To all our customers

# Regarding the change of names mentioned in the document, such as Hitachi Electric and Hitachi XX, to Renesas Technology Corp.

The semiconductor operations of Mitsubishi Electric and Hitachi were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Renesas Technology Home Page: http://www.renesas.com

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

## Cautions

Keep safety first in your circuit designs!

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

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.

# HD404459 Series

Rev. 6.0
Sept. 1998

## Description

The HD404459 Series is a member of the 4-bit HMCS400-series microcomputers with large-capacity memory and architecture providing high program productivity. Each microcomputer has a 32-kHz oscillator for clock, low-voltage (1.8 V) operating mode, and four low-power dissipation modes.

The HD404459 Series includes three chips: the HD404458 with an 8-kword ROM; the HD404459 with a 16-kword ROM; and the HD4074459 with a 16-kword PROM (ZTAT™ version).

The HD4074459 is a PROM version (ZTAT™ microcomputer). A program can be written to the PROM by a PROM writer, thus dramatically shortening system development periods and turnaround time (ZTAT™ versions are 27256-compatible).

ZTAT™: Zero Turn Around Time  ZTAT is a trademark of Hitachi, Ltd.

## Features

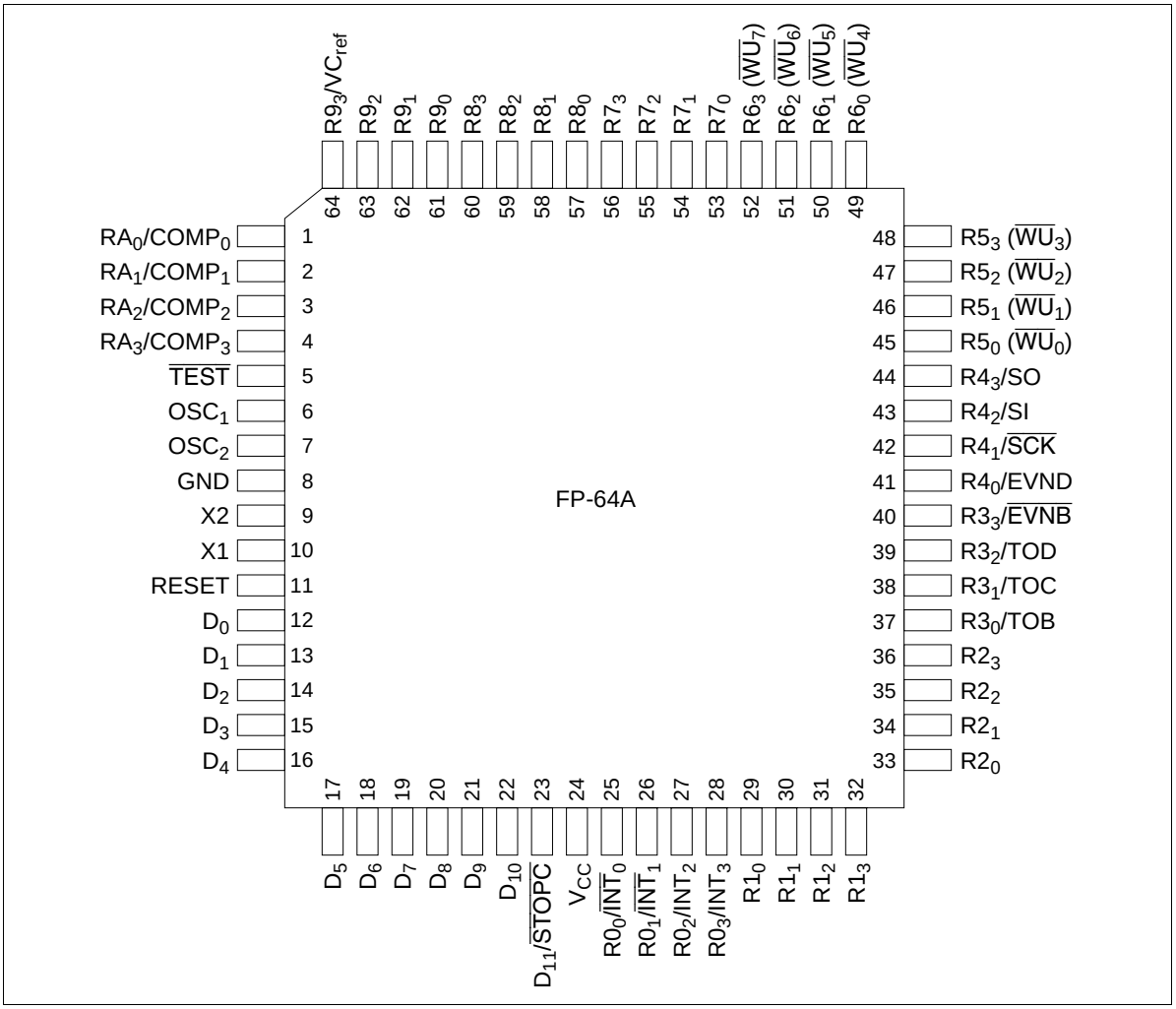
- 8,192-word × 10-bit ROM (HD404458)
  16,384-word × 10-bit ROM (HD404459 and HD4074459)
- 512-digit × 4-bit RAM (HD404458)
  768-digit × 4-bit RAM (HD404459 and HD4074459)
- 56 I/O pins, including seven input pins
- Four timer/counters
- 1-channel × 8-bit input capture circuit
- Three timer outputs (including two PWM outputs)
- Two event counter inputs (including one double-edge function)
- 8-bit clock-synchronous serial interface
- Eight wakeup inputs
- Four-channel voltage comparator (external or internal reference power supply can be selected)
- Built-in oscillators
  — Main clock: 4-MHz ceramic or crystal oscillator (an external clock is also possible)
  — Subclock: 32.768-kHz crystal

- Ten interrupt sources
  - — Five by external sources, including two double-edge function
  - — Five by internal sources
- Subroutine stack up to 16 levels, including interrupts
- Four low-power dissipation modes (transition time shortened)
  - — Stop mode
  - — Standby mode
  - — Watch mode
  - — Subactive mode (optional)
- One external input for transition from stop mode to active mode
- Instruction cycle time
  - — For HD404458/HD404459:
    1, 2, 4, 8 µs ($f_{OSC}$ = 4 MHz; 1/4, 1/8, 1/16, 1/32 division ratio)
  - — For HD4074459:
    1, 2, 4, 8 µs ($f_{OSC}$ = 4 MHz; 1/4, 1/8, 1/16, 1/32 division ratio; power voltage of 2.7 V or higher)
    2, 4, 8, 16 µs ($f_{OSC}$ = 2 MHz; 1/4, 1/8, 1/16, 1/32 division ratio; power voltage of 2.2 V or higher)
- Two general operating conditions
  - — MCU or PROM mode for HD4074459
  - — MCU mode only for HD404458/HD404459

## Ordering Information

| Type | Product Name | Model Name | ROM (Words) | RAM (Digits) | Package |
|---|---|---|---|---|---|
| Mask ROM | HD404458 | HD404458H | 8,192 | 512 | 64-pin plastic QFP (FP-64A) |
| | HD404459 | HD404459H | 16,384 | 768 | 64-pin plastic QFP (FP-64A) |
| ZTAT™ | HD4074459 | HD4074459H | 16,384 | 768 | 64-pin plastic QFP (FP-64A) |

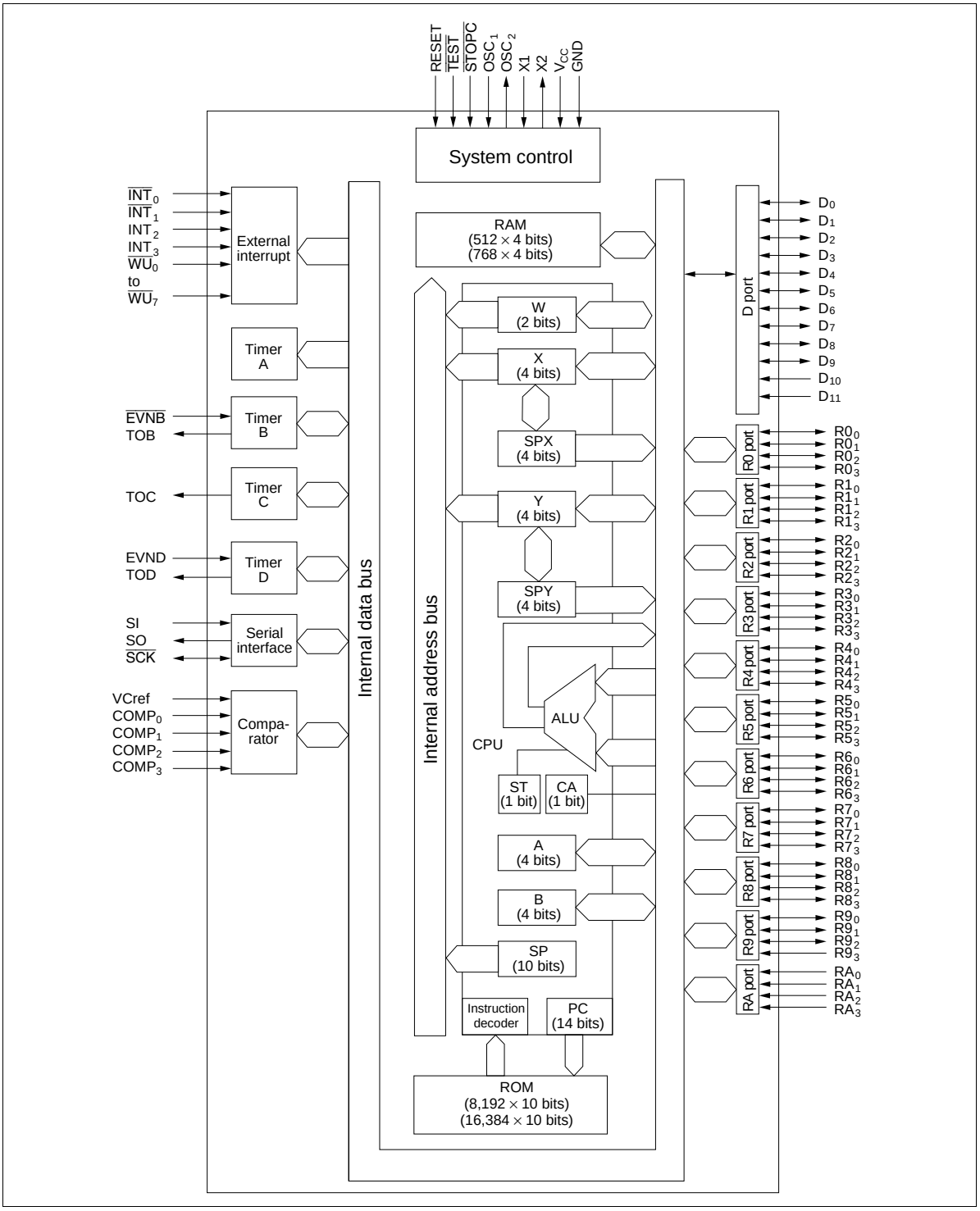
## Pin Arrangement



## Pin Description

| Item | Symbol | Pin Number FP-64A | I/O | Function |
|---|---|---|---|---|
| Power supply | $V_{CC}$ | 24 | | Power voltage |
| | GND | 8 | | Ground |
| Test | $\overline{TEST}$ | 5 | I | Used for factory testing only: Connect this pin to $V_{CC}$ |
| Reset | RESET | 11 | I | Resets the MCU |
| Oscillator | $OSC_1$ | 6 | I | Input/output pins for the internal oscillator circuit: Connect them to a ceramic, crystal, or connect only $OSC_1$ to an external oscillator circuit |
| | $OSC_2$ | 7 | O | |
| | X1 | 10 | I | Used for a 32.768-kHz crystal for clock purposes. If not to be used, fix the X1 pin to $V_{CC}$ and leave the X2 pin open. |
| | X2 | 9 | O | |
| Ports | $D_0$–$D_9$ | 12–21 | I/O | Input/output pins addressable by individual bits |
| | $D_{10}$, $D_{11}$ | 22, 23 | I | Input pins addressable by individual bits |
| | $R0_0$–$R9_3$ | 25–64 | I/O | Input/output pins addressable in 4-bit units. The $R9_3$ port is an input-only pin. |
| | $RA_0$–$RA_3$ | 1–4 | I | Input pins addressable in 4-bit units |
| Interrupts | $\overline{INT}_0$, $\overline{INT}_1$, $INT_2$, $INT_3$, $\overline{WU}_0$–$\overline{WU}_7$ | 25–28, 45–52 | I | Input pins for external interrupts |
| Stop clear | $\overline{STOPC}$ | 23 | I | Input pin for transition from stop mode to active mode |
| Serial interface | $\overline{SCK}$ | 42 | I/O | Serial clock input/output pin |
| | SI | 43 | I | Serial receive data input pin |
| | SO | 44 | O | Serial transmit data output pin |
| Timers | TOB, TOC, TOD | 37–39 | O | Timer output pins |
| | $\overline{EVNB}$, EVND | 40, 41 | I | Event count input pins |
| Voltage comparator | $COMP_0$–$COMP_3$ | 1–4 | I | Analog input pins for voltage comparator |
| | $VC_{ref}$ | 64 | I | Standard voltage pin for inputting the threshold voltage of analog input pins |

## Block Diagram



## Memory Map

### ROM Memory Map

See the ROM memory map of figure 1.

**Vector Address Area ($0000–$000F):** Reserved for JMPL instructions that branch to the start addresses of the reset and interrupt routines. After MCU reset or an interrupt, program execution continues from the vector address.

**Zero-Page Subroutine Area ($0000–$003F):** Reserved for subroutines. The program branches to a subroutine in this area in response to the CAL instruction.

**Pattern Area ($0000–$0FFF):** Contains ROM data that can be referenced with the P instruction.

**Program Area ($0000–$1FFF for HD404458, $0000–$3FFF for HD404459/HD4074459):** Used for program coding.

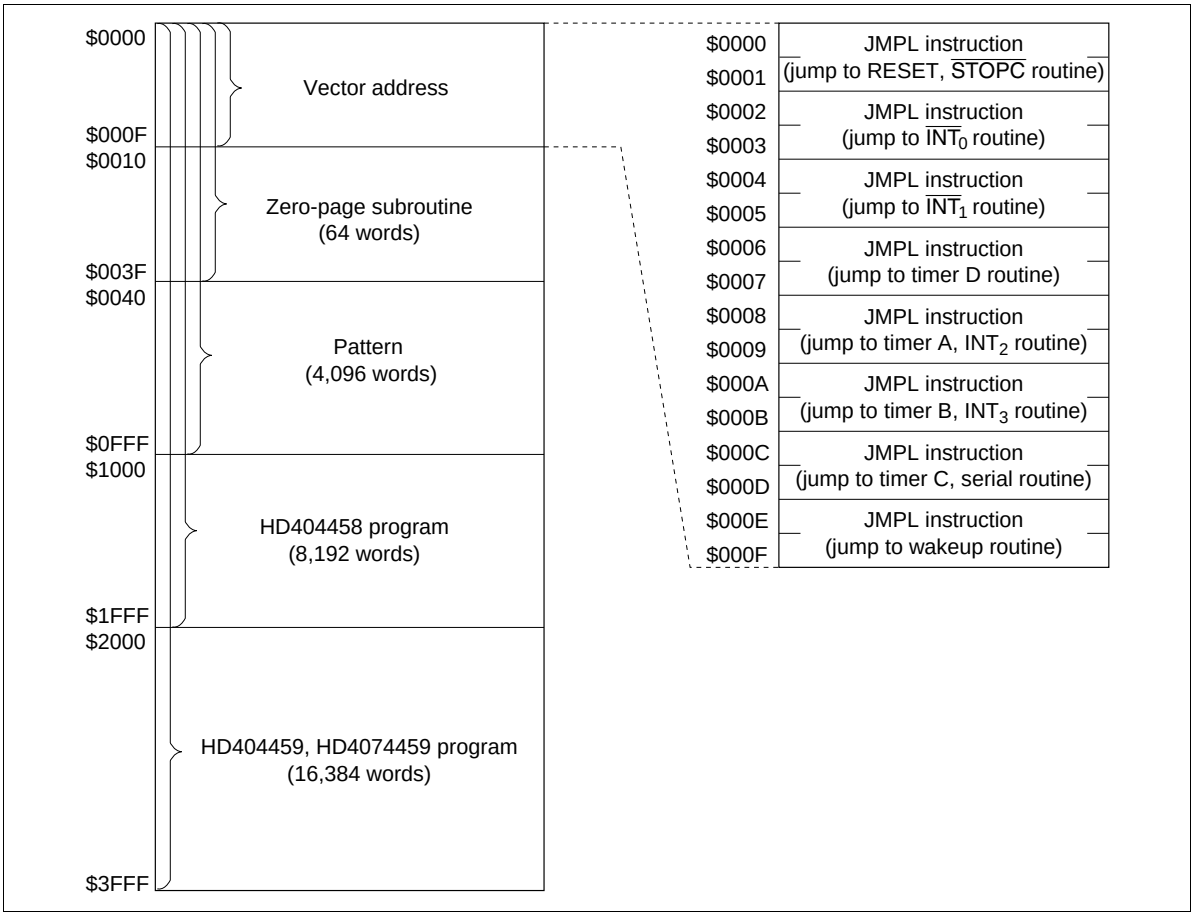



**Figure 1 ROM Memory Map**

### RAM Memory Map

The HD404458 MCU contains a 512-digit × 4-bit RAM area. The HD404459 and HD4074459 MCUs contain 768-digit × 4-bit RAM areas. Both of these RAM areas consist of a memory register area, a data area, and a stack area. In addition, an interrupt control bits area, special register area, and register flag area are mapped onto the same RAM memory space labeled as the RAM-mapped register area. See the RAM memory map of figure 2.

**RAM-Mapped Register Area ($000–$03F):**

- Interrupt control bits area ($000–$003)

  This area is used for interrupt control bits (figure 3). These bits can be accessed only by RAM bit manipulation instructions (SEM/SEMD, REM/REMD, and TM/TMD). However, note that not all the instructions can be used for each bit. For limitations on using the instructions, refer to figure 4.

- Special function register area ($004–$01F, $024–$03F)

  This area is used as mode registers and data registers for external interrupts, serial interface, timer/counters, and as data control registers for I/O ports. See figures 2 and 5. These registers can be classified into three types: write-only (W), read-only (R), and read/write (R/W). RAM bit manipulation instructions cannot be used for these registers.

- Register flag area ($020–$023)

  This area is used for the DTON, WDON, and other register flags and interrupt control bits (figure 3). These bits can be accessed only by RAM bit manipulation instructions (SEM/SEMD, REM/REMD, and TM/TMD). However, note that not all the instructions can be used for each bit. For limitations on using the instructions, refer to figure 4.

**Memory Register (MR) Area ($040–$04F):** Consisting of 16 addresses, this area (MR0–MR15) can be accessed by register-register instructions (LAMR and XMRA). See figure 6.

**Data Area ($050–$1FF for HD404458, $050–$2FF for HD404459/HD4074459)**

**Stack Area ($3C0–$3FF):** Used for saving the contents of the program counter (PC), status flag (ST), and carry flag (CA) at subroutine call (CAL or CALL instruction) and for interrupts. This area can be used as a 16-level nesting subroutine stack in which one level requires four digits. See figure 6 for the data to be saved and the save conditions.

The program counter is restored by either the RTN or RTNI instruction, but the status and carry flags can only be restored by the RTNI instruction. Any unused space in this area can be used for data storage.

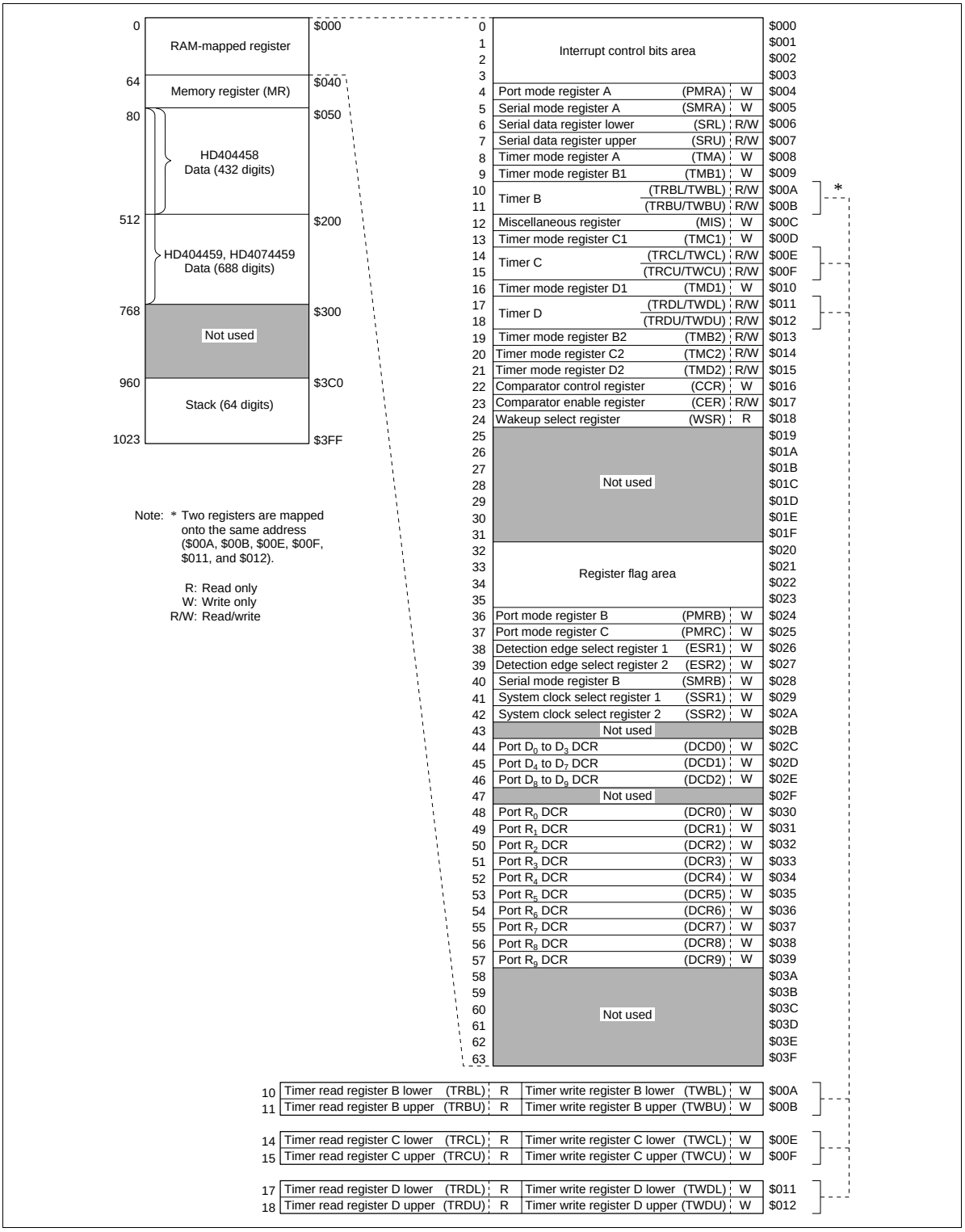



**Figure 2 RAM Memory Map**

**Interrupt control bits area**

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| 0 | IM0 (IM of $\overline{INT_0}$) | IF0 (IF of $\overline{INT_0}$) | RSP (Reset SP bit) | IE (Interrupt enable flag) | $000 |
| 1 | IMTD (IM of timer D) | IFTD (IF of timer D) | IM1 (IM of $\overline{INT_1}$) | IF1 (IF of $\overline{INT_1}$) | $001 |
| 2 | IMTB (IM of timer B) | IFTB (IF of timer B) | IMTA (IM of timer A) | IFTA (IF of timer A) | $002 |
| 3 | IMWU (IM of wakeup) | IFWU (IF of wakeup) | IMTC (IM of timer C) | IFTC (IF of timer C) | $003 |

**Register flag area**

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| 32 | DTON (Direct transfer on flag) | CMSF (Comparator start flag) | WDON (Watchdog on flag) | LSON (Low speed on flag) | $020 |
| 33 | RAME (RAM enable flag) | Not used | ICEF (Input capture error flag) | ICSF (Input capture status flag) | $021 |
| 34 | IM3 (IM of $INT_3$) | IF3 (IF of $INT_3$) | IM2 (IM of $INT_2$) | IF2 (IF of $INT_2$) | $022 |
| 35 | Not used | Not used | IMS (IM of serial) | IFS (IF of serial) | $023 |

IF: Interrupt request flag
IM: Interrupt mask
SP: Stack pointer

**Figure 3 Configuration of Interrupt Control Bits and Register Flag Areas**

| | SEM/SEMD | REM/REMD | TM/TMD |
|---|---|---|---|
| IE | Allowed | Allowed | Allowed |
| IM | | | |
| LSON | | | |
| IF | Not executed | Allowed | Allowed |
| ICSF | | | |
| ICEF | | | |
| RAME | | | |
| RSP | Not executed | Allowed | Inhibited |
| WDON | Allowed | Not executed | Inhibited |
| CMSF | Allowed | Inhibited | Allowed |
| DTON | Not executed in active mode | Allowed | Allowed |
| | Used in subactive mode | | |
| Not used | Not executed | Not executed | Inhibited |

Note: WDON is reset by MCU reset or by $\overline{STOPC}$ enable for stop mode cancellation.
The REM or REMD instuction must not be executed for CMSF during comparator operation. DTON is always reset in active mode.
If the TM or TMD instruction is executed for the inhibited bits or non-existing bits, the value in ST becomes undefined.

**Figure 4 Usage Limitations of RAM Bit Manipulation Instructions**

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|
| $000 | Interrupt control bits area | | | |
| $003 | | | | |
| PMRA $004 | Not used | Not used | $R4_2$/SI | $R4_3$/SO |
| SMRA $005 | $R4_1/\overline{SCK}$ | Serial transmit clock speed selection | | |
| SRL $006 | Serial data register (lower digit) | | | |
| SRU $007 | Serial data register (upper digit) | | | |
| TMA $008 | Timer-A/timer-base | Clock source selection (timer A) | | |
| TMB1 $009 | Auto-reload on/off | Clock source selection (timer B) | | |
| TRBL/TWBL $00A | Timer B register (lower digit) | | | |
| TRBU/TWBU $00B | Timer B register (upper digit) | | | |
| MIS $00C | Pull-up MOS control | SO PMOS control | Interrupt frame period selection | |
| TMC1 $00D | Auto-reload on/off | Clock source selection (timer C) | | |
| TRCL/TWCL$00E | Timer C register (lower digit) | | | |
| TRCU/TWCU $00F | Timer C register (upper digit) | | | |
| TMD1 $010 | Auto-reload on/off | Clock source selection (timer D) | | |
| TRDL/TWDL $011 | Timer D register (lower digit) | | | |
| TRDU/TWDU $012 | Timer D register (upper digit) | | | |
| TMB2 $013 | Not used | Not used | Timer B output mode selection | |
| TMC2 $014 | Not used | Timer C output mode selection | | |
| TMD2 $015 | Input capture selection | Timer D output mode selection | | |
| CCR $016 | Internal reference voltage level selection | | | |
| CER $017 | Voltage comparison result | Reference power supply selection | $COMP_0$ to $COMP_3$ selection | |
| WSR $018 | $\overline{WU}_7$ enable | $\overline{WU}_6$ enable | $\overline{WU}_5$ to $\overline{WU}_4$ enable | $\overline{WU}_3$ to $\overline{WU}_0$ enable |
| | Not used | | | |
| $020 | Register flag area | | | |
| $023 | | | | |
| PMRB $024 | $R0_3$/$INT_3$ | $R0_2$/$INT_2$ | $R0_1/\overline{INT}_1$ | $R0_0/\overline{INT}_0$ |
| PMRC $025 | Not used | $D_{11}/\overline{STOPC}$ | $R4_0$/EVND | $R3_3/\overline{EVNB}$ |
| ESR1 $026 | $INT_3$ detection edge selection | | $INT_2$ detection edge selection | |
| ESR2 $027 | EVND detection edge selection | | Not used | Not used |
| SMRB $028 | Not used | Not used | SO output level control in idle states | Serial clock source selection |
| SSR1 $029 | 32-kHz oscillation stop | 32-kHz oscillation division ratio selection | 32-kHz oscillation sampling selection | Not used |
| SSR2 $02A | Not used | Not used | OSC division ratio selection | |
| | Not used | | | |
| DCD0 $02C | Port D3 DCR | Port D2 DCR | Port D1 DCR | Port D0 DCR |
| DCD1 $02D | Port D7 DCR | Port D6 DCR | Port D5 DCR | Port D4 DCR |
| DCD2 $02E | Not used | Not used | Port D9 DCR | Port D8 DCR |
| | Not used | | | |
| DCR0 $030 | Port $R0_3$ DCR | Port $R0_2$ DCR | Port $R0_1$ DCR | Port $R0_0$ DCR |
| DCR1 $031 | Port $R1_3$ DCR | Port $R1_2$ DCR | Port $R1_1$ DCR | Port $R1_0$ DCR |
| DCR2 $032 | Port $R2_3$ DCR | Port $R2_2$ DCR | Port $R2_1$ DCR | Port $R2_0$ DCR |
| DCR3 $033 | Port $R3_3$ DCR | Port $R3_2$ DCR | Port $R3_1$ DCR | Port $R3_0$ DCR |
| DCR4 $034 | Port $R4_3$ DCR | Port $R4_2$ DCR | Port $R4_1$ DCR | Port $R4_0$ DCR |
| DCR5 $035 | Port $R5_3$ DCR | Port $R5_2$ DCR | Port $R5_1$ DCR | Port $R5_0$ DCR |
| DCR6 $036 | Port $R6_3$ DCR | Port $R6_2$ DCR | Port $R6_1$ DCR | Port $R6_0$ DCR |
| DCR7 $037 | Port $R7_3$ DCR | Port $R7_2$ DCR | Port $R7_1$ DCR | Port $R7_0$ DCR |
| DCR8 $038 | Port $R8_3$ DCR | Port $R8_2$ DCR | Port $R8_1$ DCR | Port $R8_0$ DCR |
| DCR9 $039 | Not used | Port $R9_2$ DCR | Port $R9_1$ DCR | Port $R9_0$ DCR |
| | Not used | | | |
| $03F | | | | |

**Figure 5 Special Function Register Area**

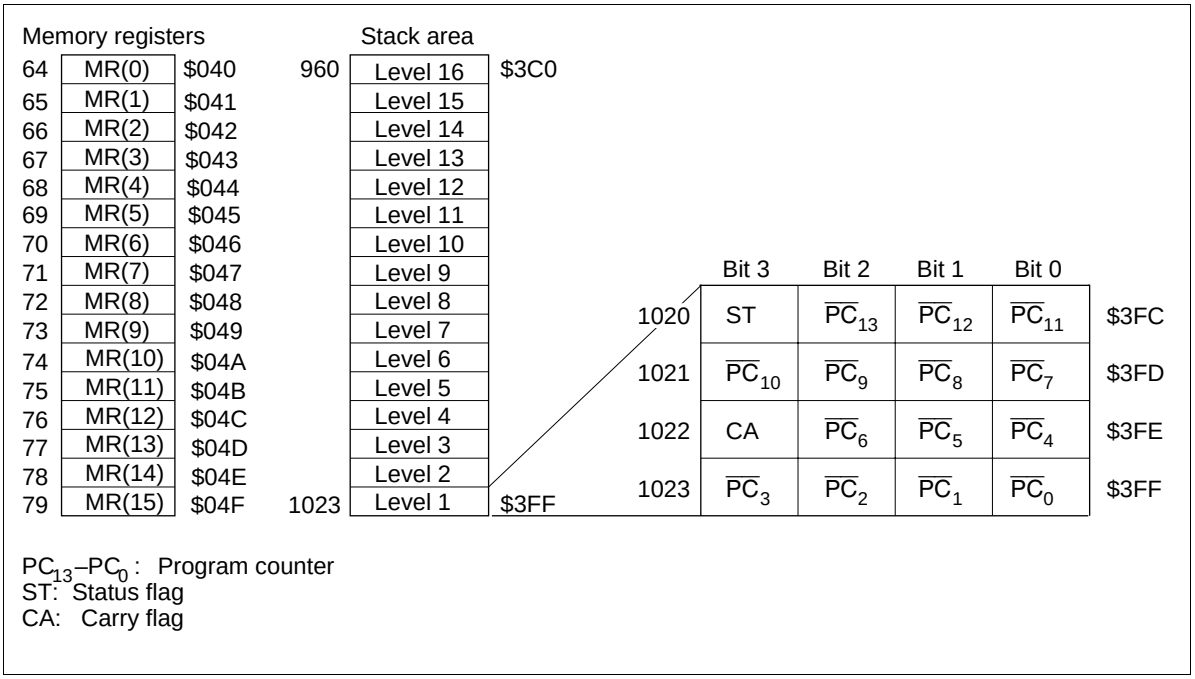



**Figure 6 Configuration of Memory Registers, Stack Area, and Stack Position**

## Functional Description

### Registers and Flags

The MCU has nine registers and two flags for CPU operations (figure 7).

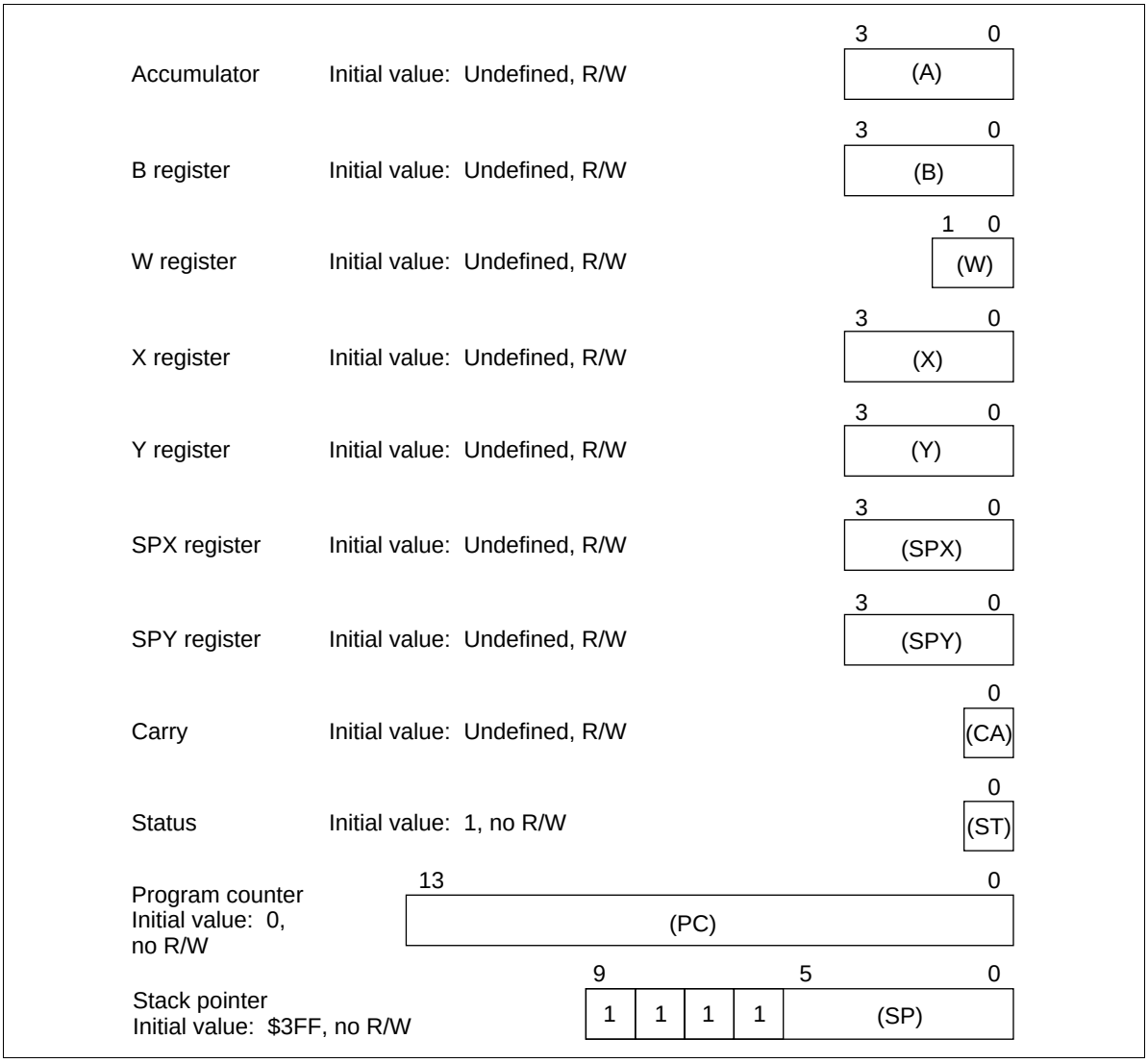



**Figure 7 Registers and Flags**

**Accumulator (A), B Register (B):** Four-bit registers used to hold the results from the arithmetic logic unit (ALU) and transfer data between memory, I/O, and other registers.

**W Register (W), X Register (X), Y Register (Y):** Two-bit (W) and four-bit (X and Y) registers used for indirect RAM addressing. The Y register is also used for D-port addressing.

**SPX Register (SPX), SPY Register (SPY):** Four-bit registers used to supplement the X and Y registers.

**Carry Flag (CA):** One-bit flag that stores any ALU overflow generated by an arithmetic operation. CA is affected by the SEC, REC, ROTL, and ROTR instructions. A carry is pushed onto the stack during an interrupt and popped from the stack by the RTNI instruction—but not by the RTN instruction.

**Status Flag (ST):** One-bit flag that latches any overflow generated by an arithmetic or compare instruction, not-zero decision from the ALU, or result of a bit test. ST is used as a branch condition of the BR, BRL, CAL, and CALL instructions. The contents of ST remain unchanged until the next arithmetic, compare, or bit test instruction is executed, but become 1 after the BR, BRL, CAL, or CALL instruction is read, regardless of whether the instruction is executed or skipped. The contents of ST are pushed onto the stack during an interrupt and popped from the stack by the RTNI instruction—but not by the RTN instruction.

**Program Counter (PC):** 14-bit binary counter that points to the ROM address of the instruction being executed.

**Stack Pointer (SP):** Ten-bit pointer that contains the address of the stack area to be used next. The SP is initialized to $3FF by MCU reset. It is decremented by 4 when data is pushed onto the stack, and incremented by 4 when data is popped from the stack. The top four bits of the SP are fixed at 1111, so a stack can be used up to 16 levels.

The SP can be initialized to $3FF also by resetting the RSP bit with the REM or REMD instruction.

### Reset

The MCU is reset by inputting a high-level voltage to the RESET pin. At power-on or when stop mode is cancelled, RESET must be high for at least one $t_{RC}$ to enable the oscillator to stabilize. During operation, RESET must be high for at least two instruction cycles.

See table 1 for initial values after MCU reset.

### Interrupts

The MCU has 10 interrupt sources: four external signals ($\overline{INT_0}$, $\overline{INT_1}$, $INT_2$, $INT_3$), four timer/counters (timers A, B, C, and D), serial interface, and wakeup.

An interrupt request flag (IF), interrupt mask (IM), and vector address are provided for each interrupt source, and an interrupt enable flag (IE) controls the entire interrupt process.

Some vector addresses are shared by two different interrupts. They are timer A and $INT_2$, timer B and $INT_3$, timer C and serial interface. So the type of request that has occurred must be checked at the beginning of interrupt processing.

**Interrupt Control Bits and Interrupt Processing:** Locations $000 to $003 and $022 to $023 in RAM are reserved for the interrupt control bits which can be accessed by RAM bit manipulation instructions.

The interrupt request flag (IF) cannot be set by software. MCU reset initializes the interrupt enable flag (IE) and the IF to 0 and the interrupt mask (IM) to 1.

Refer to figure 8 for the block diagram of the interrupt control circuit, table 2 for interrupt priorities and vector addresses, and table 3 for interrupt processing conditions for the 10 interrupt sources.

An interrupt request occurs when the IF is set to 1 and the IM is set to 0. If the IE is 1 at that point, the interrupt is processed. A priority programmable logic array (PLA) generates the vector address assigned to that interrupt source.

For the interrupt processing sequence, see figure 9, and figure 10 for an interrupt processing flowchart. After an interrupt is acknowledged, the previous instruction is completed in the first cycle. The IE is reset in the second cycle, the carry, status, and program counter values are pushed onto the stack during the second and third cycles, and the program jumps to the vector address to execute the instruction in the third cycle.

Program the JMPL instruction at each vector address, to branch the program to the start address of the interrupt program, and reset the IF by a software instruction within the interrupt program.

**Table 1 Initial Values After MCU Reset**

| Item | | Abbr. | Initial Value | Contents |
|---|---|---|---|---|
| Program counter | | (PC) | $0000 | Indicates program execution point from start address of ROM area |
| Status flag | | (ST) | 1 | Enables conditional branching |
| Stack pointer | | (SP) | $3FF | Stack level 0 |
| Interrupt flags/mask | Interrupt enable flag | (IE) | 0 | Inhibits all interrupts |
| | Interrupt request flag | (IF) | 0 | Indicates there is no interrupt request |
| | Interrupt mask | (IM) | 1 | Prevents (masks) interrupt requests |
| I/O | Port data register | (PDR) | All bits 1 | Enables output at level 1 |
| | Data control register | (DCD0, DCD1) | All bits 0 | Turns output buffer off (to high impedance) |
| | | (DCD2) | - - 00 | |
| | | (DCR0–DCR8) | All bits 0 | |
| | | (DCR9) | - 000 | |
| | Port mode register A | (PMRA) | - - 00 | Refer to description of port mode register A |
| | Port mode register B | (PMRB) | 0000 | Refer to description of port mode register B |
| | Port mode register C bits 2, 1, 0 | (PMRC2, PMRC1, PMRC0) | - 000 | Refer to description of port mode register C |
| | Detection edge select register 1 | (ESR1) | 0000 | Disables edge detection |
| | Detection edge select register 2 | (ESR2) | 00 - - | Disables edge detection |
| Timers/ counters, serial interface | Timer mode register A | (TMA) | 0000 | Refer to description of timer mode register A |
| | Timer mode register B1 | (TMB1) | 0000 | Refer to description of timer mode register B1 |
| | Timer mode register B2 | (TMB2) | - - 00 | Refer to description of timer mode register B2 |
| | Timer mode register C1 | (TMC1) | 0000 | Refer to description of timer mode register C1 |
| | Timer mode register C2 | (TMC2) | - 000 | Refer to description of timer mode register C2 |
| | Timer mode register D1 | (TMD1) | 0000 | Refer to description of timer mode register D1 |
| | Timer mode register D2 | (TMD2) | 0000 | Refer to description of timer mode register D2 |
| | Serial mode register A | (SMRA) | 0000 | Refer to description of serial mode register A |
| | Serial mode register B | (SMRB) | - - 00 | Refer to description of serial mode register B |

| Item | | Abbr. | Initial Value | Contents |
|---|---|---|---|---|
| Timers/ counters, serial interface | Prescaler S | (PSS) | $000 | — |
| | Prescaler W | (PSW) | $00 | — |
| | Timer counter A | (TCA) | $00 | — |
| | Timer counter B | (TCB) | $00 | — |
| | Timer counter C | (TCC) | $00 | — |
| | Timer counter D | (TCD) | $00 | — |
| | Timer write register B | (TWBU, TWBL) | $X0 | — |
| | Timer write register C | (TWCU, TWCL) | $X0 | — |
| | Timer write register D | (TWDU, TWDL) | $X0 | — |
| | Octal counter | | 000 | — |
| I/O | Wakeup set register | (WSR) | 0000 | — |
| Voltage comparator | Comparator enable register | (CER) | 0000 | — |
| | Comparator control register | (CCR) | 0000 | — |
| Bit register | Low speed on flag | (LSON) | 0 | Refer to description of operating modes |
| | Watchdog timer on flag | (WDON) | 0 | Refer to description of timer C |
| | Comparator start flag | (CMSF) | 0 | Refer to description of voltage comparator |
| | Direct transfer on flag | (DTON) | 0 | Refer to description of operating modes |
| | Input capture status flag | (ICSF) | 0 | Refer to description of timer D |
| | Input capture error flag | (ICEF) | 0 | Refer to description of timer D |
| Others | Miscellaneous register | (MIS) | 0000 | Refer to description of operating modes, and oscillator circuit |
| | System clock select register 1 bits 2, 1 | (SSR12–SSR11) | 00 | Refer to description of operating modes, and oscillator circuit |
| | System clock select register 2 | (SSR2) | - - 00 | Switches OSC division ratio |

Notes: 1. The statuses of other registers and flags after MCU reset are shown in the following table.
2. X indicates invalid value. - indicates that the bit does not exist.

| Item | Abbr. | Status After Cancellation of Stop Mode by $\overline{\text{STOPC}}$ Input | Status After Cancellation of Stop Mode by MCU Reset | Status After all Other Types of Reset |
|---|---|---|---|---|
| Carry flag | (CA) | Pre-stop-mode values are not guaranteed; values must be initialized by program | | Pre-MCU-reset values are not guaranteed; values must be initialized by program |
| Accumulator | (A) | | | |
| B register | (B) | | | |
| W register | (W) | | | |
| X/SPX register | (X/SPX) | | | |
| Y/SPY register | (Y/SPY) | | | |
| Serial data register | (SRL, SRU) | | | |
| RAM | | Pre-stop-mode values are retained | | |
| RAM enable flag | (RAME) | 1 | 0 | 0 |
| Port mode register C bit 2 | (PMRC) | Pre-stop-mode values are retained | 0 | 0 |
| System clock select register1 bit 3 | (SSR13) | | | |

**Table 2 Vector Addresses and Interrupt Priorities**

| Reset/Interrupt | Priority | Vector Address |
|---|---|---|
| RESET, $\overline{\text{STOPC}}$* | — | $0000 |
| $\overline{\text{INT}}_0$ | 1 | $0002 |
| $\overline{\text{INT}}_1$ | 2 | $0004 |
| Timer D | 3 | $0006 |
| Timer A, $\text{INT}_2$ | 4 | $0008 |
| Timer B, $\text{INT}_3$ | 5 | $000A |
| Timer C, Serial | 6 | $000C |
| Wakeup | 7 | $000E |

Note: * The $\overline{\text{STOPC}}$ interrupt request is valid only in stop mode.

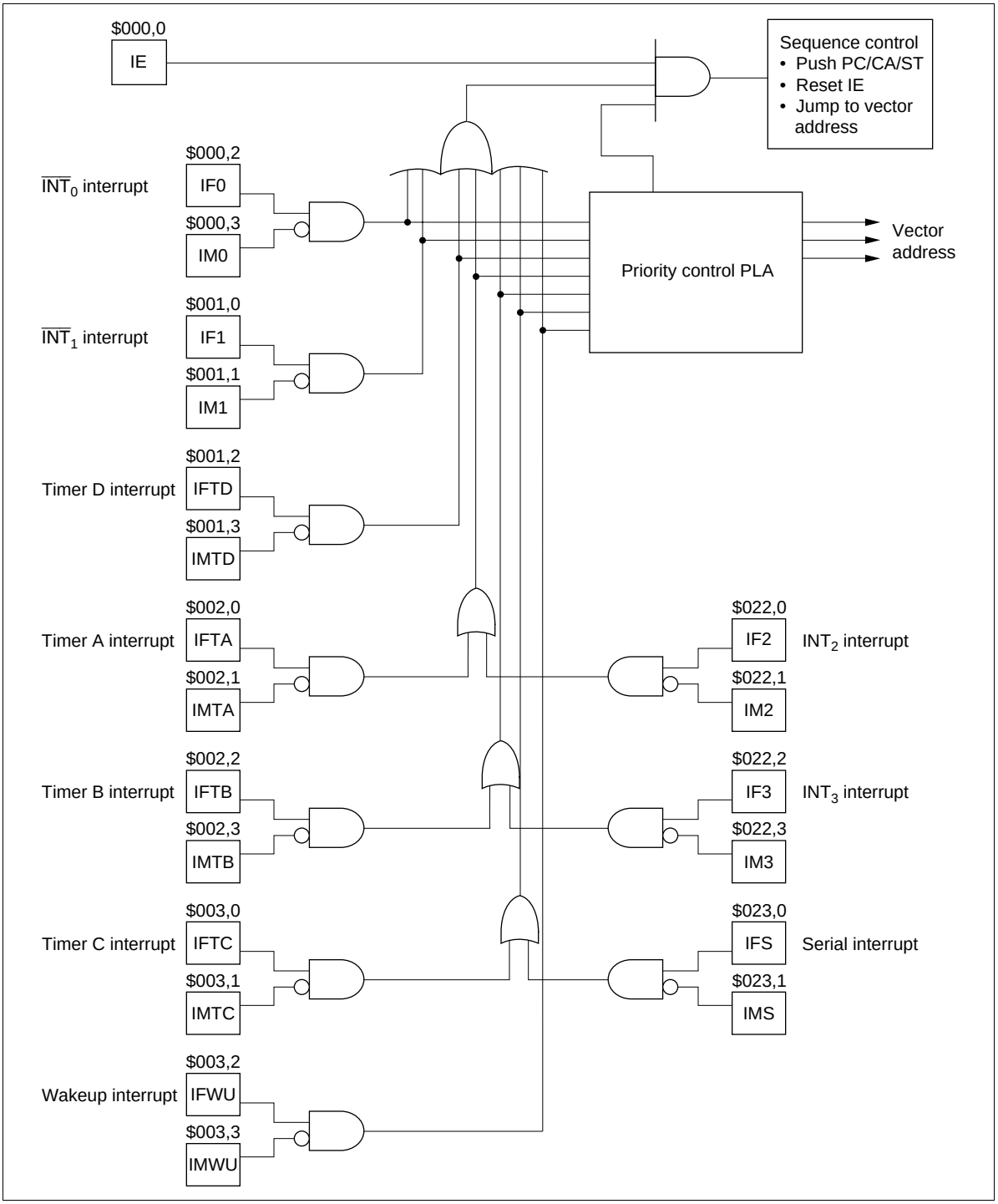



**Figure 8 Interrupt Control Circuit**

**Table 3 Interrupt Processing and Activation Conditions**

| Interrupt Control Bit | Interrupt Source | | | | | | |
|---|---|---|---|---|---|---|---|
| | $\overline{INT_0}$ | $\overline{INT_1}$ | Timer D | Timer A or $INT_2$ | Timer B or $INT_3$ | Timer C or Serial | Wakeup |
| IE | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| IF0 · $\overline{IM0}$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| IF1 · $\overline{IM1}$ | * | 1 | 0 | 0 | 0 | 0 | 0 |
| IFTD · $\overline{IMTD}$ | * | * | 1 | 0 | 0 | 0 | 0 |
| IFTA · $\overline{IMTA}$ + IF2 · $\overline{IM2}$ | * | * | * | 1 | 0 | 0 | 0 |
| IFTB · $\overline{IMTB}$ + IF3 · $\overline{IM3}$ | * | * | * | * | 1 | 0 | 0 |
| IFTC · $\overline{IMTC}$ + IFS · $\overline{IMS}$ | * | * | * | * | * | 1 | 0 |
| IFWU · $\overline{IMWU}$ | * | * | * | * | * | * | 1 |

Note: Bits marked by * can be either 0 or 1. Their values have no effect on operation.

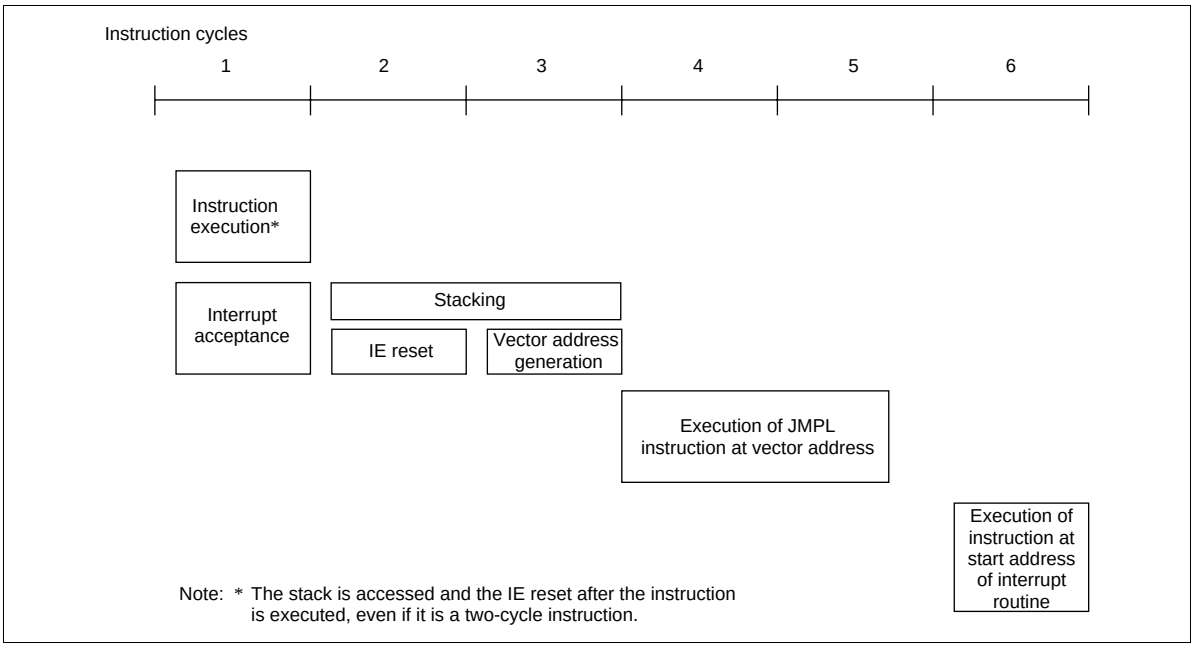



**Figure 9 Interrupt Processing Sequence**

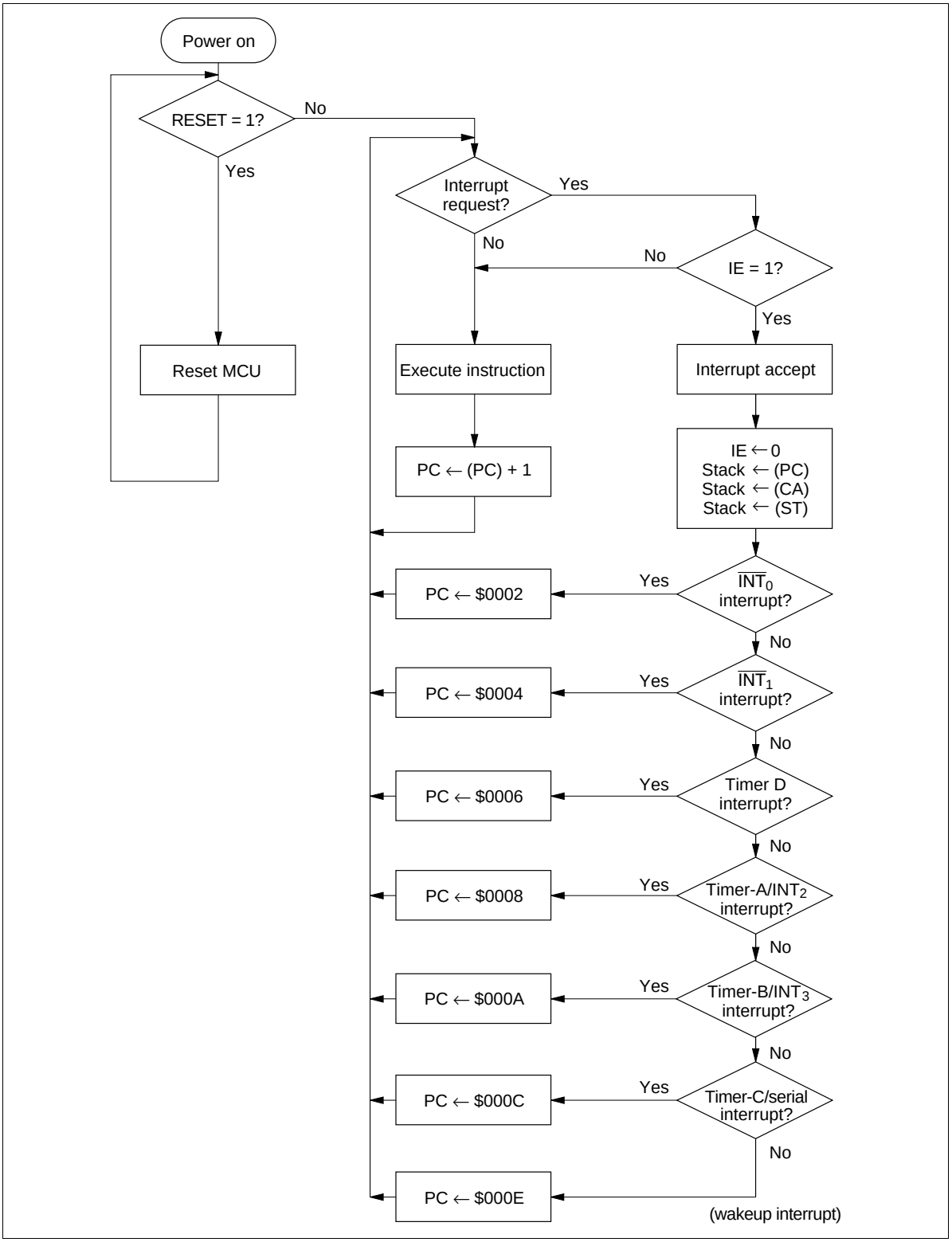



**Figure 10 Interrupt Processing Flowchart**

**Interrupt Enable Flag (IE: $000, Bit 0):** Controls the entire interrupt process. It is reset by the interrupt processing and set by the RTNI instruction. Refer to table 4.

**Table 4 Interrupt Enable Flag (IE: $000, Bit 0)**

| IE | Interrupt Enabled/Disabled |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

**External Interrupts ($\overline{INT}_0$, $\overline{INT}_1$, $INT_2$, $INT_3$, $\overline{WU}_0$–$\overline{WU}_7$):** Five external interrupt signals.

**External Interrupt Request Flags (IF0, IF1, IF2, IF3, IFWU: $000, $001, $003, $022):** IF0, IF1, and IFWU are set at the falling edge of input signals, and IF2 and IF3 are set at the rising or falling edge or both rising and falling edges of input signals (table 5). $INT_2$ and $INT_3$ interrupt edges are selected by the detection edge select register (ESR1: $026) (figure 11).

**Table 5 External Interrupt Request Flags (IF0–IF3, IFWU: $000, $001, $003, $022)**

| IF0–IF3, IFWU | Interrupt Request |
|---|---|
| 0 | No |
| 1 | Yes |

**Detection edge selection register 1 (ESR1: $026)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | ESR13 | ESR12 | ESR11 | ESR10 |

| ESR13 | ESR12 | $INT_3$ detection edge |
|---|---|---|
| 0 | 0 | No detection |
| | 1 | Falling-edge detection |
| 1 | 0 | Rising-edge detection |
| | 1 | Double-edge detection* |

| ESR11 | ESR10 | $INT_2$ detection edge |
|---|---|---|
| 0 | 0 | No detection |
| | 1 | Falling-edge detection |
| 1 | 0 | Rising-edge detection |
| | 1 | Double-edge detection* |

Note: * Both falling and rising edges are detected.

**Figure 11 Detection Edge Selection Register 1 (ESR1)**

**External Interrupt Masks (IM0, IM1, IM2, IM3, IMWU: $000, $001, $003, $022):** Prevent (mask) interrupt requests caused by the corresponding external interrupt request flags (table 6).

**Table 6 External Interrupt Masks (IM0–1M3, IMWU: $000, $001, $003, $022)**

| IM0–IM3, IMWU | Interrupt Request |
|---|---|
| 0 | Enabled |
| 1 | Disabled (masked) |

**Timer A Interrupt Request Flag (IFTA: $002, Bit 0):** Set by overflow output from timer A (table 7).

**Table 7 Timer A Interrupt Request Flag (IFTA: $002, Bit 0)**

| IFTA | Interrupt Request |
|---|---|
| 0 | No |
| 1 | Yes |

**Timer A Interrupt Mask (IMTA: $002, Bit 1):** Prevents (masks) an interrupt request caused by the timer A interrupt request flag (table 8).

**Table 8 Timer A Interrupt Mask (IMTA: $002, Bit 1)**

| IMTA | Interrupt Request |
|---|---|
| 0 | Enabled |
| 1 | Disabled (masked) |

**Timer B Interrupt Request Flag (IFTB: $002, Bit 2):** Set by overflow output from timer B (table 9).

**Table 9 Timer B Interrupt Request Flag (IFTB: $002, Bit 2)**

| IFTB | Interrupt Request |
|---|---|
| 0 | No |
| 1 | Yes |

**Timer B Interrupt Mask (IMTB: $002, Bit 3):** Prevents (masks) an interrupt request caused by the timer B interrupt request flag (table 10).

**Table 10 Timer B Interrupt Mask (IMTB: $002, Bit 3)**

| IMTB | Interrupt Request |
|---|---|
| 0 | Enabled |
| 1 | Disabled (masked) |

**Timer C Interrupt Request Flag (IFTC: $003, Bit 0):** Set by overflow output from timer C (table 11).

**Table 11 Timer C Interrupt Request Flag (IFTC: $003, Bit 0)**

| IFTC | Interrupt Request |
|---|---|
| 0 | No |
| 1 | Yes |

**Timer C Interrupt Mask (IMTC: $003, Bit 1):** Prevents (masks) an interrupt request caused by the timer C interrupt request flag (table 12).

**Table 12 Timer C Interrupt Mask (IMTC: $003, Bit 1)**

| IMTC | Interrupt Request |
|---|---|
| 0 | Enabled |
| 1 | Disabled (masked) |

**Timer D Interrupt Request Flag (IFTD: $001, Bit 2):** Set by overflow output from timer D, or by the rising or falling edge of signals input to EVND when the input capture function is used (table 13).

**Table 13 Timer D Interrupt Request Flag (IFTD: $001, Bit 2)**

| IFTD | Interrupt Request |
|---|---|
| 0 | No |
| 1 | Yes |

**Timer D Interrupt Mask (IMTD: $001, Bit 3):** Prevents (masks) an interrupt request caused by the timer D interrupt request flag (table 14).

**Table 14 Timer D Interrupt Mask (IMTD: $001, Bit 3)**

| IMTD | Interrupt Request |
|---|---|
| 0 | Enabled |
| 1 | Disabled (masked) |

**Serial Interrupt Request Flags (IFS: $023, Bit 0):** Set when data transfer is completed or when data transfer is suspended (table 15).

**Table 15 Serial Interrupt Request Flag (IFS: $023, Bit 0)**

| IFS | Interrupt Request |
|---|---|
| 0 | No |
| 1 | Yes |

**Serial Interrupt Mask (IMS: $023, Bit 1):** Prevents (masks) an interrupt request caused by the serial interrupt request flag (table 16).

**Table 16 Serial Interrupt Mask (IMS: $023, Bit 1)**

| IMS | Interrupt Request |
|---|---|
| 0 | Enabled |
| 1 | Disabled (masked) |

**Wakeup Interrupt Request Flag (IFWU: $003, Bit 2):** Set by the falling edge of signals input to wakeup (table 17).

**Table 17 Wakeup Interrupt Request Flag (IFWU: $003, Bit 2)**

| IFWU | Interrupt Request |
|---|---|
| 0 | No |
| 1 | Yes |

**Wakeup Interrupt Mask (IMWU: $003, Bit 3):** Prevents (masks) an interrupt request caused by the wakeup interrupt request flag (table 18).

**Table 18 Wakeup Interrupt Mask (IMWU: $003, Bit 3)**

| IMWU | Interrupt Request |
|---|---|
| 0 | Enabled |
| 1 | Disabled (masked) |

**Wakeup Function:** Detects the falling edge of wakeup input signals and sets the wakeup interrupt request flag (IFWU: $003, bit 2). Refer to figure 12 for a block diagram showing the wakeup interrupt. The wakeup select register (WSR: $018) can select from one to eight wakeup inputs ($\overline{WU}_0$–$\overline{WU}_7$) (figure 13). The wakeup function can operate in any mode other than stop mode. When the wakeup interrupt is received, the CPU generates an independent vector address ($000E).

Note: The wakeup select register (WSR: $018) controls whether the wakeup input is to be valid or invalid, but it can not switch the pin inputs between the R ports and wakeup. When using the pins only as R ports, nullify wakeup input or set the wakeup interrupt mask (IMWU: $003, bit 3).

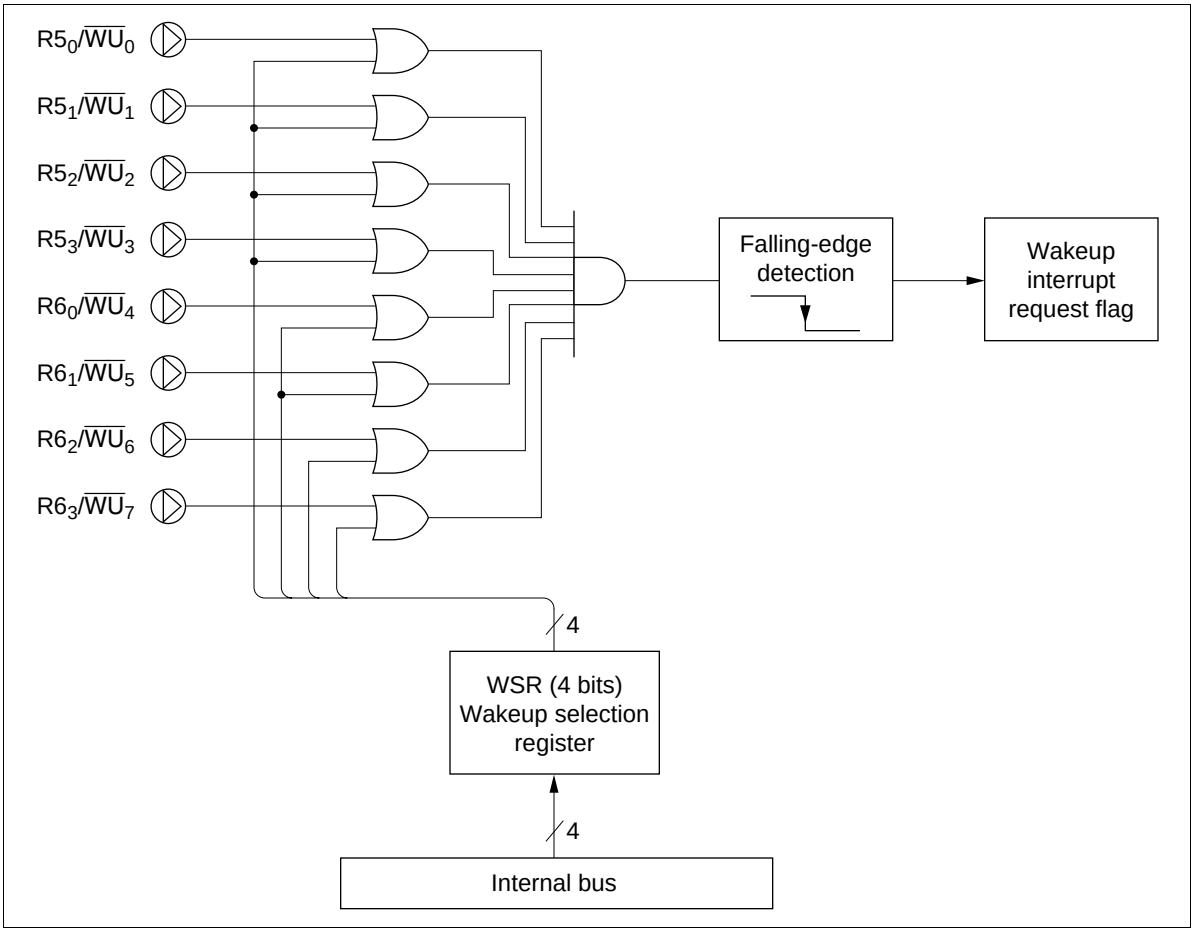



**Figure 12 Wakeup Interrupt**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | WSR3 | WSR2 | WSR1 | WSR0 |

| WSR0 | $\overline{WU}_0$ to $\overline{WU}_3$ control |
|---|---|
| 0 | Invalid |
| 1 | Valid |

| WSR1 | $\overline{WU}_4$ to $\overline{WU}_5$ control |
|---|---|
| 0 | Invalid |
| 1 | Valid |

| WSR2 | $\overline{WU}_6$ control |
|---|---|
| 0 | Invalid |
| 1 | Valid |

| WSR3 | $\overline{WU}_7$ control |
|---|---|
| 0 | Invalid |
| 1 | Valid |

**Figure 13 Wakeup Select Register (WSR)**

## Operating Modes

The MCU has five operating modes (table 19). Refer to tables 20 and 21 for the operations in each mode, and figure 14 for the transitions between operating modes.

**Active Mode:** All MCU functions operate according to the clock generated by the system oscillators $OSC_1$ and $OSC_2$.

**Table 19 Operating Modes and Clock Status**

| | | Mode Name | | | | |
|---|---|---|---|---|---|---|
| | | **Active** | **Standby** | **Stop** | **Watch** | **Subactive*[2]** |
| Activation method | | RESET cancellation, interrupt request, $\overline{STOPC}$ cancellation in stop mode, STOP/SBY instruction in subactive mode (when direct transfer is selected) | SBY instruction | STOP instruction when TMA3 = 0 | STOP instruction when TMA3 = 1 | $\overline{INT_0}$, timer A or wakeup interrupt request from watch mode |
| Status | System oscillator | OP | OP | Stopped | Stopped | Stopped |
| | Subsystem oscillator | OP | OP | OP*[1] | OP | OP |
| Cancellation method | | RESET input, STOP/SBY instruction | RESET input, interrupt request | RESET input, $\overline{STOPC}$ input in stop mode | RESET input, $\overline{INT_0}$, timer A or wakeup interrupt request | RESET input, STOP/SBY instruction |

Note: OP implies in operation

1. Operating or stopping the oscillator can be selected by setting bit 3 of the system clock select register (SSR1 : $029).
2. Subactive mode is an optional function; specify it on the function option list.

**Table 20 Operations in Low-Power Dissipation Modes**

| Function | Stop Mode | Watch Mode | Standby Mode | Subactive Mode*[2] |
|---|---|---|---|---|
| CPU | Reset | Retained | Retained | OP |
| RAM | Retained | Retained | Retained | OP |
| Timer A | Reset | OP | OP | OP |
| Timer B | Reset | Stopped | OP | OP |
| Timer C | Reset | Stopped | OP | OP |
| Timer D | Reset | Stopped | OP | OP |
| SCI | Reset | Stopped*[3] | OP | OP |
| Comparator | Reset | Stopped | OP | Stopped |
| I/O | Reset*[1] | Retained | Retained | OP |

Note: OP implies in operation

1. Output pins are at high impedance.
2. Subactive mode is an optional function to be specified on the function option list.
3. Transmission/reception is activated if a clock is input in external clock mode. However, all interrupts stop.

**Table 21 I/O Status in Low-Power Dissipation Modes**

| | Output | | Input |
|---|---|---|---|
| | Standby Mode, Watch Mode | Stop Mode | Active Mode, Subactive Mode |
| $D_0$–$D_9$ | Retained | High impedance | Input enabled |
| $D_{10}$–$D_{11}$ | — | — | Input enabled |
| R0–R8<br>$R9_0$, $R9_1$, $R9_2$ | Retained or output of peripheral functions | High impedance | Input enabled |
| $R9_3$, RA | — | — | Input enabled |

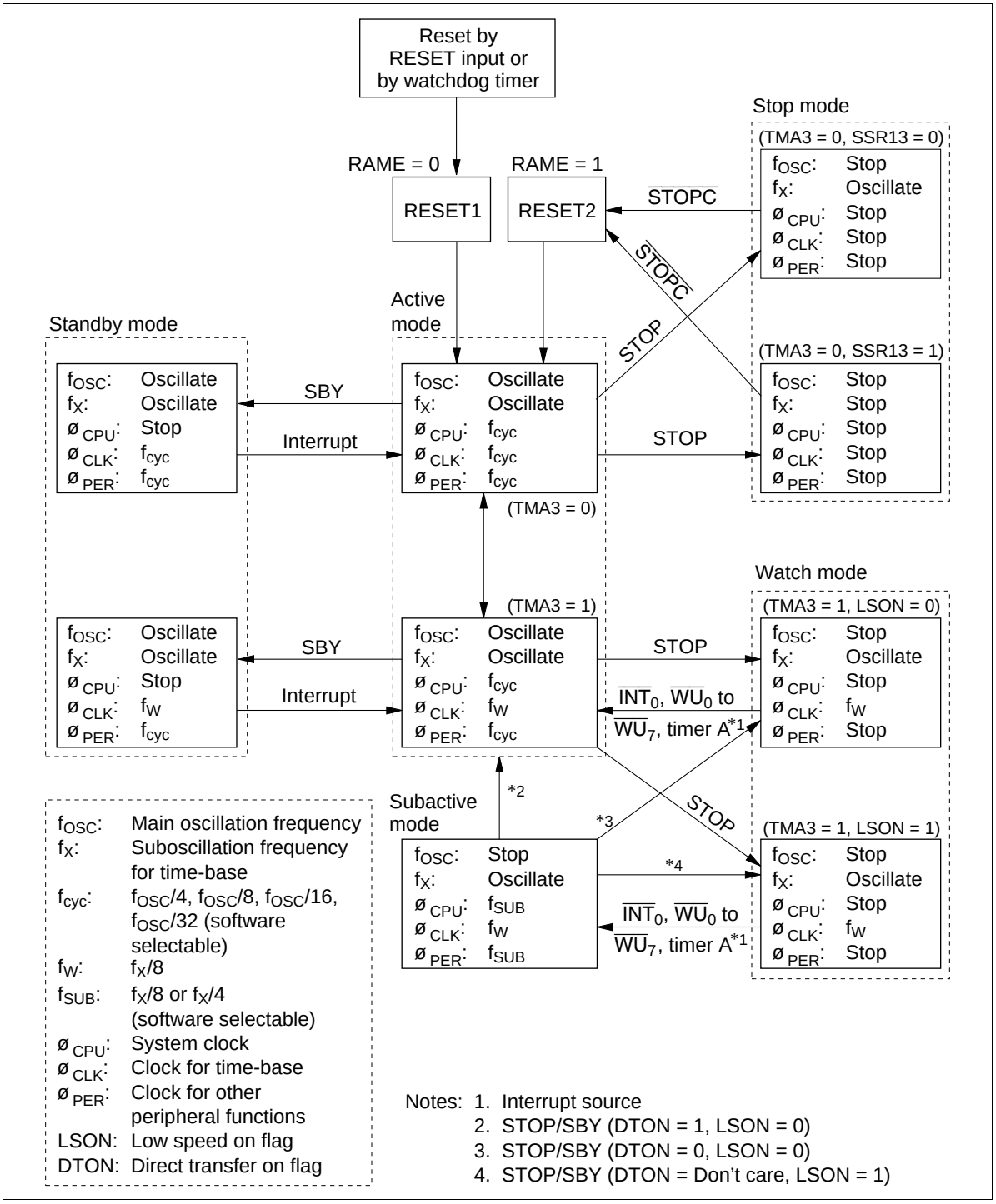



**Figure 14 MCU Status Transitions**

**Standby Mode:** In standby mode, the oscillators continue to operate, but the clocks related to instruction execution stop. Therefore, the CPU operation stops, but all RAM and register contents are retained, and the D or R port status, when set to output, is maintained. Peripheral functions such as interrupts, timers, and serial interface continue to operate. The power dissipation in this mode is lower than in active mode since the CPU halts.

The MCU enters standby mode when the SBY instruction is executed in active mode.

Standby mode is terminated by RESET input or an interrupt request. If it is terminated by RESET, the MCU is reset as well. After an interrupt request, the MCU enters active mode and executes the next instruction after the SBY instruction. If the interrupt enable flag is 1, the interrupt is then processed; if it is 0, the interrupt request is left pending and normal instruction execution continues. See figure 15 for the flowchart of operation in standby mode.

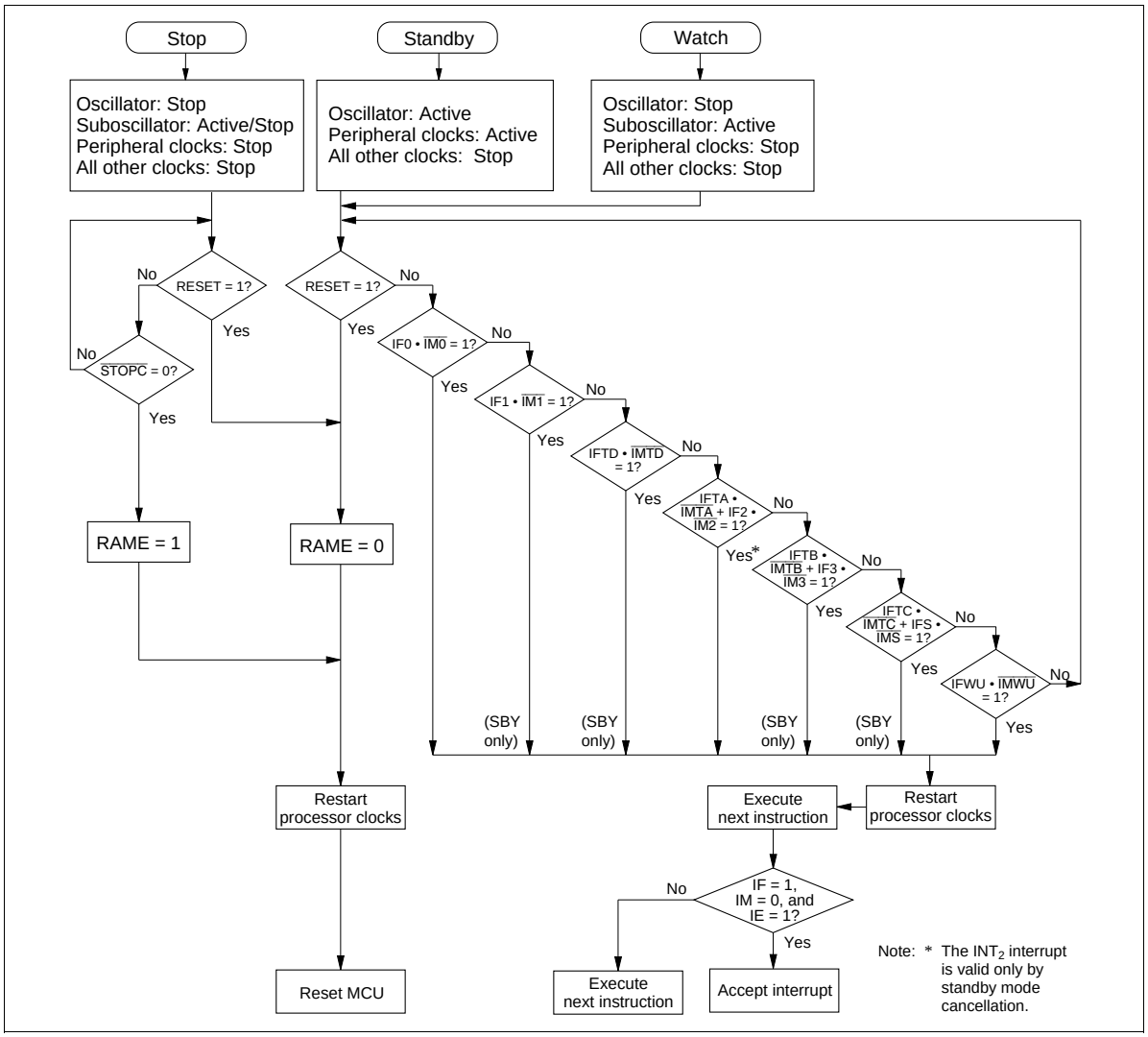



**Figure 15 MCU Operation Flowchart**

**Stop Mode:** In stop mode, all MCU operations stop and RAM data is retained. Therefore, the power dissipation in this mode is the least of all modes. The $OSC_1$ and $OSC_2$ oscillator stops. Operation of the X1 and X2 oscillator can be selected by setting bit 3 of the system clock select register (SSR1: $029; operating: SSR13 = 0, stop: SSR13 = 1) (figure 24). The MCU enters stop mode if the STOP instruction is executed in active mode when bit 3 of timer mode register A (TMA: $008) is set to 0 (TMA3 = 0) (figure 40).

Stop mode is terminated by RESET input or $\overline{STOPC}$ input (figure 16). RESET or $\overline{STOPC}$ must be applied for at least one $t_{RC}$ to stabilize oscillation (refer to the AC Characteristics section). When the MCU restarts after stop mode is cancelled, all RAM contents before entering stop mode are retained, but the accuracy of the contents of the accumulator, B register, W register, X/SPX register, Y/SPY register, carry flag, and serial data register cannot be guaranteed.

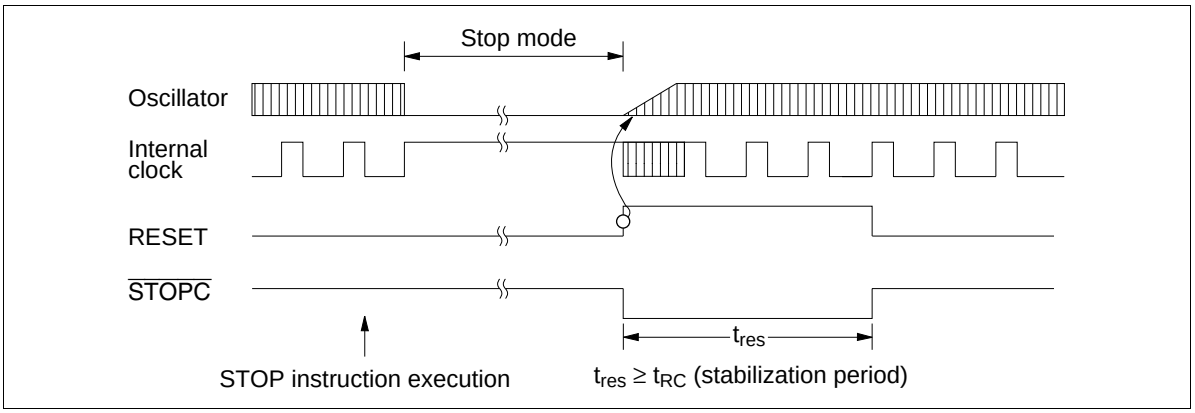



**Figure 16 Timing of Stop Mode Cancellation**

**Watch Mode:** In watch mode, the clock function (timer A) using the X1 and X2 oscillator operate but other function operations stop. Therefore, the power dissipation in this mode is the second least to stop mode, and is also convenient when only clock display is used. In this mode, the $OSC_1$ and $OSC_2$ oscillator stops, but the X1 and X2 oscillator operates. The MCU enters watch mode if the STOP instruction is executed in active mode when TMA3 = 1, or if the STOP or SBY instruction is executed in subactive mode.

Watch mode is terminated by a RESET input, timer A interrupt request, $\overline{INT_0}$ interrupt request, or wakeup interrupt request. For details of RESET input, refer to the Stop Mode section. When terminated by a timer A interrupt request, an $\overline{INT_0}$ nterrupt request, or wakeup interrupt request, the MCU enters active mode if LSON is 0 or subactive mode if LSON is 1. After an interrupt request is generated, the time required to enter active mode is $t_{RC}$ for a timer A interrupt, and $T_X$ (where $T + t_{RC} < T_X < 2T + t_{RC}$) for an $\overline{INT_0}$ interrupt, as shown in figure 17.

Operation during mode transition is the same as that at standby mode cancellation (figure 15).

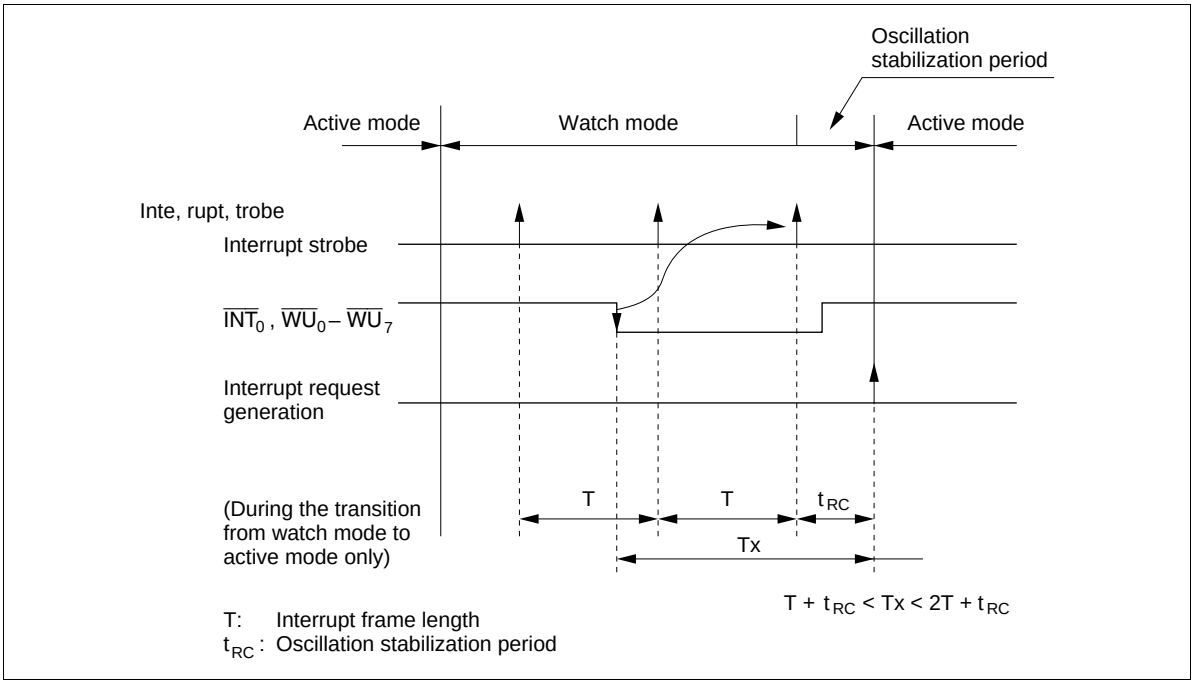



**Figure 17 Interrupt Frame**

**Subactive Mode:** The $OSC_1$ and $OSC_2$ oscillator stops and the MCU operates with a clock generated by the X1 and X2 oscillator. In this mode, functions other than the voltage comparator operate. However, because the operating clock is slow, the power dissipation becomes low, next to watch mode.

The CPU instruction execution speed can be selected as 244 µs or 122 µs by setting bit 2 (SSR12) of the system clock select register (SSR1: $029). Note that the SSR12 value must be changed in active mode. If the value is changed in subactive mode, the MCU may malfunction.

When the STOP or SBY instruction is executed in subactive mode, the MCU enters either watch or active mode, depending on the statuses of the low speed on flag (LSON: $020, bit 0) and the direct transfer on flag (DTON: $020, bit 3).

Subactive mode is an optional function that the user must specify on the function option list.

**Interrupt Frame:** In watch and subactive modes, $ø_{CLK}$ is applied to timer A and the $\overline{INT_0}$ and $\overline{WU_0}$–$\overline{WU_7}$ circuits. Prescaler W and timer A operate as the time-base and generate the timing clock for the interrupt frame. Three interrupt frame lengths (T) can be selected by setting the miscellaneous register (MIS: $00C) (figure 18).

In watch and subactive modes, a timer A/ $\overline{INT_0}$ wakeup interrupt is generated synchronously with the interrupt frame. The interrupt request is generated synchronously with the interrupt strobe timing except during transition to active mode. The falling edge of the $\overline{INT_0}$ and $\overline{WU_0}$–$\overline{WU_7}$ signals is input asynchronously with the interrupt frame timing, but it is regarded as input synchronously with the second interrupt strobe clock after the falling edge. An overflow and interrupt request in timer A is generated synchronously with the interrupt strobe timing.

**Miscellaneous register (MIS: $00C)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | MIS3 | MIS2 | MIS1 | MIS0 |

| MIS3 | MIS2 |
|---|---|
| Buffer control. Refer to figure 39. | |

| MIS1 | MIS0 | T *1 | $t_{RC}$ *1 | Oscillation circuit conditions |
|---|---|---|---|---|
| 0 | 0 | 0.24414 ms | 0.12207 ms | External clock input |
| | | | 0.24414 ms*2 | |
| | 1 | 15.625 ms | 7.8125 ms | |
| 1 | 0 | 125 ms | 62.5 ms | Ceramic or crystal oscillator |
| | 1 | Not used | | — |

Notes: 1. The values of T and $t_{RC}$ are applied when a 32.768-kHz crystal oscillator is used.
2. The value is applied only when direct transfer operation is used.

**Figure 18 Miscellaneous Register (MIS)**

**Direct Transition from Subactive Mode to Active Mode:** Available by controlling the direct transfer on flag (DTON: $020, bit 3) and the low speed on flag (LSON: $020, bit 0). The procedures are described below:

1. Set LSON to 0 and DTON to 1 in subactive mode.
2. Execute the STOP or SBY instruction.
3. The MCU automatically enters active mode from subactive mode after waiting for the MCU internal processing time and oscillation stabilization time (figure 19).

Notes: The DTON flag ($020, bit 3) can be set only in subactive mode. It is always reset in active mode.
The transition time ($T_D$) from subactive mode to active mode is:
$t_{RC} < T_D < T + t_{RC}$

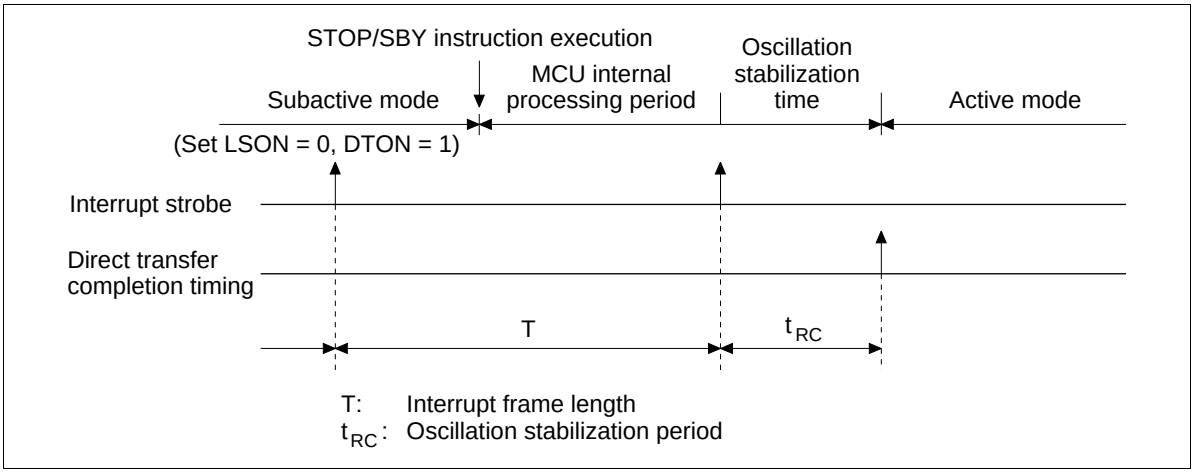



**Figure 19 Direct Transition Timing**

**Stop Mode Cancellation by $\overline{\text{STOPC}}$ :** The MCU enters active mode from stop mode by a $\overline{\text{STOPC}}$ input as well as by RESET. In either case, the MCU starts instruction execution from the starting address (address 0) of the program. However, the value of the RAM enable flag (RAME: $021, bit 3) differs between cancellation by $\overline{\text{STOPC}}$ and by RESET. When stop mode is cancelled by RESET, RAME = 0; when cancelled by $\overline{\text{STOPC}}$, RAME = 1. RESET can cancel all modes, but $\overline{\text{STOPC}}$ is valid only in stop mode; $\overline{\text{STOPC}}$ input is ignored in other modes. Therefore, when the program requires to confirm that stop mode has been cancelled by $\overline{\text{STOPC}}$ (i.e., when the RAM contents before entering stop mode are used after transition to active mode), execute the TEST instruction on the RAM enable flag (RAME) at the beginning of the program.

**MCU Operation Sequence:** See figures 20 to 22 for the MCU operation sequences. It is reset by an asynchronous RESET input, regardless of its status.

The low-power mode operation sequence is shown in figure 22. With the IE flag cleared and an interrupt flag set together with its interrupt mask cleared, if a STOP/SBY instruction is executed, the instruction is cancelled (regarded as an NOP) and the following instruction is executed. Before executing a STOP/SBY instruction, make sure all interrupt flags are cleared or all interrupts are masked.

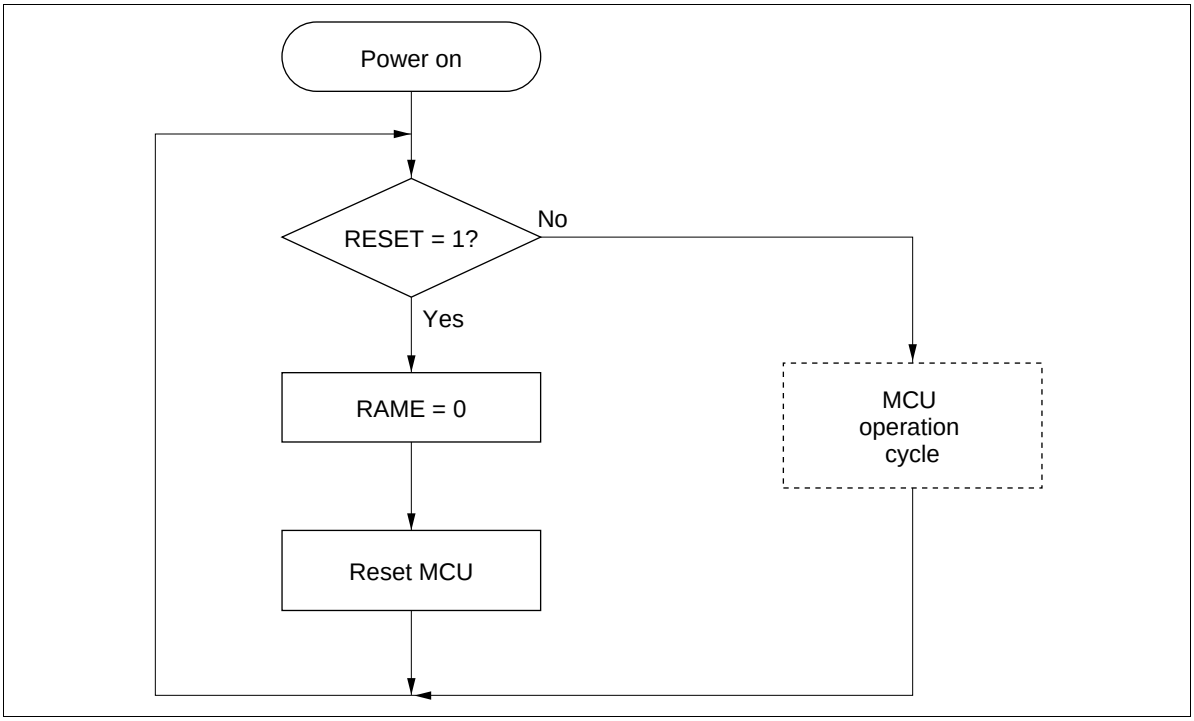
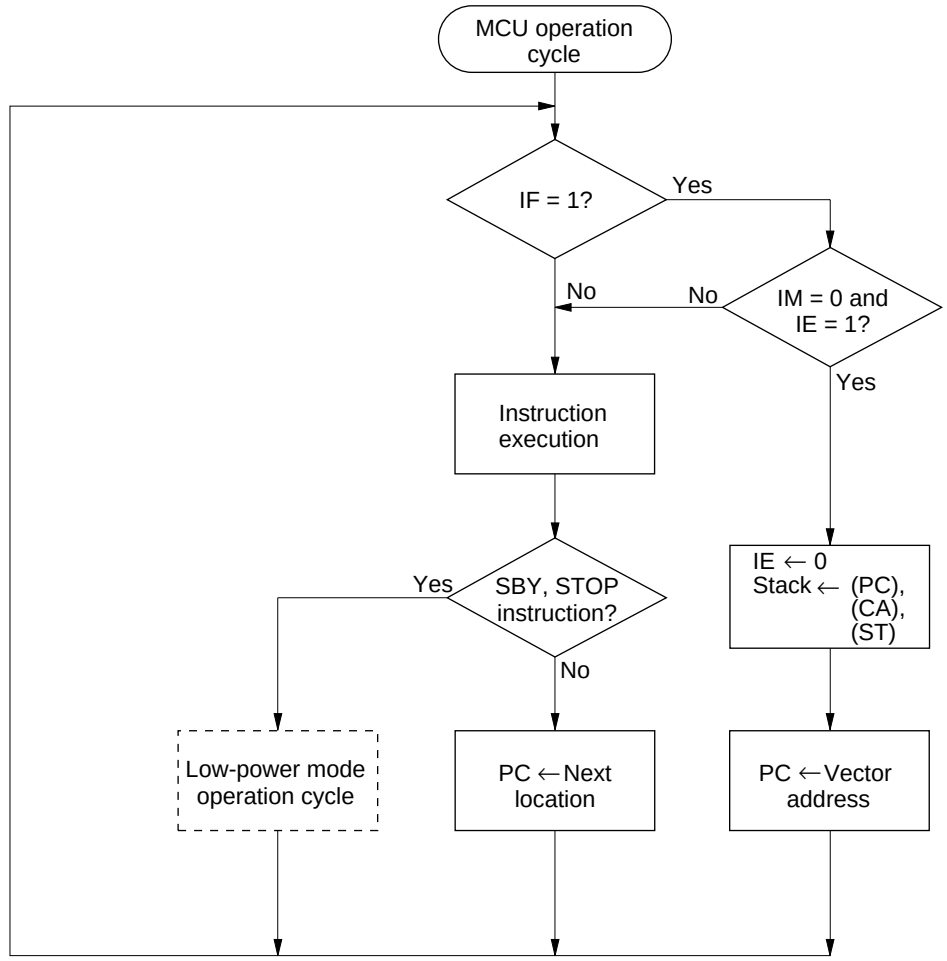
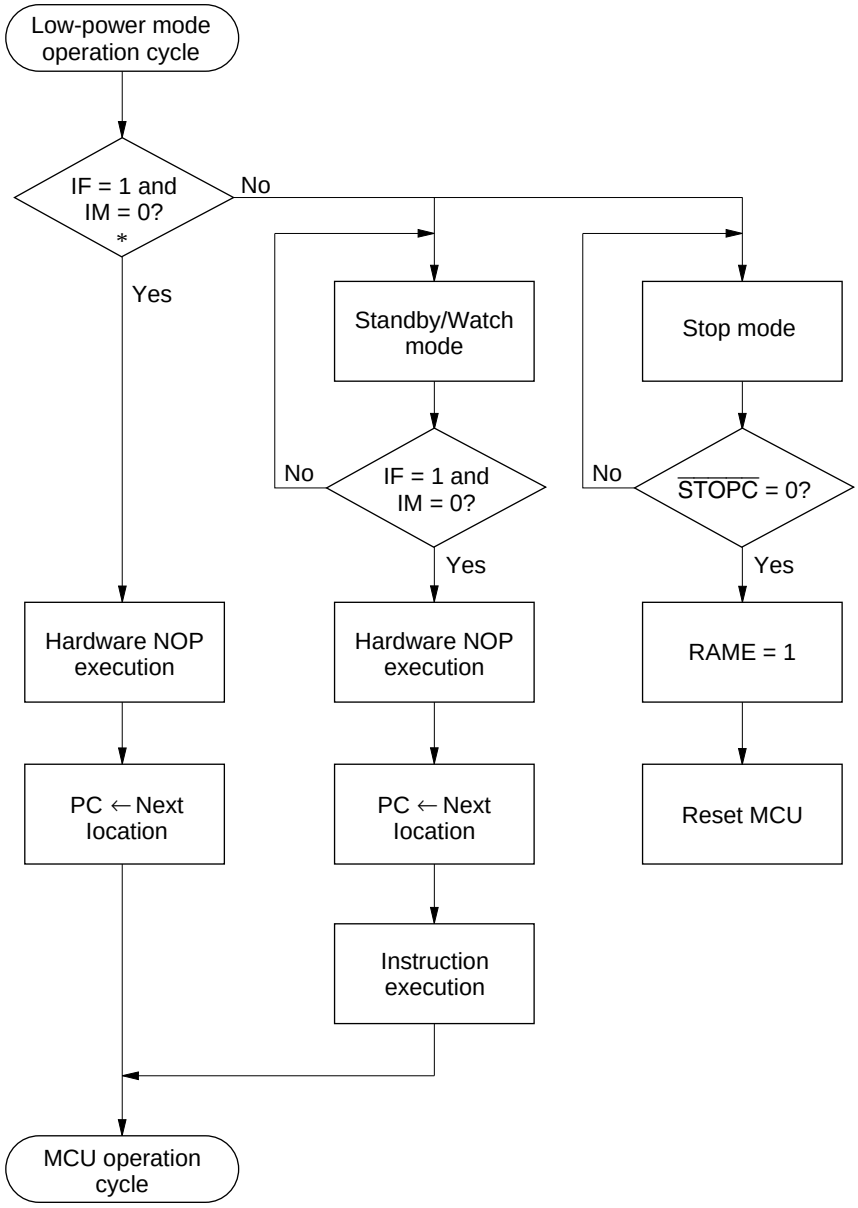



**Figure 20 MCU Operating Sequence (Power On)**




IF: Interrupt request flag
IM: Interrupt mask
IE: Interrupt enable flag
PC: Program counter
CA: Carry flag
ST: Status flag

**Figure 21 MCU Operating Sequence (MCU Operation Cycle)**




Note: * For IF and IM operation, refer to figure 15.

**Figure 22 MCU Operating Sequence (Low-Power Mode Operation)**

**Notes on Use:**

- When the MCU is in watch mode or subactive mode, if the high level period before the falling edge of $\overline{INT_0}$ and $\overline{WU_0}$–$\overline{WU_7}$ is shorter than the interrupt frame, $\overline{INT_0}$ and $\overline{WU_0}$–$\overline{WU_7}$ will not be detected. Also, if the low level period after the falling edge of $\overline{INT_0}$ and $\overline{WU_0}$–$\overline{WU_7}$ is shorter than the interrupt frame, $\overline{INT_0}$ and $\overline{WU_0}$–$\overline{WU_7}$ will not be detected.

  Edge detection is shown in figure 23. The level of the $\overline{INT_0}$ and $\overline{WU_0}$–$\overline{WU_7}$ signals are sampled by a sampling clock. When this sampled value changes from high to low, a falling edge is detected.

  In figure 24, the level of the $\overline{INT_0}$ and $\overline{WU_0}$–$\overline{WU_7}$ signals are sampled by an interrupt frame. In (a) the sampled value is low at point A, and also low at point B. Therefore, a falling edge will not be detected. In (b), the sampled value is high at point A, and also high at point B. A falling edge will not be detected in this case either.

  When the MCU is in watch mode or subactive mode, keep the high level and low level periods of $\overline{INT_0}$ and $\overline{WU_0}$–$\overline{WU_7}$ longer than interrupt frame.

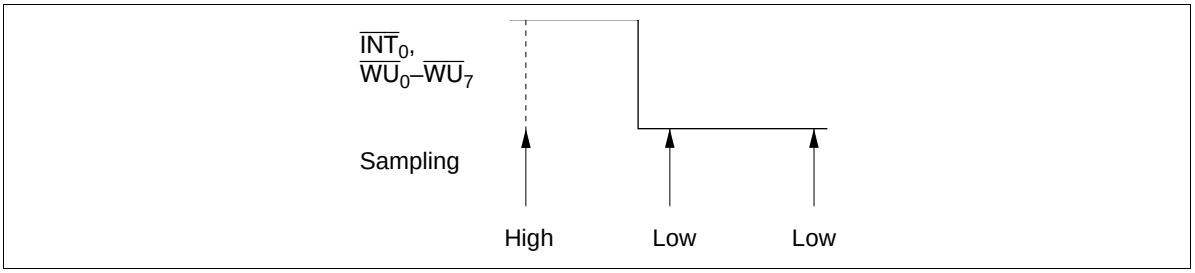



**Figure 23 Edge Detection**

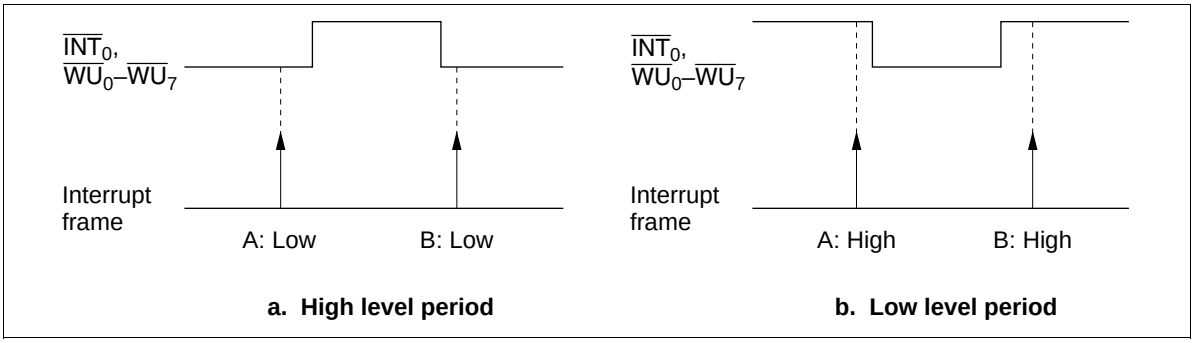


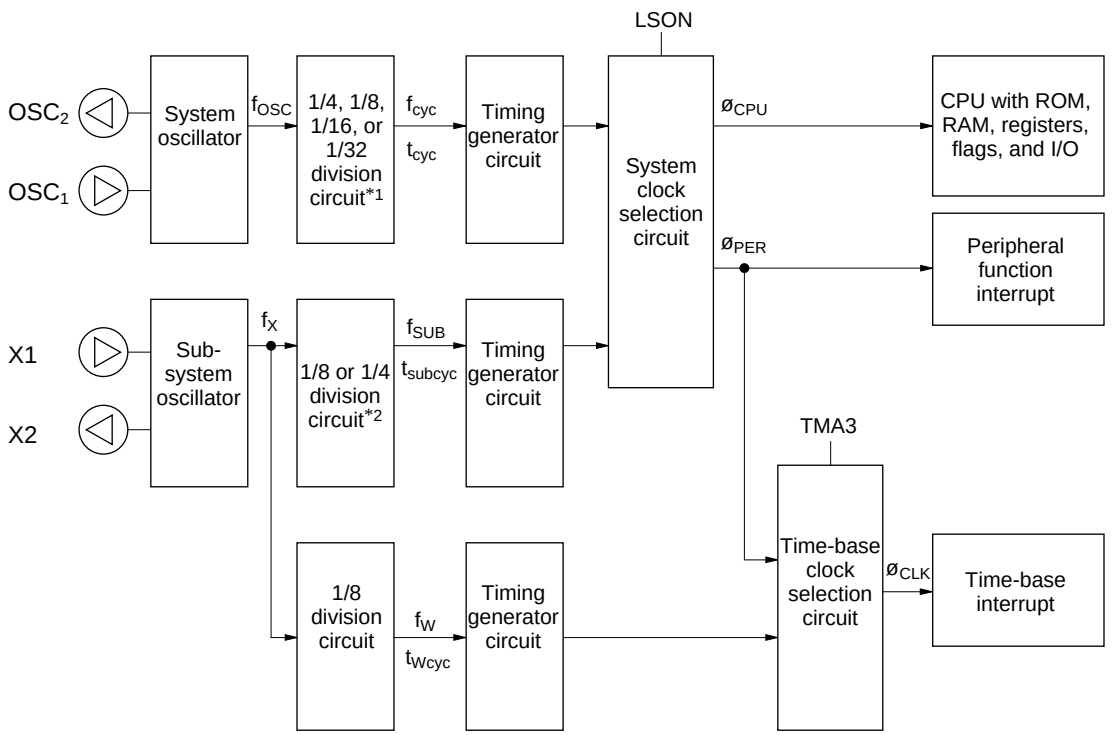

**Figure 24 Sampling Example**

## Internal Oscillator Circuit

### Clock Generation Circuit

See figure 25 for a block diagram of the clock generation circuit. A ceramic oscillator or crystal oscillator can be connected to $OSC_1$ and $OSC_2$, and a 32.768-kHz oscillator can be connected to X1 and X2 (table 22). The system oscillator can also be operated by an external clock. Bit 1 (SSR11) of system clock select register 1 (SSR1: $029) must be selected according to the frequency of the oscillator connected to $OSC_1$ and $OSC_2$(figure 26).

Note: If the system clock select register 1 (SSR1: $029) setting does not match the oscillator frequency, subsystems using the 32.768-kHz oscillation will malfunction.




Notes: 1. 1/4, 1/8, 1/16, or 1/32 division ratio can be selected by setting bits 0 and 1 of system clock select register 2 (SSR2).
2. 1/8 or 1/4 division ratio can be selected by setting bit 2 of system clock select register 1 (SSR1).

**Figure 25 Clock Generation Circuit**

**Selection of Division Ratio**

**Division Ratio of the System Clock:** 1/4, 1/8, 1/16, or 1/32 division ratio of the system clock can be selected by setting bits 0 and 1 (SSR20 and SSR21) of system clock select register 2 (SSR2: $02A). The values of SSR20 and SSR21 become valid when entering the watch mode after making the ratio selection. (However, the value of SSR2 becomes valid immediately after the selection.) Therefore, when changing the division ratio, the system clock must be stopped. There are two methods for selecting the division ratio of the system clock as follows.

- Division ratio is selected by writing to SSR20 and SSR21 in active mode. The selected values of SSR20 and SSR21 are valid before the MCU enters watch mode. The division ratio of the system clock becomes the written value when the MCU returns to the active mode from the watch mode.
- Division ratio is selected by writing to SSR20 and SSR21 in subactive mode. The division ratio of the system clock becomes the selected value when the MCU returns to active mode after entering watch mode.

Note: SSR2 is cleared in the reset and stop modes. Therefore, 1/4 division ratio of the system clock is selected when the MCU returns from stop mode after reset.

**Division Ratio of the Subsystem Clock:** 1/4 or 1/8 division ratio of the subsystem clock can be selected by setting bit 2 (SSR12) of system clock select register 1 (SSR1: $029). The value of SSR12 becomes valid immediately after the ratio selection. When the value of SSR12 is changed, the MCU must be in active mode. If the value of SSR12 is changed in subactive mode, the MCU may malfunction.

**System clock select register 1 (SSR1: $029)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | — |
| Read/Write | W | W | W | — |
| Bit name | SSR13* | SSR12 | SSR11 | Not used |

| SSR11 | System oscillation frequency selection |
|---|---|
| 0 | 1.6 to 4.0 MHz |
| 1 | 0.4 to 1.0 MHz |

| SSR12 | 32-kHz oscillation division ratio selection |
|---|---|
| 0 | $f_{sub}$ = fx/8 |
| 1 | $f_{sub}$ = fx/4 |

| SSR13 | 32-kHz oscillation stop |
|---|---|
| 0 | Oscillation operates in stop mode |
| 1 | Oscillation stops in stop mode |

Note: * SSR13 is reset to 0 only by RESET input. When $\overline{\text{STOPC}}$ is input in stop mode, SSR13 is not reset but retains its value. SSR13 is not reset in stop mode.

**Figure 26 System Clock Select Register 1 (SSR1: $029)**

**System clock select register 2 (SSR2: $02A)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | 0 | 0 |
| Read/Write | — | — | W | W |
| Bit name | Not used | Not used | SSR21 | SSR20 |

| SSR21 | SSR20 | System clock division ratio selection |
|---|---|---|
| 0 | 0 | 1/4 |
| | 1 | 1/8 |
| 1 | 0 | 1/16 |
| | 1 | 1/32 |

**Figure 27 System Clock Select Register 2 (SSR2: $02A)**

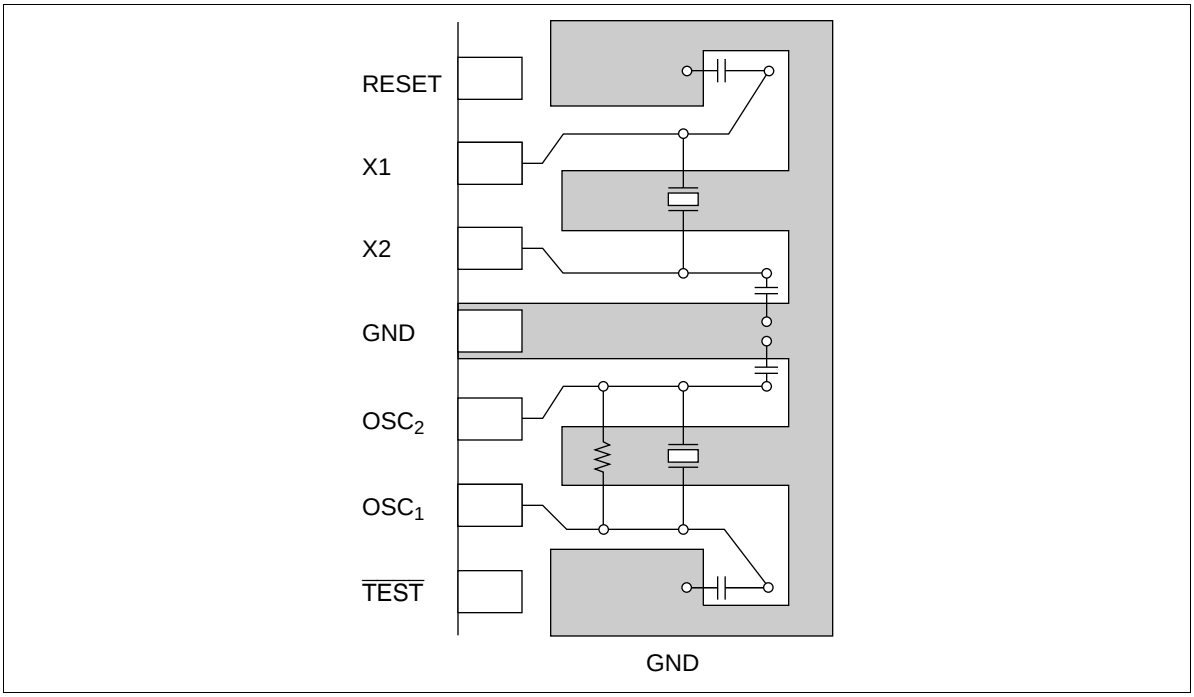
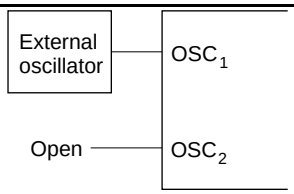
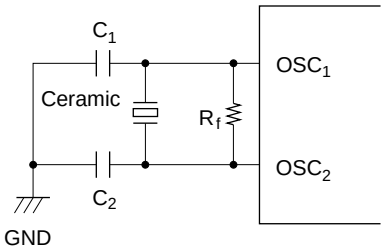
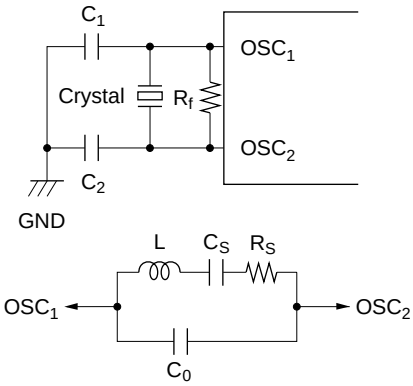
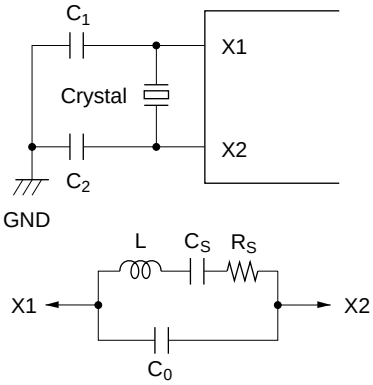



**Figure 28 Typical Layout of Crystal and Ceramic Oscillators**

**Table 22 Oscillator Circuit Examples**

| | Circuit Configuration | Circuit Constants |
|---|---|---|
| External clock operation |  | |
| Ceramic oscillator ($OSC_1$, $OSC_2$) |  | Ceramic oscillator: CSA4.00MG (Murata)<br>$R_f$ = 1 MΩ ± 20%<br>$C_1 = C_2$ = 30 pF |
| Crystal oscillator ($OSC_1$, $OSC_2$) |  | $R_f$ = 1 MΩ ± 20%<br>$C_1 = C_2$ = 10–22 pF ± 20%<br>Crystal: Equivalent to circuit shown below<br>$C_0$: 7 pF max.<br>$R_S$: 100 Ω max. |
| Crystal oscillator (X1, X2) |  | Crystal: 32.768 kHz: MX38T<br>(Nippon Denpa Kogyo)<br>$C_1 = C_2$ = 15 pF ± 5%<br>$R_S$: 14 kΩ<br>$C_0$: 1.5 pF |

Notes: 1. Since the circuit constants change depending on the crystal or ceramic resonator and stray capacitance of the board, the user should consult with the crystal or ceramic oscillator manufacturer to determine the circuit parameters.

2. Wiring among $OSC_1$, $OSC_2$, X1, X2, and elements should be as short as possible, and must not cross other wiring (figure 28).

3. If the 32.768-kHz crystal oscillator is not used, the X1 pin must be fixed to GND and X2 must be open.

## Input/Output

The MCU has 49 input/output pins ($D_0$–$D_9$, R0–R8, $R9_0$–$R9_2$) and 7 input pins ($D_{10}$, $D_{11}$, $R9_3$, RA). The features are described as follows.

- The $D_{11}$, R0, R3–R6, $R9_3$, and RA pins are multiplexed with peripheral function pins such as those for timers or the serial interface. See table 24. For these pins, the peripheral function setting is done prior to the D or R port setting. Therefore, when a peripheral function is selected for a pin, the pin function and input/output selection are automatically switched according to the setting. However, pins input to the wakeup function are not switched. Only the valid/invalid statuses of wakeup input are controlled.
- Peripheral function output pins are CMOS out-put pins. See table 23. Only the SO pin and $R4_3$ port can be set to NMOS open-drain output by software.
- In stop mode, the MCU is reset, and therefore peripheral function selection is cancelled. Input/output pins are set at high-impedance.
- Each input/output pin has a built-in pull-up MOS (figure 29), which can be individually turned on or off by software.

**Table 23 Programmable I/O Circuits**

| **MIS3 (Bit 3 of MIS)** | | **0** | | | | **1** | | | |
|---|---|---|---|---|---|---|---|---|---|
| **DCD, DCR** | | **0** | | **1** | | **0** | | **1** | |
| **PDR** | | **0** | **1** | **0** | **1** | **0** | **1** | **0** | **1** |
| CMOS buffer | PMOS | — | — | — | On | — | — | — | On |
| | NMOS | — | — | On | — | — | — | On | — |
| Pull-up MOS | | — | — | — | — | — | On | — | On |

Note: — indicates off status.

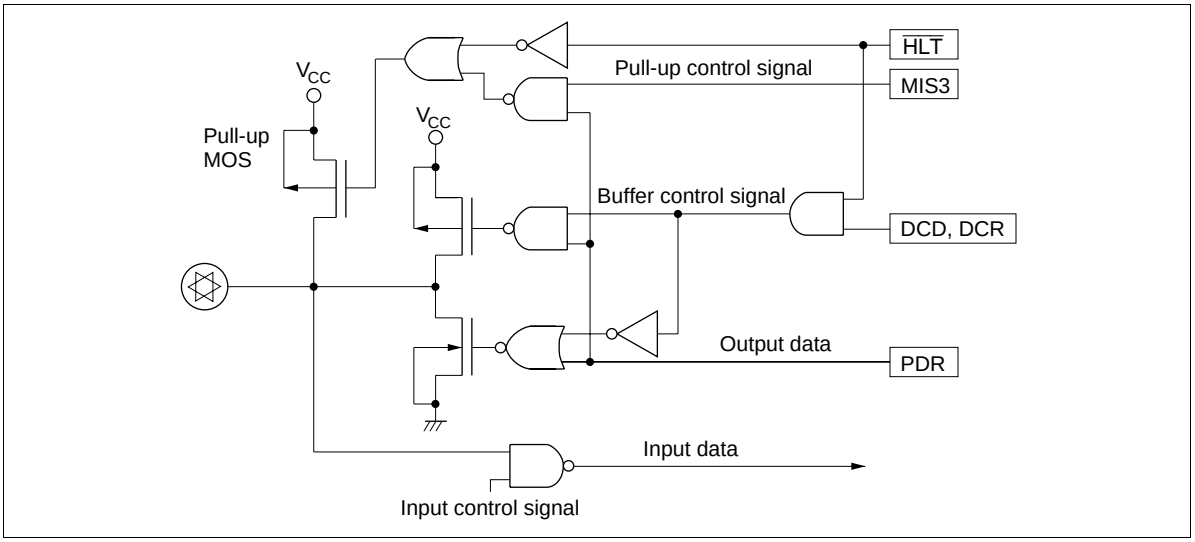
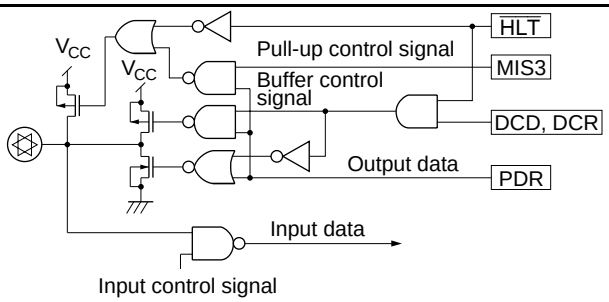
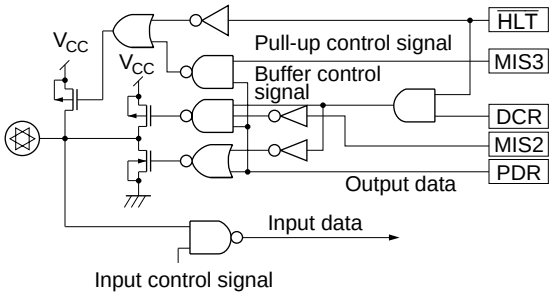
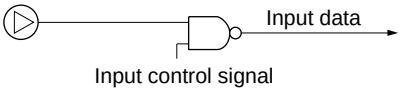
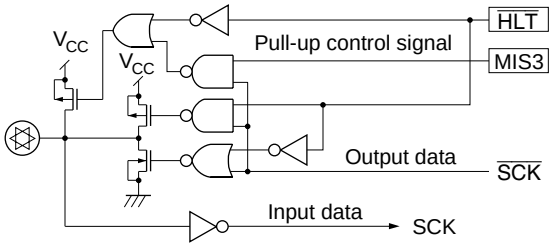
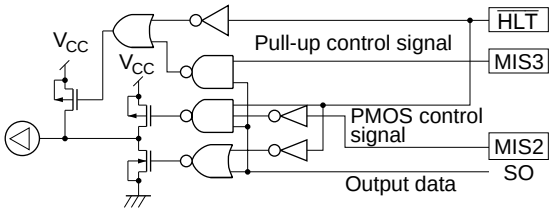
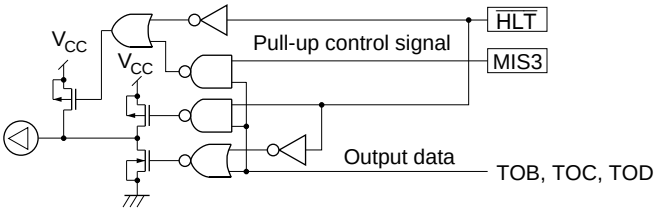
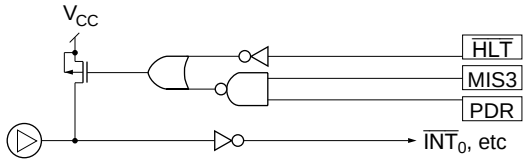
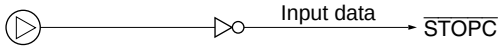



**Figure 29 I/O Buffer Configuration**

**Table 24 Circuit Configurations of I/O Pins**

| I/O Pin Type | | Circuit | Pins |
|---|---|---|---|
| Input/output pins | |  | $D_0$–$D_9$, $R0_0$–$R0_3$, $R1_0$–$R1_3$, $R2_0$–$R2_3$, $R3_0$–$R3_3$, $R4_0$–$R4_2$, $R5_0$–$R5_3$, $R6_0$–$R6_3$, $R7_0$–$R7_3$, $R8_0$–$R8_3$, $R9_0$–$R9_2$ |
| | |  | $R4_3$ |
| Input pins | |  | $D_{10}$, $D_{11}$, $R9_3$, $RA_0$–$RA_3$ |
| Peripheral function pins | Input/ output pins |  | $\overline{SCK}$ |
| | Output pins |  | SO |
| | |  | TOB, TOC, TOD |

| I/O Pin Type | | Circuit | Pins |
|---|---|---|---|
| Peripheral function pins | Input pins |  | SI, $\overline{INT}_0$, $\overline{INT}_1$, $INT_2$, $INT_3$, $\overline{WU}_0$–$\overline{WU}_7$, $\overline{EVNB}$, EVND |
| | |  | $\overline{STOPC}$ |

Notes: 1. In stop mode, the MCU is reset and peripheral function selection is cancelled. The $\overline{HLT}$ signal becomes low, and input/output pins enter high-impedance state.
2. The $\overline{HLT}$ signal is 1 in watch and subactive modes.

**D Port ($D_0$–$D_{11}$):** Consist of 10 input/output pins and 2 input pins addressed by one bit. $D_0$–$D_9$ are input/output pins, and $D_{10}$ and $D_{11}$ are input-only pins.

Pins $D_0$–$D_9$ are set by the SED and SEDD instructions, and reset by the RED and REDD instructions. Output data is stored in the port data register (PDR) for each pin. All pins $D_0$–$D_{11}$ are tested by the TD and TDD instructions.

The on/off statuses of the output buffers are controlled by D-port data control registers (DCD0–DCD2: $02C–$02E) that are mapped to memory addresses (figure 30).

Pin $D_{11}$ is multiplexed with peripheral function pin $\overline{STOPC}$. The peripheral function mode of this pin is selected by bit 2 (PMRC2) of port mode register C (PMRC: $025) (figure 35).

**R Ports (R0–RA):** 39 input/output pins and 5 input pins addressed in 4-bit units. Data is input to these ports by the LAR and LBR instructions, and output from them by the LRA and LRB instructions. Output data is stored in the port data register (PDR) for each pin. The on/off statuses of the output buffers of the R ports are controlled by R-port data control registers (DCR0–DCR9: $030–$039) that are mapped to memory addresses (figure 30).

**Data control register (DCD0 to DCD2: $02C to $02E)**
**(DCR0 to DCR9: $030 to $039)**

**DCD0, DCD1**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | DCD03, DCD13 | DCD02, DCD12 | DCD01, DCD11 | DCD00, DCD10 |

**DCD2**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | 0 | 0 |
| Read/Write | — | — | W | W |
| Bit name | Not used | Not used | DCD21 | DCD20 |

**DCR0 to DCR8**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | DCR03–DCR83 | DCR02–DCR82 | DCR01–DCR81 | DCR00–DCR80 |

**DCR9**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | 0 | 0 | 0 |
| Read/Write | — | W | W | W |
| Bit name | Not used | DCR92 | DCR91 | DCR90 |

| All Bits | CMOS Buffer On/Off Selection |
|---|---|
| 0 | Off (high-impedance) |
| 1 | On |

Correspondence between ports and DCD/DCR bits

| Register Name | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|
| DCD0 | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
| DCD1 | $D_7$ | $D_6$ | $D_5$ | $D_4$ |
| DCD2 | — | — | $D_9$ | $D_8$ |
| DCR0 | $R0_3$ | $R0_2$ | $R0_1$ | $R0_0$ |
| DCR1 | $R1_3$ | $R1_2$ | $R1_1$ | $R1_0$ |
| DCR2 | $R2_3$ | $R2_2$ | $R2_1$ | $R2_0$ |
| DCR3 | $R3_3$ | $R3_2$ | $R3_1$ | $R3_0$ |
| DCR4 | $R4_3$ | $R4_2$ | $R4_1$ | $R4_0$ |
| DCR5 | $R5_3$ | $R5_2$ | $R5_1$ | $R5_0$ |
| DCR6 | $R6_3$ | $R6_2$ | $R6_1$ | $R6_0$ |
| DCR7 | $R7_3$ | $R7_2$ | $R7_1$ | $R7_0$ |
| DCR8 | $R8_3$ | $R8_2$ | $R8_1$ | $R8_0$ |
| DCR9 | — | $R9_2$ | $R9_1$ | $R9_0$ |

**Figure 30 Data Control Registers (DCD, DCR)**

Pins $R0_0$–$R0_3$ are multiplexed with peripheral pins $\overline{INT}_0$–$INT_3$, respectively. The peripheral function modes of these pins are selected by bits 0–3 (PMRB0–PMRB3) of port mode register B (PMRB: $024) (figure 31).

**Port mode register B (PMRB: $024)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | PMRB3 | PMRB2 | PMRB1 | PMRB0 |

| PMRB0 | $R0_0/\overline{INT}_0$ mode selection |
|---|---|
| 0 | $R0_0$ |
| 1 | $\overline{INT}_0$ |

| PMRB1 | $R0_1/\overline{INT}_1$ mode selection |
|---|---|
| 0 | $R0_1$ |
| 1 | $\overline{INT}_1$ |

| PMRB2 | $R0_2/INT_2$ mode selection |
|---|---|
| 0 | $R0_2$ |
| 1 | $INT_2$ |

| PMRB3 | $R0_3/INT_3$ mode selection |
|---|---|
| 0 | $R0_3$ |
| 1 | $INT_3$ |

**Figure 31 Port Mode Register B (PMRB)**

Pins $R3_0$–$R3_2$ are multiplexed with peripheral pins TOB, TOC, and TOD, respectively. The peripheral function modes of these pins are selected by bits 0 and 1 (TMB20, TMB21) of timer mode register B2 (TMB2: $013), bits 0–2 (TMC20–TMC22) of timer mode register C2 (TMC2: $014), and bits 0–3 (TMD20–TMD23) of timer mode register D2 (TMD2: $015) (figures 32, 33, and 34).

**Timer mode register B2 (TMB2: $013)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | 0 | 0 |
| Read/Write | — | — | R/W | R/W |
| Bit name | Not used | Not used | TMB21 | TMB20 |

| TMB21 | TMB20 | $R3_0$/TOB mode selection | |
|---|---|---|---|
| 0 | 0 | $R3_0$ | $R3_0$ port |
| | 1 | TOB | Toggle output |
| 1 | 0 | TOB | 0 output |
| | 1 | TOB | 1 output |

**Figure 32 Timer Mode Register B2 (TMB2)**

**Timer mode register C2 (TMC2: $014)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W |
| Bit name | Not used | TMC22 | TMC21 | TMC20 |

| TMC22 | TMC21 | TMC20 | $R3_1$/TOC mode selection | |
|---|---|---|---|---|
| 0 | 0 | 0 | $R3_1$ | $R3_1$ port |
| | | 1 | TOC | Toggle output |
| | 1 | 0 | TOC | 0 output |
| | | 1 | TOC | 1 output |
| 1 | 0 | 0 | TOC | Not used |
| | | 1 | TOC | Not used |
| | 1 | 0 | TOC | Not used |
| | | 1 | TOC | PWM output |

**Figure 33 Timer Mode Register C2 (TMC2)**

**Timer mode register D2 (TMD2: $015)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W |
| Bit name | TMD23 | TMD22 | TMD21 | TMD20 |

| TMD23 | TMD22 | TMD21 | TMD20 | $R3_2$/TOD mode selection | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $R3_2$ | $R3_2$ port |
| | | | 1 | TOD | Toggle output |
| | | 1 | 0 | TOD | 0 output |
| | | | 1 | TOD | 1 output |
| | 1 | 0 | 0 | TOD | Not used |
| | | | 1 | TOD | Not used |
| | | 1 | 0 | TOD | Not used |
| | | | 1 | TOD | PWM output |
| 1 | Don't care | Don't care | Don't care | $R3_2$ | Input capture ($R3_2$ port) |

**Figure 34 Timer Mode Register D2 (TMD2)**

Pins $R3_3$ and $R4_0$ are multiplexed with peripheral pins $\overline{EVNB}$ and EVND, respectively. The peripheral function modes of these pins are selected by bits 0 and 1 (PMRC0, PMRC1) of port mode register C (PMRC: $025) (figure 35).

**Port mode register C (PMRC: $025)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | 0 | 0 | 0 |
| Read/Write | — | W | W | W |
| Bit name | Not used | PMRC2* | PMRC1 | PMRC0 |

| PMRC0 | $R3_3/\overline{EVNB}$ mode selection |
|---|---|
| 0 | $R3_3$ |
| 1 | $\overline{EVNB}$ |

| PMRC1 | $R4_0$/EVND mode selection |
|---|---|
| 0 | $R4_0$ |
| 1 | EVND |

| PMRC2 | $D_{11}/\overline{STOPC}$ mode selection |
|---|---|
| 0 | $D_{11}$ |
| 1 | $\overline{STOPC}$ |

Note: * PMRC2 is reset to 0 only by RESET input. When $\overline{STOPC}$ is input in stop mode, PMRC2 is not reset but retains its value.

**Figure 35 Port Mode Register C (PMRC)**

Pins $R4_1$–$R4_3$ are multiplexed with peripheral pins $\overline{SCK}$, SI, and SO, respectively. The peripheral function modes of these pins are selected by bit 3 (SMRA3) of serial mode register A (SMRA: $005), and bits 0 and 1 (PMRA0, PMRA1) port mode register A (PMRA: $004) (figures 36 and 37).

**Port mode register A (PMRA: $004)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | 0 | 0 |
| Read/Write | — | — | W | W |
| Bit name | Not used | Not used | PMRA1 | PMRA0 |

| PMRA0 | $R4_3$/SO mode selection |
|---|---|
| 0 | $R4_3$ |
| 1 | SO |

| PMRA1 | $R4_2$/SI mode selection |
|---|---|
| 0 | $R4_2$ |
| 1 | SI |

**Figure 36 Port Mode Register A (PMRA)**

**Serial mode register A (SMRA: $005)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | SMRA3 | SMRA2 | SMRA1 | SMRA0 |

| SMRA3 | $R4_1$/$\overline{SCK}$ mode selection |
|---|---|
| 0 | $R4_1$ port |
| 1 | $\overline{SCK}$ |

| SMRA2 | SMRA1 | SMRA0 | $\overline{SCK}$ | Clock source | Prescaler division ratio |
|---|---|---|---|---|---|
| 0 | 0 | 0 | Output | Prescaler | ÷2048 |
| | | 1 | Output | Prescaler | ÷512 |
| | 1 | 0 | Output | Prescaler | ÷128 |
| | | 1 | Output | Prescaler | ÷32 |
| 1 | 0 | 0 | Output | Prescaler | ÷8 |
| | | 1 | Output | Prescaler | ÷2 |
| | 1 | 0 | Output | System clock | — |
| | | 1 | Input | External clock | — |

**Figure 37 Serial Mode Register A (SMRA)**

Ports R5 and R6 are multiplexed with pins $\overline{WU}_0$–$\overline{WU}_7$. The wakeup modes of these pins can be selected by the wakeup select register (WSR: $018). Even if wakeup input is valid, the R port functions normally (figure 38).

**Wakeup select register (WSR: $018)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | WSR3 | WSR2 | WSR1 | WSR0 |

| WSR0 | $\overline{WU}_0$ to $\overline{WU}_3$ control |
|---|---|
| 0 | Invalid |
| 1 | Valid |

| WSR1 | $\overline{WU}_4$ to $\overline{WU}_5$ control |
|---|---|
| 0 | Invalid |
| 1 | Valid |

| WSR2 | $\overline{WU}_6$ control |
|---|---|
| 0 | Invalid |
| 1 | Valid |

| WSR3 | $\overline{WU}_7$ control |
|---|---|
| 0 | Invalid |
| 1 | Valid |

**Figure 38 Wakeup Select Register (WSR)**

**Pull-Up MOS Transistor Control:** A program-controlled pull-up MOS transistor is provided for each input/output pin other than input-only pins $D_{10}$, $D_{11}$, $R9_3$, and $RA_0$–$RA_3$. The on/off status of all these transistors is controlled by bit 3 (MIS3) of the miscellaneous register (MIS: $00C), and the on/off status of an individual transistor can also be controlled by the port data register (PDR) of the corresponding pin—enabling on/off control of that pin alone (table 23 and figure 39).

The on/off status of each transistor and the peripheral function mode of each pin can be set independently.

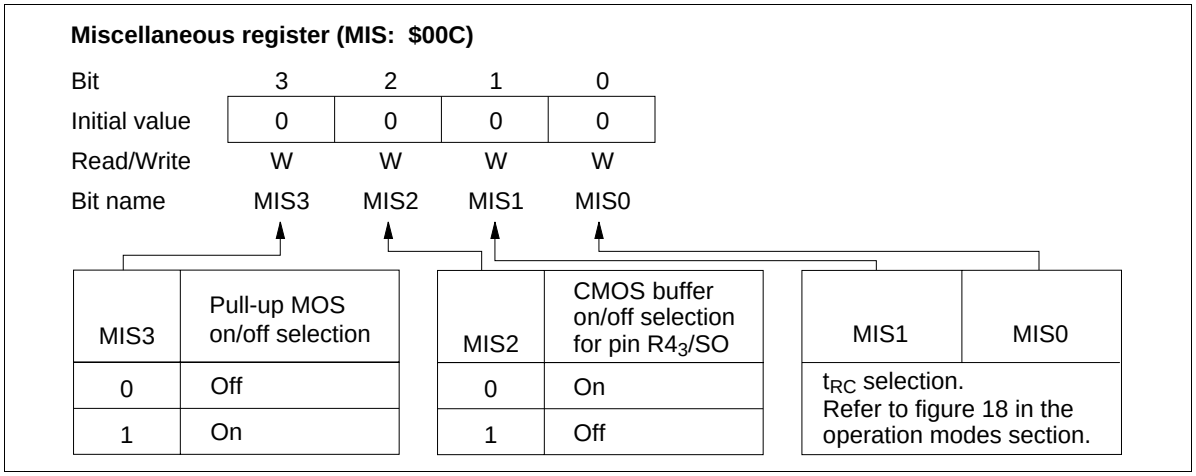



**Figure 39 Miscellaneous Register (MIS)**

**How to Deal with Unused I/O Pins:** I/O pins that are not needed by the user system (those that remain floating) must be connected to $V_{CC}$ to prevent LSI malfunctions due to noise. These pins must either be pulled up to $V_{CC}$ by their pull-up MOS transistors or by resistors of about 100 kΩ.

## Prescalers

The MCU has two prescalers, S and W. See table 25 and figure 40.

Both the timers A–D input clocks except external events and the serial transmit clock except the external clock are selected from the prescaler outputs, depending on corresponding mode registers.

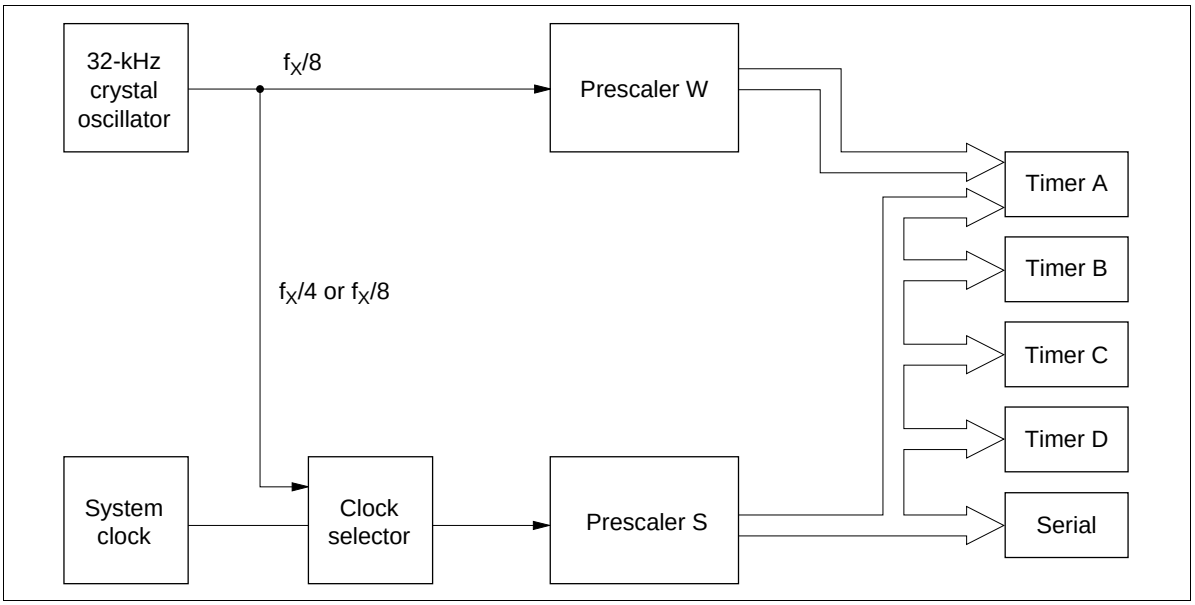



**Figure 40 Prescaler Output Supply**

### Prescaler Operation

**Prescaler S:** 11-bit counter that inputs a system clock signal. After being reset to $000 by MCU reset, prescaler S divides the system clock. Prescaler S keeps counting, except in watch and stop modes and at MCU reset.

**Prescaler W:** Five-bit counter that inputs the X1 input clock signal (32-kHz crystal oscillation) divided. After being reset to $00 by MCU reset, prescaler W divides the input clock. Prescaler W can be reset by software.

**Table 25 Prescaler Operating Conditions**

| Prescaler | Input Clock | Reset Conditions | Stop Conditions |
|---|---|---|---|
| Prescaler S | System clock (in active and standby mode), Subsystem clock (in subactive mode) | MCU reset | MCU reset, stop mode, watch mode |
| Prescaler W | 32-kHz crystal oscillation | MCU reset, software | MCU reset, stop mode |

## Timers

The MCU has four timer/counters (A to D).

— Timer A: Free-running timer
— Timer B: Multifunction timer
— Timer C: Multifunction timer
— Timer D: Multifunction timer

Timer A is an 8-bit free-running timer. Timers B–D are 8-bit multifunction timers (table 26). The operating modes are selected by software.

**Table 26 Timer Functions**

| Functions | | Timer A | Timer B | Timer C | Timer D |
|---|---|---|---|---|---|
| Clock source | Prescaler S | Available | Available | Available | Available |
| | Prescaler W | Available | — | — | — |
| | External event | — | Available | — | Available |
| Timer functions | Free-running | Available | Available | Available | Available |
| | Time-base | Available | — | — | — |
| | Event counter | — | Available | — | Available |
| | Reload | — | Available | Available | Available |
| | Watchdog | — | — | Available | — |
| | Input capture | — | — | — | Available |
| Timer outputs | Toggle | — | Available | Available | Available |
| | 0 output | — | Available | Available | Available |
| | 1 output | — | Available | Available | Available |
| | PWM | — | — | Available | Available |

Note: — means not available.

### Timer A

**Timer A Functions:** Timer A (figure 41) has the following functions.

- Free-running timer
- Clock time-base

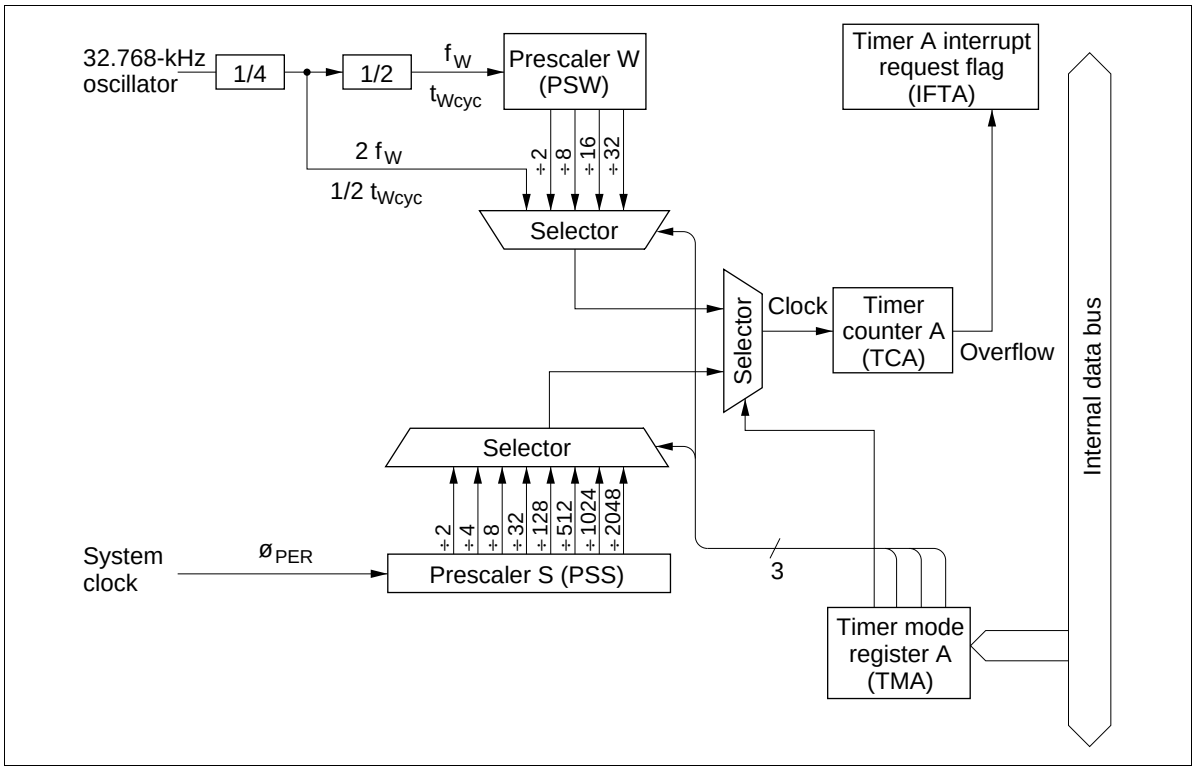



**Figure 41 Block Diagram of Timer A**

**Timer A Operations:**

- Free-running timer operation: The input clock for timer A is selected by timer mode register A (TMA: $008).

  Timer A is reset to $00 by MCU reset and incremented at each input clock. If an input clock is applied to timer A after it has reached $FF, an overflow is generated, and timer A is reset to $00. The overflow sets the timer A interrupt request flag (IFTA: $002, bit 0). Timer A continues to be incremented after reset to $00, and therefore it generates regular interrupts every 256 clocks.

- Clock time-base operation: Timer A is used as a clock time-base by setting bit 3 (TMA3) of timer mode register A (TMA: $008) to 1. The prescaler W output is applied to timer A, and timer A generates interrupts at the correct timing based on the 32.768-kHz crystal oscillation. In this case, prescaler W and timer A can be reset to $00 by software.

**Registers for Timer A Operation:** Timer A operating modes are set by the following registers.

- Timer mode register A (TMA: $008): Four-bit write-only register that selects timer A's operating mode and input clock source (figure 42).

**Timer mode register A (TMA: $008)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | TMA3 | TMA2 | TMA1 | TMA0 |

| TMA3 | TMA2 | TMA1 | TMA0 | Source prescaler | Input clock frequency | Operating mode |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | PSS | $2048t_{cyc}$ | Timer A mode |
| | | | 1 | PSS | $1024t_{cyc}$ | |
| | | 1 | 0 | PSS | $512t_{cyc}$ | |
| | | | 1 | PSS | $128t_{cyc}$ | |
| | 1 | 0 | 0 | PSS | $32t_{cyc}$ | |
| | | | 1 | PSS | $8t_{cyc}$ | |
| | | 1 | 0 | PSS | $4t_{cyc}$ | |
| | | | 1 | PSS | $2t_{cyc}$ | |
| 1 | 0 | 0 | 0 | PSW | $32t_{Wcyc}$ | Time-base mode |
| | | | 1 | PSW | $16t_{Wcyc}$ | |
| | | 1 | 0 | PSW | $8t_{Wcyc}$ | |
| | | | 1 | PSW | $2t_{Wcyc}$ | |
| | 1 | 0 | 0 | PSW | $1/2t_{Wcyc}$ | |
| | | | 1 | Not used | | |
| | | 1 | Don't care | PSW and TCA reset | | |

Note: 1. $t_{Wcyc}$ = 244.14 µs (when a 32.768-kHz crystal oscillator is used)
2. Timer counter overflow output period (seconds) = input clock period (seconds) × 256.
3. The division ratio must not be modified during time-base mode operation, otherwise an overflow cycle error will occur.

**Figure 42 Timer Mode Register A (TMA)**

### Timer B

**Timer B Functions:** Timer B (figure 43) has the following functions.

- Free-running/reload timer
- External event counter
- Timer output operation (toggle, 0, and 1 outputs)

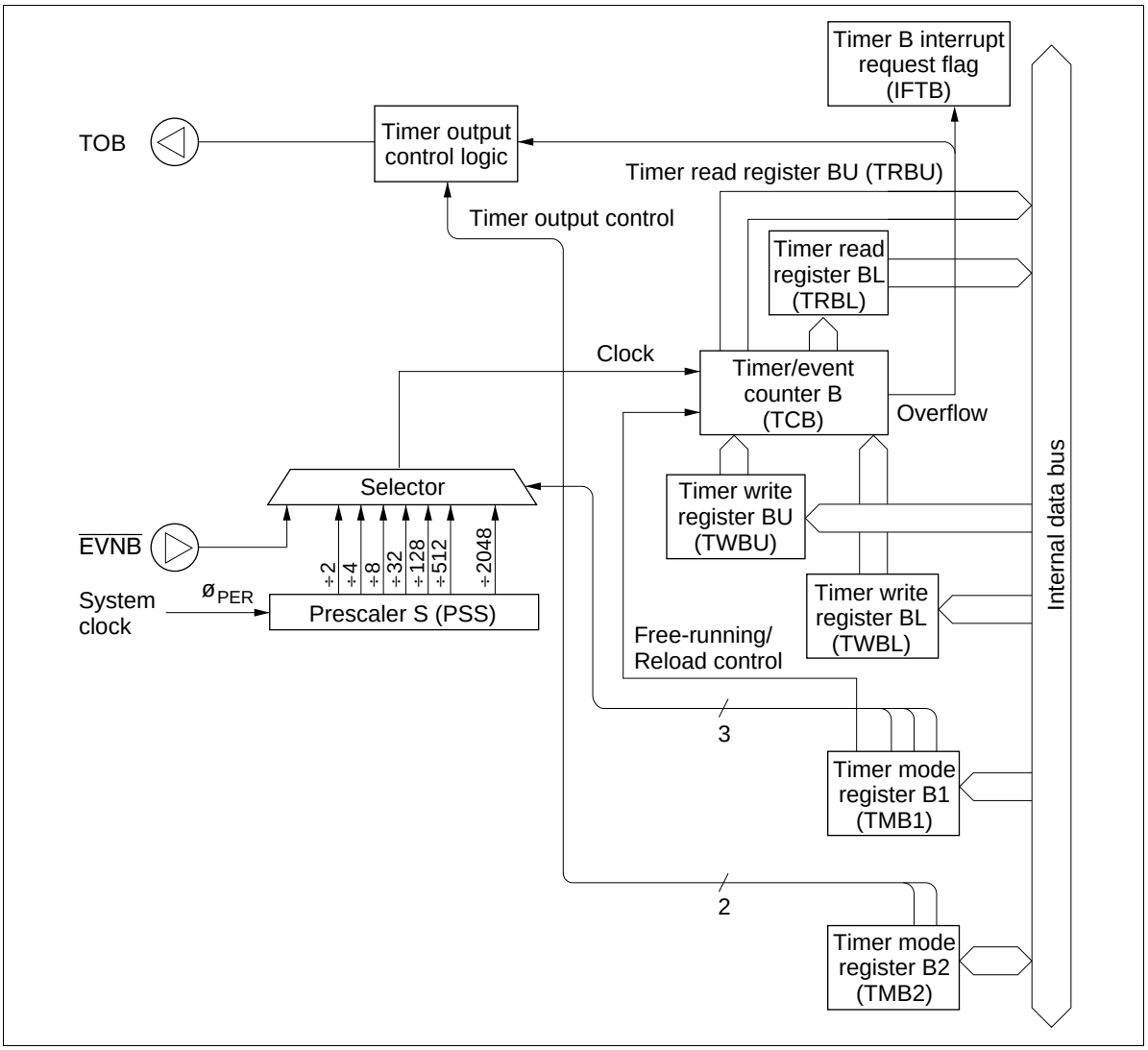



**Figure 43 Block Diagram of Timer B**

**Timer B Operations:**

- Free-running/reload timer operation: The free-running/reload operation, input clock source, and prescaler division ratio are selected by timer mode register B1 (TMB1: $009).

  Timer B is initialized to the value set in timer write register B (TWBL: $00A, TWBU: $00B) by software and incremented by one at each clock input. If an input clock is applied to timer B after it has reached $FF, an overflow is generated. In this case, if the reload timer function is enabled, timer B is initialized to its initial value set in timer write register B; if the free-running timer function is enabled, the timer is initialized to $00 and then incremented again.

  The overflow sets the timer B interrupt request flag (IFTB: $002, bit 2). IFTB can be reset by software or MCU reset. Refer to figure 3 and table 1 for details.

- External event counter operation: Timer B is used as an external event counter by selecting external event input as the input clock source. In this case, pin $R3_3/\overline{EVNB}$ must be set to $\overline{EVNB}$ by port mode register C (PMRC: $025).

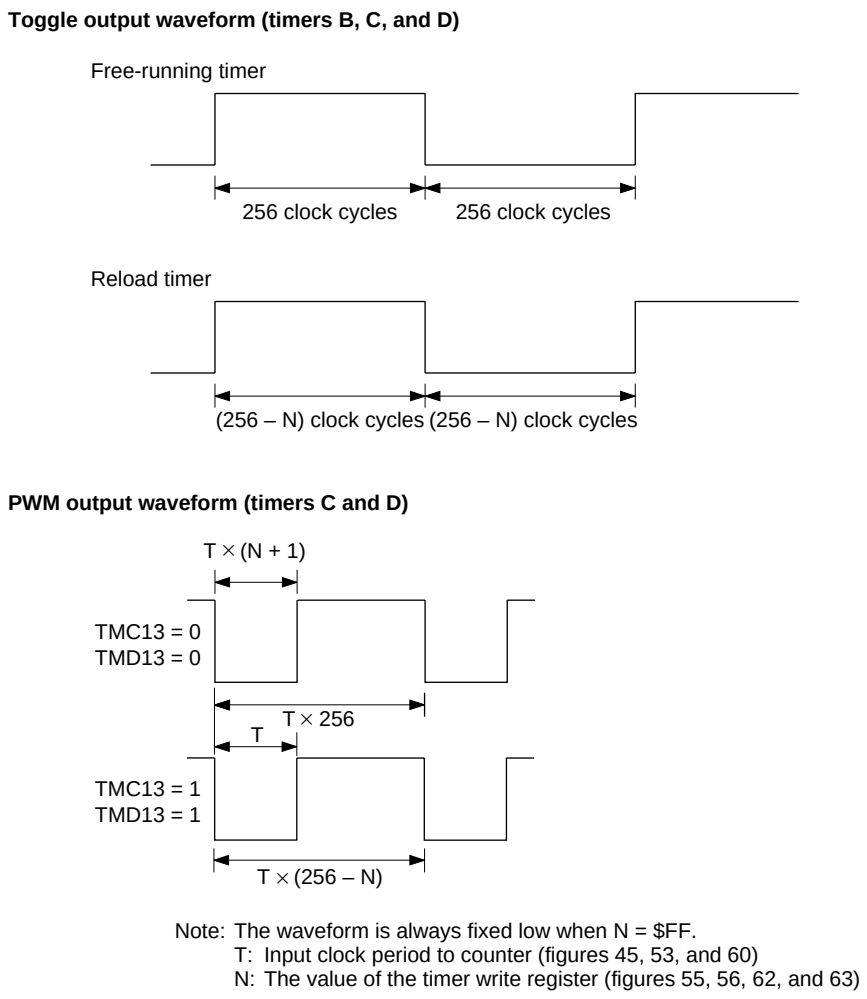
  Timer B is incremented by one at each falling edge of signals input to pin $\overline{EVNB}$. The other operations are basically the same as the free-running/ reload timer operation.

- Timer output operation: The following three output modes can be selected for timer B by setting timer mode register B2 (TMB2: $013).

  — Toggle
  — 0 output
  — 1 output

  By selecting the timer output mode, pin $R3_0$/TOB is set to TOB. The output from TOB is reset low by MCU reset.

  — Toggle output: When toggle output mode is selected, the output level is inverted if a clock is input after timer B has reached $FF. By using this function and reload timer function, clock signals can be output at a required frequency for a buzzer. Refer to figure 44 for the output waveform.
  — 0 output: When 0 output mode is selected, the output level is pulled low if a clock is input after timer B has reached $FF. Note that this function must be used only when the output level is high.
  — 1 output: When 1 output mode is selected, the output level is set high if a clock is input after timer B has reached $FF. Note that this function must be used only when the output level is low.




Note: The waveform is always fixed low when N = $FF.
T: Input clock period to counter (figures 45, 53, and 60)
N: The value of the timer write register (figures 55, 56, 62, and 63)

**Figure 44 Timer Output Waveform**

**Registers for Timer B Operation:** By using the following registers, timer B operation modes are selected and the timer B count is read and written.

— Timer mode register B1 (TMB1: $009)
— Timer mode register B2 (TMB2: $013)
— Timer write register B (TWBL: $00A, TWBU: $00B)
— Timer read register B (TRBL: $00A, TRBU: $00B)
— Port mode register C (PMRC: $025)

- Timer mode register B1 (TMB1: $009): Four-bit write-only register that selects the free-running/reload timer function, input clock source, and the prescaler division ratio (figure 45). It is reset to $0 by MCU reset.

The mode change of this register is valid from the second instruction execution cycle after the execution of the previous timer mode register B1 write instruction. Setting timer B's initialization by writing to timer write register B (TWBL: $00A, TWBU: $00B) must be done after a mode change becomes valid.

**Timer mode register B1 (TMB1: $009)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | TMB13 | TMB12 | TMB11 | TMB10 |

| TMB13 | Free-running/reload timer selection |
|---|---|
| 0 | Free-running timer |
| 1 | Reload timer |

| TMB12 | TMB11 | TMB10 | Input clock period and input clock source |
|---|---|---|---|
| 0 | 0 | 0 | $2048t_{cyc}$ |
| | | 1 | $512t_{cyc}$ |
| | 1 | 0 | $128t_{cyc}$ |
| | | 1 | $32t_{cyc}$ |
| 1 | 0 | 0 | $8t_{cyc}$ |
| | | 1 | $4t_{cyc}$ |
| | 1 | 0 | $2t_{cyc}$ |
| | | 1 | $R3_3/\overline{EVNB}$ (external event input) |

**Figure 45 Timer Mode Register B1 (TMB1)**

- Timer mode register B2 (TMB2: $013): Two-bit read/write register that selects the timer B output mode (figure 46). It is reset to $0 by MCU reset.

**Timer mode register B2 (TMB2: $013)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | 0 | 0 |
| Read/Write | — | — | R/W | R/W |
| Bit name | Not used | Not used | TMB21 | TMB20 |

| TMB21 | TMB20 | $R3_0$/TOB mode selection | |
|---|---|---|---|
| 0 | 0 | $R3_0$ | $R3_0$ port |
| | 1 | TOB | Toggle output |
| 1 | 0 | TOB | 0 output |
| | 1 | TOB | 1 output |

**Figure 46 Timer Mode Register B2 (TMB2)**

- Timer write register B (TWBL: $00A, TWBU: $00B): Write-only register consisting of a lower digit (TWBL) and an upper digit (TWBU) (figures 47 and 48). The lower digit is reset to $0 by MCU reset, but the upper digit value is undefined.

  Timer B is initialized by writing to timer write register B (TWBL: $00A, TWBU: $00B). In this case, the lower digit (TWBL) must be written to first, but writing only to the lower digit does not change the timer B value. Timer B is initialized to the value in timer write register B at the same time the upper digit (TWBU) is written to. When timer write register B is written to again and if the lower digit value needs no change, writing only to the upper digit initializes timer B.

**Timer write register B (lower digit) (TWBL: $00A)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | TWBL3 | TWBL2 | TWBL1 | TWBL0 |

**Figure 47 Timer Write Register B Lower Digit (TWBL)**

**Timer write register B (upper digit) (TWBU: $00B)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | W | W | W | W |
| Bit name | TWBU3 | TWBU2 | TWBU1 | TWBU0 |

**Figure 48 Timer Write Register B Upper Digit (TWBU)**

- Timer read register B (TRBL: $00A, TRBU: $00B): Read-only register consisting of a lower digit (TRBL) and an upper digit (TRBU) that holds the count of the timer B upper digit.

  The upper digit (TRBU) must be read first, which will result in the count of the timer B upper digit to be obtained and the count of the timer B lower digit to be latched to the lower digit (TRBL). Then by reading TRBL, the count of timer B can be obtained when TRBU is read.

**Timer read register B (lower digit) (TRBL: $00A)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R | R | R | R |
| Bit name | TRBL3 | TRBL2 | TRBL1 | TRBL0 |

**Figure 49 Timer Read Register B Lower Digit (TRBL)**

**Timer read register B (upper digit) (TRBU: $00B)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R | R | R | R |
| Bit name | TRBU3 | TRBU2 | TRBU1 | TRBU0 |

**Figure 50 Timer Read Register B Upper Digit (TRBU)**

- Port mode register C (PMRC: $025): Write-only register that selects the $R3_3/\overline{EVNB}$ pin function (figure 51). It is reset to $0 by MCU reset.

**Port mode register C (PMRC: $025)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | 0 | 0 | 0 |
| Read/Write | — | W | W | W |
| Bit name | Not used | PMRC2 | PMRC1 | PMRC0 |

| PMRC0 | $R3_3/\overline{EVNB}$ mode selection |
|---|---|
| 0 | $R3_3$ |
| 1 | $\overline{EVNB}$ |

| PMRC1 | $R4_0$/EVND mode selection |
|---|---|
| 0 | $R4_0$ |
| 1 | EVND |

| PMRC2 | $D_{11}/\overline{STOPC}$ mode selection |
|---|---|
| 0 | $D_{11}$ |
| 1 | $\overline{STOPC}$ |

**Figure 51 Port Mode Register C (PMRC)**

**Timer C**

**Timer C Functions:** Timer C (figure 52) has the following functions.

— Free-running/reload timer
— Watchdog timer
— Timer output operation (toggle, 0, 1, and PWM outputs)

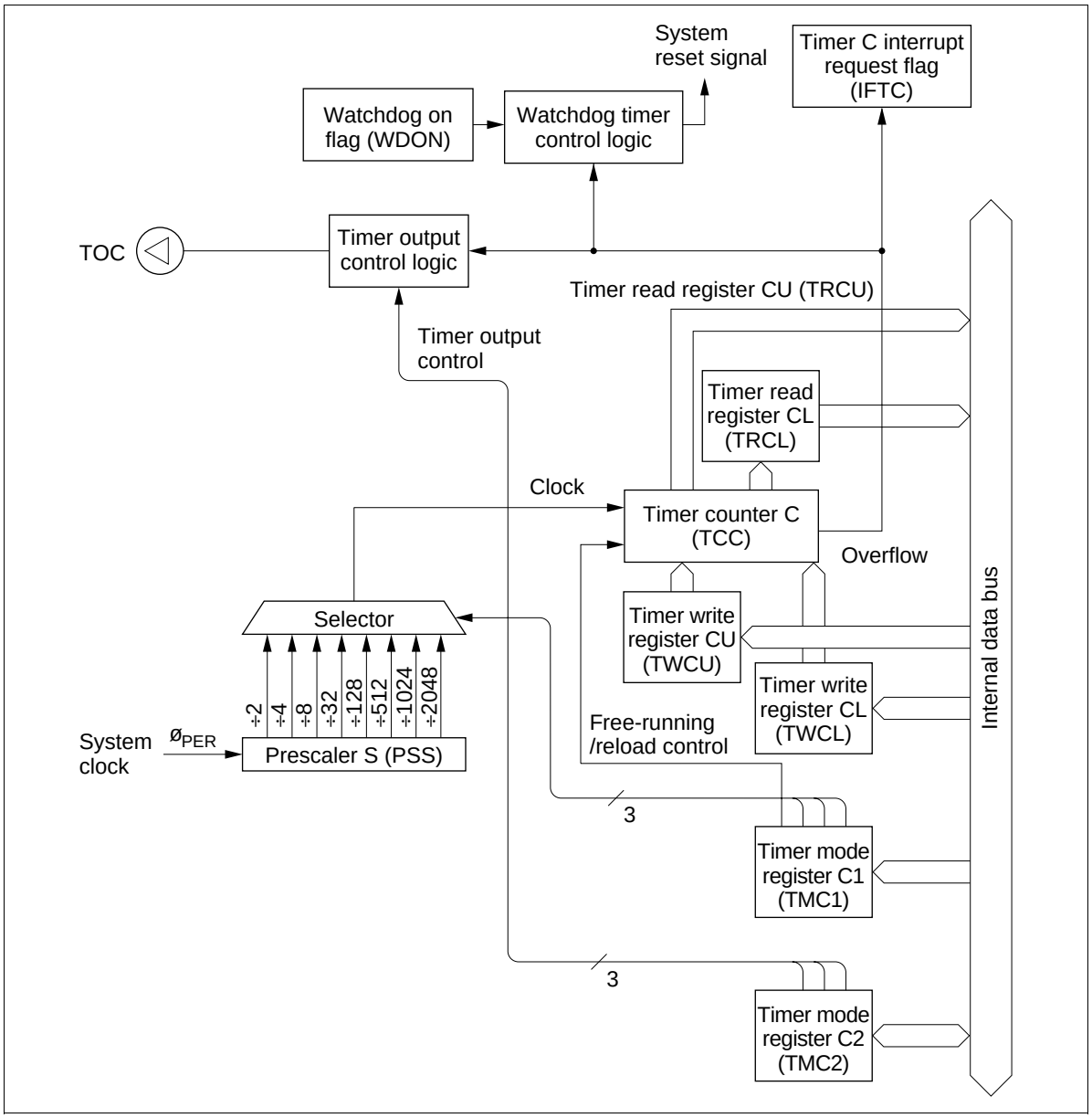



**Figure 52 Block Diagram of Timer C**

**Timer C Operations:**

- Free-running/reload timer operation: The free-running/reload operation, input clock source, and prescaler division ratio are selected by timer mode register C1 (TMC1: $00D).

  Timer C is initialized to the value set in timer write register C (TWCL: $00E, TWCU: $00F) by software and incremented by one at each clock input. If an input clock is applied to timer C after it has reached $FF, an overflow is generated. In this case, if the reload timer function is enabled, timer C is initialized to its initial value set in timer write register C; if the free-running timer function is enabled, the timer is initialized to $00 and then incremented again.

  The overflow sets the timer C interrupt request flag (IFTC: $003, bit 0). IFTC can be reset by software or MCU reset. Refer to figure 3 and table 1 for details.

- Watchdog timer operation: Timer C is used as a watchdog timer for detecting out-of-control program routines by setting the watchdog on flag (WDON: $020, bit 1) to 1. If a program routine runs out of control and an overflow is generated, the MCU is reset. Program run can be controlled by initializing timer C by software before it reaches $FF.

- Timer output operation: The following four output modes can be selected for timer C by setting timer mode register C2 (TMC2: $014).

  — Toggle
  — 0 output
  — 1 output
  — PWM output

  By selecting the timer output mode, pin $R3_1$/TOC is set to TOC. The output from TOC is reset low by MCU reset.

  — Toggle output: The operation is basically the same as that of timer-B's toggle output.
  — 0 output: The operation is basically the same as that of timer-B's 0 output.
  — 1 output: The operation is basically the same as that of timer-B's 1 output.
  — PWM output (figure 44): When PWM output mode is selected, timer C provides the variable-duty pulse output function. The output waveform differs depending on the contents of timer mode register C1 (TMC1: $00D) and timer write register C (TWCL: $00E, TWCU: $00F).

**Registers for Timer C Operation:** By using the following registers, timer C operation modes are selected and the timer C count is read and written.

— Timer mode register C1 (TMC1: $00D)
— Timer mode register C2 (TMC2: $014)
— Timer write register C (TWCL: $00E, TWCU: $00F)
— Timer read register C (TRCL: $00E, TRCU: $00F)

- Timer mode register C1 (TMC1: $00D): Four-bit write-only register that selects the free-running/ reload timer function, input clock source, and prescaler division ratio (figure 53). It is reset to $0 by MCU reset.

The mode change of this register is valid from the second instruction execution cycle after the execution of the previous timer mode register C1 write instruction. Setting timer C's initialization by writing to timer write register C (TWCL: $00E, TWCU: $00F) must be done after a mode change becomes valid.

**Timer mode register C1 (TMC1: $00D)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | TMC13 | TMC12 | TMC11 | TMC10 |

| TMC13 | Free-running/reload timer selection |
|---|---|
| 0 | Free-running timer |
| 1 | Reload timer |

| TMC12 | TMC11 | TMC10 | Input clock period |
|---|---|---|---|
| 0 | 0 | 0 | $2048t_{cyc}$ |
| | | 1 | $1024t_{cyc}$ |
| | 1 | 0 | $512t_{cyc}$ |
| | | 1 | $128t_{cyc}$ |
| 1 | 0 | 0 | $32t_{cyc}$ |
| | | 1 | $8t_{cyc}$ |
| | 1 | 0 | $4t_{cyc}$ |
| | | 1 | $2t_{cyc}$ |

**Figure 53 Timer Mode Register C1 (TMC1)**

- Timer mode register C2 (TMC2: $014): Three-bit read/write register that selects the timer C output mode (figure 54). It is reset to $0 by MCU reset.

**Timer mode register C2 (TMC2: $014)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | 0 | 0 | 0 |
| Read/Write | — | R/W | R/W | R/W |
| Bit name | Not used | TMC22 | TMC21 | TMC20 |

| TMC22 | TMC21 | TMC20 | $R3_1$/TOC mode selection | |
|---|---|---|---|---|
| 0 | 0 | 0 | $R3_1$ | $R3_1$ port |
| | | 1 | TOC | Toggle output |
| | 1 | 0 | TOC | 0 output |
| | | 1 | TOC | 1 output |
| 1 | 0 | 0 | TOC | Not used |
| | | 1 | | |
| | 1 | 0 | | |
| | | 1 | TOC | PWM output |

**Figure 54 Timer Mode Register C2 (TMC2)**

- Timer write register C (TWCL: $00E, TWCU: $00F): Write-only register consisting of a lower digit (TWCL) and an upper digit (TWCU) (figures 55 and 56). The operation of timer write register C is basically the same as that of timer write register B (TWBL: $00A, TWBU: $00B).

**Timer write register C (lower digit) (TWCL: $00E)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | TWCL3 | TWCL2 | TWCL1 | TWCL0 |

**Figure 55 Timer Write Register C Lower Digit (TWCL)**

**Timer write register C (upper digit) (TWCU: $00F)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | W | W | W | W |
| Bit name | TWCU3 | TWCU2 | TWCU1 | TWCU0 |

**Figure 56 Timer Write Register C Upper Digit (TWCU)**

- Timer read register C (TRCL: $00E, TRCU: $00F): Read-only register consisting of a lower digit (TRCL) and an upper digit (TRCU) that holds the count of the timer C upper digit(figures 57 and 58). The operation of timer read register C is basically the same as that of timer read register B (TRBL: $00A, TRBU:$00B).

**Timer read register C (lower digit) (TRCL: $00E)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R | R | R | R |
| Bit name | TRCL3 | TRCL2 | TRCL1 | TRCL0 |

**Figure 57 Timer Read Register C Lower Digit (TRCL)**

**Timer read register C (upper digit) (TRCU: $00F)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R | R | R | R |
| Bit name | TRCU3 | TRCU2 | TRCU1 | TRCU0 |

**Figure 58 Timer Read Register C Upper Digit(TRCU)**

**Timer D**

**Timer D Functions:** Timer D (figures 59 (A) and (B)) has the following functions.

— Free-running/reload timer
— External event counter
— Timer output operation (toggle, 0, 1, and PWM outputs)
— Input capture timer

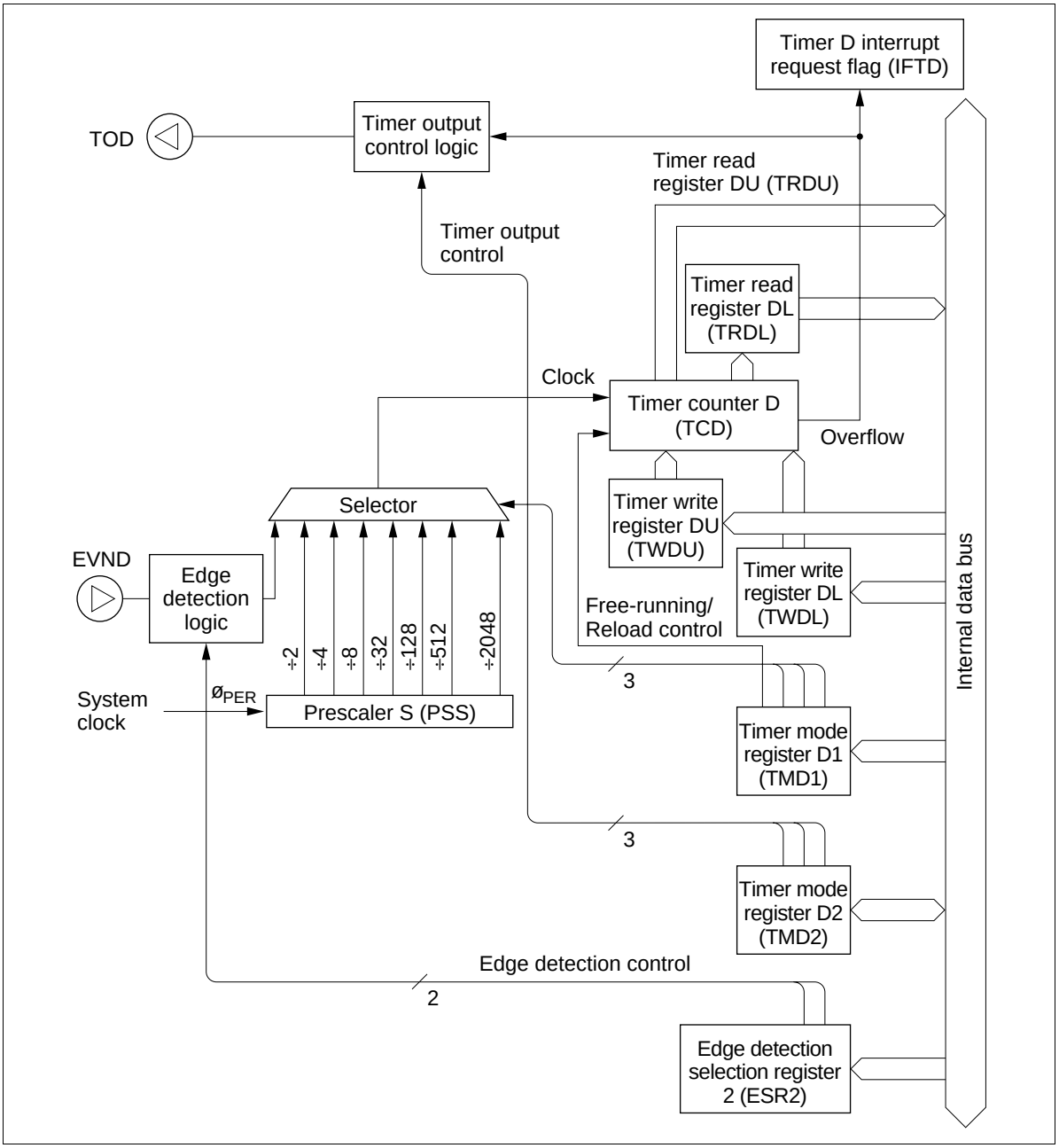



**Figure 59(A) Block Diagram of Timer D (Free-Running/Reload Timer)**

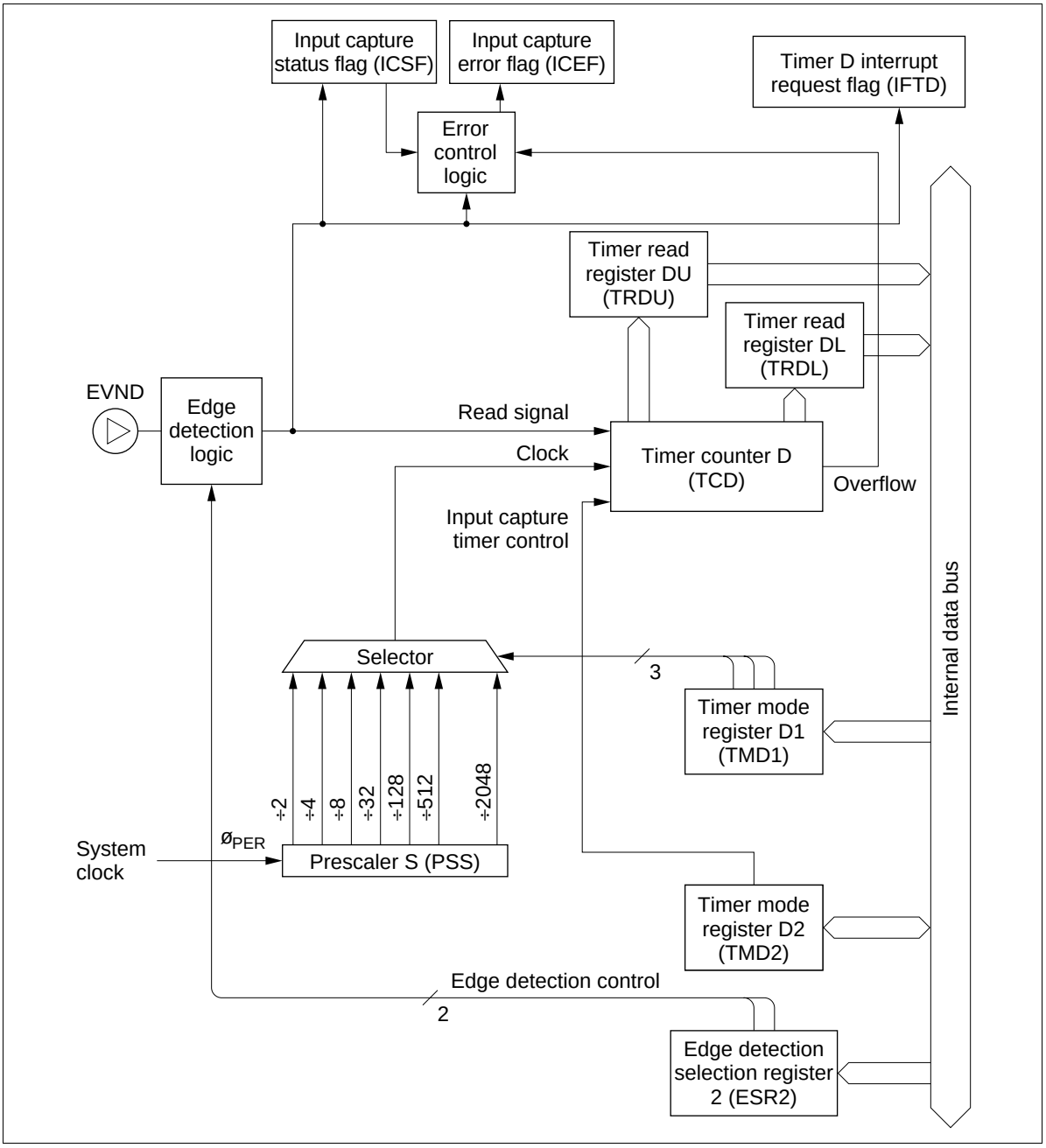



**Figure 59(B) Block Diagram of Timer D (Input Capture Timer)**

**Timer D Operations:**

- Free-running/reload timer operation: The free-running/reload operation, input clock source, and prescaler division ratio are selected by timer mode register D1 (TMD1: $010).

  Timer D is initialized to the value set in timer write register D (TWDL: $011, TWDU: $012) by software and incremented by one at each clock input. If an input clock is applied to timer D after it has reached $FF, an overflow is generated. In this case, if the reload timer function is enabled, timer D is initialized to its initial value set in timer write register D; if the free-running timer function is enabled, the timer is initialized to $00 and then incremented again.

  The overflow sets the timer D interrupt request flag (IFTD: $001, bit 2). IFTD can be reset by software or MCU reset. Refer to figure 3 and table 1 for details.

- External event counter operation: Timer D is used as an external event counter by selecting the external event input as an input clock source. In this case, pin $R4_0$/EVND must be set to EVND by port mode register C (PMRC: $025).

  Either falling or rising edge, or both falling and rising edges of input signals can be selected as the external event detection edge by detection edge select register 2 (ESR2: $027). When both rising and falling edges detection is selected, the time between the falling edge and rising edge of input signals must be $2t_{cyc}$ or longer.

  Timer D is incremented by one at each detection edge selected by detection edge select register 2 (ESR2: $027). The other operation is basically the same as the free-running/reload timer operation.

- Timer output operation: The following four output modes can be selected for timer D by setting timer mode register D2 (TMD2: $015).

  — Toggle
  — 0 output
  — 1 output
  — PWM output

  By selecting the timer output mode, pin $R3_2$/TOD is set to TOD. The output from TOD is reset low by MCU reset.

  — Toggle output: The operation is basically the same as that of timer-B's toggle output.
  — 0 output: The operation is basically the same as that of timer-B's 0 output.
  — 1 output: The operation is basically the same as that of timer-B's 1 output.
  — PWM output: The operation is basically the same as that of timer-C's PWM output.

- Input capture timer operation: The input capture timer counts the clock cycles between trigger edges input to pin EVND.

  Either falling or rising edge, or both falling and rising edges of input signals can be selected as the trigger input edge by detection edge select register 2 (ESR2: $027).

When a trigger edge is input to EVND, the count of timer D is written to timer read register D (TRDL: $011, TRDU: $012), and the timer D interrupt request flag (IFTD: $001, bit 2) and the input capture status flag (ICSF: $021, bit 0) are set. Timer D is reset to $00, and then incremented again. While ICSF is set, if a trigger input edge is applied to timer D, or if timer D generates an overflow, the input capture error flag (ICEF: $021, bit 1) is set. ICSF and ICEF can be reset to 0 by MCU reset or by writing 0.

By selecting the input capture operation, pin $R3_2$/TOD is set to $R3_2$ and timer D is reset to $00.

**Registers for Timer D Operation:** By using the following registers, timer D operation modes are selected and the timer D count is read and written.

— Timer mode register D1 (TMD1: $010)
— Timer mode register D2 (TMD2: $015)
— Timer write register D (TWDL: $011, TWDU: $012)
— Timer read register D (TRDL: $011, TRDU: $012)
— Port mode register C (PMRC: $025)
— Detection edge select register 2 (ESR2: $027)

- Timer mode register D1 (TMD1: $010): Four-bit write-only register that selects the free-running/reload timer function, input clock source, and the prescaler division ratio (figure 60). It is reset to $0 by MCU reset.

  The mode change of this register is valid from the second instruction execution cycle after the execution of the previous timer mode register D1 (TMD1: $010) write instruction. Setting timer D's initialization by writing to timer write register D (TWDL: $011, TWDU: $012) must be done after a mode change becomes valid.

  When selecting the input capture timer operation, select the internal clock as the input clock source.

**Timer mode register D1 (TMD1: $010)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | TMD13 | TMD12 | TMD11 | TMD10 |

| TMD13 | Free-running/reload timer selection |
|---|---|
| 0 | Free-running timer |
| 1 | Reload timer |

| TMD12 | TMD11 | TMD10 | Input clock period and input clock source |
|---|---|---|---|
| 0 | 0 | 0 | $2048t_{cyc}$ |
| | | 1 | $512t_{cyc}$ |
| | 1 | 0 | $128t_{cyc}$ |
| | | 1 | $32t_{cyc}$ |
| 1 | 0 | 0 | $8t_{cyc}$ |
| | | 1 | $4t_{cyc}$ |
| | 1 | 0 | $2t_{cyc}$ |
| | | 1 | $R4_0$/EVND (external event input) |

**Figure 60 Timer Mode Register D1 (TMD1)**

- Timer mode register D2 (TMD2: $015): Four-bit read/write register that selects the timer D output mode and input capture operation (figure 61). It is reset to $0 by MCU reset.

**Timer mode register D2 (TMD2: $015)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W |
| Bit name | TMD23 | TMD22 | TMD21 | TMD20 |

| TMD23 | TMD22 | TMD21 | TMD20 | $R3_2$/TOD mode selection | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $R3_2$ | $R3_2$ port |
| | | | 1 | TOD | Toggle output |
| | | 1 | 0 | TOD | 0 output |
| | | | 1 | TOD | 1 output |
| | 1 | 0 | 0 | TOD | Not used |
| | | | 1 | | |
| | | 1 | 0 | | |
| | | | 1 | TOD | PWM output |
| 1 | Don't care | Don't care | Don't care | $R3_2$ | Input capture ($R3_2$ port) |

**Figure 61 Timer Mode Register D2(TMD2)**

- Timer write register D (TWDL: $011, TWDU: $012): Write-only register consisting of a lower digit (TWDL) and an upper digit (TWDU) (figures 62 and 63). The operation of timer write register D is basically the same as that of timer write register B (TWBL: $00A, TWBU: $00B).

**Timer write register D (lower digit) (TWDL: $011)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | TWDL3 | TWDL2 | TWDL1 | TWDL0 |

**Figure 62 Timer Write Register D Lower Digit (TWDL)**

**Timer write register D (upper digit) (TWDU: $012)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | W | W | W | W |
| Bit name | TWDU3 | TWDU2 | TWDU1 | TWDU0 |

**Figure 63 Timer Write Register D Upper Digit (TWDU)**

- Timer read register D (TRDL: $011, TRDU: $012): Read-only register consisting of a lower digit (TRDL) and an upper digit (TRDU) (figures 64 and 65). The operation of timer read register D is basically the same as that of timer read register B (TRBL: $00A, TRBU: $00B).

  When the input capture timer operation is selected and if the count of timer D is read after a trigger is input, either the lower or upper digit can be read first.

**Timer read register D (lower digit) (TRDL: $011)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R | R | R | R |
| Bit name | TRDL3 | TRDL2 | TRDL1 | TRDL0 |

**Figure 64 Timer Read Register D Lower Digit (TRDL)**

**Timer read register D (upper digit) (TRDU: $012)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R | R | R | R |
| Bit name | TRDU3 | TRDU2 | TRDU1 | TRDU0 |

**Figure 65 Timer Read Register D Upper Digit (TRDU)**

- Port mode register C (PMRC: $025): Write-only register that selects $R4_0$/EVND pin function (figure 51). It is reset to $0 by MCU reset.

- Detection edge select register 2 (ESR2: $027): Write-only register that selects the detection edge of signals input to pin EVND (figure 66). It is reset to $0 by MCU reset.

**Detection edge register 2 (ESR2: $027)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | — | — |
| Read/Write | W | W | — | — |
| Bit name | ESR23 | ESR22 | Not used | Not used |

| ESR23 | ESR22 | EVND detection edge |
|---|---|---|
| 0 | 0 | No detection |
| | 1 | Falling-edge detection |
| 1 | 0 | Rising-edge detection |
| | 1 | Double-edge detection* |

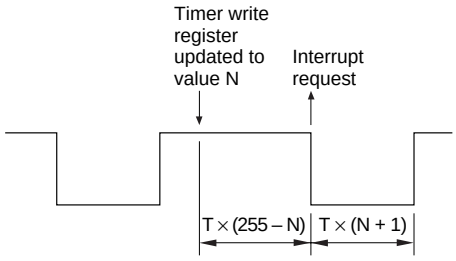
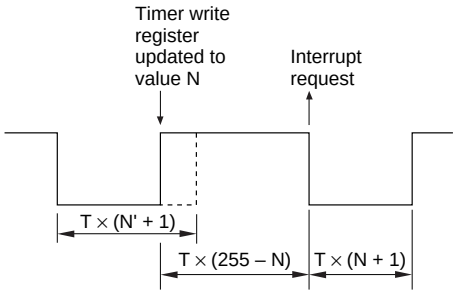
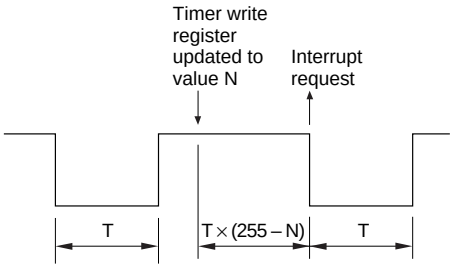
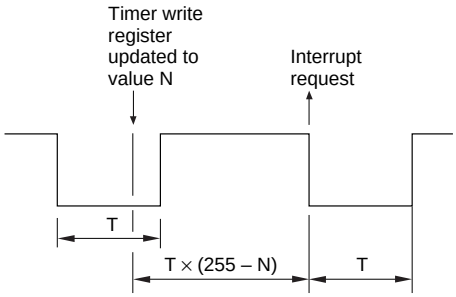
Note: * Both falling and rising edges are detected.

**Figure 66 Detection Edge Select Register 2 (ESR2)**

**Notes on Use**

When using the timer output as PWM output, note the following point. From the update of the timer write register until the occurrence of the overflow interrupt, the PWM output differs from the period and duty settings, as shown in table 27. The PWM output should therefore not be used until after the overflow interrupt following the update of the timer write register. After the overflow, the PWM output will have the set period and duty cycle.

**Table 27 PWM Output following Update of Timer Write Register**

| | PWM Output | |
|---|---|---|
| **Mode** | **Timer Write Register is Updated during High PWM Output** | **Timer Write Register is Updated during Low PWM Output** |
| Free running |  |  |
| Reload |  |  |

## Serial Interface

The MCU has a serial interface (figure 67). The serial interface serially transfers or receives 8-bit data, and includes the following features.

- Multiple transmit clock sources
  - External clock
  - Internal prescaler output clock
  - System clock
- Output level control in idle states

Five registers, an octal counter, and a multiplexer are also configured for the serial interface as follows.

- Serial data register (SRL: $006, SRU: $007)
- Serial mode register A (SMRA: $005)
- Serial mode register B (SMRB: $028)
- Port mode register A (PMRA: $004)
- Miscellaneous register (MIS: $00C)
- Octal counter (OC)
- Selector

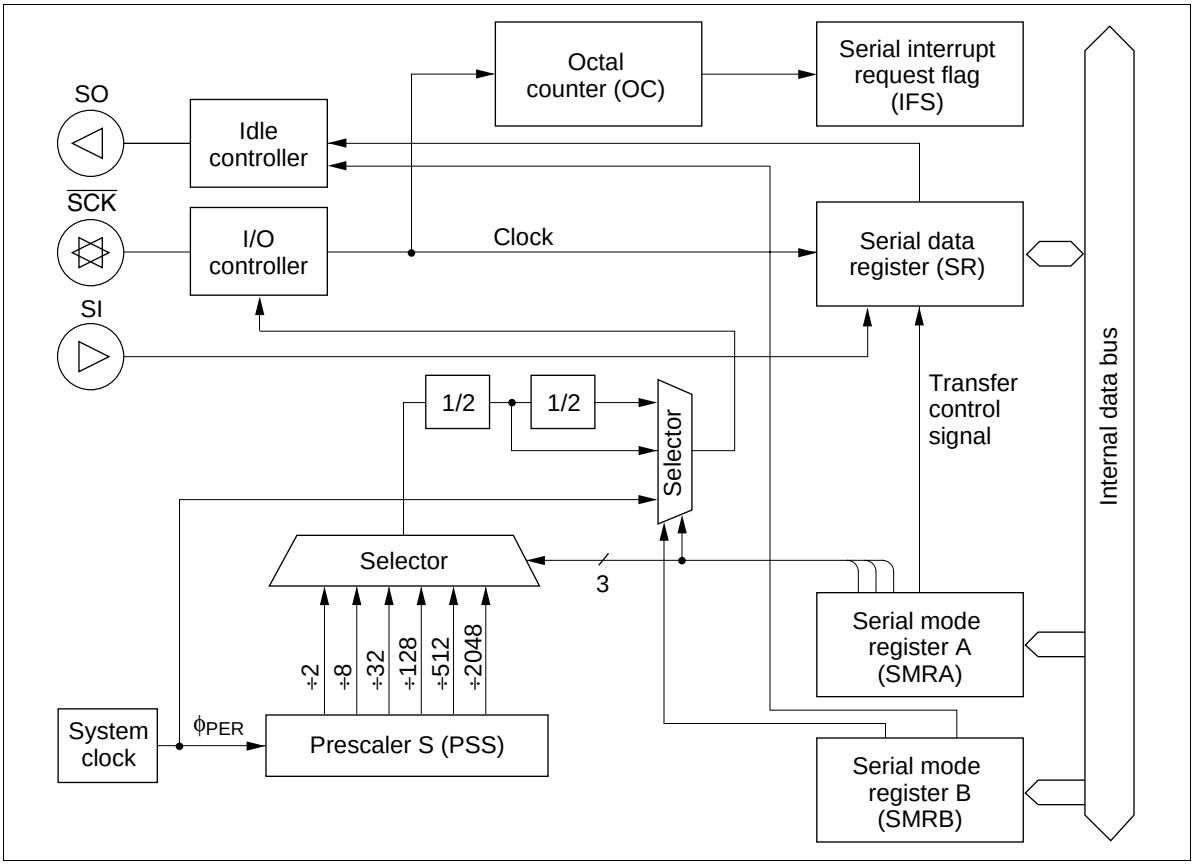



**Figure 67 Serial Interface Block Diagram**

**Serial Interface Operation**

**Selecting and Changing the Operating Mode:** To select an operating mode, use one of these combinations of port mode register A (PMRA: $004) and serial mode register A (SMRA: $005) settings (table 28); to change the operating mode of the serial interface, always initialize the serial interface internally by writing data to serial mode register A. Note that the serial interface is initialized by writing data to serial mode register A. Refer to the following section, Registers for Serial Interface, for details.

**Pin Setting:** The $R4_1/\overline{SCK}$ pin is controlled by writing data to serial mode register A (SMRA: $005). Pins $R4_2$/SI and $R4_3$/SO are controlled by writing data to port mode register A (PMRA: $004). Refer to the following section, Registers for Serial Interface, for details.

**Transmit Clock Source Setting:** The transmit clock source of the serial interface is set by writing data to serial mode register A (SMRA: $005) and serial mode register B (SMRB: $028). Refer to the following section, Registers for Serial Interface, for details.

**Data Setting:** Transmit data of the serial interface is set by writing data to the serial data register (SRL: $006, SRU: $007). Receive data of the serial interface is obtained by reading the contents of the serial data register. The serial data is shifted by each serial interface transmit clock and is input from or output to an external system.

The output level of the SO pins is undefined until the first data of each serial interface is output after MCU reset, or until the output level control in idle states is performed.

**Transfer Control:** The serial interface is activated by the STS instruction. The octal counter is reset to 000 by the STS instruction and is incremented at the rising edge of the transmit clock for the serial interface. When the eighth transmit clock signal is input or when serial transmission/reception is discontinued, the octal counter is reset to 000, the serial interrupt request flag (IFS: $023, bit 0) for serial interface is set, and the transfer stops.

When the prescaler output is selected as the transmit clock of the serial interface, the transmit clock frequency is selected as $4t_{cyc}$ to $8192t_{cyc}$ by setting bits 0 to 2 (SMRA0–SMRA2) of serial mode register A (SMRA: $005) and bit 0 (SMRB0) of serial mode register B (SMRB: $028) (table 29).

**Table 28 Serial Interface Operating Mode**

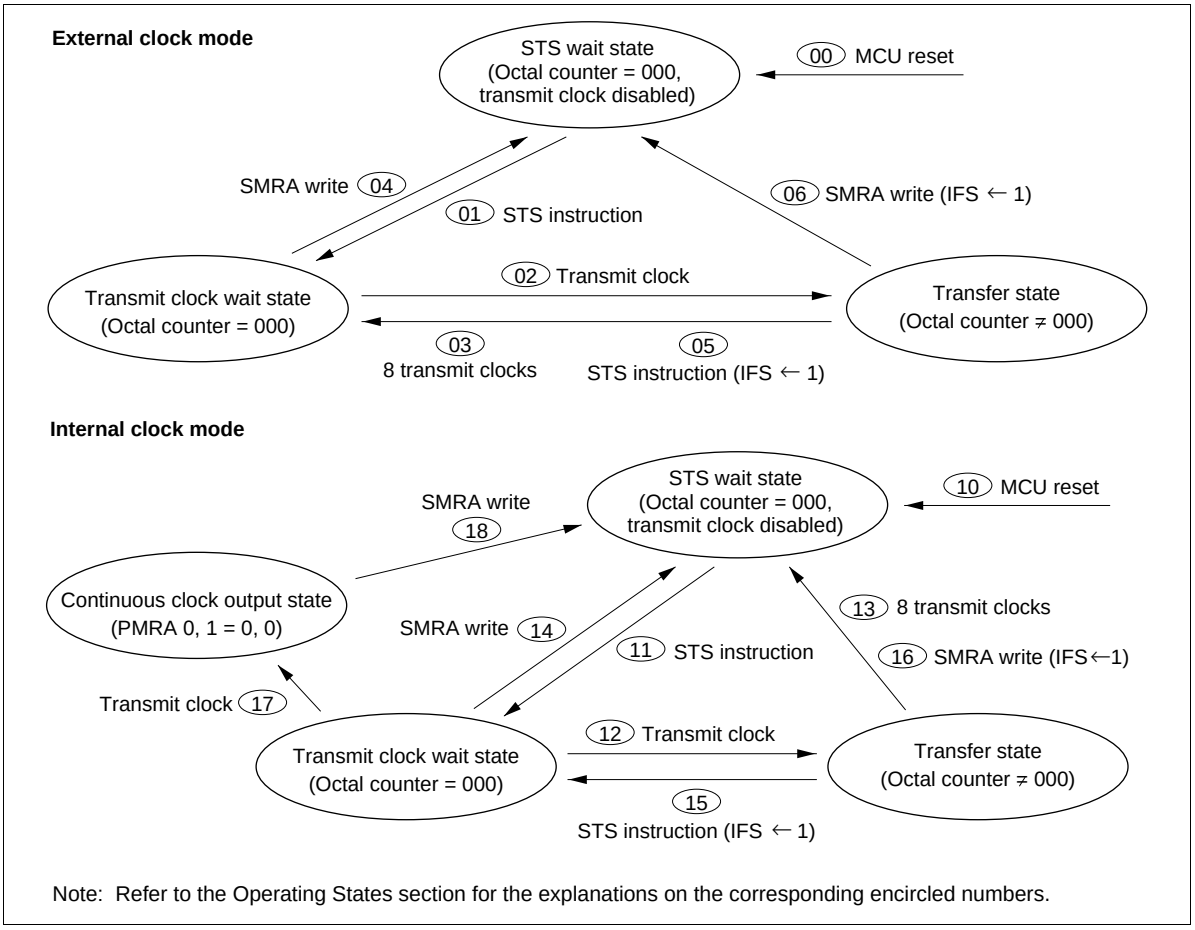
| SMRA | PMRA | | |
|---|---|---|---|
| **Bit 3** | **Bit 1** | **Bit 0** | **Operating Mode** |
| 1 | 0 | 0 | Continuous clock output mode |
| | | 1 | Transmit mode |
| | 1 | 0 | Receive mode |
| | | 1 | Transmit/receive mode |

**Table 29 Serial Transmit Clock (Prescaler Output)**

| SMRB | SMRA | | | | |
|---|---|---|---|---|---|
| **Bit 0** | **Bit 2** | **Bit 1** | **Bit 0** | **Prescaler Division Ratio** | **Transmit Clock Frequency** |
| 0 | 0 | 0 | 0 | ÷ 2048 | $4096t_{cyc}$ |
| | | | 1 | ÷ 512 | $1024t_{cyc}$ |
| | | 1 | 0 | ÷ 128 | $256t_{cyc}$ |
| | | | 1 | ÷ 32 | $64t_{cyc}$ |
| | 1 | 0 | 0 | ÷ 8 | $16t_{cyc}$ |
| | | | 1 | ÷ 2 | $4t_{cyc}$ |
| 1 | 0 | 0 | 0 | ÷ 4096 | $8192t_{cyc}$ |
| | | | 1 | ÷ 1024 | $2048t_{cyc}$ |
| | | 1 | 0 | ÷ 256 | $512t_{cyc}$ |
| | | | 1 | ÷ 64 | $128t_{cyc}$ |
| | 1 | 0 | 0 | ÷ 16 | $32t_{cyc}$ |
| | | | 1 | ÷ 4 | $8t_{cyc}$ |

**Operating States:** The serial interface has the following operating states, which allow transitions to occur between them (figure 68).

— STS wait state
— Transmit clock wait state
— Transfer state
Continuous clock output state (only in internal clock mode)




Note: Refer to the Operating States section for the explanations on the corresponding encircled numbers.

**Figure 68 Serial Interface State Transitions**

- STS wait state: The serial interface enters STS wait state by MCU reset (00 and 10 in figure 68). In STS wait state, the serial interface is initialized and the transmit clock is ignored. If the STS instruction is then executed (01 and 11), the serial interface enters transmit clock wait state.

- Transmit clock wait state: Transmit clock wait state is the period between the STS execution and the falling edge of the first transmit clock. In transmit clock wait state, input of the transmit clock (02 and 12) increments the octal counter, shifts the serial data register (SRL: $006, SRU: $007), and enters the serial interface in transfer state. However, note that if continuous clock output state is selected in internal clock mode, the serial interface does not enter transfer state but enters continuous clock output state (17).

  The serial interface enters STS wait state by writing data to serial mode register A (SMRA: $005) (04 and 14) in transmit clock wait state.

- Transfer state: Transfer state is the period between the falling edge of the first clock and the rising edge of the eighth clock. In transfer state, the input of eight clocks or the execution of the STS instruction sets the octal counter to 000, and the serial interface enters another state. When the STS instruction is executed (05 and 15), transmit clock wait state is entered. When eight clocks are input, transmit clock wait state is entered (03) in external clock mode, or STS wait state is entered (13) in internal clock mode. In internal clock mode, the transmit clock stops after outputting eight clocks.

  In transfer state, writing data to serial mode register A (SMRA: $005) (06 and 16) initializes the serial interface, and STS wait state is entered.

  If the state changes from transfer to another state, the serial interrupt request flag (IFS: $023, bit 0) is set by the octal counter that is reset to 000.

- Continuous clock output state (only in internal clock mode): Continuous clock output state is entered only in internal clock mode. In this state, the serial interface does not transmit/receive data but only outputs the transmit clock from the $\overline{\text{SCK}}$ pin.

  When bits 0 and 1 (PMRA0, PMRA1) of port mode register A (PMRA: $004) are 00 in transmit clock wait state and if the transmit clock is input (17), the serial interface enters continuous clock output state. If serial mode register A (SMRA: $005) is written to in continuous clock output mode (18), STS wait state is entered.

**Output Level Control in Idle States:** When the serial interface is in STS instruction wait state and transmit clock wait state, the output of serial output pin SO can be controlled by setting bit 1 (SMRB1) of serial mode register B (SMRB: $028) to 0 or 1. See figure 69 for an output level control example of the serial interface. Note that the output level cannot be controlled in transfer state.

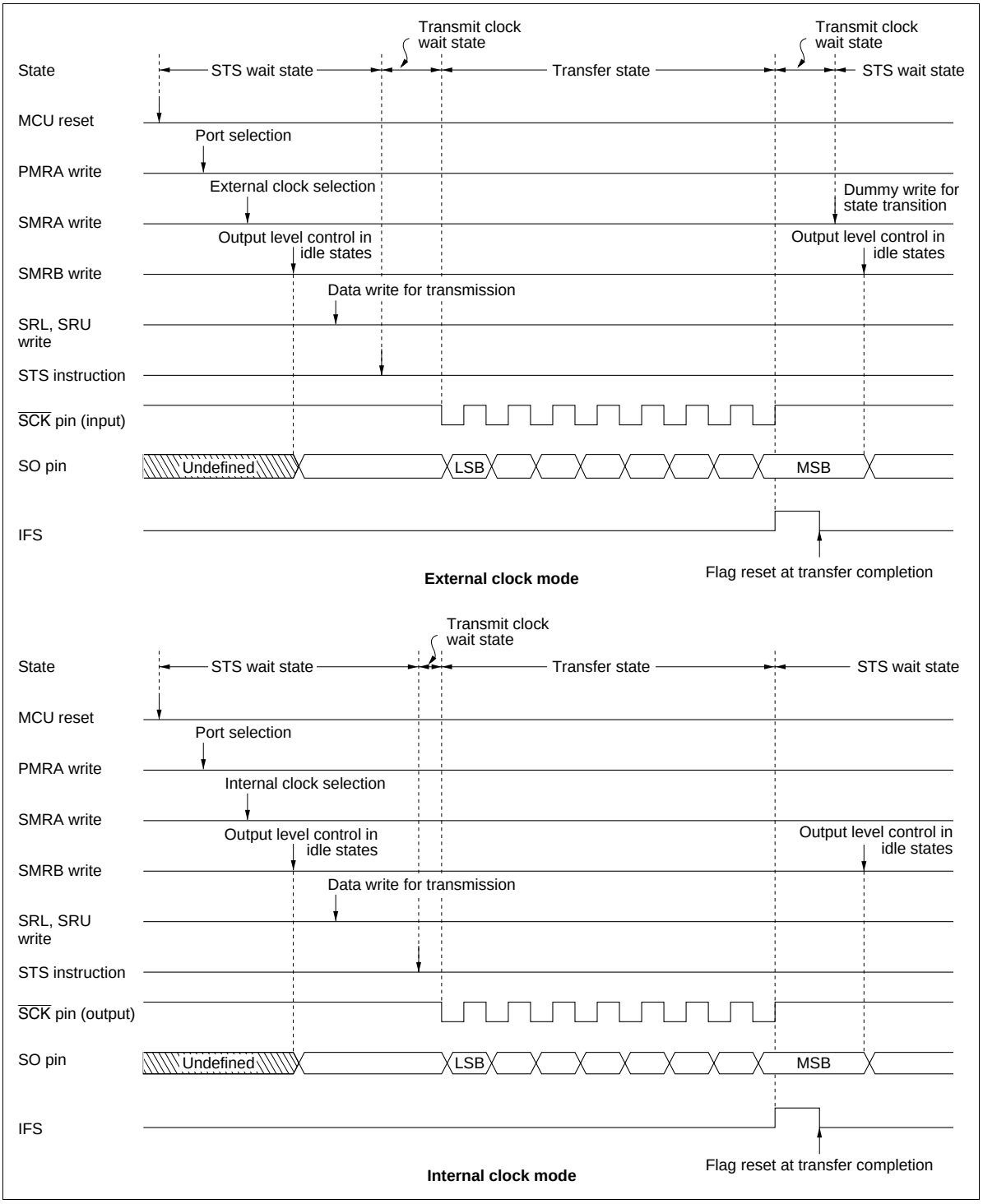


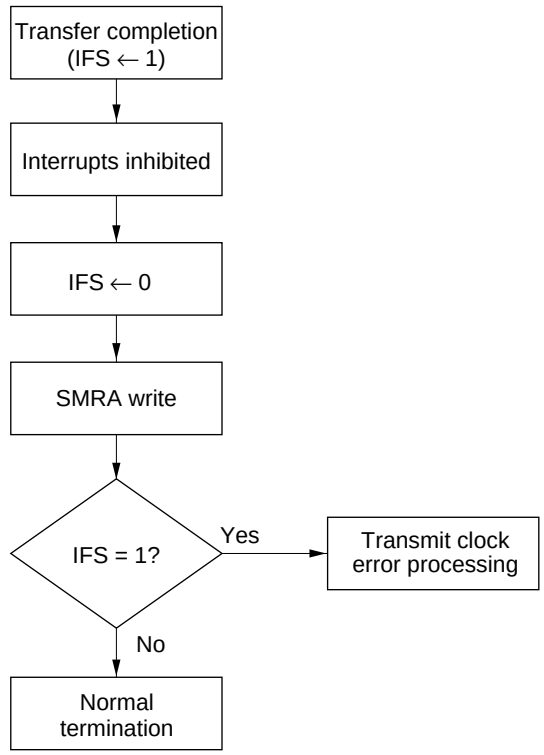

**Figure 69 Example of Serial Interface Operation Sequence**

**Transmit Clock Error Detection (In External Clock Mode):** The serial interface will malfunction if a spurious pulse caused by external noise conflicts with a normal transmit clock during transfer. A transmit clock error of this type can be detected (figure 70).

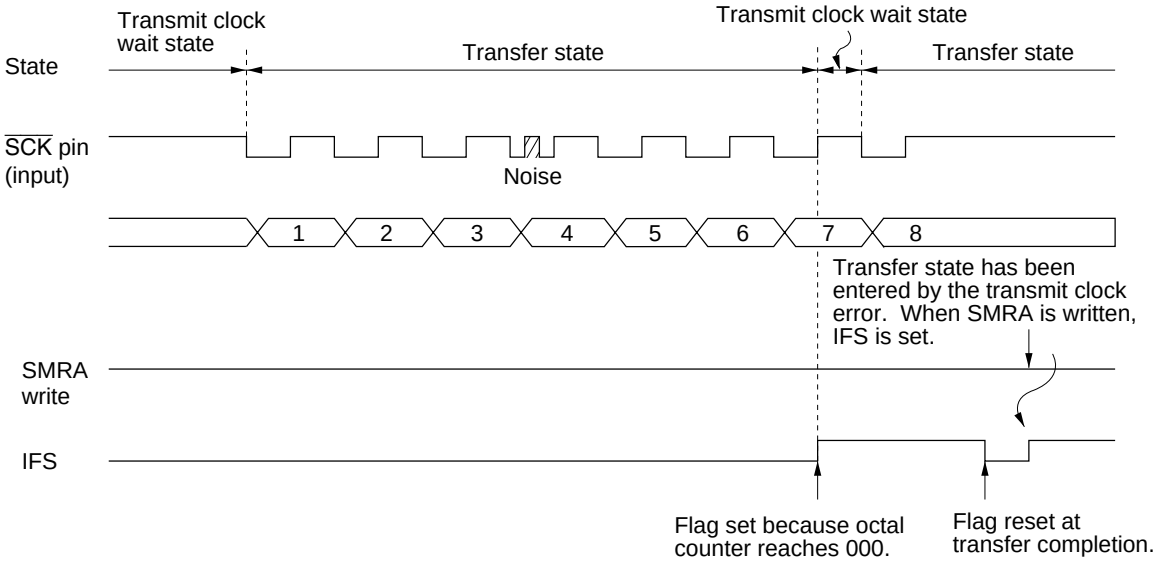
If more than eight transmit clocks are input in transfer state, at the eighth clock including a spurious pulse by noise, the octal counter reaches 000, the serial interrupt request flag (IFS: $023, bit 0) is set, and transmit clock wait state is entered. At the falling edge of the next normal clock signal, the transfer state is again entered. After the transfer is completed and IFS is reset, writing to serial mode register A (SMRA: $005) then changes the state from transfer to STS wait. However, during the time the serial interface was in the transfer state with the serial interrupt request flag (IFS: $023, bit 0) being set again, the error can be detected.

**Notes on Use:**

- Initialization after writing to registers: If port mode register A (PMRA: $004) is written to in transmit clock wait state or in transfer state, the serial interface must be initialized by writing to serial mode register A (SMRA: $005) again.
- Serial interrupt request flag (IFS: $023, bit 0) set: For the serial interface, if the state is changed from transfer state to another by writing to serial mode register A (SMRA: $005) or executing the STS instruction during the first low pulse of the transmit clock, the serial interrupt request flag (IFS: $023, bit 0) is not set. To set the serial interrupt request flag (IFS: $023, bit 0), a serial mode register A (SMRA: $005) write or STS instruction execution must be programmed to be executed after confirming that the $\overline{\text{SCK}}$ pin is at 1, that is, after executing the input instruction to port R4.




**Transmit clock error detection flowchart**




**Transmit clock error detection procedures**

**Figure 70 Transmit Clock Error Detection**

**Registers for Serial Interface**

The serial interface operation is selected, and serial data is read and written by the following registers.

— Serial mode register A (SMRA: $005)
— Serial mode register B (SMRB: $028)
— Serial data register (SRL: $006, SRU: $007)
— Port mode register A (PMRA: $004)
— Miscellaneous register (MIS: $00C)

**Serial Mode Register A (SMRA: $005):** This register has the following functions (figure 71).

- $R4_1/\overline{SCK}$ pin function selection
- Transmit clock selection
- Prescaler division ratio selection
- Serial interface initialization

Serial mode register A (SMRA: $005) is a 4-bit write-only register. It is reset to $0 by MCU reset.

A write signal input to serial mode register A (SMRA: $005) discontinues the input of the transmit clock to the serial data register (SRL: $006, SRU: $007) and octal counter, and the octal counter is reset to 000. Therefore, if a write is performed during data transfer, the serial interrupt request flag (IFS: $023, bit 0) is set.

Written data is valid from the second instruction execution cycle after the write operation, so the STS instruction must be executed at least two cycles after that.

**Serial mode register A (SMRA: $005)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | SMRA3 | SMRA2 | SMRA1 | SMRA0 |

| SMRA3 | $R4_1/\overline{SCK}$ mode selection |
|---|---|
| 0 | $R4_1$ |
| 1 | $\overline{SCK}$ |

| SMRA2 | SMRA1 | SMRA0 | $\overline{SCK}$ | Clock source | Prescaler division ratio |
|---|---|---|---|---|---|
| 0 | 0 | 0 | Output | Prescaler | Refer to table 29 |
| | | 1 | | | |
| | 1 | 0 | | | |
| | | 1 | | | |
| 1 | 0 | 0 | | | |
| | | 1 | | | |
| | 1 | 0 | Output | System clock | — |
| | | 1 | Input | External clock | — |

**Figure 71 Serial Mode Register A (SMRA)**

**Serial Mode Register B (SMRB: $028):** This register has the following functions (figure 72).

- Prescaler division ratio selection
- Output level control in idle states

Serial mode register B (SMRB: $028) is a 2-bit write-only register. It cannot be written during data transfer.

By setting bit 0 (SMRB0) of this register, the prescaler division ratio is selected. Only bit 0 (SMRB0) can be reset to 0 by MCU reset. By setting bit 1 (SMRB1), the output level of the SO pin is controlled in idle states of the serial interface. The output level changes at the same time that SMRB1 is written to.

