4510 Group

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

DESCRIPTION

The 4510 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 serial I/O, four 8-bit timers (each timer has a reload register), A-D converter, and zero cross detection circuit.

The various microcomputers in the 4510 Group include variations of the built-in memory type and package as shown in the table below.

FEATURES

- Minimum instruction execution time ..... 0.5 μs (at 6.0 MHz system clock frequency)
- Supply voltage
  4.5 V to 5.5 V (at 6.0 MHz system clock frequency)
  4.0 V to 5.5 V (at 4.0 MHz system clock frequency)
  2.5 V to 5.5 V (at 2.0 MHz system clock frequency)
  2.0 V to 5.5 V (at 1.5 MHz system clock frequency: only for Mask ROM versions)
- Timers
  Timer 1 ..... 8-bit timer with a reload register
  Timer 2 ..... 8-bit timer with a reload register
  Timer 3 ..... 8-bit timer with a reload register
  Timer 4 ..... 8-bit timer with a reload register
- Interrupt ..... 8 sources
- Serial I/O ..... 8 bit-wide
- A-D converter ..... 8-bit successive comparison method X 4ch
- Zero cross detection circuit 1
- Real time output circuit 1
- Watchdog timer ..... 16 bits
- Voltage drop detection circuit 1
- Clock generating circuit 1
- LED drive directly enabled (port D)

APPLICATION

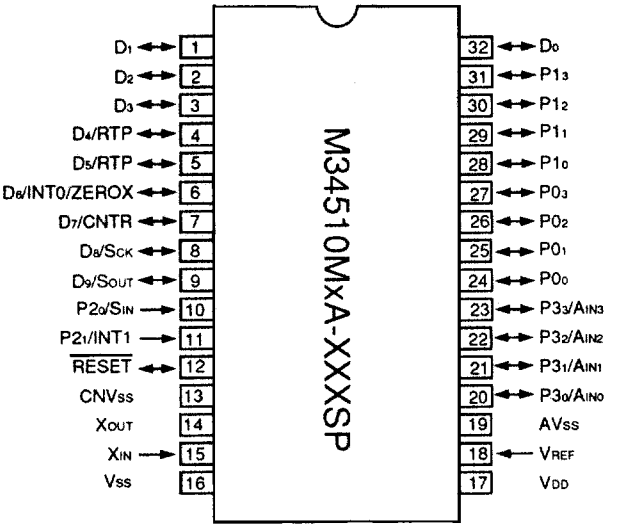
VCR, microwave oven, rice cooker, audio, telephone, office equipment

Product	ROM (PROM) size (X 10 bits)	RAM size (X 4 bits)	Package	ROM type
M34510M2A-XXXSP/FP	2048 words	128 words	SP: 32P4B FP: 36P2R-A	Mask ROM
M34510M4A-XXXSP/FP	4096 words	256 words		
M34510E8-XXXSP/FP *	8192 words	384 words		One Time PROM
M34510E8SS **			32S1B	EPROM

*: Shipped after writing (shipped in blank: M34510E8SP/FP)
**: For program development

PIN CONFIGURATION (TOP VIEW)

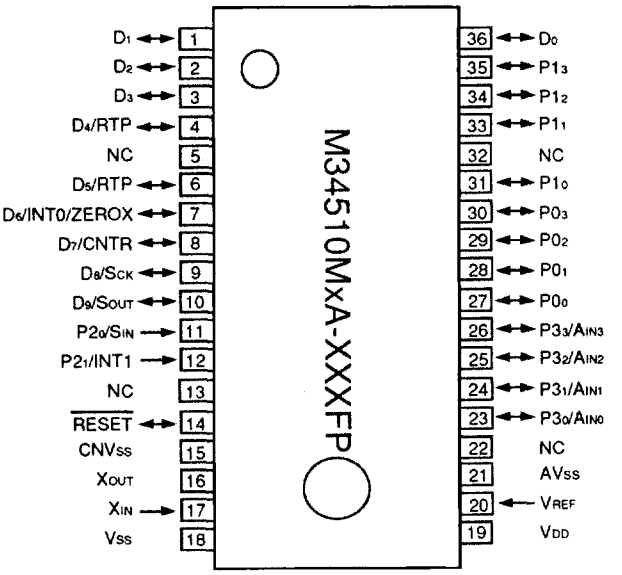
M34510MxA-XXXSP



Outline 32P4B

PIN CONFIGURATION (TOP VIEW)

M34510MxA-XXXFP

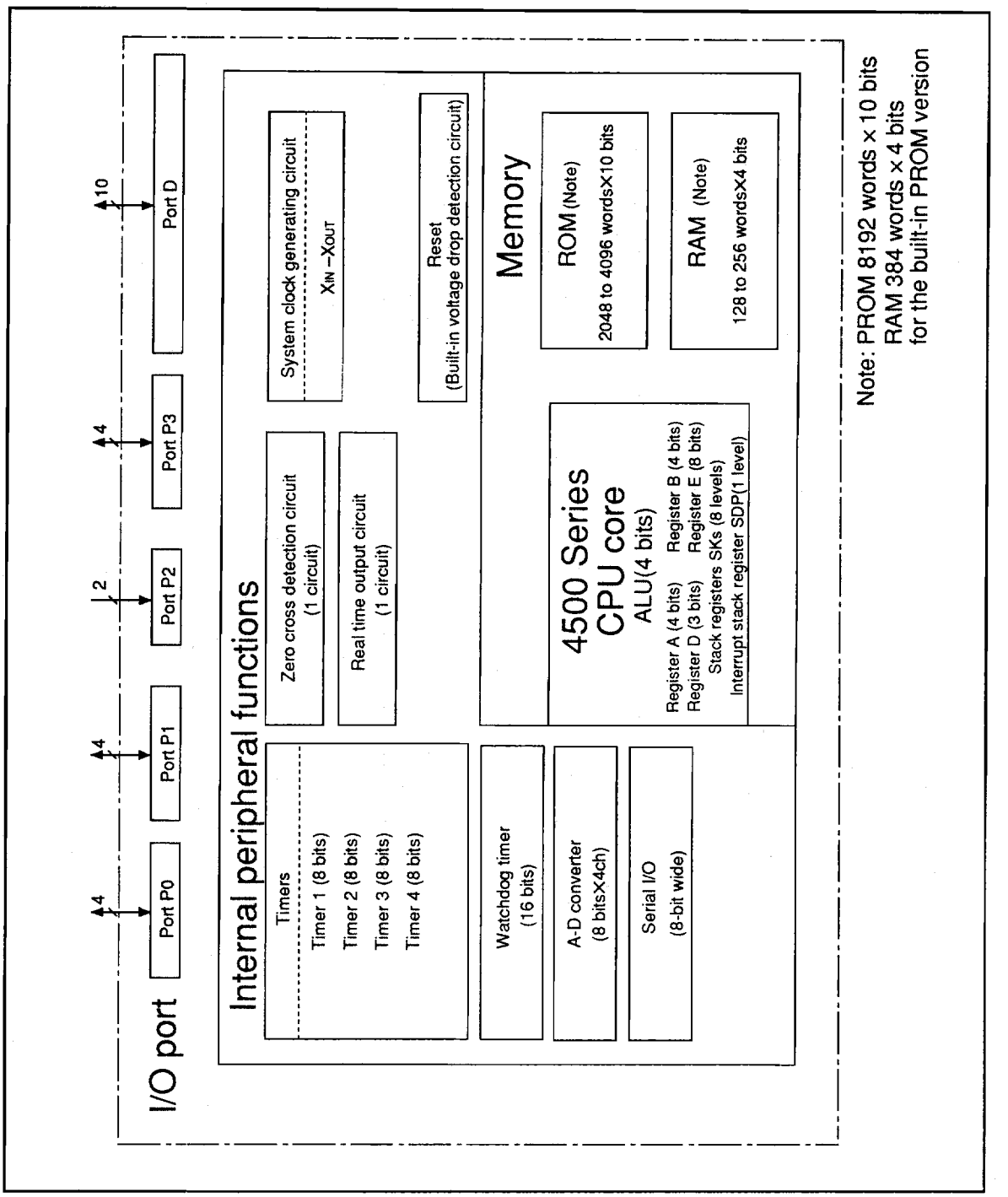


NC : No connected
(not connected to internal circuit)

Outline 36P2R-A

MITSUBISHI ELECTRIC

BLOCK DIAGRAM



PERFORMANCE OVERVIEW

Parameter			Function
Number of basic instructions			122
Minimum instruction execution time			0.5 μs (at 6.0 MHz system clock frequency) (Refer to the electrical characteristics because the minimum instruction execution time depends on the supply voltage.)
Memory sizes	ROM	M34510M2A	2048 words × 10 bits
		M34510M4A	4096 words × 10 bits
		M34510E8	8192 words × 10 bits
	RAM	M34510M2A	128 words × 4 bits
		M34510M4A	256 words × 4 bits
		M34510E8	384 words × 4 bits
Input/Output ports	D_0–D_9	I/O	Ten independent I/O ports; ports D_4 and D_5 are also used as RTP. Ports D_6–D_9 are also used as INT0, CNTR, SCK, and SOUT, respectively.
	$P0_0$–$P0_3$	I/O	4-bit I/O port; each pin is equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software.
	$P1_0$–$P1_3$	I/O	4-bit I/O port; each pin is equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software.
	$P2_0$–$P2_1$	Input	2-bit input port; ports $P2_0$ and $P2_1$ are also used as SIN and INT1, respectively.
	$P3_0$–$P3_3$	I/O	4-bit I/O port; these pins are also used as four analog input pins.
	CNTR	I/O	Timer I/O; CNTR pin is also used as port D_7.
	INT0	Input	Interrupt input; INT0 pin is also used as port D_6 and zero cross input pin. INT0 pin is equipped with a key-on wakeup function.
	INT1	Input	Interrupt input; INT1 pin is also used as port $P2_1$.
	RTP	Output	Real time output; RTP pins are also used as ports D_4 and D_5.
	ZEROX	Input	Zero cross input; ZEROX pin is also used as port D_6.
Timers	Timer 1		8-bit programmable timer with a reload register
	Timer 2		8-bit programmable timer with a reload register is also used as an event counter.
	Timer 3		8-bit programmable timer with a reload register
	Timer 4		8-bit programmable timer with a reload register is also used as an event counter.
A-D converter			8 bits × 1; comparator can be used by software.
Serial I/O			8-bit wide
Interrupt	Sources		8 (two for external, four for timer, one for A-D, and one for serial I/O)
	Nesting		1 level
Subroutine nesting			8 levels
Device structure			CMOS silicon gate
Package	M34510MxA-XXXSP		32-pin plastic molded SDIP (32P4B)
	M34510MxA-XXXFP		36-pin plastic molded SSOP (36P2R-A)
Operating temperature range			−20 °C to 85 °C (−20 °C to 70 °C for the built-in EPROM version, M34510E8SS)
Supply voltage			2.0 V to 5.5 V (Refer to the electrical characteristics because the supply voltage depends on the system clock frequency.)
Power dissipation (typical value)	Active mode		4 mA (at 6.0 MHz system clock frequency, V_{DD} = 5 V, output transistors in the cut-off state)
			2 mA (at 1.5 MHz system clock frequency, V_{DD} = 5 V, output transistors in the cut-off state)
	RAM back-up mode		0.1 μA (at room temperature, V_{DD} = 5 V, output transistors in the cut-off state)

PIN DESCRIPTION

Pin	Name	Input/Output	Function
V_{DD}	Power supply	—	Connected to a plus power supply.
V_{SS}	Ground	—	Connected to a 0 V power supply.
CNVss	CNVss	—	Connect CNVss to Vss and apply "L" (0V) to CNVss certainly.
AVss	Ground for analog power supply	—	Connected to a 0 V power supply for A-D converter. Connect AVss to Vss.
V_{REF}	Reference voltage input	—	Reference voltage input pin for the A-D converter.
A_{IN0}–A_{IN3}	Analog input	Input	Analog input pins for A-D converter. A_{IN0}–A_{IN3} are also used as ports $P3_0$–$P3_3$.
$\overline{RESET}$	Reset I/O	I/O	An N-channel open-drain I/O pin for a system reset. When the watchdog timer causes the system to be reset, the $\overline{RESET}$ pin outputs "L" level.
X_{IN}	System clock input	Input	I/O pins of the system clock generating circuit. X_{IN} and X_{OUT} can be connected to a ceramic resonator. A feedback resistor is built-in between them.
X_{OUT}	System clock output	Output	
D_0–D_9	I/O port D	I/O	Each pin of port D has an independent 1-bit wide I/O function. Each pin has an output latch. For input use, set the latch of the specified bit to "1." The output structure is N-channel open-drain. Ports D_4 and D_5 are also used as RTP. Ports D_6–D_9 are also used as INT0/ZEROX, CNTR, S_{CK}, and S_{OUT}, respectively.
$P0_0$–$P0_3$	I/O port P0	I/O	Each of ports P0 and P1 serves as a 4-bit I/O port, and it can be used as inputs when the output latch is set to "1." The output structure is N-channel open-drain. Every pin of the ports has a key-on wakeup function and a pull-up function. Both functions can be switched by software.
$P1_0$–$P1_3$	I/O port P1		
$P2_0$–$P2_1$	Input port P2	Input	2-bit input port. Ports $P2_0$ and $P2_1$ are also used as S_{IN} and INT1, respectively.
$P3_0$–$P3_3$	I/O port P3	I/O	4-bit I/O port. The output structure is N-channel open-drain. Port P3 is also used as analog input pins.
CNTR	Timer input/output	I/O	CNTR pin has the function to input the clock for the timer 2 and timer 4 event counters, and to output the timer 1 underflow signal divided by 2 or the signal of X_{IN} divided by 2. CNTR pin is also used as port D_7.
INT0	Interrupt input	Input	INT0 pin accepts an external interrupt. It also accepts the input signal to return the system from the RAM back-up state. INT0 pin is also used as port D_6.
INT1	Interrupt input	Input	INT1 pin accepts an external interrupt. INT1 pin is also used as port $P2_1$.
ZEROX	Zero cross input	Input	ZEROX pin is also used as the zero cross input pin by software. ZEROX pin is also used as D_6/INT0 pin.
S_{IN}	Serial data input	Input	S_{IN} pin is used to input serial data signals by software. S_{IN} pin is also used as port $P2_0$.
S_{OUT}	Serial data output	Output	S_{OUT} pin is used to output serial data signals by software. S_{OUT} pin is also used as port D_9.
S_{CK}	Serial I/O clock input/output	I/O	S_{CK} pin is used to input and output synchronous clock signals for serial data transfer by software. S_{CK} pin is also used as port D_8.
RTP	Real time output	Output	RTP pins are also used as real time output pins by software. RTP pins are also used as ports D_4 and D_5, respectively.

MULTIFUNCTION

Pin	Multifunction	Pin	Multifunction
D_4	RTP	RTP (Note 2)	D_4
D_5	RTP	RTP	D_5
D_6	INT0/ZEROX	INT0 (Note 2)	D_6/ZEROX
D_7	CNTR	INT1	$P2_1$
D_8	S_{CK}	S_{CK} (Note 2)	D_8
D_9	S_{OUT}	S_{OUT}	D_9
$P2_0$	S_{IN}	S_{IN}	$P2_0$
$P2_1$	INT1	CNTR (Note 2)	D_7
$P3_0$	A_{IN0}	A_{IN0} (Note 2)	$P3_0$
$P3_1$	A_{IN1}	A_{IN1}	$P3_1$
$P3_2$	A_{IN2}	A_{IN2}	$P3_2$
$P3_3$	A_{IN3}	A_{IN3}	$P3_3$
		ZEROX (Note 2)	D_6/INT0

Notes 1: Pins except above have just single function.

2: The I/O of D_6, the input of D_4, D_5, D_7–D_9, $P2_0$ and $P2_1$, and the output of $P3_0$–$P3_3$ can be used even when INT0/ZEROX, RTP, CNTR, S_{CK}, S_{OUT}, S_{IN}, INT1 and A_{IN0}–A_{IN3} are selected.

CONNECTIONS OF UNUSED PINS

Pin	Connection	Pin	Connection
X_{OUT}	Open (when using an external clock)	$P2_0$/S_{IN}	Connect to Vss
AVss/V_{REF}	Connect to Vss	$P2_1$/INT1	
D_0–D_3	Connect to Vss, or set the output latch to "0" and open.	$P3_0$/A_{IN0}–$P3_3$/A_{IN3}	Connect to Vss, or set the output latch to "0" and open.
D_4/RTP		$P0_0$–$P0_3$	Open or connect to Vss (Note)
D_5/RTP		$P1_0$–$P1_3$	Open or connect to Vss (Note)
D_6/INT0/ZEROX			
D_7/CNTR			
D_8/S_{CK}			
D_9/S_{OUT}			

Note: When the $P0_0$–$P0_3$ and $P1_0$–$P1_3$ are connected to Vss, turn off their pull-up transistors (register $PU0_i$="0") and also invalidate the key-on wakeup functions (register $K0_i$="0") by software. When these pins are connected to Vss while the key-on wakeup functions are left valid, the system fails to return from RAM back-up state. When these pins are open, turn on their pull-up transistors (register $PU0_i$="1") by software, or set the output latch to "0."
Be sure to select the key-on wakeup functions and the pull-up functions with every two pins. If only one of the two pins for the key-on wakeup function is used, turn on their pull-up transistors by software and also disconnect the other pin. (i = 0, 1, 2, or 3.)

(Note when the output latch is set to "0" and pins are open)

- After system is released from reset, port is in a high-impedance state until it is set the output latch to "0" by software. Accordingly, the voltage level of pins is undefined and the excess of the supply current may occur while the port is in a high-impedance state.
- To set the output latch periodically by software is recommended because value of output latch may change by noise or a program run away (caused by noise).

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

- Connect the unused pins to Vss or V_{DD} using the thickest wire at the shortest distance against noise.

PORT FUNCTION

Port	Pin	Input/ Output	Output structure	Control bits	Control instructions	Control registers	Remark
Port D	D_0–D_3	I/O (10)	N-channel open-drain	1	SD		
	D_4/RTP				RD	RTR	
	D_5/RTP				SZD	RTR	
	D_6/INT0/ZEROX				CLD	I1	Key-on wakeup function
	D_7/CNTR				SNZI0 (Note 1)	W6	
	D_8/SCK					J1	
	D_9/SOUT					J1	
Port P0	$P0_0$–$P0_3$	I/O (4)	N-channel open-drain	4	OP0A IAP0	PU0 K0	Built-in programmable pull-up functions Key-on wakeup functions (programmable)
Port P1	$P1_0$–$P1_3$	I/O (4)	N-channel open-drain	4	OP1A IAP1	PU0 K0	Built-in programmable pull-up functions Key-on wakeup functions (programmable)
Port P2	$P2_0$/SIN $P2_1$/INT1	Input (2)		2	IAP2 SNZI1 (Note 2)		
Port P3	$P3_0$/AIN0 $P3_1$/AIN1 $P3_2$/AIN2 $P3_3$/AIN3	I/O (4)	N-channel open-drain	4	IAP3 OP3A	Q2	

Notes 1: Level of the D_6/INT0 pin can be examined with the SNZI0 instruction.
2: Level of the $P2_1$/INT1 pin can be examined with the SNZI1 instruction.

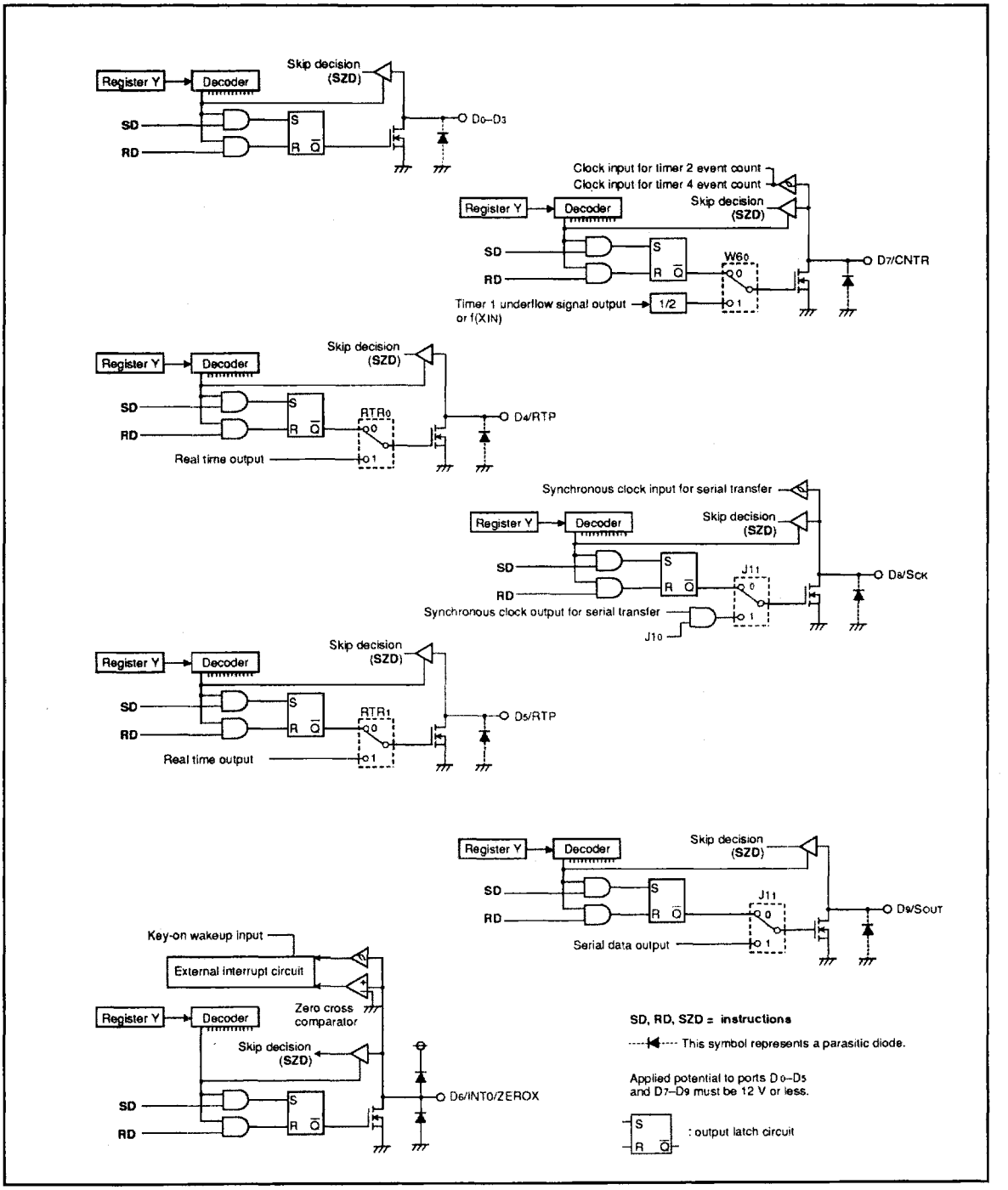
DEFINITION OF CLOCK AND CYCLE

●System clock
This is the source clock input to the XIN pin. Either the clock obtained by connecting an oscillator between the XIN and XOUT pins, or the external clock supplied through the XIN pin can be specified.

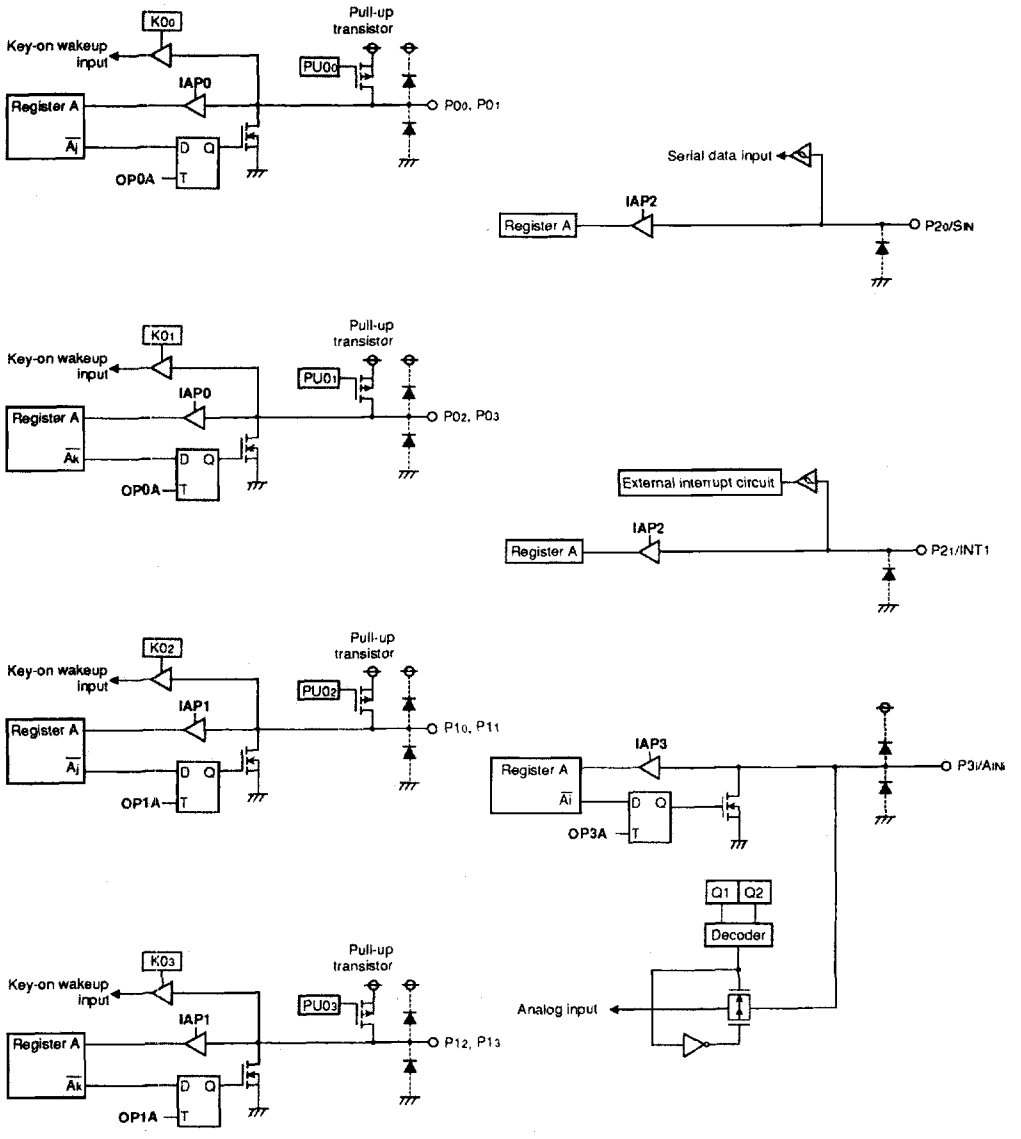
●Instruction clock
The instruction clock is a signal derived by dividing the system clock by 3, and is the basic clock for controlling this product.

●Machine cycle
One machine cycle is the time required to execute the minimum instruction (one-cycle instruction). The machine cycle is equivalent to the instruction clock cycle.

PORT BLOCK DIAGRAMS



PORT BLOCK DIAGRAMS (continued)



IAP0, OP0A, IAP1, OP1A, IAP2, IAP3 and OP3A = Instructions

D Q
T
: output latch circuit

----◄---- This symbol represents a parasitic diode.

Applied potential to ports P20 and P21 must be VDD or less.

i represents 0, 1, 2 or 3.
j represents 0 or 1.
k represents 2 or 3.

FUNCTION BLOCK OPERATIONS
CPU

(1) Arithmetic logic unit (ALU)

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

(2) Register A and carry flag

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

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

It is unchanged with both A n instruction and AM instruction. The value of A_0 is stored in carry flag CY with the RAR instruction (Figure 2).

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

(3) Registers B and E

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

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

(4) Register D

Register D is a 3-bit register.

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

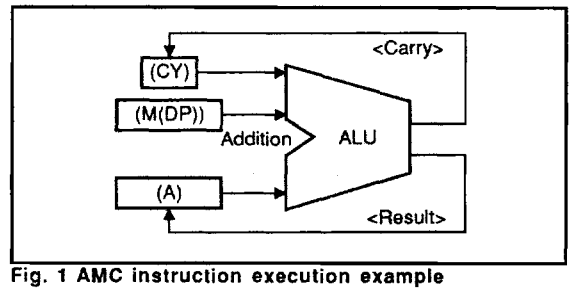


Fig. 1 AMC instruction execution example

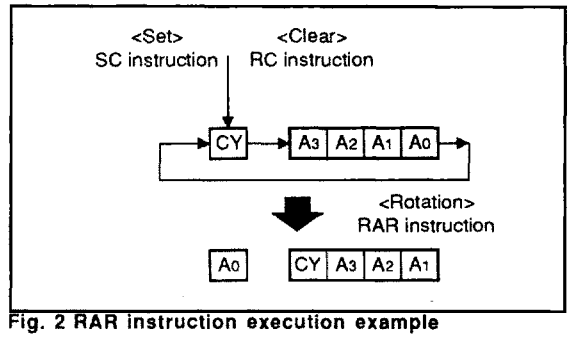


Fig. 2 RAR instruction execution example

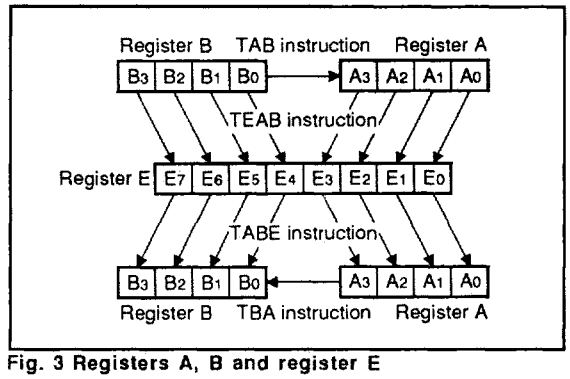


Fig. 3 Registers A, B and register E

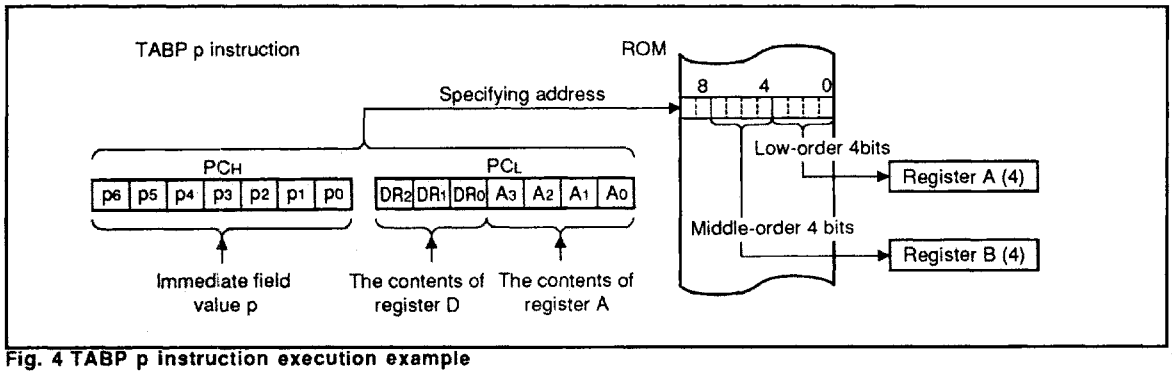


Fig. 4 TABP p instruction execution example

MITSUBISHI ELECTRIC

(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;

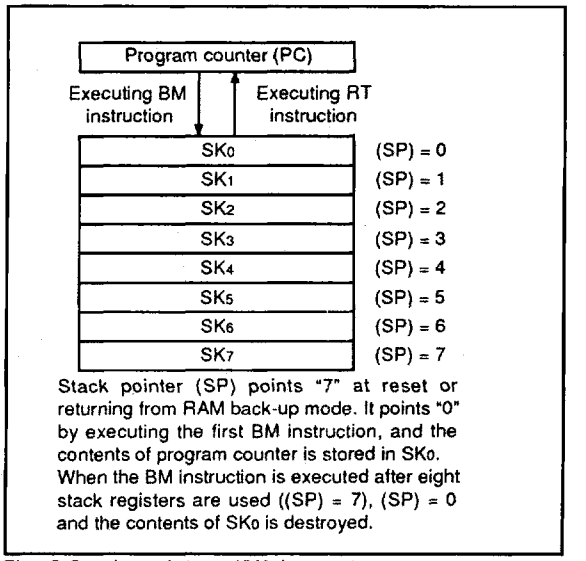
- branching to an interrupt service routine (referred to as an interrupt service routine),
- performing a subroutine call, or
- executing the table reference instruction (TABP p).

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.



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

Fig. 5 Stack registers (SKs) structure

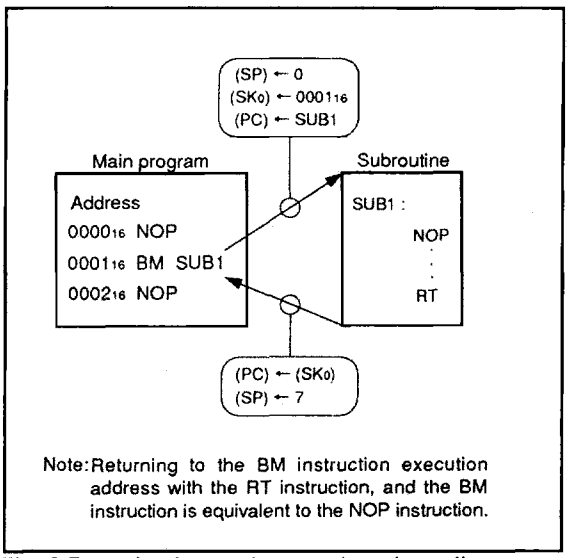
(6) Interrupt stack register (SDP)

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.



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

Fig. 6 Example of operation at subroutine call

(8) Program counter (PC)

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

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

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

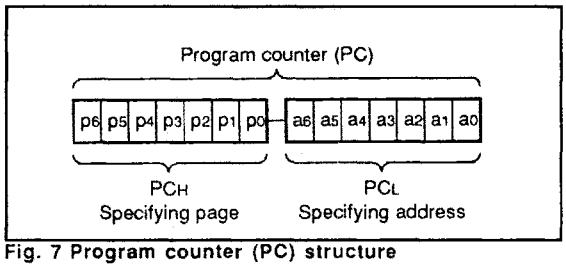


Fig. 7 Program counter (PC) structure

(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).

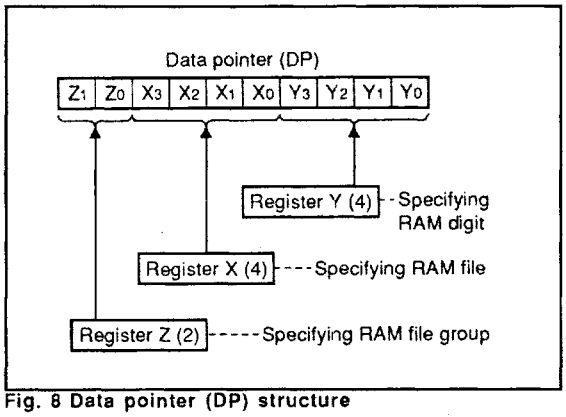


Fig. 8 Data pointer (DP) structure

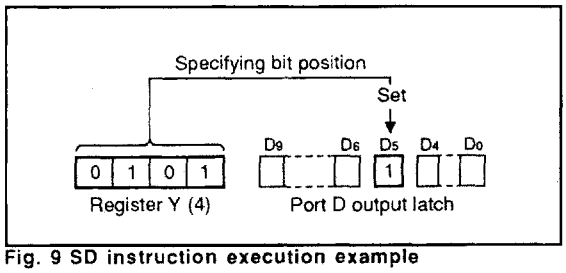


Fig. 9 SD instruction execution example

PROGRAM MEMORY (ROM)

The program memory is a mask ROM. 1 word of ROM is composed of 10 bits. ROM is separated every 128 words by the unit of page (addresses 0 to 127). Table 1 shows the ROM size and pages. Figure 10 shows the ROM map of M34510E8.

Table 1 ROM size and pages

Product	ROM size (X 10 bits)	Pages
M34510M2A	2048 words	16 (0 to 15)
M34510M4A	4096 words	32 (0 to 31)
M34510E8	8192 words	64 (0 to 63)

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

Page 2 (addresses 0100_{16} to $017F_{16}$) is the special page for subroutine calls. Subroutines written in this page can be called from any page with the 1-word instruction (BM). Subroutines extending from page 2 to another page can also be called with the BM instruction when it starts on page 2. ROM pattern (bits 7 to 0) of all addresses can be used as data areas with the TABP p instruction.

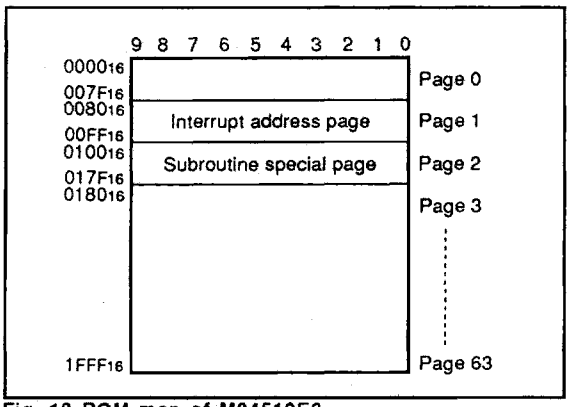


Fig. 10 ROM map of M34510E8

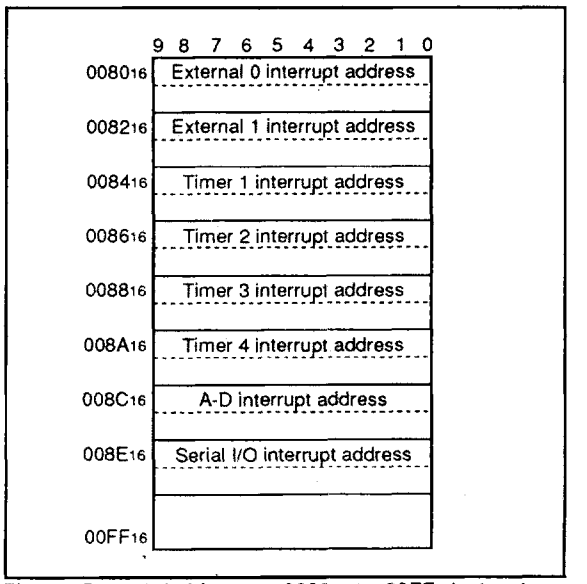


Fig. 11 Page 1 (addresses 0080_{16} to $00FF_{16}$) structure

DATA MEMORY (RAM)

1 word of RAM is composed of 4 bits, but 1-bit manipulation (with the SB j, RB j, and SZB j instructions) is enabled for the entire memory area. A RAM address is specified by a data pointer. The data pointer consists of registers Z, X, and Y. Set a value to the data pointer certainly when executing an Instruction to access RAM.
Table 2 shows the RAM size. Figure 12 shows the RAM map.

Table 2 RAM size

Product	RAM size
M34510M2A	128 words × 4 bits (512 bits)
M34510M4A	256 words × 4 bits (1024 bits)
M34510E8	384 words × 4 bits (1536 bits)

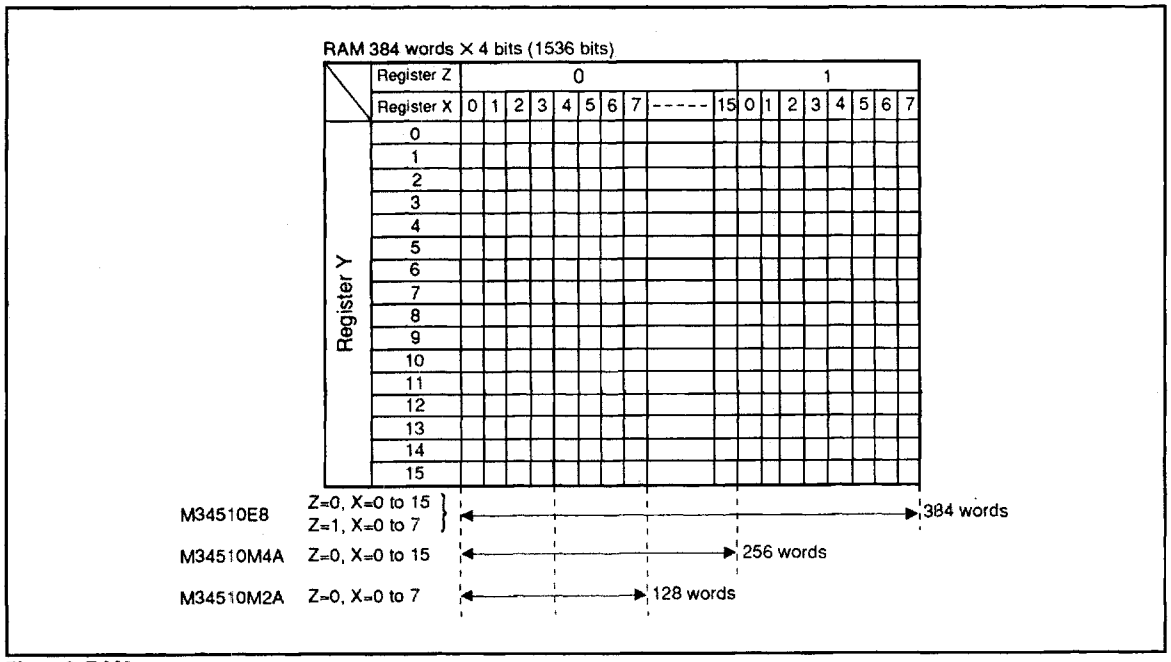


Fig. 12 RAM map

INTERRUPT FUNCTION

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

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

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

(1) Interrupt enable flag (INTE)

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

(2) Interrupt enable bit

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

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

Table 5 shows the interrupt enable bit function.

(3) Interrupt request flag

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

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

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

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

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

Table 3 Interrupt sources

Priority level	Interrupt name	Activated condition	Interrupt address
1	External 0 interrupt	Level change of INT0 pin	Address 0 in page 1
2	External 1 interrupt	Level change of INT1 pin	Address 2 in page 1
3	Timer 1 interrupt	Timer 1 underflow	Address 4 in page 1
4	Timer 2 interrupt	Timer 2 underflow	Address 6 in page 1
5	Timer 3 interrupt	Timer 3 underflow	Address 8 in page 1
6	Timer 4 interrupt	Timer 4 underflow	Address A in page 1
7	A-D interrupt	Completion of A-D conversion	Address C in page 1
8	Serial I/O interrupt	Completion of serial I/O transfer	Address E in page 1

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

Interrupt name	Request flag	Skip instruction	Enable bit
External 0 interrupt	EXF0	SNZ0	$V1_0$
External 1 interrupt	EXF1	SNZ1	$V1_1$
Timer 1 interrupt	T1F	SNZT1	$V1_2$
Timer 2 interrupt	T2F	SNZT2	$V1_3$
Timer 3 interrupt	T3F	SNZT3	$V2_0$
Timer 4 interrupt	T4F	SNZT4	$V2_1$
A-D interrupt	ADF	SNZAD	$V2_2$
Serial I/O interrupt	SIOF	SNZSI	$V2_3$

Table 5 Interrupt enable bit function

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

(4) Internal state during an interrupt

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

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

(5) Interrupt processing

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

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

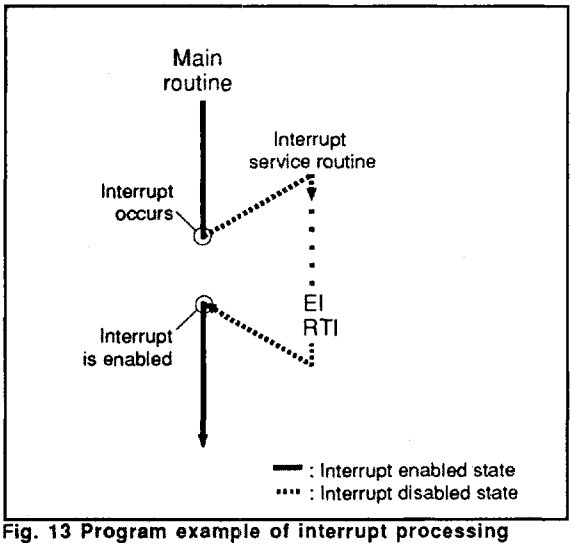


Fig. 13 Program example of interrupt processing

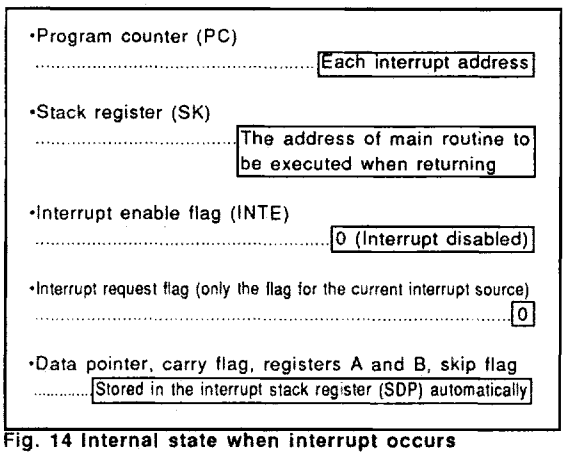


Fig. 14 Internal state when interrupt occurs

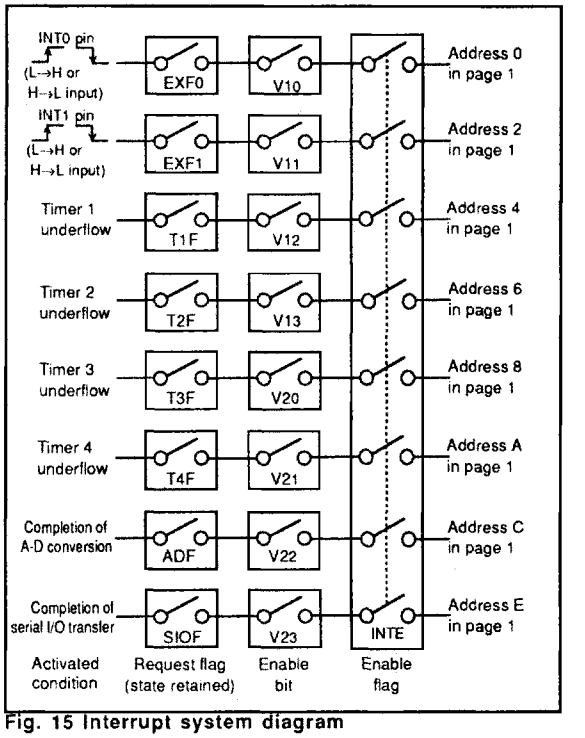


Fig. 15 Interrupt system diagram

(6) Interrupt control registers

- Interrupt control register V1

Interrupt enable bits of external 0, external 1, 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

Interrupt enable bits of timer 3, timer 4, A-D and serial I/O 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.

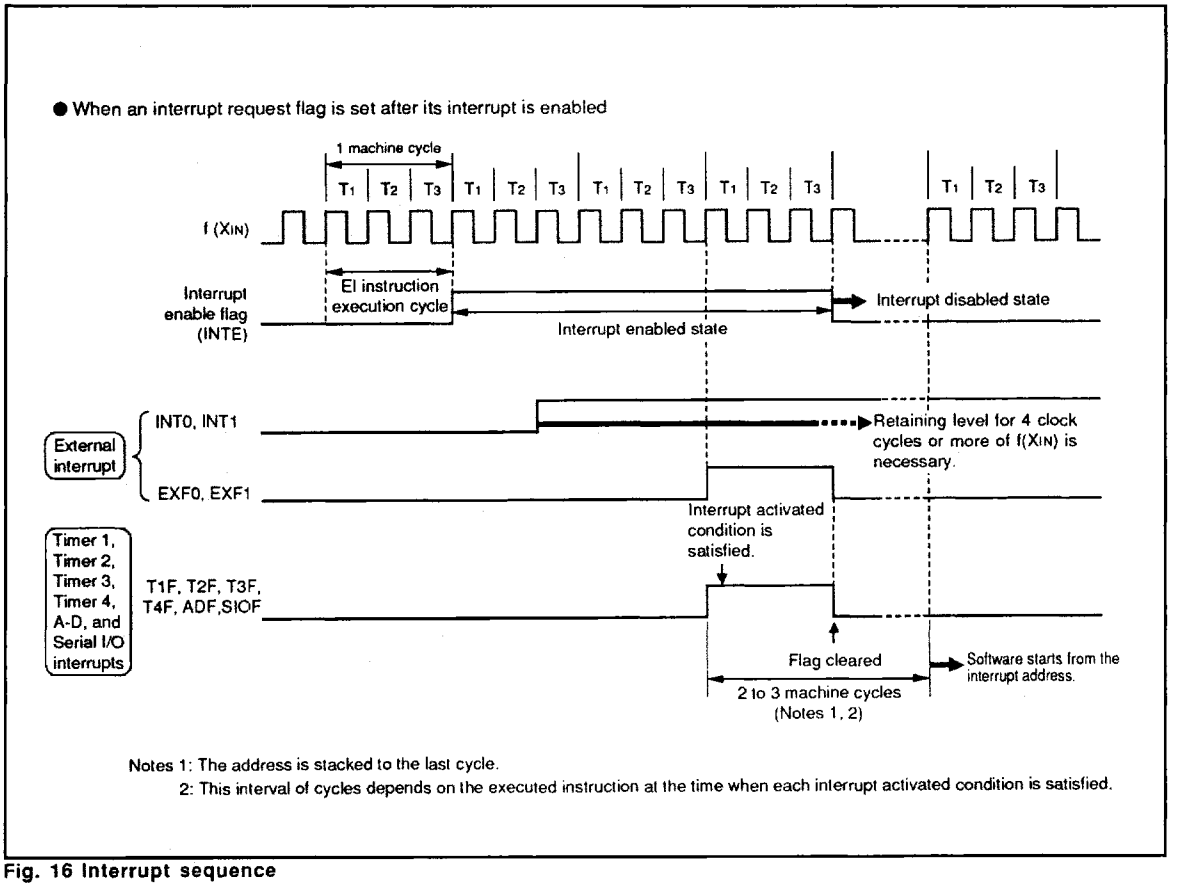
Table 6 Interrupt control registers

Interrupt control register V1		at reset : 0000_2	at RAM back-up : 0000_2	R/W
$V1_3$	Timer 2 interrupt enable bit	0	Interrupt disabled (SNZT2 instruction is valid)	
		1	Interrupt enabled (SNZT2 instruction is invalid)	
$V1_2$	Timer 1 interrupt enable bit	0	Interrupt disabled (SNZT1 instruction is valid)	
		1	Interrupt enabled (SNZT1 instruction is invalid)	
$V1_1$	External 1 interrupt enable bit	0	Interrupt disabled (SNZ1 instruction is valid)	
		1	Interrupt enabled (SNZ1 instruction is invalid)	
$V1_0$	External 0 interrupt enable bit	0	Interrupt disabled (SNZ0 instruction is valid)	
		1	Interrupt enabled (SNZ0 instruction is invalid)	
Interrupt control register V2		at reset : 0000_2	at RAM back-up : 0000_2	R/W
$V2_3$	Serial I/O interrupt enable bit	0	Interrupt disabled (SNZSI instruction is valid)	
		1	Interrupt enabled (SNZSI instruction is invalid)	
$V2_2$	A-D interrupt enable bit	0	Interrupt disabled (SNZAD instruction is valid)	
		1	Interrupt enabled (SNZAD instruction is invalid)	
$V2_1$	Timer 4 interrupt enable bit	0	Interrupt disabled (SNZT4 instruction is valid)	
		1	Interrupt enabled (SNZT4 instruction is invalid)	
$V2_0$	Timer 3 interrupt enable bit	0	Interrupt disabled (SNZT3 instruction is valid)	
		1	Interrupt enabled (SNZT3 instruction is invalid)	

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

(7) Interrupt sequence

Interrupts only occur when the respective INTE flag, interrupt enable bits ($V1_0$–$V1_3$ and $V2_0$–$V2_3$), 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).



Notes 1: The address is stacked to the last cycle.
2: This interval of cycles depends on the executed instruction at the time when each interrupt activated condition is satisfied.

Fig. 16 Interrupt sequence

EXTERNAL INTERRUPTS

The 4510 Group has two external interrupts (external 0 and external 1). An external interrupt request occurs when a valid waveform is input to an interrupt input pin (edge detection).

External 0 interrupt is equipped with a noise elimination circuit and a zero cross detection circuit. The zero cross detection circuit detects the point when the voltage level of an alternating waveform passes 0 V. The external interrupts can be controlled with the interrupt control registers I1 and I2.

Table 7 External interrupt activated conditions

Name	Input pin	Activated condition	Valid waveform selection bit
External 0 interrupt	D6/INT0/ZEROX	When the next waveform is input to D6/INT0/ZEROX pin • Falling waveform ("H"→"L") • Rising waveform ("L"→"H") • Both rising and falling waveforms	$I1_1$ $I1_2$
External 1 interrupt	$P2_1$/INT1	When the next waveform is input to $P2_1$/INT1 pin • Falling waveform ("H"→"L") • Rising waveform ("L"→"H")	$I2_3$

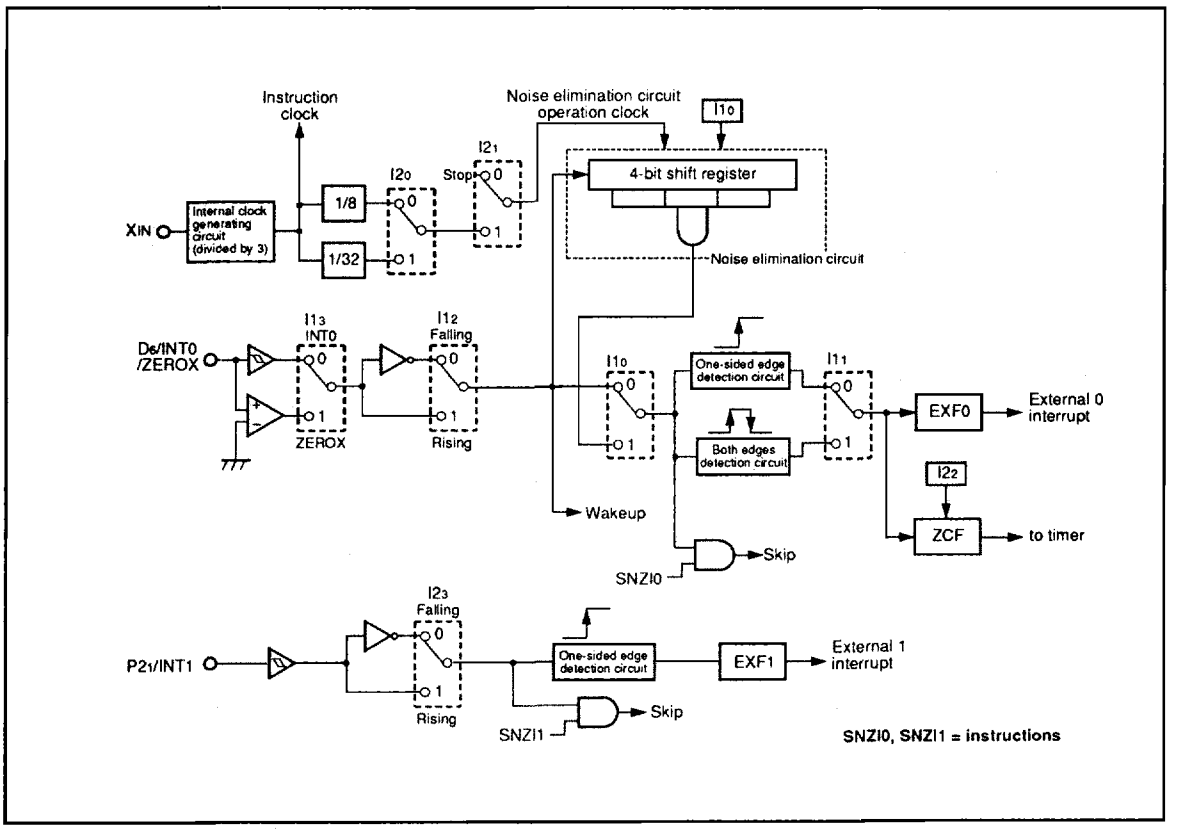


Fig. 17 External interrupt circuit structure

(1) External 0 interrupt request flag (EXF0)

External 0 interrupt request flag (EXF0) is set to "1" when a valid waveform is input to D_6/INT0/ZEROX pin.

External 0 interrupt is equipped with a noise elimination circuit and a zero cross detection circuit. 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.

The D_6/INT0/ZEROX pin need not be selected the external interrupt input INT0 function or the normal I/O port D_6 function. However, the EXF0 flag is set to "1" when a valid waveform is input even if it is used as an I/O port D_6.

● External 0 interrupt activated condition

External 0 interrupt activated condition is satisfied when a valid waveform is input to D_6/INT0/ZEROX 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 Clear the bits 0 and 3 of register I1 to "0," and select the valid waveform with the bits 1 and 2 of register I1.
② Clear the EXF0 flag to "0" with the SNZ0 instruction.
③ Set the NOP instruction for the case when a skip is performed with the SNZ0 instruction.
④ Set both the external 0 interrupt enable bit ($V1_0$) and the INTE flag to "1."

The external 0 interrupt is now enabled. Now when a valid waveform is input to the D_6/INT0/ZEROX pin, the EXF0 flag is set to "1" and the external 0 interrupt occurs.

(2) External 1 interrupt request flag (EXF1)

External 1 interrupt request flag (EXF1) is set to "1" when a valid waveform is input to $P2_1$/INT1 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 EXF1 flag can be examined with the skip instruction (SNZ1). Use the interrupt control register V1 to select the interrupt or the skip instruction. The EXF1 flag is cleared to "0" when an interrupt occurs or when the next instruction is skipped with the skip instruction.

The $P2_1$/INT1 pin need not be selected the external interrupt input INT1 function or the normal input port $P2_1$ function. However, the EXF1 flag is set to "1" when a valid waveform is input even if it is used as an input port $P2_1$.

● External 1 interrupt activated condition

External 1 interrupt activated condition is satisfied when a valid waveform is input to $P2_1$/INT1 pin.

The valid waveform can be selected from rising waveform or falling waveform. An example of how to use the external 1 interrupt is as follows.

① Select the valid waveform with the bit 3 of register I2.
② Clear the EXF1 flag to "0" with the SNZ1 instruction.
③ Set the NOP instruction for the case when a skip is performed with the SNZ1 instruction.
④ Set both the external 1 interrupt enable bit ($V1_1$) and the INTE flag to "1."

The external 1 interrupt is now enabled. Now when a valid waveform is input to the $P2_1$/INT1 pin, the EXF1 flag is set to "1" and the external 1 interrupt occurs.

(3) External interrupt control registers

● Interrupt control register I1

Register I1 controls the noise elimination circuit, valid waveform for the external 0 interrupt, return level (valid level of wakeup signal) from the RAM back-up mode, and D_6/INT0/ZEROX pin function. 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.

● Interrupt control register I2

Register I2 controls the sampling clock of noise elimination circuit, valid waveform for the external 1 interrupt, set/clear of zero cross input flag. Set the contents of this register through register A with the TI2A instruction. The TAI2 instruction can be used to transfer the contents of register I2 to register A.

Table 8 External interrupt control registers

Interrupt control register I1		at reset : 0000_2	at RAM back-up : state retained	R/W
$I1_3$	D_6/INT0/ZEROX pin function selection bit	0	INT0/D_6 (input/output)	
		1	ZEROX/D_6 (input/output)	
$I1_2$	Interrupt valid waveform for INT0 pin/ return level selection bit (Note 2)	0	Falling waveform ("L" level of INT0 pin is recognized with the SNZI0 instruction)/"L" level	
		1	Rising waveform ("H" level of INT0 pin is recognized with the SNZI0 instruction)/"H" level	
$I1_1$	Edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I1_0$	Noise elimination circuit control bit	0	Disabled	
		1	Enabled	
Interrupt control register I2		at reset : 0000_2	at RAM back-up : state retained	R/W
$I2_3$	Interrupt valid waveform for INT1 pin selection bit (Note 2)	0	Falling waveform ("L" level of INT1 pin is recognized with the SNZI1 instruction.)	
		1	Rising waveform ("H" level of INT1 pin is recognized with the SNZI1 instruction.)	
$I2_2$	Zero cross input flag control bit	0	Zero cross input flag cleared (set impossible)	
		1	Zero cross input flag set possible	
$I2_1$	Noise elimination circuit sampling clock control bit	0	Stop	
		1	Operating	
$I2_0$	Noise elimination circuit sampling clock selection bit	0	Instruction clock divided by 8	
		1	Instruction clock divided by 32	

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

2: When the contents of $I1_2$ and $I2_3$ are changed, the external interrupt request flags EXF0 and EXF1 may be set, respectively. Accordingly, clear the EXF0 and EXF1 flags to "0" with the SNZ0 and SNZ1 instructions, respectively.

(4) Noise elimination circuit

External 0 interrupt is equipped with a noise elimination circuit. This noise elimination circuit takes sampling four times with the noise elimination circuit sampling clock which is the instruction clock divided by 8 or 32 when the level of external 0 interrupt input is changed. As a result, if values of the noise elimination circuit 4-bit shift register become "1111" or "0000," the output of the noise elimination circuit changes "1" or "0," respectively. An example of how to use the noise elimination circuit is as follows.

① Set the bit 0 of register I1 to "1," select the valid waveform (= valid edge) with the bits 1 and 2 of register I1, and select the pin function with the bit 3 of register I1.

② Clear the EXF0 flag to "0" with the SNZ0 instruction.

③ Set the NOP instruction for the case when a skip is performed with the SNZ0 instruction.

④ Set both the external 0 interrupt enable bit ($V1_0$) and the INTE flag to "1."

⑤ Select the sampling clock of the noise elimination circuit with the bit 0 of register I2, and set the bit 1 of register I2 to "1."

The external 0 interrupt is now enabled. Now when a valid waveform is input to the D_6/INT0/ZEROX pin, the EXF0 flag is set to "1" and the external 0 interrupt occurs.

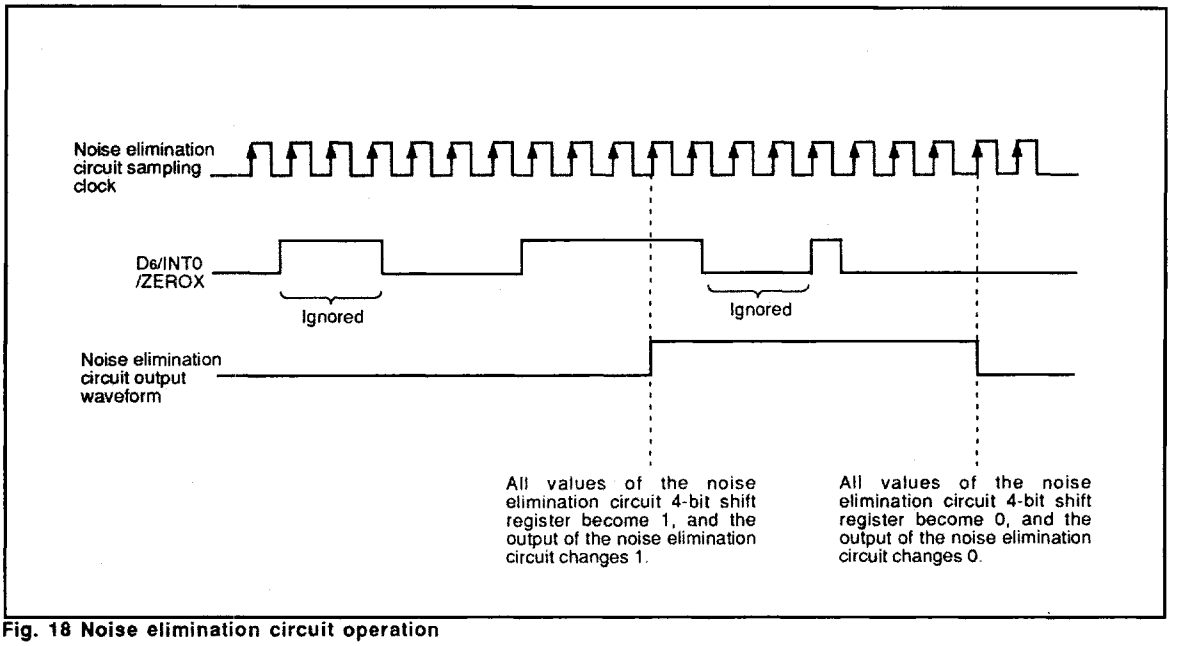


Fig. 18 Noise elimination circuit operation

ZERO CROSS DETECTION CIRCUIT

External 0 interrupt is equipped with a zero cross detection circuit which is used to detect the point when the voltage level of an alternating waveform passes 0 V (zero cross). Zero cross point to be detected can be selected when changing from minus to plus, plus to minus, or both.

The zero cross detection circuit can be used together with the noise elimination circuit.

In addition, the zero cross point which is detected by the zero cross detection circuit can be used as the start trigger for the timer 3 count operation (Refer to the zero cross input flag).

An example of how to use the zero cross detection circuit is as follows.

① Set the bit 3 of register I1 to "1" and select the valid waveform (= zero cross point to be detected) with bits 1 and 2.
② Clear the EXF0 flag to "0" with the SNZ0 instruction.
③ Set the NOP instruction for the case when a skip is performed with the SNZ0 instruction.
④ Set both the external 0 interrupt enable bit ($V1_0$) and the INTE flag to "1."

The external 0 interrupt is now enabled. Now when an alternating waveform is input to the D_6/INT0/ZEROX pin, zero cross point is detected, the EXF0 flag is set to "1," and the external 0 interrupt occurs.

The interval from zero cross is detected until program at the interrupt address is executed is 4 to 5 machine cycles (the delay time is added when using the noise elimination circuit).

(1) Zero cross input flag (ZCF)

The zero cross input flag (ZCF) is used to start timer 3 count operation by the zero cross point which is detected with the zero cross detection circuit.

The ZCF flag is set to "1" when the voltage level of an alternating waveform input to the D_6/INT0/ZEROX pin passes 0 V. When the bit 2 of timer control register W3 is set to "1," timer 3 starts counting when the ZCF flag is set to "1" and stops counting when it is cleared to "0."

The ZCF flag can be controlled by the bit 2 ($I2_2$) of register I2. It can be set (operation state) when $I2_2$ is set to "1." It cannot be set (stop state) when $I2_2$ is cleared to "0" because it is fixed to "0." Clear $I2_2$ to "0" to clear the ZCF flag. However, when it is to be used after clear, set $I2_2$ to "1" and return it to the operation state.

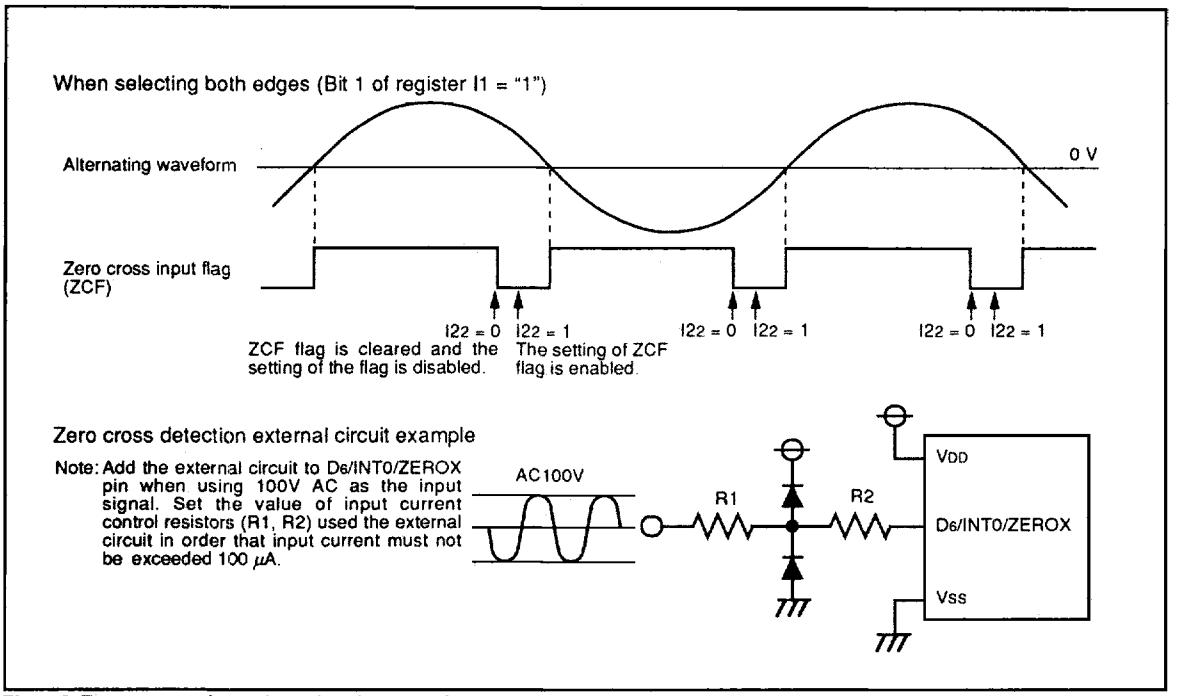


Fig. 19 Zero cross detection circuit operation

TIMERS

The 4510 Group has the programmable 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.

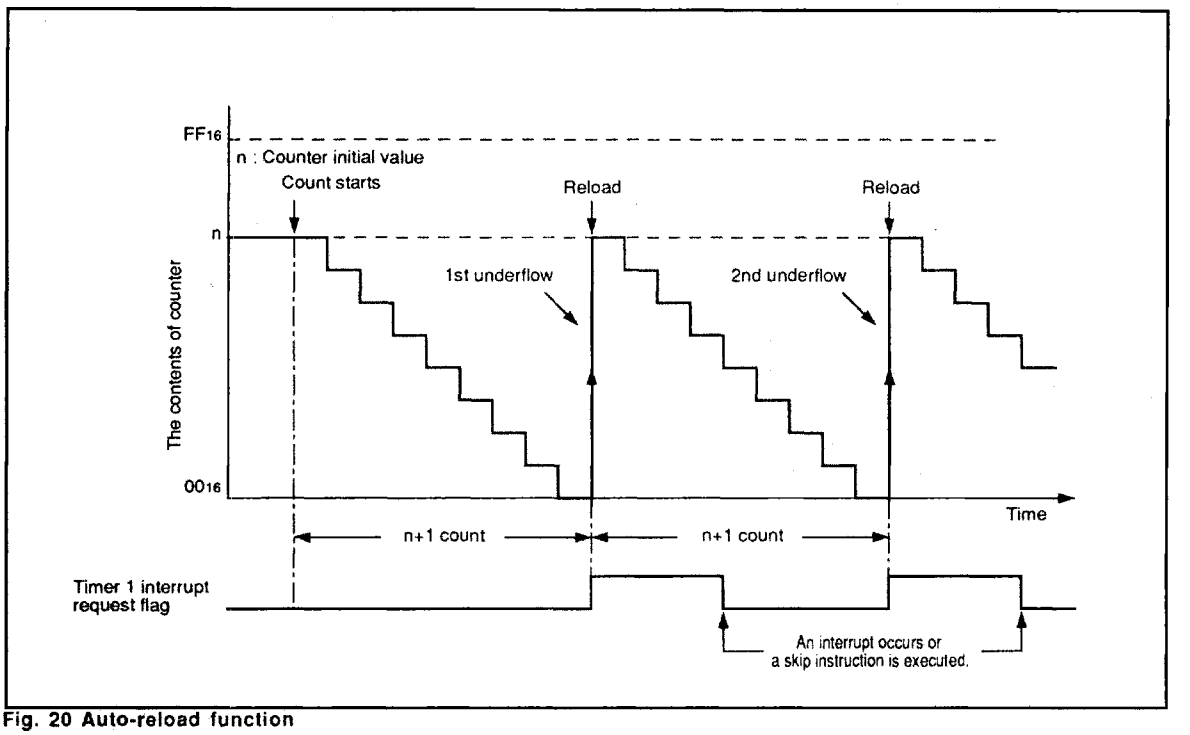


Fig. 20 Auto-reload function

The 4510 Group timer consists of the following circuits.
- Prescaler : frequency divider
- Timer 1 : 8-bit programmable timer
- Timer 2 : 8-bit programmable timer
- Timer 3 : 8-bit programmable timer
- Timer 4 : 8-bit programmable timer

(Timers 1 to 4 have the interrupt function, respectively)
- Watchdog timer

These timers can be controlled with the timer control registers W1 to W6. Each function is described below.

Table 9 Function related timers

Circuit	Structure	Count source	Frequency dividing ratio	Use of output signal	Control register
Prescaler	Frequency divider	• Instruction clock	2, 4	• Timer 1, 2, 3 and 4 count sources	W1
Timer 1	8-bit programmable binary down counter	• Prescaler output (ORCLK)	1 to 256	• Timer 2 count source • CNTR output • Timer 1 interrupt	W1 W6
Timer 2	8-bit programmable binary down counter	• Timer 1 underflow • Prescaler output (ORCLK) • CNTR input	1 to 256	• Timer 3 count source • Timer 2 interrupt	W2
Timer 3	8-bit programmable binary down counter (link to ZCF)	• Timer 2 underflow • Prescaler output (ORCLK)	1 to 256	• Timer 4 count source • Timer 3 interrupt	W3
Timer 4	8-bit programmable binary down counter	• Timer 3 underflow • Prescaler output (ORCLK) • CNTR input	1 to 256	• Real time output • Timer 4 interrupt	W4
Watchdog timer	16-bit fixed dividing frequency	• Instruction clock	65536	• System reset	

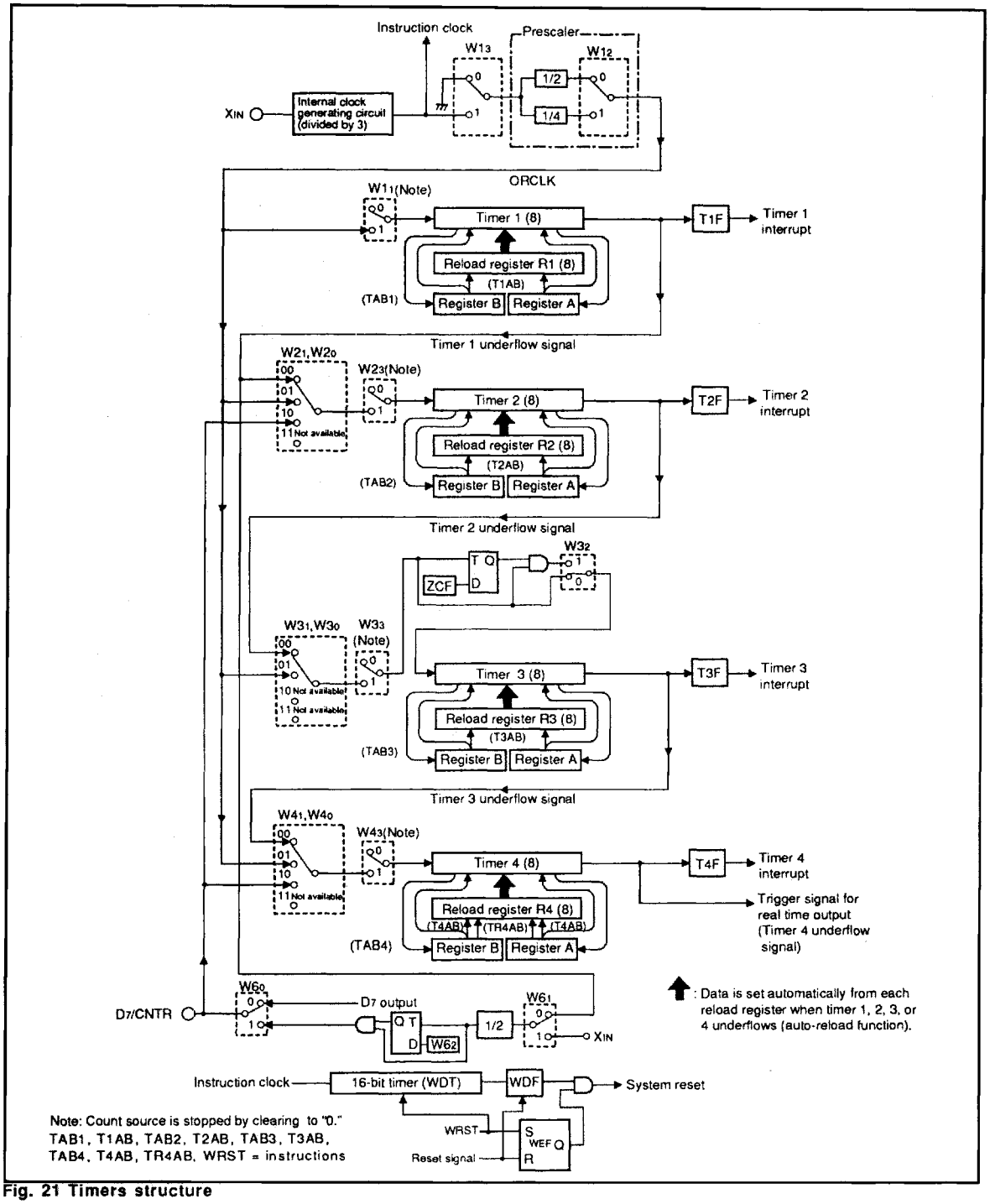


Fig. 21 Timers structure

Table 10 Timer control registers

Timer control register W1		at reset : 0000_2		at RAM back-up : 0000_2	R/W
$W1_3$	Prescaler control bit	0	Stop (state initialized)		
		1	Operating		
$W1_2$	Prescaler dividing ratio selection bit	0	Instruction clock divided by 2		
		1	Instruction clock divided by 4		
$W1_1$	Timer 1 control bit	0	Stop (state retained)		
		1	Operating		
$W1_0$	Not used	0	This bit has no function, but read/write is enabled.		
		1			

Timer control register W2		at reset : 0000_2		at RAM back-up : state retained	R/W
$W2_3$	Timer 2 control bit	0	Stop (state retained)		
		1	Operating		
$W2_2$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$W2_1$	Timer 2 count source selection bits	$W2_1$	$W2_0$	Count source	
$W2_0$		0	0	Timer 1 underflow signal	
		0	1	Prescaler output	
		1	0	CNTR input	
		1	1	Not available	

Timer control register W3		at reset : 0000_2		at RAM back-up : state retained	R/W
$W3_3$	Timer 3 control bit	0	Stop (state retained)		
		1	Operating		
$W3_2$	Timer 3 count start synchronous circuit control bit	0	Count start synchronous circuit not selected		
		1	Count start synchronous circuit selected		
$W3_1$	Timer 3 count source selection bits	$W3_1$	$W3_0$	Count source	
$W3_0$		0	0	Timer 2 underflow signal	
		0	1	Prescaler output	
		1	0	Not available	
		1	1	Not available	

Timer control register W4		at reset : 0000_2		at RAM back-up : state retained	R/W
$W4_3$	Timer 4 control bit	0	Stop (state retained)		
		1	Operating		
$W4_2$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$W4_1$	Timer 4 count source selection bits	$W4_1$	$W4_0$	Count source	
$W4_0$		0	0	Timer 3 underflow signal	
		0	1	Prescaler output	
		1	0	CNTR input	
		1	1	Not available	

Timer control register W6		at reset : 0000_2		at RAM back-up : state retained	R/W
$W6_3$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$W6_2$	CNTR output control bit	0	Output stop		
		1	Output		
$W6_1$	CNTR output selection bit	0	Timer 1 underflow signal divided by 2		
		1	X_{IN} divided by 2		
$W6_0$	D_7/CNTR function selection bit	0	D_7(I/O)/CNTR input		
		1	CNTR output/D_7(input)		

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

(1) Timer control registers

- Timer control register W1
  Register W1 controls the count operation of timer 1, and the frequency dividing ratio and count operation of prescaler. Set the contents of this register through register A with the TW1A instruction. The TAW1 instruction can be used to transfer the contents of register W1 to register A.
- Timer control register W2
  Register W2 controls the count operation and count source of timer 2. Set the contents of this register through register A with the TW2A instruction. The TAW2 instruction can be used to transfer the contents of register W2 to register A.
- Timer control register W3
  Register W3 controls the count operation and count source of timer 3 and the selection of count start synchronous circuit. Set the contents of this register through register A with the TW3A instruction. The TAW3 instruction can be used to transfer the contents of register W3 to register A.
- Timer control register W4
  Register W4 controls the count operation and count source of timer 4. Set the contents of this register through register A with the TW4A instruction. The TAW4 instruction can be used to transfer the contents of register W4 to register A.
- Timer control register W6
  Register W6 controls the D_7/CNTR pin function, the selection and operation of the CNTR. 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) Precautions

Note the following for the use of timers.

- Prescaler
  Stop the prescaler operation to change its frequency dividing ratio.
- Count source
  Stop timer 1, 2, 3, or 4 counting to change its count source.
- Reading the count value
  Stop timer 1, 2, 3, or 4 counting and then execute the TAB1, TAB2, TAB3, or TAB4 instruction to read its data.

(3) Prescaler

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

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

(4) Timer 1 (interrupt function)

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

Timer 1 starts counting after the following process;

① set data in timer 1, and

② set the bit 1 of register W1 to "1."

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

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

Data can be read from timer 1 with the TAB1 instruction. When reading the data, stop the counter and then execute the TAB1 instruction. Timer 1 underflow signal divided by 2 can be output from D_7/CNTR pin.

(5) Timer 2 (interrupt function)

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

Timer 2 starts counting after the following process;

① set data in timer 2,

② select the count source with the bits 0 and 1 of register W2, and

③ set the bit 3 of register W2 to "1."

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

When a value set in reload register R2 is n, timer 2 divides the count source signal by n + 1 (n = 0 to 255).

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

(6) Timer 3 (interrupt function)

Timer 3 is an 8-bit binary down counter with the timer 3 reload register (R3). Data can be set simultaneously in timer 3 and the reload register (R3) with the T3AB instruction.

Timer 3 starts counting after the following process;

① set data in timer 3,

② select the count source with the bits 0 and 1 of register W3, and

③ set the bit 3 of register W3 to "1."

However, the ZCF flag can be used as a start trigger for timer 3 count operation by setting the bit 2 of register W3 to "1." (refer to the external interrupt).

Once count is started, when timer 3 underflows (the next count pulse is input after the contents of timer 3 becomes "0"), the timer 3 interrupt request flag (T3F) is set to "1," new data is loaded from reload register R3, and count continues (auto-reload function).

When a value set in reload register R3 is n, timer 3 divides the count source signal by n + 1 (n = 0 to 255).

Data can be read from timer 3 with the TAB3 instruction. When reading the data, stop the counter and then execute the TAB3 instruction.

(7) Timer 4 (interrupt function)

Timer 4 is an 8-bit binary down counter with the timer 4 reload register (R4). Data can be set simultaneously in timer 4 and the reload register (R4) with the T4AB instruction.

Data can be written to reload register (R4) with the TR4AB instruction.

When writing data to reload register R4 with the TR4AB instruction, the downcount after the underflow is started from the setting value of reload register R4.

Timer 4 starts counting after the following process;

① set data in timer 4,

② select the count source with the bits 0 and 1 of register W4, and

③ set the bit 3 of register W4 to "1."

Once count is started, when timer 4 underflows (the next count pulse is input after the contents of timer 4 becomes "0"), the timer 4 interrupt request flag (T4F) is set to "1," new data is loaded from reload register R4, and count continues (auto-reload function).

When a value set in reload register R4 is n, timer 4 divides the count source signal by n + 1 (n = 0 to 255).

Data can be read from timer 4 with the TAB4 instruction. When reading the data, stop the counter and then execute the TAB4 instruction.

Timer 4 underflow signal is also used as the trigger signal for the real timer output.

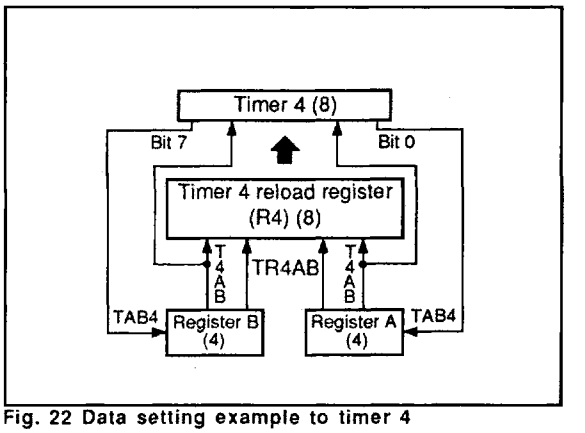


Fig. 22 Data setting example to timer 4

(8) Timer I/O pin (D_7/CNTR)

Timer I/O pin (D_7/CNTR) has functions to input the timer 2 and 4 count sources, and to output the timer 1 underflow signal and the $f(X_{IN})$ signal divided by 2. The selection of D_7/CNTR pin function can be controlled with the bit 0 of register W6. The timer 1 underflow signal divided by 2 or the $f(X_{IN})$ signal divided by 2 can be selected as the CNTR output signal with the bit 1 of register W6. CNTR output can be controlled with the bit 2 of register W6.

Timers 2 and 4 count the rising waveform of CNTR input when the CNTR input is selected as the count source.

(9) Timer interrupt request flags (T1F, T2F, T3F, and T4F)

Each timer interrupt request flag is set to "1" when the each timer underflows. The state of these flags can be examined with the skip instructions (SNZT1, SNZT2, SNZT3, and SNZT4).

Use the interrupt control registers 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.

WATCHDOG TIMER

Watchdog timer provides a method to reset the system when a program runs wild. Watchdog timer consists of timer (WDT), watchdog timer enable flag (WEF), and watchdog timer flag (WDF).

When the WRST instruction is executed after system is released from reset, the WEF flag is set to "1" and the timer WDT is set to "FFFF16." At this time, the watchdog timer starts operating.

The timer WDT downcounts the instruction clocks as the count source. The underflow signal occurs when the count value reaches "000016." On generation of this underflow signal, the WDF flag is set to "1" and the $\overline{\text{RESET}}$ pin outputs "L" level to reset the microcomputer. Execute the WRST instruction before the timer WDT underflows by a program when using watchdog timer to keep the microcomputer operating normally.

To prevent the WDT stopping in the event of misoperation, it is designed not to stop once the WRST instruction has been executed. Note also that, if the WRST instruction is never executed, the watchdog timer does not start.

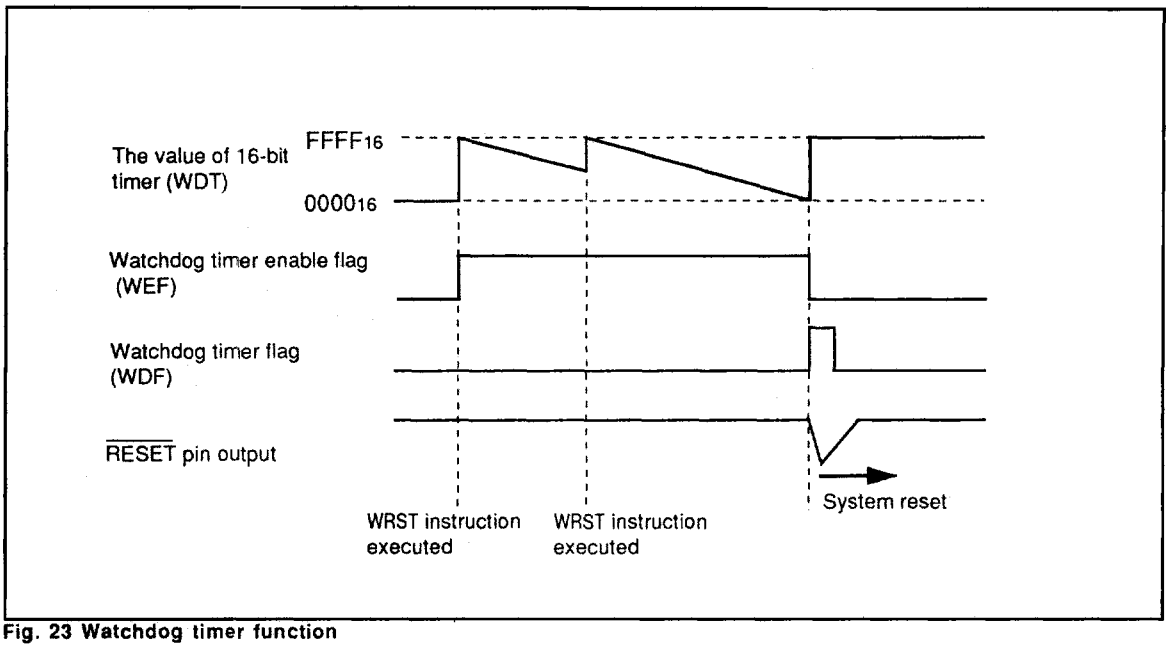


Fig. 23 Watchdog timer function

The contents of the WEF flag, the WDF flag and the timer WDT are initialized at the RAM back-up mode.

If the WDF flag is set to "1" at the same time that the microcomputer enters the RAM back-up state, system reset may be performed.

When using the watchdog timer and the RAM back-up mode, initialize the WDF flag with the WRST instruction just before the microcomputer enters the RAM back-up state (refer to Figure 24)

```
      ⋮
WRST      ; 16-bit timer (WDT) reset
EPOF      ; POF instruction enabled
POF
  ↓
Oscillation stop   (RAM back-up state)
```

Fig. 24 Program example to enter the RAM back-up mode when using the watchdog timer

REAL TIME OUTPUT

The real time output function uses timer 4 underflow as the trigger to output the contents of the real time output register RTP from the real time output pin.
Each time a timer 4 underflow occurs, the data set in the register RTP is transferred to the real time output latch and at the same time is output from the real time output pin. D_4/RTP and D_5/RTP can be used for real time output. An example of how to use the real time output is shown below.

① Set the function of D_4/RTP and D_5/RTP to real time output with the real time output control register RTR.
② Set the initial output value in real time output latch. (The real time output latch can be set to "1" with the RTPS instruction and cleared to "0" with the RTPR instruction.
③ Set data in the register RTP. (Set the data in the register RTP through register A with the TRTPA instruction.)
④ Select the timer 4 count source with the bits 0 and 1 of register W4.
⑤ Set data in timer 4 and reload register R4, and then start timer 4 count operation with the bit 3 of register W4.

Now the contents of the register RTP is output from the real time output pin each time the timer 4 underflows.

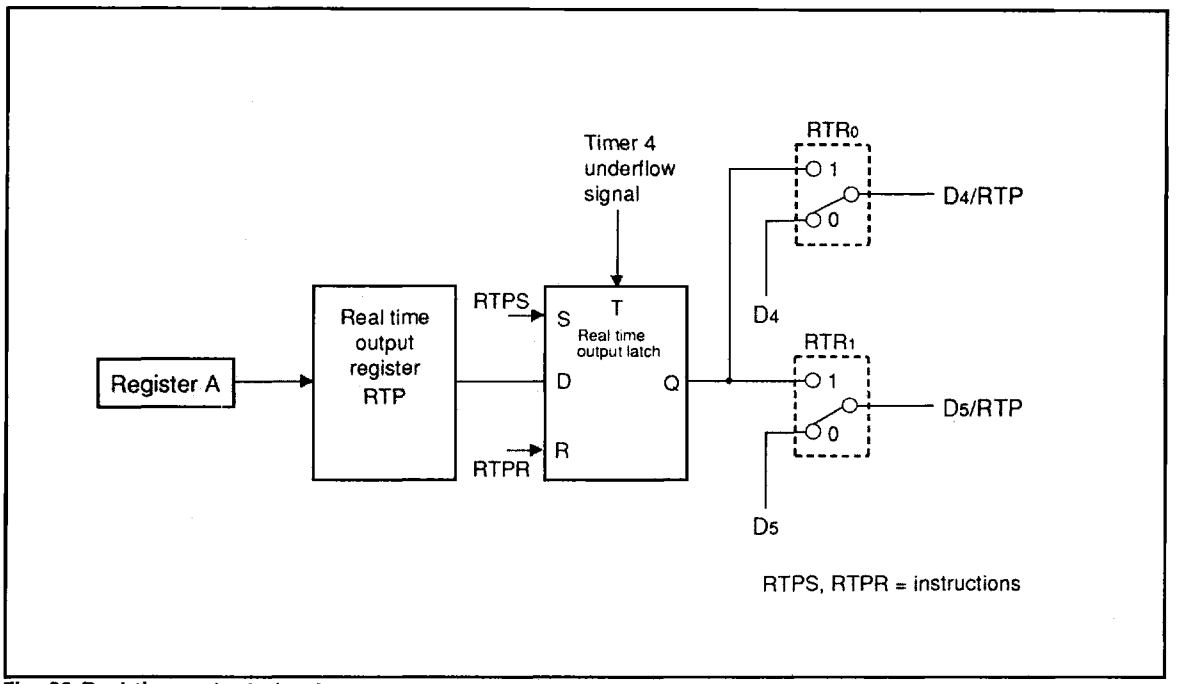
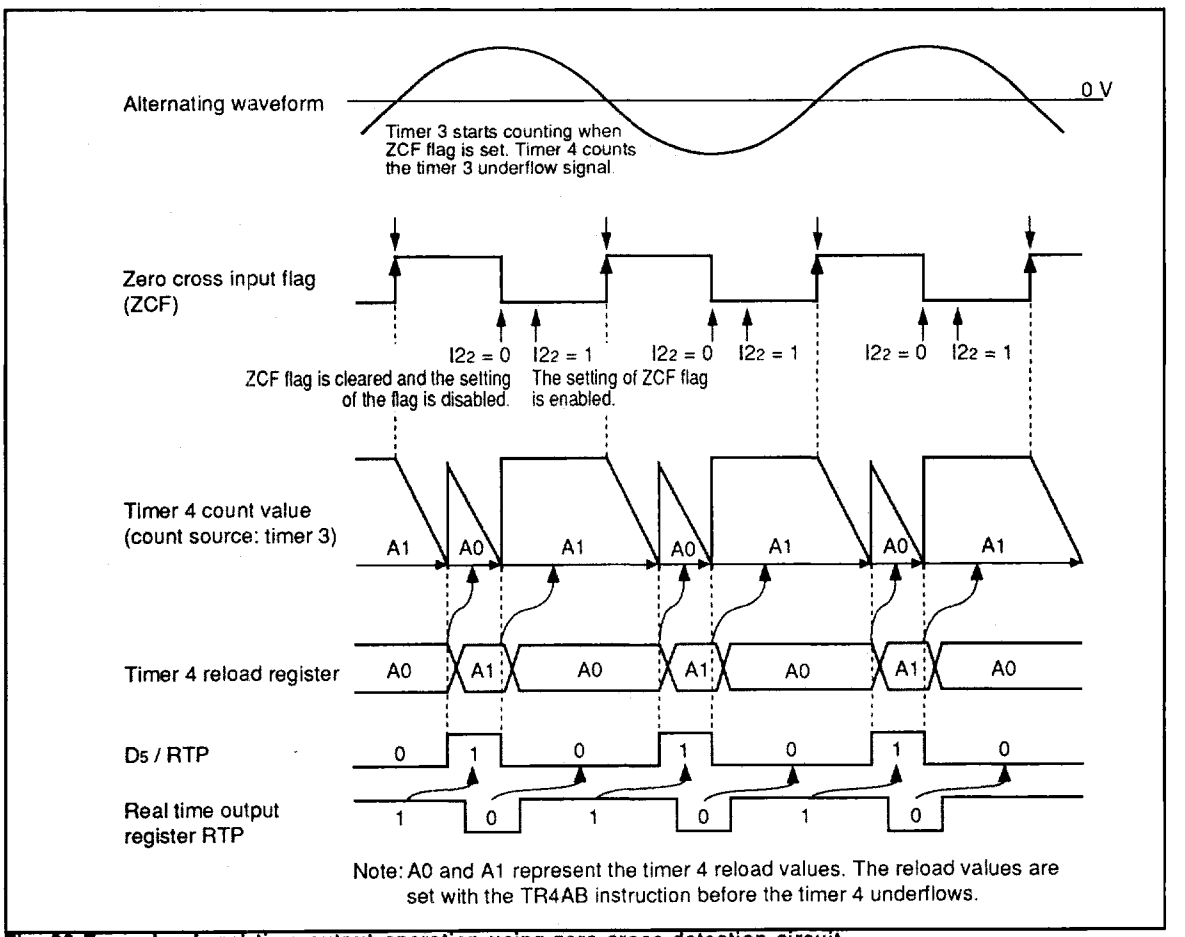


Fig. 25 Real time output structure

Real time output can be used together with the zero cross detection circuit. An example of how to use the real time output using the zero cross detection circuit is shown below.

① Repeat steps ① to ③ described above.
② Set $W4_1$ = "0," $W4_0$ = "0"
(timer 4 count source = timer 3 underflow signal)
③ Set $I2_2$ = "0," $W3_2$ = "1"
(ZCF flag is cleared and its setting is disabled. Timer 3 starts counting when the ZCF flag is set to "1" by selecting the timer 3 count start synchronous circuit.)
④ Set $I1_0$ = "1," $I1_3$ = "1"
(the noise elimination circuit is used and the ZEROX function is selected)
Select the valid waveform (= zero cross point to be detected) with $I1_1$ and $I1_2$
⑤ Set data in timer 3 and reload register R3, and timer 4 and reload register R4
⑥ Set $W3_3$ = "1," $W4_3$ = "1"
(timer 3 and timer 4 count operation start)
⑦ Set $I2_2$ = "1"
(the setting of ZCF flag is enabled)

Now the zero cross of alternating waveform input to D_8/INT0/ZEROX pin is detected and timer 3 count operation starts. Timer 4 counts the timer 3 underflow signal and the contents of the real time output latch is output from the real time output pin each time timer 4 underflows.



Note: A0 and A1 represent the timer 4 reload values. The reload values are set with the TR4AB instruction before the timer 4 underflows.

Fig. 26 Example of real time output operation using zero cross detection circuit

SERIAL I/O

The 4510 Group has a built-in clock synchronous serial I/O which can serially transmit or receive 8-bit data.
Serial I/O consists of;

- serial I/O register SI
- serial I/O mode register J1
- serial I/O transmission/reception completion flag (SIOF)
- serial I/O counter

Registers A and B are used to perform data transfer with internal CPU, and the serial I/O pins are used for external data transfer.
The pin functions of the serial I/O pins can be set with the register J1.

Table 11 Serial I/O pins

Pin	Pin function when selecting serial I/O
D_8/SCK	Clock I/O (SCK)
D_9/SOUT	Serial data output (SOUT)
$P2_0$/SIN	Serial data input (SIN)

Note: Input port $P2_0$ can be used regardless of register J1.

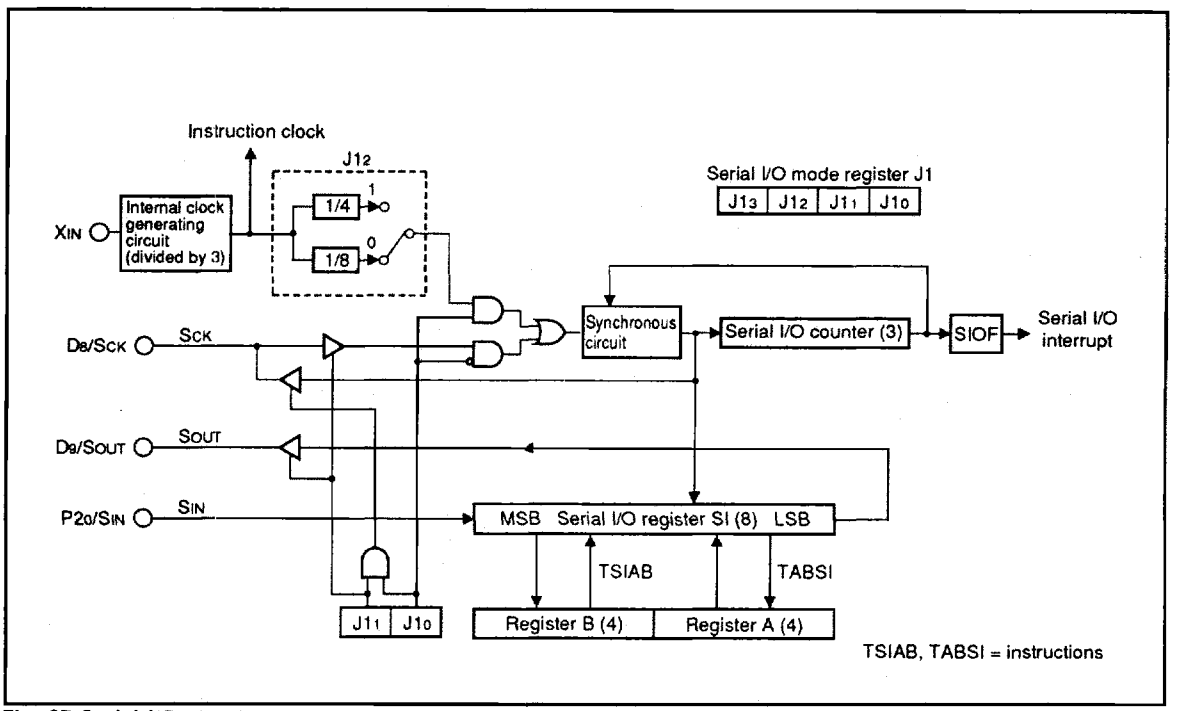


Fig. 27 Serial I/O structure

Table 12 Serial I/O mode register

Serial I/O mode register J1		at reset : 0000_2	at RAM back-up : state retained	R/W
$J1_3$	Not used	0 / 1	This bit has no function, but read/write is enabled.	
$J1_2$	Serial I/O internal clock dividing ratio selection bit	0	Instruction clock signal divided by 8	
		1	Instruction clock signal divided by 4	
$J1_1$	Serial I/O port selection bit	0	I/O ports D_8, D_9, and input port $P2_0$ selected	
		1	Serial I/O ports SCK, SOUT, and SIN/input ports D_8, D_9 and $P2_0$ selected	
$J1_0$	Serial I/O synchronous clock selection bit	0	External clock	
		1	Internal clock (instruction clock divided by 4 or 8)	

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

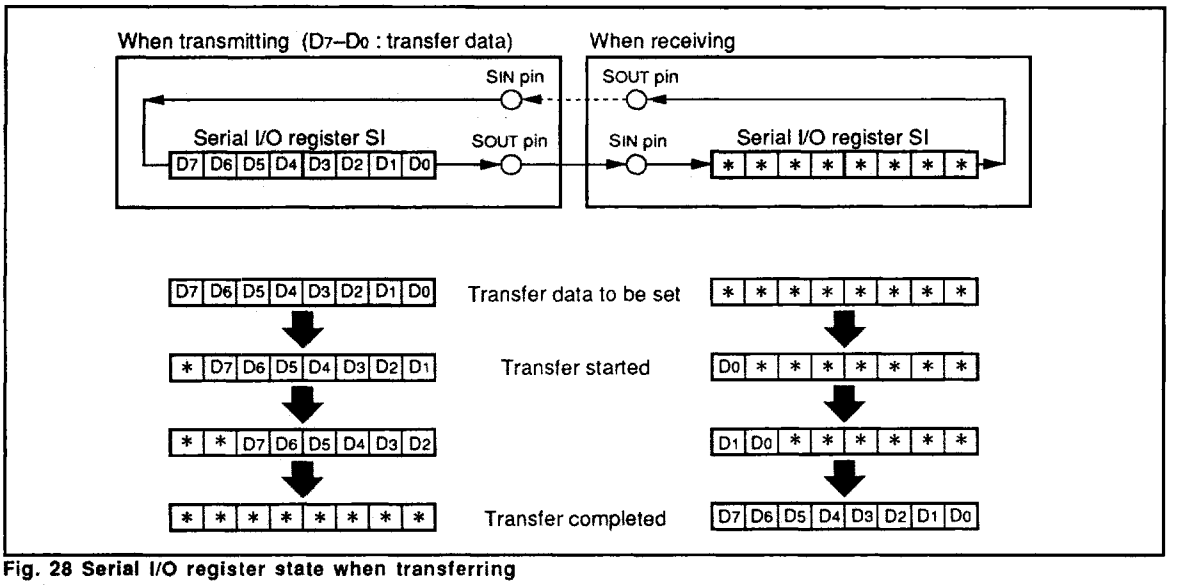


Fig. 28 Serial I/O register state when transferring

(1) Serial I/O register SI

Serial I/O 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 work registers without using serial I/O, pull up the S_{CK} pin or set the pin function to an I/O port D_8.

(2) Serial I/O transmission/reception completion flag (SIOF)

Serial I/O transmission/reception 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 I/O start instruction (SST)

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

(4) Serial I/O mode register J1

Register J1 controls the synchronous clock, D_8/S_{CK} and D_9/S_{OUT} 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 I/O

Figure 29 shows the serial I/O connection example. Serial I/O interrupt is not used in this example. In the actual wiring, pull up the wiring between each pin with a resistor. Figure 30 shows the data transfer timing and Table 13 shows the data transfer sequence.

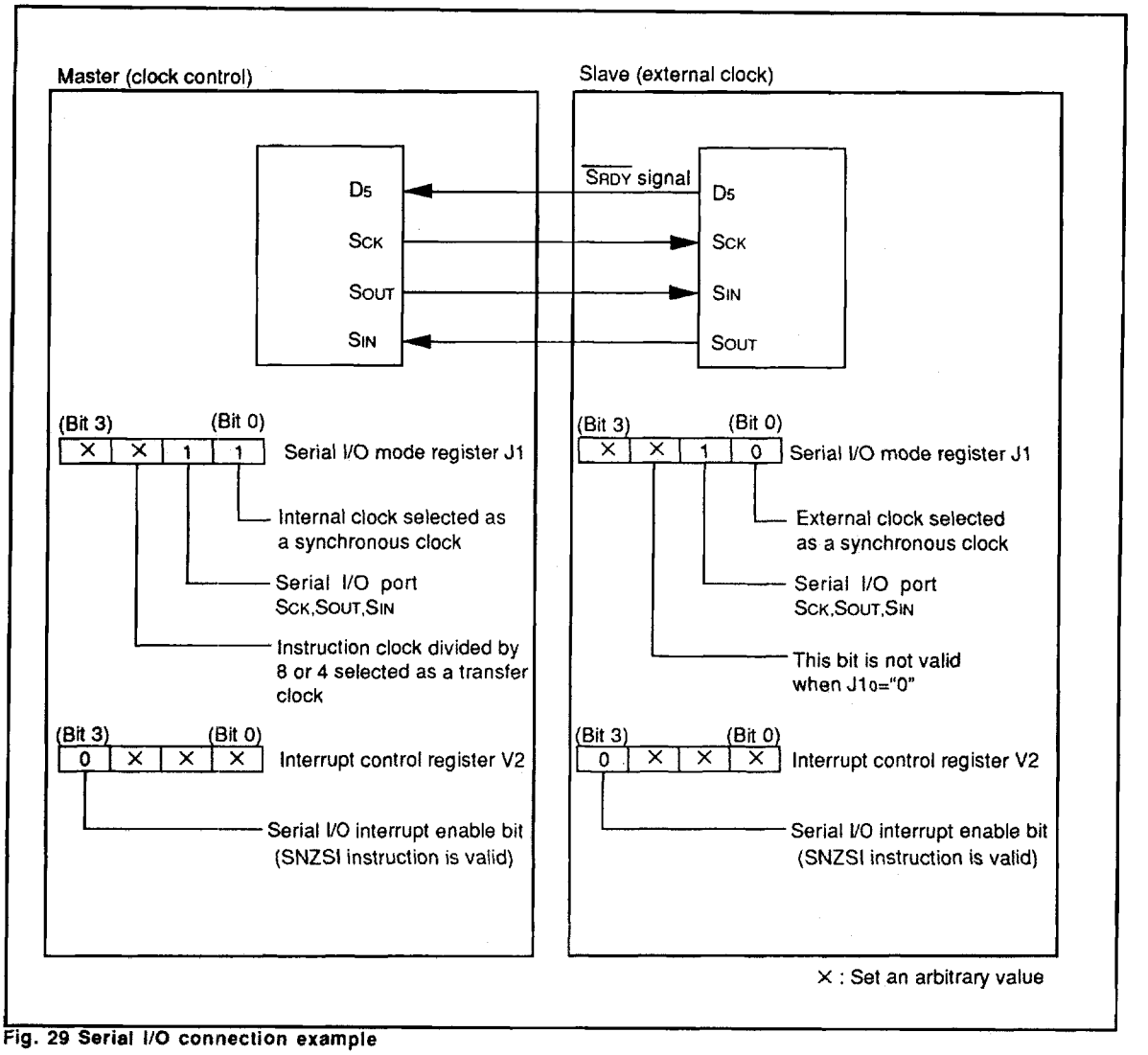
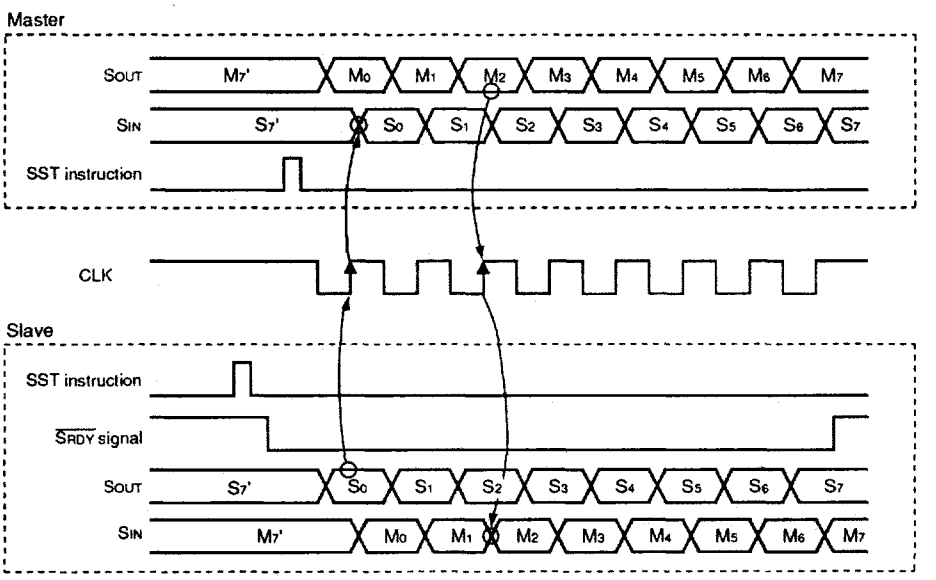


Fig. 29 Serial I/O connection example



M0–M7 : the contents of master serial I/O register S0–S7 : the contents of slave serial I/O register
Rising of CLK : serial input Falling of CLK : serial output

Fig. 30 Timing of serial I/O data transfer

Table 13 Processing sequence of data transfer from master to slave

Master (transmission)	Slave (reception)
[Initial setting] •Setting the serial I/O mode register J1 and interrupt control register V2 shown in Figure 29.	[Initial setting] •Setting serial I/O mode register J1, and interrupt control register V2 shown in Figure 29.
TJ1A and TV2A instructions	TJ1A and TV2A instructions
•Setting the port received the reception enable signal ($\overline{SRDY}$) to the input mode. (Port D_5 is used in this example)	•Setting the port transmitted the reception enable signal ($\overline{SRDY}$) and outputting "H" level (reception impossible). (Port D_5 is used in this example)
SD instruction	SD instruction
*[Transmission enable state] •Storing transmission data to serial I/O register SI.	*[Reception enable state] •The SIOF flag is cleared to "0."
TSIAB instruction	SST instruction
	•"L" level (reception possible) is output from port D_5.
	RD instruction
[Transmission] •Check port D_5 is "L" level.	[Reception]
SZD instruction	
•Serial transfer starts.	
SST instruction	
•Check transmission completes.	•Check reception completes.
SNZSI instruction	SNZSI instruction
•Wait (timing when continuously transferring)	•"H" level is output from port D_5.
	SD instruction
	[Data processing]

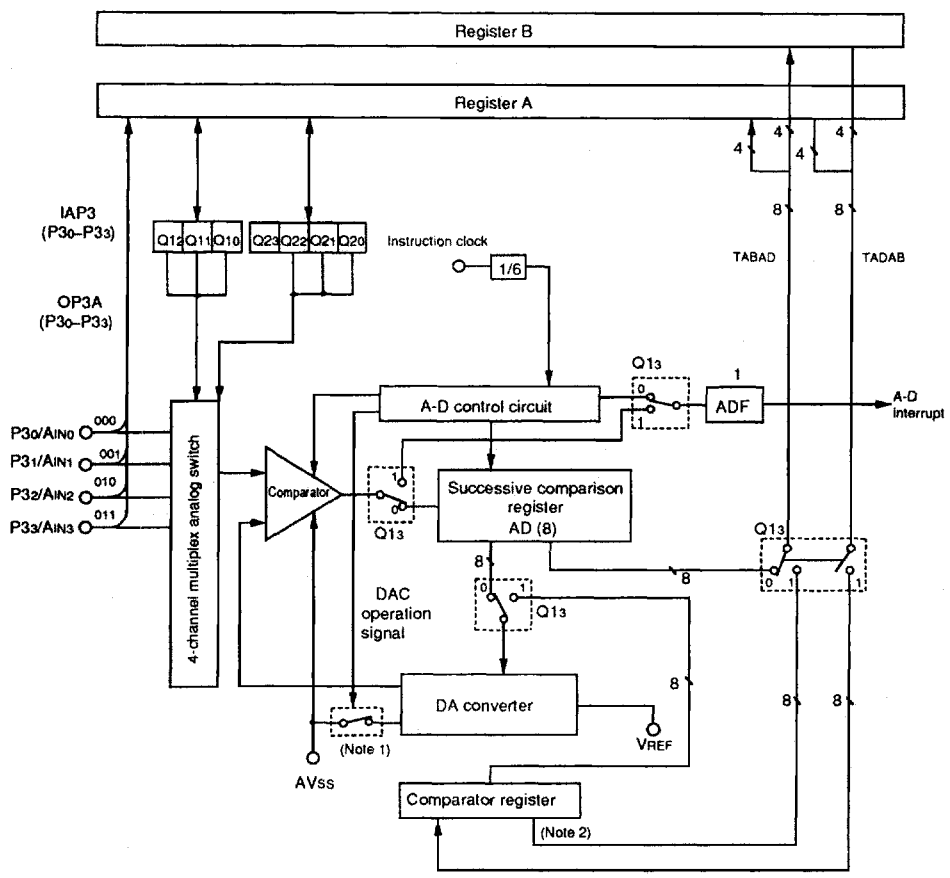
1-byte data is serially transferred on this process. Subsequently, data can be transferred continuously by repeating the process from *.
When an external clock is selected as a synchronous clock, the clock is not controlled internally. Control the clock externally because serial transfer is performed 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."

A-D CONVERTER, COMPARATOR

The 4510 Group has a built-in A-D conversion circuit that performs conversion by 8-bit successive comparison method. Table 14 shows the characteristics of this A-D converter. This A-D converter can also be used as a comparator to compare analog voltages input from the analog input pin with preset values.

Table 14 A-D converter function

Parameter	Characteristics
Conversion format	Successive comparison method
Resolution	8 bits
Absolute accuracy	±2LSB
Conversion speed	25 μs (at 6.0 MHz system clock frequency)
Analog input pin	4 (selected from AIN0–AIN3)



Notes 1: This switch is turned on only when operating A-D converter, and the comparison voltage is generated.
2: The read/write to the comparator register is enabled only at the comparator mode (Q13=1).
The contents of comparator register is retained even when changing to the A-D conversion mode (Q13=0) because the comparator register is separated from the successive comparison register AD.
IAP3, OP3A, TABAD, TADAB = instructions

Fig. 31 A-D conversion circuit structure

Table 15 A-D control registers

A-D control register Q1		at reset : 0000_2			at RAM back-up : state retained	R/W
$Q1_3$	A-D operating mode control bit	0			A-D conversion mode	
		1			Comparator mode	
$Q1_2$	Analog input pin selection bits (Note 2)	$Q1_2$	$Q1_1$	$Q1_0$	Selected pins	
		0	0	0	A_{IN0}	
		0	0	1	A_{IN1}	
$Q1_1$		0	1	0	A_{IN2}	
		0	1	1	A_{IN3}	
		1	0	0	Not available	
$Q1_0$		1	0	1	Not available	
		1	1	0	Not available	
		1	1	1	Not available	
A-D control register Q2		at reset : 0000_2			at RAM back-up : state retained	R/W
$Q2_3$	Not used	0			This bit has no function, but read/write is enabled.	
		1				
$Q2_2$	$P3_3/A_{IN3}$ and $P3_2/A_{IN2}$ pin function selection bit	0			$P3_3$, $P3_2$ (I/O)	
		1			A_{IN3}, $A_{IN2}/P3_3$, $P3_2$ (output)	
$Q2_1$	$P3_1/A_{IN1}$ pin function selection bit	0			$P3_1$ (I/O)	
		1			$A_{IN1}/P3_1$ (output)	
$Q2_0$	$P3_0/A_{IN0}$ pin function selection bit	0			$P3_0$ (I/O)	
		1			$A_{IN0}/P3_0$ (output)	

Notes 1: "R" represents read enabled, "W" represents write enabled.
2: Select A_{IN0}–A_{IN3} with register Q1 after setting register Q2.

(1) Operating at A-D conversion mode

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

(2) Successive comparison register AD

Register AD stores the A-D conversion result of an analog input in 8-bit digital data format. The contents of this register can be stored in register B (high-order 4 bits) and register A (low-order 4 bits) with the TABAD instruction. However, do not execute this instruction during A-D conversion.

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

Logic value of comparison voltage V_{ref}

When n = 0 $V_{ref} = 0$

When n = 1 to 255 $V_{ref} = \frac{V_{REF}}{256} \times (n - 0.5)$

n: The value of register AD (decimal expression)

(3) A-D conversion completion flag (ADF)

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

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

(4) A-D conversion start instruction (ADST)

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

(5) A-D control register Q1

Register Q1 is used to select one of the 4 analog input pins. After set the pin function with the register Q2, select the analog input with register Q1.

(6) A-D control register Q2

Register Q2 is used to select the pin function of $P3_0/A_{IN0}$, $P3_1/A_{IN1}$, $P3_2/A_{IN2}$, and $P3_3/A_{IN3}$. After set this register, select the analog input with register Q1.

Even when register Q2 is used to set the pins for analog input, they continue to function as $P3_0$–$P3_3$ outputs. Accordingly, when using the output function of a pin that is not set for analog input, make sure to set the outputs of pins that are set for analog input to "1."

(7) 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 A-D conversion starts, the register AD is cleared to "00_{16}."
② Next, the topmost bit of the register AD is set to "1," and the comparison voltage V_{ref} is compared with the analog input voltage V_{IN}.
③ When the comparison result is $V_{ref} < V_{IN}$, the topmost bit of the register AD remains set to "1." When the comparison result is $V_{ref} > V_{IN}$, it is cleared to "0."

The 4510 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 50 machine cycles (25 μs when f(X_{IN}) = 6.0 MHz) 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 16 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 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{512}$
2nd comparison	*1 1 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{512}$
3rd comparison	*1 *2 1 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{512}$
After 8th comparison completes	A-D conversion result: *1 *2 *3 *4 *5 *6 *7 *8	

*1: 1st comparison result *2: 2nd comparison result
*3: 3rd comparison result *4: 4th comparison result
*5: 5th comparison result *6: 6th comparison result
*7: 7th comparison result *8: 8th comparison result

(8) A-D conversion timing chart

Figure 32 shows the A-D conversion timing chart.

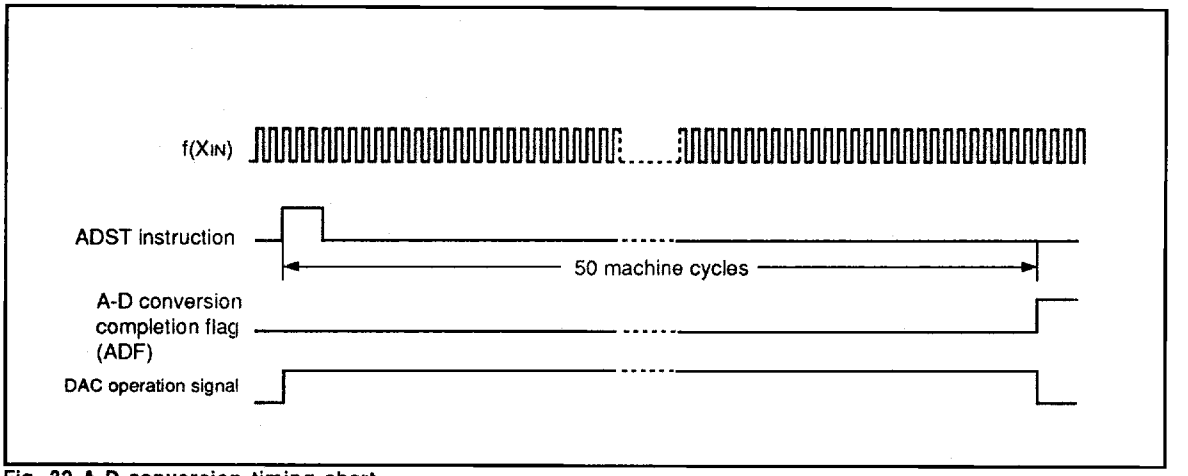


Fig. 32 A-D conversion timing chart

(9) How to use A-D conversion

How to use A-D conversion is explained using as example in which the analog input from P30/AIN0 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), and the low-order 4 bits in address M(Z, X, Y) = (0, 0, 1) of RAM. The A-D interrupt is not used in this example.

① After selecting the P30/AIN0 pin function with the bit 0 of the register Q2, select P30/AIN0 pin and A-D conversion mode with the register Q1.
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 converted data to registers A and B (TABAD instruction).
⑤ Transfer the contents of register A to M (Z, X, Y) = (0, 0, 1).
⑥ Transfer the contents of register B to M (Z, X, Y) = (0, 0, 0) through register A.

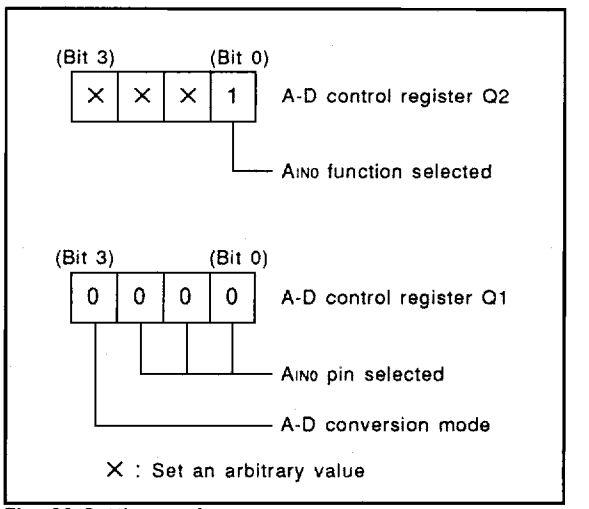


Fig. 33 Setting registers

(10) 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.

(11) Comparator register

In comparator mode, the built-in DA converter is connected to the 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 register AD 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}

When n = 0 $V_{ref} = 0$

When n = 1 to 255 $V_{ref} = \frac{V_{REF}}{256} \times (n - 0.5)$

n: The value of register AD (decimal expression)

(12) 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.

(13) 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 (4 μs at f(XIN) = 6.0 MHz). When the analog input voltage is lower than the comparison voltage, the ADF flag is set to "1."

(14) Notes for the use of A-D conversion

Note the following when using the analog input pins also for I/O port P3 functions:

•P3 output

Even when pins P30/AIN0, P31/AIN1, P32/AIN2, and P33/AIN3 are set for analog input, they also function as port P3 output. When executing the OP3A instruction, set the outputs of pins set for analog input to "1."

•P3 input

Pins that are set for analog input do not function as port P3 input. After executing the IAP3 instruction, the values of bits corresponding to pins set for analog input are transmitted to register A as "0."

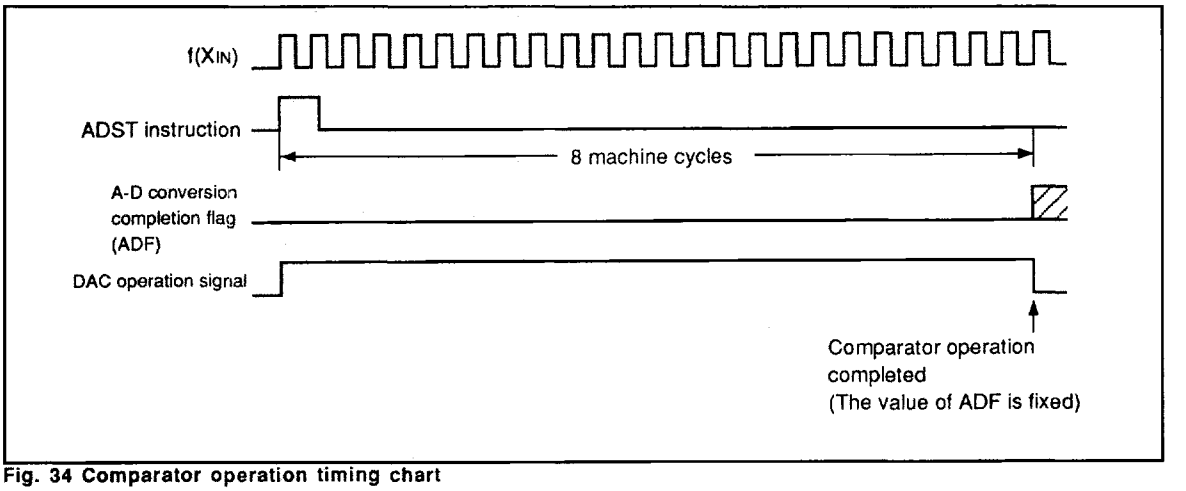


Fig. 34 Comparator operation timing chart

RESET FUNCTION

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

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

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

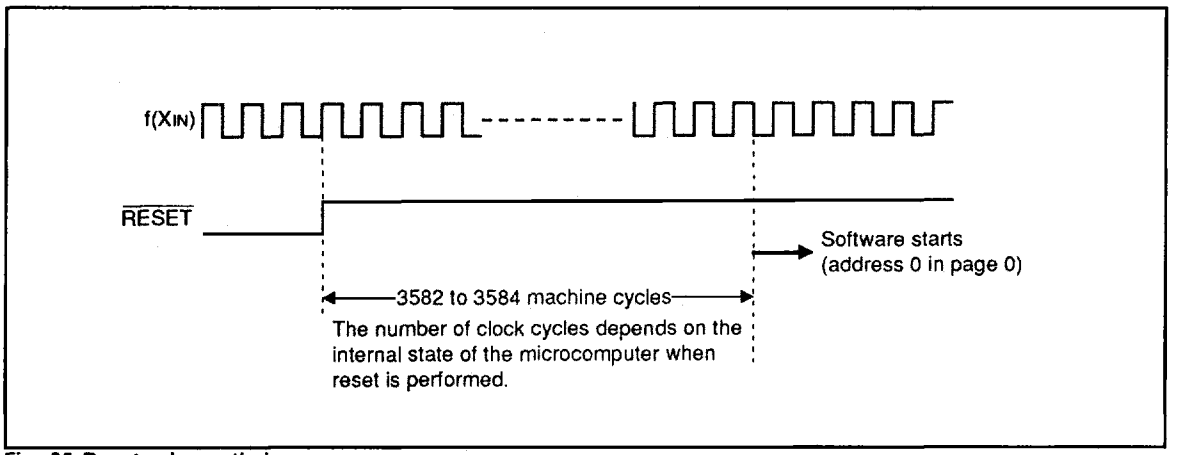


Fig. 35 Reset release timing

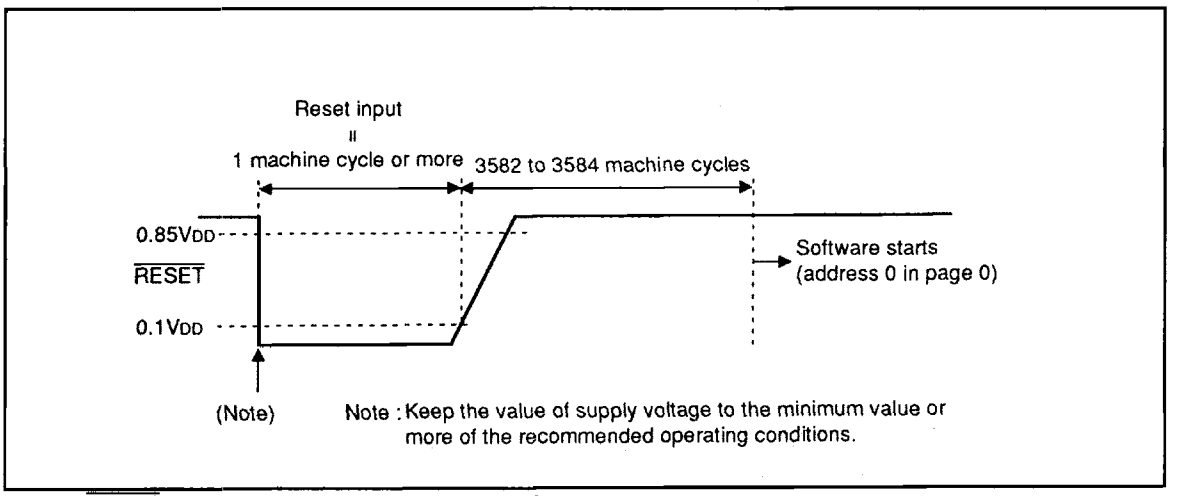


Fig. 36 $\overline{\text{RESET}}$ pin input waveform and reset operation

(1) Power-on reset

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

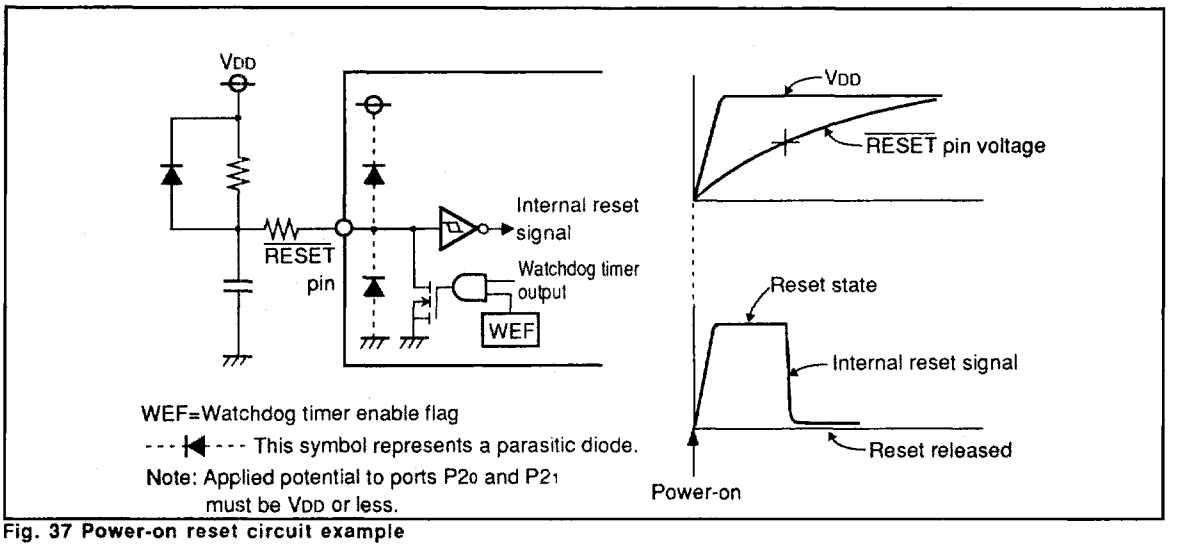


Fig. 37 Power-on reset circuit example

(2) Internal state at reset

Table 17 shows port state at reset, and Figure 38 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 38 are undefined, so set the initial value to them.

Table 17 Port state at reset

Name	Function	State
D0–D3	D0–D3	High impedance (Note)
D4/RTP, D5/RTP D6/CNTR D7/INT0/ZEROX D8/SCK, D9/SOUT	D4–D9	
P00–P03	P00–P03	
P10–P13	P10–P13	
P20/SIN, P21/INT1	P20, P21	High impedance
P30/AIN0–P33/AIN3	P30–P33	High impedance (Note)

Note: Output latch is set to "1."

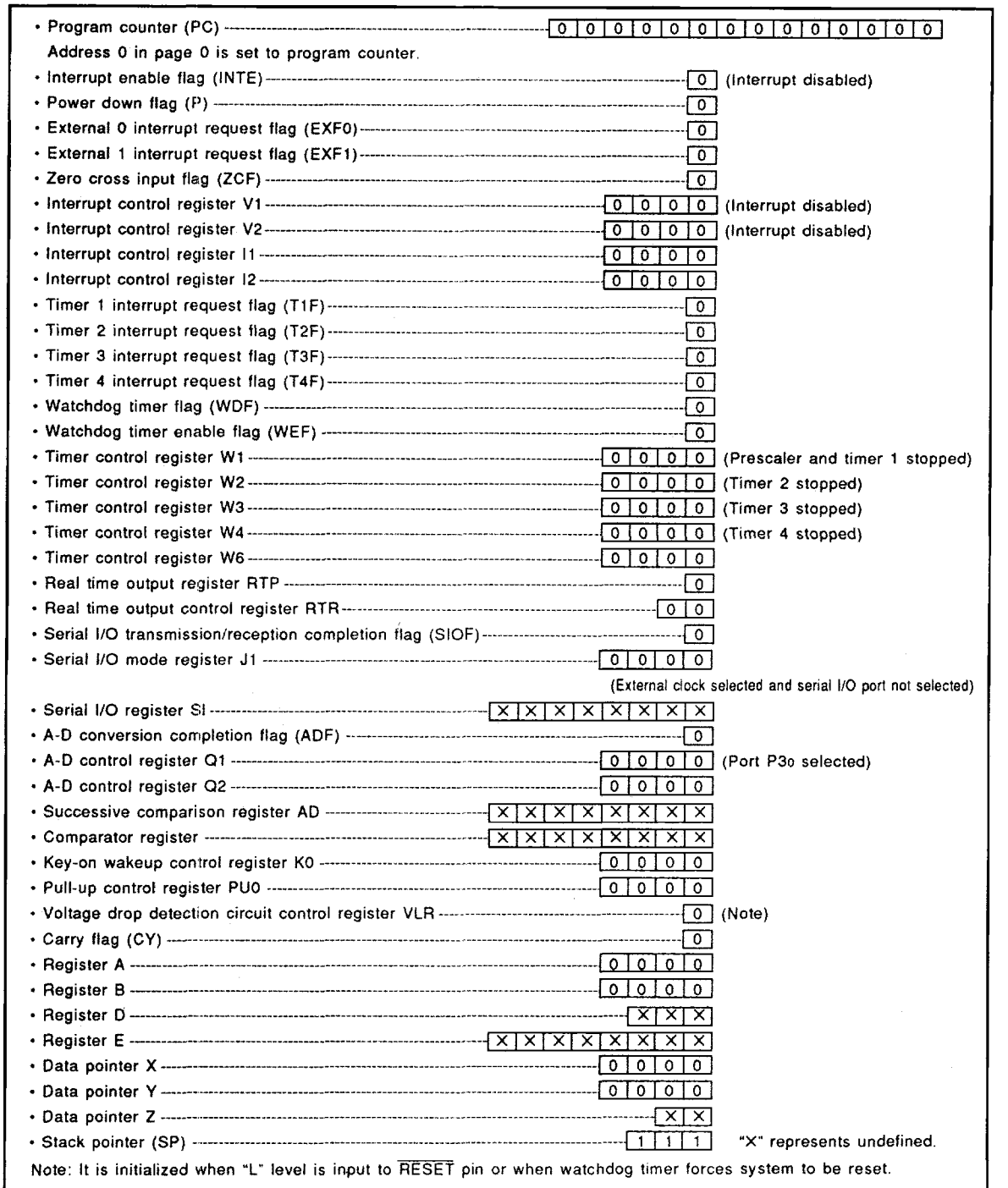


Fig. 38 Internal state at reset

VOLTAGE DROP DETECTION CIRCUIT

The built-in voltage drop detection circuit is designed to detect a drop in voltage and to reset the microcomputer if the supply voltage drops below a set value.
This circuit starts operating when the bit 0 of voltage drop detection circuit control register VLR is set to "1." Note that register VLR is initialized when an "L" level signal is input to the $\overline{\text{RESET}}$ pin ($VLR_0 = 0$).

When the contents of register VLR is set with the TVLRA instruction, only the first TVLRA instruction is valid and other TVLRA instructions are equivalent to the NOP instruction.

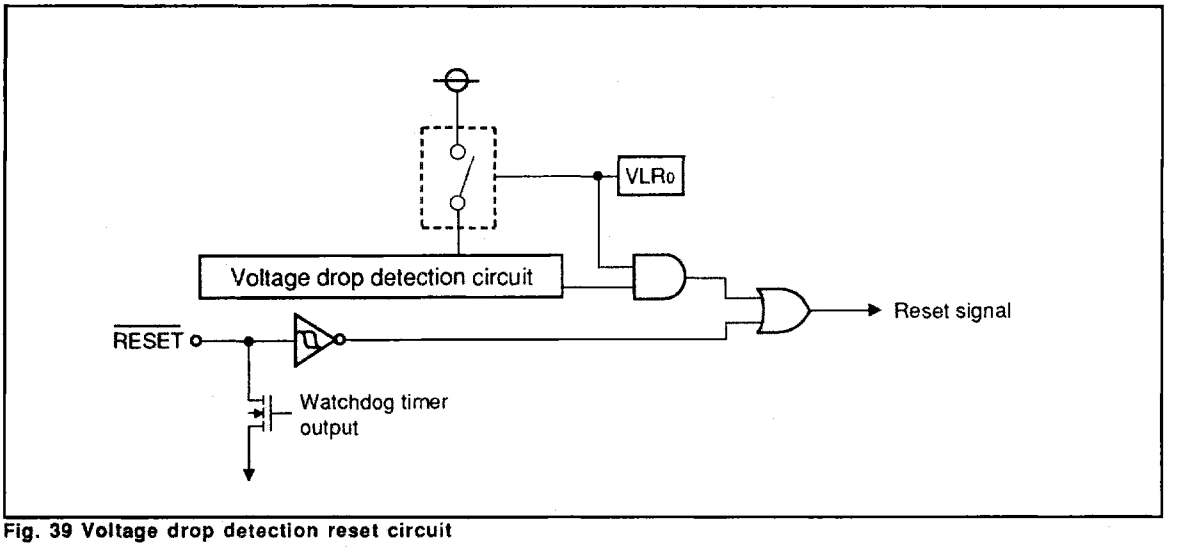


Fig. 39 Voltage drop detection reset circuit

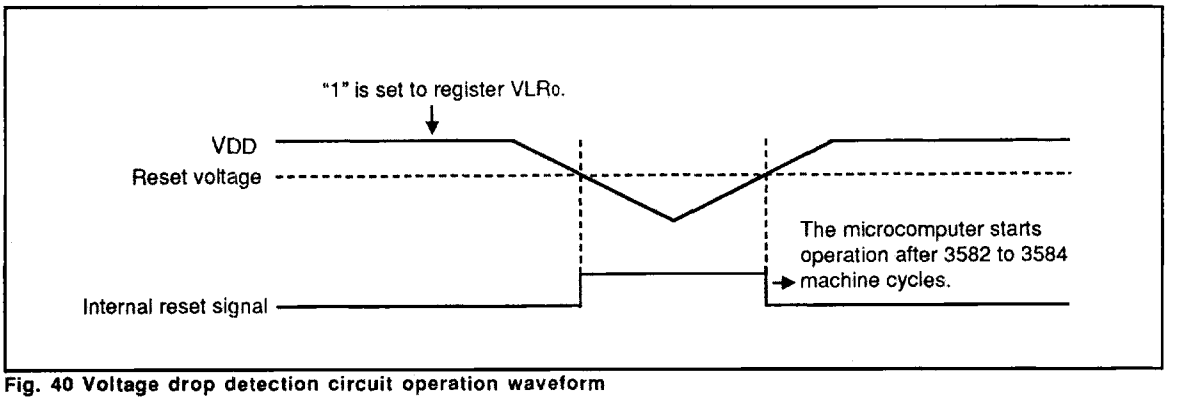


Fig. 40 Voltage drop detection circuit operation waveform

MITSUBISHI ELECTRIC

RAM BACK-UP MODE

The 4510 Group has the RAM back-up mode.
When the EPOF and POF instructions are executed continuously, 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 18 shows the function and states retained at RAM back-up. Figure 41 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 and POF instructions continuously, the CPU starts executing the program from address 0 in page 0. In this case, the P flag is "1."

(3) Cold start condition

The CPU starts executing the program from address 0 in page 0 when;
- reset pulse is input to $\overline{\text{RESET}}$ pin, or
- reset by watchdog timer is performed, or
- voltage drop detection circuit detects the voltage drop.

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

Table 18 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)	×
Contents of RAM	O
Port level	O
Timer control register W1	×
Timer control registers W2 to W4, W6	O
Interrupt control registers V1, V2	×
Interrupt control registers I1, I2	O
Timer 1 function	×
Timer 2 function	(Note 3)
Timer 3 function	(Note 3)
Timer 4 function	(Note 3)
A-D conversion function	×
A-D control registers Q1, Q2	O
Serial I/O function	×
Serial I/O mode register J1	O
Pull-up control register PU0	O
Key-on wakeup control register K0	O
Real time output register RTP	O
External 0 interrupt request flag (EXF0)	×
External 1 interrupt request flag (EXF1)	×
Zero cross input flag ZCF	×
Timer 1 interrupt request flag (T1F)	×
Timer 2 interrupt request flag (T2F)	(Note 3)
Timer 3 interrupt request flag (T3F)	(Note 3)
Timer 4 interrupt request flag (T4F)	(Note 3)
Watchdog timer flag (WDF)	× (Note 4)
Watchdog timer enable flag (WEF)	× (Note 4)
16-bit timer (WDT)	× (Note 4)
Real time output control register RTR	O
Voltage drop detection circuit control register VLR	O
A-D conversion completion flag (ADF)	×
Serial I/O transmission/reception completion flag (SIOF)	×
Interrupt enable flag (INTE)	×

Notes 1:"O" represents that the function can be retained, and "X" represents that the function is initialized. Registers and flags other than the above are undefined at RAM back-up, and set an initial value after returning.
2:The stack pointer (SP) points the level of the stack register and is initialized to "7" at RAM back-up.
3:The state of the timer is undefined.
4:Initialize the watchdog timer with the WRST instruction, and then execute the POF instruction.

(4) Return signal

An external wakeup signal is used to return from the RAM back-up mode because the oscillation is stopped. Table 19 shows the return condition for each return source.

(5) Ports P0 and P1 control registers

● Key-on wakeup control register K0

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

● Pull-up control register PU0

Register PU0 controls the ON/OFF of the ports P0 and P1 pull-up transistor. Set the contents of this register through register A with the TPU0A instruction. In addition, the TAPU0 instruction can be used to transfer the contents of register PU0 to register A.

Table 19 Return source and return condition

Return source		Return condition	Remarks
External wakeup signal	Ports P0, P1	Return by an external falling edge input ("H"→"L").	Set the port using the key-on wakeup function selected with register K0 to "H" level before going into the RAM back-up state because the port P0 shares the falling edge detection circuit with port P1.
	Port D6/INT0.	Return by an external "H" level or "L" level input. The EXF0 flag is not set.	Select the return level ("L" level or "H" level) with the bit 2 of register I1 according to the external state before going into the RAM back-up state.

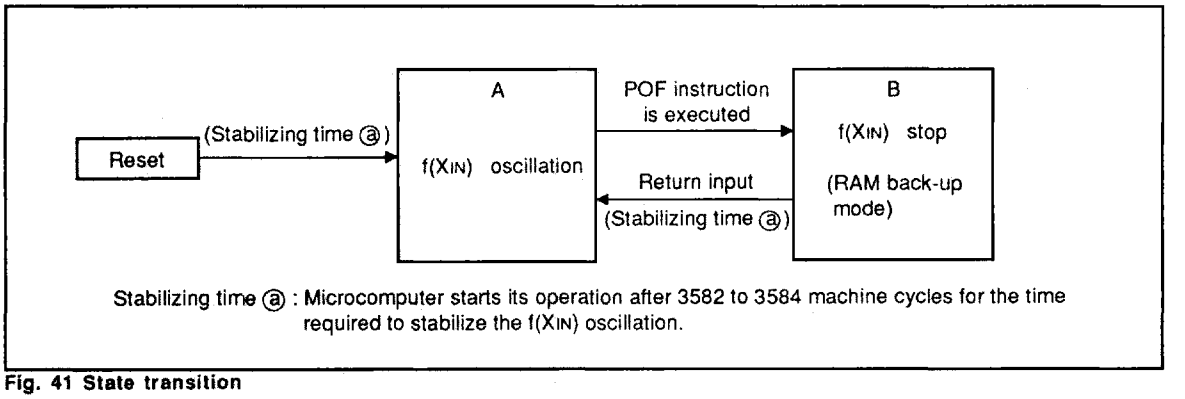


Fig. 41 State transition

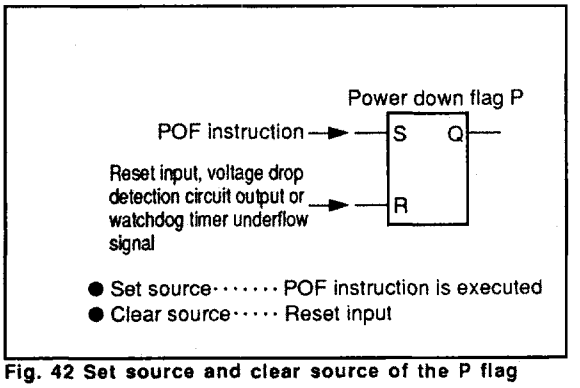


Fig. 42 Set source and clear source of the P flag

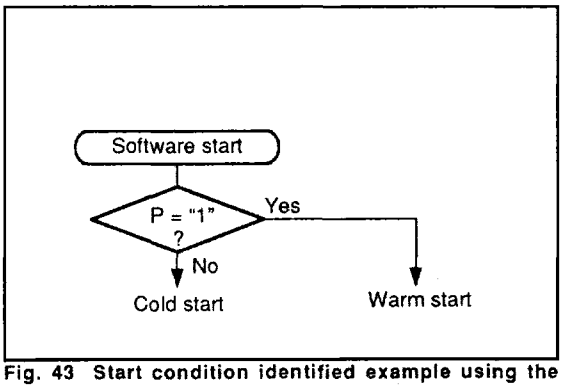


Fig. 43 Start condition identified example using the SNZP instruction

Table 20 Key-on wakeup control register, pull-up control register, voltage drop detection control register, and real time output control register

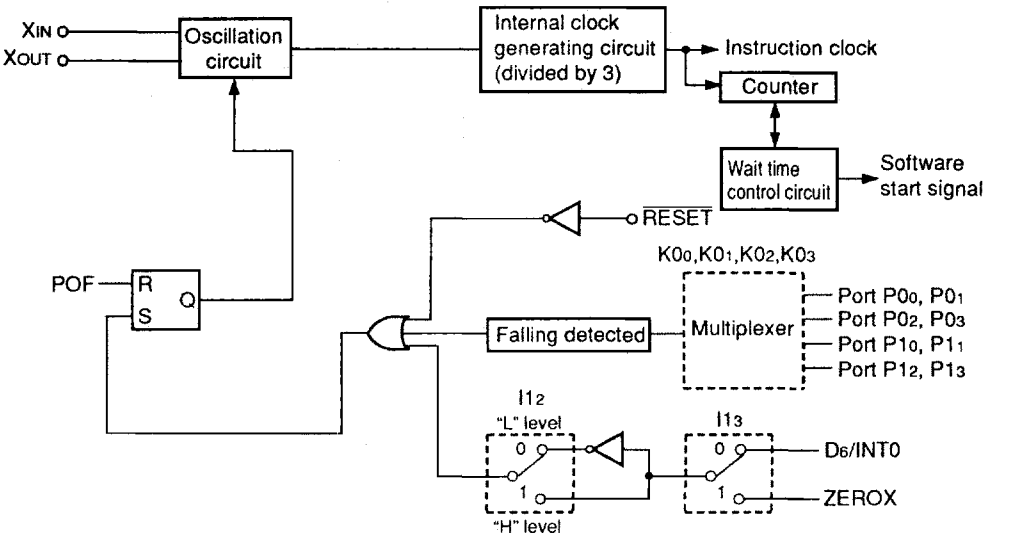
Key-on wakeup control register K0		at reset : 0000_2	at RAM back-up : state retained	R/W
$K0_3$	Pins $P1_2$ and $P1_3$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_2$	Pins $P1_0$ and $P1_1$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_1$	Pins $P0_2$ and $P0_3$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_0$	Pins $P0_0$ and $P0_1$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
Pull-up control register PU0		at reset : 0000_2	at RAM back-up : state retained	R/W
$PU0_3$	Pins $P1_2$ and $P1_3$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
$PU0_2$	Pins $P1_0$ and $P1_1$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
$PU0_1$	Pins $P0_2$ and $P0_3$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
$PU0_0$	Pins $P0_0$ and $P0_1$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
Voltage drop detection circuit control register VLR		at reset : 0_2	at RAM back-up : state retained	W
VLR_0	Voltage drop detection circuit control bit	0	Voltage drop detection circuit stop	
		1	Voltage drop detection circuit operating	
Real time output control register RTR		at reset : 00_2	at RAM back-up : state retained	W
RTR_1	D_5/real time output function selection bit	0	D_5 (I/O)	
		1	Real time output/D_5 (input)	
RTR_0	D_4/real time output function selection bit	0	D_4 (I/O)	
		1	Real time output/D_4 (input)	

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

CLOCK CONTROL

The clock control circuit consists of the following circuits.

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



POF = instruction

Note: The wait time control circuit is used to start the microcomputer operation after 3582 to 3584 machine cycles for the time required to stabilize the f(XIN) oscillation.

Fig. 44 Clock control circuit structure

Clock signal f(XIN) is obtained by externally connecting a ceramic resonator. Connect this external circuit to pins XIN and XOUT at the shortest distance. A feedback resistor is built in between pins XIN and XOUT.
When an external clock signal is input, connect the clock source to XIN and leave XOUT open. When using an external clock, the maximum value of external clock oscillating frequency is shown in Table 21.

Table 21 Maximum value of external clock oscillation frequency

Supply voltage	Oscillation frequency (duty ratio)
4.5 V to 5.5 V	4.0 MHz (40 % to 60 %)
4.0 V to 5.5 V	3.0 MHz (40 % to 60 %)
2.5 V to 5.5 V	1.0 MHz (30 % to 70 %)
2.0 V to 5.5 V	0.8 MHz (30 % to 70 %)

Note: 2.5 V to 5.5 V for the One Time PROM version and the built-in EPROM version.

ROM ORDERING METHOD

Please submit the information described below when ordering Mask ROM.

(1) M34510M2A-XXXSP/FP Mask ROM Order Confirmation Form or M34510M4A-XXXSP/FP Mask ROM Order Confirmation Form 1
(2) Data to be written into mask ROM ..... EPROM (three sets containing the identical data)
(3) Mark Specification Form 1

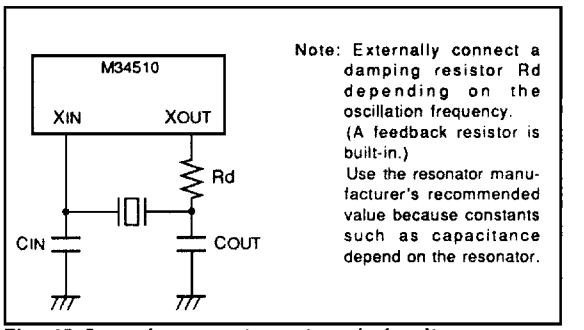


Fig. 45 Ceramic resonator external circuit

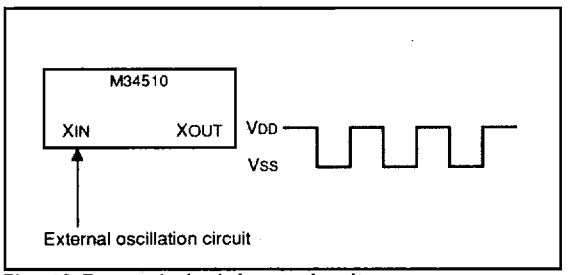


Fig. 46 External clock input circuit

LIST OF PRECAUTIONS

① Noise and latch-up prevention

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

- connect a bypass capacitor (approx. 0.1 μF) between pins V_{DD} and Vss at the shortest distance,
- equalize its wiring in width and length, and
- use relatively thick wire.

In the built-in PROM version, CNVss pin is also used as V_{PP} pin. Accordingly, when using this pin, connect this pin to Vss through a resistor about 5 kΩ in series at the shortest distance.

② Prescaler

Stop the prescaler operation to change its frequency dividing ratio.

③ Timer count source

Stop timer 1, 2, 3, or 4 counting to change its count source.

④ Reading the count value

Stop timer 1, 2, 3, or 4 counting and then execute the TAB1, TAB2, TAB3, or TAB4 instruction to read its data.

⑤ D_6/INT0/ZEROX pin

When the interrupt valid waveform of the D_6/INT0/ZEROX pin is changed with the bit 2 of register I1 in a program, be careful about the following notes.

- Clear the bit 0 of register V1 to "0" before the interrupt valid waveform of D_6/INT0/ZEROX pin is changed with the bit 2 of register I1 (refer to Figure 47①).
- Depending on the input state of the D_6/INT0/ZEROX pin, the external 0 interrupt request flag (EXF0) may be set when the interrupt valid waveform is changed. Accordingly, set a value to register I1, and execute the SNZ0 instruction to clear the EXF0 flag after executing at least one instruction (refer to Figure 47②).

```
  :
LA   4      ; (XXX0₂)
TV1A        ; The SNZ0 instruction is valid  ①
LA   4
TI1A        ; Change of the interrupt valid waveform
NOP                                          ②
SNZ0        ; The SNZ0 instruction is executed
NOP
  :      X : this bit is not related to the setting of
            INT0.
```

Fig. 47 External 0 interrupt program example

⑥ $P2_1$/INT1 pin

When the interrupt valid waveform of $P2_1$/INT1 pin is changed with the bit 3 of register I2 in a program, be careful about the following notes.

- Clear the bit 1 of register V1 to "0" before the interrupt valid waveform of $P2_1$/INT1 pin is changed with the bit 3 of register I2 (refer to Figure 48③).
- Depending on the input state of the $P2_1$/INT1 pin, the external 1 interrupt request flag (EXF1) may be set when the interrupt valid waveform is changed. Accordingly, set a value to register I2 and execute the SNZ1 instruction to clear the EXF1 flag after executing at least one instruction (refer to Figure 48④).

```
  :
LA   8      ; (XX0X₂)
TV1A        ; The SNZ1 instruction is valid  ③
LA   8
TI2A        ; Change of the interrupt valid waveform
NOP                                          ④
SNZ1
NOP
  :      X : this bit is not related to the setting of
            INT1.
```

Fig. 48 External 1 interrupt program example

⑦ Built-in PROM version

The operating power voltage of the built-in EPROM version and One Time PROM version is 2.5 V to 5.5 V. Operating temperature range of the built-in EPROM version is −20 °C to 70 °C.

Built-in EPROM version is the microcomputer for program development. Use this microcomputer only for program development and prototype test.

⑧ Multifunction

Be sure that the I/O of D_6, the input of D_4, D_5, D_7–D_9, $P2_0$ and $P2_1$, and the output of $P3_0$–$P3_3$ can be used even when INT0/ZEROX, RTP, CNTR, S_{CK}, S_{OUT}, S_{IN}, INT1 and A_{IN0}–A_{IN3} are selected.

⑨ A-D converter-1

When the operating mode of the A-D converter is changed from the comparator mode to the A-D conversion mode with the bit 3 of register Q1 in a program, be careful about the following notes.

- Clear the bit 2 of register V2 to "0" to change the operating mode of the A-D converter from the comparator mode to the A-D conversion mode with the bit 3 of register Q1 (refer to Figure 49⑤).
- 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 register Q1, and execute the SNZAD instruction to clear the ADF flag.

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 during operating the A-D converter.

```
:
LA   8      ; (X0XX2)
TV2A        ; The SNZAD instruction is valid  ⑤
LA   0      ; (0XXX2)
TQ1A        ; Change of the operating mode of the
              A-D converter from the comparator
              mode to the A-D conversion mode
SNZAD
NOP
:
```

X : this bit is not related to the change of the operating mode of the A-D conversion.

Fig. 49 A-D converter operating mode program example

⑩ A-D converter-2

A-D conversion circuit is equipped with a capacitor which is used to compare the analog voltage. Accordingly, when the analog voltage is input from the circuit with high-impedance and, charge/discharge noise is generated and the sufficient A-D accuracy may not be obtained. Therefore, reduce the impedance or, connect a capacitor (0.01 μF to 1 μF) to analog input pins (Figure 50).

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 51. In addition, test the application products sufficiently.

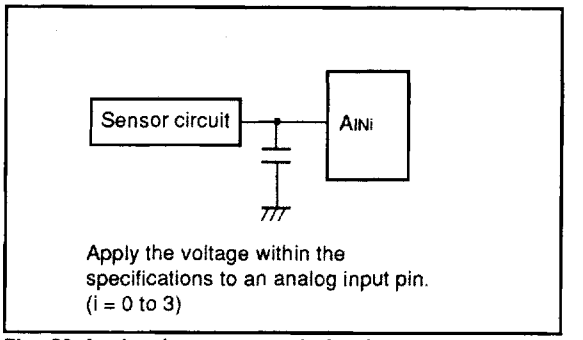


Fig. 50 Analog input external circuit example-1

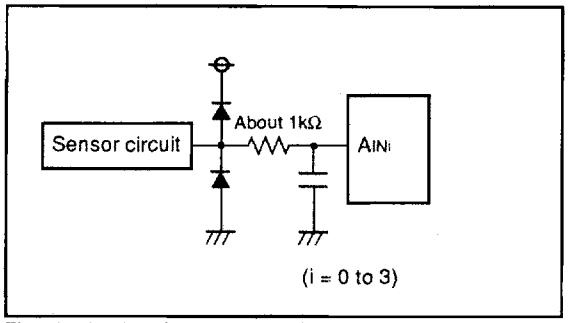


Fig. 51 Analog input external circuit example-2

⑪ POF instruction

Execute the POF instruction immediately after executing the EPOF instruction to enter the RAM back-up. 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.

⑫ Analog input pins

Note the following when using the analog input pins also for I/O port P3 functions:

- P3 output

  Even when pins P3₀/AIN0, P3₁/AIN1, P3₂/AIN2, and P3₃/AIN3 are set for analog input, they also function as port P3 output. When executing the OP3A instruction, set the outputs of pins set for analog input to "1."
- P3 input

  Pins that are set for analog input do not function as port P3 input. After executing the IAP3 instruction, the values of bits corresponding to pins set for analog input are transmitted to register A as "0."

⑬ Program counter

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

SYMBOL

The symbols shown below are used in the following instruction function table and instruction list.

Symbol	Contents	Symbol	Contents
A	Register A (4 bits)	T1	Timer 1
B	Register B (4 bits)	T2	Timer 2
DR	Register D (3 bits)	T3	Timer 3
E	Register E (8 bits)	T4	Timer 4
Q1	A-D control register Q1 (4 bits)	T1F	Timer 1 interrupt request flag
Q2	A-D control register Q2 (4 bits)	T2F	Timer 2 interrupt request flag
AD	Successive comparison register AD (8 bits)	T3F	Timer 3 interrupt request flag
J1	Serial I/O mode register J1 (4 bits)	T4F	Timer 4 interrupt request flag
SI	Serial I/O register SI (8 bits)	WDF	Watchdog timer flag
V1	Interrupt control register V1 (4 bits)	INTE	Interrupt enable flag
V2	Interrupt control register V2 (4 bits)	EXF0	External 0 interrupt request flag
I1	Interrupt control register I1 (4 bits)	EXF1	External 1 interrupt request flag
I2	Interrupt control register I2 (4 bits)	ZCF	Zero cross input flag
W1	Timer control register W1 (4 bits)	P	Power down flag
W2	Timer control register W2 (4 bits)	ADF	A-D conversion completion flag
W3	Timer control register W3 (4 bits)	SIOF	Serial I/O transmission/reception completion flag
W4	Timer control register W4 (4 bits)		
W6	Timer control register W6 (4 bits)	D	Port D (10 bits)
RTR	Real time output control register RTR (2 bits)	P0	Port P0 (4 bits)
K0	Key-on wakeup control register K0 (4 bits)	P1	Port P1 (4 bits)
PU0	Pull-up control register PU0 (4 bits)	P2	Port P2 (2 bits)
VLR	Voltage drop detection circuit control register VLR (1 bit)	P3	Port P3 (4 bits)
X	Register X (4 bits)	x	Hexadecimal variable
Y	Register Y (4 bits)	y	Hexadecimal variable
Z	Register Z (2 bits)	z	Hexadecimal variable
DP	Data pointer (10 bits)	p	Hexadecimal variable
	(It consists of registers X, Y, and Z)	n	Hexadecimal constant
PC	Program counter (14 bits)	i	Hexadecimal constant
PC_H	High-order 7 bits of program counter	j	Hexadecimal constant
PC_L	Low-order 7 bits of program counter	$A_3A_2A_1A_0$	Binary notation of hexadecimal variable A
SK	Stack register (14 bits × 8)		(same for others)
SP	Stack pointer (3 bits)		
CY	Carry flag	←	Direction of data movement
R1	Timer 1 reload register	↔	Data exchange between a register and memory
R2	Timer 2 reload register	?	Decision of state shown before "?"
R3	Timer 3 reload register	( )	Contents of registers and memories
R4	Timer 4 reload register	—	Negate, Flag unchanged after executing instruction
		M(DP)	RAM address pointed by the data pointer
		a	Label indicating address a_6 a_5 a_4 a_3 a_2 a_1 a_0
		p, a	Label indicating address a_6 a_5 a_4 a_3 a_2 a_1 a_0 in page p_5 p_4 p_3 p_2 p_1 p_0
		C	Hex. C + Hex. number x (also same for others)
		+	
		x	

Note : The 4510 Group just invalidates the next instruction when a skip is performed. The contents of program counter is not increased by 2. Accordingly, the number of cycles does not change even if skip is not performed. However, the cycle count becomes "1" if the TABP p, RT, or RTS instruction is skipped.

LIST OF INSTRUCTION FUNCTION

Grouping	Mnemonic	Function
Register to register transfer	TAB	(A) ← (B)
	TBA	(B) ← (A)
	TAY	(A) ← (Y)
	TYA	(Y) ← (A)
	TEAB	(E_7–E_4) ← (B) (E_3–E_0) ← (A)
	TABE	(B) ← (E_7–E_4) (A) ← (E_3–E_0)
	TDA	(DR_2–DR_0) ← (A_2–A_0)
	TAD	(A_2–A_0) ← (DR_2–DR_0) (A_3) ← 0
	TAZ	(A_1, A_0) ← (Z_1, Z_0) (A_3, A_2) ← 0
	TAX	(A) ← (X)
	TASP	(A_2–A_0) ← (SP_2–SP_0) (A_3) ← 0
RAM addresses	LXY x, y	(X) ← x, x = 0 to 15 (Y) ← y, y = 0 to 15
	LZ z	(Z) ← z, z = 0 to 3
	INY	(Y) ← (Y) + 1
	DEY	(Y) ← (Y) − 1
RAM to register transfer	TAM j	(A) ← (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15
	XAM j	(A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15
	XAMD j	(A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) − 1

Grouping	Mnemonic	Function
RAM to register transfer	XAMI j	(A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) + 1
	TMA j	(M(DP)) ← (A) (X) ← (X)EXOR(j) j = 0 to 15
Arithmetic operation	LA n	(A) ← n n = 0 to 15
	TABP p	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p (PC_L) ← (DR_2–DR_0, A_3–A_0) (B) ← (ROM(PC))7–4 (A) ← (ROM(PC))3–0 (PC) ← (SK(SP)) (SP) ← (SP) − 1
	AM	(A) ← (A) + (M(DP))
	AMC	(A) ← (A) + (M(DP)) + (CY) (CY) ← Carry
	A n	(A) ← (A) + n n = 0 to 15
	AND	(A) ← (A) AND (M(DP))
	OR	(A) ← (A) OR (M(DP))
	SC	(CY) ← 1
	RC	(CY) ← 0
	SZC	(CY) = 0 ?
	CMA	(A) ← ($\overline{A}$)
	RAR	CY → $A_3A_2A_1A_0$

Grouping	Mnemonic	Function
Bit operation	SB j	(Mj(DP)) ← 1 j = 0 to 3
	RB j	(Mj(DP)) ← 0 j = 0 to 3
	SZB j	(Mj(DP)) = 0 ? j = 0 to 3
Comparison operation	SEAM	(A) = (M(DP)) ?
	SEA n	(A) = n ? n = 0 to 15
Branch operation	B a	(PC_L) ← a_6–a_0
	BL p, a	(PC_H) ← p (PC_L) ← a_6–a_0
	BLA p	(PC_H) ← p (PC_L) ← (DR_2–DR_0, A_3–A_0)
Subroutine operation	BM a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← 2 (PC_L) ← a_6–a_0
	BML p, a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p (PC_L) ← a_6–a_0
	BMLA p	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p (PC_L) ← (DR_2–DR_0, A_3–A_0)
Return operation	RTI	(PC) ← (SK(SP)) (SP) ← (SP) − 1
	RT	(PC) ← (SK(SP)) (SP) ← (SP) − 1
	RTS	(PC) ← (SK(SP)) (SP) ← (SP) − 1

LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function
Interrupt operation	DI	(INTE) ← 0
	EI	(INTE) ← 1
	SNZ0	(EXF0) = 1 ? After skipping (EXF0) ← 0
	SNZ1	(EXF1) = 1 ? After skipping (EXF1) ← 0
	SNZI0	$I1_2$ = 1 : (INT0) = "H" $I1_2$ = 0 : (INT0) = "L"
	SNZI1	$I2_3$ = 1 : (INT1) = "H" $I2_3$ = 0 : (INT1) = "L"
	TAV1	(A) ← (V1)
	TV1A	(V1) ← (A)
	TAV2	(A) ← (V2)
	TV2A	(V2) ← (A)
	TAI1	(A) ← (I1)
	TI1A	(I1) ← (A)
	TAI2	(A) ← (I2)
	TI2A	(I2) ← (A)
Timer operation	TAW1	(A) ← (W1)
	TW1A	(W1) ← (A)
	TAW2	(A) ← (W2)
	TW2A	(W2) ← (A)
	TAW3	(A) ← (W3)
	TW3A	(W3) ← (A)
	TAW4	(A) ← (W4)
	TW4A	(W4) ← (A)
	TAW6	(A) ← (W6)
	TW6A	(W6) ← (A)
Timer operation	TAB1	(B) ← ($T1_7$–$T1_4$) (A) ← ($T1_3$–$T1_0$)
	T1AB	($R1_7$–$R1_4$) ← (B) ($T1_7$–$T1_4$) ← (B) ($R1_3$–$R1_0$) ← (A) ($T1_3$–$T1_0$) ← (A)
	TAB2	(B) ← ($T2_7$–$T2_4$) (A) ← ($T2_3$–$T2_0$)
	T2AB	($R2_7$–$R2_4$) ← (B) ($T2_7$–$T2_4$) ← (B) ($R2_3$–$R2_0$) ← (A) ($T2_3$–$T2_0$) ← (A)
	TAB3	(B) ← ($T3_7$–$T3_4$) (A) ← ($T3_3$–$T3_0$)
	T3AB	($R3_7$–$R3_4$) ← (B) ($T3_7$–$T3_4$) ← (B) ($R3_3$–$R3_0$) ← (A) ($T3_3$–$T3_0$) ← (A)
	TAB4	(B) ← ($T4_7$–$T4_4$) (A) ← ($T4_3$–$T4_0$)
	T4AB	($R4_7$–$R4_4$) ← (B) ($T4_7$–$T4_4$) ← (B) ($R4_3$–$R4_0$) ← (A) ($T4_3$–$T4_0$) ← (A)
	TR4AB	($R4_7$–$R4_4$) ← (B) ($R4_3$–$R4_0$) ← (A)
	SNZT1	(T1F) = 1 ? After skipping (T1F) ← 0
	SNZT2	(T2F) = 1 ? After skipping (T2F) ← 0
	SNZT3	(T3F) = 1 ? After skipping (T3F) ← 0
	SNZT4	(T4F) = 1 ? After skipping (T4F) ← 0
Input/Output operation	IAP0	(A) ← (P0)
	OP0A	(P0) ← (A)
	IAP1	(A) ← (P1)
	OP1A	(P1) ← (A)
	IAP2	(A_1–A_0) ← ($P2_1$–$P2_0$) (A_3, A_2) ← 0
	IAP3	(A) ← (P3)
	OP3A	(P3) ← (A)
	CLD	(D) ← 1
	RD	(D(Y)) ← 0 (Y) = 0 to 9
	SD	(D(Y)) ← 1 (Y) = 0 to 9
	SZD	(D(Y)) = 0 ? (Y) = 0 to 9
	TK0A	(K0) ← (A)
	TAK0	(A) ← (K0)
	TPU0A	(PU0) ← (A)
	TAPU0	(A) ← (PU0)
Serial I/O control operation	TABSI	(A) ← (SI_3–SI_0) (B) ← (SI_7–SI_4)
	TSIAB	(SI_3–SI_0) ← (A) (SI_7–SI_4) ← (B)
	TAJ1	(A) ← (J1)
	TJ1A	(J1) ← (A)
	SST	(SIOF) ← 0 Serial I/O starting
	SNZSI	(SIOF) = 1 ? After skipping (SIOF) ← 0

LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function
A-D conversion operation	TABAD	(A) ← (AD_3–AD_0) (B) ← (AD_7–AD_4)
	TADAB	(AD_3–AD_0) ← (A) (AD_7–AD_4) ← (B)
	TAQ1	(A) ← (Q1)
	TQ1A	(Q1) ← (A)
	ADST	(ADF) ← 0 A-D conversion starting
	SNZAD	(ADF) = 1 ? After skipping, (ADF) ← 0
	TAQ2	(A) ← (Q2)
	TQ2A	(Q2) ← (A)
Other operation	NOP	(PC) ← (PC) + 1
	POF	RAM back-up
	EPOF	POF instruction valid
	SNZP	(P) = 1 ?
	WRST	(WDF) ← 0, (WEF) ← 1 (WDT) = $FFFF_{16}$
	RTPS	(RTP) ← 1
	RTPR	(RTP) ← 0
	TRTPA	(RTP) ← (A_0)
	TRTRA	(RTR_1, RTR_0) ← (A_1, A_0)
	TVLRA	(VLR) ← (A_0)

INSTRUCTION CODE TABLE

D9–D4 / D3–D0	Hex. notation	000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111	010000 to 010111	011000 to 011111
		00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10 to 17	18 to 1F
0000	0	NOP	BLA	SZB 0	BMLA	—	TASP	A 0	LA 0	TABP 0	TABP 16**	TABP 32*	TABP 48*	BML	BML**	BL	BL**	BM	B
0001	1	—	CLD	SZB 1	—	—	TAD	A 1	LA 1	TABP 1	TABP 17**	TABP 33*	TABP 49*	BML	BML**	BL	BL**	BM	B
0010	2	POF	—	SZB 2	—	—	TAX	A 2	LA 2	TABP 2	TABP 18**	TABP 34*	TABP 50*	BML	BML**	BL	BL**	BM	B
0011	3	SNZP	INY	SZB 3	—	—	TAZ	A 3	LA 3	TABP 3	TABP 19**	TABP 35*	TABP 51*	BML	BML**	BL	BL**	BM	B
0100	4	DI	RD	SZD	—	RT	TAV1	A 4	LA 4	TABP 4	TABP 20**	TABP 36*	TABP 52*	BML	BML**	BL	BL**	BM	B
0101	5	EI	SD	SEAn	—	RTS	TAV2	A 5	LA 5	TABP 5	TABP 21**	TABP 37*	TABP 53*	BML	BML**	BL	BL**	BM	B
0110	6	RC	—	SEAM	—	RTI	—	A 6	LA 6	TABP 6	TABP 22**	TABP 38*	TABP 54*	BML	BML**	BL	BL**	BM	B
0111	7	SC	DEY	—	—	—	—	A 7	LA 7	TABP 7	TABP 23**	TABP 39*	TABP 55*	BML	BML**	BL	BL**	BM	B
1000	8	—	AND	—	SNZ0	LZ 0	—	A 8	LA 8	TABP 8	TABP 24**	TABP 40*	TABP 56*	BML	BML**	BL	BL**	BM	B
1001	9	—	OR	TDA	SNZ1	LZ 1	—	A 9	LA 9	TABP 9	TABP 25**	TABP 41*	TABP 57*	BML	BML**	BL	BL**	BM	B
1010	A	AM	TEAB	TABE	SNZI0	LZ 2	—	A 10	LA 10	TABP 10	TABP 26**	TABP 42*	TABP 58*	BML	BML**	BL	BL**	BM	B
1011	B	AMC	—	—	SNZI1	LZ 3	EPOF	A 11	LA 11	TABP 11	TABP 27**	TABP 43*	TABP 59*	BML	BML**	BL	BL**	BM	B
1100	C	TYA	CMA	—	—	RB 0	SB 0	A 12	LA 12	TABP 12	TABP 28**	TABP 44*	TABP 60*	BML	BML**	BL	BL**	BM	B
1101	D	—	RAR	—	—	RB 1	SB 1	A 13	LA 13	TABP 13	TABP 29**	TABP 45*	TABP 61*	BML	BML**	BL	BL**	BM	B
1110	E	TBA	TAB	—	TV2A	RB 2	SB 2	A 14	LA 14	TABP 14	TABP 30**	TABP 46*	TABP 62*	BML	BML**	BL	BL**	BM	B
1111	F	—	TAY	SZC	TV1A	RB 3	SB 3	A 15	LA 15	TABP 15	TABP 31**	TABP 47*	TABP 63*	BML	BML**	BL	BL**	BM	B

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

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

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

* and ** cannot be used at M34510M2A-XXXSP/FP.

* cannot be used at M34510M4A-XXXSP/FP.

INSTRUCTION CODE TABLE (continued)

D9–D4 / D3–D0 Hex. notation	100000	100001	100010	100011	100100	100101	100110	100111	101000	101001	101010	101011	101100	101101	101110	101111	110000 to 111111
	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30 to 3F
0000 0	—	TW3A	OP0A	T1AB	—	TAW6	IAP0	TAB1	SNZT1	TVLRA	WRST	TMA 0	TAM 0	XAM 0	XAMI 0	XAMD 0	LXY
0001 1	—	TW4A	OP1A	T2AB	—	—	IAP1	TAB2	SNZT2	TRTRA	RTPS	TMA 1	TAM 1	XAM 1	XAMI 1	XAMD 1	LXY
0010 2	TJ1A	—	—	T3AB	TAJ1	—	IAP2	TAB3	SNZT3	—	RTPR	TMA 2	TAM 2	XAM 2	XAMI 2	XAMD 2	LXY
0011 3	—	TW6A	OP3A	T4AB	—	TAI1	IAP3	TAB4	SNZT4	—	—	TMA 3	TAM 3	XAM 3	XAMI 3	XAMD 3	LXY
0100 4	TQ1A	—	—	—	TAQ1	TAI2	—	—	—	—	—	TMA 4	TAM 4	XAM 4	XAMI 4	XAMD 4	LXY
0101 5	TQ2A	—	—	—	TAQ2	—	—	—	—	—	—	TMA 5	TAM 5	XAM 5	XAMI 5	XAMD 5	LXY
0110 6	—	—	—	—	—	TAK0	—	—	—	—	—	TMA 6	TAM 6	XAM 6	XAMI 6	XAMD 6	LXY
0111 7	—	TI1A	—	—	—	TAPU0	—	—	SNZAD	—	—	TMA 7	TAM 7	XAM 7	XAMI 7	XAMD 7	LXY
1000 8	—	TI2A	—	TSIAB	—	—	—	TABSI	SNZSI	—	—	TMA 8	TAM 8	XAM 8	XAMI 8	XAMD 8	LXY
1001 9	—	TRTPA	—	TADAB	—	—	—	TABAD	—	—	—	TMA 9	TAM 9	XAM 9	XAMI 9	XAMD 9	LXY
1010 A	—	—	—	—	—	—	—	—	—	—	—	TMA 10	TAM 10	XAM 10	XAMI 10	XAMD 10	LXY
1011 B	—	TK0A	—	—	TAW1	—	—	—	—	—	—	TMA 11	TAM 11	XAM 11	XAMI 11	XAMD 11	LXY
1100 C	—	—	—	TR4AB	TAW2	—	—	—	—	—	—	TMA 12	TAM 12	XAM 12	XAMI 12	XAMD 12	LXY
1101 D	—	—	TPU0A	—	TAW3	—	—	—	—	—	—	TMA 13	TAM 13	XAM 13	XAMI 13	XAMD 13	LXY
1110 E	TW1A	—	—	—	TAW4	—	—	—	—	SST	—	TMA 14	TAM 14	XAM 14	XAMI 14	XAMD 14	LXY
1111 F	TW2A	—	—	—	—	—	—	—	—	ADST	—	TMA 15	TAM 15	XAM 15	XAMI 15	XAMD 15	LXY

The above table shows the relationship between machine language codes and machine language instructions. D_3–D_0 show the low-order 4 bits of the machine language code, and D_9–D_4 show the high-order 6 bits of the machine language code. The hexadecimal representation of the code is also provided. There are one-word instructions and two-word instructions, but only the first word of each instruction is shown. Do not use the code marked "—."

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

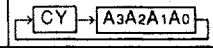
	The second Word
BL	10 paaa aaaa
BML	10 paaa aaaa
BLA	10 pp00 pppp
BMLA	10 pp00 pppp
SEA	00 0111 nnnn
SZD	00 0010 1011

MACHINE INSTRUCTIONS

Parameter / Type of instructions	Mnemonic	Instruction code										Hexadecimal notation			Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0						
Register to register transfer	TAB	0	0	0	0	0	1	1	1	1	0	0	1	E	1	1	(A) ← (B)
	TBA	0	0	0	0	0	0	1	1	1	0	0	0	E	1	1	(B) ← (A)
	TAY	0	0	0	0	0	1	1	1	1	1	0	1	F	1	1	(A) ← (Y)
	TYA	0	0	0	0	0	0	1	1	0	0	0	0	C	1	1	(Y) ← (A)
	TEAB	0	0	0	0	0	1	1	0	1	0	0	1	A	1	1	(E_7–E_4) ← (B) (E_3–E_0) ← (A)
	TABE	0	0	0	0	1	0	1	0	1	0	0	2	A	1	1	(B) ← (E_7–E_4) (A) ← (E_3–E_0)
	TDA	0	0	0	0	1	0	1	0	0	1	0	2	9	1	1	(DR_2–DR_0) ← (A_2–A_0)
	TAD	0	0	0	1	0	1	0	0	0	1	0	5	1	1	1	(A_2–A_0) ← (DR_2–DR_0) (A_3) ← 0
	TAZ	0	0	0	1	0	1	0	0	1	1	0	5	3	1	1	(A_1, A_0) ← (Z_1, Z_0) (A_3, A_2) ← 0
	TAX	0	0	0	1	0	1	0	0	1	0	0	5	2	1	1	(A) ← (X)
	TASP	0	0	0	1	0	1	0	0	0	0	0	5	0	1	1	(A_2–A_0) ← (SP_2–SP_0) (A_3) ← 0
RAM addresses	LXY x, y	1	1	x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0	3	x	y	1	1	(X) ← x, x = 0 to 15 (Y) ← y, y = 0 to 15
	LZ z	0	0	0	1	0	0	1	0	z_1	z_0	0	4	8 + z	1	1	(Z) ← z, z = 0 to 3
	INY	0	0	0	0	0	1	0	0	1	1	0	1	3	1	1	(Y) ← (Y) + 1
	DEY	0	0	0	0	0	1	0	1	1	1	0	1	7	1	1	(Y) ← (Y) − 1
RAM to register transfer	TAM j	1	0	1	1	0	0	j	j	j	j	2	C	j	1	1	(A) ← (M(DP)) (X) ← (X)EXOR(j), j = 0 to 15
	XAM j	1	0	1	1	0	1	j	j	j	j	2	D	j	1	1	(A) ← → (M(DP)) (X) ← (X)EXOR(j), j = 0 to 15
	XAMD j	1	0	1	1	1	1	j	j	j	j	2	F	j	1	1	(A) ← → (M(DP)) (X) ← (X)EXOR(j), j = 0 to 15 (Y) ← (Y) − 1
	XAMI j	1	0	1	1	1	0	j	j	j	j	2	E	j	1	1	(A) ← → (M(DP)) (X) ← (X)EXOR(j), j = 0 to 15 (Y) ← (Y) + 1
	TMA j	1	0	1	0	1	1	j	j	j	j	2	B	j	1	1	(M(DP)) ← (A) (X) ← (X)EXOR(j), j = 0 to 15

Skip condition	Carry flag CY	Datailed description
–	–	Transfers the contents of register B to register A.
–	–	Transfers the contents of register A to register B.
–	–	Transfers the contents of register Y to register A.
–	–	Transfers the contents of register A to register Y.
–	–	Transfers the contents of registers A and B to register E.
–	–	Transfers the contents of register E to registers A and B.
–	–	Transfers the contents of register A to register D.
–	–	Transfers the contents of register D to register A.
–	–	Transfers the contents of register Z to register A.
–	–	Transfers the contents of register X to register A.
–	–	Transfers the contents of stack pointer (SP) to 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.
(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.
–	–	After transferring the contents of M(DP) to register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
–	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
(Y) = 15	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped.
(Y) = 0	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped.
–	–	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 (continued)

Parameter / Type of instructions	Mnemonic	Instruction code D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
Arithmetic operation	LA n	0	0	0	1	1	1	n	n	n	n	0 7 n	1	1	(A) ← n n = 0 to 15
	TABP p	0	0	1	0	p_5	p_4	p_3	p_2	p_1	p_0	0 8 p +p	1	3	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p (PC_L) ← $(DR_2$–DR_0, A_3–$A_0)$ (B) ← $(ROM(PC))_{7-4}$ (A) ← $(ROM(PC))_{3-0}$ (PC) ← (SK(SP)) (SP) ← (SP) − 1 (Note)
	AM	0	0	0	0	0	0	1	0	1	0	0 0 A	1	1	(A) ← (A) + (M(DP))
	AMC	0	0	0	0	0	0	1	0	1	1	0 0 B	1	1	(A) ← (A) + (M(DP))+ (CY) (CY) ← Carry
	A n	0	0	0	1	1	0	n	n	n	n	0 6 n	1	1	(A) ← (A) + n n = 0 to 15
	AND	0	0	0	0	0	1	1	0	0	0	0 1 8	1	1	(A) ← (A) AND (M(DP))
	OR	0	0	0	0	0	1	1	0	0	1	0 1 9	1	1	(A) ← (A) OR (M(DP))
	SC	0	0	0	0	0	0	0	1	1	1	0 0 7	1	1	(CY) ← 1
	RC	0	0	0	0	0	0	0	1	1	0	0 0 6	1	1	(CY) ← 0
	SZC	0	0	0	0	1	0	1	1	1	1	0 2 F	1	1	(CY) = 0 ?
	CMA	0	0	0	0	0	1	1	1	0	0	0 1 C	1	1	(A) ← $(\overline{A})$
	RAR	0	0	0	0	0	1	1	1	0	1	0 1 D	1	1	
Bit operation	SB j	0	0	0	1	0	1	1	1	j	j	0 5 C +j	1	1	(Mj(DP)) ← 1 j = 0 to 3
	RB j	0	0	0	1	0	0	1	1	j	j	0 4 C +j	1	1	(Mj(DP)) ← 0 j = 0 to 3
	SZB j	0	0	0	0	1	0	0	0	j	j	0 2 j	1	1	(Mj(DP)) = 0 ? j = 0 to 3
Comparison operation	SEAM	0	0	0	0	1	0	0	1	1	0	0 2 6	1	1	(A) = (M(DP)) ?
	SEA n	0	0	0	0	1	0	0	1	0	1	0 2 5	2	2	(A) = n ? n = 0 to 15
		0	0	0	1	1	1	n	n	n	n	0 7 n			

Note : p is 0 to 63 for M34510E8, p is 0 to 31 for M34510M4A, and p is 0 to 15 for M34510M2A.

Skip condition	Carry flag CY	Datailed description
Continuous description	–	Loads the value n in the immediate field to register A. When the LA instructions are continuously coded and executed, only the first LA instruction is executed and other LA instructions coded continuously are skipped.
–	–	Transfers bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 7 to 0 are the ROM pattern in address $(DR_2\ DR_1\ DR_0\ A_3\ A_2\ A_1\ A_0)_2$ specified by registers A and D in page p. When this instruction is executed, 1 stage of stack register is used.
–	–	Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged.
–	0/1	Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY.
Overflow = 0	–	Adds the value n in the immediate field to register A. The contents of carry flag CY remains unchanged. Skips the next instruction when there is no overflow as the result of operation.
–	–	Takes the AND operation between the contents of register A and the contents of M(DP), and stores the result in register A.
–	–	Takes the OR operation between the contents of register A and the contents of M(DP), and stores the result in register A.
–	1	Sets (1) to carry flag CY.
–	0	Clears (0) to carry flag CY.
(CY) = 0	–	Skips the next instruction when the contents of carry flag CY is "0."
–	–	Stores the one's complement for register A's contents in register A.
–	0/1	Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right.
–	–	Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).
–	–	Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).
(Mj(DP)) = 0 j = 0 to 3	–	Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is "0."
(A) = (M(DP))	–	Skips the next instruction when the contents of register A is equal to the contents of M(DP).
(A) = n	–	Skips the next instruction when the contents of register A is equal to the value n in the immediate field.

MACHINE INSTRUCTIONS (continued)

Type of instructions	Mnemonic	Instruction code D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
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 (PCL) ← a6–a0 (Note)
		1	0	p5	a6	a5	a4	a3	a2	a1	a0	2 p a +a			
	BLA p	0	0	0	0	0	1	0	0	0	0	0 1 0	2	2	(PCH) ← p (PCL) ← (DR2–DR0, A3–A0) (Note)
		1	0	p5	p4	0	0	p3	p2	p1	p0	2 p p			
Subroutine operation	BM a	0	1	0	a6	a5	a4	a3	a2	a1	a0	1 a a	1	1	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← 2 (PCL) ← a6–a0
	BML p, a	0	0	1	1	0	p4	p3	p2	p1	p0	0 C p +p	2	2	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← p (PCL) ← a6–a0 (Note)
		1	0	p5	a6	a5	a4	a3	a2	a1	a0	2 p 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 (PCL) ← (DR2–DR0, A3–A0) (Note)
		1	0	p5	p4	0	0	p3	p2	p1	p0	2 p p			
Return operation	RTI	0	0	0	1	0	0	0	1	1	0	0 4 6	1	1	(PC) ← (SK(SP)) (SP) ← (SP) – 1
	RT	0	0	0	1	0	0	0	1	0	0	0 4 4	1	2	(PC) ← (SK(SP)) (SP) ← (SP) – 1
	RTS	0	0	0	1	0	0	0	1	0	1	0 4 5	1	2	(PC) ← (SK(SP)) (SP) ← (SP) – 1
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	(EXF0) = 1 ? After skipping (EXF0) ← 0
	SNZ1	0	0	0	0	1	1	1	0	0	1	0 3 9	1	1	(EXF1) = 1 ? After skipping (EXF1) ← 0

Note : p is 0 to 63 for M34510E8, p is 0 to 31 for M34510M4A, and p is 0 to 15 for M34510M2A.

Skip condition	Carry flag CY	Datailed description
–	–	Branch within a page : Branches to address a in the identical page.
–	–	Branch out of a page : Branches to address a in page p.
–	–	Branch out of a page : Branches to address $(DR_2\ DR_1\ DR_0\ A_3\ A_2\ A_1\ A_0)_2$ specified by registers D and A in page p.
–	–	Call the subroutine in page 2 : Calls the subroutine at address a in page 2.
–	–	Call the subroutine : Calls the subroutine at address a in page p.
–	–	Call the subroutine : Calls the subroutine at address $(DR_2\ DR_1\ DR_0\ A_3\ A_2\ A_1\ A_0)_2$ specified by registers D and A in page p.
–	–	Returns from interrupt service routine to main routine. Returns each value of data pointer (X, Y, Z), carry flag, skip status, NOP mode status by the continuous description of the LA/LXY instruction, register A and register B to the states just before interrupt.
–	–	Returns from subroutine to the routine called the subroutine.
Skip at uncondition	–	Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition.
–	–	Clears (0) to the interrupt enable flag INTE, and disables the interrupt.
–	–	Sets (1) to the interrupt enable flag INTE, and enables the interrupt.
(EXF0) = 1	–	Skips the next instruction when the contents of EXF0 flag is "1." After skipping, clears the EXF0 flag.
(EXF1) = 1	–	Skips the next instruction when the contents of EXF1 flag is "1." After skipping, clears the EXF1 flag.

MACHINE INSTRUCTIONS (continued)

Parameter / Type of instructions	Mnemonic	Instruction code										Hexadecimal notation	Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				
Interrupt operation	SNZI0	0	0	0	0	1	1	1	0	1	0	0 3 A	1	1	$I1_2$ = 1 : (INT0) = "H" ? $I1_2$ = 0 : (INT0) = "L" ?
	SNZI1	0	0	0	0	1	1	1	0	1	1	0 3 B	1	1	$I2_3$ = 1 : (INT1) = "H" ? $I2_3$ = 0 : (INT1) = "L" ?
	TAV1	0	0	0	1	0	1	0	1	0	0	0 5 4	1	1	(A) ← (V1)
	TV1A	0	0	0	0	1	1	1	1	1	1	0 3 F	1	1	(V1) ← (A)
	TAV2	0	0	0	1	0	1	0	1	0	1	0 5 5	1	1	(A) ← (V2)
	TV2A	0	0	0	0	1	1	1	1	1	0	0 3 E	1	1	(V2) ← (A)
	TAI1	1	0	0	1	0	1	0	0	1	1	2 5 3	1	1	(A) ← (I1)
	TI1A	1	0	0	0	0	1	0	1	1	1	2 1 7	1	1	(I1) ← (A)
	TAI2	1	0	0	1	0	1	0	1	0	0	2 5 4	1	1	(A) ← (I2)
	TI2A	1	0	0	0	0	1	1	0	0	0	2 1 8	1	1	(I2) ← (A)
Timer operation	TAW1	1	0	0	1	0	0	1	0	1	1	2 4 B	1	1	(A) ← (W1)
	TW1A	1	0	0	0	0	0	1	1	1	0	2 0 E	1	1	(W1) ← (A)
	TAW2	1	0	0	1	0	0	1	1	0	0	2 4 C	1	1	(A) ← (W2)
	TW2A	1	0	0	0	0	0	1	1	1	1	2 0 F	1	1	(W2) ← (A)
	TAW3	1	0	0	1	0	0	1	1	0	1	2 4 D	1	1	(A) ← (W3)
	TW3A	1	0	0	0	0	1	0	0	0	0	2 1 0	1	1	(W3) ← (A)
	TAW4	1	0	0	1	0	0	1	1	1	0	2 4 E	1	1	(A) ← (W4)
	TW4A	1	0	0	0	0	1	0	0	0	1	2 1 1	1	1	(W4) ← (A)
	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)

Skip condition	Carry flag CY	Datailed description
(INT0) = "H" However, $I1_2$ = 1	–	When bit 2 ($I1_2$) of register I1 is "1" : Skips the next instruction when the level of INT0 pin is "H."
(INT0) = "L" However, $I1_2$ = 0	–	When bit 2 ($I1_2$) of register I1 is "0" : Skips the next instruction when the level of INT0 pin is "L."
(INT1) = "H" However, $I2_3$ = 1	–	When bit 3 ($I2_3$) of register I2 is "1" : Skips the next instruction when the level of INT1 pin is "H."
(INT1) = "L" However, $I2_3$ = 0	–	When bit 3 ($I2_3$) of register I2 is "0" : Skips the next instruction when the level of INT1 pin is "L."
–	–	Transfers the contents of interrupt control register V1 to register A.
–	–	Transfers the contents of register A to interrupt control register V1.
–	–	Transfers the contents of interrupt control register V2 to register A.
–	–	Transfers the contents of register A to interrupt control register V2.
–	–	Transfers the contents of interrupt control register I1 to register A.
–	–	Transfers the contents of register A to interrupt control register I1.
–	–	Transfers the contents of interrupt control register I2 to register A.
–	–	Transfers the contents of register A to interrupt control register I2.
–	–	Transfers the contents of timer control register W1 to register A.
–	–	Transfers the contents of register A to timer control register W1.
–	–	Transfers the contents of timer control register W2 to register A.
–	–	Transfers the contents of register A to timer control register W2.
–	–	Transfers the contents of timer control register W3 to register A.
–	–	Transfers the contents of register A to timer control register W3.
–	–	Transfers the contents of timer control register W4 to register A.
–	–	Transfers the contents of register A to timer control register W4.
–	–	Transfers the contents of timer control register W6 to register A.
–	–	Transfers the contents of register A to timer control register W6.

MACHINE INSTRUCTIONS (continued)

Parameter / Type of instructions	Mnemonic	Instruction code										Hexadecimal notation	Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				
Timer operation	TAB1	1	0	0	1	1	1	0	0	0	0	2 7 0	1	1	(B) ← (T_{17}–T_{14}) (A) ← (T_{13}–T_{10})
	T1AB	1	0	0	0	1	1	0	0	0	0	2 3 0	1	1	(R_{17}–R_{14}) ← (B) (T_{17}–T_{14}) ← (B) (R_{13}–R_{10}) ← (A) (T_{13}–T_{10}) ← (A)
	TAB2	1	0	0	1	1	1	0	0	0	1	2 7 1	1	1	(B) ← (T_{27}–T_{24}) (A) ← (T_{23}–T_{20})
	T2AB	1	0	0	0	1	1	0	0	0	1	2 3 1	1	1	(R_{27}–R_{24}) ← (B) (T_{27}–T_{24}) ← (B) (R_{23}–R_{20}) ← (A) (T_{23}–T_{20}) ← (A)
	TAB3	1	0	0	1	1	1	0	0	1	0	2 7 2	1	1	(B) ← (T_{37}–T_{34}) (A) ← (T_{33}–T_{30})
	T3AB	1	0	0	0	1	1	0	0	1	0	2 3 2	1	1	(R_{37}–R_{34}) ← (B) (T_{37}–T_{34}) ← (B) (R_{33}–R_{30}) ← (A) (T_{33}–T_{30}) ← (A)
	TAB4	1	0	0	1	1	1	0	0	1	1	2 7 3	1	1	(B) ← (T_{47}–T_{44}) (A) ← (T_{43}–T_{40})
	T4AB	1	0	0	0	1	1	0	0	1	1	2 3 3	1	1	(R_{47}–R_{44}) ← (B) (T_{47}–T_{44}) ← (B) (R_{43}–R_{40}) ← (A) (T_{43}–T_{40}) ← (A)
	TR4AB	1	0	0	0	1	1	1	1	0	0	2 3 C	1	1	(R_{47}–R_{44}) ← (B) (R_{43}–R_{40}) ← (A)
	SNZT1	1	0	1	0	0	0	0	0	0	0	2 8 0	1	1	(T1F) = 1 ? After skipping (T1F) ← 0
	SNZT2	1	0	1	0	0	0	0	0	0	1	2 8 1	1	1	(T2F) = 1 ? After skipping (T2F) ← 0
	SNZT3	1	0	1	0	0	0	0	0	1	0	2 8 2	1	1	(T3F) = 1 ? After skipping (T3F) ← 0
	SNZT4	1	0	1	0	0	0	0	0	1	1	2 8 3	1	1	(T4F) = 1 ? After skipping (T4F) ← 0

Skip condition	Carry flag CY	Datailed description
–	–	Transfers the contents of timer 1 to registers A and B.
–	–	Transfers the contents of registers A and B to timer 1 and timer 1 reload register.
–	–	Transfers the contents of timer 2 to registers A and B.
–	–	Transfers the contents of registers A and B to timer 2 and timer 2 reload register.
–	–	Transfers the contents of timer 3 to registers A and B.
–	–	Transfers the contents of registers A and B to timer 3 and timer 3 reload register.
–	–	Transfers the contents of timer 4 to registers A and B.
–	–	Transfers the contents of registers A and B to timer 4 and timer 4 reload register.
–	–	Transfers the contents of registers A and B to timer 4 reload register.
(T1F) = 1	–	Skips the next instruction when the contents of T1F flag is "1." After skipping, clears T1F flag.
(T2F) =1	–	Skips the next instruction when the contents of T2F flag is "1." After skipping, clears T2F flag.
(T3F) = 1	–	Skips the next instruction when the contents of T3F flag is "1." After skipping, clears T3F flag.
(T4F) = 1	–	Skips the next instruction when the contents of T4F flag is "1." After skipping, clears T4F flag.

MACHINE INSTRUCTIONS (continued)

Type of instructions	Mnemonic	Instruction code D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
Input/Output operation	IAP0	1	0	0	1	1	0	0	0	0	0	2 6 0	1	1	(A) ← (P0)
	OP0A	1	0	0	0	1	0	0	0	0	0	2 2 0	1	1	(P0) ← (A)
	IAP1	1	0	0	1	1	0	0	0	0	1	2 6 1	1	1	(A) ← (P1)
	OP1A	1	0	0	0	1	0	0	0	0	1	2 2 1	1	1	(P1 ← (A)
	IAP2	1	0	0	1	1	0	0	0	1	0	2 6 2	1	1	(A_1–A_0) ← ($P2_1$–$P2_0$) (A_3, A_2) ← 0
	IAP3	1	0	0	1	1	0	0	0	1	1	2 6 3	1	1	(A) ← (P3)
	OP3A	1	0	0	0	1	0	0	0	1	1	2 2 3	1	1	(P3) ← (A)
	CLD	0	0	0	0	0	1	0	0	0	1	0 1 1	1	1	(D) ← 1
	RD	0	0	0	0	0	1	0	1	0	0	0 1 4	1	1	(D(Y)) ← 0 (Y) = 0 to 9
	SD	0	0	0	0	0	1	0	1	0	1	0 1 5	1	1	(D(Y)) ← 1 (Y) = 0 to 9
	SZD	0	0	0	0	1	0	0	1	0	0	0 2 4	2	2	(D(Y)) = 0 ? (Y) = 0 to 9
		0	0	0	0	1	0	1	0	1	1	0 2 B			
	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)
	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)
Serial I/O control operation	TABSI	1	0	0	1	1	1	1	0	0	0	2 7 8	1	1	(A) ← (SI_3–SI_0) (B) ← (SI_7–SI_4)
	TSIAB	1	0	0	0	1	1	1	0	0	0	2 3 8	1	1	(SI_3–SI_0) ← (A) (SI_7–SI_4) ← (B)
	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)
	SST	1	0	1	0	0	1	1	1	1	0	2 9 E	1	1	(SIOF) ← 0 Serial I/O starting
	SNZSI	1	0	1	0	0	0	1	0	0	0	2 8 8	1	1	(SIOF) = 1 ? After skipping (SIOF) ← 0

Skip condition	Carry flag CY	Datailed description
–	–	Transfers the input of port P0 to register A.
–	–	Outputs the contents of register A to port P0.
–	–	Transfers the input of port P1 to register A.
–	–	Outputs the contents of register A to port P1.
–	–	Transfers the input of port P2 to register A.
–	–	Transfers the input of port P3 to register A.
–	–	Outputs the contents 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 (Y) = 0 to 9	–	Skips the next instruction when a bit of port D specified by register Y is "0."
–	–	Transfers the contents of register A to key-on wakeup control register K0.
–	–	Transfers the contents of key-on wakeup control register K0 to register A.
–	–	Transfers the contents of register A to pull-up control register PU0.
–	–	Transfers the contents of pull-up control register PU0 to register A.
–	–	Transfers the contents of serial I/O register SI to registers A and B.
–	–	Transfers the contents of registers A and B to serial I/O register SI.
–	–	Transfers the contents of serial I/O mode register J1 to register A.
–	–	Transfers the contents of register A to serial I/O mode register J1.
–	–	Clears (0) to SIOF flag and starts serial I/O.
(SIOF) = 1	–	Skips the next instruction when the contents of SIOF flag is "1." After skipping, clears (0) to SIOF flag.

MACHINE INSTRUCTIONS (continued)

Parameter / Type of instructions	Mnemonic	Instruction code										Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			
A-D conversion operation	TABAD	1	0	0	1	1	1	1	0	0	1	279	1	1	(A) ← (AD_3–AD_0) (B) ← (AD_7–AD_4)
	TADAB	1	0	0	0	1	1	1	0	0	1	239	1	1	(AD_3–AD_0) ← (A) (AD_7–AD_4) ← (B)
	TAQ1	1	0	0	1	0	0	0	1	0	0	244	1	1	(A) ← (Q1)
	TQ1A	1	0	0	0	0	0	0	1	0	0	204	1	1	(Q1) ← (A)
	ADST	1	0	1	0	0	1	1	1	1	1	29F	1	1	(ADF) ← 0 A-D conversion starting
	SNZAD	1	0	1	0	0	0	0	1	1	1	287	1	1	(ADF) = 1 ? After skipping, (ADF) ← 0
	TAQ2	1	0	0	1	0	0	0	1	0	1	245	1	1	(A) ← (Q2)
	TQ2A	1	0	0	0	0	0	0	1	0	1	205	1	1	(Q2) ← (A)
Other operation	NOP	0	0	0	0	0	0	0	0	0	0	000	1	1	(PC) ← (PC) + 1
	POF	0	0	0	0	0	0	0	0	1	0	002	1	1	RAM back-up
	EPOF	0	0	0	1	0	1	1	0	1	1	05B	1	1	POF instruction valid
	SNZP	0	0	0	0	0	0	0	0	1	1	003	1	1	(P) = 1 ?
	WRST	1	0	1	0	1	0	0	0	0	0	2A0	1	1	(WDF) ← 0, (WEF) ← 1 (WDT) ← $FFFF_{16}$
	RTPS	1	0	1	0	1	0	0	0	0	1	2A1	1	1	(RTP) ← 1
	RTPR	1	0	1	0	1	0	0	0	1	0	2A2	1	1	(RTP) ← 0
	TRTPA	1	0	0	0	0	1	1	0	0	1	219	1	1	(RTP) ← (A_0)
	TRTRA	1	0	1	0	0	1	0	0	0	1	291	1	1	(RTR_1, RTR_0) ← (A_1, A_0)
	TVLRA	1	0	1	0	0	1	0	0	0	0	290	1	1	(VLR) ← (A_0)

Skip condition	Carry flag CY	Datailed description
–	–	Transfers the contents of register AD (comparator register at the comparator mode) to registers A and B.
–	–	Transfers the contents of registers A and B to the comparator register at the comparator mode.
–	–	Transfers the contents of the A-D control register Q1 to register A.
–	–	Transfers the contents of register A to the A-D control register Q1.
–	–	Clears the ADF flag, and the A-D conversion at the A-D conversion mode or the comparator operation at the comparator mode is started.
(ADF) = 1	–	Skips the next instruction when the contents of ADF flag is "1". After skipping, clears (0) the contents of ADF flag.
–	–	Transfers the contents of the A-D control register Q2 to register A.
–	–	Transfers the contents of register A to the A-D control register Q2.
–	–	No operation
–	–	Puts the system in RAM back-up state by executing the POF instruction after executing the EPOF instruction.
–	–	Makes the immediate POF instruction by executing the EPOF instruction.
(P) = 1	–	Skips the next instruction when P flag is "1". After skipping, P flag remains unchanged.
–	–	Operates the watchdog timer and initializes the watchdog timer flag WDF and the watchdog timer WDT.
–	–	Sets (1) the real time output latch RTP.
–	–	Clears (0) the real time output latch RTP.
–	–	Transfers the contents of register A to the real time output register RTP.
–	–	Transfers the contents of register A to the real time output control register RTR.
–	–	Transfers the contents of register A to the voltage drop detection circuit control register VLR.

CONTROL REGISTERS

Interrupt control register V1		at reset : 0000_2	at RAM back-up : 0000_2	R/W
$V1_3$	Timer 2 interrupt enable bit	0	Interrupt disabled (SNZT2 instruction is valid)	
		1	Interrupt enabled (SNZT2 instruction is invalid)	
$V1_2$	Timer 1 interrupt enable bit	0	Interrupt disabled (SNZT1 instruction is valid)	
		1	Interrupt enabled (SNZT1 instruction is invalid)	
$V1_1$	External 1 interrupt enable bit	0	Interrupt disabled (SNZ1 instruction is valid)	
		1	Interrupt enabled (SNZ1 instruction is invalid)	
$V1_0$	External 0 interrupt enable bit	0	Interrupt disabled (SNZ0 instruction is valid)	
		1	Interrupt enabled (SNZ0 instruction is invalid)	
Interrupt control register V2		at reset : 0000_2	at RAM back-up : 0000_2	R/W
$V2_3$	Serial I/O interrupt enable bit	0	Interrupt disabled (SNZSI instruction is valid)	
		1	Interrupt enabled (SNZSI instruction is invalid)	
$V2_2$	A-D interrupt enable bit	0	Interrupt disabled (SNZAD instruction is valid)	
		1	Interrupt enabled (SNZAD instruction is invalid)	
$V2_1$	Timer 4 interrupt enable bit	0	Interrupt disabled (SNZT4 instruction is valid)	
		1	Interrupt enabled (SNZT4 instruction is invalid)	
$V2_0$	Timer 3 interrupt enable bit	0	Interrupt disabled (SNZT3 instruction is valid)	
		1	Interrupt enabled (SNZT3 instruction is invalid)	
Interrupt control register I1		at reset : 0000_2	at RAM back-up : state retained	R/W
$I1_3$	D_6/INT0/ZEROX pin function selection bit	0	INT0/D_6 (input/output)	
		1	ZEROX/D_6 (input/output)	
$I1_2$	Interrupt valid waveform for INT0 pin/ return level selection bit (Note 2)	0	Falling waveform ("L" level of INT0 pin is recognized with the SNZI0 instruction)/"L" level	
		1	Rising waveform ("H" level of INT0 pin is recognized with the SNZI0 instruction)/"H" level	
$I1_1$	Edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I1_0$	Noise elimination circuit control bit	0	Disabled	
		1	Enabled	
Interrupt control register I2		at reset : 0000_2	at RAM back-up : state retained	R/W
$I2_3$	Interrupt valid waveform for INT1 pin selection bit (Note 2)	0	Falling waveform ("L" level of INT1 pin is recognized with the SNZI1 instruction.)	
		1	Rising waveform ("H" level of INT1 pin is recognized with the SNZI1 instruction.)	
$I2_2$	Zero cross input flag control bit	0	Zero cross input flag cleared (set impossible)	
		1	Zero cross input flag set possible	
$I2_1$	Noise elimination circuit sampling clock control bit	0	Stop	
		1	Operating	
$I2_0$	Noise elimination circuit sampling clock selection bit	0	Instruction clock divided by 8	
		1	Instruction clock divided by 32	

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

2: When the contents of $I1_2$ and $I2_3$ are changed, the external interrupt request flags EXF0 and EXF1 may be set, respectively. Accordingly, clear the EXF0 and EXF1 flags to "0" with the SNZ0 and SNZ1 instructions, respectively.

CONTROL REGISTERS (continued)

Timer control register W1		at reset : 0000_2		at RAM back-up : 0000_2	R/W
$W1_3$	Prescaler control bit	0	Stop (state initialized)		
		1	Operating		
$W1_2$	Prescaler dividing ratio selection bit	0	Instruction clock divided by 2		
		1	Instruction clock divided by 4		
$W1_1$	Timer 1 control bit	0	Stop (state retained)		
		1	Operating		
$W1_0$	Not used	0	This bit has no function, but read/write is enabled.		
		1			

Timer control register W2		at reset : 0000_2		at RAM back-up : state retained	R/W
$W2_3$	Timer 2 control bit	0	Stop (state retained)		
		1	Operating		
$W2_2$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$W2_1$	Timer 2 count source selection bits	$W2_1$	$W2_0$	Count source	
		0	0	Timer 1 underflow signal	
$W2_0$		0	1	Prescaler output	
		1	0	CNTR input	
		1	1	Not available	

Timer control register W3		at reset : 0000_2		at RAM back-up : state retained	R/W
$W3_3$	Timer 3 control bit	0	Stop (state retained)		
		1	Operating		
$W3_2$	Timer 3 count start synchronous circuit control bit	0	Count start synchronous circuit not selected		
		1	Count start synchronous circuit selected		
$W3_1$	Timer 3 count source selection bits	$W3_1$	$W3_0$	Count source	
		0	0	Timer 2 underflow signal	
$W3_0$		0	1	Prescaler output	
		1	0	Not available	
		1	1	Not available	

Timer control register W4		at reset : 0000_2		at RAM back-up : state retained	R/W
$W4_3$	Timer 4 control bit	0	Stop (state retained)		
		1	Operating		
$W4_2$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$W4_1$	Timer 4 count source selection bits	$W4_1$	$W4_0$	Count source	
		0	0	Timer 3 underflow signal	
$W4_0$		0	1	Prescaler output	
		1	0	CNTR input	
		1	1	Not available	

Timer control register W6		at reset : 0000_2		at RAM back-up : state retained	R/W
$W6_3$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$W6_2$	CNTR output control bit	0	Output stop		
		1	Output		
$W6_1$	CNTR output selection bit	0	Timer 1 underflow signal divided by 2		
		1	X_{IN} divided by 2		
$W6_0$	D_7/CNTR function selection bit	0	D_7(I/O)/CNTR input		
		1	CNTR output/D_7(input)		

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

CONTROL REGISTERS (continued)

Serial I/O mode register J1		at reset : 0000_2	at RAM back-up : state retained	R/W
$J1_3$	Not used	0 / 1	This bit has no function, but read/write is enabled.	
$J1_2$	Serial I/O internal clock dividing ratio selection bit	0	Instruction clock signal divided by 8	
		1	Instruction clock signal divided by 4	
$J1_1$	Serial I/O port selection bit	0	I/O ports D_8, D_9, and input port $P2_0$ selected	
		1	Serial I/O ports S_{CK}, S_{OUT} and S_{IN}/input ports D_8, D_9 and $P2_0$ selected	
$J1_0$	Serial I/O synchronous clock selection bit	0	External clock	
		1	Internal clock (instruction clock divided by 4 or 8)	

A-D control register Q1		at reset : 0000_2	at RAM back-up : state retained	R/W
$Q1_3$	A-D operating mode control bit	0	A-D conversion mode	
		1	Comparator mode	
$Q1_2$ / $Q1_1$ / $Q1_0$	Analog input pin selection bits (Note 2)	$Q1_2$ $Q1_1$ $Q1_0$	Selected pins	
		0 0 0	A_{IN0}	
		0 0 1	A_{IN1}	
		0 1 0	A_{IN2}	
		0 1 1	A_{IN3}	
		1 0 0	Not available	
		1 0 1	Not available	
		1 1 0	Not available	
		1 1 1	Not available	

A-D control register Q2		at reset : 0000_2	at RAM back-up : state retained	R/W
$Q2_3$	Not used	0 / 1	This bit has no function, but read/write is enabled.	
$Q2_2$	$P3_3/A_{IN3}$ and $P3_2/A_{IN2}$ pin function selection bit	0	$P3_3$, $P3_2$ (I/O)	
		1	A_{IN3}, $A_{IN2}/P3_3$, $P3_2$ (output)	
$Q2_1$	$P3_1/A_{IN1}$ pin function selection bit	0	$P3_1$ (I/O)	
		1	$A_{IN1}/P3_1$ (output)	
$Q2_0$	$P3_0/A_{IN0}$ pin function selection bit	0	$P3_0$ (I/O)	
		1	$A_{IN0}/P3_0$ (output)	

Notes 1: "R" represents read enabled, "W" represents write enabled.
2: Select A_{IN0}–A_{IN3} with register Q1 after setting register Q2.

CONTROL REGISTERS (continued)

Key-on wakeup control register K0		at reset : 0000_2	at RAM back-up : state retained	R/W
$K0_3$	Pins $P1_2$ and $P1_3$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_2$	Pins $P1_0$ and $P1_1$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_1$	Pins $P0_2$ and $P0_3$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_0$	Pins $P0_0$ and $P0_1$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
Pull-up control register PU0		at reset : 0000_2	at RAM back-up : state retained	R/W
$PU0_3$	Pins $P1_2$ and $P1_3$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
$PU0_2$	Pins $P1_0$ and $P1_1$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
$PU0_1$	Pins $P0_2$ and $P0_3$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
$PU0_0$	Pins $P0_0$ and $P0_1$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
Voltage drop detection circuit control register VLR		at reset : 0_2	at RAM back-up : state retained	W
VLR_0	Voltage drop detection circuit control bit	0	Voltage drop detection circuit stop	
		1	Voltage drop detection circuit operating	
Real time output control register RTR		at reset : 00_2	at RAM back-up : state retained	W
RTR_1	D_5/real time output function selection bit	0	D_5 (I/O)	
		1	Real time output/D_5 (input)	
RTR_0	D_4/real time output function selection bit	0	D_4 (I/O)	
		1	Real time output/D_4 (input)	

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

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions		Ratings	Unit
V_{DD}	Supply voltage			−0.3 to 7.0	V
V_I	Input voltage P0, P1, P2, P3, $\overline{RESET}$, X_{IN}			−0.3 to V_{DD}+0.3	V
V_I	Input voltage D_0–D_5, D_7–D_9			−0.3 to 13	V
V_I	Input voltage AIN_0–AIN_3, V_{REF}			−0.3 to V_{DD}+0.3	V
V_I	Input voltage D_6/INT0/ZEROX			−0.7 to V_{DD}+0.7	V
I_I	Input current D_6/INT0/ZEROX			−100 to 100	μA
V_O	Output voltage D_6	Output transistors in cut-off state		−0.7 to V_{DD}+0.7	V
V_O	Output voltage P0, P1, P3, $\overline{RESET}$			−0.3 to V_{DD}+0.3	V
V_O	Output voltage D_0–D_5, D_7–D_9			−0.3 to 13	V
V_O	Output voltage X_{OUT}			−0.3 to V_{DD}+0.3	V
Pd	Power dissipation	Ta = 25 °C	M34510MxA-XXXSP	1100	mW
			M34510MxA-XXXFP	300	
Topr	Operating temperature range	(Note)		−20 to 85	°C
Tstg	Storage temperature range			−40 to 125	°C

Note: −20 °C to 70 °C for the M34510E8SS.

NOTE ON THE ZERO CROSS INPUT

The high-voltage is supplied to V_{DD} and Vss through a parasitic diode in the microcomputer when the high-voltage is applied to port D_6 while V_{DD} and Vss are open. Apply the voltage applied to port D_6 within the limits of −0.7 V to 7.7 V in order to prevent the destruction of a microcomputer by applying the high-voltage.
And keep the current to port D_6 within the limits of −100 μA to 100 μA.

RECOMMENDED OPERATING CONDITIONS 1

(Mask ROM version:Ta = −20 °C to 85 °C, V_{DD} = 2.0 V to 5.5 V, unless otherwise noted)
(One Time PROM version:Ta = −20 °C to 85 °C, V_{DD} = 2.5 V to 5.5 V, unless otherwise noted)
(Built-in EPROM version:Ta = −20 °C to 70 °C, V_{DD} = 2.5 V to 5.5 V, unless otherwise noted)

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{DD}	Supply voltage (Note 1) (Note 2) (Note 3)	Mask ROM version	$f(X_{IN}) \leq 6.0$ MHz	4.5	5.0	5.5	V
			$f(X_{IN}) \leq 4.0$ MHz	4.0		5.5	
			$f(X_{IN}) \leq 2.0$ MHz	2.5		5.5	
			$f(X_{IN}) \leq 1.5$ MHz	2.0		5.5	
		One Time PROM version Built-in EPROM version	$f(X_{IN}) \leq 6.0$ MHz	4.5	5.0	5.5	
			$f(X_{IN}) \leq 4.0$ MHz	4.0		5.5	
			$f(X_{IN}) \leq 2.0$ MHz	2.5		5.5	
V_{RAM}	RAM back-up voltage (at RAM back-up mode)	Mask ROM version		1.8		5.5	V
		One Time PROM version Built-in EPROM version		2.0		5.5	
Vss	Supply voltage				0		V
V_{IH}	"H" level input voltage	P0, P1, P2, P3, D_6, X_{IN}		0.8V_{DD}		V_{DD}	V
V_{IH}	"H" level input voltage	D_0–D_5, D_7–D_9		0.8V_{DD}		12	V
V_{IH}	"H" level input voltage	$\overline{RESET}$		0.85V_{DD}		V_{DD}	V
V_{IH}	"H" level input voltage	CNTR, S_{IN}, S_{CK}, INT0, INT1		0.85V_{DD}		V_{DD}	V
V_{IL}	"L" level input voltage	P0, P1, P2, P3, D_0–D_9, X_{IN}		0		0.2V_{DD}	V
V_{IL}	"L" level input voltage	$\overline{RESET}$		0		0.1V_{DD}	V
V_{IL}	"L" level input voltage	CNTR, S_{IN}, S_{CK}, INT0, INT1		0		0.15V_{DD}	V
I_{OL}(peak)	"L" level peak output current	P3, $\overline{RESET}$				10	mA
I_{OL}(peak)	"L" level peak output current	D_4, D_5				40	mA
I_{OL}(peak)	"L" level peak output current	D_0–D_3, D_6–D_9				24	mA
I_{OL}(peak)	"L" level peak output current	P0, P1				24	mA
I_{OL}(avg)	"L" level average output current	P3, $\overline{RESET}$				5	mA
I_{OL}(avg)	"L" level average output current	D_4, D_5				30	mA
I_{OL}(avg)	"L" level average output current	D_0–D_3, D_6–D_9				15	mA
I_{OL}(avg)	"L" level average output current	P0, P1				12	mA
ΣI_{OL}(avg)	"L" level total average current	D, $\overline{RESET}$				80	mA
ΣI_{OL}(avg)	"L" level total average current	P0, P1, P3				60	mA

Notes 1: When using the A-D converter, refer to the recommended operating conditions for the A-D converter.
2: When using the zero cross detection circuit, refer to the recommended operating conditions for the zero cross detection circuit.
3: When using the voltage drop detection circuit, refer to the recommended operating conditions for the voltage drop detection circuit.

RECOMMENDED OPERATING CONDITIONS 2

(Mask ROM version:Ta = −20 °C to 85 °C, V_{DD} = 2.0 V to 5.5 V, unless otherwise noted)
(One Time PROM version:Ta = −20 °C to 85 °C, V_{DD} = 2.5 V to 5.5 V, unless otherwise noted)
(Built-in EPROM version:Ta = −20 °C to 70 °C, V_{DD} = 2.5 V to 5.5 V, unless otherwise noted)

Symbol	Parameter	Conditions			Limits			Unit
					Min.	Typ.	Max.	
f(XIN)	System clock frequency (with a ceramic resonator) (Note 1) (Note 2) (Note 3)	Mask ROM version	V_{DD} = 4.5 V to 5.5 V				6.0	MHz
			V_{DD} = 4.0 V to 5.5 V				4.0	
			V_{DD} = 2.5 V to 5.5 V				2.0	
			V_{DD} = 2.0 V to 5.5 V				1.5	
		One Time PROM version Built-in EPROM version	V_{DD} = 4.5 V to 5.5 V				6.0	
			V_{DD} = 4.0 V to 5.5 V				4.0	
			V_{DD} = 2.5 V to 5.5 V				2.0	
f(XIN)	System clock frequency (with external clock input) (Note 1) (Note 2) (Note 3)	Mask ROM version	V_{DD} = 4.5 V to 5.5 V	40 to 60 %			4.0	MHz
			V_{DD} = 4.0 V to 5.5 V				3.0	
			V_{DD} = 2.5 V to 5.5 V	30 to 70 %			1.0	
			V_{DD} = 2.0 V to 5.5 V				0.8	
		One Time PROM version Built-in EPROM version	V_{DD} = 4.5 V to 5.5 V	40 to 60 %			4.0	
			V_{DD} = 4.0 V to 5.5 V				3.0	
			V_{DD} = 2.5 V to 5.5 V	30 to 70 %			1.0	
tw(SCK)	Serial I/O external clock period ("H" and "L" pulse width)	Mask ROM version	V_{DD} = 4.5 V to 5.5 V		500			ns
			V_{DD} = 4.0 V to 5.5 V		750			
			V_{DD} = 2.5 V to 5.5 V		1.5			µs
			V_{DD} = 2.0 V to 5.5 V		2.0			
		One Time PROM version Built-in EPROM version	V_{DD} = 4.5 V to 5.5 V		500			ns
			V_{DD} = 4.0 V to 5.5 V		750			
			V_{DD} = 2.5 V to 5.5 V		1.5			µs
tw(CNTR)	Timer external input period ("H" and "L" pulse width)	Mask ROM version	V_{DD} = 4.5 V to 5.5 V		500			ns
			V_{DD} = 4.0 V to 5.5 V		750			
			V_{DD} = 2.5 V to 5.5 V		1.5			µs
			V_{DD} = 2.0 V to 5.5 V		2.0			
		One Time PROM version Built-in EPROM version	V_{DD} = 4.5 V to 5.5 V		500			ns
			V_{DD} = 4.0 V to 5.5 V		750			
			V_{DD} = 2.5 V to 5.5 V		1.5			µs

Notes 1: When using the A-D converter, refer to the recommended operating conditions for the A-D converter.
2: When using the zero cross detection circuit, refer to the recommended operating conditions for the zero cross detection circuit.
3: When using the voltage drop detection circuit, refer to the recommended operating conditions for the voltage drop detection circuit.

ELECTRICAL CHARACTERISTICS

(Mask ROM version:Ta = −20 °C to 85 °C, V_{DD} = 2.0 V to 5.5 V, unless otherwise noted)
(One Time PROM version:Ta = −20 °C to 85 °C, V_{DD} = 2.5 V to 5.5 V, unless otherwise noted)
(Built-in EPROM version:Ta = −20 °C to 70 °C, V_{DD} = 2.5 V to 5.5 V, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OL}	"L" level output voltage P0, P1	V_{DD} = 5 V	I_{OL} = 12 mA			2	V
		V_{DD} = 3 V	I_{OL} = 6 mA			0.9	
V_{OL}	"L" level output voltage P3, $\overline{RESET}$	V_{DD} = 5 V	I_{OL} = 5 mA			2	V
		V_{DD} = 3 V	I_{OL} = 2 mA			0.9	
V_{OL}	"L" level output voltage D_4, D_5	V_{DD} = 5 V	I_{OL} = 30 mA			1	V
			I_{OL} = 10 mA			0.4	
		V_{DD} = 3 V	I_{OL} = 15 mA			1	V
			I_{OL} = 5 mA			0.4	
V_{OL}	"L" level output voltage D_0–D_3, D_6–D_9	V_{DD} = 5 V	I_{OL} = 15 mA			2	V
		V_{DD} = 3 V	I_{OL} = 3 mA			0.9	
I_{IH}	"H" level input current P0, P1, P2, P3, D_6, $\overline{RESET}$	V_I = V_{DD} and port P3 selected				1	μA
I_{IH}	"H" level input current D_0–D_5, D_7–D_9	V_I = 12 V				1	μA
I_{IL}	"L" level input current P0, P1, P2, P3, D_6, $\overline{RESET}$	V_I = 0 V No pull-up of ports P0 and P1, port P3 selected		−1			μA
I_{IL}	"L" level input current D_0–D_5, D_7–D_9	V_I = 0 V		−1			μA
I_{OZ}	Output current at off-state D_0–D_5, D_7–D_9	V_O = 12 V				1	μA
I_{OZ}	Output current at off-state P0, P1, P3, D_6, $\overline{RESET}$	V_O = V_{DD}				1	μA
I_{DD}	Supply current, at active mode	V_{DD} = 5 V	f(X_{IN}) = 6.0 MHz		4	12	mA
			f(X_{IN}) = 500 kHz		0.9	2.7	
		V_{DD} = 3 V	f(X_{IN}) = 2.0 MHz		0.9	2.7	
			f(X_{IN}) = 500 kHz		0.3	0.9	
	at RAM back-up mode	Ta = 25 °C			0.1	1	μA
		V_{DD} = 5 V				10	
		V_{DD} = 3 V				6	
R_{PU}	Pull-up resistor value P0, P1	V_{DD} = 5 V	V_I = 0 V	20	50	125	kΩ
		V_{DD} = 3 V		40	100	250	
$V_{T+} - V_{T-}$	Hysteresis INT0, INT1, CNTR, S_{IN}, S_{CK}				0.3		V
$V_{T+} - V_{T-}$	Hysteresis $\overline{RESET}$	V_{DD} = 5 V			1.8		V
		V_{DD} = 3 V			0.7		

RECOMMENDED OPERATING CONDITIONS FOR A-D CONVERTER

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

Symbol	Parameter	Conditions			Limits			Unit
					Min.	Typ.	Max.	
V_{DD}	Supply voltage (Notes 1, 2)	$f(X_{IN})$ = 0.4 MHz to 6 MHz			4.5	5.0	5.5	V
		$f(X_{IN})$ = 0.4 MHz to 4 MHz			4.0		5.5	
		$f(X_{IN})$ = 0.4 MHz to 2 MHz			2.7		5.5	
AVss	Analog supply voltage	AVss = Vss				0		V
V_{REF}	Reference input voltage				2.0		V_{DD}	V
V_{IA}	Analog input voltage				0		V_{REF}	V
$f(X_{IN})$	System clock frequency (with a ceramic resonator) (Note 2)	V_{DD} = 4.5 V to 5.5 V			0.4		6.0	MHz
		V_{DD} = 4.0 V to 5.5 V			0.4		4.0	
		V_{DD} = 2.7 V to 5.5 V			0.4		2.0	
$f(X_{IN})$	System clock frequency (with external clock input) (Note 2)	V_{DD} = 4.5 V to 5.5 V	Duty ratio	40 % to 60 %	0.4		4.0	MHz
		V_{DD} = 4.0 V to 5.5 V			0.4		3.0	
		V_{DD} = 2.7 V to 5.5 V		30 % to 70 %	0.4		1.0	

Notes 1: When using the zero cross detection circuit, refer to the recommended operating conditions for the zero cross detection circuit.

2: When using the voltage drop detection circuit, refer to the recommended operating conditions for the voltage drop detection circuit.

CHARACTERISTICS FOR A-D CONVERTER

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

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
I_{ADD}	A–D operating current	V_{DD} = 5.0 V	$f(X_{IN})$ = 0.4 MHz to 6 MHz		0.6	1.8	mA
		V_{DD} = 3.0 V	$f(X_{IN})$ = 0.4 MHz to 2 MHz		0.1	0.3	
I_{VREF}	Reference input current	V_{REF} = 5.0 V		50	150	200	μA
		V_{REF} = 3.0 V		30	90	120	
—	Resolution					8	bit
—	Absolute accuracy	V_{DD} = V_{REF} = 5.0 V	Quantization error excepted			±2	LSB
		V_{DD} = V_{REF} = 3.0 V					
T_{CONV}	Conversion time	$f(X_{IN})$ = 6 MHz				25	μs
		$f(X_{IN})$ = 4 MHz				37.5	
		$f(X_{IN})$ = 2 MHz				75	

RECOMMENDED OPERATING CONDITIONS FOR ZERO CROSS DETECTION CIRCUIT

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

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{DD}	Supply voltage (Note 1)	$f(X_{IN}) \leq 6$ MHz		4.5	5.0	5.5	V
		$f(X_{IN}) \leq 4$ MHz		4.0		5.5	
Vss	Supply voltage				0		V
$f(X_{IN})$	System clock frequency (Notes 1, 2) (with a ceramic resonator)	V_{DD} = 4.5 V to 5.5 V				6.0	MHz
		V_{DD} = 4.0 V to 5.5 V				4.0	
$f(X_{IN})$	System clock frequency (Notes 1, 2) (with external clock input)	V_{DD} = 4.5 V to 5.5 V	Duty ratio			4.0	MHz
		V_{DD} = 4.0 V to 5.5 V	40 to 60 %			3.0	
f_{ZEROX}	Input frequency for zero cross detection				50, 60	1000	Hz

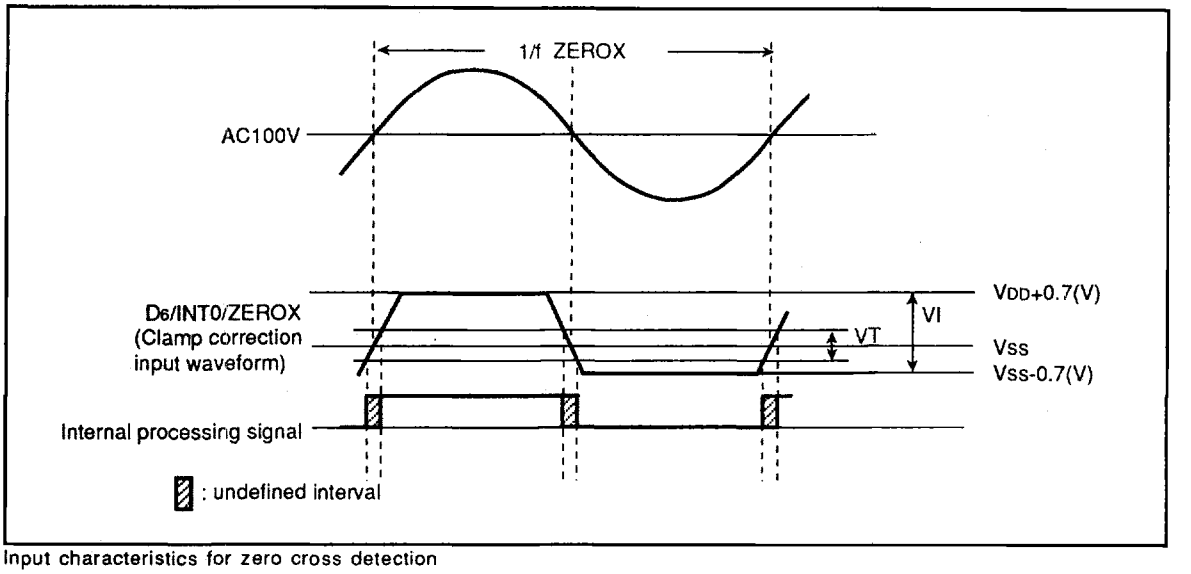
Notes 1: When using the voltage drop detection circuit, refer to the recommended operating conditions for voltage drop detection circuit.

2: When using the A-D converter, refer to the recommended operating conditions for the A-D converter.

CHARACTERISTICS FOR ZERO CROSS DETECTION CIRCUIT

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
I_{ZDD}	Zero cross comparator operating current	V_{DD} = 5.0 V		1.0	3.0	mA
V_T	Zero cross detection decision voltage	50 Hz or 60 Hz	−200	0	200	mV



Input characteristics for zero cross detection

RECOMMENDED OPERATING CONDITIONS FOR VOLTAGE DROP DETECTION CIRCUIT

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

Symbol	Parameter	Conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{DD}	Supply voltage		4.5	5.0	5.5	V
$f(X_{IN})$	System clock frequency (Note)			1.5	2.0	MHz

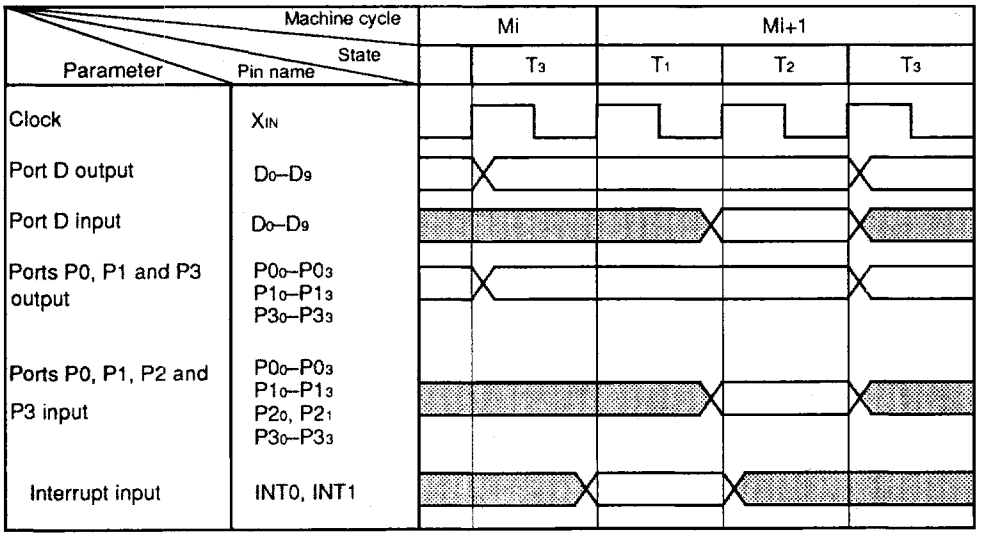
Note: When using the A-D converter, refer to the recommended operating conditions for the A-D converter.

CHARACTERISTICS FOR VOLTAGE DROP DETECTION CIRCUIT

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{RST}	Detection voltage	$f(X_{IN}) \leq 2.0$ MHz	1.5	3.0	4.5	V
I_{RST}	Supply current when operating the voltage drop detection circuit	V_{DD} = 5.0 V at RAM back-up state			20	μA

BASIC TIMING DIAGRAM



MITSUBISHI ELECTRIC

BUILT-IN PROM VERSION

In addition to the mask ROM versions, the 4510 Group has programmable ROM versions software compatible with mask ROM. One is the window-type EPROM version supplied with a built-in EPROM which can be written to and erased. Others are the One Time PROM versions whose PROMs can only be written to and not be erased. Since the functions of the built-in EPROM and One Time PROM versions are exactly the same, except erasure, all of them referred to as built-in PROM versions in this explanation, unless otherwise noted.

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

Table 22 shows the product of built-in PROM version. Figure 52 and 53 show the pin configurations of built-in PROM versions. The One Time PROM versions have pin-compatibility with the mask ROM version. The built-in EPROM version has different outline.

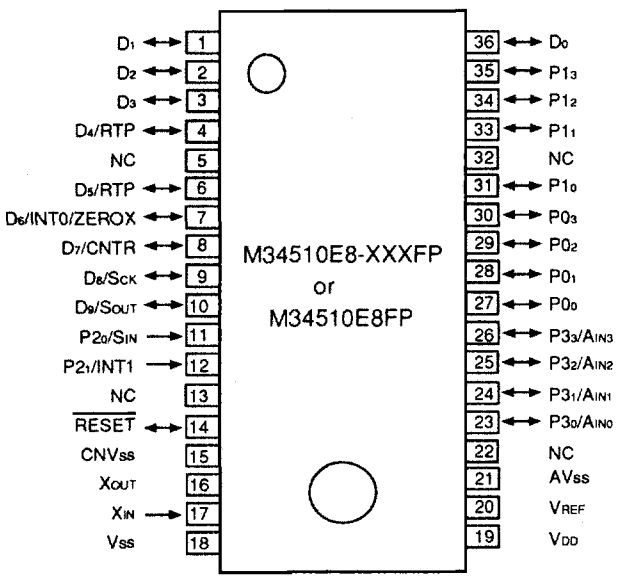
The built-in EPROM version is the microcomputer for program development. Use this microcomputer only for program development and prototype test.

Table 22 Product of built-in PROM version

Product	PROM size (X 10 bits)	RAM size (X 4 bits)	Package	ROM type
M34510E8-XXXSP/FP	8192 words	384 words	SP: 32P4B FP: 36P2R-A	One Time PROM [shipped after writing] (shipped after writing and test in factory)
M34510E8SP/FP				One Time PROM [shipped in blank]
M34510E8SS *			SS: 32S1B	Built-in EPROM version

*: For program development

PIN CONFIGURATION (TOP VIEW)

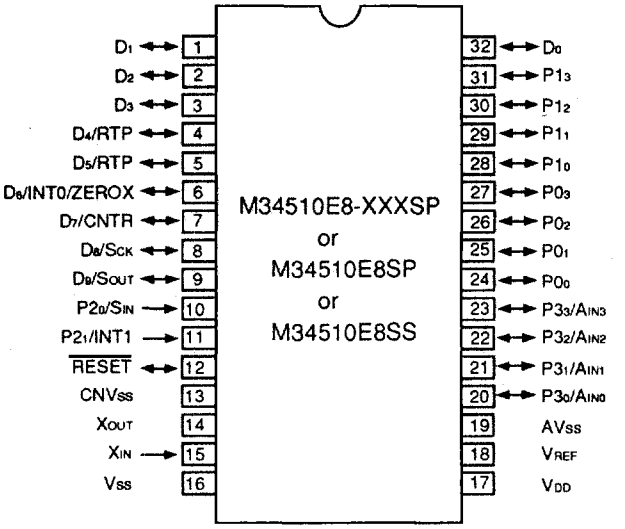


NC : No connected
(not connected to internal circuit)

Outline 36P2R-A (One Time PROM version)

Fig. 52 Pin configuration of built-in PROM version

PIN CONFIGURATION (TOP VIEW)



Outline 32P4B (One Time PROM version)
32S1B (Built-in EPROM version)

Fig. 53 Pin configuration of built-in PROM version (continued)

MITSUBISHI ELECTRIC

(1) PROM mode

Each built-in PROM version has a PROM mode in addition to a normal operation mode. The PROM mode is used to write to and read from the built-in PROM.

In the PROM mode, the programming adapter can be used with a general-purpose PROM programmer to write to or read from the built-in PROM as if it were M5M27C256K. Programming adapters are listed in Table 23.

- Writing and reading of built-in PROM

  Programming voltage is 12.5 V. Write the program in the PROM of the built-in PROM version as shown in Figure 54.
- Erasing

  Only the built-in EPROM (M34510E8SS) version has a transparent window for erasing on the top surface of the package. The EPROM is erased when it is exposed to ultraviolet light with a wavelength of 2537Å to an integrated dose of 15 W•s/cm² or more through the window.

(2) Notes on handling

① Sunlight and fluorescent lamp contain light that can erase written information. Be sure to cover the transparent glass portion with a seal or other similar materials except when erasing.

② Mitsubishi Electric corp. provides the seal for covering the transparent glass. Take care that the seal does not touch lead pins.

③ Clean the transparent glass before erasing. Fingers' fat and paste disturb the passage of ultraviolet light and may affect badly the erasure capability.

④ A high-voltage is used for writing. Take care that overvoltage is not applied. Take care especially at turning on the power.

⑤ For the One Time PROM version shipped in blank, Mitsubishi Electric corp. does not perform PROM writing test and screening in the assembly process and following processes. In order to improve reliability after writing, performing writing and test according to the flow shown in Figure 55 before using is recommended (Products shipped in blank: PROM contents is not written in factory when shipped)

Table 23 Programming adapters

Microcomputer	Programming adapter
M34510E8-XXXSP, M34510E8SP, M34510E8SS	PCA4764
M34510E8-XXXFP, M34510E8FP	PCA4765

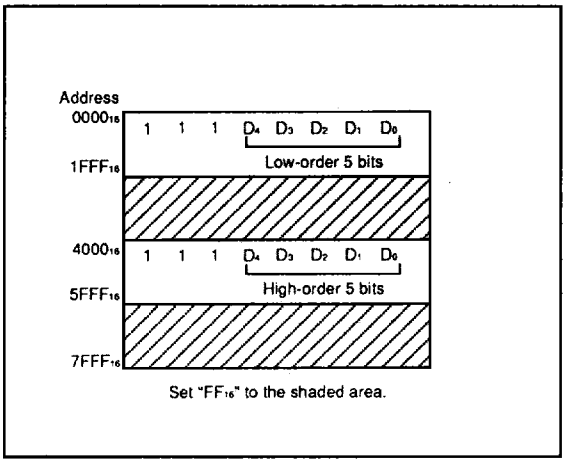


Fig. 54 PROM memory map

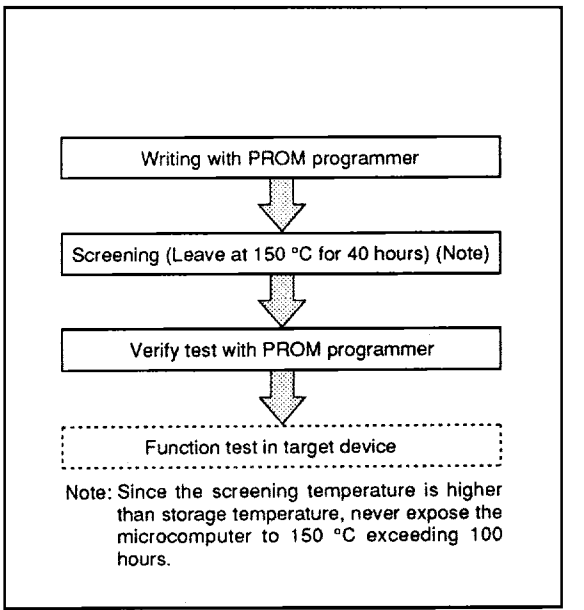


Fig. 55 Flow of writing and test of the product shipped in blank