

DESCRIPTION

The 4552 Group is a 4-bit single-chip microcomputer designed with CMOS technology. Its CPU is that of the 4500 series using a simple, high-speed instruction set. The computer is equipped with two 8-bit timers (each timer has one or two reload registers), a 16-bit timer for clock count, interrupts, and oscillation circuit switch function.

The various microcomputers in the 4552 Group include variations of the built-in memory size as shown in the table below.

FEATURES

- Minimum instruction execution time
 - Mask ROM version 0.5 μ s
(at 6 MHz oscillation frequency, in high-speed through-mode)
 - One Time PROM version 0.68 μ s
(at 4.4 MHz oscillation frequency, in high-speed through-mode)
- Supply voltage
 - Mask ROM version 1.8 to 5.5 V
 - One Time PROM version 1.8 to 3.6 V
(It depends on operation source clock, oscillation frequency and operation mode)
- Timers
 - Timer 1 8-bit timer with a reload register
 - Timer 2 8-bit timer with two reload registers
 - Timer 3 16-bit timer (fixed dividing frequency)

- Interrupt 4 sources
- Key-on wakeup function pins 9
- LCD control circuit
 - Segment output 28
 - Common output 4
- Voltage drop detection circuit (only H version)
 - Reset occurrence Typ. 1.8 V (Ta = 25 °C)
 - Reset release Typ. 1.9 V (Ta = 25 °C)
- Watchdog timer
- Clock generating circuit
 - Built-in clock
(on-chip oscillator)
 - Main clock
(ceramic resonator/RC oscillation)
 - Sub-clock
(quartz-crystal oscillation)
- LED drive directly enabled (port D)

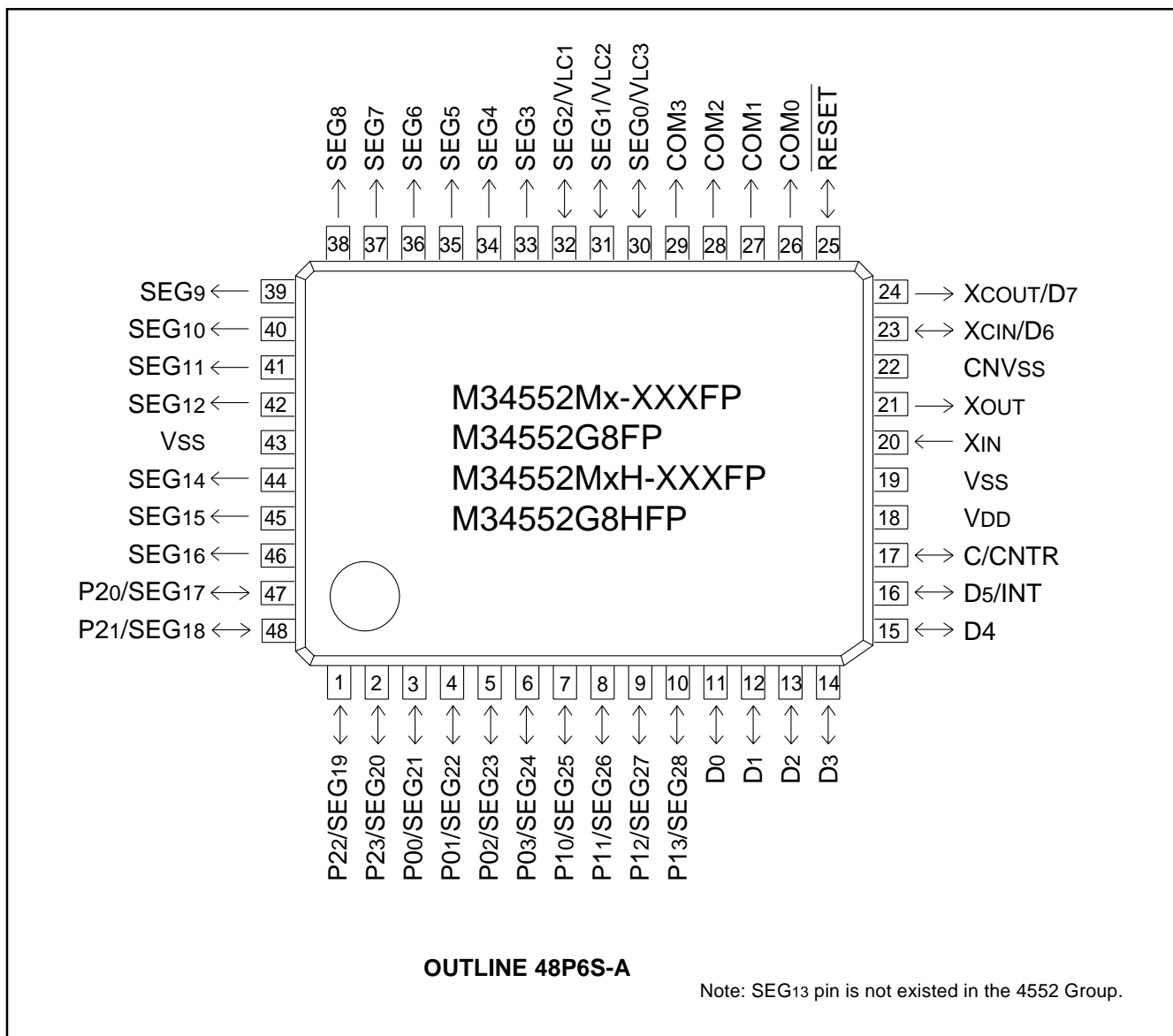
APPLICATION

Remote control transmitter

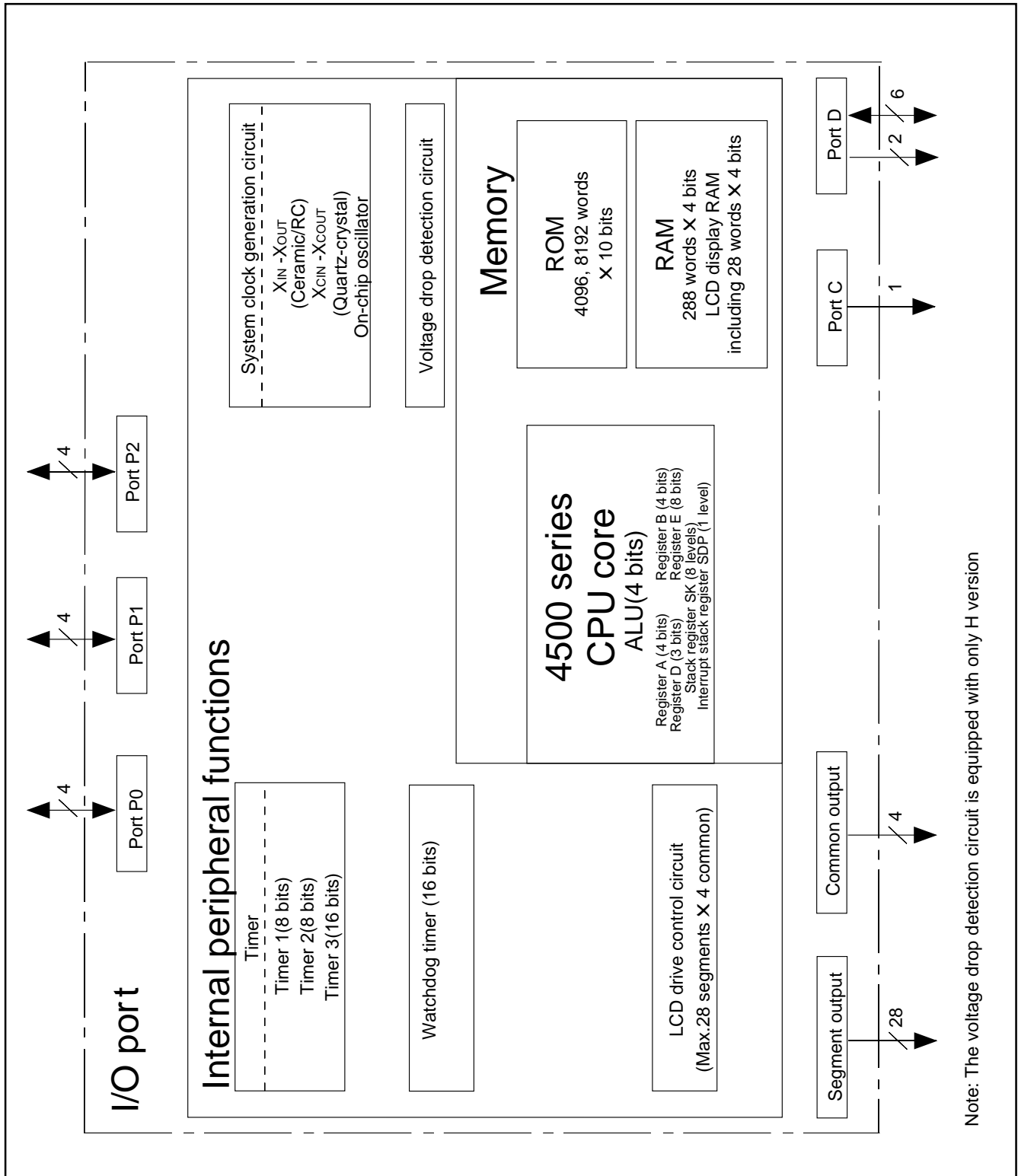
| Part number | | ROM (PROM) size (X 10 bits) | RAM size (X 4 bits) | Package | ROM type |
|-------------|-----------------------------|--------------------------------|------------------------|---------|---------------|
| 4552 Group | M34552M4-XXXFP | 4096 words | 288 words | 48P6S-A | Mask ROM |
| | M34552M8-XXXFP | 8192 words | 288 words | 48P6S-A | Mask ROM |
| | M34552G8FP (Note) | 8192 words | 288 words | 48P6S-A | One Time PROM |
| | M34552M4H-XXXFP | 4096 words | 288 words | 48P6S-A | Mask ROM |
| | M34552M8H-XXXFP | 8192 words | 288 words | 48P6S-A | Mask ROM |
| | M34552G8HFP (Note) | 8192 words | 288 words | 48P6S-A | One Time PROM |

Note: Shipped in blank.

PIN CONFIGURATION



Pin configuration (top view) (4552 Group)



Note: The voltage drop detection circuit is equipped with only H version

Block diagram (4552 Group)

PERFORMANCE OVERVIEW

| Parameter | | Function | |
|------------------------------------|--|---|--|
| Number of basic instructions | M34552M4/M8/G8 | 123 | |
| | M34552M4H/M8H/G8H | 124 | |
| Minimum instruction execution time | Mask ROM version | 0.5 μ s (at 6 MHz oscillation frequency, in high-speed through mode) | |
| | One Time PROM version | 0.68 μ s (at 4.4 MHz oscillation frequency, in high-speed through mode) | |
| Memory sizes | ROM | M34552M4 | 4096 words X 10 bits |
| | | M34552M4H | |
| | | M34552M8/G8 | 8192 words X 10 bits |
| | | M34552M8H/G8H | |
| | RAM | M34552M4/M8/G8 | 288 words X 4 bits (including LCD display RAM 28 words X 4 bits) |
| | | M34552M4H/M8H/G8H | |
| Input/Output ports | D0–D5 | I/O | Six independent I/O ports. Input is examined by skip decision. The output structure can be switched by software. Port D5 is also used as INT pin. |
| | D6, D7 | Output | Two independent output ports. Ports D6 and D7 are also used as XCIN and XCOUT, respectively. |
| | P00–P03 | I/O | 4-bit I/O port; A pull-up function, a key-on wakeup function and output structure can be switched by software. Ports P00–P03 are also used as SEG21–SEG24, respectively. |
| | P10–P13 | I/O | 4-bit I/O port; A pull-up function, a key-on wakeup function and output structure can be switched by software. Ports P10–P13 are also used as SEG25–SEG28, respectively. |
| | P20–P23 | I/O | 4-bit I/O port; The output structure can be switched by software. Ports P20–P23 are also used as SEG17–SEG20, respectively. |
| | C | Output | 1-bit output; Port C is also used as CNTR pin. |
| Timers | Timer 1 | | 8-bit programmable timer with a reload register and has an event counter. |
| | Timer 2 | | 8-bit programmable timer with two reload registers and PWM output function. |
| | Timer 3 | | 16-bit timer, fixed dividing frequency (timer for clock count) |
| | Timer LC | | 4-bit timer with a reload register (for LCD clock) |
| | Watchdog timer | | 16-bit timer (fixed dividing frequency) (for watchdog) |
| LCD control circuit | Selective bias value | | 1/2, 1/3 bias |
| | Selective duty value | | 2, 3, 4 duty |
| | Common output | | 4 |
| | Segment output | | 28 |
| | Internal resistor for power supply | | $2r \times 3$, $2r \times 2$, $r \times 3$, $r \times 2$ ($r = 80 \text{ k}\Omega$, ($T_a = 25 \text{ }^\circ\text{C}$, Typical value)) |
| Interrupt | Sources | | 4 (one for external, three for timer) |
| | Nesting | | 1 level |
| Subroutine nesting | | | 8 levels |
| Device structure | | | CMOS silicon gate |
| Package | | | 48-pin plastic molded QFP (48P6S-A) |
| Operating temperature range | | | -20 $^\circ\text{C}$ to 85 $^\circ\text{C}$ |
| Supply voltage | Mask ROM version | | 1.8 to 5.5 V (It depends on operation source clock, oscillation frequency and operation mode) |
| | One Time PROM version | | 1.8 to 3.6 V (It depends on operation source clock, oscillation frequency and operation mode) |
| Power dissipation (Typ. value) | Active mode (Mask ROM version) | | 2.2 mA (at room temperature, $V_{DD} = 5 \text{ V}$, $f(XIN) = 6 \text{ MHz}$, $f(XCIN) = \text{stop}$, $f(RING) = \text{stop}$, $f(STCK) = f(XIN)/1$) |
| | At clock operating mode (Mask ROM version) | | 6 μA (at room temperature, $V_{DD} = 5 \text{ V}$, $f(XCIN) = 32 \text{ kHz}$) |
| | At RAM back-up (Mask ROM version) | | 0.1 μA (at room temperature, $V_{DD} = 5 \text{ V}$, output transistor is cut-off state) |

PIN DESCRIPTION

| Pin | Name | Input/Output | Function |
|----------------------|---|--------------|--|
| VDD | Power supply | — | Connected to a plus power supply. |
| VSS | Ground | — | Connected to a 0 V power supply. |
| CNVSS | CNVSS | — | Connect CNVSS to VSS and apply "L" (0V) to CNVSS certainly. |
| RESET | Reset input/output | I/O | An N-channel open-drain I/O pin for a system reset. When the SRST instruction, watchdog timer, the built-in power-on reset or the voltage drop detection circuit causes the system to be reset, the RESET pin outputs "L" level. |
| XIN | Main clock input | Input | I/O pins of the main clock generating circuit. When using a ceramic resonator, connect it between pins XIN and XOUT. A feedback resistor is built-in between them. When using the RC oscillation, connect a resistor and a capacitor to XIN, and leave XOUT pin open. |
| XOUT | Main clock output | Output | |
| XCIN | Sub-clock input | Input | I/O pins of the sub-clock generating circuit. Connect a 32.768 kHz quartz-crystal oscillator between pins XCIN and XCOU. A feedback resistor is built-in between them. XCIN and XCOU pins are also used as ports D6 and D7, respectively. |
| XCOU | Sub-clock output | Output | |
| D0–D5 | I/O port D Input is examined by skip decision. | I/O | Each pin of port D has an independent 1-bit wide I/O function. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Port D5 is also used as INT pin. |
| D6, D7 | Output port D | Output | Each pin of port D has an independent 1-bit wide output function. The output structure is N-channel open-drain. Ports D6 and D7 are also used as XCIN pin and XCOU pin, respectively. |
| P00–P03 | I/O port P0 | I/O | Port P0 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Port P0 has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports P00–P03 are also used as SEG21–SEG24, respectively. |
| P10–P13 | I/O port P1 | I/O | Port P1 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Port P1 has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports P10–P13 are also used as SEG25–SEG28, respectively. |
| P20–P23 | I/O port P2 | I/O | Port P2 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Ports P20–P23 are also used as SEG17–SEG20, respectively. |
| Port C | Output port C | Output | 1-bit output port. The output structure is CMOS. Port C is also used as CNTR pin. |
| COM0–COM3 | Common output | Output | LCD common output pins. Pins COM0 and COM1 are used at 1/2 duty, pins COM0–COM2 are used at 1/3 duty and pins COM0–COM3 are used at 1/4 duty. |
| SEG0–SEG28 (Note) | Segment output | Output | LCD segment output pins. SEG0–SEG2 pins are used as VLC3–VLC1 pins, respectively. SEG17–SEG28 pins are used as Ports P20–P23, Ports P00–P03 and Ports P10–P13, respectively. |
| CNTR | Timer input/output | I/O | CNTR pin has the function to input the clock for the timer 1 event counter and to output the PWM signal generated by timer 2. CNTR pin is also used as Port C. |
| INT | Interrupt input | Input | INT pin accepts external interrupts. They have the key-on wakeup function which can be switched by software. INT pin is also used as Port D5. |

Note: SEG13 pin is not existed in the 4552 Group.

MULTIFUNCTION

| Pin | Multifunction | Pin | Multifunction | Pin | Multifunction | Pin | Multifunction |
|------|---------------|-------|---------------|------|---------------|-------|---------------|
| XCIN | D6 | D6 | XCIN | P20 | SEG17 | SEG17 | P20 |
| XCOU | D7 | D7 | XCOU | P21 | SEG18 | SEG18 | P21 |
| P00 | SEG21 | SEG21 | P00 | P22 | SEG19 | SEG19 | P22 |
| P01 | SEG22 | SEG22 | P01 | P23 | SEG20 | SEG20 | P23 |
| P02 | SEG23 | SEG23 | P02 | D5 | INT | INT | D5 |
| P03 | SEG24 | SEG24 | P03 | C | CNTR | CNTR | C |
| P10 | SEG25 | SEG25 | P10 | SEG0 | VLC3 | VLC3 | SEG0 |
| P11 | SEG26 | SEG26 | P11 | SEG1 | VLC2 | VLC2 | SEG1 |
| P12 | SEG27 | SEG27 | P12 | SEG2 | VLC1 | VLC1 | SEG2 |
| P13 | SEG28 | SEG28 | P13 | | | | |

Notes 1: Pins except above have just single function.

2: The input/output of D5 can be used even when INT is selected.

The threshold value is different between port D5 and INT. Accordingly, be careful when the input of both is used.

3: The port C "H" output function can be used even when CNTR (output) is selected.

DEFINITION OF CLOCK AND CYCLE

● Operation source clock

The operation source clock is the source clock to operate this product. In this product, the following clocks are used.

- Clock ($f(XIN)$) by the external ceramic resonator
- Clock ($f(XIN)$) by the external RC oscillation
- Clock ($f(XIN)$) by the external input
- Clock ($f(RING)$) of the on-chip oscillator which is the internal oscillator
- Clock ($f(XCIN)$) by the external quartz-crystal resonator

● System clock (STCK)

The system clock is the basic clock for controlling this product. The system clock is selected by the clock control register MR shown as the table below.

● Instruction clock (INSTCK)

The instruction clock is the basic clock for controlling CPU. The instruction clock (INSTCK) is a signal derived by dividing the system clock (STCK) by 3. The one instruction clock cycle generates the one machine cycle.

● Machine cycle

The machine cycle is the standard cycle required to execute the instruction.

Table Selection of system clock

| Register MR | | | | System clock | Operation mode |
|-------------|-----|-----|-----|-----------------------|--|
| MR3 | MR2 | MR1 | MR0 | | |
| 1 | 1 | 0 | 0 | $f(STCK) = f(RING)/8$ | Internal frequency divided by 8 mode |
| 1 | 0 | 0 | 0 | $f(STCK) = f(RING)/4$ | Internal frequency divided by 4 mode |
| 0 | 1 | 0 | 0 | $f(STCK) = f(RING)/2$ | Internal frequency divided by 2 mode |
| 0 | 0 | 0 | 0 | $f(STCK) = f(RING)$ | Internal frequency through mode |
| 1 | 1 | 0 | 1 | $f(STCK) = f(XIN)/8$ | High-speed frequency divided by 8 mode |
| 1 | 0 | 0 | 1 | $f(STCK) = f(XIN)/4$ | High-speed frequency divided by 4 mode |
| 0 | 1 | 0 | 1 | $f(STCK) = f(XIN)/2$ | High-speed frequency divided by 2 mode |
| 0 | 0 | 0 | 1 | $f(STCK) = f(XIN)$ | High-speed through mode |
| 1 | 1 | 1 | 0 | $f(STCK) = f(XCIN)/8$ | Low-speed frequency divided by 8 mode |
| 1 | 0 | 1 | 0 | $f(STCK) = f(XCIN)/4$ | Low-speed frequency divided by 4 mode |
| 0 | 1 | 1 | 0 | $f(STCK) = f(XCIN)/2$ | Low-speed frequency divided by 2 mode |
| 0 | 0 | 1 | 0 | $f(STCK) = f(XCIN)$ | Low-speed through mode |

Note: The $f(RING)/8$ is selected after system is released from reset.

PORT FUNCTION

| Port | Pin | Input Output | Output structure | I/O unit | Control instructions | Control registers | Remark |
|---------|---------------------|--------------|---------------------------|----------|----------------------|--------------------------|--|
| Port D | D0–D4, D5/INT | I/O (6) | N-channel open-drain/CMOS | 1 | SD, RD SZD CLD | FR1, FR2 I1, K2 | Output structure selection function (programmable) |
| | XCIN/D6, XCOU/D7 | Output (2) | N-channel open-drain | | | RG | |
| Port P0 | P00/SEG21–P03/SEG24 | I/O (4) | N-channel open-drain/CMOS | 4 | OP0A IAP0 | FR0, PU0 K0 C1 | Built-in pull-up functions, key-on wakeup functions and output structure selection function (programmable) |
| Port P1 | P10/SEG25–P13/SEG28 | I/O (4) | N-channel open-drain/CMOS | 4 | OP1A IAP1 | FR0, PU1 K0, K1 C2 | Built-in pull-up functions, key-on wakeup functions and output structure selection function (programmable) |
| Port P2 | P20/SEG17–P23/SEG20 | I/O (4) | N-channel open-drain/CMOS | 4 | OP2A IAP2 | FR2 L3 | Output structure selection function (programmable) |
| Port C | C/CNTR | Output (1) | CMOS | 1 | RCP SCP | W1 | |

CONNECTIONS OF UNUSED PINS

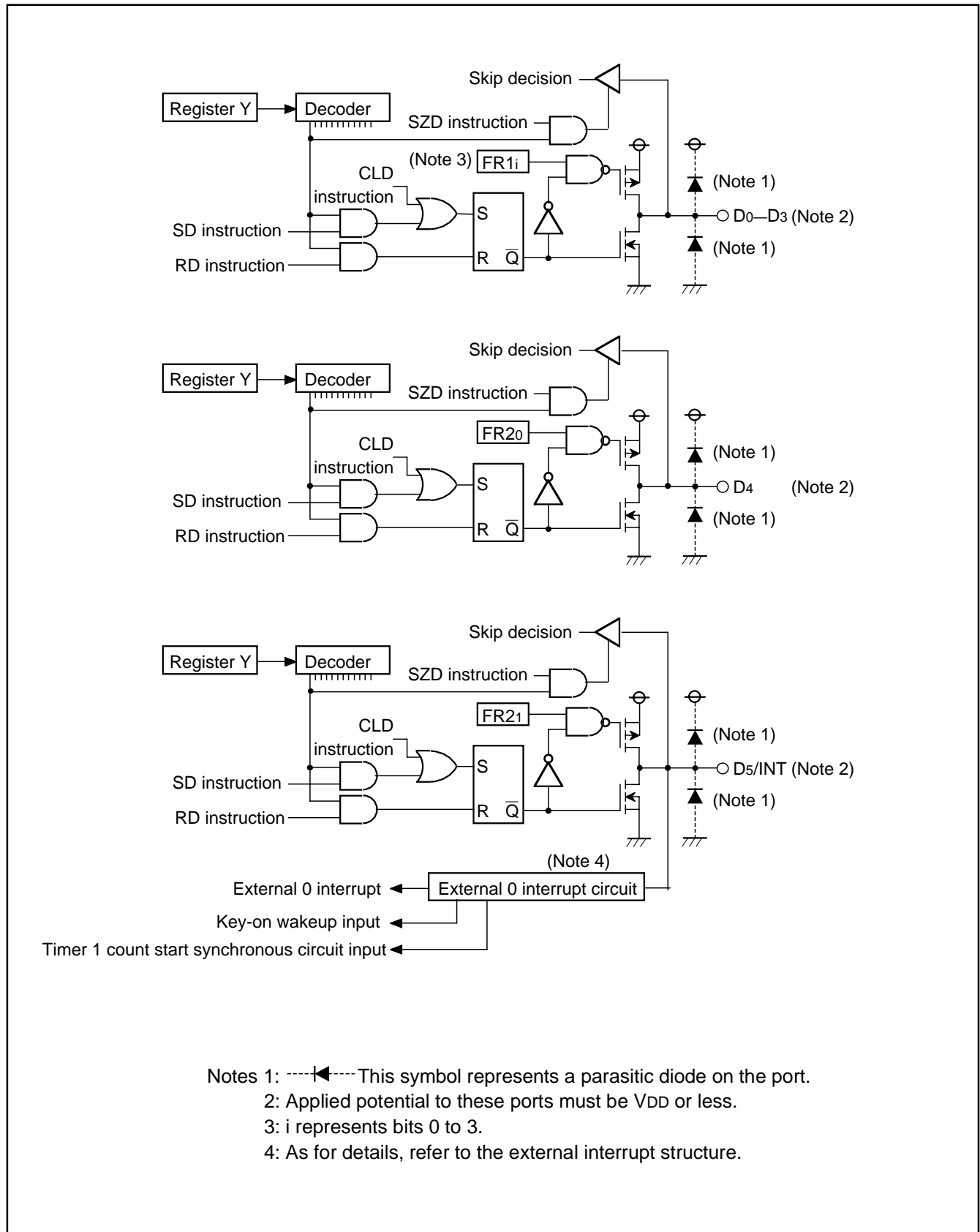
| Pin | Connection | Usage condition |
|-------------------------|-----------------|---|
| XIN | Connect to VSS. | RC oscillator is not selected |
| XOUT | Open. | ————— |
| XCIN/D6 | Connect to VSS. | ————— |
| XCOUT/D7 | Open. | ————— |
| D0–D4 | Open. | ————— |
| | Connect to VSS. | N-channel open-drain is selected for the output structure. |
| D5/INT | Open. | INT pin input is disabled. |
| | Connect to VSS. | N-channel open-drain is selected for the output structure. |
| C/CNTR | Open. | CNTR input is not selected for timer 1 count source. |
| P00/SEG21– P03/SEG24 | Open. | The key-on wakeup function is invalid. |
| | Connect to VSS. | Segment output is not selected. N-channel open-drain is selected for the output structure. Pull-up transistor is OFF. The key-on wakeup function is invalid. |
| P10/SEG25– P13/SEG28 | Open. | The key-on wakeup function is invalid. |
| | Connect to VSS. | Segment output is not selected. N-channel open-drain is selected for the output structure. Pull-up transistor is OFF. The key-on wakeup function is invalid. |
| P20/SEG17– P23/SEG20 | Open. | ————— |
| | Connect to VSS. | Segment output is not selected. N-channel open-drain is selected for the output structure. |
| COM0–COM3 | Open. | ————— |
| SEG0/VLC3 | Open. | SEG0 pin is selected. |
| SEG1/VLC2 | Open. | SEG1 pin is selected. |
| SEG2/VLC1 | Open. | SEG2 pin is selected. |
| SEG3–SEG16 (Note) | Open. | ————— |

Note: SEG13 pin is not existed in the 4552 Group.

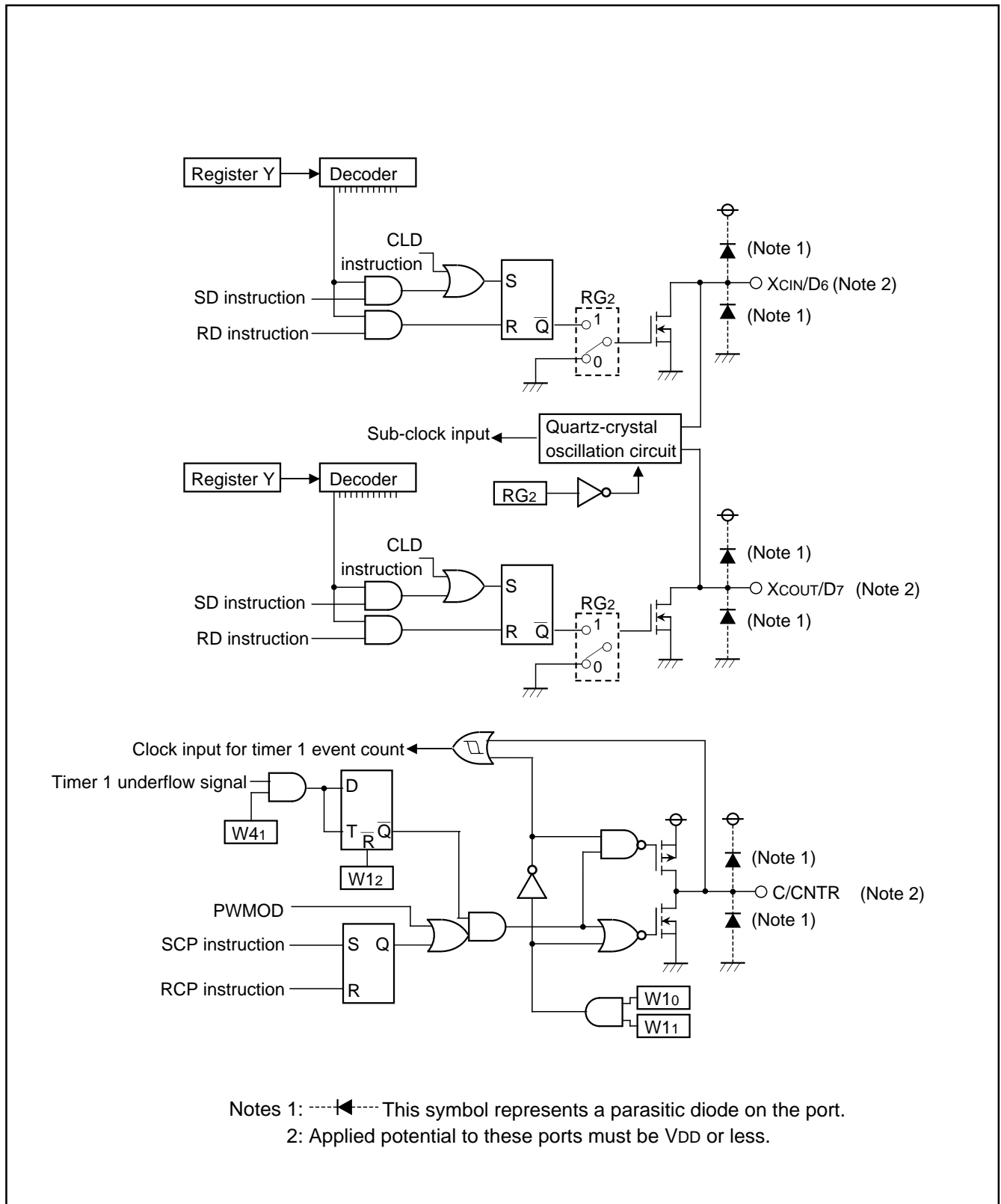
(Note when connecting to VSS and VDD)

- Connect the unused pins to VSS and VDD using the thickest wire at the shortest distance against noise.

PORT BLOCK DIAGRAMS

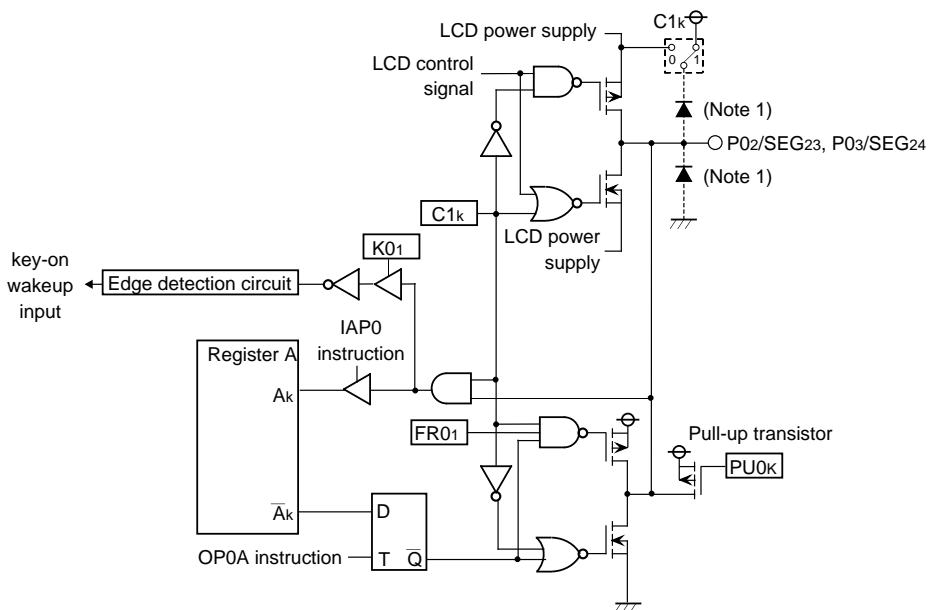
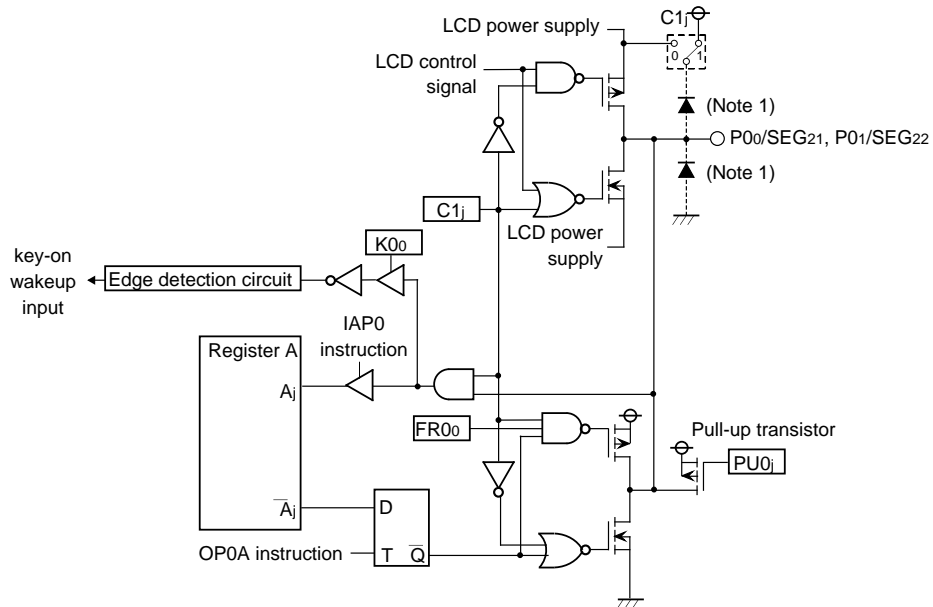


Port block diagram (1)



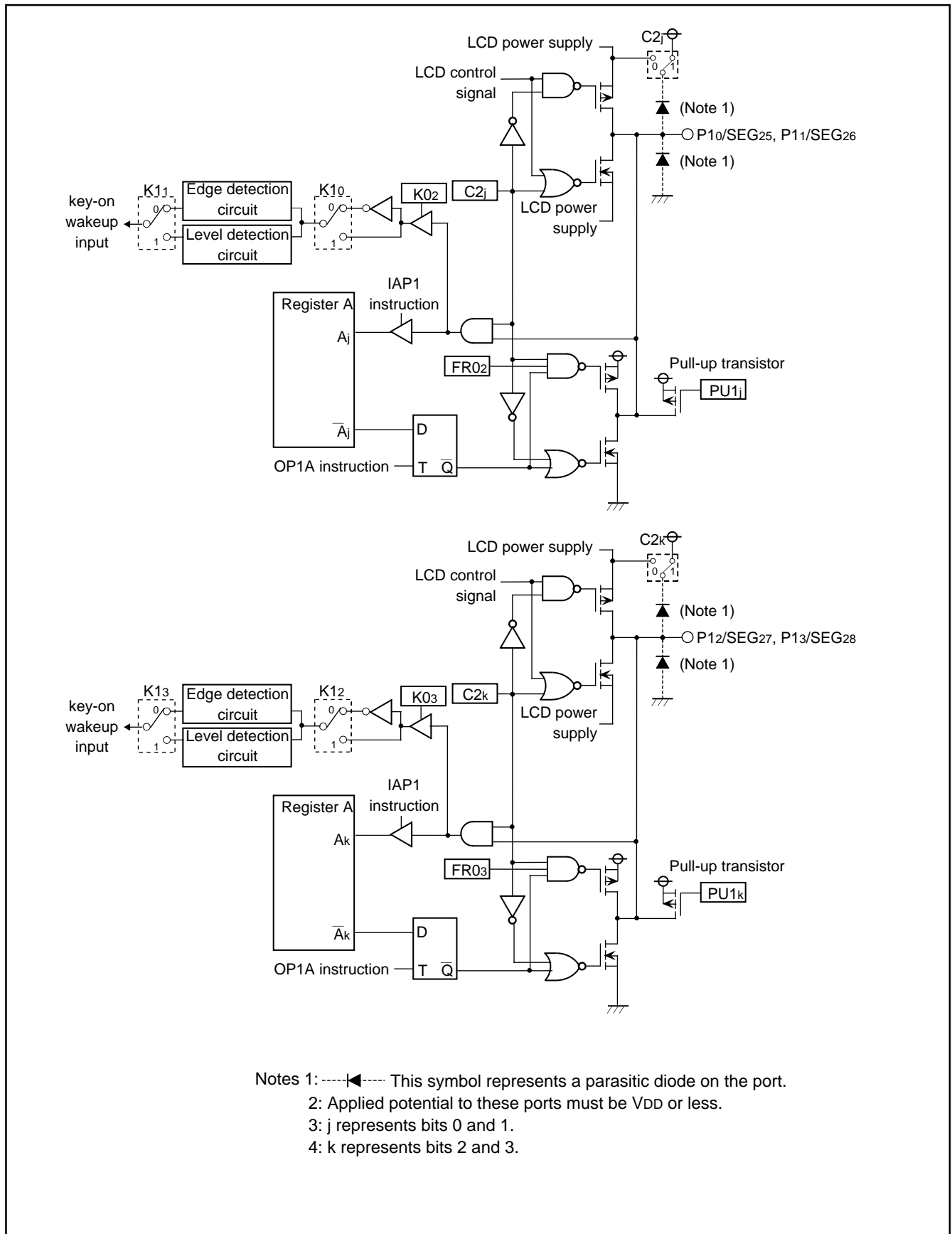
Notes 1: -----<----- This symbol represents a parasitic diode on the port.
 2: Applied potential to these ports must be VDD or less.

Port block diagram (2)

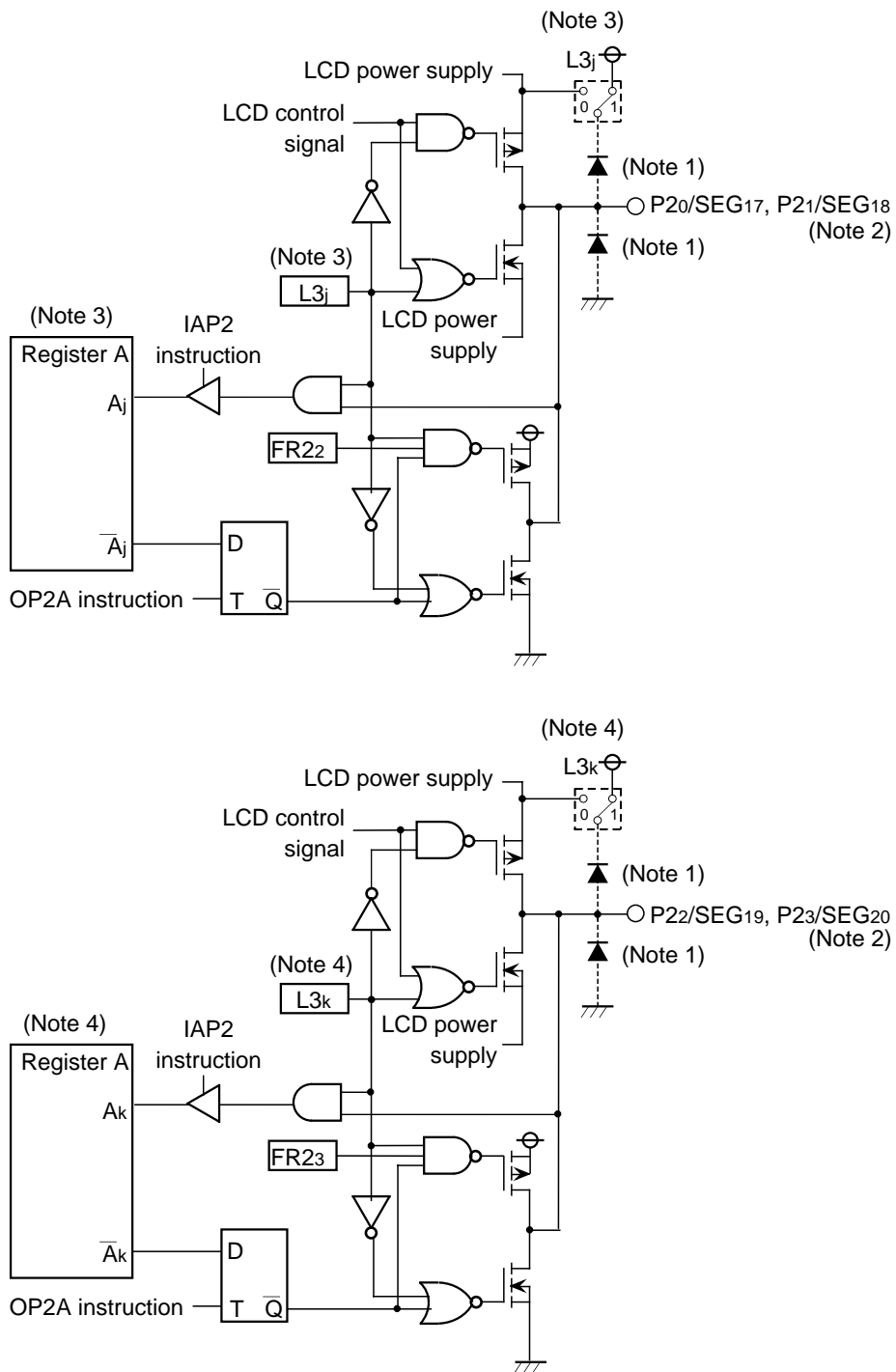



- Notes 1: -----<----- This symbol represents a parasitic diode on the port.
 2: Applied potential to these ports must be VDD or less.
 3: j represents bits 0 and 1.
 4: k represents bits 2 and 3.

Port block diagram (3)

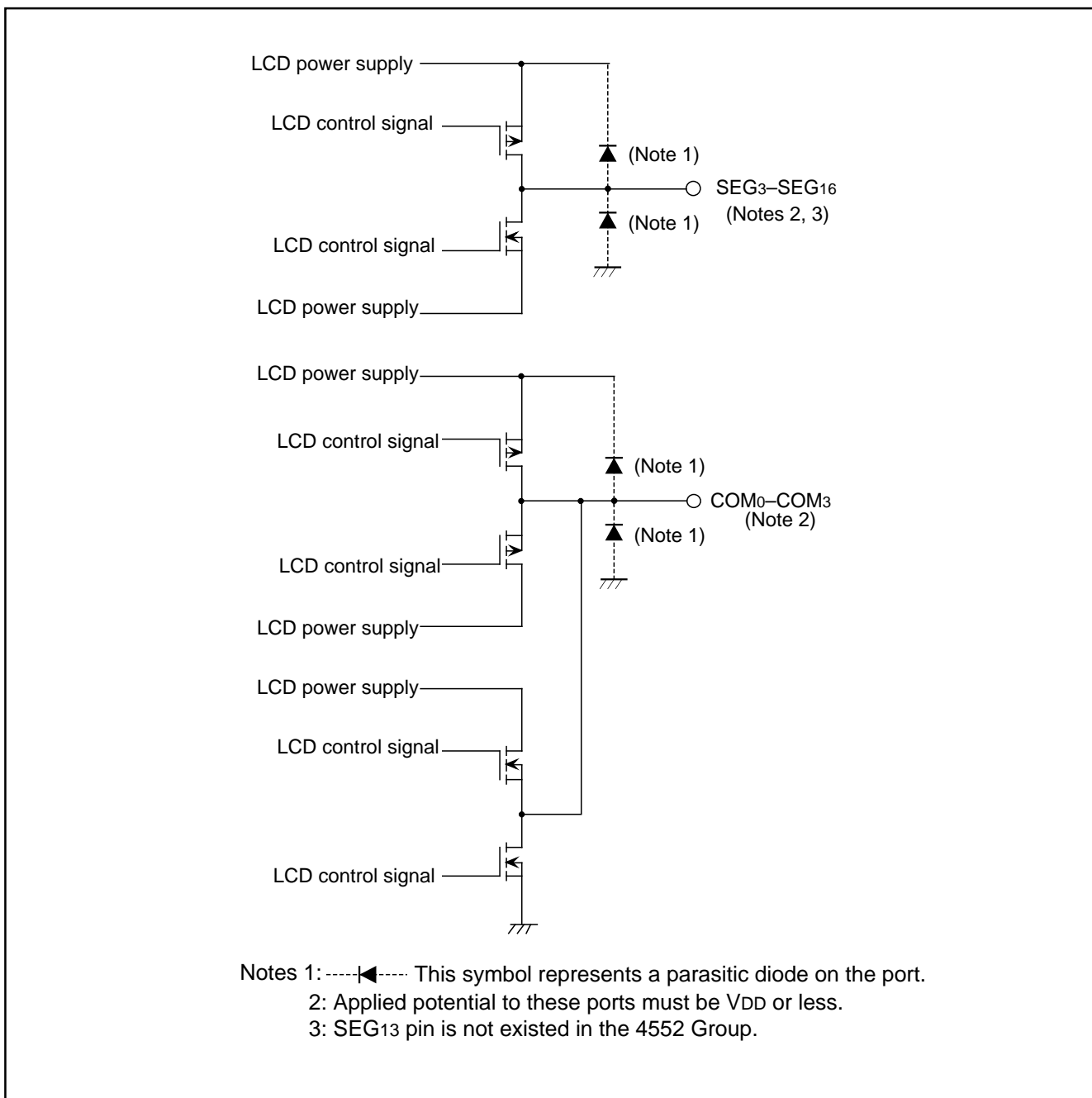


Port block diagram (4)

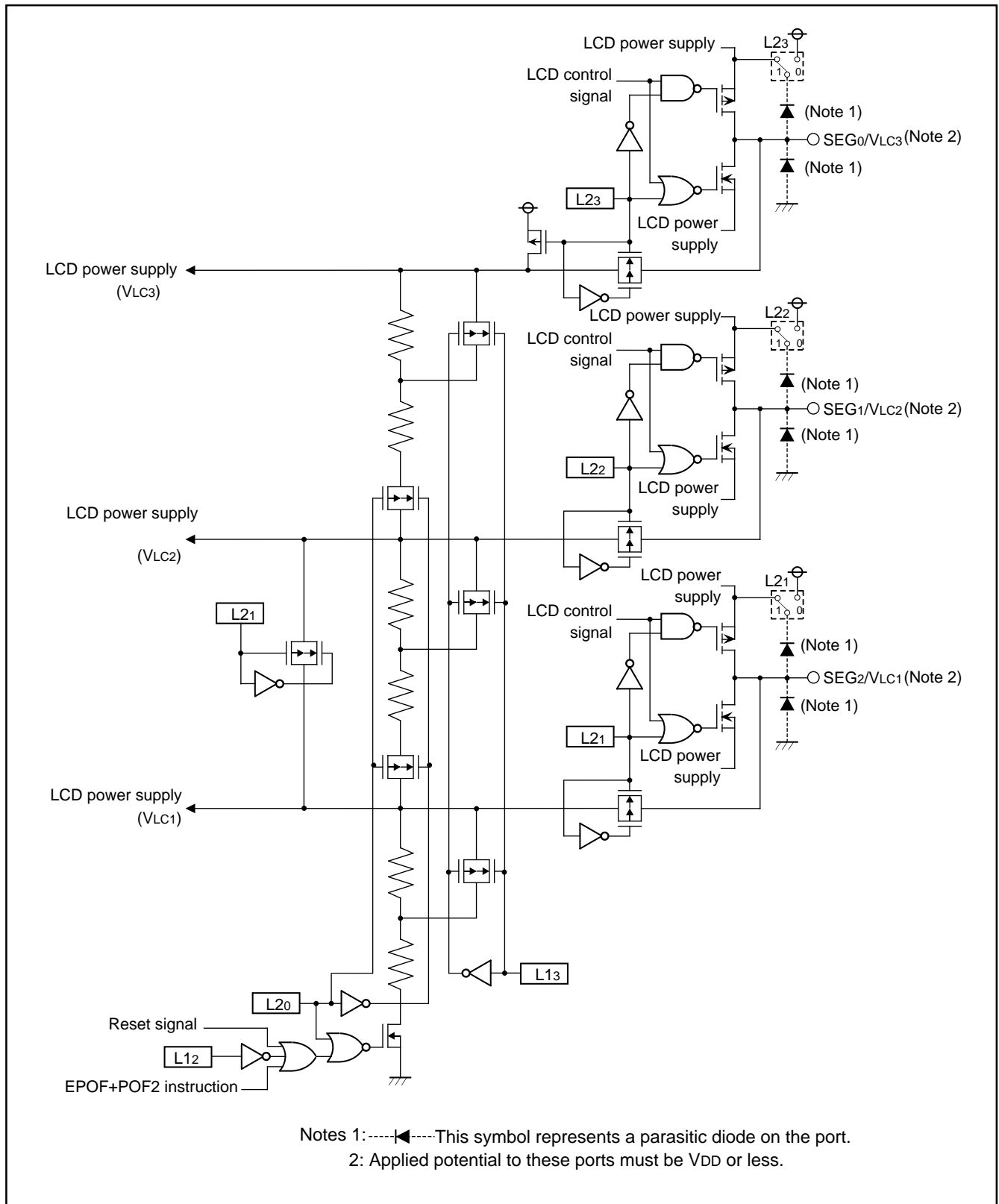


- Notes 1:  This symbol represents a parasitic diode on the port.
 2: Applied potential to these ports must be VDD or less.
 3: j represents bits 0 and 1.
 4: k represents bits 2 and 3.

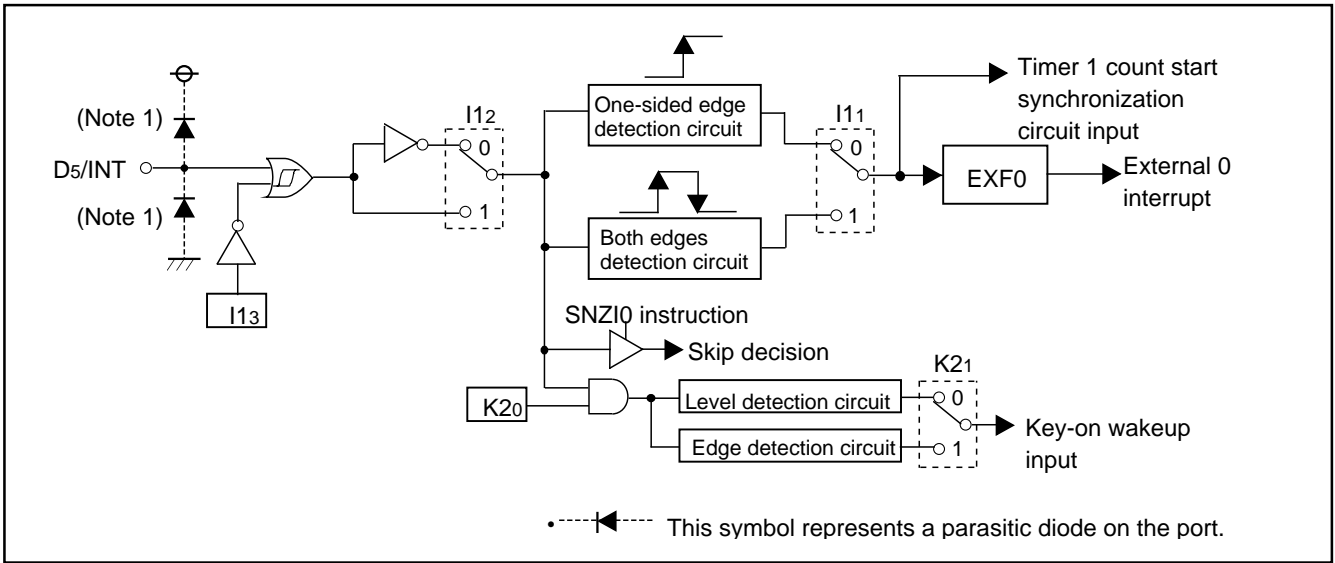
Port block diagram (5)



Port block diagram (6)



Port block diagram (7)



Block diagram of external interrupt

**FUNCTION BLOCK OPERATIONS
CPU**

(1) Arithmetic logic unit (ALU)

The arithmetic logic unit ALU performs 4-bit arithmetic such as 4-bit data addition, comparison, AND operation, OR operation, and bit manipulation.

