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.

# HD404654 Series

4-Bit Single-Chip Microcomputer

Rev. 7.0
Sept. 1999

## Description

The HD404654 Series is a member of the HMCS400-series of microcomputers designed to increase program productivity with large-capacity memory. Each microcomputer has a high-precision dual-tone multi-frequency (DTMF) generator, three timers, serial interface, voltage comparator, and input capture circuit.

The HD404654 Series includes three chips: the HD404652 with 2 k-word ROM; the HD404654 with 4 k-word ROM; and the HD4074654 with 4 k-word PROM (ZTAT™ version).

The HD4074654 is a PROM version (ZTAT™ microcomputer). A program can be written to the PROM by a PROM writer, which can dramatically shorten system development periods and smooth the process from debugging to mass production. (The ZTAT™ version is 27256-compatible.)

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

## Features

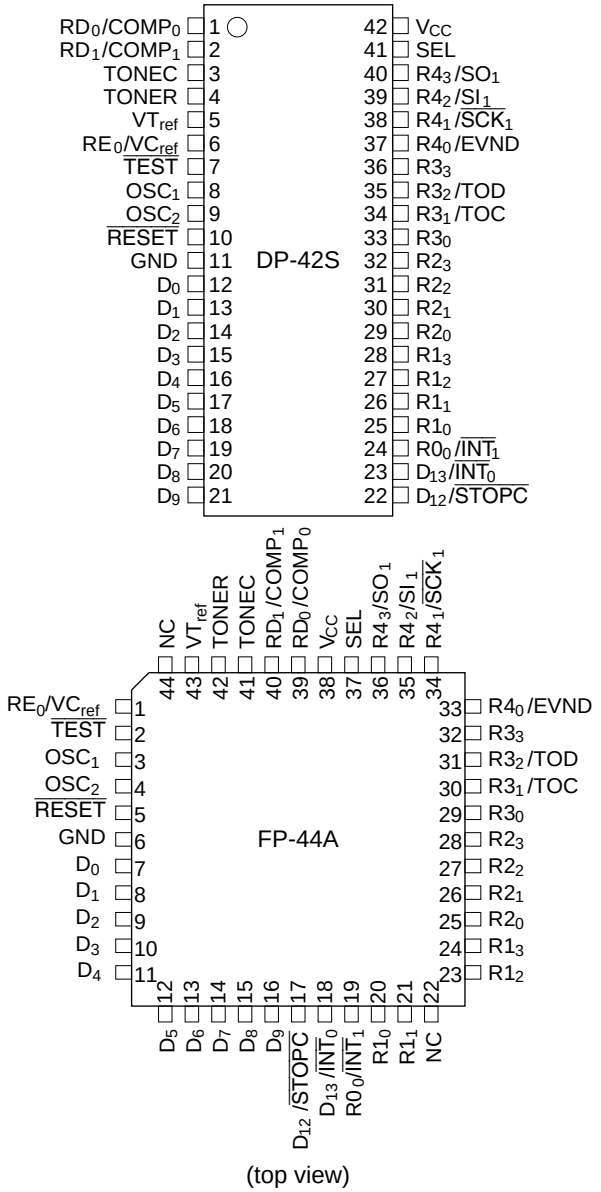
- 27 I/O pins and 5 dedicated input pins
  - 10 high-current output pins: Six 15-mA sinks and four 10-mA sources
- Three timer/counters
- Eight-bit input capture circuit
- Two timer outputs (including two PWM outputs)
- One event counter input (including one double-edge function)
- One clock-synchronous 8-bit serial interface
- Voltage comparator (2 channels)
- On-chip DTMF generator ($f_{OSC}$ = 400 kHz, 800 kHz, 2 MHz, 3.58 MHz or 4 MHz)
- Built-in oscillators
  - Main clock: Ceramic or crystal oscillator (an external clock is also possible)
- Six interrupt sources
  - Two by external sources
  - Four by internal sources
- Subroutine stack up to 16 levels, including interrupts

- Two low-power dissipation modes
  - Standby mode
  - Stop mode
- One external input for transition from stop mode to active mode
- Instruction cycle time: 1 µs ($f_{OSC}$ = 4 MHz at 1/4 division ratio)
  - 1/4 or 1/32 division ratio can be selected by hardware
- Two operating modes
  - MCU mode
  - MCU/PROM mode (HD4074654)

## Ordering Information

| Type | Product Name | Model Name | ROM (Words) | RAM (digit) | Package |
|---|---|---|---|---|---|
| Mask ROM | HD404652 | HD404652H | 2,048 | 512 | FP-44A |
| | | HD404652S | | | DP-42S |
| | HD404654 | HD404654H | 4,096 | | FP-44A |
| | | HD404654S | | | DP-42S |
| ZTAT™ | HD4074654 | HD4074654H | 4,096 | | FP-44A |
| | | HD4074654S | | | DP-42S |

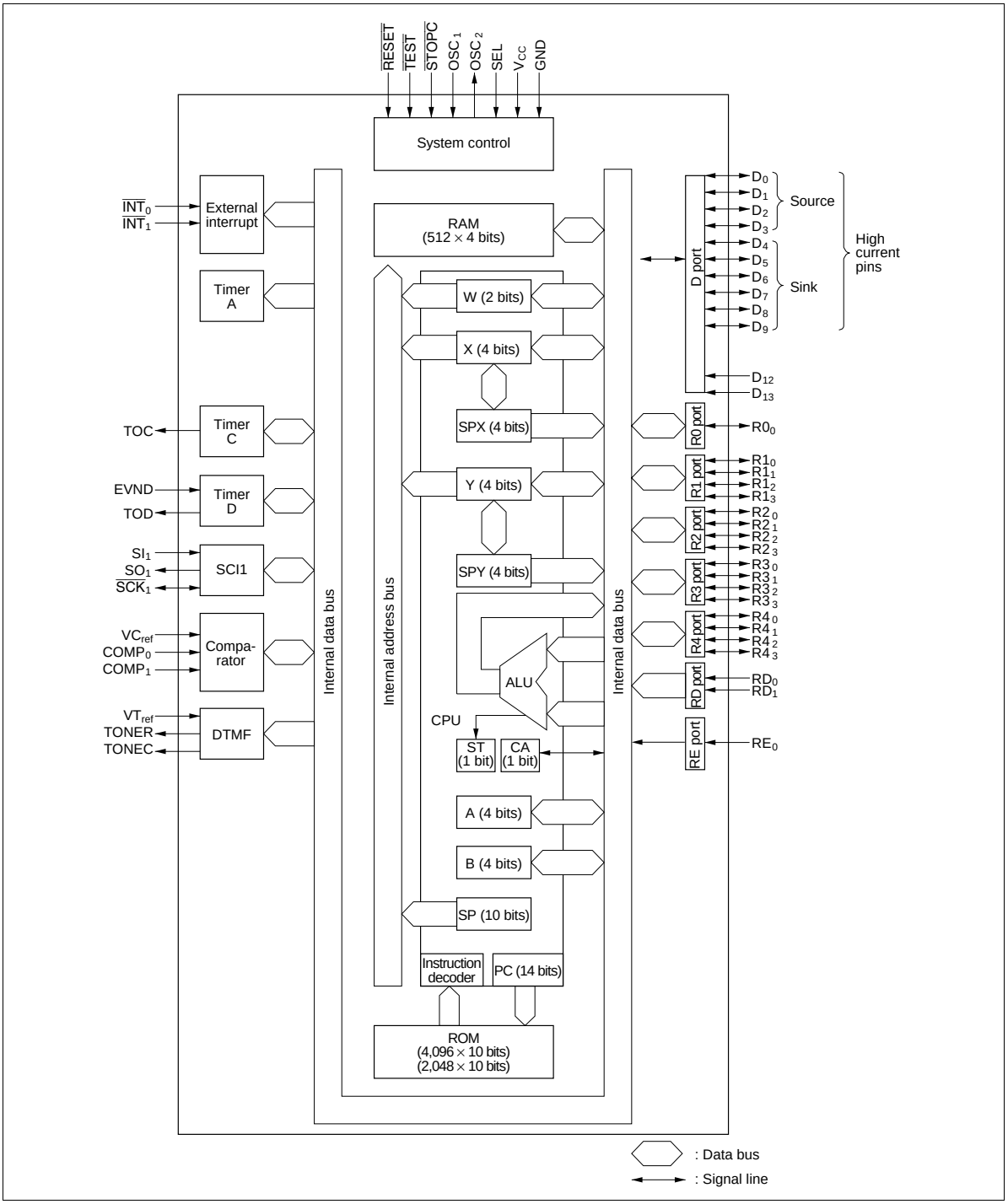
## Pin Arrangement



## Pin Description

| Item | Symbol | Pin Number DP-42S | Pin Number FP-44A | I/O | Function |
|---|---|---|---|---|---|
| Power supply | $V_{CC}$ | 42 | 38 | | Applies power voltage |
| | GND | 11 | 6 | | Connected to ground |
| Test | $\overline{TEST}$ | 7 | 2 | I | Used for factory testing only: Connect this pin to $V_{CC}$ |
| Reset | $\overline{RESET}$ | 10 | 5 | I | Resets the MCU |
| Oscillator | $OSC_1$ | 8 | 3 | I | |
| | $OSC_2$ | 9 | 4 | O | |
| Port | $D_0$–$D_9$ | 12–21 | 7–16 | I/O | Input/output pins addressed by individual bits; pins $D_4$–$D_9$ are high-current sink pins that can each supply up to 15 mA, $D_0$– $D_3$ are large-current source pins that can each supply up to 10 mA |
| | $D_{12}$, $D_{13}$ | 22, 23 | 17, 18 | I | Input pins addressable by individual bits |
| | $R0_0$–$R4_3$ | 24–40 | 19–21, 23–36 | I/O | Input/output pins addressable in 4-bit units |
| | $RD_0$, $RD_1$, $RE_0$ | 1, 2, 6 | 39, 40,1 | I | Input pins addressable in 4-bit units |
| Interrupt | $\overline{INT}_0$, $\overline{INT}_1$ | 23, 24 | 18, 19 | I | Input pins for external interrupts |
| Stop clear | $\overline{STOPC}$ | 22 | 17 | I | Input pin for transition from stop mode to active mode |
| Serial | $\overline{SCK}_1$ | 38 | 34 | I/O | Serial clock input/output pin |
| | $SI_1$ | 39 | 35 | I | Serial receive data input pin |
| | $SO_1$ | 40 | 36 | O | Serial transmit data output pin |
| Timer | TOC, TOD | 34, 35 | 30, 31 | O | Timer output pins |
| | EVND | 37 | 33 | I | Event count input pins |
| DTMF | TONER | 4 | 42 | O | Output pin for DTMF row signals |
| | TONEC | 3 | 41 | O | Output pin for DTMF column signals. |
| | $VT_{ref}$ | 5 | 43 | | Reference voltage pin for DTMF signals<br>Voltage condition is $V_{CC} \geq VT_{ref} \geq GND$. |
| Comparator | $COMP_0$, $COMP_1$ | 1, 2 | 39, 40 | I | Analog input pins for voltage comparator |
| | $VC_{ref}$ | 6 | 1 | | Reference voltage pin for inputting the threshold voltage of the analog input pin. |
| Division rate | SEL | 41 | 37 | I | Input pin for selecting system clock division rate rate after $\overline{RESET}$ input or after stop mode cancellation.<br>1/4 division rate: Connect it to $V_{CC}$<br>1/32 division rate: Connect it to GND |

## Block Diagram



## Memory Map

### ROM Memory Map

The ROM memory map is shown in figure 1 and described below.

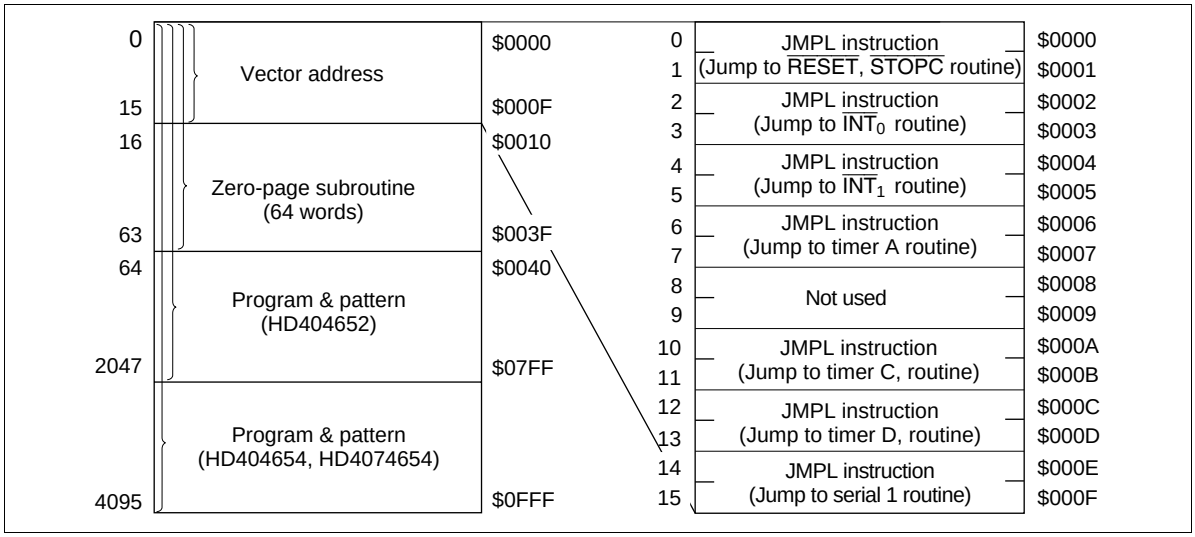



**Figure 1 ROM Memory Map**

**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–$07FF (HD404652), $0000–$0FFF (HD404654, HD4074654)):** Used for program coding.

### RAM Memory Map

The MCU contains a 512-digit × 4-bit RAM area consisting 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 as a RAM-mapped register area outside the above areas. The RAM memory map is shown in figure 2 and described as follows.

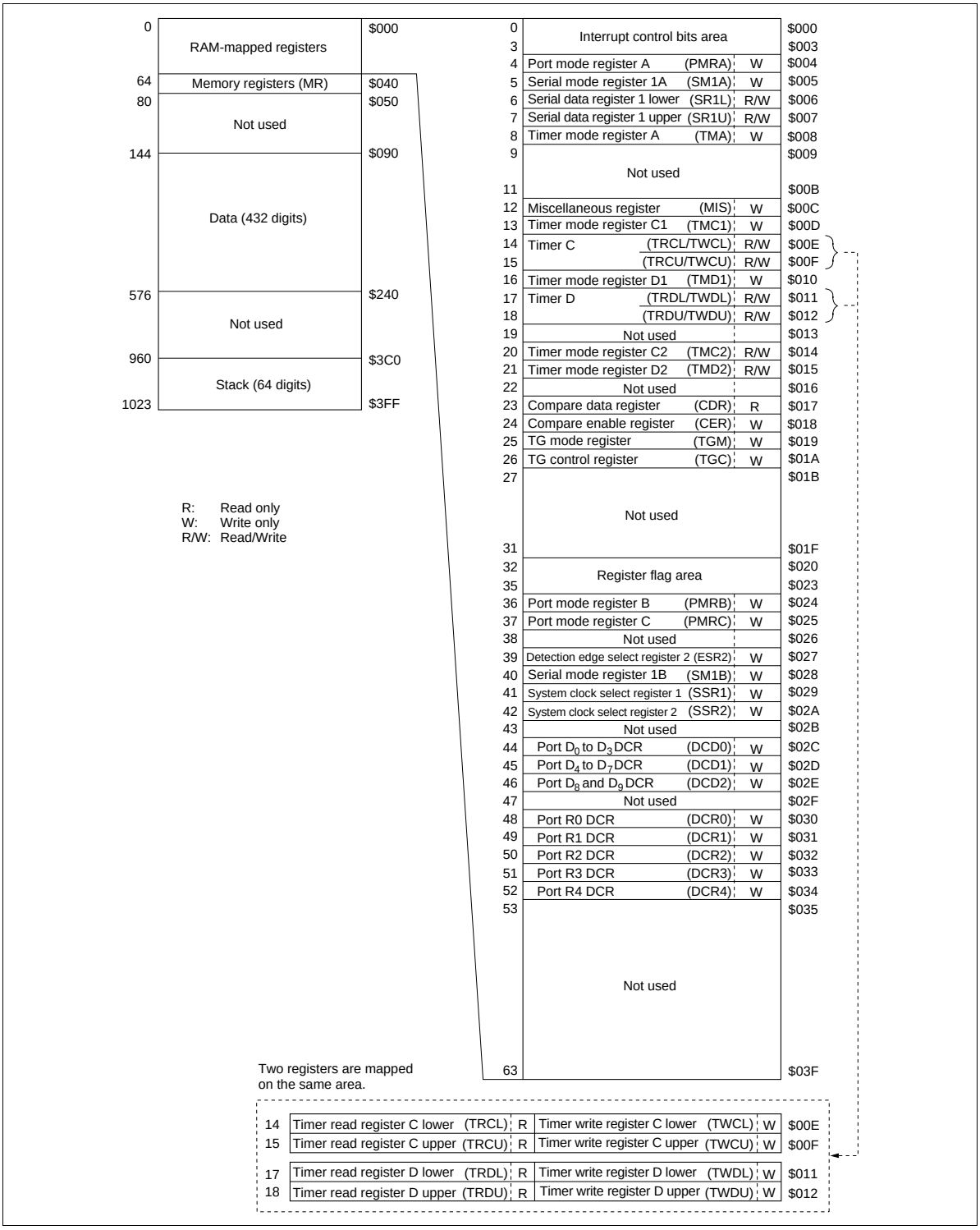



**Figure 2 RAM Memory Map**

**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. Limitations on using the instructions are shown in figure 4.

- Special Function Register Area ($004–$01A, $024–$034)

  This area is used as mode registers and data registers for external interrupts, serial interface 1, timer/counters, and the comparator, and as data control registers for I/O ports. The structure is shown in 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 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. Limitations on using the instructions are shown in 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). The structure is shown in figure 6.

**Data Area ($090–$23F):** 432 digits from $090 to $23F.

**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. The data to be saved and the save conditions are shown in figure 6.

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 is used for data storage.

| | 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 | IMTA (IM of timer A) | IFTA (IF of timer A) | IM1 (IM of $\overline{INT_1}$) | IF1 (IF of $\overline{INT_1}$) | $001 |
| 2 | IMTC (IM of timer C) | IFTC (IF of timer C) | Not used | Not used | $002 |
| 3 | IMS1 (IM of serial interface 1) | IFS1 (IF of serial interface 1) | IMTD (IM of timer D) | IFTD (IF of timer D) | $003 |

Interrupt control bits area

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| 32 | Not used | Not used | WDON (Watchdog on flag) | Not used | $020 |
| 33 | RAME (RAM enable flag) | Not used | ICEF (Input capture error flag) | ICSF (Input capture status flag) | $021 |

Register flag area

IF: Interrupt request flag
IM: Interrupt mask
IE: Interrupt enable flag
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 | | | |
| IF | Not executed | Allowed | Allowed |
| ICSF | | | |
| ICEF | | | |
| RAME | | | |
| RSP | Not executed | Allowed | Inhibited |
| WDON | Allowed | Not executed | Inhibited |
| Not used | Not executed | Not executed | Inhibited |

Note: WDON is reset by MCU reset or by $\overline{STOPC}$ enable for stop mode cancellation.
If the TM or TDM instruction is executed for the inhibited bits or non-existing bits, the value in ST becomes invalid.

**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 | | | $R4_2/SI_1$ | $R4_3/SO_1$ |
| SM1A $005 | $R4_1/\overline{SCK}_1$ | Serial transmit clock speed selection 1 | | |
| SR1L $006 | Serial data register 1 (lower digit) | | | |
| SR1U $007 | Serial data register 1 (upper digit) | | | |
| TMA $008 | | Clock source selection (timer A) | | |
| | | | | |
| MIS $00C | *2 | $SO_1$ PMOS control | Interrupt frame period selection | |
| TMC1 $00D | *1 | Clock source selection (timer C) | | |
| TRCL/TWCL $00E | Timer C register (lower digit) | | | |
| TRCU/TWCU $00F | Timer C register (upper digit) | | | |
| TMD1 $010 | *1 | Clock source selection (timer D) | | |
| TRDL/TWDL $011 | Timer D register (lower digit) | | | |
| TRDU/TWDU $012 | Timer D register (upper digit) | | | |
| $013 | | | | |
| TMC2 $014 | | Timer-C output mode selection | | |
| TMD2 $015 | *3 | Timer-D output mode selection | | |
| $016 | | | | |
| CDR $017 | Result of each analog input comparison | | | |
| CER $018 | *4 | | *5 | |
| TGM $019 | TONEC output frequency | | TONER output frequency | |
| TGC $01A | *6 | *7 | DTMF enable | |
| | | | | |
| $020 | Register flag area | | | |
| $023 | | | | |
| PMRB $024 | | | | $R0_0/\overline{INT}_1$ |
| PMRC $025 | $D_{13}/\overline{INT}_0$ | $D_{12}/\overline{STOPC}$ | $R4_0$/EVND | |
| $026 | | | | |
| ESR2 $027 | EVND detection edge selection | | | |
| SM1B $028 | | | *8 | *9 |
| SSR1 $029 | | | System clock selection | |
| SSR2 $02A | | *10 | | |
| | | | | |
| DCD0 $02C | Port $D_3$ DCR | Port $D_2$ DCR | Port $D_1$ DCR | Port $D_0$ DCR |
| DCD1 $02D | Port $D_7$ DCR | Port $D_6$ DCR | Port $D_5$ DCR | Port $D_4$ DCR |
| DCD2 $02E | | | Port $D_9$ DCR | Port $D_8$ DCR |
| | | | | |
| DCR0 $030 | | | | 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 |
| | | | | |
| $03F | | | | |

: Not used

Notes:
1. Auto-reload on/off
2. Pull-up MOS control
3. Input capture selection
4. Comparator switch
5. Port/comparator selection
6. TONEC output control
7. TONER output control
8. $SO_1$ output level control in idle states
9. Serial clock source selection 1
10. System clock selection

**Figure 5 Special Function Register Area**

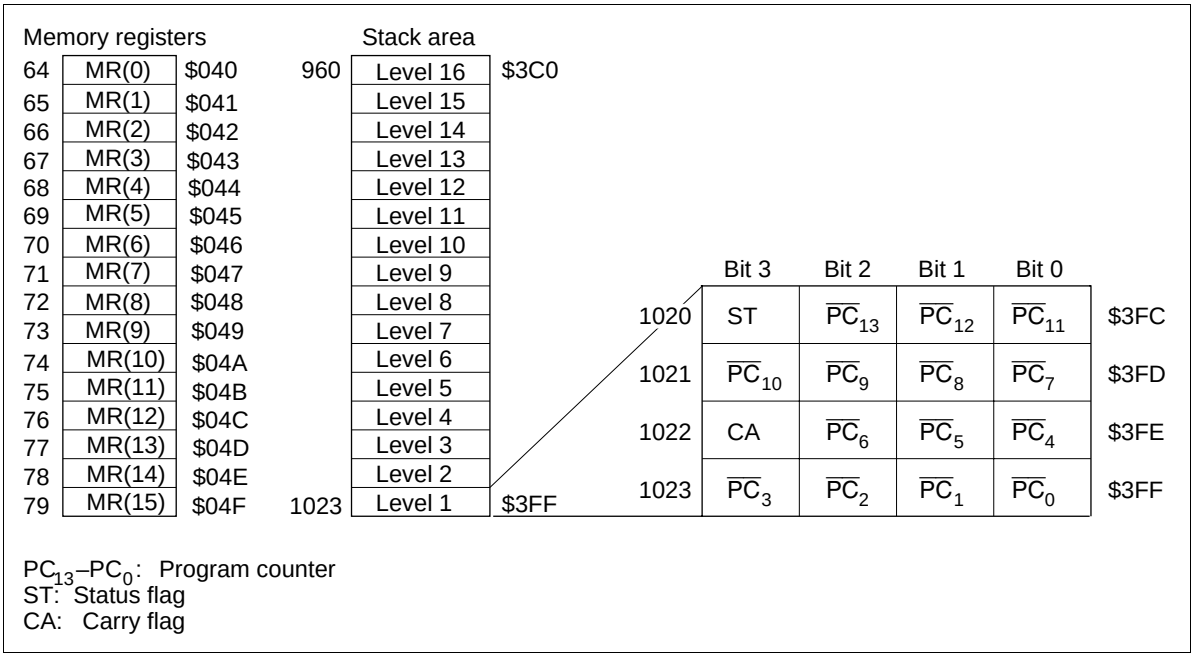



**Figure 6 Configuration of Memory Registers and Stack Area, and Stack Position**

## Functional Description

### Registers and Flags

The MCU has nine registers and two flags for CPU operations. They are shown in figure 7 and described below.

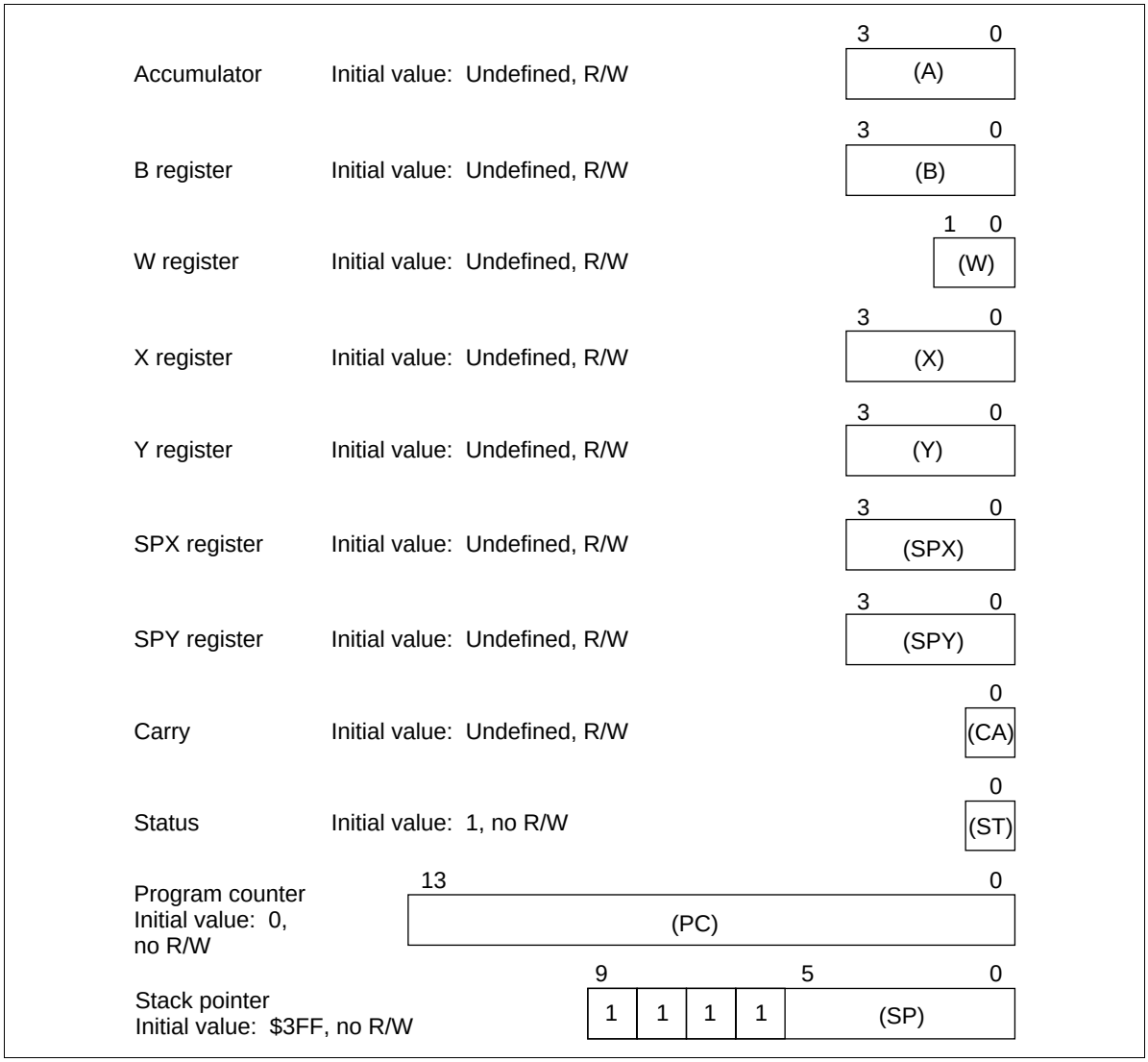



**Figure 7 Registers and Flags**

**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 in another way: by resetting the RSP bit with the REM or REMD instruction.

