To all our customers

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

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

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

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

MITSUBISHI 4-BIT SINGLE-CHIP MICROCOMPUTER
4500 SERIES

4513/4514 Group

User's Manual

Renesas Technology Corp.

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.

Preface

This user's manual describes the hardware and instructions of Mitsubishi's 4513/4514 Group CMOS 4-bit microcomputer.
After reading this manual, the user should have a through knowledge of the functions and features of the 4513/4514 Group and should be able to fully utilize the product. The manual starts with specifications and ends with application examples.
In this manual, the 4514 Group is mainly described. The differences from the 4513 Group are described at the related points.

BEFORE USING THIS USER'S MANUAL

This user's manual consists of the following three chapters. Refer to the chapter appropriate to your conditions, such as hardware design or software development.

1. Organization

CHAPTER 1 HARDWARE
This chapter describes features of the microcomputer and operation of each peripheral function.

CHAPTER 2 APPLICATION
This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of related registers.

CHAPTER 3 APPENDIX
This chapter includes precautions for systems development using the microcomputer, the mask ROM confirmation forms (mask ROM version), and mark specification forms which are to be submitted when ordering.
Be sure to refer to this chapter because this chapter also includes necessary information for systems development.

Note: In this manual, the 4514 Group is mainly described. The differences from the 4513 Group are described at the related points.

Table of contents

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

List of figures

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

List of tables

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

CHAPTER 1
HARDWARE

DESCRIPTION
FEATURES
APPLICATION
PIN CONFIGURATION
BLOCK DIAGRAM
PERFORMANCE OVERVIEW
PIN DESCRIPTION
FUNCTION BLOCK OPERATIONS
ROM ORDERING METHOD
LIST OF PRECAUTIONS
SYMBOL
LIST OF INSTRUCTION FUNCTION
INSTRUCTION CODE TABLE
MACHINE INSTRUCTIONS
CONTROL REGISTERS
BUILT-IN PROM VERSION

DESCRIPTION

The 4513/4514 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 4513/4514 Group include variations of the built-in memory type and package 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
  - Middle-speed mode
    ...... 2.5 V to 5.5 V (at 4.2 MHz oscillation frequency, for Mask ROM version and One Time PROM version)
    ...... 2.0 V to 5.5 V (at 3.0 MHz oscillation frequency, for Mask ROM version)
    (Operation voltage of A-D conversion: 2.7 V to 5.5 V)
  - High-speed mode
    ...... 4.0 V to 5.5 V (at 4.2 MHz oscillation frequency, for Mask ROM version and One Time PROM version)
    ...... 2.5 V to 5.5 V (at 2.0 MHz oscillation frequency, for Mask ROM version and One Time PROM version)
    ...... 2.0 V to 5.5 V (at 1.5 MHz oscillation frequency, for Mask ROM version)
    (Operation voltage of A-D conversion: 2.7 V to 5.5 V)
- 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
- Voltage comparator .... 2 circuits
- Watchdog timer .... 16 bits
- Voltage drop detection circuit
- Clock generating circuit (ceramic resonator)
- LED drive directly enabled (port D)

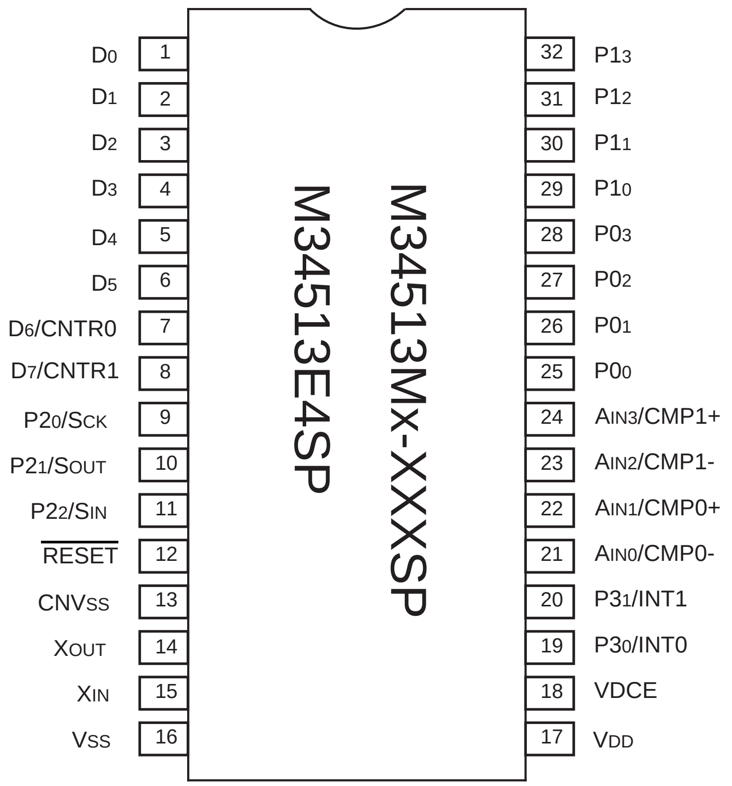
APPLICATION

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

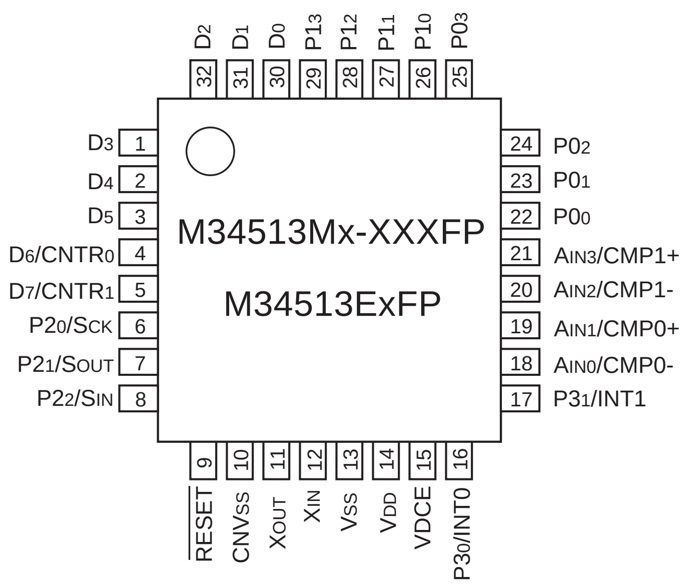
Product	ROM (PROM) size (✕ 10 bits)	RAM size (✕ 4 bits)	Package	ROM type
M34513M2-XXXSP/FP	2048 words	128 words	SP: 32P4B FP: 32P6B-A	Mask ROM
M34513M4-XXXSP/FP	4096 words	256 words	SP: 32P4B FP: 32P6B-A	Mask ROM
M34513E4SP/FP (Note)	4096 words	256 words	SP: 32P4B FP: 32P6B-A	One Time PROM
M34513M6-XXXFP	6144 words	384 words	32P6B-A	Mask ROM
M34513M8-XXXFP	8192 words	384 words	32P6B-A	Mask ROM
M34513E8FP (Note)	8192 words	384 words	32P6B-A	One Time PROM
M34514M6-XXXFP	6144 words	384 words	42P2R-A	Mask ROM
M34514M8-XXXFP	8192 words	384 words	42P2R-A	Mask ROM
M34514E8FP (Note)	8192 words	384 words	42P2R-A	One Time PROM

Note: shipped in blank

PIN CONFIGURATION (TOP VIEW) 4513 Group

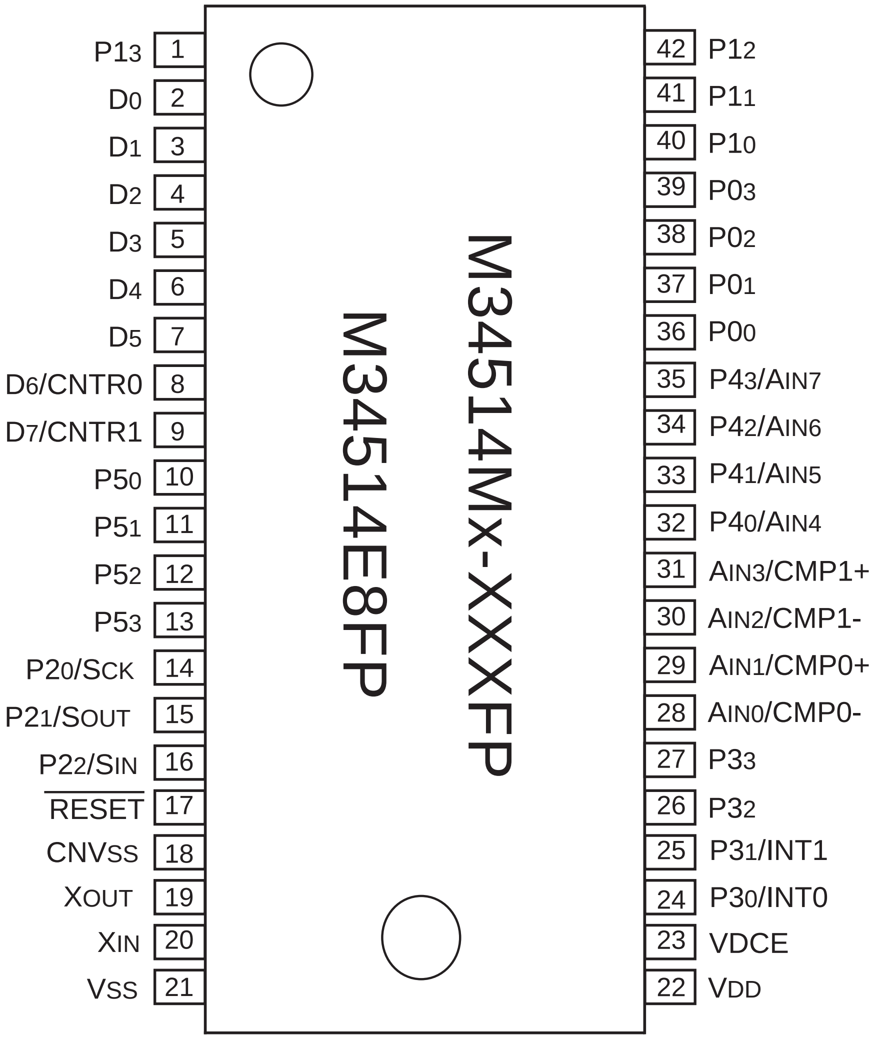


Outline 32P4B



Outline 32P6B-A

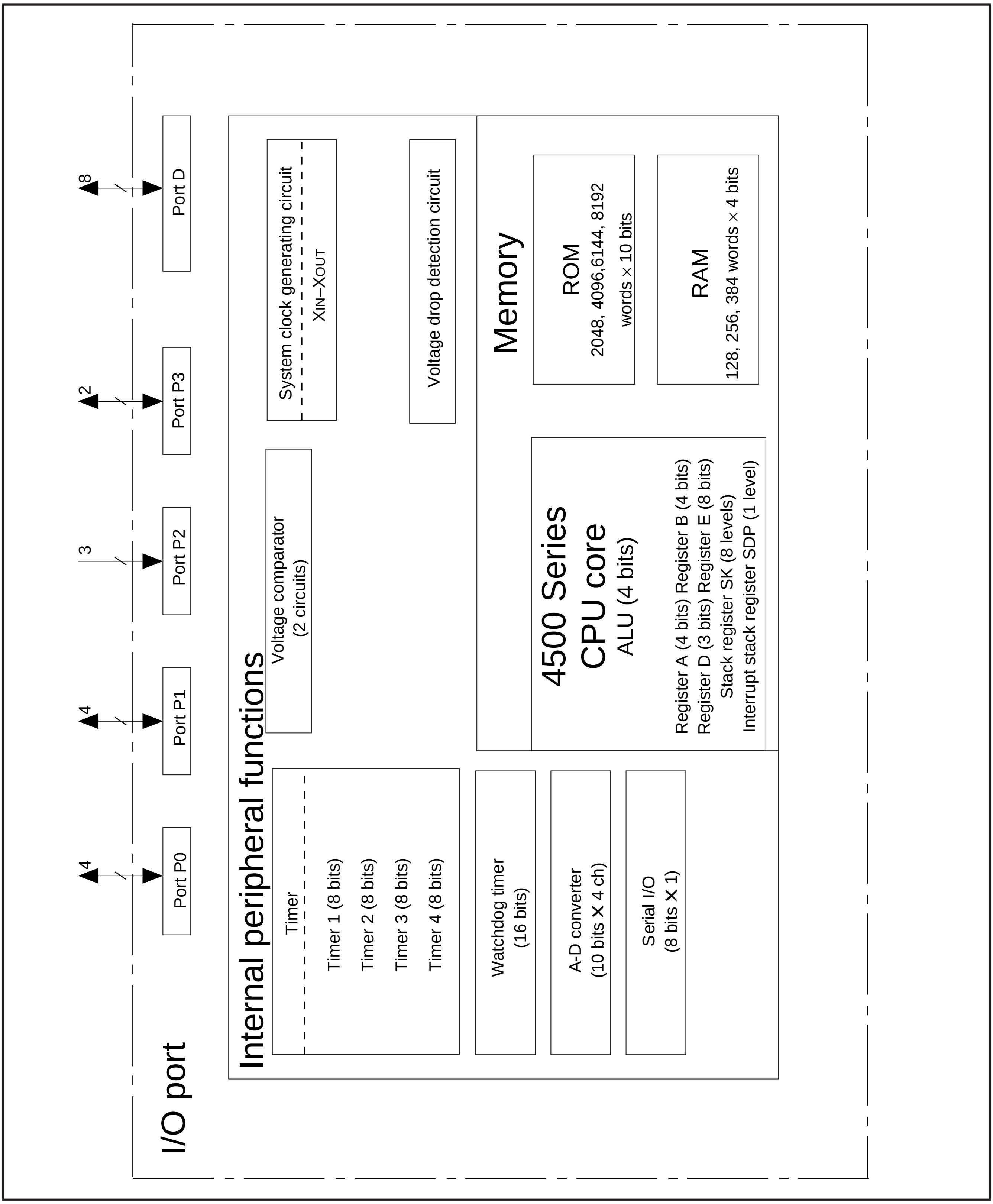
PIN CONFIGURATION (TOP VIEW) 4514 Group



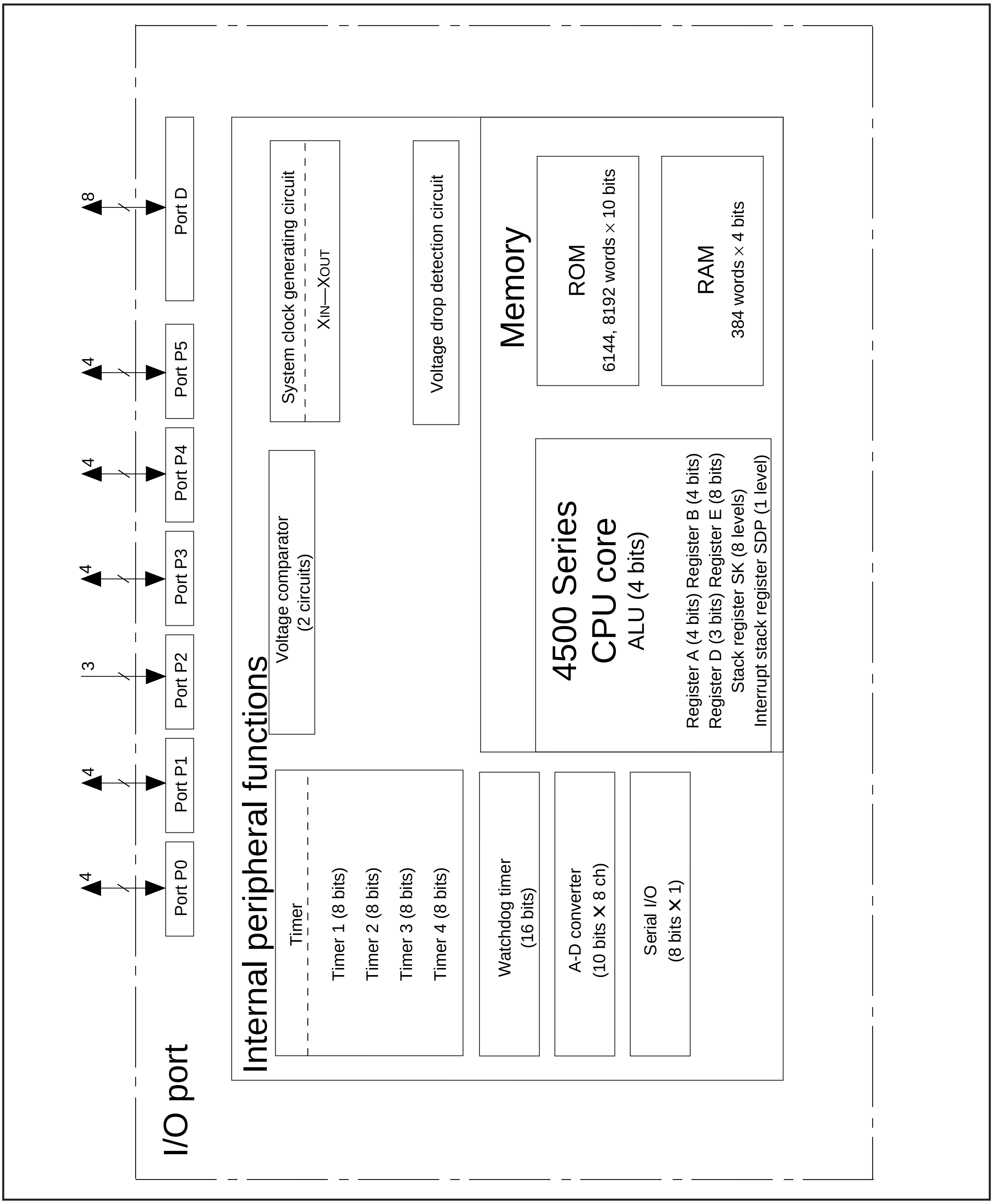
Outline 42P2R-A

BLOCK DIAGRAM

BLOCK DIAGRAM (4513 Group)



BLOCK DIAGRAM (4514 Group)



PERFORMANCE OVERVIEW

Parameter			Function
Number of basic instructions		4513 Group	123
		4514 Group	128
Minimum instruction execution time			0.75 μs (at 4.0 MHz oscillation frequency, in high-speed mode)
Memory sizes	ROM	M34513M2	2048 words × 10 bits
		M34513M4/E4	4096 words × 10 bits
		M34513M6	6144 words × 10 bits
		M34513M8/E8	8192 words × 10 bits
		M34514M6	6144 words × 10 bits
		M34514M8/E8	8192 words × 10 bits
	RAM	M34513M2	128 words × 4 bits
		M34513M4/E4	256 words × 4 bits
		M34513M6	384 words × 4 bits
		M34513M8/E8	384 words × 4 bits
		M34514M6	384 words × 4 bits
		M34514M8/E8	384 words × 4 bits
Input/Output ports	D_0–D_7	I/O (Input is examined by skip decision)	Eight independent I/O ports; ports D_6 and D_7 are also used as CNTR0 and CNTR1, respectively.
	$P0_0$–$P0_3$	I/O	4-bit I/O port; each pin is equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software.
	$P1_0$–$P1_3$	I/O	4-bit I/O port; each pin is equipped with a pull-up function and a key-on wakeup function. Both functions can be switched by software.
	$P2_0$–$P2_2$	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.
	$P3_0$–$P3_3$	I/O	4-bit I/O port (2-bit I/O port for the 4513 Group); ports $P3_0$ and $P3_1$ are also used as INT0 and INT1, respectively. The 4513 Group does not have ports $P3_2$, $P3_3$.
	$P4_0$–$P4_3$	I/O	4-bit I/O port; The 4513 Group does not have this port.
	$P5_0$–$P5_3$	I/O	4-bit I/O port with a direction register; The 4513 Group does not have this port.
	CNTR0	I/O	1-bit I/O; CNTR0 pin is also used as port D_6.
	CNTR1	I/O	1-bit I/O; CNTR1 pin is also used as port D_7.
	INT0	Input	1-bit input; INT0 pin is also used as port $P3_0$ and equipped with a key-on wakeup function.
	INT1	Input	1-bit input; INT1 pin is also used as port $P3_1$ and 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 is also used as an event counter.
	Timer 3		8-bit programmable timer with a reload register.
	Timer 4		8-bit programmable timer with a reload register is also used as an event counter.
A-D converter			10-bit wide, This is equipped with an 8-bit comparator function.
Voltage comparator			2 circuits (CMP0, CMP1)
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	4513 Group		32-pin plastic molded SDIP (32P4B)/LQFP(32P6B-A)
	4514 Group		42-pin plastic molded SSOP (42P2R-A)
Operating temperature range			–20 °C to 85 °C
Supply voltage			2.0 V to 5.5 V for Mask ROM version, 2.5 V to 5.5 V for One Time PROM version (Refer to the electrical characteristics because the supply voltage depends on the oscillation frequency.)
Power dissipation (typical value)	Active mode		1.8 mA (at V_{DD} = 5.0 V, 4.0 MHz oscillation frequency, in middle- speed mode, output transistors in the cut-off state)
			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.
VDCE	Voltage drop detection circuit enable	Input	VDCE pin is used to control the operation/stop of the voltage drop detection circuit. When “H” level is input to this pin, the circuit is operating. When “L” level is inpu to this pin, the circuit is stopped.
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 or system reset is performed by the voltage drop detection circuit, 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 (Input is examined by skip decision.)	I/O	Each pin of port D has an independent 1-bit wide I/O function. Each pin has an output latch. For input use, set the latch of the specified bit to “1.” The output structure is N-channel open-drain. Ports D_6 and D_7 are also used as CNTR0 and CNTR1, respectively.
$P0_0$–$P0_3$	I/O port P0	I/O	Each of ports P0 and P1 serves as a 4-bit I/O port, and it can be used as inputs when the output latch is set to “1.” The output structure is N-channel open-drain. Every pin of the ports has a key-on wakeup function and a pull-up function. Both functions can be switched by software.
$P1_0$–$P1_3$	I/O port P1	I/O	
$P2_0$–$P2_2$	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.
$P3_0$–$P3_3$	I/O port P3	I/O	4-bit I/O port (2-bit I/O port for the 4513 Group). For input use, set the latch of the specified bit to “1.” The output structure is N-channel open-drain. Ports $P3_0$ and $P3_1$ are also used as INT0 and INT1, respectively. The 4513 Group does not have ports $P3_2$, $P3_3$.
$P4_0$–$P4_3$	I/O port P4	I/O	4-bit I/O port. For input use, set the latch of the specified bit to “1.” The output structure is N-channel open-drain. Ports $P4_0$–$P4_3$ are also used as analog input pins A_{IN4}–A_{IN7}, respectively. The 4513 Group does not have port P4.
$P5_0$–$P5_3$	I/O port P5	I/O	4-bit I/O port. Each pin has a direction register and an independent 1-bit wide I/O function. For input use, set the direction register to “0.” For output use, set the direction regiser to “1.” The output structure is CMOS. The 4513 Group does not have port P5.
A_{IN0}–A_{IN7}	Analog input	Input	Analog input pins for A-D converter. A_{IN0}–A_{IN3} are also used as voltage comparator input pins and A_{IN4}–A_{IN7} are also used as port P4. The 4513 Group does not have A_{IN4}–A_{IN7}.
CNTR0	Timer input/output	I/O	CNTR0 pin has the function to input the clock for the timer 2 event counter, and to output the timer 1 underflow signal divided by 2. CNTR0 pin is also used as port D_6.
CNTR1	Timer input/output	I/O	CNTR1 pin has the function to input the clock for the timer 4 event counter, and to output the timer 3 underflow signal divided by 2. CNTR1 pin is also used as port D_7.
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. INT0, INT1 pins are also used as ports $P3_0$ and $P3_1$, respectively.
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 $P2_0$.
CMP0- CMP0+	Voltage comparator input	Input	CMP0-, CMP0+ pins are used as the voltage comparator input pin when the voltage comparator function is selected by software. CMP0-, CMP0+ pins are also used as A_{IN0} and A_{IN1}.
CMP1- CMP1+	Voltage comparator input	Input	CMP1-, CMP1+ pins are used as the voltage comparator input pin when the voltage comparator function is selected by software. CMP1-, CMP1+ pins are also used as A_{IN2} and A_{IN3}.

MULTIFUNCTION

Pin	Multifunction
D_6	CNTR0
D_7	CNTR1
$P2_0$	S_{CK}
$P2_1$	S_{OUT}
$P2_2$	S_{IN}
$P3_0$	INT0
$P3_1$	INT1

Pin	Multifunction
CNTR0	D_6
CNTR1	D_7
S_{CK}	$P2_0$
S_{OUT}	$P2_1$
S_{IN}	$P2_2$
INT0	$P3_0$
INT1	$P3_1$

Pin	Multifunction
A_{IN0}	CMP0-
A_{IN1}	CMP0+
A_{IN2}	CMP1-
A_{IN3}	CMP1+
$P4_0$	A_{IN4}
$P4_1$	A_{IN5}
$P4_2$	A_{IN6}
$P4_3$	A_{IN7}

Pin	Multifunction
CMP0-	A_{IN0}
CMP0+	A_{IN1}
CMP1-	A_{IN2}
CMP1+	A_{IN3}
A_{IN4}	$P4_0$
A_{IN5}	$P4_1$
A_{IN6}	$P4_2$
A_{IN7}	$P4_3$

Notes 1: Pins except above have just single function.
2: The input of D_6, D_7, $P2_0$–$P2_2$, CMP0-, CMP0+, CMP1-, CMP1+ and the input/output of $P3_0$, $P3_1$, $P4_0$–$P4_3$ can be used even when CNTR0, CNTR1, S_{CK}, S_{OUT}, S_{IN}, INT0, INT1, and A_{IN0}–A_{IN7} are selected.
3: The 4513 Group does not have $P4_0$/A_{IN4}–$P4_3$/A_{IN7}.

CONNECTIONS OF UNUSED PINS

Pin	Connection
X_{OUT}	Open (when using an external clock).
VDCE	Connect to V_{SS}.
D_0–D_5 D_6/CNTR0 D_7/CNTR1	Connect to V_{SS}, or set the output latch to “0” and open.
$P2_0$/S_{CK} $P2_1$/S_{OUT} $P2_2$/S_{IN}	Connect to V_{SS}.
$P3_0$/INT0 $P3_1$/INT1 $P3_2$, $P3_3$	Connect to V_{SS}, or set the output latch to “0” and open.
$P4_0$/A_{IN4}–$P4_3$/A_{IN7}	Connect to V_{SS}, or set the output latch to “0” and open.
$P5_0$–$P5_3$ (Note 1)	When the input mode is selected by software, pull-up to V_{DD} through a resistor or pull-down to V_{DD}. When selecting the output mode, open.
A_{IN0}/CMP0- A_{IN1}/CMP0+ A_{IN2}/CMP1- A_{IN3}/CMP1+	Connect to V_{SS}.
$P0_0$–$P0_3$	Open or connect to V_{SS} (Note 2)
$P1_0$–$P1_3$	Open or connect to V_{SS} (Note 2)

Notes 1: After system is released from reset, port P5 is in an input mode (direction register FR0 = 0000_2)
2: When the $P0_0$–$P0_3$ and $P1_0$–$P1_3$ are connected to V_{SS}, 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 V_{SS} 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 V_{SS} and V_{DD})

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

PORT FUNCTION

Port	Pin	Input Output	Output structure	I/O unit	Control instructions	Control registers	Remark
Port D	D_0–D_5 D_6/CNTR0 D_7/CNTR1	I/O (8)	N-channel open-drain	1	SD, RD SZD CLD	 W6	
Port P0	$P0_0$–$P0_3$	I/O (4)	N-channel open-drain	4	OP0A IAP0	PU0, K0	Built-in programmable pull-up functions Key-on wakeup functions (programmable)
Port P1	$P1_0$–$P1_3$	I/O (4)	N-channel open-drain	4	OP1A IAP1	PU0, K0	Built-in programmable pull-up functions Key-on wakeup functions (programmable)
Port P2	$P2_0$/SCK $P2_1$/SOUT $P2_2$/SIN	Input (3)		3	IAP2	J1	
Port P3 (Note 1)	$P3_0$/INT0 $P3_1$/INT1 $P3_2$, $P3_3$	I/O (4)	N-channel open-drain	4	OP3A IAP3	I1, I2	Built-in key-on wakeup function ($P3_0$/INT0, $P3_1$/INT1)
Port P4 (Note 2)	$P4_0$/A_{IN4} –$P4_3$/A_{IN7}	I/O (4)	N-channel open-drain	4	OP4A IAP4	Q2	
Port P5 (Note 2)	$P5_0$–$P5_3$	I/O (4)	CMOS	4	OP5A IAP5	FR0	

Notes 1: The 4513 Group does not have $P3_2$ and $P3_3$.
2: The 4513 Group does not have these ports.

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

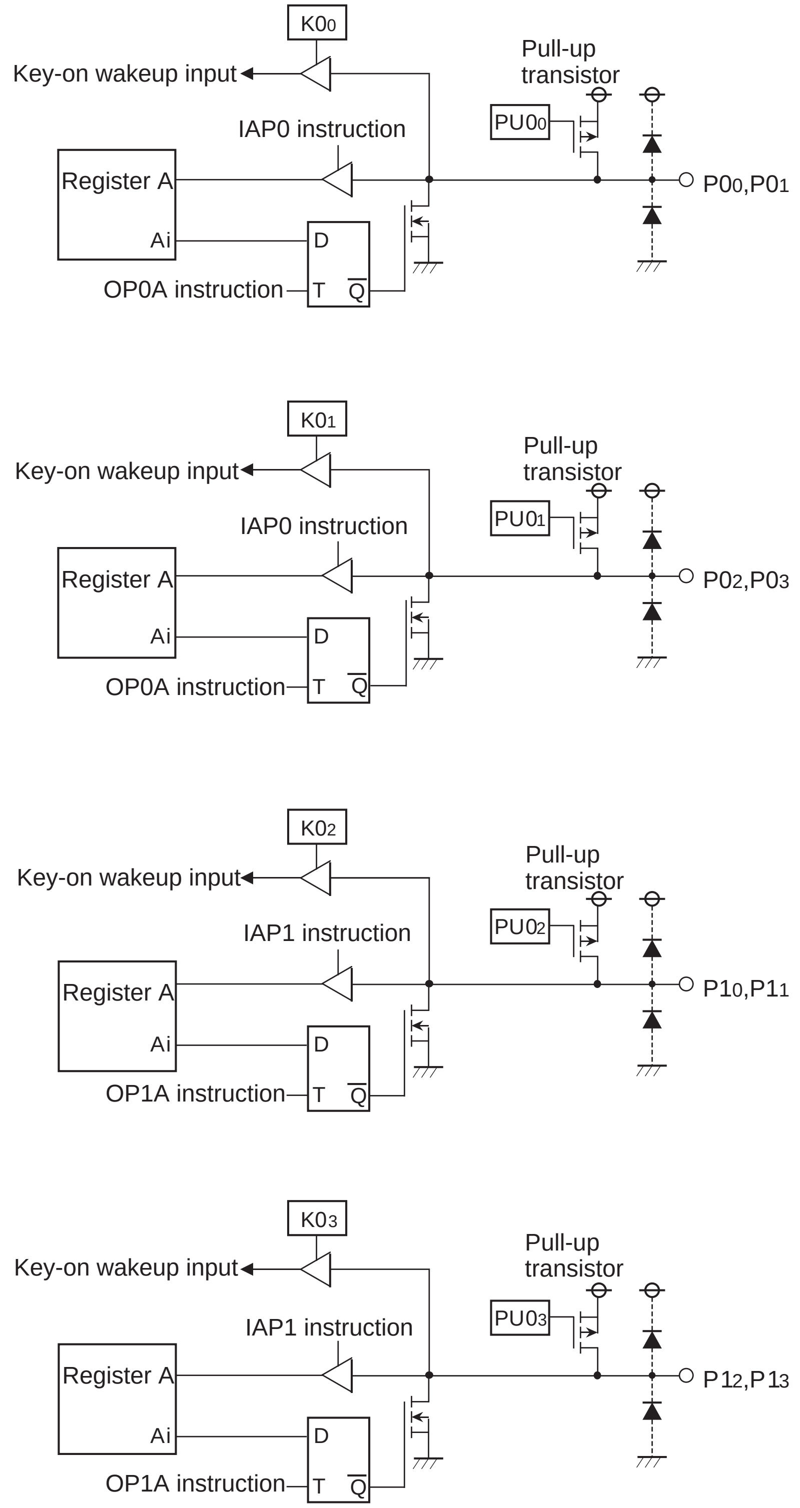
Register MR MR_3	System clock
0	f(X_{IN})
1	f(X_{IN})/2

Note: f(X_{IN})/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.

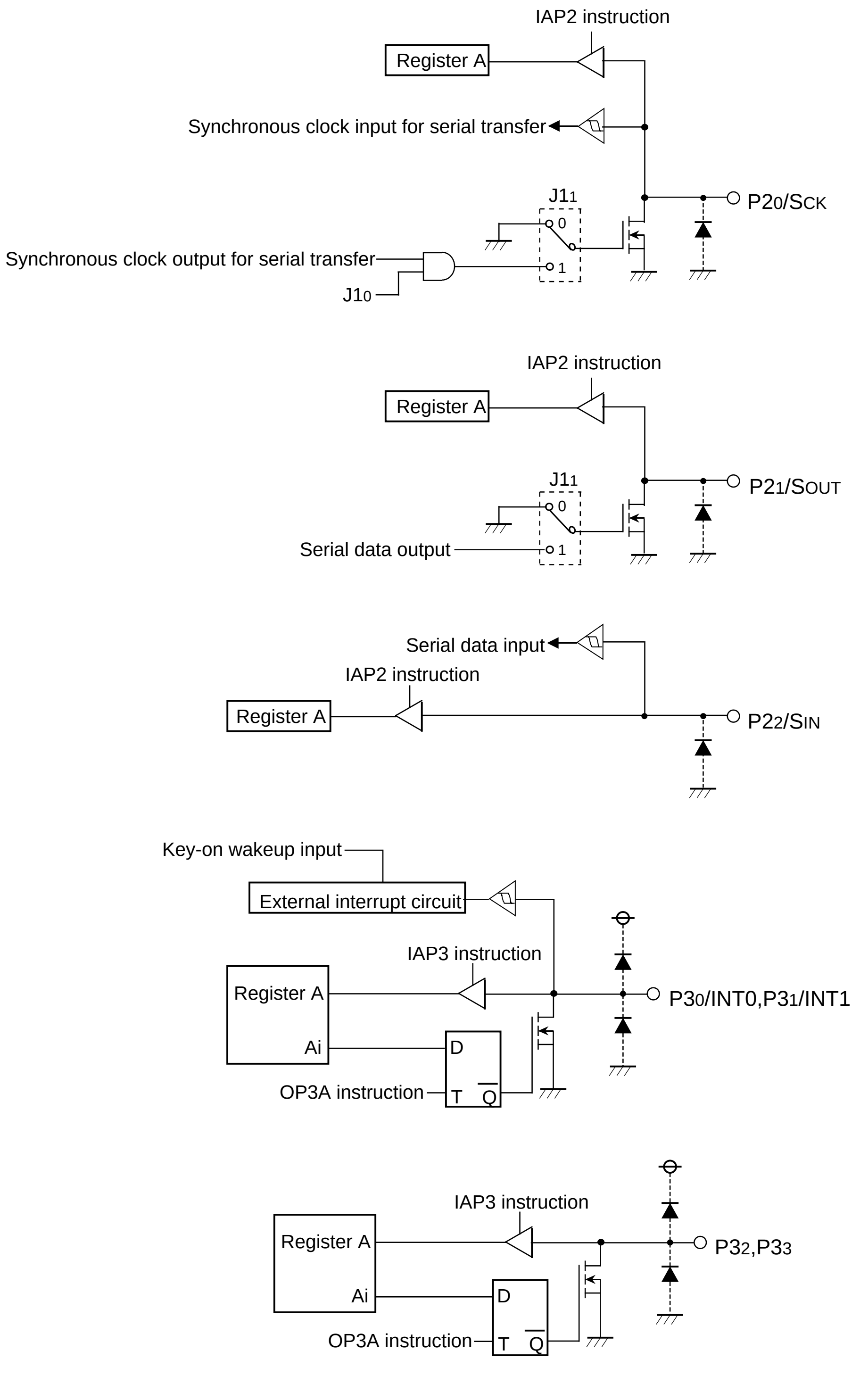
● Machine cycle
The machine cycle is the standard cycle required to execute the instruction.

PORT BLOCK DIAGRAMS



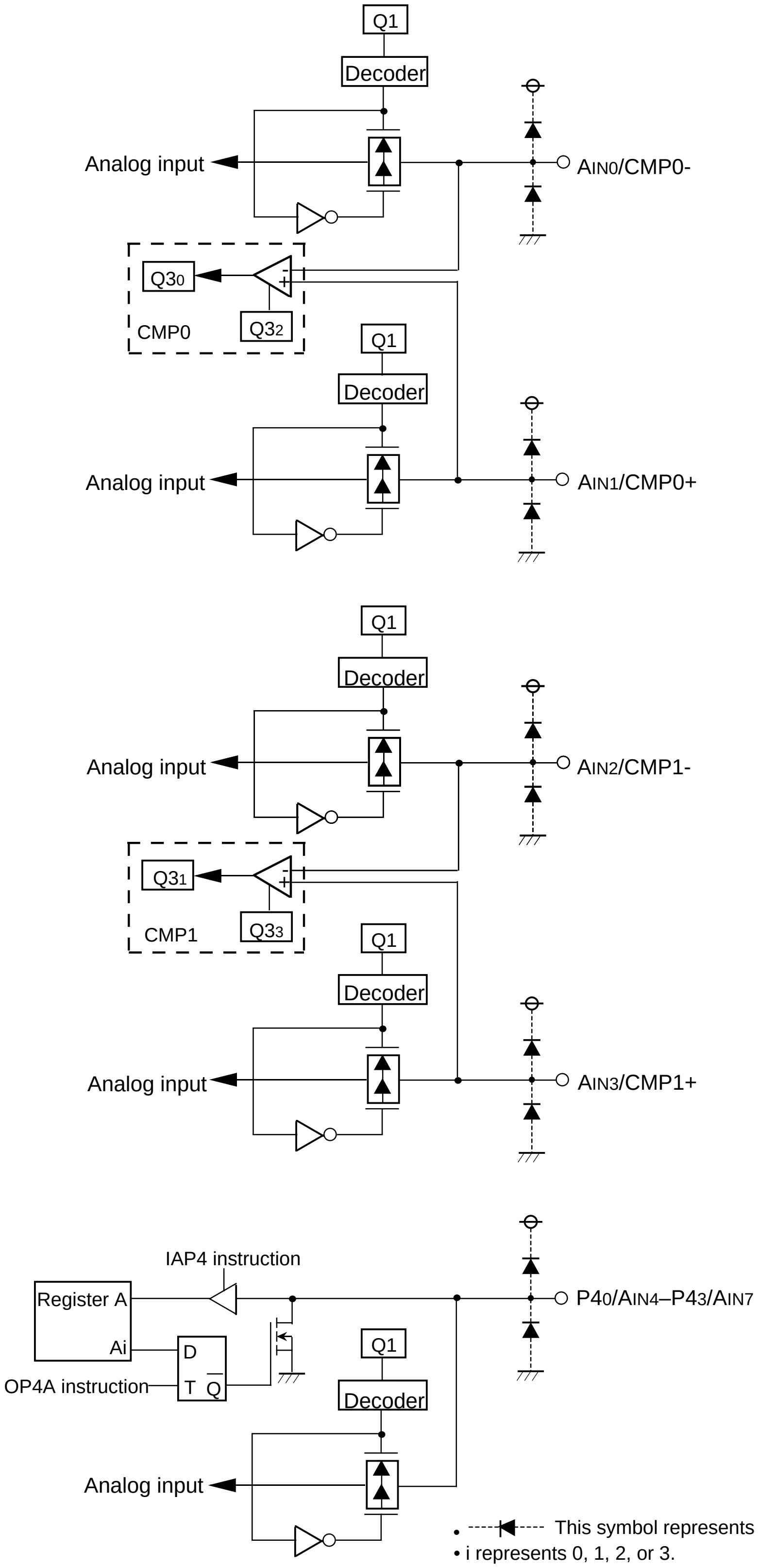
• This symbol represents a parasitic diode on the port.
• i represents 0, 1, 2, or 3.

PORT BLOCK DIAGRAMS (continued)



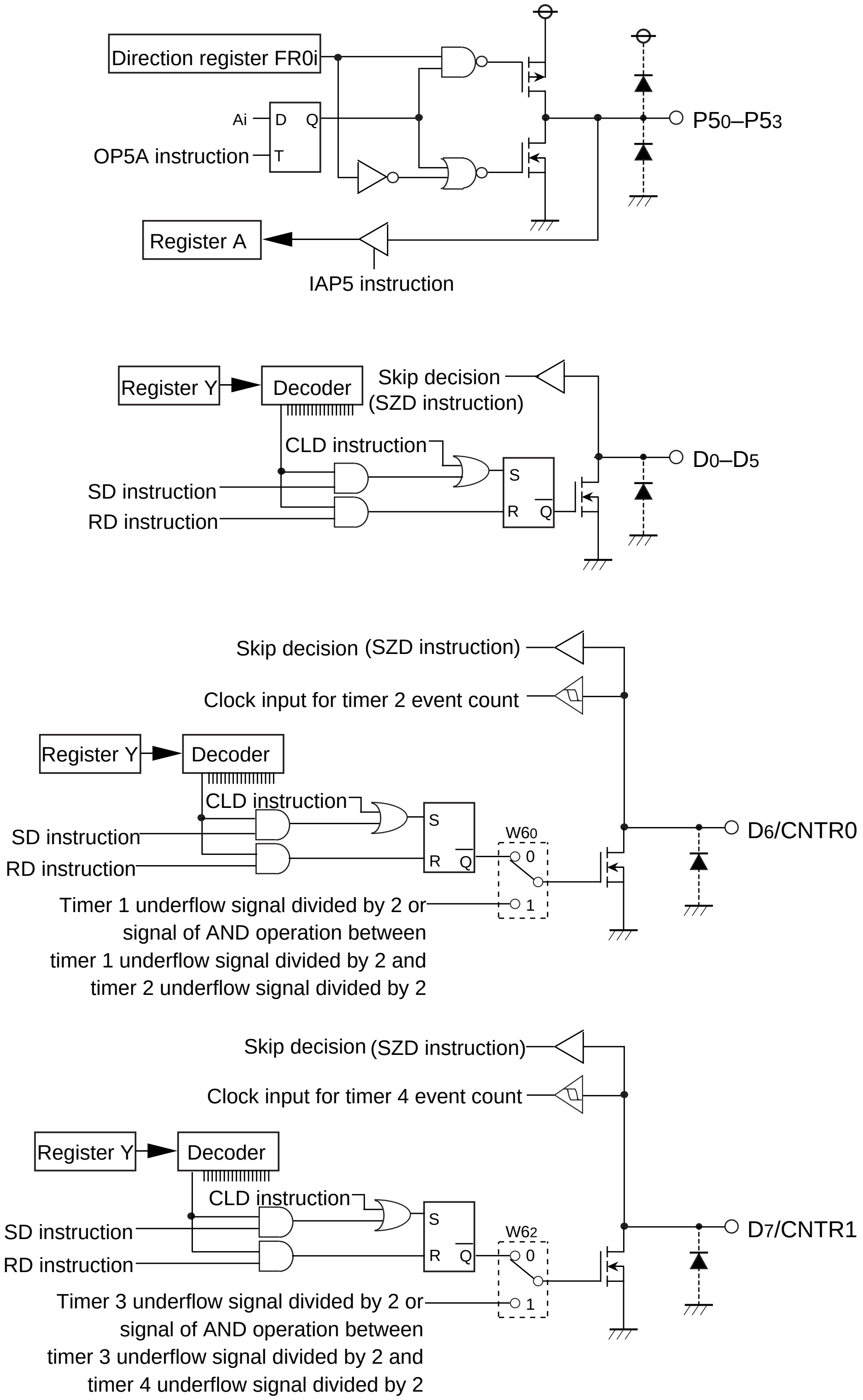
- This symbol represents a parasitic diode on the port.
- Applied potential to ports P20—P22 must be VDD.
- i represents 0, 1, 2, or 3.
- The 4513 Group does not have ports P32, P33.

PORT BLOCK DIAGRAMS (continued)

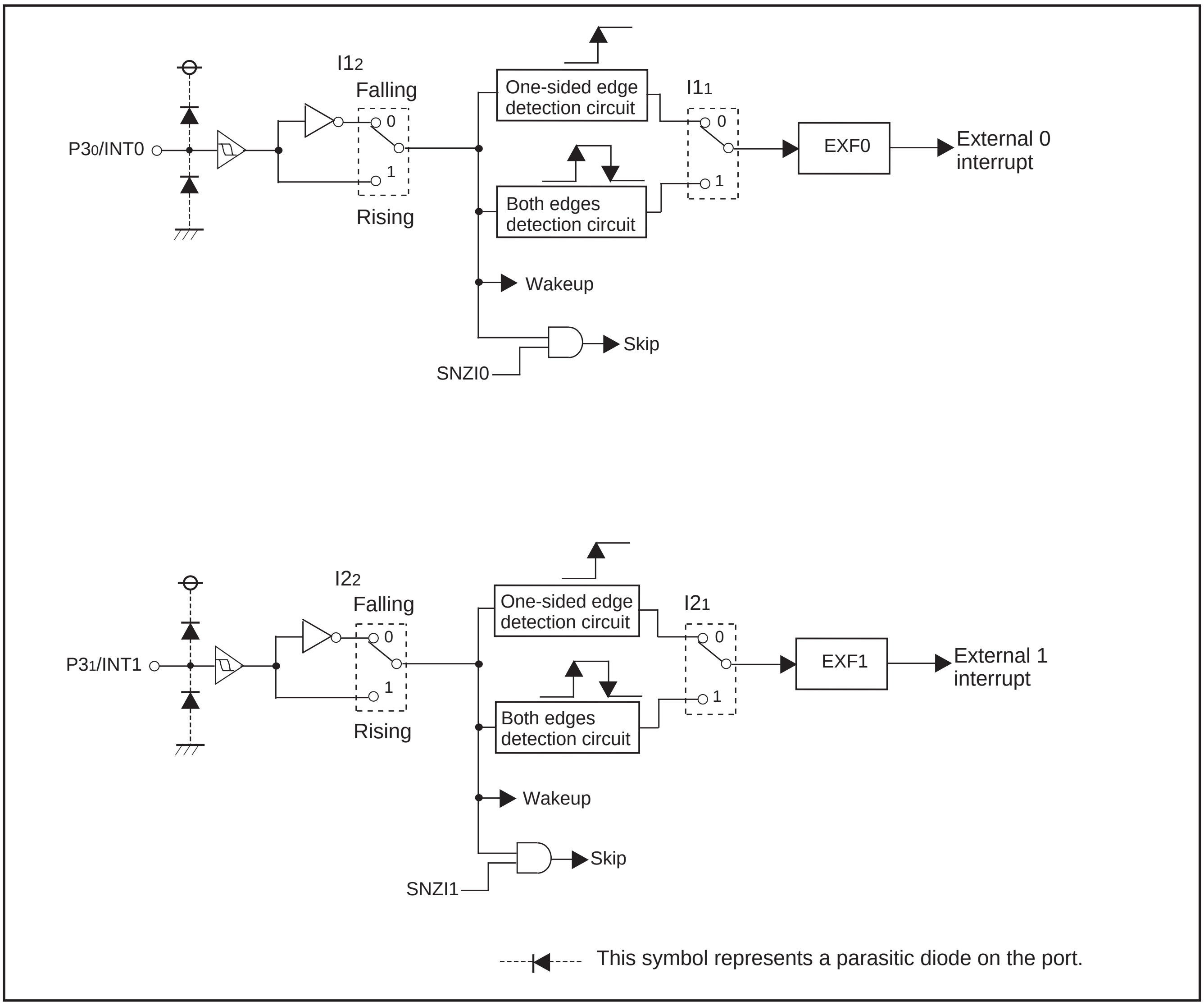


- This symbol represents a parasitic diode on the port.
- i represents 0, 1, 2, or 3.
- The 4513 Group does not have port P4.

PORT BLOCK DIAGRAMS (continued)



- This symbol represents a parasitic diode on the port.
- Applied potential to ports D0–D7 must be 12 V.
- i represents 0, 1, 2, or 3.
- The 4513 Group does not have port P5.



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).

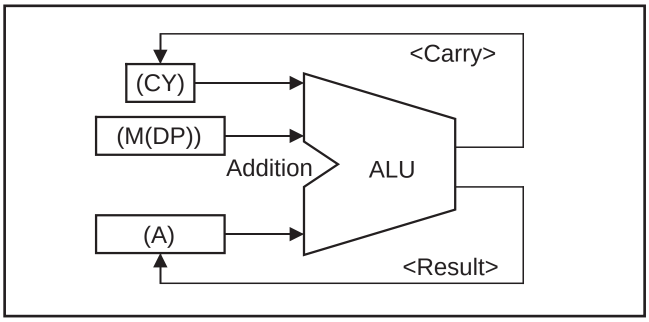


Fig. 1 AMC instruction execution example

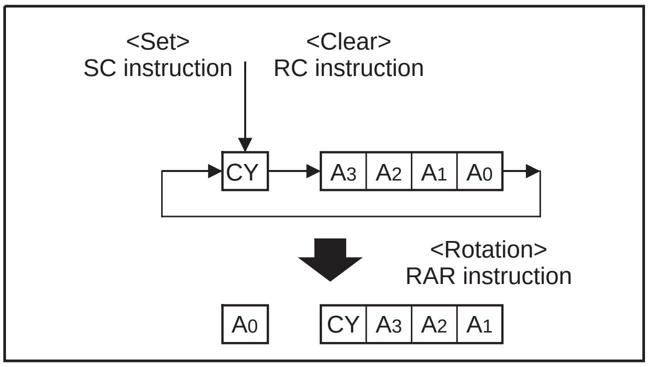


Fig. 2 RAR instruction execution example

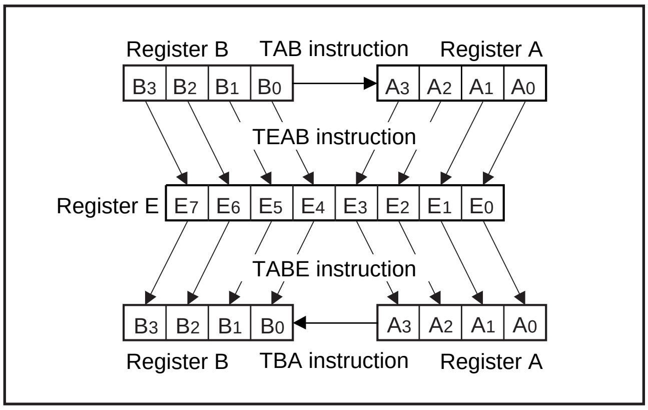


Fig. 3 Registers A, B and register E

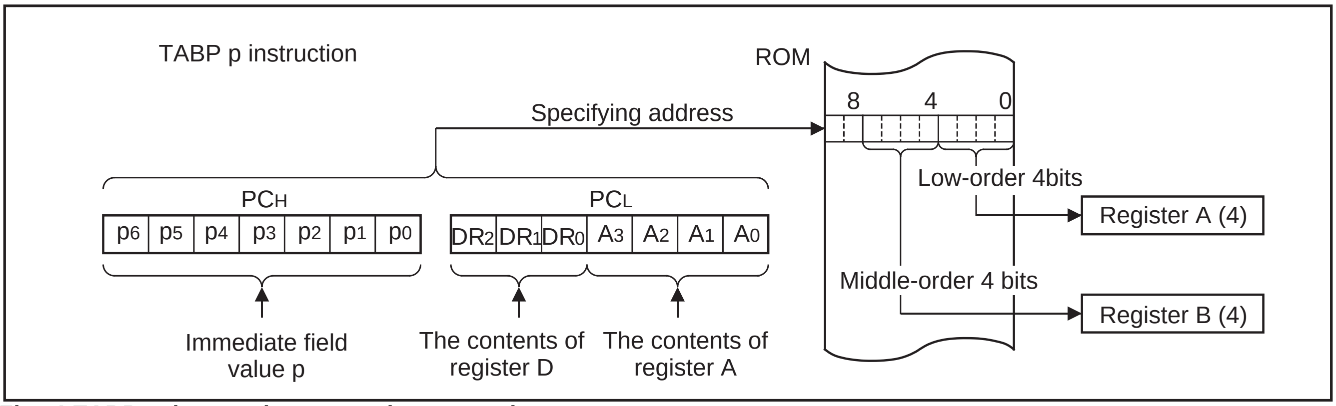


Fig. 4 TABP p instruction execution example

(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.

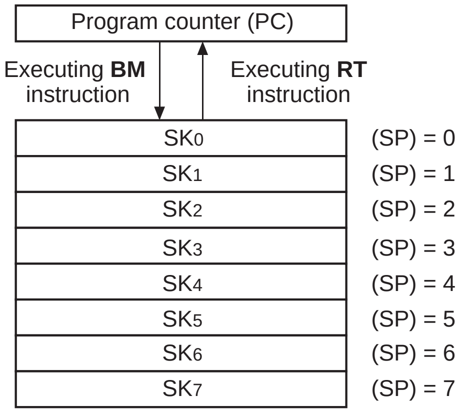
(6) Interrupt stack register (SDP)

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.

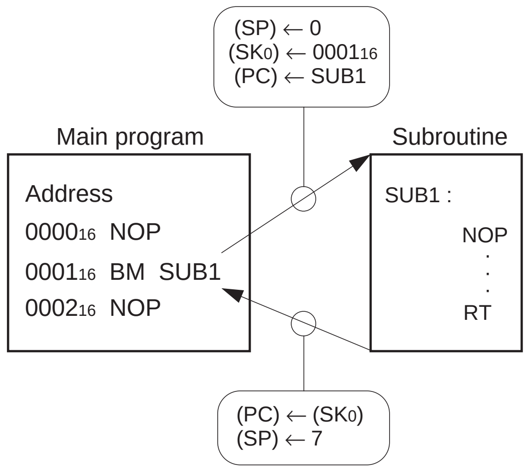
(7) Skip flag

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.

(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).

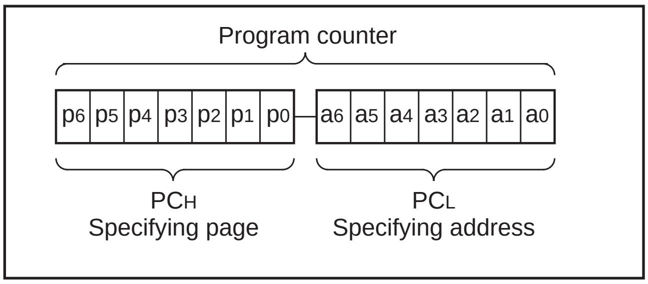


Fig. 7 Program counter (PC) structure

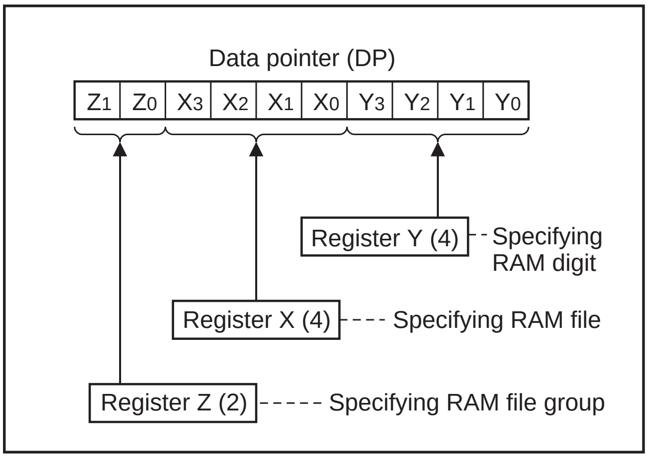


Fig. 8 Data pointer (DP) structure

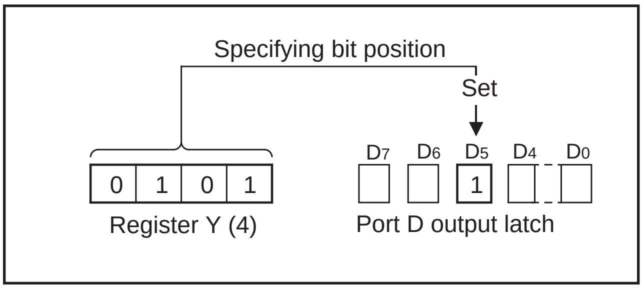


Fig. 9 SD instruction execution example

PROGRAM MEMOY (ROM)

The program memory is a mask ROM. 1 word of ROM is composed of 10 bits. ROM is separated every 128 words by the unit of page (addresses 0 to 127). Table 1 shows the ROM size and pages. Figure 10 shows the ROM map of M34514M8/E8.

Table 1 ROM size and pages

Product	ROM size (× 10 bits)	Pages
M34513M2	2048 words	16 (0 to 15)
M34513M4/E4	4096 words	32 (0 to 31)
M34513M6	6144 words	48 (0 to 47)
M34513M8/E8	8192 words	64 (0 to 63)
M34514M6	6144 words	48 (0 to 47)
M34514M8/E8	8192 words	64 (0 to 63)

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

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

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

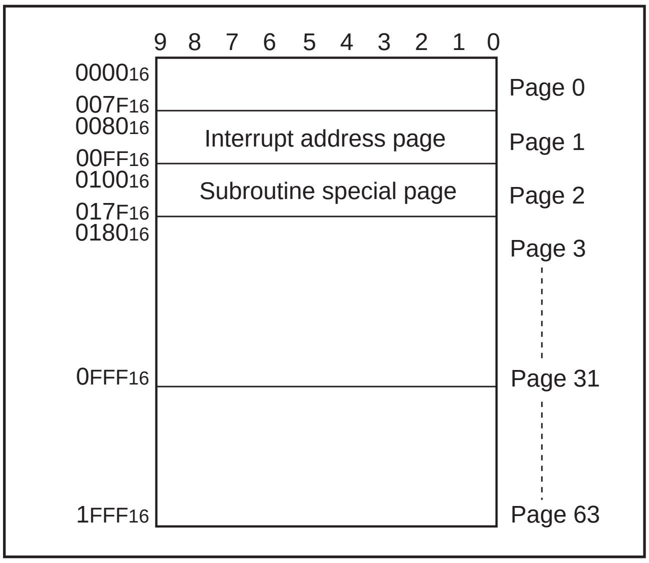


Fig. 10 ROM map of M34514M8/E8

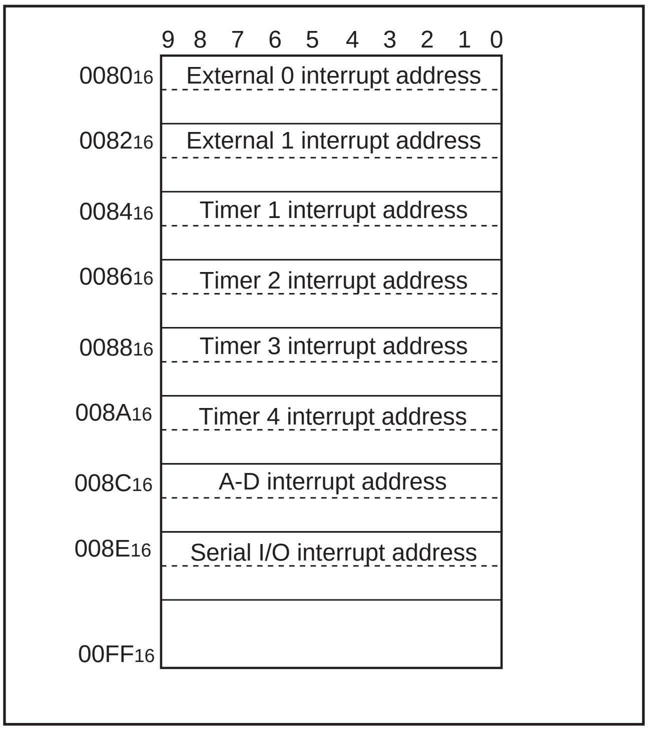


Fig. 11 Page 1 (addresses 0080_{16} to $00FF_{16}$) structure

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
M34513M2	128 words ✕ 4 bits (512 bits)
M34513M4/E4	256 words ✕ 4 bits (1024 bits)
M34513M6	384 words ✕ 4 bits (1536 bits)
M34513M8/E8	384 words ✕ 4 bits (1536 bits)
M34514M6	384 words ✕ 4 bits (1536 bits)
M34514M8/E8	384 words ✕ 4 bits (1536 bits)

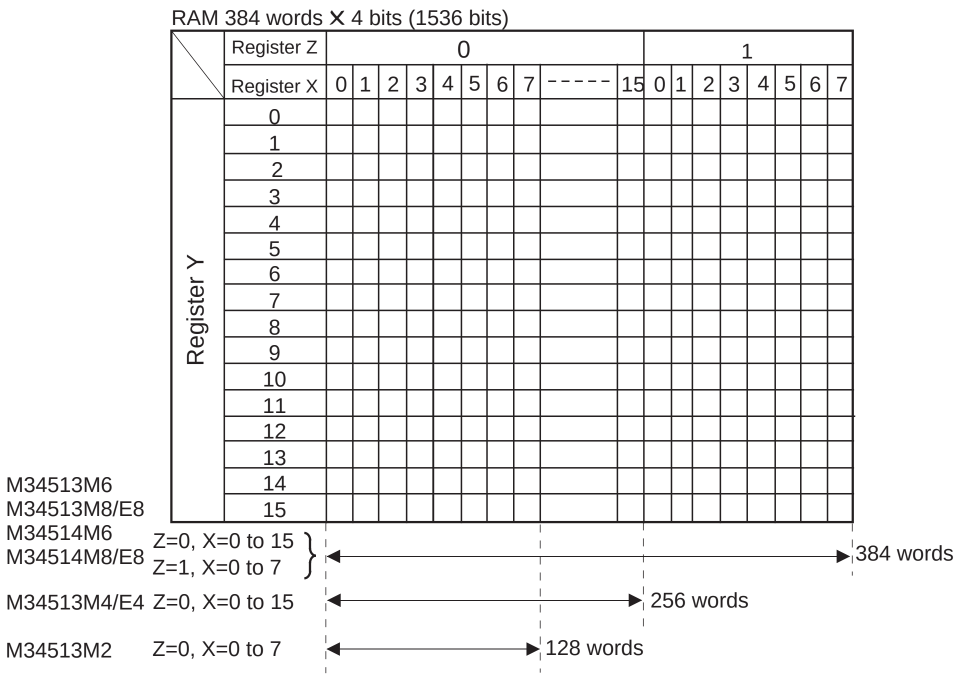


Fig. 12 RAM map

INTERRUPT FUNCTION

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

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

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

(1) Interrupt enable flag (INTE)

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

(2) Interrupt enable bit

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

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

Table 5 shows the interrupt enable bit function.

(3) Interrupt request flag

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

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

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

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

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

Table 3 Interrupt sources

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

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

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

Table 5 Interrupt enable bit function

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

(4) Internal state during an interrupt

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

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

(5) Interrupt processing

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

Use the RTI instruction to return from an interrupt service routine.

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

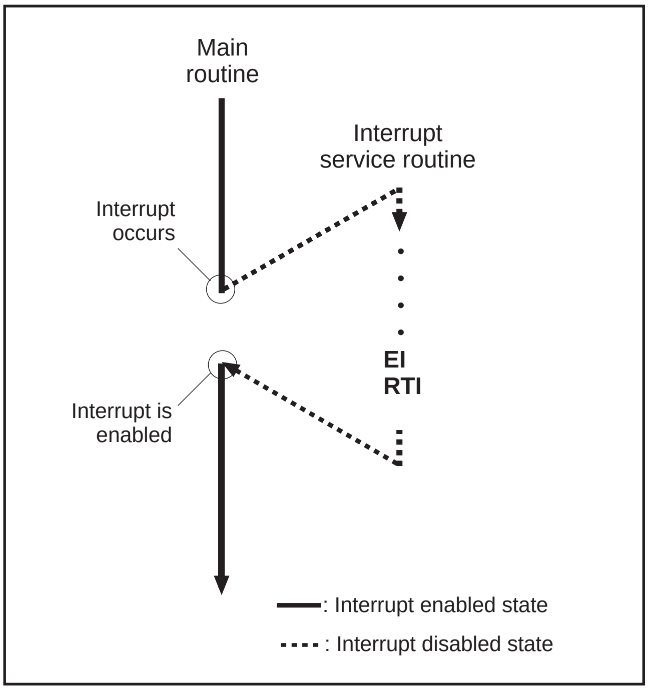


Fig. 13 Program example of interrupt processing

- 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

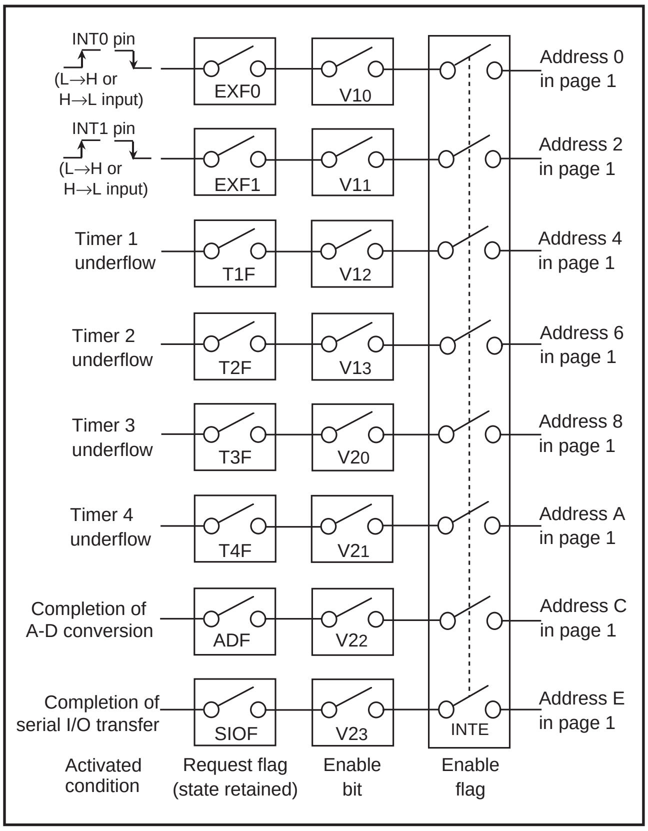


Fig. 15 Interrupt system diagram

(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

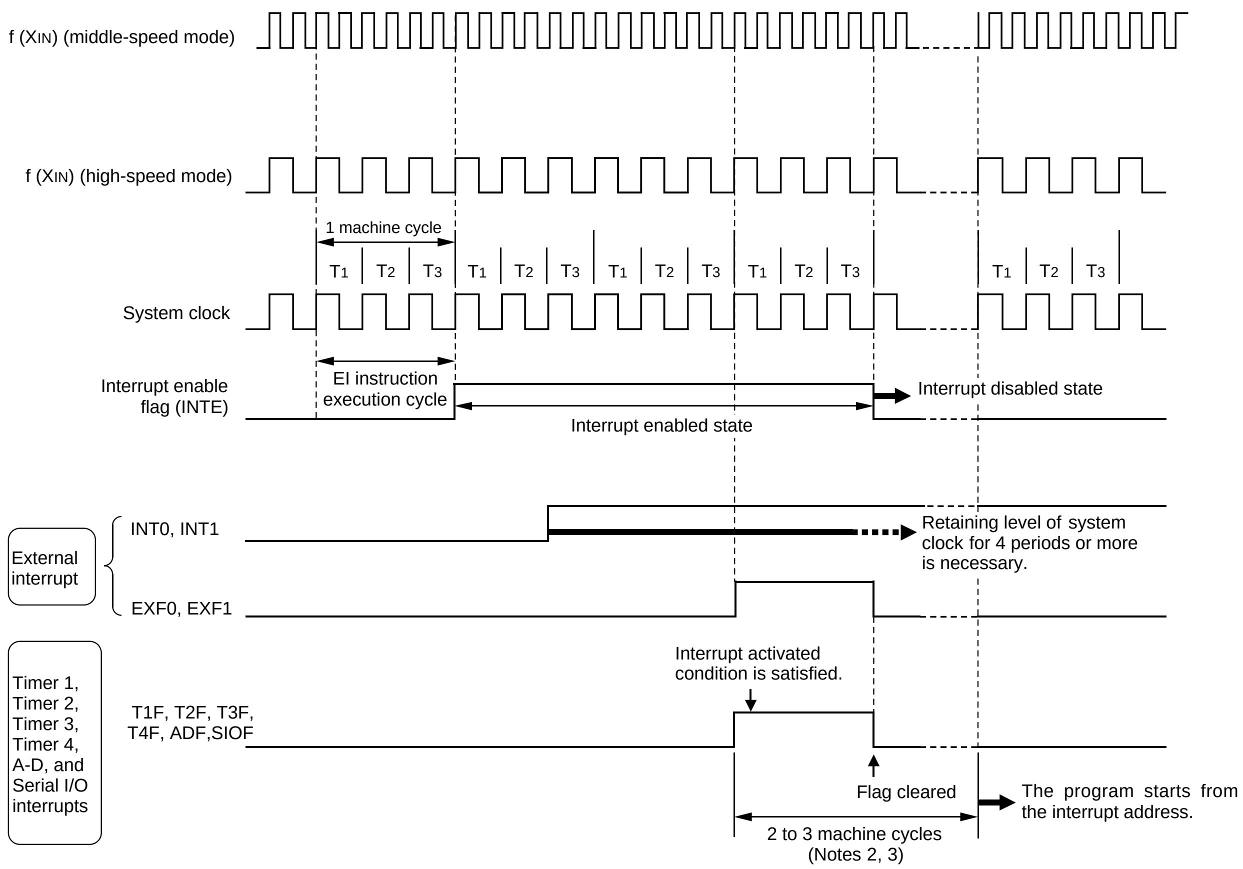
Interrupt control register V1		at reset : 0000_2		at RAM back-up : 0000_2	R/W
$V1_3$	Timer 2 interrupt enable bit	0	Interrupt disabled (SNZT2 instruction is valid)		
		1	Interrupt enabled (SNZT2 instruction is invalid)		
$V1_2$	Timer 1 interrupt enable bit	0	Interrupt disabled (SNZT1 instruction is valid)		
		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 ($V1_0$–$V1_3$ and $V2_0$–$V2_3$), and interrupt request flag are "1." The interrupt actually occurs 2 to 3 machine cycles after the cycle in which all three conditions are satisfied. The interrupt occurs after 3 machine cycles only when the three interrupt conditions are satisfied on execution of other than one-cycle instructions (Refer to Figure 16).

● When an interrupt request flag is set after its interrupt is enabled (Note 1)



Notes 1: The 4513/4514 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 4513/4514 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	$P3_0$/INT0	When the next waveform is input to $P3_0$/INT0 pin • Falling waveform (“H”→“L”) • Rising waveform (“L”→“H”) • Both rising and falling waveforms	$I1_1$ $I1_2$
External 1 interrupt	$P3_1$/INT1	When the next waveform is input to $P3_1$/INT1 pin • Falling waveform (“H”→“L”) • Rising waveform (“L”→“H”) • Both rising and falling waveforms	$I2_1$ $I2_2$

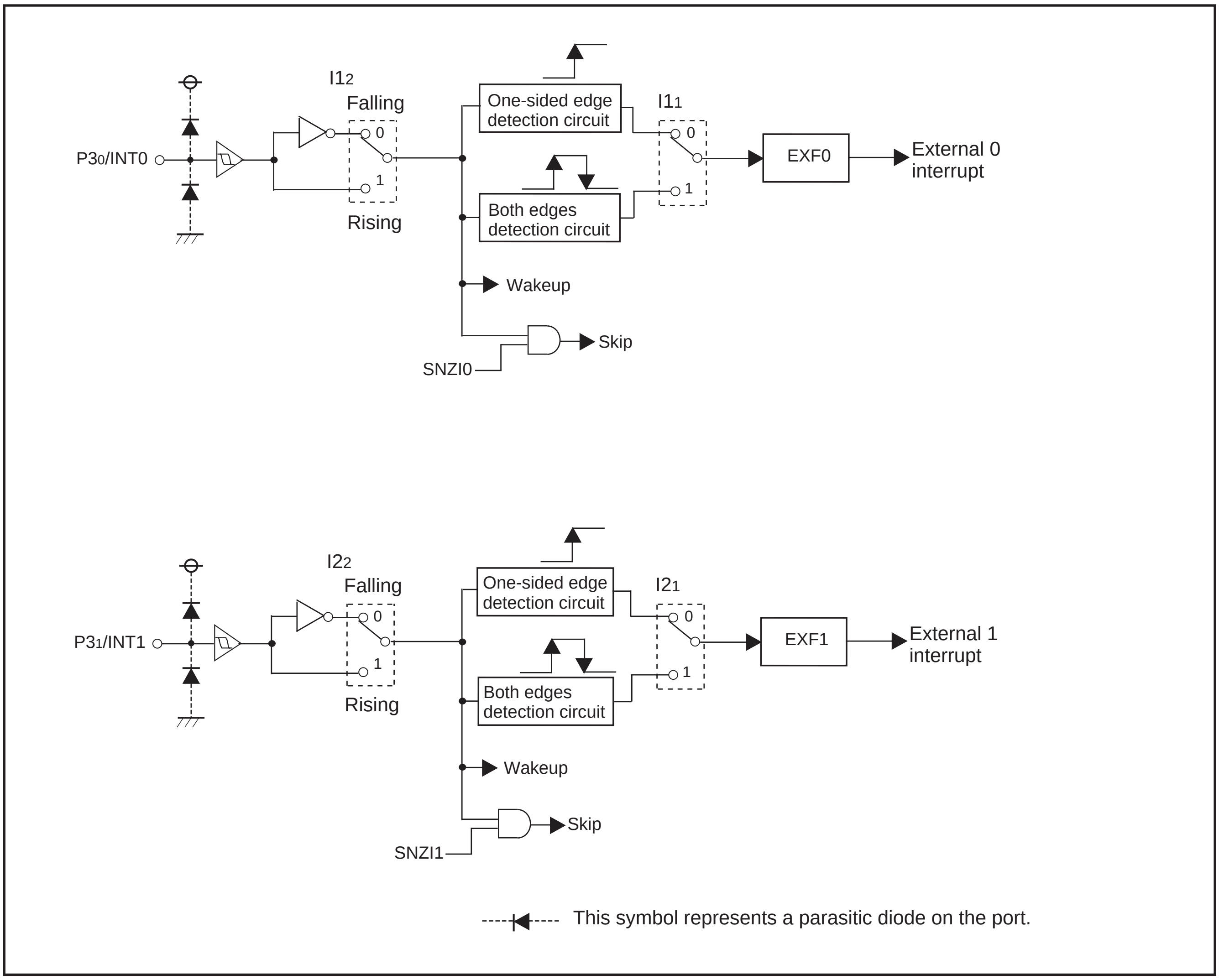


Fig. 17 External interrupt circuit structure

(1) External 0 interrupt request flag (EXF0)

External 0 interrupt request flag (EXF0) is set to “1” when a valid waveform is input to $P3_0$/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.
The $P3_0$/INT0 pin need not be selected the external interrupt input INT0 function or the normal I/O port $P3_0$ function. However, the EXF0 flag is set to “1” when a valid waveform is input even if it is used as an I/O port $P3_0$.

• External 0 interrupt activated condition
External 0 interrupt activated condition is satisfied when a valid waveform is input to $P3_0$/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 $P3_0$/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 $P3_1$/INT1 pin.
The valid waveforms causing the interrupt must be retained at their level for 4 clock cycles or more of the system clock (Refer to Figure 16).
The state of EXF1 flag can be examined with the skip instruction (SNZ1). Use the interrupt control register V1 to select the interrupt or the skip instruction. The EXF1 flag is cleared to “0” when an interrupt occurs or when the next instruction is skipped with the skip instruction.
The $P3_1$/INT1 pin need not be selected the external interrupt input INT1 function or the normal I/O port $P3_1$ function. However, the EXF1 flag is set to “1” when a valid waveform is input even if it is used as an I/O port $P3_1$.

• External 1 interrupt activated condition
External 1 interrupt activated condition is satisfied when a valid waveform is input to $P3_1$/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 $P3_1$/INT1 pin, the EXF1 flag is set to “1” and the external 1 interrupt occurs.

(3) External interrupt control registers

• Interrupt control register I1

Register I1 controls the 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 4513/4514 Group has the programmable timers.

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

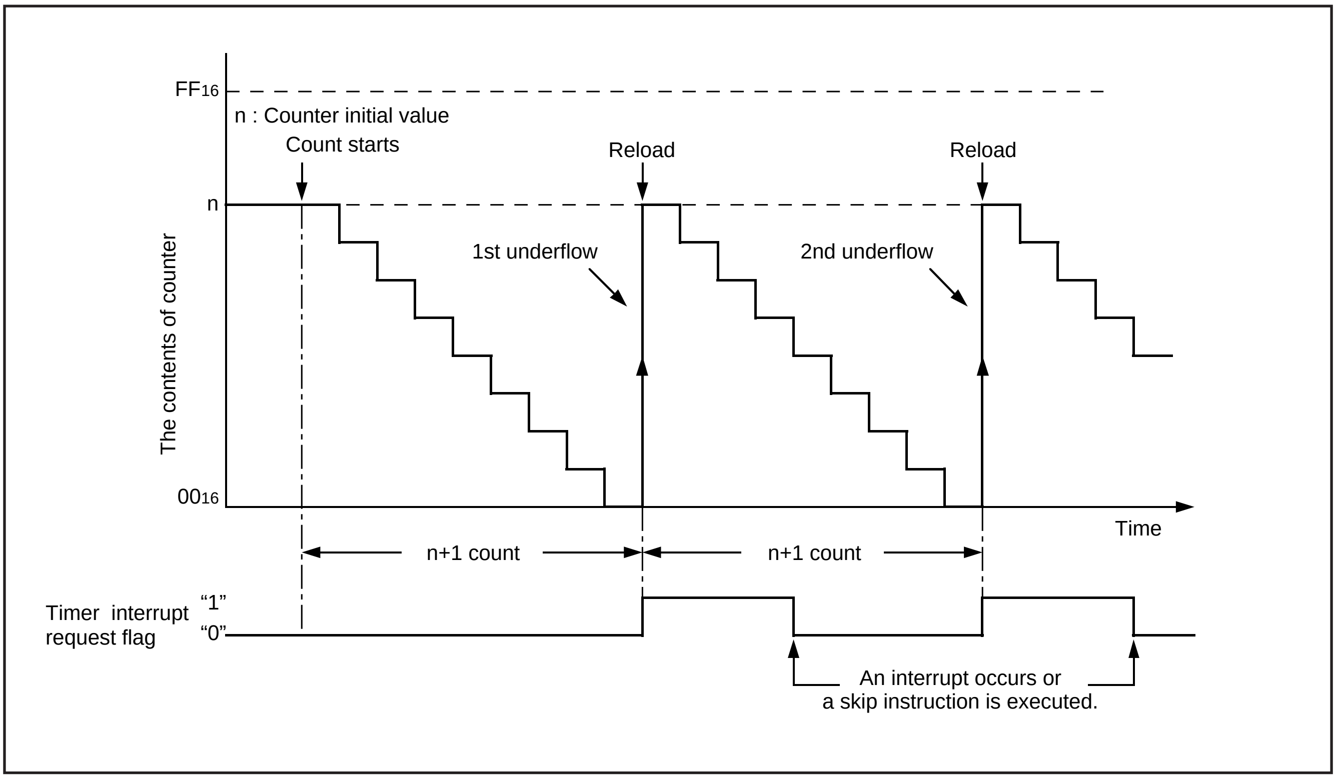


Fig. 18 Auto-reload function

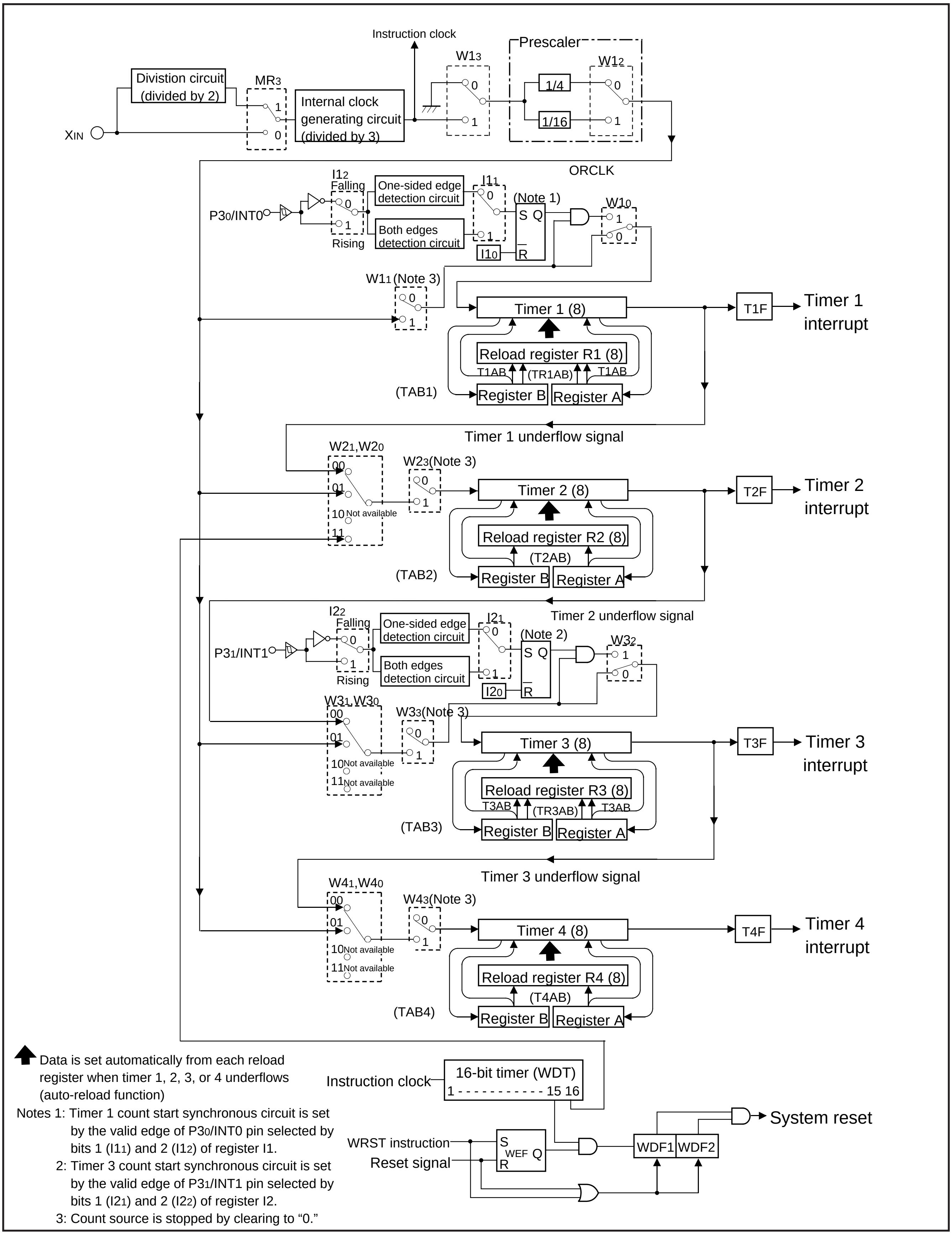
The 4513/4514 Group timer consists of the following circuits.

- 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 W6. The 16-bit timer is a free counter which is not controlled with the control register.
Each function is described below.

Table 9 Function related timers

Circuit	Structure	Count source	Frequency dividing ratio	Use of output signal	Control register
Prescaler	Frequency divider	• Instruction clock	4, 16	• Timer 1, 2, 3 and 4 count sources	W1
Timer 1	8-bit programmable binary down counter (link to $P3_0$/INT0 input)	• Prescaler output (ORCLK)	1 to 256	• Timer 2 count source • CNTR0 output • Timer 1 interrupt	W1 W6
Timer 2	8-bit programmable binary down counter	• Timer 1 underflow • Prescaler output (ORCLK) • CNTR0 input • 16-bit timer underflow	1 to 256	• Timer 3 count source • Timer 2 interrupt • CNTR0 output	W2 W6
Timer 3	8-bit programmable binary down counter (link to $P3_1$/INT1 input)	• Timer 2 underflow • Prescaler output (ORCLK)	1 to 256	• Timer 4 count source • Timer 3 interrupt • CNTR1 output	W3 W6
Timer 4	8-bit programmable binary down counter	• Timer 3 underflow • Prescaler output (ORCLK) • CNTR1 input	1 to 256	• Timer 4 interrupt • CNTR1 output	W4 W6
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 (I11) and 2 (I12) of register I1.

2: Timer 3 count start synchronous circuit is set by the valid edge of P31/INT1 pin selected by bits 1 (I21) and 2 (I22) 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		
$W2_1$, $W2_0$	Timer 2 count source selection bits	$W2_1$	$W2_0$	Count source
		0	0	Timer 1 underflow signal
		0	1	Prescaler output
		1	0	CNTR0 input
		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
$W3_1$, $W3_0$	Timer 3 count source selection bits	$W3_1$	$W3_0$	Count source
		0	0	Timer 2 underflow signal
		0	1	Prescaler output
		1	0	Not available
		1	1	Not available
Timer control register W4		at reset : 0000_2	at RAM back-up : state retained	R/W
$W4_3$	Timer 4 control bit	0		Stop (state retained)
		1		Operating
$W4_2$	Not used	0		This bit has no function, but read/write is enabled.
		1		
$W4_1$, $W4_0$	Timer 4 count source selection bits	$W4_1$	$W4_0$	Count source
		0	0	Timer 3 underflow signal
		0	1	Prescaler output
		1	0	CNTR1 input
		1	1	Not available
Timer control register W6		at reset : 0000_2	at RAM back-up : state retained	R/W
$W6_3$	CNTR1 output control bit	0		Timer 3 underflow signal output divided by 2
		1		CNTR1 output control by timer 4 underflow signal divided by 2
$W6_2$	D_7/CNTR1 function selection bit	0		D_7(I/O)/CNTR1 input
		1		CNTR1 (I/O)/D_7(input)
$W6_1$	CNTR0 output control bit	0		Timer 1 underflow signal output divided by 2
		1		CNTR0 output control by timer 2 underflow signal divided by 2
$W6_0$	D_6/CNTR0 output control bit	0		D_6(I/O)/CNTR0 input
		1		CNTR0 (I/O)/D_6(input)

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

(1) Timer control registers

- Timer control register W1
  Register W1 controls the count operation of timer 1, 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.
- Timer control register W6
  Register W6 controls the D_6/CNTR0 pin and D_7/CNTR1 functions, the selection and operation of the CNTR0 and CNTR1 output. Set the contents of this register through register A with the TW6A instruction. The TAW6 instruction can be used to transfer the contents of register W6 to register A.

(2) 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, $P3_0$/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 in timer 1 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. Timer 1 underflow signal divided by 2 can be output from D_6/CNTR0 pin.

(5) Timer 2 (interrupt function)

Timer 2 is an 8-bit binary down counter with the timer 2 reload register (R2). Data can be set simultaneously in timer 2 and the reload register (R2) with the T2AB instruction.
Timer 2 starts counting after the following process;
① set data in timer 2,
② select the count source with the bits 0 and 1 of register W2, and
③ set the bit 3 of register W2 to “1.”
When a value set in timer 2 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. The output from D_6/CNTR0 pin by timer 2 underflow signal divided by 2 can be controlled.

(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, $P3_1$/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 in timer 3 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. Timer 3 underflow signal divided by 2 can be output from D_7/CNTR1 pin.

(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 in timer 4 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. The output from D_7/CNTR1 pin by timer 4 underflow signal divided by 2 can be controlled.

(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) Timer I/O pin (D_6/CNTR0, D_7/CNTR1)

D_6/CNTR0 pin has functions to input the timer 2 count source, and to output the timer 1 and timer 2 underflow signals divided by 2. D_7/CNTR1 pin has functions to input the timer 4 count source, and to output the timer 3 and timer 4 underflow signals divided by 2.

The selection of D_6/CNTR0 pin function can be controlled with the bit 0 of register W6. The selection of D_7/CNTR1 pin function can be controlled with the bit 2 of register W6.

The following signals can be selected for the CNTR0 output signal with the bit 1 of register W6.
- timer 1 underflow signal divided by 2
- the signal of AND operation between timer 1 underflow signal divided by 2 and timer 2 underflow signal divide by 2

The following signals can be selected for the CNTR1 output signal with the bit 3 of register W6.
- timer 3 underflow signal divided by 2
- the signal of AND operation between timer 3 underflow signal divided by 2 and timer 4 underflow signal divide by 2

Timer 2 counts the rising waveform of CNTR0 input when the CNTR0 input is selected as the count source.

Timer 4 counts the rising waveform of CNTR1 input when the CNTR1 input is selected as the count source.

(10) Count start synchronous circuit (timer 1 and 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 runs wild. 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. 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.

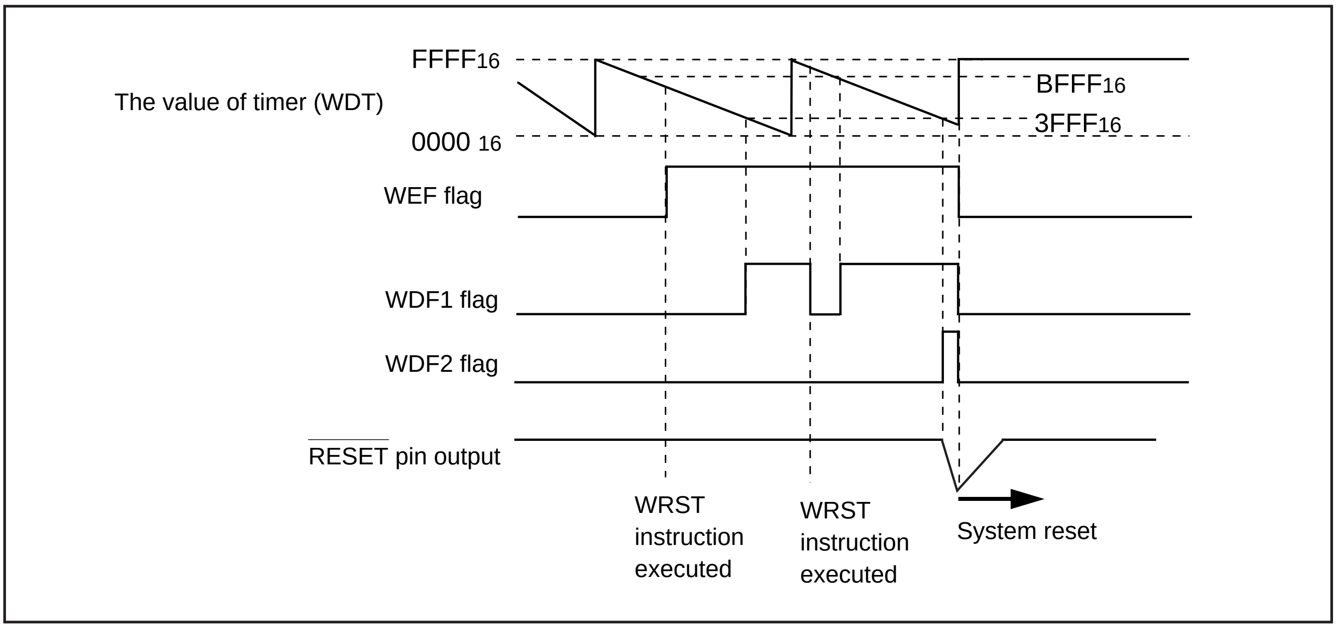


Fig. 20 Watchdog timer function

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 4513/4514 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.

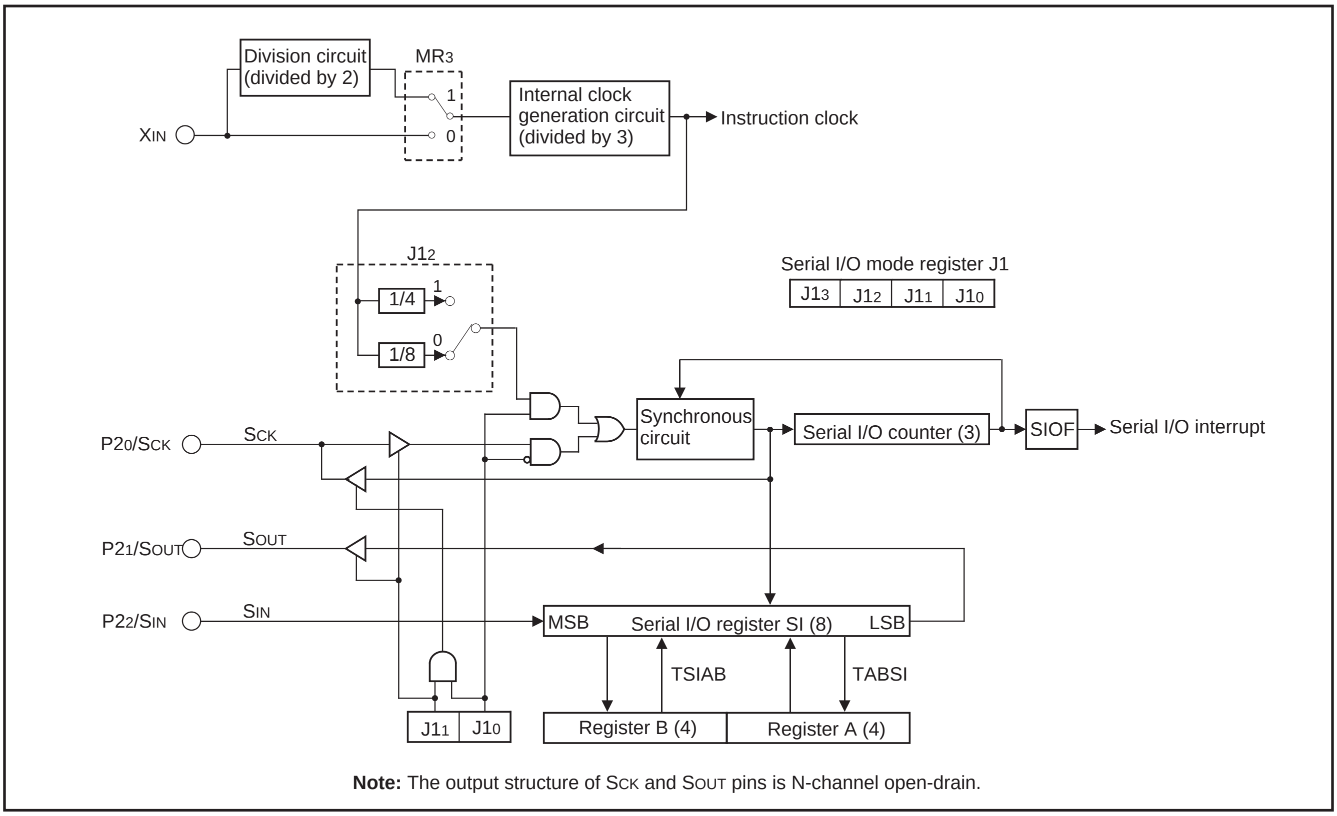


Fig. 22 Serial I/O structure

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.

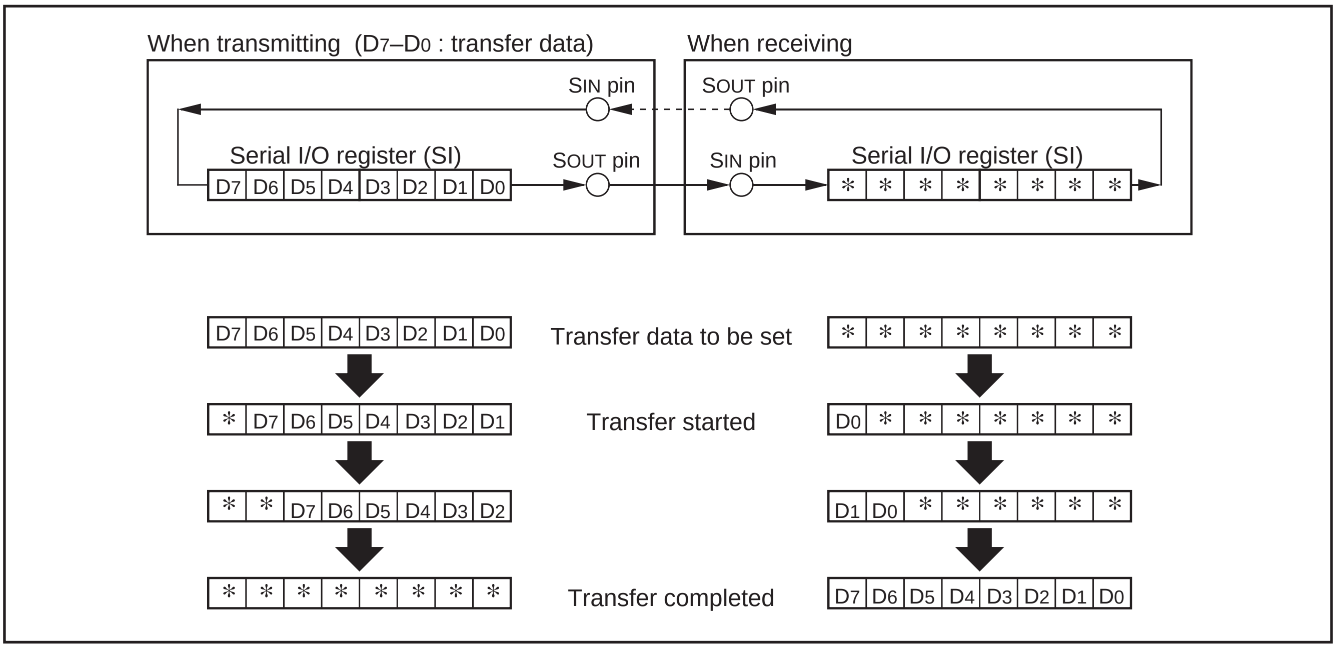


Fig. 23 Serial I/O register state when transferring

(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 $P2_0$.

(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, $P2_0$/SCK, $P2_1$/SOUT and $P2_2$/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.

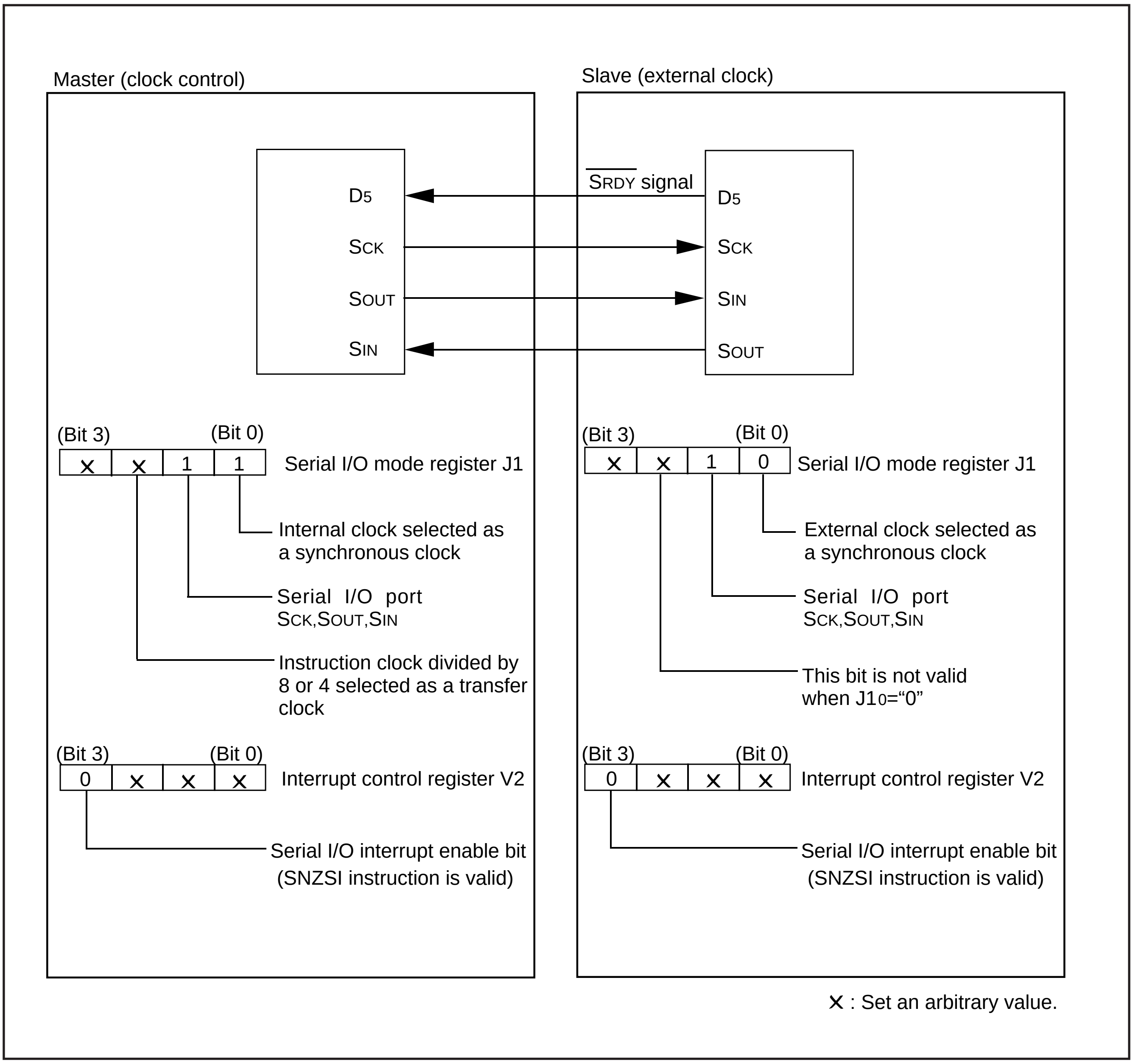
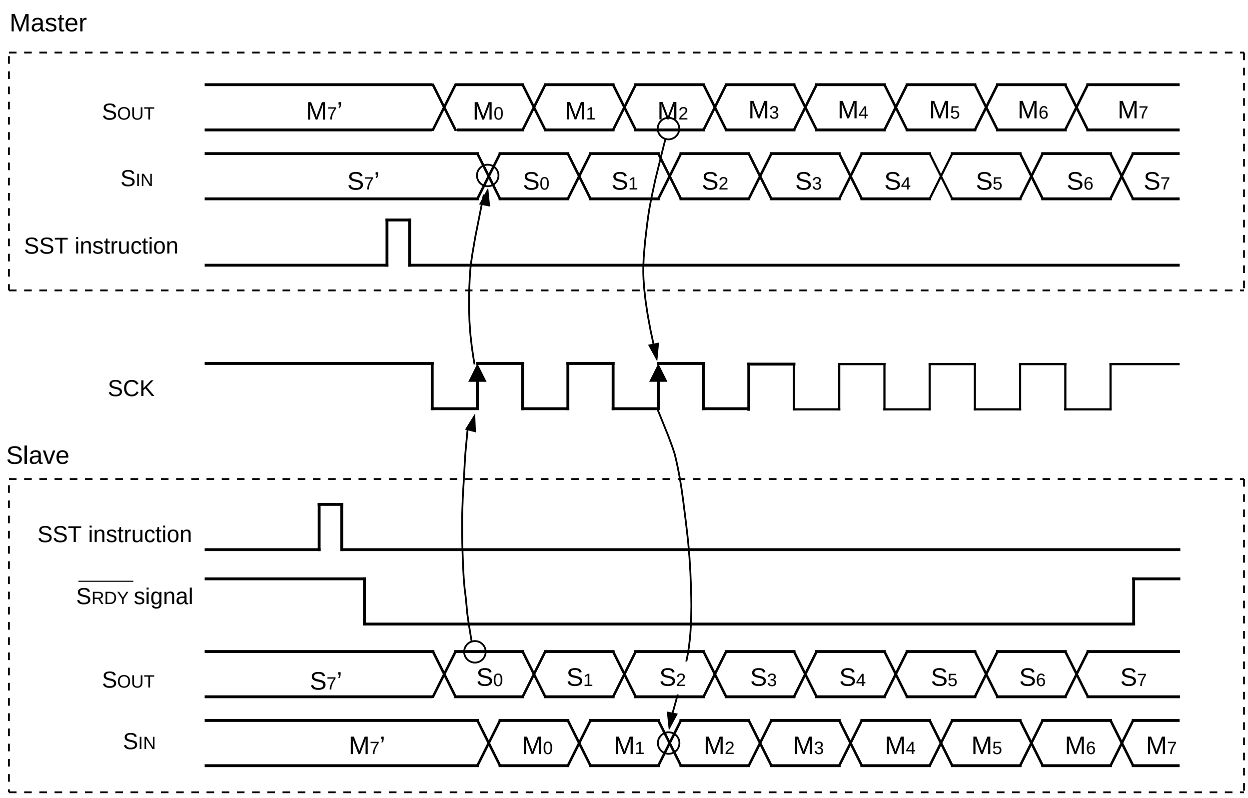


Fig. 24 Serial I/O connection example



M_0–M_7 : the contents of master serial I/O S_0–S_7 : the contents of slave serial I/O register
Rising of SCK : serial input Falling of SCK : 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]	[Initial setting]
• Setting the serial I/O mode register J1 and interrupt control register V2 shown in Figure 24.	• 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 ($\overline{S_{RDY}}$) to the input mode. (Port D_5 is used in this example)	• Setting the port transmitted the reception enable signal ($\overline{S_{RDY}}$) 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]

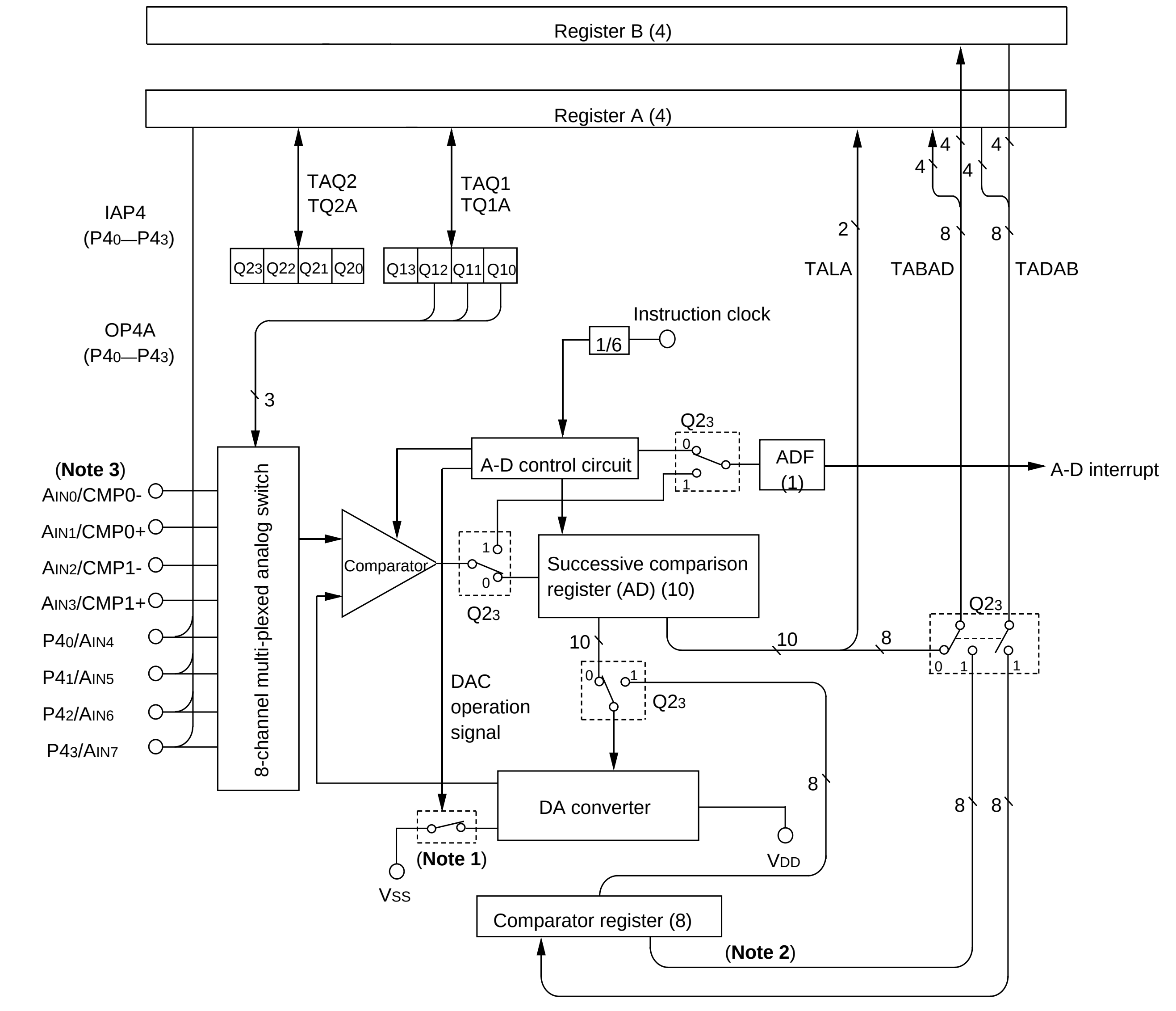
1-byte data is serially transferred on this process. Subsequently, data can be transferred continuously by repeating the process from *.
When an external clock is selected as a synchronous clock, the clock is not controlled internally. Control the clock externally because serial transfer is performed 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 4513/4514 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 for 4513 Group
	8 for 4514 Group



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.

3: The 4513 Group does not have ports P40/AIN4–P43/AIN7 and the IAP4 and OP4A instructions.

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 (Note 2)	$Q1_2$	$Q1_1$	$Q1_0$	Selected pins	
		0	0	0	A_{IN0}	
		0	0	1	A_{IN1}	
$Q1_1$		0	1	0	A_{IN2}	
		0	1	1	A_{IN3}	
		1	0	0	A_{IN4} (Not available for the 4513 Group)	
$Q1_0$		1	0	1	A_{IN5} (Not available for the 4513 Group)	
		1	1	0	A_{IN6} (Not available for the 4513 Group)	
		1	1	1	A_{IN7} (Not available for the 4513 Group)	

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	
$Q2_2$	$P4_3/A_{IN7}$ and $P4_2/A_{IN6}$ pin function selection bit (Not used for the 4513 Group)	0	$P4_3$, $P4_2$	(read/write enabled for the 4513 Group)
		1	A_{IN7}, $A_{IN6}/P4_3$, $P4_2$	(read/write enabled for the 4513 Group)
$Q2_1$	$P4_1/A_{IN5}$ pin function selection bit (Not used for the 4513 Group)	0	$P4_1$	(read/write enabled for the 4513 Group)
		1	$A_{IN5}/P4_1$	(read/write enabled for the 4513 Group)
$Q2_0$	$P4_0/A_{IN4}$ pin function selection bit (Not used for the 4513 Group)	0	$P4_0$	(read/write enabled for the 4513 Group)
		1	$A_{IN4}/P4_0$	(read/write enabled for the 4513 Group)

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: Select A_{IN4}–A_{IN7} with register Q1 after setting register Q2.

(1) Operating at A-D conversion mode

The A-D conversion mode is set by setting the bit 3 of register 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. The 4513 Group does not have A_{IN4}–A_{IN7}. Accordingly, do not select these pins with register Q1.

(6) A-D control register Q2

Register Q2 is used to select the pin function of $P4_0/A_{IN4}$, $P4_1/A_{IN5}$, $P4_2/A_{IN6}$, and $P4_3/A_{IN7}$. 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." After set this register, select the analog input with register Q1.

Even when register Q2 is used to set the pins for analog input, $P4_0/A_{IN4}$–$P4_3/A_{IN7}$ continue to function as $P4_0$–$P4_3$ I/O. Accordingly, when any of them are used as I/O port P4 and others are used as analog input pins, make sure to set the outputs of pins that are set for analog input to "1." Also, for the port input, the port input function of the pin functions as analog input is undefined.

(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 4513/4514 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.

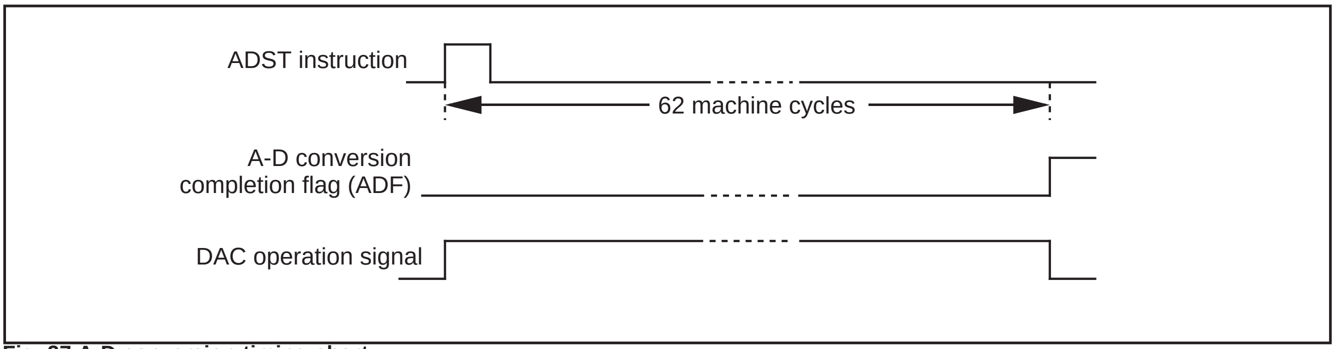


Fig. 27 A-D conversion timing chart

(9) How to use A-D conversion

How to use A-D conversion is explained using as example in which the analog input from P4$_0$/A$_{IN4}$ pin is A-D converted, and the high-order 4 bits of the converted data are stored in address M(Z, X, Y) = (0, 0, 0), the middle-order 4 bits in address M(Z, X, Y) = (0, 0, 1), and the low-order 2 bits in address M(Z, X, Y) = (0, 0, 2) of RAM. The A-D interrupt is not used in this example.

① After selecting the A$_{IN4}$ pin function with the bit 0 of the register Q2, select A$_{IN4}$ pin and A-D conversion mode with the register Q1 (refer to Figure 28).
② Execute the ADST instruction and start A-D conversion.
③ Examine the state of ADF flag with the SNZAD instruction to determine the end of A-D conversion.
④ Transfer the low-order 2 bits of converted data to the high-order 2 bits of register A (TALA instruction).
⑤ Transfer the contents of register A to M (Z, X, Y) = (0, 0, 2).
⑥ Transfer the high-order 8 bits of converted data to registers A and B (TABAD instruction).
⑦ Transfer the contents of register A to M (Z, X, Y) = (0, 0, 1).
⑧ Transfer the contents of register B to register A, and then, store into M(Z, X, Y) = (0, 0, 0).

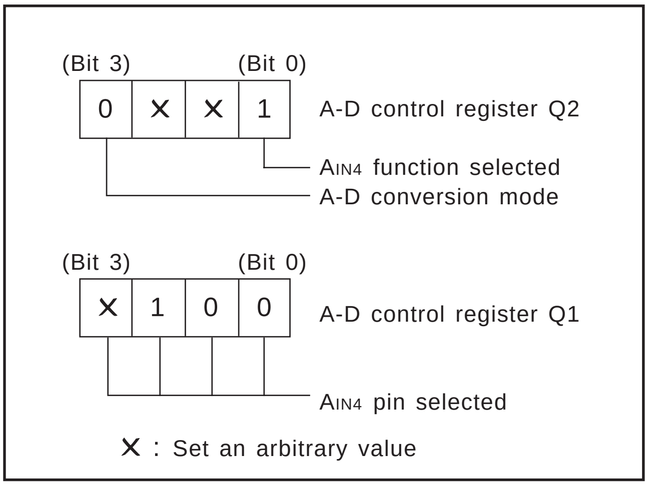


Fig. 28 Setting registers

(10) 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.

(11) 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)

(12) Comparison result store flag (ADF)

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

(13) Comparator operation start instruction (ADST instruction)

In comparator mode, executing ADST starts the comparator operating.
The comparator stops 8 machine cycles after it has started (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."

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

Note the following when using the analog input pins also for I/O port P4 functions:

- Even when $P4_0/A_{IN4}$–$P4_3/A_{IN7}$ are set to pins for analog input, they continue to function as $P4_0$–$P4_3$ I/O. Accordingly, when any of them are used as I/O port P4 and others are used as analog input pins, make sure to set the outputs of pins that are set for analog input to "1." Also, the port input function of the pin functions as an analog input is undefined.
- TALA instruction
  When the TALA instruction is executed, the low-order 2 bits of register AD is transferred to the high-order 2 bits of register A, simultaneously, the low-order 2 bits of register A is "0."

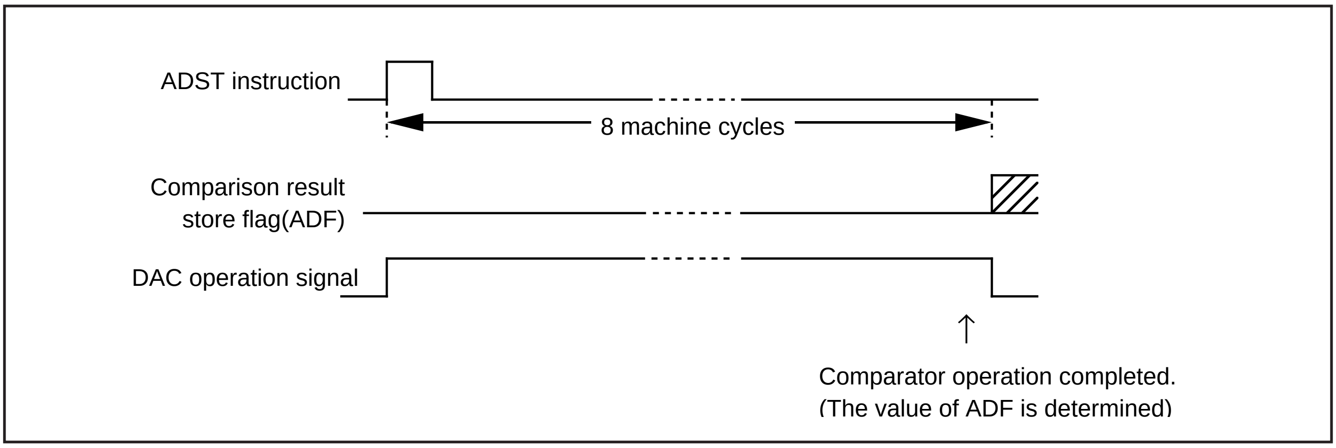


Fig. 29 Comparator operation timing chart

(15) 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.

(16) Definition of A-D converter accuracy

The A-D conversion accuracy is defined below (refer to Figure 30).

- 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.

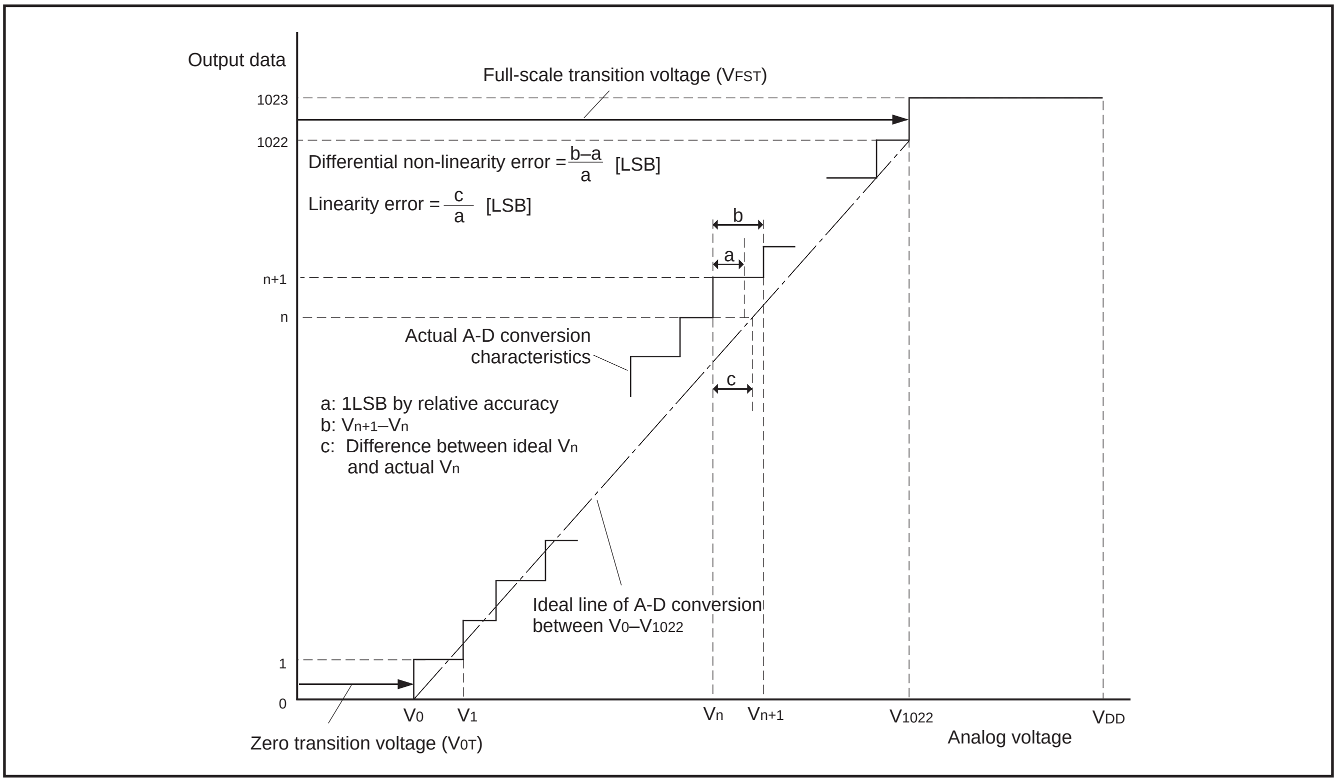


Fig. 30 Definition of A-D conversion accuracy

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

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

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

VOLTAGE COMPARATOR

The 4513/4514 Group has 2 voltage comparator circuits that perform comparison of voltage between 2 pins. Table 17 shows the characteristics of this voltage comparison.

Table 17 Voltage comparator characteristics

Parameter	Characteristics
Voltage comparator function	2 circuits (CMP0, CMP1)
Input pin	CMP0-, CMP0+ (also used as A_{IN0}, A_{IN1})
	CMP1-, CMP1+ (also used as A_{IN2}, A_{IN3})
Supply voltage	3.0 V to 5.5 V
Input voltage	0.3 V_{DD} to 0.7 V_{DD}
Comparison check error	Typ. 20 mV, Max.100 mV
Response time	Max. 20 µs

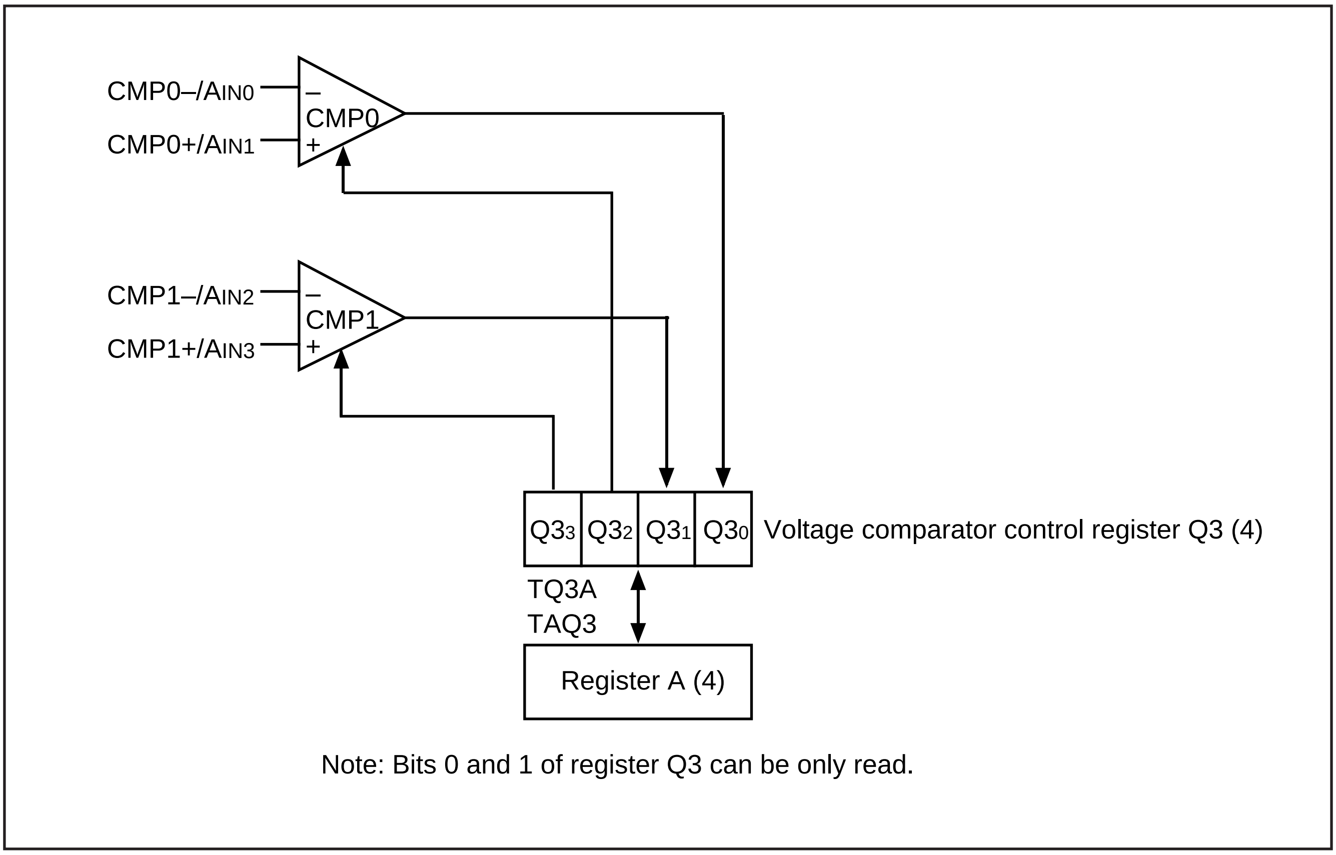


Fig. 31 Voltage comparator structure

Table 18 Voltage comparator control register Q3

Voltage comparator control register Q3 (Note 2)		at reset : 0000_2		at RAM back-up : state retained	R/W
$Q3_3$	Voltage comparator (CMP1) control bit	0	Voltage comparator (CMP1) invalid		
		1	Voltage comparator (CMP1) valid		
$Q3_2$	Voltage comparator (CMP0) control bit	0	Voltage comparator (CMP0) invalid		
		1	Voltage comparator (CMP0) valid		
$Q3_1$	CMP1 comparison result store bit	0	CMP1- > CMP1+		
		1	CMP1- < CMP1+		
$Q3_0$	CMP0 comparison result store bit	0	CMP0- > CMP0+		
		1	CMP0- < CMP0+		

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: Bits 0 and 1 of register Q3 can be only read.

(1) Voltage comparator control register Q3

Register Q3 controls the function of the voltage comparator.
The function of the voltage comparator CMP0 becomes valid by setting bit 2 of register Q3 to "1," and becomes invalid by setting bit 2 of register Q3 to "0." The comparison result of the voltage comparator CMP0 is stored into bit 0 of register Q3.
The function of the voltage comparator CMP1 becomes valid by setting bit 3 of register Q3 to "1," and becomes invalid by setting bit 3 of register Q3 to "0." The comparison result of the voltage comparator CMP1 is stored into bit 1 of register Q3.

(2) Operation description of voltage comparator

The voltage comparator function becomes valid by setting each control bit of register Q3 to "1" and compares the voltage of the input pin. The comparison result is stored into each comparison result store bit of register Q3.
The comparison result is as follows;
- When CMP0- > CMP0+, $Q3_0$ = "0"
  When CMP0- < CMP0+, $Q3_0$ = "1"
- When CMP1- > CMP1+, $Q3_1$ = "0"
  When CMP1- < CMP1+, $Q3_1$ = "1"

(3) Precautions

When the voltage comparator is used, note the following;
- Voltage comparator function
  When the voltage comparator function is valid with the voltage comparator control register Q3, it is operating even in the RAM back-up mode. Accordingly, be careful about such state because it causes the increase of the operation current in the RAM back-up mode.
  In order to reduce the operation current in the RAM back-up mode, invalidate (bits 2 and 3 of register Q3 = "0") the voltage comparator function by software before the POF instruction is executed.
  Also, while the voltage comparator function is valid, current is always consumed by voltage comparator. On the system required for the low-power dissipation, invalidate the voltage comparator by software when it is unused.

- Register Q3
  Bits 0 and 1 of register Q3 can be only read. Note that they cannot be written.

- Reading the comparison result of voltage comparator
  Read the voltage comparator comparison result from register Q3 after the voltage comparator response time (max. 20 μs) is passed from the voltage comparator function becomes valid.

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.

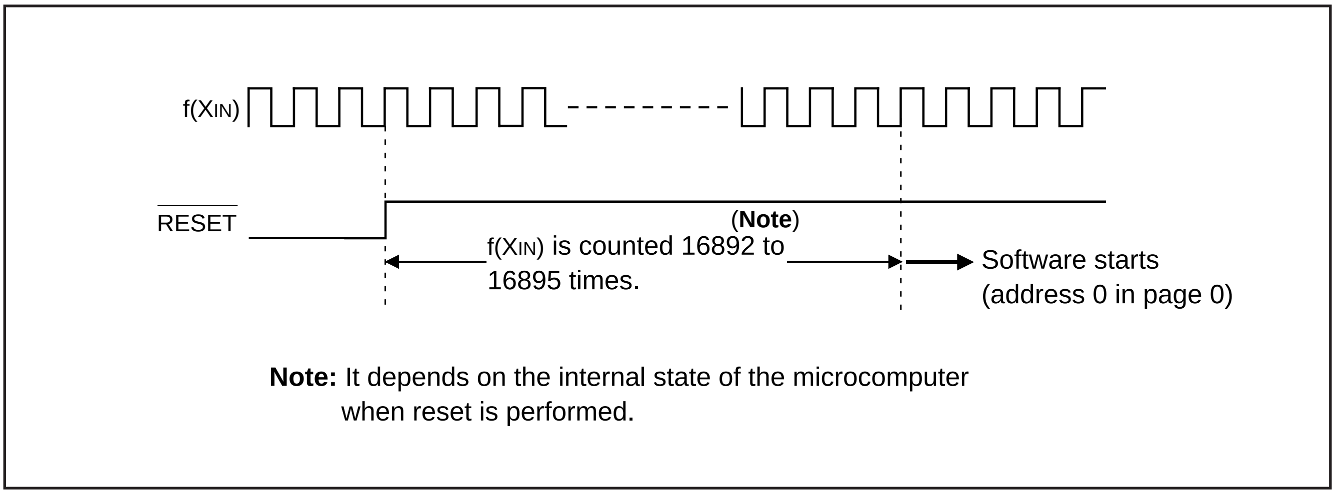


Fig. 32 Reset release timing

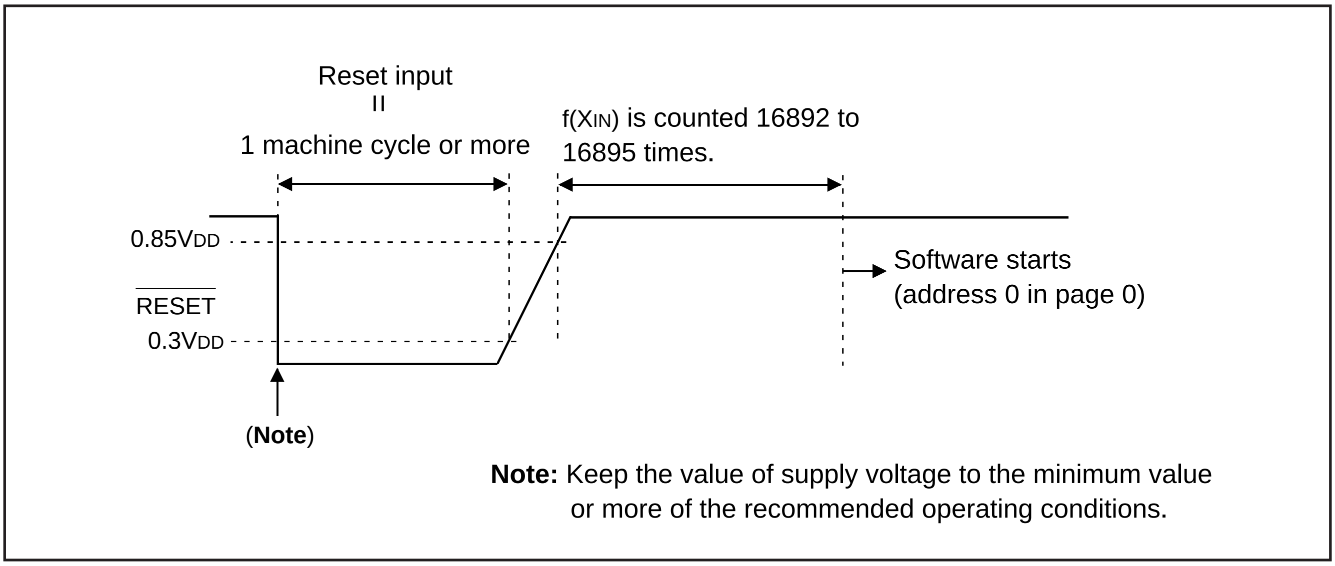


Fig. 33 $\overline{\text{RESET}}$ pin input waveform and reset operation

(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.

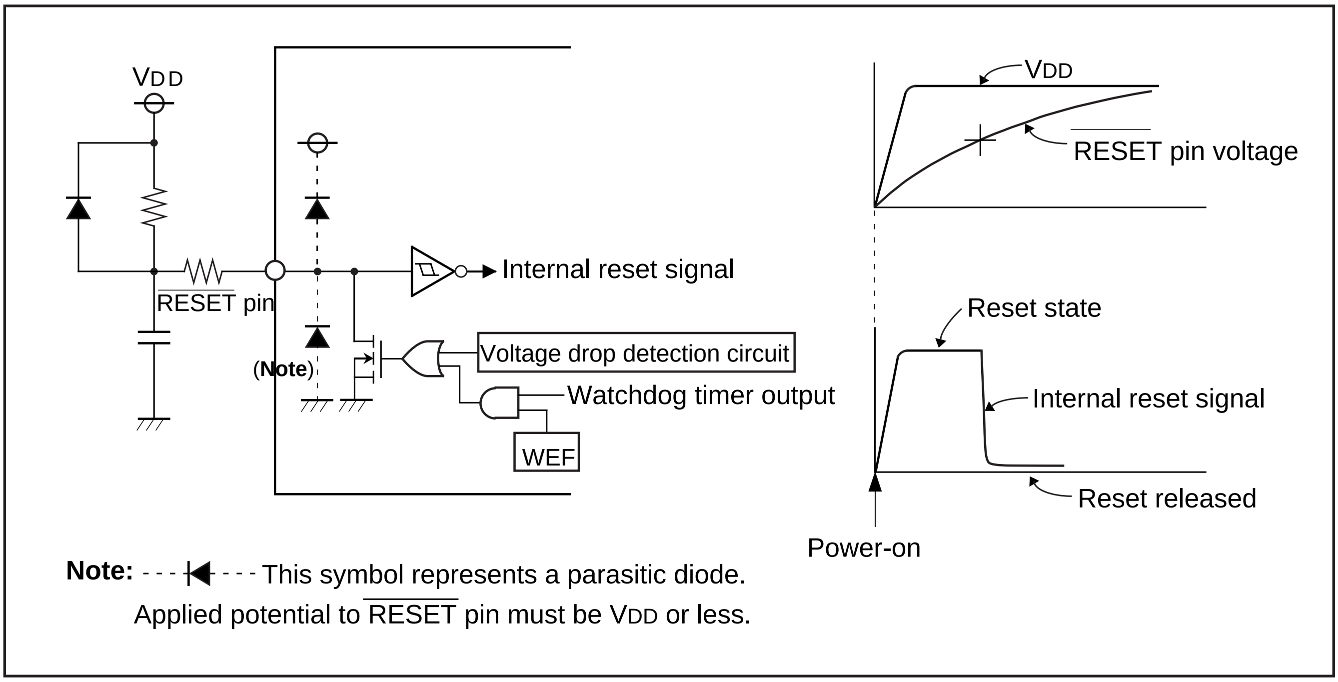


Fig. 34 Power-on reset circuit example

(2) Internal state at reset

Table 19 shows port state at reset, and Figure 35 shows internal state at reset (they are the same after system is released from reset). The contents of timers, registers, flags and RAM except shown in Figure 35 are undefined, so set the initial value to them.

Table 19 Port state at reset

Name	Function	State
D_0–D_5	D_0–D_5	High impedance (Note)
D_6/CNTR0, D_7/CNTR1	D_6, D_7	
$P0_0$–$P0_3$	$P0_0$–$P0_3$	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
$P3_0$/INT0, $P3_1$/INT1	$P3_0$, $P3_1$	High impedance (Note 1)
$P3_2$, $P3_3$ (Note 4)	$P3_2$, $P3_3$	
$P4_0$/A_{IN4}–$P4_3$/A_{IN7} (Note 4)	$P4_0$–$P4_3$	High impedance (Note 1)
$P5_0$–$P5_3$ (Note 4)	$P5_0$–$P5_3$	High impedance (Note 3)

Notes 1: Output latch is set to “1.”
2: Pull-up transistor is turned OFF.
3: After system is released from reset, port P5 is in the input mode. (Direction register FR0 = 0000_2)
4: The 4513 Group does not have these ports.

• 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)
• Timer control register W6	0 0 0 0	
• 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	× × × × × × × ×	
• 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	
• Voltage comparator control register Q3	0 0 0 0	
• Successive comparison register AD	× × × × × × × × × ×	
• Comparator register	× × × × × × × ×	
• Key-on wakeup control register K0	0 0 0 0	
• Pull-up control register PU0	0 0 0 0	
• Direction register FR0	0 0 0 0	(Port P5: input mode)
• Carry flag (CY)	0	
• Register A	0 0 0 0	
• Register B	0 0 0 0	
• Register D	× × ×	
• Register E	× × × × × × × ×	
• Register X	0 0 0 0	
• Register Y	0 0 0 0	
• Register Z	× ×	
• Stack pointer (SP)	1 1 1	

"×" represents undefined.

Fig. 35 Internal state at reset

VOLTAGE DROP DETECTION CIRCUIT

The built-in voltage drop detection circuit is designed to detect a drop in voltage and to reset the microcomputer if the supply voltage drops below a set value.

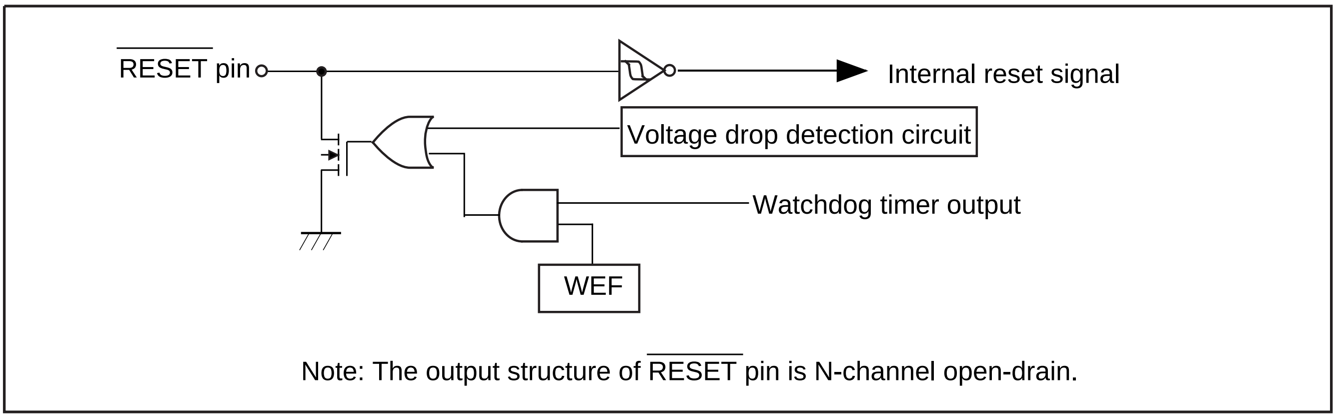


Fig. 36 Voltage drop detection reset circuit

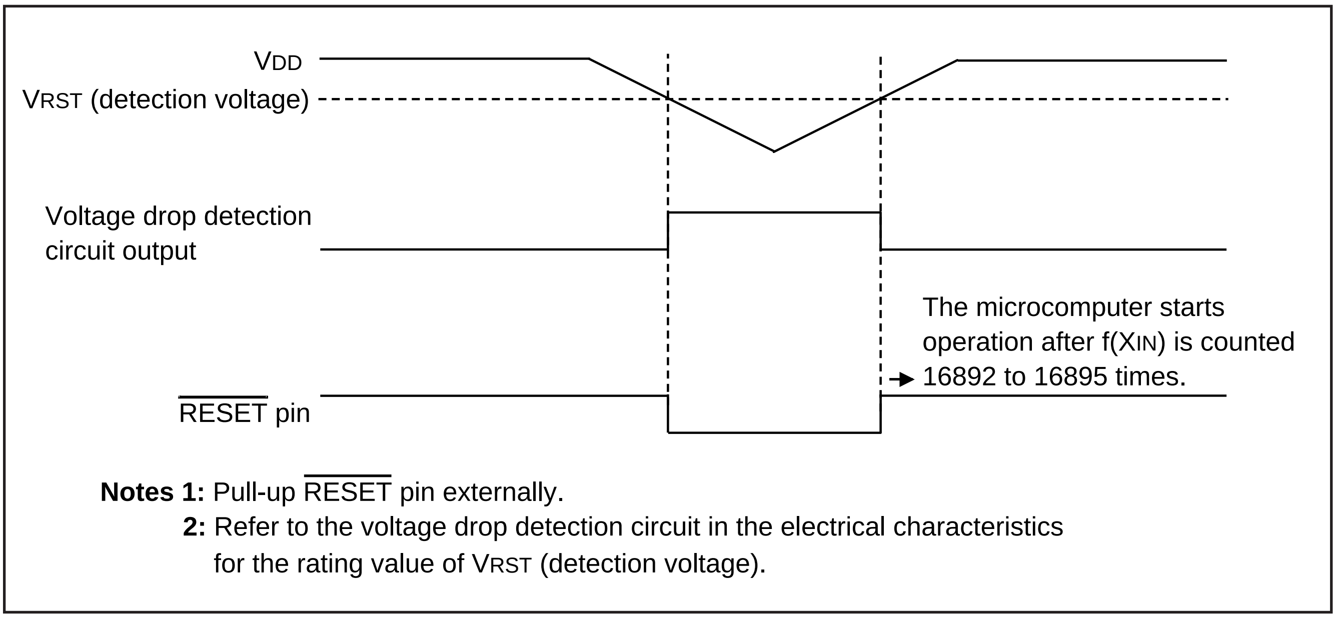


Fig. 37 Voltage drop detection circuit operation waveform

RAM BACK-UP MODE

The 4513/4514 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 20 shows the function and states retained at RAM back-up. Figure 38 shows the state transition.

(1) Identification of the start condition

Warm start (return from the RAM back-up state) or cold start (return from the normal reset state) can be identified by examining the state of the power down flag (P) with the SNZP instruction.

(2) Warm start condition

When the external wakeup signal is input after the system enters the RAM back-up state by executing the EPOF and POF instructions continuously, the CPU starts executing the program from address 0 in page 0. In this case, the P flag is “1.”

(3) Cold start condition

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

- reset pulse is input to $\overline{\text{RESET}}$ pin, or
- reset by watchdog timer is performed, or
- voltage drop detection circuit detects the voltage drop.

In this case, the P flag is “0.”

Table 20 Functions and states retained at RAM back-up

Function	RAM back-up
Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) (Note 2)	×
Contents of RAM	O
Port level	O
Timer control register W1	×
Timer control registers W2 to W4, W6	O
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
Voltage comparator function	O (Note 5)
Voltage comparator control register Q3	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
Direction register FR0	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.
5: The state is retained when the voltage comparator function is selected with the voltage comparator control register Q3.

(4) Return signal

An external wakeup signal is used to return from the RAM back-up mode because the oscillation is stopped. Table 21 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 21 Return source and return condition

Return source		Return condition	Remarks
External wakeup signal	Ports P0, P1	Return by an external falling edge input (“H”→“L”).	Set the port using the key-on wakeup function selected with register K0 to “H” level before going into the RAM back-up state because the port P0 shares the falling edge detection circuit with port P1.
	Port $P3_0$/INT0	Return by an external “H” level or “L” level input. The EXF0 flag is not set.	Select the return level (“L” level or “H” level) with the bit 2 of register I1 according to the external state before going into the RAM back-up state.
	Port $P3_1$/INT1	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.

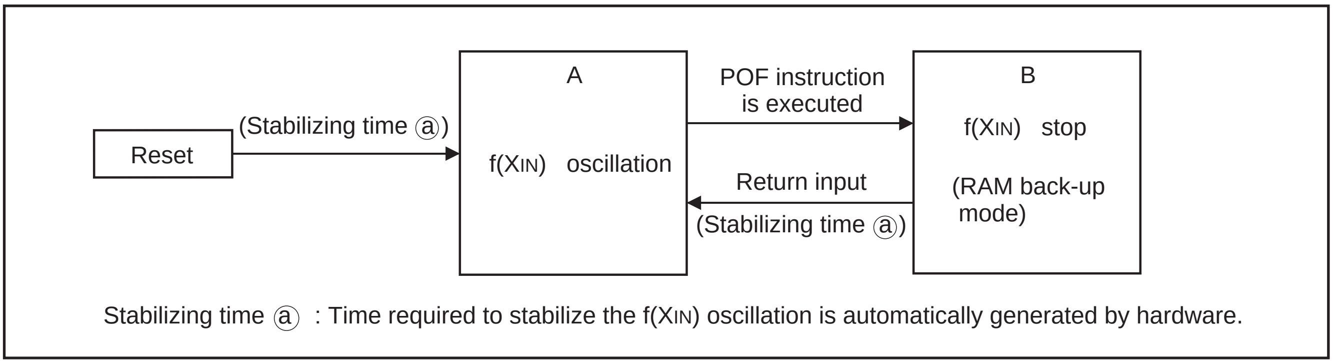


Fig. 38 State transition

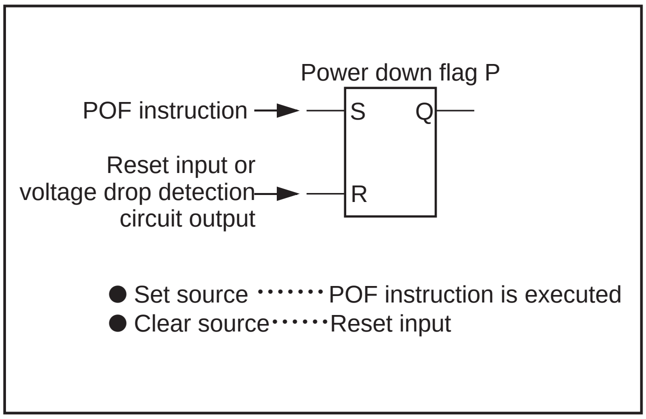


Fig. 39 Set source and clear source of the P flag

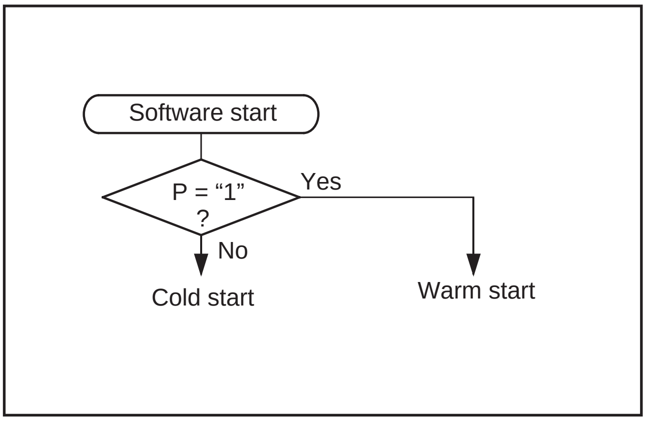


Fig. 40 Start condition identified example using the SNZP instruction

Table 22 Key-on wakeup control register, pull-up control register, and interrupt control register

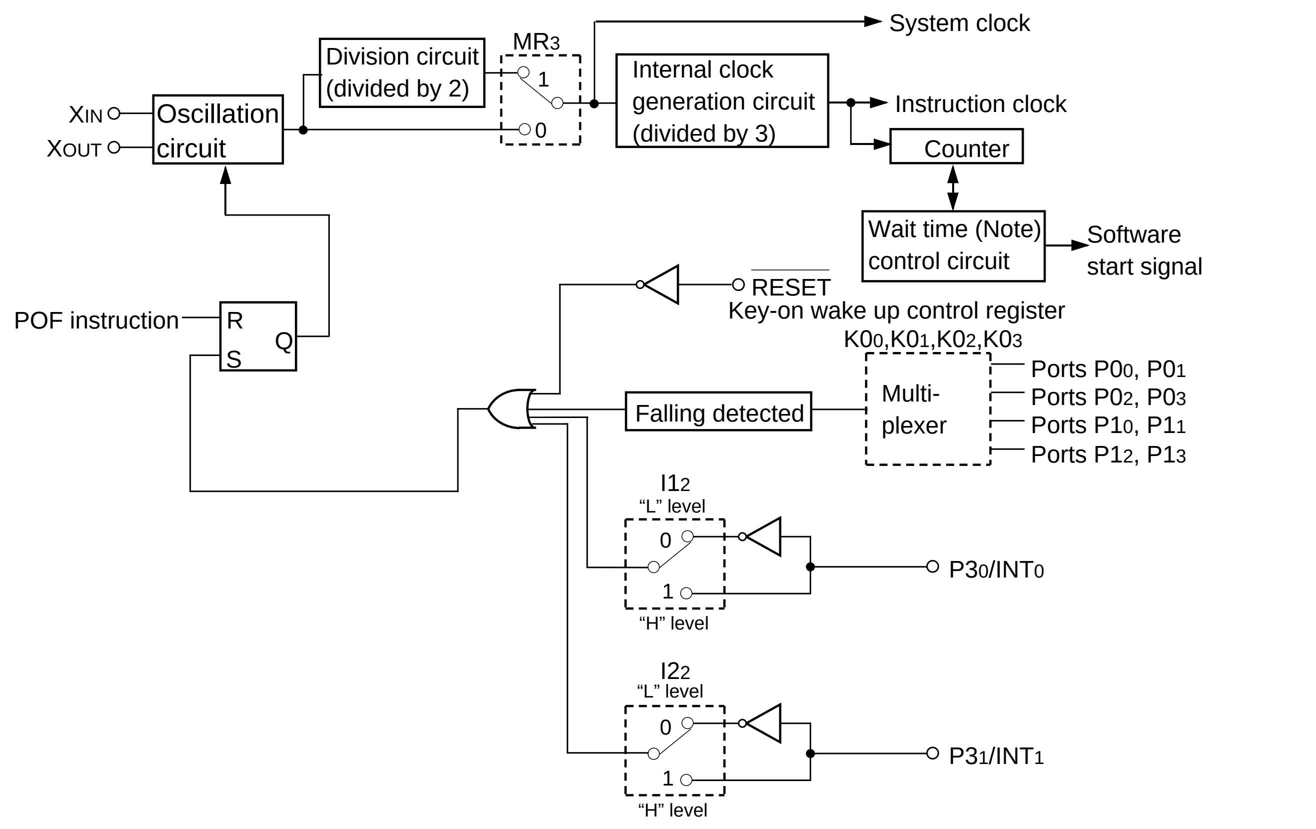
Key-on wakeup control register K0	at reset : 0000_2	at RAM back-up : state retained	R/W
$K0_3$	Pins $P1_2$ and $P1_3$ key-on wakeup control bit	0	Key-on wakeup not used
		1	Key-on wakeup used
$K0_2$	Pins $P1_0$ and $P1_1$ key-on wakeup control bit	0	Key-on wakeup not used
		1	Key-on wakeup used
$K0_1$	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. 41 Clock control circuit structure

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(X_{IN}) is obtained by externally connecting a ceramic resonator.
Connect this external circuit to pins X_{IN} and X_{OUT} at the shortest distance. A feedback resistor is built in between pins X_{IN} and X_{OUT}.
When an external clock signal is input, connect the clock source to X_{IN} and leave X_{OUT} open. When using an external clock, the maximum value of external clock oscillating frequency is shown in Table 24.

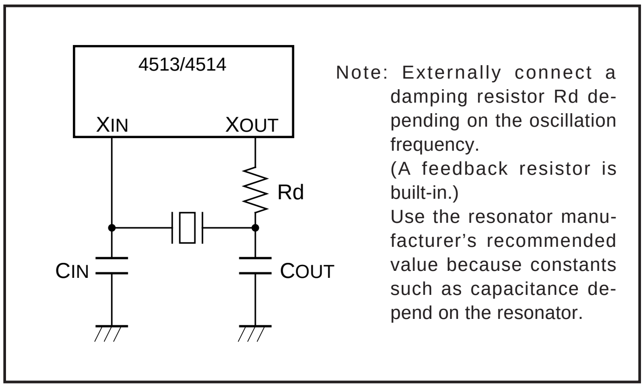


Fig. 42 Ceramic resonator external circuit

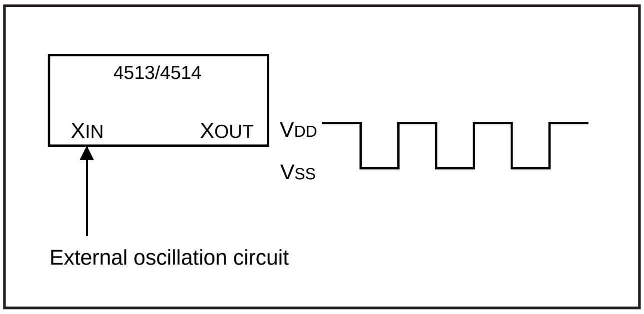


Fig. 43 External clock input circuit

Table 24 Maximum value of external clock oscillation frequency

		Supply voltage	Oscillation frequency (duty ratio)
Mask ROM version	Middle-speed mode	V_{DD} = 2.0 V to 5.5 V	3.0 MHz (40 % to 60 %)
	High-speed mode	V_{DD} = 4.0 V to 5.5 V	3.0 MHz (40 % to 60 %)
		V_{DD} = 2.5 V to 5.5 V	1.0 MHz (40 % to 60 %)
		V_{DD} = 2.0 V to 5.5 V	0.8 MHz (40 % to 60 %)
One Time PROM version	Middle-speed mode	V_{DD} = 2.5 V to 5.5 V	3.0 MHz (40 % to 60 %)
	High-speed mode	V_{DD} = 4.0 V to 5.5 V	3.0 MHz (40 % to 60 %)
		V_{DD} = 2.5 V to 5.5 V	1.0 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

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 VDD and VSS at the shortest distance,
- equalize its wiring in width and length, and
- use relatively thick wire.

In the One Time PROM version, CNVSS pin is also used as VPP pin. Accordingly, when using this pin, connect this pin to VSS through a resistor about 5 kΩ in series at the shortest distance.

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.

⑥P30/INT0 pin

When the interrupt valid waveform of the P30/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 P30/INT0 pin is changed with the bit 2 of register I1 (refer to Figure 44①).
- Depending on the input state of the P30/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 44②)

```
    ⋮
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. 44 External 0 interrupt program example

⑦P31/INT1 pin

When the interrupt valid waveform of P31/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 P31/INT1 pin is changed with the bit 2 of register I2 (refer to Figure 45③).
- Depending on the input state of the P31/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 45④).

```
    ⋮
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. 45 External 1 interrupt program example

⑧One Time PROM version

The operating power voltage of the One Time PROM version is 2.5 V to 5.5 V.

⑨Multifunction

The input of D6, D7, P20–P22, I/O of P30 and P31, input of CMP0-, CMP0+, CMP1-, CMP1+, and I/O of P40–P43 can be used even when CNTR0, CNTR1, SCK, SOUT, SIN, INT0, INT1, AIN0–AIN3 and AIN4–AIN7 are selected.

⑩ 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 46⑤).
- 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      ; (✕0✕✕2)
TV2A          ; The SNZAD instruction is valid ........⑤
LA     0      ; (0✕✕✕2)
TQ2A          ; Change of the operating mode of the A-D
                converter from the comparator mode to the
                A-D conversion mode
SNZAD
NOP
      ⋮
            ✕: this bit is not related to the change of the
               operating mode of the A-D conversion.
```

Fig. 46 A-D converter operating mode program example

⑪ 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 47).

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 48. In addition, test the application products sufficiently.

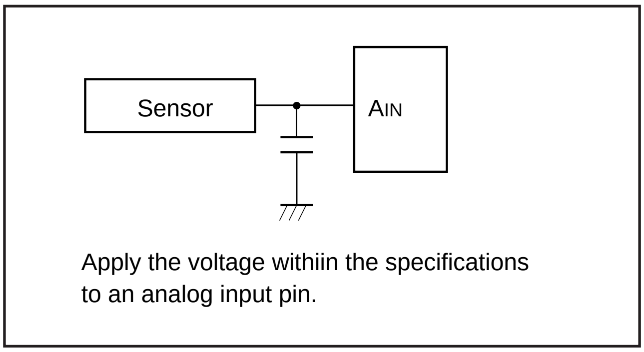


Fig. 47 Analog input external circuit example-1

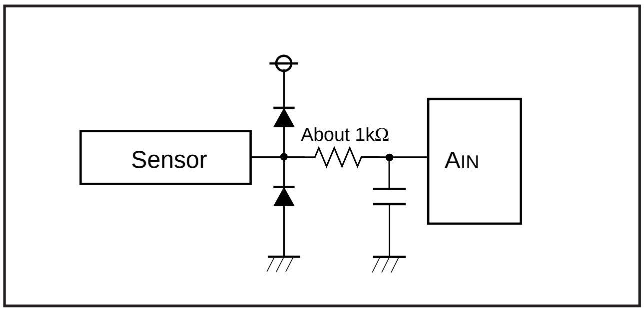


Fig. 48 Analog input external circuit example-2

⑫ 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.

⑬ Analog input pins

Note the following when using the analog input pins also for I/O port P4 functions:

- Even when $P4_0/A_{IN4}$–$P4_3/A_{IN7}$ are set to pins for analog input, they continue to function as $P4_0$–$P4_3$ I/O. Accordingly, when any of them are used as I/O port P4 and others are used as analog input pins, make sure to set the outputs of pins that are set for analog input to “1.” Also, the port input function of the pin functions as an analog input is undefined.
- TALA instruction

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

⑭ Program counter

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

⑮ Port P3

In the 4513 Group, when the IAP3 instruction is executed, note that the high-order 2 bits of register A is undefined.

⑯ $\boxed{\text{Voltage comparator function}}$

When the voltage comparator function is valid with the voltage comparator control register Q3, it is operating even in the RAM back-up mode. Accordingly, be careful about such state because it causes the increase of the operation current in the RAM back-up mode.
In order to reduce the operation current in the RAM back-up mode, invalidate (bits 2 and 3 of register Q3 = “0”) the voltage comparator function by software before the POF instruction is executed.
Also, while the voltage comparator function is valid, current is always consumed by voltage comparator. On the system required for the low-power dissipation, invalidate the voltage comparator when it is unused by software.

⑰ $\boxed{\text{Register Q3}}$

Bits 0 and 1 of register Q3 can be only read. Note that they cannot be written.

⑱ $\boxed{\text{Reading the comparison result of voltage comparator}}$

Read the voltage comparator comparison result from register Q3 after the voltage comparator response time (max. 20 μs) is passed from the voltage comparator function become valid.

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	Watchdog timer flag
Q2	A-D control register Q2 (4 bits)	WEF	Watchdog timer enable flag
Q3	Voltage comparator control register Q3 (4 bits)	INTE	Interrupt enable flag
AD	Successive comparison register AD (10 bits)	EXF0	External 0 interrupt request flag
J1	Serial I/O mode register J1 (4 bits)	EXF1	External 1 interrupt request flag
SI	Serial I/O register SI (8 bits)	P	Power down flag
V1	Interrupt control register V1 (4 bits)	ADF	A-D conversion completion flag
V2	Interrupt control register V2 (4 bits)	SIOF	Serial I/O transmission/reception completion flag
I1	Interrupt control register I1 (4 bits)		
I2	Interrupt control register I2 (4 bits)	D	Port D (8 bits)
W1	Timer control register W1 (4 bits)	P0	Port P0 (4 bits)
W2	Timer control register W2 (4 bits)	P1	Port P1 (4 bits)
W3	Timer control register W3 (4 bits)	P2	Port P2 (3 bits)
W4	Timer control register W4 (4 bits)	P3	Port P3 (4 bits)
W6	Timer control register W6 (4 bits)	P4	Port P4 (4 bits)
MR	Clock control register MR (4 bits)	P5	Port P5 (4 bits)
K0	Key-on wakeup control register K0 (4 bits)		
PU0	Pull-up control register PU0 (4 bits)	x	Hexadecimal variable
FR0	Direction register FR0 (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 4513/4514 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	$(E_7–E_4) \leftarrow (B)$ $(E_3–E_0) \leftarrow (A)$
	TABE	$(B) \leftarrow (E_7–E_4)$ $(A) \leftarrow (E_3–E_0)$
	TDA	$(DR_2–DR_0) \leftarrow (A_2–A_0)$
	TAD	$(A_2–A_0) \leftarrow (DR_2–DR_0)$ $(A_3) \leftarrow 0$
	TAZ	$(A_1, A_0) \leftarrow (Z_1, Z_0)$ $(A_3, A_2) \leftarrow 0$
	TAX	(A) ← (X)
	TASP	$(A_2–A_0) \leftarrow (SP_2–SP_0)$ $(A_3) \leftarrow 0$
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) $(PC_H) \leftarrow p$ $(PC_L) \leftarrow (DR_2–DR_0, A_3–A_0)$ $(B) \leftarrow (ROM(PC))_{7–4}$ $(A) \leftarrow (ROM(PC))_{3–0}$ (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	$(A) \leftarrow (\overline{A})$
	RAR	$\rightarrow \boxed{CY} \rightarrow \boxed{A_3A_2A_1A_0}$
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	$(PC_L) \leftarrow a_6–a_0$
	BL p, a	$(PC_H) \leftarrow p$ $(PC_L) \leftarrow a_6–a_0$
	BLA p	$(PC_H) \leftarrow p$ $(PC_L) \leftarrow (DR_2–DR_0, A_3–A_0)$
Subroutine operation	BM a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) $(PC_H) \leftarrow 2$ $(PC_L) \leftarrow a_6–a_0$
	BML p, a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) $(PC_H) \leftarrow p$ $(PC_L) \leftarrow a_6–a_0$
	BMLA p	(SP) ← (SP) + 1 (SK(SP)) ← (PC) $(PC_H) \leftarrow p$ $(PC_L) \leftarrow (DR_2–DR_0, A_3–A_0)$
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)
	TAW6	(A) ← (W6)
	TW6A	(W6) ← (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
	IAP3	(A) ← (P3)
	OP3A	(P3) ← (A)
	IAP4*	(A) ← (P4)
	OP4A*	(P4) ← (A)
	IAP5*	(A) ← (P5)
	OP5A*	(P5) ← (A)
	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

*: The 4513 Group does not have these instructions.

LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function
Input/Output operation	TK0A	(K0) ← (A)
	TAK0	(A) ← (K0)
	TPU0A	(PU0) ← (A)
	TAPU0	(A) ← (PU0)
	TFR0A*	(FR0) ← (A)
Serial I/O control operation	TABSI	(A) ← (SI_3–SI_0) (B) ← (SI_7–SI_4)
	TSIAB	(SI_3–SI_0) ← (A) (SI_7–SI_4) ← (B)
	TAJ1	(A) ← (J1)
	TJ1A	(J1) ← (A)
	SST	(SIOF) ← 0 Serial I/O starting
	SNZSI	(SIOF) = 1 ? After skipping (SIOF) ← 0
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)
Other operation	NOP	(PC) ← (PC) + 1
	POF	RAM back-up
	EPOF	POF instruction valid
	SNZP	(P) = 1 ?
	WRST	(WDF1) ← 0, (WEF) ← 1
	TAMR	(A) ← (MR)
	TMRA	(MR) ← (A)
	TAQ3	(A) ← (Q3)
	TQ3A	($Q3_3$, $Q3_2$) ← (A_3, A_2) ($Q3_1$) ← (CMP1 comparison result) ($Q3_0$) ← (CMP0 comparison result)

*: The 4513 Group does not have these instructions.

INSTRUCTION CODE TABLE (for 4513 Group)

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

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

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

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

- *, **, and *** cannot be used in the M34513M2-XXXSP/FP.
- * and ** cannot be used in the M34513M4-XXXSP/FP.
- * and ** cannot be used in the M34513E4FP.
- * cannot be used in the M34513M6-XXXFP.

INSTRUCTION CODE TABLE (continued) (for 4513 Group)

D_3–D_0 \ D_9–D_4		100000	100001	100010	100011	100100	100101	100110	100111	101000	101001	101010	101011	101100	101101	101110	101111	110000 111111
	Hex. notation	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30–3F
0000	0	–	TW3A	OP0A	T1AB	–	TAW6	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	–	TW6A	OP3A	T4AB	–	TAI1	IAP3	TAB4	SNZT4	–	–	TMA 3	TAM 3	XAM 3	XAMI 3	XAMD 3	LXY
0100	4	TQ1A	–	–	–	TAQ1	TAI2	–	–	–	–	–	TMA 4	TAM 4	XAM 4	XAMI 4	XAMD 4	LXY
0101	5	TQ2A	–	–	–	TAQ2	–	–	–	–	–	–	TMA 5	TAM 5	XAM 5	XAMI 5	XAMD 5	LXY
0110	6	TQ3A	TMRA	–	–	TAQ3	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. D_3–D_0 show the low-order 4 bits of the machine language code, and D_9–D_4 show the high-order 6 bits of the machine language code. The hexadecimal representation of the code is also provided. There are one-word instructions and two-word instructions, but only the first word of each instruction is shown. Do not use code marked "–."

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

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

INSTRUCTION CODE TABLE (for 4514 Group)

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

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

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

	The second word
BL	10 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 M34514M6-XXXFP.

INSTRUCTION CODE TABLE (continued) (for 4514 Group)

D3–D0 \ D9–D4		100000	100001	100010	100011	100100	100101	100110	100111	101000	101001	101010	101011	101100	101101	101110	101111	110000 111111
	Hex. notation	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30–3F
0000	0	–	TW3A	OP0A	T1AB	–	TAW6	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	–	TW6A	OP3A	T4AB	–	TAI1	IAP3	TAB4	SNZT4	–	–	TMA 3	TAM 3	XAM 3	XAMI 3	XAMD 3	LXY
0100	4	TQ1A	–	OP4A	–	TAQ1	TAI2	IAP4	–	–	–	–	TMA 4	TAM 4	XAM 4	XAMI 4	XAMD 4	LXY
0101	5	TQ2A	–	OP5A	–	TAQ2	–	IAP5	–	–	–	–	TMA 5	TAM 5	XAM 5	XAMI 5	XAMD 5	LXY
0110	6	TQ3A	TMRA	–	–	TAQ3	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	TFR0A	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. D_3–D_0 show the low-order 4 bits of the machine language code, and D_9–D_4 show the high-order 6 bits of the machine language code. The hexadecimal representation of the code is also provided. There are one-word instructions and two-word instructions, but only the first word of each instruction is shown. Do not use code marked "–."

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

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

MACHINE INSTRUCTIONS

Type of instructions \ Parameter	Mnemonic	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation	Number of words	Number of cycles	Function
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	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	→ $\boxed{CY}$ → $\boxed{A_3A_2A_1A_0}$
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	0 0	0 1	1 1	0 1	0 n	1 n	0 n	1 n	0 2 5 0 7 n	2	2	(A) = n ? n = 0 to 15

Note : p is 0 to 15 for M34513M2, p is 0 to 31 for M34513M4/E4, p is 0 to 47 for M34513M6 and M34514M6, and p is 0 to 63 for M34513M8/E8 and M34514M8/E8.

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

MACHINE INSTRUCTIONS (continued)

Type of instructions / Parameter	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	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow p$ $(PC_L) \leftarrow a_6–a_0$ (Note)
		1	0	p_5	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2 p a +a			
	BMLA p	0	0	0	0	1	1	0	0	0	0	0 3 0	2	2	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow p$ $(PC_L) \leftarrow (DR_2–DR_0,A_3–A_0)$ (Note)
		1	0	p_5	p_4	0	0	p_3	p_2	p_1	p_0	2 p p			
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 M34513M2, p is 0 to 31 for M34513M4/E4, p is 0 to 47 for M34513M6 and M34514M6, and p is 0 to 63 for M34513M8/E8 and M34514M8/E8.

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)
	TAW6	1	0	0	1	0	1	0	0	0	0	2 5 0	1	1	(A) ← (W6)
	TW6A	1	0	0	0	0	1	0	0	1	1	2 1 3	1	1	(W6) ← (A)

Skip condition	Carry flag CY	Datailed description
(INT0) = “H” However, $I1_2$ = 1	–	When bit 2 ($I1_2$) of register I1 is “1” : Skips the next instruction when the level of INT0 pin is “H.”
(INT0) = “L” However, $I1_2$ = 0	–	When bit 2 ($I1_2$) of register I1 is “0” : Skips the next instruction when the level of INT0 pin is “L.”
(INT1) = “H” However, $I2_2$ = 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 control register W6 to register A.
–	–	Transfers the contents of register A to timer control register W6.

MACHINE INSTRUCTIONS (continued)

Type of instructions	Mnemonic	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
Timer operation	TAB1	1	0	0	1	1	1	0	0	0	0	2 7 0	1	1	$(B) \leftarrow (T1_7–T1_4)$ $(A) \leftarrow (T1_3–T1_0)$
	T1AB	1	0	0	0	1	1	0	0	0	0	2 3 0	1	1	$(R1_7–R1_4) \leftarrow (B)$ $(T1_7–T1_4) \leftarrow (B)$ $(R1_3–R1_0) \leftarrow (A)$ $(T1_3–T1_0) \leftarrow (A)$
	TAB2	1	0	0	1	1	1	0	0	0	1	2 7 1	1	1	$(B) \leftarrow (T2_7–T2_4)$ $(A) \leftarrow (T2_3–T2_0)$
	T2AB	1	0	0	0	1	1	0	0	0	1	2 3 1	1	1	$(R2_7–R2_4) \leftarrow (B)$ $(T2_7–T2_4) \leftarrow (B)$ $(R2_3–R2_0) \leftarrow (A)$ $(T2_3–T2_0) \leftarrow (A)$
	TAB3	1	0	0	1	1	1	0	0	1	0	2 7 2	1	1	$(B) \leftarrow (T3_7–T3_4)$ $(A) \leftarrow (T3_3–T3_0)$
	T3AB	1	0	0	0	1	1	0	0	1	0	2 3 2	1	1	$(R3_7–R3_4) \leftarrow (B)$ $(T3_7–T3_4) \leftarrow (B)$ $(R3_3–R3_0) \leftarrow (A)$ $(T3_3–T3_0) \leftarrow (A)$
	TAB4	1	0	0	1	1	1	0	0	1	1	2 7 3	1	1	$(B) \leftarrow (T4_7–T4_4)$ $(A) \leftarrow (T4_3–T4_0)$
	T4AB	1	0	0	0	1	1	0	0	1	1	2 3 3	1	1	$(R4_7–R4_4) \leftarrow (B)$ $(T4_7–T4_4) \leftarrow (B)$ $(R4_3–R4_0) \leftarrow (A)$ $(T4_3–T4_0) \leftarrow (A)$
	TR1AB	1	0	0	0	1	1	1	1	1	1	2 3 F	1	1	$(R1_7–R1_4) \leftarrow (B)$ $(R1_3–R1_0) \leftarrow (A)$
	TR3AB	1	0	0	0	1	1	1	0	1	1	2 3 B	1	1	$(R3_7–R3_4) \leftarrow (B)$ $(R3_3–R3_0) \leftarrow (A)$
	SNZT1	1	0	1	0	0	0	0	0	0	0	2 8 0	1	1	(T1F) = 1 ? After skipping $(T1F) \leftarrow 0$
	SNZT2	1	0	1	0	0	0	0	0	0	1	2 8 1	1	1	(T2F) = 1 ? After skipping $(T2F) \leftarrow 0$
	SNZT3	1	0	1	0	0	0	0	0	1	0	2 8 2	1	1	(T3F) = 1 ? After skipping $(T3F) \leftarrow 0$
	SNZT4	1	0	1	0	0	0	0	0	1	1	2 8 3	1	1	(T4F) = 1 ? After skipping $(T4F) \leftarrow 0$

Skip condition	Carry flag CY	Datailed description
–	–	Transfers the contents of timer 1 to registers A and B.
–	–	Transfers the contents of registers A and B to timer 1 and timer 1 reload register.
–	–	Transfers the contents of timer 2 to registers A and B.
–	–	Transfers the contents of registers A and B to timer 2 and timer 2 reload register.
–	–	Transfers the contents of timer 3 to registers A and B.
–	–	Transfers the contents of registers A and B to timer 3 and timer 3 reload register.
–	–	Transfers the contents of timer 4 to registers A and B.
–	–	Transfers the contents of registers A and B to timer 4 and timer 4 reload register.
–	–	Transfers the contents of registers A and B to timer 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)

Type of instructions	Mnemonic	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
Input/Output operation	IAP0	1	0	0	1	1	0	0	0	0	0	2 6 0	1	1	(A) ← (P0)
	OP0A	1	0	0	0	1	0	0	0	0	0	2 2 0	1	1	(P0) ← (A)
	IAP1	1	0	0	1	1	0	0	0	0	1	2 6 1	1	1	(A) ← (P1)
	OP1A	1	0	0	0	1	0	0	0	0	1	2 2 1	1	1	(P1) ← (A)
	IAP2	1	0	0	1	1	0	0	0	1	0	2 6 2	1	1	$(A_2–A_0) \leftarrow (P2_2–P2_0)$ $(A_3) \leftarrow 0$
	IAP3	1	0	0	1	1	0	0	0	1	1	2 6 3	1	1	(A) ← (P3)
	OP3A	1	0	0	0	1	0	0	0	1	1	2 2 3	1	1	(P3) ← (A)
	IAP4*	1	0	0	1	1	0	0	1	0	0	2 6 4	1	1	(A) ← (P4)
	OP4A*	1	0	0	0	1	0	0	1	0	0	2 2 4	1	1	(P4) ← (A)
	IAP5*	1	0	0	1	1	0	0	1	0	1	2 6 5	1	1	(A) ← (P5)
	OP5A*	1	0	0	0	1	0	0	1	0	1	2 2 5	1	1	(P5) ← (A)
	CLD	0	0	0	0	0	1	0	0	0	1	0 1 1	1	1	(D) ← 1
	RD	0	0	0	0	0	1	0	1	0	0	0 1 4	1	1	(D(Y)) ← 0 (Y) = 0 to 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)
	TFR0A*	1	0	0	0	1	0	1	0	0	0	2 2 8	1	1	(FR0) ← (A)

*: The 4513 Group does not have these instructions.

Skip condition	Carry flag CY	Datailed description
–	–	Transfers the input of port P0 to register A.
–	–	Outputs the contents of register A to port P0.
–	–	Transfers the input of port P1 to register A.
–	–	Outputs the contents of register A to port P1.
–	–	Transfers the input of port P2 to register A.
–	–	Transfers the input of port P3 to register A.
–	–	Outputs the contents of register A to port P3.
–	–	Transfers the input of port P4 to register A.
–	–	Outputs the contents of register A to port P4.
–	–	Transfers the input of port P5 to register A.
–	–	Outputs the contents of register A to port P5.
–	–	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 register A to direction register FR0.

MACHINE INSTRUCTIONS (continued)

Type of instructions	Mnemonic	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
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
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)
	TAQ3	1	0	0	1	0	0	0	1	1	0	2 4 6	1	1	(A) ← (Q3)
	TQ3A	1	0	0	0	0	0	0	1	1	0	2 0 6	1	1	($Q3_3$, $Q3_2$) ← (A_3, A_2) ($Q3_1$) ← (CMP1 comparison result) ($Q3_0$) ← (CMP0 comparison result)

Skip condition	Carry flag CY	Datailed description
–	–	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.
–	–	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.
–	–	Transfers the contents of the voltage comparator control register Q3 to register A.
–	–	Transfers the contents of the high-order 2 bits of register A to the high-order 2 bits of voltage comparator control register Q3, and the comparison result of the voltage comparator is transferred to the low-order 2 bits of the register Q3.

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.

<table>
<tr><td colspan="2">Timer control register W1</td><td colspan="2">at reset : 0000_2</td><td>at RAM back-up : 0000_2</td><td>R/W</td></tr>
<tr><td rowspan="2">$W1_3$</td><td rowspan="2">Prescaler control bit</td><td colspan="2">0</td><td colspan="2">Stop (state initialized)</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Operating</td></tr>
<tr><td rowspan="2">$W1_2$</td><td rowspan="2">Prescaler dividing ratio selection bit</td><td colspan="2">0</td><td colspan="2">Instruction clock divided by 4</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Instruction clock divided by 16</td></tr>
<tr><td rowspan="2">$W1_1$</td><td rowspan="2">Timer 1 control bit</td><td colspan="2">0</td><td colspan="2">Stop (state retained)</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Operating</td></tr>
<tr><td rowspan="2">$W1_0$</td><td rowspan="2">Timer 1 count start synchronous circuit control bit</td><td colspan="2">0</td><td colspan="2">Count start synchronous circuit not selected</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Count start synchronous circuit selected</td></tr>
<tr><td colspan="2">Timer control register W2</td><td colspan="2">at reset : 0000_2</td><td>at RAM back-up : state retained</td><td>R/W</td></tr>
<tr><td rowspan="2">$W2_3$</td><td rowspan="2">Timer 2 control bit</td><td colspan="2">0</td><td colspan="2">Stop (state retained)</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Operating</td></tr>
<tr><td rowspan="2">$W2_2$</td><td rowspan="2">Not used</td><td colspan="2">0</td><td colspan="2" rowspan="2">This bit has no function, but read/write is enabled.</td></tr>
<tr><td colspan="2">1</td></tr>
<tr><td rowspan="2">$W2_1$</td><td rowspan="5">Timer 2 count source selection bits</td><td>$W2_1$</td><td>$W2_0$</td><td colspan="2">Count source</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">Timer 1 underflow signal</td></tr>
<tr><td rowspan="3">$W2_0$</td><td>0</td><td>1</td><td colspan="2">Prescaler output</td></tr>
<tr><td>1</td><td>0</td><td colspan="2">CNTR0 input</td></tr>
<tr><td>1</td><td>1</td><td colspan="2">16 bit timer (WDT) underflow signal</td></tr>
<tr><td colspan="2">Timer control register W3</td><td colspan="2">at reset : 0000_2</td><td>at RAM back-up : state retained</td><td>R/W</td></tr>
<tr><td rowspan="2">$W3_3$</td><td rowspan="2">Timer 3 control bit</td><td colspan="2">0</td><td colspan="2">Stop (state retained)</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Operating</td></tr>
<tr><td rowspan="2">$W3_2$</td><td rowspan="2">Timer 3 count start synchronous circuit control bit</td><td colspan="2">0</td><td colspan="2">Count start synchronous circuit not selected</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Count start synchronous circuit selected</td></tr>
<tr><td rowspan="2">$W3_1$</td><td rowspan="5">Timer 3 count source selection bits</td><td>$W3_1$</td><td>$W3_0$</td><td colspan="2">Count source</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">Timer 2 underflow signal</td></tr>
<tr><td rowspan="3">$W3_0$</td><td>0</td><td>1</td><td colspan="2">Prescaler output</td></tr>
<tr><td>1</td><td>0</td><td colspan="2">Not available</td></tr>
<tr><td>1</td><td>1</td><td colspan="2">Not available</td></tr>
<tr><td colspan="2">Timer control register W4</td><td colspan="2">at reset : 0000_2</td><td>at RAM back-up : state retained</td><td>R/W</td></tr>
<tr><td rowspan="2">$W4_3$</td><td rowspan="2">Timer 4 control bit</td><td colspan="2">0</td><td colspan="2">Stop (state retained)</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Operating</td></tr>
<tr><td rowspan="2">$W4_2$</td><td rowspan="2">Not used</td><td colspan="2">0</td><td colspan="2" rowspan="2">This bit has no function, but read/write is enabled.</td></tr>
<tr><td colspan="2">1</td></tr>
<tr><td rowspan="2">$W4_1$</td><td rowspan="5">Timer 4 count source selection bits</td><td>$W4_1$</td><td>$W4_0$</td><td colspan="2">Count source</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">Timer 3 underflow signal</td></tr>
<tr><td rowspan="3">$W4_0$</td><td>0</td><td>1</td><td colspan="2">Prescaler output</td></tr>
<tr><td>1</td><td>0</td><td colspan="2">CNTR1 input</td></tr>
<tr><td>1</td><td>1</td><td colspan="2">Not available</td></tr>
<tr><td colspan="2">Timer control register W6</td><td colspan="2">at reset : 0000_2</td><td>at RAM back-up : state retained</td><td>R/W</td></tr>
<tr><td rowspan="2">$W6_3$</td><td rowspan="2">CNTR1 output control bit</td><td colspan="2">0</td><td colspan="2">Timer 3 underflow signal output divided by 2</td></tr>
<tr><td colspan="2">1</td><td colspan="2">CNTR1 output control by timer 4 underflow signal divided by 2</td></tr>
<tr><td rowspan="2">$W6_2$</td><td rowspan="2">D_7/CNTR1 function selection bit</td><td colspan="2">0</td><td colspan="2">D_7(I/O)/CNTR1 input</td></tr>
<tr><td colspan="2">1</td><td colspan="2">CNTR1 (I/O)/D_7(input)</td></tr>
<tr><td rowspan="2">$W6_1$</td><td rowspan="2">CNTR0 output control bit</td><td colspan="2">0</td><td colspan="2">Timer 1 underflow signal output divided by 2</td></tr>
<tr><td colspan="2">1</td><td colspan="2">CNTR0 output control by timer 2 underflow signal divided by 2</td></tr>
<tr><td rowspan="2">$W6_0$</td><td rowspan="2">D_6/CNTR0 output control bit</td><td colspan="2">0</td><td colspan="2">D_6(I/O)/CNTR0 input</td></tr>
<tr><td colspan="2">1</td><td colspan="2">CNTR0 (I/O)/D_6(input)</td></tr>
</table>

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

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 S_{CK}, S_{OUT}, S_{IN}/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
$Q1_3$	Note used	0			This bit has no function, but read/write is enabled.	
		1				
		$Q1_2$	$Q1_1$	$Q1_0$	Selected pins	
$Q1_2$	Analog input pin selection bits (Note 2)	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	A_{IN4} (Not available for the 4513 Group)	
$Q1_0$		1	0	1	A_{IN5} (Not available for the 4513 Group)	
		1	1	0	A_{IN6} (Not available for the 4513 Group)	
		1	1	1	A_{IN7} (Not available for the 4513 Group)	

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	
$Q2_2$	$P4_3$/A_{IN7} and $P4_2$/A_{IN6} pin function selection bit (Not used for the 4513 Group)	0	$P4_3$, $P4_2$ (read/write enabled for the 4513 Group)	
		1	A_{IN7}, A_{IN6}/$P4_3$, $P4_2$ (read/write enabled for the 4513 Group)	
$Q2_1$	$P4_1$/A_{IN5} pin function selection bit (Not used for the 4513 Group)	0	$P4_1$ (read/write enabled for the 4513 Group)	
		1	A_{IN5}/$P4_1$ (read/write enabled for the 4513 Group)	
$Q2_0$	$P4_0$/A_{IN4} pin function selection bit (Not used for the 4513 Group)	0	$P4_0$ (read/write enabled for the 4513 Group)	
		1	A_{IN4}/$P4_0$ (read/write enabled for the 4513 Group)	

Comparator control register Q3 (Note 3)		at reset : 0000_2	at RAM back-up : state retained	R/W
$Q3_3$	Voltage comparator (CMP1) control bit	0	Voltage comparator (CMP1) invalid	
		1	Voltage comparator (CMP1) valid	
$Q3_2$	Voltage comparator (CMP0) control bit	0	Voltage comparator (CMP0) invalid	
		1	Voltage comparator (CMP0) valid	
$Q3_1$	CMP1 comparison result store bit	0	CMP1- > CMP1+	
		1	CMP1- < CMP1+	
$Q3_0$	CMP0 comparison reslut store bit	0	CMP0- > CMP0+	
		1	CMP0- < CMP0+	

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		

Notes 1: "R" represents read enabled, "W" represents write enabled.
2: Select A_{IN4}–A_{IN7} with register Q1 after setting register Q2.
3: Bits 0 and 1 of register Q3 can be only read.

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	
Direction register FR0 (Note 2)		at reset : 0000_2	at RAM back-up : state retained	W
$FR0_3$	Port $P5_3$ input/output control bit	0	Port $P5_3$ input	
		1	Port $P5_3$ output	
$FR0_2$	Port $P5_2$ input/output control bit	0	Port $P5_2$ input	
		1	Port $P5_2$ output	
$FR0_1$	Port $P5_1$ input/output control bit	0	Port $P5_1$ input	
		1	Port $P5_1$ output	
$FR0_0$	Port $P5_0$ input/output control bit	0	Port $P5_0$ input	
		1	Port $P5_0$ output	

Notes 1: “R” represents read enabled, and “W” represents write enabled.
2: The 4513 Group does not have the direction register FR0.

BUILT-IN PROM VERSION

In addition to the mask ROM versions, the 4513/4514 Group has programmable ROM version software compatible with mask ROM. The built-in PROM of One Time PROM version can be written to and not be erased.
The built-in PROM versions have functions similar to those of the mask ROM versions, but they have PROM mode that enables writing to built-in PROM.
Table 25 shows the product of built-in PROM version. Figure 49 and 50 show the pin configurations of built-in PROM versions.

Table 25 Product of built-in PROM version

Product	PROM size (× 10 bits)	RAM size (× 4 bits)	Package	ROM type
M34513E4SP/FP	4096 words	256 words	SP: 32P4B FP: 32P6B-A	One Time PROM version [shipped in blank]
M34513E8FP	8192 words	384 words	32P6B-A	
M34514E8FP	8192 words	384 words	42P2R-A	

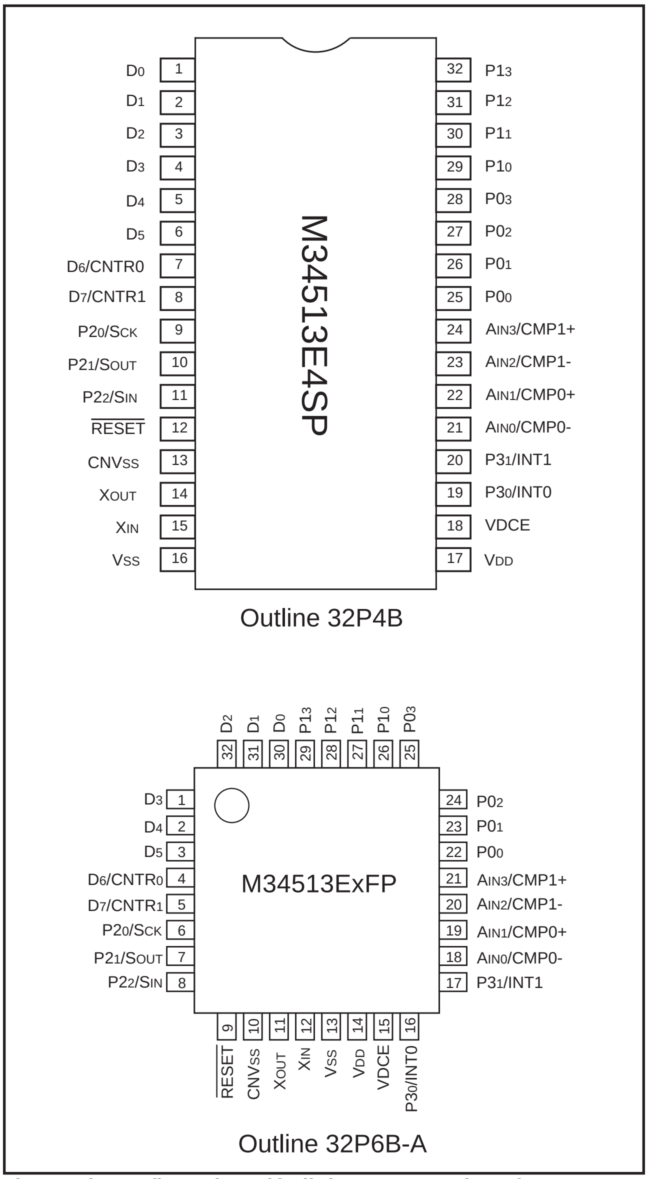


Fig. 49 Pin configuration of built-in PROM version of 4513 Group

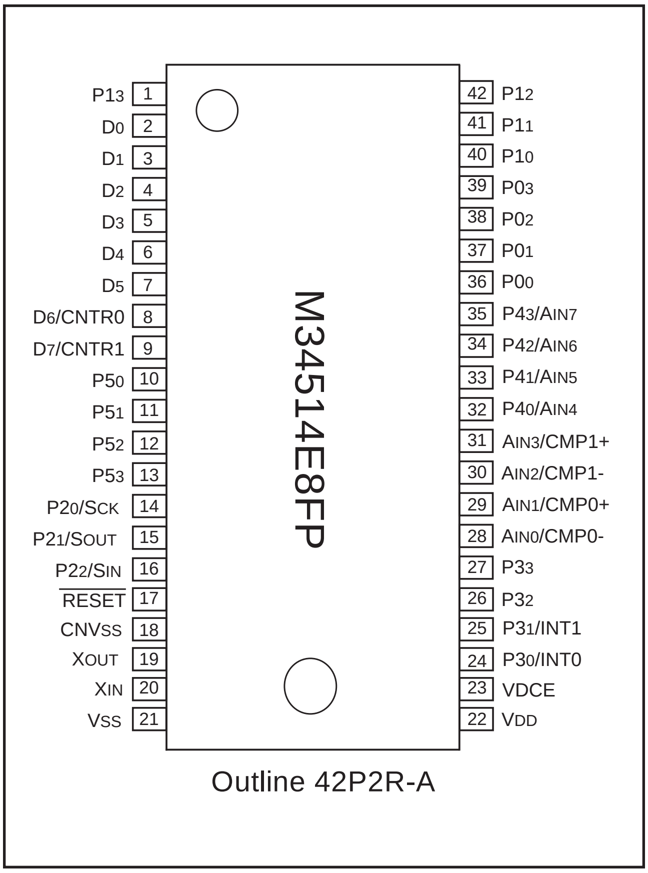


Fig. 50 Pin configuration of built-in PROM version of 4514 Group

(1) PROM mode

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

In the PROM mode, the programming adapter can be used with a general-purpose PROM programmer to write to or read from the built-in PROM as if it were M5M27C256K. Programming adapters are listed in Table 26.Contact addresses at the end of this sheet for the appropriate PROM programmer.

- Writing and reading of built-in PROM
  Programming voltage is 12.5 V. Write the program in the PROM of the built-in PROM version as shown in Figure 51.

(2) Notes on handling

①A high-voltage is used for writing. Take care that overvoltage is not applied. Take care especially at turning on the power.

②For the One Time PROM version shipped in blank, Mitsubishi Electric corp. does not perform PROM writing test and screening in the assembly process and following processes. In order to improve reliability after writing, performing writing and test according to the flow shown in Figure 52 before using is recommended (Products shipped in blank: PROM contents is not written in factory when shipped).

Table 26 Programming adapters

Microcomputer	Programming adapter
M34513E4SP	PCA7442SP
M34513E4FP, M34513E8FP	PCA7442FP
M34514E8FP	PCA7441

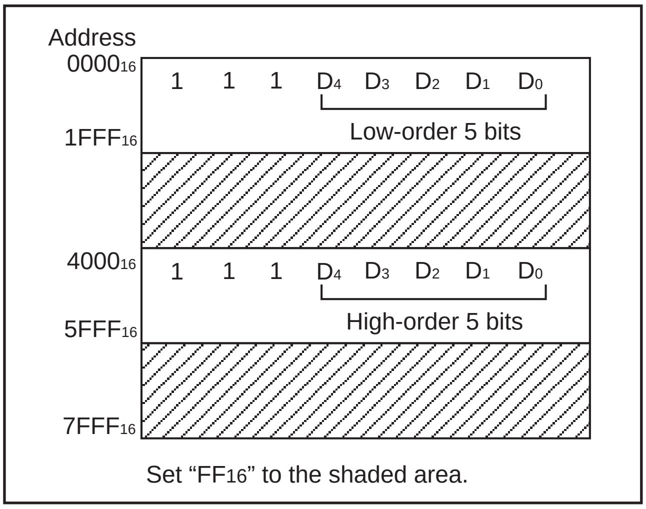


Fig. 51 PROM memory map

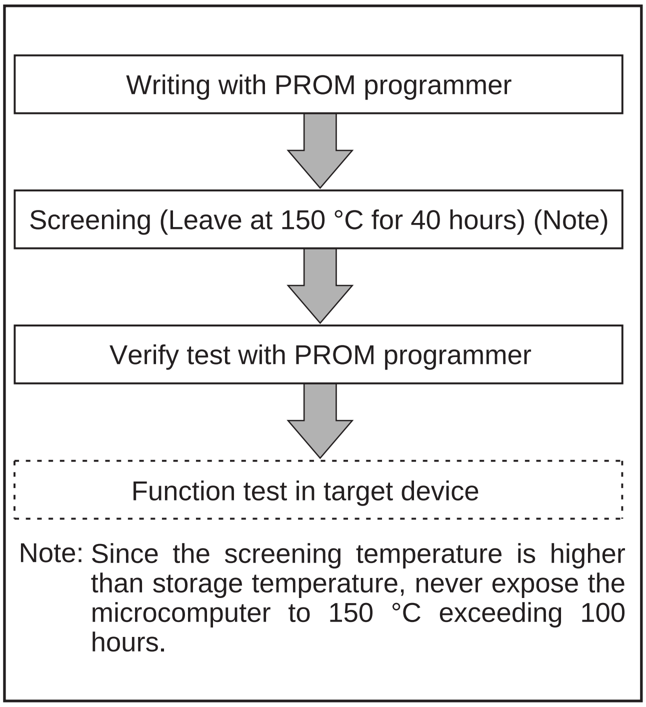


Fig. 52 Flow of writing and test of the product shipped in blank

CHAPTER 2
APPLICATION

2.1 I/O pins

The 4513/4514 Group has the twenty-eight I/O pins (eighteen I/O pins for 4513 Group), three input pins. (Ports $P2_0$–$P2_2$, $P3_0$, $P3_1$, D_6 and D_7 are also used as serial I/O pins SCK, SOUT, SIN, and INT0, INT1, CNTR0 and CNTR1 pins, respectively).
This section describes each port I/O function, related registers, application example using each port function and notes.

2.1.1 I/O ports

(1) Port P0

Port P0 is a 4-bit I/O port.
Port P0 has the key-on wakeup function which turns ON/OFF with register K0 and pull-up transistor which turns ON/OFF with register PU0.

■ **Input/output of port P0**

● Data input to port P0
Set the output latch of specified port P0i (i=0 to 3) to “1” with the **OP0A** instruction. If the output latch is set to “0,” “L” level is input.
The state of port P0 is transferred to register A when the **IAP0** instruction is executed.

● Data output from port P0
The contents of register A is output to port P0 with the **OP0A** instruction.
The output structure is an N-channel open-drain.

(2) Port P1

Port P1 is a 4-bit I/O port.
Port P1 has the key-on wakeup function which turns ON/OFF with register K0 and pull-up transistor which turns ON/OFF with register PU0.

■ **Input/output of port P1**

● Data input to port P1
Set the output latch of specified port P1i (i=0 to 3) to “1” with the **OP1A** instruction. If the output latch is set to “0,” “L” level is input.
The state of port P1 is transferred to register A when the **IAP1** instruction is executed.

● Data output from port P1
The contents of register A is output to port P1 with the **OP1A** instruction.
The output structure is an N-channel open-drain.

(3) Port P2

Port P2 is a 3-bit input port.

■ **Input of port P2**

● Data input to port P2
The state of port P2 is transferred to register A when the **IAP2** instruction is executed. However, port P2 is 3 bits and A_3 is fixed to “0.”

(4) Port P3

Port P3 is a 4-bit I/O port for the 4514 Group, and a 2-bit I/O port for the 4513 Group.

■ **Input/output of port P3**

● Data input to port P3

Set the output latch of specified port P3i (i=0 to 3) to "1" with the **OP3A** instruction. If the output latch is set to "0," "L" level is input.
The state of port P3 is transferred to register A when the **IAP3** instruction is executed.
However, A_2 and A_3 are undefined in the 4513 Group.

● Data output from port P3

The contents of register A is output to port P3 with the **OP3A** instruction.
The output structure is an N-channel open-drain.

(5) Port P4 (The 4513 Group does not have this port.)

Port P4 is a 4-bit I/O port.

■ **Input/output of port P4**

Ports $P4_0$–$P4_3$ are also used as A_{IN4}–A_{IN7}. Therefore, when $P4_0/A_{IN4}$–$P4_3/A_{IN7}$ are used as port P4, set corresponding bits of A-D control register Q2 to "0".

● Data input to port P4

Set the output latch of specified port P4i (i=0 to 3) to "1" with the **OP4A** instruction. If the output latch is set to "0," "L" level is input.
The state of port P4 is transferred to register A when the **IAP4** instruction is executed.

● Data output from port P4

The contents of register A is output to port P4 with the **OP4A** instruction.
The output structure is an N-channel open-drain.

(6) Port P5 (The 4513 Group does not have this port.)

Port P5 is a 4-bit I/O port.

■ **Input/output of port P5**

Port P5 has direction register FR0 to input/output by the bit.

● Data input to port P5

Set the bit of register FR0i(i=0 to 3) corresponding to specified port P5i (i=0 to 3) to "0." When the register FR0 is set to "1," the value of output latch is input.
The state of port P5 is transferred to register A when the **IAP5** instruction is executed.

● Data output from port P5

Set the bit of register FR0i(i=0 to 3) corresponding to specified port P5i (i=0 to 3) to "1." When the register FR0 is set to "0," specified port P5i is in the high-impedance state.
The contents of register A is output to port P5 with the **OP5A** instruction.
The output structure is CMOS.

(7) Port D

D_0–D_7 are eight independent I/O ports.

■ **Input/output of port D**

Each pin of port D has an independent 1-bit wide I/O function. For I/O of ports D_0–D_7, select one of port D with the register Y of the data pointer first.

● Data input to port D

Set the output latch of specified port Di (i = 0 to 7) to “1” with the **SD** instruction.
When the output latch is set to “0,” “L” level is input.
When the **SZD** instruction is executed, if the port specified by register Y is “0,” the next instruction is skipped. If it is “1,” the next instruction is executed.

● Data output from port D

Set the output level to the output latch with the **SD** and **RD** instructions.
The state of pin enters the high-impedance state when the **SD** instruction is executed.
The states of all port D enter the high-impedance state when the **CLD** instruction is executed.
The state of pin becomes “L” level when the **RD** instruction is executed.
The output structure is an N-channel open-drain.

Notes 1: When the **SD** and **RD** instructions are used, do not set “1000_2” or more to register Y.
2: Port D_6 is also used as CNTR0, and port D_7 is also used as CNTR1. Accordingly, when using ports D_6 and D_7 functions, set bit 0 ($W6_0$) and bit 2 ($W6_2$) of timer control register W6 to “0.”

2.1.2 Related registers

(1) Pull-up control register PU0

Register PU0 controls the ON/OFF of the ports $P0_0$–$P0_3$ and $P1_0$–$P1_3$ pull-up transistor.
Set the contents of this register through register A with the **TPU0A** instruction.
The contents of register PU0 is transferred to register A with the **TAPU0** instruction.
Table 2.1.1 shows the pull-up control register PU0.

Table 2.1.1 Pull-up control register PU0

Pull-up control register PU0		at reset : 0000_2	at RAM back-up : state retained	R/W
$PU0_3$	Ports $P1_2$, $P1_3$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
$PU0_2$	Ports $P1_0$, $P1_1$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
$PU0_1$	Ports $P0_2$, $P0_3$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
$PU0_0$	Ports $P0_0$, $P0_1$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	

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

(2) Key-on wakeup control register K0

Register K0 controls the ON/OFF of the key-on wakeup function of ports $P0_0$–$P0_3$ and $P1_0$–$P1_3$.
Set the contents of this register through register A with the **TK0A** instruction.
The contents of register K0 is transferred to register A with the **TAK0** instruction.
Table 2.1.2 shows the key-on wakeup control register K0.

Table 2.1.2 Key-on wakeup control register K0

Key-on wakeup control register K0		at reset : 0000_2	at RAM back-up : state retained	R/W
$K0_3$	Ports $P1_2$, $P1_3$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_2$	Ports $P1_0$, $P1_1$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_1$	Ports $P0_2$, $P0_3$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_0$	Ports $P0_0$, $P0_1$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	

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

(3) A-D control register Q2

Bits 0 to 2 of register Q2 controls the pin function selection bits.
Set the contents of this register through register A with the **TQ2A** instruction.
The contents of register Q2 is transferred to register A with the **TAQ2** instruction.
Table 2.1.3 shows the A-D control register Q2.

Table 2.1.3 A-D control register Q2

A-D control register Q2		at reset : 0000_2	at RAM back-up : state retained	R/W
$Q2_3$	A-D operation mode control bit	0	A-D conversion mode	
		1	Comparator mode	
$Q2_2$	$P4_3$/A_{IN7}, $P4_2$/A_{IN6} pin function selection bit (**Note 3**)	0	$P4_3$, $P4_2$ (I/O) (**Note 4**)	
		1	A_{IN7}, A_{IN6}/$P4_3$, $P4_2$ (Output) (**Note 4**)	
$Q2_1$	$P4_1$/A_{IN5} pin function selection bit (**Note 3**)	0	$P4_1$ (I/O) (**Note 4**)	
		1	A_{IN5}/$P4_1$ (Output) (**Note 4**)	
$Q2_0$	$P4_0$/A_{IN4} pin function selection bit (**Note 3**)	0	$P4_0$ (I/O) (**Note 4**)	
		1	A_{IN4}/$P4_0$ (Output) (**Note 4**)	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: Select A_{IN4}–A_{IN7} with register Q1 after setting register Q2.
3: For the 4513 Group, these bits are not used.
4: For the 4513 Group, only read/write of these bits is enabled.
5: When setting ports, $Q2_3$ is not used.

(4) Direction register FR0 (The 4513 Group does not have this register.)
Register FR0 is used to switch to input/output of $P5_0$–$P5_3$.
Set the contents of this register through register A with the **TFR0A** instruction.
Table 2.1.4 shows the direction register FR0.

Table 2.1.4 Direction register FR0

Direction register FR0 (**Note 2**)		at reset : 0000_2		at RAM back-up : state retained	W
$FR0_3$	Port $P5_3$ input/output control bit	0	Port $P5_3$ input		
		1	Port $P5_3$ output		
$FR0_2$	Port $P5_2$ input/output control bit	0	Port $P5_2$ input		
		1	Port $P5_2$ output		
$FR0_1$	Port $P5_1$ input/output control bit	0	Port $P5_1$ input		
		1	Port $P5_1$ output		
$FR0_0$	Port $P5_0$ input/output control bit	0	Port $P5_0$ input		
		1	Port $P5_0$ output		

Notes 1: "W" represents write enabled.
2: The 4513 Group does not have register FR0.

(5) Timer control register W6
D_6/CNTR0 function selection bit is assigned to bit 0, D_7/CNTR1 function selection bit is assigned to bit 2.
Set the contents of this register through register A with the **TW6A** instruction.
The contents of register W6 is transferred to register A with the **TAW6** instruction.
Table 2.1.5 shows the timer control register W6.

Table 2.1.5 Timer control register W6

Timer control register W6		at reset : 0000_2		at RAM back-up : state retained	R/W
$W6_3$	CNTR1 output control bit	0	Timer 3 underflow signal output divided by 2		
		1	CNTR1 output control by timer 4 underflow signal divided by 2		
$W6_2$	D_7/CNTR1 function selection bit	0	D_7(I/O)/CNTR1 input		
		1	CNTR1 I/O/D_7 (input)		
$W6_1$	CNTR0 output control bit	0	Timer 1 underflow signal output divided by 2		
		1	CNTR0 output control by timer 2 underflow signal divided by 2		
$W6_0$	D_6/CNTR0 function selection bit	0	D_6 (I/O)/CNTR0 input		
		1	CNTR0 I/O/D_6 (input)		

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: When setting ports, $W6_3$ and $W6_1$ are not used.

2.1.3 Port application examples

(1) Key input by key scan

Key matrix can be set up by connecting keys externally because port D output structure is an N-channel open-drain and port P0 has the pull-up resistor.

Outline: The connecting required external part is just keys.
Specifications: Port D is used to output "L" level and port P0 is used to input 16 keys. Multiple key inputs are not detected.

Figure 2.1.1 shows the key input and Figure 2.1.2 shows the key input timing.

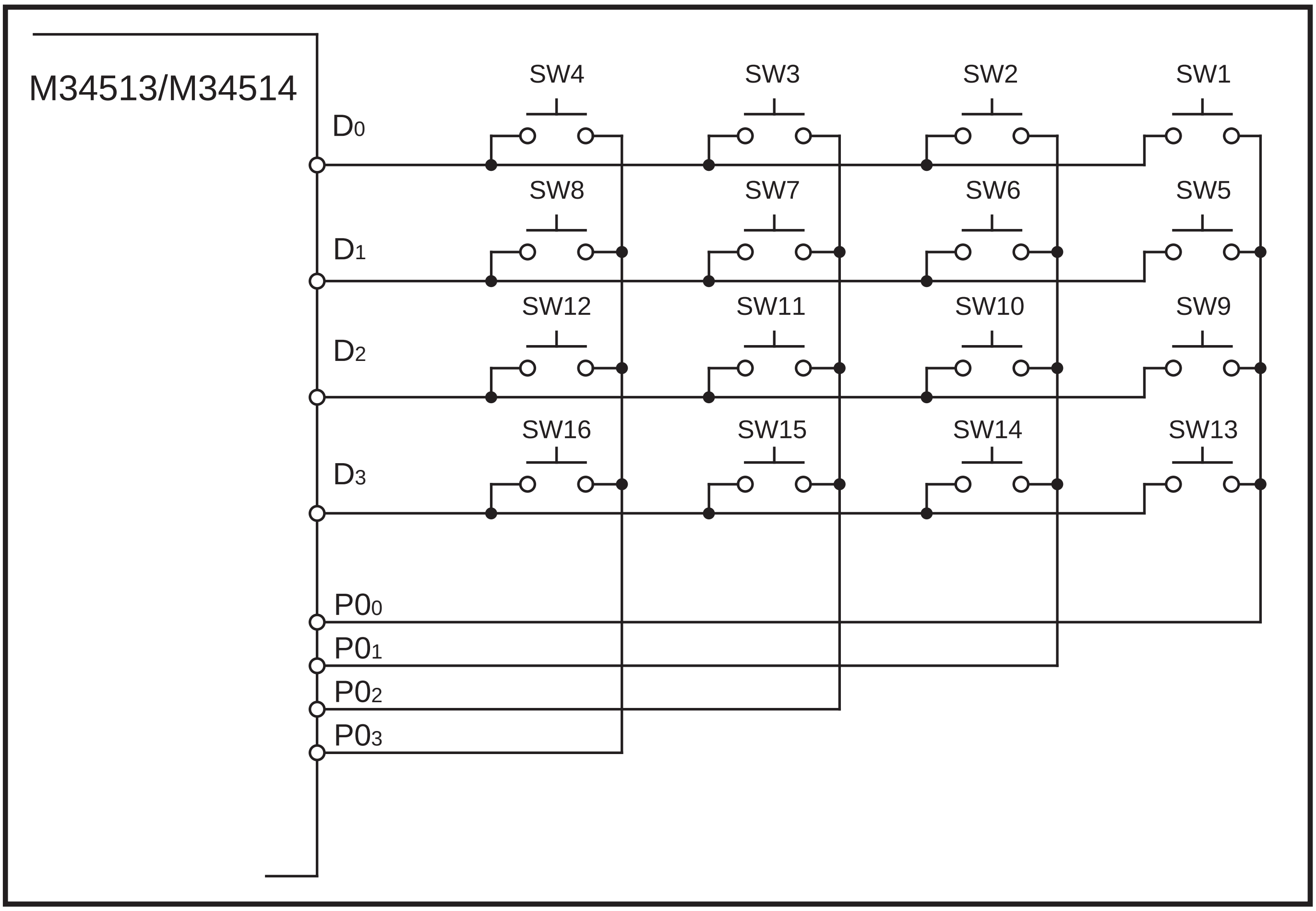
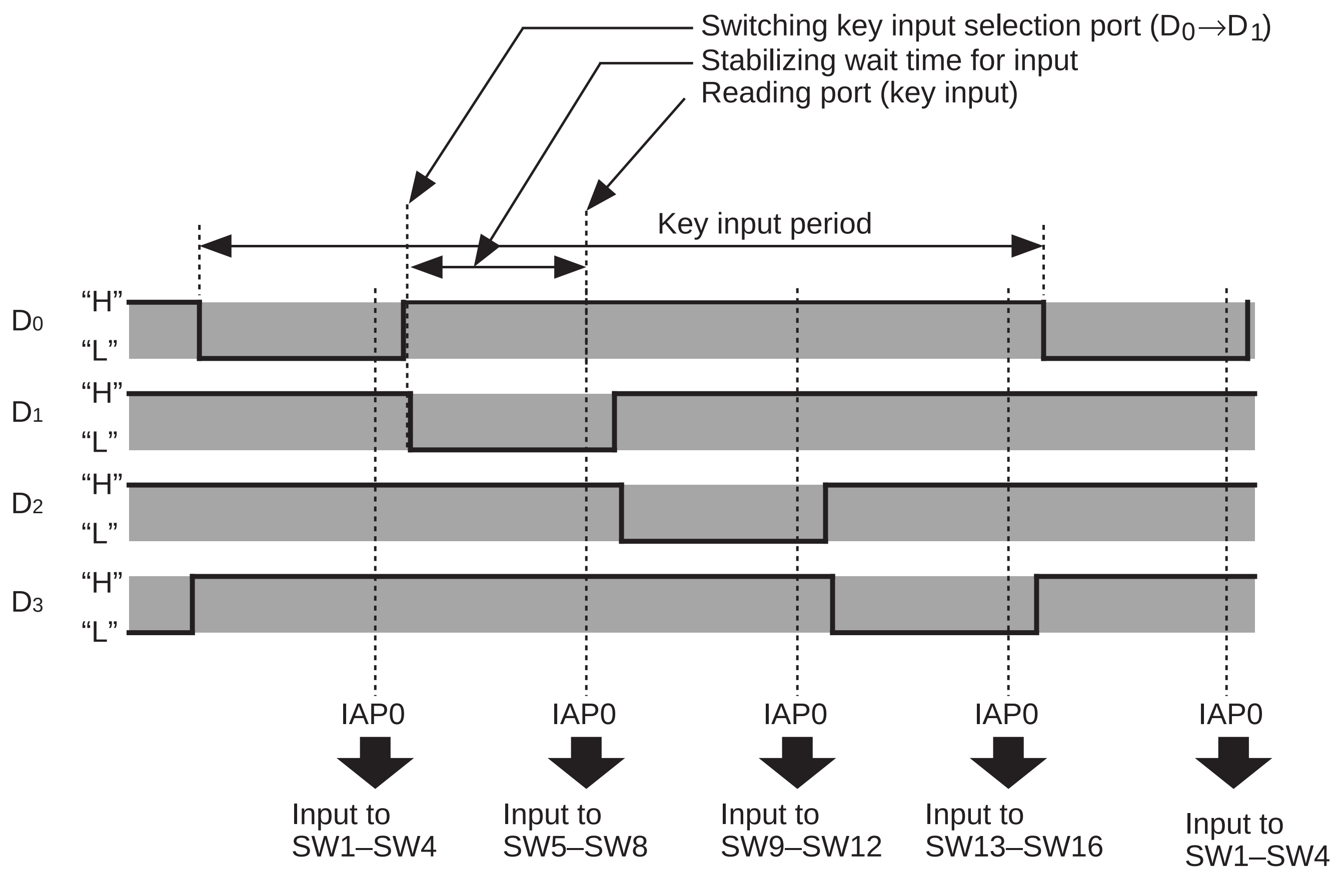


Fig. 2.1.1 Key input by key scan



Note: "H" output of port D becomes high-impedance state.

Fig. 2.1.2 Key scan input timing

2.1.4 Notes on use

(1) Note when an I/O port except port P5 is used as an input port

Set the output latch to "1" and input the port value before input. If the output latch is set to "0," "L" level can be input.

(2) Noise and latch-up prevention

Connect an approximate 0.1 μF bypass capacitor directly to the V_{SS} line and the V_{DD} line with the thickest possible wire at the shortest distance, and equalize its wiring in width and length.

The CNV_{SS} pin is also used as the V_{PP} pin (programming voltage = 12.5 V) at the built-in PROM version.

Connect the CNV_{SS}/V_{PP} pin to V_{SS} through an approximate 5 kΩ resistor which is connected to the CNV_{SS}/V_{PP} pin at the shortest distance.

(3) Note on multifunction

The input of D_6, D_7, $P2_0$–$P2_2$, CMP0-, CMP0+, CMP1-, CMP1+ and the input/output of $P3_0$, $P3_1$, $P4_0$–$P4_3$ can be used even when CNTR0, CNTR1, S_{CK}, S_{OUT}, S_{IN}, A_{IN0}–A_{IN3}, INT0, INT1, and A_{IN4}–A_{IN7} are selected.

(4) Connection of unused pins

Table 2.1.6 shows the connections of unused pins.

(5) SD, RD instructions

When the **SD** and **RD** instructions are used, do not set "1000_2" or more to register Y.

(6) Analog input pins

When both analog input A_{IN4}–A_{IN7} and I/O port P4 function are used, note the following;

- **Notes when selecting analog input pins**

  Even when register Q2 is used to set the pins for analog input, $P4_0/A_{IN4}$–$P4_3/A_{IN7}$ continue to function as $P4_0$–$P4_3$ I/O. Accordingly, when any of them are used as I/O port P4 and others are used as analog input pins, make sure to set the outputs of pins that are set for analog input to "1." Also, for the port input, the port input function of the pin functions as analog input is undefined.

(7) Notes on port P3

In the 4513 Group, when the **IAP3** instruction is executed, the contents of high-order 2 bits of register A are undefined.

Table 2.1.6 connections of unused pins

Pin	Connection
XOUT	Open (when using an external clock).
VDCE	Connect to VSS.
D0–D5	Connect to VSS, or set the output latch to “0” and open.
D6/CNTR0	
D7/CNTR1	
P20/SCK	Connect to VSS.
P21/SOUT	
P22/SIN	
P30/INT0	Connect to VSS, or set the output latch to “0” and open.
P31/INT1	
P32, P33	
P40/AIN4–P43/AIN7	Connect to VSS, or set the output latch to “0” and open.
P50–P53 (**Note 1**)	When the input mode is selected by software, pull-up to VDD through a resistor or pull-down to VSS. When selecting the output mode, open.
AIN0/CMP0-	Connect to VSS.
AIN1/CMP0+	
AIN2/CMP1-	
AIN3/CMP1+	
P00–P03	Open or connect to VSS (**Note 2**).
P10–P13	Open or connect to VSS (**Note 2**).

Notes 1: After system is released from reset, port P5 is in an input mode (direction register FR0 = 0000_2)

2: 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 in order to set the output latch to “0” and make pins open)

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

(Note in order to connect unused pins to VSS or VDD)

- To avoid noise, connect the unused pins to VSS or VDD at the shortest distance using a thick wire.

2.2 Interrupts

The 4513/4514 Group has eight interrupt sources : external (INT0, INT1), timer 1, timer 2, timer 3, timer 4, A-D, and serial I/O.
This section describes individual types of interrupts, related registers, application examples using interrupts and notes.

2.2.1 Interrupt functions

(1) External 0 interrupt (INT0)
The interrupt request occurs by the change of input level of INT0 pin.
The interrupt valid waveform can be selected by the bits 1 and 2 of the interrupt control register I1.

■ **External 0 interrupt INT0 processing**

● When the interrupt is used
The interrupt occurrence is enabled when the bit 0 of the interrupt control register V1 and the interrupt enable flag INTE are set to “1.” When the external 0 interrupt occurs, the interrupt processing is executed from address 0 in page 1.

● When the interrupt is not used
The interrupt is disabled and the **SNZ0** instruction is valid when the bit 0 of register V1 is set to “0.”

(2) External 1 interrupt (INT1)
The interrupt request occurs by the change of input level of INT1 pin.
The interrupt valid waveform can be selected by the bits 1 and 2 of the interrupt control register I2.

■ **External 1 interrupt INT1 processing**

● When the interrupt is used
The interrupt occurrence is enabled when the bit 1 of the interrupt control register V1 and the interrupt enable flag INTE are set to “1.” When the external 1 interrupt occurs, the interrupt processing is executed from address 2 in page 1.

● When the interrupt is not used
The interrupt is disabled and the **SNZ1** instruction is valid when the bit 1 of register V1 is set to “0.”

(3) Timer 1 interrupt
The interrupt request occurs by the timer 1 underflow.

■ **Timer 1 interrupt processing**

● When the interrupt is used
The interrupt occurrence is enabled when the bit 2 of the interrupt control register V1 and the interrupt enable flag INTE are set to “1.” When the timer 1 interrupt occurs, the interrupt processing is executed from address 4 in page 1.

● When the interrupt is not used
The interrupt is disabled and the **SNZT1** instruction is valid when the bit 2 of register V1 is set to “0.”

(4) Timer 2 interrupt
The interrupt request occurs by the timer 2 underflow.

■ **Timer 2 interrupt processing**

● When the interrupt is used
The interrupt occurrence is enabled when the bit 3 of the interrupt control register V1 and the interrupt enable flag INTE are set to “1.” When the timer 2 interrupt occurs, the interrupt processing is executed from address 6 in page 1.

● When the interrupt is not used
The interrupt is disabled and the **SNZT2** instruction is valid when the bit 3 of register V1 is set to “0.”

(5) Timer 3 interrupt
The interrupt request occurs by the timer 3 underflow.

■ **Timer 3 interrupt processing**

● When the interrupt is used
The interrupt occurrence is enabled when the bit 0 of the interrupt control register V2 and the interrupt enable flag INTE are set to “1.” When the timer 3 interrupt occurs, the interrupt processing is executed from address 8 in page 1.

● When the interrupt is not used
The interrupt is disabled and the **SNZT3** instruction is valid when the bit 0 of register V2 is set to “0.”

(6) Timer 4 interrupt
The interrupt request occurs by the timer 4 underflow.

■ **Timer 4 interrupt processing**

● When the interrupt is used
The interrupt occurrence is enabled when the bit 1 of the interrupt control register V2 and the interrupt enable flag INTE are set to “1.” When the timer 4 interrupt occurs, the interrupt processing is executed from address A in page 1.

● When the interrupt is not used
The interrupt is disabled and the **SNZT4** instruction is valid when the bit 1 of register V2 is set to “0.”

(7) A-D interrupt

The interrupt request occurs by the end of the A-D conversion.

■ **A-D interrupt processing**

● When the interrupt is used
The interrupt occurrence is enabled when the bit 2 of the interrupt control register V2 and the interrupt enable flag INTE are set to “1.” When the A-D interrupt occurs, the interrupt processing is executed from address C in page 1.

● When the interrupt is not used
The interrupt is disabled and the **SNZAD** instruction is valid when the bit 2 of register V2 is set to “0.”

(8) Serial I/O interrupt

The interrupt request occurs by the end of the serial I/O transmit/receive.

■ **Serial I/O interrupt processing**

● When the interrupt is used
The interrupt occurrence is enabled when the bit 3 of the interrupt control register V2 and the interrupt enable flag INTE are set to “1.” When the serial I/O interrupt occurs, the interrupt processing is executed from address E in page 1.

● When the interrupt is not used
The interrupt is disabled and the **SNZSI** instruction is valid when the bit 3 of register V2 is set to “0.”

2.2.2 Related registers

(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.

Note: The interrupt enabled with the **EI** instruction is performed after the **EI** instruction and one more instruction.

(2) 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.
In addition, the **TAV1** instruction can be used to transfer the contents of register V1 to register A.
Table 2.2.1 shows the interrupt control register V1.

Table 2.2.1 Interrupt control register V1

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)	

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

(3) 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.
In addition, the **TAV2** instruction can be used to transfer the contents of register V2 to register A.
Table 2.2.2 shows the interrupt control register V2.

Table 2.2.2 Interrupt control register V2

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.

(4) Interrupt request flag

The activated condition for each interrupt is examined. Each interrupt request flag is set to "1" when the activated condition is satisfied, even if the interrupt is disabled by the INTE flag or its interrupt enable bit.
Each interrupt request flag is cleared to "0" when either;
•an interrupt occurs, or
•the next instruction is skipped with a skip instruction.

(5) Interrupt control register I1

The INT0 pin timer 1 control enable bit is assigned to bit 0, INT0 pin edge detection circuit control bit is assigned to bit 1, and interrupt valid waveform for INT0 pin/return level selection bit is assigned to bit 2.

Set the contents of this register through register A with the **TI1A** instruction.

In addition, the **TAI1** instruction can be used to transfer the contents of register I1 to register A.

Table 2.2.3 shows the interrupt control register I1.

Table 2.2.3 Interrupt control register I1

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	

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.

(6) Interrupt control register I2

The INT1 pin timer 3 control enable bit is assigned to bit 0, the INT1 pin edge detection circuit control bit is assigned to bit 1 and the interrupt valid waveform for INT1 pin/return level selection bit is assigned to bit 2.

Set the contents of this register through register A with the **TI2A** instruction.

In addition, the **TAI2** instruction can be used to transfer the contents of register I2 to register A.

Table 2.2.4 shows the interrupt control register I2.

Table 2.2.4 Interrupt control register I2

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 2**)	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 $I2_2$ is changed, the external interrupt request flag EXF1 may be set. Accordingly, clear EXF1 flag with the **SNZ1** instruction.

2.2.3 Interrupt application examples

(1) External 0 interrupt

The INT0 pin is used for external 0 interrupt, of which valid waveforms can be chosen, which can recognize the change of both edges (“H”→“L” or “L”→“H”).

Outline: An external 0 interrupt can be used by dealing with the change of edge (“H”→“L” or “L”→“H”) in both directions as a trigger.
Specifications: An interrupt occurs by the change of an external signals edge (“H”→“L” or “L”→“H”).

Figure 2.2.1 shows an operation example of an external 0 interrupt, and Figure 2.2.2 shows a setting example of an external 0 interrupt.

(2) External 1 interrupt

The INT1 pin is used for external 1 interrupt, of which valid waveforms can be chosen, which can recognize the change of both edges (“H”→“L” or “L”→“H”).

Outline: An external 1 interrupt can be used by dealing with the change of edge (“H”→“L” or “L”→“H”) in both directions as a trigger.
Specifications: An interrupt occurs by the change of an external signals edge (“H”→“L” or “L”→“H”).

Figure 2.2.3 shows an operation example of an external 1 interrupt, and Figure 2.2.4 shows a setting example of an external 1 interrupt.

(3) Timer 1 interrupt

Constant period interrupts by a setting value to timer 1 can be used.

Outline: The constant period interrupts by the timer 1 underflow signal can be used.
Specifications: Prescaler and timer 1 divide the system clock frequency $f(X_{IN})$ = 4.0 MHz, and the timer 1 interrupt occurs every 1 ms.

Figure 2.2.5 shows a setting example of the timer 1 constant period interrupt.

(4) Timer 2 interrupt

Constant period interrupts by a setting value to timer 2 can be used.

Outline: The constant period interrupts by the timer 2 underflow signal can be used.
Specifications: Timer 2 divides the 16-bit fixed dividing frequency timer, and the timer 2 interrupt occurs every about 2 sec.

Figure 2.2.6 shows a setting example of the timer 2 constant period interrupt.

(5) Timer 3 interrupt

Constant period interrupts by a setting value to timer 3 can be used.

Outline: The constant period interrupts by the timer 3 underflow signal can be used.
Specifications: Prescaler and timer 3 divide the system clock frequency $f(X_{IN})$ = 4.0 MHz, and the timer 3 interrupt occurs every 1 ms.

Figure 2.2.7 shows a setting example of the timer 3 constant period interrupt.

(6) Timer 4 interrupt

Constant period interrupts by a setting value to timer 4 can be used.

Outline: The constant period interrupts by the timer 4 underflow signal can be used.
Specifications: Prescaler, timer 3 and timer 4 divide the system clock frequency f(XIN) = 4.0 MHz, and the timer 4 interrupt occurs every 250 ms.

Figure 2.2.8 shows a setting example of the timer 4 constant period interrupt.

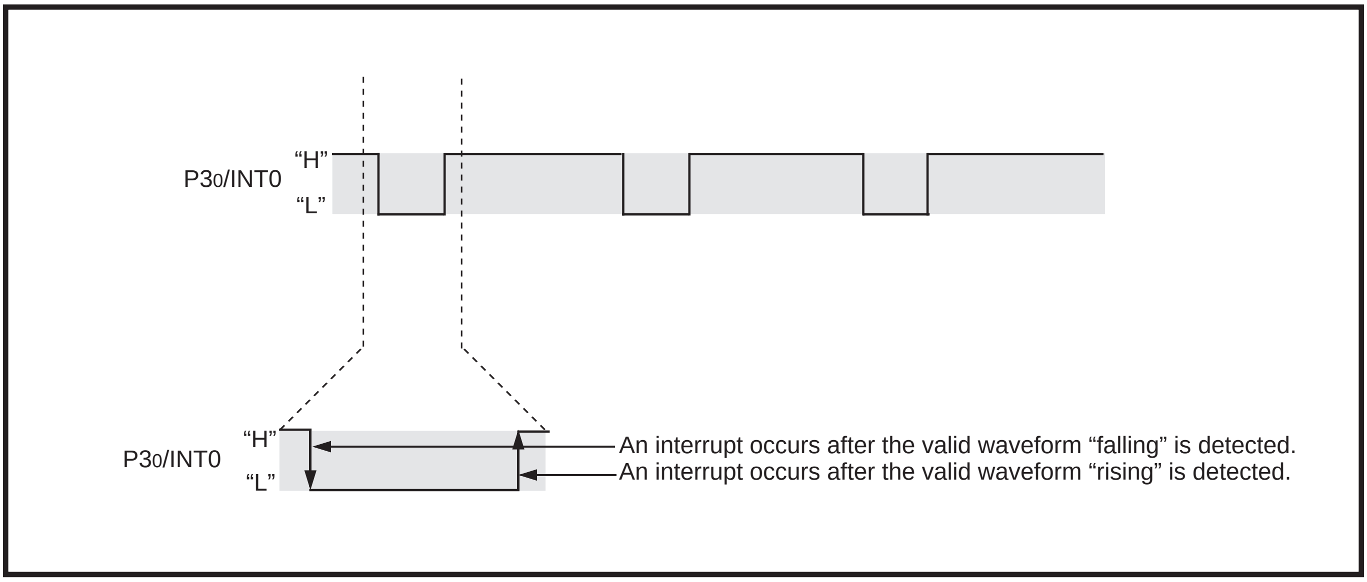


Fig. 2.2.1 INT0 interrupt operation example

① Disable Interrupts

INT0 interrupt is temporarily disabled.

Interrupt enable flag INTE "0" All interrupts disabled (**DI** instruction)

Interrupt control register V1 (b3–b0) ✕ ✕ ✕ 0 INT0 interrupt occurrence disabled (**TV1A** instruction)

↓

② Set Port

Port used for INT0 interrupt is set to input port.

Port P3₀ output latch (b3–b0) ✕ ✕ ✕ 1 Set to input (**OP3A** instruction)

↓

③ Set Valid Waveform

Valid waveform of INT pin is selected.
Both edges detection selected

Interrupt control register I1 (b3–b0) ✕ ✕ 1 ✕ Both edges detection selected (**TI1A** instruction)

↓

④ Clear Interrupt Request

External interrupt activated condition is cleared.

INT0 interrupt request flag EXF0 "0" INT0 interrupt activated condition cleared (**SNZ0** instruction)

↓

(
Note when the interrupt request is cleared

When ④ is executed, considering the skip of the next instruction according to the interrupt request flag EXF0, insert the **NOP** instruction after the **SNZ0** instruction.
)

↓

⑤ Enable Interrupts

The INT0 interrupt which is temporarily disabled is enabled.

Interrupt control register V1 (b3–b0) ✕ ✕ ✕ 1 INT0 interrupt occurrence enabled (**TV1A** instruction)

Interrupt enable flag INTE "1" All interrupts enabled (**EI** instruction)

↓

INT0 interrupt execution started

"✕": it can be "0" or "1."

Fig. 2.2.2 INT0 interrupt setting example

Note: The valid waveforms causing the interrupt must be retained at their level for 4 cycles or more of system clock.

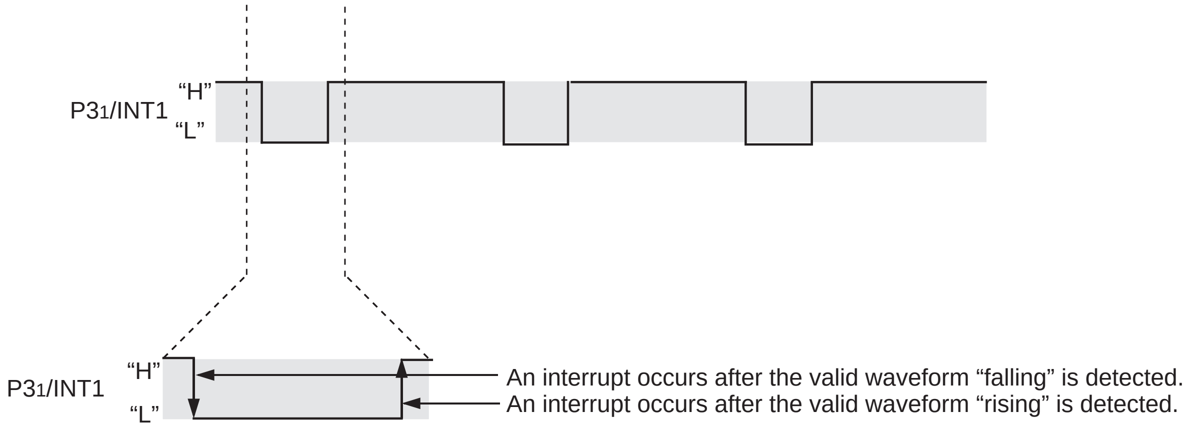


Fig. 2.2.3 INT1 interrupt operation example

① Disable Interrupts

INT1 interrupt is temporarily disabled.

Interrupt enable flag INTE	"0"	All interrupts disabled (**DI** instruction)
Interrupt control register V1	b3 × × 0 × b0	INT1 interrupt occurrence disabled (**TV1A** instruction)

↓

② Set Port

Port used for INT1 interrupt is set to input port.

Port $P3_1$ output latch	b3 × × 1 × b0	Set to input (**OP3A** instruction)

↓

③ Set Valid Waveform

Valid waveform of INT pin is selected.
Both edges detection selected

Interrupt control register I2	b3 × × 1 × b0	Both edges detection selected (**TI2A** instruction)

↓

④ Clear Interrupt Request

External interrupt activated condition is cleared.

INT1 interrupt request flag EXF1	"0"	INT1 interrupt activated condition cleared (**SNZ1** instruction)

↓

Note when the interrupt request is cleared

When ④ is executed, considering the skip of the next instruction according to the interrupt request flag EXF1, insert the **NOP** instruction after the **SNZ1** instruction.

↓

⑤ Enable Interrupts

The INT1 interrupt which is temporarily disabled is enabled.

Interrupt control register V1	b3 × × 1 × b0	INT1 interrupt occurrence enabled (**TV1A** instruction)
Interrupt enable flag INTE	"1"	All interrupts enabled (**EI** instruction)

↓

INT1 interrupt execution started

"×": it can be "0" or "1."

Fig. 2.2.4 INT1 interrupt setting example

Note: The valid waveforms causing the interrupt must be retained at their level for 4 cycles or more of system clock.

① Disable Interrupts
Timer 1 interrupt is temporarily disabled.

Interrupt enable flag INTE "0" All interrupts disabled (**DI** instruction)

Interrupt control register V1 (b3–b0) ×0×× Timer 1 interrupt occurrence disabled (**TV1A** instruction)

↓

② Stop Timer Operation
Timer 1 and prescaler are temporarily stopped.
Dividing ratio of prescaler is selected.

Timer control register W1 (b3–b0) 010× Timer 1 stop (**TW1A** instruction)
Prescaler stop
Prescaler divided by 16 selected

↓

③ Set Timer Value
Timer 1 count time is set. (The formula is shown ❋A below.)

Timer 1 reload register R1 "52_{16}" Timer count value 82 set (**T1AB** instruction)

↓

④ Clear Interrupt Request
Timer 1 interrupt activated condition is cleared.

Timer 1 interrupt request flag T1F "0" Timer 1 interrupt activated condition cleared (**SNZT1** instruction)

↓

Note when the interrupt request is cleared
When ④ is executed, considering the skip of the next instruction according to the interrupt request flag T1F, insert the **NOP** instruction after the **SNZT1** instruction.

↓

⑤ Start Timer Operation
Timer 1 and prescaler temporarily stopped are restarted.

Timer control register W1 (b3–b0) 111× Timer 1 operation start (**TW1A** instruction)
Prescaler operation stop

↓

⑥ Enable Interrupts
The Timer 1 interrupt which is temporarily disabled is enabled.

Interrupt control register V1 (b3–b0) ×1×× Timer 1 interrupt occurrence enabled (**TV1A** instruction)

Interrupt enable flag INTE "1" All interrupts enabled (**EI** instruction)

↓

Constant period interrupt execution start

❋A The prescaler dividing ratio and timer 1 count value to make the interrupt occur every 1 ms are set as follows.

$$1\text{ ms} \cong \underbrace{(4.0\text{ MHz})^{-1}}_{\text{System clock}} \times \underbrace{3}_{\text{Instruction clock}} \times \underbrace{16}_{\text{Prescaler dividing ratio}} \times \underbrace{(82+1)}_{\text{Timer 1 count value}}$$

"×": it can be "0" or "1."

Fig. 2.2.5 Timer 1 constant period interrupt setting example

① Disable Interrupts

Timer 2 interrupt is temporarily disabled.

Interrupt enable flag INTE "0" — All interrupts disabled (**DI** instruction)

Interrupt control register V1 (b3–b0) 0 X X X — Timer 2 interrupt occurrence disabled (**TV1A** instruction)

↓

② Stop Timer Operation

Timer is temporarily stopped.
Timer 2 count source is selected.

Timer control register W2 (b3–b0) 0 X 1 1 — Timer 2 stop (**TW2A** instruction); 16-bit timer (WDT) underflow signal selected for count source

↓

③ Set Timer Value

Timer 2 count time is set. (The formula is shown ❋A below.)

Timer 2 reload register R2 "27_{16}" — Timer count value 39 set (**T2AB** instruction)

↓

④ Clear Interrupt Request

Timer 2 interrupt activated condition is cleared.

Timer 2 interrupt request flag T2F "0" — Timer 2 interrupt activated condition cleared (**SNZT2** instruction)

↓

Note when the interrupt request is cleared
When ④ is executed, considering the skip of the next instruction according to the interrupt request flag T2F, insert the **NOP** instruction after the **SNZT2** instruction.

↓

⑤ Start Timer 2 Operation

Timer 2 temporarily stopped is restarted.

Timer control register W2 (b3–b0) 1 X 1 1 — Timer 2 operation start (**TW2A** instruction)

↓

⑥ Enable Interrupts

The timer 2 interrupt which is temporarily disabled is enabled.

Interrupt control register V1 (b3–b0) 1 X X X — Timer 2 interrupt occurrence enabled (**TV1A** instruction)

Interrupt enable flag INTE "1" — All interrupts enabled (**EI** instruction)

↓

Constant period interrupt execution start

❋A The timer 2 count value to make the interrupt occur every about 2 s is set as follows.

$$2\text{ s} \cong \underbrace{(4.0\text{ MHz})^{-1}}_{\text{System clock}} \times \underbrace{3}_{\text{Instruction clock}} \times \underbrace{2^{16}}_{\text{16-bit fixed dividing frequency}} \times \underbrace{(39+1)}_{\text{Timer 2 count value}}$$

"X": it can be "0" or "1."

Fig. 2.2.6 Timer 2 constant period interrupt setting example

① Disable Interrupts

Timer 3 interrupt is temporarily disabled.

Interrupt enable flag INTE "0" — All interrupts disabled (**DI** instruction)

Interrupt control register V2 (b3–b0): × × × 0 — Timer 3 interrupt occurrence disabled (**TV2A** instruction)

↓

② Stop Timer 3 Operation

Timer 3 and prescaler are temporarily stopped.
Dividing ratio of prescaler is selected.

Timer control register W1 (b3–b0): 0 1 × × — Prescaler stop (**TW1A** instruction)
Prescaler divided by 16 selected

Timer control register W3 (b3–b0): 0 × 0 1 — Timer 3 stop (**TW3A** instruction)
Prescaler selected for count source

↓

③ Set Timer Value

Timer 3 count time is set. (The formula is shown ❋A below.)

Timer 3 reload register R3 "52_{16}" — Timer count value 82 set (**T3AB** instruction)

↓

④ Clear Interrupt Request

Timer 3 interrupt activated condition is cleared.

Timer 3 interrupt request flag T3F "0" — Timer 3 interrupt activated condition cleared (**SNZT3** instruction)

↓

Note when the interrupt request is cleared

When ④ is executed, considering the skip of the next instruction according to the interrupt request flag T3F, insert the **NOP** instruction after the **SNZT3** instruction.

↓

⑤ Start Timer 3 Operation

Timer 3 and prescaler temporarily stopped are restarted.

Timer control register W3 (b3–b0): 1 × 0 1 — Timer 3 operation start (**TW3A** instruction)

Timer control register W1 (b3–b0): 1 1 × × — Prescaler operation start (**TW1A** instruction)

↓

⑥ Enable Interrupts

The timer 3 interrupt which is temporarily disabled is enabled.

Interrupt control register V2 (b3–b0): × × × 1 — Timer 3 interrupt occurrence enabled (**TV2A** instruction)

Interrupt enable flag INTE "1" — All interrupts enabled (**EI** instruction)

↓

Constant period interrupt execution start

❋A The prescaler dividing ratio and time 3 count value to make the interrupt occur every 1 ms are set as follows.

$$1\text{ ms} \cong \underbrace{(4.0\text{ MHz})^{-1}}_{\text{System clock}} \times \underbrace{3}_{\text{Instruction clock}} \times \underbrace{16}_{\text{Prescaler dividing ratio}} \times \underbrace{(82+1)}_{\text{Timer 3 count value}}$$

"×": it can be "0" or "1."

Fig. 2.2.7 Timer 3 constant period interrupt setting example

① Disable Interrupts

Timer 4 interrupt is temporarily disabled.

Interrupt enable flag INTE "0" — All interrupts disabled (**DI** instruction)

Interrupt control register V2 (b3–b0): × × 0 × — Timer 4 interrupt occurrence disabled (**TV2A** instruction)

↓

② Stop Timer Operation

Timer 4, timer 3 and prescaler are temporarily stopped.
Dividing ratio of prescaler is selected.

Timer control register W1 (b3–b0): 0 1 × × — Prescaler stop (**TW1A** instruction)
Prescaler divided by 16 selected

Timer control register W3 (b3–b0): 0 × 0 1 — Timer 3 stop (**TW3A** instruction)
Prescaler selected for count source

Timer control register W4 (b3–b0): 0 × 0 0 — Timer 4 stop (**TW4A** instruction)
Timer 3 underflow signal selected for count source

↓

③ Set Timer Value

Timer 3 and timer 4 count times are set. (The formula is shown ❋A below.)

Timer 3 reload register R3 "52_{16}" — Timer count value 82 set (**T3AB** instruction)
Timer 4 reload register R4 "$F9_{16}$" — Timer count value 249 set (**T4AB** instruction)

↓

④ Clear Interrupt Request

Timer 4 interrupt activated condition is cleared.

Timer 4 interrupt request flag T4F "0" — Timer 4 interrupt activated condition cleared (**SNZT4** instruction)

↓

<u>Note when the interrupt request is cleared</u>
When ④ is executed, considering the skip of the next instruction according to the interrupt request flag T4F, insert the **NOP** instruction after the **SNZT4** instruction.

↓

⑤ Start Timer 4 Operation

Timer 4, timer 3 and prescaler temporarily stopped are restarted.

Timer control register W4 (b3–b0): 1 × 0 0 — Timer 4 operation start (**TW4A** instruction)

Timer control register W3 (b3–b0): 1 × 0 1 — Timer 3 operation start (**TW3A** instruction)

Timer control register W1 (b3–b0): 1 1 × × — Prescaler operation start (**TW1A** instruction)

↓

⑥ Enable Interrupts

The timer 4 interrupt which is temporarily disabled is enabled.

Interrupt control register V2 (b3–b0): × × 1 × — Timer 4 interrupt occurrence enabled (**TV2A** instruction)

Interrupt enable flag INTE "1" — All interrupts enabled (**EI** instruction)

↓

Constant period interrupt execution start

❋A The prescaler dividing ratio, time 3 count value and timer 4 count value to make the interrupt occur every 250 ms are set as follows.

$$250\text{ ms} \cong \underbrace{(4.0\text{ MHz})^{-1}}_{\text{System clock}} \times \underbrace{3}_{\text{Instruction clock}} \times \underbrace{16}_{\text{Prescaler dividing ratio}} \times \underbrace{(82+1)}_{\text{Timer 3 count value}} \times \underbrace{(249+1)}_{\text{Timer 4 count value}}$$

"×": it can be "0" or "1."

Fig. 2.2.8 Timer 4 constant period interrupt setting example

2.2.4 Notes on use

(1) Setting of INT0 interrupt valid waveform
Depending on the input state of $P3_0$/INT0 pin, the external interrupt request flag (EXF0) may be set to “1” when the interrupt valid waveform is changed. Accordingly, set a value to the bit 2 of register I1, and execute the **SNZ0** instruction to clear the EXF0 flag to “0” after executing at least one instruction.

(2) Setting of INT1 interrupt valid waveform
Depending on the input state of $P3_1$/INT1 pin, the external interrupt request flag (EXF1) may be set to “1” when the interrupt valid waveform is changed. Accordingly, set a value to the bit 2 of register I2, and execute the **SNZ1** instruction to clear the EXF1 flag to “0” after executing at least one instruction.

(3) Multiple interrupts
Multiple interrupts cannot be used in the 4513/4514 Group.

(4) Notes on interrupt processing
When the interrupt occurs, at the same time, the interrupt enable flag INTE is cleared to “0” (interrupt disable state). In order to enable the interrupt at the same time when system returns from the interrupt, write **EI** and **RTI** instructions continuously.

(5) $P3_0$/INT0 pin
The $P3_0$/INT0 pin need not be selected the external interrupt input INT function or the normal output port $P3_0$ function. However, the EXF0 flag is set to “1” when a valid waveform is input to INT0 pin even if it is used as an I/O port $P3_0$.

(6) $P3_1$/INT1 pin
The $P3_1$/INT1 pin need not be selected the external interrupt input INT function or the normal output port $P3_1$ function. However, the EXF1 flag is set to “1” when a valid waveform is input to INT1 pin even if it is used as an I/O port $P3_1$.

(7) EPOF instruction
Be sure to disable interrupts by executing the **DI** instruction before executing the **EPOF** instruction.

2.3 Timers

The 4513/4514 Group has four 8-bit timers (each has a reload register) and a 16-bit fixed dividing frequency timer which has the watchdog timer function.
This section describes individual types of timers, related registers, application examples using timers and notes.

2.3.1 Timer functions

(1) Timer 1

■ **Timer operation**
(Timer 1 has the timer 1 count start trigger function from $P3_0$/INT0 pin input)

(2) Timer 2

■ **Timer operation**

(3) Timer 3

■ **Timer operation**
(Timer 3 has the timer 3 count start trigger function from $P3_1$/INT1 pin input)

(4) Timer 4

■ **Timer operation**

(5) 16-bit timer

■ **Timer 2 count source**
(16-bit fixed dividing frequency)

■ **Watchdog function**
Watchdog timer provides a method to reset the system when a program runs incorrectly.
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" to reset the microcomputer.

2.3.2 Related registers

(1) Interrupt control register V1

The timer 1 interrupt enable bit is assigned to bit 2, and the timer 2 interrupt enable bit is assigned to bit 3.

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.

Table 2.3.1 shows the interrupt control register V1.

Table 2.3.1 Interrupt control register V1

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)	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: When timer is used, $V1_1$ and $V1_0$ are not used.

(2) Interrupt control register V2

The timer 3 interrupt enable bit is assigned to bit 0, and the timer 4 interrupt enable bit is assigned to bit 1.

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 2.3.2 shows the interrupt control register V2.

Table 2.3.2 Interrupt control register V2

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)	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: When timer is used, $V2_2$ and $V2_3$ are not used.

(3) Timer control register W1

The timer 1 count start synchronous circuit control bit is assigned to bit 0, the timer 1 control bit is assigned to bit 1, the prescaler dividing ratio selection bit is assigned to bit 2, and the prescaler control bit is assigned to bit 3.

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.

Table 2.3.3 shows the timer control register W1.

Table 2.3.3 Timer control register W1

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 synchronous circuit control bit	0	Count start synchronous circuit not selected	
		1	Count start synchronous circuit selected	

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

(4) Timer control register W2

The timer 2 count source selection bits are assigned to bits 0 and 1, and the timer 2 control bit is assigned to bit 3.

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.

Table 2.3.4 shows the timer control register W2.

Table 2.3.4 Timer control register W2

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	
		0	1	Prescaler output	
$W2_0$		1	0	CNTR0 input	
		1	1	16-bit timer (WDT) underflow signal	

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

(5) Timer control register W3

The timer 3 count source selection bits are assigned to bits 0 and 1, the timer 3 count start synchronous circuit control bit is assigned to bit 2 and the timer 3 control bit is assigned to bit 3.
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.
Table 2.3.5 shows the timer control register W3.

Table 2.3.5 Timer control register W3

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	

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

(6) Timer control register W4

The timer 4 count source selection bits are assigned to bits 0 and 1, and the timer 4 control bit is assigned to bit 3.
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.
Table 2.3.6 shows the timer control register W4.

Table 2.3.6 Timer control register W4

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	CNTR1 input	
		1	1	Not available	

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

2.3.3 Timer application examples

(1) Timer operation: measurement of constant period

The constant period by the setting timer count value can be measured.

Outline: The constant period by the timer 1 underflow signal can be measured.
Specifications: Timer 1 and prescaler divides the system clock frequency f(XIN) = 4.0 MHz, and the timer 1 interrupt request occurs every 3 ms.

Figure 2.3.3 shows the setting example of the constant period measurement.

(2) CNTR0 output operation: piezoelectric buzzer output

Outline: Square wave output from timer 1 can be used for piezoelectric buzzer output.
Specifications: 4 kHz square wave is output from the CNTR0 pin at system clock frequency f(XIN) = 4.0 MHz. Also, timer 1 interrupt occurs simultaneously.

Figure 2.3.1 shows the peripheral circuit example, and Figure 2.3.4 shows the setting example of CNTR0 output.

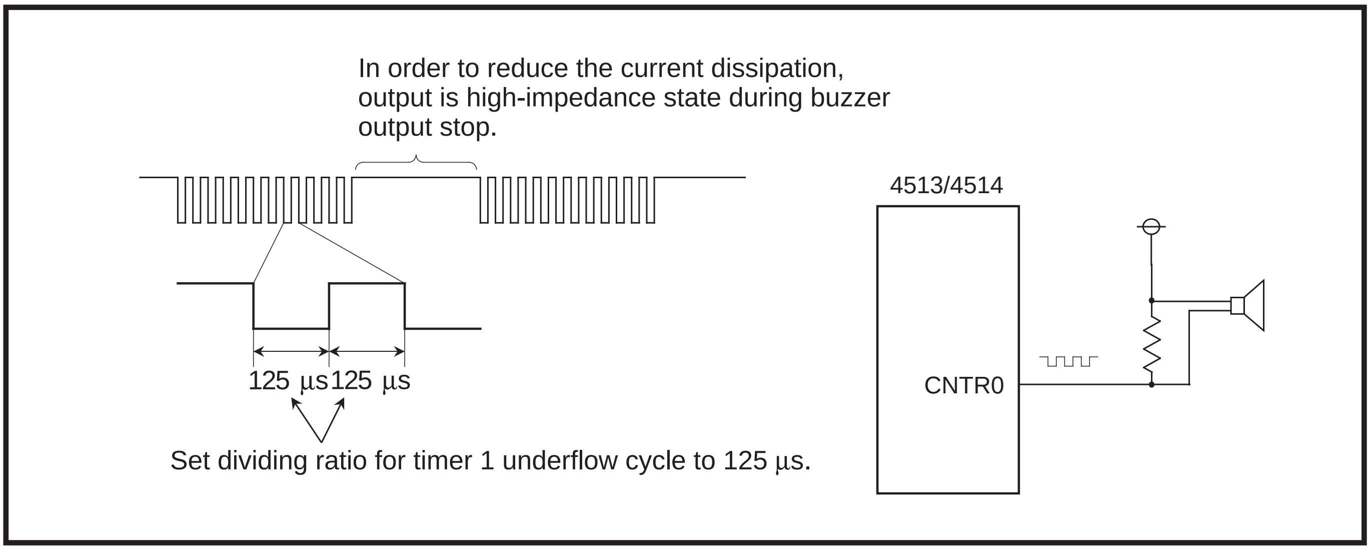


Fig. 2.3.1 Peripheral circuit example

(3) CNTR0 input operation: event count

Outline: Count operation can be performed by using the signal (rising waveform) input from CNTR0 pin as the event.
Specifications: The low-frequency pulse from external as the timer 2 count source is input to CNTR0 pin, and the timer 2 interrupt request occurs every 100 counts.

Figure 2.3.5 shows the setting example of CNTR0 input.

(4) CNTR1 output control: square wave output control

Outline: The output/stop of square wave from timer 3 every timer 4 underflow can be controlled.
Specifications: 4 kHz square wave is output from timer 3 at system clock frequency f(XIN) = 4.0 MHz. Also, timer 4 controls ON/OFF of square wave every constant period.

Figure 2.3.6 shows the setting example of CNTR1 output.

(5) Timer operation: timer start by external input

Outline: The constant period can be measured by external input.
Specifications: Timer 1 operates by INT0 input as a trigger and an interrupt occurs after 1 ms.

Figure 2.3.7 and Figure 2.3.8 show the setting example of timer start.

(6) Watchdog timer
Watchdog timer provides a method to reset the system when a program run-away occurs.
In the 4513/4514 Group, bit 15 of 16-bit timer is counted twice for the watchdog timer.
Accordingly, when the watchdog timer function is set to be valid, execute the **WRST** instruction at a certain period which consists of timer 16-bit timers' 32767 counts or less (execute **WRST** instruction at a cycle of 32766 machine cycles or less).

Outline: Execute the **WRST** instruction in 16-bit timer's 32767 counts at the normal operation. If a program runs incorrectly, the **WRST** instruction is not executed and system reset occurs.
Specifications: System clock frequency f(XIN) = 4.0 MHz is used, and program run-away is detected by executing the **WRST** instruction in 24 ms.

Figure 2.3.2 shows the watchdog timer function, and Figure 2.3.9 shows the example of watchdog timer.

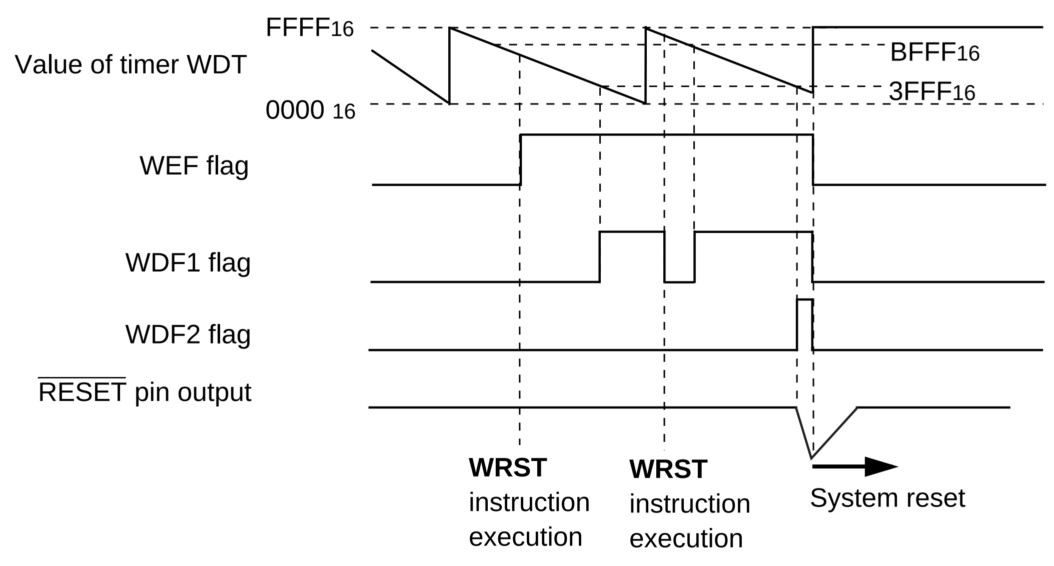


Fig. 2.3.2 Watchdog timer function

① Disable Interrupts

Timer 1 interrupt is temporarily disabled.

Interrupt enable flag INTE "0" — All interrupts disabled (**DI** instruction)

Interrupt control register V1 (b3–b0) ✕ 0 ✕ ✕ — Timer 1 interrupt occurrence disabled (**TV1A** instruction)

↓

② Stop Timer Operation

Timer 1 and prescaler are temporarily stopped.
Dividing ratio of prescaler is selected.

Timer control register W1 (b3–b0) 0 1 0 ✕ — Timer 1 stop (**TW1A** instruction)
Prescaler stop
Prescaler divided by 16 selected

↓

③ Set Timer Value

Timer 1 count time is set. (The formula is shown ❋A below.)

Timer 1 reload register R1 "$F9_{16}$" — Timer count value 249 set (**T1AB** instruction)

↓

④ Clear Interrupt Request

Timer 1 interrupt activated condition is cleared.

Timer 1 interrupt request flag T1F "0" — Timer 1 interrupt activated condition cleared (**SNZT1** instruction)

↓

Note when the interrupt request is cleared

When ④ is executed, considering the skip of the next instruction according to the interrupt request flag T1F, insert the **NOP** instruction after the **SNZT1** instruction.

↓

⑤ Start Timer 1 Operation

Timer 1 and prescaler temporarily stopped are restarted.

Timer control register W1 (b3–b0) 1 1 1 ✕ — Timer 1 operation start (**TW1A** instruction)
Prescaler operation start

↓

⑥ Enable Interrupts

The timer 1 interrupt which is temporarily disabled is enabled.

Interrupt control register V1 (b3–b0) ✕ 1 ✕ ✕ — Timer 1 interrupt occurrence enabled (**TV1A** instruction)

Interrupt enable flag INTE "1" — All interrupts enabled (**EI** instruction)

↓

Constant period interrupt execution start

❋A The prescaler dividing ratio and timer 1 count value to make the interrupt occur every 4 ms are set as follows.

$$4\ \text{ms} \cong \underbrace{(4.0\ \text{MHz})^{-1}}_{\text{System clock}} \times \underbrace{3}_{\text{Instruction clock}} \times \underbrace{16}_{\text{Prescaler dividing ratio}} \times \underbrace{(249+1)}_{\text{Timer 1 count value}}$$

"✕": it can be "0" or "1."

Fig. 2.3.3 Constant period measurement setting example

① Disable Interrupts

Timer 1 interrupt is temporarily disabled.

Interrupt enable flag INTE "0" — All interrupts disabled (**DI** instruction)

Interrupt control register V1 (b3–b0): × 0 × × — Timer 1 interrupt occurrence disabled (**TV1A** instruction)

↓

② Stop Timer Operation

Timer 1 and prescaler are temporarily stopped.
Dividing ratio of prescaler is selected.

Timer control register W1 (b3–b0): 0 0 0 × — Timer 1 stop (**TW1A** instruction)
Prescaler stop
Prescaler divided by 4 selected

↓

③ Set Timer Value, Select CNTR0 Output

CNTR0 output is selected.
Timer 1 count time is set.

Timer control register W6 (b3–b0): × × 0 1 — CNTR0 output selected (**TW6A** instruction)

Timer 1 reload register R1 "29_{16}" — Timer count value 41 set (**T1AB** instruction)

↓

④ Clear Interrupt Request

Timer 1 interrupt activated condition is cleared.

Timer 1 interrupt request flag T1F "0" — Timer 1 interrupt activated condition cleared (**SNZT1** instruction)

↓

Note when the interrupt request is cleared

When ④ is executed, considering the skip of the next instruction according to the interrupt request flag T1F, insert the **NOP** instruction after the **SNZT1** instruction.

↓

⑤ Start Timer 1 Operation

Timer 1 and prescaler temporarily stopped are restarted.

Timer control register W1 (b3–b0): 1 0 1 × — Timer 1 operation start (**TW1A** instruction)
Prescaler operation start

↓

⑥ Enable Interrupts

The timer 1 interrupt which is temporarily disabled is enabled.

Interrupt control register V1 (b3–b0): × 1 × × — Timer 1 interrupt occurrence enabled (**TV1A** instruction)

Interrupt enable flag INTE "1" — All interrupts enabled (**EI** instruction)

↓

⑦ Stop CNTR0 Output

D_6/CNTR0 pin is set to CNTR0 input pin, and it is set to the high-impedance state.

Timer control register W6 (b3–b0): × × 0 0 — CNTR0 input pin set (**TW6A** instruction)

Output latch of port D_6 is set to "1." — (**SD** instruction)

"×": it can be "0" or "1."

Fig. 2.3.4 CNTR0 output setting example

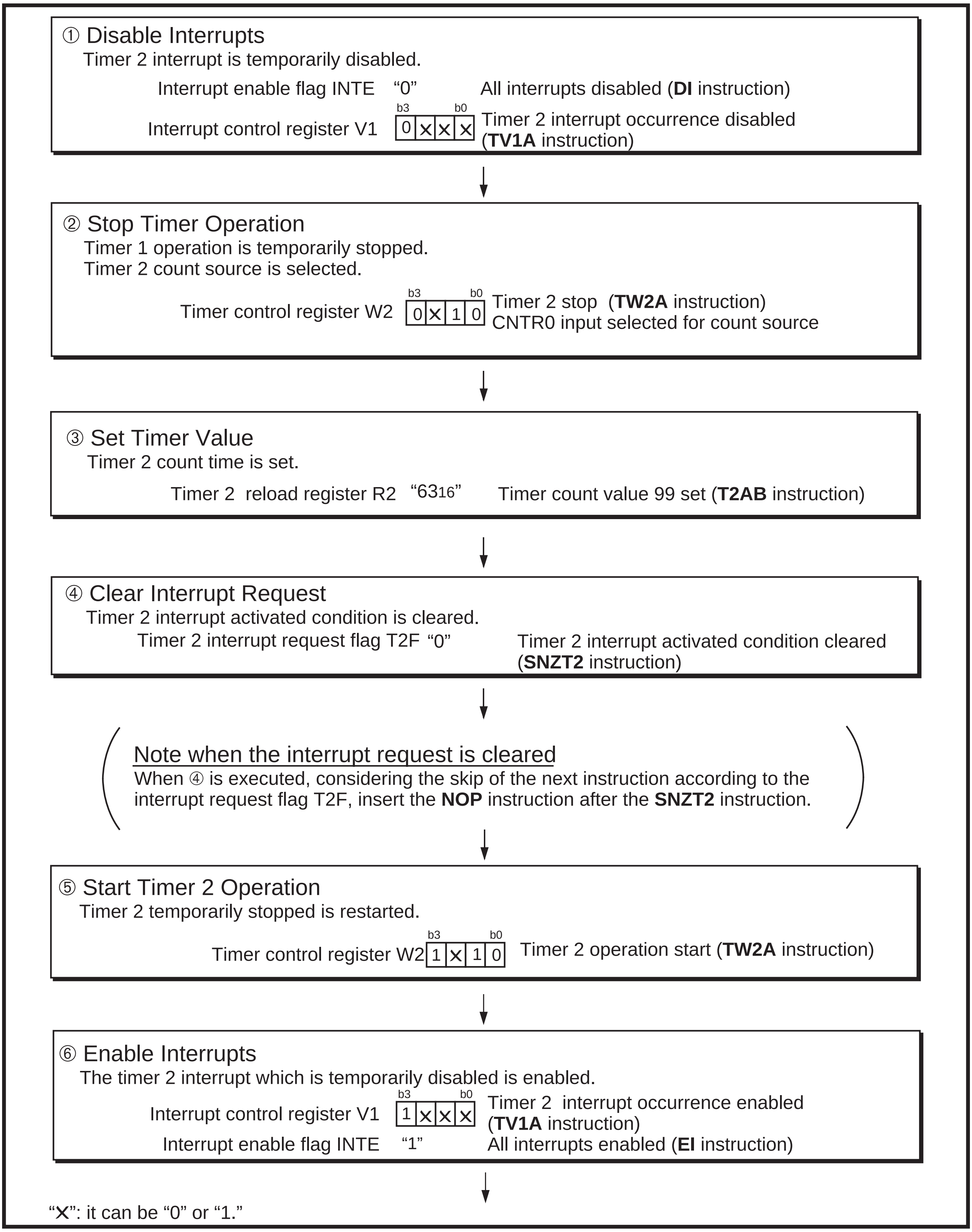


Fig. 2.3.5 CNTR1 input setting example

However, specify the pulse width input to CNTR0 pin/CNTR1 pin. Refer to section “**2.3.4 Notes on use**” for the timer external input period condition.

① Disable Interrupts

Timer 3 and timer 4 interrupt are temporarily disabled.

Interrupt enable flag INTE "0" All interrupts disabled (**DI** instruction)

Interrupt control register V2 (b3–b0) × × 0 0 Timer 3 and timer 4 interrupt occurrence disabled (**TV2A** instruction)

↓

② Stop Timer Operation

Timer is temporarily stopped.
Dividing ratio of prescaler is selected.
Timer 3 count source is selected.
Timer 4 count source is selected.

Timer control register W3 (b3–b0) 0 × 0 1 Timer 3 stop (**TW3A** instruction)
Prescaler selected for count source

Timer control register W4 (b3–b0) 0 × 0 0 Timer 4 stop (**TW4A** instruction)
Timer 3 underflow selected for count source

Timer control register W1 (b3–b0) 1 0 × × Instruction clock divided by 4 selected (**TW1A** instruction)

↓

③ Set Timer Value, Select CNTR1 Output

CNTR1 output is selected.
Timer 3 and timer 4 count time are set.

Timer control register W6 (b3–b0) 1 1 × × CNTR1 output selected (**TW6A** instruction)

Timer 3 reload register R3 "29_{16}" Timer count value 41 set (**T3AB** instruction)
Timer 3 reload register R4 "FF_{16}" Timer count value 255 set (**T4AB** instruction)

↓

④ Start Timer Operation

Timer 3 and timer 4 temporarily stopped are restarted.

Timer control register W3 (b3–b0) 1 × 0 1 Timer 3 operation start (**TW3A** instruction)

Timer control register W4 (b3–b0) 1 × 0 0 Timer 4 operation start (**TW4A** instruction)

↓

⑤ Enable Interrupts

Interrupt enable flag INTE "1" All interrupts enabled (**EI** instruction)

↓

"×": it can be "0" or "1."

Fig. 2.3.6 CNTR0 output control setting example

① Disable Interrupts

Timer 1 interrupt is temporarily disabled.

Interrupt enable flag INTE "0" — All interrupts disabled (**DI** instruction)

Interrupt control register V1 (b3–b0) × 0 × 0 — Timer 1 interrupt occurrence disabled (**TV1A** instruction); INT0 interrupt occurrence disabled

↓

② Stop Timer Operation

Timer 1 and prescaler are temporarily stopped.
Dividing ratio of prescaler is selected.

Timer control register W1 (b3–b0) 0 0 0 × — Timer 1 stop (**TW1A** instruction); Prescaler stop; Prescaler divided by 4 selected

↓

③ Set Timer Value

Timer 1 count time is set.

Timer 1 reload register R1 "52_{16}" — Timer count value 82 set (**T1AB** instruction)

↓

④ Set Port

$P3_0$/INT0 pin is set to INT0 input.

Port $P3_0$ output latch (b3–b0) × × × 1 — INT0 input set (**OP3A** instruction)

↓

⑤ Set Valid Waveform

Valid waveform of INT0 pin is selected. Timer 1 control is enabled.

Interrupt control register I1 (b3–b0) × 1 0 1 — Rising edge detected (**TI1A** instruction)

↓

⑥ Clear Interrupt Request

Timer 1 interrupt activated condition is cleared.
INT0 interrupt activated condition is cleared.

Timer 1 interrupt request flag T1F "0" — Timer 1 interrupt activated condition cleared (**SNZT1** instruction)

INT0 interrupt request flag EXF0 "0" — INT0 interrupt activated condition cleared (**SNZ0** instruction)

↓

(Note when the interrupt request is cleared
When ⑥ is executed, considering the skip of the next instruction according to the interrupt request flags T1F and EXF0, insert the **NOP** instruction after the **SNZT1** and **SNZ0** instructions.)

↓

⑦ Enable Interrupts

The timer 1 interrupt which is temporarily disabled is enabled.

Interrupt control register V1 (b3–b0) × 1 × × — Timer 1 interrupt occurrence enabled (**TV1A** instruction)

Interrupt enable flag INTE "1" — All interrupts enabled (**EI** instruction)

↓

Timer start by external input

"×": it can be "0" or "1."

Fig. 2.3.7 Timer start by external input setting example (1)

Processing in interrupt service routine

⑧ Stop Timer

Timer 1 control disabled   Interrupt control register I1   b3 [× | 1 | 0 | 0] b0   (**TI1A** instruction)

↓

⑨ Reset Timer

Timer 1 reload register R1 "52_{16}"   Timer count value 82 set (**T1AB** instruction)

Timer 1 control enabled   Interrupt control register I1   b3 [× | 1 | 0 | 1] b0   (**TI1A** instruction)

"×": it can be "0" or "1."

Fig. 2.3.8 Timer start by external input setting example (2)

① Activate Watchdog Timer

Watchdog timer is activated.

Watchdog timer enable flag WEF "1" — Watchdog timer enable flag WEF set (**WRST** instruction)

↓

Main routine (every 20 ms)

Reset Flag WDF

Watchdog timer flag WDF1 is reset. "0" — Watchdog timer flag WDF1 cleared (**WRST** instruction)

↓

Main routine execution

↓

Repeat

Do not clear watchdog timer WDF flag in interrupt service routine.
Interrupt may be executed even if program run-away occurs.

When going to RAM back-up mode

⋮

WRST ; WDF flag cleared
EPOF ; **POF** instruction enabled
POF
↓
Oscillation stop (RAM back-up mode)

In the RAM back-up mode, WEF, WDF1 and WDF2 flags are initialized. However, when WDF2 flag is set to "1", at the same time, system enters RAM back-up mode, microcomputer may be reset. When watchdog timer and RAM back-up mode are used, execute the **WRST** instruction before system enters the RAM back-up mode to initialize WDF flag.

"✕": it can be "0" or "1."

Fig. 2.3.9 Watchdog timer setting example

2.3.4 Notes on use

(1) Prescaler
Stop the prescaler operation to change its frequency dividing ratio.

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

(3) Reading the count values
Stop timer 1, 2, 3, or 4 counting and then execute the **TAB1**, **TAB2**, **TAB3**, or **TAB4** instruction to read its data.

(4) Writing to reload registers R1, R3
When writing data to reload registers R1, R3 while timer 1 and 3 are operating, avoid a timing when timers 1 and 3 underflow.

2.4 Serial I/O

The 4513/4514 Group has a clock-synchronous serial I/O which can be used to transmit and receive 8-bit data.
This section describes serial I/O functions, related registers, application examples using serial I/O and notes.

2.4.1 Carrier functions

Serial I/O consists of the serial I/O register SI, serial I/O mode register J1, serial I/O transmit/receive completion flag SIOF and serial I/O counter.
A clock-synchronous serial I/O uses the shift clock generated by the clock control circuit as a synchronous clock. Accordingly, the data transmit and receive operations are synchronized with this shift clock.
In transmit operation, data is transmitted bit by bit from the SOUT pin synchronously with the falling edges of the shift clock.
In receive operation, data is received bit by bit from the SIN pin synchronously with the rising edges of the shift clock.

Note: 4513/4514 Group only supports LSB-first transmission and reception.

■ **Shift clock**
When using the internal clock of 4513/4514 Group as a synchronous clock, eight shift clock pulses are output from the SCK pin when a transfer operation is started. Also, when using some external clock as a synchronous clock, the clock that is input from the SCK pin is used as the shift clock.

■ **Data transfer rate (baudrate)**
When using the internal clock, the data transfer rate can be determined by selecting the instruction clock divided by 4 or 8.
When using an external clock, the clock frequency input to the SCK pin determines the data transfer rate.

Figure 2.4.1 shows the serial I/O block diagram.

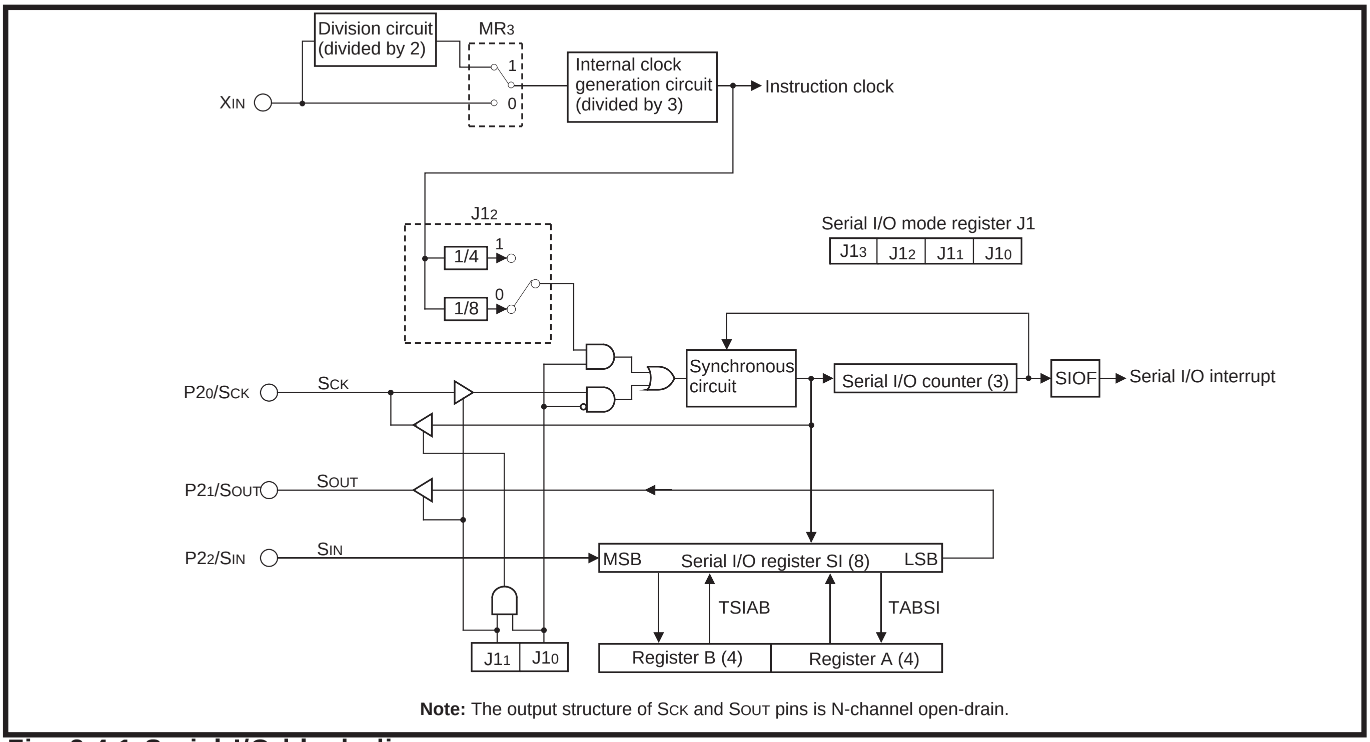


Fig. 2.4.1 Serial I/O block diagram

2.4.2 Related registers

(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.

(2) Serial I/O mode register J1

Serial I/O synchronous clock selection bit is assigned to bit 0, serial I/O port selection bit is assigned to bit 1 and serial I/O internal clock dividing ratio selection bit is assigned to bit 2.
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.

Table 2.4.1 shows the serial I/O mode register J1.

Table 2.4.1 Serial I/O mode register J1

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 S_{CK}, S_{OUT}, S_{IN}/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.

(3) 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**).

2.4.3 Operation description

Figure 2.4.2 shows the serial I/O connection example, Figure 2.4.3 shows the serial I/O register state, and Figure 2.4.4 shows the serial I/O transfer timing.

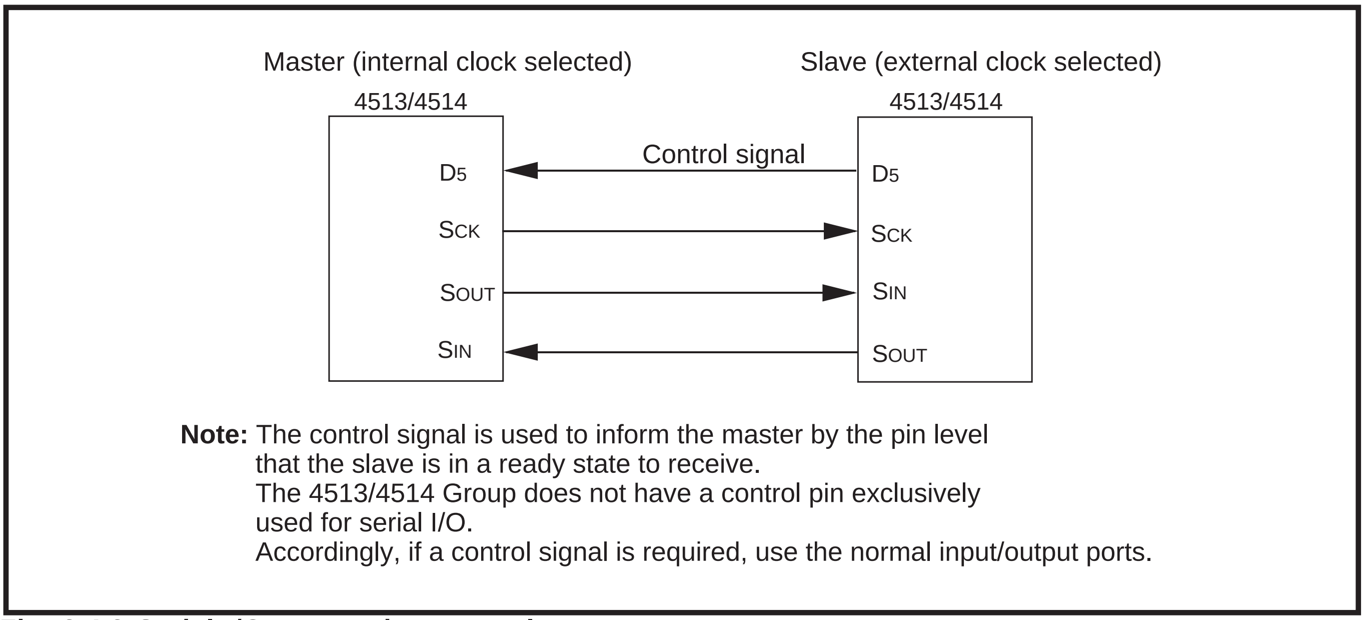


Fig. 2.4.2 Serial I/O connection example

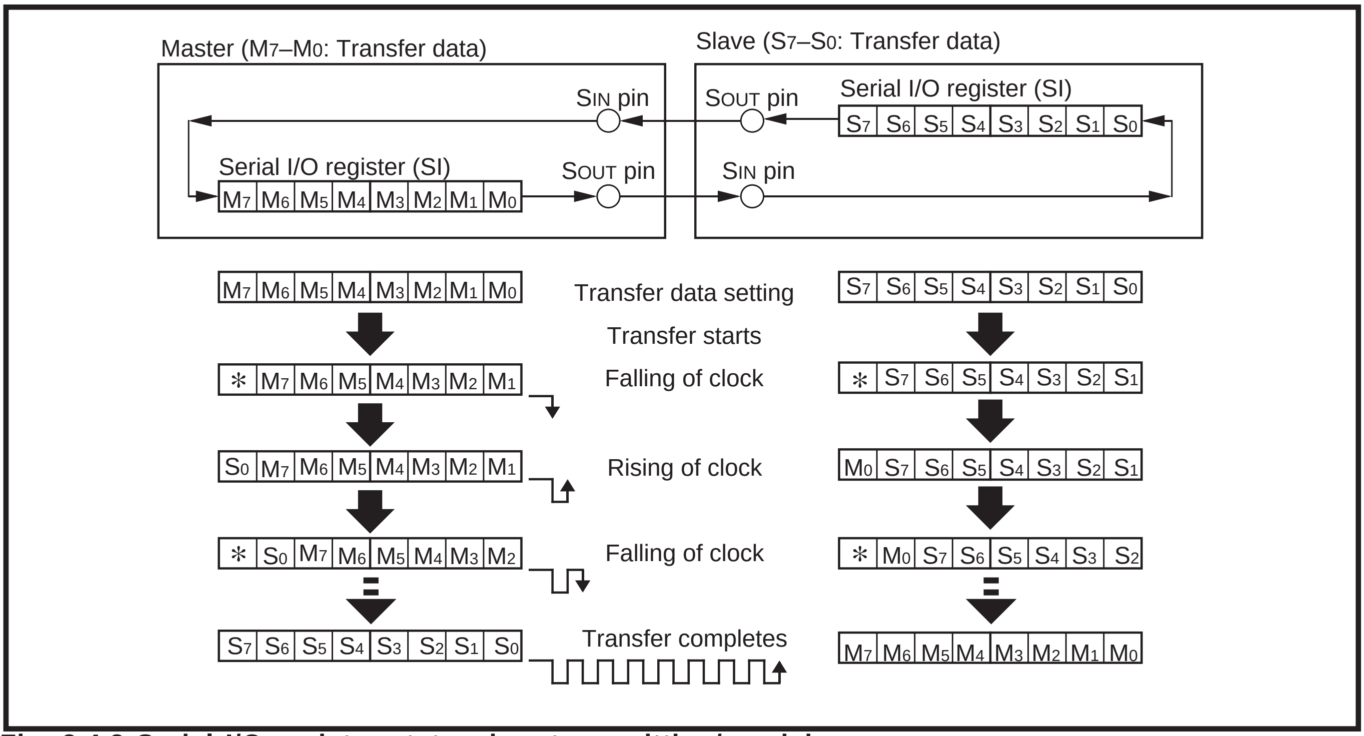


Fig. 2.4.3 Serial I/O register state when transmitting/receiving

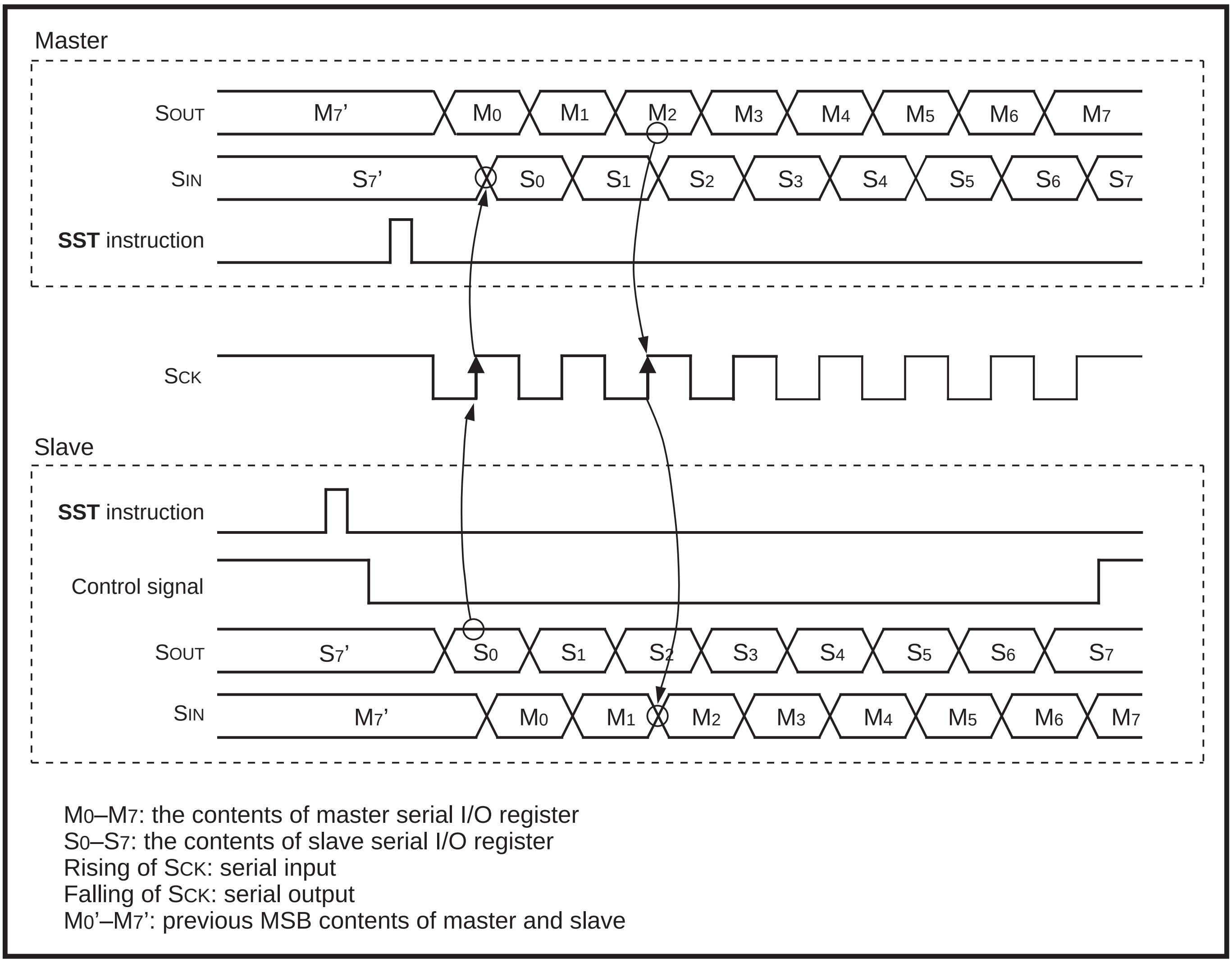


Fig. 2.4.4 Serial I/O transfer timing

The full duplex communication of master and slave is described using the connection example shown in Figure 2.4.2.

(1) Transmit/receive operation of master

① The transmit data is written into the serial I/O register SI with the **TSIAB** instruction. When the **TSIAB** instruction is executed, the contents of register A are transferred to the low-order 4 bits of register SI and the contents of register B are transferred to the high-order 4 bits of register SI.

② Whether the microcomputer on the receiving side is ready to receive or not is checked. In the connection example in Figure 2.4.2, check that the input level of control signal is “L” level.

③ Serial transfer is started with the **SST** instruction.
When the **SST** instruction is executed, the serial I/O transmit/receive completion flag (SIOF) is cleared to “0.”

④ The transmit data is output from the SOUT pin synchronously with the falling edges of the shift clock.

⑤ The transmit data is output bit by bit beginning with the LSB bit of register SI. Each time one bit is output, the contents of register SI is shifted one bit position toward the LSB.

⑥ Also, the receive data is input from the SIN pin synchronously with the rising edges of the shift clock.

⑦ The receive data is input bit by bit to the MSB bit of register SI.

⑧ A serial I/O interrupt request occurs when the transfer of transmit data and receive data is completed, and the SIOF flag is set to “1.”

⑨ The receive data is taken in within the serial I/O interrupt service routine; or the data is taken in after examining the completion of the transmit/receive operation with the **SNZSI** instruction without using an interrupt.
Also, the SIOF flag is cleared to “0” when an interrupt occurs or the **SNZSI** instruction is executed.

Notes 1: Repeat steps ① through ⑨ to transmit or receive multiple data in succession.
2: For the program on the master side, make sure that transmission is not started before the control signal is released back “H” after a transmit operation is started first.

(2) Transmit/receive operation of slave

① The transmit data is written into the serial I/O register SI with the **TSIAB** instruction. When the **TSIAB** instruction is executed, the contents of register A are transferred to the low-order bits of register SI and the contents of register B are transferred to the high-order bits of register SI. At this time, the SCK pin must be at the “H” level.

② Serial transfer is started with the **SST** instruction. However, in Figure 2.4.2 where an external clock is selected, transfer is not started until the clock is input. When the **SST** instruction is executed, the serial I/O transmit/receive completion flag (SIOF) is cleared to “0.”

③ The microcomputer on the transmitting side is informed that the receiving side is ready to receive. In the connection example in Figure 2.4.2, this notification is done by pulling the control signal “L” level.

④ The transmit data is output from the SOUT pin synchronously with the falling edges of the shift clock.

⑤ The transmit data is output bit by bit beginning with the LSB bit of register SI. Each time one bit is output, the contents of register SI are shifted to one bit position toward the LSB.

⑥ Also, the receive data is input from the SIN pin synchronously with the rising edges of the shift clock.

⑦ The receive data is input bit by bit to the MSB bit of register SI.

⑧ A serial I/O interrupt request occurs when the transmit/receive of data is completed, and the SIOF flag is set to “1.”

⑨ The receive data is taken in within the serial I/O interrupt service routine; or the data is taken in after examining the completion of the transmit/receive operation with the **SNZSI** instruction without using an interrupt.
Also, the SIOF flag is cleared to “0” when an interrupt occurs or the **SNZSI** instruction is executed.
Make sure that the control signal pin level is “H” after the receive operation is completed.

Note: Repeat steps ① through ⑨ to transmit or receive multiple data in succession.

2.4.4 Serial I/O application example

(1) Serial I/O

Outline: The 4513/4514 Group can communicate with peripheral ICs.
Specifications: Figure 2.4.2 Serial I/O connection example.

Figure 2.4.5 shows the master serial I/O setting example, and Figure 2.4.6 shows the slave serial I/O setting example.

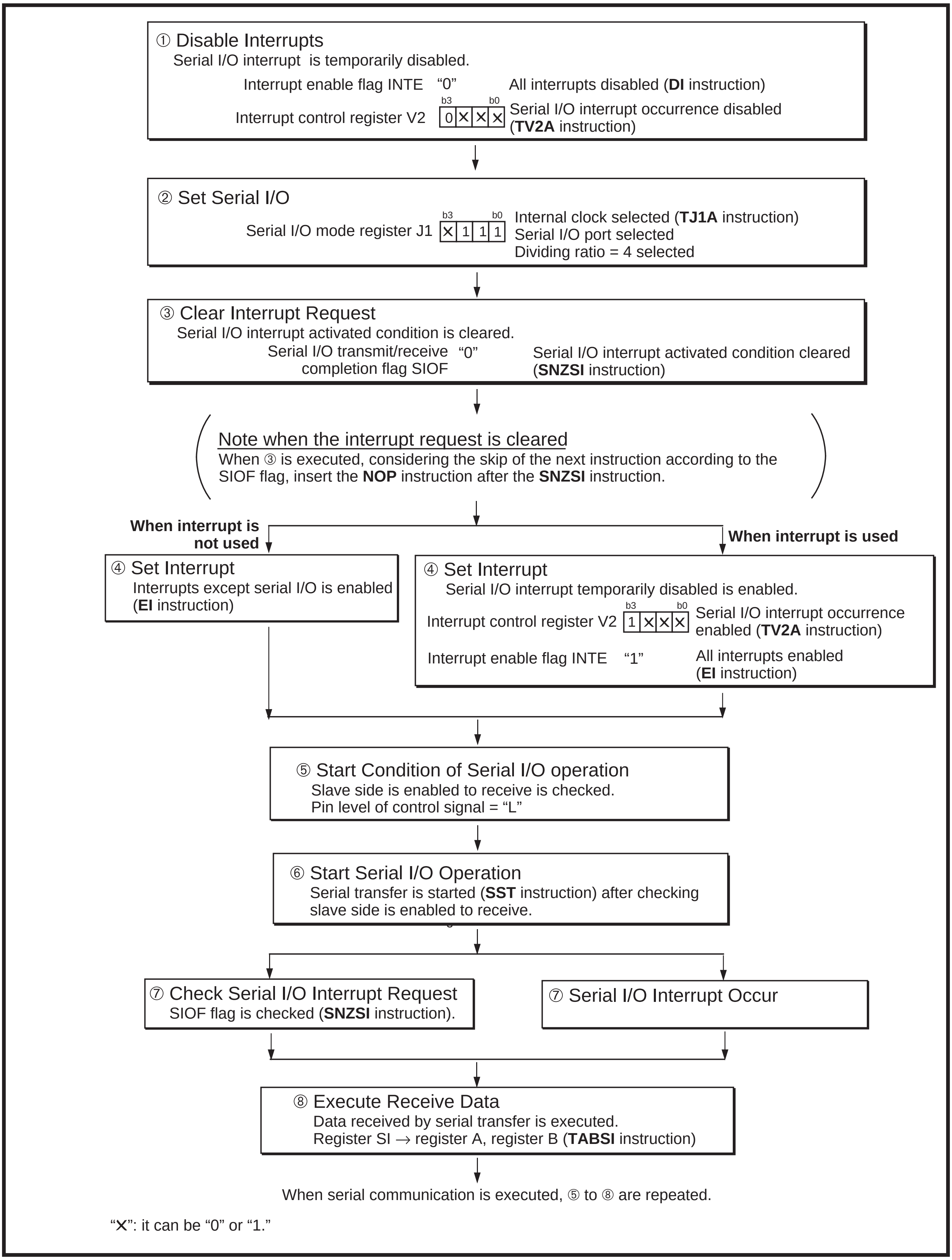


Fig. 2.4.5 Master serial I/O setting example

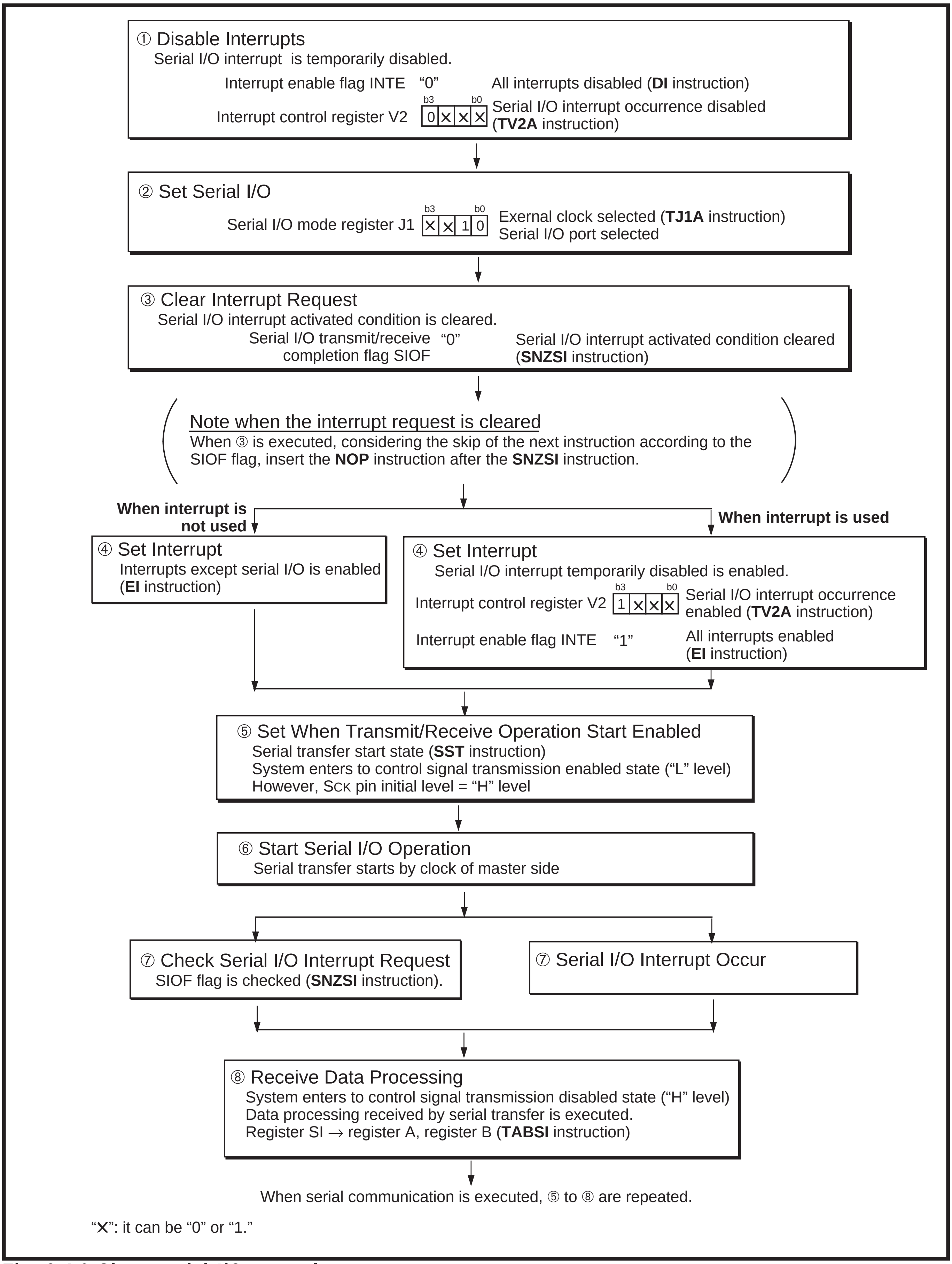


Fig. 2.4.6 Slave serial I/O example

2.4.5 Notes on use

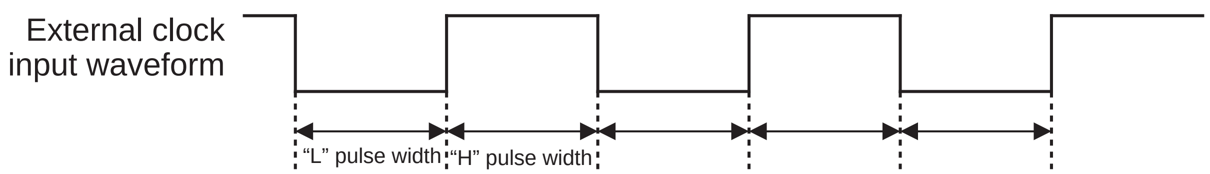
(1) Note when an external clock is used as a synchronous clock:

- An external clock is selected as the synchronous clock, the clock is not controlled internally.
- Serial transfer is continued as long as an external clock is input. If an external clock is input 9 times or more and serial transfer is continued, the receive data is transferred directly as transmit data, so that be sure to control the clock externally.
  Note also that the SIOF flag is set when a clock is counted 8 times.
- Make sure that the initial input level on the external clock pin is always “H” level.
- Table 2.4.2 shows the recommended operating conditions when using serial I/O with an external clock. Figure 2.4.7 shows an input waveform of external clock.

Table 2.4.2 Recommended operating conditions (serial I/O)

Parameter	Condition		Limits			Unit
			Min.	Typ.	Max.	
Serial I/O external clock period (**Note 1**)	Middle-speed mode	V_{DD} = 4.0 V to 5.5 V	1.5			µs
		V_{DD} = 2.5 V to 5.5 V	3.0			
		V_{DD} = 2.0 V to 5.5 V (**Note 2**)	4.0			
	High-speed mode	V_{DD} = 4.0 V to 5.5 V	750			ns
		V_{DD} = 2.5 V to 5.5 V	1.5			µs
		V_{DD} = 2.0 V to 5.5 V (**Note 2**)	2.0			

Notes 1: Limits shown in Table 2.4.2 represent the pulse widths of “H” and “L.”
2: It is effective only for mask version.



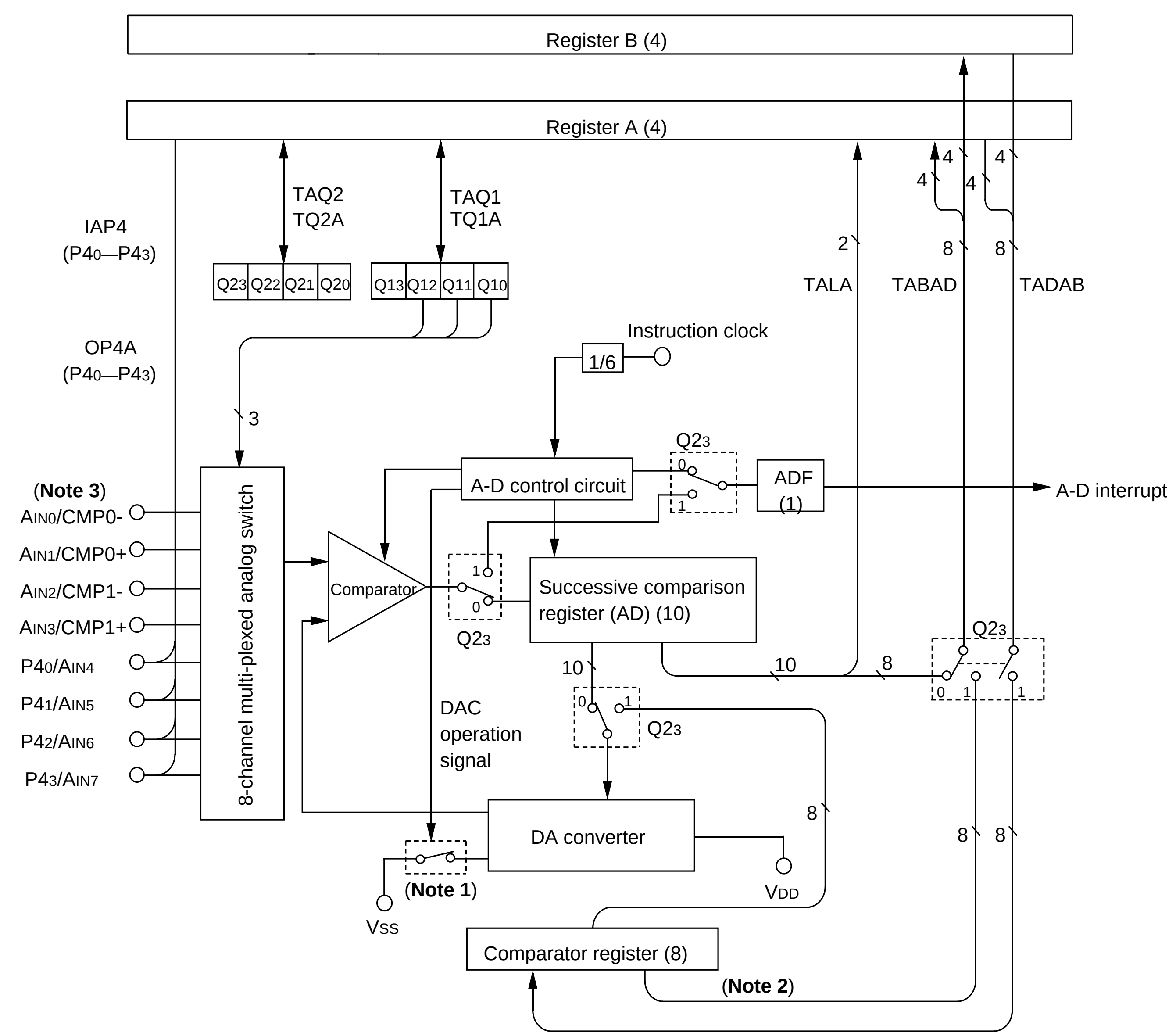
Note: Set “H” and “L” pulse width for external waveform according to using supply voltage and recommended operating conditions.

Fig. 2.4.7 Input waveform of external clock

2.5 A-D converter

The 4513/4514 Group has an A-D converter with the 10-bit successive comparison method: 4 channels for the 4513 Group, 8 channels for the 4514 Group.
This A-D converter can also be used as a comparator to compare analog voltages input from the analog input pin with preset values.
This section describes the related registers, application examples using the A-D converter and notes.

Figure 2.5.1 shows the A-D converter block diagram.



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 ($Q2_3$=1).
The value of the comparator register is retained even when the mode is switched to the A-D conversion mode ($Q2_3$=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.
3: The 4513 Group does not have ports $P4_0$/A_{IN4}–$P4_3$/A_{IN7} and the IAP4 and OP4A instructions.

Fig. 2.5.1 A-D converter structure

2.5 A-D converter

2.5.1 Related registers

(1) A-D control register Q1

Analog input pin selection bits are assigned to register Q1.
Set the contents of this register through register A with the **TQ1A** instruction. The **TAQ1** instruction can be used to transfer the contents of register Q1 to register A.
Table 2.5.1 shows the A-D control register Q1.

Table 2.5.1 A-D control register Q1

A-D control register Q1		at reset : 0000_2			at power down : state retained	R/W
$Q1_3$	Not used	0			This bit has no function, but read/write is enabled.	
		1				
		$Q1_2$	$Q1_1$	$Q1_0$	Selected pin	
$Q1_2$	Analog input pin selection bits (**Note 2**)	0	0	0	AIN0	
		0	0	1	AIN1	
$Q1_1$		0	1	0	AIN2	
		0	1	1	AIN3	
		1	0	0	AIN4 (Not available for 4513 Group)	
$Q1_0$		1	0	1	AIN5 (Not available for 4513 Group)	
		1	1	0	AIN6 (Not available for 4513 Group)	
		1	1	1	AIN7 (Not available for 4513 Group)	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: Select AIN4–AIN7 with register Q1 after setting register Q2.

(2) A-D control register Q2

Analog input pin selection bits and A-D operation mode control bit are assigned to register Q2.
Set the contents of this register through register A with the **TQ2A** instruction. The **TAQ2** instruction can be used to transfer the contents of register Q2 to register A.
Table 2.5.2 shows the A-D control register Q2.

Table 2.5.2 A-D control register Q2

A-D control register Q2		at reset : 0000_2	at power down : state retained	R/W
$Q2_3$	A-D operation mode control bit	0	A-D conversion mode	
		1	Comparator mode	
$Q2_2$	$P4_3$/AIN7, $P4_2$/AIN6 pin function selection bit (**Note 3**)	0	$P4_3$, $P4_2$ (I/O) (**Note 4**)	
		1	AIN7, AIN6/$P4_3$, $P4_2$ (Output) (**Note 4**)	
$Q2_1$	$P4_1$/AIN5 pin function selection bit (**Note 3**)	0	$P4_1$ (I/O) (**Note 4**)	
		1	AIN5/$P4_1$ (Output) (**Note 4**)	
$Q2_0$	$P4_0$/AIN4 pin function selection bit (**Note 3**)	0	$P4_0$ (I/O) (**Note 4**)	
		1	AIN4/$P4_0$ (Output) (**Note 4**)	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: Select AIN4–AIN7 with register Q1 after setting register Q2.
3: In the 4513 Group, these bits are not used.
4: In the 4513 Group, only read/write of these bits is enabled.

2.5.2 A-D converter application examples

(1) A-D conversion mode

Outline: Analog input signal from a sensor can be converted into digital values.
Specifications: Analog voltage values from a sensor is converted into digital values by using a 10-bit successive comparison method. Use the AIN0 pin for this analog input.

Figure 2.5.2 shows the A-D conversion mode setting example.

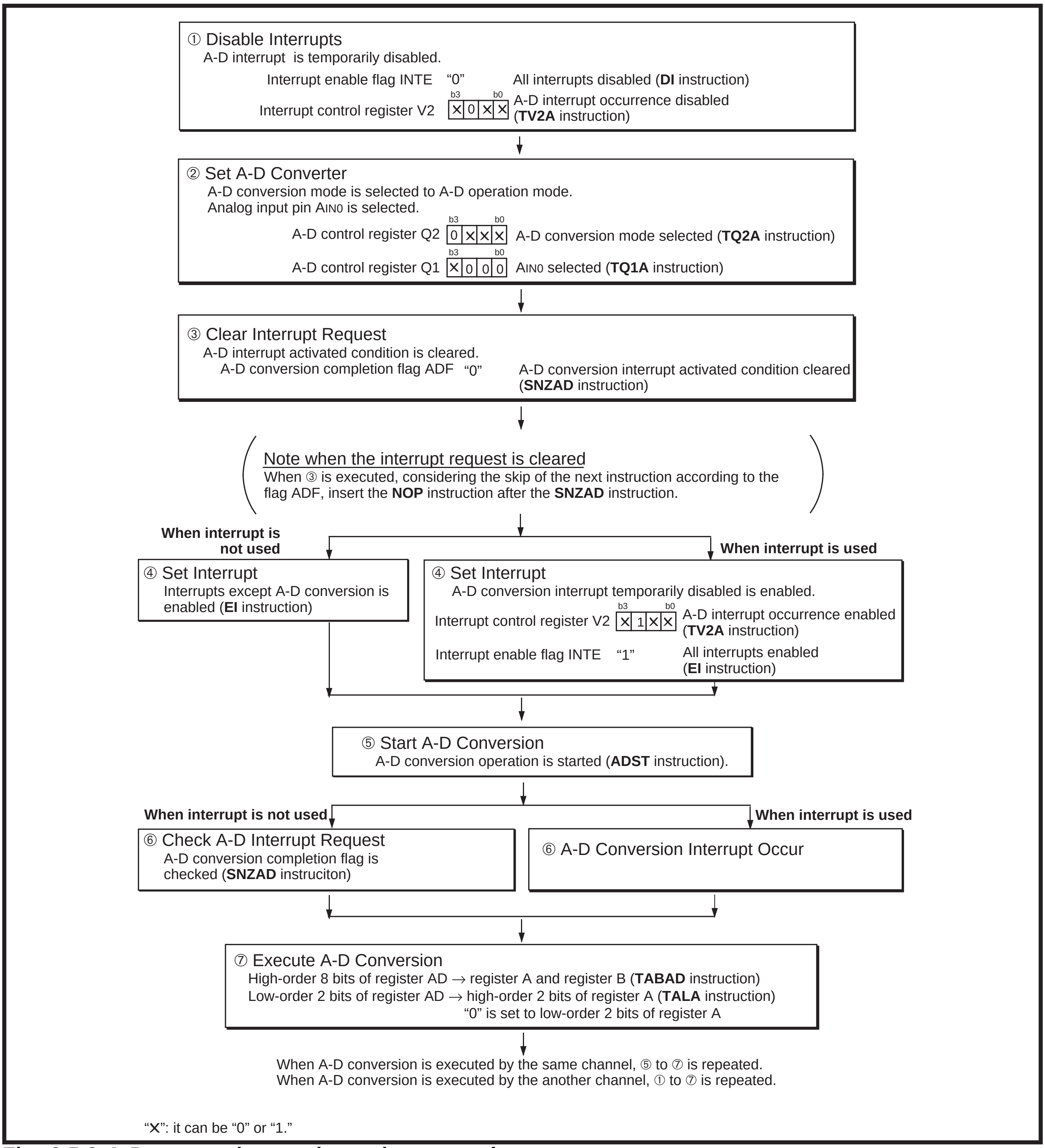


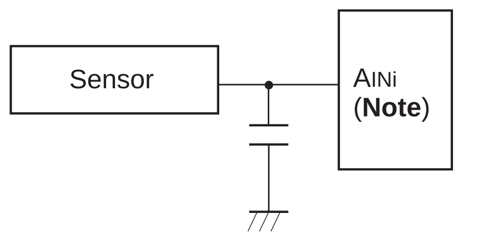
Fig. 2.5.2 A-D conversion mode setting example

2.5.3 Notes on use

(1) Note when the A-D conversion starts again
When the A-D conversion starts again with the **ADST** instruction during A-D conversion, the previous input data is invalidated and the A-D conversion starts again.

(2) A-D control register Q2
Select AIN4–AIN7 with register Q1 after setting register Q2.

(3) A-D converter-1
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 2.5.3 shows the analog input external circuit example-1.
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 2.5.4. In addition, test the application products sufficiently.



Note: Apply the voltage within the specifications to an analog input pin. (i = 0 to 7)

Fig. 2.5.3 Analog input external circuit example-1

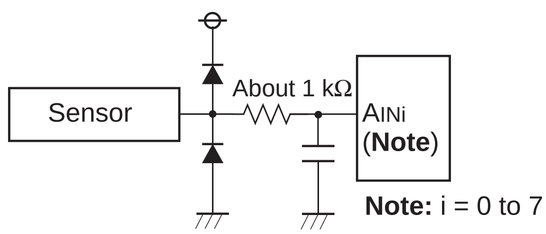


Fig. 2.5.4 Analog input external circuit example-2

(4) Notes for the use of A-D conversion 2
When the operating mode of the A-D converter is changed from the comparator mode to the A-D conversion mode with bit 3 of register Q2 in a program, be careful about the following notes.
- Clear 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 bit 3 of register Q2 (refer to Figure 2.5.5①).
- 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 bit 3 of register Q2 during operating the A-D converter.

⋮
Clear bit 2 of register V2 to “0”.......①
↓
Change of the operating mode of the A-D converter from the comparator mode to the A-D conversion mode
↓
Clear the ADF flag to “0” with the **SNZAD** instruction
↓
Execute the **NOP** instruction for the case when a skip is performed with the **SNZAD** instruction
⋮

Fig. 2.5.5 A-D converter operating mode program example

(5) A-D converter is used at the comparator mode

The analog input voltage is higher than the comparison voltage as a result of comparison, the contents of ADF flag retains “0,” not set to “1.”
In this case, the A-D interrupt does not occur even when the usage of the A-D interrupt is enabled. Accordingly, consider the time until the comparator operation is completed, and examine the state of ADF flag by software. The comparator operation is completed after 8 machine cycles.

(6) Analog input pins

Even when $P4_0/A_{IN4}$–$P4_3/A_{IN7}$ are set to pins for analog input, they continue to function as $P4_0$–$P4_3$ I/O. Accordingly, when any of them are used as I/O port P4 and others are used as analog input pins, make sure to set the outputs of pins that are set for analog input to “1.” Also, the port input function of the pin functions as an analog input is undefined.

(7) 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, and simultaneously, the low-order 2 bits of register A is “0.”

(8) Recommended operating conditions when using A-D converter

The recommended operating conditions of supply voltage and system clock frequency when using A-D converter are different from those when not using A-D converter.
Table 2.5.3 shows the recommended operating conditions when using A-D converter.

Table 2.5.3 Recommended operating conditions (when using A-D converter)

Parameter	Condition		Limits Min.	Limits Typ.	Limits Max.	Unit
System clock frequency (at ceramic resonance)	V_{DD} = 4.5 V to 5.5 V (high-speed mode)		0.4		4.2	MHz
	V_{DD} = 4.0 V to 5.5 V (high-speed mode)		0.4		2.0	MHz
	V_{DD} = 2.7 V to 5.5 V (middle-speed mode)		0.4		4.2	MHz
System clock frequency (at external clock input)	V_{DD} = 4.5 V to 5.5 V (high-speed mode)	Duty 40 % to 60 %	0.4		3.0	MHz
	V_{DD} = 4.0 V to 5.5 V (high-speed mode)		0.4		1.0	MHz
	V_{DD} = 2.7 V to 5.5 V (middle-speed mode)		0.4		3.0	MHz

2.6 Voltage comparator

The 4513/4514 Group has two voltage comparators; CMP0-, CMP0+, CMP1-, CMP1+.
This section describes the voltage comparator function, related registers, and notes.

2.6.1 Voltage comparator function

(1) CMP0

■ **Voltage comparison**

The voltage of CMP0- is compared with that of CMP0+, and the result is stored into bit 0 of the voltage comparator control register Q3.

(2) CMP1

■ **Voltage comparison**

The voltage of CMP1- is compared with that of CMP1+, and the result is stored into bit 1 of the voltage comparator control register Q3.

2.6.2 Related registers

(1) Voltage comparator control register Q3

The voltage comparator (CMP1) control bit is assigned to bit 3, the voltage comparator (CMP0) control bit is assigned to bit 2, the CMP1 comparison result store bit is assigned to bit 1 and the CMP0 comparison result store bit is assigned to bit 0.
Set the contents of this register through register A with the **TQ3A** instruction. The **TAQ3** instruction can be used to transfer the contents of register Q3 to register A.
Table 2.6.1 shows the voltage comparator control register Q3.

Table 2.6.1 Voltage comparator control register Q3

Voltage comparator control register Q3 (**Note 2**)		at reset : 0000_2	at RAM back-up : state retained	R/W
$Q3_3$	Voltage comparator (CMP1) control bit	0	Voltage comparator (CMP1) invalid	
		1	Voltage comparator (CMP1) valid	
$Q3_2$	Voltage comparator (CMP0) control bit	0	Voltage comparator (CMP0) invalid	
		1	Voltage comparator (CMP0) valid	
$Q3_1$	CMP1 comparison result store bit	0	CMP1- > CMP1+	
		1	CMP1- < CMP1+	
$Q3_0$	CMP0 comparison reslut store bit	0	CMP0- > CMP0+	
		1	CMP0- < CMP0+	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: Bits 0 and 1 of register Q3 can be only read.

2.6.3 Notes on use

- **Voltage comparator function**
  When the voltage comparator function is valid with the voltage comparator control register Q3, it is operating even in the RAM back-up mode. Accordingly, be careful about such state because it causes the increase of the operation current in the RAM back-up mode.
  In order to reduce the operation current in the RAM back-up mode, invalidate (bits 2 and 3 of register Q3 = "0") the voltage comparator function by software before the **POF** instruction is executed.
  Also, while the voltage comparator function is valid, current is always consumed by voltage comparator. On the system required for the low-power dissipation, invalidate the voltage comparator when it is unused by software.

- **Register Q3**
  Bits 0 and 1 of register Q3 can be only read. Note that they cannot be written.

- **Reading the comparison result of voltage comparator**
  Read the voltage comparator comparison result from register Q3 after the voltage comparator response time (max. 20 μs) is passed from the voltage comparator function become valid.

2.7 Reset

System reset is performed by applying "L" level to the RESET pin for 1 machine cycle or more when the following conditions are satisfied:

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

Then when "H" level is applied to RESET pin, the software starts from address 0 in page 0 after elapsing of the internal oscillation stabilizing time (f(XIN) is counted for 16892 to 16895 machine cycles). Figure 2.7.2 shows the oscillation stabilizing time.

2.7.1 Reset circuit

The 4513/4514 Group has the power-on reset circuit and voltage drop detection circuit.

(1) Power-on reset

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

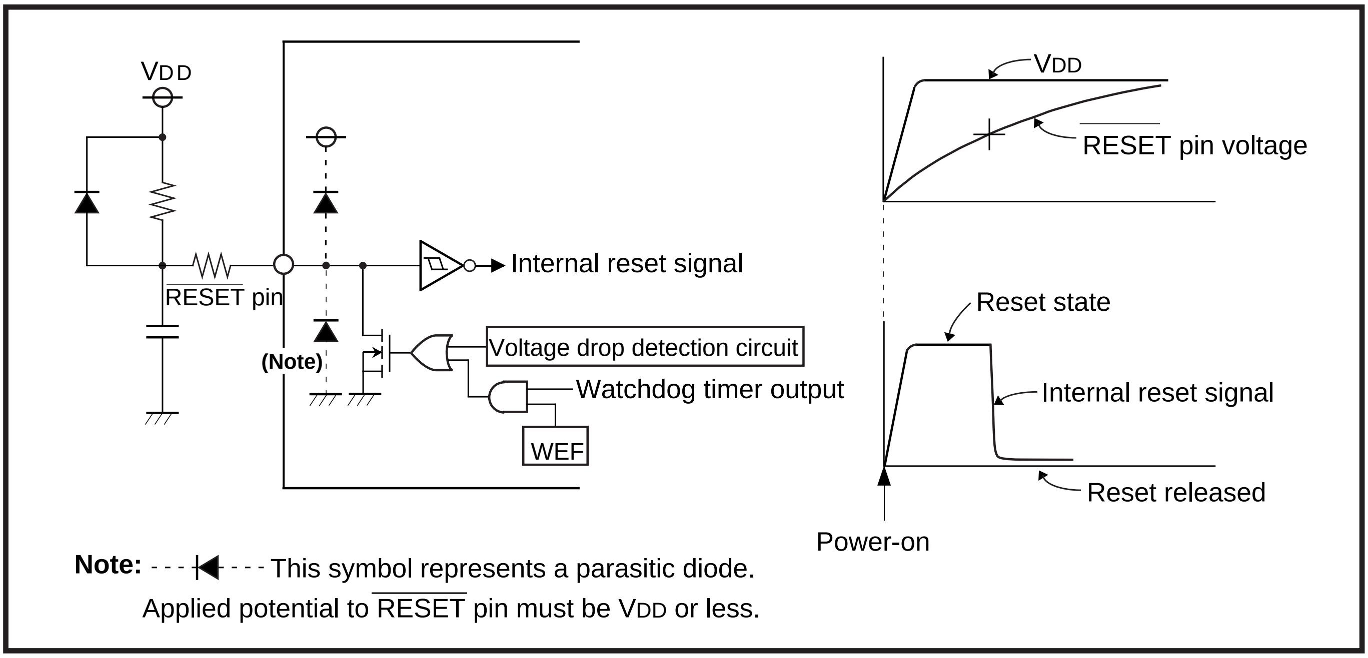


Fig. 2.7.1 Power-on reset circuit example

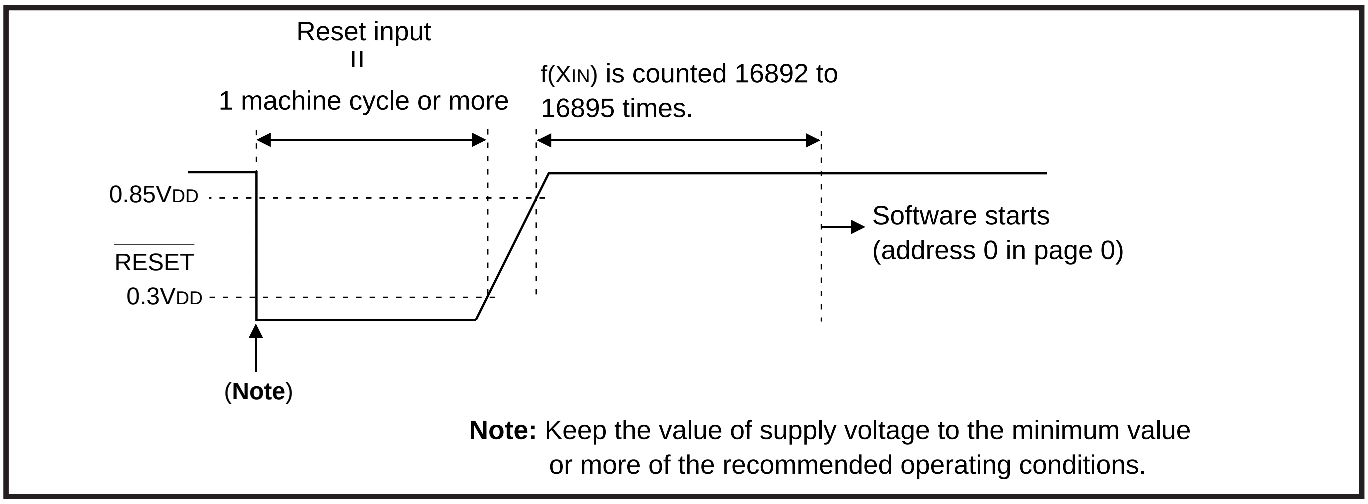


Fig. 2.7.2 Oscillation stabilizing time after system is released from reset

2.7.2 Internal state at reset

Figure 2.7.3 shows the internal state at reset. The contents of timers, registers, flags and RAM other than shown in Figure 2.7.3 are undefined, so that set them to initial values.

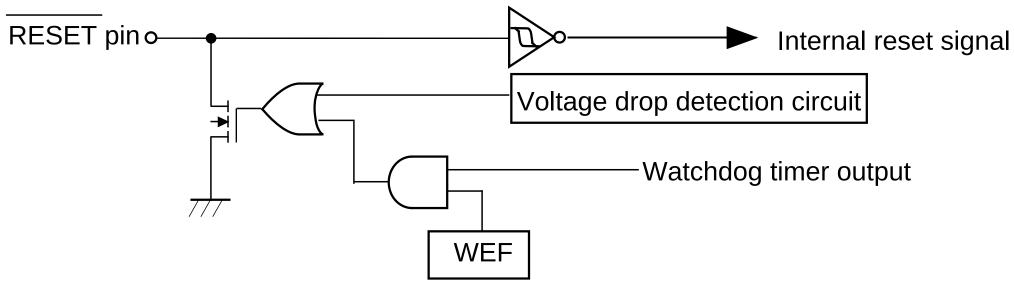
• Program counter (PC)	0 0 0 0 0 0 0 0 0 0 0 0 0 0	
Address 0 in page 0 is set to program counter.		
• Interrupt enable flag (INTE)	0	(Interrupt disabled)
• Power down flag (P)	0	
• External 0 interrupt request flag (EXF0)	0	
• External 1 interrupt request flag (EXF1)	0	
• Interrupt control register 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, 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)
• Timer control register W6	0 0 0 0	
• Clock control register MR	1 0 0 0	
• Serial I/O transmit/receive completion flag	0	
• Serial I/O mode register J1	0 0 0 0	(External clock selected, 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	
• Voltage comparator control register Q3	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	
• Direction register FR0	0 0 0 0	(Port P5 input mode)
• 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. 2.7.3 Internal state at reset

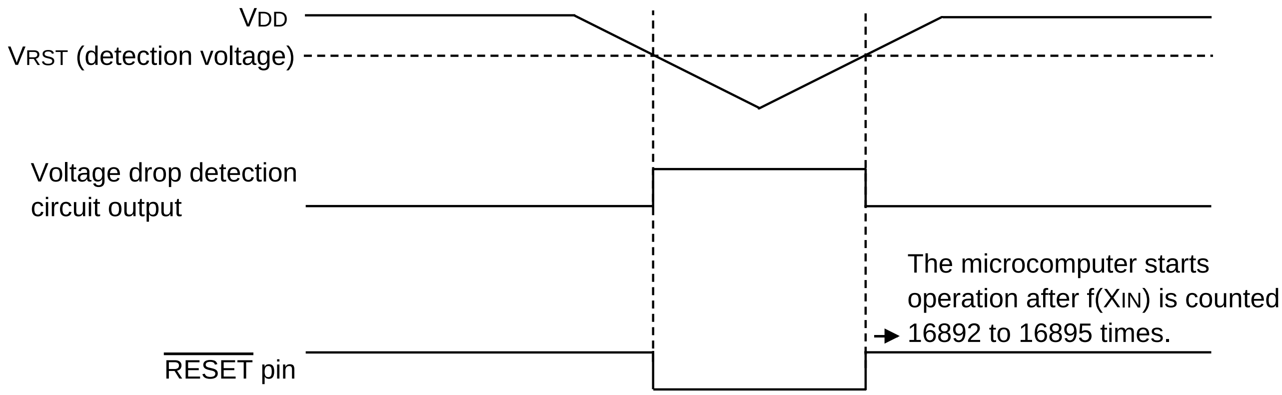
2.8 Voltage drop detection circuit

The built-in voltage drop detection circuit is designed to detect a drop in voltage and to reset the microcomputer if the supply voltage drops below a set value.
Figure 2.8.1 shows the voltage drop detection reset circuit, and Figure 2.8.2 shows the operation waveform example of the voltage drop detection circuit.



Note: The output structure of $\overline{\text{RESET}}$ pin is N-channel open-drain.

Fig. 2.8.1 Voltage drop detection reset circuit



Notes 1: Pull-up $\overline{\text{RESET}}$ pin externally.
2: Refer to the voltage drop detection circuit in the electrical characteristics for the rating value of V_{RST} (detection voltage).

Fig. 2.8.2 Voltage drop detection circuit operation waveform

Note: Refer to section "**3.1 Electrical characteristics**" for the reset voltage of the voltage drop detection circuit.

2.9 RAM back-up

2.9.1 RAM back-up mode

The system enters RAM back-up mode when the **POF** instruction is executed after the **EPOF** instruction is executed. Table 2.9.1 shows the function and state retained at RAM back-up mode. Also, Table 2.9.2 shows the return source from this state.

(1) RAM back-up mode

As oscillation stops with RAM, the state of reset circuit retained, current dissipation can be reduced without losing the contents of RAM.

Table 2.9.1 Functions and states retained at RAM back-up mode

Function	RAM back-up	Function	RAM back-up
Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) (**Note 2**)	X	Pull-up control register PU0	O
		Key-on wakeup control register K0	O
Contents of RAM	O	Direction register FR0	O
Port level	O	External 0 interrupt request flag (EXF0)	X
Timer control register W1	X	External 1 interrupt request flag (EXF1)	X
Timer control registers W2 to W4. W6	O	Timer 1 interrupt request flag (T1F)	X
Clock control register MR	X	Timer 2 interrupt request flag (T2F)	(**Note 3**)
Interrupt control registers V1, V2	X	Timer 3 interrupt request flag (T3F)	(**Note 3**)
Interrupt control registers I1, I2	O	Timer 4 interrupt request flag (T4F)	(**Note 3**)
Timer 1 function	X	Watchdog timer flags (WDF1, WDF2)	X (**Note 4**)
Timer 2 function	(**Note 3**)	Watchdog timer enable flag (WEF)	X (**Note 4**)
Timer 3 function	(**Note 3**)	16-bit timer (WDT)	X (**Note 4**)
Timer 4 function	(**Note 3**)	A-D conversion completion flag (ADF)	X
A-D function	X	Serial I/O transmit/receive completion flag (SIOF)	X
A-D control registers Q1, Q2	O		
Voltage comparator function	O (**Note 5**)	Interrupt enable flag (INTE)	X
Voltage comparator control register Q3	O		
Serial I/O function	X		
Serial I/O mode register J1	O		

Notes 1: "O" represents that the function can be retained, and "X" represents that the function is initialized. Registers and flags other than the above are undefined at RAM back-up, and set an initial value after returning.

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

3: The state of the timer is undefined.

4: Initialize the watchdog timer with the **WRST** instruction, and then execute the **POF** instruction.

5: The state is retained when the voltage comparator function is selected with the voltage comparator control register Q3.

2.9 RAM back-up

Table 2.9.2 Return source and return condition

Return source		Return condition	Remarks
External wakeup signal	Ports P0, P1	Return by an external falling edge input (“H”→“L”).	Set the port using the key-on wakeup function selected with register K0 to “H” level before going into the RAM back-up state because the port P0 shares the falling edge detection circuit with port P1.
	Port $P3_0$/INT0	Return by an external “H” level or “L” level input. The EXF0 flag is not set.	Select the return level (“L” level or “H” level) with the bit 2 of register I1 according to the external state before going into the RAM back-up state.
	Port $P3_1$/INT1	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.

(2) Start condition identification

When system returns from both RAM back-up mode and reset, software is started from address 0 in page 0.

The start condition (warm start or cold start) can be identified by examining the state of the power down flag (P) with the **SNZP** instruction.

Table 2.9.3 Start condition identification

Return condition	P flag
External wakeup signal input	1
Reset	0

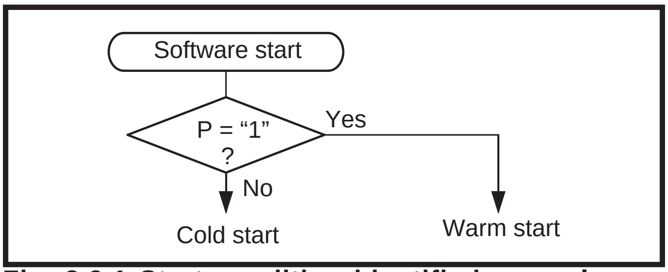


Fig. 2.9.1 Start condition identified example

2.9.2 Related register

(1) Key-on wakeup control register K0

Key-on wakeup control register K0 controls key-on wakeup functions of ports $P0_0$–$P0_3$, $P1_0$–$P1_3$.

Set the contents of this register through register A with the **TK0A** instruction. The **TAK0** instruction can be used to transfer the contents of register K0 to register A.

Table 2.9.4 shows the key-on wakeup control register K0.

Table 2.9.4 Key-on wakeup control register K0

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	

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

(2) Pull-up control register PU0

Pull-up control register PU0 controls the pull-up functions of ports $P0_0$–$P0_3$, $P1_0$–$P1_3$.
Set the contents of this register through register A with the **TPU0A** instruction. The **TAPU0** instruction can be used to transfer the contents of register PU0 to register A.
Table 2.9.5 shows the pull-up control register PU0.

Table 2.9.5 Pull-up control register PU0

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	

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

(3) Interrupt control register I1

The interrupt valid waveform for INT0 pin/return level selection bit is assigned to bit 2, INT0 pin edge detection circuit control bit is assigned to bit 1, and INT0 pin timer 1 control enable bit is assigned to bit 0.
Set the contents of this register through register A with the **TI1A** instruction.
In addition, the **TAI1** instruction can be used to transfer the contents of register I1 to register A.
Table 2.9.6 shows the interrupt control register I1.

Table 2.9.6 Interrupt control register I1

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	

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.

(4) Interrupt control register I2

The interrupt valid waveform for INT1 pin/return level selection bit is assigned to bit 2, the INT1 pin edge detection circuit control bit is assigned to bit 1, and the INT1 pin timer 1 control enable bit is assigned to bit 1.

Set the contents of this register through register A with the **TI2A** instruction.

In addition, the **TAI2** instruction can be used to transfer the contents of register I2 to register A.

Table 2.9.7 shows the interrupt control register I2.

Table 2.9.7 Interrupt control register I2

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 2**)	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 $I2_2$ is changed, the external interrupt request flag EXF1 may be set. Accordingly, clear EXF1 flag with the **SNZ1** instruction.

2.9.3 Notes on use

(1) Key-on wakeup function

After setting ports (P1 specified with register PU0 and P0) which key-on wakeup function is valid to “H,” execute the **POF** instruction.

“L” level is input to the falling edge detection circuit even if one of ports which key-on wakeup function is valid is in the “L” level state, and the edge is not detected.

(2) POF instruction

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

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

Be sure to disable interrupts by executing the **DI** instruction before executing the **EPOF** instruction.

(3) Return from RAM back-up

After system returns from RAM back-up, set the undefined registers and flags.

Especially, be sure to set data pointer (registers Z, X, Y).

2.10 Oscillation circuit

The 4513/4514 Group has an internal oscillation circuit to produce the clock required for microcomputer operation.
The clock signal f(XIN) is obtained by connecting a ceramic resonator to XIN pin and XOUT pin.

2.10.1 Oscillation circuit

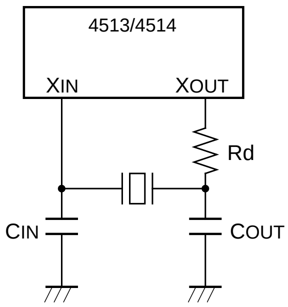
(1) f(XIN) clock generating circuit

The clock signal f(XIN) is obtained by connecting a ceramic resonator externally.
Connect this external circuit to pins XIN and XOUT at the shortest distance. A feed-back resistor is built-in between XIN pin and XOUT pin.
Figure 2.10.1 shows an example of an oscillation circuit connecting a ceramic resonator externally.
Keep the maximum value of oscillation frequency within the range listed Table 2.10.1.

Table 2.10.1 Maximum value of oscillation frequency and supply voltage

Supply voltage	(System clock)		Oscillation frequency
2.5 V to 5.5 V	(f(XIN)/2)	Middle-speed mode	4.2 MHz
4.0 V to 5.5 V	(f(XIN))	High-speed mode	4.2 MHz
2.5 V to 5.5 V	(f(XIN))	High-speed mode	2.0 MHz
2.0 V to 5.5 V (**Note**)	(f(XIN)/2)	Middle-speed mode	3.0 MHz
2.0 V to 5.5 V (**Note**)	(f(XIN))	High-speed mode	1.5 MHz

Note: 2.5 V to 5.5 V for the One Timer PROM version.



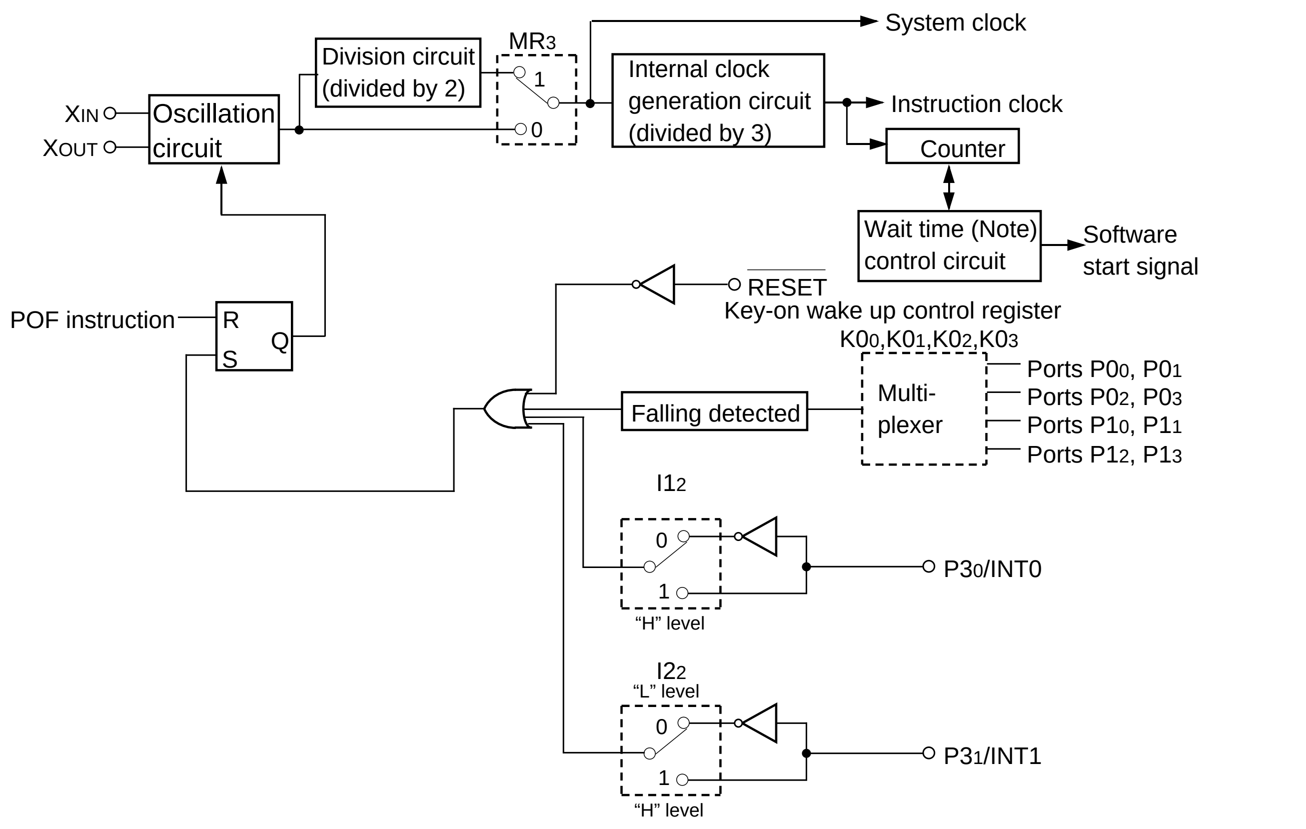
Note: Externally connect a damping resistor Rd depending on the oscillation frequency. (A feed-back resistor is built-in.) Use the resonator manufacturer's recommended value because constants such as capacitance depend on the resonator.

Fig. 2.10.1 Oscillation circuit example connecting ceramic resonator externally

2.10.2 Oscillation operation

System clock is supplied to CPU and peripheral device as the standard clock for the microcomputer operation. For the 4513/4514 Group, the clock (f(XIN)), (f(XIN)/2) which is supplied from the oscillation circuit is selected with the register MR.

Figure 2.10.2 shows the structure of the clock control circuit.



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

Fig. 2.10.2 Structure of clock control circuit

2.10.3 Notes on use

(1) Value of a part connected to an oscillator

Values of a capacitor and a resistor of the oscillation circuit depend on the connected oscillator and the board. Accordingly, consult the oscillator manufacturer for values of each part connected the oscillator.

CHAPTER 3
APPENDIX

3.1 Electrical characteristics
3.2 Typical characteristics
3.3 List of precautions
3.4 Notes on noise
3.5 Mask ROM confirmation form
3.6 Mark specification form
3.7 Package outline

3.1 Electrical characteristics

3.1.1 Absolute maximum ratings

Table 3.1.1 Absolute maximum ratings

Symbol	Parameter	Conditions		Ratings	Unit
V_{DD}	Supply voltage			–0.3 to 7.0	V
V_I	Input voltage P0, P1, P2, P3, P4, P5, $\overline{RESET}$, X_{IN}, VDCE			–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_{IN7}			–0.3 to V_{DD}+0.3	V
V_O	Output voltage P0, P1, P3, P4, P5, $\overline{RESET}$	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
P_d	Power dissipation	Ta = 25 °C	Package: 42P2R	300	mW
			Package: 32P6B	300	
			Package: 32P4B	1100	
Topr	Operating temperature range			–20 to 85	°C
Tstg	Storage temperature range			–40 to 125	°C

3.1.2 Recommended operating conditions

Table 3.1.2 Recommended operating conditions 1

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

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{DD}	Supply voltage	Mask ROM version Middle-speed mode	$f(X_{IN}) \le 4.2$ MHz	2.5		5.5	V
			$f(X_{IN}) \le 3.0$ MHz	2.0		5.5	
		Mask ROM version High-speed mode	$f(X_{IN}) \le 4.2$ MHz	4.0		5.5	
			$f(X_{IN}) \le 2.0$ MHz	2.5		5.5	
			$f(X_{IN}) \le 1.5$ MHz	2.0		5.5	
		One Time PROM version Middle-speed mode	$f(X_{IN}) \le 4.2$ MHz	2.5		5.5	
		One Time PROM version High-speed mode	$f(X_{IN}) \le 4.2$ MHz	4.0		5.5	
			$f(X_{IN}) \le 2.0$ MHz	2.5		5.5	
V_{RAM}	RAM back-up voltage (at RAM back-up mode)	Mask ROM version		1.8			V
		One Time PROM version		2.0			
V_{SS}	Supply voltage				0		V
V_{IH}	"H" level input voltage	P0, P1, P2, P3, P4, P5, X_{IN}, VDCE		$0.8V_{DD}$		V_{DD}	V
V_{IH}	"H" level input voltage	D_0–D_7		$0.8V_{DD}$		12	V
V_{IH}	"H" level input voltage	$\overline{RESET}$		$0.85V_{DD}$		V_{DD}	V
V_{IH}	"H" level input voltage	CNTR0, CNTR1, S_{IN}, S_{CK}, INT0, INT1		$0.85V_{DD}$		V_{DD}	V
V_{IL}	"L" level input voltage	P0, P1, P2, P3, P4, P5, D_0–D_7, X_{IN}, VDCE		0		$0.2V_{DD}$	V
V_{IL}	"L" level input voltage	$\overline{RESET}$		0		$0.3V_{DD}$	V
V_{IL}	"L" level input voltage	CNTR0, CNTR1, S_{IN}, S_{CK}, INT0, INT1		0		$0.15V_{DD}$	V
I_{OH}(peak)	"H" level peak output current	P5	V_{DD} = 5.0 V	–20			mA
			V_{DD} = 3.0 V	–10			
I_{OH}(avg)	"H" level average output current	P5 (Note)	V_{DD} = 5.0 V	–10			mA
			V_{DD} = 3.0 V	–5			
I_{OL}(peak)	"L" level peak output current	P3, $\overline{RESET}$	V_{DD} = 5.0 V			10	mA
			V_{DD} = 3.0 V			4	
I_{OL}(peak)	"L" level peak output current	D_6, D_7	V_{DD} = 5.0 V			40	mA
			V_{DD} = 3.0 V			30	
I_{OL}(peak)	"L" level peak output current	D_0–D_5	V_{DD} = 5.0 V			24	mA
			V_{DD} = 3.0 V			12	
I_{OL}(peak)	"L" level peak output current	P0, P1, P4, P5, S_{CK}, S_{OUT}	V_{DD} = 5.0 V			24	mA
			V_{DD} = 3.0 V			12	
I_{OL}(avg)	"L" level average output current	P3, $\overline{RESET}$ (Note)	V_{DD} = 5.0 V			5	mA
			V_{DD} = 3.0 V			2	
I_{OL}(avg)	"L" level average output current	D_6, D_7 (Note)	V_{DD} = 5.0 V			30	mA
			V_{DD} = 3.0 V			15	
I_{OL}(avg)	"L" level average output current	D_0–D_5 (Note)	V_{DD} = 5.0 V			15	mA
			V_{DD} = 3.0 V			7	
I_{OL}(avg)	"L" level average output current	P0, P1, P4, P5, S_{CK}, S_{OUT} (Note)	V_{DD} = 5.0 V			12	mA
			V_{DD} = 3.0 V			6	
ΣI_{OH}(avg)	"H" level total average current	P5		–30			mA
ΣI_{OL}(avg)	"L" level total average current	P5, D, $\overline{RESET}$, S_{CK}, S_{OUT}				80	
		P0, P1, P3, P4				80	

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

Table 3.1.3 Recommended operating conditions 2

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

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
$f(X_{IN})$	Oscillation frequency (with a ceramic resonator)	Mask ROM version Middle-speed mode	V_{DD} = 2.5 V to 5.5 V			4.2	MHz
			V_{DD} = 2.0 V to 5.5 V			3.0	
		One Time PROM version Middle-speed mode	V_{DD} = 2.5 V to 5.5 V			4.2	
		Mask ROM version High-speed mode	V_{DD} = 4.0 V to 5.5 V			4.2	
			V_{DD} = 2.5 V to 5.5 V			2.0	
			V_{DD} = 2.0 V to 5.5 V			1.5	
		One Time PROM version High-speed mode	V_{DD} = 4.0 V to 5.5 V			4.2	
			V_{DD} = 2.5 V to 5.5 V			2.0	
$f(X_{IN})$	Oscillation frequency (with external clock input)	Mask ROM version Middle-speed mode	V_{DD} = 2.0 V to 5.5 V			3.0	MHz
		One Time PROM version Middle-speed mode	V_{DD} = 2.5 V to 5.5 V			3.0	
		Mask ROM version High-speed mode	V_{DD} = 4.0 V to 5.5 V			3.0	
			V_{DD} = 2.5 V to 5.5 V			1.0	
			V_{DD} = 2.0 V to 5.5 V			0.8	
		One Time PROM version High-speed mode	V_{DD} = 4.0 V to 5.5 V			3.0	
			V_{DD} = 2.5 V to 5.5 V			1.0	
$t_w(S_{CK})$	Serial I/O external clock period (“H” and “L” pulse width)	Mask ROM version Middle-speed mode	V_{DD} = 4.0 V to 5.5 V	1.5			µs
			V_{DD} = 2.5 V to 5.5 V	3.0			
			V_{DD} = 2.0 V to 5.5 V	4.0			
		One Time PROM version Middle-speed mode	V_{DD} = 4.0 V to 5.5 V	1.5			
			V_{DD} = 2.5 V to 5.5 V	3.0			
		Mask ROM version High-speed mode	V_{DD} = 4.0 V to 5.5 V	750			ns
			V_{DD} = 2.5 V to 5.5 V	1.5			µs
			V_{DD} = 2.0 V to 5.5 V	2.0			
		One Time PROM version High-speed mode	V_{DD} = 4.0 V to 5.5 V	750			ns
			V_{DD} = 2.5 V to 5.5 V	1.5			µs
$t_w(CNTR)$	Timer external input period (“H” and “L” pulse width)	Mask ROM version Middle-speed mode	V_{DD} = 4.0 V to 5.5 V	1.5			µs
			V_{DD} = 2.5 V to 5.5 V	3.0			
			V_{DD} = 2.0 V to 5.5 V	4.0			
		One Time PROM version Middle-speed mode	V_{DD} = 4.0 V to 5.5 V	1.5			
			V_{DD} = 2.5 V to 5.5 V	3.0			
		Mask ROM version High-speed mode	V_{DD} = 4.0 V to 5.5 V	750			ns
			V_{DD} = 2.5 V to 5.5 V	1.5			µs
			V_{DD} = 2.0 V to 5.5 V	2.0			
		One Time PROM version High-speed mode	V_{DD} = 4.0 V to 5.5 V	750			ns
			V_{DD} = 2.5 V to 5.5 V	1.5			µs

3.1.3 Electrical characteristics

Table 3.1.4 Electrical characteristics

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

Symbol	Parameter		Test conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	"H" level output voltage P5		V_{DD} = 5 V	I_{OH} = –10 mA	3			V
			V_{DD} = 3 V	I_{OH} = –5 mA	2			
V_{OL}	"L" level output voltage P0, P1, P4, P5		V_{DD} = 5 V	I_{OL} = 12 mA			2	V
			V_{DD} = 3 V	I_{OL} = 6 mA			0.9	
V_{OL}	"L" level output voltage P3, $\overline{\text{RESET}}$		V_{DD} = 5 V	I_{OL} = 5 mA			2	V
			V_{DD} = 3 V	I_{OL} = 2 mA			0.9	
V_{OL}	"L" level output voltage D_6, D_7		V_{DD} = 5 V	I_{OL} = 30 mA			2	V
				I_{OL} = 10 mA			0.9	
			V_{DD} = 3 V	I_{OL} = 15 mA			2	V
				I_{OL} = 5 mA			0.9	
V_{OL}	"L" level output voltage D_0–D_5		V_{DD} = 5 V	I_{OL} = 15 mA			2	V
			V_{DD} = 3 V	I_{OL} = 3 mA			0.9	
I_{IH}	"H" level input current P0, P1, P2, P3, P4, P5, $\overline{\text{RESET}}$, VDCE		V_I = V_{DD}, port P4 selected, port P5: input state				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, P3, P4, P5, $\overline{\text{RESET}}$, VDCE		V_I = 0 V No pull-up of ports P0 and P1, port P4 selected, port P5: input state		–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 Middle-speed mode	$f(X_{IN})$ = 4.0 MHz		1.8	5.5	mA
				$f(X_{IN})$ = 400 kHz		0.5	1.5	
			V_{DD} = 3 V Middle-speed mode	$f(X_{IN})$ = 4.0 MHz		0.9	2.7	
				$f(X_{IN})$ = 400 kHz		0.2	0.6	
			V_{DD} = 5 V High-speed mode	$f(X_{IN})$ = 4.0 MHz		3.0	9.0	
				$f(X_{IN})$ = 400 kHz		0.6	1.8	
			V_{DD} = 3 V High-speed mode	$f(X_{IN})$ = 2.0 MHz		0.9	2.7	
				$f(X_{IN})$ = 400 kHz		0.3	0.9	
		at RAM back-up mode	Ta = 25 °C			0.1	1	µA
			V_{DD} = 5 V				10	
			V_{DD} = 3 V				6	
R_{PU}	Pull-up resistor value		V_{DD} = 5 V	V_I = 0 V	20	50	125	kΩ
			V_{DD} = 3 V		40	100	250	
$V_{T+} – V_{T–}$	Hysteresis INT0, INT1, CNTR0, CNTR1, S_{IN}, S_{CK}		V_{DD} = 5 V			0.3		V
			V_{DD} = 3 V			0.3		
$V_{T+} – V_{T–}$	Hysteresis $\overline{\text{RESET}}$		V_{DD} = 5 V			1.5		V
			V_{DD} = 3 V			0.6		

3.1.4 A-D converter recommended operating conditions

Table 3.1.5 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		2.7		5.5	V
V_{IA}	Analog input voltage		0		V_{DD}	V
$f(X_{IN})$	Oscillation frequency	Middle-speed mode, $V_{DD} \geq 2.7$ V	0.8			MHz
		High-speed mode, $V_{DD} \geq 2.7$ V	0.4			MHz

Table 3.1.6 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, V_{DD} = 2.7 V to 5.5 V				±2	LSB
		Ta = –25 °C to 85 ° C, V_{DD} = 3.0 V to 5.5 V					
–	Differential non-linearity error	Ta = 25 °C, V_{DD} = 2.7 V to 5.5 V				±0.9	LSB
		Ta = –25 °C to 85 ° C, V_{DD} = 3.0 V to 5.5 V					
V_{0T}	Zero transition voltage	V_{DD} = 5.12 V		0	5	20	mV
		V_{DD} = 3.072 V		0	3	15	
V_{FST}	Full-scale transition voltage	V_{DD} = 5.12 V		5105	5115	5125	mV
		V_{DD} = 3.072 V		3060	3069	3075	
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
		V_{DD} = 3.0 V	$f(X_{IN})$ = 0.4 MHz to 2.0 MHz		0.2	0.4	
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
		V_{DD} = 3.072 V				±15	
–	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 ideal 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)

3.1.5 Voltage drop detection circuit characteristics

Table 3.1.7 Voltage drop detection circuit characteristics

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{RST}	Detection voltage		2.7		4.1	V
		Ta = 25 °C	3.3	3.5	3.7	
I_{RST}	Operation current of voltage drop detection circuit	V_{DD} = 5.0 V		50	100	µA

3.1.6 Voltage comparator characteristics

Table 3.1.8 Voltage comparator recommended operating conditions

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

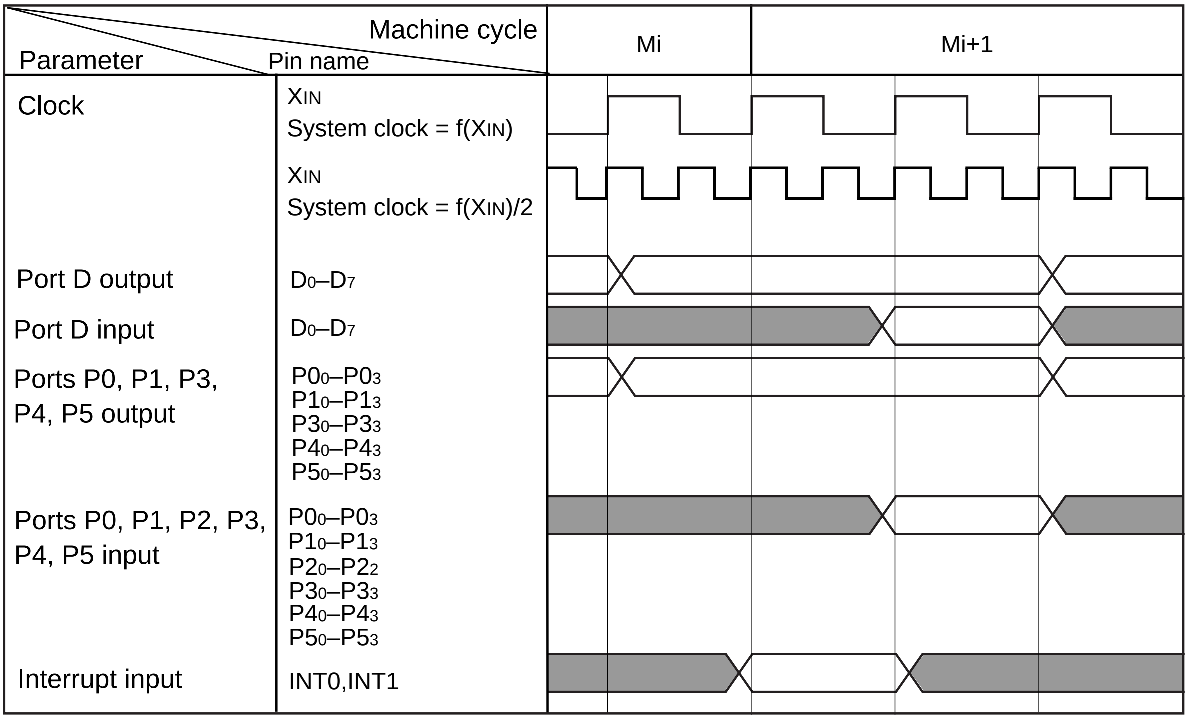
Symbol	Parameter	Conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{DD}	Supply voltage		3.0		5.5	V
V_{INCMP}	Voltage comparator input voltage	V_{DD} = 3.0 V to 5.5 V	0.3V_{DD}		0.7V_{DD}	V
t_{CMP}	Voltage comparator response time	V_{DD} = 3.0 V to 5.5 V			20	µs

Table 3.1.9 Voltage comparator characteristics

(Ta = –20 °C to 85 °C, V_{DD} = 3.0 V to 5.5 V, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits Min.	Limits Typ.	Limits Max.	Unit
–	Comparison decision voltage error	CMP0- > CMP0+, CMP0- < CMP0+ CMP1- > CMP1+, CMP1- < CMP1+		20	100	mV
I_{CMP}	Voltage comparator operation current	V_{DD} = 5.0 V		15	50	µA

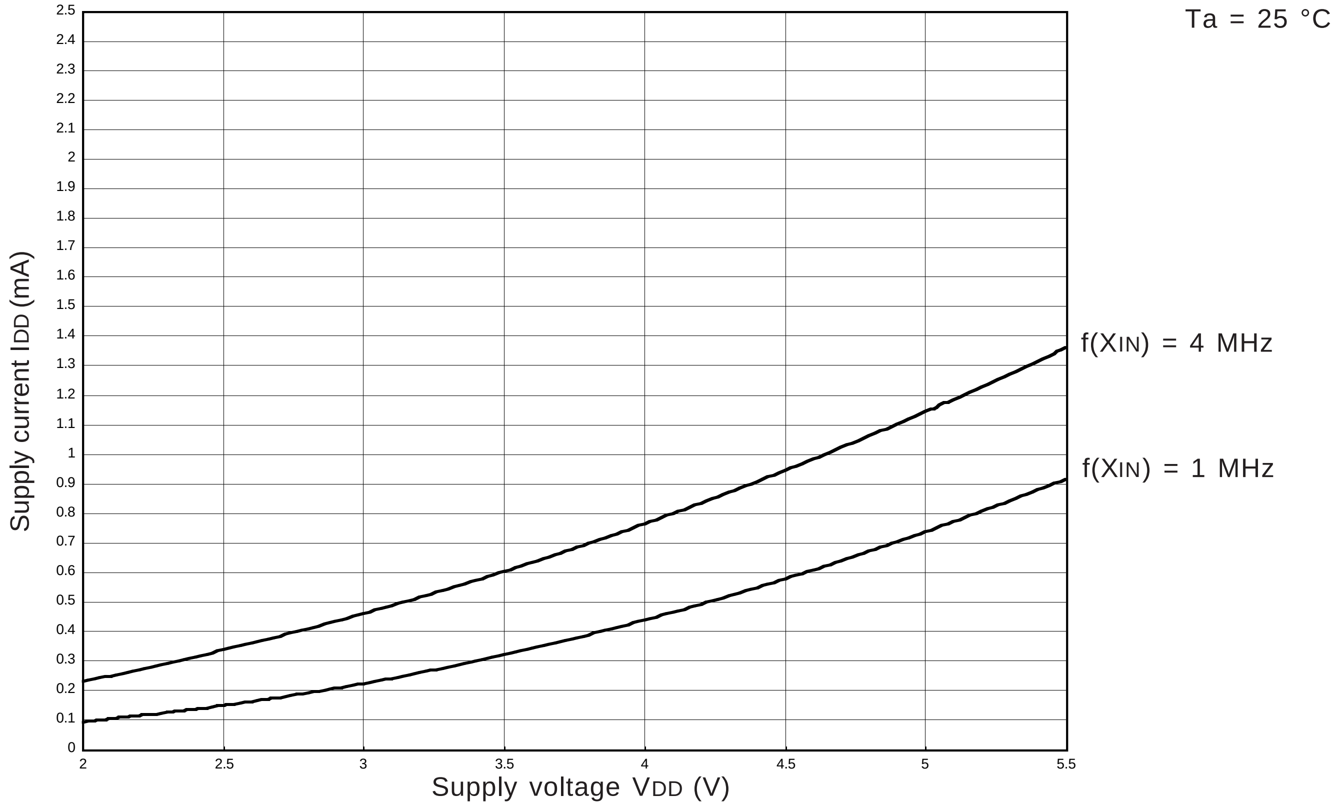
3.1.7 Basic timing diagram



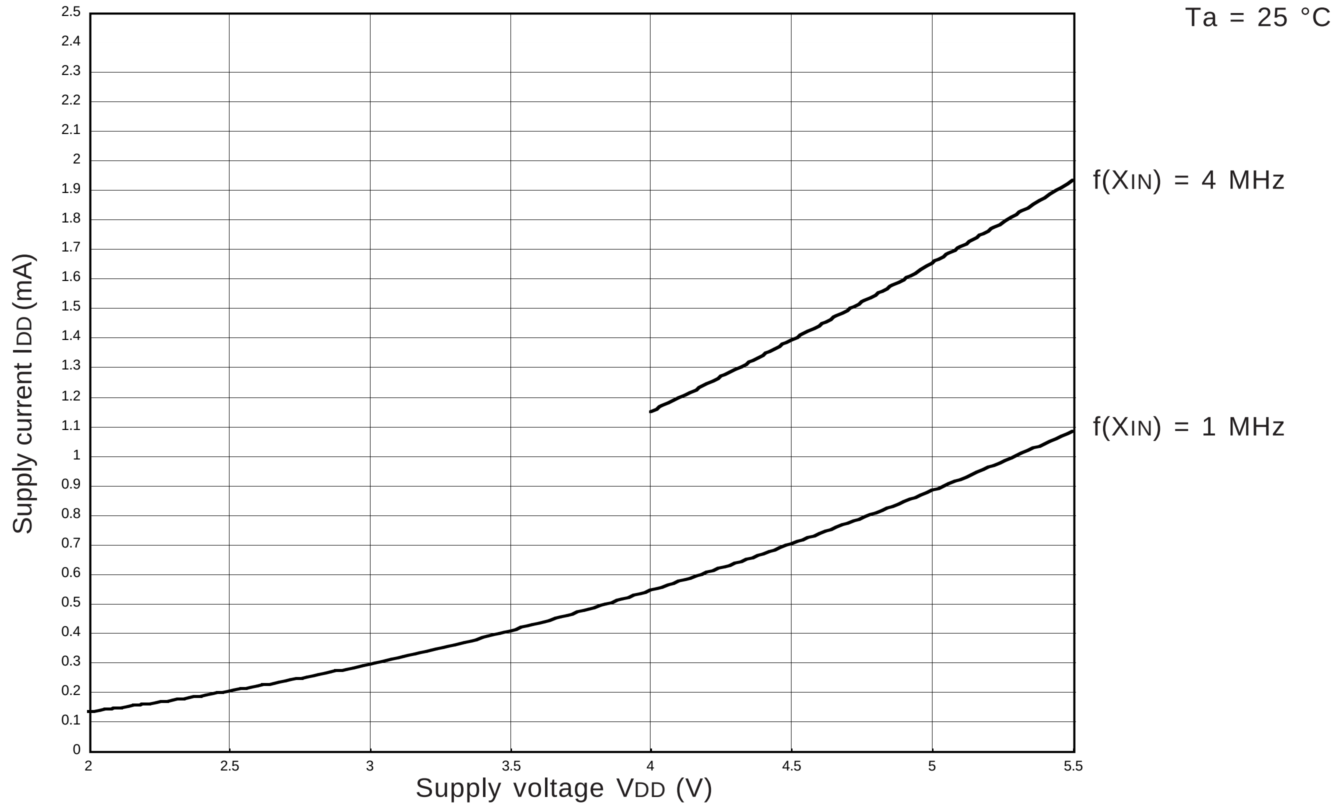
3.2 Typical characteristics

3.2.1 VDD–IDD characteristics

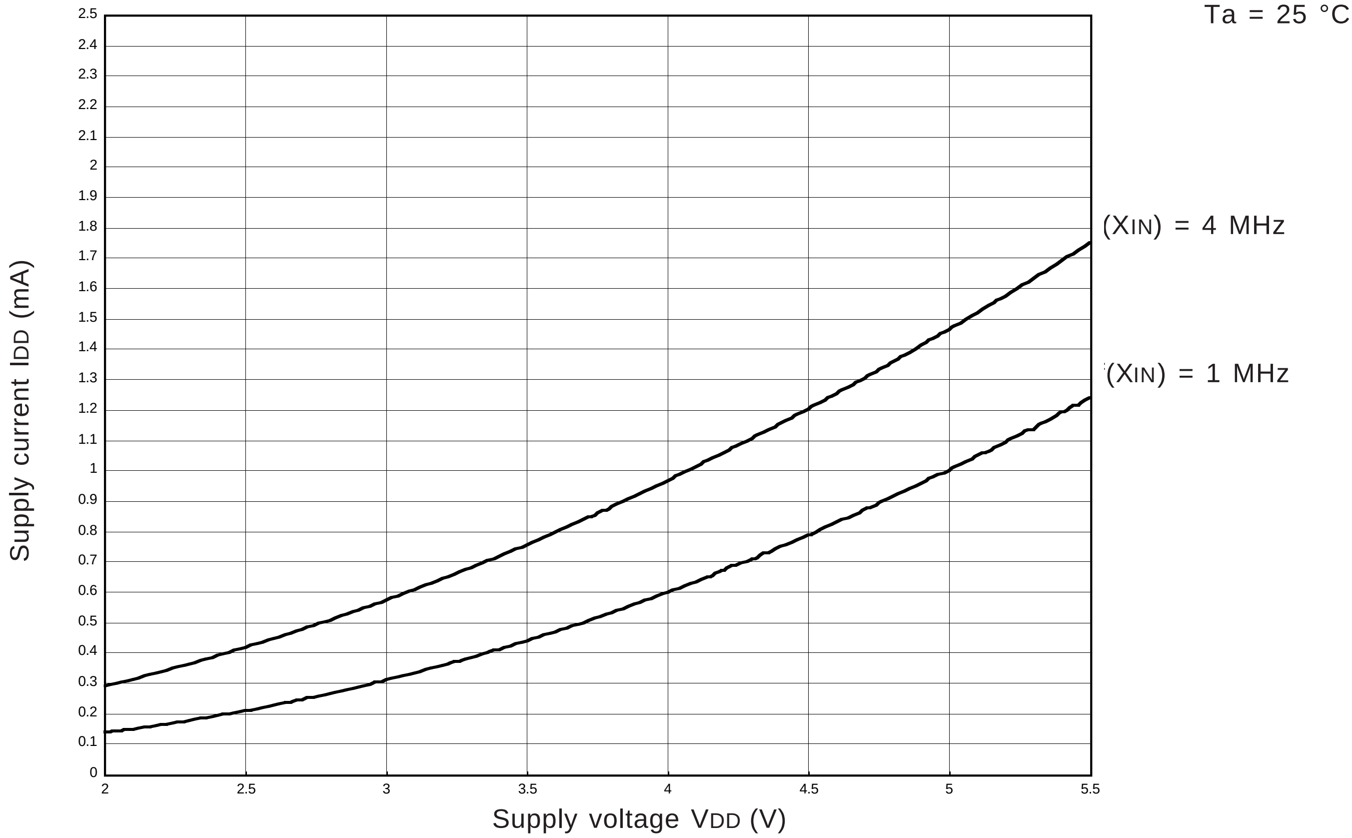
(1) CPU operating, middle-speed mode



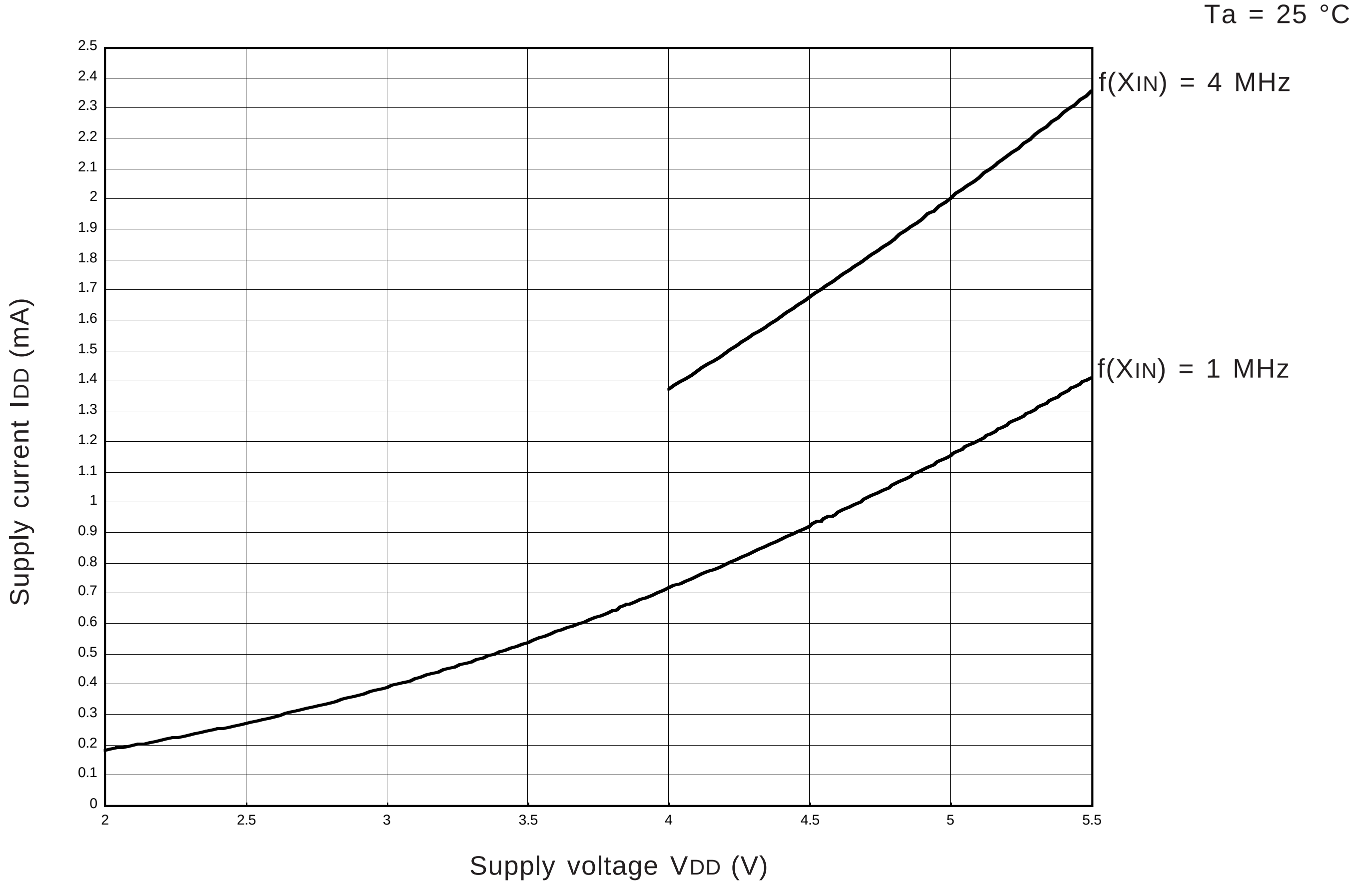
(2) CPU operating, high-speed mode



(3) A-D operating, middle-speed mode

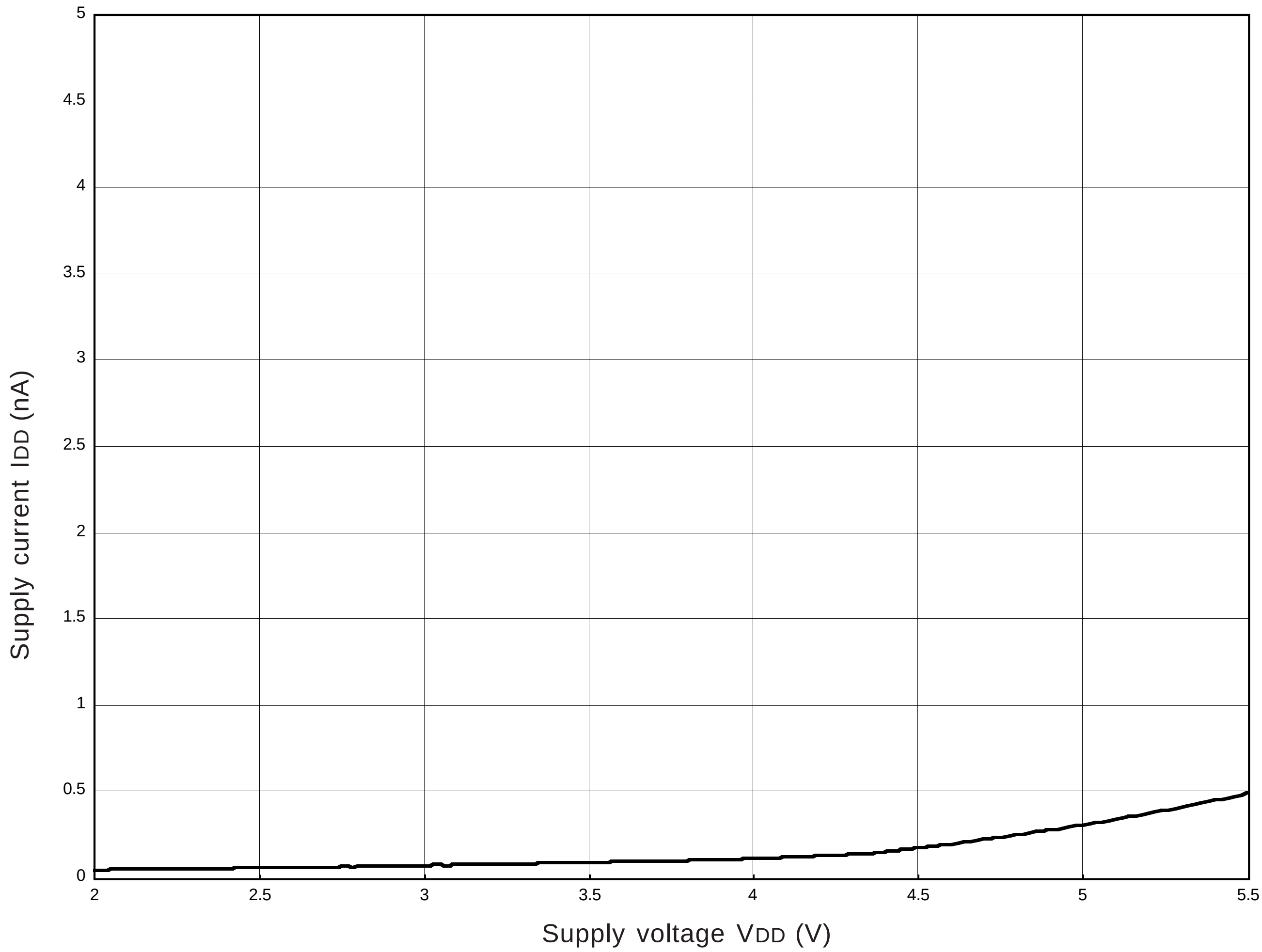


(4) A-D operating, high-speed mode



(5) RAM back-up

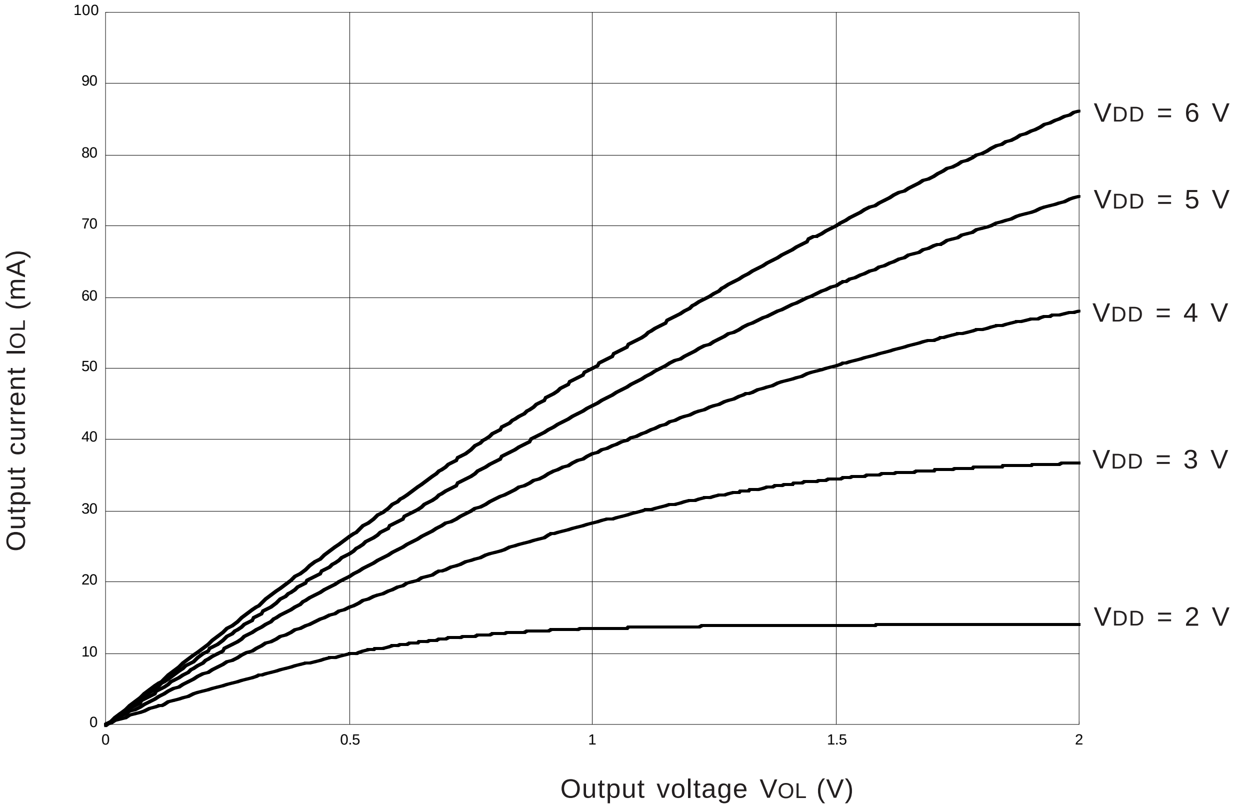
Ta = 25 °C



3.2.2 VOL–IOL characteristics

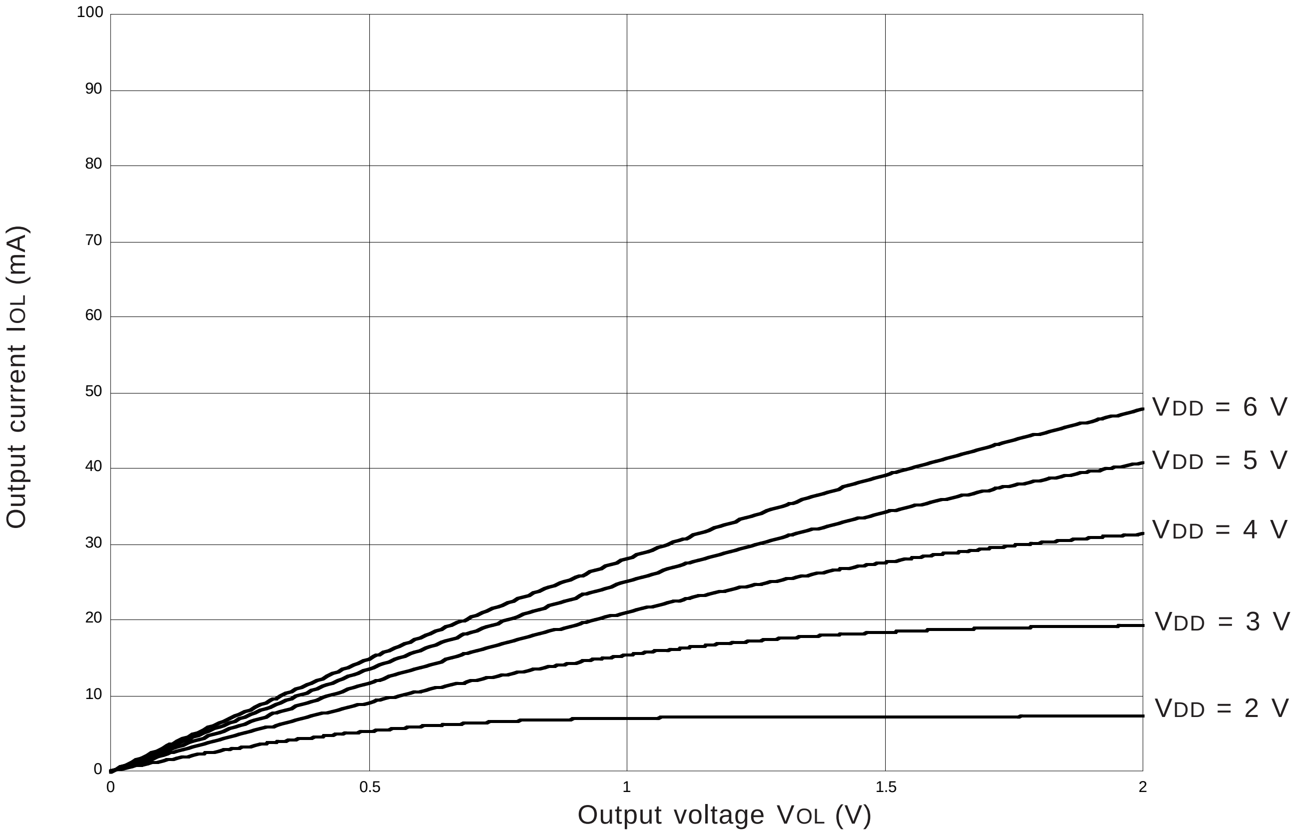
(1) Ports P0, P1, P4, P5, SCK, SOUT

Ta = 25 °C

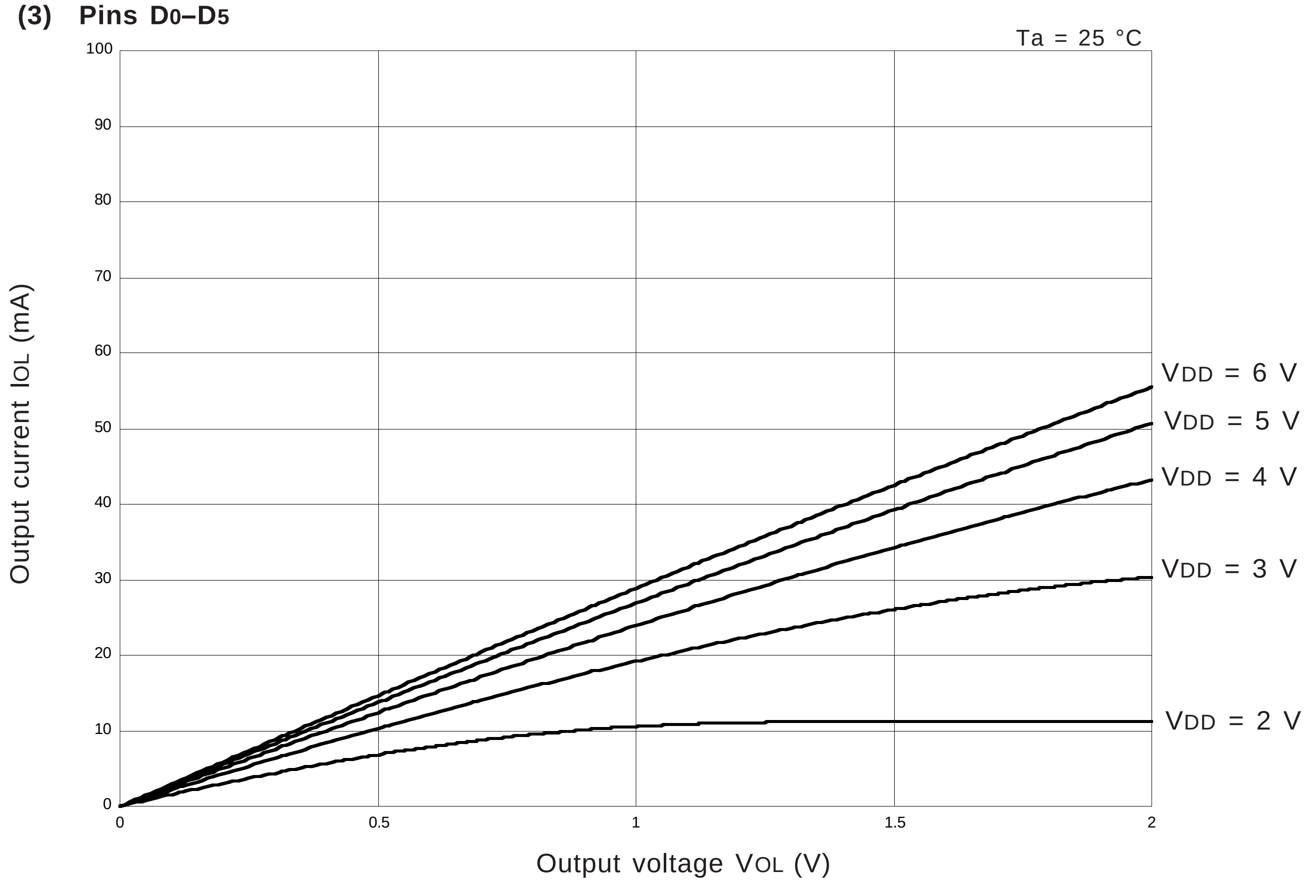


(2) Port P3, $\overline{\text{RESET}}$ pin

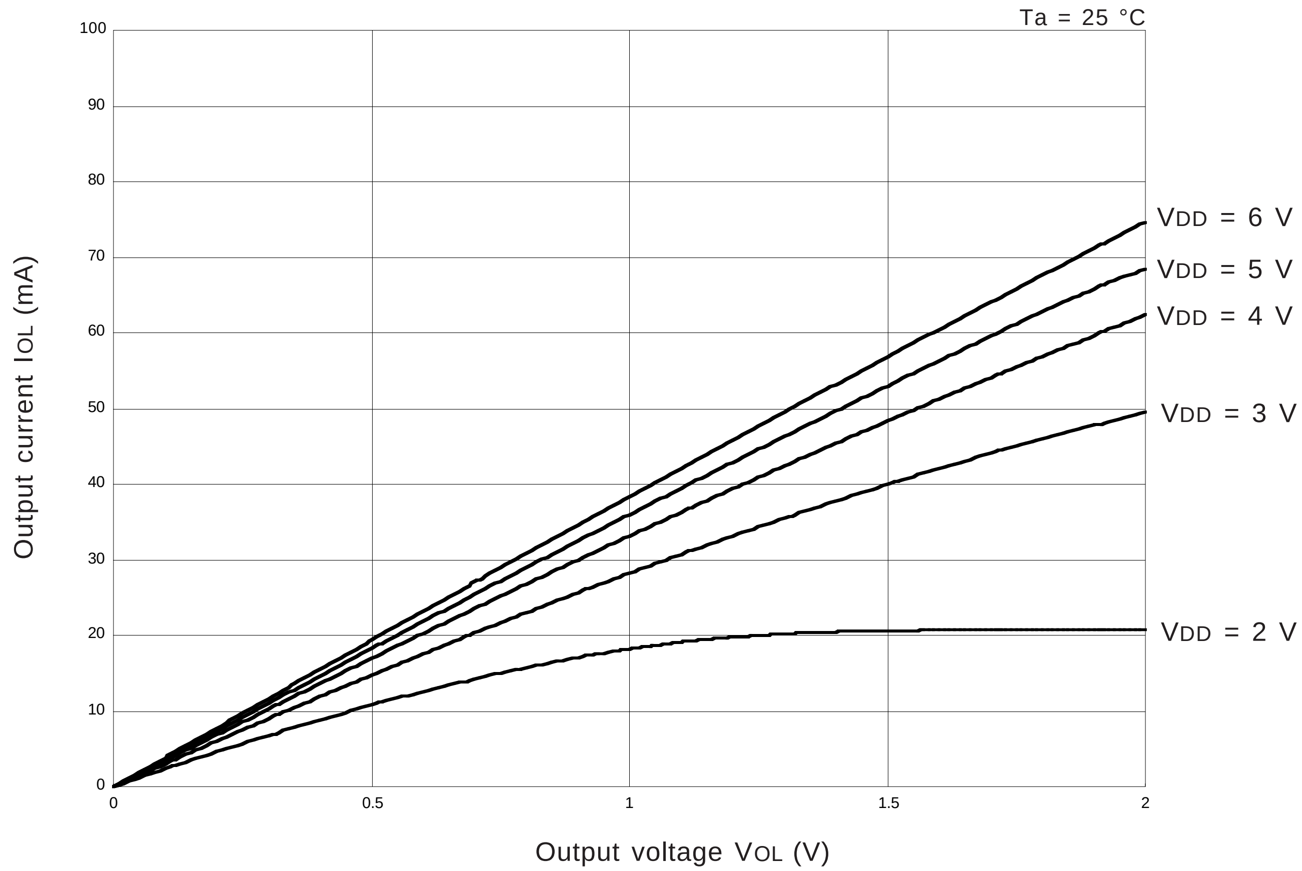
Ta = 25 °C



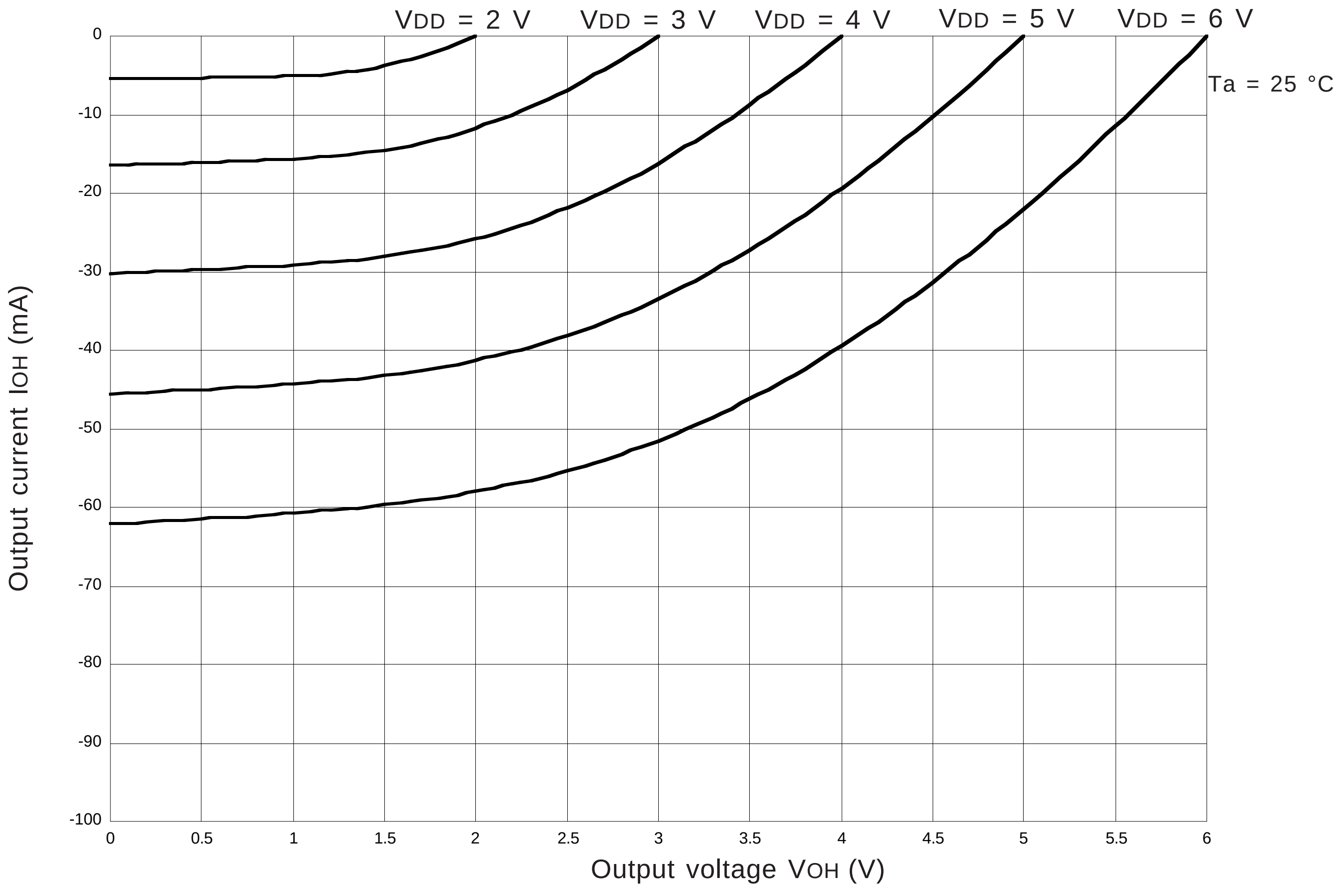
(3) Pins D0–D5



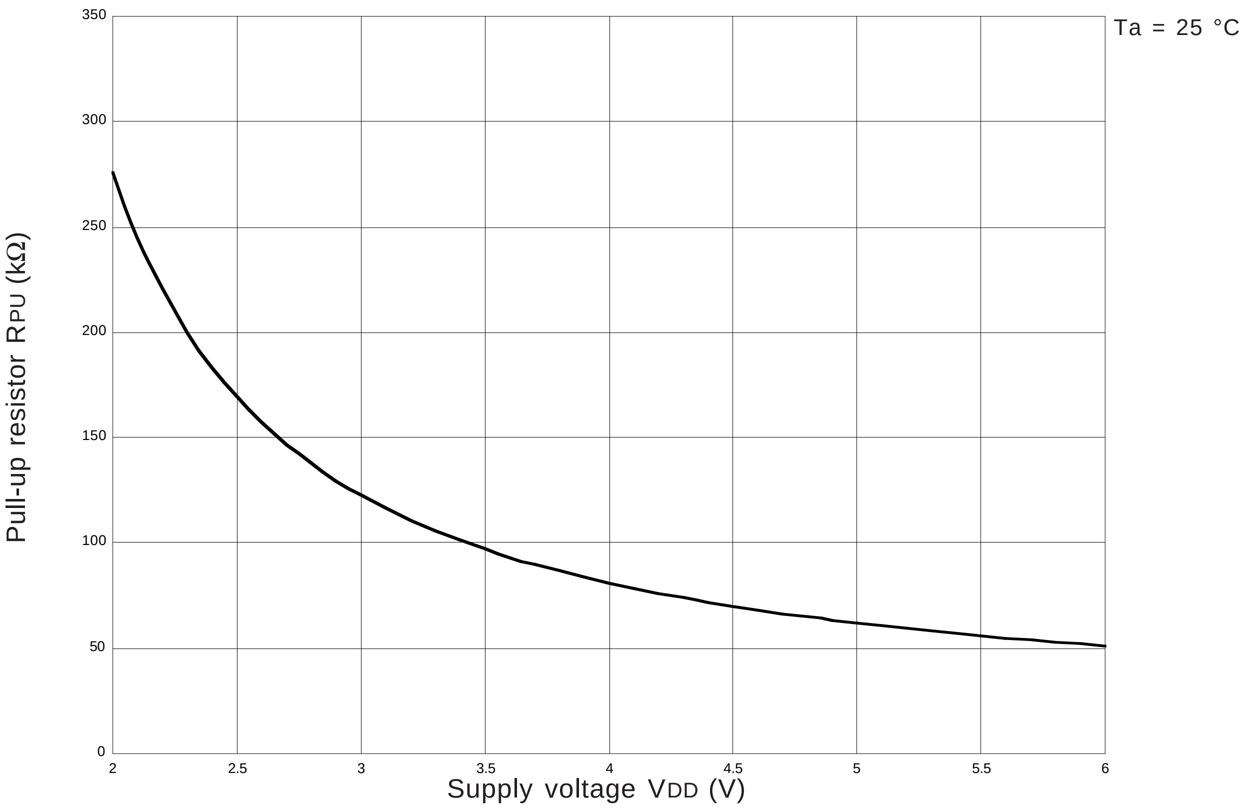
(4) Pins D6/CNTR0, D7/CNTR1



3.2.3 VOH–IOH characteristics (Port P5)



3.2.4 VDD–RPU characteristics (Ports P0, P1)



3.2.5 A-D converter typical characteristics

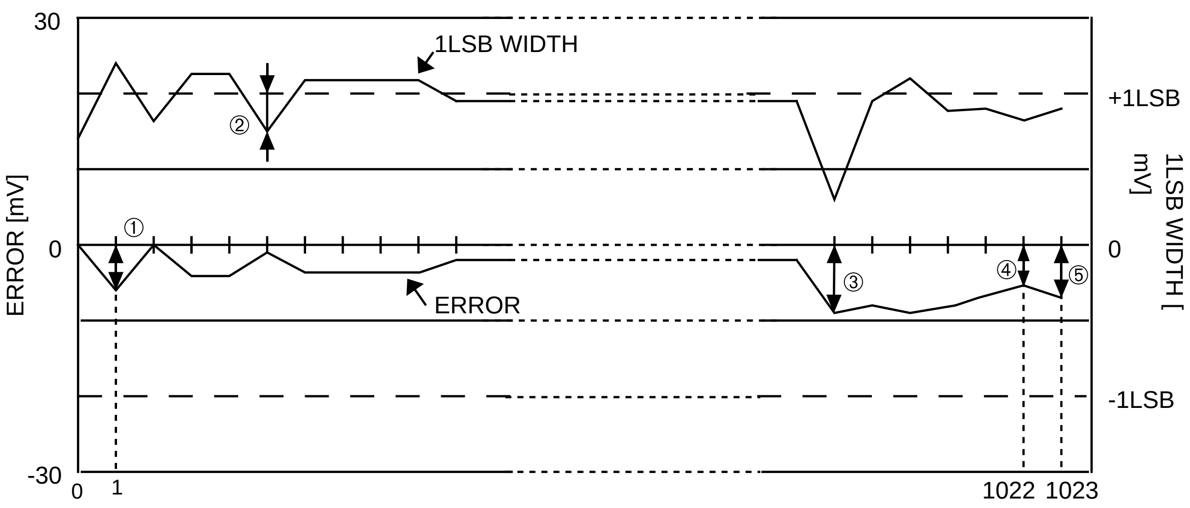


Fig. 3.2.1 A-D conversion characteristics data

Figure 3.2.1 shows the A-D accuracy measurement data.

(1) Non-linearity error .... This means a deviation from the ideal characteristics between V_0 to V_{1022} of actual A-D conversion characteristics. In Figure 3.2.1, it is (④–①)/1LSB.

(2) Differencial non-linearity error .... This means a deviation from the ideal characteristics between the input voltages V_0 to V_{1022} necessary to change the output data to "1." In Figure 3.2.1, this is ②/1LSB.

(3) Zero transition error .... This means a deviation from the ideal characteristics between the input voltages 0 to V_{DD} when the output data changes from "0" to "1." In Figure 3.2.1, this is the value of ①.

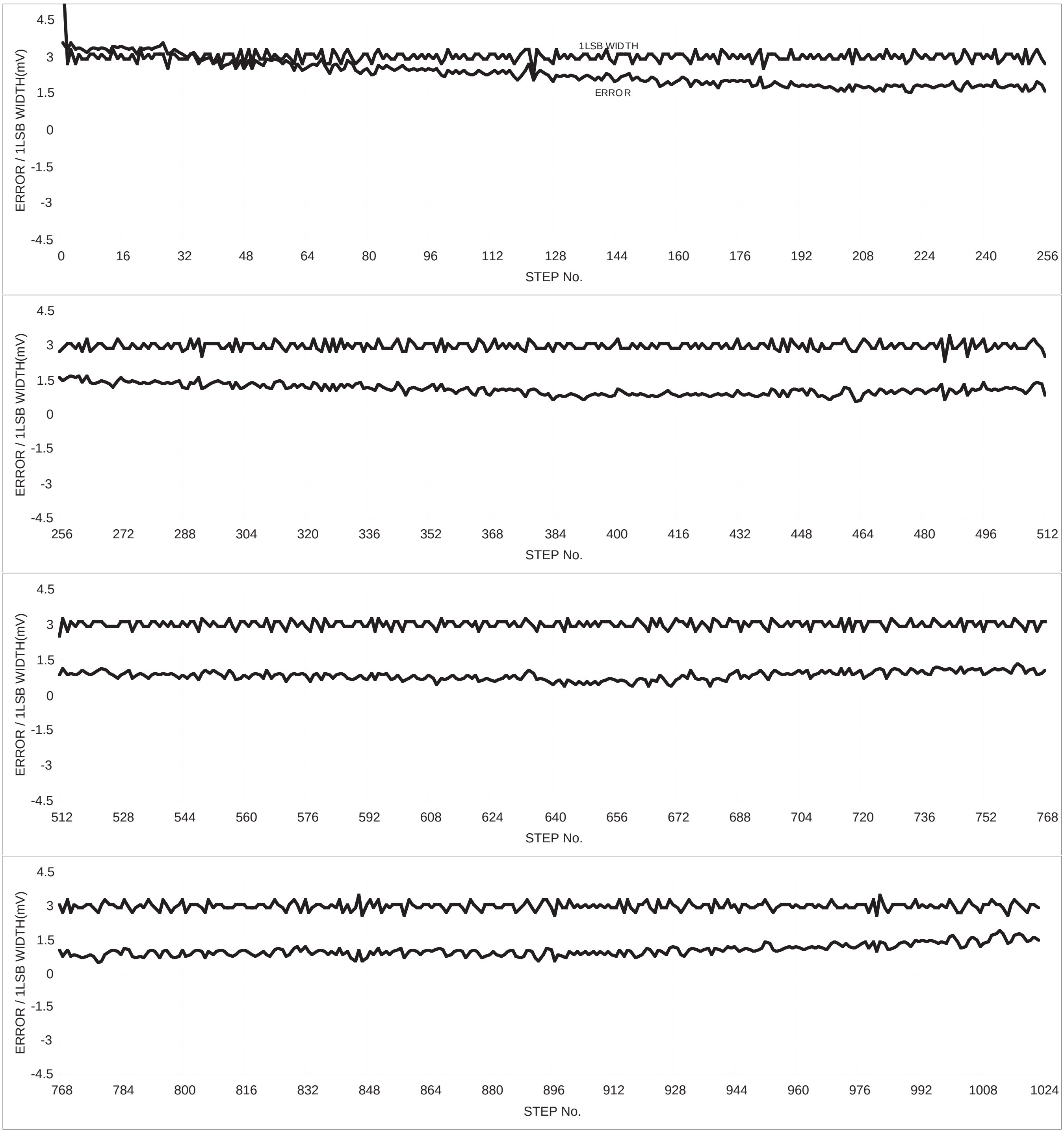
(4) Full-scale transition error .... This means a deviation from the ideal characteristics between the input voltages 0 to V_{DD} when the output data changes from "1022" to "1023." In Figure 3.2.1, this is the value of ⑤.

(5) Absolute accuracy .... This menas a deviation from the ideal characteristics between 0 to V_{DD} of actual A-D conversion characteristics. In Figure 3.2.1, this is the value of ERROR in each of ①, ③, ④ and ⑤.

For the A-D converter characteristics, refer to the section 3.1 Electrical characteristics.

(1) VDD = 3.072 V, f(XIN) = 2 MHz, high-speed mode

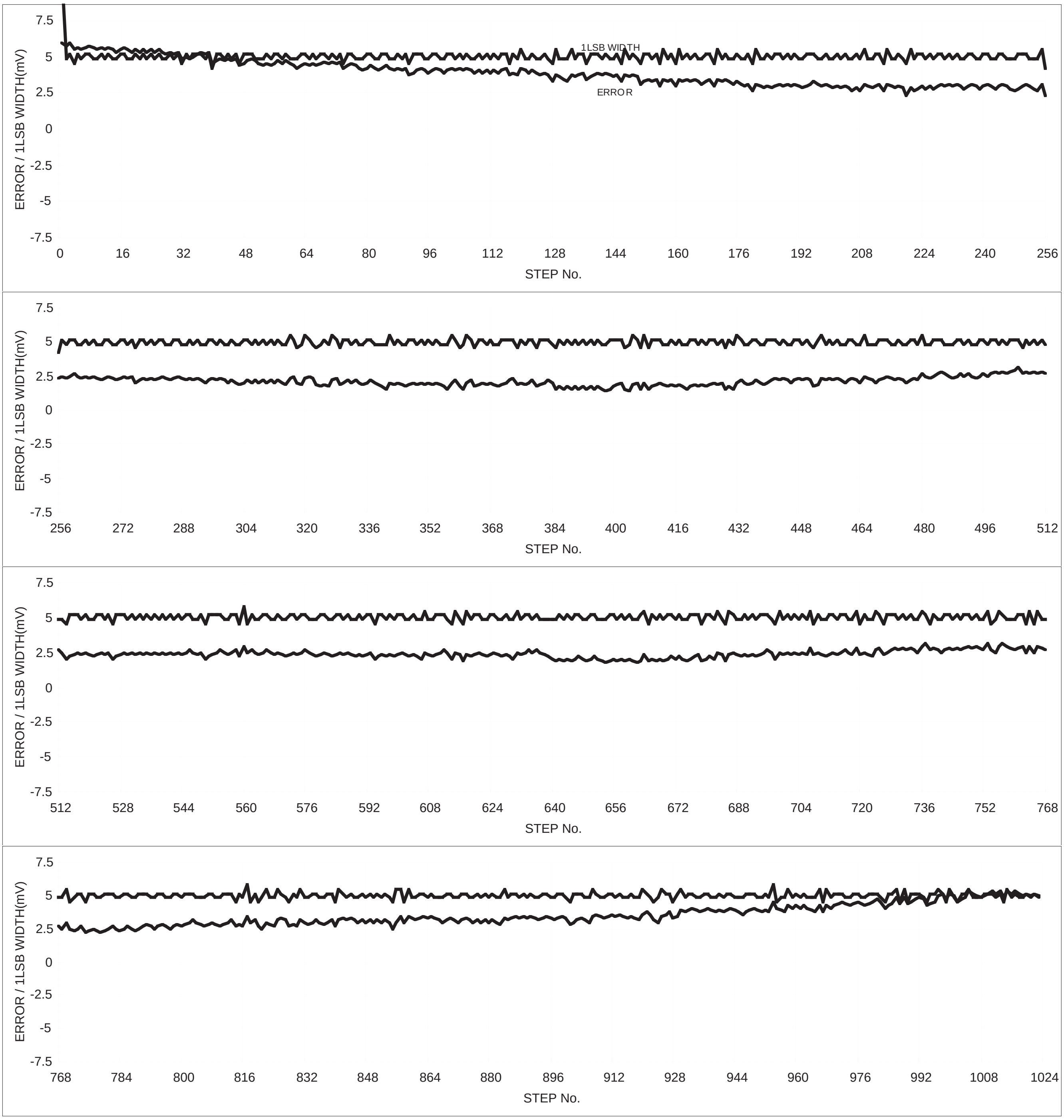
Ta = 25 °C



3.2 Typical characteristics

(2) V_{DD} = 5.12 V, $f(X_{IN})$ = 4 MHz, high-speed mode

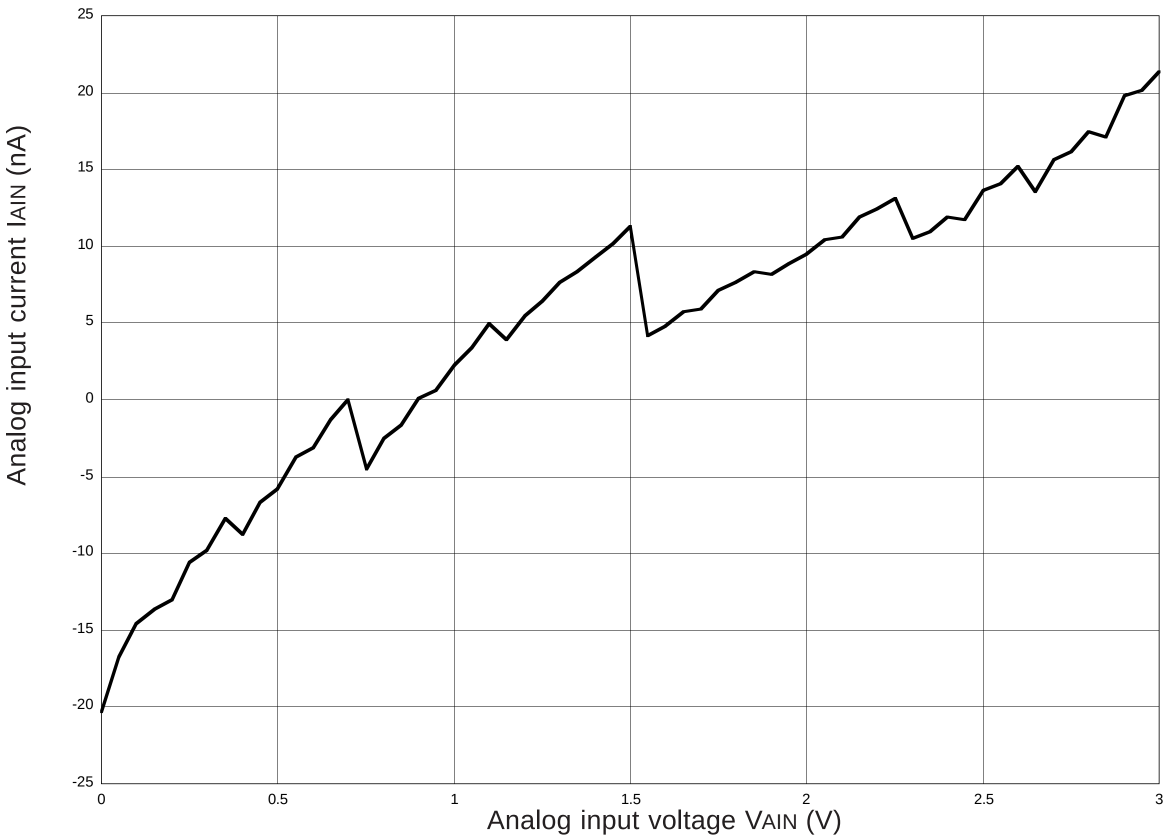
Ta = 25 °C



3.2.6 Analog input current characteristics pins AIN0–AIN7

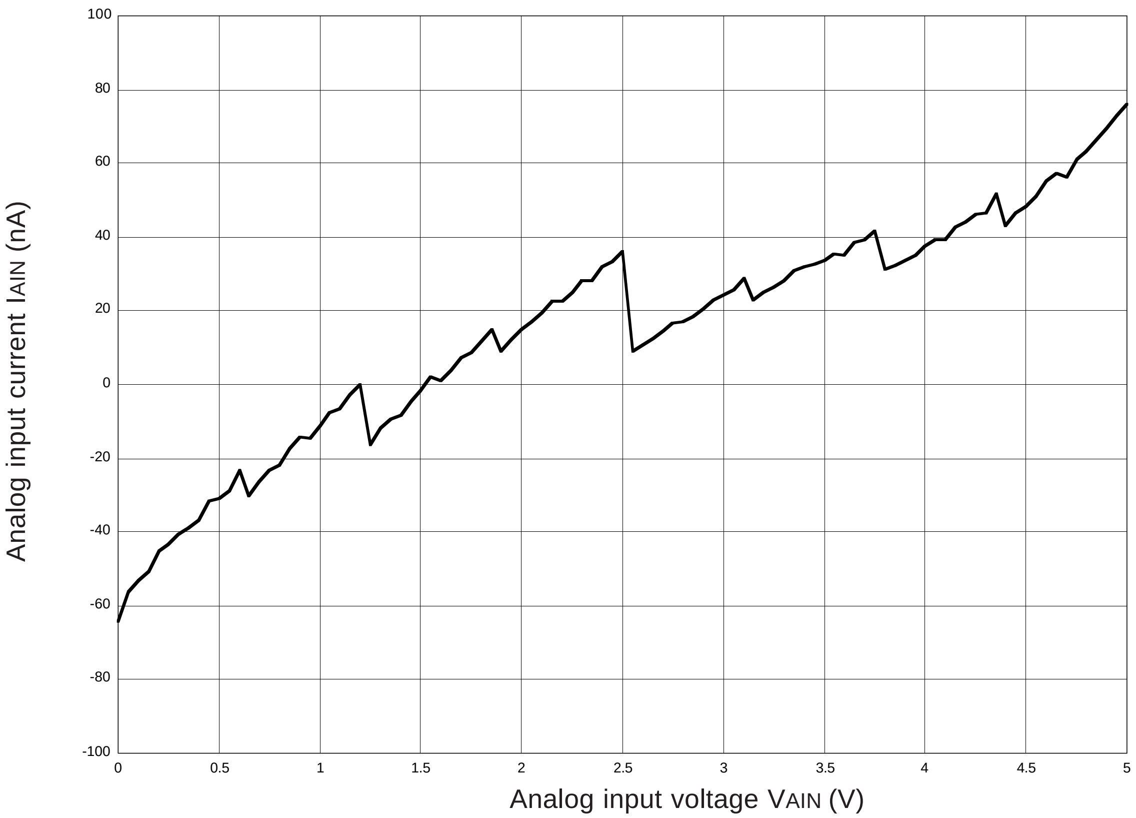
(1) VDD = 3.0 V, f(XIN) = 2 MHz, middle-speed mode

Ta = 25 °C



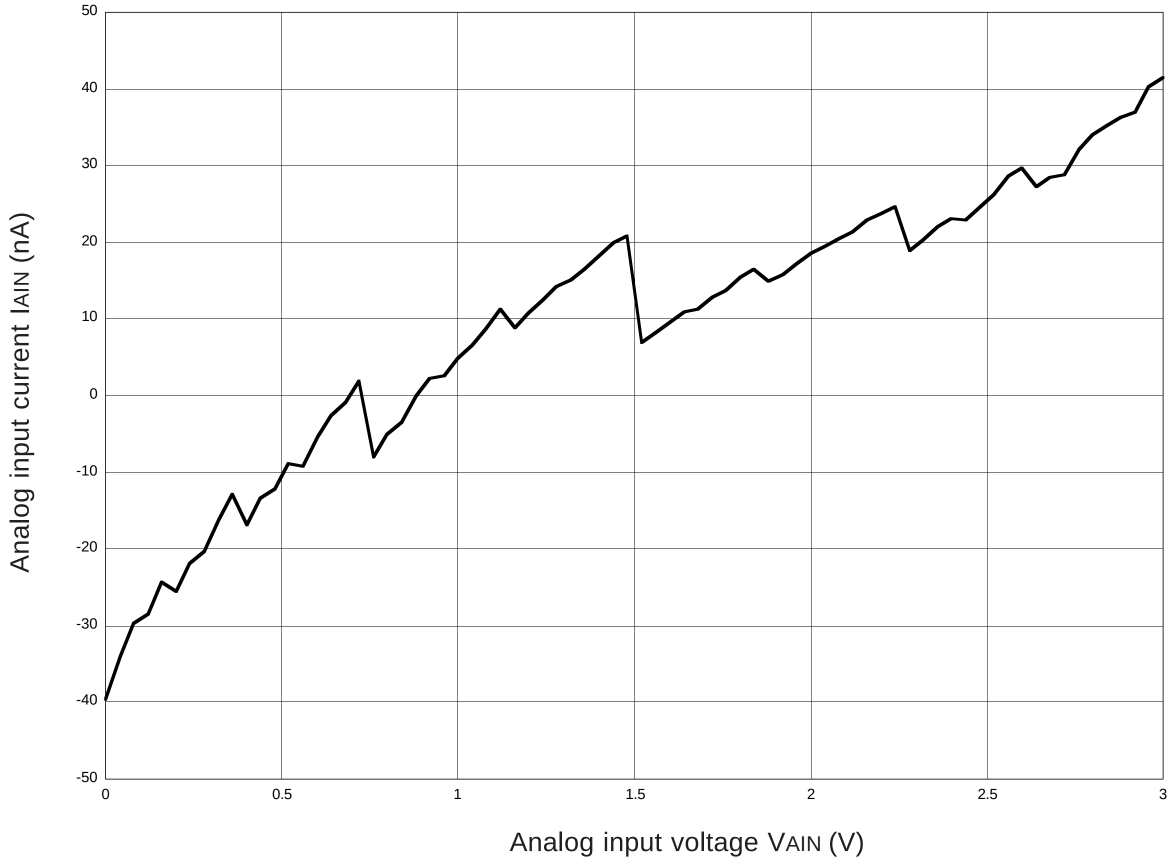
(2) VDD = 3.0 V, f(XIN) = 4 MHz, middle-speed mode

Ta = 25 °C



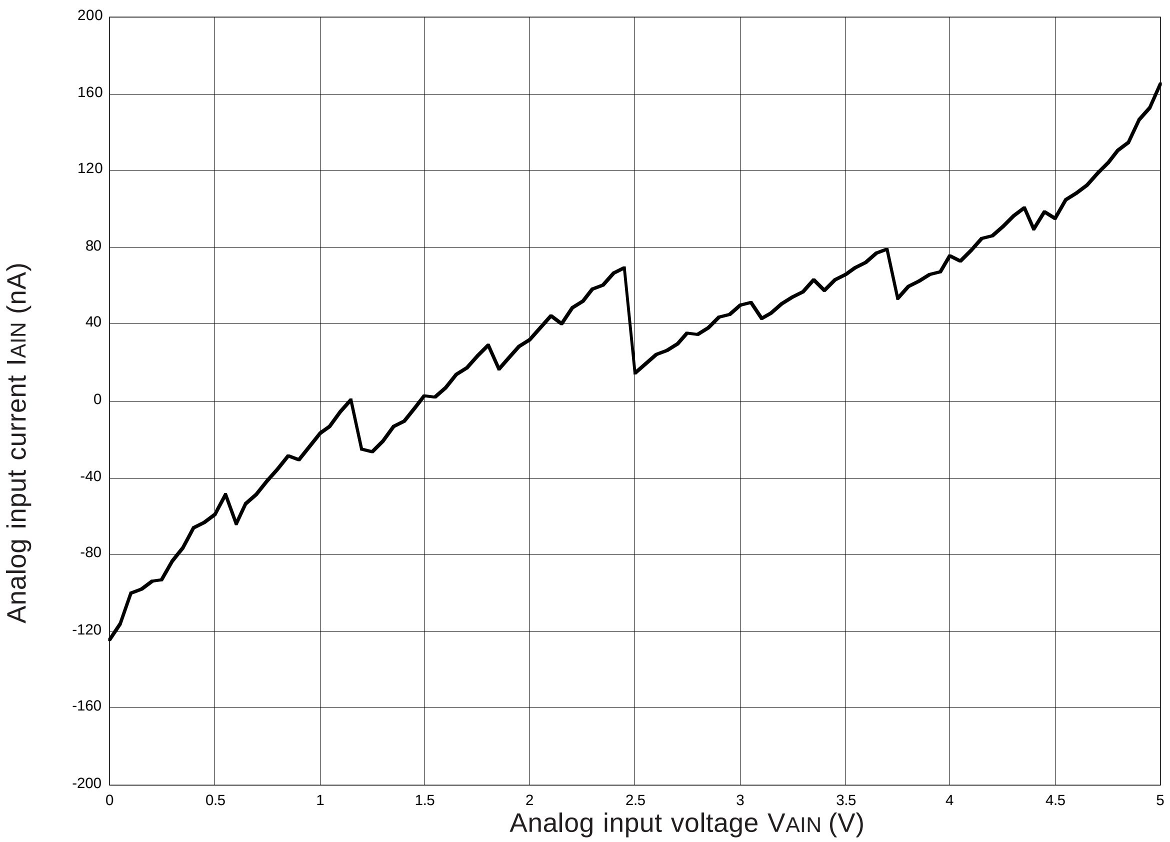
(3) VDD = 3.0 V, f(XIN) = 2 MHz, high-speed mode

Ta = 25 °C



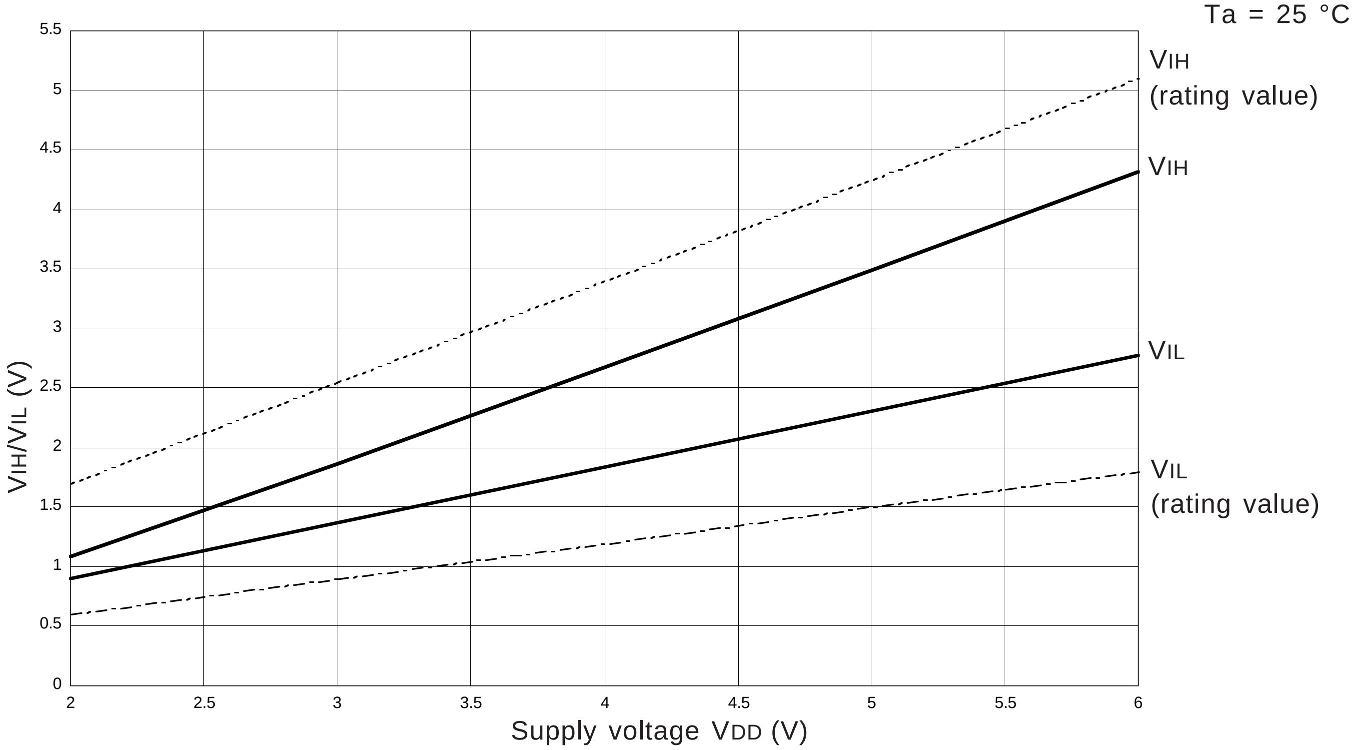
(4) VDD = 5.0 V, f(XIN) = 4 MHz, high-speed mode

Ta = 25 °C

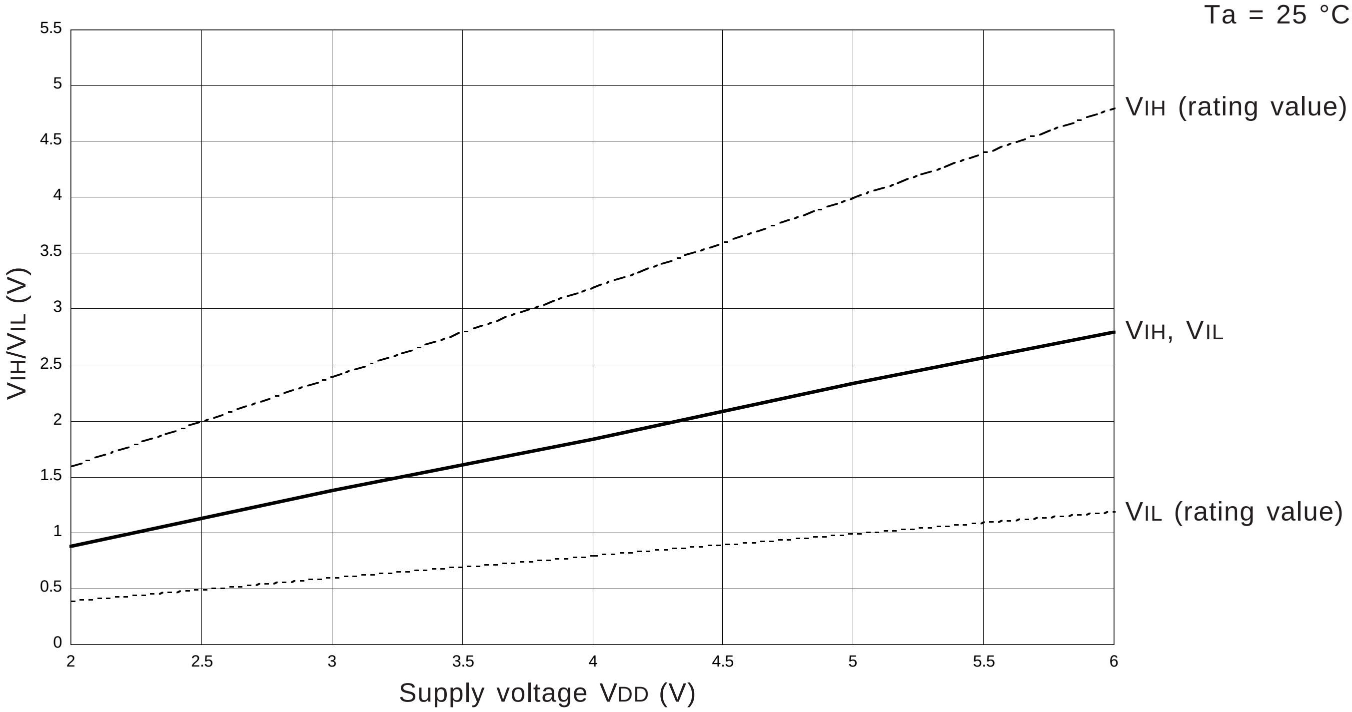


3.2.7 V_{DD}–V_{IH}/V_{IL} characteristics

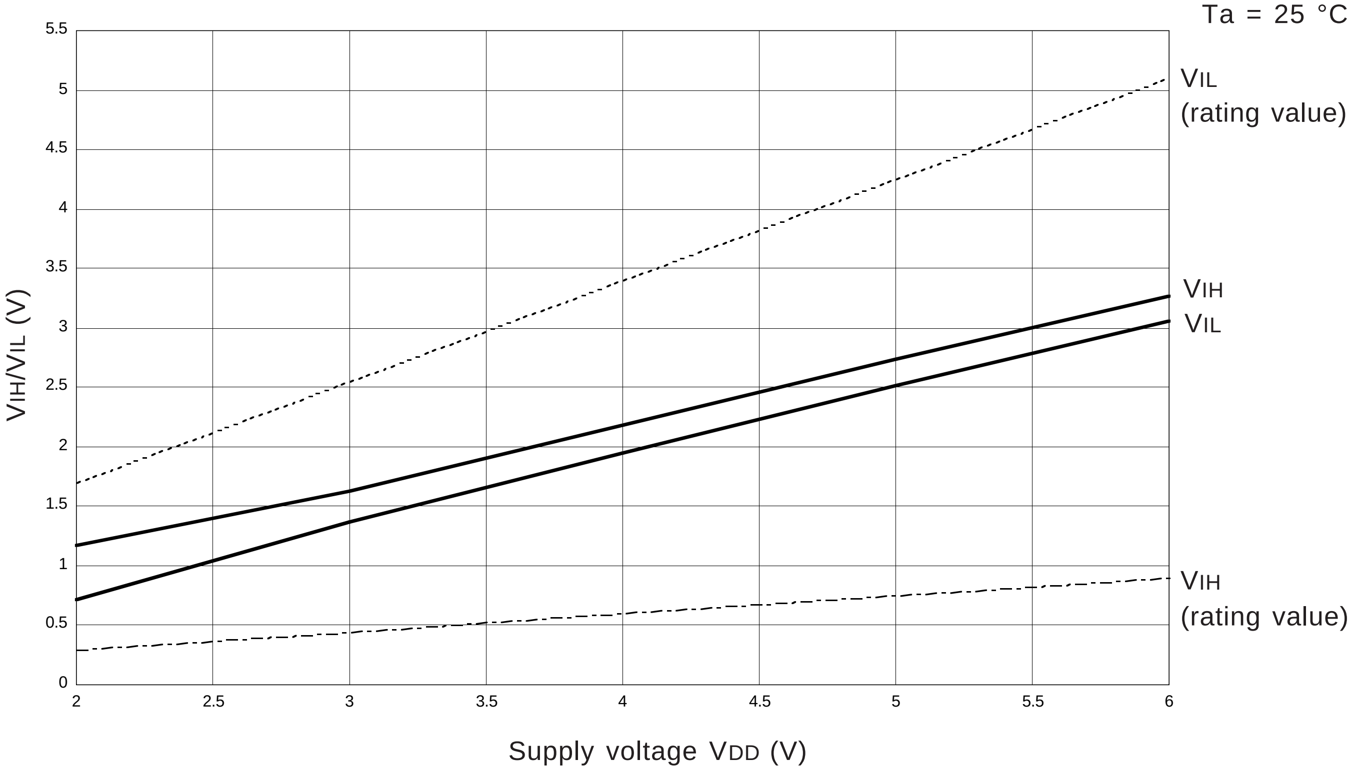
(1) $\overline{\text{RESET}}$ pin



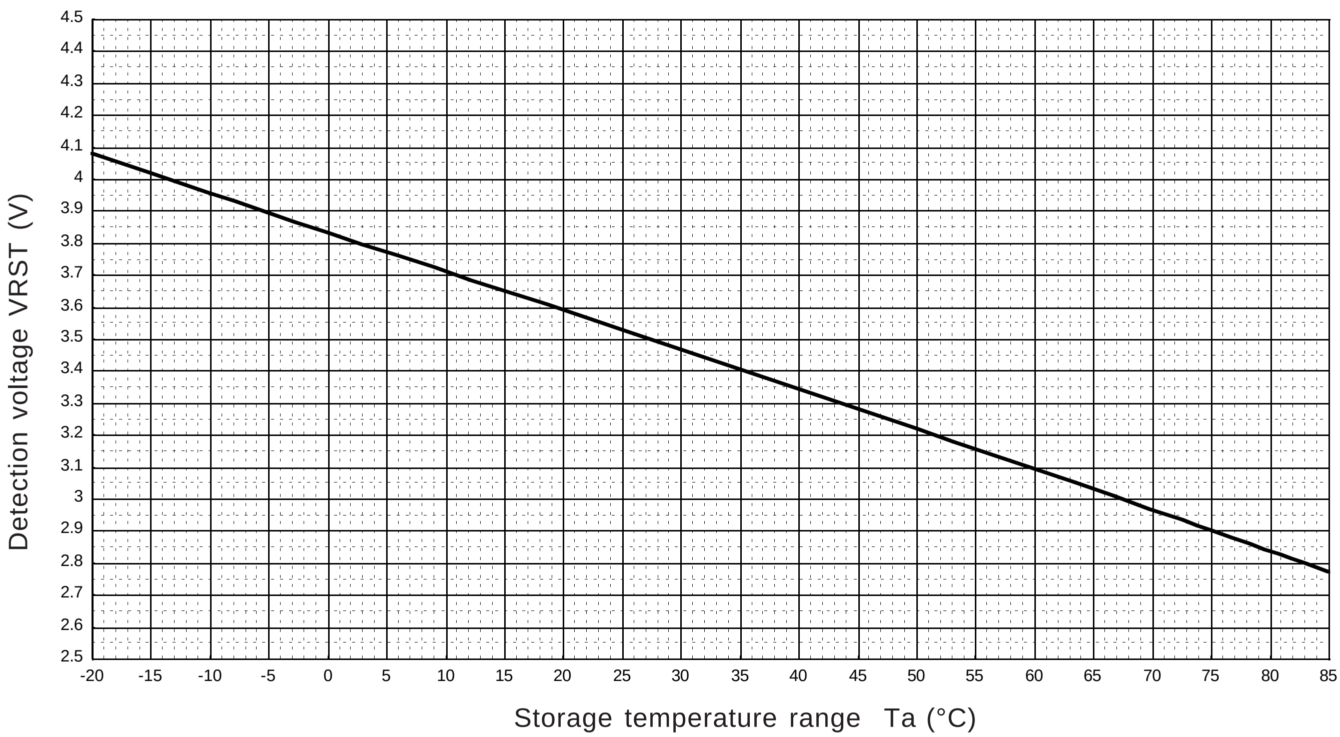
(2) Ports P0, P1, P2, P3, P4, P5, D, X_{IN} pin, VDCE pin



(3) Pins INT0, INT1, CNTR0, CNTR1, SCK, SIN



3.2.8 Detection voltage temperature characteristics of voltage drop detection circuit



3.3 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 VDD and VSS at the shortest distance,
- equalize its wiring in width and length, and
- use relatively thick wire.

In the One Time PROM version, CNVSS pin is also used as VPP pin. Accordingly, when using this pin, connect this pin to VSS through a resistor about 5 kΩ in series at the shortest distance.

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.

⑥P30/INT0 pin

When the interrupt valid waveform of the P30/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 P30/INT0 pin is changed with the bit 2 of register I1 (refer to Figure 44①).
- Depending on the input state of the P30/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 44②)

```
  ⋮
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. 44 External 0 interrupt program example

⑦P31/INT1 pin

When the interrupt valid waveform of P31/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 P31/INT1 pin is changed with the bit 2 of register I2 (refer to Figure 45③).
- Depending on the input state of the P31/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 45④).

```
  ⋮
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. 45 External 1 interrupt program example

⑧One Time PROM version

The operating power voltage of the One Time PROM version is 2.5 V to 5.5 V.

⑨Multifunction

The input of D6, D7, P20–P22, I/O of P30 and P31, input of CMP0-, CMP0+, CMP1-, CMP1+, and I/O of P40–P43 can be used even when CNTR0, CNTR1, SCK, SOUT, SIN, INT0, INT1, AIN0–AIN3 and AIN4–AIN7 are selected.

⑩ 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 46⑤).
- 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          ; (✕0✕✕2)
TV2A               ; The SNZAD instruction is valid ........⑤
LA      0          ; (0✕✕✕2)
TQ2A               ; Change of the operating mode of the A-D
                     converter from the comparator mode to the
                     A-D conversion mode
SNZAD
NOP
        ⋮
                  ✕: this bit is not related to the change of the
                     operating mode of the A-D conversion.
```

Fig. 46 A-D converter operating mode program example

⑪ 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 47).

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 48. In addition, test the application products sufficiently.

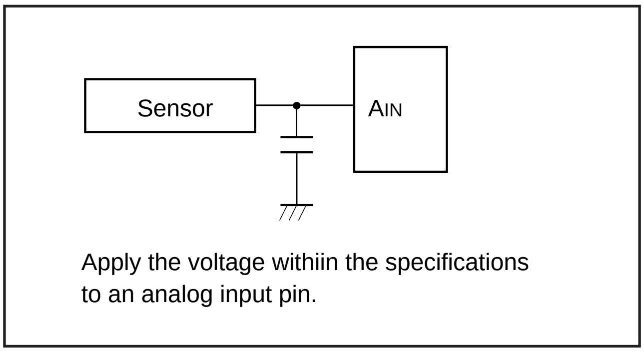


Fig. 47 Analog input external circuit example-1

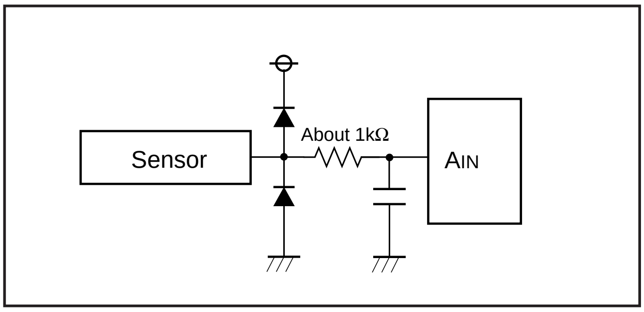


Fig. 48 Analog input external circuit example-2

⑫ 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.

⑬ Analog input pins

Note the following when using the analog input pins also for I/O port P4 functions:

- Even when $P4_0$/A_{IN4}–$P4_3$/A_{IN7} are set to pins for analog input, they continue to function as $P4_0$–$P4_3$ I/O. Accordingly, when any of them are used as I/O port P4 and others are used as analog input pins, make sure to set the outputs of pins that are set for analog input to "1." Also, the port input function of the pin functions as an analog input is undefined.
- TALA instruction
  When the TALA instruction is executed, the low-order 2 bits of register AD is transferred to the high-order 2 bits of register A, simultaneously, the low-order 2 bits of register A is "0."

⑭ Program counter

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

⑮ Port P3

In the 4513 Group, when the IAP3 instruction is executed, note that the high-order 2 bits of register A is undefined.

⑯ Voltage comparator function

When the voltage comparator function is valid with the voltage comparator control register Q3, it is operating even in the RAM back-up mode. Accordingly, be careful about such state because it causes the increase of the operation current in the RAM back-up mode.
In order to reduce the operation current in the RAM back-up mode, invalidate (bits 2 and 3 of register Q3 = “0”) the voltage comparator function by software before the POF instruction is executed.
Also, while the voltage comparator function is valid, current is always consumed by voltage comparator. On the system required for the low-power dissipation, invalidate the voltage comparator when it is unused by software.

⑰ Register Q3

Bits 0 and 1 of register Q3 can be only read. Note that they cannot be written.

⑱ Reading the comparison result of voltage comparator

Read the voltage comparator comparison result from register Q3 after the voltage comparator response time (max. 20 μs) is passed from the voltage comparator function become valid.

3.4 Notes on noise

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

3.4.1 Shortest wiring length

The wiring on a printed circuit board can function as an antenna which feeds noise into the microcomputer.

The shorter the total wiring length (by mm unit), the less the possibility of noise insertion into a microcomputer.

(1) Package

Select the smallest possible package to make the total wiring length short.

● **Reason**

The wiring length depends on a microcomputer package. Use of a small package, for example QFP and not DIP, makes the total wiring length short to reduce influence of noise.

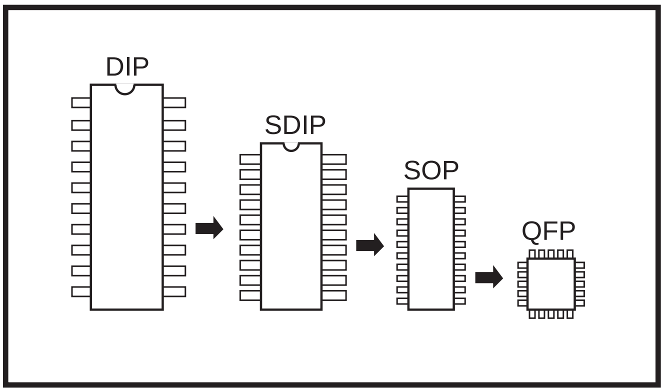


Fig. 3.4.1 Selection of packages

(2) Wiring for $\overline{\text{RESET}}$ input pin

Make the length of wiring which is connected to the $\overline{\text{RESET}}$ input pin as short as possible. Especially, connect a capacitor across the $\overline{\text{RESET}}$ input pin and the Vss pin with the shortest possible wiring.

● **Reason**

In order to reset a microcomputer correctly, 1 machine cycle or more of the width of a pulse input into the $\overline{\text{RESET}}$ pin is required. If noise having a shorter pulse width than this is input to the $\overline{\text{RESET}}$ input pin, the reset is released before the internal state of the microcomputer is completely initialized. This may cause a program runaway.

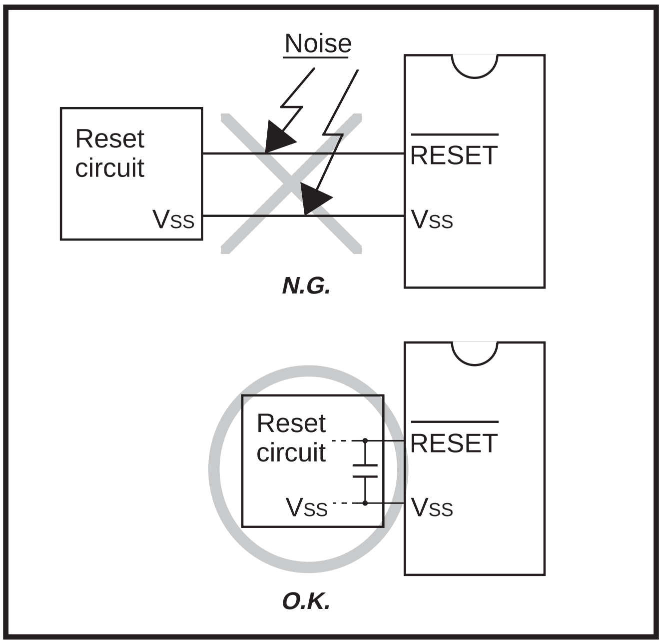


Fig. 3.4.2 Wiring for the $\overline{\text{RESET}}$ input pin

(3) Wiring for clock input/output pins

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

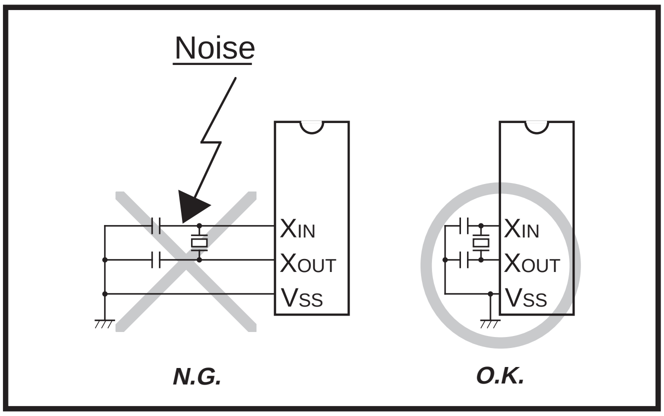


Fig. 3.4.3 Wiring for clock I/O pins

● **Reason**

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

(4) Wiring to CNVss pin

Connect the CNVss pin to the Vss pin with the shortest possible wiring.

● **Reason**

The operation mode of a microcomputer is influenced by a potential at the CNVss pin. If a potential difference is caused by the noise between pins CNVss and Vss, the operation mode may become unstable. This may cause a microcomputer malfunction or a program runaway.

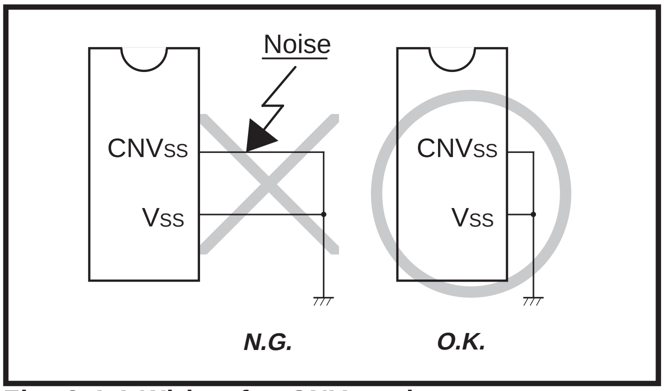


Fig. 3.4.4 Wiring for CNVss pin

(5) Wiring to VPP pin of One Time PROM version
In the built-in PROM version of the 4513/4514 Group, the CNVSS pin is also used as the built-in PROM power supply input pin VPP.

● **When the VPP pin is also used as the CNVss pin**
Connect an approximately 5 kΩ resistor to the VPP pin the shortest possible in series and also to the VSS pin. When not connecting the resistor, make the length of wiring between the VPP pin and the VSS pin the shortest possible (refer to **Figure 3.4.5**)

Note: Even when a circuit which included an approximately 5 kΩ resistor is used in the Mask ROM version, the microcomputer operates correctly.

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

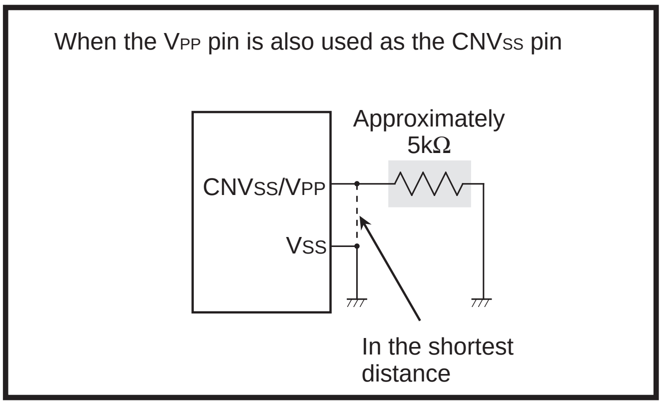


Fig. 3.4.5 Wiring for the VPP pin of the One Time PROM version

3.4.2 Connection of bypass capacitor across VSS line and VDD line

Connect an approximately 0.1 μF bypass capacitor across the VSS line and the VDD line as follows:

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

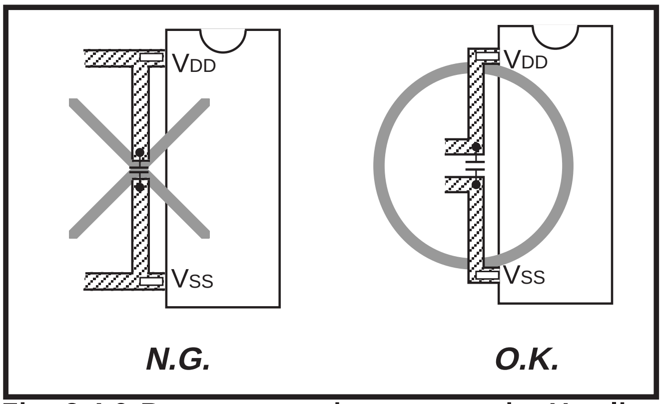


Fig. 3.4.6 Bypass capacitor across the Vss line and the VDD line

3.4.3 Wiring to analog input pins

- Connect an approximately 100 Ω to 1 kΩ resistor to an analog signal line which is connected to an analog input pin in series. Besides, connect the resistor to the microcomputer as close as possible.
- Connect an approximately 1000 pF capacitor across the VSS pin and the analog input pin. Besides, connect the capacitor to the VSS pin as close as possible. Also, connect the capacitor across the analog input pin and the VSS pin at equal length.

● **Reason**

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

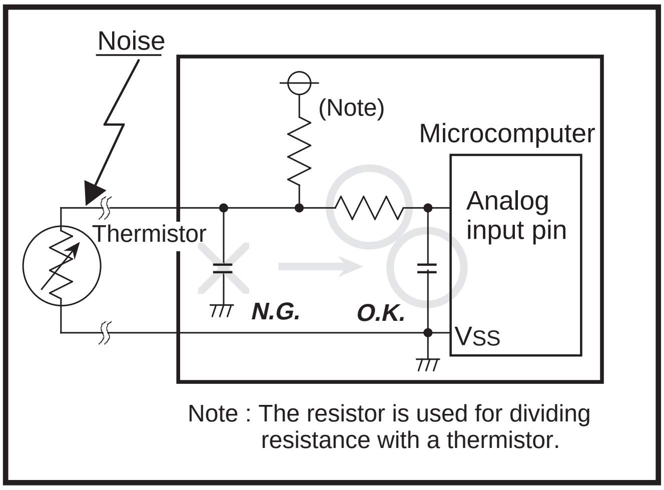


Fig. 3.4.7 Analog signal line and a resistor and a capacitor

3.4.4 Oscillator concerns

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

(1) Keeping oscillator away from large current signal lines

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

● **Reason**

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

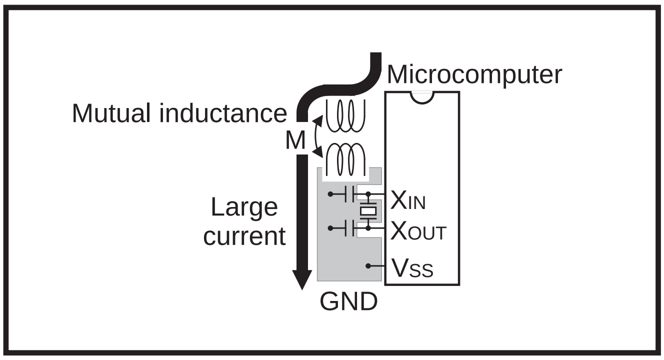


Fig. 3.4.8 Wiring for a large current signal line

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

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

● **Reason**

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

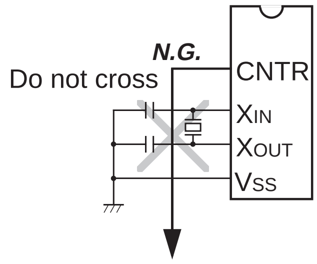


Fig. 3.4.9 Wiring to a signal line where potential levels change frequently

(3) Oscillator protection using Vss pattern

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

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

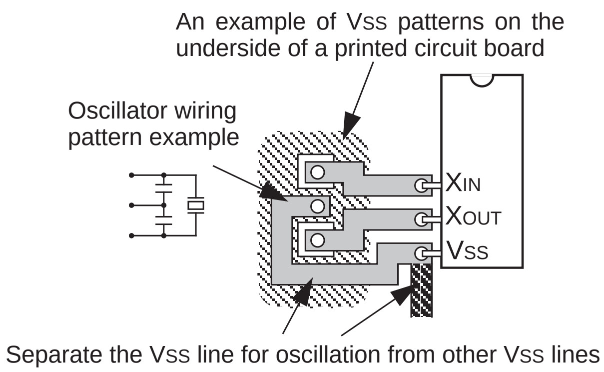


Fig. 3.4.10 Vss pattern on the underside of an oscillator

3.4.5 Setup for I/O ports

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

<Hardware>

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

<Software>

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

3.4.6 Providing of watchdog timer function by software

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

In the following example, to reset a microcomputer to normal operation, the main routine detects errors of the interrupt processing routine and the interrupt processing routine detects errors of the main routine. This example assumes that interrupt processing is repeated multiple times in a single main routine processing.

<The main routine>

• Assigns a single word of RAM to a software watchdog timer (SWDT) and writes the initial value N in the SWDT once at each execution of the main routine. The initial value N should satisfy the following condition:

N+1≥ (Counts of interrupt processing executed in each main routine)

As the main routine execution cycle may change because of an interrupt processing or others, the initial value N should have a margin.

• Watches the operation of the interrupt processing routine by comparing the SWDT contents with counts of interrupt processing after the initial value N has been set.
• Detects that the interrupt processing routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
  If the SWDT contents do not change after interrupt processing.

<The interrupt processing routine>

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

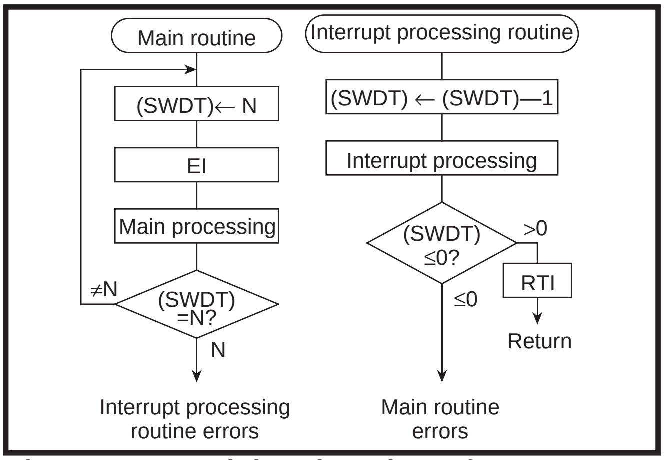


Fig. 3.4.11 Watchdog timer by software

3.5 Mask ROM order confirmation form

GZZ-SH52-45B <81A0>

Mask ROM number	

4500 SERIES MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M34513M2-XXXSP/FP
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.

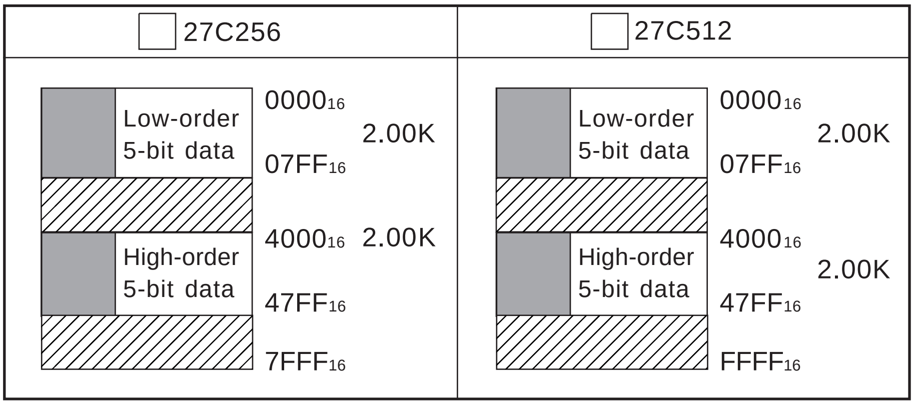
Three sets of EPROMs are required for each pattern (check in the approximate box).

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.

Microcomputer name: ☐M34513M2-XXXSP ☐M34513M2-XXXFP

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 (32P4B for M34513M2-XXXSP, 32P6B-A for M34513M2-XXXFP) and attach to the Mask ROM Order Confirmation Form.

✻ 3. Comments

GZZ-SH52-44B <81A0>

Mask ROM number	

4500 SERIES MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M34513M4-XXXSP/FP
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.

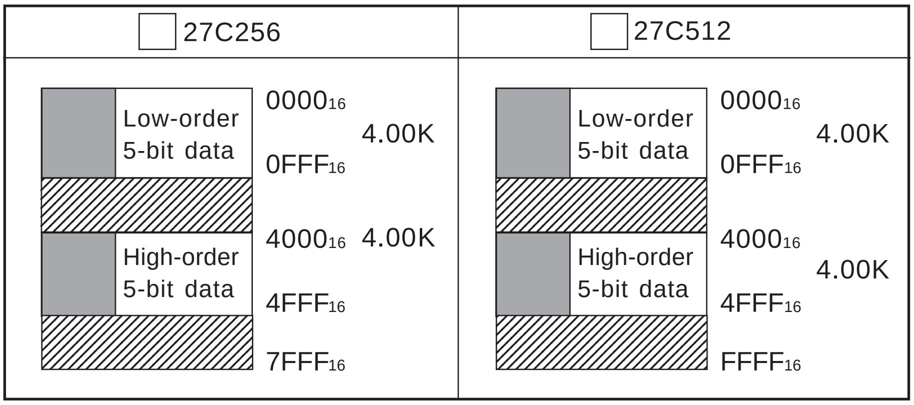
Three sets of EPROMs are required for each pattern (check in the approximate box).

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.

Microcomputer name: ☐ M34513M4-XXXSP ☐ M34513M4-XXXFP

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 (32P4B for M34513M4-XXXSP, 32P6B-A for M34513M4-XXXFP) and attach to the Mask ROM Order Confirmation Form.

✻ 3. Comments

GZZ-SH53-01B <85A0>

Mask ROM number	

4500 SERIES MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M34513M6-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

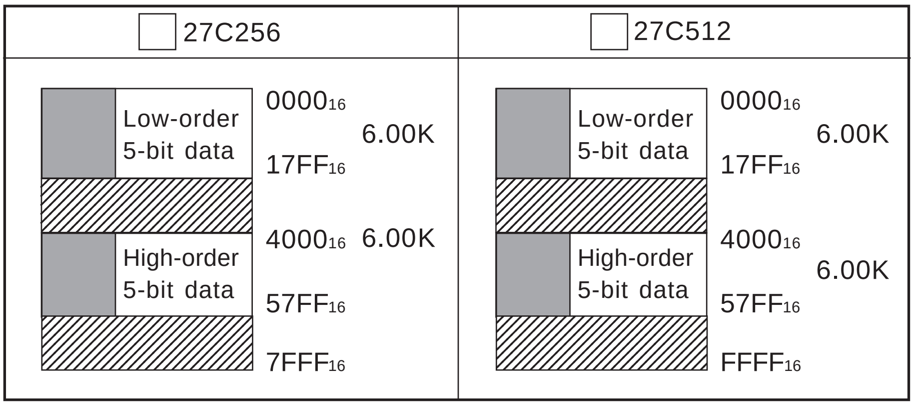
Specify the type of EPROMs submitted.

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 M34513M6-XXXFP) and attach to the Mask ROM Order Confirmation Form.

✻ 3. Comments

GZZ-SH52-99B <85A0>

Mask ROM number	

4500 SERIES MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M34513M8-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

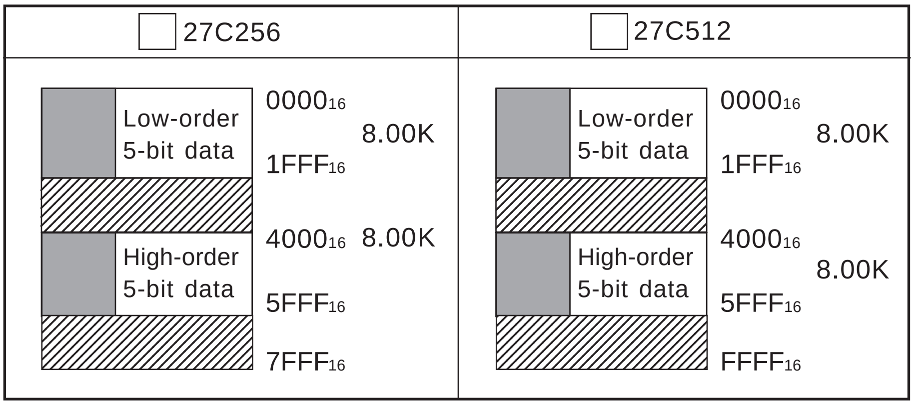
Specify the type of EPROMs submitted.

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 M34513M8-XXXFP) and attach to the Mask ROM Order Confirmation Form.

✻ 3. Comments

GZZ-SH52-41B <81A0>

Mask ROM number	

4500 SERIES MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M34514M6-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

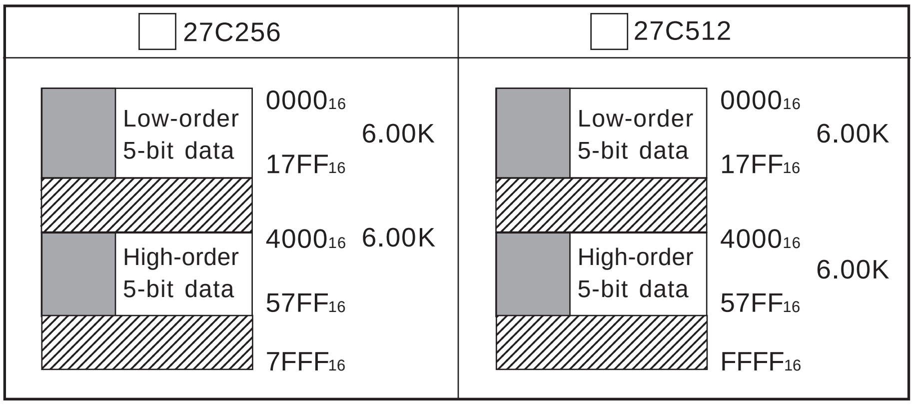
Specify the type of EPROMs submitted.

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 (42P2R-A for M34514M6-XXXFP) and attach to the Mask ROM Order Confirmation Form.

✻ 3. Comments

GZZ-SH52-40B <81A0>

Mask ROM number	

4500 SERIES MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP MICROCOMPUTER M34514M8-XXXFP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Please fill in all items marked ✻ .

✻ Customer	Company name	TEL ( )
	Date issued	Date:

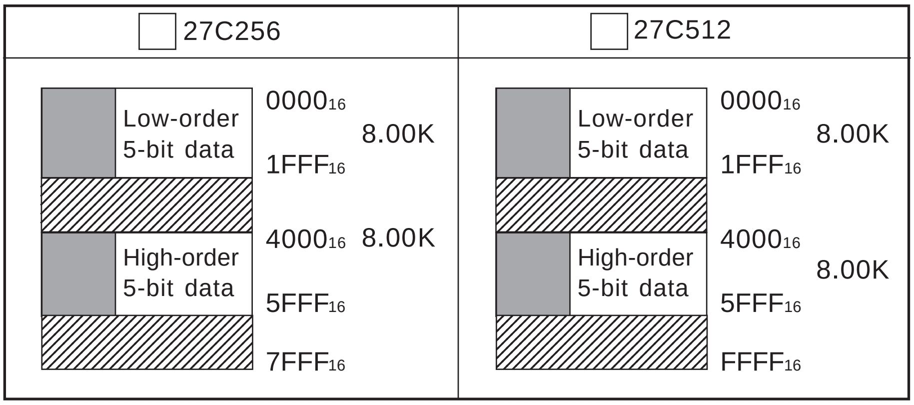
Issuance signature	Responsible officer	Supervisor

✻ 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 (42P2R-A for M34514M8-XXXFP) and attach to the Mask ROM Order Confirmation Form.

✻ 3. Comments

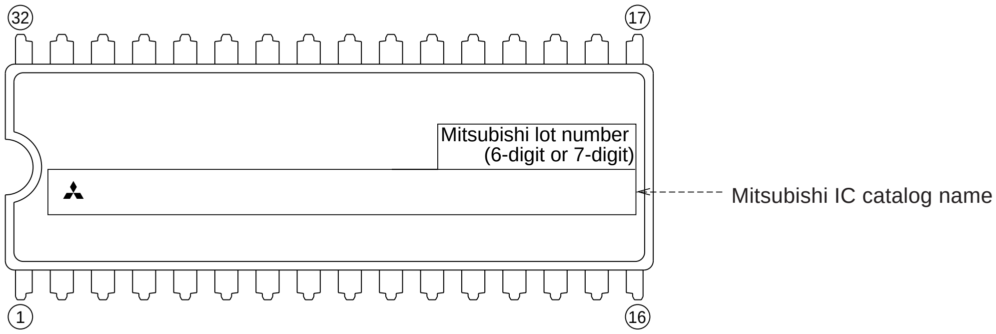
3.6 Mark specification form

32P4B (32-PIN SHRINK DIP) MARK SPECIFICATION FORM

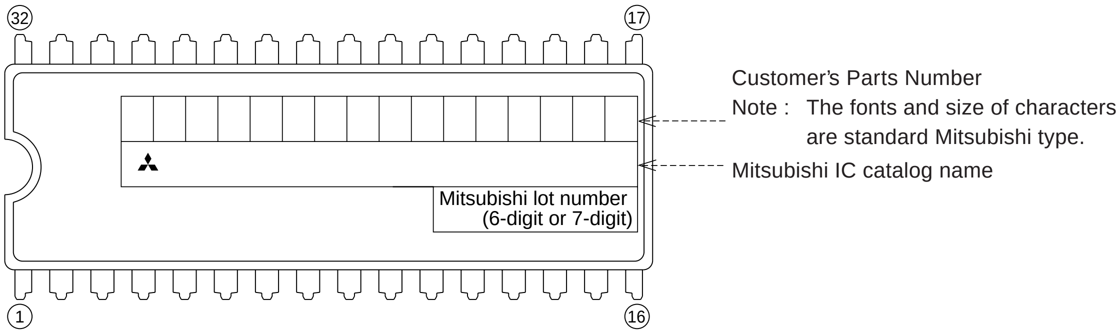
Mitsubishi IC catalog name

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

A. Standard Mitsubishi Mark



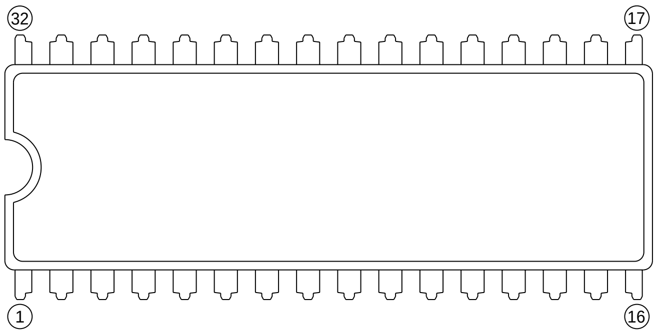
B. Customer’s Parts Number + Mitsubishi catalog name



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 16 characters : Only 0 ~ 9, A ~ Z, +, –, /, (,), &, ©, . (periods), and , (commas) are usable.
4 : If the Mitsubishi logo is not required, check the box on the right.

Mitsubishi logo is not required ☐

C. Special Mark Required



Note1 : If the Special Mark is to be Printed, indicate the desired layout of the mark in the upper figure. The layout will be duplicated as close as possible. Mitsubishi lot number (6-digit or 7-digit) and Mask ROM number (3-digit) are always marked.
2 : If the customer’s trade mark logo must be used in the Special Mark, check the box on the right. Please submit a clean original of the logo. For the new special character fonts a clean font original (ideally logo drawing) must be submitted.
3 : The standard Mitsubishi font is used for all characters except for a logo.

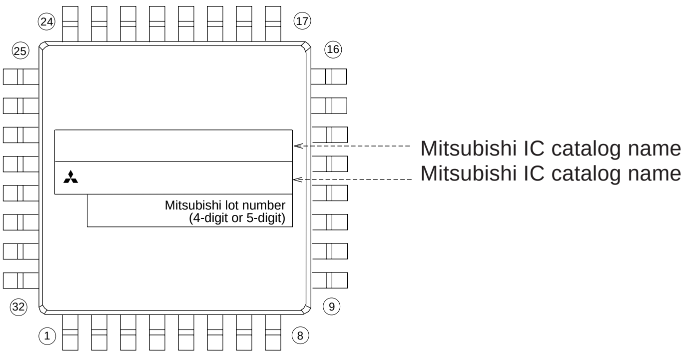
Special logo required ☐

32P6B (32-PIN LQFP) MARK SPECIFICATION FORM

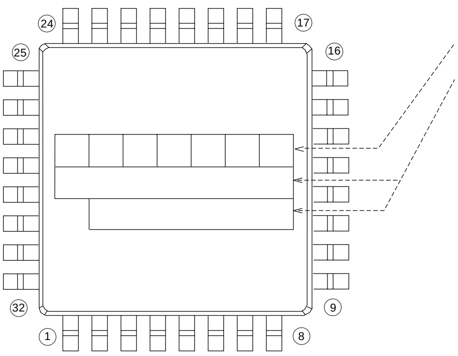
Mitsubishi IC catalog name

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



B. Customer's Parts Number + Mitsubishi catalog name



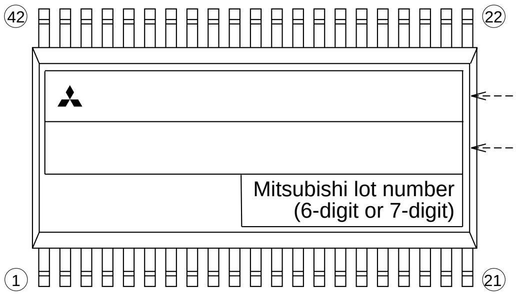
Customer's Parts Number
Note : The fonts and size of characters are standard Mitsubishi type.
Mitsubishi IC catalog name
Note1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's Parts Number can be up to 7 characters : Only 0 ~ 9, A ~ Z, +, –, /, (,), &, ©, . (periods), , (commas) are usable.

42P2R-A (42-PIN SHRINK SOP) MARK SPECIFICATION FORM

Mitsubishi IC catalog name

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

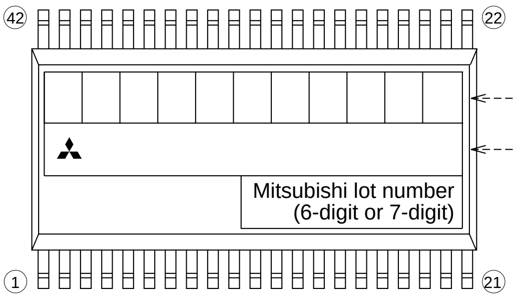
A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

Mitsubishi IC catalog name

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.

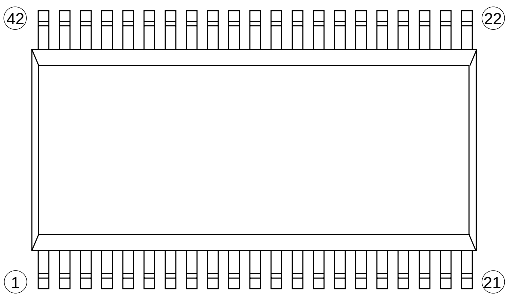
2 : The fonts and size of characters are standard Mitsubishi type.

3 : Customer's Parts Number can be up to 11 characters : Only 0 ~ 9, A ~ Z, +, –, /, (,), &, ©, . (periods), , (commas) are usable.

4 : If the Mitsubishi logo is not required, check the box below.

Mitsubishi logo is not required

C. Special Mark Required



Note1 : If the Special Mark is to be Printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated as close as possible.
Mitsubishi lot number (6-digit or 7-digit) and Mask ROM number (3-digit) are always marked.

2 : If the customer's trade mark logo must be used in the Special Mark, check the box below.
Please submit a clean original of the logo.
For the new special character fonts a clean font original (ideally logo drawing) must be submitted.

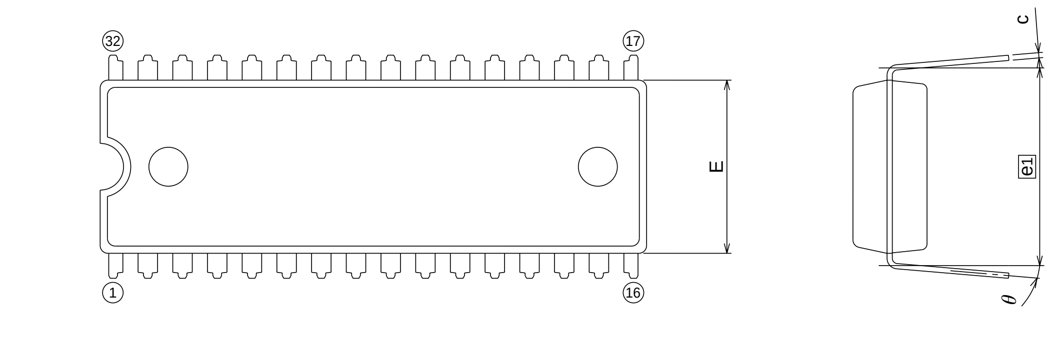
Special logo required

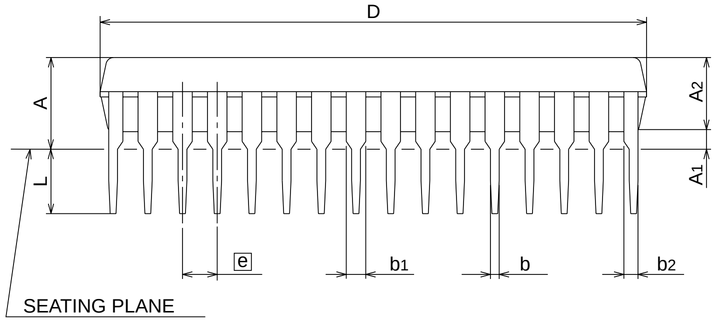
3.7 Package outline

32P4B

Plastic 32pin 400mil SDIP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SDIP32-P-400-1.78	–	2.2	Alloy 42/Cu Alloy



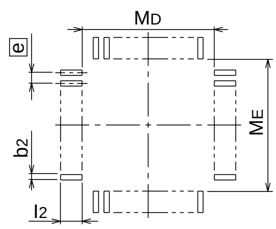


Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	5.08
A_1	0.51	–	–
A_2	–	3.8	–
b	0.35	0.45	0.55
b_1	0.9	1.0	1.3
b_2	0.63	0.73	1.03
c	0.22	0.27	0.34
D	27.8	28.0	28.2
E	8.75	8.9	9.05
$\boxed{e}$	–	1.778	–
$\boxed{e_1}$	–	10.16	–
L	3.0	–	–
θ	0°	–	15°

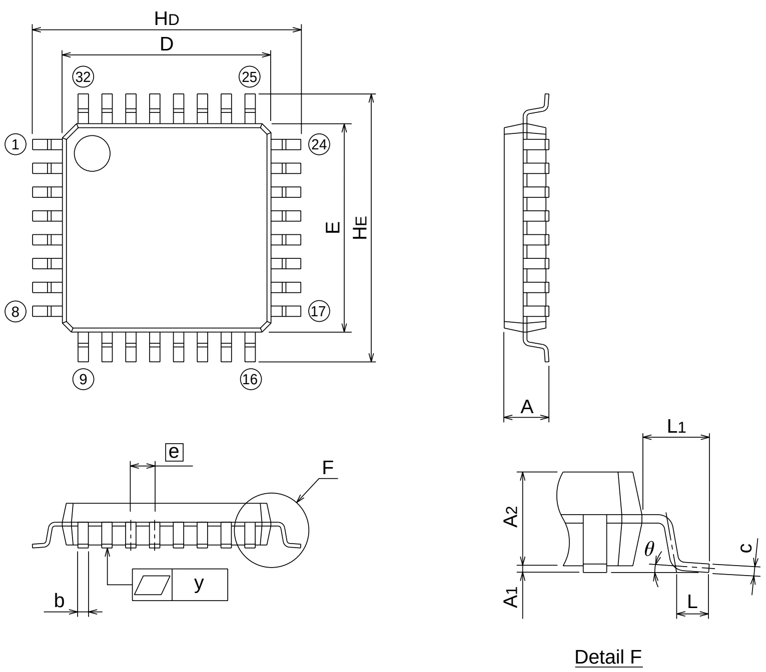
32P6B-A

Plastic 32pin 7×7mm body LQFP

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



Recommended Mount Pad

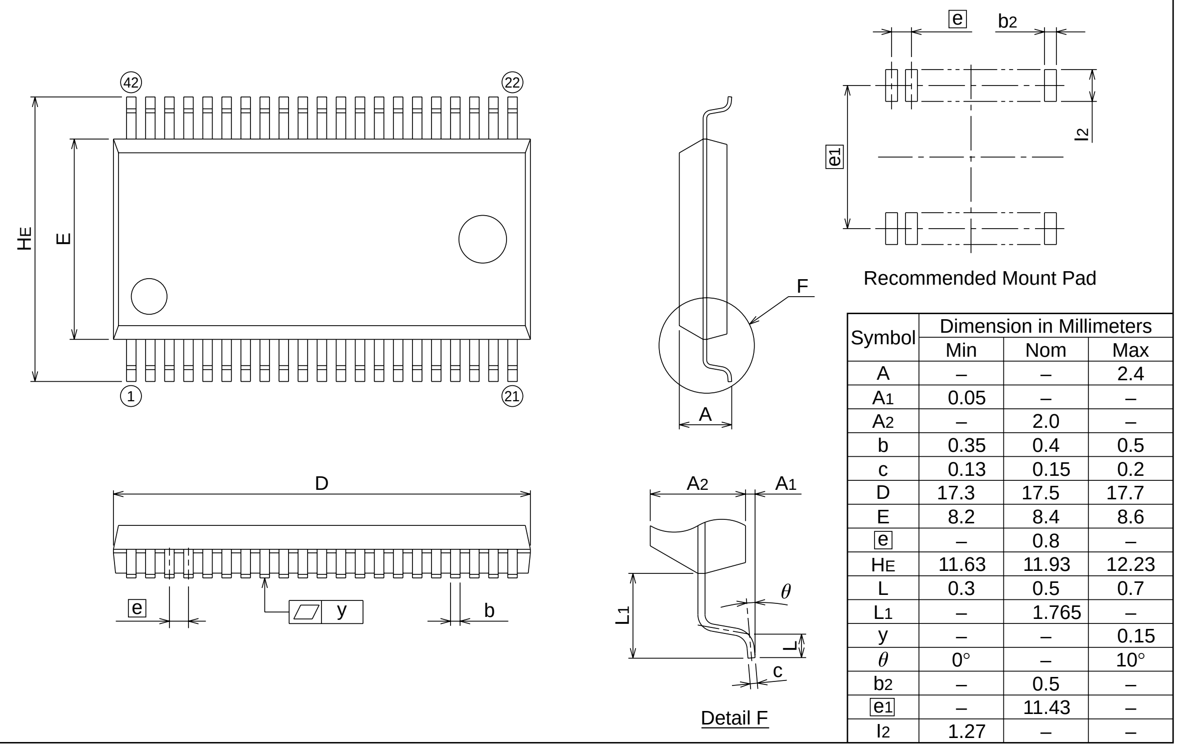


Detail F

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
$\boxed{e}$	–	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	–
I_2	1.0	–	–
M_D	–	7.4	–
M_E	–	7.4	–

42P2R-A **Plastic 42pin 450mil SSOP**

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SSOP42-P-450-0.80	–	0.63	Alloy 42/Cu Alloy



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	2.4
A_1	0.05	–	–
A_2	–	2.0	–
b	0.35	0.4	0.5
c	0.13	0.15	0.2
D	17.3	17.5	17.7
E	8.2	8.4	8.6
$\boxed{e}$	–	0.8	–
H_E	11.63	11.93	12.23
L	0.3	0.5	0.7
L_1	–	1.765	–
y	–	–	0.15
θ	0°	–	10°
b_2	–	0.5	–
$\boxed{e_1}$	–	11.43	–
I_2	1.27	–	–

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
4513/4514 Group

Dec. First Edition 1998

Editioned by
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

REVISION DESCRIPTION LIST	4513/4514 GROUP USER'S MANUAL

Rev. No.	Revision Description	Rev. date
1.0	First Edition	981211

User's Manual
4513/4514 Group

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

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