

To all our customers

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

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

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

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

MITSUBISHI MICROCOMPUTERS 4250 Group

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

DESCRIPTION

The 4250 Group is a 4-bit single-chip microcomputer designed with CMOS technology. Its CPU is that of the 720 series using a simple instruction set. The computer is equipped with one 8-bit timer which has a reload register and the interrupt function. The various microcomputers in the 4250 Group include variations of the built-in memory type as shown in the table below.

FEATURES

- Minimum instruction execution time 1.0 μ s (at 4.0 MHz system clock frequency, V_{DD} =4.5 V to 5.5 V)
- Supply voltage
 - 4.5 V to 5.5 V (at 4.0 MHz system clock frequency)
 - 2.5 V to 5.5 V (at 1.0 MHz system clock frequency)
 - 2.2 V to 5.5 V (at 1.0 MHz system clock frequency: only for Mask ROM version)

- Timer
 - Timer 1 8-bit timer with a reload register
- Interrupt 2 sources
- CR oscillation circuit (Capacitor and Resistor connected externally)
- Logic operation instruction
- RAM back-up function
- Key-on wakeup function (ports G and S, INT pin)

APPLICATION

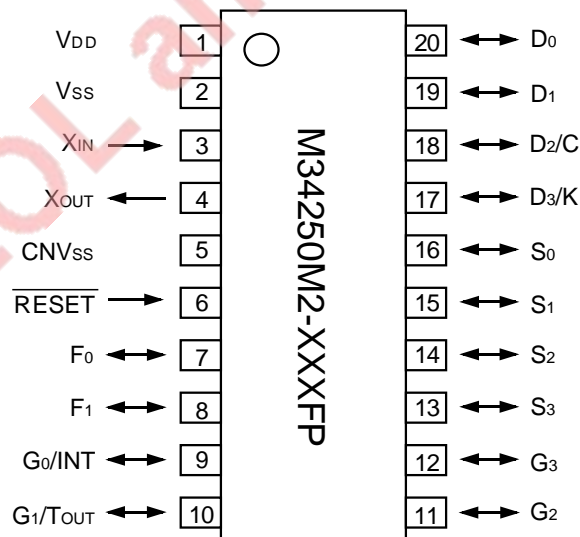
Electric household appliances, consumer electronics products (mouse, etc.)

| Product | ROM (PROM) size (X 9 bits) | RAM size (X 4 bits) | Package | ROM type |
|------------------|-------------------------------|------------------------|---------|---------------|
| M34250M2-XXXFP | 2048 words | 64 words | 20P2N-A | Mask ROM |
| M34250E2-XXXFP * | 2048 words | 64 words | 20P2N-A | One Time PROM |

*: Shipped after writing (shipped in blank: M34250E2FP)

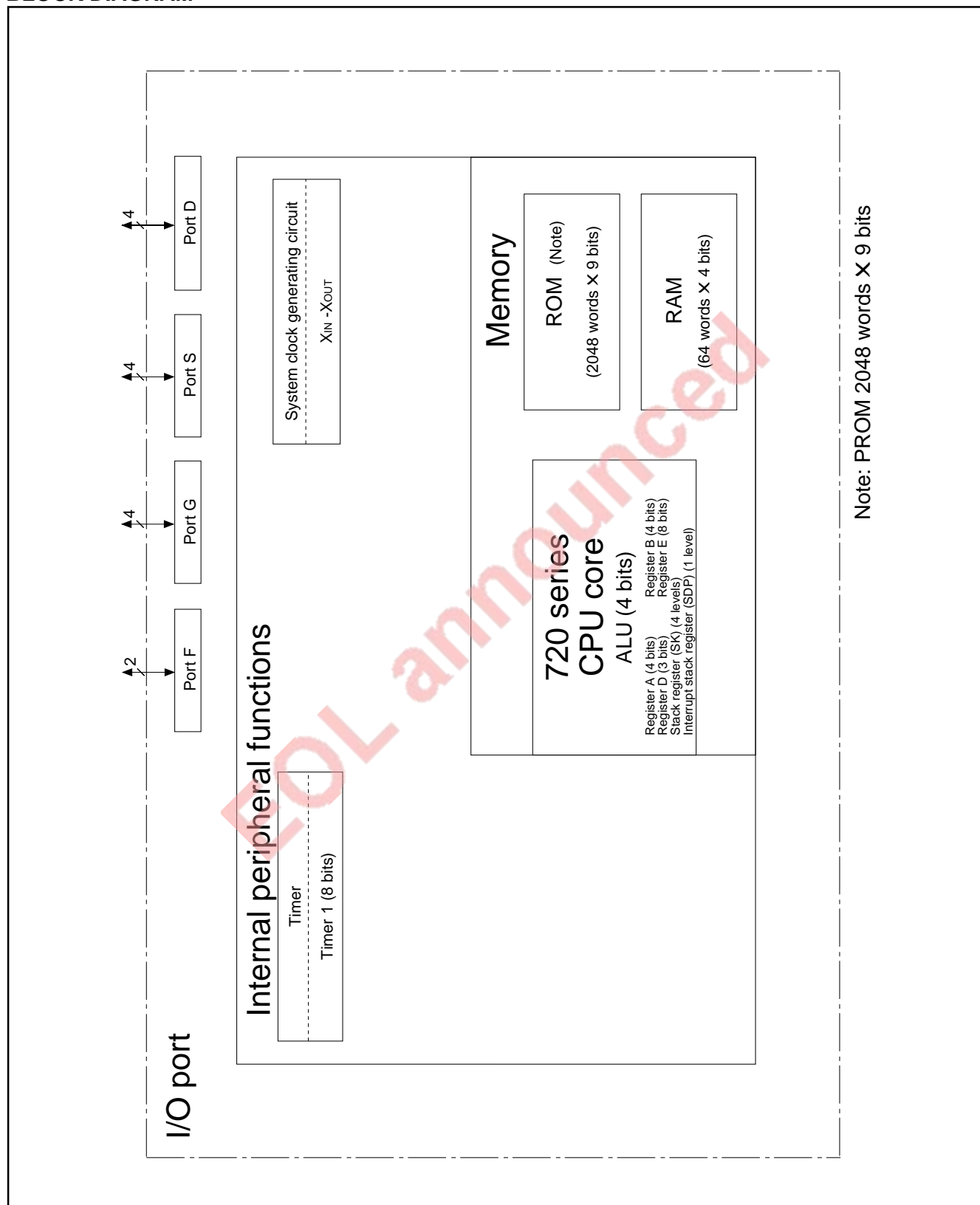
PIN CONFIGURATION (TOP VIEW)

M34250M2-XXXFP



Outline 20P2N-A

BLOCK DIAGRAM



PERFORMANCE OVERVIEW

| Parameter | | Function |
|------------------------------------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Number of basic instructions | | 70 |
| Minimum instruction execution time | | 1.0 μ s (at 4.0 MHz system clock frequency) (Refer to the electrical characteristics because the minimum instruction execution time depends on the supply voltage.) |
| Memory sizes | ROM | M34250M2/ 2048 words X 9 bits |
| | RAM | E2 64 words X 4 bits |
| Input/Output ports | D ₀ -D ₃ | I/O Four independent I/O ports; ports D ₂ and D ₃ are also used as ports C and K, respectively. |
| | S ₀ -S ₃ | I/O 4-bit I/O port |
| | C | I/O 1-bit I/O port; port C is also used as port D ₂ . |
| | K | I/O 1-bit I/O port; port K is also used as port D ₃ . |
| | F ₀ , F ₁ | I/O 2-bit I/O port |
| | G ₀ -G ₃ | I/O 4-bit I/O port; ports G ₀ and G ₁ are also used as pins INT and TOUT. |
| | INT | Input Interrupt input; INT pin is also used as port G ₀ . |
| | TOUT | Output Timer output; TOUT pin is also used as port G ₁ . |
| Timer | Timer 1 | 8-bit timer with a reload register |
| Interrupt | Sources | 2 (one for external and one for timer) |
| | Nesting | 1 level |
| Oscillation circuit | | CR oscillation circuit (a capacitor and a resistor connected externally) Frequency error: $\pm 17\%$ (V _{DD} = 5 V $\pm 10\%$, V _{DD} = 3 V $\pm 10\%$, the error of the external capacitor and resistor excluded) |
| Subroutine nesting | | 4 levels |
| Device structure | | CMOS silicon gate |
| Package | | 20-pin plastic molded SOP (20P2N-A) |
| Operating temperature range | | -20 °C to 85 °C |
| Supply voltage | | 2.2 V to 5.5 V (Refer to the electrical characteristics because the supply voltage depends on the system clock frequency.) |
| Power dissipation (typical value) | Active mode | 1.5 mA (at 4.0 MHz system clock frequency, V _{DD} = 5 V, output transistors in the cut-off state) |
| | RAM back-up mode | 0.1 μ A (at room temperature, V _{DD} = 5 V, output transistors in the cut-off state) |

PIN DESCRIPTION

| Pin | Name | Input/Output | Function |
|---------------------------------|---------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| V _{DD} | Power supply | — | Connected to a plus power supply. |
| V _{SS} | Ground | — | Connected to a 0 V power supply. |
| CNV _{SS} | CNV _{SS} | — | Connect CNV _{SS} to V _{SS} and apply "L" (0V) to CNV _{SS} certainly. |
| RESET | Reset input | Input | Reset pulse input pin |
| X _{IN} | System clock input | Input | I/O pins of the system clock generating circuit. Connect pins X _{IN} and X _{OUT} directly. Then, pull up X _{IN} pin through a resistor and pull down X _{OUT} pin through a capacitor. |
| X _{OUT} | System clock output | Output | |
| F ₀ , F ₁ | I/O port F | I/O | 2-bit I/O port; for input use, set the latch of the specified bit to "1." The output structure is N-channel open-drain. |
| G ₀ –G ₃ | I/O port G | I/O | 4-bit I/O port. For input use, set the latch of the specified bit to "1." The output structure is N-channel open-drain. Every pin of the ports has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports G ₀ and G ₁ are also used as pins INT and T _{OUT} , respectively. |
| S ₀ –S ₃ | I/O port S | I/O | 4-bit I/O port. For input use, set the latch of the specified bit to "1." The output structure is N-channel open-drain. Every pin of the ports has a key-on wakeup function which can be switched by software. Also, it is used to perform the logic operation using register A. |
| D ₀ –D ₃ | I/O port D | I/O | Each pin of port D has an independent 1-bit wide I/O function. For input use, set the latch of the specified bit to "1." The output structure is N-channel open-drain. Ports D ₂ and D ₃ are also used as ports C and K, respectively. |
| C | I/O port C | I/O | 1-bit I/O port. For input use, set the latch of the specified bit to "1." The output structure is N-channel open-drain. Port C has a pull-up function which can be switched by software. It is also used as port D ₂ . |
| K | I/O port K | I/O | 1-bit I/O port. For input use, set the latch of the specified bit to "1." The output structure is N-channel open-drain. Port K has a pull-up function which can be switched by software. It is also used as port D ₃ . |
| T _{OUT} | Timer output | Output | T _{OUT} pin has the function to output the timer 1 underflow signal divided by 2. It is also used as port G ₁ . |
| INT | Interrupt input | Input | INT pin accepts an external interrupt. It also accepts the input signal to return the system from the RAM back-up state. It is also used as port G ₀ . |

MULTIFUNCTION

| Pin | Multifunction | Pin | Multifunction |
|----------------|------------------|---------------------------|----------------|
| G ₀ | INT | INT (Note 2) | G ₀ |
| G ₁ | T _{OUT} | T _{OUT} (Note 2) | G ₁ |
| D ₂ | C | C (Note 2) | D ₂ |
| D ₃ | K | K (Note 2) | D ₃ |

Notes 1: Pins except above have just single function.

2: The I/O of ports D₂, D₃ and G₀, and the input of port G₁ can be used even when ports C and K and pins INT and T_{OUT} are selected.

CONNECTIONS OF UNUSED PINS

| Pin | Connection | Pin | Connection |
|-------------------------------------------------------|--------------------------------------------------|--------------------------------------|--------------------------------------------------|
| F ₀ , F ₁ | Connect to V _{SS} pin. | D ₀ , D ₁ | Connect to V _{SS} pin. |
| G ₀ /INT, G ₁ /T _{OUT} | Open or connect to V _{SS} pin. (Note 1) | D ₂ /C, D ₃ /K | Open or connect to V _{SS} pin. (Note 3) |
| G ₂ , G ₃ | | | |
| S ₀ –S ₃ | Connect to V _{SS} pin. (Note 2) | | |

Notes 1: When pins G₀/INT, G₁/T_{OUT}, G₂ and G₃ are connected to V_{SS} pin, turn off their pull-up transistors (Pull-up control register PU0="X02") and also invalidate the key-on wakeup functions of pins G₁/T_{OUT}, G₂ and G₃ (Key-on wakeup control register K0="XX0X2") by software. When the POF instruction is executed while these pins are connected to V_{SS} and the key-on wakeup functions are left valid, the system returns from RAM back-up state by recognizing the return condition immediately after going into the RAM back-up state. When these pins are open, turn on their pull-up transistors (Pull-up control register PU0="X12") by software.

2: When ports S₀–S₃ are connected to V_{SS} pin, invalidate the key-on wakeup functions (Key-on wakeup control register K0="XXX02") by software. When the POF instruction is executed while these pins are connected to V_{SS} and the key-on wakeup functions are left valid, the system returns from RAM back-up state by recognizing the return condition immediately after going into the RAM back-up state.

3: When ports D₂/C and D₃/K are connected to V_{SS} pin, turn off their pull-up transistors (register PU0="0X2") by software. When these pins are open, turn on their pull-up transistors (register PU0="1X2") by software.

(Note when connecting to V_{SS} and V_{DD})

- Connect the unused pins to V_{SS} or V_{DD} at the shortest distance and use the thick wire against noise.

PORT FUNCTION

| Port | Pin | Input/ Output | Output structure | Control bits | Control instructions | Control registers | Remark | | |
|--------|---------------------------------|------------------|----------------------|--------------|-------------------------------------------------------------|-------------------|--------------------------------------------------------------------------------------------|---------------|-------------------------------------------|
| Port D | D ₀ , D ₁ | I/O (4) | N-channel open-drain | 1 | SD RD SZD CLD SCP RCP SNZCP OKA IAK | PU0 | Pull-up function (programmable) | | |
| | D ₂ /C | | | | | | | | |
| | D ₃ /K | | | | | | | | |
| Port S | S ₀ –S ₃ | I/O (4) | N-channel open-drain | 4 | OSA IAS LGOP | K0 LO | Logic operation function (programmable) Key-on wakeup functions (programmable) | | |
| Port G | G ₀ /INT | I/O (4) | N-channel open-drain | 4 | OGA IAG | PU0, K0 | Pull-up functions Key-on wakeup functions (only pull-up function is programmable) | | |
| | G ₁ /TOUT | | | | | | | PU0, K0 V1 | Pull-up functions (programmable) |
| | G ₂ , G ₃ | | | | | | | PU0, K0 | Key-on wakeup functions (programmable) |
| Port F | F ₀ , F ₁ | I/O (2) | N-channel open-drain | 2 | OFA IAF | | | | |

DEFINITION OF CLOCK AND CYCLE

- System clock
This is the source clock input to the X_{IN} pin. Connect pins X_{IN} and X_{OUT} directly. Then, pull up X_{IN} pin through a resistor and pull down X_{OUT} pin through a capacitor.
- Instruction clock
The instruction clock is a signal derived by dividing the system clock by 4, and is the basic clock for controlling this product.
- Machine cycle
One machine cycle is the time required to execute the minimum instruction (one-cycle instruction). The machine cycle is equivalent to the instruction clock cycle.

I/O PORT

(1) Port D (D₀–D₃)

Each pin of port D has an independent 1-bit wide I/O function. Each pin has an output latch. For input/output of ports D₀–D₃, select one of port D with the register Y of data pointer first. For input use, set the latch of the specified bit to “1.” All port D output latches can be set to “1” with the CLD instruction. The output structure is the N-channel open-drain. Ports D₂ and D₃ are also used as ports C and K, respectively. Accordingly, when port D₂/C is used as port D₂, set the port C output latch to “1.” When port D₃/K is used as port D₃, set the port K output latch to “1.”

(2) Port C

1-bit I/O port.
Port C output latch can be set to “1” with the SCP instruction. Port C output latch can be cleared to “0” with the RCP instruction. Port C input level can be examined by executing the skip (SNZCP) instruction. For input use, set the latch of the specified bit to “1.” The output structure is the N-channel open-drain. The pull-up transistor of port C is turned on when the bit 1 of register PU0 is set to “1” by software. Port C is also used as port D₂. Accordingly, when port D₂/C is used as port C, set the port D₂ output latch to “1.”

(3) Port K

1-bit I/O port.
For input use, set the latch of the specified bit to “1.” The output structure is the N-channel open-drain. The pull-up transistor of port K is turned on when the bit 1 of register PU0 is set to “1” by software. Port K is also used as port D₃. Accordingly, when port D₃/K is used as port K, set the port D₃ output latch to “1.”

(4) Port G (G₀–G₃)

4-bit I/O port.
For input use, set the latch of the specified bit to “1.” The output structure is the N-channel open-drain. The pull-up transistor of port G is turned on when the bit 0 of register PU0 is set to “1” by software. Ports G₀ and G₁ are also used as INT pin and TOUT pin, respectively.

Pull-up control register

| Pull-up control register PU0 | | at reset : 00 ₂ | at RAM back-up : state retained | W |
|------------------------------|--------------------------------------|----------------------------|---------------------------------|---|
| PU0 ₁ | Ports C and K | 0 | Pull-up transistor OFF | |
| | pull-up transistor control bit | 1 | Pull-up transistor ON | |
| PU0 ₀ | Ports G ₀ –G ₃ | 0 | Pull-up transistor OFF | |
| | pull-up transistor control bit | 1 | Pull-up transistor ON | |

Note: “W” represents write enabled.

(5) Port F (F₀, F₁)

2-bit I/O port.

For input use, set the latch of the specified bit to "1." The output structure is the N-channel open-drain.

(6) Port S (S₀–S₃)

4-bit I/O port.

Port S has the logic operation (LGOP) function. For input (logic operation included) use, set the latch of the specified bit to "1." The output structure is the N-channel open-drain. When performing the logic operation, select the logic operation function with the logic operation selection register LO. Set the contents of register LO through register A with the TLOA instruction.

When the LGOP instruction is executed, the logic operation selected with the register LO is performed between the contents of register A and the contents of port S, and its result is stored in register A.

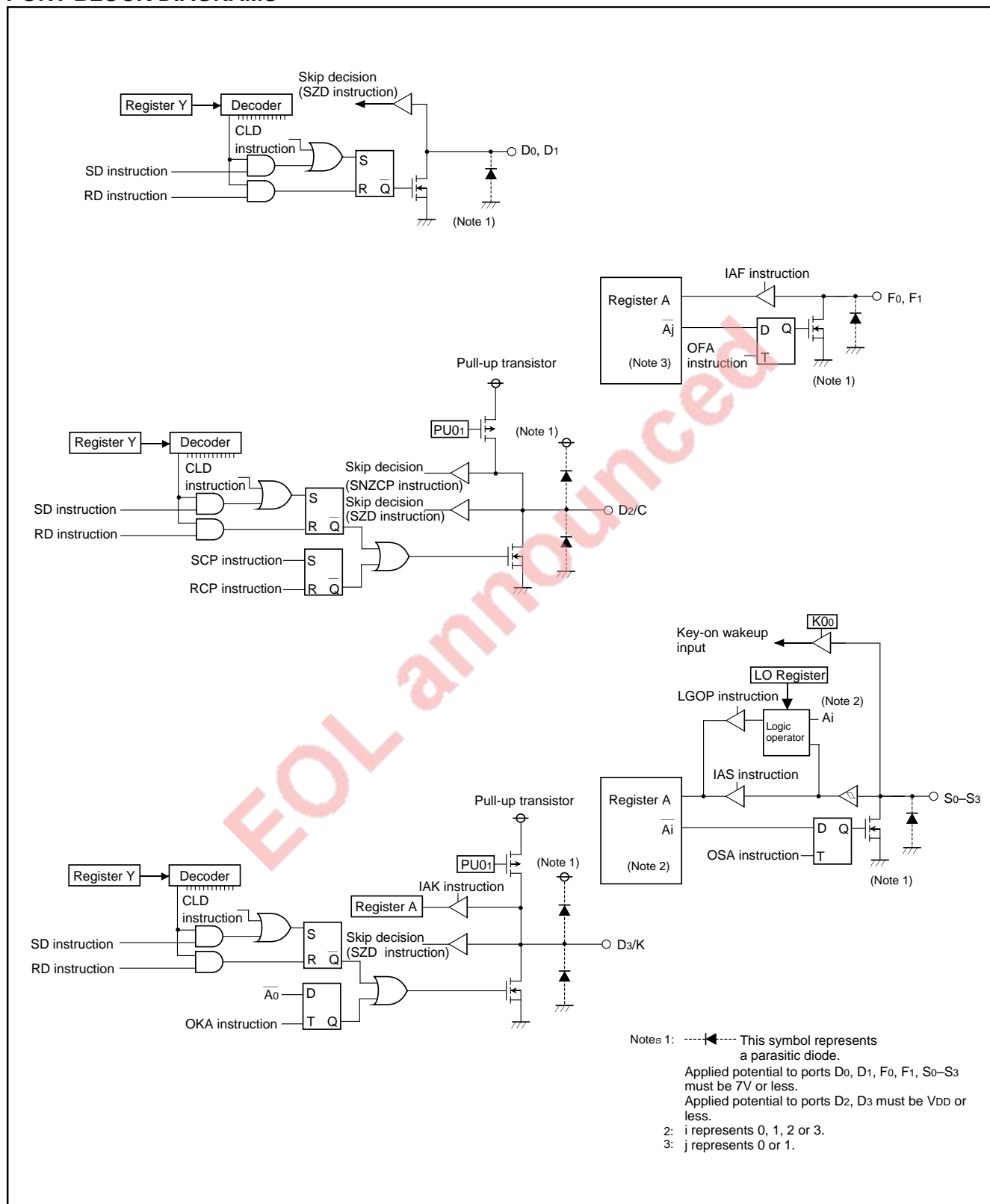
Logic operation selection register

| Logic operation selection register LO | | at reset : 00 ₂ | | at RAM back-up : 00 ₂ | W |
|---------------------------------------|-----------------------------------------|----------------------------|-----------------|----------------------------------|---|
| LO ₁ | Logic operation function selection bits | LO ₁ | LO ₀ | Functions | |
| | | 0 | 0 | XOR operation | |
| | | 0 | 1 | OR operation | |
| LO ₀ | | 1 | 0 | AND operation | |
| | | 1 | 1 | Not available | |

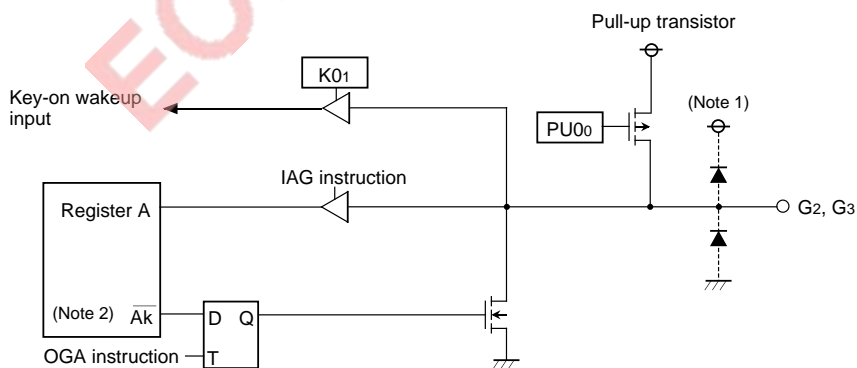
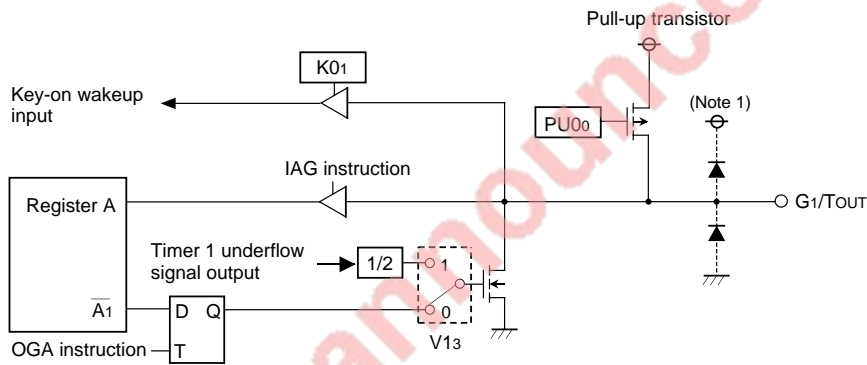
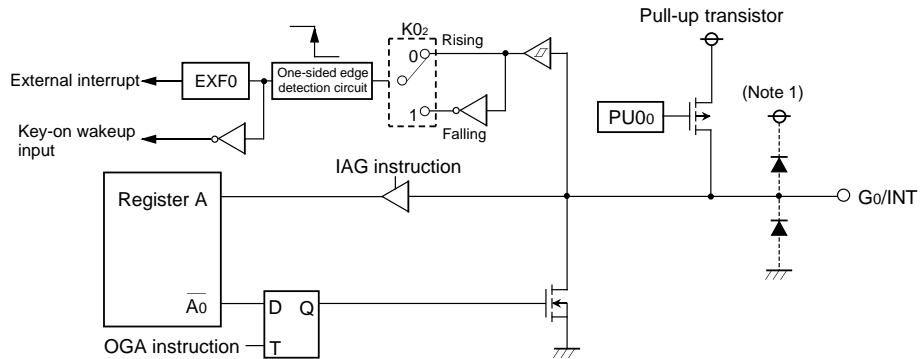
Note: "W" represents write enabled.

EOL announced

PORT BLOCK DIAGRAMS



PORT BLOCK DIAGRAMS (CONTINUED)



Notes 1: This symbol represents a parasitic diode.
Applied potential to ports G0-G3 must be V_{DD} or less.
2: k represents 2 or 3.

**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 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).

(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).

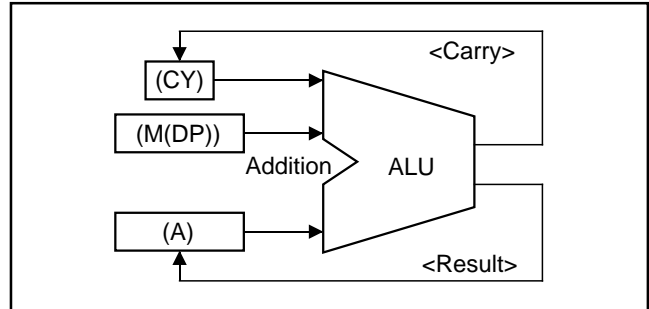


Fig. 1 AMC instruction execution example

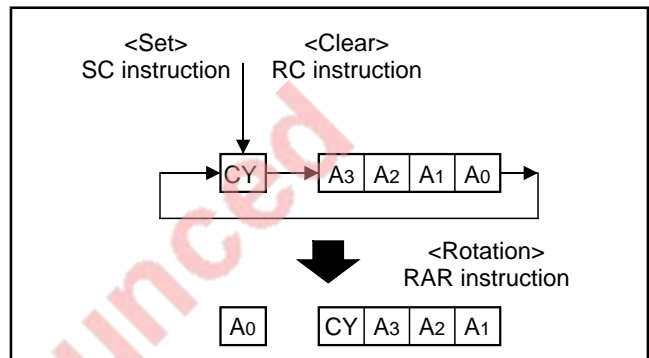


Fig. 2 RAR instruction execution example

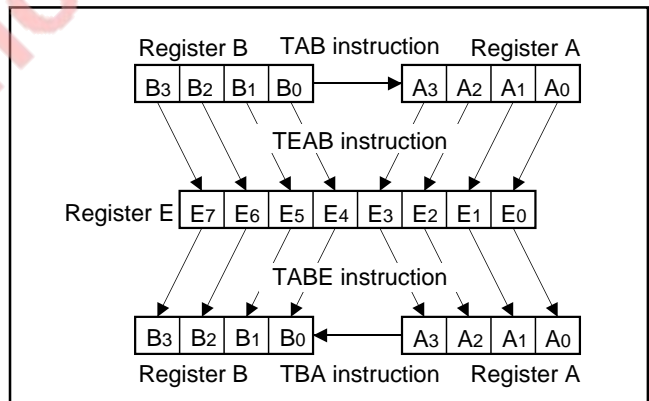


Fig. 3 Registers A, B and register E

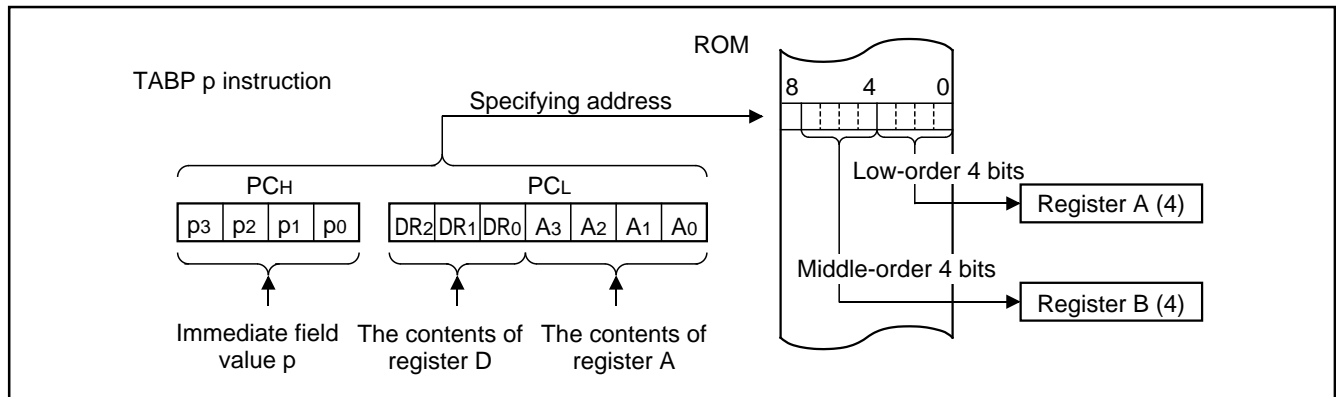


Fig. 4 TABP p instruction execution example

(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 four identical registers, so that subroutines can be nested up to 4 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 4 levels are exceeded.

The register SK nesting level is pointed automatically by 2-bit stack pointer (SP).

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 and skip flag 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.

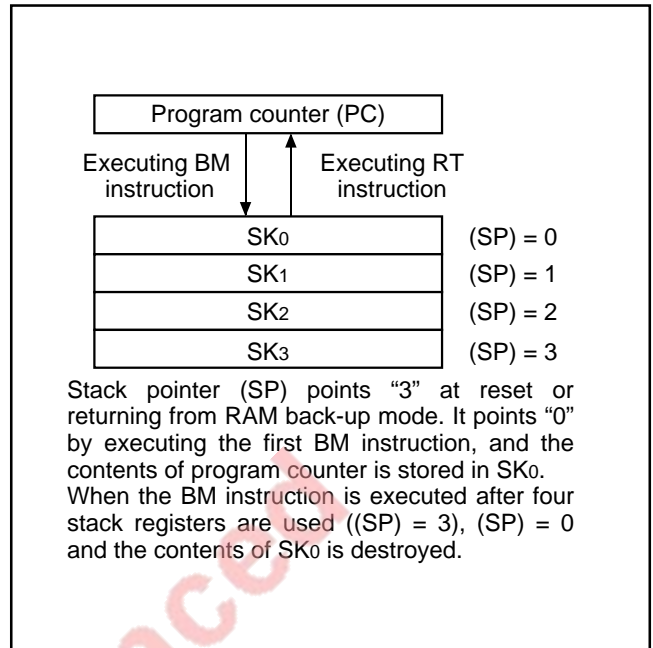


Fig. 5 Stack registers (SKs) structure

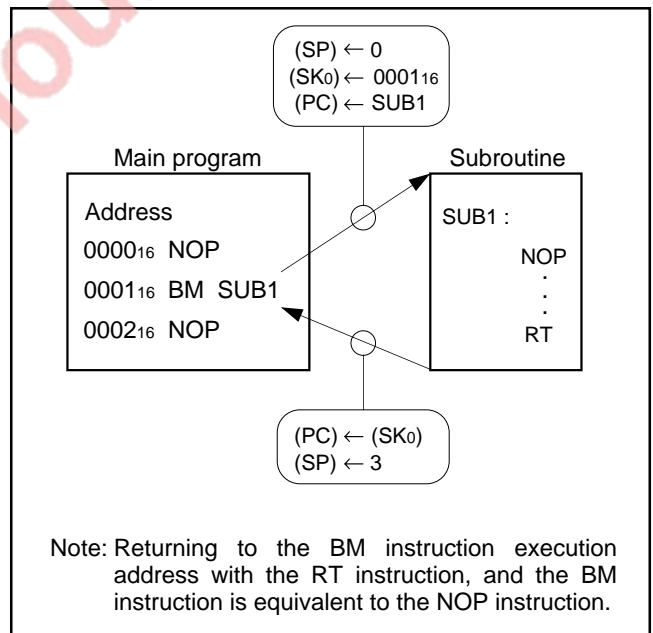


Fig. 6 Example of operation at subroutine call

Note: Returning to the BM instruction execution address with the RT instruction, and the BM instruction is equivalent to the NOP instruction.

(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 PC_H (most significant bit to bit 7) which specifies to a ROM page and PC_L (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 PC_H does not exceed after the last page of the built-in ROM.

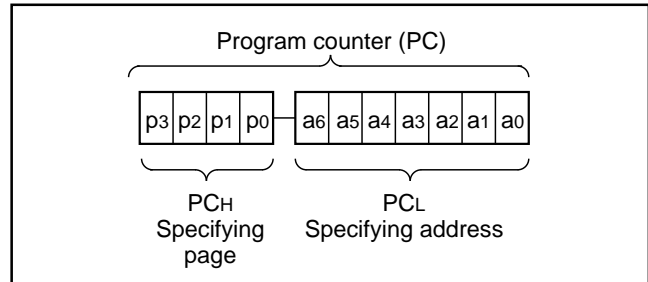


Fig. 7 Program counter (PC) structure

(9) Data pointer (DP)

Data pointer (DP) is used to specify a RAM address and consists of registers X and Y. 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).