### Reset

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

Initial values after MCU reset are listed in table 1.

### Interrupts

The MCU has 6 interrupt sources: Two external signals ($\overline{\text{INT}}_0$, $\overline{\text{INT}}_1$), three timer/counters (timers A, C, and D), and one serial interface (serial 1).

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.

**Interrupt Control Bits and Interrupt Processing:** Locations $000 to $003 and $020 to $021 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.

A block diagram of the interrupt control circuit is shown in figure 8, interrupt priorities and vector addresses are listed in table 2, and interrupt processing conditions for the 11 interrupt sources are listed in table 3.

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.

The interrupt processing sequence is shown in figure 9 and an interrupt processing flowchart is shown in figure 10. 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–DCD2) | All bits 0 | Turns output buffer off (to high impedance) |
| | | (DCR0–DCR4) | All bits 0 | |
| | Port mode register A | (PMRA) | - - 00 | Refer to description of port mode register A |
| | Port mode register B | (PMRB) | - - - 0 | Refer to description of port mode register B |
| | Port mode register C bits 3, 1, 0 | (PMRC3, PMRC1, PMRC0) | 000 - | Refer to description of port mode register C |
| | Detection edge select register 2 | (ESR2) | 00 - - | Disables edge detection |
| Timer/counters, serial interface | Timer mode register A | (TMA) | - 000 | Refer to description of timer mode register A |
| | 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 1A | (SM1A) | 0000 | Refer to description of serial mode register 1A |
| | Serial mode register 1B | (SM1B) | - - X0 | Refer to description of serial mode register 1B |
| | Prescaler S | (PSS) | $000 | — |
| | Timer counter A | (TCA) | $00 | — |

| Item | | Abbr. | Initial Value | Contents |
|---|---|---|---|---|
| Timer/counters, serial interface | Timer counter C | (TCC) | $00 | — |
| | Timer counter D | (TCD) | $00 | — |
| | Timer write register C | (TWCU, TWCL) | $X0 | — |
| | Timer write register D | (TWDU, TWDL) | $X0 | — |
| | Octal counter | | 000 | — |
| Comparator | Compare enable register | (CER) | 0 - 00 | Refer to description of voltage comparator |
| Bit register | Watchdog timer on flag | (WDON) | 0 | Refer to description of timer C |
| | 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) | 00 - - | Refer to description of operating modes, and oscillator circuit |
| | System clock select register 1 bits 1, 0 | (SSR11, SSR10) | 00 | Refer to description of operating modes, and oscillator circuit |
| | System clock select register 2 | (SSR2) | - 0 - - | |

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 1 bit 2 | (PMRC12) | Pre-stop-mode values are retained | 0 | 0 |
| System clock select register 1 bit 3 | (SSR13) | | | |

**Table 2 Vector Addresses and Interrupt Priorities**

| Reset/Interrupt | Priority | Vector Address |
|---|---|---|
| $\overline{\text{RESET}}$, $\overline{\text{STOPC}}$* | — | $0000 |
| $\overline{\text{INT}}_0$ | 1 | $0002 |
| $\overline{\text{INT}}_1$ | 2 | $0004 |
| Timer A | 3 | $0006 |
| Not used | 4 | $0008 |
| Timer C | 5 | $000A |
| Timer D | 6 | $000C |
| Serial 1 | 7 | $000E |

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

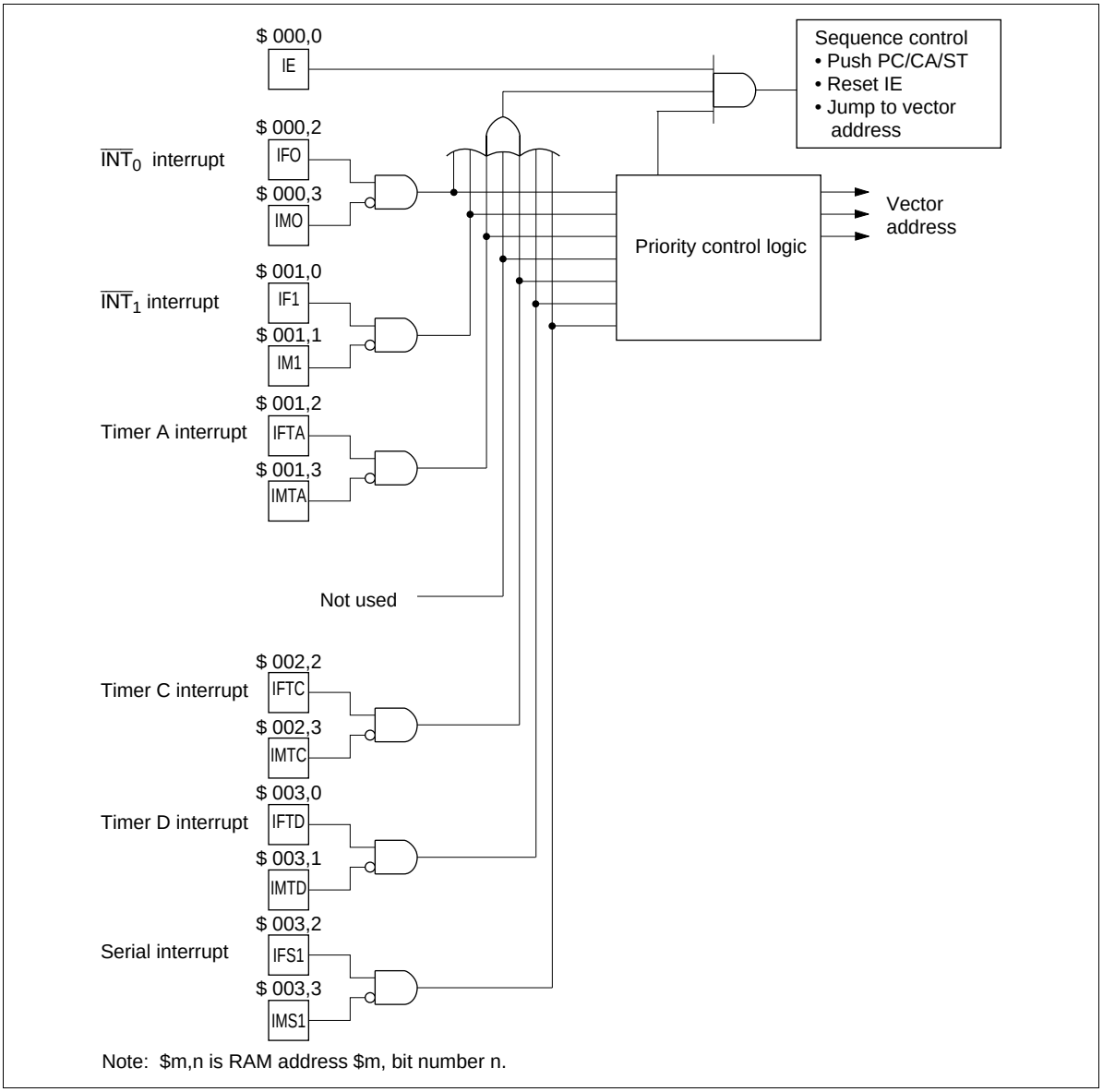
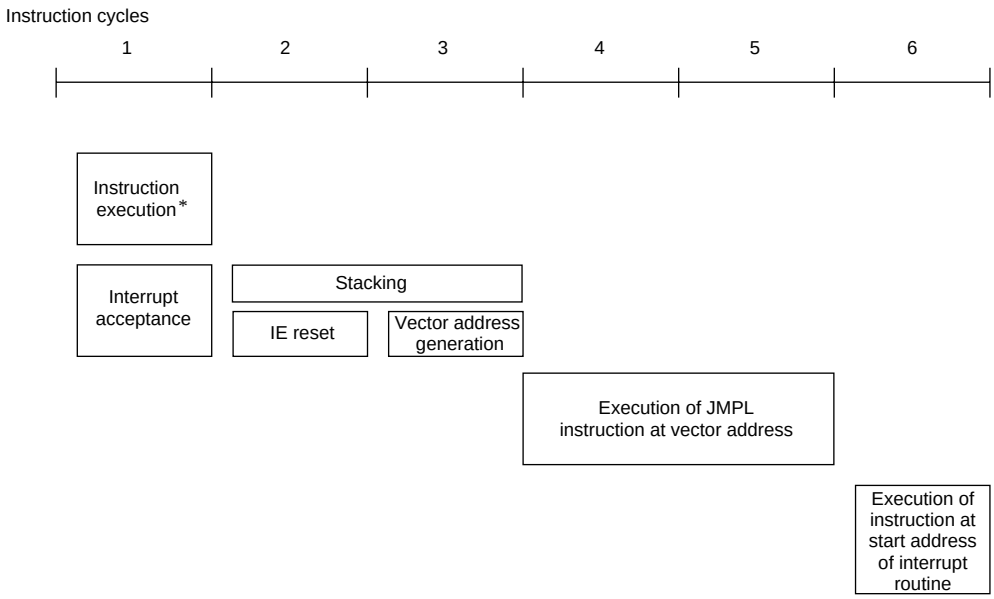



**Figure 8 Interrupt Control Circuit**

**Table 3 Interrupt Processing and Activation Conditions**

| | Interrupt Source | | | | | |
|---|---|---|---|---|---|---|
| **Interrupt Control Bit** | **$\overline{INT_0}$** | **$\overline{INT_1}$** | **Timer A** | **Timer C** | **Timer D** | **Serial 1** |
| IE | 1 | 1 | 1 | 1 | 1 | 1 |
| IF0 · $\overline{IM0}$ | 1 | 0 | 0 | 0 | 0 | 0 |
| IF1 · $\overline{IM1}$ | * | 1 | 0 | 0 | 0 | 0 |
| IFTA · $\overline{IMTA}$ | * | * | 1 | 0 | 0 | 0 |
| IFTC · $\overline{IMTC}$ | * | * | * | 1 | 0 | 0 |
| IFTD · $\overline{IMTD}$ | * | * | * | * | 1 | 0 |
| IFS1 · $\overline{IMS1}$ | * | * | * | * | * | 1 |

Note: * Can be either 0 or 1. Their values have no effect on operation.




Note: * The stack is accessed and the IE reset after the instruction is executed, even if it is a two-cycle instruction.

**Figure 9 Interrupt Processing Sequence**

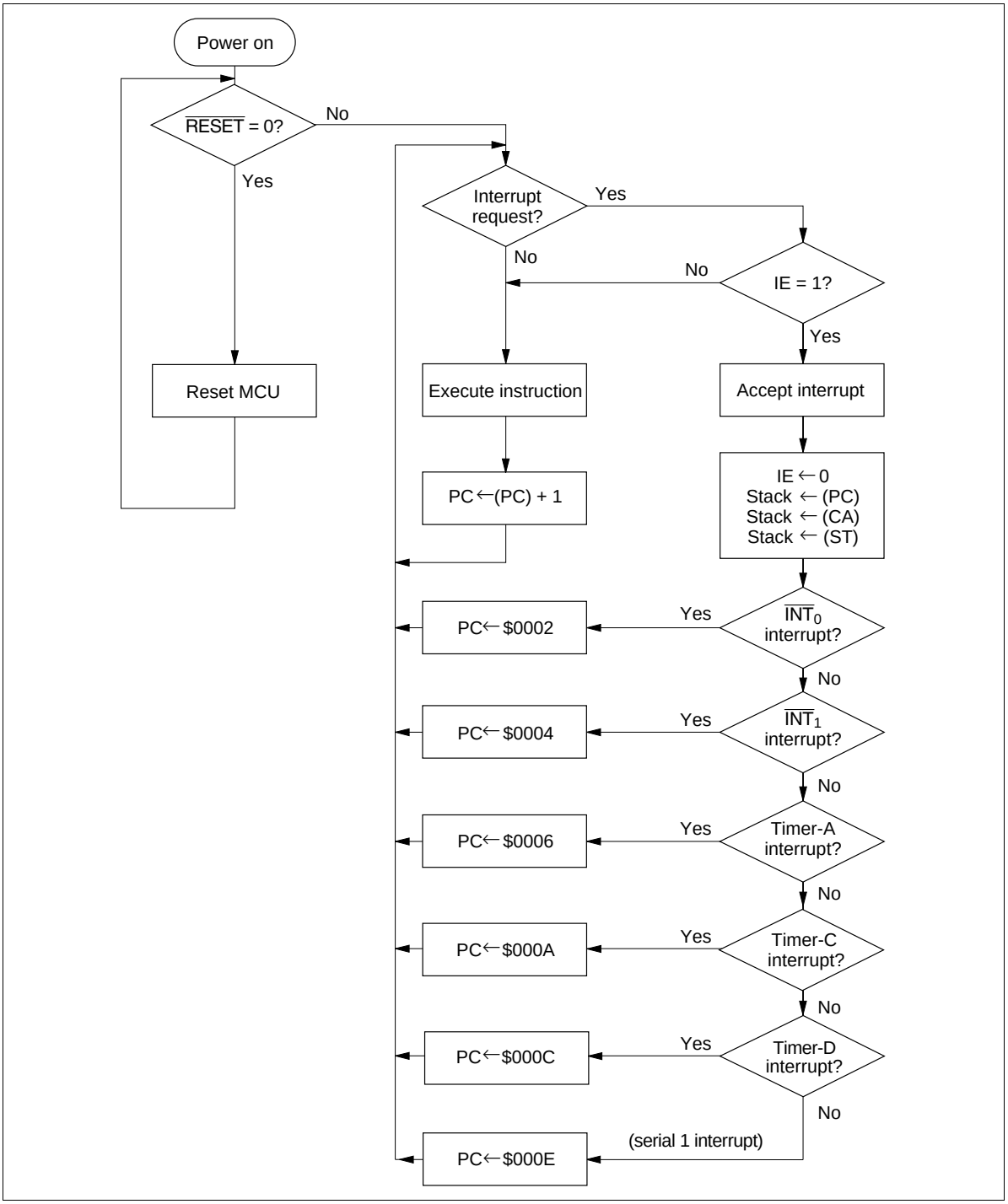



**Figure 10 Interrupt Processing Flowchart**

**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, as listed in table 4.

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

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

**External Interrupts ($\overline{INT}_0$, $\overline{INT}_1$):** Two external interrupt signals.

**External Interrupt Request Flags (IF0, IF1: $000, $001):** IF0 and IF1 are set at the falling edge of signals input to $\overline{INT}_0$ and $\overline{INT}_1$ as listed in table 5.

**Table 5 External Interrupt Request Flags (IF0, IF1: $000, $001)**

| IF0, IF1 | Interrupt Request |
|---|---|
| 0 | No |
| 1 | Yes |

**External Interrupt Masks (IM0, IM1: $000, $001):** Prevent (mask) interrupt requests caused by the corresponding external interrupt request flags, as listed in table 6.

**Table 6 External Interrupt Masks (IM0, IM1: $000, $001)**

| IM0, IM1 | Interrupt Request |
|---|---|
| 0 | Enabled |
| 1 | Disabled (Masked) |

**Timer A Interrupt Request Flag (IFTA: $001, Bit 2):** Set by overflow output from timer A, as listed in table 7.

**Table 7 Timer A Interrupt Request Flag (IFTA: $001, Bit 2)**

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

**Timer A Interrupt Mask (IMTA: $001, Bit 3):** Prevents (masks) an interrupt request caused by the timer A interrupt request flag, as listed in table 8.

**Table 8 Timer A Interrupt Mask (IMTA: $001, Bit 3)**

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

**Timer C Interrupt Request Flag (IFTC: $002, Bit 2):** Set by overflow output from timer C, as listed in table 9.

**Table 9 Timer C Interrupt Request Flag (IFTC: $002, Bit 2)**

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

**Timer C Interrupt Mask (IMTC: $002, Bit 3):** Prevents (masks) an interrupt request caused by the timer C interrupt request flag, as listed in table 10.

**Table 10 Timer C Interrupt Mask (IMTC: $002, Bit 3)**

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

**Timer D Interrupt Request Flag (IFTD: $003, Bit 0):** 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, as listed in table 11.

**Table 11 Timer D Interrupt Request Flag (IFTD: $003, Bit 0)**

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

**Timer D Interrupt Mask (IMTD: $003, Bit 1):** Prevents (masks) an interrupt request caused by the timer D interrupt request flag, as listed in table 12.

**Table 12 Timer D Interrupt Mask (IMTD: $003, Bit 1)**

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

**Serial Interrupt Request Flags (IFS1: $003, Bit 2):** Set when data transfer is completed or when data transfer is suspended, as listed in table 13.

**Table 13 Serial Interrupt Request Flag (IFS1: $003, Bit 2)**

| IFS1 | Interrupt Request |
|---|---|
| 0 | No |
| 1 | Yes |

**Serial Interrupt Masks (IMS1: $003, Bit 3):** Prevents (masks) an interrupt request caused by the serial interrupt request flag, as listed in table 14.

**Table 14 Serial Interrupt Mask (IMS1: $003, Bit 3)**

| IMS1 | Interrupt Request |
|---|---|
| 0 | Enabled |
| 1 | Disabled (Masked) |

## Operating Modes

The MCU has three operating modes as shown in table 15. The operations in each mode are listed in tables 16 and 17. Transitions between operating modes are shown in figure 11.

**Table 15 Operating Modes and Clock Status**

| | | Mode Name | | |
|---|---|---|---|---|
| | | **Active** | **Standby** | **Stop** |
| Activation method | | $\overline{\text{RESET}}$ cancellation, interrupt request, $\overline{\text{STOPC}}$ cancellation in stop mode | SBY instruction | STOP instruction |
| Status | System oscillator | OP | OP | Stopped |
| Cancellation method | | $\overline{\text{RESET}}$ input, STOP/SBY instruction | $\overline{\text{RESET}}$ input, interrupt request | $\overline{\text{RESET}}$ input, $\overline{\text{STOPC}}$ input in stop mode |

Note: OP implies in operation

**Table 16 Operations in Low-Power Dissipation Modes**

| Function | Stop Mode | Standby Mode |
|---|---|---|
| CPU | Reset | Retained |
| RAM | Retained | Retained |
| Timer A | Reset | OP |
| Timer C | Reset | OP |
| Timer D | Reset | OP |
| Serial 1 | Reset | OP |
| DTMF | Reset | OP |
| Comparator | Reset | Stopped |
| I/O | Reset* | Retained |

Notes: OP implies in operation
* Output pins are at high impedance.

**Table 17 I/O Status in Low-Power Dissipation Modes**

| | Output | | Input |
|---|---|---|---|
| | **Standby Mode** | **Stop Mode** | **Active Mode** |
| $D_0$–$D_9$ | Retained | High impedance | Input enabled |
| $D_{12}$–$D_{13}$, $RD_0$, $RD_1$, $RE_0$ | — | — | Input enabled |
| R0–R4 | Retained or output of peripheral functions | High impedance | Input enabled |

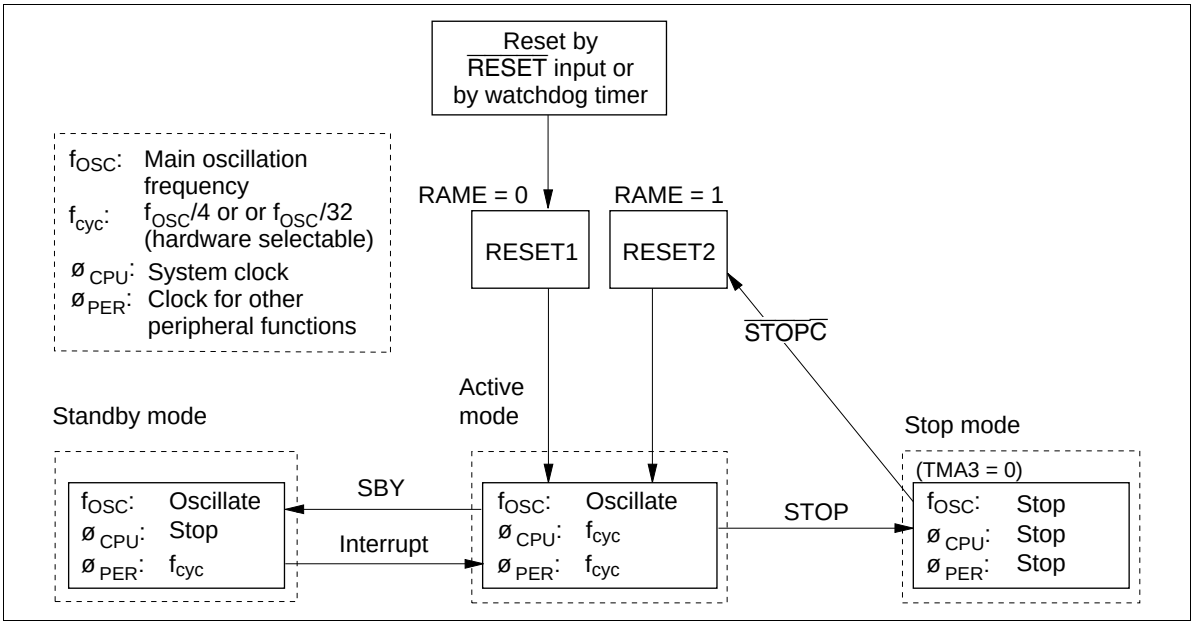



**Figure 11 MCU Status Transitions**

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

**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 because the CPU stops.

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

Standby mode is terminated by a $\overline{RESET}$ input or an interrupt request. If it is terminated by $\overline{RESET}$ input, 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. A flowchart of operation in standby mode is shown in figure 12.

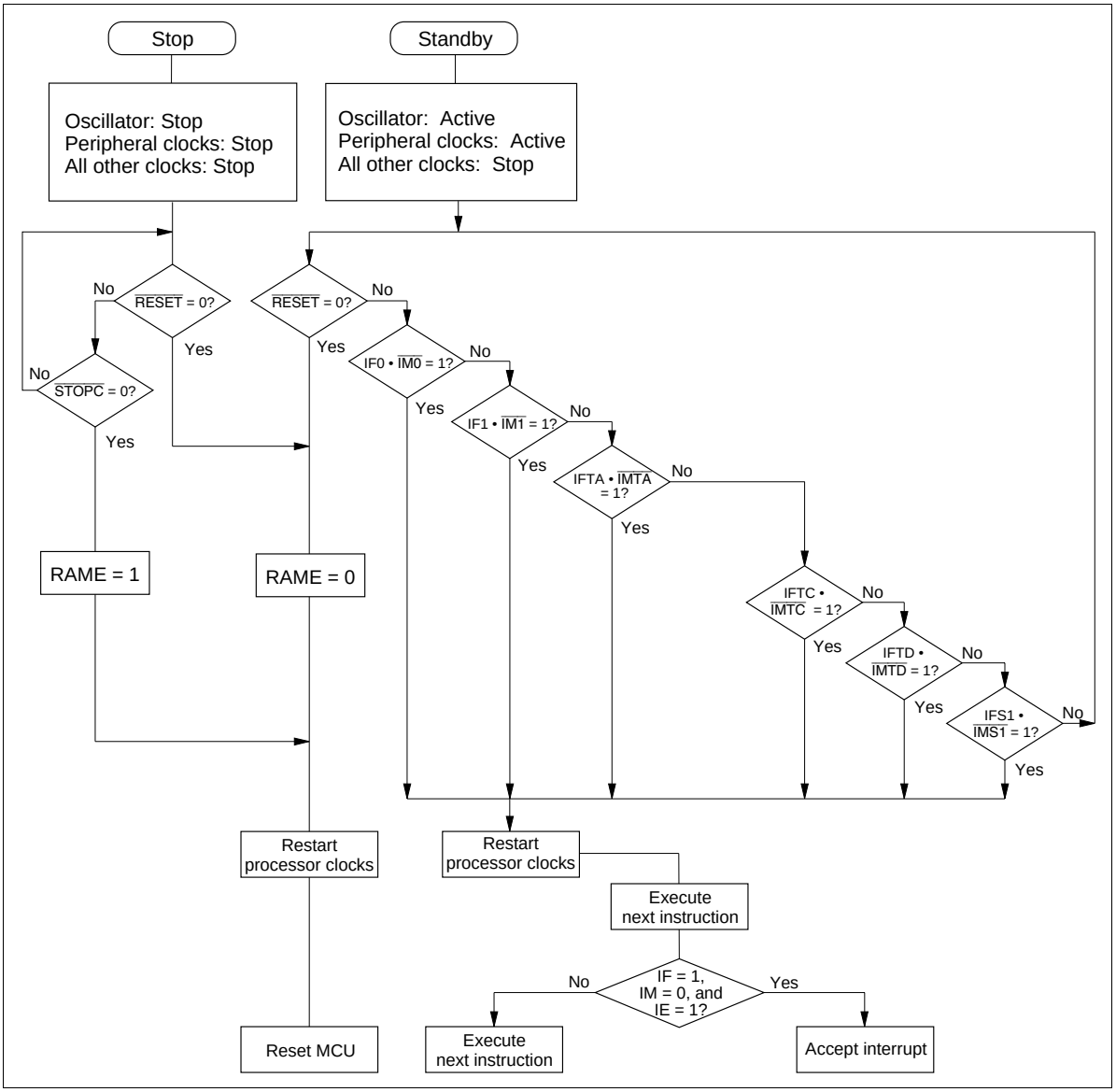



**Figure 12 MCU Operation Flowchart**

**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. The MCU enters stop mode if the STOP instruction is executed in active mode.

Stop mode is terminated by a $\overline{RESET}$ input or a $\overline{STOPC}$ input as shown in figure 13. $\overline{RESET}$ 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.

**Stop Mode Cancellation by $\overline{\text{STOPC}}$:** The MCU enters active mode from stop mode by inputting $\overline{\text{STOPC}}$ as well as by $\overline{\text{RESET}}$. 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 $\overline{\text{RESET}}$. When stop mode is cancelled by $\overline{\text{RESET}}$, RAME = 0; when cancelled by $\overline{\text{STOPC}}$, RAME = 1. $\overline{\text{RESET}}$ 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 $\overline{\text{STOPC}}$ (for example, 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.

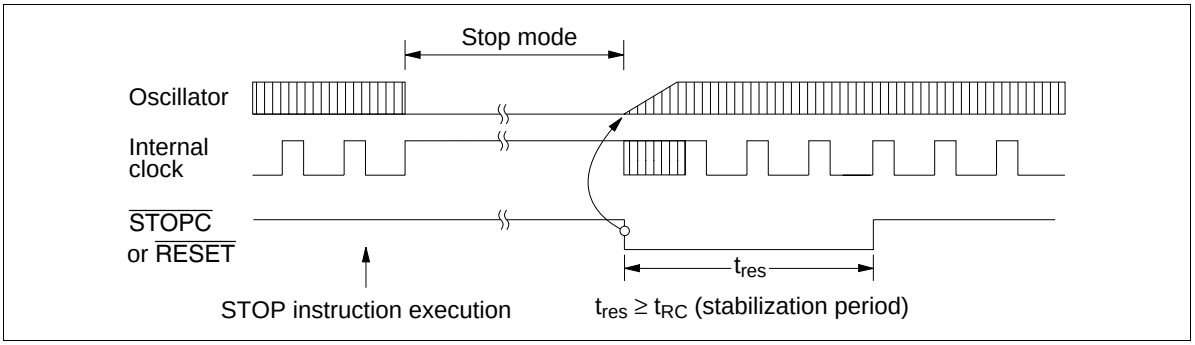



**Figure 13 Timing of Stop Mode Cancellation**

**MCU Operation Sequence:** The MCU operates in the sequences shown in figures 14 to 16. It is reset by an asynchronous $\overline{\text{RESET}}$ input, regardless of its status.

The low-power mode operation sequence is shown in figure 16. 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.

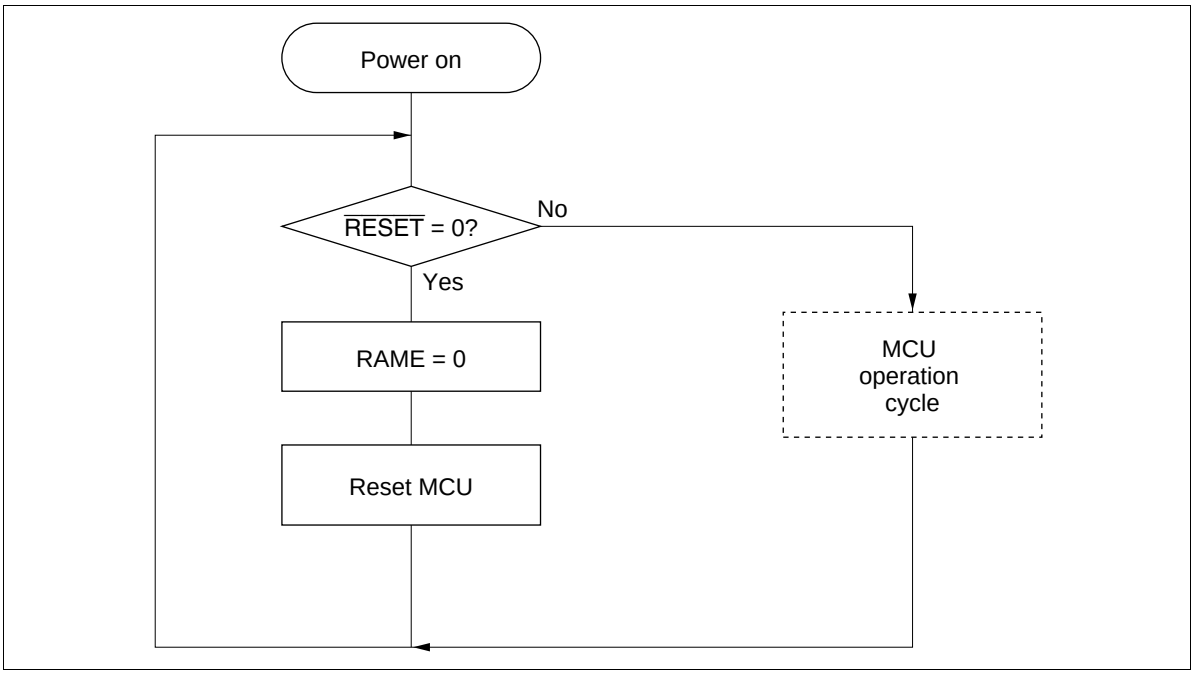



**Figure 14 MCU Operating Sequence (power on)**

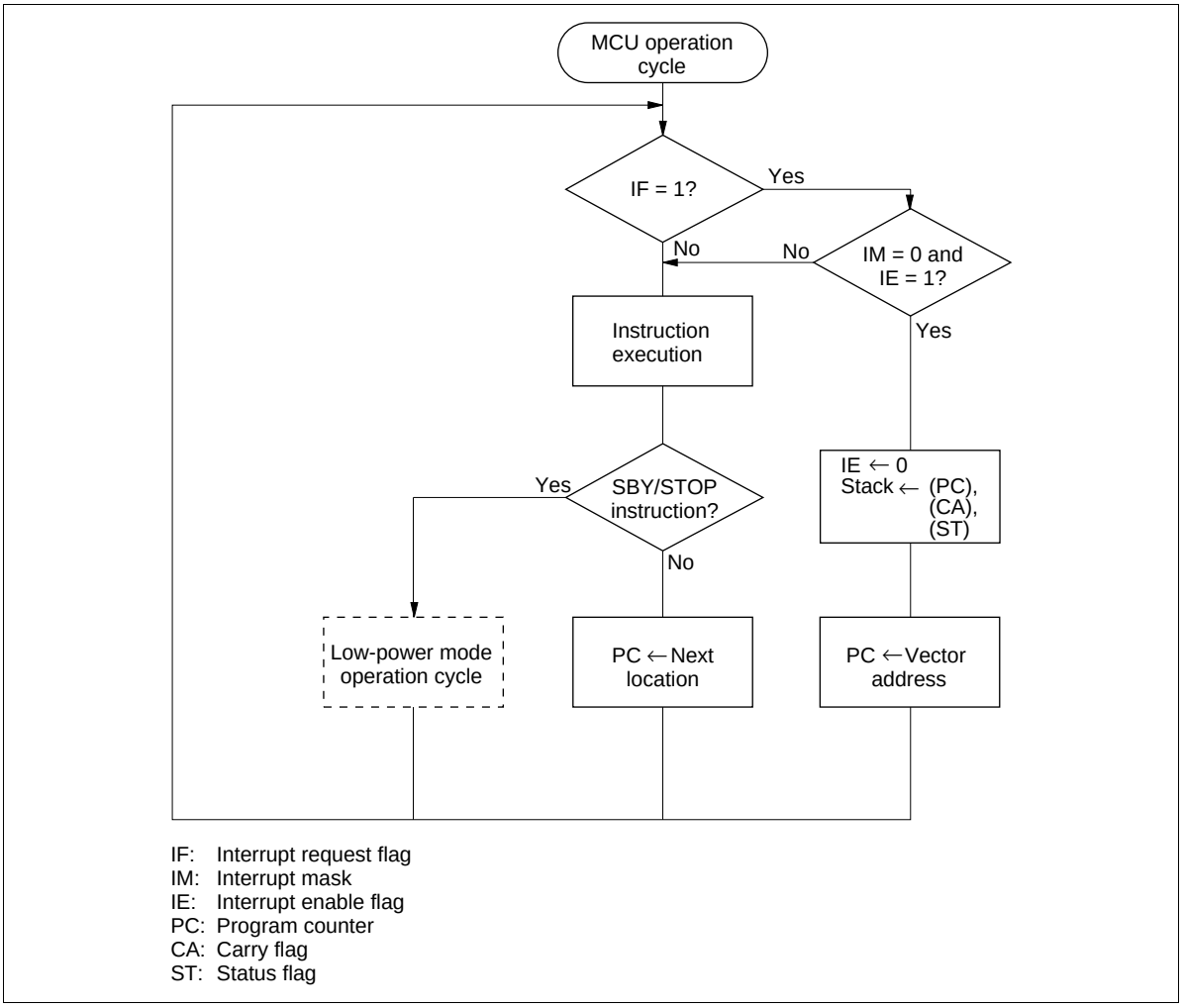
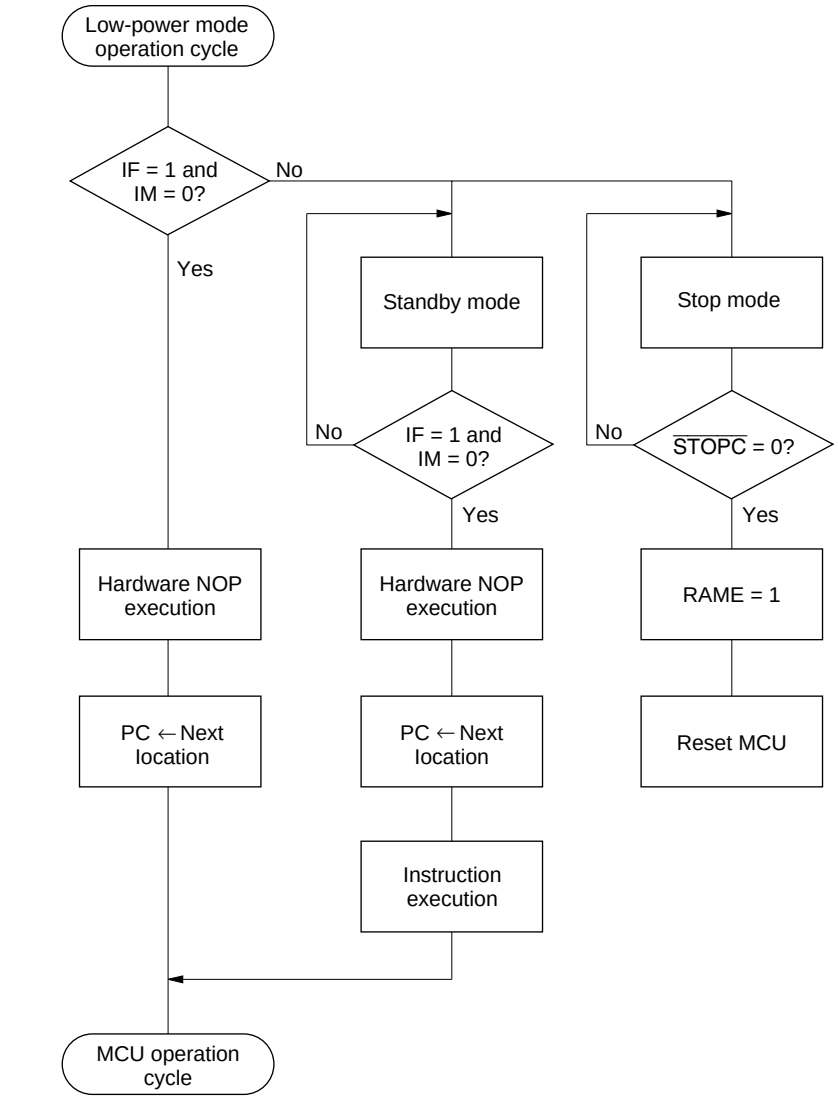



**Figure 15 MCU Operating Sequence (MCU operation cycle)**




For IF and IM operation, refer to figure 12.

**Figure 16 MCU Operating Sequence (Low-Power Mode Operation)**

## Internal Oscillator Circuit

A block diagram of the clock generation circuit is shown in figure 17. As shown in table 18, a ceramic oscillator or crystal oscillator can be connected to $OSC_1$ and $OSC_2$. The system oscillator can also be operated by an external clock. Bit 1 (SSR11) of system clock select register 1 (SSR1: $029) and bit 2 (SSR22) of system clock select register 2 (SSR2: $02A) must be selected according to the frequency of the oscillator connected to $OSC_1$ and $OSC_2$ (figure 18).

Note: If the SSR10, SSR11 and SSR22 setting does not match the oscillator frequency, the DTMF generator will malfunction.

After $\overline{RESET}$ input or after stop mode has been cancelled, the division ratio of the system clock can be selected as 1/4 or 1/32 by setting the SEL pin level.

- 1/4 division ratio: Connect SEL to $V_{CC}$.
- 1/32 division ratio: Connect SEL to GND.

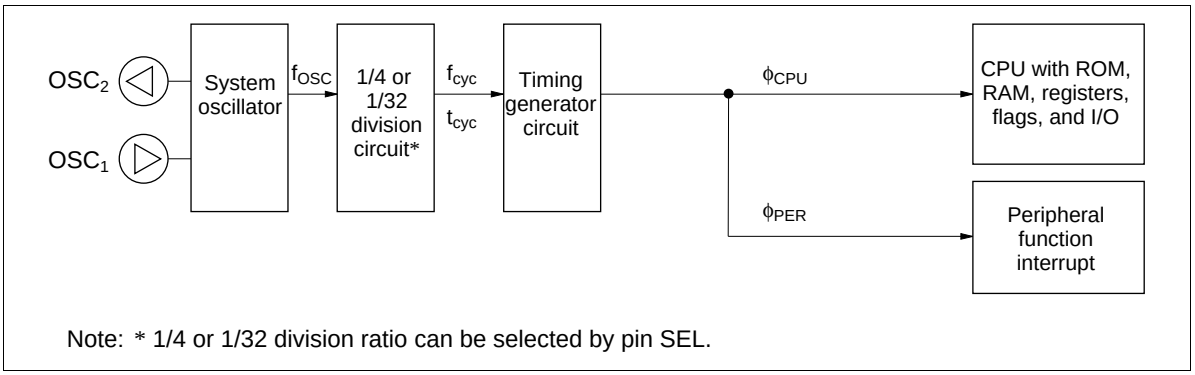



**Figure 17 Clock Generation Circuit**

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

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | 0 | 0 |
| Read/Write | — | — | W | W |
| Bit name | Not used | Not used | SSR11 | SSR10 |

| SSR22 | SSR11 | SSR10 | System clock selection |
|---|---|---|---|
| 0 | 0 | 0 | 400 kHz |
| 0 | 0 | 1 | 800 kHz |
| 0 | 1 | 0 | 2 MHz |
| 0 | 1 | 1 | 4 MHz |
| 1 | × | × | 3.58 MHz |

×: Don't care

**Figure 18 System Clock Select Register 1**

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

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | 0 | — | — |
| Read/Write | — | W | — | — |
| Bit name | Not used | SSR22 | Not used | Not used |

| SSR22 | System clock selection |
|---|---|
| 0 | Selected from 400 kHz, 800 kHz, 2 MHz, 4 MHz * |
| 1 | 3.58 MHz |

Note: * Refer to system clock select register 1 (SSR1) of figure 18.

**Figure 19 System Clock Select Register 2**

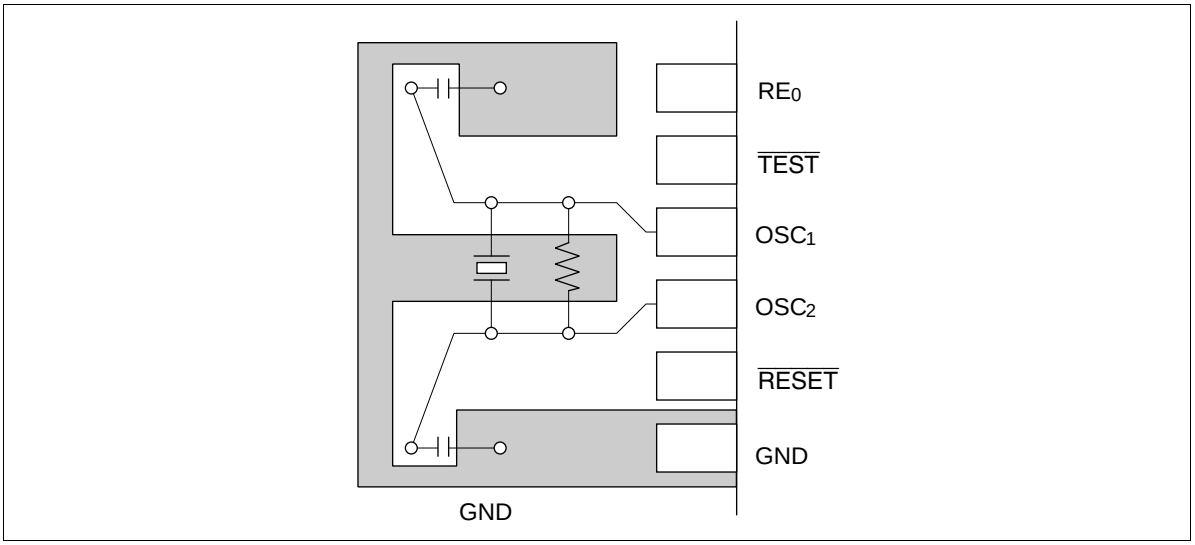
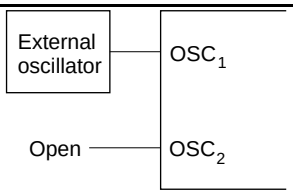
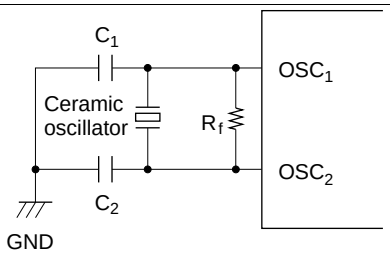



**Figure 20 Typical Layout of Crystal and Ceramic Oscillators**

**Table 18 Oscillator Circuit Examples**

| Circuit Configuration | | Circuit Constants |
|---|---|---|
| External clock operation |  | |
| Ceramic oscillator ($OSC_1$, $OSC_2$) |  | Ceramic oscillator: CSB400P22 (Murata), CSB400P (Murata)<br>$R_f$ = 1 MΩ ± 20%<br>$C_1$ = $C_2$ = 220 pF ± 5% |
| | | Ceramic oscillator: CSB800J122 (Murata), CSB800J (Murata)<br>$R_f$ = 1 MΩ ± 20%<br>$C_1$ = $C_2$ = 220 pF ± 5% |
| | | Ceramic oscillator: CSA2.00MG (Murata)<br>$R_f$ = 1 MΩ ± 20%<br>$C_1$ = $C_2$ = 30 pF ± 20% |
| | | Ceramic oscillator: CSA4.00MG (Murata)<br>$R_f$ = 1 MΩ ± 20%<br>$C_1$ = $C_2$ = 30 pF ± 20% |
| | | Ceramic oscillator: CSA3.58MG (Murata)<br>$R_f$ = 1 MΩ ± 20%<br>$C_1$ = $C_2$ = 30 pF ± 20% |

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

2. Wiring among $OSC_1$, $OSC_2$, and elements should be as short as possible, and must not cross other wiring (see figure 20).

## Input/Output

The MCU has 27 input/output pins ($D_0$–$D_9$, $R0_0$–$R4_3$) and 5 input pins ($D_{12}$, $D_{13,}$ $RD_0$, $RD_1$, $RE_0$). The features are described below.

- A maximum current of 15 mA is allowed for each of the pins $D_4$ to $D_9$ with a total maximum current of less than 105 mA. In addition, $D_0$–$D_3$ can each act as a 10-mA maximum current source.
- Some input/output pins are multiplexed with peripheral function pins such as for the timers or serial interface. 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.
- Input or output selection for input/output pins and port or peripheral function selection for multiplexed pins are set by software.
- Peripheral function output pins are CMOS output pins. Only the $R4_3/SO_1$ pin 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 in high-impedance state.
- Pins $D_0$–$D_3$ have built-in pull-down MOSs, and other input/output pins have built-in pull-up MOSs, which can be individually turned on or off by software.

The I/O buffer configuration is shown in figure 21 and 22, programmable I/O circuits are listed in table 19, and I/O pin circuit types are shown in table 20.

**Table 19 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 |
| Pull-down MOS | | — | — | — | — | On | — | On | — |

Note: — indicates off status.

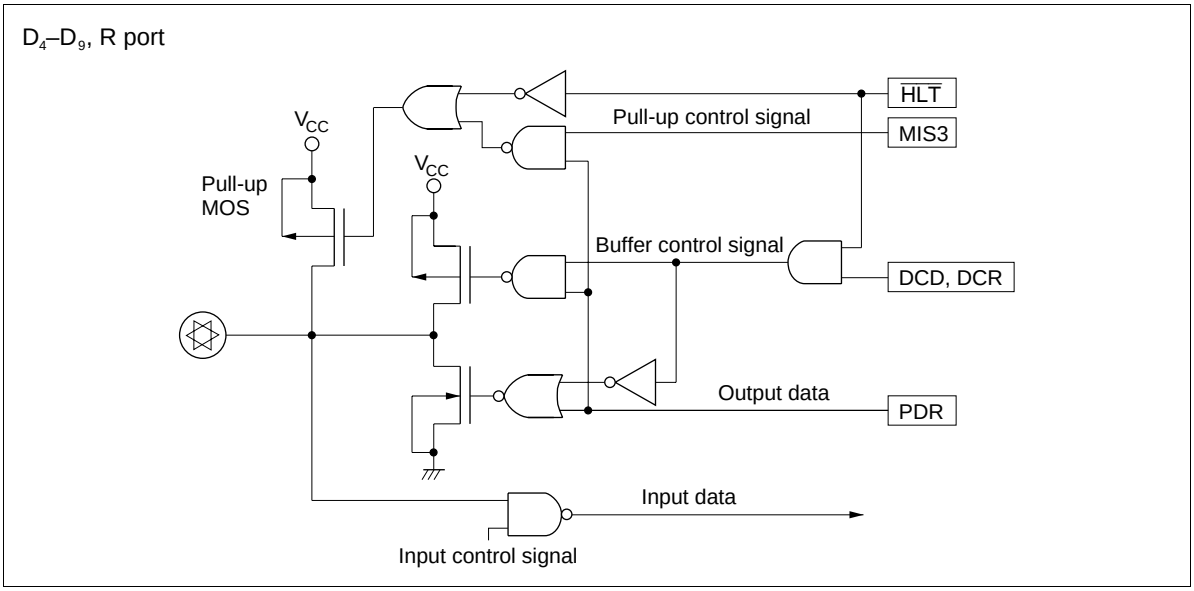



**Figure 21 I/O Buffer Configuration (with Pull-Up MOS)**

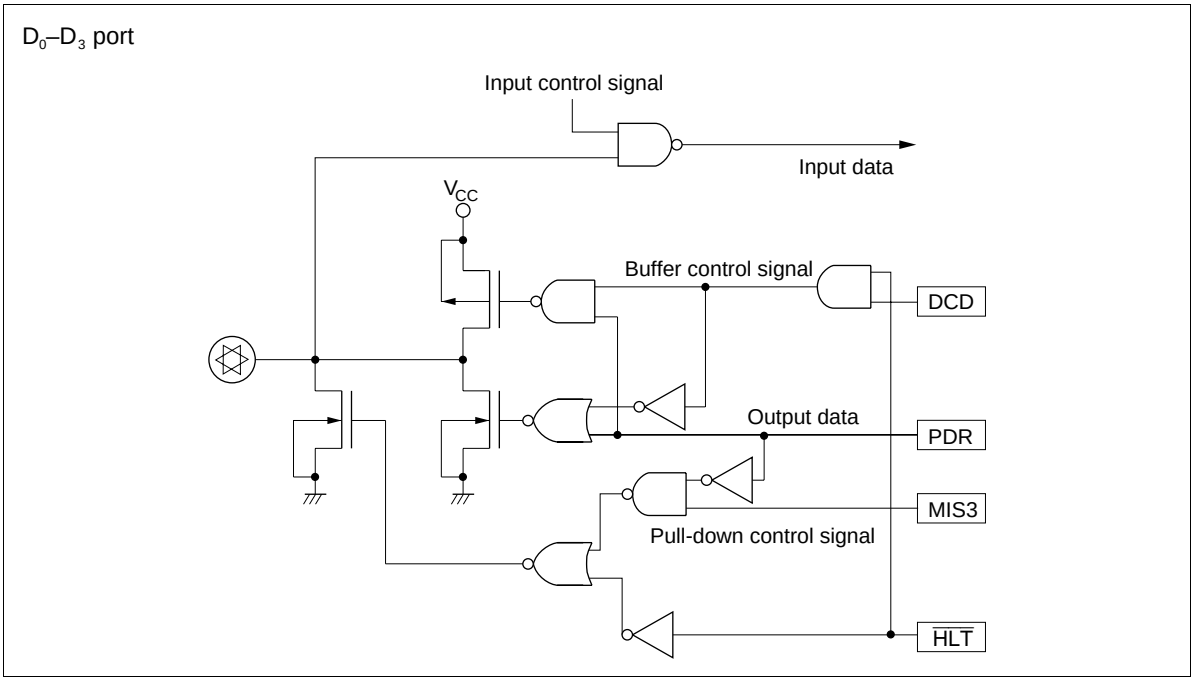
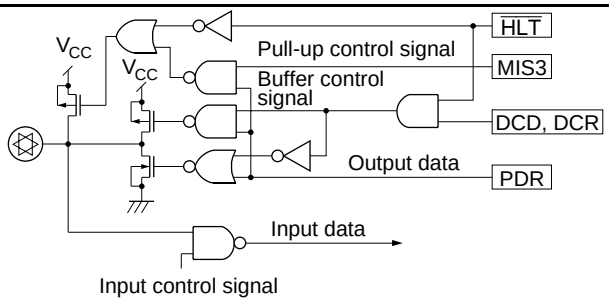
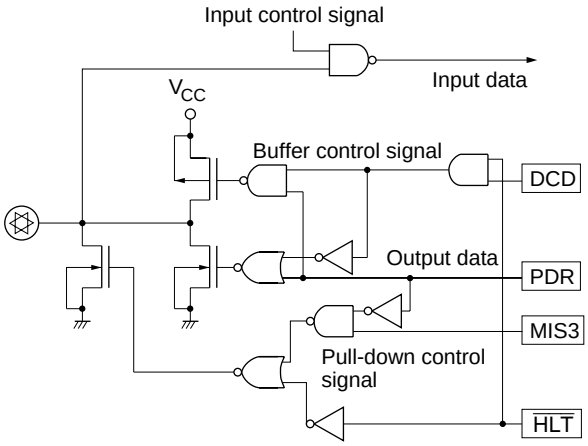
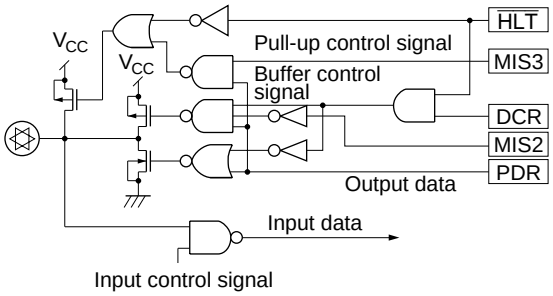
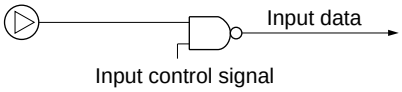
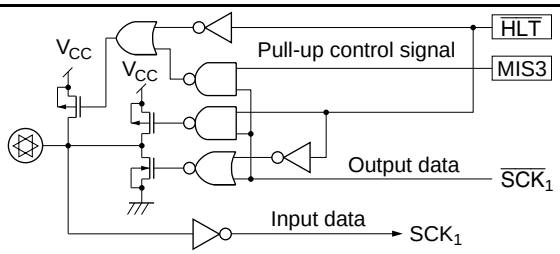
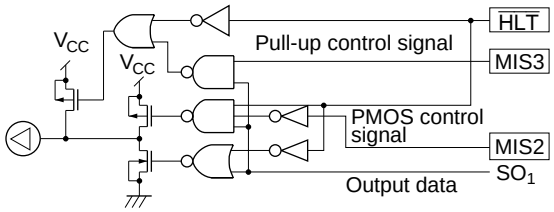
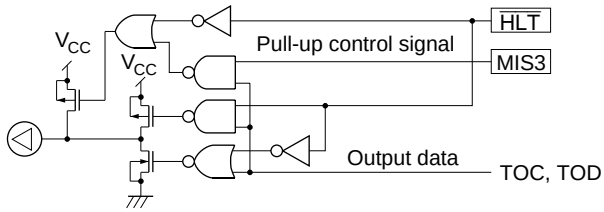
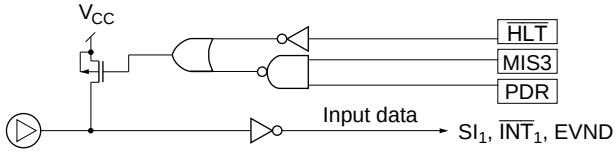
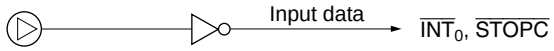



**Figure 22 I/O Buffer Configuration (with Pull-Down MOS)**

**Table 20-1 Circuit Configurations of I/O Pins**

| I/O Pin Type | Circuit | Pins |
|---|---|---|
| Input/output pins |  | $D_4$–$D_9$, $R0_0$, $R1_0$–$R1_3$, $R2_0$–$R2_3$ $R3_0$–$R3_3$, $R4_0$–$R4_2$ |
| |  | $D_0$–$D_3$ |
| |  | $R4_3$ |
| Input pins |  | $D_{12}$, $D_{13}$ $RD_0$, $RD_1$, $RE_0$ |

| I/O Pin Type | | Circuit | Pins |
|---|---|---|---|
| Peripheral function pins | Input/ output pins |  | $\overline{SCK_1}$ |
| Peripheral function pins | Output pins |  | $SO_1$ |
| | |  | TOC, TOD |
| | Input pins |  | $SI_1$, $\overline{INT_1}$, EVND |
| | |  | $\overline{INT_0}$, $\overline{STOPC}$ |

Note: The MCU is reset in stop mode, and peripheral function selection is cancelled. The $\overline{HLT}$ signal becomes low, and input/output pins enter high-impedance state.

**D Port ($D_0$–$D_{13}$):** Consist of 10 input/output pins and 2 input pins addressed by one bit. $D_0$–$D_3$ are high-current sources, $D_4$–$D_9$ are large-current sinks, and $D_{12}$ and $D_{13}$ 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 $D_0$–$D_{13}$ 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 23).

Pins $D_{12}$ and $D_{13}$ are multiplexed with peripheral function pins $\overline{STOPC}$ and $\overline{INT_0}$, respectively. The peripheral function modes of these pins are selected by bits 2 and 3 (PMRC2, PMRC3) of port mode register C (PMRC: $025) (figure 24).

**R Ports ($R0_0$, $R1_0$–$R4_3$, $RD_0$, $RD_1$, $RE_0$):** 17 input/output pins and 3 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–DCR4: $030–$034) that are mapped to memory addresses (figure 23).

Pin $R0_0$ is multiplexed with peripheral pin $\overline{INT_1}$. The peripheral function mode of this pins is selected by bit 0 (PMRB0) of port mode register B (PMRB: $024) (figure 25).

Pins $R3_1$–$R3_2$ are multiplexed with peripheral pins TOC and TOD respectively. The peripheral function modes of these pins are selected by 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 26, and 27).

Pin $R4_0$ is multiplexed with peripheral pin EVND. The peripheral function mode of this pins is selected by bit 1 (PMRC1) of port mode register C (PMRC: $025) (figure 24).

Pins $R4_1$–$R4_3$ are multiplexed with peripheral pins $\overline{SCK_1}$, $SI_1$, and $SO_1$, respectively. The peripheral function modes of these pins are selected by bit 3 (SM1A3) of serial mode register 1A (SM1A: $005), and bits 0 and 1 (PMRA0, PMRA1) of port mode register A (PMRA: $004), as shown in figures 28 and 29.

Ports $RD_0$ and $RD_1$ are multiplexed with peripheral function pins $COMP_0$ and $COMP_1$, respectively. The function modes of these pins are selected by bit 3 (CER3) of the compare enable register (CER: $018), as shown in figure 30.

Port $RE_0$ is multiplexed with peripheral function pin $VC_{ref}$. While functioning as $VC_{ref}$, do not use this pin as an R port at the same time, otherwise, the MCU may malfunction.

**Pull-Up or Pull-Down MOS Transistor Control:** A program-controlled pull-up or pull-down MOS transistor is provided for each input/output pin other than input-only pins $D_{12}$ and $D_{13}$. 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 19 and figure 31).

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

**How to Deal with Unused I/O Pins:** I/O pins that are not needed by the user system (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 or pulled down to GND by their pull-down MOS transistors.

**Data control register** **(DCD0 to 2: $02C to $02E)**
**(DCR0 to 4: $030 to $034)**

**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**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | — | 0 |
| Read/Write | — | — | — | W |
| Bit name | Not used | Not used | Not used | DCR00 |

**DCR1 to DCR4**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | DCR13–DCR43 | DCR12–DCR42 | DCR11–DCR41 | DCR10–DCR40 |

| 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 | — | — | — | $R0_0$ |
| 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 | $R4_3$ | $R4_2$ | $R4_1$ | $R4_0$ |

**Figure 23 Data Control Registers (DCD, DCR)**

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

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | — |
| Read/Write | W | W | W | — |
| Bit name | PMRC3 | PMRC2* | PMRC1 | Not used |

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

| PMRC2 | $D_{12}$/$\overline{STOPC}$ mode selection |
|---|---|
| 0 | $D_{12}$ |
| 1 | $\overline{STOPC}$ |

| PMRC3 | $D_{13}$/$\overline{INT}_0$ mode selection |
|---|---|
| 0 | $D_{13}$ |
| 1 | $\overline{INT}_0$ |

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

**Figure 24 Port Mode Register C (PMRC )**

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

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | — | 0 |
| Read/Write | — | — | — | W |
| Bit name | Not used | Not used | Not used | PMRB0 |

| PMRB0 | $R0_0$/$\overline{INT}_1$ mode selection |
|---|---|
| 0 | $R0_0$ |
| 1 | $\overline{INT}_1$ |

**Figure 25 Port Mode Register B (PMRB)**

**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 | — | Inhibited |
| | | 1 | | |
| | 1 | 0 | | |
| | | 1 | TOC | PWM output |

**Figure 26 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 | — | Inhibited |
| | | | 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 27 Timer Mode Register D2 (TMD2)**

**Serial mode register 1A (SM1A: $005)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | SM1A3 | SM1A2 | SM1A1 | SM1A0 |

| SM1A3 | $R4_1/\overline{SCK}_1$ mode selection |
|---|---|
| 0 | $R4_1$ |
| 1 | $\overline{SCK}_1$ |

| SM1A2 | SM1A1 | SM1A0 | $\overline{SCK}_1$ | 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 28 Serial Mode Register 1A (SM1A)**

**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 | $R4_3/SO_1$ mode selection |
|---|---|
| 0 | $R4_3$ |
| 1 | $SO_1$ |

| PMRA1 | $R4_2/SI_1$ mode selection |
|---|---|
| 0 | $R4_2$ |
| 1 | $SI_1$ |

**Figure 29 Port Mode Register A (PMRA)**

**Compare enable register (CER: $018)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | — | 0 | 0 |
| Read/Write | W | — | W | W |
| Bit name | CER3 | Not used | CER1 | CER0 |

| CER3 | Digital/Analog selection |
|---|---|
| 0 | Digital input mode: $RD_0/COMP_0$ and $RD_1/COMP_1$ operate as an R port. |
| 1 | Analog input mode: $RD_0/COMP_0$ and $RD_1/COMP_1$ operate as analog input. |

| CER1 | CER0 | Analog input pin selection |
|---|---|---|
| 0 | 0 | $COMP_0$ |
| 0 | 1 | $COMP_1$ |
| 1 | 0 | Not used |
| 1 | 1 | Not used |

**Figure 30 Compare Enable Register**

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

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | — | — |
| Read/Write | W | W | — | — |
| Bit name | MIS3 | MIS2 | Not used | Not used |

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

| MIS2 | CMOS buffer on/off selection for pin $R4_3/SO_1$ |
|---|---|
| 0 | On |
| 1 | Off |

**Figure 31 Miscellaneous Register (MIS)**

## Prescalers

The MCU has the following prescaler S.

The prescaler operating conditions are listed in table 21, and the prescaler output supply is shown in figure 32. The timer 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.

### 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 at MCU reset.

**Table 21 Prescaler Operating Conditions**

| Prescaler | Input Clock | Reset Condition | Stop Conditions |
|---|---|---|---|
| Prescaler S | System clock | MCU reset | MCU reset, stop mode |

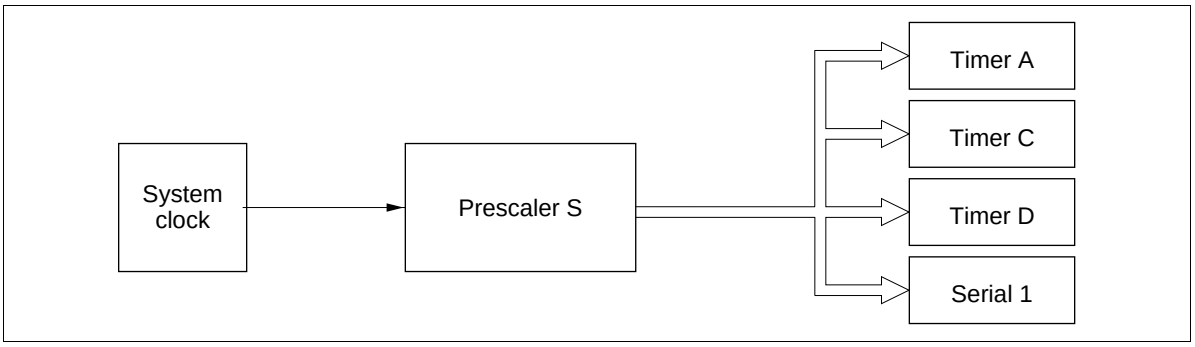



**Figure 32 Prescaler Output Supply**

## Timers

The MCU has three timer/counters (A, C, and D).

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

Timer A is an 8-bit free-running timer. Timers C and D are 8-bit multifunction timers, whose functions are listed in table 22. The operating modes are selected by software.

**Table 22 Timer Functions**

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

Note: — means not available.

**Timer A**

**Timer A Functions:** Timer A has the following functions.

- Free-running timer
  The block diagram of timer A is shown in figure 33.

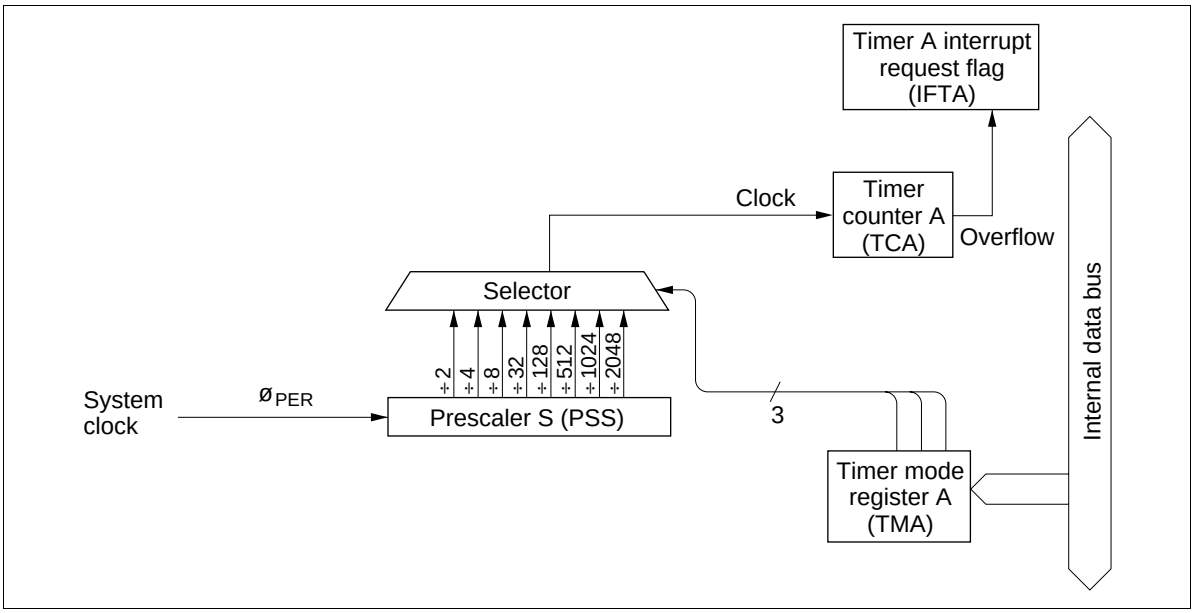



**Figure 33 Block Diagram of Timer A**

**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: $001, bit 2). Timer A continues to be incremented after reset to $00, and therefore it generates regular interrupts every 256 clocks.

**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 as shown in figure 34.

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

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | 0 | 0 | 0 |
| Read/Write | — | W | W | W |
| Bit name | Not used | TMA2 | TMA1 | TMA0 |

| TMA2 | TMA1 | TMA0 | Source prescaler | Input clock frequency | Operating mode |
|---|---|---|---|---|---|
| 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 | $2t_{cyc}$ | |

Note: Timer counter overflow output period (seconds) = input clock period (seconds) × 256.

**Figure 34 Timer Mode Register A (TMA)**

## Timer C

**Timer C Functions:** Timer C has the following functions.

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

The block diagram of timer C is shown in figure 35.

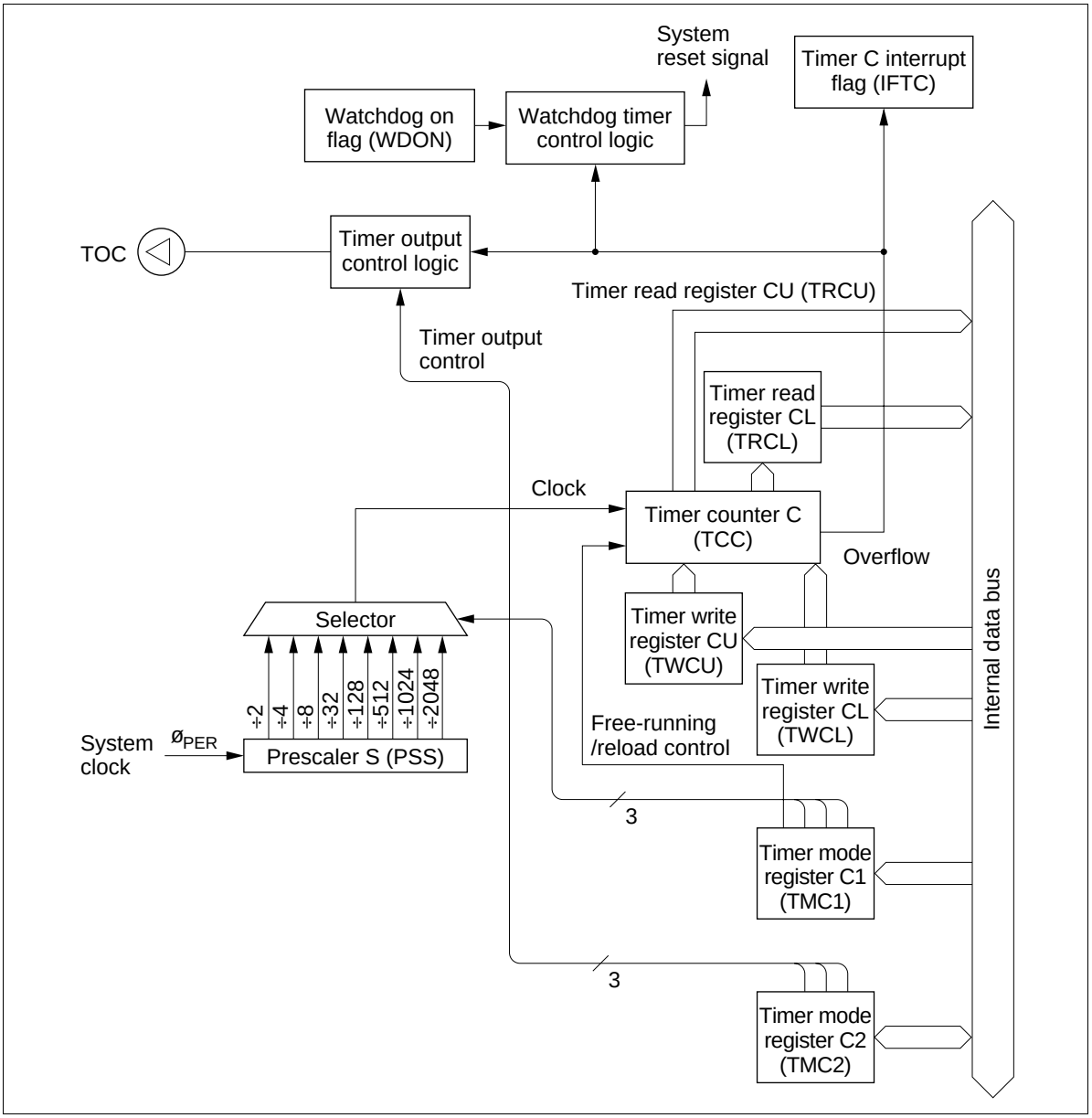



**Figure 35 Block Diagram of Timer C**

**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: $002, bit 2). IFTC is reset by software or MCU reset. Refer to figure 3 and table 1 for details.

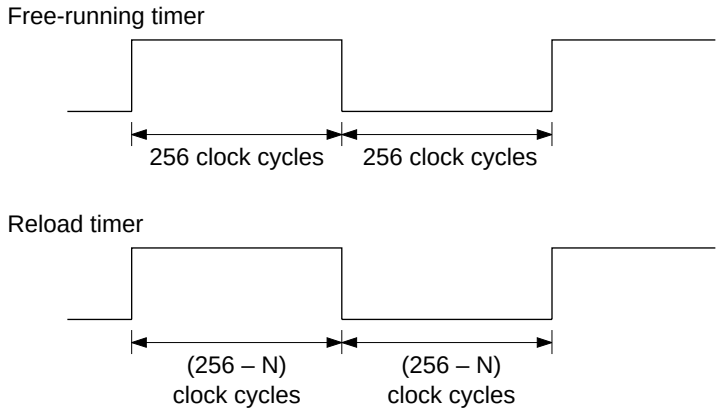
- Watchdog timer operation: Timer C is used as a watchdog timer for detecting out-of-control program routines by setting the watchdog on flag (WDON: $020, bit 1) to 1. If a program routine runs out of control and an overflow is generated, the MCU is reset. Program run can be controlled by initializing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

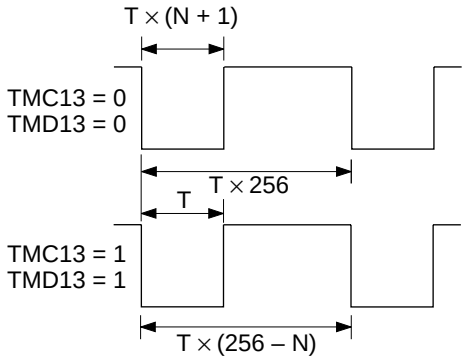
  By selecting the timer output mode, pin $R3_1$/TOC is set to TOC. The output from TOC is reset low by MCU reset.

  — Toggle output: When toggle output mode is selected, the output level is inverted if a clock is input after timer C has reached $FF. By using this function and the reload timer function, clock signals can be output at a required frequency for the buzzer. The output waveform is shown in figure 36.
  — PWM output: 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). The output waveform is shown in figure 36.
  — 0 output: When 0 output mode is selected, the output level is pulled low if a clock is input after timer C 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 C has reached $FF. Note that this function must be used only when the output level is low.

**Toggle output waveform (timers C, and D)**




**PWM output waveform (timers C and D)**




Notes: The waveform is always fixed low when N = $FF.
T: Input clock period to counter (figures 37 and 44)
N: The value of the timer write register

**Figure 36 Timer Output Waveform**

**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 the prescaler division ratio as shown in figure 37. It is reset to $0 by MCU reset.

Writing to 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 37 Timer Mode Register C1 (TMC1)**

- Timer mode register C2 (TMC2: $014): Three-bit read/write register that selects the timer C output mode as shown in figure 38. 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 | — | Inhibited |
| | | 1 | | |
| | 1 | 0 | | |
| | | 1 | TOC | PWM output |

**Figure 38 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) as shown in figures 39 and 40. The lower digit is reset to $0 by MCU reset, but the upper digit value is invalid.

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

**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 39 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 40 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 as shown in figures 41 and 42. The upper digit (TRCU) must be read first. At this time, the count of the timer C upper digit is obtained, and the count of the timer C lower digit is latched to the lower digit (TRCL). After this, by reading TRCL, the count of timer C when TRCU is read can be obtained.

**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 41 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 42 Timer Read Register C Upper Digit (TRCU)**

**Timer D**

**Timer D Functions:** Timer D has the following functions.

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

The block diagram for each operation mode of timer D is shown in figures 43 (A) and (B).

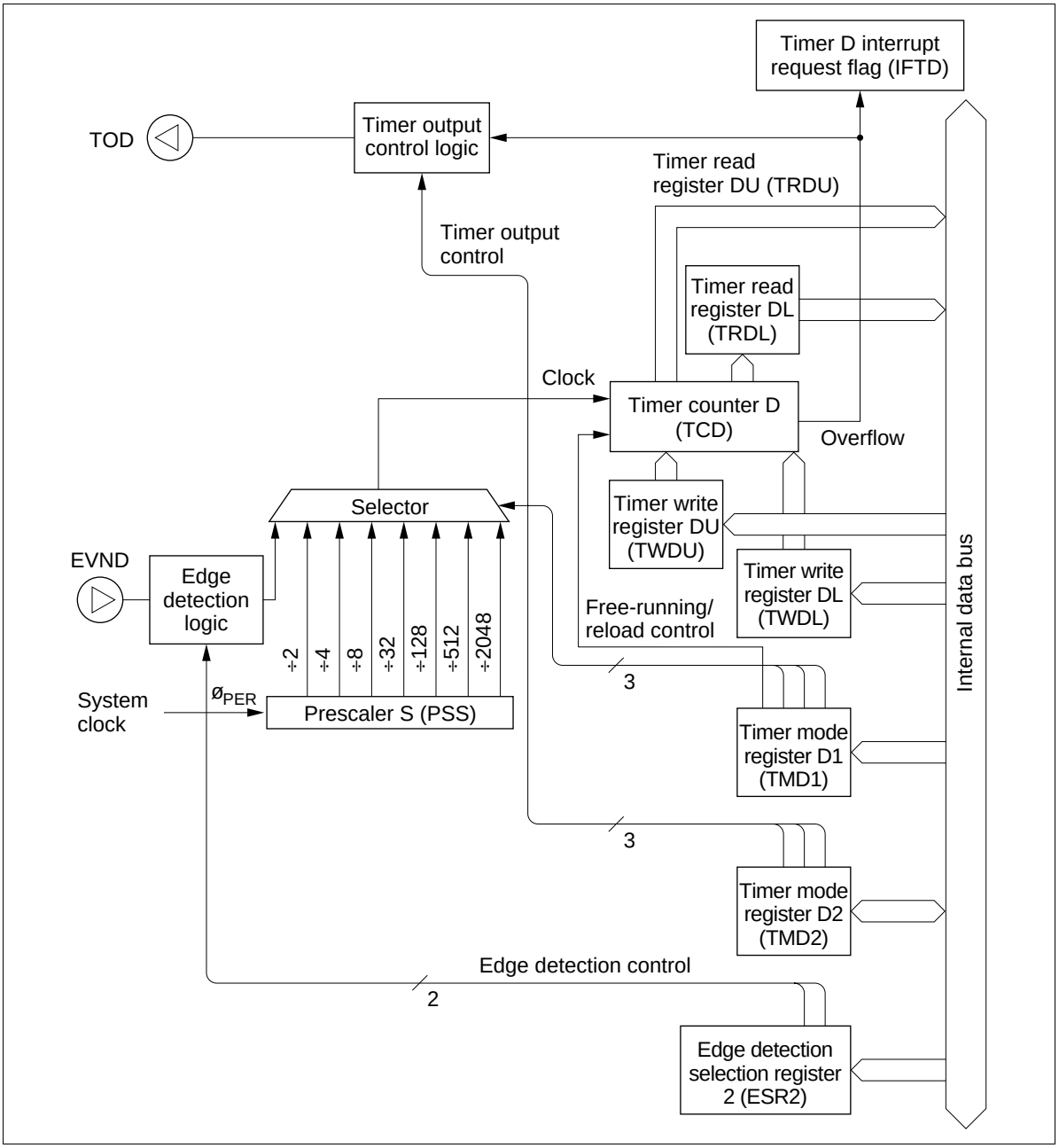



**Figure 43 (A) Block Diagram of Timer D (Free-Running/Reload Timer)**

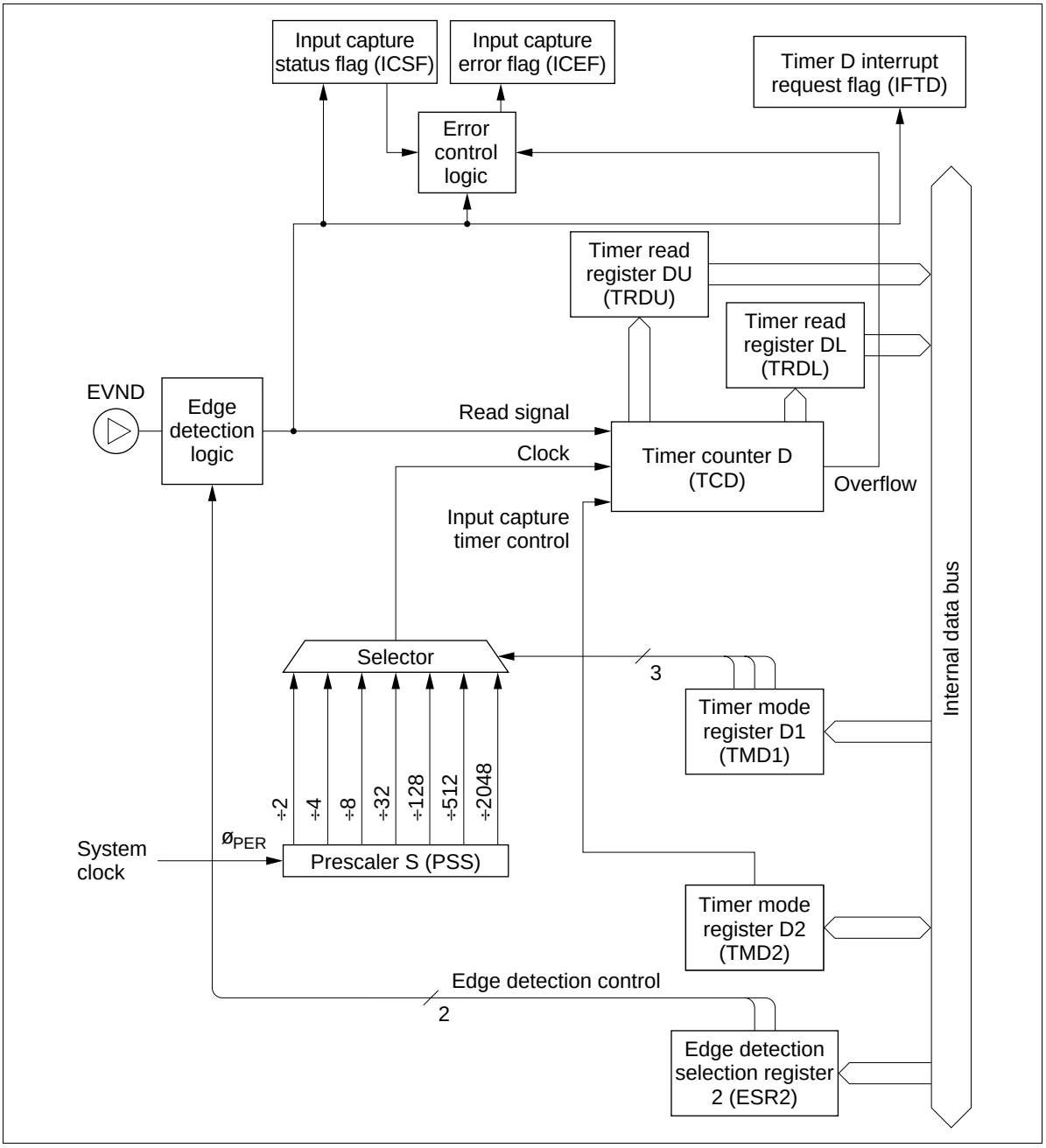



**Figure 43 (B) Block Diagram of Timer D (in Input Capture Timer Mode)**

**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: $003, bit 0). IFTD is 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-C's toggle output.
  — 0 output: The operation is basically the same as that of timer-C's 0 output.
  — 1 output: The operation is basically the same as that of timer-C'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: $003, bit 0) 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 are 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 as shown in figure 44. It is reset to $0 by MCU reset.