(2) Register A and carry flag

Register A is a 4-bit register used for arithmetic, transfer, exchange, and I/O operation.

Carry flag CY is a 1-bit flag that is set to "1" when there is a carry with the AMC instruction (Figure 1).

It is unchanged with both A n instruction and AM instruction. The value of A0 is stored in carry flag CY with the RAR instruction (Figure 2).

Carry flag CY can be set to "1" with the SC instruction and cleared to "0" with the RC instruction.

(3) Registers B and E

Register B is a 4-bit register used for temporary storage of 4-bit data, and for 8-bit data transfer together with register A.

Register E is an 8-bit register. It can be used for 8-bit data transfer with register B used as the high-order 4 bits and register A as the low-order 4 bits (Figure 3).

Register E is undefined after system is released from reset and returned from the power down mode. Accordingly, set the initial value.

(4) Register D

Register D is a 3-bit register.

It is used to store a 7-bit ROM address together with register A and is used as a pointer within the specified page when the TABP p, BLA p, or BMLA p instruction is executed (Figure 4).

Also, when the TABP p instruction is executed at UPTF flag = "1", the high-order 2 bits of ROM reference data is stored to the low-order 2 bits of register D, the high-order 1 bit of register D is "0". When the TABP p instruction is executed at UPTF flag = "0", the contents of register D remains unchanged. The UPTF flag is set to "1" with the SUPT instruction and cleared to "0" with the RUPT instruction. The initial value of UPTF flag is "0".

Register D is undefined after system is released from reset and returned from the power down mode. Accordingly, set the initial value.

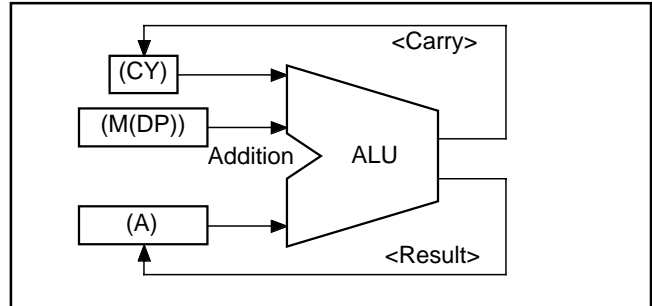


Fig. 1 AMC instruction execution example

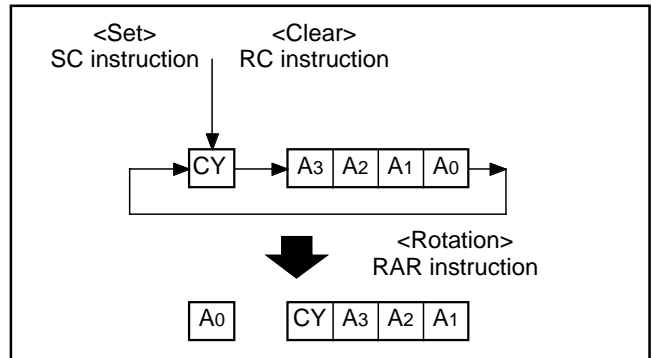


Fig. 2 RAR instruction execution example

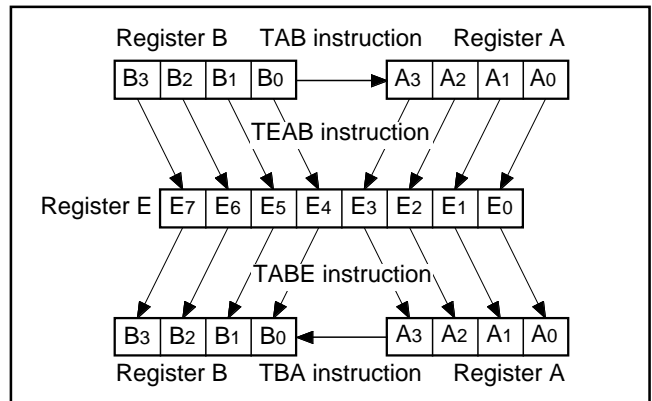


Fig. 3 Registers A, B and register E

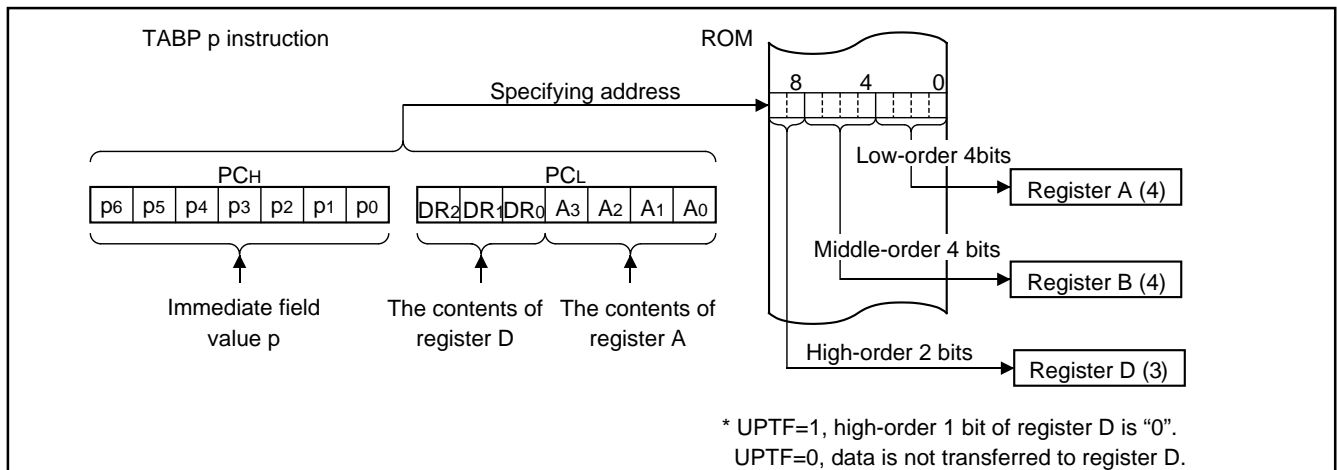


Fig. 4 TABP p instruction execution example

(5) Stack registers (SKs) and stack pointer (SP)

Stack registers (SKs) are used to temporarily store the contents of program counter (PC) just before branching until returning to the original routine when;

- branching to an interrupt service routine (referred to as an interrupt service routine),
- performing a subroutine call, or
- executing the table reference instruction (TABP p).

Stack registers (SKs) are eight identical registers, so that subroutines can be nested up to 8 levels. However, one of stack registers is used respectively when using an interrupt service routine and when executing a table reference instruction. Accordingly, be careful not to over the stack when performing these operations together. The contents of registers SKs are destroyed when 8 levels are exceeded.

The register SK nesting level is pointed automatically by 3-bit stack pointer (SP). The contents of the stack pointer (SP) can be transferred to register A with the TASP instruction.

Figure 5 shows the stack registers (SKs) structure.

Figure 6 shows the example of operation at subroutine call.

(6) Interrupt stack register (SDP)

Interrupt stack register (SDP) is a 1-stage register. When an interrupt occurs, this register (SDP) is used to temporarily store the contents of data pointer, carry flag, skip flag, register A, and register B just before an interrupt until returning to the original routine.

Unlike the stack registers (SKs), this register (SDP) is not used when executing the subroutine call instruction and the table reference instruction.

(7) Skip flag

Skip flag controls skip decision for the conditional skip instructions and continuous described skip instructions. When an interrupt occurs, the contents of skip flag is stored automatically in the interrupt stack register (SDP) and the skip condition is retained.

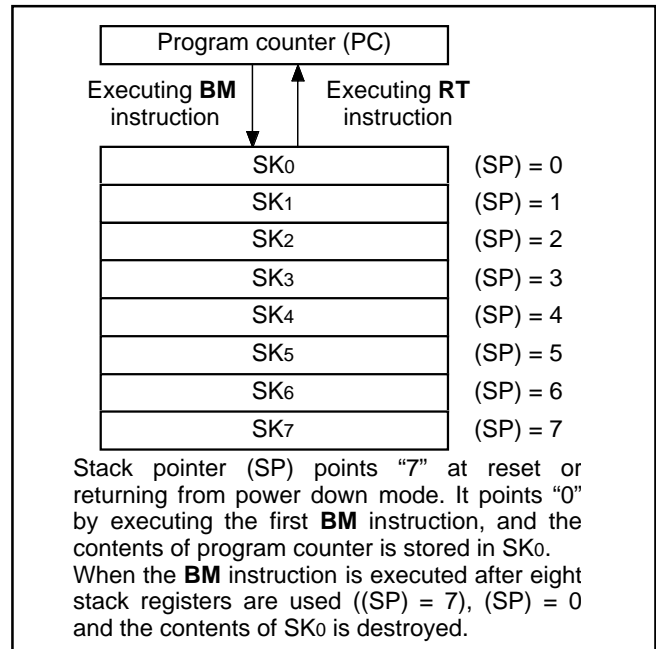


Fig. 5 Stack registers (SKs) structure

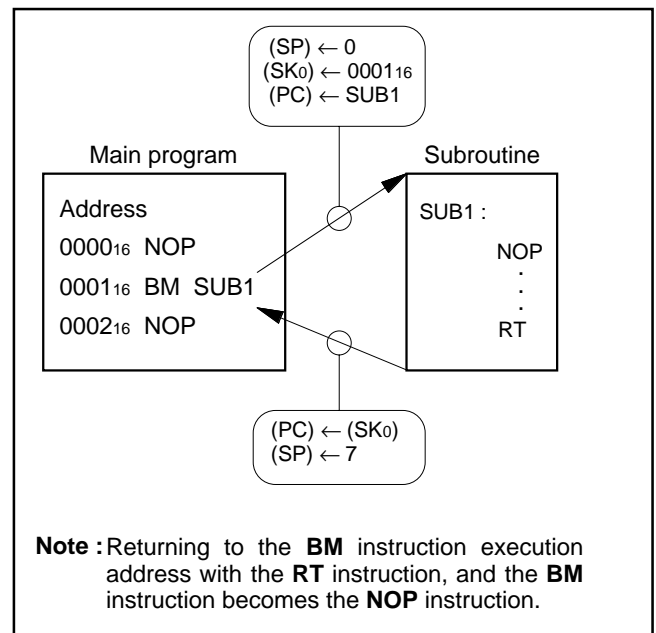


Fig. 6 Example of operation at subroutine call

(8) Program counter (PC)

Program counter (PC) is used to specify a ROM address (page and address). It determines a sequence in which instructions stored in ROM are read. It is a binary counter that increments the number of instruction bytes each time an instruction is executed. However, the value changes to a specified address when branch instructions, subroutine call instructions, return instructions, or the table reference instruction (TABP p) is executed.

Program counter consists of PCH (most significant bit to bit 7) which specifies to a ROM page and PCL (bits 6 to 0) which specifies an address within a page. After it reaches the last address (address 127) of a page, it specifies address 0 of the next page (Figure 7).

Make sure that the PCH does not specify after the last page of the built-in ROM.

(9) Data pointer (DP)

Data pointer (DP) is used to specify a RAM address and consists of registers Z, X, and Y. Register Z specifies a RAM file group, register X specifies a file, and register Y specifies a RAM digit (Figure 8).

Register Y is also used to specify the port D bit position. When using port D, set the port D bit position to register Y certainly and execute the SD, RD, or SZD instruction (Figure 9).

• Note

Register Z of data pointer is undefined after system is released from reset.

Also, registers Z, X and Y are undefined in the power down mode. After system is returned from the power down mode, set these registers.

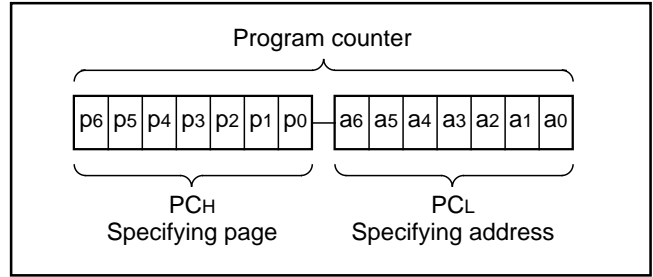


Fig. 7 Program counter (PC) structure

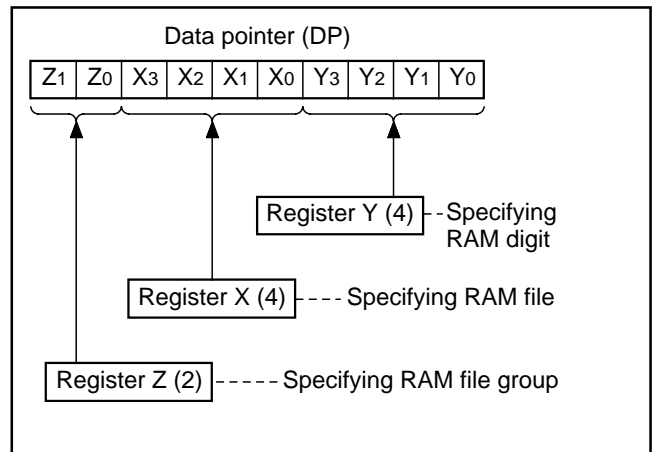


Fig. 8 Data pointer (DP) structure

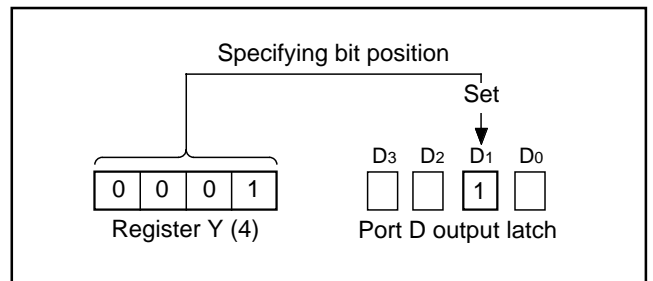


Fig. 9 SD instruction execution example

PROGRAM MEMORY (ROM)

The program memory is a mask ROM. 1 word of ROM is composed of 10 bits. ROM is separated every 128 words by the unit of page (addresses 0 to 127). Table 1 shows the ROM size and pages. Figure 10 shows the ROM map of M34552ED.

Table 1 ROM size and pages

| Part number | ROM (PROM) size (X 10 bits) | Pages |
|-------------|--------------------------------|--------------|
| M34552M4 | 4096 words | 32 (0 to 31) |
| M34552M4H | | |
| M34552M8 | 8192 words | 64 (0 to 63) |
| M34552M8H | | |
| M34552G8 | | |
| M34552G8H | | |
| M34552G8H | | |

A part of page 1 (addresses 0080₁₆ to 00FF₁₆) is reserved for interrupt addresses (Figure 11). When an interrupt occurs, the address (interrupt address) corresponding to each interrupt is set in the program counter, and the instruction at the interrupt address is executed. When using an interrupt service routine, write the instruction generating the branch to that routine at an interrupt address.

Page 2 (addresses 0100₁₆ to 017F₁₆) is the special page for subroutine calls. Subroutines written in this page can be called from any page with the 1-word instruction (BM). Subroutines extending from page 2 to another page can also be called with the BM instruction when it starts on page 2.

ROM pattern (bits 7 to 0) of all addresses can be used as data areas with the TABP p instruction.

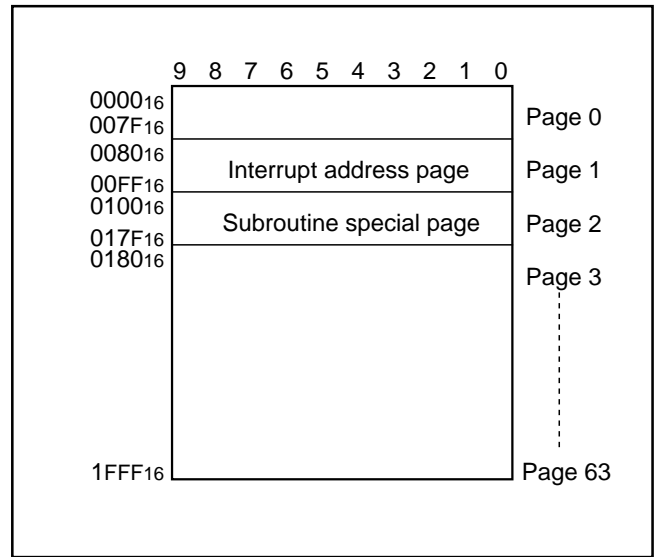


Fig. 10 ROM map of M34552M8/M8H/G8/G8H

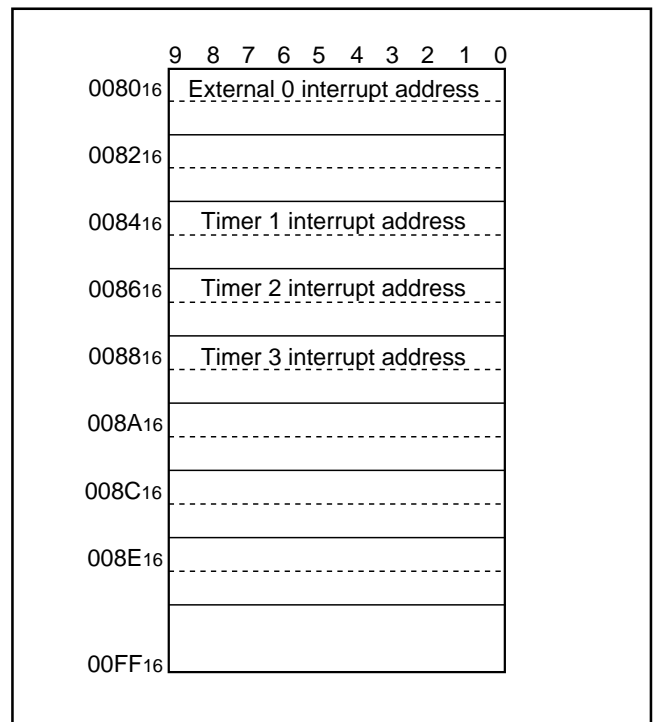


Fig. 11 Page 1 (addresses 0080₁₆ to 00FF₁₆) structure

INTERRUPT FUNCTION

The interrupt type is a vectored interrupt branching to an individual address (interrupt address) according to each interrupt source. An interrupt occurs when the following 3 conditions are satisfied.

- An interrupt activated condition is satisfied (request flag = "1")
- Interrupt enable bit is enabled ("1")
- Interrupt enable flag is enabled (INTE = "1")

Table 3 shows interrupt sources. (Refer to each interrupt request flag for details of activated conditions.)

(1) Interrupt enable flag (INTE)

The interrupt enable flag (INTE) controls whether the every interrupt enable/disable. Interrupts are enabled when INTE flag is set to "1" with the EI instruction and disabled when INTE flag is cleared to "0" with the DI instruction. When any interrupt occurs, the INTE flag is automatically cleared to "0," so that other interrupts are disabled until the EI instruction is executed.

(2) Interrupt enable bit

Use an interrupt enable bit of interrupt control registers V1 and V2 to select the corresponding interrupt or skip instruction.

Table 4 shows the interrupt request flag, interrupt enable bit and skip instruction.

Table 5 shows the interrupt enable bit function.

(3) Interrupt request flag

When the activated condition for each interrupt is satisfied, the corresponding interrupt request flag is set to "1." Each interrupt request flag is cleared to "0" when either;

- an interrupt occurs, or
- the next instruction is skipped with a skip instruction.

Each interrupt request flag is set when the activated condition is satisfied even if the interrupt is disabled by the INTE flag or its interrupt enable bit. Once set, the interrupt request flag retains set until a clear condition is satisfied.

Accordingly, an interrupt occurs when the interrupt disable state is released while the interrupt request flag is set.

If more than one interrupt request flag is set when the interrupt disable state is released, the interrupt priority level is as follows shown in Table 3.

Table 3 Interrupt sources

| Priority level | Interrupt name | Activated condition | Interrupt address |
|----------------|----------------------|-------------------------|---------------------|
| 1 | External 0 interrupt | Level change of INT pin | Address 0 in page 1 |
| 2 | Timer 1 interrupt | Timer 1 underflow | Address 4 in page 1 |
| 3 | Timer 2 interrupt | Timer 2 underflow | Address 6 in page 1 |
| 4 | Timer 3 interrupt | Timer 3 underflow | Address 8 in page 1 |

Table 4 Interrupt request flag, interrupt enable bit and skip instruction

| Interrupt name | Request flag | Skip instruction | Enable bit |
|----------------------|--------------|------------------|------------|
| External 0 interrupt | EXF0 | SNZ0 | V10 |
| Timer 1 interrupt | T1F | SNZT1 | V12 |
| Timer 2 interrupt | T2F | SNZT2 | V13 |
| Timer 3 interrupt | T3F | SNZT3 | V20 |

Table 5 Interrupt enable bit function

| Interrupt enable bit | Occurrence of interrupt | Skip instruction |
|----------------------|-------------------------|------------------|
| 1 | Enabled | Invalid |
| 0 | Disabled | Valid |

(4) Internal state during an interrupt

The internal state of the microcomputer during an interrupt is as follows (Figure 14).

- Program counter (PC)
An interrupt address is set in program counter. The address to be executed when returning to the main routine is automatically stored in the stack register (SK).
- Interrupt enable flag (INTE)
INTE flag is cleared to "0" so that interrupts are disabled.
- Interrupt request flag
Only the request flag for the current interrupt source is cleared to "0."
- Data pointer, carry flag, skip flag, registers A and B
The contents of these registers and flags are stored automatically in the interrupt stack register (SDP).

(5) Interrupt processing

When an interrupt occurs, a program at an interrupt address is executed after branching a data store sequence to stack register. Write the branch instruction to an interrupt service routine at an interrupt address.

Use the RTI instruction to return from an interrupt service routine. Interrupt enabled by executing the EI instruction is performed after executing 1 instruction (just after the next instruction is executed). Accordingly, when the EI instruction is executed just before the RTI instruction, interrupts are enabled after returning the main routine. (Refer to Figure 13)

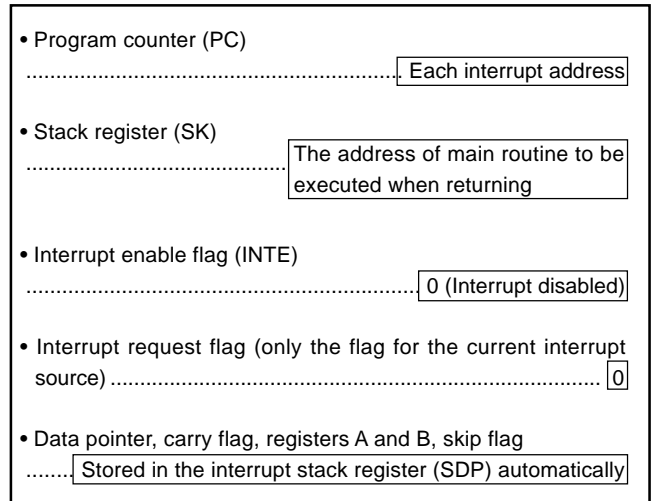


Fig. 14 Internal state when interrupt occurs

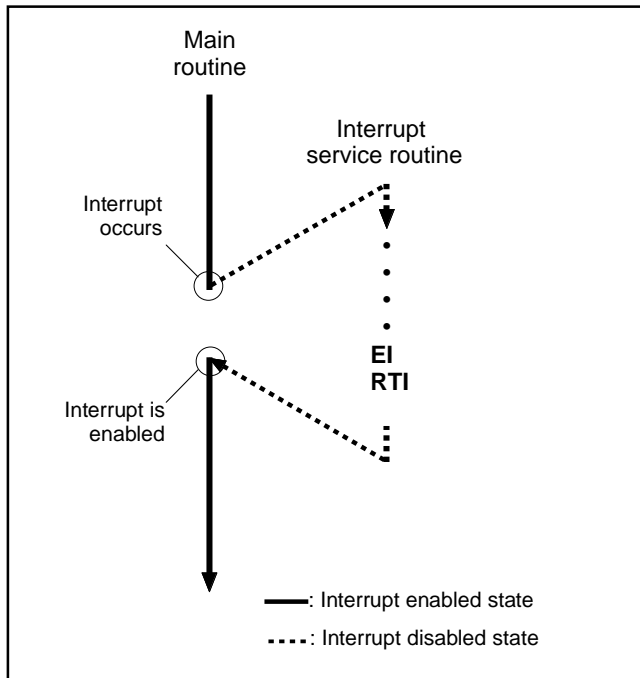


Fig. 13 Program example of interrupt processing

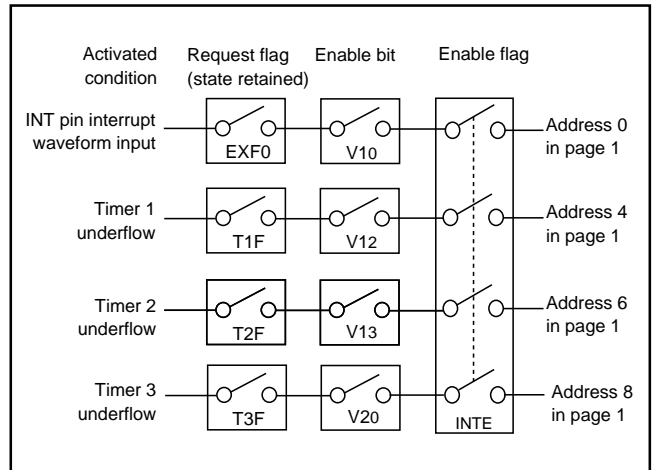


Fig. 15 Interrupt system diagram

(6) Interrupt control registers

- Interrupt control register V1

Interrupt enable bits of external 0, timer 1 and timer 2 are assigned to register V1. Set the contents of this register through register A with the TV1A instruction. The TAV1 instruction can be used to transfer the contents of register V1 to register A.

- Interrupt control register V2

The timer 3 interrupt enable bit is assigned to register V2. Set the contents of this register through register A with the TV2A instruction. The TAV2 instruction can be used to transfer the contents of register V2 to register A.

Table 6 Interrupt control registers

| Interrupt control register V1 | | at reset : 00002 | at power down : 00002 | R/W TAV1/TV1A |
|-------------------------------|---------------------------------|------------------|--|------------------|
| V13 | Timer 2 interrupt enable bit | 0 | Interrupt disabled (SNZT2 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZT2 instruction is invalid) | |
| V12 | Timer 1 interrupt enable bit | 0 | Interrupt disabled (SNZT1 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZT1 instruction is invalid) | |
| V11 | Not used | 0 | This bit has no function, but read/write is enabled. | |
| | | 1 | | |
| V10 | External 0 interrupt enable bit | 0 | Interrupt disabled (SNZ0 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZ0 instruction is invalid) | |

| Interrupt control register V2 | | at reset : 00002 | at power down : 00002 | R/W TAV2/TV2A |
|-------------------------------|------------------------------|------------------|--|------------------|
| V23 | Not used | 0 | This bit has no function, but read/write is enabled. | |
| | | 1 | | |
| V22 | Not used | 0 | This bit has no function, but read/write is enabled. | |
| | | 1 | | |
| V21 | Not used | 0 | This bit has no function, but read/write is enabled. | |
| | | 1 | | |
| V20 | Timer 3 interrupt enable bit | 0 | Interrupt disabled (SNZT3 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZT3 instruction is invalid) | |

Note: "R" represents read enabled, and "W" represents write enabled.

(7) Interrupt sequence

Interrupts only occur when the respective INTE flag, interrupt enable bits (V10, V12, V13, V20), and interrupt request flag are "1." The interrupt actually occurs 2 to 3 machine cycles after the cycle in which all three conditions are satisfied. The interrupt occurs after 3 machine cycles only when the three interrupt conditions are satisfied on execution of other than one-cycle instructions (Refer to Figure 16).

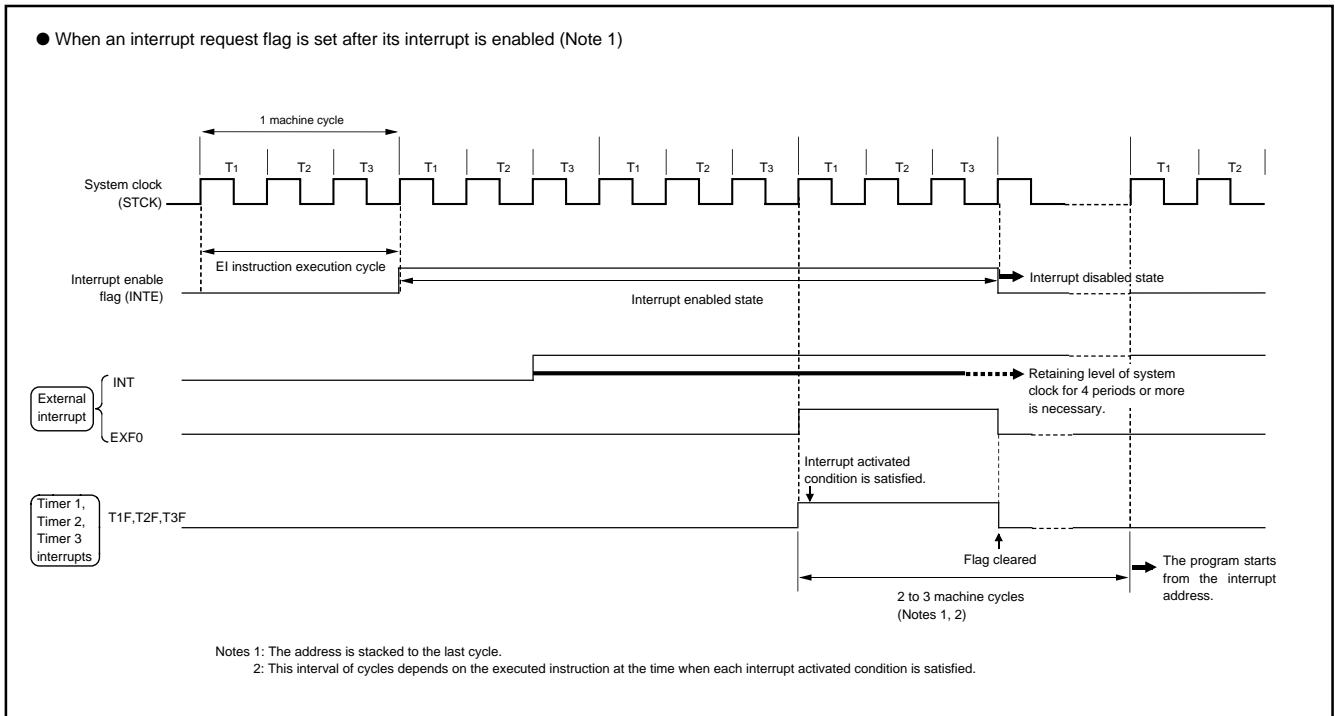


Fig. 16 Interrupt sequence

EXTERNAL INTERRUPTS

The 4552 Group has the external 0 interrupt.
 An external interrupt request occurs when a valid waveform is input to an interrupt input pin (edge detection).
 The external interrupt can be controlled with the interrupt control register I1.

Table 7 External interrupt activated conditions

| Name | Input pin | Activated condition | Valid waveform selection bit |
|----------------------|-----------|--|------------------------------|
| External 0 interrupt | D5/INT | When the next waveform is input to D5/INT pin <ul style="list-style-type: none"> • Falling waveform (“H”→“L”) • Rising waveform (“L”→“H”) • Both rising and falling waveforms | I11 I12 |

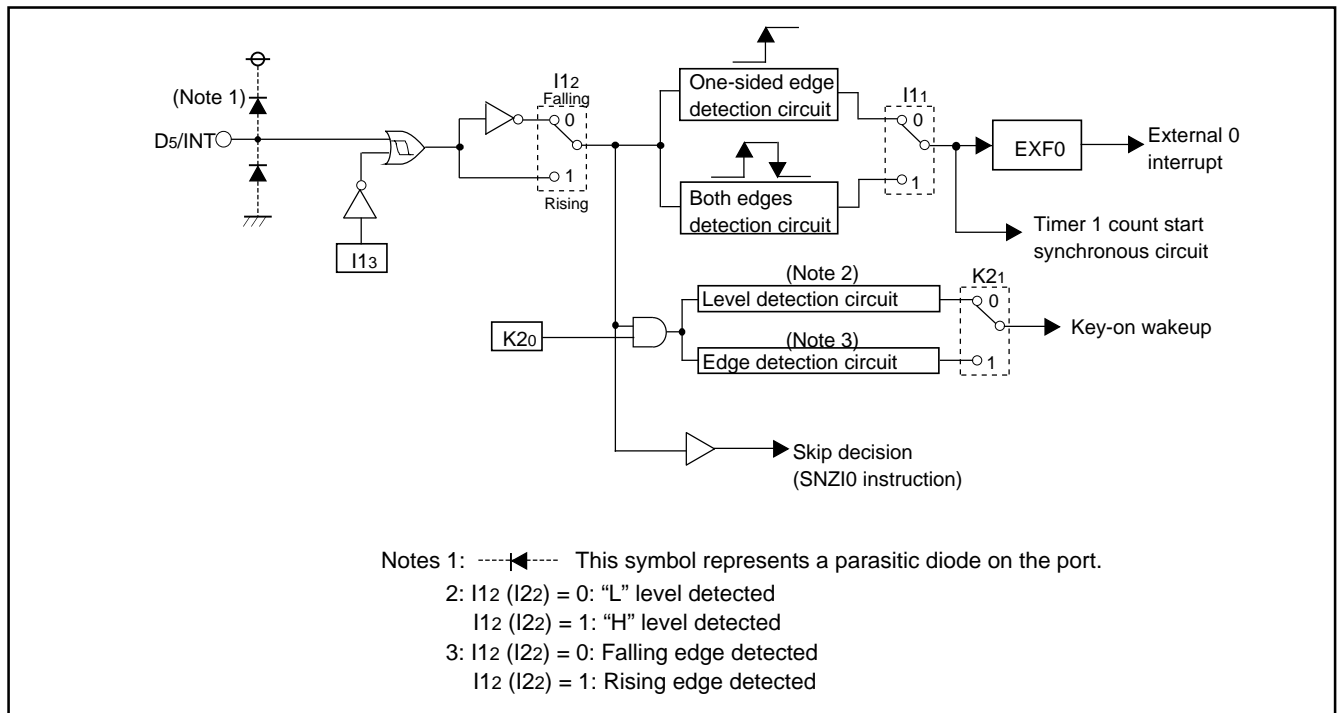


Fig. 17 External interrupt circuit structure

(1) External 0 interrupt request flag (EXF0)

External 0 interrupt request flag (EXF0) is set to "1" when a valid waveform is input to D5/INT pin.

The valid waveforms causing the interrupt must be retained at their level for 4 clock cycles or more of the system clock (Refer to Figure 16). The state of EXF0 flag can be examined with the skip instruction (SNZ0). Use the interrupt control register V1 to select the interrupt or the skip instruction. The EXF0 flag is cleared to "0" when an interrupt occurs or when the next instruction is skipped with the skip instruction.

- External 0 interrupt activated condition

External 0 interrupt activated condition is satisfied when a valid waveform is input to D5/INT pin.

The valid waveform can be selected from rising waveform, falling waveform or both rising and falling waveforms. An example of how to use the external 0 interrupt is as follows.

- ① Set the bit 3 of register I1 to "1" for the INT pin to be in the input enabled state.
- ② Select the valid waveform with the bits 1 and 2 of register I1.
- ③ Clear the EXF0 flag to "0" with the SNZ0 instruction.
- ④ Set the NOP instruction for the case when a skip is performed with the SNZ0 instruction.
- ⑤ Set both the external 0 interrupt enable bit (V10) and the INTE flag to "1."

The external 0 interrupt is now enabled. Now when a valid waveform is input to the D5/INT pin, the EXF0 flag is set to "1" and the external 0 interrupt occurs.

(2) External interrupt control registers

- Interrupt control register I1

Register I1 controls the valid waveform for the external 0 interrupt. Set the contents of this register through register A with the T11A instruction. The TAI1 instruction can be used to transfer the contents of register I1 to register A.

Table 8 External interrupt control register

| Interrupt control register I1 | | at reset : 0000 ₂ | at power down : state retained | R/W TAI1/TI1A |
|-------------------------------|--|------------------------------|---|------------------|
| I13 | INT pin input control bit (Note 2) | 0 | INT pin input disabled | |
| | | 1 | INT pin input enabled | |
| I12 | Interrupt valid waveform for INT pin/ return level selection bit (Note 2) | 0 | Falling waveform/"L" level ("L" level is recognized with the SNZIO instruction) | |
| | | 1 | Rising waveform/"H" level ("H" level is recognized with the SNZIO instruction) | |
| I11 | INT pin edge detection circuit control bit | 0 | One-sided edge detected | |
| | | 1 | Both edges detected | |
| I10 | INT pin Timer 1 count start synchronous circuit selection bit | 0 | Timer 1 count start synchronous circuit not selected | |
| | | 1 | Timer 1 count start synchronous circuit selected | |

Notes 1: "R" represents read enabled, and "W" represents write enabled.

2: When the contents of these bits (I12, I13) are changed, the external interrupt request flag (EXF0) may be set.

(3) Notes on External 0 interrupts

① Note [1] on bit 3 of register I1

When the input of the INT pin is controlled with the bit 3 of register I1 in software, be careful about the following notes.

- Depending on the input state of the D5/INT pin, the external 0 interrupt request flag (EXF0) may be set when the bit 3 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 18①) and then, change the bit 3 of register I1. In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 18②). Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 18③).

```

    ⋮
    LA    4      ; (XXX02)
    TV1A      ; The SNZ0 instruction is valid ..... ①
    LA    8      ; (1XXX2)
    T11A      ; Control of INT pin input is changed
    NOP      ..... ②
    SNZ0      ; The SNZ0 instruction is executed
              (EXF0 flag cleared)
    NOP      ..... ③
    ⋮
    X : these bits are not used here.
    
```

Fig. 18 External 0 interrupt program example-1

② Note [2] on bit 3 of register I1

When the bit 3 of register I1 is cleared to "0", the RAM back-up mode is selected and the input of INT pin is disabled, be careful about the following notes.

- When the key-on wakeup function of INT pin is not used (register K20 = "0"), clear bits 2 and 3 of register I1 before system enters to the power down mode. (refer to Figure 19①).

```

    ⋮
    LA    0      ; (00XX2)
    T11A      ; Input of INT disabled ..... ①
    DI
    EPOF
    POF2      ; power down mode
    ⋮
    X : these bits are not used here.
    
```

Fig. 19 External 0 interrupt program example-2

③ Note on bit 2 of register I1

When the interrupt valid waveform of the D5/INT pin is changed with the bit 2 of register I1 in software, be careful about the following notes.

- Depending on the input state of the D5/INT pin, the external 0 interrupt request flag (EXF0) may be set when the bit 2 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 20①) and then, change the bit 2 of register I1. In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 20②). Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 20③).

```

    ⋮
    LA    4      ; (XXX02)
    TV1A      ; The SNZ0 instruction is valid ..... ①
    LA    12     ; (11XXX2)
    T11A      ; Interrupt valid waveform is changed
    NOP      ..... ②
    SNZ0      ; The SNZ0 instruction is executed
              (EXF0 flag cleared)
    NOP      ..... ③
    ⋮
    X : these bits are not used here.
    
```

Fig. 20 External 0 interrupt program example-3

TIMERS

The 4552 Group has the following timers.

- Programmable timer

The programmable timer has a reload register and enables the frequency dividing ratio to be set. It is decremented from a setting value n . When it underflows (count to $n + 1$), a timer interrupt request flag is set to "1," new data is loaded from the reload register, and count continues (auto-reload function).

- Fixed dividing frequency timer

The fixed dividing frequency timer has the fixed frequency dividing ratio (n). An interrupt request flag is set to "1" after every n count of a count pulse.

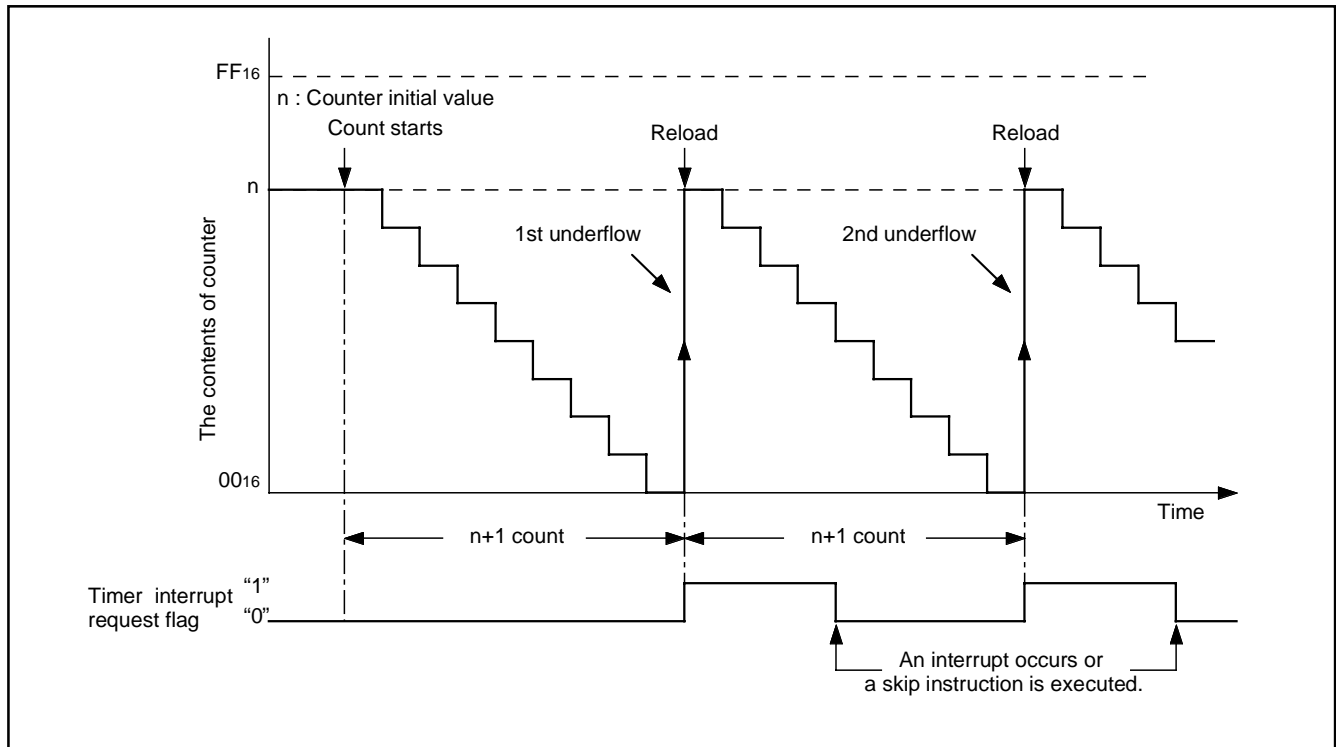


Fig. 21 Auto-reload function

The 4552 Group timer consists of the following circuits.

- Prescaler : 8-bit programmable timer
- Timer 1 : 8-bit programmable timer
- Timer 2 : 8-bit programmable timer
- Timer 3 : 16-bit fixed dividing frequency timer
- Timer LC : 4-bit programmable timer
- Watchdog timer : 16-bit fixed dividing frequency timer
(Timers 1, 2, and 3 have the interrupt function, respectively)

Prescaler and timers 1, 2, 3 and LC can be controlled with the timer control registers PA, W1 to W4. The watchdog timer is a free counter which is not controlled with the control register.

Each function is described below.

Table 9 Function related timers

| Circuit | Structure | Count source | Frequency dividing ratio | Use of output signal | Control register |
|----------------|--|--|---------------------------------|--|------------------|
| Prescaler | 8-bit programmable binary down counter | • Instruction clock (INSTCK) | 1 to 256 | • Timer 1, 2, and 3 count sources | PA |
| Timer 1 | 8-bit programmable binary down counter (link to INT input) | • PWM output (PWMOUT) • Prescaler output (ORCLK) • Timer 3 underflow (T3UDF) • CNTR input | 1 to 256 | • CNTR output control • Timer 1 interrupt | W1 |
| Timer 2 | 8-bit programmable binary down counter (PWM output function) | • XIN input • Prescaler output (ORCLK) divided by 2 | 1 to 256 | • Timer 1 count source • CNTR output • Timer 2 interrupt | W2 |
| Timer 3 | 16-bit fixed dividing frequency | • XCIN input • ORCLK | 8192 16384 32768 65536 | • Timer 1 count source • Timer 3 interrupt • Timer LC count source | W3 |
| Timer LC | 4-bit programmable binary down counter | • Bit 4 of timer 3 • System clock (STCK) | 1 to 16 | • LCD clock | W4 |
| Watchdog timer | 16-bit fixed dividing frequency | • Instruction clock (INSTCK) | 65534 | • System reset (count twice) • WDF flag decision | |

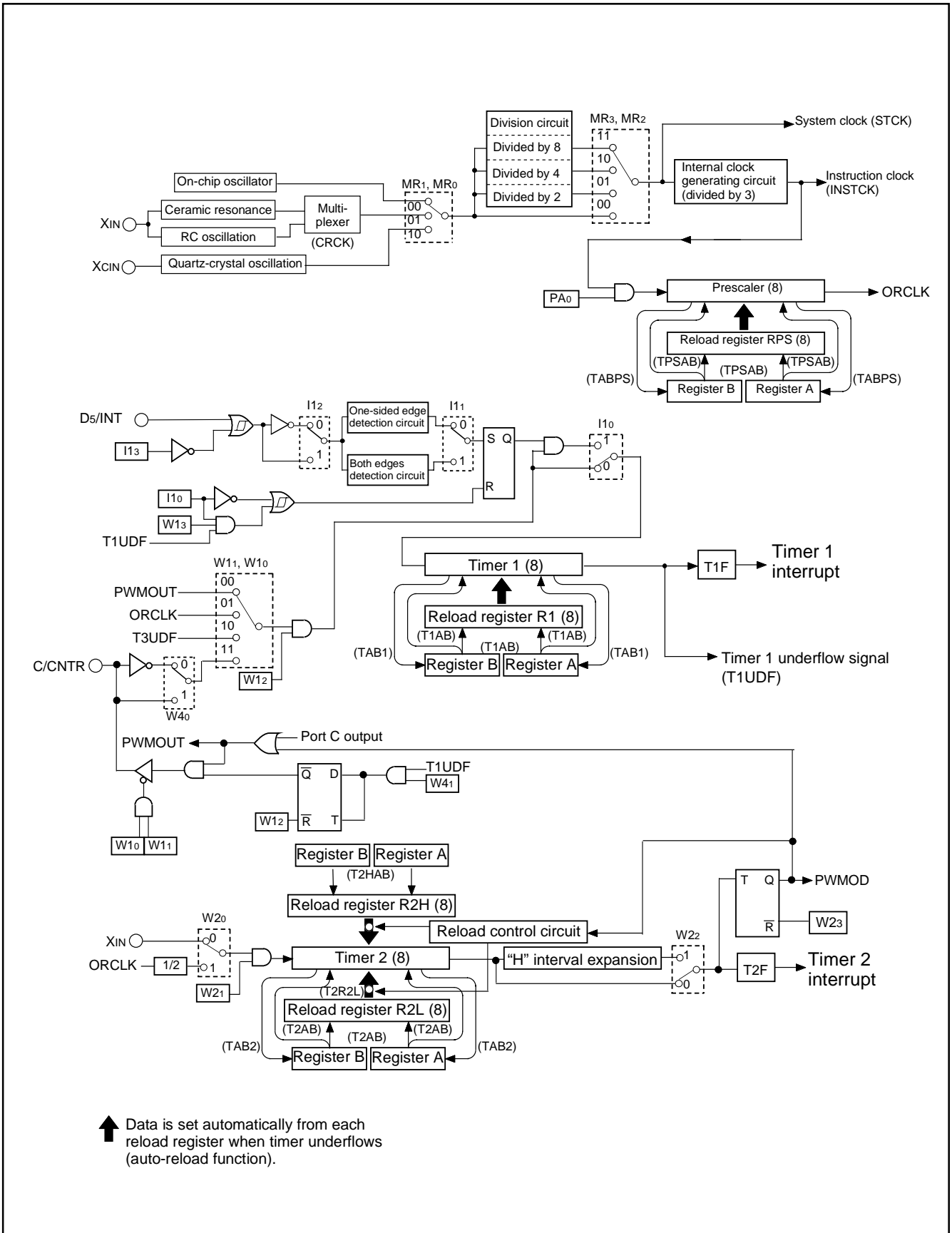


Fig. 22 Timer structure (1)

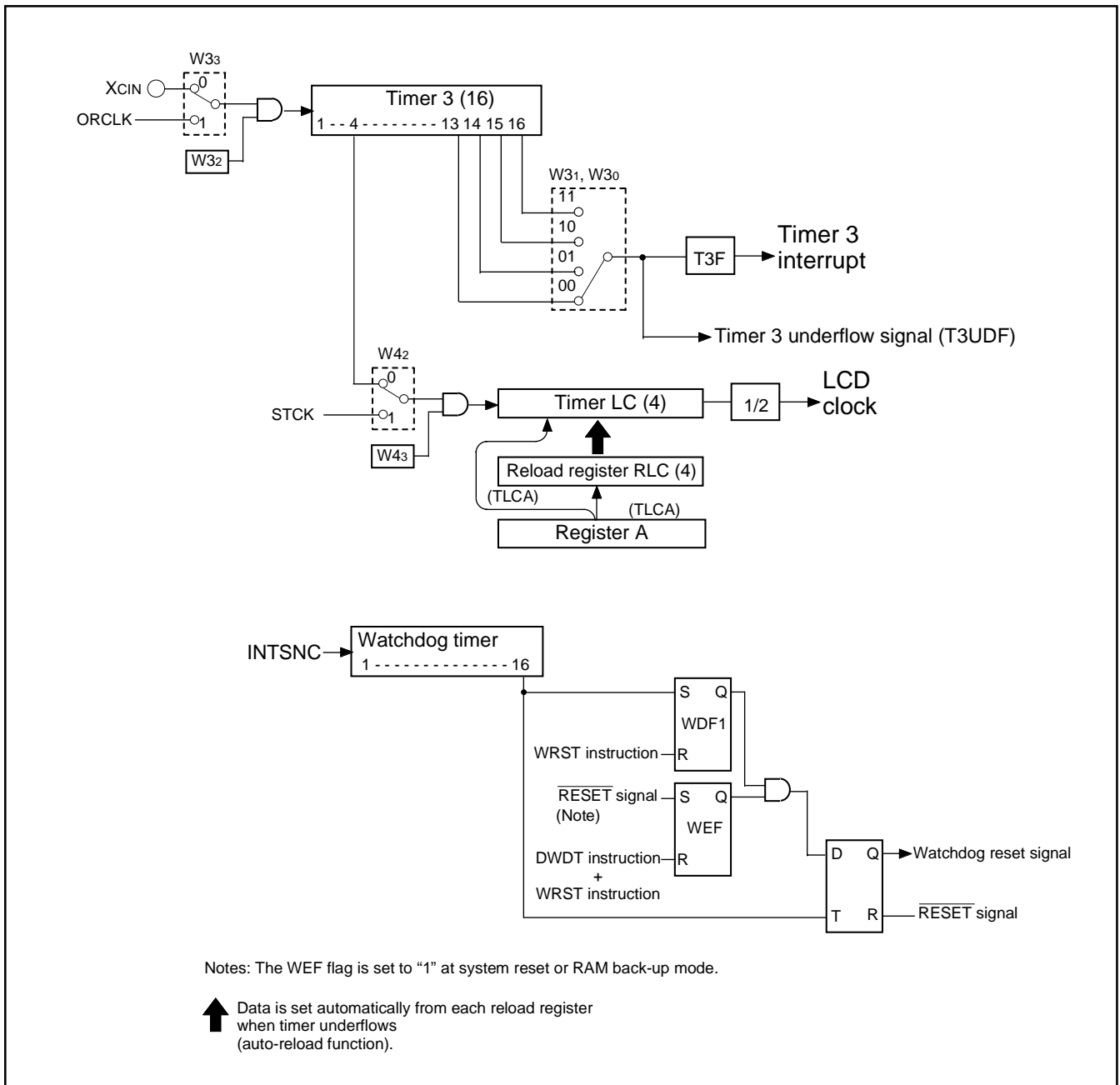


Fig. 23 Timer structure (2)

Table 10 Timer related registers

| Timer control register PA | | at reset : 02 | | at power down : 02 | | W TPAA | | |
|---------------------------|-----------------------|---------------|-----------------------|--------------------|--|-----------|--|--|
| PA0 | Prescaler control bit | 0 | Stop (state retained) | | | | | |
| | | 1 | Operating | | | | | |

| Timer control register W1 | | at reset : 00002 | | at power down : state retained | | R/W TAW1/TW1A | | | |
|---------------------------|--|------------------|--|----------------------------------|--|------------------|--|--|--|
| W13 | Timer 1 count auto-stop circuit selection bit (Note 2) | 0 | Timer 1 count auto-stop circuit not selected | | | | | | |
| | | 1 | Timer 1 count auto-stop circuit selected | | | | | | |
| W12 | Timer 1 control bit | 0 | Stop (state retained) | | | | | | |
| | | 1 | Operating | | | | | | |
| W11 | Timer 1 count source selection bits (Note 3) | W11 | W10 | Count source | | | | | |
| | | 0 | 0 | PWM signal (PWMOUT) | | | | | |
| W10 | Timer 1 count source selection bits (Note 3) | 0 | 1 | Prescaler output (ORCLK) | | | | | |
| | | 1 | 0 | Timer 3 underflow signal (T3UDF) | | | | | |
| | | 1 | 1 | CNTR input | | | | | |

| Timer control register W2 | | at reset : 00002 | | at power down : 00002 | | R/W TAW2/TW2A | | |
|---------------------------|--|------------------|--|-----------------------|--|------------------|--|--|
| W23 | CNTR pin output control bit | 0 | CNTR pin output invalid | | | | | |
| | | 1 | CNTR pin output valid | | | | | |
| W22 | PWM signal interrupt valid waveform/return level selection bit | 0 | PWM signal "H" interval expansion function invalid | | | | | |
| | | 1 | PWM signal "H" interval expansion function valid | | | | | |
| W21 | Timer 2 control bit | 0 | Stop (state retained) | | | | | |
| | | 1 | Operating | | | | | |
| W20 | Timer 2 count source selection bit | 0 | XIN input | | | | | |
| | | 1 | Prescaler output (ORCLK)/2 signal output | | | | | |

| Timer control register W3 | | at reset : 00002 | | at power down : state retained | | R/W TAW3/TW3A | | | |
|---------------------------|---|------------------|--------------------------|-------------------------------------|--|------------------|--|--|--|
| W33 | Timer 3 count auto-stop circuit selection bit | 0 | XCIN input | | | | | | |
| | | 1 | Prescaler output (ORCLK) | | | | | | |
| W32 | Timer 3 control bit | 0 | Stop (Initial state) | | | | | | |
| | | 1 | Operating | | | | | | |
| W31 | Timer 3 count value selection bits | W31 | W30 | Count value | | | | | |
| | | 0 | 0 | Underflow occurs every 8192 counts | | | | | |
| W30 | Timer 3 count value selection bits | 0 | 1 | Underflow occurs every 16384 counts | | | | | |
| | | 1 | 0 | Underflow occurs every 32768 counts | | | | | |
| | | 1 | 1 | Underflow occurs every 65536 counts | | | | | |

| Timer control register W4 | | at reset : 00002 | | at power down : state retained | | R/W TAW4/TW4A | | |
|---------------------------|--|------------------|---|--------------------------------|--|------------------|--|--|
| W43 | Timer LC control bit | 0 | Stop (state retained) | | | | | |
| | | 1 | Operating | | | | | |
| W42 | Timer LC count source selection bit | 0 | Bit 4 (T34) of timer 3 | | | | | |
| | | 1 | System clock (STCK) | | | | | |
| W41 | CNTR output auto-control circuit selection bit | 0 | CNTR output auto-control circuit not selected | | | | | |
| | | 1 | CNTR output auto-control circuit selected | | | | | |
| W40 | CNTR pin input count edge selection bit | 0 | Falling edge | | | | | |
| | | 1 | Rising edge | | | | | |

Notes 1: "R" represents read enabled, and "W" represents write enabled.

2: This function is valid only when the timer 1 count start synchronous circuit is selected (I10="1").

3: Port C output is invalid when CNTR input is selected for the timer 1 count source.

(1) Timer control registers

- Timer control register PA
Register PA controls the count operation of prescaler. Set the contents of this register through register A with the TPAA instruction.
- Timer control register W1
Register W1 controls the selection of timer 1 count auto-stop circuit, and the count operation and count source of timer 1. Set the contents of this register through register A with the TW1A instruction. The TAW1 instruction can be used to transfer the contents of register W1 to register A.
- Timer control register W2
Register W2 controls the CNTR output, the expansion of “H” interval of PWM output, and the count operation and count source of timer 2. Set the contents of this register through register A with the TW2A instruction. The TAW2 instruction can be used to transfer the contents of register W2 to register A.
- Timer control register W3
Register W3 controls the count operation and count source of timer 3. Set the contents of this register through register A with the TW3A instruction. The TAW3 instruction can be used to transfer the contents of register W3 to register A.
- Timer control register W4
Register W4 controls the operation and count source of timer LC, the selection of CNTR output auto-control circuit and the count edge of CNTR input. Set the contents of this register through register A with the TW4A instruction. The TAW4 instruction can be used to transfer the contents of register W4 to register A..

(2) Prescaler (interrupt function)

Prescaler is an 8-bit binary down counter with the prescaler reload register RPS. Data can be set simultaneously in prescaler and the reload register RPS with the TPSAB instruction. Data can be read from reload register RPS with the TABPS instruction.

Stop counting and then execute the TPSAB or TABPS instruction to read or set prescaler data.

Prescaler starts counting after the following process;

- ① set data in prescaler, and
- ② set the bit 0 of register PA to “1.”

When a value set in reload register RPS is n , prescaler divides the count source signal by $n + 1$ ($n = 0$ to 255).

Count source for prescaler is the instruction clock (INSTCK).

Once count is started, when prescaler underflows (the next count pulse is input after the contents of prescaler becomes “0”), new data is loaded from reload register RPS, and count continues (auto-reload function).

The output signal (ORCLK) of prescaler can be used for timer 1, 2, and 3 count sources.

(3) Timer 1 (interrupt function)

Timer 1 is an 8-bit binary down counter with the timer 1 reload register (R1). Data can be set simultaneously in timer 1 and the reload register (R1) with the T1AB instruction. Data can be written to reload register (R1) with the TR1AB instruction. Data can be read from timer 1 with the TAB1 instruction.

Stop counting and then execute the T1AB or TAB1 instruction to read or set timer 1 data.

When executing the TR1AB instruction to set data to reload register R1 while timer 1 is operating, avoid a timing when timer 1 underflows.

Timer 1 starts counting after the following process;

- ① set data in timer 1
- ② set count source by bits 0 and 1 of register W1, and
- ③ set the bit 2 of register W1 to “1.”

When a value set in reload register R1 is n , timer 1 divides the count source signal by $n + 1$ ($n = 0$ to 255).

Once count is started, when timer 1 underflows (the next count pulse is input after the contents of timer 1 becomes “0”), the timer 1 interrupt request flag (T1F) is set to “1,” new data is loaded from reload register R1, and count continues (auto-reload function).

INT pin input can be used as the start trigger for timer 1 count operation by setting the bit 0 of register I1 to “1.”

Also, in this time, the auto-stop function by timer 1 underflow can be performed by setting the bit 3 of register W1 to “1.”

(4) Timer 2 (interrupt function)

Timer 2 is an 8-bit binary down counter with two timer 2 reload registers (R2L, R2H). Data can be set simultaneously in timer 2 and the reload register R2L with the T2AB instruction. Data can be set in the reload register R2H with the T2HAB instruction. The contents of reload register R2L set with the T2AB instruction can be set to timer 2 again with the T2R2L instruction. Data can be read from timer 2 with the TAB2 instruction.

Stop counting and then execute the T2AB or TAB2 instruction to read or set timer 2 data.

When executing the T2HAB instruction to set data to reload register R2H while timer 2 is operating, avoid a timing when timer 2 underflows.

Timer 2 starts counting after the following process;

- ① set data in timer 2
- ② set count source by bit 0 of register W2, and
- ③ set the bit 1 of register W2 to "1."

When a value set in reload register R2L is n , timer 2 divides the count source signal by $n + 1$ ($n = 0$ to 255).

Once count is started, when timer 2 underflows (the next count pulse is input after the contents of timer 2 becomes "0"), the timer 2 interrupt request flag (T2F) is set to "1," new data is loaded from reload register R2L, and count continues (auto-reload function).

When bit 3 of register W2 is set to "1", timer 2 reloads data from reload register R2L and R2H alternately each underflow.

Timer 2 generates the PWM signal (PWMOOUT) of the "L" interval set as reload register R2L, and the "H" interval set as reload register R2H. The PWM signal (PWMOOUT) is output from CNTR pin.

When bit 2 of register W2 is set to "1" at this time, the interval (PWM signal "H" interval) set to reload register R2H for the counter of timer 2 is extended for a half period of count source.

In this case, when a value set in reload register R2H is n , timer 2 divides the count source signal by $n + 1.5$ ($n = 1$ to 255).

When this function is used, set "1" or more to reload register R2H.

When bit 1 of register W4 is set to "1", the PWM signal output to CNTR pin is switched to valid/invalid each timer 1 underflow. However, when timer 1 is stopped (bit 2 of register W1 is cleared to "0"), this function is canceled.

Even when bit 1 of a register W2 is cleared to "0" in the "H" interval of PWM signal, timer 2 does not stop until it next timer 2 underflow. When clearing bit 1 of register W2 to "0" to stop timer 2, avoid a timing when timer 2 underflows.

(5) Timer 3 (interrupt function)

Timer 3 is a 16-bit binary down counter.

Timer 3 starts counting after the following process;

- ① set count value by bits 0 and 1 of register W3,
- ② set count source by bit 3 of register W3, and
- ③ set the bit 2 of register W3 to "1."

Once count is started, when timer 3 underflows (the set count value is counted), the timer 3 interrupt request flag (T3F) is set to "1", and count continues.

Bit 4 of timer 3 can be used as the timer LC count source for the LCD clock generating.

When bit 2 of register W3 is cleared to "0", timer 3 is initialized to "FFFF₁₆" and count is stopped.

Timer 3 can be used as the counter for clock because it can be operated at clock operating mode (POF instruction execution). When timer 3 underflow occurs at clock operating mode, system returns from the power down state.

When operating timer 3 during clock operating mode, set 1 cycle or more of count source to the following period; from setting bit 2 of register W3 to "1" till executing the POF instruction.

(6) Timer LC

Timer LC is a 4-bit binary down counter with the timer LC reload register (RLC). Data can be set simultaneously in timer LC and the reload register (RLC) with the TLCA instruction. Data cannot be read from timer LC. Stop counting and then execute the TLCA instruction to set timer LC data.

Timer LC starts counting after the following process;

- ① set data in timer LC,
- ② select the count source with the bit 2 of register W4, and
- ③ set the bit 3 of register W4 to "1."

When a value set in reload register RLC is n , timer LC divides the count source signal by $n + 1$ ($n = 0$ to 15).

Once count is started, when timer LC underflows (the next count pulse is input after the contents of timer LC becomes "0"), new data is loaded from reload register RLC, and count continues (auto-reload function).

Timer LC underflow signal divided by 2 can be used for the LCD clock.

(7) Timer input/output pin (C/CNTR pin)

CNTR pin is used to input the timer 1 count source and output the PWM signal generated by timer 2. When the PWM signal is output from C/CNTR pin, set "0" to the output latch of port C.

The selection of CNTR output signal can be controlled by bit 3 of register W2.

When the CNTR input is selected for timer 1 count source, timer 1 counts the waveform of CNTR input selected by bit 0 of register W4. Also, when the CNTR input is selected, the output of port C is invalid (high-impedance state).

(8) Timer interrupt request flags (T1F, T2F, T3F)

Each timer interrupt request flag is set to "1" when each timer underflows. The state of these flags can be examined with the skip instructions (SNZT1, SNZT2, SNZT3).

Use the interrupt control register V1, V2 to select an interrupt or a skip instruction.

An interrupt request flag is cleared to "0" when an interrupt occurs or when the next instruction is skipped with a skip instruction.

(9) Count start synchronization circuit (timer 1)

Timer 1 has the count start synchronous circuit which synchronizes the input of INT pin, and can start the timer count operation.

Timer 1 count start synchronous circuit function is selected by setting the bit 0 of register I1 to "1" and the control by INT pin input can be performed.

When timer 1 count start synchronous circuit is used, the count start synchronous circuit is set, the count source is input to each timer by inputting valid waveform to INT pin.

The valid waveform of INT pin to set the count start synchronous circuit is the same as the external interrupt activated condition.

Once set, the count start synchronous circuit is cleared by clearing the bit I10 to "0" or reset.

However, when the count auto-stop circuit is selected, the count start synchronous circuit is cleared (auto-stop) at the timer 1 underflow.

(10) Count auto-stop circuit (timer 1)

Timer 1 has the count auto-stop circuit which is used to stop timer 1 automatically by the timer 1 underflow when the count start synchronous circuit is used.

The count auto-stop circuit is valid by setting the bit 3 of register W1 to "1". It is cleared by the timer 1 underflow and the count source to timer 1 is stopped.

This function is valid only when the timer 1 count start synchronous circuit is selected.

(11) Precautions

Note the following for the use of timers.

- Prescaler
 - Stop counting and then execute the TABPS instruction to read from prescaler data.
 - Stop counting and then execute the TPSAB instruction to set prescaler data.
- Timer count source
 - Stop timer 1, 2, and LC counting to change its count source.
- Reading the count value
 - Stop timer 1 or 2 counting and then execute the data read instruction (TAB1, TAB2) to read its data.
- Writing to the timer
 - Stop timer 1, 2 or LC counting and then execute the data write instruction (T1AB, T2AB, TLCA) to write its data.
- Writing to reload register R1, R2H
 - When writing data to reload register R1 or reload register R2H while timer 1 or timer 2 is operating, avoid a timing when timer 1 or timer 2 underflows.
- Timer 2
 - Avoid a timing when timer 2 underflows to stop timer 2 at PWM output function used.
 - When "H" interval extension function of the PWM signal is set to be "valid", set "1" or more to reload register R2H.
- Timer 3
 - Stop timer 3 counting to change its count source.
- Timer input/output pin
 - Set the port C output latch to "0" to output the PWM signal from C/CNTR pin.

- Prescaler and Timer 1 count start timing and count time when operation starts
 Count starts from the first rising edge of the count source (2) after Prescaler and Timer 1 operations start (1).
 Time to first underflow (3) is shorter (for up to 1 period of the count source) than time among next underflow (4) by the timing to start the timer and count source operations after count starts.
 When selecting CNTR input as the count source of Timer 1, Timer 1 operates synchronizing with the falling edge of CNTR input.

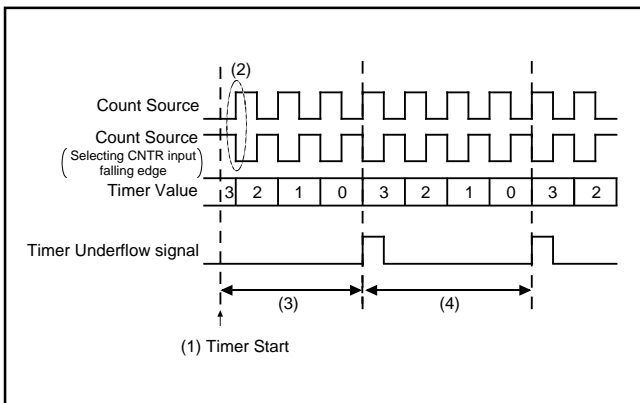


Fig. 24 Timer count start timing and count time when operation starts (Prescaler and Timer 1)

- Timer 2 and Timer LC count start timing and count time when operation starts
 Count starts from the rising edge (2) after the first falling edge of the count source, after Timer 2 and Timer LC operations start (1).
 Time to first underflow (3) is different from time among next underflow (4) by the timing to start the timer and count source operations after count starts.

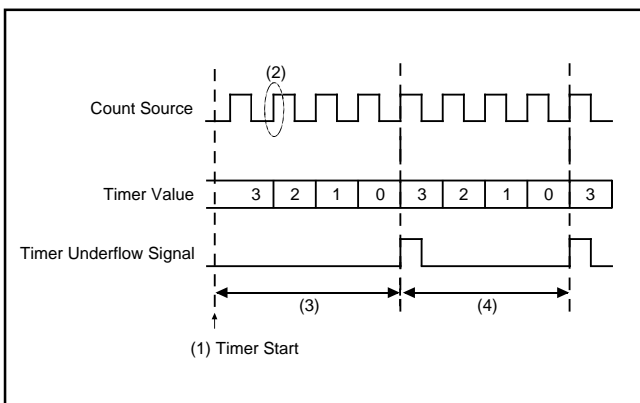


Fig. 25 Timer count start timing and count time when operation starts (Timer 2 and Timer LC)

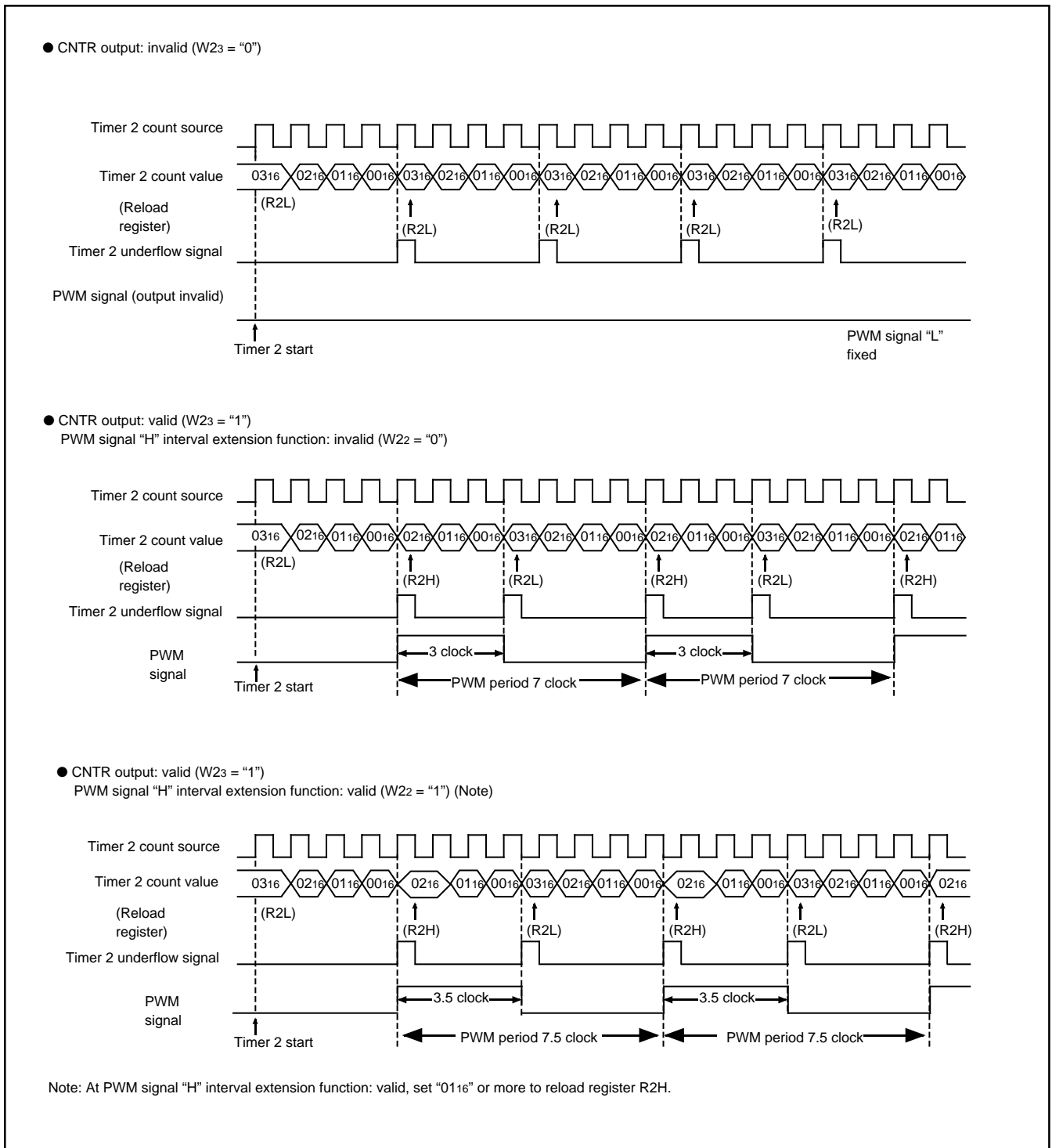
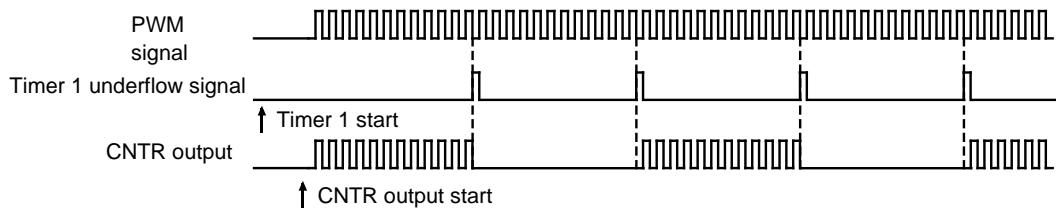


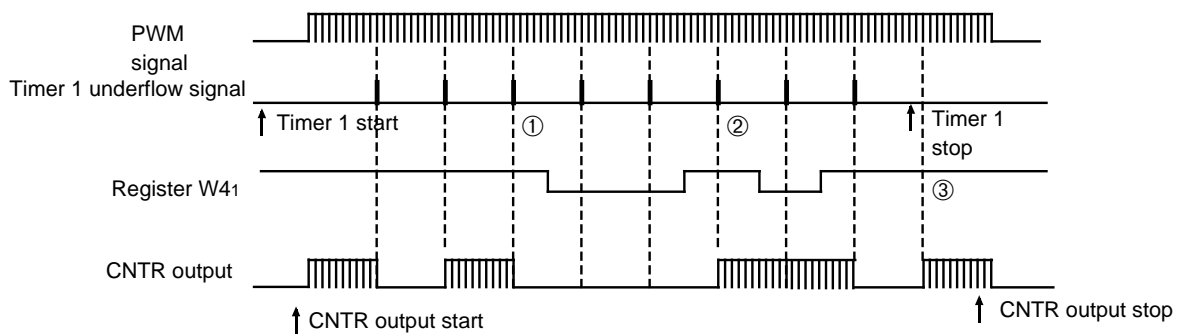
Fig. 26 Timer 2 operation (reload register R2L: "0316", R2H: "0216")

CNTR output auto-control circuit by timer 1 is selected.

- CNTR output: valid ($W23 = "1"$)
CNTR output auto-control circuit selected ($W41 = "1"$)



- CNTR output auto-control function

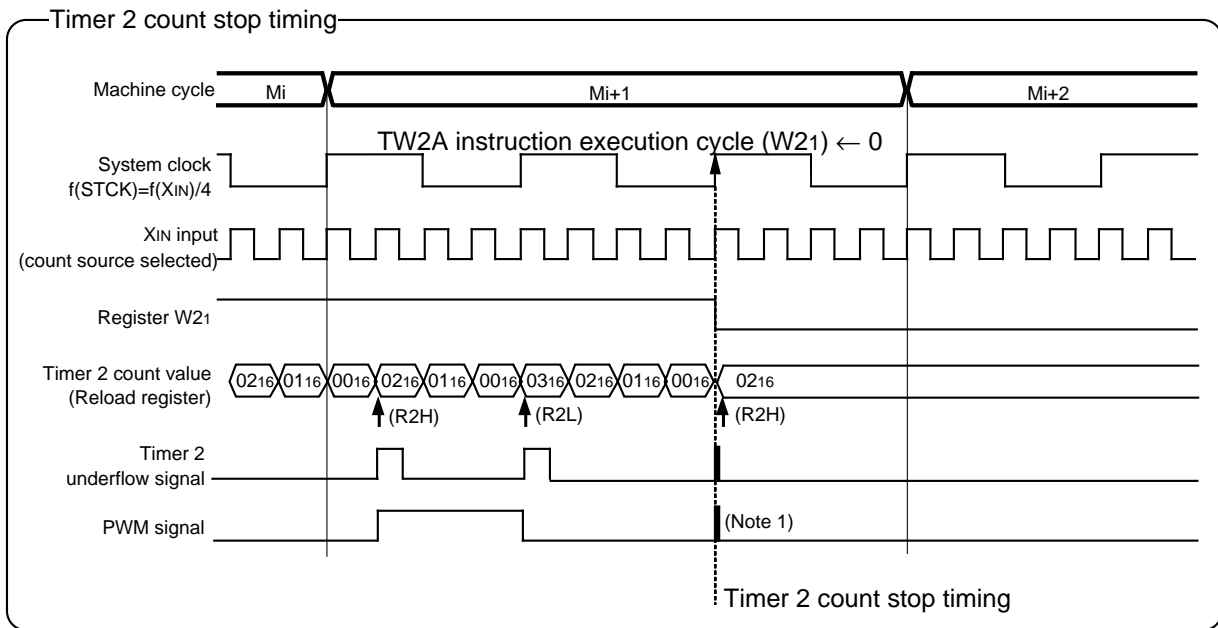
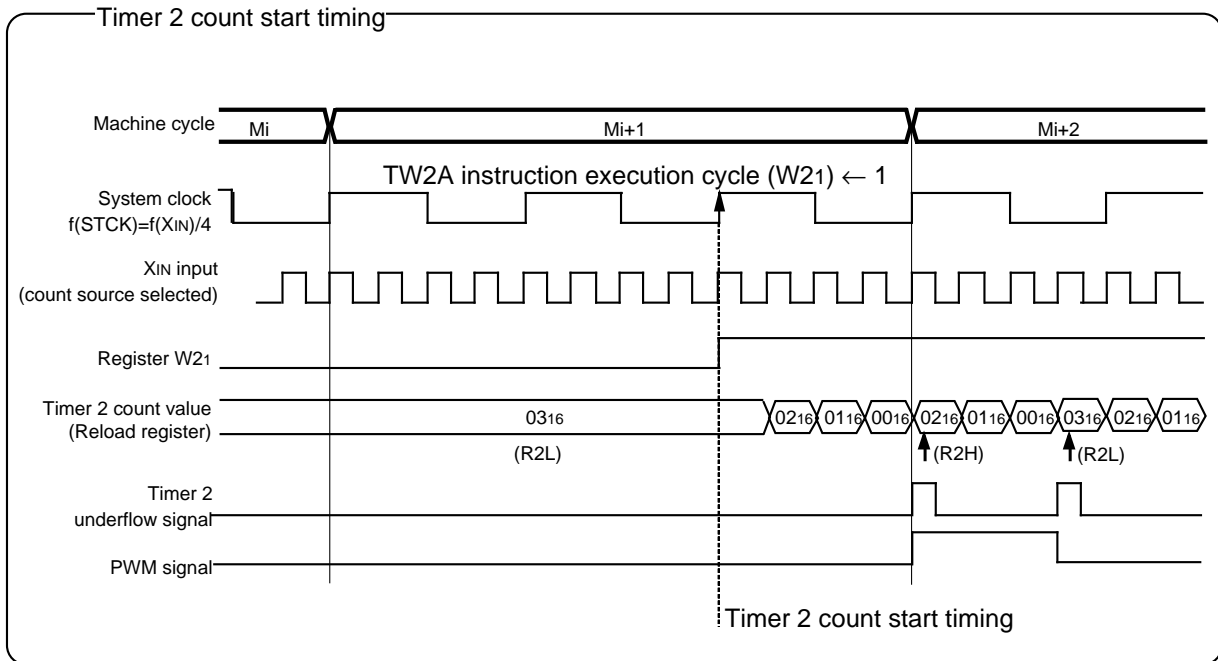


- ① When the CNTR output auto-control function is set to be invalid while the CNTR output is invalid, the CNTR output invalid state is retained.
- ② When the CNTR output auto-control function is set to be invalid while the CNTR output is valid, the CNTR output valid state is retained.
- ③ When timer 1 is stopped, the CNTR output auto-control function becomes invalid.

Note: When the PWM signal is output from C/CNTR pin, set the output latch of port C to "0".

Fig. 27 CNTR output auto-control function by timer 1

- Waveform extension function of CNTR output “H” interval: Invalid ($W22 = “0”$),
 CNTR output: valid ($W23 = “1”$),
 Count source: XIN input selected ($W20 = “0”$),
 Reload register R2L: “0316”
 Reload register R2H: “0216”



Notes 1: In order to stop timer 2 at CNTR output valid ($W23 = “1”$), avoid a timing when timer 2 underflows.
 If these timings overlap, a hazard may occur in a CNTR output waveform.
 2: At CNTR output valid, timer 2 stops after “H” interval of PWM signal set by reload register R2H is output.

Fig. 28 Timer 2 count start/stop timing

WATCHDOG TIMER

Watchdog timer provides a method to reset the system when a program run-away occurs. Watchdog timer consists of timer WDT(16-bit binary counter), watchdog timer enable flag (WEF), and watchdog timer flags (WDF1, WDF2).

The timer WDT downcounts the instruction clocks as the count source from "FFFF₁₆" after system is released from reset.

After the count is started, when the timer WDT underflow occurs (after the count value of timer WDT reaches "0000₁₆," the next count pulse is input), the WDF1 flag is set to "1."

If the WRST instruction is never executed until the timer WDT underflow occurs (until timer WDT counts 65534), WDF2 flag is set to "1," and the RESET pin outputs "L" level to reset the microcomputer.

Execute the WRST instruction at each period of 65534 machine cycle or less by software when using watchdog timer to keep the microcomputer operating normally.

When the WEF flag is set to "1" after system is released from reset, the watchdog timer function is valid.

When the DWDT instruction and the WRST instruction are executed continuously, the WEF flag is cleared to "0" and the watchdog timer function is invalid.

The WEF flag is set to "1" at system reset or RAM back-up mode. The WRST instruction has the skip function. When the WRST instruction is executed while the WDF1 flag is "1," the WDF1 flag is cleared to "0" and the next instruction is skipped.

When the WRST instruction is executed while the WDF1 flag is "0," the next instruction is not skipped.

The skip function of the WRST instruction can be used even when the watchdog timer function is invalid.

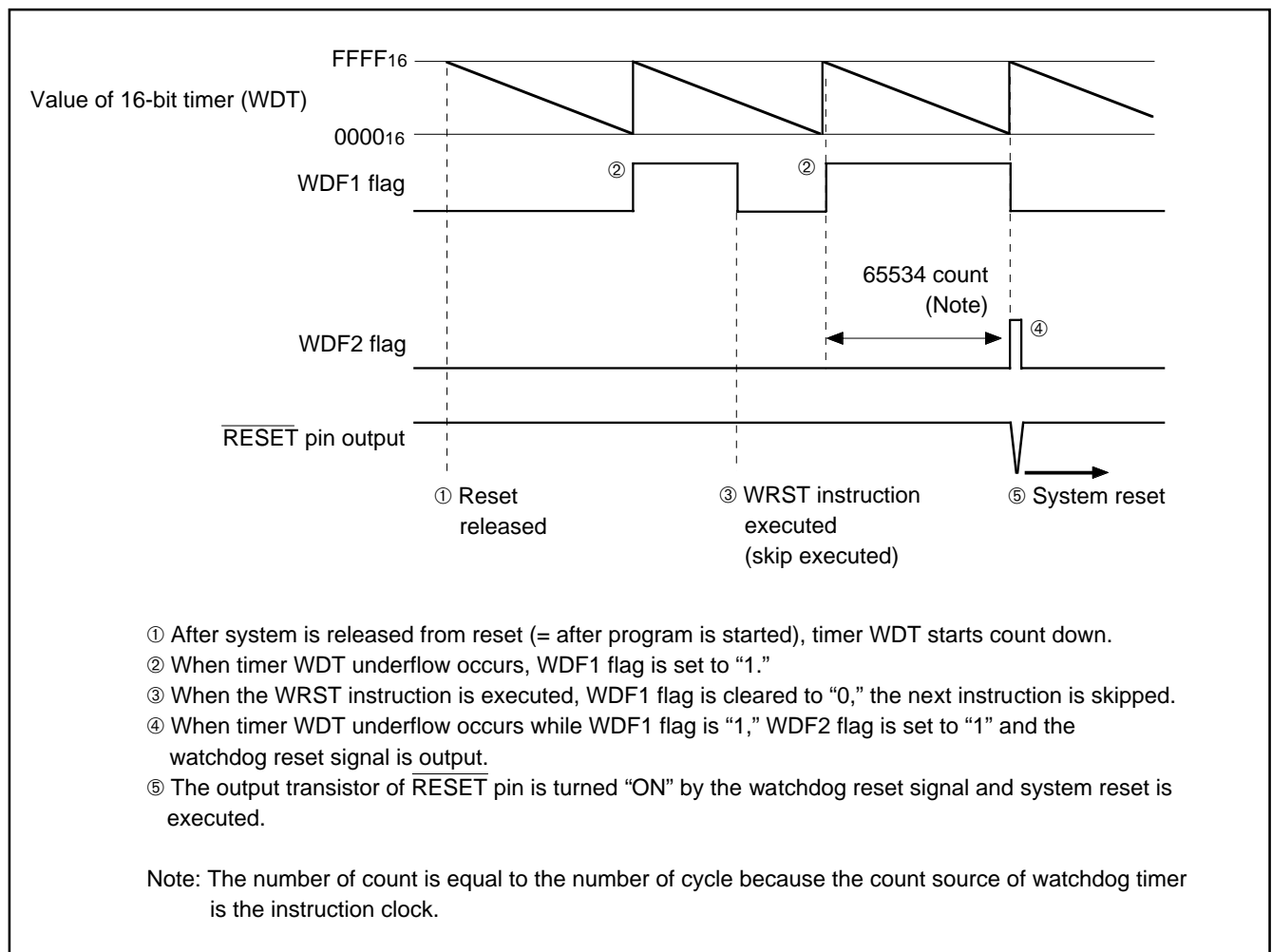


Fig. 29 Watchdog timer function

When the watchdog timer is used, clear the WDF1 flag at the period of 65534 machine cycles or less with the WRST instruction.

When the watchdog timer is not used, execute the DWDT instruction and the WRST instruction continuously (refer to Figure 30).

The watchdog timer is not stopped with only the DWDT instruction. The contents of WDF1 flag and timer WDT are initialized at the power down mode.

When using the watchdog timer and the power down mode, initialize the WDF1 flag with the WRST instruction just before the microcomputer enters the power down state (refer to Figure 31).

The watchdog timer function is valid after system is returned from the power down. When not using the watchdog timer function, execute the DWDT instruction and the WRST instruction continuously every system is returned from the power down, and stop the watchdog timer function.

```

:
WRST      ; WDF1 flag cleared
:
DI
DWDT     ; Watchdog timer function enabled/disabled
WRST     ; WEF and WDF1 flags cleared
:

```

Fig. 30 Program example to start/stop watchdog timer

```

:
WRST     ; WDF1 flag cleared
NOP
DI       ; Interrupt disabled
EPOF    ; POF instruction enabled
POF
↓
Oscillation stop
:

```

Fig. 31 Program example to enter the mode when using the watchdog timer

LCD FUNCTION

The 4552 Group has an LCD (Liquid Crystal Display) controller/driver. When the proper voltage is applied to LCD power supply input pins (VLC1–VLC3) and data are set in timer control register (W4), timer LC, LCD control registers (L1, L2, L3, C1, C2), and LCD RAM, the LCD controller/driver automatically reads the display data and controls the LCD display by setting duty and bias. 4 common signal output pins and 28 segment signal output pins can be used to drive the LCD. By using these pins, up to 112 segments (when 1/4 duty and 1/3 bias are selected) can be controlled to display. The LCD power input pins (VLC1–VLC3) are also used as pins SEG0–SEG2. When SEG0–SEG2 are selected, the internal power (VDD) is used for the LCD power.

(1) Duty and bias

There are 3 combinations of duty and bias for displaying data on the LCD. Use bits 0 and 1 of LCD control register (L1) to select the proper display method for the LCD panel being used.

- 1/2 duty, 1/2 bias
- 1/3 duty, 1/3 bias
- 1/4 duty, 1/3 bias

Table 11 Duty and maximum number of displayed pixels

| Duty | Maximum number of displayed pixels | Used COM pins |
|------|------------------------------------|-------------------|
| 1/2 | 56 segments | COM0, COM1 (Note) |
| 1/3 | 84 segments | COM0–COM2 (Note) |
| 1/4 | 112 segments | COM0–COM3 |

Note: Leave unused COM pins open.

(2) LCD clock control

The LCD clock is determined by the timer LC count source selection bit (W42), timer LC control bit (W43), and timer LC. Accordingly, the frequency (F) of the LCD clock is obtained by the following formula. Numbers (① to ③) shown below the formula correspond to numbers in Figure 32, respectively.

- When using the prescaler output (ORCLK) as timer LC count source (W42="1")

$$F = \underbrace{\text{ORCLK}}_{\text{①}} \times \underbrace{\frac{1}{\text{LC} + 1}}_{\text{②}} \times \underbrace{\frac{1}{2}}_{\text{③}}$$

- When using the bit 4 of timer 3 as timer LC count source (W42="0")

$$F = \underbrace{\text{T34}}_{\text{①}} \times \underbrace{\frac{1}{\text{LC} + 1}}_{\text{②}} \times \underbrace{\frac{1}{2}}_{\text{③}}$$

[LC: 0 to 15]

The frame frequency and frame period for each display method can be obtained by the following formula:

$$\text{Frame frequency} = \frac{F}{n} \text{ (Hz)}$$

$$\text{Frame period} = \frac{n}{F} \text{ (s)}$$

[F: LCD clock frequency
1/n: Duty]

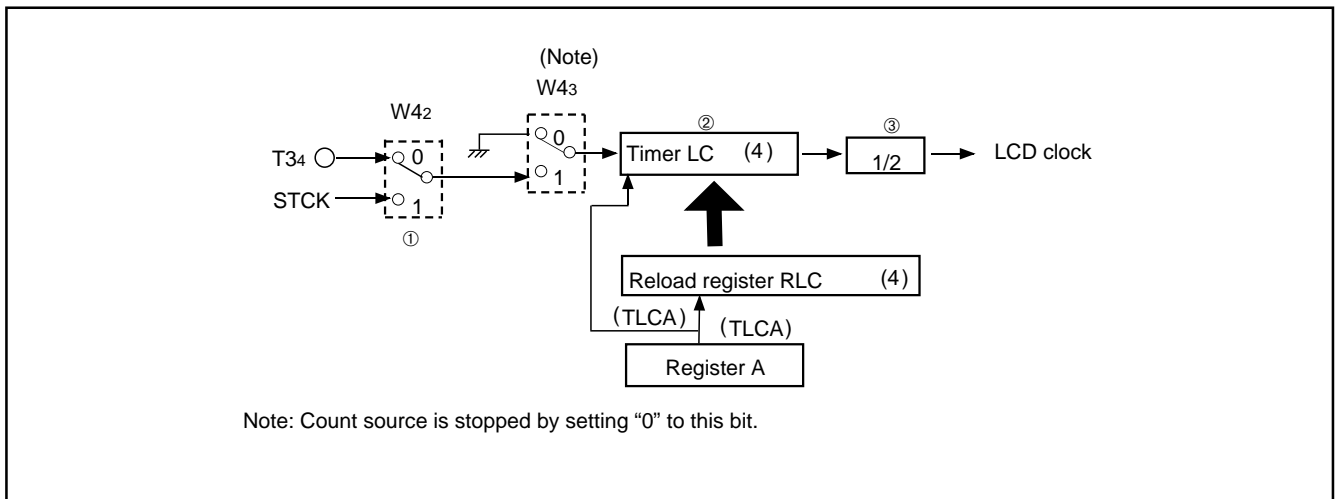


Fig. 32 LCD clock control circuit structure

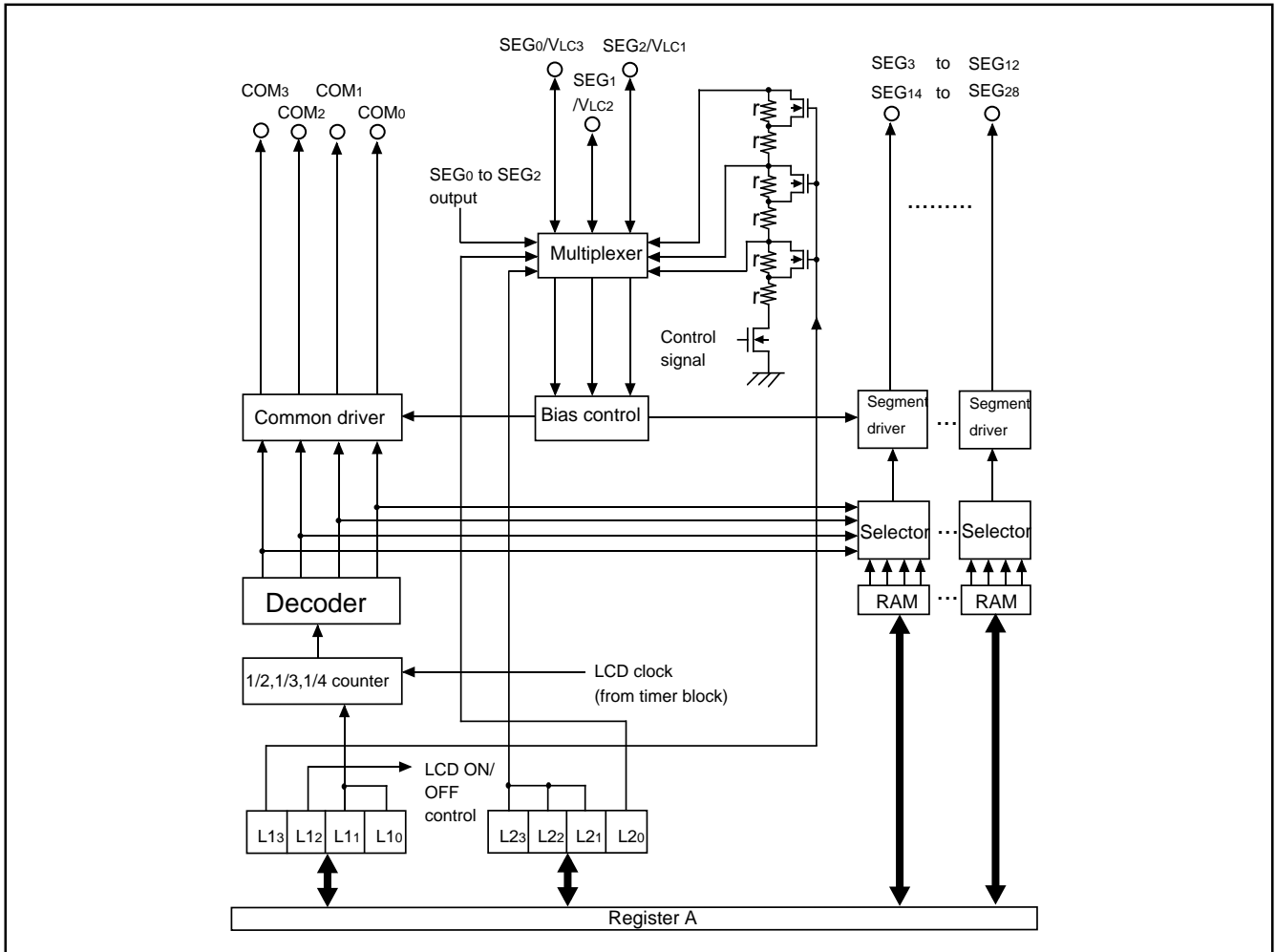


Fig. 33 LCD controller/driver

(3) LCD RAM

RAM contains areas corresponding to the liquid crystal display. When “1” is written to this LCD RAM, the display pixel corresponding to the bit is automatically displayed.

(4) LCD drive waveform

When “1” is written to a bit in the LCD RAM data, the voltage difference between common pin and segment pin which correspond to the bit automatically becomes $|V_{LC3}|$ and the display pixel at the cross section turns on.

When returning from reset, and in the RAM back-up mode, a display pixel turns off because every segment output pin and common output pin becomes V_{LC3} level.

| Z X Y | Bits | | | | | | | | | | | | | | | |
|-------------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | 0 | | | | 1 | | | | 2 | | | | 3 | | | |
| Bits | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 |
| 8 | SEG0 | SEG0 | SEG0 | SEG0 | SEG8 | SEG8 | SEG8 | SEG8 | SEG16 | SEG16 | SEG16 | SEG16 | SEG24 | SEG24 | SEG24 | SEG24 |
| 9 | SEG1 | SEG1 | SEG1 | SEG1 | SEG9 | SEG9 | SEG9 | SEG9 | SEG17 | SEG17 | SEG17 | SEG17 | SEG25 | SEG25 | SEG25 | SEG25 |
| 10 | SEG2 | SEG2 | SEG2 | SEG2 | SEG10 | SEG10 | SEG10 | SEG10 | SEG18 | SEG18 | SEG18 | SEG18 | SEG26 | SEG26 | SEG26 | SEG26 |
| 11 | SEG3 | SEG3 | SEG3 | SEG3 | SEG11 | SEG11 | SEG11 | SEG11 | SEG19 | SEG19 | SEG19 | SEG19 | SEG27 | SEG27 | SEG27 | SEG27 |
| 12 | SEG4 | SEG4 | SEG4 | SEG4 | SEG12 | SEG12 | SEG12 | SEG12 | SEG20 | SEG20 | SEG20 | SEG20 | SEG28 | SEG28 | SEG28 | SEG28 |
| 13 | SEG5 | SEG5 | SEG5 | SEG5 | — | — | — | — | SEG21 | SEG21 | SEG21 | SEG21 | — | — | — | — |
| 14 | SEG6 | SEG6 | SEG6 | SEG6 | SEG14 | SEG14 | SEG14 | SEG14 | SEG22 | SEG22 | SEG22 | SEG22 | — | — | — | — |
| 15 | SEG7 | SEG7 | SEG7 | SEG7 | SEG15 | SEG15 | SEG15 | SEG15 | SEG23 | SEG23 | SEG23 | SEG23 | — | — | — | — |
| COM | COM3 | COM2 | COM1 | COM0 | COM3 | COM2 | COM1 | COM0 | COM3 | COM2 | COM1 | COM0 | COM3 | COM2 | COM1 | COM0 |

Note: The area marked “ — ” is not the LCD display RAM.

Fig. 34 LCD RAM map

Table 12 LCD control registers (1)

| LCD control register L1 | | at reset : 0000 ₂ | | at power down : state retained | R/W TAL1/TL1A |
|-------------------------|--|------------------------------|----------------|--------------------------------|------------------|
| L13 | Internal dividing resistor for LCD power supply selection bit (Note 2) | 0 | 2r X 3, 2r X 2 | | |
| | | 1 | r X 3, r X 2 | | |
| L12 | LCD control bit | 0 | Stop | | |
| | | 1 | Operating | | |
| L11 | LCD duty and bias selection bits | L11 | L10 | Duty | Bias |
| | | 0 | 0 | Not available | |
| L10 | | 0 | 1 | 1/2 | 1/2 |
| | | 1 | 0 | 1/3 | 1/3 |
| | | 1 | 1 | 1/4 | 1/3 |

| LCD control register L2 | | at reset : 0000 ₂ | | at power down : state retained | W TL2A |
|-------------------------|---|------------------------------|------------------------------------|--------------------------------|-----------|
| L23 | SEG0/VLC3 pin function switch bit (Note 3) | 0 | SEG0 | | |
| | | 1 | VLC3 | | |
| L22 | SEG1/VLC2 pin function switch bit (Note 4) | 0 | SEG1 | | |
| | | 1 | VLC2 | | |
| L21 | SEG2/VLC1 pin function switch bit (Note 4) | 0 | SEG2 | | |
| | | 1 | VLC1 | | |
| L20 | Internal dividing resistor for LCD power supply control bit | 0 | Internal dividing resistor valid | | |
| | | 1 | Internal dividing resistor invalid | | |

| LCD control register L3 | | at reset : 1111 ₂ | | at power down : state retained | W TL3A |
|-------------------------|-----------------------------------|------------------------------|-------|--------------------------------|-----------|
| L33 | P23/SEG20 pin function switch bit | 0 | SEG20 | | |
| | | 1 | P23 | | |
| L32 | P22/SEG19 pin function switch bit | 0 | SEG19 | | |
| | | 1 | P22 | | |
| L31 | P21/SEG18 pin function switch bit | 0 | SEG18 | | |
| | | 1 | P21 | | |
| L30 | P20/SEG17 pin function switch bit | 0 | SEG17 | | |
| | | 1 | P20 | | |

Notes 1: "R" represents read enabled, and "W" represents write enabled.

2: "r (resistor) multiplied by 3" is used at 1/3 bias, and "r multiplied by 2" is used at 1/2 bias.

3: VLC3 is connected to VDD internally when SEG0 pin is selected.

4: Use internal dividing resistor when SEG1 and SEG2 pins are selected.

Table 12 LCD control registers (2)

| LCD control register C1 | | at reset : 11112 | | at power down : state retained | W TC1A |
|-------------------------|-----------------------------------|------------------|-------|--------------------------------|-----------|
| C13 | P03/SEG24 pin function switch bit | 0 | SEG24 | | |
| | | 1 | P03 | | |
| C12 | P02/SEG23 pin function switch bit | 0 | SEG23 | | |
| | | 1 | P02 | | |
| C11 | P01/SEG22 pin function switch bit | 0 | SEG22 | | |
| | | 1 | P01 | | |
| C10 | P00/SEG21 pin function switch bit | 0 | SEG21 | | |
| | | 1 | P00 | | |

| LCD control register C2 | | at reset : 11112 | | at power down : state retained | W TC2A |
|-------------------------|-----------------------------------|------------------|-------|--------------------------------|-----------|
| C23 | P13/SEG28 pin function switch bit | 0 | SEG28 | | |
| | | 1 | P13 | | |
| C22 | P12/SEG27 pin function switch bit | 0 | SEG27 | | |
| | | 1 | P12 | | |
| C21 | P11/SEG26 pin function switch bit | 0 | SEG26 | | |
| | | 1 | P11 | | |
| C20 | P10/SEG25 pin function switch bit | 0 | SEG25 | | |
| | | 1 | P10 | | |

Note: "R" represents read enabled, and "W" represents write enabled.

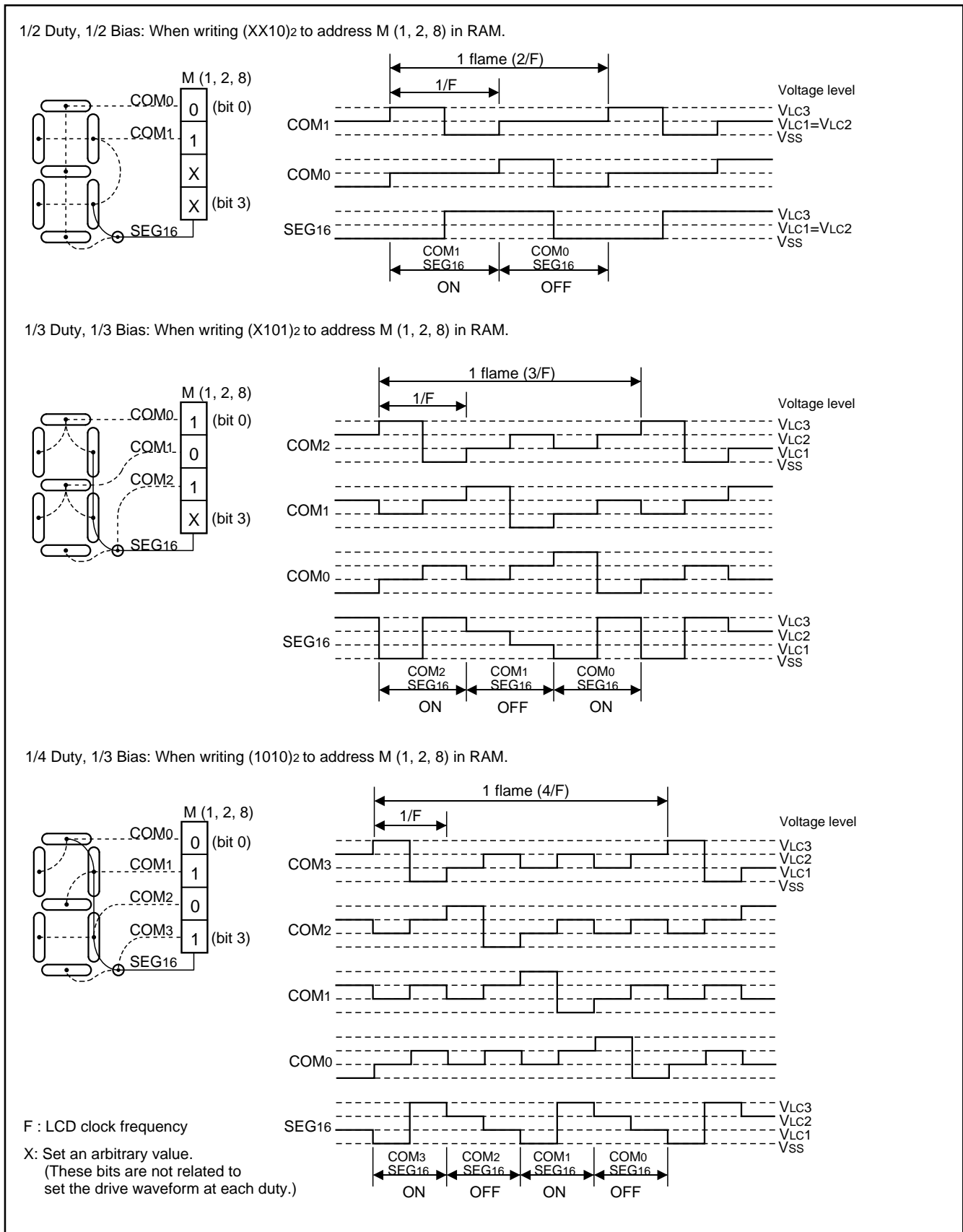


Fig. 35 LCD controller/driver structure

(5) LCD power supply circuit

Select the LCD power supply circuit suitable for the using LCD panel.

