To all our customers

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

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

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

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

RENESAS
Renesas Technology Corp.

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

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 µs
  (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)

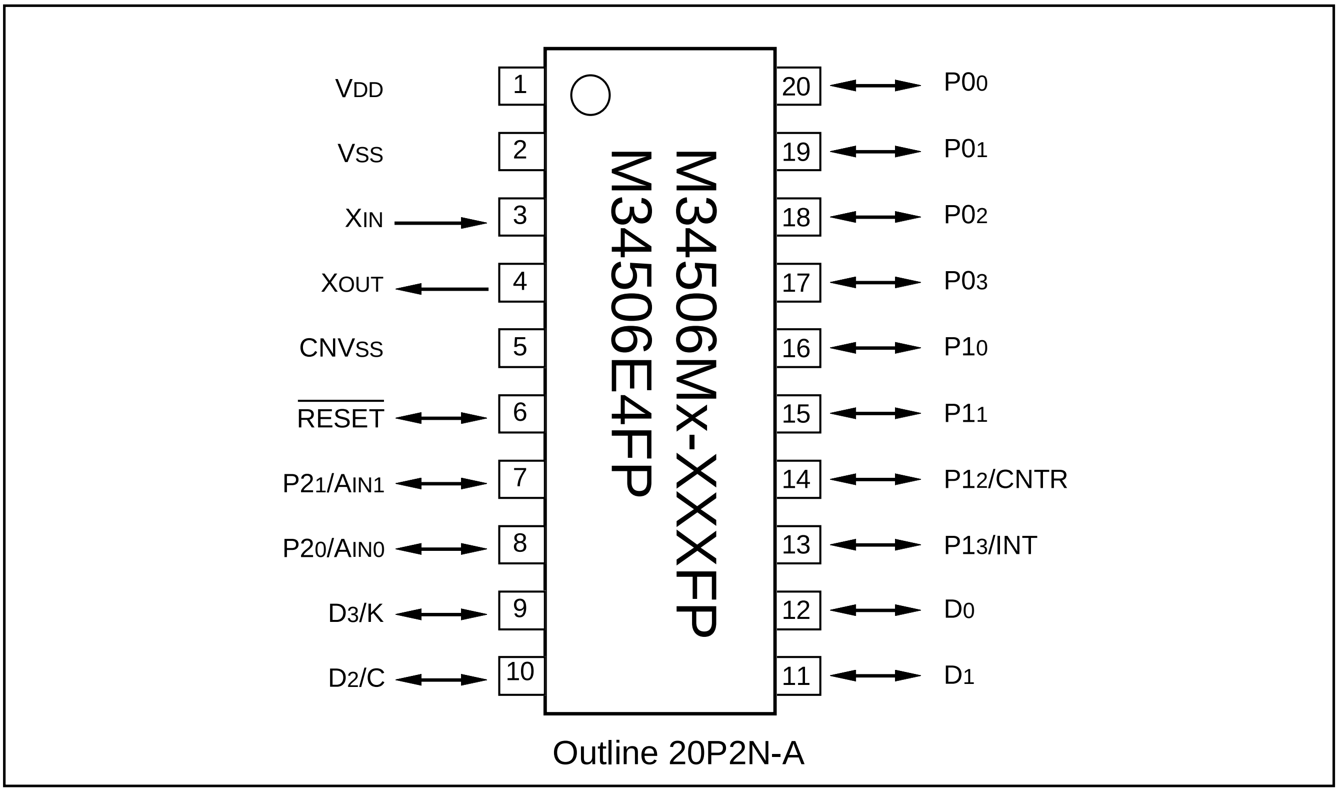
APPLICATION

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

Product	ROM (PROM) size (× 10 bits)	RAM size (× 4 bits)	Package	ROM type
M34506M2-XXXFP	2048 words	128 words	20P2N-A	Mask ROM
M34506M4-XXXFP	4096 words	256 words	20P2N-A	Mask ROM
M34506E4FP (**Note**)	4096 words	256 words	20P2N-A	One Time PROM

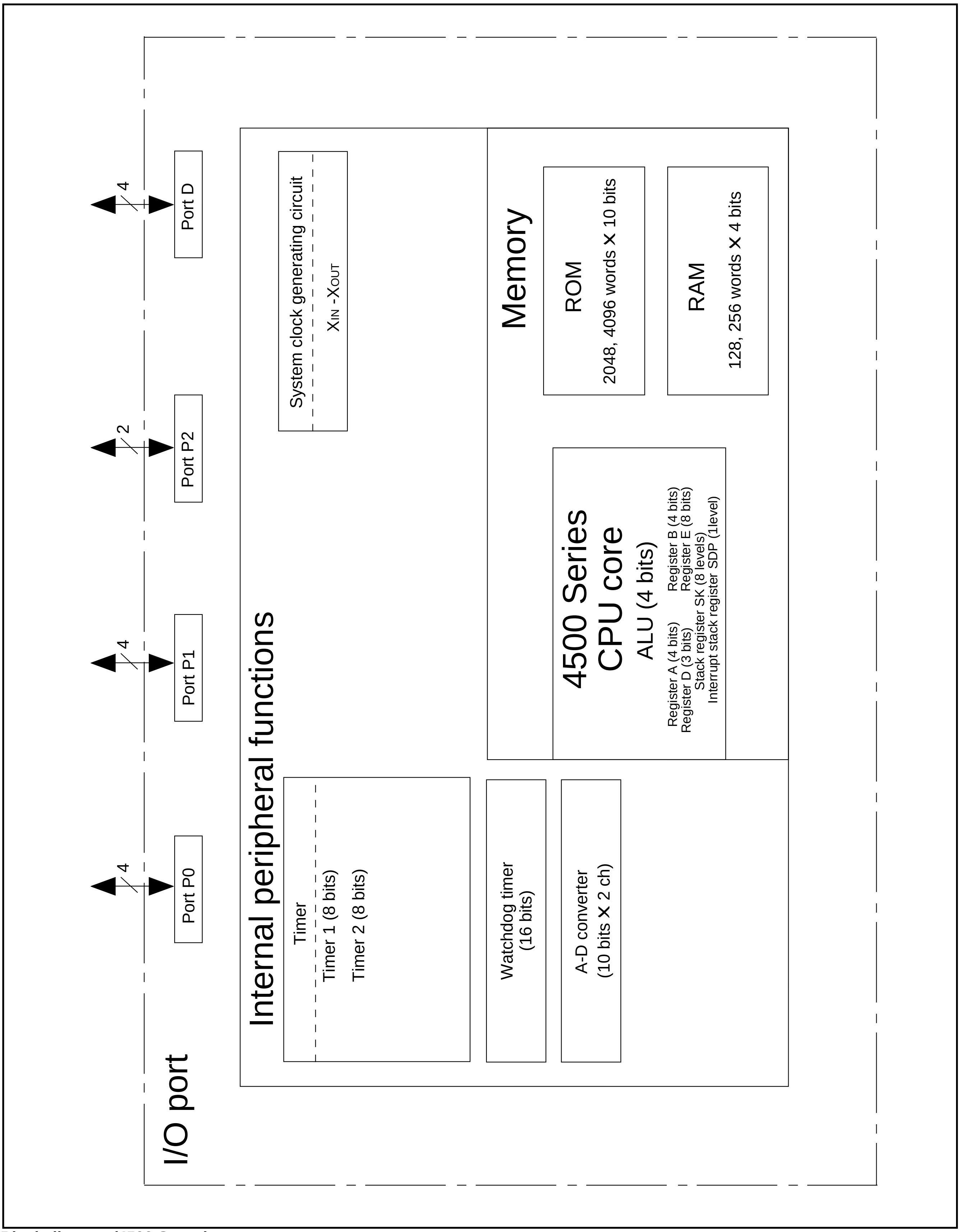
Note: Shipped in blank.

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 μs (at 4.4 MHz oscillation frequency, in high-speed mode)
Memory sizes	ROM	M34506M2	2048 words × 10 bits
		M34506M4/E4	4096 words × 10 bits
	RAM	M34506M2	128 words × 4 bits
		M34506M4/E4	256 words × 4 bits
Input/Output ports	D_0–D_3	I/O	Four independent I/O ports . Input is examined by skip decision. Ports D_2 and D_3 are equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software. Ports D_2 and D_3 are also used as ports C and K, respectively.
	$P0_0$–$P0_3$	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.
	$P1_0$–$P1_3$	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 $P1_2$ and $P1_3$ are also used as CNTR and INT, respectively.
	$P2_0$, $P2_1$	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 $P2_0$ and $P2_1$ are also used as A_{IN0} and A_{IN1}, respectively.
	C	I/O	1-bit I/O; Port C is also used as port D_2.
	K	I/O	1-bit I/O; Port K is also used as port D_3.
	CNTR	Timer I/O	1-bit I/O; CNTR pin is also used as port $P1_2$.
	INT	Interrupt input	1-bit input; INT pin is also used as port $P1_3$.
	A_{IN0}, A_{IN1}	Analog input	Two independent I/O ports; A_{IN0}, A_{IN1} are also used as $P2_0$ and $P2_1$, 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.
	Analog input		2 channel (A_{IN0} pin, A_{IN1} pin)
Interrupt	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 (20P2N-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 (at V_{DD} = 5.0 V, 4.0 MHz oscillation frequency, in high-speed mode, output transistors in the cut-off state)
			0.5 mA (at V_{DD} = 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 μA (at room temperature, V_{DD} = 5 V, output transistors in the cut-off state)

PIN DESCRIPTION

Pin	Name	Input/Output	Function
V_{DD}	Power supply	—	Connected to a plus power supply.
V_{SS}	Ground	—	Connected to a 0 V power supply.
CNV_{SS}	CNV_{SS}	—	Connect CNV_{SS} to V_{SS} and apply “L” (0V) to CNV_{SS} certainly.
RESET	Reset input/output	I/O	An N-channel open-drain I/O pin for a system reset. When the watchdog timer cause the system to be reset, the RESET pin outputs “L” level.
X_{IN}	System clock input	Input	I/O pins of the system clock generating circuit. When using a ceramic resonator, connect it between pins X_{IN} and X_{OUT}. A feedback resistor is built-in between them. When using the RC oscillation, connect a resistor and a capacitor to X_{IN}, and leave X_{OUT} pin open.
X_{OUT}	System clock output	Output	
D_0–D_3	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 D_2 and D_3 are equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software. Ports D_2 and D_3 are also used as ports C and K, respectively.
$P0_0$–$P0_3$	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.
$P1_0$–$P1_3$	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 $P1_2$ and $P1_3$ are also used as CNTR and INT, respectively.
$P2_0$, $P2_1$	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 $P2_0$ and $P2_1$ are also used as A_{IN0} and A_{IN1}, 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 D_2.
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 D_3.
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 $P1_2$.
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 $P1_3$.
A_{IN0}–A_{IN1}	Analog input	Input	A-D converter analog input pins. A_{IN0} and A_{IN1} are also used as ports $P2_0$ and $P2_1$, respectively.

MULTIFUNCTION

Pin	Multifunction	Pin	Multifunction	Pin	Multifunction	Pin	Multifunction
D_2	C	C	D_2	$P2_0$	A_{IN0}	A_{IN0}	$P2_0$
D_3	K	K	D_3	$P2_1$	A_{IN1}	A_{IN1}	$P2_1$
$P1_2$	CNTR	CNTR	$P1_2$				
$P1_3$	INT	INT	$P1_3$				

Notes 1: Pins except above have just single function.
2: The input/output of D_2, D_3, $P1_2$ and $P1_3$ can be used even when C, K, INT and CNTR (input) are selected.
3: The input of $P1_2$ can be used even when CNTR (output) is selected.
4: The input/output of $P2_0$, $P2_1$ can be used even when A_{IN0}, A_{IN1} 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 ring 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.

Table Selection of system clock

Register MR		System clock (Note 1)	Operation mode
MR_3	MR_2		
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(XIN)/4 or f(RING)/4	Low-speed mode
1	1	f(XIN)/8 or f(RING)/8	Default mode

Notes 1: The ring 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.

- 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.

PORT FUNCTION

Port	Pin	Input Output	Output structure	I/O unit	Control instructions	Control registers	Remark
Port D	D_0, D_1	I/O (4)	N-channel open-drain	1	SD, RD SZD, CLD SCP, RCP SNZCP IAK, OKA		
	D_2/C D_3/K					PU2, K2	Built-in programmable pull-up functions Key-on wakeup functions (programmable)
Port P0	$P0_0$–$P0_3$	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	$P1_0$, $P1_1$ $P1_2$/CNTR, $P1_3$/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	$P2_0/A_{IN0}$ $P2_1/A_{IN1}$	I/O (2)	N-channel open-drain	2	OP2A IAP2	PU2, K2 Q1	Built-in programmable pull-up functions Key-on wakeup functions (programmable)

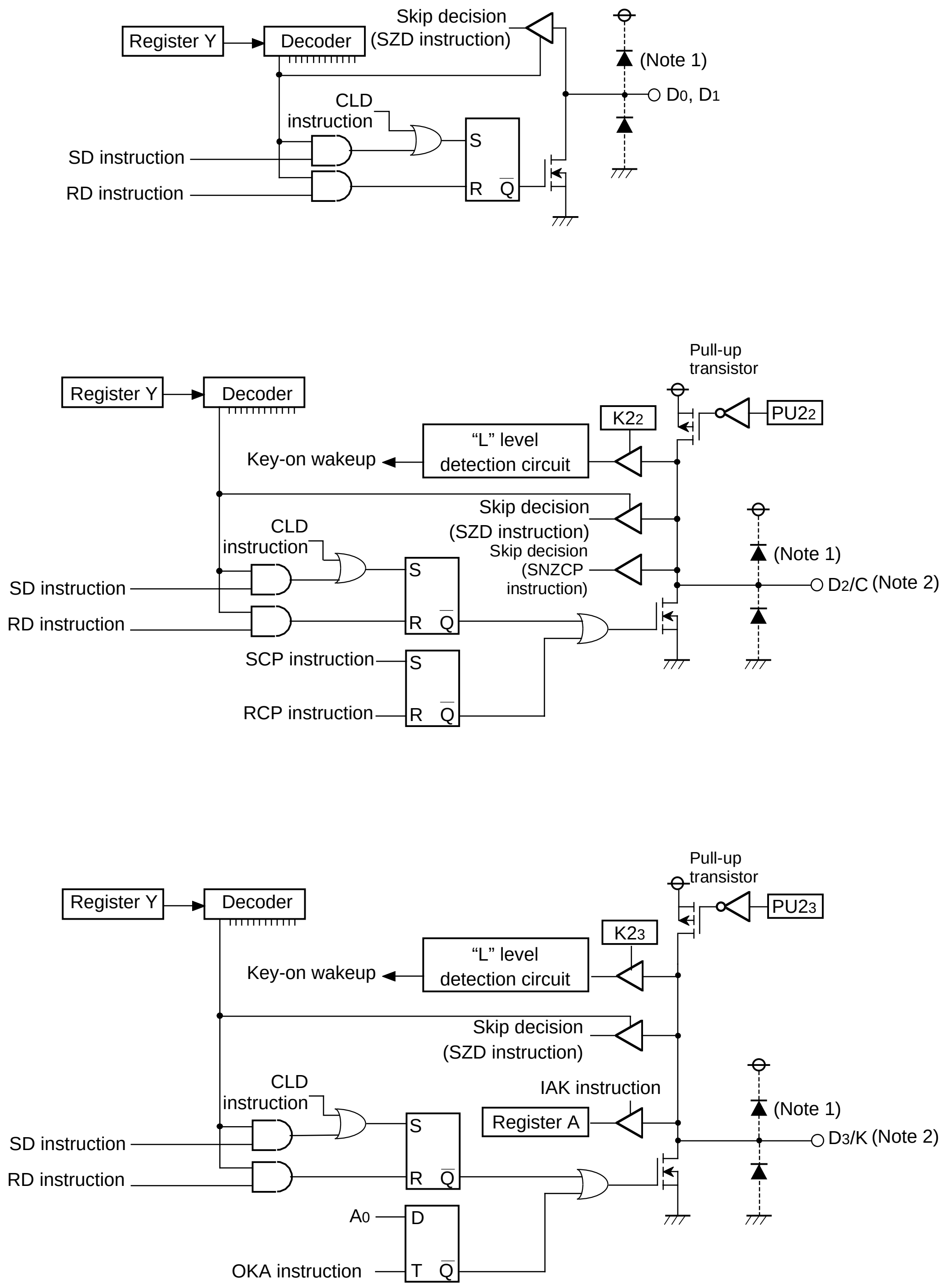
CONNECTIONS OF UNUSED PINS

Pin	Connection	Usage condition
X_{IN}	Connect to V_{SS}.	System operates by the ring oscillator. (Note 1)
X_{OUT}	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 ring oscillator. (Note 1)
D_0, D_1	Open. (Output latch is set to “1.”)	――――
	Open. (Output latch is set to “0.”)	――――
	Connect to V_{SS}.	――――
D_2/C D_3/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 V_{SS}.	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
$P0_0$–$P0_3$	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 V_{SS}.	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
$P1_0$, $P1_1$ $P1_2$/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 V_{SS}.	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
$P1_3$/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 V_{SS}.	The pull-up function and the key-on wakeup function are not selected. (Notes 2, 3)
$P2_0$/A_{IN0} $P2_1$/A_{IN1}	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 V_{SS}.	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 ring 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 ($I1_3$) of register I1 to “0” to disable to input to INT pin (after reset: $I1_3$ = “0”)

(Note when connecting to V_{SS} and V_{DD})
● Connect the unused pins to V_{SS} and V_{DD} using the thickest wire at the shortest distance against noise.

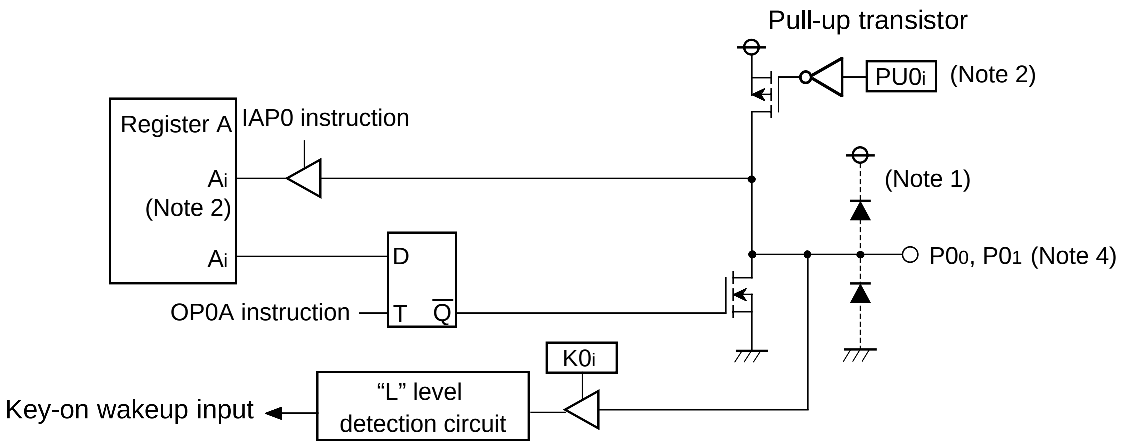
PORT BLOCK DIAGRAMS

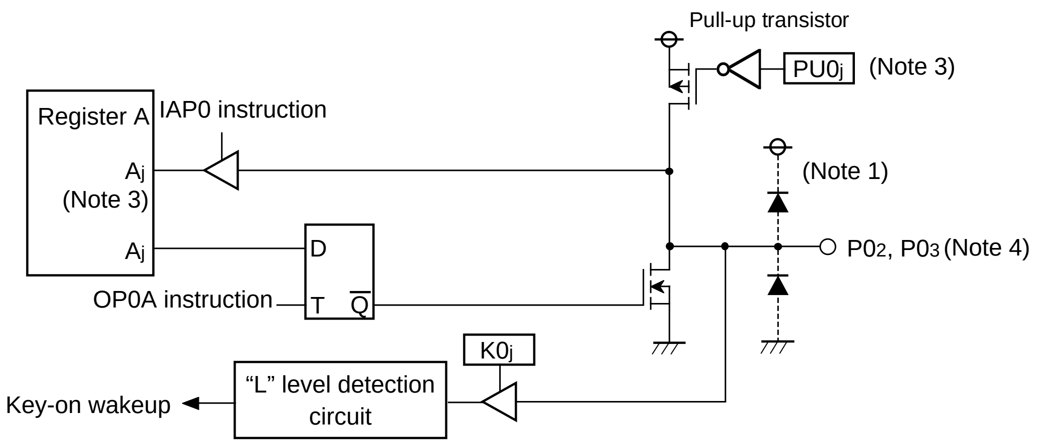


Notes 1: ----|◀---- This symbol represents a parasitic diode on the port.

2: Applied potential to ports D_2/C and D_3/K must be V_{DD} or less.

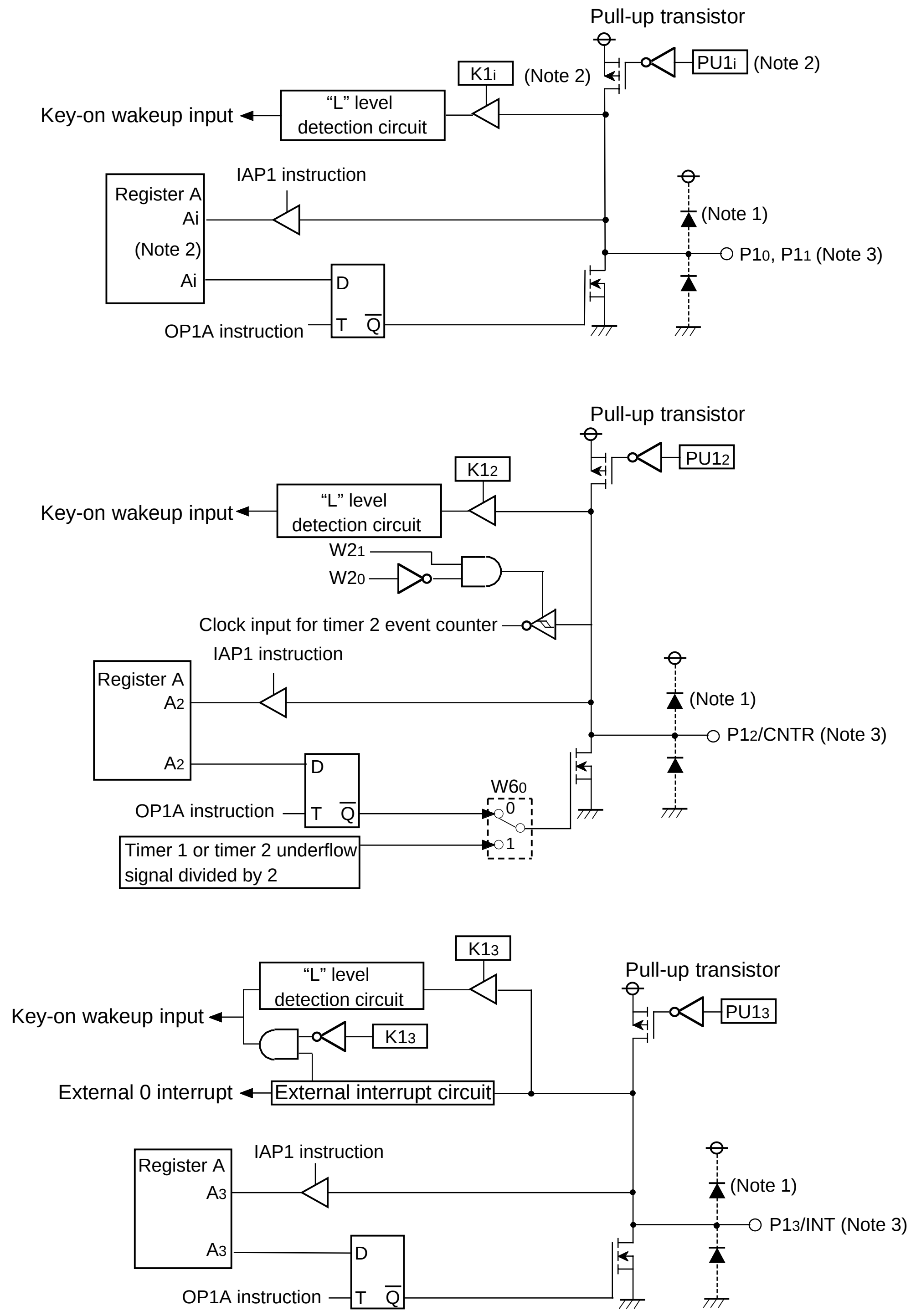
Port block diagram (1)





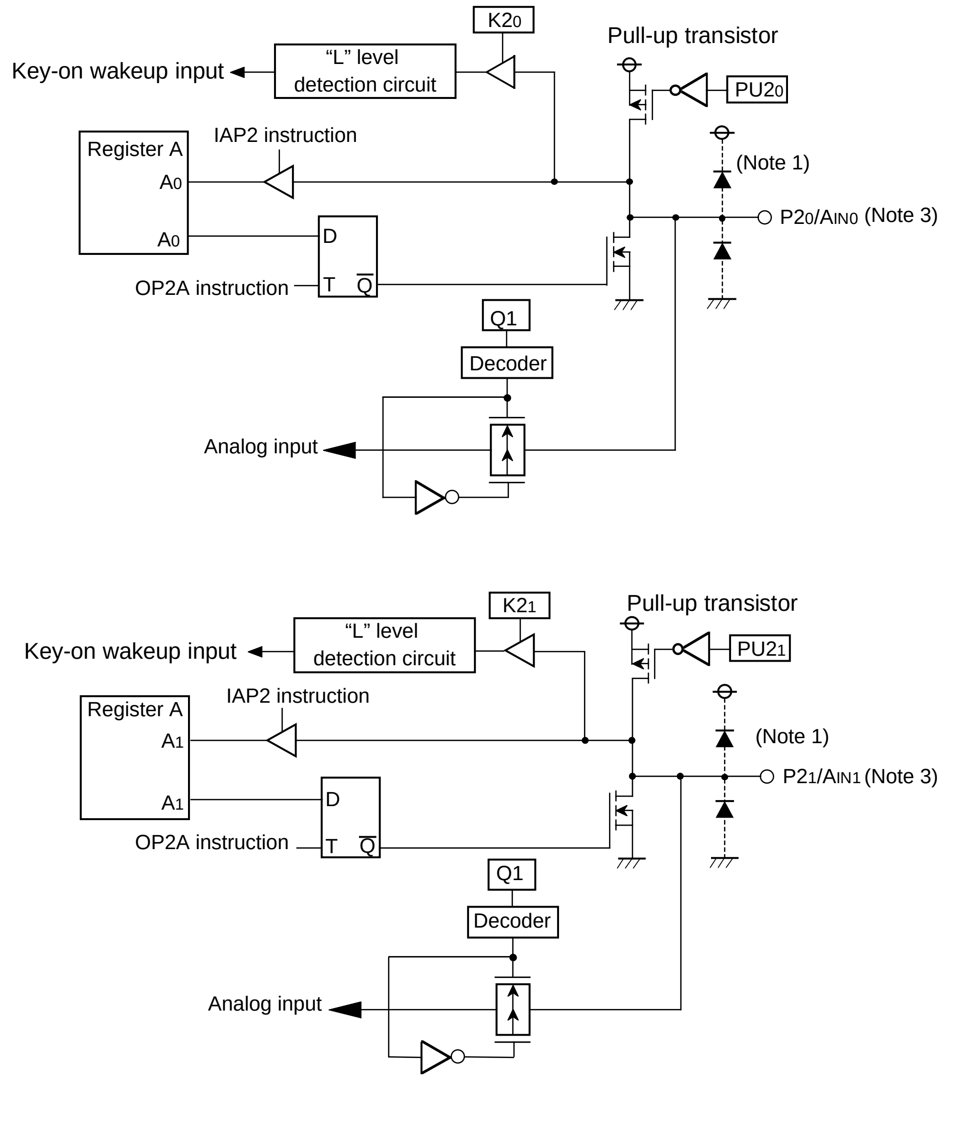
Notes 1: ----|◀---- This symbol represents a parasitic diode on the port.
2: i represents 0 or 1.
3: j represents 2 or 3.
4: Applied potential to port P0 must be V_{DD} or less.

