

**DESCRIPTION**

The 4509 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 two reload registers), interrupts, 10-bit A/D converter, Serial interface and oscillation circuit switch function.

**FEATURES**

- Minimum instruction execution time ..... 0.5  $\mu$ s  
(at 6 MHz oscillation frequency, in through-mode)
- Supply voltage ..... 1.8 V to 5.5 V  
(It depends on operation source clock, oscillation frequency and operating mode.)
- Timers
  - Timer 1 ..... 8-bit timer with two reload registers
  - Timer 2 ..... 8-bit timer with two reload registers
- Interrupt ..... 5 sources
- Key-on wakeup function pins ..... 12
- Input/Output port ..... 18

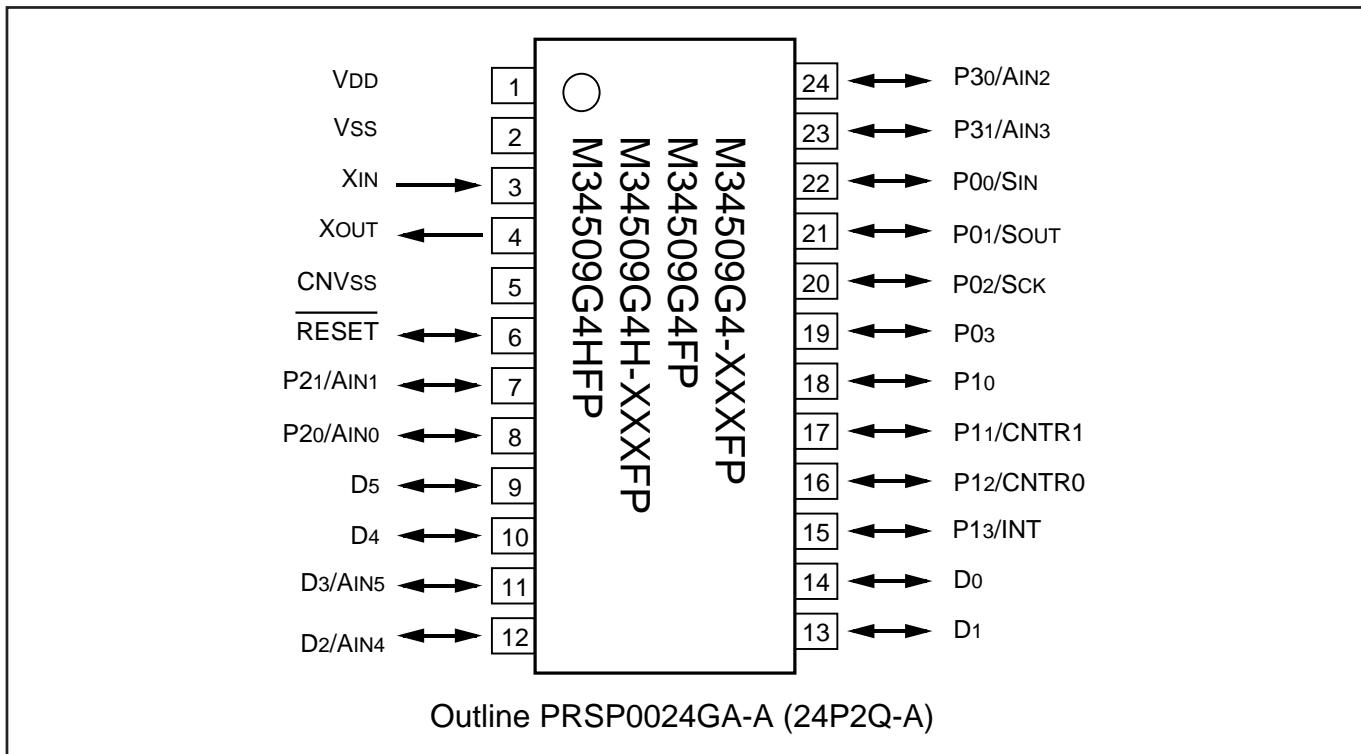
- A/D converter  
10-bit successive comparison method ..... 6 channel
- Serial interface ..... 8-bit  $\times$  1
- Voltage drop detection circuit (only for H version)  
Reset occurrence ..... Typ. 2.6 V (Ta = 25 °C)  
Reset release ..... Typ. 2.7 V (Ta = 25 °C)
- Power-on reset circuit (only for H version)
- Watchdog timer
- Clock generating circuit (on-chip oscillator/ceramic resonator/RC oscillation)
- LED drive directly enabled (port D)

**APPLICATION**

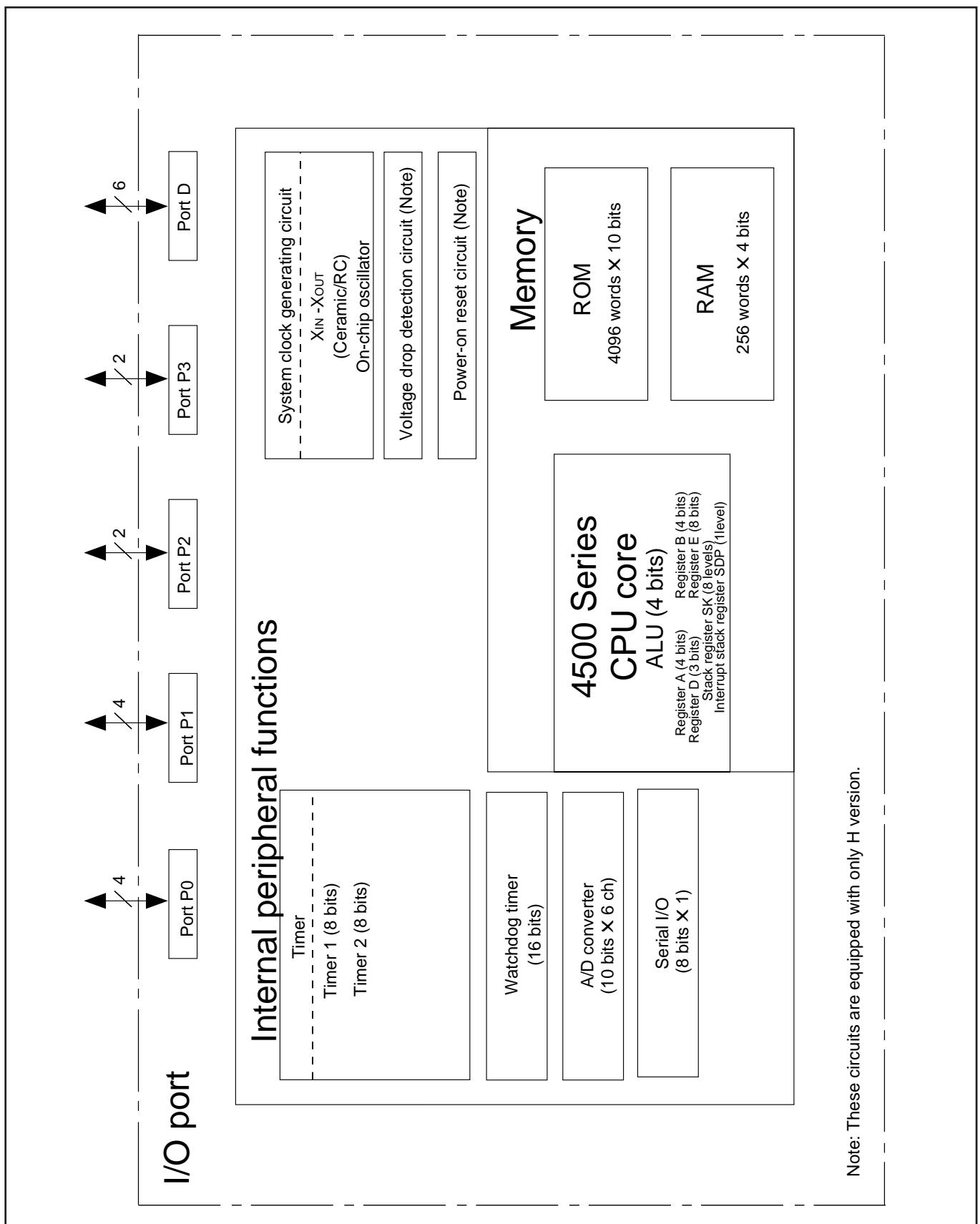
Electrical household appliance, consumer electronic products, office automation equipment, etc.

Part number	ROM (PROM) size ( $\times$ 10 bits)	RAM size ( $\times$ 4 bits)	Package	ROM type
M34509G4FP (Note)	4096 words	256 words	PRSP0024GA-A	QzROM
M34509G4-XXXFP	4096 words	256 words	PRSP0024GA-A	QzROM
M34509G4HFP (Note)	4096 words	256 words	PRSP0024GA-A	QzROM
M34509G4H-XXXFP	4096 words	256 words	PRSP0024GA-A	QzROM

**Note:** Shipped in blank.

**PIN CONFIGURATION**

Pin configuration (top view) (4509 Group)



Block diagram (4509 Group)

## PERFORMANCE OVERVIEW

Parameter			Function		
Number of basic instructions	M34509G4	134			
	M34509G4H	135			
Minimum instruction execution time			0.5 $\mu$ s (at 6 MHz oscillation frequency, in through mode)		
Memory sizes	ROM	4096 words $\times$ 10 bits			
	RAM	256 words $\times$ 4 bits			
Input/Output ports	D0–D5	I/O	Six independent I/O ports. Input is examined by skip decision. Ports D2 and D3 are equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software. Ports D2 and D3 are also used as AIN4, and AIN5, respectively.		
	P00–P03	I/O	4-bit I/O port; each pin is equipped with a pull-up function and a key-on wakeup function. Both functions and output structure can be switched by software. Ports P00, P01 and P02 are also used as SIN, SOUT and SCK, respectively.		
	P10–P13	I/O	4-bit I/O port; each pin is equipped with a pull-up function and a key-on wakeup function. Both functions and output structure can be switched by software. Ports P11, P12 and P13 are also used as CNTR1, CNTR0 and INT, respectively.		
	P20, P21	I/O	2-bit I/O port; each pin is equipped with a pull-up function and a key-on wakeup function. Both functions and output structure can be switched by software. Ports P20 and P21 are also used as AIN0 and AIN1, respectively.		
	P30, P31	I/O	2-bit I/O port; The output structure can be switched by software. Ports P30 and P31 are also used as AIN2 and AIN3, respectively.		
	CNTR0, CNTR1	Timer I/O	Two independent I/O; CNTR1 and CNTR0 pins are also used as ports P11 and P12, respectively.		
	INT	Interrupt input	1-bit input; INT pin is also used as port P13.		
	SIN, SOUT	Serial interface	Three independent I/O;		
	SCK	input/output	SIN, SOUT, and SCK are also used as ports P00, P01, and P02, respectively.		
Timers	AIN0–AIN5	Analog input	Six independent input; AIN0–AIN5 are also used as P20, P21, P30, P31, D2 and D3, respectively.		
	Timer 1		8-bit programmable timer/event counter with two reload registers and PWM output function.		
	Timer 2		8-bit programmable timer/event counter with two reload registers and PWM output function.		
Watchdog timer function			16-bit timer (fixed dividing frequency) (for watchdog)		
A/D converter	10-bit wide, This is equipped with an 8-bit comparator function.				
	Analog input		6 channel (AIN0–AIN5 pins)		
Serial interface			8-bit $\times$ 1		
Voltage drop detection circuit (Note)	Reset occurrence		Typ. 2.6 V (Ta = 25 °C)		
	Reset release		Typ. 2.7 V (Ta = 25 °C)		
Power-on reset circuit (Note)			Built-in type		
Interrupt	Sources		5 (one for external, two for timer, one for A/D, one for Serial interface)		
	Nesting		1 level		
Subroutine nesting			8 levels		
Device structure			CMOS silicon gate		
Package			24-pin plastic molded SSOP (PRSP0024GA-A)		
Operating temperature range			–20 °C to 85 °C		
Supply voltage			1.8 V to 5.5 V (It depends on operation source clock, oscillation frequency and operating mode.)		
Power dissipation (typical value)	Active mode		2.2 mA (Ta = 25°C, VDD = 5.0 V, f(XIN) = 6.0 MHz, f(STCK) = f(XIN)/1)		
	RAM back-up mode		0.1 $\mu$ A (Ta = 25°C, VDD = 5.0 V, output transistors in the cut-off state)		

Note: These circuits are equipped with only the H version.

**PIN DESCRIPTION**

Pin	Name	Input/Output	Function
VDD	Power supply	—	Connected to a plus power supply.
Vss	Ground	—	Connected to a 0 V power supply.
CNVss	CNVss	—	Connect CNVss to Vss and apply "L" (0V) to CNVss certainly.
RESET	Reset input/output	I/O	An N-channel open-drain I/O pin for a system reset. When the SRST instruction, watchdog timer, the voltage drop detection circuit (only for H version) or the built-in power-on reset (only for H version) causes the system to be reset, the RESET pin outputs "L" level.
XIN	System clock input	Input	I/O pins of the main clock generating circuit. When using a ceramic resonator, connect it between pins XIN and XOUT. A feedback resistor is built-in between them. When using the RC oscillation, connect a resistor and a capacitor to XIN, and leave XOUT pin open.
XOUT	System clock output	Output	
D0–D5	I/O port D Input is examined by skip decision.	I/O	Each pin of port D has an independent 1-bit wide I/O function. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Ports D2 and D3 are equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software. Ports D2 and D3 are also used as AIN4 and AIN5, respectively.
P00–P03	I/O port P0	I/O	Port P0 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Port P0 has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports P00, P01 and P02 are also used as SIN, SOUT and SCK, respectively.
P10–P13	I/O port P1	I/O	Port P1 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Port P1 has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports P11, P12 and P13 are also used as CNTR1, CNTR0 and INT, respectively.
P20, P21	I/O port P2	I/O	Port P2 serves as a 2-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Port P2 has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports P20 and P21 are also used as AIN0 and AIN1, respectively.
P30, P31	I/O port P3	I/O	Port P3 serves as a 2-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Ports P30 and P31 are also used as AIN2 and AIN3, respectively.
CNTR0	Timer input/output	I/O	CNTR0 pin has the function to input the clock for the timer 2 event counter, and to output the PWM signal generated by timer 1. This pin is also used as port P12.
CNTR1	Timer input/output	I/O	CNTR1 pin has the function to input the clock for the timer 1 event counter, and to output the PWM signal generated by timer 2. This pin is also used as port P11.
INT	Interrupt input	Input	INT pin accepts external interrupts. It has the key-on wakeup function which can be switched by software. This pin is also used as port P13.
AIN0–AIN5	Analog input	Input	A/D converter analog input pins. AIN0–AIN5 are also used as ports P20, P21, P30, P31, D2 and D3, respectively.
SCK	Serial interface clock I/O	I/O	Serial interface data transfer synchronous clock I/O pin. SCK pin is also used as port P02.
SOUT	Serial interface data output	Output	Serial interface data output pin. SOUT pin is also used as port P01.
SIN	Serial interface data input	Input	Serial interface data input pin. SIN pin is also used as port P00.

**MULTIFUNCTION**

Pin	Multifunction	Pin	Multifunction	Pin	Multifunction	Pin	Multifunction
P00	SIN	SIN	P00	P20	AIN0	AIN0	P20
P01	SOUT	SOUT	P01	P21	AIN1	AIN1	P21
P02	SCK	SCK	P02	P30	AIN2	AIN2	P30
P11	CNTR1	CNTR1	P11	P31	AIN3	AIN3	P31
P12	CNTR0	CNTR0	P12	D2	AIN4	AIN4	D2
P13	INT	INT	P13	D3	AIN5	AIN5	D3

Notes 1: Pins except above have just single function.

2: The input/output of P00 can be used even when SIN is used. Be careful when using inputs of both SIN and P00 since the input threshold value of SIN pin is different from that of port P00.

3: The input of P01 can be used even when SOUT is used.

4: The input of P02 can be used even when SCK is used. Be careful when using inputs of both SCK and P02 since the input threshold value of SCK pin is different from that of port P02.

5: The input of P11 can be used even when CNTR1 (output) is selected.

The input/output of P11 can be used even when CNTR1 (input) is selected. Be careful when using inputs of both CNTR1 and P11 since the input threshold value of CNTR1 pin is different from that of port P11.

6: The input of P12 can be used even when CNTR0 (output) is selected.

The input/output of P12 can be used even when CNTR0 (input) is selected. Be careful when using inputs of both CNTR0 and P12 since the input threshold value of CNTR0 pin is different from that of port P12.

7: The input/output of P13 can be used even when INT is used. Be careful when using inputs of both INT and P13 since the input threshold value of INT pin is different from that of port P13.

8: The input/output of P20, P21, P30, P31, D2, D3 can be used even when AIN0–AIN5 are used.

**PORT FUNCTION**

Port	Pin	Input Output	Output structure	I/O unit	Control instructions	Control registers	Remark
Port D	D0, D1, D4, D5	I/O (6)	N-channel open-drain/ CMOS	1	SD, RD SZD, CLD	FR3, C1	Programmable output structure selection function
	D2/AIN4 D3/AIN5					FR3, PU2 K2 Q1	Programmable pull-up function Programmable key-on wakeup function Programmable output structure selection function
Port P0	P00/SIN, P01/SOUT, P02/SCK, P03	I/O (4)	N-channel open-drain/ CMOS	4	OP0A IAP0	FR0, PU0 K0 J1	Programmable pull-up function Programmable key-on wakeup function Programmable output structure selection function
Port P1	P10, P11/CNTR1, P12/CNT0, P13/INT	I/O (4)	N-channel open-drain/ CMOS	4	OP1A IAP1	FR1, PU1 K1, L1, I1 W1, W2 W5, W6	Programmable pull-up function Programmable key-on wakeup function Programmable output structure selection function
Port P2	P20/AIN0 P21/AIN1	I/O (2)	N-channel open-drain/ CMOS	2	OP2A IAP2	FR2, PU2 Q1 K2	Programmable pull-up function Programmable key-on wakeup function Programmable output structure selection function
Port P3	P30/AIN2 P31/AIN3	I/O (2)	N-channel open-drain/ CMOS	2	OP3A IAP3	C1 Q1	Programmable output structure selection functions

## DEFINITION OF CLOCK AND CYCLE

### ● Operation source clock

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

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

### ● System clock

The system clock is the basic clock for controlling this product. The system clock is selected by the register MR and register RG.

### ● Instruction clock

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

### ● Machine cycle

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

**Table Selection of system clock**

Register MR, RG					System clock	Operation mode
MR3	MR2	MR1	MR0	RG0		
1	1	–	1	0	$f(STCK) = f(RING)/8$	Internal frequency divided by 8 mode
1	0	–	1	0	$f(STCK) = f(RING)/4$	Internal frequency divided by 4 mode
0	1	–	1	0	$f(STCK) = f(RING)/2$	Internal frequency divided by 2 mode
0	0	–	1	0	$f(STCK) = f(RING)$	Internal frequency through mode
1	1	0	0	–	$f(STCK) = f(XIN)/8$	High-speed frequency divided by 8 mode
1	0	0	0	–	$f(STCK) = f(XIN)/4$	High-speed frequency divided by 4 mode
0	1	0	0	–	$f(STCK) = f(XIN)/2$	High-speed frequency divided by 2 mode
0	0	0	0	–	$f(STCK) = f(XIN)$	High-speed through mode

**Note:** The internal frequency divided by 8 is selected after system is released from reset.

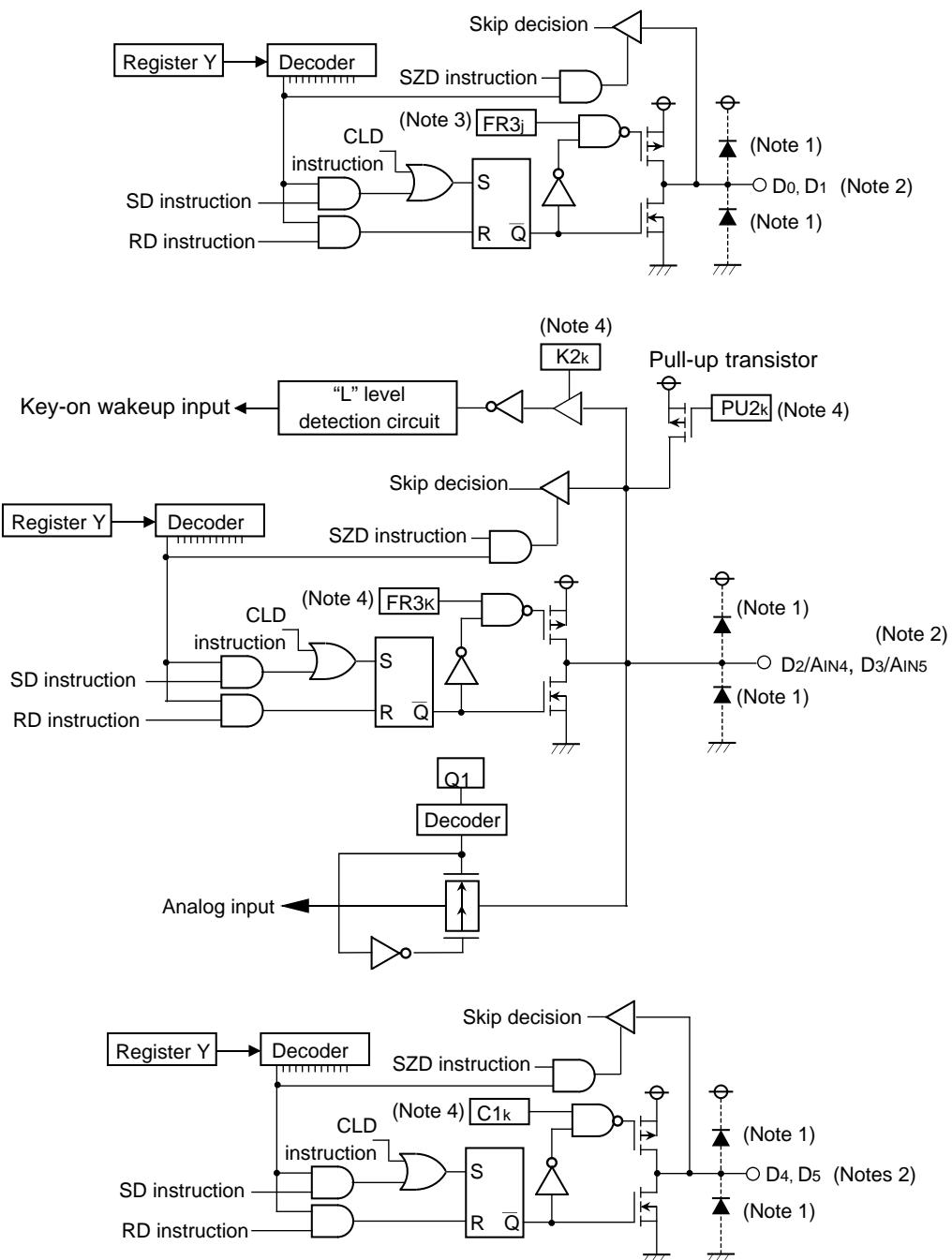
## CONNECTIONS OF UNUSED PINS

Pin	Connection	Usage condition
XIN	Connect to Vss.	RC oscillation circuit is not selected. (CRCK instruction is not executed.)
XOUT	Open.	_____
D0, D1, D4, D5	Open.	_____
	Connect to Vss.	N-channel open-drain is selected for the output structure (FR30, FR31, C12, C13 = "0").
D2/AIN4, D3/AIN5	Open.	The key-on wakeup function is invalid (K22, K23 = "0").
	Connect to Vss.	N-channel open-drain is selected for the output structure (FR32, FR33 = "0"). Pull-up transistor is OFF (PU22, PU23 = "0"). The key-on wakeup function is invalid (K22, K23 = "0").
P00/SIN	Open.	SIN pin is not selected (J11 = "0"). The key-on wakeup function is invalid (K00 = "0").
	Connect to Vss.	N-channel open-drain is selected for the output structure (FR00 = "0"). Pull-up transistor is OFF (PU00 = "0"). The key-on wakeup function is invalid (K00 = "0").
P01/SOUT	Open.	The key-on wakeup function is invalid (K01 = "0").
	Connect to Vss.	N-channel open-drain is selected for the output structure (FR01 = "0"). Pull-up transistor is OFF (PU01 = "0"). The key-on wakeup function is invalid (K01 = "0").
P02/SCK	Open.	SCK pin is not selected (J11J10 = "00"). The key-on wakeup function is invalid (K02 = "0").
	Connect to Vss.	N-channel open-drain is selected for the output structure (FR02 = "0"). Pull-up transistor is OFF (PU02 = "0"). The key-on wakeup function is invalid (K02 = "0").
P03	Open.	The key-on wakeup function is invalid (K03 = "0").
	Connect to Vss.	N-channel open-drain is selected for the output structure (FR03 = "0"). Pull-up transistor is OFF (PU03 = "0"). The key-on wakeup function is invalid (K03 = "0").
P10	Open.	The key-on wakeup function is invalid (K10 = "0").
	Connect to Vss.	N-channel open-drain is selected for the output structure (FR10 = "0"). Pull-up transistor is OFF (PU10 = "0"). The key-on wakeup function is invalid (K10 = "0").
P11/CNTR1	Open.	CNTR1 input is not selected for the timer 1 count source (W11, W10 ≠ "10"). The key-on wakeup function is invalid (K11 = "0").
	Connect to Vss.	N-channel open-drain is selected for the output structure (FR11 = "0"). Pull-up transistor is OFF (PU11 = "0"). The key-on wakeup function is invalid (K11 = "0").
P12/CNTR0	Open.	CNTR0 input is not selected for the timer 2 count source (W21, W20 ≠ "10"). The key-on wakeup function is invalid (K12 = "0").
	Connect to Vss.	N-channel open-drain is selected for the output structure (FR12 = "0"). Pull-up transistor is OFF (PU12 = "0"). The key-on wakeup function is invalid (K12 = "0").
P13/INT	Open.	INT pin input is disabled (I13 = "0"). The key-on wakeup function is invalid (K13 = "0").
	Connect to Vss.	N-channel open-drain is selected for the output structure (FR13 = "0"). Pull-up transistor is OFF (PU13 = "0"). The key-on wakeup function is invalid (K13 = "0").
P20/AIN0, P21/AIN1	Open.	The key-on wakeup function is invalid (K20, K21 = "0").
	Connect to Vss.	N-channel open-drain is selected for the output structure (FR20, FR21 = "0"). Pull-up transistor is OFF (PU20, PU21 = "0"). The key-on wakeup function is invalid (K20, K21 = "0").
P30/AIN2, P31/AIN3	Open.	_____
	Connect to Vss.	N-channel open-drain is selected for the output structure (C11, C10 = "0").

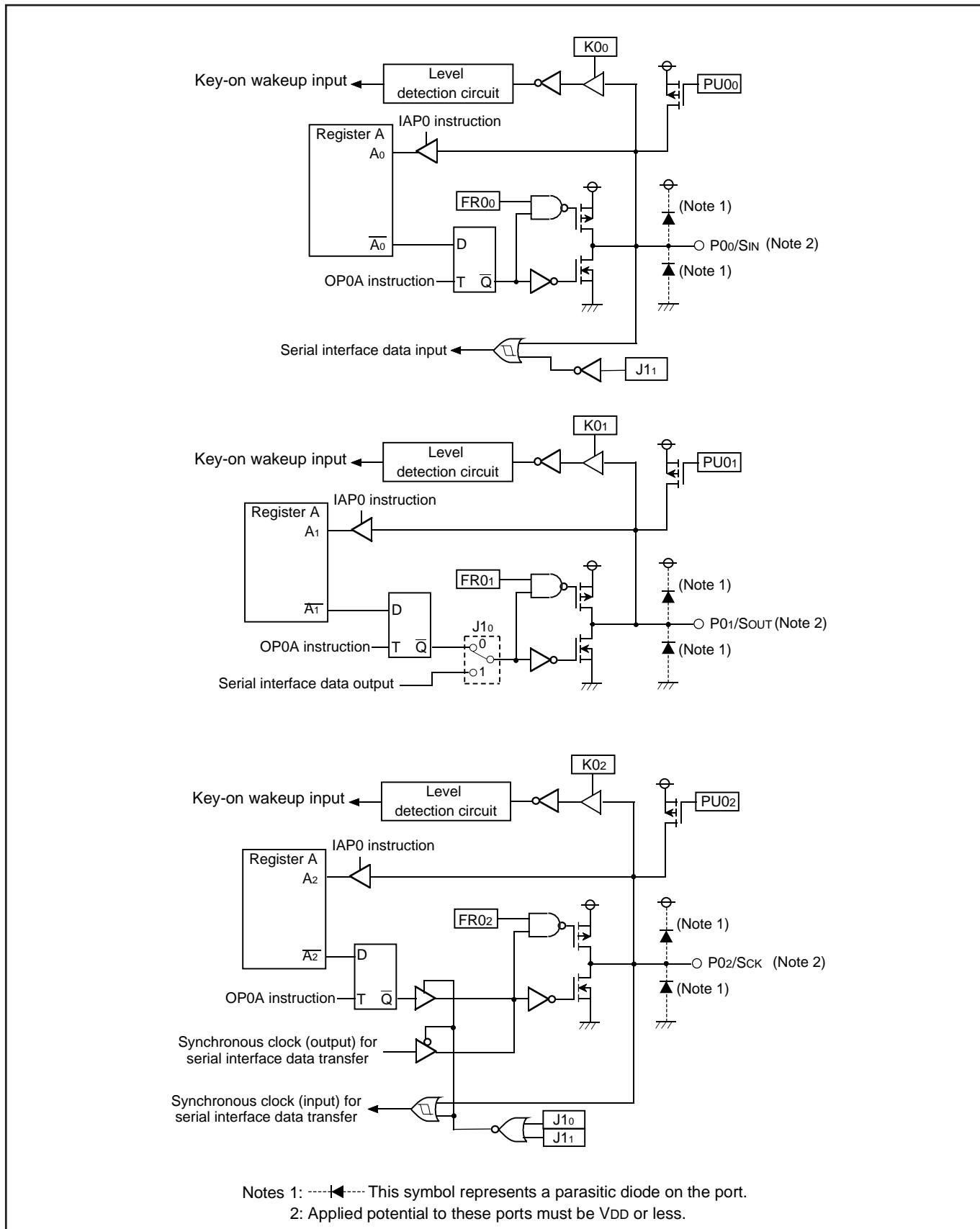
(Note when connecting to Vss or VDD)

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

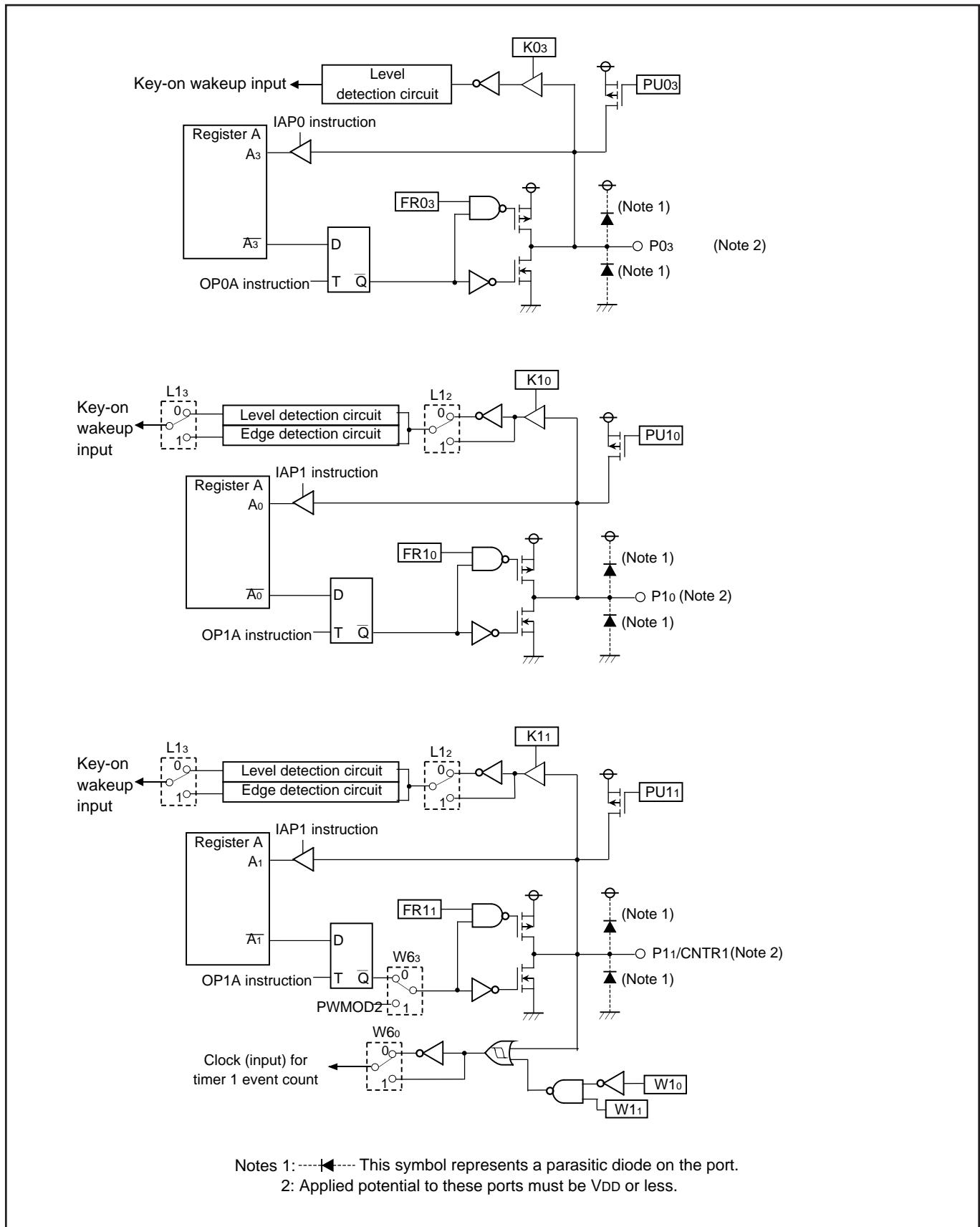
## PORT BLOCK DIAGRAMS



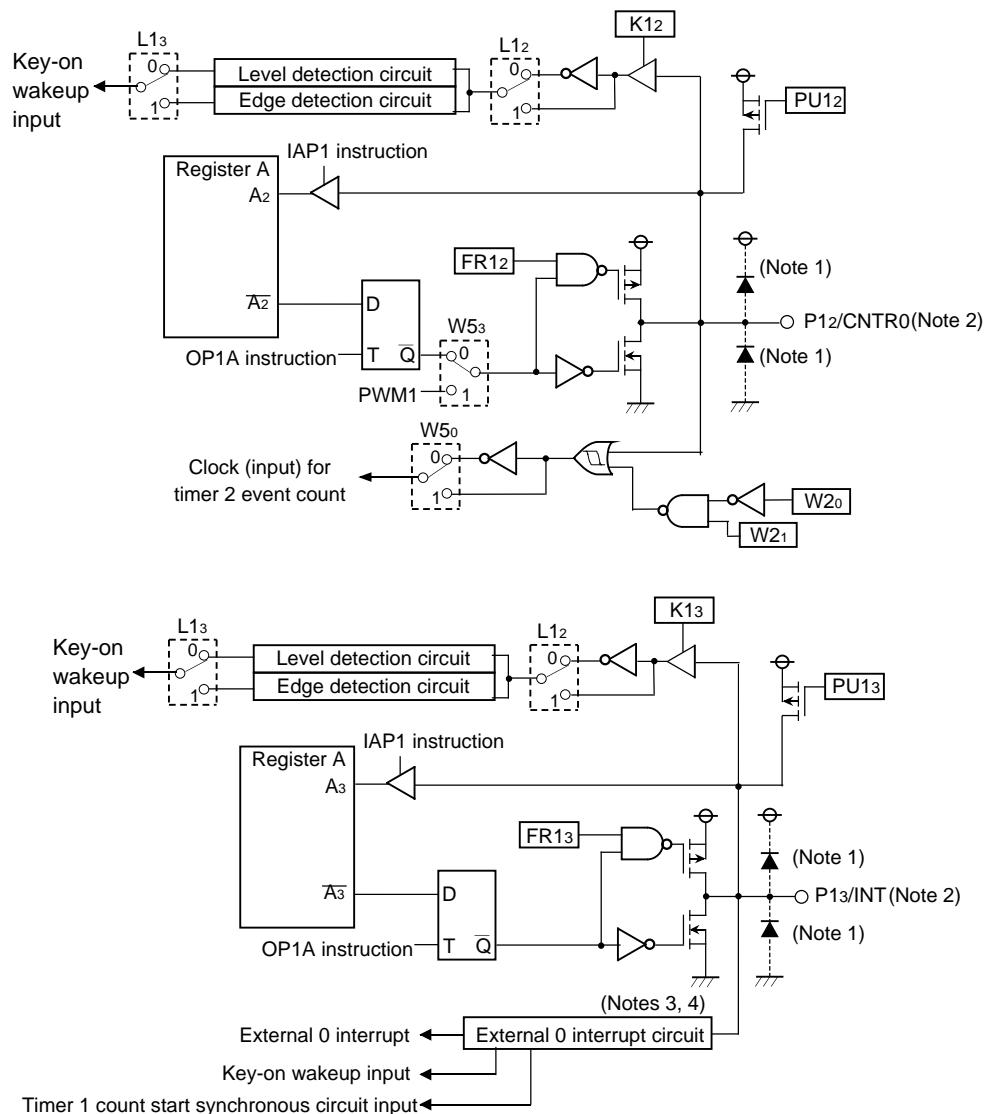
Port block diagram (1)



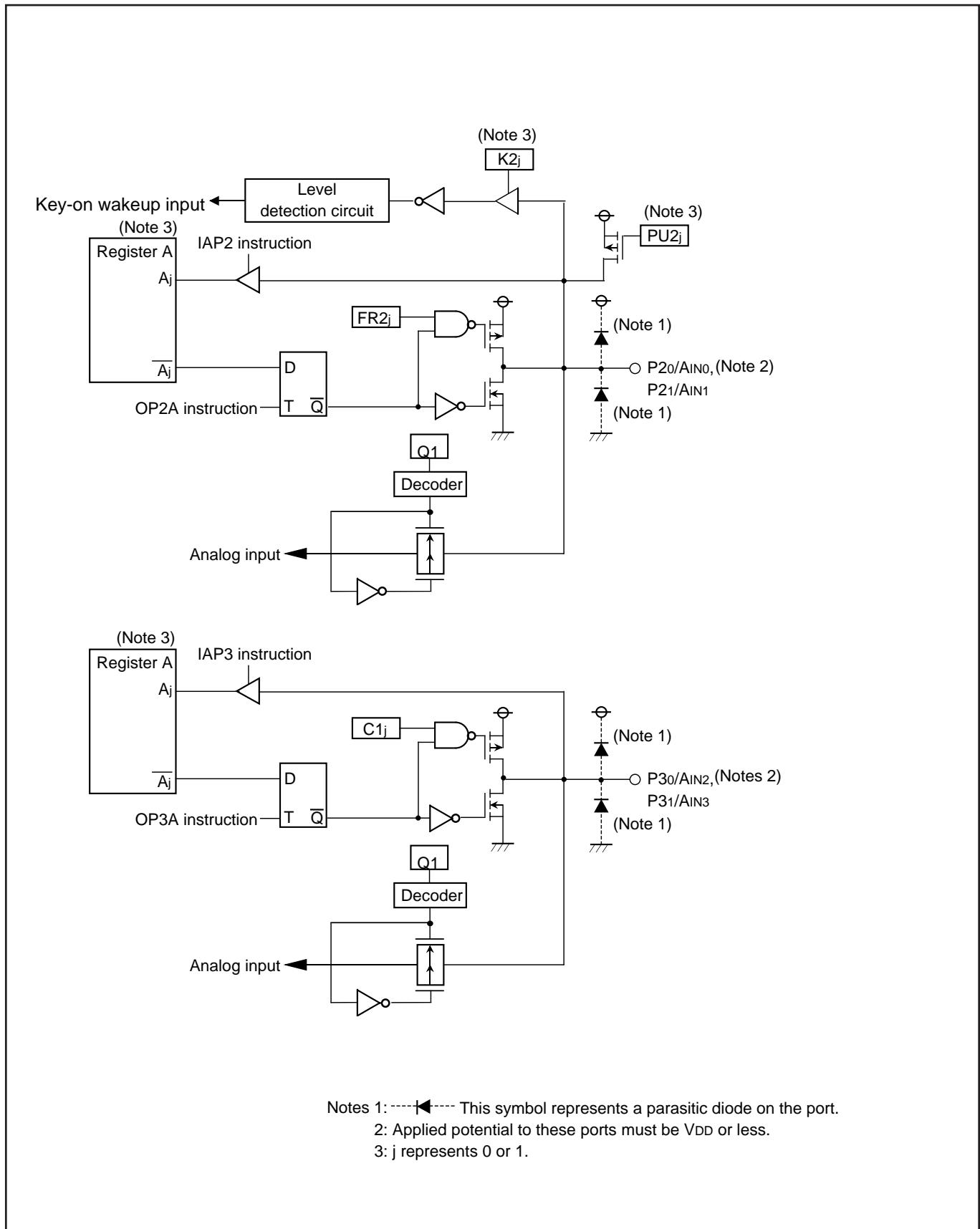
Port block diagram (2)



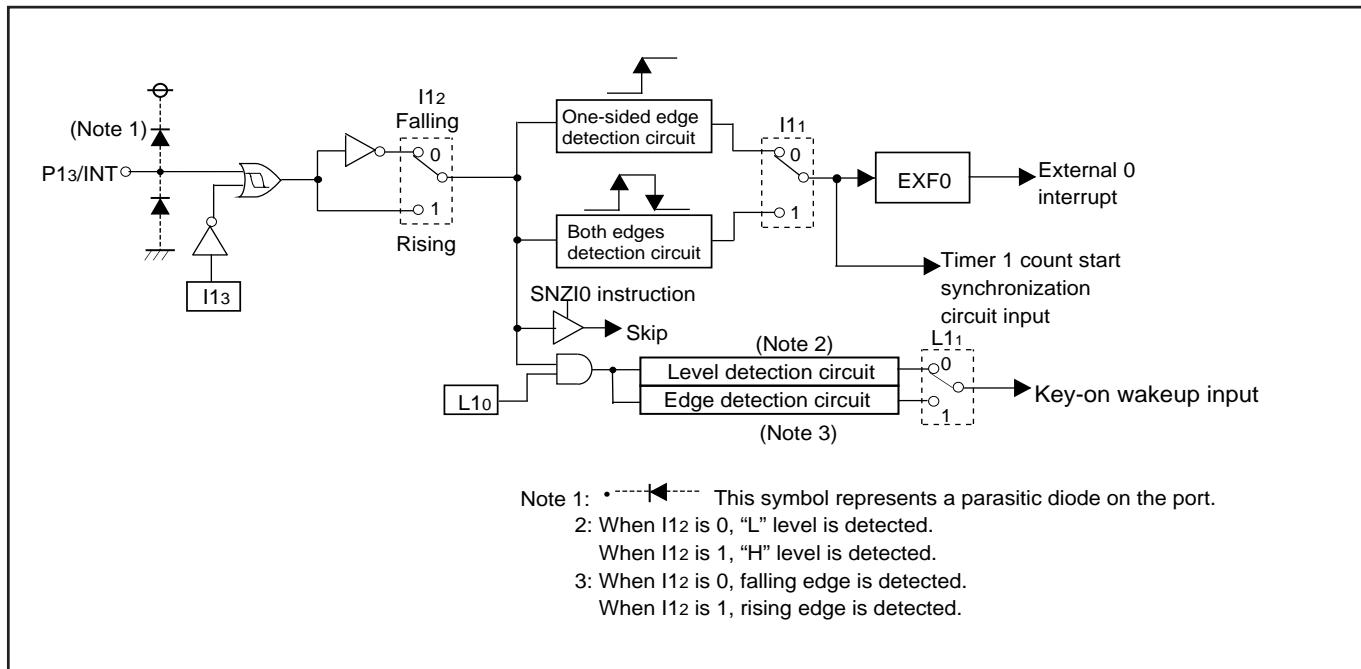
Port block diagram (3)



Port block diagram (4)



Port block diagram (5)



External interrupt circuit structure

## FUNCTION BLOCK OPERATIONS CPU

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

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

### (2) Register A and carry flag

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

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

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

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

### (3) Registers B and E

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

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

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

### (4) Register D

Register D is a 3-bit register.

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

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

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

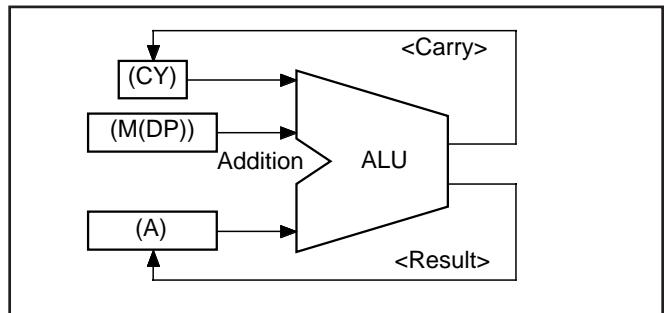


Fig. 1 AMC instruction execution example

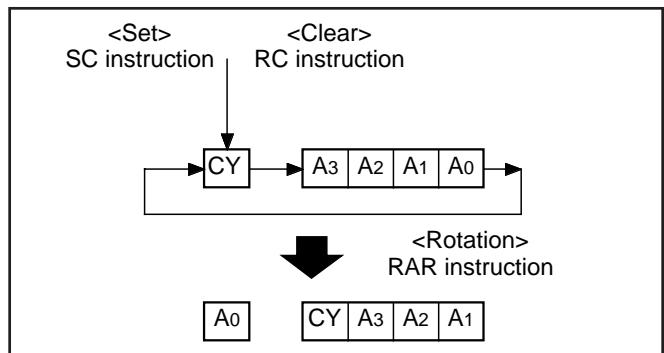


Fig. 2 RAR instruction execution example

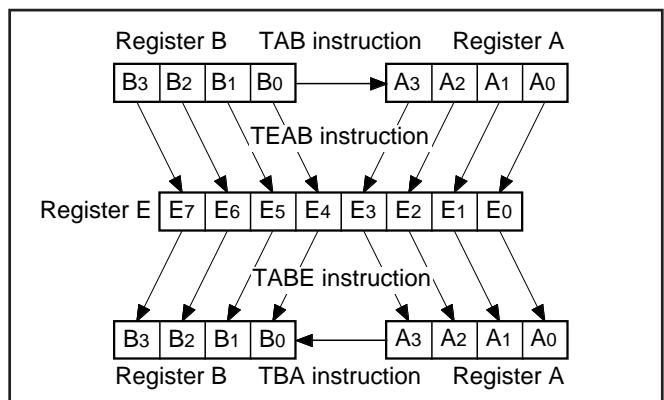


Fig. 3 Registers A, B and register E

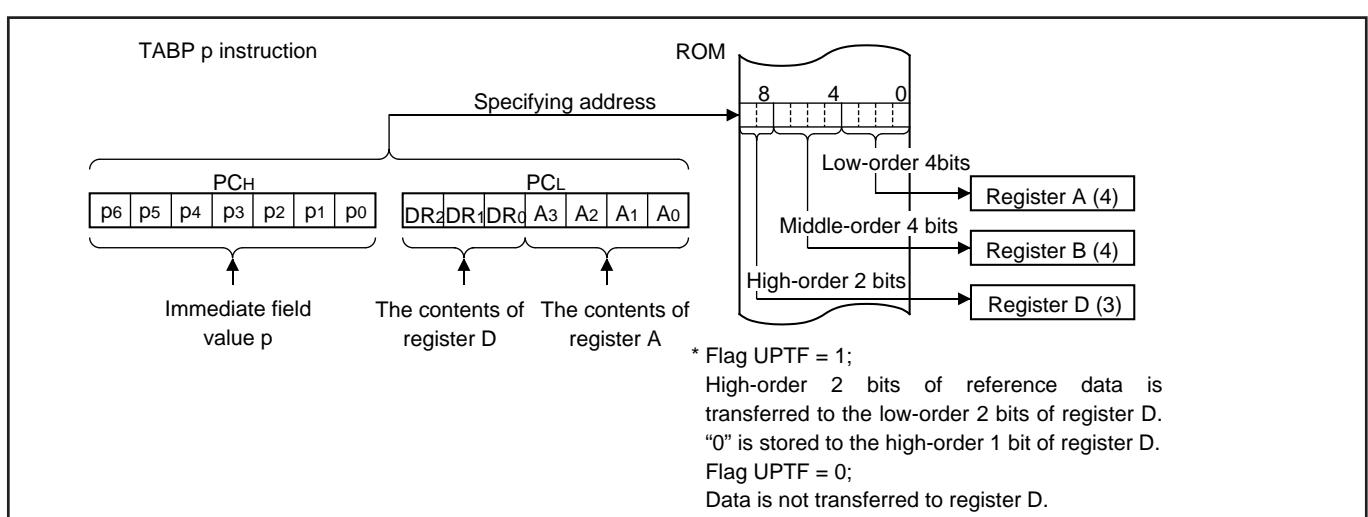


Fig. 4 TABP p instruction execution example

## (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.

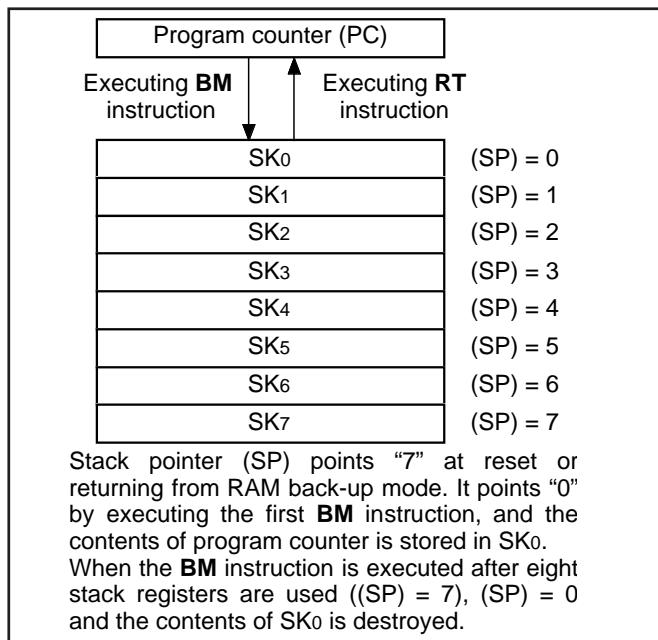


Fig. 5 Stack registers (SKs) structure

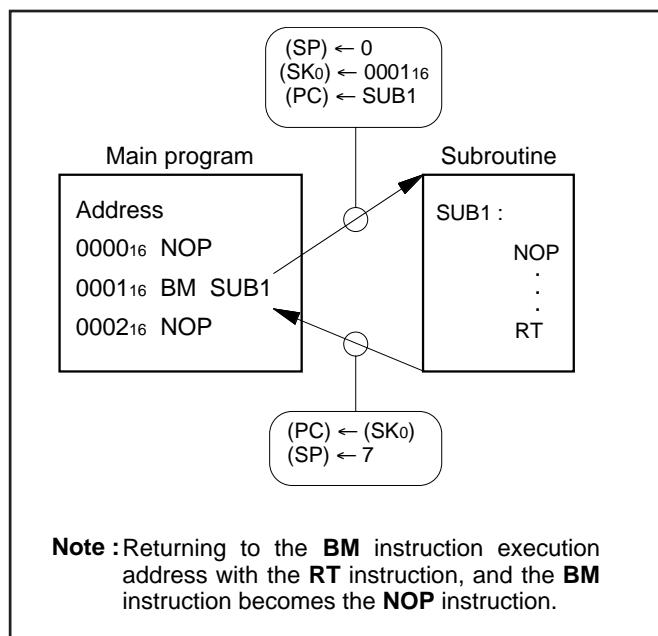


Fig. 6 Example of operation at subroutine call

## (8) Program counter (PC)

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

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

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

## (9) Data pointer (DP)

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

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

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

- Note

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

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

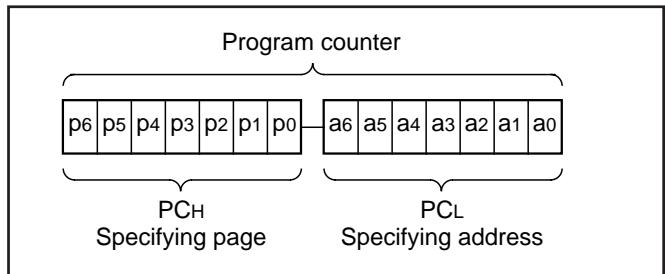


Fig. 7 Program counter (PC) structure

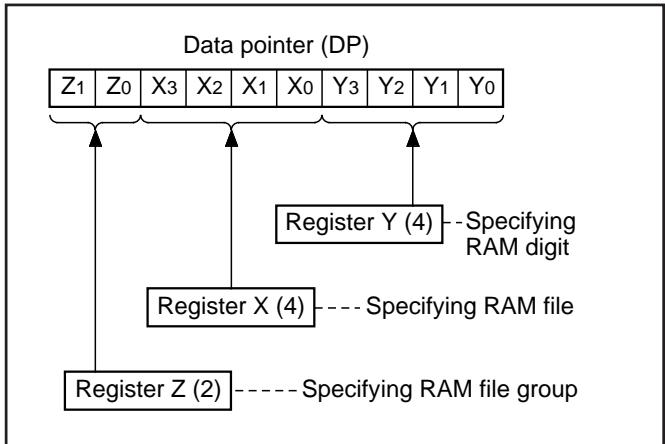


Fig. 8 Data pointer (DP) structure

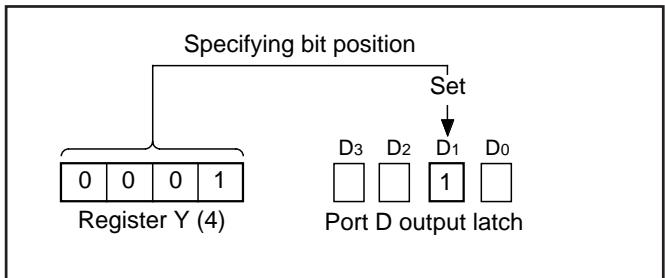


Fig. 9 SD instruction execution example

## PROGRAM MEMOY (ROM)

1 word of program memory 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 M34509G4.

Table 1 ROM size and pages

Part number	ROM (PROM) size ( $\times 10$ bits)	Pages
M34509G4	4096 words	32 (0 to 31)
M34509G4H	4096 words	32 (0 to 31)

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

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

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

## ROM Code Protect Address

When selecting the protect bit write by using a serial programmer or selecting protect enabled for writing shipment by Renesas Technology corp., reading or writing from/to QzROM is disabled by a serial programmer.

As for the QzROM product in blank, the ROM code is protected by selecting the protect bit write at ROM writing with a serial programmer.

As for the QzROM product shipped after writing, whether the ROM code protect is used or not can be selected as ROM option setup ("MASK option" written in the mask file converter) when ordering.

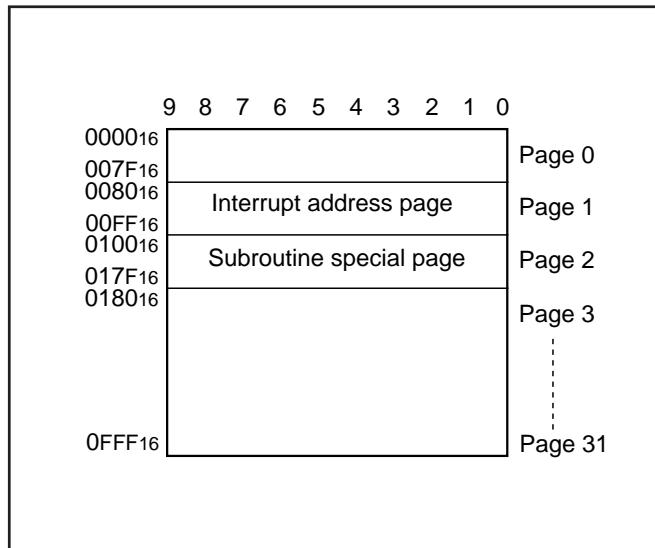


Fig. 10 ROM map of M34509G4

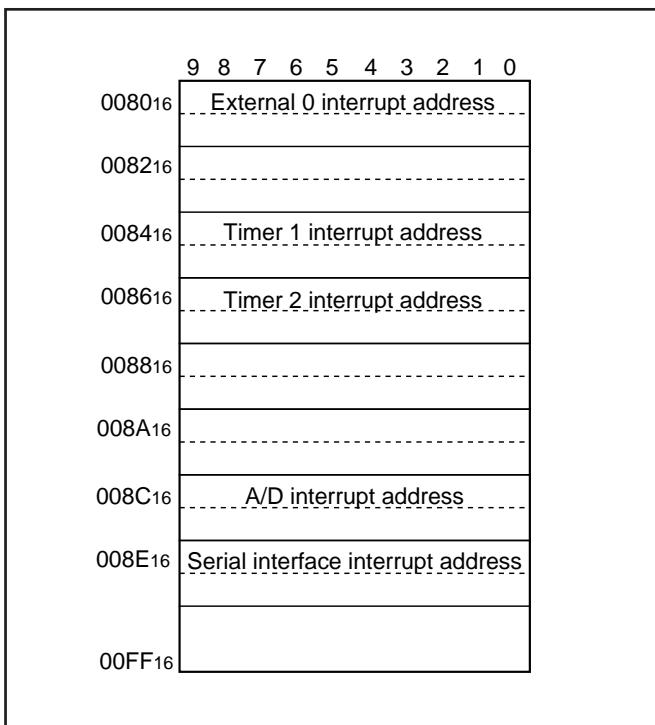


Fig. 11 Page 1 (addresses 008016 to 00FF16) structure

## 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**

Part number	RAM size
M34509G4	256 words X 4 bits (1024 bits)
M34509G4H	256 words X 4 bits (1024 bits)

**RAM 256 words X 4 bits (1024 bits)**

Register Z	0															
	0	1	2	3	...	6	7	.....	15							
Register X	0	1	2	3	...	6	7	.....	15							
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																

**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
5	Serial interface interrupt	Completion of serial interface transmit/recieve	Address E in page 1

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

Interrupt name	Interrupt request flag	Skip instruction	Interrupt enable bit
External 0 interrupt	EXF0	SNZ0	V10
Timer 1 interrupt	T1F	SNZT1	V12
Timer 2 interrupt	T2F	SNZT2	V13
A/D interrupt	ADF	SNZAD	V22
Serial interface interrupt	SIOF	SNZSI	V23

**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).

• 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

#### (5) Interrupt processing

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

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

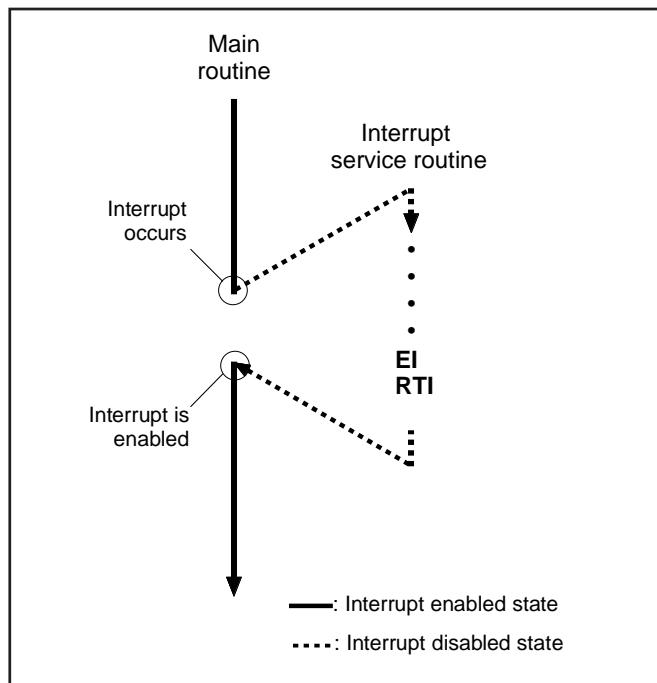


Fig. 13 Program example of interrupt processing

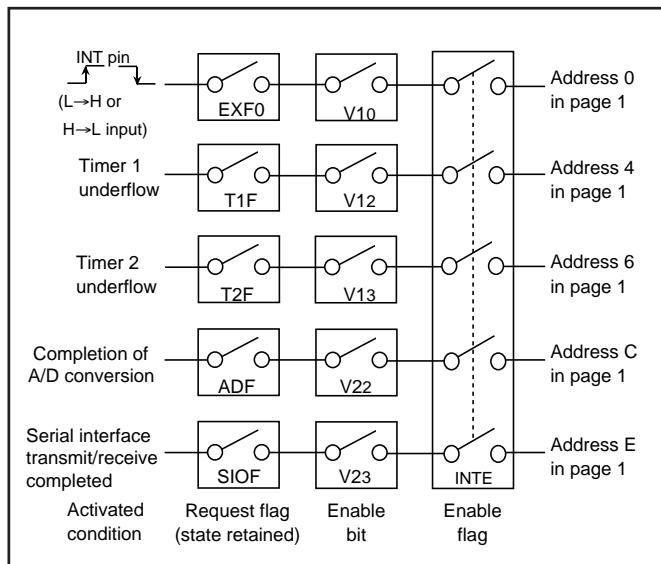


Fig. 15 Interrupt system diagram

## (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 and serial interface interrupt enable bit are assigned to register V2. Set the contents of this register through register A with the TV2A instruction. The TAV2 instruction can be used to transfer the contents of register V2 to register A.

**Table 6 Interrupt control registers**

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

Interrupt control register V2		at reset : 00002		at RAM back-up : 00002	R/W TAV2/TV2A
V23	Serial interface interrupt enable bit	0	Interrupt disabled (SNZSI instruction is valid)		
		1	Interrupt enabled (SNZSI instruction is invalid)		
V22	A/D interrupt enable bit	0	Interrupt disabled (SNZAD instruction is valid)		
		1	Interrupt enabled (SNZAD instruction is invalid)		
V21	Not used	0	This bit has no function, but read/write is enabled.		
		1			
V20	Not used	0	This bit has no function, but read/write is enabled.		
		1			

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

## (7) Interrupt sequence

Interrupts only occur when the respective INTE flag, interrupt enable bits (V10, V12, V13, V22, V23), 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).

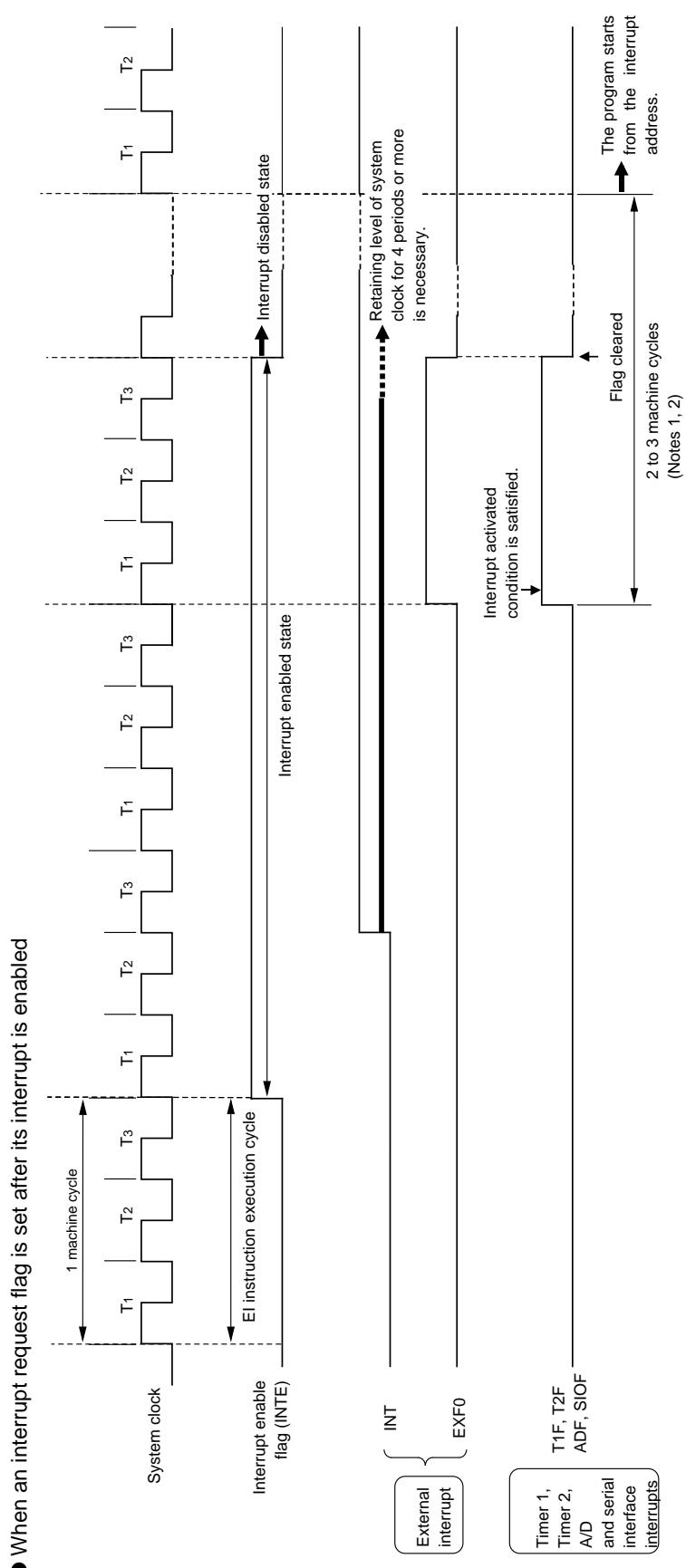


Fig. 16 Interrupt sequence

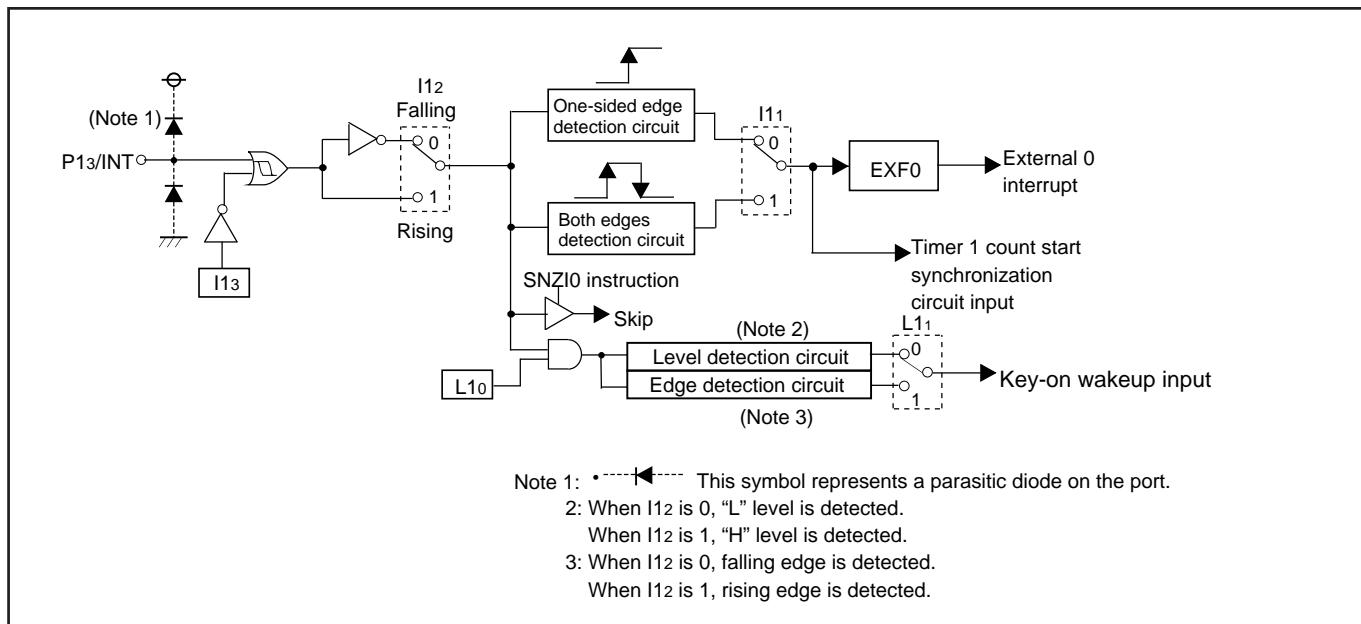
## EXTERNAL INTERRUPTS

The 4509 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	P13/INT	When the next waveform is input to P13/INT pin <ul style="list-style-type: none"> <li>• Falling waveform ("H"→"L")</li> <li>• Rising waveform ("L"→"H")</li> <li>• Both rising and falling waveforms</li> </ul>	I11 I12



**Fig. 17 External interrupt circuit structure**

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

External 0 interrupt request flag (EXF0) is set to "1" when a valid waveform is input to P13/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 P13/INT pin.

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

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

The external 0 interrupt is now enabled. Now when a valid waveform is input to the P13/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 : 00002		at RAM back-up : state retained	R/W TAI1/TI1A
I13	INT pin input control bit (Note 2)	0	INT pin input disabled		
		1	INT pin input enabled		
I12	Interrupt valid waveform for INT pin/return level selection bit (Note 2)	0	Falling waveform ("L" level 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		
I11	INT pin edge detection circuit control bit	0	One-sided edge detected		
I10	INT pin timer 1 control enable bit	0	Disabled		
		1	Enabled		

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

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

### (3) Notes on interrupts

#### ① Note [1] on bit 3 of register I1

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

- Depending on the input state of the P13/INT pin, the external 0 interrupt request flag (EXF0) may be set when the bit 3 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 18①) and then, change the bit 3 of register I1.

In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 18②).

Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 18③).

#### ③ Note [3] on bit 2 of register I1

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

- Depending on the input state of the P13/INT pin, the external 0 interrupt request flag (EXF0) may be set when the bit 2 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 20①) and then, change the bit 2 of register I1 is changed.

In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 20②).

Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 20③).

```
•
LA 4 ; (XXX02)
TV1A ; The SNZ0 instruction is valid .....①
LA 8 ; (1XXX2)
TI1A ; Control of INT pin input is changed
NOP .....②
SNZ0 ; The SNZ0 instruction is executed
      (EXF0 flag cleared)
NOP .....③
•
```

X : these bits are not used here.

Fig. 18 External 0 interrupt program example-1

#### ② Note [2] on bit 3 of register I1

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

- When the INT pin input is disabled (register I13 = "0"), set the key-on wakeup of INT pin to be invalid (register L10 = "0") before system enters to the RAM back-up mode. (refer to Figure 19①).

```
•
LA 0 ; (XXX02)
TI1A ; INT key-on wakeup disabled .....①
DI
EPOF
POF ; RAM back-up
•
X : these bits are not used here.
```

Fig. 19 External 0 interrupt program example-2

```
•
LA 4 ; (XXX02)
TV1A ; The SNZ0 instruction is valid .....①
LA 12 ; (1XXX2)
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 4509 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.

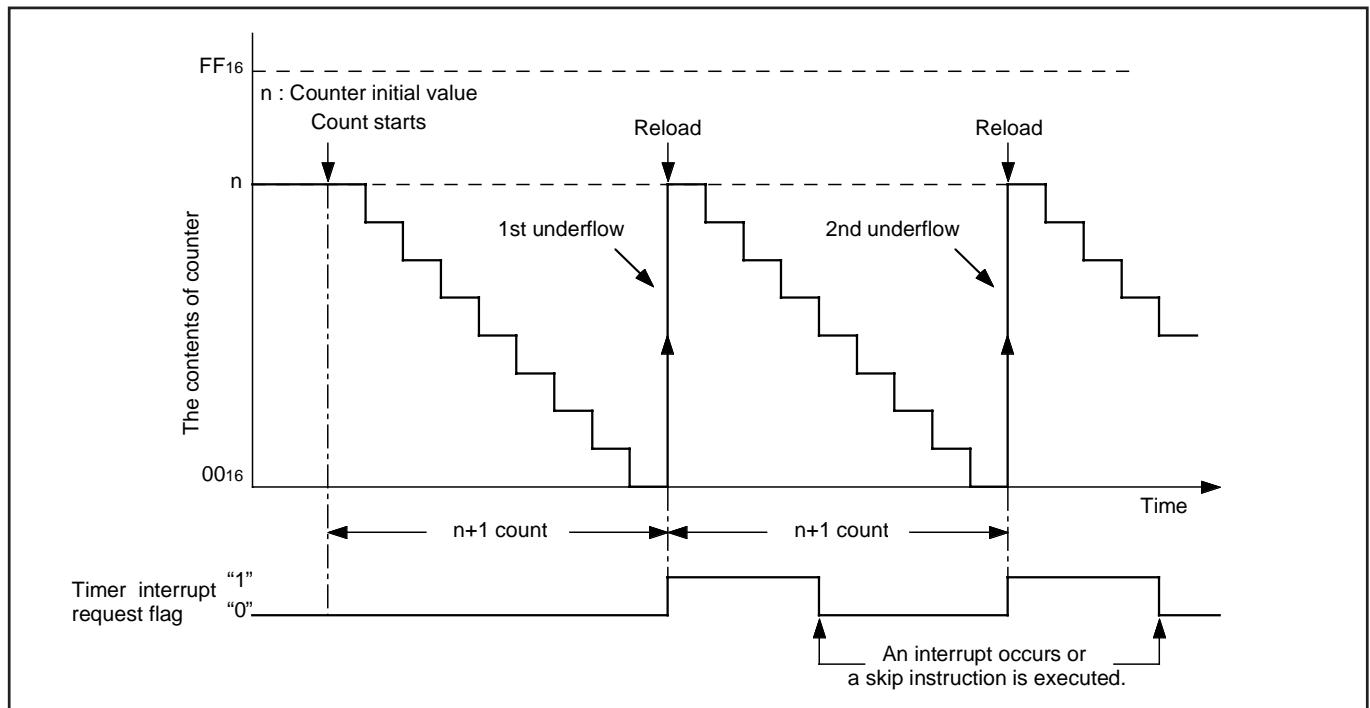


Fig. 21 Auto-reload function

The 4509 Group timer consists of the following circuits.

- Prescaler : 8-bit programmable timer
- 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 PA, W1, W2, W5 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	8-bit programmable binary down counter	• Instruction clock (INSTCK)	1 to 256	• Timer 1 and 2 count sources	PA
Timer 1	8-bit programmable binary down counter (link to INT input) (with PWM output function)	• PWM2 signal (PWMOD2) • Prescaler output (ORCLK) • CNTR1 input • On-chip oscillator clock (f(RING))	1 to 256	• Timer 2 count source • CNTR0 output • Timer 1 interrupt	W1 W5 W6
Timer 2	8-bit programmable binary down counter (INT input period count function) (with PWM output function)	• Timer 1 underflow (T1UDF) • Prescaler output (ORCLK) • CNTR0 input • System clock (STCK)	1 to 256	• Timer 1 count source • CNTR1 output • Timer 2 interrupt	W2 W5 W6
Watchdog timer	16-bit fixed dividing frequency	• Instruction clock (INSTCK)	65536	• System reset (counting twice) • Decision of flag WDF1	-

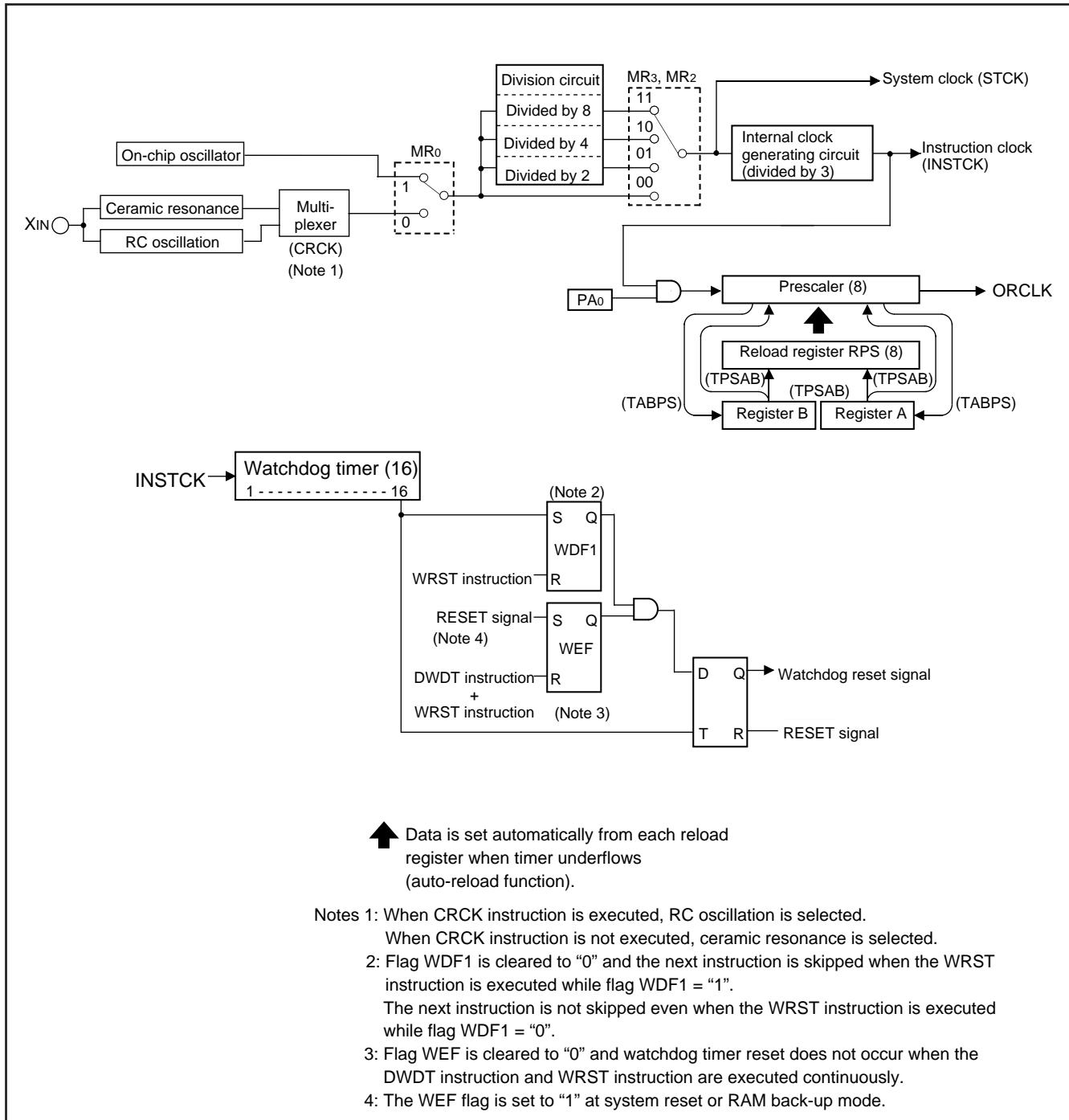


Fig. 22 Timers structure (1)

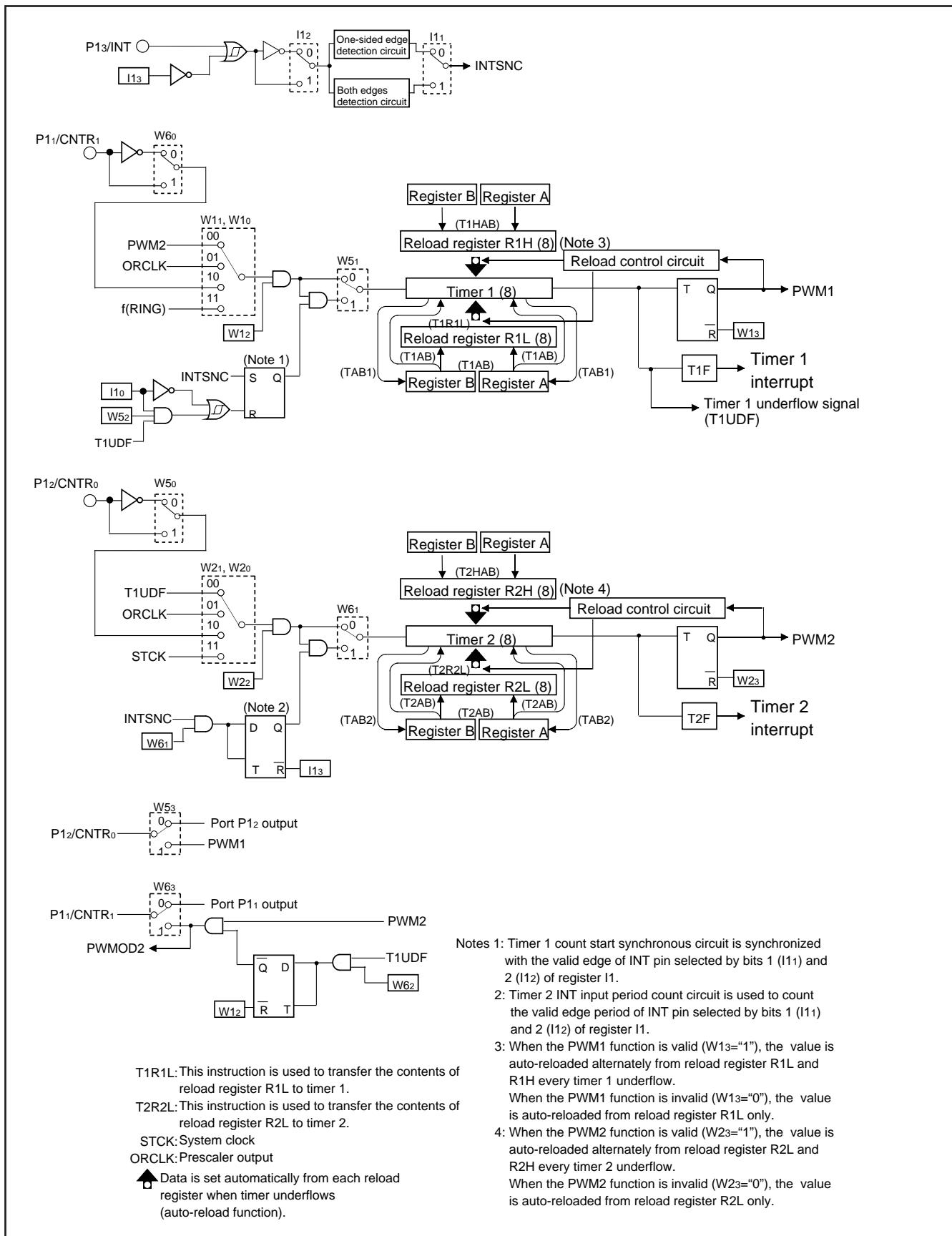


Fig. 23 Timers structure (2)

**Table 10 Timer control registers**

Timer control register PA		at reset : 02		at RAM back-up : 02	W TPAA
PA0	Prescaler control bit	0	Stop (state initialized)		
		1	Operating		

Timer control register W1		at reset : 00002		at RAM back-up : 00002	R/W TAW1/TW1A
W13	PWM1 function control bit	0	PWM1 function invalid		
		1	PWM1 function valid		
W12	Timer 1 control bit	0	Stop (state retained)		
		1	Operating		
W11	Timer 1 count source selection bits	W11	W10	Count source	
		0	0	PWM2 signal	
		0	1	Prescaler output (ORCLK)	
		1	0	CNTR1 input	
W10		1	1	On-chip oscillator clock (f(RING))	

Timer control register W2		at reset : 00002		at RAM back-up : 00002	R/W TAW2/TW2A
W23	PWM2 function control bit	0	PWM2 function invalid		
		1	PWM2 function valid		
W22	Timer 2 control bit	0	Stop (state retained)		
		1	Operating		
W21	Timer 2 count source selection bits	W21	W20	Count source	
		0	0	Timer 1 underflow signal (T1UDF)	
		0	1	Prescaler output (ORCLK)	
		1	0	CNTR0 input	
W20		1	1	System clock (STCK)	

Timer control register W5		at reset : 00002		at RAM back-up : state retained	R/W TAW5/TW5A
W53	P12/CNTR0 pin function selection bit	0	P12 (I/O) / CNTR0 (input)		
		1	P12 (input) /CNTR0 (I/O)		
W52	Timer 1 count auto-stop circuit selection bit (Note 2)	0	Count auto-stop circuit not selected		
		1	Count auto-stop circuit selected		
W51	Timer 1 count start synchronous circuit selection bit (Note 3)	0	Count start synchronous circuit not selected		
		1	Count start synchronous circuit selected		
W50	CNTR0 pin input count edge selection bit	0	Falling edge		
		1	Rising edge		

Timer control register W6		at reset : 00002		at RAM back-up : state retained	R/W TAW6/TW6A
W63	P11/CNTR1 pin function selection bit	0	P11 (I/O) / CNTR1 (input)		
		1	P11 (input) /CNTR1 (I/O)		
W62	CNTR 1 pin output auto-control circuit selection bit	0	Output auto-control circuit not selected		
		1	Output auto-control circuit selected		
W61	Timer 2 INT pin input period count circuit selection bit	0	INT pin input period count circuit not selected		
		1	INT pin input period count circuit selected		
W60	CNTR1 pin input count edge selection bit	0	Falling edge		
		1	Rising edge		

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

2: This function is valid only when the INT pin/timer 1 control is enabled (I10="1") and the timer 1 count start synchronous circuit is selected (W51="1").

3: This function is valid only when the INT pin/timer 1 control is enabled (I10="1").

## (1) Timer control registers

- Timer control register PA

Register PA controls the count operation of prescaler. Set the contents of this register through register A with the TPAA instruction.

- Timer control register W1

Register W1 controls the count operation and count source of timer 1, and PWM1 function. 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 count operation and count source of timer 2, and PWM2 function. 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 W5

Register W5 controls the input count edge of CNTR0 pin, timer 1 count start synchronous circuit, timer 1 auto-stop circuit and P12/CNTR0 pin function. Set the contents of this register through register A with the TW5A instruction. The TAW5 instruction can be used to transfer the contents of register W5 to register A.

- Timer control register W6

Register W6 controls the input count edge of CNTR1 pin, the INT pin input count start synchronous circuit and CNTR1 pin output auto-control circuit and the P11/CNTR1 pin function. 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 an 8-bit binary down counter with the prescaler reload register RPS. Data can be set simultaneously in prescaler and the reload register RPS with the TPSAB instruction. Data can be read from reload register RPS with the TABPS instruction.

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

Prescaler starts counting after the following process;

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

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

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

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

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

## (3) Timer 1 (interrupt function)

Timer 1 is an 8-bit binary down counter with two timer 1 reload registers (R1L, R1H). Data can be set simultaneously in timer 1 and the reload register R1L with the T1AB instruction. Data can be set in the reload register R1H with the T1HAB instruction. The contents of reload register R1L set with the T1AB instruction can be set to timer 1 again with the T1R1L instruction. Data can be read from timer 1 with the TAB1 instruction.

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

When executing the T1HAB instruction to set data to reload register R1H while timer 1 is operating, avoid a timing when timer 1 underflows.

Timer 1 starts counting after the following process;

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

When a value set in reload register R1L is n and a value set in reload register R1H is m, timer 1 divides the count source signal by  $n + 1$  or  $m + 1$  ( $n = 0$  to 255,  $m = 0$  to 255).

<Bit 3 of register W1 = "0" (PWM1 function invalid)>

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 R1L, and count continues (auto-reload function).

<Bit 3 of register W1 = "1" (PWM1 function valid)>

Timer 1 generates the PWM1 signal of the "L" interval set as reload register R1L, and the "H" interval set as reload register R1H. The PWM1 signal generated by timer 1 is output from CNTR0 pin by setting "1" to bit 3 of register W5.

After timer 1 control by INT pin is enabled by setting the bit 0 of register I1 to "1," INT pin input can be used as the start trigger for timer 1 count operation by setting the bit 1 of register W5 to "1".

Also, in this time, the auto-stop function by timer 1 underflow can be performed by setting the bit 2 of register W5 to "1."

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

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

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

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

Timer 2 starts counting after the following process;

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

When a value set in reload register R2L is n and a value set in reload register R2H is m, timer 2 divides the count source signal by  $n + 1$  or  $m + 1$  ( $n = 0$  to 255,  $m = 0$  to 255).

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

<Bit 3 of register W2 = "0" (PWM2 function invalid)>

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

<Bit 3 of register W2 = "1" (PWM2 function valid)>

Timer 2 generates the PWM2 signal of the "L" interval set as reload register R2L, and the "H" interval set as reload register R2H. The PWM2 signal generated by timer 2 is output from CNTR1 pin by setting "1" to bit 3 of register W6.

PWM2 output to CNTR1 pin combined with timer 1 can be controlled by setting the bit 2 of register W6 to "1."

Input period of INT pin by timer 2 can be counted by setting the bit 1 of register W6 to "1."

#### (5) 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 can be selected after timer 1 control by INT pin is enabled by setting the bit 0 of register I1 to "1" and its function is selected by setting the bit 1 of register W5 to "1".

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

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

Once set, the count start synchronous circuit is cleared by clearing the bit I1 to "0" or system reset.

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

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

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

The count auto-stop circuit is valid by setting the bit 2 of register W5 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.

#### (7) INT pin input period count circuit (timer 2)

Timer 2 has the INT pin input period count circuit to count the valid waveform input interval of the INT pin.

When bit 1 of register W6 is set to "1", the INT pin input period count circuit of timer 2 becomes valid, and the count source is input. The count source input is stopped by the next input of valid waveform to the INT pin.

Then, every a valid waveform is input to the INT pin, start/stop of the count source input is alternately repeated.

A valid waveform of the INT pin input is the same as the activated condition of an external interrupt.

The INT pin input period count circuit set once is cleared by setting the INT pin input to be disabled state. The INT pin input can be disabled by clearing bit 3 of register I1 to "0".

#### (8) Timer input/output pin (P12/CNTR0 pin, P11/CNTR1 pin)

CNTR0 pin is used to input the timer 2 count source and output the PWM1 signal generated by timer 1.

CNTR1 pin is used to input the timer 1 count source and output the PWM2 signal generated by timer 2.

The P12/CNTR0 pin function can be selected by bit 3 of register W5.

The P11/CNTR1 pin function can be selected by bit 3 of register W6.

When the CNTR0 input is selected for timer 2 count source, timer 2 counts the falling or rising waveform of CNTR0 input. The count edge is selected by bit 0 of register W5.

When the CNTR1 input is selected for timer 1 count source, timer 1 counts the falling or rising waveform of CNTR1 input. The count edge is selected by bit 0 of register W6.

## (9) PWM1 output function (P12/CNTR0, timer 1)

When bit 3 of register W1 is set to "1", the data is reloaded alternately from reload register R1L and R1H every timer 1 underflow.

Timer 1 generates the PWM1 signal of the "L" interval set as reload register R1L, and the "H" interval set as reload register R1H.

In this time, the PWM1 signal generated by timer 1 is output from CNTR0 pin by setting "1" to bit 3 of register W5.

When the TW1A instruction is executed while the PWM1 signal is "H", the contents of register W1 is changed after the "H" interval of the PWM1 signal is ended.

## (10) PWM2 output function (P11/CNTR1, timer 1, timer 2)

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

Timer 2 generates the PWM2 signal of the "L" interval set as reload register R2L, and the "H" interval set as reload register R2H.

In this time, the PWM2 signal generated by timer 2 is output from CNTR1 pin by setting "1" to bit 3 of register W6.

When bit 2 of register W6 is set to "1", the PWM2 signal output to CNTR1 pin is switched to valid/invalid alternately each timer 1 underflow. However, when timer 1 is stopped (bit 2 of register W1 is cleared to "0"), this function is canceled.

When the TW2A instruction is executed while the PWM2 signal is "H", the contents of register W2 is changed after the "H" interval of the PWM2 signal is ended.

## (11) 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, V2 to select an interrupt or a skip instruction.

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

## (12) Precautions

### - Prescaler

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

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

### - Timer 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, T1R1L, T2AB or T2R2L instruction to write data to timer.

### - Writing to reload register

In order to write a data to the reload register R1H while the timer 1 is operating, execute the T1HAB instruction except a timing of the timer 1 underflow.

In order to write a data to the reload register R2H while the timer 2 is operating, execute the T2HAB instruction except a timing of the timer 2 underflow.

### - PWM signal (PWM1, PWM2)

If the timer 1 count stop timing and the timer 1 underflow timing overlap during output of the PWM1 signal, a hazard may occur in the PWM1 output waveform.

If the timer 2 count stop timing and the timer 2 underflow timing overlap during output of the PWM2 signal, a hazard may occur in the PWM2 output waveform.

### - Prescaler, timer 1 and timer 2 count start timing and count time when operation starts

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

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

When selecting CNTR input as the count source of timer, timer operates synchronizing with the count edge (falling edge or rising edge) of CNTR input selected by software.

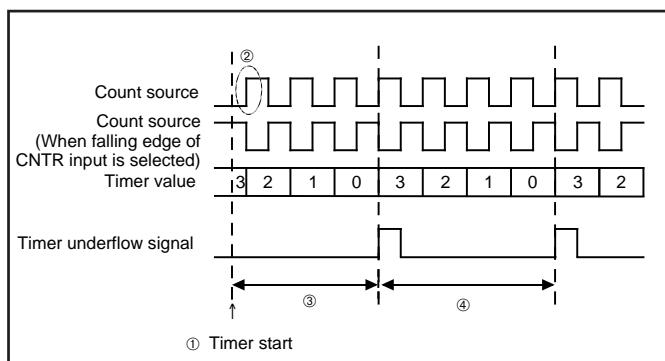
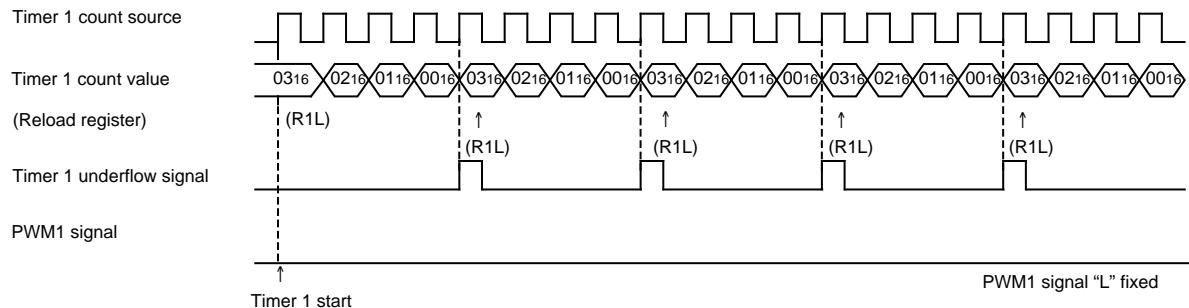
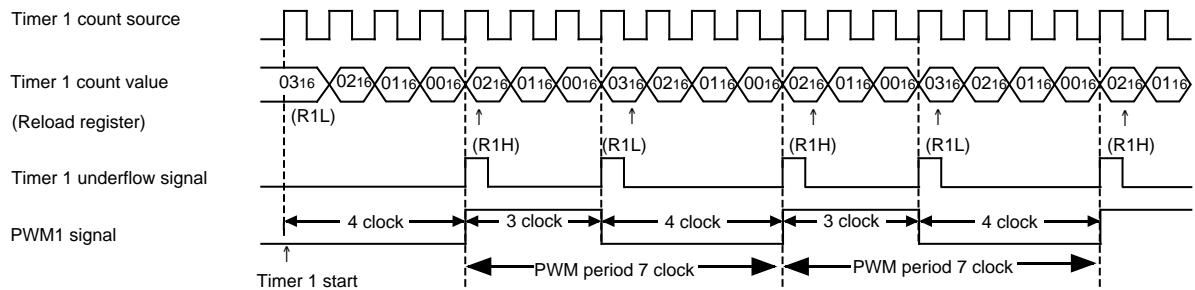


Fig. 24 Timer count start timing and count time when operation starts

- PWM1 function invalid (W13 = "0")



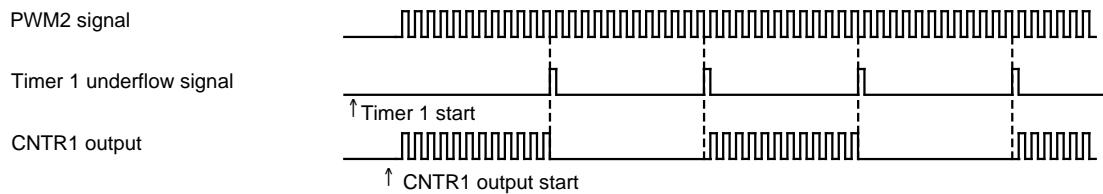
- PWM1 function valid (W13 = "1")



\* : "0316" is set to reload register R1L and "0216" is set to reload register R1H.

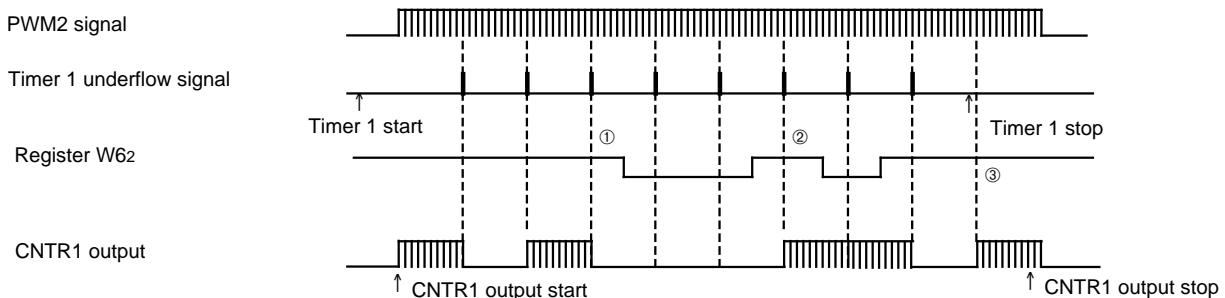
Fig. 25 Timer 1 operation example

● CNTR1 output auto-control circuit operation example 1 (W23 = "1", W63 = "1", W62 = "1")



\* When the CNTR1 output auto-control circuit is selected, valid/invalid of CNTR1 output is repeated every timer 1 underflows.

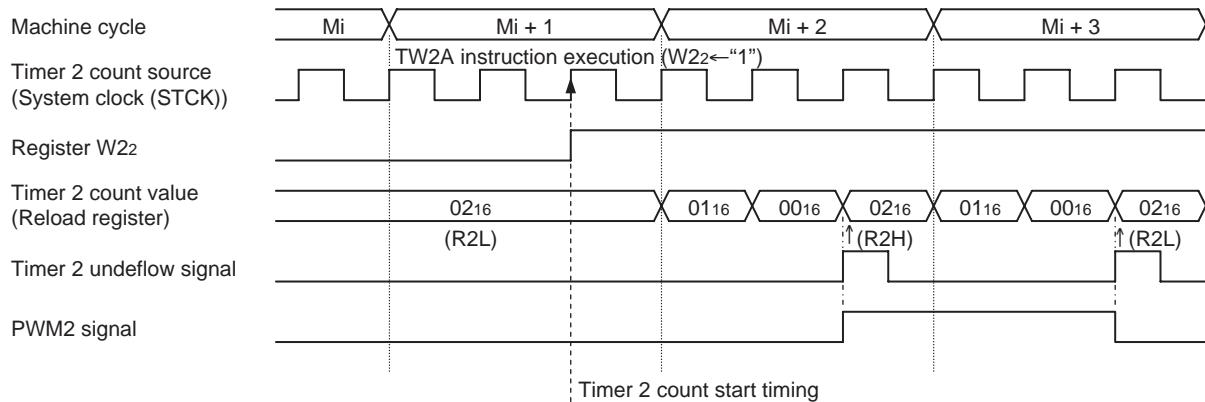
● CNTR1 output auto-control circuit operation example 2 (W23 = "1", W63 = "1")



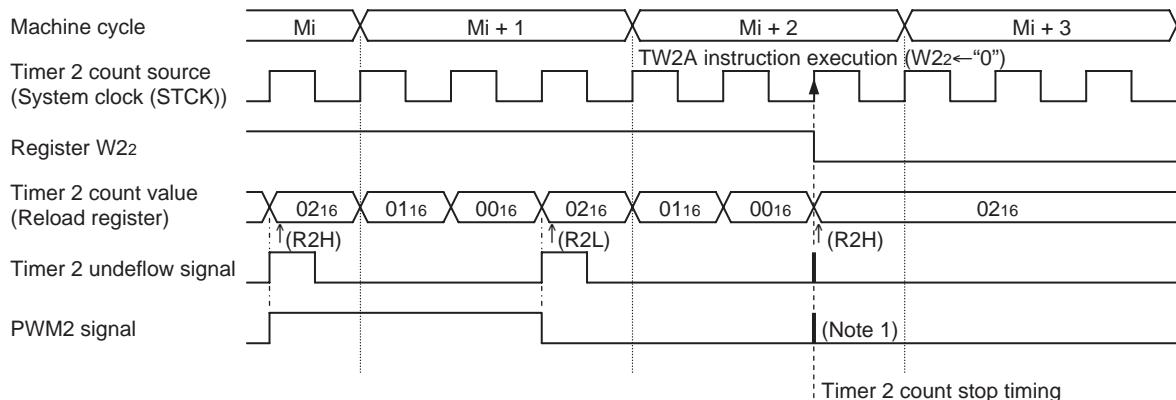
- ① When the CNTR1 output auto-control function is not selected while the CNTR output is invalid, CNTR1 output invalid state is retained.
- ② When the CNTR1 output auto-control function is not selected while the CNTR output is valid, CNTR1 output valid state is retained.
- ③ When the timer 1 is stopped, the CNTR1 output auto-control function becomes invalid.

Fig. 26 CNTR1 output auto-control function by timer 1

- Timer 2 count start timing (R2L = "0216", R2H = "0216", W23 = "1")



- Timer 2 count stop timing (R2L = "0216", R2H = "0216", W23 = "1")



Notes 1: If the timer count stop timing and the timer underflow timing overlap while the PWM function is valid ( $W13=1$  or  $W23=1$ ), a hazard may occur in the PWM signal waveform.

2: When timer count is stopped during "H" duration of the PWM signal, timer is stopped after the end of the "H" output duration.

Fig. 27 Timer count start/stop timing

## WATCHDOG TIMER

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

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

After the count is started, when the timer WDT underflow occurs (after the count value of timer WDT reaches "FFFF16," the next count pulse is input), the WDF1 flag is set to "1."

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

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

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

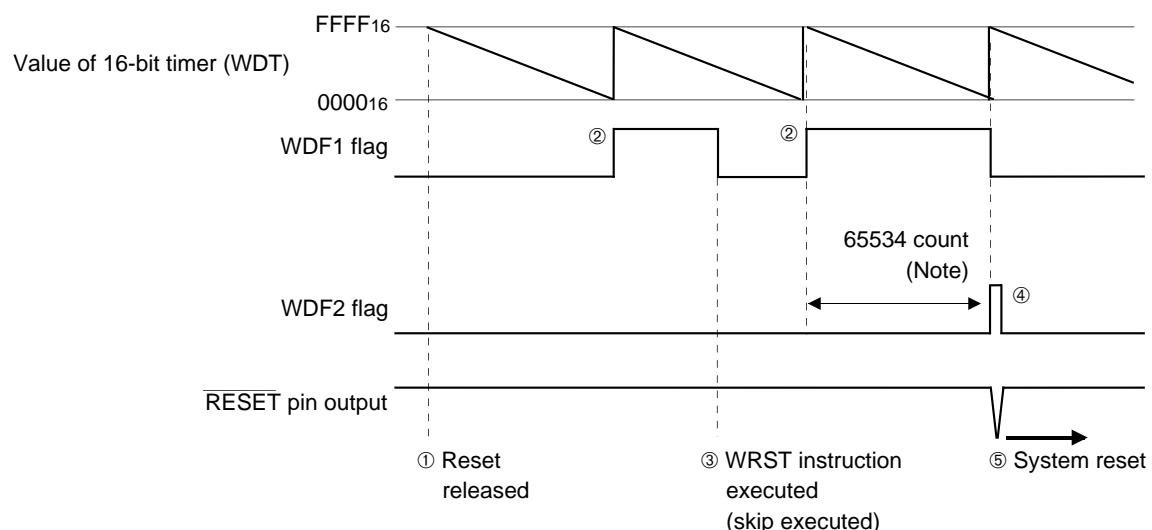
When the DWDT instruction and the WRST instruction are executed continuously, the WEF flag is cleared to "0" and the watchdog timer function is invalid.

The WEF flag is set to "1" at system reset or RAM back-up mode.

The WRST instruction has the skip function. When the WRST instruction is executed while the WDF1 flag is "1," the WDF1 flag is cleared to "0" and the next instruction is skipped.

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

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



- 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 while the WDF1 flag is "1", 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 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 machine cycle because the count source of watchdog timer is the instruction clock.

Fig. 28 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 29).

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 30).

Also, set the NOP instruction after the WRST instruction, for the case when a skip is performed with the WRST instruction.

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

Fig. 29 Program example to start/stop watchdog timer

```
⋮
WRST      ; WDF1 flag cleared
NOP
DI        ; Interrupt disabled
EPOF      ; POF instruction enabled
POF       ; RAM back-up mode
↓
Oscillation stop
⋮
```

Fig. 30 Program example to enter the RAM back-up mode when using the watchdog timer

## A/D CONVERTER

The 4509 Group has a built-in A/D conversion circuit that performs conversion by 10-bit successive comparison method. Table 11 shows the characteristics of this A/D converter. This A/D converter can also be used as an 8-bit comparator to compare analog voltages input from the analog input pin with preset values.

**Table 11 A/D converter characteristics**

Parameter	Characteristics
Conversion format	Successive comparison method
Resolution	10 bits
Relative accuracy	Linearity error: $\pm 2$ LSB (VDD=2.7 to 5.5 V) Differential non-linearity error: $\pm 0.9$ LSB (VDD=2.7 to 5.5 V)
Conversion speed	31 $\mu$ s (f(XIN)=6 MHz, f(STCK)=f(XIN))
Analog input pin	6

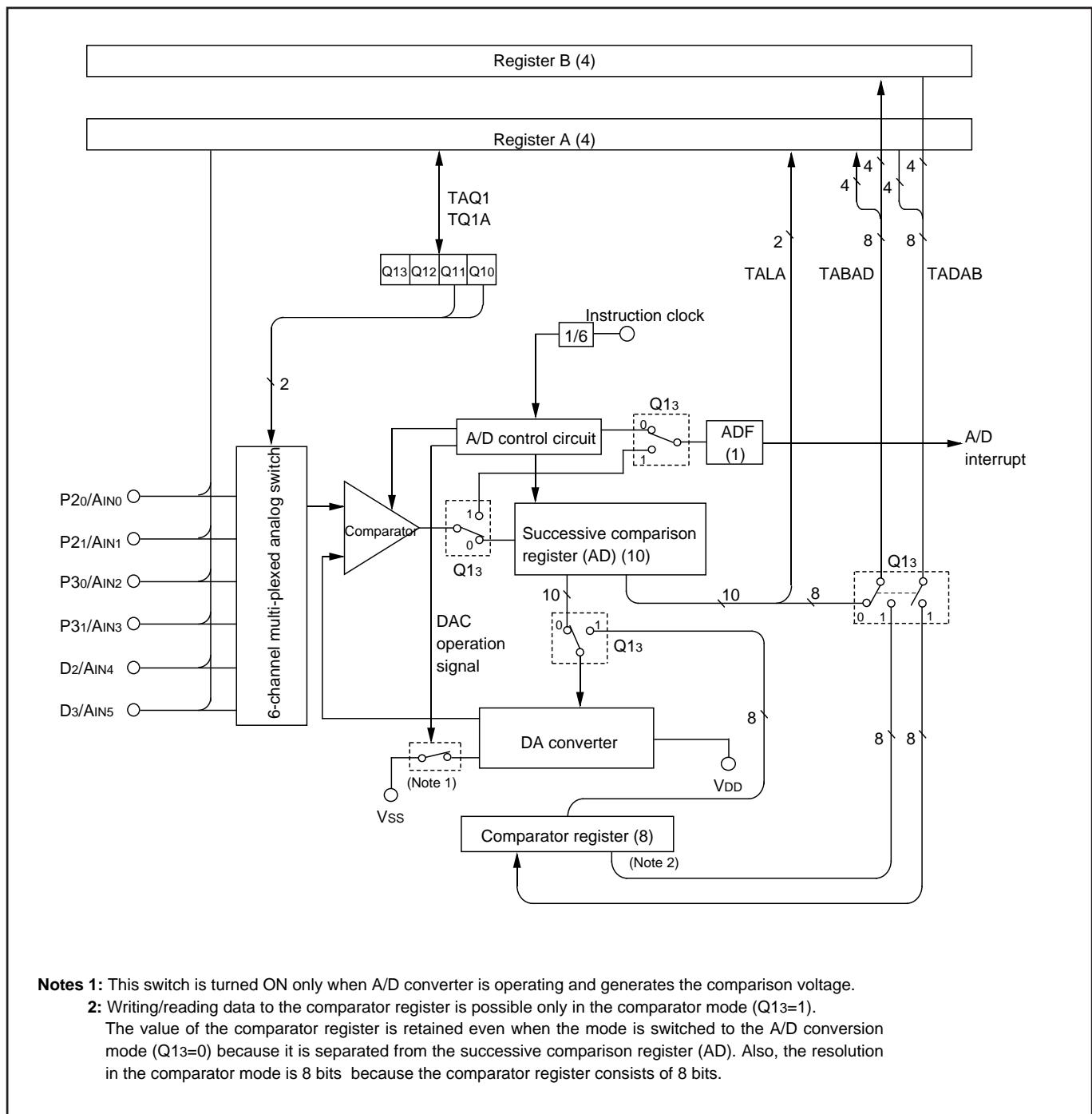


Fig. 31 A/D conversion circuit structure

**Table 12 A/D control registers**

A/D control register Q1		at reset : 00002			at RAM back-up : state retained	R/W TAQ1/TQ1A
Q13	A/D operation mode selection bit	0		A/D conversion mode		
		1		Comparator mode		
Q12	Analog input pin selection bits	Q12	Q11	Q10	Selected pins	
		0	0	0	AIN0	
		0	0	1	AIN1	
		0	1	0	AIN2	
		0	1	1	AIN3	
		1	0	0	AIN4	
		1	0	1	AIN5	
		1	1	0	Not available	
		1	1	1	Not available	

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

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

Register Q1 is used to select the operation mode and one of analog input pins. Set the contents of this register through register A with the TQ1A instruction. The TAQ1 instruction can be used to transfer the contents of register Q1 to register A.

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

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

### (3) 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 VDD 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)

### (4) 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.

### (5) 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.

### (6) Operation description

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

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

The 4509 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 (31  $\mu$ s when  $f(XIN) = 6.0$  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 32).

**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	$\frac{V_{DD}}{2} \pm \dots \pm \frac{V_{DD}}{1024}$

\*1: 1st comparison result

\*2: 2nd comparison result

\*3: 3rd comparison result

\*8: 8th comparison result

\*9: 9th comparison result

\*A: 10th comparison result

## (7) A/D conversion timing chart

Figure 32 shows the A/D conversion timing chart.

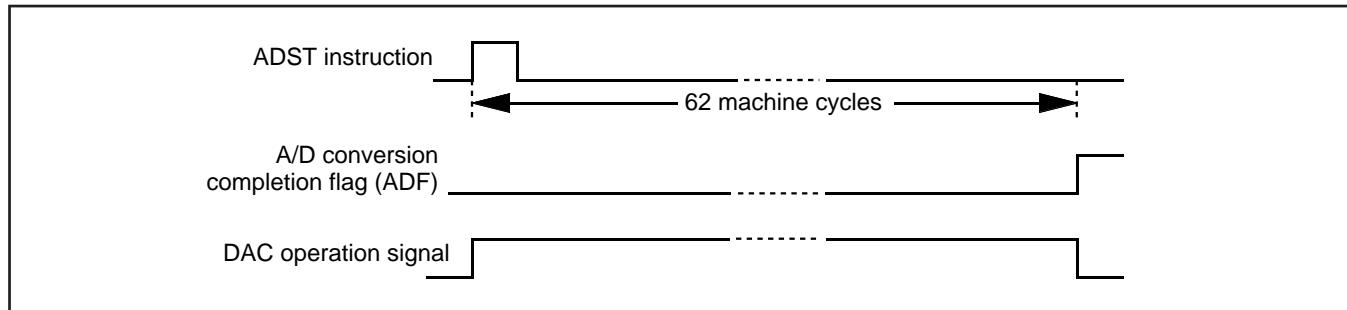


Fig. 32 A/D conversion timing chart

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

How to use A/D conversion is explained using as example in which the analog input from P20/A<sub>IN0</sub> 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 A<sub>IN0</sub> pin function and A/D conversion mode with the register Q1 (refer to Figure 33).
- 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).

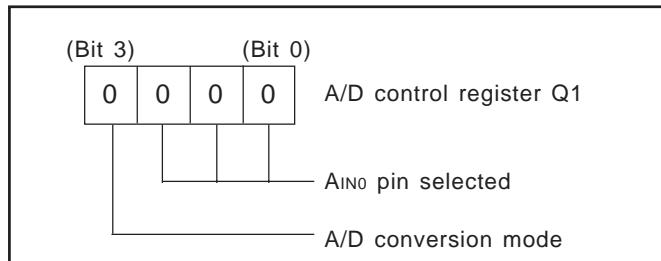


Fig. 33 Setting registers

## (9) Operation at comparator mode

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

Below, the operation at comparator mode is described.

## (10) Comparator register

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

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

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

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

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  $\mu$ s at  $f(XIN) = 4.0$  MHz in high-speed through 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

- 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."

- Operating mode of A/D converter

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.

Clear the bit 2 of register V2 to "0" to change the operating mode from the comparator mode to A/D conversion mode.

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.

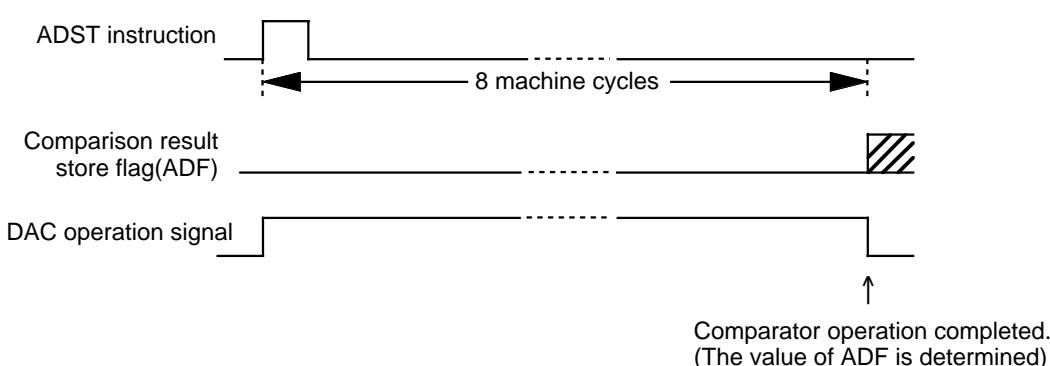


Fig. 34 Comparator operation timing chart

## (14) Definition of A/D converter accuracy

The A/D conversion accuracy is defined below (refer to Figure 35).

- 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)

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

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

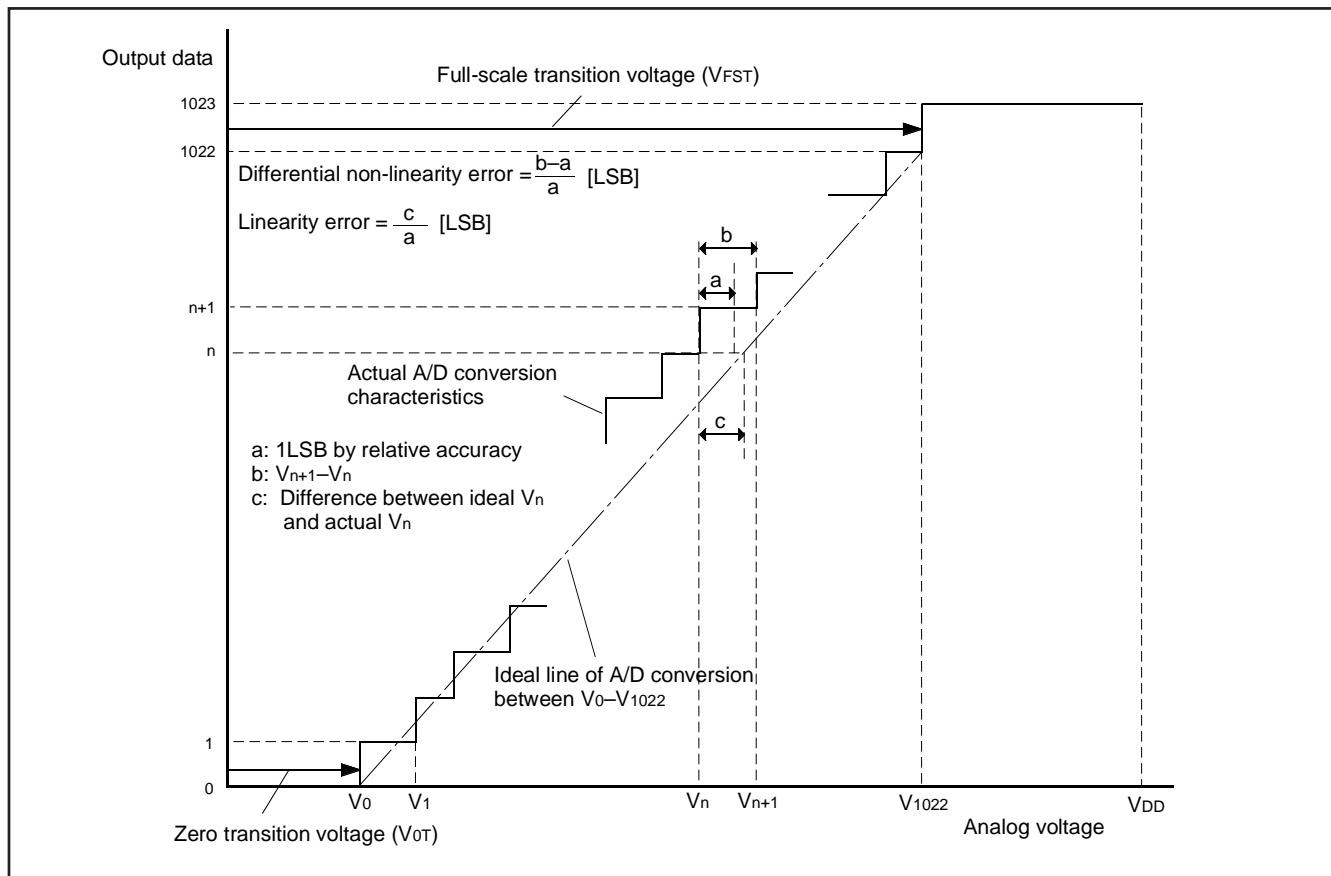


Fig. 35 Definition of A/D conversion accuracy

## SERIAL INTERFACE

The 4509 Group has a built-in clock synchronous serial interface which can serially transmit or receive 8-bit data.

Serial interface consists of;

- Serial interface register SI
- Serial interface control register J1
- Serial interface transmit/receive completion flag (SIOF)
- Serial interface counter

Registers A and B are used to perform data transfer with internal CPU.

The pin functions of the serial interface pins can be set with the register J1.

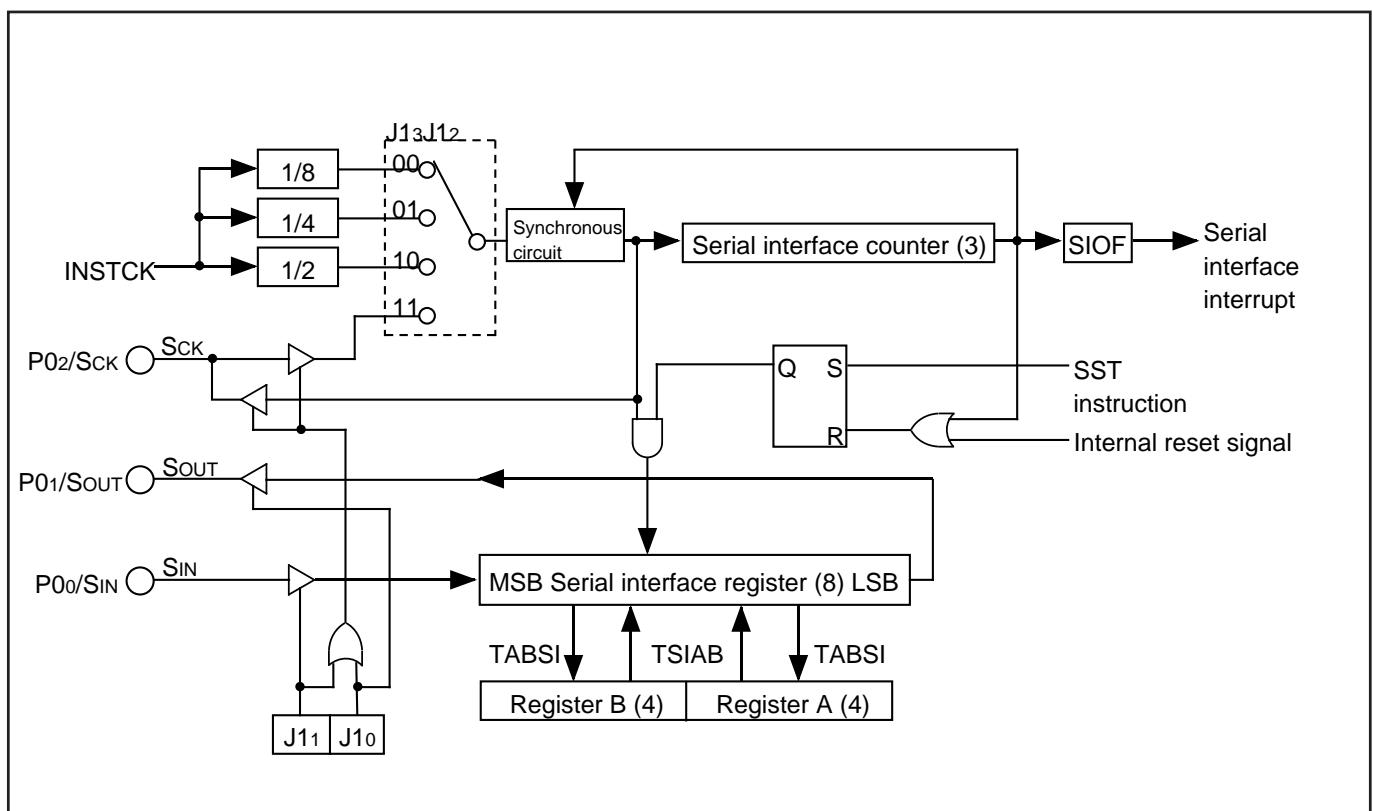
**Table 14 Serial interface pins**

Pin	Pin function when selecting serial interface
P02/SCK	Clock I/O (Sck)
P01/SOUT	Serial data output (SOUT)
P03/SIN	Serial data input (SIN)

Note: Even when the SIN pin function is used, the I/O of port P00 is valid.

Even when the SOUT pin function is used, the input of port P01 is valid.

The input of P02 can be used even when Sck is used. Be careful when using inputs of both Sck and P02 since the input threshold value of Sck pin is different from that of port P02.



**Fig. 36 Serial interface structure**

**Table 15 Serial interface control register**

Serial interface control register J1		at reset : 00002		at RAM back-up : state retained		R/W	
J13	Serial interface synchronous clock selection bits	J13	J12	Synchronous clock			
		0	0	Instruction clock (INSTCK) divided by 8			
		0	1	Instruction clock (INSTCK) divided by 4			
		1	0	Instruction clock (INSTCK) divided by 2			
J12		1	1	External clock (Sck input)			
		J11 J10		Port function			
J11		0	0	P00, P01, P02 selected/SIN, SOUT, SCK not selected			
		0	1	P00, SOUT, SCK selected/SIN, P01, P02 not selected			
		1	0	SIN, P01, SCK selected/P00, SOUT, P02 not selected			
J10	Serial interface port function selection bits	1	1	SIN, SOUT, SCK selected/P00, P01, P02 not selected			

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

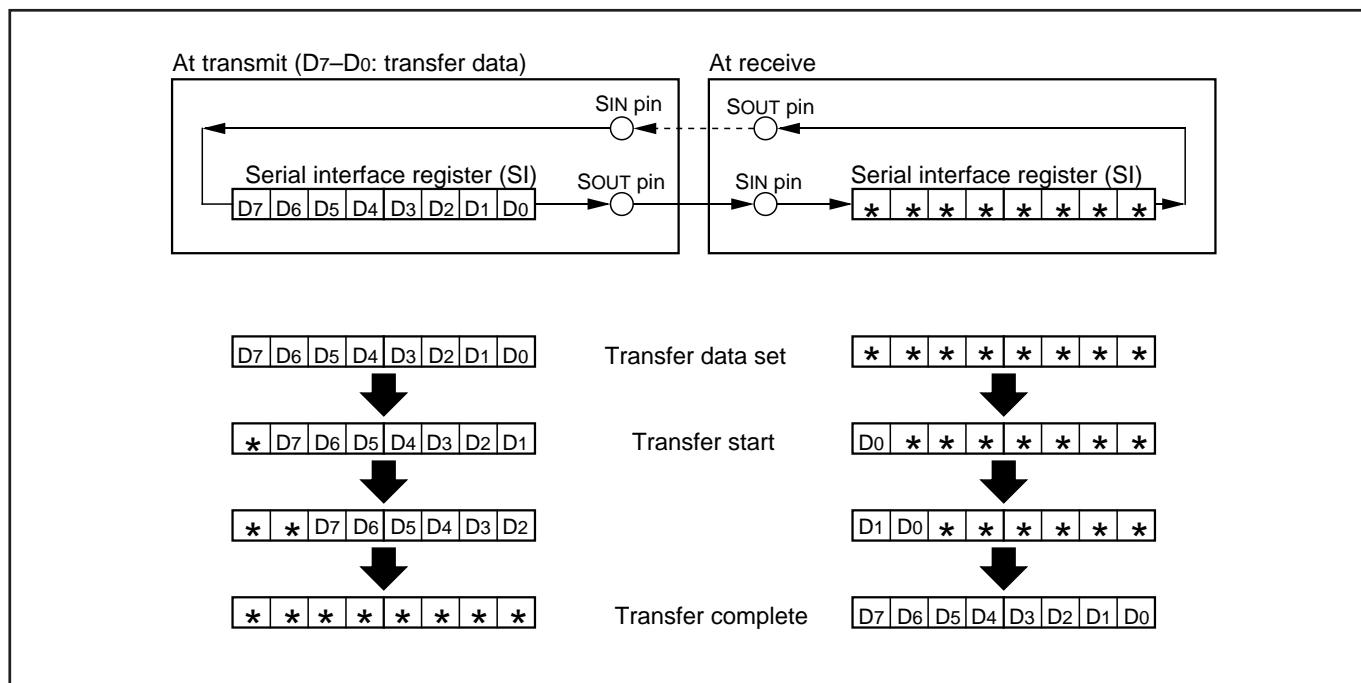


Fig. 37 Serial interface register state when transferring

### (1) Serial interface register SI

Serial interface register SI is the 8-bit data transfer serial/parallel conversion register. Data can be set to register SI through registers A and B with the TSIAB instruction. The contents of register A is transmitted to the low-order 4 bits of register SI, and the contents of register B is transmitted to the high-order 4 bits of register SI. During transmission, each bit data is transmitted LSB first from the lowermost bit (bit 0) of register SI, and during reception, each bit data is received LSB first to register SI starting from the topmost bit (bit 7).

When register SI is used as a work register without using serial interface, do not select the Sck pin.

### (2) Serial interface transmit/receive completion flag (SIOF)

Serial interface transmit/receive completion flag (SIOF) is set to "1" when serial data transmission or reception completes. The state of SIOF flag can be examined with the skip instruction (SNZSI). Use the interrupt control register V2 to select the interrupt or the skip instruction.

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

### (3) Serial interface start instruction (SST)

When the SST instruction is executed, the SIOF flag is cleared to "0" and then serial interface transmission/reception is started.

### (4) Serial interface control register J1

Register J1 controls the synchronous clock, P02/Sck, P01/Sout and P00/Sin pin function. Set the contents of this register through register A with the TJ1A instruction. The TAJ1 instruction can be used to transfer the contents of register J1 to register A.

## (5) How to use serial interface

Figure 38 shows the serial interface connection example. Serial interface interrupt is not used in this example. In the actual wiring, pull

up the wiring between each pin with a resistor. Figure 38 shows the data transfer timing and Table 16 shows the data transfer sequence.

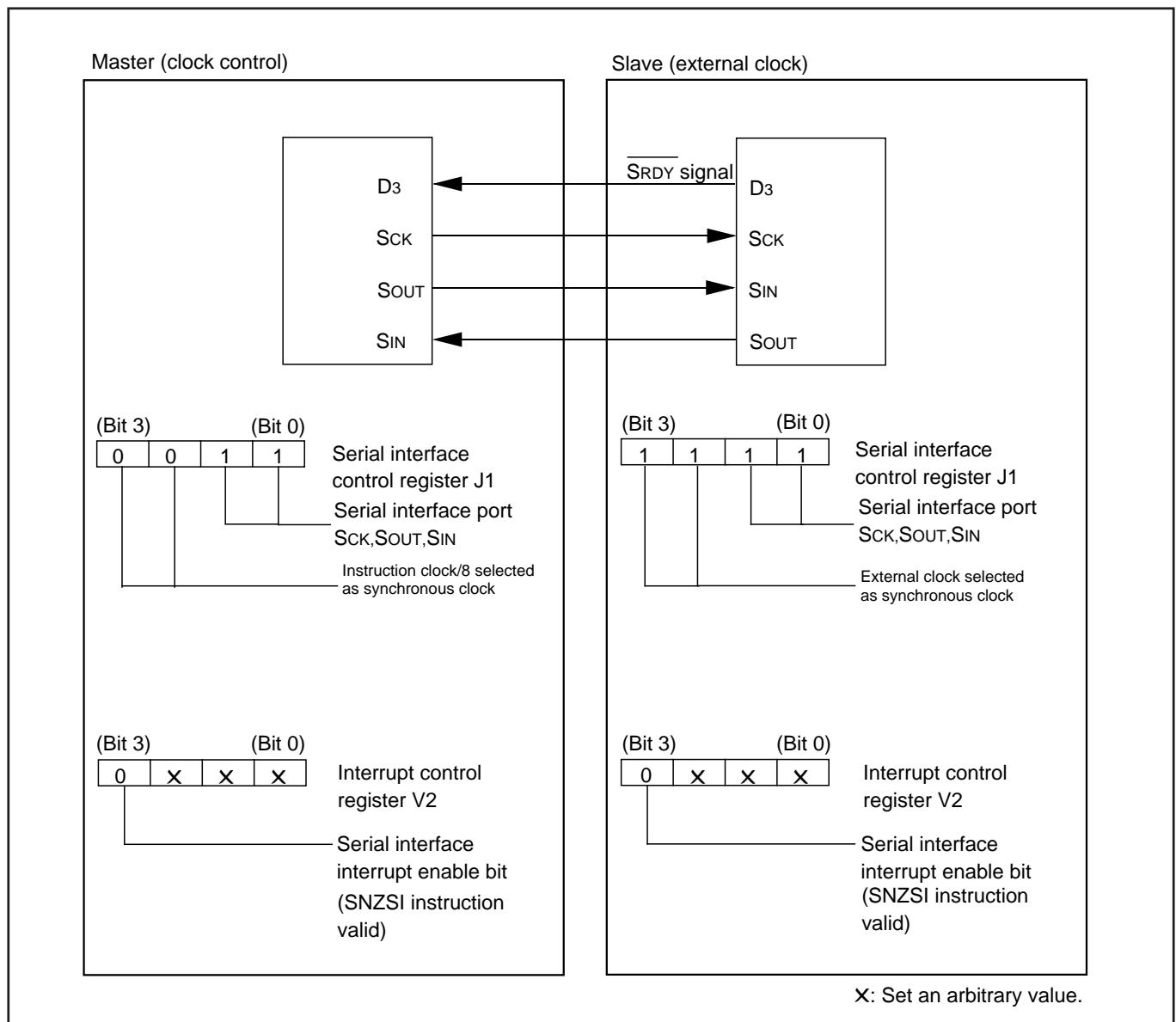


Fig. 38 Serial interface connection example

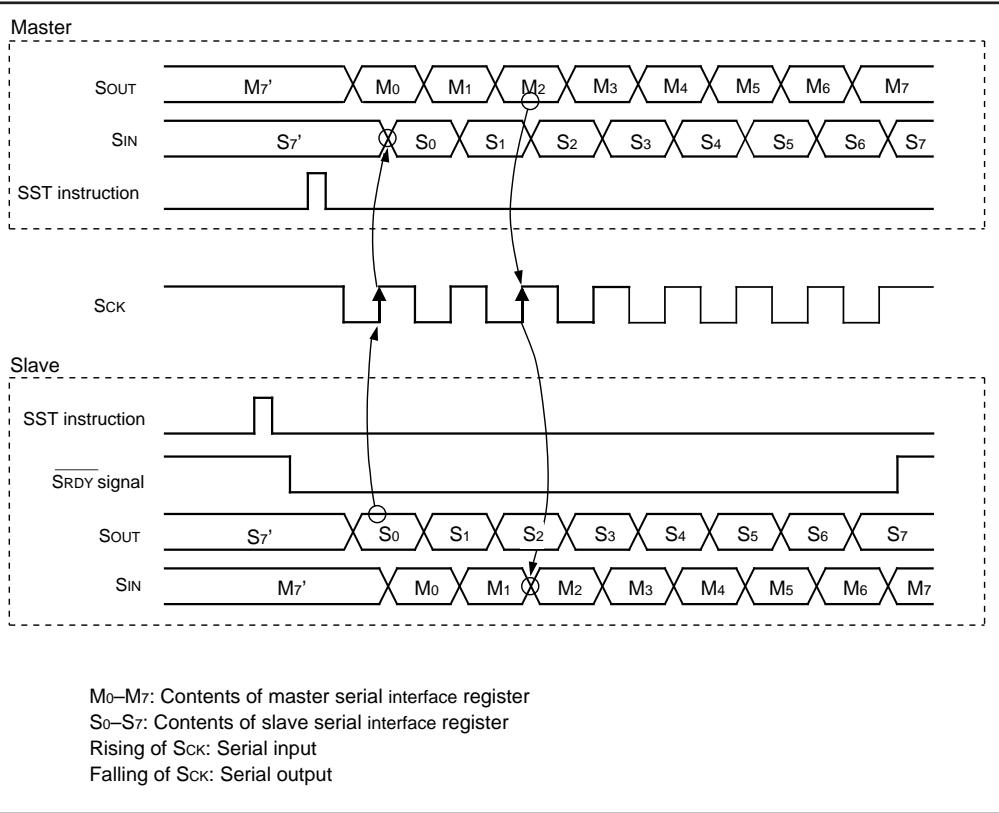


Fig. 39 Timing of serial interface data transfer

Table 16 Processing sequence of data transfer from master to slave

Master (transmission)	Slave (reception)
[Initial setting] <ul style="list-style-type: none"> <li>Setting the serial interface control register J1 and interrupt control register V2 shown in Figure 38.</li> </ul>	[Initial setting] <ul style="list-style-type: none"> <li>Setting serial interface control register J1, and interrupt control register V2 shown in Figure 38.</li> </ul>
TJ1A and TV2A instructions	TJ1A and TV2A instructions
<ul style="list-style-type: none"> <li>Setting the port received the reception enable signal (SRDY) to the input mode. (Port D3 is used in this example)</li> </ul>	<ul style="list-style-type: none"> <li>Setting the port transmitted the reception enable signal (SRDY) and outputting "H" level. (Port D3 is used in this example)</li> </ul>
SD instruction	SD instruction
* [Transmission enable state] <ul style="list-style-type: none"> <li>Storing transmission data to serial interface register SI.</li> </ul>	* [Reception enable state] <ul style="list-style-type: none"> <li>The SIOF flag is cleared to "0."</li> </ul>
TSIAB instruction	SST instruction <ul style="list-style-type: none"> <li>"L" level (reception possible) is output from port D3.</li> </ul>
	RD instruction
[Transmission] <ul style="list-style-type: none"> <li>Check port D3 is "L" level.</li> </ul>	[Reception]
SZD instruction	
*Serial transfer starts.	
SST instruction	
*Check transmission completes.	<ul style="list-style-type: none"> <li>Check reception completes.</li> </ul>
SNZSI instruction	SNZSI instruction
*Wait (timing when continuously transferring)	<ul style="list-style-type: none"> <li>"H" level is output from port D3.</li> </ul>
	SD instruction
	[Data processing]

1-byte data is serially transferred on this process. Subsequently, data can be transferred continuously by repeating the process from \*.

When an external clock is selected as a synchronous clock, control the clock externally because serial transfer is performed as long as clock is externally input. (Unlike an internal clock, an external clock is not stopped when serial transfer is completed.) However, the

SIOF flag is set to "1" when the clock is counted 8 times after executing the SST instruction. Be sure to set the initial level of the external clock to "H."

## RESET FUNCTION

System reset is performed by the followings:

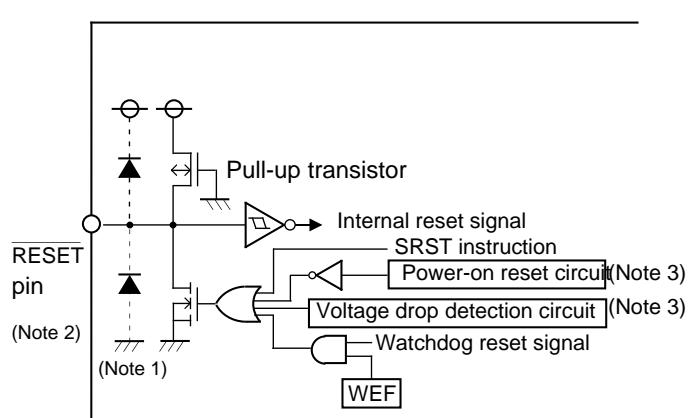
- "L" level is applied to the  $\overline{\text{RESET}}$  pin externally,
- System reset instruction (SRST) is executed,
- Reset occurs by watchdog timer,
- Reset occurs by built-in power-on reset (only for H version)
- Reset occurs by voltage drop detection circuit (only for H version)

Then when "H" level is applied to  $\overline{\text{RESET}}$  pin, software starts from address 0 in page 0.

### (1) RESET pin input

System reset is performed certainly by applying "L" level to  $\overline{\text{RESET}}$  pin for 1 machine cycle or more when the following condition is satisfied;

the value of supply voltage is the minimum value or more of the recommended operating conditions.



**Notes 1:** --- This symbol represents a parasitic diode.

**2:** Applied potential to  $\overline{\text{RESET}}$  pin must be  $\text{VDD}$  or less.

**3:** These are equipped with only H version.

Fig. 40 Structure of reset pin and its peripherals

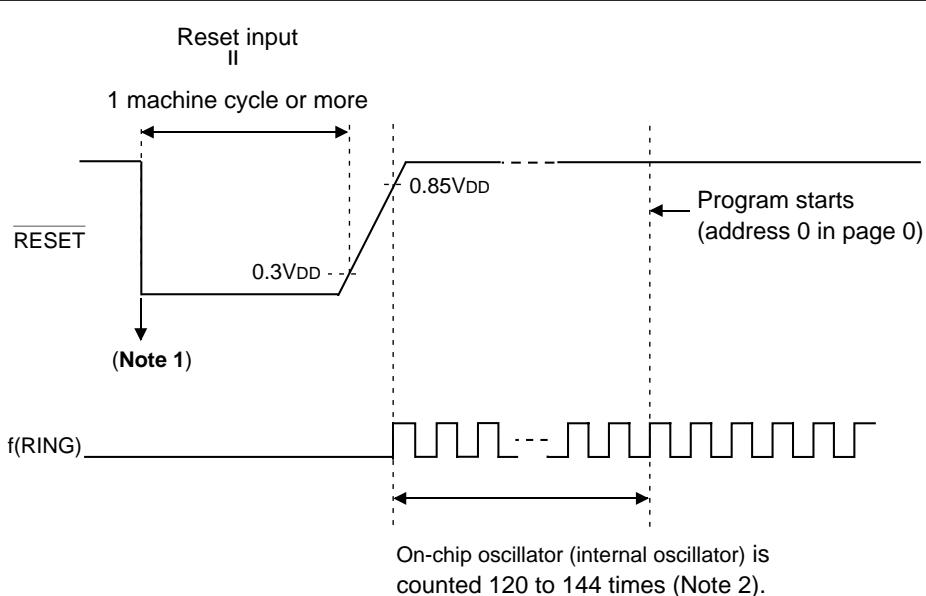


Fig. 41 RESET pin input waveform and reset release timing

## (2) Power-on reset (only for H version)

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

If the rising time exceeds 100  $\mu$ s, connect a capacitor between the RESET pin and V<sub>SS</sub> at the shortest distance, and input "L" level to RESET pin until the value of supply voltage reaches the minimum operating voltage.

## (3) System reset instruction (SRST)

By executing the SRST instruction, "L" level is output to RESET pin and system reset is performed.

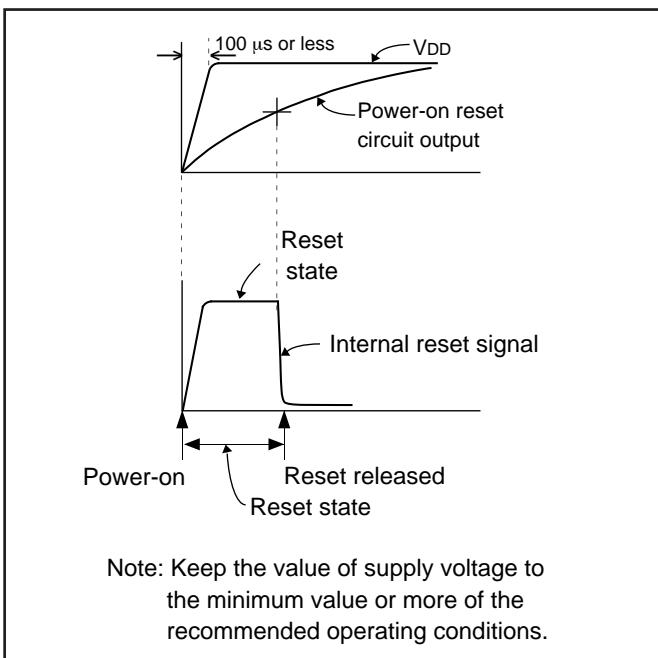


Fig. 42 Power-on reset operation

Table 17 Port state at reset

Name	Function	State
D0, D1	D0, D1	High-impedance (Notes 1, 2)
D2/AIN4, D3/AIN5	D2, D3	High-impedance (Notes 1, 2, 3)
D4, D5	D4, D5	High-impedance (Notes 1, 2)
P00/SIN, P01/SOUT, P02/SCK	P00, P01, P02	High-impedance (Notes 1, 2, 3)
P03	P03	High-impedance (Notes 1, 2, 3)
P10	P10	High-impedance (Notes 1, 2, 3)
P11/CNTR1	P11	High-impedance (Notes 1, 2, 3)
P12/CNTR0	P12	High-impedance (Notes 1, 2, 3)
P13/INT	P13	High-impedance (Notes 1, 2, 3)
P20/AIN0, P21/AIN1	P20, P21	High-impedance (Notes 1, 2, 3)
P30/AIN2, P31/AIN3	P30, P31	High-impedance (Notes 1, 2)

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

2: The output structure is N-channel open-drain.

3: Pull-up transistor is turned OFF.

#### (4) Internal state at reset

Figure 43 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 43 are undefined, so set the initial value to them.

• Program counter (PC) .....	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Address 0 in page 0 is set to program counter.	
• Interrupt enable flag (INTE) .....	0 (Interrupt disabled)
• Power down flag (P) .....	0
• External 0 interrupt request flag (EXF0) .....	0
• Interrupt control register V1 .....	0 0 0 0 (Interrupt disabled)
• Interrupt control register V2 .....	0 0 0 0 (Interrupt disabled)
• Interrupt control register I1 .....	0 0 0 0
• Timer 1 interrupt request flag (T1F) .....	0
• Timer 2 interrupt request flag (T2F) .....	0
• Watchdog timer flags (WDF1, WDF2) .....	0
• Watchdog timer enable flag (WEF) .....	1
• Timer control register PA .....	0 (Prescaler stopped)
• Timer control register W1 .....	0 0 0 0 (Timer 1 stopped)
• Timer control register W2 .....	0 0 0 0 (Timer 2 stopped)
• Timer control register W5 .....	0 0 0 0
• Timer control register W6 .....	0 0 0 0
• Clock control register MR .....	1 1 0 1
• Clock control register RG .....	0 (On-chip oscillator operating)
• Serial interface transmit/receive completion flag (SIOF) .....	0
• Serial interface control register J1 .....	0 0 0 0 (Serial interface port not selected)
• Serial interface register SI .....	X X X X X X X X X X
• A/D conversion completion flag (ADF) .....	0
• A/D control register Q1 .....	0 0 0 0
• Successive comparison register AD .....	X X X X X X X X X X
• Comparator register .....	X X X X X X X X X X
• 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
• Key-on wakeup control register L1 .....	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
• Port output structure control register FR0 .....	0 0 0 0
• Port output structure control register FR1 .....	0 0 0 0
• Port output structure control register FR2 .....	0 0 0 0
• Port output structure control register FR3 .....	0 0 0 0
• Port output structure control register C1 .....	0 0 0 0
• Carry flag (CY) .....	0
• Register A .....	0 0 0 0
• Register B .....	0 0 0 0
• Register D .....	X X X
• Register E .....	X X X X X X X X X X
• Register X .....	0 0 0 0
• Register Y .....	0 0 0 0
• Register Z .....	X X
• Stack pointer (SP) .....	1 1 1
• Operation source clock .....	On-chip oscillator (operating)
• Ceramic resonator circuit .....	Operating
• RC oscillation circuit .....	Stop

"X" represents undefined.

Fig. 43 Internal state at reset

## VOLTAGE DROP DETECTION CIRCUIT (only for H version)

The built-in voltage drop detection circuit is designed to detect a drop in voltage and to reset the microcomputer by outputting "L" level to RESET pin if the supply voltage drops below a set value.

### (1) SVDE instruction

If the SVDE instruction is not executed (initial state), the voltage drop detection circuit becomes invalid at RAM back-up mode. When the SVDE instruction is executed, the voltage drop detection circuit is valid even after system enters into the RAM back-up mode. The SVDE instruction can be executed only once.

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

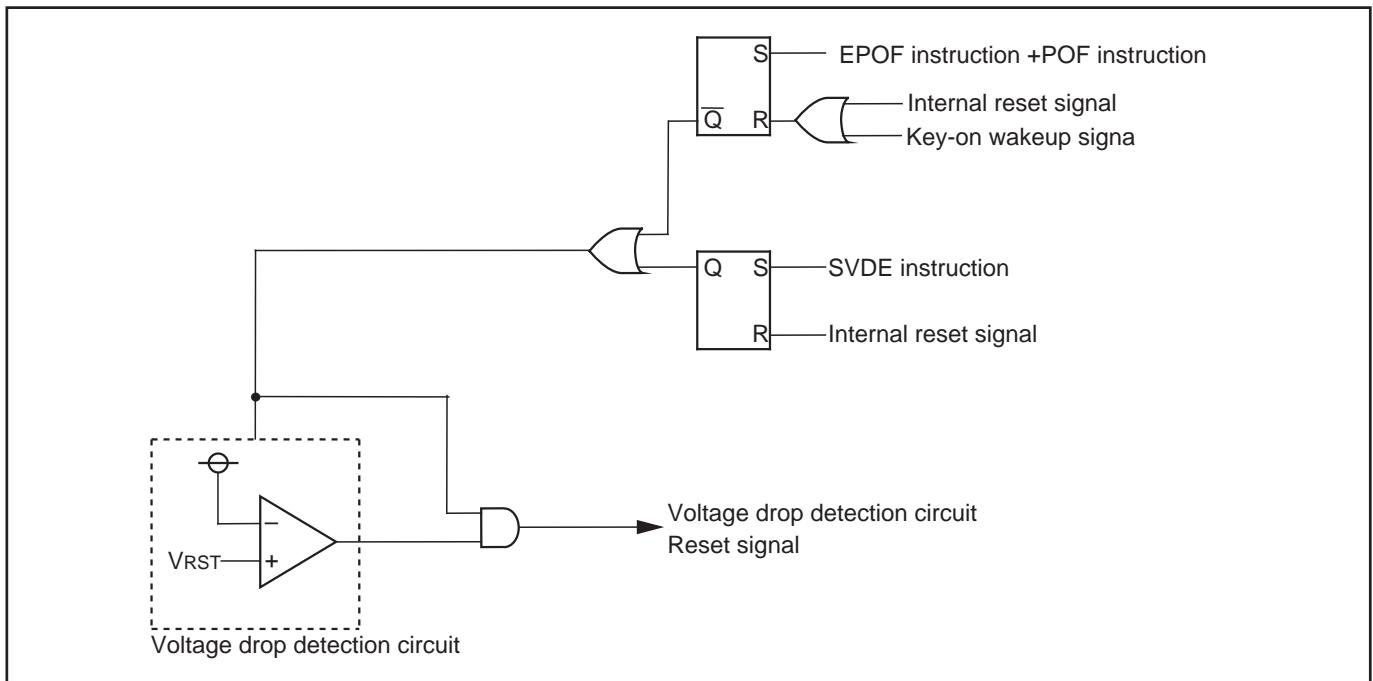


Fig. 44 Voltage drop detection reset circuit

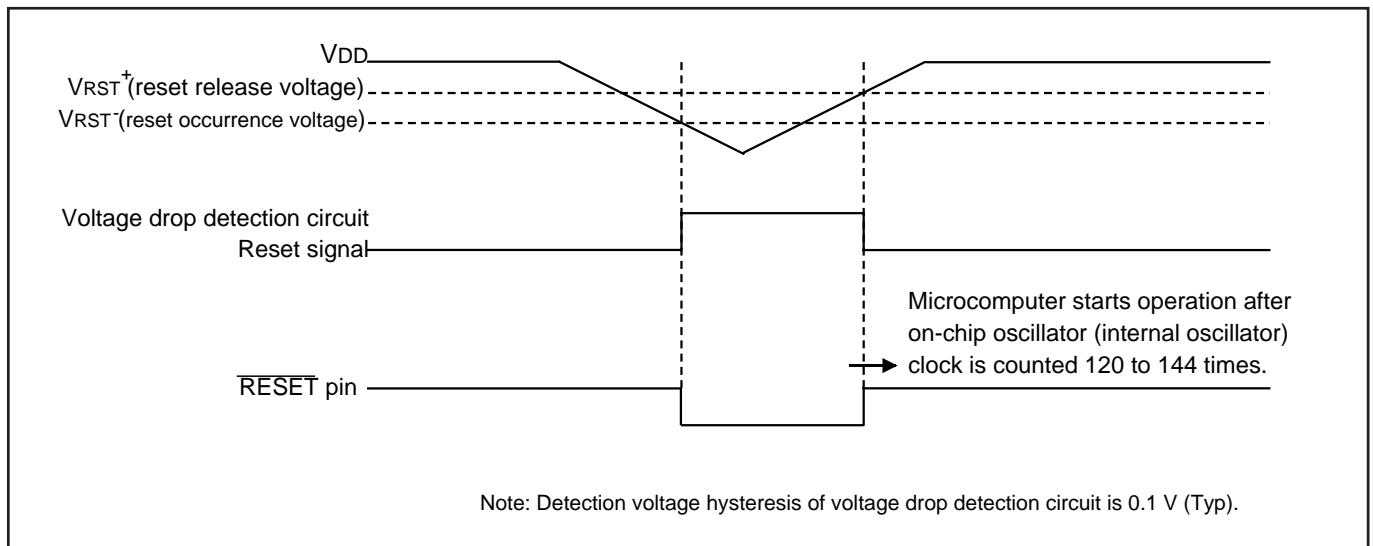


Fig. 45 Voltage drop detection circuit operation waveform

Table 18 Voltage drop detection circuit operation state

	At CPU operating	At RAM back-up mode
SVDE instruction not executed	Valid	Invalid
SVDE instruction executed	Valid	Valid

## RAM BACK-UP MODE

The 4509 Group has the RAM back-up mode.

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

The POF instruction is equal to the NOP instruction when the EPOF instruction is not executed before the POF 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 19 shows the function and states retained at RAM back-up. Figure 46 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 POF 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;

- "L" level is applied to RESET pin,
- system reset (SRST) is performed,
- reset by watchdog timer is performed,
- reset by the built-in power-on reset circuit is performed (only for H version), or
- reset by the voltage drop detection circuit is performed (only for H version).

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

**Table 19 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
Interrupt control registers V1, V2	X
Interrupt control register I1	O
Selected oscillation circuit (execution of CRCK)	O
Clock control register MR	X
Clock control register RG	X
Timer 1, Timer 2 function	(Note 3)
Watchdog timer function	X (Note 4)
Timer control register PA	X
Timer control registers W1, W2	X
Timer control registers W5, W6	O
Serial interface function	X
Serial interface control register J1	O
A/D conversion function	X
A/D control register Q1	O
Voltage drop detection circuit	(Note 5)
Port level	O
Key-on wakeup control registers K0 to K2, L1	O
Pull-up control registers PU0 to PU2	O
Port output structure control registers FR0 to FR3, C1	O
External interrupt request flag (EXFO)	X
Timer interrupt request flags (T1F, T2F)	(Note 3)
A/D conversion completion flag (ADF)	X
Serial interface transmit/receive completion flag (SIOF)	X
Interrupt enable flag (INTE)	X
Watchdog timer flags (WDF1, WDF2)	X (Note 4)
Watchdog timer enable flag (WEF)	X (Note 4)

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

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

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

3: The state of the timer is undefined.

4: Initialize the watchdog timer flag WDF1 with the WRST instruction, and then set the system to be in the RAM back-up mode.

5: The voltage drop detection circuit is equipped with only H version. In the RAM back-up mode, when the SVDE instruction is not executed, the voltage drop detection circuit is invalid, and when the SVDE instruction is executed, the voltage drop detection circuit is valid.

#### (4) Return signal

An external wakeup signal is used to return from the RAM back-up mode because the oscillation is stopped. Table 20 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 K1 to register A.

- Key-on wakeup control register K2

Register K2 controls the ports P2, D2 and D3 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.

- Key-on wakeup control register L1

Register L1 controls the selection of the return condition and valid waveform/level of port P1, and the selection of the INT pin return condition and INT pin key-on wakeup function. Set the contents of this register through register A with the TL1A instruction. In addition, the TAL1 instruction can be used to transfer the contents of register L1 to register A.

- Pull-up control register PU0

Register PU0 controls the ON/OFF of the port P0 pull-up transistor. Set the contents of this register through register A with the TPU0A instruction. In addition, the TAK1 instruction can be used to transfer the contents of register K0 to register A.

- Pull-up control register PU1

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

- Pull-up control register PU2

Register PU2 controls the ON/OFF of the ports P2, D2 and D3 pull-up transistor. Set the contents of this register through register A with the TPU2A instruction. In addition, the TAPU2 instruction can be used to transfer the contents of register PU2 to register A.

- Interrupt control register I1

Register I1 controls the valid waveform/level of the external 0 interrupt and the input control of INT pin. 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 20 Return source and return condition**

Return source	Return condition	Remarks
External wakeup signal	Port P00–P03 Port P20, P21 Port D2, D3	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 P10–P13	Return by an external "H" level or "L" level input, or falling edge ("H"→"L") or rising edge ("L"→"H"). The key-on wakeup function can be selected by one port unit. Select the return level ("L" level or "H" level) and return condition (level or edge) with the register L1 according to the external state before going into the RAM back-up state. Before going into the RAM backup state, set an opposite level of the selected return level (edge) to the port using the key-on wakeup function.
	INT pin	Return by an external "H" level or "L" level input, or falling edge ("H"→"L") or rising edge ("L"→"H"). When the return level is input, the EXF0 flag is not set. The key-on wakeup function can be selected by one port unit. Select the return level ("L" level or "H" level) with the register I1 and return condition (level or edge) with the register L1 according to the external state before going into the RAM back-up state.

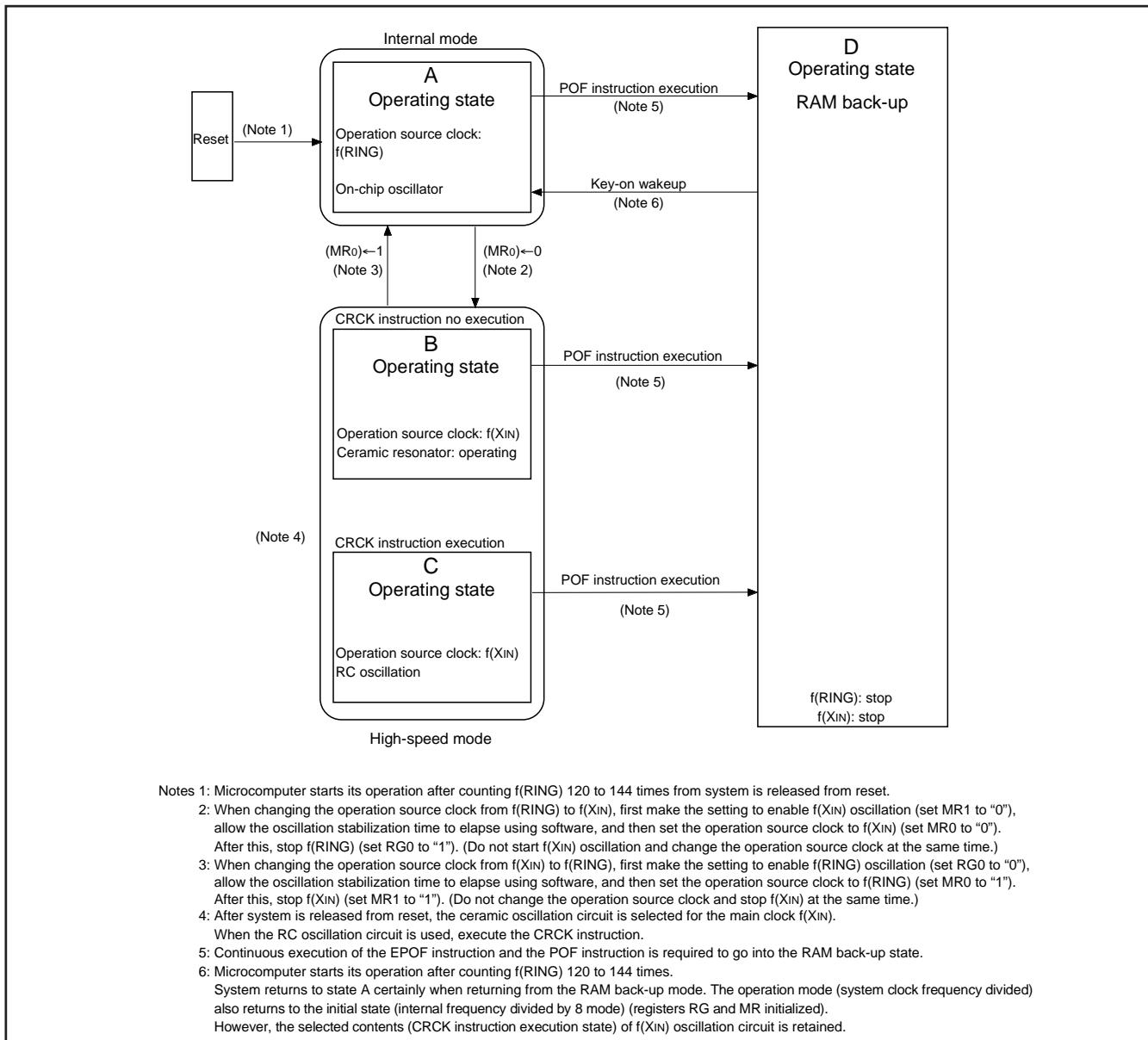


Fig. 46 State transition

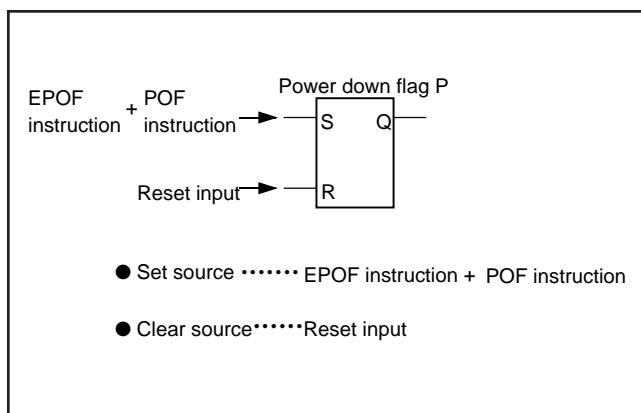


Fig. 47 Set source and clear source of the P flag

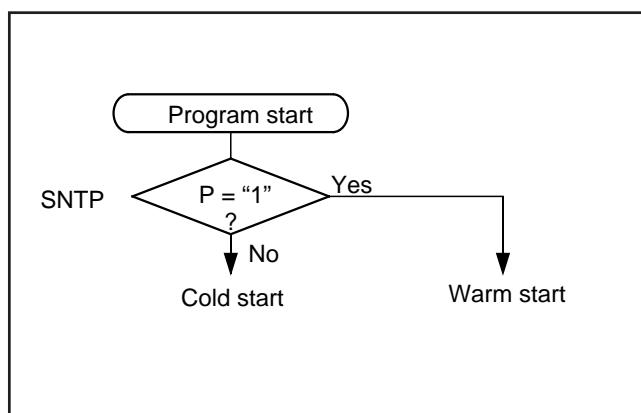


Fig. 48 Start condition identified example using the SNZP instruction

**Table 21 Key-on wakeup control register**

Key-on wakeup control register K0		at reset : 00002		at RAM back-up : state retained	R/W TAK0/TK0A
K03	Port P03 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K02	Port P02 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K01	Port P01 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K00	Port P00 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K1		at reset : 00002		at RAM back-up : state retained	R/W TAK1/TK1A
K13	Port P13 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K12	Port P12 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K11	Port P11 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K10	Port P10 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K2		at reset : 00002		at RAM back-up : state retained	R/W TAK2/TK2A
K23	Port D3 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K22	Port D2 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K21	Port P21 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K20	Port P20 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register L1		at reset : 00002		at RAM back-up : state retained	R/W TAL1/TL1A
L13	Ports P10–P13 return condition selection bit	0	Return by level		
		1	Return by edge		
L12	Ports P10–P13 valid waveform/level selection bit	0	Falling waveform/"L" level		
		1	Rising waveform/"H" level		
L11	INT pin return condition selection bit	0	Return by level		
		1	Return by edge		
L10	INT pin key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

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

**Table 22 Pull-up control register and interrupt control register**

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

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

Pull-up control register PU2		at reset : 00002		at RAM back-up : state retained	R/W TAPU2/TPU2A
PU23	Port D3 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU22	Port D2 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU21	Port P21 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU20	Port P20 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.

## CLOCK CONTROL

The clock control circuit consists of the following circuits.

- On-chip oscillator (internal oscillator)
- Ceramic oscillation circuit
- 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 49 shows the structure of the clock control circuit.

The 4509 Group operates by the on-chip 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 4509 Group.

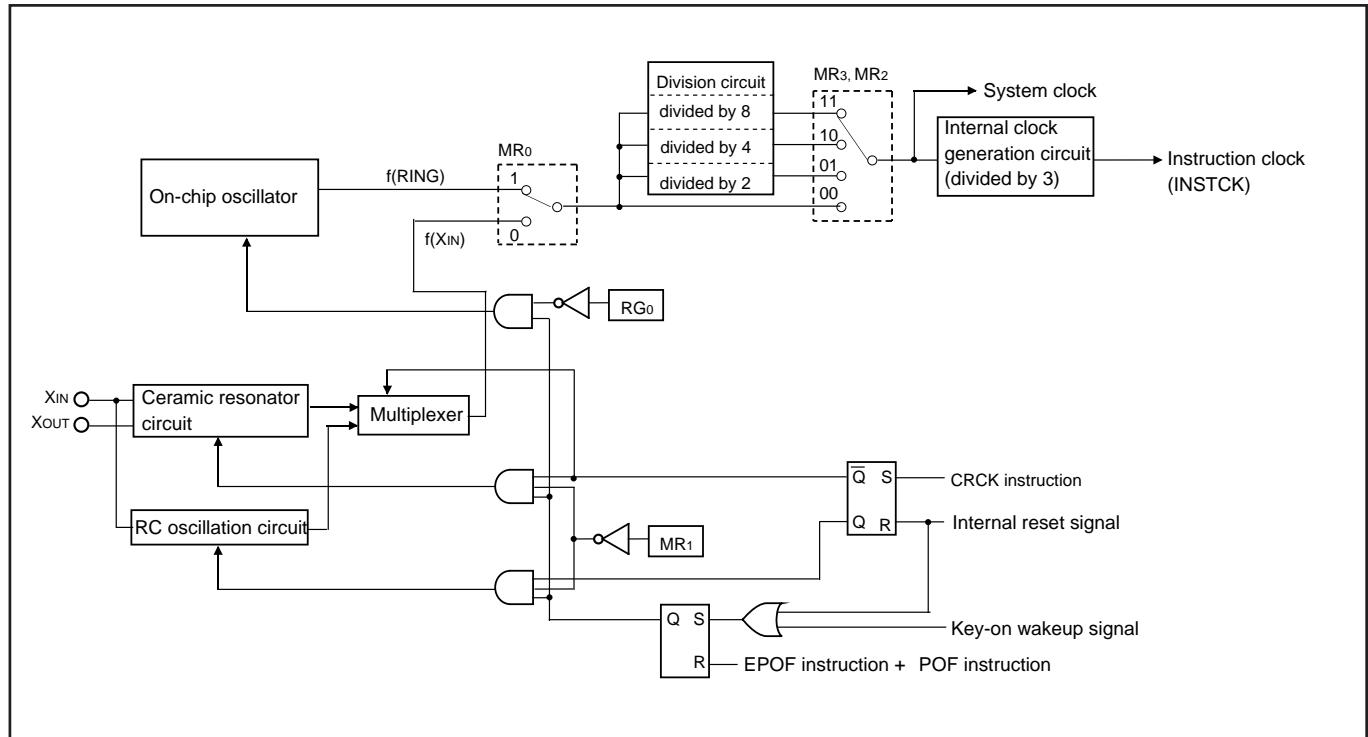


Fig. 49 Clock control circuit structure

## (1) On-chip oscillator operation

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

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

Be careful that variable frequencies when designing application products.

## (2) Main clock generating circuit (f(XIN))

The ceramic resonator or RC oscillation can be used for the main clock of this product.

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

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

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

The execution of the CRCK instruction can be valid only once.

Register MR controls the enable/disable of the oscillation and the selection of the operation source clock.

Also, when the MCU operates only by the on-chip oscillator without using main clock f(XIN), connect XIN pin to Vss and leave XOUT pin open, and do not execute the CRCK instruction (Figure 51).

## (3) Ceramic resonator

When the ceramic resonator is used as the main clock (f(XIN)), connect the ceramic resonator and the external circuit to pins XIN and XOUT at the shortest distance. A feedback resistor is built in between pins XIN and XOUT (Figure 52).

Do not execute the CRCK instruction.

Set "0" to bit 0 of register MR after the oscillation stabilizing wait time is generated by software to select the clock generated by the ceramic oscillation circuit for the source oscillation clock.

## (4) RC oscillation

When the RC oscillation is used as the main clock (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 53).

The frequency is affected by a capacitor, a resistor and a microcomputer. So, set the constants within the recommended operating condition of the frequency limits.

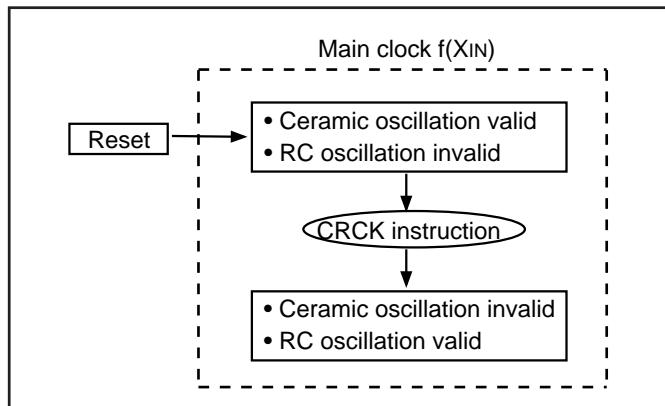


Fig. 50 Switch to ceramic oscillation/RC oscillation

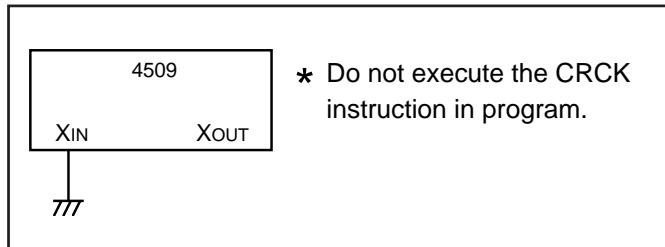


Fig. 51 Handling of XIN and XOUT when main clock is not used

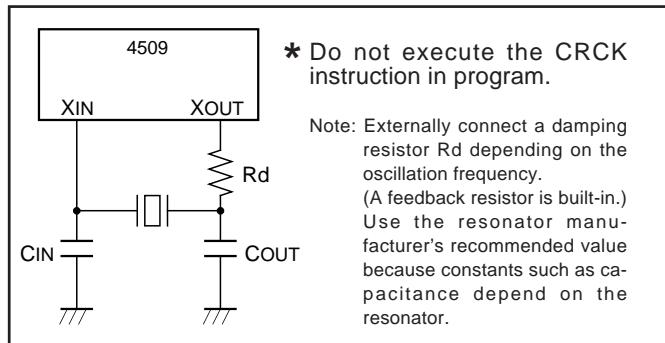


Fig. 52 Ceramic resonator external circuit

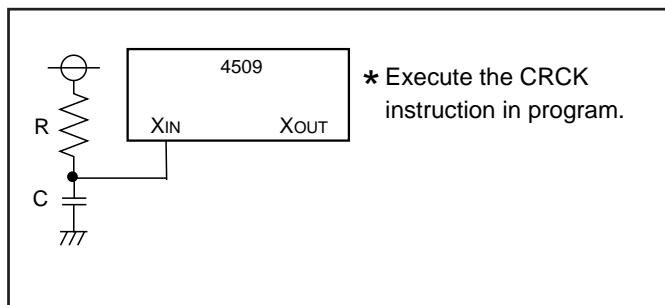


Fig. 53 External RC circuit

## (5) External clock

When the external signal clock is used for the main clock ( $f(XIN)$ ), connect the  $XIN$  pin to the clock source and leave  $XOUT$  pin open (Figure 54). Do not execute the CRCK instruction in program.

Be careful that the maximum value of the oscillation frequency when using the external clock differs from the value when using the ceramic resonator (refer to the recommended operating condition).

Also, note that the RAM back-up mode (POF instruction) cannot be used when using the external clock.

## (6) Clock control register MR

Register MR controls the selection of operation mode and the operation source clock, and enable/stop of main 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.

## (7) Clock control register RG

Register RG controls the on-chip oscillator. Set the contents of this register through register A with the TRGA instruction.

**Table 23 Clock control register MR**

Clock control register MR		at reset : 11012		at RAM back-up : 11012	R/W TAMR/TMRA
MR3	Operation mode selection bits	MR3	MR2	Operation mode	
		0	0	Through mode (frequency not divided)	
		0	1	Frequency divided by 2 mode	
		1	0	Frequency divided by 4 mode	
MR2		1	1	Frequency divided by 8 mode	
		0		Main clock ( $f(XIN)$ ) oscillation enabled	
MR1	Main clock $f(XIN)$ control bit (Notes 2, 5)	1		Main clock ( $f(XIN)$ ) oscillation stop	
		0		Main clock ( $f(XIN)$ )	
MR0	Operation source clock selection bit (Notes 3, 5)	1		On-chip oscillator clock ( $f(RING)$ )	

Clock control register RG		at reset : 02		at RAM back-up : 02	W TRGA
RG0	On-chip oscillator ( $f(RING)$ ) control bit (Note 4)	0		On-chip oscillator ( $f(RING)$ ) oscillation enabled	
		1		On-chip oscillator ( $f(RING)$ ) oscillation stop	

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

2: Main clock cannot be stopped when the main clock is selected for the operation source clock.

3: The stopped clock cannot be selected for the operation source clock. In order to switch the operation source clock, generate the oscillation stabilizing wait time by software first and set the oscillation of the destination clock to be enabled.

4: On-chip oscillator cannot be stopped when the on-chip oscillator is selected for the operation source clock.

5: When changing the setting of MR1 and MR0 from "00" to "11", make settings in the sequence "00" → "01" → "11".

When changing the setting of MR1 and MR0 from "11" to "00", make settings in the sequence "11" → "01" → "00".

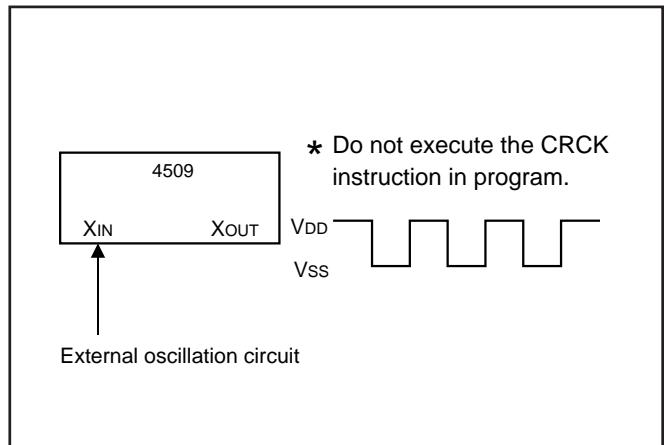


Fig. 54 External clock input circuit

## QzROM Writing Mode

In the QzROM writing mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a serial programmer which is applicable for this microcomputer.

Table 24 lists the pin description (QzROM writing mode) and Figure 55 shows the pin connections.

Refer to Figure 56 for examples of a connection with a serial programmer.

Contact the manufacturer of your serial programmer for serial programmer. Refer to the user's manual of your serial programmer for details on how to use it.

**Table 24 Pin description (QzROM writing mode)**

Pin	Name	I/O	Function
VDD	Power source	—	• Power supply voltage pin.
Vss	GND	—	• GND pin.
CNVss	VPP input	—	• QzROM programmable power source pin. • VPP input is possible with Vss connected via a resistor of about 5 kΩ.
P20/AIn0	SDA input/output	I/O	• QzROM serial data I/O pin.
P21/AIn1	SCLK input	Input	• QzROM serial clock input pin.
D3/AIn5	PGM input	Input	• QzROM read/program pulse input pin.
RESET	Reset input	Input	• Reset input pin. • Input "L" level signal.
XIN	Clock input	—	• Either connect an oscillation circuit or connect XIN pin to Vss and leave the XOUT pin open.
XOUT	Clock output	—	
D0, D1, D2/AIn4, D4, D5, P00/SIN, P01/SOUT, P02/Sck, P03, P10, P11/CNTR1, P12/CNTR0, P13/INT, P30/AIn2, P31/AIn3	I/O port	I/O	• Input "H" or "L" level signal or leave the pin open.

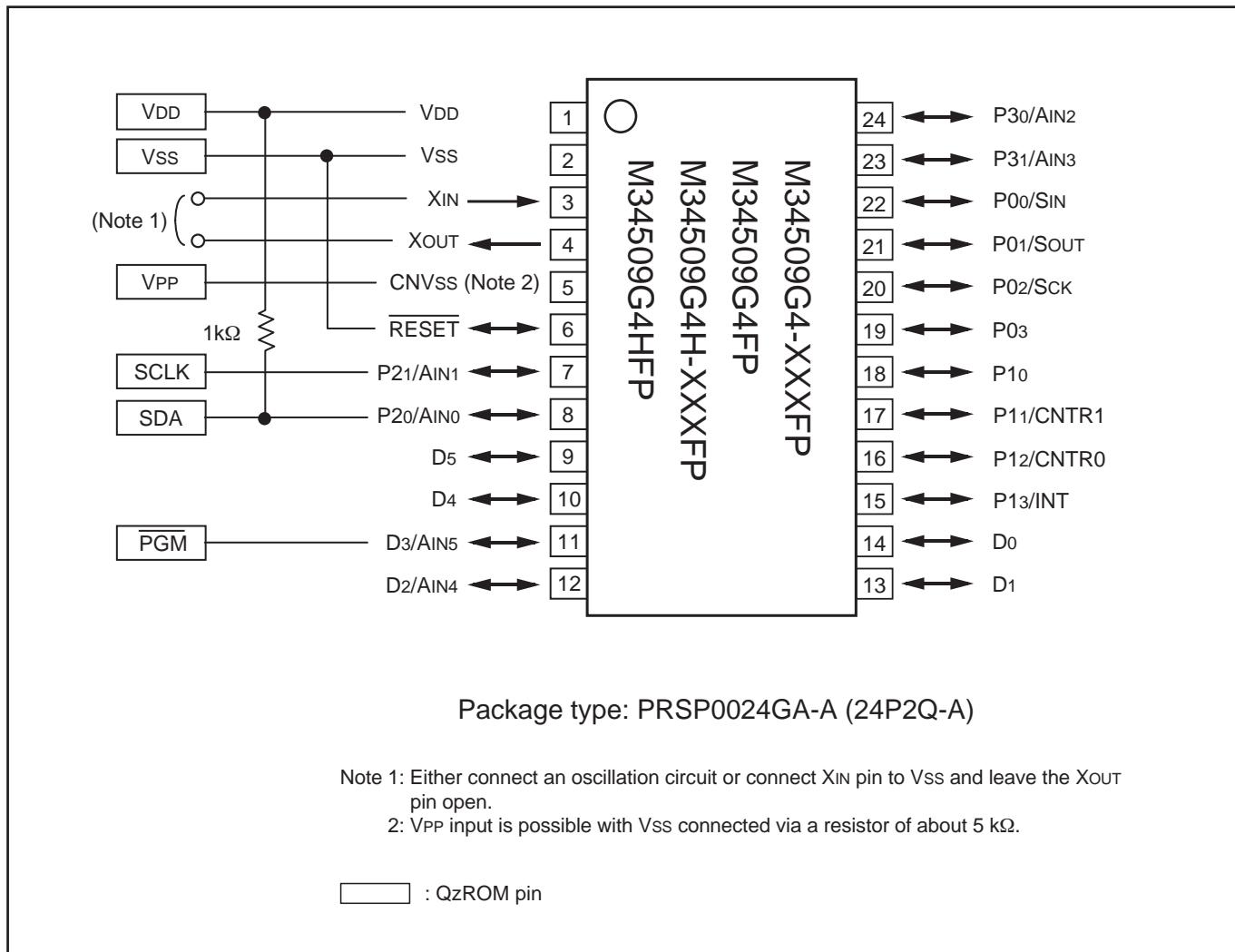


Fig. 55 Pin connection diagram

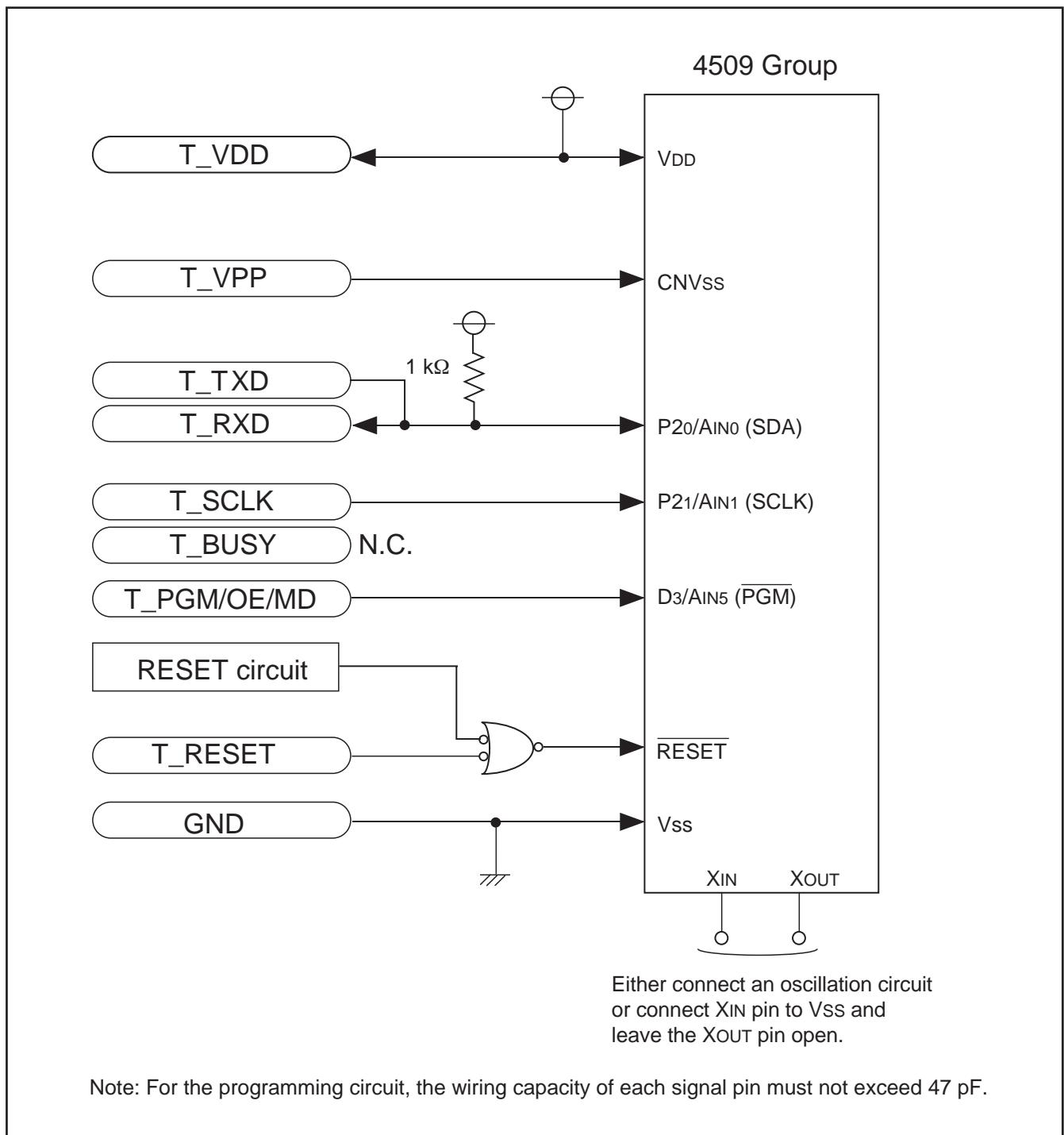


Fig. 56 When using programmer of Suisei Electronics System Co., LTD, connection example

## DATA REQUIRED FOR QzROM WRITING ORDERS

The following are necessary when ordering a QzROM product shipped after writing:

1. QzROM Writing Confirmation Form\*
2. Mark Specification Form\*
3. ROM data.....Mask file

\* For the QzROM writing confirmation form and the mark specification form, refer to the "Renesas Technology Corp." Homepage (<http://www.renesas.com/homepage.jsp>).

Note that we cannot deal with special font marking (customer's trademark etc.) in QzROM microcomputer.

## LIST OF PRECAUTIONS

### ① Noise and latch-up prevention

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

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

CNVss pin is also used as VPP pin. Accordingly, when using this pin, connect this pin to Vss through a resistor about 5 k $\Omega$  (connect this resistor to CNVss/VPP pin as close as possible).

### ② Note on Power Source Voltage

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

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

### ③ 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)

### ④ 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)

### ⑤ Program counter

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

### ⑥ 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.

### ⑦ Multifunction

- The input/output of P00 can be used even when SIN is used. Be careful when using inputs of both SIN and P00 since the input threshold value of SIN pin is different from that of port P00.
- The input of P01 can be used even when SOUT is used.
- The input of P02 can be used even when SCK is used. Be careful when using inputs of both SCK and P02 since the input threshold value of SCK pin is different from that of port P02.
- The input of P11 can be used even when CNTR1 (output) is selected. The input/output of P11 can be used even when CNTR1 (input) is selected. Be careful when using inputs of both CNTR1 and P11 since the input threshold value of CNTR1 pin is different from that of port P11.
- The input of P12 can be used even when CNTR0 (output) is selected. The input/output of P12 can be used even when CNTR0 (input) is selected. Be careful when using inputs of both CNTR0 and P12 since the input threshold value of CNTR0 pin is different from that of port P12.
- The input/output of P13 can be used even when INT is used. Be careful when using inputs of both INT and P13 since the input threshold value of INT pin is different from that of port P13.
- The input/output of P20, P21, P30, P31, D2, D3 can be used even when AIN0–AIN5 are used.

### ⑧ Power-on reset (only for H version)

When the built-in power-on reset circuit is used, set the time for the supply voltage to rise from 0 V to the minimum voltage of recommended operating conditions to 100  $\mu$ s or less. If the rising time exceeds 100  $\mu$ s, connect a capacitor between the RESET pin and Vss at the shortest distance, and input "L" level to RESET pin until the value of supply voltage reaches the minimum operating voltage.

### ⑨ POF instruction

When the POF 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 POF instruction.

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

## ⑩ 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 57①) and then, change the bit 3 of register I1.

In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 57②).

Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 57③).

```

    :
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. 57 External 0 interrupt program example-1

## ② Note [2] on bit 3 of register I1

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

- When the INT pin input is disabled (register I13 = "0"), set the key-on wakeup of INT pin to be invalid (register L10 = "0") before system enters to the RAM back-up mode. (refer to Figure 58①).

```

    :
LA  0      ; (XXX02)
TI1A      ; INT key-on wakeup disabled .....①
DI
EPOF
POF2      ; RAM back-up
    :
X : these bits are not used here.

```

Fig. 58 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 59①) and then, change the bit 2 of register I1.

In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 59②).

Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 59③).

```

    :
LA  4      ; (XXX02)
TV1A      ; The SNZ0 instruction is valid .....①
LA 12      ; (1XXX2)
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. 59 A/D conversion interrupt program example

## ⑩ Prescaler

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

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

## ⑪ Timer count source

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

## ⑫ Reading the count value

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

## ⑬ Writing to the timer

Stop timer 1 or 2 counting and then execute the T1AB, T1R1L, T2AB or T2R2L instruction to write data to timer.

## ⑭ Writing to reload register

In order to write a data to the reload register R1H while the timer 1 is operating, execute the T1HAB instruction except a timing of the timer 1 underflow.

In order to write a data to the reload register R2H while the timer 2 is operating, execute the T2HAB instruction except a timing of the timer 2 underflow.

## ⑮ Prescaler, timer 1 and timer 2 count start timing and count time when operation starts

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

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

When selecting CNTR input as the count source of timer, timer operates synchronizing with the count edge (falling edge or rising edge) of CNTR input selected by software.

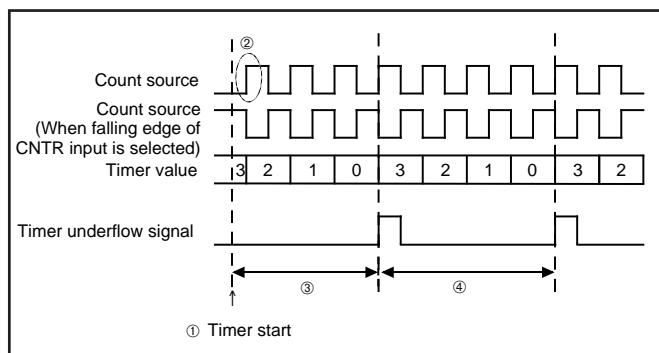


Fig. 60 Timer count start timing and count time when operation starts

## ⑯ PWM signal (PWM1, PWM2)

If the timer 1 count stop timing and the timer 1 underflow timing overlap during output of the PWM1 signal, a hazard may occur in the PWM1 output waveform.

If the timer 2 count stop timing and the timer 2 underflow timing overlap during output of the PWM2 signal, a hazard may occur in the PWM2 output waveform.

## ⑰ 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 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.

Also, set the NOP instruction after the WRST instruction, for the case when a skip is performed with the WRST instruction.

## ⑱ Clock control

When the RC oscillation is used as the main clock f(XIN), execute the CRCK instruction in the initial setting routine of program (executing it in address 0 in page 0 is recommended).

The oscillation circuit by the CRCK instruction can be selected only once. When the CRCK instruction is not executed, the ceramic oscillation is selected for the main clock f(XIN).

Also, when the MCU operates only by the on-chip oscillator without using main clock f(XIN), connect XIN pin to Vss and leave XOUT pin open, and do not execute the CRCK instruction.

In order to switch the operation source clock (f(RING)) or f(XIN)), generate the oscillation stabilizing wait time by software first and set the oscillation of the destination clock to be enabled.

Registers RG and MR are initialized when system returns from RAM back-up mode.

However, the selected contents (CRCK instruction execution state) of main clock (f(XIN)) oscillation circuit is retained.

## ⑲ On-chip oscillator

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

Be careful that variable frequencies when designing application products.

Also, when considering the oscillation stabilize wait time for switching clock, be careful that the variable frequency of the on-chip oscillator clock.

## ⑳ External clock

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

Be careful that the maximum value of the oscillation frequency when using the external clock differs from the value when using the ceramic resonator (refer to the recommended operating condition).

Also, note that the RAM back-up mode (POF instruction) cannot be used when using the external clock.

## ② Notes for the use of A/D conversion 1

## • 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."

- 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.
- Clear the bit 2 of register V2 to "0" to change the operating mode from the comparator mode to A/D conversion mode.
- 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      ; (X0XX2)
TV2A      ; The SNZAD instruction is valid .....①
LA 0      ; (0XXX2)
TQ1A      ; Operation mode of A/D converter is
          ; changed from comparator mode to A/D
          ; conversion mode.

```

SNZAD  
NOP

⋮  
X : this bit is not related to change the operation mode of A/D converter.

Fig. 61 External 0 interrupt program example-3

## ③ Notes for the use of A/D conversion 2

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  $\mu$ F to 1  $\mu$ F) to analog input pins (Figure 60).

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 61. In addition, test the application products sufficiently.

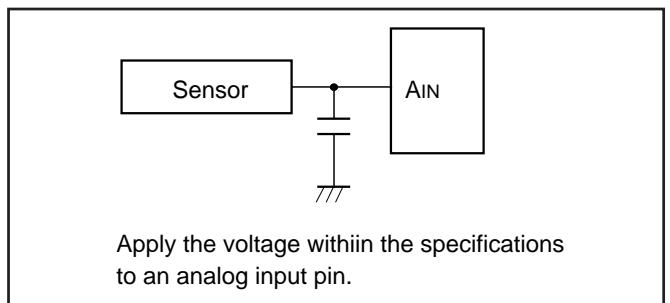


Fig. 62 Analog input external circuit example-1

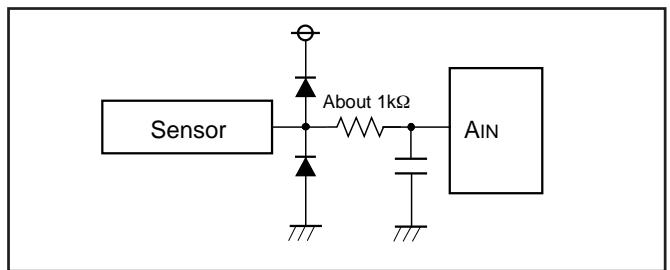


Fig. 63 Analog input external circuit example-2

## ④ QzROM

(1) Be careful not to apply overvoltage to MCU. The contents of QzROM may be overwritten because of overvoltage. Take care especially at turning on the power.

(2) As for the product shipped in blank, Renesas does not perform the writing test to user ROM area after the assembly process though the QzROM writing test is performed enough before the assembly process. Therefore, a writing error of approx.0.1 % may occur. Moreover, please note the contact of cables and foreign bodies on a socket, etc. because a writing environment may cause some writing errors.

⑤ Notes On ROM Code Protect  
(QzROM product shipped after writing)

As for the QzROM product shipped after writing, the ROM code protect is specified according to the ROM option setup data in the mask file which is submitted at ordering.

The ROM option setup data in the mask file is "0016" for protect enabled or "FF16" for protect disabled.

Note that the mask file which has nothing at the ROM option data or has the data other than "0016" and "FF16" can not be accepted.

## NOTES ON NOISE

Countermeasures against noise are described below.

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

### 1. Shortest wiring length

#### (1) Wiring for RESET pin

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

#### <Reason>

In order to reset a microcomputer correctly, 1 machine cycle or more of the width of a pulse input into the RESET pin is required.

If noise having a shorter pulse width than this is input to the RESET input pin, the reset is released before the internal state of the microcomputer is completely initialized.

This may cause a program runaway.

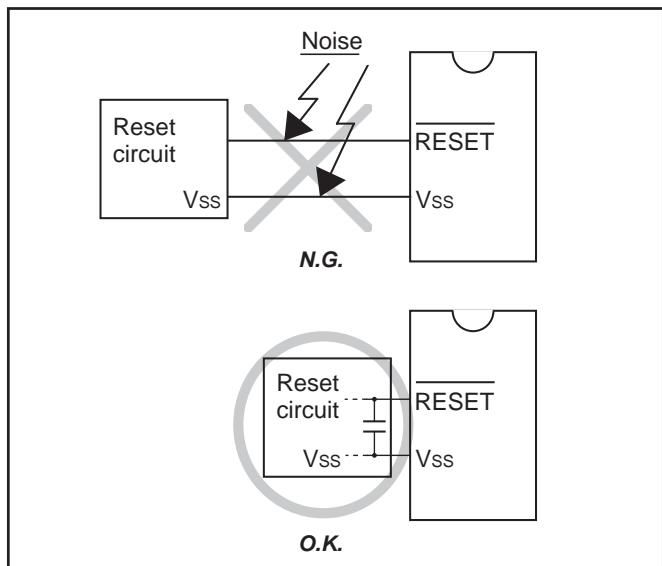


Fig. 64 Wiring for the RESET pin

#### (2) Wiring for clock input/output pins

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

#### <Reason>

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

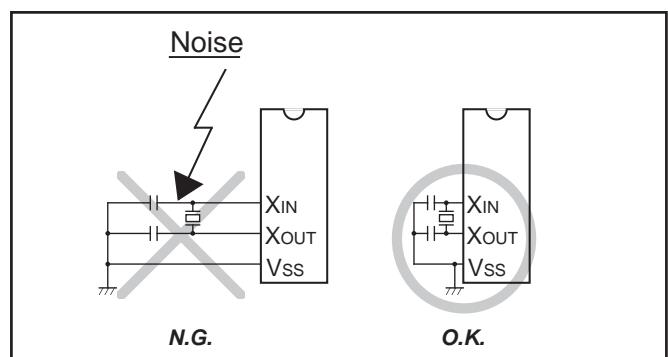


Fig. 65 Wiring for clock I/O pins

#### (3) Wiring to CNVss pin

Connect CNVss pin to a GND pattern at the shortest distance. The GND pattern is required to be as close as possible to the GND supplied to Vss.

In order to improve the noise reduction, to connect a  $5\text{ k}\Omega$  resistor serially to the CNVss pin - GND line may be valid.

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

#### <Reason>

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

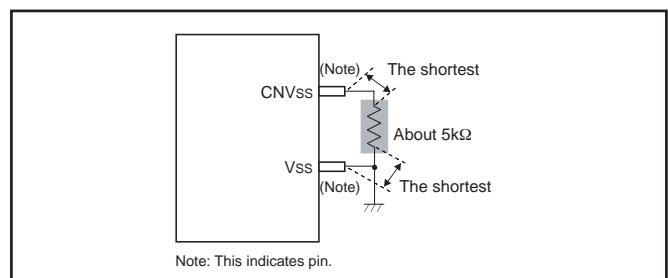


Fig. 66 Wiring for the CNVss pin of the QzPROM

## 2. Connection of bypass capacitor across Vss line and VDD line

Connect an approximately  $0.1 \mu\text{F}$  bypass capacitor across the Vss line and the VDD line as follows:

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

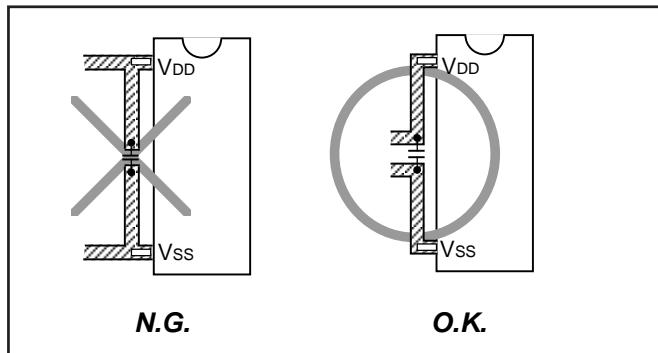


Fig. 67 Bypass capacitor across the Vss line and the VDD line

## 3. Wiring to analog input pins

- Connect an approximately  $100 \Omega$  to  $1 \text{k}\Omega$  resistor to an analog signal line which is connected to an analog input pin in series. Besides, connect the resistor to the microcomputer as close as possible.
- Connect an approximately  $1000 \text{ pF}$  capacitor across the Vss pin and the analog input pin. Besides, connect the capacitor to the Vss pin as close as possible. Also, connect the capacitor across the analog input pin and the Vss pin at equal length.

### <Reason>

Signals which is input in an analog input pin (such as an A/D converter/comparator input pin) are usually output signals from sensor. The sensor which detects a change of event is installed far from the printed circuit board with a microcomputer, the wiring to an analog input pin is longer necessarily. This long wiring functions as an antenna which feeds noise into the microcomputer, which causes noise to an analog input pin.

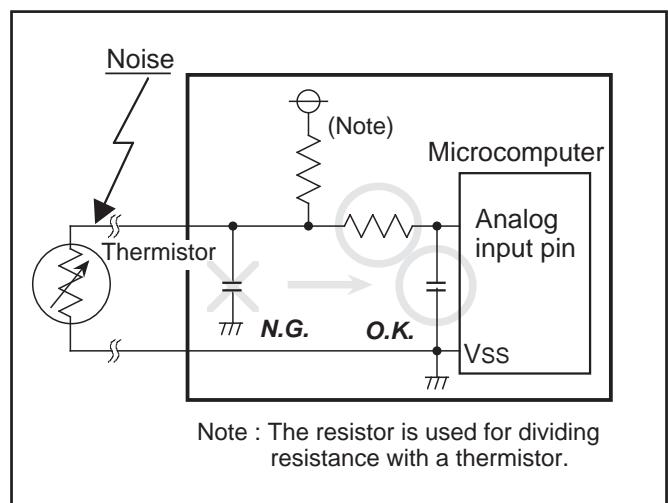


Fig. 68 Analog signal line and a resistor and a capacitor

#### 4. Oscillator concerns

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

##### (1) Keeping oscillator away from large current signal lines

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

##### <Reason>

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

##### (2) Installing oscillator away from signal lines where potential levels change frequently

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

##### <Reason>

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

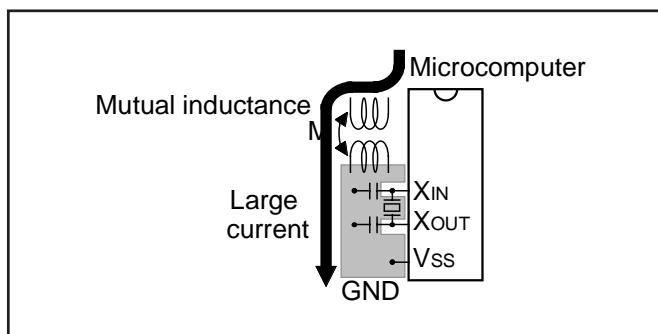


Fig. 69 Wiring for a large current signal line

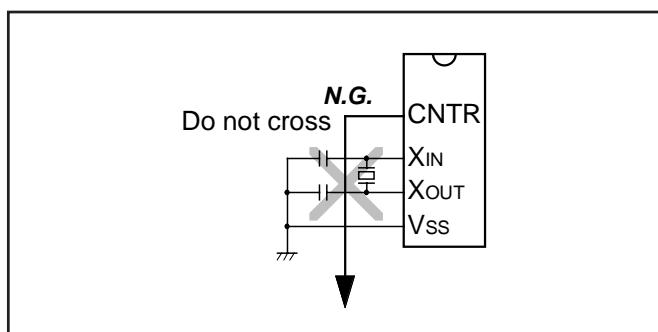


Fig. 70 Wiring to a signal line where potential levels change frequently

##### (3) Oscillator protection using Vss pattern

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

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

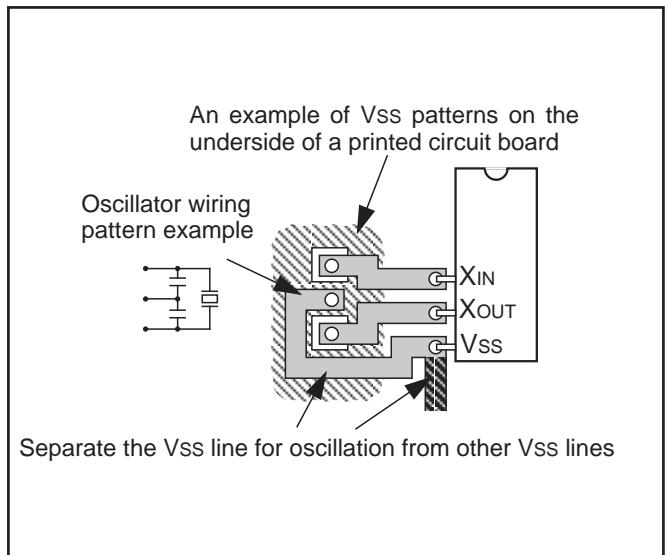


Fig. 71 Vss pattern on the underside of an oscillator

#### 5. Setup for I/O ports

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

##### <Hardware>

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

##### <Software>

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

#### 6. Providing of watchdog timer function by software

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

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

&lt;The main routine&gt;

- Assigns a single word of RAM to a software watchdog timer (SWDT) and writes the initial value N in the SWDT once at each execution of the main routine. The initial value N should satisfy the following condition:  
 $N+1 \geq (\text{Counts of interrupt processing executed in each main routine})$
- As the main routine execution cycle may change because of an interrupt processing or others, the initial value N should have a margin.
- Watches the operation of the interrupt processing routine by comparing the SWDT contents with counts of interrupt processing after the initial value N has been set.
- Detects that the interrupt processing routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:  
If the SWDT contents do not change after interrupt processing.

&lt;The interrupt processing routine&gt;

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

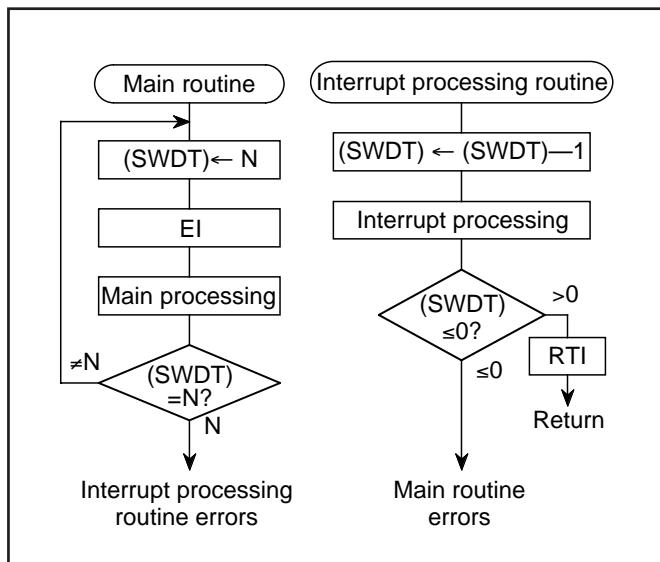


Fig. 72 Watchdog timer by software

## CONTROL REGISTERS

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

Interrupt control register V2		at reset : 00002		at RAM back-up : 00002	R/W TAV2/TV2A
V23	Serial interface interrupt enable bit	0	Interrupt disabled (SNZSI instruction is valid)		
		1	Interrupt enabled (SNZSI instruction is invalid)		
V22	A/D interrupt enable bit	0	Interrupt disabled (SNZAD instruction is valid)		
		1	Interrupt enabled (SNZAD instruction is invalid)		
V21	Not used	0	This bit has no function, but read/write is enabled.		
		1			
V20	Not used	0	This bit has no function, but read/write is enabled.		
		1			

Interrupt control register I1		at reset : 00002		at RAM back-up : state retained	R/W TAI1/TI1A
I13	INT pin input control bit (Note 2)	0	INT pin input disabled		
		1	INT pin input enabled		
I12	Interrupt valid waveform for INT pin/return level selection bit (Note 2)	0	Falling waveform ("L" level 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		
I11	INT pin edge detection circuit control bit	0	One-sided edge detected		
		1	Both edges detected		
I10	INT pin timer 1 control enable bit	0	Disabled		
		1	Enabled		

Clock control register MR		at reset : 11012		at RAM back-up : 11012	R/W TAMR/TMRA	
MR3	Operation mode selection bits	MR3	MR2	Operation mode		
		0	0	Through mode (frequency not divided)		
MR2		0	1	Frequency divided by 2 mode		
		1	0	Frequency divided by 4 mode		
MR1	Main clock f(XIN) control bit (Note 3)	1	1	Frequency divided by 8 mode		
		0	0	Main clock (f(XIN)) oscillation enabled		
MR0	Operation source clock selection bit (Note 4)	1	1	Main clock (f(XIN)) oscillation stop		
		0	0	On-chip oscillator clock (f(RING))		

Clock control register RG		at reset : 02		at RAM back-up : 02	W TRGA
RG0	On-chip oscillator (f(RING)) control bit (Note 5)	0	On-chip oscillator (f(RING)) oscillation enabled		
		1	On-chip oscillator (f(RING)) oscillation stop		

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

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

3: Main clock cannot be stopped when the main clock is selected for the operation source clock.

4: The stopped clock cannot be selected for the operation source clock. In order to switch the operation source clock, generate the oscillation stabilizing wait time by software first and set the oscillation of the destination clock to be enabled.

5: On-chip oscillator cannot be stopped when the on-chip oscillator is selected for the operation source clock.

Timer control register PA		at reset : 02		at RAM back-up : 02	W TPAA
PA0	Prescaler control bit	0	Stop (state initialized)		
		1	Operating		

Timer control register W1		at reset : 00002		at RAM back-up : 00002	R/W TAW1/TW1A
W13	PWM1 function control bit	0	PWM1 function invalid		
		1	PWM1 function valid		
W12	Timer 1 control bit	0	Stop (state retained)		
		1	Operating		
W11	Timer 1 count source selection bits	W11	W10	Count source	
		0	0	PWM2 signal	
		0	1	Prescaler output (ORCLK)	
		1	0	CNTR1 input	
		1	1	On-chip oscillator clock (f(RING))	

Timer control register W2		at reset : 00002		at RAM back-up : 00002	R/W TAW2/TW2A
W23	PWM2 function control bit	0	PWM2 function invalid		
		1	PWM2 function valid		
W22	Timer 2 control bit	0	Stop (state retained)		
		1	Operating		
W21	Timer 2 count source selection bits	W21	W20	Count source	
		0	0	Timer 1 underflow signal (T1UDF)	
		0	1	Prescaler output (ORCLK)	
		1	0	CNTR0 input	
		1	1	System clock (STCK)	

Timer control register W5		at reset : 00002		at RAM back-up : state retained	R/W TAW5/TW5A
W53	P12/CNTR0 pin function selection bit	0	P12 (I/O) / CNTR0 (input)		
		1	P12 (input) /CNTR0 (I/O)		
W52	Timer 1 count auto-stop circuit selection bit (Note 2)	0	Count auto-stop circuit not selected		
		1	Count auto-stop circuit selected		
W51	Timer 1 count start synchronous circuit selection bit (Note 3)	0	Count start synchronous circuit not selected		
		1	Count start synchronous circuit selected		
W50	CNTR0 pin input count edge selection bit	0	Falling edge		
		1	Rising edge		

Timer control register W6		at reset : 00002		at RAM back-up : state retained	R/W TAW6/TW6A
W63	P11/CNTR1 pin function selection bit	0	P11 (I/O) / CNTR1 (input)		
		1	P11 (input) /CNTR1 (I/O)		
W62	CNTR 1 pin output auto-control circuit selection bit	0	Output auto-control circuit not selected		
		1	Output auto-control circuit selected		
W61	Timer 2 INT pin input period count circuit selection bit	0	INT pin input period count circuit not selected		
		1	INT pin input period count circuit selected		
W60	CNTR1 pin input count edge selection bit	0	Falling edge		
		1	Rising edge		

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

2: This function is valid only when the INT pin/timer 1 control is enabled (I10="1") and the timer 1 count start synchronous circuit is selected (W51="1").

3: This function is valid only when the INT pin/timer 1 control is enabled (I10="1").

A/D control register Q1			at reset : 00002			at RAM back-up : state retained		R/W TAQ1/TQ1A		
Q13	A/D operation mode selection bit		0	A/D conversion mode						
			1	Comparator mode						
Q12	Analog input pin selection bits		Q12	Q11	Q10	Selected pins				
			0	0	0	AIN0				
			0	0	1	AIN1				
			0	1	0	AIN2				
			0	1	1	AIN3				
			1	0	0	AIN4				
			1	0	1	AIN5				
			1	1	0	Not available				
			1	1	1	Not available				

Serial interface control register J1			at reset : 00002			at RAM back-up : state retained		R/W TAJ1/TJ1A				
J13	Serial interface synchronous clock selection bits		J13	J12	Synchronous clock							
			0	0	Instruction clock (INSTCK) divided by 8							
J12			0	1	Instruction clock (INSTCK) divided by 4							
			1	0	Instruction clock (INSTCK) divided by 2							
J11	Serial interface port function selection bits		J13	J12	Port function							
			0	0	P00, P01, P02 selected/SIN, SOUT, Sck not selected							
			0	1	P00, SOUT, Sck selected/SIN, P01, P02 not selected							
			1	0	SIN, P01, Sck selected/P00, SOUT, P02 not selected							
			1	1	SIN, SOUT, Sck selected/P00, P01, P02 not selected							

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

Key-on wakeup control register K0		at reset : 00002		at RAM back-up : state retained	R/W TAK0/TK0A
K03	Port P03 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K02	Port P02 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K01	Port P01 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K00	Port P00 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K1		at reset : 00002		at RAM back-up : state retained	R/W TAK1/TK1A
K13	Port P13 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K12	Port P12 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K11	Port P11 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K10	Port P10 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K2		at reset : 00002		at RAM back-up : state retained	R/W TAK2/TK2A
K23	Port D3 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K22	Port D2 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K21	Port P21 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
K20	Port P20 key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register L1		at reset : 00002		at RAM back-up : state retained	R/W TAL1/TL1A
L13	Ports P10–P13 return condition selection bit	0	Return by level		
		1	Return by edge		
L12	Ports P10–P13 valid waveform/level selection bit	0	Falling waveform/"L" level		
		1	Rising waveform/"H" level		
L11	INT pin return condition selection bit	0	Return by level		
		1	Return by edge		
L10	INT pin key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

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

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

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

Pull-up control register PU2		at reset : 00002		at RAM back-up : state retained	R/W TAPU2/TPU2A
PU23	Port D3 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU22	Port D2 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU21	Port P21 pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
PU20	Port P20 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.

Port output structure control register FR0		at reset : 00002		at RAM back-up : state retained	W TFR0A
FR03	Port P03 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
FR02	Port P02 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
FR01	Port P01 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
FR00	Port P00 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		

Port output structure control register FR1		at reset : 00002		at RAM back-up : state retained	W TFR1A
FR13	Port P13 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
FR12	Port P12 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
FR11	Port P11 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
FR10	Port P10 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		

Port output structure control register FR2		at reset : 00002		at RAM back-up : state retained	W TFR2A
FR23	Not used	0			
		1	This bit has no function, but read/write is enabled.		
FR22	Not used	0			
		1	This bit has no function, but read/write is enabled.		
FR21	Port P21 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
FR20	Port P20 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		

Port output structure control register FR3		at reset : 00002		at RAM back-up : state retained	W TFR3A
FR33	Port D3 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
FR32	Port D2 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
FR31	Port D1 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
FR30	Port D0 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		

Port output structure control register C1		at reset : 00002		at power down : state retained	W TC1A
C13	Port D5 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
C12	Port D4 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
C11	Port P31 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
C10	Port P30 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		

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

**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)	RPS	Prescaler reload register (8 bits)
B	Register B (4 bits)	R1L	Timer 1 reload register (8 bits)
DR	Register D (3 bits)	R1H	Timer 1 reload register (8 bits)
E	Register E (8 bits)	R2L	Timer 2 reload register (8 bits)
Q1	A/D control register Q1 (4 bits)	R2H	Timer 2 reload register (8 bits)
V1	Interrupt control register V1 (4 bits)	PS	Prescaler
V2	Interrupt control register V2 (4 bits)	T1	Timer 1
I1	Interrupt control register I1 (4 bits)	T2	Timer 2
W1	Timer control register W1 (4 bits)	T1F	Timer 1 interrupt request flag
W2	Timer control register W2 (4 bits)	T2F	Timer 2 interrupt request flag
W5	Timer control register W5 (4 bits)	WDF1	Watchdog timer flag
W6	Timer control register W6 (4 bits)	WEF	Watchdog timer enable flag
FR0	Port output structure control register FR0 (4 bits)	INTE	Interrupt enable flag
FR1	Port output structure control register FR1 (4 bits)	EXF0	External 0 interrupt request flag
FR2	Port output structure control register FR2 (4 bits)	P	Power down flag
FR3	Port output structure control register FR3 (4 bits)	ADF	A/D conversion completion flag
C1	Port output structure control register C1 (4 bits)	SIOF	Serial interface transmit/receive completion flag
J1	Serial interface control register J1 (4 bits)	D	Port D (6 bits)
MR	Clock control register MR (4 bits)	P0	Port P0 (4 bits)
K0	Key-on wakeup control register K0 (4 bits)	P1	Port P1 (4 bits)
K1	Key-on wakeup control register K1 (4 bits)	P2	Port P2 (2 bits)
K2	Key-on wakeup control register K2 (4 bits)	P3	Port P3 (2 bits)
L1	Key-on wakeup control register L1 (4 bits)	x	Hexadecimal variable
PU0	Pull-up control register PU0 (4 bits)	y	Hexadecimal variable
PU1	Pull-up control register PU1 (4 bits)	z	Hexadecimal variable
PU2	Pull-up control register PU2 (4 bits)	p	Hexadecimal variable
X	Register X (4 bits)	n	Hexadecimal constant
Y	Register Y (4 bits)	i	Hexadecimal constant
Z	Register Z (2 bits)	j	Hexadecimal constant
DP	Data pointer (10 bits) (It consists of registers X, Y, and Z)	A3A2A1A0	Binary notation of hexadecimal variable A (same for others)
PC	Program counter (14 bits)	←	Direction of data movement
PCH	High-order 7 bits of program counter	↔	Data exchange between a register and memory
PCL	Low-order 7 bits of program counter	?	Decision of state shown before “?”
SK	Stack register (14 bits X 8)	( )	Contents of registers and memories
SP	Stack pointer (3 bits)	—	Negate, Flag unchanged after executing instruction
CY	Carry flag	M(DP)	RAM address pointed by the data pointer
		a	Label indicating address a5 a4 a3 a2 a1 a0
		p, a	Label indicating address a6 a5 a4 a3 a2 a1 a0 in page p6 p5 p4 p3 p2 p1 p0
		C	Hex. C + Hex. number x (also same for others)
		+	
		x	

Note : The 4509 Group just invalidates the next instruction when a skip is performed. The contents of program counter is not increased by 2. Accordingly, the number of cycles does not change even if skip is not performed. However, the cycle count becomes “1” if the TABP p, RT, or RTS instruction is skipped.

## INDEX LIST OF INSTRUCTION FUNCTION

Grouping	Mnemonic	Function	Grouping	Mnemonic	Function
Register to register transfer	TAB	(A) $\leftarrow$ (B)	RAM to register transfer	XAMI j	(A) $\leftarrow$ $\rightarrow$ (M(DP)) (X) $\leftarrow$ (X)EXOR(j) j = 0 to 15 (Y) $\leftarrow$ (Y) + 1
	TBA	(B) $\leftarrow$ (A)		TMA j	(M(DP)) $\leftarrow$ (A) (X) $\leftarrow$ (X)EXOR(j) j = 0 to 15
	TAY	(A) $\leftarrow$ (Y)			
	TYA	(Y) $\leftarrow$ (A)			
	TEAB	(E7–E4) $\leftarrow$ (B) (E3–E0) $\leftarrow$ (A)		LA n	(A) $\leftarrow$ n n = 0 to 15
	TABE	(B) $\leftarrow$ (E7–E4) (A) $\leftarrow$ (E3–E0)		TABP p	(SP) $\leftarrow$ (SP) + 1 (SK(SP)) $\leftarrow$ (PC) (PCh) $\leftarrow$ p (Note) (PCL) $\leftarrow$ (DR2–DR0, A3–A0) (UPTF) = 1, (DR1, DR0) $\leftarrow$ (ROM(PC)) <sup>9, 8</sup> (DR2) $\leftarrow$ 0
	TDA	(DR <sub>2</sub> –DR <sub>0</sub> ) $\leftarrow$ (A <sub>2</sub> –A <sub>0</sub> )			(B) $\leftarrow$ (ROM(PC)) <sup>7–4</sup> (A) $\leftarrow$ (ROM(PC)) <sup>3–0</sup> (PC) $\leftarrow$ (SK(SP)) (SP) $\leftarrow$ (SP) – 1
	TAD	(A <sub>2</sub> –A <sub>0</sub> ) $\leftarrow$ (DR <sub>2</sub> –DR <sub>0</sub> ) (A <sub>3</sub> ) $\leftarrow$ 0			
	TAZ	(A <sub>1</sub> , A <sub>0</sub> ) $\leftarrow$ (Z <sub>1</sub> , Z <sub>0</sub> ) (A <sub>3</sub> , A <sub>2</sub> ) $\leftarrow$ 0			
	TAX	(A) $\leftarrow$ (X)		AM	(A) $\leftarrow$ (A) + (M(DP))
RAM addresses	TASP	(A <sub>2</sub> –A <sub>0</sub> ) $\leftarrow$ (SP <sub>2</sub> –SP <sub>0</sub> ) (A <sub>3</sub> ) $\leftarrow$ 0	Arithmetic operation	AMC	(A) $\leftarrow$ (A) + (M(DP)) + (CY) (CY) $\leftarrow$ Carry
	LXY x, y	(X) $\leftarrow$ x x = 0 to 15 (Y) $\leftarrow$ y y = 0 to 15		A n	(A) $\leftarrow$ (A) + n n = 0 to 15
	LZ z	(Z) $\leftarrow$ z z = 0 to 3		AND	(A) $\leftarrow$ (A) AND (M(DP))
	INY	(Y) $\leftarrow$ (Y) + 1		OR	(A) $\leftarrow$ (A) OR (M(DP))
RAM to register transfer	DEY	(Y) $\leftarrow$ (Y) – 1		SC	(CY) $\leftarrow$ 1
	TAM j	(A) $\leftarrow$ (M(DP)) (X) $\leftarrow$ (X)EXOR(j) j = 0 to 15		RC	(CY) $\leftarrow$ 0
	XAM j	(A) $\leftarrow$ $\rightarrow$ (M(DP)) (X) $\leftarrow$ (X)EXOR(j) j = 0 to 15		SZC	(CY) = 0 ?
	XAMD j	(A) $\leftarrow$ $\rightarrow$ (M(DP)) (X) $\leftarrow$ (X)EXOR(j) j = 0 to 15 (Y) $\leftarrow$ (Y) – 1		CMA	(A) $\leftarrow$ $\overline{(A)}$
				RAR	$\rightarrow$ CY $\rightarrow$ A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>

Note: p is 0 to 31.

## INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function	Grouping	Mnemonic	Function
Bit operation	SB j	$(Mj(DP)) \leftarrow 1$ $j = 0 \text{ to } 3$	Interrupt operation	DI	$(INTE) \leftarrow 0$
	RB j	$(Mj(DP)) \leftarrow 0$ $j = 0 \text{ to } 3$		EI	$(INTE) \leftarrow 1$
	SZB j	$(Mj(DP)) = 0 ?$ $j = 0 \text{ to } 3$		SNZ0	$V10 = 0: (EXF0) = 1 ?$ $(EXF0) \leftarrow 0$ $V10 = 1: SNZ0 = NOP$
Comparison operation	SEAM	$(A) = (M(DP)) ?$		SNZI0	$I12 = 0 : (INT) = "L" ?$ $I12 = 1 : (INT) = "H" ?$
	SEA n	$(A) = n ?$ $n = 0 \text{ to } 15$		TAV1	$(A) \leftarrow (V1)$
Branch operation	B a	$(PCL) \leftarrow a6-a0$		TV1A	$(V1) \leftarrow (A)$
	BL p, a	$(PCH) \leftarrow p \text{ (Note)}$ $(PCL) \leftarrow a6-a0$		TAV2	$(A) \leftarrow (V2)$
	BLA p	$(PCH) \leftarrow p \text{ (Note)}$ $(PCL) \leftarrow (DR2-DR0, A3-A0)$		TV2A	$(V2) \leftarrow (A)$
Subroutine operation	BM a	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow 2$ $(PCL) \leftarrow a6-a0$	Timer operation	TAI1	$(A) \leftarrow (I1)$
	BML p, a	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p \text{ (Note)}$ $(PCL) \leftarrow a6-a0$		TI1A	$(I1) \leftarrow (A)$
	BMLA p	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p \text{ (Note)}$ $(PCL) \leftarrow (DR2-DR0, A3-A0)$		TPAA	$(PA) \leftarrow (A)$
Return operation	RTI	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$		TAW1	$(A) \leftarrow (W1)$
	RT	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$		TW1A	$(W1) \leftarrow (A)$
	RTS	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$		TAW2	$(A) \leftarrow (W2)$
				TW2A	$(W2) \leftarrow (A)$
				TAW5	$(A) \leftarrow (W5)$
				TW5A	$(W5) \leftarrow (A)$
				TAW6	$(A) \leftarrow (W6)$
				TW6A	$(W6) \leftarrow (A)$
				TABPS	$(B) \leftarrow (TPS7-TPS4)$ $(A) \leftarrow (TPS3-TPS0)$
				TPSAB	$(RPS7-RPS4) \leftarrow (B)$ $(TPS7-TPS4) \leftarrow (B)$ $(RPS3-RPS0) \leftarrow (A)$ $(TPS3-TPS0) \leftarrow (A)$
				TAB1	$(B) \leftarrow (T17-T14)$ $(A) \leftarrow (T13-T10)$

Note: p is 0 to 31.

## INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function	Grouping	Mnemonic	Function
Timer operation	T1AB	(R1L7–R1L4) ← (B) (T17–T14) ← (B) (R1L3–R1L0) ← (A) (T13–T10) ← (A)	Input/Output operation	CLD	(D) ← 1
	T1HAB	(R1H7–R1H4) ← (B) (R1H3–R1H0) ← (A)		RD	(D(Y)) ← 0 (Y) = 0 to 5
	TAB2	(B) ← (T27–T24) (A) ← (T23–T20)		SD	(D(Y)) ← 1 (Y) = 0 to 5
	T2AB	(R2L7–R2L4) ← (B) (T27–T24) ← (B) (R2L3–R2L0) ← (A) (T23–T20) ← (A)		SZD	(D(Y)) = 0 ? (Y) = 0 to 5
	T2HAB	(R2H7–R2H4) ← (B) (R2H3–R2H0) ← (A)		TFR0A	(FR0) ← (A)
	T1R1L	(T17–T10) ← (R1L7–R1L0)		TFR1A	(FR1) ← (A)
	T2R2L	(T27–T20) ← (R2L7–R2L0)		TFR2A	(FR2) ← (A)
	SNZT1	V12 = 0: (T1F) = 1 ? (T1F) ← 0 V12 = 1: SNZT1 = NOP		TFR3A	(FR3) ← (A)
	SNZT2	V13 = 0: (T2F) = 1 ? (T2F) ← 0 V13 = 1: SNZT2 = NOP		TC1A	(C1) ← (A)
	IAP0	(A) ← (P0)		TK0A	(K0) ← (A)
Input/Output operation	OP0A	(P0) ← (A)	Input/Output operation	TAK0	(A) ← (K0)
	IAP1	(A) ← (P1)		TK1A	(K1) ← (A)
	OP1A	(P1) ← (A)		TAK1	(A) ← (K1)
	IAP2	(A1, A0) ← (P21, P20) (A3, A2) ← 0		TK2A	(K2) ← (A)
	OP2A	(P21, P20) ← (A1, A0)		TAK2	(A) ← (K2)
	IAP3	(A1, A0) ← (P31, P30) (A3, A2) ← 0		TPU0A	(PU0) ← (A)
	OP3A	(P31, P30) ← (A1, A0)		TAPU0	(A) ← (PU0)
				TPU1A	(PU1) ← (A)
				TPU2A	(PU2) ← (A)
				TAPU1	(A) ← (PU1)
				TAPU2	(A) ← (PU2)
				TL1A	(L1) ← (A)
				TAL1	(A) ← (L1)

## INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function	Grouping	Mnemonic	Function
Serial interface operation	TABSI	$(B) \leftarrow (SI_7-SI_4)$ $(A) \leftarrow (SI_3-SI_0)$	Other operation	NOP	$(PC) \leftarrow (PC) + 1$
	TSIAB	$(SI_7-SI_4) \leftarrow (B)$ $(SI_3-SI_0) \leftarrow (A)$		POF	RAM back-up
	SST	$(SIOF) \leftarrow 0$ Serial interface transmit/receive starting		EPOF	POF instruction valid
	SNZSI	$V23=0: (SIOF)=1?$ $(SIOF) \leftarrow 0$ $V23 = 1: SNZSI = NOP$		SNZP	$(P) = 1 ?$
	TAJ1	$(A) \leftarrow (J1)$		DWDT	Stop of watchdog timer function enabled
	TJ1A	$(J1) \leftarrow (A)$		WRST	$(WDF1) = 1 ?,$ $(WDF1) \leftarrow 0$
Clock operation	CRCK	RC oscillator selected		SRST	System reset
	TRGA	$(RG_0) \leftarrow (A_0)$		RUPT	$(UPTF) \leftarrow 0$
	TAMR	$(A) \leftarrow (MR)$		SUPT	$(UPTF) \leftarrow 1$
	TMRA	$(MR) \leftarrow (A)$		SVDE**	Voltage drop detection circuit valid at RAM back-up
A/D conversion operation	TABAD	$Q13 = 0,$ $(B) \leftarrow (AD_9-AD_6)$ $(A) \leftarrow (AD_5-AD_2)$ $Q13 = 1,$ $(B) \leftarrow (AD_7-AD_4)$ $(A) \leftarrow (AD_3-AD_0)$			
	TALA	$(A_3, A_2) \leftarrow (AD_1, AD_0)$ $(A_1, A_0) \leftarrow 0$			
	TADAB	$Q13 = 1 : (AD_7-AD_4) \leftarrow (B)$ $(AD_3-AD_0) \leftarrow (A)$ $Q13 = 0 : TABAD = NOP$			
	TAQ1	$(A) \leftarrow (Q1)$			
	TQ1A	$(Q1) \leftarrow (A)$			
	ADST	$(ADF) \leftarrow 0$ $Q13 = 0 : A/D conversion starting$ $Q13 = 1 : Comparator operation starting$			
	SNZAD	$V22 = 0: (ADF) = 1 ?$ $(ADF) \leftarrow 0$ $V22 = 1: SNZAD = NOP$			

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

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET)****A n (Add n and accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>n</td><td>n</td><td>n</td><td>n</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	1	1	0	n	n	n	n	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>6</td><td>n</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	6	n	16			1	1	-	Overflow = 0
0	0	0	1	1	0	n	n	n	n																							
2																																
0	6	n																														
16																																
<b>Operation:</b> (A) $\leftarrow$ (A) + n n = 0 to 15						<b>Grouping:</b> Arithmetic operation																										
<b>Description:</b> Adds the value n in the immediate field to register A, and stores a result in register A. The contents of carry flag CY remains unchanged. Skips the next instruction when there is no overflow as the result of operation. Executes the next instruction when there is overflow as the result of operation.																																

**ADST (A/D conversion Start)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	1	0	0	1	1	1	1	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>9</td><td>F</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	9	F	16			1	1	-	-
1	0	1	0	0	1	1	1	1	1																							
2																																
2	9	F																														
16																																
<b>Operation:</b> (ADF) $\leftarrow$ 0 Q13 = 0: A/D conversion starting Q13 = 1: Comparator operation starting (Q13 : bit 3 of A/D control register Q1)						<b>Grouping:</b> A/D conversion operation																										
<b>Description:</b> Clears (0) to A/D conversion completion flag ADF, and the A/D conversion at the A/D conversion mode (Q13 = 0) or the comparator operation at the comparator mode (Q13 = 1) is started.																																

**AM (Add accumulator and Memory)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	0	0	0	1	0	1	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>A</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	0	A	16			1	1	-	-
0	0	0	0	0	0	1	0	1	0																							
2																																
0	0	A																														
16																																
<b>Operation:</b> (A) $\leftarrow$ (A) + (M(DP))						<b>Grouping:</b> Arithmetic operation																										
<b>Description:</b> Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	0	0	0	1	0	1	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>B</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	0	B	16			1	1	0/1	-
0	0	0	0	0	0	1	0	1	1																							
2																																
0	0	B																														
16																																
<b>Operation:</b> (A) $\leftarrow$ (A) + (M(DP)) + (CY) (CY) $\leftarrow$ Carry						<b>Grouping:</b> Arithmetic operation																										
<b>Description:</b> Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY.																																

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****AND (logical AND between accumulator and memory)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																													
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>1</td><td>8</td><td colspan="7"><sub>16</sub></td></tr></table>	0	0	0	0	0	1	1	0	0	0	<sub>2</sub>										0	1	8	<sub>16</sub>							1	1	-	-
0	0	0	0	0	1	1	0	0	0																										
<sub>2</sub>																																			
0	1	8	<sub>16</sub>																																
<b>Operation:</b> (A) $\leftarrow$ (A) AND (M(DP))				<b>Grouping:</b> Arithmetic operation <b>Description:</b> Takes the AND operation between the contents of register A and the contents of M(DP), and stores the result in register A.																															

**B a (Branch to address a)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																														
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>1</td><td>a6</td><td>a5</td><td>a4</td><td>a3</td><td>a2</td><td>a1</td><td>a0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>1</td><td>8</td><td>+a</td><td>a</td><td colspan="7"><sub>16</sub></td></tr></table>	0	1	1	a6	a5	a4	a3	a2	a1	a0	<sub>2</sub>										1	8	+a	a	<sub>16</sub>							1	1	-	-
0	1	1	a6	a5	a4	a3	a2	a1	a0																											
<sub>2</sub>																																				
1	8	+a	a	<sub>16</sub>																																
<b>Operation:</b> (PCL) $\leftarrow$ a6 to a0				<b>Grouping:</b> Branch operation <b>Description:</b> Branch within a page : Branches to address a in the identical page.																																
				<b>Note:</b> Specify the branch address within the page including this instruction.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																														
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>p4</td><td>p3</td><td>p2</td><td>p1</td><td>p0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>E</td><td>+p</td><td>p</td><td colspan="7"><sub>16</sub></td></tr></table>	0	0	1	1	1	p4	p3	p2	p1	p0	<sub>2</sub>										0	E	+p	p	<sub>16</sub>							2	2	-	-
0	0	1	1	1	p4	p3	p2	p1	p0																											
<sub>2</sub>																																				
0	E	+p	p	<sub>16</sub>																																
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>a6</td><td>a5</td><td>a4</td><td>a3</td><td>a2</td><td>a1</td><td>a0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>a</td><td>a</td><td>a</td><td colspan="7"><sub>16</sub></td></tr></table>	1	0	0	a6	a5	a4	a3	a2	a1	a0	<sub>2</sub>										2	a	a	a	<sub>16</sub>							2	2	-	-
1	0	0	a6	a5	a4	a3	a2	a1	a0																											
<sub>2</sub>																																				
2	a	a	a	<sub>16</sub>																																
<b>Grouping:</b> Branch operation				<b>Description:</b> Branch out of a page : Branches to address a in page p.																																
				<b>Note:</b> p is 0 to 31.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																														
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>1</td><td>0</td><td colspan="7"><sub>16</sub></td></tr></table>	0	0	0	0	0	1	0	0	0	0	<sub>2</sub>										0	1	0	<sub>16</sub>							2	2	-	-	
0	0	0	0	0	1	0	0	0	0																											
<sub>2</sub>																																				
0	1	0	<sub>16</sub>																																	
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>p4</td><td>0</td><td>0</td><td>p3</td><td>p2</td><td>p1</td><td>p0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>p</td><td>p</td><td>p</td><td colspan="7"><sub>16</sub></td></tr></table>	1	0	0	p4	0	0	p3	p2	p1	p0	<sub>2</sub>										2	p	p	p	<sub>16</sub>							2	2	-	-
1	0	0	p4	0	0	p3	p2	p1	p0																											
<sub>2</sub>																																				
2	p	p	p	<sub>16</sub>																																
<b>Grouping:</b> Branch operation				<b>Description:</b> Branch out of a page : Branches to address (DR2 DR1 DR0 A3 A2 A1 A0) <sub>2</sub> specified by registers D and A in page p.																																
				<b>Note:</b> p is 0 to 31.																																

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****BM a (Branch and Mark to address a in page 2)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>0</td><td>a6</td><td>a5</td><td>a4</td><td>a3</td><td>a2</td><td>a1</td><td>a0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr></table>	0	1	0	a6	a5	a4	a3	a2	a1	a0	<sub>2</sub>										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>a</td><td>a</td></tr><tr><td colspan="3"><sub>16</sub></td></tr></table>	1	a	a	<sub>16</sub>			1	1	-	-
0	1	0	a6	a5	a4	a3	a2	a1	a0																							
<sub>2</sub>																																
1	a	a																														
<sub>16</sub>																																
<b>Operation:</b> (SP) $\leftarrow$ (SP) + 1 (SK(SP)) $\leftarrow$ (PC) (PCH) $\leftarrow$ 2 (PCL) $\leftarrow$ a6-a0						<b>Grouping:</b> Subroutine call operation																										
						<b>Description:</b> Call the subroutine in page 2 : Calls the subroutine at address a in page 2.																										
						<b>Note:</b> Subroutine extending from page 2 to another page can also be called with the BM instruction when it starts on page 2. Be careful not to over the stack because the maximum level of subroutine nesting is 8.																										

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																												
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>p4</td><td>p3</td><td>p2</td><td>p1</td><td>p0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr></table>	0	0	1	1	0	p4	p3	p2	p1	p0	<sub>2</sub>										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>C</td><td>+p</td><td>p</td></tr><tr><td colspan="4"><sub>16</sub></td></tr></table>	0	C	+p	p	<sub>16</sub>				2	2	-	-
0	0	1	1	0	p4	p3	p2	p1	p0																									
<sub>2</sub>																																		
0	C	+p	p																															
<sub>16</sub>																																		
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>a6</td><td>a5</td><td>a4</td><td>a3</td><td>a2</td><td>a1</td><td>a0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr></table>	1	0	0	a6	a5	a4	a3	a2	a1	a0	<sub>2</sub>										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>a</td><td>a</td></tr><tr><td colspan="3"><sub>16</sub></td></tr></table>	2	a	a	<sub>16</sub>			<b>Grouping:</b> Subroutine call operation					
1	0	0	a6	a5	a4	a3	a2	a1	a0																									
<sub>2</sub>																																		
2	a	a																																
<sub>16</sub>																																		
<b>Operation:</b> (SP) $\leftarrow$ (SP) + 1 (SK(SP)) $\leftarrow$ (PC) (PCH) $\leftarrow$ p (PCL) $\leftarrow$ a6-a0						<b>Description:</b> Call the subroutine : Calls the subroutine at address a in page p.																												
						<b>Note:</b> p is 0 to 31. Be careful not to over the stack because the maximum level of subroutine nesting is 8.																												

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr></table>	0	0	0	0	1	1	0	0	0	0	<sub>2</sub>										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>3</td><td>0</td></tr><tr><td colspan="3"><sub>16</sub></td></tr></table>	0	3	0	<sub>16</sub>			2	2	-	-
0	0	0	0	1	1	0	0	0	0																							
<sub>2</sub>																																
0	3	0																														
<sub>16</sub>																																
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>p4</td><td>0</td><td>0</td><td>p3</td><td>p2</td><td>p1</td><td>p0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr></table>	1	0	0	p4	0	0	p3	p2	p1	p0	<sub>2</sub>										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>p</td><td>p</td></tr><tr><td colspan="3"><sub>16</sub></td></tr></table>	2	p	p	<sub>16</sub>			<b>Grouping:</b> Subroutine call operation			
1	0	0	p4	0	0	p3	p2	p1	p0																							
<sub>2</sub>																																
2	p	p																														
<sub>16</sub>																																
<b>Operation:</b> (SP) $\leftarrow$ (SP) + 1 (SK(SP)) $\leftarrow$ (PC) (PCH) $\leftarrow$ p (PCL) $\leftarrow$ (DR2-DR0, A3-A0)						<b>Description:</b> Call the subroutine : Calls the subroutine at address (DR2 DR1 DR0 A3 A2 A1 A0) <sub>2</sub> specified by registers D and A in page p.																										
						<b>Note:</b> p is 0 to 31. Be careful not to over the stack because the maximum level of subroutine nesting is 8.																										

**CLD (CLeaR port D)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td colspan="10"><sub>2</sub></td></tr></table>	0	0	0	0	0	1	0	0	0	1	<sub>2</sub>										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="3"><sub>16</sub></td></tr></table>	0	1	1	<sub>16</sub>			1	1	-	-
0	0	0	0	0	1	0	0	0	1																							
<sub>2</sub>																																
0	1	1																														
<sub>16</sub>																																
<b>Operation:</b> (D) $\leftarrow$ 1						<b>Grouping:</b> Input/Output operation																										
						<b>Description:</b> Sets (1) to port D.																										

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****CMA (CoMplement of Accumulator)**

Instruction	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
code	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table> <sub>2</sub>	0	0	0	0	0	1	1	1	0	0	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>C</td></tr></table> <sub>16</sub>	0	1	C	1	1	-	-
0	0	0	0	0	1	1	1	0	0										
0	1	C																	
<b>Operation:</b> (A) $\leftarrow \overline{(A)}$				<b>Grouping:</b> Arithmetic operation															
<b>Description:</b> Stores the one's complement for register A's contents in register A.																			

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

Instruction	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
code	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table> <sub>2</sub>	1	0	1	0	0	1	1	0	1	1	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>9</td><td>B</td></tr></table> <sub>16</sub>	2	9	B	1	1	-	-
1	0	1	0	0	1	1	0	1	1										
2	9	B																	
<b>Operation:</b> RC oscillation circuit selected				<b>Grouping:</b> Other operation															
<b>Description:</b> Selects the RC oscillation circuit for main clock f(XIN).																			

**DEY (DEcrement register Y)**

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

**DI (Disable Interrupt)**

Instruction	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
code	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table> <sub>2</sub>	0	0	0	0	0	0	0	1	0	0	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>4</td></tr></table> <sub>16</sub>	0	0	4	1	1	-	-
0	0	0	0	0	0	0	1	0	0										
0	0	4																	
<b>Operation:</b> (INTE) $\leftarrow 0$				<b>Grouping:</b> Interrupt control operation															
<b>Description:</b> Clears (0) to interrupt enable flag INTE, and disables the interrupt.																			
<b>Note:</b> Interrupt is disabled by executing the DI instruction after executing 1 machine cycle.																			

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****DWDT (Disable WatchDog Timer)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																											
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>9</td><td>C</td><td colspan="5"><sub>16</sub></td></tr></table>	1	0	1	0	0	1	1	1	0	0	<sub>2</sub>										2	9	C	<sub>16</sub>					1	1	-	-
1	0	1	0	0	1	1	1	0	0																								
<sub>2</sub>																																	
2	9	C	<sub>16</sub>																														
<b>Operation:</b> Stop of watchdog timer function enabled				<b>Grouping:</b> Other operation																													
<b>Description:</b> Stops the watchdog timer function by the WRST instruction after executing the DWDT instruction.																																	

**EI (Enable Interrupt)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																											
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>0</td><td>5</td><td colspan="5"><sub>16</sub></td></tr></table>	0	0	0	0	0	0	0	1	0	1	<sub>2</sub>										0	0	5	<sub>16</sub>					1	1	-	-
0	0	0	0	0	0	0	1	0	1																								
<sub>2</sub>																																	
0	0	5	<sub>16</sub>																														
<b>Operation:</b> (INTE) ← 1				<b>Grouping:</b> Interrupt control operation																													
<b>Description:</b> Sets (1) to interrupt enable flag INTE, and enables the interrupt.																																	
<b>Note:</b> Interrupt is enabled by executing the EI instruction after executing 1 machine cycle.																																	

**EPOF (Enable POF instruction)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																											
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>5</td><td>B</td><td colspan="5"><sub>16</sub></td></tr></table>	0	0	0	1	0	1	1	0	1	1	<sub>2</sub>										0	5	B	<sub>16</sub>					1	1	-	-
0	0	0	1	0	1	1	0	1	1																								
<sub>2</sub>																																	
0	5	B	<sub>16</sub>																														
<b>Operation:</b> POF instruction valid				<b>Grouping:</b> Other operation																													
<b>Description:</b> Makes the immediate after POF instruction valid by executing the EPOF instruction.																																	

**IAP0 (Input Accumulator from port P0)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																											
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>6</td><td>0</td><td colspan="5"><sub>16</sub></td></tr></table>	1	0	0	1	1	0	0	0	0	0	<sub>2</sub>										2	6	0	<sub>16</sub>					1	1	-	-
1	0	0	1	1	0	0	0	0	0																								
<sub>2</sub>																																	
2	6	0	<sub>16</sub>																														
<b>Operation:</b> (A) ← (P0)				<b>Grouping:</b> Input/Output operation																													
<b>Description:</b> Transfers the input of port P0 to register A.																																	

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****IAP1 (Input Accumulator from port P1)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	1	1	0	0	0	0	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>6</td><td>1</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	6	1	16			1	1	-	-
1	0	0	1	1	0	0	0	0	1																							
2																																
2	6	1																														
16																																
<b>Operation:</b> (A) $\leftarrow$ (P1)				<b>Grouping:</b> Input/Output operation <b>Description:</b> Transfers the input of port P1 to register A.																												

**IAP2 (Input Accumulator from port P2)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	1	1	0	0	0	1	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>6</td><td>2</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	6	2	16			1	1	-	-
1	0	0	1	1	0	0	0	1	0																							
2																																
2	6	2																														
16																																
<b>Operation:</b> (A1, A0) $\leftarrow$ (P21, P20) (A3, A2) $\leftarrow$ 0				<b>Grouping:</b> Input/Output operation <b>Description:</b> Transfers the input of port P2 to the low-order 2 bits (A1, A0) of register A. <b>Note:</b> After this instruction is executed, "0" is stored to the high-order 2 bits (A3, A2) of register A.																												

**IAP3 (Input Accumulator from port P3)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	1	1	0	0	0	1	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>6</td><td>3</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	6	3	16			1	1	-	-
1	0	0	1	1	0	0	0	1	1																							
2																																
2	6	3																														
16																																
<b>Operation:</b> (A1, A0) $\leftarrow$ (P31, P30) (A3, A2) $\leftarrow$ 0				<b>Grouping:</b> Input/Output operation <b>Description:</b> Transfers the input of port P3 to the low-order 2 bits (A1, A0) of register A. <b>Note:</b> After this instruction is executed, "0" is stored to the high-order 2 bits (A3, A2) of register A.																												

**INY (INcrement register Y)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	0	0	1	0	0	1	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>3</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	1	3	16			1	1	-	(Y) = 0
0	0	0	0	0	1	0	0	1	1																							
2																																
0	1	3																														
16																																
<b>Operation:</b> (Y) $\leftarrow$ (Y) + 1				<b>Grouping:</b> RAM addresses <b>Description:</b> Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. When the contents of register Y is not 0, the next instruction is executed.																												

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****LA n (Load n in Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>n</td><td>n</td><td>n</td><td>n</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	1	1	1	n	n	n	n	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>7</td><td>n</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	7	n	16			1	1	-	Continuous description
0	0	0	1	1	1	n	n	n	n																							
2																																
0	7	n																														
16																																
<b>Operation:</b> (A) $\leftarrow$ n n = 0 to 15						<b>Grouping:</b> Arithmetic operation																										
<b>Description:</b> Loads the value n in the immediate field to register A. When the LA instructions are continuously coded and executed, only the first LA instruction is executed and other LA instructions coded continuously are skipped.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>1</td><td>x3</td><td>x2</td><td>x1</td><td>x0</td><td>y3</td><td>y2</td><td>y1</td><td>y0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	1	x3	x2	x1	x0	y3	y2	y1	y0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>3</td><td>x</td><td>y</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	3	x	y	16			1	1	-	Continuous description
1	1	x3	x2	x1	x0	y3	y2	y1	y0																							
2																																
3	x	y																														
16																																
<b>Operation:</b> (X) $\leftarrow$ x x = 0 to 15 (Y) $\leftarrow$ y y = 0 to 15						<b>Grouping:</b> RAM addresses																										
<b>Description:</b> Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y. When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																												
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>z1</td><td>z0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	1	0	0	1	0	z1	z0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>4</td><td>8</td><td>+z</td></tr><tr><td colspan="4" style="text-align: right;">16</td></tr></table>	0	4	8	+z	16				1	1	-	-
0	0	0	1	0	0	1	0	z1	z0																									
2																																		
0	4	8	+z																															
16																																		
<b>Operation:</b> (Z) $\leftarrow$ z z = 0 to 3						<b>Grouping:</b> RAM addresses																												
<b>Description:</b> Loads the value z in the immediate field to register Z.																																		

**NOP (No OPeration)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	0	0	0	0	0	0	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	0	0	16			1	1	-	-
0	0	0	0	0	0	0	0	0	0																							
2																																
0	0	0																														
16																																
<b>Operation:</b> (PC) $\leftarrow$ (PC) + 1						<b>Grouping:</b> Other operation																										
<b>Description:</b> No operation; Adds 1 to program counter value, and others remain unchanged.																																

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****OP0A (Output port P0 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																							
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	0	1	0	0	0	0	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>2</td><td>0</td></tr></table>	2	2	0	1	1	-	-
1	0	0	0	1	0	0	0	0	0																				
2																													
2	2	0																											
<b>Operation:</b> (P0) $\leftarrow$ (A)						<b>Grouping:</b> Input/Output operation																							
<b>Description:</b> Outputs the contents of register A to port P0.																													

**OP1A (Output port P1 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																							
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	0	1	0	0	0	0	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>2</td><td>1</td></tr></table>	2	2	1	1	1	-	-
1	0	0	0	1	0	0	0	0	1																				
2																													
2	2	1																											
<b>Operation:</b> (P1) $\leftarrow$ (A)						<b>Grouping:</b> Input/Output operation																							
<b>Description:</b> Outputs the contents of register A to port P1.																													

**OP2A (Output port P2 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																							
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	0	1	0	0	0	1	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>2</td><td>2</td></tr></table>	2	2	2	1	1	-	-
1	0	0	0	1	0	0	0	1	0																				
2																													
2	2	2																											
<b>Operation:</b> (P21, P20) $\leftarrow$ (A1, A0)						<b>Grouping:</b> Input/Output operation																							
<b>Description:</b> Outputs the contents of the low-order 2 bits (A1, A0) of register A to port P2.																													

**OP3A (Output port P3 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																							
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	0	1	0	0	0	1	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>2</td><td>3</td></tr></table>	2	2	3	1	1	-	-
1	0	0	0	1	0	0	0	1	1																				
2																													
2	2	3																											
<b>Operation:</b> (P31, P30) $\leftarrow$ (A1, A0)						<b>Grouping:</b> Input/Output operation																							
<b>Description:</b> Outputs the contents of the low-order 2 bits (A1, A0) of register A to port P3.																													

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****OR (logical OR between accumulator and memory)**

Instruction	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
code	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	0	0	1	1	0	0	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>9</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	1	9	16			1	1	-	-
0	0	0	0	0	1	1	0	0	1																							
2																																
0	1	9																														
16																																
<b>Operation:</b> (A) $\leftarrow$ (A) OR (M(DP))				<b>Grouping:</b> Arithmetic operation <b>Description:</b> Takes the OR operation between the contents of register A and the contents of M(DP), and stores the result in register A.																												

**POF (Power OFF)**

Instruction	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
code	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	0	0	0	0	0	1	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>2</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	0	2	16			1	1	-	-
0	0	0	0	0	0	0	0	1	0																							
2																																
0	0	2																														
16																																
<b>Operation:</b> RAM back-up				<b>Grouping:</b> Other operation <b>Description:</b> Puts the system in RAM back-up state by executing the POF instruction after executing the EPOF instruction.																												
				<b>Note:</b> If the EPOF instruction is not executed just before this instruction, this instruction is equivalent to the NOP instruction.																												

**RAR (Rotate Accumulator Right)**

Instruction	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
code	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	0	0	1	1	1	0	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>D</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	1	D	16			1	1	0/1	-
0	0	0	0	0	1	1	1	0	1																							
2																																
0	1	D																														
16																																
<b>Operation:</b> $\rightarrow$ [CY] $\rightarrow$ [A3A2A1A0]				<b>Grouping:</b> Arithmetic operation <b>Description:</b> Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right.																												

**RB j (Reset Bit)**

Instruction	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																												
code	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>j</td><td>j</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	1	0	0	1	1	j	j	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>4</td><td>C</td><td>+j</td></tr><tr><td colspan="4" style="text-align: right;">16</td></tr></table>	0	4	C	+j	16				1	1	-	-
0	0	0	1	0	0	1	1	j	j																									
2																																		
0	4	C	+j																															
16																																		
<b>Operation:</b> (Mj(DP)) $\leftarrow$ 0 j = 0 to 3				<b>Grouping:</b> Bit operation <b>Description:</b> Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).																														

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****RC (Reset Carry flag)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition												
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr></table> <sub>2</sub>	0	0	0	0	0	0	1	1	0	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>6</td></tr></table> <sub>16</sub>	0	0	6	1	1	0	-
0	0	0	0	0	0	1	1	0										
0	0	6																
<b>Operation:</b> (CY) $\leftarrow$ 0				<b>Grouping:</b> Arithmetic operation														
				<b>Description:</b> Clears (0) to carry flag CY.														

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table> <sub>2</sub>	0	0	0	0	0	1	0	1	0	0	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>4</td></tr></table> <sub>16</sub>	0	1	4	1	1	-	-
0	0	0	0	0	1	0	1	0	0										
0	1	4																	
<b>Operation:</b> (D(Y)) $\leftarrow$ 0 However, (Y) = 0 to 5				<b>Grouping:</b> Input/Output operation															
				<b>Description:</b> Clears (0) to a bit of port D specified by register Y. <b>Note:</b> (Y) = 0 to 5. Do not execute this instruction if values except above are set to register Y.															

**RT (ReTurn from subroutine)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table> <sub>2</sub>	0	0	0	1	0	0	0	1	0	0	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>4</td><td>4</td></tr></table> <sub>16</sub>	0	4	4	1	2	-	-
0	0	0	1	0	0	0	1	0	0										
0	4	4																	
<b>Operation:</b> (PC) $\leftarrow$ (SK(SP)) (SP) $\leftarrow$ (SP) - 1				<b>Grouping:</b> Return operation															
				<b>Description:</b> Returns from subroutine to the routine called the subroutine.															

**RTI (ReTurn from Interrupt)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition													
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr></table> <sub>2</sub>	0	0	0	1	0	0	0	1	1	0	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>4</td><td>6</td></tr></table> <sub>16</sub>	0	4	6	1	1	-	-
0	0	0	1	0	0	0	1	1	0										
0	4	6																	
<b>Operation:</b> (PC) $\leftarrow$ (SK(SP)) (SP) $\leftarrow$ (SP) - 1				<b>Grouping:</b> Return operation															
				<b>Description:</b> Returns from interrupt service routine to main routine. Returns each value of data pointer (X, Y, Z), carry flag, skip status, NOP mode status by the continuous description of the LA/LXY instruction, register A and register B to the states just before interrupt.															

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****RTS (ReTurn from subroutine and Skip)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	1	0	0	0	1	0	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>4</td><td>5</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	4	5	16			1	2	-	Skip at uncondition
0	0	0	1	0	0	0	1	0	1																							
2																																
0	4	5																														
16																																
<b>Operation:</b> (PC) $\leftarrow$ (SK(SP)) (SP) $\leftarrow$ (SP) - 1						<b>Grouping:</b> Return operation																										
						<b>Description:</b> Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition.																										

**RUPT (Reset UPT flag)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	1	0	1	1	0	0	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>5</td><td>8</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	5	8	16			1	1	-	-
0	0	0	1	0	1	1	0	0	0																							
2																																
0	5	8																														
16																																
<b>Operation:</b> (UPTF) $\leftarrow$ 0						<b>Grouping:</b> Other operation																										
						<b>Description:</b> Clears (0) to the high-order bit reference enable flag UPTF.																										

**SB j (Set Bit)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																												
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>j</td><td>j</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	1	0	1	1	1	j	j	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>5</td><td>C</td><td>+j</td></tr><tr><td colspan="4" style="text-align: right;">16</td></tr></table>	0	5	C	+j	16				1	1	-	-
0	0	0	1	0	1	1	1	j	j																									
2																																		
0	5	C	+j																															
16																																		
<b>Operation:</b> (Mj(DP)) $\leftarrow$ 0 j = 0 to 3						<b>Grouping:</b> Bit operation																												
						<b>Description:</b> Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).																												

**SC (Set Carry flag)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	0	0	0	0	1	1	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>7</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	0	7	16			1	1	1	-
0	0	0	0	0	0	0	1	1	1																							
2																																
0	0	7																														
16																																
<b>Operation:</b> (CY) $\leftarrow$ 1						<b>Grouping:</b> Arithmetic operation																										
						<b>Description:</b> Sets (1) to carry flag CY.																										

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****SD (Set port D specified by register Y)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																												
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>1</td><td>5</td><td colspan="6"><sub>16</sub></td></tr></table>	0	0	0	0	0	1	0	1	0	1	<sub>2</sub>										0	1	5	<sub>16</sub>						1	1	-	-
0	0	0	0	0	1	0	1	0	1																									
<sub>2</sub>																																		
0	1	5	<sub>16</sub>																															
<b>Operation:</b> (D(Y)) $\leftarrow$ 1 (Y) = 0 to 5						<b>Grouping:</b> Input/Output operation																												
						<b>Description:</b> Sets (1) to a bit of port D specified by register Y. <b>Note:</b> (Y) = 0 to 5. Do not execute this instruction if values except above are set to register Y.																												

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																														
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>2</td><td>5</td><td colspan="6"><sub>16</sub></td></tr></table>	0	0	0	0	1	0	0	1	0	1	<sub>2</sub>										0	2	5	<sub>16</sub>						2	2	-	(A) = n		
0	0	0	0	1	0	0	1	0	1																											
<sub>2</sub>																																				
0	2	5	<sub>16</sub>																																	
<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>n</td><td>n</td><td>n</td><td>n</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>7</td><td>n</td><td colspan="6"><sub>16</sub></td></tr></table>								0	0	0	1	1	1	n	n	n	n	<sub>2</sub>										0	7	n	<sub>16</sub>					
0	0	0	1	1	1	n	n	n	n																											
<sub>2</sub>																																				
0	7	n	<sub>16</sub>																																	
<b>Operation:</b> (A) = n ? n = 0 to 15						<b>Grouping:</b> Comparison operation																														
						<b>Description:</b> Skips the next instruction when the contents of register A is equal to the value n in the immediate field. Executes the next instruction when the contents of register A is not equal to the value n in the immediate field.																														

**SEAM (Skip Equal, Accumulator with Memory)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																												
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>2</td><td>6</td><td colspan="6"><sub>16</sub></td></tr></table>	0	0	0	0	1	0	0	1	1	0	<sub>2</sub>										0	2	6	<sub>16</sub>						1	1	-	(A) = (M(DP))
0	0	0	0	1	0	0	1	1	0																									
<sub>2</sub>																																		
0	2	6	<sub>16</sub>																															
<b>Operation:</b> (A) = (M(DP)) ?						<b>Grouping:</b> Comparison operation																												
						<b>Description:</b> Skips the next instruction when the contents of register A is equal to the contents of M(DP). Executes the next instruction when the contents of register A is not equal to the contents of M(DP).																												

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																												
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>3</td><td>8</td><td colspan="6"><sub>16</sub></td></tr></table>	0	0	0	0	1	1	1	0	0	0	<sub>2</sub>										0	3	8	<sub>16</sub>						1	1	-	V10 = 0: (EXF0) = 1
0	0	0	0	1	1	1	0	0	0																									
<sub>2</sub>																																		
0	3	8	<sub>16</sub>																															
<b>Operation:</b> V10 = 0: (EXF0) = 1 ? (EXF0) $\leftarrow$ 0 V10 = 1: SNZ0 = NOP (V10 : bit 0 of the interrupt control register V1)						<b>Grouping:</b> Interrupt operation																												
						<b>Description:</b> When V10 = 0 : Clears (0) to the EXF0 flag and skips the next instruction when external 0 interrupt request flag EXF0 is "1." When the EXF0 flag is "0," executes the next instruction. When V10 = 1 : This instruction is equivalent to the NOP instruction.																												

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****SNZAD (Skip if Non Zero condition of A/D conversion completion flag)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																							
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td colspan="9" style="text-align: right;">2</td></tr><tr><td>2</td><td>8</td><td>7</td><td colspan="3" style="text-align: right;">16</td></tr></table>	1	0	1	0	0	0	1	1	1	2									2	8	7	16			1	1	-	V22 = 0: (ADF) = 1
1	0	1	0	0	0	1	1	1																					
2																													
2	8	7	16																										
<b>Operation:</b> V22 = 0: (ADF) = 1 ? (ADF) ← 0 V22 = 1: SNZAD = NOP (V22 : bit 2 of the interrupt control register V2)						<b>Grouping:</b> A/D conversion operation <b>Description:</b> When V22 = 0 : Clears (0) to the ADF flag and skips the next instruction when A/D conversion completion flag ADF is "1." After skipping, . When the ADF flag is "0," executes the next instruction. When V22 = 1 : This instruction is equivalent to the NOP instruction.																							

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																							
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td colspan="9" style="text-align: right;">2</td></tr><tr><td>0</td><td>3</td><td>A</td><td colspan="3" style="text-align: right;">16</td></tr></table>	0	0	0	0	1	1	1	0	1	2									0	3	A	16			1	1	-	I12 = 0 : (INT) = "L" I12 = 1 : (INT) = "H"
0	0	0	0	1	1	1	0	1																					
2																													
0	3	A	16																										
<b>Operation:</b> I12 = 0 : (INT) = "L" ? I12 = 1 : (INT) = "H" ? (I12 : bit 2 of the interrupt control register I1)						<b>Grouping:</b> Interrupt operation <b>Description:</b> When I12 = 0 : Skips the next instruction when the level of INT pin is "L." Executes the next instruction when the level of INT pin is "H." When I12 = 1 : Skips the next instruction when the level of INT pin is "H." Executes the next instruction when the level of INT pin is "L."																							

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																							
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="9" style="text-align: right;">2</td></tr><tr><td>0</td><td>0</td><td>3</td><td colspan="3" style="text-align: right;">16</td></tr></table>	0	0	0	0	0	0	0	1	1	2									0	0	3	16			1	1	-	(P) = 1
0	0	0	0	0	0	0	1	1																					
2																													
0	0	3	16																										
<b>Operation:</b> (P) = 1 ?						<b>Grouping:</b> Other operation <b>Description:</b> Skips the next instruction when the P flag is "1". After skipping, the P flag remains unchanged. Executes the next instruction when the P flag is "0."																							

**SNZSI (Skip if Non Zero condition of Serial Interface interrupt request flag)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																							
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td colspan="9" style="text-align: right;">2</td></tr><tr><td>2</td><td>8</td><td>8</td><td colspan="3" style="text-align: right;">16</td></tr></table>	1	0	1	0	0	0	1	0	0	2									2	8	8	16			1	1	-	V23 = 0: (SIOF) = 1
1	0	1	0	0	0	1	0	0																					
2																													
2	8	8	16																										
<b>Operation:</b> V23=0: (SIOF)=1? (SIOF) ← 0 V23 = 1: SNZSI = NOP						<b>Grouping:</b> Serial interface operation <b>Description:</b> Clears (0) to SIOF flag and skips the next instruction when the contents of bit 3 (V23) of interrupt control register V2 is "0" and contents of SIOF flag is "1." When V23 = 1: This instruction is equivalent to the NOP instruction.																							

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****SNZT1 (Skip if Non Zero condition of Timer 1 interrupt request flag)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 1 0 0 0 0 0 0 <sub>2</sub>	2 8 0 <sub>16</sub>	1	1	-	V12 = 0: (T1F) = 1
<b>Operation:</b> V12 = 0: (T1F) = 1 ? (T1F) ← 0 V12 = 1: SNZT1 = NOP (V12 = bit 2 of interrupt control register V1)						<b>Grouping:</b> Timer operation <b>Description:</b> When V12 = 0 : Clears (0) to the T1F flag and skips the next instruction when timer 1 interrupt request flag T1F is "1." When the T1F flag is "0," executes the next instruction. When V12 = 1 : This instruction is equivalent to the NOP instruction.

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 1 0 0 0 0 0 1 <sub>2</sub>	2 8 1 <sub>16</sub>	1	1	-	V13 = 0: (T2F) = 1
<b>Operation:</b> V13 = 0: (T2F) = 1 ? (T2F) ← 0 V13 = 1: SNZT2 = NOP (V13 = bit 3 of interrupt control register V1)						<b>Grouping:</b> Timer operation <b>Description:</b> When V13 = 0 : Clears (0) to the T2F flag and skips the next instruction when timer 2 interrupt request flag T2F is "1." When the T2F flag is "0," executes the next instruction. When V13 = 1 : This instruction is equivalent to the NOP instruction.

**SRST (System ReSet)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0 0 0 0 0 0 0 0 0 1 <sub>2</sub>	0 0 1 <sub>16</sub>	1	1	-	-
<b>Operation:</b> System reset						<b>Grouping:</b> Other operation <b>Description:</b> System reset occurs.

**SST (Serial interface transmission/reception STart)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 1 0 0 1 1 1 1 0 <sub>2</sub>	2 9 E <sub>16</sub>	1	1	-	-
<b>Operation:</b> (SIOF) ← 0 Serial interface transmit/receive starting						<b>Grouping:</b> Serial interface operation <b>Description:</b> Clears (0) to SIOF flag and starts serial interface.

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

**SUPT** (Set UPT flag)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0 0 0 1 0 1 1 0 0 1 <sub>2</sub>	0 5 9 <sub>16</sub>	1	1	-	-
Operation:	(UPTF) ← 1				<b>Grouping:</b> Other operation	
					<b>Description:</b> Sets (1) to the high-order bit reference enable flag UPTF. When the table reference instruction (TABP p) is executed, the high-order 2 bits of ROM reference data is transferred to the low-order 2 bits of register D.	

**SVDE** (Set Voltage Detector Enable flag)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 1 0 0 1 0 0 1 1 <sub>2</sub>	2 9 3 <sub>16</sub>	1	1	-	-
Operation:	Voltage drop detection circuit valid at RAM back-up				<b>Grouping:</b> Other operation	
					<b>Description:</b> Validates the voltage drop detection circuit at RAM back-up.	
					<b>Note:</b> This instruction can be executed only for the H version.	

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

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

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

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

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****SZD (Skip if Zero, port D specified by register Y)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	0	1	0	0	1	0	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>2</td><td>4</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	2	4	16						
0	0	0	0	1	0	0	1	0	0																							
2																																
0	2	4																														
16																																
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	0	1	0	1	0	1	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>2</td><td>B</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	2	B	16			2	2	-	$(D(Y)) = 0$ $(Y) = 0$ to 5
0	0	0	0	1	0	1	0	1	1																							
2																																
0	2	B																														
16																																
<b>Operation:</b>	$(D(Y)) = 0 ?$ $(Y) = 0$ to 5			<b>Grouping:</b>	Input/Output operation																											
				<b>Description:</b>	Skips the next instruction when a bit of port D specified by register Y is "0." Executes the next instruction when the bit is "1."																											
				<b>Note:</b>	$(Y) = 0$ to 5. Do not execute this instruction if values except above are set to register Y.																											

**T1AB (Transfer data to timer 1 and register R1L from Accumulator and register B)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	0	1	1	0	0	0	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>3</td><td>0</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	3	0	16			1	1	-	-
1	0	0	0	1	1	0	0	0	0																							
2																																
2	3	0																														
16																																
<b>Operation:</b>	$(R1L7-R1L4) \leftarrow (B)$ $(T17-T14) \leftarrow (B)$ $(R1L3-R1L0) \leftarrow (A)$ $(T13-T10) \leftarrow (A)$			<b>Grouping:</b>	Timer operation																											
				<b>Description:</b>	Transfers the contents of register B to the high-order 4 bits of timer 1 and timer 1 reload register R1L. Transfers the contents of register A to the low-order 4 bits of timer 1 and timer 1 reload register R1L.																											

**T1HAB (Transfer data to register R1H from Accumulator and register B)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	1	0	0	1	0	0	1	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>9</td><td>2</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	9	2	16			1	1	-	-
1	0	1	0	0	1	0	0	1	0																							
2																																
2	9	2																														
16																																
<b>Operation:</b>	$(R1H7-R1H4) \leftarrow (B)$ $(R1H3-R1H0) \leftarrow (A)$			<b>Grouping:</b>	Timer operation																											
				<b>Description:</b>	Transfers the contents of register B to the high-order 4 bits of timer 1 reload register R1H. Transfers the contents of register A to the low-order 4 bits of timer 1 reload register R1H.																											

**T1R1L (Transfer data to timer 1 from register R1L)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	1	0	1	0	0	1	1	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>A</td><td>7</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	A	7	16			1	1	-	-
1	0	1	0	1	0	0	1	1	1																							
2																																
2	A	7																														
16																																
<b>Operation:</b>	$(T17-T10) \leftarrow (R1L7-R1L0)$			<b>Grouping:</b>	Timer operation																											
				<b>Description:</b>	Transfers the contents of timer 1 reload register R1L to timer 1.																											

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****T2AB** (Transfer data to timer 2 and register R2L from Accumulator and register B)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	0	1	1	0	0	0	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>3</td><td>1</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	3	1	16			1	1	-	-
1	0	0	0	1	1	0	0	0	1																							
2																																
2	3	1																														
16																																
<b>Operation:</b> (R2L7–R2L4) ← (B) (T27–T24) ← (B) (R2L3–R2L0) ← (A) (T23–T20) ← (A)						<b>Grouping:</b> Timer operation <b>Description:</b> Transfers the contents of register B to the high-order 4 bits of timer 2 and timer 2 reload register R2L. Transfers the contents of register A to the low-order 4 bits of timer 2 and timer 2 reload register R2L.																										

**T2HAB** (Transfer data to register R2H from Accumulator and register B)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	1	0	0	1	0	1	0	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>9</td><td>4</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	9	4	16			1	1	-	-
1	0	1	0	0	1	0	1	0	0																							
2																																
2	9	4																														
16																																
<b>Operation:</b> (R2H7–R2H4) ← (B) (R2H3–R2H0) ← (A)						<b>Grouping:</b> Timer operation <b>Description:</b> Transfers the contents of register B to the high-order 4 bits of timer 2 reload register R2H. Transfers the contents of register A to the low-order 4 bits of timer 2 reload register R2H.																										

**T2R2L** (Transfer data to timer 2 from register R2L)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	1	0	0	1	0	1	0	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>9</td><td>5</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	9	5	16			1	1	-	-
1	0	1	0	0	1	0	1	0	1																							
2																																
2	9	5																														
16																																
<b>Operation:</b> (T27–T20) ← (R2L7–R2L0)						<b>Grouping:</b> Timer operation <b>Description:</b> Transfers the contents of timer 2 reload register R2L to timer 2.																										

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	0	0	1	1	1	1	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>E</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	1	E	16			1	1	-	-
0	0	0	0	0	1	1	1	1	0																							
2																																
0	1	E																														
16																																
<b>Operation:</b> (A) ← (B)						<b>Grouping:</b> Register to register transfer <b>Description:</b> Transfers the contents of register B to register A.																										

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TAB1** (Transfer data to Accumulator and register B from timer 1)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 1 1 0 0 0 0 <sub>2</sub>	2 7 0 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (B) ← (T17–T14) (A) ← (T13–T10)		<b>Grouping:</b> Timer operation <b>Description:</b> Transfers the high-order 4 bits (T17–T14) of timer 1 to register B. Transfers the low-order 4 bits (T13–T10) of timer 1 to register A.				

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 1 1 0 0 0 1 <sub>2</sub>	2 7 1 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (B) ← (T27–T24) (A) ← (T23–T20)		<b>Grouping:</b> Timer operation <b>Description:</b> Transfers the high-order 4 bits (T27–T24) of timer 2 to register B. Transfers the low-order 4 bits (T23–T20) of timer 2 to register A.				

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 1 1 1 0 0 1 <sub>2</sub>	2 7 9 <sub>16</sub>	1	1	-	-
<b>Operation:</b> In A/D conversion mode (Q13 = 0), (B) ← (AD9–AD6) (A) ← (AD5–AD2) In comparator mode (Q13 = 1), (B) ← (AD7–AD4) (A) ← (AD3–AD0) (Q13 : bit 3 of A/D control register Q1)		<b>Grouping:</b> A/D conversion operation <b>Description:</b> In the A/D conversion mode (Q13 = 0), transfers the high-order 4 bits (AD9–AD6) of register AD to register B, and the middle-order 4 bits (AD5–AD2) of register AD to register A. In the comparator mode (Q13 = 1), transfers the high-order 4 bits (AD7–AD4) of comparator register to register B, and the low-order 4 bits (AD3–AD0) of comparator register to register A.				

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

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

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TABP p (Transfer data to Accumulator and register B from Program memory in page p)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>p4</td><td>p3</td><td>p2</td><td>p1</td><td>p0</td></tr><tr><td>2</td></tr></table>	0	0	1	0	0	p4	p3	p2	p1	p0	2	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>8</td><td>+p</td><td>p</td></tr><tr><td>16</td></tr></table>	0	8	+p	p	16	1	3	-	-
0	0	1	0	0	p4	p3	p2	p1	p0													
2																						
0	8	+p	p																			
16																						
<b>Operation:</b> $(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ $(PCL) \leftarrow (DR_2-DR_0, A_3-A_0)$ $(B) \leftarrow (ROM(PC))^{7-4}$ $(A) \leftarrow (ROM(PC))^{3-0}$ $(UPTF) \leftarrow 1$ $(DR_1, DR_0) \leftarrow (ROM(PC))^{9, 8}$ $(DR_2) \leftarrow 0$ $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$		<b>Grouping:</b> Arithmetic operation <b>Description:</b> Transfers bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 7 to 0 are the ROM pattern in address $(DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0)_2$ specified by registers A and D in page p. When UPTF is 1, Transfers bits 9, 8 to the low-order 2 bits (DR1, DR0) of register D, and "0" is stored to the least significant bit (DR2) of register D. <b>Note:</b> When this instruction is executed, 1 stage of stack register (SK) is used. p is 0 to 31. When this instruction is executed, be careful not to over the stack because 1 stage of stack register is used.																				

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition														
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>2</td><td>7</td><td>5</td></tr><tr><td>16</td></tr></table>	1	0	0	1	1	1	0	1	0	1	2	7	5	16		1	1	-	-
1	0	0	1	1	1	0	1	0	1											
2	7	5																		
16																				
<b>Operation:</b> $(B) \leftarrow (TPS_7-TPS_4)$ $(A) \leftarrow (TPS_3-TPS_0)$		<b>Grouping:</b> Timer operation <b>Description:</b> Transfers the high-order 4 bits of prescaler to register B. Transfers the low-order 4 bits of prescaler to register A.																		

**TABSI (Transfer data to Accumulator and register B from register SI)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition														
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>2</td><td>7</td><td>8</td></tr><tr><td>16</td></tr></table>	1	0	0	1	1	1	1	0	0	0	2	7	8	16		1	1	-	-
1	0	0	1	1	1	1	0	0	0											
2	7	8																		
16																				
<b>Operation:</b> $(B) \leftarrow (SI_7-SI_4)$ $(A) \leftarrow (SI_3-SI_0)$		<b>Grouping:</b> Serial interface operation <b>Description:</b> Transfers the high-order 4 bits of serial interface register SI to register B, and transfers the low-order 4 bits of serial interface register SI to register A.																		

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition															
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>2</td><td>0</td><td>5</td><td>1</td></tr><tr><td>16</td></tr></table>	0	0	0	1	0	1	0	0	0	1	2	0	5	1	16		1	1	-	-
0	0	0	1	0	1	0	0	0	1												
2	0	5	1																		
16																					
<b>Operation:</b> $(A_2-A_0) \leftarrow (DR_2-DR_0)$ $(A_3) \leftarrow 0$		<b>Grouping:</b> Register to register transfer <b>Description:</b> Transfers the contents of register D to the low-order 3 bits (A2-A0) of register A. <b>Note:</b> When this instruction is executed, "0" is stored to the bit 3 (A3) of register A.																			

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TADAB** (Transfer data to register AD from Accumulator from register B)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 1 1 1 0 0 1 <sub>2</sub>	2 3 9 <sub>16</sub>	1	1	-	-
<b>Operation:</b> Q13 = 1: (AD7-AD4) ← (B) (AD3-AD0) ← (A) Q13 = 0: TADAB = NOP						<b>Grouping:</b> A/D conversion operation
						<b>Description:</b> In the comparator mode (Q13 = 1), transfers the contents of register B to the high-order 4 bits (AD7-AD4) of comparator register, and the contents of register A to the low-order 4 bits (AD3-AD0) of comparator register. In the A/D conversion mode (Q13 = 0), this instruction is equivalent to the NOP instruction. (Q13 = bit 3 of A/D control register Q1)

**TAI1** (Transfer data to Accumulator from register I1)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 1 0 0 1 1 <sub>2</sub>	2 5 3 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) ← (I1)						<b>Grouping:</b> Interrupt operation
						<b>Description:</b> Transfers the contents of interrupt control register I1 to register A.

**TAJ1** (Transfer data to Accumulator from register J1)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 0 0 0 0 1 0 <sub>2</sub>	2 4 2 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) ← (J1)						<b>Grouping:</b> Serial interface operation
						<b>Description:</b> Transfers the contents of serial interface control register J1 to register A.

**TAK0** (Transfer data to Accumulator from register K0)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 1 0 1 1 0 <sub>2</sub>	2 5 6 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) ← (K0)						<b>Grouping:</b> Input/Output operation
						<b>Description:</b> Transfers the contents of key-on wakeup control register K0 to register A.

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TAK1** (Transfer data to Accumulator from register K1)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 1 1 0 0 1 <sub>2</sub>	2 5 9 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) ← (K1)						<b>Grouping:</b> Input/Output operation
<b>Description:</b> Transfers the contents of key-on wakeup control register K1 to register A.						

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 1 1 0 1 0 <sub>2</sub>	2 5 A <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) ← (K2)						<b>Grouping:</b> Input/Output operation
<b>Description:</b> Transfers the contents of key-on wakeup control register K2 to register A.						

**TAL1** (Transfer data to Accumulator from register L1)

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 1 0 0 0 1 0 1 0 <sub>2</sub>	2 4 A <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) ← (L1)						<b>Grouping:</b> Input/Output operation
<b>Description:</b> Transfers the contents of key-on wakeup control register L1 to register A.						

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 0 1 0 0 1 <sub>2</sub>	2 4 9 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A <sub>3</sub> , A <sub>2</sub> ) ← (AD <sub>1</sub> , AD <sub>0</sub> ) (A <sub>1</sub> , A <sub>0</sub> ) ← 0						<b>Grouping:</b> A/D conversion operation
<b>Description:</b> Transfers the low-order 2 bits (AD <sub>1</sub> , AD <sub>0</sub> ) of register AD to the high-order 2 bits (A <sub>3</sub> , A <sub>2</sub> ) of register A. "0" is stored to the low-order 2 bits (A <sub>1</sub> , A <sub>0</sub> ) of register A.						

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TAM j (Transfer data to Accumulator from Memory)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 1 1 0 0 j j j j <sub>2</sub>	2 C j <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) $\leftarrow$ (M(DP)) (X) $\leftarrow$ (X)EXOR(j) j = 0 to 15						<b>Grouping:</b> RAM to register transfer
						<b>Description:</b> After transferring the contents of M(DP) to register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 1 0 0 1 0 <sub>2</sub>	2 5 2 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) $\leftarrow$ (MR)						<b>Grouping:</b> Clock operation
						<b>Description:</b> Transfers the contents of clock control register MR to register A.

**TAPU0 (Transfer data to Accumulator from register PU0)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 1 0 1 1 1 <sub>2</sub>	2 5 7 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) $\leftarrow$ (PU0)						<b>Grouping:</b> Input/Output operation
						<b>Description:</b> Transfers the contents of pull-up control register PU0 to register A.

**TAPU1 (Transfer data to Accumulator from register PU1)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 1 1 1 1 0 <sub>2</sub>	2 5 E <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) $\leftarrow$ (PU1)						<b>Grouping:</b> Input/Output operation
						<b>Description:</b> Transfers the contents of pull-up control register PU1 to register A.

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TAPU2 (Transfer data to Accumulator from register PU2)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 1 1 1 1 <sub>2</sub>	2 5 F <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) ← (PU2)						<b>Grouping:</b> Input/Output operation
<b>Description:</b> Transfers the contents of pull-up control register PU2 to register A.						

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

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

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0 0 0 1 0 1 0 0 0 0 <sub>2</sub>	0 5 0 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A <sub>2</sub> –A <sub>0</sub> ) ← (SP <sub>2</sub> –SP <sub>0</sub> ) (A <sub>3</sub> ) ← 0						<b>Grouping:</b> Register to register transfer
<b>Description:</b> Transfers the contents of stack pointer (SP) to the low-order 3 bits (A <sub>2</sub> –A <sub>0</sub> ) of register A. "0" is stored to the bit 3 (A <sub>3</sub> ) of register A.						

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	0 0 0 1 0 1 0 1 0 0 <sub>2</sub>	0 5 4 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) ← (V1)						<b>Grouping:</b> Interrupt operation
<b>Description:</b> Transfers the contents of interrupt control register V1 to register A.						

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TAV2 (Transfer data to Accumulator from register V2)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	0	0	0	1	0	1	0	1	0	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>5</td><td>5</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	5	5	16			1	1	-	-
0	0	0	1	0	1	0	1	0	1																							
2																																
0	5	5																														
16																																
<b>Operation:</b> (A) $\leftarrow$ (V2)						<b>Grouping:</b> Interrupt operation																										
<b>Description:</b> Transfers the contents of interrupt control register V2 to register A.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	1	0	0	1	0	1	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>4</td><td>B</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	4	B	16			1	1	-	-
1	0	0	1	0	0	1	0	1	1																							
2																																
2	4	B																														
16																																
<b>Operation:</b> (A) $\leftarrow$ (W1)						<b>Grouping:</b> Timer operation																										
<b>Description:</b> Transfers the contents of timer control register W1 to register A.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	1	0	0	1	1	0	0	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>4</td><td>C</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	4	C	16			1	1	-	-
1	0	0	1	0	0	1	1	0	0																							
2																																
2	4	C																														
16																																
<b>Operation:</b> (A) $\leftarrow$ (W2)						<b>Grouping:</b> Timer operation																										
<b>Description:</b> Transfers the contents of timer control register W2 to register A.																																

**TAW5 (Transfer data to Accumulator from register W5)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2</td></tr></table>	1	0	0	1	0	0	1	1	1	1	2										<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>4</td><td>F</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	4	F	16			1	1	-	-
1	0	0	1	0	0	1	1	1	1																							
2																																
2	4	F																														
16																																
<b>Operation:</b> (A) $\leftarrow$ (W5)						<b>Grouping:</b> Timer operation																										
<b>Description:</b> Transfers the contents of timer control register W5 to register A.																																

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TAW6 (Transfer data to Accumulator from register W6)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>5</td><td>0</td><td colspan="4"><sub>16</sub></td></tr></table>	1	0	0	1	0	1	0	0	0	0	<sub>2</sub>										2	5	0	<sub>16</sub>				1	1	-	-
1	0	0	1	0	1	0	0	0	0																							
<sub>2</sub>																																
2	5	0	<sub>16</sub>																													
<b>Operation:</b> (A) ← (W6)						<b>Grouping:</b> Timer operation																										
<b>Description:</b> Transfers the contents of timer control register W6 to register A.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>5</td><td>2</td><td colspan="4"><sub>16</sub></td></tr></table>	0	0	0	1	0	1	0	0	1	0	<sub>2</sub>										0	5	2	<sub>16</sub>				1	1	-	-
0	0	0	1	0	1	0	0	1	0																							
<sub>2</sub>																																
0	5	2	<sub>16</sub>																													
<b>Operation:</b> (A) ← (X)						<b>Grouping:</b> Register to register transfer																										
<b>Description:</b> Transfers the contents of register X to register A.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>1</td><td>F</td><td colspan="4"><sub>16</sub></td></tr></table>	0	0	0	0	0	1	1	1	1	1	<sub>2</sub>										0	1	F	<sub>16</sub>				1	1	-	-
0	0	0	0	0	1	1	1	1	1																							
<sub>2</sub>																																
0	1	F	<sub>16</sub>																													
<b>Operation:</b> (A) ← (Y)						<b>Grouping:</b> Register to register transfer																										
<b>Description:</b> Transfers the contents of register Y to register A.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>5</td><td>3</td><td colspan="4"><sub>16</sub></td></tr></table>	0	0	0	1	0	1	0	0	1	1	<sub>2</sub>										0	5	3	<sub>16</sub>				1	1	-	-
0	0	0	1	0	1	0	0	1	1																							
<sub>2</sub>																																
0	5	3	<sub>16</sub>																													
<b>Operation:</b> (A1, A0) ← (Z1, Z0) (A3, A2) ← 0						<b>Grouping:</b> Register to register transfer																										
<b>Description:</b> Transfers the contents of register Z to the low-order 2 bits (A1, A0) of register A. "0" is stored to the high-order 2 bits (A3, A2) of register A.																																

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TBA (Transfer data to register B from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																								
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td colspan="9" style="text-align: right;">2</td></tr></table>	0	0	0	0	0	1	1	1	0	2									<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>E</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	0	E	16			1	1	-	-
0	0	0	0	0	1	1	1	0																						
2																														
0	0	E																												
16																														
<b>Operation:</b> (B) $\leftarrow$ (A)						<b>Grouping:</b> Register to register transfer																								
<b>Description:</b> Transfers the contents of register A to register B.																														

**TC1A (Transfer data to register C1 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																								
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td colspan="9" style="text-align: right;">2</td></tr></table>	1	0	1	0	1	0	1	0	0	2									<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>A</td><td>8</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	2	A	8	16			1	1	-	-
1	0	1	0	1	0	1	0	0																						
2																														
2	A	8																												
16																														
<b>Operation:</b> (C1) $\leftarrow$ (A)						<b>Grouping:</b> Input/Output operation																								
<b>Description:</b> Transfers the contents of register A to port output structure control register C1.																														

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																								
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td colspan="9" style="text-align: right;">2</td></tr></table>	0	0	0	0	1	0	1	0	0	2									<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>2</td><td>9</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	2	9	16			1	1	-	-
0	0	0	0	1	0	1	0	0																						
2																														
0	2	9																												
16																														
<b>Operation:</b> (DR2-DR0) $\leftarrow$ (A2-A0)						<b>Grouping:</b> Register to register transfer																								
<b>Description:</b> Transfers the contents of the low-order 3 bits (A2-A0) of register A to register D.																														

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																								
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td colspan="9" style="text-align: right;">2</td></tr></table>	0	0	0	0	0	1	1	0	1	2									<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>A</td></tr><tr><td colspan="3" style="text-align: right;">16</td></tr></table>	0	1	A	16			1	1	-	-
0	0	0	0	0	1	1	0	1																						
2																														
0	1	A																												
16																														
<b>Operation:</b> (E7-E4) $\leftarrow$ (B) (E3-E0) $\leftarrow$ (A)						<b>Grouping:</b> Register to register transfer																								
<b>Description:</b> Transfers the contents of register B to the high-order 4 bits (E7-E4) of register E, and the contents of register A to the low-order 4 bits (E3-E0) of register E.																														

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TFR0A (Transfer data to register FR0 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 1 0 1 0 0 0 <sub>2</sub>	2 2 8 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (FR0) ← (A)						<b>Grouping:</b> Input/Output operation
<b>Description:</b> Transfers the contents of register A to port output structure control register FR0.						

**TFR1A (Transfer data to register FR1 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 1 0 1 0 0 1 <sub>2</sub>	2 2 9 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (FR1) ← (A)						<b>Grouping:</b> Input/Output operation
<b>Description:</b> Transfers the contents of register A to port output structure control register FR1.						

**TFR2A (Transfer data to register FR2 from Accumulator)**

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

**TFR3A (Transfer data to register FR3 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 1 0 1 0 1 1 <sub>2</sub>	2 2 B <sub>16</sub>	1	1	-	-
<b>Operation:</b> (FR3) ← (A)						<b>Grouping:</b> Input/Output operation
<b>Description:</b> Transfers the contents of register A to port output structure control register FR3.						

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TI1A (Transfer data to register I1 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>1</td><td>7</td><td colspan="4"><sub>16</sub></td></tr></table>	1	0	0	0	0	1	0	1	1	1	<sub>2</sub>										2	1	7	<sub>16</sub>				1	1	-	-
1	0	0	0	0	1	0	1	1	1																							
<sub>2</sub>																																
2	1	7	<sub>16</sub>																													
<b>Operation:</b> (I1) $\leftarrow$ (A)						<b>Grouping:</b> Interrupt operation																										
<b>Description:</b> Transfers the contents of register A to interrupt control register I1.																																

**TJ1A (Transfer data to register J1 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>0</td><td>2</td><td colspan="4"><sub>16</sub></td></tr></table>	1	0	0	0	0	0	0	0	1	0	<sub>2</sub>										2	0	2	<sub>16</sub>				1	1	-	-
1	0	0	0	0	0	0	0	1	0																							
<sub>2</sub>																																
2	0	2	<sub>16</sub>																													
<b>Operation:</b> (J1) $\leftarrow$ (A)						<b>Grouping:</b> Serial interface operation																										
<b>Description:</b> Transfers the contents of register A to serial interface control register J1.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>1</td><td>B</td><td colspan="4"><sub>16</sub></td></tr></table>	1	0	0	0	0	1	1	0	1	1	<sub>2</sub>										2	1	B	<sub>16</sub>				1	1	-	-
1	0	0	0	0	1	1	0	1	1																							
<sub>2</sub>																																
2	1	B	<sub>16</sub>																													
<b>Operation:</b> (K0) $\leftarrow$ (A)						<b>Grouping:</b> Input/Output operation																										
<b>Description:</b> Transfers the contents of register A to key-on wakeup control register K0.																																

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																										
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>1</td><td>4</td><td colspan="4"><sub>16</sub></td></tr></table>	1	0	0	0	0	1	0	1	0	0	<sub>2</sub>										2	1	4	<sub>16</sub>				1	1	-	-
1	0	0	0	0	1	0	1	0	0																							
<sub>2</sub>																																
2	1	4	<sub>16</sub>																													
<b>Operation:</b> (K1) $\leftarrow$ (A)						<b>Grouping:</b> Input/Output operation																										
<b>Description:</b> Transfers the contents of register A to key-on wakeup control register K1.																																

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TK2A (Transfer data to register K2 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																				
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td colspan="10" style="text-align: right;">2 1 5</td></tr></table> <sub>16</sub>	1	0	0	0	0	1	0	1	0	1	2 1 5											1	1	-	-
1	0	0	0	0	1	0	1	0	1																	
2 1 5																										
<b>Operation:</b> (K2) $\leftarrow$ (A)				<b>Grouping:</b>	Input/Output operation																					
<b>Description:</b> Transfers the contents of register A to key-on wakeup control register K2.																										

**TL1A (Transfer data to register L1 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																				
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2 0 A</td></tr></table> <sub>16</sub>	1	0	0	0	0	0	1	0	1	0	2 0 A											1	1	-	-
1	0	0	0	0	0	1	0	1	0																	
2 0 A																										
<b>Operation:</b> (L1) $\leftarrow$ (A)				<b>Grouping:</b>	Input/Output operation																					
<b>Description:</b> Transfers the contents of register A to key-on wakeup control register L1.																										

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																				
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>j</td><td>j</td><td>j</td><td>j</td></tr><tr><td colspan="10" style="text-align: right;">2 B j</td></tr></table> <sub>16</sub>	1	0	1	0	1	1	j	j	j	j	2 B j											1	1	-	-
1	0	1	0	1	1	j	j	j	j																	
2 B j																										
<b>Operation:</b> (M(DP)) $\leftarrow$ (A) (X) $\leftarrow$ (X)EXOR(j) j = 0 to 15				<b>Grouping:</b>	RAM to register transfer																					
<b>Description:</b> After transferring the contents of register A to M(DP), an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.																										

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																				
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td colspan="10" style="text-align: right;">2 1 6</td></tr></table> <sub>16</sub>	1	0	0	0	0	1	0	1	1	0	2 1 6											1	1	-	-
1	0	0	0	0	1	0	1	1	0																	
2 1 6																										
<b>Operation:</b> (MR) $\leftarrow$ (A)				<b>Grouping:</b>	Clock operation																					
<b>Description:</b> Transfers the contents of register A to clock control register MR.																										

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TPAA (Transfer data to register PA from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 1 0 1 0 1 0 1 0 <sub>2</sub>	2 A A <sub>16</sub>	1	1	-	-
<b>Operation:</b> (PA0) ← (A0)		<b>Grouping:</b> Timer operation <b>Description:</b> Transfers the least significant bit of register A to timer control register PA.				

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 1 1 0 1 0 1 <sub>2</sub>	2 3 5 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (RPS7–RPS4) ← (B) (TPS7–TPS4) ← (B) (RPS3–RPS0) ← (A) (TPS3–TPS0) ← (A)		<b>Grouping:</b> Timer operation <b>Description:</b> Transfers the contents of register B to the high-order 4 bits of prescaler and prescaler reload register RPS. Transfers the contents of register A to the low-order 4 bits of prescaler and prescaler reload register RPS.				

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 1 0 1 1 0 1 <sub>2</sub>	2 2 D <sub>16</sub>	1	1	-	-
<b>Operation:</b> (PU0) ← (A)		<b>Grouping:</b> Input/Output operation <b>Description:</b> Transfers the contents of register A to pull-up control register PU0.				

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

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

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TPU2A (Transfer data to register PU2 from Accumulator)**

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

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

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

**TRGA (Transfer data to register RG from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 0 0 1 0 0 1 2	2 0 9 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (RG0) ← (A0)				<b>Grouping:</b> Clock operation		
<b>Description:</b> Transfers the least significant bit (A0) of register A to clock control register RG.						

**TSIAB (Transfer data to register SI from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 1 1 1 0 0 0 2	2 3 8 <sub>16</sub>	1	1	-	-
<b>Operation:</b> (SI7-SI4) ← (B) (SI3-SI0) ← (A)				<b>Grouping:</b> Serial interface operation		
<b>Description:</b> Transfers the contents of register B to the high-order 4 bits of serial interface register SI, and transfers the contents of register A to the low-order 4 bits of serial interface register SI.						

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TV1A (Transfer data to register V1 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition														
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td><sub>2</sub></td></tr></table>	0	0	0	0	1	1	1	1	1	<sub>2</sub>	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>3</td><td>F</td><td><sub>16</sub></td></tr></table>	0	3	F	<sub>16</sub>	1	1	-	-
0	0	0	0	1	1	1	1	1	<sub>2</sub>											
0	3	F	<sub>16</sub>																	
<b>Operation:</b> (V1) $\leftarrow$ (A)						<b>Grouping:</b> Interrupt operation														
<b>Description:</b> Transfers the contents of register A to interrupt control register V1.																				

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition														
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td><sub>2</sub></td></tr></table>	0	0	0	0	1	1	1	1	0	<sub>2</sub>	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>3</td><td>E</td><td><sub>16</sub></td></tr></table>	0	3	E	<sub>16</sub>	1	1	-	-
0	0	0	0	1	1	1	1	0	<sub>2</sub>											
0	3	E	<sub>16</sub>																	
<b>Operation:</b> (V2) $\leftarrow$ (A)						<b>Grouping:</b> Interrupt operation														
<b>Description:</b> Transfers the contents of register A to interrupt control register V2.																				

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition															
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td><sub>2</sub></td></tr></table>	1	0	0	0	0	0	1	1	1	0	<sub>2</sub>	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>0</td><td>E</td><td><sub>16</sub></td></tr></table>	2	0	E	<sub>16</sub>	1	1	-	-
1	0	0	0	0	0	1	1	1	0	<sub>2</sub>											
2	0	E	<sub>16</sub>																		
<b>Operation:</b> (W1) $\leftarrow$ (A)						<b>Grouping:</b> Timer operation															
<b>Description:</b> Transfers the contents of register A to timer control register W1.																					

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition															
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td><sub>2</sub></td></tr></table>	1	0	0	0	0	0	1	1	1	1	<sub>2</sub>	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2</td><td>0</td><td>F</td><td><sub>16</sub></td></tr></table>	2	0	F	<sub>16</sub>	1	1	-	-
1	0	0	0	0	0	1	1	1	1	<sub>2</sub>											
2	0	F	<sub>16</sub>																		
<b>Operation:</b> (W2) $\leftarrow$ (A)						<b>Grouping:</b> Timer operation															
<b>Description:</b> Transfers the contents of register A to timer control register W2.																					

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****TW5A (Transfer data to register W5 from Accumulator)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																												
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>1</td><td>2</td><td colspan="6"><sub>16</sub></td></tr></table>	1	0	0	0	0	1	0	0	1	0	<sub>2</sub>										2	1	2	<sub>16</sub>						1	1	-	-
1	0	0	0	0	1	0	0	1	0																									
<sub>2</sub>																																		
2	1	2	<sub>16</sub>																															
<b>Operation:</b> (W5) ← (A)						<b>Grouping:</b> Timer operation																												
<b>Description:</b> Transfers the contents of register A to timer control register W5.																																		

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																												
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>1</td><td>3</td><td colspan="6"><sub>16</sub></td></tr></table>	1	0	0	0	0	1	0	0	1	1	<sub>2</sub>										2	1	3	<sub>16</sub>						1	1	-	-
1	0	0	0	0	1	0	0	1	1																									
<sub>2</sub>																																		
2	1	3	<sub>16</sub>																															
<b>Operation:</b> (W6) ← (A)						<b>Grouping:</b> Timer operation																												
<b>Description:</b> Transfers the contents of register A to timer control register W6.																																		

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

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																												
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>0</td><td>0</td><td>C</td><td colspan="6"><sub>16</sub></td></tr></table>	0	0	0	0	0	0	1	1	0	0	<sub>2</sub>										0	0	C	<sub>16</sub>						1	1	-	-
0	0	0	0	0	0	1	1	0	0																									
<sub>2</sub>																																		
0	0	C	<sub>16</sub>																															
<b>Operation:</b> (Y) ← (A)						<b>Grouping:</b> Register to register transfer																												
<b>Description:</b> Transfers the contents of register A to register Y.																																		

**WRST (Watchdog timer ReSeT)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition																												
	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="10"><sub>2</sub></td></tr><tr><td>2</td><td>A</td><td>0</td><td colspan="6"><sub>16</sub></td></tr></table>	1	0	1	0	1	0	0	0	0	0	<sub>2</sub>										2	A	0	<sub>16</sub>						1	1	-	(WDF1) = 1
1	0	1	0	1	0	0	0	0	0																									
<sub>2</sub>																																		
2	A	0	<sub>16</sub>																															
<b>Operation:</b> (WDF1) = 1 ? (WDF1) ← 0						<b>Grouping:</b> Other operation																												
<b>Description:</b> Clears (0) to the WDF1 flag and skips the next instruction when watchdog timer flag WDF1 is "1." When the WDF1 flag is "0," executes the next instruction. Also, stops the watchdog timer function when executing the WRST instruction immediately after the DWDT instruction.																																		

**MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)****XAM j (eXchange Accumulator and Memory data)**

Instruction code	D9	D0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 1 1 0 1 j j j j <sub>2</sub>	2 D j <sub>16</sub>	1	1	-	-
<b>Operation:</b> (A) $\leftrightarrow$ (M(DP)) (X) $\leftarrow$ (X)EXOR(j) j = 0 to 15						
<b>Grouping:</b> RAM to register transfer <b>Description:</b> After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.						

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

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

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

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

## MACHINE INSTRUCTIONS (INDEX BY TYPES)

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

Skip condition	Carry flag CY	Detailed description
–	–	Transfers the contents of register B to register A.
–	–	Transfers the contents of register A to register B.
–	–	Transfers the contents of register Y to register A.
–	–	Transfers the contents of register A to register Y.
–	–	Transfers the contents of register B to the high-order 4 bits (E7–E4) of register E, and the contents of register A to the low-order 4 bits (E3–E0) of register E.
–	–	Transfers the high-order 4 bits (E7–E4) of register E to register B, and low-order 4 bits of register E to register A.
–	–	Transfers the contents of the low-order 3 bits (A2–A0) of register A to register D.
–	–	Transfers the contents of register D to the low-order 3 bits (A2–A0) of register A. "0" is stored to the bit 3 (A3) of register A.
–	–	Transfers the contents of register Z to the low-order 2 bits (A1, A0) of register A. "0" is stored to the high-order 2 bits (A3, A2) of register A.
–	–	Transfers the contents of register X to register A.
–	–	Transfers the contents of stack pointer (SP) to the low-order 3 bits (A2–A0) of register A. "0" is stored to the bit 3 (A3) of register A.
Continuous description	–	Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y. When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped.
–	–	Loads the value z in the immediate field to register Z.
(Y) = 0	–	Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. When the contents of register Y is not 0, the next instruction is executed.
(Y) = 15	–	Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped. When the contents of register Y is not 15, the next instruction is executed.
–	–	After transferring the contents of M(DP) to register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
–	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
(Y) = 15	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped. When the contents of register Y is not 15, the next instruction is executed.
(Y) = 0	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. When the contents of register Y is not 0, the next instruction is executed.
–	–	After transferring the contents of register A to M(DP), an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.

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

Parameter Type of instructions	Mnemonic	Instruction code									Hexadecimal notation	Number of words	Number of cycles	Function	
		D9	D8	D7	D6	D5	D4	D3	D2	D1					
Arithmetic operation	LA n	0	0	0	1	1	1	n	n	n	n	0 7 n	1	1	(A) $\leftarrow$ n n = 0 to 15
	TABP p	0	0	1	0	0	p4	p3	p2	p1	p0	0 8 p +p	1	3	(SP) $\leftarrow$ (SP) + 1 (SK(SP)) $\leftarrow$ (PC) (PCH) $\leftarrow$ p (Note) (PCL) $\leftarrow$ (DR2-DR0, A3-A0) (B) $\leftarrow$ (ROM(PC))7-4 (A) $\leftarrow$ (ROM(PC))3-0 (UPTF) = 1 (DR1, DR0) $\leftarrow$ (ROM(PC))9, 8 (DR2) $\leftarrow$ 0 (PC) $\leftarrow$ (SK(SP)) (SP) $\leftarrow$ (SP) - 1
	AM	0	0	0	0	0	0	1	0	1	0	0 0 A	1	1	(A) $\leftarrow$ (A) + (M(DP))
	AMC	0	0	0	0	0	0	1	0	1	1	0 0 B	1	1	(A) $\leftarrow$ (A) + (M(DP)) + (CY) (CY) $\leftarrow$ Carry
	A n	0	0	0	1	1	0	n	n	n	n	0 6 n	1	1	(A) $\leftarrow$ (A) + n n = 0 to 15
	AND	0	0	0	0	0	1	1	0	0	0	0 1 8	1	1	(A) $\leftarrow$ (A) AND (M(DP))
	OR	0	0	0	0	0	1	1	0	0	1	0 1 9	1	1	(A) $\leftarrow$ (A) OR (M(DP))
	SC	0	0	0	0	0	0	0	1	1	1	0 0 7	1	1	(CY) $\leftarrow$ 1
	RC	0	0	0	0	0	0	0	1	1	0	0 0 6	1	1	(CY) $\leftarrow$ 0
	SZC	0	0	0	0	1	0	1	1	1	1	0 2 F	1	1	(CY) = 0 ?
Bit operation	CMA	0	0	0	0	0	1	1	1	0	0	0 1 C	1	1	(A) $\leftarrow$ ( $\bar{A}$ )
	RAR	0	0	0	0	0	1	1	1	0	1	0 1 D	1	1	$\boxed{CY \rightarrow A_3A_2A_1A_0}$
	SB j	0	0	0	1	0	1	1	1	j	j	0 5 C +j	1	1	(Mj(DP)) $\leftarrow$ 1 j = 0 to 3
Comparison operation	RB j	0	0	0	1	0	0	1	1	j	j	0 4 C +j	1	1	(Mj(DP)) $\leftarrow$ 0 j = 0 to 3
	SZB j	0	0	0	0	1	0	0	0	j	j	0 2 j	1	1	(Mj(DP)) = 0 ? j = 0 to 3
Comparison operation	SEAM	0	0	0	0	1	0	0	1	1	0	0 2 6	1	1	(A) = (M(DP)) ?
	SEA n	0	0	0	0	1	0	0	1	0	1	0 2 5	2	2	(A) = n ? n = 0 to 15
		0	0	0	1	1	1	n	n	n	n	0 7 n			

Note : p is 0 to 31.

Skip condition	Carry flag CY	Detailed description
Continuous description	–	<ul style="list-style-type: none"> <li>– 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.</li> </ul>
	–	<ul style="list-style-type: none"> <li>– Transfers bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 7 to 0 are the ROM pattern in address (DR2 DR1 DR0 A3 A2 A1 A0)<sub>2</sub> specified by registers A and D in page p. When UPTF is 1, Transfers bits 9, 8 to the low-order 2 bits (DR1, DR0) of register D, and "0" is stored to the least significant bit (DR2) of register D. When this instruction is executed, 1 stage of stack register (SK) is used.</li> </ul>
	–	<ul style="list-style-type: none"> <li>– Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged.</li> </ul>
	–	<ul style="list-style-type: none"> <li>– 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.</li> </ul>
Overflow = 0	–	<ul style="list-style-type: none"> <li>– 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.</li> </ul>
	–	<ul style="list-style-type: none"> <li>– Takes the AND operation between the contents of register A and the contents of M(DP), and stores the result in register A.</li> </ul>
	–	<ul style="list-style-type: none"> <li>– Takes the OR operation between the contents of register A and the contents of M(DP), and stores the result in register A.</li> </ul>
	–	<ul style="list-style-type: none"> <li>– 1 Sets (1) to carry flag CY.</li> </ul>
	–	<ul style="list-style-type: none"> <li>– 0 Clears (0) to carry flag CY.</li> </ul>
(CY) = 0	–	<ul style="list-style-type: none"> <li>– Skips the next instruction when the contents of carry flag CY is "0."</li> </ul>
	–	<ul style="list-style-type: none"> <li>– Stores the one's complement for register A's contents in register A.</li> </ul>
	–	<ul style="list-style-type: none"> <li>– 0/1 Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right.</li> </ul>
	–	<ul style="list-style-type: none"> <li>– Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).</li> </ul>
	–	<ul style="list-style-type: none"> <li>– Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).</li> </ul>
(M <sub>j</sub> (DP)) = 0 j = 0 to 3	–	<ul style="list-style-type: none"> <li>– 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."</li> </ul>
(A) = (M(DP))	–	<ul style="list-style-type: none"> <li>– 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).</li> </ul>
(A) = n n = 0 to 15	–	<ul style="list-style-type: none"> <li>– 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.</li> </ul>

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

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

Note : p is 0 to 31.

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

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

Parameter Type of instructions	Mnemonic	Instruction code									Hexadecimal notation	Number of words	Number of cycles	Function	
		D9	D8	D7	D6	D5	D4	D3	D2	D1					
Interrupt operation	DI	0	0	0	0	0	0	0	1	0	0	0 0 4	1	1	(INTE) ← 0
	EI	0	0	0	0	0	0	0	1	0	1	0 0 5	1	1	(INTE) ← 1
	SNZ0	0	0	0	0	1	1	1	0	0	0	0 3 8	1	1	V10 = 0: (EXFO) = 1 ? (EXFO) ← 0 V10 = 1: SNZ0 = NOP
	SNZI0	0	0	0	0	1	1	1	0	1	0	0 3 A	1	1	I12 = 0 : (INT) = "L" ? I12 = 1 : (INT) = "H" ?
	TAV1	0	0	0	1	0	1	0	1	0	0	0 5 4	1	1	(A) ← (V1)
	TV1A	0	0	0	0	1	1	1	1	1	1	0 3 F	1	1	(V1) ← (A)
	TAV2	0	0	0	1	0	1	0	1	0	1	0 5 5	1	1	(A) ← (V2)
	TV2A	0	0	0	0	1	1	1	1	1	0	0 3 E	1	1	(V2) ← (A)
	TAI1	1	0	0	1	0	1	0	0	1	1	2 5 3	1	1	(A) ← (I1)
	TI1A	1	0	0	0	0	1	0	1	1	1	2 1 7	1	1	(I1) ← (A)
Timer operation	TPAA	1	0	1	0	1	0	1	0	1	0	2 A A	1	1	(PA0) ← (A0)
	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	1	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)
	TAW5	1	0	0	1	0	0	1	1	1	1	2 4 F	1	1	(A) ← (W5)
	TW5A	1	0	0	0	0	1	0	0	1	0	2 1 2	1	1	(W5) ← (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)
	TABPS	1	0	0	1	1	1	0	1	0	1	2 7 5	1	1	(B) ← (TPS7–TPS4) (A) ← (TPS3–TPS0)
	TPSAB	1	0	0	0	1	1	0	1	0	1	2 3 5	1	1	(RPS7–RPS4) ← (B) (TPS7–TPS4) ← (B) (RPS3–RPS0) ← (A) (TPS3–TPS0) ← (A)
	TAB1	1	0	0	1	1	1	0	0	0	0	2 7 0	1	1	(B) ← (T17–T14) (A) ← (T13–T10)
	T1AB	1	0	0	0	1	1	0	0	0	0	2 3 0	1	1	(R1L7–R1L4) ← (B) (T17–T14) ← (B) (R1L3–R1L0) ← (A) (T13–T10) ← (A)
	T1HAB	1	0	1	0	0	1	0	0	1	0	2 9 2	1	1	(R1H7–R1H4) ← (B) (R1H3–R1H0) ← (A)

Skip condition	Carry flag CY	Detailed description
–	–	Clears (0) to interrupt enable flag INTE, and disables the interrupt.
–	–	Sets (1) to interrupt enable flag INTE, and enables the interrupt.
V10 = 0: (EXF0) = 1	–	When V10 = 0 : Clears (0) to the EXF0 flag and skips the next instruction when external 0 interrupt request flag EXF0 is “1.” When the EXF0 flag is “0,” executes the next instruction. When V10 = 1 : This instruction is equivalent to the NOP instruction. (V10: bit 0 of interrupt control register V1)
(INT) = “L” However, I12 = 0	–	When I12 = 0 : Skips the next instruction when the level of INT pin is “L.” Executes the next instruction when the level of INT pin is “H.”
(INT) = “H” However, I12 = 1	–	When I12 = 1 : Skips the next instruction when the level of INT pin is “H.” Executes the next instruction when the level of INT pin is “L.” (I12: bit 2 of interrupt control register I1)
–	–	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 register A to timer control register PA.
–	–	Transfers the contents of timer control register W1 to register A.
–	–	Transfers the contents of register A to timer control register W1.
–	–	Transfers the contents of timer control register W2 to register A.
–	–	Transfers the contents of register A to timer control register W2.
–	–	Transfers the contents of timer control register W5 to register A.
–	–	Transfers the contents of register A to timer control register W5.
–	–	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 of prescaler to register B. Transfers the low-order 4 bits of prescaler to register A.
–	–	Transfers the contents of register B to the high-order 4 bits of prescaler and prescaler reload register RPS. Transfers the contents of register A to the low-order 4 bits of prescaler and prescaler reload register RPS.
–	–	Transfers the high-order 4 bits (T17–T14) of timer 1 to register B. Transfers the low-order 4 bits (T13–T10) 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 R1L. Transfers the contents of register A to the low-order 4 bits of timer 1 and timer 1 reload register R1L.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 1 reload register R1H. Transfers the contents of register A to the low-order 4 bits of timer 1 reload register R1H.

Parameter Type of instructions	Mnemonic	Instruction code									Number of words	Number of cycles	Function	
		D9	D8	D7	D6	D5	D4	D3	D2	D1	D0			
Timer operation	TAB2	1	0	0	1	1	1	0	0	0	1	2 7 1	1	1 (B) ← (T27–T24) (A) ← (T23–T20)
	T2AB	1	0	0	0	1	1	0	0	0	1	2 3 1	1	1 (R2L7–R2L4) ← (B) (T27–T24) ← (B) (R2L3–R2L0) ← (A) (T23–T20) ← (A)
	T2HAB	1	0	1	0	0	1	0	1	0	0	2 9 4	1	1 (R2H7–R2H4) ← (B) (R2H3–R2H0) ← (A)
	T1R1L	1	0	1	0	1	0	0	1	1	1	2 A 7	1	1 (T1) ← (R1L)
	T2R2L	1	0	1	0	0	1	0	1	0	1	2 9 5	1	1 (T2) ← (R2L)
	SNZT1	1	0	1	0	0	0	0	0	0	0	2 8 0	1	1 V12 = 0: (T1F) = 1 ? (T1F) ← 0 V12 = 1: SNZT1 = NOP
	SNZT2	1	0	1	0	0	0	0	0	0	1	2 8 1	1	1 V13 = 0: (T2F) = 1 ? (T2F) ← 0 V13 = 1: SNZT2 = NOP
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 (A1, A0) ← (P21, P20) (A3, A2) ← 0
	OP2A	1	0	0	0	1	0	0	0	1	0	2 2 2	1	1 (P21, P20) ← (A1, A0)
	IAP3	1	0	0	1	1	0	0	0	1	1	2 6 3	1	1 (A1, A0) ← (P31, P30) (A3, A2) ← 0
	OP3A	1	0	0	0	1	0	0	0	1	1	2 2 3	1	1 (P31, P30) ← (A1, A0)
	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 5
	SD	0	0	0	0	0	1	0	1	0	1	0 1 5	1	1 (D(Y)) ← 1 (Y) = 0 to 5
	SZD	0	0	0	0	1	0	0	1	0	0	0 2 4	2	2 (D(Y)) = 0 ? (Y) = 0 to 5
		0	0	0	0	1	0	1	0	1	1	0 2 B		

Skip condition	Carry flag CY	Detailed description
–	–	Transfers the high-order 4 bits (T27–T24) of timer 2 to register B. Transfers the low-order 4 bits (T23–T20) of timer 2 to register A.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 2 and timer 2 reload register R2L. Transfers the contents of register A to the low-order 4 bits of timer 2 and timer 2 reload register R2L.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 2 reload register R2H. Transfers the contents of register A to the low-order 4 bits of timer 2 reload register R2H.
–	–	Transfers the contents of timer 1 reload register R1L to timer 1.
–	–	Transfers the contents of timer 2 reload register R2L to timer 2.
V12 = 0: (T1F) = 1	–	When V12 = 0 : Clears (0) to the T1F flag and skips the next instruction when timer 1 interrupt request flag T1F is “1.” When the T1F flag is “0,” executes the next instruction. When V12 = 1 : This instruction is equivalent to the NOP instruction. (V12: bit 2 of interrupt control register V1)
V13 = 0: (T2F) = 1	–	When V13 = 0 : Clears (0) to the T2F flag and skips the next instruction when timer 2 interrupt request flag T2F is “1.” When the T2F flag is “0,” executes the next instruction. When V13 = 1 : This instruction is equivalent to the NOP instruction. (V13: bit 3 of interrupt control register V1)
–	–	Transfers the input of port P0 to register A.
–	–	Outputs the contents of register A to port P0.
–	–	Transfers the input of port P1 to register A.
–	–	Outputs the contents of register A to port P1.
–	–	Transfers the input of port P2 to the low-order 2 bits (A1, A0) of register A. “0” is stored to the bit 3 (A3) of register A.
–	–	Outputs the contents of the low-order 2 bits (A1, A0) of register A to port P2.
–	–	Transfers the input of port P3 to the low-order 2 bits (A1, A0) of register A. “0” is stored to the bit 3 (A3) of register A.
–	–	Outputs the contents of the low-order 2 bits (A1, A0) of register A to port P3.
–	–	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 ?	–	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.”

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

Parameter Type of instructions	Mnemonic	Instruction code										Hexadecimal notation	Number of words	Number of cycles	Function
		D9	D8	D7	D6	D5	D4	D3	D2	D1	D0				
Input/Output operation	TFR0A	1	0	0	0	1	0	1	0	0	0	2 2 8	1	1	(FR0) ← (A)
	TFR1A	1	0	0	0	1	0	1	0	0	1	2 2 9	1	1	(FR1) ← (A)
	TFR2A	1	0	0	0	1	0	1	0	1	0	2 2 A	1	1	(FR2) ← (A)
	TFR3A	1	0	0	0	1	0	1	0	1	1	2 2 B	1	1	(FR3) ← (A)
	TC1A	1	0	1	0	1	0	1	0	0	0	2 A 8	1	1	(C1) ← (A)
	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)
	TAPU0	1	0	0	1	0	1	0	1	1	1	2 5 7	1	1	(A) ← (PU0)
	TPU1A	1	0	0	0	1	0	1	1	1	0	2 2 E	1	1	(PU1) ← (A)
	TAPU1	1	0	0	1	0	1	1	1	1	0	2 5 E	1	1	(A) ← (PU1)
	TPU2A	1	0	0	0	1	0	1	1	1	1	2 2 F	1	1	(PU2) ← (A)
	TAPU2	1	0	0	1	0	1	1	1	1	1	2 5 F	1	1	(A) ← (PU2)
Serial interface operation	TL1A	1	0	0	0	0	0	1	0	1	0	2 0 A	1	1	(L1) ← (A)
	TAL1	1	0	0	1	0	0	1	0	1	0	2 4 A	1	1	(A) ← (L1)
	TABSI	1	0	0	1	1	1	1	0	0	0	2 7 8	1	1	(B) ← (SI7-SI4) (A) ← (SI3-SI0)
	TSIAB	1	0	0	0	1	1	1	0	0	0	2 3 8	1	1	(SI7-SI4) ← (B) (SI3-SI0) ← (A)
	SST	1	0	1	0	0	1	1	1	1	0	2 9 E	1	1	(SIOF) ← 0 Serial interface transmit/receive starting
	SNZSI	1	0	1	0	0	0	1	0	0	0	2 8 8	1	1	V23=0: (SIOF)=1? (SIOF) ← 0 V23 = 1: SNZSI = NOP
Clock operation	TAJ1	1	0	0	1	0	0	0	0	1	0	2 4 2	1	1	(A) ← (J1)
	TJ1A	1	0	0	0	0	0	0	0	1	0	2 0 2	1	1	(J1) ← (A)
	CRCK	1	0	1	0	0	1	1	0	1	1	2 9 B	1	1	RC oscillator selected
	TRGA	1	0	0	0	0	0	1	0	0	1	2 0 9	1	1	(RG0) ← (A0)
Clock operation	TAMR	1	0	0	1	0	1	0	0	1	0	2 5 2	1	1	(A) ← (MR)
	TMRA	1	0	0	0	0	1	0	1	1	0	2 1 6	1	1	(MR) ← (A)

Skip condition	Carry flag CY	Detailed description
–	–	<ul style="list-style-type: none"> <li>Transfers the contents of register A to port output structure control register FR0.</li> <li>Transfers the contents of register A to port output structure control register FR1.</li> <li>Transfers the contents of register A to port output structure control register FR2.</li> <li>Transfers the contents of register A to port output structure control register FR3.</li> <li>Transfers the contents of register A to port output structure control register C1.</li> <li>Transfers the contents of register A to key-on wakeup control register K0.</li> <li>Transfers the contents of key-on wakeup control register K0 to register A.</li> <li>Transfers the contents of register A to key-on wakeup control register K1.</li> <li>Transfers the contents of key-on wakeup control register K1 to register A.</li> <li>Transfers the contents of register A to key-on wakeup control register K2.</li> <li>Transfers the contents of key-on wakeup control register K2 to register A.</li> <li>Transfers the contents of register A to pull-up control register PU0.</li> <li>Transfers the contents of pull-up control register PU0 to register A.</li> <li>Transfers the contents of register A to pull-up control register PU1.</li> <li>Transfers the contents of pull-up control register PU1 to register A.</li> <li>Transfers the contents of register A to pull-up control register PU2.</li> <li>Transfers the contents of pull-up control register PU2 to register A.</li> <li>Transfers the contents of register A to key-on wakeup control register L1.</li> <li>Transfers the contents of key-on wakeup control register L1 to register A.</li> </ul>
V23 = 0: (SIOF) =1	–	<ul style="list-style-type: none"> <li>Transfers the high-order 4 bits of serial interface register SI to register B, and transfers the low-order 4 bits of serial interface register SI to register A.</li> <li>Transfers the contents of register B to the high-order 4 bits of serial interface register SI, and transfers the contents of register A to the low-order 4 bits of serial interface register SI.</li> <li>Clears (0) to SIOF flag and starts serial interface transmit/receive.</li> <li>Clears (0) to SIOF flag and skips the next instruction when the contents of bit 3 (V23) of interrupt control register V2 is "0" and contents of SIOF flag is "1." When V23 = 1: This instruction is equivalent to the NOP instruction.</li> <li>Transfers the contents of serial interface control register J1 to register A.</li> <li>Transfers the contents of register A to serial interface control register J1.</li> </ul>
–	–	<ul style="list-style-type: none"> <li>Selects the RC oscillation circuit for main clock f(XIN).</li> <li>Transfers the least significant bit (A0) of register A to clock control register RG.</li> <li>Transfers the contents of clock control register MR to register A.</li> <li>Transfers the contents of register A to clock control register MR.</li> </ul>

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

Parameter Type of instructions	Mnemonic	Instruction code									Hexadecimal notation	Number of words	Number of cycles	Function	
		D9	D8	D7	D6	D5	D4	D3	D2	D1					
A/D conversion operation	TABAD	1	0	0	1	1	1	1	0	0	1	2 7 9	1	1	Q13 = 0: (B) $\leftarrow$ (AD9–AD6) (A) $\leftarrow$ (AD5–AD2) Q13 = 1: (B) $\leftarrow$ (AD7–AD4) (A) $\leftarrow$ (AD3–AD0)
	TALA	1	0	0	1	0	0	1	0	0	1	2 4 9	1	1	(A3, A2) $\leftarrow$ (AD1, AD0) (A1, A0) $\leftarrow$ 0
	TADAB	1	0	0	0	1	1	1	0	0	1	2 3 9	1	1	Q13 = 0: (AD7–AD4) $\leftarrow$ (B) (AD3–AD0) $\leftarrow$ (A) Q13 = 1: TADAB = NOP
	TAQ1	1	0	0	1	0	0	0	1	0	0	2 4 4	1	1	(A) $\leftarrow$ (Q1)
	TQ1A	1	0	0	0	0	0	0	1	0	0	2 0 4	1	1	(Q1) $\leftarrow$ (A)
	ADST	1	0	1	0	0	1	1	1	1	1	2 9 F	1	1	(ADF) $\leftarrow$ 0 Q13 = 0: A/D conversion starting Q13 = 1: Comparator operation starting
	SNZAD	1	0	1	0	0	0	0	1	1	1	2 8 7	1	1	V22 = 0: (ADF) = 1 ? (ADF) $\leftarrow$ 0 V22 = 1: SNZAD = NOP
Other operation	NOP	0	0	0	0	0	0	0	0	0	0	0 0 0	1	1	(PC) $\leftarrow$ (PC) + 1
	POF	0	0	0	0	0	0	0	0	1	0	0 0 2	1	1	RAM back-up
	EPOF	0	0	0	1	0	1	1	0	1	1	0 5 B	1	1	POF 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 ?, (WDF1) $\leftarrow$ 0
	SRST	0	0	0	0	0	0	0	0	0	1	0 0 1	1	1	System reset
	RUPT	0	0	0	1	0	1	1	0	0	0	0 5 8	1	1	(UPTF) $\leftarrow$ 0
	SUPT	0	0	0	1	0	1	1	0	0	1	0 5 9	1	1	(UPTF) $\leftarrow$ 1
	SVDE**	1	0	1	0	0	1	0	0	1	1	2 9 3	1	1	Voltage drop detection circuit valid at RAM back-up

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

Skip condition	Carry flag CY	Detailed description
		<ul style="list-style-type: none"> <li>- In the A/D conversion mode (Q13 = 0), transfers the high-order 4 bits (AD9–AD6) of register AD to register B, and the middle-order 4 bits (AD5–AD2) of register AD to register A. In the comparator mode (Q13 = 1), transfers the high-order 4 bits (AD7–AD4) of comparator register to register B, and the low-order 4 bits (AD3–AD0) of comparator register to register A. (Q13: bit 3 of A/D control register Q1)</li> <li>- Transfers the low-order 2 bits (AD1, AD0) of register AD to the high-order 2 bits (A3, A2) of register A. "0" is stored to the least significant bit (A0) of register A.</li> <li>- In the comparator mode (Q13 = 1), transfers the contents of register B to the high-order 4 bits (AD7–AD4) of comparator register, and the contents of register A to the low-order 4 bits (AD3–AD0) of comparator register. In the A/D conversion mode (Q13 = 0), this instruction is equivalent to the NOP instruction. (Q13 = bit 3 of A/D control register Q1)</li> <li>- Transfers the contents of A/D control register Q1 to register A.</li> <li>- Transfers the contents of register A to A/D control register Q1.</li> <li>- Clears (0) to A/D conversion completion flag ADF, and the A/D conversion at the A/D conversion mode (Q13 = 0) or the comparator operation at the comparator mode (Q13 = 1) is started. (Q13 = bit 3 of A/D control register Q1)</li> </ul>
V22 = 0: (ADF) = 1		<ul style="list-style-type: none"> <li>- When V22 = 0 : Clears (0) to the ADF flag and skips the next instruction when A/D conversion completion flag ADF is "1." When the ADF flag is "0," executes the next instruction. When V22 = 1 : This instruction is equivalent to the NOP instruction. (V22: bit 2 of interrupt control register V2)</li> </ul>

## INSTRUCTION CODE TABLE

		D9-D4																010000 011000			
		010111 011111																010000 011000			
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	SRST	CLD	SZB 1	—	—	TAD	A 1	LA 1	TABP 1	TABP 17	—	—	BML	BML	BL	BL	BM	B		
0010	2	POF	—	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	—	AND	—	SNZ0	LZ 0	RUPT	A 8	LA 8	TABP 8	TABP 24	—	—	BML	BML	BL	BL	BM	B		
1001	9	—	OR	TDA	—	LZ 1	SUPT	A 9	LA 9	TABP 9	TABP 25	—	—	BML	BML	BL	BL	BM	B		
1010	A	AM	TEAB	TABE	SNZ10	LZ 2	—	A 10	LA 10	TABP 10	TABP 26	—	—	BML	BML	BL	BL	BM	B		
1011	B	AMC	—	—	—	LZ 3	EPOF	A 11	LA 11	TABP 11	TABP 27	—	—	BML	BML	BL	BL	BM	B		
1100	C	TYA	CMA	—	—	RB 0	SB 0	A 12	LA 12	TABP 12	TABP 28	—	—	BML	BML	BL	BL	BM	B		
1101	D	—	RAR	—	—	RB 1	SB 1	A 13	LA 13	TABP 13	TABP 29	—	—	BML	BML	BL	BL	BM	B		
1110	E	TBA	TAB	—	TV2A	RB 2	SB 2	A 14	LA 14	TABP 14	TABP 30	—	—	BML	BML	BL	BL	BM	B		
1111	F	—	TAY	SZC	TV1A	RB 3	SB 3	A 15	LA 15	TABP 15	TABP 31	—	—	BML	BML	BL	BL	BM	B		

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

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

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

## 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	TMA0	TAM0	XAM0	XAMI0	XAMD0	LXY
0001	1	—	—	OP1A	T2AB	—	—	IAP1	TAB2	SNZT2	—	—	TMA1	TAM1	XAM1	XAMI1	XAMD1	LXY
0010	2	TJ1A	TW5A	OP2A	—	TAJ1	TAMR	IAP2	—	—	T1HAB	—	TMA2	TAM2	XAM2	XAMI2	XAMD2	LXY
0011	3	—	TW6A	OP3A	—	—	TAI1	IAP3	—	—	SVDE*	—	TMA3	TAM3	XAM3	XAMI3	XAMD3	LXY
0100	4	TQ1A	TK1A	—	—	TAQ1	—	—	—	T2HAB	—	TMA4	TAM4	XAM4	XAMI4	XAMD4	LXY	
0101	5	—	TK2A	—	TPSAB	—	—	—	TABPS	—	T2R2L	—	TMA5	TAM5	XAM5	XAMI5	XAMD5	LXY
0110	6	—	TMRA	—	—	—	TAK0	—	—	—	—	—	TMA6	TAM6	XAM6	XAMI6	XAMD6	LXY
0111	7	—	TI1A	—	—	—	TAPU0	—	—	SNZAD	—	T1R1L	TMA7	TAM7	XAM7	XAMI7	XAMD7	LXY
1000	8	—	—	TFR0ATSIAB	—	—	—	TABSI	SNZSI	—	TC1A	TMA8	TAM8	XAM8	XAMI8	XAMD8	LXY	
1001	9	TRGA	—	TFR1ATADAB	TALA	TAK1	—	TABAD	—	—	—	TMA9	TAM9	XAM9	XAMI9	XAMD9	LXY	
1010	A	TL1A	—	TFR2A	—	TAL1	TAK2	—	—	—	TPAA	TMA10	TAM10	XAM10	XAMI10	XAMD10	LXY	
1011	B	—	TK0A	TFR3A	—	TAW1	—	—	—	CRCK	—	TMA11	TAM11	XAM11	XAMI11	XAMD11	LXY	
1100	C	—	—	—	—	TAW2	—	—	—	DWDT	—	TMA12	TAM12	XAM12	XAMI12	XAMD12	LXY	
1101	D	—	—	TPU0A	—	—	—	—	—	—	—	TMA13	TAM13	XAM13	XAMI13	XAMD13	LXY	
1110	E	TW1A	—	TPU1A	—	—	TAPU1	—	—	—	SST	—	TMA14	TAM14	XAM14	XAMI14	XAMD14	LXY
1111	F	TW2A	—	TPU2A	—	TAW5	TAPU2	—	—	—	ADST	—	TMA15	TAM15	XAM15	XAMI15	XAMD15	LXY

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

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

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

\* can be used only in the H version.

## Electrical characteristics

### Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
VDD	Supply voltage	–	–0.3 to 6.5	V
VI	Input voltage P0, P1, P2, P3, D0–D5, RESET, XIN	–	–0.3 to VDD+0.3	V
VI	Input voltage INT, CNTR0, CNTR1, SIN, SCK	–	–0.3 to VDD+0.3	V
VI	Input voltage AIN0–AIN5	–	–0.3 to VDD+0.3	V
VO	Output voltage P0, P1, P2, P3, D0–D5, RESET	Output transistors in cut-off state	–0.3 to VDD+0.3	V
VO	Output voltage CNTR0, CNTR1, SOUT, SCK	Output transistors in cut-off state	–0.3 to VDD+0.3	V
VO	Output voltage XOUT	–	–0.3 to VDD+0.3	V
Pd	Power dissipation	Ta = 25 °C	300	mW
Topr	Operating temperature range	–	–20 to 85	°C
Tstg	Storage temperature range	–	–40 to 125	°C

## Recommended operating conditions 1

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

Symbol	Parameter	Conditions	Limits			Unit
			Min.	Typ.	Max.	
VDD	Supply voltage (with a ceramic resonator)	f(STCK) ≤ 6 MHz	4		5.5	V
		f(STCK) ≤ 4.4 MHz	2.7		5.5	
		f(STCK) ≤ 2.2 MHz	2.0		5.5	
		f(STCK) ≤ 1.1 MHz	1.8		5.5	
VDD	Supply voltage (with RC oscillation)	f(STCK) ≤ 4.4 MHz	2.7		5.5	V
VDD	Supply voltage (with an on-chip oscillator)		1.8		5.5	V
VRAM	RAM back-up voltage	(at RAM back-up)	1.6		5.5	V
VSS	Supply voltage			0		V
VIH	"H" level input voltage	P0, P1, P2, P3, D0–D5	0.8VDD		VDD	V
		XIN	0.7VDD		VDD	
		RESET	0.85VDD		VDD	
		INT, CNTR0, CNTR1, SIN, SCK	0.85VDD		VDD	
VIL	"L" level input voltage	P0, P1, P2, P3, D0–D5	0		0.2VDD	V
		XIN	0		0.3VDD	
		RESET	0		0.3VDD	
		INT, CNTR0, CNTR1, SIN, SCK	0		0.15VDD	
IOH(peak)	"H" level peak output current	P0, P1, P2, P3, D0–D5	VDD = 5.0 V		-20	mA
		CNTR0, CNTR1, SOUT, SCK	VDD = 3.0 V		-10	
IOH(avg)	"H" level average output current (Note)	P0, P1, P2, P3, D0–D5	VDD = 5.0 V		-10	mA
		CNTR0, CNTR1, SOUT, SCK	VDD = 3.0 V		-5	
IOL(peak)	"L" level peak output current	P0, P1	VDD = 5.0 V		24	mA
		CNTR0, CNTR1, SOUT, SCK	VDD = 3.0 V		12	
		P2, P3, RESET	VDD = 5.0 V		10	
			VDD = 3.0 V		4.0	
		D0, D1, D4, D5	VDD = 5.0 V		40	
			VDD = 3.0 V		30	
		D2, D3	VDD = 5.0 V		24	
			VDD = 3.0 V		12	
IOL(avg)	"L" level average output current	P0, P1	VDD = 5.0 V		12	mA
		CNTR0, CNTR1, SOUT, SCK	VDD = 3.0 V		6.0	
		P2, P3, RESET	VDD = 5.0 V		5.0	
			VDD = 3.0 V		2.0	
		D0, D1, D4, D5	VDD = 5.0 V		30	
			VDD = 3.0 V		15	
		D2, D3	VDD = 5.0 V		15	
			VDD = 3.0 V		7.0	
ΣIOH(avg)	"H" level total average current	P0, P1, P3, CNTR0, CNTR1, SOUT, SCK			-40	mA
		P2, D0–D5			-40	
ΣIOL(avg)	"L" level total average current	P0, P1, P3, CNTR0, CNTR1, SOUT, SCK			60	mA
		P2, D0–D5, RESET			60	

Notes 1: The average output current (IOH, IOL) is the average value during 100 ms.

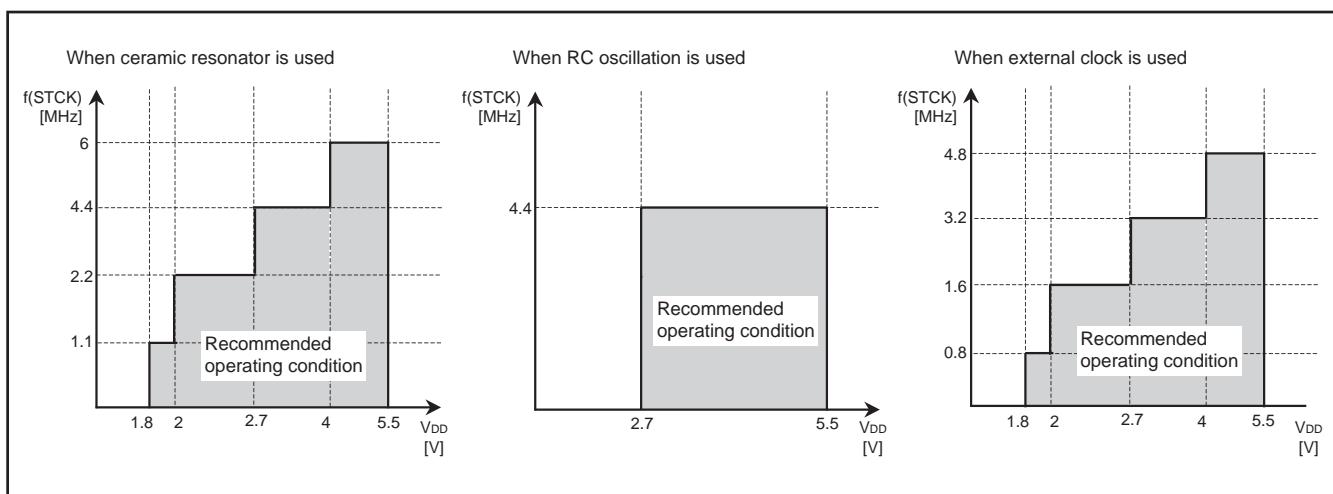
## Recommended operating conditions 2

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

Symbol	Parameter	Conditions	Limits			Unit
			Min.	Typ.	Max.	
f(XIN)	Oscillation frequency (with a ceramic resonator)	Through mode	VDD = 4.0 V to 5.5 V		6	MHz
			VDD = 2.7 V to 5.5 V		4.4	
			VDD = 2.0 V to 5.5 V		2.2	
			VDD = 1.8 V to 5.5 V		1.1	
		Internal frequency divided by 2	VDD = 2.7 V to 5.5 V		6	
			VDD = 2.0 V to 5.5 V		4.4	
			VDD = 1.8 V to 5.5 V		2.2	
		Internal frequency divided by 4, 8	VDD = 2.0 V to 5.5 V		6	
			VDD = 1.8 V to 5.5 V		4.4	
f(XIN)	Oscillation frequency (with RC oscillation) (Note 1)	VDD = 2.7 V to 5.5 V			4.4	MHz
f(XIN)	Oscillation frequency (with a ceramic oscillation selected, external clock input)	Through mode	VDD = 4.0 V to 5.5 V		4.8	MHz
			VDD = 2.7 V to 5.5 V		3.2	
			VDD = 2.0 V to 5.5 V		1.6	
			VDD = 1.8 V to 5.5 V		0.8	
		Internal frequency divided by 2	VDD = 2.7 V to 5.5 V		4.8	
			VDD = 2.0 V to 5.5 V		3.2	
			VDD = 1.8 V to 5.5 V		1.6	
		Internal frequency divided by 4, 8	VDD = 2.0 V to 5.5 V		4.8	
			VDD = 1.8 V to 5.5 V		3.2	
f(CNTR)	Timer external input frequency	CNTR0, CNTR1			f(STCK)/6	Hz
tw(CNTR)	Timer external input period ("H" and "L" pulse width)	CNTR0, CNTR1	3/f(STCK)			s
f(Sck)	Serial interface external input frequency	Sck			f(STCK)/6	Hz
tw(Sck)	Serial interface external input period ("H" and "L" pulse width)	Sck	3/f(STCK)			s
TPON	Power-on reset circuit valid supply voltage rising time (Note 2)	VDD = 0 → 1.8 V			100	μs

Notes 1: The frequency at RC oscillation is affected by a capacitor, a resistor and a microcomputer. So, set the constants within the range of the frequency limits.

2: If the rising time exceeds the maximum rating value, connect a capacitor between the RESET pin and Vss at the shortest distance, and input "L" level to RESET pin until the value of supply voltage reaches the minimum operating voltage.



System clock (STCK) operating condition map

**Electrical characteristics 1** (Ta = -20 °C to 85 °C, V<sub>DD</sub> = 1.8 to 5.5 V, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit	
			Min.	Typ.	Max.		
V <sub>OH</sub>	"H" level output voltage P0, P1, P2, P3, D <sub>0</sub> –D <sub>5</sub> CNTR0, CNTR1, S <sub>OUT</sub> , S <sub>Ck</sub>	V <sub>DD</sub> = 5.0 V	I <sub>OH</sub> = -10 mA	3.0		V	
			I <sub>OH</sub> = -3.0 mA	4.1			
		V <sub>DD</sub> = 3.0 V	I <sub>OH</sub> = -5.0 mA	2.1			
			I <sub>OH</sub> = -1.0 mA	2.4			
V <sub>OL</sub>	"L" level output voltage P0, P1 CNTR0, CNTR1, S <sub>OUT</sub> , S <sub>Ck</sub>	V <sub>DD</sub> = 5.0 V	I <sub>OL</sub> = 12 mA		2.0	V	
			I <sub>OL</sub> = 4.0 mA		0.9		
		V <sub>DD</sub> = 3.0 V	I <sub>OL</sub> = 6.0 mA		0.9		
			I <sub>OL</sub> = 2.0 mA		0.6		
V <sub>OL</sub>	"L" level output voltage P2, P3, <u>RESET</u>	V <sub>DD</sub> = 5.0 V	I <sub>OL</sub> = 5.0 mA		2.0	V	
			I <sub>OL</sub> = 1.0 mA		0.6		
		V <sub>DD</sub> = 3.0 V	I <sub>OL</sub> = 2.0 mA		0.9		
V <sub>OL</sub>	"L" level output voltage D <sub>0</sub> , D <sub>1</sub> , D <sub>4</sub> , D <sub>5</sub>	V <sub>DD</sub> = 5.0 V	I <sub>OL</sub> = 30 mA		2.0	V	
			I <sub>OL</sub> = 10 mA		0.9		
		V <sub>DD</sub> = 3.0 V	I <sub>OL</sub> = 15 mA		2.0		
			I <sub>OL</sub> = 5.0 mA		0.9		
V <sub>OL</sub>	"L" level output voltage D <sub>2</sub> , D <sub>3</sub>	V <sub>DD</sub> = 5.0 V	I <sub>OL</sub> = 15 mA		2.0	V	
			I <sub>OL</sub> = 5.0 mA		0.9		
		V <sub>DD</sub> = 3.0 V	I <sub>OL</sub> = 9.0 mA		1.4		
			I <sub>OL</sub> = 3.0 mA		0.9		
I <sub>IH</sub>	"H" level input current P0, P1, P2, P3, D <sub>0</sub> –D <sub>5</sub> <u>RESET</u> , INT CNTR0, CNTR1, S <sub>IN</sub> , S <sub>Ck</sub>	V <sub>I</sub> = V <sub>DD</sub>			2.0	μA	
I <sub>IL</sub>	"L" level input current P0, P1, P2, P3, D <sub>0</sub> –D <sub>5</sub> <u>RESET</u> , INT CNTR0, CNTR1, S <sub>IN</sub> , S <sub>Ck</sub>	V <sub>I</sub> = 0 V P0, P1, P2, D <sub>2</sub> , D <sub>3</sub> No pull-up			-2.0	μA	
R <sub>PU</sub>	Pull-up resistor value P0, P1, P2, D <sub>2</sub> , D <sub>3</sub> , <u>RESET</u>	V <sub>I</sub> = 0 V	V <sub>DD</sub> = 5.0 V	30	60	125	kΩ
			V <sub>DD</sub> = 3.0 V	50	120	250	
V <sub>T+</sub> – V <sub>T-</sub>	Hysteresis <u>RESET</u>	V <sub>DD</sub> = 5.0 V			1.0		V
			V <sub>DD</sub> = 3.0 V		0.4		
V <sub>T+</sub> – V <sub>T-</sub>	Hysteresis INT, CNTR0, CNTR1 S <sub>IN</sub> , S <sub>Ck</sub>	V <sub>DD</sub> = 5.0 V			0.2		V
			V <sub>DD</sub> = 3.0 V		0.2		
f(RING)	On-chip oscillator clock frequency	V <sub>DD</sub> = 5.0 V		200	500	700	kHz
			V <sub>DD</sub> = 3.0 V	100	250	400	
			V <sub>DD</sub> = 1.8 V	30	120	200	
Δ f(XIN)	Oscillation frequency error (Note 1) (at RC oscillation, error value of external R, C not included)	V <sub>DD</sub> = 5.0 V ± 10 %, Ta = center 25 °C			±17	%	
			V <sub>DD</sub> = 3.0 V ± 10 %, Ta = center 25 °C		±17		

Notes 1: When the RC oscillation is used, use a 33 pF capacitor externally.

**Electrical characteristics 2** ( $T_a = -20$  °C to 85 °C,  $V_{DD} = 1.8$  to 5.5 V, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit	
			Min.	Typ.	Max.		
IDD	Supply current	at active mode (with a ceramic resonator) (Notes 1, 2)	VDD = 5.0 V  f(XIN) = 6.0 MHz  f(RING) = stop	f(STCK) = f(XIN)/8  f(STCK) = f(XIN)/4  f(STCK) = f(XIN)/2  f(STCK) = f(XIN)	1.2  1.3  1.6  2.2	2.4  2.6  3.2  4.4	mA
			VDD = 5.0 V  f(XIN) = 4.0 MHz  f(RING) = stop	f(STCK) = f(XIN)/8  f(STCK) = f(XIN)/4  f(STCK) = f(XIN)/2  f(STCK) = f(XIN)	0.9  1  1.2	1.8  2  2.4	
			VDD = 3.0 V  f(XIN) = 2.0 MHz  f(RING) = stop	f(STCK) = f(XIN)/8  f(STCK) = f(XIN)/4  f(STCK) = f(XIN)/2  f(STCK) = f(XIN)	0.2  0.25  0.3	0.4  0.5  0.6	
			VDD = 5.0 V  f(XIN) = stop  f(RING) = operating	f(STCK) = f(RING)/8  f(STCK) = f(RING)/4  f(STCK) = f(RING)/2  f(STCK) = f(RING)	50  60  80	100  120  160	μA
			VDD = 3.0 V  f(XIN) = stop  f(RING) = operating	f(STCK) = f(RING)/8  f(STCK) = f(RING)/4  f(STCK) = f(RING)/2  f(STCK) = f(RING)	10  13  19	20  26  38	
			Ta = 25 °C  VDD = 5.0 V  VDD = 3.0 V			0.1  10  6	μA
						3	
						10	
						6	

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

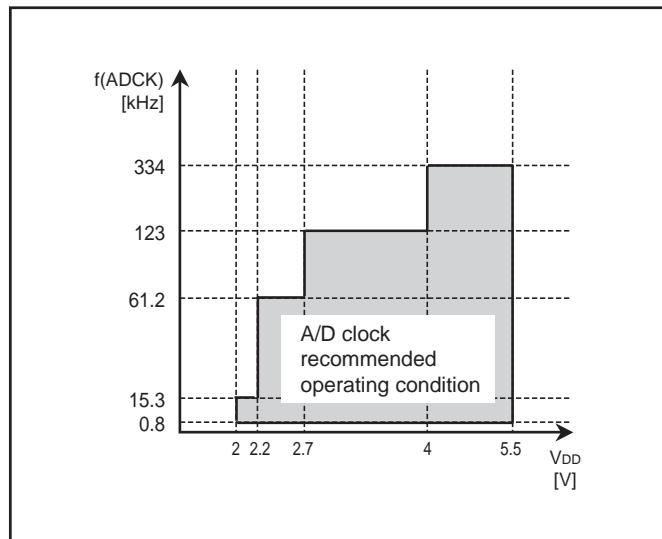
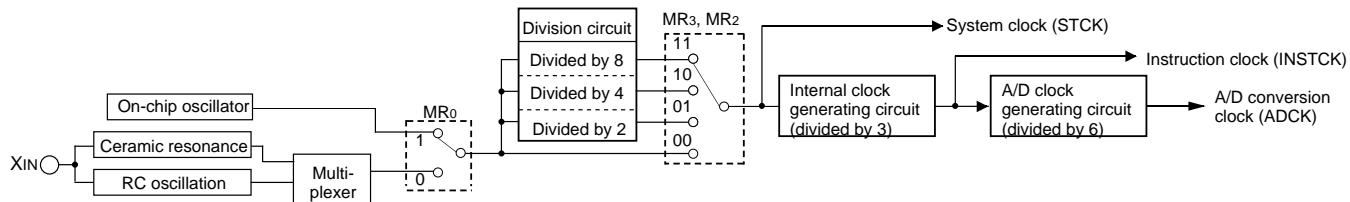
2: In the M34509G4H, the voltage drop detection circuit operation current (IRST) is added.

3: In the M34509G4H, when the SVDE instruction is executed, the voltage drop detection circuit operation current (IRST) is added.

**A/D converter recommended operating conditions**(Comparator mode included,  $T_a = -20$  °C to 85 °C, unless otherwise noted)

Symbol	Parameter	Conditions	Limits			Unit
			Min.	Typ.	Max.	
VDD	Supply voltage	$T_a = 0$ °C to 50 °C	2.0		5.5	V
		$T_a = -20$ °C to 85 °C	2.7		5.5	
VIA	Analog input voltage		0		VDD	V
f(ADCK)	A/D clock frequency (Note)	$VDD = 4.0$ V to 5.5 V	0.8		334	kHz
		$VDD = 2.7$ V to 5.5 V	0.8		123	
		$VDD = 2.2$ V to 5.5 V	0.8		61.2	
		$VDD = 2.0$ V to 5.5 V	0.8		15.3	

Note: Definition of A/D conversion clock (ADCK)

**A/D clock (ADCK) operating condition map**

**A/D converter characteristics**

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
-	Resolution				10	bits
-	Linearity error	Ta = 0 °C to 50 °C, 2.2 V ≤ VDD < 2.7 V			±4.0	LSB
		Ta = -20 °C to 85 °C, 2.7 V ≤ VDD ≤ 5.5 V			±2.0	
-	Differential non-linearity error	Ta = 0 °C to 50 °C, 2.2 V ≤ VDD < 2.7 V			±0.9	LSB
		Ta = -20 °C to 85 °C, 2.7 V ≤ VDD ≤ 5.5 V			±0.9	
V <sub>0T</sub>	Zero transition voltage	VDD = 2.56 V	0	7.5	15	mV
		VDD = 3.075 V	0	7.5	15	
		VDD = 5.12 V	0	10	20	
V <sub>FST</sub>	Full-scale transition voltage	VDD = 2.56 V	2552.5	2560	2567.5	mV
		VDD = 3.075 V	3064.5	3072	3079.5	
		VDD = 5.12 V	5100	5110	5120	
-	Absolute accuracy (Quantization error excluded)	Ta = 0 °C to 50 °C, 2.0 V ≤ VDD < 2.2 V			±8.0	LSB
I <sub>ADD</sub>	A/D operating current (Note 1)	VDD = 5.0 V		300	900	μA
		VDD = 3.0 V		100	300	
T <sub>CONV</sub>	A/D conversion time	f(ADCK) = 334 kHz			31	μs
		f(ADCK) = 123 kHz			85	
		f(ADCK) = 61.2 kHz			169	
		f(ADCK) = 15.3 kHz			676	
-	Comparator resolution				8	bits
-	Comparator error (Note 2)	VDD = 2.56 V			± 15	mV
		VDD = 3.072 V			± 15	
		VDD = 5.12 V			± 20	
-	Comparator comparison time	f(ADCK) = 334 kHz			4	μs
		f(ADCK) = 123 kHz			11	
		f(ADCK) = 61.2 kHz			22	
		f(ADCK) = 15.3 kHz			88	

Notes 1: When the A/D converter is used, the I<sub>ADD</sub> is added to I<sub>DD</sub>.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<sub>ref</sub> which is generated by the built-in DA converter can be obtained by the following formula.— Logic value of comparison voltage V<sub>ref</sub> —

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

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

## VOLTAGE DROP DETECTION CIRCUIT CHARACTERISTICS

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
VRST <sup>-</sup>	Detection voltage (reset occurs) (Note 2)	Ta = 25 °C		2.6		V
		-20 °C ≤ Ta < 0 °C	2.5		3.1	
		0 °C ≤ Ta < 50 °C	2.2		3	
		50 °C ≤ Ta ≤ 85 °C	2		2.7	
VRST <sup>+</sup>	Detection voltage (reset release) (Note 3)	Ta = 25 °C		2.7		V
		-20 °C ≤ Ta < 0 °C	2.6		3.2	
		0 °C ≤ Ta < 50 °C	2.3		3.1	
		50 °C ≤ Ta ≤ 85 °C	2.1		2.8	
VRST <sup>+</sup> – VRST <sup>-</sup>	Detection voltage hysteresis			0.1		V
IRST	Operation current (Note 4)	VDD = 5 V		50	100	μA
		VDD = 3 V		30	60	
TRST	Detection time (Note 5)	VDD → [VRST <sup>-</sup> – 0.1 V]		0.2	1.2	ms

Notes 1: The voltage drop detection circuit is equipped with only the M34509G4H.

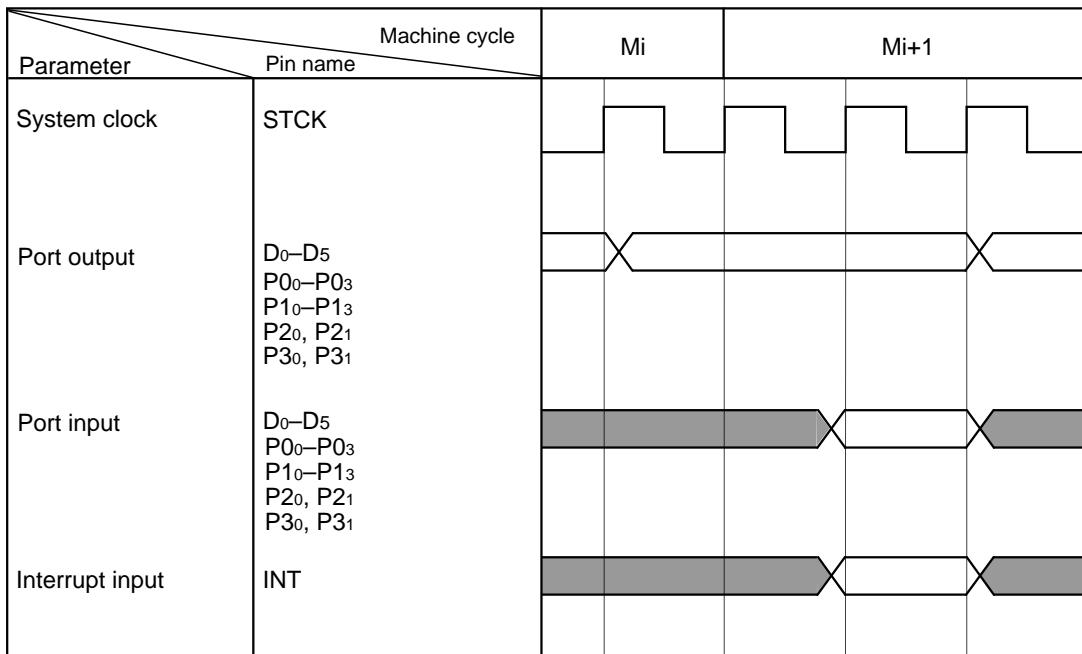
2: The detection voltage (VRST<sup>-</sup>) is defined as the voltage when reset occurs when the supply voltage (VDD) is falling.

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

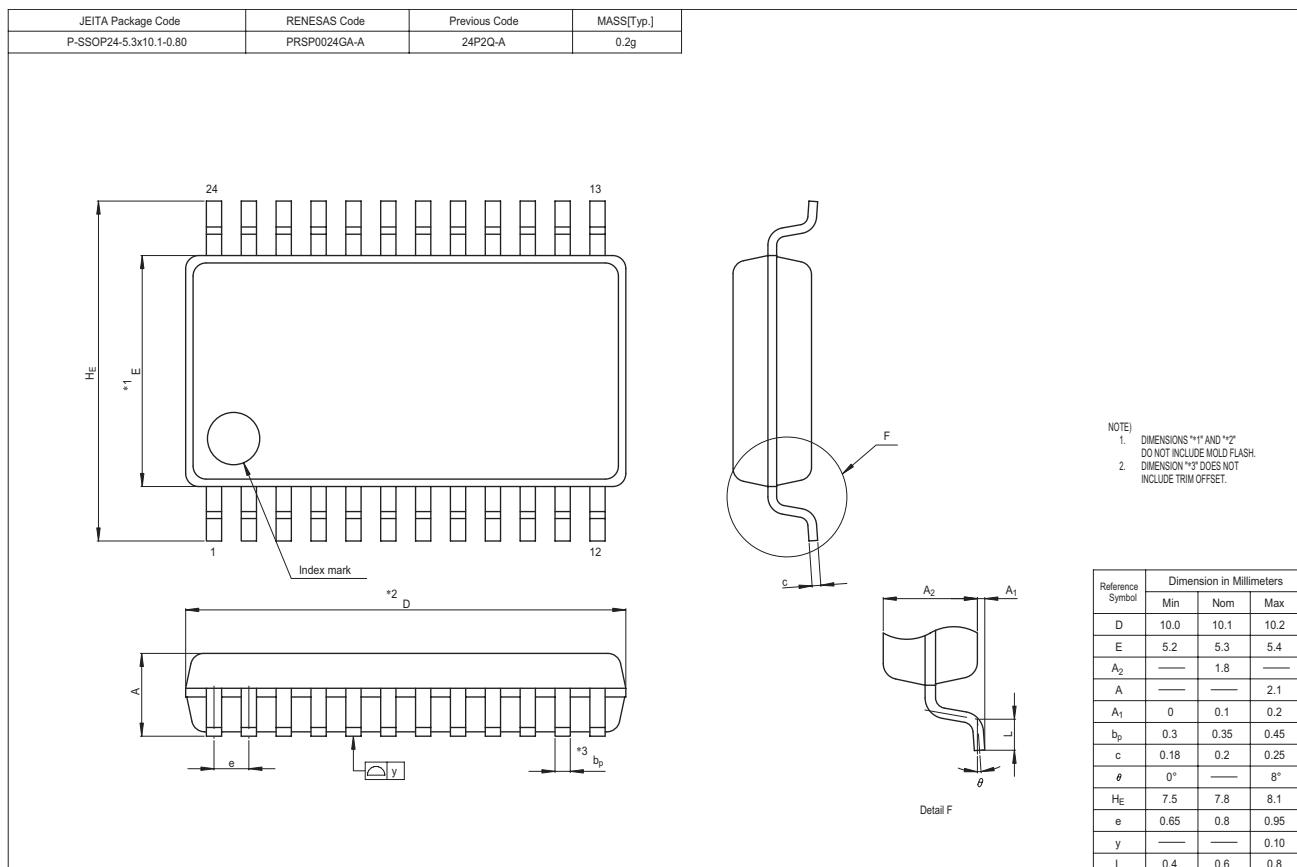
4: In the M34509G4H, IRST is added to IDD (supply current).

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

### Basic timing diagram



## Package outline



## REVISION HISTORY

## 4509 Group Data Sheet

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 18, 2005	–	First edition issued
1.01	Aug. 12, 2005	17 52 57 58 62 130 131	ROM Code Protect Address added. Table 20: Some description about Port P1 added. Fig.52 revised. Fig.54 revised. “DATA REQUIRED FOR QzROM WRITING ORDERS” added. Notes On ROM Code Protect added. A/D converter characteristics: Linearity error, Differential non-linearity error and Absolute accuracy → Parameters and Test conditions revised. Voltage drop detection circuit characteristics: VRST <sup>–</sup> , VRST <sup>+</sup> → Test conditions revised.
1.02	Dec. 22, 2006	5 26 28 30 43 53 58 59 to 61 63 67 to 70 76 102 117 134 135 137 139 →	MULFUNCTION: Note 4 revised. TIMER: Description revised and Structure of Timer 2 in Table 9 revised. Fig.23: INSTCK (wrong) → INTSNC (correct) (2) Prescaler: <u>PRS</u> → <u>RPS</u> (3) Timer 3 → Timer 1 SERIAL I/O: Table 14: Note revised. Fig. 46: Notes revised. Table 23: Changes referring ahead and note 5 added. QzROM Writing Mode added. LIST OF PRECAUTIONS: Malfuction revised. NOTES ON NOISE added. Description of Port output structure control register FR2 and FR3 revised. Instruction code of TAL1 revised. Description of TALA revised. Detailed description of TEAB revised. f(Sck): Serial interface external input <u>frequency</u> → Serial interface external input <u>period</u> Δ f(XIN): Ta = <u>around</u> 25 °C → <u>center</u> 25 °C Figure title revised, “When ceramic resonator is used” deleted. Note 4: ( <u>power current</u> ) → ( <u>supply current</u> ) → Pages 79–81, 93–95, 114, 122–129: Description of SNZ0, SNZT1, SNZT2, SNZAD, SNZSI and WRST instructions revised.
1.03	Jul. 27, 2009	50 134 136	Fig 45: Note revised. f(Sck): Serial interface external input <u>period</u> → Serial interface external input <u>frequency</u> Note 1: ...., the A/D operation current (IADD) is <u>included</u> . → ...., the A/D operation current (IADD) is <u>added</u> .

Rev.	Date	Description	
		Page	Summary
		138	<p>Linearity error: <math>T_a = 0 \text{ }^{\circ}\text{C}</math> to <math>50 \text{ }^{\circ}\text{C}</math>, <math>2.2 \text{ V} \leq V_{DD} \leq 2.7 \text{ V}</math> <math>\rightarrow</math></p> <p><math>T_a = 0 \text{ }^{\circ}\text{C}</math> to <math>50 \text{ }^{\circ}\text{C}</math>, <math>2.2 \text{ V} \leq V_{DD} &lt; 2.7 \text{ V}</math></p> <p>Note 1: ...., the I<sub>ADD</sub> is <u>included</u> to I<sub>DD</sub>. <math>\rightarrow</math> ...., the I<sub>ADD</sub> is <u>added</u> to I<sub>DD</sub>.</p>

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

### Notes:

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



### RENESAS SALES OFFICES

<http://www.renesas.com>

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

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

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

**Renesas Technology (Shanghai) Co., Ltd.**  
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120  
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

**Renesas Technology Hong Kong Ltd.**  
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong  
Tel: <852> 2265-6688, Fax: <852> 2377-3473

**Renesas Technology Taiwan Co., Ltd.**  
10th Floor, No.99, Fushing North Road, Taipei, Taiwan  
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

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

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

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