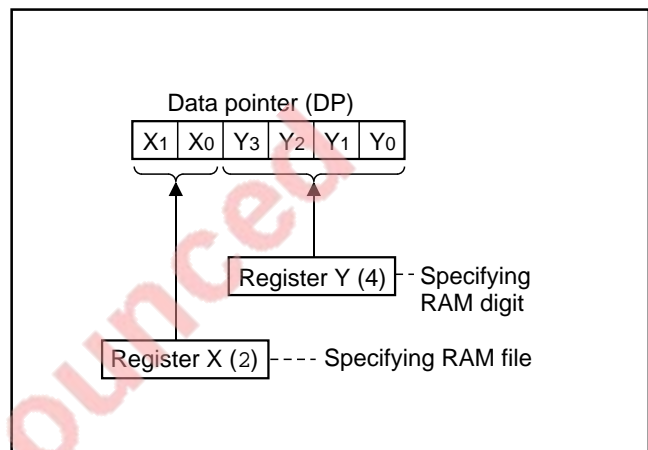


Fig. 8 Data pointer (DP) structure

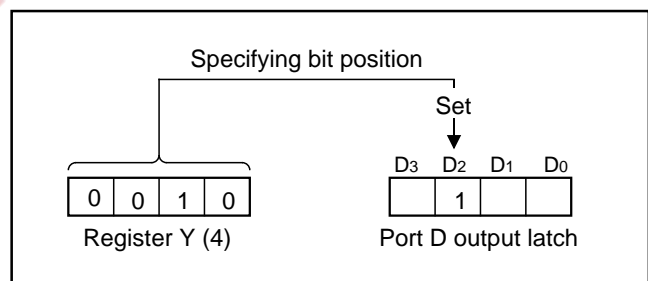


Fig. 9 SD instruction execution example

PROGRAM MEMORY (ROM)

The program memory is a mask ROM. 1 word of ROM is composed of 9 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 M34250M2.

Table 1 ROM size and pages

| Product | ROM size (X 9 bits) | Pages |
|----------|---------------------|--------------|
| M34250M2 | 2048 words | 16 (0 to 15) |
| M34250E2 | | |

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.

DATA MEMORY (RAM)

1 word of RAM is composed of 4 bits, but 1-bit manipulation (with the SB j, RB j, and SZB j instructions) is enabled for the entire memory area. A RAM address is specified by a data pointer. The data pointer consists of registers X and Y. Set a value to the data pointer certainly when executing an instruction to access RAM.

Table 2 shows the RAM size. Figure 12 shows the RAM map.

Table 2 RAM size

| Product | RAM size |
|----------|------------------------------|
| M34250M2 | 64 words X 4 bits (256 bits) |
| M34250E2 | |

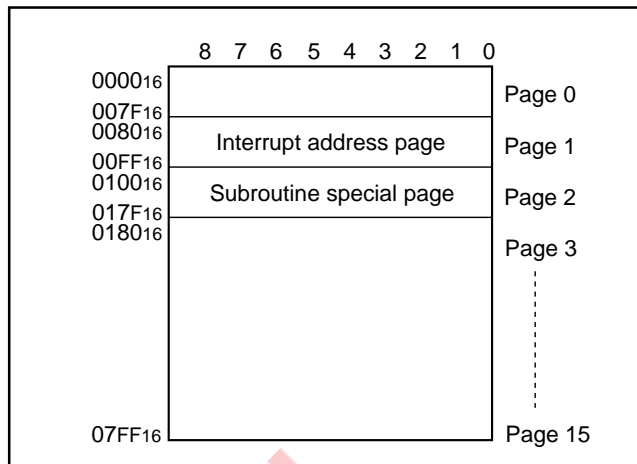


Fig. 10 ROM map of M34250M2

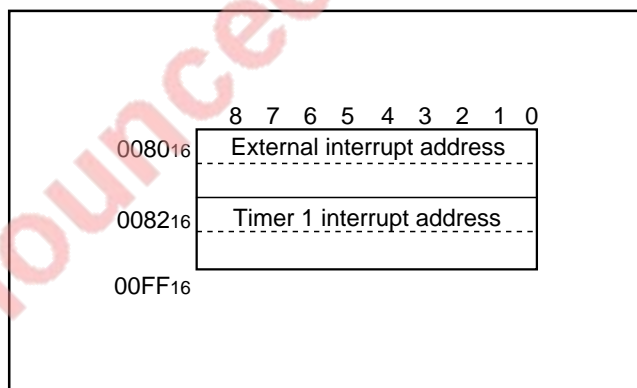


Fig. 11 Page 1 (addresses 0080₁₆ to 00FF₁₆) structure

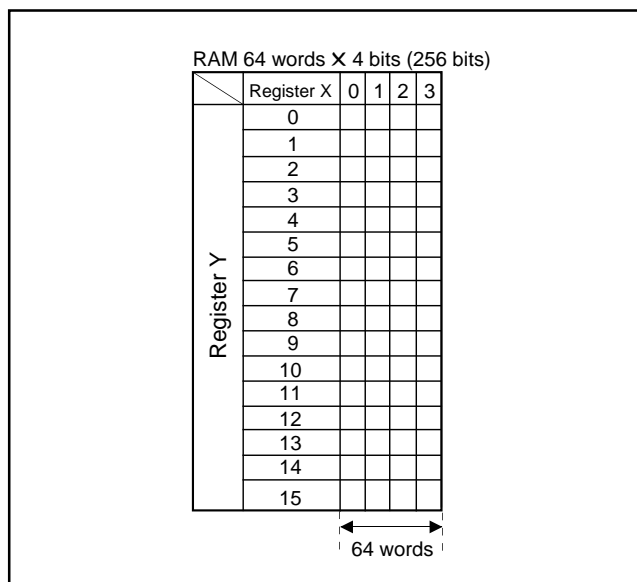


Fig. 12 RAM map

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 = "1" (interrupt request occurrence enabled)
- Interrupt enable flag (INTE) = "1" (interrupt enabled)

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 (V1₀, V1₁)

Use an interrupt enable bit of interrupt control register V1 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 interrupt | Level change of INT pin | Address 0 in page 1 |
| 2 | Timer 1 interrupt | Timer 1 underflow | Address 2 in page 1 |

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

| Interrupt name | Request flag | Enable bit | Skip instruction |
|--------------------|--------------|-----------------|------------------|
| External interrupt | EXF0 | V1 ₀ | SNZ0 |
| Timer 1 interrupt | T1F | V1 ₁ | SNZ1 |

Table 5 Interrupt enable bit function

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

(6) Control register related to interrupt

- Timer control register V1
 Interrupt enable bits of external and timer 1 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.

Table 6 Control register related to interrupt

| Timer control register V1 | | at reset : 0000 ₂ | at RAM back-up : 0000 ₂ | R/W |
|---------------------------|---------------------------------------|------------------------------|----------------------------------------------------------------|-----|
| V13 | G1/TOUT pin function selection bit | 0 | Port G1 (I/O) | |
| | | 1 | TOUT pin (output)/port G1(input) | |
| V12 | Prescaler/timer 1 operation start bit | 0 | Prescaler stop (initial state) / timer 1 stop (state retained) | |
| | | 1 | Prescaler / timer 1 operation | |
| V11 | Timer 1 interrupt enable bit | 0 | Interrupt disabled (SNZ1 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZ1 instruction is invalid) | |
| V10 | External interrupt enable bit | 0 | Interrupt disabled (SNZ0 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZ0 instruction is invalid) | |

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

(7) Interrupt sequence

Interrupts occur only when the respective INTE flag, interrupt enable bits (V10, V11), and interrupt request flags (EXF0, T1F) 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).

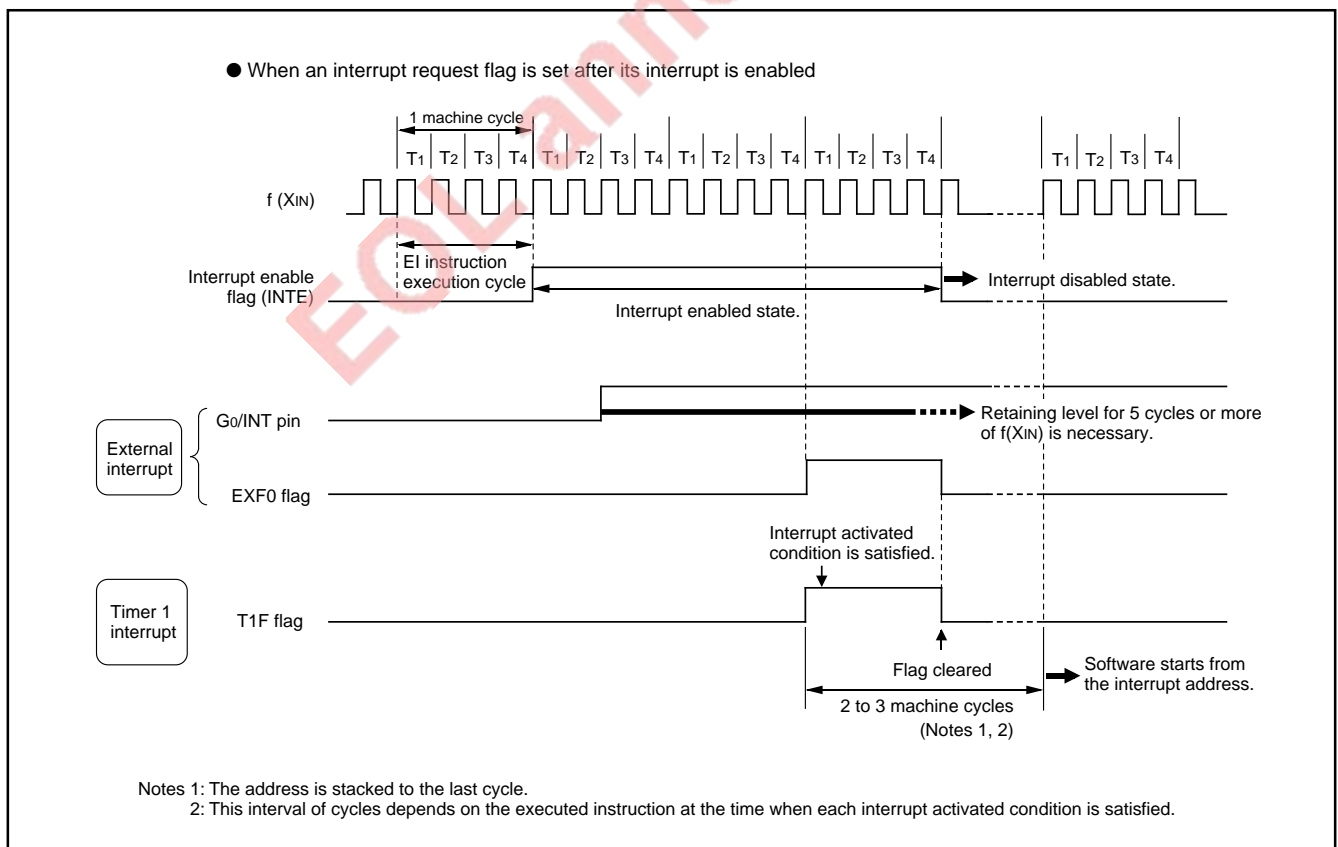


Fig. 16 Interrupt sequence

EXTERNAL INTERRUPTS

The 4250 Group has an external 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 key-on wakeup control register K0.

Table 7 External interrupt activated condition

| Name | Input pin | Valid waveform | Valid waveform selection bit(K0 ₂) |
|--------------------|-----------|----------------------------|------------------------------------------------|
| External interrupt | Go/INT | Falling waveform ("H"→"L") | 1 |
| | | Rising waveform ("L"→"H") | 0 |

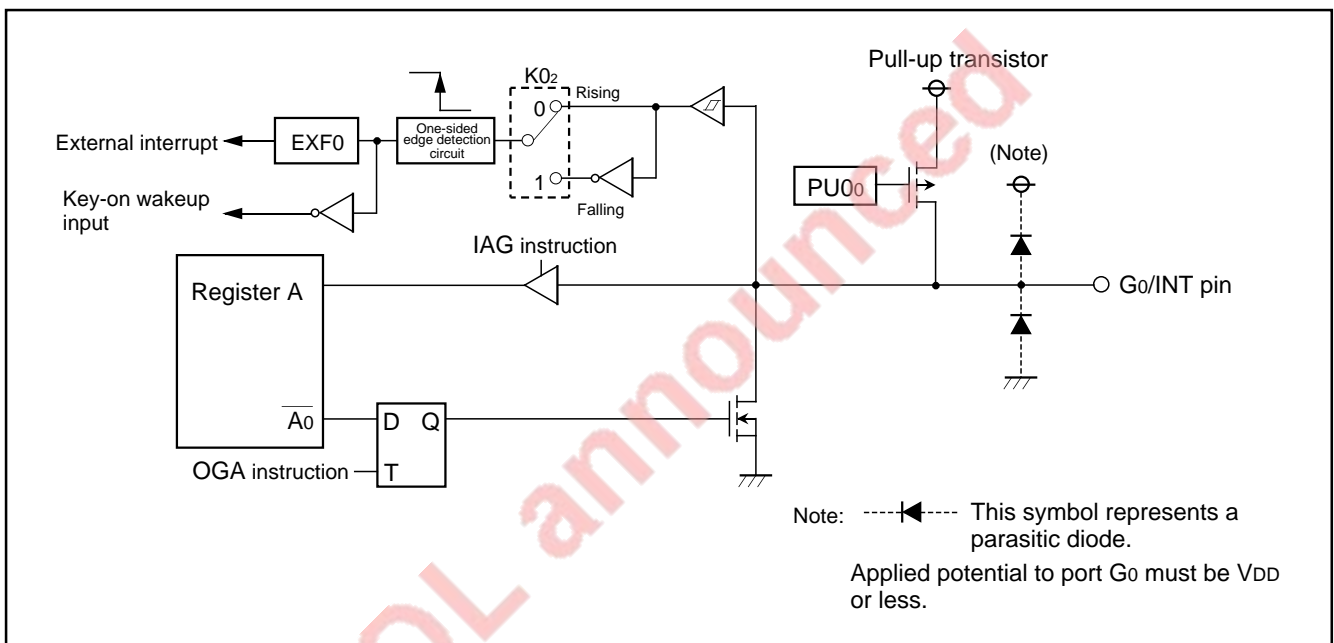


Fig. 17 External interrupt circuit structure

(1) External interrupt request flag (EXF0)

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

The valid waveforms causing the interrupt must be retained at their level for 5 cycles or more of $f(X_{IN})$ (Refer to Figure 16).

The state of EXF0 flag can be examined with the skip instruction (SNZ0). Use the timer 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 interrupt activated condition
External interrupt activated condition is satisfied when a valid waveform is input to Go/INT pin.
The valid waveform can be selected from rising waveform or falling waveform. An example of how to use the external interrupt is as follows.

- ① Select the valid waveform with the bit 2 of register K0.
- ② Clear the EXF0 flag to "0" with the SNZ0 instruction.
- ③ Set the NOP instruction for the case when a skip is performed with the SNZ0 instruction.
- ④ Set both the external interrupt enable bit (V1₀) and the INTE flag to "1."

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

(2) Control register related to external interrupt

- Key-on wakeup control register K0
Register K0 controls the valid waveform for the external interrupt and key-on wakeup function. Set the contents of this register through register A with the TK0A instruction. The TAK0 instruction can be used to transfer the contents of register K0 to register A.

Table 8 Control register related to external interrupt

| Key-on wakeup control register K0 | | at reset : 0000 ₂ | | at RAM back-up : state retained | R/W |
|-----------------------------------|----------------------------------------------------------------------------------------------|------------------------------|-------------------------------------------|---------------------------------|-----|
| K0 ₃ | Prescaler dividing ratio selection bit | 0 | Instruction clock divided by 4 | | |
| | | 1 | Instruction clock divided by 512 | | |
| K0 ₂ | Interrupt valid waveform for INT pin/ key-on wakeup valid waveform selection bit (Note 2) | 0 | Rising waveform ("L" → "H") | | |
| | | 1 | Falling waveform ("H" → "L") | | |
| K0 ₁ | Ports G ₁ –G ₃ key-on wakeup control bit | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used ("L" level recognized) | | |
| K0 ₀ | Ports S ₀ –S ₃ key-on wakeup control bit | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used ("L" level recognized) | | |

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

- 2: Set a value to the bit 2 of register K0, and execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction. According to the input state of Go/INT pin, the external interrupt request flag (EXF0) may be set to "1" when the interrupt valid waveform is changed.

TIMERS

The 4250 Group has the programmable timer.

- 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).

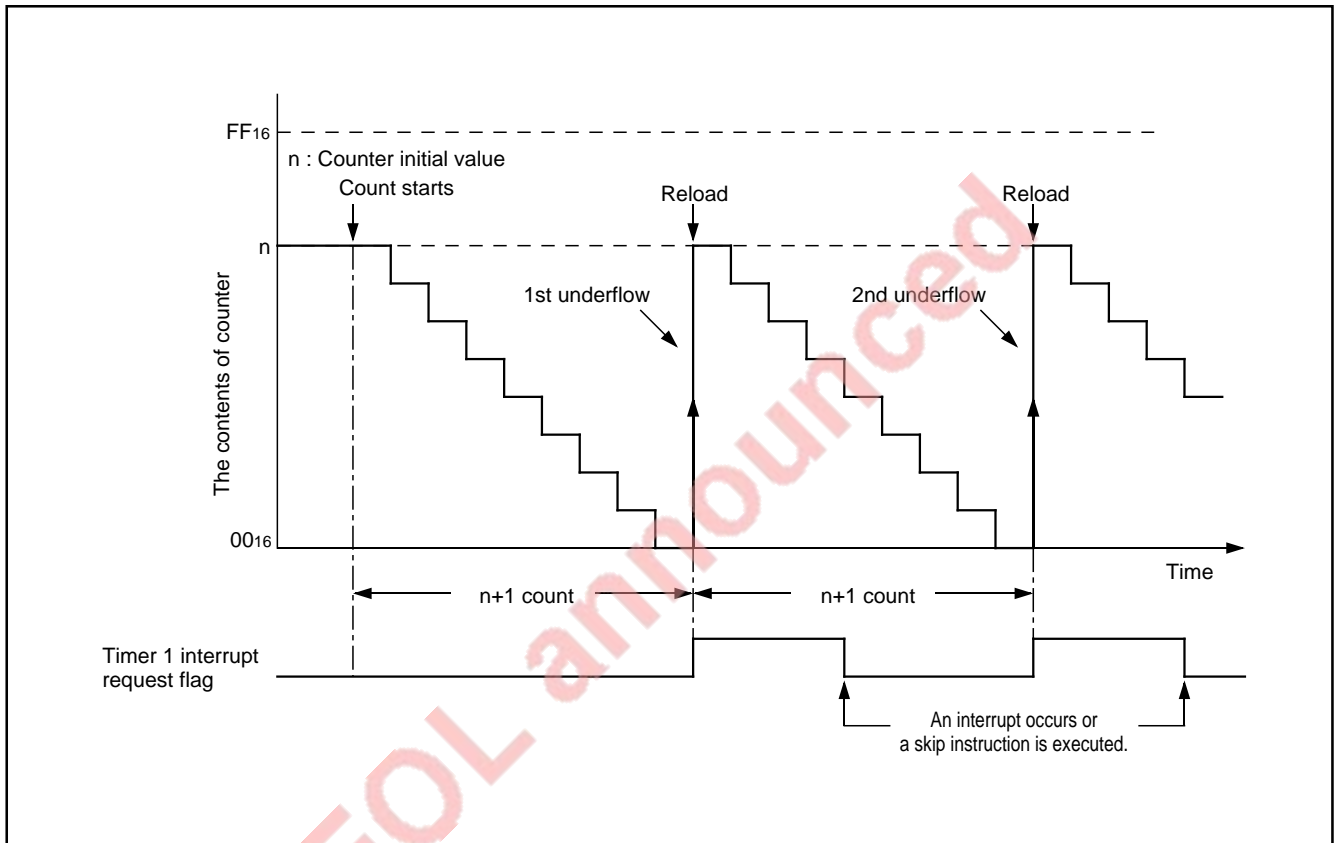


Fig. 18 Auto-reload function

The 4250 Group timer consists of the following circuits.

- Prescaler : frequency divider
- Timer 1 : 8-bit programmable timer with the interrupt function

These timers can be controlled with the timer control register V1 and key-on wakeup control register K0. Each function is described below.

Table 9 Function related timers

| Circuit | Structure | Count source | Frequency dividing ratio | Use of output signal | Control register |
|-----------|----------------------------------------|--------------------------|--------------------------|-----------------------------------|------------------|
| Prescaler | Frequency divider | Instruction clock | 4, 512 | • Timer 1 count source | V1 K0 |
| Timer 1 | 8-bit programmable binary down counter | Prescaler output (ORCLK) | 1 to 256 | • Tout pin • Timer 1 interrupt | V1 |

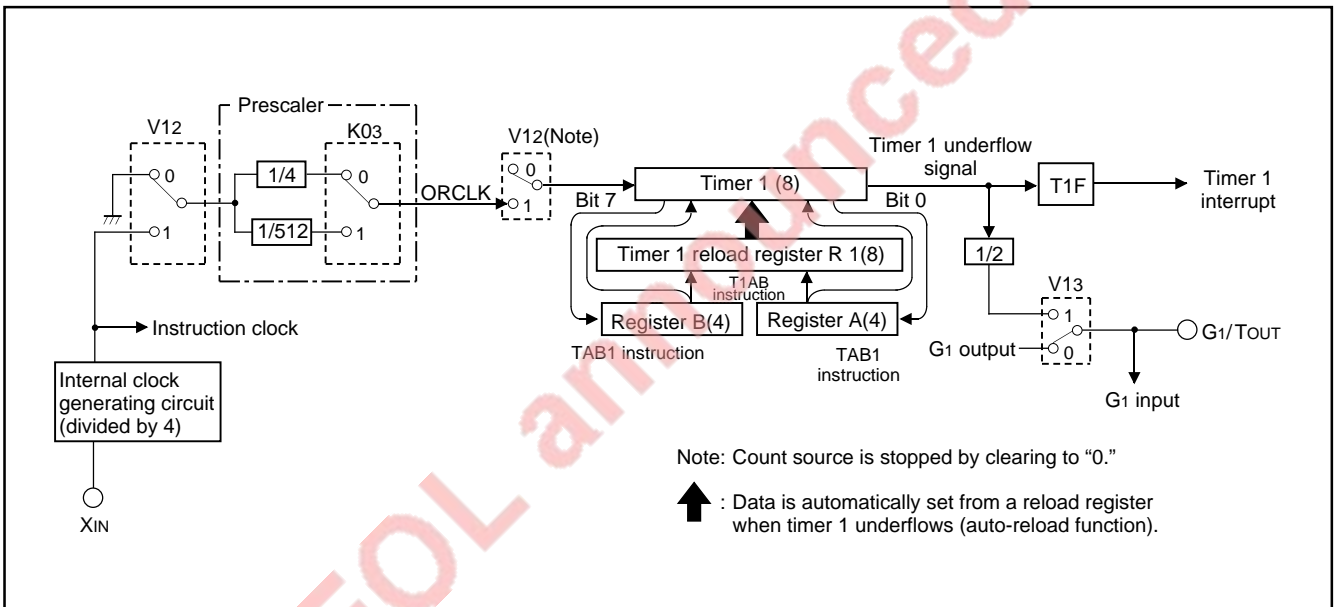


Fig. 19 Timers structure

Table 10 Control registers related to timer

| Timer control register V1 | | at reset : 0000 ₂ | at RAM back-up : 0000 ₂ | R/W |
|-----------------------------------|-------------------------------------------------------------------------------------------------|------------------------------|----------------------------------------------------------------|-----|
| V1 ₃ | G ₁ /T _{OUT} pin function selection bit | 0 | Port G ₁ (I/O) | |
| | | 1 | T _{OUT} pin (output)/port G ₁ (input) | |
| V1 ₂ | Prescaler/timer 1 operation start bit | 0 | Prescaler stop (initial state) / timer 1 stop (state retained) | |
| | | 1 | Prescaler / timer 1 operation | |
| V1 ₁ | Timer 1 interrupt enable bit | 0 | Interrupt disabled (SNZ1 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZ1 instruction is invalid) | |
| V1 ₀ | External interrupt enable bit | 0 | Interrupt disabled (SNZ0 instruction is valid) | |
| | | 1 | Interrupt enabled (SNZ0 instruction is invalid) | |
| Key-on wakeup control register K0 | | at reset : 0000 ₂ | at RAM back-up : state retained | R/W |
| K0 ₃ | Prescaler dividing ratio selection bit | 0 | Instruction clock divided by 4 | |
| | | 1 | Instruction clock divided by 512 | |
| K0 ₂ | Interrupt valid waveform for INT pin/ key-on wakeup valid waveform selection bit (Note 2) | 0 | Rising waveform ("L" → "H") | |
| | | 1 | Falling waveform ("H" → "L") | |
| K0 ₁ | Ports G ₁ –G ₃ key-on wakeup control bit | 0 | Key-on wakeup not used | |
| | | 1 | Key-on wakeup used ("L" level recognized) | |
| K0 ₀ | Ports S ₀ –S ₃ key-on wakeup control bit | 0 | Key-on wakeup not used | |
| | | 1 | Key-on wakeup used ("L" level recognized) | |

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

2: Set a value to the bit 2 of register K0, and execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction. According to the input state of G₀/INT pin, the external interrupt request flag (EXF0) may be set to "1" when the interrupt valid waveform is changed.

(1) Control registers related to timer

- Timer control register V1
G₁/T_{OUT} pin function selection bit and prescaler/timer 1 operation start bit 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.
- Key-on wakeup control register K0
Prescaler dividing ratio selection bit is assigned to register K0. Set the contents of this register through register A with the TK0A instruction. The TAK0 instruction can be used to transfer the contents of register K0 to register A.

(2) Precautions

Note the following for the use of timers.

- Prescaler
Stop the prescaler operation to change its frequency dividing ratio.
- Reading the count value
Stop timer 1 counting and then execute the TAB1 instruction to read its data.

(3) Prescaler

Prescaler is a frequency divider. Its frequency dividing ratio can be selected. The count source of prescaler is the instruction clock.

Use the bit 3 of register K0 to select the prescaler dividing ratio and the bit 2 of register V1 to start and stop its operation. Prescaler is initialized, and the output signal (ORCLK) stops when the bit 2 of register V1 is cleared to "0."

(4) 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.

Timer 1 starts counting after the following process;

- ① set data in timer 1, and
- ② set the bit 2 of register V1 to "1."

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).

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

Data can be read from timer 1 to registers A and B with the TAB1 instruction. When reading the data, stop the counter and then execute the TAB1 instruction. Timer 1 underflow signal divided by 2 can be output from G₁/T_{OUT} pin.

(5) Timer output pin (G₁/T_{OUT})

Timer output pin (G₁/T_{OUT}) has the function to output the timer 1 underflow signal divided by 2. The selection of G₁/T_{OUT} pin function can be controlled with the bit 3 of register V1.

(6) Timer interrupt request flag (T1F)

Timer interrupt request flag is set to "1" when the timer underflows. The state of this flag can be examined with the skip instruction (SNZ1).

Use the register V1 to select an interrupt or a skip instruction. T1F flag is cleared to "0" when an interrupt occurs or when the next instruction is skipped with a skip instruction.

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.

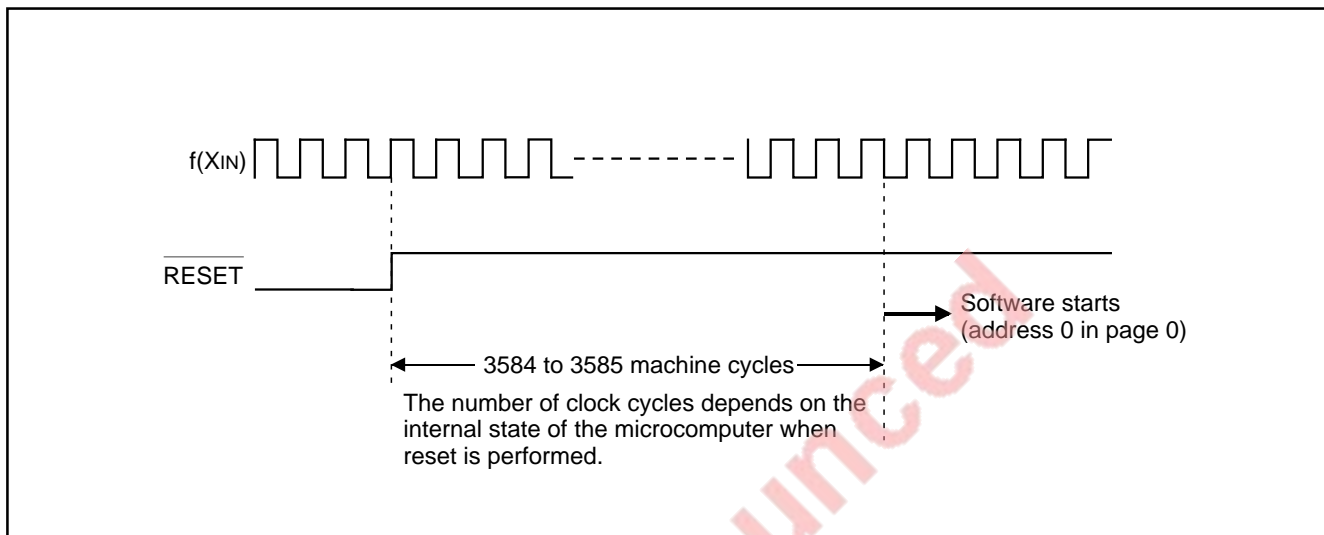


Fig. 20 Reset release timing

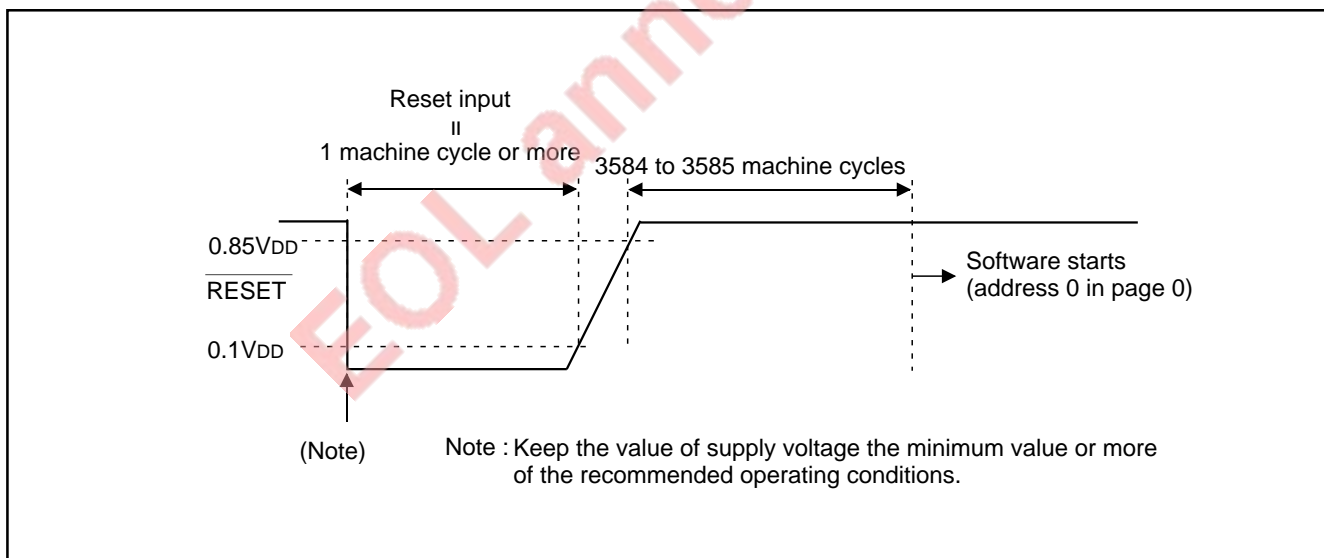


Fig. 21 $\overline{\text{RESET}}$ pin input waveform and reset operation

(1) Power-on reset

Reset can be automatically performed at power on (power-on reset) by connecting a resistor, a diode, and a capacitor to RESET pin. Connect RESET pin and the external circuit at the shortest distance.

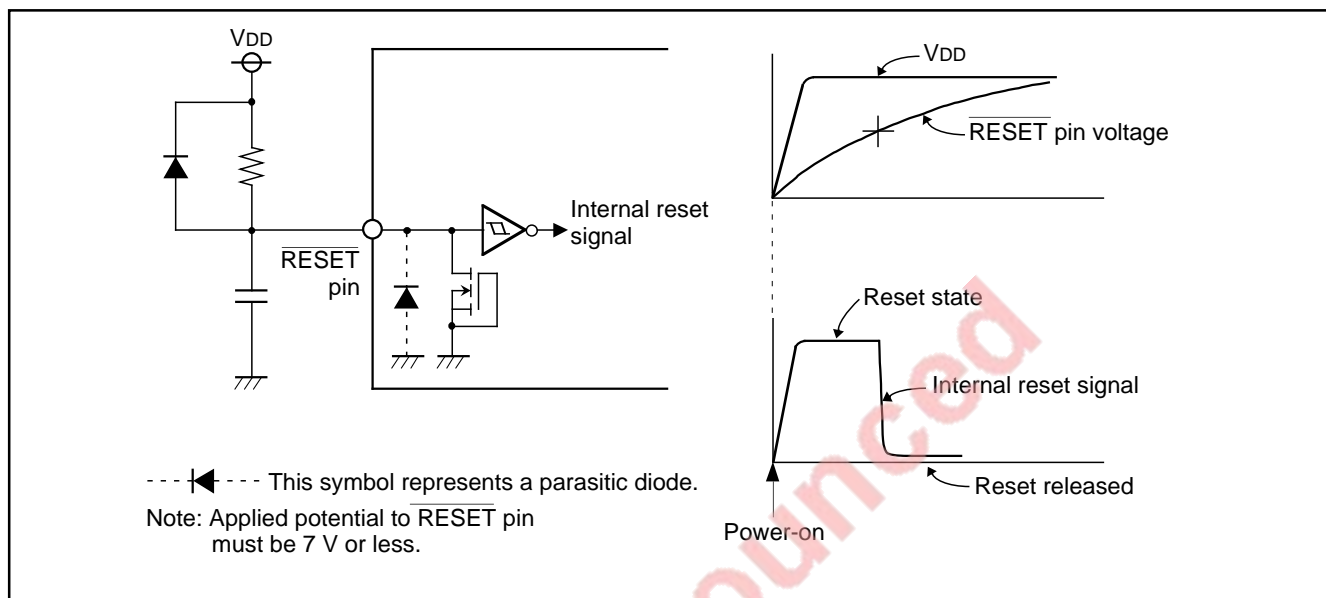


Fig. 22 Power-on reset circuit example

(2) Internal state at reset

Table 11 shows port state at reset, and Figure 23 shows internal state at reset (they are retained after system is released from reset).

The contents of timers, registers, flags and RAM except shown in Figure 23 are undefined, so set the initial value to them.

Table 11 Port state at reset

| Name | Function | State |
|------------------------------------------------------------------------|------------------------------------------------------------------------|--------------------------|
| D ₀ , D ₁ , D ₂ /C, D ₃ /K | D ₀ , D ₁ , D ₂ /C, D ₃ /K | High impedance (Note) |
| S ₀ –S ₃ | S ₀ –S ₃ | |
| G ₀ /INT, G ₁ /TOUT | G ₀ /INT, G ₁ | |
| G ₂ , G ₃ | G ₂ , G ₃ | |
| F ₀ , F ₁ | F ₀ , F ₁ | |

Note: Output latch is set to "1."

| | |
|-------------------------------------------------|-------------------------|
| • Program counter (PC) | 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 interrupt request flag (EXF0) | 0 |
| • Timer 1 interrupt request flag (T1F) | 0 |
| • Timer control register V1 | 0 0 0 0 |
| (Interrupt disabled, prescaler/timer 1 stopped) | |
| • Key-on wakeup control register K0 | 0 0 0 0 |
| • Pull-up control register PU0 | 0 0 |
| • Logic operation selection register LO | 0 0 |
| • Carry flag (CY) | 0 |
| • Register A | 1 1 1 1 |
| • Register B | 1 1 1 1 |
| • Stack pointer (SP) | 1 1 |

Fig. 23 Internal state at reset

EOL announced

RAM BACK-UP MODE

The 4250 Group has the RAM back-up mode. When the POF instruction is executed continuously, system enters the RAM back-up state.

As oscillation stops retaining RAM, the function of reset circuit and states at RAM back-up mode, current dissipation can be reduced without losing the contents of RAM. Table 12 shows the function and states retained at RAM back-up. Figure 24 shows the state transition.

(1) Identification of the start condition

Warm start (return from the RAM back-up state) or cold start (return from the normal reset state) can be identified by examining the state of the power down flag (P) with the SNZP instruction.

(2) Warm start condition

When the external wakeup signal is input after the system enters the RAM back-up state by executing the POF instruction continuously, the CPU starts executing the software from address 0 in page 0. In this case, the P flag is "1."

(3) Cold start condition

The CPU starts executing the software from address 0 in page 0 when reset pulse is input to RESET pin. In this case, the P flag is "0."

(4) Return signal

An external wakeup signal is used to return from the RAM back-up mode. Table 13 shows the return condition for each return source.

Table 12 Functions and states retained at RAM back-up

| Function | RAM back-up |
|------------------------------------------------------------------------------------|-------------|
| Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) (Note 2) | X |
| Contents of RAM | O |
| Port | X |
| Timer control register V1 | X |
| Timer 1 function | X |
| Pull-up control register PU0 | O |
| Key-on wakeup control register K0 | O |
| Logic operation selection register LO | X |
| External interrupt request flag (EXF0) | X |
| Timer 1 interrupt request flag (T1F) | X |
| Interrupt enable flag (INTE) | X |

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 "3" at RAM back-up.

Table 13 Return source and return condition

| Return source | Return condition | Remarks |
|------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| G ₀ /INT pin | Return by an external rising edge input ("L"→"H") or falling edge input ("H"→"L"). The EXF0 flag is not set. | Select the return edge (rising edge or falling edge) with the bit 2 of register K0 according to the external state before going into the RAM back-up state. |
| Ports G ₁ –G ₃ S ₀ –S ₃ | Return by an external "L" level input. | Set the port using the key-on wakeup function selected with register K0 to "H" level before going into the RAM back-up state. |

Note: G₀/INT pin and ports G₁–G₃, S₀–S₃ share the circuit which is used to detect the edge and to recognize "L" level.
The G₀/INT pin cannot be set to "no key-on wakeup."

(5) Key-on wakeup control register K0

- Key-on wakeup control register K0
The interrupt valid waveform for INT pin/key-on wakeup valid waveform selection bit, the ports G₁–G₃ key-on wakeup control bit and the ports S₀–S₃ key-on wakeup control bit are assigned to the register K0. Set the contents

of this register through register A with the TK0A instruction. The TAK0 instruction can be used to transfer the contents of register K0 to register A.

Table 14 Key-on wakeup control register

| Key-on wakeup control register K0 | | at reset : 0000 ₂ | at RAM back-up : state retained | R/W |
|-----------------------------------|----------------------------------------------------------------------------------------------|------------------------------|-------------------------------------------|-----|
| K0 ₃ | Prescaler dividing ratio selection bit | 0 | Instruction clock divided by 4 | |
| | | 1 | Instruction clock divided by 512 | |
| K0 ₂ | Interrupt valid waveform for INT pin/ key-on wakeup valid waveform selection bit (Note 2) | 0 | Rising waveform ("L" → "H") | |
| | | 1 | Falling waveform ("H" → "L") | |
| K0 ₁ | Ports G ₁ –G ₃ key-on wakeup control bit | 0 | Key-on wakeup not used | |
| | | 1 | Key-on wakeup used ("L" level recognized) | |
| K0 ₀ | Ports S ₀ –S ₃ key-on wakeup control bit | 0 | Key-on wakeup not used | |
| | | 1 | Key-on wakeup used ("L" level recognized) | |

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

- 2: Set a value to the bit 2 of register K0, and execute the SNZ0 instruction to clear the EXF0 flag after executing at least one instruction. According to the input state of G₀/INT pin, the external interrupt request flag (EXF0) may be set when the interrupt valid waveform is changed.

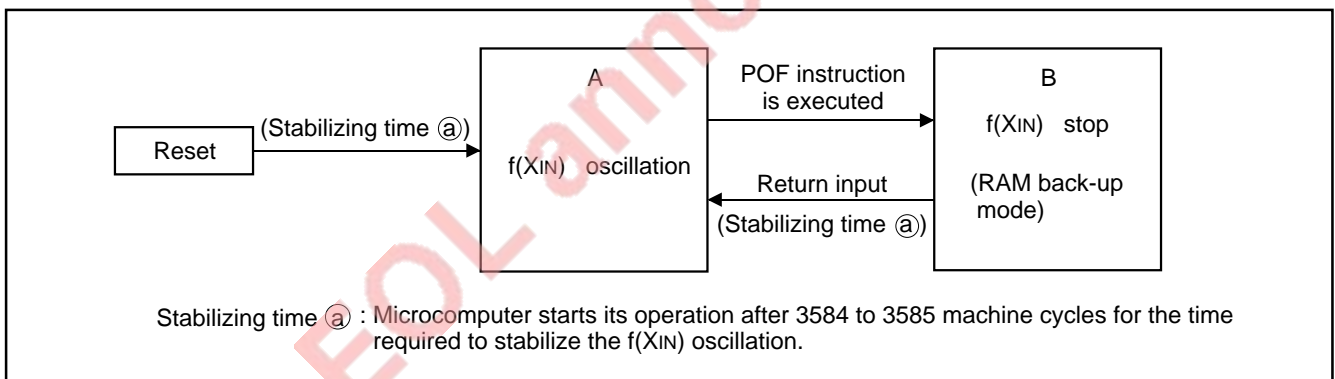


Fig. 24 State transition

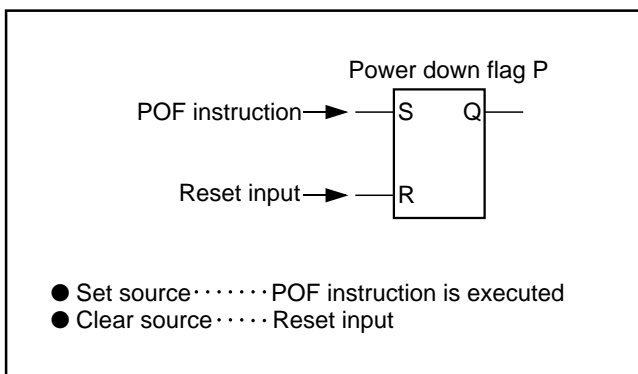


Fig. 25 Set source and clear source of the P flag

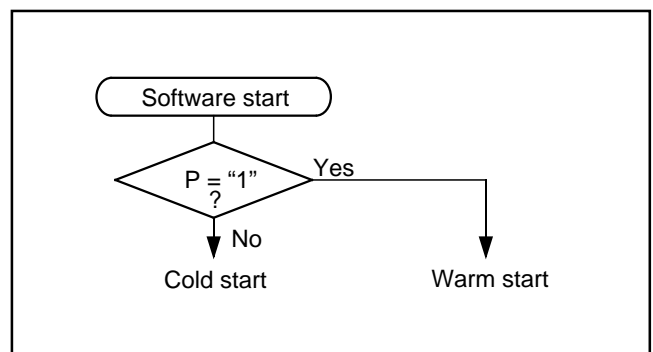


Fig. 26 Start condition identified example using the SNZP instruction

CLOCK CONTROL

The clock control circuit consists of the following circuits.

- System clock generating circuit
- Control circuit to stop the clock oscillation
- Control circuit to return from the RAM back-up state

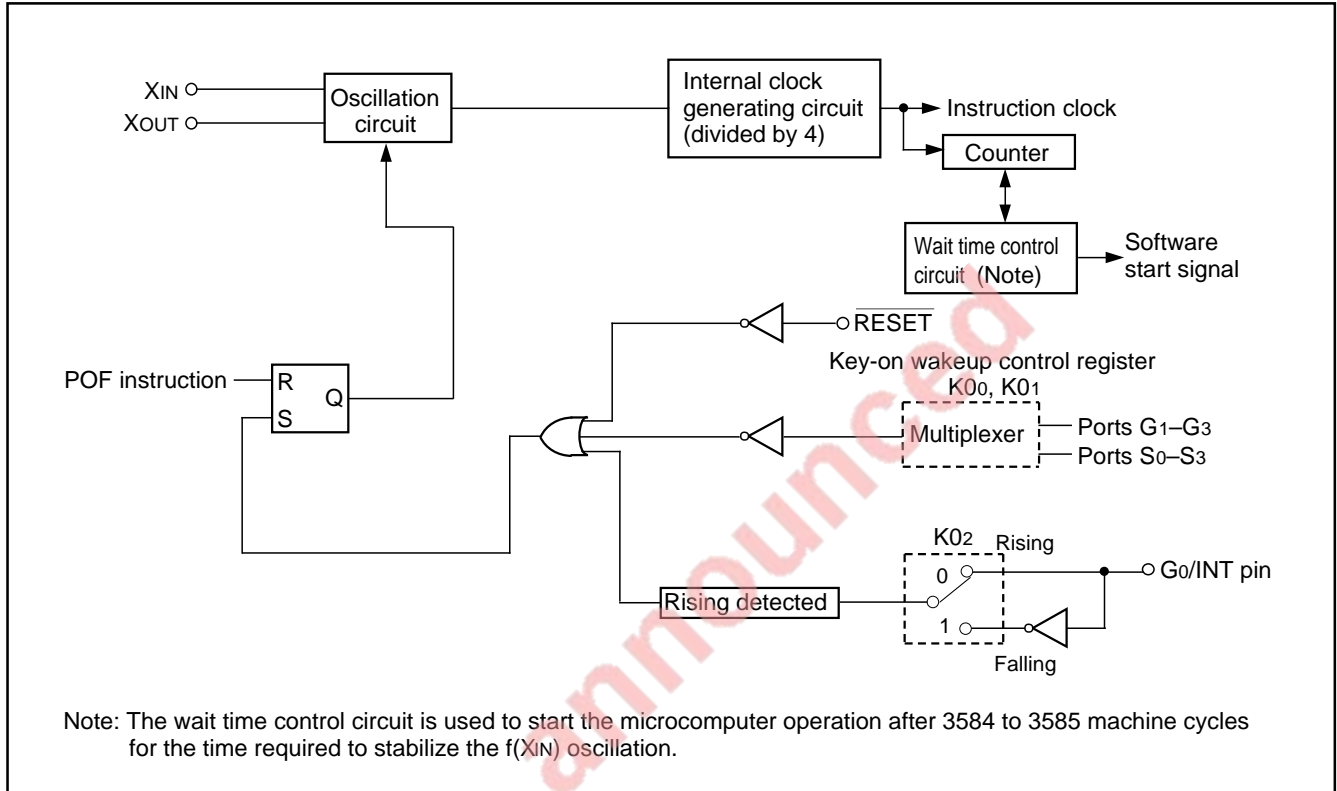


Fig. 27 Clock control circuit structure

Clock signal $f(X_{IN})$ is obtained by connecting X_{IN} pin and X_{OUT} pin directly, and externally connecting a resistor to X_{IN} and a capacitor to X_{OUT} . Connect this external circuit to pins X_{IN} and X_{OUT} at the shortest distance.

When an external clock signal is input, note the input waveform (refer to the list of precaution).

ROM ORDERING METHOD

Please submit the information described below when ordering Mask ROM.

- (1) M34250M2-XXXXFP Mask ROM Order Confirmation Form 1
- (2) Data to be written into mask ROM EPROM (three sets containing the identical data)
- (3) Mark Specification Form 1

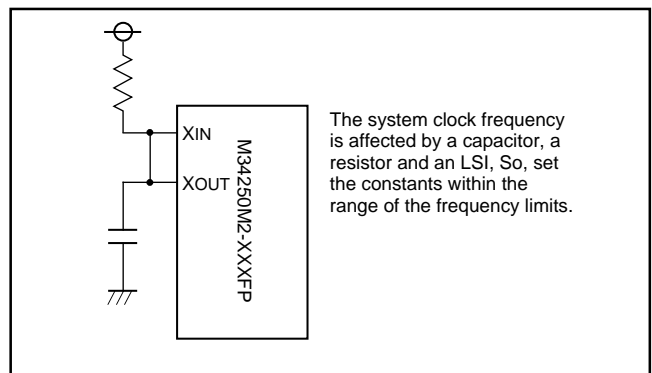


Fig. 28 Resistor and capacitor external circuit

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. 0.01 μ F) between pins V_{DD} and V_{SS} at the shortest distance,
- equalize its wiring in width and length, and
- use the thickest wire.

In the One Time PROM version, CNV_{SS} pin is also used as V_{PP} pin. Connect this pin to V_{SS} through the resistor about 5 k Ω which is assigned to CNV_{SS}/V_{PP} pin as close as possible at the shortest distance.

② **Prescaler**

Stop the prescaler operation to change its frequency dividing ratio.

③ **Timer count source**

Stop timer 1 counting to change its count source.

④ **Program counter**

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

⑤ **G_0/INT pin**

When the interrupt valid waveform of the G_0/INT pin is changed with the bit 2 of register K_0 in software, be careful about the following notes.

- After clear the bit 0 of register V_1 to "0" (Figure 29①), change the interrupt valid waveform of G_0/INT pin with the bit 2 of register K_0 .
- Set a value to bit 2 of register K_0 and execute the SNZ_0 instruction to clear the external interrupt request flag (EXF_0) after executing at least one instruction (refer to Figure 29②). Depending on the input state of the G_0/INT pin, the EXF_0 flag may be set when the interrupt valid waveform is changed.

⑥ **Notes on unused pins**

- When pins G_0/INT , G_1/T_{OUT} , G_2 and G_3 are connected to V_{SS} pin, turn off their pull-up transistors (register $PU_0="X0_2"$) and also invalidate the key-on wakeup functions of pins G_1/T_{OUT} , G_2 and G_3 (register $K_0="XX0X_2"$) by software. When the POF instruction is executed while these pins are connected to V_{SS} and the key-on wakeup functions are left valid, the system returns from RAM back-up state by recognizing the return condition immediately after going into the RAM back-up state. When these pins are open, turn on their pull-up transistors (register $PU_0="X1_2"$) by software.
- When ports S_0-S_3 are connected to V_{SS} pin, invalidate the key-on wakeup functions (register $K_0="XXX0_2"$) by software. When the POF instruction is executed while these pins are connected to V_{SS} and the key-on wakeup functions are left valid, the system returns from RAM back-up state by recognizing the return condition immediately after going into the RAM back-up state.
- When ports D_2/C and D_3/K are connected to V_{SS} pin, turn off their pull-up transistors (register $PU_0="0X_2"$) by software. When these pins are open, turn on their pull-up transistors (register $PU_0="1X_2"$) by software.

(Note when connecting to V_{SS} and V_{DD})

- Connect the unused pins to V_{SS} or V_{DD} at the shortest distance (within 20 mm) and use the thick wire against noise.

⑦ **Multifunction**

- G_0/INT pin can be also used as an I/O port G_0 even when it is used as INT pin.
- G_1/T_{OUT} pin can be also used as input port G_1 even when it is used as T_{OUT} pin.
- D_2/C pin can be also used as I/O port D_2 even when it is used as port C .
- D_3/K pin can be also used as I/O port D_3 even when it is used as port K .

```

:
LA 4      ; (XXX02)
TV1A     ; The SNZ0 instruction is valid ..... ①
LA 4
TK0A     ; Change of the interrupt valid waveform
NOP      ..... ②
SNZ0     ; The SNZ0 instruction is executed
NOP
:
X : this bit is not related to the setting of  $G_0/INT$  pin.
    
```

Fig. 29 External interrupt program example

⑩ Key-on wakeup

When system returns from RAM back-up state by using the G_0 /INT pin, select the return edge (rising edge or falling edge) with the bit 2 of register K0 according to the external state before going into the RAM back-up state.

When system returns from RAM back-up state by using the ports G_1 – G_3 and S_0 – S_3 , set the port using the key-on wakeup function selected with register K0 to “H” level before going into the RAM back-up state.

G_0 /INT pin and ports G_1 – G_3 , S_0 – S_3 share the circuit which is used to detect the edge and to recognize “L” level.

The G_0 /INT pin cannot be set to “no key-on wakeup.”

⑩ External clock input waveform

When the external clock is used, open X_{OUT} pin, and input the clock waveform into X_{IN} pin shown below. (Refer to Figure 30)

•Duty ratio = 50 %.

•“H” level input voltage= V_{DD} (V), “L” level input voltage= V_{SS} (V).

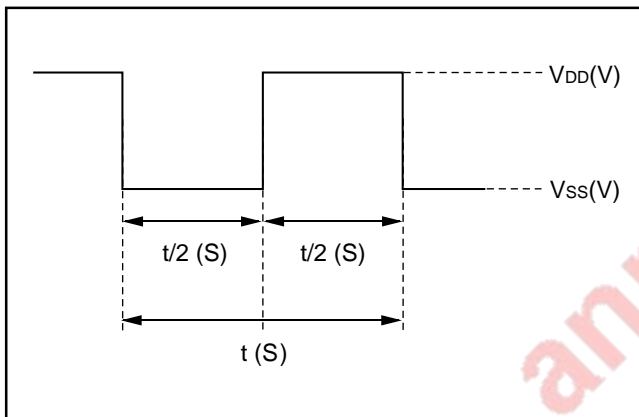


Fig. 30 External clock input waveform

⑩ CR oscillation constant

Use the external 30 pF capacitor and enable to change the frequency by the external resistor.

Test the system sufficiently because the oscillation constant depends on the ROM type (mask ROM or PROM).

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) | D | Port D (4 bits) |
| B | Register B (4 bits) | F | Port F (2 bits) |
| DR | Register D (3 bits) | G | Port G (4 bits) |
| E | Register E (8 bits) | S | Port S (4 bits) |
| V1 | Timer control register V1 (4 bits) | K | Port K (1 bit) |
| K0 | Key-on wakeup control register K0 (4 bits) | C | Port C (1 bit) |
| PU0 | Pull-up control register PU0 (2 bits) | x | Hexadecimal variable |
| LO | Logic operation selection register LO (2 bits) | y | Hexadecimal variable |
| X | Register X (2 bits) | p | Hexadecimal variable |
| Y | Register Y (4 bits) | n | Hexadecimal constant which represents the immediate value |
| DP | Data pointer (6 bits) (It consists of registers X and Y) | j | Hexadecimal constant which represents the immediate value |
| PC | Program counter (11 bits) | A ₃ A ₂ A ₁ A ₀ | Binary notation of hexadecimal variable A (same for others) |
| PC _H | High-order 4 bits of program counter | ← | Direction of data movement |
| PC _L | Low-order 7 bits of program counter | ↔ | Data exchange between a register and memory |
| SK | Stack register (11 bits X 4) | ? | Decision of state shown before “?” |
| SP | Stack pointer (2 bits) | () | Contents of registers and memories |
| CY | Carry flag | — | Negate, Flag unchanged after executing instruction |
| R1 | Timer 1 reload register | M(DP) | RAM address pointed by the data pointer |
| T1 | Timer 1 | a | Label indicating address a ₆ a ₅ a ₄ a ₃ a ₂ a ₁ a ₀ |
| T1F | Timer 1 interrupt request flag | p, a | Label indicating address a ₆ a ₅ a ₄ a ₃ a ₂ a ₁ a ₀ in page p ₃ p ₂ p ₁ p ₀ |
| INTE | Interrupt enable flag | C | Hex. C + Hex. number x (also same for others) |
| EXF0 | External interrupt request flag | + | |
| P | Power down flag | x | |

Note : The 4250 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.

LIST OF INSTRUCTION FUNCTION

| Grouping | Mnemonic | Function | Grouping | Mnemonic | Function | Grouping | Mnemonic | Function | |
|-------------------------------|----------|----------------------------------------------------------------------------------------------------------------------------|----------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| Register to register transfer | TAB | $(A) \leftarrow (B)$ | Arithmetic operation | LA n | $(A) \leftarrow n$ $n = 0 \text{ to } 15$ | Comparison operation | SEAM | $(A) = (M(DP)) ?$ | |
| | TBA | $(B) \leftarrow (A)$ | | TABP p | $(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow p$ $(PC_L) \leftarrow (DR_2-DR_0, A_3-A_0)$ | | SEA n | $(A) = n ?$ $n = 0 \text{ to } 15$ | |
| | TAY | $(A) \leftarrow (Y)$ | | AM | $(A) \leftarrow (A) + (M(DP))$ | Branch operation | B a | $(PC_L) \leftarrow a_6-a_0$ | |
| | TYA | $(Y) \leftarrow (A)$ | | AMC | $(A) \leftarrow (A) + (M(DP)) + (CY)$ $(CY) \leftarrow \text{Carry}$ | | BL p, a | $(PC_H) \leftarrow p$ $(PC_L) \leftarrow a_6-a_0$ | |
| | TEAB | $(E_7-E_4) \leftarrow (B)$ $(E_3-E_0) \leftarrow (A)$ | | A n | $(A) \leftarrow (A) + n$ $n = 0 \text{ to } 15$ | | BA a | $(PC_L) \leftarrow (a_6-a_4, A_3-A_0)$ | |
| | TABE | $(B) \leftarrow (E_7-E_4)$ $(A) \leftarrow (E_3-E_0)$ | | SC | $(CY) \leftarrow 1$ | | BLA p, a | $(PC_H) \leftarrow p$ $(PC_L) \leftarrow (a_6-a_4, A_3-A_0)$ | |
| | TDA | $(DR_2-DR_0) \leftarrow (A_2-A_0)$ | | RC | $(CY) \leftarrow 0$ | Subroutine operation | BM a | $(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow 2$ $(PC_L) \leftarrow a_6-a_0$ | |
| RAM addresses | LXY x, y | $(X) \leftarrow x, x = 0 \text{ to } 3$ $(Y) \leftarrow y, y = 0 \text{ to } 15$ | | SZC | $(CY) = 0 ?$ | | BML p, a | $(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow p$ $(PC_L) \leftarrow a_6-a_0$ | |
| | INX | $(Y) \leftarrow (Y) + 1$ | | CMA | $(A) \leftarrow (\bar{A})$ | | BMLA p, a | $(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow p$ $(PC_L) \leftarrow (a_6-a_4, A_3-A_0)$ | |
| | DEY | $(Y) \leftarrow (Y) - 1$ | | RAR | $\rightarrow \boxed{CY} \rightarrow \boxed{A_3A_2A_1A_0}$ | Return operation | RTI | $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$ | |
| RAM to register transfer | TAM j | $(A) \leftarrow (M(DP))$ $(X) \leftarrow (X) \text{ EXOR}(j)$ $j = 0 \text{ to } 3$ | | LGOP | Logic operation instruction XOR, OR, AND | | RT | $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$ | |
| | XAM j | $(A) \leftrightarrow (M(DP))$ $(X) \leftarrow (X) \text{ EXOR}(j)$ $j = 0 \text{ to } 3$ | | Bit operation | SB j | | $(M_j(DP)) \leftarrow 1$ $j = 0 \text{ to } 3$ | RTS | $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$ |
| | XAMD j | $(A) \leftrightarrow (M(DP))$ $(X) \leftarrow (X) \text{ EXOR}(j)$ $j = 0 \text{ to } 3$ $(Y) \leftarrow (Y) - 1$ | | | RB j | | $(M_j(DP)) \leftarrow 0$ $j = 0 \text{ to } 3$ | | |
| | XAMI j | $(A) \leftrightarrow (M(DP))$ $(X) \leftarrow (X) \text{ EXOR}(j)$ $j = 0 \text{ to } 3$ $(Y) \leftarrow (Y) + 1$ | | | SZB j | $(M_j(DP)) = 0 ?$ $j = 0 \text{ to } 3$ | | | |

LIST OF INSTRUCTION FUNCTION (CONTINUED)

| Grouping | Mnemonic | Function | Grouping | Mnemonic | Function | Grouping | Mnemonic | Function |
|---------------------|-----------------------------------|--------------------------------------------------------------------------|------------------------|----------|----------------------------------|-----------------|----------|-----------------|
| Interrupt operation | DI | INTE ← 0 | Input/Output operation | CLD | (D) ← 1 | Other operation | NOP | (PC) ← (PC) + 1 |
| | EI | INTE ← 1 | | RD | (D(Y)) ← 0 (Y) = 0 to 3 | | POF | RAM back-up |
| | SNZ0 | (EXF0) = 1 ? After skipping the next instruction (EXF0) ← 0 | | SD | (D(Y)) ← 1 (Y) = 0 to 3 | | SNZP | (P) = 1 ? |
| Timer operation | TAB1 | (B) ← (T17-T14) (A) ← (T13-T10) | | SZD | (D(Y)) = 0 ? (Y) = 0 to 3 | | TLOA | (LO) ← (A1, A0) |
| | T1AB | (R17-R14) ← (B) (T17-T14) ← (B) (R13-R10) ← (A) (T13-T10) ← (A) | | SCP | (C) ← 1 | | TV1A | (V1) ← (A) |
| | SNZ1 | (T1F) = 1 ? After skipping the next instruction (T1F) ← 0 | | OFA | (F) ← (A1, A0) | | TAV1 | (A) ← (V1) |
| | | | | IAF | (A1, A0) ← (F) (A3, A2) ← (0) | | TK0A | (K0) ← (A) |
| | | | | OGA | (G) ← (A) | | TAK0 | (A) ← (K0) |
| | | | | IAG | (A) ← (G) | | TPU0A | (PU0) ← (A) |
| | | | | OSA | (S) ← (A) | | | |
| | | | | IAS | (A) ← (S) | | | |
| | | | | OKA | (K) ← (A0) | | | |
| IAK | (A0) ← (K), (A3, A2, A1) ← (0) | | | | | | | |

INSTRUCTION CODE TABLE

| D3-D0 | D8-D4 Hex. notation | 00000 | 00001 | 00010 | 00011 | 00100 | 00101 | 00110 | 00111 | 01000 | 01001 | 01010 | 01011 | 01100 | 01101 | 01110 | 01111 | 10000 10111 | 11000 11111 |
|-------|---------------------------|-------|-------|----------|-------|---------|---------|-----------|-------|-------|------------|---------|----------|-------------|-------------|-------------|-------------|----------------|----------------|
| | | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 to 17 | 18 to 1F |
| 0000 | 0 | NOP | BLA | SZB 0 | BL | — | BMLA | XAM 0 | BML | OGA | TABP 0 | A 0 | LA 0 | LXY 0,0 | LXY 1,0 | LXY 2,0 | LXY 3,0 | BM | B |
| 0001 | 1 | BA | CLD | SZB 1 | BL | LGOP | — | XAM 1 | BML | OKA | TABP 1 | A 1 | LA 1 | LXY 0,1 | LXY 1,1 | LXY 2,1 | LXY 3,1 | BM | B |
| 0010 | 2 | — | — | SZB 2 | BL | — | — | XAM 2 | BML | SCP | TABP 2 | A 2 | LA 2 | LXY 0,2 | LXY 1,2 | LXY 2,2 | LXY 3,2 | BM | B |
| 0011 | 3 | SNZP | INY | SZB 3 | BL | — | — | XAM 3 | BML | RCP | TABP 3 | A 3 | LA 3 | LXY 0,3 | LXY 1,3 | LXY 2,3 | LXY 3,3 | BM | B |
| 0100 | 4 | DI | RD | SZD | BL | RT | — | TAM 0 | BML | OFA | TABP 4 | A 4 | LA 4 | LXY 0,4 | LXY 1,4 | LXY 2,4 | LXY 3,4 | BM | B |
| 0101 | 5 | EI | SD | SEAn | BL | RTS | IAS | TAM 1 | BML | T1AB | TABP 5 | A 5 | LA 5 | LXY 0,5 | LXY 1,5 | LXY 2,5 | LXY 3,5 | BM | B |
| 0110 | 6 | RC | — | SEAM | BL | RTI | IAF | TAM 2 | BML | TV1A | TABP 6 | A 6 | LA 6 | LXY 0,6 | LXY 1,6 | LXY 2,6 | LXY 3,6 | BM | B |
| 0111 | 7 | SC | DEY | — | BL | — | IAK | TAM 3 | BML | TK0A | TABP 7 | A 7 | LA 7 | LXY 0,7 | LXY 1,7 | LXY 2,7 | LXY 3,7 | BM | B |
| 1000 | 8 | — | — | IAG | BL | — | TLOA | XAMI 0 | BML | TAV1 | TABP 8 | A 8 | LA 8 | LXY 0,8 | LXY 1,8 | LXY 2,8 | LXY 3,8 | BM | B |
| 1001 | 9 | — | — | TDA | BL | — | — | XAMI 1 | BML | TAK0 | TABP 9 | A 9 | LA 9 | LXY 0,9 | LXY 1,9 | LXY 2,9 | LXY 3,9 | BM | B |
| 1010 | A | AM | TEAB | TABE | BL | — | — | XAMI 2 | BML | TAB1 | TABP 10 | A 10 | LA 10 | LXY 0,10 | LXY 1,10 | LXY 2,10 | LXY 3,10 | BM | B |
| 1011 | B | AMC | OSA | — | BL | — | — | XAMI 3 | BML | TPU0A | TABP 11 | A 11 | LA 11 | LXY 0,11 | LXY 1,11 | LXY 2,11 | LXY 3,11 | BM | B |
| 1100 | C | TYA | CMA | — | BL | RB 0 | SB 0 | XAMD 0 | BML | SNZ1 | TABP 12 | A 12 | LA 12 | LXY 0,12 | LXY 1,12 | LXY 2,12 | LXY 3,12 | BM | B |
| 1101 | D | POF | RAR | — | BL | RB 1 | SB 1 | XAMD 1 | BML | SNZCP | TABP 13 | A 13 | LA 13 | LXY 0,13 | LXY 1,13 | LXY 2,13 | LXY 3,13 | BM | B |
| 1110 | E | TBA | TAB | — | BL | RB 2 | SB 2 | XAMD 2 | BML | — | TABP 14 | A 14 | LA 14 | LXY 0,14 | LXY 1,14 | LXY 2,14 | LXY 3,14 | BM | B |
| 1111 | F | — | TAY | SZC | BL | RB 3 | SB 3 | XAMD 3 | BML | SNZ0 | TABP 15 | A 15 | LA 15 | LXY 0,15 | LXY 1,15 | LXY 2,15 | LXY 3,15 | 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 D8-D4 show the high-order 5 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.

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

| The second word | |
|-----------------|-------------------|
| BL | 1 1 a a a a a a a |
| BML | 1 0 a a a a a a a |
| BA | 1 1 a a a a a a a |
| BLA | 1 1 a a a p p p p |
| BMLA | 1 0 a a a p p p p |
| SEA | 0 1 0 1 1 n n n n |
| SZD | 0 0 0 1 0 1 0 1 1 |

Do not use the code marked “—.”

MACHINE INSTRUCTIONS

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | Hexadecimal notation | Number of words | Number of cycles | Function | |
|-----------------------------------|----------|------------------|----|----|----|----|----|----|----|----------------------|-----------------|------------------|----------|---------------------------------------------|
| | | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | | | | | D0 |
| Register to register transfer | TAB | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 1 E | 1 | 1 | (A) ← (B) |
| | TBA | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 0 E | 1 | 1 | (B) ← (A) |
| | TAY | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 1 F | 1 | 1 | (A) ← (Y) |
| | TYA | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 0 C | 1 | 1 | (Y) ← (A) |
| | TEAB | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 1 A | 1 | 1 | (E7-E4) ← (B) (E3-E0) ← (A) |
| | TABE | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 2 A | 1 | 1 | (B) ← (E7-E4) (A) ← (E3-E0) |
| | TDA | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 2 9 | 1 | 1 | (DR2-DR0) ← (A2-A0) |
| RAM addresses | LXY x, y | 0 | 1 | 1 | x1 | x0 | y3 | y2 | y1 | y0 | 0 C y +x | 1 | 1 | (X) ← x, x = 0 to 3 (Y) ← y, y = 0 to 15 |
| | INY | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 1 3 | 1 | 1 | (Y) ← (Y) + 1 |
| | DEY | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 1 7 | 1 | 1 | (Y) ← (Y) - 1 |

EOL announced

| 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 registers A and B to register E. |
| - | - | Transfers the contents of register E to registers A and B. |
| - | - | Transfers the contents of register A to register D. |
| Continuous description | - | <p>Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y.</p> <p>When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped.</p> |
| (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. |
| (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. |

MACHINE INSTRUCTIONS (CONTINUED)

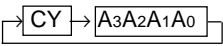
| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | Hexadecimal notation | Number of words | Number of cycles | Function | |
|-----------------------------------|----------|------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------------|-----------------|------------------|----------|-------------------------------------------------------------------|
| | | D ₈ | D ₇ | D ₆ | D ₅ | D ₄ | D ₃ | D ₂ | D ₁ | | | | | D ₀ |
| RAM to register transfer | TAM j | 0 | 0 | 1 | 1 | 0 | 0 | 1 | j ₁ | j ₀ | 0 6 4 +j | 1 | 1 | (A) ← (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 |
| | XAM j | 0 | 0 | 1 | 1 | 0 | 0 | 0 | j ₁ | j ₀ | 0 6 j | 1 | 1 | (A) ↔ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 |
| | XAMD j | 0 | 0 | 1 | 1 | 0 | 1 | 1 | j ₁ | j ₀ | 0 6 C +j | 1 | 1 | (A) ↔ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 (Y) ← (Y) - 1 |
| | XAMI j | 0 | 0 | 1 | 1 | 0 | 1 | 0 | j ₁ | j ₀ | 0 6 8 +j | 1 | 1 | (A) ↔ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 (Y) ← (Y) + 1 |

EOL announced

| Skip condition | Carry flag CY | Detailed 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. |
| - | - | 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. |
| (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. |

EOL announced

MACHINE INSTRUCTIONS (CONTINUED)

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | Hexadecimal notation | Number of words | Number of cycles | Function | |
|-----------------------------------|----------|------------------|----|----|----|----|----------------|----------------|----------------|----------------------|-----------------|------------------|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | | | | | D0 |
| Arithmetic operation | LA n | 0 | 1 | 0 | 1 | 1 | n ₃ | n ₂ | n ₁ | n ₀ | 0 B n | 1 | 1 | (A) ← n n = 0 to 15 |
| | TABP p | 0 | 1 | 0 | 0 | 1 | p ₃ | p ₂ | p ₁ | p ₀ | 0 9 p | 1 | 3 | (SK(SP)) ← (PC) (SP) ← (SP) + 1 (PC _H) ← p (PC _L) ← (DR ₂ –DR ₀ , A ₃ –A ₀) (B) ← (ROM(PC)) _{7 to 4} (A) ← (ROM(PC)) _{3 to 0} (SP) ← (SP) – 1 (PC) ← (SK(SP)) (Note) |
| | AM | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 0 A | 1 | 1 | (A) ← (A) + (M(DP)) |
| | AMC | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 0 B | 1 | 1 | (A) ← (A) + (M(DP)) + (CY) (CY) ← Carry |
| | A n | 0 | 1 | 0 | 1 | 0 | n ₃ | n ₂ | n ₁ | n ₀ | 0 A n | 1 | 1 | (A) ← (A) + n n = 0 to 15 |
| | SC | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 0 7 | 1 | 1 | (CY) ← 1 |
| | RC | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 0 6 | 1 | 1 | (CY) ← 0 |
| | SZC | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 2 F | 1 | 1 | (CY) = 0 ? |
| | CMA | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 1 C | 1 | 1 | (A) ← (A̅) |
| | RAR | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 1 D | 1 | 1 |  |
| LGOP | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 4 1 | 1 | 1 | Logic operation instruction XOR, OR, AND | |

Note : p is 0 to 15 for M34250E2, and p is 0 to 15 for M34250M2.

| 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. |
| - | - | Transfers 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 (DR ₂ DR ₁ DR ₀ A ₃ A ₂ A ₁ A ₀) specified by registers A and D in page p. When this instruction is executed, 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. The contents of carry flag CY remains unchanged. Skips the next instruction when there is no overflow as the result of operation. |
| - | 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. |
| - | - | Execute the logic operation selected by logic operation selection register LO between the contents of register A and port S, and stores the result in register A. |

MACHINE INSTRUCTIONS (CONTINUED)

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | Hexadecimal notation | Number of words | Number of cycles | Function | |
|-----------------------------------|----------|------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------------|-----------------|------------------|----------|---------------------------------------------------------------------------------------------------|
| | | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | | | | | D0 |
| Bit operation | SB j | 0 | 0 | 1 | 0 | 1 | 1 | 1 | j ₁ | j ₀ | 0 5 C +j | 1 | 1 | (Mj(DP)) ← 1 j = 0 to 3 |
| | RB j | 0 | 0 | 1 | 0 | 0 | 1 | 1 | j ₁ | j ₀ | 0 4 C +j | 1 | 1 | (Mj(DP)) ← 0 j = 0 to 3 |
| | SZB j | 0 | 0 | 0 | 1 | 0 | 0 | 0 | j ₁ | j ₀ | 0 2 j | 1 | 1 | (Mj(DP)) = 0 ? j = 0 to 3 |
| Comparison operation | SEAM | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 2 6 | 1 | 1 | (A) = (M(DP)) ? |
| | SEA n | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 2 5 | 2 | 2 | (A) = n ? n = 0 to 15 |
| | | 0 | 1 | 0 | 1 | 1 | n ₃ | n ₂ | n ₁ | n ₀ | 0 B n | | | |
| Branch operation | B a | 1 | 1 | a ₆ | a ₅ | a ₄ | a ₃ | a ₂ | a ₁ | a ₀ | 1 8 a +a | 1 | 1 | (PCL) ← a ₆ -a ₀ |
| | BL p, a | 0 | 0 | 0 | 1 | 1 | p ₃ | p ₂ | p ₁ | p ₀ | 0 3 p | 2 | 2 | (PCH) ← p (PCL) ← a ₆ -a ₀ (Note) |
| | | 1 | 1 | a ₆ | a ₅ | a ₄ | a ₃ | a ₂ | a ₁ | a ₀ | 1 8 a +a | | | |
| | BA a | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 0 1 | 2 | 2 | (PCL) ← (a ₆ -a ₄ , A ₃ -A ₀) |
| | | 1 | 1 | a ₆ | a ₅ | a ₄ | a ₃ | a ₂ | a ₁ | a ₀ | 1 8 a +a | | | |
| | BLA p, a | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 1 0 | 2 | 2 | (PCH) ← p (PCL) ← (a ₆ -a ₄ , A ₃ -A ₀) (Note) |
| | 1 | 1 | a ₆ | a ₅ | a ₄ | p ₃ | p ₂ | p ₁ | p ₀ | 1 8 p +a | | | | |

Note : p is 0 to 15 for M34250E2, and p is 0 to 15 for M34250M2.

| Skip condition | Carry flag CY | Detailed description |
|----------------------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| – | – | 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." |
| (A) = (M(DP)) | – | Skips the next instruction when the contents of register A is equal to the contents of M(DP). |
| (A) = n n = 0 to 15 | – | Skips the next instruction when the contents of register A is equal to the value n in the immediate field. |
| – | – | 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 within a page : Branches to address (a6 a5 a4 A3 A2 A1 A0) determined by replacing the low-order 4 bits of the address a with register A in the identical page. |
| – | – | Branch out of a page : Branches to address (a6 a5 a4 A3 A2 A1 A0) determined by replacing the low-order 4 bits of the address a with register A in page p. |

MACHINE INSTRUCTIONS (CONTINUED)

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | | | Hexadecimal notation | Number of words | Number of cycles | Function | |
|-----------------------------------|-----------|------------------|----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|---|----------------------|-----------------|------------------|----------|-----------------------------------------------------------------------------------------------------|
| | | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | | | | | | |
| Subroutine operation | BM a | 1 | 0 | a ₆ | a ₅ | a ₄ | a ₃ | a ₂ | a ₁ | a ₀ | 1 | a | a | 1 | 1 | (SK(SP)) ← (PC) (SP) ← (SP) + 1 (PCH) ← 2 (PCL) ← a ₆ -a ₀ |
| | BML p, a | 0 | 0 | 1 | 1 | 1 | p ₃ | p ₂ | p ₁ | p ₀ | 0 | 7 | p | 2 | 2 | (SK(SP)) ← (PC) (SP) ← (SP) + 1 (PCH) ← p (PCL) ← a ₆ -a ₀ (Note) |
| | BMLA p, a | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 2 | 2 |
| Return operation | RTI | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 4 | 6 | 1 | 1 | (PC) ← (SK(SP)) (SP) ← (SP) - 1 |
| | RT | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 4 | 4 | 1 | 2 | (PC) ← (SK(SP)) (SP) ← (SP) - 1 |
| | RTS | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 4 | 5 | 1 | 2 | (PC) ← (SK(SP)) (SP) ← (SP) - 1 |
| Interrupt operation | DI | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 4 | 1 | 1 | (INTE) ← 0 |
| | EI | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 5 | 1 | 1 | (INTE) ← 1 |
| | SNZ0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 8 | F | 1 | 1 | (EXF0) = 1 ? After skipping the next instruction (EXF0) ← 0 |

Note : p is 0 to 15 for M34250E2, and p is 0 to 15 for M34250M2.

| Skip condition | Carry flag CY | Detailed description |
|---------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| - | - | 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 (a ₆ a ₅ a ₄ A ₃ A ₂ A ₁ A ₀) determined by replacing the low-order 4 bits of address a with register A in page p. |
| - | - | Returns from interrupt service routine to main routine. Returns each value of data pointer (X, Y), carry flag, skip status, NOP mode status by the continuous description of the LA/LXY instruction to the states just before interrupt. |
| Skip at uncondition | - | Returns from subroutine to the routine called the subroutine. Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition. |
| - | - | Clears (0) to the interrupt enable flag INTE, and disables the interrupt. |
| - | - | Sets (1) to the interrupt enable flag INTE, and enables the interrupt. |
| (EXF0) = 1 | - | Skips the next instruction when the contents of EXF0 flag is "1." After skipping, clears the EXF0 flag. |

MACHINE INSTRUCTIONS (CONTINUED)

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | Hexadecimal notation | Number of words | Number of cycles | Function | |
|-----------------------------------|----------|------------------|----|----|----|----|----|----|----|----------------------|-----------------|------------------|----------|--------------------------------------------------------------------------|
| | | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | | | | | D0 |
| Timer operation | TAB1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 8 A | 1 | 1 | (B) ← (T17-T14) (A) ← (T13-T10) |
| | T1AB | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 8 5 | 1 | 1 | (R17-R14) ← (B) (T17-T14) ← (B) (R13-R10) ← (A) (T13-T10) ← (A) |
| | SNZ1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 8 C | 1 | 1 | (T1F) = 1 ? After skipping the next instruction (T1F) ← 0 |
| Input/Output operation | CLD | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 1 1 | 1 | 1 | (D) ← 1 |
| | RD | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 1 4 | 1 | 1 | (D(Y)) ← 0 (Y) = 0 to 3 |
| | SD | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 1 5 | 1 | 1 | (D(Y)) ← 1 (Y) = 0 to 3 |
| | SZD | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 2 4 | 2 | 2 | (D(Y)) = 0 ? (Y) = 0 to 3 |
| | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 2 B | | | | |

EOL announced

| Skip condition | Carry flag CY | Detailed description |
|----------------------------|---------------|-------------------------------------------------------------------------------------------------------------|
| - | - | Transfers the contents of timer 1 to registers A and B. |
| - | - | Transfers the contents of registers A and B to timer 1 and timer 1 reload register. |
| (T1F) = 1 | - | Skips the next instruction when the contents of T1F flag is "1." After skipping, clears (0) to T1F flag. |
| - | - | Sets (1) to port D (high-impedance state). |
| - | - | Clears (0) to a bit of port D specified by register Y. |
| - | - | Sets (1) to a bit of port D specified by register Y (high-impedance state). |
| (D(Y)) = 0 (Y) = 0 to 3 | - | Skips the next instruction when a bit of port D specified by register Y is "0." |

EOL announced

MACHINE INSTRUCTIONS (CONTINUED)

| Parameter Type of instructions | Mnemonic | Instruction code | | | | | | | | Hexadecimal notation | Number of words | Number of cycles | Function | |
|-----------------------------------|----------|------------------|----|----|----|----|----|----|----|----------------------|-----------------|------------------|-------------|------------------------------|
| | | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | | | | | D0 |
| Input/Output operation | OFA | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 8 4 | 1 | 1 | (F) ← (A1, A0) |
| | IAF | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 5 6 | 1 | 1 | (A1, A0) ← (F), (A3, A2) ← 0 |
| | OGA | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 8 0 | 1 | 1 | (G) ← (A) |
| | IAG | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 2 8 | 1 | 1 | (A) ← (G) |
| | OSA | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 1 B | 1 | 1 | (S) ← (A) |
| | IAS | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 5 5 | 1 | 1 | (A) ← (S) |
| | OKA | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 8 1 | 1 | 1 | (K) ← (A0) |
| | IAK | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 5 7 | 1 | 1 | (A0) ← (K), (A3-A1) ← 0 |
| | SCP | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 8 2 | 1 | 1 | (C) ← 1 |
| | RCP | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 8 3 | 1 | 1 | (C) ← 0 |
| | SNZCP | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 8 D | 1 | 1 | (C) = 1 ? |
| Other operation | NOP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 | 1 | 1 | (PC) ← (PC) + 1 |
| | POF | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 0 D | 1 | 1 | RAM back-up |
| | SNZP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 0 3 | 1 | 1 | (P) = 1 ? |
| | TLOA | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 5 8 | 1 | 1 | (LO) ← (A1, A0) |
| | TV1A | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 8 6 | 1 | 1 | (V1) ← (A) |
| | TAV1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 8 8 | 1 | 1 | (A) ← (V1) |
| | TK0A | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 8 7 | 1 | 1 | (K0) ← (A) |
| | TAK0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 8 9 | 1 | 1 | (A) ← (K0) |
| TPU0A | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 8 B | 1 | 1 | (PU0) ← (A) | |

| Skip condition | Carry flag CY | Detailed description |
|----------------|---------------|---------------------------------------------------------------------------------------------|
| - | - | Outputs the contents of register A to port F. |
| - | - | Transfers the contents of port F to register A. |
| - | - | Outputs the contents of register A to port G. |
| - | - | Transfers the contents of port G to register A. |
| - | - | Outputs the contents of register A to port S. |
| - | - | Transfers the contents of port S to register A. |
| - | - | Outputs the contents of register A to port K. |
| - | - | Transfers the contents of port K to register A. |
| - | - | Sets (1) to port C. |
| - | - | Clears (0) to port C. |
| (C) = 1 | - | Skips the next instruction when the contents of port C is "1." |
| - | - | No operation |
| - | - | Puts the system in RAM back-up state. |
| (P) = 1 | - | Skips the next instruction when P flag is "1." After skipping, P flag remains unchanged. |
| - | - | Transfers the contents of register A to the logic operation selection register LO. |
| - | - | Transfers the contents of register A to register V1. |
| - | - | Transfers the contents of register V1 to register A. |
| - | - | Transfers the contents of register A to register K0. |
| - | - | Transfers the contents of register K0 to register A. |
| - | - | Transfers the contents of register A to register PU0. |

CONTROL REGISTERS

| Timer control register V1 | | at reset : 0000 ₂ | | at RAM back-up : 0000 ₂ | R/W |
|---------------------------------------|----------------------------------------------------------------------------------------------|------------------------------|----------------------------------------------------------------|------------------------------------|-----|
| V1 ₃ | G ₁ /T _{OUT} pin function selection bit | 0 | Port G ₁ (I/O) | | |
| | | 1 | T _{OUT} pin (output) / port G ₁ (input) | | |
| V1 ₂ | Prescaler/timer 1 operation start bit | 0 | Prescaler stop (initial state) / timer 1 stop (state retained) | | |
| | | 1 | Prescaler/timer 1 operation | | |
| V1 ₁ | Timer 1 interrupt enable bit | 0 | Interrupt disabled (SNZ1 instruction is valid) | | |
| | | 1 | Interrupt enabled (SNZ1 instruction is invalid) | | |
| V1 ₀ | External interrupt enable bit | 0 | Interrupt disabled (SNZ0 instruction is valid) | | |
| | | 1 | Interrupt enabled (SNZ0 instruction is invalid) | | |
| Key-on wakeup control register K0 | | at reset : 0000 ₂ | | at RAM back-up : state retained | R/W |
| K0 ₃ | Prescaler dividing ratio selection bit | 0 | Instruction clock divided by 4 | | |
| | | 1 | Instruction clock divided by 512 | | |
| K0 ₂ | Interrupt valid waveform for INT pin/ key-on wakeup valid waveform selection bit (Note 2) | 0 | Rising waveform ("L" → "H") | | |
| | | 1 | Falling waveform ("H" → "L") | | |
| K0 ₁ | Ports G ₁ –G ₃ key-on wakeup control bit | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used ("L" level recognized) | | |
| K0 ₀ | Ports S ₀ –S ₃ key-on wakeup control bit | 0 | Key-on wakeup not used | | |
| | | 1 | Key-on wakeup used ("L" level recognized) | | |
| Pull-up control register PU0 | | at reset : 00 ₂ | | at RAM back-up : state retained | W |
| PU0 ₁ | Ports C and K pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| PU0 ₀ | Ports G ₀ –G ₃ pull-up transistor control bit | 0 | Pull-up transistor OFF | | |
| | | 1 | Pull-up transistor ON | | |
| Logic operation selection register LO | | at reset : 00 ₂ | | at RAM back-up : 00 ₂ | W |
| LO ₁ | Logic operation function selection bits | LO ₁ | LO ₀ | Functions | |
| | | 0 | 0 | XOR operation | |
| 0 | | 1 | OR operation | | |
| LO ₀ | | 1 | 0 | AND operation | |
| | 1 | 1 | Not available | | |

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

2: Set a value to the bit 2 of register K0, and execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction. According to the input state of G₀/INT pin, the external interrupt request flag (EXF0) may be set to "1" when the interrupt valid waveform is changed.

ABSOLUTE MAXIMUM RATINGS

| Symbol | Parameter | Conditions | Ratings | Unit |
|------------------|--------------------------------------------------------------------------------------------------------------------------|----------------------------------------|------------------------------|------|
| V _{DD} | Supply voltage | | -0.3 to 7.0 | V |
| V _I | Input voltage X _{IN} , G ₀ -G ₃ , D ₂ /C, D ₃ /K | | -0.3 to V _{DD} +0.3 | V |
| V _I | Input voltage F ₀ , F ₁ , S ₀ -S ₃ , D ₀ , D ₁ , RESET | | -0.3 to 8.0 | V |
| V _O | Output voltage X _{OUT} | | -0.3 to V _{DD} +0.3 | V |
| V _O | Output voltage F ₀ , F ₁ , S ₀ -S ₃ , D ₀ , D ₁ | Output transistors in cut-off state | -0.3 to 8.0 | V |
| V _O | Output voltage G ₀ -G ₃ , D ₂ /C, D ₃ /K | | -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 |

RECOMMENDED OPERATING CONDITIONS

(T_a = -20 °C to 85 °C, V_{DD} = 2.2 V to 5.5 V, unless otherwise noted)

| Symbol | Parameter | Conditions | Limits | | | Unit |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------------------|------|---------------------|------|
| | | | Min. | Typ. | Max. | |
| V _{DD} | Supply voltage | 0.4 MHz ≤ f(X _{IN}) ≤ 4.4 MHz | 4.5 | 5.0 | 5.5 | V |
| | | 0.4 MHz ≤ f(X _{IN}) ≤ 1.1 MHz | 2.2 | | 5.5 | V |
| V _{RAM} | RAM back-up voltage (at RAM back-up mode) | | 2.0 | | 5.5 | V |
| V _{SS} | Supply voltage | | | 0 | | V |
| V _{IH} | "H" level input voltage F ₀ , F ₁ , D ₀ , D ₁ | | 0.7V _{DD} | | 7 | V |
| V _{IH} | "H" level input voltage G ₀ -G ₃ , D ₂ , D ₃ | | 0.7V _{DD} | | V _{DD} | V |
| V _{IH} | "H" level input voltage INT | | 0.85V _{DD} | | V _{DD} | V |
| V _{IH} | "H" level input voltage C, K | V _{DD} = 4.5 V to 5.5 V | 0.5V _{DD} | | V _{DD} | V |
| | | V _{DD} = 2.2 V to 5.5 V | 0.7V _{DD} | | V _{DD} | V |
| V _{IH} | "H" level input voltage S ₀ -S ₃ | V _{DD} = 4.5 V to 5.5 V | 0.4V _{DD} | | 7 | V |
| | | V _{DD} = 2.2 V to 5.5 V | 0.6V _{DD} | | 7 | V |
| V _{IH} | "H" level input voltage RESET | | 0.85V _{DD} | | 7 | V |
| V _{IL} | "L" level input voltage C, K | | 0 | | 0.16V _{DD} | V |
| V _{IL} | "L" level input voltage S ₀ -S ₃ | | 0 | | 0.2V _{DD} | V |
| V _{IL} | "L" level input voltage F ₀ , F ₁ , G ₀ -G ₃ , D ₀ -D ₃ | | 0 | | 0.3V _{DD} | V |
| V _{IL} | "L" level input voltage INT | | 0 | | 0.15V _{DD} | V |
| V _{IL} | "L" level input voltage RESET | | 0 | | 0.1V _{DD} | V |
| I _{OL(peak)} | "L" level peak output current F ₀ , F ₁ , S ₀ -S ₃ , D ₀ , D ₁ , D ₂ /C, D ₃ /K | | | | 24 | mA |
| | | "L" level peak output current G ₀ , G ₁ /T _{OUT} , G ₂ , G ₃ | | | 10 | mA |
| I _{OL(avg)} | "L" level average output current F ₀ , F ₁ , S ₀ -S ₃ , D ₀ , D ₁ , D ₂ /C, D ₃ /K | (Note 1) | | | 12 | mA |
| | | (Note 1) | | | 5 | mA |
| f(X _{IN}) | System clock frequency (Note 2) | V _{DD} = 4.5 V to 5.5 V | 0.4 | 4.0 | 4.4 | MHz |
| | | V _{DD} = 2.2 V to 5.5 V | 0.4 | 1.0 | 1.1 | |
| Δf(X _{IN}) | Frequency error (errors of external capacitor and resistor not included) Note: Use the 30 pF capacitor externally and enable the change of frequency by external resistor. | V _{DD} = 5 V ±10 % T _a = 25 °C [reference] (-20 °C to 85 °C) | | | ±17 | % |
| | | V _{DD} = 3 V ±10 % T _a = 25 °C [reference] (-20 °C to 85 °C) | | | ±17 | |

Notes 1: Keep the total currents of I_{OL(avg)} for ports S₀-S₃, D₀, D₁, D₂/C, D₃/K to 50 mA or less.

Keep the total currents of I_{OL(avg)} for ports F₀, F₁, G₀, G₂, G₃ and G₁/T_{OUT} pin to 30 mA or less.

2: The system clock frequency is affected by the external capacitor, resistor and LSI. Accordingly, set the constants so as not to exceed the frequency limits.

Be careful about the input waveform when using the external clock. Refer to the notes on use.

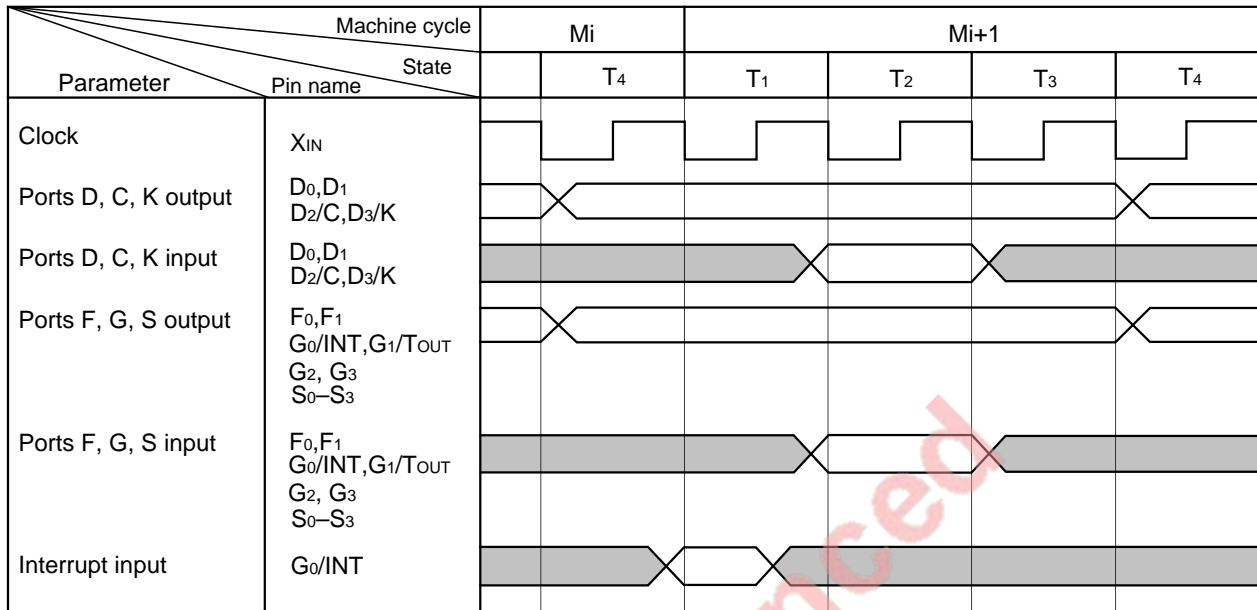
ELECTRICAL CHARACTERISTICS

(Ta = -20 °C to 85 °C, VDD = 2.2 V to 5.5 V, unless otherwise noted)

| Symbol | Parameter | Test conditions | | Limits | | | Unit | |
|-----------------------------------|--------------------------------------------------------------------------------------------|---------------------|-------------|-----------|------------------|------|------|----|
| | | | | Min. | Typ. | Max. | | |
| VOL | "L" level output voltage F0, F1, S0-S3, D0, D1, D2/C, D3/K | VDD = 5 V | IOL = 12 mA | | | 2 | V | |
| | | VDD = 3 V | IOL = 6 mA | | | 0.9 | V | |
| VOL | "L" level output voltage G0, G1/TOUT, G2, G3 | VDD = 5 V | IOL = 5 mA | | | 2 | V | |
| | | VDD = 3 V | IOL = 2 mA | | | 0.9 | V | |
| IiH | "H" level input current F0, F1, S0-S3, D0, D1, RESET | Vi = 7 V | | | | 1 | μA | |
| IiH | "H" level input current G0/INT, G1, G2, G3, D2/C, D3/K | Vi = VDD | | | | 1 | μA | |
| IiL | "L" level input current F0, F1, S0-S3, D0, D1, D2/C, D3/K, G0/INT, G1, G2, G3, RESET | Vi = 0 V (Note) | | | | -1 | μA | |
| IoZH | Output current at off-state F0, F1, S0-S3, D0, D1 | Vo = 7 V | | | | 1 | μA | |
| IoZH | Output current at off-state G0, G1/TOUT, G2, G3, D2/C, D3/K | Vo = VDD | | | | 1 | μA | |
| IDD | Supply current | at active mode | | VDD = 5 V | f(XIN) = 4.0 MHz | 1.5 | 5 | mA |
| | | | | VDD = 3 V | f(XIN) = 1.0 MHz | 0.3 | 1 | mA |
| | at RAM back-up mode | | Ta = 25 °C | | | 0.1 | 1 | μA |
| | | | VDD = 5 V | | | | 10 | μA |
| | | VDD = 3 V | | | | 6 | μA | |
| RPU | Pull-up transistor G0/INT, G1, G2, G3, D2/C, D3/K | VDD = 5 V, Vi = 0 V | | 5 | 11 | 25 | kΩ | |
| V _{T+} - V _{T-} | Hysteresis INT | | | | 0.3 | | V | |
| V _{T+} - V _{T-} | Hysteresis S0-S3 | VDD = 5 V | | 0.1 | | | V | |
| V _{T+} - V _{T-} | Hysteresis RESET | VDD = 5 V | | | 1.8 | | V | |
| | | VDD = 3 V | | | 0.7 | | V | |

Note: In this case, the pull-up transistors for G0/INT pin and ports G1, G2, G3, D2/C and D3/K are not selected.

BASIC TIMING DIAGRAM



EOL announced

BUILT-IN PROM VERSION

In addition to the mask ROM versions, the 4250 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 15 shows the product of built-in PROM version. Figure 31 and 32 show the pin configurations of built-in PROM versions. The One Time PROM version has pin-compatibility with the mask ROM version.

Table 15 Product of built-in PROM version

| Product | PROM size (X 9 bits) | RAM size (X 4 bits) | Package | ROM type |
|-------------------|----------------------|---------------------|---------|--------------------------------------------------------------------------------------|
| M34250E2-XXXXFP * | 2048 words | 64 words | 20P2N-A | One Time PROM [shipped after writing] (shipped after writing and test in factory) |
| M34250E2FP* | | | | One Time PROM [shipped in blank] |

*: Under development

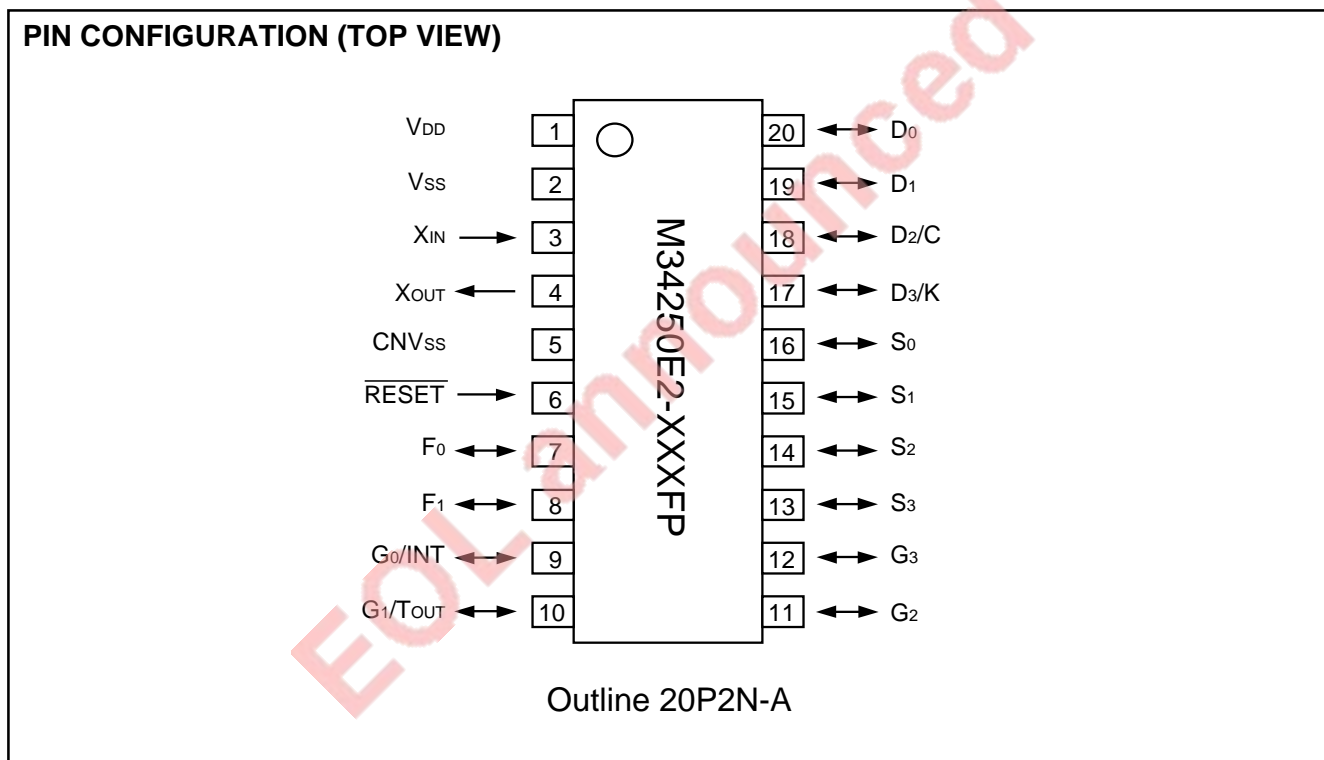
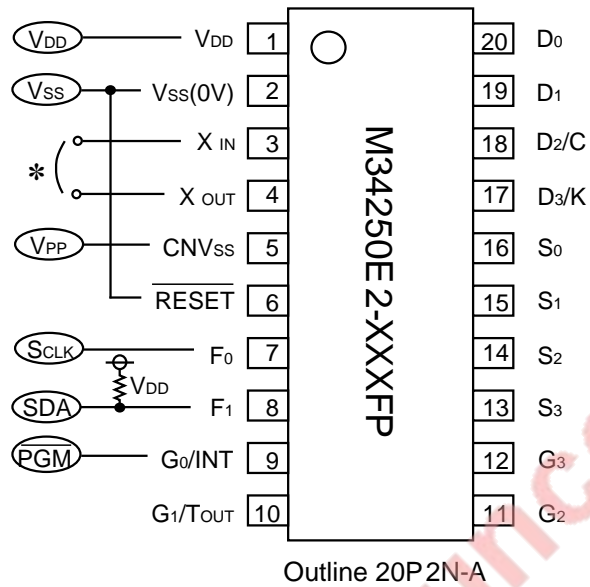


Fig. 31 Pin configuration of built-in PROM version

PIN CONFIGURATION (TOP VIEW)



* : A resistor is connected to X_{IN} pin.
 A capacitor is connected to X_{OUT} pin.
 Note: The state of each disconnected pin is the same as that at reset.

Fig. 32 Pin configuration of built-in PROM version (continued)

(1) PROM mode

The 4250 Group 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 setting pins SDA (serial data input/output), SCLK (serial clock input), and PGM to "H" after connecting wires as shown in Figure 32 and powering on the VDD pin, and then applying 12 V to the VPP 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). Use the special-purpose serial programmer when performing serial read/program.

Refer to the Mitsubishi Data Book "DEVELOPMENT SUPPORT TOOLS FOR MICROCOMPUTERS" about the serial programmer (serial programmer and control software, etc.) for the Mitsubishi single-chip microcomputers.

(2) Notes on handling

- ① A high-voltage is used for writing. Take care that overvoltage is not applied. Take care especially at turning on the power.
- ② For the One Time PROM version shipped in blank, Mitsubishi Electric 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 33 before using is recommended (Products shipped in blank: PROM contents is not written in factory when shipped)

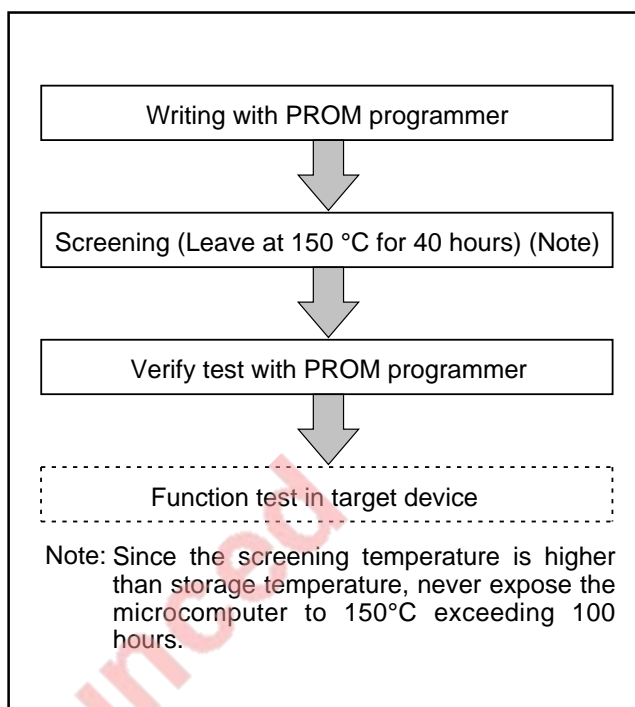


Fig. 33 Flow of writing and test of the product shipped in blank

EOL announcement

EOL announced

Renesas Technology Corp.

Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan

Keep safety first in your circuit designs!

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

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams and charts, represent information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
- Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

REVISION DESCRIPTION LIST

4250 GROUP DATA SHEET

| Rev. No. | Revision Description | Rev. date |
|----------------------------------------------------------------------------------------------------|----------------------|-----------|
| 1.0 | First Edition | 971130 |
| <p style="color: red; font-size: 48px; transform: rotate(-30deg); opacity: 0.5;">EOL announced</p> | | |