The LCD power supply circuit is fixed by the followings;

- The internal dividing resistor is controlled by bit 0 of register L2.
- The internal dividing resistor is selected by bit 3 of register L1.
- The bias condition is selected by bits 0 and 1 of register L1.

● Internal dividing resistor

The 4552 Group has the internal dividing resistor for LCD power supply.

When bit 0 of register L2 is set to "0", the internal dividing resistor is valid. However, when the LCD is turned off by setting bit 2 of register L1 to "0", the internal dividing resistor is turned off.

The same six resistor (r) is prepared for the internal dividing resistor. According to the setting value of bit 3 of register L1 and using bias condition, the resistor is prepared as follows;

- L13 = "0", 1/3 bias used: $2r \times 3 = 6r$
- L13 = "0", 1/2 bias used: $2r \times 2 = 4r$
- L13 = "1", 1/3 bias used: $r \times 3 = 3r$
- L13 = "1", 1/2 bias used: $r \times 2 = 2r$

● VLC3/SEG0 pin

The selection of VLC3/SEG0 pin function is controlled with the bit 3 of register L2.

When the VLC3 pin function is selected, apply voltage of $VLC3 < VDD$ to the pin externally.

When the SEG0 pin function is selected, VLC3 is connected to VDD internally.

● VLC2/SEG1, VLC1/SEG2 pin

The selection of VLC2/SEG1 pin function is controlled with the bit 2 of register L2.

The selection of VLC1/SEG2 pin function is controlled with the bit 1 of register L2.

When the VLC2 pin and VLC1 pin functions are selected and the internal dividing resistor is not used, apply voltage of $0 < VLC1 < VLC2 < VLC3$ to these pins. Short the VLC2 pin and VLC1 pin at 1/2 bias.

When the VLC2 pin and VLC1 pin functions are selected and the internal dividing resistor is used, the dividing voltage value generated internally is output from the VLC1 pin and VLC2 pin. The VLC2 pin and VLC1 pin have the same electric potential at 1/2 bias. When SEG1 and SEG2 pin functions are selected, use the internal dividing resistor. In this time, VLC2 and VLC1 are connected to the generated dividing voltage.

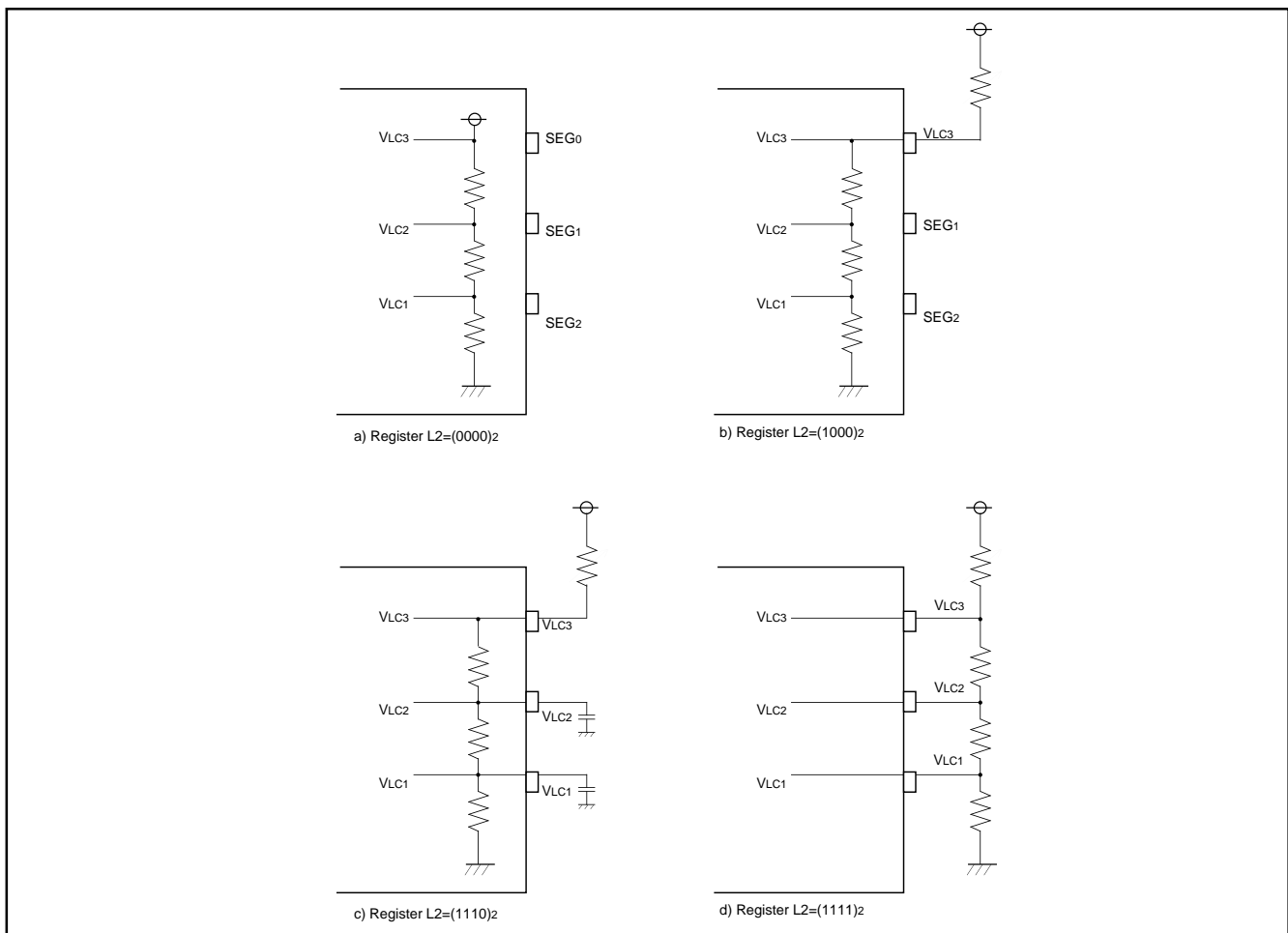


Fig. 36 LCD power supply circuit example (1/3 bias condition selected)

RESET FUNCTION

System reset is performed by applying “L” level to $\overline{\text{RESET}}$ pin for 1 machine cycle or more when the following condition is satisfied; the value of supply voltage is the minimum value or more of the recommended operating conditions. Then when “H” level is applied to $\overline{\text{RESET}}$ pin, software starts from address 0 in page 0.

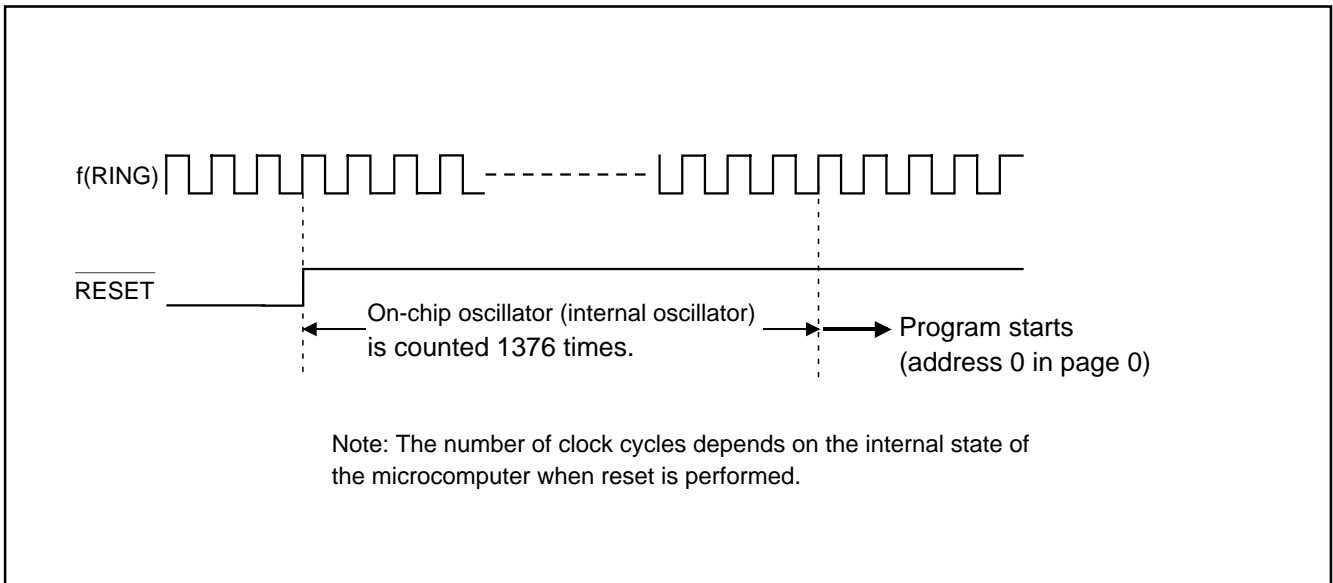


Fig. 37 Reset release timing

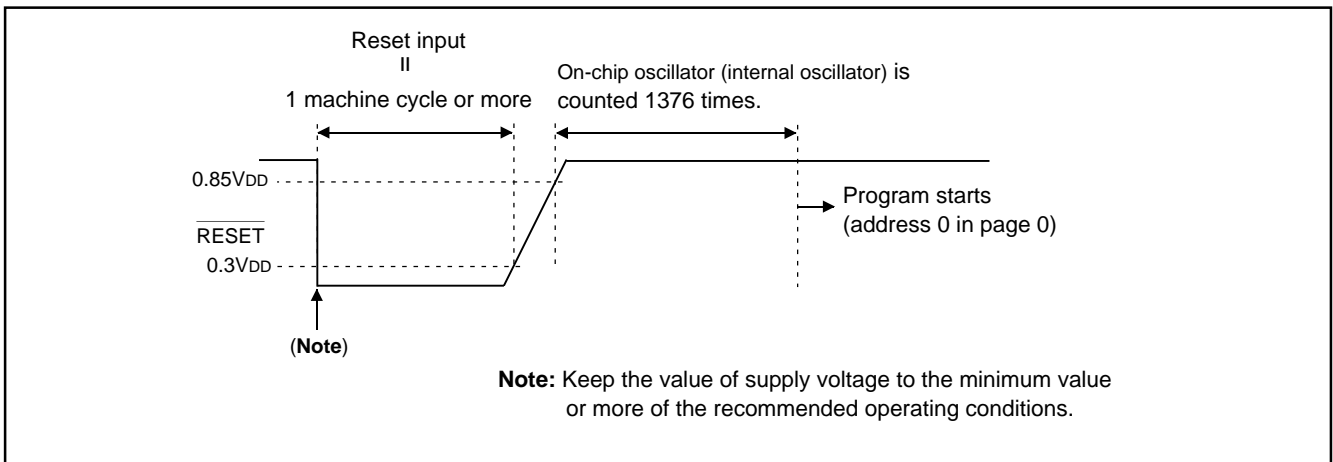


Fig. 38 $\overline{\text{RESET}}$ pin input waveform and reset operation

(1) Power-on reset

Reset can be automatically performed at power on (power-on reset) by the built-in power-on reset circuit. When the built-in power-on reset circuit is used, set the time for the supply voltage to rise from 0 V to the minimum voltage of recommended operating conditions to 100 μs or less.

If the rising time exceeds 100 μs, connect a capacitor between the RESET pin and Vss at the shortest distance, and input “L” level to RESET pin until the value of supply voltage reaches the minimum operating voltage.

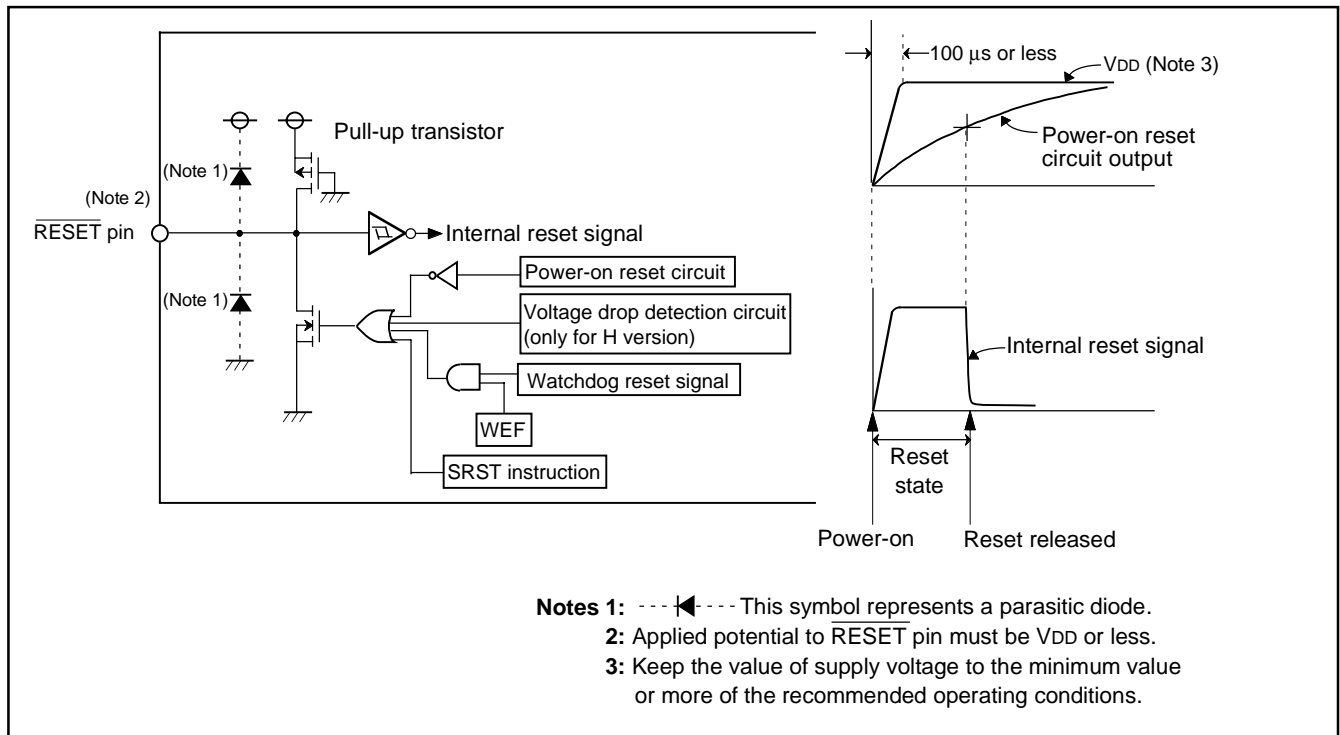


Fig. 39 Structure of reset pin and its peripherals,, and power-on reset operation

Table 13 Port state at reset

| Name | Function | State |
|-------------------------|-------------------------|--------------------------------|
| D0–D4 | D0–D4 | High-impedance (Notes 1, 2) |
| D5/INT | D5 | High-impedance (Notes 1, 2) |
| XCIN/D6, XCOUT/D7 | XCIN, XCOUT | Sub-clock input |
| P00/SEG21–P03/SEG24 | P00–P03 | High-impedance (Notes 1, 2, 3) |
| P10/SEG25–P13/SEG28 | P10–P13 | High-impedance (Notes 1, 2, 3) |
| P20/SEG17–P23/SEG20 | P20–P23 | High-impedance (Notes 1, 2, 3) |
| SEG0/VLC3–SEG2/VLC1 | SEG0–SEG2 | VLC3 (VDD) level |
| SEG3–SEG12, SEG14–SEG16 | SEG3–SEG12, SEG14–SEG16 | VLC3 (VDD) level |
| COM0–COM3 | COM0–COM3 | VLC3 (VDD) level |
| C/CNTR | C | “L” (Vss) level |

Notes 1: Output latch is set to “1.”
 2: Output structure is N-channel open-drain.
 3: Pull-up transistor is turned OFF.

(2) Internal state at reset

Figure 40 shows internal state at reset (they are the same after system is released from reset). The contents of timers, registers, flags and RAM except shown in Figure 40 are undefined, so set the initial value to them.

| | | |
|---|--------------------------------|----------------------|
| • Program counter (PC) | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | |
| Address 0 in page 0 is set to program counter. | | |
| • Interrupt enable flag (INTE) | 0 | (Interrupt disabled) |
| • Power down flag (P) | 0 | |
| • External 0 interrupt request flag (EXF0) | 0 | |
| • Interrupt control register V1 | 0 0 0 0 | (Interrupt disabled) |
| • Interrupt control register V2 | 0 0 0 0 | (Interrupt disabled) |
| • Interrupt control register I1 | 0 0 0 0 | |
| • Timer 1 interrupt request flag (T1F) | 0 | |
| • Timer 2 interrupt request flag (T2F) | 0 | |
| • Timer 3 interrupt request flag (T3F) | 0 | |
| • Watchdog timer flags (WDF1, WDF2) | 0 | |
| • Watchdog timer enable flag (WEF) | 1 | |
| • Timer control register PA | 0 | (Prescaler stopped) |
| • Timer control register W1 | 0 0 0 0 | (Timer 1 stopped) |
| • Timer control register W2 | 0 0 0 0 | (Timer 2 stopped) |
| • Timer control register W3 | 0 0 0 0 | (Timer 3 stopped) |
| • Timer control register W4 | 0 0 0 0 | (Timer LC stopped) |
| • Clock control register MR | 1 1 0 0 | |
| • Clock control register RG | 0 0 0 | |
| • LCD control register L1 | 0 0 0 0 | |
| • LCD control register L2 | 0 0 0 0 | |
| • LCD control register L3 | 1 1 1 1 | |
| • LCD control register C1 | 1 1 1 1 | |
| • LCD control register C2 | 1 1 1 1 | |
| • Key-on wakeup control register K0 | 0 0 0 0 | |
| • Key-on wakeup control register K1 | 0 0 0 0 | |
| • Key-on wakeup control register K2 | 0 0 0 0 | |
| • Pull-up control register PU0 | 0 0 0 0 | |
| • Pull-up control register PU1 | 0 0 0 0 | |
| • Port output structure control register FR0 | 0 0 0 0 | |
| • Port output structure control register FR1 | 0 0 0 0 | |
| • Port output structure control register FR2 | 0 0 0 0 | |
| • Carry flag (CY) | 0 | |
| • High-order bit reference enable flag (UPTF) | 0 | |
| • Register A | 0 0 0 0 | |
| • Register B | 0 0 0 0 | |
| • Register D | X X X | |
| • Register E | X X X X X X X X | |
| • Register X | 0 0 0 0 | |
| • Register Y | 0 0 0 0 | |
| • Register Z | X X | |
| • Stack pointer (SP) | 1 1 1 | |
| • Operation source clock | On-chip oscillator (operating) | |
| • Ceramic resonator circuit | Operating | |
| • RC oscillation circuit | Stop | |
| • Quartz-crystal oscillator | Operating | |

“X” represents undefined.

Fig. 40 Internal state at reset

VOLTAGE DROP DETECTION CIRCUIT (only for H version)

The built-in voltage drop detection circuit is designed to detect a drop in voltage and to reset the microcomputer if the supply voltage drops below a set value.

(1) SVDE instruction

When the SVDE instruction is executed, the voltage drop detection circuit is valid even after system enters into the power down mode. The SVDE instruction can be executed only once.

In order to release the execution of the SVDE instruction, the system reset is required.

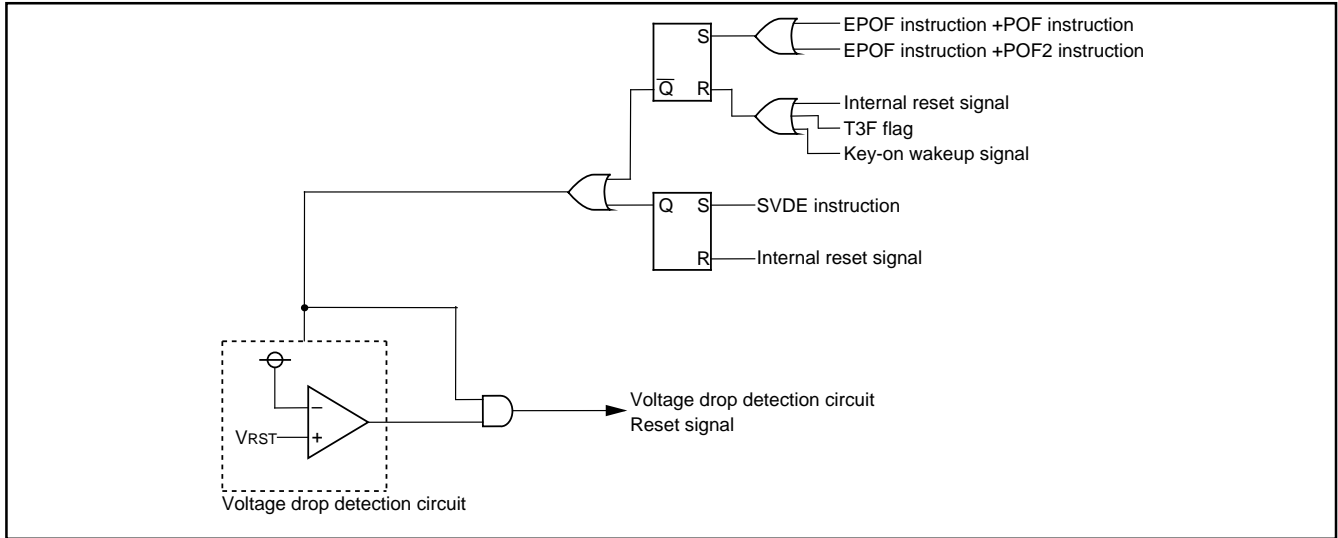


Fig. 41 Voltage drop detection reset circuit

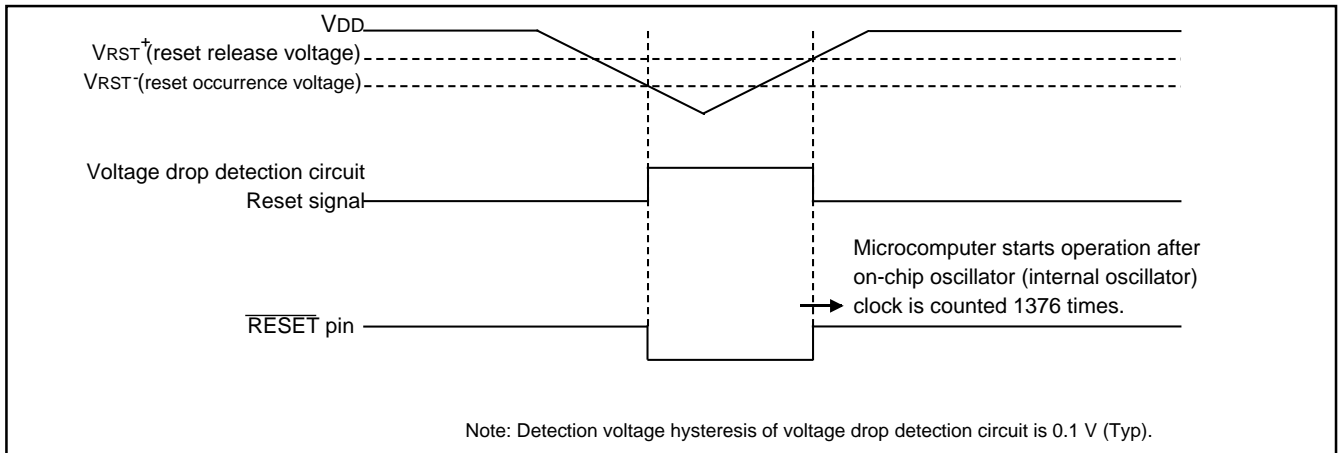


Fig. 42 Voltage drop detection circuit operation waveform

(2) Note on voltage drop detection circuit

The voltage drop detection circuit detection voltage of this product is set up lower than the minimum value of the supply voltage of the recommended operating conditions.

When the supply voltage of a microcomputer falls below to the minimum value of recommended operating conditions and re-goes up (ex. battery exchange of an application product), depending on the capacity value of the bypass capacitor added to the power supply pin, the following case may cause program failure (Figure 43);

supply voltage does not fall below to VRST+, and its voltage re-goes up with no reset.
In such a case, please design a system which supply voltage is once reduced below to VRST- and re-goes up after that.

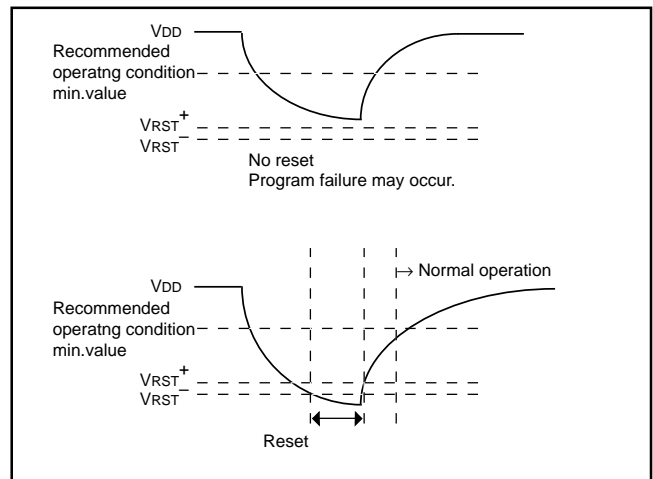


Fig. 43 VDD and VRST

POWER DOWN FUNCTION

The 4552 Group has 2-type power down functions.

System enters into each power down state by executing the following instructions.

- Clock operating mode EPOF and POF instructions
- RAM back-up mode EPOF and POF2 instructions

When the EPOF instruction is not executed before the POF or POF2 instruction is executed, these instructions are equivalent to the NOP instruction.

(1) Clock operating mode

The following functions and states are retained.

- RAM
- Reset circuit
- XCIN–XCOUT oscillation
- LCD display
- Timer 3

(2) RAM back-up mode

The following functions and states are retained.

- RAM
- Reset circuit

(3) Warm start condition

The system returns from the power down state when;

- External wakeup signal is input
- Timer 3 underflow occurs

in the power down mode.

In either case, the CPU starts executing the software from address 0 in page 0. In this case, the P flag is "1."

(4) Cold start condition

The CPU starts executing the software from address 0 in page 0 when;

- reset pulse is input to $\overline{\text{RESET}}$ pin,
- reset by watchdog timer is performed, or
- reset by the voltage drop detection circuit is performed.

In this case, the P flag is "0."

(5) Identification of the start condition

Warm start or cold start can be identified by examining the state of the power down flag (P) with the SNZP instruction. The warm start condition from the clock operating mode can be identified by examining the state of T3F flag.

Table 15 Functions and states retained at power down mode

| Function | Power down mode | |
|--|-----------------|-------------|
| | Clock operating | RAM back-up |
| Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) (Note 2) | X | X |
| Contents of RAM | O | O |
| Interrupt control registers V1, V2 | X | X |
| Interrupt control register I1 | O | O |
| Selected oscillation circuit | O | O |
| Clock control register MR, RG | O | O |
| Timer 1 to timer 2 functions | (Note 3) | (Note 3) |
| Timer 3 function | O | (Note 3) |
| Timer LC function | O | (Note 3) |
| Watchdog timer function | X (Note 4) | X (Note 4) |
| Timer control registers PA | X | X |
| Timer control registers W1 to W4 | O | O |
| LCD display function | O | (Note 5) |
| LCD control registers L1 to L3, C1, C2 | O | O |
| Voltage drop detection circuit | (Note 6) | (Note 6) |
| Port level | (Note 7) | (Note 7) |
| Pull-up control registers PU0, PU1 | O | O |
| Key-on wakeup control registers K0 to K2 | O | O |
| Port output structure control registers FR0 to FR2 | O | O |
| External interrupt request flag (EXF0) | X | X |
| Timer interrupt request flags (T1F, T2F) | (Note 3) | (Note 3) |
| Timer interrupt request flag (T3F) | O | (Note 3) |
| Interrupt enable flag (INTE) | X | X |
| Watchdog timer flags (WDF1, WDF2) | X (Note 4) | X (Note 4) |
| Watchdog timer enable flag (WEF) | X (Note 4) | X (Note 4) |

Notes 1: "O" represents that the function can be retained, and "X" represents that the function is initialized.

Registers and flags other than the above are undefined at RAM back-up, and set an initial value after returning.

2: The stack pointer (SP) points the level of the stack register and is initialized to "7" at RAM back-up.

3: The state of the timer is undefined.

4: Initialize the watchdog timer with the WRST instruction, and then go into the power down state.

5: LCD is turned off.

6: When the SVDE instruction is executed, this function is valid at power down.

7: In the RAM back-up mode, C/CNTR pin outputs "L" level.

However, when the CNTR input is selected (W11, W10="11"), C/CNTR pin is in an input enabled state (output = high-impedance). Other ports retain their respective output levels.

(6) Return signal

An external wakeup signal or timer 3 interrupt request flag (T3F) is used to return from the clock operating mode.

An external wakeup signal is used to return from the RAM back-up mode because the oscillation is stopped.

Table 16 shows the return condition for each return source.

(7) Control registers

- Key-on wakeup control register K0

Register K0 controls the ports P0 and P1 key-on wakeup function. Set the contents of this register through register A with the TK0A instruction. In addition, the TAK0 instruction can be used to transfer the contents of register K0 to register A.

- Key-on wakeup control register K1

Register K1 controls the return condition and the selection of valid waveform/level of port P1. Set the contents of this register through register A with the TK1A instruction. In addition, the TAK1 instruction can be used to transfer the contents of register K0 to register A.

- Key-on wakeup control register K2

Register K2 controls the INT pin key-on wakeup function and the selection of return condition. Set the contents of this register through register A with the TK2A instruction. In addition, the TAK2 instruction can be used to transfer the contents of register K2 to register A.

- Pull-up control register PU0

Register PU0 controls the ON/OFF of the port P0 pull-up transistor. Set the contents of this register through register A with the TPU0A instruction. In addition, the TAPU0 instruction can be used to transfer the contents of register PU0 to register A.

- Pull-up control register PU1

Register PU1 controls the ON/OFF of the port P1 pull-up transistor. Set the contents of this register through register A with the TPU1A instruction. In addition, the TAPU1 instruction can be used to transfer the contents of register PU1 to register A.

- External interrupt control register I1

Register I1 controls the valid waveform of the external 0 interrupt, the input control of INT pin and the return input level. Set the contents of this register through register A with the TI1A instruction. In addition, the TAI1 instruction can be used to transfer the contents of register I1 to register A.

Table 16 Return source and return condition

| | Return source | Return condition | Remarks |
|------------------------|--------------------------------------|--|---|
| External wakeup signal | Ports P00–P03 | Return by an external falling edge (“H”→“L”). | The key-on wakeup function can be selected by two port unit. |
| | Ports P10–P13 | Return by an external “H” level or “L” level input, or rising edge (“L”→“H”) or falling edge (“H”→“L”). Return by an external “L” level input. | The key-on wakeup function can be selected by two port unit. Select the return level (“L” level or “H” level) and return condition (return by level or edge) with register K1 according to the external state before going into the power down state. |
| | INT pin | Return by an external “H” level or “L” level input, or rising edge (“L”→“H”) or falling edge (“H”→“L”). When the return level is input, the interrupt request flag (EXF0) is not set. | Select the return level (“L” level or “H” level) with register I1 and return condition (return by level or edge) with register K2 according to the external state before going into the power down state. |
| | Timer 3 interrupt request flag (T3F) | Return by timer 3 underflow or by setting T3F to “1”. It can be used in the clock operating mode. | Clear T3F with the SNZT3 instruction before system enters into the power down state. When system enters into the power down state while T3F is “1”, system returns from the state immediately because it is recognized as return condition. |

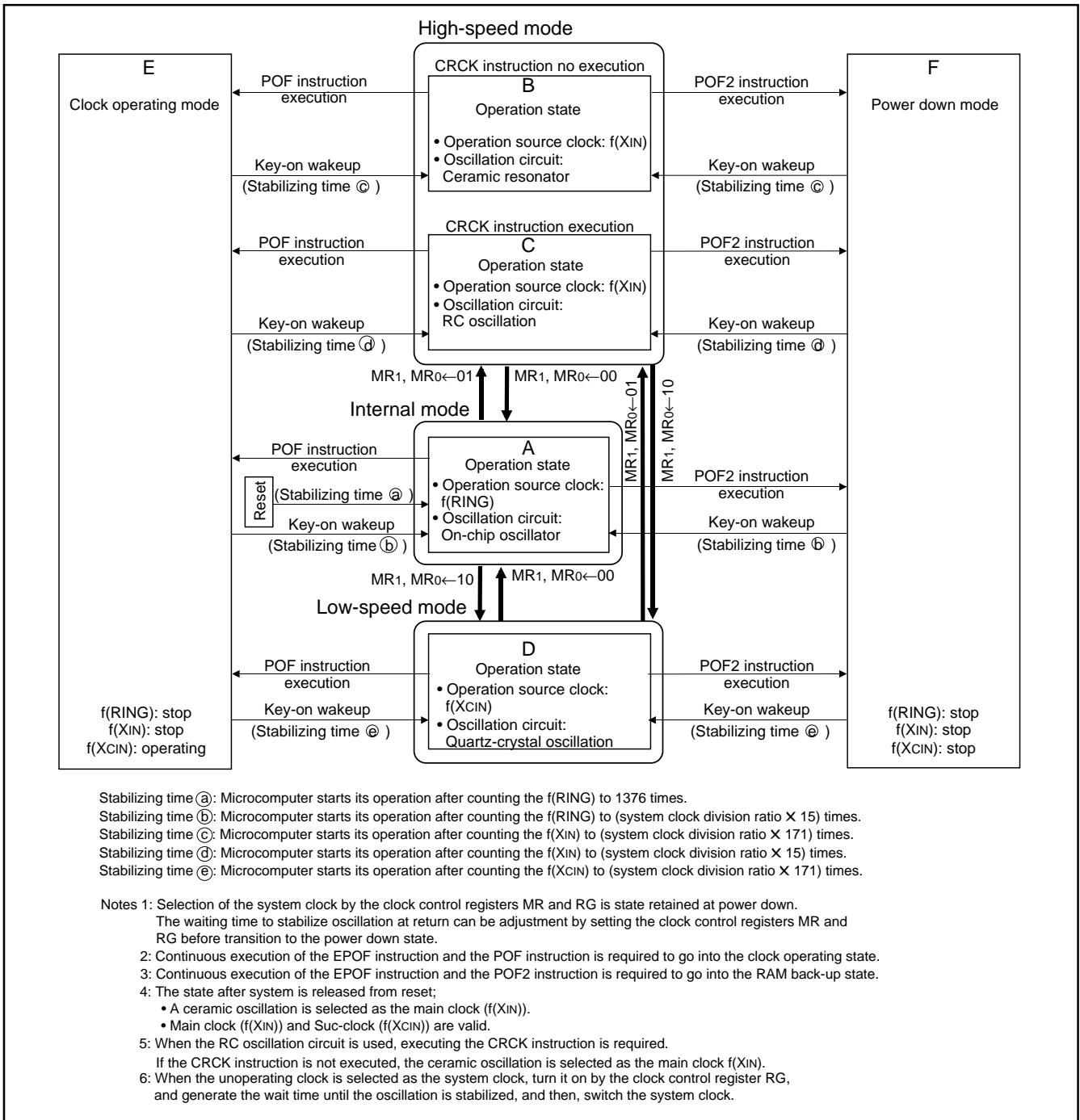


Fig. 44 State transition

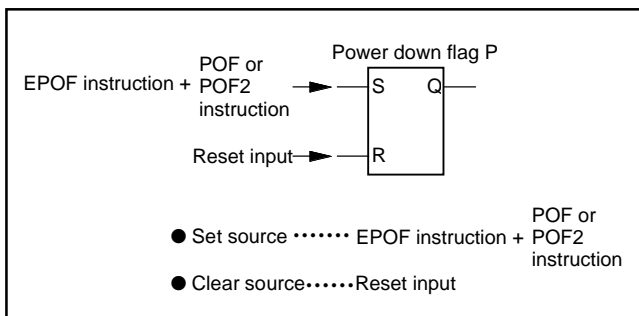


Fig. 45 Set source and clear source of the P flag

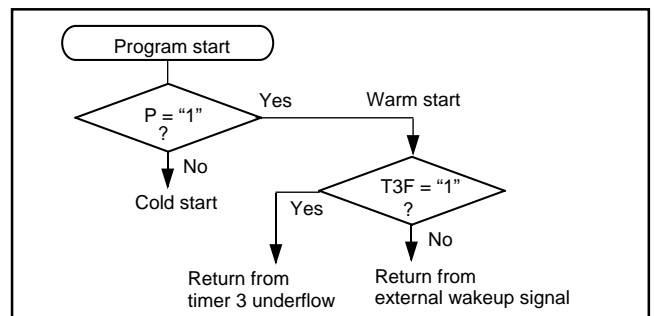


Fig. 46 Start condition identified example using the SNZP instruction

Table 17 Key-on wakeup control register, pull-up control register and interrupt control register

| Key-on wakeup control register K0 | | at reset : 00002 | | at power down : state retained | R/W TAK0/ TK0A |
|-----------------------------------|--|------------------|------------------------|--------------------------------|----------------------|
| K03 | Port P12, P13 key-on wakeup control bit (Note 3) | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used | | |
| K02 | Port P10, P11 key-on wakeup control bit (Note 2) | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used | | |
| K01 | Port P02, P03 key-on wakeup control bit | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used | | |
| K00 | Port P00, P01 key-on wakeup control bit | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used | | |

| Key-on wakeup control register K1 | | at reset : 00002 | | at power down : state retained | R/W TAK1/ TK1A |
|-----------------------------------|--|------------------|----------------------------|--------------------------------|----------------------|
| K13 | Ports P12, P13 return condition selection bit (Note 3) | 0 | Returned by edge | | |
| | | 1 | Returned by level | | |
| K12 | Ports P12, P13 valid waveform/level selection bit (Note 3) | 0 | Falling waveform/"L" level | | |
| | | 1 | Rising waveform/"H" level | | |
| K11 | Ports P10, P11 return condition selection bit (Note 2) | 0 | Returned by edge | | |
| | | 1 | Returned by level | | |
| K10 | Ports P10, P11 valid waveform/level selection bit (Note 2) | 0 | Falling waveform/"L" level | | |
| | | 1 | Rising waveform/"H" level | | |

| Key-on wakeup control register K2 | | at reset : 00002 | | at power down : state retained | R/W TAK2/ TK2A |
|-----------------------------------|--|------------------|--|--------------------------------|----------------------|
| K23 | Not used | 0 | This bit has no function, but read/write is enabled. | | |
| | | 1 | | | |
| K22 | Not used | 0 | This bit has no function, but read/write is enabled. | | |
| | | 1 | | | |
| K21 | INT pin return condition selection bit | 0 | Returned by level | | |
| | | 1 | Returned by edge | | |
| K20 | INT pin key-on wakeup control bit | 0 | Key-on wakeup invalid | | |
| | | 1 | Key-on wakeup valid | | |

Notes 1: "R" represents read enabled, and "W" represents write enabled.

2: To be invalid (K02 = "0") key-on wakeup of ports P10 and P11, set the registers K10 and K11 to "0".

3: To be invalid (K03 = "0") key-on wakeup of ports P12 and P13, set the registers K12 and K13 to "0".

| Pull-up control register PU0 | | at reset : 00002 | | at power down : state retained | R/W TAPU0/ TPU0A |
|------------------------------|---|------------------|------------------------|--------------------------------|------------------------|
| PU03 | Port P03 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU02 | Port P02 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU01 | Port P01 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU00 | Port P00 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |

| Pull-up control register PU1 | | at reset : 00002 | | at power down : state retained | R/W TAPU1/ TPU1A |
|------------------------------|---|------------------|------------------------|--------------------------------|------------------------|
| PU13 | Port P13 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU12 | Port P12 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU11 | Port P11 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU10 | Port P10 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |

| Interrupt control register I1 | | at reset : 00002 | | at power down : state retained | R/W TAI1/T11A |
|-------------------------------|--|------------------|---|--------------------------------|------------------|
| I13 | INT pin input control bit (Note 2) | 0 | INT pin input disabled | | |
| | | 1 | INT pin input enabled | | |
| I12 | Interrupt valid waveform for INT pin/ return level selection bit (Note 2) | 0 | Falling waveform/"L" level ("L" level is recognized with the SNZIO instruction) | | |
| | | 1 | Rising waveform/"H" level ("H" level is recognized with the SNZIO instruction) | | |
| I11 | INT pin edge detection circuit control bit | 0 | One-sided edge detected | | |
| | | 1 | Both edges detected | | |
| I10 | INT pin Timer 1 count start synchronous circuit selection bit | 0 | Timer 1 count start synchronous circuit not selected | | |
| | | 1 | Timer 1 count start synchronous circuit selected | | |

Notes 1: "R" represents read enabled, and "W" represents write enabled.

2: When the contents of I12 and I13 are changed, the external interrupt request flag (EXF0) may be set.

CLOCK CONTROL

The clock control circuit consists of the following circuits.

- On-chip oscillator (internal oscillator)
- Ceramic resonator
- RC oscillation circuit
- Quartz-crystal oscillation circuit
- Multi-plexer (clock selection circuit)
- Frequency divider
- Internal clock generating circuit

The system clock and the instruction clock are generated as the source clock for operation by these circuits.

Figure 47 shows the structure of the clock control circuit.

The 4552 Group operates by the on-chip oscillator clock ($f(\text{RING})$) which is the internal oscillator after system is released from reset. Also, the ceramic resonator or the RC oscillation can be used for the main clock ($f(\text{XIN})$) of the 4552 Group.

The quartz-crystal oscillator can be used for sub-clock ($f(\text{XCIN})$).

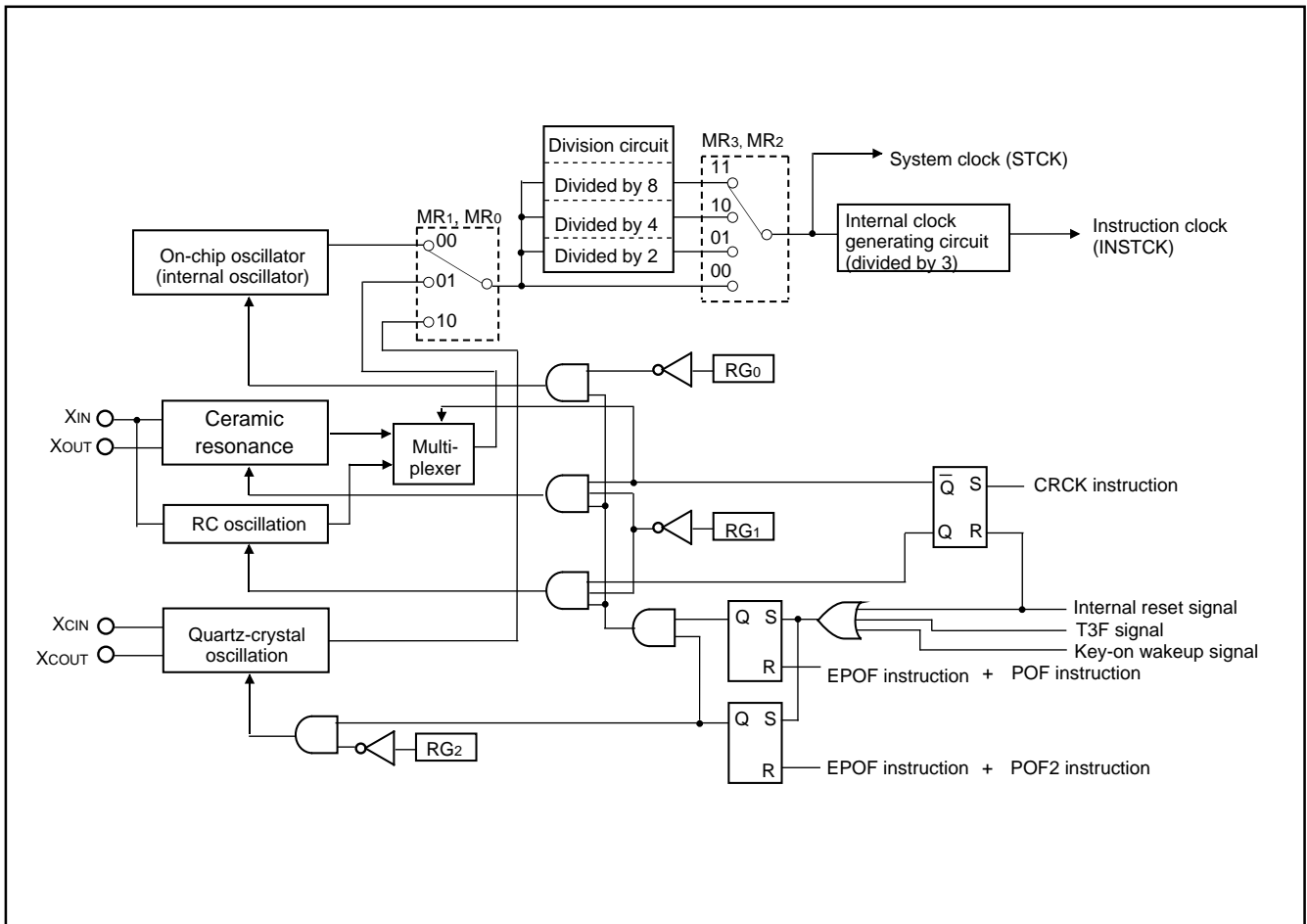


Fig. 47 Clock control circuit structure

(1) On-chip oscillator operation

After system is released from reset, the MCU starts operation by the clock output from the on-chip oscillator which is the internal oscillator.

The clock frequency of the on-chip oscillator depends on the supply voltage and the operation temperature range.

Be careful that variable frequencies when designing application products.

(2) Main clock generating circuit ($f(X_{IN})$)

When the MCU operates by the ceramic resonator or the RC oscillator as the main clock ($f(X_{IN})$).

After system is released from reset, the ceramic oscillation is valid for main clock.

The ceramic oscillation is invalid and the RC oscillation circuit is valid with the CRCK instruction.

The CRCK instruction can be executed only once.

Execute the CRCK instruction in the initial setting routine (executing it in address 0 in page 0 is recommended).

When the main clock ($f(X_{IN})$) is not used, connect X_{IN} pin to V_{SS} and leave X_{OUT} pin open, and do not execute the CRCK instruction (Figure 49).

(3) Ceramic resonator

When the ceramic resonator is used as the main clock ($f(X_{IN})$), connect the ceramic resonator and the external circuit to pins X_{IN} and X_{OUT} at the shortest distance. A feedback resistor is built in between pins X_{IN} and X_{OUT} (Figure 50). Do not execute the CRCK instruction in program.

(4) RC oscillation

When the RC oscillation is used as the main clock ($f(X_{IN})$), connect the X_{IN} pin to the external circuit of resistor R and the capacitor C at the shortest distance and leave X_{OUT} pin open. Then, execute the CRCK instruction (Figure 51).

The frequency is affected by a capacitor, a resistor and a micro-computer. So, set the constants within the range of the frequency limits.

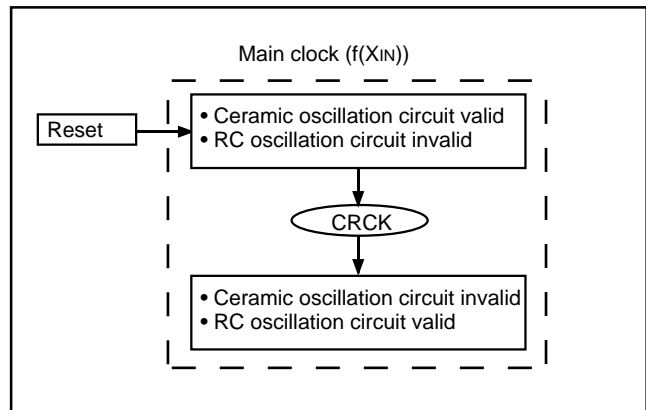


Fig. 48 Switch to ceramic oscillation/RC oscillation

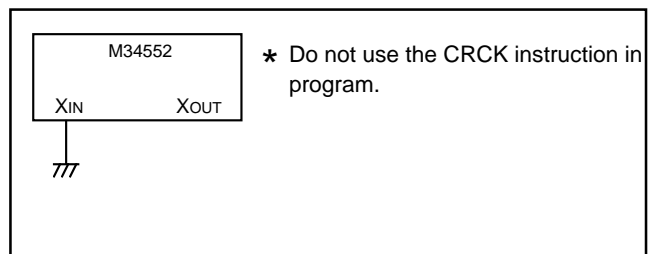


Fig. 49 Handling of X_{IN} and X_{OUT} when operating on-chip oscillator

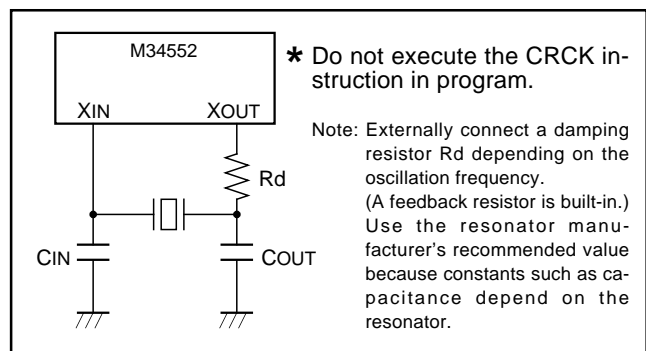


Fig. 50 Ceramic resonator external circuit

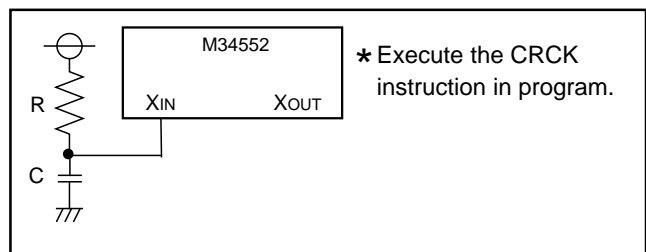


Fig. 51 External RC circuit

(5) External clock

When the external clock signal is used as the main clock ($f(XIN)$), connect the XIN pin to the clock source and leave XOUT pin open. (Figure 52). Do not execute the CRCK instruction in program.

Be careful that the maximum value of the oscillation frequency when using the external clock differs from the value when using the ceramic resonator (refer to the recommended operating condition). Also, note that the power down mode (POF and POF2 instructions) cannot be used when using the external clock.

(6) Sub-clock generating circuit $f(XCIN)$

Sub-clock signal $f(XCIN)$ is obtained by externally connecting a quartz-crystal oscillator. Connect this external circuit and a quartz-crystal oscillator to pins XCIN and XCOU at the shortest distance. A feedback resistor is built in between pins XCIN and XCOU (Figure 53). XCIN pin and XCOU pin are also used as ports D6 and D7, respectively. The sub-clock oscillation circuit is invalid and the function of ports D6 and D7 are valid by setting bit 2 of register RG to "1". When sub-clock, ports D6 and D7 are not used, connect XCIN/D6 to VSS and leave XCOU/D7 open.

(7) Clock control register MR

Register MR controls system clock. Set the contents of this register through register A with the TMRA instruction. In addition, the TAMR instruction can be used to transfer the contents of register MR to register A.

(8) Clock control register RG

Register RG controls the start/stop of each oscillation circuit. Set the contents of this register through register A with the TRGA instruction.

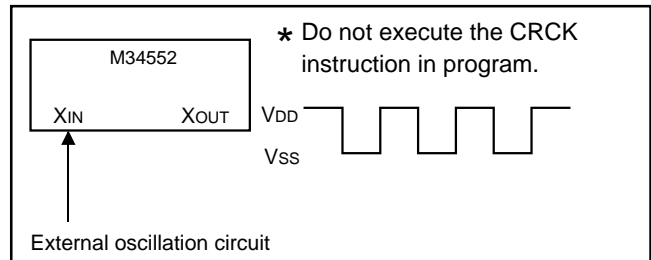


Fig. 52 External clock input circuit

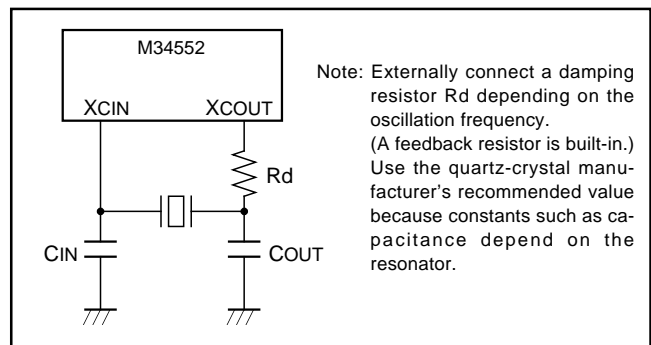


Fig. 53 External quartz-crystal circuit

ROM ORDERING METHOD

- 1.Mask ROM Order Confirmation Form*
- 2.Mark Specification Form*
- 3.Data to be written to ROM... one floppy disk.

