	•	•	•	•
																																	•	•	•	•	•	•	•	•
			96	5	2	8E	5	3																									•	•	•	•	•	•	•	•
94	5	2				8C	5	3																									•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•
																																	N	•	•	•	•	•	Z	•
																																	•	•	•	•	•	•	•	•

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

3.9 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

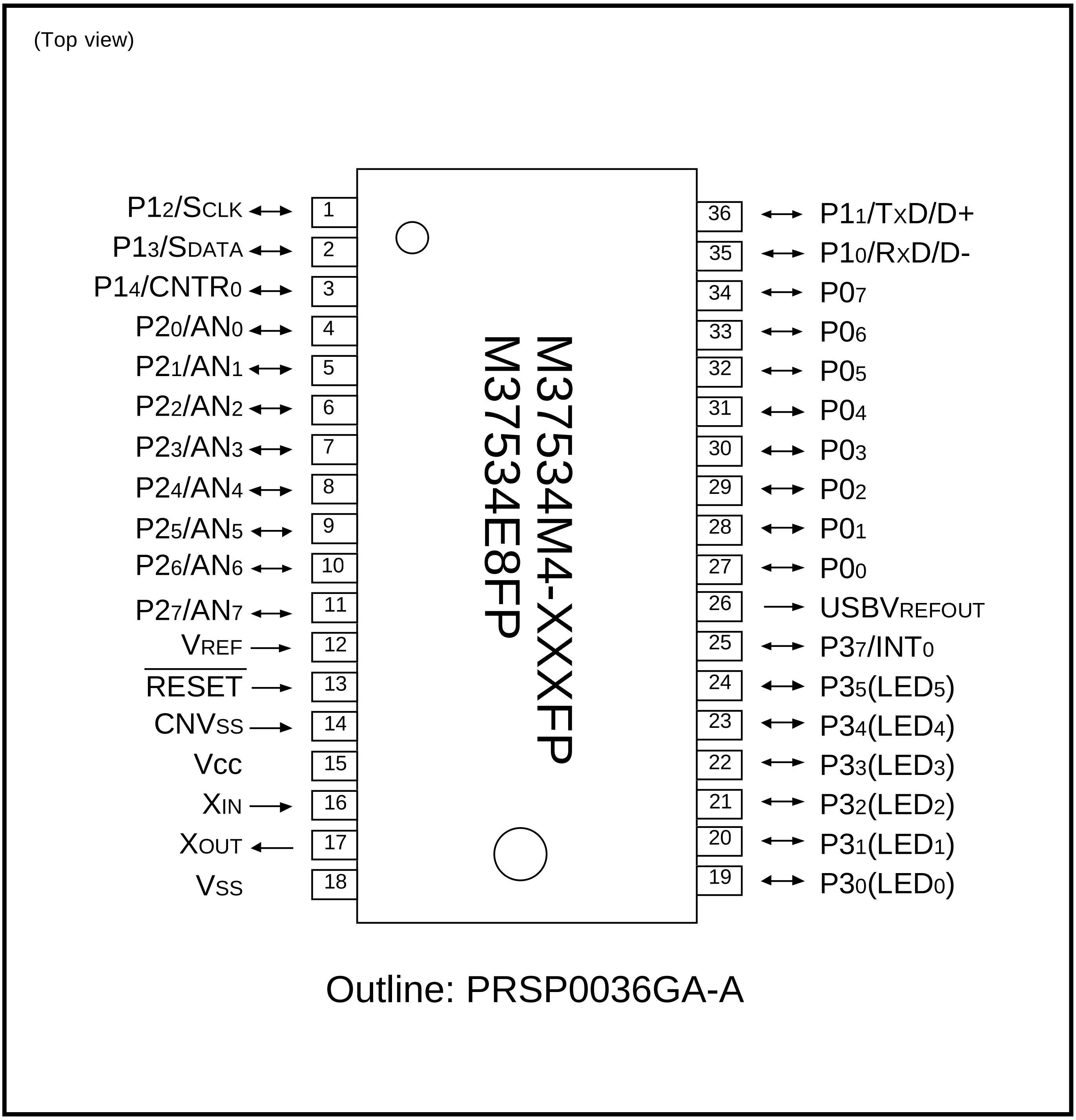


Fig. 3.10.1 M37534M4-XXXFP, M37534E8FP pin configuration

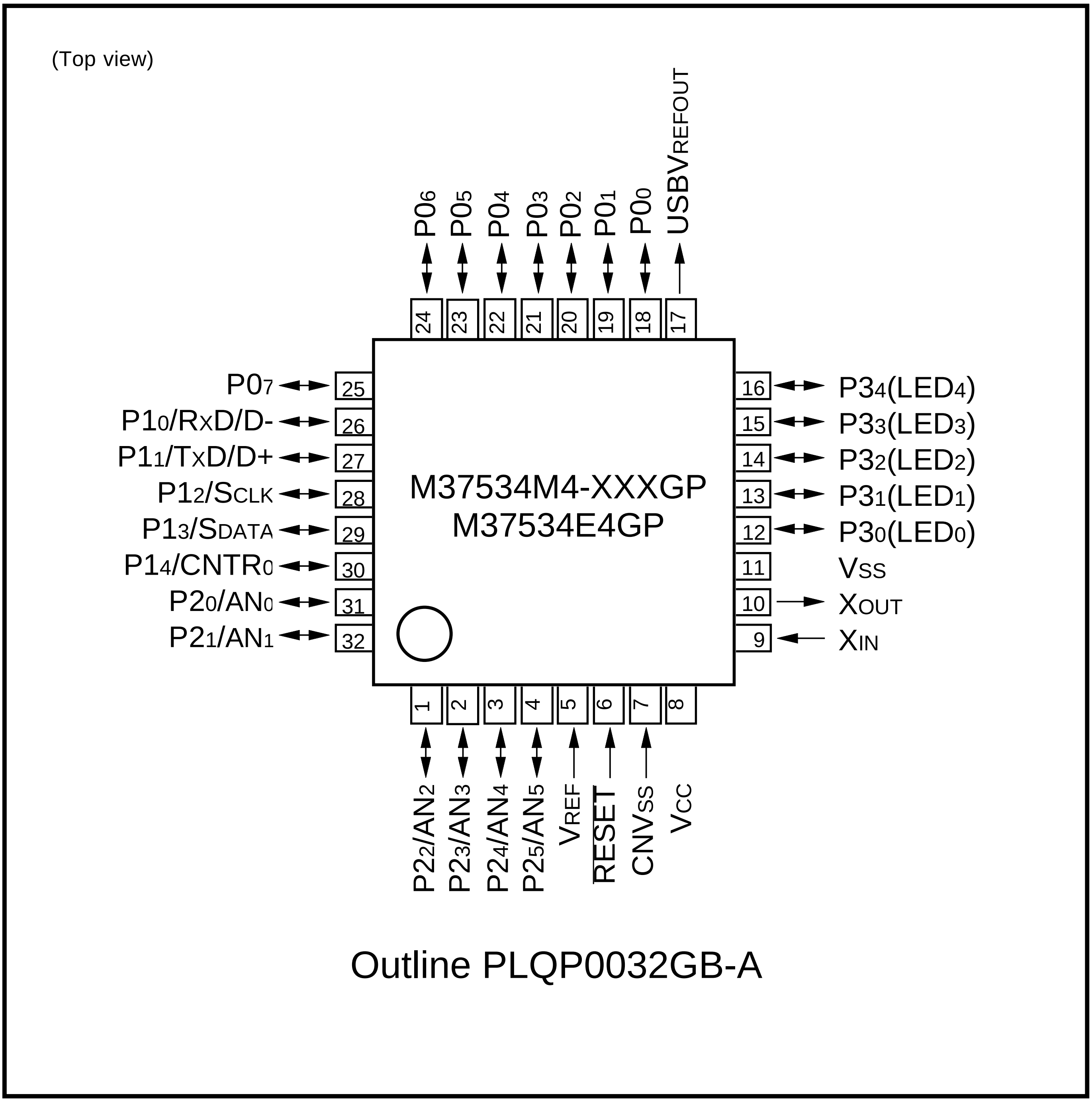


Fig. 3.10.2 M37534M4-XXXGP, M37534E4GP pin configuration

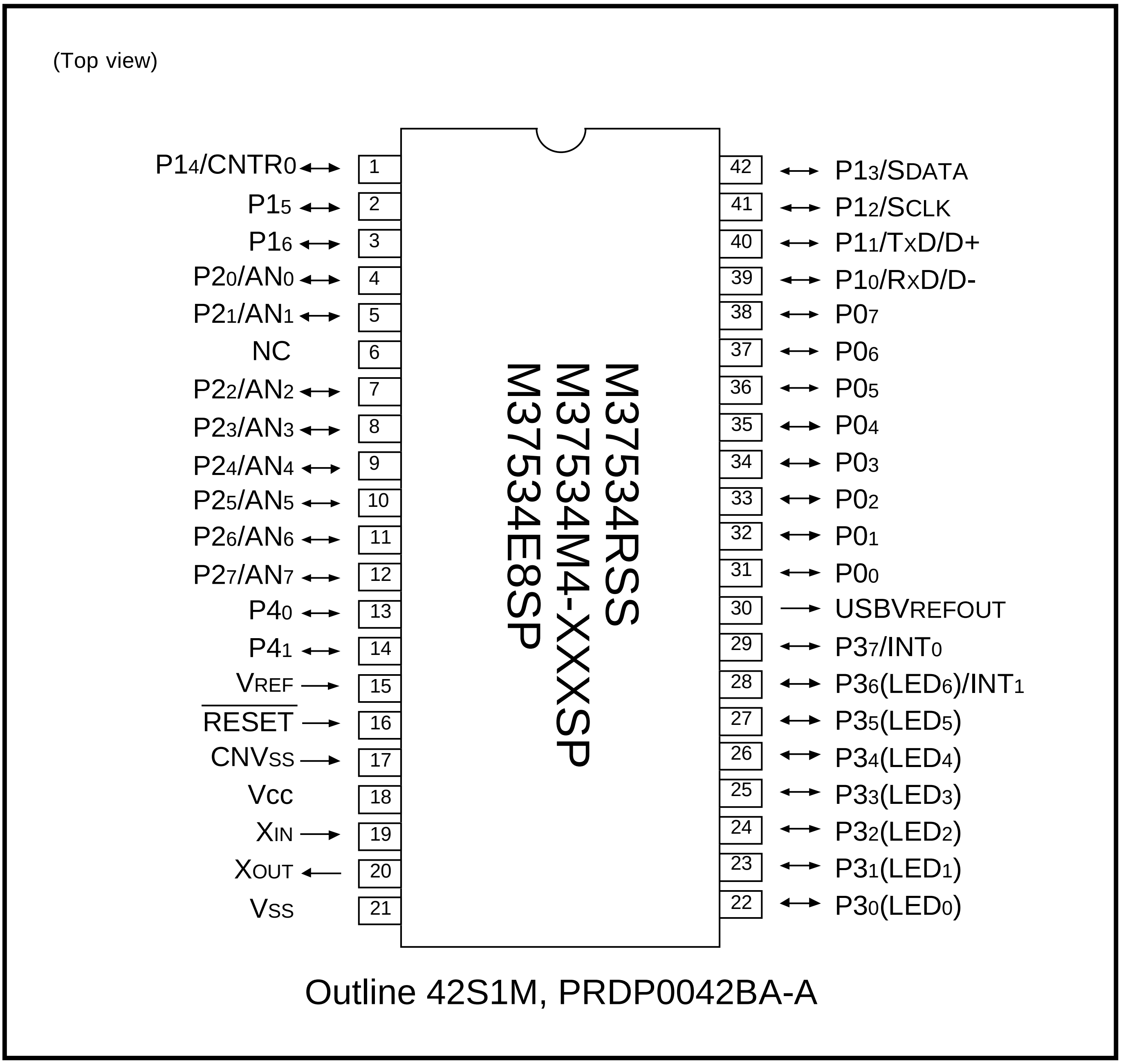


Fig. 3.10.3 M37534M4-XXXSP, M37534E8SP, M37534RSS pin configuration

