4512 Group

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

DESCRIPTION

The 4512 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), and 10-bit A-D converter.

The various microcomputers in the 4512 Group include variations of the built-in memory size as shown in the table below.

FEATURES

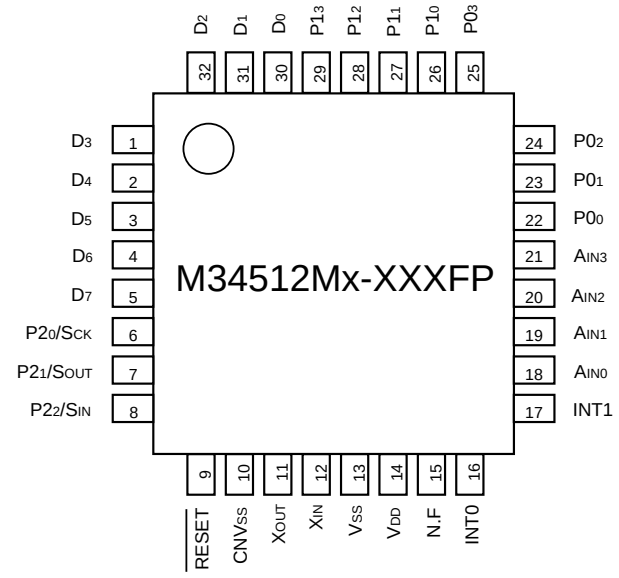
- Minimum instruction execution time ..... 0.75 μs (at 4.0 MHz oscillation frequency, in high-speed mode, V_{DD} = 4.0 V to 5.5 V)
- Supply voltage ..... 4.0 V to 5.5 V (at 4.2 MHz oscillation frequency)
- Timers
  - Timer 1 ..... 8-bit timer with a reload register
  - Timer 2 ..... 8-bit timer with 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 ..... 10-bit successive comparison method
- Watchdog timer ..... 16 bits
- Clock generating circuit (ceramic resonator)
- LED drive directly enabled (port D)

APPLICATION

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

Product	ROM (PROM) size (× 10 bits)	RAM size (× 4 bits)	Package	ROM type
M34512M2-XXXFP	2048 words	128 words	32P6B-A	Mask ROM
M34512M4-XXXFP	4096 words	256 words	32P6B-A	Mask ROM

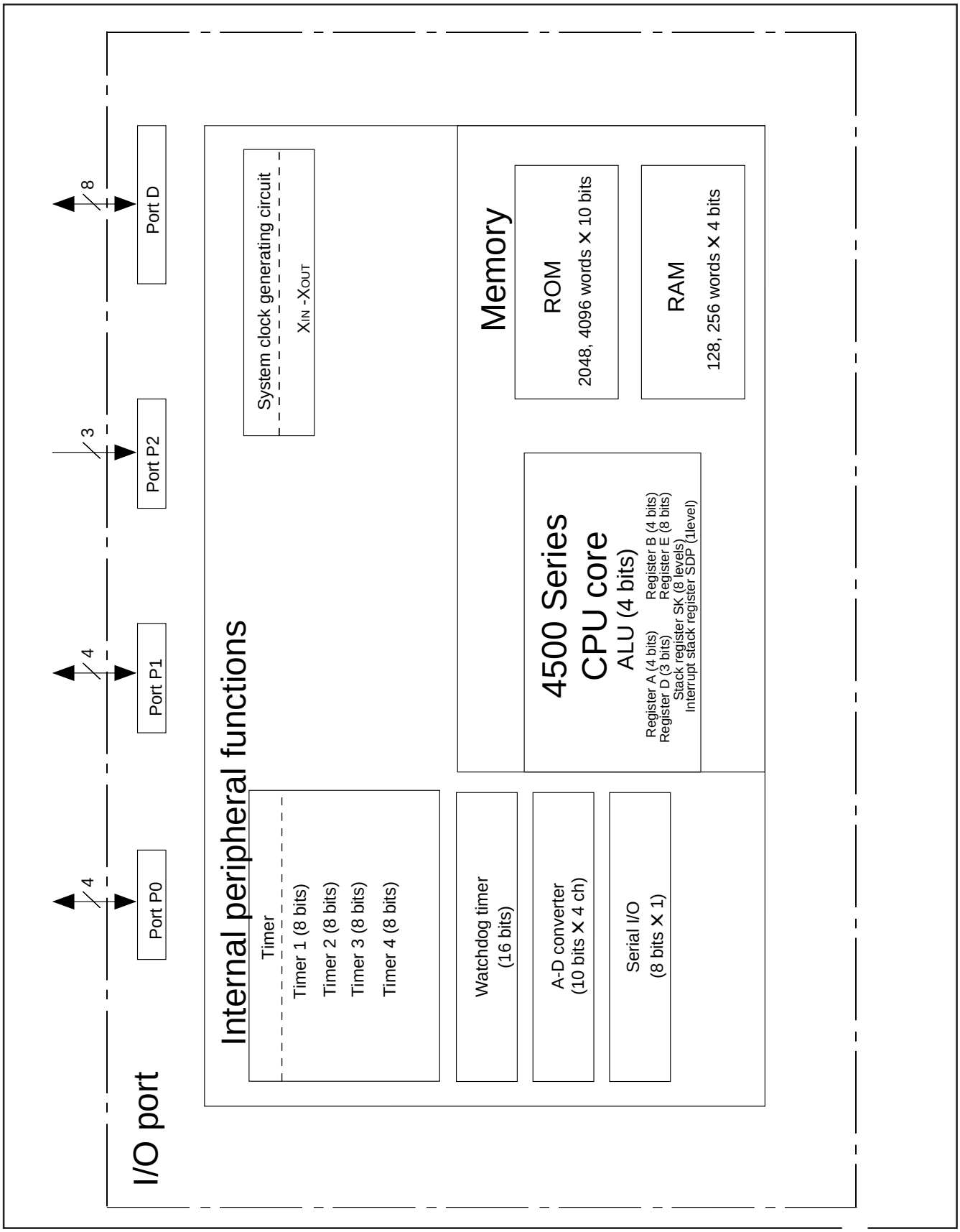
PIN CONFIGURATION (TOP VIEW)



Outline 32P6B-A

N.F: No Function
However, connect to V_{SS} as an unused pin.

BLOCK DIAGRAM



PERFORMANCE OVERVIEW

Parameter			Function
Number of basic instructions			117
Minimum instruction execution time			0.75 μs (at 4.0 MHz oscillation frequency, in high-speed mode)
Memory sizes	ROM	M34512M2	2048 words × 10 bits
		M34512M4	4096 words × 10 bits
	RAM	M34512M2	128 words × 4 bits
		M34512M4	256 words × 4 bits
Input/Output ports	D_0–D_7	I/O	Eight independent I/O ports. Input is examined by skip decision.
	$P0_0$–$P0_3$	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.
	$P2_0$–$P2_2$	Input	3-bit input port; ports $P2_0$, $P2_1$ and $P2_2$ are also used as SCK, SOUT and SIN, respectively.
	INT0	Input	1-bit input; INT0 pin is equipped with a key-on wakeup function.
	INT1	Input	1-bit input; INT1 pin is equipped with a key-on wakeup function.
Timers	Timer 1		8-bit programmable timer with a reload register.
	Timer 2		8-bit programmable timer with a reload register.
	Timer 3		8-bit programmable timer with a reload register.
	Timer 4		8-bit programmable timer with a reload register.
A-D converter			10-bit wide, This is equipped with an 8-bit comparator function.
Serial I/O			8-bit × 1
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			32-pin plastic molded LQFP(32P6B-A)
Operating temperature range			–20 °C to 85 °C
Supply voltage			4.0 V to 5.5 V
Power dissipation (typical value)	Active mode		3.0 mA (at V_{DD} = 5.0 V, 4.0 MHz oscillation frequency, in high-speed mode, output transistors in the cut-off state)
	RAM back-up mode		0.1 μA (at room temperature, V_{DD} = 5 V, output transistors in the cut-off state)

PIN DESCRIPTION

Pin	Name	Input/Output	Function
V_{DD}	Power supply	—	Connected to a plus power supply.
V_{SS}	Ground	—	Connected to a 0 V power supply.
N.F	No Function	—	This pin has no function, and connect to V_{SS} as an unused pin.
CNV_{SS}	CNV_{SS}	—	Connect CNV_{SS} to V_{SS} and apply "L" (0V) to CNV_{SS} certainly.
$\overline{RESET}$	Reset input	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 ceramic resonator. A feedback resistor is built-in between them.
X_{OUT}	System clock output	Output	
D_0–D_7	I/O port D	I/O	Each pin of port D has an independent 1-bit wide I/O function. Each pin has an output latch. For input use, set the latch of the specified bit to "1." Input is examined by skip decision. The output structure is N-channel open-drain.
$P0_0$–$P0_3$	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		
$P2_0$–$P2_2$	Input port P2	Input	3-bit input port. Ports $P2_0$, $P2_1$ and $P2_2$ are also used as S_{CK}, S_{OUT} and S_{IN}, respectively.
A_{IN0}–A_{IN3}	Analog input	Input	Analog input pins for A-D converter.
INT0, INT1	Interrupt input	Input	INT0, INT1 pins accept external interrupts. They also accept the input signal to return the system from the RAM back-up state.
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 $P2_2$.
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 $P2_1$.
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 $P2_0$.

MULTIFUNCTION

Pin	Multifunction	Pin	Multifunction
P20	SCK	SCK	P20
P21	SOUT	SOUT	P21
P22	SIN	SIN	P22

Notes 1: Pins except above have just single function.
2: The input of P20–P22 can be used even when SCK, SOUT, SIN are selected.

DEFINITION OF CLOCK AND CYCLE

● System clock
The system clock is the basic clock for controlling this product. The system clock is selected by the bit 3 of the clock control register MR.

Table Selection of system clock

Register MR MR3	System clock
0	f(XIN)
1	f(XIN)/2

Note: f(XIN)/2 is selected after system is released from reset.

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

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

CONNECTIONS OF UNUSED PINS

Pin	Connection
XOUT	Open (when using an external clock).
N.F	Connect to VSS.
D0–D7	Connect to VSS, or set the output latch to "0" and open.
P20/SCK P21/SOUT P22/SIN	Connect to VSS.
INT0 INT1	Connect to VSS.
AIN0–AIN3	Connect to VSS.
P00–P03	Open or connect to VSS (Note)
P10–P13	Open or connect to VSS (Note)

Note: When the P00–P03 and P10–P13 are connected to VSS, turn off their pull-up transistors (register PU0i="0") and also invalidate the key-on wakeup functions (register K0i="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 PU0i="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.)

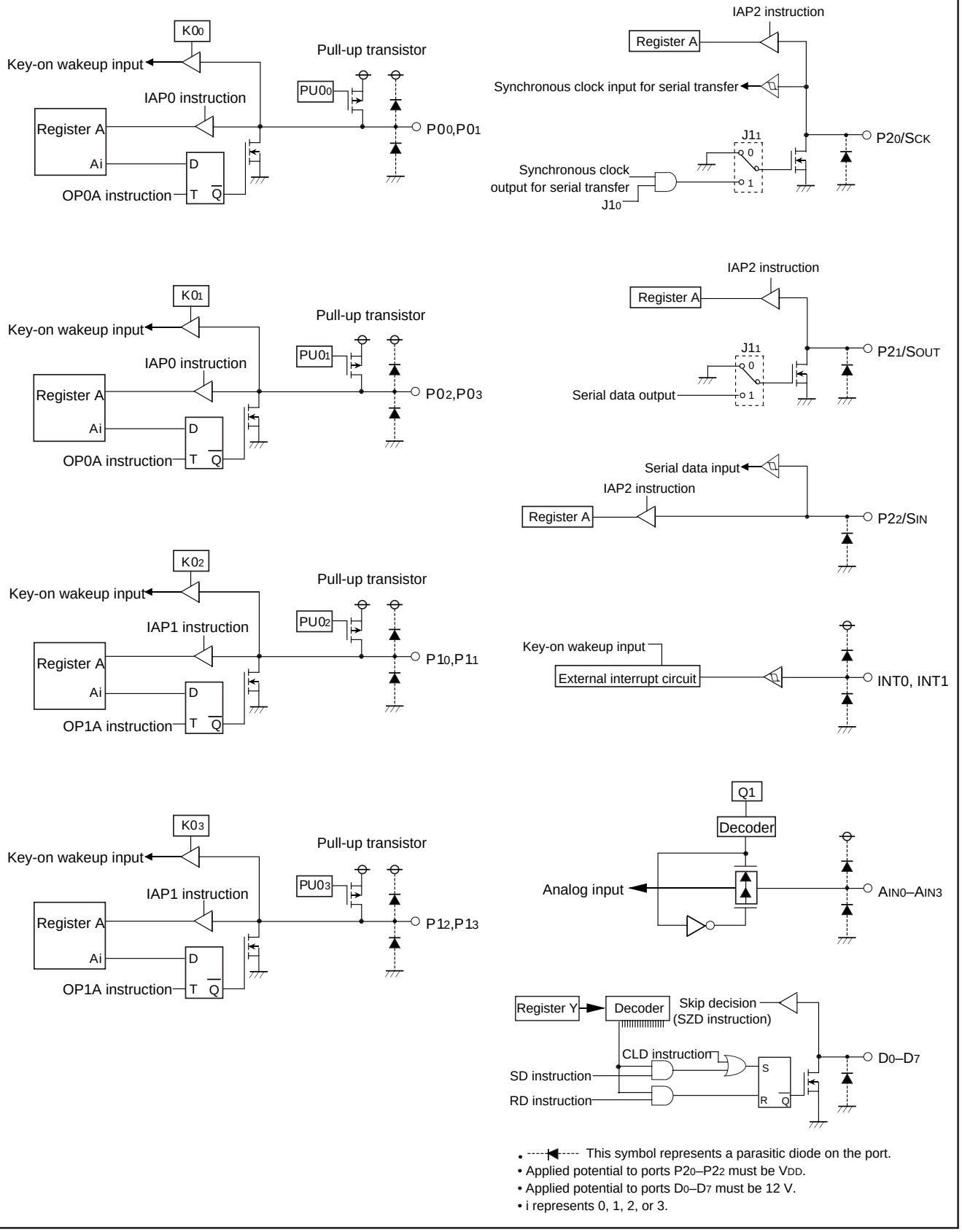
(Note when the output latch is set to "0" and pins are open)
● After system is released from reset, port is in a high-impedance state until it is set the output latch to "0" by software. Accordingly, the voltage level of pins is undefined and the excess of the supply current may occur while the port 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 VDD)
● Connect the unused pins to VSS and VDD using the thickest wire at the shortest distance against noise.

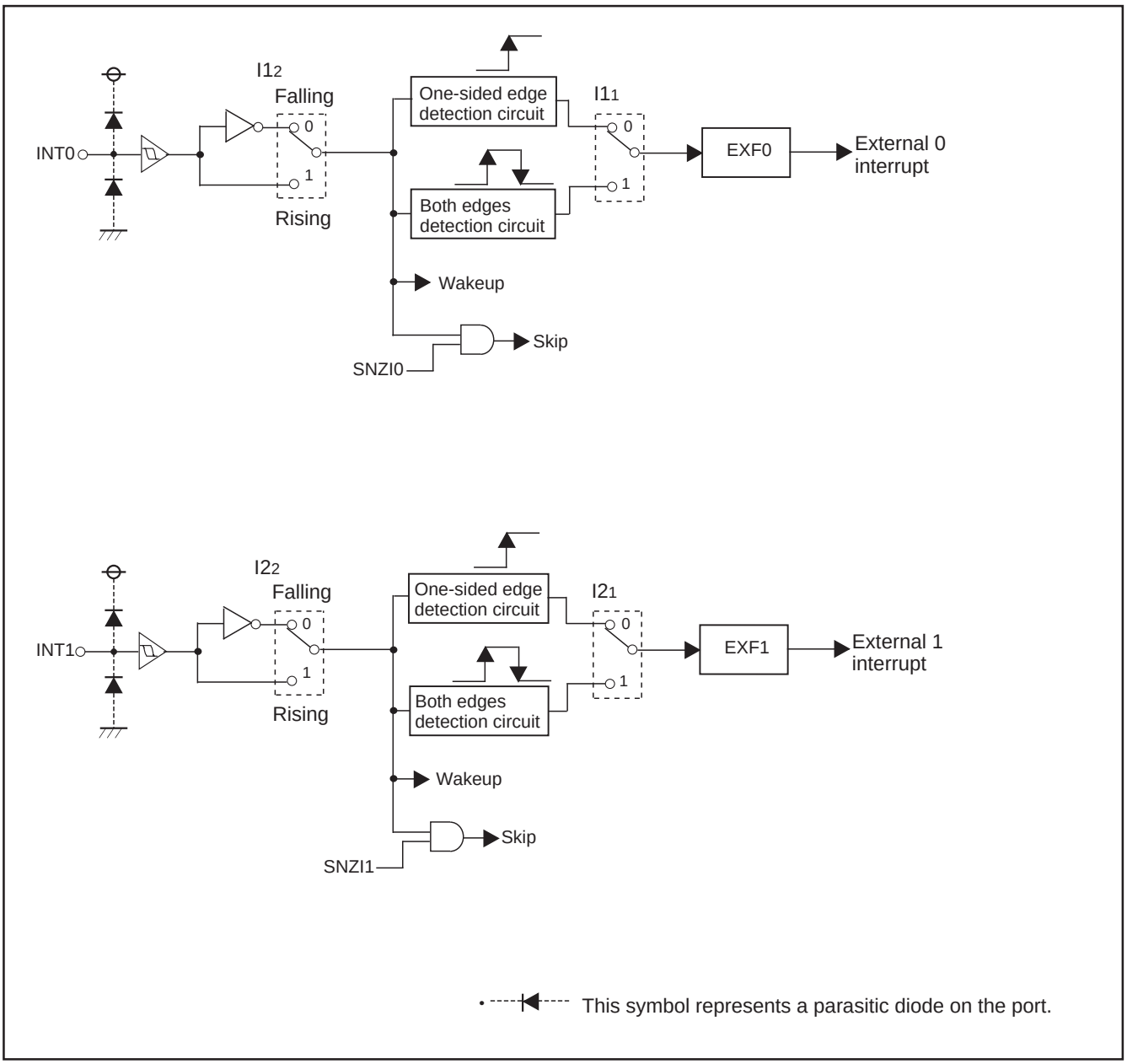
PORT FUNCTION

Port	Pin	Input Output	Output structure	I/O unit	Control instructions	Control registers	Remark
Port D	D0–D7	I/O (8)	N-channel open-drain	1	SD, RD SZD CLD		
Port P0	P00–P03	I/O (4)	N-channel open-drain	4	OP0A IAP0	PU0, K0	Built-in programmable pull-up functions Key-on wakeup functions (programmable)
Port P1	P10–P13	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	P20/SCK P21/SOUT P22/SIN	Input (3)		3	IAP2	J1	

PORT BLOCK DIAGRAMS



- ----|◄---- This symbol represents a parasitic diode on the port.
- Applied potential to ports P20–P22 must be VDD.
- Applied potential to ports D0–D7 must be 12 V.
- i represents 0, 1, 2, or 3.



External interrupt circuit structure

FUNCTION BLOCK OPERATIONS
CPU

(1) Arithmetic logic unit (ALU)

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

(2) Register A and carry flag

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

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

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

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

(3) Registers B and E

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

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

(4) Register D

Register D is a 3-bit register.

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

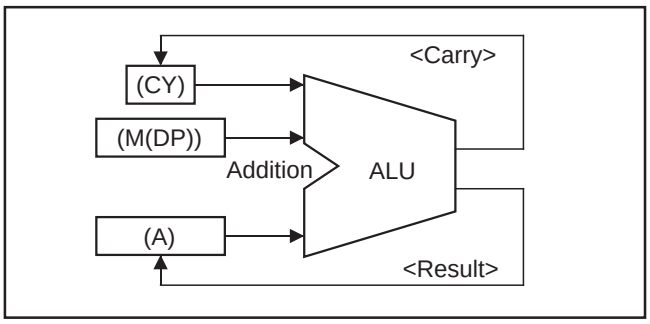


Fig. 1 AMC instruction execution example

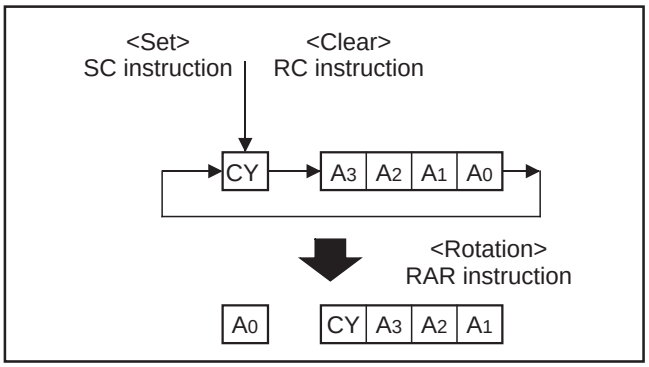


Fig. 2 RAR instruction execution example

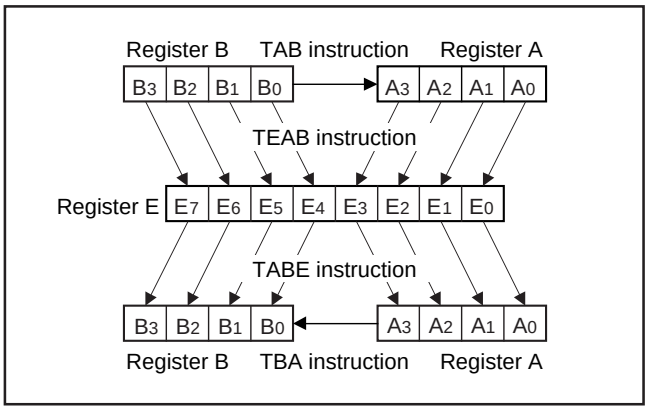


Fig. 3 Registers A, B and register E

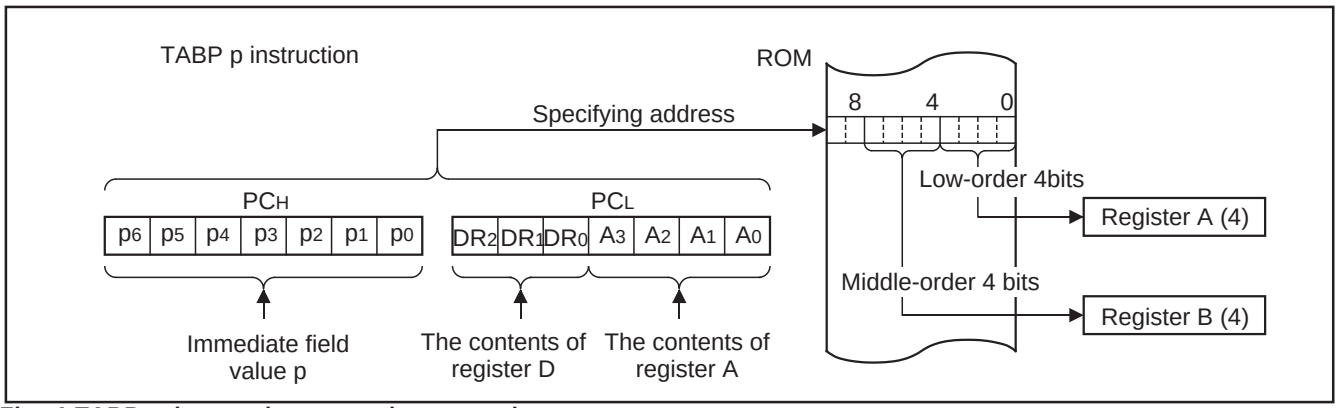


Fig. 4 TABP p instruction execution example

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

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

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

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

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

Figure 5 shows the stack registers (SKs) structure.

Figure 6 shows the example of operation at subroutine call.

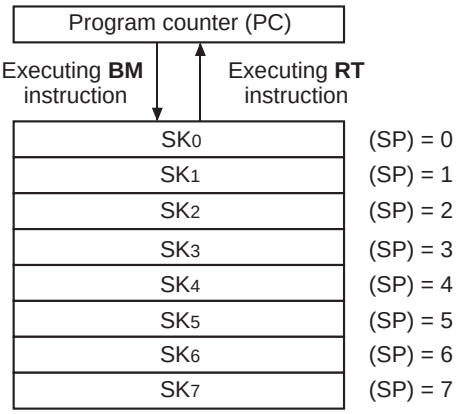
(6) Interrupt stack register (SDP)

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

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

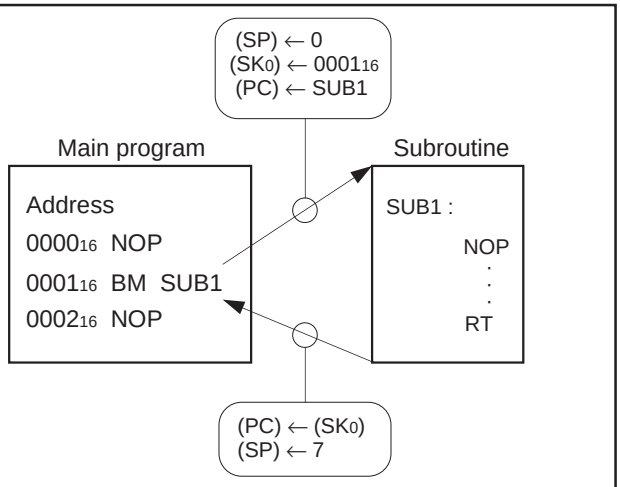
(7) Skip flag

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



Stack pointer (SP) points "7" at reset or returning from RAM back-up mode. It points "0" by executing the first **BM** instruction, and the contents of program counter is stored in SK_0. When the **BM** instruction is executed after eight stack registers are used ((SP) = 7), (SP) = 0 and the contents of SK_0 is destroyed.

Fig. 5 Stack registers (SKs) structure



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

Fig. 6 Example of operation at subroutine call

(8) Program counter (PC)

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

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

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

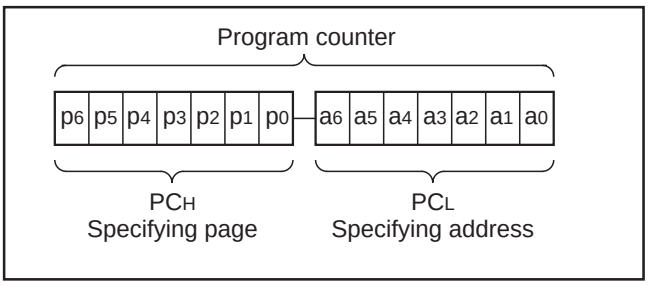


Fig. 7 Program counter (PC) structure

(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).

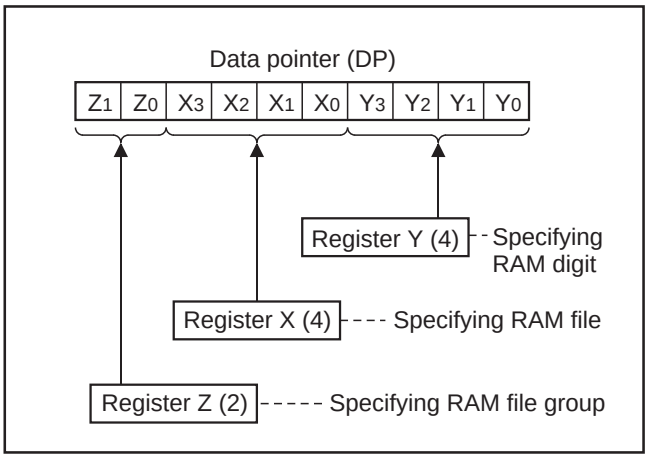


Fig. 8 Data pointer (DP) structure

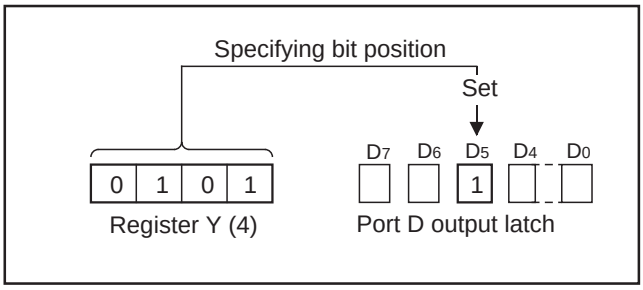


Fig. 9 SD instruction execution example

PROGRAM MEMOY (ROM)

The program memory is a mask ROM. 1 word of ROM is composed of 10 bits. ROM is separated every 128 words by the unit of page (addresses 0 to 127). Table 1 shows the ROM size and pages. Figure 10 shows the ROM map of M34512M4.

Table 1 ROM size and pages

Product	ROM size (× 10 bits)	Pages
M34512M2	2048 words	16 (0 to 15)
M34512M4	4096 words	32 (0 to 31)

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

Page 2 (addresses 0100_{16} to $017F_{16}$) is the special page for subroutine calls. Subroutines written in this page can be called from any page with the 1-word instruction (BM). Subroutines extending from page 2 to another page can also be called with the BM instruction when it starts on page 2.

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

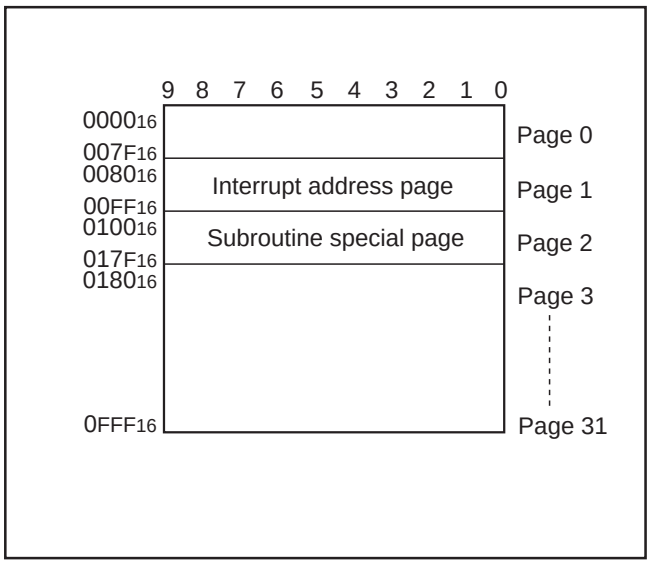


Fig. 10 ROM map of M34512M4

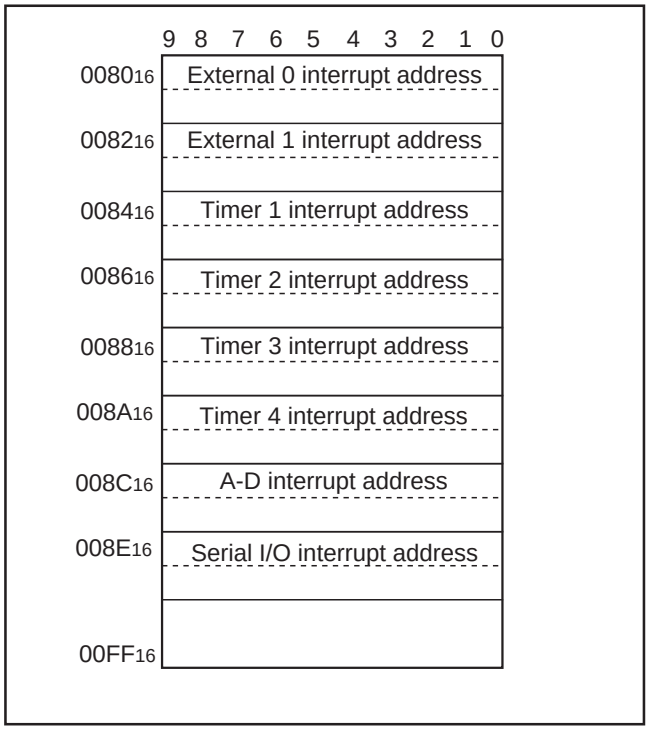


Fig. 11 Page 1 (addresses 0080_{16} to $00FF_{16}$) structure

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
M34512M2	128 words × 4 bits (512 bits)
M34512M4	256 words × 4 bits (1024 bits)

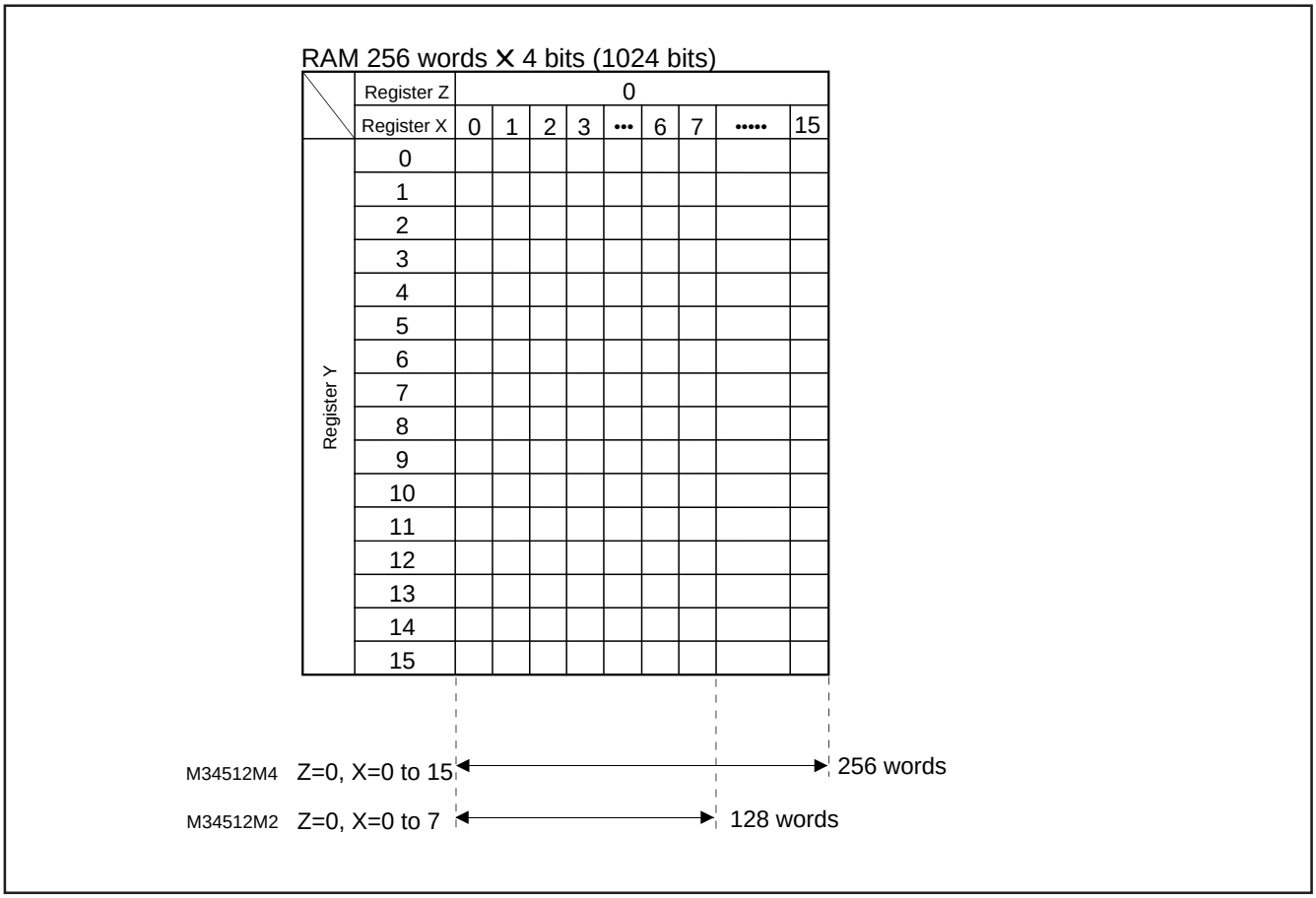


Fig. 12 RAM map

INTERRUPT FUNCTION

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

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

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

(1) Interrupt enable flag (INTE)

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

(2) Interrupt enable bit

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

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

Table 5 shows the interrupt enable bit function.

(3) Interrupt request flag

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

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

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

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

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

Table 3 Interrupt sources

Priority level	Interrupt name	Activated condition	Interrupt address
1	External 0 interrupt	Level change of 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)

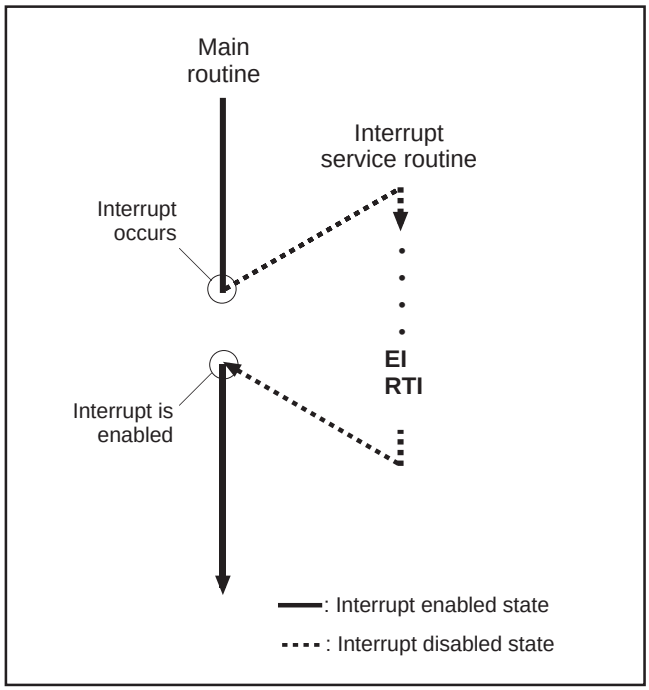


Fig. 13 Program example of interrupt processing

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

Fig. 14 Internal state when interrupt occurs

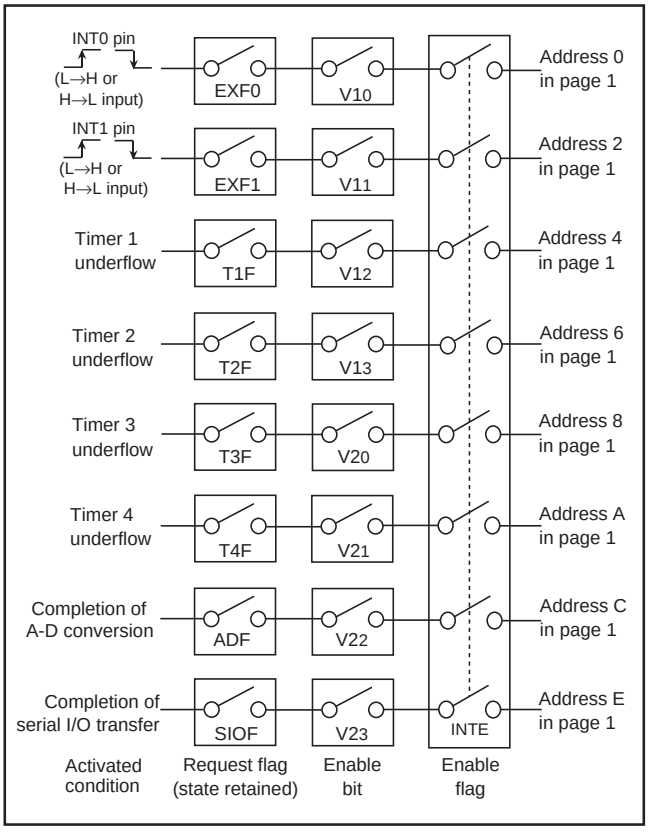


Fig. 15 Interrupt system diagram

(6) Interrupt control registers

- Interrupt control register V1
  Interrupt enable bits of external 0, 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.

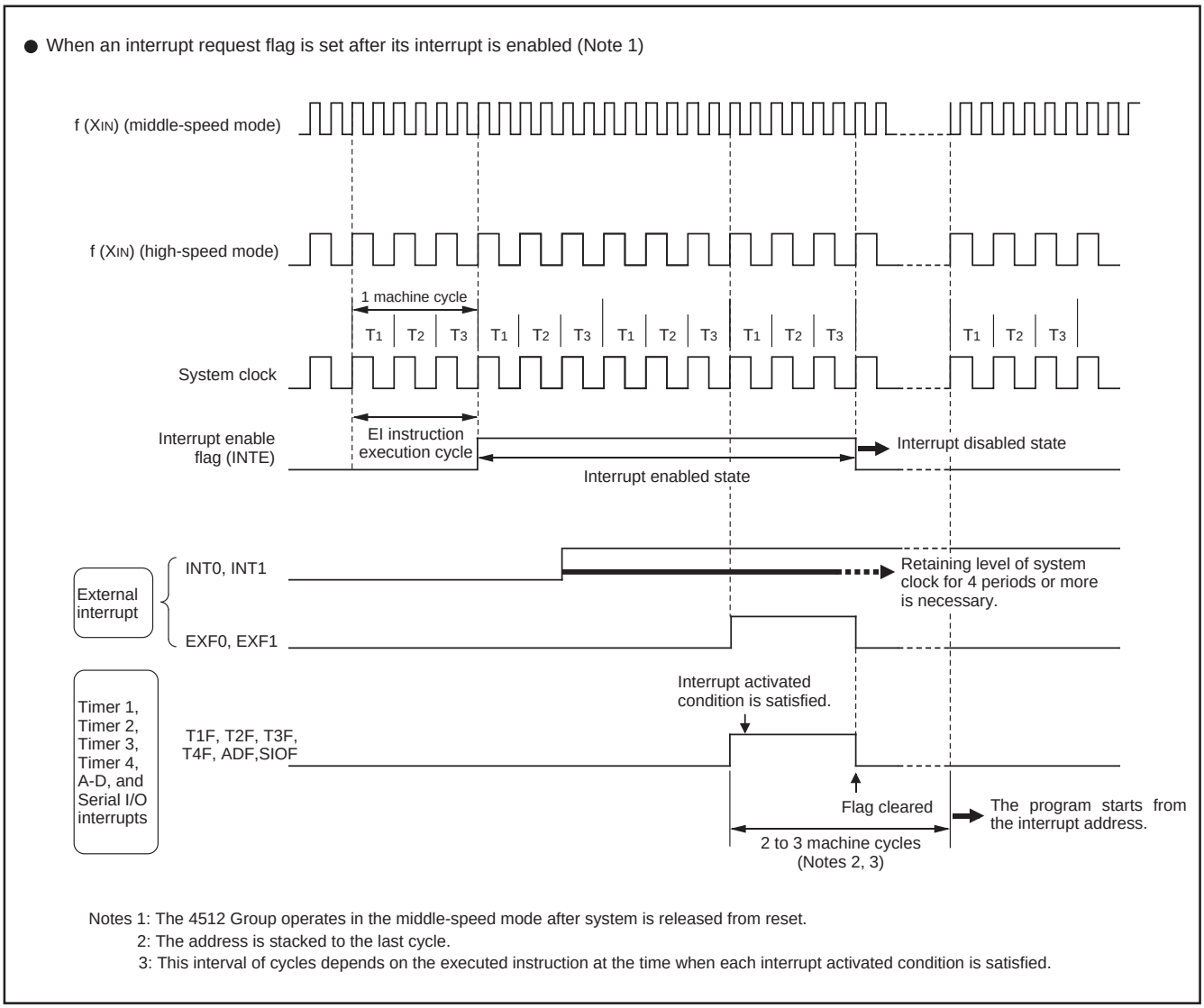
Table 6 Interrupt control registers

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 (V10–V13 and V20–V23), and interrupt request flag are "1." The interrupt actually occurs 2 to 3 machine cycles after the cycle in which all three conditions are satisfied. The interrupt occurs after 3 machine cycles only when the three interrupt conditions are satisfied on execution of other than one-cycle instructions (Refer to Figure 16).