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

Table 18 Clock control registers

| Clock control register MR | | at reset : 11002 | | at power down : state retained | R/W TAMR/ TMRA |
|---------------------------|--------------------------------------|------------------|-----|--------------------------------|----------------------|
| MR3 | Operation mode selection bits | MR3 | MR2 | Operation mode | |
| | | 0 | 0 | Through mode | |
| | | 0 | 1 | Frequency divided by 2 mode | |
| MR2 | | 1 | 0 | Frequency divided by 4 mode | |
| | | 1 | 1 | Frequency divided by 8 mode | |
| MR3 | System clock selection bits (Note 2) | MR1 | MR0 | System clock | |
| | | 0 | 0 | $f(RING)$ | |
| | | 0 | 1 | $f(XIN)$ | |
| MR2 | | 1 | 0 | $f(XCIN)$ | |
| | | 1 | 1 | Not available (Note 3) | |

| Clock control register RG | | at reset : 0002 | | at power down : state retained | W TRGA |
|---------------------------|---|-----------------|---|--------------------------------|-----------|
| RG2 | Sub-clock ($f(XCIN)$) control bit (Note 4) | 0 | Sub-clock ($f(XCIN)$) oscillation available, ports D6 and D7 not selected | | |
| | | 1 | Sub-clock ($f(XCIN)$) oscillation stop, ports D6 and D7 selected | | |
| RG1 | Main-clock ($f(XIN)$) control bit (Note 4) | 0 | Main clock ($f(XIN)$) oscillation available | | |
| | | 1 | Main clock ($f(XIN)$) oscillation stop | | |
| RG0 | On-chip oscillator ($f(RING)$) control bit (Note 4) | 0 | On-chip oscillator ($f(RING)$) oscillation available | | |
| | | 1 | On-chip oscillator ($f(RING)$) oscillation stop | | |

Notes 1: "R" represents read enabled, and "W" represents write enabled.

2: The stopped clock cannot be selected for system clock.

3: "11" cannot be set to the low-order 2 bits (MR1, MR0) of register MR.

4: The oscillation circuit selected for system clock cannot be stopped.

NOTES ON NOISE

Countermeasures against noise are described below.

The following countermeasures are effective against noise in theory, however, it is necessary not only to take measures as follows but to evaluate before actual use.

1. Shortest wiring length

(1) Wiring for $\overline{\text{RESET}}$ pin

Make the length of wiring which is connected to the $\overline{\text{RESET}}$ pin as short as possible. Especially, connect a capacitor across the $\overline{\text{RESET}}$ pin and the Vss pin with the shortest possible wiring.

<Reason>

In order to reset a microcomputer correctly, 1 machine cycle or more of the width of a pulse input into the $\overline{\text{RESET}}$ pin is required. If noise having a shorter pulse width than this is input to the $\overline{\text{RESET}}$ input pin, the reset is released before the internal state of the microcomputer is completely initialized. This may cause a program runaway.

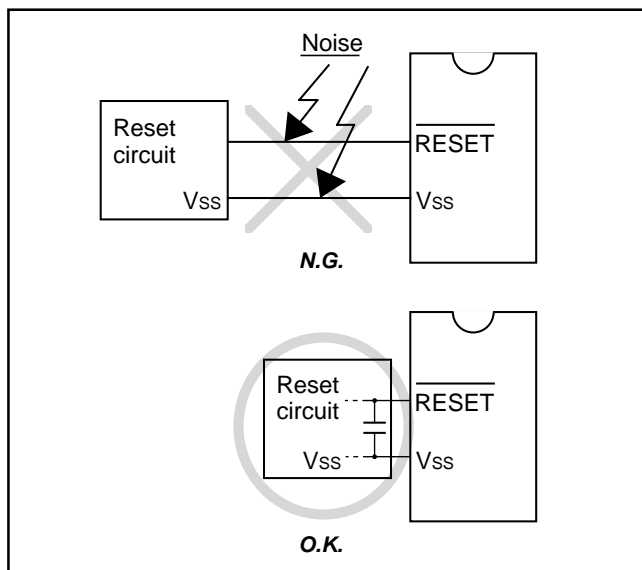


Fig. 54 Wiring for the $\overline{\text{RESET}}$ pin

(2) Wiring for clock input/output pins

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

<Reason>

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

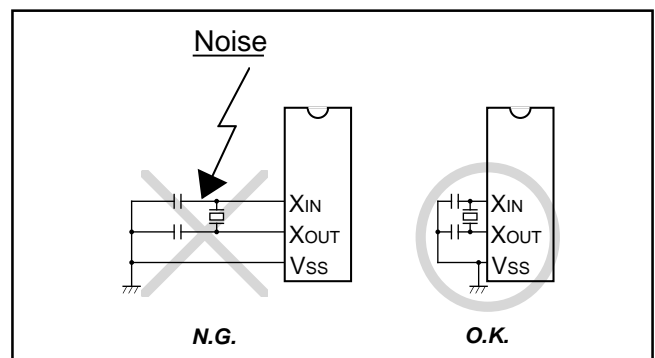


Fig. 55 Wiring for clock I/O pins

(3) Wiring to CNVss pin

Connect CNVss pin to a GND pattern at the shortest distance.

The GND pattern is required to be as close as possible to the GND supplied to Vss.

In order to improve the noise reduction, to connect a 5 k Ω resistor serially to the CNVss pin - GND line may be valid.

As well as the above-mentioned, in this case, connect to a GND pattern at the shortest distance. The GND pattern is required to be as close as possible to the GND supplied to Vss.

<Reason>

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

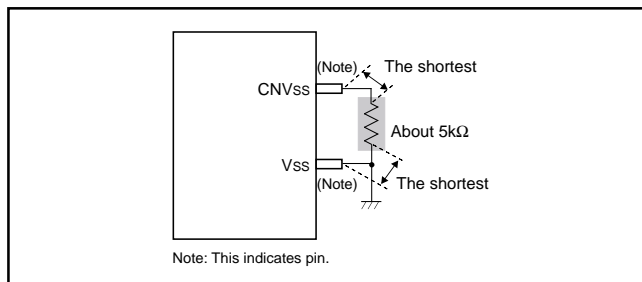


Fig. 56 Wiring for the CNVss pin of the One Time PROM

2. Connection of bypass capacitor across Vss line and Vdd line

Connect an approximately 0.1 μ F bypass capacitor across the Vss line and the Vdd line as follows:

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

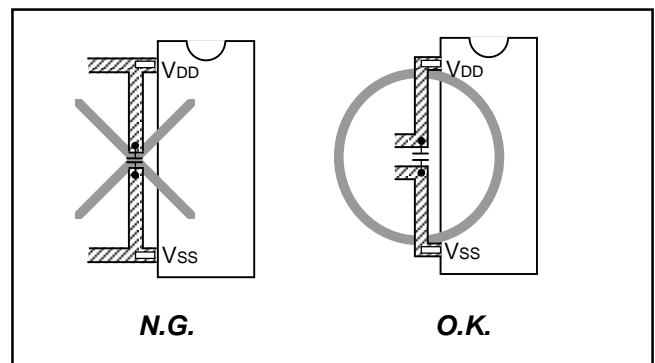


Fig. 57 Bypass capacitor across the Vss line and the Vdd line

3. Oscillator concerns

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

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

<Reason>

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

(2) Installing oscillator away from signal lines where potential levels change frequently
 Install an oscillator and a connecting pattern of an oscillator away from signal lines where potential levels change frequently. Also, do not cross such signal lines over the clock lines or the signal lines which are sensitive to noise.

<Reason>

Signal lines where potential levels change frequently (such as the CNTR pin signal line) may affect other lines at signal rising edge or falling edge. If such lines cross over a clock line, clock waveforms may be deformed, which causes a microcomputer failure or a program runaway.

(3) Oscillator protection using Vss pattern

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

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

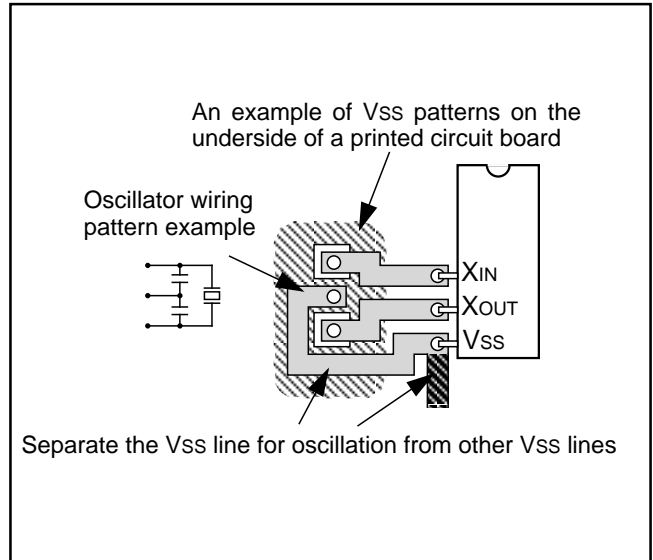


Fig. 60 Vss pattern on the underside of an oscillator

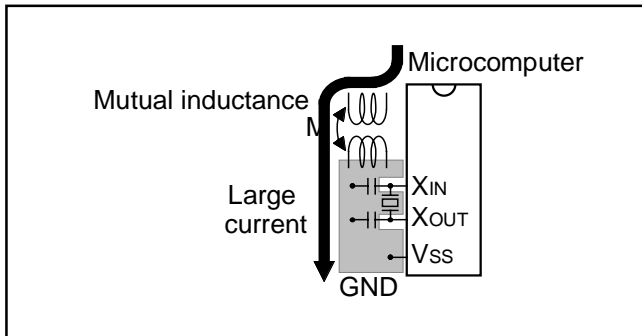


Fig. 58 Wiring for a large current signal line

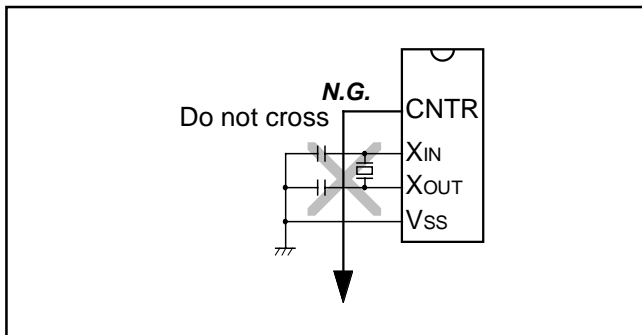


Fig. 59 Wiring to a signal line where potential levels change frequently

4. Setup for I/O ports

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

<Hardware>

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

<Software>

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

5. Providing of watchdog timer function by software

If a microcomputer runs away because of noise or others, it can be detected by a software watchdog timer and the microcomputer can be reset to normal operation. This is equal to or more effective than program runaway detection by a hardware watchdog timer. The following shows an example of a watchdog timer provided by software.

In the following example, to reset a microcomputer to normal operation, the main routine detects errors of the interrupt processing routine and the interrupt processing routine detects errors of the main routine.

This example assumes that interrupt processing is repeated multiple times in a single main routine processing.

<The main routine>

- Assigns a single word of RAM to a software watchdog timer (SWDT) and writes the initial value N in the SWDT once at each execution of the main routine. The initial value N should satisfy the following condition:
 $N+1 \geq (\text{Counts of interrupt processing executed in each main routine})$

As the main routine execution cycle may change because of an interrupt processing or others, the initial value N should have a margin.

- Watches the operation of the interrupt processing routine by comparing the SWDT contents with counts of interrupt processing after the initial value N has been set.
- Detects that the interrupt processing routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
 If the SWDT contents do not change after interrupt processing.

<The interrupt processing routine>

- Decrements the SWDT contents by 1 at each interrupt processing.
- Determines that the main routine operates normally when the SWDT contents are reset to the initial value N at almost fixed cycles (at the fixed interrupt processing count).
- Detects that the main routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
 If the SWDT contents are not initialized to the initial value N but continued to decrement and if they reach 0 or less.

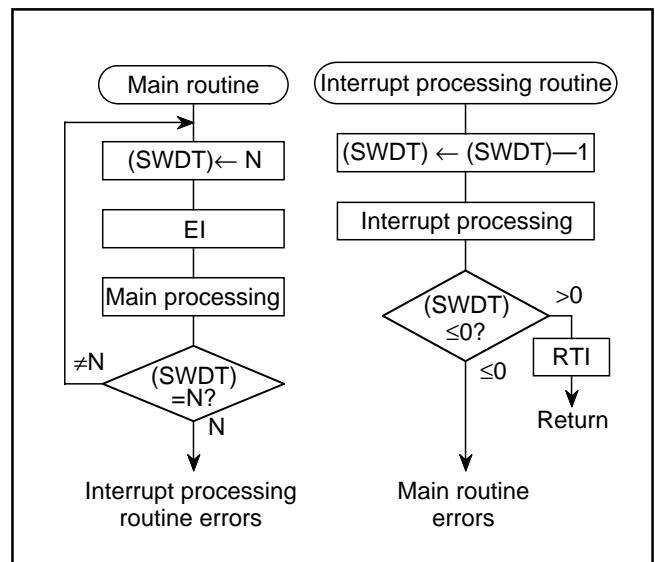


Fig. 61 Watchdog timer by software

LIST OF PRECAUTIONS

① Noise and latch-up prevention

Connect a capacitor on the following condition to prevent noise and latch-up;

- connect a bypass capacitor (approx. 0.1 μ F) between pins VDD and VSS at the shortest distance,
- equalize its wiring in width and length, and
- use relatively thick wire.

In the One Time PROM version, CNVSS pin is also used as VPP pin. Accordingly, when using this pin, connect this pin to Vss through a resistor about 5 k Ω (connect this resistor to CNVSS/VPP pin as close as possible).

In addition, the MCU may be replaced with mask ROM version without the need to remove the resistor from the circuit and without any adverse effect on operation.

② Register initial values 1

The initial value of the following registers are undefined after system is released from reset. After system is released from reset, set initial values.

- Register Z (2 bits)
- Register D (3 bits)
- Register E (8 bits)

③ Register initial values 2

The initial value of the following registers are undefined at RAM back-up. After system is returned from RAM back-up, set initial values.

- Register Z (2 bits)
- Register X (4 bits)
- Register Y (4 bits)
- Register D (3 bits)
- Register E (8 bits)

④ Stack registers (SKs)

Stack registers (SKs) are eight identical registers, so that subroutines can be nested up to 8 levels. However, one of stack registers is used respectively when using an interrupt service routine and when executing a table reference instruction. Accordingly, be careful not to over the stack when performing these operations together.

⑤ Prescaler

Stop counting and then execute the TABPS instruction to read from prescaler data.

Stop counting and then execute the TPSAB instruction to set prescaler data.

⑥ Timer count source

Stop timer 1, 2 and LC counting to change its count source.

⑦ Reading the count value

Stop timer 1 or 2 counting and then execute the data read instruction (TAB1, TAB2) to read its data.

⑧ Writing to the timer

Stop timer 1, 2 or LC counting and then execute the data write instruction (T1AB, T2AB, TLCA) to write its data.

⑨ Writing to reload register R1, R2H

When writing data to reload register R1, reload register R2H while timer 1 or timer 2 is operating, avoid a timing when timer 1 or timer 2 underflows.

⑩ Timer 2

Avoid a timing when timer 2 underflows to stop timer 2 at PWM output function used.

When "H" interval extension function of the PWM signal is set to be "valid", set "1" or more to reload register R2H.

⑪ Timer 3

Stop timer 3 counting to change its count source.

⑫ Timer input/output pin

Set the port C output latch to "0" to output the PWM signal from C/CNTR pin.

⑬ Prescaler and Timer 1 count start timing and count time when operation starts

Count starts from the first rising edge of the count source (2) after Prescaler and Timer 1 operations start (1). Time to first underflow (3) is shorter (for up to 1 period of the count source) than time among next underflow (4) by the timing to start the timer and count source operations after count starts. When selecting CNTR input as the count source of Timer 1, Timer 1 operates synchronizing with the falling edge of CNTR input.

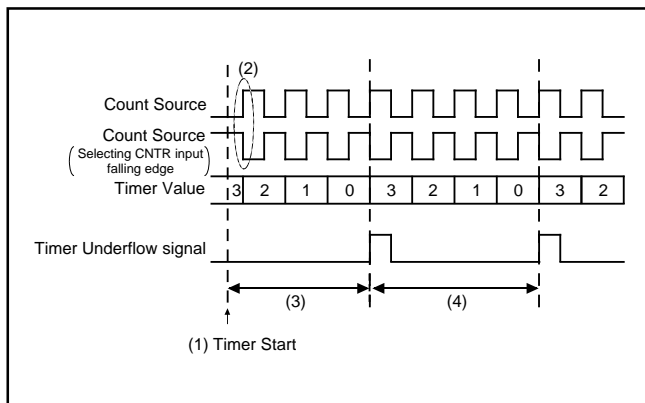


Fig. 62 Timer count start timing and count time when operation starts (Prescaler and Timer 1)

⑭ Timer 2 and Timer LC count start timing and count time when operation starts

Count starts from the rising edge (2) after the first falling edge of the count source, after Timer 2 and Timer LC operations start (1). Time to first underflow (3) is different from time among next underflow (4) by the timing to start the timer and count source operations after count starts.

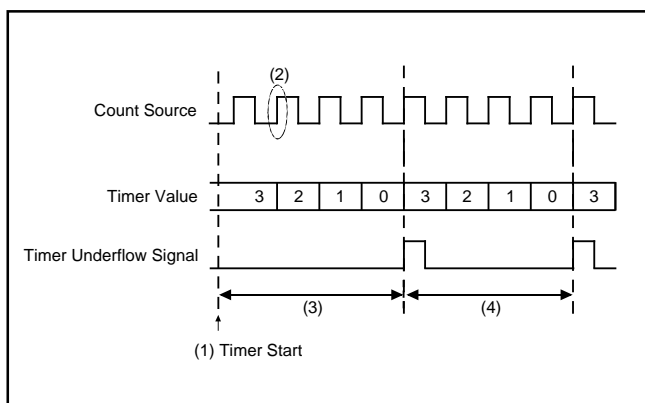


Fig. 63 Timer count start timing and count time when operation starts (Timer 2 and Timer LC)

⑮ Watchdog timer

- The watchdog timer function is valid after system is released from reset. When not using the watchdog timer function, execute the DWDT instruction and the WRST instruction continuously, and clear the WEF flag to "0" to stop the watchdog timer function.
- The watchdog timer function is valid after system is returned from the power down state. When not using the watchdog timer function, execute the DWDT instruction and the WRST instruction continuously every system is returned from the power down state, and stop the watchdog timer function.
- When the watchdog timer function and power down function are used at the same time, execute the WRST instruction before system enters into the power down state and initialize the flag WDF1.

⑯ Multifunction

- Be careful that the output of port D5 can be used even when INT pin is selected. The threshold value is different between port D5 and INT. Accordingly, be careful when the input of both is used.
- Be careful that the "H" output of port C can be used even when output of CNTR pin are selected.

⑰ Program counter

Make sure that the PCH does not specify after the last page of the built-in ROM.

⑩ D5/INT pin

① Note [1] on bit 3 of register I1

When the input of the INT pin is controlled with the bit 3 of register I1 in software, be careful about the following notes.

- Depending on the input state of the D5/INT pin, the external 0 interrupt request flag (EXF0) may be set when the bit 3 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 64①) and then, change the bit 3 of register I1. In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 64②). Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 64③).

```

    ⋮
    LA   4   ; (XXX02)
    TV1A ; The SNZ0 instruction is valid ..... ①
    LA   8   ; (1XXX2)
    TI1A ; Control of INT pin input is changed
    NOP   ..... ②
    SNZ0 ; The SNZ0 instruction is executed
          (EXF0 flag cleared)
    NOP   ..... ③
    ⋮
    X : these bits are not used here.
  
```

Fig. 64 External 0 interrupt program example-1

② Note on bit 2 of register I1

When the interrupt valid waveform of the D5/INT pin is changed with the bit 2 of register I1 in software, be careful about the following notes.

- Depending on the input state of the D5/INT pin, the external 0 interrupt request flag (EXF0) may be set when the bit 2 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 66①) and then, change the bit 2 of register I1. In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 66②). Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 66③).

```

    ⋮
    LA   4   ; (XXX02)
    TV1A ; The SNZ0 instruction is valid ..... ①
    LA  12   ; (11XXX2)
    TI1A ; Interrupt valid waveform is changed
    NOP   ..... ②
    SNZ0 ; The SNZ0 instruction is executed
          (EXF0 flag cleared)
    NOP   ..... ③
    ⋮
    X : these bits are not used here.
  
```

Fig. 66 External 0 interrupt program example-3

③ Note [2] on bit 3 of register I1

When the bit 3 of register I1 is cleared to "0", the RAM back-up mode is selected and the input of INT pin is disabled, be careful about the following notes.

- When the key-on wakeup function of INT pin is not used (register K20 = "0"), clear bits 2 and 3 of register I1 before system enters to the power down mode. (refer to Figure 65①).

```

    ⋮
    LA   0   ; (00XX2)
    TI1A ; Input of INT disabled ..... ①
    DI
    EPOF
    POF2 ; Power down mode
    ⋮
    X : these bits are not used here.
  
```

Fig. 65 External 0 interrupt program example-2

⑬ POF and POF2 instructions

When the POF or POF2 instruction is executed continuously after the EPOF instruction, system enters the power down state.

Note that system cannot enter the power down state when executing only the POF or POF2 instruction.

Be sure to disable interrupts by executing the DI instruction before executing the EPOF instruction and the POF or POF2 instruction continuously.

⑭ Power-on reset

When the built-in power-on reset circuit is used, set the time for the supply voltage to rise from 0 V to the minimum voltage of recommended operating conditions to 100 μ s or less.

If the rising time exceeds 100 μ s, connect a capacitor between the RESET pin and V_{SS} at the shortest distance, and input "L" level to RESET pin until the value of supply voltage reaches the minimum operating voltage.

⑮ Voltage drop detection circuit (only in H version)

The voltage drop detection circuit detection voltage of this product is set up lower than the minimum value of the supply voltage of the recommended operating conditions.

When the supply voltage of a microcomputer falls below to the minimum value of recommended operating conditions and re-goes up (ex. battery exchange of an application product), depending on the capacity value of the bypass capacitor added to the power supply pin, the following case may cause program failure (Figure 67);

supply voltage does not fall below to $VRST^+$, and its voltage re-goes up with no reset.

In such a case, please design a system which supply voltage is once reduced below to $VRST^-$ and re-goes up after that.

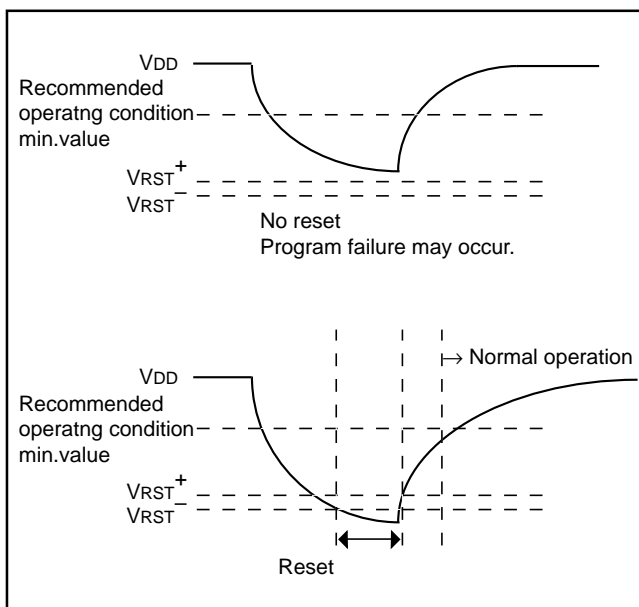


Fig. 67 VDD and $VRST^-$

⑯ Clock control

Execute the CRCK instruction in the initial setting routine of program (executing it in address 0 in page 0 is recommended).

The oscillation circuit by the CRCK instruction can be selected only once.

⑰ On-chip oscillator

The clock frequency of the on-chip oscillator depends on the supply voltage and the operation temperature range.

Be careful that variable frequencies when designing application products.

Also, the oscillation stabilize wait time after system is released from reset is generated by the on-chip oscillator clock. When considering the oscillation stabilize wait time after system is released from reset, be careful that the variable frequency of the on-chip oscillator clock.

⑱ External clock

When the external signal clock is used as the source oscillation ($f(XIN)$), note that the power down mode (POF and POF2 instructions) cannot be used.

⑳ Difference between Mask ROM version and One Time PROM version

Mask ROM version and One Time PROM version have some difference of the following characteristics within the limits of an electrical property by difference of a manufacture process, built-in ROM, and a layout pattern.

- a characteristic value
- a margin of operation
- the amount of noise-proof
- noise radiation, etc.,

Accordingly, be careful of them when switching.

㉑ Note on Power Source Voltage

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

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

CONTROL REGISTERS

| Interrupt control register V1 | | at reset : 0000 ₂ | at power down : 0000 ₂ | R/W TAV1/TV1A |
|-------------------------------|---------------------------------|------------------------------|--|------------------|
| V13 | Timer 2 interrupt enable bit | 0 | Interrupt disabled (SNZT2 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZT2 instruction is invalid) | |
| V12 | Timer 1 interrupt enable bit | 0 | Interrupt disabled (SNZT1 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZT1 instruction is invalid) | |
| V11 | Not used | 0 | This bit has no function, but read/write is enabled. | |
| | | 1 | | |
| V10 | External 0 interrupt enable bit | 0 | Interrupt disabled (SNZ0 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZ0 instruction is invalid) | |

| Interrupt control register V2 | | at reset : 0000 ₂ | at power down : 0000 ₂ | R/W TAV2/TV2A |
|-------------------------------|------------------------------|------------------------------|--|------------------|
| V23 | Not used | 0 | This bit has no function, but read/write is enabled. | |
| | | 1 | | |
| V22 | Not used | 0 | This bit has no function, but read/write is enabled. | |
| | | 1 | | |
| V21 | Not used | 0 | This bit has no function, but read/write is enabled. | |
| | | 1 | | |
| V20 | Timer 3 interrupt enable bit | 0 | Interrupt disabled (SNZT3 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZT3 instruction is invalid) | |

| Interrupt control register I1 | | at reset : 0000 ₂ | at power down : state retained | R/W TAI1/TI1A |
|-------------------------------|--|------------------------------|---|------------------|
| I13 | INT pin input control bit (Note 2) | 0 | INT pin input disabled | |
| | | 1 | INT pin input enabled | |
| I12 | Interrupt valid waveform for INT pin/ return level selection bit (Note 3) | 0 | Falling waveform/"L" level ("L" level is recognized with the SNZIO instruction) | |
| | | 1 | Rising waveform/"H" level ("H" level is recognized with the SNZIO instruction) | |
| I11 | INT pin edge detection circuit control bit | 0 | One-sided edge detected | |
| | | 1 | Both edges detected | |
| I10 | INT pin Timer 1 count start synchronous circuit selection bit | 0 | Timer 1 count start synchronous circuit not selected | |
| | | 1 | Timer 1 count start synchronous circuit selected | |

| Clock control register MR | | at reset : 1100 ₂ | at power down : state retained | R/W TAMR/ TMRA | |
|---------------------------|--------------------------------------|------------------------------|--------------------------------|------------------------|-----------------------------|
| MR3 | Operation mode selection bits | MR3 | Operation mode | | |
| | | 0 | | 0 | Through mode |
| | | 0 | | 1 | Frequency divided by 2 mode |
| | | 1 | | 0 | Frequency divided by 4 mode |
| MR2 | System clock selection bits (Note 3) | MR1 | System clock | | |
| | | 0 | | 0 | f(RING) |
| | | 0 | | 1 | f(XIN) |
| | | 1 | | 0 | f(XCIN) |
| | | 1 | 1 | Not available (Note 4) | |

Notes 1: "R" represents read enabled, and "W" represents write enabled.

2: When the contents of I12 and I13 are changed, the external interrupt request flag (EXF0) may be set.

3: The stopped clock cannot be selected for system clock.

4: "11" cannot be set to the low-order 2 bits (MR1, MR0) of register MR.

| Clock control register RG | | at reset : 0002 | | at power down : state retained | W TRGA |
|---------------------------|---|-----------------|---|--------------------------------|-----------|
| RG2 | Sub-clock (f(XCIN)) control bit (Note 2) | 0 | Sub-clock (f(XCIN)) oscillation available, ports D6 and D7 not selected | | |
| | | 1 | Sub-clock (f(XCIN)) oscillation stop, ports D6 and D7 selected | | |
| RG1 | Main-clock (f(XIN)) control bit (Note 2) | 0 | Main clock (f(XIN)) oscillation available | | |
| | | 1 | Main clock (f(XIN)) oscillation stop | | |
| RG0 | On-chip oscillator (f(RING)) control bit (Note 2) | 0 | On-chip oscillator (f(RING)) oscillation available | | |
| | | 1 | On-chip oscillator (f(RING)) oscillation stop | | |

| Timer control register PA | | at reset : 02 | | at power down : 02 | W TPAA |
|---------------------------|-----------------------|---------------|-----------------------|--------------------|-----------|
| PA0 | Prescaler control bit | 0 | Stop (state retained) | | |
| | | 1 | Operating | | |

| Timer control register W1 | | at reset : 00002 | | at power down : state retained | R/W TAW1/TW1A |
|---------------------------|--|------------------|--|----------------------------------|------------------|
| W13 | Timer 1 count auto-stop circuit selection bit (Note 3) | 0 | Timer 1 count auto-stop circuit not selected | | |
| | | 1 | Timer 1 count auto-stop circuit selected | | |
| W12 | Timer 1 control bit | 0 | Stop (state retained) | | |
| | | 1 | Operating | | |
| W11 | Timer 1 count source selection bits (Note 4) | W11 | W10 | Count source | |
| | | 0 | 0 | PWM signal (PWMOUT) | |
| W10 | | 0 | 1 | Prescaler output (ORCLK) | |
| | | 1 | 0 | Timer 3 underflow signal (T3UDF) | |
| | | 1 | 1 | CNTR input | |

| Timer control register W2 | | at reset : 00002 | | at power down : 00002 | R/W TAW2/TW2A |
|---------------------------|--|------------------|--|-----------------------|------------------|
| W23 | CNTR pin output control bit | 0 | CNTR pin output invalid | | |
| | | 1 | CNTR pin output valid | | |
| W22 | PWM signal interrupt valid waveform/return level selection bit | 0 | PWM signal "H" interval expansion function invalid | | |
| | | 1 | PWM signal "H" interval expansion function valid | | |
| W21 | Timer 2 control bit | 0 | Stop (state retained) | | |
| | | 1 | Operating | | |
| W20 | Timer 2 count source selection bit | 0 | XIN input | | |
| | | 1 | Prescaler output (ORCLK)/2 signal output | | |

| Timer control register W3 | | at reset : 00002 | | at power down : state retained | R/W TAW3/TW3A |
|---------------------------|---|------------------|--------------------------|-------------------------------------|------------------|
| W33 | Timer 3 count auto-stop circuit selection bit | 0 | XCIN input | | |
| | | 1 | Prescaler output (ORCLK) | | |
| W32 | Timer 3 control bit | 0 | Stop (Initial state) | | |
| | | 1 | Operating | | |
| W31 | Timer 3 count value selection bits | W31 | W30 | Count value | |
| | | 0 | 0 | Underflow occurs every 8192 counts | |
| W30 | | 0 | 1 | Underflow occurs every 16384 counts | |
| | | 1 | 0 | Underflow occurs every 32768 counts | |
| | | 1 | 1 | Underflow occurs every 65536 counts | |

Notes 1: "R" represents read enabled, and "W" represents write enabled.

2: The oscillation circuit selected for system clock cannot be stopped.

3: This function is valid only when the timer 1 count start synchronous circuit is selected (I10="1").

4: Port C output is invalid when CNTR input is selected for the timer 1 count source.

| Timer control register W4 | | at reset : 00002 | | at power down : state retained | R/W TAW4/TW4A |
|---------------------------|--|------------------|---|--------------------------------|------------------|
| W43 | Timer LC control bit | 0 | Stop (state retained) | | |
| | | 1 | Operating | | |
| W42 | Timer LC count source selection bit | 0 | Bit 4 (T34) of timer 3 | | |
| | | 1 | System clock (STCK) | | |
| W41 | CNTR output auto-control circuit selection bit | 0 | CNTR output auto-control circuit not selected | | |
| | | 1 | CNTR output auto-control circuit selected | | |
| W40 | CNTR pin input count edge selection bit | 0 | Falling edge | | |
| | | 1 | Rising edge | | |

| LCD control register L1 | | at reset : 00002 | | at power down : state retained | R/W TAL1/TL1A |
|-------------------------|--|------------------|----------------|--------------------------------|------------------|
| L13 | Internal dividing resistor for LCD power supply selection bit (Note 2) | 0 | 2r X 3, 2r X 2 | | |
| | | 1 | r X 3, r X 2 | | |
| L12 | LCD control bit | 0 | Stop | | |
| | | 1 | Operating | | |
| L11 | LCD duty and bias selection bits | L11 | L10 | Duty | Bias |
| | | 0 | 0 | Not available | |
| 0 | | 1 | 1/2 | 1/2 | |
| L10 | | 1 | 0 | 1/3 | 1/3 |
| | 1 | 1 | 1/4 | 1/3 | |

| LCD control register L2 | | at reset : 00002 | | at power down : state retained | W TL2A |
|-------------------------|---|------------------|------------------------------------|--------------------------------|-----------|
| L23 | SEG0/VLC3 pin function switch bit (Note 3) | 0 | SEG0 | | |
| | | 1 | VLC3 | | |
| L22 | SEG1/VLC2 pin function switch bit (Note 4) | 0 | SEG1 | | |
| | | 1 | VLC2 | | |
| L21 | SEG2/VLC1 pin function switch bit (Note 4) | 0 | SEG2 | | |
| | | 1 | VLC1 | | |
| L20 | Internal dividing resistor for LCD power supply control bit | 0 | Internal dividing resistor valid | | |
| | | 1 | Internal dividing resistor invalid | | |

| LCD control register L3 | | at reset : 11112 | | at power down : state retained | W TL3A |
|-------------------------|-----------------------------------|------------------|-------|--------------------------------|-----------|
| L33 | P23/SEG20 pin function switch bit | 0 | SEG20 | | |
| | | 1 | P23 | | |
| L32 | P22/SEG19 pin function switch bit | 0 | SEG19 | | |
| | | 1 | P22 | | |
| L31 | P21/SEG18 pin function switch bit | 0 | SEG18 | | |
| | | 1 | P21 | | |
| L30 | P20/SEG17 pin function switch bit | 0 | SEG17 | | |
| | | 1 | P20 | | |

Notes 1: "R" represents read enabled, and "W" represents write enabled.

2: "r (resistor) multiplied by 3" is used at 1/3 bias, and "r multiplied by 2" is used at 1/2 bias.

3: VLC3 is connected to VDD internally when SEG0 pin is selected.

4: Use internal dividing resistor when SEG1 and SEG2 pins are selected.

| LCD control register C1 | | at reset : 11112 | | at power down : state retained | W TC1A |
|-------------------------|-----------------------------------|------------------|-------|--------------------------------|-----------|
| C13 | P03/SEG24 pin function switch bit | 0 | SEG24 | | |
| | | 1 | P03 | | |
| C12 | P02/SEG23 pin function switch bit | 0 | SEG23 | | |
| | | 1 | P02 | | |
| C11 | P01/SEG22 pin function switch bit | 0 | SEG22 | | |
| | | 1 | P01 | | |
| C10 | P00/SEG21 pin function switch bit | 0 | SEG21 | | |
| | | 1 | P00 | | |

| LCD control register C2 | | at reset : 11112 | | at power down : state retained | W TC2A |
|-------------------------|-----------------------------------|------------------|-------|--------------------------------|-----------|
| C23 | P13/SEG28 pin function switch bit | 0 | SEG28 | | |
| | | 1 | P13 | | |
| C22 | P12/SEG27 pin function switch bit | 0 | SEG27 | | |
| | | 1 | P12 | | |
| C21 | P11/SEG26 pin function switch bit | 0 | SEG26 | | |
| | | 1 | P11 | | |
| C20 | P10/SEG25 pin function switch bit | 0 | SEG25 | | |
| | | 1 | P10 | | |

| Pull-up control register PU0 | | at reset : 00002 | | at power down : state retained | R/W TAPU0/ TPU0A |
|------------------------------|---|------------------|------------------------|--------------------------------|------------------------|
| PU03 | Port P03 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU02 | Port P02 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU01 | Port P01 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU00 | Port P00 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |

| Pull-up control register PU1 | | at reset : 00002 | | at power down : state retained | R/W TAPU1/ TPU1A |
|------------------------------|---|------------------|------------------------|--------------------------------|------------------------|
| PU13 | Port P13 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU12 | Port P12 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU11 | Port P11 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU10 | Port P10 pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |

Note: "W" represents write enabled.

| Port output structure control register FR0 | | at reset : 00002 | | at power down : state retained | W TFR0A |
|--|---|------------------|-----------------------------|--------------------------------|------------|
| FR03 | Ports P12, P13 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |
| FR02 | Ports P10, P11 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |
| FR01 | Ports P02, P03 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |
| FR00 | Ports P00, P01 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |

| Port output structure control register FR1 | | at reset : 00002 | | at power down : state retained | W TFR1A |
|--|--|------------------|-----------------------------|--------------------------------|------------|
| FR13 | Port D3 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |
| FR12 | Port D2 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |
| FR11 | Port D1 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |
| FR10 | Port D0 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |

| Port output structure control register FR2 | | at reset : 00002 | | at power down : state retained | W TFR2A |
|--|---|------------------|-----------------------------|--------------------------------|------------|
| FR23 | Ports P22, P23 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |
| FR22 | Ports P20, P21 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |
| FR21 | Port D5 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |
| FR20 | Port D4 output structure selection bit | 0 | N-channel open-drain output | | |
| | | 1 | CMOS output | | |

Note: "W" represents write enabled.

| Key-on wakeup control register K0 | | at reset : 00002 | | at power down : state retained | R/W TAK0/ TK0A |
|-----------------------------------|--|------------------|------------------------|--------------------------------|----------------------|
| K03 | Port P12, P13 key-on wakeup control bit (Note 3) | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used | | |
| K02 | Port P10, P11 key-on wakeup control bit (Note 2) | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used | | |
| K01 | Port P02, P03 key-on wakeup control bit | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used | | |
| K00 | Port P00, P01 key-on wakeup control bit | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used | | |

| Key-on wakeup control register K1 | | at reset : 00002 | | at power down : state retained | R/W TAK1/ TK1A |
|-----------------------------------|--|------------------|----------------------------|--------------------------------|----------------------|
| K13 | Ports P12, P13 return condition selection bit (Note 3) | 0 | Returned by edge | | |
| | | 1 | Returned by level | | |
| K12 | Ports P12, P13 valid waveform/level selection bit (Note 3) | 0 | Falling waveform/"L" level | | |
| | | 1 | Rising waveform/"H" level | | |
| K11 | Ports P10, P11 return condition selection bit (Note 2) | 0 | Returned by edge | | |
| | | 1 | Returned by level | | |
| K10 | Ports P10, P11 valid waveform/level selection bit (Note 2) | 0 | Falling waveform/"L" level | | |
| | | 1 | Rising waveform/"H" level | | |

| Key-on wakeup control register K2 | | at reset : 00002 | | at power down : state retained | R/W TAK2/ TK2A |
|-----------------------------------|--|------------------|--|--------------------------------|----------------------|
| K23 | Not used | 0 | This bit has no function, but read/write is enabled. | | |
| | | 1 | | | |
| K22 | Not used | 0 | This bit has no function, but read/write is enabled. | | |
| | | 1 | | | |
| K21 | INT pin return condition selection bit | 0 | Returned by level | | |
| | | 1 | Returned by edge | | |
| K20 | INT pin key-on wakeup control bit | 0 | Key-on wakeup invalid | | |
| | | 1 | Key-on wakeup valid | | |

Notes 1: "R" represents read enabled, and "W" represents write enabled.

2: To be invalid (K02 = "0") key-on wakeup of ports P10 and P11, set the registers K10 and K11 to "0".

3: To be invalid (K03 = "0") key-on wakeup of ports P12 and P13, set the registers K12 and K13 to "0".

INSTRUCTIONS

The 4552 Group has the 124 (123) instructions. Each instruction is described as follows;

- (1) Index list of instruction function
- (2) Machine instructions (index by alphabet)
- (3) Machine instructions (index by function)
- (4) Instruction code table

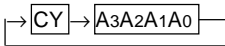
SYMBOL

The symbols shown below are used in the following list of instruction function and the machine instructions.

| Symbol | Contents | Symbol | Contents |
|--------|--|----------|---|
| A | Register A (4 bits) | PS | Prescaler |
| B | Register B (4 bits) | T1 | Timer 1 |
| DR | Register DR (3 bits) | T2 | Timer 2 |
| E | Register E (8 bits) | T3 | Timer 3 |
| V1 | Interrupt control register V1 (4 bits) | TLC | Timer LC |
| V2 | Interrupt control register V2 (4 bits) | T1F | Timer 1 interrupt request flag |
| I1 | Interrupt control register I1 (4 bits) | T2F | Timer 2 interrupt request flag |
| MR | Clock control register MR (4 bits) | T3F | Timer 3 interrupt request flag |
| RG | Clock control register RG (3 bits) | WDF1 | Watchdog timer flag |
| PA | Timer control register PA (1 bit) | WEF | Watchdog timer enable flag |
| W1 | Timer control register W1 (4 bits) | INTE | Interrupt enable flag |
| W2 | Timer control register W2 (4 bits) | EXF0 | External 0 interrupt request flag |
| W3 | Timer control register W3 (4 bits) | P | Power down flag |
| W4 | Timer control register W4 (4 bits) | | |
| L1 | LCD control register L1 (4 bits) | D | Port D (8 bits) |
| L2 | LCD control register L2 (4 bits) | P0 | Port P0 (4 bits) |
| L3 | LCD control register L3 (4 bits) | P1 | Port P1 (4 bits) |
| C1 | LCD control register C1 (4 bits) | P2 | Port P2 (4 bits) |
| C2 | LCD control register C2 (4 bits) | C | Port C (1 bit) |
| PU0 | Pull-up control register PU0 (4 bits) | | |
| PU1 | Pull-up control register PU1 (4 bits) | x | Hexadecimal variable |
| FR0 | Port output structure control register FR0 (4 bits) | y | Hexadecimal variable |
| FR1 | Port output structure control register FR1 (4 bits) | z | Hexadecimal variable |
| FR2 | Port output structure control register FR2 (4 bits) | p | Hexadecimal variable |
| K0 | Key-on wakeup control register K0 (4 bits) | n | Hexadecimal constant |
| K1 | Key-on wakeup control register K1 (4 bits) | i | Hexadecimal constant |
| K2 | Key-on wakeup control register K2 (4 bits) | j | Hexadecimal constant |
| X | Register X (4 bits) | A3A2A1A0 | Binary notation of hexadecimal variable A (same for others) |
| Y | Register Y (4 bits) | | |
| Z | Register Z (2 bits) | ← | Direction of data movement |
| DP | Data pointer (10 bits) (It consists of registers X, Y, and Z) | ↔ | Data exchange between a register and memory |
| PC | Program counter (14 bits) | ? | Decision of state shown before “?” |
| PCH | High-order 7 bits of program counter | () | Contents of registers and memories |
| PCL | Low-order 7 bits of program counter | — | Negate, Flag unchanged after executing instruction |
| SK | Stack register (14 bits X 8) | M(DP) | RAM address pointed by the data pointer |
| SP | Stack pointer (3 bits) | a | Label indicating address a6 a5 a4 a3 a2 a1 a0 |
| CY | Carry flag | p, a | Label indicating address a6 a5 a4 a3 a2 a1 a0 in page p6 p5 p4 p3 p2 p1 p0 |
| UPTF | High-order bit reference enable flag | C | Hex. C + Hex. number x |
| RPS | Prescaler reload register (8 bits) | + | |
| R1 | Timer 1 reload register (8 bits) | x | |
| R3 | Timer 3 reload register (8 bits) | | |
| R2L | Timer 2 reload register (8 bits) | | |
| R2H | Timer 2 reload register (8 bits) | | |
| RLC | Timer LC reload register (4 bits) | | |

Note : Some instructions of the 4552 Group has the skip function to unexecute the next described instruction. The 4552 Group just invalidates the next instruction when a skip is performed. The contents of program counter is not increased by 2. Accordingly, the number of cycles does not change even if skip is not performed. However, the cycle count becomes “1” if the TABP p, RT, or RTS instruction is skipped.

INDEX LIST OF INSTRUCTION FUNCTION

| Grouping | Mnemonic | Function | Grouping | Mnemonic | Function | |
|-------------------------------|---|--|--------------------------|---|--|---|
| Register to register transfer | TAB | $(A) \leftarrow (B)$ | RAM to register transfer | XAMI j | $(A) \leftarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15 $(Y) \leftarrow (Y) + 1$ | |
| | TBA | $(B) \leftarrow (A)$ | | TMA j | $(M(DP)) \leftarrow (A)$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15 | |
| | TAY | $(A) \leftarrow (Y)$ | | Arithmetic operation | LA n | $(A) \leftarrow n$ n = 0 to 15 |
| | TYA | $(Y) \leftarrow (A)$ | | | TABP p | $(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ (Note) $(PCL) \leftarrow (DR2-DR0, A3-A0)$ at (UPTF) = 0 $(B) \leftarrow (ROM(PC))_{7-4}$ $(A) \leftarrow (ROM(PC))_{3-0}$ at (UPTF) = 1 $(DR2) \leftarrow (0)$ $(DR1, DR0) \leftarrow (ROM(PC))_{9,8}$ $(B) \leftarrow (ROM(PC))_{7-4}$ $(A) \leftarrow (ROM(PC))_{3-0}$ $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$ |
| | TEAB | $(E7-E4) \leftarrow (B)$ $(E3-E0) \leftarrow (A)$ | | | AM | $(A) \leftarrow (A) + (M(DP))$ |
| | TABE | $(B) \leftarrow (E7-E4)$ $(A) \leftarrow (E3-E0)$ | | | AMC | $(A) \leftarrow (A) + (M(DP)) + (CY)$ $(CY) \leftarrow \text{Carry}$ |
| | TDA | $(DR2-DR0) \leftarrow (A2-A0)$ | | | A n | $(A) \leftarrow (A) + n$ n = 0 to 15 |
| | TAD | $(A2-A0) \leftarrow (DR2-DR0)$ $(A3) \leftarrow 0$ | | | AND | $(A) \leftarrow (A) AND (M(DP))$ |
| | TAZ | $(A1, A0) \leftarrow (Z1, Z0)$ $(A3, A2) \leftarrow 0$ | | | OR | $(A) \leftarrow (A) OR (M(DP))$ |
| | TAX | $(A) \leftarrow (X)$ | | | SC | $(CY) \leftarrow 1$ |
| TASP | $(A2-A0) \leftarrow (SP2-SP0)$ $(A3) \leftarrow 0$ | RC | $(CY) \leftarrow 0$ | | | |
| RAM addresses | LXY x, y | $(X) \leftarrow x$ x = 0 to 15 $(Y) \leftarrow y$ y = 0 to 15 | SZC | | $(CY) = 0 ?$ | |
| | LZ z | $(Z) \leftarrow z$ z = 0 to 3 | CMA | $(A) \leftarrow \overline{(A)}$ | | |
| | INY | $(Y) \leftarrow (Y) + 1$ | RAR |  | | |
| | DEY | $(Y) \leftarrow (Y) - 1$ | | | | |
| RAM to register transfer | TAM j | $(A) \leftarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15 | | | | |
| | XAM j | $(A) \leftarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15 | | | | |
| | XAMD j | $(A) \leftarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15 $(Y) \leftarrow (Y) - 1$ | | | | |

Note: p is 0 to 31 for M34552M4/M4H.

p is 0 to 63 for M34552M8/M8H/G8/G8H.

INDEX LIST OF INSTRUCTION FUNCTION (continued)

| Grouping | Mnemonic | Function | Grouping | Mnemonic | Function |
|----------------------|----------|---|---------------------|--|---|
| Bit operation | SB j | $(M_j(DP)) \leftarrow 1$ j = 0 to 3 | Interrupt operation | DI | $(INTE) \leftarrow 0$ |
| | RB j | $(M_j(DP)) \leftarrow 0$ j = 0 to 3 | | EI | $(INTE) \leftarrow 1$ |
| | SZB j | $(M_j(DP)) = 0 ?$ j = 0 to 3 | | SNZ0 | V10 = 0: (EXF0) = 1 ? (EXF0) \leftarrow 0 V10 = 1: SNZ0 = NOP |
| Comparison operation | SEAM | $(A) = (M(DP)) ?$ | | SNZI0 | I12 = 1 : (INT) = "H" ? I12 = 0 : (INT) = "L" ? |
| | SEA n | $(A) = n ?$ n = 0 to 15 | | TAV1 | $(A) \leftarrow (V1)$ |
| Branch operation | B a | $(PCL) \leftarrow a6-a0$ | | TV1A | $(V1) \leftarrow (A)$ |
| | BL p, a | $(PCH) \leftarrow p$ $(PCL) \leftarrow a6-a0$ | | TAV2 | $(A) \leftarrow (V2)$ |
| | BLA p | $(PCH) \leftarrow p$ $(PCL) \leftarrow (DR2-DR0, A3-A0)$ | | TV2A | $(V2) \leftarrow (A)$ |
| Subroutine operation | BM a | $(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow 2$ $(PCL) \leftarrow a6-a0$ | | TAI1 | $(A) \leftarrow (I1)$ |
| | BML p, a | $(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ $(PCL) \leftarrow a6-a0$ | | TI1A | $(I1) \leftarrow (A)$ |
| | BMLA p | $(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ $(PCL) \leftarrow (DR2-DR0, A3-A0)$ | Timer operation | TPAA | $(PA) \leftarrow (A)$ |
| Return operation | RTI | $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$ | | TAW1 | $(A) \leftarrow (W1)$ |
| | RT | $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$ | | TW1A | $(W1) \leftarrow (A)$ |
| | RTS | $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$ | | TAW2 | $(A) \leftarrow (W2)$ |
| | | | | TW2A | $(W2) \leftarrow (A)$ |
| | | | | TAW3 | $(A) \leftarrow (W3)$ |
| | | | | TW3A | $(W3) \leftarrow (A)$ |
| | | | | TAW4 | $(A) \leftarrow (W4)$ |
| | | | | TW4A | $(W4) \leftarrow (A)$ |
| | | | | TABPS | $(B) \leftarrow (TPS7-TPS4)$ $(A) \leftarrow (TPS3-TPS0)$ |
| | | | TPSAB | $(RPS7-RPS4) \leftarrow (B)$ $(TPS7-TPS4) \leftarrow (B)$ $(RPS3-RPS0) \leftarrow (A)$ $(TPS3-TPS0) \leftarrow (A)$ | |

Note: p is 0 to 31 for M34552M4/M4H.

p is 0 to 63 for M34552M8/M8H/G8/G8H.

INDEX LIST OF INSTRUCTION FUNCTION (continued)

| Grouping | Mnemonic | Function | Grouping | Mnemonic | Function |
|------------------------|---|--|------------------------|------------------------|------------------------------|
| Timer operation | TAB1 | (B) ← (T17–T14) (A) ← (T13–T10) | Input/Output operation | CLD | (D) ← 1 |
| | T1AB | (R17–R14) ← (B) (T17–T14) ← (B) (R13–R10) ← (A) (T13–T10) ← (A) | | RD | (D(Y)) ← 0 (Y) = 0 to 7 |
| | TAB2 | (B) ← (T27–T24) (A) ← (T23–T20) | | SD | (D(Y)) ← 1 (Y) = 0 to 7 |
| | T2AB | (R27–R24) ← (B) (T27–T24) ← (B) (R23–R20) ← (A) (T23–T20) ← (A) | | SZD | (D(Y)) = 0 ? (Y) = 0 to 5 |
| | T2HAB | (R2H7–R2H4) ← (B) (R2H3–R2H0) ← (A) | | RCP | (C) ← 0 |
| | TR1AB | (R17–R14) ← (B) (R13–R10) ← (A) | | SCP | (C) ← 1 |
| | T2R2L | (T27–T24) ← (R2L7–R2L4) (T23–T20) ← (R2L3–R2L0) | | TAPU0 | (A) ← (PU0) |
| | TLCA | (LC) ← (A) (RLC) ← (A) | | TPU0A | (PU0) ← (A) |
| | SNZT1 | V12 = 0: (T1F) = 1 ? (T1F) ← 0 V12 = 1: SNZT1 = NOP | | TAPU1 | (A) ← (PU1) |
| | SNZT2 | V13 = 0: (T2F) = 1 ? (T2F) ← 0 V13 = 1: SNZT2 = NOP | | TPU1A | (PU1) ← (A) |
| SNZT3 | V20 = 0: (T3F) = 1 ? (T3F) ← 0 V20 = 1: SNZT3 = NOP | TAK0 | (A) ← (K0) | | |
| Input/Output operation | IAP0 | (A) ← (P0) | TK0A | (K0) ← (A) | |
| | OP0A | (P0) ← (A) | TAK1 | (A) ← (K1) | |
| | IAP1 | (A) ← (P1) | TK1A | (K1) ← (A) | |
| | OP1A | (P1) ← (A) | TAK2 | (A) ← (K2) | |
| | IAP2 | (A) ← (P2) | TK2A | (K2) ← (A) | |
| | OP2A | (P2) ← (A) | TFR0A | (FR0) ← (A) | |
| Clock operation | | | TFR1A | (FR1) ← (A) | |
| | | | TFR2A | (FR2) ← (A) | |
| | | | CRCK | RC oscillator selected | |
| | | | TAMR | (A) ← (MR) | |
| | | TMRA | (MR) ← (A) | | |
| | | TRGA | (RG) ← (A) | | |

INDEX LIST OF INSTRUCTION FUNCTION (continued)

| Group- ing | Mnemonic | Function |
|-----------------|----------------|--|
| LCD operation | TAL1 | $(A) \leftarrow (L1)$ |
| | TL1A | $(L1) \leftarrow (A)$ |
| | TL2A | $(L2) \leftarrow (A)$ |
| | TL3A | $(L3) \leftarrow (A)$ |
| | TC1A | $(C1) \leftarrow (A)$ |
| | TC2A | $(C2) \leftarrow (A)$ |
| Other operation | NOP | $(PC) \leftarrow (PC) + 1$ |
| | POF | Transition to clock operating mode |
| | POF2 | Transition to RAM back-up mode |
| | EPOF | POF, POF2 instructions valid |
| | SNZP | $(P) = 1 ?$ |
| | DWDT | Stop of watchdog timer function enabled |
| | SRST | System reset |
| | WRST | $(WDF1) = 1 ?$ $(WDF1) \leftarrow 0$ |
| | RUPT | $(UPTF) \leftarrow 0$ |
| | SUPT | $(UPTF) \leftarrow 1$ |
| | SVDE (Note) | At power down mode, voltage drop detection circuit valid |

Note: The SVDE instruction can be used only for the H version.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET)

A n (Add n and accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|---------------------|--|----|-----------------|------------------|---------|----------------|---|---|---|---|------------------|----------------------|---|---|----|---|---|---|--------------|
| | 0 | 0 | 0 | 1 | 1 | 0 | n | n | n | n | 2 | 0 | 6 | n | 16 | 1 | 1 | – | Overflow = 0 |
| Operation: | $(A) \leftarrow (A) + n$ $n = 0 \text{ to } 15$ | | | | | | | | | | Grouping: | Arithmetic operation | | | | | | | |
| Description: | Adds the value n in the immediate field to register A, and stores a result in register A. The contents of carry flag CY remains unchanged. Skips the next instruction when there is no overflow as the result of operation. Executes the next instruction when there is overflow as the result of operation. | | | | | | | | | | | | | | | | | | |

AM (Add accumulator and Memory)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|---------------------|---|----|-----------------|------------------|---------|----------------|---|---|---|---|------------------|----------------------|---|---|----|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | A | 16 | 1 | 1 | – | – |
| Operation: | $(A) \leftarrow (A) + (M(DP))$ | | | | | | | | | | Grouping: | Arithmetic operation | | | | | | | |
| Description: | Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged. | | | | | | | | | | | | | | | | | | |

AMC (Add accumulator, Memory and Carry)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|---------------------|--|----|-----------------|------------------|---------|----------------|---|---|---|---|------------------|----------------------|---|---|----|---|---|-----|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 2 | 0 | 0 | B | 16 | 1 | 1 | 0/1 | – |
| Operation: | $(A) \leftarrow (A) + (M(DP)) + (CY)$ $(CY) \leftarrow \text{Carry}$ | | | | | | | | | | Grouping: | Arithmetic operation | | | | | | | |
| Description: | Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY. | | | | | | | | | | | | | | | | | | |

AND (logical AND between accumulator and memory)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|---------------------|--|----|-----------------|------------------|---------|----------------|---|---|---|---|------------------|----------------------|---|---|----|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 2 | 0 | 1 | 8 | 16 | 1 | 1 | – | – |
| Operation: | $(A) \leftarrow (A) \text{ AND } (M(DP))$ | | | | | | | | | | Grouping: | Arithmetic operation | | | | | | | |
| Description: | Takes the AND operation between the contents of register A and the contents of M(DP), and stores the result in register A. | | | | | | | | | | | | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

B a (Branch to address a)

| Instruction code | D9 | D0 | | | | | | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | |
|-------------------|------------------|----|---|----|----|----|----|----|----|----|---------------------|--|------------------|---------|----------------|---|---|---|---|
| | 0 | 1 | 1 | a6 | a5 | a4 | a3 | a2 | a1 | a0 | 2 | 1 | 8 +a | a | 16 | 1 | 1 | - | - |
| Operation: | (PCL) ← a6 to a0 | | | | | | | | | | Grouping: | Branch operation | | | | | | | |
| | | | | | | | | | | | Description: | Branch within a page : Branches to address a in the identical page. | | | | | | | |
| | | | | | | | | | | | Note: | Specify the branch address within the page including this instruction. | | | | | | | |

BL p, a (Branch Long to address a in page p)

| Instruction code | D9 | D0 | | | | | | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | |
|-------------------|-------------------------------|----|----|----|----|----|----|----|----|----|---------------------|---|------------------|---------|----------------|---|---|---|---|
| | 0 | 0 | 1 | 1 | 1 | p4 | p3 | p2 | p1 | p0 | 2 | 0 | E +p | p | 16 | 2 | 2 | - | - |
| | 1 | 0 | p5 | a6 | a5 | a4 | a3 | a2 | a1 | a0 | 2 | 2 | p +a | a | 16 | | | | |
| Operation: | (PCH) ← p (PCL) ← a6 to a0 | | | | | | | | | | Grouping: | Branch operation | | | | | | | |
| | | | | | | | | | | | Description: | Branch out of a page : Branches to address a in page p. | | | | | | | |
| | | | | | | | | | | | Note: | p is 0 to 31 for M34552M4/M4H and p is 0 to 63 for M34552M8/M8H/G8/G8H. | | | | | | | |

BLA p (Branch Long to address (D) + (A) in page p)

| Instruction code | D9 | D0 | | | | | | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | |
|-------------------|---------------------------------------|----|----|----|---|---|----|----|----|----|---------------------|---|------------------|---------|----------------|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 16 | 2 | 2 | - | - |
| | 1 | 0 | p5 | p4 | 0 | 0 | p3 | p2 | p1 | p0 | 2 | 2 | p | p | 16 | | | | |
| Operation: | (PCH) ← p (PCL) ← (DR2-DR0, A3-A0) | | | | | | | | | | Grouping: | Branch operation | | | | | | | |
| | | | | | | | | | | | Description: | Branch out of a page : Branches to address (DR2 DR1 DR0 A3 A2 A1 A0) ₂ specified by registers D and A in page p. | | | | | | | |
| | | | | | | | | | | | Note: | p is 0 to 31 for M34552M4/M4H and p is 0 to 63 for M34552M8/M8H/G8/G8H. | | | | | | | |

BM a (Branch and Mark to address a in page 2)

| Instruction code | D9 | D0 | | | | | | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | |
|-------------------|--|----|---|----|----|----|----|----|----|----|---------------------|---|------------------|---------|----------------|---|---|---|---|
| | 0 | 1 | 0 | a6 | a5 | a4 | a3 | a2 | a1 | a0 | 2 | 1 | a | a | 16 | 1 | 1 | - | - |
| Operation: | (SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← 2 (PCL) ← a6-a0 | | | | | | | | | | Grouping: | Subroutine call operation | | | | | | | |
| | | | | | | | | | | | Description: | Call the subroutine in page 2 : Calls the subroutine at address a in page 2. | | | | | | | |
| | | | | | | | | | | | Note: | Subroutine extending from page 2 to another page can also be called with the BM instruction when it starts on page 2. Be careful not to over the stack because the maximum level of subroutine nesting is 8. | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

BML p, a (Branch and Mark Long to address a in page p)

| Instruction code | D9 | D0 | | | | | | | | | | Number of words | | | Number of cycles | | | Flag CY | | | Skip condition | | |
|---------------------|---|----|----|----|----|----|----|----|----|----|---|-----------------|---|----|------------------|----|---|---------|---|---|----------------|---|--|
| | 0 | 0 | 1 | 1 | 0 | p4 | p3 | p2 | p1 | p0 | 2 | 0 | C | +p | p | 16 | 2 | 2 | - | - | - | - | |
| | 1 | 0 | p5 | a6 | a5 | a4 | a3 | a2 | a1 | a0 | 2 | 2 | p | +a | a | 16 | | | | | | | |
| Operation: | $(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ $(PCL) \leftarrow a6-a0$ | | | | | | | | | | | | | | | | | | | | | | |
| Grouping: | Subroutine call operation | | | | | | | | | | | | | | | | | | | | | | |
| Description: | Call the subroutine : Calls the subroutine at address a in page p. | | | | | | | | | | | | | | | | | | | | | | |
| Note: | p is 0 to 31 for M34552M4/M4H and p is 0 to 63 for M34552M8/M8H/G8/G8H. Be careful not to over the stack because the maximum level of subroutine nesting is 8. | | | | | | | | | | | | | | | | | | | | | | |

BMLA p (Branch and Mark Long to address (D) + (A) in page p)

| Instruction code | D9 | D0 | | | | | | | | | | Number of words | | | Number of cycles | | | Flag CY | | | Skip condition | | |
|---------------------|---|----|----|----|---|---|----|----|----|----|---|-----------------|---|---|------------------|---|---|---------|---|---|----------------|--|--|
| | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 0 | 16 | 2 | 2 | - | - | - | - | | |
| | 1 | 0 | p5 | p4 | 0 | 0 | p3 | p2 | p1 | p0 | 2 | 2 | p | p | 16 | | | | | | | | |
| Operation: | $(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ $(PCL) \leftarrow (DR2-DR0, A3-A0)$ | | | | | | | | | | | | | | | | | | | | | | |
| Grouping: | Subroutine call operation | | | | | | | | | | | | | | | | | | | | | | |
| Description: | Call the subroutine : Calls the subroutine at address (DR2 DR1 DR0 A3 A2 A1 A0) ₂ specified by registers D and A in page p. | | | | | | | | | | | | | | | | | | | | | | |
| Note: | p is 0 to 31 for M34552M4/M4H and p is 0 to 63 for M34552M8/M8H/G8/G8H. Be careful not to over the stack because the maximum level of subroutine nesting is 8. | | | | | | | | | | | | | | | | | | | | | | |

CLD (CLear port D)

| Instruction code | D9 | D0 | | | | | | | | | | Number of words | | | Number of cycles | | | Flag CY | | | Skip condition | | |
|---------------------|------------------------|----|---|---|---|---|---|---|---|---|---|-----------------|---|---|------------------|----|---|---------|---|---|----------------|---|--|
| | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 1 | 1 | 16 | 1 | 1 | - | - | - | - | |
| Operation: | $(D) \leftarrow 1$ | | | | | | | | | | | | | | | | | | | | | | |
| Grouping: | Input/Output operation | | | | | | | | | | | | | | | | | | | | | | |
| Description: | Sets (1) to port D. | | | | | | | | | | | | | | | | | | | | | | |

CMA (CoMplement of Accumulator)

| Instruction code | D9 | D0 | | | | | | | | | | Number of words | | | Number of cycles | | | Flag CY | | | Skip condition | | |
|---------------------|--|----|---|---|---|---|---|---|---|---|---|-----------------|---|---|------------------|---|---|---------|---|---|----------------|--|--|
| | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 2 | 0 | 1 | C | 16 | 1 | 1 | - | - | - | - | | |
| Operation: | $(A) \leftarrow \overline{(A)}$ | | | | | | | | | | | | | | | | | | | | | | |
| Grouping: | Arithmetic operation | | | | | | | | | | | | | | | | | | | | | | |
| Description: | Stores the one's complement for register A's contents in register A. | | | | | | | | | | | | | | | | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

CRCK (Clock select: Rc oscillation Clock)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|---------------------------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|---|---|----|---|---|---|---|
| | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 2 | 2 | 9 | B | 16 | 1 | 1 | – | – |
| Operation: | RC oscillation circuit selected | | | | | | | | | | | Grouping: | Clock control operation | | | | | | |
| | | | | | | | | | | | | Description: | Selects the RC oscillation circuit for main clock f(XIN). | | | | | | |

DEY (DEcrement register Y)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|--------------------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|---|---|----|---|---|---|----------|
| | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 2 | 0 | 1 | 7 | 16 | 1 | 1 | – | (Y) = 15 |
| Operation: | $(Y) \leftarrow (Y) - 1$ | | | | | | | | | | | Grouping: | RAM addresses | | | | | | |
| | | | | | | | | | | | | Description: | Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped. When the contents of register Y is not 15, the next instruction is executed. | | | | | | |

DI (Disable Interrupt)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|-----------------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|--|---|----|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 4 | 16 | 1 | 1 | – | – |
| Operation: | $(INTE) \leftarrow 0$ | | | | | | | | | | | Grouping: | Interrupt control operation | | | | | | |
| | | | | | | | | | | | | Description: | Clears (0) to interrupt enable flag INTE, and disables the interrupt. | | | | | | |
| | | | | | | | | | | | | Note: | Interrupt is disabled by executing the DI instruction after executing 1 machine cycle. | | | | | | |

DWDT (Disable WatchDog Timer)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|---|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|---|---|----|---|---|---|---|
| | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 2 | 2 | 9 | C | 16 | 1 | 1 | – | – |
| Operation: | Stop of watchdog timer function enabled | | | | | | | | | | | Grouping: | Other operation | | | | | | |
| | | | | | | | | | | | | Description: | Stops the watchdog timer function by the WRST instruction after executing the DWDT instruction. | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

EI (Enable Interrupt)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|---|---|----|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 0 | 0 | 5 | 16 | 1 | 1 | – | – |
| Operation: | (INTE) ← 1 | | | | | | | | | | | Grouping: | Interrupt control operation | | | | | | |
| | | | | | | | | | | | | Description: | Sets (1) to interrupt enable flag INTE, and enables the interrupt. | | | | | | |
| | | | | | | | | | | | | Note: | Interrupt is enabled by executing the EI instruction after executing 1 machine cycle. | | | | | | |

EPOF (Enable POF instruction)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|---|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|--|---|----|---|---|---|---|
| | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 2 | 0 | 5 | B | 16 | 1 | 1 | – | – |
| Operation: | POF instruction, POF2 instruction valid | | | | | | | | | | | Grouping: | Other operation | | | | | | |
| | | | | | | | | | | | | Description: | Makes the immediate after POF instruction or POF2 instruction valid by executing the EPOF instruction. | | | | | | |

IAP0 (Input Accumulator from port P0)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|---|---|----|---|---|---|---|
| | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 6 | 0 | 16 | 1 | 1 | – | – |
| Operation: | (A) ← (P0) | | | | | | | | | | | Grouping: | Input/Output operation | | | | | | |
| | | | | | | | | | | | | Description: | Transfers the input of port P0 to register A. | | | | | | |

IAP1 (Input Accumulator from port P1)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|---|---|----|---|---|---|---|
| | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 6 | 1 | 16 | 1 | 1 | – | – |
| Operation: | (A) ← (P1) | | | | | | | | | | | Grouping: | Input/Output operation | | | | | | |
| | | | | | | | | | | | | Description: | Transfers the input of port P1 to register A. | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

IAP2 (Input Accumulator from port P2)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|------------|-------------------|---------------------|---|---------|----------------|
| | 1 | 0 0 1 1 0 0 0 1 0 | 1 | 1 | – | – |
| | | | | | | |
| Operation: | (A) ← (P2) | | Grouping: | Input/Output operation | | |
| | | | Description: | Transfers the input of port P2 to register A. | | |

INY (INcrement register Y)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|---------------|-------------------|---------------------|---|---------|----------------|
| | 0 | 0 0 0 0 1 0 0 1 1 | 1 | 1 | – | (Y) = 0 |
| | | | | | | |
| Operation: | (Y) ← (Y) + 1 | | Grouping: | RAM addresses | | |
| | | | Description: | Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. When the contents of register Y is not 0, the next instruction is executed. | | |

LA n (Load n in Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|------------------------|-------------------|---------------------|--|---------|------------------------|
| | 0 | 0 0 1 1 1 n n n n | 1 | 1 | – | Continuous description |
| | | | | | | |
| Operation: | (A) ← n n = 0 to 15 | | Grouping: | Arithmetic operation | | |
| | | | Description: | Loads the value n in the immediate field to register A. When the LA instructions are continuously coded and executed, only the first LA instruction is executed and other LA instructions coded continuously are skipped. | | |

LXY x, y (Load register X and Y with x and y)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|--|---------------------------|---------------------|--|---------|------------------------|
| | 1 | 1 x3 x2 x1 x0 y3 y2 y1 y0 | 1 | 1 | – | Continuous description |
| | | | | | | |
| Operation: | (X) ← x x = 0 to 15 (Y) ← y y = 0 to 15 | | Grouping: | RAM addresses | | |
| | | | Description: | Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y. When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped. | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| LZ z (Load register Z with z) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | |
|--|----|----|---|---|---|------------------|---------|----------------|----|---|---|---|---|---|----|----|---|---|---|---|
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | z1 | z0 | 2 | 0 | 4 | 8 | +z | 16 | 1 | 1 | – | – |
| Operation: (Z) ← z z = 0 to 3 | | | | | Grouping: RAM addresses | | | | | Description: Loads the value z in the immediate field to register Z. | | | | | | | | | | |
| NOP (No OPeration) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | – | – |
| Operation: (PC) ← (PC) + 1 | | | | | Grouping: Other operation | | | | | Description: No operation; Adds 1 to program counter value, and others remain unchanged. | | | | | | | | | | |
| OP0A (Output port P0 from Accumulator) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | | |
| | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 16 | 1 | 1 | – | – | |
| Operation: (P0) ← (A) | | | | | Grouping: Input/Output operation | | | | | Description: Outputs the contents of register A to port P0. | | | | | | | | | | |
| OP1A (Output port P1 from Accumulator) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | | |
| | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 1 | 16 | 1 | 1 | – | – | |
| Operation: (P1) ← (A) | | | | | Grouping: Input/Output operation | | | | | Description: Outputs the contents of register A to port P1. | | | | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

OP2A (Output port P2 from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|------------|---------------------|---------------------|--|---------|----------------|
| | 1 | 0 0 0 0 1 0 0 0 1 0 | 1 | 1 | – | – |
| | | | | | | |
| Operation: | (P2) ← (A) | | Grouping: | Input/Output operation | | |
| | | | Description: | Outputs the contents of register A to port P2. | | |

OR (logical OR between accumulator and memory)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|----------------------|---------------------|---------------------|---|---------|----------------|
| | 0 | 0 0 0 0 0 1 1 0 0 1 | 1 | 1 | – | – |
| | | | | | | |
| Operation: | (A) ← (A) OR (M(DP)) | | Grouping: | Arithmetic operation | | |
| | | | Description: | Takes the OR operation between the contents of register A and the contents of M(DP), and stores the result in register A. | | |

POF (Power OFF)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|------------------------------------|---------------------|---------------------|---|---------|----------------|
| | 0 | 0 0 0 0 0 0 0 0 1 0 | 1 | 1 | – | – |
| | | | | | | |
| Operation: | Transition to clock operating mode | | Grouping: | Other operation | | |
| | | | Description: | Puts the system in clock operating mode by executing the POF2 instruction after executing the EPOF instruction. | | |
| | | | Note: | If the EPOF instruction is not executed before executing this instruction, this instruction is equivalent to the NOP instruction. | | |

POF2 (Power OFF2)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|--------------------------------|---------------------|---------------------|---|---------|----------------|
| | 0 | 0 0 0 0 0 0 1 0 0 0 | 1 | 1 | – | – |
| | | | | | | |
| Operation: | Transition to RAM back-up mode | | Grouping: | Other operation | | |
| | | | Description: | Puts the system in RAM back-up state by executing the POF2 instruction after executing the EPOF instruction. | | |
| | | | Note: | If the EPOF instruction is not executed before executing this instruction, this instruction is equivalent to the NOP instruction. | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| RAR (Rotate Accumulator Right) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
|--------------------------------|---|----|---|---|-----------------|------------------|---------|----------------|---|---|---|---|---|-----|---------------------|---|---|-----|---|
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 2 | 0 | 1 | D | 16 | 1 | 1 | 0/1 | – |
| Operation: | | | | | | | | | | | | | | | Grouping: | Arithmetic operation | | | |
| | | | | | | | | | | | | | | | Description: | Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right. | | | |
| RB j (Reset Bit) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | j | j | 2 | 0 | 4 | C+j | 16 | 1 | 1 | – | – |
| Operation: | $(M_j(DP)) \leftarrow 0$ $j = 0 \text{ to } 3$ | | | | | | | | | | | | | | Grouping: | Bit operation | | | |
| | | | | | | | | | | | | | | | Description: | Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP). | | | |
| RC (Reset Carry flag) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 0 | 0 | 6 | 16 | 1 | 1 | 0 | – |
| Operation: | $(CY) \leftarrow 0$ | | | | | | | | | | | | | | Grouping: | Arithmetic operation | | | |
| | | | | | | | | | | | | | | | Description: | Clears (0) to carry flag CY. | | | |
| RCP (Reset Port C) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | |
| | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 2 | 8 | C | 16 | 1 | 1 | 0 | – |
| Operation: | $(C) \leftarrow 0$ | | | | | | | | | | | | | | Grouping: | Input/Output operation | | | |
| | | | | | | | | | | | | | | | Description: | Clears (0) to carry flag CY. | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| RD (Reset port D specified by register Y) | | | | | | | | | | Number of words | Number of cycles | Flag CY | Skip condition |
|---|---|--|--|--|--|--|--|--|--|--|------------------|---------|---------------------|
| Instruction code | $\begin{array}{cccccccccc} \text{D}_9 & & & & & & & & & \text{D}_0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ & & & & & & & & & & \text{D}_2 & & & & & & & & & \text{D}_0 \\ & & & & & & & & & & 0 & 1 & 4 & & & & & & & \text{D}_2 \end{array}$ | | | | | | | | | 1 | 1 | – | – |
| Operation: | $(D(Y)) \leftarrow 0$ However, $(Y) = 0 \text{ to } 7$ | | | | | | | | | Grouping: Input/Output operation Description: Clears (0) to a bit of port D specified by register Y. | | | |
| RT (ReTurn from subroutine) | | | | | | | | | | Number of words | Number of cycles | Flag CY | Skip condition |
| Instruction code | $\begin{array}{cccccccccc} \text{D}_9 & & & & & & & & & \text{D}_0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ & & & & & & & & & & \text{D}_2 & & & & & & & & & \text{D}_0 \\ & & & & & & & & & & 0 & 4 & 4 & & & & & & & \text{D}_2 \end{array}$ | | | | | | | | | 1 | 2 | – | – |
| Operation: | $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$ | | | | | | | | | Grouping: Return operation Description: Returns from subroutine to the routine called the subroutine. | | | |
| RTI (ReTurn from Interrupt) | | | | | | | | | | Number of words | Number of cycles | Flag CY | Skip condition |
| Instruction code | $\begin{array}{cccccccccc} \text{D}_9 & & & & & & & & & \text{D}_0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ & & & & & & & & & & \text{D}_2 & & & & & & & & & \text{D}_0 \\ & & & & & & & & & & 0 & 4 & 6 & & & & & & & \text{D}_2 \end{array}$ | | | | | | | | | 1 | 1 | – | – |
| Operation: | $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$ | | | | | | | | | Grouping: Return operation Description: Returns from interrupt service routine to main routine. Returns each value of data pointer (X, Y, Z), carry flag, skip status, NOP mode status by the continuous description of the LA/LXY instruction, register A and register B to the states just before interrupt. | | | |
| RTS (ReTurn from subroutine and Skip) | | | | | | | | | | Number of words | Number of cycles | Flag CY | Skip condition |
| Instruction code | $\begin{array}{cccccccccc} \text{D}_9 & & & & & & & & & \text{D}_0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ & & & & & & & & & & \text{D}_2 & & & & & & & & & \text{D}_0 \\ & & & & & & & & & & 0 & 4 & 5 & & & & & & & \text{D}_2 \end{array}$ | | | | | | | | | 1 | 2 | – | Skip at uncondition |
| Operation: | $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$ | | | | | | | | | Grouping: Return operation Description: Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition. | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

RUPT (Reset UPTF flag)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
|-------------------|------------|----|-----------------|------------------|---------|----------------|---------------------|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 5 | 8 | 1 | 1 | – | – |
| Operation: | (UPTF) ← 0 | | | | | | Grouping: | Other operation | | | | | | | | | |
| | | | | | | | Description: | Clears (0) to the high-order bit reference enable flag. | | | | | | | | | |

SB j (Set Bit)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
|-------------------|----------------------------|----|-----------------|------------------|---------|----------------|---------------------|--|---|---|---|---|-----|---|---|---|---|
| | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | j | j | 0 | 5 | C+j | 1 | 1 | – | – |
| Operation: | (Mj(DP)) ← 1 j = 0 to 3 | | | | | | Grouping: | Bit operation | | | | | | | | | |
| | | | | | | | Description: | Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP). | | | | | | | | | |

SC (Set Carry flag)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
|-------------------|----------|----|-----------------|------------------|---------|----------------|---------------------|----------------------------|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 7 | 1 | 1 | 1 | – |
| Operation: | (CY) ← 1 | | | | | | Grouping: | Arithmetic operation | | | | | | | | | |
| | | | | | | | Description: | Sets (1) to carry flag CY. | | | | | | | | | |

SCP (Set Port C)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
|-------------------|---------|----|-----------------|------------------|---------|----------------|---------------------|------------------------|---|---|---|---|---|---|---|---|---|
| | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 2 | 8 | D | 1 | 1 | – | – |
| Operation: | (C) ← 1 | | | | | | Grouping: | Input/Output operation | | | | | | | | | |
| | | | | | | | Description: | Sets (1) to port C. | | | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

SD (Set port D specified by register Y)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|----------------------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|--|---|----|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 0 | 1 | 5 | 16 | 1 | 1 | – | – |
| Operation: | (D(Y)) ← 1 (Y) = 0 to 7 | | | | | | | | | | | Grouping: | Input/Output operation | | | | | | |
| | | | | | | | | | | | | Description: | Sets (1) to a bit of port D specified by register Y. | | | | | | |

SEA n (Skip Equal, Accumulator with immediate data n)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|--------------------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---|---|---|----|---------------------------------------|---|---|---|
| | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 2 | 0 | 2 | 5 | 16 | 2 | 2 | – | (A) = n n = 0 to 15 |
| | 0 | 0 | 0 | 1 | 1 | 1 | n | n | n | n | 2 | 0 | 7 | n | 16 | Grouping: Comparison operation | | | Description: Skips the next instruction when the contents of register A is equal to the value n in the immediate field. Executes the next instruction when the contents of register A is not equal to the value n in the immediate field. |
| Operation: | (A) = n ? n = 0 to 15 | | | | | | | | | | | | | | | | | | |

SEAM (Skip Equal, Accumulator with Memory)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|-----------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|---|---|----|---|---|---|---------------|
| | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 2 | 0 | 2 | 6 | 16 | 1 | 1 | – | (A) = (M(DP)) |
| Operation: | (A) = (M(DP)) ? | | | | | | | | | | | Grouping: | Comparison operation | | | | | | |
| | | | | | | | | | | | | Description: | Skips the next instruction when the contents of register A is equal to the contents of M(DP). Executes the next instruction when the contents of register A is not equal to the contents of M(DP). | | | | | | |

SNZ0 (Skip if Non Zero condition of external 0 interrupt request flag)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|--|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|--|---|----|---|---|---|---------------------|
| | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 2 | 0 | 3 | 8 | 16 | 1 | 1 | – | V10 = 0: (EXF0) = 1 |
| Operation: | V10 = 0: (EXF0) = 1 ? (EXF0) ← 0 V10 = 1: SNZ0 = NOP (V10 : bit 0 of the interrupt control register V1) | | | | | | | | | | | Grouping: | Interrupt operation | | | | | | |
| | | | | | | | | | | | | Description: | When V10 = 0 : Clears (0) to the EXF0 flag and skips the next instruction when external 0 interrupt request flag EXF0 is "1." When the EXF0 flag is "0," executes the next instruction. When V10 = 1 : This instruction is equivalent to the NOP instruction. | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

SNZIO (Skip if Non Zero condition of external 0 Interrupt input pin)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|--|---------------------|---------------------|--|---------|--|
| | 0 | 0 0 0 0 1 1 1 0 1 0 | 1 | 1 | – | I12 = 0 : (INT) = "L" I12 = 1 : (INT) = "H" |
| Operation: | I12 = 0 : (INT) = "L" ? I12 = 1 : (INT) = "H" ? (I12 : bit 2 of the interrupt control register I1) | | Grouping: | Interrupt operation | | |
| | | | Description: | When I12 = 0 : Skips the next instruction when the level of INT pin is "L." Executes the next instruction when the level of INT pin is "H." When I12 = 1 : Skips the next instruction when the level of INT pin is "H." Executes the next instruction when the level of INT pin is "L." | | |

SNZP (Skip if Non Zero condition of Power down flag)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|-----------|---------------------|---------------------|--|---------|----------------|
| | 0 | 0 0 0 0 0 0 0 0 1 1 | 1 | 1 | – | (P) = 1 |
| Operation: | (P) = 1 ? | | Grouping: | Other operation | | |
| | | | Description: | Skips the next instruction when the P flag is "1". After skipping, the P flag remains unchanged. Executes the next instruction when the P flag is "0." | | |

SNZT1 (Skip if Non Zero condition of Timer 1 interrupt request flag)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|---|---------------------|---------------------|--|---------|--------------------|
| | 1 | 0 1 0 0 0 0 0 0 0 0 | 1 | 1 | – | V12 = 0: (T1F) = 1 |
| Operation: | V12 = 0: (T1F) = 1 ? (T1F) ← 0 V12 = 1: SNZT1 = NOP (V12 = bit 2 of interrupt control register V1) | | Grouping: | Timer operation | | |
| | | | Description: | When V12 = 0 : Clears (0) to the T1F flag and skips the next instruction when timer 1 interrupt request flag T1F is "1." When the T1F flag is "0," executes the next instruction. When V12 = 1 : This instruction is equivalent to the NOP instruction. | | |

SNZT2 (Skip if Non Zero condition of Timer 2 interrupt request flag)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|---|---------------------|---------------------|--|---------|--------------------|
| | 1 | 0 1 0 0 0 0 0 0 0 1 | 1 | 1 | – | V13 = 0: (T2F) = 1 |
| Operation: | V13 = 0: (T2F) = 1 ? (T2F) ← 0 V13 = 1: SNZT2 = NOP (V13 = bit 3 of interrupt control register V1) | | Grouping: | Timer operation | | |
| | | | Description: | When V13 = 0 : Clears (0) to the T2F flag and skips the next instruction when timer 2 interrupt request flag T2F is "1." When the T2F flag is "0," executes the next instruction. When V13 = 1 : This instruction is equivalent to the NOP instruction. | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| SNZT3 (Skip if Non Zero condition of Timer 3 interrupt request flag) | | | | | | |
|--|---|---------------------|---------------------|--|---------|--------------------|
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
| | 1 | 0 1 0 0 0 0 0 0 1 0 | 2 | 2 8 2 | 16 | |
| | | | 1 | 1 | – | V20 = 0: (T3F) = 1 |
| Operation: | V20 = 0: (T3F) = 1 ? (T3F) ← 0 V20 = 1: SNZT3 = NOP (V20 = bit 0 of interrupt control register V2) | | Grouping: | Timer operation | | |
| | | | Description: | When V20 = 0 : Clears (0) to the T3F flag and skips the next instruction when timer 3 interrupt request flag T3F is "1." When the T3F flag is "0," executes the next instruction. When V20 = 1 : This instruction is equivalent to the NOP instruction. | | |
| SRST (System ReSeT) | | | | | | |
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
| | 0 | 0 0 0 0 0 0 0 0 0 1 | 2 | 0 0 1 | 16 | |
| | | | 1 | 1 | – | – |
| Operation: | System reset occurrence | | Grouping: | Other operation | | |
| | | | Description: | System reset occurs. | | |
| SUPT (Set UPTF flag) | | | | | | |
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
| | 0 | 0 0 1 0 1 1 0 0 0 1 | 2 | 0 5 9 | 16 | |
| | | | 1 | 1 | – | – |
| Operation: | (UPTF) ← 1 | | Grouping: | Other operation | | |
| | | | Description: | Sets (1) to high-order bit reference enable flag. | | |
| SVDE (Se Voltage Detector Enable flag) | | | | | | |
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
| | 1 | 0 1 0 0 1 0 0 1 1 | 2 | 2 9 3 | 16 | |
| | | | 1 | 1 | – | – |
| Operation: | Voltage drop detection circuit valid at powerdown mode. | | Grouping: | Other operation | | |
| | | | Description: | Voltage drop detection circuit is valid at powerdown mode (clock operating mode, RAM back-up mode) Note: This instruction can be used only for H version. | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| SZB j (Skip if Zero, Bit) | | | | | | | | | | | | | | | | | | | |
|--|--|----|-----------------|------------------|---------|----------------|---|---|---|---|---------------------|--|---|---|----|---|---|---|----------------------------|
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | j | j | 2 | 0 | 2 | j | 16 | 1 | 1 | – | (Mj(DP)) = 0 j = 0 to 3 |
| Operation: | (Mj(DP)) = 0 ? j = 0 to 3 | | | | | | | | | | Grouping: | Bit operation | | | | | | | |
| | | | | | | | | | | | Description: | Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is "0." Executes the next instruction when the contents of bit j of M(DP) is "1." | | | | | | | |
| SZC (Skip if Zero, Carry flag) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 2 | 0 | 2 | F | 16 | 1 | 1 | – | (CY) = 0 |
| Operation: | (CY) = 0 ? | | | | | | | | | | Grouping: | Arithmetic operation | | | | | | | |
| | | | | | | | | | | | Description: | Skips the next instruction when the contents of carry flag CY is "0." After skipping, the CY flag remains unchanged. Executes the next instruction when the contents of the CY flag is "1." | | | | | | | |
| SZD (Skip if Zero, port D specified by register Y) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 2 | 4 | 16 | 2 | 2 | – | (D(Y)) = 0 (Y) = 0 to 7 |
| | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 2 | 0 | 2 | B | 16 | | | | |
| Operation: | (D(Y)) = 0 ? (Y) = 0 to 7 | | | | | | | | | | Grouping: | Input/Output operation | | | | | | | |
| | | | | | | | | | | | Description: | Skips the next instruction when a bit of port D specified by register Y is "0." Executes the next instruction when the bit is "1." | | | | | | | |
| | | | | | | | | | | | Note: | (Y) = 0 to 5. Do not execute this instruction if values except above are set to register Y. | | | | | | | |
| T1AB (Transfer data to timer 1 and register R1 from Accumulator and register B) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
| | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 2 | 2 | 3 | 0 | 16 | 1 | 1 | – | – |
| Operation: | (T17–T14) ← (B) (R17–R14) ← (B) (T13–T10) ← (A) (R13–R10) ← (A) | | | | | | | | | | Grouping: | Timer operation | | | | | | | |
| | | | | | | | | | | | Description: | Transfers the contents of register B to the high-order 4 bits of timer 1 and timer 1 reload register R1. Transfers the contents of register A to the low-order 4 bits of timer 1 and timer 1 reload register R1. | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| T2AB (Transfer data to timer 2 and register R2L from Accumulator and register B) | | | | | | | | | | | | | | | | | | | |
|--|-------------------------|----|---|---|---|-----------------|------------------|---------|----------------|---|---------------------|---|---|---|--|---|---|---|---|
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 2 | 3 | 1 | 16 | 1 | 1 | – | – |
| Operation: | (R2L7–R2L4) ← (B) | | | | | | | | | | Grouping: | | | | Timer operation | | | | |
| | (T27–T24) ← (B) | | | | | | | | | | Description: | | | | Transfers the contents of register B to the high-order 4 bits of timer 2 and timer 2 reload register R2L. Transfers the contents of register A to the low-order 4 bits of timer 2 and timer 2 reload register R2L. | | | | |
| | (R2L3–R2L0) ← (A) | | | | | | | | | | | | | | | | | | |
| | (T23–T20) ← (A) | | | | | | | | | | | | | | | | | | |
| T2HAB (Transfer data to register R2H from Accumulator and register B) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 2 | 2 | 9 | 4 | 16 | 1 | 1 | – | – |
| Operation: | (R2H7–R2H4) ← (B) | | | | | | | | | | Grouping: | | | | Timer operation | | | | |
| | (R2H3–R2H0) ← (A) | | | | | | | | | | Description: | | | | Transfers the contents of register B to the high-order 4 bits of timer 2 and timer 2 reload register R2H. Transfers the contents of register A to the low-order 4 bits of timer 2 and timer 2 reload register R2H. | | | | |
| T2R2L (Transfer data to timer 2 from register R2L) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 2 | 9 | 5 | 16 | 1 | 1 | – | – |
| Operation: | (T27–T20) ← (R2L7–R2L0) | | | | | | | | | | Grouping: | | | | Timer operation | | | | |
| | | | | | | | | | | | Description: | | | | Transfers the contents of reload register R2L to timer 2. | | | | |
| TAB (Transfer data to Accumulator from register B) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 2 | 0 | 1 | E | 16 | 1 | 1 | – | – |
| Operation: | (A) ← (B) | | | | | | | | | | Grouping: | | | | Register to register transfer | | | | |
| | | | | | | | | | | | Description: | | | | Transfers the contents of register B to register A. | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| TAB1 (Transfer data to Accumulator and register B from timer 1) | | | | | | | | | | | | | | | | | | | | |
|--|--------------------------------|----|---|---|--|-----------------|------------------|----------------|--|----------------|---|---|---|----|----|----|---|---|---|---|
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 2 | 2 | 7 | 0 | 16 | 1 | 1 | – | – | | |
| Operation: | (B) ← (T17–T14) | | | | (A) ← (T13–T10) | | | | Grouping: Timer operation | | | | Description: Transfers the high-order 4 bits (T17–T14) of timer 1 to register B. Transfers the low-order 4 bits (T13–T10) of timer 1 to register A. | | | | | | | |
| TAB2 (Transfer data to Accumulator and register B from timer 2) | | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 2 | 7 | 1 | 16 | 1 | 1 | – | – | |
| Operation: | (B) ← (T27–T24) | | | | (A) ← (T23–T20) | | | | Grouping: Timer operation | | | | Description: Transfers the high-order 4 bits (T27–T24) of timer 2 to register B. Transfers the low-order 4 bits (T23–T20) of timer 2 to register A. | | | | | | | |
| TABE (Transfer data to Accumulator and register B from register E) | | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 2 | 0 | 2 | A | 16 | 1 | 1 | – | – | |
| Operation: | (B) ← (E7–E4) | | | | (A) ← (E3–E0) | | | | Grouping: Register to register transfer | | | | Description: Transfers the high-order 4 bits (E7–E4) of register E to register B, and low-order 4 bits of register E to register A. | | | | | | | |
| TABP p (Transfer data to Accumulator and register B from Program memory in page p) | | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| | 0 | 0 | 1 | 0 | p ₅ | p ₄ | p ₃ | p ₂ | p ₁ | p ₀ | 2 | 0 | 8 | +p | p | 16 | 1 | 3 | – | – |
| Operation: | (SP) ← (SP) + 1 | | | | (SK(SP)) ← (PC) | | | | (PCH) ← p (Note) | | | | (PCL) ← (DR2–DR0, A3–A0) | | | | Grouping: Arithmetic operation | | | |
| | at (UPTF) = 0 | | | | at (UPTF) = 1 | | | | (B) ← (ROM(PC)) _{7–4} | | | | (DR2) ← (0) | | | | Description: | | | |
| | (A) ← (ROM(PC)) _{3–0} | | | | (DR1, DR0) ← (ROM(PC)) _{9, 8} | | | | (B) ← (ROM(PC)) _{7–4} | | | | (A) ← (ROM(PC)) _{3–0} | | | | UPTF = 0: Transfers bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 9 to 0 are the ROM pattern in address (DR2 DR1 DR0 A3 A2 A1 A0) ₂ specified by registers A and D in page p. | | | |
| | (PC) ← (SK(SP)) | | | | (SP) ← (SP) – 1 | | | | | | | | | | | | UPTF = 1: Transfers bits 9, 8 to register D, bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 7 to 0 are the ROM pattern in address (DR2 DR1 DR0 A3 A2 A1 A0) ₂ specified by registers A and D in page p. | | | |
| | | | | | | | | | | | | | | | | | Note: p is 0 to 31 for M34552M4/M4H, and p is 0 to 63 for M34552M8/M8H/G8/G8H. When this instruction is executed, be careful not to over the stack because 1 stage of stack register is used. | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| TABPS (Transfer data to Accumulator and register B from PreScaler) | | | | | | | | | | | | | | | | | | | | |
|---|--|----|---|---|---|-----------------|------------------|---------|----------------|---|--|---|---|---|----|----|---|---|---|---|
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 2 | 2 | 7 | 5 | 16 | 1 | 1 | – | – | |
| Operation: | (B) ← (TPS7–TPS4) (A) ← (TPS3–TPS0) | | | | | | | | | | Grouping: Timer operation | | | | | | | | | |
| | | | | | | | | | | | Description: Transfers the high-order 4 bits (TPS7–TPS4) of prescaler to register B, and transfers the low-order 4 bits (TPS3–TPS0) of prescaler to register A. | | | | | | | | | |
| TAD (Transfer data to Accumulator from register D) | | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 5 | 1 | 16 | 1 | 1 | – | – |
| Operation: | (A2–A0) ← (DR2–DR0) (A3) ← 0 | | | | | | | | | | Grouping: Register to register transfer | | | | | | | | | |
| | | | | | | | | | | | Description: Transfers the contents of register D to the low-order 3 bits (A2–A0) of register A. | | | | | | | | | |
| | | | | | | | | | | | Note: When this instruction is executed, “0” is stored to the bit 3 (A3) of register A. | | | | | | | | | |
| TAI1 (Transfer data to Accumulator from register I1) | | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 2 | 2 | 5 | 3 | 16 | 1 | 1 | – | – | |
| Operation: | (A) ← (I1) | | | | | | | | | | Grouping: Interrupt operation | | | | | | | | | |
| | | | | | | | | | | | Description: Transfers the contents of interrupt control register I1 to register A. | | | | | | | | | |
| TAK0 (Transfer data to Accumulator from register K0) | | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 2 | 2 | 5 | 6 | 16 | 1 | 1 | – | – | |
| Operation: | (A) ← (K0) | | | | | | | | | | Grouping: Input/Output operation | | | | | | | | | |
| | | | | | | | | | | | Description: Transfers the contents of key-on wakeup control register K0 to register A. | | | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| TAK1 (Transfer data to Accumulator from register K1) | | | | | | | | | | | | | | | | | | | |
|---|---|----|---|---|---|-----------------|------------------|---------|----------------|---|---------------------|--|---|---|----|---|---|---|---|
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 2 | 2 | 5 | 9 | 16 | 1 | 1 | – | – |
| Operation: | $(A) \leftarrow (K1)$ | | | | | | | | | | Grouping: | Input/Output operation | | | | | | | |
| | | | | | | | | | | | Description: | Transfers the contents of key-on wakeup control register K1 to register A. | | | | | | | |
| TAK2 (Transfer data to Accumulator from register K2) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 2 | 2 | 5 | A | 16 | 1 | 1 | – | – |
| Operation: | $(A) \leftarrow (K2)$ | | | | | | | | | | Grouping: | Input/Output operation | | | | | | | |
| | | | | | | | | | | | Description: | Transfers the contents of key-on wakeup control register K2 to register A. | | | | | | | |
| TAL1 (Transfer data to Accumulator from register L1) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 2 | 2 | 4 | A | 16 | 1 | 1 | – | – |
| Operation: | $(A) \leftarrow (L1)$ | | | | | | | | | | Grouping: | LCD control operation | | | | | | | |
| | | | | | | | | | | | Description: | Transfers the contents of LCD control register L1 to register A. | | | | | | | |
| TAM j (Transfer data to Accumulator from Memory) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 1 | 0 | 1 | 1 | 0 | 0 | j | j | j | j | 2 | 2 | C | j | 16 | 1 | 1 | – | – |
| Operation: | $(A) \leftarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ $j = 0 \text{ to } 15$ | | | | | | | | | | Grouping: | RAM to register transfer | | | | | | | |
| | | | | | | | | | | | Description: | After transferring the contents of M(DP) to register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| TAMR (Transfer data to Accumulator from register MR) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | |
|---|---------------------------------|----|---|---|-----------------|------------------|---------|----------------|---|---|---|--|---|---|---|----|---|---|---|---|
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 2 | 2 | 5 | 2 | 16 | 1 | 1 | – | – | |
| Operation: | (A) ← (MR) | | | | | | | | | | | Grouping: Clock operation | | | | | | | | |
| | | | | | | | | | | | | | | | Description: Transfers the contents of clock control register MR to register A. | | | | | |
| TAPU0 (Transfer data to Accumulator from register PU0) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 5 | 7 | 16 | 1 | 1 | – | – |
| Operation: | (A) ← (PU0) | | | | | | | | | | | Grouping: Input/Output operation | | | | | | | | |
| | | | | | | | | | | | | | | | Description: Transfers the contents of pull-up control register PU0 to register A. | | | | | |
| TAPU1 (Transfer data to Accumulator from register PU1) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2 | 2 | 5 | E | 16 | 1 | 1 | – | – |
| Operation: | (A) ← (PU1) | | | | | | | | | | | Grouping: Input/Output operation | | | | | | | | |
| | | | | | | | | | | | | | | | Description: Transfers the contents of pull-up control register PU1 to register A. | | | | | |
| TASP (Transfer data to Accumulator from Stack Pointer) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 5 | 0 | 16 | 1 | 1 | – | – |
| Operation: | (A2–A0) ← (SP2–SP0) (A3) ← 0 | | | | | | | | | | | Grouping: Register to register transfer | | | | | | | | |
| | | | | | | | | | | | | | | | Description: Transfers the contents of stack pointer (SP) to the low-order 3 bits (A2–A0) of register A. | | | | | |
| | | | | | | | | | | | | | | | Note: After this instruction is executed, “0” is stored to the bit 3 (A3) of register A. | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| TAV1 (Transfer data to Accumulator from register V1) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
|---|----|----|---|---|--------------------------------------|------------------|---------|----------------|--|---|---|---|---|---|----|---|---|---|---|
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 2 | 0 | 5 | 4 | 16 | 1 | 1 | – | – |
| Operation: (A) ← (V1) | | | | | Grouping: Interrupt operation | | | | Description: Transfers the contents of interrupt control register V1 to register A. | | | | | | | | | | |
| TAV2 (Transfer data to Accumulator from register V2) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 0 | 5 | 5 | 16 | 1 | 1 | – | – |
| Operation: (A) ← (V2) | | | | | Grouping: Interrupt operation | | | | Description: Transfers the contents of interrupt control register V2 to register A. | | | | | | | | | | |
| TAW1 (Transfer data to Accumulator from register W1) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 2 | 2 | 4 | B | 16 | 1 | 1 | – | – |
| Operation: (A) ← (W1) | | | | | Grouping: Timer operation | | | | Description: Transfers the contents of timer control register W1 to register A. | | | | | | | | | | |
| TAW2 (Transfer data to Accumulator from register W2) | | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | | | | | | | | | | | | | | |
| | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 2 | 4 | C | 16 | 1 | 1 | – | – |
| Operation: (A) ← (W2) | | | | | Grouping: Timer operation | | | | Description: Transfers the contents of timer control register W2 to register A. | | | | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAW3 (Transfer data to Accumulator from register W3)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|------------|-------------------|--|------------------|---------|----------------|
| | 1 | 0 0 1 0 0 1 1 0 1 | 1 | 1 | – | – |
| | | 2 4 D | | | | |
| Operation: | (A) ← (W3) | | Grouping: Timer operation | | | |
| | | | Description: Transfers the contents of timer control register W3 to register A. | | | |

TAW4 (Transfer data to Accumulator from register W4)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|------------|-------------------|--|------------------|---------|----------------|
| | 1 | 0 0 1 0 0 1 1 1 0 | 1 | 1 | – | – |
| | | 2 4 E | | | | |
| Operation: | (A) ← (W4) | | Grouping: Timer operation | | | |
| | | | Description: Transfers the contents of timer control register W4 to register A. | | | |

TAX (Transfer data to Accumulator from register X)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|-----------|-------------------|---|------------------|---------|----------------|
| | 0 | 0 0 1 0 1 0 0 1 0 | 1 | 1 | – | – |
| | | 2 0 5 2 | | | | |
| Operation: | (A) ← (X) | | Grouping: Register to register transfer | | | |
| | | | Description: Transfers the contents of register X to register A. | | | |

TAY (Transfer data to Accumulator from register Y)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|-----------|-------------------|---|------------------|---------|----------------|
| | 0 | 0 0 0 0 1 1 1 1 1 | 1 | 1 | – | – |
| | | 2 0 1 F | | | | |
| Operation: | (A) ← (Y) | | Grouping: Register to register transfer | | | |
| | | | Description: Transfers the contents of register Y to register A. | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAZ (Transfer data to Accumulator from register Z)

| Instruction code | D9 | D0 | | | | | | | | Number of | | | Flag CY | Skip condition | | | |
|---------------------|--|----|---|---|---|---|---|---|---|-----------|---|---|---------|----------------|----|--|--|
| | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 5 | 3 | 16 | | |
| | | | | | | | | | | | | 1 | 1 | - | - | | |
| Operation: | (A1, A0) ← (Z1, Z0) | | | | | | | | | | | | | | | | |
| | (A3, A2) ← 0 | | | | | | | | | | | | | | | | |
| Grouping: | Register to register transfer | | | | | | | | | | | | | | | | |
| Description: | Transfers the contents of register Z to the low-order 2 bits (A1, A0) of register A. | | | | | | | | | | | | | | | | |
| Note: | After this instruction is executed, "0" is stored to the high-order 2 bits (A3, A2) of register A. | | | | | | | | | | | | | | | | |

TBA (Transfer data to register B from Accumulator)

| Instruction code | D9 | D0 | | | | | | | | Number of | | | Flag CY | Skip condition | | | |
|---------------------|---|----|---|---|---|---|---|---|---|-----------|---|---|---------|----------------|----|--|--|
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 2 | 0 | 0 | E | 16 | | |
| | | | | | | | | | | | | 1 | 1 | - | - | | |
| Operation: | (B) ← (A) | | | | | | | | | | | | | | | | |
| Grouping: | Register to register transfer | | | | | | | | | | | | | | | | |
| Description: | Transfers the contents of register A to register B. | | | | | | | | | | | | | | | | |

TC1A (Transfer data to register C1 from Accumulator)

| Instruction code | D9 | D0 | | | | | | | | Number of | | | Flag CY | Skip condition | | | |
|---------------------|--|----|---|---|---|---|---|---|---|-----------|---|---|---------|----------------|----|--|--|
| | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 2 | 2 | A | 8 | 16 | | |
| | | | | | | | | | | | | 1 | 1 | - | - | | |
| Operation: | (C1) ← (A) | | | | | | | | | | | | | | | | |
| Grouping: | LCD control operation | | | | | | | | | | | | | | | | |
| Description: | Transfers the contents of register A to the LCD control register C1. | | | | | | | | | | | | | | | | |

TC2A (Transfer data to register C2 from Accumulator)

| Instruction code | D9 | D0 | | | | | | | | Number of | | | Flag CY | Skip condition | | | |
|---------------------|--|----|---|---|---|---|---|---|---|-----------|---|---|---------|----------------|----|--|--|
| | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 2 | A | 9 | 16 | | |
| | | | | | | | | | | | | 1 | 1 | - | - | | |
| Operation: | (C2) ← (A) | | | | | | | | | | | | | | | | |
| Grouping: | LCD control operation | | | | | | | | | | | | | | | | |
| Description: | Transfers the contents of register A to the LCD control register C2. | | | | | | | | | | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| TDA (Transfer data to register D from Accumulator and register B) | | | | | | | | | | | | | | | | | | | |
|---|--|----|-----------------|------------------|---------|----------------|---|---|---|---|---------------------|--|---|---|----|---|---|---|---|
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 0 | 2 | 9 | 16 | 1 | 1 | – | – |
| Operation: | $(DR_2-DR_0) \leftarrow (A_2-A_0)$ | | | | | | | | | | Grouping: | Register to register transfer | | | | | | | |
| | | | | | | | | | | | Description: | Transfers the low-order 3 bits (A_2-A_0) of register A to register D. | | | | | | | |
| TEAB (Transfer data to register E from Accumulator and register B) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 2 | 0 | 1 | A | 16 | 1 | 1 | – | – |
| Operation: | $(E_7-E_4) \leftarrow (B)$ $(E_3-E_0) \leftarrow (A)$ | | | | | | | | | | Grouping: | Register to register transfer | | | | | | | |
| | | | | | | | | | | | Description: | Transfers the contents of register B to the high-order 4 bits (E_7-E_4) of register E, and the contents of register A to the low-order 4 bits (E_3-E_0) of register E. | | | | | | | |
| TFR0A (Transfer data to register FR0 from Accumulator) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
| | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 2 | 2 | 2 | 8 | 16 | 1 | 1 | – | – |
| Operation: | $(FR_0) \leftarrow (A)$ | | | | | | | | | | Grouping: | Input/Output operation | | | | | | | |
| | | | | | | | | | | | Description: | Transfers the contents of register A to the port output structure control register FR0. | | | | | | | |
| TFR1A (Transfer data to register FR1 from Accumulator) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
| | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 2 | 2 | 9 | 16 | 1 | 1 | – | – |
| Operation: | $(FR_1) \leftarrow (A)$ | | | | | | | | | | Grouping: | Input/Output operation | | | | | | | |
| | | | | | | | | | | | Description: | Transfers the contents of register A to the port output structure control register FR1. | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TFR2A (Transfer data to register FR2 from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | |
|---------------------|---|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---|---|----|---|---|---|---|
| | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 2 | 2 | A | 16 | 1 | 1 | – | – |
| Operation: | (FR2) ← (A) | | | | | | | | | | | | | | | | | |
| Grouping: | Input/Output operation | | | | | | | | | | | | | | | | | |
| Description: | Transfers the contents of register A to the port output structure control register FR2. | | | | | | | | | | | | | | | | | |