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	Preface: Home page address revised.
		5	BEFORE USING THIS MANUAL: Home page address revised.
		1-8	Table 1 Function description of V_{SS}, V_{CC} revised.
		1-9	Fig. 7 “Under development” eliminated.
		1-13	“Note on stack page” added.
		1-19	Fig. 17 Ports $P3_6$, $P3_7$ revised.
		1-42	NOTES ON PROGRAMMING “Note on Stack Page” added.
		1-43	Table 8 32P6U-A added. Name of Programming Adapter revised.
2.00	Jun. 21, 2004	All pages	Words standardized: On-chip oscillator, A/D converter
		1-42	Electric Characteristic Difference Among Mask ROM and One Time PROM Version MCUs added.
		1-43	Note on Power Source Voltage added.
			DATA REQUIRED FOR MASK ORDERS revised.
		3-26	3.3.12 Electric Characteristic Difference Among Mask ROM and One Time PROM Version MCUs,
			3.3.13 Note on Power Source Voltage added.
		3-53	32P6U-A revised.
3.00	Oct. 23, 2006	All pages	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
		Chapter 1	
		40	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
		41	Fig. 48; NOTE 2 added
		42	NOTES ON PROGRAMMING; Watchdog Timer added
			NOTES ON USE; USB Communication added
		43	NOTES ON USE; Note on A/D Converter added
		Chapter 2	
		50	2.4.1 Transfer type: “Hi-Speed function: H.S.” added
		69	2.4.5; USB Communication added
		77	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 Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan