REJ09B0175-0100Z

4

4519 Group

User's Manual

RENESAS 4-BIT CISC SINGLE-CHIP MICROCOMPUTER

720 FAMILY / 4500 SERIES

User's Manual

Before using this material, please visit our website to confirm that this is the most current document available.

Rev. 1.00
Revision date: Aug 06, 2004

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. 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 nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. 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 Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. 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 Renesas Technology Corp. or an authorized Renesas Technology Corp. 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.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. 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.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

REVISION HISTORY	4519 Group User's Manual

Rev.	Date	Description	
		Page	Summary
1.00	Aug 06, 2004	–	First edition issued

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 necessary information for systems development using the microcomputer, such as the electrical characteristics, the list of registers.

  As for the Mask ROM confirmation form, the ROM programming confirmation form, and the Mark specification form which are to be submitted when ordering, refer to the "Renesas Technology Corp." Hompage (http://www.renesas.com/en/rom).

  As for the Development tools and related documents, refer to the Product Info - 4519 Group (http://www.renesas.com/eng/products/mpumcu/specific/lcd_mcu/expand/e4519.htm) of "Renesas Technology Corp." Homepage.

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

The 4519 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 one or two reload registers), a 10-bit A/D converter, interrupts, and oscillation circuit switch function.

The various microcomputers in the 4519 Group include variations of the built-in memory size as shown in the table below.

FEATURES

- Minimum instruction execution time ..... 0.5 μs
  (at 6 MHz oscillation frequency, in X_{IN} through-mode)
- Supply voltage
  Mask ROM version ..... 1.8 to 5.5 V
  One Time PROM version ..... 2.5 to 5.5 V
  (It depends on operation source clock, oscillation frequency and operation mode)
- 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 3 ..... 8-bit timer with two reload registers
- Interrupt ..... 8 sources
- Key-on wakeup function pins 10
- Serial I/O ..... 8 bits × 1
- A/D converter ..... 10-bit successive comparison method, 8ch
- Voltage drop detection circuit
  Reset occurrence ..... Typ. 3.5 V (Ta = 25 °C)
  Reset release ..... Typ. 3.7 V (Ta = 25 °C)
- Watchdog timer
- Clock generating circuit
  (ceramic resonator/RC oscillation/quartz-crystal oscillation/on-chip oscillator)
- LED drive directly enabled (port D)

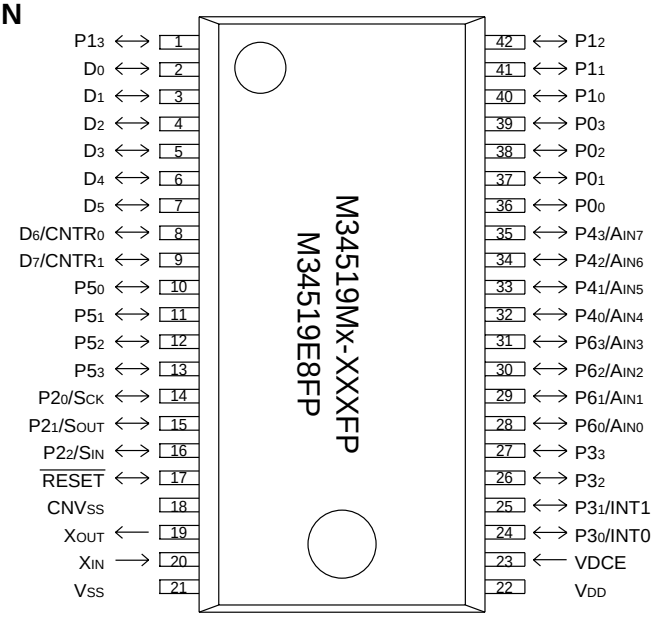
APPLICATION

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

Part number	ROM (PROM) size (× 10 bits)	RAM size (× 4 bits)	Package	ROM type
M34519M6-XXXFP	6144 words	384 words	42P2R-A	Mask ROM
M34519M8-XXXFP	8192 words	384 words	42P2R-A	Mask ROM
M34519E8FP (**Note**)	8192 words	384 words	42P2R-A	One Time PROM

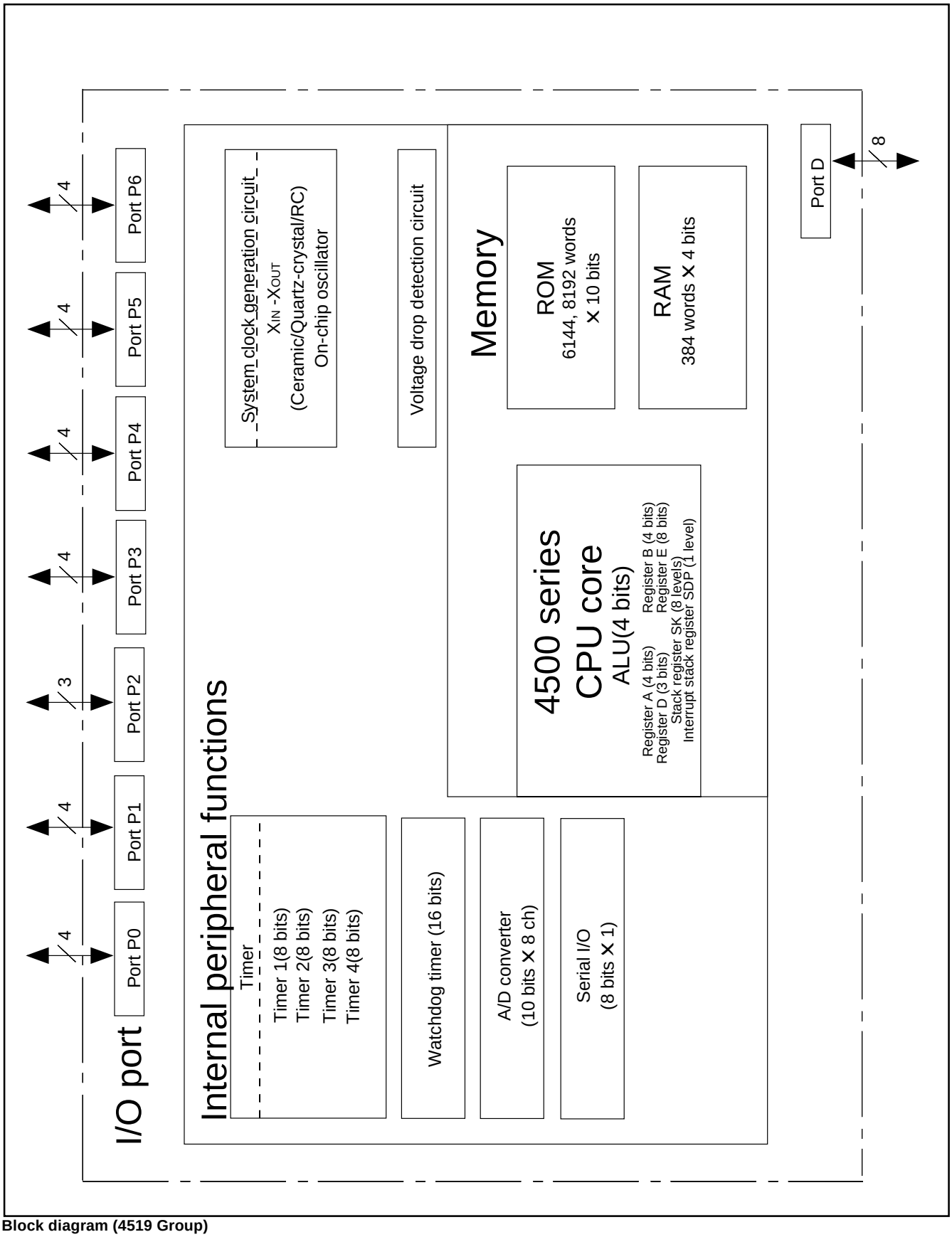
Note: Shipped in blank.

PIN CONFIGURATION



OUTLINE 42P2R-A

Pin configuration (top view) (4519 Group)



Block diagram (4519 Group)

PERFORMANCE OVERVIEW

Parameter			Function
Number of basic instructions			153
Minimum instruction execution time			0.5 μs (at 6.0 MHz oscillation frequency, in XIN through-mode)
Memory sizes	ROM	M34519M6	6144 words × 10 bits
		M34519M8/E8	8192 words × 10 bits
	RAM	M34519M6/M8/E8	384 words × 4 bits
Input/Output ports	D0–D7	I/O (Input is examined by skip decision)	Eight independent I/O ports; Ports D6 and D7 are also used as CNTR0 and CNTR1, respectively. The output structure is switched by software.
	P00–P03	I/O	4-bit I/O port; a pull-up function, a key-on wakeup function and output structure can be switched by software.
	P10–P13	I/O	4-bit I/O port; a pull-up function, a key-on wakeup function and output structure can be switched by software.
	P20–P22	I/O	3-bit I/O port; ports P20, P21 and P22 are also used as SCK, SOUT and SIN, respectively.
	P30–P33	I/O	4-bit I/O port ; ports P30 and P31 are also used as INT0 and INT1, respectively.
	P40–P43	I/O	4-bit I/O port ; ports P40–P43 are also used as AIN4–AIN7, respectively.
	P50–P53	I/O	4-bit I/O port ; the output structure is switched by software.
	P60–P63	I/O	4-bit I/O port ; ports P60–P63 are also used as AIN0–AIN3, respectively.
Timers	Timer 1		8-bit timer with a reload register is also used as an event counter. Also, this is equipped with a period/pulse width measurement function.
	Timer 2		8-bit timer with a reload register.
	Timer 3		8-bit timer with a reload register is also used as an event counter.
	Timer 4		8-bit timer with two reload registers and PWM output function.
A/D converter			10-bit wide × 8 ch, This is equipped with an 8-bit comparator function.
Serial I/O			8-bit × 1
Interrupt	Sources		8 (two for external, four for timer, one for A/D, and one for serial I/O)
	Nesting		1 level
Subroutine nesting			8 levels
Device structure			CMOS silicon gate
Package			42-pin plastic molded SSOP (42P2R-A)
Operating temperature range			–20 °C to 85 °C
Supply voltage	Mask ROM version		1.8 V to 5.5 V (It depends on operation source clock, oscillation frequency and operating mode.)
	One Time PROM version		2.5 V to 5.5 V (It depends on operation source clock, oscillation frequency and operating mode.)
Power dissipation (typical value)	Active mode		2.8 mA (Ta=25°C, VDD=5V, f(XIN)=6 MHz, f(STCK)=f(XIN), on-chip oscillator stop)
			70 µA (Ta=25°C, VDD=5V, f(XIN)=32 kHz, f(STCK)=f(XIN), on-chip oscillator stop)
			150 µA (Ta=25°C, VDD=5V, on-chip oscillator is used, f(STCK)=f(RING), f(XIN) stop)
	RAM back-up mode		0.1 μA (Ta=25°C, VDD = 5 V, output transistors in the cut-off state)

PIN DESCRIPTION

Pin	Name	Input/Output	Function
V_{DD}	Power supply	—	Connected to a plus power supply.
V_{SS}	Ground	—	Connected to a 0 V power supply.
CNV_{SS}	CNV_{SS}	—	Connect CNV_{SS} to V_{SS} and apply "L" (0V) to CNV_{SS} certainly.
VDCE	Voltage drop detection circuit enable	Input	This pin is used to operate/stop the voltage drop detection circuit. When "H" level is input to this pin, the circuit starts operating. When "L" level is input to this pin, the circuit stops operating.
$\overline{RESET}$	Reset input/output	I/O	An N-channel open-drain I/O pin for a system reset. When the SRST instruction, watchdog timer, the built-in power-on reset or the voltage drop detection circuit causes the system to be reset, the $\overline{RESET}$ pin outputs "L" level.
X_{IN}	Main clock input	Input	I/O pins of the main clock generating circuit. When using a ceramic resonator, connect it between pins X_{IN} and X_{OUT}. When using a 32 kHz quartz-crystal oscillator, connect it between pins X_{IN} and X_{OUT}. A feedback resistor is built-in between them. When using the RC oscillation, connect a resistor and a capacitor to X_{IN}, and leave X_{OUT} pin open.
X_{OUT}	Main 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. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Ports D_6, D_7 is also used as CNTR0 pin and CNTR1 pin, respectively.
$P0_0$–$P0_3$	I/O port P0	I/O	Port P0 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Port P0 has a key-on wakeup function and a pull-up function. Both functions can be switched by software.
$P1_0$–$P1_3$	I/O port P1	I/O	Port P1 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain. Port P1 has a key-on wakeup function and a pull-up function. Both functions can be switched by software.
$P2_0$–$P2_3$	I/O port P2	I/O	Port P2 serves as a 3-bit I/O port. The output structure is N-channel open-drain. For input use, set the latch of the specified bit to "1". Ports $P2_0$–$P2_2$ are also used as S_{CK}, S_{OUT}, S_{IN}, respectively.
$P3_0$–$P3_3$	I/O port P3	I/O	Port P3 serves as a 4-bit I/O port. The output structure is N-channel open-drain. For input use, set the latch of the specified bit to "1". Ports $P3_0$ and $P3_1$ are also used as INT0 pin and INT1 pin, respectively.
$P4_0$–$P4_3$	I/O port P4	I/O	Port P4 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain. For input use, set the latch of the specified bit to "1". Ports $P4_0$–$P4_3$ are also used as A_{IN4}–A_{IN7}, respectively.
$P5_0$–$P5_3$	I/O port P5	I/O	Port P5 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to "1" and select the N-channel open-drain.
$P6_0$–$P6_3$	I/O port P6	I/O	Port P6 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain. For input use, set the latch of the specified bit to "1". Ports $P6_0$–$P6_3$ are also used as A_{IN0}–A_{IN3}, respectively.
CNTR0, CNTR1	Timer input/output	I/O	CNTR0 pin has the function to input the clock for the timer 1 event counter, and to output the timer 1 or timer 2 underflow signal divided by 2. CNTR1 pin has the function to input the clock for the timer 3 event counter, and to output the PWM signal generated by timer 4.CNTR0 pin and CNTR1 pin are also used as Ports D_6 and D_7, respectively.
INT0, INT1	Interrupt input	Input	INT0 pin and INT1 pin accept external interrupts. They have the key-on wakeup function which can be switched by software. INT0 pin and INT1 pin are also used as Ports $P3_0$ and $P3_1$, respectively.
A_{IN0}–A_{IN7}	Analog input	Input	A/D converter analog input pins. A_{IN0}–A_{IN7} are also used as ports $P6_0$–$P6_3$ and $P4_0$–$P4_3$, respectively.
S_{CK}	Serial I/O data I/O	I/O	Serial I/O data transfer synchronous clock I/O pin. S_{CK} pin is also used as port $P2_0$..
S_{OUT}	Serial I/O data output	Output	Serial I/O data output pin. S_{OUT} pin is also used as port $P2_1$.
S_{IN}	Serial I/O clock input	Input	Serial I/O data input pin. S_{IN} pin is also used as port $P2_2$.

MULTIFUNCTION

Pin	Multifunction	Pin	Multifunction	Pin	Multifunction	Pin	Multifunction
D_6	CNTR0	CNTR0	D_6	$P6_0$	AIN_0	AIN_0	$P6_0$
D_7	CNTR1	CNTR1	D_7	$P6_1$	AIN_1	AIN_1	$P6_1$
$P2_0$	SCK	SCK	$P2_0$	$P6_2$	AIN_2	AIN_2	$P6_2$
$P2_1$	SOUT	SOUT	$P2_1$	$P6_3$	AIN_3	AIN_3	$P6_3$
$P2_2$	SIN	SIN	$P2_2$	$P4_0$	AIN_4	AIN_4	$P4_0$
$P3_0$	INT0	INT0	$P3_0$	$P4_1$	AIN_5	AIN_5	$P4_1$
$P3_1$	INT1	INT1	$P3_1$	$P4_2$	AIN_6	AIN_6	$P4_2$
				$P4_3$	AIN_7	AIN_7	$P4_3$

Notes 1: Pins except above have just single function.
2: The input/output of $P3_0$ and $P3_1$ can be used even when INT0 and INT1 are selected.
3: The input of ports $P2_0$–$P2_2$ can be used even when SIN, SOUT and SCK are selected.
4: The input/output of D_6 can be used even when CNTR0 (input) is selected.
5: The input of D_6 can be used even when CNTR0 (output) is selected.
6: The input/output of D_7 can be used even when CNTR1 (input) is selected.
7: The input of D_7 can be used even when CNTR1 (output) is selected.

DEFINITION OF CLOCK AND CYCLE

● Operation source clock
The operation source clock is the source clock to operate this product. In this product, the following clocks are used.
- Clock (f(XIN)) by the external ceramic resonator
- Clock (f(XIN)) by the external RC oscillation
- Clock (f(XIN)) by the external input
- Clock (f(RING)) of the on-chip oscillator which is the internal oscillator
- Clock (f(XIN)) by the external quartz-crystal oscillation

● System clock (STCK)
The system clock is the basic clock for controlling this product. The system clock is selected by the clock control register MR shown as the table below.

● Instruction clock (INSTCK)
The instruction clock is the basic clock for controlling CPU. The instruction clock (INSTCK) is a signal derived by dividing the system clock (STCK) by 3. The one instruction clock cycle generates the one machine cycle.

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

Table Selection of system clock

Register MR				System clock	Operation mode
MR_3	MR_2	MR_1	MR_0		
0	0	0	0	f(STCK) = f(XIN)	XIN through mode
		×	1	f(STCK) = f(RING)	On-chip oscillator through mode
0	1	0	0	f(STCK) = f(XIN)/2	XIN divided by 2 mode
		×	1	f(STCK) = f(RING)/2	On-chip oscillator divided by 2 mode
1	0	0	0	f(STCK) = f(XIN)/4	XIN divided by 4 mode
		×	1	f(STCK) = f(RING)/4	On-chip oscillator divided by 4 mode
1	1	0	0	f(STCK) = f(XIN)/8	XIN divided by 8 mode
		×	1	f(STCK) = f(RING)/8	On-chip oscillator divided by 8 mode

×: 0 or 1

Note: The f(RING)/8 is selected after system is released from reset. When on-chip oscillator clock is selected for main clock, set the on-chip oscillator to be operating state.

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/ CMOS	1	SD, RD SZD CLD	FR1, FR2 W6 W4	Output structure selection function (programmable)
Port P0	$P0_0$–$P0_3$	I/O (4)	N-channel open-drain/ CMOS	4	OP0A IAP0	FR0 PU0 K0, K1	Built-in programmable pull-up functions, key-on wakeup functions and output structure selection functions
Port P1	$P1_0$–$P1_3$	I/O (4)	N-channel open-drain/ CMOS	4	OP1A IAP1	FR0 PU1 K0	Built-in programmable pull-up functions, key-on wakeup functions and output structure selection functions
Port P2	$P2_0$/SCK, $P2_1$/SOUT $P2_2$/SIN	I/O (3)	N-channel open-drain	3	OP2A IAP2	J1	
Port P3	$P3_0$/INT0, $P3_1$/INT1 $P3_2$, $P3_3$	I/O (4)	N-channel open-drain	4	OP3A IAP3	I1, I2 K2	
Port P4	$P4_0$/AIN4–$P4_3$/AIN7	I/O (4)	N-channel open-drain	4	OP4A IAP4	Q1 Q2	
Port P5	$P5_0$–$P5_3$	I/O (4)	N-channel open-drain/ CMOS	4	OP5A IAP5	FR3	Output structure selection function (programmable)
Port P6	$P6_0$/AIN0–$P6_3$/AIN3	I/O (4)	N-channel open-drain	4	OP6A IAP6	Q2 Q1	

CONNECTIONS OF UNUSED PINS

Pin	Connection	Usage condition	
X_{IN}	Open.	Internal oscillator is selected.	(Note 1)
X_{OUT}	Open.	Internal oscillator is selected.	(Note 1)
		RC oscillator is selected.	(Note 2)
		External clock input is selected for main clock.	(Note 3)
D_0–D_5	Open.		
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.	(Note 4)
D_6/CNTR0	Open.	CNTR0 input is not selected for timer 1 count source.	
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.	(Note 4)
D_7/CNTR1	Open.	CNTR1 input is not selected for timer 3 count source.	
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.	(Note 4)
$P0_0$–$P0_3$	Open.	The key-on wakeup function is not selected.	(Note 6)
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.	(Note 5)
		The pull-up function is not selected.	(Note 4)
		The key-on wakeup function is not selected.	(Note 6)
$P1_0$–$P1_3$	Open.	The key-on wakeup function is not selected.	(Note 7)
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.	(Note 5)
		The pull-up function is not selected.	(Note 4)
		The key-on wakeup function is not selected.	(Note 7)
$P2_0$/S_{CK}	Open.	S_{CK} pin is not selected.	
	Connect to V_{SS}.	————	
$P2_1$/S_{OUT}	Open.	————	
	Connect to V_{SS}.	————	
$P2_2$/S_{IN}	Open.	S_{IN} pin is not selected.	
	Connect to V_{SS}.	————	
$P3_0$/INT0	Open.	“0” is set to output latch.	
	Connect to Vss.	————	
$P3_1$/INT1	Open.	“0” is set to output latch.	
	Connect to Vss.	————	
$P3_2$, $P3_3$	Open.	————	
	Connect to Vss.	————	
$P4_0$/A_{IN4}–$P4_3$/A_{IN7}	Open.	————	
	Connect to Vss.	————	
$P5_0$–$P5_3$	Open.	————	
	Connect to Vss.	N-channel open-drain is selected for the output structure.	
$P6_0$/A_{IN0}–$P6_3$/A_{IN3}	Open.	————	
	Connect to Vss.	————	

Notes 1: After system is released from reset, the internal oscillation (on-chip oscillator) is selected for system clock (RG_0=0, MR_0=1).

2: When the CRCK instruction is executed, the RC oscillation circuit becomes valid. Be careful that the swich of system clock is not executed at oscillation start only by the CRCK instruction execution.
In order to start oscillation, setting the main clock f(X_{IN}) oscillation to be valid (MR_1=0) is required. (If necessary, generate the oscillation stabilizing wait time by software.)
Also, when the main clock (f(X_{IN})) is selected as system clock, set the main clock f(X_{IN}) oscillation (MR_1=0) to be valid, and select main clock f(X_{IN}) (MR_0=0). Be careful that the switch of system clock cannot be executed at the same time when main clock oscillation is started.

3: In order to use the external clock input for the main clock, select the ceramic resonance by executing the CMCK instruction at the beggining of software, and then set the main clock (f(X_{IN})) oscillation to be valid (MR_1=0). Until the main clock (f(X_{IN})) oscillation becomes valid (MR_1=0) after ceramic resonance becomes valid, X_{IN} pin is fixed to “H”. When an external clock is used, insert a 1 kΩ resistor to X_{IN} pin in series for limits of current.

4: Be sure to select the output structure of ports D_0–D_5 and the pull-up function of $P0_0$–$P0_3$ and $P1_0$–$P1_3$ with every one port. Set the corresponding bits of registers for each port.

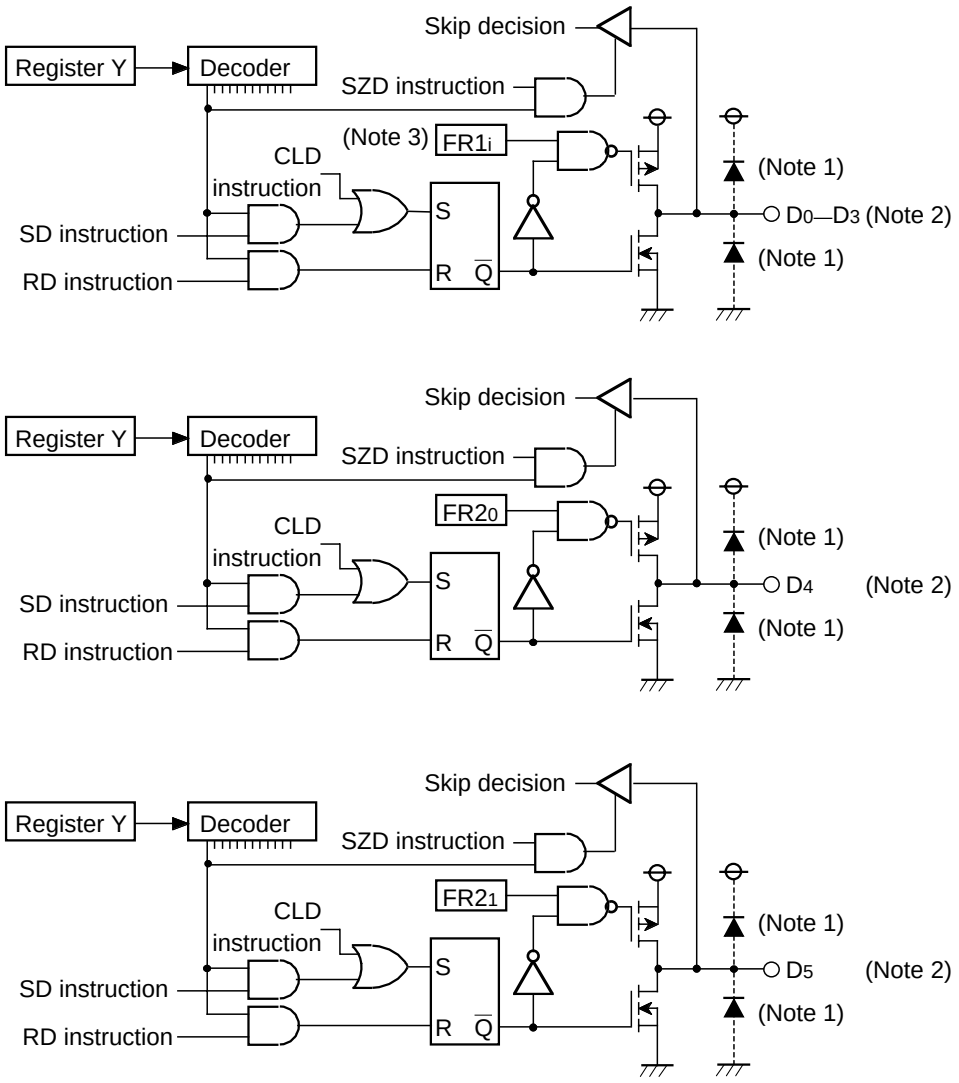
5: Be sure to select the output structure of ports $P0_0$–$P0_3$ and $P1_0$–$P1_3$ with every two ports. If only one of the two pins is used, leave another one open.

6: The key-on wakeup function is selected with every two bits. When only one of key-on wakeup function is used, considering that the value of key-on wake-up control register K1, set the unused 1-bit to “H” input (turn pull-up transistor ON and open) or “L” input (connect to V_{SS}, or open and set the output latch to “0”).

7: The key-on wakeup function is selected with every two bits. When one of key-on wakeup function is used, turn pull-up transistor of unused one ON and open.

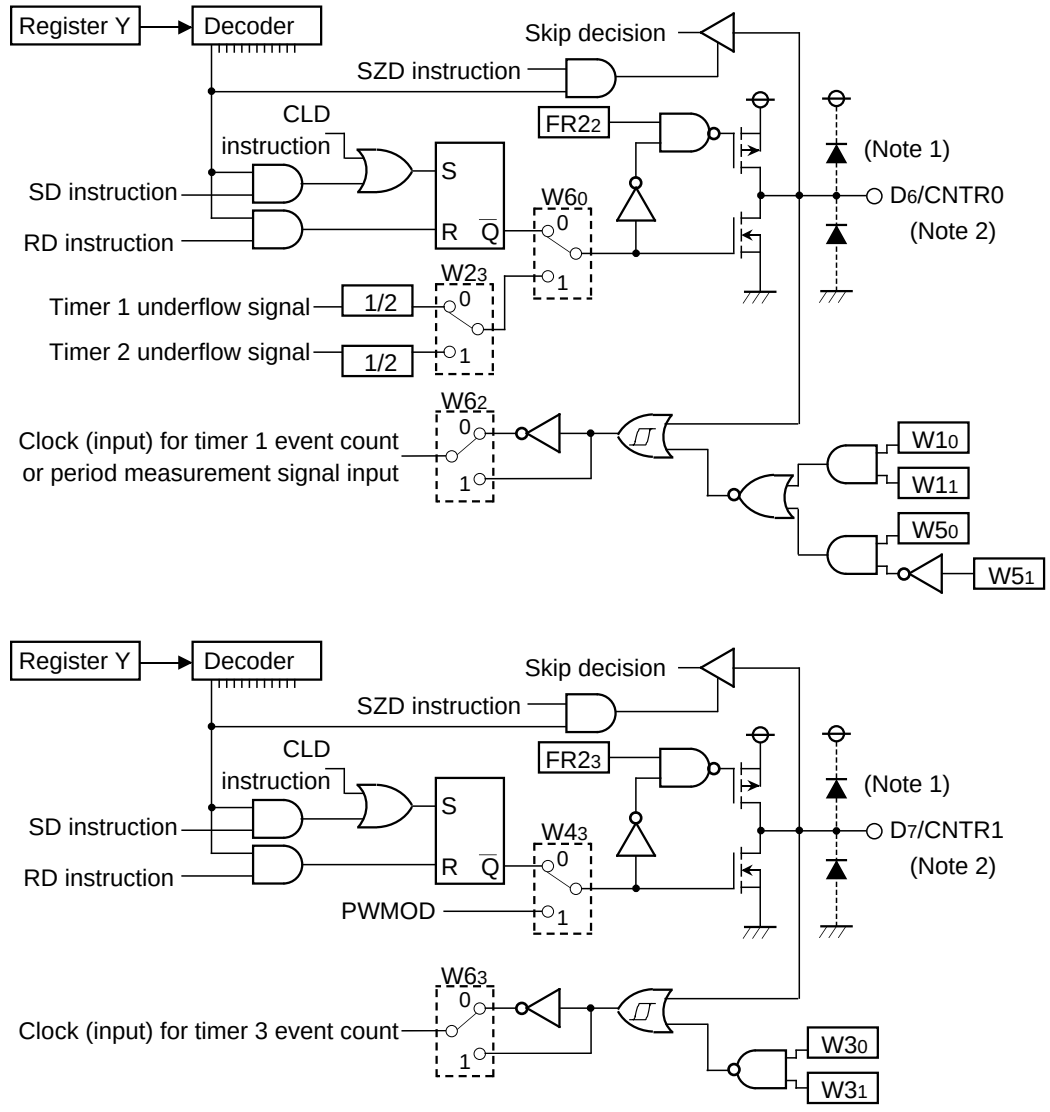
(Note when connecting to V_{SS} and V_{DD})
● Connect the unused pins to V_{SS} and V_{DD} using the thickest wire at the shortest distance against noise.

PORT BLOCK DIAGRAMS



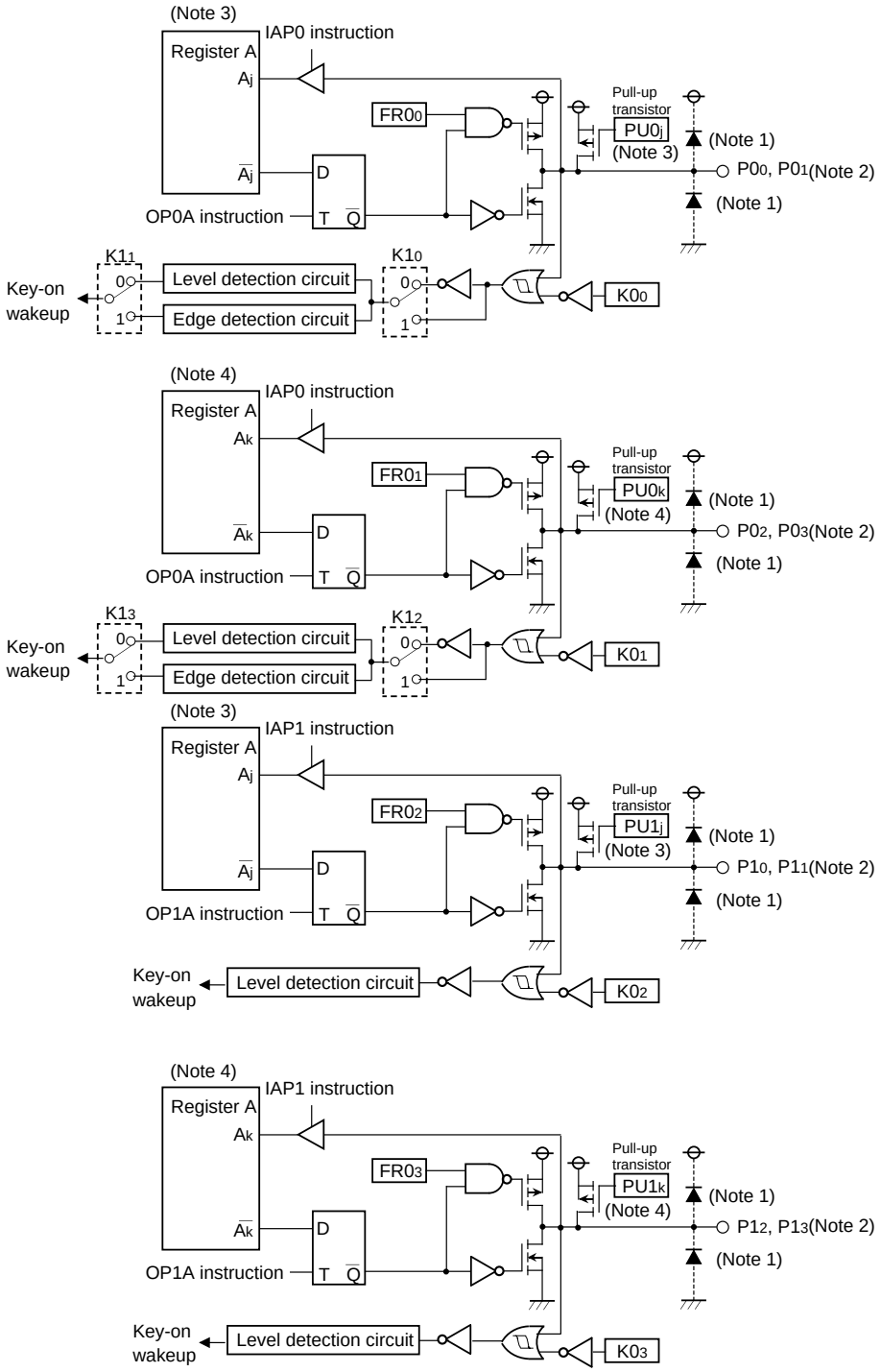
Notes 1: ----|◀---- This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be V_{DD} or less.
3: i represents bits 0 to 3.

Port block diagram (1)



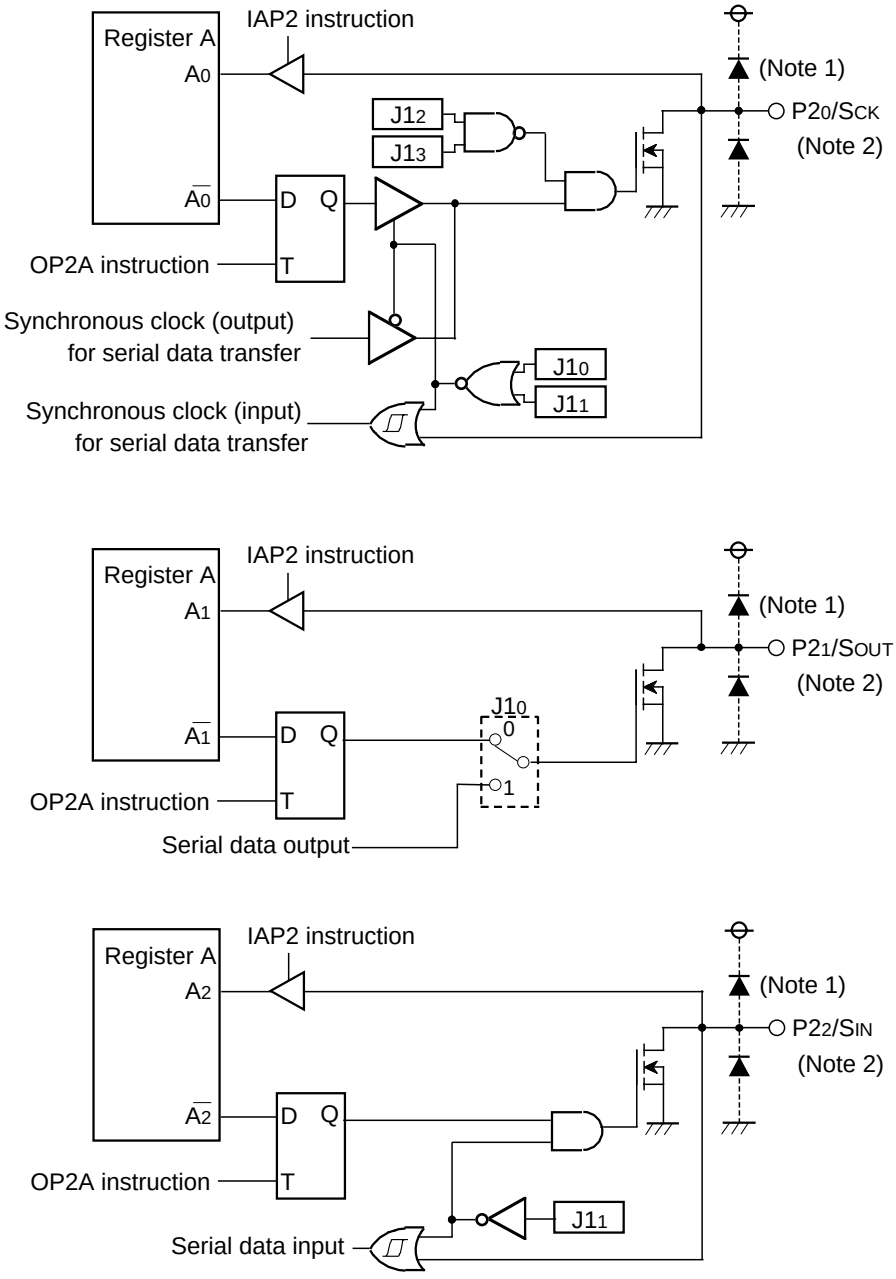
Notes 1: ----|◀---- This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be V_{DD} or less.

Port block diagram (2)



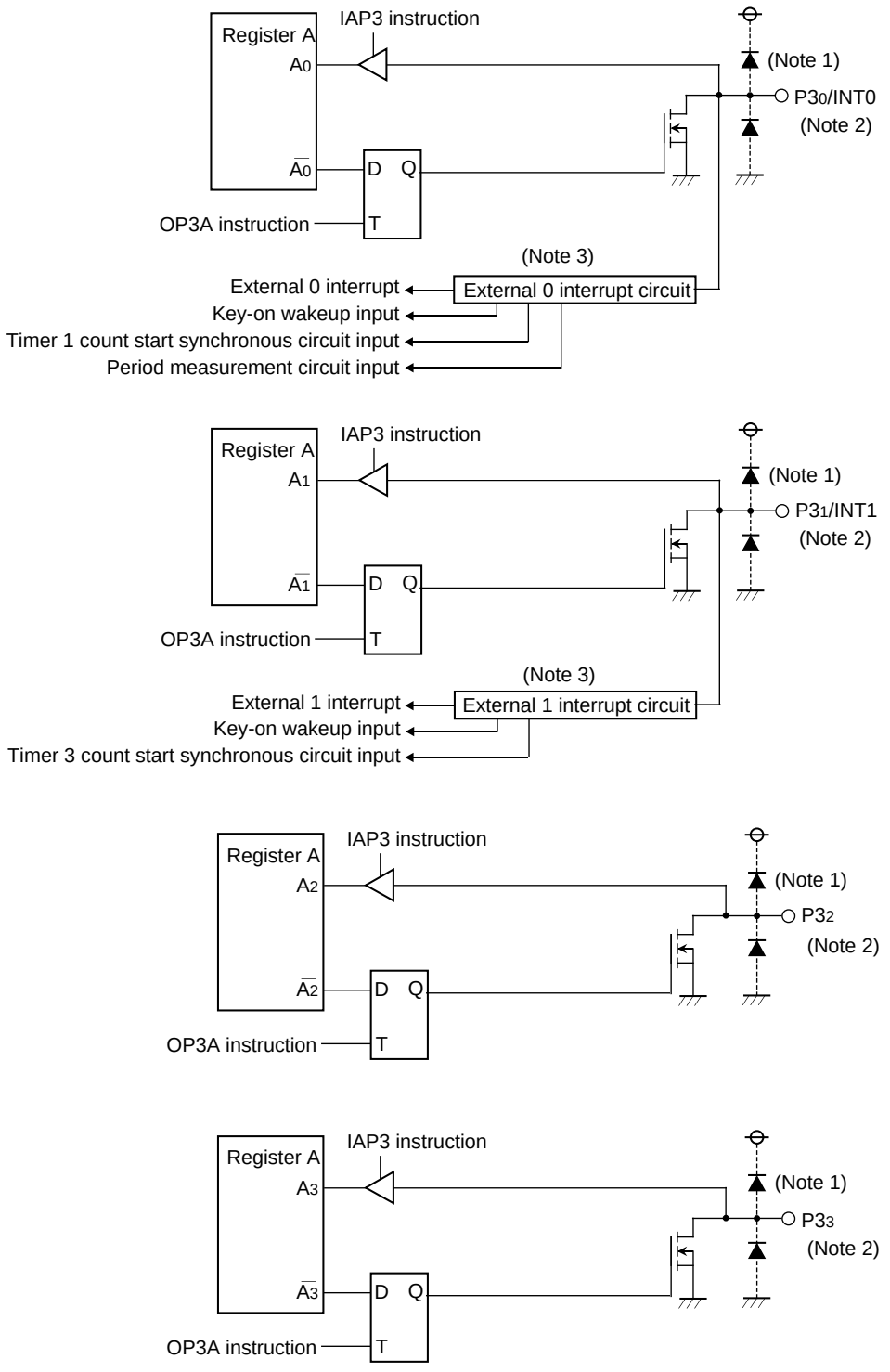
Notes 1: ----|◀---- This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be V_{DD} or less.
3: j represents bits 0 and 1.
4: k represents bits 2 and 3.

Port block diagram (3)



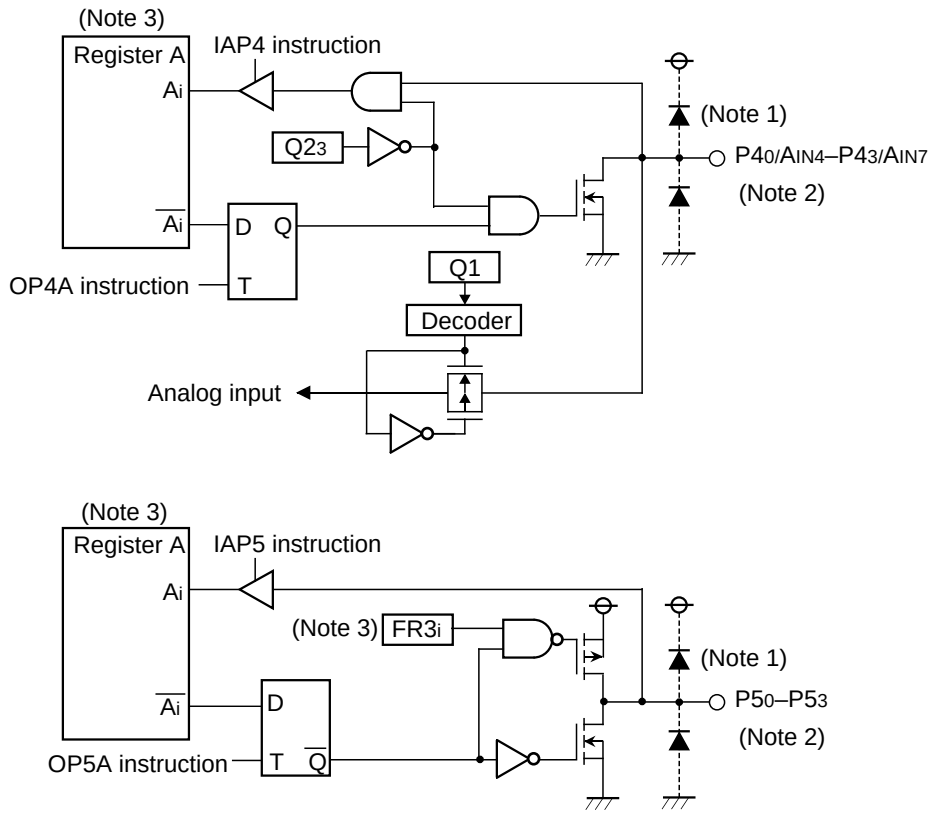
Notes 1: ----|◀---- This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be V_{DD} or less.

Port block diagram (4)



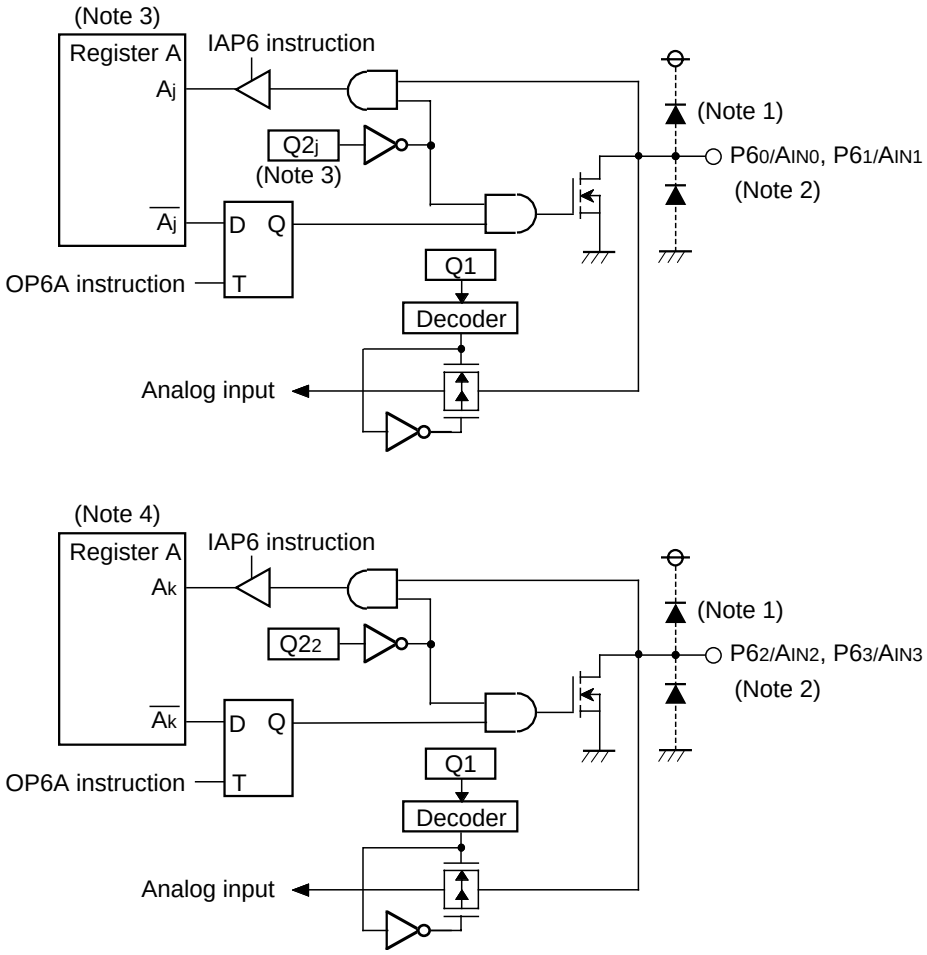
Notes 1: ----|◀---- This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be V_{DD} or less.
3: As for details, refer to the external interrupt circuit structure.

Port block diagram (5)



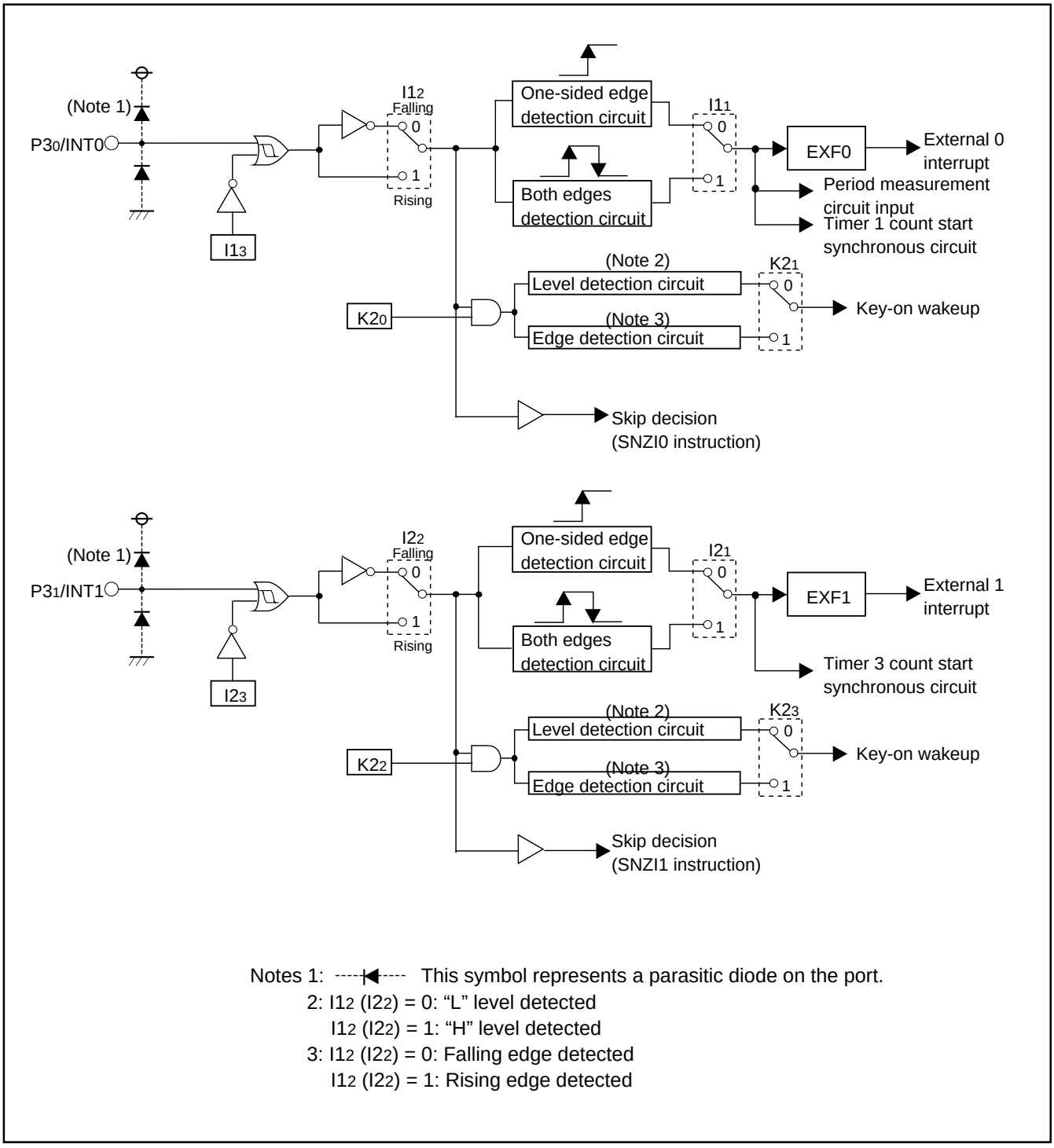
Notes 1: This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be V_{DD} or less.
3: i represents bits 0 to 3.

Port block diagram (6)



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

Port block diagram (7)



Notes 1: ----|◄---- This symbol represents a parasitic diode on the port.
2: I12 (I22) = 0: "L" level detected
I12 (I22) = 1: "H" level detected
3: I12 (I22) = 0: Falling edge detected
I12 (I22) = 1: Rising edge detected

Port block diagram (8)

FUNCTION BLOCK OPERATIONS
CPU

(1) Arithmetic logic unit (ALU)

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

(2) Register A and carry flag

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

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

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

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

(3) Registers B and E

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

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

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

(4) Register D

Register D is a 3-bit register.

It is used to store a 7-bit ROM address together with register A and is used as a pointer within the specified page when the TABP p, BLA p, or BMLA p instruction is executed. Also, when the TABP p instruction is executed, the high-order 2 bits of the reference data in ROM is stored to the low-order 2 bits of register D, and the contents of the high-order 1 bit of register D is “0”. (Figure 4).

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

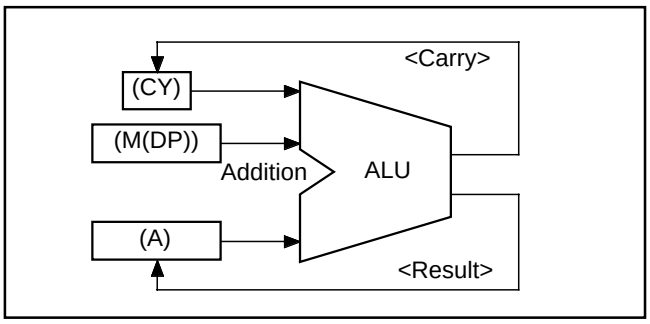


Fig. 1 AMC instruction execution example

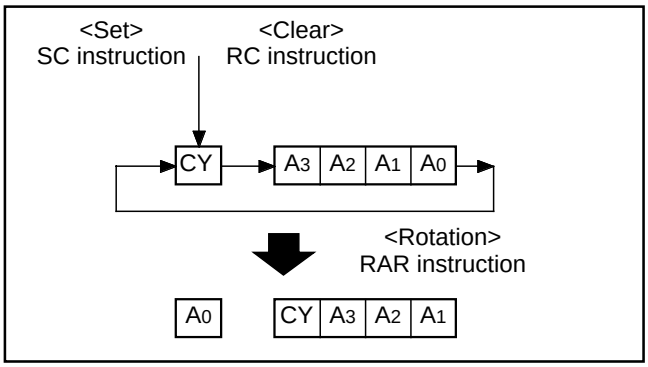


Fig. 2 RAR instruction execution example

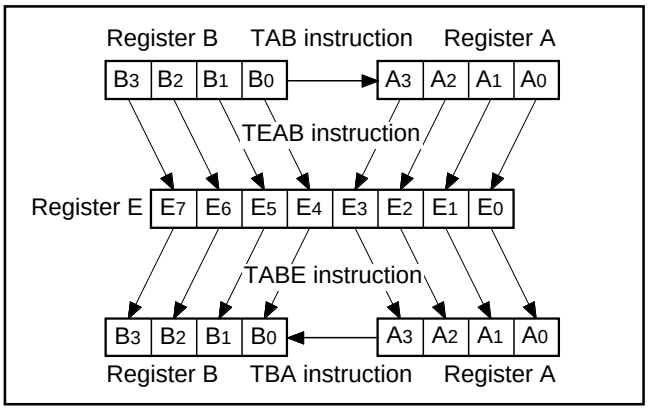


Fig. 3 Registers A, B and register E

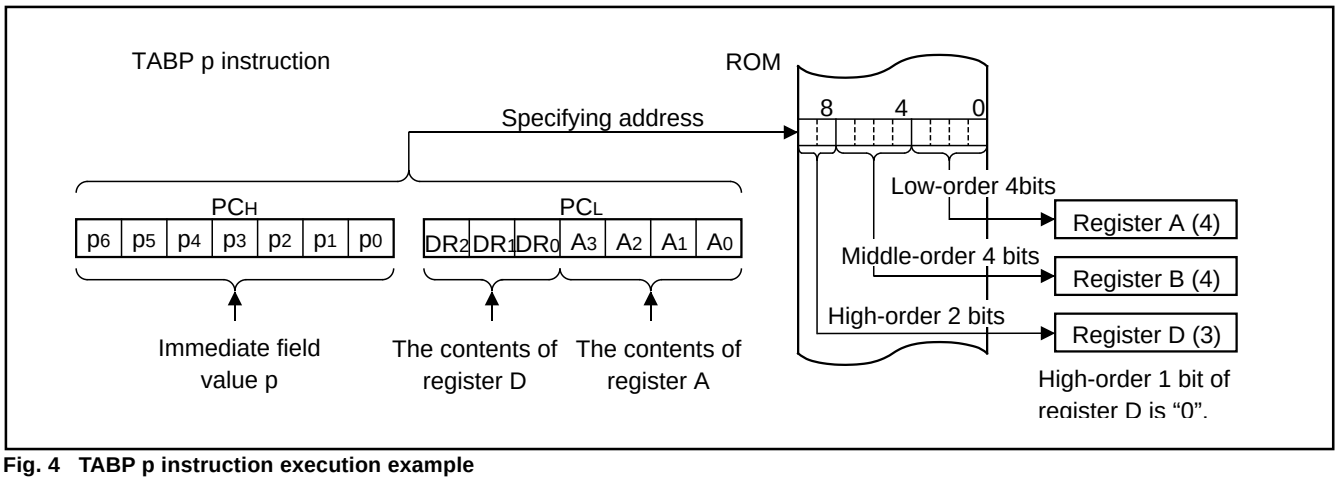


Fig. 4 TABP p instruction execution example

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

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

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

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

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

Figure 5 shows the stack registers (SKs) structure.

Figure 6 shows the example of operation at subroutine call.

(6) Interrupt stack register (SDP)

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

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

(7) Skip flag

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

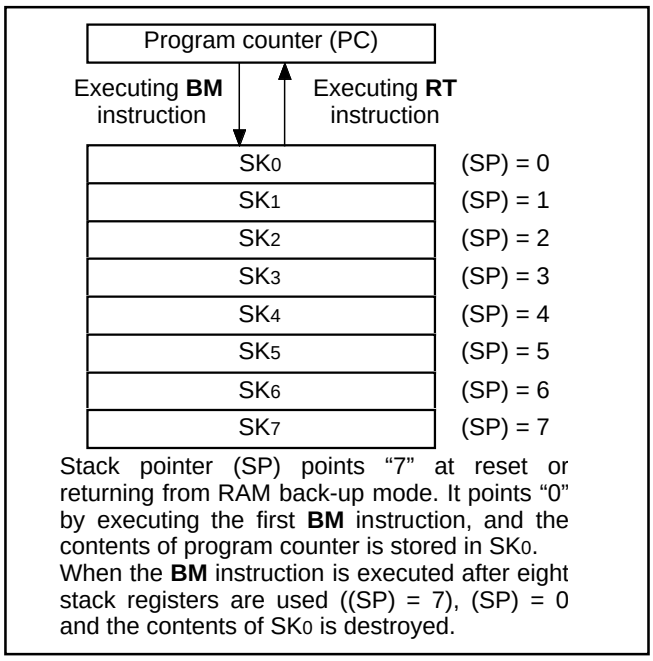


Fig. 5 Stack registers (SKs) structure

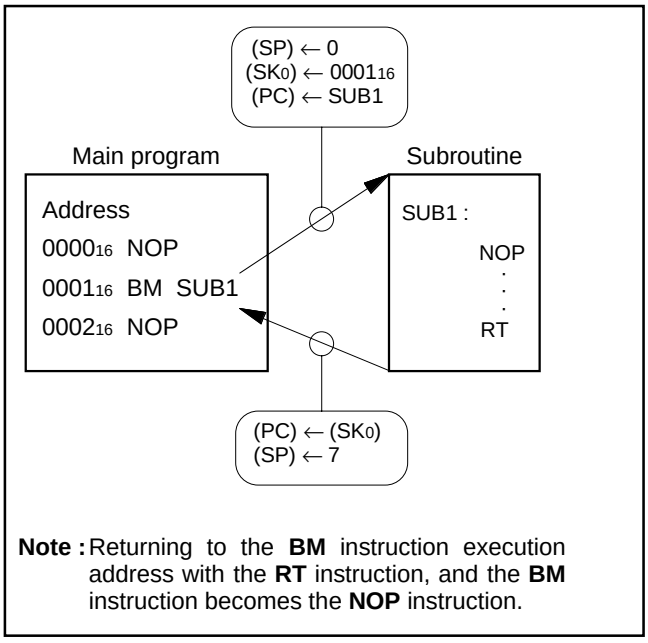


Fig. 6 Example of operation at subroutine call

(8) Program counter (PC)

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

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

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

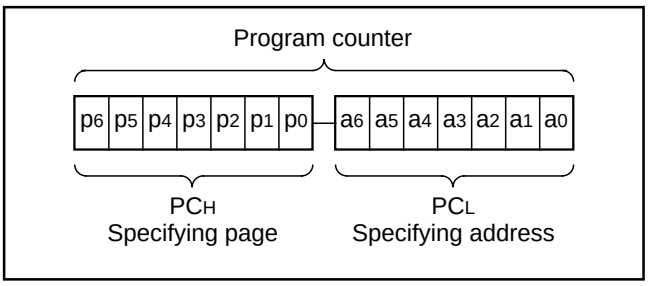


Fig. 7 Program counter (PC) structure

(9) Data pointer (DP)

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

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

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

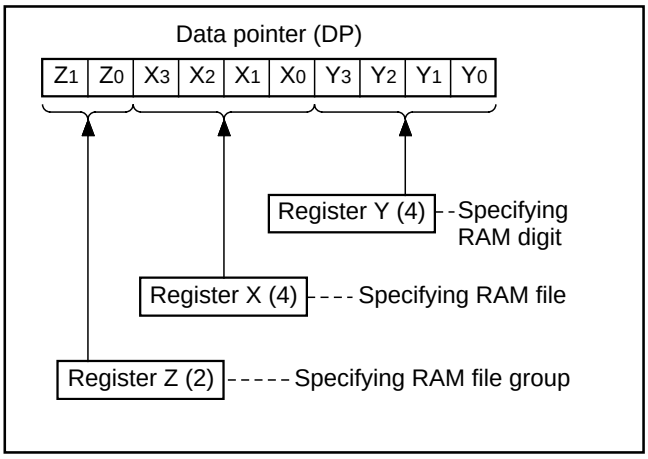


Fig. 8 Data pointer (DP) structure

• Note

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

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

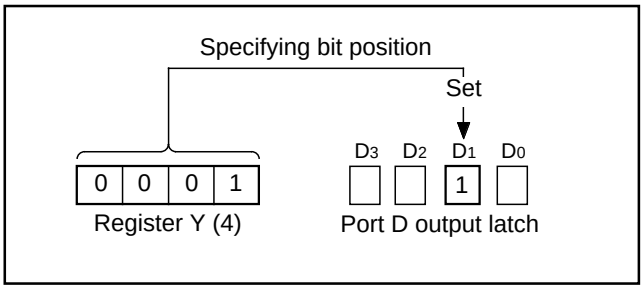


Fig. 9 SD instruction execution example

PROGRAM MEMORY (ROM)

The program memory is a mask ROM. 1 word of ROM is composed of 10 bits. ROM is separated every 128 words by the unit of page (addresses 0 to 127). Table 1 shows the ROM size and pages. Figure 10 shows the ROM map of M34519M8/E8.

Table 1 ROM size and pages

Part number	ROM (PROM) size (× 10 bits)	Pages
M34519M6	6144 words	48 (0 to 47)
M34519M8/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 9 to 0) of all addresses can be used as data areas with the TABP p instruction.

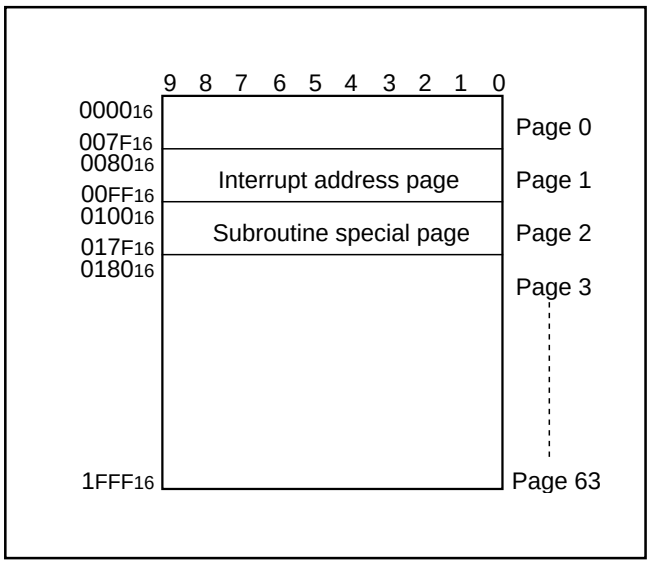


Fig. 10 ROM map of M34519M8/E8

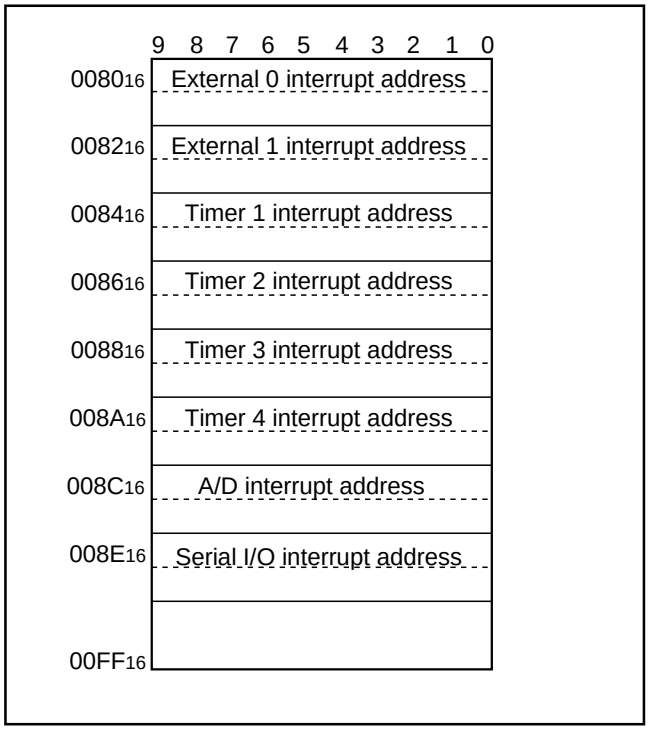


Fig. 11 Page 1 (addresses 0080_{16} to $00FF_{16}$) structure

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 (also, set a value after system returns from RAM back-up). Table 2 shows the RAM size. Figure 12 shows the RAM map.

• Note

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

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

Table 2 RAM size

Part number	RAM size
M34519M6	384 words × 4 bits (1536 bits)
M34519M8/E8	

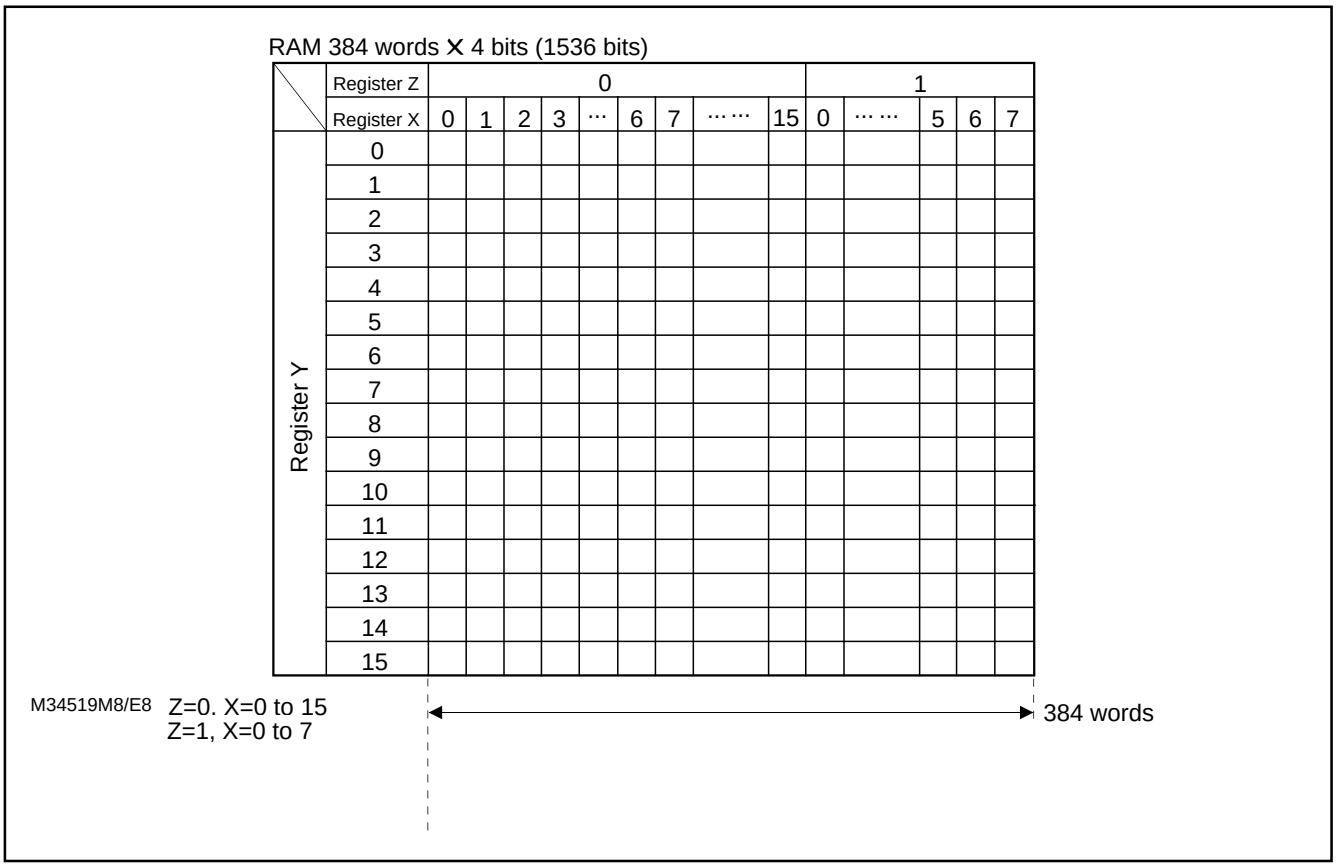


Fig. 12 RAM map

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 transmit/receive	Address E in page 1

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

Interrupt name	Interrupt request flag	Skip instruction	Interrupt enable bit
External 0 interrupt	EXF0	SNZ0	$V1_0$
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).

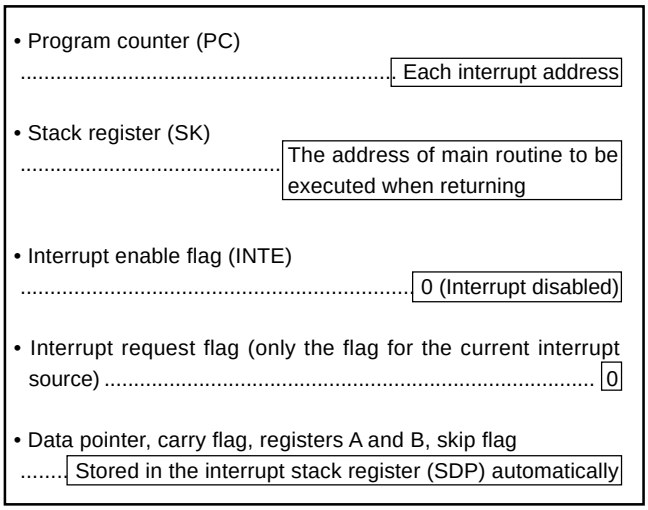


Fig. 14 Internal state when interrupt occurs

(5) Interrupt processing

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

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

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

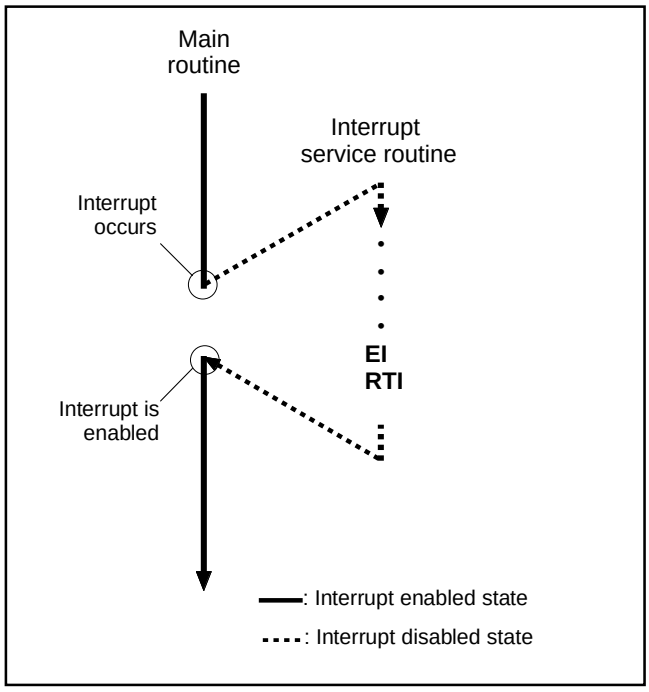


Fig. 13 Program example of interrupt processing

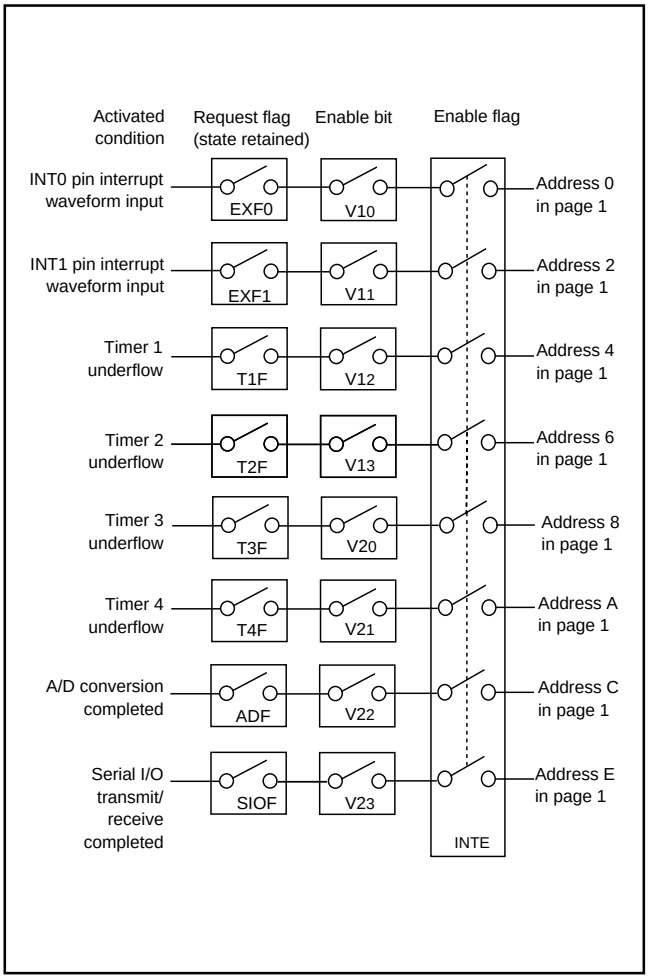


Fig. 15 Interrupt system diagram

(6) Interrupt control registers

- Interrupt control register V1
  Interrupt enable bits of external 0, 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
  The timer 3, timer 4, A/D and serial I/O interrupt enable bit is assigned to register V2. Set the contents of this register through register A with the TV2A instruction. The TAV2 instruction can be used to transfer the contents of register V2 to register A.

Table 6 Interrupt control registers

Interrupt control register V1		at reset : 0000_2		at RAM back-up : 0000_2	R/W TAV1/TV1A
$V1_3$	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 TAV2/TV2A
$V2_3$	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 ($V1_0$–$V1_3$, $V2_0$–$V2_3$), 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).