Notes 1: The 4512 Group operates in the middle-speed mode after system is released from reset.
2: The address is stacked to the last cycle.
3: 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 4512 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).
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	INT0	When the next waveform is input to INT0 pin • Falling waveform (“H”→“L”) • Rising waveform (“L”→“H”) • Both rising and falling waveforms	$I1_1$ $I1_2$
External 1 interrupt	INT1	When the next waveform is input to INT1 pin • Falling waveform (“H”→“L”) • Rising waveform (“L”→“H”) • Both rising and falling waveforms	$I2_1$ $I2_2$

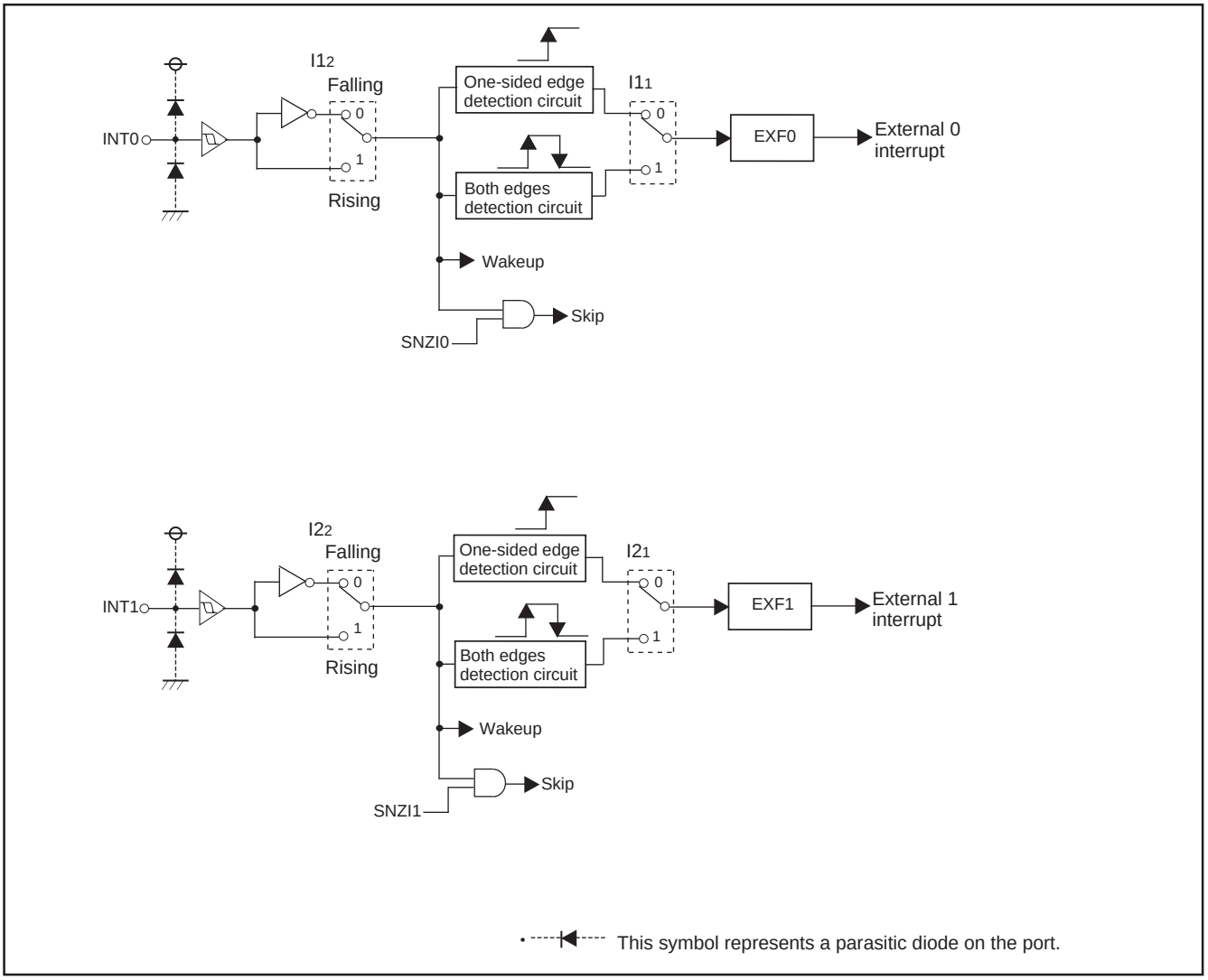


Fig. 17 External interrupt circuit structure

(1) External 0 interrupt request flag (EXF0)

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

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

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

• External 0 interrupt activated condition

External 0 interrupt activated condition is satisfied when a valid waveform is input to INT0 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 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 INT0 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 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.

• External 1 interrupt activated condition

External 1 interrupt activated condition is satisfied when a valid waveform is input to INT1 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 1 interrupt is as follows.

① Select the valid waveform with the bits 1 and 2 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 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 valid waveform for the external 0 interrupt. Set the contents of this register through register A with the TI1A instruction. The TAI1 instruction can be used to transfer the contents of register I1 to register A.

• Interrupt control register I2

Register I2 controls the valid waveform for the external 1 interrupt. 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
$I1_3$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$I1_2$	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	
$I1_1$	INT0 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I1_0$	INT0 pin timer 1 control enable bit	0	Disabled	
		1	Enabled	
Interrupt control register I2		at reset : 0000_2	at RAM back-up : state retained	R/W
$I2_3$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$I2_2$	Interrupt valid waveform for INT1 pin/ return level selection bit (Note 3)	0	Falling waveform (“L” level of INT1 pin is recognized with the SNZI1 instruction)/“L” level	
		1	Rising waveform (“H” level of INT1 pin is recognized with the SNZI1 instruction)/“H” level	
$I2_1$	INT1 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I2_0$	INT1 pin timer 3 control enable bit	0	Disabled	
		1	Enabled	

Notes 1: “R” represents read enabled, and “W” represents write enabled.
2: When the contents of $I1_2$ is changed, the external interrupt request flag EXF0 may be set. Accordingly, clear EXF0 flag with the SNZ0 instruction.
3: When the contents of $I2_2$ is changed, the external interrupt request flag EXF1 may be set. Accordingly, clear EXF1 flag with the SNZ1 instruction.

TIMERS

The 4512 Group has the following timers.

- Programmable timer
  The programmable timer has a reload register and enables the frequency dividing ratio to be set. It is decremented from a setting value n. When it underflows (count to n + 1), a timer interrupt request flag is set to “1,” new data is loaded from the reload register, and count continues (auto-reload function).
- Fixed dividing frequency timer
  The fixed dividing frequency timer has the fixed frequency dividing ratio (n). An interrupt request flag is set to “1” after every n count of a count pulse.

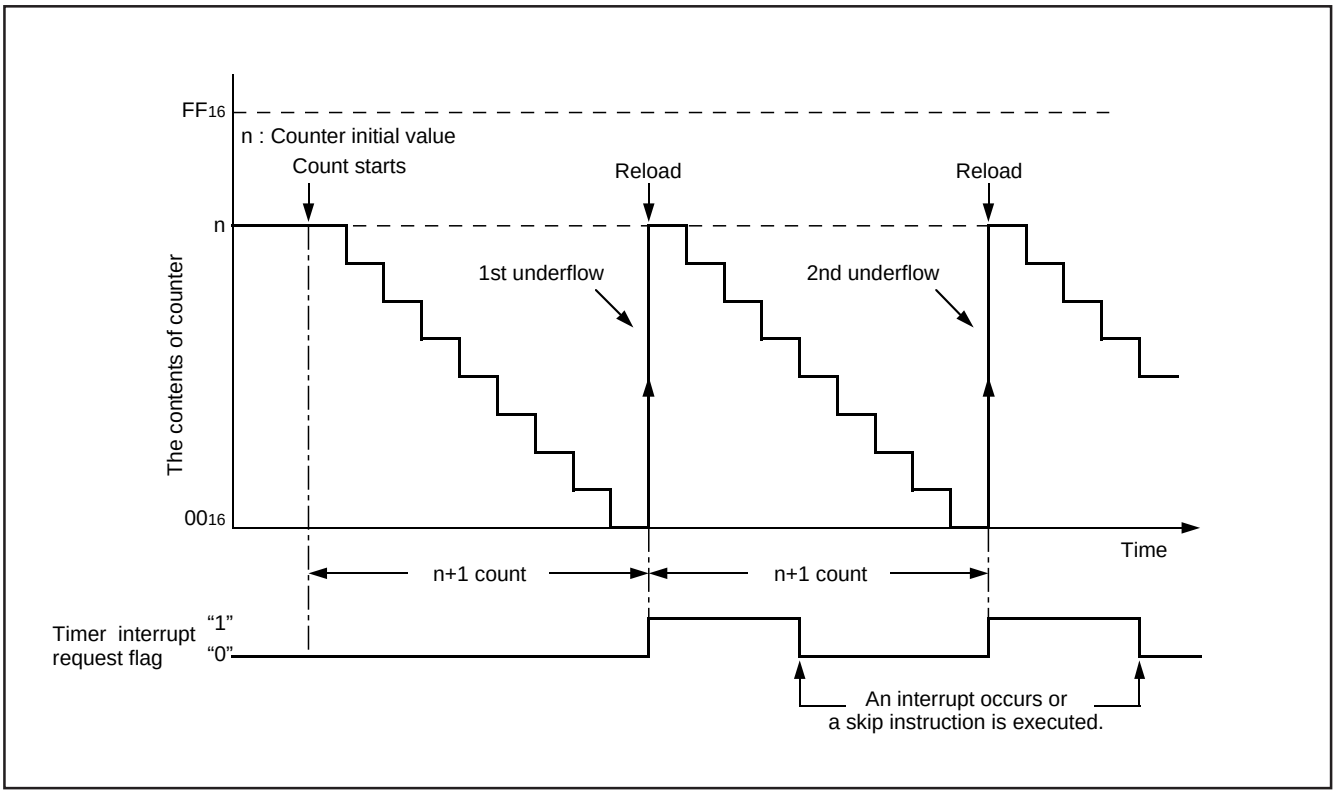


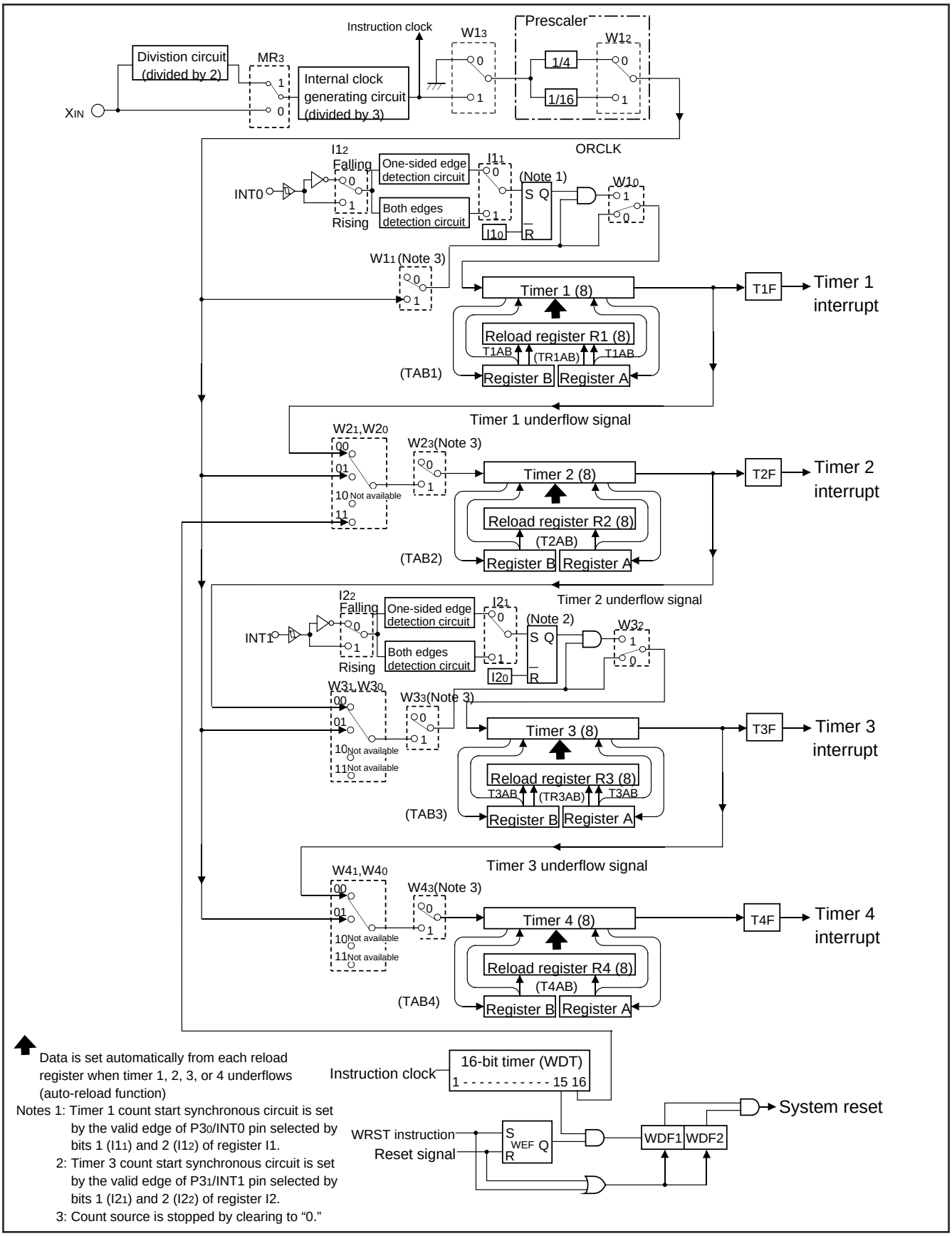
Fig. 18 Auto-reload function

The 4512 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)
- 16-bit timer

Prescaler and timers 1 to 4 can be controlled with the timer control registers W1 to W4. The 16-bit timer is a free counter which is not controlled with the control register.
Each function is described below.

Table 9 Function related timers

Circuit	Structure	Count source	Frequency dividing ratio	Use of output signal	Control register
Prescaler	Frequency divider	• Instruction clock	4, 16	• Timer 1, 2, 3 and 4 count sources	W1
Timer 1	8-bit programmable binary down counter (link to INT0 input)	• Prescaler output (ORCLK)	1 to 256	• Timer 2 count source • Timer 1 interrupt	W1
Timer 2	8-bit programmable binary down counter	• Timer 1 underflow • Prescaler output (ORCLK) • 16-bit timer underflow	1 to 256	• Timer 3 count source • Timer 2 interrupt	W2
Timer 3	8-bit programmable binary down counter (link to INT1 input)	• 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)	1 to 256	• Timer 4 interrupt	W4
16-bit timer	16-bit fixed dividing frequency	• Instruction clock	65536	• Watchdog timer (The 15th bit is counted twice) • Timer 2 count source (16-bit timer underflow)	



Data is set automatically from each reload register when timer 1, 2, 3, or 4 underflows (auto-reload function)

Notes 1: Timer 1 count start synchronous circuit is set by the valid edge of P30/INT0 pin selected by bits 1 ($I1_1$) and 2 ($I1_2$) of register I1.

2: Timer 3 count start synchronous circuit is set by the valid edge of P31/INT1 pin selected by bits 1 ($I2_1$) and 2 ($I2_2$) of register I2.

3: Count source is stopped by clearing to "0."

Fig. 19 Timers structure

Table 10 Timer control registers

Timer control register W1		at reset : 0000_2		at RAM back-up : 0000_2	R/W
$W1_3$	Prescaler control bit	0	Stop (state initialized)		
		1	Operating		
$W1_2$	Prescaler dividing ratio selection bit	0	Instruction clock divided by 4		
		1	Instruction clock divided by 16		
$W1_1$	Timer 1 control bit	0	Stop (state retained)		
		1	Operating		
$W1_0$	Timer 1 count start synchronous circuit control bit	0	Count start synchronous circuit not selected		
		1	Count start synchronous circuit selected		

Timer control register W2		at reset : 0000_2		at RAM back-up : state retained	R/W
$W2_3$	Timer 2 control bit	0	Stop (state retained)		
		1	Operating		
$W2_2$	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	Not available	
		1	1	16 bit timer (WDT) underflow signal	

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	Not available	
		1	1	Not available	

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, the selection of count start synchronous circuit, and the frequency dividing ratio and count operation of prescaler. Set the contents of this register through register A with the TW1A instruction. The TAW1 instruction can be used to transfer the contents of register W1 to register A.
- Timer control register W2
  Register W2 controls the 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.

(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.
- Writing to reload registers R1 and R3
  When writing data to reload registers R1 or R3 while timer 1 or timer 3 is operating, avoid a timing when timer 1 or timer 3 underflows.

(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. Data can be written to reload register (R1) with the TR1AB instruction.
When writing data to reload register R1 with the TR1AB instruction, the downcount after the underflow is started from the setting value of reload register R1.
Timer 1 starts counting after the following process;
① set data in timer 1, and
② set the bit 1 of register W1 to “1.”
However, INT0 pin input can be used as the start trigger for timer 1 count operation by setting the bit 0 of register W1 to “1.”
When a value set is n, timer 1 divides the count source signal by n + 1 (n = 0 to 255).
Once count is started, when timer 1 underflows (the next count pulse is input after the contents of timer 1 becomes “0”), the timer 1 interrupt request flag (T1F) is set to “1,” new data is loaded from reload register R1, and count continues (auto-reload function).
Data can be read from timer 1 with the TAB1 instruction. When reading the data, stop the counter and then execute the TAB1 instruction.

(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.”
When a value set is n, timer 2 divides the count source signal by n + 1 (n = 0 to 255).
Once count is started, when timer 2 underflows (the next count pulse is input after the contents of timer 2 becomes “0”), the timer 2 interrupt request flag (T2F) is set to “1,” new data is loaded from reload register R2, and count continues (auto-reload function).
Data can be read from timer 2 with the TAB2 instruction. When reading the data, stop the counter and then execute the TAB2 instruction.

(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. Data can be written to reload register (R3) with the TR3AB instruction.

When writing data to reload register R3 with the TR3AB instruction, the downcount after the underflow is started from the setting value of reload register R3.

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, INT1 pin input can be used as the start trigger for timer 3 count operation by setting the bit 2 of register W3 to "1."

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

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).

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.

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."

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

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).

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

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

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

(9) Count start synchronization circuit (timer 1, timer 3)

Each of timer 1 and timer 3 has the count start synchronous circuit which synchronizes $P3_0$/INT0 pin and $P3_1$/INT1 pin, respectively, and can start the timer count operation.

Timer 1 count start synchronous circuit function is selected by setting the bit 0 of register W1 to "1." The control by $P3_0$/INT0 pin input can be performed by setting the bit 0 of register I1 to "1."

The count start synchronous circuit is set by level change ("H"→"L" or "L"→"H") of $P3_0$/INT0 pin input. This valid waveform is selected by bits 1 ($I1_1$) and 2 ($I1_2$) of register I1 as follows;

• $I1_1$ = "0": Synchronized with one-sided edge (falling or rising)

• $I1_1$ = "1": Synchronized with both edges (both falling and rising)

When register $I1_1$="0" (synchronized with the one-sided edge), the rising or falling waveform can be selected by bit 2 of register I1;

• $I1_2$ = "0": Falling waveform

• $I1_2$ = "1": Rising waveform

Timer 3 count start synchronous circuit function is selected by setting the bit 2 of register W3 to "1." The control by $P3_1$/INT1 pin input can be performed by setting the bit 0 of register I2 to "1."

The count start synchronous circuit is set by level change ("H"→"L" or "L"→"H") of $P3_1$/INT1 pin input. This valid waveform is selected by bits 1 ($I2_1$) and 2 ($I2_2$) of register I2 as follows;

• $I2_1$ = "0": Synchronized with one-sided edge (falling or rising)

• $I2_1$ = "1": Synchronized with both edges (both falling and rising)

When register $I2_1$="0" (synchronized with the one-sided edge), the rising or falling waveform can be selected by bit 2 of register I2;

• $I2_2$ = "0": Falling waveform

• $I2_2$ = "1": Rising waveform

When timer 1 and timer 3 count start synchronous circuits are used, the count start synchronous circuits are set, the count source is input to each timer by inputting valid waveform to $P3_0$/INT0 pin and $P3_1$/INT1 pin. Once set, the count start synchronous circuit is cleared by clearing the bit $I1_0$ or $I2_0$ to "0" or reset.

WATCHDOG TIMER

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

The timer WDT downcounts the instruction clocks as the count source after system is released from reset. The underflow signal is generated when the count value reaches "0000_{16}." This underflow signal can be used as the timer 2 count source.

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

When the count value of timer WDT reaches "$BFFF_{16}$" or "$3FFF_{16}$," the WDF1 flag is set to "1." If the WRST instruction is never executed while timer WDT counts 32767, WDF2 flag is set to "1," and the $\overline{\text{RESET}}$ pin outputs "L" level to reset the microcomputer. Execute the WRST instruction at each period of 32766 machine cycle or less by software when using watchdog timer to keep the microcomputer operating normally.

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

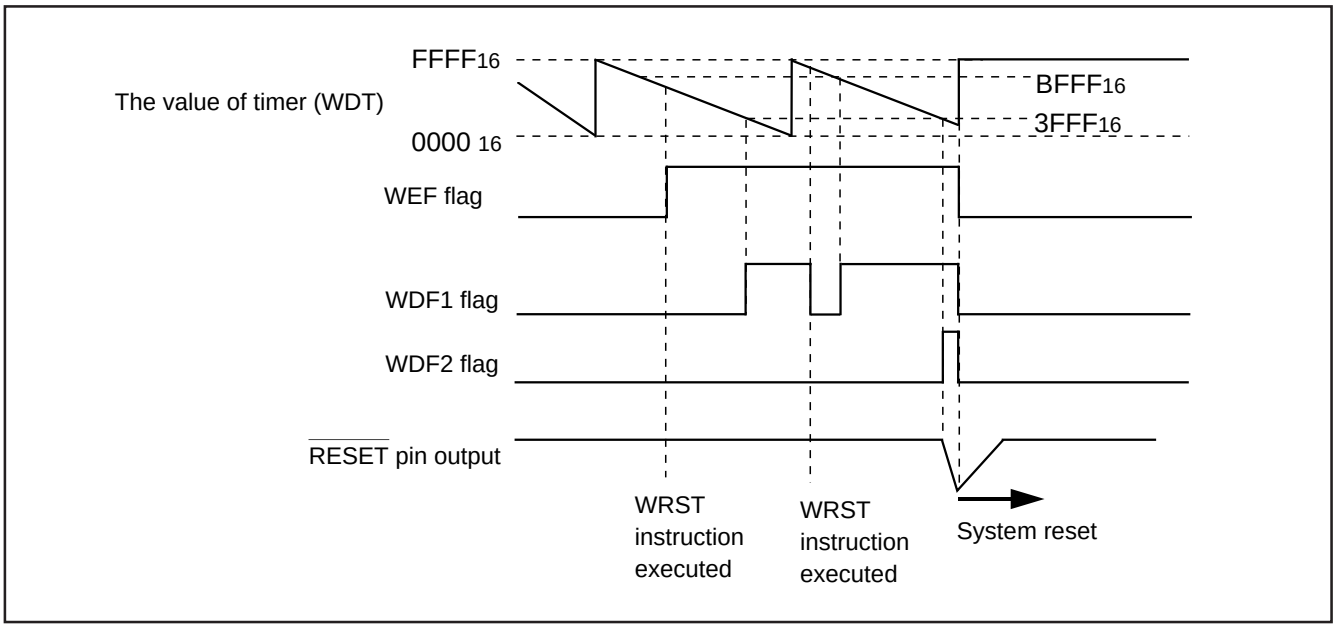


Fig. 20 Watchdog timer function

The contents of WEF, WDF1 and WDF2 flags and timer WDT are initialized at the RAM back-up mode.

If WDF2 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 WDF1 flag with the WRST instruction just before the microcomputer enters the RAM back-up state (refer to Figure 21)

```
      .
      .
      .
WRST        ; WDF1 flag reset
EPOF        ; POF instruction enabled
POF
 ↓
Oscillation stop   (RAM back-up state)
```

Fig. 21 Program example to enter the RAM back-up mode when using the watchdog timer

SERIAL I/O

The 4512 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

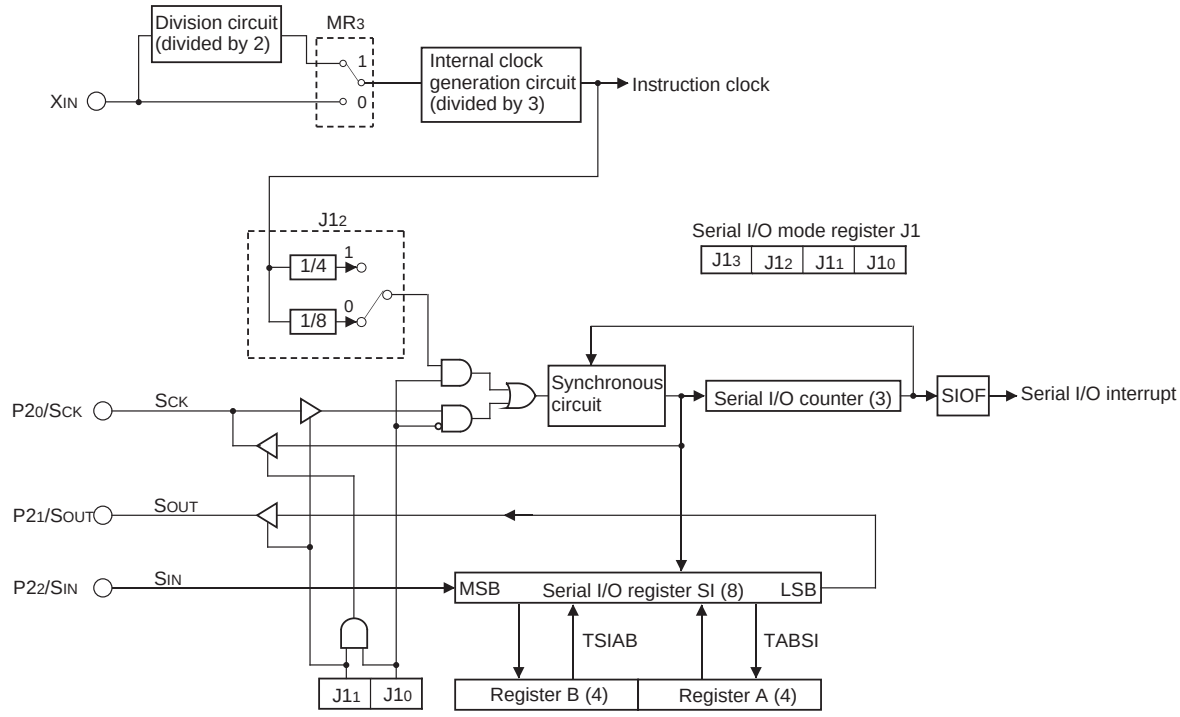
Registers A and B are used to perform data transfer with internal CPU, and the serial I/O pins are used for external data transfer.

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
$P2_0$/SCK	Clock I/O (SCK)
$P2_1$/SOUT	Serial data output (SOUT)
$P2_2$/SIN	Serial data input (SIN)

Note: Input ports $P2_0$–$P2_2$ can be used regardless of register J1.



Note: The output structure of SCK and SOUT pins is N-channel open-drain.

Fig. 22 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	This bit has no function, but read/write is enabled.	
		1		
$J1_2$	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	Input ports $P2_0$, $P2_1$, $P2_2$ selected	
		1	Serial I/O ports SCK, SOUT, SIN/input ports $P2_0$, $P2_1$, $P2_2$ 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.

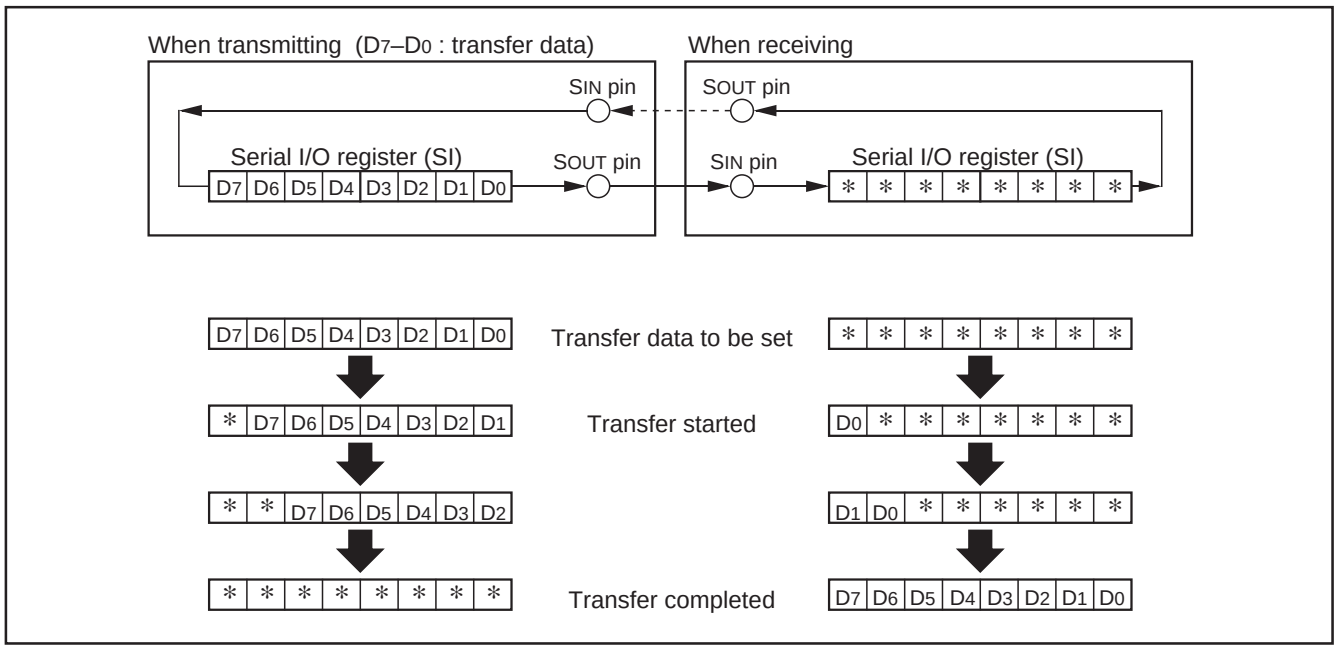


Fig. 23 Serial I/O register state when transferring

(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 a work register without using serial I/O, pull up the SCK pin or set the pin function to an input port P20.

(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, P20/SCK, P21/SOUT and P22/SIN pin function. Set the contents of this register through register A with the TJ1A instruction. The TAJ1 instruction can be used to transfer the contents of register J1 to register A.

(5) How to use serial I/O

Figure 24 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 25 shows the data transfer timing and Table 13 shows the data transfer sequence.

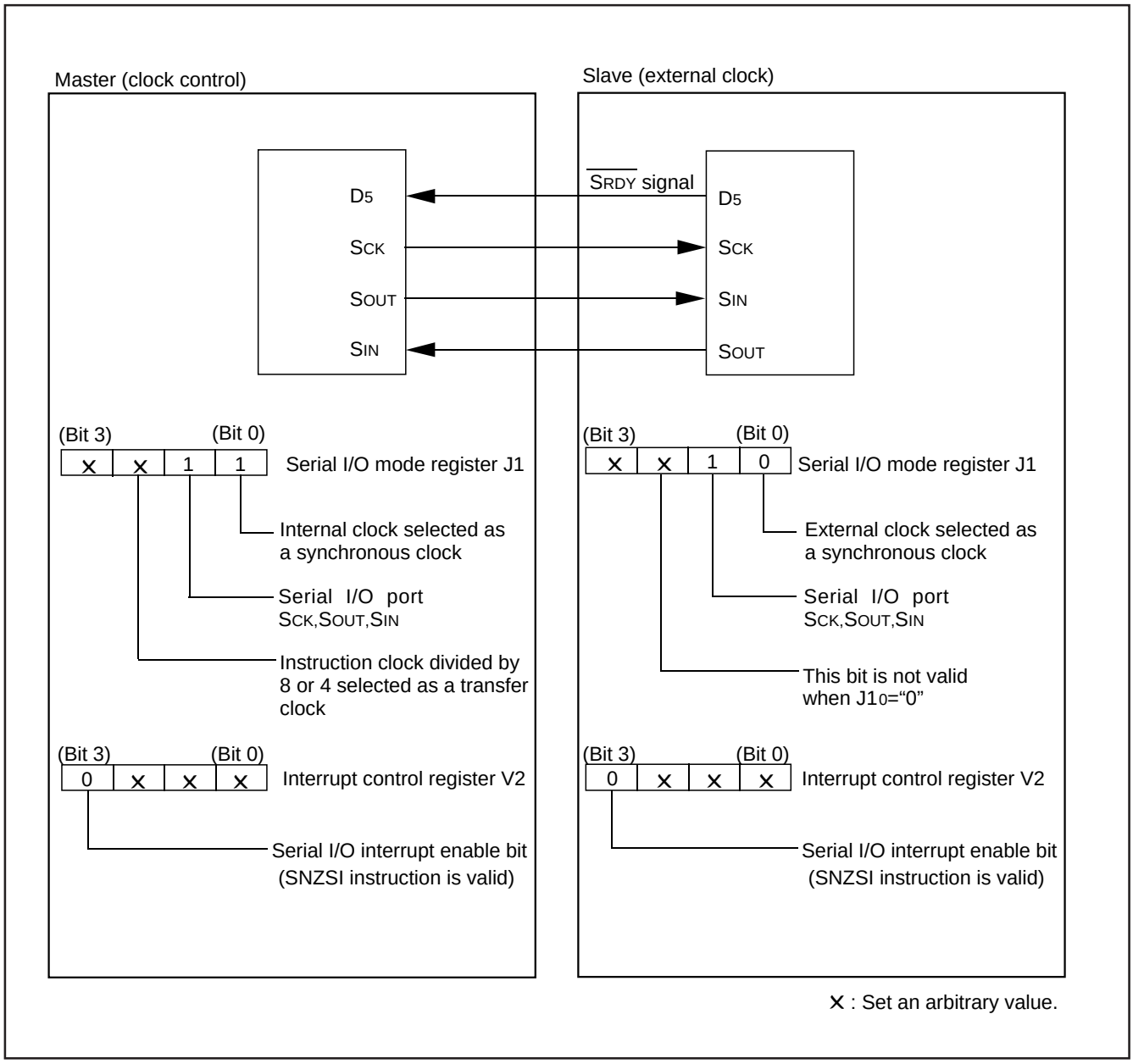
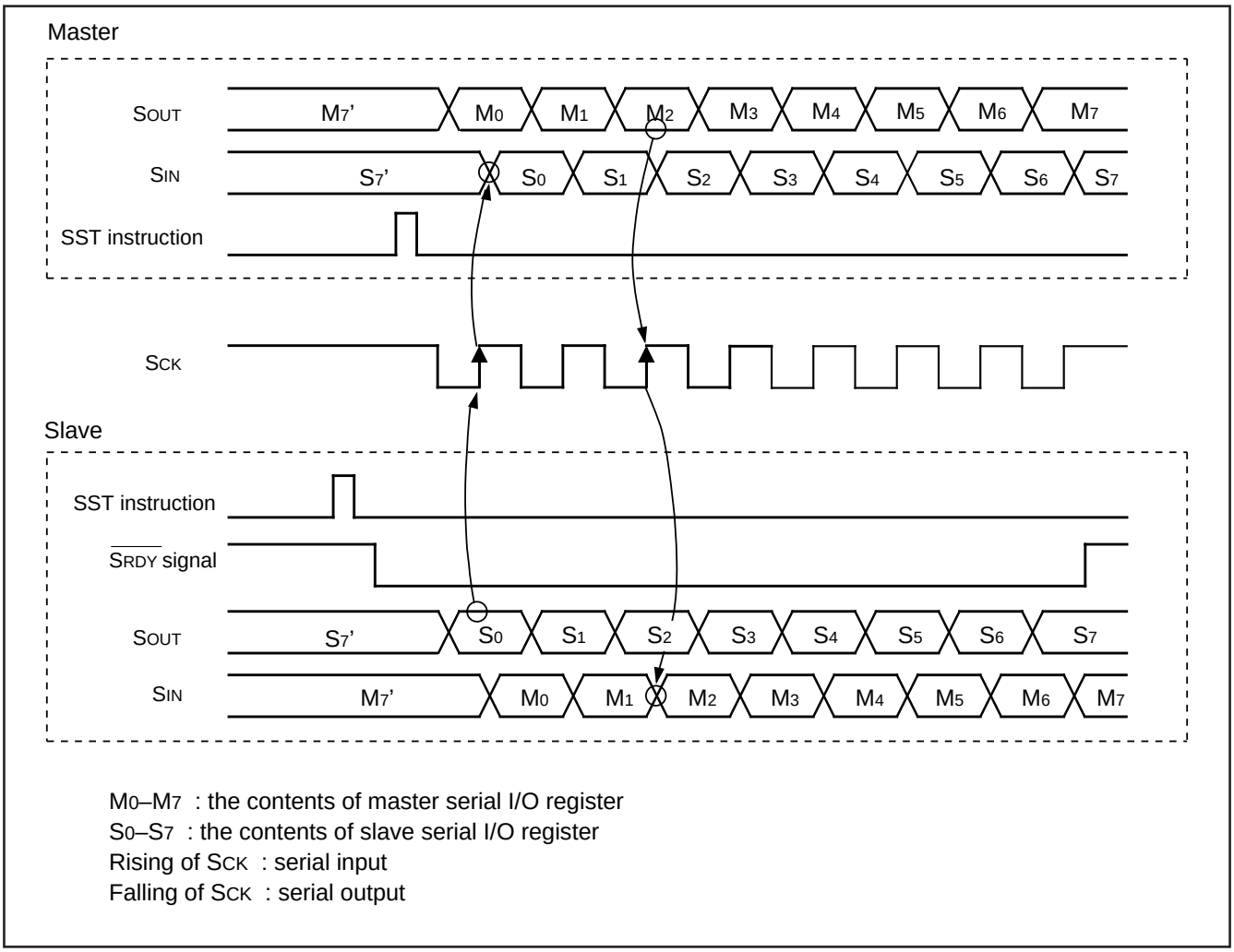


Fig. 24 Serial I/O connection example



M_0–M_7 : the contents of master serial I/O register
S_0–S_7 : the contents of slave serial I/O register
Rising of S_{CK} : serial input
Falling of S_{CK} : serial output

Fig. 25 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 24.	[Initial setting] • Setting serial I/O mode register J1, and interrupt control register V2 shown in Figure 24.
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 *.

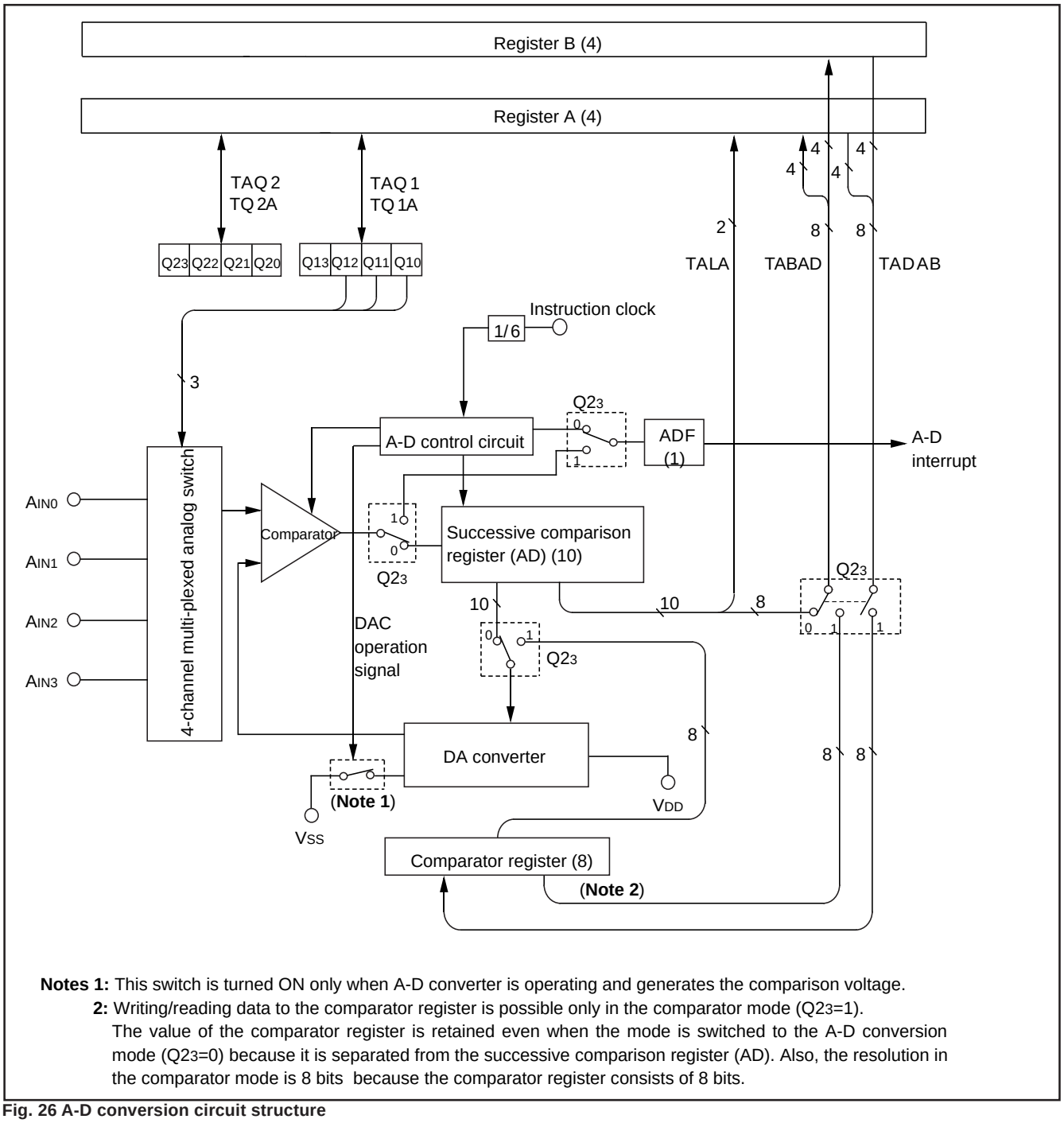
When an external clock is selected as a synchronous clock, the clock is not controlled internally. Control the clock externally because serial transfer is performed as long as clock is externally input. (Unlike an internal clock, an external clock is not stopped when serial transfer is completed.) However,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

The 4512 Group has a built-in A-D conversion circuit that performs conversion by 10-bit successive comparison method. Table 14 shows the characteristics of this A-D converter. This A-D converter can also be used as an 8-bit comparator to compare analog voltages input from the analog input pin with preset values.

Table 14 A-D converter characteristics

Parameter	Characteristics
Conversion format	Successive comparison method
Resolution	10 bits
Relative accuracy	Linearity error: ±2LSB
	Non-linearity error: ±0.9LSB
Conversion speed	46.5 μs (High-speed mode at 4.0 MHz oscillation frequency)
Analog input pin	4



Notes 1: This switch is turned ON only when A-D converter is operating and generates the comparison voltage.

2: Writing/reading data to the comparator register is possible only in the comparator mode (Q23=1).
The value of the comparator register is retained even when the mode is switched to the A-D conversion mode (Q23=0) because it is separated from the successive comparison register (AD). Also, the resolution in the comparator mode is 8 bits because the comparator register consists of 8 bits.

Fig. 26 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
$Q1_3$	Not used	0			This bit has no function, but read/write is enabled.	
		1				
$Q1_2$	Analog input pin selection bits	$Q1_2$	$Q1_1$	$Q1_0$	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
$Q2_3$	A-D operation mode selection bit	0			A-D conversion mode	
		1			Comparator mode	
$Q2_2$	Not used	0			This bit has no function, but read/write is enabled.	
		1				
$Q2_1$	Not used	0			This bit has no function, but read/write is enabled.	
		1				
$Q2_0$	Not used	0			This bit has no function, but read/write is enabled.	
		1				

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

(1) Operating at A-D conversion mode

The A-D conversion mode is set by setting the bit 3 of register Q2 to "0."

(2) Successive comparison register AD

Register AD stores the A-D conversion result of an analog input in 10-bit digital data format. The contents of the high-order 8 bits of this register can be stored in register B and register A with the TABAD instruction. The contents of the low-order 2 bits of this register can be stored into the high-order 2 bits of register A with the TALA instruction. However, do not execute 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_{DD} by the following formula:

Logic value of comparison voltage V_{ref}

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

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

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

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

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

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

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

(5) A-D control register Q1

Register Q1 is used to select one of analog input pins.

(6) A-D control register Q2

Register Q2 controls the A-D conversion mode. The A-D conversion mode is selected when the bit 3 of register Q2 is "0," and the comparator mode is selected when the bit 3 of register Q2 is "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 "000$_{16}$."
② Next, the topmost bit of the register AD is set to "1," and the comparison voltage V_{ref} is compared with the analog input voltage V_{IN}.
③ When the comparison result is $V_{ref} < V_{IN}$, the topmost bit of the register AD remains set to "1." When the comparison result is $V_{ref} > V_{IN}$, it is cleared to "0."

The 4512 Group repeats this operation to the lowermost bit of the register AD to convert an analog value to a digital value. A-D conversion stops after 62 machine cycles (46.5 μs when f(X_{IN}) = 4.0 MHz in high-speed mode) from the start, and the conversion result is stored in the register AD. An A-D interrupt activated condition is satisfied and the ADF flag is set to "1" as soon as A-D conversion completes (Figure 27).

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	$\frac{V_{DD}}{2}$
2nd comparison	✻1 1 0 ----- 0 0 0	$\frac{V_{DD}}{2} \pm \frac{V_{DD}}{4}$
3rd comparison	✻1 ✻2 1 ----- 0 0 0	$\frac{V_{DD}}{2} \pm \frac{V_{DD}}{4} \pm \frac{V_{DD}}{8}$
After 10th comparison completes	A-D conversion result ✻1 ✻2 ✻3 ----- ✻8 ✻9 ✻A	$\frac{V_{DD}}{2} \pm \cdots \pm \frac{V_{DD}}{1024}$

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

(8) A-D conversion timing chart

Figure 27 shows the A-D conversion timing chart.

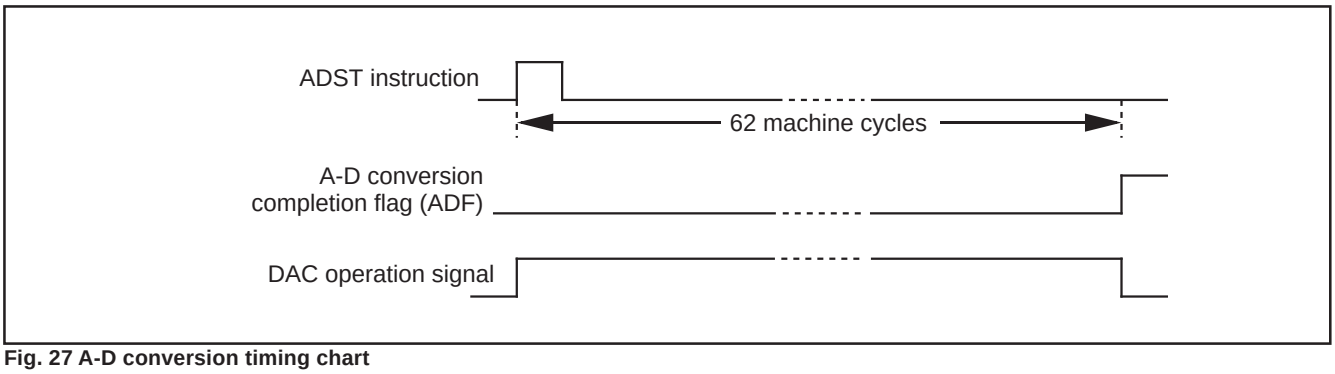


Fig. 27 A-D conversion timing chart

(9) Operation at comparator mode

The A-D converter is set to comparator mode by setting bit 3 of the register Q2 to "1."
Below, the operation at comparator mode is described.

(10) Comparator register

In comparator mode, the built-in DA comparator is connected to the comparator register as a register for setting comparison voltages. The contents of register B is stored in the high-order 4 bits of the comparator register and the contents of register A is stored in the low-order 4 bits of the comparator register with the TADAB instruction.
When changing from A-D conversion mode to comparator mode, the result of A-D conversion (register AD) is undefined.
However, because the comparator register is separated from register AD, the value is retained even when changing from comparator mode to A-D conversion mode. Note that the comparator register can be written and read at only comparator mode.
If the value in the comparator register is n, the logic value of comparison voltage V_{ref} generated by the built-in DA converter can be determined from the following formula:

Logic value of comparison voltage V_{ref}

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

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

(11) Comparison result store flag (ADF)

In comparator mode, the ADF flag, which shows completion of A-D conversion, stores the results of comparing the analog input voltage with the comparison voltage. When the analog input voltage is lower than the comparison voltage, the ADF flag is set to "1." The state of ADF flag can be examined with the skip instruction (SNZAD). Use the interrupt control register V2 to select the interrupt or the skip instruction.
The ADF flag is cleared to "0" when the interrupt occurs or when the next instruction is skipped with the skip instruction.

(12) Comparator operation start instruction (ADST instruction) (Figure 28)

In comparator mode, executing ADST starts the comparator operating.
The comparator stops 8 machine cycles after it has started (6 μs at f(X_{IN}) = 4.0 MHz in high-speed mode). When the analog input voltage is lower than the comparison voltage, the ADF flag is set to "1."

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

- TALA instruction
  When the TALA instruction is executed, the low-order 2 bits of register AD is transferred to the high-order 2 bits of register A, simultaneously, the low-order 2 bits of register A is "0."

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

Do not change the operating mode (both A-D conversion mode and comparator mode) of A-D converter with bit 3 of register Q2 while A-D converter is operating.
When the operating mode of A-D converter is changed from the comparator mode to A-D conversion mode with the bit 3 of register Q2, note the following;

- Clear bit 2 of register V2 to "0" to change the operating mode of the A-D converter from the comparator mode to A-D conversion mode with the bit 3 of register Q2.
- 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 Q2, and execute the SNZAD instruction to clear the ADF flag.

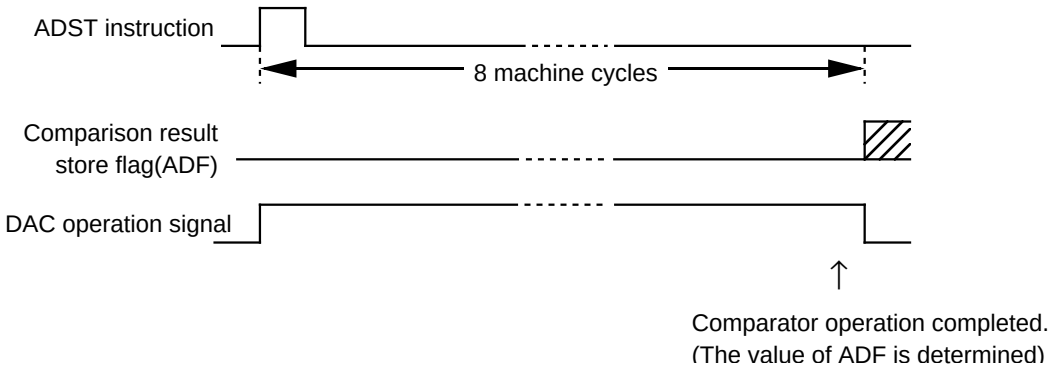


Fig. 28 Comparator operation timing chart

(15) Definition of A-D converter accuracy

The A-D conversion accuracy is defined below (refer to Figure 29).

- Relative accuracy
  - ① Zero transition voltage (V_{0T})
    This means an analog input voltage when the actual A-D conversion output data changes from “0” to “1.”
  - ② Full-scale transition voltage (V_{FST})
    This means an analog input voltage when the actual A-D conversion output data changes from “1023” to ”1022.”
  - ③ Linearity error
    This means a deviation from the line between V_{0T} and V_{FST} of a converted value between V_{0T} and V_{FST}.
  - ④ Differential non-linearity error
    This means a deviation from the input potential difference required to change a converter value between V_{0T} and V_{FST} by 1 LSB at the relative accuracy.

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

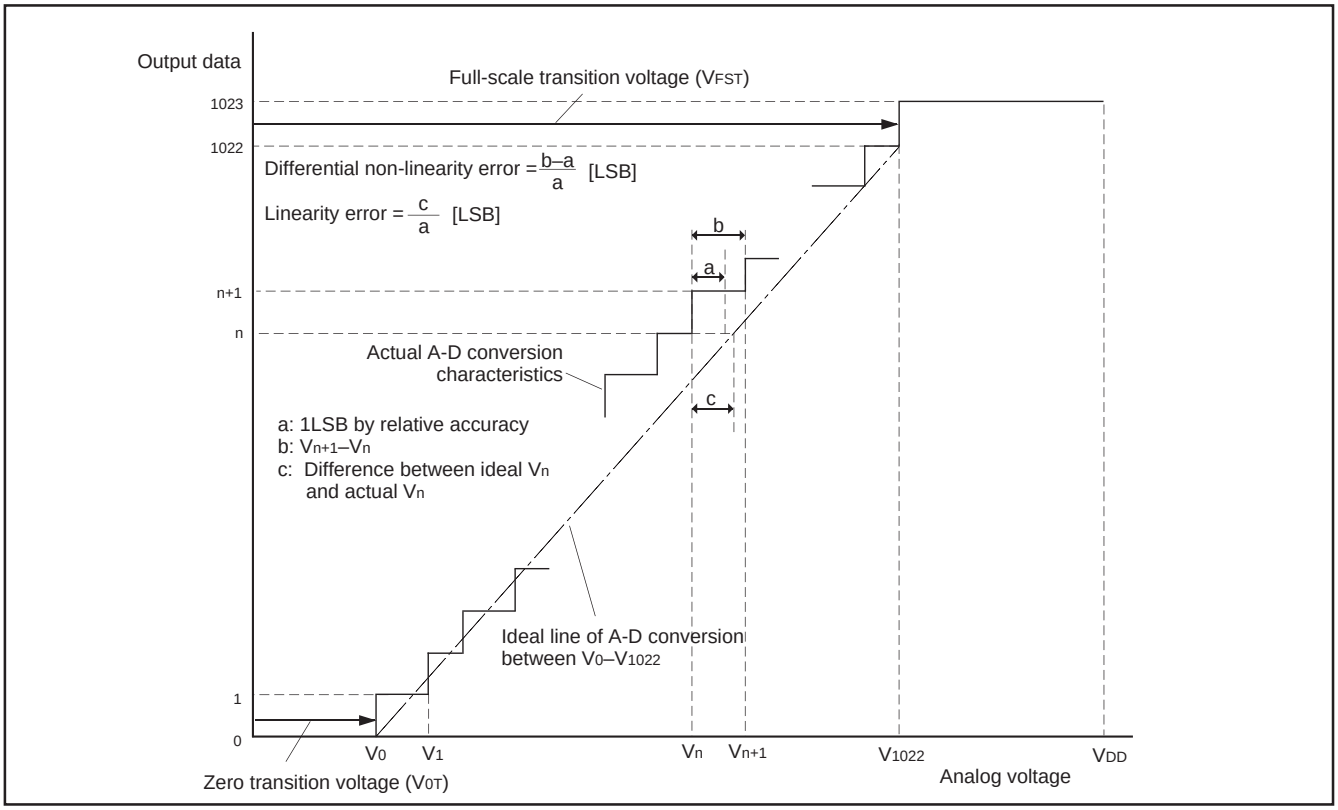


Fig. 29 Definition of A-D conversion accuracy

Vn: Analog input voltage when the output data changes from “n” to “n+1” (n = 0 to 1022)

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

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

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.

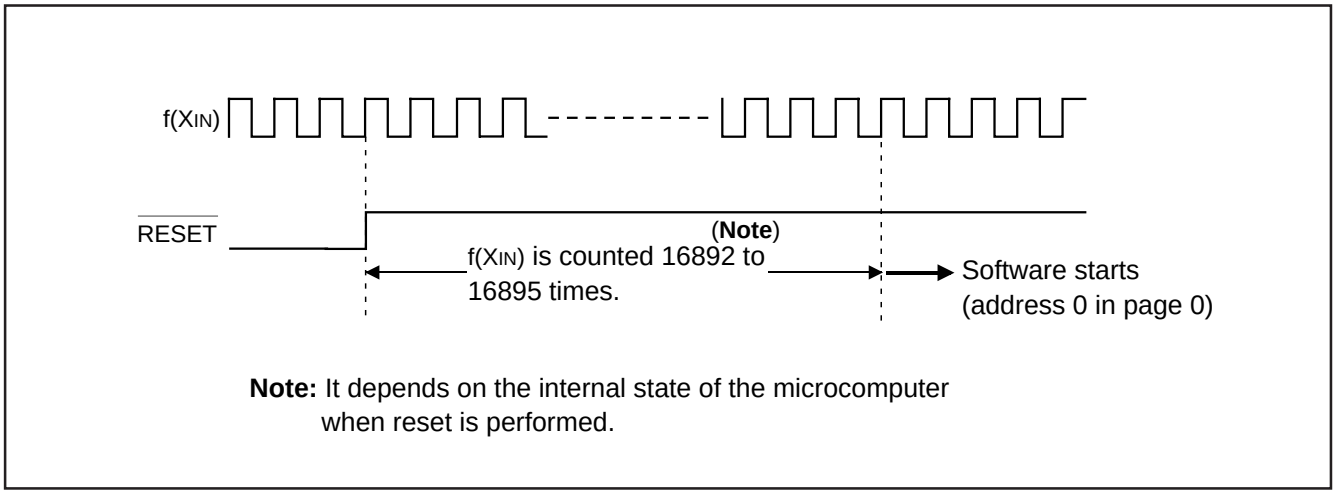


Fig. 30 Reset release timing

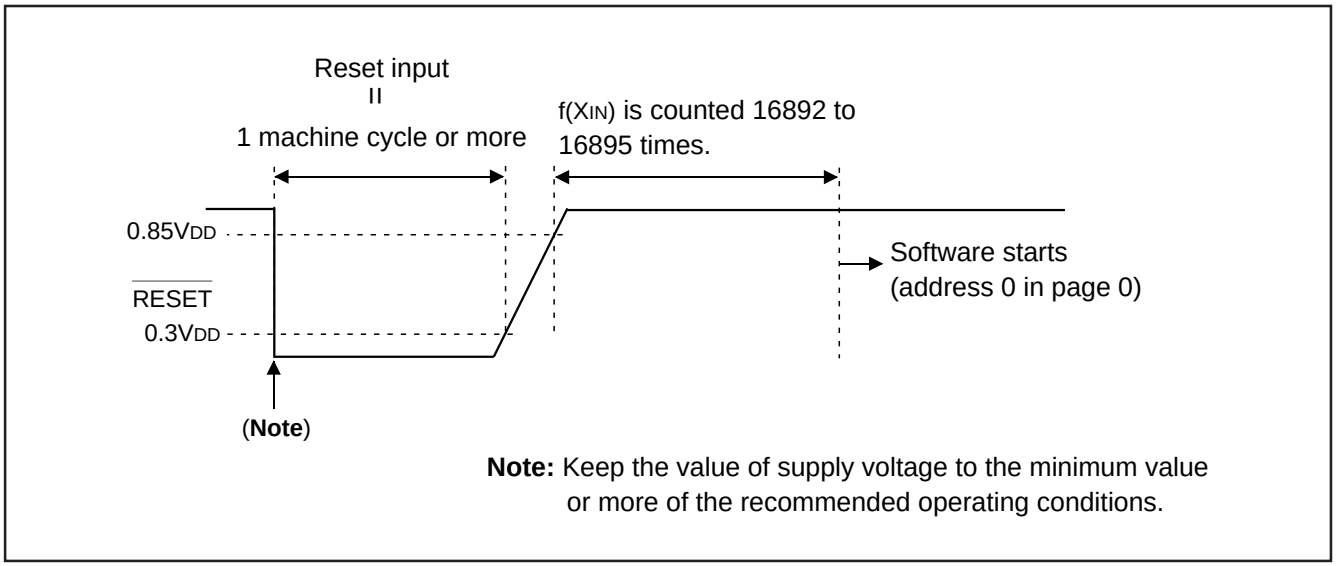


Fig. 31 $\overline{\text{RESET}}$ pin input waveform and reset operation

(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.

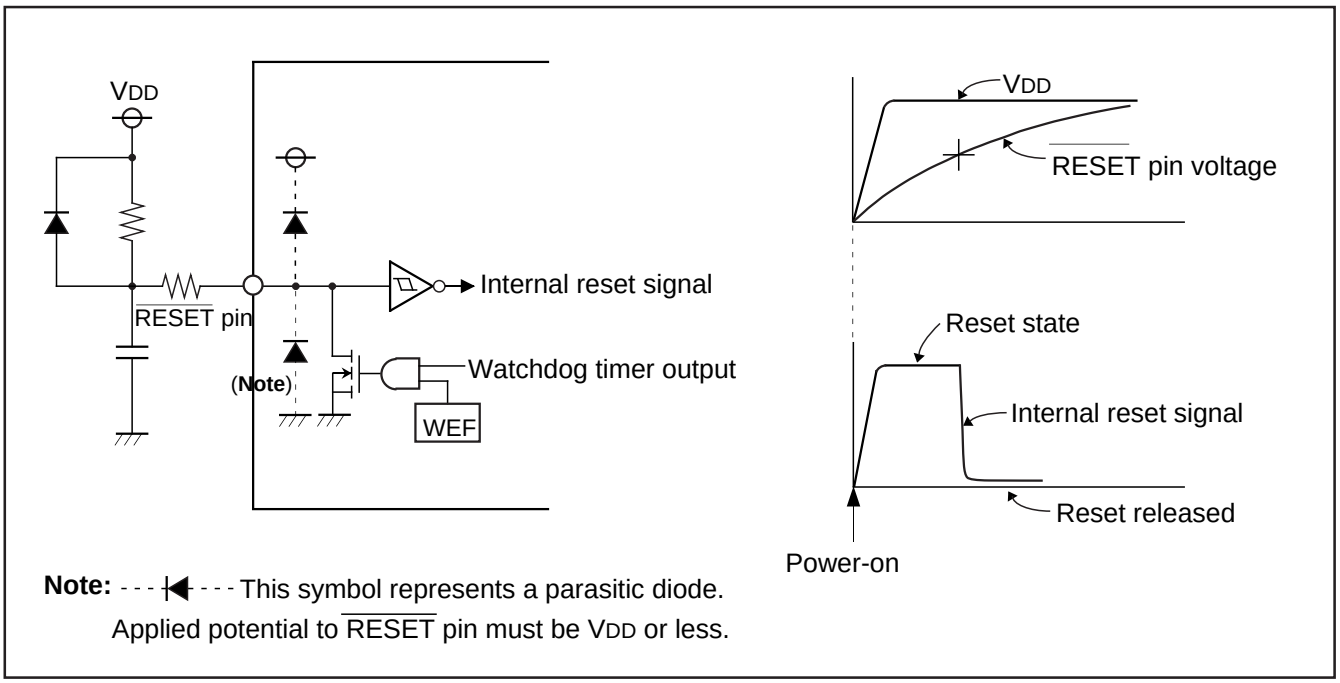


Fig. 32 Power-on reset circuit example

(2) Internal state at reset

Table 17 shows port state at reset, and Figure 33 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 33 are undefined, so set the initial value to them.

Table 17 Port state at reset

Name	Function	State
D_0–D_7	D_0–D_7	High impedance (Note 1)
$P0_0$–$P0_3$	$P0_0$–$P0_3$	High impedance (Notes 1, 2)
$P1_0$–$P1_3$	$P1_0$–$P1_3$	
$P2_0$/SCK, $P2_1$/SOUT, $P2_2$/SIN	$P2_0$–$P2_2$	High impedance

Notes 1: Output latch is set to "1."
2: Pull-up transistor is turned OFF.

- Program counter (PC) 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  Address 0 in page 0 is set to program counter.
- Interrupt enable flag (INTE) ..... 0 (Interrupt disabled)
- Power down flag (P) 0
- External 0 interrupt request flag (EXF0) 0
- External 1 interrupt request flag (EXF1) 0
- Interrupt control register V1 ..... 0 0 0 0 (Interrupt disabled)
- Interrupt control register V2 ..... 0 0 0 0 (Interrupt disabled)
- Interrupt control register I1 ..... 0 0 0 0
- Interrupt control register I2 ..... 0 0 0 0
- Timer 1 interrupt request flag (T1F) 0
- Timer 2 interrupt request flag (T2F) 0
- Timer 3 interrupt request flag (T3F) 0
- Timer 4 interrupt request flag (T4F) 0
- Watchdog timer flags (WDF1, WDF2) 0
- Watchdog timer enable flag (WEF) 0
- Timer control register W1 ..... 0 0 0 0 (Prescaler and timer 1 stopped)
- Timer control register W2 ..... 0 0 0 0 (Timer 2 stopped)
- Timer control register W3 ..... 0 0 0 0 (Timer 3 stopped)
- Timer control register W4 ..... 0 0 0 0 (Timer 4 stopped)
- Clock control register MR 1 0 0 0
- Serial I/O transmission/reception completion flag (SIOF) 0
- Serial I/O mode register J1 ..... 0 0 0 0 (External clock selected and serial I/O port not selected)
- Serial I/O register SI ..... X X X X X X X X
- A-D conversion completion flag (ADF) 0
- A-D control register Q1 ..... 0 0 0 0
- A-D control register Q2 ..... 0 0 0 0
- Successive comparison register AD ..... X X X X X X X X X X
- Comparator register ..... X X X X X X X X
- Key-on wakeup control register K0 ..... 0 0 0 0
- Pull-up control register PU0 ..... 0 0 0 0
- Carry flag (CY) 0
- Register A 0 0 0 0
- Register B 0 0 0 0
- Register D ..... X X X
- Register E ..... X X X X X X X X
- Register X 0 0 0 0
- Register Y 0 0 0 0
- Register Z ..... X X
- Stack pointer (SP) 1 1 1

"X" represents undefined.

Fig. 33 Internal state at reset

RAM BACK-UP MODE

The 4512 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 34 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

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	O
Clock control register MR	×
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
External 0 interrupt request flag (EXF0)	×
External 1 interrupt request flag (EXF1)	×
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 flags (WDF1, WDF2)	× (Note 4)
Watchdog timer enable flag (WEF)	× (Note 4)
16-bit timer (WDT)	× (Note 4)
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 "×" 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.
	INT0 pin	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.
	INT1 pin	Return by an external “H” level or “L” level input. The EXF1 flag is not set.	Select the return level (“L” level or “H” level) with the bit 2 of register I2 according to the external state before going into the RAM back-up state.

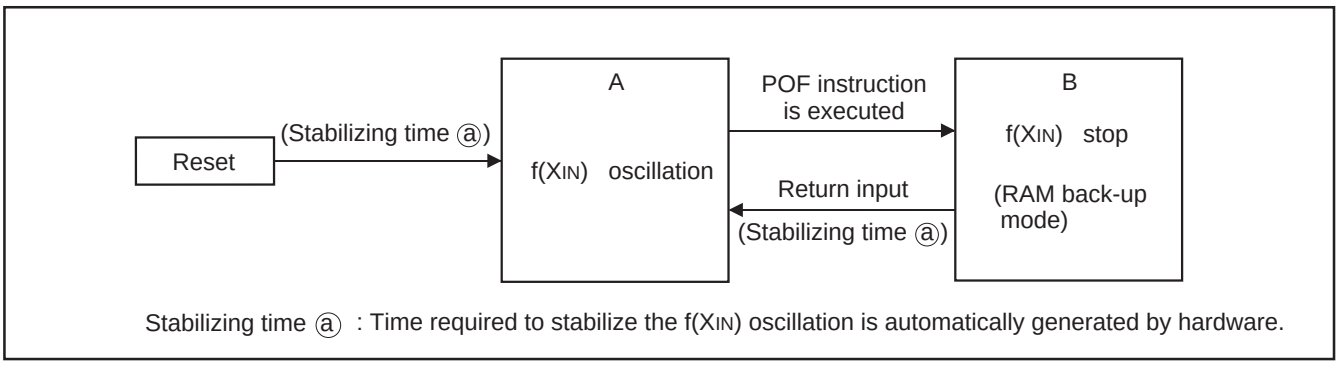


Fig. 34 State transition

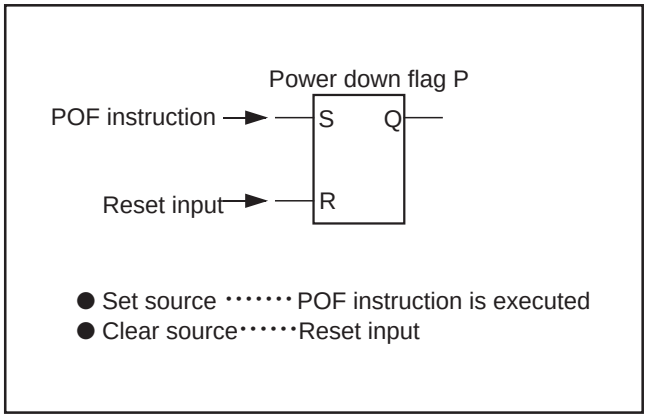


Fig. 35 Set source and clear source of the P flag

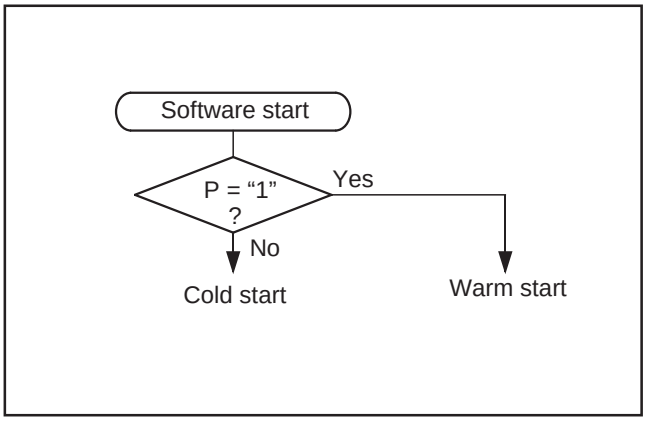


Fig. 36 Start condition identified example using the SNZP instruction

Table 20 Key-on wakeup control register, pull-up control register, and interrupt control register

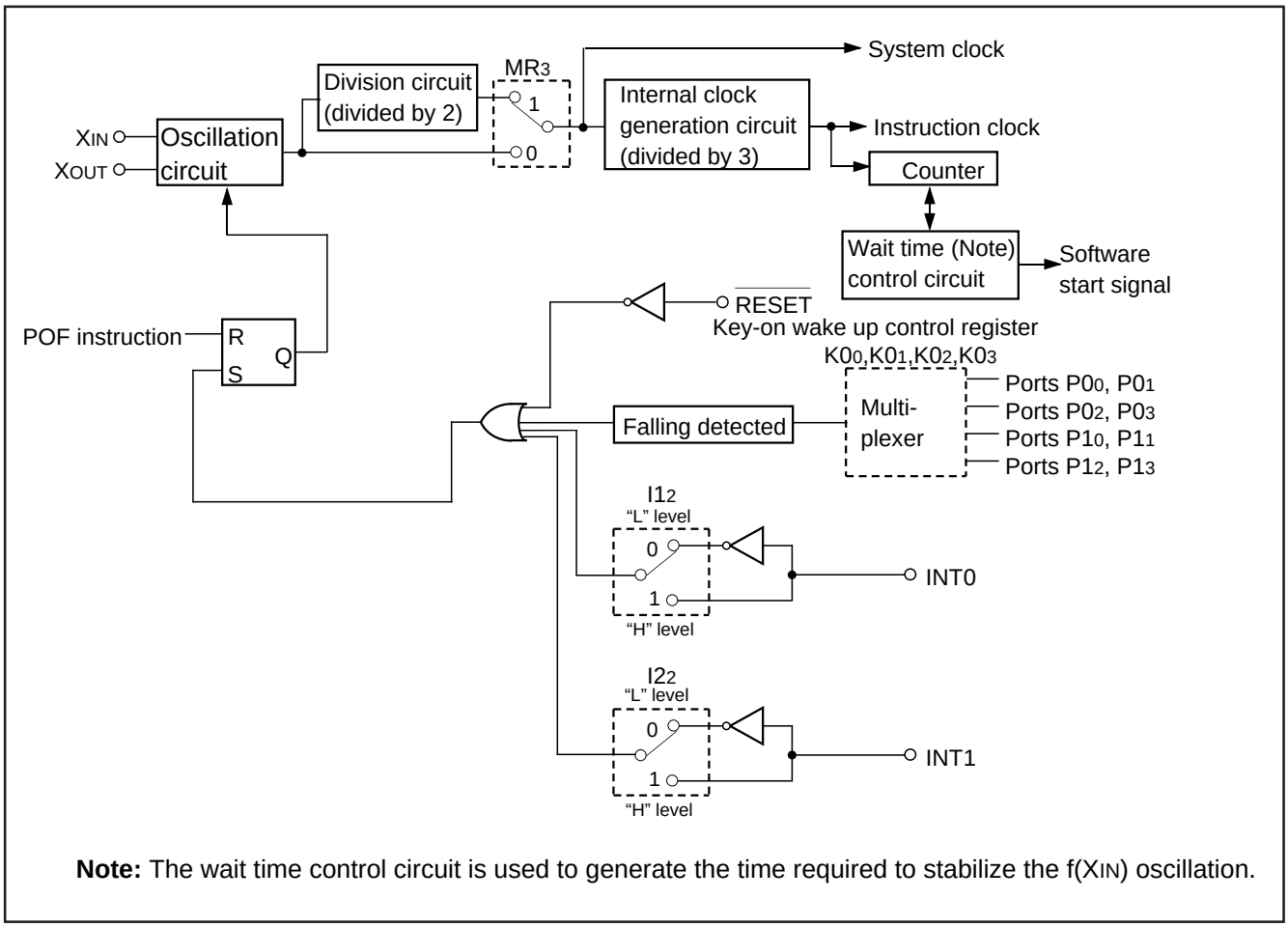
Key-on wakeup control register K0		at reset : 0000_2		at RAM back-up : state retained	R/W
$K0_3$	Pins $P1_2$ and $P1_3$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_2$	Pins $P1_0$ and $P1_1$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_1$	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		
Interrupt control register I1		at reset : 0000_2		at RAM back-up : state retained	R/W
$I1_3$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$I1_2$	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		
$I1_1$	INT0 pin edge detection circuit control bit	0	One-sided edge detected		
		1	Both edges detected		
$I1_0$	INT0 pin timer 1 control enable bit	0	Disabled		
		1	Enabled		
Interrupt control register I2		at reset : 0000_2		at RAM back-up : state retained	R/W
$I2_3$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$I2_2$	Interrupt valid waveform for INT1 pin/ return level selection bit (Note 3)	0	Falling waveform (“L” level of INT1 pin is recognized with the SNZI1 instruction)/“L” level		
		1	Rising waveform (“H” level of INT1 pin is recognized with the SNZI1 instruction)/“H” level		
$I2_1$	INT1 pin edge detection circuit control bit	0	One-sided edge detected		
		1	Both edges detected		
$I2_0$	INT1 pin timer 3 control enable bit	0	Disabled		
		1	Enabled		

Notes 1: “R” represents read enabled, and “W” represents write enabled.
2: When the contents of $I1_2$ is changed, the external interrupt request flag EXF0 may be set. Accordingly, clear EXF0 flag with the SNZ0 instruction.
3: When the contents of $I2_2$ is changed, the external interrupt request flag EXF1 may be set. Accordingly, clear EXF1 flag with the SNZ1 instruction.

CLOCK CONTROL

The clock control circuit consists of the following circuits.

- System clock generating circuit
- Control circuit to stop the clock oscillation
- Control circuit to switch the middle-speed mode and high-speed mode
- Control circuit to return from the RAM back-up state



Note: The wait time control circuit is used to generate the time required to stabilize the f(XIN) oscillation.

Fig. 37 Clock control circuit structure

(1) Clock control register

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

Table 23 Clock control register MR

Clock control register MR		at reset : 1000_2		at RAM back-up : 1000_2	R/W
MR_3	System clock selection bit	0	f(XIN) (high-speed mode)		
		1	f(XIN)/2 (middle-speed mode)		
MR_2	Not used	0	This bit has no function, but read/write is enabled.		
		1			
MR_1	Not used	0	This bit has no function, but read/write is enabled.		
		1			
MR_0	Not used	0	This bit has no function, but read/write is enabled.		
		1			

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

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 22.

Table 22 Maximum value of external clock oscillation frequency

Supply voltage	Oscillation frequency (duty ratio)
VDD = 4.0 V to 5.5 V	3.0 MHz (40 % to 60 %)

ROM ORDERING METHOD

Please submit the information described below when ordering Mask ROM.

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

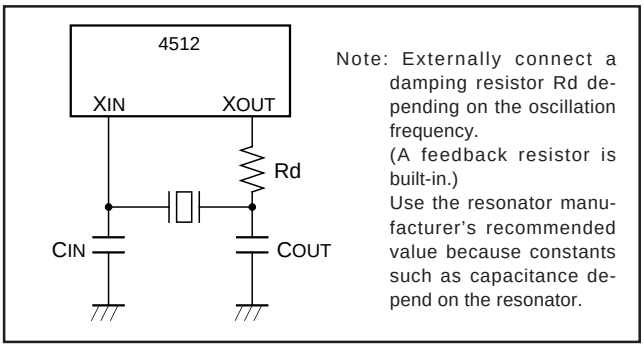


Fig. 38 Ceramic resonator external circuit

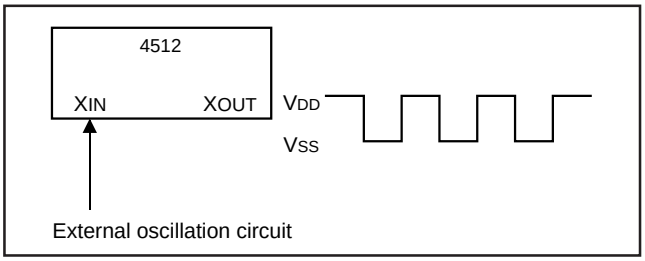


Fig. 39 External clock input circuit

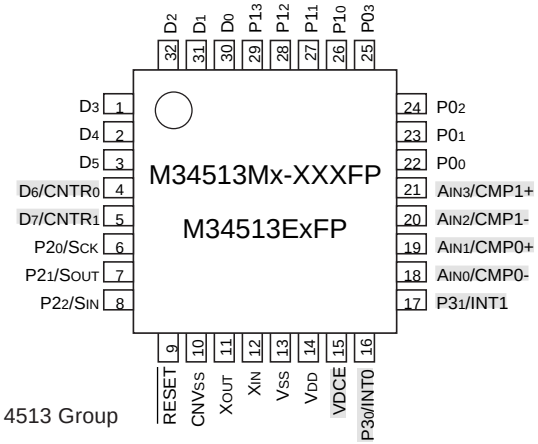
Differences from 4513 Group

The 4512 Group has pin-compatibility with the 4513 Group, but its function and recommended operating condition are different from the 4513 Group. The differences are as follows:

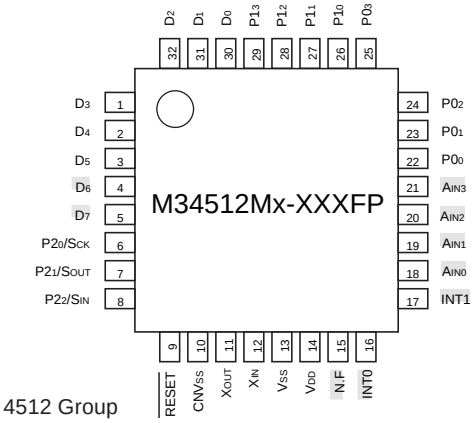
Parameter		4513 Group	4512 Group
Hardware	Basic instructions	123	117 (Note 2)
	Voltage drop detection circuit	O	×
	Voltage comparator	O	×
	Timer I/O (CNTR0, CNTR1)	O	×
	Ports $P3_0$, $P3_1$	O	×
	Timer control register W6	O	×
	Voltage comparator control register W6	O	×
Recommended operating condition	Supply voltage	2.5 V to 5.5 V	4.0 V to 5.5 V
		2.0 V to 5.5 V	
		4.0 V to 5.5 V	
		2.5 V to 5.5 V	
		2.0 V to 5.5 V	
Others	Large size memory version supported	O (M34513M6/M8/E8)	×
	DIP package supported	O (M34513M2/M4/E4)	×
	One Time PROM version supported	O (M34513E4/E8)	× (M34513E4 can be used)

Notes 1: "O" represents that the function is supported, "X" represents that the function is not supported.
2: For the 4512 Group, TQ3A, TAQ3, TW6A, TAW6, OP3A and IAP3 instructions cannot be used.

PIN CONFIGURATIONS



4513 Group



4512 Group

4513 Group	Pin number	4512 Group
D_3	1	D_3
D_4	2	D_4
D_5	3	D_5
D_6/CNTR0	4	D_6
D_7/CNTR1	5	D_7
$P2_0$/SCK	6	$P2_0$/SCK
$P2_1$/SOUT	7	$P2_1$/SOUT
$P2_2$/SIN	8	$P2_2$/SIN
$\overline{\text{RESET}}$	9	$\overline{\text{RESET}}$
CNVSS	10	CNVSS
XOUT	11	XOUT
XIN	12	XIN
VSS	13	VSS
VDD	14	VDD
VDCE	15	N.F
$P3_0$/INT0	16	INT0
$P3_1$/INT1	17	INT1
AIN0/CMP0-	18	AIN0
AIN1/CMP0+	19	AIN1
AIN2/CMP1-	20	AIN2
AIN3/CMP1+	21	AIN3
$P0_0$	22	$P0_0$
$P0_1$	23	$P0_1$
$P0_2$	24	$P0_2$
$P0_3$	25	$P0_3$
$P1_0$	26	$P1_0$
$P1_1$	27	$P1_1$
$P1_2$	28	$P1_2$
$P1_3$	29	$P1_3$
D_0	30	D_0
D_1	31	D_1
D_2	32	D_2

LIST OF PRECAUTIONS

①Noise 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 V_{SS} at the shortest distance,
- equalize its wiring in width and length, and
- use relatively thick wire.

②Prescaler

Stop the prescaler operation to change its frequency dividing ratio.

③Timer count source

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

④Reading 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.

⑤Writing to reload registers R1 and R3

When writing data to reload registers R1 or R3 while timer 1 or timer 3 is operating, avoid a timing when timer 1 or timer 3 underflows.

⑥INT0 pin

When the interrupt valid waveform of the INT0 pin is changed with the bit 2 of register I1 in software, be careful about the following notes.

- Clear the bit 0 of register V1 to "0" before the interrupt valid waveform of INT0 pin is changed with the bit 2 of register I1 (refer to Figure 40①).
- Depending on the input state of the INT0 pin, the external 0 interrupt request flag (EXF0) may be set when the interrupt valid waveform is changed. Accordingly, clear bit 2 of register I1, and execute the SNZ0 instruction to clear the EXF0 flag after executing at least one instruction (refer to Figure 40②)

```
    ⋮
LA    4        ; (XXX0₂)
TV1A           ; The SNZ0 instruction is valid .....①
LA    4        ;
TI1A           ; Interrupt valid waveform is changed
NOP            ..... ②
SNZ0           ; The SNZ0 instruction is executed
NOP
    ⋮
```

X : this bit is not related to the setting of INT0 pin.

Fig. 40 External 0 interrupt program example

⑦INT1 pin

When the interrupt valid waveform of INT1 pin is changed with the bit 2 of register I2 in software, be careful about the following notes.

- Clear the bit 1 of register V1 to "0" before the interrupt valid waveform of INT1 pin is changed with the bit 2 of register I2 (refer to Figure 41③).
- Depending on the input state of the INT1 pin, the external 1 interrupt request flag (EXF1) may be set when the interrupt valid waveform is changed. Accordingly, clear bit 2 of register I2 and execute the SNZ1 instruction to clear the EXF1 flag after executing at least one instruction (refer to Figure 41④).

```
    ⋮
LA    8        ; (XX0X₂)
TV1A           ; The SNZ1 instruction is valid .....③
LA    8
TI2A           ; Change of the interrupt valid waveform
NOP            ..... ④
SNZ1           ; The SNZ1 instruction is executed
NOP
    ⋮
```

X : this bit is not related to the setting of INT1.

Fig. 41 External 1 interrupt program example

⑧Multifunction

The input of $P2_0$–$P2_2$ can be used even when SCK, SOUT, SIN 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 Q2 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 Q2 (refer to Figure 42⑤).
- 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 Q2, 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 Q2 during operating the A-D converter.

```
    ⋮
LA    8        ; (X0XX2)
TV2A           ; The SNZAD instruction is valid .....⑤
LA    0        ; (0XXX2)
TQ2A           ; 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. 42 A-D converter operating mode program example

⑩ A-D converter-2

Each analog input pin is equipped with a capacitor which is used to compare the analog voltage. Accordingly, when the analog voltage is input from the circuit with high-impedance and, charge/discharge noise is generated and the sufficient A-D accuracy may not be obtained. Therefore, reduce the impedance or, connect a capacitor (0.01 μF to 1 μF) to analog input pins (Figure 43).

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 44. In addition, test the application products sufficiently.

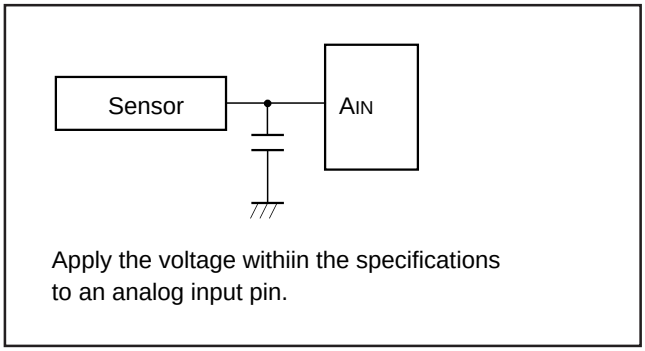


Fig. 43 Analog input external circuit example-1

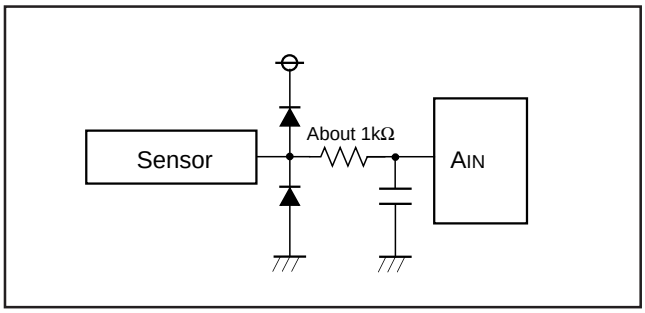


Fig. 44 Analog input external circuit example-2

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 TALA instruction

When the TALA instruction is executed, the low-order 2 bits of register AD is transferred to the high-order 2 bits of register A, simultaneously, the low-order 2 bits of register A is "0."

⑬ Program counter

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

⑭ N.F pin

This pin has no function, and connect to V_{SS} as an unused pin.

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
$I1_3$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$I1_2$	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	
$I1_1$	INT0 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I1_0$	INT0 pin timer 1 control enable bit	0	Disabled	
		1	Enabled	

Interrupt control register I2		at reset : 0000_2	at RAM back-up : state retained	R/W
$I2_3$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$I2_2$	Interrupt valid waveform for INT1 pin/ return level selection bit (Note 3)	0	Falling waveform (“L” level of INT1 pin is recognized with the SNZI1 instruction)/“L” level	
		1	Rising waveform (“H” level of INT1 pin is recognized with the SNZI1 instruction)/“H” level	
$I2_1$	INT1 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I2_0$	INT1 pin timer 3 control enable bit	0	Disabled	
		1	Enabled	

Notes 1: “R” represents read enabled, and “W” represents write enabled.
2: When the contents of $I1_2$ is changed, the external interrupt request flag EXF0 may be set. Accordingly, clear EXF0 flag with the SNZ0 instruction.
3: When the contents of $I2_2$ is changed, the external interrupt request flag EXF1 may be set. Accordingly, clear EXF1 flag with the SNZ1 instruction.

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 4		
		1	Instruction clock divided by 16		
$W1_1$	Timer 1 control bit	0	Stop (state retained)		
		1	Operating		
$W1_0$	Timer 1 count start synchronous circuit control bit	0	Count start synchronous circuit not selected		
		1	Count start synchronous circuit selected		

Timer control register W2		at reset : 0000_2		at RAM back-up : state retained	R/W
$W2_3$	Timer 2 control bit	0	Stop (state retained)		
		1	Operating		
$W2_2$	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	Not available	
		1	1	16 bit timer (WDT) underflow signal	

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	Not available	
		1	1	Not available	

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	This bit has no function, but read/write is enabled.	
		1		
$J1_2$	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	Input ports $P2_0$, $P2_1$, $P2_2$ selected	
		1	Serial I/O ports SCK, SOUT, SIN/input ports $P2_0$, $P2_1$, $P2_2$ 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
$Q1_3$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$Q1_2$	Analog input pin selection bits	$Q1_2$ $Q1_1$ $Q1_0$	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
$Q2_3$	A-D operation mode selection bit	0	A-D conversion mode	
		1	Comparator mode	
$Q2_2$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$Q2_1$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$Q2_0$	Not used	0	This bit has no function, but read/write is enabled.	
		1		

Note: “R” represents read enabled, and “W” represents write enabled.

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		
Clock control register MR		at reset : 1000_2		at RAM back-up : 1000_2	R/W
MR_3	System clock selection bit	0	$f(X_{IN})$ (high-speed mode)		
		1	$f(X_{IN})/2$ (middle-speed mode)		
MR_2	Not used	0	This bit has no function, but read/write is enabled.		
		1			
MR_1	Not used	0	This bit has no function, but read/write is enabled.		
		1			
MR_0	Not used	0	This bit has no function, but read/write is enabled.		
		1			

Note : “R” represents read enabled, and “W” represents write enabled.

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)	T1F	Timer 1 interrupt request flag
B	Register B (4 bits)	T2F	Timer 2 interrupt request flag
DR	Register D (3 bits)	T3F	Timer 3 interrupt request flag
E	Register E (8 bits)	T4F	Timer 4 interrupt request flag
Q1	A-D control register Q1 (4 bits)	WDF1, WDF2	Watchdog timer flag
Q2	A-D control register Q2 (4 bits)	WEF	Watchdog timer enable flag
AD	Successive comparison register AD (10 bits)	INTE	Interrupt enable flag
J1	Serial I/O mode register J1 (4 bits)	EXF0	External 0 interrupt request flag
SI	Serial I/O register SI (8 bits)	EXF1	External 1 interrupt request flag
V1	Interrupt control register V1 (4 bits)	P	Power down flag
V2	Interrupt control register V2 (4 bits)	ADF	A-D conversion completion flag
I1	Interrupt control register I1 (4 bits)	SIOF	Serial I/O transmission/reception completion flag
I2	Interrupt control register I2 (4 bits)		
W1	Timer control register W1 (4 bits)	D	Port D (8 bits)
W2	Timer control register W2 (4 bits)	P0	Port P0 (4 bits)
W3	Timer control register W3 (4 bits)	P1	Port P1 (4 bits)
W4	Timer control register W4 (4 bits)	P2	Port P2 (3 bits)
MR	Clock control register MR (4 bits)		
K0	Key-on wakeup control register K0 (4 bits)	x	Hexadecimal variable
PU0	Pull-up control register PU0 (4 bits)	y	Hexadecimal variable
X	Register X (4 bits)	z	Hexadecimal variable
Y	Register Y (4 bits)	p	Hexadecimal variable
Z	Register Z (2 bits)	n	Hexadecimal constant
DP	Data pointer (10 bits)	i	Hexadecimal constant
	(It consists of registers X, Y, and Z)	j	Hexadecimal constant
PC	Program counter (14 bits)	$A_3A_2A_1A_0$	Binary notation of hexadecimal variable A
PC_H	High-order 7 bits of program counter		(same for others)
PC_L	Low-order 7 bits of program counter		
SK	Stack register (14 bits × 8)	←	Direction of data movement
SP	Stack pointer (3 bits)	↔	Data exchange between a register and memory
CY	Carry flag	?	Decision of state shown before "?"
R1	Timer 1 reload register	( )	Contents of registers and memories
R2	Timer 2 reload register	—	Negate, Flag unchanged after executing instruction
R3	Timer 3 reload register	M(DP)	RAM address pointed by the data pointer
R4	Timer 4 reload register	a	Label indicating address a_6 a_5 a_4 a_3 a_2 a_1 a_0
T1	Timer 1	p, a	Label indicating address a_6 a_5 a_4 a_3 a_2 a_1 a_0
T2	Timer 2		in page p_5 p_4 p_3 p_2 p_1 p_0
T3	Timer 3	C	Hex. C + Hex. number x (also same for others)
T4	Timer 4	+	
		x	

Note : The 4512 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
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 → $A_3A_2A_1A_0$ →
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	$I2_2$ = 1 : (INT1) = "H" ? $I2_2$ = 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)
Timer operation	TAW4	(A) ← (W4)
	TW4A	(W4) ← (A)
	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)
	TR1AB	($R1_7$–$R1_4$) ← (B) ($R1_3$–$R1_0$) ← (A)
	TR3AB	($R3_7$–$R3_4$) ← (B) ($R3_3$–$R3_0$) ← (A)
Timer operation	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_2–A_0) ← ($P2_2$–$P2_0$) (A_3) ← 0
	CLD	(D) ← 1
	RD	(D(Y)) ← 0 (Y) = 0 to 7
	SD	(D(Y)) ← 1 (Y) = 0 to 7
	SZD	(D(Y)) = 0 ? (Y) = 0 to 7
	TK0A	(K0) ← (A)
	TAK0	(A) ← (K0)
	TPU0A	(PU0) ← (A)
	TAPU0	(A) ← (PU0)

LIST OF INSTRUCTION FUNCTION (continued)

Group-ing	Mnemonic	Function	Group-ing	Mnemonic	Function
Serial I/O control operation	TABSI	(A) ← (SI_3–SI_0) (B) ← (SI_7–SI_4)	Other operation	NOP	(PC) ← (PC) + 1
				POF	RAM back-up
	TSIAB	(SI_3–SI_0) ← (A) (SI_7–SI_4) ← (B)		EPOF	POF instruction valid
	TAJ1	(A) ← (J1)		SNZP	(P) = 1 ?
	TJ1A	(J1) ← (A)		WRST	(WDF1) ← 0, (WEF) ← 1
	SST	(SIOF) ← 0 Serial I/O starting		TAMR	(A) ← (MR)
				TMRA	(MR) ← (A)
	SNZSI	(SIOF) = 1 ? After skipping (SIOF) ← 0			
A-D conversion operation	TABAD	(A) ← (AD_5–AD_2) (B) ← (AD_9–AD_6) However, in the comparator mode, (A) ← (AD_3–AD_0) (B) ← (AD_7–AD_4)			
	TALA	(A) ← (AD_1, AD_0, 0, 0)			
	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)			

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 010111	011000 011111
		00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10–17	18–1F
0000	0	NOP	BLA	SZB 0	BMLA	–	TASP	A 0	LA 0	TABP 0	TABP 16***	–	–	BML	BML***	BL	BL***	BM	B
0001	1	–	CLD	SZB 1	–	–	TAD	A 1	LA 1	TABP 1	TABP 17***	–	–	BML	BML***	BL	BL***	BM	B
0010	2	POF	–	SZB 2	–	–	TAX	A 2	LA 2	TABP 2	TABP 18***	–	–	BML	BML***	BL	BL***	BM	B
0011	3	SNZP	INY	SZB 3	–	–	TAZ	A 3	LA 3	TABP 3	TABP 19***	–	–	BML	BML***	BL	BL***	BM	B
0100	4	DI	RD	SZD	–	RT	TAV1	A 4	LA 4	TABP 4	TABP 20***	–	–	BML	BML***	BL	BL***	BM	B
0101	5	EI	SD	SEAn	–	RTS	TAV2	A 5	LA 5	TABP 5	TABP 21***	–	–	BML	BML***	BL	BL***	BM	B
0110	6	RC	–	SEAM	–	RTI	–	A 6	LA 6	TABP 6	TABP 22***	–	–	BML	BML***	BL	BL***	BM	B
0111	7	SC	DEY	–	–	–	–	A 7	LA 7	TABP 7	TABP 23***	–	–	BML	BML***	BL	BL***	BM	B
1000	8	–	AND	–	SNZ0	LZ 0	–	A 8	LA 8	TABP 8	TABP 24***	–	–	BML	BML***	BL	BL***	BM	B
1001	9	–	OR	TDA	SNZ1	LZ 1	–	A 9	LA 9	TABP 9	TABP 25***	–	–	BML	BML***	BL	BL***	BM	B
1010	A	AM	TEAB	TABE	SNZI0	LZ 2	–	A 10	LA 10	TABP 10	TABP 26***	–	–	BML	BML***	BL	BL***	BM	B
1011	B	AMC	–	–	SNZI1	LZ 3	EPOF	A 11	LA 11	TABP 11	TABP 27***	–	–	BML	BML***	BL	BL***	BM	B
1100	C	TYA	CMA	–	–	RB 0	SB 0	A 12	LA 12	TABP 12	TABP 28***	–	–	BML	BML***	BL	BL***	BM	B
1101	D	–	RAR	–	–	RB 1	SB 1	A 13	LA 13	TABP 13	TABP 29***	–	–	BML	BML***	BL	BL***	BM	B
1110	E	TBA	TAB	–	TV2A	RB 2	SB 2	A 14	LA 14	TABP 14	TABP 30***	–	–	BML	BML***	BL	BL***	BM	B
1111	F	–	TAY	SZC	TV1A	RB 3	SB 3	A 15	LA 15	TABP 15	TABP 31***	–	–	BML	BML***	BL	BL***	BM	B

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

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

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

• *** cannot be used in the M34512M2-XXXFP.

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 111111
		20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30–3F
0000	0	–	TW3A	OP0A	T1AB	–	–	IAP0	TAB1	SNZT1	–	WRST	TMA 0	TAM 0	XAM 0	XAMI 0	XAMD 0	LXY
0001	1	–	TW4A	OP1A	T2AB	–	–	IAP1	TAB2	SNZT2	–	–	TMA 1	TAM 1	XAM 1	XAMI 1	XAMD 1	LXY
0010	2	TJ1A	–	–	T3AB	TAJ1	TAMR	IAP2	TAB3	SNZT3	–	–	TMA 2	TAM 2	XAM 2	XAMI 2	XAMD 2	LXY
0011	3	–	–	–	T4AB	–	TAI1	–	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	–	TMRA	–	–	–	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	–	–	–	TADAB	TALA	–	–	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	–	TR3AB	TAW1	–	–	–	–	–	–	TMA 11	TAM 11	XAM 11	XAMI 11	XAMD 11	LXY
1100	C	–	–	–	–	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	–	–	TR1AB	–	–	–	–	–	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. D3–D0 show the low-order 4 bits of the machine language code, and D9–D4 show the high-order 6 bits of the machine language code. The hexadecimal representation of the code is also provided. There are one-word instructions and two-word instructions, but only the first word of each instruction is shown. Do not use code marked "–."

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

	The second word
BL	10 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											Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			
Register to register transfer	TAB	0	0	0	0	0	1	1	1	1	0	0 1 E	1	1	(A) ← (B)
	TBA	0	0	0	0	0	0	1	1	1	0	0 0 E	1	1	(B) ← (A)
	TAY	0	0	0	0	0	1	1	1	1	1	0 1 F	1	1	(A) ← (Y)
	TYA	0	0	0	0	0	0	1	1	0	0	0 0 C	1	1	(Y) ← (A)
	TEAB	0	0	0	0	0	1	1	0	1	0	0 1 A	1	1	(E_7–E_4) ← (B) (E_3–E_0) ← (A)
	TABE	0	0	0	0	1	0	1	0	1	0	0 2 A	1	1	(B) ← (E_7–E_4) (A) ← (E_3–E_0)
	TDA	0	0	0	0	1	0	1	0	0	1	0 2 9	1	1	(DR_2–DR_0) ← (A_2–A_0)
	TAD	0	0	0	1	0	1	0	0	0	1	0 5 1	1	1	(A_2–A_0) ← (DR_2–DR_0) (A_3) ← 0
	TAZ	0	0	0	1	0	1	0	0	1	1	0 5 3	1	1	(A_1, A_0) ← (Z_1, Z_0) (A_3, A_2) ← 0
	TAX	0	0	0	1	0	1	0	0	1	0	0 5 2	1	1	(A) ← (X)
	TASP	0	0	0	1	0	1	0	0	0	0	0 5 0	1	1	(A_2–A_0) ← (SP_2–SP_0) (A_3) ← 0
RAM addresses	LXY x, y	1	1	x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0	3 x y	1	1	(X) ← x, x = 0 to 15 (Y) ← y, y = 0 to 15
	LZ z	0	0	0	1	0	0	1	0	z_1	z_0	0 4 8 +z	1	1	(Z) ← z, z = 0 to 3
	INY	0	0	0	0	0	1	0	0	1	1	0 1 3	1	1	(Y) ← (Y) + 1
	DEY	0	0	0	0	0	1	0	1	1	1	0 1 7	1	1	(Y) ← (Y) – 1
RAM to register transfer	TAM j	1	0	1	1	0	0	j	j	j	j	2 C j	1	1	(A) ← (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15
	XAM j	1	0	1	1	0	1	j	j	j	j	2 D j	1	1	(A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15
	XAMD j	1	0	1	1	1	1	j	j	j	j	2 F j	1	1	(A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) – 1
	XAMI j	1	0	1	1	1	0	j	j	j	j	2 E j	1	1	(A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) + 1
	TMA j	1	0	1	0	1	1	j	j	j	j	2 B j	1	1	(M(DP)) ← (A) (X) ← (X)EXOR(j) j = 0 to 15

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	→ CY → $A_3A_2A_1A_0$ →
Bit operation	SB j	0	0	0	1	0	1	1	1	j	j	0 5 C +j	1	1	(Mj(DP)) ← 1 j = 0 to 3
	RB j	0	0	0	1	0	0	1	1	j	j	0 4 C +j	1	1	(Mj(DP)) ← 0 j = 0 to 3
	SZB j	0	0	0	0	1	0	0	0	j	j	0 2 j	1	1	(Mj(DP)) = 0 ? j = 0 to 3
Comparison operation	SEAM	0	0	0	0	1	0	0	1	1	0	0 2 6	1	1	(A) = (M(DP)) ?
	SEA n	0	0	0	0	1	0	0	1	0	1	0 2 5	2	2	(A) = n ? n = 0 to 15
		0	0	0	1	1	1	n	n	n	n	0 7 n			

Note : p is 0 to 15 for M34512M2, p is 0 to 31 for M34512M4.

MITSUBISHI ELECTRIC

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)

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
Branch operation	B a	0	1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 8 a +a	1	1	$(PC_L) \leftarrow a_6$–a_0
	BL p, a	0	0	1	1	1	p_4	p_3	p_2	p_1	p_0	0 E p +p	2	2	$(PC_H) \leftarrow p$ $(PC_L) \leftarrow a_6$–a_0 (Note)
		1	0	p_5	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2 p a +a			
	BLA p	0	0	0	0	0	1	0	0	0	0	0 1 0	2	2	$(PC_H) \leftarrow p$ $(PC_L) \leftarrow (DR_2$–DR_0, A_3–$A_0)$ (Note)
		1	0	p_5	p_4	0	0	p_3	p_2	p_1	p_0	2 p p			
Subroutine operation	BM a	0	1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 a a	1	1	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow 2$ $(PC_L) \leftarrow a_6$–a_0
	BML p, a	0	0	1	1	0	p_4	p_3	p_2	p_1	p_0	0 C p +p	2	2	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow p$
		1	0	p_5	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2 p a +a			$(PC_L) \leftarrow a_6$–a_0 (Note)
	BMLA p	0	0	0	0	1	1	0	0	0	0	0 3 0	2	2	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow p$
		1	0	p_5	p_4	0	0	p_3	p_2	p_1	p_0	2 p p			$(PC_L) \leftarrow (DR_2$–DR_0, A_3–$A_0)$ (Note)
Return operation	RTI	0	0	0	1	0	0	0	1	1	0	0 4 6	1	1	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$
	RT	0	0	0	1	0	0	0	1	0	0	0 4 4	1	2	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$
	RTS	0	0	0	1	0	0	0	1	0	1	0 4 5	1	2	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$
Interrupt operation	DI	0	0	0	0	0	0	0	1	0	0	0 0 4	1	1	$(INTE) \leftarrow 0$
	EI	0	0	0	0	0	0	0	1	0	1	0 0 5	1	1	$(INTE) \leftarrow 1$
	SNZ0	0	0	0	0	1	1	1	0	0	0	0 3 8	1	1	(EXF0) = 1 ? After skipping $(EXF0) \leftarrow 0$
	SNZ1	0	0	0	0	1	1	1	0	0	1	0 3 9	1	1	(EXF1) = 1 ? After skipping $(EXF1) \leftarrow 0$

Note : p is 0 to 15 for M34512M2, p is 0 to 31 for M34512M4.

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 (0) to the EXF0 flag.
(EXF1) = 1	–	Skips the next instruction when the contents of EXF1 flag is “1.” After skipping, clears (0) to the EXF1 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			
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	$I2_2$ = 1 : (INT1) = "H" ?
															$I2_2$ = 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)
	TAB1	1	0	0	1	1	1	0	0	0	0	2 7 0	1	1	(B) ← ($T1_7$–$T1_4$) (A) ← ($T1_3$–$T1_0$)
	T1AB	1	0	0	0	1	1	0	0	0	0	2 3 0	1	1	($R1_7$–$R1_4$) ← (B) ($T1_7$–$T1_4$) ← (B) ($R1_3$–$R1_0$) ← (A) ($T1_3$–$T1_0$) ← (A)
	TAB2	1	0	0	1	1	1	0	0	0	1	2 7 1	1	1	(B) ← ($T2_7$–$T2_4$) (A) ← ($T2_3$–$T2_0$)
	T2AB	1	0	0	0	1	1	0	0	0	1	2 3 1	1	1	($R2_7$–$R2_4$) ← (B) ($T2_7$–$T2_4$) ← (B) ($R2_3$–$R2_0$) ← (A) ($T2_3$–$T2_0$) ← (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, $I2_2$ = 1	–	When bit 2 ($I2_2$) of register I2 is "1" : Skips the next instruction when the level of INT1 pin is "H."
(INT1) = "L" However, $I2_2$ = 0	–	When bit 2 ($I2_2$) 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 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.

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			
Timer operation	TAB3	1	0	0	1	1	1	0	0	1	0	2 7 2	1	1	(B) ← ($T3_7$–$T3_4$) (A) ← ($T3_3$–$T3_0$)
	T3AB	1	0	0	0	1	1	0	0	1	0	2 3 2	1	1	($R3_7$–$R3_4$) ← (B) ($T3_7$–$T3_4$) ← (B) ($R3_3$–$R3_0$) ← (A) ($T3_3$–$T3_0$) ← (A)
	TAB4	1	0	0	1	1	1	0	0	1	1	2 7 3	1	1	(B) ← ($T4_7$–$T4_4$) (A) ← ($T4_3$–$T4_0$)
	T4AB	1	0	0	0	1	1	0	0	1	1	2 3 3	1	1	($R4_7$–$R4_4$) ← (B) ($T4_7$–$T4_4$) ← (B) ($R4_3$–$R4_0$) ← (A) ($T4_3$–$T4_0$) ← (A)
	TR1AB	1	0	0	0	1	1	1	1	1	1	2 3 F	1	1	($R1_7$–$R1_4$) ← (B) ($R1_3$–$R1_0$) ← (A)
	TR3AB	1	0	0	0	1	1	1	0	1	1	2 3 B	1	1	($R3_7$–$R3_4$) ← (B) ($R3_3$–$R3_0$) ← (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 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 1 reload register.
–	–	Transfers the contents of registers A and B to timer 3 reload register.
(T1F) = 1	–	Skips the next instruction when the contents of T1F flag is “1.” After skipping, clears (0) to T1F flag.
(T2F) =1	–	Skips the next instruction when the contents of T2F flag is “1.” After skipping, clears (0) to T2F flag.
(T3F) = 1	–	Skips the next instruction when the contents of T3F flag is “1.” After skipping, clears (0) to T3F flag.
(T4F) = 1	–	Skips the next instruction when the contents of T4F flag is “1.” After skipping, clears (0) to T4F flag.

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
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_2–A_0) ← ($P2_2$–$P2_0$) (A_3) ← 0
	CLD	0	0	0	0	0	1	0	0	0	1	0 1 1	1	1	(D) ← 1
	RD	0	0	0	0	0	1	0	1	0	0	0 1 4	1	1	(D(Y)) ← 0 (Y) = 0 to 7
	SD	0	0	0	0	0	1	0	1	0	1	0 1 5	1	1	(D(Y)) ← 1 (Y) = 0 to 7
	SZD	0	0	0	0	1	0	0	1	0	0	0 2 4	2	2	(D(Y)) = 0 ? (Y) = 0 to 7
		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.
–	–	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 7	–	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	2 7 9	1	1	(A) ← (AD_5–AD_2) (B) ← (AD_9–AD_6) However, in the comparator mode, (A) ← (AD_3–AD_0) (B) ← (AD_7–AD_4)
	TALA	1	0	0	1	0	0	1	0	0	1	2 4 9	1	1	(A) ← (AD_1, AD_0, 0, 0)
	TADAB	1	0	0	0	1	1	1	0	0	1	2 3 9	1	1	(AD_3–AD_0) ← (A) (AD_7–AD_4) ← (B)
	TAQ1	1	0	0	1	0	0	0	1	0	0	2 4 4	1	1	(A) ← (Q1)
	TQ1A	1	0	0	0	0	0	0	1	0	0	2 0 4	1	1	(Q1) ← (A)
	ADST	1	0	1	0	0	1	1	1	1	1	2 9 F	1	1	(ADF) ← 0 A-D conversion starting
	SNZAD	1	0	1	0	0	0	0	1	1	1	2 8 7	1	1	(ADF) = 1 ? After skipping (ADF) ← 0
	TAQ2	1	0	0	1	0	0	0	1	0	1	2 4 5	1	1	(A) ← (Q2)
	TQ2A	1	0	0	0	0	0	0	1	0	1	2 0 5	1	1	(Q2) ← (A)
Other operation	NOP	0	0	0	0	0	0	0	0	0	0	0 0 0	1	1	(PC) ← (PC) + 1
	POF	0	0	0	0	0	0	0	0	1	0	0 0 2	1	1	RAM back-up
	EPOF	0	0	0	1	0	1	1	0	1	1	0 5 B	1	1	POF instruction valid
	SNZP	0	0	0	0	0	0	0	0	1	1	0 0 3	1	1	(P) = 1 ?
	WRST	1	0	1	0	1	0	0	0	0	0	2 A 0	1	1	(WDF1) ← 0 (WEF) ← 1
	TAMR	1	0	0	1	0	1	0	0	1	0	2 5 2	1	1	(A) ← (MR)
	TMRA	1	0	0	0	0	1	0	1	1	0	2 1 6	1	1	(MR) ← (A)

Skip condition	Carry flag CY	Datailed description
–	–	Transfers the high-order 8 bits of the contents of register AD to registers A and B.
–	–	Transfers the low-order 2 bits of the contents of register AD to the high-order 2 bits of the contents of register A. Simultaneously, the low-order 2 bits of the contents of the register A is “0.”
–	–	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 valid 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 WDF1.
–	–	Transfers the contents of the clock control register MR to register A.
–	–	Transfers the contents of register A to the clock control register MR.

ABSOLUTE MAXIMUM RAINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{DD}	Supply voltage		–0.3 to 7.0	V
V_I	Input voltage P0, P1, P2, INT0, INT1, $\overline{RESET}$, X_{IN}		–0.3 to V_{DD}+0.3	V
V_I	Input voltage D_0–D_7		–0.3 to 13	V
V_I	Input voltage A_{IN0}–A_{IN3}		–0.3 to V_{DD}+0.3	V
V_O	Output voltage P0, P1, $\overline{RESET}$	Output transistors in cut-off state	–0.3 to V_{DD}+0.3	V
V_O	Output voltage D_0–D_7		–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	300	mW
Topr	Operating temperature range		–20 to 85	°C
Tstg	Storage temperature range		–40 to 125	°C

RECOMMENDED OPERATING CONDITIONS (Ta = –20 °C to 85 °C, V_{DD} = 4.0 V to 5.5 V, unless otherwise noted)

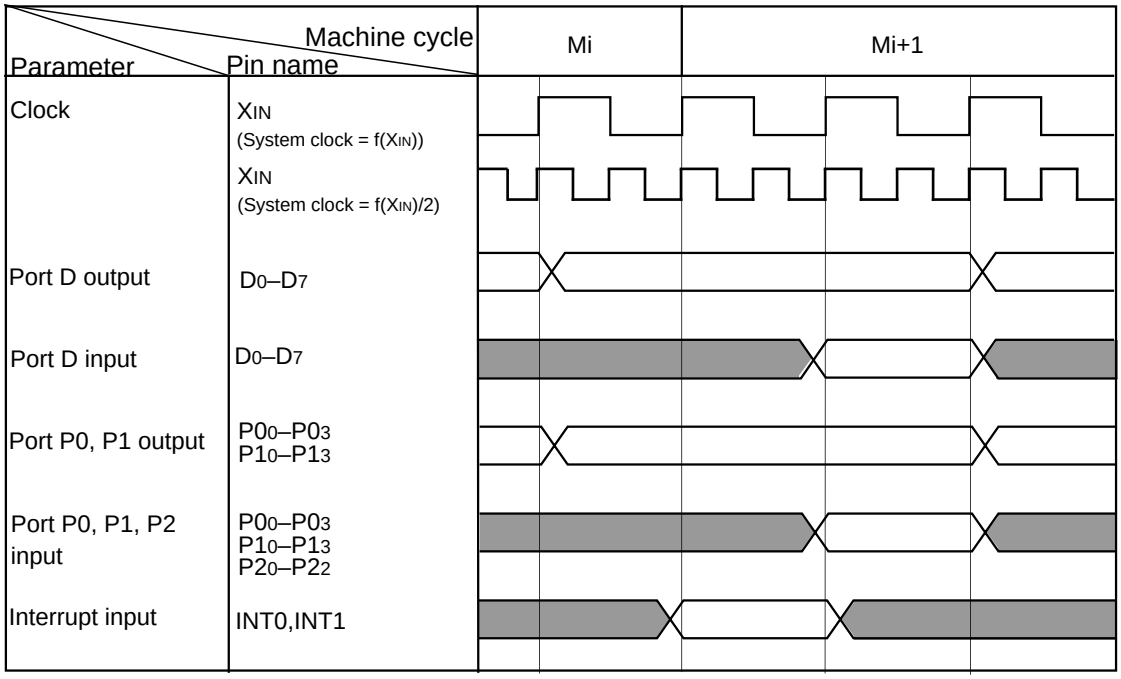
Symbol	Parameter	Conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{DD}	Supply voltage	$f(X_{IN}) \leq 4.2$ MHz	4.0		5.5	V
V_{RAM}	RAM back-up voltage (at RAM back-up mode)		1.8			
V_{SS}	Supply voltage			0		
V_{IH}	“H” level input voltage P0, P1, P2, X_{IN}		0.8V_{DD}		V_{DD}	
V_{IH}	“H” level input voltage D_0–D_7		0.8V_{DD}		12	
V_{IH}	“H” level input voltage $\overline{RESET}$		0.85V_{DD}		V_{DD}	
V_{IH}	“H” level input voltage S_{IN}, S_{CK}, INT0, INT1		0.85V_{DD}		V_{DD}	
V_{IL}	“L” level input voltage P0, P1, P2, D_0–D_7, X_{IN}		0		0.2V_{DD}	
V_{IL}	“L” level input voltage $\overline{RESET}$		0		0.3V_{DD}	
V_{IL}	“L” level input voltage S_{IN}, S_{CK}, INT0, INT1		0		0.15V_{DD}	
I_{OL}(peak)	“L” level peak output current $\overline{RESET}$	V_{DD} = 5.0 V			10	mA
I_{OL}(peak)	“L” level peak output current D_6, D_7	V_{DD} = 5.0 V			40	
I_{OL}(peak)	“L” level peak output current D_0–D_5	V_{DD} = 5.0 V			24	
I_{OL}(peak)	“L” level peak output current P0, P1, S_{CK}, S_{OUT}	V_{DD} = 5.0 V			24	
I_{OL}(avg)	“L” level average output current $\overline{RESET}$ (Note)	V_{DD} = 5.0 V			5	
I_{OL}(avg)	“L” level average output current D_6, D_7 (Note)	V_{DD} = 5.0 V			30	
I_{OL}(avg)	“L” level average output current D_0–D_5 (Note)	V_{DD} = 5.0 V			15	
I_{OL}(avg)	“L” level average output current P0, P1, S_{CK}, S_{OUT} (Note)	V_{DD} = 5.0 V			12	
ΣI_{OL}(avg)	“L” level total average current D, $\overline{RESET}$, S_{CK}, S_{OUT}				80	
	“L” level total average current P0, P1				80	
f(X_{IN})	Oscillation frequency (at ceramic resonance)	V_{DD} = 4.0 V to 5.5 V			4.2	MHz
	Oscillation frequency (at external clock input)	V_{DD} = 4.0 V to 5.5 V			3.0	
tw(S_{CK})	Serial I/O external clock cycle (pulse width of “H” and “L“)	V_{DD} = 4.0 V to 5.5 V	750			ns

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

ELECTRICAL CHARACTERISTICS (Ta = –20 °C to 85 °C, V_{DD} = 4.0 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_{OL}	"L" level output voltage $\overline{RESET}$		V_{DD} = 5 V	I_{OL} = 5 mA			2	V
V_{OL}	"L" level output voltage D_6, D_7		V_{DD} = 5 V	I_{OL} = 30 mA			2	V
				I_{OL} = 10 mA			0.9	
V_{OL}	"L" level output voltage D_0–D_5		V_{DD} = 5 V	I_{OL} = 15 mA			2	V
I_{IH}	"H" level input current P0, P1, P2, $\overline{RESET}$		V_I = V_{DD}				1	μA
I_{IH}	"H" level input current D_0–D_7		V_I = 12 V				1	μA
I_{IL}	"L" level input current P0, P1, P2, $\overline{RESET}$		V_I = 0 V No pull-up of ports P0 and P1		–1			μA
I_{IL}	"L" level input current D_0–D_7		V_I = 0 V		–1			μA
I_{DD}	Supply current	at active mode	V_{DD} = 5 V	$f(X_{IN})$ = 4.0 MHz		1.8	5.5	mA
			Middle-speed mode	$f(X_{IN})$ = 400 kHz		0.5	1.5	
			V_{DD} = 5 V	$f(X_{IN})$ = 4.0 MHz		3.0	9.0	
			High-speed mode	$f(X_{IN})$ = 400 kHz		0.6	1.8	
		at RAM back-up mode	Ta = 25 °C			0.1	1	μA
			V_{DD} = 5 V				10	
R_{PU}	Pull-up resistor value		V_{DD} = 5 V	V_I = 0 V	20	50	125	kΩ
$V_{T+} – V_{T–}$	Hysteresis INT0, INT1, S_{IN}, S_{CK}		V_{DD} = 5 V			0.3		V
$V_{T+} – V_{T–}$	Hysteresis $\overline{RESET}$		V_{DD} = 5 V			1.5		V

BASIC TIMING DIAGRAM



A-D CONVERTER RECOMMENDED OPERATING CONDITIONS

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

Symbol	Parameter	Conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{DD}	Supply voltage		4.0		5.5	V
V_{IA}	Analog input voltage		0		V_{DD}	V
f(X_{IN})	Oscillation frequency	Middle-speed mode	0.8			MHz
		High-speed mode	0.4			MHz

A-D CONVERTER CHARACTERISTICS (Ta = –20 °C to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
–	Resolution				10	bits
–	Linearity error	Ta = –25 °C to 85 ° C, V_{DD} = 4.0 V to 5.5 V			±2	LSB
–	Differential non-linearity error	Ta = –25 °C to 85 ° C, V_{DD} = 4.0 V to 5.5 V			±0.9	LSB
V_{0T}	Zero transition voltage	V_{DD} = 5.12 V	0	5	20	mV
V_{FST}	Full-scale transition voltage	V_{DD} = 5.12 V	5105	5115	5125	mV
I_{ADD}	A–D operating current	V_{DD} = 5.0 V, f(X_{IN}) = 0.4 MHz to 4.0 MHz		0.7	2.0	mA
T_{CONV}	A-D conversion time	f(X_{IN}) = 4.0 MHz, Middle-speed mode			93.0	µs
		f(X_{IN}) = 4.0 MHz, High-speed mode			46.5	
–	Comparator resolution	Comparator mode			8	bits
–	Comparator error (Note)	V_{DD} = 5.12 V			±20	mV
–	Comparator comparison time	f(X_{IN}) = 4.0 MHz, Middle-speed mode			12	µs
		f(X_{IN}) = 4.0 MHz, High-speed mode			6	

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

Logic value of comparison voltage V_{ref}

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

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

GZZ-SH52-47B <81A0>

Mask ROM number	

4500 SERIES MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M34512M2-XXXFP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Please fill in all items marked ✻.

✻ Customer	Company name	TEL ( )	
	Date issued	Date:	

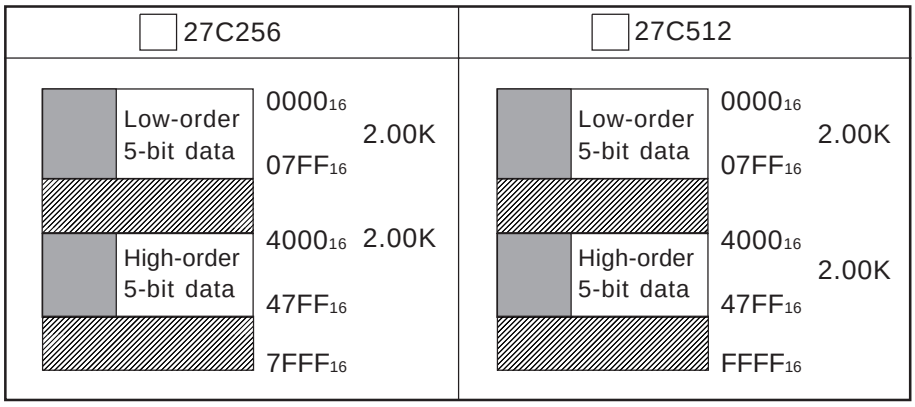
Issuance signature	Responsible officer	Supervisor

✻ 1. Confirmation

Specify the type of EPROMs submitted.
Three sets of EPROMs are required for each pattern (check in the approximate box).
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM area ☐☐☐☐ (hexadecimal notation)

EPROM Type:



Set "FF$_{16}$" in the shaded area.
Set "111$_2$" in the area ▒ of low-order and high-order 5-bit data.

✻ 2. Mark Specification

Mark specification must be submitted using the correct form for the type of package being ordered. Fill out the approximate Mark Specification Form (32P6B-A for M34512M2-XXXFP) and attach to the Mask ROM Order Confirmation Form.

✻ 3. Comments

GZZ-SH52-46B <81A0>

Mask ROM number	

4500 SERIES MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M34512M4-XXXFP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Issuance signature	Responsible officer	Supervisor

Please fill in all items marked ✻.

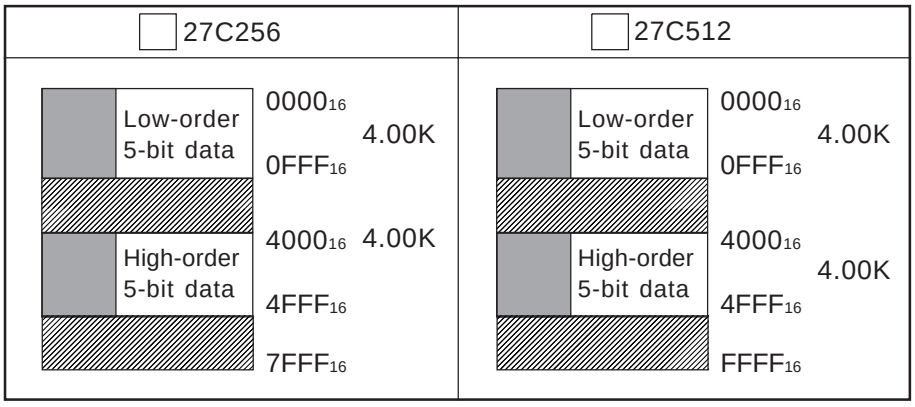
✻ Customer	Company name	TEL ( )	
	Date issued	Date:	

✻ 1. Confirmation

Specify the type of EPROMs submitted.
Three sets of EPROMs are required for each pattern (check in the approximate box).
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM area ☐☐☐☐ (hexadecimal notation)

EPROM Type:



Set "FF_{16}" in the shaded area.
Set "111_2" in the area ▒ of low-order and high-order 5-bit data.

✻ 2. Mark Specification

Mark specification must be submitted using the correct form for the type of package being ordered. Fill out the approximate Mark Specification Form (32P6B-A for M34512M4-XXXFP) and attach to the Mask ROM Order Confirmation Form.

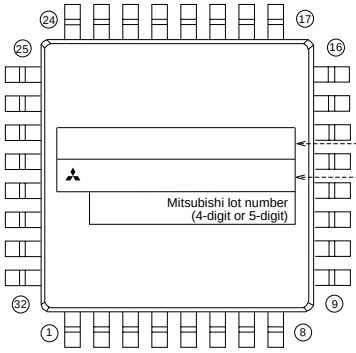
✻ 3. Comments

32P6B (32-PIN LQFP) MARK SPECIFICATION FORM

Mitsubishi IC catalog name ☐

Please choose one of the marking types below (A, B), and enter the Mitsubishi catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark

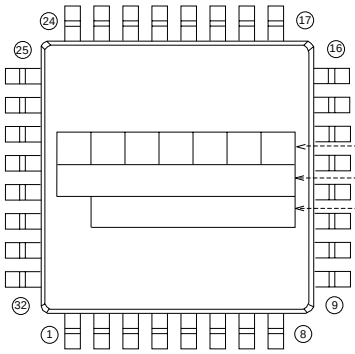


Mitsubishi IC catalog name
Mitsubishi IC catalog name

Note: If the Mitsubishi logo is not required, check the box on the right.

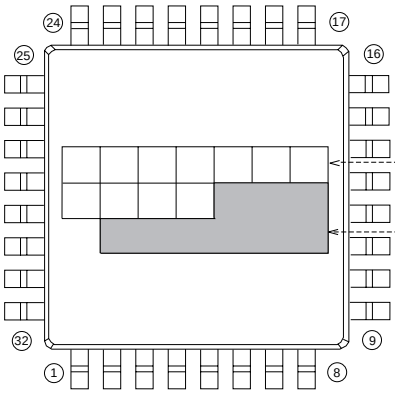
Mitsubishi logo is not required ☐

B. Customer's Parts Number + Mitsubishi catalog name



Customer's Parts Number
Note : The fonts and size of characters are standard Mitsubishi type.
Mitsubishi IC catalog name
Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
(The Mitsubishi logo is not marked.)
3 : Customer's Parts Number can be up to 7 characters : Only 0 ~ 9, A ~ Z, —(hyphen),. (periods),, (commas) are usable.

C. Customer's Parts Number



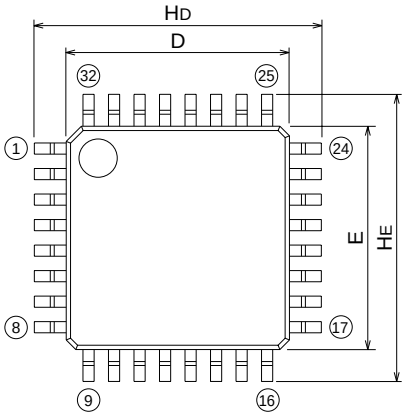
Customer's Parts Number
Note : The fonts and size of characters are standard Mitsubishi type.
Shaded area: Mitsubishi lot number (4-digit or 5-digit) and Mask ROM number (3-digit) are always marked, for products classification.
Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's Parts Number can be up to 11 characters : Only 0 ~ 9, A ~ Z, —(hyphen),. (periods),, (commas) are usable.

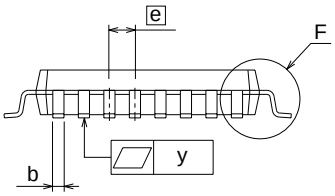
PACKAGE OUTLINE

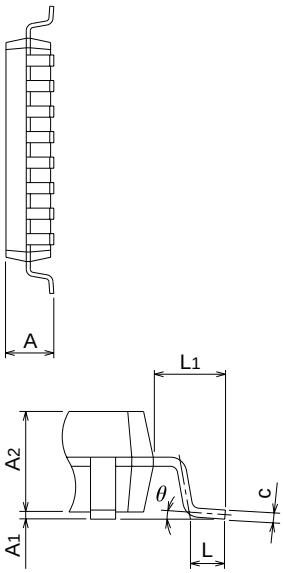
32P6B-A

Plastic 32pin 7×7mm body LQFP

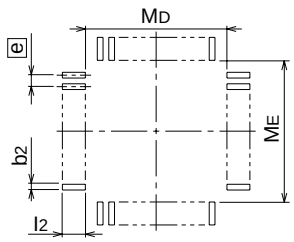
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP32-P-77-0.80	–		Alloy 42







Detail F



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	1.7
A_1	0	0.1	0.2
A_2	–	1.4	–
b	0.3	0.35	0.45
c	0.105	0.125	0.175
D	6.9	7.0	7.1
E	6.9	7.0	7.1
e	–	0.8	–
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
L	0.3	0.5	0.7
L_1	–	1.0	–
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.5	–
l_2	1.0	–	–
M_D	–	7.4	–
M_E	–	7.4	–

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

New publication, effective Dec. 1998.
Specifications subject to change without notice.

MITSUBISHI ELECTRIC

REVISION DESCRIPTION LIST	4512 GROUP DATA SHEET

Rev. No.	Revision Description	Rev. date
1.0	First Edition	981218