TI1A (Transfer data to register I1 from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|---------------------|--|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---|---|---|----|---|---|---|---|
| | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 2 | 2 | 1 | 7 | 16 | 1 | 1 | – | – |
| Operation: | (I1) ← (A) | | | | | | | | | | | | | | | | | | |
| Grouping: | Interrupt operation | | | | | | | | | | | | | | | | | | |
| Description: | Transfers the contents of register A to interrupt control register I1. | | | | | | | | | | | | | | | | | | |

TK0A (Transfer data to register K0 from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|---------------------|--|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---|---|---|----|---|---|---|---|
| | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 2 | 2 | 1 | B | 16 | 1 | 1 | – | – |
| Operation: | (K0) ← (A) | | | | | | | | | | | | | | | | | | |
| Grouping: | Input/Output operation | | | | | | | | | | | | | | | | | | |
| Description: | Transfers the contents of register A to key-on wakeup control register K0. | | | | | | | | | | | | | | | | | | |

TK1A (Transfer data to register K1 from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|---------------------|--|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---|---|---|----|---|---|---|---|
| | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 2 | 2 | 1 | 4 | 16 | 1 | 1 | – | – |
| Operation: | (K1) ← (A) | | | | | | | | | | | | | | | | | | |
| Grouping: | Input/Output operation | | | | | | | | | | | | | | | | | | |
| Description: | Transfers the contents of register A to key-on wakeup control register K1. | | | | | | | | | | | | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

| TLCA (Transfer data to register LC from Accumulator) | | | | | | | | | | | | | | | | | | | |
|--|--|----|---|---|---|-----------------|------------------|---------|----------------|---|---|---|---|---|----|---|---|---|---|
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 2 | 2 | 0 | D | 16 | 1 | 1 | – | – |
| Operation: | (LC) ← (A) (RLC) ← (A) | | | | | | | | | | Grouping: Timer operation Description: Transfers the contents of register A to timer LC and reload register RLC. | | | | | | | | |
| TMA j (Transfer data to Memory from Accumulator) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 1 | 0 | 1 | 0 | 1 | 1 | j | j | j | j | 2 | 2 | B | j | 16 | 1 | 1 | – | – |
| Operation: | (M(DP)) ← (A) (X) ← (X)EXOR(j) j = 0 to 15 | | | | | | | | | | Grouping: RAM to register transfer Description: After transferring the contents of register A to M(DP), an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. | | | | | | | | |
| TMRA (Transfer data to register MR from Accumulator) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 2 | 2 | 1 | 6 | 16 | 1 | 1 | – | – |
| Operation: | (MR) ← (A) | | | | | | | | | | Grouping: Other operation Description: Transfers the contents of register A to clock control register MR. | | | | | | | | |
| TPAA (Transfer data to register PA from Accumulator) | | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | |
| | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 2 | 2 | A | A | 16 | 1 | 1 | – | – |
| Operation: | (PA0) ← (A0) | | | | | | | | | | Grouping: Timer operation Description: Transfers the contents of lowermost bit (A0) register A to timer control register PA. | | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TPSAB (Transfer data to Pre-Scaler from Accumulator and register B)

| Instruction code | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | |
|-------------------|--|----|----|----|----|----|----|----|----|----|-----------------|------------------|-----------------|---------------------|--|
| | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 2 | 3 | 5 | 16 | |
| Operation: | (RPS7–RPS4) ← (B) (TPS7–TPS4) ← (B) (RPS3–RPS0) ← (A) (TPS3–TPS0) ← (A) | | | | | | | | | | | Grouping: | Timer operation | Description: | Transfers the contents of register B to the high-order 4 bits of prescaler and prescaler reload register RPS, and transfers the contents of register A to the low-order 4 bits of prescaler and prescaler reload register RPS. |

TPU0A (Transfer data to register PU0 from Accumulator)

| Instruction code | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | |
|-------------------|-------------|----|----|----|----|----|----|----|----|----|-----------------|------------------|------------------------|---------------------|---|
| | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 2 | 2 | D | 16 | |
| Operation: | (PU0) ← (A) | | | | | | | | | | | Grouping: | Input/Output operation | Description: | Transfers the contents of register A to pull-up control register PU0. |

TPU1A (Transfer data to register PU1 from Accumulator)

| Instruction code | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | |
|-------------------|-------------|----|----|----|----|----|----|----|----|----|-----------------|------------------|------------------------|---------------------|---|
| | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 2 | 2 | E | 16 | |
| Operation: | (PU1) ← (A) | | | | | | | | | | | Grouping: | Input/Output operation | Description: | Transfers the contents of register A to pull-up control register PU1. |

TR1AB (Transfer data to register R1 from Accumulator and register B)

| Instruction code | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | |
|-------------------|------------------------------------|----|----|----|----|----|----|----|----|----|-----------------|------------------|-----------------|---------------------|--|
| | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | F | 16 | |
| Operation: | (R17–R14) ← (B) (R13–R10) ← (A) | | | | | | | | | | | Grouping: | Timer operation | Description: | Transfers the contents of register B to the high-order 4 bits (R17–R14) of timer 1 reload register R1, and the contents of register A to the low-order 4 bits (R13–R10) of timer 1 reload register R1. |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TRGA (Transfer data to register RG from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|--|---|----|---|---|---|---|
| | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 2 | 0 | 9 | 16 | 1 | 1 | – | – |
| Operation: | (RG) ← (A) | | | | | | | | | | | Grouping: | Clock control operation | | | | | | |
| | | | | | | | | | | | | Description: | Transfers the contents of register A to register RG. | | | | | | |

TV1A (Transfer data to register V1 from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | | |
|-------------------|------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|--|---|---|----|---|---|---|---|
| | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 0 | 3 | F | 16 | 1 | 1 | – | – |
| Operation: | (V1) ← (A) | | | | | | | | | | | Grouping: | Interrupt operation | | | | | | | |
| | | | | | | | | | | | | Description: | Transfers the contents of register A to interrupt control register V1. | | | | | | | |

TV2A (Transfer data to register V2 from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | | |
|-------------------|------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|--|---|---|----|---|---|---|---|
| | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 2 | 0 | 3 | E | 16 | 1 | 1 | – | – |
| Operation: | (V2) ← (A) | | | | | | | | | | | Grouping: | Interrupt operation | | | | | | | |
| | | | | | | | | | | | | Description: | Transfers the contents of register A to interrupt control register V2. | | | | | | | |

TW1A (Transfer data to register W1 from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | | | | | |
|-------------------|------------|----|-----------------|------------------|---------|----------------|---|---|---|---|---|---------------------|--|---|----|---|---|---|---|
| | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 2 | 2 | 0 | E | 16 | 1 | 1 | – | – |
| Operation: | (W1) ← (A) | | | | | | | | | | | Grouping: | Timer operation | | | | | | |
| | | | | | | | | | | | | Description: | Transfers the contents of register A to timer control register W1. | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TW2A (Transfer data to register W2 from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|------------|---------------------|--|------------------|---------|----------------|
| | 1 | 0 0 0 0 0 0 1 1 1 1 | 1 | 1 | – | – |
| | | 2 0 F | | | | |
| Operation: | (W2) ← (A) | | Grouping: Timer operation | | | |
| | | | Description: Transfers the contents of register A to timer control register W2. | | | |

TW3A (Transfer data to register W3 from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|------------|---------------------|--|------------------|---------|----------------|
| | 1 | 0 0 0 0 0 1 0 0 0 0 | 1 | 1 | – | – |
| | | 2 1 0 | | | | |
| Operation: | (W3) ← (A) | | Grouping: Timer operation | | | |
| | | | Description: Transfers the contents of register A to timer control register W3. | | | |

TW4A (Transfer data to register W4 from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|------------|---------------------|--|------------------|---------|----------------|
| | 1 | 0 0 0 0 0 1 0 0 0 1 | 1 | 1 | – | – |
| | | 2 1 1 | | | | |
| Operation: | (W4) ← (A) | | Grouping: Timer operation | | | |
| | | | Description: Transfers the contents of register A to timer control register W4. | | | |

TYA (Transfer data to register Y from Accumulator)

| Instruction code | D9 | D0 | Number of words | Number of cycles | Flag CY | Skip condition |
|-------------------|-----------|---------------------|---|------------------|---------|----------------|
| | 0 | 0 0 0 0 0 0 1 1 0 0 | 1 | 1 | – | – |
| | | 0 0 C | | | | |
| Operation: | (Y) ← (A) | | Grouping: Register to register transfer | | | |
| | | | Description: Transfers the contents of register A to register Y. | | | |

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

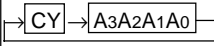
| WRST (Watchdog timer ReSeT) | | | | | | | | | | | | | | | | | | |
|---|---|----|---|---|------------------|--------------------------|------------------|---------|----------------|---------------------|--|---|---|----|---|---|---|------------|
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | |
| | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | A | 0 | 16 | 1 | 1 | – | (WDF1) = 1 |
| Operation: | (WDF1) = 1 ? (WDF1) ← 0 | | | | Grouping: | Other operation | | | | Description: | Clears (0) to the WDF1 flag and skips the next instruction when watchdog timer flag WDF1 is "1." When the WDF1 flag is "0," executes the next instruction. Also, stops the watchdog timer function when executing the WRST instruction immediately after the DWDT instruction. | | | | | | | |
| XAM j (eXchange Accumulator and Memory data) | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | |
| | 1 | 0 | 1 | 1 | 0 | 1 | j | j | j | j | 2 | D | j | 16 | 1 | 1 | – | – |
| Operation: | (A) ↔ (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 | | | | Grouping: | RAM to register transfer | | | | Description: | After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. | | | | | | | |
| XAMD j (eXchange Accumulator and Memory data and Decrement register Y and skip) | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | |
| | 1 | 0 | 1 | 1 | 1 | 1 | j | j | j | j | 2 | F | j | 16 | 1 | 1 | – | (Y) = 15 |
| Operation: | (A) ↔ (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) – 1 | | | | Grouping: | RAM to register transfer | | | | Description: | After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped. When the contents of register Y is not 15, the next instruction is executed. | | | | | | | |
| XAMI j (eXchange Accumulator and Memory data and Increment register Y and skip) | | | | | | | | | | | | | | | | | | |
| Instruction code | D9 | D0 | | | | Number of words | Number of cycles | Flag CY | Skip condition | | | | | | | | | |
| | 1 | 0 | 1 | 1 | 1 | 0 | j | j | j | j | 2 | E | j | 16 | 1 | 1 | – | (Y) = 0 |
| Operation: | (A) ↔ (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) + 1 | | | | Grouping: | RAM to register transfer | | | | Description: | After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. when the contents of register Y is not 0, the next instruction is executed. | | | | | | | |

MACHINE INSTRUCTIONS (INDEX BY TYPES)

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | | | | Number of words | Number of cycles | Function | | |
|-----------------------------------|----------|------------------|----|----|----|----|----|----|----|----|----|----------------------|-----------------|------------------|----------|---|-------------------------------------|
| | | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Hexadecimal notation | | | | | |
| Register to register transfer | TAB | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | E | 1 | 1 | (A) ← (B) |
| | TBA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | E | 1 | 1 | (B) ← (A) |
| | TAY | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | F | 1 | 1 | (A) ← (Y) |
| | TYA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | C | 1 | 1 | (Y) ← (A) |
| | TEAB | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | A | 1 | 1 | (E7–E4) ← (B) (E3–E0) ← (A) |
| | TABE | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 2 | A | 1 | 1 | (B) ← (E7–E4) (A) ← (E3–E0) |
| | TDA | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 2 | 9 | 1 | 1 | (DR2–DR0) ← (A2–A0) |
| | TAD | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 5 | 1 | 1 | 1 | (A2–A0) ← (DR2–DR0) (A3) ← 0 |
| | TAZ | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 5 | 3 | 1 | 1 | (A1, A0) ← (Z1, Z0) (A3, A2) ← 0 |
| | TAX | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 5 | 2 | 1 | 1 | (A) ← (X) |
| | TASP | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 1 | 1 | (A2–A0) ← (SP2–SP0) (A3) ← 0 |
| RAM addresses | LXY x, y | 1 | 1 | x3 | x2 | x1 | x0 | y3 | y2 | y1 | y0 | 3 | x y | 1 | 1 | (X) ← x x = 0 to 15 (Y) ← y y = 0 to 15 | |
| | LZ z | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | z1 | z0 | 0 | 4 8 +z | 1 | 1 | (Z) ← z z = 0 to 3 | |
| | INY | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 3 | 1 | 1 | (Y) ← (Y) + 1 | |
| | DEY | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 7 | 1 | 1 | (Y) ← (Y) – 1 | |
| RAM to register transfer | TAM j | 1 | 0 | 1 | 1 | 0 | 0 | j | j | j | j | 2 | C j | 1 | 1 | (A) ← (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 | |
| | XAM j | 1 | 0 | 1 | 1 | 0 | 1 | j | j | j | j | 2 | D j | 1 | 1 | (A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 | |
| | XAMD j | 1 | 0 | 1 | 1 | 1 | 1 | j | j | j | j | 2 | F j | 1 | 1 | (A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) – 1 | |
| | XAMI j | 1 | 0 | 1 | 1 | 1 | 0 | j | j | j | j | 2 | E j | 1 | 1 | (A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) + 1 | |
| | TMA j | 1 | 0 | 1 | 0 | 1 | 1 | j | j | j | j | 2 | B j | 1 | 1 | (M(DP)) ← (A) (X) ← (X)EXOR(j) j = 0 to 15 | |

| Skip condition | Carry flag CY | Detailed description |
|------------------------|---------------|--|
| - | - | Transfers the contents of register B to register A. |
| - | - | Transfers the contents of register A to register B. |
| - | - | Transfers the contents of register Y to register A. |
| - | - | Transfers the contents of register A to register Y. |
| - | - | Transfers the contents of register B to the high-order 4 bits (E7–E4) of register E, and the contents of register A to the low-order 4 bits (E3–E0) of register E. |
| - | - | Transfers the high-order 4 bits (E7–E4) of register E to register B, and low-order 4 bits (E3–E0) of register E to register A. |
| - | - | Transfers the contents of the low-order 3 bits (A2–A0) of register A to register D. |
| - | - | Transfers the contents of register D to the low-order 3 bits (A2–A0) of register A. |
| - | - | Transfers the contents of register Z to the low-order 2 bits (A1, A0) of register A. |
| - | - | Transfers the contents of register X to register A. |
| - | - | Transfers the contents of stack pointer (SP) to the low-order 3 bits (A2–A0) of register A. |
| Continuous description | - | Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y. When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped. |
| - | - | Loads the value z in the immediate field to register Z. |
| (Y) = 0 | - | Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. When the contents of register Y is not 0, the next instruction is executed. |
| (Y) = 15 | - | Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped. When the contents of register Y is not 15, the next instruction is executed. |
| - | - | After transferring the contents of M(DP) to register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. |
| - | - | After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. |
| (Y) = 15 | - | After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped. When the contents of register Y is not 15, the next instruction is executed. |
| (Y) = 0 | - | After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. When the contents of register Y is not 0, the next instruction is executed. |
| - | - | After transferring the contents of register A to M(DP), an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. |

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | | | Hexadecimal notation | Number of words | Number of cycles | Function |
|-----------------------------------|----------|------------------|----|----|----|----|----|----|----|----|-------|----------------------|-----------------|---|---|
| | | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | | | | |
| Arithmetic operation | LA n | 0 | 0 | 0 | 1 | 1 | 1 | n | n | n | n | 0 7 n | 1 | 1 | (A) ← n n = 0 to 15 |
| | TABP p | 0 | 0 | 1 | 0 | p5 | p4 | p3 | p2 | p1 | p0 | 0 8 p +p | 1 | 3 | (SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← p (Note) (PCL) ← (DR2–DR0, A3–A0) at (UPTF) = 0 (B) ← (ROM(PC))7–4 (A) ← (ROM(PC))3–0 at (UPTF) = 1 (DR2) ← (0) (DR1, DR0) ← (ROM(PC))9, 8 (B) ← (ROM(PC))7–4 (A) ← (ROM(PC))3–0 (PC) ← (SK(SP)) (SP) ← (SP) – 1 |
| | AM | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 0 A | 1 | 1 | (A) ← (A) + (M(DP)) |
| | AMC | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 0 B | 1 | 1 | (A) ← (A) + (M(DP)) + (CY) (CY) ← Carry |
| | A n | 0 | 0 | 0 | 1 | 1 | 0 | n | n | n | n | 0 6 n | 1 | 1 | (A) ← (A) + n n = 0 to 15 |
| | AND | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 1 8 | 1 | 1 | (A) ← (A) AND (M(DP)) |
| | OR | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 1 9 | 1 | 1 | (A) ← (A) OR (M(DP)) |
| | SC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 0 7 | 1 | 1 | (CY) ← 1 |
| | RC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 0 6 | 1 | 1 | (CY) ← 0 |
| | SZC | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 2 F | 1 | 1 | (CY) = 0 ? |
| CMA | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 1 C | 1 | 1 | (A) ← (A̅) | |
| RAR | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 1 D | 1 | 1 |  | |
| Bit operation | SB j | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | j | j | 0 5 C +j | 1 | 1 | (Mj(DP)) ← 1 j = 0 to 3 |
| | RB j | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | j | j | 0 4 C +j | 1 | 1 | (Mj(DP)) ← 0 j = 0 to 3 |
| | SZB j | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | j | j | 0 2 j | 1 | 1 | (Mj(DP)) = 0 ? j = 0 to 3 |
| Comparison operation | SEAM | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 2 6 | 1 | 1 | (A) = (M(DP)) ? |
| | SEA n | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 2 5 | 2 | 2 | (A) = n ? n = 0 to 15 |
| | | 0 | 0 | 0 | 1 | 1 | 1 | n | n | n | n | 0 7 n | | | |

Note: p is 0 to 31 for M3452M4/M4H. p is 0 to 63 for M3452M8/M8H/G8/G8H.

| Skip condition | Carry flag CY | Detailed description |
|----------------------------|---------------|---|
| Continuous description | – | Loads the value n in the immediate field to register A. When the LA instructions are continuously coded and executed, only the first LA instruction is executed and other LA instructions coded continuously are skipped. |
| – | – | UPTF = 0: Transfers bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 9 to 0 are the ROM pattern in address (DR2 DR1 DR0 A3 A2 A1 A0) ₂ specified by registers A and D in page p. When this instruction is executed, be careful not to over the stack because 1 stage of stack register is used. UPTF = 1: Transfers bits 9, 8 to register D, bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 7 to 0 are the ROM pattern in address (DR2 DR1 DR0 A3 A2 A1 A0) ₂ specified by registers A and D in page p. When this instruction is executed, be careful not to over the stack because 1 stage of stack register is used. |
| – | – | Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged. |
| – | 0/1 | Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY. |
| Overflow = 0 | – | Adds the value n in the immediate field to register A, and stores a result in register A. The contents of carry flag CY remains unchanged. Skips the next instruction when there is no overflow as the result of operation. Executes the next instruction when there is overflow as the result of operation. |
| – | – | Takes the AND operation between the contents of register A and the contents of M(DP), and stores the result in register A. |
| – | – | Takes the OR operation between the contents of register A and the contents of M(DP), and stores the result in register A. |
| – | 1 | Sets (1) to carry flag CY. |
| – | 0 | Clears (0) to carry flag CY. |
| (CY) = 0 | – | Skips the next instruction when the contents of carry flag CY is "0." |
| – | – | Stores the one's complement for register A's contents in register A. |
| – | 0/1 | Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right. |
| – | – | Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP). |
| – | – | Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP). |
| (Mj(DP)) = 0 j = 0 to 3 | – | Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is "0." Executes the next instruction when the contents of bit j of M(DP) is "1." |
| (A) = (M(DP)) | – | Skips the next instruction when the contents of register A is equal to the contents of M(DP). Executes the next instruction when the contents of register A is not equal to the contents of M(DP). |
| (A) = n | – | Skips the next instruction when the contents of register A is equal to the value n in the immediate field. Executes the next instruction when the contents of register A is not equal to the value n in the immediate field. |

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | | | | Hexadecimal notation | Number of words | Number of cycles | Function |
|-----------------------------------|----------|------------------|----|----|----|----|----|----|----|----|----------|-------------|----------------------|-----------------|--|----------|
| | | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | | | | | |
| Branch operation | B a | 0 | 1 | 1 | a6 | a5 | a4 | a3 | a2 | a1 | a0 | 1 8 a +a | 1 | 1 | (PCL) ← a6–a0 | |
| | BL p, a | 0 | 0 | 1 | 1 | 1 | p4 | p3 | p2 | p1 | p0 | 0 E p +p | 2 | 2 | (PCH) ← p (Note) (PCL) ← a6–a0 | |
| | | 1 | p6 | p5 | a6 | a5 | a4 | a3 | a2 | a1 | a0 | 2 p a +p +a | | | | |
| | BLA p | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 1 0 | 2 | 2 | (PCH) ← p (Note) (PCL) ← (DR2–DR0, A3–A0) | |
| 1 | | p6 | p5 | p4 | 0 | 0 | p3 | p2 | p1 | p0 | 2 p p +p | | | | | |
| Subroutine operation | BM a | 0 | 1 | 0 | a6 | a5 | a4 | a3 | a2 | a1 | a0 | 1 a a | 1 | 1 | (SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← 2 (PCL) ← a6–a0 | |
| | BML p, a | 0 | 0 | 1 | 1 | 0 | p4 | p3 | p2 | p1 | p0 | 0 C p +p | 2 | 2 | (SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← p (Note) (PCL) ← a6–a0 | |
| | | 1 | p6 | p5 | a6 | a5 | a4 | a3 | a2 | a1 | a0 | 2 p a +p +a | | | | |
| | BMLA p | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 3 0 | 2 | 2 | (SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← p (Note) (PCL) ← (DR2–DR0, A3–A0) | |
| 1 | | p6 | p5 | p4 | 0 | 0 | p3 | p2 | p1 | p0 | 2 p p +p | | | | | |
| Return operation | RTI | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 4 6 | 1 | 1 | (PC) ← (SK(SP)) (SP) ← (SP) – 1 | |
| | RT | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 4 4 | 1 | 2 | (PC) ← (SK(SP)) (SP) ← (SP) – 1 | |
| | RTS | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 4 5 | 1 | 2 | (PC) ← (SK(SP)) (SP) ← (SP) – 1 | |

Note: p is 0 to 31 for M34552M4/M4H.

p is 0 to 63 for M34552M8/M8H/G8/G8H.

| Skip condition | Carry flag CY | Detailed description |
|---------------------|---------------|---|
| - | - | Branch within a page : Branches to address a in the identical page. |
| - | - | Branch out of a page : Branches to address a in page p. |
| - | - | Branch out of a page : Branches to address (DR2 DR1 DR0 A3 A2 A1 A0) ₂ specified by registers D and A in page p. |
| - | - | Call the subroutine in page 2 : Calls the subroutine at address a in page 2. |
| - | - | Call the subroutine : Calls the subroutine at address a in page p. |
| - | - | Call the subroutine : Calls the subroutine at address (DR2 DR1 DR0 A3 A2 A1 A0) ₂ specified by registers D and A in page p. |
| - | - | Returns from interrupt service routine to main routine. Returns each value of data pointer (X, Y, Z), carry flag, skip status, NOP mode status by the continuous description of the LA/LXY instruction, register A and register B to the states just before interrupt. |
| - | - | Returns from subroutine to the routine called the subroutine. |
| Skip at uncondition | - | Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition. |

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | | | | Number of words | Number of cycles | Function |
|-----------------------------------|----------|------------------|----|----|----|----|----|----|----|----|-------|----------------------|-----------------|------------------|--|
| | | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Hexadecimal notation | | | |
| Interrupt operation | DI | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 0 4 | 1 | 1 | (INTE) ← 0 |
| | EI | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 0 5 | 1 | 1 | (INTE) ← 1 |
| | SNZ0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 3 8 | 1 | 1 | V10 = 0: (EXF0) = 1 ? (EXF0) ← 0 V10 = 1: SNZ0 = NOP |
| | SNZI0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 3 A | 1 | 1 | I12 = 1 : (INT) = "H" ? I12 = 0 : (INT) = "L" ? |
| | TAV1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 5 4 | 1 | 1 | (A) ← (V1) |
| | TV1A | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 3 F | 1 | 1 | (V1) ← (A) |
| | TAV2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 5 5 | 1 | 1 | (A) ← (V2) |
| | TV2A | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 3 E | 1 | 1 | (V2) ← (A) |
| | TAI1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 2 5 3 | 1 | 1 | (A) ← (I1) |
| TI1A | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 2 1 7 | 1 | 1 | (I1) ← (A) | |

Note: p is 0 to 31 for M34552M4/M4H.

p is 0 to 63 for M34552M8/M8H/G8/G8H.

| Skip condition | Carry flag CY | Detailed description |
|---------------------------------|---------------|--|
| - | - | Clears (0) to interrupt enable flag INTE, and disables the interrupt. |
| - | - | Sets (1) to interrupt enable flag INTE, and enables the interrupt. |
| V10 = 0: (EXF0) = 1 | - | When V10 = 0 : Clears (0) to the EXF0 flag and skips the next instruction when external 0 interrupt request flag EXF0 is "1." When the EXF0 flag is "0," executes the next instruction. When V10 = 1 : This instruction is equivalent to the NOP instruction. (V10: bit 0 of interrupt control register V1) |
| (INT) = "H" However, I12 = 1 | - | When I12 = 1 : Skips the next instruction when the level of INT pin is "H." (I12: bit 2 of interrupt control register I1) |
| (INT) = "L" However, I12 = 0 | - | When I12 = 0 : Skips the next instruction when the level of INT pin is "L." |
| - | - | Transfers the contents of interrupt control register V1 to register A. |
| - | - | Transfers the contents of register A to interrupt control register V1. |
| - | - | Transfers the contents of interrupt control register V2 to register A. |
| - | - | Transfers the contents of register A to interrupt control register V2. |
| - | - | Transfers the contents of interrupt control register I1 to register A. |
| - | - | Transfers the contents of register A to interrupt control register I1. |

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | | | | Hexadecimal notation | Number of words | Number of cycles | Function |
|-----------------------------------|----------|------------------|----|----|----|----|----|----|----|----|-------|-------|----------------------|---|--|----------|
| | | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | | | | | |
| Timer operation | TPAA | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 2 A A | 1 | 1 | (PA) ← (A) | |
| | TAW1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 2 4 B | 1 | 1 | (A) ← (W1) | |
| | TW1A | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 2 0 E | 1 | 1 | (W1) ← (A) | |
| | TAW2 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 2 4 C | 1 | 1 | (A) ← (W2) | |
| | TW2A | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 0 F | 1 | 1 | (W2) ← (A) | |
| | TAW3 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 2 4 D | 1 | 1 | (A) ← (W3) | |
| | TW3A | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 1 0 | 1 | 1 | (W3) ← (A) | |
| | TAW4 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 2 4 E | 1 | 1 | (A) ← (W4) | |
| | TW4A | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 2 1 1 | 1 | 1 | (W4) ← (A) | |
| | TABPS | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 2 7 5 | 1 | 1 | (B) ← (TPS7–TPS4) (A) ← (TPS3–TPS0) | |
| | TPSAB | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 2 3 5 | 1 | 1 | (RPS7–RPS4) ← (B) (TPS7–TPS4) ← (B) (RPS3–RPS0) ← (A) (TPS3–TPS0) ← (A) | |
| | TAB1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 2 7 0 | 1 | 1 | (B) ← (T17–T14) (A) ← (T13–T10) | |
| | T1AB | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 2 3 0 | 1 | 1 | (R17–R14) ← (B) (T17–T14) ← (B) (R13–R10) ← (A) (T13–T10) ← (A) | |
| | TAB2 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 2 7 1 | 1 | 1 | (B) ← (T27–T24) (A) ← (T23–T20) | |
| | T2AB | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 3 1 | 1 | 1 | (R2L7–R2L4) ← (B) (T27–T24) ← (B) (R2L3–R2L0) ← (A) (T23–T20) ← (A) | |
| | T2HAB | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 2 9 4 | 1 | 1 | (R2H7–R2H4) ← (B) (R2H3–R2H0) ← (A) | |
| | TR1AB | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 3 F | 1 | 1 | (R17–R14) ← (B) (R13–R10) ← (A) | |
| | T2R2L | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 2 9 5 | 1 | 1 | (T27–T20) ← (R2L7–R2L0) | |
| | TLCA | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 2 0 D | 1 | 1 | (LC) ← (A) (RLC) ← (A) | |
| | SNZT1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 8 0 | 1 | 1 | V12 = 0: (T1F) = 1 ? (T1F) ← 0 V12 = 1: SNZT1 = NOP | |
| SNZT2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 8 1 | 1 | 1 | V13 = 0: (T2F) = 1 ? (T2F) ← 0 V13 = 1: SNZT2 = NOP | | |
| SNZT3 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 8 2 | 1 | 1 | V20 = 0: (T3F) = 1 ? (T3F) ← 0 V20 = 1: SNZT3 = NOP | | |

| Skip condition | Carry flag CY | Detailed description |
|--------------------|---------------|--|
| – | – | Transfers the contents of register A to timer control register PA. |
| – | – | Transfers the contents of timer control register W1 to register A. |
| – | – | Transfers the contents of register A to timer control register W1. |
| – | – | Transfers the contents of timer control register W2 to register A. |
| – | – | Transfers the contents of register A to timer control register W2. |
| – | – | Transfers the contents of timer control register W3 to register A. |
| – | – | Transfers the contents of register A to timer control register W3. |
| – | – | Transfers the contents of timer control register W4 to register A. |
| – | – | Transfers the contents of register A to timer control register W4. |
| – | – | Transfers the high-order 4 bits of prescaler to register B, and transfers the low-order 4 bits of prescaler to register A. |
| – | – | Transfers the contents of register B to the high-order 4 bits of prescaler and prescaler reload register RPS, and transfers the contents of register A to the low-order 4 bits of prescaler and prescaler reload register RPS. |
| – | – | Transfers the high-order 4 bits of timer 1 to register B, and transfers the low-order 4 bits of timer 1 to register A. |
| – | – | Transfers the contents of register B to the high-order 4 bits of timer 1 and timer 1 reload register R1, and transfers the contents of register A to the low-order 4 bits of timer 1 and timer 1 reload register R1. |
| – | – | Transfers the high-order 4 bits of timer 2 to register B, and transfers the low-order 4 bits of timer 2 to register A. |
| – | – | Transfers the contents of register B to the high-order 4 bits of timer 2 and timer 2 reload register R2L, and transfers the contents of register A to the low-order 4 bits of timer 2 and timer 2 reload register R2L. |
| – | – | Transfers the contents of register B to the high-order 4 bits of timer 2 reload register R2H, and transfers the contents of register A to the low-order 4 bits of timer 2 reload register R2H. |
| – | – | Transfers the contents of register B to the high-order 4 bits of timer 1 reload register R1, and transfers the contents of register A to the low-order 4 bits of timer 1 reload register R1. |
| – | – | Transfers the contents of timer 2 reload register R2L to timer 2. |
| – | – | Transfers the contents of register A to timer LC and timer LC reload register RLC. |
| V12 = 0: (T1F) = 1 | – | When V12 = 0 : Clears (0) to the T1F flag and skips the next instruction when timer 1 interrupt request flag T1F is "1". When the T1F flag is "0", executes the next instruction. When V12 = 1 : This instruction is equivalent to the NOP instruction. (V12: bit 2 of interrupt control register V1) |
| V13 = 0: (T2F) = 1 | – | When V13 = 0 : Clears (0) to the T2F flag and skips the next instruction when timer 2 interrupt request flag T2F is "1". When the T2F flag is "0", executes the next instruction. When V13 = 1 : This instruction is equivalent to the NOP instruction. (V13: bit 3 of interrupt control register V1) |
| V20 = 0: (T3F) = 1 | – | When V20 = 0 : Clears (0) to the T3F flag and skips the next instruction when timer 3 interrupt request flag T3F is "1". When the T3F flag is "0", executes the next instruction. When V20 = 1 : This instruction is equivalent to the NOP instruction. (V20: bit 0 of interrupt control register V2) |

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | | | | Number of words | Number of cycles | Function |
|-----------------------------------|----------|------------------|----|----|----|----|----|----|----|----|----|----------------------|-----------------|------------------|------------------------------|
| | | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Hexadecimal notation | | | |
| Input/Output operation | IAP0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 6 0 | 1 | 1 | (A) ← (P0) |
| | OP0A | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 2 0 | 1 | 1 | (P0) ← (A) |
| | IAP1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 2 6 1 | 1 | 1 | (A) ← (P1) |
| | OP1A | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 2 2 1 | 1 | 1 | (P1) ← (A) |
| | IAP2 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 2 6 2 | 1 | 1 | (A) ← (P2) |
| | OP2A | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 2 2 2 | 1 | 1 | (P2) ← (A) |
| | CLD | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 1 1 | 1 | 1 | (D) ← 1 |
| | RD | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 1 4 | 1 | 1 | (D(Y)) ← 0 (Y) = 0 to 7 |
| | SD | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 1 5 | 1 | 1 | (D(Y)) ← 1 (Y) = 0 to 7 |
| | SZD | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 2 4 | 1 | 1 | (D(Y)) = 0 ? (Y) = 0 to 5 |
| | | | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 2 B | 1 | 1 | |
| | RCP | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 2 8 C | 1 | 1 | (C) ← 0 |
| | SCP | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 2 8 D | 1 | 1 | (C) ← 1 |
| | TAPU0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 2 5 7 | 1 | 1 | (A) ← (PU0) |
| | TPU0A | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 2 2 D | 1 | 1 | (PU0) ← (A) |
| | TAPU1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 2 5 E | 1 | 1 | (A) ← (PU1) |
| | TPU1A | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 2 2 E | 1 | 1 | (PU1) ← (A) |
| | TAK0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 2 5 6 | 1 | 1 | (A) ← (K0) |
| | TK0A | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 2 1 B | 1 | 1 | (K0) ← (A) |
| | TAK1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 2 5 9 | 1 | 1 | (A) ← (K1) |
| | TK1A | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 2 1 4 | 1 | 1 | (K1) ← (A) |
| | TAK2 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 2 5 A | 1 | 1 | (A) ← (K2) |
| | TK2A | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 2 1 5 | 1 | 1 | (K2) ← (A) |
| | TFR0A | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 2 2 8 | 1 | 1 | (FR0) ← (A) |
| | TFR1A | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 2 9 | 1 | 1 | (FR1) ← (A) |
| | TFR2A | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 2 2 A | 1 | 1 | (FR2) ← (A) |

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | | | | Hexadecimal notation | Number of words | Number of cycles | Function |
|-----------------------------------|----------|------------------|----|----|----|----|----|----|----|----|-------|-------|----------------------|--|---|----------|
| | | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | | | | | |
| LCD operation | TAL1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 2 4 A | 1 | 1 | (A) ← (L1) | |
| | TL1A | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 2 0 A | 1 | 1 | (L1) ← (A) | |
| | TL2A | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 2 0 B | 1 | 1 | (L2) ← (A) | |
| | TL3A | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 2 0 C | 1 | 1 | (L3) ← (A) | |
| | TC1A | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 2 A 8 | 1 | 1 | (C1) ← (A) | |
| | TC2A | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 A 9 | 1 | 1 | (C2) ← (A) | |
| Clock operation | CRCK | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 2 9 B | 1 | 1 | RC oscillator selected | |
| | TAMR | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 2 5 2 | 1 | 1 | (A) ← (MR) | |
| | TMRA | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 2 1 6 | 1 | 1 | (MR) ← (A) | |
| | TRGA | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 0 9 | 1 | 1 | (RG) ← (A) | |
| Other operation | NOP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 | 1 | 1 | (PC) ← (PC) + 1 | |
| | POF | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 0 2 | 1 | 1 | Transition to clock operating mode | |
| | POF2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 0 8 | 1 | 1 | Transition to RAM back-up mode | |
| | EPOF | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 5 B | 1 | 1 | POF, POF2 instructions valid | |
| | SNZP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 0 3 | 1 | 1 | (P) = 1 ? | |
| | WRST | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 A 0 | 1 | 1 | (WDF1) = 1 ? (WDF1) ← 0 | |
| | DWDT | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 2 9 C | 1 | 1 | Stop of watchdog timer function enabled | |
| | SRST | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 0 1 | 1 | 1 | System reset | |
| | RUPT | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 5 8 | 1 | 1 | (UPTF) ← 0 | |
| | SUPT | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 5 9 | 1 | 1 | (UPTF) ← 1 | |
| SVDE | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 2 9 3 | 1 | 1 | At power down mode, voltage drop detection circuit valid | | |

Note: SVDE instruction can be used only in H version.

| Skip condition | Carry flag CY | Detailed description |
|----------------|---------------|--|
| – | – | Transfers the contents of LCD control register L1 to register A. |
| – | – | Transfers the contents of register A to LCD control register L1. |
| – | – | Transfers the contents of register A to LCD control register L2. |
| – | – | Transfers the contents of register A to LCD control register L3. |
| – | – | Transfers the contents of register A to LCD control register C1. |
| – | – | Transfers the contents of register A to LCD control register C2. |
| – | – | Selects the RC oscillation circuit for main clock, stops the on-chip oscillator (internal oscillator). |
| – | – | Transfers the contents of clock control register MR to register A. |
| – | – | Transfers the contents of register A to clock control register MR. |
| – | – | Transfers the contents of register A to clock control register RG. |
| – | – | No operation; Adds 1 to program counter value, and others remain unchanged. |
| – | – | Puts the system in clock operating mode by executing the POF instruction after executing the EPOF instruction. |
| – | – | Puts the system in RAM back-up mode by executing the POF2 instruction after executing the EPOF instruction. |
| – | – | Makes the immediate after POF or POF2 instruction valid by executing the EPOF instruction. |
| (P) = 1 | – | Skips the next instruction when the P flag is "1". After skipping, the P flag remains unchanged. |
| (WDF1) = 1 | – | Clears (0) to the WDF1 flag and skips the next instruction when watchdog timer flag WDF1 is "1." When the WDF1 flag is "0", executes the next instruction. Also, stops the watchdog timer function when executing the WRST instruction immediately after the DWDT instruction. |
| – | – | Stops the watchdog timer function by the WRST instruction. |
| – | – | System reset occurs. |
| – | – | Clears (0) to the high-order bit reference enable flag UPTF. |
| – | – | Sets (1) to the high-order bit reference enable flag UPTF. |
| – | – | Validates the voltage drop detection circuit at power down (clock operating mode and RAM back-up mode). |

INSTRUCTION CODE TABLE

| D3-D0 | D9-D4 Hex. notation | 000000 | 000001 | 000010 | 000011 | 000100 | 000101 | 000110 | 000111 | 001000 | 001001 | 001010 | 001011 | 001100 | 001101 | 001110 | 001111 | 010000 | 011000 |
|-------|---------------------------|--------|--------|----------|--------|---------|---------|---------|----------|------------|------------|-------------|-------------|--------|--------|--------|--------|--------|--------|
| | | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10-17 | 18-1F |
| 0000 | 0 | NOP | BLA | SZB 0 | BMLA | - | TASP | A 0 | LA 0 | TABP 0 | TABP 16 | TABP 32* | TABP 48* | BML | BML | BL | BL | BM | B |
| 0001 | 1 | SRST | CLD | SZB 1 | - | - | TAD | A 1 | LA 1 | TABP 1 | TABP 17 | TABP 33* | TABP 49* | BML | BML | BL | BL | BM | B |
| 0010 | 2 | POF | - | SZB 2 | - | - | TAX | A 2 | LA 2 | TABP 2 | TABP 18 | TABP 34* | TABP 50* | BML | BML | BL | BL | BM | B |
| 0011 | 3 | SNZP | INY | SZB 3 | - | - | TAZ | A 3 | LA 3 | TABP 3 | TABP 19 | TABP 35* | TABP 51* | BML | BML | BL | BL | BM | B |
| 0100 | 4 | DI | RD | SZD | - | RT | TAV1 | A 4 | LA 4 | TABP 4 | TABP 20 | TABP 36* | TABP 52* | BML | BML | BL | BL | BM | B |
| 0101 | 5 | EI | SD | SEAn | - | RTS | TAV2 | A 5 | LA 5 | TABP 5 | TABP 21 | TABP 37* | TABP 53* | BML | BML | BL | BL | BM | B |
| 0110 | 6 | RC | - | SEAM | - | RTI | - | A 6 | LA 6 | TABP 6 | TABP 22 | TABP 38* | TABP 54* | BML | BML | BL | BL | BM | B |
| 0111 | 7 | SC | DEY | - | - | - | - | A 7 | LA 7 | TABP 7 | TABP 23 | TABP 39* | TABP 55* | BML | BML | BL | BL | BM | B |
| 1000 | 8 | POF2 | AND | - | SNZ0 | LZ 0 | RUPT | A 8 | LA 8 | TABP 8 | TABP 24 | TABP 40* | TABP 56* | BML | BML | BL | BL | BM | B |
| 1001 | 9 | - | OR | TDA | - | LZ 1 | SUPT | A 9 | LA 9 | TABP 9 | TABP 25 | TABP 41* | TABP 57* | BML | BML | BL | BL | BM | B |
| 1010 | A | AM | TEAB | TABE | SNZ10 | LZ 2 | - | A 10 | LA 10 | TABP 10 | TABP 26 | TABP 42* | TABP 58* | BML | BML | BL | BL | BM | B |
| 1011 | B | AMC | - | - | - | LZ 3 | EPOF | A 11 | LA 11 | TABP 11 | TABP 27 | TABP 43* | TABP 59* | BML | BML | BL | BL | BM | B |
| 1100 | C | TYA | CMA | - | - | RB 0 | SB 0 | A 12 | LA 12 | TABP 12 | TABP 28 | TABP 44* | TABP 60* | BML | BML | BL | BL | BM | B |
| 1101 | D | - | RAR | - | - | RB 1 | SB 1 | A 13 | LA 13 | TABP 13 | TABP 29 | TABP 45* | TABP 61* | BML | BML | BL | BL | BM | B |
| 1110 | E | TBA | TAB | - | TV2A | RB 2 | SB 2 | A 14 | LA 14 | TABP 14 | TABP 30 | TABP 46* | TABP 62* | BML | BML | BL | BL | BM | B |
| 1111 | F | - | TAY | SZC | TV1A | RB 3 | SB 3 | A 15 | LA 15 | TABP 15 | TABP 31 | TABP 47* | TABP 63* | BML | BML | BL | BL | BM | B |

The above table shows the relationship between machine language codes and machine language instructions. D3-D0 show the low-order 4 bits of the machine language code, and D9-D4 show the high-order 6 bits of the machine language code. The hexadecimal representation of the code is also provided. There are one-word instructions and two-word instructions, but only the first word of each instruction is shown. Do not use code marked "-."

The codes for the second word of a two-word instruction are described below.

| | The second word |
|------|-----------------|
| BL | 1p paaa aaaa |
| BML | 1p paaa aaaa |
| BLA | 1p pp00 pppp |
| BMLA | 1p pp00 pppp |
| SEA | 00 0111 nnnn |
| SZD | 00 0010 1011 |

* * cannot be used in the M3455xM4/M4H.

INSTRUCTION CODE TABLE (continued)

| D3-D0 | Hex. notation | D9-D4 | 100000 | 100001 | 100010 | 100011 | 100100 | 100101 | 100110 | 100111 | 101000 | 101001 | 101010 | 101011 | 101100 | 101101 | 101110 | 101111 | 110000 |
|-------|---------------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 2A | 2B | 2C | 2D | 2E | 2F | 30-3F | |
| 0000 | 0 | - | TW3A | OP0A | T1AB | - | - | IAP0 | TAB1 | SNZT1 | - | WRST | TMA0 | TAM0 | XAM0 | XAMI0 | XAMD0 | LXY | |
| 0001 | 1 | - | TW4A | OP1A | T2AB | - | - | IAP1 | TAB2 | SNZT2 | - | - | TMA1 | TAM1 | XAM1 | XAMI1 | XAMD1 | LXY | |
| 0010 | 2 | - | - | OP2A | - | - | TAMR | IAP2 | - | SNZT3 | - | - | TMA2 | TAM2 | XAM2 | XAMI2 | XAMD2 | LXY | |
| 0011 | 3 | - | - | - | - | - | TAI1 | - | - | - | SVDE** | - | TMA3 | TAM3 | XAM3 | XAMI3 | XAMD3 | LXY | |
| 0100 | 4 | - | TK1A | - | - | - | - | - | - | - | T2HAB | - | TMA4 | TAM4 | XAM4 | XAMI4 | XAMD4 | LXY | |
| 0101 | 5 | - | TK2A | - | TPSAB | - | - | - | TABPS | - | T2R2L | - | TMA5 | TAM5 | XAM5 | XAMI5 | XAMD5 | LXY | |
| 0110 | 6 | - | TMRA | - | - | - | TAK0 | - | - | - | - | - | TMA6 | TAM6 | XAM6 | XAMI6 | XAMD6 | LXY | |
| 0111 | 7 | - | TI1A | - | - | - | TAPU0 | - | - | - | - | - | TMA7 | TAM7 | XAM7 | XAMI7 | XAMD7 | LXY | |
| 1000 | 8 | - | - | TFR0A | - | - | - | - | - | - | - | TC1A | TMA8 | TAM8 | XAM8 | XAMI8 | XAMD8 | LXY | |
| 1001 | 9 | TRGA | - | TFR1A | - | - | TAK1 | - | - | - | - | TC2A | TMA9 | TAM9 | XAM9 | XAMI9 | XAMD9 | LXY | |
| 1010 | A | TL1A | - | TFR2A | - | TAL1 | TAK2 | - | - | - | - | TPAA | TMA10 | TAM10 | XAM10 | XAMI10 | XAMD10 | LXY | |
| 1011 | B | TL2A | TK0A | - | - | TAW1 | - | - | - | - | CRCK | - | TMA11 | TAM11 | XAM11 | XAMI11 | XAMD11 | LXY | |
| 1100 | C | TL3A | - | - | - | TAW2 | - | - | - | RCP | DWDT | - | TMA12 | TAM12 | XAM12 | XAMI12 | XAMD12 | LXY | |
| 1101 | D | TLCA | - | TPU0A | - | TAW3 | - | - | - | SCP | - | - | TMA13 | TAM13 | XAM13 | XAMI13 | XAMD13 | LXY | |
| 1110 | E | TW1A | - | TPU1A | - | TAW4 | TAPU1 | - | - | - | - | - | TMA14 | TAM14 | XAM14 | XAMI14 | XAMD14 | LXY | |
| 1111 | F | TW2A | - | - | TR1AB | - | - | - | - | - | - | - | TMA15 | TAM15 | XAM15 | XAMI15 | XAMD15 | LXY | |

The above table shows the relationship between machine language codes and machine language instructions. D3-D0 show the low-order 4 bits of the machine language code, and D9-D4 show the high-order 6 bits of the machine language code. The hexadecimal representation of the code is also provided. There are one-word instructions and two-word instructions, but only the first word of each instruction is shown. Do not use code marked "-."

The codes for the second word of a two-word instruction are described below.

| | The second word |
|------|-----------------|
| BL | 1p paaa aaaa |
| BML | 1p paaa aaaa |
| BLA | 1p pp00 pppp |
| BMLA | 1p pp00 pppp |
| SEA | 00 0111 nnnn |
| SZD | 00 0010 1011 |

• ** can be used only in the M3455xM4H/M8H/G8H.

ELECTRICAL CHARACTERISTICS

(1) Mask ROM version

ABSOLUTE MAXIMUM RATINGS (Mask ROM version)

| Symbol | Parameter | Conditions | Ratings | Unit |
|------------------|--|-------------------------------------|------------------------------|------|
| V _{DD} | Supply voltage | | -0.3 to 6.5 | V |
| V _I | Input voltage P0, P1, P2, D0–D5, $\overline{\text{RESET}}$, INT, X _{IN} , X _{CIN} | | -0.3 to V _{DD} +0.3 | V |
| V _I | Input voltage CNTR | | -0.3 to V _{DD} +0.3 | V |
| V _O | Output voltage P0, P1, P2, D0–D7, $\overline{\text{RESET}}$, CNTR | Output transistors in cut-off state | -0.3 to V _{DD} +0.3 | V |
| V _O | Output voltage C, X _{OUT} , X _{COU} T | | -0.3 to V _{DD} +0.3 | V |
| V _O | Output voltage SEG0–SEG28, COM0–COM3 (Note) | | -0.3 to V _{DD} +0.3 | V |
| P _d | Power dissipation | T _a = 25 °C | 300 | mW |
| T _{opr} | Operating temperature range | | -20 to 85 | °C |
| T _{stg} | Storage temperature range | | -40 to 125 | °C |

Note: SEG13 pin is not equipped with the 4552 Group.

RECOMMENDED OPERATING CONDITIONS 1

(Mask ROM version: Ta = -20 °C to 85 °C, VDD = 1.8 to 5.5 V, unless otherwise noted)

| Symbol | Parameter | Conditions | Limits | | | Unit |
|-----------|--|------------------------------|-----------|------|---------|------|
| | | | Min. | Typ. | Max. | |
| VDD | Supply voltage (when ceramic resonator is used) | f(STCK) ≤ 6 MHz | 4 | | 5.5 | V |
| | | f(STCK) ≤ 4.4 MHz | 2.7 | | 5.5 | |
| | | f(STCK) ≤ 2.2 MHz | 2 | | 5.5 | |
| | | f(STCK) ≤ 1.1 MHz | 1.8 | | 5.5 | |
| VDD | Supply voltage (when quartz-crystal/on-chip oscillation is used) | | 1.8 | | 5.5 | V |
| VDD | Supply voltage (when RC oscillation is used) | f(STCK) ≤ 4.4 MHz | 2.7 | | 5.5 | V |
| VRAM | RAM back-up voltage | at RAM back-up mode | 1.6 | | | V |
| VSS | Supply voltage | | | 0 | | V |
| VLC3 | LCD power supply (Note 1) | | 1.8 | | VDD | V |
| VIH | "H" level input voltage | P0, P1, P2, D0–D5 | 0.8VDD | | VDD | V |
| | | XIN, XCIN | 0.7VDD | | VDD | |
| | | RESET | 0.85VDD | | VDD | |
| | | INT | 0.85VDD | | VDD | |
| | | CNTR | 0.8VDD | | VDD | |
| VIL | "L" level input voltage | P0, P1, P2, D0–D5 | 0 | | 0.2VDD | V |
| | | XIN, XCIN | 0 | | 0.3VDD | |
| | | RESET | 0 | | 0.3VDD | |
| | | INT | 0 | | 0.15VDD | |
| | | CNTR | 0 | | 0.15VDD | |
| IOH(peak) | "H" level peak output current | P0, P1, P2, D0–D5 | VDD = 5 V | | -20 | mA |
| | | | VDD = 3 V | | -10 | |
| | | C CNTR | VDD = 5 V | | -30 | |
| | | | VDD = 3 V | | -15 | |
| IOH(avg) | "H" level average output current (Note 2) | P0, P1, P2, D0–D5 | VDD = 5 V | | -10 | mA |
| | | | VDD = 3 V | | -5 | |
| | | C CNTR | VDD = 5 V | | -20 | |
| | | | VDD = 3 V | | -10 | |
| IOL(peak) | "L" level peak output current | P0, P1, P2, D0–D7, C CNTR | VDD = 5 V | | 24 | mA |
| | | | VDD = 3 V | | 12 | |
| | | RESET | VDD = 5 V | | 10 | |
| | | | VDD = 3 V | | 4 | |
| IOL(avg) | "L" level average output current (Note 2) | P0, P1, P2, D0–D7, C CNTR | VDD = 5 V | | 15 | mA |
| | | | VDD = 3 V | | 7 | |
| | | RESET | VDD = 5 V | | 5 | |
| | | | VDD = 3 V | | 2 | |
| ΣIOH(avg) | "H" level total average current | P0, P1, P2, D0–D5, C, CNTR | | | -40 | mA |
| ΣIOL(avg) | "L" level total average current | P0, P1, P2, D0–D5, C, CNTR | | | 60 | mA |
| | | D6, D7, RESET | | | 60 | |

Notes 1: At 1/2 bias: VLC1 = VLC2 = (1/2)•VLC3

At 1/3 bias: VLC1 = (1/3)•VLC3, VLC2 = (2/3)•VLC3

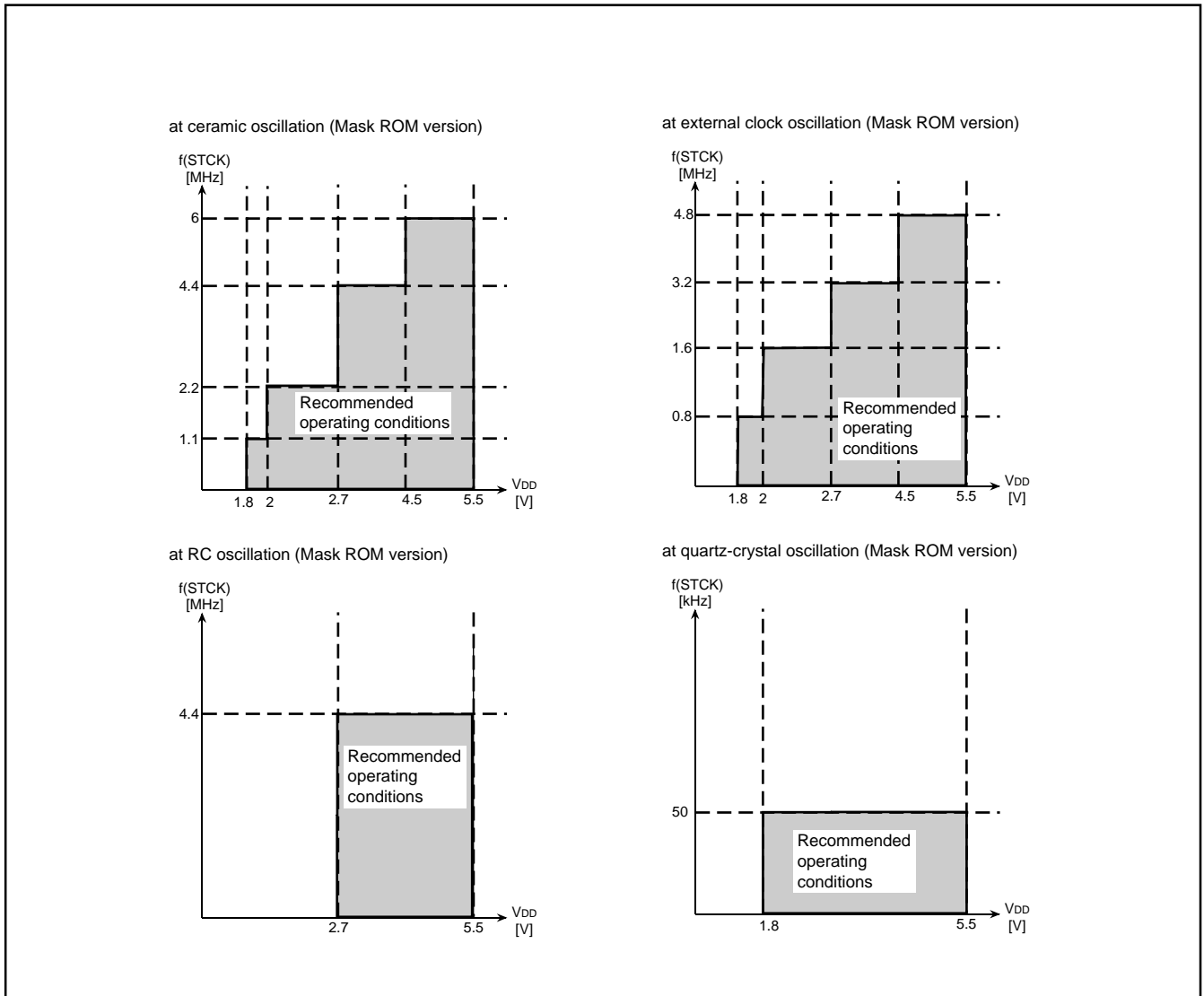
2: The average output current is the average value during 100 ms.