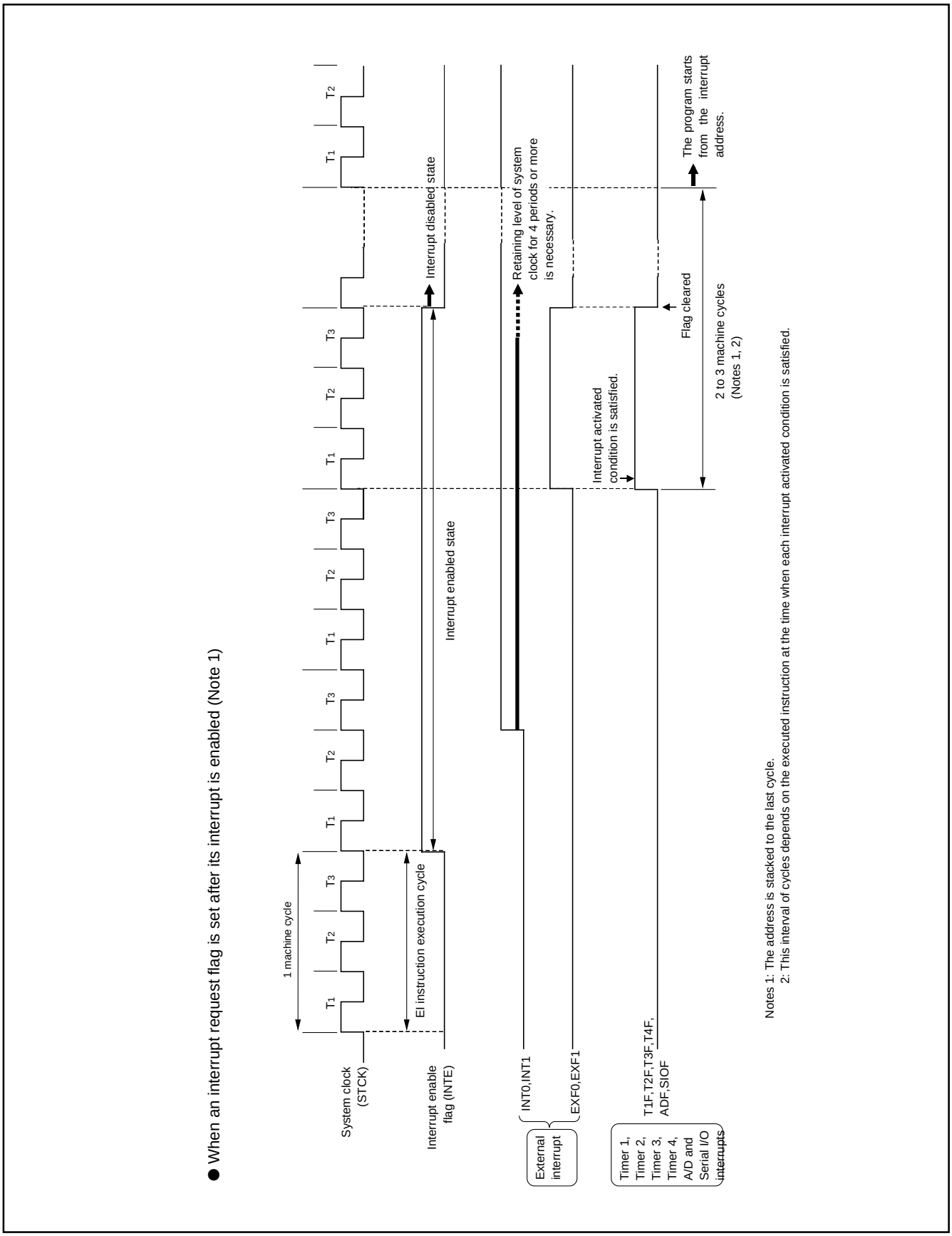


Fig. 16 Interrupt sequence

EXTERNAL INTERRUPTS

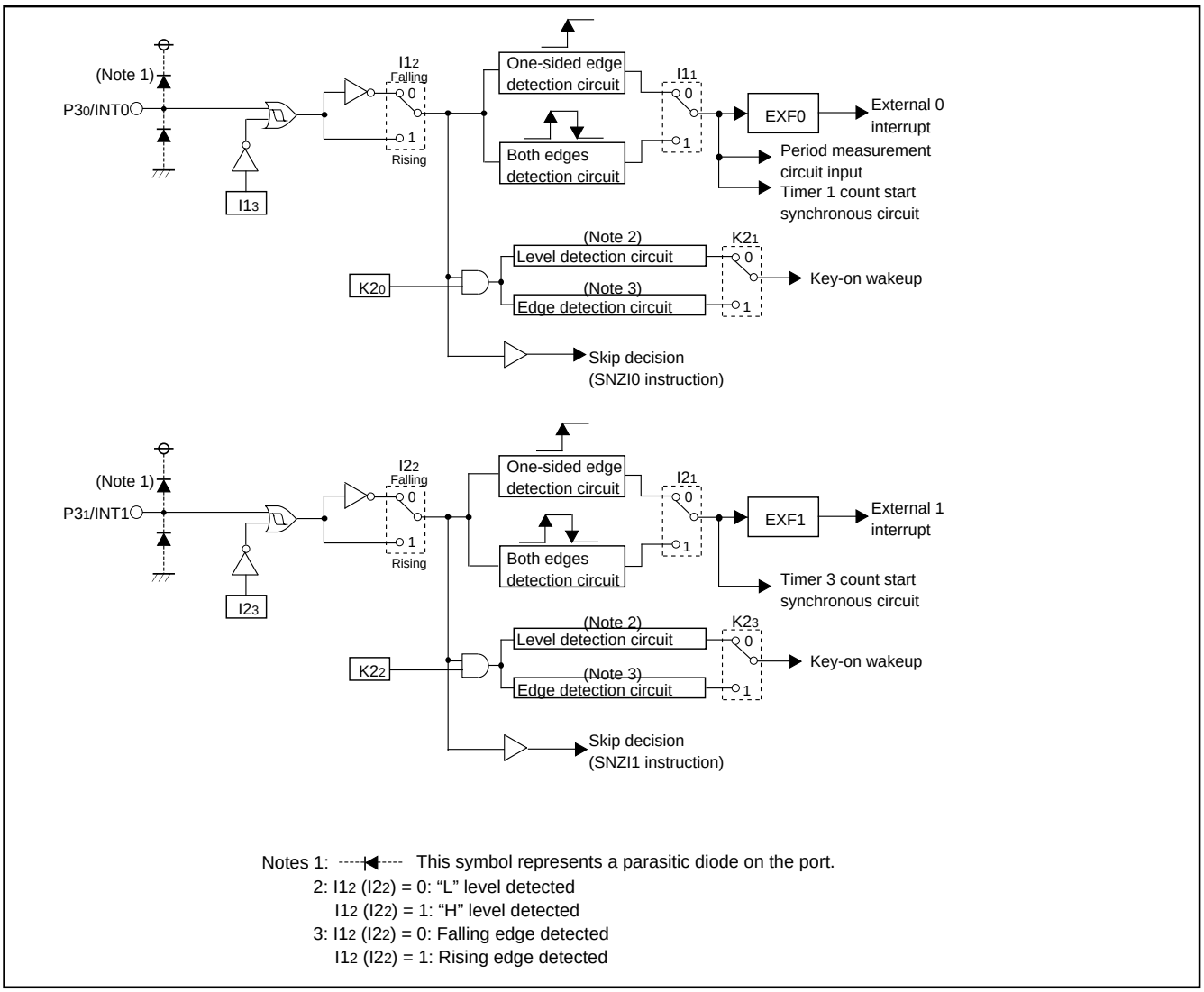
The 4519 Group has the external 0 interrupt and external 1 interrupt.

An external interrupt request occurs when a valid waveform is input to an interrupt input pin (edge detection).

The external interrupt can be controlled with the interrupt control registers I1 and I2.

Table 7 External interrupt activated conditions

Name	Input pin	Activated condition	Valid waveform selection bit
External 0 interrupt	P30/INT0	When the next waveform is input to P30/INT0 pin • Falling waveform (“H”→“L”) • Rising waveform (“L”→“H”) • Both rising and falling waveforms	I11 I12
External 1 interrupt	P31/INT1	When the next waveform is input to P31/INT1 pin • Falling waveform (“H”→“L”) • Rising waveform (“L”→“H”) • Both rising and falling waveforms	I21 I22



Notes 1: ----|◄---- This symbol represents a parasitic diode on the port.
2: I12 (I22) = 0: “L” level detected
I12 (I22) = 1: “H” level detected
3: I12 (I22) = 0: Falling edge detected
I12 (I22) = 1: Rising edge detected

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.

• 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 Set the bit 3 of register I1 to "1" for the INT0 pin to be in the input enabled state.
② Select the valid waveform with the bits 1 and 2 of register I1.
③ Clear the EXF0 flag to "0" with the SNZ0 instruction.
④ Set the NOP instruction for the case when a skip is performed with the SNZ0 instruction.
⑤ Set both the external 0 interrupt enable bit ($V1_0$) and the INTE flag to "1."

The external 0 interrupt is now enabled. Now when a valid waveform is input to the $P3_0$/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.

• 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 Set the bit 3 of register I2 to "1" for the INT1 pin to be in the input enabled state.
② Select the valid waveform with the bits 1 and 2 of register I2.
③ Clear the EXF1 flag to "0" with the SNZ1 instruction.
④ Set the NOP instruction for the case when a skip is performed with the SNZ1 instruction.
⑤ 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 register

Interrupt control register I1		at reset : 0000_2	at RAM back-up : state retained	R/W TAI1/TI1A
$I1_3$	INT0 pin input control bit	0	INT0 pin input disabled	
		1	INT0 pin input enabled	
$I1_2$	Interrupt valid waveform for INT0 pin/ return level selection bit	0	Falling waveform/“L” level (“L” level is recognized with the SNZI0 instruction)	
		1	Rising waveform/“H” level (“H” level is recognized with the SNZI0 instruction)	
$I1_1$	INT0 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I1_0$	INT0 pin Timer 1 count start synchronous circuit selection bit	0	Timer 1 count start synchronous circuit not selected	
		1	Timer 1 count start synchronous circuit selected	

Interrupt control register I2		at reset : 0000_2	at RAM back-up : state retained	R/W TAI2/TI2A
$I2_3$	INT1 pin input control bit (Note 2)	0	INT1 pin input disabled	
		1	INT1 pin input enabled	
$I2_2$	Interrupt valid waveform for INT1 pin/ return level selection bit (Note 2)	0	Falling waveform/“L” level (“L” level is recognized with the SNZI1 instruction)	
		1	Rising waveform/“H” level (“H” level is recognized with the SNZI1 instruction)	
$I2_1$	INT1 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I2_0$	INT1 pin Timer 3 count start synchronous circuit selection bit	0	Timer 3 count start synchronous circuit not selected	
		1	Timer 3 count start synchronous circuit selected	

Notes 1: “R” represents read enabled, and “W” represents write enabled.
2: When the contents of $I1_2$, $I1_3$ $I2_2$ and $I2_3$ are changed, the external interrupt request flag (EXF0, EXF1) may be set.

(4) Notes on External 0 interrupt

① Note [1] on bit 3 of register I1

When the input of the INT0 pin is controlled with the bit 3 of register I1 in software, be careful about the following notes.

• Depending on the input state of the P3$_0$/INT0 pin, the external 0 interrupt request flag (EXF0) may be set when the bit 3 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 18 ①) and then, change the bit 3 of register I1.
In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 18 ②).
Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 18 ③).

```
    ⋮
LA    4        ; (XXX0₂)
TV1A           ; The SNZ0 instruction is valid ..... ①
LA    8        ; (1XXX₂)
TI1A           ; Control of INT0 pin input is changed
NOP            ..... ②
SNZ0           ; The SNZ0 instruction is executed
                 (EXF0 flag cleared)
NOP            ..... ③
    ⋮
   X : these bits are not used here.
```

Fig. 18 External 0 interrupt program example-1

② Note [2] on bit 3 of register I1

When the bit 3 of register I1 is cleared to "0", the RAM back-up mode is selected and the input of INT0 pin is disabled, be careful about the following notes.

• When the input of INT0 pin is disabled (register I1$_3$ = "0"), set the key-on wakeup function to be invalid (register K2$_0$ = "0") before system enters to the RAM back-up mode. (refer to Figure 19①).

```
    ⋮
LA    0        ; (XXX0₂)
TK2A           ; Input of INT0 key-on wakeup invalid .. ①
DI
EPOF
POF            ; RAM back-up
    ⋮
   X : these bits are not used here.
```

Fig. 19 External 0 interrupt program example-2

③ Note on bit 2 of register I1

When the interrupt valid waveform of the P3$_0$/INT0 pin is changed with the bit 2 of register I1 in software, be careful about the following notes.

• Depending on the input state of the P3$_0$/INT0 pin, the external 0 interrupt request flag (EXF0) may be set when the bit 2 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 20①) and then, change the bit 2 of register I1.
In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 20②).
Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 20③).

```
    ⋮
LA    4        ; (XXX0₂)
TV1A           ; The SNZ0 instruction is valid ..... ①
LA    12       ; (X1XX₂)
TI1A           ; Interrupt valid waveform is changed
NOP            ..... ②
SNZ0           ; The SNZ0 instruction is executed
                 (EXF0 flag cleared)
NOP            ..... ③
    ⋮
   X : these bits are not used here.
```

Fig. 20 External 0 interrupt program example-3

(5) Notes on External 1 interrupt

① Note [1] on bit 3 of register I2

When the input of the INT1 pin is controlled with the bit 3 of register I2 in software, be careful about the following notes.

• Depending on the input state of the P3$_1$/INT1 pin, the external 1 interrupt request flag (EXF1) may be set when the bit 3 of register I2 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 1 of register V1 to "0" (refer to Figure 21①) and then, change the bit 3 of register I2.
In addition, execute the SNZ1 instruction to clear the EXF1 flag to "0" after executing at least one instruction (refer to Figure 21②).
Also, set the NOP instruction for the case when a skip is performed with the SNZ1 instruction (refer to Figure 21③).

```
    ⋮
LA    4       ; (XX0X2)
TV1A          ; The SNZ1 instruction is valid ..... ①
LA    8       ; (1XXX2)
TI2A          ; Control of INT1 pin input is changed
NOP           ..... ②
SNZ1          ; The SNZ1 instruction is executed
              (EXF1 flag cleared)
NOP           ..... ③
    ⋮
```

X : these bits are not used here.

Fig. 21 External 1 interrupt program example-1

② Note [2] on bit 3 of register I2

When the bit 3 of register I2 is cleared to "0", the RAM back-up mode is selected and the input of INT1 pin is disabled, be careful about the following notes.

• When the input of INT1 pin is disabled (register I2$_3$ = "0"), set the key-on wakeup function to be invalid (register K2$_2$ = "0") before system enters to the RAM back-up mode. (refer to Figure 22①).

```
    ⋮
LA    0       ; (X0XX2)
TK2A          ; Input of INT1 key-on wakeup invalid .. ①
DI
EPOF
POF           ; RAM back-up
    ⋮
```

X : these bits are not used here.

Fig. 22 External 1 interrupt program example-2

③ Note on bit 2 of register I2

When the interrupt valid waveform of the P3$_1$/INT1 pin is changed with the bit 2 of register I2 in software, be careful about the following notes.

• Depending on the input state of the P3$_1$/INT1 pin, the external 1 interrupt request flag (EXF1) may be set when the bit 2 of register I2 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 1 of register V1 to "0" (refer to Figure 23①) and then, change the bit 2 of register I2.
In addition, execute the SNZ1 instruction to clear the EXF1 flag to "0" after executing at least one instruction (refer to Figure 23②).
Also, set the NOP instruction for the case when a skip is performed with the SNZ1 instruction (refer to Figure 23③).

```
    ⋮
LA    4       ; (XX0X2)
TV1A          ; The SNZ1 instruction is valid ..... ①
LA    12      ; (X1XX2)
TI2A          ; Interrupt valid waveform is changed
NOP           ..... ②
SNZ1          ; The SNZ1 instruction is executed
              (EXF1 flag cleared)
NOP           ..... ③
    ⋮
```

X : these bits are not used here.

Fig. 23 External 1 interrupt program example-3

TIMERS

The 4519 Group has the following timers.

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

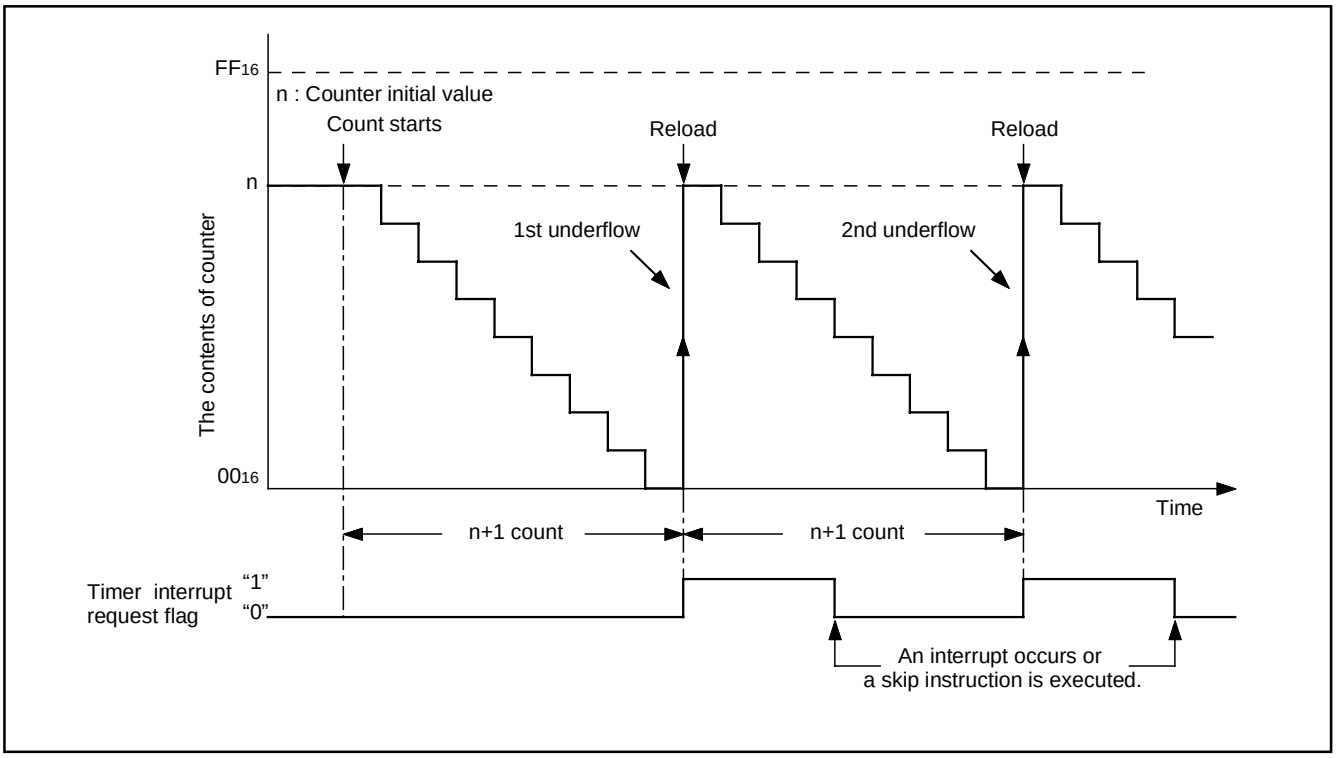


Fig. 24 Auto-reload function

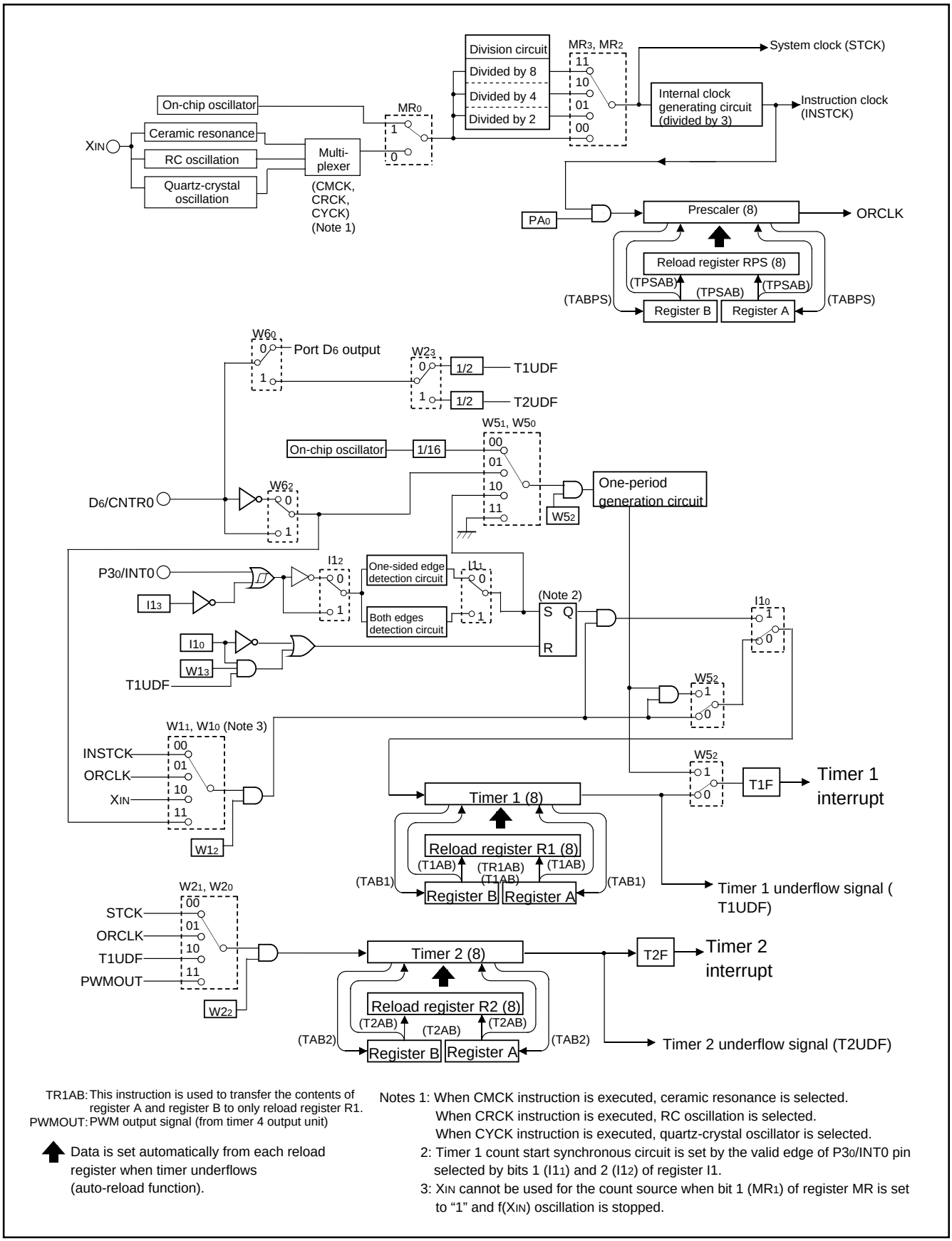
The 4519 Group timer consists of the following circuits.

- Prescaler : 8-bit programmable timer
- Timer 1 : 8-bit programmable timer
- Timer 2 : 8-bit programmable timer
- Timer 3 : 8-bit programmable timer
- Timer 4 : 8-bit programmable timer
- Watchdog timer : 16-bit fixed dividing frequency timer
  (Timers 1, 2, 3, and 4 have the interrupt function, respectively)

Prescaler and timers 1, 2, 3, and 4 can be controlled with the timer control registers PA, W1 to W6. The watchdog timer is a free counter which is not controlled with the control register.
Each function is described below.

Table 9 Function related timers

Circuit	Structure	Count source	Frequency dividing ratio	Use of output signal	Control register
Prescaler	8-bit programmable binary down counter	• Instruction clock (INSTCK)	1 to 256	• Timer 1, 2, 3, amd 4 count sources	PA
Timer 1	8-bit programmable binary down counter (link to INT0 input) (period/pulse width measurement function)	• Instruction clock (INSTCK) • Prescaler output (ORCLK) • XIN input • CNTR0 input	1 to 256	• Timer 2 count source • CNTR0 output • Timer 1 interrupt	W1 W2 W5
Timer 2	8-bit programmable binary down counter	• System clock (STCK) • Prescaler output (ORCLK) • Timer 1 underflow (T1UDF) • PWM output (PWMOUT)	1 to 256	• Timer 3 count source • CNTR0 output • Timer 2 interrupt	W2
Timer 3	8-bit programmable binary down counter (link to INT1 input)	• PWM output (PWMOUT) • Prescaler output (ORCLK) • Timer 2 underflow (T2UDF) • CNTR1 input	1 to 256	• CNTR1 output control • Timer 3 interrupt	W3
Timer 4	8-bit programmable binary down counter (PWM output function)	• XIN input • Prescaler output (ORCLK)	1 to 256	• Timer 2, 3 count source • CNTR1 output • Timer 4 interrupt	W4
Watchdog timer	16-bit fixed dividing frequency	• Instruction clock (INSTCK)	65534	• System reset (count twice) • WDF flag decision	

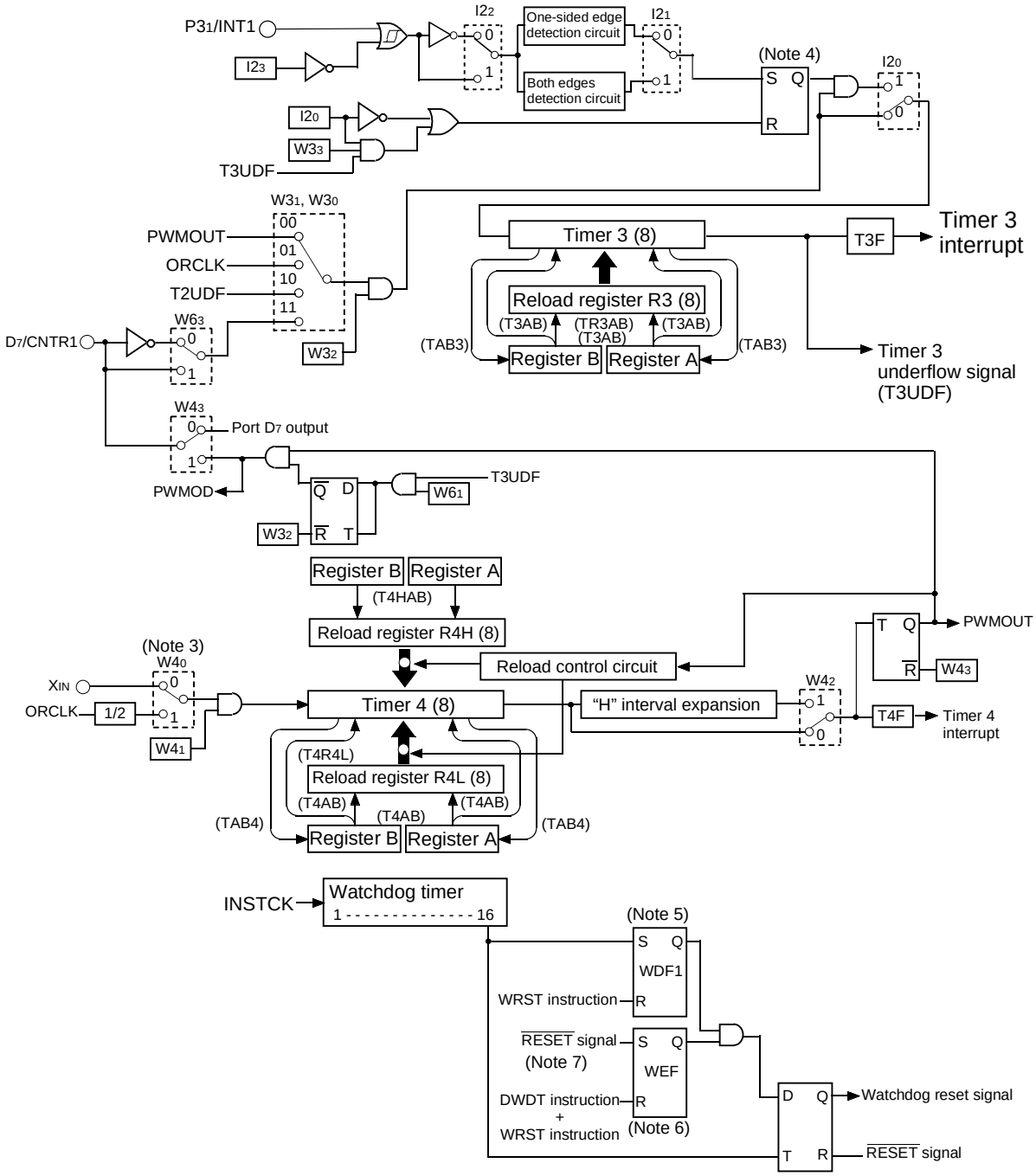


TR1AB: This instruction is used to transfer the contents of register A and register B to only reload register R1.
PWMOUT: PWM output signal (from timer 4 output unit)

Data is set automatically from each reload register when timer underflows (auto-reload function).