**Serial mode register B (SMRB: $028)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | Undefined | 0 |
| Read/Write | — | — | W | W |
| Bit name | Not used | Not used | SMRB1 | SMRB0 |

| SMRB1 | Output level control in idle states |
|---|---|
| 0 | Low level |
| 1 | High level |

| SMRB0 | Serial clock division ratio |
|---|---|
| 0 | Prescaler output divided by 2 |
| 1 | Prescaler output divided by 4 |

**Figure 72 Serial Mode Register B (SMRB)**

**Serial Data Register (SRL: $006, SRU: $007):** This register has the following functions (figures 73 and 74).

- Transmission data write and shift
- Receive data shift and read

Writing data in this register is output from the SO pin, LSB first, synchronously with the falling edge of the transmit clock (figure 75); data is input, LSB first, through the SI pin at the rising edge of the transmit clock.

Data cannot be read or written during serial data transfer. If a read/write occurs during transfer, the accuracy of the resultant data cannot be guaranteed.

**Serial data register (lower digit) (SRL: $006)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R/W | R/W | R/W | R/W |
| Bit name | SR3 | SR2 | SR1 | SR0 |

**Figure 73 Serial Data Register Lower Digit (SRL)**

**Serial data register (upper digit) (SRU: $007)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R/W | R/W | R/W | R/W |
| Bit name | SR7 | SR6 | SR5 | SR4 |

**Figure 74 Serial Data Register Upper Digit (SRU)**

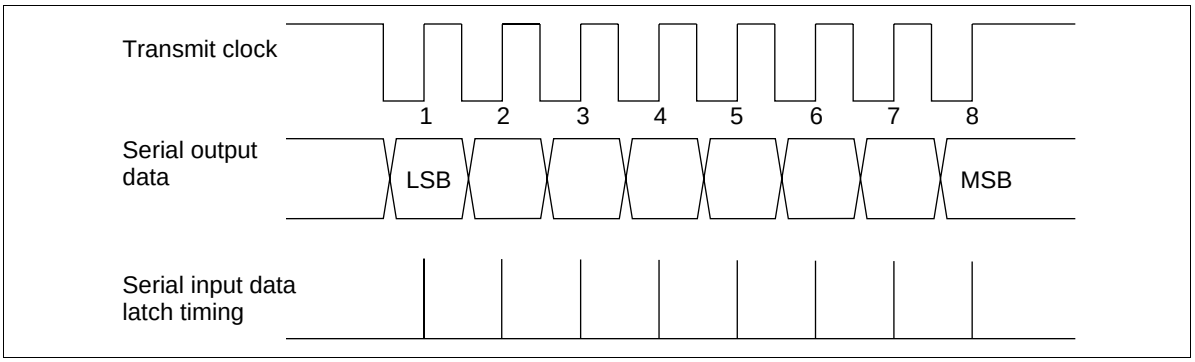



**Figure 75 Serial Interface Output Timing**

**Port Mode Register A (PMRA: $004):** This register has the following functions (figure 76).

- $R4_2$/SI pin function selection
- $R4_3$/SO pin function selection

Port mode register A (PMRA: $004) is a 2-bit write-only register, and is reset to $0 by MCU reset.

**Port mode register A (PMRA: $004)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | 0 | 0 |
| Read/Write | — | — | W | W |
| Bit name | Not used | Not used | PMRA1 | PMRA0 |

| PMRA0 | $R4_3$/SO mode selection |
|---|---|
| 0 | $R4_3$ |
| 1 | SO |

| PMRA1 | $R4_2$/SI mode selection |
|---|---|
| 0 | $R4_2$ |
| 1 | SI |

**Figure 76 Port Mode Register A (PMRA)**

**Miscellaneous Register (MIS: $00C):** This register has the following functions (figure 77).

- $R4_3$/SO pin PMOS control

Miscellaneous register (MIS: $00C) is a 4-bit write-only register and is reset to $0 by MCU reset.

**Miscellaneous register (MIS: $00C)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | MIS3 | MIS2 | MIS1 | MIS0 |

| MIS1 | MIS0 | $t_{RC}$ |
|---|---|---|
| 0 | 0 | 0.12207 ms |
| | 1 | 7.8125 ms |
| 1 | 0 | 62.5 ms |
| | 1 | Not used |

| MIS2 | $R4_3$/SO PMOS on/off selection |
|---|---|
| 0 | On |
| 1 | Off |

| MIS3 | Pull-up MOS on/off selection |
|---|---|
| 0 | Off |
| 1 | On |

**Figure 77 Miscellaneous Register (MIS)**

## Comparator

The comparator (figure 78) compares an analog input voltage with a reference voltage. Either a 16-level internal or external reference power supply can be selected.

The voltage comparison is started by writing 1 to the comparator start flag (CMSF: $020, bit 2), and is completed after $4t_{cyc}$. The comparison result is stored into bit 3 (CER: $017, bit 3) of the comparator enable register, and can be read by the bit test instruction (TM or TMD). The comparison result must be read after confirming that the comparator start flag (CMSF: $020, bit 2) is at 0 (figure 79).

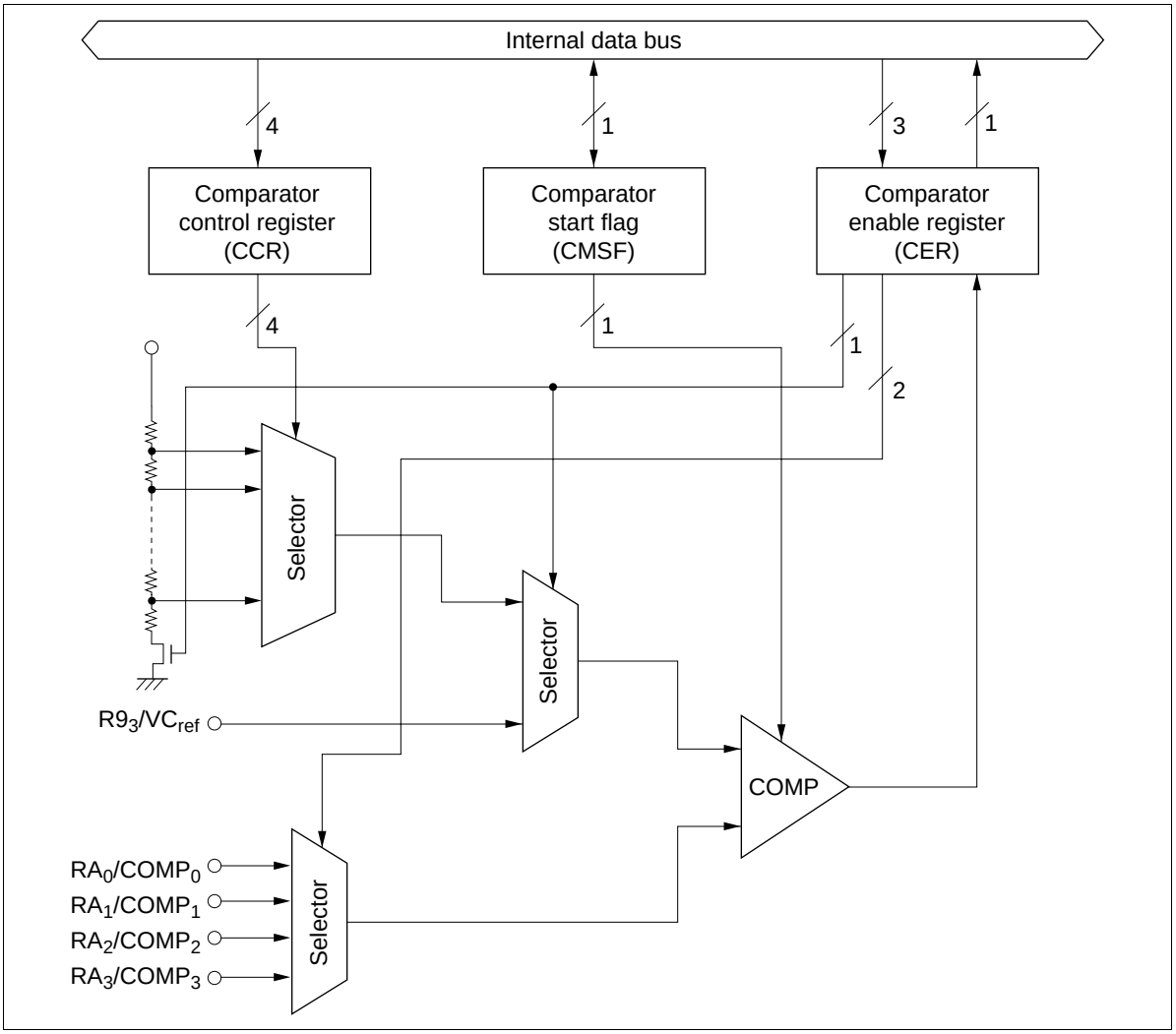



**Figure 78 Block Diagram of Comparator**

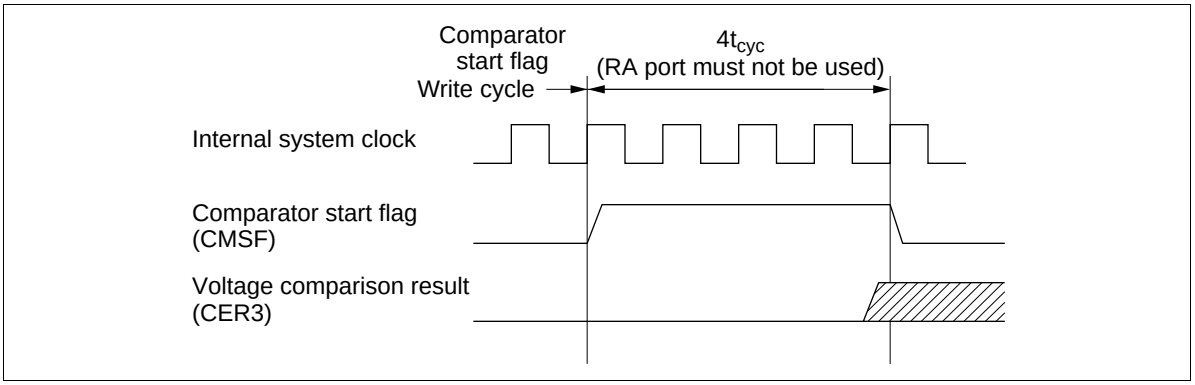



**Figure 79 Comparator Operation Timing**

**Comparator Control Register (CCR: $016):** Four-bit write-only register which selects a 16-level internal reference power supply (figure 80). The comparator control register (CCR: $016) is reset to $0 by MCU reset.

**Comparator control register (CCR: $016)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | CCR3 | CCR2 | CCR1 | CCR0 |

| CCR3 | CCR2 | CCR1 | CCR0 | Reference power supply selection |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1/17 $V_{CC}$ |
| | | | 1 | 2/17 $V_{CC}$ |
| | | 1 | 0 | 3/17 $V_{CC}$ |
| | | | 1 | 4/17 $V_{CC}$ |
| | 1 | 0 | 0 | 5/17 $V_{CC}$ |
| | | | 1 | 6/17 $V_{CC}$ |
| | | 1 | 0 | 7/17 $V_{CC}$ |
| | | | 1 | 8/17 $V_{CC}$ |
| 1 | 0 | 0 | 0 | 9/17 $V_{CC}$ |
| | | | 1 | 10/17 $V_{CC}$ |
| | | 1 | 0 | 11/17 $V_{CC}$ |
| | | | 1 | 12/17 $V_{CC}$ |
| | 1 | 0 | 0 | 13/17 $V_{CC}$ |
| | | | 1 | 14/17 $V_{CC}$ |
| | | 1 | 0 | 15/17 $V_{CC}$ |
| | | | 1 | 16/17 $V_{CC}$ |

**Figure 80 Comparator Control Register (CCR)**

**Comparator Enable Register (CER: $017):** This register consists of a 3-bit write-only register and a 1-bit read-only register. It selects the analog input pins and reference voltage, and indicates the voltage comparison result. The comparison result output is 0 when an analog input voltage is lower than the reference voltage, and is 1 when an analog input voltage is higher than the reference voltage. The comparison result is read by the bit test instruction (TM or TMD). The comparator enable register (CER: $017) is reset to $0 by MCU reset.

**Comparator enable register (CER: $017)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | R | W | W | W |
| Bit name | CER3 | CER2 | CER1 | CER0 |

| CER1 | CER0 | Analog input mode selection |
|---|---|---|
| 0 | 0 | $COMP_0$ |
| | 1 | $COMP_1$ |
| 1 | 0 | $COMP_2$ |
| | 1 | $COMP_3$ |

| CER2 | Reference power supply selection |
|---|---|
| 0 | External reference power supply |
| 1 | Internal reference power supply |

| CER3 | Voltage comparison result |
|---|---|
| 0 | Analog input voltage is lower than reference voltage |
| 1 | Analog input voltage is higher than reference voltage |

**Figure 81 Comparator Enable Register (CER)**

**Comparator Start Flag (CMSF: $020, Bit 2):** Starts the comparator operation. The comparator starts the voltage comparison by writing 1 to the comparator start flag (CMSF: $020, bit 2), and automatically completes the voltage comparison after $4t_{cyc}$. The comparator start flag is then reset to 0. The comparison result must be read after confirming that the comparator start flag is at 0. The comparator start flag is reset to 0 by MCU reset.

**Notes on Use**: $RA_0/COMP_0$–$RA_3/COMP_3$ pins are used only for the comparator during voltage comparison. These pins cannot be used for R ports.

The comparator operates only in the active and standby modes.

The switch for the internal power supply is turned on when the internal power supply is selected. The switch is turned off except in active and standby modes.

When the external power supply is used for a reference voltage, $R9_3/VC_{ref}$ must not be used as an R port.

## Notes on Mounting

Assemble all parts including the HD404458/HD404459 on a board, noting the points described below.

Between the $V_{CC}$ and GND lines, connect capacitors designed for use in ordinary power supply circuits. An example connection is described in figure 82.

No resistors can be inserted in series in the power supply circuit, so the capacitors should be connected in parallel.

The capacitors are a large capacitance $C_1$ and a small capacitance $C_2$.

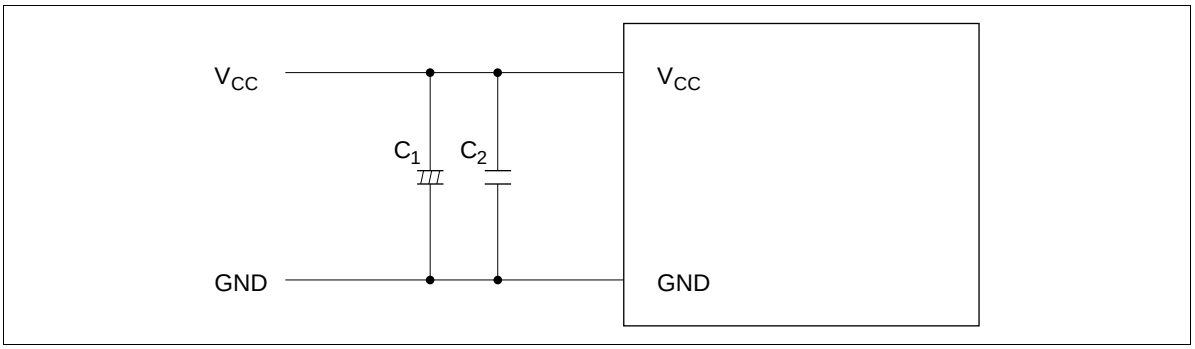



**Figure 82 Example of Connections**

## Programmable ROM (HD4074459)

The HD4074459 is a ZTAT™ microcomputer with a built-in PROM that can be programmed in PROM mode.

### PROM Mode Pin Description

| Pin No. | MCU Mode | | PROM Mode | | Pin No. | MCU Mode | | PROM Mode | |
|---|---|---|---|---|---|---|---|---|---|
| FP-64A | Pin Name | I/O | Pin Name | I/O | FP-64A | Pin Name | I/O | Pin Name | I/O |
| 1 | $RA_0/COMP_0$ | I | | | 29 | $R1_0$ | I/O | $A_5$ | I |
| 2 | $RA_1/COMP_1$ | I | | | 30 | $R1_1$ | I/O | $A_6$ | I |
| 3 | $RA_2/COMP_2$ | I | | | 31 | $R1_2$ | I/O | $A_7$ | I |
| 4 | $RA_3/COMP_3$ | I | | | 32 | $R1_3$ | I/O | $A_8$ | I |
| 5 | $\overline{TEST}$ | I | $\overline{TEST}$ | I | 33 | $R2_0$ | I/O | $A_0$ | I |
| 6 | $OSC_1$ | I | $V_{CC}$ | | 34 | $R2_1$ | I/O | $A_{10}$ | I |
| 7 | $OSC_2$ | O | | | 35 | $R2_2$ | I/O | $A_{11}$ | I |
| 8 | GND | — | GND | — | 36 | $R2_3$ | I/O | $A_{12}$ | I |
| 9 | X2 | O | | | 37 | $R3_0$/TOB | I/O | | |
| 10 | X1 | I | GND | | 38 | $R3_1$/TOC | I/O | | |
| 11 | RESET | I | RESET | I | 39 | $R3_2$/TOD | I/O | | |
| 12 | $D_0$ | I/O | $O_0$ | I/O | 40 | $R3_3/\overline{EVNB}$ | I/O | | |
| 13 | $D_1$ | I/O | $O_1$ | I/O | 41 | $R4_0$/EVND | I/O | | |
| 14 | $D_2$ | I/O | $O_2$ | I/O | 42 | $R4_1/\overline{SCK}$ | I/O | | |
| 15 | $D_3$ | I/O | $O_3$ | I/O | 43 | $R4_2$/SI | I/O | | |
| 16 | $D_4$ | I/O | $O_4$ | I/O | 44 | $R4_3$/SO | I/O | | |
| 17 | $D_5$ | I/O | $O_5$ | I/O | 45 | $R5_0/(\overline{WU}_0)$ | I/O | | |
| 18 | $D_6$ | I/O | $O_6$ | I/O | 46 | $R5_1/(\overline{WU}_1)$ | I/O | | |
| 19 | $D_7$ | I/O | $O_7$ | I/O | 47 | $R5_2/(\overline{WU}_2)$ | I/O | | |
| 20 | $D_8$ | I/O | $A_{13}$ | I | 48 | $R5_3/(\overline{WU}_3)$ | I/O | | |
| 21 | $D_9$ | I/O | $A_{14}$ | I | 49 | $R6_0/(\overline{WU}_4)$ | I/O | $\overline{CE}$ | I |
| 22 | $D_{10}$ | I | $V_{PP}$ | I | 50 | $R6_1/(\overline{WU}_5)$ | I/O | $\overline{OE}$ | I |
| 23 | $D_{11}/\overline{STOPC}$ | I | $A_9$ | I | 51 | $R6_2/(\overline{WU}_6)$ | I/O | $V_{CC}$ | |
| 24 | $V_{CC}$ | — | $V_{CC}$ | | 52 | $R6_3/(\overline{WU}_7)$ | I/O | $V_{CC}$ | |
| 25 | $R0_0/\overline{INT}_0$ | I/O | $\overline{M}_0$ | I | 53 | $R7_0$ | I/O | $A_1$ | I |
| 26 | $R0_1/\overline{INT}_1$ | I/O | $\overline{M}_1$ | I | 54 | $R7_1$ | I/O | $A_2$ | I |
| 27 | $R0_2/INT_2$ | I/O | | | 55 | $R7_2$ | I/O | $A_3$ | I |
| 28 | $R0_3/INT_3$ | I/O | | | 56 | $R7_3$ | I/O | $A_4$ | I |

| Pin No. | MCU Mode | | PROM Mode | | Pin No. | MCU Mode | | PROM Mode | |
|---|---|---|---|---|---|---|---|---|---|
| FP-64A | Pin Name | I/O | Pin Name | I/O | FP-64A | Pin Name | I/O | Pin Name | I/O |
| 57 | $R8_0$ | I/O | $O_4$ | I/O | 61 | $R9_0$ | I/O | $O_0$ | I/O |
| 58 | $R8_1$ | I/O | $O_3$ | I/O | 62 | $R9_1$ | I/O | $V_{CC}$ | |
| 59 | $R8_2$ | I/O | $O_2$ | I/O | 63 | $R9_2$ | I/O | | |
| 60 | $R8_3$ | I/O | $O_1$ | I/O | 64 | $R9_3/VC_{ref}$ | I | | |

Notes: 1. I/O: Input/output pin, I: Input pin, O: Output pin
2. Each of $O_0$–$O_4$ has two pins; before using them, each pair must be connected together.

## Programming the Built-In PROM

The MCU's built-in PROM is programmed in PROM mode. This PROM mode is set by pulling $\overline{TEST}$, $\overline{M}_0$, and $\overline{M}_1$ low, and RESET high (figure 83). In PROM mode, the MCU does not operate, but it can be programmed in the same way as any other commercial 27256-type EPROM using a standard PROM programmer and a 64-to-28-pin socket adapter. Refer to table 31 for the Recommended PROM programmers and socket adapters of the HD4074459.

Since an HMCS400-series instruction is ten bits long, the HMCS400-series MCU has a built-in conversion circuit to enable the use of a general-purpose PROM programmer. This circuit splits each instruction into five lower bits and five upper bits that are read from or written to consecutive addresses. This means that if, for example, 16 kwords of built-in PROM are to be programmed by a general-purpose PROM programmer, a 32-kbyte address space ($0000–$7FFF) must be specified.

### Warnings

1. Always specify addresses $0000 to $7FFF when programming with a PROM programmer. If address $8000 or higher is accessed, the PROM may not be programmed or verified correctly. Set all data in unused addresses to $FF.

   Note that the plastic-package versions cannot be erased or reprogrammed.

2. Make sure that the PROM programmer, socket adapter, and LSI are aligned correctly (their pin 1 positions match), otherwise overcurrents may damage the LSI. Before starting programming, make sure that the LSI is firmly fixed in the socket adapter and the socket adapter is firmly fixed onto the programmer.

3. PROM programmers have two voltages ($V_{PP}$): 12.5 V and 21 V. Remember that ZTAT™ devices require a $V_{PP}$ of 12.5 V—the 21-V setting will damage them. 12.5 V is the Intel 27256 setting.

### Programming and Verification

The built-in PROM of the MCU can be program med at high speed without risk of voltage stress or damage to data reliability.

Refer to table 30 for programming and verification modes.

For details of PROM programming, refer to the preface section, Notes on PROM Programming.

**Table 30 PROM Mode Selection**

| | Pin | | | |
|---|---|---|---|---|
| **Mode** | **$\overline{CE}$** | **$\overline{OE}$** | **$V_{PP}$** | **$O_0$–$O_7$** |
| Programming | Low | High | $V_{PP}$ | Data input |
| Verification | High | Low | $V_{PP}$ | Data output |
| Programming inhibited | High | High | $V_{PP}$ | High impedance |

**Table 31 Recommended PROM Programmers and Socket Adapters**

| PROM Programmer | | Socket Adapter | | |
|---|---|---|---|---|
| **Manufacturer** | **Model Name** | **Package** | **Model Name** | **Manufacturer** |
| DATA I/O Corp. | 121B | FP-64A | HS4459ESH01H | Hitachi |
| AVAL Corp. | PKW-1000 | FP-64A | HS4459ESH01H | Hitachi |

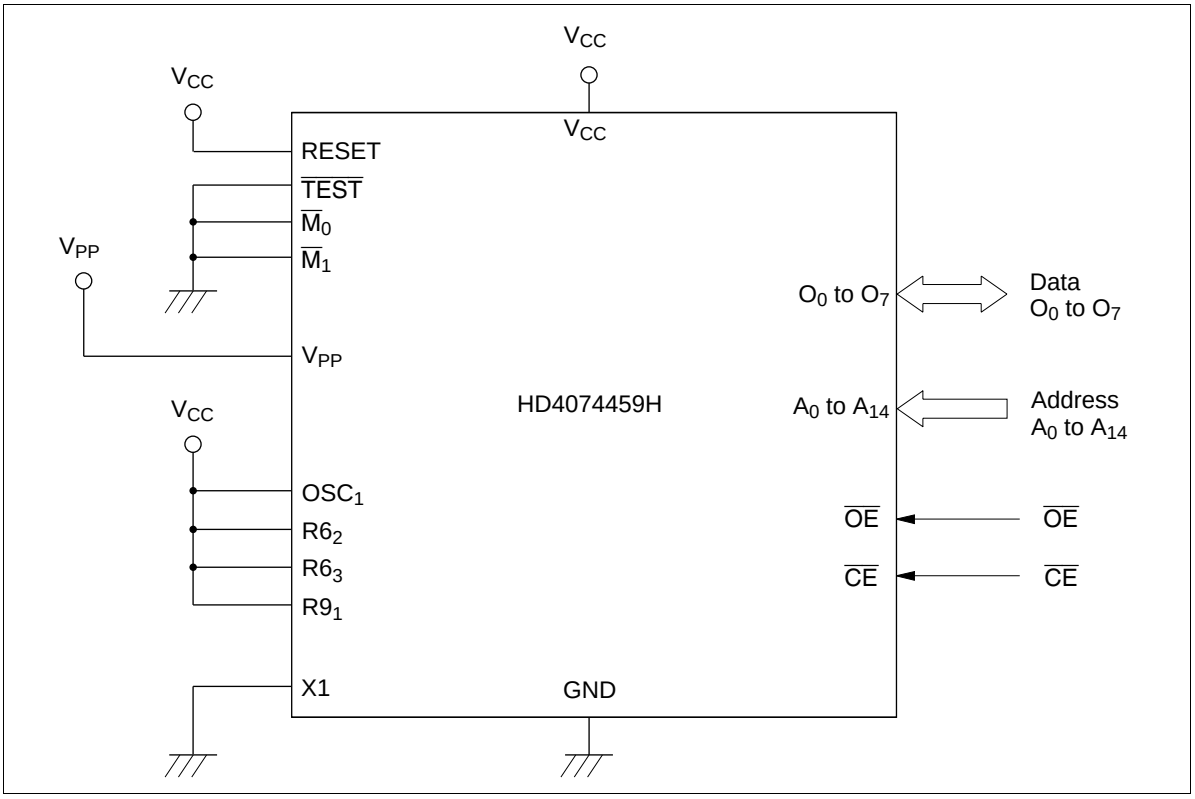



**Figure 83 PROM Mode Connections**

## Addressing Modes

### RAM Addressing Modes

The MCU has three RAM addressing modes (figure 84).

**Register Indirect Addressing Mode:** The contents of the W, X, and Y registers (10 bits in total) are used for RAM addressing.

**Direct Addressing Mode:** A direct addressing instruction consists of two words. The first word contains the opcode, and the contents of the second word (10 bits) are used for RAM addressing.

**Memory Register Addressing Mode:** The memory registers (MR), which are located in 16 addresses from $040 to $04F, are accessed with the LAMR and XMRA instructions.

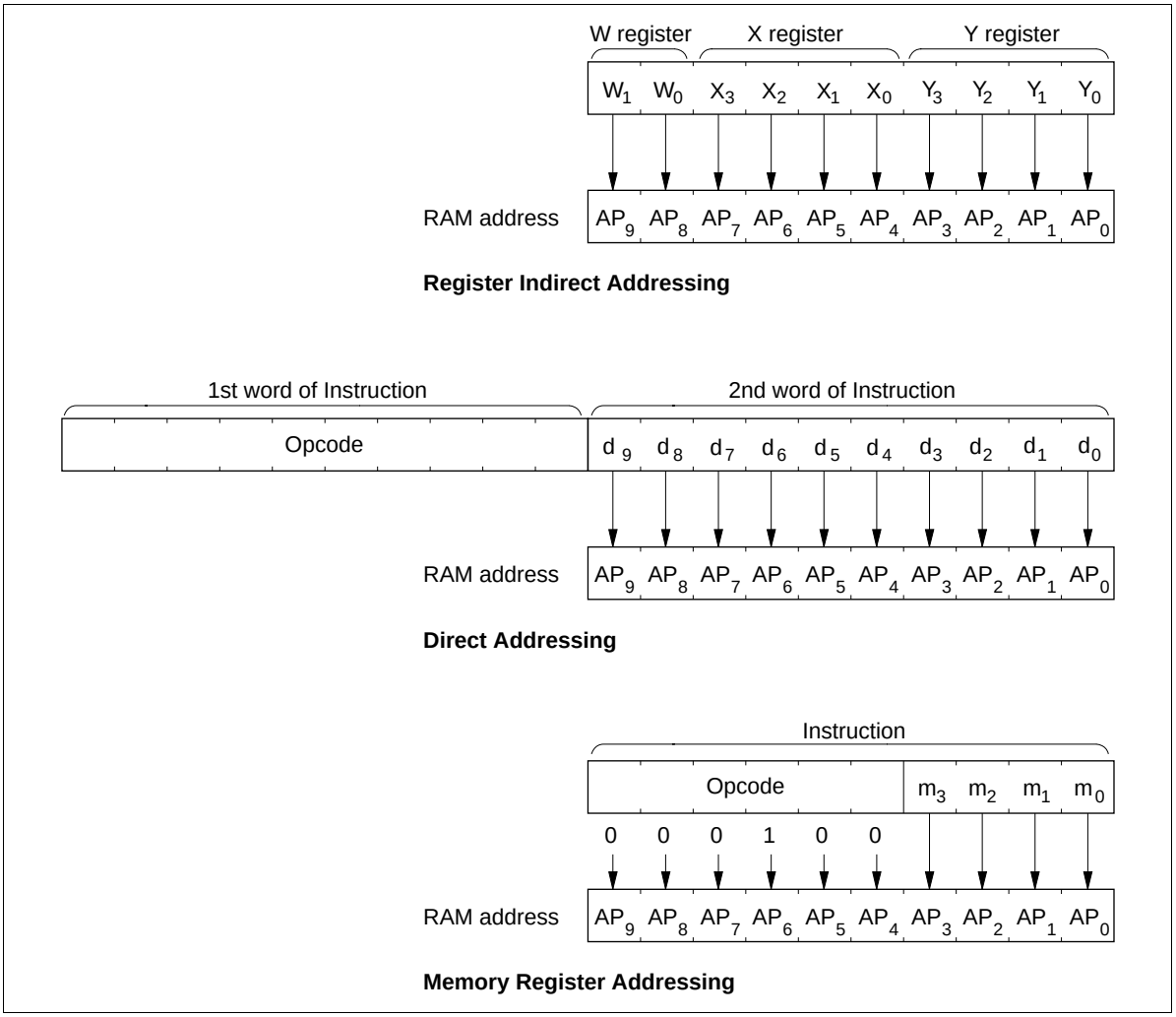



**Figure 84 RAM Addressing Modes**

**ROM Addressing Modes and the P Instruction**

The MCU has four ROM addressing modes (figure 85).

**Direct Addressing Mode:** A program can branch to any address in ROM memory space by executing the JMPL, BRL, or CALL instruction. Each of these instructions replaces the 14 program counter bits ($PC_{13}$–$PC_0$) with 14-bit immediate data.

**Current Page Addressing Mode:** The MCU has 64 pages of ROM with 256 words per page. A program can branch to any address in the current page by executing the BR instruction. This instruction replaces the eight low-order bits of the program counter ($PC_7$–$PC_0$) with eight-bit immediate data. If the BR instruction is on a page boundary (address 256n + 255), executing that instruction transfers the PC contents to the next physical page (figure 87). This means that the execution of the BR instruction on a page boundary will make the program branch to the next page.

Note that the HMCS400-series cross macroassembler has an automatic paging feature for ROM pages.

**Zero-Page Addressing Mode:** A program can branch to the zero-page subroutine area located at $0000–$003F by executing the CAL instruction. When the CAL instruction is executed, 6 bits of immediate data are placed in the six low-order bits of the program counter ($PC_5$–$PC_0$), and 0s are placed in the eight high-order bits ($PC_{13}$–$PC_6$).

**Table Data Addressing Mode:** A program can branch to an address determined by the contents of four-bit immediate data, the accumulator, and the B register by executing the TBR instruction.

**P Instruction:** ROM data addressed in table data addressing mode can be referenced with the P instruction (figure 86). If bit 8 of the ROM data is 1, eight bits of ROM data are written to the accumulator and the B register. If bit 9 is 1, eight bits of ROM data are written to the R1 and R2 port output registers. If both bits 8 and 9 are 1, ROM data is written to the accumulator and the B register, and also to the R1 and R2 port output registers at the same time.

The P instruction has no effect on the program counter.