  Writing to 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 44 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 as shown in figure 45. 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 | — | Inhibited |
| | | | 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 45 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) as shown in figures 46 and 47. The operation of timer write register D is basically the same as that of timer write register C (TWCL: $00E, TWCU: $00F).

**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 46 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 47 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) as shown in figures 48 and 49. The operation of timer read register D is basically the same as that of timer read register C (TRCL: $00E, TRCU: $00F).

  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 48 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 49 Timer Read Register D Upper Digit (TRDU)**

- Port mode register C (PMRC: $025): Write-only register that selects $R4_0$/EVND pin function as shown in figure 50. 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 | PMRC3 | PMRC2 | PMRC1 | Not used |

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

| PMRC2 | $D_{12}/\overline{STOPC}$ mode selection |
|---|---|
| 0 | $D_{12}$ |
| 1 | $\overline{STOPC}$ |

| PMRC3 | $D_{13}/\overline{INT}_0$ mode selection |
|---|---|
| 0 | $D_{13}$ |
| 1 | $\overline{INT}_0$ |

**Figure 50 Port Mode Register C (PMRC)**

- Detection edge select register 2 (ESR2: $027): Write-only register that selects the detection edge of signals input to pin EVND as shown in figure 51. 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 * |

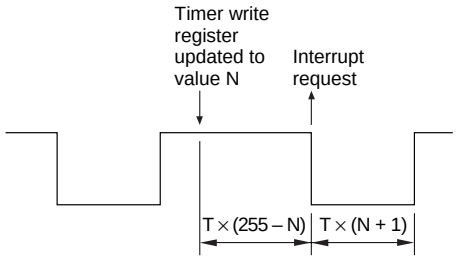
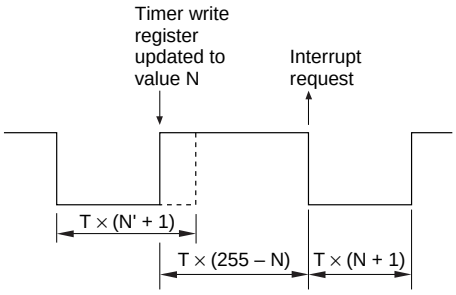
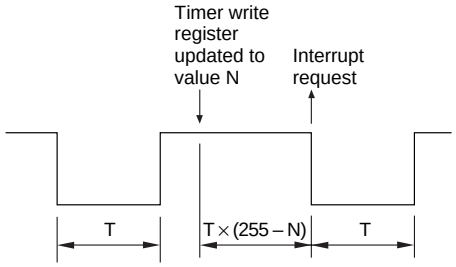
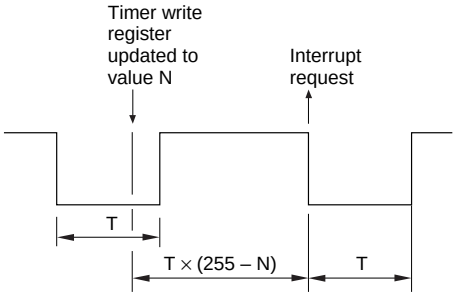
Note: * Both falling and rising edges are detected.

**Figure 51 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 23. 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 23 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 1

The MCU has one channel of serial interface. 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

**Serial interface 1**

- Serial data register 1 (SR1L: $006, SR1U: $007)
- Serial mode register 1A (SM1A: $005)
- Serial mode register 1B (SM1B: $028)
- Port mode register A (PMRA: $004)
- Miscellaneous register (MIS: $00C)
- Octal counter (OC)
- Selector

The block diagram of serial interface 1 is shown in figure 52.

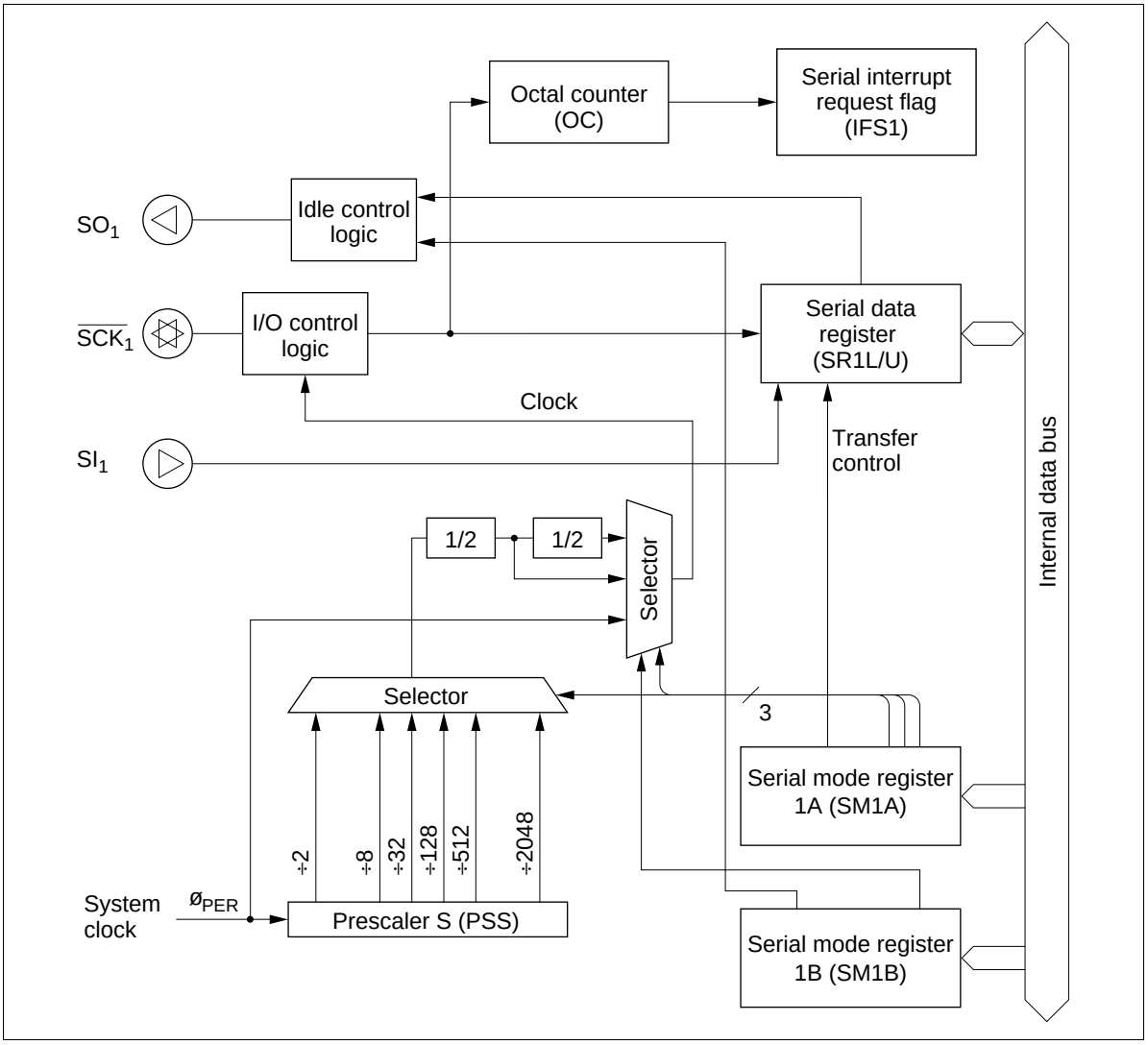



**Figure 52 Block Diagram of Serial Interface 1**

### Serial Interface Operation

**Selecting and Changing the Operating Mode:** Table 24 lists the serial interfaces' operating modes. To select an operating mode, use one of these combinations of port mode register A (PMRA: $004), and serial mode register 1A (SM1A: $005) settings; to change the operating mode of serial interface 1, always initialize the serial interface internally by writing data to serial mode register 1A. Note that serial interface 1 is initialized by writing data to serial mode register 1A. Refer to the following section Registers for Serial Interface for details.

**Table 24 Serial Interface 1 Operating Modes**

| SM1A | 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 |

**Pin Setting:** The $R4_1/\overline{SCK}_1$ pin is controlled by writing data to serial mode register 1A (SM1A: $005). Pins $R4_2/SI_1$ and $R4_3/SO_1$ 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 serial interface 1 is set by writing data to serial mode register 1A (SM1A: $005) and serial mode register 1B (SM1B: $028). Refer to the following section Registers for Serial Interface for details.

**Data Setting:** Transmit data of serial interface 1 is set by writing data to serial data register 1 (SR1L: $006, SR1U: $007). Receive data of serial interface 1 is obtained by reading the contents of serial data register 1. The serial data is shifted by the transmit clock and is input from or output to an external system.

The output level of the $SO_1$ pin is invalid until the first data is output after MCU reset, or until the output level control in idle states is performed.

**Transfer Control:** Serial interface 1 is activated by the STS instruction. The octal counter is reset to 000 by the STS instruction, and it increments at the rising edge of the transmit clock for 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 1 interrupt request flag (IFS1: $003, bit 2) is set, and the transfer stops.

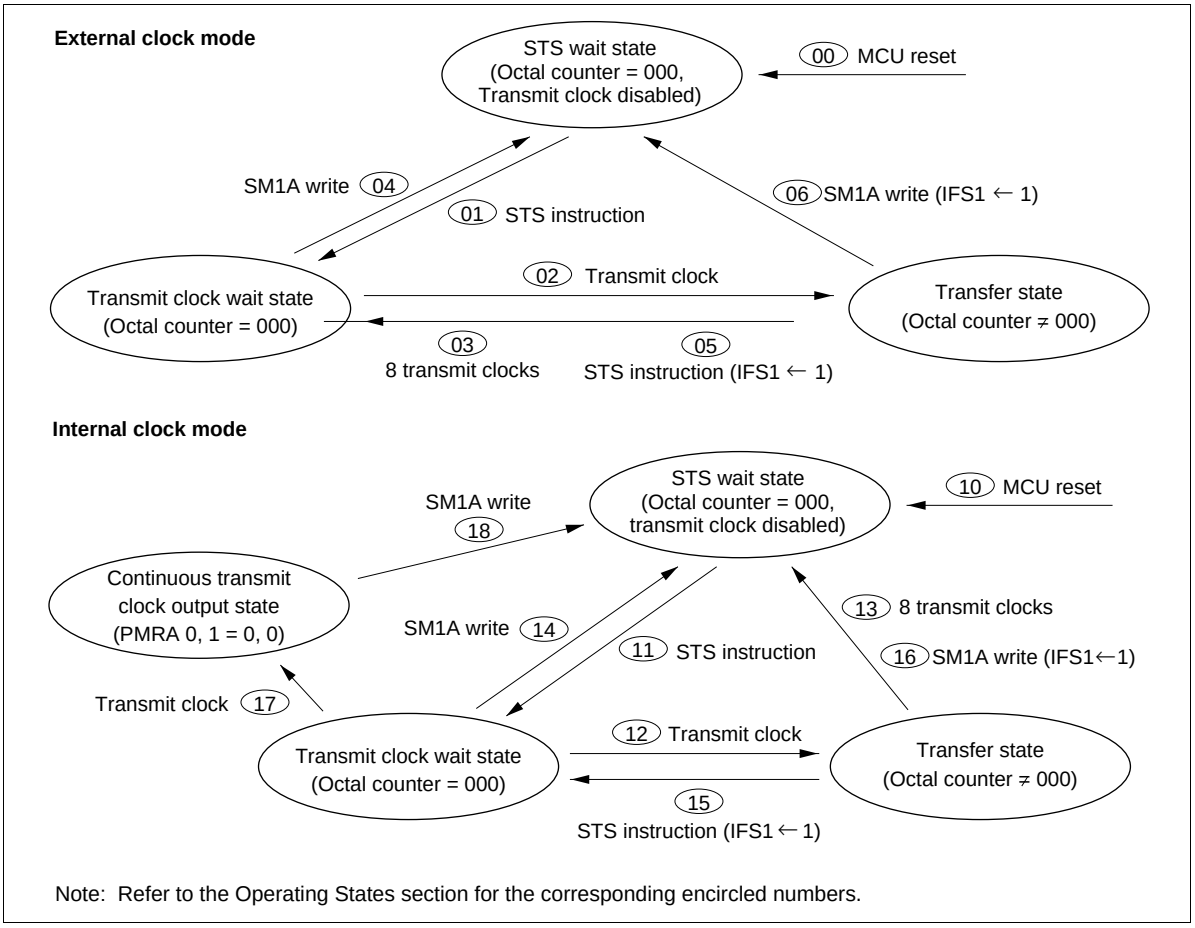
When the prescaler output is selected as the transmit clock, the transmit clock frequency is selected as $4t_{cyc}$ to $8192t_{cyc}$ by setting bits 0 to 2 (SM1A0–SM1A2) of serial mode register 1A (SM1A: $005) and bit 0 (SM1B0) of serial mode register 1B (SM1B: $028) as listed in table 25.

**Table 25 Serial Transmit Clock (prescaler output)**

| SM1B | SM1A | | | | |
|---|---|---|---|---|---|
| **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 | $8_{tcyc}$ |

**Operating States:** Serial interface 1 has the following operating states; transitions between them are shown in figure 53.

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




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

**Figure 53 Serial Interface State Transitions**

- STS wait state: Serial interface 1 enters STS wait state by MCU reset (00, 10 in figure 53). In STS wait state, serial interface 1 is initialized and the transmit clock is ignored. If the STS instruction is then executed (01, 11), serial interface 1 enters transmit clock wait state.

- Transmit clock wait state: Transmit clock wait state is the period between STS execution and the falling edge of the first transmit clock. In transmit clock wait state, input of the transmit clock (02, 12) increments the octal counter, shifts serial data register 1 (SR1L: $006, SR1U: $007), and enters the serial interface in transfer state. However, note that if continuous clock output mode 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 1A (SM1A: $005) (04, 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, 15), transmit clock wait state is entered. When eight clocks are input, transmit clock wait

state is entered (03) in external clock mode, and 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 1A (SM1A: $005) (06, 16) initializes serial interface 1, and STS wait state is entered.

If the state changes from transfer to another state, the serial 1 interrupt request flag (IFS1: $003, bit 2) 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 $\overline{SCK}_1$ 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 1A (SM1A: $005) is written to in continuous clock output mode (18), STS wait state is entered.

**Output Level Control in Idle States:** When serial interface 1 is in STS instruction wait state, the output of serial output pin $SO_1$ can be controlled by setting bit 1 (SM1B1) of serial mode register 1B (SM1B: $028) to 0 or 1. The output level control example of serial interface 1 is shown in figure 54. Note that the output level cannot be controlled in transfer state.

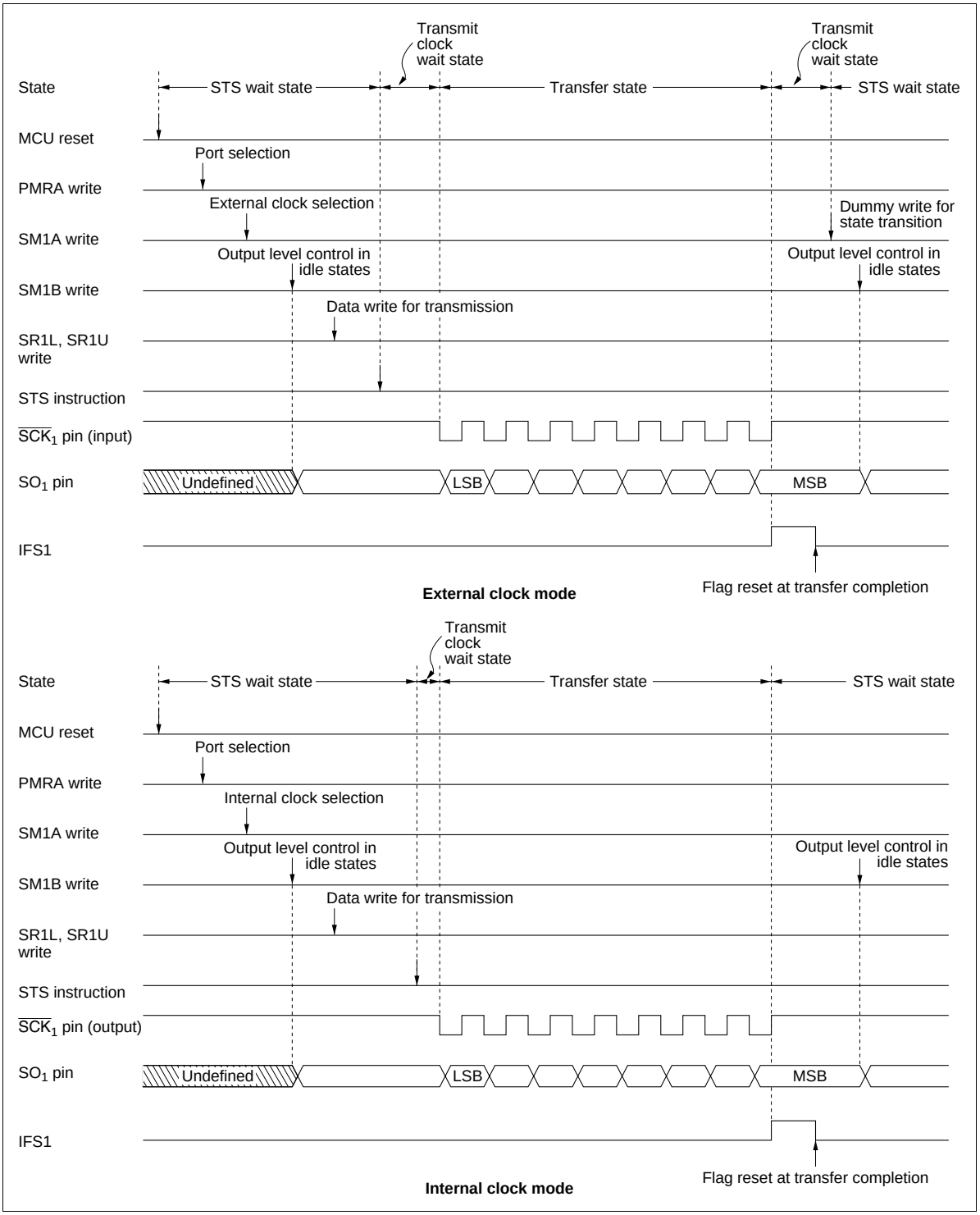



**Figure 54 Example of Serial Interface 1 Operation Sequence**

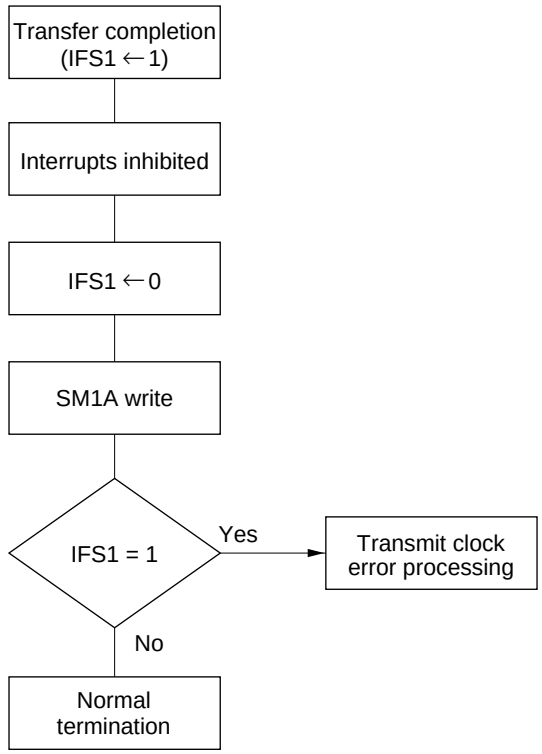
**Transmit Clock Error Detection (In External Clock Mode):** Serial interface 1 will malfunction if a spurious pulse caused by external noise conflicts with a normal transmit clock during transfer. A transmit clock error of this type can be detected as shown in figure 55.

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 1 interrupt request flag (IFS1: $003, bit 2) is set, and transmit clock wait state is entered. At the falling edge of the next normal clock signal, the transfer state is entered. After the transfer is completed and IFS is reset, writing to serial mode register 1A (SM1A: $005) changes the state from transfer to STS wait. At this time serial interface 1 is in the transfer state, and the serial 1 interrupt request flag (IFS1: $003, bit 2) is set again, and therefore the error can be detected.

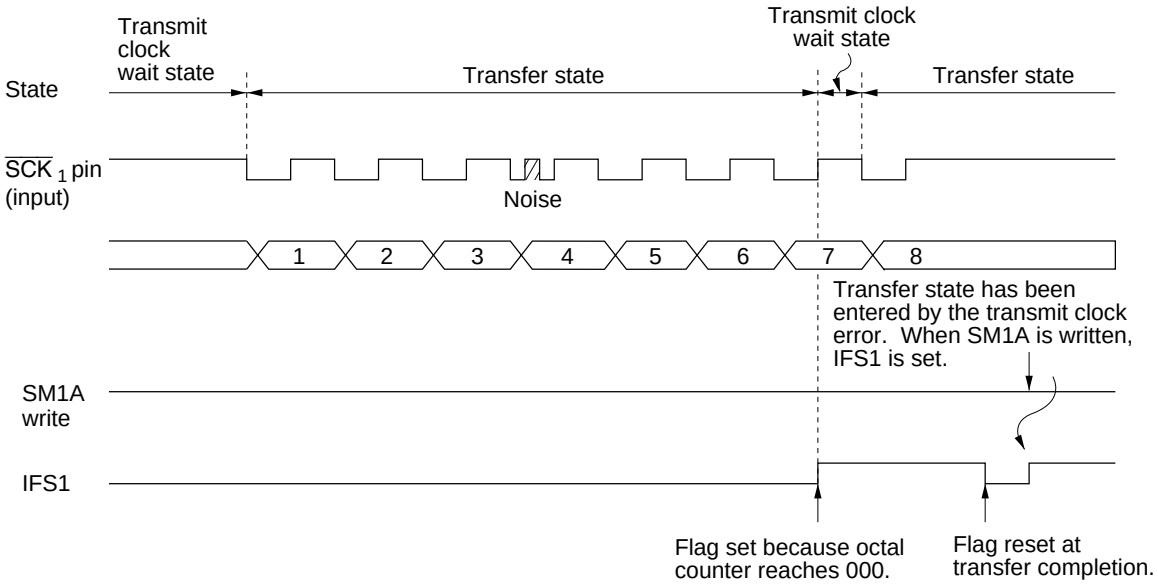
**Notes on Use:**

- Initialization after writing to registers: If port mode register A (PMRA: $004) is written to in transmit clock wait state or in transfer state, serial interface 1 must be initialized by writing to serial mode register 1A (SM1A: $005) again.

- Serial 1 interrupt request flag (IFS1: $003, bit 2) set: For serial interface 1, if the state is changed from transfer state to another by writing to serial mode register 1A (SM1A: $005) or executing the STS instruction during the first low pulse of the transmit clock, the serial 1 interrupt request flag (IFS1: $003, bit 2) is not set. To set the serial 1 interrupt request flag (IFS1: $003, bit 2), a serial mode register 1A (SM1A: $005) write or STS instruction execution must be programmed to be executed after confirming that the $\overline{SCK}_1$ 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 55 Transmit Clock Error Detection**

**Registers for Serial Interface 1**

When serial interface 1 operation is selected, serial data is read and written by the following registers.

- Serial mode register 1A (SM1A: $005)
- Serial mode register 1B (SM1B: $028)
- Serial data register 1 (SR1L: $006, SR1U: $007)
- Port mode register A (PMRA: $004)
- Miscellaneous register (MIS: $00C)

**Serial Mode Register 1A (SM1A: $005):** This register has the following functions (figure 56).

- $R4_1/\overline{SCK}_1$ pin function selection
- Serial interface 1 transmit clock selection
- Serial interface 1 prescaler division ratio selection
- Serial interface 1 initialization

Serial mode register 1A is a 4-bit write-only register. It is reset to $0 by MCU reset.

A write signal input to serial mode register 1A (SM1A: $005) discontinues the input of the transmit clock to serial data register 1 (SR1L: $006, SR1U: $007) and the octal counter, and the octal counter is reset to 000. Therefore, if a write is performed during data transfer, the serial 1 interrupt request flag (IFS1: $003, bit 2) 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 1A (SM1A: $005)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | SM1A3 | SM1A2 | SM1A1 | SM1A0 |

| SM1A3 | $R4_1/\overline{SCK}_1$ mode selection |
|---|---|
| 0 | $R4_1$ |
| 1 | $\overline{SCK}_1$ |

| SM1A2 | SM1A1 | SM1A0 | $\overline{SCK}_1$ | Clock source | Prescaler division ratio |
|---|---|---|---|---|---|
| 0 | 0 | 0 | Output | Prescaler | Refer to table 25 |
| | | 1 | | | |
| | 1 | 0 | | | |
| | | 1 | | | |
| 1 | 0 | 0 | | | |
| | | 1 | | | |
| | 1 | 0 | Output | System clock | — |
| | | 1 | Input | External clock | — |

**Figure 56 Serial Mode Register 1A (SM1A)**

**Serial Mode Register 1B (SM1B: $028):** This register has the following functions (figure 57).

- Serial interface 1 prescaler division ratio selection
- Serial interface 1 output level control in idle states

Serial mode register 1B is a 2-bit write-only register. It cannot be written during data transfer.

By setting bit 0 (SM1B0) of this register, the serial interface 1 prescaler division ratio is selected. Only bit 0 (SM1B0) can be reset to 0 by MCU reset. By setting bit 1 (SM1B1), the output level of the $SO_1$ pin is controlled in idle states of serial interface 1. The output level changes at the same time that SM1B1 is written to.

**Serial mode register 1B (SM1B: $028)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | Undefined | 0 |
| Read/Write | — | — | W | W |
| Bit name | Not used | Not used | SM1B1 | SM1B0 |

| SM1B1 | Output level control in idle states |
|---|---|
| 0 | Low level |
| 1 | High level |

| SM1B0 | Serial clock division ratio |
|---|---|
| 0 | Prescaler output divided by 2 |
| 1 | Prescaler output divided by 4 |

**Figure 57 Serial Mode Register 1B (SM1B)**

**Serial Data Register 1 (SR1L: $006, SR1U: $007):** This register has the following functions (figures 58 and 59)

- Serial interface 1 transmission data write and shift
- Serial interface 1 receive data shift and read

Writing data in this register is output from the $SO_1$ pin, LSB first, synchronously with the falling edge of the transmit clock; data is input, LSB first, through the $SI_1$ pin at the rising edge of the transmit clock. Input/output timing is shown in figure 60.

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 1 (lower digit) (SR1L: $006)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R/W | R/W | R/W | R/W |
| Bit name | SR13 | SR12 | SR11 | SR10 |

**Figure 58 Serial Data Register 1 (SR1L)**

**Serial data register 1 (upper digit) (SR1U: $007)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R/W | R/W | R/W | R/W |
| Bit name | SR17 | SR16 | SR15 | SR14 |

**Figure 59 Serial Data Register 1 (SR1U)**

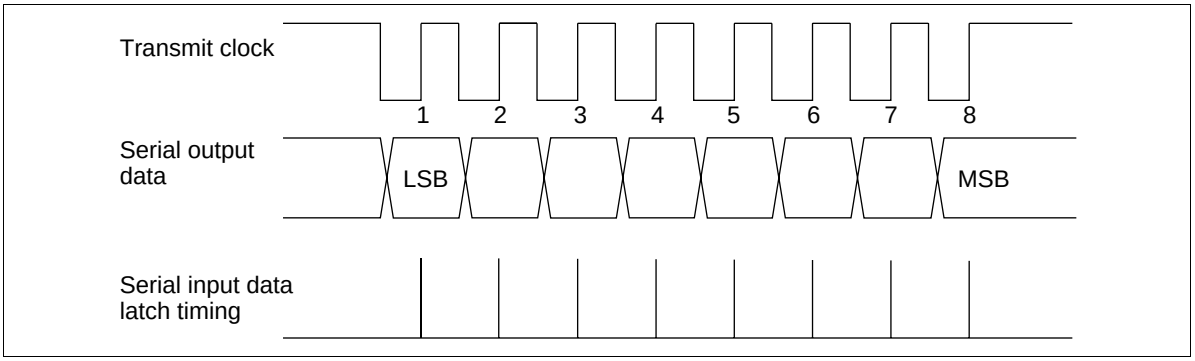



**Figure 60 Serial Interface Output Timing**

**Port Mode Register A (PMRA: $004):** This register has the following functions (figure 61).

- $R4_2/SI_1$ pin function selection
- $R4_3/SO_1$ pin function selection

Port mode register A is a 4-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 | $R4_3/SO_1$ mode selection |
|---|---|
| 0 | $R4_3$ |
| 1 | $SO_1$ |

| PMRA1 | $R4_2/SI_1$ mode selection |
|---|---|
| 0 | $R4_2$ |
| 1 | $SI_1$ |

**Figure 61 Port Mode Register A (PMRA)**

**Miscellaneous Register (MIS: $00C):** This register has the following functions (figure 62).

- $R4_3/SO_1$ pin PMOS control

Miscellaneous register 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 | — | — |
| Read/Write | W | W | — | — |
| Bit name | MIS3 | MIS2 | Not used | Not used |

| MIS2 | $R4_3/SO_1$ PMOS on/off selection |
|---|---|
| 0 | On |
| 1 | Off |

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

**Figure 62 Miscellaneous Register (MIS)**

## DTMF Generator Circuit

The MCU provides a dual-tone multifrequency (DTMF) generator circuit. The DTMF signal consists of two sine waves to access the switching system.

Figure 63 shows the DTMF keypad and frequencies. Each key enables tones to be generated corresponding to each frequency. Figure 64 shows a block diagram of the DTMF circuit.

The OSC clock (400 kHz, 800 kHz, 2 MHz, 3.58 MHz or 4 MHz) is changed into four clock signals through the division circuit (1/2, 1/5, 1/9 and 1/10). The DTMF circuit uses one of the four clock signals, which is selected by system clock select register 1 (SSR1: $029) and system clock select register 2 (SSR2: $02A) depending on the OSC clock frequency. The DTMF circuit has transformed programmable dividers, sine wave counters, and control registers.

The DTMF generation circuit is controlled by the following three registers.

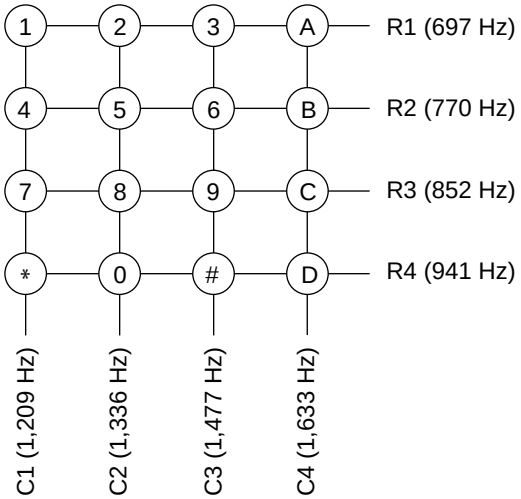



**Figure 63 DTMF Keypad and Frequencies**

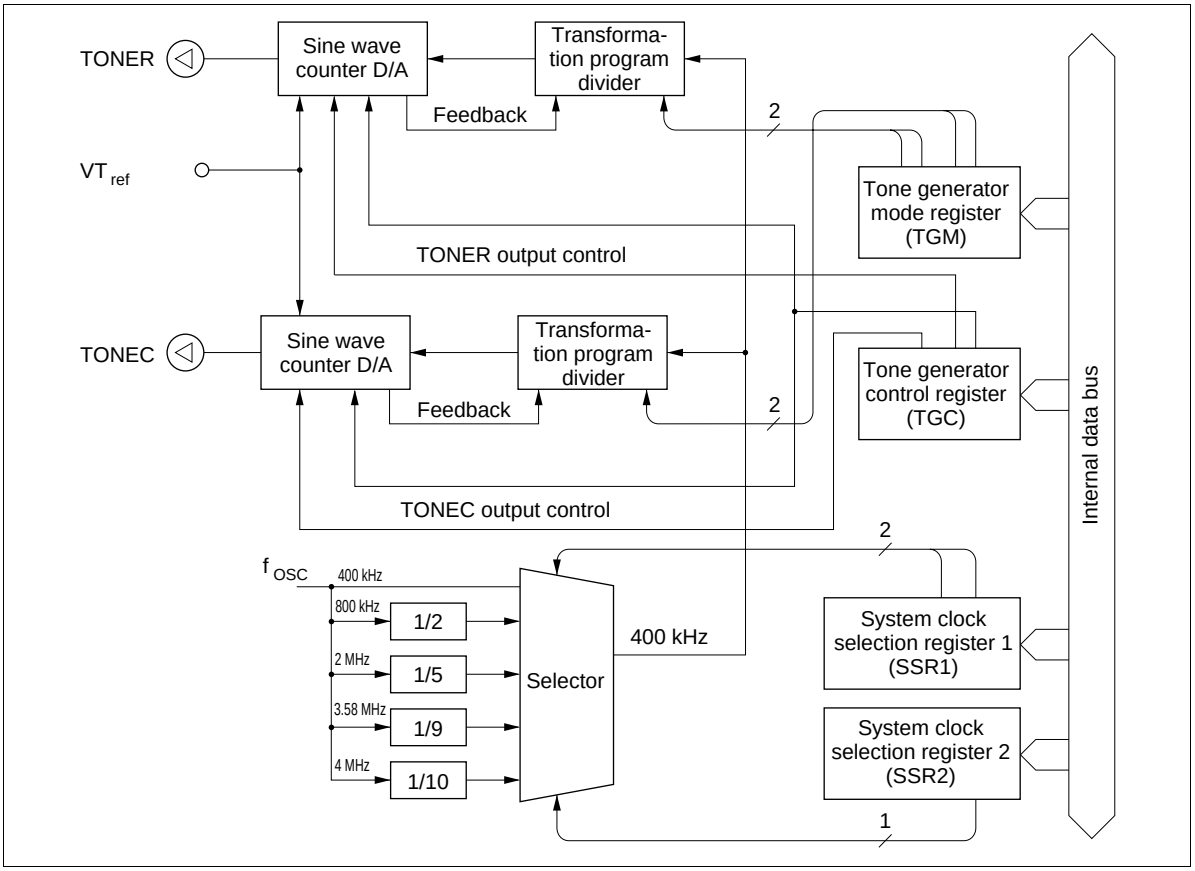



**Figure 64 Block Diagram of DTMF Generator Circuit**

**Tone Generator Mode Register (TGM: $019):** Four-bit write-only register, which controls output frequencies as shown in figure 65, and is reset to $0 by MCU reset.

**Tone generator mode register (TGM: $019)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | 0 |
| Read/Write | W | W | W | W |
| Bit name | TGM3 | TGM2 | TGM1 | TGM0 |

| TGM3 | TGM2 | TONEC output frequencies |
|---|---|---|
| 0 | 0 | $f_{C1}$ (1,209 Hz) |
| 0 | 1 | $f_{C2}$ (1,336 Hz) |
| 1 | 0 | $f_{C3}$ (1,477 Hz) |
| 1 | 1 | $f_{C4}$ (1,633 Hz) |

| TGM1 | TGM0 | TONER output frequencies |
|---|---|---|
| 0 | 0 | $f_{R1}$ (697 Hz) |
| 0 | 1 | $f_{R2}$ (770 Hz) |
| 1 | 0 | $f_{R3}$ (852 Hz) |
| 1 | 1 | $f_{R4}$ (941 Hz) |

**Figure 65 Tone Generator Mode Register (TGM)**

**Tone Generator Control Register (TGC: $01A):** Three-bit write-only register, which controls the start/stop of the DTMF signal output as shown in figure 66, and is reset to $0 by MCU reset. TONER and TONEC output can be independently controlled by bits 2 and 3 (TGC2, TGC3), and the DTMF circuit is controlled by bit 1 (TGC1) of this register.

**Tone generator control register (TGC: $01A)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | 0 | 0 | — |
| Read/Write | W | W | W | — |
| Bit name | TGC3 | TGC2 | TGC1 | Not used |

| TGC3 | TONEC output control (column) |
|---|---|
| 0 | No output |
| 1 | TONEC output (active) |

| TGC1 | DTMF enable bit |
|---|---|
| 0 | DTMF disable |
| 1 | DTMF enable |

| TGC2 | TONER output control (row) |
|---|---|
| 0 | No output |
| 1 | TONER output (active) |

**Figure 66 Tone Generator Control Register (TGC)**

**System Clock Select Registers 1 and 2 (SSR1: $029, SSR2: $02A):** Four-bit write-only registers. These registers must be set to the value specified in figures 67 and 68 depending on the frequency of the oscillator connected to the $OSC_1$ and $OSC_2$ pins. Note that if the combination of the oscillation frequency and the values in these registers is different from that specified in figures 67 and 68, the DTMF output frequencies will differ from the correct frequencies as listed in Table 26.

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

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | 0 | 0 |
| Read/Write | — | — | W | W |
| Bit name | Not used | Not used | SSR11 | SSR10 |

| SSR22 | SSR11 | SSR10 | System clock selection |
|---|---|---|---|
| 0 | 0 | 0 | 400 kHz |
| 0 | 0 | 1 | 800 kHz |
| 0 | 1 | 0 | 2 MHz |
| 0 | 1 | 1 | 4 MHz |
| 1 | × | × | 3.58 MHz |

×: Don't care

**Figure 67 System Clock Select Register 1 (SSR1)**

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

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | 0 | — | — |
| Read/Write | — | W | — | — |
| Bit name | Not used | SSR22 | Not used | Not used |

| SSR22 | System clock selection |
|---|---|
| 0 | Selected from 400 kHz, 800 kHz, 2 MHz, 4 MHz * |
| 1 | 3.58 MHz |

Note: * Refer to system clock select register 1 (SSR1).

**Figure 68 System Clock Select Register 2 (SSR2)**

**DTMF Output:** The sine waves of the row-group and column-group are individually converted in the D/A conversion circuit which provides a high-precision ladder resistance. The DTMF output pins (TONER, TONEC) transmit the sine waves of the row-group and column-group, respectively. Figure 69 shows the tone output equivalent circuit. Figure 70 shows the output waveform. One cycle of this wave consists of 32 slots. Therefore, the output waveform is stable with little distortion. Table 26 lists the frequency deviation of the MCU from standard DTMF signals.

**Table 26 Frequency Deviation of the MCU from Standard DTMF**

| | | fosc = 400 kHz, 800 kHz, 2 MHz, 4 MHz | | fosc = 3.58 MHz | |
|---|---|---|---|---|---|
| | Standard DTMF (Hz) | MCU (Hz) | Deviation from Standard (%) | MCU (Hz) | Deviation from Standard (%) |
| R1 | 697 | 694.44 | –0.37 | 690.58 | –0.92 |
| R2 | 770 | 769.23 | –0.10 | 764.96 | –0.65 |
| R3 | 852 | 851.06 | –0.11 | 846.33 | –0.67 |
| R4 | 941 | 938.97 | –0.22 | 933.75 | –0.77 |
| C1 | 1,209 | 1,212.12 | 0.26 | 1,205.39 | –0.30 |
| C2 | 1,336 | 1,333.33 | –0.20 | 1,325.92 | –0.75 |
| C3 | 1,477 | 1,481.48 | 0.30 | 1,473.25 | –0.25 |
| C4 | 1,633 | 1,639.34 | 0.39 | 1,630.23 | –0.17 |

Notes: This frequency deviation value does not include the frequency deviation due to the oscillator element. Also note that in this case the ratio of the high level and low level widths in the oscillator waveform due to the oscillator element will be 50%:50%.

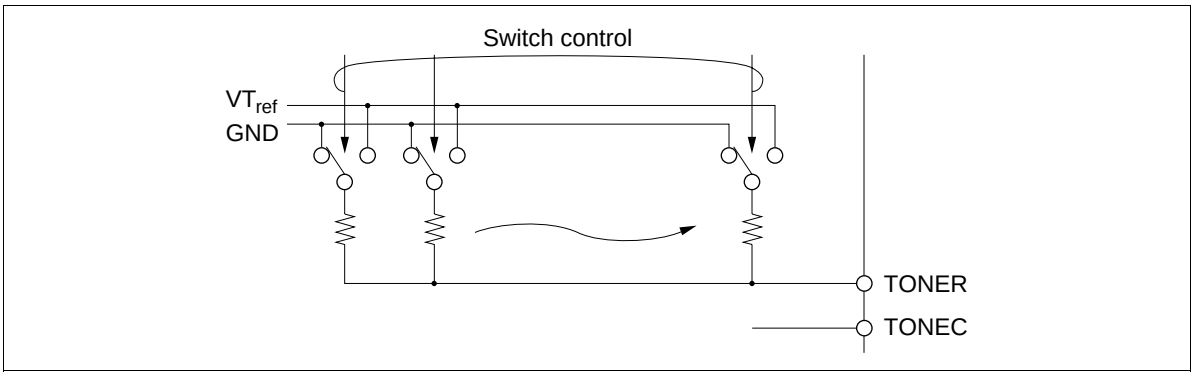



**Figure 69 Tone Output Equivalent Circuit**

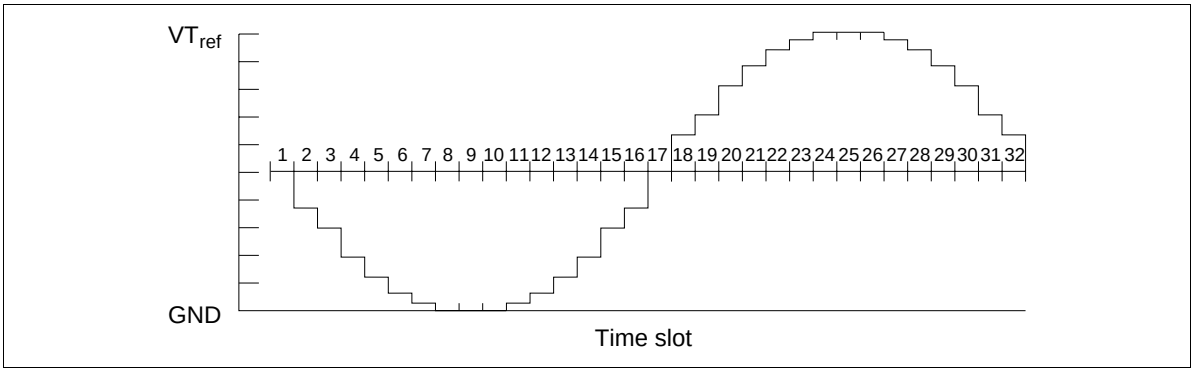



**Figure 70 Waveform of Tone Output**

## Comparator

The block diagram of the comparator is shown in figure 71. The comparator compares input voltage with the reference voltage.

Setting 1 to bit 3 (CER3) of the compare enable register (CER: $018) executes a voltage comparison. If an input voltage at $COMP_0$ or $COMP_1$ is higher than the reference voltage, the TM or TMD command sets the status flag (ST) high for the corresponding bits of the compare data register (CDR: $017) to $COMP_0$ or $COMP_1$. On the other hand, if an input voltage at $COMP_0$ or $COMP_1$ is lower than the reference voltage, the TM or TMD command clears the ST to 0.

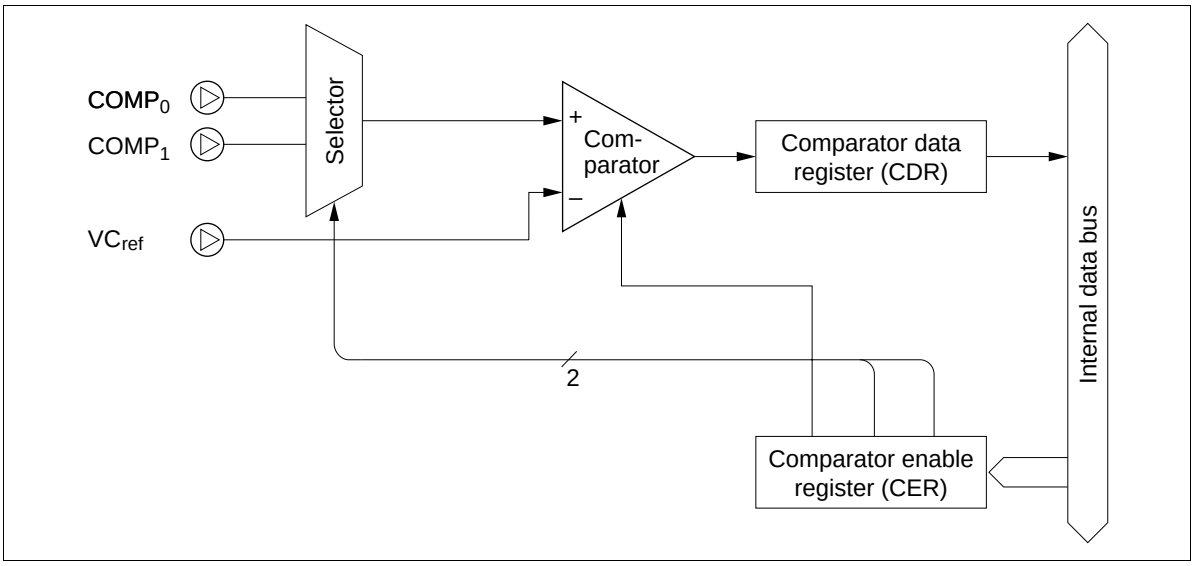



**Figure 71 Block Diagram of Comparator**

**Compare Enable Register (CER: $018):** Three-bit write-only register which enables comparator operation, and selects the reference voltage and the analog input pin.

**Compare Data Register (CDR: $017):** Two-bit read-only register which latches the result of the comparison between the analog input pins and the reference voltage. Bits 0 and 1 reflect the results of comparison with $COMP_0$ and $COMP_1$, respectively. This register can be read only by the TM or TMD command. Only bit CER3 corresponds to the analog input pin, which is selected by bits CER0 and CER1. After a compare operation, the data in this register is not retained.

**Note on Use:** During compare operation, pins $RD_0/COMP_0$ and $RD_1/COMP_1$ operate as analog inputs and cannot operate as R ports.

The comparator can operate in active mode but is disabled in other modes.

$RE_0/VC_{ref}$ cannot operate as an R port when the external input voltage is selected as the reference.

**Compare enable register (CER: $018)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | 0 | — | 0 | 0 |
| Read/Write | W | — | W | W |
| Bit name | CER3 | Not used | CER1 | CER0 |

| CER3 | Digital/Analog selection |
|---|---|
| 0 | Digital input mode: $RD_0$ /$COMP_0$, $RD_1$/$COMP_1$ operate as R port |
| 1 | Analog input mode: $RD_0$ /$COMP_0$, $RD_1$/$COMP_1$ operate as analog input |

| CER1 | CER0 | Analog input pin selection |
|---|---|---|
| 0 | 0 | $COMP_0$ |
| 0 | 1 | $COMP_1$ |
| 1 | 0 | Not used |
| 1 | 1 | Not used |

**Figure 72 Compare Enable Register**

**Compare data register (CDR: $017)**

| Bit | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Initial value | — | — | Undefined | Undefined |
| Read/Write | — | — | R | R |
| Bit name | Not used | Not used | CDR1 | CDR0 |

CDR0 — Result of $COMP_0$ comparison

CDR1 — Result of $COMP_1$ comparison

**Figure 73 Compare Data Register**

## Programmable ROM (HD4074654)

The HD4074654 is a ZTAT™ microcomputer with built-in PROM that can be programmed in PROM mode.

### PROM Mode Pin Description

| Pin No. | | MCU Mode | | PROM Mode | |
|---|---|---|---|---|---|
| **DP-42S** | **FP-44A** | **Pin Name** | **I/O** | **Pin Name** | **I/O** |
| 1 | 39 | $RD_0/COMP_0$ | I | $\overline{CE}$ | I |
| 2 | 40 | $RD_1/COMP_1$ | I | $\overline{OE}$ | I |
| 3 | 41 | TONEC | O | | |
| 4 | 42 | TONER | O | | |
| 5 | 43 | $VT_{ref}$ | I | $V_{CC}$ | |
| 6 | 1 | $RE_0/VC_{ref}$ | I | $M_1$ | I |
| 7 | 2 | $\overline{TEST}$ | I | $\overline{TEST}$ | I |
| 8 | 3 | $OSC_1$ | I | $V_{CC}$ | |
| 9 | 4 | $OSC_2$ | O | | |
| 10 | 5 | $\overline{RESET}$ | I | $\overline{RESET}$ | I |
| 11 | 6 | GND | I | GND | |
| 12 | 7 | $D_0$ | I/O | | O |
| 13 | 8 | $D_1$ | I/O | | O |
| 14 | 9 | $D_2$ | I/O | $V_{CC}$ | |
| 15 | 10 | $D_3$ | I/O | $V_{CC}$ | |
| 16 | 11 | $D_4$ | I/O | $O_4$ | I/O |
| 17 | 12 | $D_5$ | I/O | $O_5$ | I/O |
| 18 | 13 | $D_6$ | I/O | $O_6$ | I/O |
| 19 | 14 | $D_7$ | I/O | $O_7$ | I/O |
| 20 | 15 | $D_8$ | I/O | $A_{13}$ | I |
| 21 | 16 | $D_9$ | I/O | $A_{14}$ | I |
| 22 | 17 | $D_{12}/\overline{STOPC}$ | I | $A_9$ | I |
| 23 | 18 | $D_{13}/\overline{INT}_0$ | I | $V_{PP}$ | |
| 24 | 19 | $R0_0/\overline{INT}_1$ | I/O | $\overline{M}_0$ | I |
| 25 | 20 | $R1_0$ | I/O | $A_5$ | I |
| 26 | 21 | $R1_1$ | I/O | $A_6$ | I |
| 27 | 23 | $R1_2$ | I/O | $A_7$ | I |
| 28 | 24 | $R1_3$ | I/O | $A_8$ | I |

| Pin No. | | MCU Mode | | PROM Mode | |
|---|---|---|---|---|---|
| **DP-42S** | **FP-44A** | **Pin Name** | **I/O** | **Pin Name** | **I/O** |
| 29 | 25 | $R2_0$ | I/O | $A_0$ | I |
| 30 | 26 | $R2_1$ | I/O | $A_{10}$ | I |
| 31 | 27 | $R2_2$ | I/O | $A_{11}$ | I |
| 32 | 28 | $R2_3$ | I/O | $A_{12}$ | I |
| 33 | 29 | $R3_0$ | I/O | $A_1$ | I |
| 34 | 30 | $R3_1$/TOC | I/O | $A_2$ | I |
| 35 | 31 | $R3_2$/TOD | I/O | $A_3$ | I |
| 36 | 32 | $R3_3$ | I/O | $A_4$ | I |
| 37 | 33 | $R4_0$/EVND | I/O | $O_0$ | I/O |
| 38 | 34 | $R4_1/\overline{SCK}_1$ | I/O | $O_1$ | I/O |
| 39 | 35 | $R4_2/SI_1$ | I/O | $O_2$ | I/O |
| 40 | 36 | $R4_3/SO_1$ | I/O | $O_3$ | I/O |
| 41 | 37 | SEL | I | | |
| 42 | 38 | $V_{CC}$ | I | $V_{CC}$ | |
| – | 22 | NC | – | | |
| – | 44 | NC | – | | |

Note: I/O: Input/output pin, I: Input pin, O: Output pin

### Programming the Built-In PROM

The MCU's built-in PROM is programmed in PROM mode. PROM mode is set by pulling $\overline{TEST}$, $\overline{M}_0$, and $\overline{M}_1$ low, and $\overline{RESET}$ low as shown in figure 74. 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 42-to-28-pin socket adapter. Recommended PROM programmers and socket adapters of the HD4074654 are listed in table 28.