Notes 1: When CMCK instruction is executed, ceramic resonance is selected.
When CRCK instruction is executed, RC oscillation is selected.
When CYCK instruction is executed, quartz-crystal oscillator is selected.
2: 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.
3: XIN cannot be used for the count source when bit 1 (MR1) of register MR is set to "1" and f(XIN) oscillation is stopped.

Fig. 25 Timer structure (1)



TR3AB: This instruction is used to transfer the contents of register A and register B to only reload register R3.
T4R4L: This instruction is used to transfer the contents of reload register R4L to timer 4.
INSTCK: Instruction clock (system clock divided by 3)
ORCLK: Prescaler output (instruction clock divided by 1 to 256)

Data is set automatically from each reload register when timer underflows (auto-reload function).

Notes 3: XIN cannot be used for the count source when bit 1 (MR1) of register MR is set to "1" and f(XIN) oscillation is stopped.
4: 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.
5: Flag WDF1 is cleared to "0" and the next instruction is skipped when the WRST instruction is executed while flag WDF1 = "1". The next instruction is not skipped even when the WRST instruction is executed while flag WDF1 = "0".
6: Flag WEF is cleared to "0" and watchdog timer reset does not occur when the DWDT instruction and WRST instruction are executed continuously.
7: The WEF flag is set to "1" at system reset or RAM back-up mode.

Fig. 26 Timer structure (2)

Table 10 Timer related registers

Timer control register PA		at reset : 0_2		at RAM back-up : 0_2	W TPAA
PA_0	Prescaler control bit	0	Stop (state initialized)		
		1	Operating		

Timer control register W1		at reset : 0000_2		at RAM back-up : state retained	R/W TAW1/TW1A
$W1_3$	Timer 1 count auto-stop circuit selection bit (Note 2)	0	Timer 1 count auto-stop circuit not selected		
		1	Timer 1 count auto-stop circuit selected		
$W1_2$	Timer 1 control bit	0	Stop (state retained)		
		1	Operating		
$W1_1$	Timer 1 count source selection bits	$W1_1$	$W1_0$	Count source	
		0	0	Instruction clock (INSTCK)	
$W1_0$		0	1	Prescaler output (ORCLK)	
		1	0	X_{IN} input	
		1	1	CNTR0 input	

Timer control register W2		at reset : 0000_2		at RAM back-up : state retained	R/W TAW2/TW2A
$W2_3$	CNTR0 output signal selection bit (Note 2)	0	Timer 1 underflow signal divided by 2 output		
		1	Timer 2 underflow signal divided by 2 output		
$W2_2$	Timer 2 control bit	0	Stop (state retained)		
		1	Operating		
$W2_1$	Timer 2 count source selection bits	$W2_1$	$W2_0$	Count source	
		0	0	System clock (STCK)	
$W2_0$		0	1	Prescaler output (ORCLK)	
		1	0	Timer 1 underflow signal (T1UDF)	
		1	1	PWM signal (PWMOUT)	

Timer control register W3		at reset : 0000_2		at RAM back-up : state retained	R/W TAW3/TW3A
$W3_3$	Timer 3 count auto-stop circuit selection bit (Note 3)	0	Timer 3 count auto-stop circuit not selected		
		1	Timer 3 count auto-stop circuit selected		
$W3_2$	Timer 3 control bit	0	Stop (state retained)		
		1	Operating		
$W3_1$	Timer 3 count source selection bits	$W3_1$	$W3_0$	Count source	
		0	0	PWM signal (PWMOUT)	
$W3_0$		0	1	Prescaler output (ORCLK)	
		1	0	Timer 2 underflow signal (T2UDF)	
		1	1	CNTR1 input	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: This function is valid only when the timer 1 count start synchronous circuit is selected ($I1_0$="1").
3: This function is valid only when the timer 3 count start synchronous circuit is selected ($I2_0$="1").

Timer control register W4		at reset : 0000_2	at RAM back-up : 0000_2	R/W TAW4/TW4A
$W4_3$	D_7/CNTR1 pin function selection bit	0	D_7 (I/O) / CNTR1 (input)	
		1	CNTR1 (I/O) / D_7 (input)	
$W4_2$	PWM signal “H” interval expansion function control bit	0	PWM signal “H” interval expansion function invalid	
		1	PWM signal “H” interval expansion function valid	
$W4_1$	Timer 4 control bit	0	Stop (state retained)	
		1	Operating	
$W4_0$	Timer 4 count source selection bit	0	X_{IN} input	
		1	Prescaler output (ORCLK) divided by 2	

Timer control register W5		at reset : 0000_2		at RAM back-up : state retained	R/W TAW5/TW5A
$W5_3$	Not used	0		This bit has no function, but read/write is enabled.	
		1			
$W5_2$	Period measurement circuit control bit	0		Stop	
		1		Operating	
$W5_1$	Signal for period measurement selection bits	$W5_1$	$W5_0$	Count source	
		0	0	On-chip oscillator (f(RING/16))	
$W5_0$		0	1	$CNTR_0$ pin input	
		1	0	INT0 pin input	
		1	1	Not available	

Timer control register W6		at reset : 0000_2	at RAM back-up : state retained	R/W TAW6/TW6A
$W6_3$	CNTR1 pin input count edge selection bit	0	Falling edge	
		1	Rising edge	
$W6_2$	CNTR0 pin input count edge selection bit	0	Falling edge	
		1	Rising edge	
$W6_1$	CNTR1 output auto-control circuit selection bit	0	CNTR1 output auto-control circuit not selected	
		1	CNTR1 output auto-control circuit selected	
$W6_0$	D_6/CNTR0 pin function selection 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 PA
  Register PA controls the count operation of prescaler. Set the contents of this register through register A with the TPAA instruction.
- Timer control register W1
  Register W1 controls the selection of timer 1 count auto-stop circuit, and the count operation and count source of timer 1. Set the contents of this register through register A with the TW1A instruction. The TAW1 instruction can be used to transfer the contents of register W1 to register A.
- Timer control register W2
  Register W2 controls the selection of CNTR0 output, and the count operation and count source of timer 2. Set the contents of this register through register A with the TW2A instruction. The TAW2 instruction can be used to transfer the contents of register W2 to register A.
- Timer control register W3
  Register W3 controls the selection of the count operation and count source of timer 3 count auto-stop 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 D_7/CNTR1 output, the expansion of "H" interval of PWM output, and 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 W5
  Register W5 controls the period measurement circuit and target signal for period measurement. Set the contents of this register through register A with the TW5A instruction. The TAW5 instruction can be used to transfer the contents of register W5 to register A.
- Timer control register W6
  Register W6 controls the count edges of CNTR0 pin and CNTR1 pin, selection of CNTR1 output auto-control circuit and the D_6/CNTR0 pin function. Set the contents of this register through register A with the TW6A instruction. The TAW6 instruction can be used to transfer the contents of register W6 to register A..

(2) Prescaler

Prescaler is an 8-bit binary down counter with the prescaler reload register PRS. Data can be set simultaneously in prescaler and the reload register RPS with the TPSAB instruction. Data can be read from reload register RPS with the TABPS instruction.
Stop counting and then execute the TPSAB or TABPS instruction to read or set prescaler data.
Prescaler starts counting after the following process;
① set data in prescaler, and
② set the bit 0 of register PA to "1."
When a value set in reload register RPS is n, prescaler divides the count source signal by n + 1 (n = 0 to 255).
Count source for prescaler is the instruction clock (INSTCK).
Once count is started, when prescaler underflows (the next count pulse is input after the contents of prescaler becomes "0"), new data is loaded from reload register RPS, and count continues (auto-reload function).
The output signal (ORCLK) of prescaler can be used for timer 1, 2, 3, and 4 count sources.

(3) Timer 1 (interrupt function)

Timer 1 is an 8-bit binary down counter with the timer 1 reload register (R1). Data can be set simultaneously in timer 1 and the reload register (R1) with the T1AB instruction. Data can be written to reload register (R1) with the TR1AB instruction. Data can be read from timer 1 with the TAB1 instruction.
Stop counting and then execute the T1AB or TAB1 instruction to read or set timer 1 data.
When executing the TR1AB instruction to set data to reload register R1 while timer 1 is operating, avoid a timing when timer 1 underflows.
Timer 1 starts counting after the following process;
① set data in timer 1
② set count source by bits 0 and 1 of register W1, and
③ set the bit 2 of register W1 to "1."

When a value set in reload register R1 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).
INT0 pin input can be used as the start trigger for timer 1 count operation by setting the bit 0 of register I1 to "1."
Also, in this time, the auto-stop function by timer 1 underflow can be performed by setting the bit 3 of register W1 to "1."
Timer 1 underflow signal divided by 2 can be output from CNTR0 pin by clearing bit 3 of register W2 to "0" and setting bit 0 of register W6 to "1".
The period measurement circuit starts operating by setting bit 2 of register W5 to "1" and timer 1 is used to count the one-period of the target signal for the period measurement. In this time, the timer 1 interrupt request flag (T1F) is not set by the timer 1 underflow signal, it is the flag for detecting the completion of period measurement.

(4) Timer 2 (interrupt function)

Timer 2 is an 8-bit binary down counter with the timer 2 reload register (R2). Data can be set simultaneously in timer 2 and the reload register (R2) with the T2AB instruction. Data can be read from timer 2 with the TAB2 instruction. Stop counting and then execute the T2AB or TAB2 instruction to read or set timer 2 data.
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 2 of register W2 to "1."

When a value set in reload register R2 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).
Timer 2 underflow signal divided by 2 can be output from CNTR0 pin by setting bit 3 of register W2 to "1" and setting bit 0 of register W6 to "1".

(5) 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. Data can be read from timer 3 with the TAB3 instruction.
Stop counting and then execute the T3AB or TAB3 instruction to read or set timer 3 data.
When executing the TR3AB instruction to set data to reload register R3 while timer 3 is operating, avoid a timing when timer 3 underflows.
Timer 3 starts counting after the following process;
① set data in timer 3
② set count source by bits 0 and 1 of register W3, and
③ set the bit 2 of register W3 to "1."

When a value set in reload register R3 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).
INT1 pin input can be used as the start trigger for timer 3 count operation by setting the bit 0 of register I2 to "1."
Also, in this time, the auto-stop function by timer 3 underflow can be performed by setting the bit 3 of register W3 to "1."

(6) Timer 4 (interrupt function)

Timer 4 is an 8-bit binary down counter with two timer 4 reload registers (R4L, R4H). Data can be set simultaneously in timer 4 and the reload register R4L with the T4AB instruction. Data can be set in the reload register R4H with the T4HAB instruction. The contents of reload register R4L set with the T4AB instruction can be set to timer 4 again with the T4R4L instruction. Data can be read from timer 4 with the TAB4 instruction.
Stop counting and then execute the T4AB or TAB4 instruction to read or set timer 4 data.
When executing the T4HAB instruction to set data to reload register R4H while timer 4 is operating, avoid a timing when timer 4 underflows.
Timer 4 starts counting after the following process;
① set data in timer 4
② set count source by bit 0 of register W4, and
③ set the bit 1 of register W4 to "1."

When a value set in reload register R4L 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 R4L, and count continues (auto-reload function).
The PWM signal generated by timer 4 can be output from CNTR1 pin by setting bit 3 of the timer control register W4 to "1".
Timer 4 can control the PWM output to CNTR1 pin with timer 3 by setting bit 1 of the timer control register W6 to "1".

(7) Period measurement function (Timer 1, period measurement circuit)

Timer 1 has the period measurement circuit which performs timer count operation synchronizing with the one cycle of the signal divided by 16 of the on-chip oscillator, D6/CNTR0 pin input, or P30/INT0 pin input (one cycle, "H", or "L" pulse width at the case of a P30/INT0 pin input).

When the target signal for period measurement is set by bits 0 and 1 of register W5, a period measurement circuit is started by setting the bit 2 of register W5 to "1".

Then, if a XIN input is set as the count source of a timer 1 and the bit 2 of register W1 is set to "1", timer 1 starts operation.

Timer 1 starts operation synchronizing with the falling edge of the target signal for period measurement, and stops count operation synchronizing with the next falling edge (one-period generation circuit).

When selecting D6/CNTR0 pin input as target signal for period measurement, the period measurement synchronous edge can be changed into a rising edge by setting the bit 2 of register W6 to "1".

When selecting P30/INT0 pin input as target signal for period measurement, period measurement synchronous edge can be changed into a rising edge by setting the bit 2 of register I1 to "1".

A timer 1 interrupt request flag (T1F) is set to "1" after completing measurement operation.

When a period measurement circuit is set to be operating, timer 1 interrupt request flag (T1F) is not set by timer 1 underflow signal, but turns into a flag which detects the completion of period measurement.

In addition, a timer 1 underflow signal can be used as timer 2 count source.

Once period measurement operation is completed, even if period measurement valid edge is input next, timer 1 is in a stop state and measurement data is held.

When a period measurement circuit is used again, stop a period measurement circuit at once by setting the bit 2 of register W5 to "0", and change a period measurement circuit into a state of operation by setting the bit 2 of register W5 to "1" again.

When a period measurement circuit is used, clear bit 0 of register I1 to "0", and set a timer 1 count start synchronous circuit to be "not selected".

Start timer operation immediately after operation of a period measurement circuit is started.

When the target edge for measurement is input until timer operation is started from the operation of period measurement circuit is started, the count operation is not executed until the timer operation becomes valid. Accordingly, be careful of count data.

When data is read from timer, stop the timer and clear bit 2 of register W5 to "0" to stop the period measurement circuit, and then execute the data read instruction.

Depending on the state of timer 1, the timer 1 interrupt request flag (T1F) may be set to "1" when the period measurement circuit is stopped by clearing bit 2 of register W5 to "0". In order to avoid the occurrence of an unexpected interrupt, clear the bit 2 of register V1 to "0" (refer to Figure 27①) and then, stop the bit 2 of register W5 to "0" to stop the period measurement circuit.

In addition, execute the SNZT1 instruction to clear the T1F flag after executing at least one instruction (refer to Figure 27②).

Also, set the NOP instruction for the case when a skip is performed with the SNZT1 instruction (refer to Figure 27③).

```
    ⋮
LA    0      ; (X0XX2)
TV1A         ; The SNZT1 instruction is valid .....①
LA    0      ; (X0XX2)
TW5A         ; Period measurement circuit stop
NOP          ..... ②
SNZT1        ; The SNZT1 instruction is executed
               (T1F flag cleared)
NOP          ..... ③
    ⋮
```

X : these bits are not used here.

Fig. 27 Period measurement circuit program example

When a period measurement circuit is used, select the sufficiently higher-speed frequency than the signal for measurement for the count source of a timer 1.

When the target signal for period measurement is D6/CNTR0 pin input, do not select D6/CNTR0 pin input as timer 1 count source. (The XIN input is recommended as timer 1 count source at the time of period measurement circuit use.)

(8) Pulse width measurement function (timer 1, period measurement circuit)

A period measurement circuit can measure "H" pulse width (from rising to falling) or "L" pulse width (from falling to rising) of P30/INT0 pin input (pulse width measurement function) when the following is set;

• Set the bit 0 of register W5 to "0", and set a bit 1 to "1" (target for period measurement circuit: 30/INT0 pin input).

• Set the bit 1 of register I1 to "1" (INT0 pin edge detection circuit: both edges detection)

The measurement pulse width ("H" or "L") is decided by the period measurement circuit and the P30/INT0 pin input level at the start time of timer operation.

At the time of the start of a period measurement circuit and timer operation, "L" pulse width (from falling to rising) when the input level of P30/INT0 pin is "H" or "H" pulse width (from rising to falling) when its level is "L" is measured.

When the input of P30/INT0 pin is selected as the target for measurement, set the bit 3 of register I1 to "1", and set the input of INT0 pin to be enabled.

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

Timer 1 and timer 3 have the count start synchronous circuit which synchronizes the input of INT0 pin and INT1 pin, and can start the timer count operation.

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

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

When timer 1 or timer 3 count start synchronous circuit is used, the count start synchronous circuit is set, the count source is input to each timer by inputting valid waveform to INT0 pin or INT1 pin.

The valid waveform of INT0 pin or INT1 pin to set the count start synchronous circuit is the same as the external interrupt activated condition.

Once set, the count start synchronous circuit is cleared by clearing the bit $I1_0$ or $I2_0$ to "0" or reset.

However, when the count auto-stop circuit is selected, the count start synchronous circuit is cleared (auto-stop) at the timer 1 or timer 3 underflow.

(10) Count auto-stop circuit (timer 1, timer 3)

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

The count auto-stop cicuit is valid by setting the bit 3 of register W1 to "1". It is cleared by the timer 1 underflow and the count source to timer 1 is stopped.

This function is valid only when the timer 1 count start synchronous circuit is selected.

Timer 3 has the count auto-stop circuit which is used to stop timer 3 automatically by the timer 3 underflow when the count start synchronous circuit is used.

The count auto-stop cicuit is valid by setting the bit 3 of register W3 to "1". It is cleared by the timer 3 underflow and the count source to timer 3 is stopped.

This function is valid only when the timer 3 count start synchronous circuit is selected.

(11) Timer input/output pin (D_6/CNTR0 pin, D_7/CNTR1 pin)

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

CNTR1 pin is used to input the timer 3 count source and output the PWM signal generated by timer 4.

The D_6/CNTR0 pin function can be selected by bit 0 of register W6.

The selection of D_7/CNTR1 output signal can be controlled by bit 3 of register W4.

When the CNTR0 input is selected for timer 1 count source, timer 1 counts the rising or falling waveform of CNTR0 input. The count edge is selected by the bit 2 of register W6.

When the CNTR1 input is selected for timer 3 count source, timer 3 counts the rising or falling waveform of CNTR1 input. The count edge is selected by the bit 3 of register W6.

(12) PWM output function (D_7/CNTR1, timer 3, timer 4)

When bit 3 of register W4 is set to "1", timer 4 reloads data from reload register R4L and R4H alternately each underflow.

Timer 4 generates the PWM signal (PWMOUT) of the "L" interval set as reload register R4L, and the "H" interval set as reload register R4H. The PWM signal (PWMOUT) is output from CNTR1 pin.

When bit 2 of register W4 is set to "1" at this time, the interval (PWM signal "H" interval) set to reload register R4H for the counter of timer 4 is extended for a half period of count source.

In this case, when a value set in reload register R4H is n, timer 4 divides the count source signal by n + 1.5 (n = 1 to 255).

When this function is used, set "1" or more to reload register R4H.

When bit 1 of register W6 is set to "1", the PWM signal output to CNTR1 pin is switched to valid/invalid each timer 3 underflow. However, when timer 3 is stopped (bit 2 of register W3 is cleared to "0"), this function is canceled.

Even when bit 1 of a register W4 is cleared to "0" in the "H" interval of PWM signal, timer 4 does not stop until it next timer 4 underflow. At CNTR1 output vaild, if a timing of timer 4 underflow overlaps with a timing to stop timer 4, a hazard may be generated in a CNTR1 output waveform. Please review sufficiently.

(13) Timer interrupt request flags (T1F, T2F, T3F, 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, SNZT4).

Use the interrupt control register V1, V2 to select an interrupt or a skip instruction.

An interrupt request flag is cleared to “0” when an interrupt occurs or when the next instruction is skipped with a skip instruction. The timer 1 interrupt request flag (T1F) is not set by the timer 1 underflow signal, it is the flag for detecting the completion of period measurement.

(14) Precautions

Note the following for the use of timers.

- Prescaler
  Stop counting and then execute the TABPS instruction to read from prescaler data.
  Stop counting and then execute the TPSAB instruction to set prescaler data.

- Timer count source
  Stop timer 1, 2, 3 and 4 counting to change its count source.

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

- Writing to the timer
  Stop timer 1, 2, 3 or 4 counting and then execute the data write instruction (T1AB, T2AB, T3AB, T4AB) to write its data.

- Writing to reload register R1, R3, R4H
  When writing data to reload register R1, reload register R3 or reload regiser R4H while timer 1, timer 3 or timer 4 is operating, avoid a timing when timer 1, timer 3 or timer 4 underflows.

- Timer 4
  At CNTR1 output vaild, if a timing of timer 4 underflow overlaps with a timing to stop timer 4, a hazard may be generated in a CNTR1 output waveform. Please review sufficiently.
  When “H” interval extension function of the PWM signal is set to be “valid”, set “1” or more to reload register R4H.

- Period measurement function
  When a period measurement circuit is used, clear bit 0 of register I1 to “0”, and set a timer 1 count start synchronous circuit to be “not selected”.
  Start timer operation immediately after operation of a period measurement circuit is started.
  When the target edge for measurement is input until timer operation is started from the operation of period measurement circuit is started, the count operation is not executed until the timer operation becomes valid. Accordingly, be careful of count data.
  When data is read from timer, stop the timer and clear bit 2 of register W5 to “0” to stop the period measurement circuit, and then execute the data read instruction.
  Depending on the state of timer 1, the timer 1 interrupt request flag (T1F) may be set to “1” when the period measurement circuit is stopped by clearing bit 2 of register W5 to “0”. In order to avoid the occurrence of an unexpected interrupt, clear the bit 2 of register V1 to “0” (refer to Figure 28①) and then, stop the bit 2 of register W5 to “0” to stop the period measurement circuit.
  In addition, execute the SNZT1 instruction to clear the T1F flag after executing at least one instruction (refer to Figure 28②).
  Also, set the NOP instruction for the case when a skip is performed with the SNZT1 instruction (refer to Figure 28③).

```
    ⋮
LA    0       ; (X0XX2)
TV1A          ; The SNZT1 instruction is valid .....①
LA    0       ; (X0XX2)
TW5A          ; Period measurement circuit stop
NOP           ..... ②
SNZT1         ; The SNZT1 instruction is executed
                (T1F flag cleared)
NOP           ..... ③
    ⋮
```

X : these bits are not used here.

Fig. 28 Period measurement circuit program example

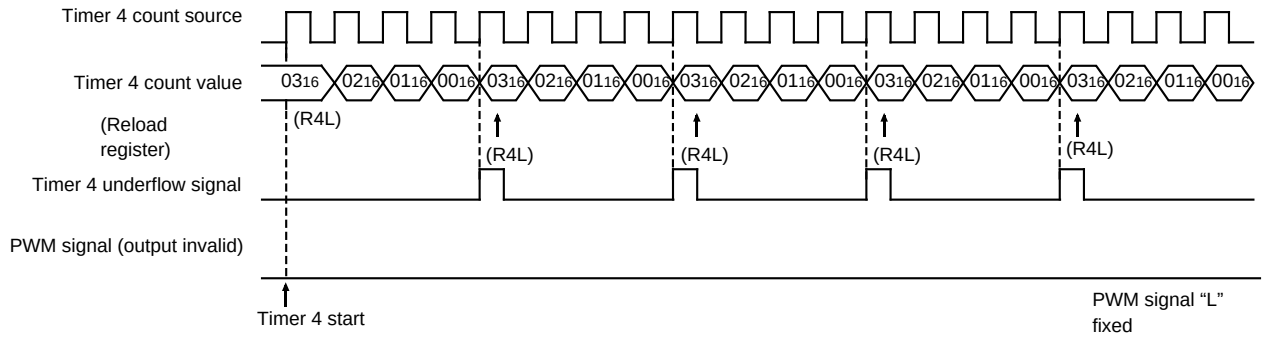
While a period measurement circuit is operating, the timer 1 interrupt request flag (T1F) is not set by the timer 1 underflow signal, it is the flag for detecting the completion of period measurement.

When a period measurement circuit is used, select the sufficiently higher-speed frequency than the signal for measurement for the count source of a timer 1.

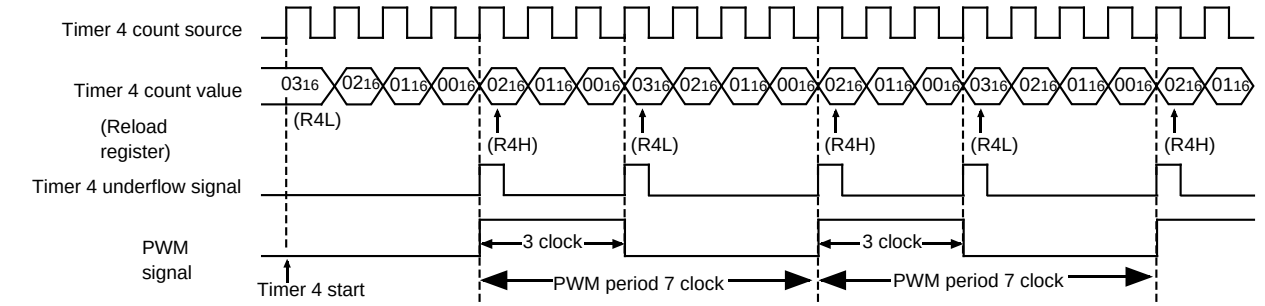
When the target signal for period measurement is D_6/CNTR0 pin input, do not select D_6/CNTR0 pin input as timer 1 count source. (The X_{IN} input is recommended as timer 1 count source at the time of period measurement circuit use.)

