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.

# 7542 Group

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER

REJ03B0006-0303
Rev.3.03
Jul 11, 2008

## DESCRIPTION

The 7542 Group is the 8-bit microcomputer based on the 740 family core technology.

The 7542 Group has serial interfaces, 8-bit timers, 16-bit timers, and an A/D converter, and is useful for control of home electric appliances and office automation equipment.

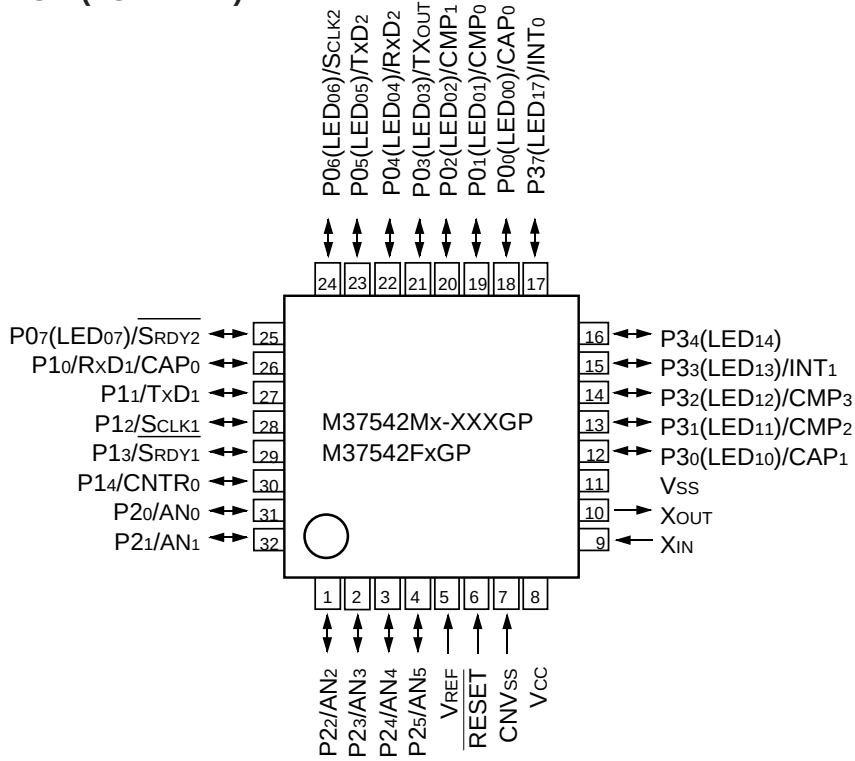
## FEATURES

- Basic machine-language instructions ..... 71
- The minimum instruction execution time ..... 0.25 μs
  (at 8 MHz oscillation frequency, double-speed mode for the shortest instruction)
- Memory size
  Flash memory version: ROM ..... 16 to 32K + 4K bytes
  RAM ..... 1024 bytes
  Mask ROM version: ROM ..... 8K to 16K bytes
  RAM ..... 384 to 512 bytes
  RSS version RAM ..... 1024 bytes
- Programmable I/O ports
  29 (25 in 32-pin version and PWQN0036KA-A package version)
- Interrupts ..... 18 sources, 16 vectors
- Timers ..... 8-bit × 2
  ..... 16-bit × 2
- Output compare ..... 4-channel
- Input capture ..... 2-channel
- Serial interface ..... 8-bit × 2 (UART or Clock-synchronized)
- A/D converter ..... 10-bit × 8 channels
  (6 channels in 32-pin version and PWQN0036KA-A package version)
- Clock generating circuit ..... Built-in type
  (low-power dissipation by an on-chip oscillator)
  (connected to external ceramic resonator or quartz-crystal oscillator permitting RC oscillation)
- Watchdog timer ..... 16-bit × 1
- Power source voltage
  XIN oscillation frequency at ceramic oscillation, in double-speed mode
  At 8 MHz ..... 4.5 to 5.5 V
  XIN oscillation frequency at ceramic oscillation, in high-speed mode
  At 8 MHz ..... 4.0 to 5.5 V
  At 4 MHz ..... 2.4 to 5.5 V
  At 2 MHz ..... 2.2 to 5.5 V
  XIN oscillation frequency at RC oscillation in high-speed mode or middle-speed mode
  At 4 MHz ..... 4.0 to 5.5 V
  At 2 MHz ..... 2.4 to 5.5 V
  At 1 MHz ..... 2.2 to 5.5 V
- Power dissipation ..... 27.5 mW (Typ.)
- Operating temperature range ..... –20 to 85 °C

## APPLICATION

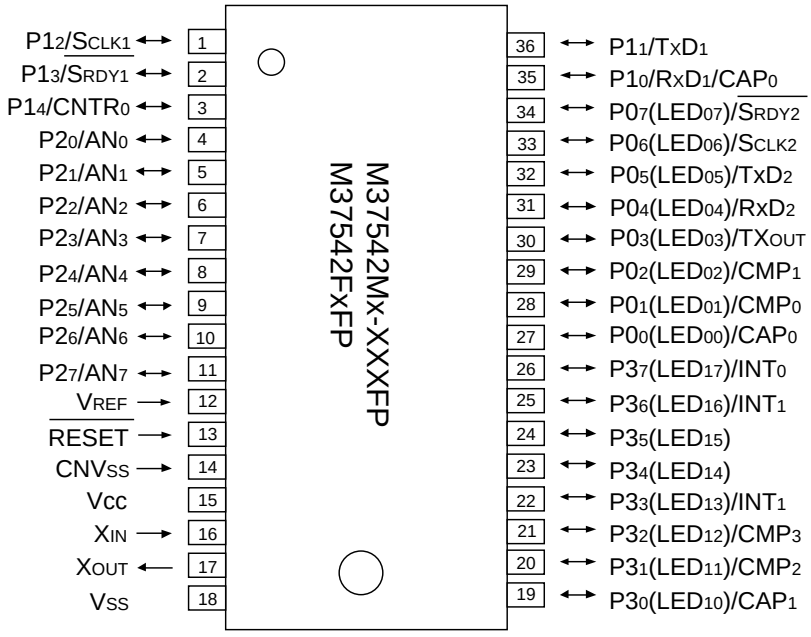
Office automation equipment, factory automation equipment, home electric appliances, consumer electronics, etc.

## PIN CONFIGURATION (TOP VIEW)



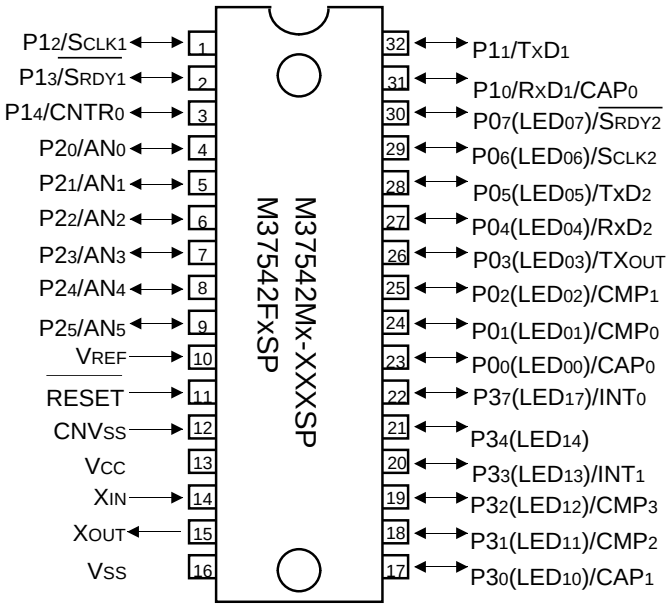

Outline PLQP0032GB-A (32P6U-A)

**Fig. 1 Pin configuration (Package type: PLQP0032GB-A)**



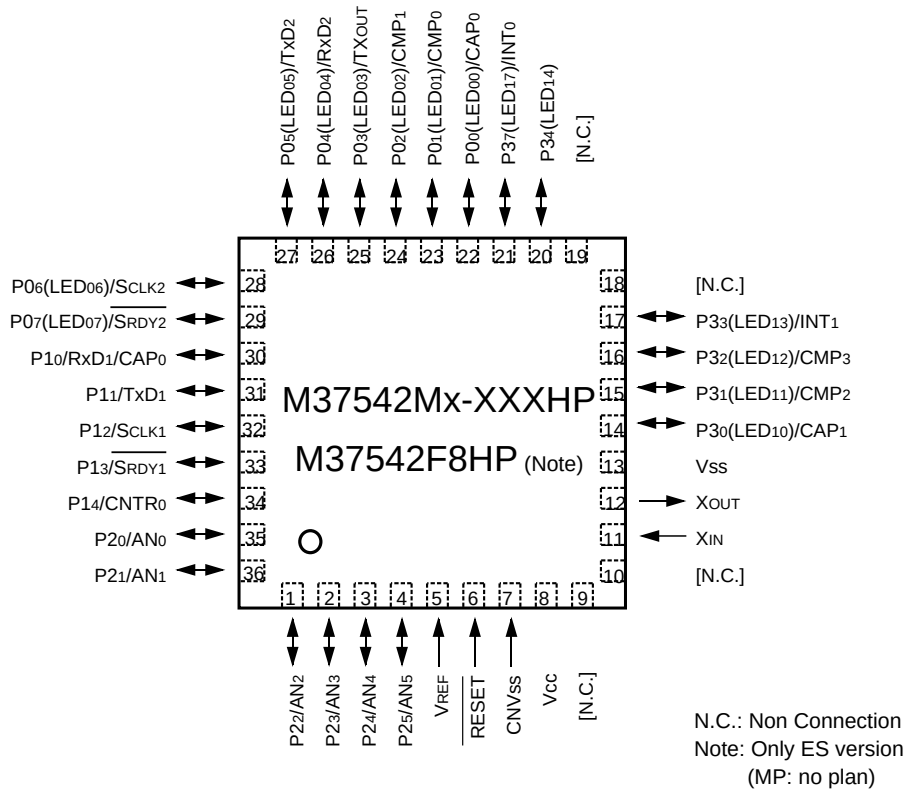

Package type: PRSP0036GA-A (36P2R-A)

**Fig. 2 Pin configuration (Package type: PRSP0036GA-A)**




Package type: PRDP0032BA-A (32P4B)

**Fig. 3 Pin configuration (Package type: PRDP0032BA-A)**




Package type: PWQN0036KA-A (36PJW-A)

**Fig. 4 Pin configuration (Package type: PWQN0036KA-A)**

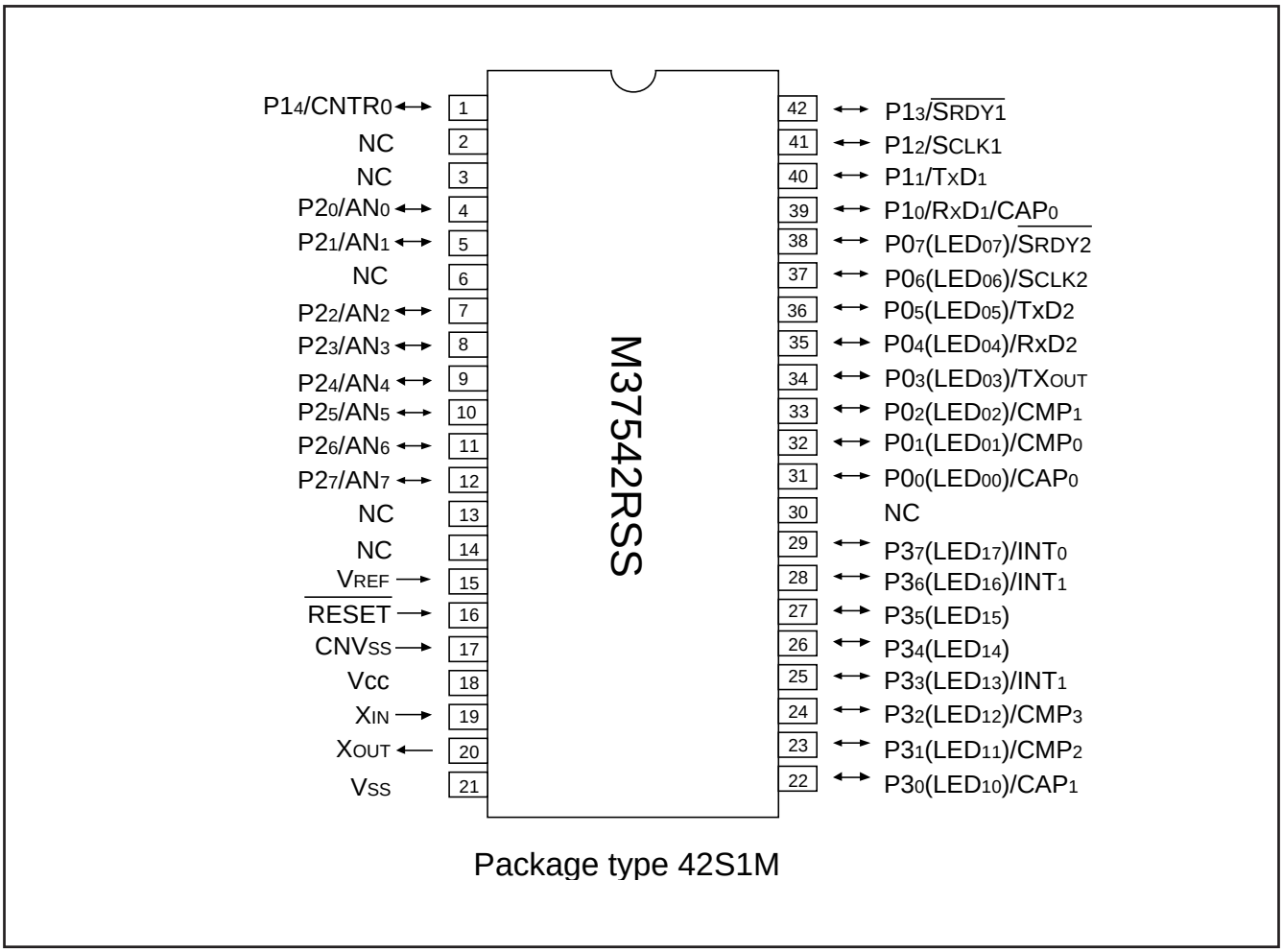



**Fig. 5 Pin configuration (Package type: 42S1M)**

Table 1 Performance overview

| Parameter | | | | Function |
|---|---|---|---|---|
| Number of basic instructions | | | | 71 |
| Instruction execution time | | | | 0.25 μs<br>(Minimum instruction, oscillation frequency 8 MHz: double-speed mode) |
| Oscillation frequency | | | | 8 MHz (max.) |
| Memory sizes | Mask ROM | ROM | | 8 K to 16 K bytes |
| | | RAM | | 384 to 512 bytes |
| | FLASH ROM | ROM | | 16 K to 32 K + 4 K bytes |
| | | RAM | | 1024 bytes |
| I/O port | P0, P1, P2, P3 | | | •8-bit × 3, 5-bit × 1 (8-bit × 1, 6-bit × 2, 5-bit × 1 for 32-pin version and PWQN0036KA-A package version) |
| Interrupts | | | | 18 sources, 16 vectors |
| Timer | | | | •8-bit × 2, 16-bit × 2 |
| Output compare | | | | 4 channel |
| Input capture | | | | 2 channel |
| Serial interface | | | | 8-bit × 2 (UART or clock synchronous) |
| A/D converter | | | | 10-bit × 8 channel<br>(6 channel for 32-pin version and PWQN0036KA-A package version) |
| Watchdog timer | | | | 16-bit × 1 |
| Clock generating circuit | | | | Built-in<br>(external ceramic resonator or quartz-crystal oscillator, RC oscillation available)<br>(Low consumption current by on-chip oscillator available) |
| Power source voltage (at ceramic resonance) | High-speed mode<br>Middle-speed mode | At 8MHz oscillation | Mask ROM | 4.0 to 5.5 V |
| | | | FLASH ROM | |
| | | At 4MHz oscillation | Mask ROM | 2.4 to 5.5 V |
| | | | FLASH ROM | 2.7 to 5.5 V |
| | | At 2MHz oscillation | Mask ROM | 2.2 to 5.5 V |
| | | | FLASH ROM | 2.7 to 5.5 V |
| | Double-speed mode | At 8MHz oscillation | Mask ROM | 4.5 to 5.5 V |
| | | | FLASH ROM | |
| | | At 6.5MHz oscillation | Mask ROM | 4.5 to 5.5 V |
| | | | FLASH ROM | |
| | | At 2MHz oscillation | Mask ROM | 2.4 to 5.5 V |
| | | | FLASH ROM | 2.7 to 5.5 V |
| | | At 1MHz oscillation | Mask ROM | 2.2 to 5.5 V |
| | | | FLASH ROM | 2.7 to 5.5 V |
| Power source voltage (at RC oscillation) | High-speed mode<br>Middle-speed mode | At 4MHz oscillation | Mask ROM | 4.5 to 5.5 V |
| | | | FLASH ROM | |
| | | At 2MHz oscillation | Mask ROM | 2.4 to 5.5 V |
| | | | FLASH ROM | 2.7 to 5.5 V |
| | | At 1MHz oscillation | Mask ROM | 2.2 to 5.5 V |
| | | | FLASH ROM | 2.7 to 5.5 V |
| Power dissipation | | | Mask ROM | 27.5 mW (Typ.) |
| | | | FLASH ROM | 24.0 mW (Typ.) |
| Operating temperature range | | | | -20 to 85 °C |
| Device structure | | | | CMOS silicon gate |
| Package | | | | 32-pin plastic molded SDIP/LQFP, 36-pin plastic molded SSOP/WQFN |

## FUNCTIONAL BLOCK

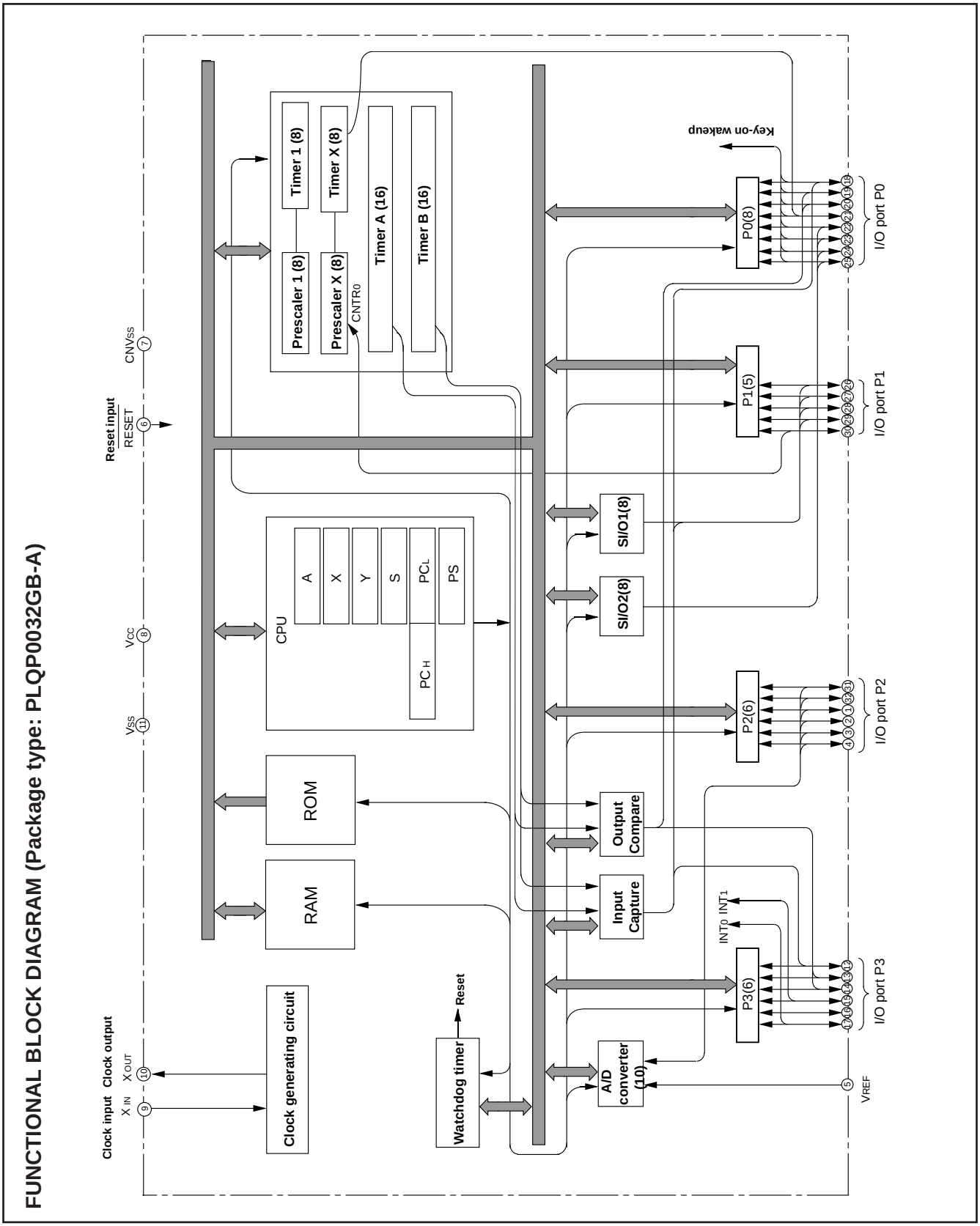



**Fig. 6 Functional block diagram (Package type: PLQP0032GB-A)**

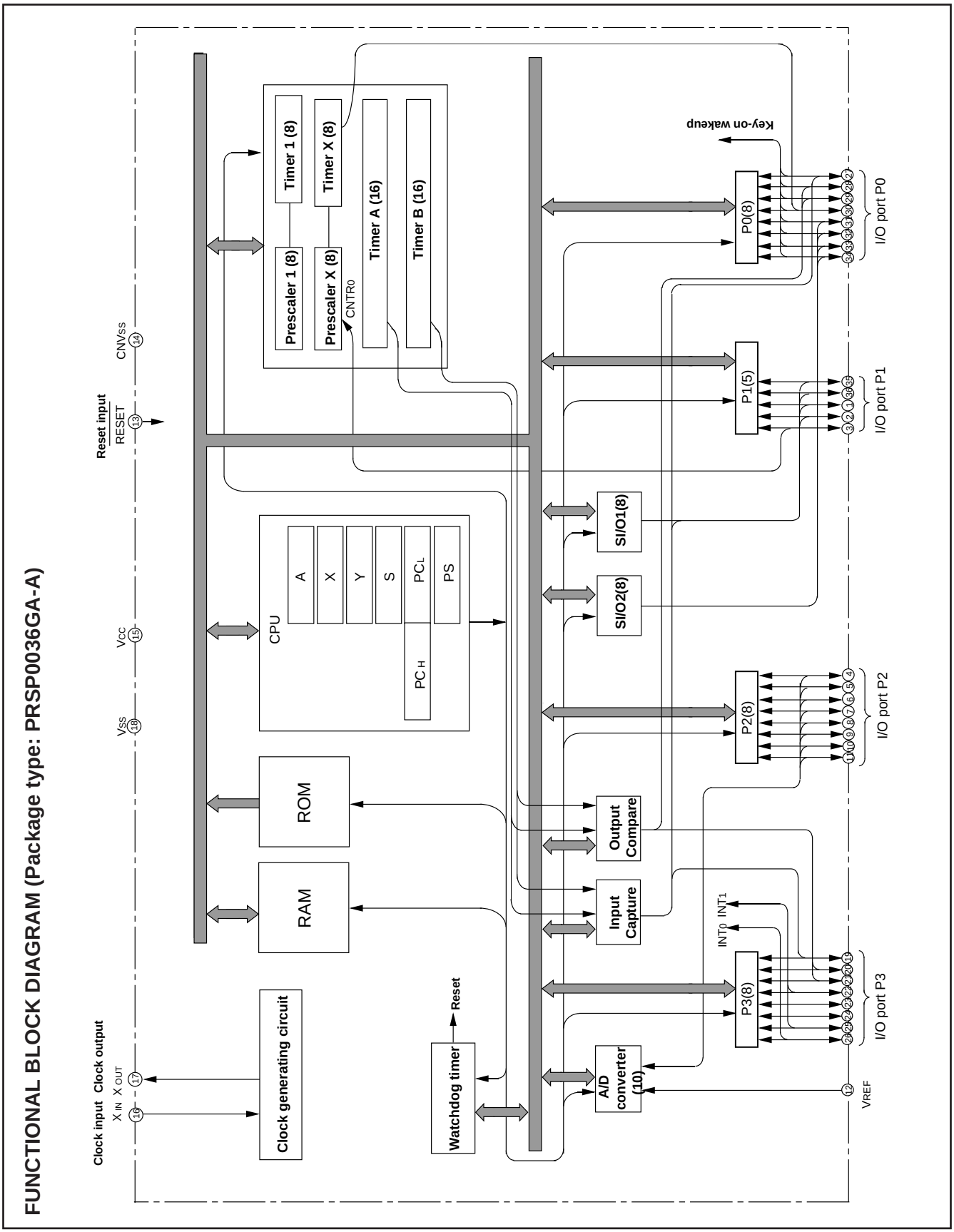



**Fig. 7 Functional block diagram (Package type: PRSP0036GA-A)**

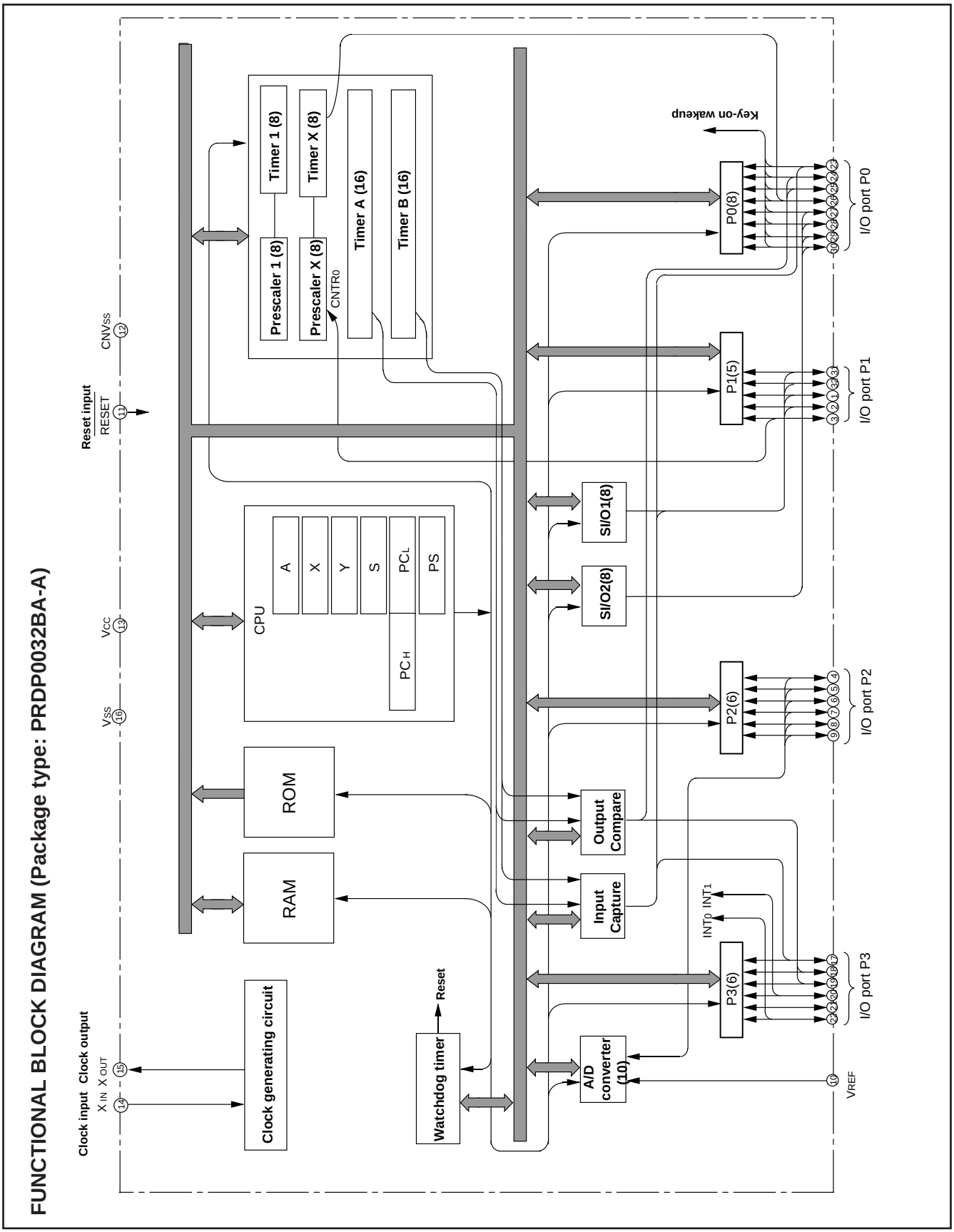



**Fig. 8 Functional block diagram (Package type: PRDP0032BA-A)**

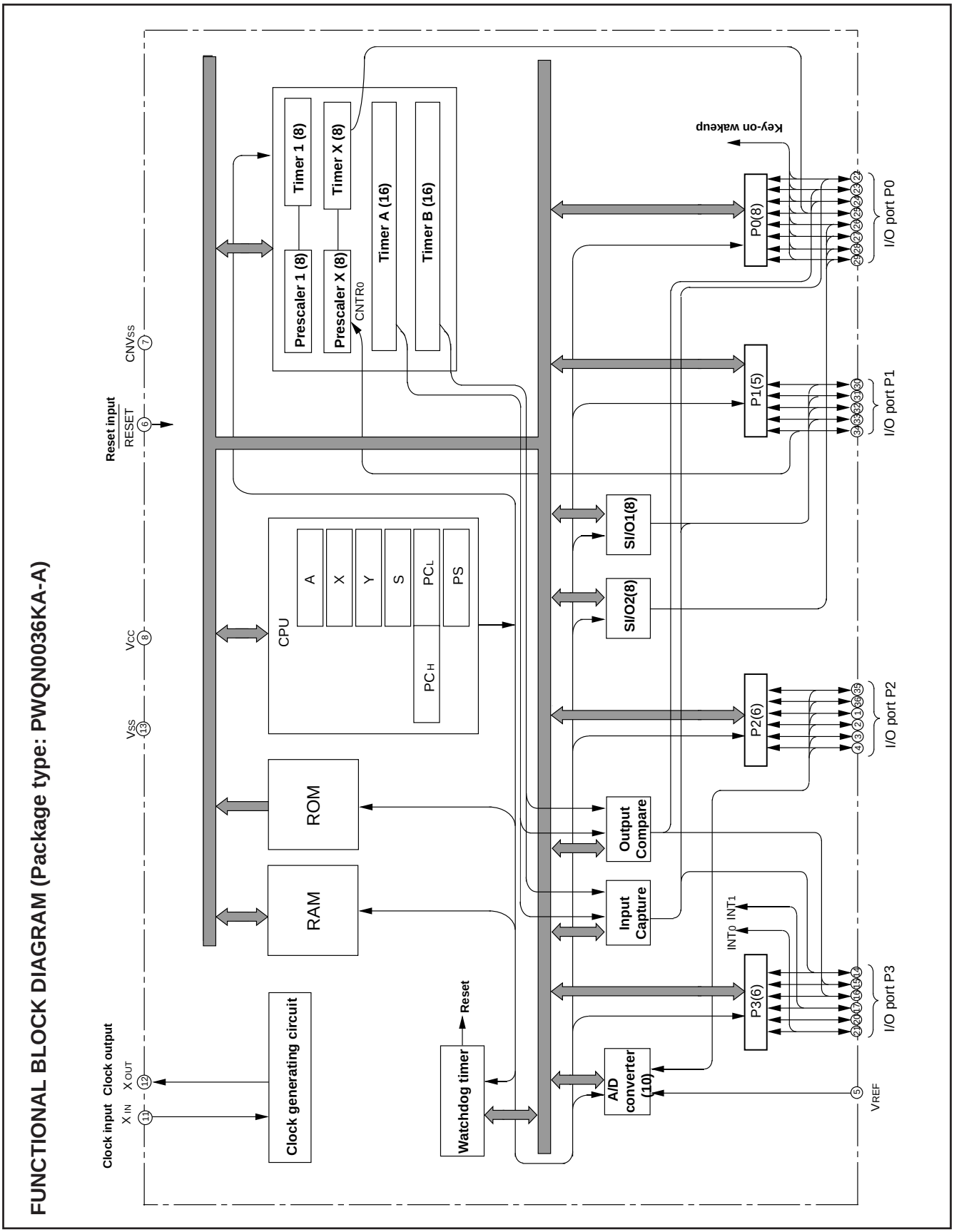



**Fig. 9 Functional block diagram (Package type: PWQN0036KA-A)**

## PIN DESCRIPTION

**Table 2 Pin description**

| Pin | Name | Function | | Function expect a port function | |
|---|---|---|---|---|---|
| Vcc, Vss | Power source | Mask ROM version | Apply voltage of 2.2 to 5.5 V to Vcc, and 0 V to Vss. | | |
| | | FLASH ROM version | Apply voltage of 2.7 to 5.5 V to Vcc, and 0 V to Vss. | | |
| VREF | Analog reference voltage | •Reference voltage input pin for A/D converter. | | | |
| 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.<br>•Connect a ceramic resonator or quartz crystal oscillator between the XIN and XOUT pins.<br>•For using RC oscillator, short between the XIN and XOUT pins, and connect the capacitor and resistor.<br>•If an external clock is used, connect the clock source to the XIN pin and leave the XOUT pin open.<br>•When the on-chip oscillator is selected as the main clock, connect XIN pin to VCC and leave XOUT open. | | | |
| XOUT | Clock output | | | | |
| P00(LED00)/CAP0 | I/O port P0 | •8-bit I/O port.<br>•I/O direction register allows each pin to be individually programmed as either input or output.<br>•CMOS compatible input level<br>•CMOS 3-state output structure<br>•Whether a built-in pull-up resistor is to be used or not can be determined by program.<br>• High drive capacity for LED drive port can be selected by program. | | • Capture function pin<br>• Compare function pin<br>• Timer X function pin<br>• Serial I/O2 function pin | • Key-input (key-on wake up interrupt input) pin |
| P01(LED01)/CMP0 | | | | | |
| P02(LED02)/CMP1 | | | | | |
| P03(LED03)/TXOUT | | | | | |
| P04(LED04)/RxD2 | | | | | |
| P05(LED05)/TxD2 | | | | | |
| P06(LED06)/SCLK2 | | | | | |
| P07(LED07)/$\overline{\text{SRDY2}}$ | | | | | |
| P10/RxD1/CAP0 | I/O port P1 | •5-bit I/O port<br>•I/O direction register allows each pin to be individually programmed as either input or output.<br>•CMOS compatible input level<br>•CMOS 3-state output structure<br>•CMOS/TTL level can be switched for P10, P12 and P13 | | • Serial I/O1 function pin<br>• Capture function pin | |
| P11/TxD1 | | | | • Serial I/O1 function pin | |
| P12/SCLK1 | | | | | |
| P13/$\overline{\text{SRDY1}}$ | | | | | |
| P14/CNTR0 | | | | • Timer X function pin | |
| P20/AN0–P27/AN7 | I/O port P2 (Note 1) | •8-bit I/O port having almost the same function as P0<br>•CMOS compatible input level<br>•CMOS 3-state output structure | | • Input pins for A/D converter | |
| P30(LED10)/CAP1 | I/O port P3 (Note 2) | •8-bit I/O port<br>•I/O direction register allows each pin to be individually programmed as either input or output.<br>•CMOS compatible input level (CMOS/TTL level can be switched for P36 and P37).<br>•CMOS 3-state output structure<br>•Whether a built-in pull-up resistor is to be used or not can be determined by program.<br>• High drive capacity for LED drive port can be selected by program. | | • Capture function pin | |
| P31(LED11)/CMP2 | | | | • Compare function pin | |
| P32(LED12)/CMP3 | | | | | |
| P33(LED13)/INT1 | | | | • Interrupt input pin | |
| P34(LED14) | | | | | |
| P35(LED15) | | | | | |
| P36(LED16)/INT1 | | | | • Interrupt input pin | |
| P37(LED17)/INT0 | | | | | |

Notes 1: P26/AN6 and P27/AN7 do not exist for the 32-pin version and PWQN0036KA-A package, so that Port P2 is a 6-bit I/O port.
2: P35 and P36/INT1 do not exist for the 32-pin version and PWQN0036KA-A package, so that Port P3 is a 6-bit I/O port.

## GROUP EXPANSION

Renesas plans to expand the 7542 group as follow:

**Memory type**

Support for Mask ROM version, Flash memory version, and Emulator MCU .

**Memory size**

Flash memory size ..... 16 to 32 K + 4 K bytes
Mask ROM size ..... 8 K to 16 K bytes
RAM size ..... 384 to 1024 bytes

**Package**

PRDP0032BA-A ..... 32-pin plastic molded SDIP
PLQP0032GB-A ..... 0.8 mm-pitch 32-pin plastic molded LQFP
PRSP0036GA-A ..... 0.8 mm-pitch 36-pin plastic molded SSOP
PWQN0036KA-A ..... 0.5 mm-pitch 36-pin plastic molded WQFN
42S1M ..... 42-pin shrink ceramic PIGGY BACK

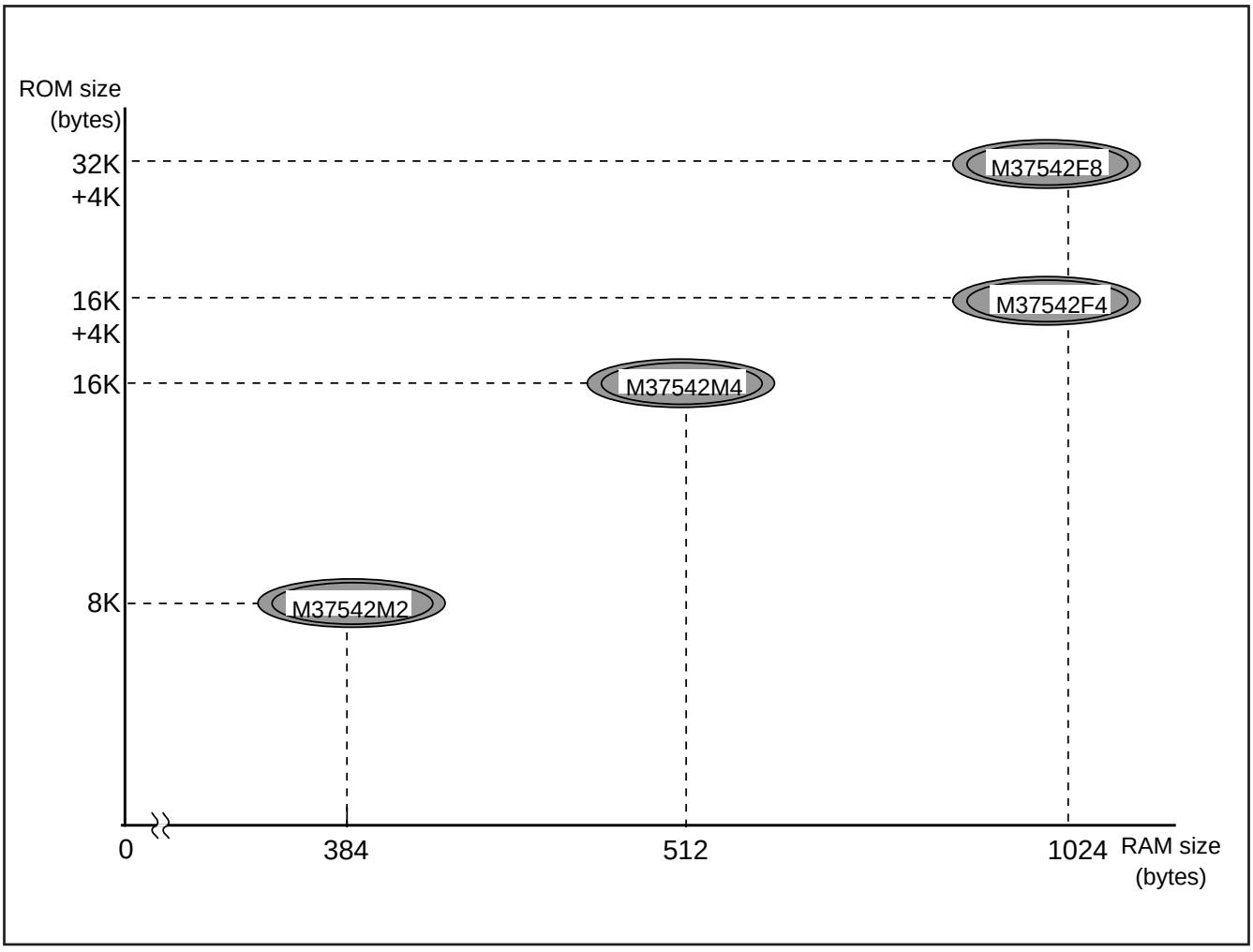



**Fig. 10 Memory expansion plan**

Currently supported products are listed below.

**Table 3 List of supported products**

| Product | ROM size (bytes)<br>ROM size for User ( ) | RAM size (bytes) | Package | Remarks |
|---|---|---|---|---|
| M37542M2-XXXSP | 8192<br>(8062) | 384 | PRDP0032BA-A | Mask ROM version |
| M37542M2-XXXHP | | | PWQN0036KA-A | Mask ROM version |
| M37542M2-XXXFP | | | PRSP0036GA-A | Mask ROM version |
| M37542M2-XXXGP | | | PLQP0032GB-A | Mask ROM version |
| M37542M4-XXXSP | 16384<br>(16254) | 512 | PRDP0032BA-A | Mask ROM version |
| M37542M4-XXXHP | | | PWQN0036KA-A | Mask ROM version |
| M37542M4-XXXFP | | | PRSP0036GA-A | Mask ROM version |
| M37542M4-XXXGP | | | PLQP0032GB-A | Mask ROM version |
| M37542F4SP | 16384 + 4096<br>(Note 2) | 1024 | PRDP0032BA-A | Flash memory version |
| M37542F4FP | | | PRSP0036GA-A | Flash memory version |
| M37542F4GP | | | PLQP0032GB-A | Flash memory version |
| M37542F8SP | 32768 + 4096<br>(Note 2) | 1024 | PRDP0032BA-A | Flash memory version |
| M37542F8FP | | | PRSP0036GA-A | Flash memory version |
| M37542F8GP | | | PLQP0032GB-A | Flash memory version |
| M37542F8HP<br>(Note 1) | | | PWQN0036KA-A | Flash memory version |
| M37542RSS | ———— | 1024 | 42S1M | Emulator MCU |

Notes 1: Only ES version (MP: no plan)
2: ROM size includes the ID code area.

## FUNCTIONAL DESCRIPTION

### Central Processing Unit (CPU)

The MCU uses the standard 740 family instruction set. Refer to the table of 740 family addressing modes and machine-language instructions or the SERIES 740 <SOFTWARE> USER'S MANUAL for details on each instruction set.

Machine-resident 740 family instructions are as follows:

1. The FST and SLW instructions cannot be used.
2. The MUL and DIV instructions can be used.
3. The WIT instruction can be used.
4. The STP instruction can be used.

### Accumulator (A)

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

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

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

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

### Stack pointer (S)

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

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

The Stack Page Selection Bit is located in the SFR area in the zero page. Note that the initial value of the Stack Page Selection Bit varies with each microcomputer type. Also some microcomputer types have no Stack Page Selection Bit and the upper eight bits of the stack address are fixed. The operations of pushing register contents onto the stack and popping them from the stack are shown in Fig. 12.

### Program counter (PC)

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

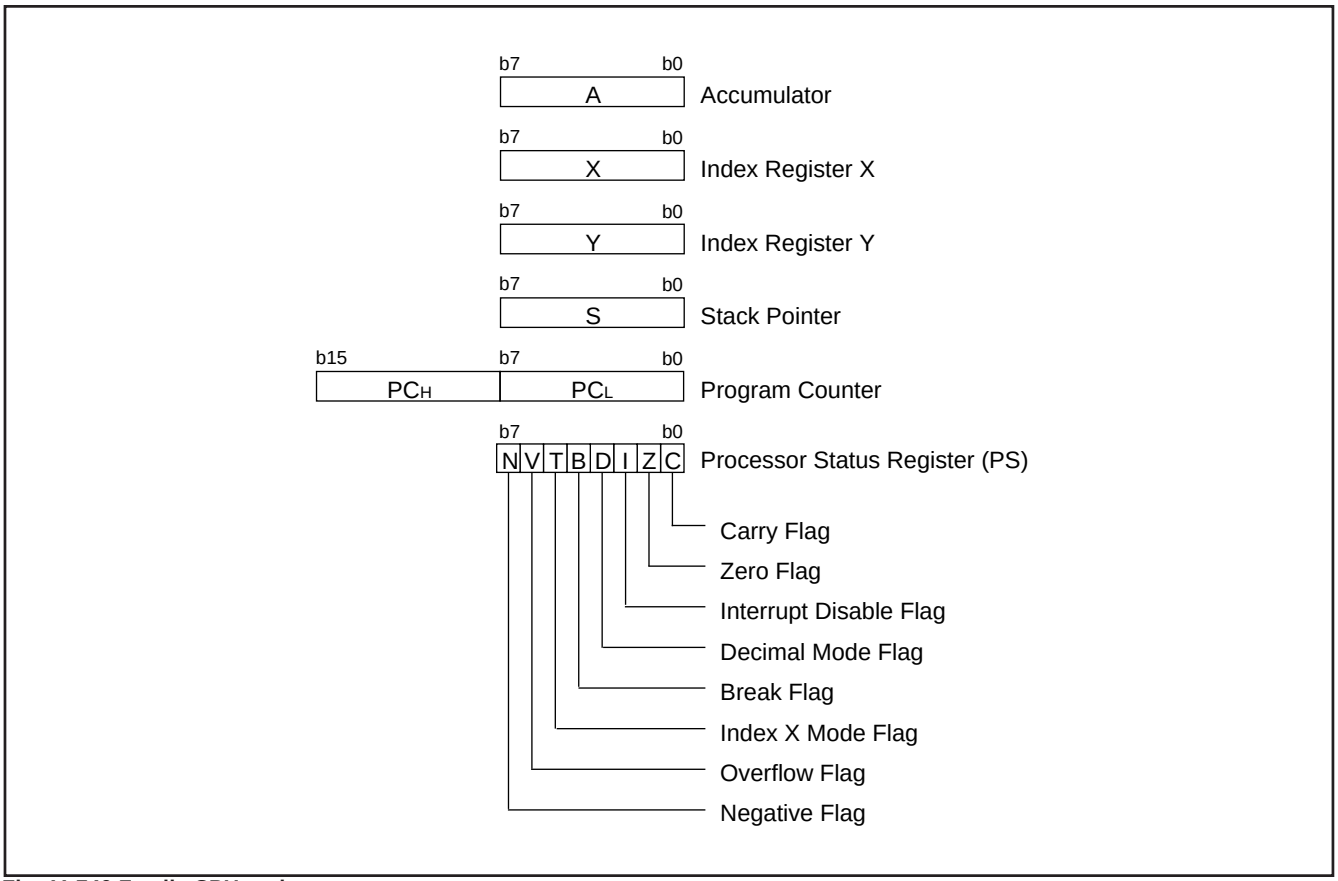



**Fig. 11 740 Family CPU register structure**

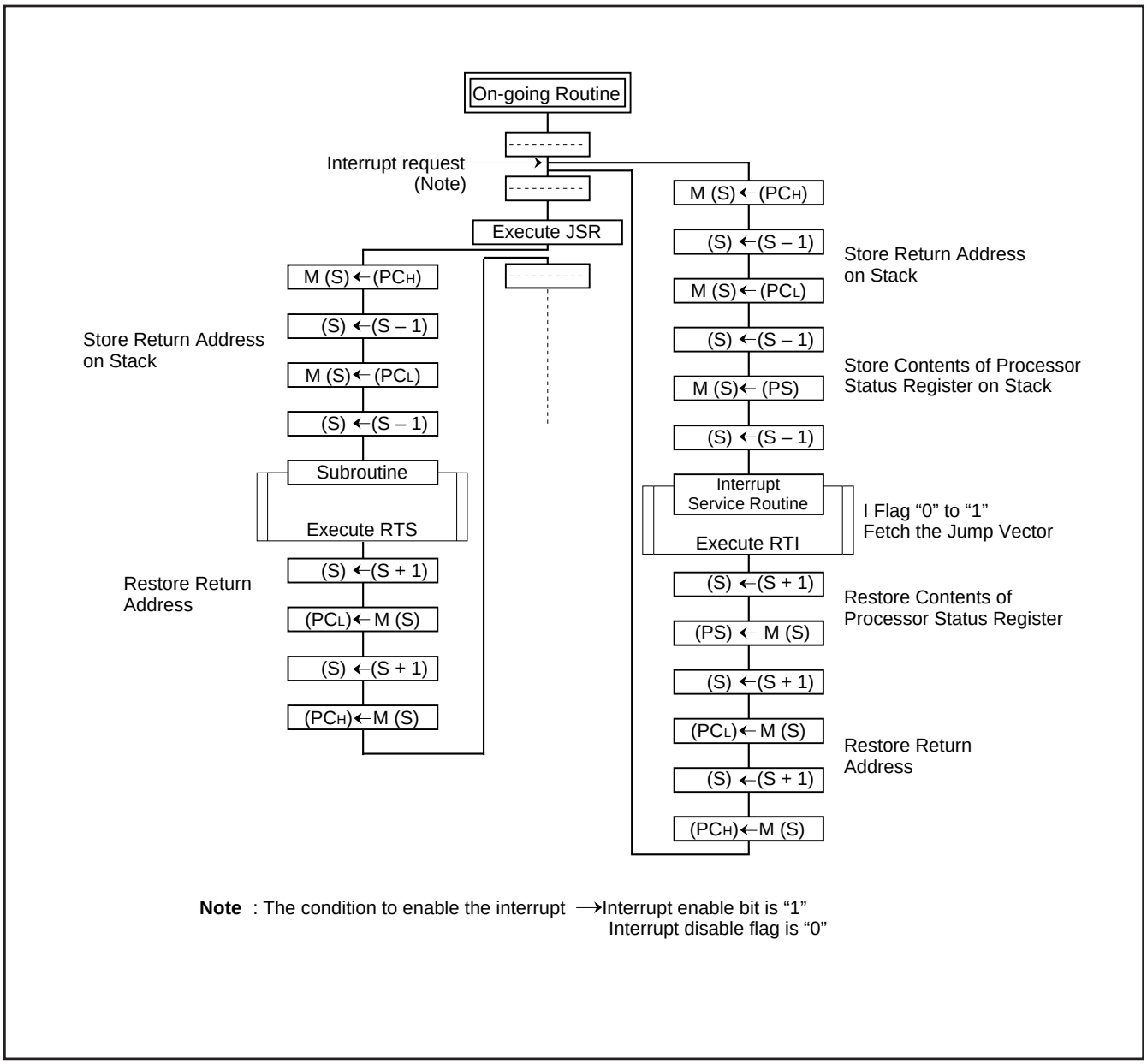



**Fig. 12 Register push and pop at interrupt generation and subroutine call**

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

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

## Processor status register (PS)

The processor status register is an 8-bit register consisting of flags which indicate the status of the processor after an arithmetic operation. Branch operations can be performed by testing the Carry (C) flag, Zero (Z) flag, Overflow (V) flag, or the Negative (N) flag. In decimal mode, the Z, V, N flags are not valid.
After reset, the Interrupt disable (I) flag is set to “1”, but all other flags are undefined. Since the Index X mode (T) and Decimal mode (D) flags directly affect arithmetic operations, they should be initialized in the beginning of a program.

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

**(2) Zero flag (Z)**
The Z flag is set if the result of an immediate arithmetic operation or a data transfer is “0”, and cleared if the result is anything other than “0”.

**(3) Interrupt disable flag (I)**
The I flag disables all interrupts except for the interrupt generated by the BRK instruction. Interrupts are disabled when the I flag is “1”.
When an interrupt occurs, this flag is automatically set to “1” to prevent other interrupts from interfering until the current interrupt is serviced.

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

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

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

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

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

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

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

**[CPU mode register] CPUM**

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

**Switching method of CPU mode register**

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

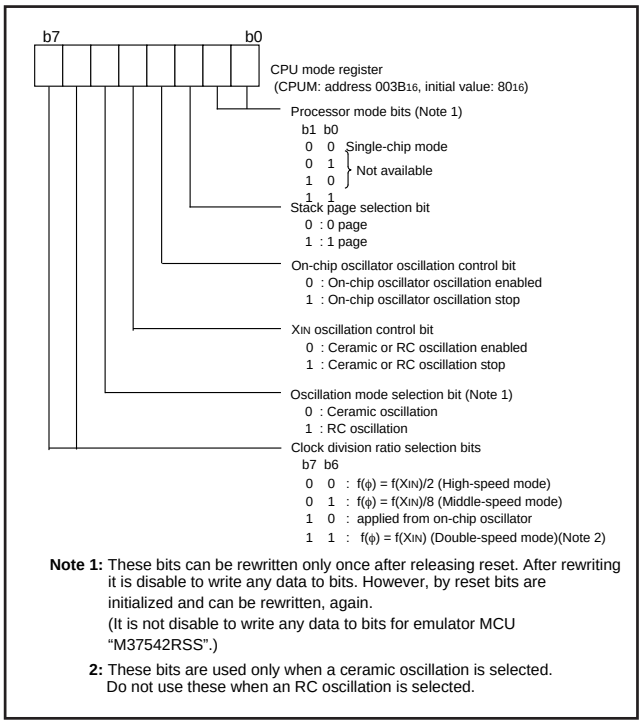



**Fig. 13 Structure of CPU mode register**

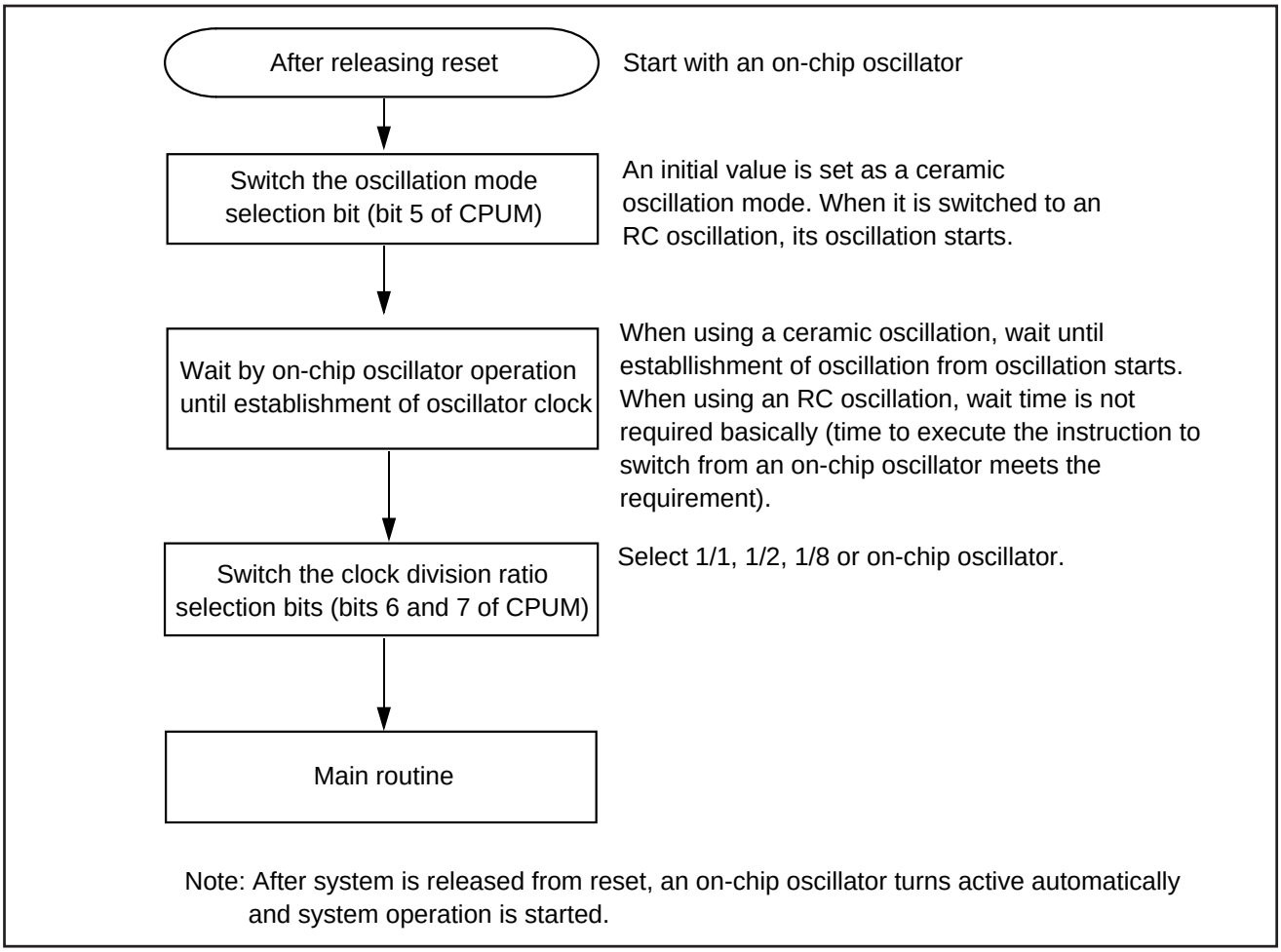



**Fig. 14 Switching method of CPU mode register**

## 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. The reserved ROM area can program/erase in the flash memory version.

**Interrupt vector area**

The interrupt vector area contains reset and interrupt vectors.

**Zero page**

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

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

**Special page**

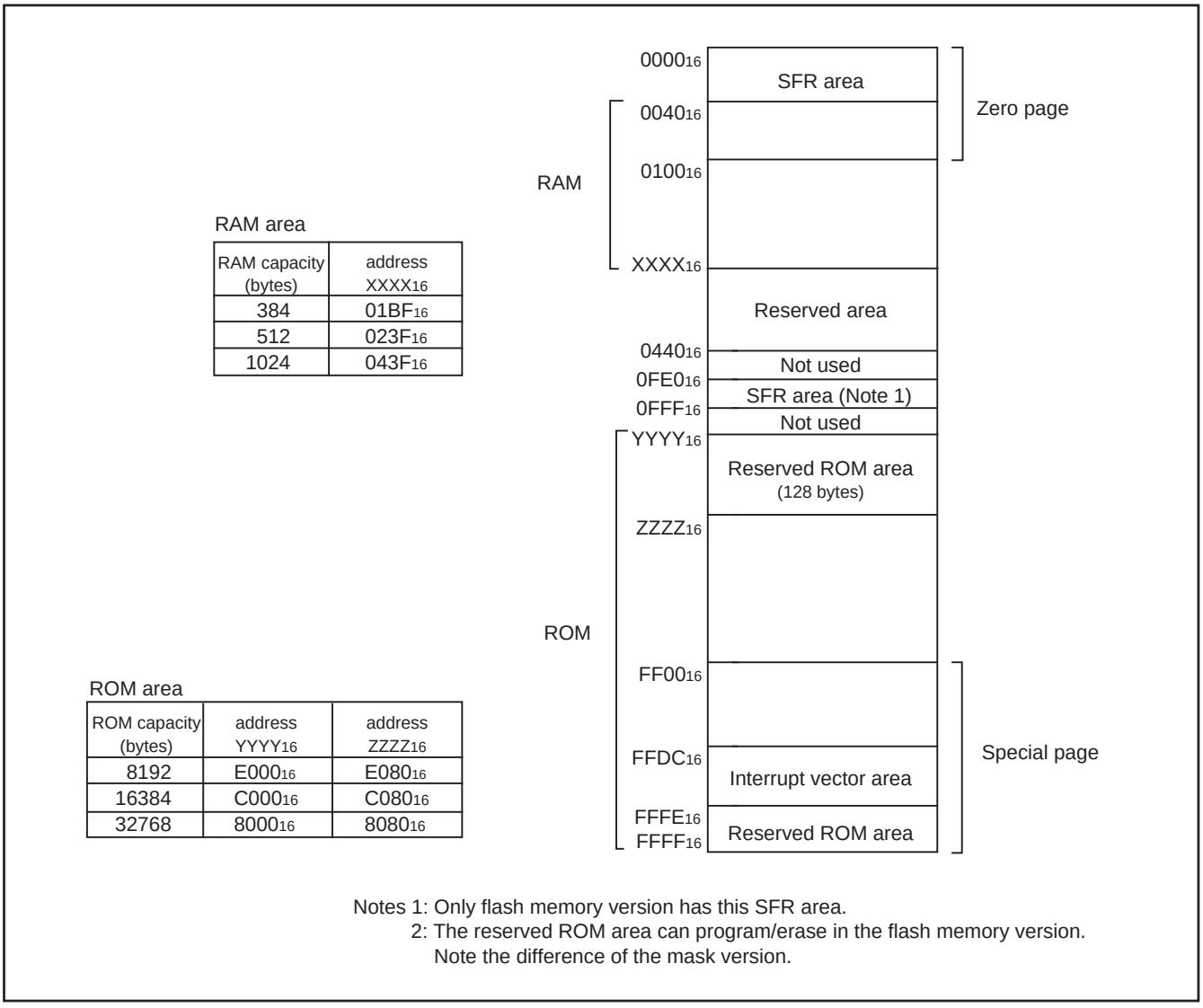
The 256 bytes from addresses $FF00_{16}$ to $FFFF_{16}$ are called the special page area. The special page addressing mode can be used to specify memory addresses in the special page area. Access to this area with only 2 bytes is possible in the special page addressing mode.

RAM area

| RAM capacity (bytes) | address XXXX16 |
|---|---|
| 384 | $01BF_{16}$ |
| 512 | $023F_{16}$ |
| 1024 | $043F_{16}$ |

ROM area

| ROM capacity (bytes) | address YYYY16 | address ZZZZ16 |
|---|---|---|
| 8192 | $E000_{16}$ | $E080_{16}$ |
| 16384 | $C000_{16}$ | $C080_{16}$ |
| 32768 | $8000_{16}$ | $8080_{16}$ |




Notes 1: Only flash memory version has this SFR area.
2: The reserved ROM area can program/erase in the flash memory version. Note the difference of the mask version.

**Fig. 15 Memory map diagram**

| | |
|---|---|
| 0000₁₆ | Port P0 (P0) |
| 0001₁₆ | Port P0 direction register (P0D) |
| 0002₁₆ | Port P1 (P1) |
| 0003₁₆ | Port P1 direction register (P1D) |
| 0004₁₆ | Port P2 (P2) |
| 0005₁₆ | Port P2 direction register (P2D) |
| 0006₁₆ | Port P3 (P3) |
| 0007₁₆ | Port P3 direction register (P3D) |
| 0008₁₆ | Reserved |
| 0009₁₆ | Reserved |
| 000A₁₆ | Interrupt source set register (INTSET) |
| 000B₁₆ | Interrupt source discrimination register (INTDIS) |
| 000C₁₆ | Capture register 0 (low-order) (CAP0L) |
| 000D₁₆ | Capture register 0 (high-order) (CAP0H) |
| 000E₁₆ | Capture register 1 (low-order) (CAP1L) |
| 000F₁₆ | Capture register 1 (high-order) (CAP1H) |
| 0010₁₆ | Compare register (low-order) (CMPL) |
| 0011₁₆ | Compare register (high-order) (CMPH) |
| 0012₁₆ | Capture/compare register R/W pointer (CCRP) |
| 0013₁₆ | Capture software trigger register (CSTR) |
| 0014₁₆ | Compare register re-load register (CMPR) |
| 0015₁₆ | Port P0P3 drive capacity control register (DCCR) |
| 0016₁₆ | Pull-up control register (PULL) |
| 0017₁₆ | Port P1P3 control register (P1P3C) |
| 0018₁₆ | Transmit 1 /Receive 1 buffer register (TB1/RB1) |
| 0019₁₆ | Serial I/O1 status register (SIO1STS) |
| 001A₁₆ | Serial I/O1 control register (SIO1CON) |
| 001B₁₆ | UART1 control register (UART1CON) |
| 001C₁₆ | Baud rate generator 1 (BRG1) |
| 001D₁₆ | Timer A, B mode register (TABM) |
| 001E₁₆ | Capture/compare port register (CCPR) |
| 001F₁₆ | Timer source selection register (TMSR) |

| | |
|---|---|
| 0020₁₆ | Capture mode register (CAPM) |
| 0021₁₆ | Compare output mode register (CMOM) |
| 0022₁₆ | Capture/compare status register (CCSR) |
| 0023₁₆ | Compare interrupt source set register (CISR) |
| 0024₁₆ | Timer A (low-order) (TAL) |
| 0025₁₆ | Timer A (high-order) (TAH) |
| 0026₁₆ | Timer B (low-order) (TBL) |
| 0027₁₆ | Timer B (high-order) (TBH) |
| 0028₁₆ | Prescaler 1 (PRE1) |
| 0029₁₆ | Timer 1 (T1) |
| 002A₁₆ | Timer count source set register (TCSS) |
| 002B₁₆ | Timer X mode register (TXM) |
| 002C₁₆ | Prescaler X (PREX) |
| 002D₁₆ | Timer X (TX) |
| 002E₁₆ | Transmit 2 / Receive 2 buffer register (TB2/RB2) |
| 002F₁₆ | Serial I/O2 status register (SIO2STS) |
| 0030₁₆ | Serial I/O2 control register (SIO2CON) |
| 0031₁₆ | UART2 control register (UART2CON) |
| 0032₁₆ | Baud rate generator 2 (BRG2) |
| 0033₁₆ | Reserved |
| 0034₁₆ | A/D control register (ADCON) |
| 0035₁₆ | A/D conversion register (low-order) (ADL) |
| 0036₁₆ | A/D conversion register (high-order) (ADH) |
| 0037₁₆ | On-chip oscillation division ratio selection register (RODR) |
| 0038₁₆ | MISRG |
| 0039₁₆ | Watchdog timer control register (WDTCON) |
| 003A₁₆ | Interrupt edge selection register (INTEDGE) |
| 003B₁₆ | CPU mode register (CPUM) |
| 003C₁₆ | Interrupt request register 1 (IREQ1) |
| 003D₁₆ | Interrupt request register 2 (IREQ2) |
| 003E₁₆ | Interrupt control register 1 (ICON1) |
| 003F₁₆ | Interrupt control register 2 (ICON2) |

| | |
|---|---|
| 0FE0₁₆ | Flash memory control register 0 (FMCR0) (Note 2) |
| 0FE1₁₆ | Flash memory control register 1 (FMCR1) (Note 2) |
| 0FE2₁₆ | Flash memory control register 2 (FMCR2) (Note 2) |

**Notes 1:** Do not access to the SFR area including nothing.
**2:** Only flash memory version has this SFR area.

**Fig. 16 Memory map of special function register (SFR)**

## I/O Ports

**[Direction registers] PiD**

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

When "1" is set to the bit corresponding to a pin, this pin becomes an output port. When "0" is set to the bit, the pin becomes an input port.

When data is read from a pin set to output, not the value of the pin itself but the value of port latch is read. Pins set to input are floating, and permit reading pin values.

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

**Note:** $P2_6$/$AN_6$, $P2_7$/$AN_7$, $P3_5$ and $P3_6$ do not exist for the 32-pin version and PWQN0036KA-A package.
Accordingly, the following settings are required;
- Select $P3_3$ for the $INT_1$ function.
- Set direction registers of ports $P2_6$ and $P2_7$ to output.
- Set direction registers of ports $P3_5$ and $P3_6$ to output.

**[Port P0P3 drive capacity control register] DCCR**

By setting the Port P0P3 drive capacity control register (address $0015_{16}$), the drive capacity of the N-channel output transistor for the port P0 and port P3 can be selected.

**[Pull-up control register] 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 register] 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.

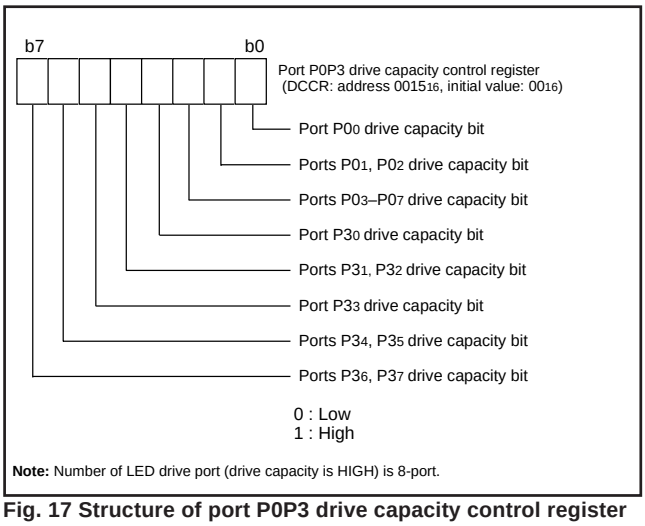



**Fig. 17 Structure of port P0P3 drive capacity control register**

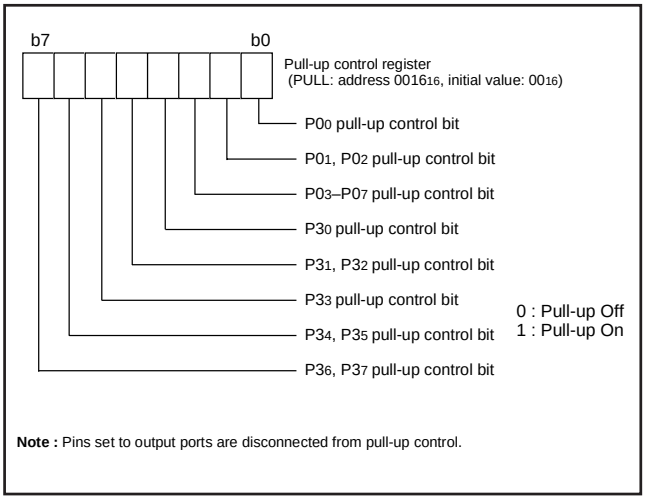



**Fig. 18 Structure of pull-up control register**

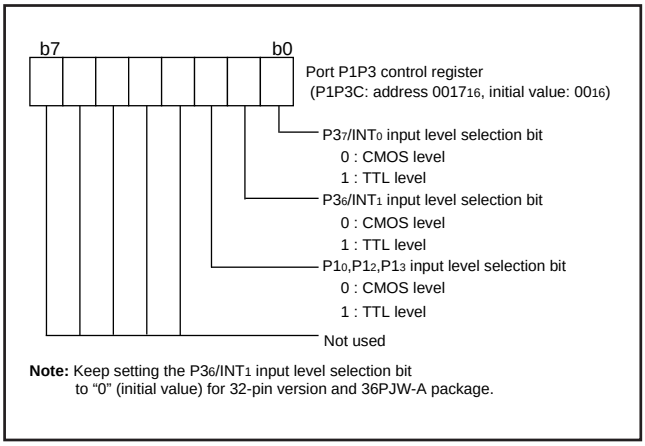



**Fig. 19 Structure of port P1P3 control register**

**Table 6 I/O port function table**

| Pin | Name | I/O format | Non-port function | SFRs related each pin | Diagram No. |
|---|---|---|---|---|---|
| P0$_0$(LED$_{00}$)/CAP$_0$ | I/O port P0 | •CMOS compatible input level (**Note 1**)<br>•CMOS 3-state output | • Capture function input<br>• Key input interrupt | Capture/Compare port register<br>Interrupt edge selection register<br>Pull-up control register<br>Port P0P3 drive capacity control register | (1) |
| P0$_1$(LED$_{01}$)/CMP$_0$<br>P0$_2$(LED$_{02}$)/CMP$_1$ | | | • Compare function output<br>• Key input interrupt | Capture/Compare port register<br>Pull-up control register<br>Port P0P3 drive capacity control register | (2) |
| P0$_3$(LED$_{03}$)/TXOUT | | | • Timer X function output<br>• Key input interrupt | Timer X mode register<br>Pull-up control register<br>Port P0P3 drive capacity control register | (3) |
| P0$_4$(LED$_{04}$)/RxD$_2$ | | | • Serial I/O2 function input/output<br>• Key input interrupt | Serial I/O2 control register<br>Interrupt edge selection register<br>Pull-up control register<br>Port P0P3 drive capacity control register | (4) |
| P0$_5$(LED$_{05}$)/TxD$_2$ | | | | Serial I/O2 control register<br>Pull-up control register<br>Port P0P3 drive capacity control register | (5) |
| P0$_6$(LED$_{06}$)/SCLK2 | | | | Serial I/O2 control register<br>Interrupt edge selection register<br>Pull-up control register<br>Port P0P3 drive capacity control register | (6) |
| P0$_7$(LED$_{07}$)/$\overline{\text{SRDY2}}$ | | | | Serial I/O2 control register<br>Pull-up control register<br>Port P0P3 drive capacity control register | (7) |
| P1$_0$/RxD$_1$/CAP$_0$ | I/O port P1 | | • Serial I/O1 function input<br>• Capture function input | Serial I/O1 control register<br>Capture/Compare port register<br>Port P1P3 control register | (8) |
| P1$_1$/TxD$_1$ | | | • Serial I/O1 function input/output | Serial I/O1 control register | (9) |
| P1$_2$/SCLK1 | | | | Serial I/O1 control register<br>Port P1P3 control register | (10) |
| P1$_3$/$\overline{\text{SRDY1}}$ | | | | Serial I/O1 control register<br>Port P1P3 control register | (11) |
| P1$_4$/CNTR$_0$ | | | • Timer X function input/output<br>• External interrupt input | Timer X mode register | (12) |
| P2$_0$/AN$_0$–P2$_7$/AN$_7$ | I/O port P2 (**Note 2**) | | • A/D conversion input | A/D control register | (13) |
| P3$_0$(LED$_{10}$)/CAP$_1$ | I/O port P3 (**Note 3**) | | • Capture function input | Capture/Compare port register<br>Pull-up control register<br>Port P0P3 drive capacity control register | (14) |
| P3$_1$(LED$_{11}$)/CMP$_2$<br>P3$_2$(LED$_{12}$)/CMP$_3$ | | | • Compare function output | Capture/Compare port register<br>Pull-up control register<br>Port P0P3 drive capacity control register | (15) |
| P3$_3$(LED$_{13}$)/INT$_1$ | | | • External interrupt input | Interrupt edge selection register<br>Pull-up control register<br>Port P0P3 drive capacity control register | (16) |
| P3$_4$(LED$_{14}$)<br>P3$_5$(LED$_{15}$) | | | | Pull-up control register<br>Port P0P3 drive capacity control register | (17) |
| P3$_6$(LED$_{16}$)/INT$_1$<br>P3$_7$(LED$_{17}$)/INT$_0$ | | | • External interrupt input | Interrupt edge selection register<br>Pull-up control register<br>Port P0P3 drive capacity control register<br>Port P1P3 control register | (18)<br>(19) |

**Notes 1:** Ports P1$_0$, P1$_2$, P1$_3$, P3$_6$, and P3$_7$ are CMOS/TTL level.
**2:** P2$_6$/AN$_6$ and P2$_7$/AN$_7$ do not exist for the 32-pin version and PWQN0036KA-A package.
**3:** P3$_5$ and P3$_6$/INT$_1$ do not exist for the 32-pin version and PWQN0036KA-A package.

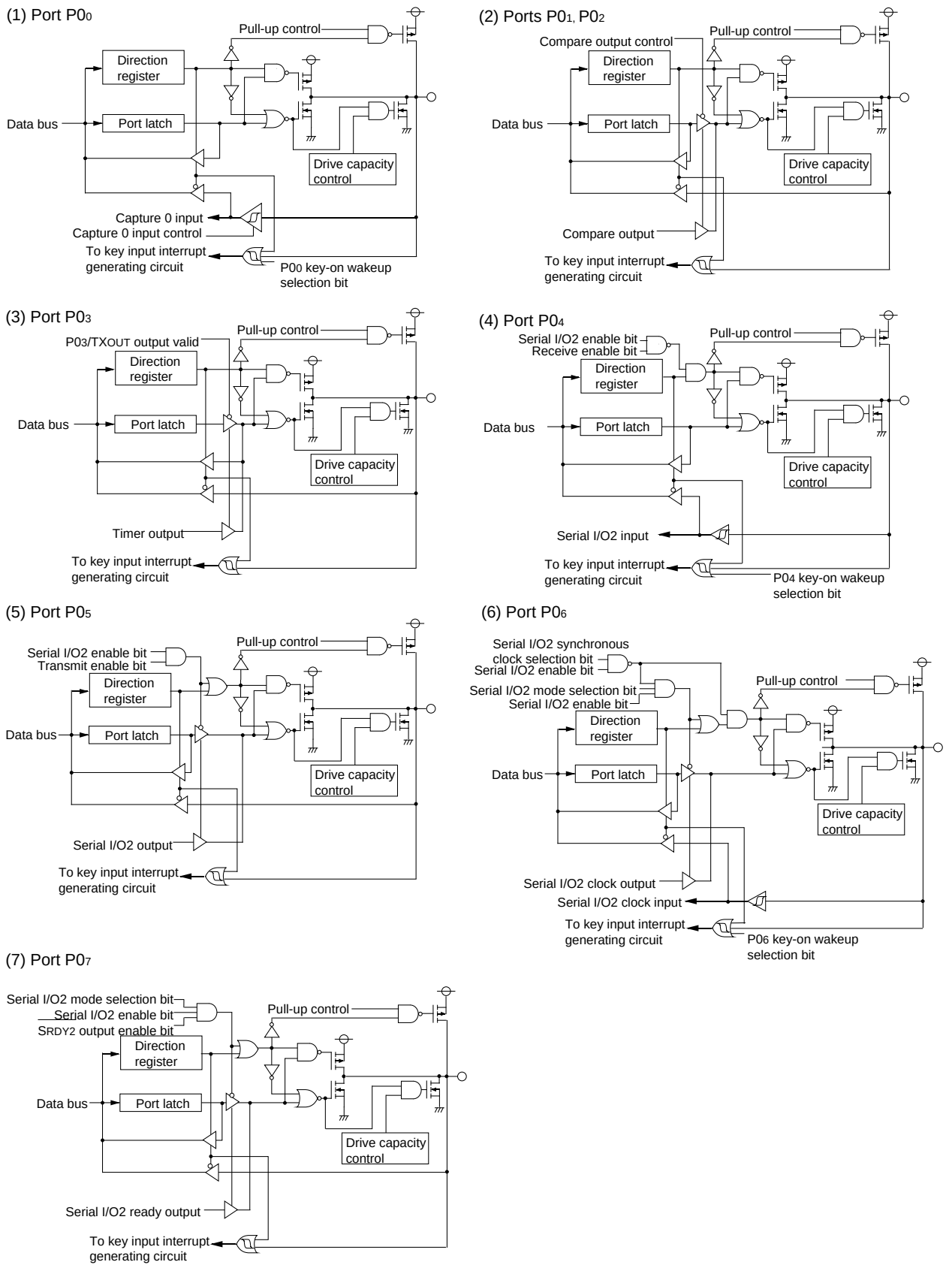
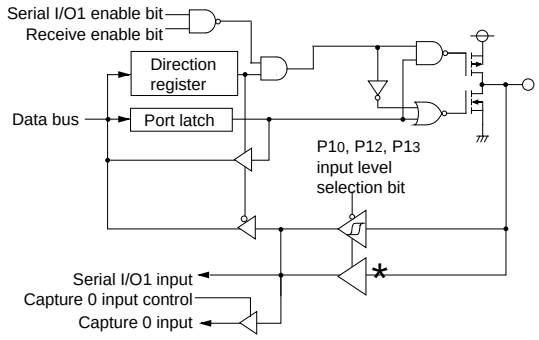
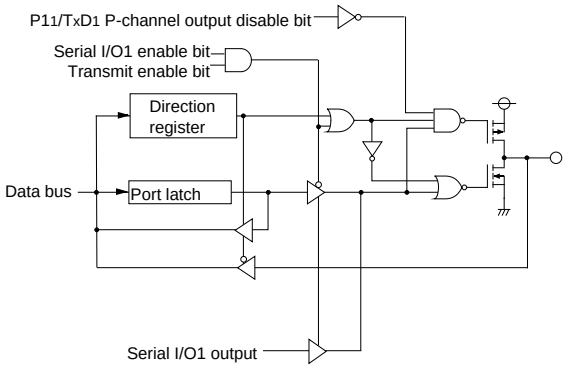



**Fig. 20 Block diagram of ports (1)**

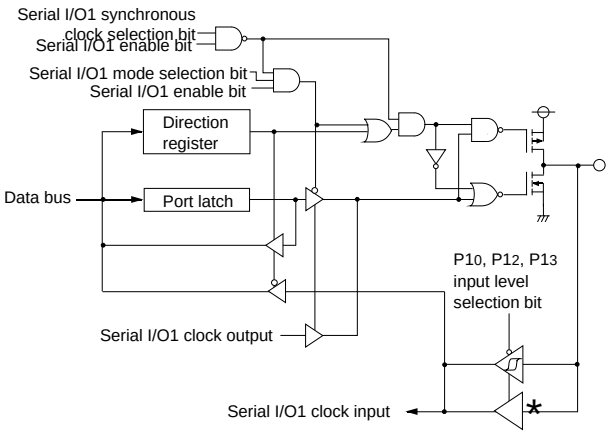
(8) Port P1$_0$



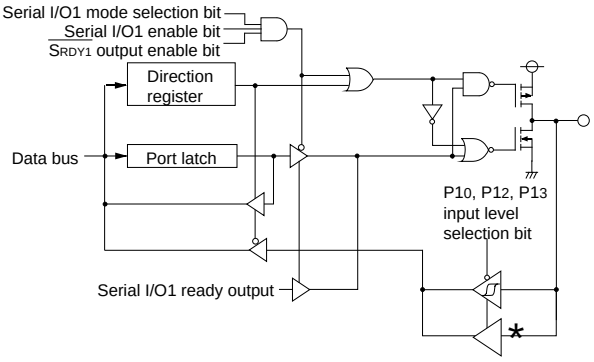

(9) Port P1$_1$



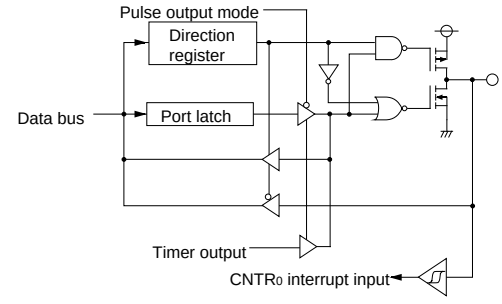

(10) Port P1$_2$



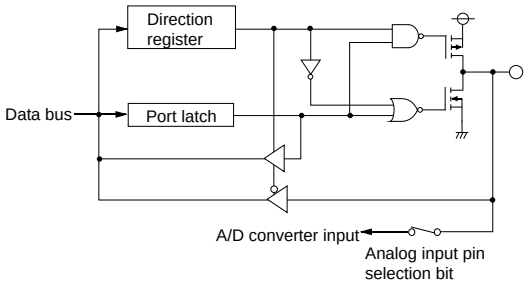

(11) Port P1$_3$




(12) Port P1$_4$

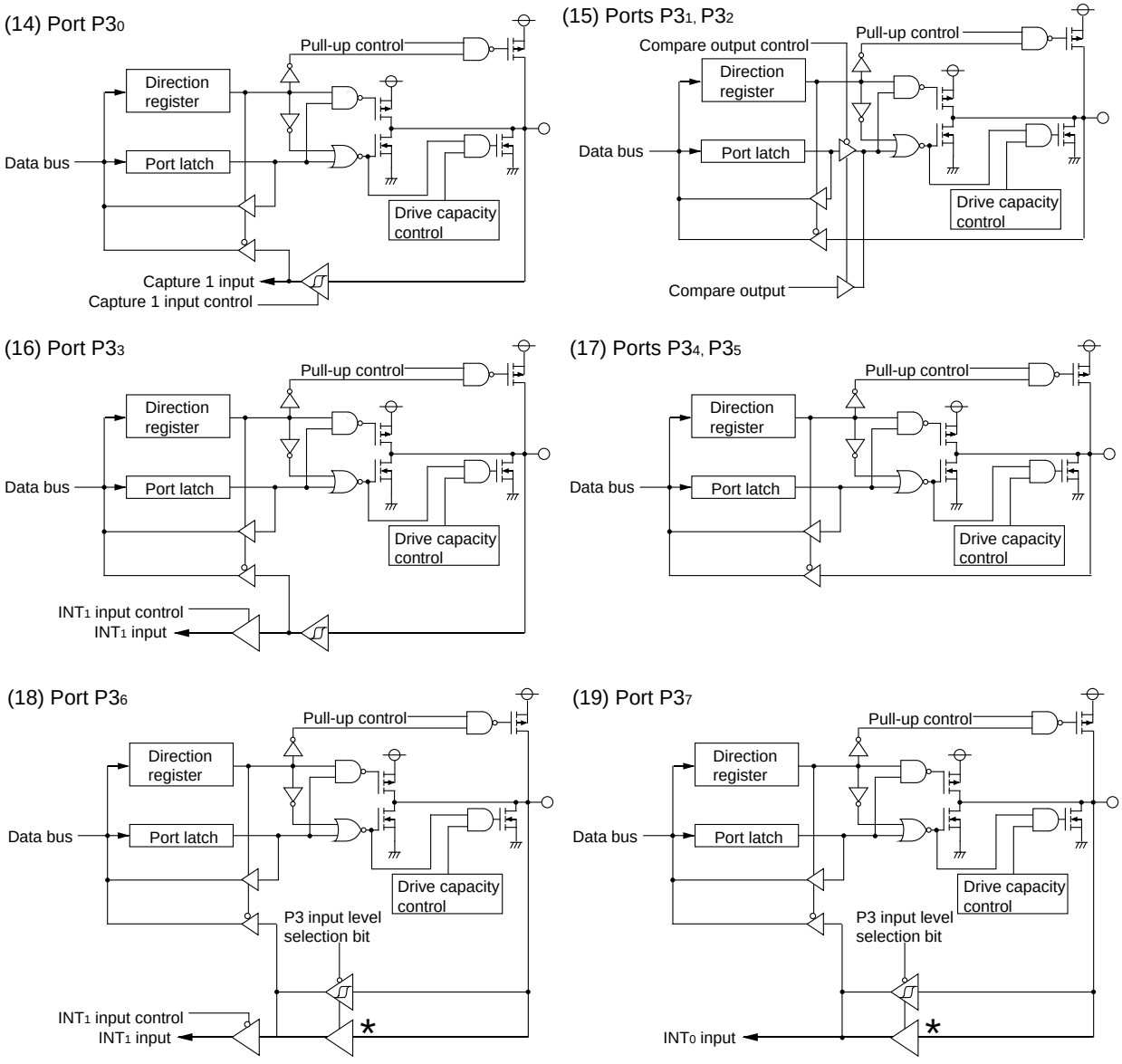



(13) Ports P2$_0$–P2$_7$




* P10, P12, P13, P36, and P37 input level 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. 21 Block diagram of ports (2)**




* P10, P12, P13, P36, and P37 input level 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. 22 Block diagram of ports (3)**

## Termination of unused pins

• Termination of common pins

I/O ports: Select an input port or an output port and follow each processing method.

Output ports: Open.

Input ports: If the input level become unstable, through current flow to an input circuit, and the power supply current may increase.

Especially, when expecting low consumption current (at STP or WIT instruction execution etc.), pull-up or pull-down input ports to prevent through current (built-in resistor can be used).

We recommend processing unused pins through a resistor which can secure $I_{OH(avg)}$ or $I_{OL(avg)}$.

Because, when an I/O port or a pin which have an output function is selected as an input port, it may operate as an output port by incorrect operation etc.

**Table 7 Termination of unused pins**

| Pin | Termination 1 (recommend) | Termination 2 | Termination 3 | Termination 4 |
|---|---|---|---|---|
| P00/CAP0 | I/O port | When selecting CAP function, perform termination of input port. | - | When selecting key-on wakeup function, perform termination of input port. |
| P01/CMP0 | | When selecting CMP0 function, perform termination of output port. | - | |
| P02/CMP1 | | When selecting CMP1 function, perform termination of output port. | - | |
| P03/TXOUT | | When selecting TXOUT function, perform termination of output port. | - | |
| P04/RxD2 | | When selecting RxD2 function, perform termination of input port. | - | |
| P05/TxD2 | | When selecting TxD2 function, perform termination of output port. | - | |
| P06/SCLK2 | | When selecting external clock input, perform termination of output port. | When selecting internal clock output, perform termination of output port. | |
| P07/$\overline{\text{SRDY2}}$ | | When selecting $\overline{\text{SRDY2}}$ function, perform termination of output port. | - | |
| P10/RxD1/CAP0 | | When selecting RxD1 function, perform termination of input port. | When selecting CAP function, perform termination of input port. | |
| P11/TxD1 | | When selecting TxD1 function, perform termination of output port. | - | - |
| P12/SCLK1 | | When selecting external clock input, perform termination of input port. | When selecting internal clock output, perform termination of output port. | - |
| P13/$\overline{\text{SRDY1}}$ | | When selecting $\overline{\text{SRDY1}}$ function, perform termination of output port. | - | - |
| P14/CNTR0 | | When selecting CNTR input function, perform termination of input port. | When selecting CNTR output function, perform termination of output port. | - |
| P20/AN0–P27/AN7 | | When selecting AN function, perform termination of input port. | - | - |
| P30/CAP1 | | When selecting CAP function, perform termination of input port. | - | - |
| P31/CMP2 | | When selecting CMP2 function, perform termination of output port. | - | - |
| P32/CMP3 | | When selecting CMP3 function, perform termination of output port. | - | - |
| P33/INT1 | | When selecting INT function, perform termination of input port. | - | - |
| P34 | | - | - | - |
| P35 | | - | - | - |
| P36/INT1 | | When selecting INT function, perform termination of input port. | - | - |
| P37/INT0 | | When selecting INT function, perform termination of input port. | - | - |
| VREF | Connect to Vss. | - | - | - |
| XIN | When only on-chip oscillator is used, connect to VCC through a resistor. | - | - | - |
| XOUT | When external clock is input or when on-chip oscillator is used, open. | - | - | - |

## Interrupts

The 7542 Group interrupts are vector interrupts with a fixed priority scheme, and generated by 16 sources among 18 sources: 6 external, 11 internal, and 1 software.
The interrupt sources, vector addresses(1) , and interrupt priority are shown in Table 8.

Each interrupt except the BRK instruction interrupt has the interrupt request bit and the interrupt enable bit. These bits and the interrupt disable flag (I flag) control the acceptance of interrupt requests. Figure 23 shows an interrupt control diagram.

An interrupt request is accepted when all of the following conditions are satisfied:

- Interrupt disable flag.....“0”
- Interrupt request bit.....“1”
- Interrupt enable bit.....“1”

Though the interrupt priority is determined by hardware, priority processing can be performed by software using the above bits and flag.

**Table 8 Interrupt vector addresses and priority**

| Interrupt source | Priority | Vector addresses (Note 1) | | Interrupt request generating conditions | Remarks |
|---|---|---|---|---|---|
| | | High-order | Low-order | | |
| Reset (Note 2) | 1 | FFFD16 | FFFC16 | At reset input | Non-maskable |
| Serial I/O1 receive | 2 | FFFB16 | FFFA16 | At completion of serial I/O1 data receive | Valid only when serial I/O1 is selected |
| Serial I/O1 transmit | 3 | FFF916 | FFF816 | At completion of serial I/O1 transmit shift or when transmit buffer is empty | Valid only when serial I/O1 is selected |
| Serial I/O2 receive | 4 | FFF716 | FFF616 | At completion of serial I/O2 data receive | Valid only when serial I/O2 is selected |
| Serial I/O2 transmit | 5 | FFF516 | FFF416 | At completion of serial I/O2 transmit shift or when transmit buffer is empty | Valid only when serial I/O2 is selected |
| INT0 | 6 | FFF316 | FFF216 | At detection of either rising or falling edge of INT0 input | External interrupt (active edge selectable) |
| INT1 | 7 | FFF116 | FFF016 | At detection of either rising or falling edge of INT1 input | External interrupt (active edge selectable) |
| Key-on wake-up/ UART1 bus collision detection (Note 3) | 8 | FFEF16 | FFEE16 | At falling of conjunction of input logical level for port P0 (at input) At detection of UART1 bus collision detection | External interrupt (valid at falling edge) When UART1 bus collision detection interrupt is enabled. |
| CNTR0 | 9 | FFED16 | FFEC16 | At detection of either rising or falling edge of CNTR0 input | External interrupt (active edge selectable) |
| Capture 0 | 10 | FFEB16 | FFEA16 | At detection of either rising or falling edge of Capture 0 input | External interrupt (active edge selectable) |
| Capture 1 | 11 | FFE916 | FFE816 | At detection of either rising or falling edge of Capture 1 input | External interrupt (active edge selectable) |
| Compare | 12 | FFE716 | FFE616 | At compare matched | Compare interrupt source is selected. |
| Timer X | 13 | FFE516 | FFE416 | At timer X underflow | |
| Timer A | 14 | FFE316 | FFE216 | At timer A underflow | |
| Timer B | 15 | FFE116 | FFE016 | At timer B underflow | |
| A/D conversion/ Timer 1 (Note 4) | 16 | FFDF16 | FFDE16 | At completion of A/D conversion At timer 1 underflow | STP release timer underflow |
| BRK instruction | 17 | FFDD16 | FFDC16 | At BRK instruction execution | Non-maskable software interrupt |

**Notes1:** Vector addresses contain internal jump destination addresses.
**2:** Reset function in the same way as an interrupt with the highest priority.
**3:** Key-on wakeup interrupt and UART1 bus collision detection interrupt can be enabled by setting of interrupt source set register. The occurrence of these interrupts are discriminated by interrupt source discrimination register.
**4:** A/D conversion interrupt and Timer 1 interrupt can be enabled by setting of interrupt source set register. The occurrence of these interrupts are discriminated by interrupt source discrimination register.

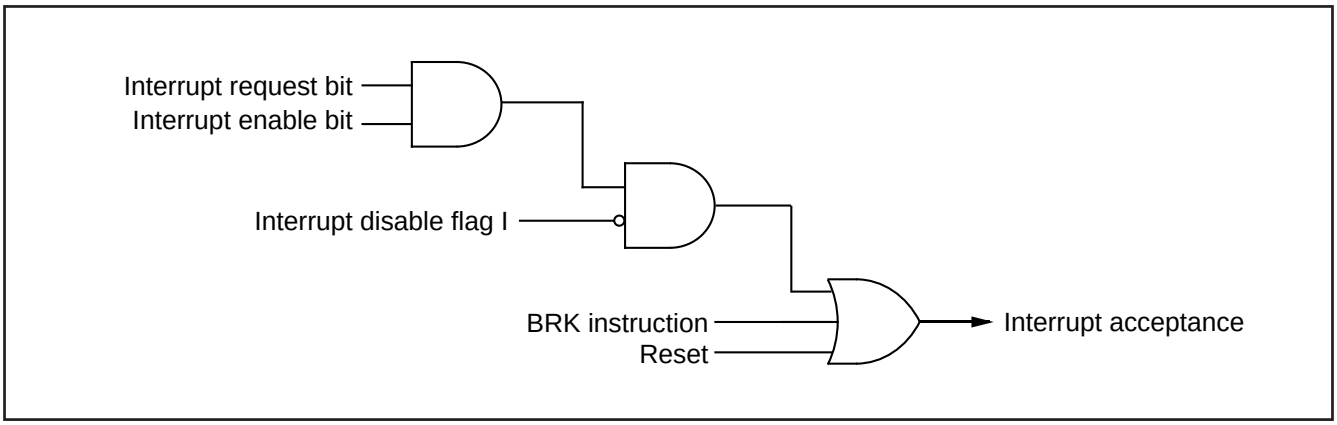



**Fig. 23 Interrupt control**

**• Interrupt Disable Flag**

The interrupt disable flag is assigned to bit 2 of the processor status register. This flag controls the acceptance of all interrupt requests except for the BRK instruction. When this flag is set to "1", the acceptance of interrupt requests is disabled. When it is set to "0", the acceptance of interrupt requests is enabled. This flag is set to "1" with the SEI instruction and set to "0" with the CLI instruction.

When an interrupt request is accepted, the contents of the processor status register are pushed onto the stack while the interrupt disable flag remains set to "0". Subsequently, this flag is automatically set to "1" and multiple interrupts are disabled.

To use multiple interrupts, set this flag to "0" with the CLI instruction within the interrupt processing routine.

The contents of the processor status register are popped off the stack with the RTI instruction.

**• Interrupt Request Bits**

Once an interrupt request is generated, the corresponding interrupt request bit is set to "1" and remains "1" until the request is accepted. When the request is accepted, this bit is automatically set to "0".

Each interrupt request bit can be set to "0", but cannot be set to "1", by software.

**• Interrupt Enable Bits**

The interrupt enable bits control the acceptance of the corresponding interrupt requests. When an interrupt enable bit is set to "0", the acceptance of the corresponding interrupt request is disabled. If an interrupt request occurs in this condition, the corresponding interrupt request bit is set to "1", but the interrupt request is not accepted. When an interrupt enable bit is set to "1", the acceptance of the corresponding interrupt request is enabled.

Each interrupt enable bit can be set to "0" or "1" by software.

The interrupt enable bit for an unused interrupt should be set to "0".

**• Interrupt Enable Setting**

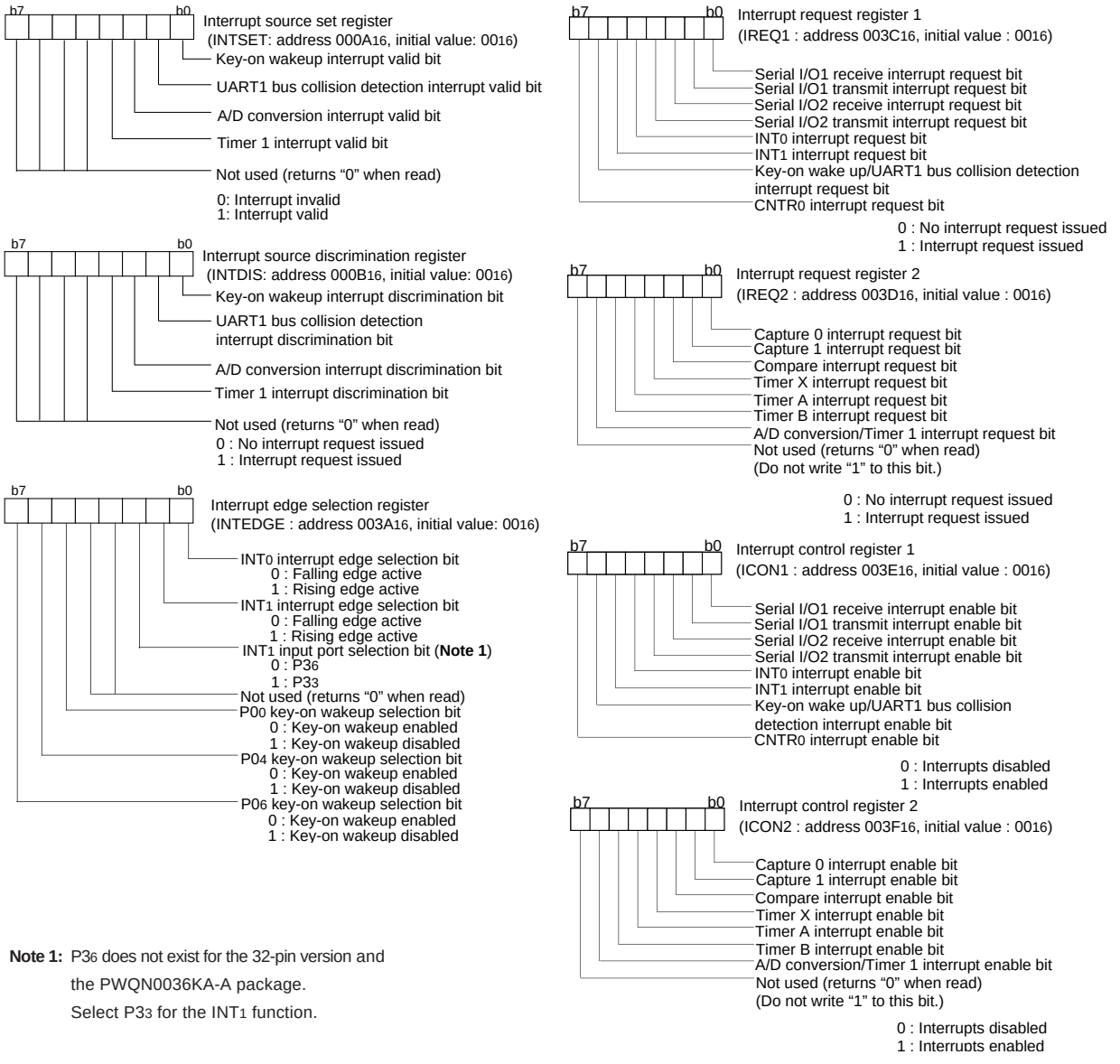
The following interrupt sources can be set to valid or invalid by the interrupt source set register ($000A_{16}$).

- Key-on wakeup
- UART1 bus collision detection interrupt
- A/D conversion
- Timer 1 interrupt

**• External Interrupt Pin Selection**

For the external interrupt $INT_1$, the external input pin $P3_3$ or $P3_6$ can be selected by the $INT_1$ input port selection bit in the interrupt edge selection register (bit 2 of address $003A_{16}$).

However, since there is no $P3_6/INT_1$ pin in the 32-pin version PWQN0036KA-A package, select $P3_3/INT_1$ pin. By the key-on wakeup selection bit, enable/disable of a key-on wakeup of $P0_0$, $P0_4$, and $P0_6$ pins can be selected, respectively.




**Note 1:** P36 does not exist for the 32-pin version and the PWQN0036KA-A package.
Select P33 for the INT1 function.

**Fig. 24 Structure of Interrupt-related registers**

**• Interrupt Request Generation, Acceptance, and Handling**

Interrupts have the following three phases.

(i) Interrupt Request Generation
An interrupt request is generated by an interrupt source (external interrupt signal input, timer underflow, etc.) and the corresponding request bit is set to “1”.

(ii) Interrupt Request Acceptance
Based on the interrupt acceptance timing in each instruction cycle, the interrupt control circuit determines acceptance conditions (interrupt request bit, interrupt enable bit, and interrupt disable flag) and interrupt priority levels for accepting interrupt requests. When two or more interrupt requests are generated simultaneously, the highest priority interrupt is accepted. The value of the interrupt request bit for an unaccepted interrupt remains the same and acceptance is determined at the next interrupt acceptance timing point.

(iii) Handling of Accepted Interrupt Request
The accepted interrupt request is processed.

Figure 25 shows the time up to execution in the interrupt processing routine, and Figure 26 shows the interrupt sequence.
Figure 27 shows the timing of interrupt request generation, interrupt request bit, and interrupt request acceptance.

**• Interrupt Handling Execution**

When interrupt handling is executed, the following operations are performed automatically.

(1) Once the currently executing instruction is completed, an interrupt request is accepted.
(2) The contents of the program counters and the processor status register at this point are pushed onto the stack area in order from 1 to 3.
1.High-order bits of program counter (PCH)
2.Low-order bits of program counter (PCL)
3.Processor status register (PS)
(3) Concurrently with the push operation, the jump address of the corresponding interrupt (the start address of the interrupt processing routine) is transferred from the interrupt vector to the program counter.
(4) The interrupt request bit for the corresponding interrupt is set to “0”. Also, the interrupt disable flag is set to “1” and multiple interrupts are disabled.
(5) The interrupt routine is executed.
(6) When the RTI instruction is executed, the contents of the registers pushed onto the stack area are popped off in the order from 3 to 1. Then, the routine that was before running interrupt processing resumes.

As described above, it is necessary to set the stack pointer and the jump address in the vector area corresponding to each interrupt to execute the interrupt processing routine.

**■ Notes on Interrupts**

When setting the followings, the interrupt request bit may be set to “1”.

•When switching external interrupt active edge
Related registers:
Interrupt edge selection register (address $003A_{16}$)
Timer X mode register (address $002B_{16}$)
Capture mode register (address $0020_{16}$)

When not requiring the interrupt occurrence synchronized with these setting, take the following sequence.

① Set the corresponding interrupt enable bit to “0” (disabled).
② Set the interrupt edge select bit (active edge switch bit, trigger mode bit).
③ Set the corresponding interrupt request bit to “0” after 1 or more instructions have been executed.
④ Set the corresponding interrupt enable bit to “1” (enabled).

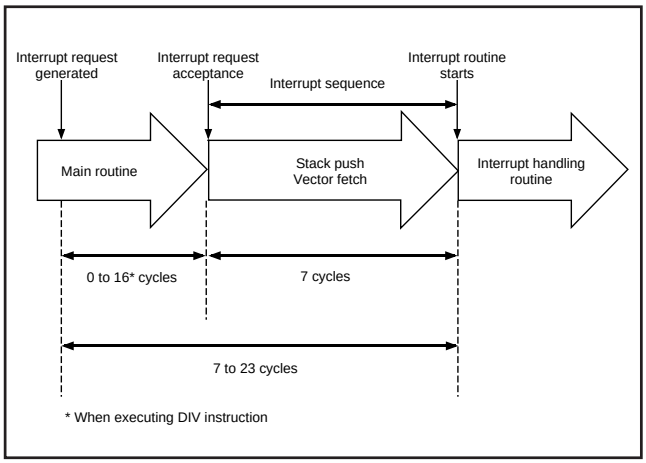
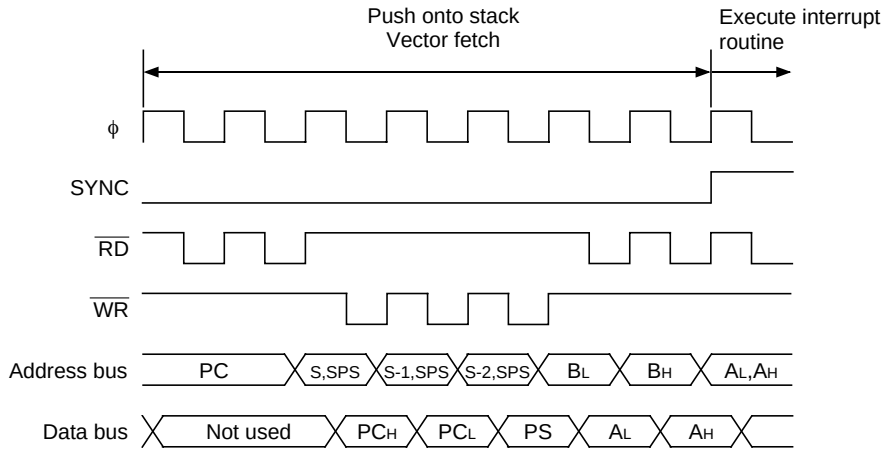
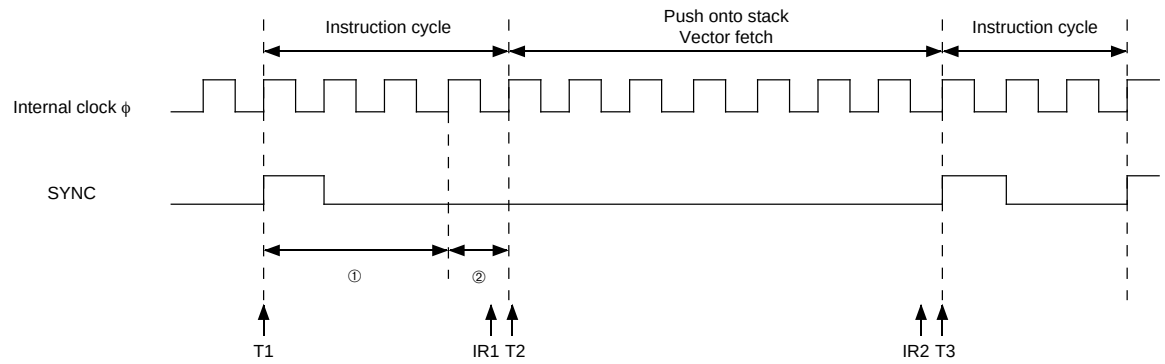



**Fig. 25 Time up to execution in interrupt routine**




SYNC : CPU operation code fetch cycle
(This is an internal signal that cannot be observed from the external unit.)
$B_L$, $B_H$ : Vector address of each interrupt
$A_L$, $A_H$ : Jump destination address of each interrupt
SPS : "$00_{16}$" or "$01_{16}$"
([SPS] is a page selected by the stack page selection bit of CPU mode register.)

**Fig. 26 Interrupt sequence**




T1 T2 T3 : Interrupt acceptance timing points
IR1 IR2 : Timings points at which the interrupt request bit is set to "1".

**Note** : Period 2 indicates the last ϕ cycle during one instruction cycle.

(1) The interrupt request bit for an interrupt request generated during period 1 is set to "1" at timing point IR1.
(2) The interrupt request bit for an interrupt request generated during period 2 is set to "1" at timing point IR1 or IR2. The timing point at which the bit is set to "1" varies depending on conditions. When two or more interrupt requests are generated during the period 2, each request bit may be set to "1" at timing point IR1 or IR2 separately.

**Fig. 27 Timing of interrupt request generation, interrupt request bit, and interrupt acceptance**

## 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 28, where an interrupt request is generated by pressing one of the keys provided as an active-low key matrix which uses ports P0$_0$ to P0$_3$ as input ports.

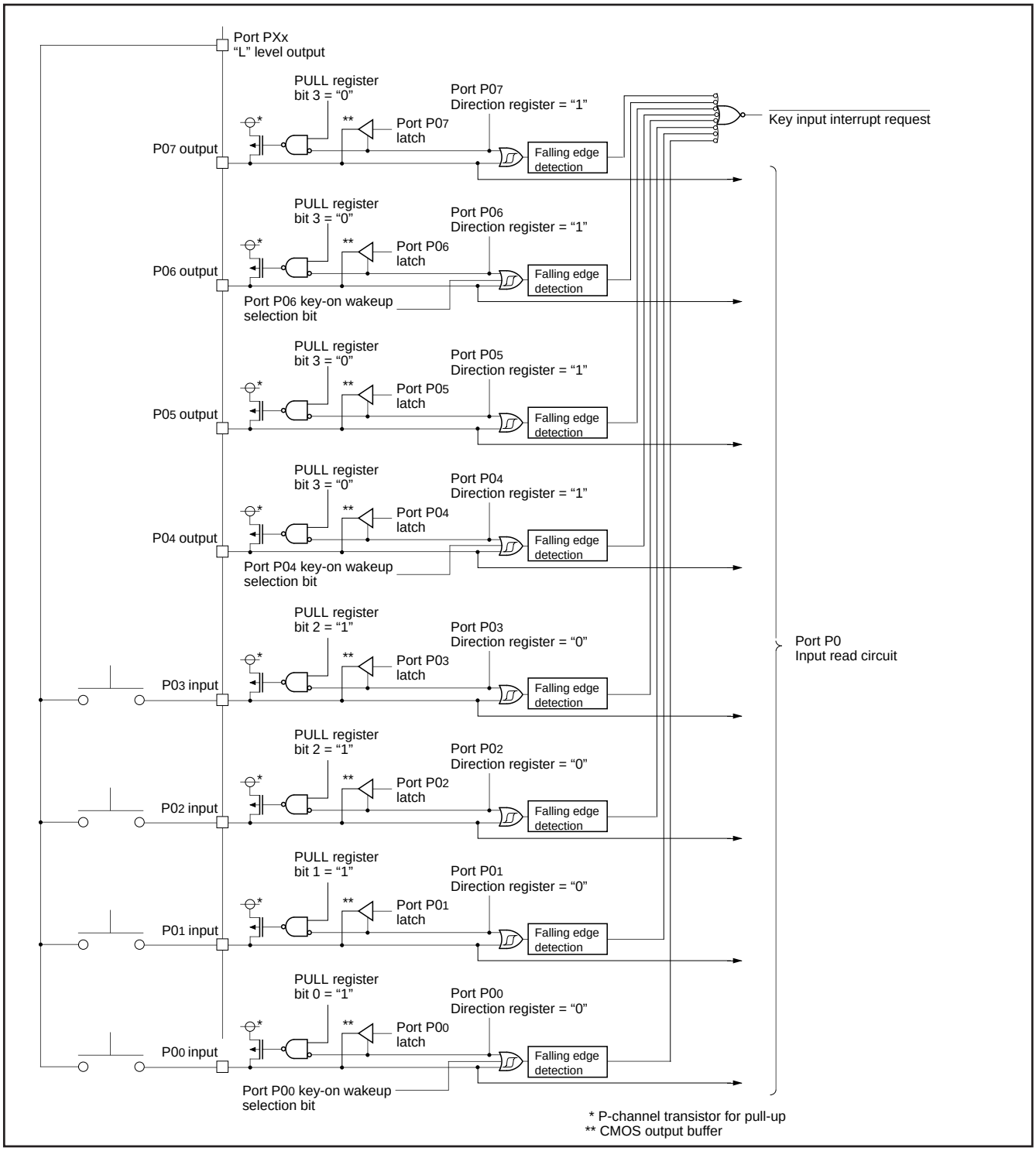



**Fig. 28 Connection example when using key input interrupt and port P0 block diagram**

## Timers

The 7542 Group has 4 timers: timer 1, timer X, timer A and timer B.

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”.

• Frequency divider for timer

According to the clock division selection bits (b7 and b6) of CPU mode register ($003B_{16}$), the count source of frequency divider is set as follows;

b7b6 = “00”(high-speed), “01”(middle-speed), “11”(double-speed): $X_{IN}$

b7b6 = “10”(On-chip oscillator): On-chip oscillator

## ●Timer 1

Timer 1 is an 8-bit timer and counts the prescaler output.

When Timer 1 underflows, the timer 1 interrupt request bit is set to “1”.

Prescaler 1 is an 8-bit prescaler and counts the signal which is the oscillation frequency divided by 16.

Prescaler 1 and Timer 1 have the prescaler 1 latch and the timer 1 latch to retain the reload value, respectively. The value of prescaler 1 latch is set to Prescaler 1 when Prescaler 1 underflows. The value of timer 1 latch is set to Timer 1 when Timer 1 underflows.

When writing to Prescaler 1 (PRE1) is executed, the value is written to both the prescaler 1 latch and Prescaler 1.

When writing to Timer 1 (T1) is executed, the value is written to both the timer 1 latch and Timer 1.

When reading from Prescaler 1 (PRE1) and Timer 1 (T1) is executed, each count value is read out.

Timer 1 always operates in the timer mode.

Prescaler 1 counts the signal which is the oscillation frequency divided by 16. Each time the count clock is input, the contents of Prescaler 1 is decremented by 1. When the contents of Prescaler 1 reach “$00_{16}$”, an underflow occurs at the next count clock, and the prescaler 1 latch is reloaded into Prescaler 1 and count continues. The division ratio of Prescaler 1 is 1/(n+1) provided that the value of Prescaler 1 is n.

The contents of Timer 1 is decremented by 1 each time the underflow signal of Prescaler 1 is input. When the contents of Timer 1 reach “$00_{16}$”, an underflow occurs at the next count clock, and the timer 1 latch is reloaded into Timer 1 and count continues. The division ratio of Timer 1 is 1/(m+1) provided that the value of Timer 1 is m. Accordingly, the division ratio of Prescaler 1 and Timer 1 is 1/((n+1)×(m+1)) provided that the value of Prescaler 1 is n and the value of Timer 1 is m.

Timer 1 cannot stop counting by software.

## ●Timer X

Timer X is an 8-bit timer and counts the prescaler X output.

When Timer X underflows, the timer X interrupt request bit is set to “1”.

Prescaler X is an 8-bit prescaler and counts the signal selected by the timer X count source selection bit.

Prescaler X and Timer X have the prescaler X latch and the timer X latch to retain the reload value, respectively. The value of prescaler X latch is set to Prescaler X when Prescaler X underflows. The value of timer X latch is set to Timer X when Timer X underflows.

When writing to Prescaler X (PREX) is executed, the value is written to both the prescaler X latch and Prescaler X.

When writing to Timer X (TX) is executed, the value is written to both the timer X latch and Timer X.

When reading from Prescaler X (PREX) and Timer X (TX) is executed, each count value is read out.

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

**(1) Timer mode**

Prescaler X counts the count source selected by the timer X count source selection bits. Each time the count clock is input, the contents of Prescaler X is decremented by 1. When the contents of Prescaler X reach “$00_{16}$”, an underflow occurs at the next count clock, and the prescaler X latch is reloaded into Prescaler X and count continues. The division ratio of Prescaler X is 1/(n+1) provided that the value of Prescaler X is n.

The contents of Timer X is decremented by 1 each time the underflow signal of Prescaler X is input. When the contents of Timer X reach “$00_{16}$”, an underflow occurs at the next count clock, and the timer X latch is reloaded into Timer X and count continues. The division ratio of Timer X is 1/(m+1) provided that the value of Timer X is m. Accordingly, the division ratio of Prescaler X and Timer X is 1/((n+1)×(m+1)) provided that the value of Prescaler X is n and the value of Timer X is m.

**(2) Pulse output mode**

In the pulse output mode, the waveform whose polarity is inverted each time timer X underflows is output from the $CNTR_0$ pin.

The output level of $CNTR_0$ pin can be selected by the $CNTR_0$ active edge switch bit. When the $CNTR_0$ active edge switch bit is “0”, the output of $CNTR_0$ pin is started at “H” level. When this bit is “1”, the output is started at “L” level.

Also, the inverted waveform of pulse output from $CNTR_0$ pin can be output from $TX_{OUT}$ pin by setting “1” to the $P0_3/TX_{OUT}$ output valid bit.

When using a timer in this mode, set the port $P1_4$ and $P0_3$ direction registers to output mode.

**(3) Event counter mode**

The timer A counts signals input from the $P1_4/CNTR_0$ pin.

Except for this, the operation in event counter mode is the same as in timer mode.

The active edge of $CNTR_0$ pin input signal can be selected from rising or falling by the $CNTR_0$ active edge switch bit .

**(4) Pulse width measurement mode**

In the pulse width measurement mode, the pulse width of the signal input to P14/CNTR0 pin is measured.

The operation of Timer X can be controlled by the level of the signal input from the CNTR0 pin.

When the CNTR0 active edge switch bit is “0”, the signal selected by the timer X count source selection bit is counted while the input signal level of CNTR0 pin is “H”. The count is stopped while the pin is “L”. Also, when the CNTR0 active edge switch bit is “1”, the signal selected by the timer X count source selection bit is counted while the input signal level of CNTR0 pin is “L”. The count is stopped while the pin is “H”.

Timer X can stop counting by setting “1” to the timer X count stop bit in any mode.

Also, when Timer X underflows, the timer X interrupt request bit is set to “1”.

Note on Timer X is described below;

**■ Note on Timer X**

(1) CNTR0 interrupt active edge selection-1

CNTR0 interrupt active edge depends on the CNTR0 active edge switch bit.

When this bit is “0”, the CNTR0 interrupt request bit is set to “1” at the falling edge of CNTR0 pin input signal. When this bit is “1”, the CNTR0 interrupt request bit is set to “1” at the rising edge of CNTR0 pin input signal.

(2) CNTR0 interrupt active edge selection-2

According to the setting value of CNTR0 active edge switch bit, the interrupt request bit may be set to “1”.

When not requiring the interrupt occurrence synchronized with these setting, take the following sequence.

① Set the corresponding interrupt enable bit to “0” (disabled).
② Set the active edge switch bit.
③ Set the corresponding interrupt request bit to “0” after 1 or more instructions have been executed.
④ Set the corresponding interrupt enable bit to “1” (enabled).

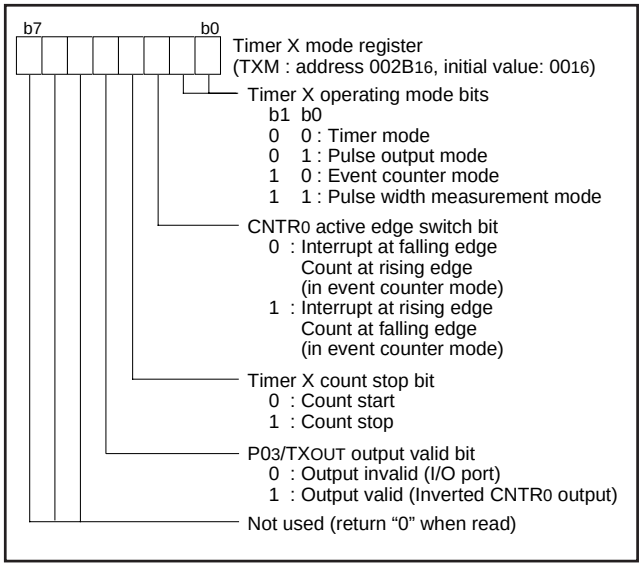



**Fig. 29 Structure of timer X mode register**

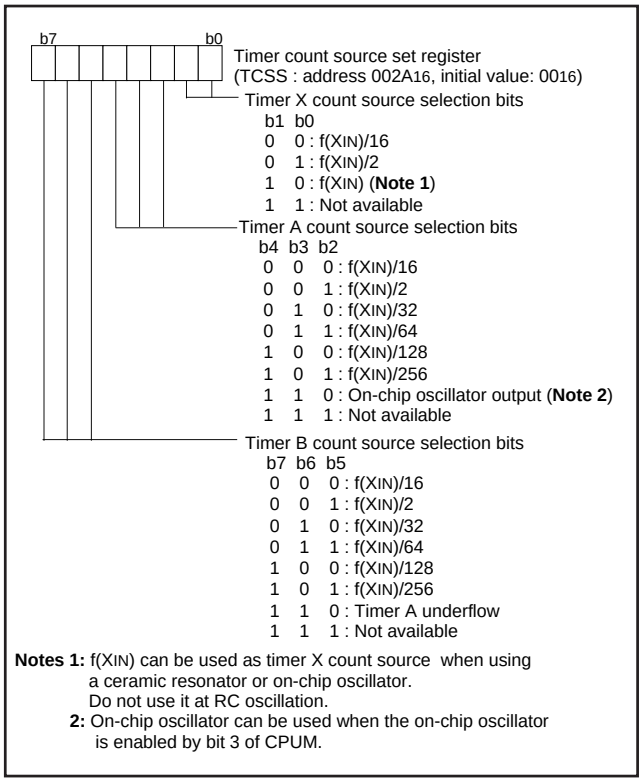



**Fig. 30 Timer count source set register**

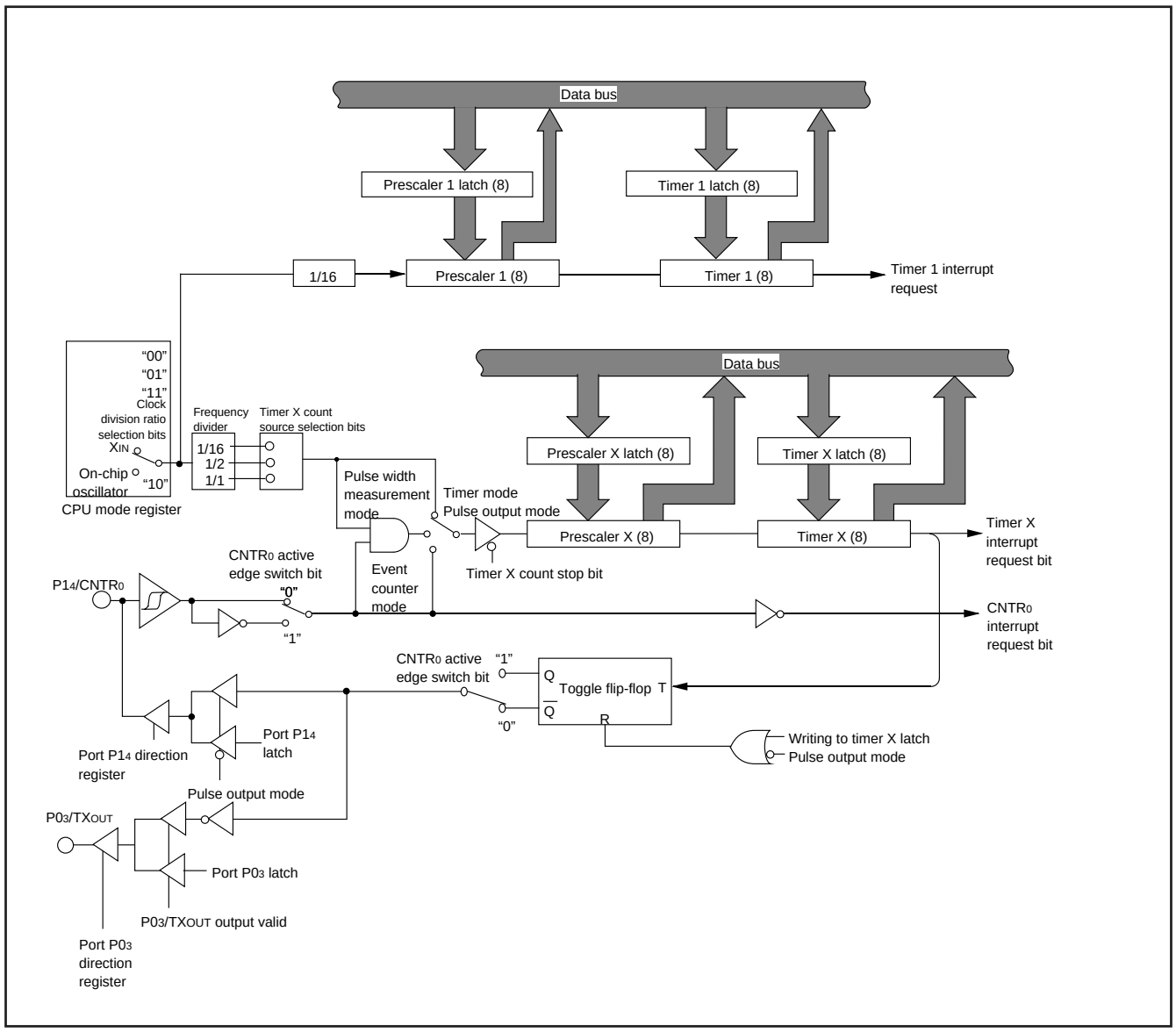



**Fig. 31 Block diagram of timer 1 and timer X**

## ●Timer A,B

Timer A and Timer B are 16-bit timers and counts the signal which is the oscillation frequency selected by setting of the timer count source set register (TCSS). Timer A and Timer B have the same function except of the count source clock selection.

The count source clock of Timer A is selected from among 1/2,1/16, 1/32, 1/64, 1/128, 1/256 of f(XIN) clock and on-chip oscillator clock.
The count source clock of Timer B is selected from among 1/2, 1/16, 1/32, 1/64, 1/128, 1/256 of f(XIN) clock and Timer A underflow.

Timer A (B) consists of the low-order of Timer A: TAL (Timer B: TBL) and the high-order of Timer A: TAH (Timer B: TBH). Timer A (B) is decremented by 1 when each time of the count clock is input. When the contents of Timer A (B) reach "$0000_{16}$", an underflow occurs at the next count clock, and the timer latch is reloaded into timer. When Timer A (B) underflows, the Timer A (B) interrupt request bit is set to "1".
Timer A (B) has the Timer A (B) latch to retain the load value. The value of timer A (B) latch is set to Timer A (B) at the timing of Timer A (B) underflow. The division ratio of Timer A (B) is 1/(n+1) provided that the value of Timer A (B) is n.

When writing to both the low-order of Timer A (B) and the high order of Timer A (B) is executed, writing to "latch only" or "latch and timer" can be selected by the setting value of the timer A (B) write control bit.
When reading from Timer A (B) register is executed, the count value of Timer A (B) is read out.

Be sure to write to/read out the low-order of Timer A (B) and the high-order of Timer A (B) in the following order;
• Read
Read the high-order of Timer A (B) first, and the low-order of Timer A (B) next and be sure to read both high-order and low-order.
• Write
Write to the low-order of Timer A (B) first, and the high-order of Timer A (B) next and be sure to write both low-order and high order.

Timer A and Timer B can be used for the timing timer of Input capture and Output compare function.

## ■ Notes on Timer A, B

(1) Setting of timer value
When "1: Write to only latch" is set to the timer A (B) write control bit, written data to timer register is set to only latch even if timer is stopped. Accordingly, in order to set the initial value for timer when it is stopped, set "0: Write to latch and timer simultaneously" to timer A (B) write control bit.

(2) Read/write of timer A
Stop timer A to read/write its data when the system is in the following state;
• CPU operation clock source: XIN oscillation
• Timer A count source: On-chip oscillator output

(3) Read/write of timer B
Stop timer B to read/write its data when the system is in the following state;
• CPU operation clock source: XIN oscillation
• Timer B count source: Timer A underflow
• Timer A count source: On-chip oscillator output

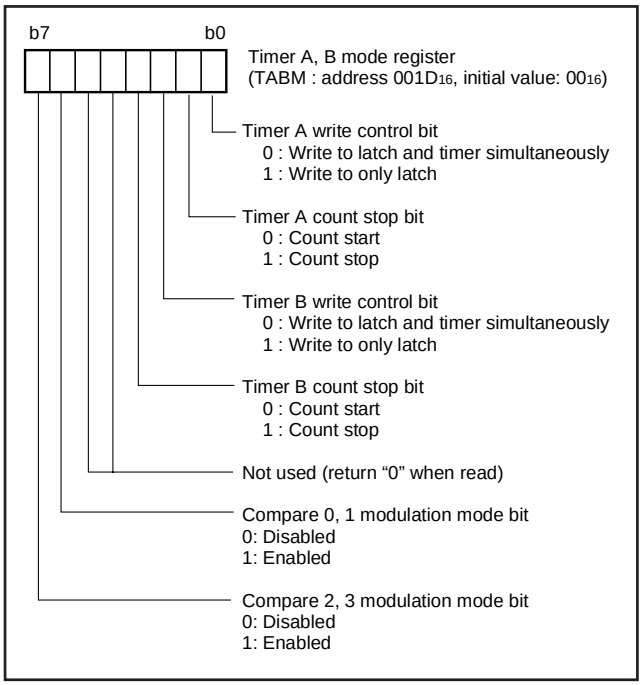



**Fig. 32 Structure of timer A, B mode register**

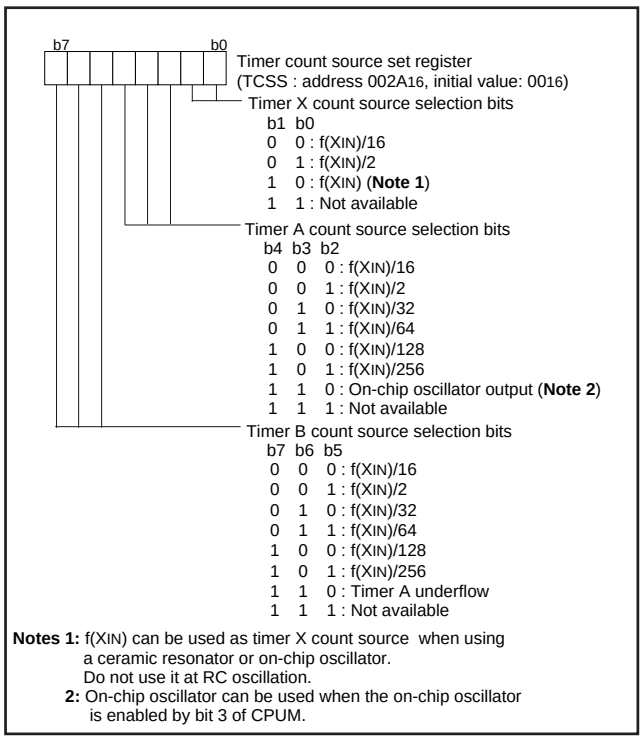



**Fig. 33 Timer count source set register**

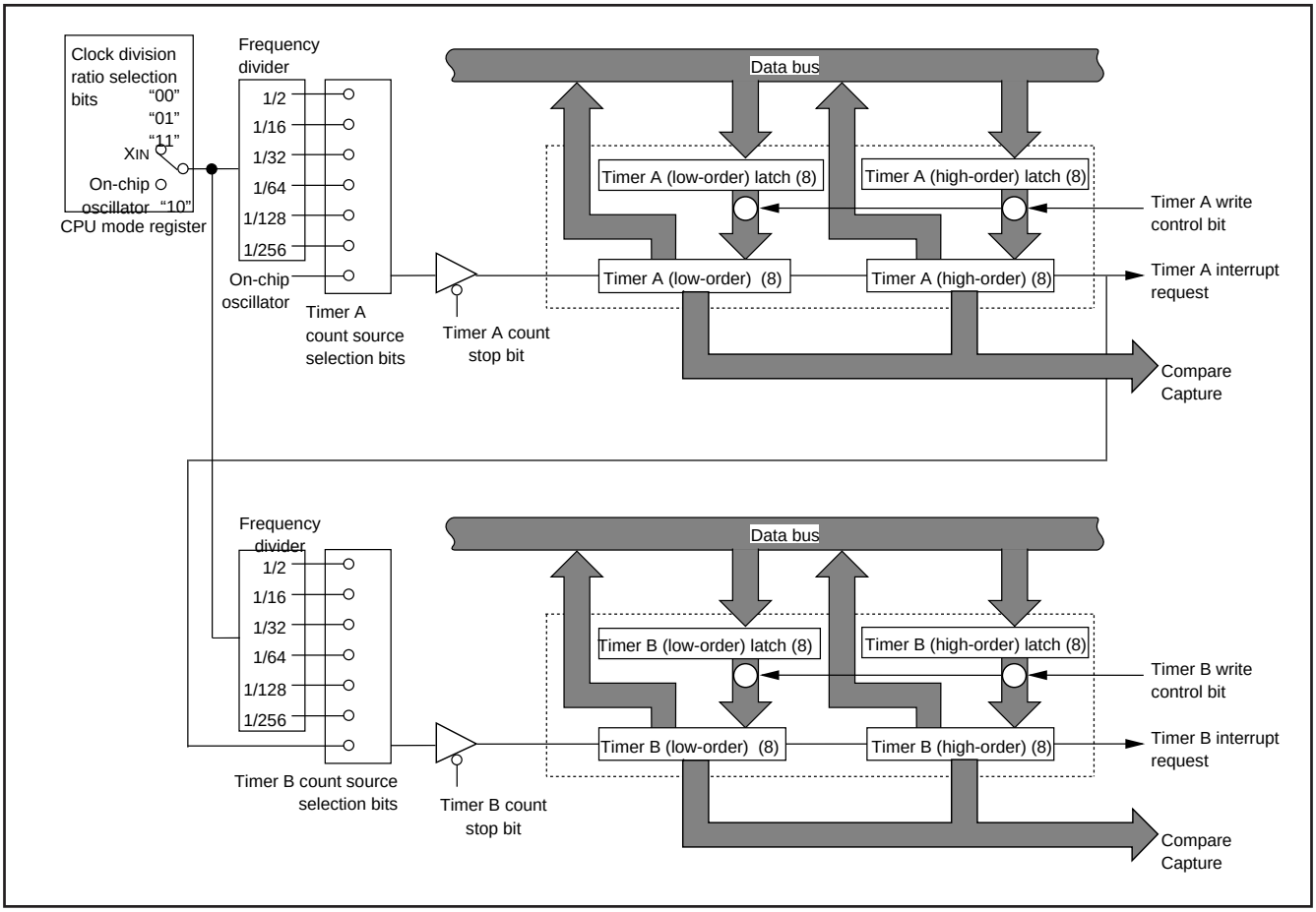



**Fig. 34 Block diagram of timer A and timer B**

## Output compare

7542 group has 4-output compare channels. Each channel (0 to 3) has the same function and can be used to output waveform by using count value of either Timer A or Timer B.
The source timer for each channel is selected by setting value of the compare x (x = 0, 1, 2, 3) timer source bit. Timer A and Timer B can be selected for the source timer to each channel, respectively.

To use each compare channel, set “1” to the compare x output port bit and set the port direction register corresponding to compare channel to output mode.

The compare value for each channel is set to the compare register (low-order) and compare register (high-order).
Writing to the register for each channel is controlled by setting value of compare register write pointer. Writing to each register is in the following order;

1. Set the value of corresponded output compare channel to the compare register write pointer.
2. Write a value to the compare register (low-order) and compare register (high-order).
3. Set “1” to the compare latch y (y = 00, 01, 10, 11, 20, 21, 30, 31) re-load bit.
   When “1” is set to the compare latch y re-load bit, the value set to the compare register is loaded to compare latch when the next timer underflow.

When count value of timer and setting value of compare latch is matched, compare output trigger occurs.
When “1: Enabled” is set to the compare trigger x enable bit, the output waveform from port is inverted by compare trigger.
When “0: Disabled” is set to the compare trigger x enable bit, the output waveform is not inverted, so port output can be fixed to “H” or “L”.

When “0: Positive” is set to the compare x output level latch, the compare output waveform is turned to “H level” at compare latch x0’s match and turned to “L level” at compare latch x1’s match.
When “1 :Negative” is set to the compare x output level latch, the compare output waveform is turned to “L level” at compare latch x0’s match and turned to “H level” at compare latch x1’s match.
The compare output level of each channel can be confirmed by reading the compare x output status bit.

Compare output interrupt is available when match of each compare channel and timer count value. The interrupt request from each channel can be disabled or enabled by setting value of compare latch y interrupt source bit.

**Compare 0,1 (2,3) modulation mode**

In compare modulation mode, modulation waveform can be generated by using compare channel 0 and 1, or compare channel 2 and 3.
To use this mode,

- Set “1: Enabled” to the compare 0,1 (2, 3) modulation mode bit.
- Set Timer A underflow for Timer B count source.
- Set Timer A for the timer source of compare channel 0 (2).
- Set Timer B for the timer source of compare channel 1 (3).

In this mode, AND waveform of compare 0 (1) and compare 2 (3) is generated from Port $P0_1$ and $P3_1$, respectively. Accordingly, in order to use this mode, set “1” to the compare 0 output port bit or compare 2 output port bit.

## ■ Notes on Output Compare

- When the selected source timer of each compare channel is stopped, written data to compare register is loaded to the compare latch simultaneously.
- Do not write the same data to both of compare latch x0 and x1.
- When setting value of the compare latch is larger than timer setting value, compare match signal is not generated. Accordingly, the output waveform is fixed to “L” or “H” level.
  However, when setting value of another compare latch is smaller than timer setting value, this compare match signal is generated. Accordingly, compare match interrupt occurs.
- When the compare x trigger enable bit is cleared to “0” (disabled), the match trigger to the waveform output circuit is disabled, and the output waveform can be fixed to “L” or “H” level.
  However, in this case, the compare match signal is generated. Accordingly, compare match interrupt occurs.

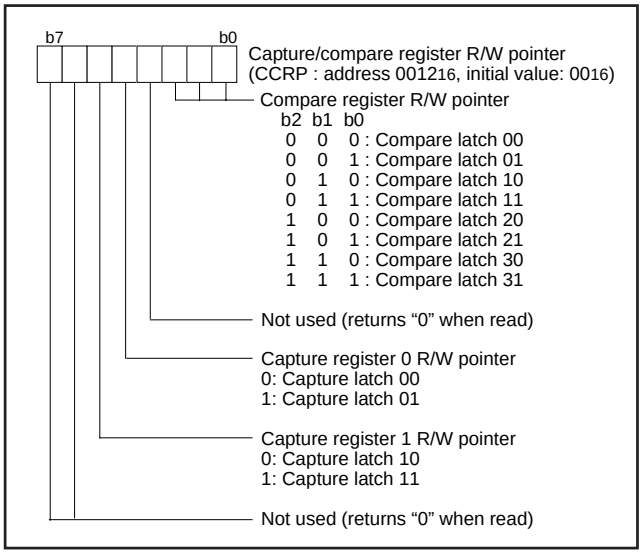



**Fig. 35 Structure of capture/compare register R/W pointer**

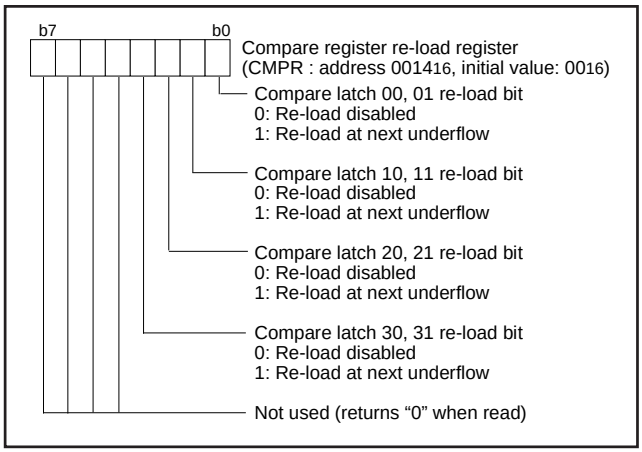



**Fig. 36 Structure of compare register re-load register**

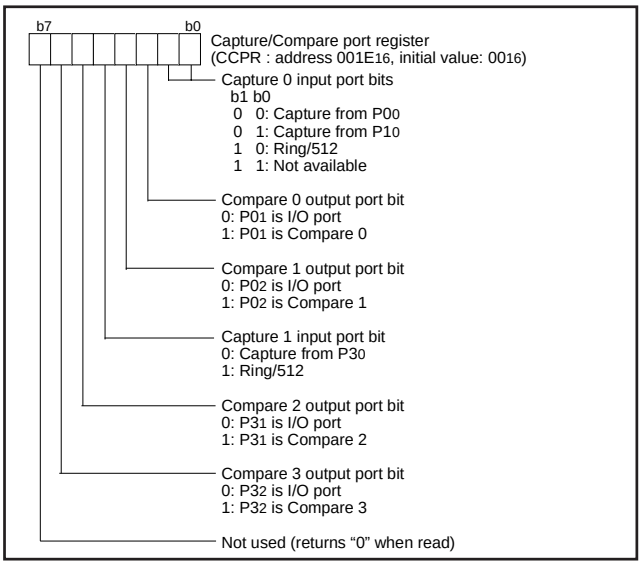



**Fig. 37 Structure of capture/compare port register**

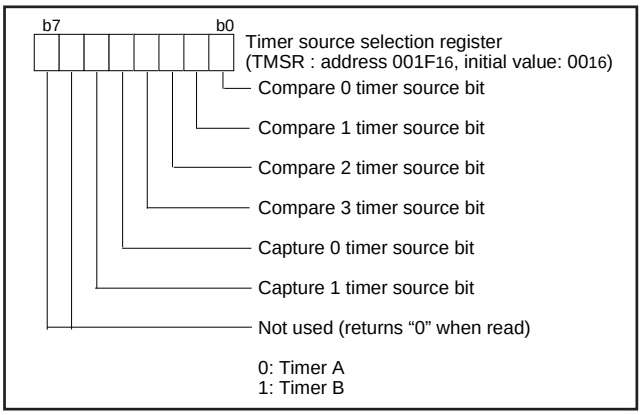



**Fig. 38 Structure of timer source selection register**

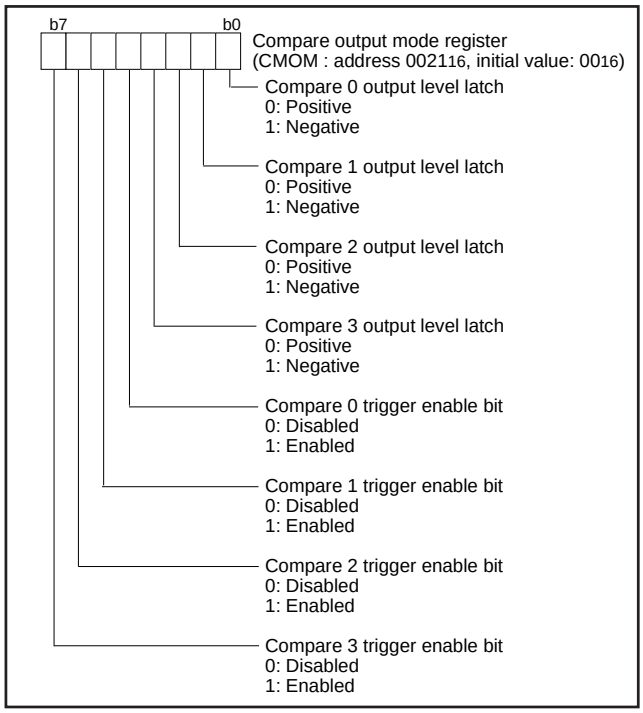



**Fig. 39 Structure of compare output mode register**

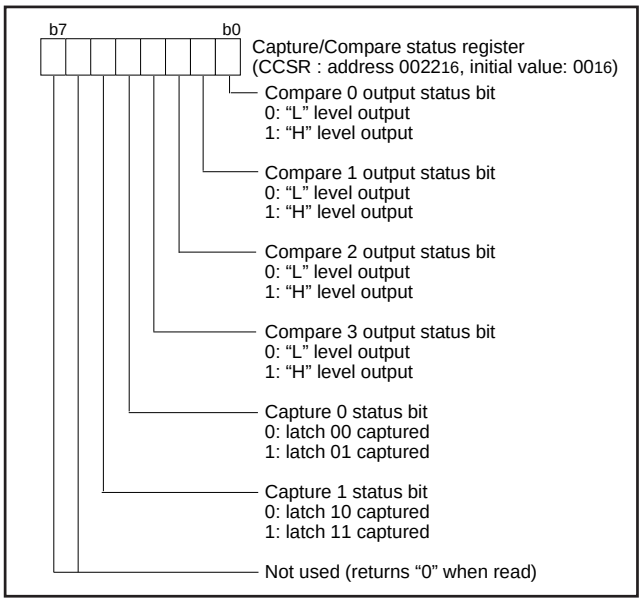



**Fig. 40 Structure of capture/compare status register**

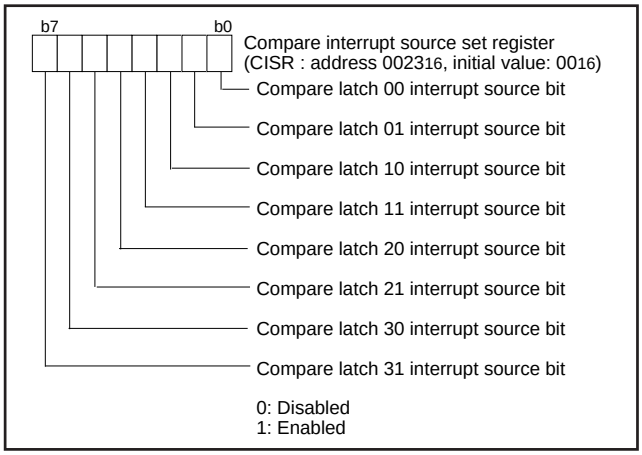



**Fig. 41 Structure of compare interrupt source register**

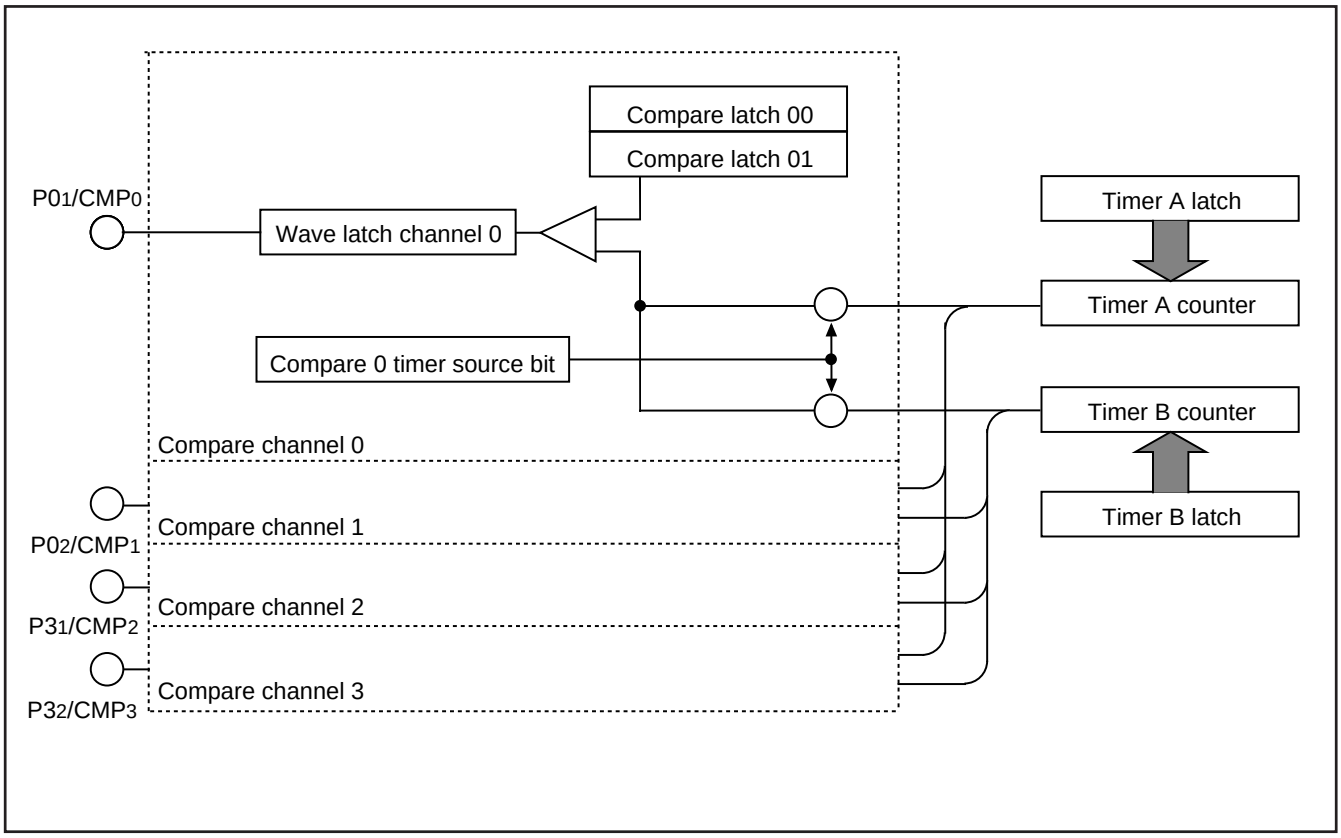



**Fig. 42 Block diagram of output compare**

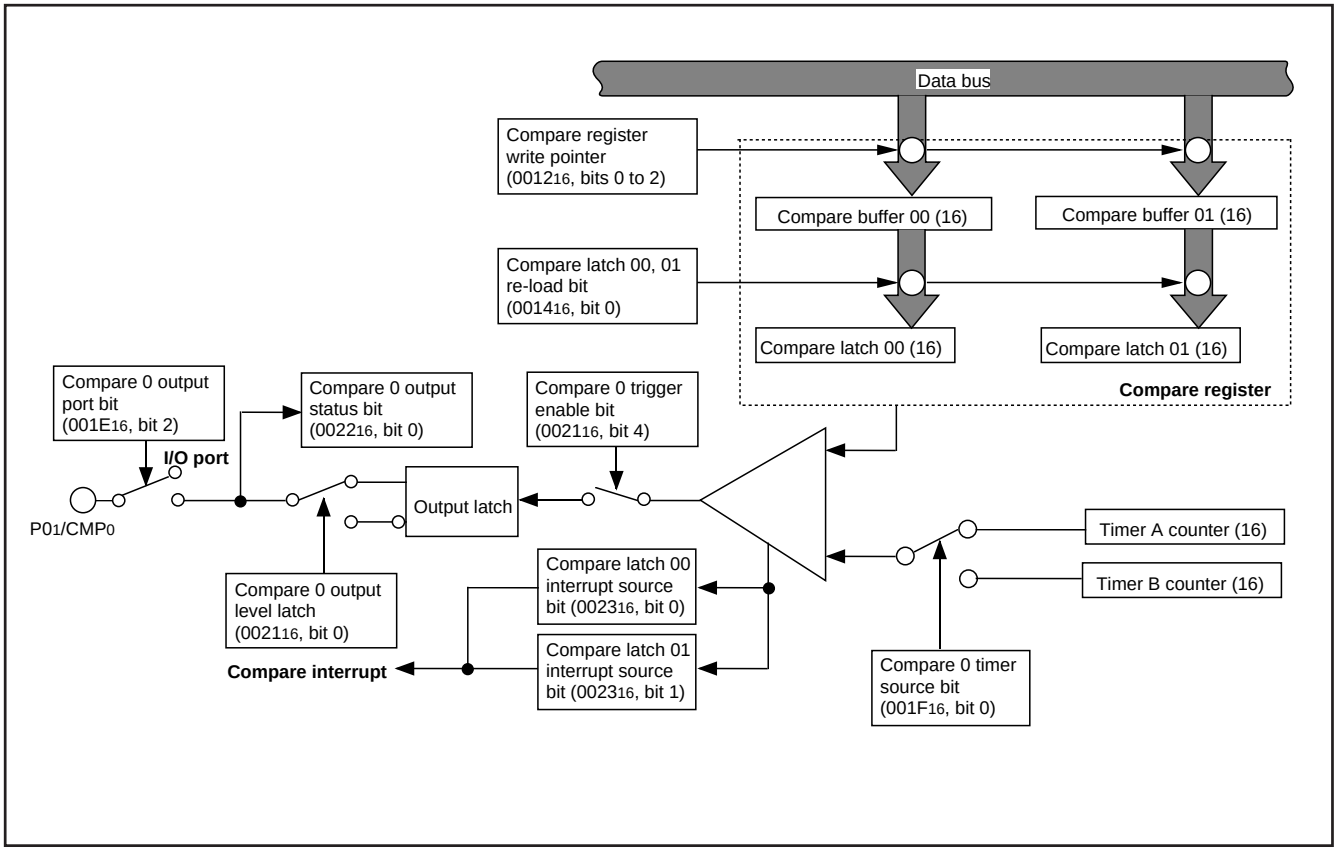



**Fig. 43 Block diagram of compare channel 0**

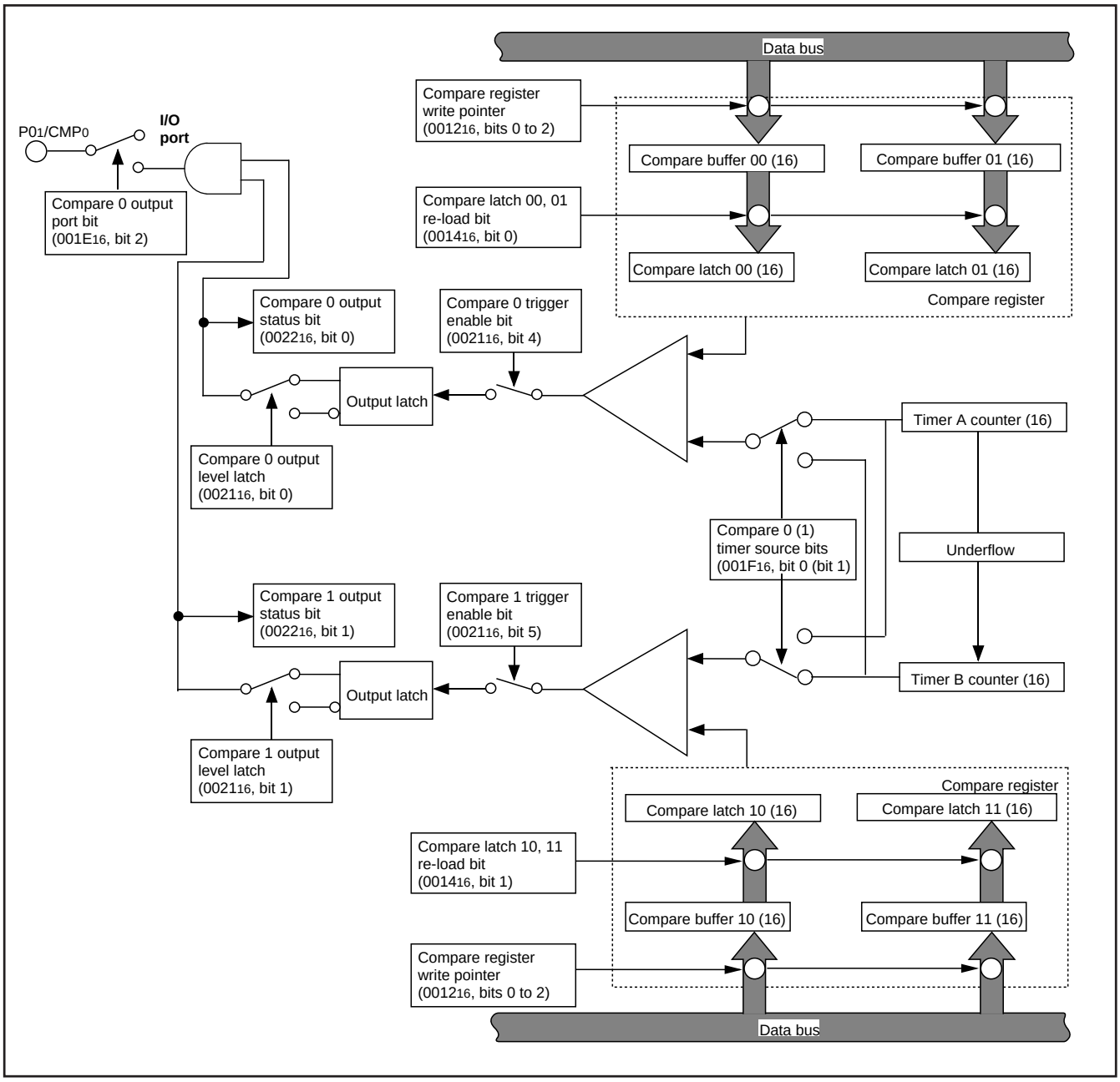



**Fig. 44 Block diagram at modulation mode**

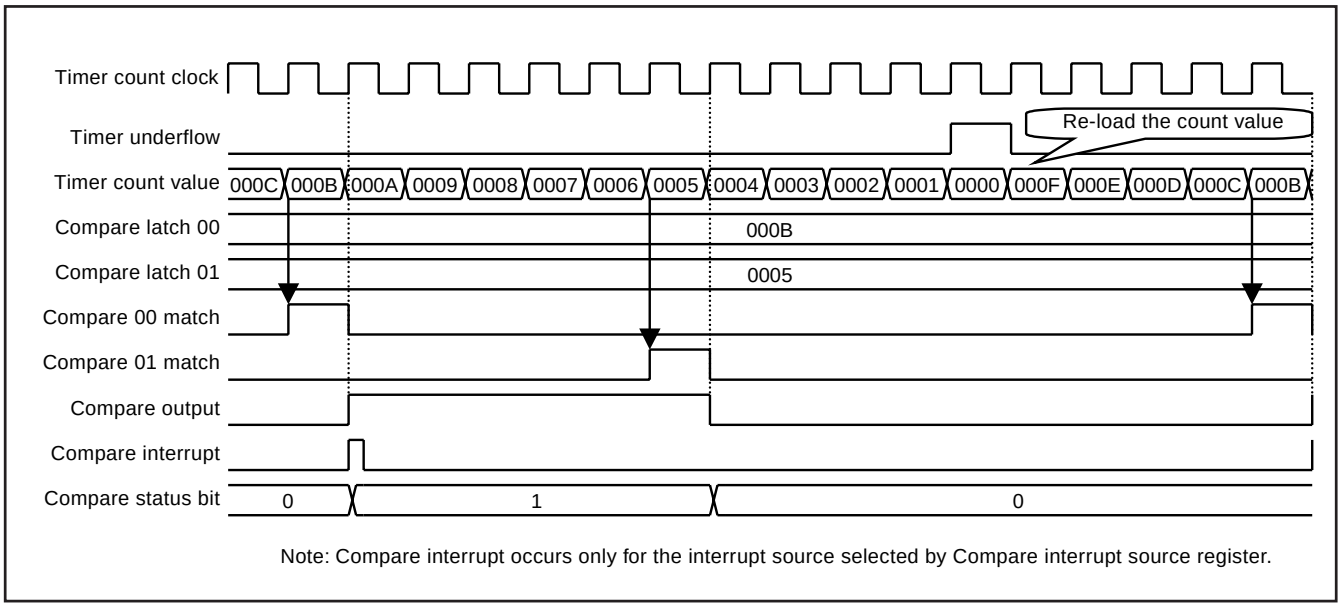



**Fig. 45 Output compare mode (general waveform)**

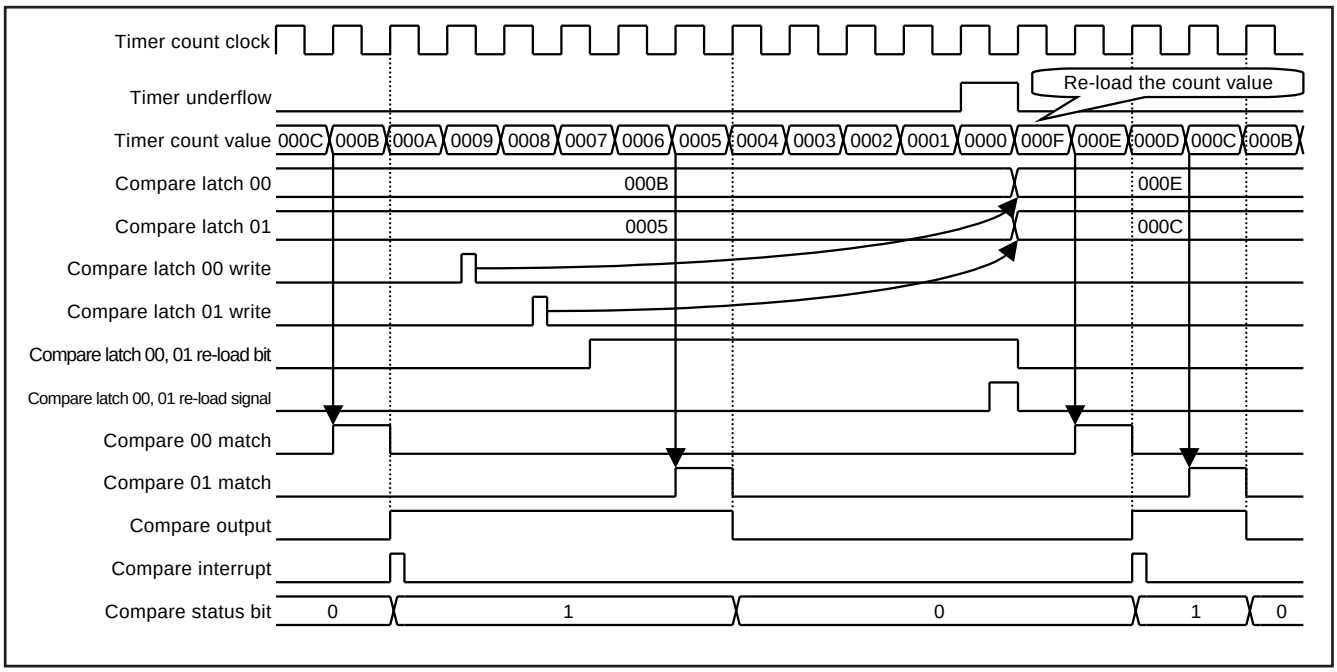
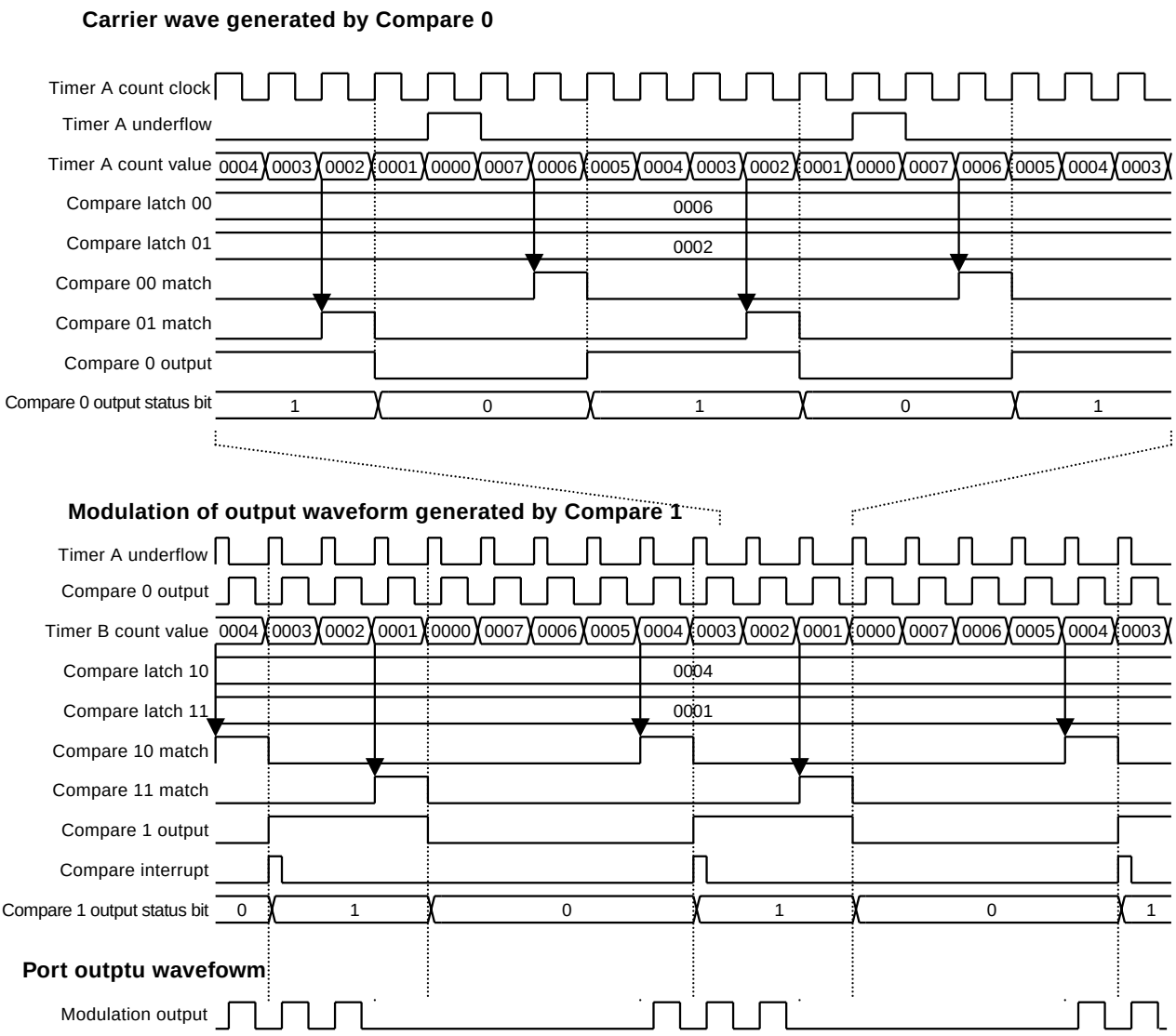



**Fig. 46 Output compare mode (compare register write timing)**




**Note:** Compare interrupt occurs only for the interrupt source selected by Compare interrupt source register.

**Fig. 47 Output compare mode (compare 0, 1 modulation mode)**

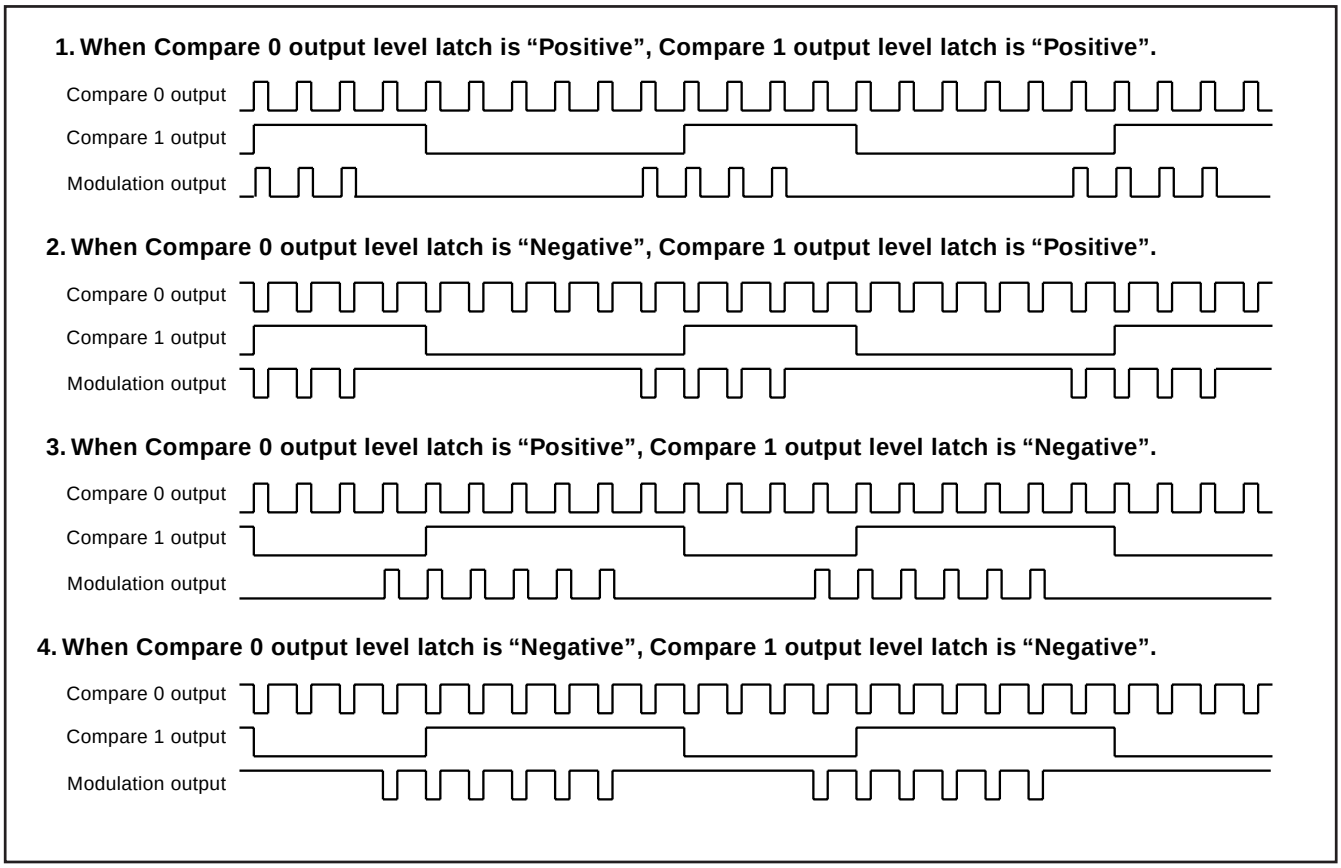



**Fig. 48 Output compare mode (compare 0, 1 modulation mode: effect of output level latch)**

## Input capture

7542 group has 2-input capture channels. Each channel (0 and 1) has the same function and can be used to capture count value of either Timer A or Timer B.
The source timer for each channel is selected by setting value of the capture x (x = 0, 1) timer source bit. Timer A and Timer B can be selected for the source timer to each channel, respectively.

To use each capture channel, set the capture x input port bits and set the port direction register corresponding to capture channel to input mode.

The input capture circuit retains the count value of selected timer when external trigger is input. The timer count value is retained to the capture latch x0 when rising edge is input and is retained to the capture latch x1 when falling edge is input.
The count value of timer can be retained by software by capture y (y = 00, 01, 10, 11) software trigger bit too. When "1" is set to this bit, count value of timer is retained to the corresponded capture latch.
When reading from the capture y software trigger bit is executed, "0" is read out.
The latest status of capture latch can be confirmed by reading of the capture x status bit. This bit indicates the capture latch which latest data is in.

The valid trigger edge for capture interrupt is set by the capture x interrupt edge selection bits. (Regardless of the setting value of capture x interrupt edge selection bits, timer count values for both edges are retained to the capture latch.)

Each capture input has the noise filter circuit that judges continuous 4-time same level with sampling clock to be valid. The sampling clock of noise filter is set by the capture x noise filter clock selection bits.

Reading from the register for each channel is controlled by setting value of the capture register read pointer. Reading from each register is in the following order;

1. Set the value of the corresponded input capture channel to the capture register read pointer.
2. Read from the capture register (low-order) and capture register (high-order).

## ■ Notes on Input Capture

- If the capture trigger is input while the capture register (low-order and high-order) is in read, captured value is changed between high-order reading and low-order reading. Accordingly, some countermeasure by software is recommended, for example comparing the values that twice of read.
- When the on-chip-oscillator is selected for Timer A count source, Timer A cannot be used for the capture source timer.
  Timer B cannot be used for the capture source timer when the system is in the following state;
  - CPU operation clock source: XIN oscillation
  - Timer B count source: Timer A underflow
  - Timer A count source: On-chip oscillator output
- When writing "1" to capture latch x0 (x1) software trigger bit of capture latch x0 and x1 at the same time, or external trigger and software trigger occur simultaneously, the set value of capture x status bit is undefined.
- When setting the interrupt active edge selection bit and noise filter clock selection bit of external interrupt CAP0, CAP1, the interrupt request bit may be set to "1".
  When not requiring the interrupt occurrence synchronized with these setting, take the following sequence.

① Set the corresponding interrupt enable bit to "0" (disabled).
② Set the interrupt edge selection bit or noise filter clock selection bit.
③ Set the corresponding interrupt request bit to "0" after 1 or more instructions have been executed.
④ Set the corresponding interrupt enable bit to "1" (enabled).

- When the capture interrupt is used as the interrupt for return from stop mode, set the capture x noise filter clock selection bits to "00 (Filter stop)".

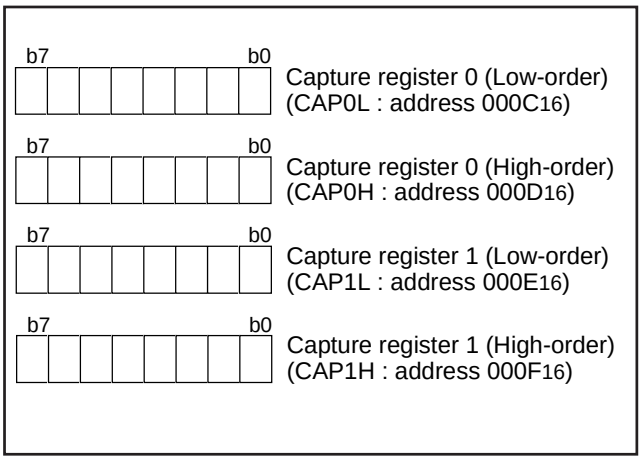



**Fig. 49 Structure of capture software trigger register**

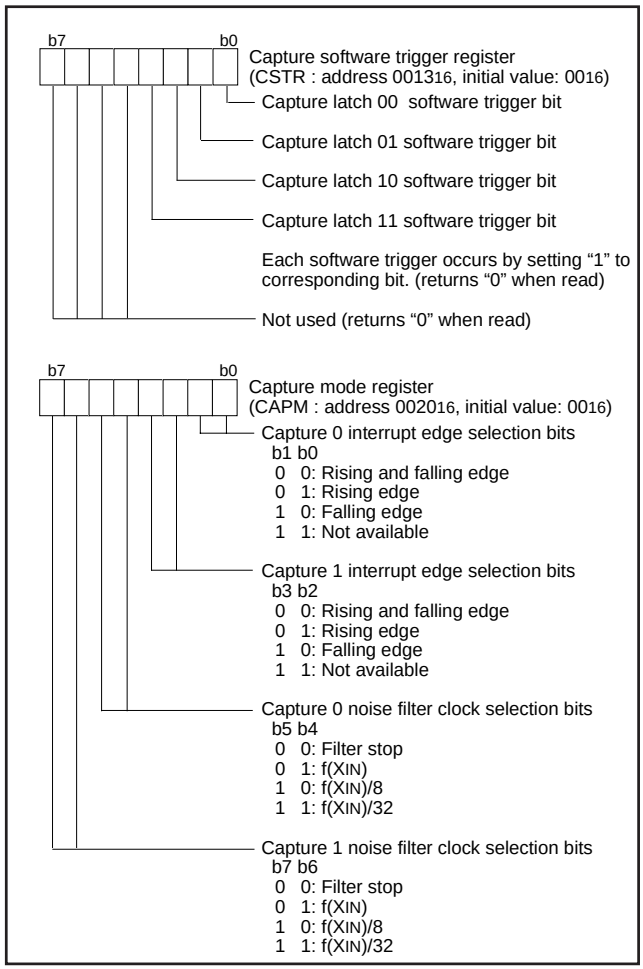



**Fig. 50 Structure of capture software trigger register/capture mode register**

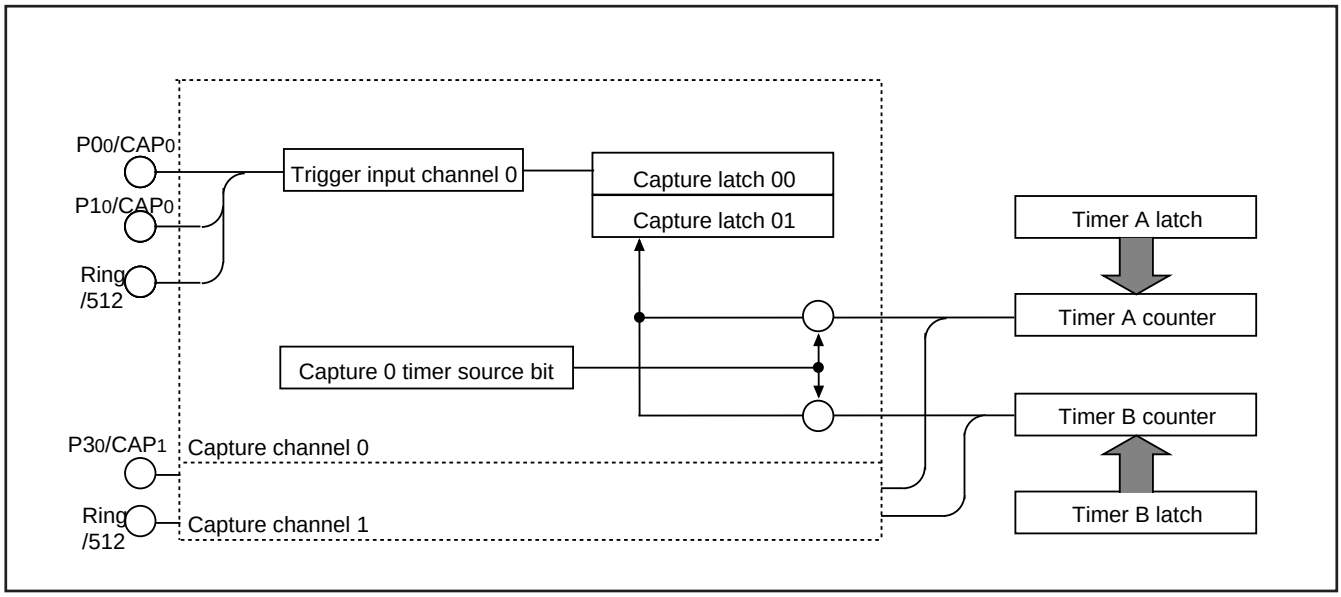



**Fig. 51 Block diagram of input capture**

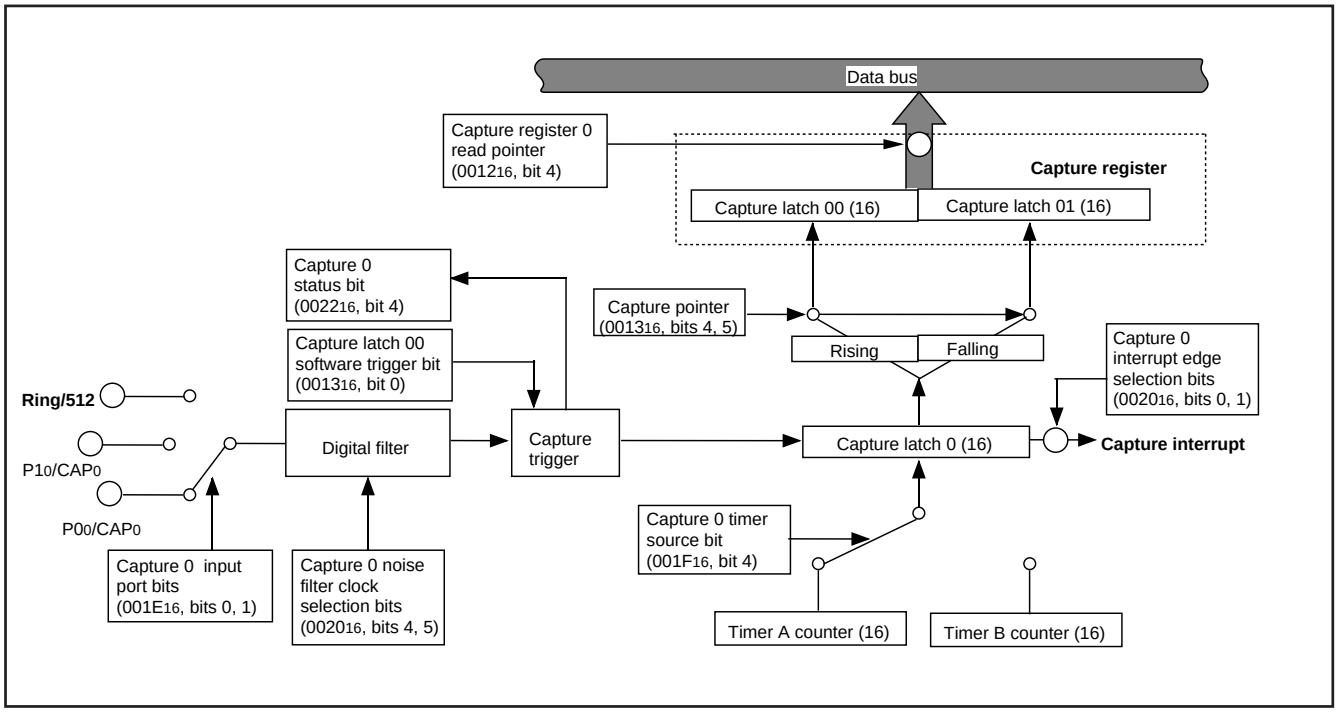



**Fig. 52 Block diagram of capture channel 0**

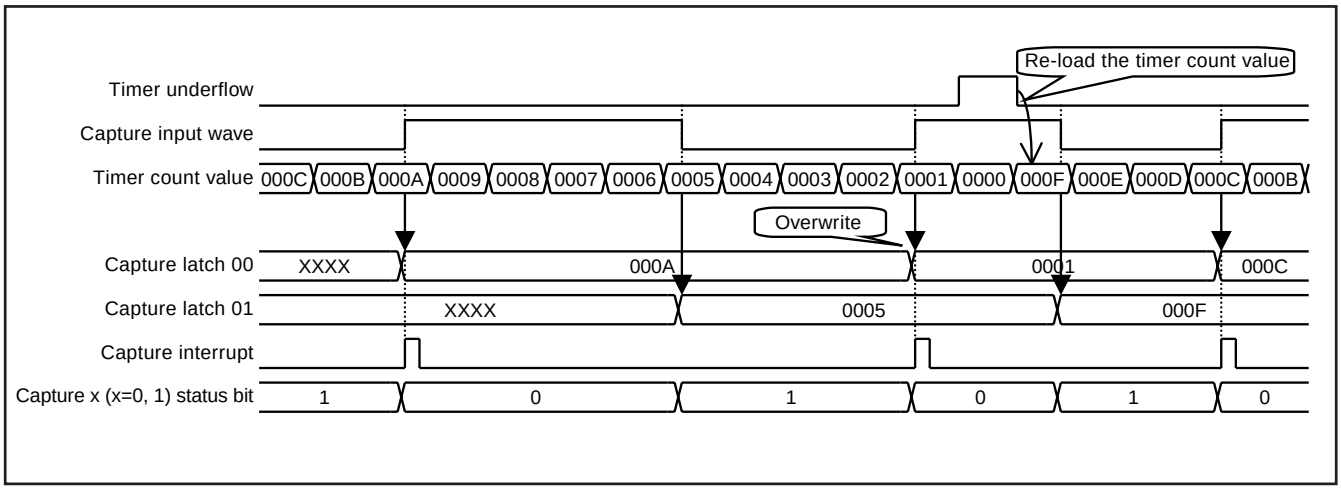



**Fig. 53 Capture interrupt edge selection = “rising edge”**

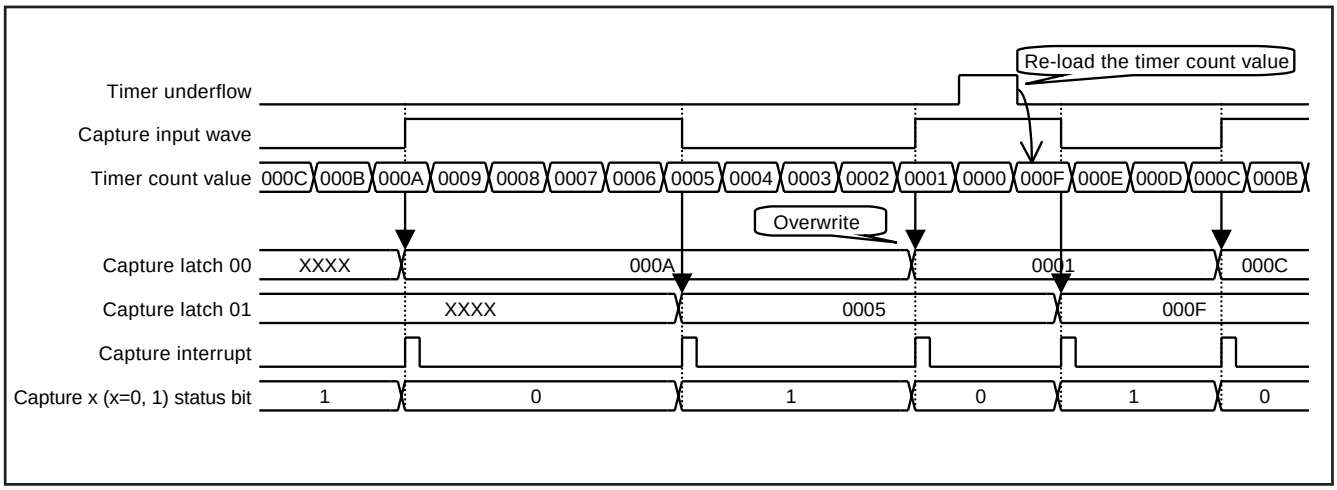



**Fig. 54 Capture interrupt edge selection = “rising and falling edge”**

## Serial Interface

The 7542 Group has Serial I/O1 and Serial I/O2. Except that Serial I/O1 has the bus collision detection function and the $T_XD_2$ output structure for Serial I/O2 is CMOS only, they have the same function.

## ●Serial I/O1

Serial I/O1 can be used as either clock synchronous or asynchronous (UART) serial I/O. A dedicated timer is also provided for baud rate generation.

**(1) Clock Synchronous Serial I/O1 Mode**

Clock synchronous serial I/O1 mode can be selected by setting the serial I/O1 mode selection bit of the serial I/O1 control register (bit 6) to "1".

For clock synchronous serial I/O1, the transmitter and the receiver must use the same clock. If an internal clock is used, transfer is started by a write signal to the TB/RB.

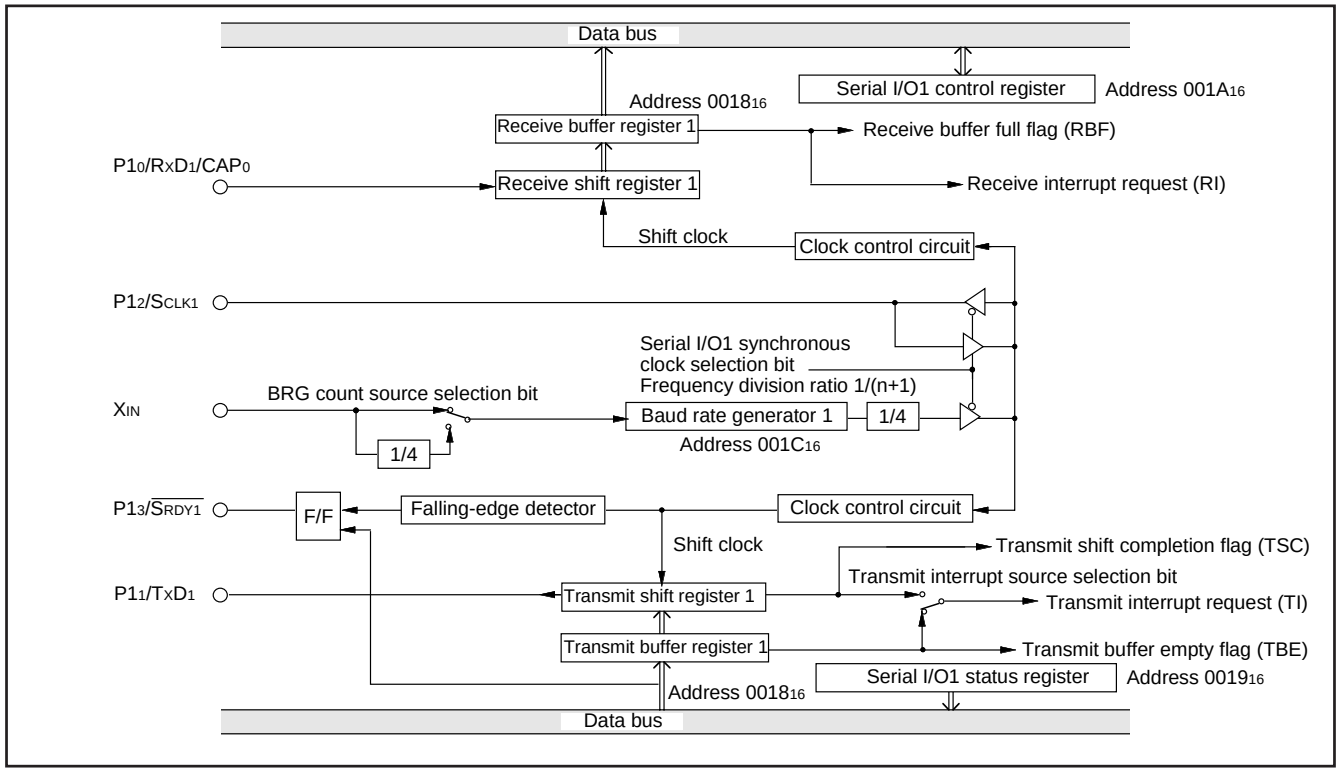
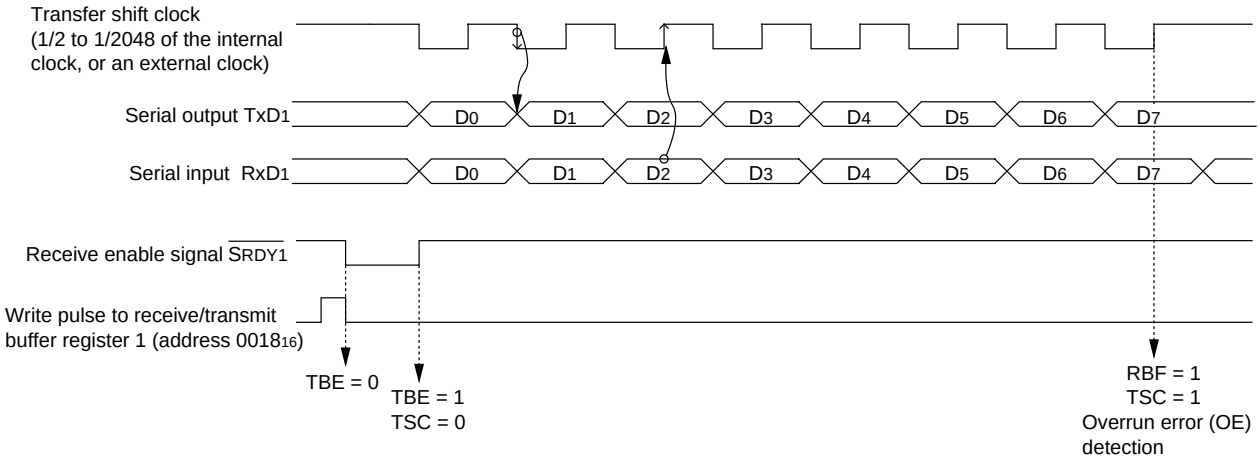



**Fig. 55 Block diagram of clock synchronous serial I/O1**




**Notes 1:** As the transmit interrupt (TI), which can be selected, either when the transmit buffer has emptied (TBE=1) or after the transmit shift operation has ended (TSC=1), by setting the transmit interrupt source selection bit (TIC) of the serial I/O1 control register.

**2:** If data is written to the transmit buffer register when TSC=0, the transmit clock is generated continuously and serial data is output continuously from the $TxD_1$ pin.

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

**Fig. 56 Operation of clock synchronous serial I/O1 function**

**(2) Asynchronous Serial I/O1 (UART) Mode**

Clock asynchronous serial I/O mode (UART) can be selected by clearing the serial I/O1 mode selection bit of the serial I/O1 control register to “0”.

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

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

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

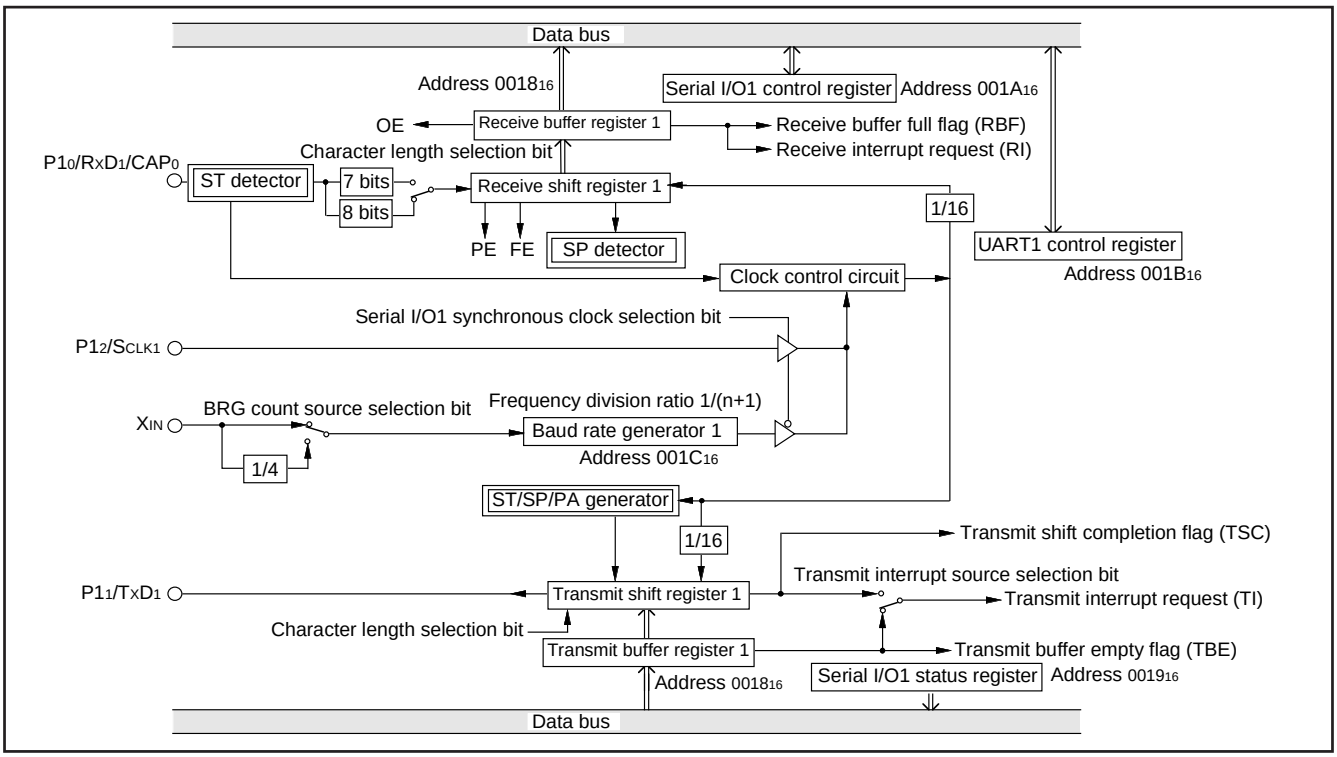
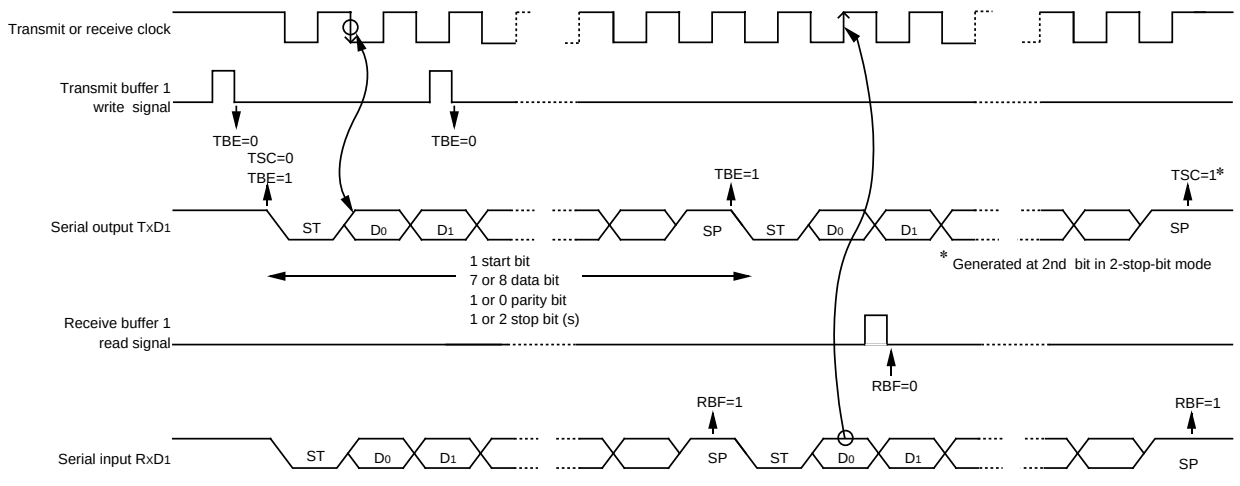



**Fig. 57 Block diagram of UART serial I/O1**




**Notes**
**1:** Error flag detection occurs at the same time that the RBF flag becomes “1” (at 1st stop bit, during reception).
**2:** As the transmit interrupt (TI), when either the TBE or TSC flag becomes “1,” can be selected to occur 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 when TSC=1, 0.5 to 1.5 cycles of the data shift cycle is necessary until changing to TSC=0.

**Fig. 58 Operation of UART serial I/O1 function**

**[Transmit buffer register 1/receive buffer register 1 (TB1/RB1)] 0018$_{16}$**

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

**[Serial I/O1 status register (SIO1STS)] 0019$_{16}$**

The read-only serial I/O1 status register consists of seven flags (bits 0 to 6) which indicate the operating status of the serial I/O1 function and various errors.

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

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

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

Bits 0 to 6 of the serial I/O1 status register are initialized to "0" at reset, but if the transmit enable bit of the serial I/O1 control register has been set to "1", the transmit shift completion flag (bit 2) and the transmit buffer empty flag (bit 0) become "1".

**[Serial I/O1 control register (SIO1CON)] 001A$_{16}$**

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

**[UART1 control register (UART1CON)] 001B$_{16}$**

The UART1 control register consists of four control bits (bits 0 to 3) which are valid when asynchronous serial I/O is selected and set the data format of an data transfer and one bit (bit 4) which is always valid and sets the output structure of the P1$_1$/TxD$_1$ pin.

**[Baud rate generator 1 (BRG1)] 001C$_{16}$**

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 on Serial I/O1

• Serial I/O interrupt

When setting the transmit enable bit to "1", the serial I/O transmit interrupt request bit is automatically set to "1". When not requiring the interrupt occurrence synchronized with the transmission enabled, take the following sequence.

① Set the serial I/O transmit interrupt enable bit to "0" (disabled).
② Set the transmit enable bit to "1".
③ Set the serial I/O transmit interrupt request bit to "0" after 1 or more instructions have been executed.
④ Set the serial I/O transmit interrupt enable bit to "1" (enabled).

• I/O pin function when serial I/O1 is enabled.

The functions of P1$_2$ and P1$_3$ are switched with the setting values of a serial I/O1 mode selection bit and a serial I/O1 synchronous clock selection bit as follows.

(1) Serial I/O1 mode selection bit → "1" :
Clock synchronous type serial I/O is selected.
Setup of a serial I/O1 synchronous clock selection bit
"0" : P1$_2$ pin turns into an output pin of a synchronous clock.
"1" : P1$_2$ pin turns into an input pin of a synchronous clock.
Setup of a $\overline{\text{SRDY1}}$ output enable bit (SRDY)
"0" : P1$_3$ pin can be used as a normal I/O pin.
"1" : P1$_3$ pin turns into a $\overline{\text{SRDY1}}$ output pin.

(2) Serial I/O1 mode selection bit → "0" :
Clock asynchronous (UART) type serial I/O is selected.
Setup of a serial I/O1 synchronous clock selection bit
"0": P1$_2$ pin can be used as a normal I/O pin.
"1": P1$_2$ pin turns into an input pin of an external clock.
When clock asynchronous (UART) type serial I/O is selected, it is P1$_3$ pin. It can be used as a normal I/O pin.

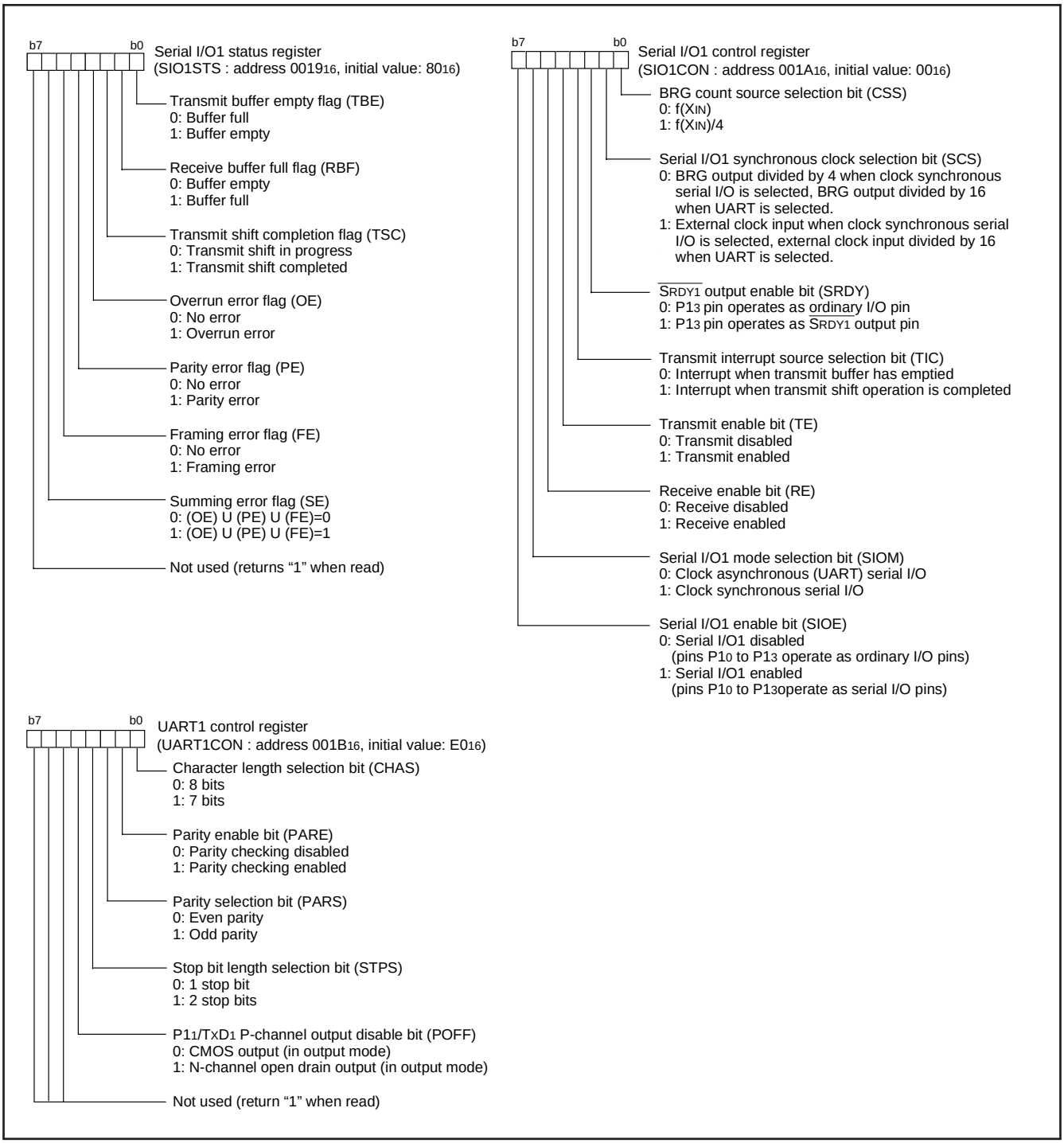



**Fig. 59 Structure of serial I/O1-related registers**

## Bus collision detection (SIO1)

SIO1 can detect a bus collision by setting UART1 bus collision detection interrupt enable bit.

When transmission is started in the clock synchronous or asynchronous (UART) serial I/O mode, the transmit pin $TxD_1$ is compared with the receive pin $RxD_1$ in synchronization with rising edge of transmit shift clock. If they do not coincide with each other, a bus collision detection interrupt request occurs.

When a transmit data collision is detected between LSB and MSB of transmit data in the clock synchronous serial I/O mode or between the start bit and stop bit of transmit data in UART mode, a bus collision detection can be performed by both the internal clock and the external clock.

A block diagram is shown in Fig. 61.
A timing diagram is shown in Fig. 62.

**Note:** Bus collision detection can be used when SIO1 is operating at full-duplex communication. When SIO1 is operating at half-duplex communication, set bus collision detection interrupt to be disabled.

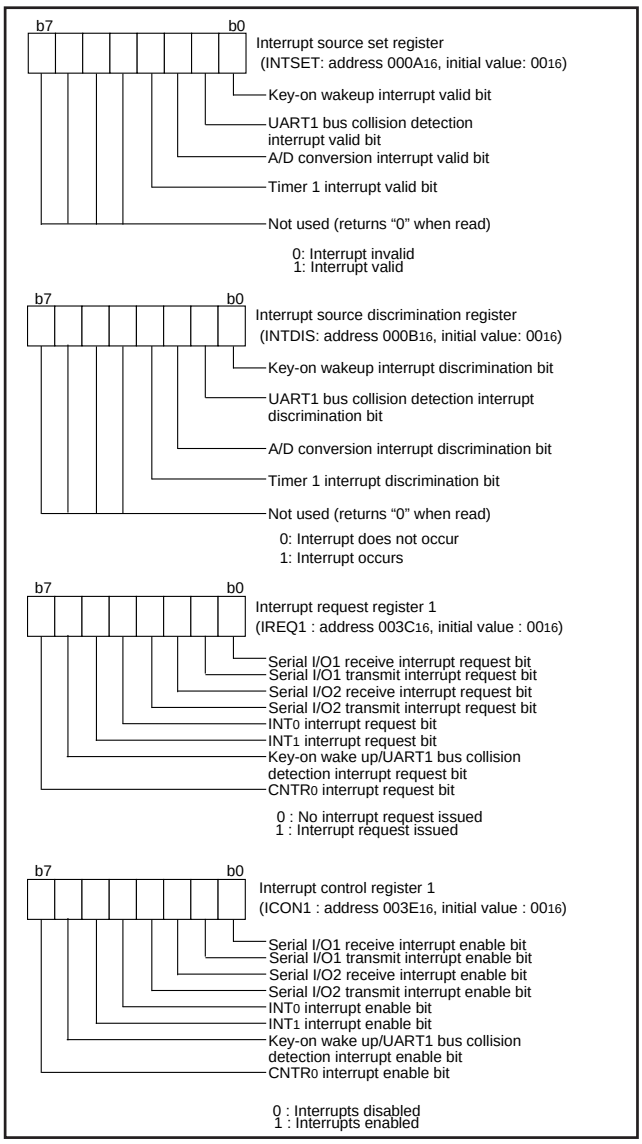



**Fig. 60 Bus collision detection circuit related registers**

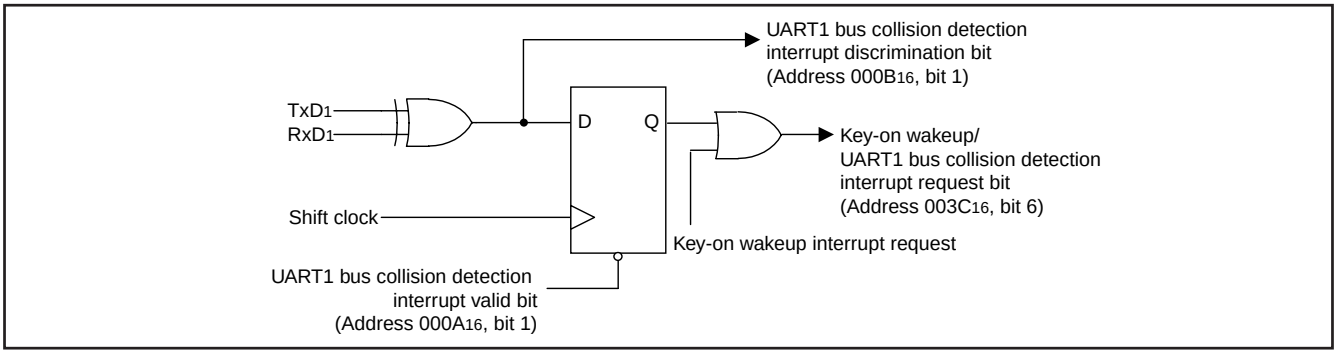



**Fig. 61 Block diagram of bus collision detection interrupt circuit**

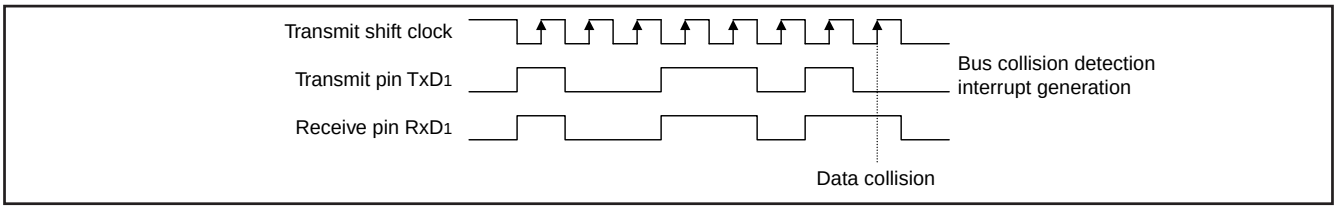



**Fig. 62 Timing diagram of bus collision detection interrupt**

## ●Serial I/O2

Serial I/O2 can be used as either clock synchronous or asynchronous (UART) serial I/O. A dedicated timer is also provided for baud rate generation.

**(1) Clock Synchronous Serial I/O2 Mode**

Clock synchronous serial I/O2 mode can be selected by setting the serial I/O2 mode selection bit of the serial I/O2 control register (bit 6) to "1".

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

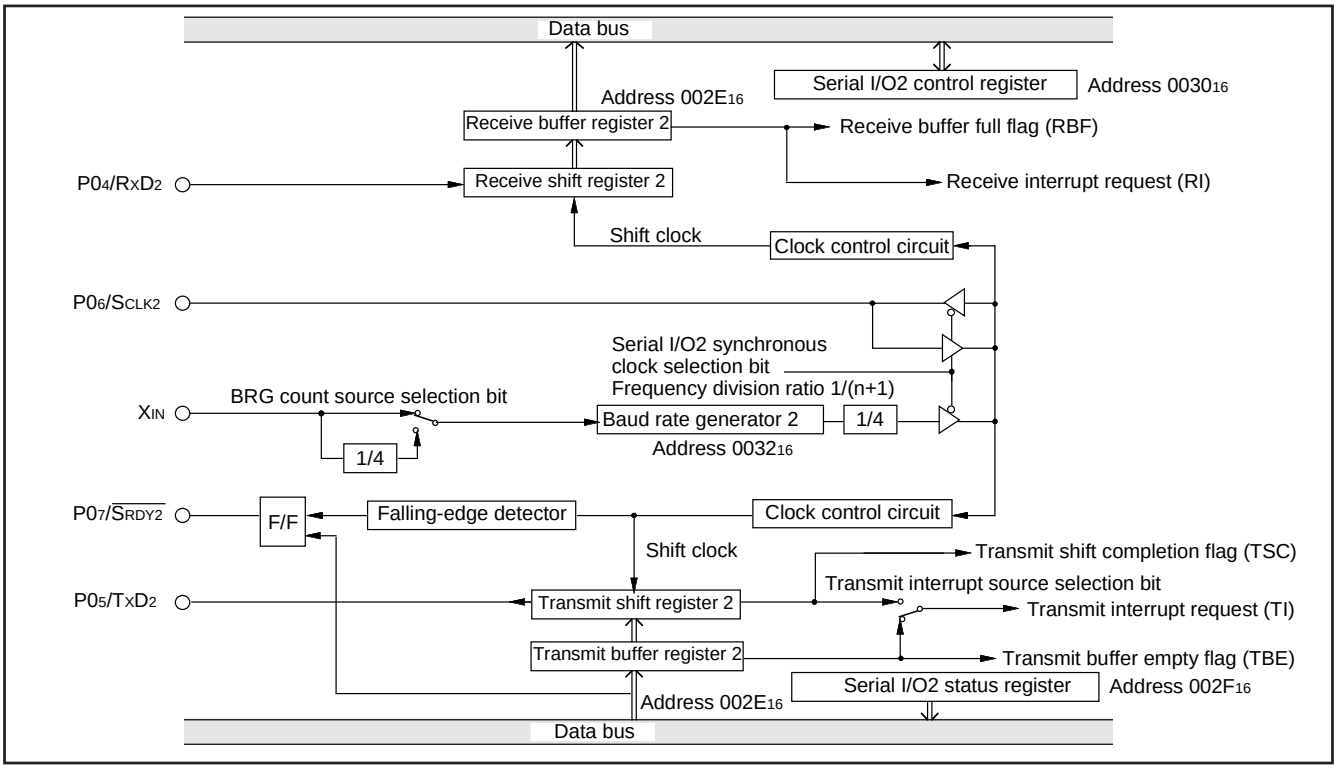
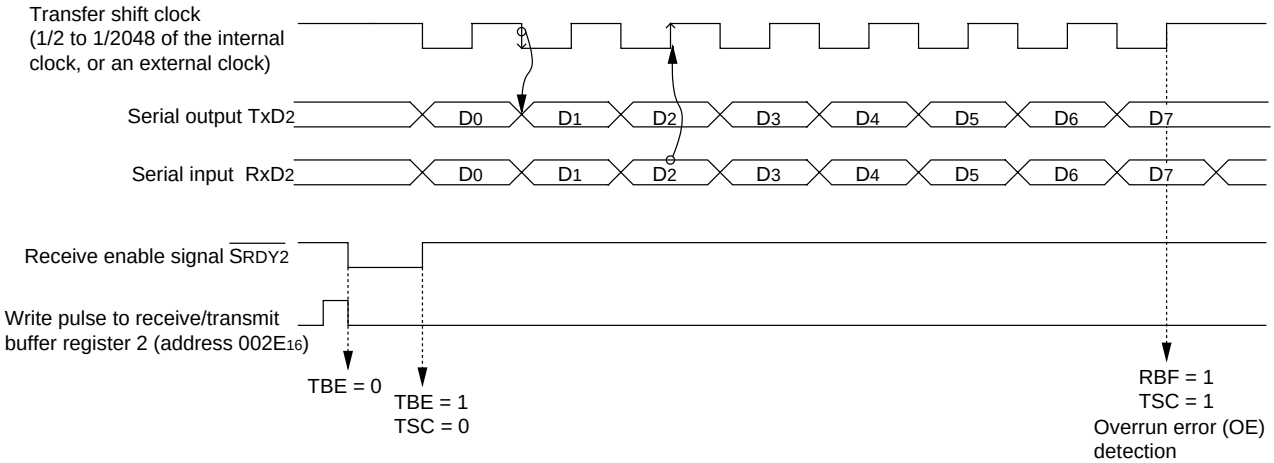



**Fig. 63 Block diagram of clock synchronous serial I/O2**




**Notes 1:** As the transmit interrupt (TI), which can be selected, either when the transmit buffer has emptied (TBE=1) or after the transmit shift operation has ended (TSC=1), by setting the transmit interrupt source selection bit (TIC) of the serial I/O2 control register.

**2:** If data is written to the transmit buffer register when TSC=0, the transmit clock is generated continuously and serial data is output continuously from the $TxD_2$ pin.

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

**Fig. 64 Operation of clock synchronous serial I/O2 function**

**(2) Asynchronous Serial I/O2 (UART) Mode**

Clock asynchronous serial I/O mode (UART) can be selected by clearing the serial I/O2 mode selection bit of the serial I/O2 control register to “0”.

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

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

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

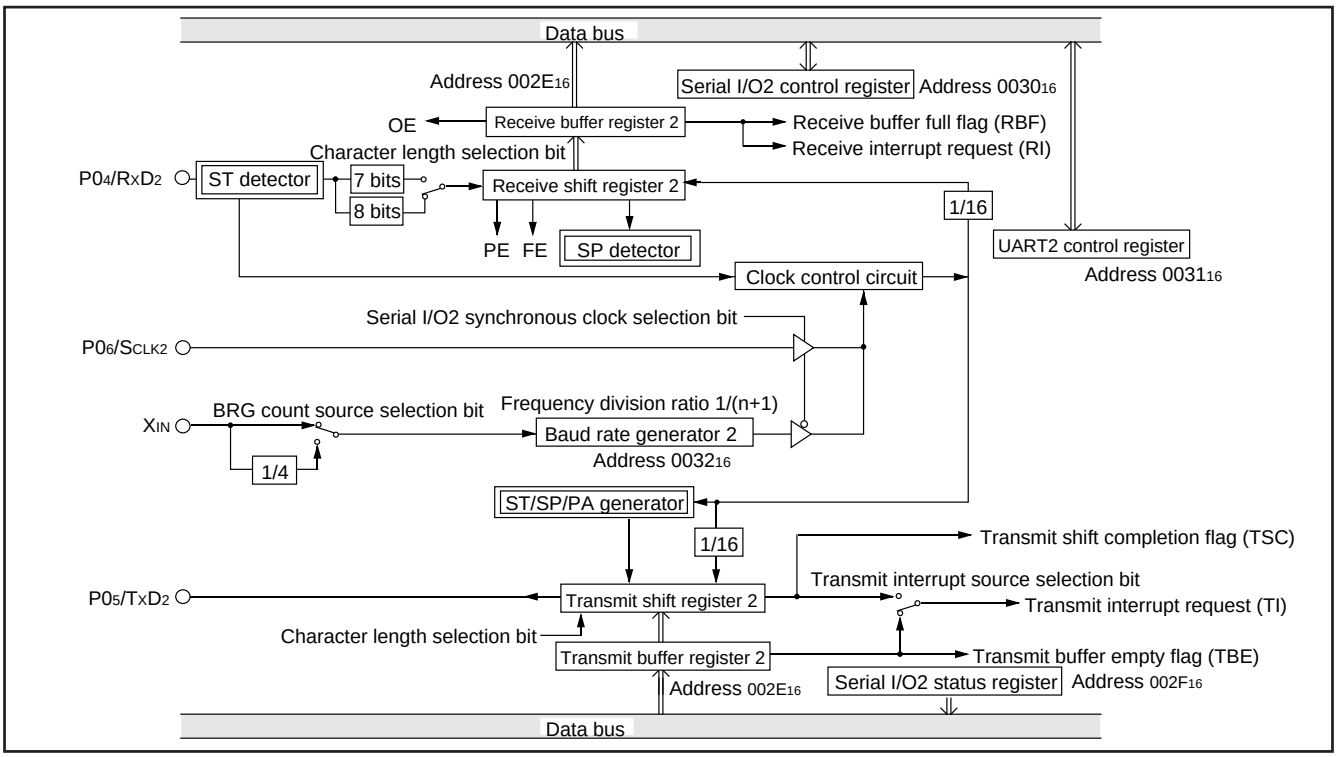
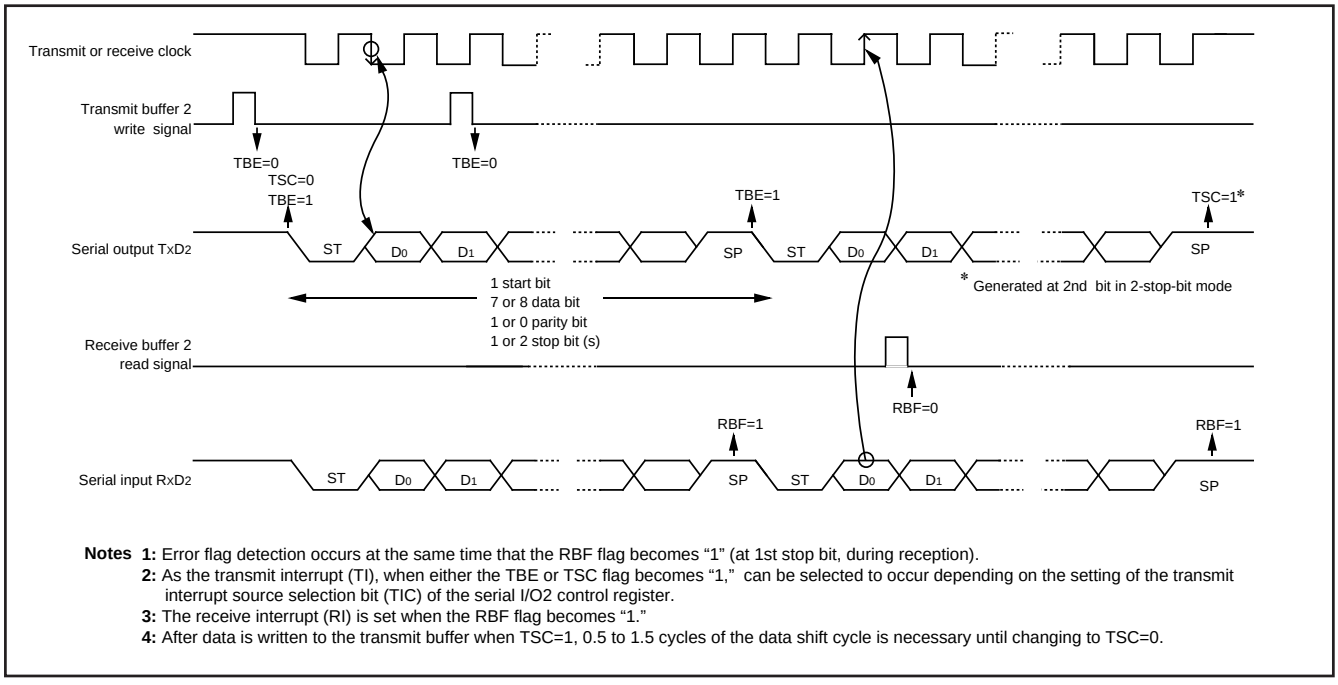



**Fig. 65 Block diagram of UART serial I/O2**




**Notes**
**1:** Error flag detection occurs at the same time that the RBF flag becomes “1” (at 1st stop bit, during reception).
**2:** As the transmit interrupt (TI), when either the TBE or TSC flag becomes “1,” can be selected to occur depending on the setting of the transmit interrupt source selection bit (TIC) of the serial I/O2 control register.
**3:** The receive interrupt (RI) is set when the RBF flag becomes “1.”
**4:** After data is written to the transmit buffer when TSC=1, 0.5 to 1.5 cycles of the data shift cycle is necessary until changing to TSC=0.

**Fig. 66 Operation of UART serial I/O2 function**

**[Transmit buffer register 2/receive buffer register 2 (TB2/RB2)] 002E16**

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

**[Serial I/O2 status register (SIO2STS)] 002F16**

The read-only serial I/O2 status register consists of seven flags (bits 0 to 6) which indicate the operating status of the serial I/O2 function and various errors.

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

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

If there is an error, it is detected at the same time that data is transferred from the receive shift register to the receive buffer register, and the receive buffer full flag is set. A write to the serial I/O1 status register clears all the error flags OE, PE, FE, and SE (bit 3 to bit 6, respectively). Writing "0" to the serial I/O2 enable bit SIOE (bit 7 of the serial I/O2 control register) also clears all the status flags, including the error flags.

Bits 0 to 6 of the serial I/O2 status register are initialized to "0" at reset, but if the transmit enable bit of the serial I/O2 control register has been set to "1", the transmit shift completion flag (bit 2) and the transmit buffer empty flag (bit 0) become "1".

**[Serial I/O2 control register (SIO2CON)] 003016**

The serial I/O2 control register consists of eight control bits for the serial I/O2 function.

**[UART2 control register (UART2CON)] 003116**

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.

**[Baud rate generator 2 (BRG2)] 003216**

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 on Serial I/O2

• Serial I/O interrupt

When setting the transmit enable bit to "1", the serial I/O transmit interrupt request bit is automatically set to "1". When not requiring the interrupt occurrence synchronized with the transmission enabled, take the following sequence.

① Set the serial I/O transmit interrupt enable bit to "0" (disabled).
② Set the transmit enable bit to "1".
③ Set the serial I/O transmit interrupt request bit to "0" after 1 or more instructions have been executed.
⑤ Set the serial I/O transmit interrupt enable bit to "1" (enabled).

• I/O pin function when serial I/O2 is enabled.

The functions of P06 and P07 are switched with the setting values of a serial I/O2 mode selection bit and a serial I/O2 synchronous clock selection bit as follows.

(1) Serial I/O2 mode selection bit → "1" :
Clock synchronous type serial I/O is selected.
Setup of a serial I/O2 synchronous clock selection bit
"0" : P06 pin turns into an output pin of a synchronous clock.
"1" : P06 pin turns into an input pin of a synchronous clock.
Setup of a $\overline{\text{SRDY2}}$ output enable bit (SRDY)
"0" : P07 pin can be used as a normal I/O pin.
"1" : P07 pin turns into a $\overline{\text{SRDY2}}$ output pin.

(2) Serial I/O2 mode selection bit → "0" :
Clock asynchronous (UART) type serial I/O is selected.
Setup of a serial I/O2 synchronous clock selection bit
"0": P06 pin can be used as a normal I/O pin.
"1": P06 pin turns into an input pin of an external clock.
When clock asynchronous (UART) type serial I/O is selected, it is P07 pin. It can be used as a normal I/O pin.

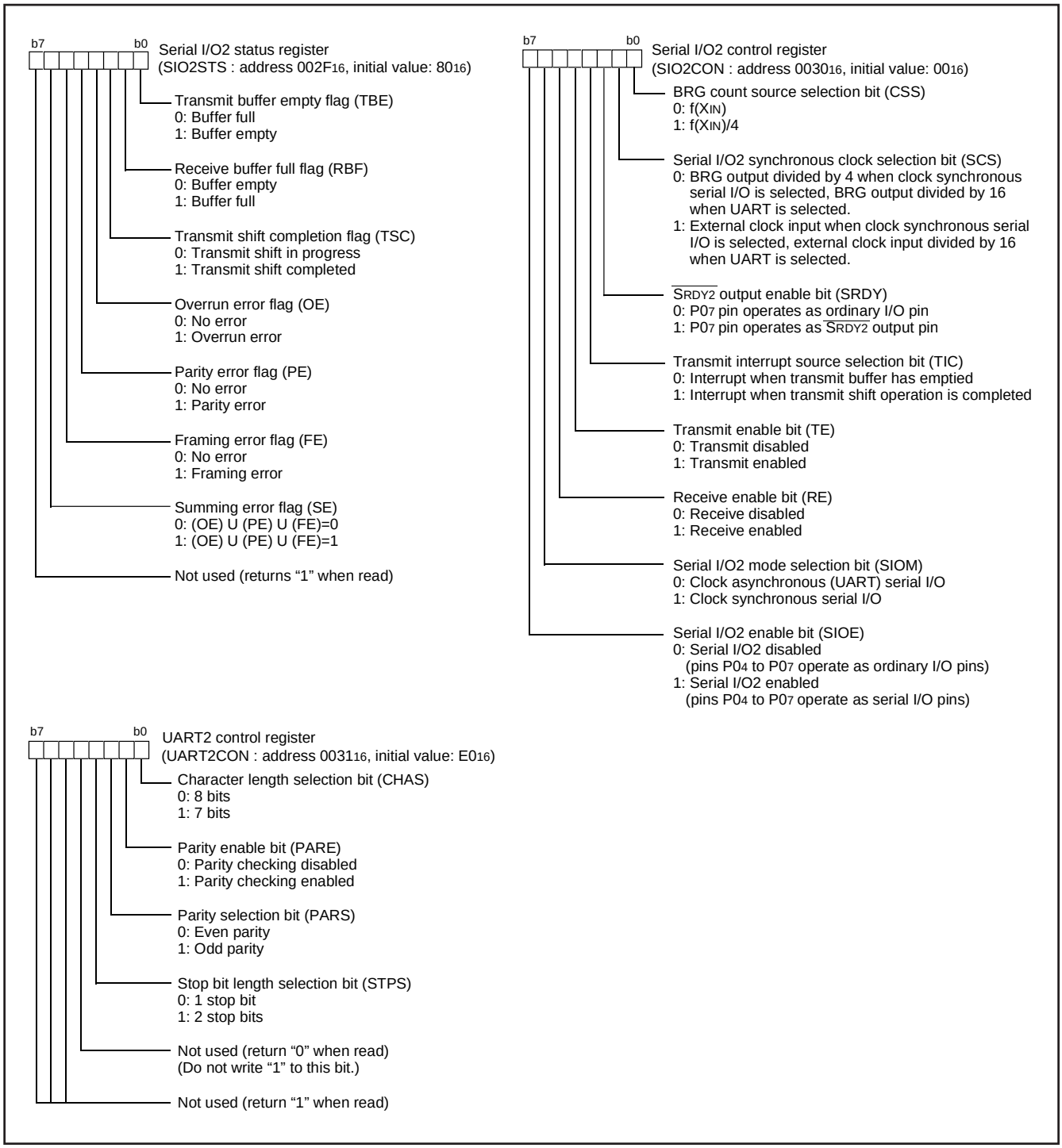



**Fig. 67 Structure of serial I/O2-related registers**

## 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 3 is the A/D conversion clock selection bit. When “0” is set to this bit, the A/D conversion clock is f(XIN)/2 and the A/D conversion time is 122 cycles of f(XIN). When “1” is set to this bit, the A/D conversion clock is f(XIN) and the A/D conversion time is 61 cycles of f(XIN).

Bit 4 is the A/D conversion completion bit. The value of this bit remains at “0” during A/D conversion, and changes to “1” at completion of A/D conversion.

A/D conversion is started by setting this bit to “0”.

**[Comparison voltage generator]**

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

**[Channel selector]**

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

**[Comparator and control circuit]**

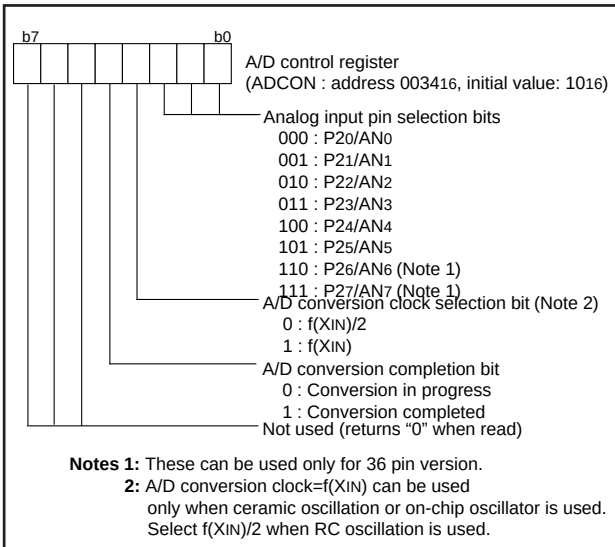
The comparator and control circuit compares an analog input voltage with the comparison voltage and stores its result into the A/D conversion register. When A/D conversion is completed, the control circuit sets the A/D conversion completion bit and the A/D interrupt request bit to “1”. Because the comparator is constructed linked to a capacitor, set f(XIN) in order that the A/D conversion clock is 250 kHz or over during A/D conversion.

**■ Notes on A/D converter**

As for AD translation accuracy, on the following operating conditions, accuracy may become low.

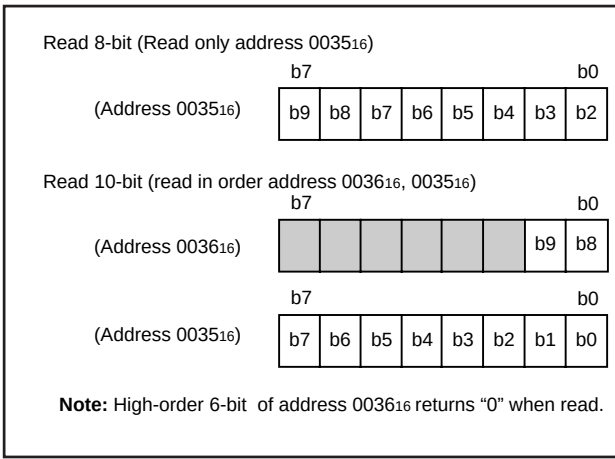
(1) Since the analog circuit inside a microcomputer becomes sensitive to noise when VREF voltage is set up lower than Vcc voltage, accuracy may become low rather than the case where VREF voltage and Vcc voltage are set up to the same value..

(2) When VREF voltage is lower than [ 3.0 V ], the accuracy at the low temperature may become extremely low compared with that at room temperature. When the system would be used at low temperature, the use at VREF=3.0 V or more is recommended.




**Notes 1:** These can be used only for 36 pin version.
**2:** A/D conversion clock=f(XIN) can be used only when ceramic oscillation or on-chip oscillator is used. Select f(XIN)/2 when RC oscillation is used.

**Fig. 68 Structure of A/D control register**




**Note:** High-order 6-bit of address 003616 returns “0” when read.

**Fig. 69 Structure of A/D conversion register**

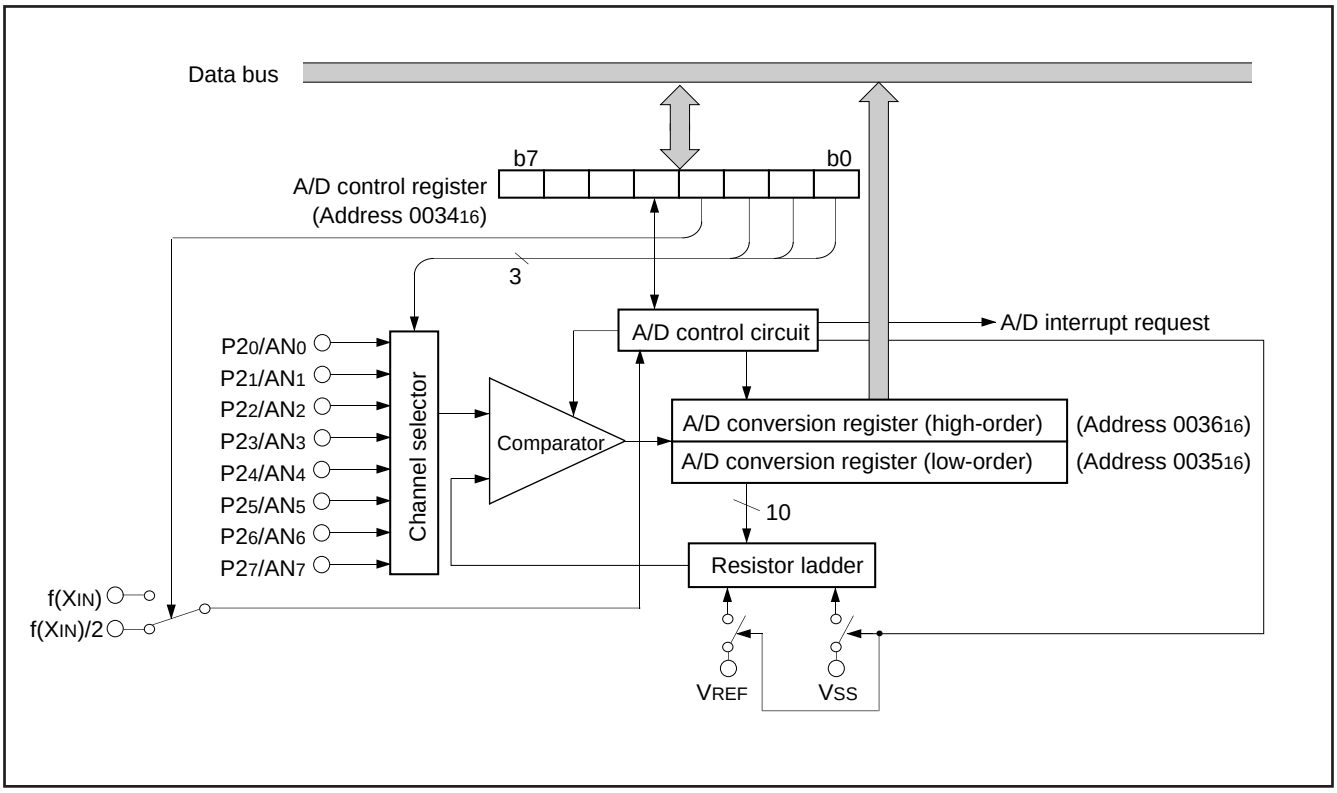



**Fig. 70 Block diagram of A/D converter**

## 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 function selection 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 131.072 ms at f(XIN)=8 MHz.
When this bit is "1", the count source becomes f(XIN)/16. In this case, the detection time is 512 μs at f(XIN)=8 MHz.
This bit is cleared to "0" after reset.

**Operation of STP instruction function selection bit**

When "0" is set to STP instruction function selection bit, system enters into the stop mode at the STP instruction execution.
When "1" is set to this bit, internal reset occurs at the STP instruction execution.
This bit is set to "1" by program, but it cannot be changed to "0" .
This bit is cleared to "0" after reset.

## ■ Notes on Watchdog Timer

1. The watchdog timer is operating during the wait mode. Write data to the watchdog timer control register to prevent timer underflow.

2. The watchdog timer stops during the stop mode. However, the watchdog timer is running during the oscillation stabilizing time after the **STP** instruction is released. In order to avoid the underflow of the watchdog timer, the watchdog timer control register must be written just before executing the **STP** instruction.

3. The **STP** instruction function selection bit (bit 6 of watchdog timer control register (address $0039_{16}$)) can be rewritten only once after releasing reset. After rewriting it is disable to write any data to this bit.

4. A count source of watchdog timer is affected by the clock division selection bit of the CPU mode register.
   The f(XIN) clock is supplied to the watchdog timer when selecting f(XIN) as the CPU clock.
   The on-chip oscillator output is supplied to the watchdog timer when selecting the on-chip oscillator output as the CPU clock.

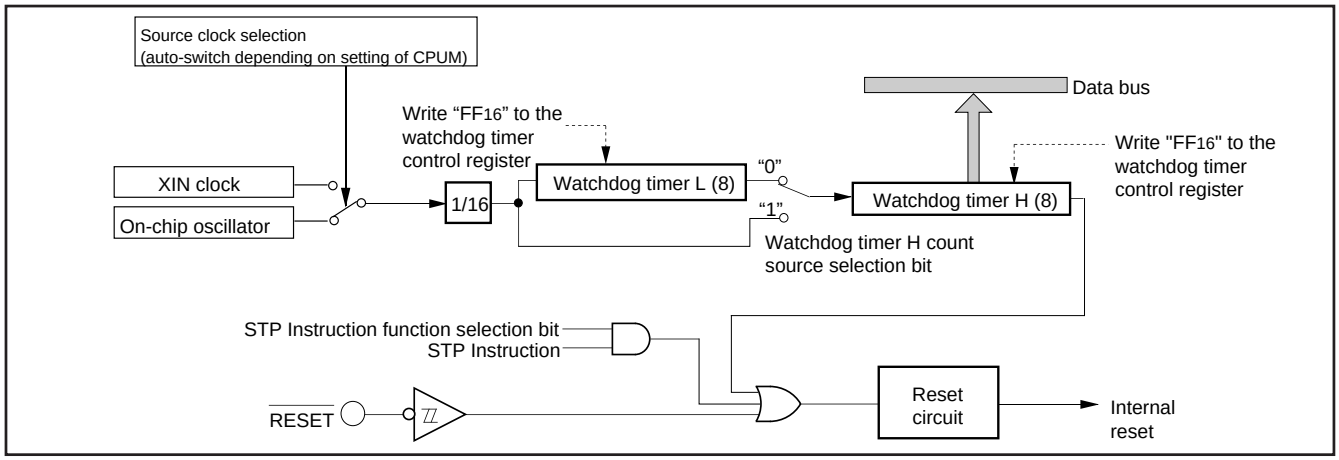



**Fig. 71 Block diagram of watchdog timer**

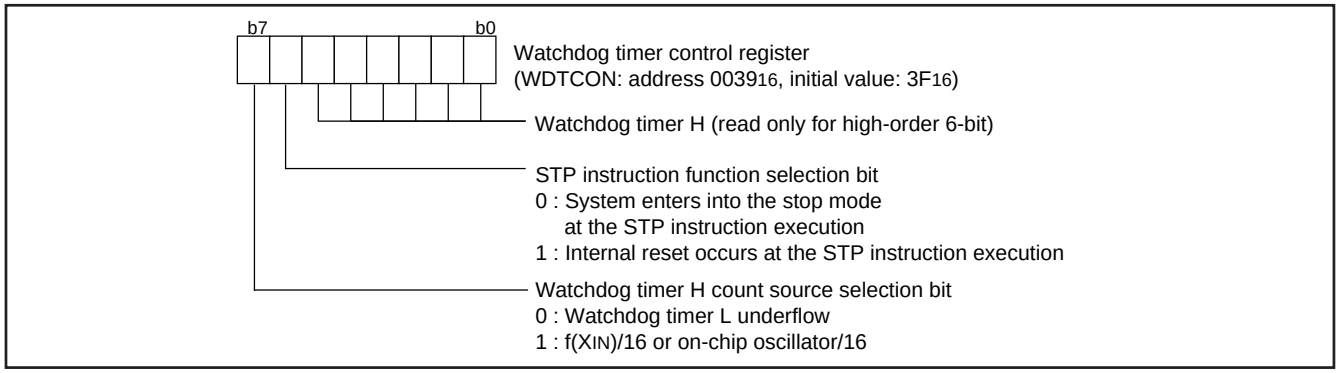



**Fig. 72 Structure of watchdog timer control register**

## Reset Circuit

The 7542 group starts operation by the on-chip oscillator after system is released from reset.
Accordingly, when the rising of power supply voltage passes 2.2V, set the reset input voltage to become below 0.2Vcc (0.44V).
Moreover, switch CPU clock to the external oscillator after the rising of power supply voltage passes the minimum operation voltage and after an oscillation is stabilized.

Note: The minimum operation voltage is decided by the division ratio of an external oscillator's frequency and a CPU clock. Decide on an external oscillator's oscillation stabilizing time after fully evaluating an oscillator's stabilizing time used.

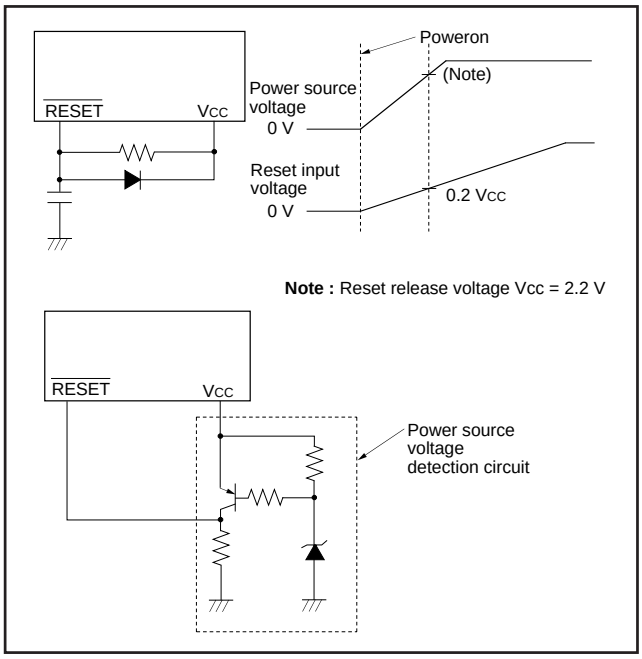



**Fig. 73 Example of reset circuit**

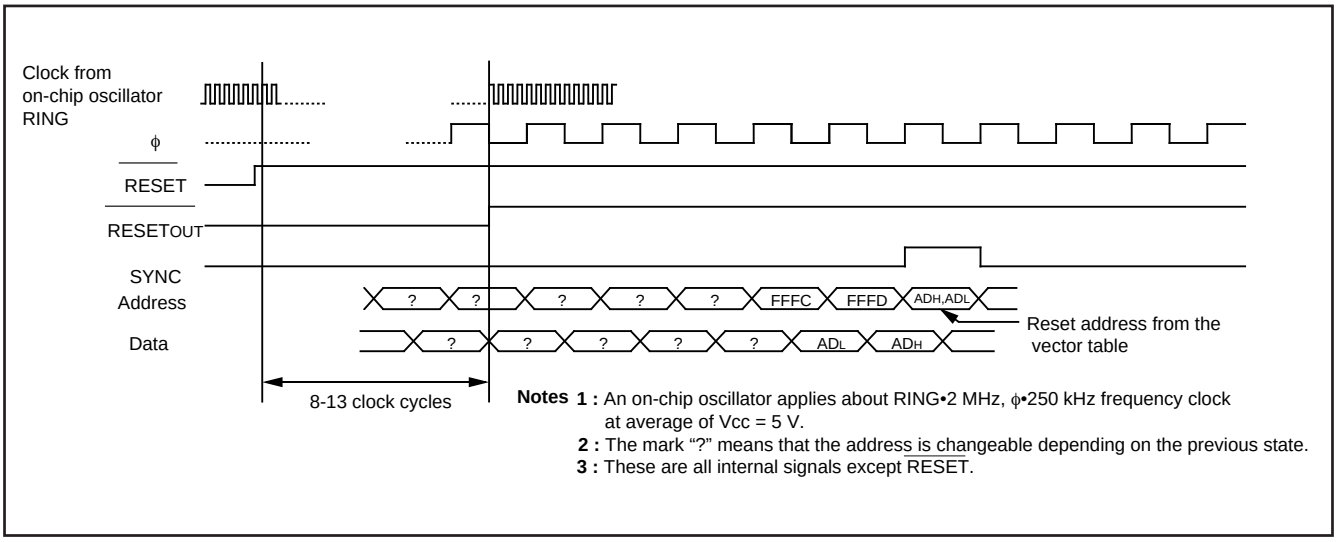



**Fig. 74 Timing diagram at reset**

| | Address | Register contents |
|---|---|---|
| (1) Port P0 direction register (P0D) | 0001₁₆ | 00₁₆ |
| (2) Port P1 direction register (P1D) | 0003₁₆ | X X X 0 0 0 0 0 |
| (3) Port P2 direction register (P2D) | 0005₁₆ | 00₁₆ |
| (4) Port P3 direction register (P3D) | 0007₁₆ | 00₁₆ |
| (5) Interrupt source set register (INTSET) | 000A₁₆ | 00₁₆ |
| (6) Interrupt source discrimination register (INTDIS) | 000B₁₆ | 00₁₆ |
| (7) Compare register (low-order) (CMPL) | 0010₁₆ | 00₁₆ |
| (8) Compare register (high-order) (CMPH) | 0011₁₆ | 00₁₆ |
| (9) Capture/Compare register R/W pointer (CCRP) | 0012₁₆ | 00₁₆ |
| (10) Capture software trigger register (CSTR) | 0013₁₆ | 00₁₆ |
| (11) Compare register re-load register (CMPR) | 0014₁₆ | 00₁₆ |
| (12) Port P0P3 drive capacity control register (DCCR) | 0015₁₆ | 00₁₆ |
| (13) Pull-up control register (PULL) | 0016₁₆ | 00₁₆ |
| (14) Port P1P3 control register (P1P3C) | 0017₁₆ | 00₁₆ |
| (15) Serial I/O1 status register (SIO1STS) | 0019₁₆ | 1 0 0 0 0 0 0 0 |
| (16) Serial I/O1 control register (SIO1CON) | 001A₁₆ | 00₁₆ |
| (17) UART1 control register (UART1CON) | 001B₁₆ | 1 1 1 0 0 0 0 0 |
| (18) Timer A, B mode register (TABM) | 001D₁₆ | 00₁₆ |
| (19) Capture/Compare port register (CCPR) | 001E₁₆ | 00₁₆ |
| (20) Timer source selection register (TMSR) | 001F₁₆ | 00₁₆ |
| (21) Capture mode register (CAPM) | 0020₁₆ | 00₁₆ |
| (22) Compare output mode register (CMOM) | 0021₁₆ | 00₁₆ |
| (23) Capture/Compare status register (CCSR) | 0022₁₆ | 00₁₆ |
| (24) Compare interrupt source register (CISR) | 0023₁₆ | 00₁₆ |
| (25) Timer A (low-order) (TAL) | 0024₁₆ | FF₁₆ |
| (26) Timer A (high-order) (TAH) | 0025₁₆ | FF₁₆ |
| (27) Timer B (low-order) (TBL) | 0026₁₆ | FF₁₆ |
| (28) Timer B (high-order) (TBH) | 0027₁₆ | FF₁₆ |
| (29) Prescaler 1 (PRE1) | 0028₁₆ | FF₁₆ |
| (30) Timer 1 (T1) | 0029₁₆ | 01₁₆ |
| (31) Timer count source set register (TCSS) | 002A₁₆ | 00₁₆ |
| (32) Timer X mode register (TXM) | 002B₁₆ | 00₁₆ |
| (33) Prescaler X (PREX) | 002C₁₆ | FF₁₆ |
| (34) Timer X (TX) | 002D₁₆ | FF₁₆ |
| (35) Serial I/O2 status register (SIO2STS) | 002F₁₆ | 1 0 0 0 0 0 0 0 |
| (36) Serial I/O2 control register (SIO2CON) | 0030₁₆ | 00₁₆ |
| (37) UART2 control register (UART2CON) | 0031₁₆ | 1 1 1 0 0 0 0 0 |
| (38) A/D control register (ADCON) | 0034₁₆ | 0 0 0 1 0 0 0 0 |
| (39) On-chip oscillation division ratio selection register (RODR) | 0037₁₆ | 0 0 0 0 0 0 1 0 |
| (40) MISRG | 0038₁₆ | 00₁₆ |
| (41) Watchdog timer control register (WDTCON) | 0039₁₆ | 0 0 1 1 1 1 1 1 |
| (42) Interrupt edge selection register (INTEDGE) | 003A₁₆ | 00₁₆ |
| (43) CPU mode register (CPUM) | 003B₁₆ | 1 0 0 0 0 0 0 0 |
| (44) Interrupt request register 1 (IREQ1) | 003C₁₆ | 00₁₆ |
| (45) Interrupt request register 2 (IREQ2) | 003D₁₆ | 00₁₆ |
| (46) Interrupt control register 1 (ICON1) | 003E₁₆ | 00₁₆ |
| (47) Interrupt control register 2 (ICON2) | 003F₁₆ | 00₁₆ |
| (48) Flash memory control register 0 (FMCR0) (Note 3) | 0FE0₁₆ | 0 0 0 0 0 0 0 1 |
| (49) Flash memory control register 1 (FMCR1) (Note 3) | 0FE1₁₆ | 0 1 0 0 0 0 0 0 |
| (50) Flash memory control register 2 (FMCR2) (Note 3) | 0FE2₁₆ | 0 0 0 0 0 0 0 1 |
| (51) Processor status register | (PS) | X X X X X 1 X X |
| (52) Program counter | ($PC_H$) | Contents of address FFFD₁₆ |
| | ($PC_L$) | Contents of address FFFC₁₆ |

**Notes 1:** X : Undefined
**2:**The content of other registers is undefined when the microcomputer is reset. The initial values must be surely set before you use it.
**3:**Only flash memory version has this register.

**Fig. 75 Internal status of microcomputer at reset**

## Clock Generating Circuit

An oscillation circuit can be formed by connecting a resonator between XIN and XOUT, and an RC oscillation circuit can be formed by connecting a resistor and a capacitor.
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.)

**(1) On-chip oscillator operation**

When the MCU operates by the on-chip oscillator for the main clock, connect XIN pin to VCC through a resistor and leave XOUT pin open.
The clock frequency of the on-chip oscillator depends on the supply voltage and the operation temperature range.
Be careful that variable frequencies when designing application products.

**(2) Ceramic resonator**

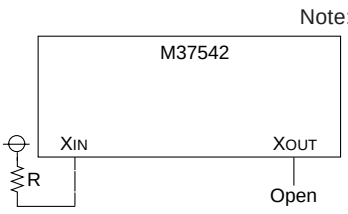
When the ceramic resonator is used for the main clock, connect the ceramic resonator and the external circuit to pins XIN and XOUT at the shortest distance. A feedback resistor is built in between pins XIN and XOUT.

**(3) RC oscillation**

When the RC oscillation is used for the main clock, connect the XIN pin and XOUT pin to the external circuit of resistor R and the capacitor C at the shortest distance.
The frequency is affected by a capacitor, a resistor and a microcomputer.
So, set the constants within the range of the frequency limits.

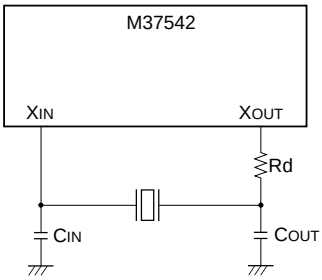
**(4) External clock**

When the external signal clock is used for the main clock, connect the XIN pin to the clock source and leave XOUT pin open.
Select “0” (ceramic oscillation) to oscillation mode selection bit of CPU mode register ($003B_{16}$).



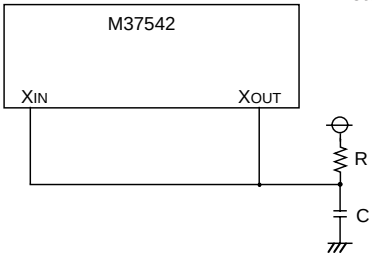

Note: The clock frequency of the on-chip oscillator depends on the supply voltage and the operation temperature range.
Be careful that variable frequencies and obtain the sufficient margin.

**Fig. 76 Processing of XIN and XOUT pins at on-chip oscillator operation**




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. 77 External circuit of ceramic resonator**




Note: Connect the external circuit of resistor R and the capacitor C at the shortest distance.
The frequency is affected by a capacitor, a resistor and a microcomputer.
So, set the constants within the range of the frequency limits.

**Fig. 78 External circuit of RC oscillation**

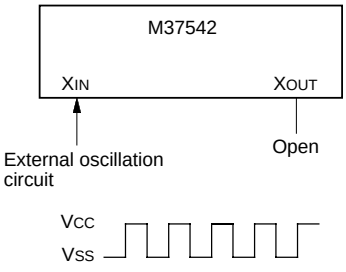



**Fig. 79 External clock input circuit**

**(1) 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 1 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 1 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 1. 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, or set the wait time by on-chip oscillator operation after system is released from reset until the oscillation is stabled.

With the FLASH version, the internal power supply circuit is changed to low power consumption mode for consumption current reduction at the time of STP instruction execution.

Although an internal power supply circuit is usually changed to the normal operation mode at the time of the return from an STP instruction, since a certain time is required to start the power supply to FLASH and operation of FLASH to be enabled, set wait time 100 μs or more with the FLASH version by the oscillation stabilization time set function after release of the STP instruction which used the timer 1.

**• 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.

## ■ Notes on Clock Generating Circuit

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

• Switch of ceramic and RC oscillations

After releasing reset the operation starts by starting an on-chip oscillator. Then, a ceramic oscillation or an RC oscillation is selected by setting bit 5 of the CPU mode register.

• Double-speed mode

When a ceramic oscillation is selected, a double-speed mode can be used. Do not use it when an RC oscillation is selected.

• CPU mode register

Bits 5, 1 and 0 of CPU mode register are used to select oscillation mode and to control operation modes of the microcomputer. In order to prevent the dead-lock by error-writing (ex. program run-away), these bits can be rewritten only once after releasing reset. After rewriting it is disable to write any data to the bit. (The emulator MCU "M37542RSS" is excluded.)

Also, when the read-modify-write instructions (SEB, CLB) are executed to bits 2 to 4, 6 and 7, bits 5, 1 and 0 are locked.

• Clock division ratio, XIN oscillation control, on-chip oscillator control

The state transition shown in Fig. 84 can be performed by setting the clock division ratio selection bits (bits 7 and 6), XIN oscillation control bit (bit 4), on-chip oscillator oscillation control bit (bit 3) of CPU mode register. Be careful of notes on use in Fig. 84.

- Count source (Timer 1, Timer A, Timer B, Timer X, Serial I/O, Serial I/O2, A/D converter, Watchdog timer)
  A count source of watchdog timer is affected by the clock division selection bit of the CPU mode register.
  The f(XIN) clock is supplied to the watchdog timer when selecting f(XIN) as the CPU clock.
  The on-chip oscillator output is supplied to the watchdog timer when selecting the on-chip oscillator output as the CPU clock.

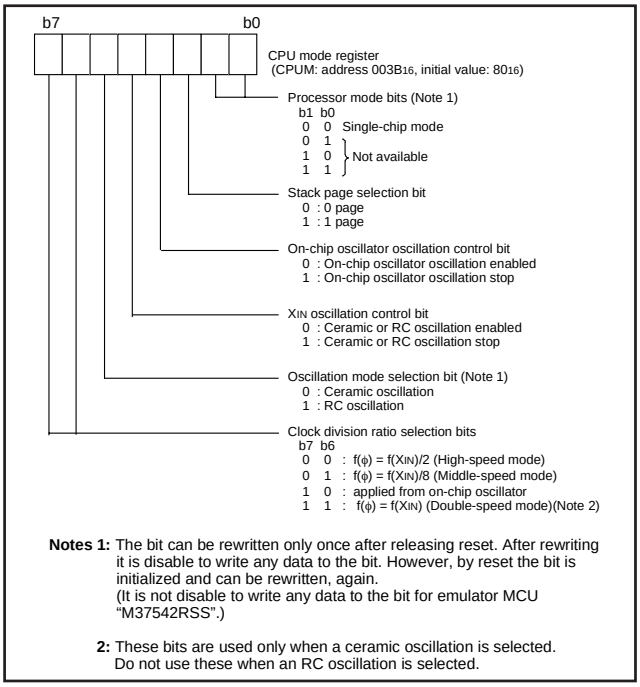



**Fig. 80 Structure of CPU mode register**

## ● On-chip oscillation division ratio

At on-chip oscillator mode, division ratio of on-chip oscillator for CPU clock is selected by setting value of on-chip oscillation division ratio selection register. The division ratio of on-chip oscillation for CPU clock is selected from among 1/1, 1/2, 1/8, 1/128. The operation clock for the peripheral function block is not changed by setting value of this register.

### ■ Notes on On-chip Oscillation Division Ratio

- When system is released from reset, ROSC/8 (on-chip oscillator middle-speed mode) is selected for CPU clock.
- When state transition from the ceramic or RC oscillation to on-chip oscillator, ROSC/8 (on-chip oscillator middle-speed mode) is selected for CPU clock.
- When the MCU operates by on-chip oscillator for the main clock without external oscillation circuit, connect XIN pin to VCC through a resistor and leave XOUT pin open.
  Set "10010x00$_2$" (x = 0 or 1) to CPUM.

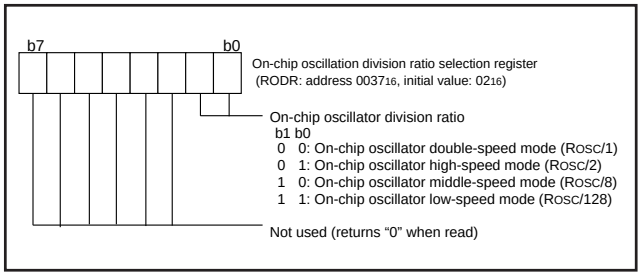



**Fig. 81 Structure of on-chip oscillation division ratio selection register**

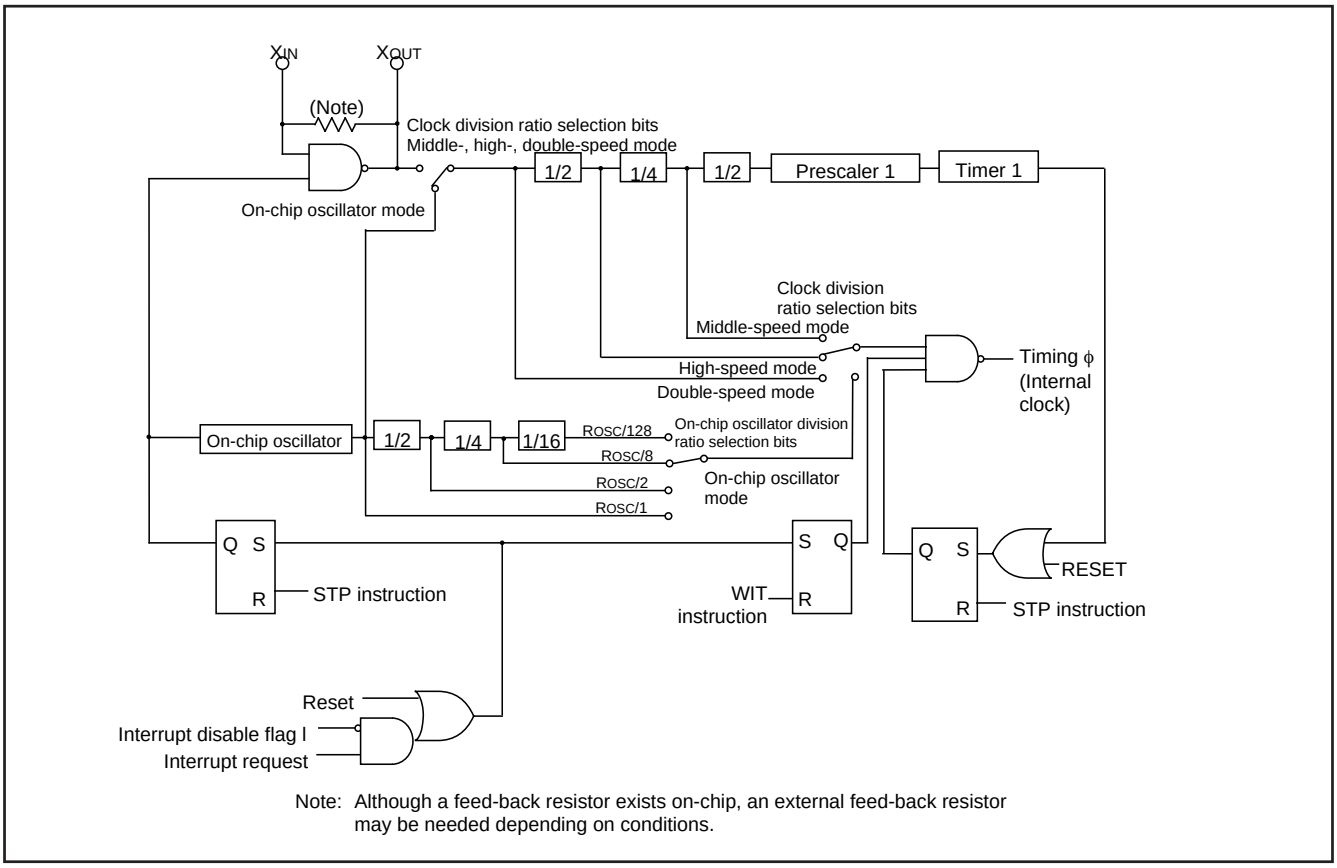



**Fig. 82 Block diagram of internal clock generating circuit (for ceramic resonator)**

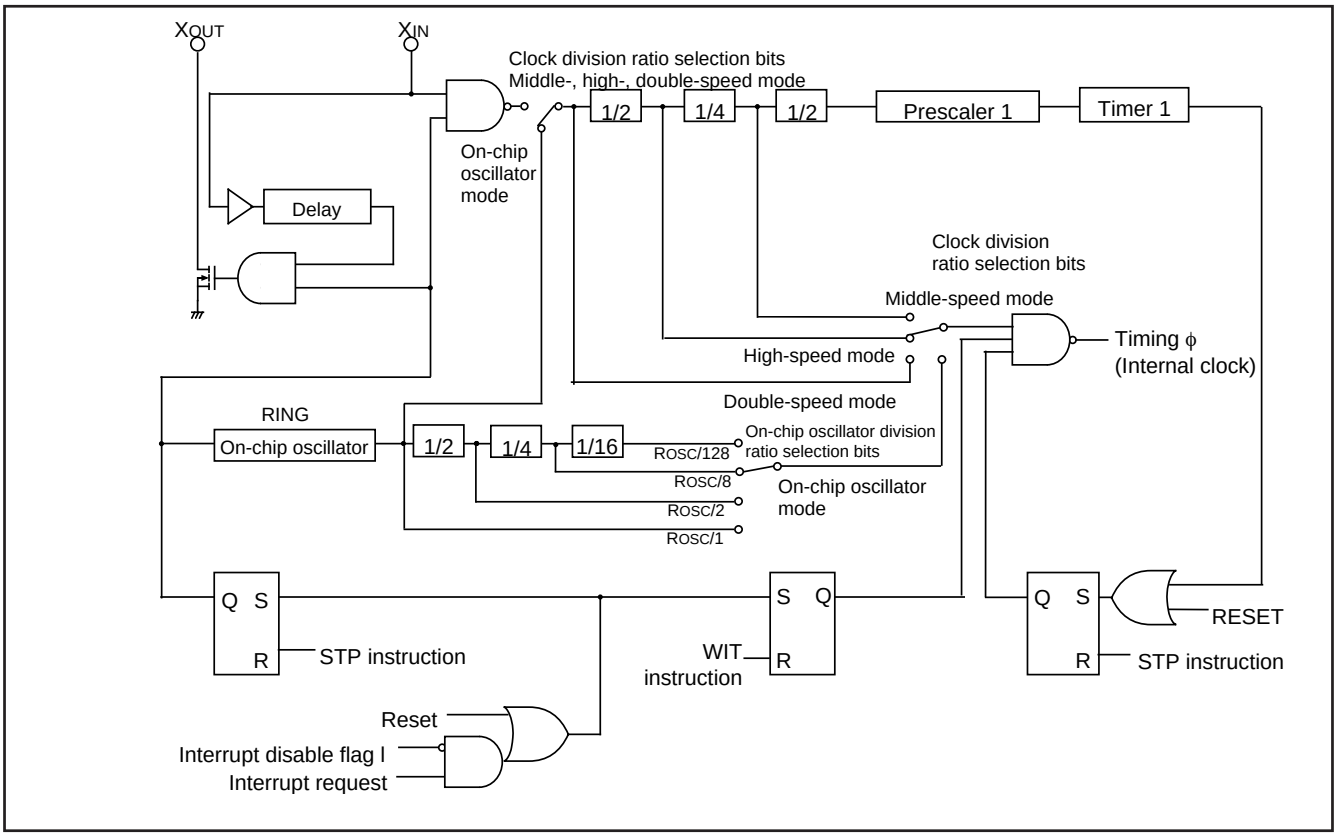
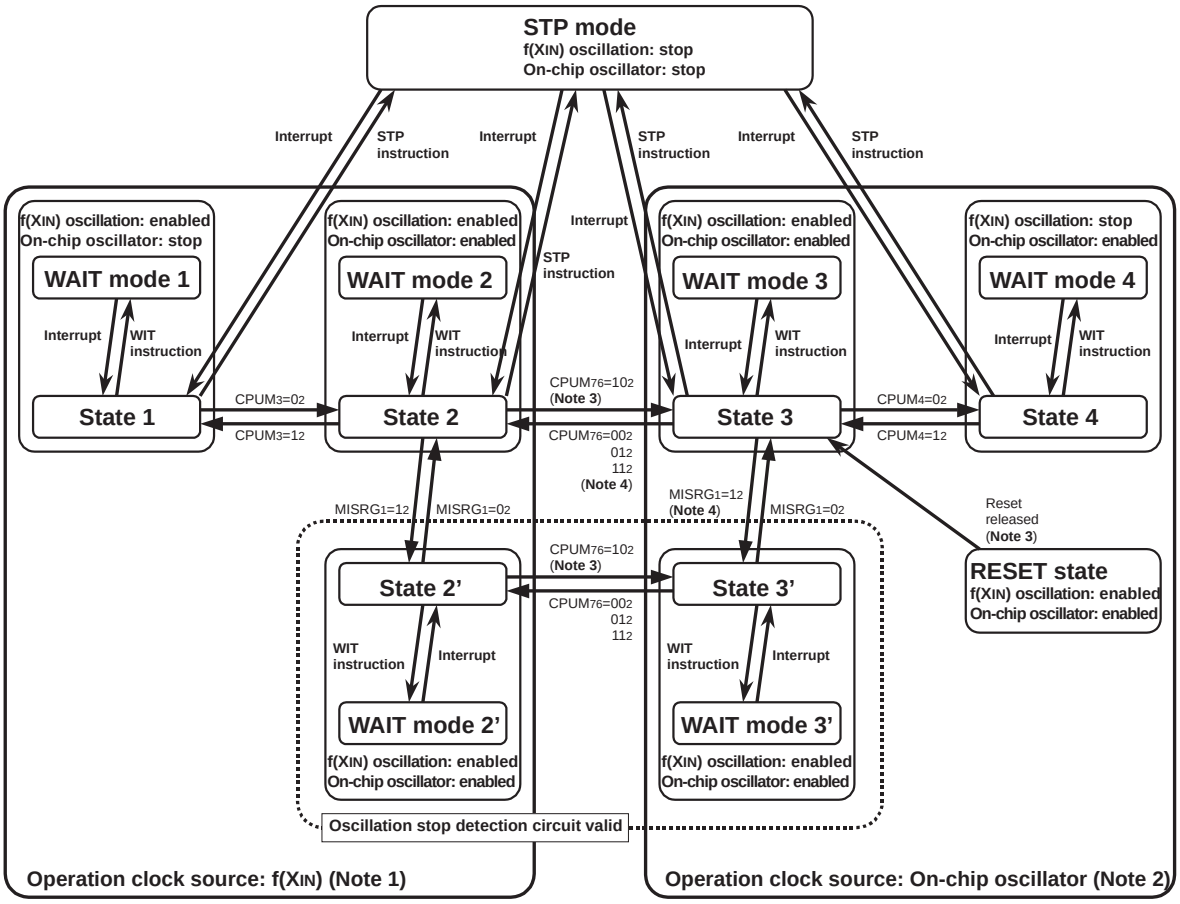



**Fig. 83 Block diagram of internal clock generating circuit (for RC oscillation)**




**Notes on switch of clock**

(1) In operation clock = f(XIN), the following can be selected for the CPU clock division ratio.
f(XIN)/2 (high-speed mode)
f(XIN)/8 (middle-speed mode)
f(XIN) (double-speed mode, only at a ceramic oscillation)

(2) In operation clock = On-chip oscillator, the following can be selected for the CPU clock division ratio.
ROSC/1 (On-chip oscillator double-speed mode)
ROSC/2 (On-chip oscillator high-speed mode)
ROSC/8 (On-chip oscillator middle-speed mode)
ROSC/128 (On-chip oscillator low-speed mode)

(3) After system is released from reset, and state transition of state 2 → state 3 and state transition of state 2' → state 3', ROSC/8 (On-chip oscillator middle-speed mode) is selected for CPU clock.

(4) Executing the state transition state 3 to 2 or state 3 to 3' after stabilizing XIN oscillation.

(5) When the state 2 → state 3 → state 4 is performed, execute the NOP instruction as shown below according to the division ratio of CPU clock.
1. CPUM76 = 102 (state 2 → state 3)
2. NOP instruction
Transition from Double-speed mode: NOP × 3
Transition from High-speed mode: NOP × 1
Transition from Middle-speed mode: NOP × 0
3. CPU4 = 12 (state 3 → state 4)

(6) When the state 3 → state 2 → state 1 is performed, execute the NOP instruction as shown below according to the division ratio of CPU clock.
1. CPUM76 = 002 or 012 or 112 (state 3 → state 2)
2. NOP instruction
Transition from On-chip oscillator double-speed mode: NOP × 4
Transition from On-chip oscillator high-speed mode: NOP × 2
Transition from On-chip oscillator middle-speed mode: NOP × 0
Transition from On-chip oscillator low-speed mode: NOP × 0
3. CPUM3 = 12 (state 2 → state 1)

**Fig. 84 State transition**

## ● Oscillation stop detection circuit

The oscillation stop detection circuit is used to detect an oscillation stop when a ceramic resonator or oscillation circuit stops due to disconnection. To use the oscillation stop detection circuit, set the on-chip oscillator to start operating.

The oscillation stop detection circuit is enabled by setting the Ceramic or RC oscillation stop detection function active bit to 1. While this circuit is enabled, the operating status of the Ceramic or RC oscillation circuit is monitored using the on-chip oscillator. If an oscillation stop is detected, the oscillation stop detection status bit is set to 1. If the oscillation stop detection reset enable bit is also set to 1, an internal reset is triggered at oscillation stop detection.

The Ceramic or RC oscillation stop detection function active bit and the oscillation stop detection status bit are not initialized if an oscillation stop detection reset is triggered and these bits retain their value of 1. Since these bits are initialized to 0 by an external reset, an oscillation stop detection reset can be determined by checking the oscillation stop status bit.

The oscillation stop detection status bit is set to 0 by writing 0 to the Ceramic or RC oscillation stop detection function active bit.

To enable the oscillation detection circuit, first write 0 to the Ceramic or RC oscillation stop detection function active bit and set the oscillation stop detection status bit to 0. Then set the Ceramic or RC oscillation stop detection function active bit to 1.

The Ceramic oscillation, RC oscillation, and external clock input are set as the clocks for oscillation stop detection. Refer to the electrical characteristics for the frequencies for oscillation stop detection.

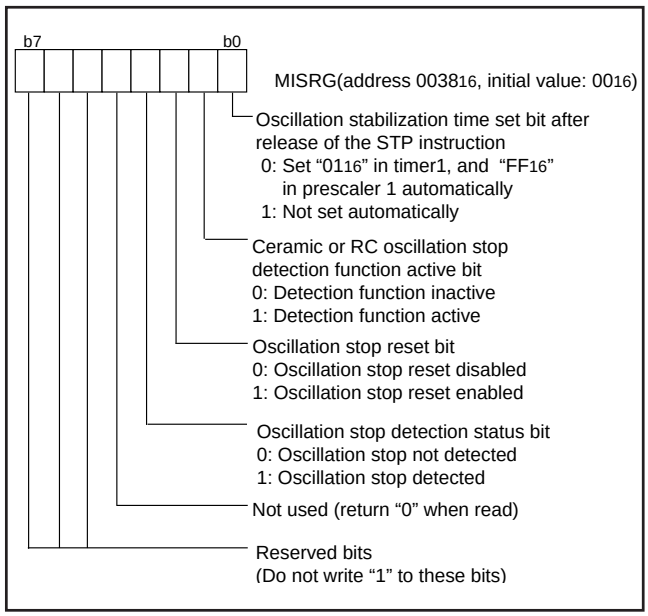



**Fig. 85 Structure of MISRG**

### ■ Notes on Oscillation Stop Detection Circuit

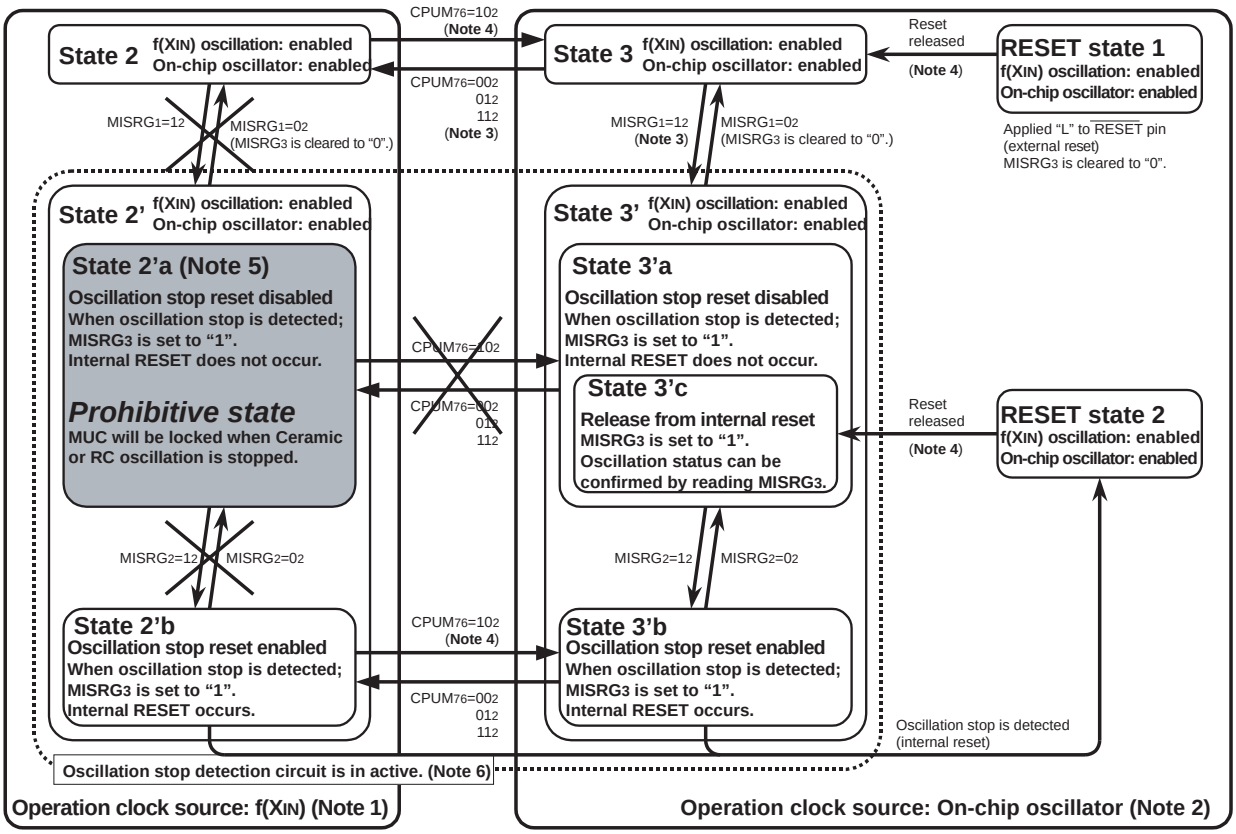
(1) Do not execute the transition to "state 2'a" shown in Figure 86 State transition of oscillation stop detection circuit. In this state, no reset is triggered and the MCU is stopped even when the XIN oscillation is stopped.

(2) After an oscillation stop detection reset, if this reset is enabled while bits Ceramic or RC oscillation stop detection function active and oscillation stop detection status are retained, a reset is triggered again.

(3) The oscillation stop detection status bit is initialized under the following conditions:
- External reset, power-on reset, low-voltage detection reset, watchdog timer reset, and reset by the STP instruction function.
- Write 0 to the Ceramic or RC oscillation stop detection function active Bit.

(4) While the oscillation stop detection function is in active, the oscillation stop detection status bit may set to 1 when the watchdog timer underflow.
When an oscillation stop detection reset is triggered, reconfirm that oscillation is stopped.

(5) The oscillation stop detection circuit is not included in the emulator MCU "M37542RSS".




**Notes on switch of clock**

(1) In operation clock = f(XIN), the following can be selected for the CPU clock division ratio.
f(XIN)/2 (High-speed mode)
f(XIN)/8 (Middle-speed mode)
f(XIN) (Double-speed mode, only at a ceramic oscillation)

(2) In operation clock = On-chip oscillator, the following can be selected for the CPU clock division ratio.
ROSC/1 (On-chip oscillator double-speed mode)
ROSC/2 (On-chip oscillator high-speed mode)
ROSC/8 (On-chip oscillator middle-speed mode)
ROSC/128 (On-chip oscillator low-speed mode)

(3) Executing the state transition state 3 to 2 or state 3 to 3' after stabilizing XIN oscillation.

(4) After system is released from reset, and state transition of state 2 → state 3 and state transition of state 2' → state 3', ROSC/8 (On-chip oscillator middle-speed mode) is selected for CPU clock.

(5) MCU cannot be returned by On-chip oscillator and its operation is stopped since internal reset does not occur at oscillation stop detected. Accordingly, do not execute the transition to state 2'a.

(6) STP instruction cannot be used when oscillation stop detection circuit is in active.

**Fig. 86 State transition 2**

# 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.

## 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.

## A/D Conversion

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

## Instruction Execution Timing

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

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

## CPU Mode Register

The oscillation mode selection bit and processor mode bits can be rewritten only once after releasing reset. However, after rewriting it is disable to write any value to the bit. (Emulator MCU is excluded.)

When a ceramic oscillation is selected, a double-speed mode of the clock division ratio selection bits can be used. Do not use it when an RC oscillation is selected.

## State transition

Do not stop the clock selected as the operation clock because of setting of CM3, 4.

# NOTES ON HARDWARE

## 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 capacitor of 0.01 μF to 0.1 μF is recommended.

# 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 *

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

## NOTES ON USE

## Countermeasures against noise

**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.

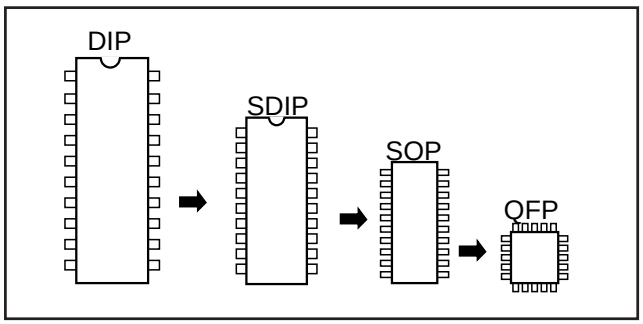



**Fig. 87 Selection of packages**

(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.

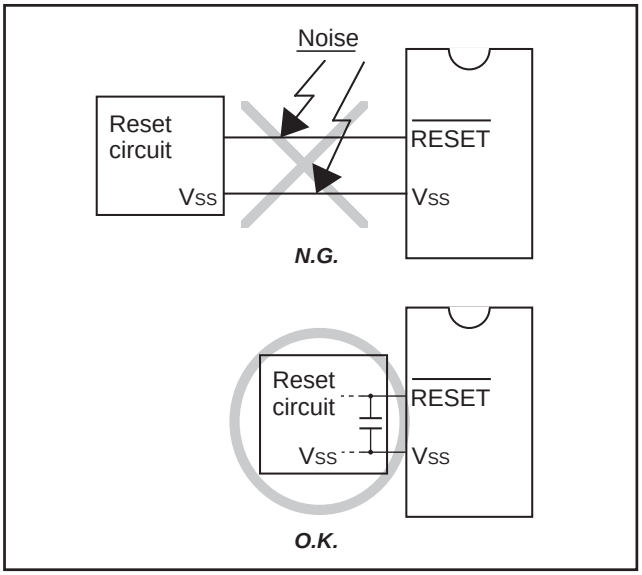



**Fig. 88 Wiring for the $\overline{\text{RESET}}$ pin**

(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 20 mm) 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.

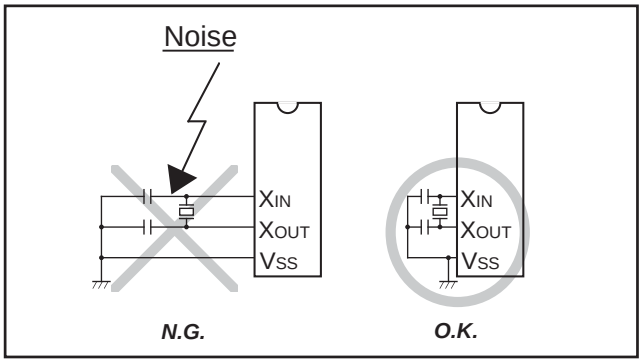



**Fig. 89 Wiring for clock I/O pins**

(4) Wiring to CNVss pin

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

In the normal microcomputer mode, disconnect a wiring of a serial rewrite circuit, which is for the flash memory version, from the MCU by a jumper switch.

<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.

A wiring of a serial rewrite circuit may function as an antenna which feeds noise into the microcomputer.

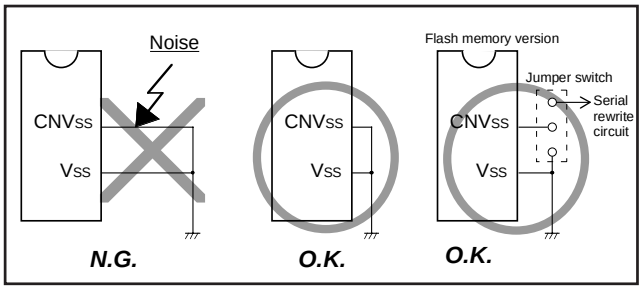



**Fig. 90 Wiring for CNVss pin**

2. Connection of bypass capacitor across Vss line and Vcc line
Connect an approximately 0.1 $\mu$F bypass capacitor across the Vss line and the Vcc line as follows:

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

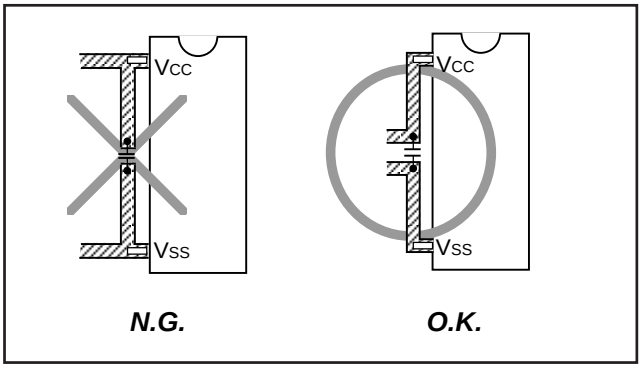



**Fig. 91 Bypass capacitor across the Vss line and the Vcc line**

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.

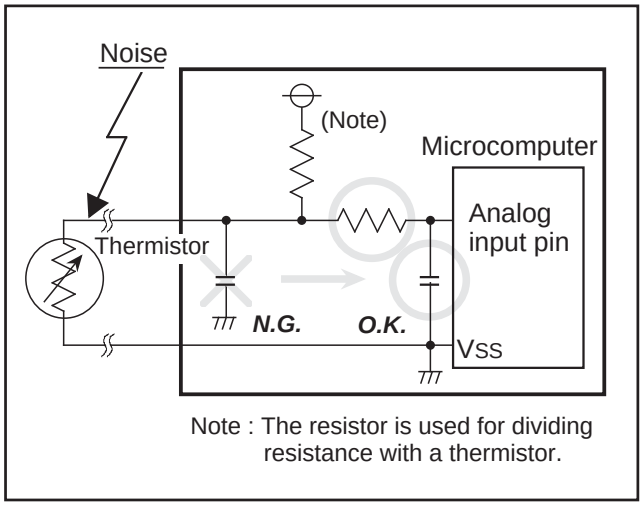



**Fig. 92 Analog signal line and a resistor and a capacitor**

- The analog input pin is connected to the capacitor of a voltage comparator. Accordingly, sufficient accuracy may not be obtained by the charge/discharge current at the time of A/D conversion when the analog signal source of high-impedance is connected to an analog input pin. In order to obtain the A/D conversion result stabilized more, please lower the impedance of an analog signal source, or add the smoothing capacitor to an analog input pin.

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.

(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.

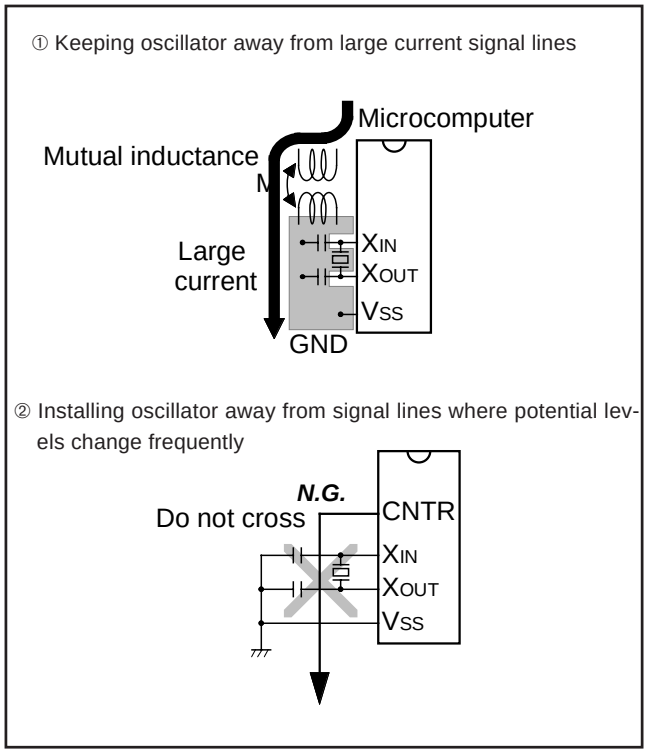



**Fig. 93 Wiring for a large current signal line/Writing of signal lines where potential levels change frequently**

(3) Oscillator protection using Vss pattern

As for a two-sided printed circuit board, print a Vss pattern on the underside (soldering side) of the position (on the component side) where an oscillator is mounted.

Connect the Vss pattern to the microcomputer Vss pin with the shortest possible wiring. Besides, separate this Vss pattern from other Vss patterns.

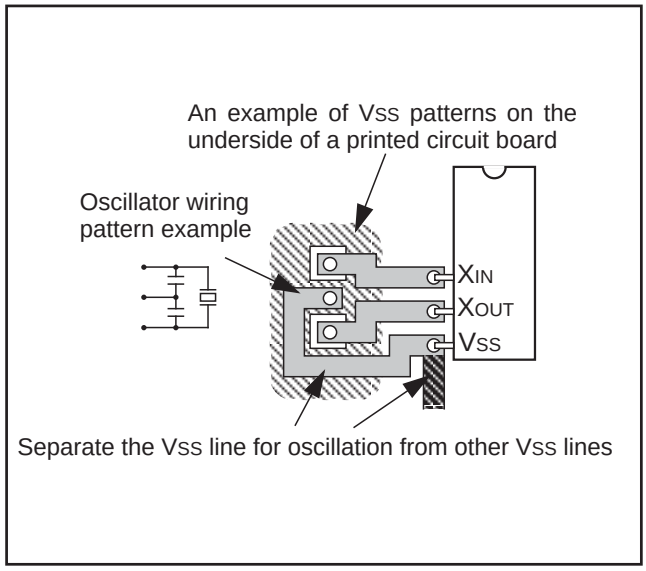



**Fig. 94 Vss pattern on the underside of an oscillator**

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.

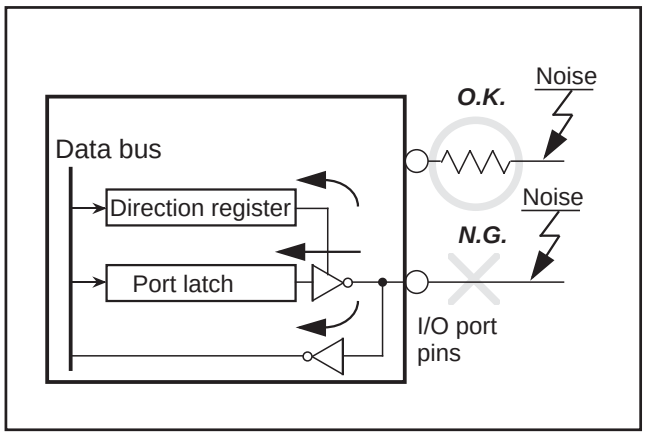



**Fig. 95 Setup for I/O ports**

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.

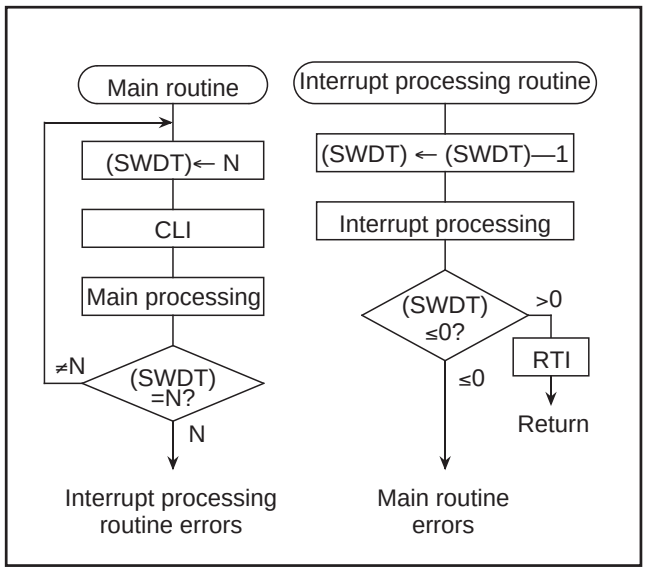



**Fig. 96 Watchdog timer by software**

## FLASH MEMORY MODE

The 7542 group's flash memory version has the flash memory that can be rewritten with a single power source.

For this flash memory, three flash memory modes are available in which to read, program, and erase: the parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and the CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU).

### ● Summary

Table 9 lists the summary of the 7542 Group (flash memory version).

This flash memory version has some blocks on the flash memory as shown in Figure 97 and each block can be erased.

In addition to the ordinary User ROM area to store the MCU operation control program, the flash memory has a Boot ROM area that is used to store a program to control rewriting in CPU rewrite and standard serial I/O modes. This Boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. However, the user can write a rewrite control program in this area that suits the user's application system. This Boot ROM area can be rewritten in only parallel I/O mode.

**Table 9 Summary of 7542 group's flash memory version**

| Item | | Specifications |
|---|---|---|
| Power source voltage (Vcc) | | $V_{CC}$ = 2.7 to 5.5 V |
| Temperature at program/erase | | Ta = 0 to 60 °C |
| Program/Erase $V_{PP}$ voltage ($V_{PP}$) | | $V_{CC}$ = 2.7 to 5.5 V |
| Flash memory mode | | 3 modes; Parallel I/O mode, Standard serial I/O mode, CPU rewrite mode |
| Erase block division | User ROM area/Data ROM area | Refer to Fig. 97. |
| | Boot ROM area (Note) | Not divided (4K bytes) |
| Program method | | In units of bytes |
| Erase method | | Block erase |
| Program/Erase control method | | Program/Erase control by software command |
| Number of commands | | 5 commands |
| Number of program/Erase times | | 100 |
| ROM code protection | | Available in parallel I/O mode and standard serial I/O mode |

**Note**: The Boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. This Boot ROM area can be erased and written in only parallel I/O mode.

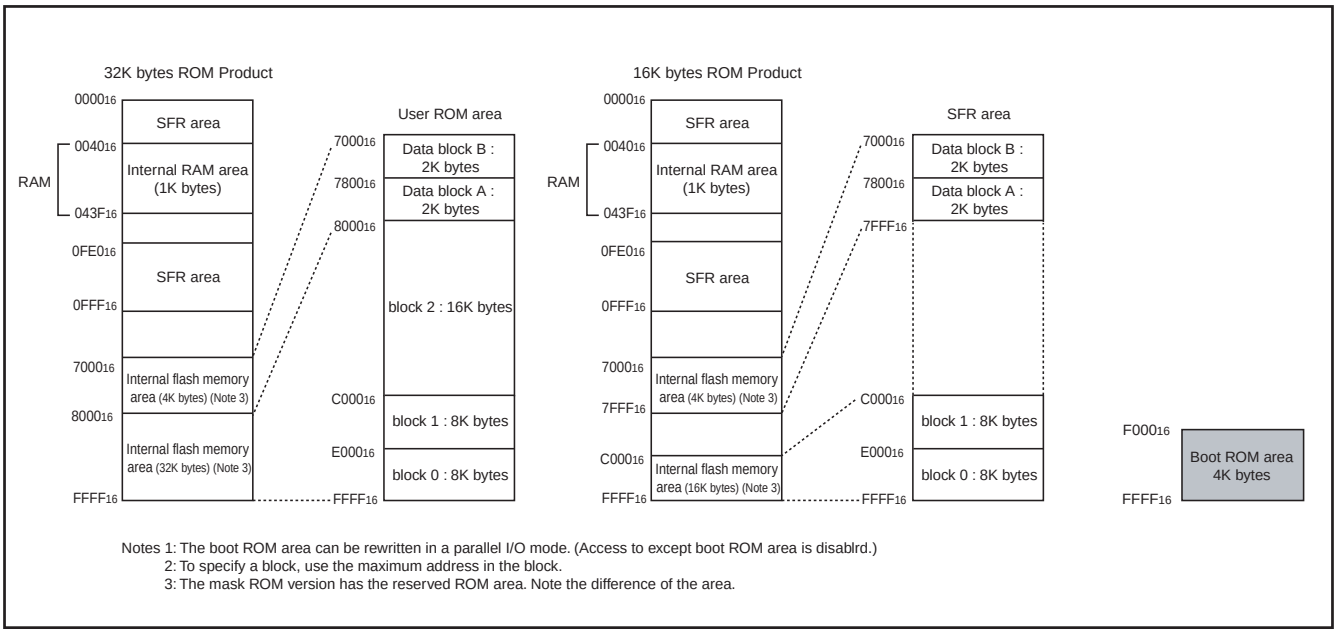



**Fig. 97 Block diagram of built-in flash memory**

## ● Boot Mode

The control program for CPU rewrite mode must be written into the User ROM or Boot ROM area in parallel I/O mode beforehand. (If the control program is written into the Boot ROM area, the standard serial I/O mode becomes unusable.)

See Figure 97 for details about the Boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNVSS pin low. In this case, the CPU starts operating using the control program in the User ROM area.

When the microcomputer is reset and the CNVSS pin high after pulling the P37(RP) pin low, P32(CE) pin high, P06/SCLK pin low and P05/TxD2 pin high, the CPU starts operating (start address of program is stored into addresses FFFC16 and FFFD16) using the control program in the Boot ROM area. This mode is called the "Boot mode". Also, User ROM area can be rewritten using the control program in the Boot ROM area.

## ● Block Address

Block addresses refer to the maximum address of each block. These addresses are used in the block erase command.

## ● CPU Rewrite Mode

In CPU rewrite mode, the internal flash memory can be operated on (read, program, or erase) under control of the Central Processing Unit (CPU).

In CPU rewrite mode, only the User ROM area shown in Figure 97 can be rewritten; the Boot ROM area cannot be rewritten. Make sure the program and block erase commands are issued for only the User ROM area and each block area.

The control program for CPU rewrite mode can be stored in either User ROM or Boot ROM area. In the CPU rewrite mode, because the flash memory cannot be read from the CPU, the rewrite control program must be transferred to internal RAM area before it can be executed.

## • Outline Performance

CPU rewrite mode is usable in the single-chip or Boot mode. The only User ROM area can be rewritten.

In CPU rewrite mode, the CPU erases, programs and reads the internal flash memory as instructed by software commands. This rewrite control program must be transferred to internal RAM area before it can be executed.

The MCU enters CPU rewrite mode by setting "1" to the CPU rewrite mode select bit (bit 1 of address 0FE016). Then, software commands can be accepted.

Use software commands to control program and erase operations. Whether a program or erase operation has terminated normally or in error can be verified by reading the status register.

**[Flash memory control registers (FMCR0 to FMCR2)]**
**0FE0$_{16}$ to 0FE2$_{16}$**

Figure 98 shows the flash memory control register 0.

Bit 0 of the flash memory control register 0 is the RY/$\overline{BY}$ status flag used exclusively to read the operating status of the flash memory. During programming and erase operations, it is "0" (busy). Otherwise, it is "1" (ready).

Bit 1 of the flash memory control register 0 is the CPU rewrite mode select bit. When this bit is set to "1", the MCU enters CPU rewrite mode. And then, software commands can be accepted. In CPU rewrite mode, the CPU becomes unable to access the internal flash memory directly. Therefore, use the control program in the internal RAM for write to bit 1. To set this bit 1 to "1", it is necessary to write "0" and then write "1" in succession to bit 1. The bit can be set to "0" by only writing "0".

Bit 2 of the flash memory control register 0 is the 8KB user block E/W mode enable bit. By setting this bit in combination with bit 4 (all user block E/W enable bit) of flash memory control register 2 (address 0FE0$_{16}$), Erase/Write to user block in CPU rewrite mode is disabled.

Bit 3 of the flash memory control register 0 is the flash memory reset bit used to reset the control circuit of internal flash memory. This bit is used when exiting CPU rewrite mode and when flash memory access has failed. When the CPU rewrite mode select bit is "1", setting "1" for this bit resets the control circuit. To release the reset, it is necessary to set this bit to "0".

Bit 5 of the flash memory control register 0 is the User ROM area select bit and is valid only in the boot mode. Setting this bit to "1" in the boot mode switches an accessible area from the boot ROM area to the user ROM area. To use the CPU rewrite mode in the boot mode, set this bit to "1". Note that when the microcomputer is booted up in the user ROM area, only the user ROM area is accessible and bit 5 is invalid; on the other hand, when the microcomputer is in the boot mode, bit 5 is valid independent of the CPU rewrite mode. To rewrite bit 5, execute the user-original reprogramming control software transferred to the internal RAM in advance.

Bit 6 of the flash memory control register 0 is the program status flag. This bit is set to "1" when writing to flash memory is failed. When program error occurs, the block cannot be used.

Bit 7 of the flash memory control register 0 is the erase status flag. This bit is set to "1" when erasing flash memory is failed. When erase error occurs, the block cannot be used.

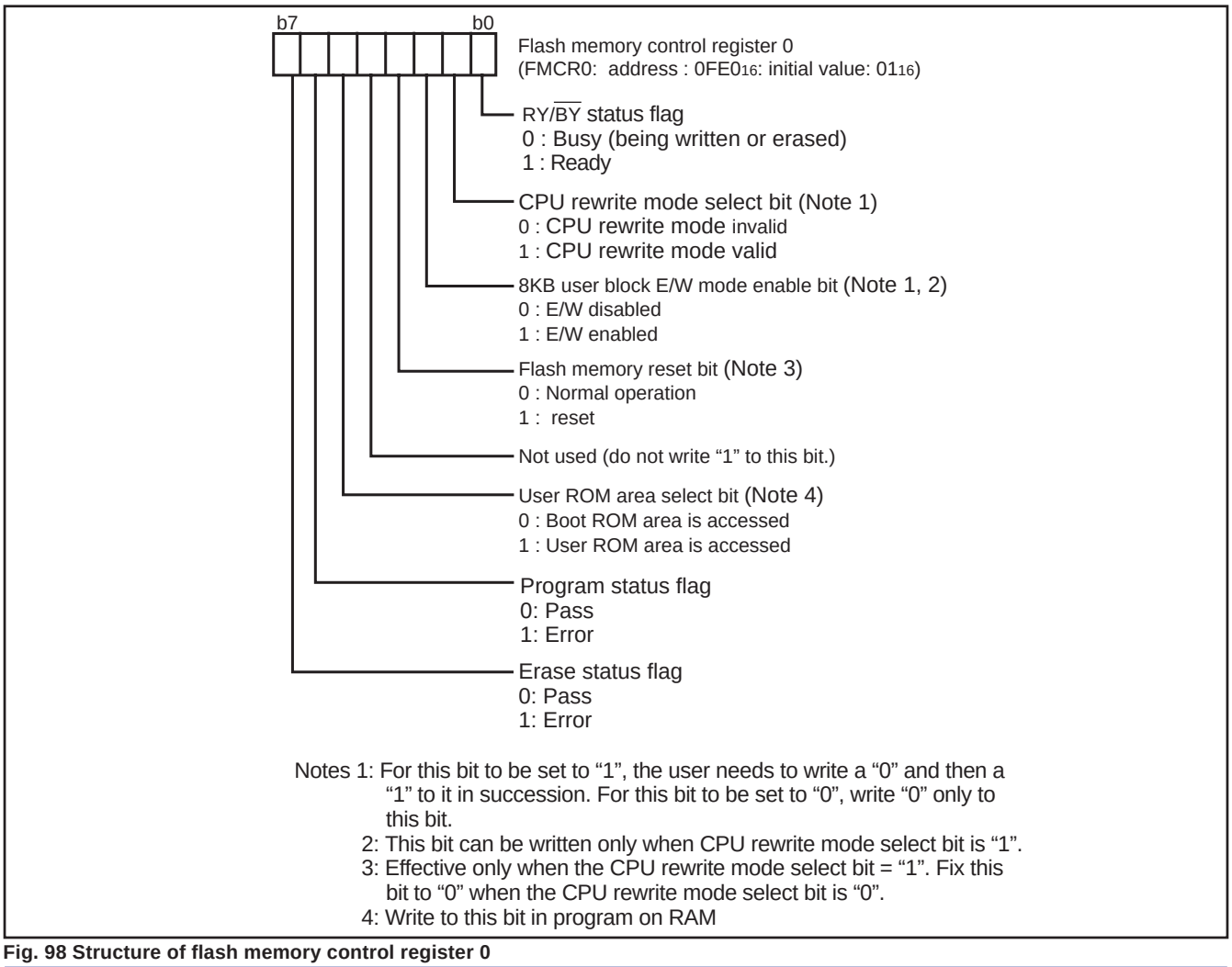



**Fig. 98 Structure of flash memory control register 0**

Figure 99 shows the flash memory control register 1.
Bit 0 of the flash memory control register 1 is the Erase suspend enable bit. By setting this bit to “1”, the erase suspend mode to suspend erase processing temporarily when block erase command is executed can be used. In order to set this bit to “1”, writing “0” and “1” in succession to bit 0. In order to set this bit to “0”, write “0” only to bit 0.
Bit 1 of the flash memory control register 1 is the erase suspend request bit. By setting this bit to “1” when erase suspend enable bit is “1”, the erase processing is suspended.
Bit 6 of the flash memory control register 1 is the erase suspend flag. This bit is cleared to “0” at the flash erasing.

Figure 100 shows the flash memory control register 2.
Bit 0 of the flash memory control register 1 is the all user block E/W enable bit. By setting this bit to “0”, Erase/Write to all user block (blocks 0, 1, 2) is disabled. As a result, error writing in program to write only to data block can be prevented.

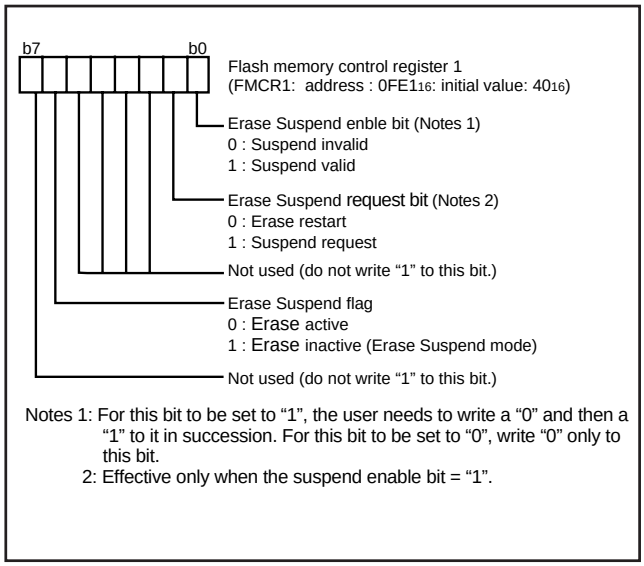



**Fig. 99 Structure of flash memory control register 1**

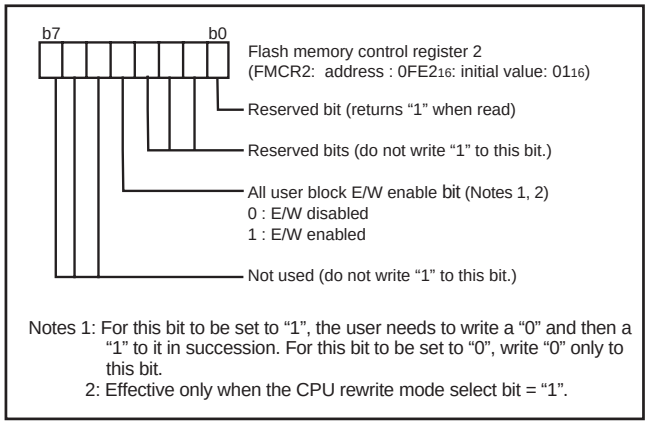
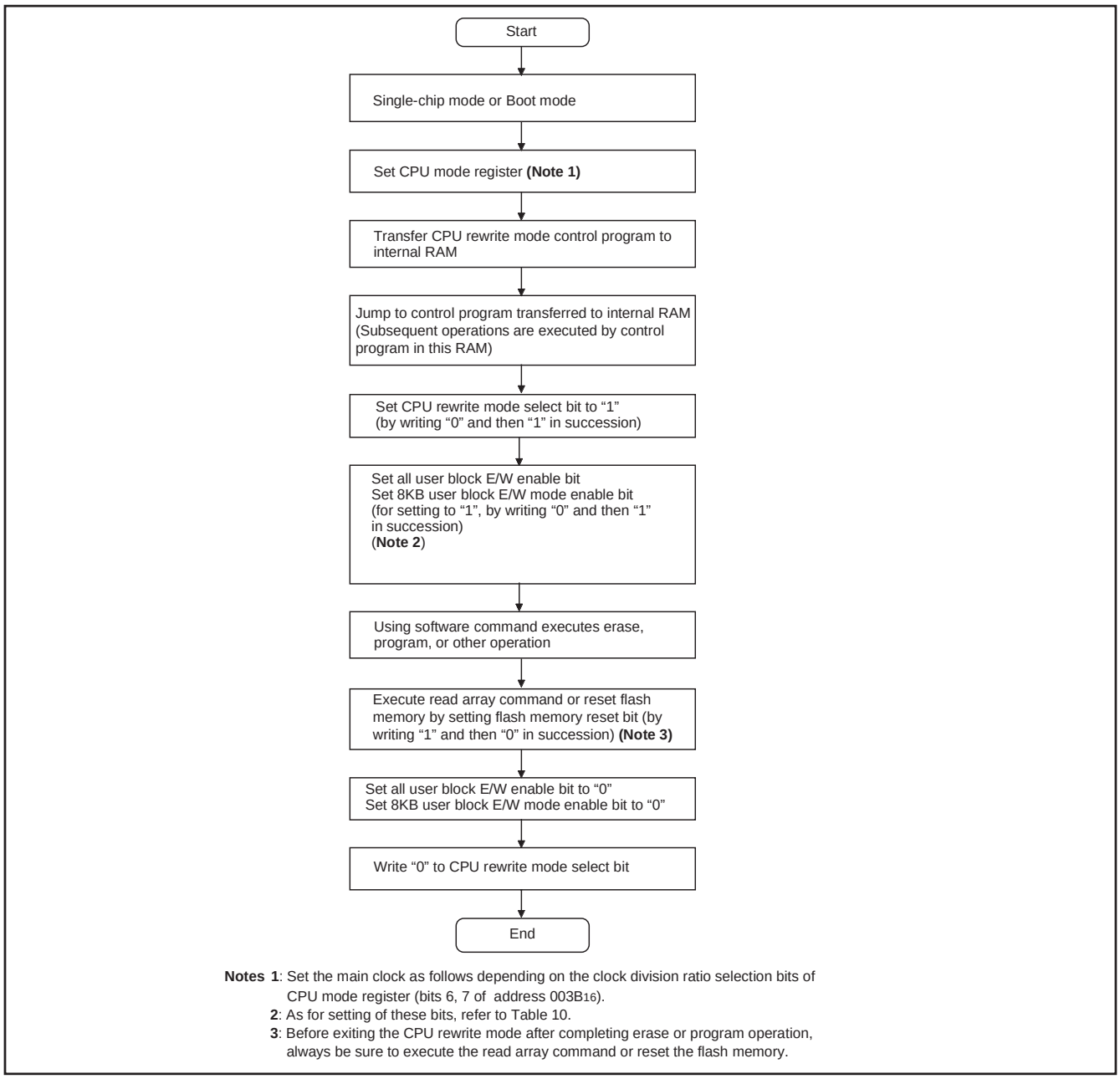



**Fig. 100 Structure of flash memory control register 2**

**Table 10 Erase/Write disable setting**

| CPU rewrite mode select bit | All user block E/W enable bit | 8KB user block E/W enable bit | Block 0: 8KB<br>Block 1: 8KB | Block 2: 16KB | Data block A: 2KB<br>Data block B: 2KB |
|---|---|---|---|---|---|
| 0 | 0 | 0 | E/W disabled (RESET) | E/W disabled (RESET) | E/W disabled (RESET) |
| 0 | 0 | 1 | E/W disabled | E/W disabled | E/W disabled |
| 0 | 1 | 0 | E/W disabled | E/W disabled | E/W disabled |
| 0 | 1 | 1 | E/W disabled | E/W disabled | E/W disabled |
| 1 | 0 | 0 | E/W disabled | E/W disabled | E/W enabled |
| 1 | 0 | 1 | E/W disabled | E/W disabled | E/W enabled |
| 1 | 1 | 0 | E/W disabled | E/W enabled | E/W enabled |
| 1 | 1 | 1 | E/W enabled | E/W enabled | E/W enabled |

Figure 101 shows a flowchart for setting/releasing CPU rewrite mode.




**Notes 1**: Set the main clock as follows depending on the clock division ratio selection bits of CPU mode register (bits 6, 7 of address $003B_{16}$).
**2**: As for setting of these bits, refer to Table 10.
**3**: Before exiting the CPU rewrite mode after completing erase or program operation, always be sure to execute the read array command or reset the flash memory.

**Fig. 101 CPU rewrite mode set/release flowchart**

## ■ Notes on CPU Rewrite Mode

Take the notes described below when rewriting the flash memory in CPU rewrite mode.

**●Operation speed**

During CPU rewrite mode, set the system clock $\phi$ to 4.0 MHz or less using the clock division ratio selection bits (bits 6 and 7 of address $003B_{16}$).

**●Instructions inhibited against use**

The instructions which refer to the internal data of the flash memory cannot be used during CPU rewrite mode.

**●Interrupts inhibited against use**

The interrupts cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory.

**●Watchdog timer**

If the watchdog timer has been already activated, internal reset due to an underflow will not occur because the watchdog timer is surely cleared during program or erase.

**●Reset**

Reset is always valid. The MCU is activated using the boot mode at release of reset in the condition of CNVss = "H", so that the program will begin at the address which is stored in addresses $FFFC_{16}$ and $FFFD_{16}$ of the boot ROM area.

## ● Software Commands

Table 11 lists the software commands.
After setting the CPU rewrite mode select bit to "1", execute a software command to specify an erase or program operation.
Each software command is explained below.

**• Read Array Command ($FF_{16}$)**

The read array mode is entered by writing the command code "$FF_{16}$" in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the contents of the specified address are read out at the data bus ($D_0$ to $D_7$).
The read array mode is retained until another command is written.

**• Read Status Register Command ($70_{16}$)**

When the command code "$70_{16}$" is written in the first bus cycle, the contents of the status register are read out at the data bus ($D_0$ to $D_7$) by a read in the second bus cycle.
The status register is explained in the next section.

**• Clear Status Register Command ($50_{16}$)**

This command is used to clear the bits SR4 and SR5 of the status register after they have been set. These bits indicate that operation has ended in an error. To use this command, write the command code "$50_{16}$" in the first bus cycle.

**• Program Command ($40_{16}$)**

Program operation starts when the command code "$40_{16}$" is written in the first bus cycle. Then, if the address and data to program are written in the 2nd bus cycle, program operation (data programming and verification) will start.
Whether the write operation is completed can be confirmed by read status register or the RY/$\overline{BY}$ status flag. When the program starts, the read status register mode is entered automatically and the contents of the status register is read at the data bus ($D_0$ to $D_7$). The status register bit 7 (SR7) is set to "0" at the same time the write operation starts and is returned to "1" upon completion of the write operation. In this case, the read status register mode remains active until the read array command ($FF_{16}$) is written.

The RY/$\overline{BY}$ status flag of the flash memory control register is "0" during write operation and "1" when the write operation is completed as is the status register bit 7.
At program end, program results can be checked by reading the status register.

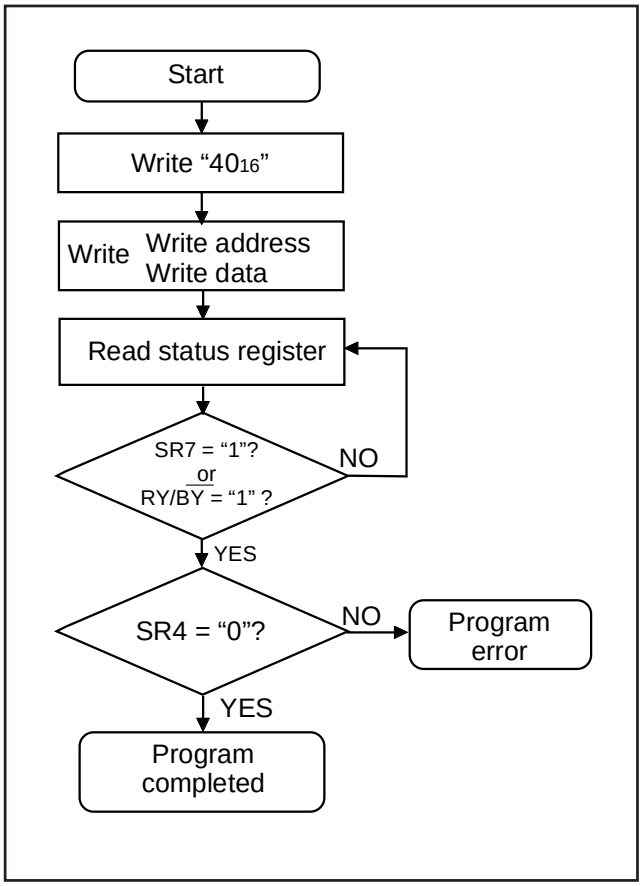



**Fig. 102 Program flowchart**

**Table 11 List of software commands (CPU rewrite mode)**

| Command | First bus cycle | | | Second bus cycle | | |
|---|---|---|---|---|---|---|
| | Mode | Address | Data ($D_0$ to $D_7$) | Mode | Address | Data ($D_0$ to $D_7$) |
| Read array | Write | × (Note 4) | $FF_{16}$ | | | |
| Read status register | Write | × | $70_{16}$ | Read | × | SRD (Note 1) |
| Clear status register | Write | × | $50_{16}$ | | | |
| Program | Write | × | $40_{16}$ | Write | WA (Note 2) | WD (Note 2) |
| Block erase | Write | × | $20_{16}$ | Write | BA (Note 3) | $D0_{16}$ |

SRD = Status Register Data
WA = Write Address, WD = Write Data
BA = Block Address to be erased (Input the maximum address of each block.)
× = × denotes a given address in the user ROM area.

**• Block Erase Command ($20_{16}$/$D0_{16}$)**

By writing the command code "$20_{16}$" in the first bus cycle and the confirmation command code "$D0_{16}$" and the block address in the second bus cycle that follows, the block erase (erase and erase verify) operation starts for the block address of the flash memory to be specified.

Whether the block erase operation is completed can be confirmed by read status register or the RY/$\overline{BY}$ status flag of flash memory control register. At the same time the block erase operation starts, the read status register mode is automatically entered, so that the contents of the status register can be read out. The status register bit 7 (SR7) is set to "0" at the same time the block erase operation starts and is returned to "1" upon completion of the block erase operation. In this case, the read status register mode remains active until the read array command ($FF_{16}$) is written.

The RY/$\overline{BY}$ status flag is "0" during block erase operation and "1" when the block erase operation is completed as is the status register bit 7.

After the block erase ends, erase results can be checked by reading the status register. For details, refer to the section where the status register is detailed.

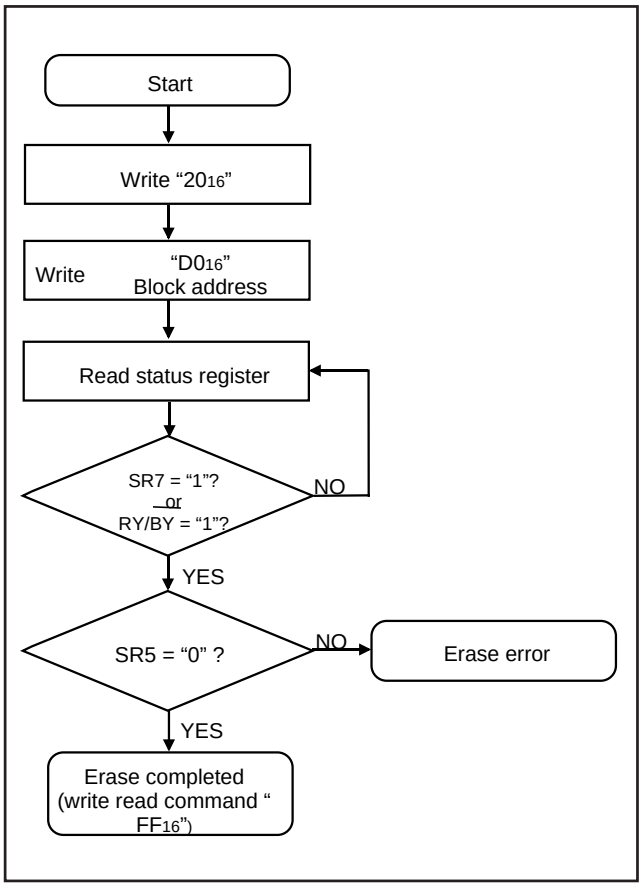



**Fig. 103 Erase flowchart**

## ● Status Register

The status register shows the operating status of the flash memory and whether erase operations and programs ended successfully or in error. It can be read in the following ways:

(1) By reading an arbitrary address from the User ROM area after writing the read status register command ($70_{16}$)

(2) By reading an arbitrary address from the User ROM area in the period from when the program starts or erase operation starts to when the read array command ($FF_{16}$) is input.

Also, the status register can be cleared by writing the clear status register command ($50_{16}$).

After reset, the status register is set to "$80_{16}$".

Table 12 shows the status register. Each bit in this register is explained below.

**•Sequencer status (SR7)**

The sequencer status indicates the operating status of the flash memory. This bit is set to "0" (busy) during write or erase operation and is set to "1" when these operations ends.

After power-on, the sequencer status is set to "1" (ready).

**•Erase status (SR5)**

The erase status indicates the operating status of erase operation. If an erase error occurs, it is set to "1". When the erase status is cleared, it is reset to "0".

**•Program status (SR4)**

The program status indicates the operating status of write operation. When a write error occurs, it is set to "1".

The program status is reset to "0" when it is cleared.

If "1" is written for any of the SR5 and SR4 bits, the read array, program, and block erase commands are not accepted. Before executing these commands, execute the clear status register command ($50_{16}$) and clear the status register.

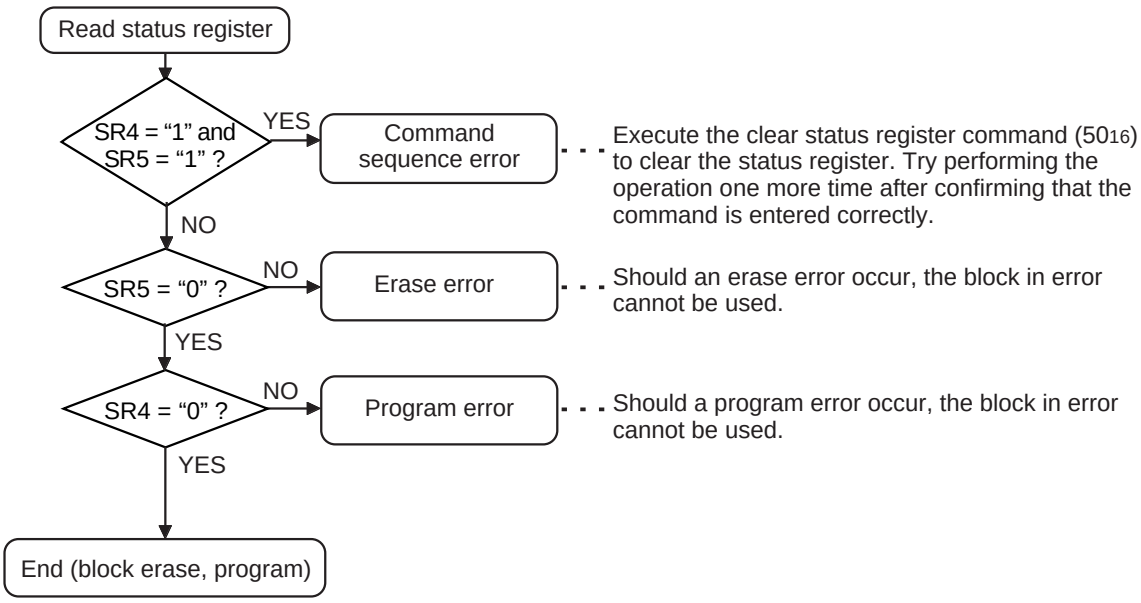
Also, if any commands are not correct, both SR5 and SR4 are set to "1".

**Table 12 Definition of each bit in status register**

| Each bit of SRD bits | Status name | Definition | |
|---|---|---|---|
| | | "1" | "0" |
| SR7 (bit7) | Sequencer status | Ready | Busy |
| SR6 (bit6) | Reserved | - | - |
| SR5 (bit5) | Erase status | Terminated in error | Terminated normally |
| SR4 (bit4) | Program status | Terminated in error | Terminated normally |
| SR3 (bit3) | Reserved | - | - |
| SR2 (bit2) | Reserved | - | - |
| SR1 (bit1) | Reserved | - | - |
| SR0 (bit0) | Reserved | - | - |

## ● Full Status Check

By performing full status check, it is possible to know the execution results of erase and program operations. Figure 104 shows a full status check flowchart and the action to be taken when each error occurs.




**Note**: When one of SR5 and SR4 is set to "1", none of the read array, program, and block erase commands is accepted. Execute the clear status register command ($50_{16}$) before executing these commands.

**Fig. 104 Full status check flowchart and remedial procedure for errors**

## ● Functions To Inhibit Rewriting Flash Memory Version

To prevent the contents of internal flash memory from being read out or rewritten easily, this MCU incorporates a ROM code protect function for use in parallel I/O mode and an ID code check function for use in standard serial I/O mode.

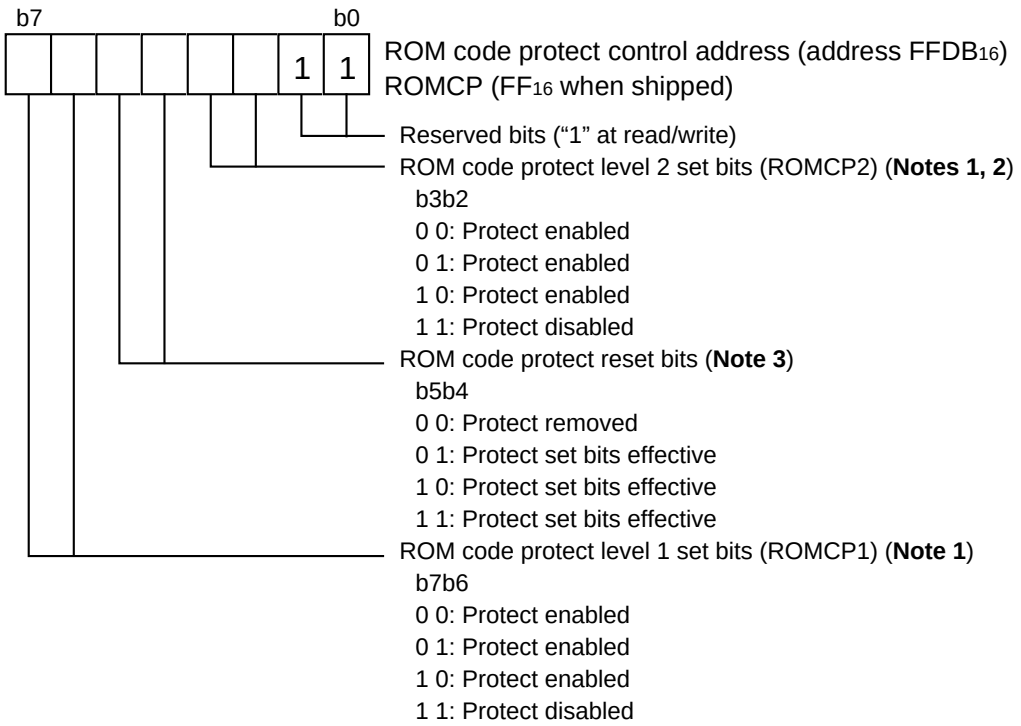
**(1) ROM Code Protect Function**

The ROM code protect function is the function to inhibit reading out or modifying the contents of internal flash memory by using the ROM code protect control address (address FFDB16) in parallel I/O mode. Figure 105 shows the ROM code protect control address (address FFDB16). (This address exists in the User ROM area.)

If one or both of the pair of ROM code protect bits is set to “0”, the ROM code protect is turned on, so that the contents of internal flash memory are protected against readout and modification. The ROM code protect is implemented in two levels. If level 2 is selected, the flash memory is protected even against readout by a shipment inspection LSI tester, etc. When an attempt is made to select both level 1 and level 2, level 2 is selected by default.

If both of the two ROM code protect reset bits are set to “00”, the ROM code protect is turned off, so that the contents of internal flash memory can be readout or modified. Once the ROM code protect is turned on, the contents of the ROM code protect reset bits cannot be modified in parallel I/O mode. Use the serial I/O or CPU rewrite mode to rewrite the contents of the ROM code protect reset bits.

Rewriting of only the ROM code protect control address (address FFDB16) cannot be performed. When rewriting the ROM code protect reset bit, rewrite the whole user ROM area (block 0) containing the ROM code protect control address.




**Notes 1**: When ROM code protect is turned on, the internal flash memory is protected against readout or modification in parallel I/O mode.

**2**: When ROM code protect level 2 is turned on, ROM code readout by a shipment inspection LSI tester, etc. also is inhibited.

**3**: The ROM code protect reset bits can be used to turn off ROM code protect level 1 and ROM code protect level 2. However, since these bits cannot be modified in parallel I/O mode, they need to be rewritten in serial I/O mode or CPU rewrite mode.

**Fig. 105 Structure of ROM code protect control address**

**(2) ID Code Check Function**

Use this function in standard serial I/O mode. When the contents of the flash memory are not blank, the ID code sent from the programmer is compared with the ID code written in the flash memory to see if they match. If the ID codes do not match, the commands sent from the programmer are not accepted. The ID code consists of 8-bit data, and its areas are FFD4$_{16}$ to FFDA$_{16}$. Write a program which has had the ID code preset at these addresses to the flash memory.

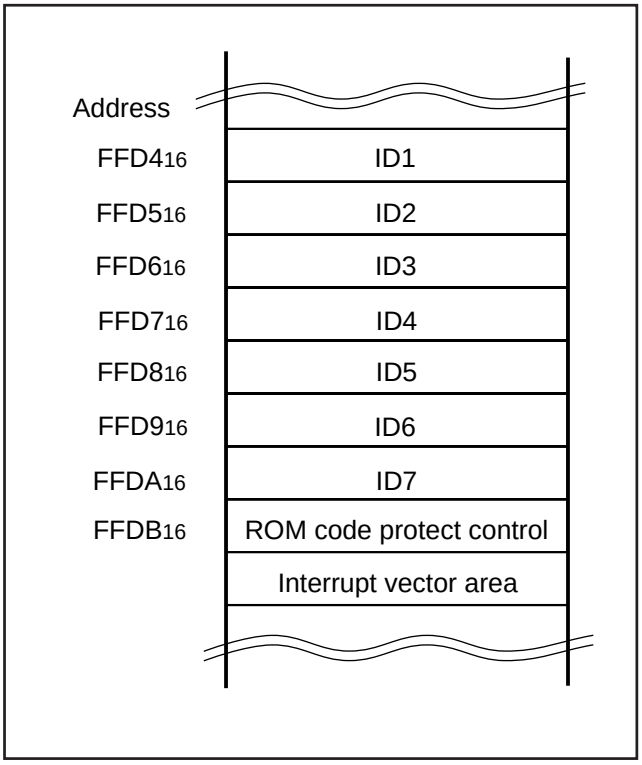



**Fig. 106 ID code store addresses**

## ● Parallel I/O Mode

The parallel I/O mode is used to input/output software commands, address and data in parallel for operation (read, program and erase) to internal flash memory.
Use the external device (writer) only for 7542 Group (flash memory version). For details, refer to the user's manual of each writer manufacturer.

### • User ROM and Boot ROM Areas

In parallel I/O mode, the User ROM and Boot ROM areas shown in Figure 97 can be rewritten. Both areas of flash memory can be operated on in the same way.
The Boot ROM area is 4 Kbytes in size and located at addresses $F000_{16}$ through $FFFF_{16}$. Make sure program and block erase operations are always performed within this address range. (Access to any location outside this address range is prohibited.)
In the Boot ROM area, an erase block operation is applied to only one 4 Kbyte block. The boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. Therefore, using the MCU in standard serial I/O mode, do not rewrite to the Boot ROM area.

## ● Standard serial I/O Mode

The standard serial I/O mode inputs and outputs the software commands, addresses and data needed to operate (read, program, erase, etc.) the internal flash memory. This I/O is clock synchronized serial. This mode requires a purpose-specific peripheral unit.

The standard serial I/O mode is different from the parallel I/O mode in that the CPU controls flash memory rewrite (uses the CPU rewrite mode), rewrite data input and so forth. The standard serial I/O mode is started when the microcomputer is reset and the CNVSS pin high after pulling the P37(RP) pin low, P32(CE) pin high, P06/SCLK2 pin low and P05/TxD2 pin high. (In the ordinary microcomputer mode, set CNVss pin to “L” level.)

This control program is written in the Boot ROM area when the product is shipped from Renesas. Accordingly, make note of the fact that the standard serial I/O mode cannot be used if the Boot ROM area is rewritten in parallel I/O mode.

The standard serial I/O mode has  standard serial I/O mode 1 of the clock synchronous serial and the standard serial I/O mode 2 of the clock asynchronous serial.

Table 13 lists the description of pin function (standard serial I/O mode 1). Figures 107 to 109 show the pin connections for the standard serial I/O mode 1.

Table 14 lists the description of pin function (standard serial I/O mode 2). Figures 112 to 114 show the pin connections for the standard serial I/O mode 2.

In standard serial I/O mode, only the User ROM area shown in Figure 97 can be rewritten. The Boot ROM area cannot be written.

In standard serial I/O mode, a 7-byte ID code is used. When there is data in the flash memory, this function determines whether the ID code sent from the peripheral unit (programmer) and those written in the flash memory match.The commands sent from the peripheral unit (programmer) are not accepted unless the ID code matches.

## (1) Standard serial I/O mode 1

**Table 13 Description of pin function (standard serial I/O mode 1)**

| Pin name | Signal name | I/O | Function |
|---|---|---|---|
| $V_{CC}$,$V_{SS}$ | Power supply | I | Apply 2.7 to 5.5 V to the Vcc pin and 0 V to the Vss pin. |
| $CNV_{SS}$ | $CNV_{SS}$ | I | After input of port is set, input “H” level. |
| $\overline{RESET}$ | Reset input | I | Reset input pin. System operates when $\overline{RESET}$ pin is set to “H” level after CNVss pin is set to “H” level. |
| $X_{IN}$ | Clock input | I | Connect an oscillation circuit between the $X_{IN}$ and $X_{OUT}$ pins.<br>As for the connection method, refer to the “clock generating circuit”.<br>(When system operates only by the on-chip oscillator, an external circuit is not required.) |
| $X_{OUT}$ | Clock output | O | |
| $V_{REF}$ | Reference voltage input | I | Apply reference voltage of A/D to this pin. |
| $P0_0$–$P0_3$ | I/O port P0 | I/O | Input “L” or “H” level, or keep open. |
| $P0_4$ | RxD input | I | Serial data input pin. |
| $P0_5$ | TxD output | O | Serial data output pin. |
| $P0_6$ | $S_{CLK}$ input | I | Serial clock input pin. |
| $P0_7$ | BUSY output | O | BUSY signal output pin. |
| $P1_0$–$P1_4$ | I/O port P1 | I/O | Input “L” or “H” level, or keep open. |
| $P2_0$–$P2_7$ | I/O port P2 | I/O | Input “L” or “H” level, or keep open. |
| $P3_0$, $P3_1$, $P3_3$–$P3_6$ | I/O port P3 | I/O | Input “L” or “H” level, or keep open. |
| $P3_2$ | CE input | I | Input “H” level. |
| $P3_7$ | RP input | I | Input “L” level. |

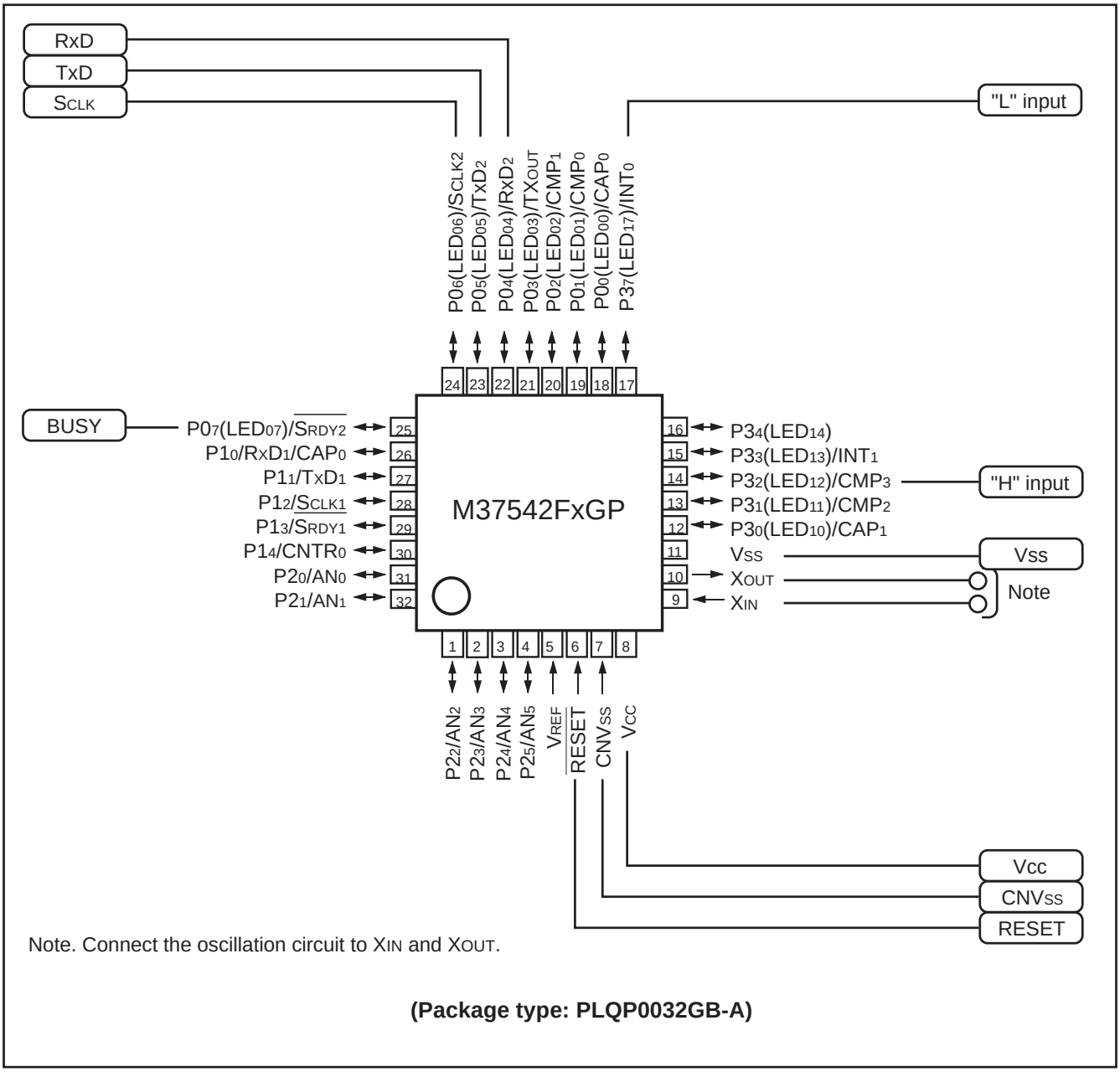
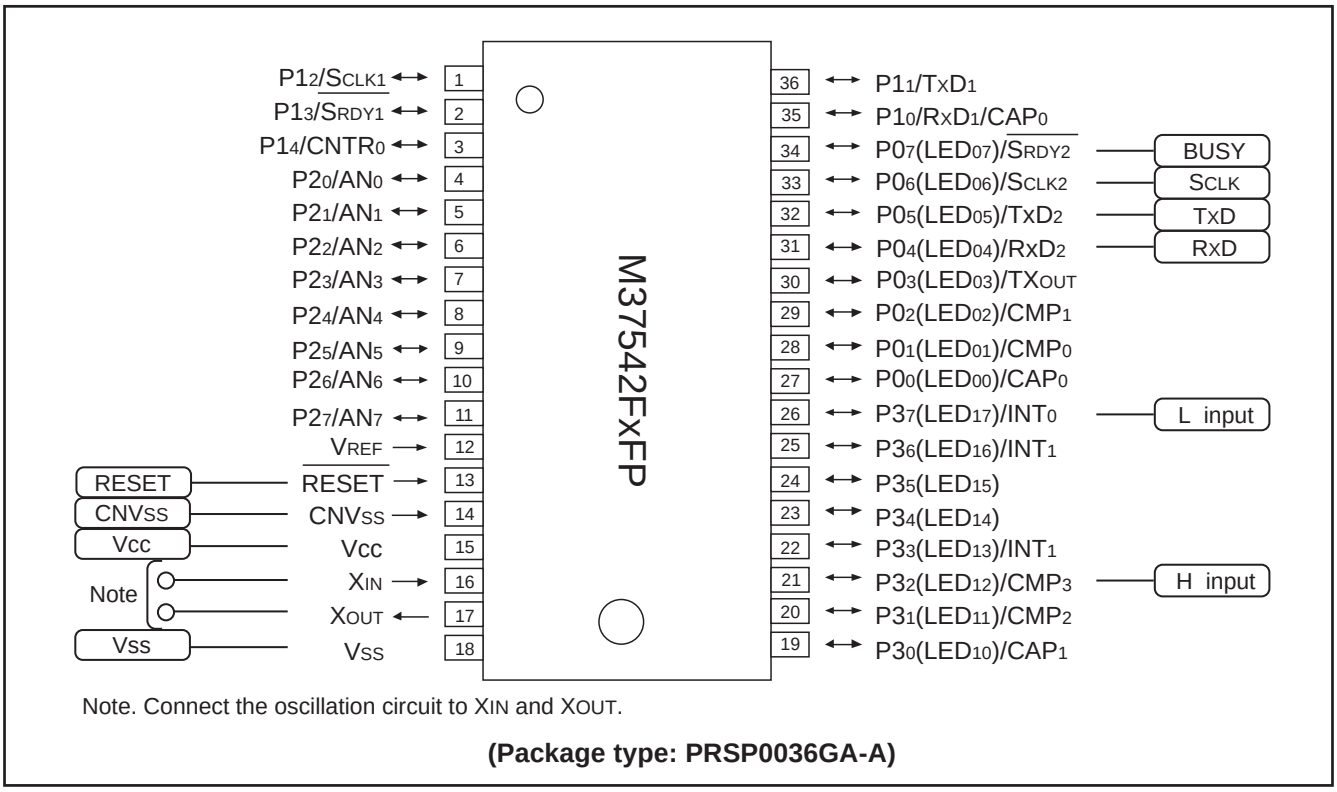


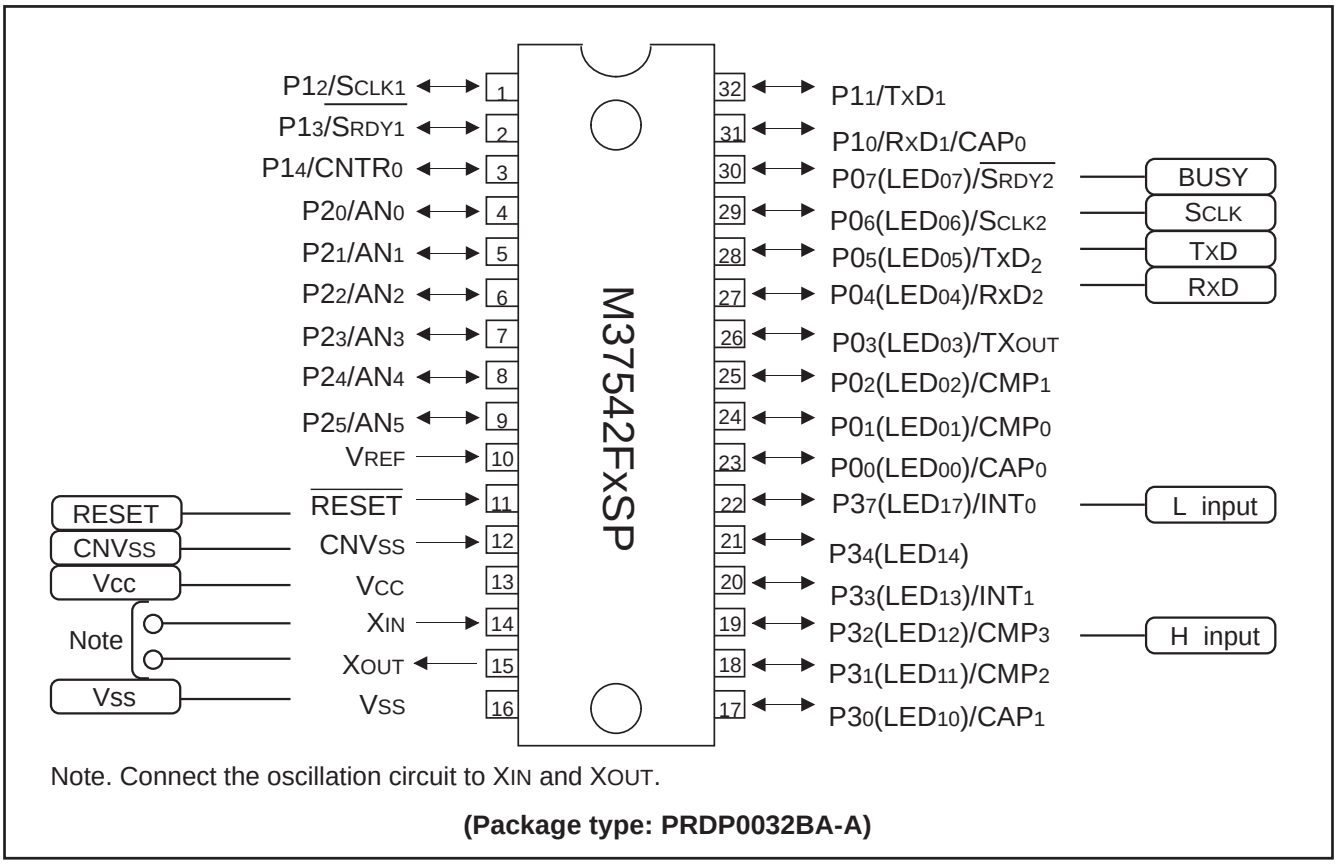

**Fig. 107 Pin connection diagram in standard serial I/O mode 1 (PLQP0032GB-A package)**




Note. Connect the oscillation circuit to XIN and XOUT.

**(Package type: PRSP0036GA-A)**

**Fig. 108 Pin connection diagram in standard serial I/O mode 1 (PRSP0036GA-A package)**



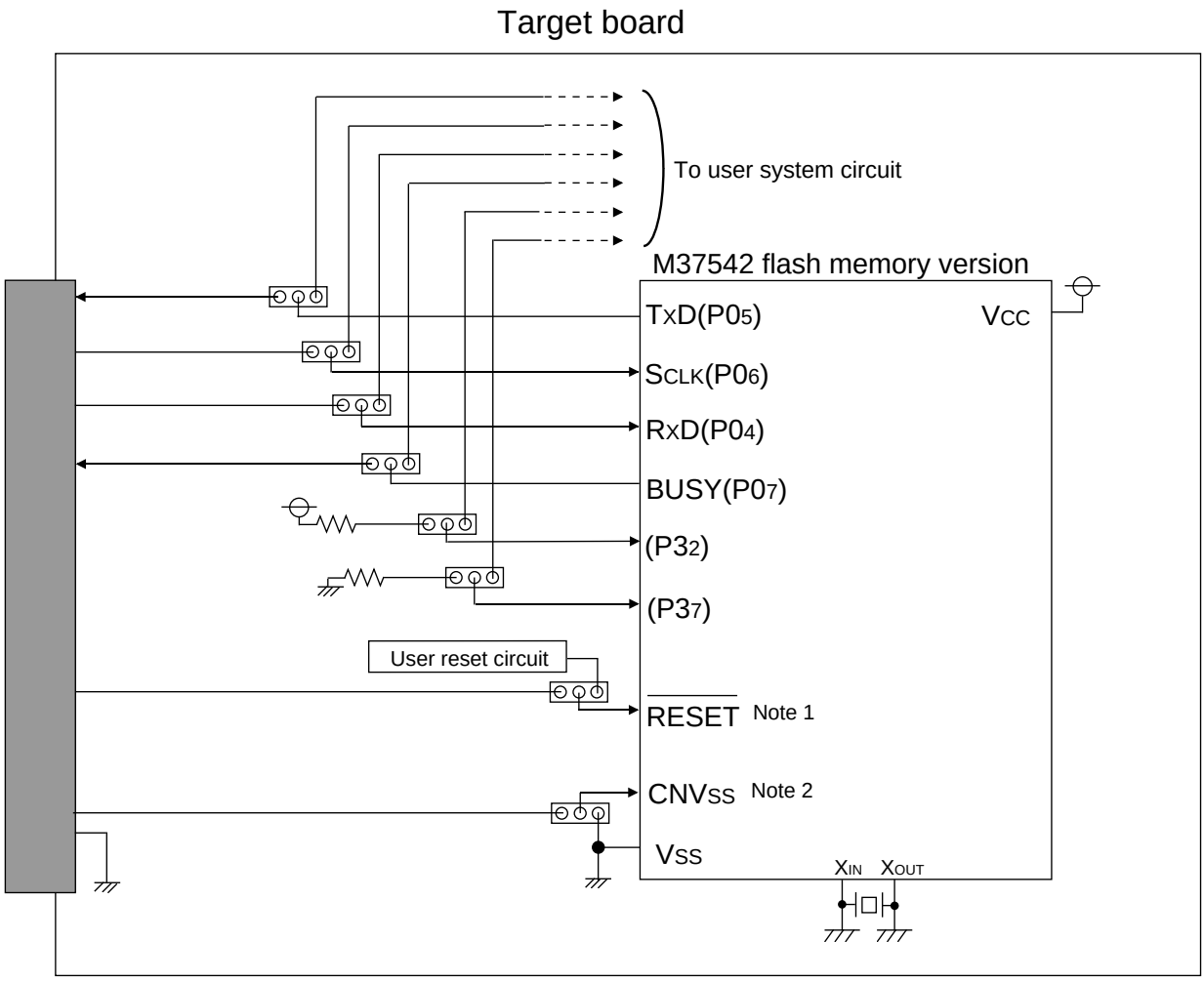

Note. Connect the oscillation circuit to XIN and XOUT.

**(Package type: PRDP0032BA-A)**

**Fig. 109 Pin connection diagram in standard serial I/O mode 1 (PRDP0032BA-A package)**

## • Standard serial I/O mode 1

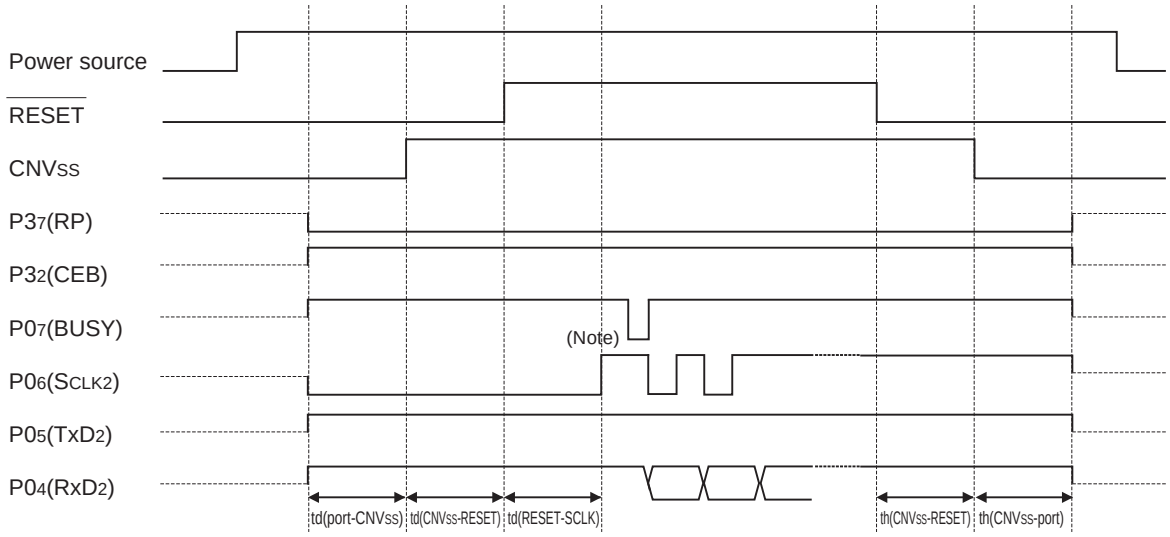
Figure 110 shows the handling example of control pins on the user system board when the standard serial I/O mode 1 is used.
Refer to the serial programmer manual of your programmer to handle pins controlled by the programmer.




Notes 1: Connect the user reset circuit to the $\overline{\text{RESET}}$ pin with the shortest possible wiring.
In the normal microcomputer mode, disconnect a wiring of a serial rewrite circuit, which is for the flash memory version, from the MCU by a jumper switch.

2: Connect the CNVss pin to the Vss pin with the shortest possible wiring.
In the normal microcomputer mode, disconnect a wiring of a serial rewrite circuit, which is for the flash memory version, from the MCU by a jumper switch.

**Fig. 110 Handling example of control pins in standard serial I/O mode 1**




| Symbol | Ratings | | | Unit |
|---|---|---|---|---|
| | Min. | Typ. | Max. | |
| td(port-CNVss) | 1 | - | - | ms |
| td(CNVss-RESET) | 1 | - | - | ms |
| td(RESET-SCLK) | 0.05 | - | 0.5 | ms |
| th(RESET-CNVss) | 1 | - | - | ms |
| th(CNVss-port) | 1 | - | - | ms |

Note: Keep input of P06 "H" until P07 turns "L".

**Fig. 111 Timing diagram in standard serial I/O mode 1**

## (2) Standard serial I/O mode 2

**Table 14 Description of pin function (standard serial I/O mode 2)**

| Pin name | Signal name | I/O | Function |
|---|---|---|---|
| $V_{CC}$,$V_{SS}$ | Power supply | I | Apply 2.7 to 5.5 V to the Vcc pin and 0 V to the Vss pin. |
| $CNV_{SS}$ | $CNV_{SS}$ | I | After input of port is set, input “H” level. |
| $\overline{RESET}$ | Reset input | I | Reset input pin. System operates when $\overline{RESET}$ pin is set to “H” level after CNVss pin is set to “H” level. |
| $X_{IN}$ | Clock input | I | Connect an oscillation circuit between the $X_{IN}$ and $X_{OUT}$ pins. As for the connection method, refer to the “clock generating circuit”. (When system operates only by the on-chip oscillator, an external circuit is not required.) |
| $X_{OUT}$ | Clock output | O | |
| $V_{REF}$ | Reference voltage input | I | Apply reference voltage of A/D to this pin. |
| $P0_0$–$P0_3$ | I/O port P0 | I/O | Input “L” or “H” level, or keep open. |
| $P0_4$ | RxD input | I | Serial data input pin. |
| $P0_5$ | TxD output | O | Serial data output pin. |
| $P0_6$ | $S_{CLK}$ input | I | Input “L” level. |
| $P0_7$ | BUSY output | O | BUSY signal output pin. |
| $P1_0$–$P1_4$ | I/O port P1 | I/O | Input “L” or “H” level, or keep open. |
| $P2_0$–$P2_7$ | I/O port P2 | I/O | Input “L” or “H” level, or keep open. |
| $P3_0$, $P3_1$, $P3_3$–$P3_6$ | I/O port P3 | I/O | Input “L” or “H” level, or keep open. |
| $P3_2$ | CE input | I | Input “H” level. |
| $P3_7$ | RP input | I | Input “L” level. |

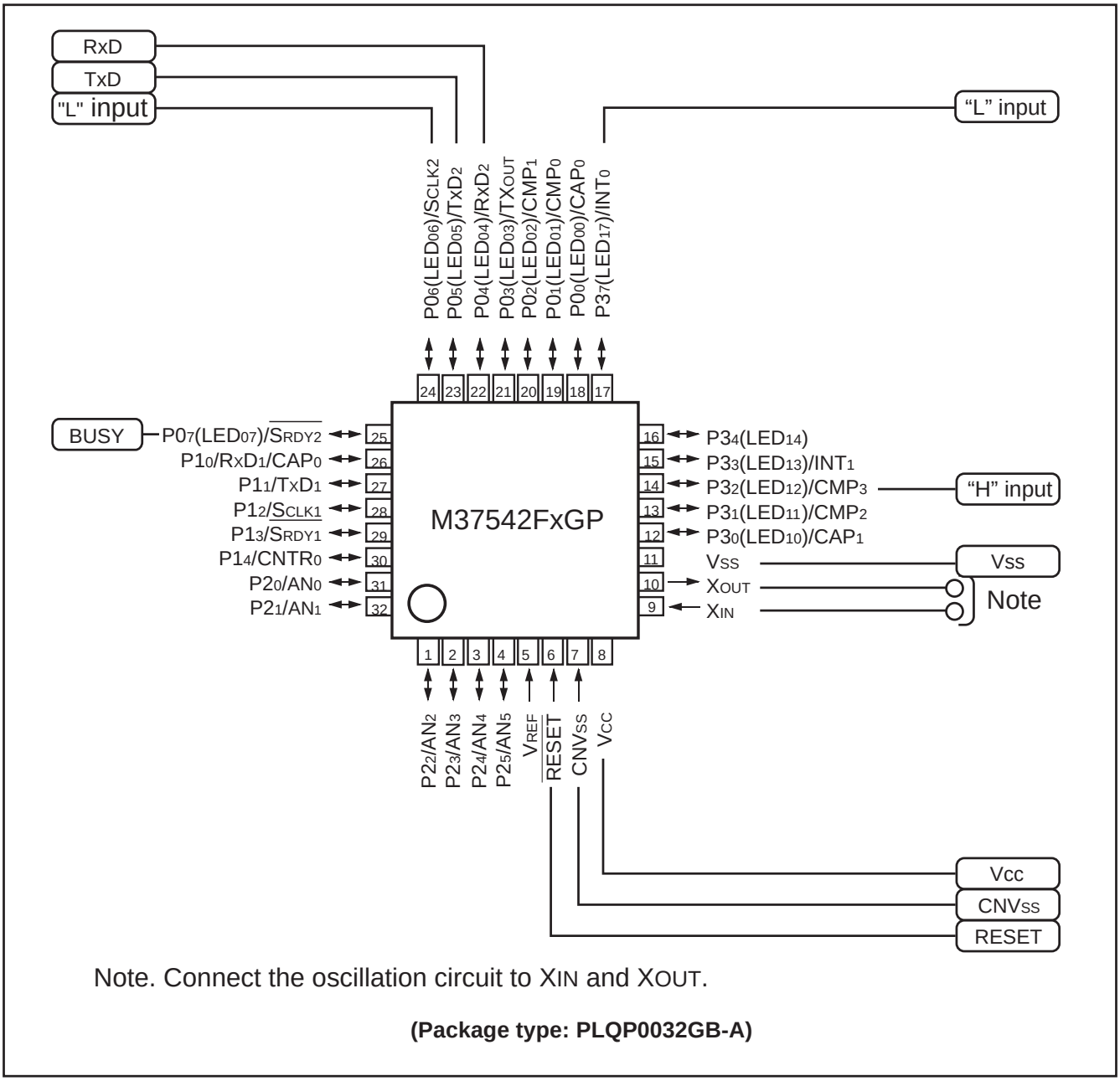
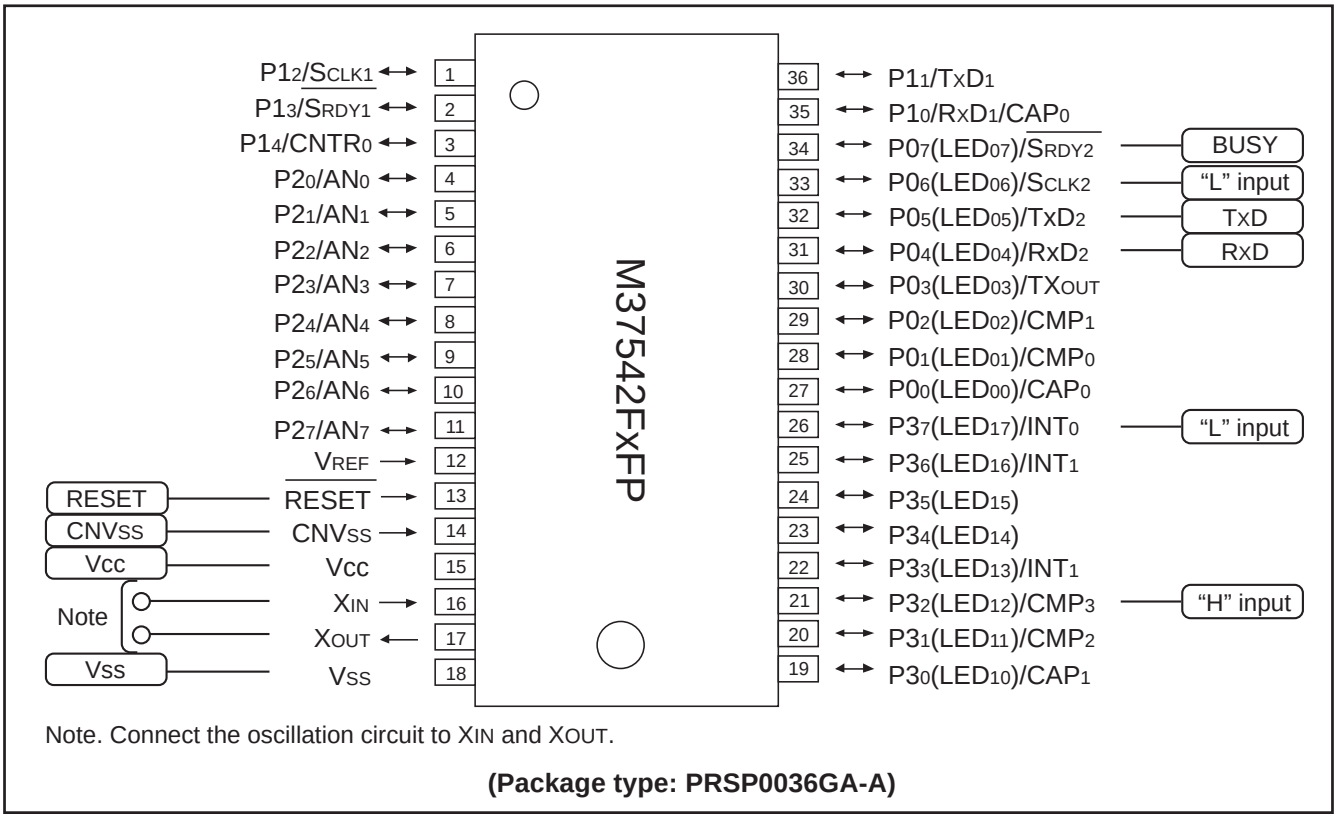


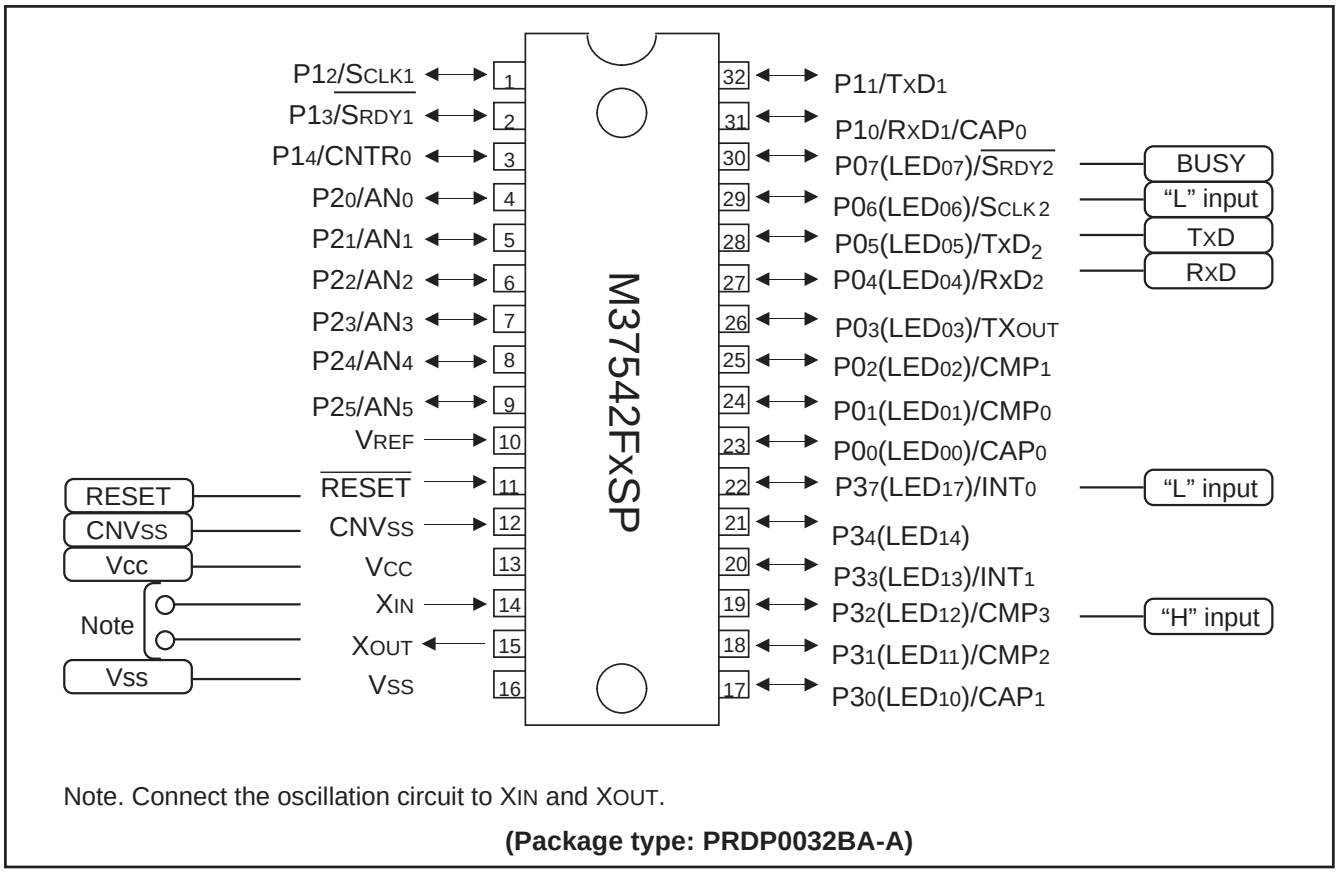

**Fig. 112 Pin connection diagram in standard serial I/O mode 2 (PLQP0032GB-A package)**




Note. Connect the oscillation circuit to XIN and XOUT.

**(Package type: PRSP0036GA-A)**

**Fig. 113 Pin connection diagram in standard serial I/O mode 2 (PRSP0036GA-A package)**



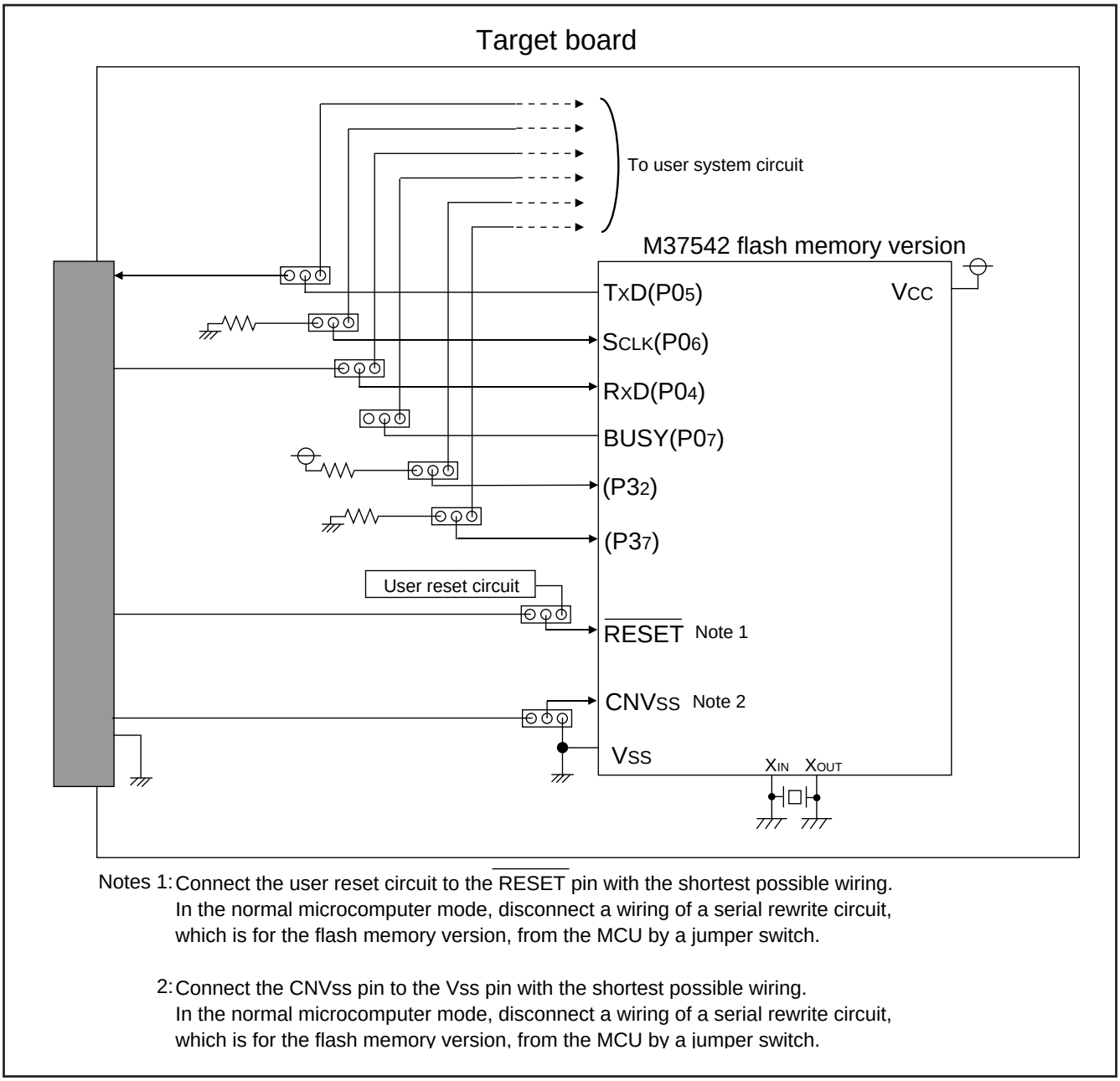

Note. Connect the oscillation circuit to XIN and XOUT.

**(Package type: PRDP0032BA-A)**

**Fig. 114 Pin connection diagram in standard serial I/O mode 2 (PRDP0032BA-A package)**

## • Standard serial I/O mode 2

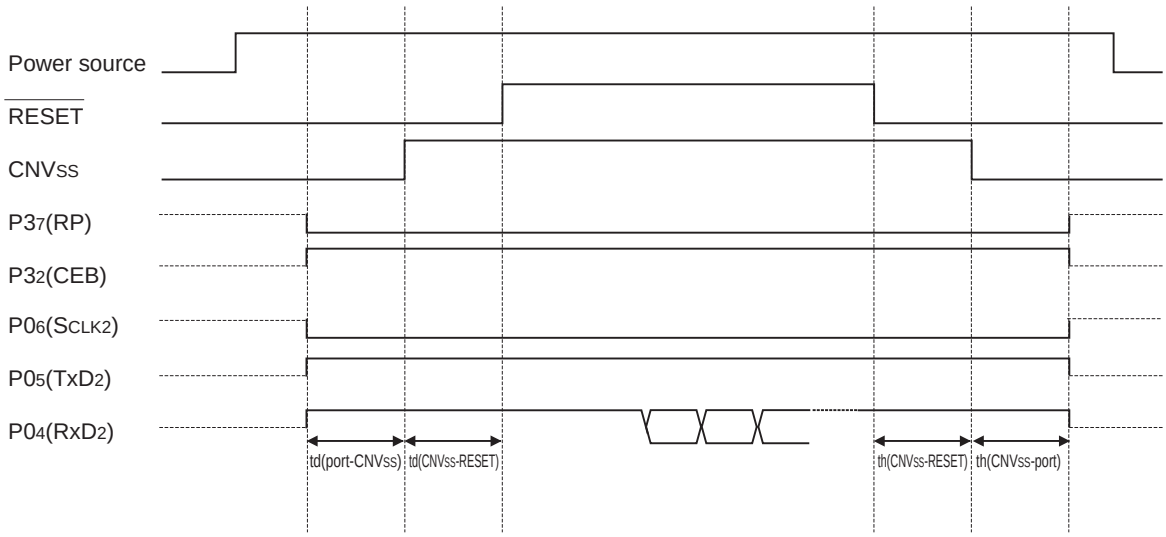
Figure 115 shows the handling example of control pins on the user system board when the standard serial I/O mode 2 is used.
Refer to the serial programmer manual of your programmer to handle pins controlled by the programmer.




Notes 1: Connect the user reset circuit to the $\overline{\text{RESET}}$ pin with the shortest possible wiring.
In the normal microcomputer mode, disconnect a wiring of a serial rewrite circuit, which is for the flash memory version, from the MCU by a jumper switch.

2: Connect the CNVss pin to the Vss pin with the shortest possible wiring.
In the normal microcomputer mode, disconnect a wiring of a serial rewrite circuit, which is for the flash memory version, from the MCU by a jumper switch.

**Fig. 115 Handling example of control pins in standard serial I/O mode 2**




| Symbol | Ratings | | | Unit |
|---|---|---|---|---|
| | Min. | Typ. | Max. | |
| td(port-CNVss) | 1 | - | - | ms |
| td(CNVss-RESET) | 1 | - | - | ms |
| th(RESET-CNVss) | 1 | - | - | ms |
| th(CNVss-port) | 1 | - | - | ms |

Note: In the standard serial I/O2, set P06 and P07 as follows;
P06: input “L” level.
P07: BUSY signal output pin. Keep open.

**Fig. 116 Timing diagram in standard serial I/O mode 2**

# ELECTRICAL CHARACTERISTICS

## 1.Absolute Maximum Ratings

**Table 15 Absolute maximum ratings**

| Symbol | Parameter | Conditions | Ratings | Unit |
|---|---|---|---|---|
| $V_{CC}$ | Power source voltage | All voltages are based on $V_{SS}$. When an input voltage is measured, output transistors are cut off. | –0.3 to 6.5 | V |
| $V_I$ | Input voltage<br>$P0_0$–$P0_7$, $P1_0$–$P1_4$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $V_{REF}$ | | –0.3 to $V_{CC}$ + 0.3 | V |
| $V_I$ | Input voltage $\overline{RESET}$, $X_{IN}$ | | –0.3 to $V_{CC}$ + 0.3 | V |
| $V_I$ | Input voltage $CNV_{SS}$ | | –0.3 to $V_{CC}$ + 0.3 | V |
| $V_O$ | Output voltage<br>$P0_0$–$P0_7$, $P1_0$–$P1_4$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $X_{OUT}$ | | –0.3 to $V_{CC}$ + 0.3 | V |
| $P_d$ | Power dissipation | Ta = 25°C | 300 (**Note**) | mW |
| $T_{opr}$ | Operating temperature | | –20 to 85 | °C |
| $T_{stg}$ | Storage temperature | | –40 to 125 | °C |

**Note:** 200 mW for the PLQP0032GB-A package product.

## Recommended Operating Conditions

**Table 16 Recommended operating conditions (1)**

**(FLASH ROM version: $V_{CC}$ = 2.7 to 5.5V, Mask ROM version: $V_{CC}$ = 2.2 to 5.5 V, $V_{SS}$ = 0 V, Ta = –20 to 85 °C, unless otherwise noted)**

| Symbol | Parameter | | | | Limits | | | Unit |
|---|---|---|---|---|---|---|---|---|
| | | | | | Min. | Typ. | Max. | |
| $V_{CC}$ | Power source voltage (ceramic) | (High-, Middle-speed mode) | f($X_{IN}$) = 8 MHz | Mask ROM | 4.0 | 5.0 | 5.5 | V |
| | | | | FLASH ROM | | | | |
| | | | f($X_{IN}$) = 4 MHz | Mask ROM | 2.4 | 5.0 | 5.5 | V |
| | | | | FLASH ROM | 2.7 | 5.0 | 5.5 | V |
| | | | f($X_{IN}$) = 2 MHz | Mask ROM | 2.2 | 5.0 | 5.5 | V |
| | | | | FLASH ROM | 2.7 | 5.0 | 5.5 | V |
| | | (Double-speed mode) | f($X_{IN}$) = 8 MHz | Mask ROM | 4.5 | 5.0 | 5.5 | V |
| | | | | FLASH ROM | | | | |
| | | | f($X_{IN}$) = 6.5 MHz | Mask ROM | 4.0 | 5.0 | 5.5 | V |
| | | | | FLASH ROM | | | | |
| | | | f($X_{IN}$) = 2 MHz | Mask ROM | 2.4 | 5.0 | 5.5 | V |
| | | | | FLASH ROM | 2.7 | 5.0 | 5.5 | V |
| | | | f($X_{IN}$) = 1 MHz | Mask ROM | 2.2 | 5.0 | 5.5 | V |
| | | | | FLASH ROM | 2.7 | 5.0 | 5.5 | V |
| | Power source voltage (RC) | (High-, Middle-speed mode) | f($X_{IN}$) = 4 MHz | Mask ROM | 4.0 | 5.0 | 5.5 | V |
| | | | | FLASH ROM | | | | |
| | | | f($X_{IN}$) = 2 MHz | Mask ROM | 2.4 | 5.0 | 5.5 | V |
| | | | | FLASH ROM | 2.7 | 5.0 | 5.5 | V |
| | | | f($X_{IN}$) = 1 MHz | Mask ROM | 2.2 | 5.0 | 5.5 | V |
| | | | | FLASH ROM | 2.7 | 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<br>$P0_0$–$P0_7$, $P1_0$–$P1_4$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ | | | | 0.8$V_{CC}$ | | $V_{CC}$ | V |
| $V_{IH}$ | “H” input voltage (TTL input level selected)<br>$P1_0$, $P1_2$, $P1_3$, $P3_6$, $P3_7$ (Note 1) | | | | 2.0 | | $V_{CC}$ | V |
| $V_{IH}$ | “H” input voltage<br>$\overline{RESET}$, $X_{IN}$ | | | | 0.8$V_{CC}$ | | $V_{CC}$ | V |
| $V_{IL}$ | “L” input voltage<br>$P0_0$–$P0_7$, $P1_0$–$P1_4$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ | | | | 0 | | 0.2$V_{CC}$ | V |
| $V_{IL}$ | “L” input voltage (TTL input level selected)<br>$P1_0$, $P1_2$, $P1_3$, $P3_6$, $P3_7$ (Note 1) | | | | 0 | | 0.8 | V |
| $V_{IL}$ | “L” input voltage<br>$\overline{RESET}$, $CNV_{SS}$ | | | | 0 | | 0.2$V_{CC}$ | V |
| $V_{IL}$ | “L” input voltage<br>$X_{IN}$ | | | | 0 | | 0.16$V_{CC}$ | V |
| $\sum I_{OH(peak)}$ | “H” total peak output current (Note 2)<br>$P0_0$–$P0_7$, $P1_0$–$P1_4$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ | | | | | | –80 | mA |
| $\sum I_{OL(peak)}$ | “L” total peak output current (Note 2)<br>$P1_0$–$P1_4$, $P2_0$–$P2_7$ | | | | | | 80 | mA |
| $\sum I_{OL(peak)}$ | “L” total peak output current (Note 2)<br>$P0_0$–$P0_7$, $P3_0$–$P3_7$ | | | | | | 80 | mA |
| $\sum I_{OH(avg)}$ | “H” total average output current (Note 2)<br>$P0_0$–$P0_7$, $P1_0$–$P1_4$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ | | | | | | –40 | mA |
| $\sum I_{OL(avg)}$ | “L” total average output current (Note 2)<br>$P1_0$–$P1_4$, $P2_0$–$P2_7$ | | | | | | 40 | mA |
| $\sum I_{OL(avg)}$ | “L” total average output current (Note 2)<br>$P0_0$–$P0_7$, $P3_0$–$P3_7$ | | | | | | 40 | mA |

**Note 1:** $V_{CC}$ = 4.0 to 5.5V

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

## Recommended Operating Conditions (continued)

**Table 17 Recommended operating conditions (2)**

**(FLASH ROM version: $V_{CC}$ = 2.7 to 5.5V, Mask ROM version: $V_{CC}$ = 2.2 to 5.5 V, $V_{SS}$ = 0 V, Ta = –20 to 85 °C, unless otherwise noted)**

| Symbol | Parameter | | Limits | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $I_{OH(peak)}$ | "H" peak output current (Note 1) | $P0_0$–$P0_7$, $P1_0$–$P1_4$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ | | | –10 | mA |
| $I_{OL(peak)}$ | "L" peak output current (Note 1) | $P0_0$–$P0_7$, $P3_0$–$P3_7$ (Drive capacity = "L")<br>$P1_0$–$P1_4$, $P2_0$–$P2_7$ | | | 10 | mA |
| $I_{OL(peak)}$ | "L" peak output current (Note 1) | $P0_0$–$P0_7$, $P3_0$–$P3_7$ (Drive capacity = "H") | | | 30 | mA |
| $I_{OH(avg)}$ | "H" average output current (Note 2) | $P0_0$–$P0_7$, $P1_0$–$P1_4$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ | | | –5 | mA |
| $I_{OL(avg)}$ | "L" average output current (Note 2) | $P0_0$–$P0_7$, $P3_0$–$P3_7$ (Drive capacity = "L")<br>$P1_0$–$P1_4$, $P2_0$–$P2_7$ | | | 5 | mA |
| $I_{OL(avg)}$ | "L" average output current (Note 2) | $P0_0$–$P0_7$, $P3_0$–$P3_7$ (Drive capacity = "H") | | | 15 | mA |
| $f(X_{IN})$ | Oscillation frequency (Note 3)<br>at ceramic oscillation or external clock input<br>Double-speed mode | Mask ROM: $V_{CC}$ = 4.5 to 5.5 V<br>FLASH ROM: $V_{CC}$ = 4.5 to 5.5 V | | | 8 | MHz |
| | Oscillation frequency (Note 3)<br>at ceramic oscillation or external clock input<br>Double-speed mode | Mask ROM: $V_{CC}$ = 4.0 to 5.5 V<br>FLASH ROM: $V_{CC}$ = 4.0 to 5.5 V | | | 6.5 | MHz |
| | Oscillation frequency (Note 3)<br>at ceramic oscillation or external clock input<br>Double-speed mode | Mask ROM: $V_{CC}$ = 2.4 to 5.5 V<br>FLASH ROM: $V_{CC}$ = 2.7 to 5.5 V | | | 2 | MHz |
| | Oscillation frequency (Note 3)<br>at ceramic oscillation or external clock input<br>Double-speed mode | Mask ROM: $V_{CC}$ = 2.2 to 5.5 V | | | 1 | MHz |
| | Oscillation frequency (Note 3)<br>at ceramic oscillation or external clock input<br>High-, Middle-speed mode | Mask ROM: $V_{CC}$ = 4.0 to 5.5 V<br>FLASH ROM: $V_{CC}$ = 4.0 to 5.5 V | | | 8 | MHz |
| | Oscillation frequency (Note 3)<br>at ceramic oscillation or external clock input<br>High-, Middle-speed mode | Mask ROM: $V_{CC}$ = 2.4 to 5.5 V<br>FLASH ROM: $V_{CC}$ = 2.7 to 5.5 V | | | 4 | MHz |
| | Oscillation frequency (Note 3)<br>at ceramic oscillation or external clock input<br>High-, Middle-speed mode | Mask ROM: $V_{CC}$ = 2.2 to 5.5 V | | | 2 | MHz |
| | Oscillation frequency (Note 3)<br>at RC oscillation<br>High-, Middle-speed mode | Mask ROM: $V_{CC}$ = 4.0 to 5.5 V<br>FLASH ROM: $V_{CC}$ = 4.0 to 5.5 V | | | 4 | MHz |
| | Oscillation frequency (Note 3)<br>at RC oscillation<br>High-, Middle-speed mode | Mask ROM: $V_{CC}$ = 2.4 to 5.5 V<br>FLASH ROM: $V_{CC}$ = 2.7 to 5.5 V | | | 2 | MHz |
| | Oscillation frequency (Note 3)<br>at RC oscillation<br>High-, Middle-speed mode | Mask ROM: $V_{CC}$ = 2.2 to 5.5 V | | | 1 | MHz |

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

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

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

## Electrical Characteristics

**Table 18 Electrical characteristics (1)**
**(FLASH ROM version: $V_{CC}$ = 2.7 to 5.5V, Mask ROM version: $V_{CC}$ = 2.2 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. | |
| $V_{OH}$ | "H" output voltage<br>$P0_0$–$P0_7$, $P1_0$–$P1_4$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ (Note 1) | $I_{OH}$ = –5 mA<br>$V_{CC}$ = 4.0 to 5.5 V | $V_{CC}$–1.5 | | | V |
| | | $I_{OH}$ = –1.0 mA<br>Mask ROM: $V_{CC}$ = 2.2 to 5.5 V<br>FLASH ROM: $V_{CC}$ = 2.7 to 5.5 V | $V_{CC}$–1.0 | | | V |
| $V_{OL}$ | "L" output voltage<br>$P0_0$–$P0_7$, $P3_0$–$P3_7$ (Drive capacity = "L")<br>$P1_0$–$P1_4$, $P2_0$–$P2_7$ | $I_{OL}$ = 5 mA<br>$V_{CC}$ = 4.0 to 5.5 V | | | 1.5 | V |
| | | $I_{OL}$ = 1.5 mA<br>$V_{CC}$ = 4.0 to 5.5 V | | | 0.3 | V |
| | | $I_{OL}$ = 1.0 mA<br>Mask ROM: $V_{CC}$ = 2.2 to 5.5 V<br>FLASH ROM: $V_{CC}$ = 2.7 to 5.5 V | | | 1.0 | V |
| $V_{OL}$ | "L" output voltage<br>$P0_0$–$P0_7$, $P3_0$–$P3_7$ (Drive capacity = "H") | $I_{OL}$ = 15 mA<br>$V_{CC}$ = 4.0 to 5.5 V | | | 2.0 | V |
| | | $I_{OL}$ = 1.5 mA<br>$V_{CC}$ = 4.0 to 5.5 V | | | 0.3 | V |
| | | $I_{OL}$ = 1.0 mA<br>Mask ROM: $V_{CC}$ = 2.2 to 5.5 V<br>FLASH ROM: $V_{CC}$ = 2.7 to 5.5 V | | | 1.0 | V |
| $V_{T+}$–$V_{T-}$ | Hysteresis<br>$CNTR_0$, $INT_0$, $INT_1$, $CAP_0$, $CAP_1$ (Note 2)<br>$P0_0$–$P0_7$ (Note 3) | | | 0.4 | | V |
| $V_{T+}$–$V_{T-}$ | Hysteresis<br>$RxD_0$, $SCLK_0$, $RxD_1$, $SCLK_1$ | | | 0.5 | | V |
| $V_{T+}$–$V_{T-}$ | Hysteresis<br>$\overline{RESET}$ | | | 0.5 | | V |
| $I_{IH}$ | "H" input current<br>$P0_0$–$P0_7$, $P1_0$–$P1_4$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ | $V_I$ = $V_{CC}$<br>(Pin floating. Pull up transistors "off") | | | 5.0 | µA |
| $I_{IH}$ | "H" input current<br>$\overline{RESET}$ | $V_I$ = $V_{CC}$ | | | 5.0 | µA |
| $I_{IH}$ | "H" input current<br>$X_{IN}$ | $V_I$ = $V_{CC}$ | | 4.0 | | µA |
| $I_{IL}$ | "L" input current<br>$P0_0$–$P0_7$, $P1_0$–$P1_4$, $P2_0$–$P2_7$, $P3_0$–$P3_7$ | $V_I$ = $V_{SS}$<br>(Pin floating. Pull up transistors "off") | | | –5.0 | µA |
| $I_{IL}$ | "L" input current<br>$\overline{RESET}$ | $V_I$ = $V_{SS}$ | | | –5.0 | µA |
| $I_{IL}$ | "L" input current<br>$X_{IN}$ | $V_I$ = $V_{SS}$ | | –4.0 | | µA |
| $I_{IL}$ | "L" input current<br>$P0_0$–$P0_7$, $P3_0$–$P3_7$ | $V_I$ = $V_{SS}$<br>(Pull up transistors "on") | | –0.2 | –0.5 | mA |
| $V_{RAM}$ | RAM hold voltage | When clock stopped | 2.0 | | 5.5 | V |
| $R_{OSC}$ | On-chip oscillator oscillation frequency | $V_{CC}$ = 5.0 V, Ta = 25 °C | 1000 | 2000 | 3000 | kHz |
| $D_{OSC}$ | Oscillation stop detection circuit detection frequency | $V_{CC}$ = 5.0 V, Ta = 25 °C | 62.5 | 125 | 187.5 | kHz |

**Notes 1:** $P1_1$ is measured when the $P1_1/TxD_1$ P-channel output disable bit of the UART1 control register (bit 4 of address $001B_{16}$) is "0".
**2:** $RxD_1$, $SCLK_1$, $INT_0$, and $INT_1$ ($P3_6$ selected) have hysteresises only when bits 0 to 2 of the port P1P3 control register are set to "0" (CMOS level).
**3:** It is available only when operating key-on wake up.

## Electrical Characteristics (continued)

**Table 19 Electrical characteristics (2)**
**(FLASH ROM version: $V_{CC}$ = 2.7 to 5.5V, Mask ROM version: $V_{CC}$ = 2.2 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 | $f(X_{IN})$ = 8 MHz<br>Output transistors “off” | Double-speed mode | Mask ROM | | 5.5 | 9.0 | mA |
| | | | | FLASH ROM | | 4.8 | 7.5 | mA |
| | | | High-speed mode | Mask ROM | | 3.5 | 6.5 | mA |
| | | | | FLASH ROM | | 3.0 | 5.5 | mA |
| | | | Middle-speed mode | Mask ROM | | 2.0 | 5.0 | mA |
| | | | | FLASH ROM | | 1.7 | 4.2 | mA |
| | | $f(X_{IN})$ = 2 MHz,<br>Mask ROM: $V_{CC}$ = 2.2 V<br>FLASH ROM: $V_{CC}$ = 2.7 V<br>Output transistors “off” | High-speed mode | Mask ROM | | 0.4 | 1.2 | mA |
| | | | | FLASH ROM | | 1.0 | 2.8 | mA |
| | | On-chip oscillator operation mode,<br>Output transistors “off” | Frequency/1 | Mask ROM | | 1.5 | 3.2 | mA |
| | | | | FLASH ROM | | 1.4 | 2.4 | mA |
| | | | Frequency/2 | Mask ROM | | 0.9 | 2.2 | mA |
| | | | | FLASH ROM | | 1.0 | 1.9 | mA |
| | | | Frequency/8 | Mask ROM | | 0.35 | 1.0 | mA |
| | | | | FLASH ROM | | 0.65 | 1.3 | mA |
| | | | Frequency/128 | Mask ROM | | 0.2 | 0.6 | mA |
| | | | | FLASH ROM | | 0.55 | 1.0 | mA |
| | | $f(X_{IN})$ = 8 MHz (in WIT state),<br>functions except timer 1 disabled,<br>Output transistors “off” | | Mask ROM | | 1.6 | 3.2 | mA |
| | | | | FLASH ROM | | 1.2 | 2.6 | mA |
| | | $f(X_{IN})$ = 2 MHz,<br>Mask ROM: $V_{CC}$ = 2.2 V<br>FLASH ROM: $V_{CC}$ = 2.7 V<br>(in WIT state),<br>functions except timer 1 disabled,<br>Output transistors “off” | | Mask ROM | | 0.2 | | mA |
| | | | | FLASH ROM | | 0.6 | | mA |
| | | On-chip oscillator operation mode,<br>(in WIT state),<br>functions except timer 1 disabled,<br>Output transistors “off” | | Mask ROM | | 0.2 | 0.6 | mA |
| | | | | FLASH ROM | | 0.12 | 0.4 | mA |
| | | Increment when A/D conversion is executed<br>$f(X_{IN})$ = 8 MHz, $V_{CC}$ = 5 V | | Mask ROM | | 0.5 | | mA |
| | | | | FLASH ROM | | 0.5 | | mA |
| | | All oscillation stopped<br>(in STP state)<br>Output transistors “off” | Ta = 25 °C | Mask ROM | | 0.1 | 1.0 | µA |
| | | | | FLASH ROM | | 0.55 | 3.0 | µA |
| | | | Ta = 85 °C | Mask ROM | | | 10 | µA |
| | | | | FLASH ROM | | | 10 | µA |

**Note:** Increment when A/D conversion is executed includes the reference power source input current ($IV_{REF}$).

## A/D Converter Characteristics

**Table 20 A/D Converter characteristics**

**($V_{CC}$ = 2.7 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 |
| — | Absolute accuracy | Ta = 25 °C | Mask ROM | | | ± 3 | LSB |
| | | $V_{CC}$ = $V_{REF}$ = 2.7 to 5.5 V | FLASH ROM | | | ± 4 | |
| $t_{CONV}$ | Conversion time | AD conversion clock = $f(X_{IN})/2$ | | | | 122 | $tc(X_{IN})$ |
| | | AD conversion clock = $f(X_{IN})$ | | | | 61 | |
| $R_{LADDER}$ | 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 | 90 | 120 | |
| $I_{I(AD)}$ | A/D port input current | | | | | 5.0 | μA |

**Note:** AD conversion accuracy may be low under the following conditions;

(1) When the $V_{REF}$ voltage is set to be lower than the $V_{CC}$ voltage, an analog circuit in this microcomputer is affected by noise. The accuracy is lower than the case the $V_{REF}$ voltage is the same as $V_{CC}$ voltage.

(2) When the $V_{REF}$ voltage is 3.0 V or less at the low temperature, the AD conversion accuracy may be very lower than at room temperature. When system is used at low temperature, that $V_{REF}$ is 3.0 V or more is recommended.

## Electrical Characteristics of 7542 Group Flash Memory

**Table 21 Electrical Characteristics of 7542 Group Flash Memory**

| Symbol | Parameter | | Test conditions | Limits | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | |
| ——— | Program/Erase endurance (Note 1) | | | 100 | | | times |
| ——— | Byte program time | | | | 50 | 400 | μs |
| ——— | Block erase time | 2Kbyte-block | | | 0.2 | 9 | s |
| | | 8Kbyte-block | | | 0.4 | 9 | s |
| | | 16Kbyte-block | | | 0.7 | 9 | s |
| $t_{d(SR-ES)}$ | Time delay from suspend request until erase suspend | | | | | 8 | ms |
| ——— | Erase suspend request interval | | | 10 | | | ms |
| ——— | Program, erase voltage | | | 2.7 | | 5.5 | V |
| ——— | Read voltage | | | 2.7 | | 5.5 | V |
| ——— | Program, erase temperature | | | 0 | | 60 | °C |
| ——— | Data hold time | | Ta = 55 °C | 20 | | | year |

Note 1. Definition of program and erase

The program and erase endurance shows an erase endurance for every block.

If the program and erase endurance is “n” times (n = 100), “n” times erase can be performed for every block.

For example, if performing 1-byte write to the distinct addresses on Block A of 2Kbyte block 2048 times and then erasing that block, program and erase endurance is counted as one time.

However, do not perform multiple programs to the same address for one time erase. (disable overwriting).

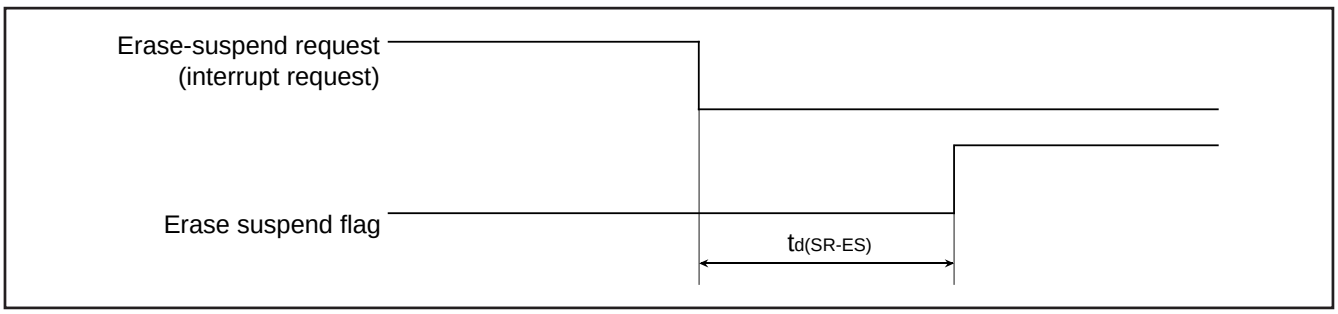



**Fig. 117 Time delay from suspend request until erase suspend**

## Timing Requirements

**Table 22 Timing requirements (1)**

**(FLASH ROM version: $V_{CC}$ = 4.0 to 5.5V, Mask ROM version: $V_{CC}$ = 4.0 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 | 2 | | | µs |
| $t_{C(XIN)}$ | External clock input cycle time | 125 | | | ns |
| $t_{WH(XIN)}$ | External clock input “H” pulse width | 50 | | | ns |
| $t_{WL(XIN)}$ | External clock input “L” pulse width | 50 | | | ns |
| $t_{C(CNTR0)}$ | $CNTR_0$ input cycle time | 200 | | | ns |
| $t_{WH(CNTR0)}$ | $CNTR_0$, $INT_0$, $INT_1$, $CAP_0$, $CAP_1$ input “H” pulse width (Note 1) | 80 | | | ns |
| $t_{WL(CNTR0)}$ | $CNTR_0$, $INT_0$, $INT_1$, $CAP_0$, $CAP_1$ input “L” pulse width (Note 1) | 80 | | | ns |
| $t_{C(SCLK1)}$ | Serial I/O1, serial I/O2 clock input cycle time (Note 2) | 800 | | | ns |
| $t_{WH(SCLK1)}$ | Serial I/O1, serial I/O2 clock input “H” pulse width (Note 2) | 370 | | | ns |
| $t_{WL(SCLK1)}$ | Serial I/O1, serial I/O2 clock input “L” pulse width (Note 2) | 370 | | | ns |
| $t_{su(RxD1–SCLK1)}$ | Serial I/O1, serial I/O2 input set up time | 220 | | | ns |
| $t_{h(SCLK1–RxD1)}$ | Serial I/O1, serial I/O2 input hold time | 100 | | | ns |

**Notes 1:** As for $CAP_0$, $CAP_1$, it is the value when noise filter is not used.

**2:** In this time, bit 6 of the serial I/O1 control register (address $001A_{16}$) is set to “1” (clock synchronous serial I/O is selected).
When bit 6 of the serial I/O1 control register is “0” (clock asynchronous serial I/O is selected), the rating values are divided by 4.
In this time, bit 6 of the serial I/O2 control register (address $0030_{16}$) is set to “1” (clock synchronous serial I/O is selected).
When bit 6 of the serial I/O2 control register is “0” (clock asynchronous serial I/O is selected), the rating values are divided by 4.

**Table 23 Timing requirements (2)**

**(FLASH ROM version: $V_{CC}$ = 2.7 to 5.5V, Mask ROM version: $V_{CC}$ = 2.4 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 | 2 | | | µs |
| $t_{C(XIN)}$ | External clock input cycle time | 250 | | | ns |
| $t_{WH(XIN)}$ | External clock input “H” pulse width | 100 | | | ns |
| $t_{WL(XIN)}$ | External clock input “L” pulse width | 100 | | | ns |
| $t_{C(CNTR0)}$ | $CNTR_0$ input cycle time | 500 | | | ns |
| $t_{WH(CNTR0)}$ | $CNTR_0$, $INT_0$, $INT_1$, $CAP_0$, $CAP_1$ input “H” pulse width (Note 1) | 230 | | | ns |
| $t_{WL(CNTR0)}$ | $CNTR_0$, $INT_0$, $INT_1$, $CAP_0$, $CAP_1$ input “L” pulse width (Note 1) | 230 | | | ns |
| $t_{C(SCLK1)}$ | Serial I/O1, serial I/O2 clock input cycle time (Note 2) | 2000 | | | ns |
| $t_{WH(SCLK1)}$ | Serial I/O1, serial I/O2 clock input “H” pulse width (Note 2) | 950 | | | ns |
| $t_{WL(SCLK1)}$ | Serial I/O1, serial I/O2 clock input “L” pulse width (Note 2) | 950 | | | ns |
| $t_{su(RxD1–SCLK1)}$ | Serial I/O1, serial I/O2 input set up time | 400 | | | ns |
| $t_{h(SCLK1–RxD1)}$ | Serial I/O1, serial I/O2 input hold time | 200 | | | ns |

**Notes 1:** As for $CAP_0$, $CAP_1$, it is the value when noise filter is not used.

**2:** In this time, bit 6 of the serial I/O1 control register (address $001A_{16}$) is set to “1” (clock synchronous serial I/O is selected).
When bit 6 of the serial I/O1 control register is “0” (clock asynchronous serial I/O1 is selected), the rating values are divided by 4.
In this time, bit 6 of the serial I/O2 control register (address $0030_{16}$) is set to “1” (clock synchronous serial I/O is selected).
When bit 6 of the serial I/O2 control register is “0” (clock asynchronous serial I/O is selected), the rating values are divided by 4.

**Table 24 Timing requirements (3)**

**(Mask ROM version: Vcc = 2.2 to 5.5 V, Vss = 0 V, Ta = –20 to 85 °C, unless otherwise noted) (This is only for the mask ROM version.)**

| Symbol | Parameter | Limits | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | |
| $t_{W(\overline{RESET})}$ | Reset input “L” pulse width | 2 | | | μs |
| $t_{C(XIN)}$ | External clock input cycle time | 500 | | | ns |
| $t_{WH(XIN)}$ | External clock input “H” pulse width | 200 | | | ns |
| $t_{WL(XIN)}$ | External clock input “L” pulse width | 200 | | | ns |
| $t_{C(CNTR0)}$ | CNTR0 input cycle time | 1000 | | | ns |
| $t_{WH(CNTR0)}$ | CNTR0, INT0, INT1, CAP0, CAP1 input “H” pulse width (Note 1) | 460 | | | ns |
| $t_{WL(CNTR0)}$ | CNTR0, INT0, INT1, CAP0, CAP1 input “L” pulse width (Note 1) | 460 | | | ns |
| $t_{C(SCLK1)}$ | Serial I/O1, serial I/O2 clock input cycle time (Note 2) | 4000 | | | ns |
| $t_{WH(SCLK1)}$ | Serial I/O1, serial I/O2 clock input “H” pulse width (Note 2) | 1900 | | | ns |
| $t_{WL(SCLK1)}$ | Serial I/O1, serial I/O2 clock input “L” pulse width (Note 2) | 1900 | | | ns |
| $t_{su(RxD1–SCLK1)}$ | Serial I/O1, serial I/O2 input set up time | 800 | | | ns |
| $t_{h(SCLK1–RxD1)}$ | Serial I/O1, serial I/O2 input hold time | 400 | | | ns |

**Notes 1:** As for CAP0, CAP1, it is the value when noise filter is not used.

**2:** In this time, bit 6 of the serial I/O1 control register (address $001A_{16}$) is set to “1” (clock synchronous serial I/O is selected).
When bit 6 of the serial I/O1 control register is “0” (clock asynchronous serial I/O1 is selected), the rating values are divided by 4.
In this time, bit 6 of the serial I/O2 control register (address $0030_{16}$) is set to “1” (clock synchronous serial I/O is selected).
When bit 6 of the serial I/O2 control register is “0” (clock asynchronous serial I/O is selected), the rating values are divided by 4.

## Switching Characteristics

**Table 25 Switching characteristics (1)**

**(FLASH ROM version: Vcc = 4.0 to 5.5V, Mask ROM version: Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = –20 to 85 °C, unless otherwise noted)**

| Symbol | Parameter | Limits | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | |
| $t_{WH}(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output “H” pulse width | $t_C(S_{CLK1})/2–30$ | | | ns |
| $t_{WL}(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output “L” pulse width | $t_C(S_{CLK1})/2–30$ | | | ns |
| $t_d(S_{CLK1}–TxD1)$ | Serial I/O1, serial I/O2 output delay time | | | 140 | ns |
| $t_v(S_{CLK1}–TxD1)$ | Serial I/O1, serial I/O2 output valid time | –30 | | | ns |
| $t_r(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output rising time | | | 30 | ns |
| $t_f(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output falling time | | | 30 | ns |
| $t_r$(CMOS) | CMOS output rising time (Note 1) | | 10 | 30 | ns |
| $t_f$(CMOS) | CMOS output falling time (Note 1) | | 10 | 30 | ns |

**Note 1:** Pin XOUT is excluded.

**Table 26 Switching characteristics (2)**

**(FLASH ROM version: Vcc = 2.7 to 5.5V, Mask ROM version: Vcc = 2.4 to 5.5 V, Vss = 0 V, Ta = –20 to 85 °C, unless otherwise noted)**

| Symbol | Parameter | Limits | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | |
| $t_{WH}(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output “H” pulse width | $t_C(S_{CLK1})/2–50$ | | | ns |
| $t_{WL}(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output “L” pulse width | $t_C(S_{CLK1})/2–50$ | | | ns |
| $t_d(S_{CLK1}–T_xD1)$ | Serial I/O1, serial I/O2 output delay time | | | 350 | ns |
| $t_v(S_{CLK1}–T_xD1)$ | Serial I/O1, serial I/O2 output valid time | –30 | | | ns |
| $t_r(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output rising time | | | 50 | ns |
| $t_f(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output falling time | | | 50 | ns |
| $t_r$(CMOS) | CMOS output rising time (Note 1) | | 20 | 50 | ns |
| $t_f$(CMOS) | CMOS output falling time (Note 1) | | 20 | 50 | ns |

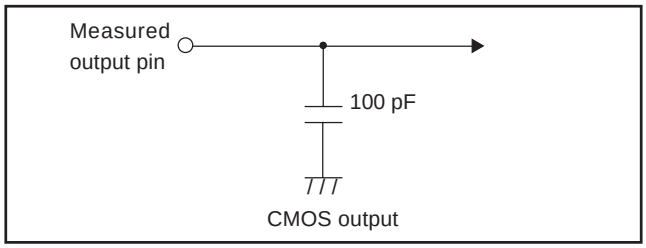
**Note 1:** Pin XOUT is excluded.

**Table 27 Switching characteristics (3)**

**(Vcc = 2.2 to 5.5 V, Vss = 0 V, Ta = –20 to 85 °C, unless otherwise noted)**

| Symbol | Parameter | Limits | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | |
| $t_{WH}(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output “H” pulse width | $t_C(S_{CLK1})/2–70$ | | | ns |
| $t_{WL}(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output “L” pulse width | $t_C(S_{CLK1})/2–70$ | | | ns |
| $t_d(S_{CLK1}–T_xD1)$ | Serial I/O1, serial I/O2 output delay time | | | 450 | ns |
| $t_v(S_{CLK1}–T_xD1)$ | Serial I/O1, serial I/O2 output valid time | –30 | | | ns |
| $t_r(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output rising time | | | 70 | ns |
| $t_f(S_{CLK1})$ | Serial I/O1, serial I/O2 clock output falling time | | | 70 | ns |
| $t_r$(CMOS) | CMOS output rising time (Note 1) | | 25 | 70 | ns |
| $t_f$(CMOS) | CMOS output falling time (Note 1) | | 25 | 70 | ns |

**Note 1:** Pin XOUT is excluded.




**Switching characteristics measurement circuit diagram**

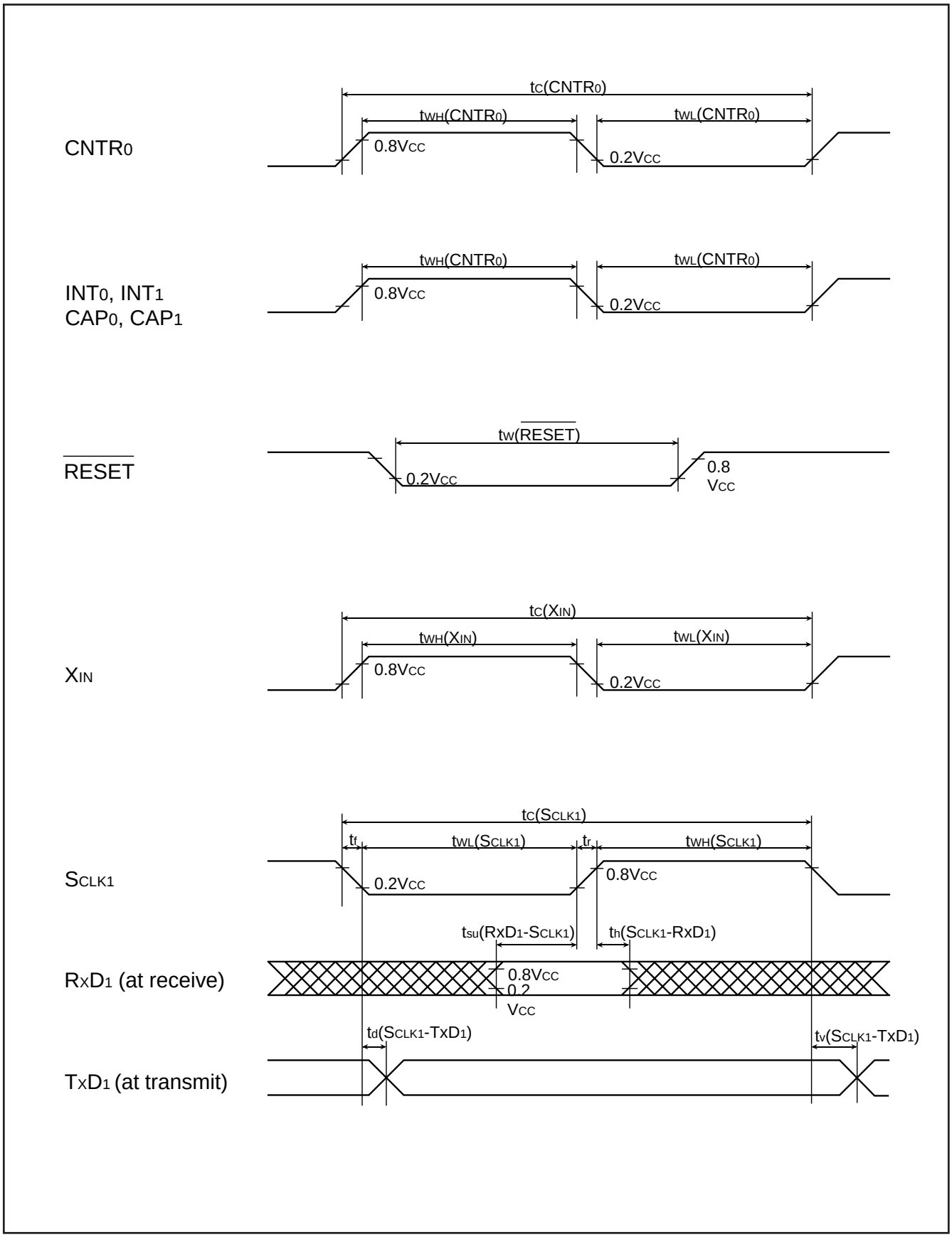
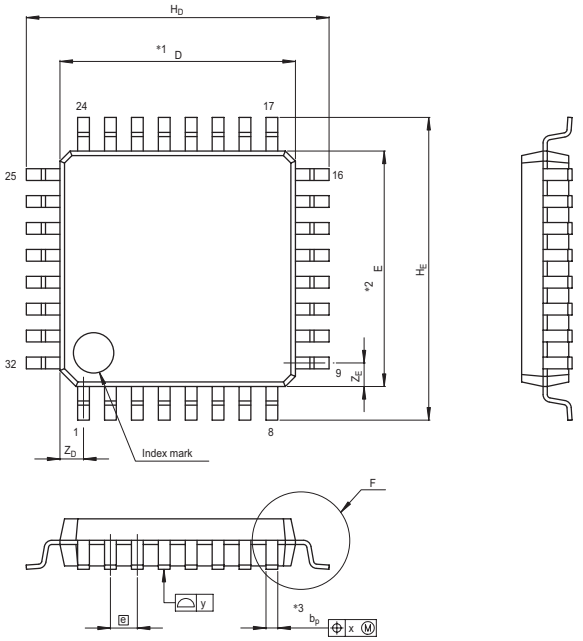


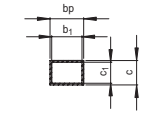

**Fig. 118 Timing chart**

## PACKAGE OUTLINE

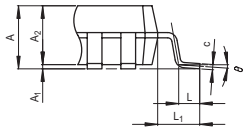
| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-LQFP32-7x7-0.80 | PLQP0032GB-A | 32P6U-A | 0.2g |






Terminal cross section

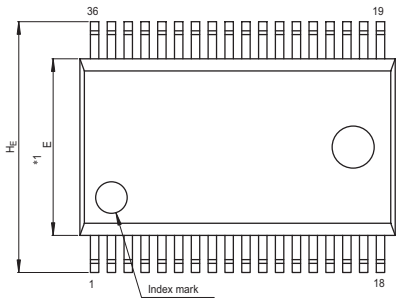


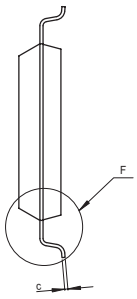

Detail F

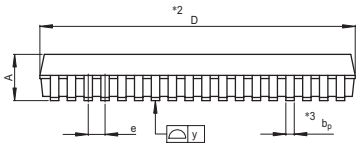
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

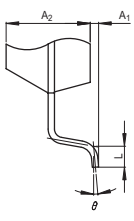
| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 6.9 | 7.0 | 7.1 |
| E | 6.9 | 7.0 | 7.1 |
| $A_2$ | —— | 1.4 | —— |
| $H_D$ | 8.8 | 9.0 | 9.2 |
| $H_E$ | 8.8 | 9.0 | 9.2 |
| A | —— | —— | 1.7 |
| $A_1$ | 0 | 0.1 | 0.2 |
| $b_p$ | 0.32 | 0.37 | 0.42 |
| $b_1$ | —— | 0.35 | —— |
| c | 0.09 | 0.145 | 0.20 |
| $c_1$ | —— | 0.125 | —— |
| θ | 0° | —— | 8° |
| 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 | —— |

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



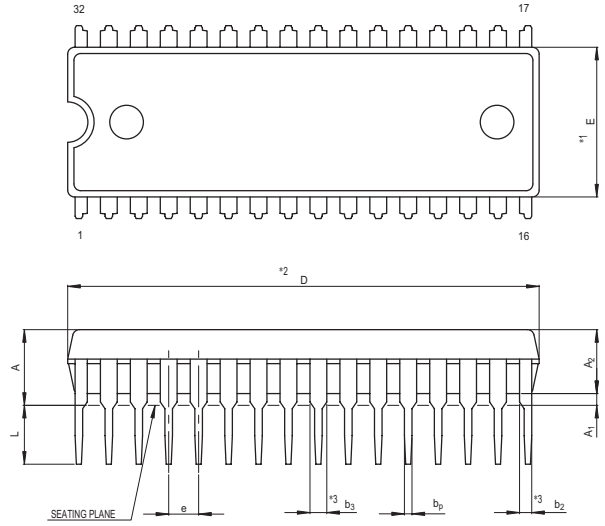




Detail F

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 |

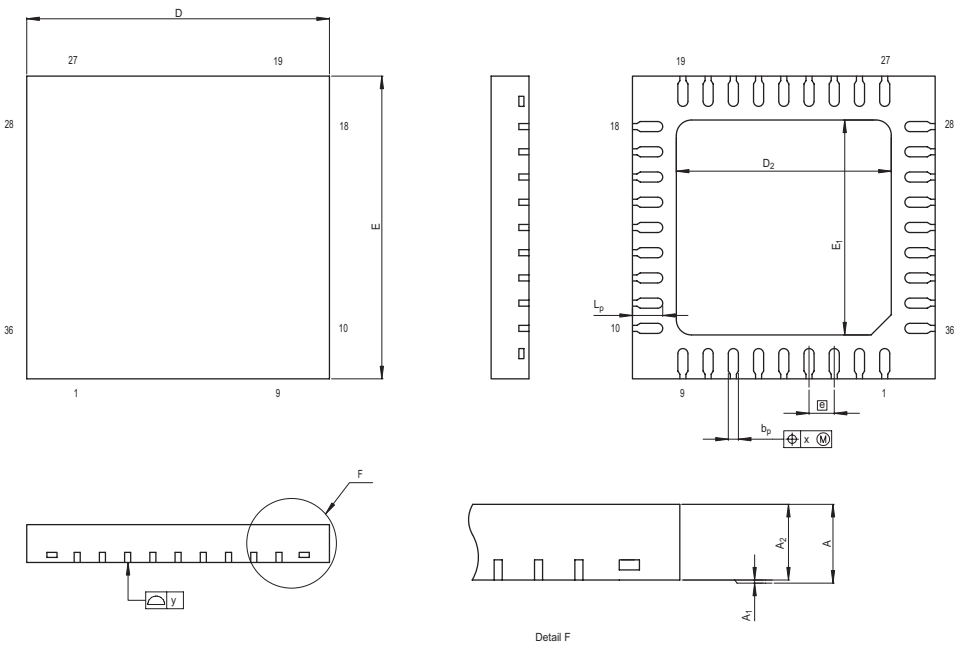
| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-SDIP32-8.9x28-1.78 | PRDP0032BA-A | 32P4B | 2.2g |




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 |
| e1 | 9.86 | 10.16 | 10.46 |
| D | 27.8 | 28.0 | 28.2 |
| E | 8.75 | 8.9 | 9.05 |
| A | —— | —— | 5.08 |
| $A_1$ | 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 | —— | —— |

| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-HWQFN36-6x6-0.50 | PWQN0036KA-A | 36PJW-A | 0.07g |




| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 5.9 | 6.0 | 6.1 |
| E | 5.9 | 6.0 | 6.1 |
| $A_2$ | —— | 0.75 | —— |
| A | —— | —— | 0.8 |
| $A_1$ | 0 | 0 | 0.05 |
| $b_p$ | 0.15 | 0.2 | 0.25 |
| e | —— | 0.5 | —— |
| $L_p$ | 0.5 | 0.6 | 0.7 |
| x | —— | —— | 0.05 |
| y | —— | —— | 0.05 |
| $D_2$ | —— | 4.26 | —— |
| $E_1$ | —— | 4.26 | —— |

# APPENDIX

## 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”.

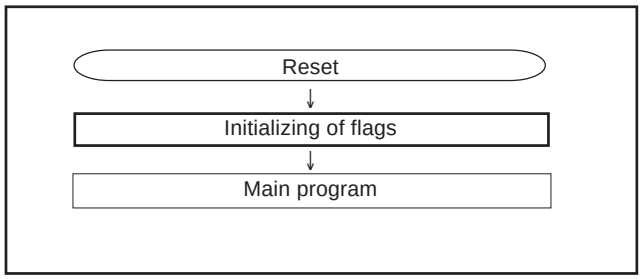



**Fig. 1 Initialization of processor status register**

(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.

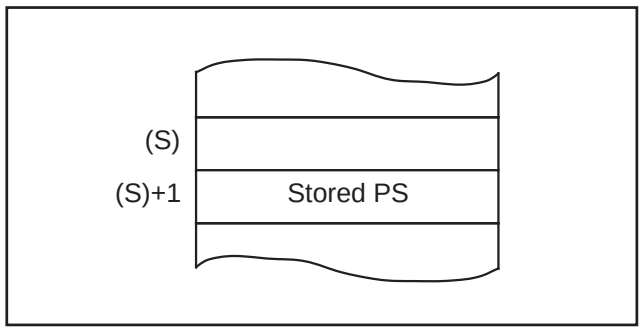



**Fig. 2 Stack memory contents after PHP instruction execution**

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.

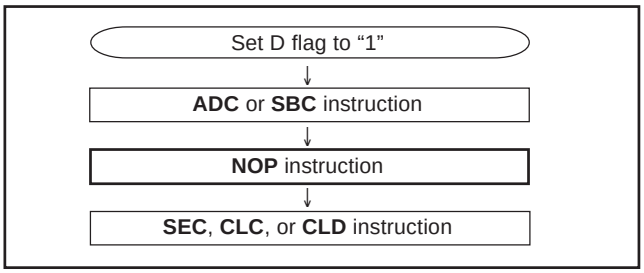



**Fig. 3 Status flag at decimal calculations**

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.

4.Multiplication and Division Instructions

(1) The index X mode (T) and the decimal mode (D) flags do not affect the MUL and DIV instruction.

(2) The execution of these instructions does not change the contents of the processor status register.

5. Read-modify-write instruction
Do not execute a read-modify-write instruction to the read invalid address (SFR).
The read-modify-write instruction operates in the following sequence: read one-byte of data from memory, modify the data, write the data back to original memory. The following instructions are classified as the read-modify-write instructions in the 740 Family.
(1) Bit management instructions: CLB, SEB
(2) Shift and rotate instructions: ASL, LSR, ROL, ROR, RRF
(3) Add and subtract instructions: DEC, INC
(4) Logical operation instructions (1's complement): COM

Add and subtract/logical operation instructions (ADC, SBC, AND, EOR, and ORA) when T flag = "1" operate in the way as the read-modify-write instruction. Do not execute the read invalid SFR.
<Reason>
When the read-modify-write instruction is executed to read invalid SFR, the instruction may cause the following consequence: the instruction reads unspecified data from the area due to the read invalid condition. Then the instruction modifies this unspecified data and writes the data to the area. The result will be random data written to the area or some unexpected event.

## NOTES ON PERIPHERAL FUNCTIONS
## Notes on I/O Ports

1. Setting of 32-pin version and PWQN0036KA-A package version
(1) Set direction registers of ports $P2_6$, $P2_7$, $P3_5$ and $P3_6$ to output.

(2) Select $P3_3$ for the $INT_1$ function by the INT1 input port selection bit (bit 2 of interrupt edge selection register (address $3A_{16}$)).

(3) Be sure to set $P3_6/INT_1$ input level selection bit (bit 1 of port P1P3 control register (address $17_{16}$)) to "0".

2. Port P0P3 drive capacity control register
The number of LED drive port (drive capacity is HIGH) is 8.

3. Pull-up control register
When using each port which built in pull-up resistor as an output port, the pull-up control bit of corresponding port becomes invalid, and pull-up resistor is not connected.
<Reason>
Pull-up control is effective only when each direction register is set to the input mode.

4. Notes in stand-by state
In stand-by state*1 for low-power dissipation, do not make input levels of an input port and an I/O port "undefined".
Pull-up (connect the port to Vcc) or pull-down (connect the port to Vss) these ports through a resistor.
When determining a resistance value, note the following points:
• External circuit
• Variation of output levels during the ordinary operation
When using a built-in pull-up resistor, note on varied current values:
• When setting as an input port : Fix its input level
• When setting as an output port : Prevent current from flowing out to external.
<Reason>
The output transistor becomes the OFF state, which causes the ports to be the high-impedance state. Note that the level becomes "undefined" depending on external circuits.
Accordingly, the potential which is input to the input buffer in a microcomputer is unstable in the state that input levels of an 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

5. 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

6. Direction register
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.

## Termination of Unused Pins

1. Terminate unused pins
Perform the following wiring at the shortest possible distance (20 mm or less) from microcomputer pins.
(1) I/O ports
Set the I/O ports for the input mode and connect each pin to $V_{CC}$ or $V_{SS}$ through each resistor of 1 kΩ to 10 kΩ. The port which can select a built-in pull-up resistor can also use the built-in pull-up resistor.
When using the I/O ports as the output mode, 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) I/O ports setting as input mode
[1] 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 (1) shown on the above "1. Terminate unused pins".

[2] Do not connect to $V_{CC}$ or $V_{SS}$ directly.
<Reason>
If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur.

[3] Do not connect multiple ports in a lump to $V_{CC}$ or $V_{SS}$ 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.

## Notes on Interrupts

1. Change of relevant register settings
When not requiring for the interrupt occurrence synchronous with the following case, take the sequence shown in Figure 4.
- When switching external interrupt active edge
- When switching interrupt sources of an interrupt vector address where two or more interrupt sources are allocated

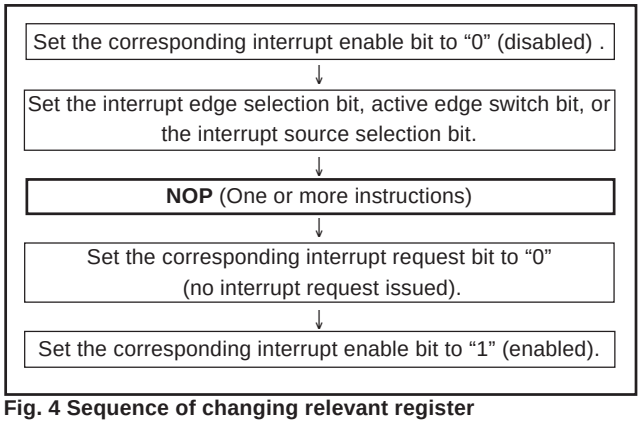



**Fig. 4 Sequence of changing relevant register**

<Reason>
When setting the followings, the interrupt request bit of the corresponding interrupt may be set to "1".
- When switching external interrupt active edge
  $INT_0$ interrupt edge selection bit
  (bit 0 of Interrupt edge selection register (address $3A_{16}$))
  $INT_1$ interrupt edge selection bit
  (bit 1 of Interrupt edge selection register)
  $CNTR_0$ active edge switch bit
  (bit 2 of timer X mode register (address $2B_{16}$))
  Capture 0 interrupt edge selection bit
  (bits 1 and 0 of capture mode register (address $20_{16}$))
  Capture 1 interrupt edge selection bit
  (bits 3 and 2 of capture mode register)

2. Check of interrupt request bit
When executing the **BBC** or **BBS** instruction to determine an interrupt request bit immediately after this bit is set to "0", take the following sequence.
<Reason>
If the BBC or BBS instruction is executed immediately after an interrupt request bit is cleared to "0", the value of the interrupt request bit before being cleared to "0" is read.

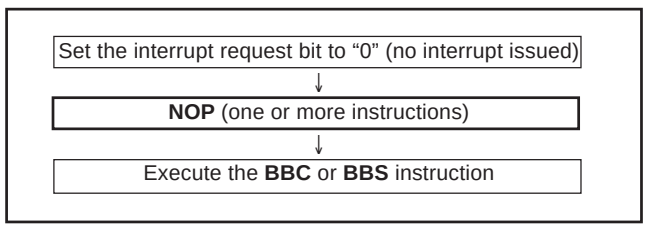



**Fig. 5 Sequence of check of interrupt request bit**

3. Interrupt discrimination bit
Use an LDM instruction to clear to “0” an interrupt discrimination bit.
LDM #%0000XXXX, $0B
Set the following values to “X”
“0”: an interrupt discrimination bit to clear
“1”: other interrupt discrimination bits
Ex.) When a key-on wakeup interrupt discrimination bit is cleared;
LDM #%00001110 and $0B.

4. Interrupt discrimination bit and interrupt request bit
For key-on wakeup, UART1 bus collision detection, A/D conversion and Timer 1 interrupt, even if each interrupt valid bit (interrupt source set register (address $0A_{16}$)) is set “0: Invalid”, each interrupt discrimination bit (interrupt source discrimination register (address $0B_{16}$)) is set to “1: interrupt occurs” when corresponding interrupt request occurs.
But corresponding interrupt request bit (interrupt request registers 1, 2 (addresses $3C_{16}$, $3D_{16}$) is not affected.

## Notes on Timers

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

2. When a count source of timer X, timer A or timer B is switched, stop a count of the timer.

## Notes on Timer X

1. $CNTR_0$ interrupt active edge selection
$CNTR_0$ interrupt active edge depends on the $CNTR_0$ active edge switch bit (bit 2 of timer X mode register (address $2B_{16}$)).
When this bit is “0”, the $CNTR_0$ interrupt request bit is set to “1” at the falling edge of $CNTR_0$ pin input signal. When this bit is “1”, the $CNTR_0$ interrupt request bit is set to “1” at the rising edge of $CNTR_0$ pin input signal.

2. Timer X count source selection
The f($X_{IN}$) (frequency not divided) can be selected by the timer X count source selection bits (bits 1 and 0 of timer count source set register (address $2A_{16}$)) only when the ceramic oscillation or the on-chip oscillator is selected.
Do not select it for the timer X count source at the RC oscillation.

3. Pulse output mode
Set the direction register of port $P1_4$, which is also used as $CNTR_0$ pin, to output.
When the $TX_{OUT}$ pin is used, set the direction register of port $P0_3$, which is also used as $TX_{OUT}$ pin, to output.

4. Pulse width measurement mode
Set the direction register of port $P1_4$, which is also used as $CNTR_0$ pin, to input.

## Notes on Timer A, B

1. Setting of timer value
When “1: Write to only latch” is set to the timer A (B) write control bit, written data to timer register is set to only latch even if timer is stopped or operating. Accordingly, in order to set the initial value for timer when it is stopped, set “0: Write to latch and timer simultaneously” to timer A (B) write control bit.

2. Read/write of timer A
Stop timer A to read/write its data in the following state;
$X_{IN}$ oscillation selected by clock division ratio selection bits (bits 7 and 6 of CPU mode register (address $3B_{16}$)), and the on-chip oscillator output is selected as the timer A count source.

3. Read/write of timer B
Stop timer B to read/write its data in the following state;
$X_{IN}$ oscillation selected by clock division ratio selection bits, the timer A underflow is selected as the timer B count source, and the on-chip oscillator output is selected as the timer A count source.

## Notes on Output Compare

1. When the selected source timer of each compare channel is stopped, written data to compare register is loaded to the compare latch simultaneously.

2. Do not write the same data to both of compare latch x0 (x=0, 1, 2, 3) and x1.

3. When setting value of the compare register is larger than timer setting value, compare match signal is not generated. Accordingly, the output waveform is fixed to “L” or “H” level.
   However, when setting value of another compare register is smaller than timer setting value, this compare match signal is generated. Accordingly, if the corresponding compare latch y (y=00, 01, 10, 11, 20, 21, 30, 31) interrupt source bit is set to “1” (valid), compare match interrupt request occurs.

4. When the compare x trigger enable bit is cleared to “0” (disabled), the match trigger to the waveform output circuit is disabled. Accordingly, the output waveform can be fixed to “L” or “H” level.
   However, in this case, the compare match signal is generated. Accordingly, if the corresponding compare latch y (y=00, 01, 10, 11, 20, 21, 30, 31) interrupt source bit is set to “1” (valid),compare match interrupt request occurs.

## Notes on Input Capture

1. If the capture trigger is input while the capture register (low-order and high-order) is in read, captured value is changed between high-order reading and low-order reading. Accordingly, some countermeasure by software is recommended, for example comparing the values that twice of read.

2. Timer A cannot be used for the capture source timer in the following state;
   - $X_{IN}$ oscillation selected by clock division ratio selection bits (bits 7 and 6 of CPU mode register (address $3B_{16}$))
   - Timer A count source: On-chip oscillator output.

   Timer B cannot be used for the capture source timer in the following state;
   - $X_{IN}$ oscillation selected by clock division ratio selection bits
   - Timer B count source: Timer A underflow
   - Timer A count source: On-chip oscillator output.

3. As shown below, when the capture input is performed to both capture latch 00 and 01 at the same time, the value of capture 0 status bit (bit 4 of capture/compare status register (address $22_{16}$)) is undefined (same as capture 1).
- When “1” is written to capture latch 00 software trigger bit (bit 0 of capture software trigger register (address $13_{16}$)) and capture latch 01 software trigger bit (bit 1 of capture software trigger register) at the same time
- When external trigger of capture latch 00 and software trigger of capture latch 01 occur at the same time
- When external trigger of capture latch 01 and software trigger of capture latch 00 occur at the same time

4. When the capture interrupt is used as the interrupt for return from stop mode, set the capture 0 noise filter clock selection bits (bits 5 and 4 of capture mode register (address $20_{16}$)) to “00 (Filter stop)” (same as capture 1).

## Notes on Serial I/Oi (i=1, 2)

1. Clock synchronous serial I/O

(1) When the transmit operation is stopped, clear the serial I/Oi enable bit and the transmit enable bit to “0” (serial I/Oi and transmit disabled).

<Reason>

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

(2) When the receive operation is stopped, clear the receive enable bit to “0” (receive disabled), or clear the serial I/Oi enable bit to “0” (serial I/Oi disabled).

(3) When the transmit/receive operation is stopped, clear both the transmit enable bit and receive enable bit to “0” (transmit and receive disabled) simultaneously. (any one of data transmission and reception cannot be stopped.)

<Reason>

In the clock synchronous serial I/O mode, the same clock is used for transmission and reception.

If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized. In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to “0” (transmit disabled). Also, the transmission circuit cannot be initialized even if the serial I/Oi enable bit is cleared to “0” (serial I/Oi disabled) (same as (1)).

(4) When signals are output from the $\overline{\text{SRDYi}}$ pin on the reception side by using an external clock, set all of the receive enable bit, the $\overline{\text{SRDYi}}$ output enable bit, and the transmit enable bit to “1”.

(5) When the $\overline{\text{SRDYi}}$ signal input is used, set the using pin to the input mode before data is written to the transmit/receive buffer register.

2. UART

When the transmit operation is stopped, clear the transmit enable bit to “0” (transmit disabled).

<Reason>

Same as (1) shown on the above “1. Clock synchronous serial I/O“.

When the receive operation is stopped, clear the receive enable bit to “0” (receive disabled).

When the transmit/receive operation is stopped, clear the transmit enable bit to “0” (transmit disabled) and receive enable bit to “0” (receive disabled).

3. Notes common to clock synchronous serial I/O and UART

(1) Set the serial I/Oi (i=1, 2) control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to “0.”

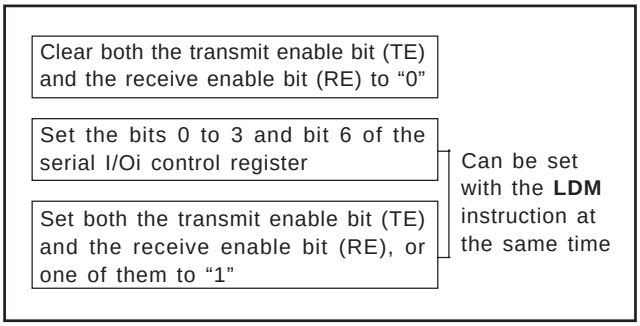



**Fig. 6 Sequence of setting serial I/Oi control register again**

(2) The transmit shift 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) When data transmission is executed at the state that an external clock input is selected as the synchronous clock, set “1” to the transmit enable bit while the SCLKi is “H” state. Also, write to the transmit buffer register while the SCLKi is “H” state.

(4) When the transmit interrupt is used, set as the following sequence.

① Serial I/Oi transmit interrupt enable bit is set to “0” (disabled).
② Serial I/Oi transmit enable bit is set to “1”.
③ Serial I/Oi transmit interrupt request bit is set to “0” after 1 or more instructions have been executed.
④ Serial I/Oi transmit interrupt enable bit is set to “1” (enabled).

<Reason>

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

Accordingly, even if the timing when any of the above flags is set to “1” is selected for the transmit interrupt source, interrupt request occurs and the transmit interrupt request bit is set.

(5) Write to the baud rate generator (BRGi) while the transmit/receive operation is stopped.

## Notes on Serial I/O1

1. I/O pin function when serial I/O1 is enabled.

The pin functions of P12/SCLK1 and P13/SRDY1 are switched to as follows according to the setting values of a serial I/O1 mode selection bit (bit 6 of serial I/O1 control register (address $1A_{16}$)) and a serial I/O1 synchronous clock selection bit (bit 1 of serial I/O1 control register).

(1) Serial I/O1 mode selection bit → "1" :

Clock synchronous type serial I/O is selected.

• Setup of a serial I/O1 synchronous clock selection bit

"0" : P12 pin turns into an output pin of a synchronous clock.

"1" : P12 pin turns into an input pin of a synchronous clock.

• Setup of a SRDY1 output enable bit (SRDY)

"0" : P13 pin can be used as a normal I/O pin.

"1" : P13 pin turns into a SRDY1 output pin.

(2) Serial I/O1 mode selection bit → "0" :

Clock asynchronous (UART) type serial I/O is selected.

• Setup of a serial I/O1 synchronous clock selection bit

"0": P12 pin can be used as a normal I/O pin.

"1": P12 pin turns into an input pin of an external clock.

• When clock asynchronous (UART) type serial I/O is selected, it functions P13 pin. It can be used as a normal I/O pin.

## Note on Bus Collision Detection

When serial I/O1 is operating at half-duplex communication, set bus collision detection interrupt to be disabled.

## Notes on Serial I/O2

1. I/O pin function when serial I/O2 is enabled

The pin functions of P06/SCLK2 and P07/SRDY2 are switched to as follows according to the setting values of a serial I/O2 mode selection bit (bit 6 of serial I/O2 control register (address $30_{16}$)) and a serial I/O2 synchronous clock selection bit (bit 2 of serial I/O2 control register).

(1) Serial I/O2 mode selection bit → "1" :

Clock synchronous type serial I/O is selected.

• Setup of a serial I/O2 synchronous clock selection bit

"0" : P06 pin turns into an output pin of a synchronous clock.

"1" : P06 pin turns into an input pin of a synchronous clock.

• Setup of a SRDY2 output enable bit (SRDY)

"0" : P07 pin can be used as a normal I/O pin.

"1" : P07 pin turns into a SRDY2 output pin.

(2) Serial I/O2 mode selection bit → "0" :

Clock asynchronous (UART) type serial I/O is selected.

• Setup of a serial I/O2 synchronous clock selection bit

"0": P06 pin can be used as a normal I/O pin.

"1": P06 pin turns into an input pin of an external clock.

• When clock asynchronous (UART) type serial I/O is selected, it functions P07 pin. It can be used as a normal I/O pin.

## Notes on A/D conversion

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. This may cause the A/D conversion precision to be worse. Accordingly, set f(XIN) in order that the A/D conversion clock is 250 kHz or over during A/D conversion.

3. A/D conversion clock selection

Select f(XIN)/2 as an A/D conversion clock by setting the A/D conversion clock selection bit (bit 3 of A/D control register (address $34_{16}$)) when RC oscillation is used.

The f(XIN) can be also used as an A/D conversion clock only when ceramic oscillation or on-chip oscillator is used.

4. Analog input pin selection

$P2_6$/$AN_6$ and $P2_7$/$AN_7$ can be used only for PRSP0036GA-A package version.

5. Read A/D conversion register

• 8-bit read

Read only the A/D conversion low-order register (address $35_{16}$).

•10-bit read

Read the A/D conversion high-ordrer register (address $36_{16}$) first, and then, read the A/D conversion low-order register (address $35_{16}$).

In this case, the high-order 6 bits of address $36_{16}$ returns “0” when read.

6. A/D conversion accuracy

As for AD translation accuracy, on the following operating conditions, accuracy may become low.

(1) Since the analog circuit inside a microcomputer becomes sensitive to noise when VREF voltage is set up lower than Vcc voltage, accuracy may become low rather than the case where VREF voltage and Vcc voltage are set up to the same value..

(2) When VREF voltage is lower than [ 3.0 V ], the accuracy at the low temperature may become extremely low compared with that at room temperature. When the system would be used at low temperature, the use at VREF=3.0 V or more is recommended.

## Notes on Watchdog Timer

1. The watchdog timer is operating during the wait mode. Write data to the watchdog timer control register to prevent timer underflow.

2. The watchdog timer stops during the stop mode. However, the watchdog timer is running during the oscillation stabilizing time after the **STP** instruction is released. In order to avoid the underflow of the watchdog timer, the watchdog timer control register must be written just before executing the **STP** instruction.

3. The **STP** instruction function selection bit (bit 6 of watchdog timer control register (address $0039_{16}$)) can be rewritten only once after releasing reset. After rewriting it is disable to write any data to this bit.

4. A count source of watchdog timer is affected by the clock division selection bit of the CPU mode register.
   The f(XIN) clock is supplied to the watchdog timer when selecting f(XIN) as the CPU clock.
   The on-chip oscillator output is supplied to the watchdog timer when selecting the on-chip oscillator output as the CPU clock.

## Notes on 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.

## Notes on Clock Generating Circuit

1. Switch of ceramic and RC oscillations
After releasing reset, the oscillation mode selection bit (bit 5 of CPU mode register (address $3B_{16}$)) is “0” (ceramic oscillation selected). When the RC oscillation is used, after releasing reset, set this bit to “1”.

2. Double-speed mode
The double-speed mode can be used only when a ceramic oscillation is selected. Do not use it when an RC oscillation is selected.

3. CPU mode register
Oscillation mode selection bit (bit 5), processor mode bits (bits 1 and 0) of CPU mode register (address $3B_{16}$) are used to select oscillation mode and to control operation modes of the microcomputer. In order to prevent the dead-lock by erroneously writing (ex. program run-away), these bits can be rewritten only once after releasing reset. After rewriting, it is disabled to write any data to the bit. (The emulator MCU “M37542RSS” is excluded.)
Also, when the read-modify-write instructions (SEB, CLB, etc.) are executed to bits 2 to 4, 6 and 7, bits 5, 1 and 0 are locked.

4. Clock division ratio, XIN oscillation control, on-chip oscillator control
The state transition shown in Fig. 84 can be performed by setting the clock division ratio selection bits (bits 7 and 6), XIN oscillation control bit (bit 4), on-chip oscillator oscillation control bit (bit 3) of CPU mode register. Be careful of notes on use in Fig. 84.

5. On-chip oscillator operation
When the MCU operates by the on-chip oscillator for the main clock, connect XIN pin to VCC through a 1 kΩ to 10 kΩ resistor and leave XOUT pin open.
The clock frequency of the on-chip oscillator depends on the supply voltage and the operation temperature range.
Be careful that this margin of frequencies when designing application products.

6. Ceramic resonator
When the ceramic resonator is used for the main clock, connect the ceramic resonator and the external circuit to pins XIN and XOUT at the shortest distance. Externally connect a damping resistor Rd depending on the oscillation frequency. A feedback resistor is built-in.
Use the resonator manufacturer’s recommended value because constants such as capacitance depend on the resonator.

7. RC oscillation
When the RC oscillation is used for the main clock, connect the XIN pin and XOUT pin to the external circuit of resistor R and the capacitor C at the shortest distance.
The frequency is affected by a capacitor, a resistor and a microcomputer.
So, set the constants within the range of the frequency limits.

8. External clock
When the external signal clock is used for the main clock, connect the XIN pin to the clock source and leave XOUT pin open.
Select “0” (ceramic oscillation) to oscillation mode selection bit.

9. Count source (Timer 1, Timer A, Timer B, Timer X, Serial I/O, Serial I/O2, A/D converter, Watchdog timer)
A count source of watchdog timer is affected by the clock division selection bit of the CPU mode register.
The f(XIN) clock is supplied to the watchdog timer when selecting f(XIN) as the CPU clock.
The on-chip oscillator output is supplied to the watchdog timer when selecting the on-chip oscillator output as the CPU clock.

## Notes on Oscillation Control

1. Oscillation stop detection circuit
(1) When the stop mode is used, set the oscillation stop detection function to “invalid”.

(2) When the ceramic or RC oscillation is stopped by the XIN oscillation control bit (bit 4 of CPU mode register (address $3B_{16}$)), set the oscillation stop detection function to “invalid”.

2. Stop mode
(1) When the stop mode is used, set the oscillation stop detection function to “invalid”.

(2) When the stop mode is used, set “0” (**STP** instruction enabled) to the **STP** instruction function selection bit of the watchdog timer control register (bit 6 of watchdog timer control register (address $39_{16}$)).

(3) The oscillation stabilizing time after release of **STP** instruction can be selected from “set automatically ”/“not set automatically” by the oscillation stabilizing time set bit after release of the **STP** instruction (bit 0 of MISRG (address $38_{16}$)). When “0” is set to this bit, “$01_{16}$” is set to timer 1 and “$FF_{16}$” is set to prescaler 1 automatically at the execution of the STP instruction. When “1” is set to this bit, set the wait time to timer 1 and prescaler 1 according to the oscillation stabilizing time of the oscillation. Also, when timer 1 is used, set values again to timer 1 and prescaler 1 after system is returned from the stop mode.

(4) Do not execute the **STP** instruction during the A/D conversion.

## Notes on On-chip Oscillation Division Ratio

• When the clock division ratio is switched from f(XIN) to on-chip oscillator by the clock division ratio selection bits (bits 7 and 6 of CPU mode register (address $3B_{16}$)), the on-chip oscillator division ratio (bits 1 and 0 of on-chip oscillation division ratio selection register (address $37_{16}$)) is "$10_2$" (on-chip oscillator middle-speed mode (ROSC/8)).

## Notes on Oscillation Stop Detection Circuit

1. After the reset by the oscillation stop detection, the value of following bits are retained, not initialized.

• Ceramic or RC oscillation stop detection function active bit
Bit 1 of MISRG (address $3B_{16}$)
• Oscillation stop detection status bit
Bit 3 of MISRG

2. Oscillation stop detection status bit is initialized ("0") by the following operation.

• External reset
• Write "0" data to the ceramic or RC oscillation stop detection function active bit.

3. The oscillation stop detection circuit is not included in the emulator MCU "M37542RSS".

## Notes on CPU Rewrite Mode

Take the notes described below when rewriting the flash memory in CPU rewrite mode.

1. Operation speed

During CPU rewrite mode, set the system clock $\phi$ to 4.0 MHz or less using the clock division ratio selection bits (bits 6 and 7 of CPU mode register).

2. Instructions inhibited against use

The instructions which refer to the internal data of the flash memory cannot be used during CPU rewrite mode.

3. Interrupts inhibited against use

The interrupts cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory.

4. Watchdog timer

If the watchdog timer has been already activated, internal reset due to an underflow will not occur because the watchdog timer is surely initialized during program or erase.

5. Reset

Reset is always valid. The MCU is activated using the boot mode at release of reset in the condition of CNVss = "H", so that the program will begin at the address which is stored in addresses $FFFC_{16}$ and $FFFD_{16}$ of the boot ROM area.

## Electric Characteristic Differences Between Mask ROM, Flash Memory MCUs

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

When manufacturing an application system with the flash memory 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.

# NOTES ON HARDWARE

## 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 capacitor of 0.01 μF to 0.1 μF is recommended.

| REVISION HISTORY | 7542 Group Datasheet |
|---|---|

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.00 | Nov 27, 2002 | – | First edition issued |
| 2.00 | Apr 21, 2003 | 1 | FEATURES; Memory size revised. |
| | | 8 | Memory size; Flash memory size revised. |
| | | | Fig.8; ROM size revised. |
| | | 9 | Table 2; ROM size revised. |
| | | 10 | Central Processing Unit (CPU); Description revised. |
| | | 28 | Fig.26; Port $P0_3$ direction register revised. |
| | | 36 | Fig.42; Modulation output revised. |
| | | 37 | Fig.43; Modulation output revised. |
| | | 53 | Reset Circuit; Description revised. |
| | | 55 | (3) RC oscillation; Description revised. |
| | | 56 | (1) Oscillation control |
| | | | • Stop mode |
| | | | Description about FLASH added. |
| | | 57 | Fig.77; revised. |
| | | 65 to 72 | FLASH MEMORY MODE added. |
| | | 73 to 82 | ELECTRICAL CHARACTERISTICS added. |
| 2.01 | Dec 03, 2003 | 1 | FEATURES: Interrupt, Power source voltage, Power dissipation revised. |
| | | 8 | Fig. 8: Development schedule revised. |
| | | 9 | Table 2: ROM size for Flash memory version revised. |
| | | 13 | Fig. 12: Note added. |
| | | 15 | Fig. 14: Flash memory control register 2 added. |
| | | 28 | Fig. 26: “CPU mode register” added, description for timer 1 interrupt request revised. |
| | | 30 | Fig. 29: “CPU mode register” added. |
| | | 33 | Fig. 37 and Fig. 38: Pin name added. |
| | | 34 | Fig. 39: Pin name added. |
| | | 40 | Fig. 46 and Fig. 47: Pin name added. |
| | | 51 | A-D Converter revised. |
| | | 54 | Fig. 70 Flash memory control register 2 added. |
| | | 59 | Fig. 79 (5), (6) revised. |
| | | 64 | ■ A-D Converter revised. |
| | | 65 | DATA REQUIRED FOR MASK ORDERS revised. |
| | | 68 | Description of flash memory control register 0 (bit 2), Fig. 83 revised. |
| | | 69 | Description of flash memory control register 2, Fig. 85 and Table 8 added. |
| | | 70 | Fig. 86 revised. |
| | | 71 | Table 9 revised. |
| | | 85 to103 | ELECTRICAL CHARACTERISTICS; |
| | | | General purpose revised. |
| | | | Extended operating temperature version added. |
| 2.02 | Jan 06, 2004 | 1 | FEATURES: The minimum instruction execution time revised. |
| | | | Note 2 eliminated. |
| | | 10 | Stack pointer (S): Reference number of Figure in description revised. |
| | | 79 | Table 12: $P0_7$ (BUSY output) added. |
| | | 82 | Fig. 95: “BUSY” added to $P0_7$. |
| | | 83 | Fig. 96, Fig. 97: “BUSY” added to $P0_7$. |
| | | 84 | Fig. 98, Fig. 99: $CNV_{SS}$ revised. |
| | | 91 | Table 19, Table 20 Timing requirements (General purpose) |
| | | | Vcc for FLASH ROM version and Mask ROM version revised. |
| | | 93 | Table 22, Table 23 Switching characteristics (General purpose) |
| | | | Vcc for FLASH ROM version and Mask ROM version revised. |

| REVISION HISTORY | 7542 Group Datasheet |
|---|---|

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 2.03 | Feb 10, 2004 | | Information about 36PJW-A package version added. |
| | | 3 | - Fig.4 Pin configuration added. |
| | | 8 | - Fig.9 Functional block diagram added. |
| | | 9 | - Table 1: Notes 2, 3 revised. |
| | | 10 | - 36PJW-A package added. |
| | | 11 | - Table 2 List of supported products revised. |
| | | 18 | - I/O Ports description and Fig. 19: Note revised. |
| | | 19 | - Table 5: Notes 2, 3 revised. |
| | | 23 | - INTEDGE revised. |
| | | 26 | - Fig.24: Note revised. |
| | | 81 | Table 12: $P0_0$–$P0_3$, $P0_7$ → $P0_0$–$P0_3$ |
| | | 86 | Fig.100, Fig.101: td(CNVss-port) → th(CNVss-port) |
| | | 91 | Table 17: $I_{CC}$ data for FLASH ROM added. |
| | | 92 | Table 18: Absolute accuracy for FLASH ROM added. |
| | | 101 | Table 29: $I_{CC}$ data for FLASH ROM added. |
| | | 102 | Table 30: Absolute accuracy for FLASH ROM added. |
| 2.04 | Apr 14, 2004 | 10 | Package: Description of 36PJW-A revised. |
| | | 11 | Table 2: M37542M2-XXXHP added. |
| | | 60 | Fig. 79, Fig. 80 a bit name revised. |
| | | 65 to 68 | Countermeasure against noise added. |
| | | | (NOTES ON PERIPHERAL FUNCTIONS are included in APPENDIX at the end of this data sheet.) |
| | | 88 | Part name revised. |
| | | 108 | 36PJW-A package added. |
| | | 109 to 118 | APPENDIX added. |
| 2.05 | Jun 08, 2004 | 1 | FEATURES • Programmable I/O ports, • A/D converter: Description added. |
| | | 3 | Fig.4: Pin 1 to Pin 3 revised. M37542F8HP: Note added. |
| | | 11 | Table 2: M37542F8HP: Note added. |
| | | 53 | Notes on A/D conversion added. |
| | | 70 | Table 7: Number of program/erase times revised. |
| | | 88 | Fig. 110: Figure title and table in figure revised, and Note added. |
| | | 89 | M37542F8HP: Note added. |
| | | 117 | Notes on A/D accuracy added. |
| | | 119 | Notes on Oscillation Stop Detection Circuit |
| | | | 1: • Each bit of Port register Pi eliminated. |
| | | | Note on Power Source Voltage added. |
| | | All pages | Words standardized: On-chip oscillator, A/D converter |
| 2.06 | Aug 24, 2004 | 73 | Fig. 97: Bits 0 to 3 revised. |
| | | 79 | (1) ROM Code Protect Function: Some description added. |
| | | 82 | ● Standard serial I/O Mode: Some description revised. |
| | | 83 to 92 | Description of standard serial I/O mode 1 and standard serial I/O mode 2 separated. |
| | | 86 | Fig. 107 Handling example of control pins in standard serial I/O mode 1 added. |
| | | 91 | Fig. 112 Handling example of control pins in standard serial I/O mode 2 added. |

| REVISION HISTORY | 7542 Group Datasheet |
|---|---|

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 3.00 | Jun 01, 2005 | 1 | ROM size of Flash memory version revised. |
| | | 2 | Fig.1 M37542F8GP → M37542FxGP |
| | | | Fig.2 M37542F8FP → M37542FxFP |
| | | 3 | Fig.3 M37542F8SP → M37542FxSP |
| | | 5 | Table 1 Performance overview added. |
| | | 10 | Table 2 Function of Vcc, Vss revised. |
| | | 11 | Flash memory size revised, and Fig.10 M37542F4 added. |
| | | 12 | Table 3 M37542F4GP,M37542F4FP,M37542F4SP added. |
| | | 21 | Fig. 20 (5) Port $P0_5$ revised. |
| | | 24 | Table 7 Termination of unused pins added. |
| | | 46 | Description of Serial I/O revised. |
| | | 53 | [UART2 control register (UART2CON)] revised. |
| | | 54 | Fig. 64 UART2 contorl register revised. |
| | | 60 | Description of Clock Generating Circuit revised. Fig. 74 revised. |
| | | 63 | Fig. 79 revised. |
| | | 72 | Table 9 Temperature at program/erase added. |
| | | 73 | Fig.94 16 Kbyte ROM Product added. |
| | | 77 | Table 11 List of software commands (CPU rewrite mode) revised. |
| | | 86 | Fig.104 M37542F8GP → M37542FxGP |
| | | 87 | Fig.105 M37542F8SP → M37542FxSP |
| | | | Fig.106 M37542F8FP → M37542FxFP |
| | | 91 | Fig.109 M37542F8GP → M37542FxGP |
| | | 92 | Fig.110 M37542F8SP → M37542FxSP |
| | | | Fig.111 M37542F8FP → M37542FxFP |
| | | 95 | M37542F4GP,M37542F4FP,M37542F4SP added. |
| | | | Table 15 Conditions: Description added. |
| | | 98 | Table 18 Note 1 added. |
| | | 100 | Table 20, Fig. 104 added. |
| | | 105 | Table 28 Conditions: Description added. |
| | | 108 | Table 31 Note 1 added. |
| | | 110 | Table 34, Fig. 106 added. |
| | | 114-122 | Extended operating temperature 125 °C version added. |
| | | 125 | (2) How to reference the processor status register revised. |
| | | | Fig. 2 revised. |
| | | - | Package revised. |
| 3.01 | Nov 02, 2005 | - | Bit name revised: STP instruction disable bit → STP instruction function selection bit |
| | | 57 | - Description for “Operation of STP instruction function selection bit” revised.<br>- Notes on Watchdog Timer added.<br>- Fig.68: Blodk diagram of watchdog timer revised.<br>- Fig.69: Bit 6 and Bit 7 of WDTCON revised.<br>Bit 6: Bit name and its description revised. (Bit function is not changed.)<br>→ STP instruction function selection bit<br>0 : System enters into the stop mode at the STP instruction execution<br>1 : Internal reset occurs at the STP instruction execution |
| | | 61 | -Notes on Clock Generating Circuit : Note on Count source added. |
| | | 132 | Notes on Watchdog Timer : Note on Count source added. |
| | | 133 | Notes on Clock Generating Circuit : Note on Count source added. |

| REVISION HISTORY | 7542 Group Datasheet |
|---|---|

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 3.02 | Oct 31, 2006 | 12 | Table 3 : ROM size revised and note 2 added. |
| | | 17 | ROM : Description added. |
| | | | Fig. 15 : Note 2 added. |
| | | 24 | Table 7 : XIN and XOUT added. |
| | | 57, 132 | Notes on Watchdog Timer : Note 3 revised. |
| | | 71 | 5. Setup for I/O ports : Note eliminated. |
| | | 73 | Fig 94 : Block diagram revised and note 3 added. |
| | | 125 | 4. BRK instruction eliminated. |
| | | 132 | 1. Analog input pin : Description revised. |
| 3.03 | Jul 11, 2008 | 1 | FEATURES: Description revised. |
| | | | DESCRIPTION, FEATURES: “serial I/O” → “serial interface” |
| | | 2 | APPLICATION: “car” deleted. |
| | | | Fig. 1, Fig. 2 revised. |
| | | 5 | Table 1: Parameter revised, Note 1 deleted. |
| | | 10 | Table 2: Note 1 deleted. |
| | | 11 | Fig. 10 revised. |
| | | 12 | Table 3 revised. |
| | | 21 | Fig. 20 revised. |
| | | 22 | Fig. 21 revised. |
| | | 25 to 29 | Interrupts revised. |
| | | 47 | “Serial I/O” → “Serial interface” |
| | | 66 | • Oscillation stop detection circuit: Description revised |
| | | 77 | Fig. 101: Note 2, Note 3 revised. |
| | | 87 | Fig. 107 revised. |
| | | 88 | Fig. 108, Fig. 109 revised. |
| | | 92 | Fig. 112 revised. |
| | | 93 | Fig. 113 revised. |
| | | 96 to 105 | ELECTRICAL CHARACTERISTICS: |
| | | | 1.7542Group (General purpose); Description revised, “(General purpose)” deleted |
| | | | 2.7542Group (Extended operating temperature version), |
| | | | 3.7542Group (Extended operating temperature 125 °C version) deleted. |
| | | 111 | Notes on Timer A, B: “(bit 0 (bit 2 of timer .... (address $1D_{16}$))” deleted. |

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan



## RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

**Renesas Technology America, Inc.**
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

**Renesas Technology Europe Limited**
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

**Renesas Technology (Shanghai) Co., Ltd.**
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

**Renesas Technology Hong Kong Ltd.**
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

**Renesas Technology Taiwan Co., Ltd.**
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

**Renesas Technology Singapore Pte. Ltd.**
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

**Renesas Technology Korea Co., Ltd.**
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

**Renesas Technology Malaysia Sdn. Bhd**
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510



Colophon .7.2

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Renesas Electronics:

M37542F8GP#U0

# OCEAN CHIPS
## Океан Электроники
## Поставка электронных компонентов

Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения «**JONHON**», а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов «**FORSTAR**».

AVIC JONHON

**«JONHON»** (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

**«FORSTAR»** (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели, кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).

FORSTAR®

Телефон: 8 (812) 309-75-97 (многоканальный)
Факс: 8 (812) 320-03-32
Электронная почта: ocean@oceanchips.ru
Web: http://oceanchips.ru/
Адрес: 198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А