Port block diagram (2)



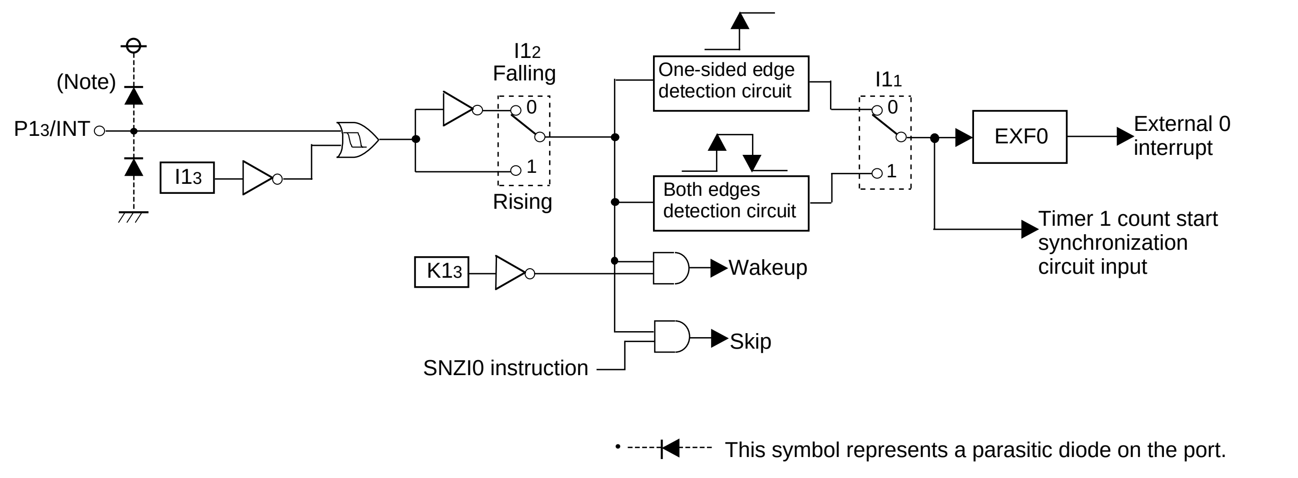
Notes 1: This symbol represents a parasitic diode on the port.
2: i represents 0 or 1.
3: Applied potential to port P1 must be V_{DD} or less.

Port block diagram (3)



Notes 1: This symbol represents a parasitic diode on the port.
2: i represents 0 or 1.
3: Applied potential to ports P2 and P3 must be VDD or less.

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 A_0 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.

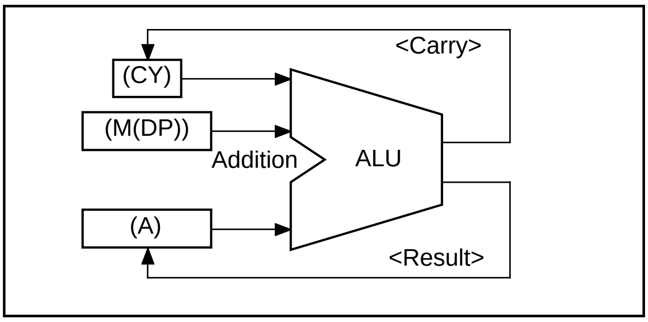


Fig. 1 AMC instruction execution example

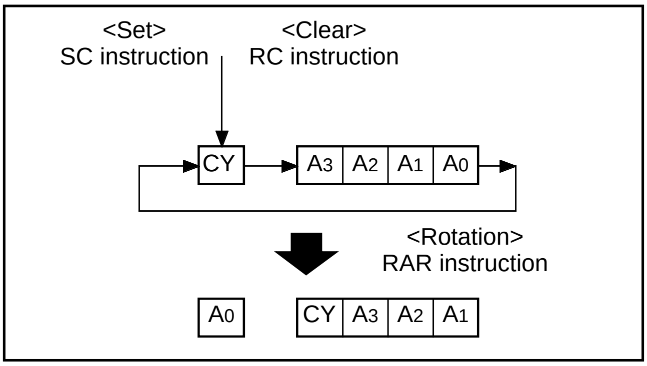


Fig. 2 RAR instruction execution example

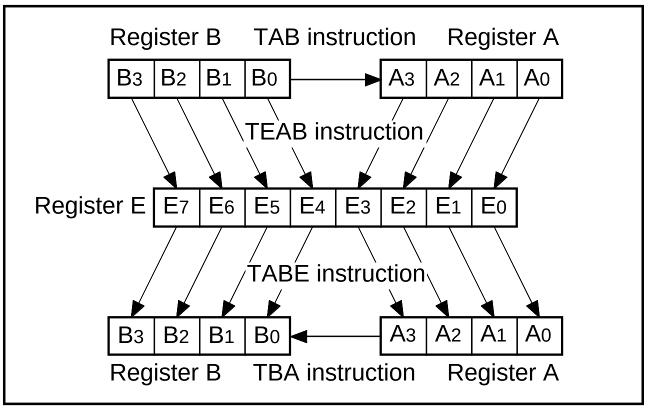


Fig. 3 Registers A, B and register E

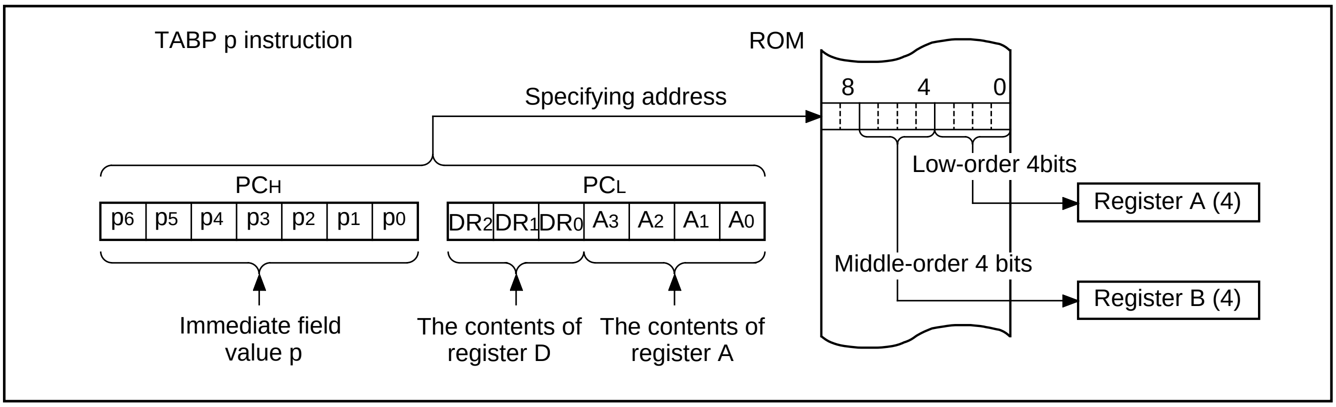


Fig. 4 TABP p instruction execution example

(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.

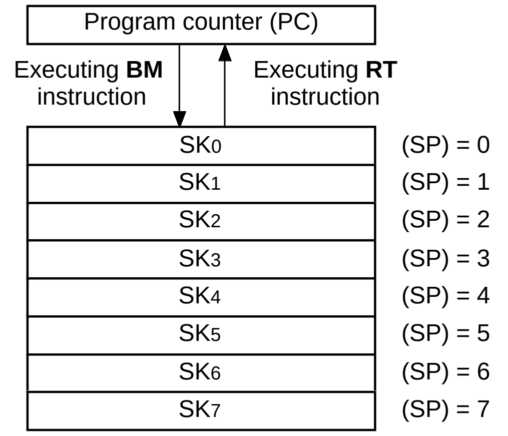
(6) Interrupt stack register (SDP)

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.

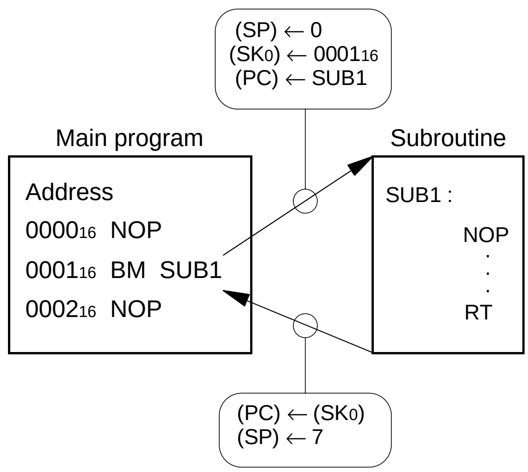
(7) Skip flag

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 SK_0. When the **BM** instruction is executed after eight stack registers are used ((SP) = 7), (SP) = 0 and the contents of SK_0 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 PC_H (most significant bit to bit 7) which specifies to a ROM page and PC_L (bits 6 to 0) which specifies an address within a page. After it reaches the last address (address 127) of a page, it specifies address 0 of the next page (Figure 7).
Make sure that the PC_H does not 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.

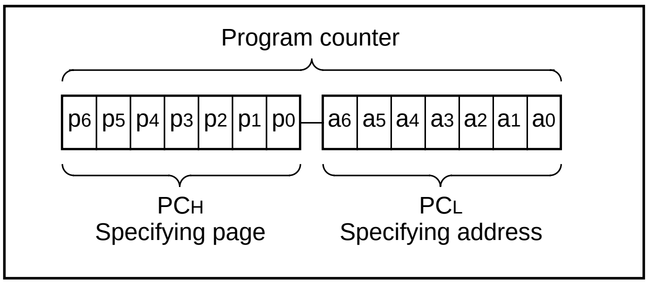


Fig. 7 Program counter (PC) structure

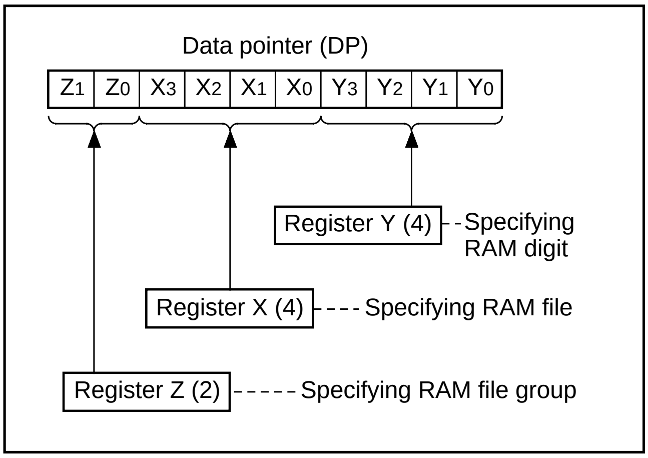


Fig. 8 Data pointer (DP) structure

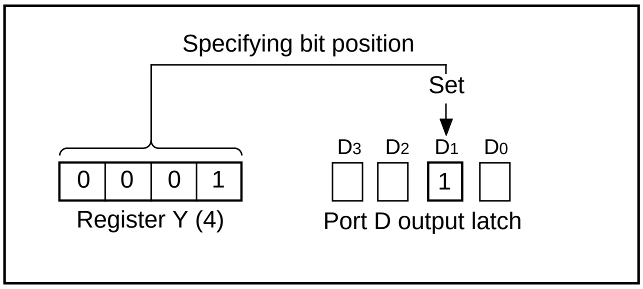


Fig. 9 SD instruction execution example

PROGRAM MEMOY (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

Product	ROM (PROM) size (× 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_{16} to $00FF_{16}$) 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_{16} to $017F_{16}$) is the special page for subroutine calls. Subroutines written in this page can be called from any page with the 1-word instruction (BM). Subroutines extending from page 2 to another page can also be called with the BM instruction when it starts on page 2.

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

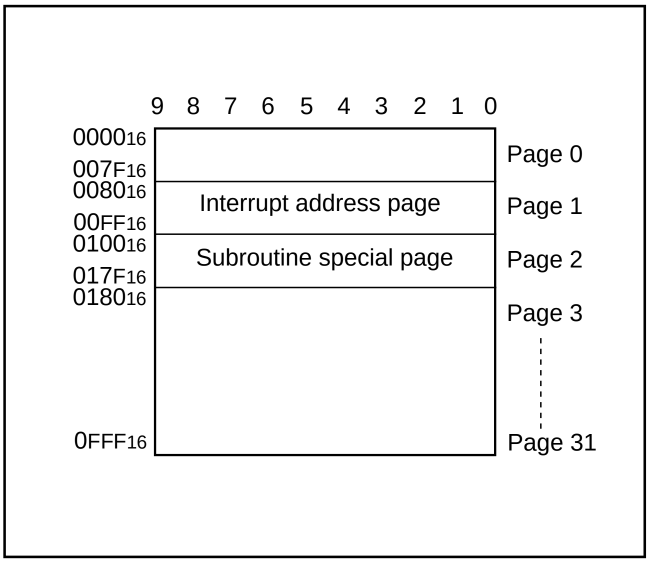


Fig. 10 ROM map of M34506M4/M34506E4

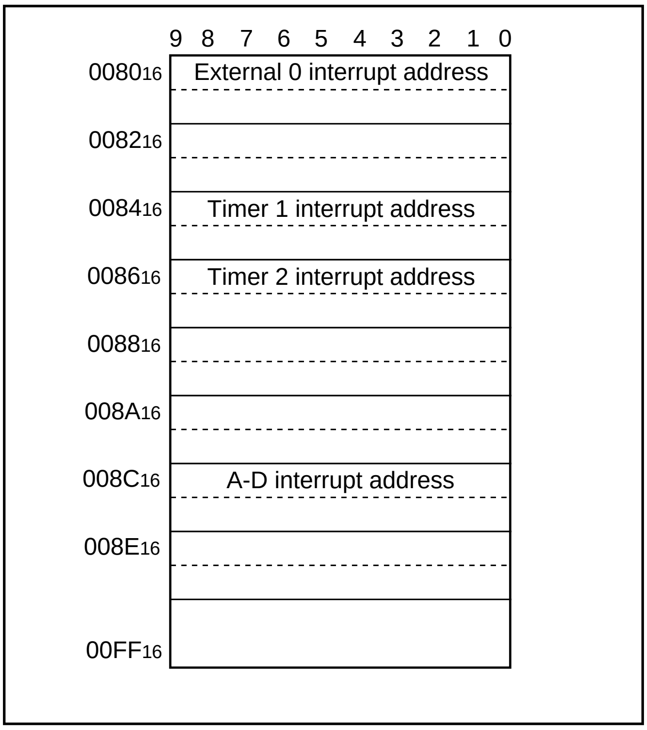


Fig. 11 Page 1 (addresses 0080_{16} to $00FF_{16}$) structure

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.

• 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.

Table 2 RAM size

Product	RAM size
M34506M2	128 words × 4 bits (512 bits)
M34506M4	256 words × 4 bits (1024 bits)
M34506E4	256 words × 4 bits (1024 bits)

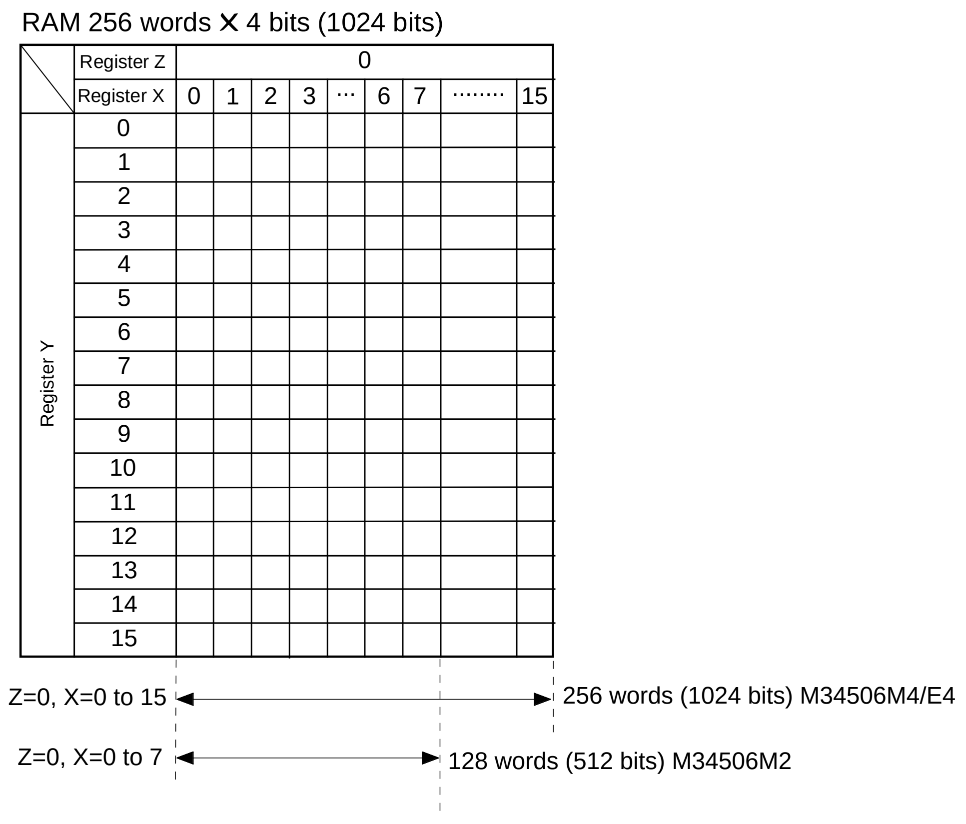


Fig. 12 RAM map

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	Request flag	Skip instruction	Enable bit
External 0 interrupt	EXF0	SNZ0	$V1_0$
Timer 1 interrupt	T1F	SNZT1	$V1_2$
Timer 2 interrupt	T2F	SNZT2	$V1_3$
A-D interrupt	ADF	SNZAD	$V2_2$

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)

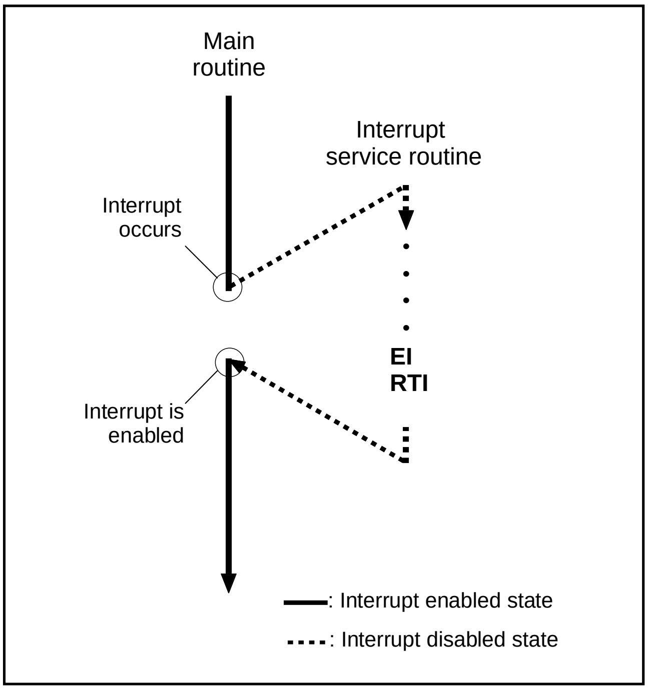


Fig. 13 Program example of interrupt processing

- Program counter (PC)
  .... Each interrupt address
- Stack register (SK)
  .... The address of main routine to be executed when returning
- Interrupt enable flag (INTE)
  .... 0 (Interrupt disabled)
- Interrupt request flag (only the flag for the current interrupt source) 0
- Data pointer, carry flag, registers A and B, skip flag
  .... Stored in the interrupt stack register (SDP) automatically

Fig. 14 Internal state when interrupt occurs

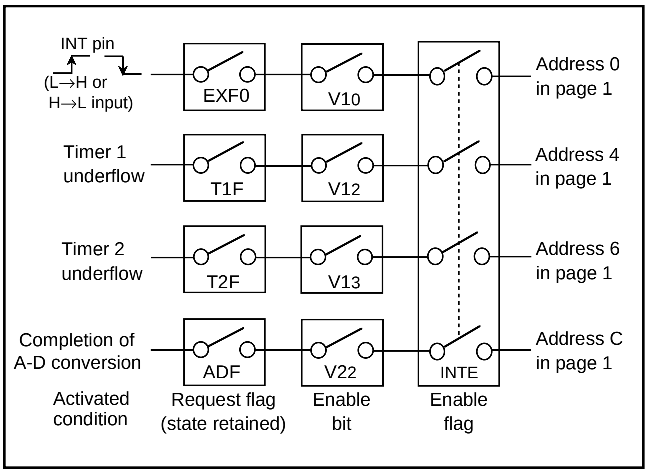


Fig. 15 Interrupt system diagram

(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_2	at RAM back-up : 0000_2	R/W
$V1_3$	Timer 2 interrupt enable bit	0	Interrupt disabled (SNZT2 instruction is valid)	
		1	Interrupt enabled (SNZT2 instruction is invalid) (Note 2)	
$V1_2$	Timer 1 interrupt enable bit	0	Interrupt disabled (SNZT1 instruction is valid)	
		1	Interrupt enabled (SNZT1 instruction is invalid) (Note 2)	
$V1_1$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$V1_0$	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_2	at RAM back-up : 0000_2	R/W
$V2_3$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$V2_2$	A-D interrupt enable bit	0	Interrupt disabled (SNZAD instruction is valid)	
		1	Interrupt enabled (SNZAD instruction is invalid) (Note 2)	
$V2_1$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$V2_0$	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 instrucion.

(7) Interrupt sequence

Interrupts only occur when the respective INTE flag, interrupt enable bits ($V1_0$, $V1_2$, $V1_3$, $V2_2$), 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).

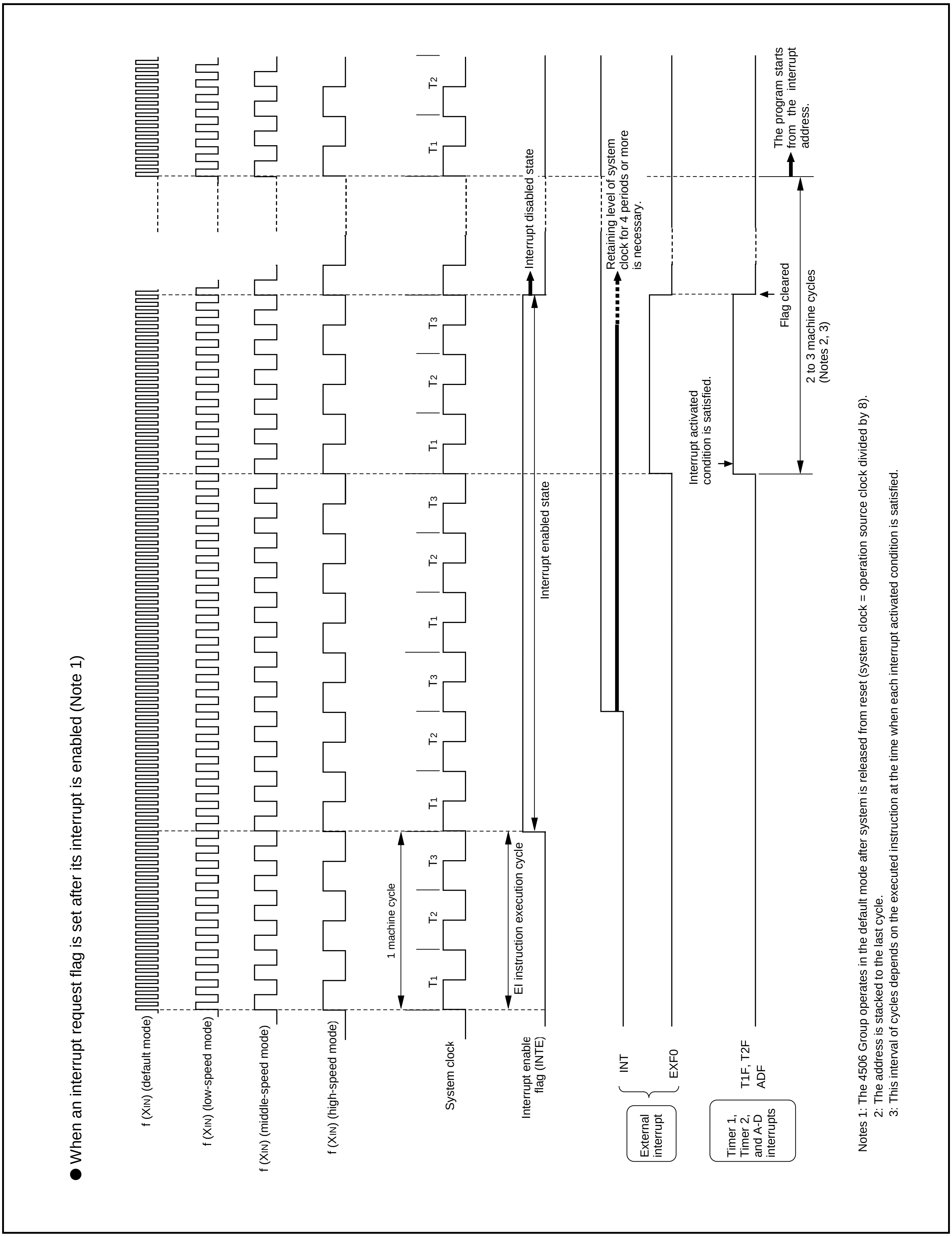


Fig. 16 Interrupt sequence

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 • Falling waveform (“H”→“L”) • Rising waveform (“L”→“H”) • Both rising and falling waveforms	$I1_1$ $I1_2$

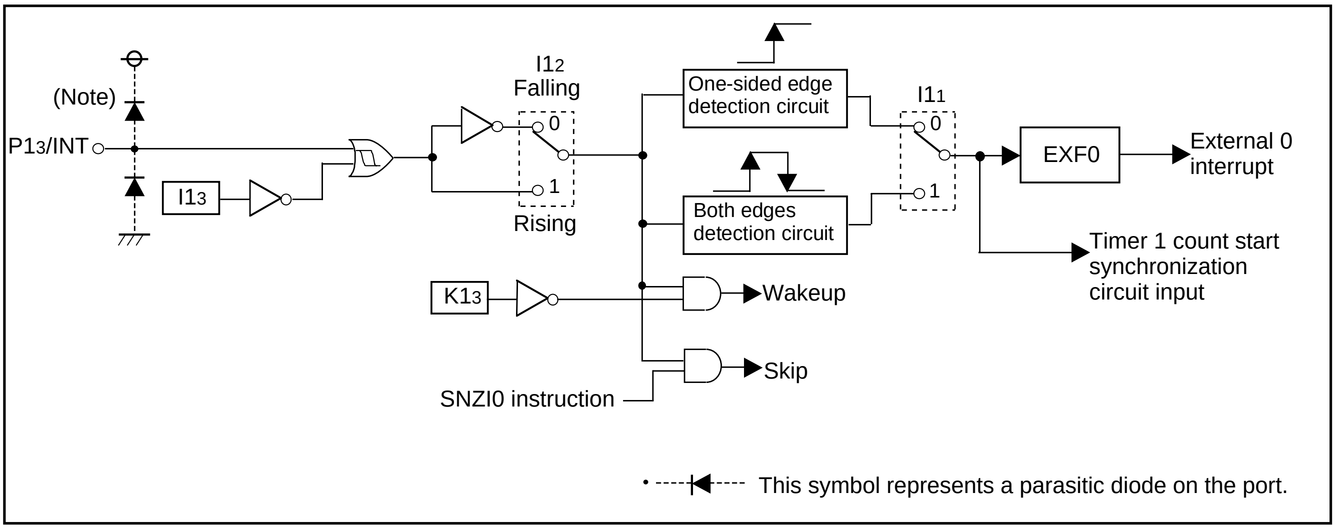


Fig. 17 External interrupt circuit structure

(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 ($V1_0$) 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 TI1A instruction. The TAI1 instruction can be used to transfer the contents of register I1 to register A.

Table 8 External interrupt control register

Interrupt control register I1		at reset : 0000_2	at RAM back-up : state retained	R/W
$I1_3$	INT pin input control bit (Note 2)	0	INT pin input disabled	
		1	INT pin input enabled	
$I1_2$	Interrupt valid waveform for INT pin/ return level selection bit (Note 2)	0	Falling waveform (“L” level of INT pin is recognized with the SNZI0 instruction)/“L” level	
		1	Rising waveform (“H” level of INT pin is recognized with the SNZI0 instruction)/“H” level	
$I1_1$	INT pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I1_0$	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 $I1_2$ and $I1_3$ are changed, the external interrupt request flag EXF0 may be set. Accordingly, clear EXF0 flag with the SNZ0 instruction when the bit 0 ($V1_0$) 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 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, 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 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
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.

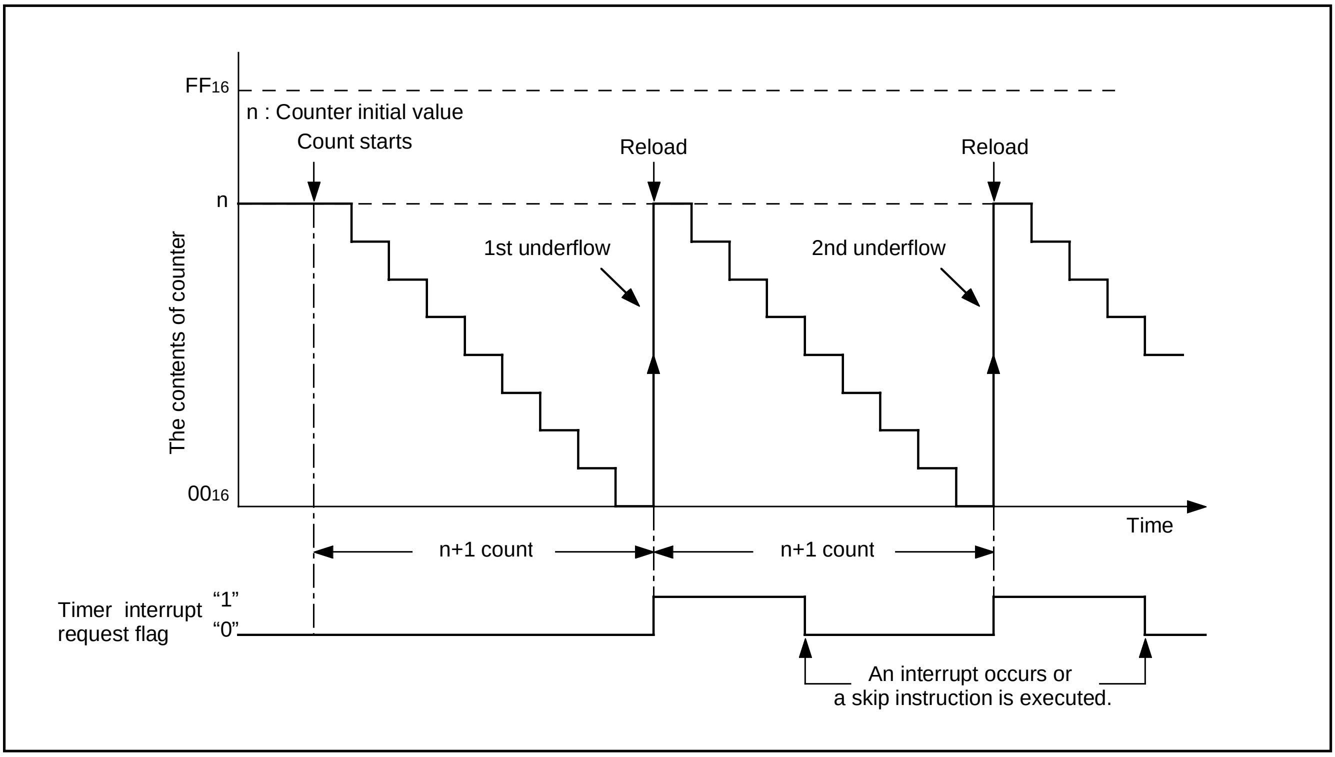


Fig. 21 Auto-reload function

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
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
16-bit timer	16-bit fixed dividing frequency binary down counter	• Instruction clock	65536	• Watchdog timer (The 16th bit is counted twice)	

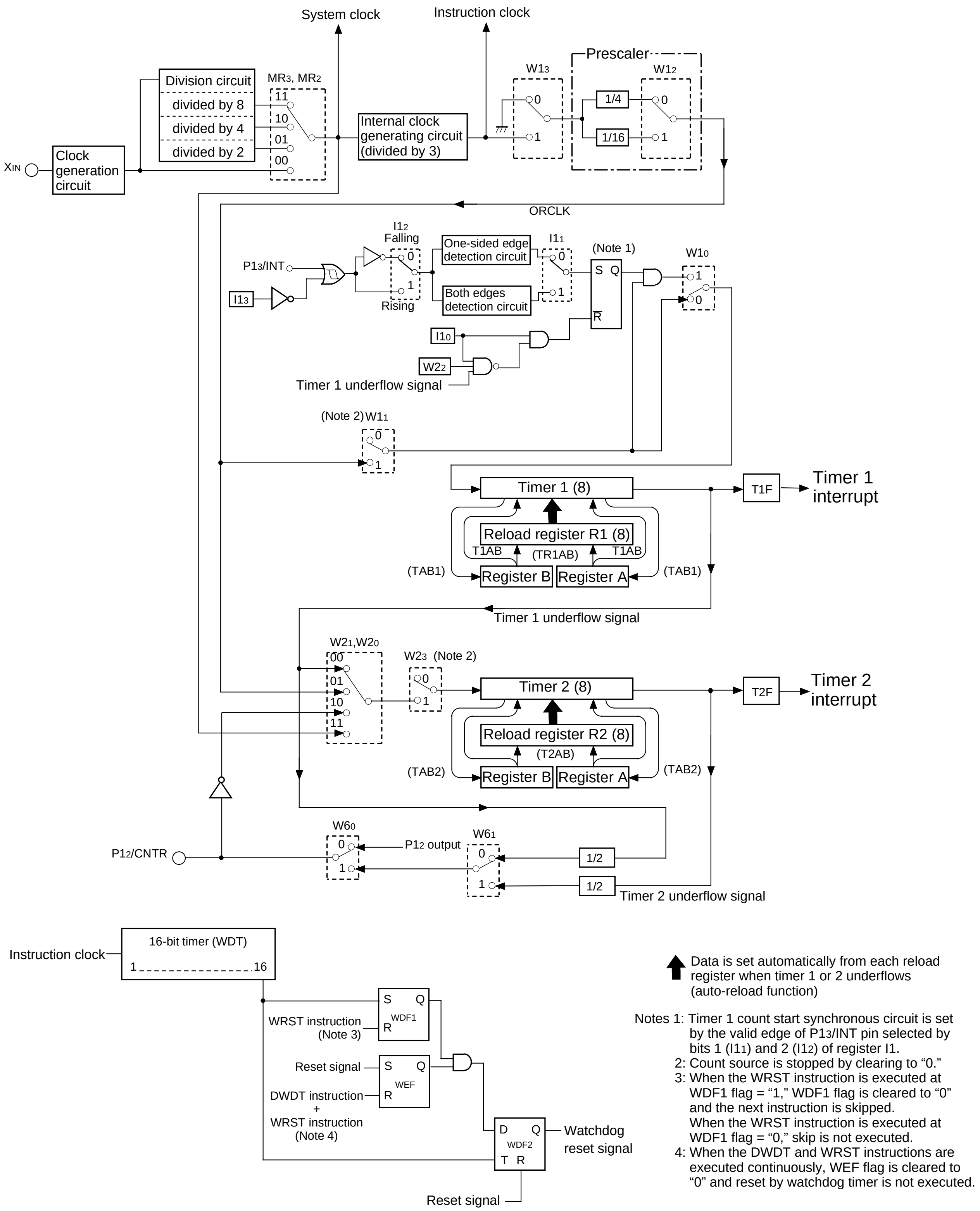


Fig. 22 Timers structure

Table 10 Timer control registers

Timer control register W1		at reset : 0000_2		at RAM back-up : 0000_2	R/W
$W1_3$	Prescaler control bit	0		Stop (state initialized)	
		1		Operating	
$W1_2$	Prescaler dividing ratio selection bit	0		Instruction clock divided by 4	
		1		Instruction clock divided by 16	
$W1_1$	Timer 1 control bit	0		Stop (state retained)	
		1		Operating	
$W1_0$	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_2		at RAM back-up : state retained	R/W
$W2_3$	Timer 2 control bit	0		Stop (state retained)	
		1		Operating	
$W2_2$	Timer 1 count auto-stop circuit selection bit (Note 2)	0		Count auto-stop circuit not selected	
		1		Count auto-stop circuit selected	
$W2_1$	Timer 2 count source selection bits	$W2_1$	$W2_0$	Count source	
		0	0	Timer 1 underflow signal	
$W2_0$		0	1	Prescaler output (ORCLK)	
		1	0	CNTR input	
		1	1	System clock	

Timer control register W6		at reset : 0000_2		at RAM back-up : state retained	R/W
$W6_3$	Not used	0		This bit has no function, but read/write is enabled.	
		1			
$W6_2$	Not used	0		This bit has no function, but read/write is enabled.	
		1			
$W6_1$	CNTR output selection bit	0		Timer 1 underflow signal divided by 2 output	
		1		Timer 2 underflow signal divided by 2 output	
$W6_0$	$P1_2$/CNTR function selection bit	0		$P1_2$(I/O)/CNTR input (Note 3)	
		1		$P1_2$ (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 $P1_2$/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 ($I1_1$) and 2 ($I1_2$) of register I1 as follows;
• $I1_1$ = "0": Synchronized with one-sided edge (falling or rising)
• $I1_1$ = "1": Synchronized with both edges (both falling and rising)
When register $I1_1$="0" (synchronized with the one-sided edge), the rising or falling waveform can be selected by the bit 2 of register I1;
• $I1_2$ = "0": Falling waveform
• $I1_2$ = "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 $I1_0$ to "0" or reset.
However, when the count auto-stop circuit is selected (register $W2_2$ = "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 cicuit 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 ($P1_2$/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 $P1_2$/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.

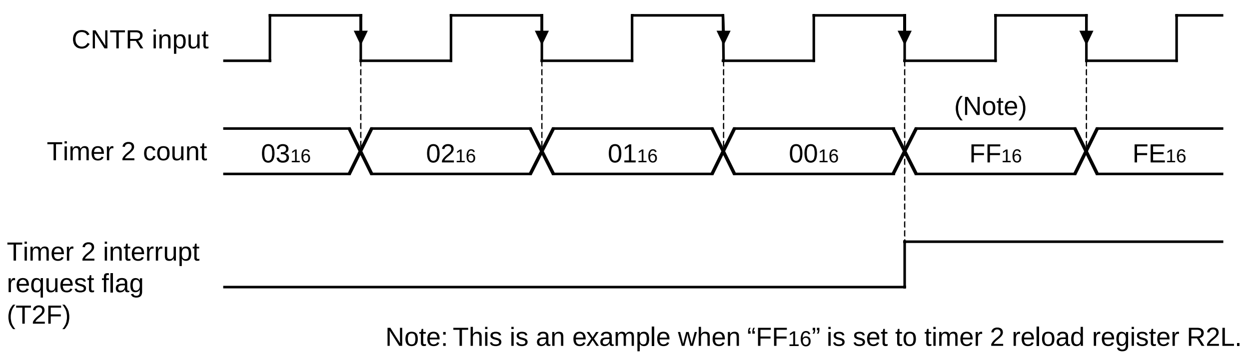
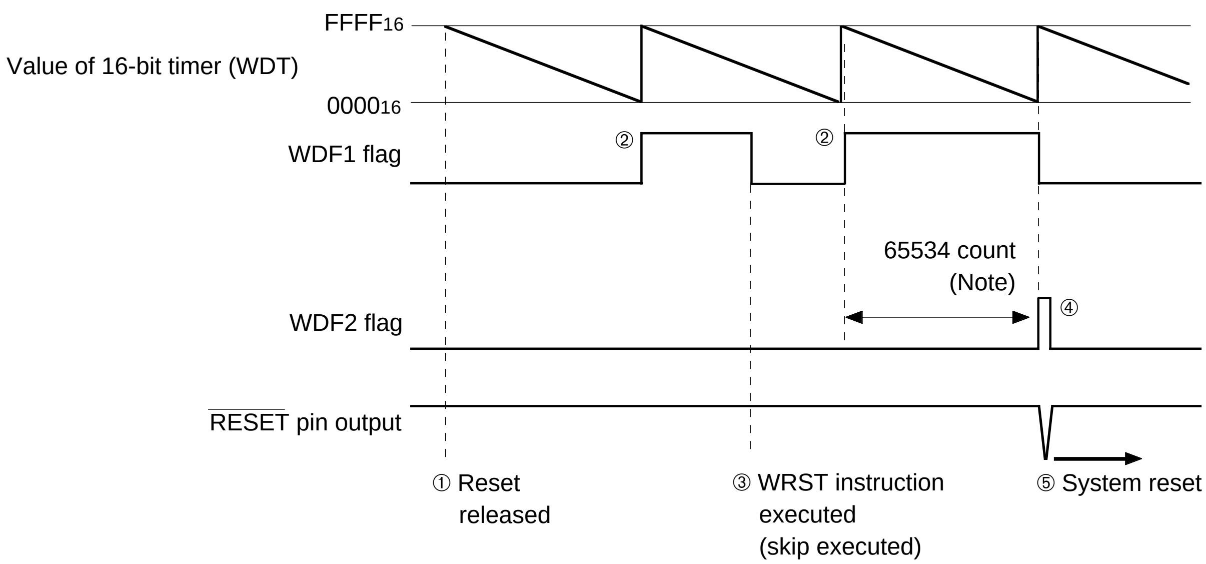


Fig. 23 Count timing diagram at CNTR input

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 "$FFFF_{16}$" 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 "$FFFF_{16}$," 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 $\overline{RESET}$ 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.
However, in order to set the WEF flag to "1" again once it has cleared to "0", execute system reset.
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.



① After system is released from reset (= after program is started), timer WDT starts count down.
② When timer WDT underflow occurs, WDF1 flag is set to "1."
③ When the WRST instruction is executed, WDF1 flag is cleared to "0," the next instruction is skipped.
④ When timer WDT underflow occurs while WDF1 flag is "1," WDF2 flag is set to "1" and the watchdog reset signal is output.
⑤ The output transistor of $\overline{RESET}$ pin is turned "ON" by the watchdog reset signal and system reset is executed.

Note: The number of count is equal to the number of cycle because the count source of watchdog timer is the instruction clock.

Fig. 24 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 25).
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 26)
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
    ⋮
DWDT        ; Watchdog timer function enabled/disabled
WRST        ; WEF and WDF1 flags cleared
    ⋮
```

Fig. 25 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)
    ⋮
```

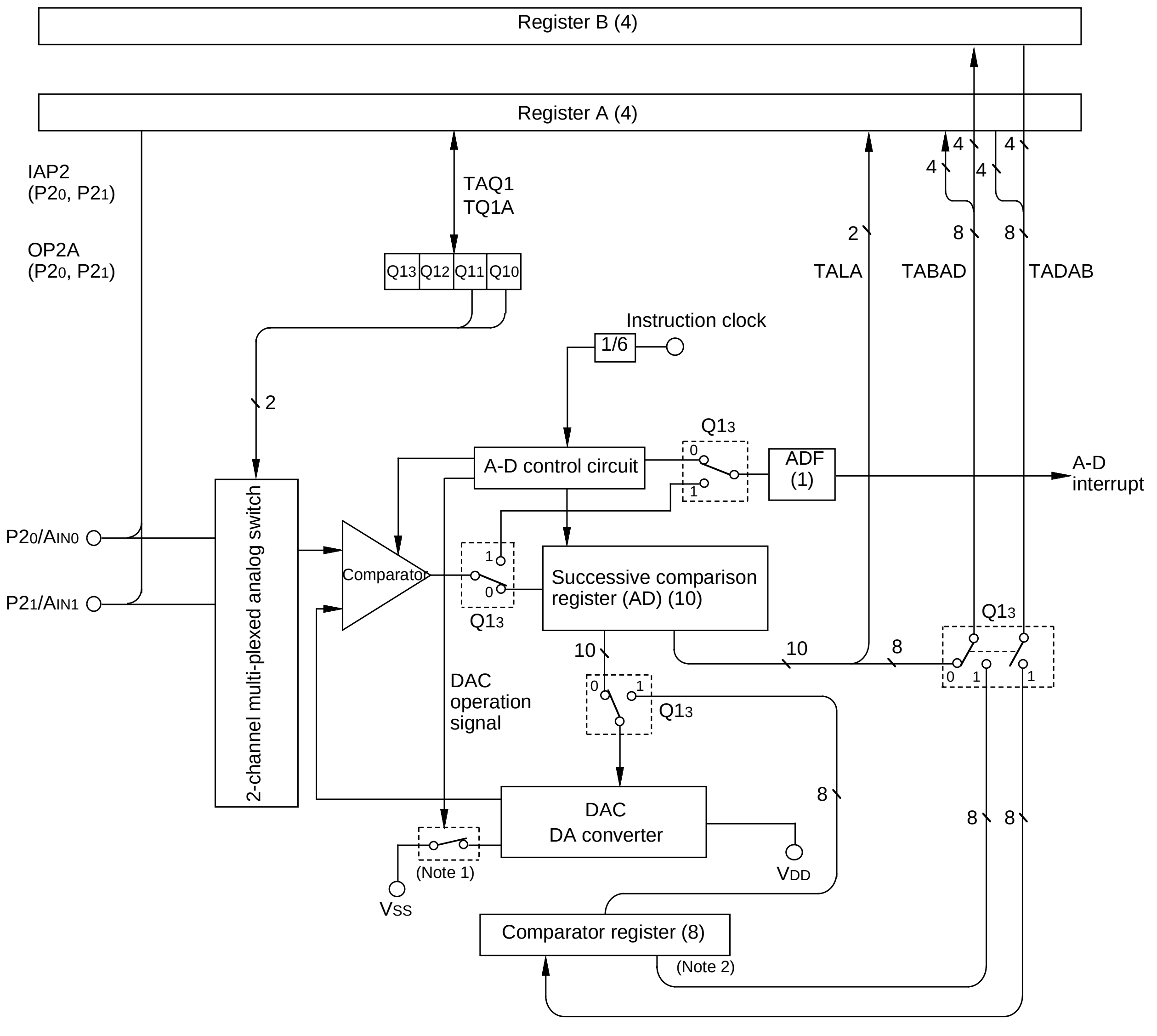
Fig. 26 Program example to enter the RAM back-up mode when using the watchdog timer

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: ±2LSB
	Non-linearity error: ±0.9LSB
Conversion speed	46.5 µs (High-speed mode at 4.0 MHz oscillation frequency)
Analog input pin	2



Notes 1: This switch is turned ON only when A-D converter is operating and generates the comparison voltage.
2: Writing/reading data to the comparator register is possible only in the comparator mode (Q13=1).
The value of the comparator register is retained even when the mode is switched to the A-D conversion mode (Q13=0) because it is separated from the successive comparison register (AD). Also, the resolution in the comparator mode is 8 bits because the comparator register consists of 8 bits.

Fig. 27 A-D conversion circuit structure

Table 12 A-D control registers

A-D control register Q1		at reset : 0000_2		at RAM back-up : state retained	R/W
$Q1_3$	A-D operation mode selection bit	0		A-D conversion mode	
		1		Comparator mode	
$Q1_2$	Not used	0		This bit has no function, but read/write is enabled.	
		1			
$Q1_1$	Analog input pin selection bits	$Q1_1$	$Q1_0$	Selected pins	
		0	0	A_{IN0}	
		0	1	A_{IN1}	
$Q1_0$		1	0	Not available	
		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:

Logic value of comparison voltage V_{ref}

$$V_{ref} = \frac{V_{DD}}{1024} \times n$$

n: The value of register AD (n = 0 to 1023)

(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 "000_{16}."

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 μs 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 28).

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

At starting conversion	Change of successive comparison register AD	Comparison voltage (V_{ref}) 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	$\frac{V_{DD}}{2} \pm \cdots \pm \frac{V_{DD}}{1024}$

✻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 28 shows the A-D conversion timing chart.

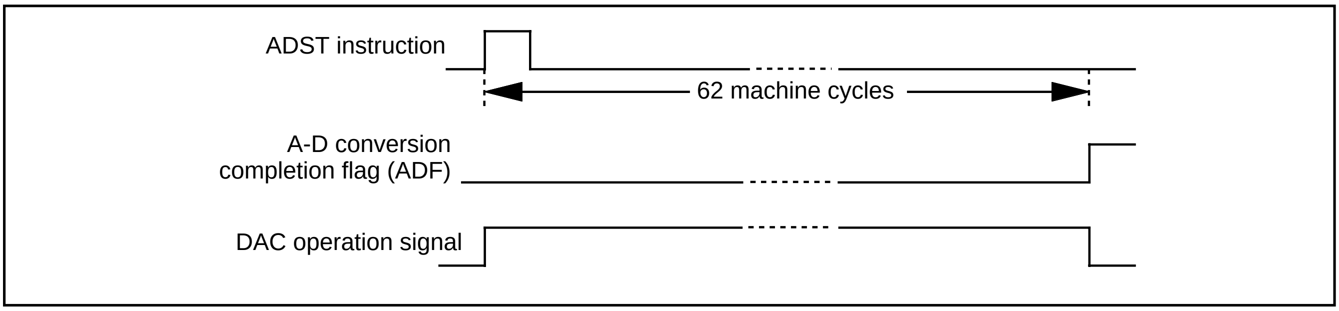


Fig. 28 A-D conversion timing chart

(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 29).
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).

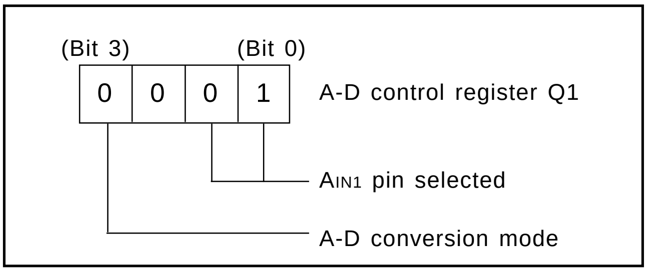


Fig. 29 Setting registers

RENESAS
Renesas Technology Corp.

(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:

Logic value of comparison voltage V_{ref}

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

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 operating.
The comparator stops 8 machine cycles after it has started (6 μs at $f(X_{IN})$ = 4.0 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 $P2_0/A_{IN0}$, $P2_1/A_{IN1}$ 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.

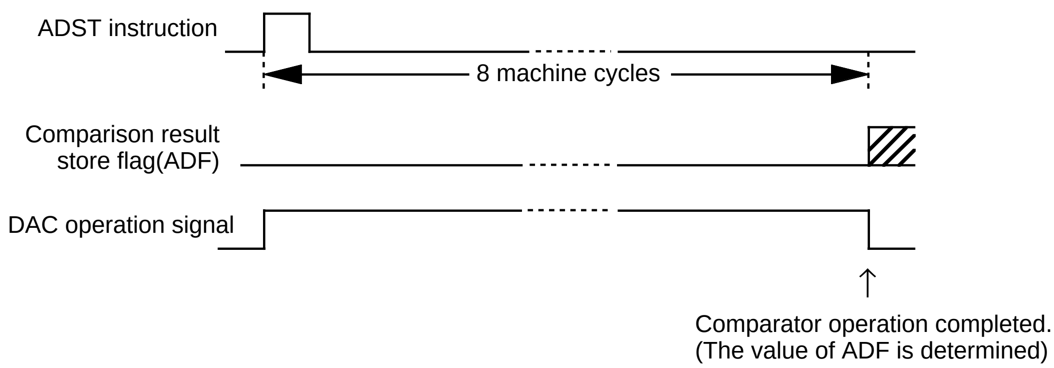


Fig. 30 Comparator operation timing chart

(15) Definition of A-D converter accuracy

The A-D conversion accuracy is defined below (refer to Figure 31).

- 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.

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

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

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

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

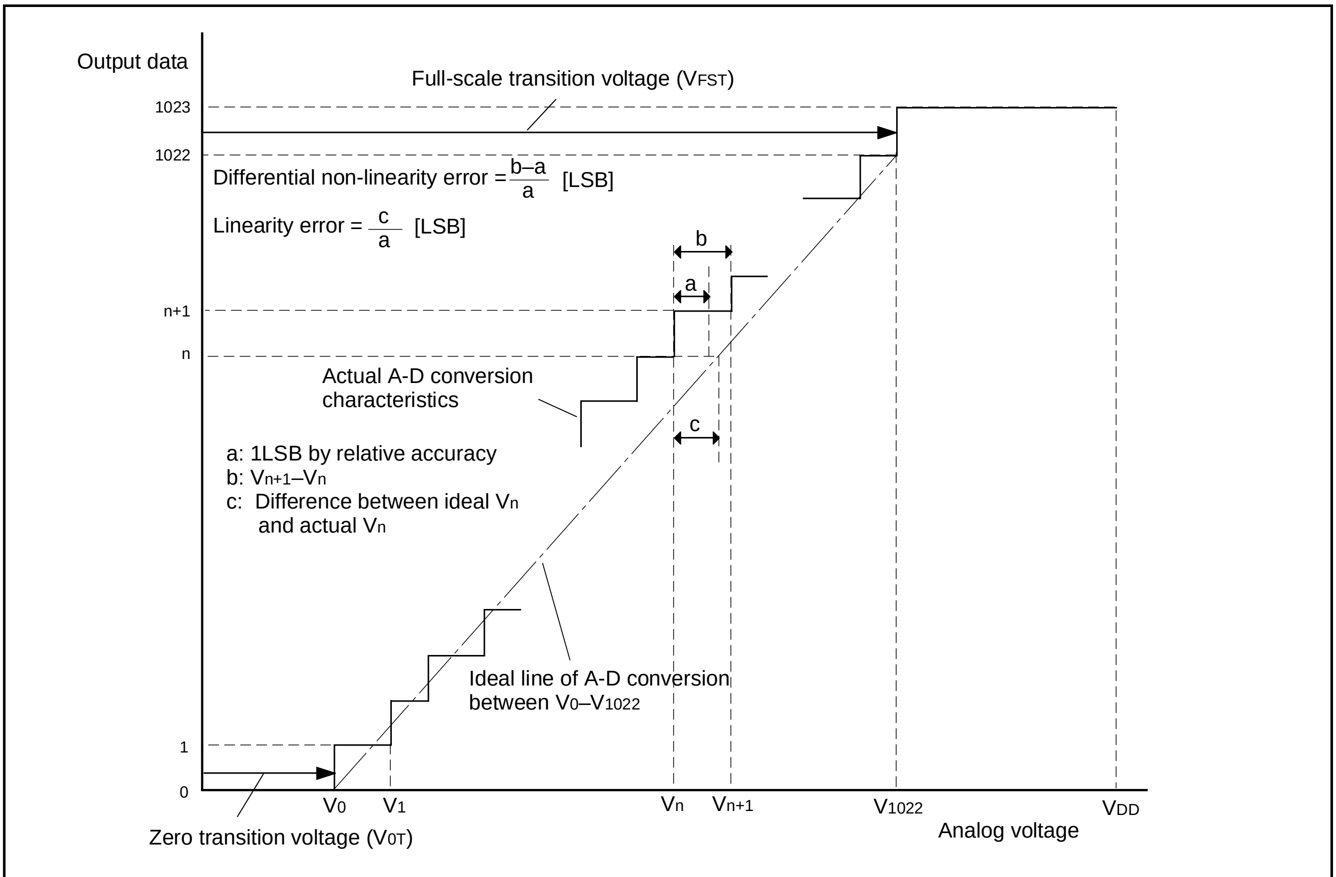


Fig. 31 Definition of A-D conversion accuracy

RENESAS
Renesas Technology Corp.

RESET FUNCTION

System reset is performed by applying “L” level to $\overline{RESET}$ 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 $\overline{RESET}$ pin, software starts from address 0 in page 0.

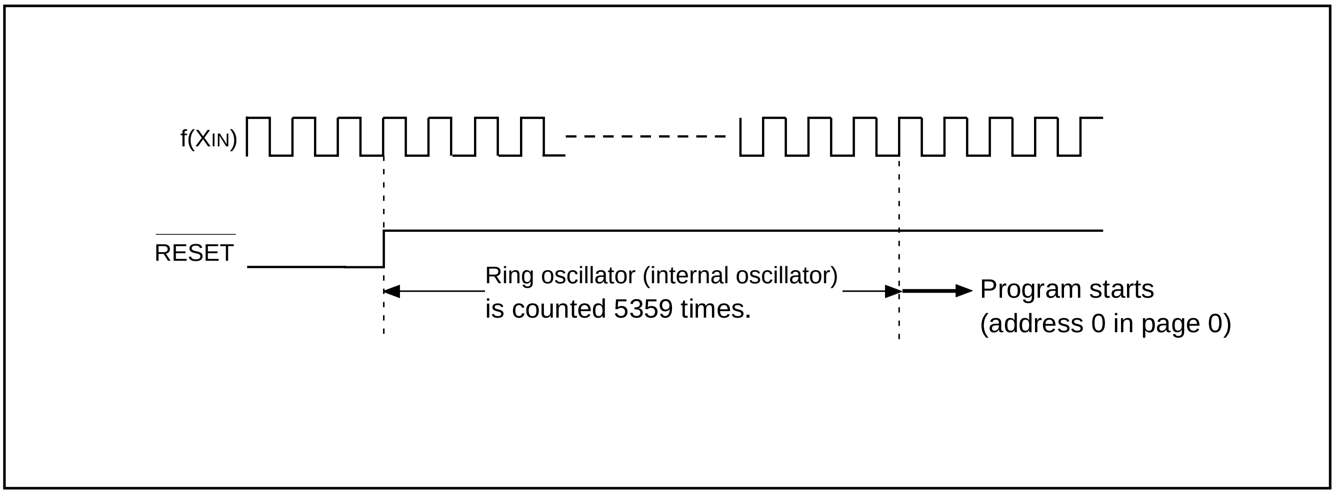


Fig. 32 Reset release timing

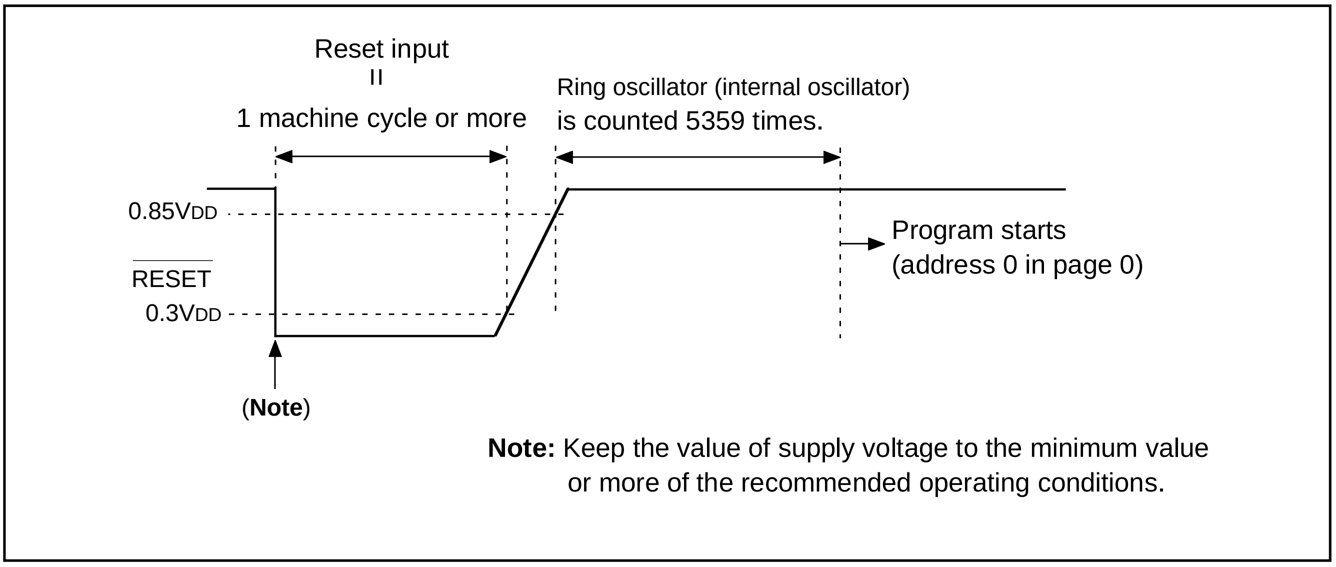


Fig. 33 $\overline{RESET}$ pin input waveform and reset operation

(1) Power-on reset

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

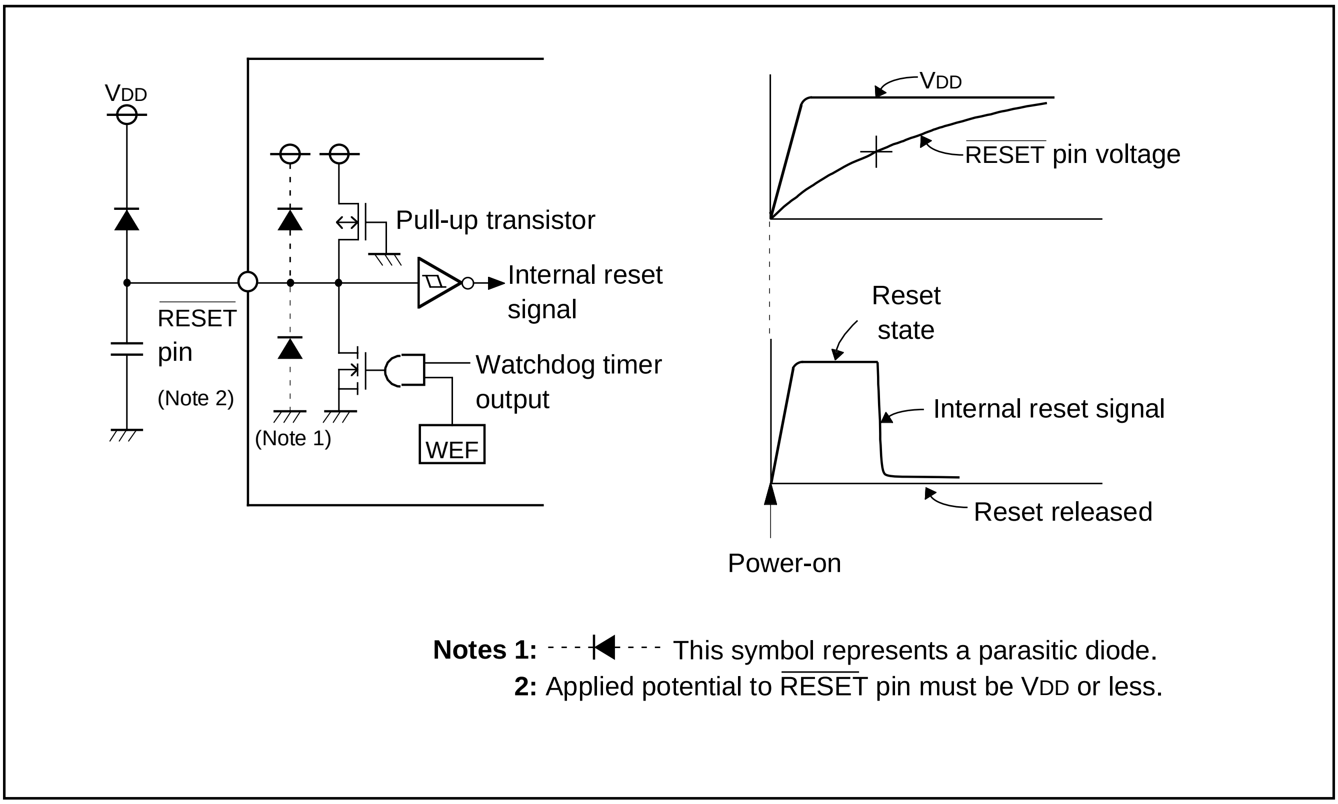


Fig. 34 Power-on reset circuit example

Table 14 Port state at reset

Name	Function	State
D_0, D_1	D_0, D_1	High-impedance (Note 1)
D_2/C, D_3/K	D_2, D_3	High-impedance (Notes 1, 2)
$P0_0$, $P0_1$, $P0_2$, $P0_3$	$P0_0$–$P0_3$	High-impedance (Notes 1, 2)
$P1_0$, $P1_1$, $P1_2$/CNTR, $P1_3$/INT	$P1_0$–$P1_3$	High-impedance (Notes 1, 2)
$P2_0$/A_{IN0}, $P2_1$/A_{IN1}	$P2_0$, $P2_1$	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 35 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 35 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
  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 .... × × ×
- Register E .... × × × × × × × ×
- Register X 0 0 0 0
- Register Y 0 0 0 0
- Register Z .... × ×
- Stack pointer (SP) 1 1 1
- Oscillation clock .... Ring oscillator (operating)
- Ceramic resonator circuit .... Operating
- RC oscillation circuit .... Stop

"×" represents undefined.

Fig. 35 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 14 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 14 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	O
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 with the WRST instruction, and then execute the POF2 instruction.

(4) Return signal

An external wakeup signal is used to return from the RAM back-up mode because the oscillation is stopped. Table 15 shows the return condition for each return source.

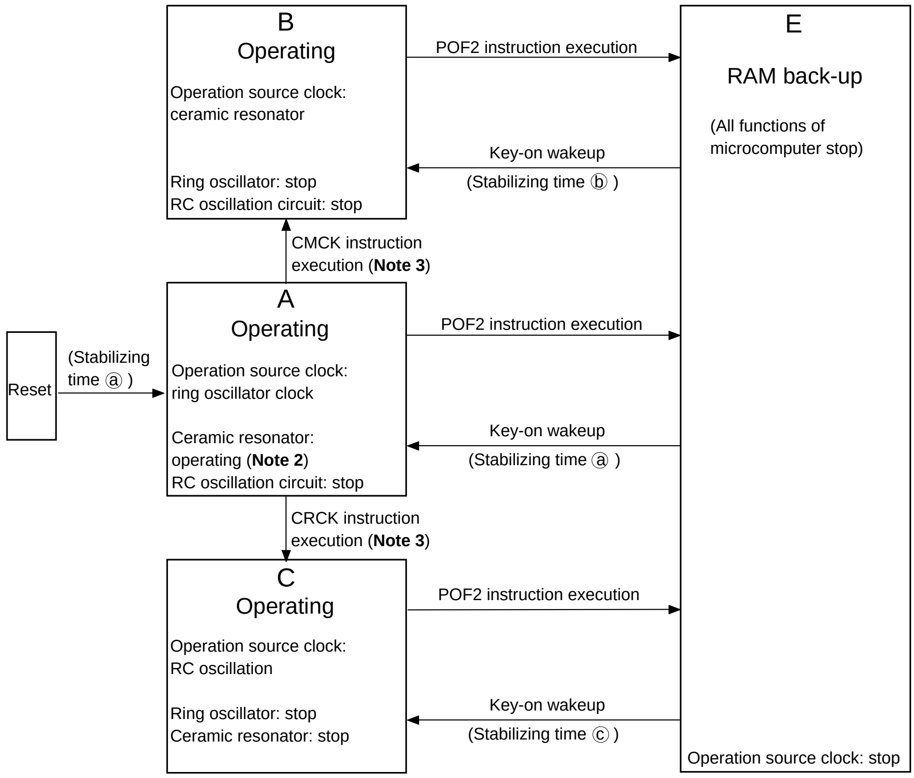
(5) Control registers

- 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, D_2/C and D_3/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, D_2/C and D_3/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 15 Return source and return condition

Return source		Return condition	Remarks
External wakeup signal	Port P0 Port P1 (Note) Port P2 Ports D_2/C, D_3/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 $P1_3$/INT (Note)	Return by an external “H” level or “L” level input. The return level can be selected with the bit 2 ($I1_2$) 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 ($K1_3$) 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 $P1_3$ is valid (“L” level).



Stabilizing time ⓐ: Microcomputer starts its operation after counting the ring oscillator clock 5359 times by hardware.
Stabilizing time ⓑ: Microcomputer starts its operation after counting the f(XIN) 5359 times by hardware.
Stabilizing time ⓒ: Microcomputer starts its operation after counting the f(XIN) 165 times by hardware.

Notes 1: Continuous execution of the EPOF instruction and POF2 instruction is required to go into the RAM back-up state.
2: Through the ceramic resonator is operating, the ring oscillator clock is selected as the operation source clock.
3: The oscillator clock corresponding to each instruction is selected as the operation source clock, and the ring oscillator is stopped.

Fig. 36 State transition

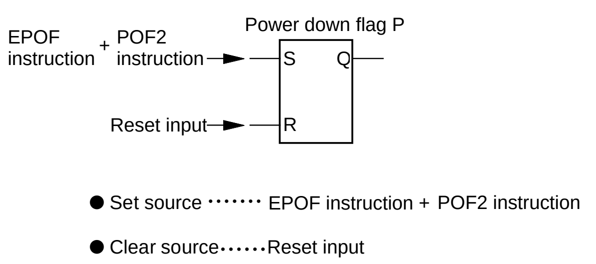


Fig. 37 Set source and clear source of the P flag

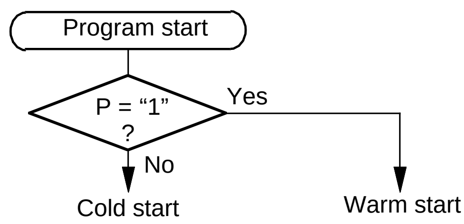


Fig. 38 Start condition identified example using the SNZP instruction

Table 16 Key-on wakeup control register

Key-on wakeup control register K0		at reset : 0000_2		at RAM back-up : state retained	R/W
$K0_3$	Port $P0_3$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_2$	Port $P0_2$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_1$	Port $P0_1$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_0$	Port $P0_0$ 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_2		at RAM back-up : state retained	R/W
$K1_3$	Port $P1_3$/INT key-on wakeup control bit	0	$P1_3$ key-on wakeup not used/INT pin key-on wakeup used		
		1	$P1_3$ key-on wakeup used/INT pin key-on wakeup not used		
$K1_2$	Port $P1_2$/CNTR key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K1_1$	Port $P1_1$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K1_0$	Port $P1_0$ 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_2		at RAM back-up : state retained	R/W
$K2_3$	Port D_3/K key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K2_2$	Port D_2/C key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K2_1$	Port $P2_1$/A_{IN1} key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K2_0$	Port $P2_0$/A_{IN0} 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 17 Pull-up control register and interrupt control register

Pull-up control register PU0	at reset : 0000_2	at RAM back-up : state retained	W
$PU0_3$	Port $P0_3$ pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON
$PU0_2$	Port $P0_2$ pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON
$PU0_1$	Port $P0_1$ pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON
$PU0_0$	Port $P0_0$ pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON

Pull-up control register PU1	at reset : 0000_2	at RAM back-up : state retained	W
$PU1_3$	Port $P1_3$/INT pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON
$PU1_2$	Port $P1_2$/CNTR pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON
$PU1_1$	Port $P1_1$ pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON
$PU1_0$	Port $P1_0$ pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON

Pull-up control register PU2	at reset : 0000_2	at RAM back-up : state retained	W
$PU2_3$	Port D_3/K pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON
$PU2_2$	Port D_2/C pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON
$PU2_1$	Port $P2_1$/A_{IN1} pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON
$PU2_0$	Port $P2_0$/A_{IN0} pull-up transistor control bit	0	Pull-up transistor OFF
		1	Pull-up transistor ON

Interrupt control register I1	at reset : 0000_2	at RAM back-up : state retained	R/W
$I1_3$	INT pin input control bit (Note 2)	0	INT pin input disabled
		1	INT pin input enabled
$I1_2$	Interrupt valid waveform for INT pin/ return level selection bit (Note 2)	0	Falling waveform (“L” level of INT pin is recognized with the SNZI0 instruction)/“L” level
		1	Rising waveform (“H” level of INT pin is recognized with the SNZI0 instruction)/“H” level
$I1_1$	INT pin edge detection circuit control bit	0	One-sided edge detected
		1	Both edges detected
$I1_0$	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 $I1_2$ and $I1_3$ are changed, the external interrupt request flag EXF0 may be set. Accordingly, clear EXF0 flag with the SNZ0 instruction when the bit 0 ($V1_0$) 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.

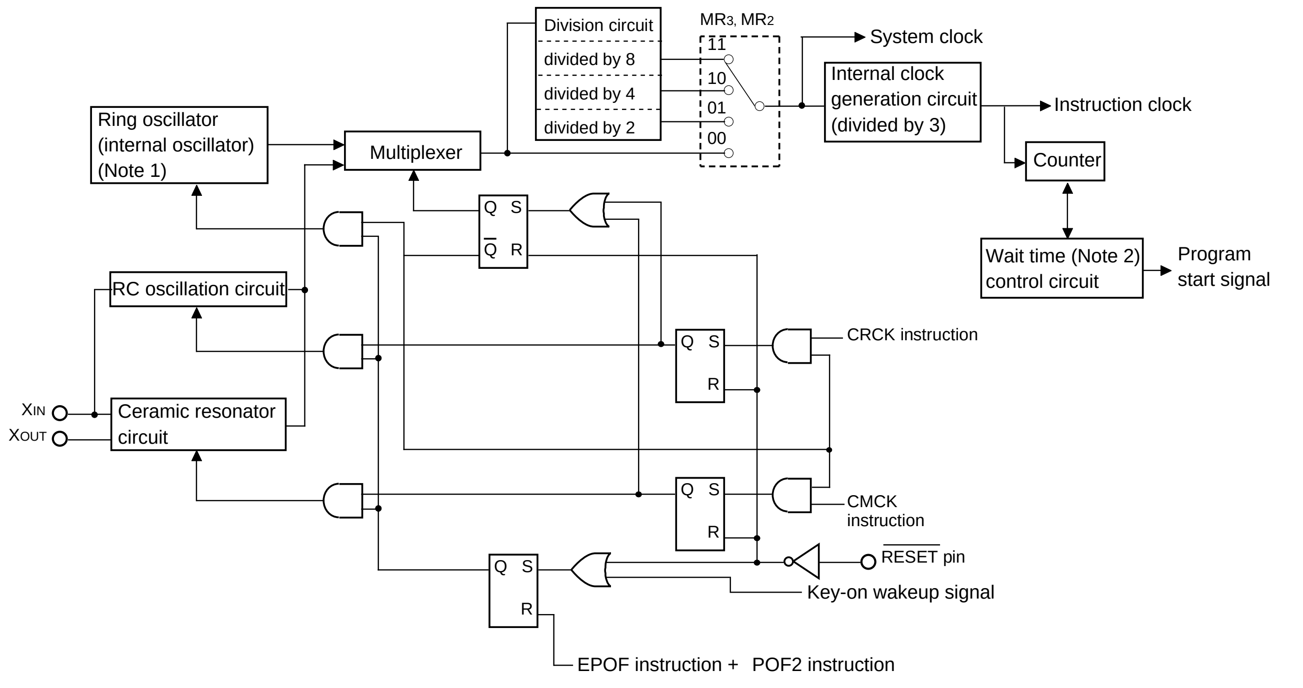
- Ring oscillator (internal oscillator)
- Ceramic oscillator
- RC oscillation circuit
- Multi-plexer (clock selection circuit)
- Frequency divider
- Internal clock generating circuit

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

Figure 39 shows the structure of the clock control circuit.

The 4506 Group operates by the ring oscillator clock (f(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(XIN)) of the 4506 Group. The CMCK instruction or CRCK instruction is executed to select the ceramic resonator or RC oscillator, respectively.



Notes 1: System operates by the ring oscillator clock (f(RING)) until the CMCK or CRCK instruction is executed after system is released from reset.

2: The wait time control circuit is used to generate the time required to stabilize the f(XIN) oscillation. After the certain oscillation stabilizing wait time elapses, the program start signal is output. This circuit operates when system is released from reset or returned from RAM back-up.

Fig. 39 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 ring 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 ring 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 ring oscillator.

(2) Ring oscillator operation

When the MCU operates by the ring 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 41).
The clock frequency of the ring 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 42).

(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 43).
The frequency is affected by a capacitor, a resistor and a microcomputer. So, set the constants within the range of the frequency limits.

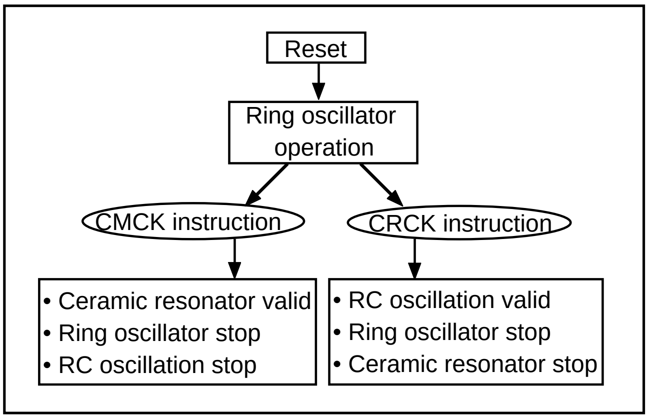


Fig. 40 Switch to ceramic resonance/RC oscillation

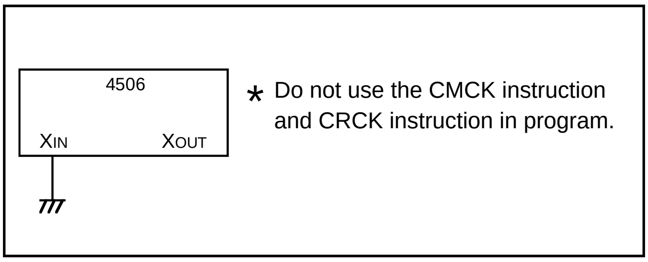


Fig. 41 Handling of XIN and XOUT when operating ring oscillator

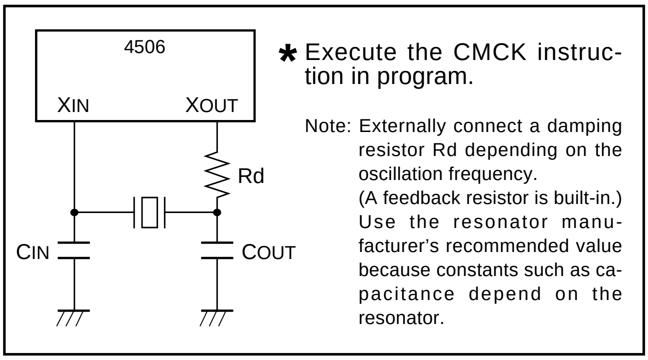


Fig. 42 Ceramic resonator external circuit

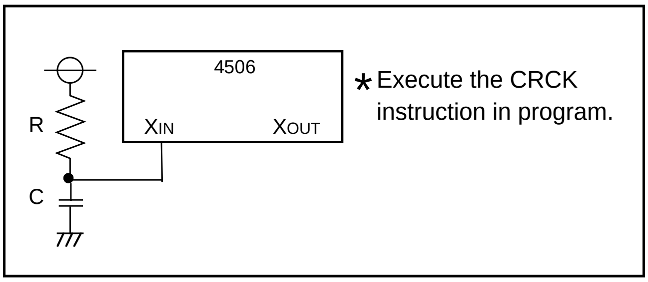


Fig. 43 External RC oscillation circuit

(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 44).
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.

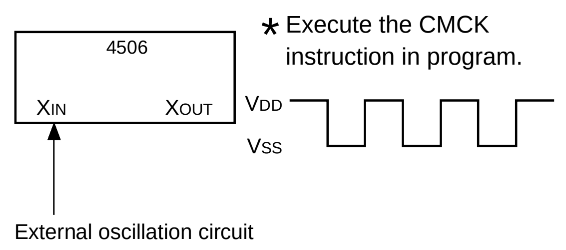


Fig. 44 External clock input circuit

Table 17 Clock control register MR

Clock control register MR		at reset : 1100_2			at RAM back-up : 1100_2	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

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 μF) 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 kΩ (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 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.

⑪ Multifunction

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

⑫ Program counter

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

⑬ 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.

⑭ 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 45①) and then, change the bit 3 of register I1.

  In addition, execute the SNZ0 instruction to clear the EXF0 flag after executing at least one instruction (refer to Figure 45②).

  Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 45③).

```
     ⋮
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. 45 External 0 interrupt program example-1

Note [2] on bit 3 of register I1

When the bit 3 of register I1 is cleared, 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 46①).

```
    ⋮
LA     0       ; (00××2)
TI1A           ; Input of INT disabled ........................ ①
DI
EPOF
POF2           ; RAM back-up
    ⋮
   × : these bits are not used here.
```

Fig. 46 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 47①) and then, change the bit 2 of register I1.
In addition, execute the SNZ0 instruction to clear the EXF0 flag 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       ; (×××02)
TV1A           ; The SNZ0 instruction is valid ........... ①
LA     12
TI1A           ; Interrupt valid waveform is changed
NOP            ........................................................ ②
SNZ0           ; The SNZ0 instruction is executed
                (EXF0 flag cleared)
NOP            ........................................................ ③
    ⋮
   × : these bits are not used here.
```

Fig. 47 External 0 interrupt program example-3

⑮ 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 ring oscillator stop.

⑯ Ring oscillator

The clock frequency of the ring 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 ring oscillator clock. When considering the oscillation stabilize wait time after system is released from reset, be careful that the variable frequency of the ring oscillator clock.

⑰ 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.

⑱ 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.”

⑲ 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 48①) 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.

```
        ⋮
LA     8        ; (✕0✕✕2)
TV2A            ; The SNZAD instruction is valid ........ ①
LA     0        ; (0✕✕✕2)
TQ1A            ; Operation mode of A-D converter is
                  changed from comparator mode to A-D
                  conversion mode.
SNZAD
NOP
        ⋮

      ✕ : this bit is not related to change the operation
          mode of A-D converter.
```

Fig. 48 External 0 interrupt program example-3

⑳ 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 49).

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 the Figure 50. In addition, test the application products sufficiently.

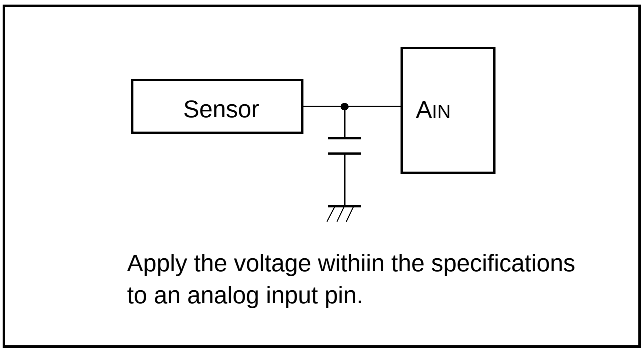


Fig. 49 Analog input external circuit example-1

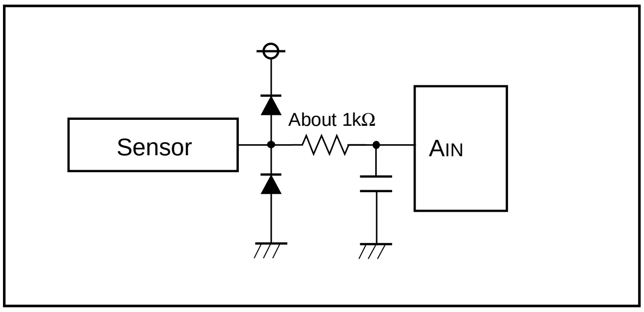


Fig. 50 Analog input external circuit example-2

CONTROL REGISTERS

Interrupt control register V1		at reset : 0000_2	at RAM back-up : 0000_2	R/W
$V1_3$	Timer 2 interrupt enable bit	0	Interrupt disabled (SNZT2 instruction is valid)	
		1	Interrupt enabled (SNZT2 instruction is invalid) (Note 2)	
$V1_2$	Timer 1 interrupt enable bit	0	Interrupt disabled (SNZT1 instruction is valid)	
		1	Interrupt enabled (SNZT1 instruction is invalid) (Note 2)	
$V1_1$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$V1_0$	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_2	at RAM back-up : 0000_2	R/W
$V2_3$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$V2_2$	A-D interrupt enable bit	0	Interrupt disabled (SNZAD instruction is valid)	
		1	Interrupt enabled (SNZAD instruction is invalid) (Note 2)	
$V2_1$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$V2_0$	Not used	0	This bit has no function, but read/write is enabled.	
		1		

Interrupt control register I1		at reset : 0000_2	at RAM back-up : state retained	R/W
$I1_3$	INT pin input control bit (Note 3)	0	INT pin input disabled	
		1	INT pin input enabled	
$I1_2$	Interrupt valid waveform for INT pin/ return level selection bit (Note 3)	0	Falling waveform (“L” level of INT pin is recognized with the SNZI0 instruction)/“L” level	
		1	Rising waveform (“H” level of INT pin is recognized with the SNZI0 instruction)/“H” level	
$I1_1$	INT pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I1_0$	INT pin timer 1 control enable bit	0	Disabled	
		1	Enabled	

Clock control register MR		at reset : 1100_2		at RAM back-up : 1100_2	R/W
MR_3	System clock selection bits	MR_3	MR_2	System clock	
		0	0	f(X_{IN}) (high-speed mode)	
		0	1	f(X_{IN})/2 (middle-speed mode)	
MR_2		1	0	f(X_{IN})/4 (low-speed mode)	
		1	1	f(X_{IN})/8 (default mode)	
MR_1	Not used	0		This bit has no function, but read/write is enabled.	
		1			
MR_0	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 $I1_2$ and $I1_3$ are changed, the external interrupt request flag EXF0 may be set. Accordingly, clear EXF0 flag with the SNZ0 instruction when the bit 0 ($V1_0$) 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 : 0000_2		at RAM back-up : 0000_2	R/W
$W1_3$	Prescaler control bit	0		Stop (state initialized)	
		1		Operating	
$W1_2$	Prescaler dividing ratio selection bit	0		Instruction clock divided by 4	
		1		Instruction clock divided by 16	
$W1_1$	Timer 1 control bit	0		Stop (state retained)	
		1		Operating	
$W1_0$	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_2		at RAM back-up : state retained	R/W
$W2_3$	Timer 2 control bit	0		Stop (state retained)	
		1		Operating	
$W2_2$	Timer 1 count auto-stop circuit selection bit (Note 2)	0		Count auto-stop circuit not selected	
		1		Count auto-stop circuit selected	
$W2_1$	Timer 2 count source selection bits	$W2_1$	$W2_0$	Count source	
		0	0	Timer 1 underflow signal	
$W2_0$		0	1	Prescaler output (ORCLK)	
		1	0	CNTR input	
		1	1	System clock	

Timer control register W6		at reset : 0000_2		at RAM back-up : state retained	R/W
$W6_3$	Not used	0		This bit has no function, but read/write is enabled.	
		1			
$W6_2$	Not used	0		This bit has no function, but read/write is enabled.	
		1			
$W6_1$	CNTR output selection bit	0		Timer 1 underflow signal divided by 2 output	
		1		Timer 2 underflow signal divided by 2 output	
$W6_0$	$P1_2$/CNTR function selection bit	0		$P1_2$(I/O)/CNTR input (Note 3)	
		1		$P1_2$ (input)/CNTR input/output (Note 3)	

A-D control register Q1		at reset : 0000_2		at RAM back-up : state retained	R/W
$Q1_3$	A-D operation mode selection bit	0		A-D conversion mode	
		1		Comparator mode	
$Q1_2$	Not used	0		This bit has no function, but read/write is enabled.	
		1			
$Q1_1$	Analog input pin selection bits	$Q1_1$	$Q1_0$	Selected pins	
		0	0	A_{IN0}	
		0	1	A_{IN1}	
$Q1_0$		1	0	Not available	
		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_2		at RAM back-up : state retained	R/W
$K0_3$	Port $P0_3$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_2$	Port $P0_2$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_1$	Port $P0_1$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_0$	Port $P0_0$ 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_2		at RAM back-up : state retained	R/W
$K1_3$	Port $P1_3$/INT key-on wakeup control bit	0	$P1_3$ key-on wakeup not used/INT pin key-on wakeup used		
		1	$P1_3$ key-on wakeup used/INT pin key-on wakeup not used		
$K1_2$	Port $P1_2$/CNTR key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K1_1$	Port $P1_1$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K1_0$	Port $P1_0$ 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_2		at RAM back-up : state retained	R/W
$K2_3$	Port D_3/K key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K2_2$	Port D_2/C key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K2_1$	Port $P2_1$/A_{IN1} key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K2_0$	Port $P2_0$/A_{IN0} 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 : 0000_2		at RAM back-up : state retained	W
$PU0_3$	Port $P0_3$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU0_2$	Port $P0_2$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU0_1$	Port $P0_1$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU0_0$	Port $P0_0$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		

Pull-up control register PU1		at reset : 0000_2		at RAM back-up : state retained	W
$PU1_3$	Port $P1_3$/INT pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU1_2$	Port $P1_2$/CNTR pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU1_1$	Port $P1_1$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU1_0$	Port $P1_0$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		

Pull-up control register PU2		at reset : 0000_2		at RAM back-up : state retained	W
$PU2_3$	Port D_3/K pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU2_2$	Port D_2/C pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU2_1$	Port $P2_1$/A_{IN1} pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU2_0$	Port $P2_0$/A_{IN0} 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)	$A_3A_2A_1A_0$	Binary notation of hexadecimal variable A
	(It consists of registers X, Y, and Z)		(same for others)
PC	Program counter (14 bits)		
PC_H	High-order 7 bits of program counter	←	Direction of data movement
PC_L	Low-order 7 bits of program counter	↔	Data exchange between a register and memory
SK	Stack register (14 bits × 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 $a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$
T1	Timer 1	p, a	Label indicating address $a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$
T2	Timer 2		in page $p_5\ p_4\ p_3\ p_2\ p_1\ p_0$
T1F	Timer 1 interrupt request flag	C + x	Hex. C + Hex. number x (also same for others)
T2F	Timer 2 interrupt request flag		

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
Register to register transfer	TAB	(A) ← (B)	75, 88
	TBA	(B) ← (A)	81, 88
	TAY	(A) ← (Y)	81, 88
	TYA	(Y) ← (A)	86, 88
	TEAB	(E_7–E_4) ← (B) (E_3–E_0) ← (A)	82, 88
	TABE	(B) ← (E_7–E_4) (A) ← (E_3–E_0)	76, 88
	TDA	(DR_2–DR_0) ← (A_2–A_0)	81, 88
	TAD	(A_2–A_0) ← (DR_2–DR_0) (A_3) ← 0	76, 88
	TAZ	(A_1, A_0) ← (Z_1, Z_0) (A_3, A_2) ← 0	81, 88
	TAX	(A) ← (X)	80, 88
	TASP	(A_2–A_0) ← (SP_2–SP_0) (A_3) ← 0	79, 88
RAM addresses	LXY x, y	(X) ← x x = 0 to 15 (Y) ← y y = 0 to 15	66, 88
	LZ z	(Z) ← z z = 0 to 3	66, 88
	INY	(Y) ← (Y) + 1	66, 88
	DEY	(Y) ← (Y) – 1	63, 88
RAM to register transfer	TAM j	(A) ← (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15	78, 88
	XAM j	(A) ←→ (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15	86, 88
	XAMD j	(A) ←→ (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) – 1	87, 88

Grouping	Mnemonic	Function	Page
RAM to register transfer	XAMI j	(A) ←→ (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) + 1	87, 88
	TMA j	(M(DP)) ← (A) (X) ← (X)EXOR(j) j = 0 to 15	83, 88
Arithmetic operation	LA n	(A) ← n n = 0 to 15	66, 90
	TABP p	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p (Note) (PC_L) ← (DR_2–DR_0, A_3–A_0) (B) ← $(ROM(PC))_{7–4}$ (A) ← $(ROM(PC))_{3–0}$ (PC) ← (SK(SP)) (SP) ← (SP) – 1	76, 90
	AM	(A) ← (A) + (M(DP))	60, 90
	AMC	(A) ← (A) + (M(DP)) + (CY) (CY) ← Carry	60, 90
	A n	(A) ← (A) + n n = 0 to 15	60, 90
	AND	(A) ← (A) AND (M(DP))	61, 90
	OR	(A) ← (A) OR (M(DP))	68, 90
	SC	(CY) ← 1	71, 90
	RC	(CY) ← 0	69, 90
	SZC	(CY) = 0 ?	74, 90
	CMA	(A) ← $(\overline{A})$	63, 90
	RAR	→ CY → $A_3A_2A_1A_0$ → (back to CY)	68, 90

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
Bit operation	SB j	(Mj(DP)) ← 1 j = 0 to 3	70, 90
	RB j	(Mj(DP)) ← 0 j = 0 to 3	69, 90
	SZB j	(Mj(DP)) = 0 ? j = 0 to 3	74, 90
Comparison operation	SEAM	(A) = (M(DP)) ?	72, 90
	SEA n	(A) = n ? n = 0 to 15	71, 90
Branch operation	B a	(PC_L) ← a_6–a_0	61, 92
	BL p, a	(PC_H) ← p (Note) (PC_L) ← a_6–a_0	61, 92
	BLA p	(PC_H) ← p (Note) (PC_L) ← (DR_2–DR_0, A_3–A_0)	61, 92
Subroutine operation	BM a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← 2 (PC_L) ← a_6–a_0	62, 92
	BML p, a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p (Note) (PC_L) ← a_6–a_0	62, 92
	BMLA p	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p (Note) (PC_L) ← (DR_2–DR_0, A_3–A_0)	62, 92
Return operation	RTI	(PC) ← (SK(SP)) (SP) ← (SP) – 1	70, 92
	RT	(PC) ← (SK(SP)) (SP) ← (SP) – 1	70, 92
	RTS	(PC) ← (SK(SP)) (SP) ← (SP) – 1	70, 92

Grouping	Mnemonic	Function	Page
Interrupt operation	DI	(INTE) ← 0	64, 94
	EI	(INTE) ← 1	64, 94
	SNZ0	$V1_0$ = 0: (EXF0) = 1 ? After skipping, (EXF0) ← 0 $V1_0$ = 1: SNZ0 = NOP	72, 94
	SNZI0	$I1_2$ = 1 : (INT) = "H" ? $I1_2$ = 0 : (INT) = "L" ?	73, 94
	TAV1	(A) ← (V1)	79, 94
	TV1A	(V1) ← (A)	85, 94
	TAV2	(A) ← (V2)	79, 94
	TV2A	(V2) ← (A)	85, 94
	TAI1	(A) ← (I1)	77, 94
	TI1A	(I1) ← (A)	82, 94
Timer operation	TAW1	(A) ← (W1)	80, 94
	TW1A	(W1) ← (A)	85, 94
	TAW2	(A) ← (W2)	80, 94
	TW2A	(W2) ← (A)	85, 94
	TAW6	(A) ← (W6)	80, 94
	TW6A	(W6) ← (A)	86, 94
	TAB1	(B) ← ($T1_7$–$T1_4$) (A) ← ($T1_3$–$T1_0$)	75, 94
	T1AB	($R1_7$–$R1_4$) ← (B) ($T1_7$–$T1_4$) ← (B) ($R1_3$–$R1_0$) ← (A) ($T1_3$–$T1_0$) ← (A)	74, 94
	TAB2	(B) ← ($T2_7$–$T2_4$) (A) ← ($T2_3$–$T2_0$)	75, 94
	T2AB	($R2_7$–$R2_4$) ← (B) ($T2_7$–$T2_4$) ← (B) ($R2_3$–$R2_0$) ← (A) ($T2_3$–$T2_0$) ← (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
Timer operation	TR1AB	$(R1_7–R1_4) \leftarrow (B)$ $(R1_3–R1_0) \leftarrow (A)$	84, 94
	SNZT1	$V1_2 = 0$: (T1F) = 1 ? After skipping, (T1F) ← 0 $V1_2 = 1$: SNZT1 = NOP	73, 94
	SNZT2	$V1_3 = 0$: (T2F) = 1 ? After skipping, (T2F) ← 0 $V1_3 = 1$: SNZT2 = NOP	73, 94
Input/Output operation	IAP0	(A) ← (P0)	65, 96
	OP0A	(P0) ← (A)	67, 96
	IAP1	(A) ← (P1)	65, 96
	OP1A	(P1) ← (A)	67, 96
	IAP2	$(A_1, A_0) \leftarrow (P2_1, P2_0)$ $(A_3, A_2) \leftarrow 0$	65, 96
	OP2A	$(P2_1, P2_0) \leftarrow (A_1, A_0)$	68, 96
	CLD	(D) ← 1	62, 96
	RD	(D(Y)) ← 0 (Y) = 0 to 3	69, 96
	SD	(D(Y)) ← 1 (Y) = 0 to 3	71, 96
	SZD	(D(Y)) = 0 ? (Y) = 0 to 3	74, 96
	SCP	(C) ← 1	71, 96
	RCP	(C) ← 0	69, 96
	SNZCP	(C) = 1 ?	72, 96
Input/Output operation	IAK	$(A_0) \leftarrow (K)$ $(A_3–A_1) \leftarrow 0$	65, 96
	OKA	$(K) \leftarrow (A_0)$	67, 96
	TK0A	(K0) ← (A)	82, 96
	TAK0	(A) ← (K0)	77, 96
	TK1A	(K1) ← (A)	82, 96
	TAK1	(A) ← (K1)	77, 96
	TK2A	(K2) ← (A)	83, 96
	TAK2	(A) ← (K2)	78, 96
	TPU0A	(PU0) ← (A)	83, 96
	TPU1A	(PU1) ← (A)	84, 96
	TPU2A	(PU2) ← (A)	84, 96
A-D conversion operation	TABAD	In A-D conversion mode ($Q1_3 = 0$), $(B) \leftarrow (AD_9–AD_6)$ $(A) \leftarrow (AD_5–AD_2)$ In comparator mode ($Q1_3 = 1$), $(B) \leftarrow (AD_7–AD_4)$ $(A) \leftarrow (AD_3–AD_0)$	76, 98
	TALA	$(A_3, A_2) \leftarrow (AD_1, AD_0)$ $(A_1, A_0) \leftarrow 0$	78, 98
	TADAB	$(AD_7–AD_4) \leftarrow (B)$ $(AD_3–AD_0) \leftarrow (A)$	77, 98
	TAQ1	(A) ← (Q1)	79, 98
	TQ1A	(Q1) ← (A)	84, 98
	ADST	(ADF) ← 0 $Q1_3 = 0$: A-D conversion starting $Q1_3 = 1$: Comparator operation starting	60, 98
	SNZAD	$V2_2 = 0$: (ADF) = 1 ? After skipping, (ADF) ← 0 $V2_2 = 1$: SNZAD = NOP	72, 98

INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function	Page
Other operation	NOP	(PC) ← (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) ← 0	86, 98
	CMCK	Ceramic oscillation circuit selected	63, 98
	CRCK	RC oscillation circuit selected	63, 98
	TAMR	(A) ← (MR)	78, 98
	TMRA	(MR) ← (A)	83, 98

MACHINE INSTRUCTIONS (INDEX BY ALPHABET)

A n (Add n and accumulator)

Instruction code D9 … D0: 0 0 0 1 1 0 n n n n (2) — 0 6 n (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	Overflow = 0

Operation: (A) ← (A) + n
n = 0 to 15

Grouping: Arithmetic operation

Description: Adds the value n in the immediate field to register A, and stores a result in register A.
The contents of carry flag CY remains unchanged.
Skips the next instruction when there is no overflow as the result of operation.
Executes the next instruction when there is overflow as the result of operation.

ADST (A-D conversion STart)

Instruction code D9 … D0: 1 0 1 0 0 1 1 1 1 1 (2) — 2 9 F (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (ADF) ← 0
$Q1_3$ = 0: A-D conversion starting
$Q1_3$ = 1: Comparator operation starting
($Q1_3$: bit 3 of A-D control register Q1)

Grouping: A-D conversion operation

Description: Clears (0) to A-D conversion completion flag ADF, and the A-D conversion at the A-D conversion mode ($Q1_3$ = 0) or the comparator operation at the comparator mode ($Q1_3$ = 1) is started.

AM (Add accumulator and Memory)

Instruction code D9 … D0: 0 0 0 0 0 0 1 0 1 0 (2) — 0 0 A (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (A) + (M(DP))

Grouping: Arithmetic operation

Description: Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged.

AMC (Add accumulator, Memory and Carry)

Instruction code D9 … D0: 0 0 0 0 0 0 1 0 1 1 (2) — 0 0 B (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	0/1	–

Operation: (A) ← (A) + (M(DP)) + (CY)
(CY) ← Carry

Grouping: Arithmetic operation

Description: Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

AND (logical AND between accumulator and memory)

Instruction code

D9									D0					
0	0	0	0	0	1	1	0	0	0	2	0	1	8	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (A) AND (M(DP))

Grouping: Arithmetic operation

Description: Takes the AND operation between the contents of register A and the contents of M(DP), and stores the result in register A.

B a (Branch to address a)

Instruction code

D9									D0					
0	1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	1	8 +a	a	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (PC_L) ← a_6 to a_0

Grouping: Branch operation

Description: Branch within a page : Branches to address a in the identical page.

Note: Specify the branch address within the page including this instruction.

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

Instruction code

D9									D0					
0	0	1	1	1	p_4	p_3	p_2	p_1	p_0	2	0	E +p	p	16
1	0	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2	a	a	16

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Operation: (PC_H) ← p
(PC_L) ← a_6 to a_0

Grouping: Branch operation

Description: Branch out of a page : Branches to address a in page p.

Note: 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					
0	0	0	0	0	1	0	0	0	0	2	0	1	0	16
1	0	0	p_4	0	0	p_3	p_2	p_1	p_0	2	2	p	p	16

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Operation: (PC_H) ← p
(PC_L) ← (DR_2–DR_0, A_3–A_0)

Grouping: Branch operation

Description: Branch out of a page : Branches to address (DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0)$_2$ specified by registers D and A in page p.

Note: 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					
0	1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	1	a	a	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
(SP) ← (SP) + 1
(SK(SP)) ← (PC)
(PC_H) ← 2
(PC_L) ← a_6–a_0

Grouping: Subroutine call operation

Description: Call the subroutine in page 2 : Calls the subroutine at address a in page 2.

Note: Subroutine extending from page 2 to another page can also be called with the BM instruction when it starts on page 2.
Be careful not to over the stack because the maximum level of subroutine nesting is 8.

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

Instruction code

D9									D0					
0	0	1	1	0	p_4	p_3	p_2	p_1	p_0	2	0	C +p	p	16
1	0	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2	a	a	16

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Operation:
(SP) ← (SP) + 1
(SK(SP)) ← (PC)
(PC_H) ← p
(PC_L) ← a_6–a_0

Grouping: Subroutine call operation

Description: Call the subroutine : Calls the subroutine at address a in page p.

Note: 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					
0	0	0	0	1	1	0	0	0	0	2	0	3	0	16
1	0	0	p_4	0	0	p_3	p_2	p_1	p_0	2	2	p	p	16

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Operation:
(SP) ← (SP) + 1
(SK(SP)) ← (PC)
(PC_H) ← p
(PC_L) ← (DR_2–DR_0, A_3–A_0)

Grouping: Subroutine call operation

Description: Call the subroutine : Calls the subroutine at address (DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0)$_2$ specified by registers D and A in page p.

Note: 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					
0	0	0	0	0	1	0	0	0	1	2	0	1	1	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (D) ← 1

Grouping: Input/Output operation

Description: Sets (1) to port D.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

CMA (CoMplement of Accumulator)

Instruction code D_9 … D_0: 0 0 0 0 0 1 1 1 0 0_2 = 0 1 C_{16}

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(A) \leftarrow \overline{(A)}$

Grouping: Arithmetic operation

Description: Stores the one's complement for register A's contents in register A.

CMCK (Clock select: ceraMic oscillation ClocK)

Instruction code D_9 … D_0: 1 0 1 0 0 1 1 0 1 0_2 = 2 9 A_{16}

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: Ceramic oscillation circuit selected

Grouping: Other operation

Description: Selects the ceramic oscillation circuit and stops the ring oscillator.

CRCK (Clock select: Rc oscillation ClocK)

Instruction code D_9 … D_0: 1 0 1 0 0 1 1 0 1 1_2 = 2 9 B_{16}

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: RC oscillation circuit selected

Grouping: Other operation

Description: Selects the RC oscillation circuit and stops the ring oscillator.

DEY (DEcrement register Y)

Instruction code D_9 … D_0: 0 0 0 0 0 1 0 1 1 1_2 = 0 1 7_{16}

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 15

Operation: $(Y) \leftarrow (Y) - 1$

Grouping: RAM addresses

Description: Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped. When the contents of register Y is not 15, the next instruction is executed.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

DI (Disable Interrupt)

Instruction code D9 … D0: 0 0 0 0 0 0 0 1 0 0 (binary) = 0 0 4 (hex, $_{16}$)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (INTE) ← 0

Grouping: Interrupt control operation

Description: Clears (0) to interrupt enable flag INTE, and disables the interrupt.

Note: Interrupt is disabled by executing the DI instruction after executing 1 machine cycle.

DWDT (Disable WatchDog Timer)

Instruction code D9 … D0: 1 0 1 0 0 1 1 1 0 0 (binary) = 2 9 C (hex, $_{16}$)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: Stop of watchdog timer function enabled

Grouping: Other operation

Description: Stops the watchdog timer function by the WRST instruction after executing the DWDT instruction.

EI (Enable Interrupt)

Instruction code D9 … D0: 0 0 0 0 0 0 0 1 0 1 (binary) = 0 0 5 (hex, $_{16}$)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (INTE) ← 1

Grouping: Interrupt control operation

Description: Sets (1) to interrupt enable flag INTE, and enables the interrupt.

Note: Interrupt is enabled by executing the EI instruction after executing 1 machine cycle.

EPOF (Enable POF instruction)

Instruction code D9 … D0: 0 0 0 1 0 1 1 0 1 1 (binary) = 0 5 B (hex, $_{16}$)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: POF2 instruction valid

Grouping: Other operation

Description: Makes the immediate after POF or POF2 instruction valid by executing the EPOF instruction.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

IAK (Input Accumulator from port K)

Instruction code	D9 … D0	Hex	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	–	–

Operation: $(A_0) \leftarrow (K)$
$(A_3–A_1) \leftarrow 0$

Grouping: Input/Output operation

Description: Transfers the contents of port K to the bit 0 (A_0) of register A.

Note: After this instruction is executed, "0" is stored to the high-order 3 bits ($A_3–A_1$) of register A.

IAP0 (Input Accumulator from port P0)

Instruction code	D9 … D0	Hex	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 1 0 0 0 0 0 (2)	2 6 0 (16)	1	1	–	–

Operation: $(A) \leftarrow (P0)$

Grouping: Input/Output operation

Description: Transfers the input of port P0 to register A.

IAP1 (Input Accumulator from port P1)

Instruction code	D9 … D0	Hex	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 1 0 0 0 0 1 (2)	2 6 1 (16)	1	1	–	–

Operation: $(A) \leftarrow (P1)$

Grouping: Input/Output operation

Description: Transfers the input of port P1 to register A.

IAP2 (Input Accumulator from port P2)

Instruction code	D9 … D0	Hex	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	–	–

Operation: $(A_1, A_0) \leftarrow (P2_1, P2_0)$
$(A_3, A_2) \leftarrow 0$

Grouping: Input/Output operation

Description: Transfers the input of port P2 to the low-order 2 bits (A_1, A_0) of register A.

Note: After this instruction is executed, "0" is stored to the high-order 2 bits (A_3, A_2) of register A.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

INY (INcrement register Y)

Instruction code

D9									D0					
0	0	0	0	0	1	0	0	1	1	2	0	1	3	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 0

Operation: (Y) ← (Y) + 1

Grouping: RAM addresses

Description: Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. When the contents of register Y is not 0, the next instruction is executed.

LA n (Load n in Accumulator)

Instruction code

D9									D0					
0	0	0	1	1	1	n	n	n	n	2	0	7	n	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	Continuous description

Operation: (A) ← n
n = 0 to 15

Grouping: Arithmetic operation

Description: Loads the value n in the immediate field to register A.
When the LA instructions are continuously coded and executed, only the first LA instruction is executed and other LA instructions coded continuously are skipped.

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

Instruction code

D9									D0					
1	1	x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0	2	3	x	y	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	Continuous description

Operation: (X) ← x x = 0 to 15
(Y) ← y y = 0 to 15

Grouping: RAM addresses

Description: Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y. When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped.

LZ z (Load register Z with z)

Instruction code

D9									D0					
0	0	0	1	0	0	1	0	z_1	z_0	2	0	4	8 +z	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (Z) ← z z = 0 to 3

Grouping: RAM addresses

Description: Loads the value z in the immediate field to register Z.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

NOP (No OPeration)

Instruction code D9–D0: 0 0 0 0 0 0 0 0 0 0 ₂ = 0 0 0 ₁₆

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (PC) ← (PC) + 1

Grouping: Other operation

Description: No operation; Adds 1 to program counter value, and others remain unchanged.

OKA (Output port K from Accumulator)

Instruction code D9–D0: 1 0 0 0 0 1 1 1 1 1 ₂ = 2 1 F ₁₆

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (K) ← (A_0)

Grouping: Input/Output operation

Description: Outputs the contents of bit 0 (A_0) of register A to port K.

OP0A (Output port P0 from Accumulator)

Instruction code D9–D0: 1 0 0 0 1 0 0 0 0 0 ₂ = 2 2 0 ₁₆

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (P0) ← (A)

Grouping: Input/Output operation

Description: Outputs the contents of register A to port P0.

OP1A (Output port P1 from Accumulator)

Instruction code D9–D0: 1 0 0 0 1 0 0 0 0 1 ₂ = 2 2 1 ₁₆

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (P1) ← (A)

Grouping: Input/Output operation

Description: Outputs the contents of register A to port P1.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

OP2A (Output port P2 from Accumulator)

Instruction code D_9 … D_0: 1 0 0 0 1 0 0 0 1 0 $_2$ = 2 2 2 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(P2_1, P2_0) \leftarrow (A_1, A_0)$

Grouping: Input/Output operation

Description: Outputs the contents of the low-order 2 bits (A_1, A_0) of register A to port P2.

OR (logical OR between accumulator and memory)

Instruction code D_9 … D_0: 0 0 0 0 0 1 1 0 0 1 $_2$ = 0 1 9 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(A) \leftarrow (A)$ OR $(M(DP))$

Grouping: Arithmetic operation

Description: Takes the OR operation between the contents of register A and the contents of M(DP), and stores the result in register A.

POF2 (Power OFf2)

Instruction code D_9 … D_0: 0 0 0 0 0 0 1 0 0 0 $_2$ = 0 0 8 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: RAM back-up

Grouping: Other operation

Description: Puts the system in RAM back-up state by executing the POF2 instruction after executing the EPOF instruction. Operations of all functions are stopped.

Note: 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 D_9 … D_0: 0 0 0 0 0 1 1 1 0 1 $_2$ = 0 1 D $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	0/1	–

Operation: → CY → $A_3A_2A_1A_0$ ┐ (rotate, A_0 back to CY)

Grouping: Arithmetic operation

Description: 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					
0	0	0	1	0	0	1	1	j	j	2	0	4	C+j	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (Mj(DP)) ← 0
j = 0 to 3

Grouping: Bit operation

Description: Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).

RC (Reset Carry flag)

Instruction code

D9									D0					
0	0	0	0	0	0	0	1	1	0	2	0	0	6	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	0	–

Operation: (CY) ← 0

Grouping: Arithmetic operation

Description: Clears (0) to carry flag CY.

RCP (Reset Port C)

Instruction code

D9									D0					
1	0	1	0	0	0	1	1	0	0	2	2	8	C	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (C) ← 0

Grouping: Input/Output operation

Description: Clears (0) to port C.

RD (Reset port D specified by register Y)

Instruction code

D9									D0					
0	0	0	0	0	1	0	1	0	0	2	0	1	4	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (D(Y)) ← 0
However,
(Y) = 0 to 3

Grouping: Input/Output operation

Description: Clears (0) to a bit of port D specified by register Y.

Note: Set 0 to 3 to register Y because port D is four ports (D_0–D_3).
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: 0 0 0 1 0 0 0 1 0 0 ($_2$) = 0 4 4 ($_{16}$)

Number of words	Number of cycles	Flag CY	Skip condition
1	2	–	–

Operation: (PC) ← (SK(SP))
(SP) ← (SP) – 1

Grouping: Return operation

Description: Returns from subroutine to the routine called the subroutine.

RTI (ReTurn from Interrupt)

Instruction code D9–D0: 0 0 0 1 0 0 0 1 1 0 ($_2$) = 0 4 6 ($_{16}$)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (PC) ← (SK(SP))
(SP) ← (SP) – 1

Grouping: Return operation

Description: Returns from interrupt service routine to main routine.
Returns each value of data pointer (X, Y, Z), carry flag, skip status, NOP mode status by the continuous description of the LA/LXY instruction, register A and register B to the states just before interrupt.

RTS (ReTurn from subroutine and Skip)

Instruction code D9–D0: 0 0 0 1 0 0 0 1 0 1 ($_2$) = 0 4 5 ($_{16}$)

Number of words	Number of cycles	Flag CY	Skip condition
1	2	–	Skip at uncondition

Operation: (PC) ← (SK(SP))
(SP) ← (SP) – 1

Grouping: Return operation

Description: Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition.

SB j (Set Bit)

Instruction code D9–D0: 0 0 0 1 0 1 1 1 j j ($_2$) = 0 5 C+j ($_{16}$)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (Mj(DP)) ← 0
j = 0 to 3

Grouping: Bit operation

Description: Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

SC (Set Carry flag)

Instruction code D_9 … D_0

0	0	0	0	0	0	0	1	1	1	$_2$	0	0	7 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	1	–

Operation: (CY) ← 1

Grouping: Arithmetic operation

Description: Sets (1) to carry flag CY.

SCP (Set Port C)

Instruction code D_9 … D_0

1	0	1	0	0	0	1	1	0	1	$_2$	2	8	D $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (C) ← 1

Grouping: Input/Output operation

Description: Sets (1) to port C.

SD (Set port D specified by register Y)

Instruction code D_9 … D_0

0	0	0	0	0	1	0	1	0	1	$_2$	0	1	5 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (D(Y)) ← 1
(Y) = 0 to 3

Grouping: Input/Output operation

Description: Sets (1) to a bit of port D specified by register Y.

Note: Set 0 to 3 to register Y because port D is four ports (D_0–D_3).
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_9 … D_0

0	0	0	0	1	0	0	1	0	1	$_2$	0	2	5 $_{16}$
0	0	0	1	1	1	n	n	n	n	$_2$	0	7	n $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	(A) = n

Operation: (A) = n ?
n = 0 to 15

Grouping: Comparison operation

Description: Skips the next instruction when the contents of register A is equal to the value n in the immediate field.
Executes the next instruction when the contents of register A is not equal to the value n in the immediate field.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

SEAM (Skip Equal, Accumulator with Memory)

Instruction code (D_9 … D_0): 0 0 0 0 1 0 0 1 1 0 $_2$ | 0 2 6 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(A) = (M(DP))

Operation: (A) = (M(DP)) ?

Grouping: Comparison operation

Description: Skips the next instruction when the contents of register A is equal to the contents of M(DP).
Executes the next instruction when the contents of register A is not equal to the contents of M(DP).

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

Instruction code (D_9 … D_0): 0 0 0 0 1 1 1 0 0 0 $_2$ | 0 3 8 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	$V1_0$ = 0: (EXF0) = 1

Operation: $V1_0$ = 0: (EXF0) = 1 ?
After skipping, (EXF0) ← 0
$V1_0$ = 1: SNZ0 = NOP
($V1_0$: bit 0 of the interrupt control register V1)

Grouping: Interrupt operation

Description: When $V1_0$ = 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 $V1_0$ = 1 : This instruction is equivalent to the NOP instruction.

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

Instruction code (D_9 … D_0): 1 0 1 0 0 0 0 1 1 1 $_2$ | 2 8 7 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	$V2_2$ = 0: (ADF) = 1

Operation: $V2_2$ = 0: (ADF) = 1 ?
After skipping, (ADF) ← 0
$V2_2$ = 1: SNZAD = NOP
($V2_2$: bit 2 of the interrupt control register V2)

Grouping: A-D conversion operation

Description: When $V2_2$ = 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 $V2_2$ = 1 : This instruction is equivalent to the NOP instruction.

SNZCP (Skip if Non Zero condition of Port C)

Instruction code (D_9 … D_0): 1 0 1 0 0 0 1 0 0 1 $_2$ | 2 8 9 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(C) = 1

Operation: (C) = 1 ?

Grouping: Input/Output operation

Description: 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)

SNZI0 (Skip if Non Zero condition of external 0 Interrupt input pin)

Instruction code D9 … D0: 0 0 0 0 1 1 1 0 1 0 (binary) = 0 3 A (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	$I1_2$ = 0 : (INT) = “L” $I1_2$ = 1 : (INT) = “H”

Operation: $I1_2$ = 0 : (INT) = “L” ?
$I1_2$ = 1 : (INT) = “H” ?
($I1_2$: bit 2 of the interrupt control register I1)

Grouping: Interrupt operation

Description: When $I1_2$ = 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 $I1_2$ = 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: 0 0 0 0 0 0 0 0 1 1 (binary) = 0 0 3 (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(P) = 1

Operation: (P) = 1 ?

Grouping: Other operation

Description: Skips the next instruction when the P flag is “1”.
After skipping, the P flag remains unchanged.
Executes the next instruction when the P flag is “0.”

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

Instruction code D9 … D0: 1 0 1 0 0 0 0 0 0 0 (binary) = 2 8 0 (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	$V1_2$ = 0: (T1F) = 1

Operation: $V1_2$ = 0: (T1F) = 1 ?
After skipping, (T1F) ← 0
$V1_2$ = 1: SNZT1 = NOP
($V1_2$ = bit 2 of interrupt control register V1)

Grouping: Timer operation

Description: When $V1_2$ = 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 $V1_2$ = 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: 1 0 1 0 0 0 0 0 0 1 (binary) = 2 8 1 (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	$V1_3$ = 0: (T2F) = 1

Operation: $V1_3$ = 0: (T2F) = 1 ?
After skipping, (T2F) ← 0
$V1_3$ = 1: SNZT2 = NOP
($V1_3$ = bit 3 of interrupt control register V1)

Grouping: Timer operation

Description: When $V1_3$ = 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 $V1_3$ = 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					
0	0	0	0	1	0	0	0	j	j	2	0	2	j	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Mj(DP)) = 0 j = 0 to 3

Operation: (Mj(DP)) = 0 ?
j = 0 to 3

Grouping: Bit operation

Description: Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is “0.”
Executes the next instruction when the contents of bit j of M(DP) is “1.”

SZC (Skip if Zero, Carry flag)

Instruction code

D9									D0					
0	0	0	0	1	0	1	1	1	1	2	0	2	F	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(CY) = 0

Operation: (CY) = 0 ?

Grouping: Arithmetic operation

Description: Skips the next instruction when the contents of carry flag CY is “0.”
After skipping, the CY flag remains unchanged.
Executes the next instruction when the contents of the CY flag is “1.“

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

Instruction code

D9									D0					
0	0	0	0	1	0	0	1	0	0	2	0	2	4	16
0	0	0	0	1	0	1	0	1	1	2	0	2	B	16

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	(D(Y)) = 0 (Y) = 0 to 3

Operation: (D(Y)) = 0 ?
(Y) = 0 to 3

Grouping: Input/Output operation

Description: Skips the next instruction when a bit of port D specified by register Y is “0.” Executes the next instruction when the bit is “1.”

Note: Set 0 to 3 to register Y because port D is four ports (D_0–D_3).
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					
1	0	0	0	1	1	0	0	0	0	2	2	3	0	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: ($T1_7$–$T1_4$) ← (B)
($R1_7$–$R1_4$) ← (B)
($T1_3$–$T1_0$) ← (A)
($R1_3$–$R1_0$) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits of timer 1 and timer 1 reload register R1. Transfers the contents of register A to the low-order 4 bits of timer 1 and timer 1 reload register R1.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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

Instruction code

D9									D0					
1	0	0	0	1	1	0	0	0	1	2	2	3	1	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
$(T2_7–T2_4) \leftarrow (B)$
$(R2_7–R2_4) \leftarrow (B)$
$(T2_3–T2_0) \leftarrow (A)$
$(R2_3–R2_0) \leftarrow (A)$

Grouping: Timer operation

Description: 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					
0	0	0	0	0	1	1	1	1	0	2	0	1	E	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(A) \leftarrow (B)$

Grouping: Other operation

Description: Transfers the contents of register B to register A.

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

Instruction code

D9									D0					
1	0	0	1	1	1	0	0	0	0	2	2	7	0	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
$(B) \leftarrow (T1_7–T1_4)$
$(A) \leftarrow (T1_3–T1_0)$

Grouping: Timer operation

Description: Transfers the high-order 4 bits ($T1_7–T1_4$) of timer 1 to register B.
Transfers the low-order 4 bits ($T1_3–T1_0$) of timer 1 to register A.

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

Instruction code

D9									D0					
1	0	0	1	1	1	0	0	0	1	2	2	7	1	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
$(B) \leftarrow (T2_7–T2_4)$
$(A) \leftarrow (T2_3–T2_0)$

Grouping: Timer operation

Description: Transfers the high-order 4 bits ($T2_7–T2_4$) of timer 2 to register B.
Transfers the low-order 4 bits ($T2_3–T2_0$) 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					
1	0	0	1	1	1	1	0	0	1	2	2	7	9	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: In A-D conversion mode ($Q1_3$ = 0),
(B) ← (AD_9–AD_6)
(A) ← (AD_5–AD_2)
In comparator mode ($Q1_3$ = 1),
(B) ← (AD_7–AD_4)
(A) ← (AD_3–AD_0)
($Q1_3$: bit 3 of A-D control register Q1)

Grouping: A-D conversion operation

Description: In the A-D conversion mode ($Q1_3$ = 0), transfers the high-order 4 bits (AD_9–AD_6) of register AD to register B, and the middle-order 4 bits (AD_5–AD_2) of register AD to register A. In the comparator mode ($Q1_3$ = 1), transfers the middle-order 4 bits (AD_7–AD_4) of register AD to register B, and the low-order 4 bits (AD_3–AD_0) of register AD to register A.

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

Instruction code

D9									D0					
0	0	0	0	1	0	1	0	1	0	2	0	2	A	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (B) ← (E_7–E_4)
(A) ← (E_3–E_0)

Grouping: Register to register transfer

Description: Transfers the high-order 4 bits (E_7–E_4) 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					
0	0	1	0	0	p_4	p_3	p_2	p_1	p_0	2	0	8 +p	p	16

Number of words	Number of cycles	Flag CY	Skip condition
1	3	–	–

Operation: (SP) ← (SP) + 1
(SK(SP)) ← (PC)
(PC_H) ← p
(PC_L) ← (DR_2–DR_0, A_3–A_0)
(B) ← $(ROM(PC))_{7-4}$
(A) ← $(ROM(PC))_{3-0}$
(PC) ← (SK(SP))
(SP) ← (SP) – 1

Grouping: Arithmetic operation

Description: Transfers bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 7 to 0 are the ROM pattern in address $(DR_2\ DR_1\ DR_0\ A_3\ A_2\ A_1\ A_0)_2$ specified by registers A and D in page p.

Note: p is 0 to 15 for M34506M2, and p is 0 to 31 for M34506M4/E4.
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					
0	0	0	1	0	1	0	0	0	1	2	0	5	1	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A_2–A_0) ← (DR_2–DR_0)
(A_3) ← 0

Grouping: Register to register transfer

Description: Transfers the contents of register D to the low-order 3 bits (A_2–A_0) of register A.

Note: When this instruction is executed, "0" is stored to the bit 3 (A_3) of register A.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TADAB (Transfer data to register AD from Accumulator from register B)

Instruction code: D9–D0: 1 0 0 0 1 1 1 0 0 1 (binary) = 2 3 9 (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(AD_7–AD_4) \leftarrow (B)$
$(AD_3–AD_0) \leftarrow (A)$

Grouping: A-D conversion operation

Description: In the A-D conversion mode ($Q1_3$ = 0), this instruction is equivalent to the NOP instruction. In the comparator mode ($Q1_3$ = 1), transfers the contents of register B to the high-order 4 bits ($AD_7–AD_4$) of comparator register, and the contents of register A to the low-order 4 bits ($AD_3–AD_0$) of comparator register.
($Q1_3$ = bit 3 of A-D control register Q1)

TAI1 (Transfer data to Accumulator from register I1)

Instruction code: D9–D0: 1 0 0 1 0 1 0 0 1 1 (binary) = 2 5 3 (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(A) \leftarrow (I1)$

Grouping: Interrupt operation

Description: Transfers the contents of interrupt control register I1 to register A.

TAK0 (Transfer data to Accumulator from register K0)

Instruction code: D9–D0: 1 0 0 1 0 1 0 1 1 0 (binary) = 2 5 6 (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(A) \leftarrow (K0)$

Grouping: Input/Output operation

Description: 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: 1 0 0 1 0 1 1 0 0 1 (binary) = 2 5 9 (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(A) \leftarrow (K1)$

Grouping: Input/Output operation

Description: 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

1	0	0	1	0	1	1	0	1	0

$_2$ = 2 5 A $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (K2)

Grouping: Input/Output operation

Description: 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

1	0	0	1	0	0	1	0	0	1

$_2$ = 2 4 9 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A_3, A_2) ← (AD_1, AD_0)
(A_1, A_0) ← 0

Grouping: A-D conversion operation

Description: Transfers the low-order 2 bits (AD_1, AD_0) of register AD to the high-order 2 bits (A_3, A_2) of register A.

Note: After this instruction is executed, “0” is stored to the low-order 2 bits (A_1, A_0) of register A.

TAM j (Transfer data to Accumulator from Memory)

Instruction code D9 … D0

1	0	1	1	0	0	j	j	j	j

$_2$ = 2 C j $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 15

Grouping: RAM to register transfer

Description: After transferring the contents of M(DP) to register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.

TAMR (Transfer data to Accumulator from register MR)

Instruction code D9 … D0

1	0	0	1	0	1	0	0	1	0

$_2$ = 2 5 2 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (MR)

Grouping: Other operation

Description: 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: 1 0 0 1 0 0 0 1 0 0 (2) = 2 4 4 (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (Q1)

Grouping: A-D conversion operation

Description: Transfers the contents of A-D control register Q1 to register A.

TASP (Transfer data to Accumulator from Stack Pointer)

Instruction code D9–D0: 0 0 0 1 0 1 0 0 0 0 (2) = 0 5 0 (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A_2–A_0) ← (SP_2–SP_0)
(A_3) ← 0

Grouping: Register to register transfer

Description: Transfers the contents of stack pointer (SP) to the low-order 3 bits (A_2–A_0) of register A.

Note: After this instruction is executed, “0” is stored to the bit 3 (A_3) of register A.

TAV1 (Transfer data to Accumulator from register V1)

Instruction code D9–D0: 0 0 0 1 0 1 0 1 0 0 (2) = 0 5 4 (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (V1)

Grouping: Interrupt operation

Description: Transfers the contents of interrupt control register V1 to register A.

TAV2 (Transfer data to Accumulator from register V2)

Instruction code D9–D0: 0 0 0 1 0 1 0 1 0 1 (2) = 0 5 5 (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (V2)

Grouping: Interrupt operation

Description: 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

1	0	0	1	0	0	1	0	1	1	₂	2	4	B	₁₆

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (W1)

Grouping: Timer operation

Description: Transfers the contents of timer control register W1 to register A.

TAW2 (Transfer data to Accumulator from register W2)

Instruction code: D9 … D0

1	0	0	1	0	0	1	1	0	0	₂	2	4	C	₁₆

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (W2)

Grouping: Timer operation

Description: Transfers the contents of timer control register W2 to register A.

TAW6 (Transfer data to Accumulator from register W6)

Instruction code: D9 … D0

1	0	0	1	0	1	0	0	0	0	₂	2	5	0	₁₆

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (W6)

Grouping: Timer operation

Description: Transfers the contents of timer control register W6 to register A.

TAX (Transfer data to Accumulator from register X)

Instruction code: D9 … D0

0	0	0	1	0	1	0	0	1	0	₂	0	5	2	₁₆

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (X)

Grouping: Register to register transfer

Description: 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					
0	0	0	0	0	1	1	1	1	1	2	0	1	F	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (Y)

Grouping: Register to register transfer

Description: Transfers the contents of register Y to register A.

TAZ (Transfer data to Accumulator from register Z)

Instruction code

D9									D0					
0	0	0	1	0	1	0	0	1	1	2	0	5	3	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A_1, A_0) ← (Z_1, Z_0)
(A_3, A_2) ← 0

Grouping: Register to register transfer

Description: Transfers the contents of register Z to the low-order 2 bits (A_1, A_0) of register A.

Note: After this instruction is executed, “0” is stored to the high-order 2 bits (A_3, A_2) of register A.

TBA (Transfer data to register B from Accumulator)

Instruction code

D9									D0					
0	0	0	0	0	0	1	1	1	0	2	0	0	E	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (B) ← (A)

Grouping: Register to register transfer

Description: Transfers the contents of register A to register B.

TDA (Transfer data to register D from Accumulator)

Instruction code

D9									D0					
0	0	0	0	1	0	1	0	0	1	2	0	2	9	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (DR_2–DR_0) ← (A_2–A_0)

Grouping: Register to register transfer

Description: Transfers the contents of the low-order 3 bits (A_2–A_0) 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 (D_9 … D_0)

0	0	0	0	0	1	1	0	1	0	$_2$	0	1	A $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(E_7–E_4) \leftarrow (B)$
$(E_3–E_0) \leftarrow (A)$

Grouping: Register to register transfer

Description: Transfers the contents of register B to the high-order 4 bits ($E_3–E_0$) of register E, and the contents of register A to the low-order 4 bits ($E_3–E_0$) of register E.

TI1A (Transfer data to register I1 from Accumulator)

Instruction code (D_9 … D_0)

1	0	0	0	0	1	0	1	1	1	$_2$	2	1	7 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(I1) \leftarrow (A)$

Grouping: Interrupt operation

Description: Transfers the contents of register A to interrupt control register I1.

TK0A (Transfer data to register K0 from Accumulator)

Instruction code (D_9 … D_0)

1	0	0	0	0	1	1	0	1	1	$_2$	2	1	B $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(K0) \leftarrow (A)$

Grouping: Input/Output operation

Description: Transfers the contents of register A to key-on wakeup control register K0.

TK1A (Transfer data to register K1 from Accumulator)

Instruction code (D_9 … D_0)

1	0	0	0	0	1	0	1	0	0	$_2$	2	1	4 $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(K1) \leftarrow (A)$

Grouping: Input/Output operation

Description: Transfers the contents of register A to key-on wakeup control register K1.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TK2A (Transfer data to register K2 from Accumulator)

Instruction code

D9									D0					
1	0	0	0	0	1	0	1	0	1	$_2$	2	1	5	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (K2) ← (A)

Grouping: Input/Output operation

Description: 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					
1	0	1	0	1	1	j	j	j	j	$_2$	2	B	j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (M(DP)) ← (A)
(X) ← (X)EXOR(j)
j = 0 to 15

Grouping: RAM to register transfer

Description: After transferring the contents of register A to M(DP), an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.

TMRA (Transfer data to register MR from Accumulator)

Instruction code

D9									D0					
1	0	0	0	0	1	0	1	1	0	$_2$	2	1	6	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (MR) ← (A)

Grouping: Other operation

Description: Transfers the contents of register A to clock control register MR.

TPU0A (Transfer data to register PU0 from Accumulator)

Instruction code

D9									D0					
1	0	0	0	1	0	1	1	0	1	$_2$	2	2	D	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (PU0) ← (A)

Grouping: Input/Output operation

Description: 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					
1	0	0	0	1	0	1	1	1	0	2	2	2	E	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (PU1) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to pull-up control register PU1.

TPU2A (Transfer data to register PU2 from Accumulator)

Instruction code

D9									D0					
1	0	0	0	1	0	1	1	1	1	2	2	2	F	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (PU2) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to pull-up control register PU2.

TQ1A (Transfer data to register Q1 from Accumulator)

Instruction code

D9									D0					
1	0	0	0	0	0	0	1	0	0	2	2	0	4	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (Q1) ← (A)

Grouping: A-D conversion operation

Description: 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					
1	0	0	0	1	1	1	1	1	1	2	2	3	F	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(R1_7–R1_4) \leftarrow (B)$
$(R1_3–R1_0) \leftarrow (A)$

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits $(R1_7–R1_4)$ of reload register R1, and the contents of register A to the low-order 4 bits $(R1_3–R1_0)$ of reload register R1.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TV1A (Transfer data to register V1 from Accumulator)

Instruction code D9 – D0: 0 0 0 0 1 1 1 1 1 1 (2) = 0 3 F (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (V1) ← (A)

Grouping: Interrupt operation

Description: Transfers the contents of register A to interrupt control register V1.

TV2A (Transfer data to register V2 from Accumulator)

Instruction code D9 – D0: 0 0 0 0 1 1 1 1 1 0 (2) = 0 3 E (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (V2) ← (A)

Grouping: Interrupt operation

Description: Transfers the contents of register A to interrupt control register V2.

TW1A (Transfer data to register W1 from Accumulator)

Instruction code D9 – D0: 1 0 0 0 0 0 1 1 1 0 (2) = 2 0 E (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (W1) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register A to timer control register W1.

TW2A (Transfer data to register W2 from Accumulator)

Instruction code D9 – D0: 1 0 0 0 0 0 1 1 1 1 (2) = 2 0 F (16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (W2) ← (A)

Grouping: Timer operation

Description: 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: 1 0 0 0 0 1 0 0 1 1 (binary) = 2 1 3 (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (W6) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register A to timer control register W6.

TYA (Transfer data to register Y from Accumulator)

Instruction code D9 … D0: 0 0 0 0 0 0 1 1 0 0 (binary) = 0 0 C (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (Y) ← (A)

Grouping: Register to register transfer

Description: Transfers the contents of register A to register Y.

WRST (Watchdog timer ReSeT)

Instruction code D9 … D0: 1 0 1 0 1 0 0 0 0 0 (binary) = 2 A 0 (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(WDF1) = 1

Operation: (WDF1) = 1 ?
After skipping, (WDF1) ← 0

Grouping: Other operation

Description: 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: 1 0 1 1 0 1 j j j j (binary) = 2 D j (hex)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ←→ (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 15

Grouping: RAM to register transfer

Description: After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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

Instruction code D9 … D0: 1 0 1 1 1 1 j j j j (base 2) = 2 F j (base 16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 15

Operation:
(A) ←→ (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 15
(Y) ← (Y) – 1

Grouping: RAM to register transfer

Description: After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped. When the contents of register Y is not 15, the next instruction is executed.

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

Instruction code D9 … D0: 1 0 1 1 1 0 j j j j (base 2) = 2 E j (base 16)

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 0

Operation:
(A) ←→ (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 15
(Y) ← (Y) + 1

Grouping: RAM to register transfer

Description: After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. when the contents of register Y is not 0, the next instruction is executed.

MACHINE INSTRUCTIONS (INDEX BY TYPES)

Type of instructions \ Parameter	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	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	(E_7–E_4) ← (B) (E_3–E_0) ← (A)
	TABE	0	0	0	0	1	0	1	0	1	0	0 2 A	1	1	(B) ← (E_7–E_4) (A) ← (E_3–E_0)
	TDA	0	0	0	0	1	0	1	0	0	1	0 2 9	1	1	(DR_2–DR_0) ← (A_2–A_0)
	TAD	0	0	0	1	0	1	0	0	0	1	0 5 1	1	1	(A_2–A_0) ← (DR_2–DR_0) (A_3) ← 0
	TAZ	0	0	0	1	0	1	0	0	1	1	0 5 3	1	1	(A_1, A_0) ← (Z_1, Z_0) (A_3, A_2) ← 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	(A_2–A_0) ← (SP_2–SP_0) (A_3) ← 0
RAM addresses	LXY x, y	1	1	x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0	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	z_1	z_0	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

RENESAS
Renesas Technology Corp.

Skip condition	Carry flag CY	Datailed 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 (E_3–E_0) of register E, and the contents of register A to the low-order 4 bits (E_3–E_0) of register E.
–	–	Transfers the high-order 4 bits (E_7–E_4) 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 (A_2–A_0) of register A to register D.
–	–	Transfers the contents of register D to the low-order 3 bits (A_2–A_0) of register A.
–	–	Transfers the contents of register Z to the low-order 2 bits (A_1, A_0) 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 (A_2–A_0) 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 D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
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	p_4	p_3	p_2	p_1	p_0	0 8 p +p	1	3	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p (Note) (PC_L) ← (DR_2–DR_0, A_3–A_0) (B) ← $(ROM(PC))_{7–4}$ (A) ← $(ROM(PC))_{3–0}$ (PC) ← (SK(SP)) (SP) ← (SP) – 1
	AM	0	0	0	0	0	0	1	0	1	0	0 0 A	1	1	(A) ← (A) + (M(DP))
	AMC	0	0	0	0	0	0	1	0	1	1	0 0 B	1	1	(A) ← (A) + (M(DP)) +(CY) (CY) ← Carry
	A n	0	0	0	1	1	0	n	n	n	n	0 6 n	1	1	(A) ← (A) + n n = 0 to 15
	AND	0	0	0	0	0	1	1	0	0	0	0 1 8	1	1	(A) ← (A) AND (M(DP))
	OR	0	0	0	0	0	1	1	0	0	1	0 1 9	1	1	(A) ← (A) OR (M(DP))
	SC	0	0	0	0	0	0	0	1	1	1	0 0 7	1	1	(CY) ← 1
	RC	0	0	0	0	0	0	0	1	1	0	0 0 6	1	1	(CY) ← 0
	SZC	0	0	0	0	1	0	1	1	1	1	0 2 F	1	1	(CY) = 0 ?
	CMA	0	0	0	0	0	1	1	1	0	0	0 1 C	1	1	(A) ← $(\overline{A})$
	RAR	0	0	0	0	0	1	1	1	0	1	0 1 D	1	1	→ CY → $A_3A_2A_1A_0$ ┐ (rotate)
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.

Skip condition	Carry flag CY	Datailed description
Continuous description	–	Loads the value n in the immediate field to register A. When the LA instructions are continuously coded and executed, only the first LA instruction is executed and other LA instructions coded continuously are skipped.
–	–	Transfers bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 7 to 0 are the ROM pattern in address (DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0)$_2$ specified by registers A and D in page p. When this instruction is executed, be careful not to over the stack because 1 stage of stack register is used.
–	–	Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged.
–	0/1	Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY.
Overflow = 0	–	Adds the value n in the immediate field to register A, and stores a result in register A. The contents of carry flag CY remains unchanged. Skips the next instruction when there is no overflow as the result of operation. Executes the next instruction when there is overflow as the result of operation.
–	–	Takes the AND operation between the contents of register A and the contents of M(DP), and stores the result in register A.
–	–	Takes the OR operation between the contents of register A and the contents of M(DP), and stores the result in register A.
–	1	Sets (1) to carry flag CY.
–	0	Clears (0) to carry flag CY.
(CY) = 0	–	Skips the next instruction when the contents of carry flag CY is “0.”
–	–	Stores the one’s complement for register A’s contents in register A.
–	0/1	Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right.
–	–	Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).
–	–	Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).
(Mj(DP)) = 0 j = 0 to 3	–	Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is “0.” Executes the next instruction when the contents of bit j of M(DP) is “1.”
(A) = (M(DP))	–	Skips the next instruction when the contents of register A is equal to the contents of M(DP). Executes the next instruction when the contents of register A is not equal to the contents of M(DP).
(A) = n	–	Skips the next instruction when the contents of register A is equal to the value n in the immediate field. Executes the next instruction when the contents of register A is not equal to the value n in the immediate field.

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Type of instructions \ Parameter	Mnemonic	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
Branch operation	B a	0	1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 8 a +a	1	1	$(PC_L) \leftarrow a_6–a_0$
	BL p, a	0	0	1	1	1	p_4	p_3	p_2	p_1	p_0	0 E p +p	2	2	$(PC_H) \leftarrow p$ (Note) $(PC_L) \leftarrow a_6–a_0$
		1	0	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2 a a			
	BLA p	0	0	0	0	0	1	0	0	0	0	0 1 0	2	2	$(PC_H) \leftarrow p$ (Note) $(PC_L) \leftarrow (DR_2–DR_0, A_3–A_0)$
		1	0	0	p_4	0	0	p_3	p_2	p_1	p_0	2 p p			
Subroutine operation	BM a	0	1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 a a	1	1	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow 2$ $(PC_L) \leftarrow a_6–a_0$
	BML p, a	0	0	1	1	0	p_4	p_3	p_2	p_1	p_0	0 C p +p	2	2	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow p$ (Note) $(PC_L) \leftarrow a_6–a_0$
		1	0	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2 a a			
	BMLA p	0	0	0	0	1	1	0	0	0	0	0 3 0	2	2	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow p$ (Note) $(PC_L) \leftarrow (DR_2–DR_0, A_3–A_0)$
		1	0	0	p_4	0	0	p_3	p_2	p_1	p_0	2 p p			
Return operation	RTI	0	0	0	1	0	0	0	1	1	0	0 4 6	1	1	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) – 1$
	RT	0	0	0	1	0	0	0	1	0	0	0 4 4	1	2	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) – 1$
	RTS	0	0	0	1	0	0	0	1	0	1	0 4 5	1	2	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) – 1$

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

Skip condition	Carry flag CY	Datailed 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 $(DR_2\ DR_1\ DR_0\ A_3\ A_2\ A_1\ A_0)_2$ specified by registers D and A in page p.
–	–	Call the subroutine in page 2 : Calls the subroutine at address a in page 2.
–	–	Call the subroutine : Calls the subroutine at address a in page p.
–	–	Call the subroutine : Calls the subroutine at address $(DR_2\ DR_1\ DR_0\ A_3\ A_2\ A_1\ A_0)_2$ specified by registers D and A in page p.
–	–	Returns from interrupt service routine to main routine. Returns each value of data pointer (X, Y, Z), carry flag, skip status, NOP mode status by the continuous description of the LA/LXY instruction, register A and register B to the states just before interrupt.
–	–	Returns from subroutine to the routine called the subroutine.
Skip at uncondition	–	Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition.

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Parameter / Type of instructions	Mnemonic	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
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	$V1_0$ = 0: (EXF0) = 1 ? After skipping, (EXF0) ← 0 $V1_0$ = 1: SNZ0 = NOP
	SNZI0	0	0	0	0	1	1	1	0	1	0	0 3 A	1	1	$I1_2$ = 0 : (INT) = “L” ? $I1_2$ = 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) ← ($T1_7$–$T1_4$) (A) ← ($T1_3$–$T1_0$)
	T1AB	1	0	0	0	1	1	0	0	0	0	2 3 0	1	1	($T1_7$–$T1_4$) ← (B) ($R1_7$–$R1_4$) ← (B) ($T1_3$–$T1_0$) ← (A) ($R1_3$–$R1_0$) ← (A)
	TAB2	1	0	0	1	1	1	0	0	0	1	2 7 1	1	1	(B) ← ($T2_7$–$T2_4$) (A) ← ($T2_3$–$T2_0$)
	T2AB	1	0	0	0	1	1	0	0	0	1	2 3 1	1	1	($T2_7$–$T2_4$) ← (B) ($R2_7$–$R2_4$) ← (B) ($T2_3$–$T2_0$) ← (A) ($R2_3$–$R2_0$) ← (A)
	TR1AB	1	0	0	0	1	1	1	1	1	1	2 3 F	1	1	($R1_7$–$R1_4$) ← (B) ($R1_3$–$R1_0$) ← (A)
	SNZT1	1	0	1	0	0	0	0	0	0	0	2 8 0	1	1	$V1_2$ = 0: (T1F) = 1 ? After skipping, (T1F) ← 0 $V1_2$ = 1: SNZT1 = NOP
	SNZT2	1	0	1	0	0	0	0	0	0	1	2 8 1	1	1	$V1_3$ = 0: (T2F) = 1 ? After skipping, (T2F) ← 0 $V1_3$ = 1: SNZT2 = NOP

Skip condition	Carry flag CY	Datailed description
–	–	Clears (0) to interrupt enable flag INTE, and disables the interrupt.
–	–	Sets (1) to interrupt enable flag INTE, and enables the interrupt.
$V1_0$ = 0: (EXF0) = 1	–	When $V1_0$ = 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 $V1_0$ = 1 : This instruction is equivalent to the NOP instruction. ($V1_0$: bit 0 of interrupt control register V1)
(INT) = “L” However, $I1_2$ = 0 (INT) = “H” However, $I1_2$ = 1	–	When $I1_2$ = 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 $I1_2$ = 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.” ($I1_2$: bit 2 of interrupt control register I1)
–	–	Transfers the contents of interrupt control register V1 to register A.
–	–	Transfers the contents of register A to interrupt control register V1.
–	–	Transfers the contents of interrupt control register V2 to register A.
–	–	Transfers the contents of register A to interrupt control register V2.
–	–	Transfers the contents of interrupt control register I1 to register A.
–	–	Transfers the contents of register A to interrupt control register I1.
–	–	Transfers the contents of timer control register W1 to register A.
–	–	Transfers the contents of register A to timer control register W1.
–	–	Transfers the contents of timer control register W2 to register A.
–	–	Transfers the contents of register A to timer control register W2.
–	–	Transfers the contents of timer control register W6 to register A.
–	–	Transfers the contents of register A to timer control register W6.
–	–	Transfers the high-order 4 bits ($T1_7$–$T1_4$) of timer 1 to register B. Transfers the low-order 4 bits ($T1_3$–$T1_0$) of timer 1 to register A.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 1 and timer 1 reload register R1. Transfers the contents of register A to the low-order 4 bits of timer 1 and timer 1 reload register R1.
–	–	Transfers the high-order 4 bits ($T2_7$–$T2_4$) of timer 2 to register B. Transfers the low-order 4 bits ($T2_3$–$T2_0$) of timer 2 to register A.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 2 and timer 2 reload register R2. Transfers the contents of register A to the low-order 4 bits of timer 2 and timer 2 reload register R2.
–	–	Transfers the contents of register B to the high-order 4 bits ($R1_7$–$R1_4$) of reload register R1, and the contents of register A to the low-order 4 bits ($R1_3$–$R1_0$) of reload register R1.
$V1_2$ = 0: (T1F) = 1	–	When $V1_2$ = 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 $V1_2$ = 1 : This instruction is equivalent to the NOP instruction. ($V1_2$: bit 2 of interrupt control register V1)
$V1_3$ = 0: (T2F) =1	–	When $V1_3$ = 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 $V1_3$ = 1 : This instruction is equivalent to the NOP instruction. ($V1_3$: bit 3 of interrupt control register V1)

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Type of instructions	Mnemonic	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
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	$(A_1, A_0) \leftarrow (P2_1, P2_0)$ $(A_3, A_2) \leftarrow 0$
	OP2A	1	0	0	0	1	0	0	0	1	0	2 2 2	1	1	$(P2_1, P2_0) \leftarrow (A_1, A_0)$
	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	$(A_0) \leftarrow (K)$ $(A_3–A_1) \leftarrow 0$
	OKA	1	0	0	0	0	1	1	1	1	1	2 1 F	1	1	$(K) \leftarrow (A_0)$
	TK0A	1	0	0	0	0	1	1	0	1	1	2 1 B	1	1	(K0) ← (A)
	TAK0	1	0	0	1	0	1	0	1	1	0	2 5 6	1	1	(A) ← (K0)
	TK1A	1	0	0	0	0	1	0	1	0	0	2 1 4	1	1	(K1) ← (A)
	TAK1	1	0	0	1	0	1	1	0	0	1	2 5 9	1	1	(A) ← (K1)
	TK2A	1	0	0	0	0	1	0	1	0	1	2 1 5	1	1	(K2) ← (A)
	TAK2	1	0	0	1	0	1	1	0	1	0	2 5 A	1	1	(A) ← (K2)
	TPU0A	1	0	0	0	1	0	1	1	0	1	2 2 D	1	1	(PU0) ← (A)
	TPU1A	1	0	0	0	1	0	1	1	1	0	2 2 E	1	1	(PU1) ← (A)
	TPU2A	1	0	0	0	1	0	1	1	1	1	2 2 F	1	1	(PU2) ← (A)

Skip condition	Carry flag CY	Datailed 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 (A_1, A_0) of register A.
–	–	Outputs the contents of the low-order 2 bits (A_1, A_0) 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 (A_0) of register A.
–	–	Outputs the contents of bit 0 (A_0) 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)

Type of instructions	Mnemonic	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
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 ($Q1_3$ = 0), (B) ← (AD_9–AD_6) (A) ← (AD_5–AD_2) In comparator mode ($Q1_3$ = 1), (B) ← (AD_7–AD_4) (A) ← (AD_3–AD_0)
	TALA	1	0	0	1	0	0	1	0	0	1	2 4 9	1	1	(A_3, A_2) ← (AD_1, AD_0) (A_1, A_0) ← 0
	TADAB	1	0	0	0	1	1	1	0	0	1	2 3 9	1	1	(AD_7–AD_4) ← (B) (AD_3–AD_0) ← (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 $Q1_3$ = 0: A-D conversion starting $Q1_3$ = 1: Comparator operation starting
	SNZAD	1	0	1	0	0	0	0	1	1	1	2 8 7	1	1	$V2_2$ = 0: (ADF) = 1 ? After skipping, (ADF) ← 0 $V2_2$ = 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	Datailed description
–	–	In the A-D conversion mode ($Q1_3$ = 0), transfers the high-order 4 bits (AD_9–AD_6) of register AD to register B, and the middle-order 4 bits (AD_5–AD_2) of register AD to register A. In the comparator mode ($Q1_3$ = 1), transfers the middle-order 4 bits (AD_7–AD_4) of register AD to register B, and the low-order 4 bits (AD_3–AD_0) of register AD to register A. ($Q1_3$: bit 3 of A-D control register Q1)
–	–	Transfers the low-order 2 bits (AD_1, AD_0) of register AD to the high-order 2 bits (AD_3, AD_2) of register A.
–	–	In the A-D conversion mode ($Q1_3$ = 0), this instruction is equivalent to the NOP instruction. In the comparator mode ($Q1_3$ = 1), transfers the contents of register B to the high-order 4 bits (AD_7–AD_4) of comparator register, and the contents of register A to the low-order 4 bits (AD_3–AD_0) of comparator register. ($Q1_3$ = 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 ($Q1_3$ = 0) or the comparator operation at the comparator mode ($Q1_3$ = 1) is started. ($Q1_3$ = bit 3 of A-D control register Q1)
$V2_2$ = 0: (ADF) = 1	–	When $V2_2$ = 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 $V2_2$ = 1 : This instruction is equivalent to the NOP instruction. ($V2_2$: 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 oscillation circuit and stops the ring oscillator.
–	–	Selects the RC oscillation circuit and stops the ring 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

D9–D4		000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111	010000 010111	011000 011111
D3–D0	Hex. notation	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	SNZI0	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)

D9–D4		100000	100001	100010	100011	100100	100101	100110	100111	101000	101001	101010	101011	101100	101101	101110	101111	110000 111111
D3–D0	Hex. notation	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	TMA 0	TAM 0	XAM 0	XAMI 0	XAMD 0	LXY
0001	1	–	–	OP1A	T2AB	–	–	IAP1	TAB2	SNZT2	–	–	TMA 1	TAM 1	XAM 1	XAMI 1	XAMD 1	LXY
0010	2	–	–	OP2A	–	–	TAMR	IAP2	–	–	–	–	TMA 2	TAM 2	XAM 2	XAMI 2	XAMD 2	LXY
0011	3	–	TW6A	–	–	–	TAI1	–	–	–	–	–	TMA 3	TAM 3	XAM 3	XAMI 3	XAMD 3	LXY
0100	4	TQ1A	TK1A	–	–	TAQ1	–	–	–	–	–	–	TMA 4	TAM 4	XAM 4	XAMI 4	XAMD 4	LXY
0101	5	–	TK2A	–	–	–	–	–	–	–	–	–	TMA 5	TAM 5	XAM 5	XAMI 5	XAMD 5	LXY
0110	6	–	TMRA	–	–	–	TAK0	–	–	–	–	–	TMA 6	TAM 6	XAM 6	XAMI 6	XAMD 6	LXY
0111	7	–	TI1A	–	–	–	–	–	–	SNZAD	–	–	TMA 7	TAM 7	XAM 7	XAMI 7	XAMD 7	LXY
1000	8	–	–	–	–	–	–	–	–	–	–	–	TMA 8	TAM 8	XAM 8	XAMI 8	XAMD 8	LXY
1001	9	–	–	–	TADAB	TALA	TAK1	–	TABAD	SNZCP	–	–	TMA 9	TAM 9	XAM 9	XAMI 9	XAMD 9	LXY
1010	A	–	–	–	–	–	TAK2	–	–	–	CMCK	–	TMA 10	TAM 10	XAM 10	XAMI 10	XAMD 10	LXY
1011	B	–	TK0A	–	–	TAW1	–	–	–	–	CRCK	–	TMA 11	TAM 11	XAM 11	XAMI 11	XAMD 11	LXY
1100	C	–	–	–	–	TAW2	–	–	–	RCP	DWDT	–	TMA 12	TAM 12	XAM 12	XAMI 12	XAMD 12	LXY
1101	D	–	–	TPU0A	–	–	–	–	–	SCP	–	–	TMA 13	TAM 13	XAM 13	XAMI 13	XAMD 13	LXY
1110	E	TW1A	–	TPU1A	–	–	–	–	–	–	–	–	TMA 14	TAM 14	XAM 14	XAMI 14	XAMD 14	LXY
1111	F	TW2A	OKA	TPU2A	TR1AB	–	–	IAK	–	–	ADST	–	TMA 15	TAM 15	XAM 15	XAMI 15	XAMD 15	LXY

The above table shows the relationship between machine language codes and machine language instructions. D_3–D_0 show the low-order 4 bits of the machine language code, and D_9–D_4 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

ABSOLUTE MAXIMUM RAINGS

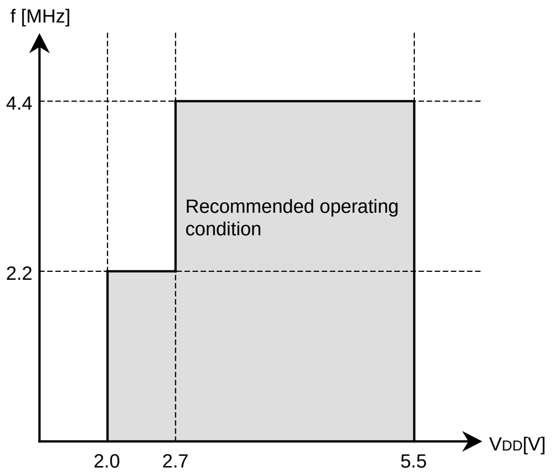
Symbol	Parameter	Conditions	Ratings	Unit
V_{DD}	Supply voltage		–0.3 to 6.5	V
V_I	Input voltage P0, P1, P2, D_0, D_1, D_2/C, D_3/K, $\overline{RESET}$, X_{IN}		–0.3 to V_{DD}+0.3	V
V_I	Input voltage A_{IN0}–A_{IN1}		–0.3 to V_{DD}+0.3	V
V_O	Output voltage P0, P1, P2, D_0, D_1, D_2/C, D_3/K, $\overline{RESET}$	Output transistors in cut-off state	–0.3 to V_{DD}+0.3	V
V_O	Output voltage X_{OUT}		–0.3 to V_{DD}+0.3	V
P_d	Power dissipation	Ta = 25 °C	300	mW
Topr	Operating temperature range		–20 to 85	°C
Tstg	Storage temperature range		–40 to 125	°C

RECOMMENDED OPERATING CONDITIONS 1 (Ta = –20 °C to 85 °C, V_{DD} = 2.0 V to 5.5 V, unless otherwise noted)

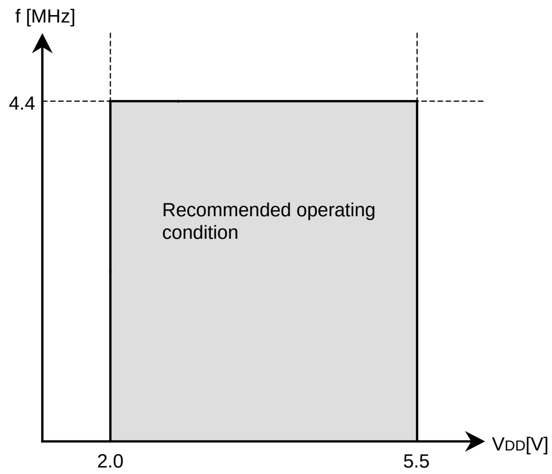
Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{DD}	Supply voltage (with a ceramic resonator)	High-speed mode	$f(X_{IN}) \le 4.4$ MHz	2.7		5.5	V
		Middle-speed mode Low-speed mode Default mode	$f(X_{IN}) \le 4.4$ MHz	2.0		5.5	
V_{DD}	Supply voltage (with RC oscillation)	High-speed mode Middle-speed mode Low-speed mode Default mode	$f(X_{IN}) \le 4.4$ MHz	2.7		5.5	V
V_{RAM}	RAM back-up voltage	(at RAM back-up)		1.8			V
V_{SS}	Supply voltage				0		V
V_{IH}	“H” level input voltage	P0, P1, P2, D_0–D_3, X_{IN}		$0.8V_{DD}$		V_{DD}	V
V_{IH}	“H” level input voltage	$\overline{\text{RESET}}$		$0.85V_{DD}$		V_{DD}	V
V_{IH}	“H” level input voltage	C, K	V_{DD} = 4.0 to 5.5 V	$0.5V_{DD}$		V_{DD}	V
			V_{DD} = 2.0 to 5.5 V	$0.7V_{DD}$		V_{DD}	
V_{IH}	“H” level input voltage	CNTR, INT		$0.85V_{DD}$		V_{DD}	V
V_{IL}	“L” level input voltage	P0, P1, P2, D_0–D_3, X_{IN}		0		$0.2V_{DD}$	V
V_{IL}	“L” level input voltage	C, K		0		$0.16V_{DD}$	V
V_{IL}	“L” level input voltage	$\overline{\text{RESET}}$		0		$0.3V_{DD}$	V
V_{IL}	“L” level input voltage	CNTR, INT		0		$0.15V_{DD}$	V
I_{OL}(peak)	“L” level peak output current	P2, $\overline{\text{RESET}}$	V_{DD} = 5.0 V			10	mA
			V_{DD} = 3.0 V			4.0	
I_{OL}(peak)	“L” level peak output current	D_0, D_1	V_{DD} = 5.0 V			40	mA
			V_{DD} = 3.0 V			30	
I_{OL}(peak)	“L” level peak output current	D_2/C, D_3/K	V_{DD} = 5.0 V			24	mA
			V_{DD} = 3.0 V			12	
I_{OL}(peak)	“L” level peak output current	P0, P1	V_{DD} = 5.0 V			24	mA
			V_{DD} = 3.0 V			12	
I_{OL}(avg)	“L” level average output current	P2, $\overline{\text{RESET}}$ (Note)	V_{DD} = 5.0 V			5.0	mA
			V_{DD} = 3.0 V			2.0	
I_{OL}(avg)	“L” level average output current	D_0, D_1 (Note)	V_{DD} = 5.0 V			30	mA
			V_{DD} = 3.0 V			15	
I_{OL}(avg)	“L” level average output current	D_2/C, D_3/K (Note)	V_{DD} = 5.0 V			15	mA
			V_{DD} = 3.0 V			7.0	
I_{OL}(avg)	“L” level average output current	P0, P1 (Note)	V_{DD} = 5.0 V			12	mA
			V_{DD} = 3.0 V			6.0	
ΣI_{OL}(avg)	“L” level total average current	P2, D, $\overline{\text{RESET}}$				80	mA
		P0, P1				80	

Note : The average output current (I_{OH}, I_{OL}) is the average value during 100 ms.

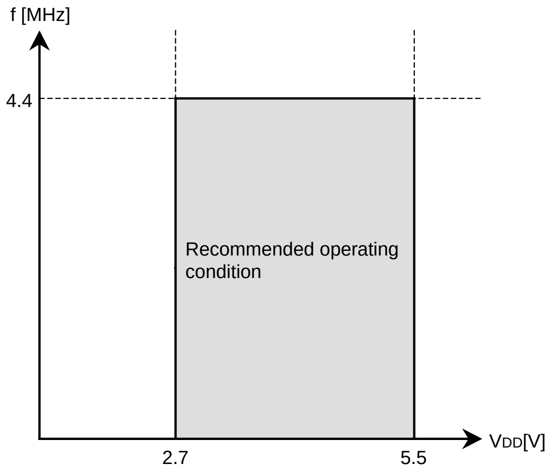
Ceramic resonator and high-speed mode selected



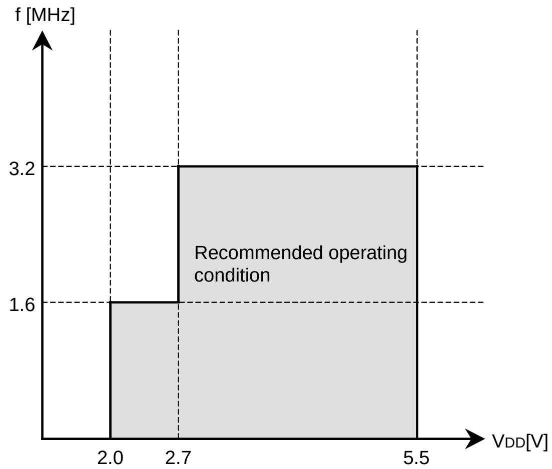
Except ceramic resonator and high-speed mode



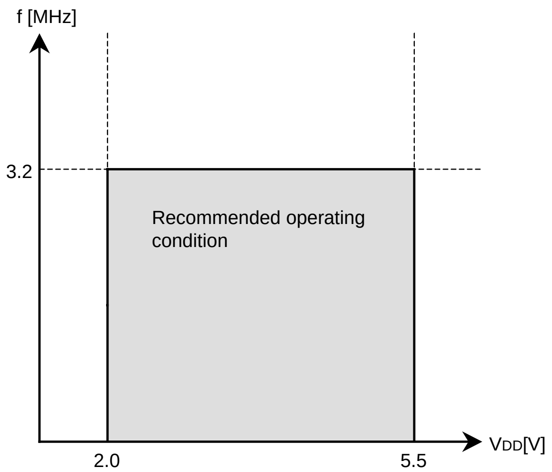
RC oscillation circuit selected



External clock input, high-speed mode selected
(ceramic resonator selected)



Except external clock input, high-speed mode
(ceramic resonator selected)



RECOMMENDED OPERATING CONDITIONS 2 (Ta = –20 °C to 85 °C, V_{DD} = 2.0 V to 5.5 V, unless otherwise noted)

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

Note: The frequency 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, V_{DD} = 2.0 V to 5.5 V, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OL}	"L" level output voltage P0, P1	V_{DD} = 5.0 V	I_{OL} = 12 mA			2.0	V
			I_{OL} = 4.0 mA			0.9	
		V_{DD} = 3.0 V	I_{OL} = 6.0 mA			0.9	
			I_{OL} = 2.0 mA			0.6	
V_{OL}	"L" level output voltage P2, $\overline{RESET}$	V_{DD} = 5.0 V	I_{OL} = 5.0 mA			2.0	V
			I_{OL} = 1.0 mA			0.6	
		V_{DD} = 3.0 V	I_{OL} = 2.0 mA			0.9	
V_{OL}	"L" level output voltage D_0, D_1	V_{DD} = 5.0 V	I_{OL} = 30 mA			2.0	V
			I_{OL} = 10 mA			0.9	
		V_{DD} = 3.0 V	I_{OL} = 15 mA			2.0	
			I_{OL} = 5.0 mA			0.9	
V_{OL}	"L" level output voltage D_2/C, D_3/K	V_{DD} = 5.0 V	I_{OL} = 15 mA			2.0	V
			I_{OL} = 5.0 mA			0.9	
		V_{DD} = 3.0 V	I_{OL} = 9.0 mA			2.0	
			I_{OL} = 3.0 mA			0.9	
I_{IH}	"H" level input current P0, P1, P2, $\overline{RESET}$	V_I = V_{DD}				1.0	µA
I_{IH}	"H" level input current D_0, D_1, D_2/C, D_3/K	V_I = V_{DD}				1.0	µA
I_{IL}	"L" level input current P0, P1, P2	V_I = 0 V P0, P1, P2 No pull-up		–1.0			µA
I_{IL}	"L" level input current D_0, D_1, D_2/C, D_3/K	V_I = 0 V, D_2/C, D_3/K, No pull-up		–1.0			µA
I_{DD}	Supply current at active mode (Note 1)	V_{DD} = 5.0 V f(X_{IN}) = 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	
		V_{DD} = 3.0 V f(X_{IN}) = 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	
	Supply current at RAM back-up mode (POF2 instruction execution)	Ta = 25 °C			0.1	1.0	µA
		V_{DD} = 5.0 V				10	
		V_{DD} = 3.0 V				6.0	
R_{PU}	Pull-up resistor value P0, P1, P2, D_2/C, D_3/K, $\overline{RESET}$	V_I = 0 V	V_{DD} = 5.0 V	30	60	150	kΩ
			V_{DD} = 3.0 V	50	120	300	
$V_{T+} – V_{T–}$	Hysteresis INT, CNTR	V_{DD} = 5.0 V			0.25		V
		V_{DD} = 3.0 V			0.25		
$V_{T+} – V_{T–}$	Hysteresis $\overline{RESET}$	V_{DD} = 5.0 V			1.2		V
		V_{DD} = 3.0 V			0.5		
f(RING)	Ring oscillator clock frequency (Note 2)	V_{DD} = 5.0 V		1.0	2.0	3.0	MHz
		V_{DD} = 3.0 V		0.5	1.0	1.8	

Notes 1: When the A-D converter is used, the A-D operation current (I_{ADD}) is included.
2: When system operates by the ring oscillator, the system clock frequency is the ring 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 Min.	Limits Typ.	Limits Max.	Unit
V_{DD}	Supply voltage	Ta = 25 °C		2.7		5.5	V
		Ta = –20 °C to 85 °C		3.0		5.5	
V_{IA}	Analog input voltage			0		V_{DD}+2LSB	V
f(X_{IN})	Oscillation frequency	V_{DD} = 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 (Ta = –20 °C to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
–	Resolution					10	bits
–	Linearity error	Ta = 25 °C, V_{DD} = 2.7 V to 5.5 V				±2.0	LSB
		Ta = –25 °C to 85 °C, V_{DD} = 3.0 V to 5.5 V					
–	Differential non-linearity error	Ta = 25 °C, V_{DD} = 2.7 V to 5.5 V				±0.9	LSB
		Ta = –25 °C to 85 °C, V_{DD} = 3.0 V to 5.5 V					
V_{0T}	Zero transition voltage	V_{DD} = 5.12 V		10	20	30	mV
		V_{DD} = 3.072 V		3	9	15	
V_{FST}	Full-scale transition voltage	V_{DD} = 5.12 V		5115	5125	5135	mV
		V_{DD} = 3.072 V		3063	3069	3075	
I_{ADD}	A–D operating current (Note 1)	V_{DD} = 5.0 V			0.3	0.9	mA
		V_{DD} = 3.0 V			0.1	0.3	
T_{CONV}	A-D conversion time	f(X_{IN}) = 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)	V_{DD} = 5.12 V				±20	mV
		V_{DD} = 3.072 V				±15	
–	Comparator comparison time	f(X_{IN}) = 4.0 MHz	High-speed mode			6.0	µs
			Middle-speed mode			12	
			Low-speed mode			24	
			Default mode			48	

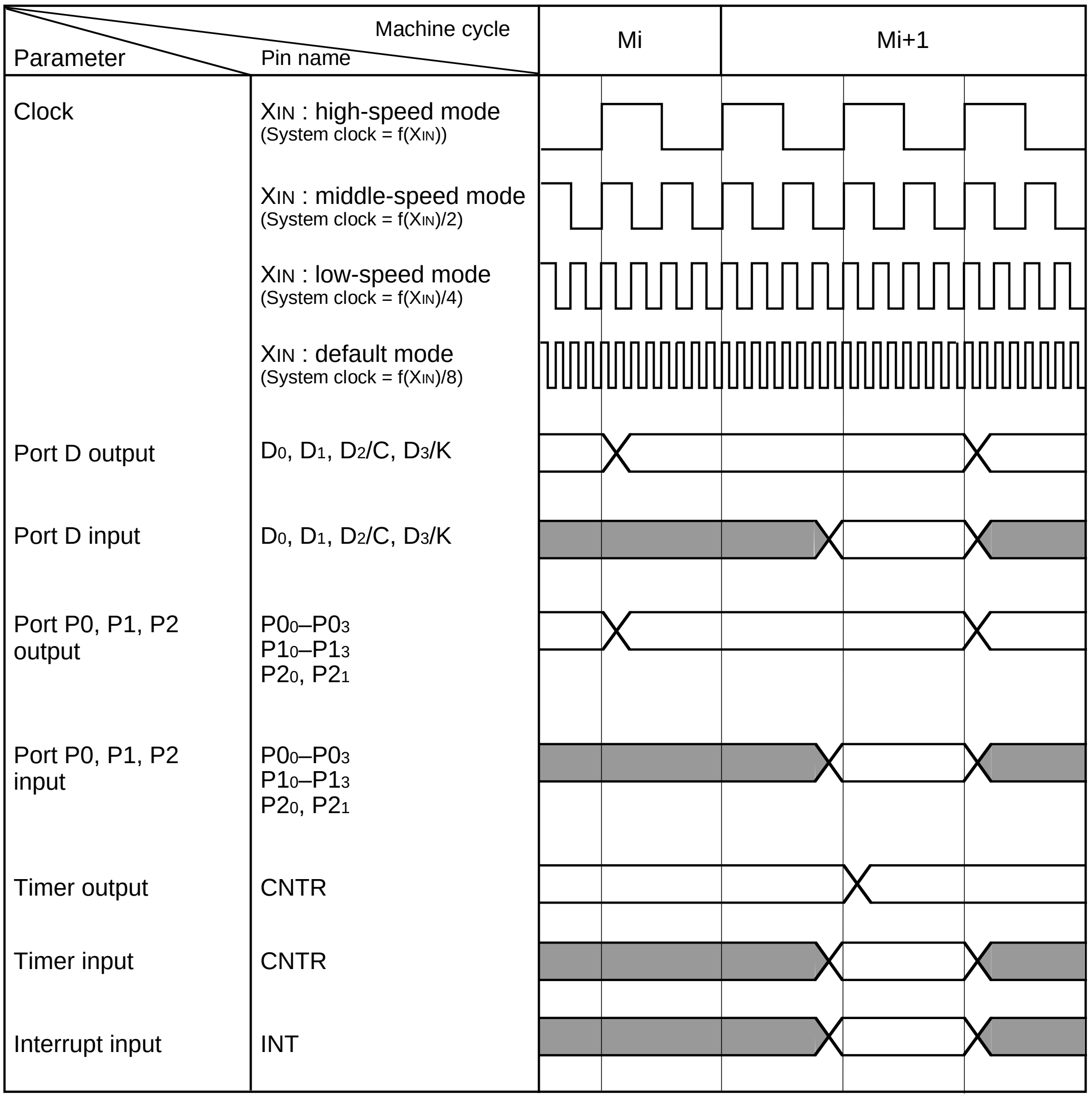
Notes 1: When the A-D converter is used, the I_{ADD} is included to I_{DD}.
2: As for the error from the logic value in the comparator mode, when the contents of the comparator register is n, the logic value of the comparison voltage V_{ref} which is generated by the built-in DA converter can be obtained by the following formula.

Logic value of comparison voltage V_{ref}

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

n = Value of register AD (n = 0 to 255)

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 24 shows the product of built-in PROM version. Figure 51 shows the pin configurations of built-in PROM versions.
The One Time PROM version has pin-compatibility with the mask ROM version.

Table 24 Product of built-in PROM version

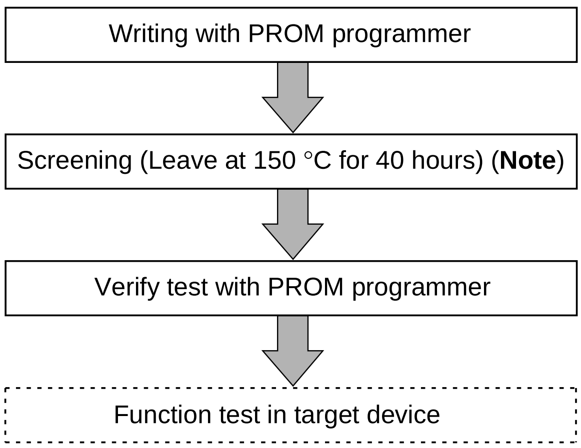
Product	PROM size (× 10 bits)	RAM size (× 4 bits)	Package	ROM type
M34506E4FP	4096 words	256 words	20P2N-A	One Time PROM [shipped in blank]

(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), $\overline{\text{PGM}}$ to “H” after connecting wires as shown in Figure 52 and powering on the V_{DD} pin, and then applying 12 V to the V_{PP} 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-perpose serial programmer when performing serial read/program.
As for the serial programmer for the Mitsubishi single-chip microcomputer (serial programmer and control software), refer to the “Mitsubishi Microcomputer Development Support Tools” Hompage (http://www.tool-spt.mesc.co.jp/index_e.htm).

(2) Notes on handling

① A high-voltage is used for writing. Take care that overvoltage is not applied. Take care especially at turning on the power.
② For the One Time PROM version shipped in blank, Mitsubishi Electric corp. does not perform PROM writing test and screening in the assembly process and following processes. In order to improve reliability after writing, performing writing and test according to the flow shown in Figure 51 before using is recommended (Products shipped in blank: PROM contents is not written in factory when shipped).



Note: Since the screening temperature is higher than storage temperature, never expose the microcomputer to 150 °C exceeding 100 hours.

Fig. 51 Flow of writing and test of the product shipped in blank

PIN CONFIGURATION (TOP VIEW)

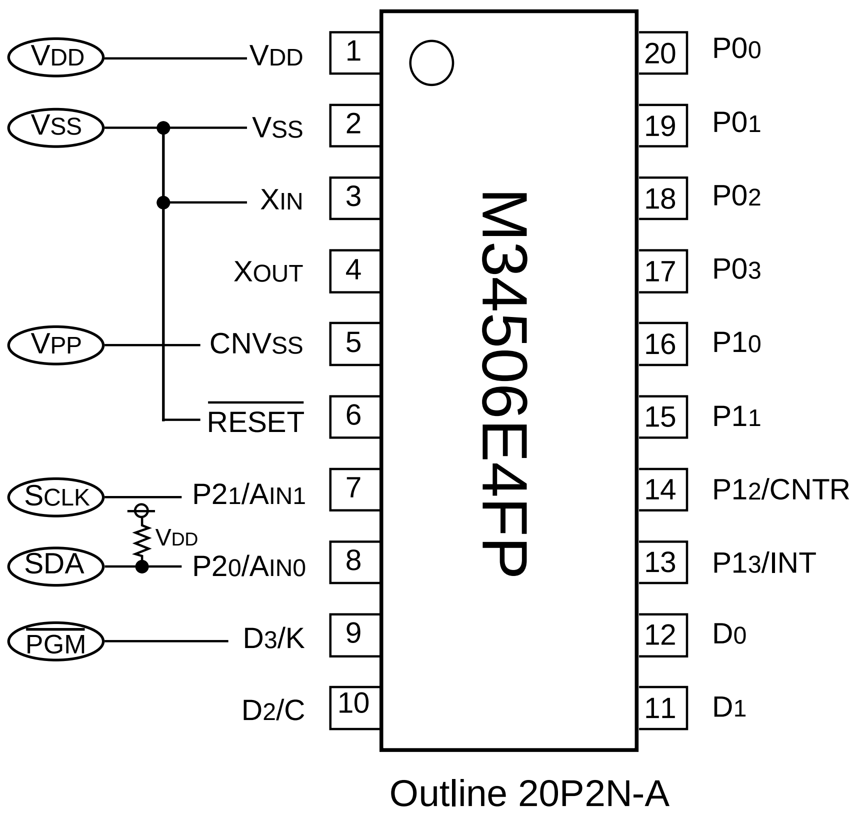


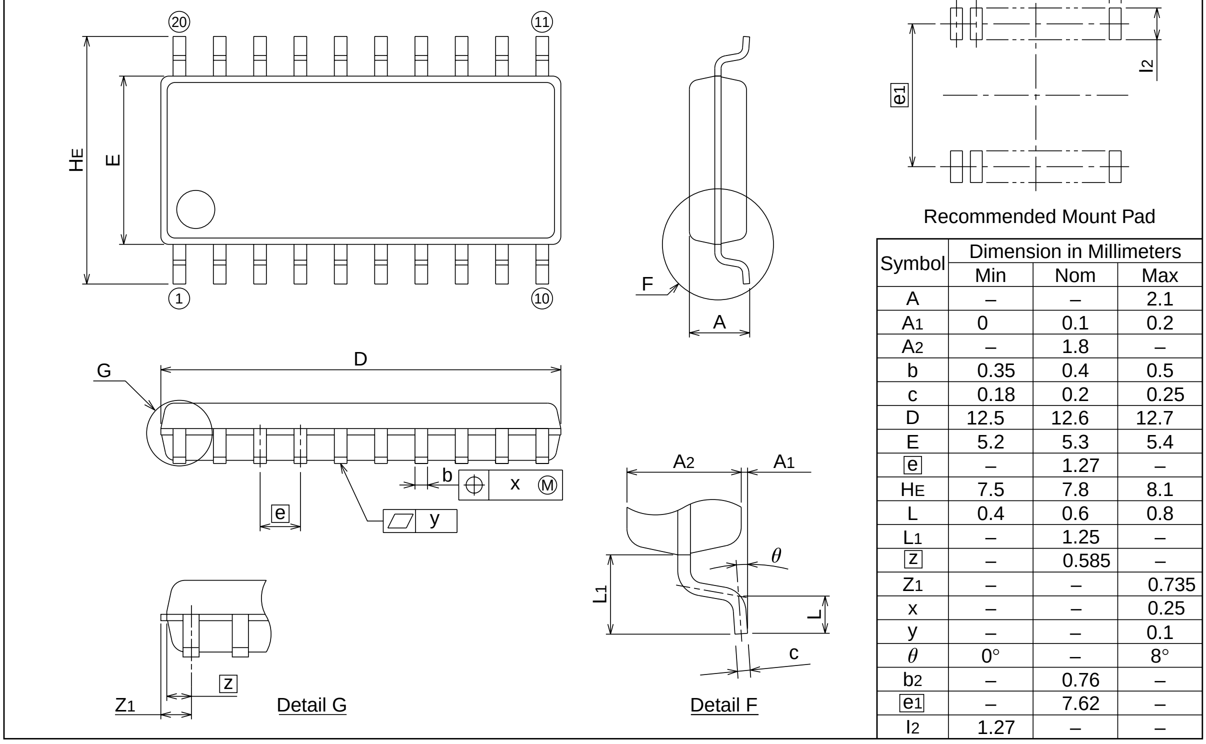
Fig. 52 Pin configuration of built-in PROM version

PACKAGE OUTLINE

20P2N-A

Plastic 20pin 300mil SOP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SOP20-P-300-1.27	–	0.26	Cu Alloy



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	2.1
A_1	0	0.1	0.2
A_2	–	1.8	–
b	0.35	0.4	0.5
c	0.18	0.2	0.25
D	12.5	12.6	12.7
E	5.2	5.3	5.4
e	–	1.27	–
H_E	7.5	7.8	8.1
L	0.4	0.6	0.8
L_1	–	1.25	–
z	–	0.585	–
Z_1	–	–	0.735
x	–	–	0.25
y	–	–	0.1
θ	0°	–	8°
b_2	–	0.76	–
e_1	–	7.62	–
I_2	1.27	–	–

Renesas Technology Corp.

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

Printed in Japan (ROD) II
New publication, effective June. 2001.
Specifications subject to change without notice.

RENESAS
Renesas Technology Corp.

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	The 4506/4507 Group data sheet is separated. Page 10: Port block diagram (3); Block diagram of $P1_2$/CNTR pin revised. Page 26: Fig. 22 Timers structure; Block diagram of $P1_2$/CNTR pin revised. Page 29: (9) Precautions → (8) Precautions (8) Timer input/output pin ($P1_2$/CNTR pin) added. Fig. 23 added. Page 30: WATCHDOG TIMER revised all. Page 31: Fig. 24 → Fig. 25, Fig. 25 → Fig. 26 Fig. 26 NOP instruction added. POF → POF2 Page 49: Fig. 46 POF → POF2 Page 61: BL p, a, BLA p instructions revised. Page 62: BML p, a, BMLA p instructions revised. Page 76: TABP p instruction revised. Page 90: TABP p instruction revised. Page 92: BL p, a, BLA p, BML p, a, BMLA p instructions revised. Page 100: BL, BML, BLA, BMLA instructions; The second word revised. Page 101: BL, BML, BLA, BMLA instructions; The second word revised. Page 102: ABSOLUTE MAXIMUM RATINGS; V_{DD} –0.3 to 6.0 → –0.3 to 6.5 Page 104: RECOMMENDED OPERATING CONDITIONS 1; Operating condition map added.	010531