

### DESCRIPTION

The 4506 Group is a 4-bit single-chip microcomputer designed with CMOS technology. Its CPU is that of the 4500 series using a simple, high-speed instruction set. The computer is equipped with two 8-bit timers (each timer has a reload register), interrupts, and 10-bit A/D converter.

The various microcomputers in the 4506 Group include variations of the built-in memory size as shown in the table below.

### FEATURES

- Minimum instruction execution time ..... 0.68  $\mu$ s  
(at 4.4 MHz oscillation frequency, in high-speed mode)
- Supply voltage ..... 2.0 V to 5.5 V  
(It depends on the oscillation frequency and operating mode.)

- Timers
  - Timer 1 ..... 8-bit timer with a reload register
  - Timer 2 ..... 8-bit timer with a reload register
- Interrupt ..... 4 sources
- Key-on wakeup function pins ..... 12
- Input/Output port ..... 14
- A/D converter ..... 10-bit successive comparison method
- Watchdog timer
- Clock generating circuit (ceramic resonator/RC oscillation)
- LED drive directly enabled (port D)

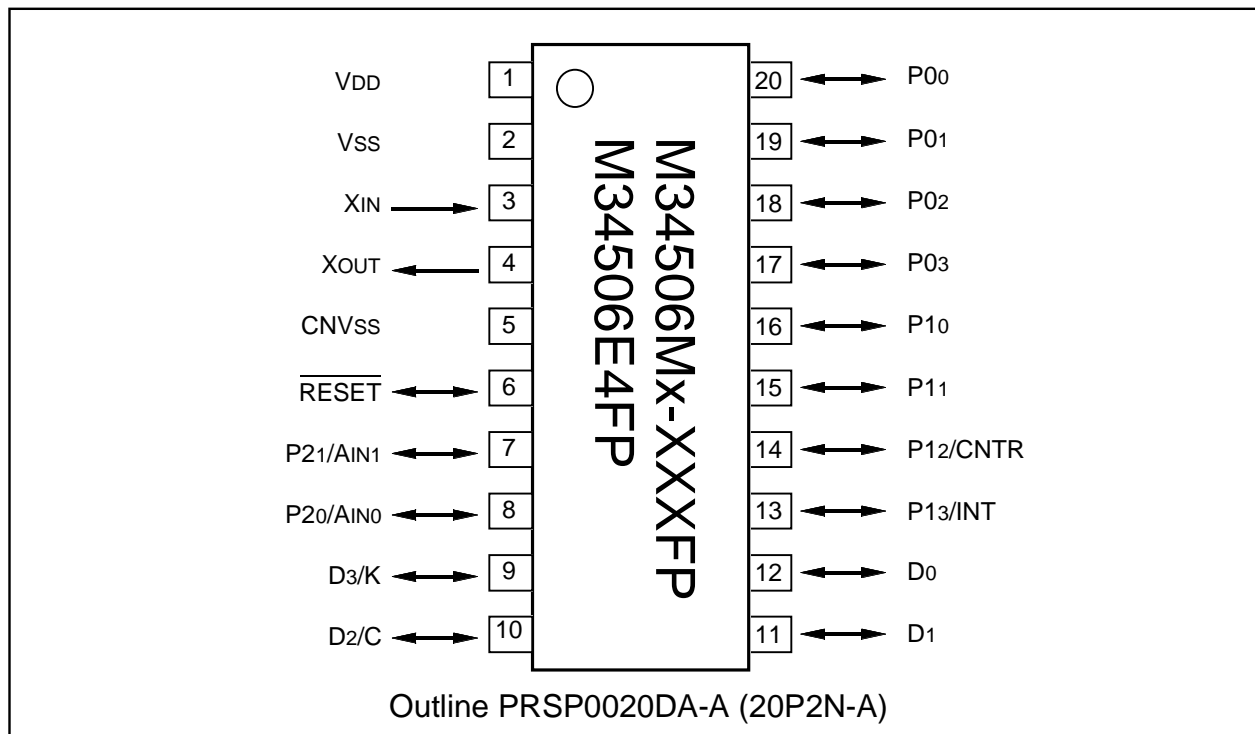
### APPLICATION

Electrical household appliance, consumer electronic products, office automation equipment, etc.

Part number	ROM (PROM) size (X 10 bits)	RAM size (X 4 bits)	Package	ROM type
M34506M2-XXXXFP	2048 words	128 words	PRSP0020DA-A	Mask ROM
M34506M4-XXXXFP	4096 words	256 words	PRSP0020DA-A	Mask ROM
M34506E4FP (Note)	4096 words	256 words	PRSP0020DA-A	One Time PROM

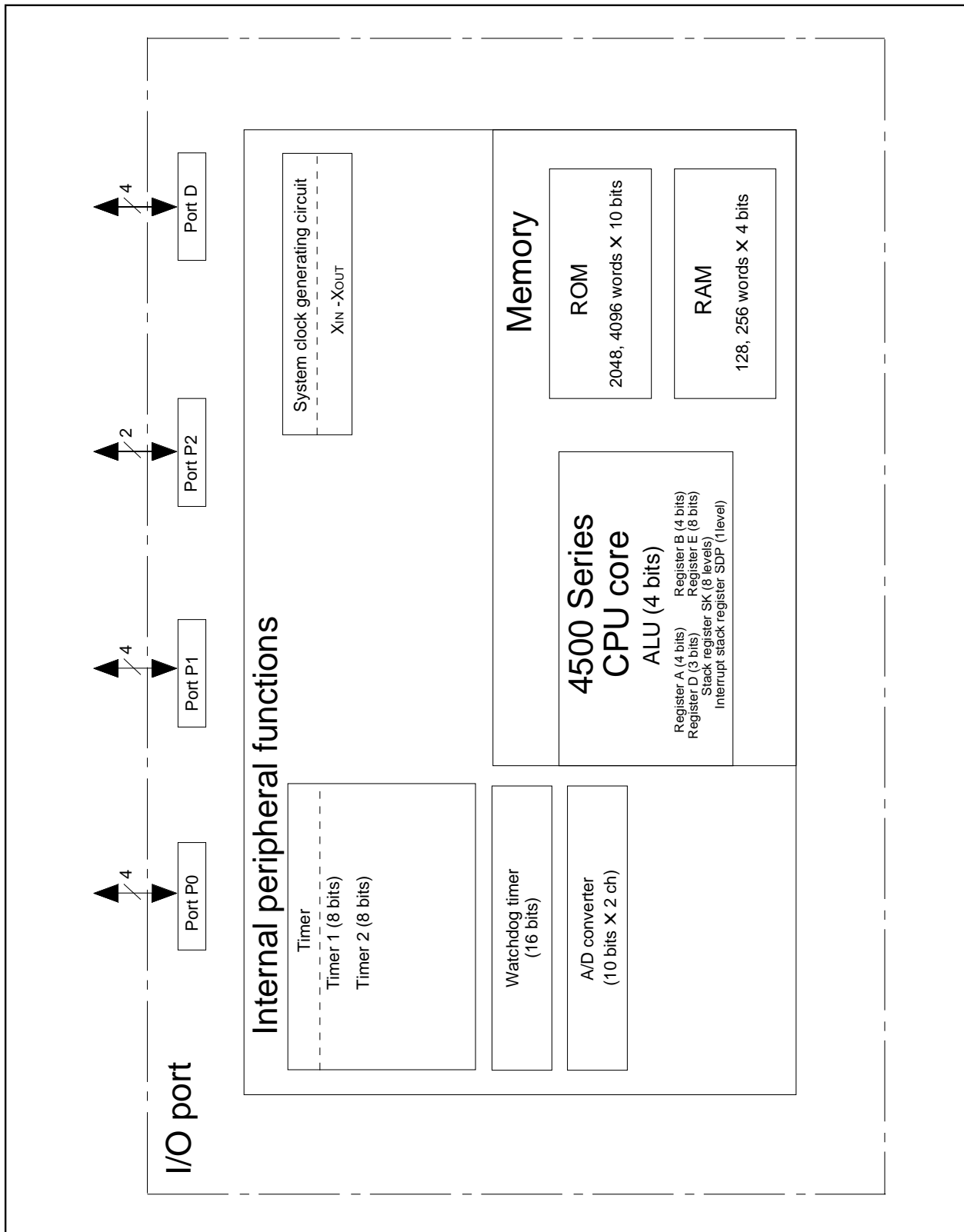
**Note:** Shipped in blank.

### PIN CONFIGURATION



Pin configuration (top view) (4506 Group)

**BLOCK DIAGRAM**



Block diagram (4506 Group)

## PERFORMANCE OVERVIEW

Parameter		Function	
Number of basic instructions		110	
Minimum instruction execution time		0.68 $\mu$ s (at 4.4 MHz oscillation frequency, in high-speed mode)	
Memory sizes	ROM	M34506M2	2048 words X 10 bits
		M34506M4/E4	4096 words X 10 bits
	RAM	M34506M2	128 words X 4 bits
		M34506M4/E4	256 words X 4 bits
Input/Output ports	D0–D3	I/O	Four independent I/O ports . Input is examined by skip decision. Ports D2 and D3 are equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software. Ports D2 and D3 are also used as ports C and K, respectively.
	P00–P03	I/O	4-bit I/O port; each pin is equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software.
	P10–P13	I/O	4-bit I/O port; each pin is equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software. Ports P12 and P13 are also used as CNTR and INT, respectively.
	P20, P21	I/O	2-bit I/O port; each pin is equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software. Ports P20 and P21 are also used as AIN0 and AIN1, respectively.
	C	I/O	1-bit I/O; Port C is also used as port D2.
	K	I/O	1-bit I/O; Port K is also used as port D3.
	CNTR	Timer I/O	1-bit I/O; CNTR pin is also used as port P12.
	INT	Interrupt input	1-bit input; INT pin is also used as port P13.
	AIN0, AIN1	Analog input	Two independent I/O ports; AIN0, AIN1 are also used as P20 and P21, respectively.
Timers	Timer 1		8-bit programmable timer with a reload register.
	Timer 2		8-bit programmable timer with a reload register and has a event counter.
A/D converter		10-bit wide, This is equipped with an 8-bit comparator function.	
Interrupt	Analog input		2 channel (AIN0 pin, AIN1 pin)
	Sources	4 (one for external, two for timer, one for A/D)	
	Nesting	1 level	
Subroutine nesting		8 levels	
Device structure		CMOS silicon gate	
Package		20-pin plastic molded SOP (PRSP0020DA-A)	
Operating temperature range		–20 °C to 85 °C	
Supply voltage		2.0 V to 5.5 V (It depends on the oscillation frequency and operating mode. Refer to the recommended operating condition.)	
Power dissipation (typical value)	Active mode		1.7 mA (Ta=25°C, VDD = 5.0 V, 4.0 MHz oscillation frequency, in high-speed mode, output transistors in the cut-off state)
			0.5 mA (Ta=25°C, VDD = 3.0 V, 2.0 MHz oscillation frequency, in high-speed mode, output transistors in the cut-off state)
	RAM back-up mode		0.1 $\mu$ A (Ta=25°C, VDD = 5 V, output transistors in the cut-off state)

**PIN DESCRIPTION**

Pin	Name	Input/Output	Function
VDD	Power supply	—	Connected to a plus power supply.
VSS	Ground	—	Connected to a 0 V power supply.
CNVSS	CNVSS	—	Connect CNVSS to VSS and apply "L" (0V) to CNVSS certainly.
RESET	Reset input/output	I/O	An N-channel open-drain I/O pin for a system reset. When the watchdog timer or the built-in power-on reset causes the system to be reset, the RESET pin outputs "L" level.
XIN	System clock input	Input	I/O pins of the system clock generating circuit. When using a ceramic resonator, connect it between pins XIN and XOUT. A feedback resistor is built-in between them. When using the RC oscillation, connect a resistor and a capacitor to XIN, and leave XOUT pin open.
XOUT	System clock output	Output	
D0–D3	I/O port D	I/O	Each pin of port D has an independent 1-bit wide I/O function. Each pin has an output latch. For input use, set the latch of the specified bit to "1." Input is examined by skip decision. The output structure is N-channel open-drain. Ports D2 and D3 are equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software. Ports D2 and D3 are also used as ports C and K, respectively.
P00–P03	I/O port P0	I/O	Port P0 serves as a 4-bit I/O port, and it can be used as inputs when the output latch is set to "1." The output structure is N-channel open-drain. Port P0 has a key-on wakeup function and a pull-up function. Both functions can be switched by software.
P10–P13	I/O port P1	I/O	Port P1 serves as a 4-bit I/O port, and it can be used as inputs when the output latch is set to "1." The output structure is N-channel open-drain. Port P1 has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports P12 and P13 are also used as CNTR and INT, respectively.
P20, P21	I/O port P2	I/O	Port P2 serves as a 2-bit I/O port, and it can be used as inputs when the output latch is set to "1." The output structure is N-channel open-drain. Port P2 has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports P20 and P21 are also used as AIN0 and AIN1, respectively.
Port C	I/O port C	I/O	1-bit I/O port. Port C can be used as inputs when the output latch is set to "1." The output structure is N-channel open-drain. Port C has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Port C is also used as port D2.
Port K	I/O port K	I/O	1-bit I/O port. Port K can be used as inputs when the output latch is set to "1." The output structure is N-channel open-drain. Port K has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Port K is also used as port D3.
CNTR	Timer input/output	I/O	CNTR pin has the function to input the clock for the timer 2 event counter, and to output the timer 1 or timer 2 underflow signal divided by 2. This pin is also used as port P12.
INT	Interrupt input	Input	INT pin accepts external interrupts. It has the key-on wakeup function which can be switched by software. This pin is also used as port P13.
AIN0–AIN1	Analog input	Input	A/D converter analog input pins. AIN0 and AIN1 are also used as ports P20 and P21, respectively.

**MULTIFUNCTION**

Pin	Multifunction	Pin	Multifunction	Pin	Multifunction	Pin	Multifunction
D2	C	C	D2	P20	AIN0	AIN0	P20
D3	K	K	D3	P21	AIN1	AIN1	P21
P12	CNTR	CNTR	P12				
P13	INT	INT	P13				

Notes 1: Pins except above have just single function.

2: The input/output of D2, D3, P12 and P13 can be used even when C, K, CNTR (input) and INT are selected.

3: The input of P12 can be used even when CNTR (output) is selected.

4: The input/output of P20, P21 can be used even when AIN0, AIN1 are selected.

## DEFINITION OF CLOCK AND CYCLE

### ● Operation source clock

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

- External ceramic resonator
- External RC oscillation
- Clock ( $f(X_{IN})$ ) by the external clock
- Clock ( $f(RING)$ ) of the on-chip oscillator which is the internal oscillator.

### ● System clock

The system clock is the basic clock for controlling this product. The system clock is selected by the bits 2 and 3 of the clock control register MR.

### ● Instruction clock

The instruction clock is a signal derived by dividing the system clock by 3. The one instruction clock cycle generates the one machine cycle.

### ● Machine cycle

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

**Table Selection of system clock**

Register MR		System clock (Note 1)	Operation mode
MR3	MR2		
0	0	$f(X_{IN})$ or $f(RING)$	High-speed mode
0	1	$f(X_{IN})/2$ or $f(RING)/2$	Middle-speed mode
1	0	$f(X_{IN})/4$ or $f(RING)/4$	Low-speed mode
1	1	$f(X_{IN})/8$ or $f(RING)/8$	Default mode

**Notes 1:** The on-chip oscillator clock is  $f(RING)$ , the clock by the ceramic resonator, RC oscillation or external clock is  $f(X_{IN})$ .

**2:** The default mode is selected after system is released from reset and is returned from RAM back-up.

## PORT FUNCTION

Port	Pin	Input Output	Output structure	I/O unit	Control instructions	Control registers	Remark
Port D	D0, D1 D2/C D3/K	I/O (4)	N-channel open-drain	1	SD, RD SZD, CLD SCP, RCP SNZCP IAK, OKA	PU2, K2	Built-in programmable pull-up functions Key-on wakeup functions (programmable)
Port P0	P00–P03	I/O (4)	N-channel open-drain	4	OP0A IAP0	PU0, K0	Built-in programmable pull-up functions Key-on wakeup functions (programmable)
Port P1	P10, P11 P12/CNTR, P13/INT	I/O (4)	N-channel open-drain	4	OP1A IAP1	PU1, K1 W6, I1	Built-in programmable pull-up functions Key-on wakeup functions (programmable)
Port P2	P20/AIN0 P21/AIN1	I/O (2)	N-channel open-drain	2	OP2A IAP2	PU2, K2 Q1	Built-in programmable pull-up functions Key-on wakeup functions (programmable)

**CONNECTIONS OF UNUSED PINS**

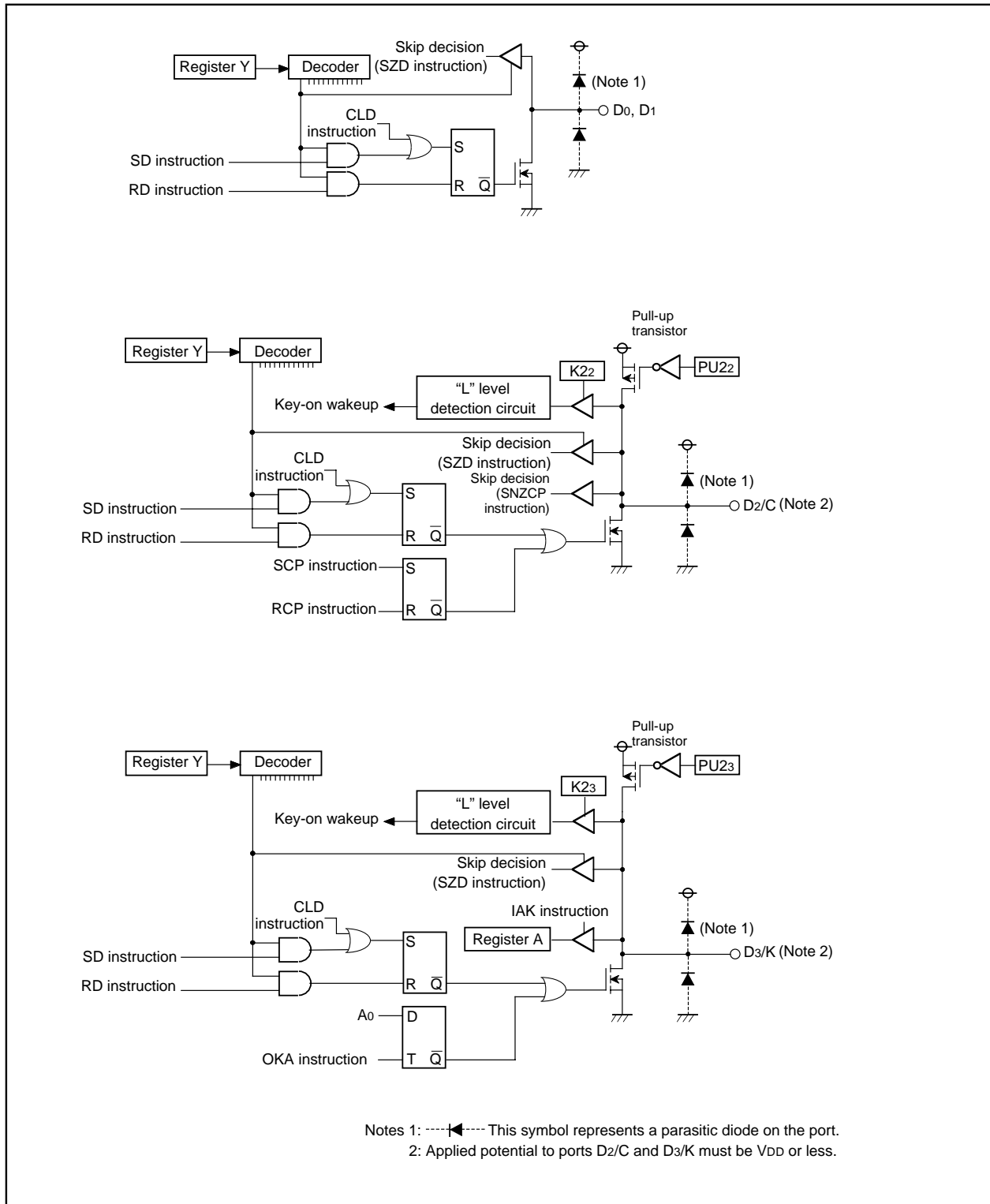
Pin	Connection	Usage condition
XIN	Connect to Vss.	System operates by the on-chip oscillator. (Note 1)
XOUT	Open.	System operates by the external clock. (The ceramic resonator is selected with the CMCK instruction.)
		System operates by the RC oscillator. (The RC oscillation is selected with the CRCK instruction.)
		System operates by the on-chip oscillator. (Note 1)
D0, D1	Open. (Output latch is set to "1.")	—————
	Open. (Output latch is set to "0.")	—————
	Connect to Vss.	—————
D2/C D3/K	Open. (Output latch is set to "1.")	The key-on wakeup function is not selected. (Note 4)
	Open. (Output latch is set to "0.")	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
	Connect to Vss.	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
P00–P03	Open. (Output latch is set to "1.")	The key-on wakeup function is not selected. (Note 4)
	Open. (Output latch is set to "0.")	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
	Connect to Vss.	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
P10, P11 P12/CNTR	Open. (Output latch is set to "1.")	The key-on wakeup function is not selected. (Note 4)
	Open. (Output latch is set to "0.")	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
	Connect to Vss.	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
P13/INT	Open. (Output latch is set to "1.")	The key-on wakeup function is not selected. The input to INT pin is disabled. (Notes 4, 5)
	Open. (Output latch is set to "0.")	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
	Connect to Vss.	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
P20/AIN0 P21/AIN1	Open. (Output latch is set to "1.")	The key-on wakeup function is not selected. (Note 4)
	Open. (Output latch is set to "0.")	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
	Connect to Vss.	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)

- Notes 1: When the ceramic resonator or the RC oscillation is not selected by program, system operates by the on-chip oscillator (internal oscillator).  
 2: When the pull-up function is left valid, the supply current is increased. Do not select the pull-up function.  
 3: When the key-on wakeup function is left valid, the system returns from the RAM back-up state immediately after going into the RAM back-up state.  
 Do not select the key-on wakeup function.  
 4: When selecting the key-on wakeup function, select also the pull-up function.  
 5: Clear the bit 3 (I13) of register I1 to "0" to disable to input to INT pin (after reset: I13 = "0")

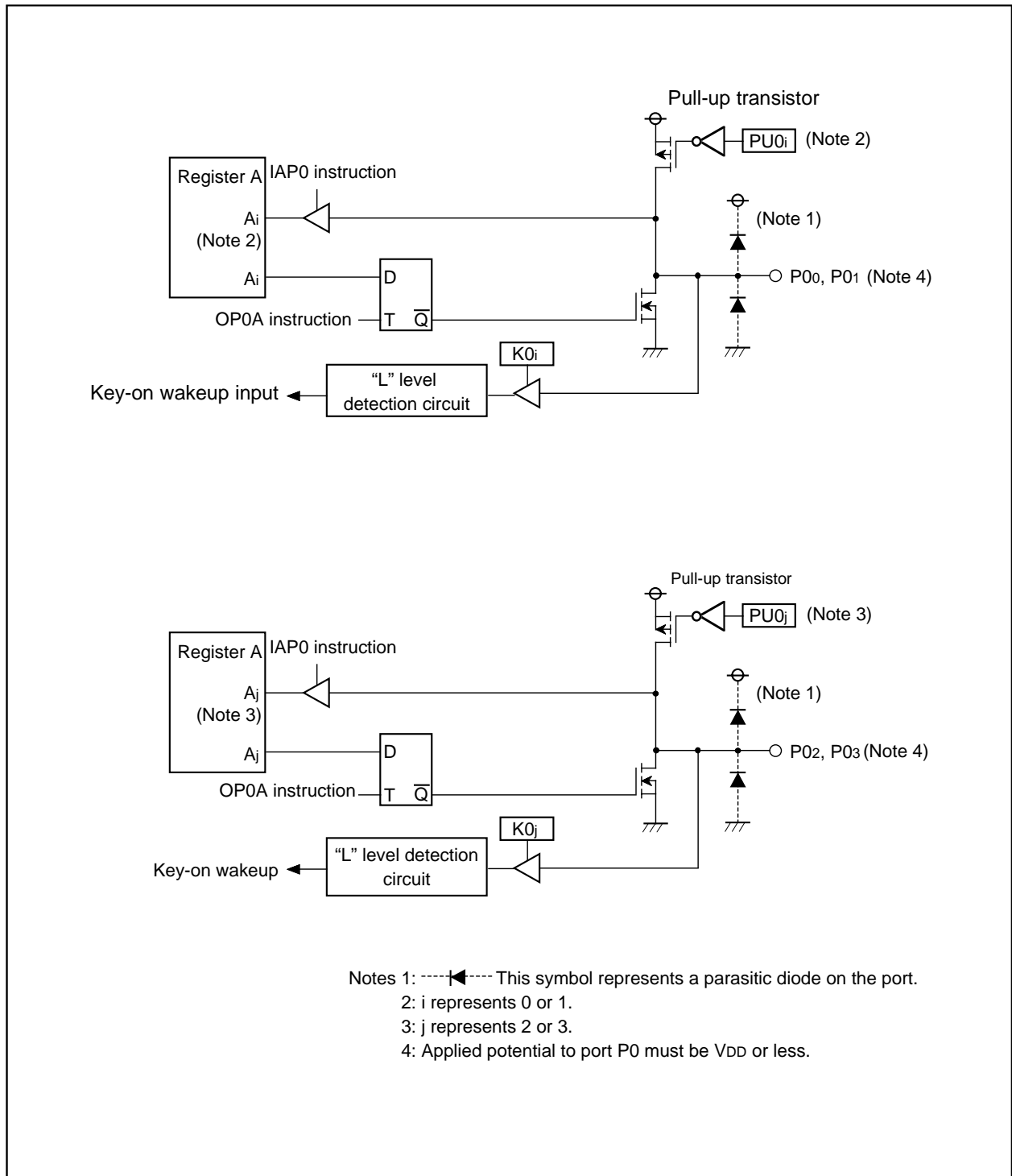
(Note when connecting to Vss)

- Connect the unused pins to Vss using the thickest wire at the shortest distance against noise.

**PORT BLOCK DIAGRAMS**

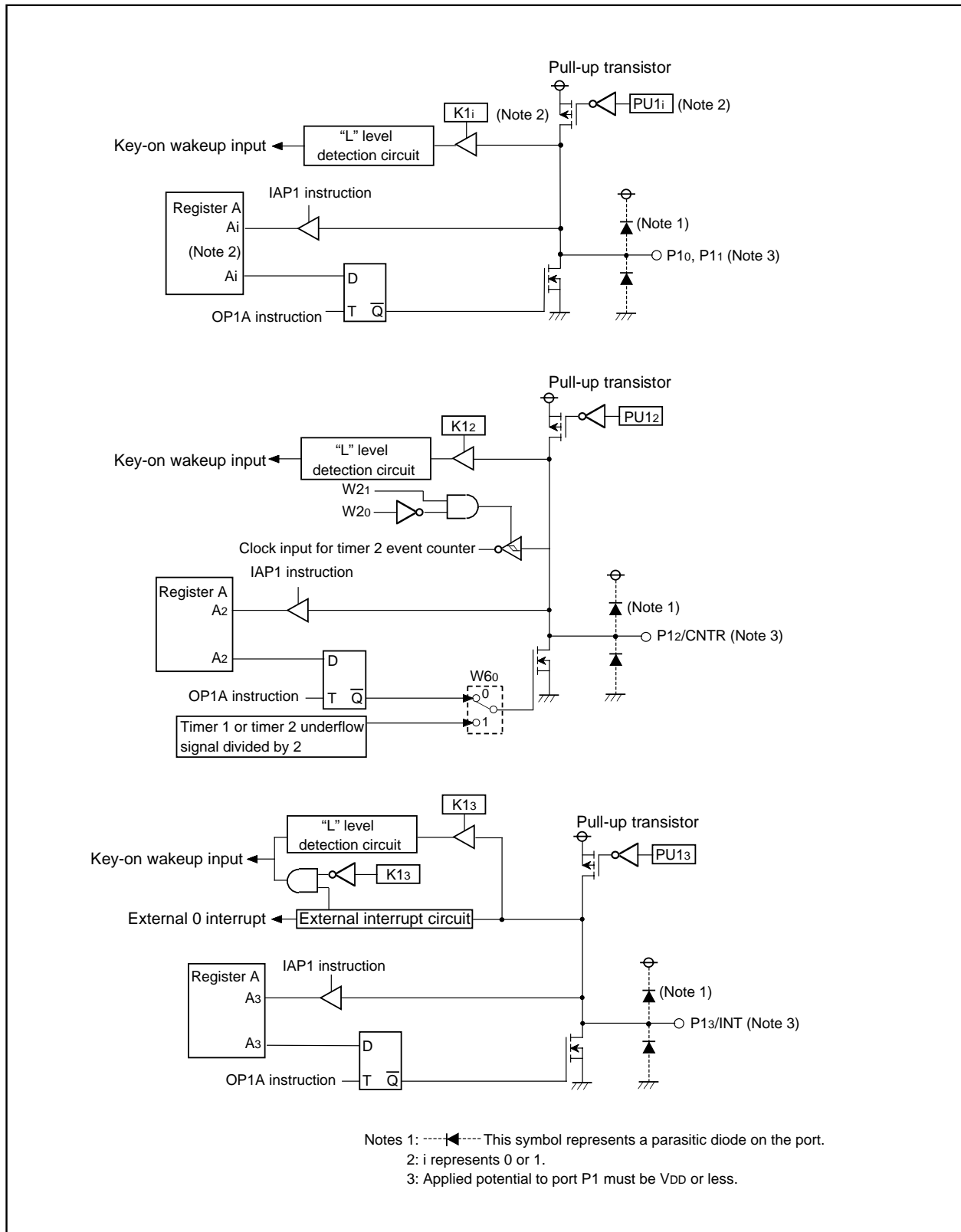


Port block diagram (1)

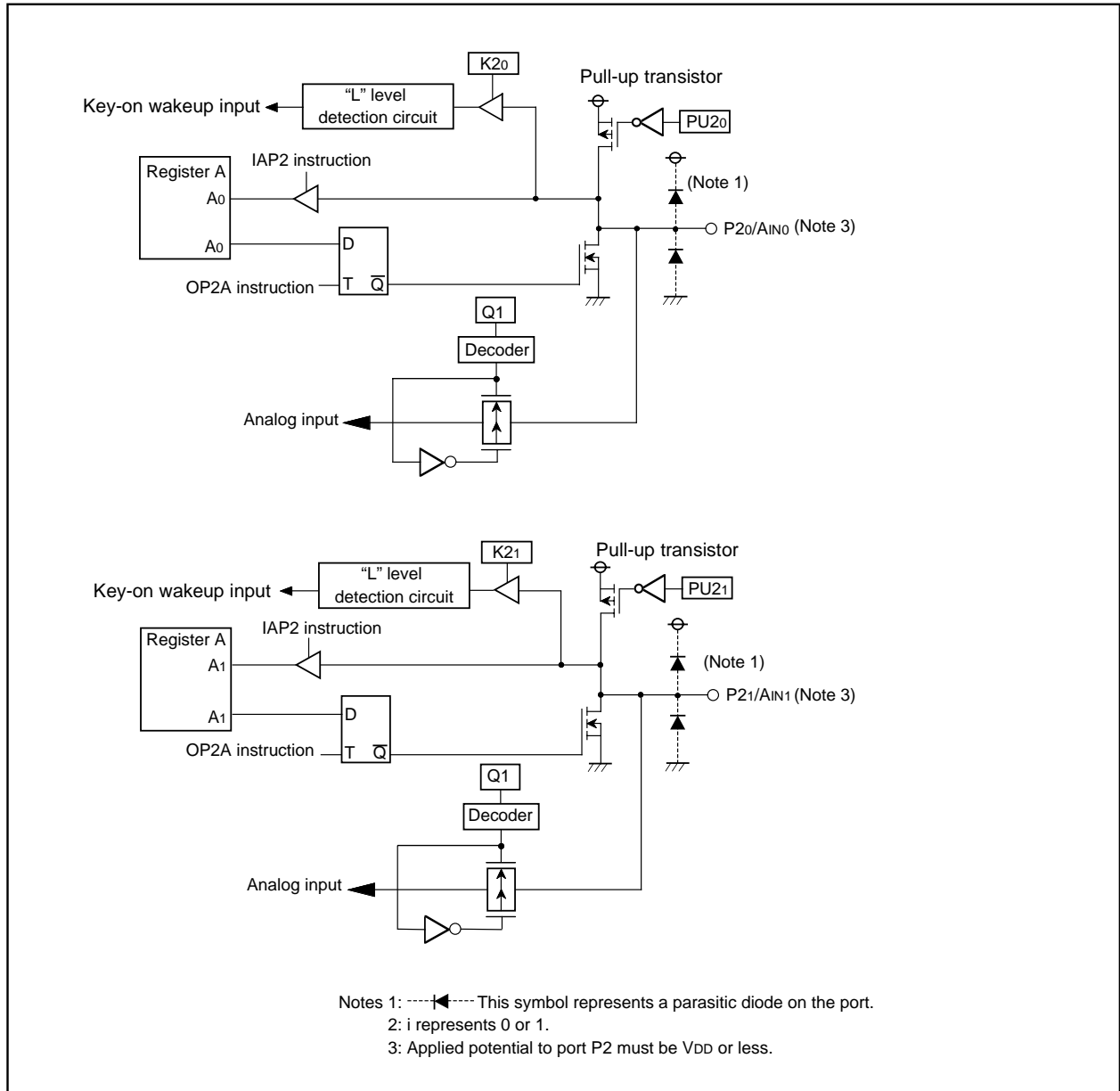


Port block diagram (2)

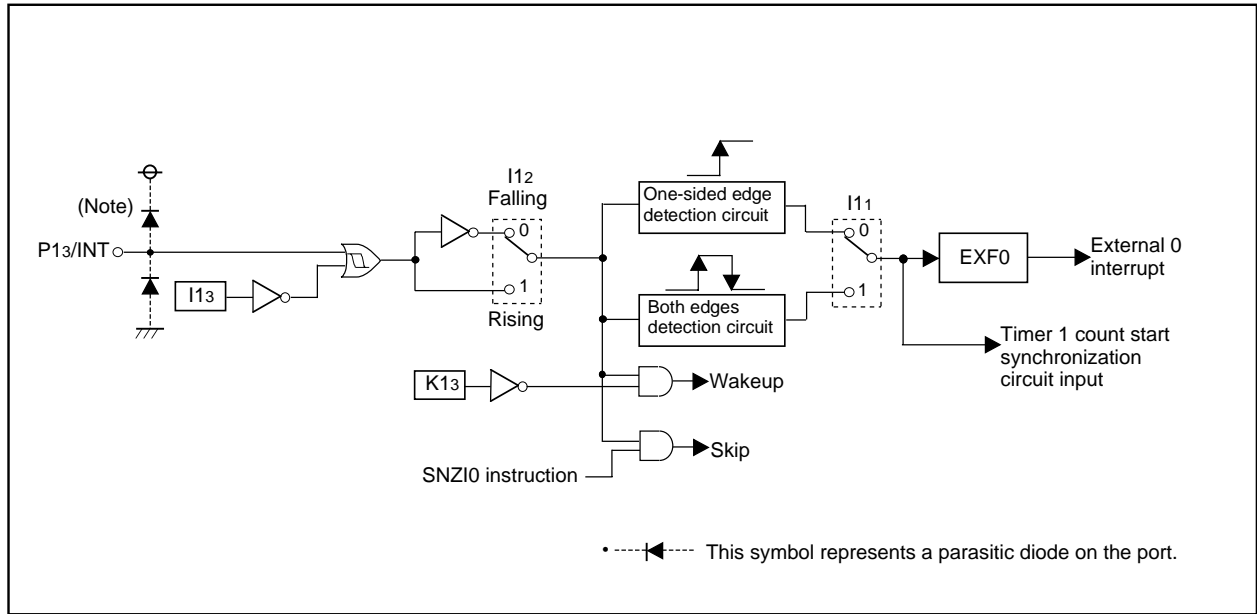




Port block diagram (3)



Port block diagram (4)



External interrupt circuit structure

## FUNCTION BLOCK OPERATIONS CPU

### (1) Arithmetic logic unit (ALU)

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

### (2) Register A and carry flag

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

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

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

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

### (3) Registers B and E

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

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

Register E is undefined after system is released from reset and returned from the RAM back-up. Accordingly, set the initial value.