When the input of $P3_0$/INT0 pin is selected for measurement, set the bit 3 of a register I1 to “1”, and set the input of INT0 pin to be enabled.

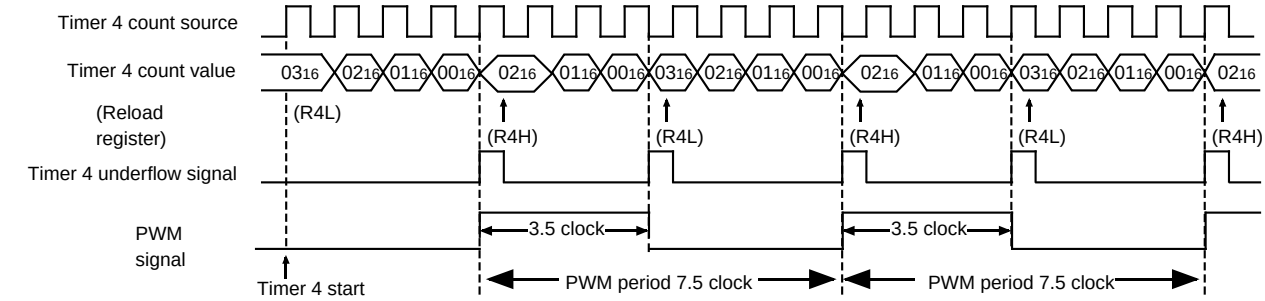
● CNTR1 output: invalid (W4$_3$ = "0")



● CNTR1 output: valid (W4$_3$ = "1")
PWM signal "H" interval extension function: invalid (W4$_2$ = "0")



● CNTR1 output: valid (W4$_3$ = "1")
PWM signal "H" interval extension function: valid (W4$_2$ = "1") (Note)

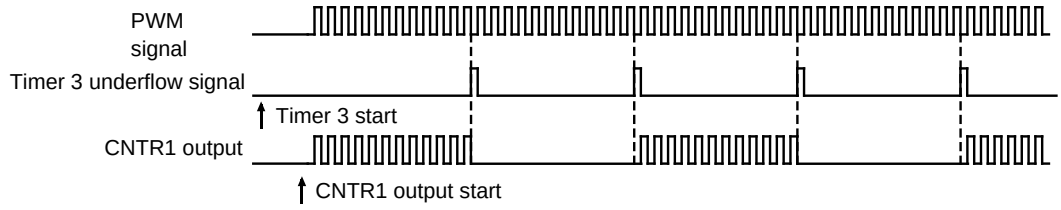


Note: At PWM signal "H" interval extension function: valid, set "01$_{16}$" or more to reload register R4H.

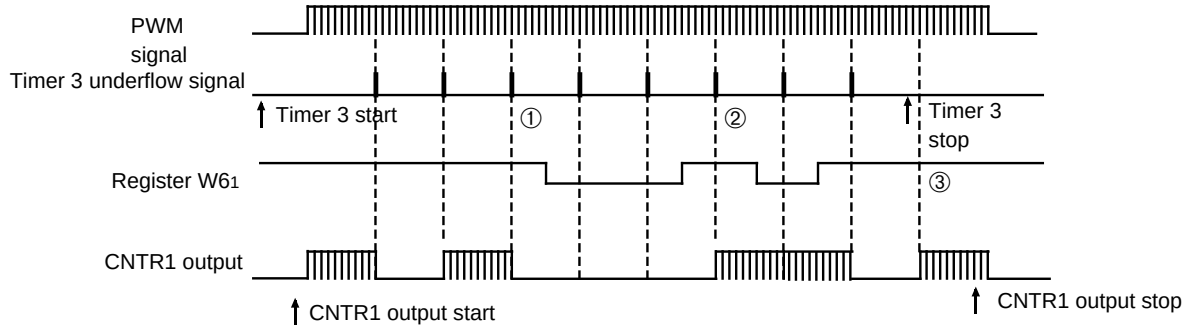
Fig. 29 Timer 4 operation (reload register R4L: "03$_{16}$", R4H: "02$_{16}$")

CNTR1 output auto-control circuit by timer 3 is selected.

● CNTR1 output: valid ($W4_3$ = "1")
CNTR1 output auto-control circuit selected ($W6_1$ = "1")



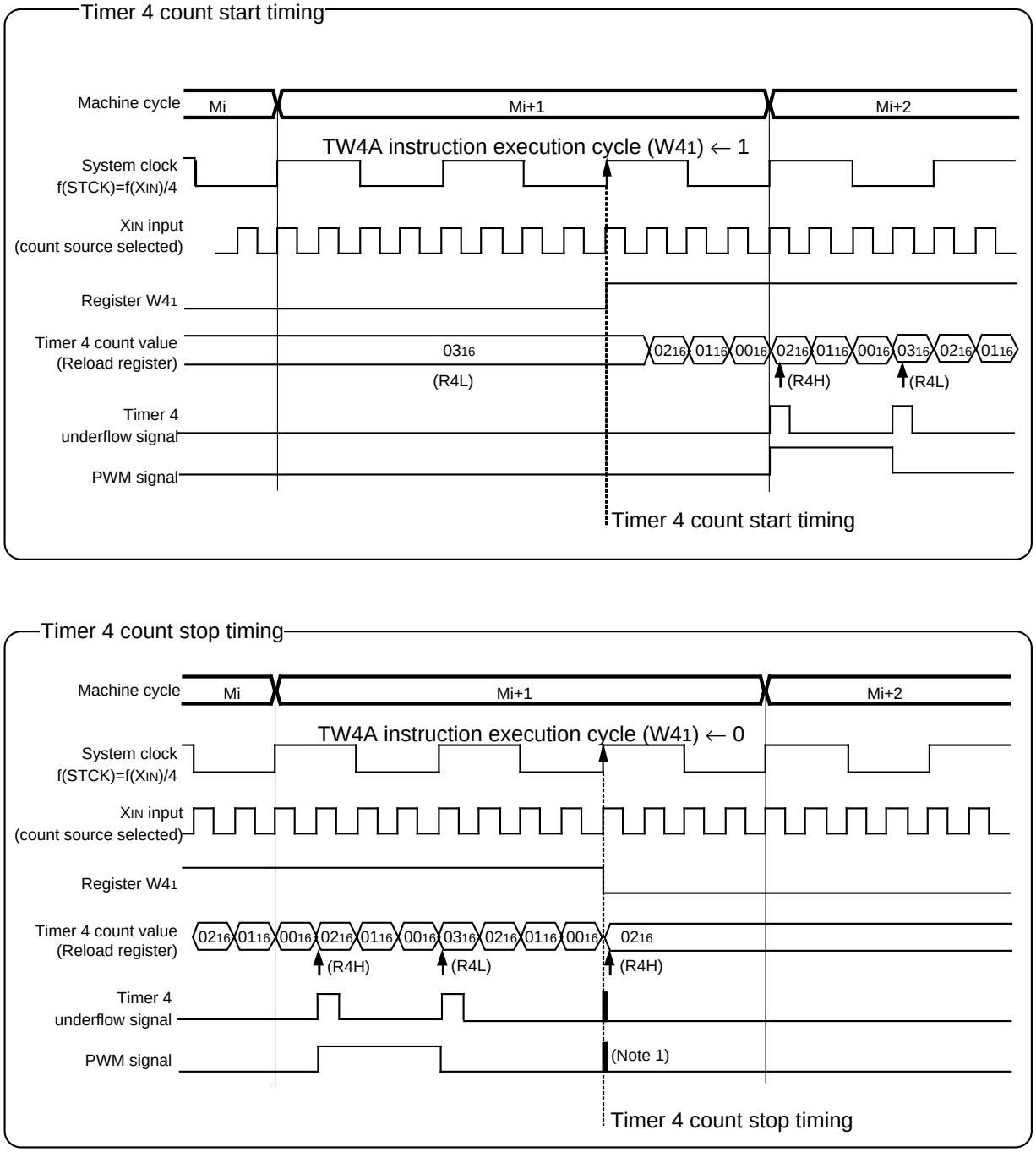
● CNTR1 output auto-control function



① When the CNTR1 output auto-control function is set to be invalid while the CNTR1 output is invalid, the CNTR1 output invalid state is retained.
② When the CNTR1 output auto-control function is set to be invalid while the CNTR1 output is valid, the CNTR1 output valid state is retained.
③ When timer 3 is stopped, the CNTR1 output auto-control function becomes invalid.

Fig. 30 CNTR1 output auto-control function by timer 3

●Waveform extension function of CNTR1 output “H” interval: Invalid (W42 = “0”),
CNTR1 output: valid (W43 = “1”),
Count source: XIN input selected (W40 = “0”),
Reload register R4L: “0316”
Reload register R4H: “0216”



Notes 1: At CNTR1 output vaild, if a timing of timer 4 underflow overlaps with a timing to stop timer 4, a hazard may be generated in a CNTR1 output waveform. Please review sufficiently.
2: At CNTR1 output valid, timer 4 stops after “H” interval of PWM signal set by reload register R4H is output.

Fig. 31 Timer 4 count start/stop timing

WATCHDOG TIMER

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

The timer WDT downcounts the instruction clocks as the count source from "$FFFF_{16}$" after system is released from reset.

After the count is started, when the timer WDT underflow occurs (after the count value of timer WDT reaches "0000_{16}," the next count pulse is input), the WDF1 flag is set to "1."

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

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

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

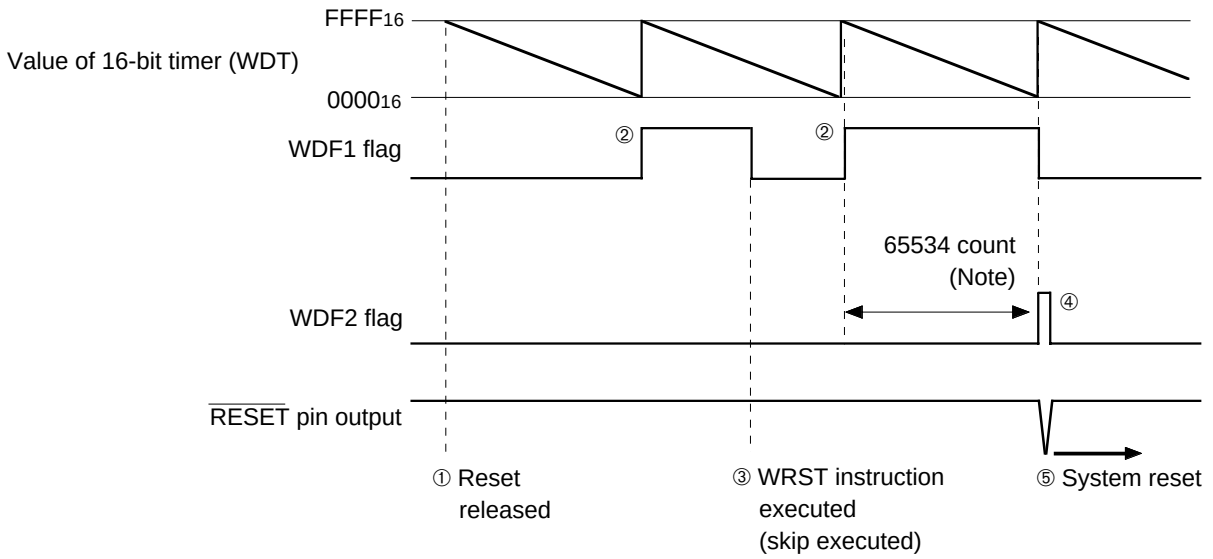
When the DWDT instruction and the WRST instruction are executed continuously, the WEF flag is cleared to "0" and the watchdog timer function is invalid.

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

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

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

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



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

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

Fig. 32 Watchdog timer function

When the watchdog timer is used, clear the WDF1 flag at the period of 65534 machine cycles or less with the WRST instruction.
When the watchdog timer is not used, execute the DWDT instruction and the WRST instruction continuously (refer to Figure 33).
The watchdog timer is not stopped with only the DWDT instruction.
The contents of WDF1 flag and timer WDT are initialized at the RAM back-up mode.
When using the watchdog timer and the RAM back-up mode, initialize the WDF1 flag with the WRST instruction just before the microcomputer enters the RAM back-up state (refer to Figure 34).
The watchdog timer function is valid after system is returned from the RAM back-up. When not using the watchdog timer function, execute the DWDT instruction and the WRST instruction continuously every system is returned from the RAM back-up, and stop the watchdog timer function.

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

Fig. 33 Program example to start/stop watchdog timer

```
    ⋮
WRST        ; WDF1 flag cleared
NOP
DI          ; Interrupt disabled
EPOF        ; POF instruction enabled
POF
  ↓
Oscillation stop
    ⋮
```

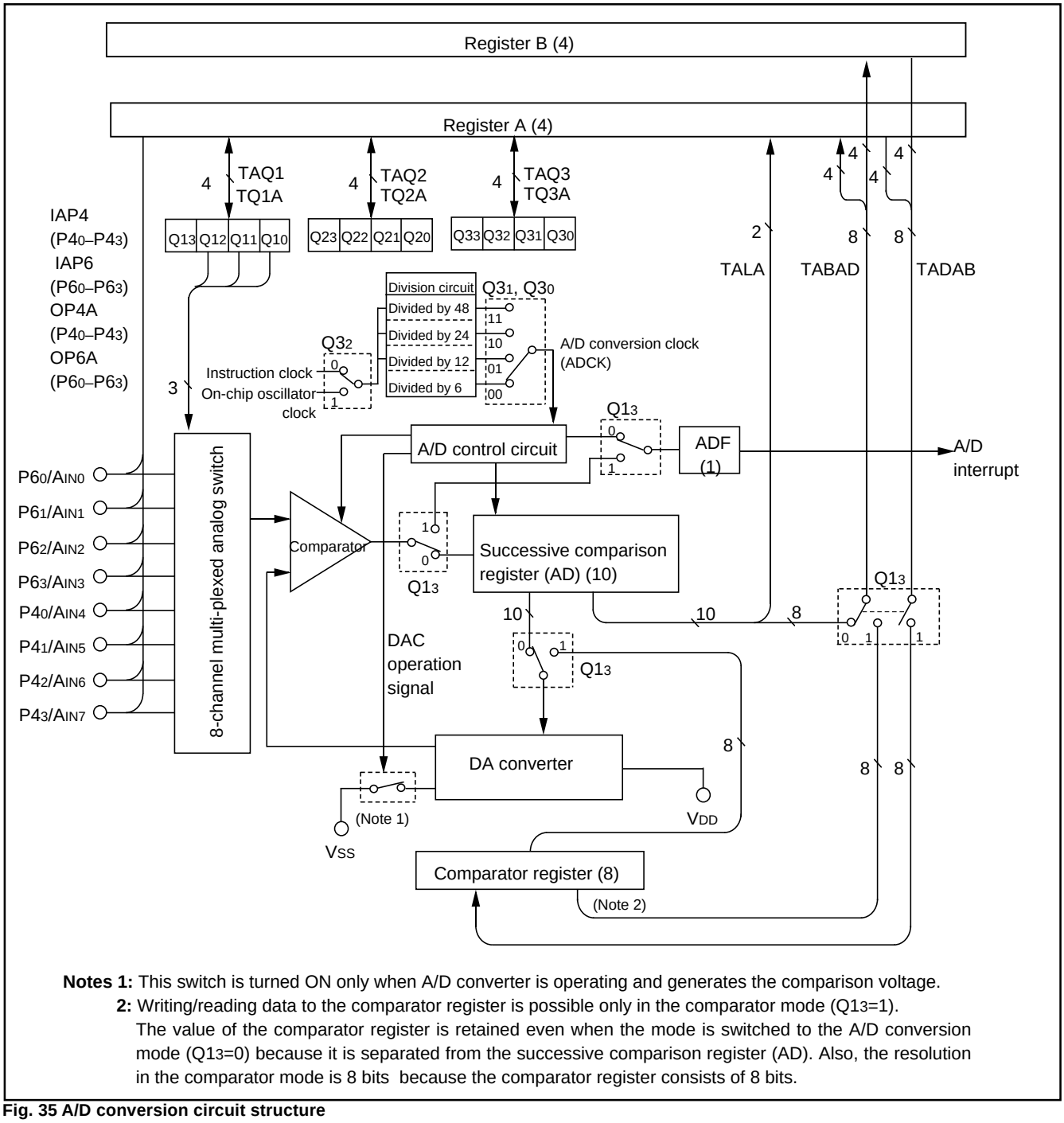
Fig. 34 Program example to enter the mode when using the watchdog timer

A/D CONVERTER (Comparator)

The 4519 Group has a built-in A/D conversion circuit that performs conversion by 10-bit successive comparison method. Table 11 shows the characteristics of this A/D converter. This A/D converter can also be used as an 8-bit comparator to compare analog voltages input from the analog input pin with preset values.

Table 11 A/D converter characteristics

Parameter	Characteristics
Conversion format	Successive comparison method
Resolution	10 bits
Relative accuracy	Linearity error: ±2LSB (2.7 V ≤ V_{DD} ≤ 5.5V)
	Differential non-linearity error: ±0.9LSB (2.2 V ≤ V_{DD} ≤ 5.5V)
Conversion speed	31 μs (f(X_{IN}) = 6 MHz, STCK = f(X_{IN}) (X_{IN} through-mode), ADCK = INSTCK/6)
Analog input pin	8



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

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

Fig. 35 A/D conversion circuit structure

Table 12 A/D control registers

A/D control register Q1		at reset : 0000_2			at RAM back-up : state retained	R/W TAQ1/TQ1A
$Q1_3$	A/D operation mode selection bit	A/D conversion mode				
		Comparator mode				
$Q1_2$	Analog input pin selection bits	$Q1_2$	$Q1_1$	$Q1_0$	Analog input 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}	
$Q1_0$		1	0	1	A_{IN5}	
		1	1	0	A_{IN6}	
		1	1	1	A_{IN7}	

A/D control register Q2		at reset : 0000_2		at RAM back-up : state retained	R/W TAQ2/TQ2A
$Q2_3$	$P4_0/A_{IN4}$, $P4_1/A_{IN5}$, $P4_2/A_{IN6}$, $P4_3/A_{IN7}$ pin function selection bit	0	$P4_0$, $P4_1$, $P4_2$, $P4_3$		
		1	A_{IN4}, A_{IN5}, A_{IN6}, A_{IN7}		
$Q2_2$	$P6_2/A_{IN2}$, $P6_3/A_{IN3}$ pin function selection bit	0	$P6_2$, $P6_3$		
		1	A_{IN2}, A_{IN3}		
$Q2_1$	$P6_1/A_{IN1}$ pin function selection bit	0	$P6_1$		
		1	A_{IN1}		
$Q2_0$	$P6_0/A_{IN0}$ pin function selection bit	0	$P6_0$		
		1	A_{IN0}		

A/D control register Q3		at reset : 0000_2		at RAM back-up : state retained	R/W TAQ3/TQ3A
$Q3_3$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$Q3_2$	A/D converter operation clock selection bit	0	Instruction clock (INSTCK)		
		1	On-chip oscillator (f(RING))		
$Q3_1$	A/D converter operation clock division ratio selection bits	$Q3_1$	$Q3_0$	Division ratio	
		0	0	Frequency divided by 6	
		0	1	Frequency divided by 12	
$Q3_0$		1	0	Frequency divided by 24	
		1	1	Frequency divided by 48	

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

(1) A/D control register

- A/D control register Q1
  Register Q1 controls the selection of A/D operation mode and the selection of analog input pins. Set the contents of this register through register A with the TQ1A instruction. The TAQ1 instruction can be used to transfer the contents of register Q1 to register A.
- A/D control register Q2
  Register Q2 controls the selection of P40/AIN4–P43/AIN7, P60/AIN0–P63/AIN3. 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.
- A/D control register Q3
  Register Q3 controls the selection of A/D converter operation clock. 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.

(2) Operating at A/D conversion mode

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

(3) Successive comparison register AD

Register AD stores the A/D conversion result of an analog input in 10-bit digital data format. The contents of the high-order 8 bits of this register can be stored in register B and register A with the TABAD instruction. The contents of the low-order 2 bits of this register can be stored into the high-order 2 bits of register A with the TALA instruction. However, do not execute these instructions during A/D conversion.

When the contents of register AD is n, the logic value of the comparison voltage Vref generated from the built-in D/A converter can be obtained with the reference voltage VDD by the following formula:

Logic value of comparison voltage Vref

$$Vref = \frac{VDD}{1024} \times n$$

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

(4) A/D conversion completion flag (ADF)

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

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

(5) A/D conversion start instruction (ADST)

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

(6) Operation description

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

① When the A/D conversion starts, the register AD is cleared to "000_{16}."
② Next, the topmost bit of the register AD is set to "1," and the comparison voltage Vref is compared with the analog input voltage VIN.
③ When the comparison result is Vref < VIN, the topmost bit of the register AD remains set to "1." When the comparison result is Vref > VIN, it is cleared to "0."

The 4519 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 2 machine cycles + A/D conversion clock (31 μs when f(XIN) = 6.0 MHz in XIN through mode, f(ADCK) = f(INSTCK)/6) 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 36).

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

At starting conversion	Change of successive comparison register AD	Comparison voltage (Vref) value
1st comparison	1 0 0 ----- 0 0 0	$\frac{VDD}{2}$
2nd comparison	✻1 1 0 ----- 0 0 0	$\frac{VDD}{2} \pm \frac{VDD}{4}$
3rd comparison	✻1 ✻2 1 ----- 0 0 0	$\frac{VDD}{2} \pm \frac{VDD}{4} \pm \frac{VDD}{8}$
After 10th comparison completes	A/D conversion result ✻1 ✻2 ✻3 ----- ✻8 ✻9 ✻A	$\frac{VDD}{2} \pm \cdots \pm \frac{VDD}{1024}$

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

(7) A/D conversion timing chart

Figure 36 shows the A/D conversion timing chart.

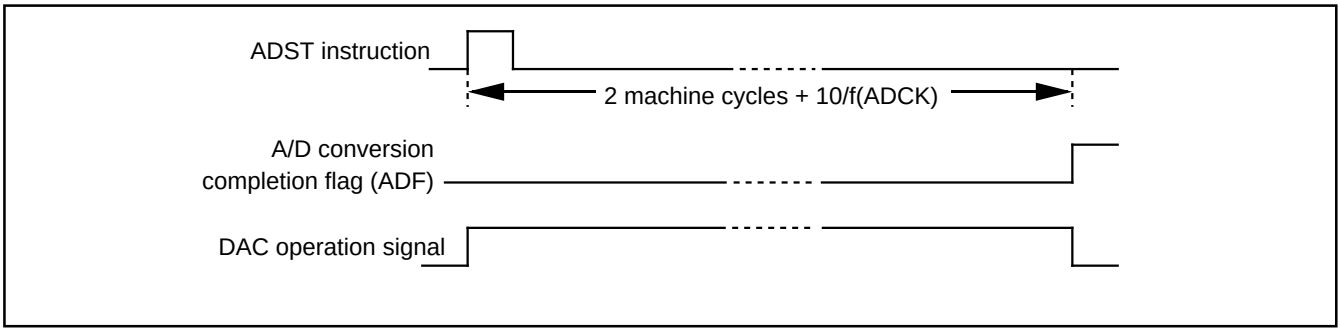


Fig. 36 A/D conversion timing chart

(8) How to use A/D conversion

How to use A/D conversion is explained using as example in which the analog input from P6$_0$/AIN$_0$ 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.
Instruction clock/6 is selected as the A/D converter operation clock.

① Select the AIN$_0$ pin function with the bit 0 of the register Q2. Select the AIN$_0$ pin function and A/D conversion mode with the register Q1. Also, the instruction clock divided by 6 is selected with the register Q3. (refer to Figure 37)
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).

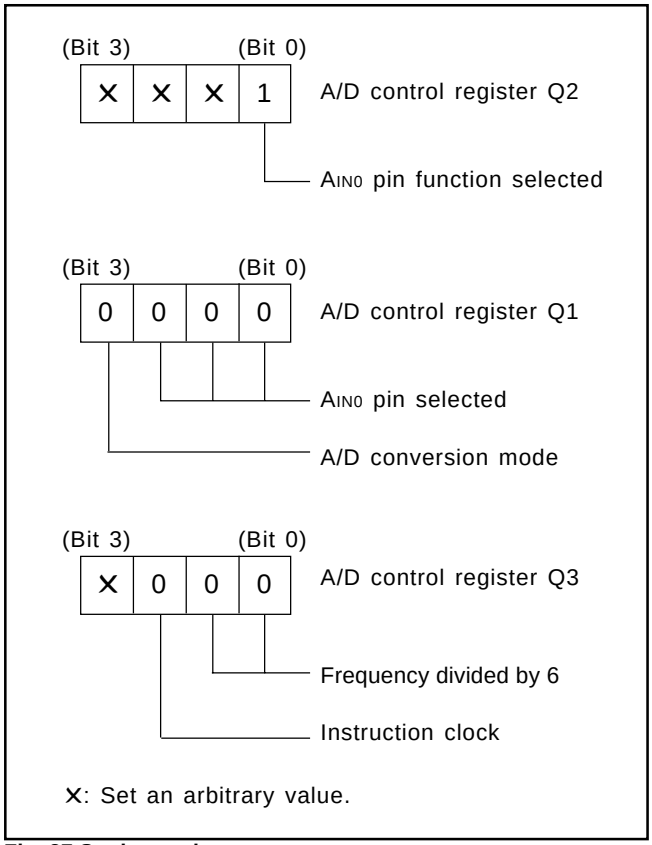


Fig. 37 Setting registers

(9) Operation at comparator mode

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

(10) Comparator register

In comparator mode, the built-in D/A comparator is connected to the 8-bit comparator register as a register for setting comparison voltages. The contents of register B is stored in the high-order 4 bits of the comparator register and the contents of register A is stored in the low-order 4 bits of the comparator register with the TADAB instruction.
When changing from A/D conversion mode to comparator mode, the result of A/D conversion (register AD) is undefined.
However, because the comparator register is separated from register AD, the value is retained even when changing from comparator mode to A/D conversion mode. Note that the comparator register can be written and read at only comparator mode.
If the value in the comparator register is n, the logic value of comparison voltage V_{ref} generated by the built-in D/A converter can be determined from the following formula:

Logic value of comparison voltage V_{ref}

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

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

(11) Comparison result store flag (ADF)

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

(12) Comparator operation start instruction (ADST instruction)

In comparator mode, executing ADST starts the comparator operating.
The comparator stops 2 machine cycles + A/D conversion clock f(ADCK) 1 clock after it has started (4 μs at f(XIN) = 6.0 MHz in XIN through mode, f(ADCK) = f(INSTCK)/6). When the analog input voltage is lower than the comparison voltage, the ADF flag is set to "1."