RECOMMENDED OPERATING CONDITIONS 2

(Mask ROM version: Ta = -20 °C to 85 °C, VDD = 1.8 to 5.5 V, unless otherwise noted)

| Symbol | Parameter | Conditions | Limits | | | Unit | | | |
|-----------------------|--|------------------|---|---------------------------|------------------|------|-----------|-----|-----|
| | | | Min. | Typ. | Max. | | | | |
| f(XIN) | Oscillation frequency (with a ceramic resonator) | Through mode | VDD = 4 to 5.5 V | | 6 | MHz | | | |
| | | | VDD = 2.7 to 5.5 V | | 4.4 | | | | |
| | | | VDD = 2 to 5.5 V | | 2.2 | | | | |
| | | | VDD = 1.8 to 5.5 V | | 1.1 | | | | |
| | | Frequency/2 mode | VDD = 2.7 to 5.5 V | | 6 | | | | |
| | | | VDD = 2 to 5.5 V | | 4.4 | | | | |
| | | | VDD = 1.8 to 5.5 V | | 2.2 | | | | |
| | | Frequency/4 mode | VDD = 2 to 5.5 V | | 6 | | | | |
| | | | VDD = 1.8 to 5.5 V | | 4.4 | | | | |
| | | Frequency/8 mode | VDD = 1.8 to 5.5 V | | 6 | | | | |
| | | f(XIN) | Oscillation frequency (at RC oscillation) (Note) | VDD = 2.7 to 5.5 V | | | | 4.4 | MHz |
| | | f(XIN) | Oscillation frequency (with a ceramic oscillation selected, external clock input) | Through mode | VDD = 4 to 5.5 V | | | 4.8 | MHz |
| VDD = 2.7 to 5.5 V | | | | | 3.2 | | | | |
| VDD = 2 to 5.5 V | | | | | 1.6 | | | | |
| VDD = 1.8 to 5.5 V | | | | | 0.8 | | | | |
| Frequency/2 mode | VDD = 2.7 to 5.5 V | | | | 4.8 | | | | |
| | VDD = 2 to 5.5 V | | | | 3.2 | | | | |
| | VDD = 1.8 to 5.5 V | | | | 1.6 | | | | |
| Frequency/4 mode | VDD = 2 to 5.5 V | | | | 4.8 | | | | |
| | VDD = 1.8 to 5.5 V | | | | 3.2 | | | | |
| Frequency/8 mode | VDD = 1.8 to 5.5 V | | | | 4.8 | | | | |
| f(XCIN) | Oscillation frequency (sub-clock) | | | Quartz-crystal oscillator | | | 50 | kHz | |
| f(CNTR) | Timer external input frequency | | | CNTR | | | f(STCK)/6 | Hz | |
| t _w (CNTR) | Timer external input period ("H" and "L" pulse width) | CNTR | 3/f(STCK) | | | s | | | |
| TPON | Power-on reset circuit valid supply voltage rising time | VDD = 0 → 1.8 V | | | 100 | μs | | | |

Note: The frequency is affected by a capacitor, a resistor and a microcomputer. So, set the constants within the range of the frequency limits.



**System clock (STCK) operating condition map
(Mask ROM version)**

ELECTRICAL CHARACTERISTICS 1

(Mask ROM version: Ta = -20 °C to 85 °C, VDD = 1.8 to 5.5 V, unless otherwise noted)

| Symbol | Parameter | Test conditions | | Limits | | | Unit |
|----------------------|---|--|--------------|--------|------|------|------|
| | | | | Min. | Typ. | Max. | |
| VOH | "H" level output voltage P0, P1, P2, D0–D5 | VDD = 5 V | IOH = -10 mA | 3 | | | V |
| | | | IOH = -3 mA | 4.1 | | | |
| | | VDD = 3 V | IOH = -5 mA | 2.1 | | | |
| | | | IOH = -1 mA | 2.4 | | | |
| VOH | "H" level output voltage C, CNTR | VDD = 5 V | IOH = -20 mA | 3 | | | V |
| | | | IOH = -6 mA | 4.1 | | | |
| | | VDD = 3 V | IOH = -10 mA | 2.1 | | | |
| | | | IOH = -3 mA | 2.4 | | | |
| VOL | "L" level output voltage P0, P1, P2, D0–D7, C, CNTR | VDD = 5 V | IOL = 15 mA | | | 2 | V |
| | | | IOL = 5 mA | | | 0.9 | |
| | | VDD = 3 V | IOL = 9 mA | | | 1.4 | |
| | | | IOL = 3 mA | | | 0.9 | |
| VOL | "L" level output voltage RESET | VDD = 5 V | IOL = 5 mA | | | 2 | V |
| | | | IOL = 1 mA | | | 0.6 | |
| | | VDD = 3 V | IOL = 2 mA | | | 0.9 | |
| I _{IH} | "H" level input current P0, P1, P2, D0–D5, X _{IN} , X _{CIN} , RESET CNTR, INT | Vi = VDD | | | | 2 | μA |
| I _{IL} | "L" level input current P0, P1, P2, D0–D5, X _{IN} , X _{CIN} , RESET CNTR, INT | Vi = 0 V P0, P1 No pull-up | | | | -2 | μA |
| RPU | Pull-up resistor value P0, P1, RESET | Vi = 0 V | VDD = 5 V | 30 | 60 | 125 | kΩ |
| | | | VDD = 3 V | 50 | 120 | 250 | |
| VT+ – VT- | Hysteresis RESET | VDD = 5 V | | | 1 | | V |
| | | VDD = 3 V | | | 0.4 | | |
| VT+ – VT- | Hysteresis INT | VDD = 5 V | | | 0.6 | | V |
| | | VDD = 3 V | | | 0.3 | | |
| VT+ – VT- | Hysteresis CNTR | VDD = 5 V | | | 0.2 | | V |
| | | VDD = 3 V | | | 0.2 | | |
| f(RING) | On-chip oscillator clock frequency | VDD = 5 V | | 200 | 500 | 700 | kHz |
| | | VDD = 3 V | | 100 | 250 | 400 | |
| Δf(X _{IN}) | Frequency error (with RC oscillation, error of external R, C not included) (Note 1) | VDD = 5 V ± 10 %, Ta = 25 °C | | | | ±17 | % |
| | | VDD = 3 V ± 10 %, Ta = 25 °C | | | | ±17 | |
| RCOM | COM output impedance (Note 2) | VDD = 5 V | | | 1.5 | 7.5 | kΩ |
| | | VDD = 3 V | | | 2 | 10 | |
| RSEG | SEG output impedance (Note 2) | VDD = 5 V | | | 1.5 | 7.5 | kΩ |
| | | VDD = 3 V | | | 2 | 10 | |
| RVLC | Internal resistor for LCD power supply | When dividing resistor 2r X 3 selected | | 300 | 480 | 960 | kΩ |
| | | When dividing resistor 2r X 2 selected | | 200 | 320 | 640 | |
| | | When dividing resistor r X 3 selected | | 150 | 240 | 480 | |
| | | When dividing resistor r X 2 selected | | 100 | 160 | 320 | |

Notes 1: When RC oscillation is used, use the external 33 pF capacitor (C).

2: The impedance state is the resistor value of the output voltage.

at VLC3 level output: Vo = 0.8 VLC3

at VLC2 level output: Vo = 0.8 VLC2

at VLC1 level output: Vo = 0.2 VLC2 + VLC1

at Vss level output: Vo = 0.2 Vss

ELECTRICAL CHARACTERISTICS 2

(Mask ROM version: Ta = -20 °C to 85 °C, VDD = 1.8 to 5.5 V, unless otherwise noted)

| Symbol | Parameter | Test conditions | Limits | | | Unit | | |
|--|---------------------|--|------------------|---------------------|------|------|-----|----|
| | | | Min. | Typ. | Max. | | | |
| IDD | Supply current | at active mode (with a ceramic resonator) | VDD = 5 V | f(STCK) = f(XIN)/8 | | 1.2 | 2.4 | mA |
| | | | f(XIN) = 6 MHz | f(STCK) = f(XIN)/4 | | 1.3 | 2.6 | |
| | | | f(RING) = stop | f(STCK) = f(XIN)/2 | | 1.6 | 3.2 | |
| | | | f(XCIN) = stop | f(STCK) = f(XIN) | | 2.2 | 4.4 | |
| | | | VDD = 5 V | f(STCK) = f(XIN)/8 | | 0.9 | 1.8 | mA |
| | | | f(XIN) = 4 MHz | f(STCK) = f(XIN)/4 | | 1 | 2 | |
| | | | f(RING) = stop | f(STCK) = f(XIN)/2 | | 1.2 | 2.4 | |
| | | | f(XCIN) = stop | f(STCK) = f(XIN) | | 1.6 | 3.2 | |
| | | | VDD = 3 V | f(STCK) = f(XIN)/8 | | 0.3 | 0.6 | mA |
| | | | f(XIN) = 4 MHz | f(STCK) = f(XIN)/4 | | 0.4 | 0.8 | |
| | | | f(RING) = stop | f(STCK) = f(XIN)/2 | | 0.5 | 1.0 | |
| | | | f(XCIN) = stop | f(STCK) = f(XIN) | | 0.7 | 1.4 | |
| | | at active mode (with an on-chip oscillator) | VDD = 5 V | f(STCK) = f(RING)/8 | | 50 | 100 | μA |
| | | | f(XIN) = stop | f(STCK) = f(RING)/4 | | 60 | 120 | |
| | | | f(RING) = active | f(STCK) = f(RING)/2 | | 80 | 160 | |
| | | | f(XCIN) = stop | f(STCK) = f(RING) | | 120 | 240 | |
| | | | VDD = 3 V | f(STCK) = f(RING)/8 | | 10 | 20 | μA |
| | | | f(XIN) = stop | f(STCK) = f(RING)/4 | | 13 | 26 | |
| | | | f(RING) = active | f(STCK) = f(RING)/2 | | 19 | 38 | |
| | | | f(XCIN) = stop | f(STCK) = f(RING) | | 31 | 62 | |
| | | at active mode (with a quartz-crystal oscillator) | VDD = 5 V | f(STCK) = f(XCIN)/8 | | 7 | 14 | μA |
| | | | f(XIN) = stop | f(STCK) = f(XCIN)/4 | | 8 | 16 | |
| | | | f(RING) = stop | f(STCK) = f(XCIN)/2 | | 10 | 20 | |
| | | | f(XCIN) = 32 kHz | f(STCK) = f(XCIN) | | 14 | 28 | |
| VDD = 3 V | f(STCK) = f(XCIN)/8 | | | 5 | 10 | μA | | |
| f(XIN) = stop | f(STCK) = f(XCIN)/4 | | | 6 | 12 | | | |
| f(RING) = stop | f(STCK) = f(XCIN)/2 | | | 7 | 14 | | | |
| f(XCIN) = 32 kHz | f(STCK) = f(XCIN) | | | 8 | 16 | | | |
| at clock operation mode (POF instruction execution) | f(XCIN) = 32 kHz | VDD = 5 V | | 6 | 12 | μA | | |
| | | VDD = 3 V | | 5 | 10 | | | |
| at RAM back-up mode (POF2 instruction execution) | | Ta = 25 °C | | 0.1 | 2 | μA | | |
| | | VDD = 5 V | | | 10 | | | |
| | | VDD = 3 V | | | 6 | | | |

VOLTAGE DROP DETECTION CIRCUIT CHARACTERISTICS

(Mask ROM version: Ta = -20 °C to 85 °C, unless otherwise noted)

| Symbol | Parameter | Test conditions | Limits | | | Unit |
|--|---|-----------------------------------|--------|------|------|------|
| | | | Min. | Typ. | Max. | |
| VRST ⁻ | Detection voltage (reset occurs) (Note 2) | Ta = 25 °C | 1.6 | 1.8 | 2 | V |
| | | -20 °C ≤ Ta < 0 °C | 1.7 | | 2.3 | |
| | | 0 °C ≤ Ta < 50 °C | 1.4 | | 2.2 | |
| | | 50 °C ≤ Ta ≤ 85 °C | 1.2 | | 1.9 | |
| VRST ⁺ | Detection voltage (reset release) (Note 3) | Ta = 25 °C | 1.7 | 1.9 | 2.1 | V |
| | | -20 °C ≤ Ta < 0 °C | 1.8 | | 2.4 | |
| | | 0 °C ≤ Ta < 50 °C | 1.5 | | 2.3 | |
| | | 50 °C ≤ Ta ≤ 85 °C | 1.3 | | 2 | |
| VRST ⁺ – VRST ⁻ | Detection voltage hysteresis | | | 0.1 | | V |
| IRST | Operation current (Note 4) | VDD = 5 V | | 50 | 100 | μA |
| | | VDD = 3 V | | 30 | 60 | |
| TRST | Detection time (Note 5) | VDD → (VRST ⁻ – 0.1 V) | | 0.2 | 1.2 | ms |

Notes 1: The voltage drop detection circuit is equipped with only the H version.

2: The detection voltage (VRST⁻) is defined as the voltage when reset occurs when the supply voltage (VDD) is falling.

3: The detection voltage (VRST⁺) is defined as the voltage when reset is released when the supply voltage (VDD) is rising from reset occurs.

4: In the H version, IRST is added to IDD (supply current).

5: The detection time (TRST) is defined as the time until reset occurs when the supply voltage (VDD) is falling to [VRST⁻ – 0.1 V].

6: The detection voltages (VRST⁺, VRST⁻) are set up lower than the minimum value of the supply voltage of the recommended operating conditions.

As for details, refer to the LIST OF PRECAUTIONS.

(2) One Time PROM version**ABSOLUTE MAXIMUM RATINGS (One Time PROM version)**

| Symbol | Parameter | Conditions | Ratings | Unit |
|--------|---|-------------------------------------|-----------------|------|
| VDD | Supply voltage | | -0.3 to 4.0 | V |
| VI | Input voltage P0, P1, P2, D0–D5, $\overline{\text{RESET}}$, INT, XIN, XCIN | | -0.3 to VDD+0.3 | V |
| VI | Input voltage CNTR | | -0.3 to VDD+0.3 | V |
| VO | Output voltage P0, P1, P2, D0–D7, $\overline{\text{RESET}}$, CNTR | Output transistors in cut-off state | -0.3 to VDD+0.3 | V |
| VO | Output voltage C, XOUT, XCOUT | | -0.3 to VDD+0.3 | V |
| VO | Output voltage SEG0–SEG28, COM0–COM3 (Note) | | -0.3 to VDD+0.3 | V |
| Pd | Power dissipation | Ta = 25 °C | 300 | mW |
| Topr | Operating temperature range | | -20 to 85 | °C |
| Tstg | Storage temperature range | | -40 to 125 | °C |

Note: SEG13 pin is not equipped with the 4552 Group.

RECOMMENDED OPERATING CONDITIONS 1

(One Time PROM version: Ta = -20 °C to 85 °C, VDD = 1.8 to 3.6 V, unless otherwise noted)

| Symbol | Parameter | Conditions | Limits | | | Unit |
|-----------|--|-------------------------------|-----------|------|---------|------|
| | | | Min. | Typ. | Max. | |
| VDD | Supply voltage (when ceramic resonator is used) | f(STCK) ≤ 4.4 MHz | 2.7 | | 3.6 | V |
| | | f(STCK) ≤ 2.2 MHz | 2 | | 3.6 | |
| | | f(STCK) ≤ 1.1 MHz | 1.8 | | 3.6 | |
| VDD | Supply voltage (when quartz-crystal/on-chip oscillator is used) | | 1.8 | | 3.6 | V |
| VDD | Supply voltage (when RC oscillation is used) | f(STCK) ≤ 4.4 MHz | 2.7 | | 3.6 | V |
| VRAM | RAM back-up voltage | at RAM back-up mode | 1.6 | | | V |
| VSS | Supply voltage | | | 0 | | V |
| VLC3 | LCD power supply (Note 1) | | 1.8 | | VDD | V |
| VIH | "H" level input voltage | P0, P1, P2, D0-D5 | 0.8VDD | | VDD | V |
| | | XIN, XCIN | 0.7VDD | | VDD | |
| | | RESET | 0.85VDD | | VDD | |
| | | INT | 0.85VDD | | VDD | |
| | | CNTR | 0.8VDD | | VDD | |
| VIL | "L" level input voltage | P0, P1, P2, D0-D5 | 0 | | 0.2VDD | V |
| | | XIN, XCIN | 0 | | 0.3VDD | |
| | | RESET | 0 | | 0.3VDD | |
| | | INT | 0 | | 0.15VDD | |
| | | CNTR | 0 | | 0.15VDD | |
| IOH(peak) | "H" level peak output current | P0, P1, P2, D0-D5 | VDD = 3 V | | -10 | mA |
| | | C, CNTR | VDD = 3 V | | -15 | |
| IOH(avg) | "H" level average output current (Note 2) | P0, P1, P2, D0-D5 | VDD = 3 V | | -5 | mA |
| | | C, CNTR | VDD = 3 V | | -10 | |
| IOL(peak) | "L" level peak output current | P0, P1, P2, D0-D7, C, CNTR | VDD = 3 V | | 12 | mA |
| | | RESET | VDD = 3 V | | 4 | |
| IOL(avg) | "L" level average output current (Note 2) | P0, P1, P2, D0-D7, C, CNTR | VDD = 3 V | | 7 | mA |
| | | RESET | VDD = 3 V | | 2 | |
| ΣIOH(avg) | "H" level total average current | P0, P1, P2, D0-D5, C, CNTR | | | -40 | mA |
| ΣIOL(avg) | "L" level total average current | P0, P1, P2, D0-D5, C, CNTR | | | 60 | mA |
| | | D6, D7, RESET | | | 60 | |

Notes 1: At 1/2 bias: VLC1 = VLC2 = (1/2)•VLC3

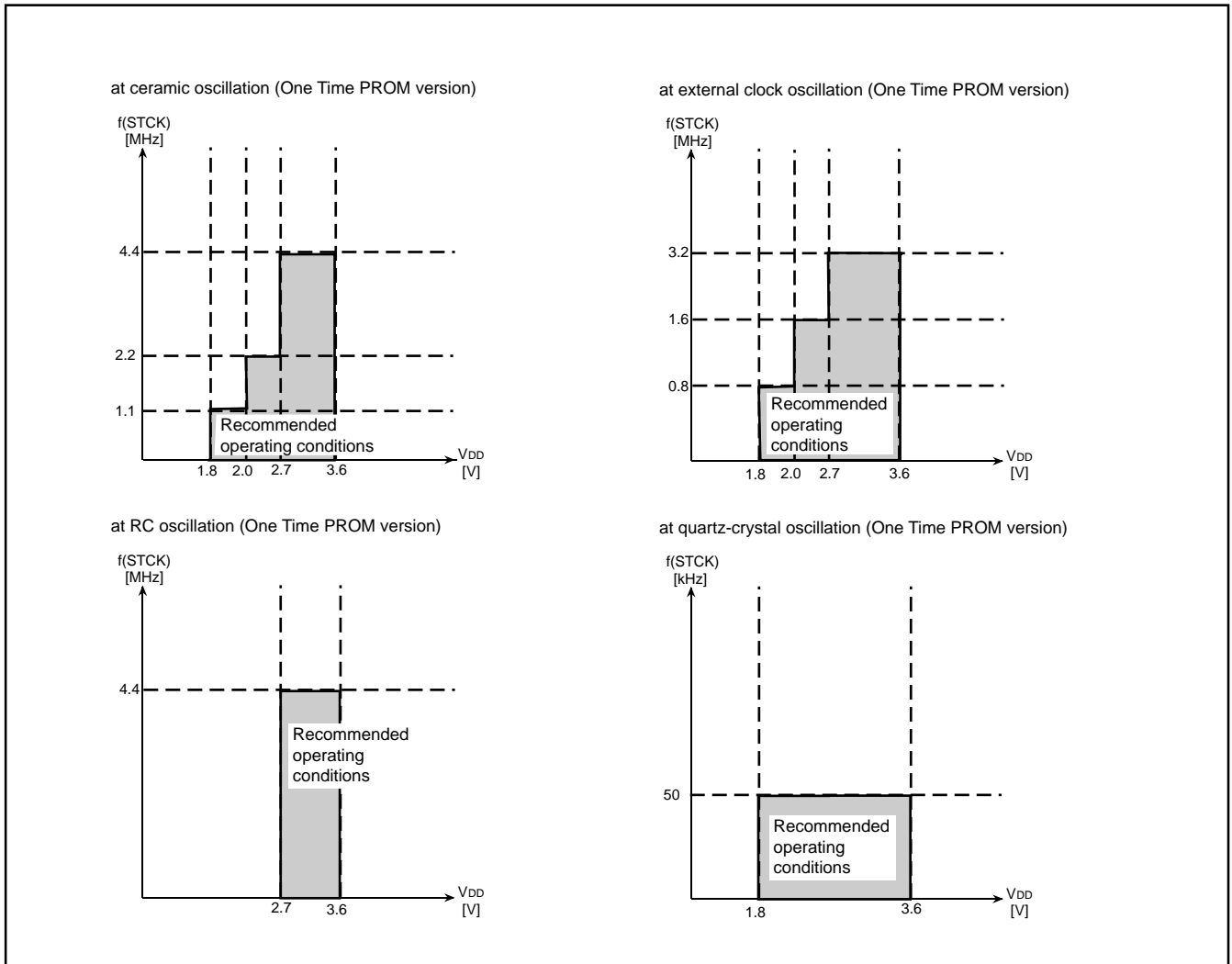
At 1/3 bias: VLC1 = (1/3)•VLC3, VLC2 = (2/3)•VLC3

2: The average output current is the average value during 100 ms.

RECOMMENDED OPERATING CONDITIONS 2(One Time PROM version: $T_a = -20\text{ }^{\circ}\text{C}$ to $85\text{ }^{\circ}\text{C}$, $V_{DD} = 1.8$ to 3.6 V , unless otherwise noted)

| Symbol | Parameter | Conditions | | Limits | | | Unit |
|-----------------------|---|---------------------------|--------------------|-----------|------|-----------|------|
| | | | | Min. | Typ. | Max. | |
| f(XIN) | Oscillation frequency (with a ceramic resonator) | Through mode | VDD = 2.7 to 3.6 V | | | 4.4 | MHz |
| | | | VDD = 2 to 3.6 V | | | 2.2 | |
| | | | VDD = 1.8 to 3.6 V | | | 1.1 | |
| | | Frequency/2 mode | VDD = 2.7 to 3.6 V | | | 6 | |
| | | | VDD = 2 to 3.6 V | | | 4.4 | |
| | | | VDD = 1.8 to 3.6 V | | | 2.2 | |
| | | Frequency/4 mode | VDD = 2 to 3.6 V | | | 6 | |
| VDD = 1.8 to 3.6 V | | | | 4.4 | | | |
| Frequency/8 mode | VDD = 1.8 to 3.6 V | | | 6 | | | |
| f(XIN) | Oscillation frequency (at RC oscillation) (Note) | VDD = 2.7 to 3.6 V | | | | 4.4 | MHz |
| f(XCIN) | Oscillation frequency (with a ceramic oscillation circuit selected, external clock input) | Through mode | VDD = 2.7 to 3.6 V | | | 3.2 | MHz |
| | | | VDD = 2 to 3.6 V | | | 1.6 | |
| | | | VDD = 1.8 to 3.6 V | | | 0.8 | |
| | | Frequency/2 mode | VDD = 2.7 to 3.6 V | | | 4.8 | |
| | | | VDD = 2 to 3.6 V | | | 3.2 | |
| | | | VDD = 1.8 to 3.6 V | | | 1.6 | |
| | | Frequency/4 mode | VDD = 2 to 3.6 V | | | 4.8 | |
| VDD = 1.8 to 3.6 V | | | | 3.2 | | | |
| Frequency/8 mode | VDD = 1.8 to 3.6 V | | | 4.8 | | | |
| f(XCIN) | Oscillation frequency (with a quartz-crystal oscillator) | Quartz-crystal oscillator | | | | 50 | kHz |
| f(CNTR) | Timer external input frequency | CNTR | | | | f(STCK)/6 | Hz |
| t _w (CNTR) | Timer external input period ("H" and "L" pulse width) | CNTR | | 3/f(STCK) | | | s |
| TPON | Power-on reset circuit valid supply voltage rising time | VDD = 0 → 1.8 V | | | | 100 | μs |

Note: The frequency is affected by a capacitor, a resistor and a microcomputer. So, set the constants within the range of the frequency limits.



**System clock (STCK) operating condition map
(One Time PROM version)**

ELECTRICAL CHARACTERISTICS

(One Time PROM version: Ta = -20 °C to 85 °C, VDD = 1.8 to 3.6 V, unless otherwise noted)

| Symbol | Parameter | Test conditions | | Limits | | | Unit |
|--|---|--|---------------------|--------|------|------|------|
| | | | | Min. | Typ. | Max. | |
| VOH | "H" level output voltage P0, P1, P2, D0–D5 | VDD = 3 V | IOH = -5 mA | 2.1 | | | V |
| | | | IOH = -1 mA | 2.4 | | | |
| VOH | "H" level output voltage C, CNTR | VDD = 3 V | IOH = -10 mA | 2.1 | | | V |
| | | | IOH = -3 mA | 2.4 | | | |
| VOL | "L" level output voltage P0, P1, P2, D0–D7, C, CNTR | VDD = 3 V | IOL = 9 mA | | | 1.4 | V |
| | | | IOL = 3 mA | | | 0.9 | |
| VOL | "L" level output voltage $\overline{\text{RESET}}$ | VDD = 3 V | IOL = 2 mA | | | 0.9 | V |
| IiH | "H" level input current P0, P1, P2, D0–D5, XIN, XCIN, $\overline{\text{RESET}}$ CNTR, INT | VI = VDD | | | | 2 | μA |
| IiL | "L" level input current P0, P1, P2, D0–D5, XIN, XCIN, $\overline{\text{RESET}}$ CNTR, INT | VI = 0 V P0, P1 No pull-up | | | | -2 | μA |
| RPU | Pull-up resistor value P0, P1, $\overline{\text{RESET}}$ | VI = 0 V VDD = 3 V | | 50 | 120 | 250 | kΩ |
| VT+ – VT- | Hysteresis $\overline{\text{RESET}}$ | VDD = 3 V | | | 0.4 | | V |
| VT+ – VT- | Hysteresis INT | VDD = 3 V | | | 0.3 | | V |
| VT+ – VT- | Hysteresis CNTR | VDD = 3 V | | | 0.2 | | V |
| f(RING) | On-chip oscillator clock frequency | VDD = 3 V | | 100 | 250 | 400 | kHz |
| Δf(XIN) | Frequency error (with RC oscillation, error of external R, C not included) (Note 1) | VDD = 3 V ± 10 %, Ta = 25 °C | | | | ±17 | % |
| RCOM | COM output impedance (Note 2) | VDD = 3 V | | | 2 | 10 | kΩ |
| RSEG | SEG output impedance (Note 2) | VDD = 3 V | | | 2 | 10 | kΩ |
| RVLC | Internal resistor for LCD power supply | When dividing resistor 2r X 3 selected | | 300 | 480 | 960 | kΩ |
| | | When dividing resistor 2r X 2 selected | | 200 | 320 | 640 | |
| | | When dividing resistor r X 3 selected | | 150 | 240 | 480 | |
| | | When dividing resistor r X 2 selected | | 100 | 160 | 320 | |
| IDD | Supply current at active mode (with a ceramic resonator) | VDD = 3 V f(XIN) = 4 MHz f(RING) = stop f(XCIN) = stop | f(STCK) = f(XIN)/8 | | 0.3 | 0.6 | mA |
| | | | f(STCK) = f(XIN)/4 | | 0.4 | 0.8 | |
| | | | f(STCK) = f(XIN)/2 | | 0.6 | 1.2 | |
| | | | f(STCK) = f(XIN) | | 0.9 | 1.8 | |
| | at active mode (with an on-chip oscillator) | VDD = 3 V f(XIN) = stop f(RING) = active f(XCIN) = stop | f(STCK) = f(RING)/8 | | 12 | 24 | μA |
| | | | f(STCK) = f(RING)/4 | | 17 | 34 | |
| | | | f(STCK) = f(RING)/2 | | 27 | 54 | |
| | | | f(STCK) = f(RING) | | 48 | 96 | |
| | at active mode (with a quartz-crystal oscillator) | VDD = 3 V f(XIN) = stop f(RING) = stop f(XCIN) = 32 kHz | f(STCK) = f(XCIN)/8 | | 5 | 10 | μA |
| | | | f(STCK) = f(XCIN)/4 | | 6 | 12 | |
| | | | f(STCK) = f(XCIN)/2 | | 7 | 14 | |
| | | | f(STCK) = f(XCIN) | | 9 | 18 | |
| at clock operation mode (POF instruction execution) | VDD = 3 V f(XCIN) = 32 kHz | | | 5 | 10 | μA | |
| at RAM back-up mode (POF2 instruction execution) | Ta = 25 °C | | | 0.1 | 2 | μA | |
| | VDD = 3 V | | | | 6 | | |

Notes 1: When RC oscillation is used, use the external 33 pF capacitor (C).

2: The impedance state is the resistor value of the output voltage.

at V_{LC3} level output: Vo = 0.8 V_{LC3}

at V_{LC2} level output: Vo = 0.8 V_{LC2}

at V_{LC1} level output: Vo = 0.2 V_{LC2} + V_{LC1}

at V_{SS} level output: Vo = 0.2 V_{SS}

VOLTAGE DROP DETECTION CIRCUIT CHARACTERISTICS

(One Time PROM version: Ta = -20 °C to 85 °C, unless otherwise noted)

| Symbol | Parameter | Test conditions | Limits | | | Unit |
|--|---|-----------------------------------|--------|------|------|------|
| | | | Min. | Typ. | Max. | |
| VRST ⁻ | Detection voltage (reset occurs) (Note 2) | Ta = 25 °C | 1.6 | 1.8 | 2 | V |
| | | -20 °C ≤ Ta < 0 °C | 1.7 | | 2.3 | |
| | | 0 °C ≤ Ta < 50 °C | 1.4 | | 2.2 | |
| | | 50 °C ≤ Ta ≤ 85 °C | 1.2 | | 1.9 | |
| VRST ⁺ | Detection voltage (reset release) (Note 3) | Ta = 25 °C | 1.7 | 1.9 | 2.1 | V |
| | | -20 °C ≤ Ta < 0 °C | 1.8 | | 2.4 | |
| | | 0 °C ≤ Ta < 50 °C | 1.5 | | 2.3 | |
| | | 50 °C ≤ Ta ≤ 85 °C | 1.3 | | 2 | |
| VRST ⁺ – VRST ⁻ | Detection voltage hysteresis | | | 0.1 | | V |
| IRST | Operation current (Note 4) | VDD = 3 V | | 30 | 60 | μA |
| TRST | Detection time (Note 5) | VDD → (VRST ⁻ – 0.1 V) | | 0.2 | 1.2 | ms |

Notes 1: The voltage drop detection circuit is equipped with only the H version.

2: The detection voltage (VRST⁻) is defined as the voltage when reset occurs when the supply voltage (VDD) is falling.

3: The detection voltage (VRST⁺) is defined as the voltage when reset is released when the supply voltage (VDD) is rising from reset occurs.

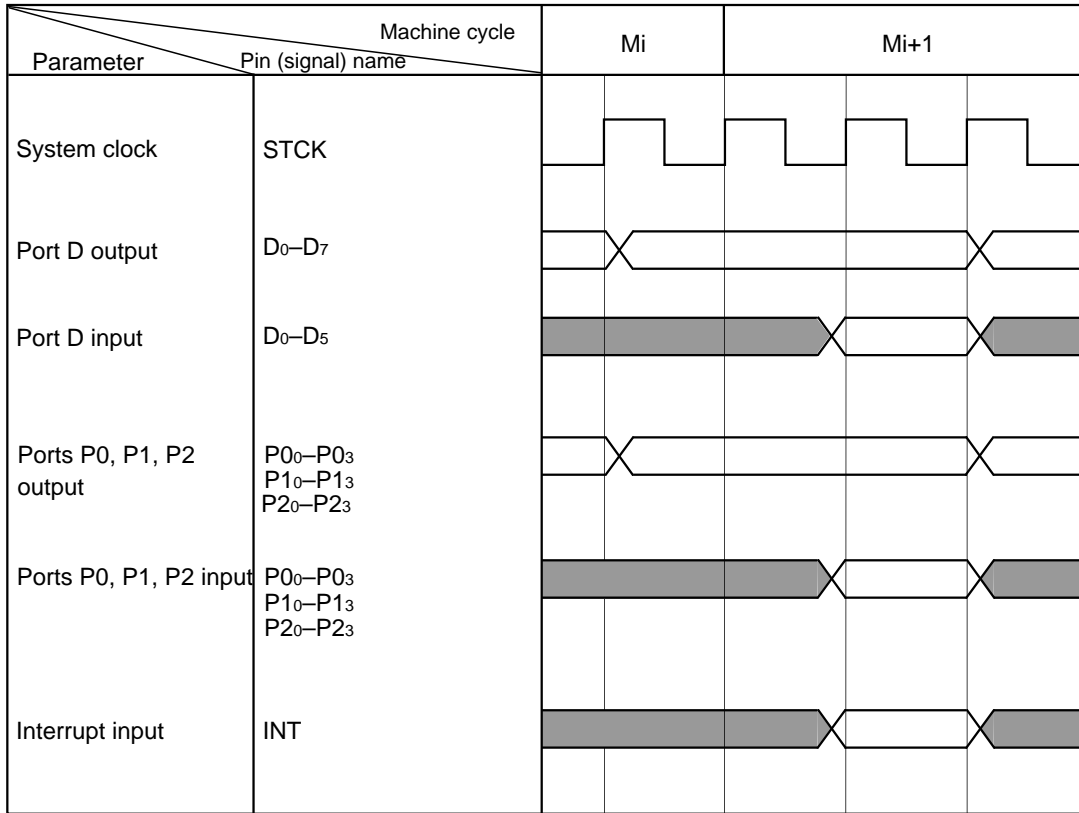
4: In the H version, IRST is added to IDD (supply current).

5: The detection time (TRST) is defined as the time until reset occurs when the supply voltage (VDD) is falling to [VRST⁻ – 0.1 V].

6: The detection voltages (VRST⁺, VRST⁻) are set up lower than the minimum value of the supply voltage of the recommended operating conditions.

As for details, refer to the LIST OF PRECAUTIONS.

BASIC TIMING DIAGRAM



BUILT-IN PROM VERSION

In addition to the mask ROM versions, the 4552 Group has the One Time PROM versions whose PROMs can only be written to and not be erased.

The built-in PROM version has functions similar to those of the mask ROM versions, but it has PROM mode that enables writing to built-in PROM.

Table 19 shows the product of built-in PROM version. Figure 61 shows the pin configurations of built-in PROM versions. The One Time PROM version has pin-compatibility with the mask ROM version.

Table 19 Product of built-in PROM version

| Part number | PROM size (X 10 bits) | RAM size (X 4 bits) | Package | ROM type |
|-------------|--------------------------|------------------------|---------|----------------------------------|
| M34552G8FP | 8192 words | 288 words | 48P6S-A | One Time PROM [shipped in blank] |
| M34552G8HFP | | | | |

(1) PROM mode

The 4552 Group has a PROM mode in addition to a normal operation mode. It has a function to serially input/output the command codes, addresses, and data required for operation (e.g., read and program) on the built-in PROM using only a few pins. This mode can be selected by muddog entry after powering on the VDD pin. In the PROM mode, three types of software commands (read, program, and program verify) can be used. Clock-synchronous serial I/O is used, beginning from the LSB (LSB first).

(2) Notes on handling

① For the One Time PROM version shipped in blank, Renesas corp. does not perform PROM writing test and screening in the assembly process and following processes. In order to improve reliability after writing, performing writing and test according to the flow shown in Figure 60 before using is recommended (Products shipped in blank: PROM contents is not written in factory when shipped).

(3) Difference between Mask ROM version and One Time PROM version

Mask ROM version and One Time PROM version have some difference of the following characteristics within the limits of an electrical property by difference of a manufacture process, built-in ROM, and a layout pattern.

- a characteristic value
- a margin of operation
- the amount of noise-proof
- noise radiation, etc.,

Accordingly, be careful of them when swithcing.

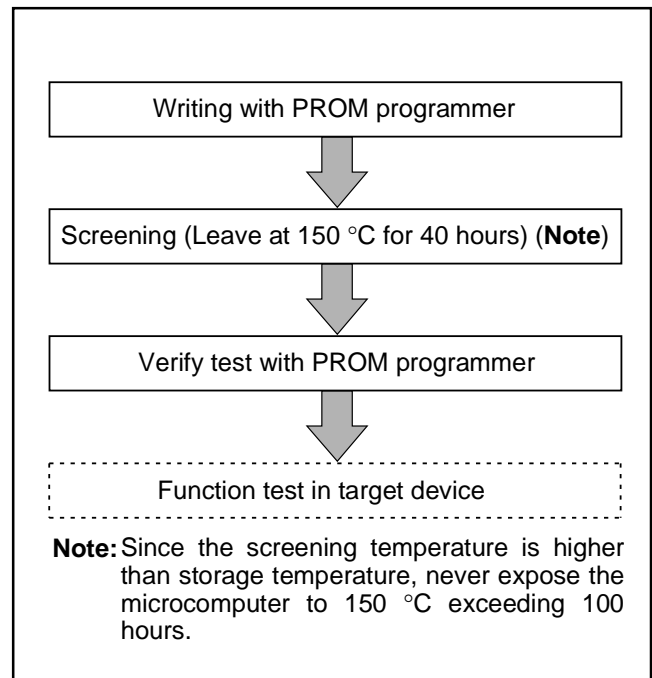


Fig. 68 Flow of writing and test of the product shipped in blank

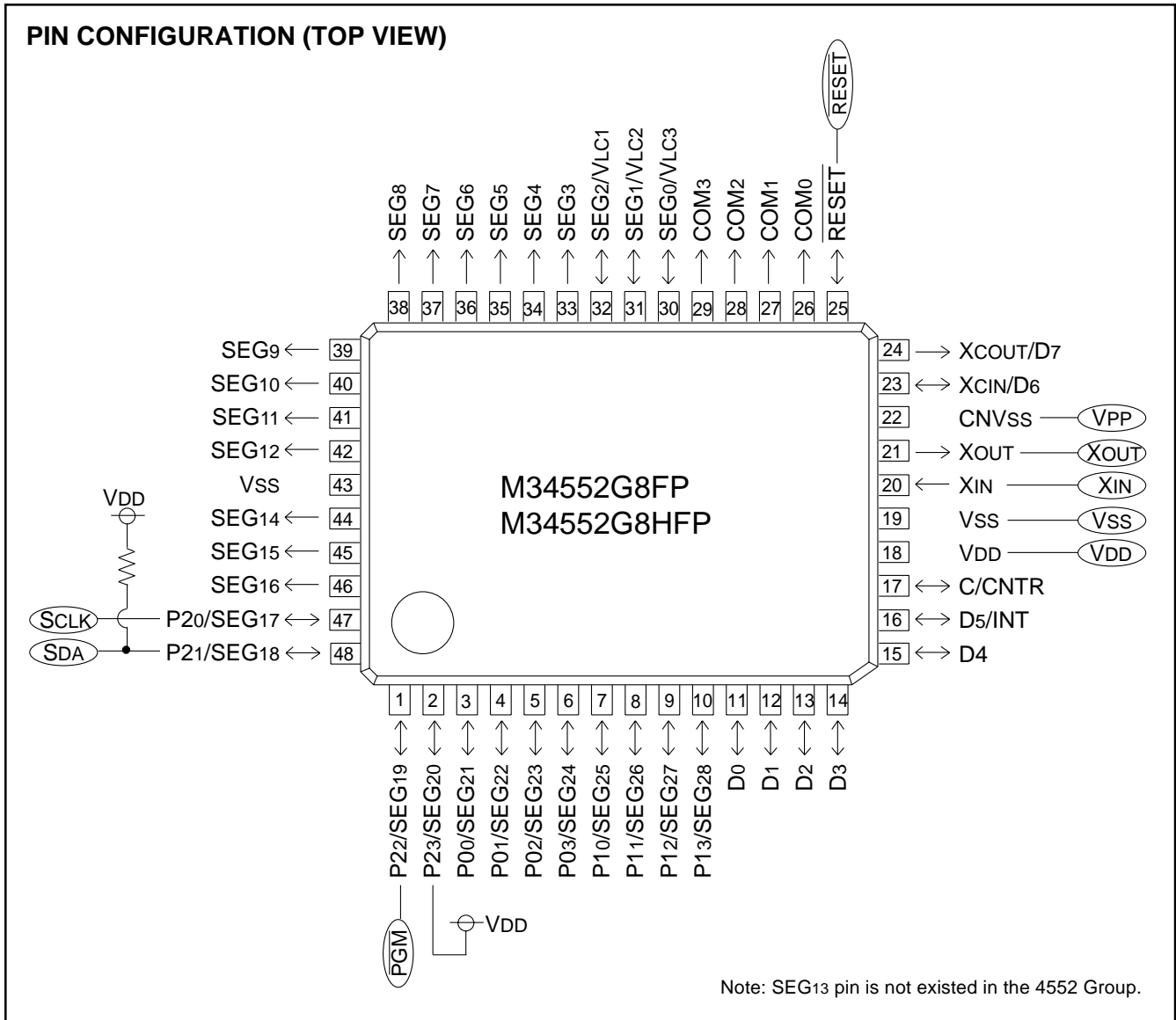


Fig. 69 Pin configuration of built-in PROM version

ROM CODE ACCESS PROTECTION

We would like to support a simple ROM code protection function that prevents a party other than the ROM-code owner to read and reprogram the built-in PROM code of the MCU.

First, Programmers must check the ID-code of the MCU.

If the ID-code is not blank, Programmer verifies it with the input ID-code. When the ID-codes do not match, Programmer will reject all further operations.

The MCU has each 10 bits of dedicated ROM spaces in address 0090₁₆ to 0096₁₆, as an ID-code (referred to as "the ID-code") enabling a Programmer to verify with the input ID-code and validate further operations.

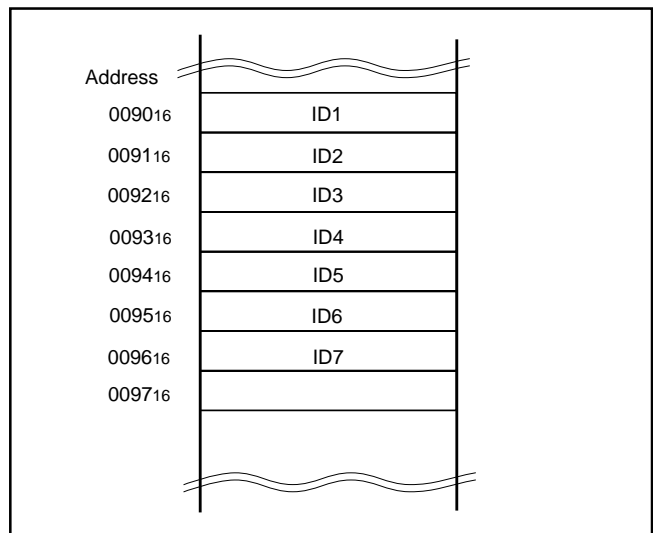


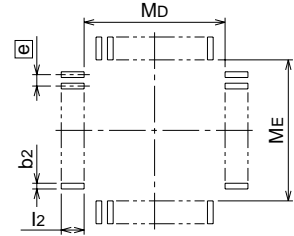
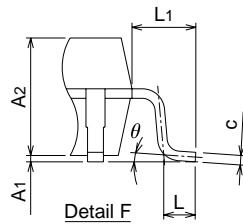
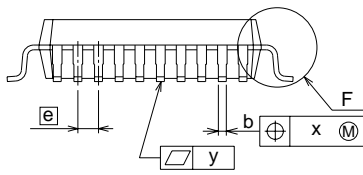
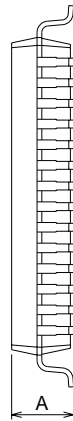
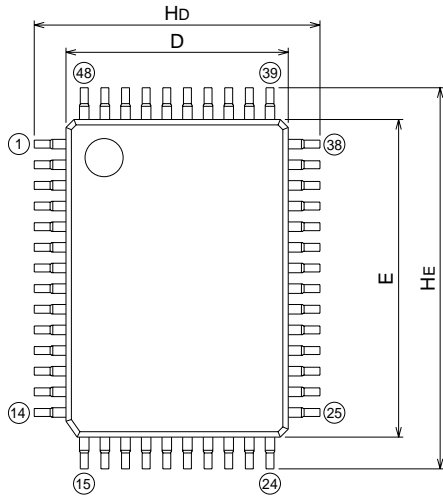
Fig. 70 ROM-Code Protection ID Location

PACKAGE OUTLINE

48P6S-A

Plastic 48pin 7X10mm body QFP

| | | | |
|-------------------|------------|-----------|---------------|
| EIAJ Package Code | JEDEC Code | Weight(g) | Lead Material |
| QFP48-P-710-0.65 | - | 0.29 | Alloy 42 |



Recommended Mount Pad

| Symbol | Dimension in Millimeters | | |
|----------|--------------------------|------|------|
| | Min | Nom | Max |
| A | - | - | 2.15 |
| A1 | 0 | 0.1 | 0.2 |
| A2 | - | 1.85 | - |
| b | 0.2 | 0.25 | 0.35 |
| c | 0.13 | 0.15 | 0.2 |
| D | 6.8 | 7.0 | 7.2 |
| E | 9.8 | 10.0 | 10.2 |
| e | - | 0.65 | - |
| HD | 8.7 | 9.0 | 9.3 |
| HE | 11.7 | 12.0 | 12.3 |
| L | 0.3 | 0.5 | 0.7 |
| L1 | - | 1.0 | - |
| x | - | - | 0.13 |
| y | - | - | 0.1 |
| θ | 0° | - | 10° |
| b2 | - | 0.35 | - |
| l2 | 1.0 | - | - |
| MD | - | 7.4 | - |
| ME | - | 10.4 | - |

REVISION HISTORY

4552 Group Data Sheet

| Rev. | Date | Description | |
|------|---------------|---|---|
| | | Page | Summary |
| 1.00 | Jul. 23, 2003 | – | First edition issued |
| 1.01 | Sep. 17, 2003 | 50 51 61 128 | Voltage drop detection circuit (only in H version) revised. Table 15 revised. Timer functions, Timer control registers, Port level, and Notes 6 and 7) ⑰ Voltage drop detection circuit (only in H version) revised. Fig.57 revised. |
| 2.00 | Feb. 24, 2004 | 1 2 4 29 48 50 58 61 120 to 132 135 | FEATURES: ● Minimum instruction execution time: time for One Time PROM version added. ● Supply voltage of One Time PROM version revised. PIN CONFIGURATION: Note added. PERFORMANCE OVERVIEW: Minimum instruction execution time: time for One Time PROM version added. Supply voltage of One Time PROM version revised. Power dissipation: Values only for Mask ROM version are listed. Table 9: Timer 3; Count source and Use of output signal revised. (1) Power-on reset : “(only for H version)” eliminated. Description revised. Fig.37: “(only for H version)” added to Voltage drop detection circuit. Fig.40: Note revised. ROM ORDERING METHOD revised. Note on ⑱ Power-on reset : revised. ELECTRICAL CHARACTERISTICS revised. The table is separated to Mask ROM version and One Time PROM version. Supply voltage and supply current revised mainly. Note 6 is added to VOLTAGE DTOP DETECTION CIRCUIT CHARACTERISTICS. Fig.57: Note added. |
| 3.00 | Jul. 09, 2004 | All pages 5 31 39 40 46 47 49 61 128 | Words standardized: On-chip oscillator Description of RESET pin revised. Fig.23: Note added. Some description revised. Fig.28: "DI" instruction added. (5) LCD power supply circuit ● Internal dividing resistor revised. Fig.34 d): “VLC3, VLC2, VLC1” added. Fig.35, Fig.36: Count revised. Fig.38: State of quartz-crystal oscillator added. Note on Power Source Voltage added. RECOMMENDED OPERATING CONDITIONS 1 VDD (RC oscillation) Max.: 3.6 |

REVISION HISTORY

4552 Group Data Sheet

| Rev. | Date | Description | |
|------|---------------|---|--|
| | | Page | Summary |
| 3.01 | Jun.15, 2005 | All pages 36 61 | Delete the following: "PRELIMINARY". •Prescaler and Timer 1 count start timing and count time when operation starts, •Timer 2 and Timer LC count start timing and count time when operation starts added. ⑬ Prescaler and Timer 1 count start timing and count time when operation starts, ⑭ Timer and Timer LC count start timing and count time when operation starts added. |
| 3.02 | Dec. 22, 2006 | 29, 33 30, 31 31 32, 69 33 34 48 52 54 55, 73 60 to 63 64 77, 120, 121 93 132 132, 138 → | Use of output signal of prescaler: LC eliminated. Fig.22, Fig.23: Note added. Fig.23: INSTCK (wrong) → INTSNC (correct) PA0: Stop (state <u>initialized</u>) → (state <u>retained</u>) W31 W30: Timer 3 count <u>source</u> selection bits → Timer 3 count <u>value</u> selection bits (2) Prescaler (interrupt function): PRS (wrong) → RPS (correct) (5) Timer 3 (interrupt function): Description added. Fig.37: Clock (wrong) → f(RING) (correct) Table 15 Timer 3 function (RAM back-up): 0 → (Note 3) Timer interrupt request flag (RAM back-up): 0 → (Note 3) Fig.44: Note 1 added. Table 17: Notes 2 and 3 added. NOTES ON NOISE added. ① Noise and latch-up prevention: Description added. SZD: (Y) = 0 to <u>7</u> → 0 to <u>5</u> SZD: Detailed description revised. VRST ⁻ , VRST ⁺ : Test condition revised. Note 4: (<u>power</u> current) → (<u>supply</u> current) Pages 16 to 18, 20, 27, 54, 66: RAM back-up mode → power down mode Pages 77, 90 to 92, 116 to 119: SNZ0, SNZT1, SNZT2, SNZT3 revised. Pages 78, 109, 122, 123: WRST revised. |
| | | | |

Notes:

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



RENESAS SALES OFFICES

<http://www.renesas.com>

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

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

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

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

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

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

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

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