### (4) Register D

Register D is a 3-bit register.

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

Register D is undefined after system is released from reset and returned from the RAM back-up. Accordingly, set the initial value.

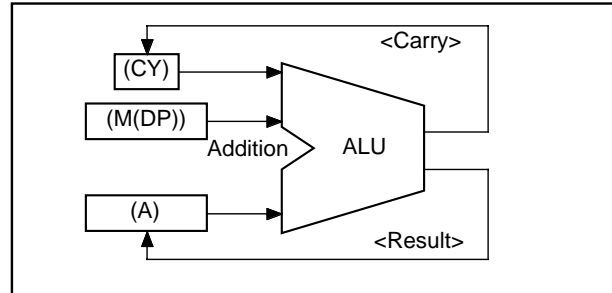


Fig. 1 AMC instruction execution example

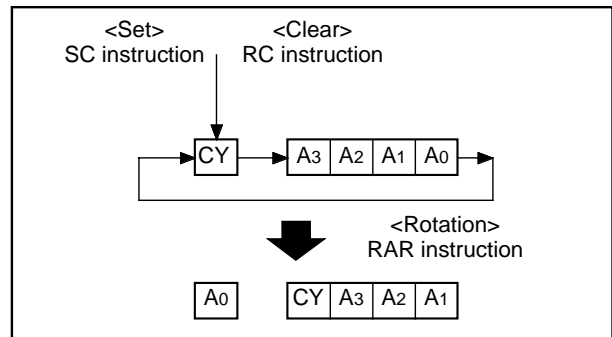


Fig. 2 RAR instruction execution example

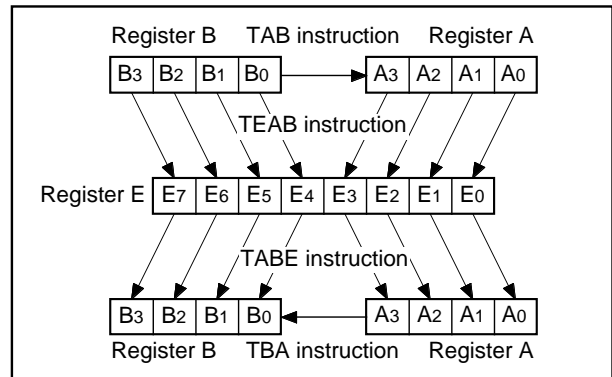


Fig. 3 Registers A, B and register E

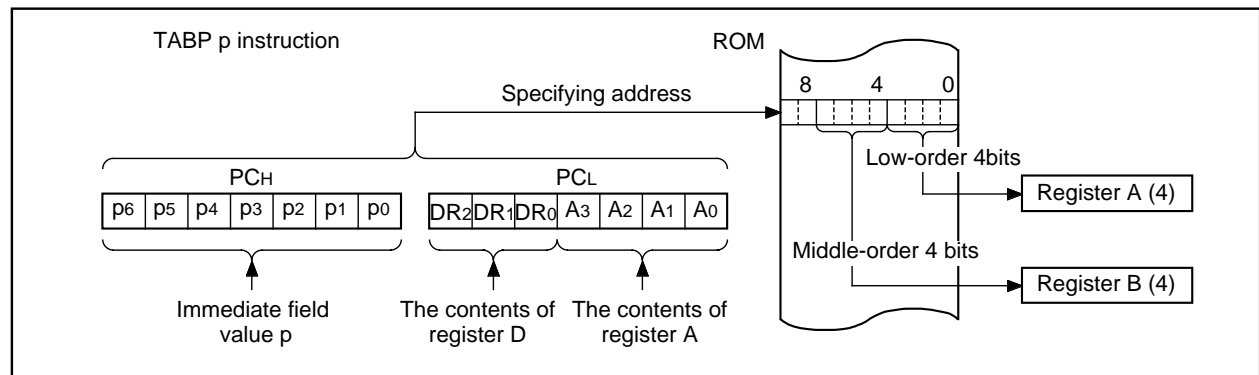


Fig. 4 TABP p instruction execution example

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

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

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

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

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

Figure 5 shows the stack registers (SKs) structure.

Figure 6 shows the example of operation at subroutine call.

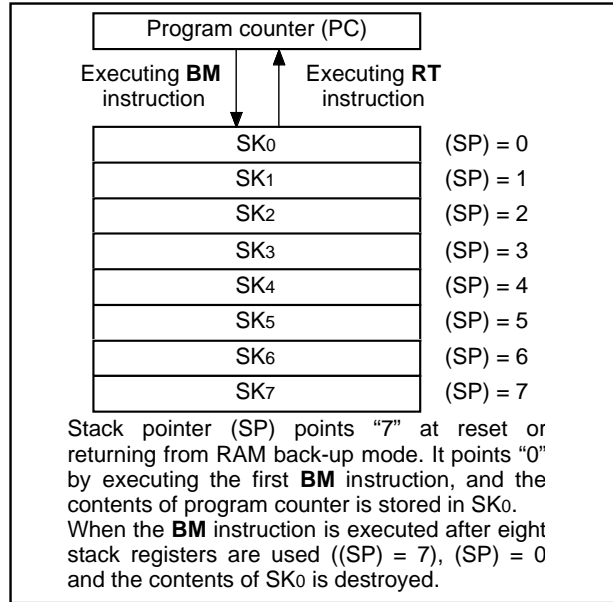
**(6) Interrupt stack register (SDP)**

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

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

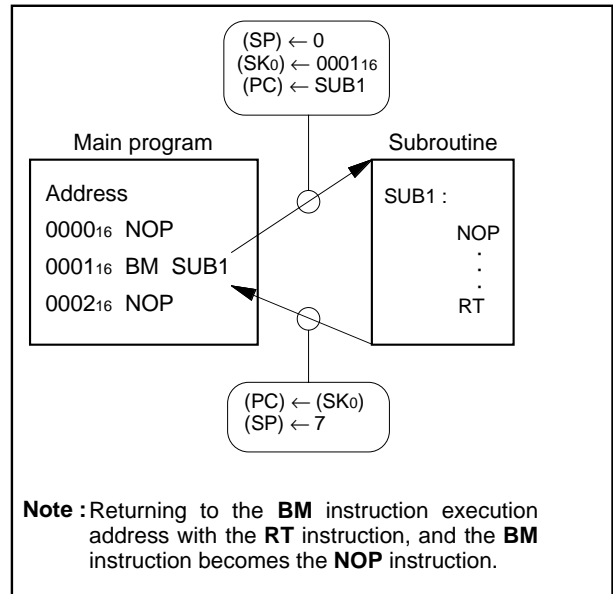
**(7) Skip flag**

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



Stack pointer (SP) points "7" at reset or returning from RAM back-up mode. It points "0" by executing the first **BM** instruction, and the contents of program counter is stored in SK0. When the **BM** instruction is executed after eight stack registers are used ((SP) = 7), (SP) = 0 and the contents of SK0 is destroyed.

Fig. 5 Stack registers (SKs) structure



**Note :** Returning to the **BM** instruction execution address with the **RT** instruction, and the **BM** instruction becomes the **NOP** instruction.

Fig. 6 Example of operation at subroutine call

**(8) Program counter (PC)**

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

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

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

**(9) Data pointer (DP)**

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

Register Y is also used to specify the port D bit position.

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

• Note

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

Also, registers Z, X and Y are undefined in the RAM back-up. After system is returned from the RAM back-up, set these registers.

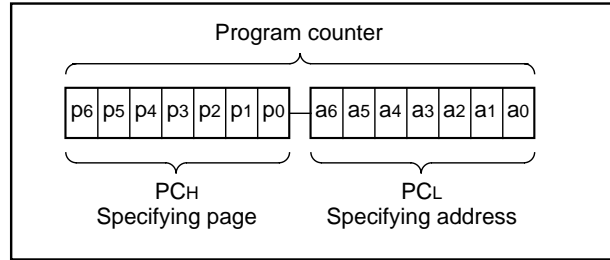


Fig. 7 Program counter (PC) structure

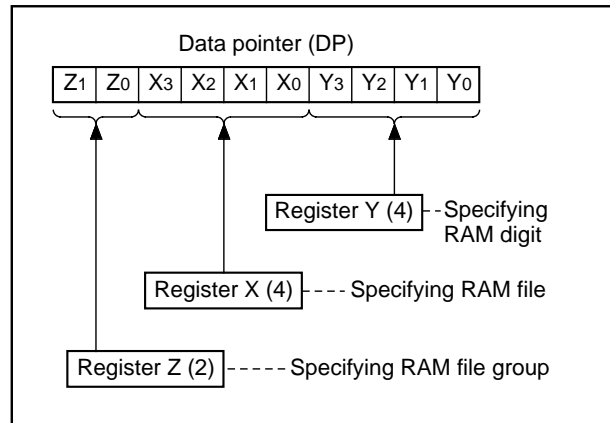


Fig. 8 Data pointer (DP) structure

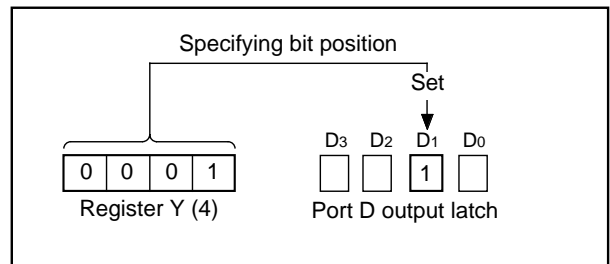


Fig. 9 SD instruction execution example

### PROGRAM MEMORY (ROM)

The program memory is a mask ROM. 1 word of ROM is composed of 10 bits. ROM is separated every 128 words by the unit of page (addresses 0 to 127). Table 1 shows the ROM size and pages. Figure 10 shows the ROM map of M34506M4.

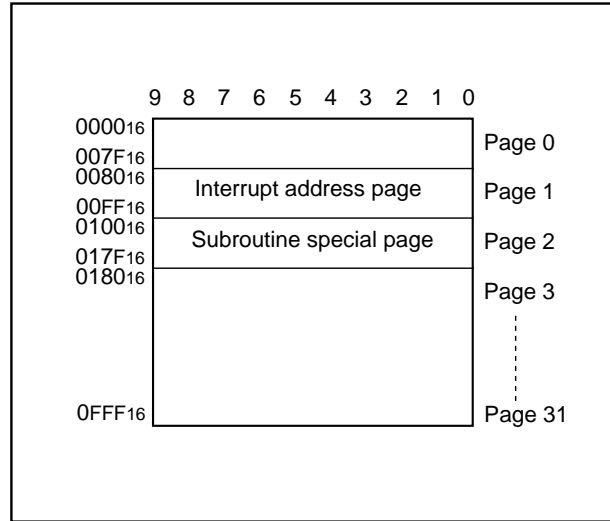
**Table 1 ROM size and pages**

Part number	ROM (PROM) size (X 10 bits)	Pages
M34506M2	2048 words	16 (0 to 15)
M34506M4	4096 words	32 (0 to 31)
M34506E4	4096 words	32 (0 to 31)

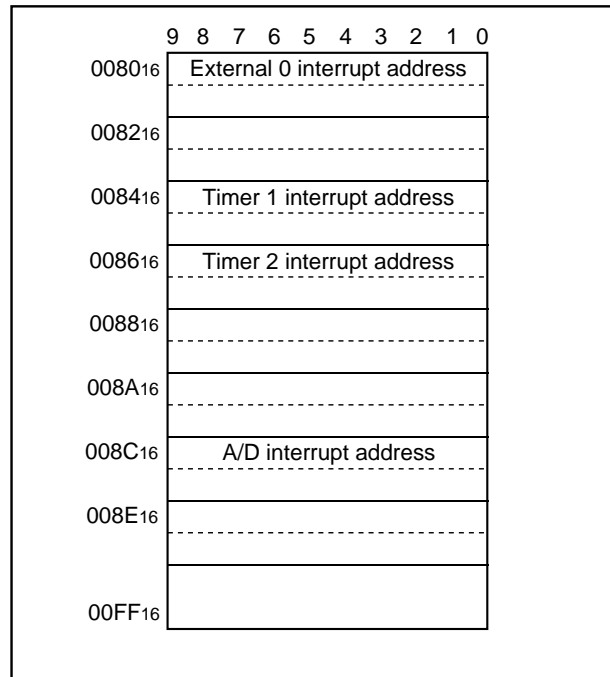
A part of page 1 (addresses 0080<sub>16</sub> to 00FF<sub>16</sub>) is reserved for interrupt addresses (Figure 11). When an interrupt occurs, the address (interrupt address) corresponding to each interrupt is set in the program counter, and the instruction at the interrupt address is executed. When using an interrupt service routine, write the instruction generating the branch to that routine at an interrupt address.

Page 2 (addresses 0100<sub>16</sub> to 017F<sub>16</sub>) 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 instruction.



**Fig. 10 ROM map of M34506M4/M34506E4**



**Fig. 11 Page 1 (addresses 0080<sub>16</sub> to 00FF<sub>16</sub>) structure**

### 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 Z, 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.

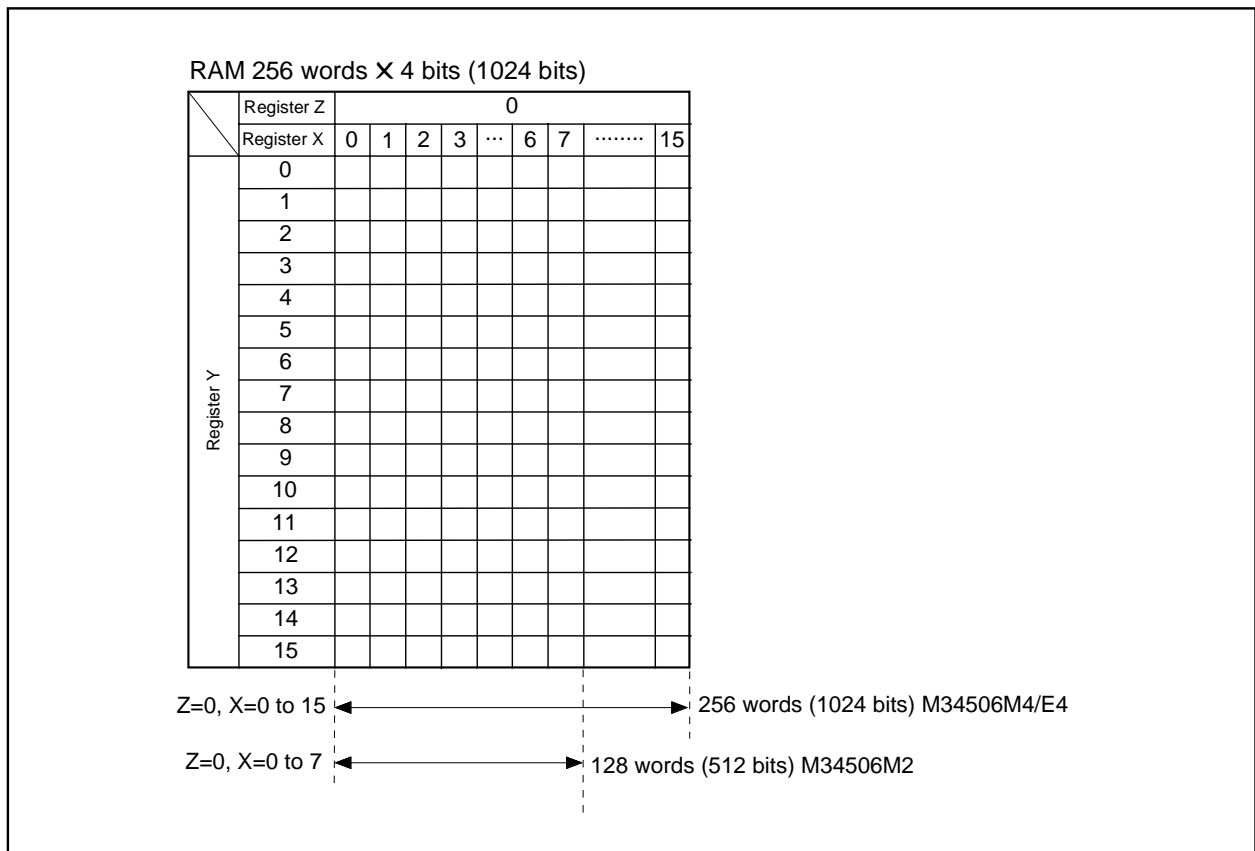
**Table 2 RAM size**

Part number	RAM size
M34506M2	128 words X 4 bits (512 bits)
M34506M4	256 words X 4 bits (1024 bits)
M34506E4	256 words X 4 bits (1024 bits)

• Note

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

Also, registers Z, X and Y are undefined in the RAM back-up. After system is returned from the RAM back-up, set these registers.



**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 is enabled ("1")
- Interrupt enable flag is enabled (INTE = "1")

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

### (1) Interrupt enable flag (INTE)

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

### (2) Interrupt enable bit

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

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

Table 5 shows the interrupt enable bit function.

### (3) Interrupt request flag

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

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

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

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

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

**Table 3 Interrupt sources**

Priority level	Interrupt name	Activated condition	Interrupt address
1	External 0 interrupt	Level change of INT pin	Address 0 in page 1
2	Timer 1 interrupt	Timer 1 underflow	Address 4 in page 1
3	Timer 2 interrupt	Timer 2 underflow	Address 6 in page 1
4	A/D interrupt	Completion of A/D conversion	Address C in page 1

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

Interrupt name	Interrupt request flag	Skip instruction	Interrupt enable bit
External 0 interrupt	EXF0	SNZ0	V10
Timer 1 interrupt	T1F	SNZT1	V12
Timer 2 interrupt	T2F	SNZT2	V13
A/D interrupt	ADF	SNZAD	V22

**Table 5 Interrupt enable bit function**

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

### (4) Internal state during an interrupt

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

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

### (5) Interrupt processing

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

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

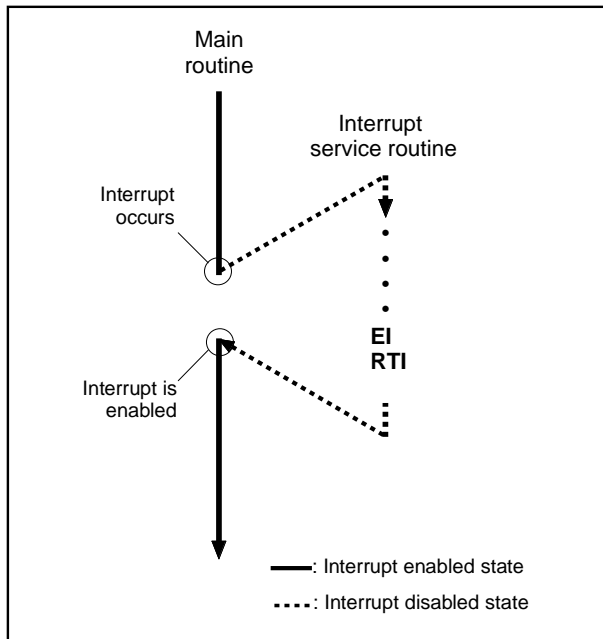


Fig. 13 Program example of interrupt processing

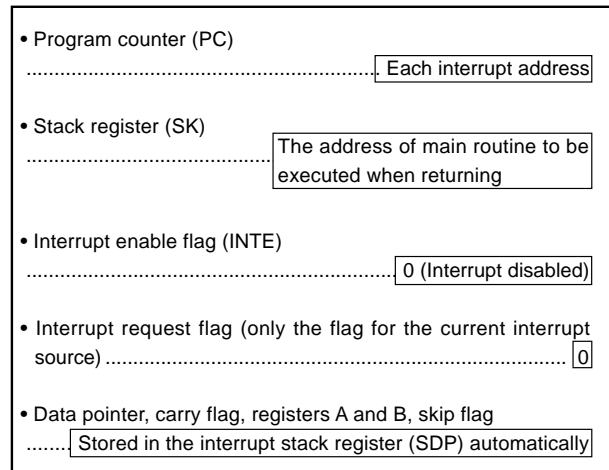


Fig. 14 Internal state when interrupt occurs

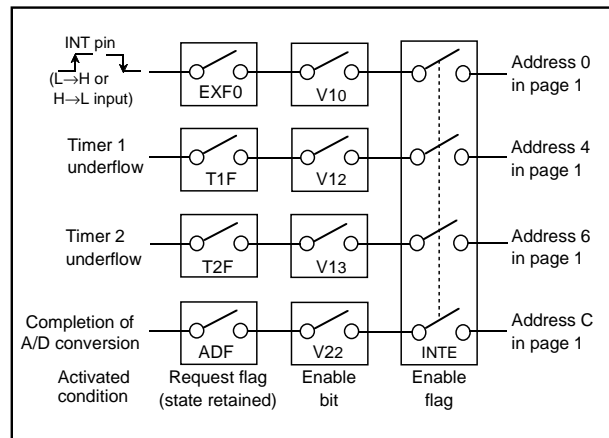


Fig. 15 Interrupt system diagram

## (6) Interrupt control registers

- Interrupt control register V1

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

- Interrupt control register V2

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

**Table 6** Interrupt control registers

Interrupt control register V1		at reset : 0000 <sub>2</sub>	at RAM back-up : 0000 <sub>2</sub>	R/W
V13	Timer 2 interrupt enable bit	0	Interrupt disabled (SNZT2 instruction is valid)	
		1	Interrupt enabled (SNZT2 instruction is invalid) (Note 2)	
V12	Timer 1 interrupt enable bit	0	Interrupt disabled (SNZT1 instruction is valid)	
		1	Interrupt enabled (SNZT1 instruction is invalid) (Note 2)	
V11	Not used	0	This bit has no function, but read/write is enabled.	
		1		
V10	External 0 interrupt enable bit	0	Interrupt disabled (SNZ0 instruction is valid)	
		1	Interrupt enabled (SNZ0 instruction is invalid) (Note 2)	

Interrupt control register V2		at reset : 0000 <sub>2</sub>	at RAM back-up : 0000 <sub>2</sub>	R/W
V23	Not used	0	This bit has no function, but read/write is enabled.	
		1		
V22	A/D interrupt enable bit	0	Interrupt disabled (SNZAD instruction is valid)	
		1	Interrupt enabled (SNZAD instruction is invalid) (Note 2)	
V21	Not used	0	This bit has no function, but read/write is enabled.	
		1		
V20	Not used	0	This bit has no function, but read/write is enabled.	
		1		

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

**2:** These instructions are equivalent to the NOP instruction.

## (7) Interrupt sequence

Interrupts only occur when the respective INTE flag, interrupt enable bits (V10, V12, V13, V22), and interrupt request flag are "1." The interrupt actually occurs 2 to 3 machine cycles after the cycle in which all three conditions are satisfied. The interrupt occurs after 3 machine cycles only when the three interrupt conditions are satisfied on execution of other than one-cycle instructions (Refer to Figure 16).

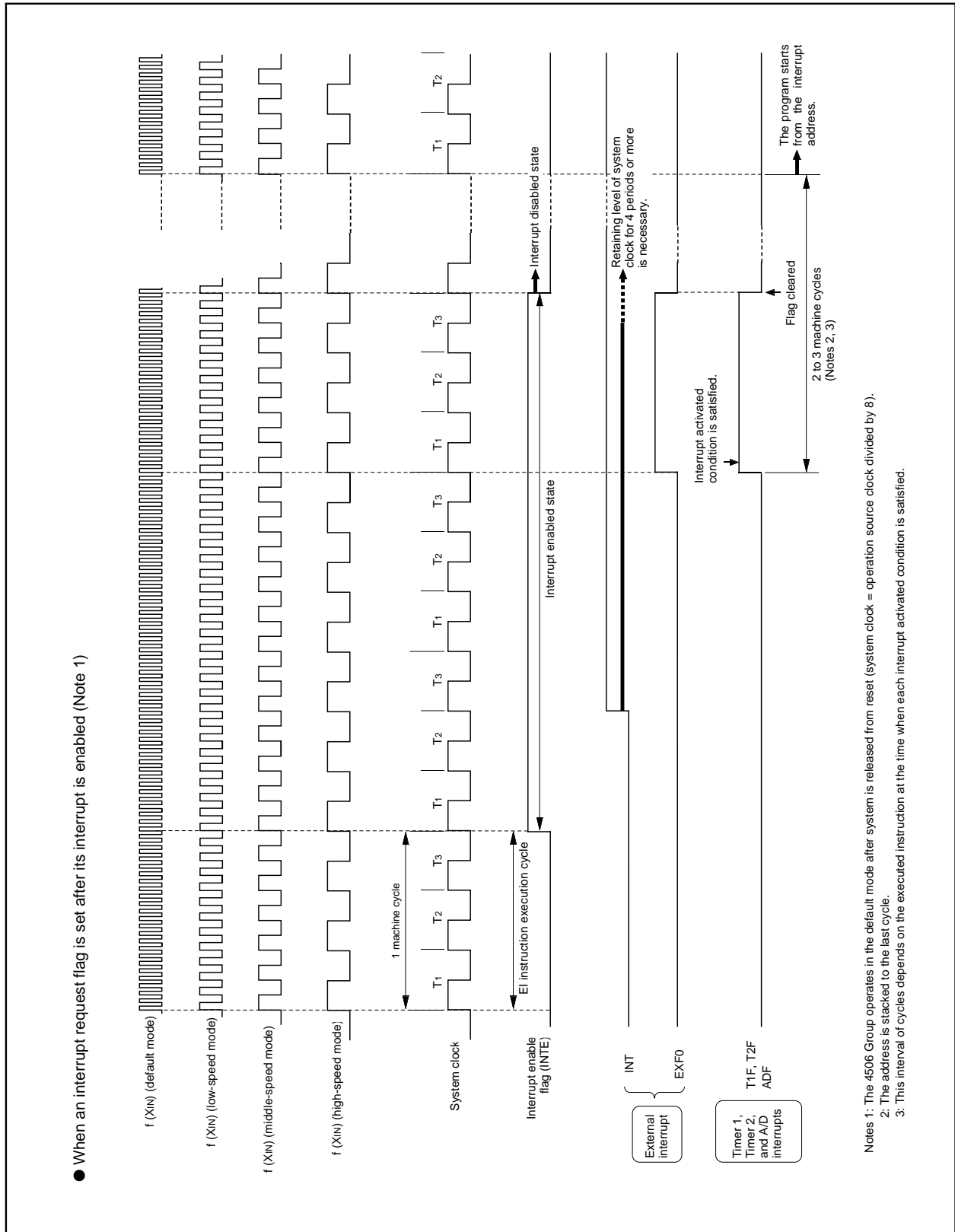


Fig. 16 Interrupt sequence

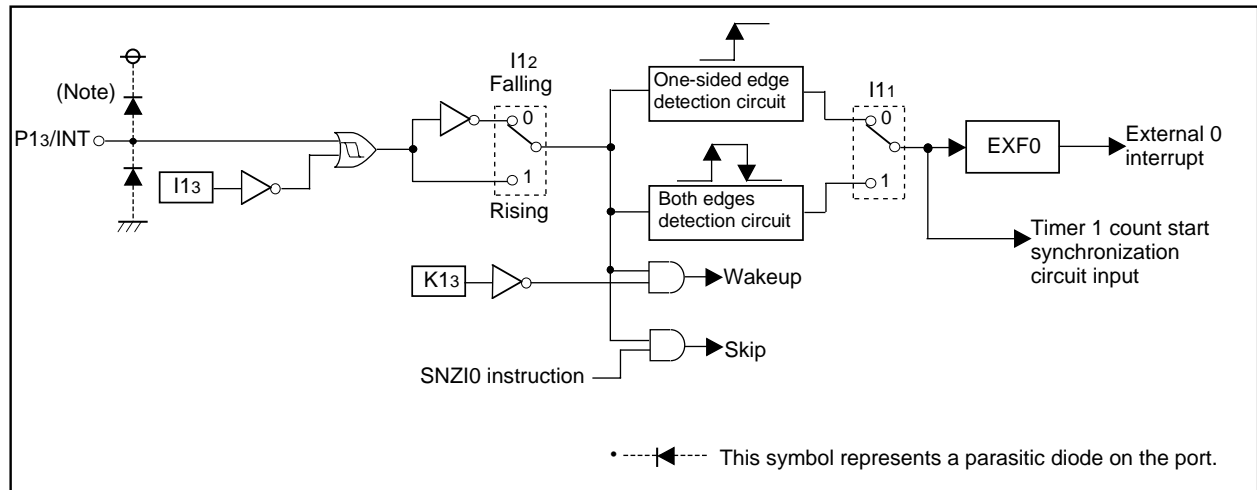
## EXTERNAL INTERRUPTS

The 4506 Group has the external 0 interrupt. An external interrupt request occurs when a valid waveform is input to an interrupt input pin (edge detection).

The external interrupt can be controlled with the interrupt control register I1.

**Table 7 External interrupt activated conditions**

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



**Fig. 17 External interrupt circuit structure**

### (1) External 0 interrupt request flag (EXF0)

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

The valid waveforms causing the interrupt must be retained at their level for 4 clock cycles or more of the system clock (Refer to Figure 16).

The state of EXF0 flag can be examined with the skip instruction (SNZ0). Use the interrupt control register V1 to select the interrupt or the skip instruction. The EXF0 flag is cleared to "0" when an interrupt occurs or when the next instruction is skipped with the skip instruction.

- External 0 interrupt activated condition

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

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

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

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

## (2) External interrupt control registers

- Interrupt control register I1

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

**Table 8 External interrupt control register**

Interrupt control register I1		at reset : 0000z	at RAM back-up : state retained	R/W
I13	INT pin input control bit (Note 2)	0	INT pin input disabled	
		1	INT pin input enabled	
I12	Interrupt valid waveform for INT pin/ return level selection bit (Note 2)	0	Falling waveform ("L" level of INT pin is recognized with the SNZIO instruction)/"L" level	
		1	Rising waveform ("H" level of INT pin is recognized with the SNZIO instruction)/"H" level	
I11	INT pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
I10	INT pin timer 1 control enable bit	0	Disabled	
		1	Enabled	

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

2: When the contents of I12 and I13 are changed, the external interrupt request flag EXF0 may be set. Accordingly, clear EXF0 flag with the SNZ0 instruction when the bit 0 (V10) of register V1 to "0". In this time, set the NOP instruction after the SNZ0 instruction, for the case when a skip is performed with the SNZ0 instruction.

### (3) Notes on interrupts

① Note [1] on bit 3 of register I1

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

- Depending on the input state of the P13/INT pin, the external 0 interrupt request flag (EXF0) may be set when the bit 3 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 18①) and then, change the bit 3 of register I1. In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 18②). Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 18③).

```

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

Fig. 18 External 0 interrupt program example-1

② Note [2] on bit 3 of register I1

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

- When the key-on wakeup function of port P13 is not used (register K13 = "0"), clear bits 2 and 3 of register I1 before system enters to the RAM back-up mode. (refer to Figure 19①).

```

    ⋮
    LA    0    ; (00XX2)
    TI1A  ; Input of INT disabled ..... ①
    DI
    EPOF
    POF   ; RAM back-up
    ⋮
    X : these bits are not used here.
    
```

Fig. 19 External 0 interrupt program example-2

③ Note [3] on bit 2 of register I1

When the interrupt valid waveform of the P13/INT pin is changed with the bit 2 of register I1 in software, be careful about the following notes.

- Depending on the input state of the P13/INT pin, the external 0 interrupt request flag (EXF0) may be set when the bit 2 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 20①) and then, change the bit 2 of register I1 is changed. In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 20②). Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 20③).

```

    ⋮
    LA    4    ; (XXX02)
    TV1A  ; The SNZ0 instruction is valid ..... ①
    LA    12   ; (X1XX2)
    TI1A  ; Interrupt valid waveform is changed
    NOP   ..... ②
    SNZ0  ; The SNZ0 instruction is executed
          ; (EXF0 flag cleared)
    NOP   ..... ③
    ⋮
    X : these bits are not used here.
    
```

Fig. 20 External 0 interrupt program example-3

## TIMERS

The 4506 Group has the following timers.

- Programmable timer

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

- Fixed dividing frequency timer

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

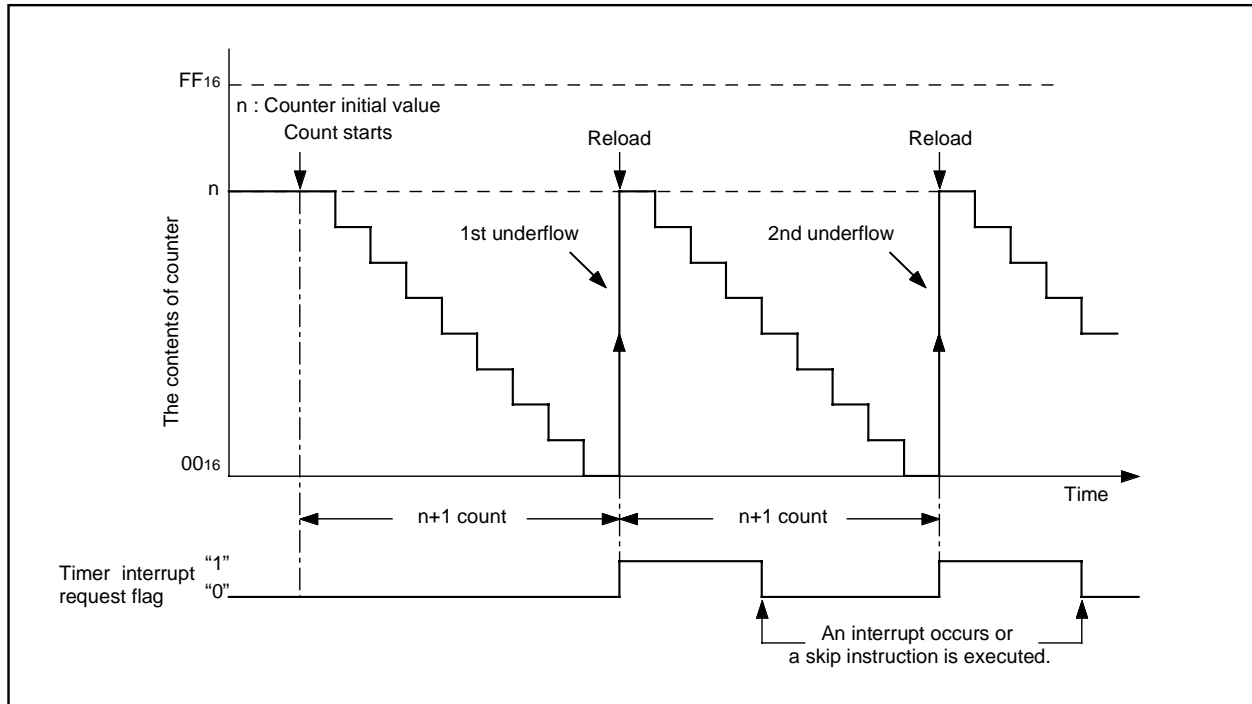


Fig. 21 Auto-reload function

The 4506 Group timer consists of the following circuits.

- Prescaler : frequency divider
- Timer 1 : 8-bit programmable timer
- Timer 2 : 8-bit programmable timer  
(Timers 1 and 2 have the interrupt function, respectively)
- 16-bit timer

Prescaler and timers 1 and 2 can be controlled with the timer control registers W1, W2 and W6. The 16-bit timer is a free counter which is not controlled with the control register.

Each function is described below.

Table 9 Function related timers

Circuit	Structure	Count source	Frequency dividing ratio	Use of output signal	Control register
Prescaler	Frequency divider	• Instruction clock	4, 16	• Timer 1 and 2 count sources	W1
Timer 1	8-bit programmable binary down counter (link to INT input)	• Prescaler output (ORCLK)	1 to 256	• Timer 2 count source • CNTR output • Timer 1 interrupt	W1 W2 W6
Timer 2	8-bit programmable binary down counter	• Timer 1 underflow • Prescaler output (ORCLK) • CNTR input • System clock	1 to 256	• CNTR output • Timer 2 interrupt	W2 W6
16-bit timer	16-bit fixed dividing frequency binary down counter	• Instruction clock	65536	• Watchdog timer (The 16th bit is counted twice)	



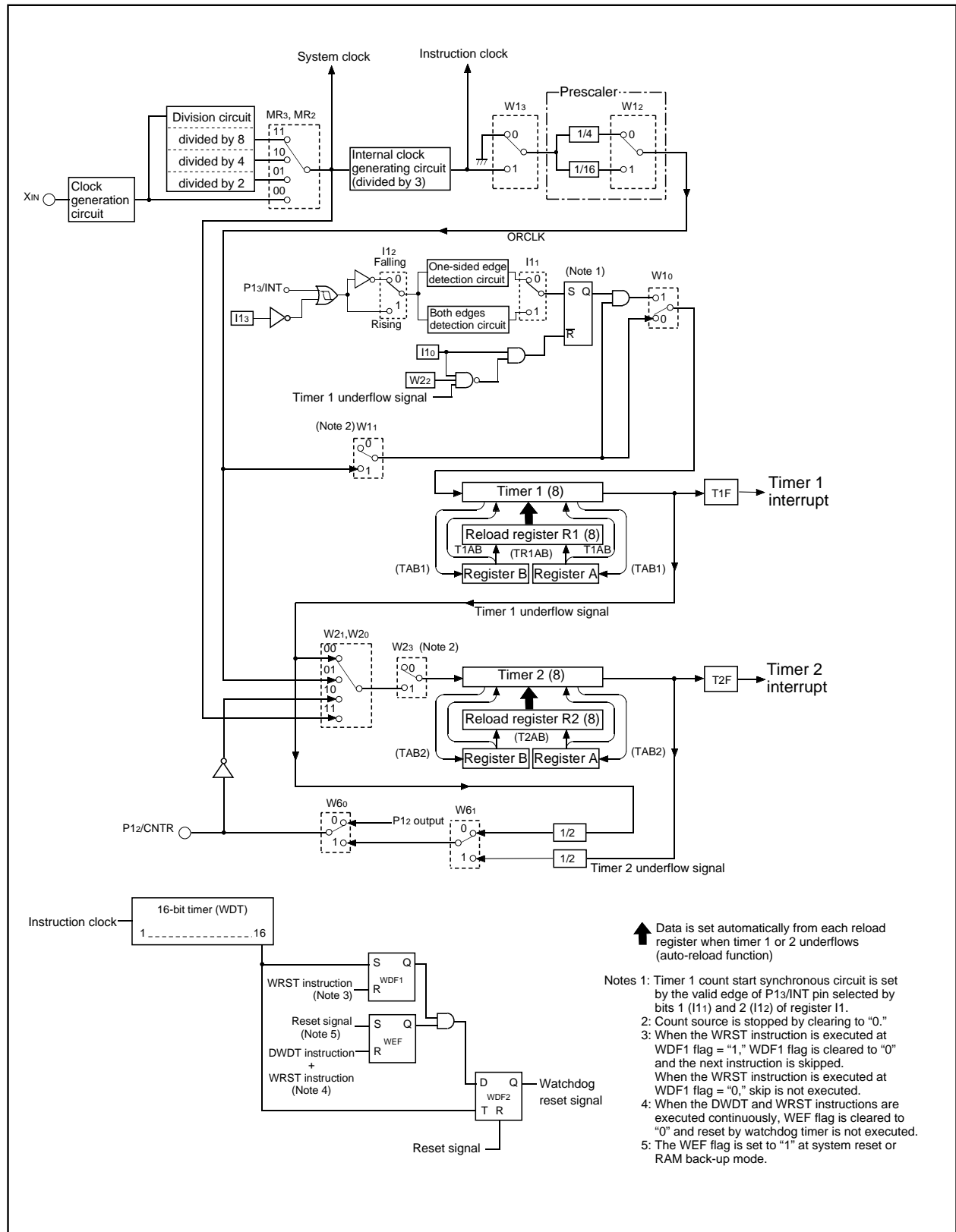


Fig. 22 Timers structure

**Table 10 Timer control registers**

Timer control register W1		at reset : 0000 <sub>2</sub>		at RAM back-up : 0000 <sub>2</sub>	R/W
W13	Prescaler control bit	0	Stop (state initialized)		
		1	Operating		
W12	Prescaler dividing ratio selection bit	0	Instruction clock divided by 4		
		1	Instruction clock divided by 16		
W11	Timer 1 control bit	0	Stop (state retained)		
		1	Operating		
W10	Timer 1 count start synchronous circuit control bit	0	Count start synchronous circuit not selected		
		1	Count start synchronous circuit selected		

Timer control register W2		at reset : 0000 <sub>2</sub>		at RAM back-up : state retained	R/W
W23	Timer 2 control bit	0	Stop (state retained)		
		1	Operating		
W22	Timer 1 count auto-stop circuit selection bit (Note 2)	0	Count auto-stop circuit not selected		
		1	Count auto-stop circuit selected		
W21	Timer 2 count source selection bits	W21 W20		Count source	
		0	0	Timer 1 underflow signal	
0		1	Prescaler output (ORCLK)		
1		0	CNTR input		
W20		1	1	System clock	

Timer control register W6		at reset : 0000 <sub>2</sub>		at RAM back-up : state retained	R/W
W63	Not used	0	This bit has no function, but read/write is enabled.		
		1			
W62	Not used	0	This bit has no function, but read/write is enabled.		
		1			
W61	CNTR output selection bit	0	Timer 1 underflow signal divided by 2 output		
		1	Timer 2 underflow signal divided by 2 output		
W60	P12/CNTR function selection bit	0	P12(I/O)/CNTR input (Note 3)		
		1	P12 (input)/CNTR input/output (Note 3)		

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

2: This function is valid only when the timer 1 count start synchronization circuit is selected.

3: CNTR input is valid only when CNTR input is selected as the timer 2 count source.

## (1) Timer control registers

### • Timer control register W1

Register W1 controls the count operation of timer 1, the selection of count start synchronous circuit, and the frequency dividing ratio and count operation of prescaler. Set the contents of this register through register A with the TW1A instruction. The TAW1 instruction can be used to transfer the contents of register W1 to register A.

### • Timer control register W2

Register W2 controls the selection of timer 1 count auto-stop circuit, and the count operation and count source of timer 2. Set the contents of this register through register A with the TW2A instruction. The TAW2 instruction can be used to transfer the contents of register W2 to register A.

### • Timer control register W6

Register W6 controls the P12/CNTR pin function and the selection of CNTR output. Set the contents of this register through register A with the TW6A instruction. The TAW6 instruction can be used to transfer the contents of register W6 to register A.

## (2) 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 2 of register W1 to select the prescaler dividing ratio and the bit 3 to start and stop its operation. Prescaler is initialized, and the output signal (ORCLK) stops when the bit 3 of register W1 is cleared to "0."

### (3) Timer 1 (interrupt function)

Timer 1 is an 8-bit binary down counter with the timer 1 reload register (R1). Data can be set simultaneously in timer 1 and the reload register (R1) with the T1AB instruction. Stop counting and then execute the T1AB instruction to set data to timer 1. Data can be written to reload register (R1) with the TR1AB instruction.

When writing data to reload register R1 with the TR1AB instruction, the downcount after the underflow is started from the setting value of reload register R1.

Timer 1 starts counting after the following process;

- ① set data in timer 1, and
- ② set the bit 1 of register W1 to "1."

However, INT pin input can be used as the start trigger for timer 1 count operation by setting the bit 0 of register W1 to "1."

Also, in this time, the auto-stop function by timer 1 underflow can be performed by setting the bit 2 of register W2 to "1."

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

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

Data can be read from timer 1 with the TAB1 instruction. When reading the data, stop the counter and then execute the TAB1 instruction.

### (4) Timer 2 (interrupt function)

Timer 2 is an 8-bit binary down counter with the timer 2 reload register (R2). Data can be set simultaneously in timer 2 and the reload register (R2) with the T2AB instruction. Stop counting and then execute the T2AB instruction to set data to timer 2.

Timer 2 starts counting after the following process;

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

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

Once count is started, when timer 2 underflows (the next count pulse is input after the contents of timer 2 becomes "0"), the timer 2 interrupt request flag (T2F) is set to "1," new data is loaded from reload register R2, and count continues (auto-reload function).

Data can be read from timer 2 with the TAB2 instruction. When reading the data, stop the counter and then execute the TAB2 instruction.

### (5) Timer interrupt request flags (T1F, T2F)

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

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

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

### (6) Count start synchronization circuit (timer 1)

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

Timer 1 count start synchronous circuit function is selected by setting the bit 0 of register W1 to "1." The control by INT pin input can be performed by setting the bit 0 of register I1 to "1."

The count start synchronous circuit is set by level change ("H"→"L" or "L"→"H") of INT pin input. This valid waveform is selected by bits 1 (I11) and 2 (I12) of register I1 as follows;

- I11 = "0": Synchronized with one-sided edge (falling or rising)
- I11 = "1": Synchronized with both edges (both falling and rising)

When register I11="0" (synchronized with the one-sided edge), the rising or falling waveform can be selected by the bit 2 of register I1;

- I12 = "0": Falling waveform
- I12 = "1": Rising waveform

When timer 1 count start synchronous circuit is used, the count start synchronous circuit is set, the count source is input to each timer by inputting valid waveform to INT pin. Once set, the count start synchronous circuit is cleared by clearing the bit I10 to "0" or reset.

However, when the count auto-stop circuit is selected (register W22 = "1"), the count start synchronous circuit is cleared (auto-stop) at the timer 1 underflow.

### (7) Count auto-stop circuit (timer 1)

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

The count auto-stop circuit is valid by setting the bit 2 of register W2 to "1". It is cleared by the timer 1 underflow and the count source to timer 1 is stopped.

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

**(8) Timer input/output pin (P12/CNTR pin)**

CNTR pin is used to input the timer 2 count source and output the timer 1 and timer 2 underflow signal divided by 2.

The P12/CNTR pin function can be selected by bit 0 of register W6. The CNTR output signal can be selected by bit 1 of register W6.

When the CNTR input is selected for timer 2 count source, timer 2 counts the falling waveform of CNTR input.

**(9) Precautions**

Note the following for the use of timers.

- Prescaler

Stop the prescaler operation to change its frequency dividing ratio.

- Count source

Stop timer 1 or 2 counting to change its count source.

- Reading the count value

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

- Writing to the timer

Stop timer 1 or 2 counting and then execute the T1AB or T2AB instruction to write its data.

- Writing to reload register R1

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

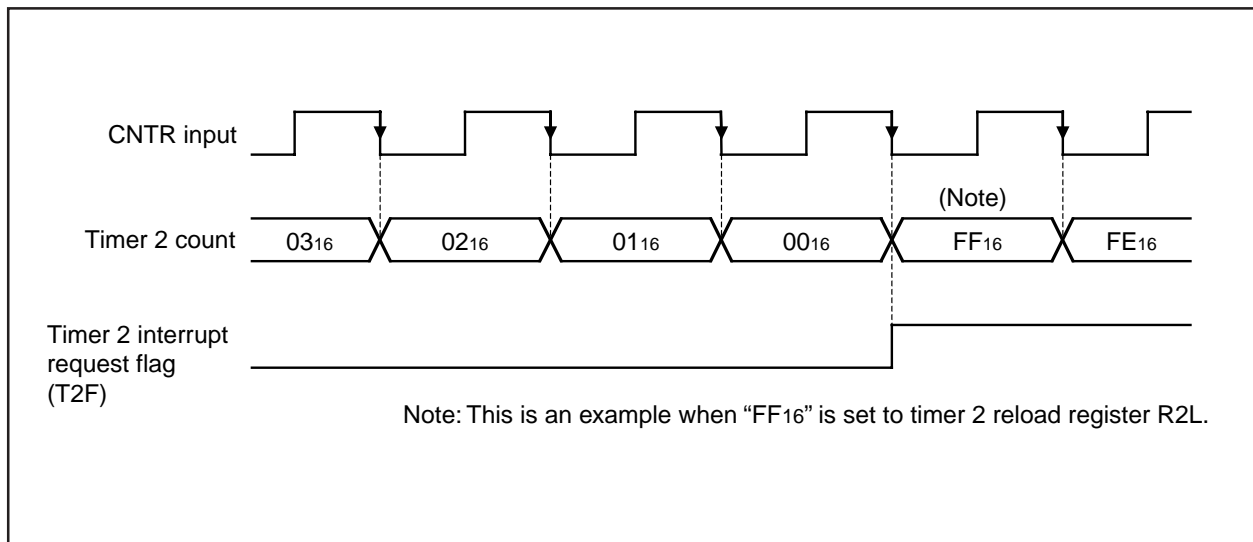


Fig. 23 Count timing diagram at CNTR input

- Timer 1 and timer 2 count start timing and count time when operation starts

Count starts from the first rising edge of the count source (2) after timer 1 and timer 2 operations start (1).

Time to first underflow (3) is shorter (for up to 1 period of the count source) than time among next underflow (4) by the timing to start the timer and count source operations after count starts.

When selecting CNTR input as the count source of timer 2, timer 2 operates synchronizing with the falling edge of CNTR input.

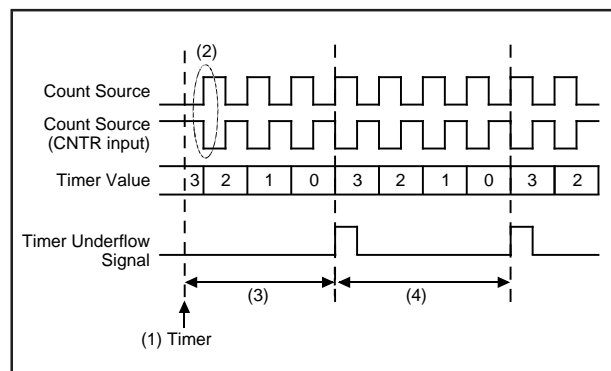


Fig. 24 Timer count start timing and count time when operation starts (T1, T2)

## WATCHDOG TIMER

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

The timer WDT downcounts the instruction clocks as the count source from "FFFF<sub>16</sub>" after system is released from reset.

After the count is started, when the timer WDT underflow occurs (after the count value of timer WDT reaches "FFFF<sub>16</sub>," the next count pulse is input), the WDF1 flag is set to "1."

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

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

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

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

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

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

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

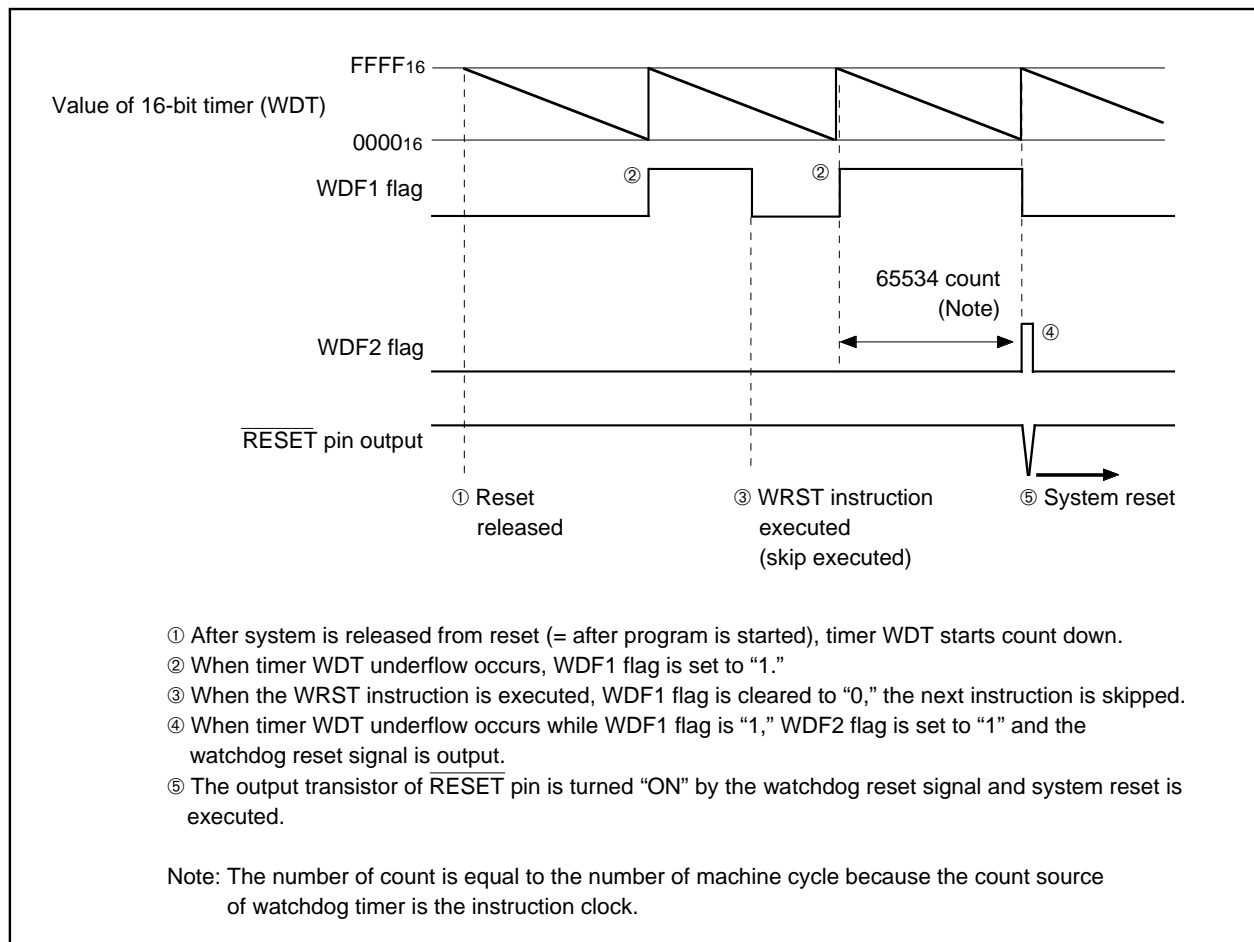


Fig. 25 Watchdog timer function

When the watchdog timer is used, clear the WDF1 flag at the period of 65534 machine cycles or less with the WRST instruction. When the watchdog timer is not used, execute the DWDT instruction and the WRST instruction continuously (refer to Figure 26). The watchdog timer is not stopped with only the DWDT instruction. The contents of WDF1 flag and timer WDT are initialized at the RAM back-up mode.

When using the watchdog timer and the RAM back-up mode, initialize the WDF1 flag with the WRST instruction just before the microcomputer enters the RAM back-up state (refer to Figure 27). The watchdog timer function is valid after system is returned from the RAM back-up. When not using the watchdog timer function, execute the DWDT instruction and the WRST instruction continuously every system is returned from the RAM back-up, and stop the watchdog timer function.

```

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

```

**Fig. 26 Program example to start/stop watchdog timer**

```

:
WRST     ; WDF1 flag cleared
NOP
DI       ; Interrupt disabled
EPOF    ; POF instruction enabled
POF2
↓
Oscillation stop (RAM back-up mode)
:

```

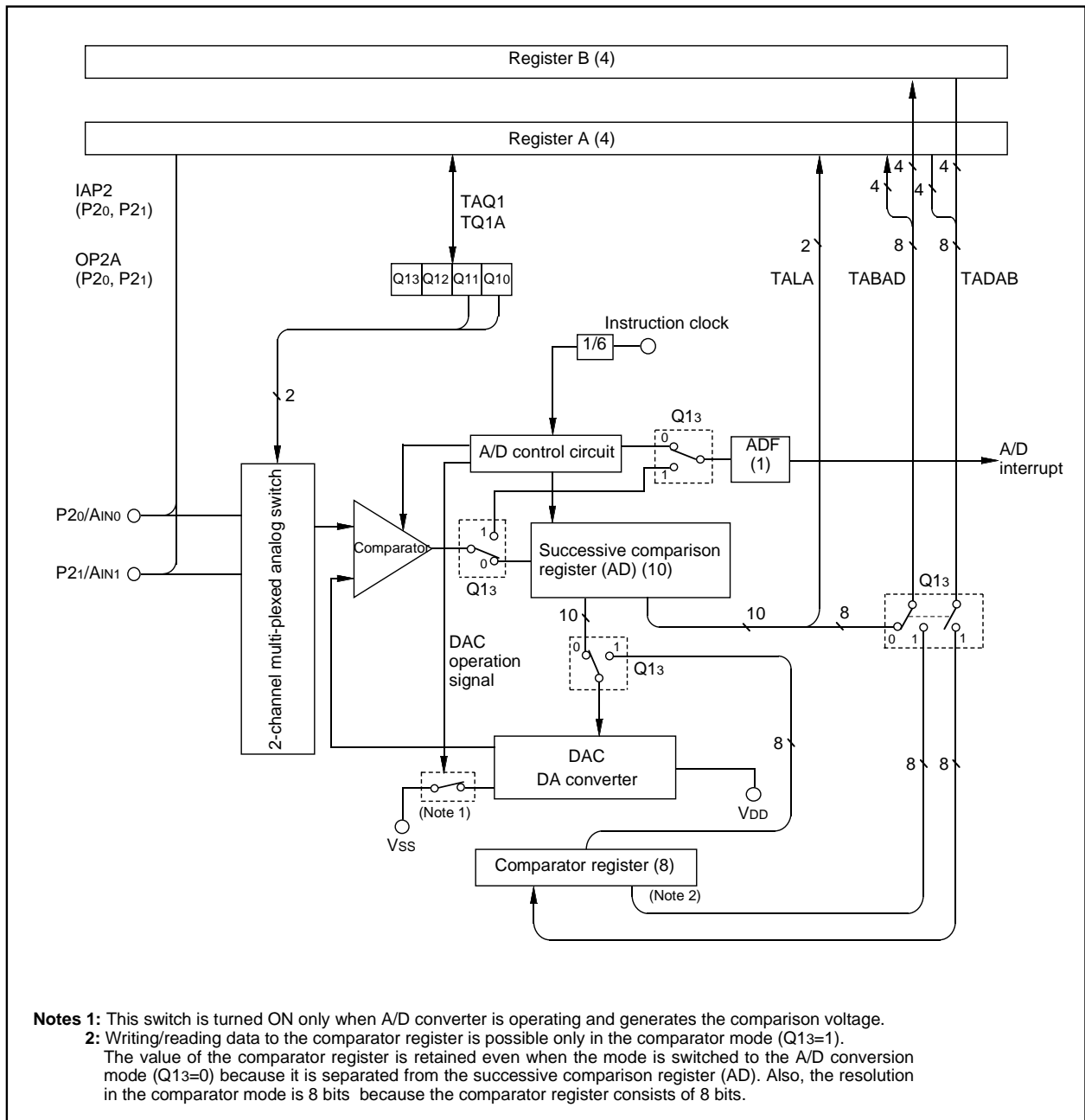
**Fig. 27 Program example to enter the RAM back-up mode when using the watchdog timer**

### A/D CONVERTER

The 4506 Group has a built-in A/D conversion circuit that performs conversion by 10-bit successive comparison method. Table 11 shows the characteristics of this A/D converter. This A/D converter can also be used as an 8-bit comparator to compare analog voltages input from the analog input pin with preset values.

**Table 11 A/D converter characteristics**

Parameter	Characteristics
Conversion format	Successive comparison method
Resolution	10 bits
Relative accuracy	Linearity error: $\pm 2\text{LSB}$
	Differential non-linearity error: $\pm 0.9\text{LSB}$
Conversion speed	46.5 $\mu\text{s}$ (High-speed mode at 4.0 MHz oscillation frequency)
Analog input pin	2



**Fig. 28 A/D conversion circuit structure**

**Table 12 A/D control registers**

A/D control register Q1		at reset : 0000z		at RAM back-up : state retained	R/W
Q13	A/D operation mode selection bit	0	A/D conversion mode		
		1	Comparator mode		
Q12	Not used	0	This bit has no function, but read/write is enabled.		
		1			
Q11	Analog input pin selection bits	Q11	Q10	Selected pins	
		0	0	AIN0	
		0	1	AIN1	
		1	0	Not available	
Q10		1	1	Not available	

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

### (1) Operating at A/D conversion mode

The A/D conversion mode is set by setting the bit 3 of register Q1 to "0."

### (2) Successive comparison register AD

Register AD stores the A/D conversion result of an analog input in 10-bit digital data format. The contents of the high-order 8 bits of this register can be stored in register B and register A with the TABAD instruction. The contents of the low-order 2 bits of this register can be stored into the high-order 2 bits of register A with the TALA instruction. However, do not execute these instructions during A/D conversion.

When the contents of register AD is n, the logic value of the comparison voltage  $V_{ref}$  generated from the built-in DA converter can be obtained with the reference voltage  $V_{DD}$  by the following formula:

<p>Logic value of comparison voltage <math>V_{ref}</math></p> $V_{ref} = \frac{V_{DD}}{1024} \times n$ <p>n: The value of register AD (n = 0 to 1023)</p>
---

### (3) A/D conversion completion flag (ADF)

A/D conversion completion flag (ADF) is set to "1" when A/D conversion completes. The state of ADF flag can be examined with the skip instruction (SNZAD). Use the interrupt control register V2 to select the interrupt or the skip instruction.

The ADF flag is cleared to "0" when the interrupt occurs or when the next instruction is skipped with the skip instruction.

### (4) A/D conversion start instruction (ADST)

A/D conversion starts when the ADST instruction is executed. The conversion result is automatically stored in the register AD.

### (5) A/D control register Q1

Register Q1 is used to select the operation mode and one of analog input pins.

### (6) Operation description

A/D conversion is started with the A/D conversion start instruction (ADST). The internal operation during A/D conversion is as follows:

- ① When the A/D conversion starts, the register AD is cleared to "00016."
- ② Next, the topmost bit of the register AD is set to "1," and the comparison voltage  $V_{ref}$  is compared with the analog input voltage  $V_{IN}$ .
- ③ When the comparison result is  $V_{ref} < V_{IN}$ , the topmost bit of the register AD remains set to "1." When the comparison result is  $V_{ref} > V_{IN}$ , it is cleared to "0."

The 4506 Group repeats this operation to the lowermost bit of the register AD to convert an analog value to a digital value. A/D conversion stops after 62 machine cycles (46.5  $\mu$ s when  $f(X_{IN}) = 4.0$  MHz in high-speed mode) from the start, and the conversion result is stored in the register AD. An A/D interrupt activated condition is satisfied and the ADF flag is set to "1" as soon as A/D conversion completes (Figure 29).



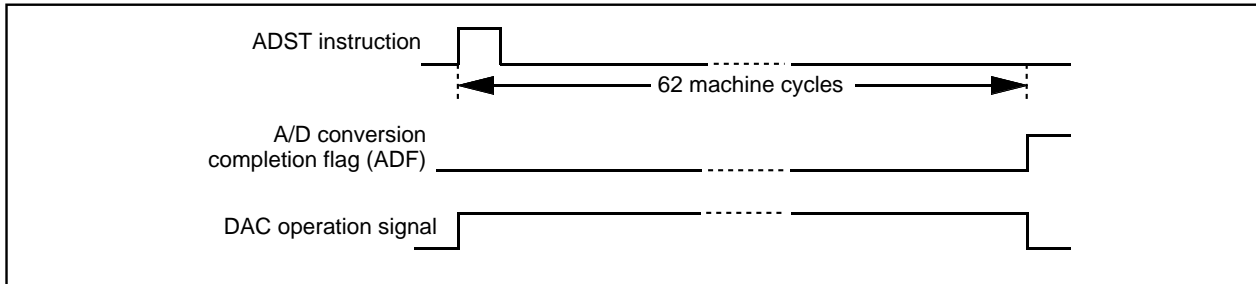
**Table 13 Change of successive comparison register AD during A/D conversion**

At starting conversion	Change of successive comparison register AD	Comparison voltage (Vref) value
1st comparison	1 0 0 ----- 0 0 0	$\frac{V_{DD}}{2}$
2nd comparison	*1 1 0 ----- 0 0 0	$\frac{V_{DD}}{2} \pm \frac{V_{DD}}{4}$
3rd comparison	*1 *2 1 ----- 0 0 0	$\frac{V_{DD}}{2} \pm \frac{V_{DD}}{4} \pm \frac{V_{DD}}{8}$
After 10th comparison completes	A/D conversion result *1 *2 *3 ----- *8 *9 *A	$\frac{V_{DD}}{2} \pm \dots \pm \frac{V_{DD}}{1024}$

\*1: 1st comparison result                      \*2: 2nd comparison result  
 \*3: 3rd comparison result                   \*8: 8th comparison result  
 \*9: 9th comparison result                   \*A: 10th comparison result

**(7) A/D conversion timing chart**

Figure 29 shows the A/D conversion timing chart.

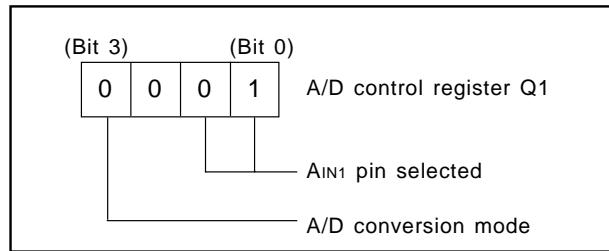


**Fig. 29 A/D conversion timing chart**

**(8) How to use A/D conversion**

How to use A/D conversion is explained using as example in which the analog input from P21/AIN1 pin is A/D converted, and the high-order 4 bits of the converted data are stored in address M(Z, X, Y) = (0, 0, 0), the middle-order 4 bits in address M(Z, X, Y) = (0, 0, 1), and the low-order 2 bits in address M(Z, X, Y) = (0, 0, 2) of RAM. The A/D interrupt is not used in this example.

- ① Select the AIN1 pin function and A/D conversion mode with the register Q1 (refer to Figure 30).
- ② Execute the ADST instruction and start A/D conversion.
- ③ Examine the state of ADF flag with the SNZAD instruction to determine the end of A/D conversion.
- ④ Transfer the low-order 2 bits of converted data to the high-order 2 bits of register A (TALA instruction).
- ⑤ Transfer the contents of register A to M (Z, X, Y) = (0, 0, 2).
- ⑥ Transfer the high-order 8 bits of converted data to registers A and B (TABAD instruction).
- ⑦ Transfer the contents of register A to M (Z, X, Y) = (0, 0, 1).
- ⑧ Transfer the contents of register B to register A, and then, store into M(Z, X, Y) = (0, 0, 0).



**Fig. 30 Setting registers**

### (9) Operation at comparator mode

The A/D converter is set to comparator mode by setting bit 3 of the register Q1 to "1."

Below, the operation at comparator mode is described.

### (10) Comparator register

In comparator mode, the built-in DA comparator is connected to the 8-bit comparator register as a register for setting comparison voltages. The contents of register B is stored in the high-order 4 bits of the comparator register and the contents of register A is stored in the low-order 4 bits of the comparator register with the TADAB instruction.

When changing from A/D conversion mode to comparator mode, the result of A/D conversion (register AD) is undefined.

However, because the comparator register is separated from register AD, the value is retained even when changing from comparator mode to A/D conversion mode. Note that the comparator register can be written and read at only comparator mode.

If the value in the comparator register is  $n$ , the logic value of comparison voltage  $V_{ref}$  generated by the built-in DA converter can be determined from the following formula:

$$V_{ref} = \frac{V_{DD}}{256} \times n$$

$n$ : The value of register AD ( $n = 0$  to  $255$ )

### (11) Comparison result store flag (ADF)

In comparator mode, the ADF flag, which shows completion of A/D conversion, stores the results of comparing the analog input voltage with the comparison voltage. When the analog input voltage is lower than the comparison voltage, the ADF flag is set to "1." The state of ADF flag can be examined with the skip instruction (SNZAD). Use the interrupt control register V2 to select the interrupt or the skip instruction.

The ADF flag is cleared to "0" when the interrupt occurs or when the next instruction is skipped with the skip instruction.

### (12) Comparator operation start instruction (ADST instruction)

In comparator mode, executing ADST starts the comparator operation.

The comparator stops 8 machine cycles after it has started ( $6 \mu\text{s}$  at  $f(XIN) = 4.0 \text{ MHz}$  in high-speed mode). When the analog input voltage is lower than the comparison voltage, the ADF flag is set to "1."

### (13) Notes for the use of A/D conversion 1

Note the following when using the analog input pins also for port P2 function:

- Selection of analog input pins  
Even when P20/AIN0, P21/AIN1 are set to pins for analog input, they continue to function as port P2 input/output. Accordingly, when any of them are used as I/O port and others are used as analog input pins, make sure to set the outputs of pins that are set for analog input to "1." Also, the port input function of the pin functions as an analog input is undefined.
- TALA instruction  
When the TALA instruction is executed, the low-order 2 bits of register AD is transferred to the high-order 2 bits of register A, simultaneously, the low-order 2 bits of register A is "0."

### (14) Notes for the use of A/D conversion 2

Do not change the operating mode (both A/D conversion mode and comparator mode) of A/D converter with the bit 3 of register Q1 while the A/D converter is operating.

When the operating mode of A/D converter is changed from the comparator mode to A/D conversion mode with the bit 3 of register Q1, note the following:

- Clear the bit 2 of register V2 to "0" to change the operating mode of the A/D converter from the comparator mode to A/D conversion mode with the bit 3 of register Q1.
- The A/D conversion completion flag (ADF) may be set when the operating mode of the A/D converter is changed from the comparator mode to the A/D conversion mode. Accordingly, set a value to the bit 3 of register Q1, and execute the SNZAD instruction to clear the ADF flag.

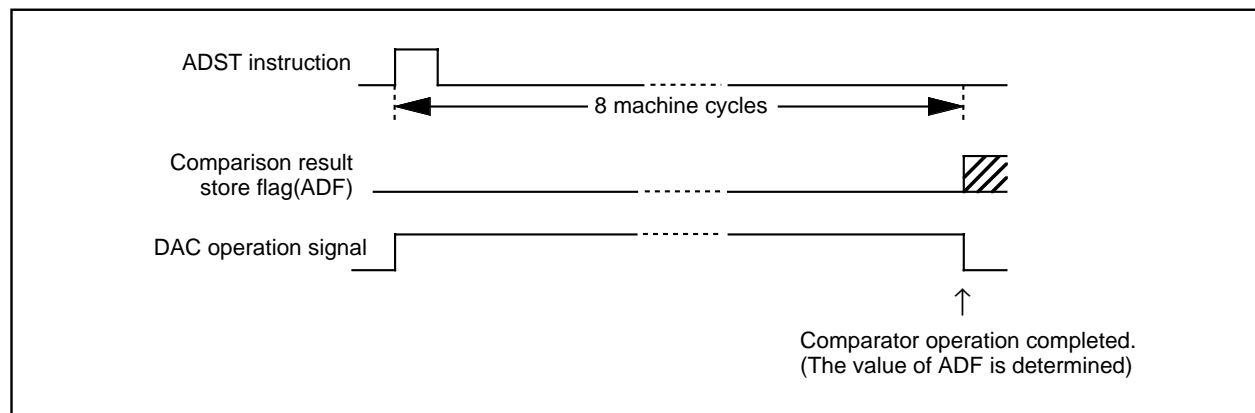


Fig. 31 Comparator operation timing chart

**(15) Definition of A/D converter accuracy**

The A/D conversion accuracy is defined below (refer to Figure 32).

$V_n$ : Analog input voltage when the output data changes from "n" to "n+1" (n = 0 to 1022)

• Relative accuracy

① Zero transition voltage ( $V_{0T}$ )

This means an analog input voltage when the actual A/D conversion output data changes from "0" to "1."

② Full-scale transition voltage ( $V_{FST}$ )

This means an analog input voltage when the actual A/D conversion output data changes from "1023" to "1022."

③ Linearity error

This means a deviation from the line between  $V_{0T}$  and  $V_{FST}$  of a converted value between  $V_{0T}$  and  $V_{FST}$ .

④ Differential non-linearity error

This means a deviation from the input potential difference required to change a converter value between  $V_{0T}$  and  $V_{FST}$  by 1 LSB at the relative accuracy.

• 1LSB at relative accuracy  $\rightarrow \frac{V_{FST}-V_{0T}}{1022}$  (V)

• 1LSB at absolute accuracy  $\rightarrow \frac{V_{DD}}{1024}$  (V)

• Absolute accuracy

This means a deviation from the ideal characteristics between 0 to  $V_{DD}$  of actual A/D conversion characteristics.

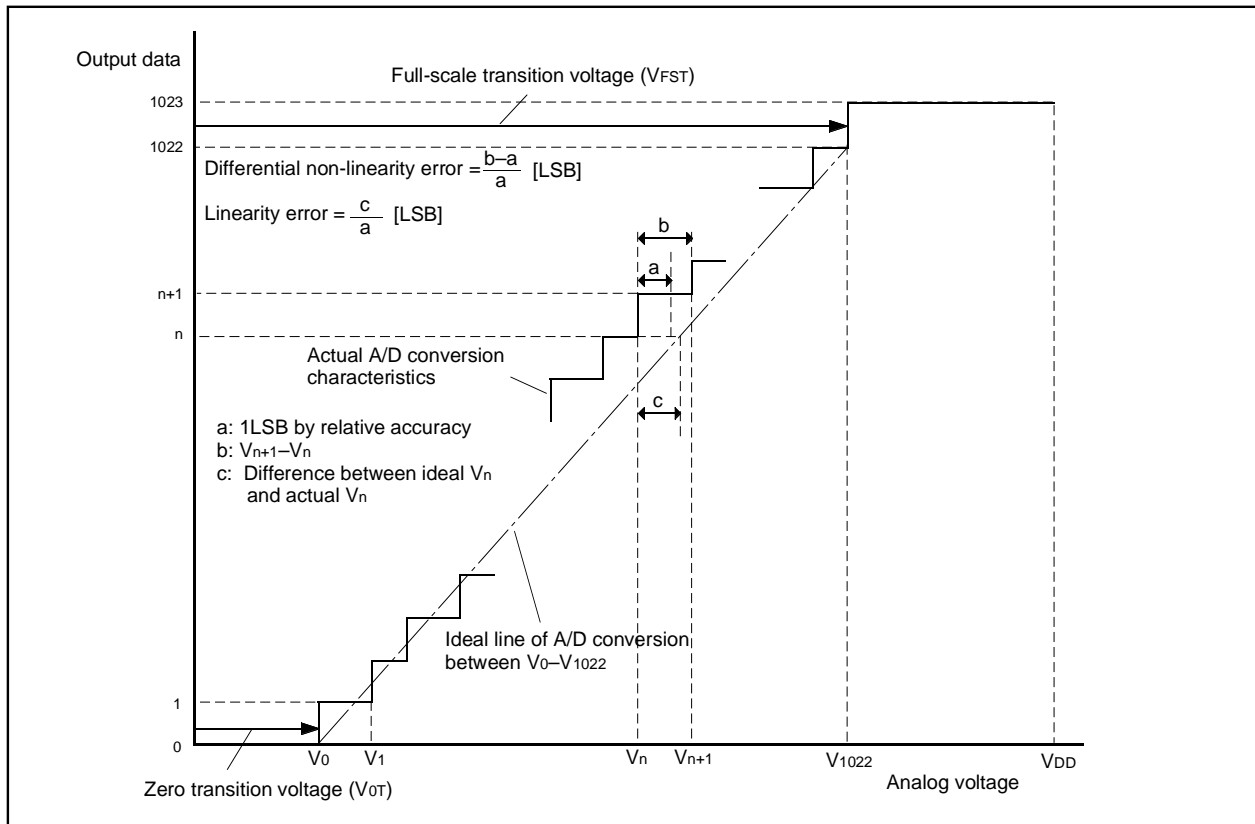


Fig. 32 Definition of A/D conversion accuracy

### 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.

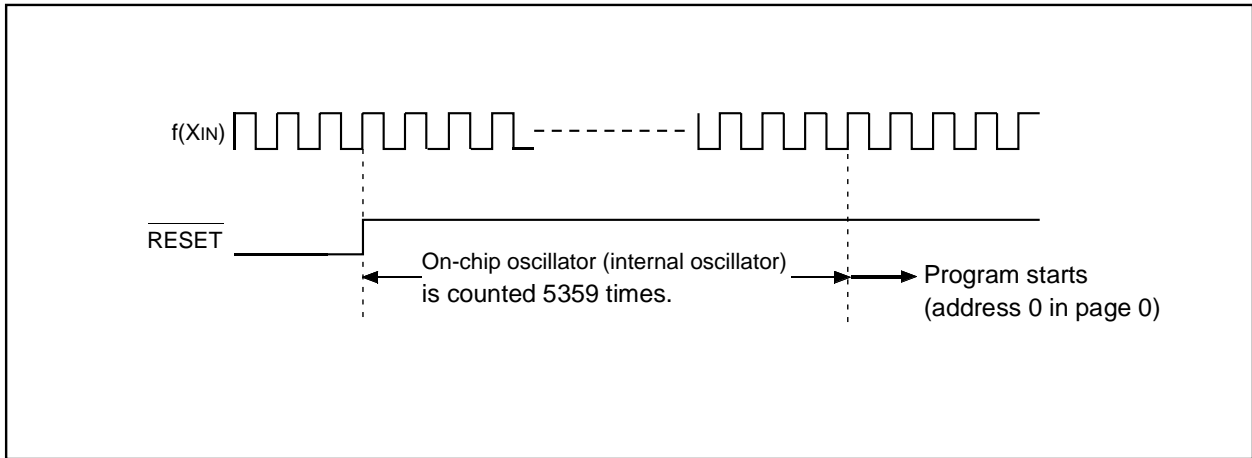


Fig. 33 Reset release timing

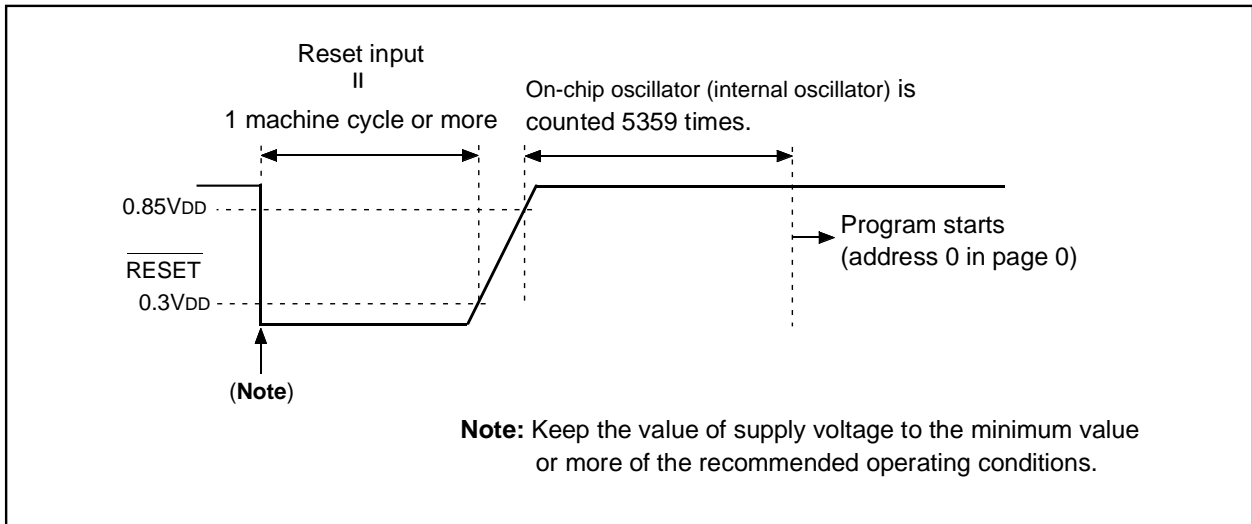
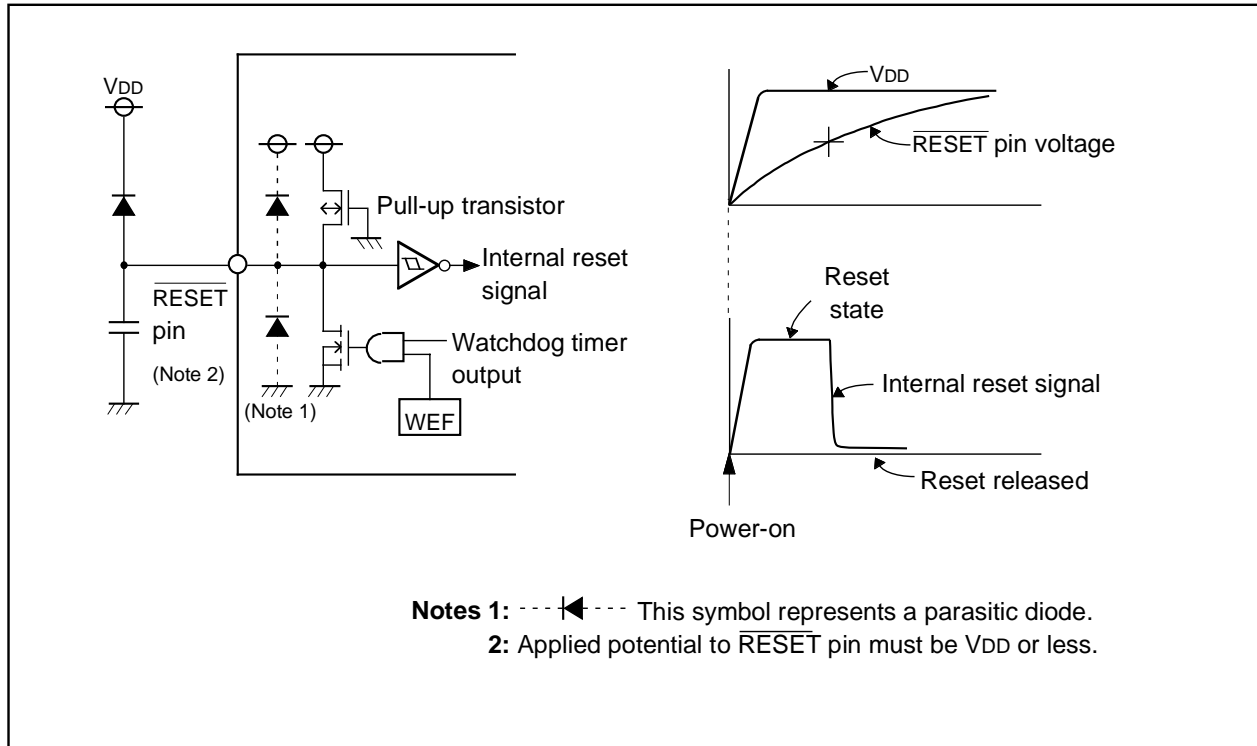


Fig. 34  $\overline{\text{RESET}}$  pin input waveform and reset operation

**(1) Power-on reset**

Reset can be performed automatically at power on (power-on reset) by connecting a diode and a capacitor to  $\overline{\text{RESET}}$  pin. Connect  $\overline{\text{RESET}}$  pin and the external circuit at the shortest distance.



**Fig. 35 Structure of reset pin and its peripherals, and power-on reset operation**

**Table 14 Port state at reset**

Name	Function	State
D0, D1	D0, D1	High-impedance (Note 1)
D2/C, D3/K	D2, D3	High-impedance (Notes 1, 2)
P00, P01, P02, P03	P00–P03	High-impedance (Notes 1, 2)
P10, P11, P12/CNTR, P13/INT	P10–P13	High-impedance (Notes 1, 2)
P20/AIN0, P21/AIN1	P20, P21	High-impedance (Notes 1, 2)

Notes 1: Output latch is set to "1."

2: Pull-up transistor is turned OFF.

## (2) Internal state at reset

Figure 36 shows internal state at reset (they are the same after system is released from reset). The contents of timers, registers, flags and RAM except shown in Figure 36 are undefined, so set the initial value to them.

• Program counter (PC) .....	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
Address 0 in page 0 is set to program counter.		
• Interrupt enable flag (INTE) .....	0	(Interrupt disabled)
• Power down flag (P) .....	0	
• External 0 interrupt request flag (EXF0) .....	0	
• Interrupt control register V1 .....	0 0 0 0	(Interrupt disabled)
• Interrupt control register V2 .....	0 0 0 0	(Interrupt disabled)
• Interrupt control register I1 .....	0 0 0 0	
• Timer 1 interrupt request flag (T1F) .....	0	
• Timer 2 interrupt request flag (T2F) .....	0	
• Watchdog timer flags (WDF1, WDF2) .....	0	
• Watchdog timer enable flag (WEF) .....	1	
• Timer control register W1 .....	0 0 0 0	(Prescaler and timer 1 stopped)
• Timer control register W2 .....	0 0 0 0	(Timer 2 stopped)
• Timer control register W6 .....	0 0 0 0	
• Clock control register MR .....	1 1 0 0	
• Key-on wakeup control register K0 .....	0 0 0 0	
• Key-on wakeup control register K1 .....	0 0 0 0	
• Key-on wakeup control register K2 .....	0 0 0 0	
• Pull-up control register PU0 .....	0 0 0 0	
• Pull-up control register PU1 .....	0 0 0 0	
• Pull-up control register PU2 .....	0 0 0 0	
• A/D conversion completion flag (ADF) .....	0	
• A/D control register Q1 .....	0 0 0 0	
• Carry flag (CY) .....	0	
• Register A .....	0 0 0 0	
• Register B .....	0 0 0 0	
• Register D .....	X X X	
• Register E .....	X X X X X X X X X	
• Register X .....	0 0 0 0	
• Register Y .....	0 0 0 0	
• Register Z .....	X X	
• Stack pointer (SP) .....	1 1 1	
• Oscillation clock .....	On-chip oscillator (operating)	
• Ceramic resonator circuit .....	Operating	
• RC oscillation circuit .....	Stop	

"X" represents undefined.

Fig. 36 Internal state at reset

## RAM BACK-UP MODE

The 4506 Group has the RAM back-up mode.

When the POF2 instruction is executed continuously after the EPOF instruction, system enters the RAM back-up state.

The POF2 instruction is equal to the NOP instruction when the EPOF instruction is not executed before the POF2 instruction.

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 15 shows the function and states retained at RAM back-up. Figure 36 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 EPOF instruction and POF2 instruction continuously, the CPU starts executing the program from address 0 in page 0. In this case, the P flag is "1."

### (3) Cold start condition

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

- reset pulse is input to  $\overline{\text{RESET}}$  pin, or
- reset by watchdog timer is performed, or

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

**Table 15 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 level	(Note 5)
Selected oscillation circuit	O
Timer control register W1	X
Timer control registers W2, W6	O
Clock control register MR	X
Interrupt control registers V1, V2	X
Interrupt control register I1	O
Timer 1 function	X
Timer 2 function	(Note 3)
A/D conversion function	X
A/D control register Q1	O
Pull-up control registers PU0 to PU2	O
Key-on wakeup control registers K0 to K2	O
External 0 interrupt request flag (EXF0)	X
Timer 1 interrupt request flag (T1F)	X
Timer 2 interrupt request flag (T2F)	(Note 3)
Watchdog timer flags (WDF1)	X (Note 4)
Watchdog timer enable flag (WEF)	X
16-bit timer (WDT)	X (Note 4)
A/D conversion completion flag (ADF)	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 "7" at RAM back-up.

3: The state of the timer is undefined.

4: Initialize the watchdog timer flag WDF1 with the WRST instruction, and then execute the POF2 instruction.

5: As for the D2/C pin, the output latch of port C is set to "1" at the RAM back-up. However, the output latch of port D2 is retained.

As for the other ports, their output levels are retained at the RAM back-up.

#### (4) Return signal

An external wakeup signal is used to return from the RAM back-up mode because the oscillation is stopped. Table 16 shows the return condition for each return source.

#### (5) Control registers

- Key-on wakeup control register K0

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

- Key-on wakeup control register K1

Register K1 controls the port P1 key-on wakeup function. Set the contents of this register through register A with the TK1A instruction. In addition, the TAK1 instruction can be used to transfer the contents of register K0 to register A.

- Key-on wakeup control register K2

Register K2 controls the ports P2, D2/C and D3/K key-on wakeup function. Set the contents of this register through register A with the TK2A instruction. In addition, the TAK2 instruction can be used to transfer the contents of register K2 to register A.

- Pull-up control register PU0

Register PU0 controls the ON/OFF of the port P0 pull-up transistor. Set the contents of this register through register A with the TPU0A instruction.

- Pull-up control register PU1

Register PU1 controls the ON/OFF of the port P1 pull-up transistor. Set the contents of this register through register A with the TPU1A instruction.

- Pull-up control register PU2

Register PU2 controls the ON/OFF of the ports P2, D2/C and D3/K pull-up transistor. Set the contents of this register through register A with the TPU2A instruction.

- Interrupt control register I1

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

**Table 16 Return source and return condition**

	Return source	Return condition	Remarks
External wakeup signal	Port P0 Port P1 (Note) Port P2 Ports D2/C, D3/K	Return by an external "L" level input.	The key-on wakeup function can be selected by one port unit. Set the port using the key-on wakeup function to "H" level before going into the RAM back-up state.
	Port P13/INT (Note)	Return by an external "H" level or "L" level input. The return level can be selected with the bit 2 (I12) of register I1. When the return level is input, the EXF0 flag is not set.	Select the return level ("L" level or "H" level) with the bit 2 of register I1 according to the external state before going into the RAM back-up state.

Note: When the bit 3 (K13) of register K1 is "0", the key-on wakeup of the INT pin is valid ("H" or "L" level). It is "1", the key-on wakeup of port P13 is valid ("L" level).



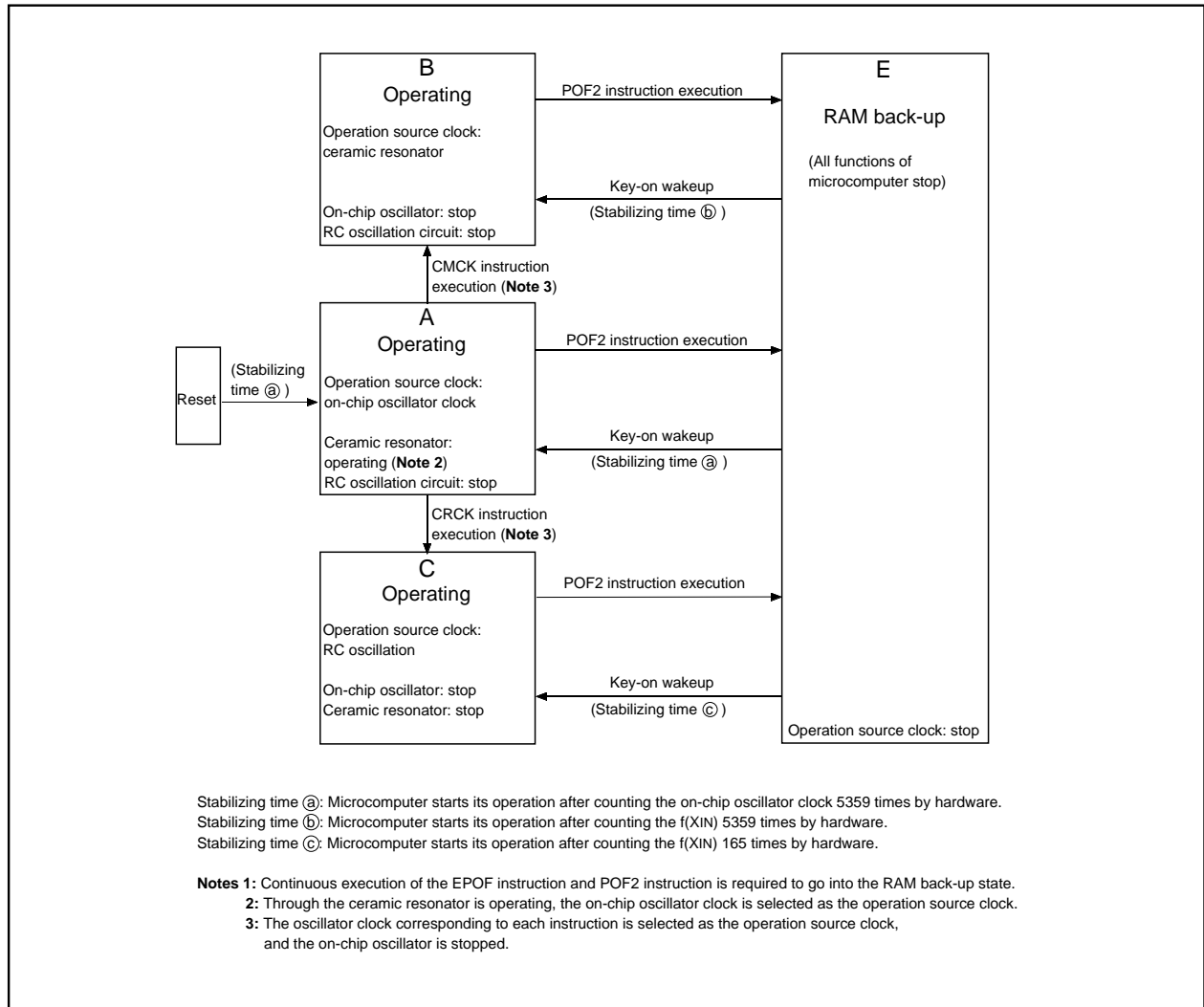


Fig. 37 State transition

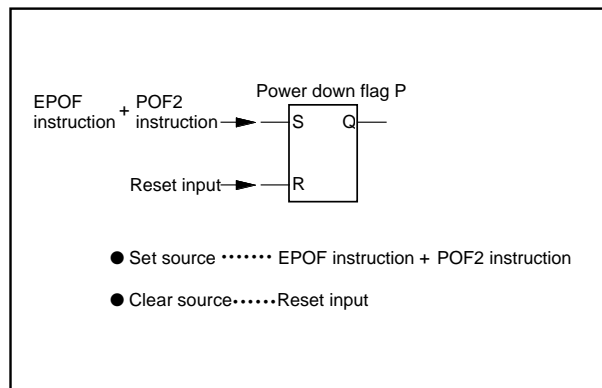


Fig. 38 Set source and clear source of the P flag

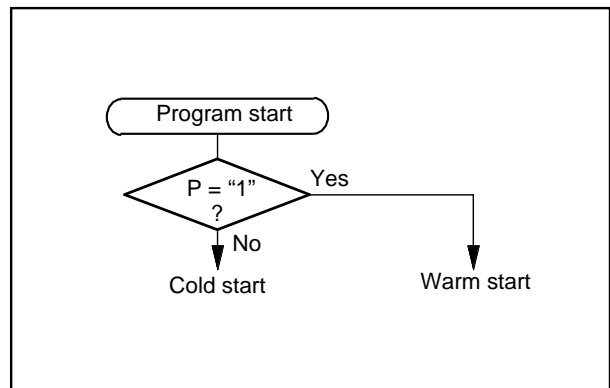


Fig. 39 Start condition identified example using the SNZP instruction

**Table 17 Key-on wakeup control register**

Key-on wakeup control register K0		at reset : 00002		at RAM back-up : state retained	R/W
K03	Port P03 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K02	Port P02 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K01	Port P01 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K00	Port P00 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K1		at reset : 00002		at RAM back-up : state retained	R/W
K13	Port P13/INT key-on wakeup control bit	0	P13 key-on wakeup not used/INT pin key-on wakeup used		
		1	P13 key-on wakeup used/INT pin key-on wakeup not used		
K12	Port P12/CNTR key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K11	Port P11 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K10	Port P10 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K2		at reset : 00002		at RAM back-up : state retained	R/W
K23	Port D3/K key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K22	Port D2/C key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K21	Port P21/AIN1 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K20	Port P20/AIN0 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

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

**Table 18 Pull-up control register and interrupt control register**

Pull-up control register PU0		at reset : 0000z		at RAM back-up : state retained	W
PU03	Port P03 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU02	Port P02 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU01	Port P01 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU00	Port P00 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		

Pull-up control register PU1		at reset : 0000z		at RAM back-up : state retained	W
PU13	Port P13/INT pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU12	Port P12/CNTR pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU11	Port P11 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU10	Port P10 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		

Pull-up control register PU2		at reset : 0000z		at RAM back-up : state retained	W
PU23	Port D3/K pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU22	Port D2/C pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU21	Port P21/AIN1 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU20	Port P20/AIN0 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		

Interrupt control register I1		at reset : 0000z		at RAM back-up : state retained	R/W
I13	INT pin input control bit (Note 2)	0	INT pin input disabled		
		1	INT pin input enabled		
I12	Interrupt valid waveform for INT pin/ return level selection bit (Note 2)	0	Falling waveform ("L" level of INT pin is recognized with the SNZIO instruction)/"L" level		
		1	Rising waveform ("H" level of INT pin is recognized with the SNZIO instruction)/"H" level		
I11	INT pin edge detection circuit control bit	0	One-sided edge detected		
		1	Both edges detected		
I10	INT pin timer 1 control enable bit	0	Disabled		
		1	Enabled		

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

2: When the contents of I12 and I13 are changed, the external interrupt request flag EXF0 may be set. Accordingly, clear EXF0 flag with the SNZ0 instruction when the bit 0 (V10) of register V1 to "0". In this time, set the NOP instruction after the SNZ0 instruction, for the case when a skip is performed with the SNZ0 instruction.

## CLOCK CONTROL

The clock control circuit consists of the following circuits.

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

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

Figure 40 shows the structure of the clock control circuit.

The 4506 Group operates by the on-chip oscillator clock ( $f(\text{RING})$ ) which is the internal oscillator after system is released from reset. Also, the ceramic resonator or the RC oscillation can be used for the source oscillation ( $f(\text{XIN})$ ) of the 4506 Group. The CMCK instruction or CRCK instruction is executed to select the ceramic resonator or RC oscillator, respectively.

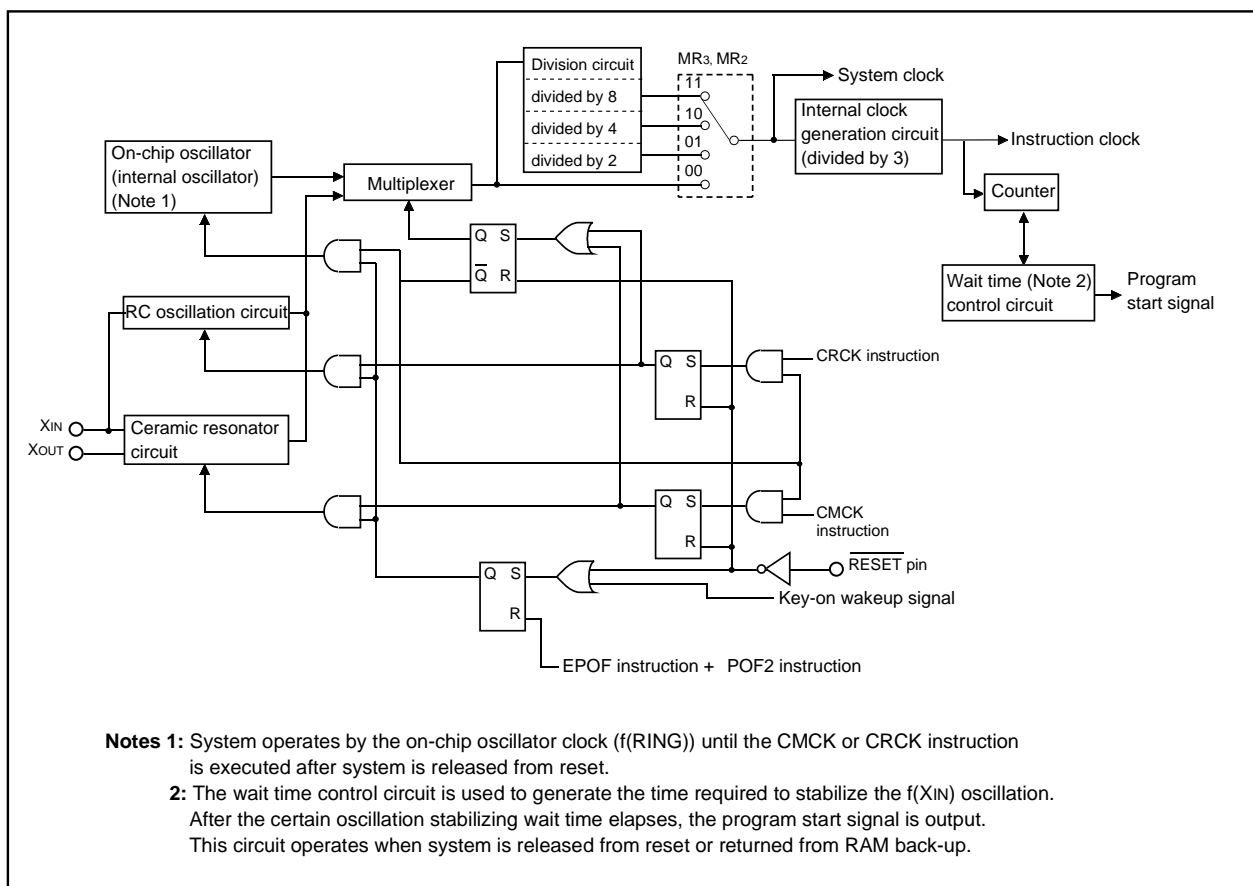


Fig. 40 Clock control circuit structure

### (1) Selection of source oscillation (f(XIN))

The ceramic resonator or RC oscillation can be used for the source oscillation of the MCU.

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

When the ceramic resonator is used, execute the CMCK instruction. When the RC oscillation is used, execute the CRCK instruction. The oscillation circuit by the CMCK or CRCK instruction can be selected only at once. The oscillation circuit corresponding to the first executed one of these two instructions is valid. Other oscillation circuit and the on-chip oscillator stop.

Execute the CMCK or the CRCK instruction in the initial setting routine of program (executing it in address 0 in page 0 is recommended). Also, when the CMCK or the CRCK instruction is not executed in program, the MCU operates by the on-chip oscillator.

### (2) On-chip oscillator operation

When the MCU operates by the on-chip oscillator as the source oscillation (f(XIN)) without using the ceramic resonator or the RC oscillator, connect XIN pin to VSS and leave XOUT pin open (Figure 42).

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

Be careful that variable frequencies when designing application products.

### (3) Ceramic resonator

When the ceramic resonator is used as the source oscillation (f(XIN)), connect the ceramic resonator and the external circuit to pins XIN and XOUT at the shortest distance. Then, execute the CMCK instruction. A feedback resistor is built in between pins XIN and XOUT (Figure 43).

### (4) RC oscillation

When the RC oscillation is used as the source oscillation (f(XIN)), connect the XIN pin to the external circuit of resistor R and the capacitor C at the shortest distance and leave XOUT pin open. Then, execute the CRCK instruction (Figure 44).

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

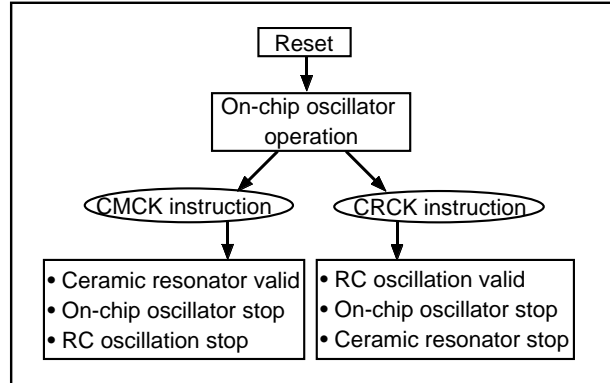


Fig. 41 Switch to ceramic resonance/RC oscillation

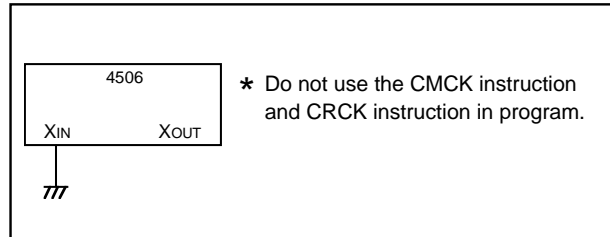


Fig. 42 Handling of XIN and XOUT when operating on-chip oscillator

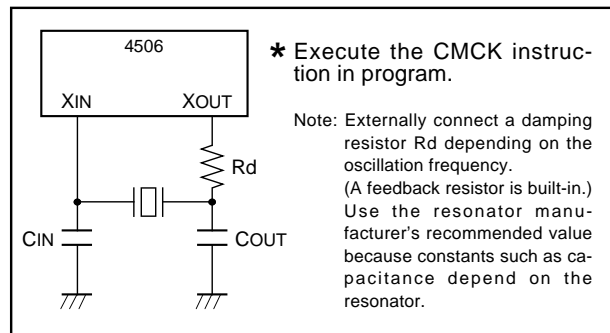


Fig. 43 Ceramic resonator external circuit

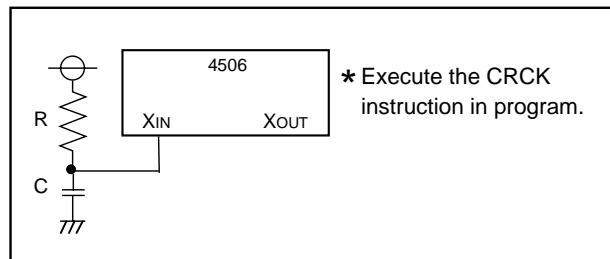


Fig. 44 External RC oscillation circuit

**(5) External clock**

When the external signal clock is used as the source oscillation (f(XIN)), connect the XIN pin to the clock source and leave XOUT pin open. Then, execute the CMCK instruction (Figure 45). Be careful that the maximum value of the oscillation frequency when using the external clock differs from the value when using the ceramic resonator (refer to the recommended operating condition). Also, note that the RAM back-up mode (POF2 instruction) cannot be used when using the external clock.

**(6) Clock control register MR**

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

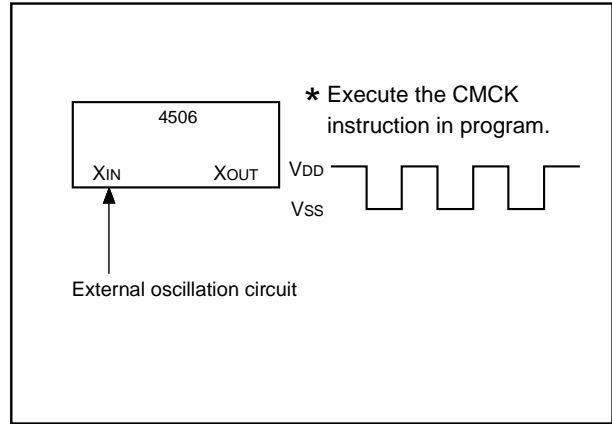


Fig. 45 External clock input circuit

Table 19 Clock control register MR

Clock control register MR		at reset : 11002		at RAM back-up : 11002	R/W
MR3	System clock selection bits	MR3	MR2	System clock	
		0	0	f(XIN) (high-speed mode)	
		0	1	f(XIN)/2 (middle-speed mode)	
MR2		1	0	f(XIN)/4 (low-speed mode)	
		1	1	f(XIN)/8 (default mode)	
MR1	Not used	0	This bit has no function, but read/write is enabled.		
		1			
MR0	Not used	0	This bit has no function, but read/write is enabled.		
		1			

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

**ROM ORDERING METHOD**

Please submit the information described below when ordering Mask ROM.

- (1) 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

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

## LIST OF PRECAUTIONS

### ① Noise and latch-up prevention

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

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

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

### ② Register initial values 1

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

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

### ③ Register initial values 2

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

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

### ④ Stack registers (SKs) and stack pointer (SP)

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

### ⑤ Prescaler

Stop the prescaler operation to change its frequency dividing ratio.

### ⑥ Timer count source

Stop timer 1 or 2 counting to change its count source.

### ⑦ Reading the count value

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

### ⑧ Writing to the timer

Stop timer 1 or 2 counting and then execute the T1AB or T2AB instruction to write its data.

### ⑨ Writing to reload register R1

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

### ⑩ Timer 1 and timer 2 count start timing and count time when operation starts

Count starts from the first rising edge of the count source (2) after timer 1 and timer 2 operations start (1).

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

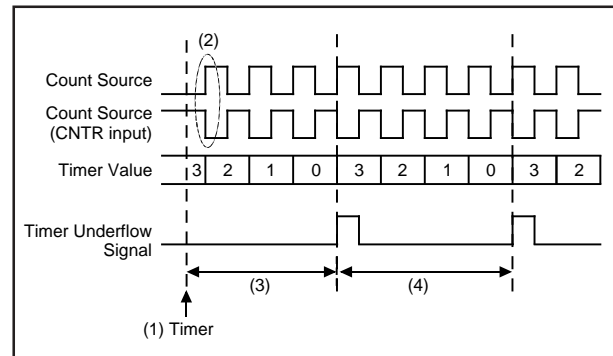


Fig. 46 Timer count start timing and count time when operation starts (T1, T2)

### ⑪ Watchdog timer

- The watchdog timer function is valid after system is released from reset. When not using the watchdog timer function, execute the DWDT instruction and the WRST instruction continuously, and clear the WEF flag to "0" to stop the watchdog timer function.
- The watchdog timer function is valid after system is returned from the RAM back-up. When not using the watchdog timer function, execute the DWDT instruction and the WRST instruction continuously every system is returned from the RAM back-up, and stop the watchdog timer function.

### ⑫ Multifunction

- The input/output of D2, D3, P12 and P13 can be used even when C, K, CNTR (input) and INT are selected.
- The input of P12 can be used even when CNTR (output) is selected.
- The input/output of P20 and P21 can be used even when AIN0 and AIN1 are selected.

### ⑬ Program counter

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

### ⑭ POF2 instruction

When the POF2 instruction is executed continuously after the EPOF instruction, system enters the RAM back-up state.

Note that system cannot enter the RAM back-up state when executing only the POF2 instruction.

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

⑤ P13/INT pin

Note [1] on bit 3 of register I1

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

- Depending on the input state of the P13/INT pin, the external 0 interrupt request flag (EXF0) may be set when the bit 3 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 47①) and then, change the bit 3 of register I1. In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 47②). Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 47③).

```

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

Fig. 47 External 0 interrupt program example-1

Note [2] on bit 3 of register I1

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

- When the key-on wakeup function of port P13 is not used (register K13 = "0"), clear bits 2 and 3 of register I1 before system enters to the RAM back-up mode. (refer to Figure 48①).

```

    ⋮
    LA    0      ; (00XX2)
    TI1A      ; Input of INT disabled .....①
    DI
    EPOF
    POF2      ; RAM back-up
    ⋮
    X : these bits are not used here.
    
```

Fig. 48 External 0 interrupt program example-2

Note [3] on bit 2 of register I1

When the interrupt valid waveform of the P13/INT pin is changed with the bit 2 of register I1 in software, be careful about the following notes.

- Depending on the input state of the P13/INT pin, the external 0 interrupt request flag (EXF0) may be set when the bit 2 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 49①) and then, change the bit 2 of register I1. In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 49②). Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 49③).

```

    ⋮
    LA    4      ; (XXX02)
    TV1A      ; The SNZ0 instruction is valid .....①
    LA    12     ; (X1XX2)
    TI1A      ; Interrupt valid waveform is changed
    NOP      ..... ②
    SNZ0      ; The SNZ0 instruction is executed
              (EXF0 flag cleared)
    NOP      ..... ③
    ⋮
    X : these bits are not used here.
    
```

Fig. 49 External 0 interrupt program example-3



⑥ Notes for the use of A/D conversion 1

Note the following when using the analog input pins also for port P2 function:

- Selection of analog input pins  
Even when P20/AIN0 and P21/AIN1 are set to pins for analog input, they continue to function as port P2 input/output. Accordingly, when any of them are used as I/O port and others are used as analog input pins, make sure to set the outputs of pins that are set for analog input to "1." Also, the port input function of the pin functions as an analog input is undefined.
- TALA instruction  
When the TALA instruction is executed, the low-order 2 bits of register AD is transferred to the high-order 2 bits of register A, simultaneously, the low-order 2 bits of register A is "0."

⑦ Notes for the use of A/D conversion 2

Do not change the operating mode (both A/D conversion mode and comparator mode) of A/D converter with the bit 3 of register Q1 while the A/D converter is operating.

When the operating mode of A/D converter is changed from the comparator mode to A/D conversion mode with the bit 3 of register Q1, note the following;

- Clear the bit 2 of register V2 to "0" (refer to Figure 50③) to change the operating mode of the A/D converter from the comparator mode to A/D conversion mode with the bit 3 of register Q1.
- The A/D conversion completion flag (ADF) may be set when the operating mode of the A/D converter is changed from the comparator mode to the A/D conversion mode. Accordingly, set a value to the bit 3 of register Q1, and execute the SNZAD instruction to clear the ADF flag.

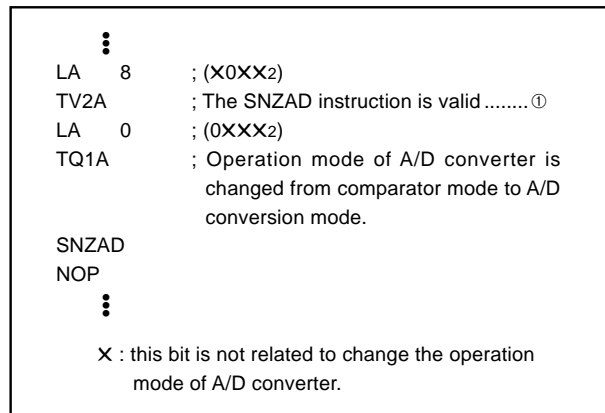


Fig. 50 A/D conversion interrupt program example

⑧ Notes for the use of A/D conversion 3

Each analog input pin is equipped with a capacitor which is used to compare the analog voltage. Accordingly, when the analog voltage is input from the circuit with high-impedance and, charge/discharge noise is generated and the sufficient A/D accuracy may not be obtained. Therefore, reduce the impedance or, connect a capacitor (0.01 μF to 1 μF) to analog input pins (Figure 51). When the overvoltage applied to the A/D conversion circuit may occur, connect an external circuit in order to keep the voltage within the rated range as shown in the Figure 52. In addition, test the application products sufficiently.

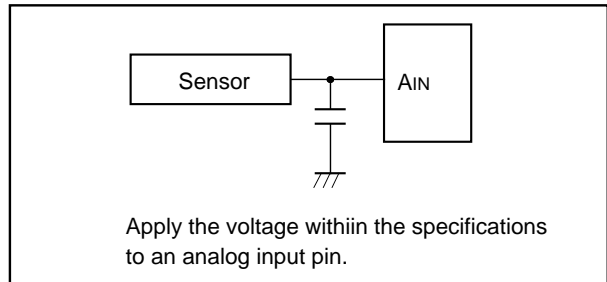


Fig. 51 Analog input external circuit example-1

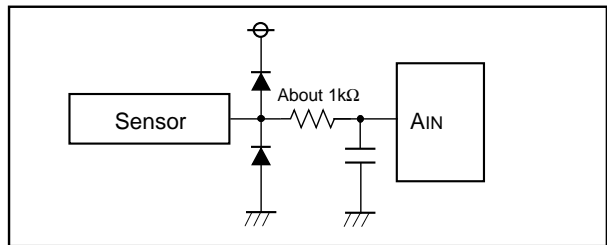


Fig. 52 Analog input external circuit example-2

**⑨ Clock control**

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

The oscillation circuit by the CMCK or CRCK instruction can be selected only at once. The oscillation circuit corresponding to the first executed one of these two instruction is valid. Other oscillation circuits and the on-chip oscillator stop.

**⑩ On-chip oscillator**

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

Be careful that variable frequencies when designing application products.

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

**⑪ External clock**

When the external signal clock is used as the source oscillation ( $f(XIN)$ ), note that the RAM back-up mode (POF2 instructions) cannot be used.

**⑫ Electric Characteristic Differences Between Mask ROM and One Time PROM Version MCU**

There are differences in electric characteristics, operation margin, noise immunity, and noise radiation between Mask ROM and One Time PROM version MCUs due to the difference in the manufacturing processes.

When manufacturing an application system with the One time PROM version and then switching to use of the Mask ROM version, please perform sufficient evaluations for the commercial samples of the Mask ROM version.

**⑬ Note on Power Source Voltage**

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

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

## CONTROL REGISTERS

Interrupt control register V1		at reset : 0000 <sub>2</sub>		at RAM back-up : 0000 <sub>2</sub>	R/W
V13	Timer 2 interrupt enable bit	0	Interrupt disabled (SNZT2 instruction is valid)		
		1	Interrupt enabled (SNZT2 instruction is invalid) (Note 2)		
V12	Timer 1 interrupt enable bit	0	Interrupt disabled (SNZT1 instruction is valid)		
		1	Interrupt enabled (SNZT1 instruction is invalid) (Note 2)		
V11	Not used	0	This bit has no function, but read/write is enabled.		
		1			
V10	External 0 interrupt enable bit	0	Interrupt disabled (SNZ0 instruction is valid)		
		1	Interrupt enabled (SNZ0 instruction is invalid) (Note 2)		

Interrupt control register V2		at reset : 0000 <sub>2</sub>		at RAM back-up : 0000 <sub>2</sub>	R/W
V23	Not used	0	This bit has no function, but read/write is enabled.		
		1			
V22	A/D interrupt enable bit	0	Interrupt disabled (SNZAD instruction is valid)		
		1	Interrupt enabled (SNZAD instruction is invalid) (Note 2)		
V21	Not used	0	This bit has no function, but read/write is enabled.		
		1			
V20	Not used	0	This bit has no function, but read/write is enabled.		
		1			

Interrupt control register I1		at reset : 0000 <sub>2</sub>		at RAM back-up : state retained	R/W
I13	INT pin input control bit (Note 3)	0	INT pin input disabled		
		1	INT pin input enabled		
I12	Interrupt valid waveform for INT pin/ return level selection bit (Note 3)	0	Falling waveform ("L" level of INT pin is recognized with the SNZIO instruction)/"L" level		
		1	Rising waveform ("H" level of INT pin is recognized with the SNZIO instruction)/"H" level		
I11	INT pin edge detection circuit control bit	0	One-sided edge detected		
		1	Both edges detected		
I10	INT pin timer 1 control enable bit	0	Disabled		
		1	Enabled		

Clock control register MR		at reset : 1100 <sub>2</sub>		at RAM back-up : 1100 <sub>2</sub>	R/W	
MR3	System clock selection bits	MR3	MR2	System clock		
		0	0			f(XIN) (high-speed mode)
		0	1			f(XIN)/2 (middle-speed mode)
		1	0			f(XIN)/4 (low-speed mode)
MR2		1	1	f(XIN)/8 (default mode)		
		0	This bit has no function, but read/write is enabled.			
1						
MR1	Not used	0	This bit has no function, but read/write is enabled.			
		1				
MR0	Not used	0	This bit has no function, but read/write is enabled.			
		1				

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

2: These instructions are equivalent to the NOP instruction.

3: When the contents of I12 and I13 are changed, the external interrupt request flag EXF0 may be set. Accordingly, clear EXF0 flag with the SNZ0 instruction when the bit 0 (V10) of register V1 to "0". In this time, set the NOP instruction after the SNZ0 instruction, for the case when a skip is performed with the SNZ0 instruction.

Timer control register W1		at reset : 0000z		at RAM back-up : 0000z	R/W
W13	Prescaler control bit	0	Stop (state initialized)		
		1	Operating		
W12	Prescaler dividing ratio selection bit	0	Instruction clock divided by 4		
		1	Instruction clock divided by 16		
W11	Timer 1 control bit	0	Stop (state retained)		
		1	Operating		
W10	Timer 1 count start synchronous circuit control bit	0	Count start synchronous circuit not selected		
		1	Count start synchronous circuit selected		

Timer control register W2		at reset : 0000z		at RAM back-up : state retained	R/W
W23	Timer 2 control bit	0	Stop (state retained)		
		1	Operating		
W22	Timer 1 count auto-stop circuit selection bit (Note 2)	0	Count auto-stop circuit not selected		
		1	Count auto-stop circuit selected		
W21	Timer 2 count source selection bits	W21	W20	Count source	
		0	0	Timer 1 underflow signal	
0		1	Prescaler output (ORCLK)		
1		0	CNTR input		
W20		1	1	System clock	

Timer control register W6		at reset : 0000z		at RAM back-up : state retained	R/W
W63	Not used	0	This bit has no function, but read/write is enabled.		
		1			
W62	Not used	0	This bit has no function, but read/write is enabled.		
		1			
W61	CNTR output selection bit	0	Timer 1 underflow signal divided by 2 output		
		1	Timer 2 underflow signal divided by 2 output		
W60	P12/CNTR function selection bit	0	P12(I/O)/CNTR input (Note 3)		
		1	P12 (input)/CNTR input/output (Note 3)		

A/D control register Q1		at reset : 0000z		at RAM back-up : state retained	R/W
Q13	A/D operation mode selection bit	0	A/D conversion mode		
		1	Comparator mode		
Q12	Not used	0	This bit has no function, but read/write is enabled.		
		1			
Q11	Analog input pin selection bits	Q11	Q10	Selected pins	
		0	0	AIN0	
		0	1	AIN1	
1		0	Not available		
Q10		1	1	Not available	

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

2: This function is valid only when the timer 1 count start synchronization circuit is selected.

3: CNTR input is valid only when CNTR input is selected as the timer 2 count source.

Key-on wakeup control register K0		at reset : 0000 <sub>2</sub>		at RAM back-up : state retained	R/W
K0 <sub>3</sub>	Port P0 <sub>3</sub> key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K0 <sub>2</sub>	Port P0 <sub>2</sub> key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K0 <sub>1</sub>	Port P0 <sub>1</sub> key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K0 <sub>0</sub>	Port P0 <sub>0</sub> key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K1		at reset : 0000 <sub>2</sub>		at RAM back-up : state retained	R/W
K1 <sub>3</sub>	Port P1 <sub>3</sub> /INT key-on wakeup control bit	0	P1 <sub>3</sub> key-on wakeup not used/INT pin key-on wakeup used		
		1	P1 <sub>3</sub> key-on wakeup used/INT pin key-on wakeup not used		
K1 <sub>2</sub>	Port P1 <sub>2</sub> /CNTR key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K1 <sub>1</sub>	Port P1 <sub>1</sub> key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K1 <sub>0</sub>	Port P1 <sub>0</sub> key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K2		at reset : 0000 <sub>2</sub>		at RAM back-up : state retained	R/W
K2 <sub>3</sub>	Port D <sub>3</sub> /K key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K2 <sub>2</sub>	Port D <sub>2</sub> /C key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K2 <sub>1</sub>	Port P2 <sub>1</sub> /AIN <sub>1</sub> key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K2 <sub>0</sub>	Port P2 <sub>0</sub> /AIN <sub>0</sub> key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

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

Pull-up control register PU0		at reset : 00002		at RAM back-up : state retained	W
PU03	Port P03 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU02	Port P02 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU01	Port P01 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU00	Port P00 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		

Pull-up control register PU1		at reset : 00002		at RAM back-up : state retained	W
PU13	Port P13/INT pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU12	Port P12/CNTR pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU11	Port P11 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU10	Port P10 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		

Pull-up control register PU2		at reset : 00002		at RAM back-up : state retained	W
PU23	Port D3/K pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU22	Port D2/C pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU21	Port P21/AIN1 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU20	Port P20/AIN0 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		

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

## INSTRUCTIONS

The 4506 Group has the 110 instructions. Each instruction is described as follows;

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

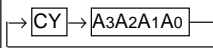
## SYMBOL

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

Symbol	Contents	Symbol	Contents
A	Register A (4 bits)	WDF1	Watchdog timer flag
B	Register B (4 bits)	WEF	Watchdog timer enable flag
DR	Register D (3 bits)	INTE	Interrupt enable flag
E	Register E (8 bits)	EXF0	External 0 interrupt request flag
Q1	A/D control register Q1 (4 bits)	P	Power down flag
V1	Interrupt control register V1 (4 bits)	ADF	A/D conversion completion flag
V2	Interrupt control register V2 (4 bits)		
I1	Interrupt control register I1 (4 bits)	D	Port D (4 bits)
W1	Timer control register W1 (4 bits)	P0	Port P0 (4 bits)
W2	Timer control register W2 (4 bits)	P1	Port P1 (4 bits)
W6	Timer control register W6 (4 bits)	P2	Port P2 (2 bits)
MR	Clock control register MR (4 bits)	C	Port C (1 bit)
K0	Key-on wakeup control register K0 (4 bits)	K	Port K (1 bit)
K1	Key-on wakeup control register K1 (4 bits)		
K2	Key-on wakeup control register K2 (4 bits)	x	Hexadecimal variable
PU0	Pull-up control register PU0 (4 bits)	y	Hexadecimal variable
PU1	Pull-up control register PU1 (4 bits)	z	Hexadecimal variable
PU2	Pull-up control register PU2 (4 bits)	p	Hexadecimal variable
X	Register X (4 bits)	n	Hexadecimal constant
Y	Register Y (4 bits)	i	Hexadecimal constant
Z	Register Z (2 bits)	j	Hexadecimal constant
DP	Data pointer (10 bits) (It consists of registers X, Y, and Z)	A3A2A1A0	Binary notation of hexadecimal variable A (same for others)
PC	Program counter (14 bits)		
PCH	High-order 7 bits of program counter	←	Direction of data movement
PCL	Low-order 7 bits of program counter	↔	Data exchange between a register and memory
SK	Stack register (14 bits X 8)	?	Decision of state shown before “?”
SP	Stack pointer (3 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
R2	Timer 2 reload register	a	Label indicating address a6 a5 a4 a3 a2 a1 a0
T1	Timer 1	p, a	Label indicating address a6 a5 a4 a3 a2 a1 a0 in page p5 p4 p3 p2 p1 p0
T2	Timer 2		
T1F	Timer 1 interrupt request flag	C	Hex. C + Hex. number x (also same for others)
T2F	Timer 2 interrupt request flag	+	
		x	

Note : Some instructions of the 4506 Group has the skip function to unexecute the next described instruction. The 4506 Group just invalidates the next instruction when a skip is performed. The contents of program counter is not increased by 2. Accordingly, the number of cycles does not change even if skip is not performed. However, the cycle count becomes “1” if the TABP p, RT, or RTS instruction is skipped.

**INDEX LIST OF INSTRUCTION FUNCTION**

Grouping	Mnemonic	Function	Page	Grouping	Mnemonic	Function	Page	
Register to register transfer	TAB	$(A) \leftarrow (B)$	75, 88	RAM to register transfer	XAMI j	$(A) \leftarrow \rightarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15 $(Y) \leftarrow (Y) + 1$	87, 88	
	TBA	$(B) \leftarrow (A)$	81, 88		TMA j	$(M(DP)) \leftarrow (A)$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15	83, 88	
	TAY	$(A) \leftarrow (Y)$	81, 88		Arithmetic operation	LA n	$(A) \leftarrow n$ n = 0 to 15	66, 90
	TYA	$(Y) \leftarrow (A)$	86, 88			TABP p	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ (Note) $(PCL) \leftarrow (DR2-DR0, A3-A0)$ $(B) \leftarrow (ROM(PC))_{7-4}$ $(A) \leftarrow (ROM(PC))_{3-0}$ $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$	76, 90
	TEAB	$(E7-E4) \leftarrow (B)$ $(E3-E0) \leftarrow (A)$	82, 88			AM	$(A) \leftarrow (A) + (M(DP))$	60, 90
	TABE	$(B) \leftarrow (E7-E4)$ $(A) \leftarrow (E3-E0)$	76, 88			AMC	$(A) \leftarrow (A) + (M(DP)) + (CY)$ $(CY) \leftarrow$ Carry	60, 90
	TDA	$(DR2-DR0) \leftarrow (A2-A0)$	81, 88			A n	$(A) \leftarrow (A) + n$ n = 0 to 15	60, 90
	TAD	$(A2-A0) \leftarrow (DR2-DR0)$ $(A3) \leftarrow 0$	76, 88			AND	$(A) \leftarrow (A) AND (M(DP))$	61, 90
	TAZ	$(A1, A0) \leftarrow (Z1, Z0)$ $(A3, A2) \leftarrow 0$	81, 88			OR	$(A) \leftarrow (A) OR (M(DP))$	68, 90
	TAX	$(A) \leftarrow (X)$	80, 88			SC	$(CY) \leftarrow 1$	71, 90
	TASP	$(A2-A0) \leftarrow (SP2-SP0)$ $(A3) \leftarrow 0$	79, 88			RC	$(CY) \leftarrow 0$	69, 90
	RAM addresses	LXY x, y	$(X) \leftarrow x$ x = 0 to 15 $(Y) \leftarrow y$ y = 0 to 15			66, 88	SZC	$(CY) = 0 ?$
LZ z		$(Z) \leftarrow z$ z = 0 to 3	66, 88	CMA	$(A) \leftarrow (\bar{A})$	63, 90		
INY		$(Y) \leftarrow (Y) + 1$	66, 88	RAR		68, 90		
DEY		$(Y) \leftarrow (Y) - 1$	63, 88					
RAM to register transfer	TAM j	$(A) \leftarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15	78, 88					
	XAM j	$(A) \leftarrow \rightarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15	86, 88					
	XAMD j	$(A) \leftarrow \rightarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15 $(Y) \leftarrow (Y) - 1$	87, 88					

Note: p is 0 to 15 for M34506M2,  
p is 0 to 31 for M34506M4/E4.



## INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function	Page	Grouping	Mnemonic	Function	Page
Bit operation	SB j	$(Mj(DP)) \leftarrow 1$ j = 0 to 3	70, 90	Interrupt operation	DI	$(INTE) \leftarrow 0$	64, 94
	RB j	$(Mj(DP)) \leftarrow 0$ j = 0 to 3	69, 90		EI	$(INTE) \leftarrow 1$	64, 94
	SZB j	$(Mj(DP)) = 0 ?$ j = 0 to 3	74, 90		SNZ0	V10 = 0: (EXF0) = 1 ? After skipping, (EXF0) $\leftarrow$ 0 V10 = 1: SNZ0 = NOP	72, 94
Comparison operation	SEAM	$(A) = (M(DP)) ?$	72, 90		SNZI0	I12 = 1 : (INT) = "H" ? I12 = 0 : (INT) = "L" ?	73, 94
	SEA n	$(A) = n ?$ n = 0 to 15	71, 90		TAV1	$(A) \leftarrow (V1)$	79, 94
Branch operation	B a	$(PCL) \leftarrow a6-a0$	61, 92		TV1A	$(V1) \leftarrow (A)$	85, 94
	BL p, a	$(PCH) \leftarrow p$ (Note) $(PCL) \leftarrow a6-a0$	61, 92		TAV2	$(A) \leftarrow (V2)$	79, 94
	BLA p	$(PCH) \leftarrow p$ (Note) $(PCL) \leftarrow (DR2-DR0, A3-A0)$	61, 92		TV2A	$(V2) \leftarrow (A)$	85, 94
Subroutine operation	BM a	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow 2$ $(PCL) \leftarrow a6-a0$	62, 92		TAI1	$(A) \leftarrow (I1)$	77, 94
	BML p, a	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ (Note) $(PCL) \leftarrow a6-a0$	62, 92		TI1A	$(I1) \leftarrow (A)$	82, 94
	BMLA p	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ (Note) $(PCL) \leftarrow (DR2-DR0, A3-A0)$	62, 92	TAW1	$(A) \leftarrow (W1)$	80, 94	
Return operation	RTI	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$	70, 92	TW1A	$(W1) \leftarrow (A)$	85, 94	
	RT	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$	70, 92	TAW2	$(A) \leftarrow (W2)$	80, 94	
	RTS	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$	70, 92	TW2A	$(W2) \leftarrow (A)$	85, 94	
				Timer operation	TAW6	$(A) \leftarrow (W6)$	80, 94
					TW6A	$(W6) \leftarrow (A)$	86, 94
					TAB1	$(B) \leftarrow (T17-T14)$ $(A) \leftarrow (T13-T10)$	75, 94
					T1AB	$(R17-R14) \leftarrow (B)$ $(T17-T14) \leftarrow (B)$ $(R13-R10) \leftarrow (A)$ $(T13-T10) \leftarrow (A)$	74, 94
					TAB2	$(B) \leftarrow (T27-T24)$ $(A) \leftarrow (T23-T20)$	75, 94
					T2AB	$(R27-R24) \leftarrow (B)$ $(T27-T24) \leftarrow (B)$ $(R23-R20) \leftarrow (A)$ $(T23-T20) \leftarrow (A)$	75, 94

Note: p is 0 to 15 for M34506M2,  
p is 0 to 31 for M34506M4/E4.

## INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function	Page	Grouping	Mnemonic	Function	Page
Timer operation	TR1AB	(R17–R14) ← (B) (R13–R10) ← (A)	84, 94	Input/Output operation	IAK	(A0) ← (K) (A3–A1) ← 0	65, 96
	SNZT1	V12 = 0: (T1F) = 1 ? After skipping, (T1F) ← 0 V12 = 1: SNZT1 = NOP	73, 94		OKA	(K) ← (A0)	67, 96
	SNZT2	V13 = 0: (T2F) = 1 ? After skipping, (T2F) ← 0 V13 = 1: SNZT2 = NOP	73, 94		TK0A	(K0) ← (A)	82, 96
Input/Output operation	IAP0	(A) ← (P0)	65, 96		TAK0	(A) ← (K0)	77, 96
	OP0A	(P0) ← (A)	67, 96		TK1A	(K1) ← (A)	82, 96
	IAP1	(A) ← (P1)	65, 96		TAK1	(A) ← (K1)	77, 96
	OP1A	(P1) ← (A)	67, 96		TK2A	(K2) ← (A)	83, 96
	IAP2	(A1, A0) ← (P21, P20) (A3, A2) ← 0	65, 96		TAK2	(A) ← (K2)	78, 96
	OP2A	(P21, P20) ← (A1, A0)	68, 96		TPU0A	(PU0) ← (A)	83, 96
	CLD	(D) ← 1	62, 96		TPU1A	(PU1) ← (A)	84, 96
	RD	(D(Y)) ← 0 (Y) = 0 to 3	69, 96	TPU2A	(PU2) ← (A)	84, 96	
	SD	(D(Y)) ← 1 (Y) = 0 to 3	71, 96	A/D conversion operation	TABAD	In A/D conversion mode (Q13 = 0), (B) ← (AD9–AD6) (A) ← (AD5–AD2) In comparator mode (Q13 = 1), (B) ← (AD7–AD4) (A) ← (AD3–AD0)	76, 98
	SZD	(D(Y)) = 0 ? (Y) = 0 to 3	74, 96		TALA	(A3, A2) ← (AD1, AD0) (A1, A0) ← 0	78, 98
	SCP	(C) ← 1	71, 96		TADAB	(AD7–AD4) ← (B) (AD3–AD0) ← (A)	77, 98
	RCP	(C) ← 0	69, 96		TAQ1	(A) ← (Q1)	79, 98
	SNZCP	(C) = 1 ?	72, 96		TQ1A	(Q1) ← (A)	84, 98
					ADST	(ADF) ← 0 Q13 = 0: A/D conversion starting Q13 = 1: Comparator operation starting	60, 98
			SNZAD		V22 = 0: (ADF) = 1 ? After skipping, (ADF) ← 0 V22 = 1: SNZAD = NOP	72, 98	

**INDEX LIST OF INSTRUCTION FUNCTION (continued)**

Grouping	Mnemonic	Function	Page
Other operation	NOP	$(PC) \leftarrow (PC) + 1$	67, 98
	POF2	RAM back-up	68, 98
	EPOF	POF2 instructions valid	64, 98
	SNZP	$(P) = 1 ?$	73, 98
	DWDT	Stop of watchdog timer function enabled	64, 98
	WRST	$(WDF1) = 1 ?$ After skipping, $(WDF1) \leftarrow 0$	86, 98
	CMCK	Ceramic resonance circuit selected	63, 98
	CRCK	RC oscillation circuit selected	63, 98
	TAMR	$(A) \leftarrow (MR)$	78, 98
	TMRA	$(MR) \leftarrow (A)$	83, 98

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET)

### A n (Add n and accumulator)

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

### ADST (A/D conversion Start)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 1 0 0 0 1 1 1 1 1	1	1	–	–
<b>Operation:</b> $(ADF) \leftarrow 0$ Q13 = 0: A/D conversion starting Q13 = 1: Comparator operation starting (Q13 : bit 3 of A/D control register Q1)			<b>Grouping:</b> A/D conversion operation <b>Description:</b> Clears (0) to A/D conversion completion flag ADF, and the A/D conversion at the A/D conversion mode (Q13 = 0) or the comparator operation at the comparator mode (Q13 = 1) is started.			

### AM (Add accumulator and Memory)

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

### AMC (Add accumulator, Memory and Carry)

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

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### AND (logical AND between accumulator and memory)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition												
	0	0	0	0	0	1	1	0	0	0	0	0	1	8	1	1	-	-
<b>Operation:</b>	(A) ← (A) AND (M(DP))											<b>Grouping:</b>	Arithmetic operation					
												<b>Description:</b>	Takes the AND operation between the contents of register A and the contents of M(DP), and stores the result in register A.					

### B a (Branch to address a)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition											
	0	1	1	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	1	8	a	1	1	-	-
<b>Operation:</b>	(PCL) ← a <sub>6</sub> to a <sub>0</sub>											<b>Grouping:</b>	Branch operation				
												<b>Description:</b>	Branch within a page : Branches to address a in the identical page.				
												<b>Note:</b>	Specify the branch address within the page including this instruction.				

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition											
	0	0	1	1	1	p <sub>4</sub>	p <sub>3</sub>	p <sub>2</sub>	p <sub>1</sub>	p <sub>0</sub>	0	E	p	2	2	-	-
	1	0	0	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	2	a	a	2	2	-	-
<b>Operation:</b>	(PCH) ← p (PCL) ← a <sub>6</sub> to a <sub>0</sub>											<b>Grouping:</b>	Branch operation				
												<b>Description:</b>	Branch out of a page : Branches to address a in page p.				
												<b>Note:</b>	p is 0 to 15 for M34506M2, and p is 0 to 31 for M34506M4/E4.				

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition											
	0	0	0	0	0	1	0	0	0	0	0	1	0	2	2	-	-
	1	0	0	p <sub>4</sub>	0	0	p <sub>3</sub>	p <sub>2</sub>	p <sub>1</sub>	p <sub>0</sub>	2	p	p	2	2	-	-
<b>Operation:</b>	(PCH) ← p (PCL) ← (DR <sub>2</sub> -DR <sub>0</sub> , A <sub>3</sub> -A <sub>0</sub> )											<b>Grouping:</b>	Branch operation				
												<b>Description:</b>	Branch out of a page : Branches to address (DR <sub>2</sub> DR <sub>1</sub> DR <sub>0</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub> ) <sub>2</sub> specified by registers D and A in page p.				
												<b>Note:</b>	p is 0 to 15 for M34506M2, and p is 0 to 31 for M34506M4/E4.				

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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

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

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

Instruction code	D9	D0										Number of words	Number of cycles	Flag CY	Skip condition				
	0	0	1	1	0	p4	p3	p2	p1	p0	2	0	C	p	16	2	2	-	-
	1	0	0	a6	a5	a4	a3	a2	a1	a0	2	2	a	a	16				
<b>Operation:</b>	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ $(PCL) \leftarrow a6-a0$																		
<b>Grouping:</b>	Subroutine call operation																		
<b>Description:</b>	Call the subroutine : Calls the subroutine at address a in page p.																		
<b>Note:</b>	p is 0 to 15 for M34506M2, and p is 0 to 31 for M34506M4/E4. Be careful not to over the stack because the maximum level of subroutine nesting is 8.																		

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

Instruction code	D9	D0										Number of words	Number of cycles	Flag CY	Skip condition				
	0	0	0	0	1	1	0	0	0	0	2	0	3	0	16	2	2	-	-
	1	0	0	p4	0	0	p3	p2	p1	p0	2	2	p	p	16				
<b>Operation:</b>	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ $(PCL) \leftarrow (DR2-DR0, A3-A0)$																		
<b>Grouping:</b>	Subroutine call operation																		
<b>Description:</b>	Call the subroutine : Calls the subroutine at address (DR2 DR1 DR0 A3 A2 A1 A0) <sub>2</sub> specified by registers D and A in page p.																		
<b>Note:</b>	p is 0 to 15 for M34506M2, and p is 0 to 31 for M34506M4/E4. Be careful not to over the stack because the maximum level of subroutine nesting is 8.																		

### CLD (Clear port D)

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

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### CMA (CoMplement of Accumulator)

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

### CMCK (Clock select: ceraMic resonance Clock)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	1	0	1	0	0	1	1	0	1	0	2	2	9	A	16	1	1	-	-
<b>Operation:</b>	Ceramic resonance circuit selected											<b>Grouping:</b>	Other operation						
												<b>Description:</b>	Selects the ceramic resonance circuit and stops the on-chip oscillator.						

### CRCK (Clock select: Rc oscillation Clock)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	1	0	1	0	0	1	1	0	1	1	2	2	9	B	16	1	1	-	-
<b>Operation:</b>	RC oscillation circuit selected											<b>Grouping:</b>	Other operation						
												<b>Description:</b>	Selects the RC oscillation circuit and stops the on-chip oscillator.						

### DEY (DEcrement register Y)

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

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### DI (Disable Interrupt)

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

### DWDT (Disable WatchDog Timer)

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

### EI (Enable Interrupt)

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

### EPOF (Enable POF instruction)

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



## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### I<sub>AK</sub> (Input Accumulator from port K)

Instruction code	D <sub>9</sub>	D <sub>0</sub>	Number of words	Number of cycles	Flag CY	Skip condition													
	1	0	0	1	1	0	1	1	1	1	2	2	6	F	16	1	1	–	–
<b>Operation:</b>	(A <sub>0</sub> ) ← (K)										<b>Grouping:</b>		Input/Output operation						
	(A <sub>3</sub> –A <sub>1</sub> ) ← 0										<b>Description:</b>		Transfers the contents of port K to the bit 0 (A <sub>0</sub> ) of register A.						
											<b>Note:</b>		After this instruction is executed, “0” is stored to the high-order 3 bits (A <sub>3</sub> –A <sub>1</sub> ) of register A.						

### I<sub>AP0</sub> (Input Accumulator from port P0)

Instruction code	D <sub>9</sub>	D <sub>0</sub>	Number of words	Number of cycles	Flag CY	Skip condition													
	1	0	0	1	1	0	0	0	0	0	2	2	6	0	16	1	1	–	–
<b>Operation:</b>	(A) ← (P0)										<b>Grouping:</b>		Input/Output operation						
											<b>Description:</b>		Transfers the input of port P0 to register A.						

### I<sub>AP1</sub> (Input Accumulator from port P1)

Instruction code	D <sub>9</sub>	D <sub>0</sub>	Number of words	Number of cycles	Flag CY	Skip condition													
	1	0	0	1	1	0	0	0	0	1	2	2	6	1	16	1	1	–	–
<b>Operation:</b>	(A) ← (P1)										<b>Grouping:</b>		Input/Output operation						
											<b>Description:</b>		Transfers the input of port P1 to register A.						

### I<sub>AP2</sub> (Input Accumulator from port P2)

Instruction code	D <sub>9</sub>	D <sub>0</sub>	Number of words	Number of cycles	Flag CY	Skip condition													
	1	0	0	1	1	0	0	0	1	0	2	2	6	2	16	1	1	–	–
<b>Operation:</b>	(A <sub>1</sub> , A <sub>0</sub> ) ← (P2 <sub>1</sub> , P2 <sub>0</sub> )										<b>Grouping:</b>		Input/Output operation						
	(A <sub>3</sub> , A <sub>2</sub> ) ← 0										<b>Description:</b>		Transfers the input of port P2 to the low-order 2 bits (A <sub>1</sub> , A <sub>0</sub> ) of register A.						
											<b>Note:</b>		After this instruction is executed, “0” is stored to the high-order 2 bits (A <sub>3</sub> , A <sub>2</sub> ) of register A.						

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### INY (INcrement register Y)

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

### LA n (Load n in Accumulator)

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

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	1 x <sub>3</sub> x <sub>2</sub> x <sub>1</sub> x <sub>0</sub> y <sub>3</sub> y <sub>2</sub> y <sub>1</sub> y <sub>0</sub>	1	1	–	Continuous description
<b>Operation:</b> $(X) \leftarrow x$ $x = 0$ to $15$ $(Y) \leftarrow y$ $y = 0$ to $15$			<b>Grouping:</b> RAM addresses			
			<b>Description:</b> Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y. When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped.			

### LZ z (Load register Z with z)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0	0 0 0 1 0 0 1 0 z <sub>1</sub> z <sub>0</sub>	1	1	–	–
<b>Operation:</b> $(Z) \leftarrow z$ $z = 0$ to $3$			<b>Grouping:</b> RAM addresses			
			<b>Description:</b> Loads the value z in the immediate field to register Z.			

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### NOP (No Operation)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-	-
<b>Operation:</b>	$(PC) \leftarrow (PC) + 1$												<b>Grouping:</b>	Other operation					
													<b>Description:</b>	No operation; Adds 1 to program counter value, and others remain unchanged.					

### OKA (Output port K from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																
	1	0	0	0	0	1	1	1	1	1	1	1	1	1	2	2	1	F	1	1	-	-
<b>Operation:</b>	$(K) \leftarrow (A_0)$												<b>Grouping:</b>	Input/Output operation								
													<b>Description:</b>	Outputs the contents of bit 0 (A <sub>0</sub> ) of register A to port K.								

### OP0A (Output port P0 from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																
	1	0	0	0	1	0	0	0	0	0	0	0	0	0	2	2	2	0	1	1	-	-
<b>Operation:</b>	$(P_0) \leftarrow (A)$												<b>Grouping:</b>	Input/Output operation								
													<b>Description:</b>	Outputs the contents of register A to port P0.								

### OP1A (Output port P1 from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																
	1	0	0	0	1	0	0	0	0	0	1	1	1	1	2	2	2	1	1	1	-	-
<b>Operation:</b>	$(P_1) \leftarrow (A)$												<b>Grouping:</b>	Input/Output operation								
													<b>Description:</b>	Outputs the contents of register A to port P1.								

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### OP2A (Output port P2 from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 0 0 0 1 0 0 0 1 0	1	1	–	–
<b>Operation:</b>	(P21, P20) ← (A1, A0)		<b>Grouping:</b>	Input/Output operation		
			<b>Description:</b>	Outputs the contents of the low-order 2 bits (A1, A0) of register A to port P2.		

### OR (logical OR between accumulator and memory)

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

### POF2 (Power Off2)

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

### RAR (Rotate Accumulator Right)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0	0 0 0 0 0 1 1 1 0 1	1	1	0/1	–
<b>Operation:</b>	→[CY]→[A3A2A1A0]		<b>Grouping:</b>	Arithmetic operation		
			<b>Description:</b>	Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right.		

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### RB j (Reset Bit)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	0	0	0	1	0	0	1	1	j	j	2	0	4	C+j	16	1	1	-	-
<b>Operation:</b>	(Mj(DP)) ← 0 j = 0 to 3											<b>Grouping:</b>	Bit operation						
												<b>Description:</b>	Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).						

### RC (Reset Carry flag)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	0	0	0	0	0	0	0	1	1	0	2	0	0	6	16	1	1	0	-
<b>Operation:</b>	(CY) ← 0											<b>Grouping:</b>	Arithmetic operation						
												<b>Description:</b>	Clears (0) to carry flag CY.						

### RCP (Reset Port C)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	1	0	1	0	0	0	1	1	0	0	2	2	8	C	16	1	1	-	-
<b>Operation:</b>	(C) ← 0											<b>Grouping:</b>	Input/Output operation						
												<b>Description:</b>	Clears (0) to port C.						

### RD (Reset port D specified by register Y)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	0	0	0	0	0	1	0	1	0	0	2	0	1	4	16	1	1	-	-
<b>Operation:</b>	(D(Y)) ← 0 However, (Y) = 0 to 3											<b>Grouping:</b>	Input/Output operation						
												<b>Description:</b>	Clears (0) to a bit of port D specified by register Y. Set 0 to 3 to register Y because port D is four ports (D0–D3). <b>Note:</b> When values except above are set to register Y, this instruction is equivalent to the NOP instruction.						

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### RT (ReTurn from subroutine)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition											
	0	0	0	1	0	0	0	1	0	0	0	4	4	1	2	-	-
<b>Operation:</b>	(PC) ← (SK(SP)) (SP) ← (SP) - 1						<b>Grouping:</b>	Return operation									
							<b>Description:</b>	Returns from subroutine to the routine called the subroutine.									

### RTI (ReTurn from Interrupt)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition											
	0	0	0	1	0	0	0	1	1	0	0	4	6	1	1	-	-
<b>Operation:</b>	(PC) ← (SK(SP)) (SP) ← (SP) - 1						<b>Grouping:</b>	Return operation									
							<b>Description:</b>	Returns from interrupt service routine to main routine. Returns each value of data pointer (X, Y, Z), carry flag, skip status, NOP mode status by the continuous description of the LA/LXY instruction, register A and register B to the states just before interrupt.									

### RTS (ReTurn from subroutine and Skip)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition											
	0	0	0	1	0	0	0	1	0	1	0	4	5	1	2	-	Skip at uncondition
<b>Operation:</b>	(PC) ← (SK(SP)) (SP) ← (SP) - 1						<b>Grouping:</b>	Return operation									
							<b>Description:</b>	Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition.									

### SB j (Set Bit)

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

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### SC (Set Carry flag)

Instruction code	D <sub>9</sub>	D <sub>0</sub>	Number of words	Number of cycles	Flag CY	Skip condition
	0 0 0 0 0 0 0 0	1 1 1	1	1	1	–
<b>Operation:</b>	(CY) ← 1		<b>Grouping:</b> Arithmetic operation			
			<b>Description:</b> Sets (1) to carry flag CY.			

### SCP (Set Port C)

Instruction code	D <sub>9</sub>	D <sub>0</sub>	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 1 0 0 0 1 1	0 1	1	1	–	–
<b>Operation:</b>	(C) ← 1		<b>Grouping:</b> Input/Output operation			
			<b>Description:</b> Sets (1) to port C.			

### SD (Set port D specified by register Y)

Instruction code	D <sub>9</sub>	D <sub>0</sub>	Number of words	Number of cycles	Flag CY	Skip condition
	0 0 0 0 0 1 0 1	0 1	1	1	–	–
<b>Operation:</b>	(D(Y)) ← 1 (Y) = 0 to 3		<b>Grouping:</b> Input/Output operation			
			<b>Description:</b> Sets (1) to a bit of port D specified by register Y. <b>Note:</b> Set 0 to 3 to register Y because port D is four ports (D <sub>0</sub> –D <sub>3</sub> ). When values except above are set to register Y, this instruction is equivalent to the NOP instruction.			

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

Instruction code	D <sub>9</sub>	D <sub>0</sub>	Number of words	Number of cycles	Flag CY	Skip condition
	0 0 0 0 1 0 0 1	0 1	2	2	–	(A) = n
	0 0 0 1 1 1	n n n n	2	2	–	(A) = n
<b>Operation:</b>	(A) = n ? n = 0 to 15		<b>Grouping:</b> Comparison operation			
			<b>Description:</b> Skips the next instruction when the contents of register A is equal to the value n in the immediate field. Executes the next instruction when the contents of register A is not equal to the value n in the immediate field.			

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****SEAM (Skip Equal, Accumulator with Memory)**

Instruction code	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0	0	0	0	1	0	0	1	1	0	1	1	0	(A) = (M(DP))
											1	1	–	(A) = (M(DP))
<b>Operation:</b>	(A) = (M(DP)) ?										<b>Grouping:</b> Comparison operation			
											<b>Description:</b> Skips the next instruction when the contents of register A is equal to the contents of M(DP). Executes the next instruction when the contents of register A is not equal to the contents of M(DP).			

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

Instruction code	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0	0	0	0	1	1	1	0	0	0	0	3	8	V10 = 0: (EXF0) = 1
											1	1	–	V10 = 0: (EXF0) = 1
<b>Operation:</b>	V10 = 0: (EXF0) = 1 ? After skipping, (EXF0) ← 0 V10 = 1: SNZ0 = NOP (V10 : bit 0 of the interrupt control register V1)										<b>Grouping:</b> Interrupt operation			
											<b>Description:</b> When V10 = 0 : Skips the next instruction when external 0 interrupt request flag EXF0 is "1." After skipping, clears (0) to the EXF0 flag. When the EXF0 flag is "0," executes the next instruction. When V10 = 1 : This instruction is equivalent to the NOP instruction.			

**SNZAD (Skip if Non Zero condition of A/D conversion completion flag)**

Instruction code	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0	1	0	0	0	0	1	1	1	2	8	7	V22 = 0: (ADF) = 1
											1	1	–	V22 = 0: (ADF) = 1
<b>Operation:</b>	V22 = 0: (ADF) = 1 ? After skipping, (ADF) ← 0 V22 = 1: SNZAD = NOP (V22 : bit 2 of the interrupt control register V2)										<b>Grouping:</b> A/D conversion operation			
											<b>Description:</b> When V22 = 0 : Skips the next instruction when A/D conversion completion flag ADF is "1." After skipping, clears (0) to the ADF flag. When the ADF flag is "0," executes the next instruction. When V22 = 1 : This instruction is equivalent to the NOP instruction.			

**SNZCP (Skip if Non Zero condition of Port C)**

Instruction code	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0	1	0	0	0	1	0	0	1	2	8	9	(C) = 1
											1	1	–	(C) = 1
<b>Operation:</b>	(C) = 1 ?										<b>Grouping:</b> Input/Output operation			
											<b>Description:</b> Skips the next instruction when the contents of port C is "1." Executes the next instruction when the contents of port C is "0."			



## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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

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

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

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

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

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

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

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

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### SZB j (Skip if Zero, Bit)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	0	0	0	0	1	0	0	0	j	j	2	0	2	j	16	1	1	–	$(M_j(DP)) = 0$ $j = 0 \text{ to } 3$
<b>Operation:</b>	$(M_j(DP)) = 0 ?$ $j = 0 \text{ to } 3$											<b>Grouping:</b> Bit operation							
												<b>Description:</b> Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is "0." Executes the next instruction when the contents of bit j of M(DP) is "1."							

### SZC (Skip if Zero, Carry flag)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	0	0	0	0	1	0	1	1	1	1	2	0	2	F	16	1	1	–	$(CY) = 0$
<b>Operation:</b>	$(CY) = 0 ?$											<b>Grouping:</b> Arithmetic operation							
												<b>Description:</b> Skips the next instruction when the contents of carry flag CY is "0." After skipping, the CY flag remains unchanged. Executes the next instruction when the contents of the CY flag is "1."							

### SZD (Skip if Zero, port D specified by register Y)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	0	0	0	0	1	0	0	1	0	0	2	0	2	4	16	2	2	–	$(D(Y)) = 0$ $(Y) = 0 \text{ to } 3$
	0	0	0	0	1	0	1	0	1	1	2	0	2	B	16				
<b>Operation:</b>	$(D(Y)) = 0 ?$ $(Y) = 0 \text{ to } 3$											<b>Grouping:</b> Input/Output operation							
												<b>Description:</b> Skips the next instruction when a bit of port D specified by register Y is "0." Executes the next instruction when the bit is "1." <b>Note:</b> Set 0 to 3 to register Y because port D is four ports (D0–D3). When values except above are set to register Y, this instruction is equivalent to the NOP instruction.							

### T1AB (Transfer data to timer 1 and register R1 from Accumulator and register B)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	1	0	0	0	1	1	0	0	0	0	2	2	3	0	16	1	1	–	–
<b>Operation:</b>	$(T17-T14) \leftarrow (B)$ $(R17-R14) \leftarrow (B)$ $(T13-T10) \leftarrow (A)$ $(R13-R10) \leftarrow (A)$											<b>Grouping:</b> Timer operation							
												<b>Description:</b> Transfers the contents of register B to the high-order 4 bits of timer 1 and timer 1 reload register R1. Transfers the contents of register A to the low-order 4 bits of timer 1 and timer 1 reload register R1.							

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### T2AB (Transfer data to timer 2 and register R2 from Accumulator and register B)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 0 0 1 1 0 0 0 1	1	1	–	–
<b>Operation:</b>	(T27–T24) ← (B) (R27–R24) ← (B) (T23–T20) ← (A) (R23–R20) ← (A)		<b>Grouping:</b> Timer operation <b>Description:</b> Transfers the contents of register B to the high-order 4 bits of timer 2 and timer 2 reload register R2. Transfers the contents of register A to the low-order 4 bits of timer 2 and timer 2 reload register R2.			

### TAB (Transfer data to Accumulator from register B)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0	0 0 0 0 1 1 1 1 0	1	1	–	–
<b>Operation:</b>	(A) ← (B)		<b>Grouping:</b> Other operation <b>Description:</b> Transfers the contents of register B to register A.			

### TAB1 (Transfer data to Accumulator and register B from timer 1)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 0 1 1 1 0 0 0 0	1	1	–	–
<b>Operation:</b>	(B) ← (T17–T14) (A) ← (T13–T10)		<b>Grouping:</b> Timer operation <b>Description:</b> Transfers the high-order 4 bits (T17–T14) of timer 1 to register B. Transfers the low-order 4 bits (T13–T10) of timer 1 to register A.			

### TAB2 (Transfer data to Accumulator and register B from timer 2)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 0 1 1 1 0 0 0 1	1	1	–	–
<b>Operation:</b>	(B) ← (T27–T24) (A) ← (T23–T20)		<b>Grouping:</b> Timer operation <b>Description:</b> Transfers the high-order 4 bits (T27–T24) of timer 2 to register B. Transfers the low-order 4 bits (T23–T20) of timer 2 to register A.			

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### TABAD (Transfer data to Accumulator and register B from register AD)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 0 1 1 1 1 0 0 1	1	1	–	–
<b>Operation:</b>	In A/D conversion mode (Q13 = 0), (B) ← (AD9–AD6) (A) ← (AD5–AD2) In comparator mode (Q13 = 1), (B) ← (AD7–AD4) (A) ← (AD3–AD0) (Q13 : bit 3 of A/D control register Q1)		<b>Grouping:</b>	A/D conversion operation		
			<b>Description:</b>	In the A/D conversion mode (Q13 = 0), transfers the high-order 4 bits (AD9–AD6) of register AD to register B, and the middle-order 4 bits (AD5–AD2) of register AD to register A. In the comparator mode (Q13 = 1), transfers the high-order 4 bits (AD7–AD4) of comparator register to register B, and the low-order 4 bits (AD3–AD0) of comparator register to register A.		

### TABE (Transfer data to Accumulator and register B from register E)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0	0 0 0 0 1 0 1 0 1 0	1	1	–	–
<b>Operation:</b>	(B) ← (E7–E4) (A) ← (E3–E0)		<b>Grouping:</b>	Register to register transfer		
			<b>Description:</b>	Transfers the high-order 4 bits (E7–E4) of register E to register B, and low-order 4 bits of register E to register A.		

### TABP p (Transfer data to Accumulator and register B from Program memory in page p)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0	0 1 0 0 p <sub>4</sub> p <sub>3</sub> p <sub>2</sub> p <sub>1</sub> p <sub>0</sub>	1	3	–	–
<b>Operation:</b>	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← p (PCL) ← (DR2–DR0, A3–A0) (B) ← (ROM(PC)) <sub>7–4</sub> (A) ← (ROM(PC)) <sub>3–0</sub> (PC) ← (SK(SP)) (SP) ← (SP) – 1		<b>Grouping:</b>	Arithmetic operation		
			<b>Description:</b>	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 (DR2 DR1 DR0 A3 A2 A1 A0) <sub>2</sub> specified by registers A and D in page p. p is 0 to 15 for M34506M2, and p is 0 to 31 for M34506M4/E4.		
			<b>Note:</b>	When this instruction is executed, be careful not to over the stack because 1 stage of stack register is used.		

### TAD (Transfer data to Accumulator from register D)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0	0 0 1 0 1 0 0 0 0 1	1	1	–	–
<b>Operation:</b>	(A2–A0) ← (DR2–DR0) (A3) ← 0		<b>Grouping:</b>	Register to register transfer		
			<b>Description:</b>	Transfers the contents of register D to the low-order 3 bits (A2–A0) of register A.		
			<b>Note:</b>	When this instruction is executed, "0" is stored to the bit 3 (A3) of register A.		

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TADAB (Transfer data to register AD from Accumulator from register B)																			
Instruction code	D9	D0				Number of words	Number of cycles	Flag CY	Skip condition										
	1	0	0	0	1	1	1	0	0	1	2	2	3	9	16	1	1	–	–
<b>Operation:</b> (AD7–AD4) ← (B) (AD3–AD0) ← (A)					<b>Grouping:</b> A/D conversion operation <b>Description:</b> In the A/D conversion mode (Q13 = 0), this instruction is equivalent to the NOP instruction. In the comparator mode (Q13 = 1), transfers the contents of register B to the high-order 4 bits (AD7–AD4) of comparator register, and the contents of register A to the low-order 4 bits (AD3–AD0) of comparator register. (Q13 = bit 3 of A/D control register Q1)														
TAI1 (Transfer data to Accumulator from register I1)																			
Instruction code	D9	D0				Number of words	Number of cycles	Flag CY	Skip condition										
	1	0	0	1	0	1	0	0	1	1	2	2	5	3	16	1	1	–	–
<b>Operation:</b> (A) ← (I1)					<b>Grouping:</b> Interrupt operation <b>Description:</b> Transfers the contents of interrupt control register I1 to register A.														
TAK0 (Transfer data to Accumulator from register K0)																			
Instruction code	D9	D0				Number of words	Number of cycles	Flag CY	Skip condition										
	1	0	0	1	0	1	0	1	1	0	2	2	5	6	16	1	1	–	–
<b>Operation:</b> (A) ← (K0)					<b>Grouping:</b> Input/Output operation <b>Description:</b> Transfers the contents of key-on wakeup control register K0 to register A.														
TAK1 (Transfer data to Accumulator from register K1)																			
Instruction code	D9	D0				Number of words	Number of cycles	Flag CY	Skip condition										
	1	0	0	1	0	1	1	0	0	1	2	2	5	9	16	1	1	–	–
<b>Operation:</b> (A) ← (K1)					<b>Grouping:</b> Input/Output operation <b>Description:</b> Transfers the contents of key-on wakeup control register K1 to register A.														

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### TAK2 (Transfer data to Accumulator from register K2)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 0 1 0 1 1 0 1 0	1	1	–	–
<b>Operation:</b>	(A) ← (K2)		<b>Grouping:</b>	Input/Output operation		
			<b>Description:</b>	Transfers the contents of key-on wakeup control register K2 to register A.		

### TALA (Transfer data to Accumulator from register LA)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 0 1 0 0 1 0 0 1	1	1	–	–
<b>Operation:</b>	(A3, A2) ← (AD1, AD0) (A1, A0) ← 0		<b>Grouping:</b>	A/D conversion operation		
			<b>Description:</b>	Transfers the low-order 2 bits (AD1, AD0) of register AD to the high-order 2 bits (A3, A2) of register A.		
			<b>Note:</b>	After this instruction is executed, "0" is stored to the low-order 2 bits (A1, A0) of register A.		

### TAM j (Transfer data to Accumulator from Memory)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 1 1 0 0 j j j j	1	1	–	–
<b>Operation:</b>	(A) ← (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15		<b>Grouping:</b>	RAM to register transfer		
			<b>Description:</b>	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.		

### TAMR (Transfer data to Accumulator from register MR)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 0 1 0 1 0 0 1 0	1	1	–	–
<b>Operation:</b>	(A) ← (MR)		<b>Grouping:</b>	Other operation		
			<b>Description:</b>	Transfers the contents of clock control register MR to register A.		

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### TAQ1 (Transfer data to Accumulator from register Q1)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 0 1 0 0 0 1 0 0	1	1	–	–
<b>Operation:</b>	(A) ← (Q1)		<b>Grouping:</b> A/D conversion operation			
			<b>Description:</b> Transfers the contents of A/D control register Q1 to register A.			

### TASP (Transfer data to Accumulator from Stack Pointer)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0	0 0 1 0 1 0 0 0 0	1	1	–	–
<b>Operation:</b>	(A <sub>2</sub> –A <sub>0</sub> ) ← (SP <sub>2</sub> –SP <sub>0</sub> ) (A <sub>3</sub> ) ← 0		<b>Grouping:</b> Register to register transfer			
			<b>Description:</b> Transfers the contents of stack pointer (SP) to the low-order 3 bits (A <sub>2</sub> –A <sub>0</sub> ) of register A.			
			<b>Note:</b> After this instruction is executed, “0” is stored to the bit 3 (A <sub>3</sub> ) of register A.			

### TAV1 (Transfer data to Accumulator from register V1)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0	0 0 1 0 1 0 1 0 0	1	1	–	–
<b>Operation:</b>	(A) ← (V1)		<b>Grouping:</b> Interrupt operation			
			<b>Description:</b> Transfers the contents of interrupt control register V1 to register A.			

### TAV2 (Transfer data to Accumulator from register V2)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0	0 0 1 0 1 0 1 0 1	1	1	–	–
<b>Operation:</b>	(A) ← (V2)		<b>Grouping:</b> Interrupt operation			
			<b>Description:</b> Transfers the contents of interrupt control register V2 to register A.			

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### TAW1 (Transfer data to Accumulator from register W1)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 0 1 0 0 1 0 1 1	1	1	–	–
<b>Operation:</b>	(A) ← (W1)		<b>Grouping:</b> Timer operation			
			<b>Description:</b> Transfers the contents of timer control register W1 to register A.			

### TAW2 (Transfer data to Accumulator from register W2)

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

### TAW6 (Transfer data to Accumulator from register W6)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1	0 0 1 0 1 0 0 0 0	1	1	–	–
<b>Operation:</b>	(A) ← (W6)		<b>Grouping:</b> Timer operation			
			<b>Description:</b> Transfers the contents of timer control register W6 to register A.			

### TAX (Transfer data to Accumulator from register X)

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



## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### TAY (Transfer data to Accumulator from register Y)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	0	0	0	0	0	1	1	1	1	1	2	0	1	F	16	1	1	–	–
<b>Operation:</b>	(A) ← (Y)											<b>Grouping:</b>	Register to register transfer						
												<b>Description:</b>	Transfers the contents of register Y to register A.						

### TAZ (Transfer data to Accumulator from register Z)

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

### TBA (Transfer data to register B from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	0	0	0	0	0	0	1	1	1	0	2	0	0	E	16	1	1	–	–
<b>Operation:</b>	(B) ← (A)											<b>Grouping:</b>	Register to register transfer						
												<b>Description:</b>	Transfers the contents of register A to register B.						

### TDA (Transfer data to register D from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	0	0	0	0	1	0	1	0	0	1	2	0	2	9	16	1	1	–	–
<b>Operation:</b>	(DR2–DR0) ← (A2–A0)											<b>Grouping:</b>	Register to register transfer						
												<b>Description:</b>	Transfers the contents of the low-order 3 bits (A2–A0) of register A to register D.						

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### TEAB (Transfer data to register E from Accumulator and register B)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	0	0	0	0	0	1	1	0	1	0	2	0	1	A	16	1	1	–	–
<b>Operation:</b>	(E7–E4) ← (B)											<b>Grouping:</b>		Register to register transfer					
	(E3–E0) ← (A)											<b>Description:</b>		Transfers the contents of register B to the high-order 4 bits (E3–E0) of register E, and the contents of register A to the low-order 4 bits (E3–E0) of register E.					

### TI1A (Transfer data to register I1 from Accumulator)

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

### TK0A (Transfer data to register K0 from Accumulator)

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

### TK1A (Transfer data to register K1 from Accumulator)

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

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### TK2A (Transfer data to register K2 from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
1 0 0 0 0 1 0 1 0 1		2 1 5	1	1	–	–
<b>Operation:</b> (K2) ← (A)			<b>Grouping:</b> Input/Output operation			
			<b>Description:</b> Transfers the contents of register A to key-on wakeup control register K2.			

### TMA j (Transfer data to Memory from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
1 0 1 0 1 1 j j j j		2 B j	1	1	–	–
<b>Operation:</b> (M(DP)) ← (A) (X) ← (X)EXOR(j) j = 0 to 15			<b>Grouping:</b> RAM to register transfer			
			<b>Description:</b> After transferring the contents of register A to M(DP), an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.			

### TMRA (Transfer data to register MR from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
1 0 0 0 0 1 0 1 1 0		2 1 6	1	1	–	–
<b>Operation:</b> (MR) ← (A)			<b>Grouping:</b> Other operation			
			<b>Description:</b> Transfers the contents of register A to clock control register MR.			

### TPU0A (Transfer data to register PU0 from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
1 0 0 0 1 0 1 1 0 1		2 2 D	1	1	–	–
<b>Operation:</b> (PU0) ← (A)			<b>Grouping:</b> Input/Output operation			
			<b>Description:</b> Transfers the contents of register A to pull-up control register PU0.			

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### TPU1A (Transfer data to register PU1 from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	1	0	0	0	1	0	1	1	1	0	2	2	2	E	16	1	1	–	–
<b>Operation:</b>	(PU1) ← (A)											<b>Grouping:</b>	Input/Output operation						
												<b>Description:</b>	Transfers the contents of register A to pull-up control register PU1.						

### TPU2A (Transfer data to register PU2 from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition														
	1	0	0	0	1	0	1	1	1	1	1	2	2	2	F	16	1	1	–	–
<b>Operation:</b>	(PU2) ← (A)											<b>Grouping:</b>	Input/Output operation							
												<b>Description:</b>	Transfers the contents of register A to pull-up control register PU2.							

### TQ1A (Transfer data to register Q1 from Accumulator)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	1	0	0	0	0	0	0	1	0	0	2	2	0	4	16	1	1	–	–
<b>Operation:</b>	(Q1) ← (A)											<b>Grouping:</b>	A/D conversion operation						
												<b>Description:</b>	Transfers the contents of register A to A/D control register Q1.						

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition														
	1	0	0	0	1	1	1	1	1	1	1	2	2	3	F	16	1	1	–	–
<b>Operation:</b>	(R17–R14) ← (B) (R13–R10) ← (A)											<b>Grouping:</b>	Timer operation							
												<b>Description:</b>	Transfers the contents of register B to the high-order 4 bits (R17–R14) of reload register R1, and the contents of register A to the low-order 4 bits (R13–R10) of reload register R1.							

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### TV1A (Transfer data to register V1 from Accumulator)

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

### TV2A (Transfer data to register V2 from Accumulator)

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

### TW1A (Transfer data to register W1 from Accumulator)

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

### TW2A (Transfer data to register W2 from Accumulator)

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

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### TW6A (Transfer data to register W6 from Accumulator)

Instruction code	D9	D0								Number of words	Number of cycles	Flag CY	Skip condition					
	1	0	0	0	0	1	0	0	1	1	2	1	3	16	1	1	-	-
<b>Operation:</b>	(W6) ← (A)																	
<b>Grouping:</b>	Timer operation																	
<b>Description:</b>	Transfers the contents of register A to timer control register W6.																	

### TYA (Transfer data to register Y from Accumulator)

Instruction code	D9	D0								Number of words	Number of cycles	Flag CY	Skip condition						
	0	0	0	0	0	0	1	1	0	0	2	0	0	C	16	1	1	-	-
<b>Operation:</b>	(Y) ← (A)																		
<b>Grouping:</b>	Register to register transfer																		
<b>Description:</b>	Transfers the contents of register A to register Y.																		

### WRST (Watchdog timer ReSeT)

Instruction code	D9	D0								Number of words	Number of cycles	Flag CY	Skip condition						
	1	0	1	0	1	0	0	0	0	0	2	2	A	0	16	1	1	-	(WDF1) = 1
<b>Operation:</b>	(WDF1) = 1 ? After skipping, (WDF1) ← 0																		
<b>Grouping:</b>	Other operation																		
<b>Description:</b>	Skips the next instruction when watchdog timer flag WDF1 is "1." After skipping, clears (0) to the WDF1 flag. When the WDF1 flag is "0," executes the next instruction. Also, stops the watchdog timer function when executing the WRST instruction immediately after the DWDT instruction.																		

### XAM j (eXchange Accumulator and Memory data)

Instruction code	D9	D0								Number of words	Number of cycles	Flag CY	Skip condition						
	1	0	1	1	0	1	j	j	j	j	2	2	D	j	16	1	1	-	-
<b>Operation:</b>	(A) ↔ (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15																		
<b>Grouping:</b>	RAM to register transfer																		
<b>Description:</b>	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.																		

## MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

### XAMD j (eXchange Accumulator and Memory data and Decrement register Y and skip)

Instruction code	D9	D0				Number of words	Number of cycles	Flag CY	Skip condition									
	1	0	1	1	1	1	j	j	j	j	2	F	j	16	1	1	-	(Y) = 15
<b>Operation:</b>	$(A) \leftrightarrow (M(DP))$ $(X) \leftarrow (X) \text{EXOR}(j)$ $j = 0 \text{ to } 15$ $(Y) \leftarrow (Y) - 1$														<b>Grouping:</b>	RAM to register transfer		
															<b>Description:</b>	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped. When the contents of register Y is not 15, the next instruction is executed.		

### XAMI j (eXchange Accumulator and Memory data and Increment register Y and skip)

Instruction code	D9	D0				Number of words	Number of cycles	Flag CY	Skip condition									
	1	0	1	1	1	0	j	j	j	j	2	E	j	16	1	1	-	(Y) = 0
<b>Operation:</b>	$(A) \leftrightarrow (M(DP))$ $(X) \leftarrow (X) \text{EXOR}(j)$ $j = 0 \text{ to } 15$ $(Y) \leftarrow (Y) + 1$														<b>Grouping:</b>	RAM to register transfer		
															<b>Description:</b>	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. when the contents of register Y is not 0, the next instruction is executed.		

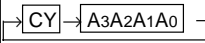
## MACHINE INSTRUCTIONS (INDEX BY TYPES)

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



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

**MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)**

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

Note : p is 0 to 15 for M34506M2, p is 0 to 31 for M34506M4/E4.



## MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

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

Note : p is 0 to 15 for M34506M2, p is 0 to 31 for M34506M4/E4.

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

## MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation			
Interrupt operation	DI	0	0	0	0	0	0	0	1	0	0	0 0 4	1	1	(INTE) ← 0
	EI	0	0	0	0	0	0	0	1	0	1	0 0 5	1	1	(INTE) ← 1
	SNZ0	0	0	0	0	1	1	1	0	0	0	0 3 8	1	1	V10 = 0: (EXF0) = 1 ? After skipping, (EXF0) ← 0 V10 = 1: SNZ0 = NOP
	SNZI0	0	0	0	0	1	1	1	0	1	0	0 3 A	1	1	I12 = 0: (INT) = "L" ?  I12 = 1: (INT) = "H" ?
	TAV1	0	0	0	1	0	1	0	1	0	0	0 5 4	1	1	(A) ← (V1)
	TV1A	0	0	0	0	1	1	1	1	1	1	0 3 F	1	1	(V1) ← (A)
	TAV2	0	0	0	1	0	1	0	1	0	1	0 5 5	1	1	(A) ← (V2)
	TV2A	0	0	0	0	1	1	1	1	1	0	0 3 E	1	1	(V2) ← (A)
	TAI1	1	0	0	1	0	1	0	0	1	1	2 5 3	1	1	(A) ← (I1)
	TI1A	1	0	0	0	0	1	0	1	1	1	2 1 7	1	1	(I1) ← (A)
Timer operation	TAW1	1	0	0	1	0	0	1	0	1	1	2 4 B	1	1	(A) ← (W1)
	TW1A	1	0	0	0	0	0	1	1	1	0	2 0 E	1	1	(W1) ← (A)
	TAW2	1	0	0	1	0	0	1	1	0	0	2 4 C	1	1	(A) ← (W2)
	TW2A	1	0	0	0	0	0	1	1	1	1	2 0 F	1	1	(W2) ← (A)
	TAW6	1	0	0	1	0	1	0	0	0	0	2 5 0	1	1	(A) ← (W6)
	TW6A	1	0	0	0	0	1	0	0	1	1	2 1 3	1	1	(W6) ← (A)
	TAB1	1	0	0	1	1	1	0	0	0	0	2 7 0	1	1	(B) ← (T17–T14) (A) ← (T13–T10)
	T1AB	1	0	0	0	1	1	0	0	0	0	2 3 0	1	1	(T17–T14) ← (B) (R17–R14) ← (B) (T13–T10) ← (A) (R13–R10) ← (A)
	TAB2	1	0	0	1	1	1	0	0	0	1	2 7 1	1	1	(B) ← (T27–T24) (A) ← (T23–T20)
	T2AB	1	0	0	0	1	1	0	0	0	1	2 3 1	1	1	(T27–T24) ← (B) (R27–R24) ← (B) (T23–T20) ← (A) (R23–R20) ← (A)
	TR1AB	1	0	0	0	1	1	1	1	1	1	2 3 F	1	1	(R17–R14) ← (B) (R13–R10) ← (A)
	SNZT1	1	0	1	0	0	0	0	0	0	0	2 8 0	1	1	V12 = 0: (T1F) = 1 ? After skipping, (T1F) ← 0 V12 = 1: SNZT1 = NOP
	SNZT2	1	0	1	0	0	0	0	0	0	1	2 8 1	1	1	V13 = 0: (T2F) = 1 ? After skipping, (T2F) ← 0 V13 = 1: SNZT2 = NOP

Skip condition	Carry flag CY	Detailed description
<ul style="list-style-type: none"> <li>-</li> <li>-</li> <li>V10 = 0: (EXF0) = 1</li> <li>(INT) = "L" However, I12 = 0</li> <li>(INT) = "H" However, I12 = 1</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> </ul>	<ul style="list-style-type: none"> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> </ul>	<ul style="list-style-type: none"> <li>- Clears (0) to interrupt enable flag INTE, and disables the interrupt.</li> <li>- Sets (1) to interrupt enable flag INTE, and enables the interrupt.</li> <li>- When V10 = 0 : Skips the next instruction when external 0 interrupt request flag EXF0 is "1." After skipping, clears (0) to the EXF0 flag. When the EXF0 flag is "0," executes the next instruction. When V10 = 1 : This instruction is equivalent to the NOP instruction. (V10: bit 0 of interrupt control register V1)</li> <li>- When I12 = 0 : Skips the next instruction when the level of INT pin is "L." Executes the next instruction when the level of INT pin is "H."</li> <li>- When I12 = 1 : Skips the next instruction when the level of INT pin is "H." Executes the next instruction when the level of INT pin is "L." (I12: bit 2 of interrupt control register I1)</li> <li>- Transfers the contents of interrupt control register V1 to register A.</li> <li>- Transfers the contents of register A to interrupt control register V1.</li> <li>- Transfers the contents of interrupt control register V2 to register A.</li> <li>- Transfers the contents of register A to interrupt control register V2.</li> <li>- Transfers the contents of interrupt control register I1 to register A.</li> <li>- Transfers the contents of register A to interrupt control register I1.</li> </ul>
<ul style="list-style-type: none"> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>V12 = 0: (T1F) = 1</li> <li>V13 = 0: (T2F) = 1</li> </ul>	<ul style="list-style-type: none"> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> <li>-</li> </ul>	<ul style="list-style-type: none"> <li>- Transfers the contents of timer control register W1 to register A.</li> <li>- Transfers the contents of register A to timer control register W1.</li> <li>- Transfers the contents of timer control register W2 to register A.</li> <li>- Transfers the contents of register A to timer control register W2.</li> <li>- Transfers the contents of timer control register W6 to register A.</li> <li>- Transfers the contents of register A to timer control register W6.</li> <li>- Transfers the high-order 4 bits (T17–T14) of timer 1 to register B. Transfers the low-order 4 bits (T13–T10) of timer 1 to register A.</li> <li>- Transfers the contents of register B to the high-order 4 bits of timer 1 and timer 1 reload register R1. Transfers the contents of register A to the low-order 4 bits of timer 1 and timer 1 reload register R1.</li> <li>- Transfers the high-order 4 bits (T27–T24) of timer 2 to register B. Transfers the low-order 4 bits (T23–T20) of timer 2 to register A.</li> <li>- Transfers the contents of register B to the high-order 4 bits of timer 2 and timer 2 reload register R2. Transfers the contents of register A to the low-order 4 bits of timer 2 and timer 2 reload register R2.</li> <li>- Transfers the contents of register B to the high-order 4 bits (R17–R14) of reload register R1, and the contents of register A to the low-order 4 bits (R13–R10) of reload register R1.</li> <li>- When V12 = 0 : Skips the next instruction when timer 1 interrupt request flag T1F is "1." After skipping, clears (0) to the T1F flag. When the T1F flag is "0," executes the next instruction. When V12 = 1 : This instruction is equivalent to the NOP instruction. (V12: bit 2 of interrupt control register V1)</li> <li>- When V13 = 0 : Skips the next instruction when timer 1 interrupt request flag T2F is "1." After skipping, clears (0) to the T2F flag. When the T2F flag is "0," executes the next instruction. When V13 = 1 : This instruction is equivalent to the NOP instruction. (V13: bit 3 of interrupt control register V1)</li> </ul>

## MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation			
Input/Output operation	IAP0	1	0	0	1	1	0	0	0	0	0	2 6 0	1	1	(A) ← (P0)
	OP0A	1	0	0	0	1	0	0	0	0	0	2 2 0	1	1	(P0) ← (A)
	IAP1	1	0	0	1	1	0	0	0	0	1	2 6 1	1	1	(A) ← (P1)
	OP1A	1	0	0	0	1	0	0	0	0	1	2 2 1	1	1	(P1) ← (A)
	IAP2	1	0	0	1	1	0	0	0	1	0	2 6 2	1	1	(A <sub>1</sub> , A <sub>0</sub> ) ← (P <sub>21</sub> , P <sub>20</sub> ) (A <sub>3</sub> , A <sub>2</sub> ) ← 0
	OP2A	1	0	0	0	1	0	0	0	1	0	2 2 2	1	1	(P <sub>21</sub> , P <sub>20</sub> ) ← (A <sub>1</sub> , A <sub>0</sub> )
	CLD	0	0	0	0	0	1	0	0	0	1	0 1 1	1	1	(D) ← 1
	RD	0	0	0	0	0	1	0	1	0	0	0 1 4	1	1	(D(Y)) ← 0 (Y) = 0 to 3
	SD	0	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	0	1	0	0	1	0	0	0 2 4	2	2	(D(Y)) = 0 ? (Y) = 0 to 3
		0	0	0	0	1	0	1	0	1	1	0 2 B			
	SCP	1	0	1	0	0	0	1	1	0	1	2 8 D	1	1	(C) ← 1
	RCP	1	0	1	0	0	0	1	1	0	0	2 8 C	1	1	(C) ← 0
	SNZCP	1	0	1	0	0	0	1	0	0	1	2 8 9	1	1	(C) = 1?
	IAK	1	0	0	1	1	0	1	1	1	1	2 6 F	1	1	(A <sub>0</sub> ) ← (K) (A <sub>3</sub> –A <sub>1</sub> ) ← 0
	OKA	1	0	0	0	0	1	1	1	1	1	2 1 F	1	1	(K) ← (A <sub>0</sub> )
	TK0A	1	0	0	0	0	1	1	0	1	1	2 1 B	1	1	(K <sub>0</sub> ) ← (A)
	TAK0	1	0	0	1	0	1	0	1	1	0	2 5 6	1	1	(A) ← (K <sub>0</sub> )
	TK1A	1	0	0	0	0	1	0	1	0	0	2 1 4	1	1	(K <sub>1</sub> ) ← (A)
	TAK1	1	0	0	1	0	1	1	0	0	1	2 5 9	1	1	(A) ← (K <sub>1</sub> )
	TK2A	1	0	0	0	0	1	0	1	0	1	2 1 5	1	1	(K <sub>2</sub> ) ← (A)
	TAK2	1	0	0	1	0	1	1	0	1	0	2 5 A	1	1	(A) ← (K <sub>2</sub> )
	TPU0A	1	0	0	0	1	0	1	1	0	1	2 2 D	1	1	(PU <sub>0</sub> ) ← (A)
	TPU1A	1	0	0	0	1	0	1	1	1	0	2 2 E	1	1	(PU <sub>1</sub> ) ← (A)
TPU2A	1	0	0	0	1	0	1	1	1	1	2 2 F	1	1	(PU <sub>2</sub> ) ← (A)	



Skip condition	Carry flag CY	Detailed description
-	-	Transfers the input of port P0 to register A.
-	-	Outputs the contents of register A to port P0.
-	-	Transfers the input of port P1 to register A.
-	-	Outputs the contents of register A to port P1.
-	-	Transfers the input of port P2 to the low-order 2 bits (A1, A0) of register A.
-	-	Outputs the contents of the low-order 2 bits (A1, A0) of register A to port P2.
-	-	Sets (1) to port D.
-	-	Clears (0) to a bit of port D specified by register Y.
-	-	Sets (1) to a bit of port D specified by register Y.
(D(Y)) = 0 ? (Y) = 0 to 3	-	Skips the next instruction when a bit of port D specified by register Y is "0." Executes the next instruction when a bit of port D specified by register Y is "1."
-	-	Sets (1) to port C.
-	-	Clears (0) to port C.
(C) = 1	-	Skips the next instruction when the contents of port C is "1." Executes the next instruction when the contents of port C is "0."
-	-	Transfers the contents of port K to the bit 0 (A0) of register A.
-	-	Outputs the contents of bit 0 (A0) of register A to port K.
-	-	Transfers the contents of register A to key-on wakeup control register K0.
-	-	Transfers the contents of key-on wakeup control register K0 to register A.
-	-	Transfers the contents of register A to key-on wakeup control register K1.
-	-	Transfers the contents of key-on wakeup control register K1 to register A.
-	-	Transfers the contents of register A to key-on wakeup control register K2.
-	-	Transfers the contents of key-on wakeup control register K2 to register A.
-	-	Transfers the contents of register A to pull-up control register PU0.
-	-	Transfers the contents of register A to pull-up control register PU1.
-	-	Transfers the contents of register A to pull-up control register PU2.

## MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Parameter Type of instructions	Mnemonic	Instruction code										Hexadecimal notation	Number of words	Number of cycles	Function
		D9	D8	D7	D6	D5	D4	D3	D2	D1	D0				
A/D conversion operation	TABAD	1	0	0	1	1	1	1	0	0	1	2 7 9	1	1	In A/D conversion mode (Q13 = 0), (B) ← (AD9–AD6) (A) ← (AD5–AD2) In comparator mode (Q13 = 1), (B) ← (AD7–AD4) (A) ← (AD3–AD0)
	TALA	1	0	0	1	0	0	1	0	0	1	2 4 9	1	1	(A3, A2) ← (AD1, AD0) (A1, A0) ← 0
	TADAB	1	0	0	0	1	1	1	0	0	1	2 3 9	1	1	(AD7–AD4) ← (B) (AD3–AD0) ← (A)
	TAQ1	1	0	0	1	0	0	0	1	0	0	2 4 4	1	1	(A) ← (Q1)
	TQ1A	1	0	0	0	0	0	0	1	0	0	2 0 4	1	1	(Q1) ← (A)
	ADST	1	0	1	0	0	1	1	1	1	1	2 9 F	1	1	(ADF) ← 0 Q13 = 0: A/D conversion starting Q13 = 1: Comparator operation starting
	SNZAD	1	0	1	0	0	0	0	1	1	1	2 8 7	1	1	V22 = 0: (ADF) = 1 ? After skipping, (ADF) ← 0 V22 = 1: SNZAD = NOP
Other operation	NOP	0	0	0	0	0	0	0	0	0	0	0 0 0	1	1	(PC) ← (PC) + 1
	POF2	0	0	0	0	0	0	1	0	0	0	0 0 8	1	1	RAM back-up
	EPOF	0	0	0	1	0	1	1	0	1	1	0 5 B	1	1	POF2 instruction valid
	SNZP	0	0	0	0	0	0	0	0	1	1	0 0 3	1	1	(P) = 1 ?
	DWDT	1	0	1	0	0	1	1	1	0	0	2 9 C	1	1	Stop of watchdog timer function enabled
	WRST	1	0	1	0	1	0	0	0	0	0	2 A 0	1	1	(WDF1) = 1 ?, after skipping, (WDF1) ← 0
	CMCK	1	0	1	0	0	1	1	0	1	0	2 9 A	1	1	Ceramic resonator selected
	CRCK	1	0	1	0	0	1	1	0	1	1	2 9 B	1	1	RC oscillation selected
	TAMR	1	0	0	1	0	1	0	0	1	0	2 5 2	1	1	(A) ← (MR)
	TMRA	1	0	0	0	0	1	0	1	1	0	2 1 6	1	1	(MR) ← (A)

Skip condition	Carry flag CY	Detailed description
-	-	In the A/D conversion mode (Q13 = 0), transfers the high-order 4 bits (AD9–AD6) of register AD to register B, and the middle-order 4 bits (AD5–AD2) of register AD to register A. In the comparator mode (Q13 = 1), transfers the high-order 4 bits (AD7–AD4) of comparator register to register B, and the low-order 4 bits (AD3–AD0) of comparator register to register A. (Q13: bit 3 of A/D control register Q1)
-	-	Transfers the low-order 2 bits (AD1, AD0) of register AD to the high-order 2 bits (AD3, AD2) of register A.
-	-	In the A/D conversion mode (Q13 = 0), this instruction is equivalent to the NOP instruction. In the comparator mode (Q13 = 1), transfers the contents of register B to the high-order 4 bits (AD7–AD4) of comparator register, and the contents of register A to the low-order 4 bits (AD3–AD0) of comparator register. (Q13 = bit 3 of A/D control register Q1)
-	-	Transfers the contents of A/D control register Q1 to register A.
-	-	Transfers the contents of register A to A/D control register Q1.
-	-	Clears (0) to A/D conversion completion flag ADF, and the A/D conversion at the A/D conversion mode (Q13 = 0) or the comparator operation at the comparator mode (Q13 = 1) is started. (Q13 = bit 3 of A/D control register Q1)
V22 = 0: (ADF) = 1	-	When V22 = 0 : Skips the next instruction when A/D conversion completion flag ADF is "1." After skipping, clears (0) to the ADF flag. When the ADF flag is "0," executes the next instruction. When V22 = 1 : This instruction is equivalent to the NOP instruction. (V22: bit 2 of interrupt control register V2)
-	-	No operation; Adds 1 to program counter value, and others remain unchanged.
-	-	Puts the system in RAM back-up state by executing the POF2 instruction after executing the EPOF instruction. Operations of all functions are stopped.
-	-	Makes the immediate after POF2 instruction valid by executing the EPOF instruction.
(P) = 1	-	Skips the next instruction when the P flag is "1". After skipping, the P flag remains unchanged. Executes the next instruction when the P flag is "0."
-	-	Stops the watchdog timer function by the WRST instruction after executing the DWDT instruction.
(WDF1) = 1	-	Skips the next instruction when watchdog timer flag WDF1 is "1." After skipping, clears (0) to the WDF1 flag. When the WDF1 flag is "0," executes the next instruction. Also, stops the watchdog timer function when executing the WRST instruction immediately after the DWDT instruction.
-	-	Selects the ceramic resonance circuit and stops the on-chip oscillator.
-	-	Selects the RC oscillation circuit and stops the on-chip oscillator.
-	-	Transfers the contents of clock control register MR to register A.
-	-	Transfers the contents of register A to clock control register MR.

**INSTRUCTION CODE TABLE**

D3-D0	D9-D4 Hex. notation	000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111	010000	011000
		00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10-17	18-1F
0000	0	NOP	BLA	SZB 0	BMLA	-	TASP	A 0	LA 0	TABP 0	TABP 16*	-	-	BML	BML*	BL	BL*	BM	B
0001	1	-	CLD	SZB 1	-	-	TAD	A 1	LA 1	TABP 1	TABP 17*	-	-	BML	BML*	BL	BL*	BM	B
0010	2	-	-	SZB 2	-	-	TAX	A 2	LA 2	TABP 2	TABP 18*	-	-	BML	BML*	BL	BL*	BM	B
0011	3	SNZP	INY	SZB 3	-	-	TAZ	A 3	LA 3	TABP 3	TABP 19*	-	-	BML	BML*	BL	BL*	BM	B
0100	4	DI	RD	SZD	-	RT	TAV1	A 4	LA 4	TABP 4	TABP 20*	-	-	BML	BML*	BL	BL*	BM	B
0101	5	EI	SD	SEAn	-	RTS	TAV2	A 5	LA 5	TABP 5	TABP 21*	-	-	BML	BML*	BL	BL*	BM	B
0110	6	RC	-	SEAM	-	RTI	-	A 6	LA 6	TABP 6	TABP 22*	-	-	BML	BML*	BL	BL*	BM	B
0111	7	SC	DEY	-	-	-	-	A 7	LA 7	TABP 7	TABP 23*	-	-	BML	BML*	BL	BL*	BM	B
1000	8	POF2	AND	-	SNZ0	LZ 0	-	A 8	LA 8	TABP 8	TABP 24*	-	-	BML	BML*	BL	BL*	BM	B
1001	9	-	OR	TDA	-	LZ 1	-	A 9	LA 9	TABP 9	TABP 25*	-	-	BML	BML*	BL	BL*	BM	B
1010	A	AM	TEAB	TABE	SNZ10	LZ 2	-	A 10	LA 10	TABP 10	TABP 26*	-	-	BML	BML*	BL	BL*	BM	B
1011	B	AMC	-	-	-	LZ 3	EPOF	A 11	LA 11	TABP 11	TABP 27*	-	-	BML	BML*	BL	BL*	BM	B
1100	C	TYA	CMA	-	-	RB 0	SB 0	A 12	LA 12	TABP 12	TABP 28*	-	-	BML	BML*	BL	BL*	BM	B
1101	D	-	RAR	-	-	RB 1	SB 1	A 13	LA 13	TABP 13	TABP 29*	-	-	BML	BML*	BL	BL*	BM	B
1110	E	TBA	TAB	-	TV2A	RB 2	SB 2	A 14	LA 14	TABP 14	TABP 30*	-	-	BML	BML*	BL	BL*	BM	B
1111	F	-	TAY	SZC	TV1A	RB 3	SB 3	A 15	LA 15	TABP 15	TABP 31*	-	-	BML	BML*	BL	BL*	BM	B

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

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

	The second word
BL	10 0aaa aaaa
BML	10 0aaa aaaa
BLA	10 0p00 pppp
BMLA	10 0p00 pppp
SEA	00 0111 nnnn
SZD	00 0010 1011

• \* cannot be used in the M34506M2-XXXFP.