(13) Notes for the use of A/D conversion

- 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."

- Operation mode of A/D converter
  Do not change the operating mode (both A/D conversion mode and comparator mode) of A/D converter with the bit 3 of register Q1 while the A/D converter is operating.
  Clear the bit 2 of register V2 to "0" to change the operating mode of the A/D converter from the comparator mode to A/D conversion mode.
  The A/D conversion completion flag (ADF) may be set when the operating mode of the A/D converter is changed from the comparator mode to the A/D conversion mode. Accordingly, set a value to the register Q1, and execute the SNZAD instruction to clear the ADF flag.

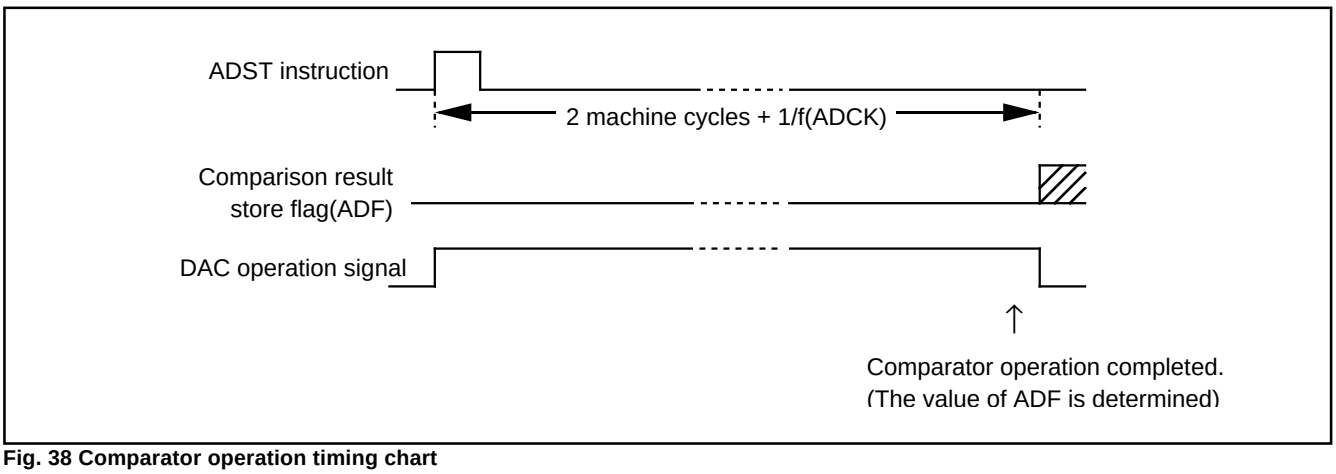


Fig. 38 Comparator operation timing chart

(14) Definition of A/D converter accuracy

The A/D conversion accuracy is defined below (refer to Figure 39).

- 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.

Vn: 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)

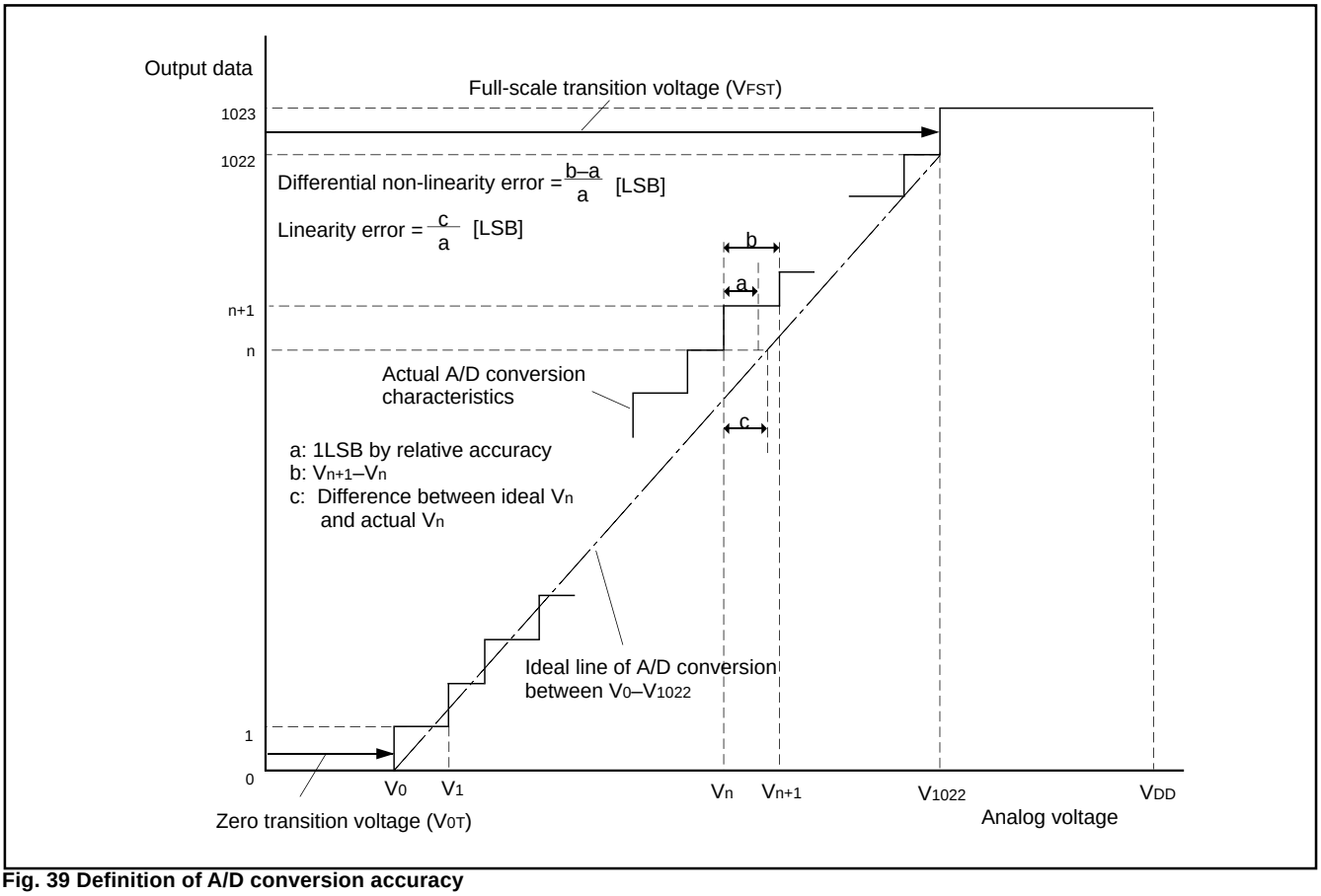


Fig. 39 Definition of A/D conversion accuracy

SERIAL I/O

The 4519 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 control register J1
- serial I/O transmit/receive 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 14 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: Even when the SCK, SOUT, SIN pin functions are used, the input of $P2_0$, $P2_1$, $P2_2$ are valid.

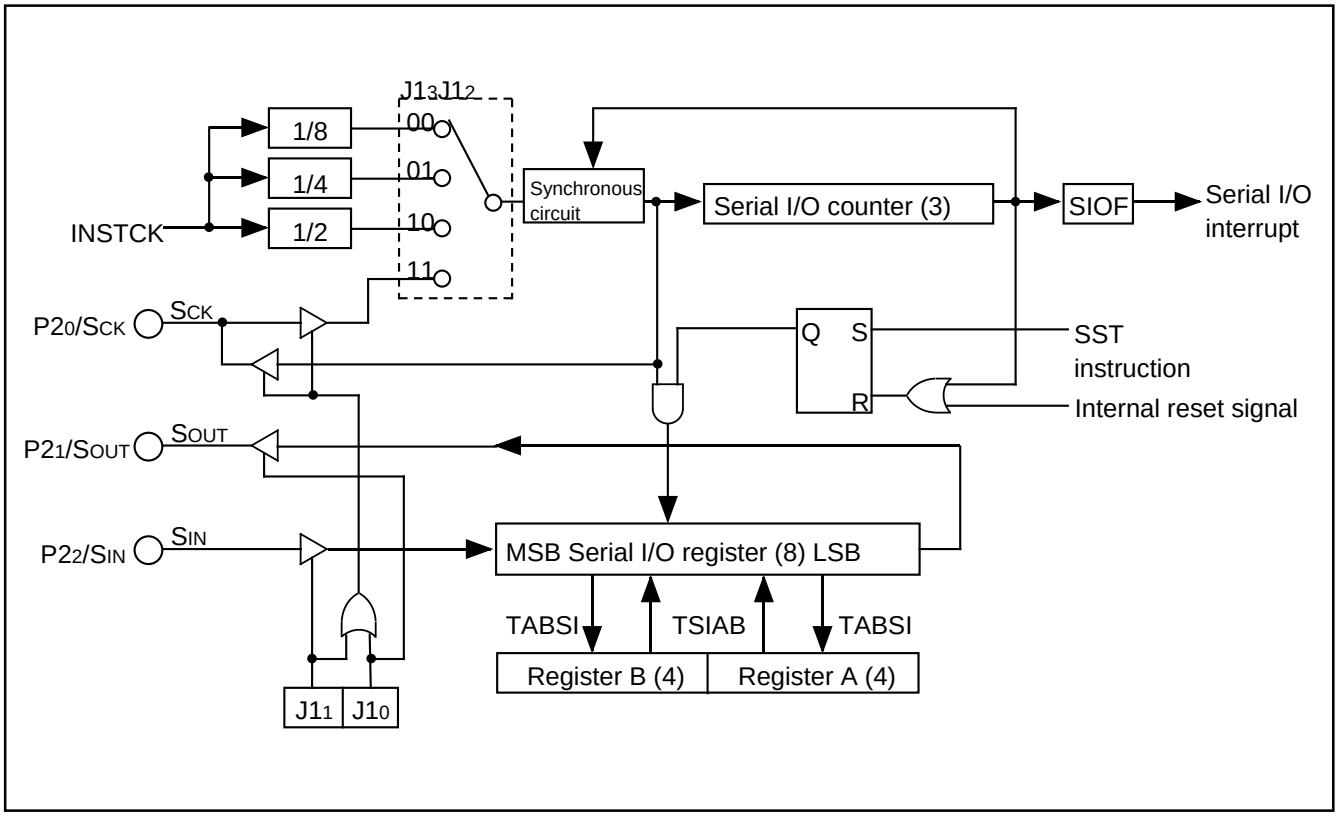


Fig. 40 Serial I/O structure

Table 15 Serial I/O control register

Serial I/O control register J1		at reset : 0000_2		at RAM back-up : state retained	R/W TAJ1/TJ1A
$J1_3$	Serial I/O synchronous clock selection bits	$J1_3$	$J1_2$	Synchronous clock	
$J1_2$		0	0	Instruction clock (INSTCK) divided by 8	
		0	1	Instruction clock (INSTCK) divided by 4	
		1	0	Instruction clock (INSTCK) divided by 2	
		1	1	External clock (SCK input)	
$J1_1$	Serial I/O port function selection bits	$J1_1$	$J1_0$	Port function	
		0	0	$P2_0$, $P2_1$,$P2_2$ selected/SCK, SOUT, SIN not selected	
$J1_0$		0	1	SCK, SOUT, $P2_2$ selected/$P2_0$, $P2_1$, SIN not selected	
		1	0	SCK, $P2_1$, SIN selected/$P2_0$, SOUT, $P2_2$ not selected	
		1	1	SCK, SOUT, SIN selected/$P2_0$, $P2_1$,$P2_2$ not selected	

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

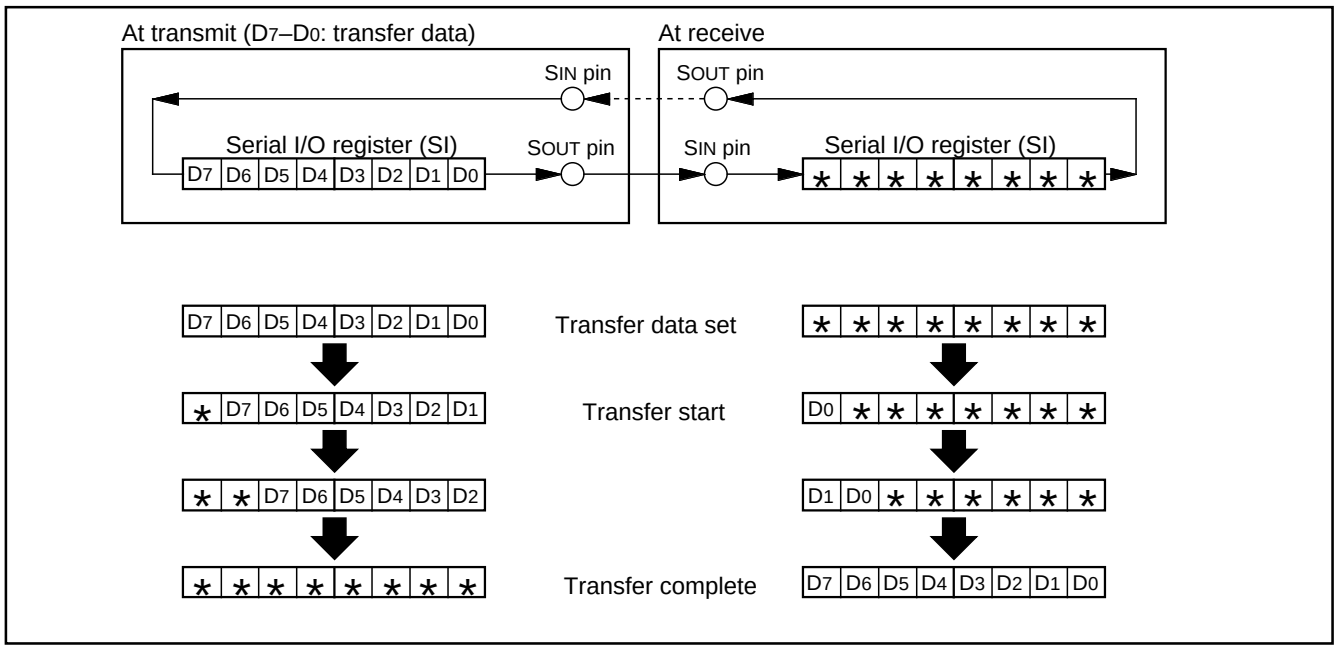


Fig. 41 Serial I/O register state when transferring

(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, do not select the SCK pin.

(2) Serial I/O transmit/receive completion flag (SIOF)

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

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

(3) Serial 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 control register J1

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

(5) How to use serial I/O

Figure 42 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 42 shows the data transfer timing and Table 16 shows the data transfer sequence.

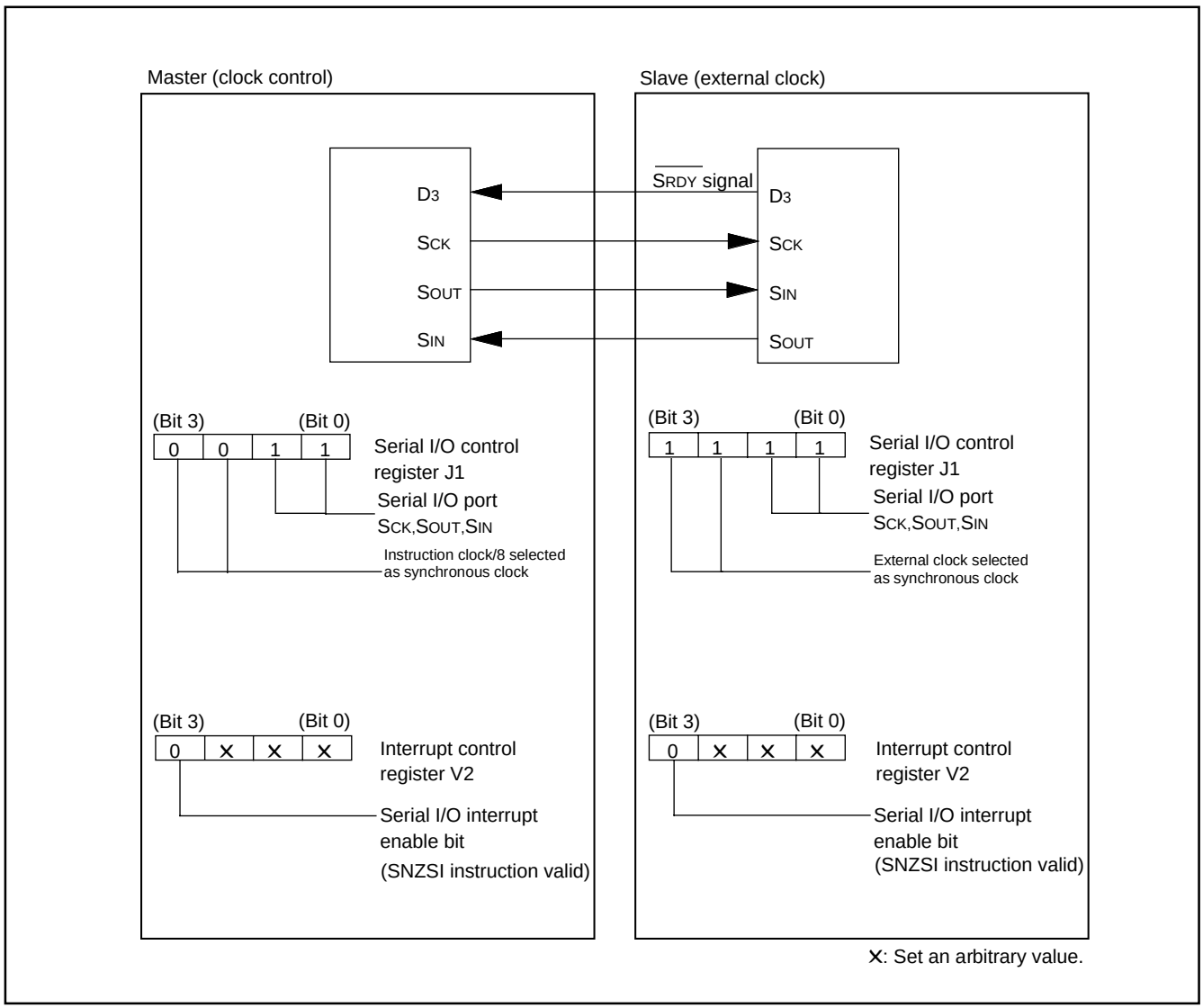


Fig. 42 Serial I/O connection example

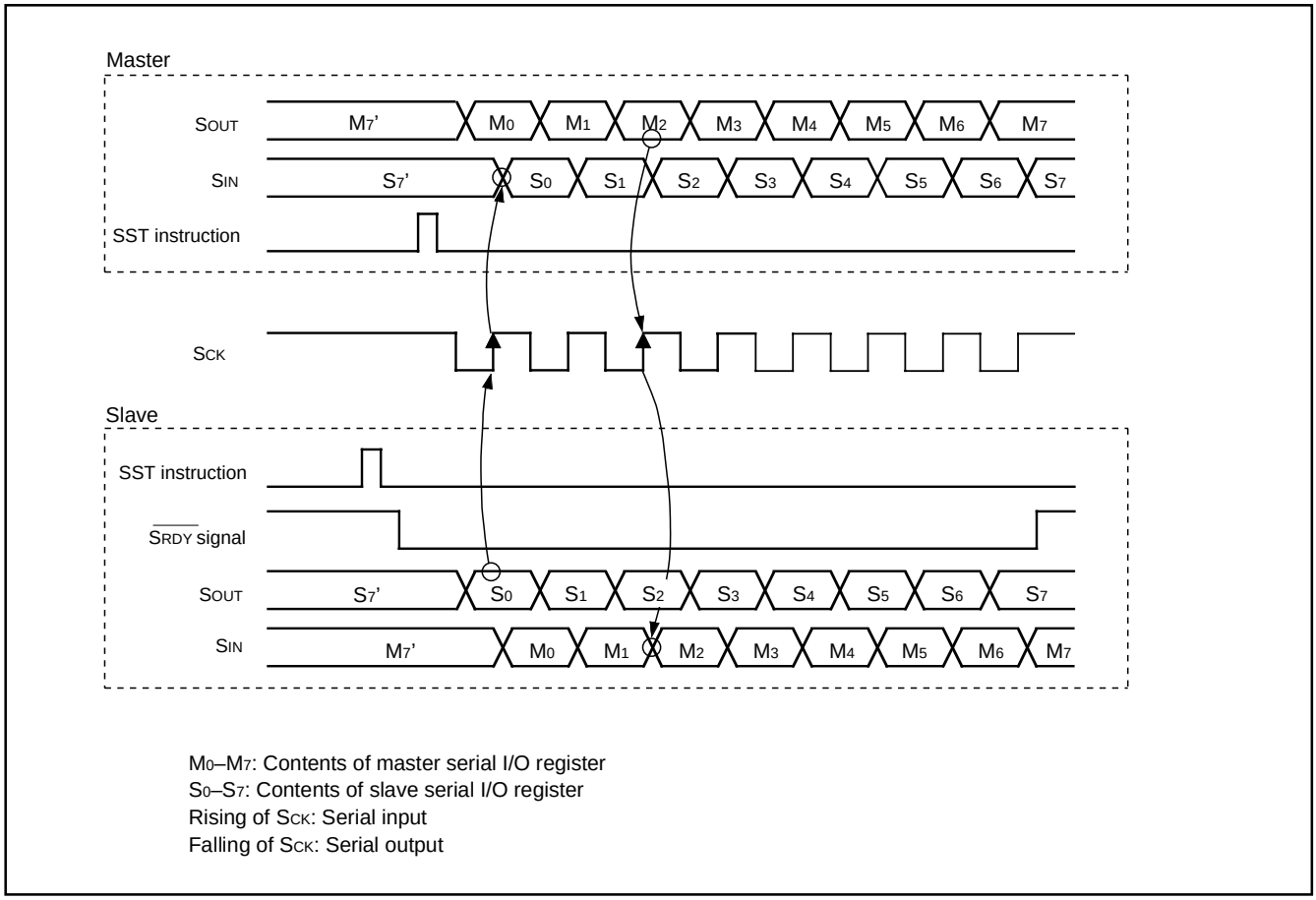


Fig. 43 Timing of serial I/O data transfer

Table 16 Processing sequence of data transfer from master to slave

Master (transmission)	Slave (reception)
[Initial setting] • Setting the serial I/O mode register J1 and interrupt control register V2 shown in Figure 42.	[Initial setting] • Setting serial I/O mode register J1, and interrupt control register V2 shown in Figure 42.
TJ1A and TV2A instructions	TJ1A and TV2A instructions
• Setting the port received the reception enable signal ($\overline{SRDY}$) to the input mode. (Port D_3 is used in this example)	• Setting the port transmitted the reception enable signal ($\overline{SRDY}$) and outputting "H" level (reception impossible). (Port D_3 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_3.
	RD instruction
[Transmission] •Check port D_3 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_3.
	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."

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.