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, 4 kwords of built-in PROM are to be programmed by a general-purpose PROM programmer, an 8 kbyte address space ($0000–$7FFF) must be specified.

### Warnings

1. Always specify addresses $0000 to $1FFF when programming with a PROM programmer. If address $2000 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 version 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 programmed at high speed without risk of voltage stress or damage to data reliability.

Programming and verification modes are selected as listed in table 27.

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

**Table 27 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 28 Recommended PROM Programmers and Socket Adapters**

| PROM Programmer | | Socket Adapter | | |
|---|---|---|---|---|
| **Manufacturer** | **Model Name** | **Package** | **Manufacturer** | **Model Name** |
| DATA I/O Corp. | 121B | DP-42S | Hitachi | HS4654ESS01H |
| AVAL Corp. | PKW-1000 | FP-44A | Hitachi | HS4654ESH01H |

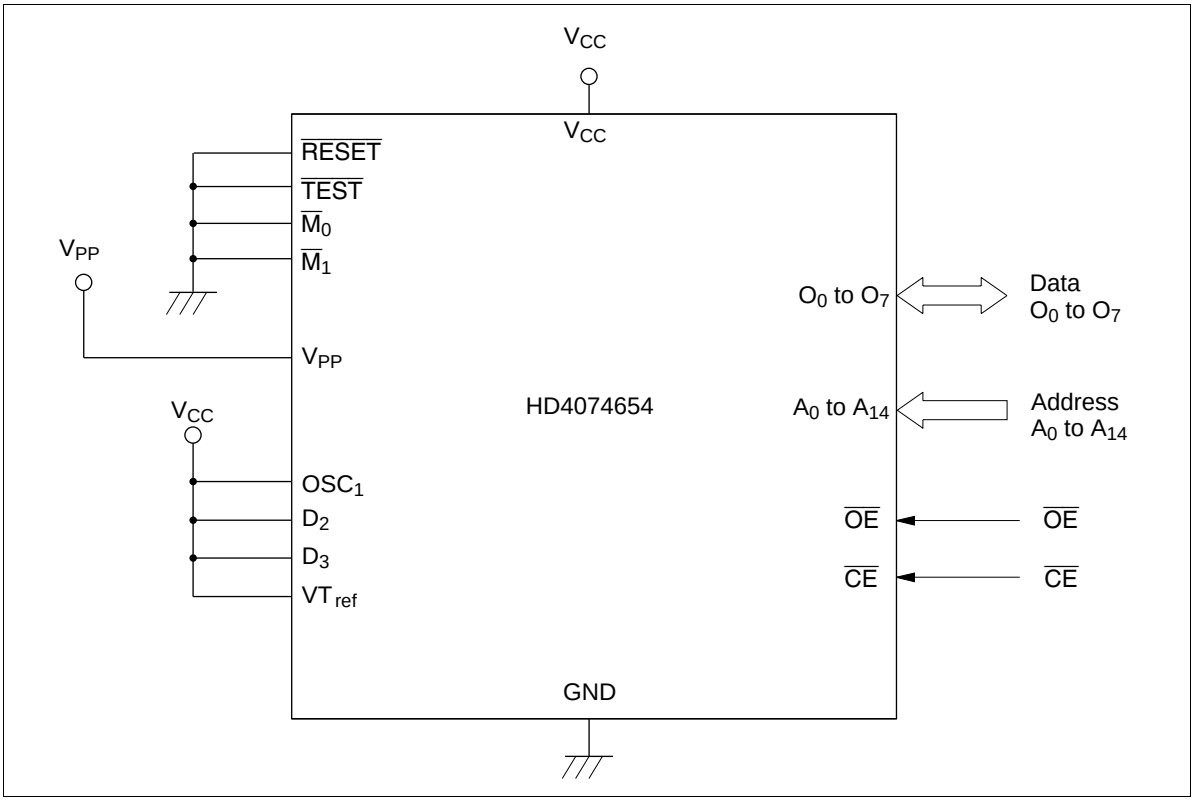



**Figure 74 PROM Mode Connections**

## Addressing Modes

### RAM Addressing Modes

The MCU has three RAM addressing modes, as shown in figure 75 and described below.

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

**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 as a RAM address.

**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.

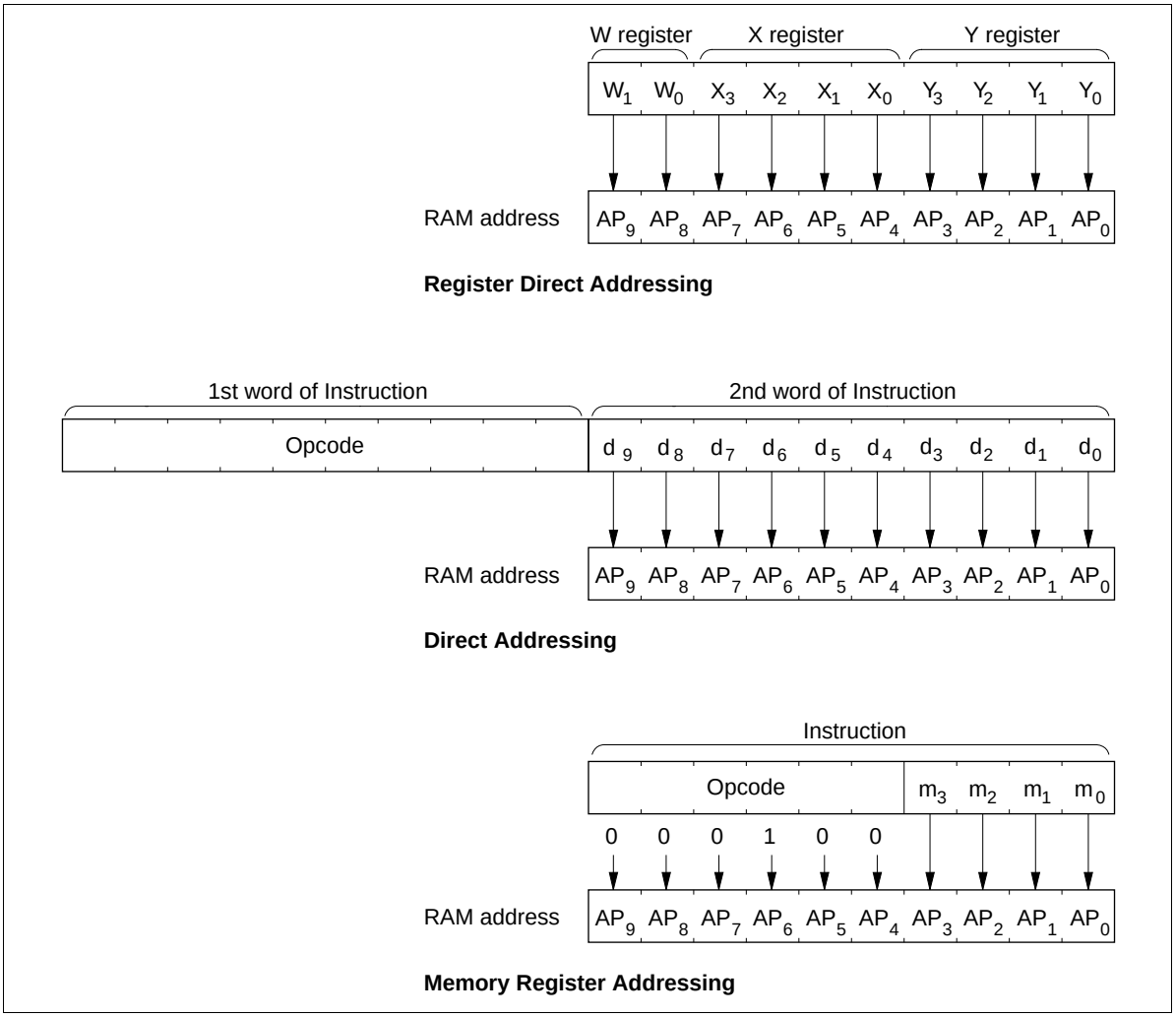



**Figure 75 RAM Addressing Modes**

**ROM Addressing Modes and the P Instruction**

The MCU has four ROM addressing modes, as shown in figure 76 and described below.

**Direct Addressing Mode:** A program can branch to any address in the 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, as shown in figure 78. 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 as shown in figure 77. 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.

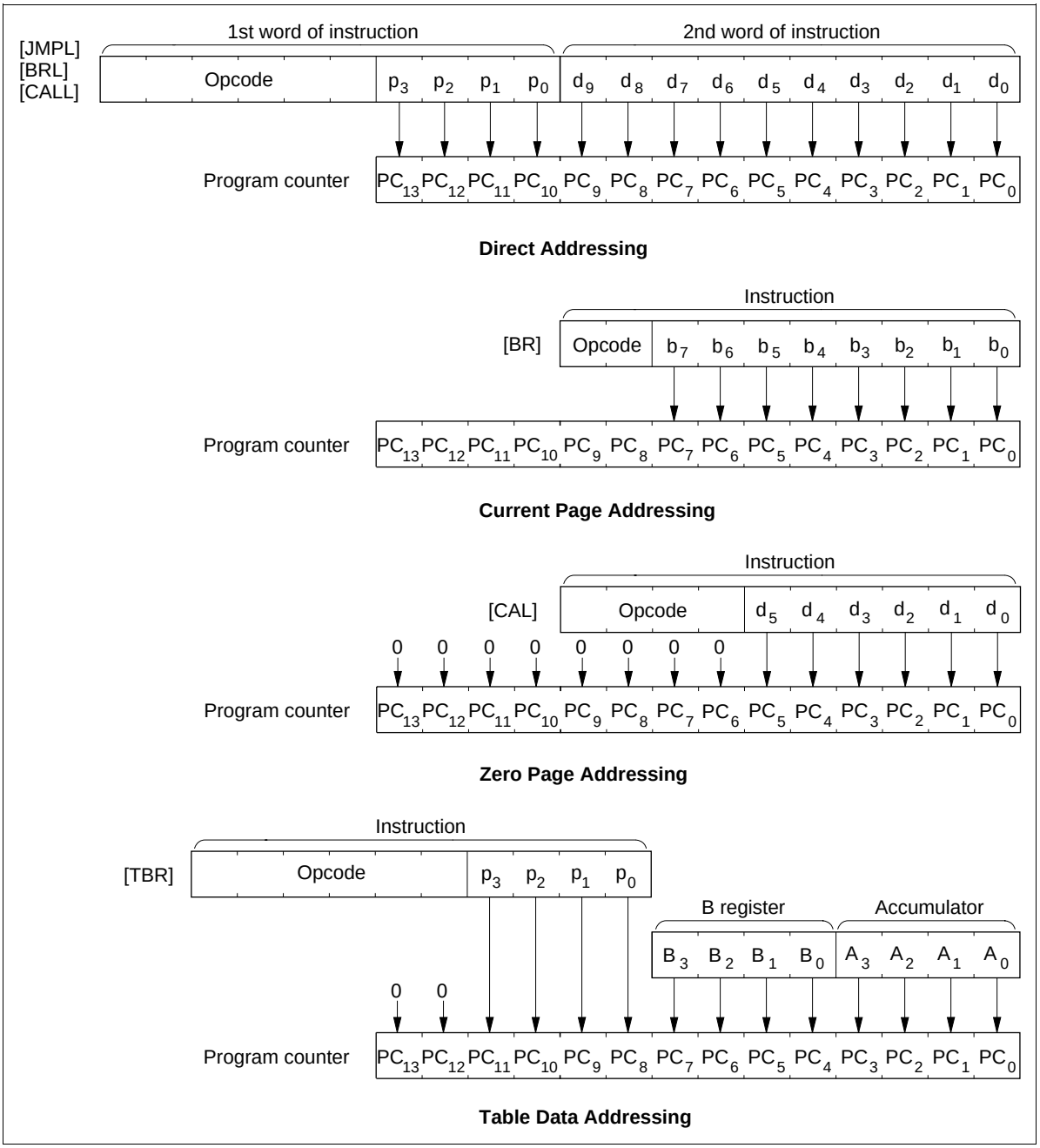



**Figure 76 ROM Addressing Modes**

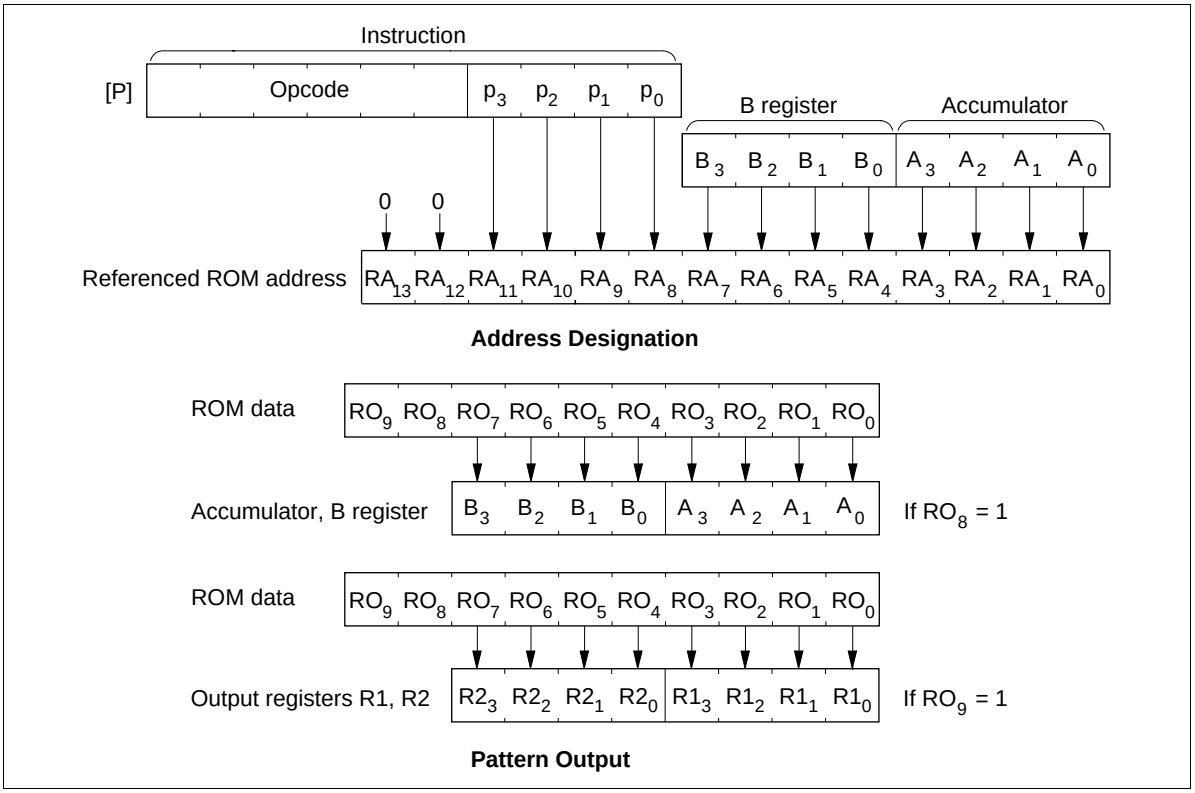



**Figure 77 P Instruction**

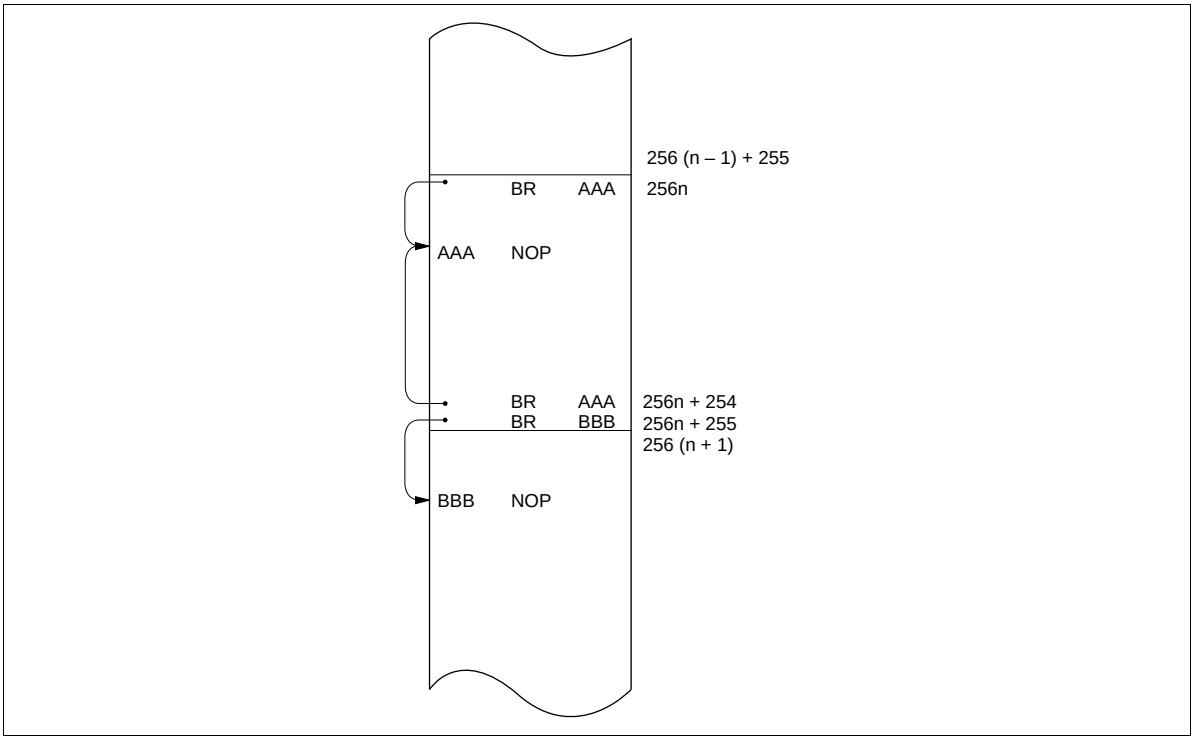



**Figure 78 Branching when the Branch Destination is on a Page Boundary**

## Absolute Maximum Ratings

| Item | Symbol | Value | Unit | Notes |
|---|---|---|---|---|
| Supply voltage | $V_{CC}$ | –0.3 to +7.0 | 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 | $\sum I_o$ | 80 | mA | 2 |
| Total permissible output current | $-\sum I_o$ | 50 | mA | 3 |
| Maximum input current | $I_o$ | 4 | mA | 4, 5 |
| | | 30 | mA | 4, 6 |
| Maximum output current | $-I_o$ | 4 | mA | 7, 8 |
| | | 20 | mA | 7, 9 |
| Operating temperature | $T_{opr}$ | –20 to +75 | °C | |
| 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 $D_{13}$ ($V_{PP}$) of the HD4074654.
2. The total permissible input current is the total of input currents simultaneously flowing in from all the I/O pins to GND.
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 GND.
5. Applies to $D_0$–$D_3$, and R0–R4.
6. Applies to $D_4$–$D_9$.
7. The maximum output current is the maximum current flowing out from $V_{CC}$ to each I/O pin.
8. Applies to $D_4$–$D_9$ and R0–R4.
9. Applies to $D_0$–$D_3$.

## Electrical Characteristics

**DC Characteristics (HD404652, HD404654: $V_{CC}$ = 1.8 to 6.0 V, GND = 0 V, $T_a$ = –20 to +75°C; HD4074654: $V_{CC}$ = 2.7 to 5.5 V, GND = 0 V, $T_a$ = –20 to +75°C, unless otherwise specified)**

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Notes |
|---|---|---|---|---|---|---|---|---|
| Input high voltage | $V_{IH}$ | $\overline{RESET}$, $\overline{STOPC}$, $\overline{INT_0}$, $\overline{INT_1}$, $\overline{SCK_1}$, $SI_1$, EVND | $0.9V_{CC}$ | — | $V_{CC}$ + 0.3 | V | | |
| | | $OSC_1$ | $V_{CC}$ – 0.3 | — | $V_{CC}$ + 0.3 | V | External clock | |
| Input low voltage | $V_{IL}$ | $\overline{RESET}$, $\overline{STOPC}$, $\overline{INT_0}$ $\overline{INT_1}$, $\overline{SCK_1}$ $SI_1$, EVND | –0.3 | — | 0.10 $V_{CC}$ | V | | |
| | | $OSC_1$ | –0.3 | — | 0.3 | V | External clock | |
| Output high voltage | $V_{OH}$ | $\overline{SCK_1}$, $SO_1$, TOC,TOD | $V_{CC}$ – 1.0 | — | — | V | $–I_{OH}$ = 0.5 mA | |
| Output low voltage | $V_{OL}$ | $\overline{SCK_1}$, $SO_1$, TOC, TOD | — | — | 0.4 | V | $I_{OL}$ = 0.4 mA | |
| I/O leakage current | $\lvert I_{IL} \rvert$ | $\overline{RESET}$, $\overline{STOPC}$, $\overline{INT_0}$, $\overline{INT_1}$, $\overline{SCK_1}$, $SI_1$, $SO_1$, EVND, $OSC_1$, TOC, TOD | — | — | 1 | µA | $V_{in}$ = 0 V to $V_{CC}$ | 1 |
| Current dissipation in active mode | $I_{CC1}$ | $V_{CC}$ | — | 5 | — | mA | $V_{CC}$ = 5 V, $f_{OSC}$ = 4 MHz Digital input mode | 2, 5 |
| | $I_{CC2}$ | | — | 0.6 | 1.8 | mA | $V_{CC}$ = 3 V, $f_{OSC}$ = 800 kHz Digital input mode | 2, 5 |
| | $I_{CMP1}$ | | — | 9 | — | mA | $V_{CC}$ = 5 V, $f_{OSC}$ = 4 MHz Analog comp. mode | 3, 5 |
| | $I_{CMP2}$ | | — | 3.1 | 4.3 | mA | $V_{CC}$ = 3 V, $f_{OSC}$ = 800 kHz Analog comp. mode | 3, 5 |
| Current dissipation in standby mode | $I_{SBY1}$ | $V_{CC}$ | — | 1.2 | — | mA | $V_{CC}$ = 5 V, $f_{OSC}$ = 4 MHz | 4, 5 |
| | $I_{SBY2}$ | | — | 0.2 | 0.7 | mA | $V_{CC}$ = 3 V $f_{OSC}$ = 800 kHz | 4, 5 |
| Current dissipation in stop mode | $I_{STOP}$ | $V_{CC}$ | — | 1 | 5 | µA | $V_{CC}$ = 3 V | 6 |

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Notes |
|---|---|---|---|---|---|---|---|---|
| Stop mode retaining voltage | $V_{STOP}$ | $V_{CC}$ | — | 1.3 | — | V | | 7 |
| Comparator input reference voltage scope | $VC_{ref}$ | $VC_{ref}$ | 0 | — | $V_{CC}$ – 1.2 | V | | |

Notes:
1. Output buffer current is excluded.
2. $I_{CC1}$ and $I_{CC2}$ are the source currents when no I/O current is flowing while the MCU is in reset state.

   Test conditions: MCU: Reset
   Pins: $\overline{RESET}$ at GND (0 to 0.3V)
   $\overline{TEST}$ at $V_{CC}$ ($V_{CC}$ –0.3 to $V_{CC}$)
3. $RD_0$, $RD_1$ pins are in analog input mode when no I/O current is flowing.

   Test conditions: MCU: DTMF does not operate
   Pins: $RD_0/COMP_0$ at GND (0 V to 0.3 V)
   $RD_1/COMP_1$ at GND (0 V to 0.3 V)
   $RE_0/VC_{ref}$ at GND (0 V to 0.3 V)
4. $I_{SBY1}$ and $I_{SBY2}$ are the source currents when no I/O current is flowing while the MCU timer is operating.

   Test conditions: MCU: I/O reset
   Serial interface stopped
   DTMF does not operate
   Standby mode
   Pins: $\overline{RESET}$ at $V_{CC}$ ($V_{CC}$ –0.3 to $V_{CC}$)
   $\overline{TEST}$ at $V_{CC}$ ($V_{CC}$ –0.3 to $V_{CC}$)
5. The current dissipation is in proportion to $f_{OSC}$ while the MCU is operating or is in standby mode. The current dissipation when $f_{OSC}$ = F MHz is given by the following equation: Maximum value ($f_{OSC}$ = F MHz) = F/4 X maximum value ($f_{OSC}$ = 4 MHz)
6. These are the source currents when no I/O current is flowing.

   Test conditions: Pins: $\overline{RESET}$ at $V_{CC}$ ($V_{CC}$ –0.3 to $V_{CC}$)
   $\overline{TEST}$ at $V_{CC}$ ($V_{CC}$ –0.3 to $V_{CC}$)
   $D_{13}$ at $V_{CC}$ ($V_{CC}$ –0.3 to $V_{CC}$)*
   Note: * Applies to HD4074654.
7. RAM data retention.

**I/O Characteristics for Standard Pins (HD404652, HD404654: $V_{CC}$ = 1.8 to 6.0 V, GND = 0 V, $T_a$ = –20 to +75°C; HD4074654: $V_{CC}$ = 2.7 to 5.5 V, GND = 0 V, $T_a$ = –20 to +75°C, unless otherwise specified)**

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Notes |
|---|---|---|---|---|---|---|---|---|
| Input high voltage | $V_{IH}$ | $D_{12}$–$D_{13}$, R0–RD, $RE_0$ | $0.7V_{CC}$ | — | $V_{CC}$ + 0.3 | V | | |
| Input low voltage | $V_{IL}$ | $D_{12}$–$D_{13}$, R0–RD, $RE_0$ | –0.3 | — | $0.3V_{CC}$ | V | | |
| Output high voltage | $V_{OH}$ | R0–R4 | $V_{CC}$ –1.0 | — | — | V | $–I_{OH}$ = 0.5 mA | |
| Output low voltage | $V_{OL}$ | R0–R4 | — | — | 0.4 | V | $I_{OL}$ = 0.4 mA | |
| I/O leakage current | $\| I_{IL} \|$ | $D_{12}$, R0–RD, $RE_0$ | — | — | 1 | µA | $V_{in}$ = 0 V to $V_{CC}$ | 1 |
| | | $D_{13}$ | — | — | 1 | µA | $V_{in}$ = 0 V to $V_{CC}$ | 1, 2 |
| | | | — | — | 1 | µA | $V_{in}$ = $V_{CC}$ – 0.3 V to $V_{CC}$ | 1, 3 |
| | | | — | — | 20 | µA | $V_{in}$ = 0 V to 0.3 V | 1, 3 |
| Pull-up MOS current | $–I_{PU}$ | R0–R4 | — | 30 | — | µA | $V_{CC}$ = 3 V, $V_{in}$ = 0 V | |
| Input high voltage | $V_{IHA}$ | $COMP_0$, $COMP_1$ | — | $VC_{ref}$ +0.05 | — | V | Analog compare mode | 4 |
| Input low | $V_{ILA}$ | $COMP_0$, $COMP_1$ | — | $VC_{ref}$ –0.05 | — | V | Analog compare mode | 4 |

Notes: 1. Output buffer current is excluded.
2. Applies to HD404652, HD404654.
3. Applies to HD4074654.
4. Use an analog input reference voltage in the range 0 V ≤ $VC_{ref}$ ≤ $V_{CC}$ – 1.2.