**INSTRUCTION CODE TABLE (continued)**

D3-D0	D9-D4						Hex. notation														
	100000	100001	100010	100011	100100	100101	100110	100111	101000	101001	101010	101011	101100	101101	101110	101111	110000	111111			
	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30-3F				
0000	0	-	-	OP0A	T1AB	-	TAW6	IAP0	TAB1	SNZT1	-	WRST	TMA0	TAM0	XAM0	XAMI0	XAMD0	LXY			
0001	1	-	-	OP1A	T2AB	-	-	IAP1	TAB2	SNZT2	-	-	TMA1	TAM1	XAM1	XAMI1	XAMD1	LXY			
0010	2	-	-	OP2A	-	-	TAMR	IAP2	-	-	-	-	TMA2	TAM2	XAM2	XAMI2	XAMD2	LXY			
0011	3	-	TW6A	-	-	-	TAI1	-	-	-	-	-	TMA3	TAM3	XAM3	XAMI3	XAMD3	LXY			
0100	4	TQ1A	TK1A	-	-	TAQ1	-	-	-	-	-	-	TMA4	TAM4	XAM4	XAMI4	XAMD4	LXY			
0101	5	-	TK2A	-	-	-	-	-	-	-	-	-	TMA5	TAM5	XAM5	XAMI5	XAMD5	LXY			
0110	6	-	TMRA	-	-	-	TAK0	-	-	-	-	-	TMA6	TAM6	XAM6	XAMI6	XAMD6	LXY			
0111	7	-	TI1A	-	-	-	-	-	-	SNZAD	-	-	TMA7	TAM7	XAM7	XAMI7	XAMD7	LXY			
1000	8	-	-	-	-	-	-	-	-	-	-	-	TMA8	TAM8	XAM8	XAMI8	XAMD8	LXY			
1001	9	-	-	-	TADAB	TALA	TAK1	-	TABAD	SNZCP	-	-	TMA9	TAM9	XAM9	XAMI9	XAMD9	LXY			
1010	A	-	-	-	-	-	TAK2	-	-	-	CMCK	-	TMA10	TAM10	XAM10	XAMI10	XAMD10	LXY			
1011	B	-	TK0A	-	-	TAW1	-	-	-	-	CRCK	-	TMA11	TAM11	XAM11	XAMI11	XAMD11	LXY			
1100	C	-	-	-	-	TAW2	-	-	-	RCP	DWDT	-	TMA12	TAM12	XAM12	XAMI12	XAMD12	LXY			
1101	D	-	-	TPU0A	-	-	-	-	-	SCP	-	-	TMA13	TAM13	XAM13	XAMI13	XAMD13	LXY			
1110	E	TW1A	-	TPU1A	-	-	-	-	-	-	-	-	TMA14	TAM14	XAM14	XAMI14	XAMD14	LXY			
1111	F	TW2A	OKA	TPU2A	TR1AB	-	-	IAK	-	-	ADST	-	TMA15	TAM15	XAM15	XAMI15	XAMD15	LXY			

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

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

	The second word		
BL	10	0aaa	aaaa
BML	10	0aaa	aaaa
BLA	10	0p00	pppp
BMLA	10	0p00	pppp
SEA	00	0111	nnnn
SZD	00	0010	1011

## Electrical characteristics

### Absolute maximum ratings

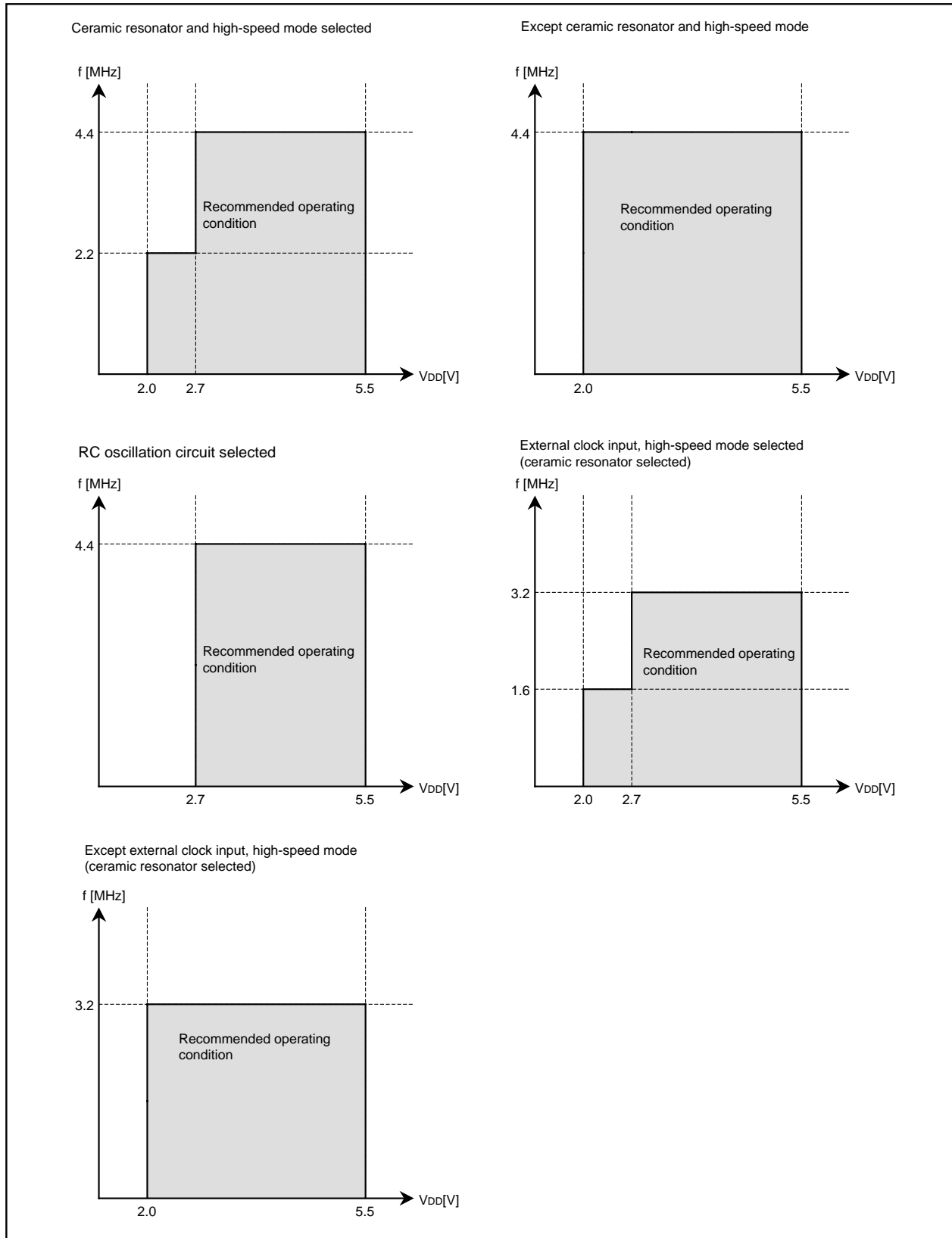
Symbol	Parameter	Conditions	Ratings	Unit
V <sub>DD</sub>	Supply voltage		-0.3 to 6.5	V
V <sub>I</sub>	Input voltage P0, P1, P2, D0, D1, D2/C, D3/K, RESET, X <sub>IN</sub>		-0.3 to V <sub>DD</sub> +0.3	V
V <sub>I</sub>	Input voltage AIN <sub>0</sub> –AIN <sub>1</sub>		-0.3 to V <sub>DD</sub> +0.3	V
V <sub>O</sub>	Output voltage P0, P1, P2, D0, D1, D2/C, D3/K, RESET	Output transistors in cut-off state	-0.3 to V <sub>DD</sub> +0.3	V
V <sub>O</sub>	Output voltage X <sub>OUT</sub>		-0.3 to V <sub>DD</sub> +0.3	V
P <sub>d</sub>	Power dissipation	T <sub>a</sub> = 25 °C	300	mW
T <sub>opr</sub>	Operating temperature range		-20 to 85	°C
T <sub>stg</sub>	Storage temperature range		-40 to 125	°C

**Recommended operating conditions 1**

(Ta = -20 °C to 85 °C, VDD = 2.0 to 5.5 V, unless otherwise noted)

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
VDD	Supply voltage (with a ceramic resonator)	High-speed mode	f(XIN) ≤ 4.4 MHz	2.7		5.5	V
		Middle-speed mode	f(XIN) ≤ 4.4 MHz	2.0		5.5	
		Low-speed mode					
		Default mode					
VDD	Supply voltage (with RC oscillation)	High-speed mode	f(XIN) ≤ 4.4 MHz	2.7		5.5	V
		Middle-speed mode					
		Low-speed mode					
		Default mode					
V <sub>RAM</sub>	RAM back-up voltage	(at RAM back-up)		1.8			V
V <sub>SS</sub>	Supply voltage				0		V
V <sub>IH</sub>	"H" level input voltage	P0, P1, P2, D0–D3, XIN		0.8V <sub>DD</sub>		V <sub>DD</sub>	V
V <sub>IH</sub>	"H" level input voltage	RESET		0.85V <sub>DD</sub>		V <sub>DD</sub>	V
V <sub>IH</sub>	"H" level input voltage	C, K	V <sub>DD</sub> = 4.0 to 5.5 V	0.5V <sub>DD</sub>		V <sub>DD</sub>	V
			V <sub>DD</sub> = 2.0 to 5.5 V	0.7V <sub>DD</sub>		V <sub>DD</sub>	
V <sub>IH</sub>	"H" level input voltage	CNTR, INT		0.85V <sub>DD</sub>		V <sub>DD</sub>	V
V <sub>IL</sub>	"L" level input voltage	P0, P1, P2, D0–D3, XIN		0		0.2V <sub>DD</sub>	V
V <sub>IL</sub>	"L" level input voltage	C, K		0		0.16V <sub>DD</sub>	V
V <sub>IL</sub>	"L" level input voltage	RESET		0		0.3V <sub>DD</sub>	V
V <sub>IL</sub>	"L" level input voltage	CNTR, INT		0		0.15V <sub>DD</sub>	V
I <sub>OL(peak)</sub>	"L" level peak output current	P2, RESET	V <sub>DD</sub> = 5.0 V			10	mA
			V <sub>DD</sub> = 3.0 V			4.0	
I <sub>OL(peak)</sub>	"L" level peak output current	D0, D1	V <sub>DD</sub> = 5.0 V			40	mA
			V <sub>DD</sub> = 3.0 V			30	
I <sub>OL(peak)</sub>	"L" level peak output current	D2/C, D3/K	V <sub>DD</sub> = 5.0 V			24	mA
			V <sub>DD</sub> = 3.0 V			12	
I <sub>OL(peak)</sub>	"L" level peak output current	P0, P1	V <sub>DD</sub> = 5.0 V			24	mA
			V <sub>DD</sub> = 3.0 V			12	
I <sub>OL(avg)</sub>	"L" level average output current	P2, RESET (Note)	V <sub>DD</sub> = 5.0 V			5.0	mA
			V <sub>DD</sub> = 3.0 V			2.0	
I <sub>OL(avg)</sub>	"L" level average output current	D0, D1 (Note)	V <sub>DD</sub> = 5.0 V			30	mA
			V <sub>DD</sub> = 3.0 V			15	
I <sub>OL(avg)</sub>	"L" level average output current	D2/C, D3/K (Note)	V <sub>DD</sub> = 5.0 V			15	mA
			V <sub>DD</sub> = 3.0 V			7.0	
I <sub>OL(avg)</sub>	"L" level average output current	P0, P1 (Note)	V <sub>DD</sub> = 5.0 V			12	mA
			V <sub>DD</sub> = 3.0 V			6.0	
ΣI <sub>OL(avg)</sub>	"L" level total average current	P2, D, RESET				80	mA
		P0, P1				80	

Note : The average output current (I<sub>OH</sub>, I<sub>OL</sub>) is the average value during 100 ms.





**Recommended operating conditions 2**

(Ta = -20 °C to 85 °C, VDD = 2.0 to 5.5 V, unless otherwise noted)

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
f(XIN)	Oscillation frequency (with a ceramic resonator)	High-speed mode	VDD = 2.7 V to 5.5 V			4.4	MHz
			VDD = 2.0 V to 5.5 V			2.2	
		Middle-speed mode	VDD = 2.0 V to 5.5 V			4.4	
		Low-speed mode Default mode					
f(XIN)	Oscillation frequency (with RC oscillation) (Note)	High-speed mode Middle-speed mode Low-speed mode Default mode	VDD = 2.7 V to 5.5 V			4.4	MHz
f(XIN)	Oscillation frequency (with a ceramic resonator selected, external clock input)	High-speed mode	VDD = 2.7 V to 5.5 V			3.2	MHz
			VDD = 2.0 V to 5.5 V			1.6	
		Middle-speed mode	VDD = 2.0 V to 5.5 V			3.2	
		Low-speed mode Default mode					
$\Delta f(XIN)$	Oscillation frequency error (at RC oscillation, error value of external R, C not included) Note: use 30 pF capacitor and vary external R	VDD = 5.0 V $\pm$ 10 %, Ta = 25 °C, -20 to 85 °C				$\pm$ 17	%
		VDD = 3.0 V $\pm$ 10 %, Ta = 25 °C, -20 to 85 °C				$\pm$ 17	
f(CNTR)	Timer external input frequency	High-speed mode				f(XIN)/6	Hz
		Middle-speed mode				f(XIN)/12	
		Low-speed mode				f(XIN)/24	
		Default mode				f(XIN)/48	
tw(CNTR)	Timer external input period ("H" and "L" pulse width)	High-speed mode		3/f(XIN)			s
		Middle-speed mode		6/f(XIN)			
		Low-speed mode		12/f(XIN)			
		Default mode		24/f(XIN)			

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

**Electrical characteristics** (Ta = -20 °C to 85 °C, VDD = 2.0 to 5.5 V, unless otherwise noted)

Symbol	Parameter		Test conditions		Limits			Unit		
					Min.	Typ.	Max.			
VOL	"L" level output voltage P0, P1		VDD = 5.0 V	IOL = 12 mA			2.0	V		
				IOL = 4.0 mA			0.9			
			VDD = 3.0 V	IOL = 6.0 mA			0.9			
				IOL = 2.0 mA			0.6			
VOL	"L" level output voltage P2, RESET		VDD = 5.0 V	IOL = 5.0 mA			2.0	V		
				IOL = 1.0 mA			0.6			
			VDD = 3.0 V	IOL = 2.0 mA			0.9			
VOL	"L" level output voltage D0, D1		VDD = 5.0 V	IOL = 30 mA			2.0	V		
				IOL = 10 mA			0.9			
			VDD = 3.0 V	IOL = 15 mA			2.0			
				IOL = 5.0 mA			0.9			
VOL	"L" level output voltage D2/C, D3/K		VDD = 5.0 V	IOL = 15 mA			2.0	V		
				IOL = 5.0 mA			0.9			
			VDD = 3.0 V	IOL = 9.0 mA			2.0			
				IOL = 3.0 mA			0.9			
I <sub>IH</sub>	"H" level input current P0, P1, P2, RESET		VI = VDD				1.0	μA		
I <sub>IH</sub>	"H" level input current D0, D1, D2/C, D3/K		VI = VDD				1.0	μA		
I <sub>IL</sub>	"L" level input current P0, P1, P2		VI = 0 V P0, P1, P2 No pull-up		-1.0			μA		
I <sub>IL</sub>	"L" level input current D0, D1, D2/C, D3/K		VI = 0 V, D2/C, D3/K, No pull-up		-1.0			μA		
IDD	Supply current	at active mode (Note 1)	VDD = 5.0 V f(XIN) = 4.0 MHz	High-speed mode		1.7	5.0	mA		
				Middle-speed mode		1.3	3.9			
				Low-speed mode		1.1	3.3			
				Default mode		1.0	3.0			
			VDD = 3.0 V f(XIN) = 2.0 MHz	High-speed mode		0.5	1.5			
				Middle-speed mode		0.4	1.2			
				Low-speed mode		0.35	1.1			
				Default mode		0.3	0.9			
		at RAM back-up mode (POF2 instruction execution)	Ta = 25 °C				0.1		1.0	μA
			VDD = 5.0 V						10	
		VDD = 3.0 V				6.0				
RPU	Pull-up resistor value P0, P1, P2, D2/C, D3/K, RESET		VI = 0 V	VDD = 5.0 V	30	60	150	kΩ		
				VDD = 3.0 V	50	120	300			
VT+ - VT-	Hysteresis INT, CNTR		VDD = 5.0 V			0.25		V		
			VDD = 3.0 V			0.25				
VT+ - VT-	Hysteresis RESET		VDD = 5.0 V			1.2		V		
			VDD = 3.0 V			0.5				
f(RING)	On-chip oscillator clock frequency (Note 2)		VDD = 5.0 V		1.0	2.0	3.0	MHz		
			VDD = 3.0 V		0.5	1.0	1.8			

Notes 1: When the A/D converter is used, the A/D operation current (IADD) is included.

2: When system operates by the on-chip oscillator, the system clock frequency is the on-chip oscillator clock divided by the dividing ratio selected with register MR.

**A/D converter recommended operating conditions**

(Comparator mode included, Ta = -20 °C to 85 °C, unless otherwise noted)

Symbol	Parameter	Conditions	Limits			Unit
			Min.	Typ.	Max.	
VDD	Supply voltage	Ta = 25 °C	2.7		5.5	V
		Ta = -20 °C to 85 °C	3.0		5.5	
VIA	Analog input voltage		0		VDD+2LSB	V
f(XIN)	Oscillation frequency	VDD = 2.7 V to 5.5 V	High-speed mode	0.1		MHz
			Middle-speed mode	0.2		
			Low-speed mode	0.4		
			Default mode	0.8		

**A/D converter characteristics**

(Comparator mode included, Ta = -20 °C to 85 °C, unless otherwise noted)

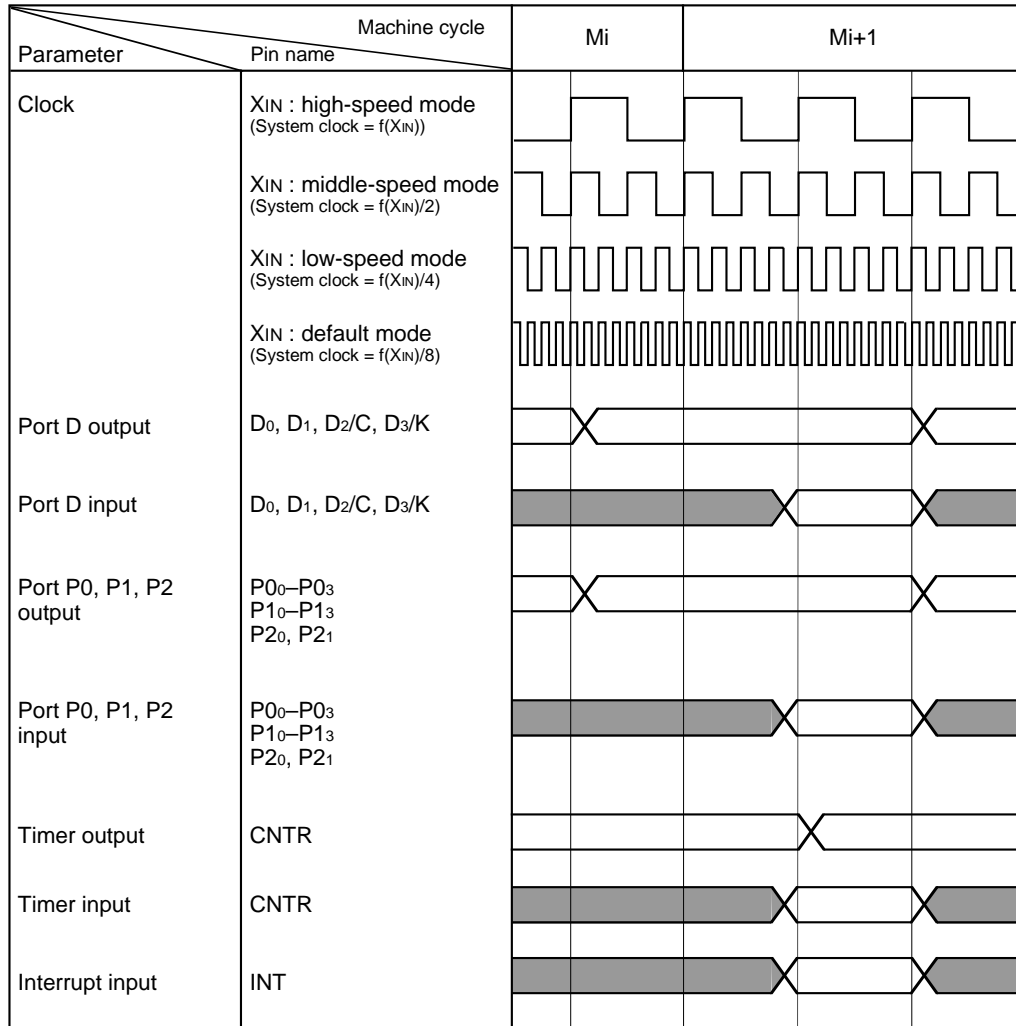
Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
-	Resolution				10	bits
-	Linearity error	Ta = 25 °C, VDD = 2.7 V to 5.5 V			±2.0	LSB
		Ta = -25 °C to 85 °C, VDD = 3.0 V to 5.5 V				
-	Differential non-linearity error	Ta = 25 °C, VDD = 2.7 V to 5.5 V			±0.9	LSB
		Ta = -25 °C to 85 °C, VDD = 3.0 V to 5.5 V				
V0T	Zero transition voltage	VDD = 5.12 V	10	20	30	mV
		VDD = 3.072 V	3	9	15	
VFST	Full-scale transition voltage	VDD = 5.12 V	5115	5125	5135	mV
		VDD = 3.072 V	3063	3069	3075	
IADD	A/D operating current (Note 1)	VDD = 5.0 V		0.3	0.9	mA
		VDD = 3.0 V		0.1	0.3	
TCONV	A/D conversion time	f(XIN) = 4.0 MHz	High-speed mode		46.5	μs
			Middle-speed mode		93.0	
			Low-speed mode		186	
			Default mode		372	
-	Comparator resolution				8	bits
-	Comparator error (Note 2)	VDD = 5.12 V			±20	mV
		VDD = 3.072 V			±15	
-	Comparator comparison time	f(XIN) = 4.0 MHz	High-speed mode		6.0	μs
			Middle-speed mode		12	
			Low-speed mode		24	
			Default mode		48	

Notes 1: When the A/D converter is used, the IADD is included to IDD.

2: As for the error from the logic value in the comparator mode, when the contents of the comparator register is n, the logic value of the comparison voltage Vref which is generated by the built-in DA converter can be obtained by the following formula.

<p>Logic value of comparison voltage Vref</p> $V_{ref} = \frac{V_{DD}}{256} \times n$ <p>n = Value of register AD (n = 0 to 255)</p>
--

**Basic timing diagram**



## BUILT-IN PROM VERSION

In addition to the mask ROM versions, the 4506 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 20 shows the product of built-in PROM version. Figure 54 shows the pin configurations of built-in PROM versions.

The One Time PROM version has pin-compatibility with the mask ROM version.

**Table 20 Product of built-in PROM version**

Part number	PROM size (X 10 bits)	RAM size (X 4 bits)	Package	ROM type
M34506E4FP	4096 words	256 words	PRSP0020DA-A	One Time PROM [shipped in blank]

### (1) PROM mode

The 4506 Group has a PROM mode in addition to a normal operation mode. It has a function to serially input/output the command codes, addresses, and data required for operation (e.g., read and program) on the built-in PROM using only a few pins. This mode can be selected by setting pins SDA (serial data input/output), SCLK (serial clock input), PGM to "H" after connecting wires as shown in Figure 54 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).

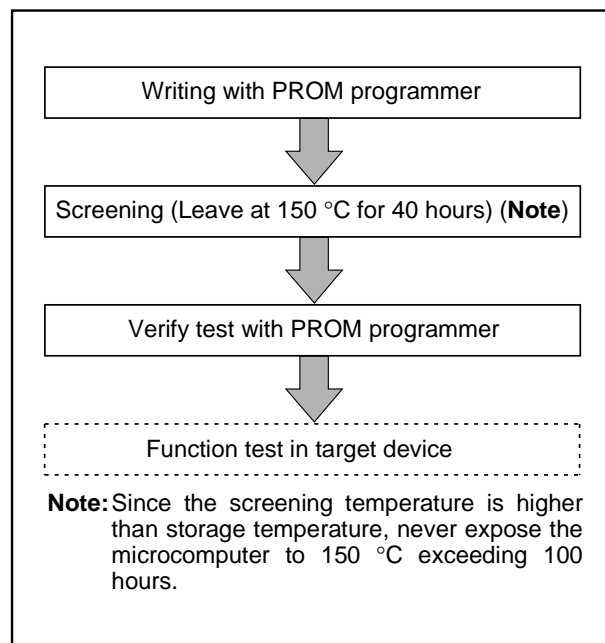
Use the special-purpose serial programmer when performing serial read/program.

As for the serial programmer for the single-chip microcomputer (serial programmer and control software), refer to the "Renesas Microcomputer Development Support Tools" Homepage (<http://www.renesas.com/en/tools>).

### (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, Renesas corp. does not perform PROM writing test and screening in the assembly process and following processes. In order to improve reliability after writing, performing writing and test according to the flow shown in Figure 53 before using is recommended (Products shipped in blank: PROM contents is not written in factory when shipped).



**Fig. 53 Flow of writing and test of the product shipped in blank**

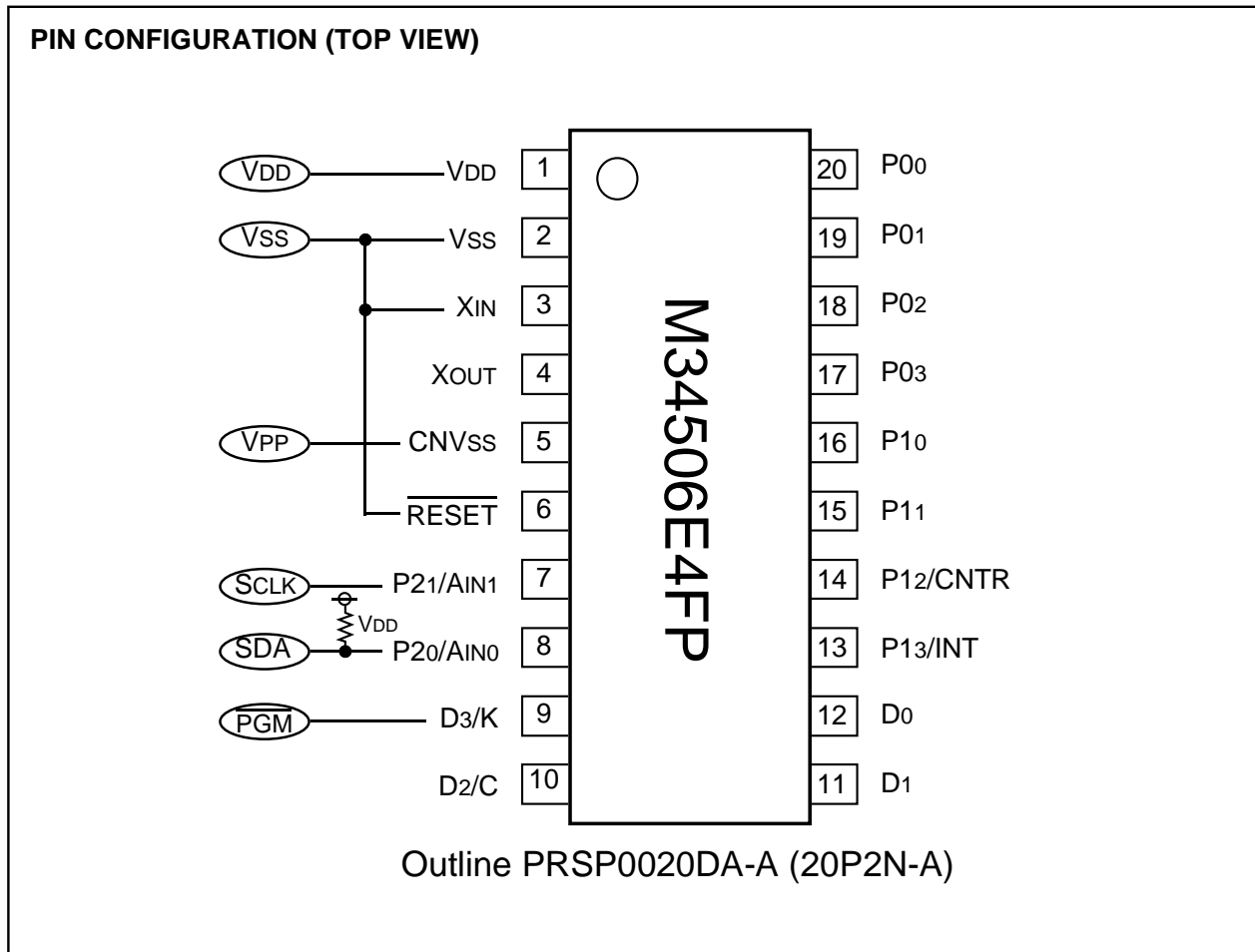
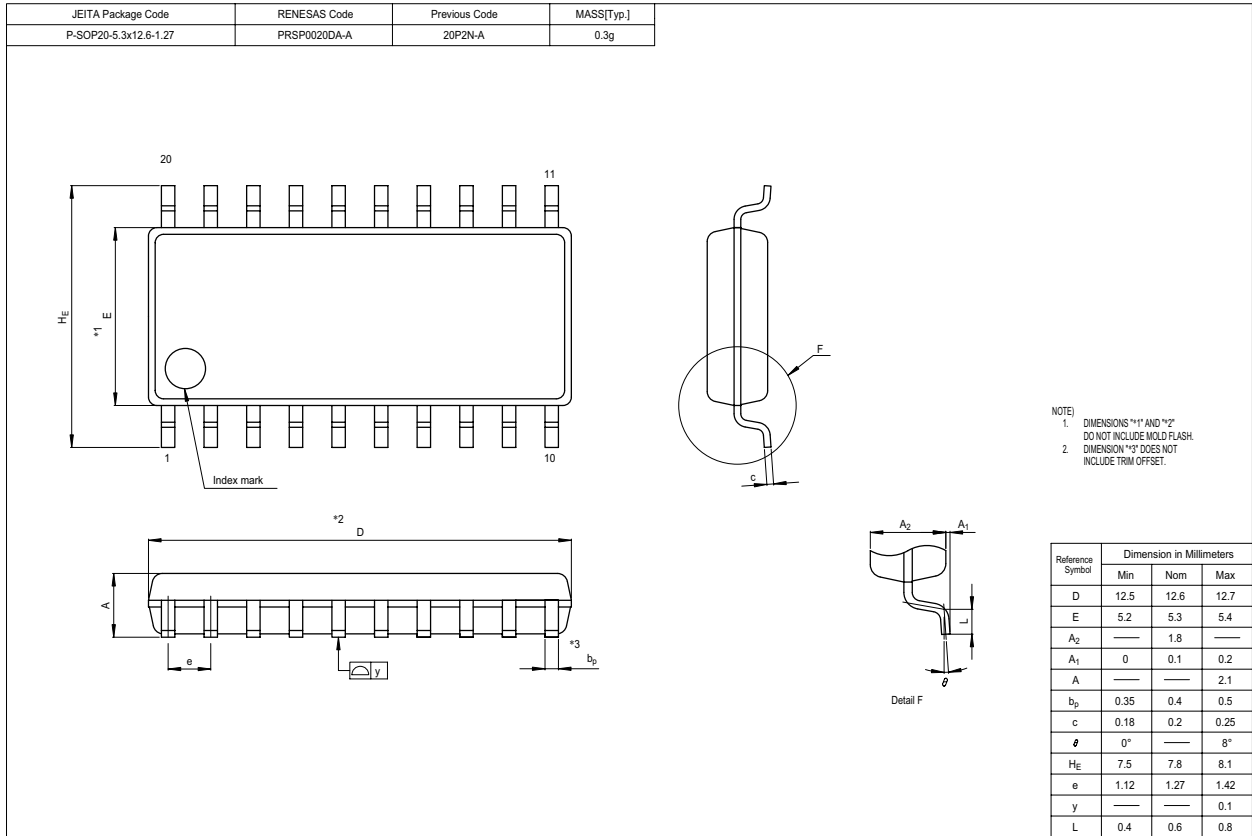


Fig. 54 Pin configuration of built-in PROM version

# Package outline



## REVISION DESCRIPTION LIST

## 4506 GROUP DATA SHEET

Rev. No.	Revision Description	Rev. date
1.0	First Edition	000808
1.1	Pages 3, 4, 22, 38 : Character fonts errors revised	000905
2.0	<p>The 4506/4507 Group data sheet is separated.</p> <p>Page 10: Port block diagram (3); Block diagram of P12/CNTR pin revised.</p> <p>Page 26: Fig. 22 Timers structure; Block diagram of P12/CNTR pin revised.</p> <p>Page 29: (9) Precautions → (8) Precautions (8) Timer input/output pin (P12/CNTR pin) added. Fig. 23 added.</p> <p>Page 30: WATCHDOG TIMER revised all.</p> <p>Page 31: Fig. 24 → Fig. 25, Fig. 25 → Fig. 26 Fig. 26 NOP instruction added. POF → POF<sub>2</sub></p> <p>Page 49: Fig. 46 POF → POF<sub>2</sub></p> <p>Page 61: BL p, a, BLA p instructions revised.</p> <p>Page 62: BML p, a, BMLA p instructions revised.</p> <p>Page 76: TABP p instruction revised.</p> <p>Page 90: TABP p instruction revised.</p> <p>Page 92: BL p, a, BLA p, BML p, a, BMLA p instructions revised.</p> <p>Page 100: BL, BML, BLA, BMLA instructions; The second word revised.</p> <p>Page 101: BL, BML, BLA, BMLA instructions; The second word revised.</p> <p>Page 102: ABSOLUTE MAXIMUM RATINGS; V<sub>DD</sub> -0.3 to 6.0 → -0.3 to 6.5</p> <p>Page 104: RECOMMENDED OPERATING CONDITIONS 1; Operating condition map added.</p>	010531
3.0	<p>All pages: Words standardized: On-chip oscillator, A/D converter</p> <p>Page 3: Power dissipation "Ta=25°C" added.</p> <p>Page 4: Description of RESET pin revised.</p> <p>Page 24: Table 9 : Control register of timer 1 and timer 2 revised.</p> <p>Page 25: Fig.22 : Note 5 added.</p> <p>Page 29: Some description revised.</p> <p>Page 30: Fig.25 : "DI" instruction added.</p> <p>Page 31: Table 11: Revised.</p> <p>Page 39: Table 15 : Port level and Note 4 revised, Note 5 added.</p> <p>Page 50: ①Electric Characteristic Differences Between Mask ROM and One Time PROM Version MCU, ②Note on Power Source Voltage added.</p> <p>Page 76: TABAD : Description revised.</p> <p>Page 99: TABAD : Description revised.</p>	040827



REVISION DESCRIPTION LIST

4506 GROUP DATA SHEET

Rev. No.	Revision Description	Rev. date
3.01	Page 1, 3: Package name revised. Page 28: •Timer 1 and timer 2 count start timing and count time when operation starts added. Page 47: ⑩Timer 1 and timer 2 count start timing and count time when operation starts added. P109, 110: Package name revised. P111: Package outline revised.	050207

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

---

**Keep safety first in your circuit designs!**

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

**Notes regarding these materials**

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



**RENESAS SALES OFFICES**

<http://www.renesas.com>

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

**Renesas Technology America, Inc.**

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

**Renesas Technology Europe Limited**

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

**Renesas Technology Hong Kong Ltd.**

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

**Renesas Technology Taiwan Co., Ltd.**

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

**Renesas Technology (Shanghai) Co., Ltd.**

Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China  
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

**Renesas Technology Singapore Pte. Ltd.**

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