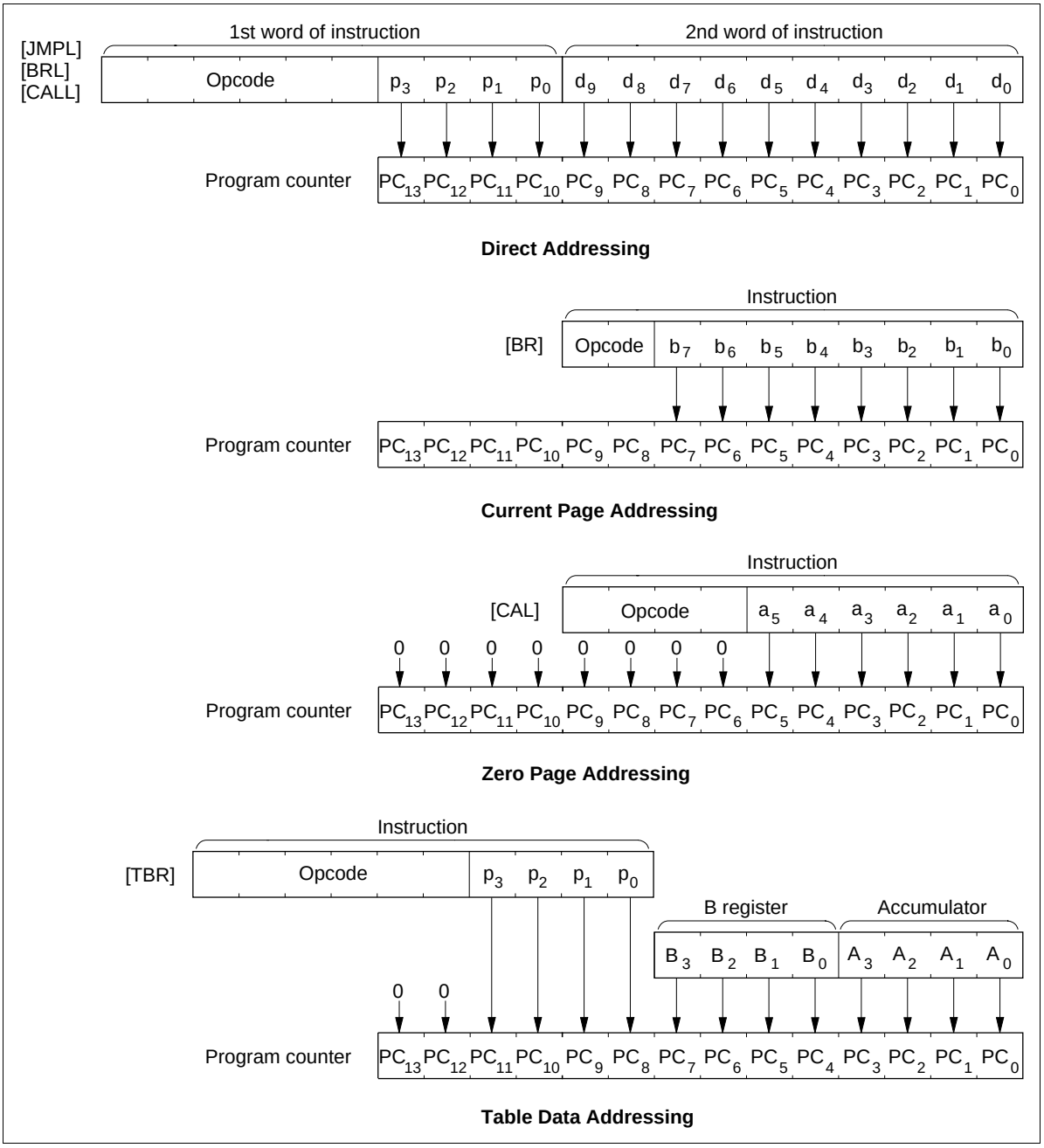



**Figure 85 ROM Addressing Modes**

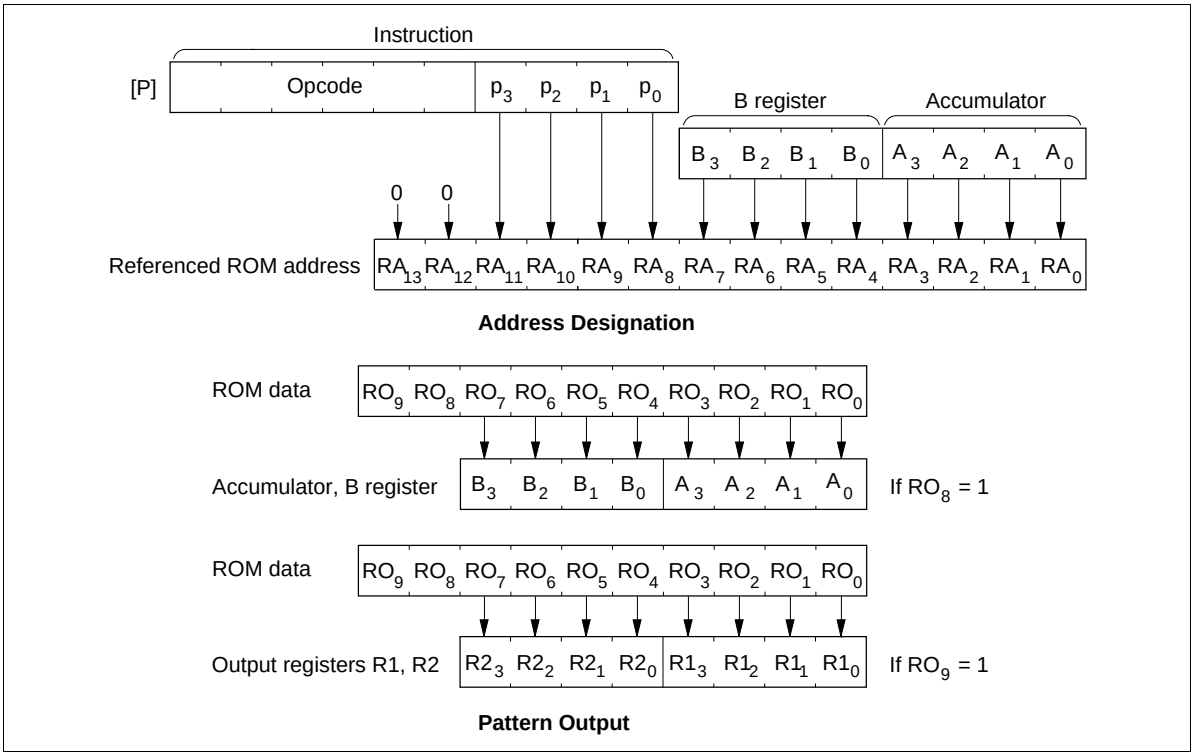



**Figure 86 P Instruction**

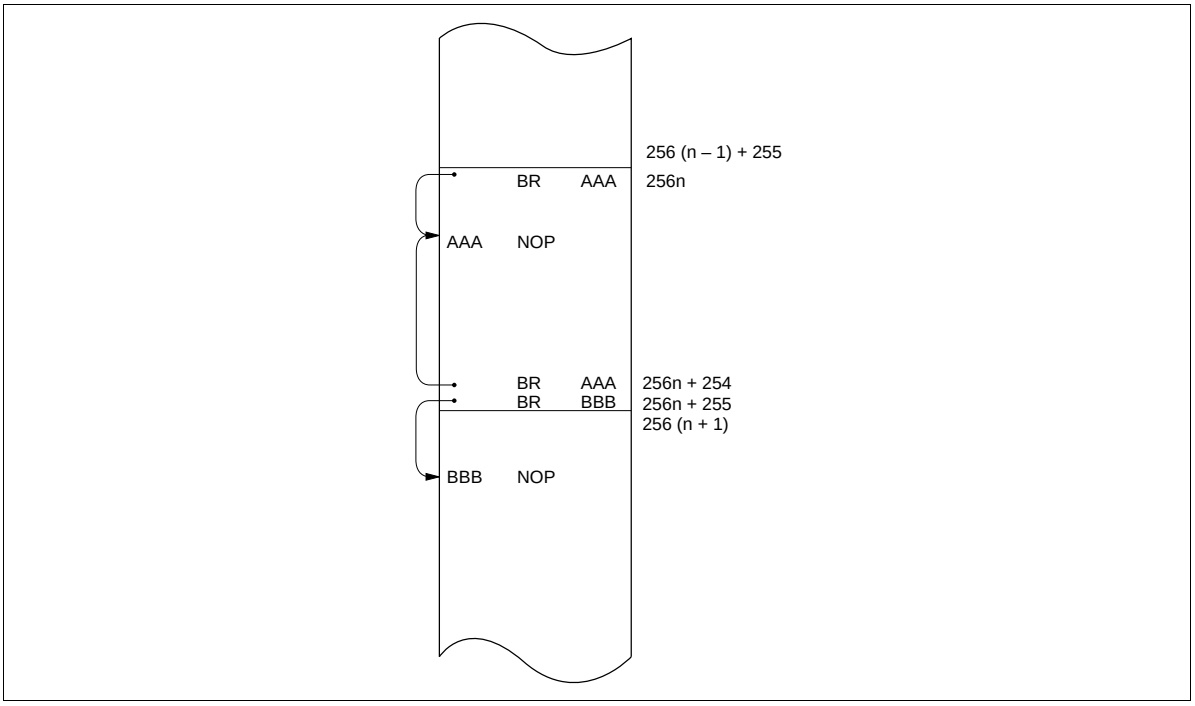



**Figure 87 Branching when the Branch Destination is on a Page Boundary**

## Absolute Maximum Ratings (HD404458/HD404459)

| Item | Symbol | Value | Unit | Notes |
|---|---|---|---|---|
| Supply voltage | $V_{CC}$ | –0.3 to +4.0 | V | |
| Pin voltage | $V_T$ | –0.3 to ($V_{CC}$ + 0.3) | V | |
| Total permissible input current | $\sum I_o$ | 50 | mA | 2 |
| Total permissible output current | $-\sum I_o$ | 50 | mA | 3 |
| Maximum input current | $I_o$ | 4 | mA | 4, 5 |
| Maximum output current | $-I_o$ | 4 | mA | 5, 6 |
| Operating temperature | $T_{opr}$ | –20 to +75 | °C | |
| Storage temperature | $T_{stg}$ | –55 to +125 | °C | |

## Absolute Maximum Ratings (HD4074459)

| Item | Symbol | Value | Unit | Notes |
|---|---|---|---|---|
| Supply voltage | $V_{CC}$ | –0.3 to +4.0 | V | |
| Programming voltage | $V_{PP}$ | –0.3 to +14.0 | V | 1 |
| Pin voltage | $V_T$ | –0.3 to ($V_{CC}$ + 0.3) | V | |
| Total permissible input current | $\sum I_o$ | 50 | mA | 2 |
| Total permissible output current | $-\sum I_o$ | 50 | mA | 3 |
| Maximum input current | $I_o$ | 4 | mA | 4, 5 |
| Maximum output current | $-I_o$ | 4 | mA | 5, 6 |
| Operating temperature | $T_{opr}$ | –20 to +75 | °C | 7 |
| Storage temperature | $T_{stg}$ | –55 to +125 | °C | |

Notes: Permanent damage may occur if these absolute maximum ratings are exceeded. Normal operation must be under the conditions stated in the electrical characteristics tables. If these conditions are exceeded, the LSI may malfunction or its reliability may be affected.

1. Applies to $D_{10}$ ($V_{PP}$) of the HD4074459.
2. The total permissible input current is the total of input currents simultaneously flowing in from all the I/O pins to ground.
3. The total permissible output current is the total of output currents simultaneously flowing out from $V_{CC}$ to all I/O pins.
4. The maximum input current is the maximum current flowing from each I/O pin to ground.
5. Applies to $D_0$–$D_9$, R0–R8, and $R9_0$–$R9_2$.
6. The maximum output current is the maximum current flowing out from $V_{CC}$ to each I/O pin.
7. Depends on the supply voltage.

## Electrical Characteristics

### DC Characteristics

**HD404458, HD404459: $V_{CC}$ = 1.8 to 3.6 V, GND = 0 V, $T_a$ = –20 to +75°C, $f_{OSC}$ = 0.4 to 4.0 MHz**
**HD4074459: $V_{CC}$ = 2.2 to 2.7 V, GND = 0 V, $T_a$ = –5 to +60°C, $f_{OSC}$ = 0.4 to 2.0 MHz;**
**$V_{CC}$ = 2.7 to 3.6 V, GND = 0 V, $T_a$ = –20 to +75°C, $f_{OSC}$ = 0.4 to 4.0 MHz, unless otherwise specified.**

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Notes |
|---|---|---|---|---|---|---|---|---|
| Input high voltage | $V_{IH}$ | RESET, $\overline{STOPC}$, $\overline{INT_0}$, $\overline{INT_1}$, $INT_2$, $INT_3$, $\overline{SCK}$, SI, $\overline{WU_0}$–$\overline{WU_7}$, $\overline{EVNB}$, EVND | $0.9V_{CC}$ | — | $V_{CC}$ + 0.3 | V | — | |
| | | $OSC_1$ | $V_{CC}$ – 0.3 | — | $V_{CC}$ + 0.3 | V | External clock operation | |
| Input low voltage | $V_{IL}$ | RESET, $\overline{STOPC}$, $\overline{INT_0}$, $\overline{INT_1}$, $INT_2$, $INT_3$, $\overline{SCK}$, SI, $\overline{WU_0}$–$\overline{WU_7}$, $\overline{EVNB}$, EVND | –0.3 | — | $0.1V_{CC}$ | V | — | |
| | | $OSC_1$ | –0.3 | — | 0.3 | V | External clock operation | |
| Output high voltage | $V_{OH}$ | $\overline{SCK}$, SO, TOB, TOC, TOD | $V_{CC}$ – 0.5 | — | — | V | $–I_{OH}$ = 0.3 mA | |
| Output low voltage | $V_{OL}$ | $\overline{SCK}$, SO, TOB, TOC, TOD | — | — | 0.4 | V | $I_{OL}$ = 0.4 mA | |
| I/O leakage current | $\lvert I_{IL} \rvert$ | RESET, $\overline{STOPC}$, $\overline{INT_0}$, $\overline{INT_1}$, $INT_2$, $INT_3$, $\overline{SCK}$, SI, $\overline{WU_0}$–$\overline{WU_7}$, SO, $\overline{EVNB}$, EVND, $OSC_1$, TOB, TOC, TOD | — | — | 1.0 | µA | $V_{in}$ = 0 V to $V_{CC}$ | 1 |
| Current dissipation in active mode | $I_{CC}$ | $V_{CC}$ | — | 3 | 6 | mA | HD404458, HD404459: $V_{CC}$ = 3.0 V, $f_{OSC}$ = 4 MHz | 2 |
| | | | — | 5 | 9 | mA | HD4074459: $V_{CC}$ = 3.0 V, $f_{OSC}$ = 4 MHz | 2 |

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Notes |
|---|---|---|---|---|---|---|---|---|
| Current dissipation in standby mode | $I_{SBY}$ | $V_{CC}$ | — | 1.2 | 3 | mA | $V_{CC}$ = 3.0 V, $f_{OSC}$ = 4 MHz | 3 |
| Current dissipation in subactive mode | $I_{SUB}$ | $V_{CC}$ | — | 35 | 70 | µA | HD404458, HD404459: $V_{CC}$ = 3.0 V, 32-kHz oscillator | |
| | | | — | 70 | 150 | µA | HD4074459: $V_{CC}$ = 3.0 V, 32-kHz oscillator | |
| Current dissipation in watch mode | $I_{WTC}$ | $V_{CC}$ | — | 8 | 15 | µA | $V_{CC}$ = 3.0 V, 32-kHz oscillator | 4 |
| Current dissipation in stop mode | $I_{STOP}$ | $V_{CC}$ | — | 1 | 10 | µA | $V_{CC}$ = 3.0 V, no 32-kHz oscillator | 4 |
| Stop mode retaining voltage | $V_{STOP}$ | $V_{CC}$ | 1.5 | — | — | V | No 32-kHz oscillator | 5 |

Notes: 1. Output buffer current is excluded.

2. $I_{CC}$ is the source current when no I/O current is flowing while the MCU is in reset state.
   Test conditions: MCU: Reset
   Pins: RESET at $V_{CC}$ (0.9$V_{CC}$ to $V_{CC}$)
   $\overline{TEST}$ at $V_{CC}$ (0.9$V_{CC}$ to $V_{CC}$)

3. $I_{SBY}$ is the source current when no I/O current is flowing while the MCU timer is operating.
   Test conditions: MCU: I/O reset
   Serial interface stopped
   Standby mode
   Pins: RESET at GND (0 V to 0.3 V)
   $\overline{TEST}$ at $V_{CC}$ (0.9$V_{CC}$ to $V_{CC}$)

4. These are the source currents when no I/O current is flowing.
   Test conditions: Pins: RESET at GND (0 V to 0.3 V)
   $\overline{TEST}$ at $V_{CC}$ (0.9$V_{CC}$ to $V_{CC}$)
   $D_{10}$* at $V_{CC}$ (0.9$V_{CC}$ to $V_{CC}$)
   Note: * Applies to HD4074459

5. RAM data retention is the voltage required for retaining RAM data.

### I/O Characteristics for Standard Pins

**HD404458, HD404459: $V_{CC}$ = 1.8 to 3.6 V, GND = 0 V, $T_a$ = –20 to +75°C, $f_{OSC}$ = 0.4 to 4.0 MHz**
**HD4074459: $V_{CC}$ = 2.2 to 2.7 V, GND = 0 V, $T_a$ = –5 to +60°C, $f_{OSC}$ = 0.4 to 2.0 MHz;**
**$V_{CC}$ = 2.7 to 3.6 V, GND = 0 V, $T_a$ = –20 to +75°C, $f_{OSC}$ = 0.4 to 4.0 MHz, unless otherwise specified.**

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Note |
|---|---|---|---|---|---|---|---|---|
| Input high voltage | $V_{IH}$ | $D_0$–$D_{11}$, R0–RA | $0.7V_{CC}$ | — | $V_{CC}$ + 0.3 | V | — | |
| Input low voltage | $V_{IL}$ | $D_0$–$D_{11}$, R0–RA | –0.3 | — | $0.3V_{CC}$ | V | — | |
| Output high voltage | $V_{OH}$ | $D_0$–$D_9$, R0–R8, $R9_0$–$R9_2$ | $V_{CC}$ – 0.5 | — | — | V | $–I_{OH}$ = 0.3 mA | |
| Output low voltage | $V_{OL}$ | $D_0$–$D_9$, R0–R8, $R9_0$–$R9_2$ | — | — | 0.4 | V | $I_{OL}$ = 0.4 mA | |
| I/O leakage current | $\lvert I_{IL} \rvert$ | $D_0$–$D_{11}$, R0–RA | — | — | 1 | µA | HD404458, HD404459: $V_{in}$ = 0 V to $V_{CC}$ | 1 |
| | | $D_0$–$D_9$, $D_{11}$, R0–RA | — | — | 1 | µA | HD4074459: $V_{in}$ = 0 V to $V_{CC}$ | 1 |
| | | $D_{10}$ | — | — | 1 | µA | HD4074459: $V_{in}$ = $V_{CC}$ – 0.3 to $V_{CC}$ | 1 |
| | | | — | — | 20 | µA | HD4074459: $V_{in}$ = 0 V to 0.3 V | 1 |
| Pull-up MOS current | $–I_{PU}$ | $D_0$–$D_9$, R0–R8, $R9_0$–$R9_2$ | 5 | 40 | 90 | µA | $V_{CC}$ = 3.0 V, $V_{in}$ = 0 V | |

Note: 1. Output buffer current is excluded.

### Voltage Comparator Characteristics

**HD404458, HD404459: $V_{CC}$ = 2.0 to 3.6 V, GND = 0 V, $T_a$ = –10 to +75°C, $f_{OSC}$ = 0.4 to 4.0 MHz**
**HD4074459: $V_{CC}$ = 2.2 to 2.7 V, GND = 0 V, $T_a$ = –5 to +60°C, $f_{OSC}$ = 0.4 to 2.0 MHz;**
**$V_{CC}$ = 2.7 to 3.6 V, GND = 0 V, $T_a$ = –10 to +75°C, $f_{OSC}$ = 0.4 to 4.0 MHz,unless otherwise specified.**

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Note |
|---|---|---|---|---|---|---|---|---|
| Input high voltage | $V_{IHA}$ | $COMP_0$–$COMP_3$ | $V_{ref}$ + 0.17 | — | — | V | — | 1 |
| Input low voltage | $V_{ILA}$ | $COMP_0$–$COMP_3$ | — | — | $V_{ref}$ – 0.03 | V | — | 1 |
| Analog input standard voltage range | $VC_{ref}$ | $VC_{ref}$ | 0 | — | $V_{CC}$ | V | — | |

Note: 1. When an internal reference voltage is selected, the standard voltage is an expected voltage of internal $V_{ref}$ specified by the comparator control register (CCR).

## AC Characteristics

**HD404458, HD404459: $V_{CC}$ = 1.8 to 3.6 V, GND = 0 V, $T_a$ = –20 to +75°C, $f_{OSC}$ = 0.4 to 4.0 MHz**
**HD4074459: $V_{CC}$ = 2.2 to 2.7 V, GND = 0 V, $T_a$ = –5 to +60°C, $f_{OSC}$ = 0.4 to 2.0 MHz;**
**$V_{CC}$ = 2.7 to 3.6 V, GND = 0 V, $T_a$ = –20 to +75°C, $f_{OSC}$ = 0.4 to 4.0 MHz, unless otherwise specified.**

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Notes |
|---|---|---|---|---|---|---|---|---|
| Clock oscillation frequency | $f_{OSC}$ | $OSC_1$, $OSC_2$ | 0.4 | — | 4.0 | MHz | HD404458, HD404459: 1/4division, $V_{CC}$ = 1.8 V to 3.6 V | |
| | | | | | | | HD4074459: 1/4 division, $V_{CC}$ = 2.7 V to 3.6 V | |
| | | | 0.4 | — | 2.0 | MHz | HD4074459: 1/4 division, $V_{CC}$ = 2.2 V to 2.7 V | |
| | | X1, X2 | — | 32.768 | — | kHz | — | |
| Instruction cycle time | $t_{cyc}$ | — | 1.0 | — | 10 | µs | HD404458, HD404459: 1/4 division, $V_{CC}$ = 1.8 V to 3.6 V | |
| | | | | | | | HD4074459: 1/4 division, $V_{CC}$ = 2.7 V to 3.6 V | |
| | | | 2.0 | — | 10 | µs | HD4074459: 1/4 division, $V_{CC}$ = 2.2 V to 2.7 V | |
| | $t_{subcyc}$ | — | — | 244.14 | — | µs | 32-kHz oscillator, 1/8 division | |
| | | | — | 122.07 | — | µs | 32-kHz oscillator, 1/4 division | |
| Oscillation stabilization time (ceramic oscillator) | $t_{RC}$ | $OSC_1$, $OSC_2$ | — | — | 60 | ms | — | 1 |
| Oscillation stabilization time (crystal oscillator) | $t_{RC}$ | $OSC_1$, $OSC_2$ | — | — | 60 | ms | — | 1 |
| | | X1, X2 | — | — | 3 | s | $T_a$ = –10°C to+60°C | 2 |
| External clock high width | $t_{CPH}$ | $OSC_1$ | 105 | — | — | ns | $f_{OSC}$ = 4 MHz | 3 |
| External clock low width | $t_{CPL}$ | $OSC_1$ | 105 | — | — | ns | $f_{OSC}$ = 4 MHz | 3 |

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Notes |
|---|---|---|---|---|---|---|---|---|
| External clock rise time | $t_{CPr}$ | $OSC_1$ | — | — | 20 | ns | — | 3 |
| External clock fall time | $t_{CPf}$ | $OSC_1$ | — | — | 20 | ns | — | 3 |
| $\overline{INT}_0$–$INT_3$, $\overline{EVNB}$, $\overline{WU}_0$–$\overline{WU}_7$, EVND high widths | $t_{IH}$ | $\overline{INT}_0$–$INT_3$, $\overline{WU}_0$–$\overline{WU}_7$, $\overline{EVNB}$, EVND | 2 | — | — | $t_{cyc}$/ $t_{subcyc}$ | — | 4, 7 |
| $\overline{INT}_0$–$INT_3$, $\overline{EVNB}$, $\overline{WU}_0$–$\overline{WU}_7$, EVND low widths | $t_{IL}$ | $\overline{INT}_0$–$INT_3$, $\overline{WU}_0$–$\overline{WU}_7$, $\overline{EVNB}$, EVND | 2 | — | — | $t_{cyc}$/ $t_{subcyc}$ | — | 4, 7 |
| RESET high width | $t_{RSTH}$ | RESET | 2 | — | — | $t_{cyc}$ | — | 5 |
| $\overline{STOPC}$ low width | $t_{STPL}$ | $\overline{STOPC}$ | 1 | — | — | $t_{RC}$ | — | 6 |
| RESET fall time | $t_{RSTf}$ | RESET | — | — | 20 | ms | — | 5 |
| $\overline{STOPC}$ rise time | $t_{STPr}$ | $\overline{STOPC}$ | — | — | 20 | ms | — | 6 |
| Input capacitance | $C_{in}$ | All pins except for $D_{10}$ | — | — | 15 | pF | f = 1 MHz, $V_{in}$ = 0 V | |
| | | $D_{10}$ | — | — | 15 | pF | HD404458, HD404459: f = 1MHz, $V_{in}$ = 0 V | |
| | | | — | — | 180 | pF | HD4074459: f = 1 MHz, $V_{in}$ = 0 V | |

Notes:
1. The oscillation stabilization time is the period required for the oscillator to stabilize after $V_{CC}$ reaches 1.8 V (2.2 V: HD4074459) at power-on, or after RESET input goes high or $\overline{STOPC}$ input goes low when stop mode is cancelled. At power-on or when stop mode is cancelled, RESET or $\overline{STOPC}$ must be input for at least $t_{RC}$ to ensure the oscillation stabilization time. If using a ceramic or crystal oscillator, contact its manufacturer to determine the required stabilization time, since it will depend on the circuit constants and stray capacitances. Set bits 0 and 1 (MIS0, MIS1) of the miscellaneous register (MIS: $00C) according to the oscillation stabilization time of the system oscillation.
2. The oscillation stabilization time is the period required for the oscillator to stabilize after $V_{CC}$ reaches 1.8 V (2.2 V: HD4074459) at power-on, or after RESET input goes high or $\overline{STOPC}$ input goes low when stop mode is cancelled. If using a crystal oscillator, contact its manufacturer to determine the required stabilization time, since it will depend on the circuit constants and stray capacitances.
3. Refer to figure 88.
4. Refer to figure 89. The $t_{cyc}$ unit applies when the MCU is in standby or active mode. The $t_{subcyc}$ unit applies when the MCU is in watch or subactive mode.
5. Refer to figure 90.
6. Refer to figure 91.
7. In watch or subactive mode, the periods when the $\overline{INT}_0$ and $\overline{WU}_0$–$\overline{WU}_7$ signals are high and when these signals are low must be equal to the interrupt frame period or longer.

### Serial Interface Timing Characteristics

**HD404458, HD404459: $V_{CC}$ = 1.8 to 3.6 V, GND = 0 V, $T_a$ = –20 to +75°C, $f_{OSC}$ = 0.4 to 4.0 MHz**
**HD4074459: $V_{CC}$ = 2.2 to 2.7 V, GND = 0 V, $T_a$ = –5 to +60°C, $f_{OSC}$ = 0.4 to 2.0 MHz;**
**$V_{CC}$ = 2.7 to 3.6 V, GND = 0 V, $T_a$ = –20 to +75°C, $f_{OSC}$ = 0.4 to 4.0 MHz, unless otherwise specified.**

### During Transmit Clock Output

| Item | Symbol | Pin | Min | Typ | Max | Unit | Test Condition | Note |
|---|---|---|---|---|---|---|---|---|
| Transmit clock cycle time | $t_{Scyc}$ | $\overline{SCK}$ | 1.0 | — | — | $t_{cyc}$ | Load shown in figure 93 | 1 |
| Transmit clock high width | $t_{SCKH}$ | $\overline{SCK}$ | 0.4 | — | — | $t_{Scyc}$ | Load shown in figure 93 | 1 |
| Transmit clock low width | $t_{SCKL}$ | $\overline{SCK}$ | 0.4 | — | — | $t_{Scyc}$ | Load shown in figure 93 | 1 |
| Transmit clock rise time | $t_{SCKr}$ | $\overline{SCK}$ | — | — | 200 | ns | Load shown in figure 93 | 1 |
| Transmit clock fall time | $t_{SCKf}$ | $\overline{SCK}$ | — | — | 200 | ns | Load shown in figure 93 | 1 |
| Serial output data delay time | $t_{DSO}$ | SO | — | — | 500 | ns | Load shown in figure 93 | 1 |
| Serial input data setup time | $t_{SSI}$ | SI | 300 | — | — | ns | — | 1 |
| Serial input data hold time | $t_{HSI}$ | SI | 300 | — | — | ns | — | 1 |

Note: 1. Refer to figure 92.

### During Transmit Clock Input

| Item | Symbol | Pin | Min | Typ | Max | Unit | Test Condition | Note |
|---|---|---|---|---|---|---|---|---|
| Transmit clock cycle time | $t_{Scyc}$ | $\overline{SCK}$ | 1.0 | — | — | $t_{cyc}$ | — | 1 |
| Transmit clock high width | $t_{SCKH}$ | $\overline{SCK}$ | 0.4 | — | — | $t_{Scyc}$ | — | 1 |
| Transmit clock low width | $t_{SCKL}$ | $\overline{SCK}$ | 0.4 | — | — | $t_{Scyc}$ | — | 1 |
| Transmit clock rise time | $t_{SCKr}$ | $\overline{SCK}$ | — | — | 200 | ns | — | 1 |
| Transmit clock fall time | $t_{SCKf}$ | $\overline{SCK}$ | — | — | 200 | ns | — | 1 |
| Serial output data delay time | $t_{DSO}$ | SO | — | — | 500 | ns | Load shown in figure 93 | 1 |
| Serial input data setup time | $t_{SSI}$ | SI | 300 | — | — | ns | — | 1 |
| Serial input data hold time | $t_{HSI}$ | SI | 300 | — | — | ns | — | 1 |

Note: 1. Refer to figure 92.

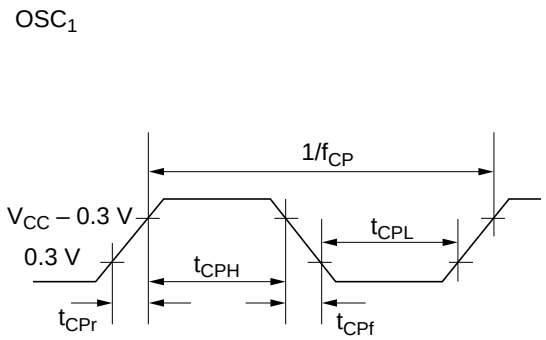



**Figure 88 External Clock Timing**

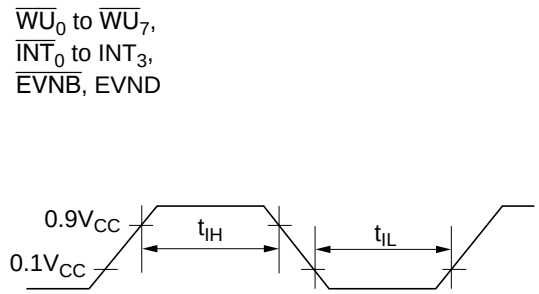



**Figure 89 Interrupt Timing**

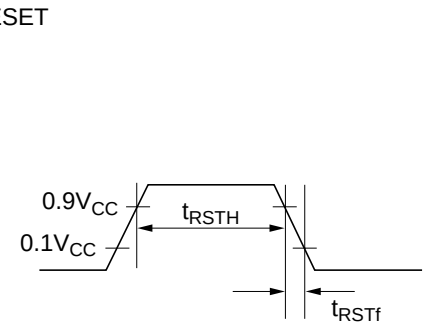



**Figure 90 Reset Timing**

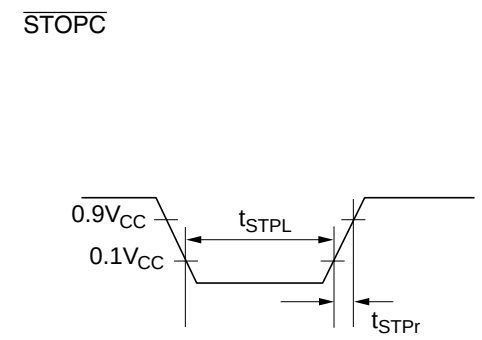
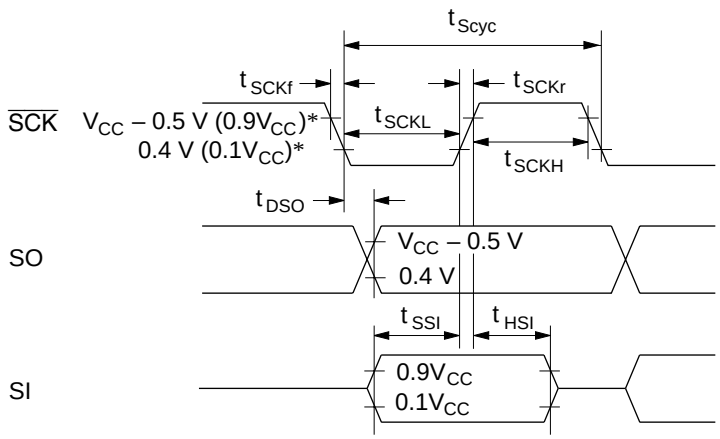



**Figure 91 $\overline{\text{STOPC}}$ Timing**




Note: * $V_{CC}$ – 0.5 V and 0.4 V are the threshold voltages for transmit clock output, and 0.9$V_{CC}$ and 0.1$V_{CC}$ are the threshold voltages for transmit clock input.

**Figure 92 Serial Interface Timing**

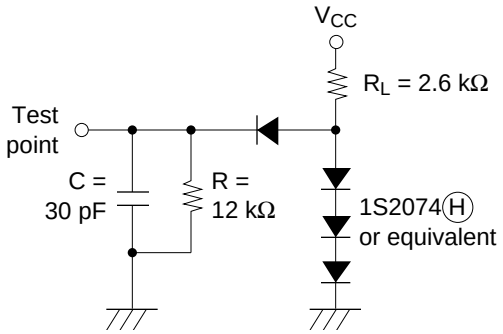


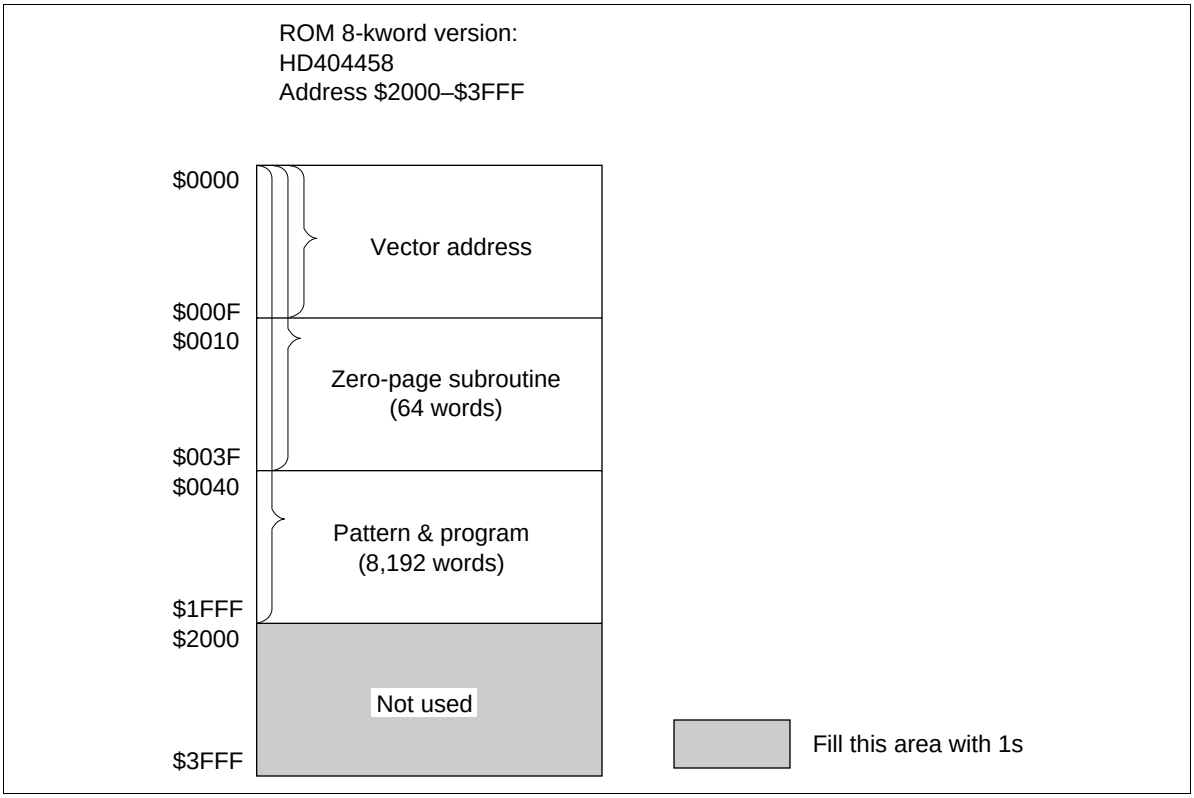

**Figure 93 Timing Load Circuit**

## Notes on ROM Out

Please pay attention to the following items regarding ROM out.

On ROM out, fill the ROM area indicated below with 1s to create the same data size as a 16-kword version (HD404459). A 16-kword data size is required to change ROM data to mask manufacturing data since the program used is for a 16-kword version.

This limitation applies when using an EPROM or a data base.



## HD404458, HD404459 Option List

Please check off the appropriate applications and enter the necessary information.

| | |
|---|---|
| Date of order | |
| Customer | |
| Department | |
| Name | |
| ROM code name | |
| LSI number | |

1. ROM size

| | |
|---|---|
| ☐ HD404458 | 8-kword |
| ☐ HD404459 | 16-kword |

2. Optional Functions

| |
|---|
| *☐ With 32-kHz CPU operation, with time-base for clock |
| *☐ Without 32-kHz CPU operation, with time-base for clock |
| ☐ Without 32-kHz CPU operation, without time-base |

Note: * Options marked with an asterisk require a subsystem crystal oscillator (X1, X2).

3. ROM code media

Please specify the first type listed below (the upper bits and lower bits are mixed together), when using the EPROM on-package microcomputer type (including ZTAT™ version).

| |
|---|
| ☐ EPROM: The upper bits and lower bits are mixed together. The upper five bits and lower five bits are programmed to the same EPROM in alternating order (i.e., LULULU...). |
| ☐ EPROM: The upper bits and lower bits are separated. The upper five bits and lower five bits are programmed to different EPROMS. |

4. Oscillator for OSC1 and OSC2

| | | |
|---|---|---|
| ☐ Ceramic oscillator | f = | MHz |
| ☐ Crystal oscillator | f = | MHz |
| ☐ External clock | f = | MHz |

5. Stop mode

| |
|---|
| ☐ Used |
| ☐ Not used |

6. Package

| |
|---|
| ■ FP-64A |

## Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

Copyright © Hitachi, Ltd., 1998. All rights reserved. Printed in Japan.