**I/O Characteristics for High-Current Pins (HD404652, HD404654: $V_{CC}$ = 1.8 to 6.0 V, GND = 0 V, $T_a$ = –20 to +75°C; HD4074654: $V_{CC}$ = 2.7 to 5.5 V, GND = 0 V, $T_a$ = –20 to +75°C, unless otherwise specified)**

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Note |
|---|---|---|---|---|---|---|---|---|
| Input high voltage | $V_{IH}$ | $D_0$–$D_9$ | 0.7 $V_{CC}$ | — | $V_{CC}$ + 0.3 | V | | |
| Input low voltage | $V_{IL}$ | $D_0$–$D_9$ | –0.3 | — | 0.3 $V_{CC}$ | V | | |
| Output high voltage | $V_{OH}$ | $D_0$–$D_9$ | $V_{CC}$ – 1.0 | — | — | V | $-I_{OH}$ = 0.5 mA | |
| | | $D_0$–$D_3$ | 2.0 | — | — | V | $-I_{OH}$ = 10 mA, $V_{CC}$ = 4.5 V to 6.0 V | 2 |
| Output low voltage | $V_{OL}$ | $D_0$–$D_9$ | — | — | 0.4 | V | $I_{OL}$ = 0.4 mA | |
| | | $D_4$–$D_9$ | — | — | 2.0 | V | $I_{OL}$ = 15 mA, $V_{CC}$ = 4.5 V to 6.0 V | 2 |
| I/O leakage current | $\lvert I_{IL} \rvert$ | $D_0$–$D_9$ | — | — | 1 | µA | $V_{in}$ = 0 V to $V_{CC}$ | 1 |
| Pull-down MOS current | $I_{PD}$ | $D_0$–$D_3$ | — | 30 | — | µA | $V_{CC}$ = 3 V, $V_{in}$ = 3 V | |
| Pull-up MOS current | $-I_{PU}$ | $D_4$–$D_9$ | — | 30 | — | µA | $V_{CC}$ = 3 V, $V_{in}$ = 0 V | |

Notes: 1. Output buffer current is excluded.
2. When using HD4074654, $V_{CC}$ = 4.5 V to 5.5 V.

**DTMF Characteristics (HD404652, HD404654: $V_{CC}$ = 1.8 to 6.0 V, GND = 0 V, $T_a$ = –20 to +75°C; HD4074654: $V_{CC}$ = 2.7 to 5.5 V, GND = 0 V, $T_a$ = –20 to +75°C, unless otherwise specified)**

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Note |
|---|---|---|---|---|---|---|---|---|
| Tone output voltage (1) | $V_{OR}$ | TONER | 500 | 660 | — | $mV_{rms}$ | $VT_{ref}$ – GND = 2.0 V, $R_L$ = 100 kΩ, $V_{CC}$ = 3.0 V | 1 |
| Tone output voltage (2) | $V_{OC}$ | TONEC | 520 | 690 | — | $mV_{rms}$ | $VT_{ref}$ – GND = 2.0 V, $R_L$ = 100 kΩ, $V_{CC}$ = 3.0 V | 1 |
| Tone output distortion | %DIS | | — | 3 | 7 | % | Short circuit between TONER and TONEC $R_L$ = 100 kΩ | 2 |
| Tone output ratio | $dB_{CR}$ | | — | 2.5 | — | dB | Short circuit between TONER and TONEC $R_L$ = 100 kΩ | 2 |

Notes: 1. See figure 79.
2. See figure 80.
These characteristics are guaranteed for an operating frequency ($f_{OSC}$) of 400 kHz, 800 kHz, 2 MHz, 3.58 MHz, or 4 MHz.

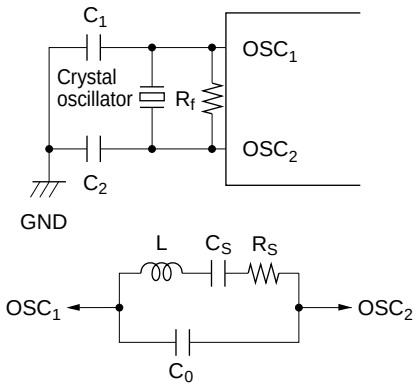
**AC Characteristics (HD404652, HD404654: $V_{CC}$ = 1.8 to 6.0 V, GND = 0 V, $T_a$ = –20 to +75°C; HD4074654: $V_{CC}$ = 2.7 to 5.5 V, GND = 0 V, $T_a$ = –20 to +75°C, unless otherwise specified)**

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Notes |
|---|---|---|---|---|---|---|---|---|
| Clock oscillation frequency | $f_{OSC}$ | $OSC_1$, $OSC_2$ | — | 400 | — | kHz | | 1 |
| | | | — | 800 | — | kHz | | 1 |
| | | | — | 2 | — | MHz | | 1 |
| | | | — | 3.58 | — | MHz | | 1 |
| | | | — | 4 | — | MHz | | 1 |
| Instruction cycle time | $t_{cyc}$ | | — | 1 | — | µs | $f_{OSC}$ = 4 MHz 1/4 division, | 2 |
| | | | — | 8 | — | µs | $f_{OSC}$ = 4 MHz, 1/32 division | 3 |
| Oscillation stabilization time | $t_{RC}$ | $OSC_1$, $OSC_2$ | — | — | 7.5 | ms | $V_{CC}$ = 2.7 V to 6.0 V | 4, 5, 13 |
| (ceramic) | | | — | — | 60 | ms | $V_{CC}$ = 1.8 V to 2.7 V | 4, 5, 12 |
| External clock high width | $t_{CPH}$ | $OSC_1$ | 1100 | — | — | ns | $f_{OSC}$ = 400 kHz | 6 |
| | | | 550 | — | — | ns | $f_{OSC}$ = 800 kHz | 6 |
| | | | 215 | — | — | ns | $f_{OSC}$ = 2 MHz | 6 |
| | | | 115 | — | — | ns | $f_{OSC}$ = 3.58 MHz | 6 |
| | | | 105 | — | — | ns | $f_{OSC}$ = 4 MHz | 6 |
| External clock low width | $t_{CPL}$ | $OSC_1$ | 1100 | — | — | ns | $f_{OSC}$ = 400 kHz | 6 |
| | | | 550 | — | — | ns | $f_{OSC}$ = 800 kHz | 6 |
| | | | 215 | — | — | ns | $f_{OSC}$ = 2 MHz | 6 |
| | | | 115 | — | — | ns | $f_{OSC}$ = 3.58 MHz | 6 |
| | | | 105 | — | — | ns | $f_{OSC}$ = 4 MHz | 6 |
| External clock rise time | $t_{CPr}$ | $OSC_1$ | — | — | 150 | ns | $f_{OSC}$ = 400 kHz | 6 |
| | | | — | — | 75 | ns | $f_{OSC}$ = 800 kHz | 6 |
| | | | — | — | 35 | ns | $f_{OSC}$ = 2 MHz | 6 |
| | | | — | — | 25 | ns | $f_{OSC}$ = 3.58 MHz | 6 |
| | | | — | — | 20 | ns | $f_{OSC}$ = 4 MHz | 6 |

| Item | Symbol | Pin(s) | Min | Typ | Max | Unit | Test Condition | Notes |
|---|---|---|---|---|---|---|---|---|
| External clock fall time | $t_{CPf}$ | $OSC_1$ | — | — | 150 | ns | $f_{OSC}$ = 400 kHz | 6 |
| | | | — | — | 75 | ns | $f_{OSC}$ = 800 kHz | 6 |
| | | | — | — | 35 | ns | $f_{OSC}$ = 2 MHz | 6 |
| | | | — | — | 25 | ns | $f_{OSC}$ = 3.58 MHz | 6 |
| | | | — | — | 20 | ns | $f_{OSC}$ = 4 MHz | 6 |
| $\overline{INT_0}$, $\overline{INT_1}$, EVND high width | $t_{IH}$ | $\overline{INT_0}$, $\overline{INT_1}$, EVND | 2 | — | — | $t_{cyc}$ | | 7 |
| $\overline{INT_0}$, $\overline{INT_1}$, EVND low width | $t_{IL}$ | $\overline{INT_0}$, $\overline{INT_1}$, EVND | 2 | — | — | $t_{cyc}$ | | 7 |
| $\overline{RESET}$ low width | $t_{RSTL}$ | $\overline{RESET}$ | 2 | — | — | $t_{cyc}$ | | 8 |
| $\overline{STOPC}$ low width | $t_{STPL}$ | $\overline{STOPC}$ | 1 | — | — | $t_{RC}$ | | 9 |
| $\overline{RESET}$ rise time | $t_{RSTr}$ | $\overline{RESET}$ | — | — | 20 | ms | | 8 |
| $\overline{STOPC}$ rise time | $t_{STPr}$ | $\overline{STOPC}$ | — | — | 20 | ms | | 9 |
| Input capacitance | $C_{in}$ | All pins except $D_{13}$, $D_4$–$D_7$ | — | — | 15 | pF | f = 1 MHz, $V_{in}$ = 0 V | |
| | | $D_4$–$D_7$ | — | — | 30 | pF | | |
| | | $D_{13}$ | — | — | 15 | pF | | |
| | | | — | — | 180 | pF | | 10 |
| Analog comparator stabilization time | $t_{CSTB}$ | $COMP_0$, $COMP_1$ | — | — | 2 | $t_{cyc}$ | $V_{CC}$ = 2.7 V to 6.0 V | 11, 12 |
| | | | — | — | 20 | $t_{cyc}$ | $V_{CC}$ = 1.8 V to 2.7 V | |

Notes:
1. Bits 0 and 1 (SSR10, SSR11) of system clock select register 1 (SSR1: $029) and bit 2 (SSR22) of system clock select register 2 (SSR2: $02A) must be set according to the system clock frequency.
2. SEL = 1
3. SEL = 0
4. The oscillation stabilization time is the period required for the oscillator to stabilize after $V_{CC}$ reaches 2.7 V (1.8 V for HD404654 and HD404652) at power-on, after $\overline{RESET}$ input goes low when stop mode is cancelled, or after $\overline{STOPC}$ input goes low when stop mode is cancelled. At power-on or when stop mode is cancelled, $\overline{RESET}$ or $\overline{STOPC}$ must be input for at least $t_{RC}$ to ensure the oscillation stabilization time. If using a ceramic oscillator, contact its manufacturer to determine what stabilization time is required, since it will depend on the circuit constants and stray capacitance.

5. Applies to ceramic oscillator only.
   When using crystal oscillator: $V_{CC}$ = 2.7 V to 6.0 V, $t_{RC}$ = 40 ms (typ) or $V_{CC}$ = 1.8 V to 2.6 V, $t_{RC}$ = 60 ms (typ)

| | | |
|---|---|---|
| Crystal oscillator<br>($OSC_1$, $OSC_2$) |  | $R_f$ = 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<br>f = 400 kHz, 800 kHz, 2 MHz, 3.58 MHz, 4 MHz |

- 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.
- Wiring among $OSC_1$, $OSC_2$, and elements should be as short as possible, and must not cross other wiring (see figure 20).

6. Refer to figure 81.
7. Refer to figure 82.
8. Refer to figure 83.
9. Refer to figure 84.
10. Applies to HD4074654.
11. Analog comparator stabilization time is the period for the analog comparator to stabilize and for correct data to be read after entering $RD_0$/$COMP_0$ and $RD_1$/$COMP_1$ into analog input mode.
12. HD4074654 : $V_{CC}$ = 2.7 V to 5.5 V

**Serial Interface Timing Characteristics (HD404652, HD404654: $V_{CC}$ = 1.8 to 6.0 V, GND = 0 V, $T_a$ = –20 to +75°C; HD4074654: $V_{CC}$ = 2.7 to 5.5 V, GND = 0 V, $T_a$ = –20 to +75°C, unless otherwise specified)**

**During Transmit Clock Output**

| Item | Symbol | Pin (s) | Min | Typ | Max | Unit | Test Condition | Note |
|---|---|---|---|---|---|---|---|---|
| Transmit clock cycle time | $t_{Scyc}$ | $\overline{SCK}_1$ | 1 | — | — | $t_{cyc}$ | Load shown in figure 86 | 1 |
| Transmit clock high width | $t_{SCKH}$ | $\overline{SCK}_1$ | 0.5 | — | — | $t_{Scyc}$ | Load shown in figure 86 | 1 |
| Transmit clock low width | $t_{SCKL}$ | $\overline{SCK}_1$ | 0.5 | — | — | $t_{Scyc}$ | Load shown in figure 86 | 1 |
| Transmit clock rise time | $t_{SCKr}$ | $\overline{SCK}_1$ | — | 100 | — | ns | Load shown in figure 86 | 1 |
| Transmit clock fall time | $t_{SCKf}$ | $\overline{SCK}_1$ | — | 100 | — | ns | Load shown in figure 86 | 1 |
| Serial output data delay time | $t_{DSO}$ | $SO_1$ | — | — | 500 | ns | Load shown in figure 86 | 1 |
| Serial input data setup time | $t_{SSI}$ | $SI_1$ | 300 | — | — | ns | | 1 |
| Serial input data hold time | $t_{HSI}$ | $SI_1$ | 300 | — | — | ns | | 1 |

Note: 1. Refer to figure 85.

**During Transmit Clock Input**

| Item | Symbol | Pin (s) | Min | Typ | Max | Unit | Test Condition | Note |
|---|---|---|---|---|---|---|---|---|
| Transmit clock cycle time | $t_{Scyc}$ | $\overline{SCK}_1$ | 1 | — | — | $t_{cyc}$ | | 1 |
| Transmit clock high width | $t_{SCKH}$ | $\overline{SCK}_1$ | 0.5 | — | — | $t_{Scyc}$ | | 1 |
| Transmit clock low width | $t_{SCKL}$ | $\overline{SCK}_1$ | 0.5 | — | — | $t_{Scyc}$ | | 1 |
| Transmit clock rise time | $t_{SCKr}$ | $\overline{SCK}_1$ | — | 100 | — | ns | | 1 |
| Transmit clock fall time | $t_{SCKf}$ | $\overline{SCK}_1$ | — | 100 | — | ns | | 1 |
| Serial output data delay time | $t_{DSO}$ | $SO_1$ | — | — | 500 | ns | Load shown in figure 86 | 1 |
| Serial input data setup time | $t_{SSI}$ | $SI_1$ | 300 | — | — | ns | | 1 |
| Serial input data hold time | $t_{HSI}$ | $SI_1$ | 300 | — | — | ns | | 1 |

Note: 1. Refer to figure 85.

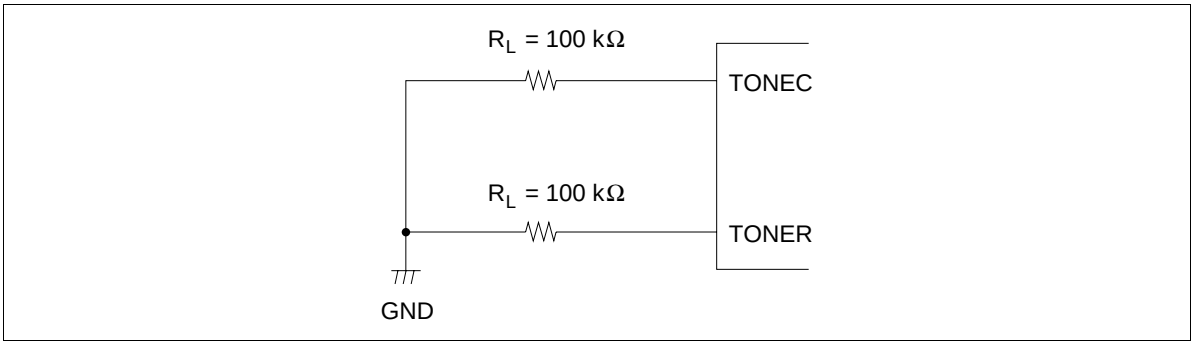



**Figure 79 TONE Output Load Circuit**

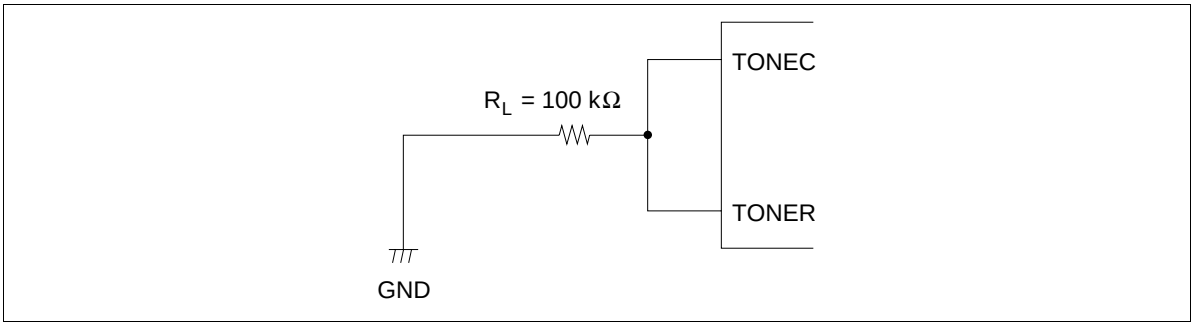



**Figure 80 Distortion $dB_{CR}$ Load Circuit**

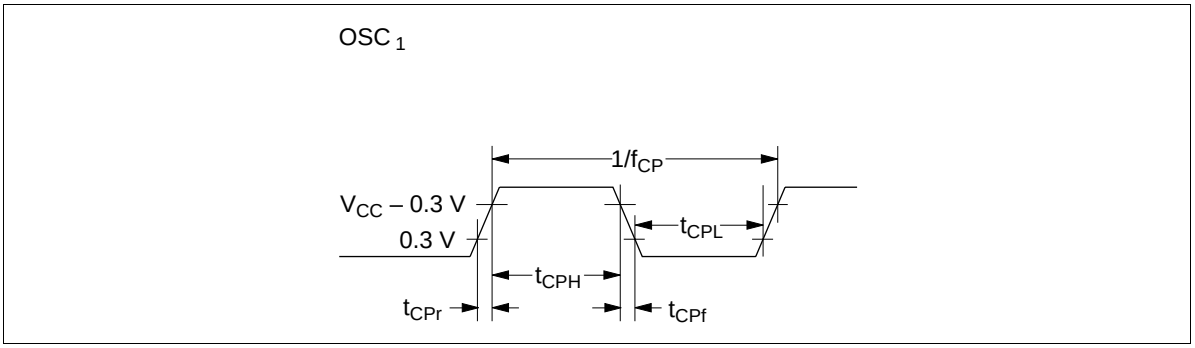



**Figure 81 External Clock Timing**

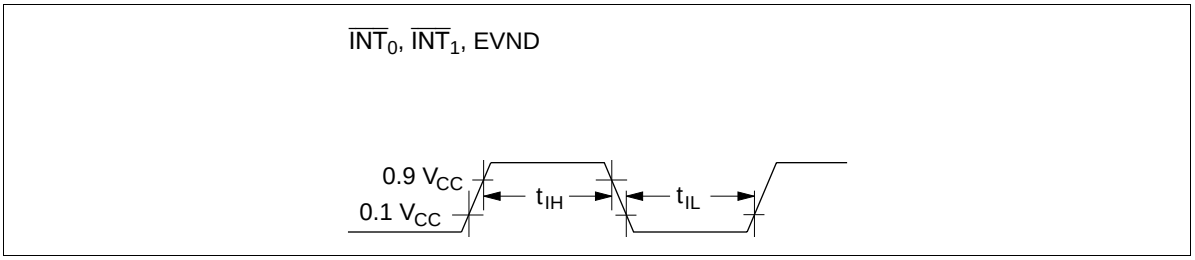



**Figure 82 Interrupt Timing**

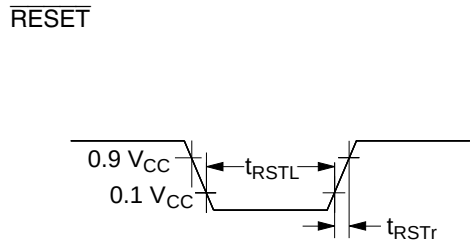



**Figure 83 Reset Timing**

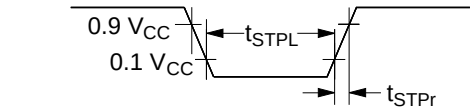
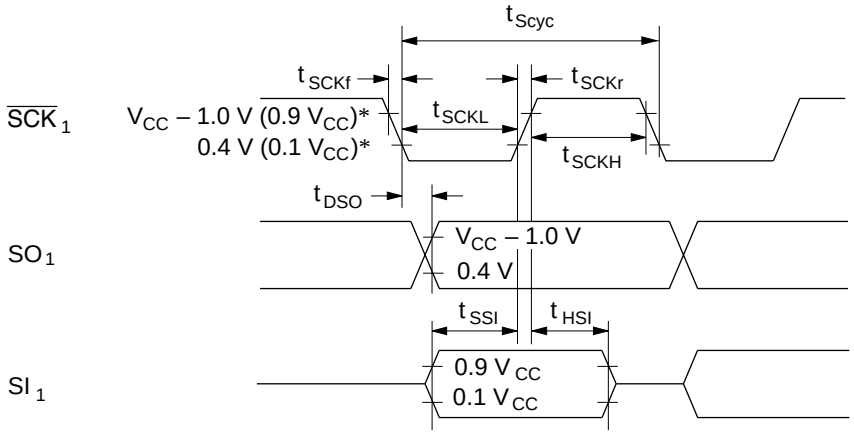



**Figure 84 $\overline{\text{STOPC}}$ Timing**




Note: * $V_{CC}$ – 1.0 V and 0.4 V are the threshold voltages for transmit clock output.
0.9 $V_{CC}$ and 0.1 $V_{CC}$ are the threshold voltages for transmit clock output.

**Figure 85 Serial Interface Timing**

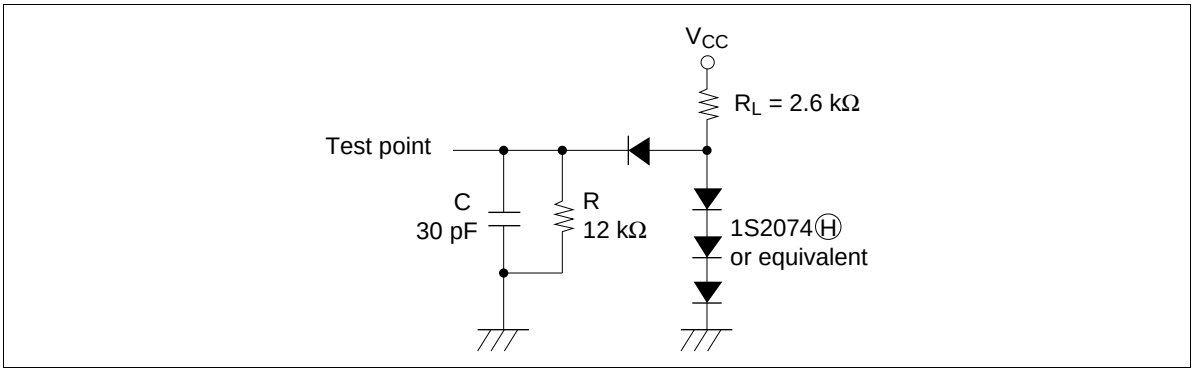


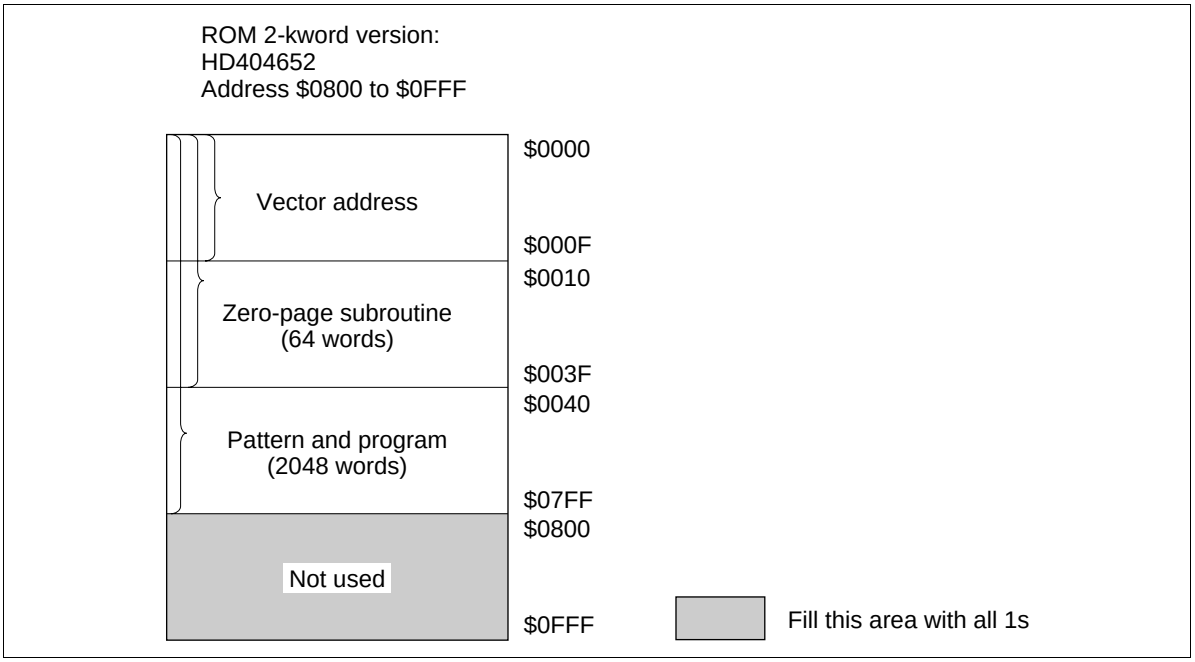

**Figure 86 Timing Load Circuit**

## Notes On ROM Out

Please pay attention to the following items regard  ing ROM out.

On ROM out, fill the ROM area indicated below with 1s to create the same data size as 4-kword version (HD404654). A 4-kword data size is required to change ROM data to mask manufacturing data since the program used is for a 4-kword version.

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



## HD404652/HD404654 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

| | | |
|---|---|---|
| ☐ | HD404652 | 2-kword |
| ☐ | HD404654 | 4-kword |

5. ROM Code Media
   Please specify the first type 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. |

6. Oscillator for OSC1 and OSC2

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

7. Stop Mode

| | |
|---|---|
| ☐ | Used |
| ☐ | Not used |

8. Package

| | |
|---|---|
| ☐ | DP-42S |
| ☐ | FP-44A |

## 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.