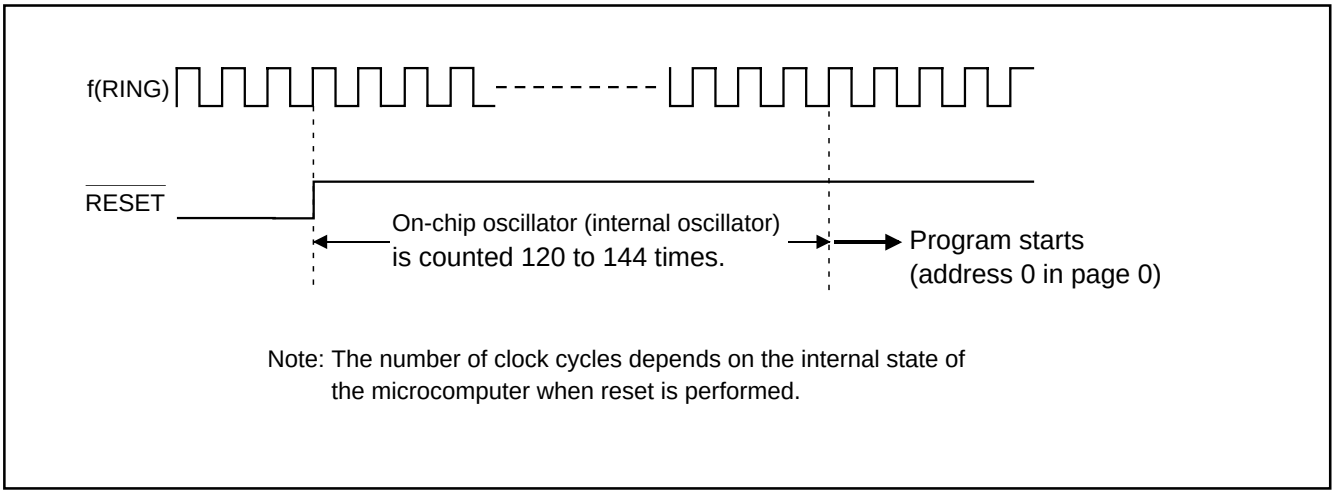


Fig. 44 Reset release timing

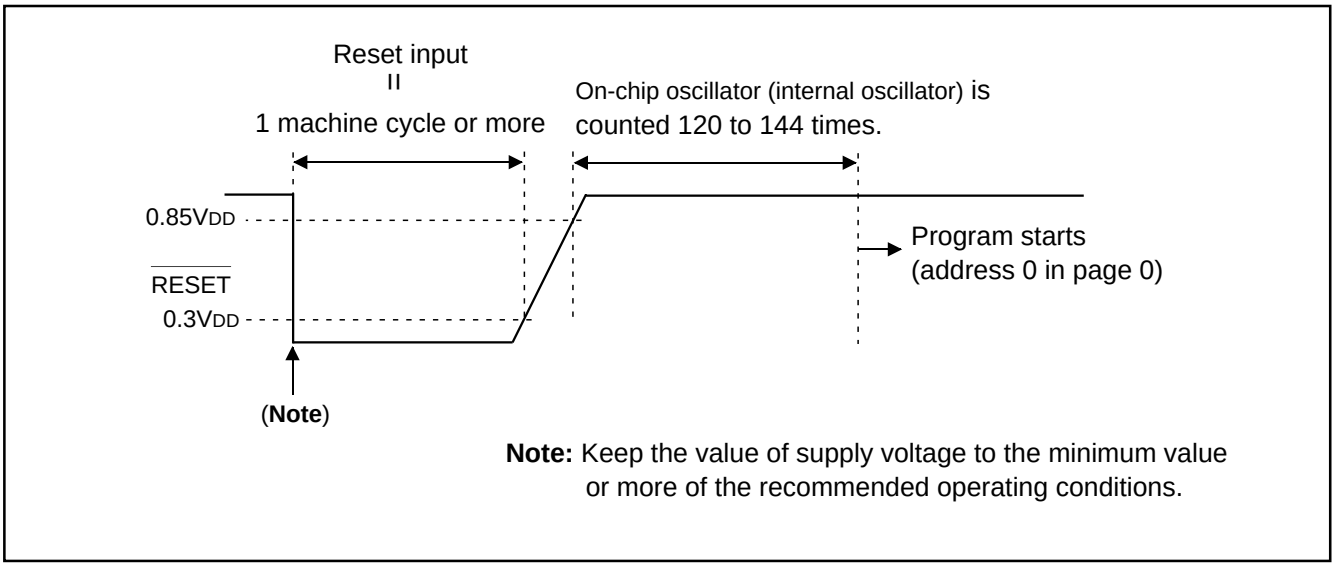
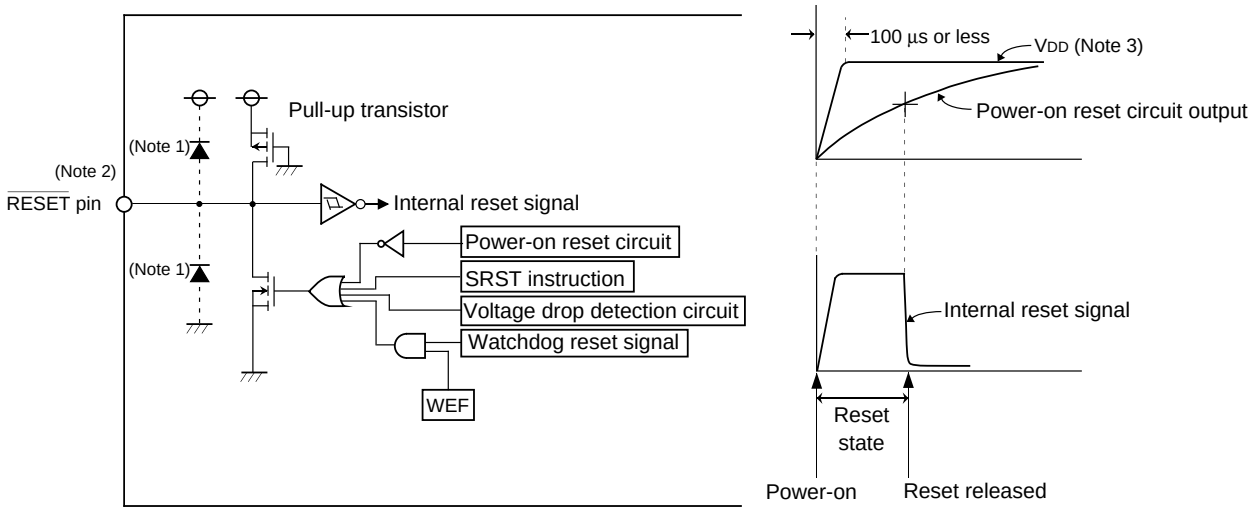


Fig. 45 $\overline{\text{RESET}}$ pin input waveform and reset operation

(1) Power-on reset

Reset can be automatically performed at power on (power-on reset) by the built-in power-on reset circuit. When the built-in power-on reset circuit is used, the time for the supply voltage to rise from 0 V until the value of supply voltage reaches the minimum operating voltage must be set to 100 µs or less.

If the rising time exceeds 100 µs, connect a capacitor between the $\overline{\text{RESET}}$ pin and V_{SS} at the shortest distance, and input "L" level to $\overline{\text{RESET}}$ pin until the value of supply voltage reaches the minimum operating voltage.



Notes 1: ----|◀---- This symbol represents a parasitic diode.
2: Applied potential to $\overline{\text{RESET}}$ pin must be V_{DD} or less.
3: Keep the value of supply voltage to the minimum value or more of the recommended operating conditions.

Fig. 46 Structure of reset pin and its peripherals, and power-on reset operation

Table 17 Port state at reset

Name	Function	State
D_0–D_5	D_0–D_5	High-impedance (Notes 1, 2)
D_6/CNTR0	D_6	High-impedance (Notes 1, 2)
D_7/CNTR1	D_7	High-impedance (Notes 1, 2)
$P0_0$–$P0_3$	$P0_0$–$P0_3$	High-impedance (Notes 1, 2, 3)
$P1_0$–$P1_3$	$P1_0$–$P1_3$	High-impedance (Notes 1, 2, 3)
$P2_0$/SCK, $P2_1$/SOUT, $P2_2$/SIN	$P2_0$–$P2_2$	High-impedance (Note 1)
$P3_0$/INT0, $P3_1$/INT1, $P3_2$, $P3_3$	$P3_0$–$P3_3$	High-impedance (Note 1)
$P4_0$/AIN4–$P4_3$/AIN7	$P4_0$–$P4_3$	High-impedance (Note 1)
$P5_0$–$P5_3$	$P5_0$–$P5_3$	High-impedance (Notes 1, 2)
$P6_0$/AIN0–$P6_3$/AIN3	$P6_0$–$P6_3$	High-impedance (Note 1)

Notes 1: Output latch is set to "1."
2: Output structure is N-channel open-drain.
3: Pull-up transistor is turned OFF.

(2) Internal state at reset

Figure 47 and 48 show 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 are undefined, so set the initial value to them.

- Program counter (PC) 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  Address 0 in page 0 is set to program counter.
- Interrupt enable flag (INTE) ..... 0 (Interrupt disabled)
- Power down flag (P) 0
- External 0 interrupt request flag (EXF0) 0
- 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) 1
- Timer control register PA ..... 0 (Prescaler stopped)
- Timer control register W1 ..... 0 0 0 0 (Timer 1 stopped)
- Timer control register W2 ..... 0 0 0 0 (Timer 2 stopped)
- Timer control register W3 ..... 0 0 0 0 (Timer 3 stopped)
- Timer control register W4 ..... 0 0 0 0 (Timer 4 stopped)
- Timer control register W5 ..... 0 0 0 0 (Period measurement circuit stopped)
- Timer control register W6 ..... 0 0 0 0
- Clock control register MR 1 1 1 1
- Clock control register RG ..... 0 (On-chip oscillator operating)
- Serial I/O transmit/receive completion flag (SIOF) 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
- A/D 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
- Key-on wakeup control register K1 ..... 0 0 0 0
- Key-on wakeup control register K2 ..... 0 0 0 0
- Pull-up control register PU0 ..... 0 0 0 0
- Pull-up control register PU1 ..... 0 0 0 0

"X" represents undefined.

Fig. 47 Internal state at reset 1

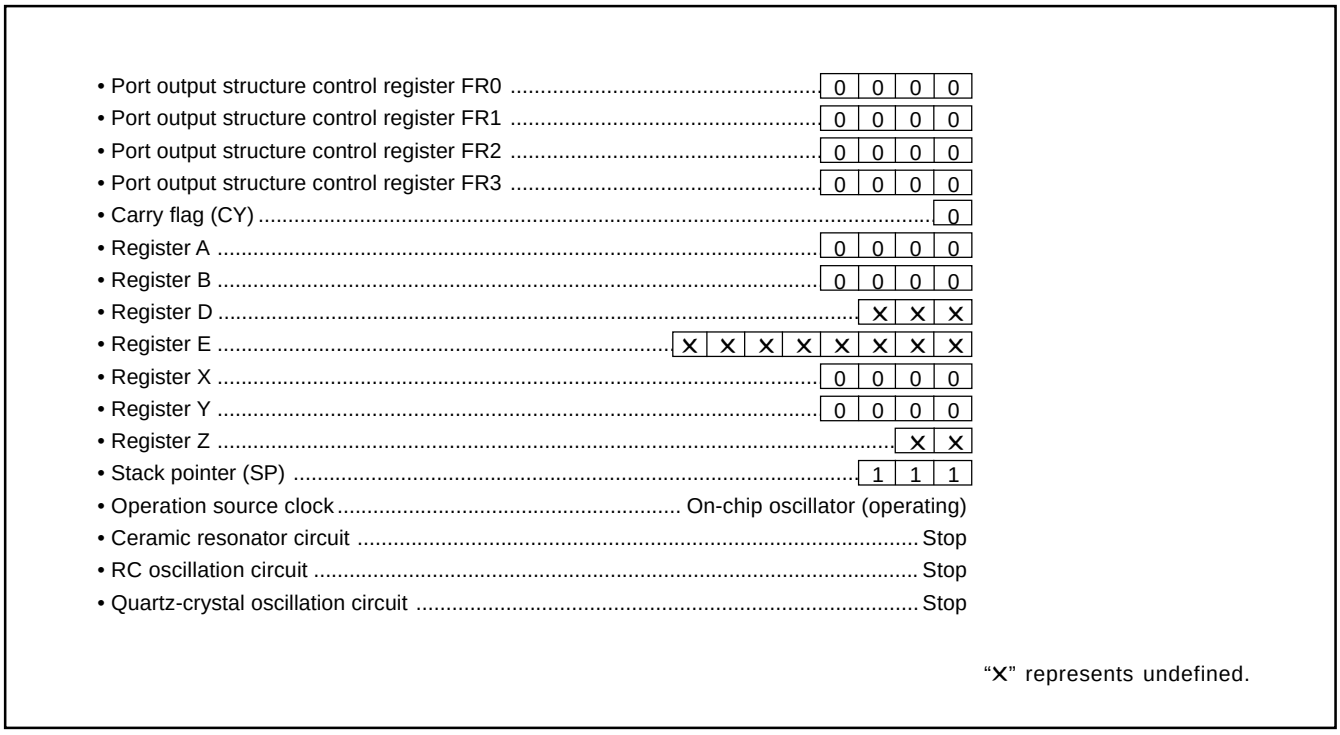


Fig. 48 Internal state at reset 2

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.

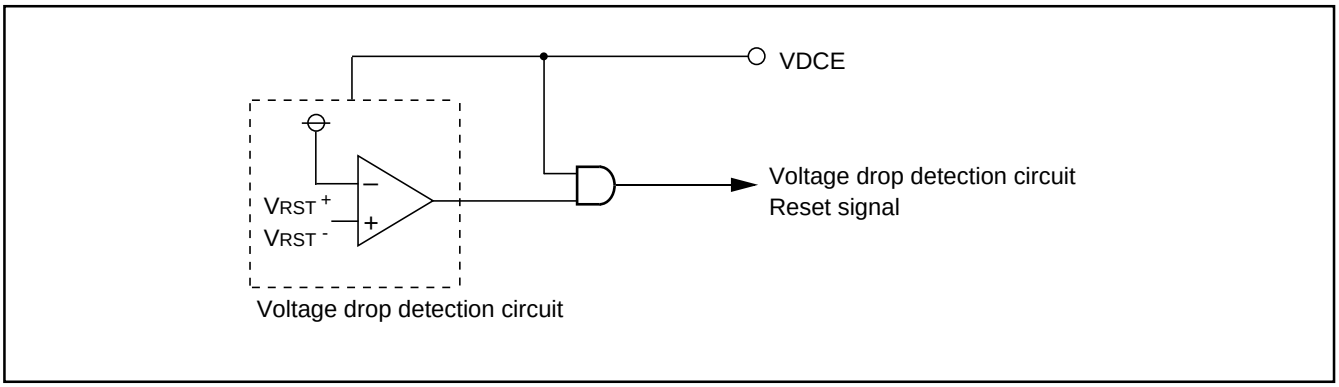


Fig. 49 Voltage drop detection reset circuit

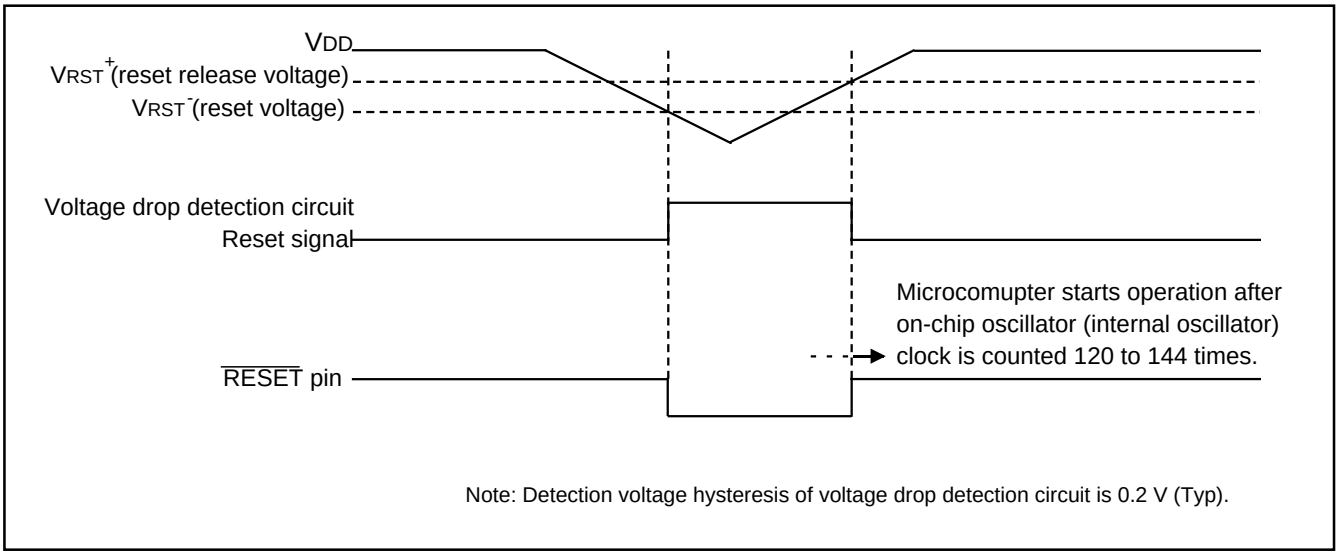


Fig. 50 Voltage drop detection circuit operation waveform

Table 18 Voltage drop detection circuit operation state

VDCE pin	At CPU operating	At RAM back-up
"L"	Invalid	Invalid
"H"	Valid	Valid

RAM BACK-UP MODE

The 4519 Group has the RAM back-up mode.

When the EPOF and POF instructions are executed continuously, system enters the RAM back-up state. The POF instruction is equal to the NOP instruction when the EPOF instruction is not executed before the POF instruction.

As oscillation stops retaining RAM, the function of reset circuit and states at RAM back-up mode, current dissipation can be reduced without losing the contents of RAM. Table 18 shows the function and states retained at RAM back-up. Figure 51 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 RAM back-up 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, or
- SRST instruction is executed.

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

Table 19 Functions and states retained at RAM back-up

Function	RAM back-up
Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) (Note 2)	×
Contents of RAM	O
Interrupt control registers V1, V2	×
Interrupt control registers I1, I2	O
Selection of oscillation circuit	O
Clock control register MR	×
Timer 1 function	(Note 3)
Timer 2 function	(Note 3)
Timer 3 function	(Note 3)
Timer 4 function	(Note 3)
Watchdog timer function	× (Note 4)
Timer control register PA, W4	×
Timer control registers W1 to W3, W5, W6	O
Serial I/O function	×
Serial I/O mode register J1	O
A/D conversion function	×
A/D control registers Q1 to Q3	O
Voltage drop detection circuit	O (Note 5)
Port level	O
Key-on wakeup control register K0 to K2	O
Pull-up control registers PU0, PU1	O
Port output direction registers FR0 to FR3	O
External 0 interrupt request flag (EXF0)	×
External 1 interrupt request flag (EXF1)	×
Timer 1 interrupt request flag (T1F)	(Note 3)
Timer 2 interrupt request flag (T2F)	(Note 3)
Timer 3 interrupt request flag (T3F)	(Note 3)
Timer 4 interrupt request flag (T4F)	(Note 3)
A/D conversion completion flag (ADF)	×
Serial I/O transmission/reception completion flag (SIOF)	×
Interrupt enable flag (INTE)	×
Watchdog timer flags (WDF1, WDF2)	× (Note 4)
Watchdog timer enable flag (WEF)	× (Note 4)

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 valid/invalid of the voltage drop detection circuit can be controlled only by VDCE pin.

(4) Return signal

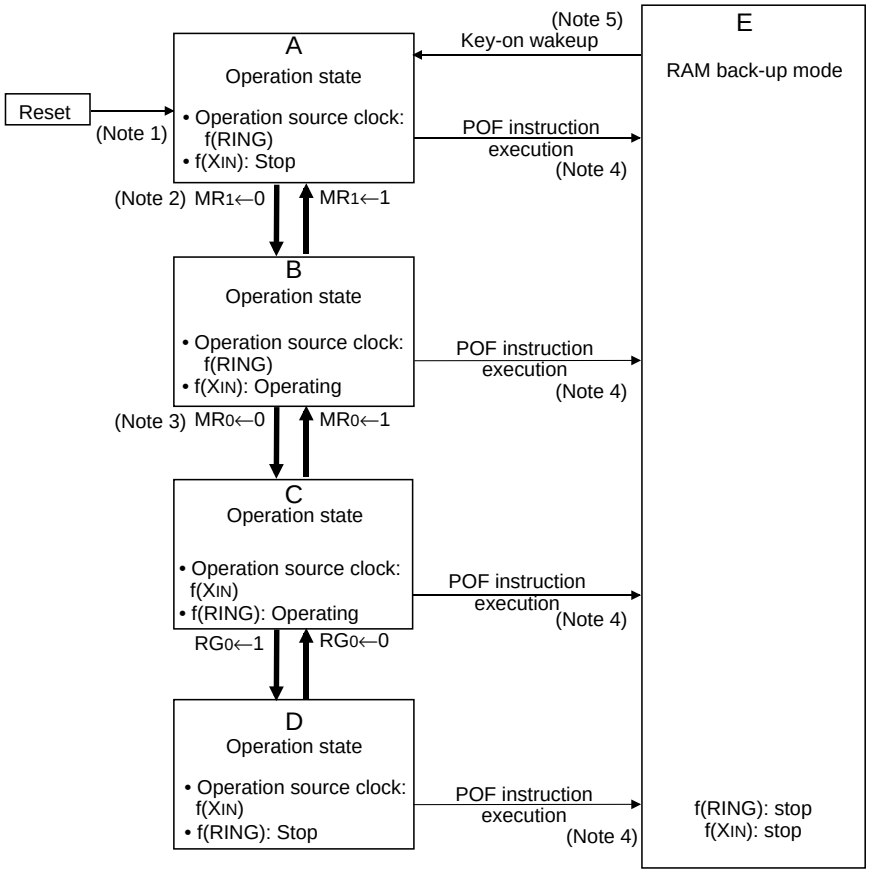
An external wakeup signal is used to return from the RAM back-up mode because the oscillation is stopped. Table 19 shows the return condition for each return source.

(5) Related 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.
- Key-on wakeup control register K1
  Register K1 controls the return condition and valid waveform/ level selection for port P0. Set the contents of this register through register A with the TK1A instruction. In addition, the TAK1 instruction can be used to transfer the contents of register K1 to register A.
- Key-on wakeup control register K2
  Register K2 controls the INT0 and INT1 key-on wakeup functions and return condition function. Set the contents of this register through register A with the TK2A instruction. In addition, the TAK2 instruction can be used to transfer the contents of register K2 to register A.
- Pull-up control register PU0
  Register PU0 controls the ON/OFF of the port P0 pull-up transistor. Set the contents of this register through register A with the TPU0A instruction. In addition, the TAPU0 instruction can be used to transfer the contents of register PU0 to register A.
- Pull-up control register PU1
  Register PU1 controls the ON/OFF of the port P1 pull-up transistor. Set the contents of this register through register A with the TPU1A instruction. In addition, the TAPU1 instruction can be used to transfer the contents of register PU0 to register A.
- External interrupt control register I1
  Register I1 controls the valid waveform of external 0 interrupt, input control of INT0 pin, and return input level. Set the contents of this register through register A with the TI1A instruction. In addition, the TAI1 instruction can be used to transfer the contents of register I1 to register A.
- External interrupt control register I2
  Register I2 controls the valid waveform of external 1 interrupt, input control of INT1 pin, and return input level. 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 20 Return source and return condition

Return source		Return condition	Remarks
External wakeup signal	Ports $P0_0$–$P0_3$	Return by an external “H” level or “L” level input, or rising edge (“L”→“H”) or falling edge (“H”→“L”).	The key-on wakeup function can be selected with 2 port units. Select the return level (“L” level or “H” level), and return condition (return by level or edge) with the register K1 according to the external state before going into the RAM back-up state.
	Ports $P1_0$–$P1_3$	Return by an external “L” level input.	The key-on wakeup function can be selected with 2 port units. Set the port using the key-on wakeup function to “H” level before going into the RAM back-up state.
	INT0 INT1	Return by an external “H” level or “L” level input, or rising edge (“L”→“H”) or falling edge (“H”→“L”). The external interrupt request flags (EXF0, EXF1) are not set.	Select the return level (“L” level or “H” level) with the registers I1 and I2 according to the external state, and return condition (return by level or edge) with the register K2 before going into the RAM back-up state.



Notes 1: Microcomputer starts its operation after counting f(RING) 120 to 144 times.
2: The f(XIN) oscillation circuit (ceramic resonance, RC oscillation or quartz-crystal oscillation) is selected by the CMCK, CRCK or CYCK instruction (the start of oscillation and the operation source clock is not switched by these instructions).
The start/stop of oscillation and the operation source is switched by register MR.
Surely, select the f(XIN) oscillation circuit by executing the CMCK, CRCK or CYCK instruction before clearing MR1 to "0".
MR1 cannot be cleared to "0" when the oscillation circuit is not selected.
3: Generate the wait time by software until the oscillation is stabilized, and then, switch the system clock.
4: Continuous execution of the EPOF instruction and the POF instruction is required to go into the RAM back-up state.
5: System returns to state A certainly when returning from the RAM back-up mode.
However, the selected contents (CMCK, CRCK, CYCK instruction execution state) of f(XIN) oscillation circuit is retained.

Fig. 51 State transition

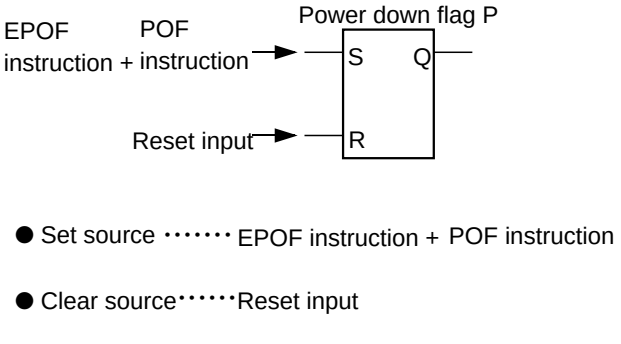


Fig. 52 Set source and clear source of the P flag

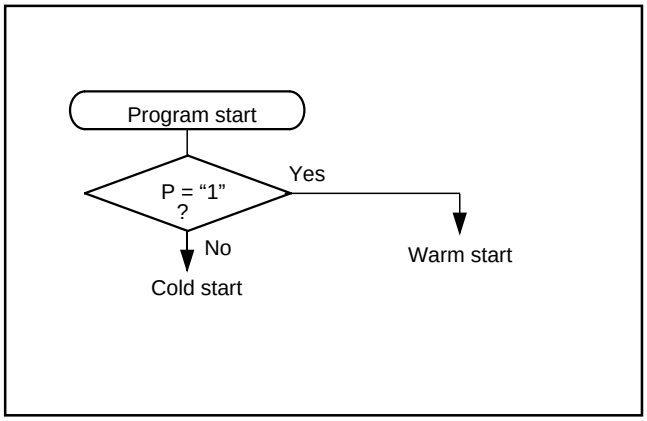


Fig. 53 Start condition identified example using the SNZP instruction

Table 21 Key-on wakeup control register, pull-up control register

Key-on wakeup control register K0		at reset : 0000_2	at RAM back-up : state retained	R/W TAK0/TK0A
$K0_3$	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	
Key-on wakeup control register K1		at reset : 0000_2	at RAM back-up : state retained	R/W TAK1/TK1A
$K1_3$	Ports $P0_2$ and $P0_3$ return condition selection bit	0	Return by level	
		1	Return by edge	
$K1_2$	Ports $P0_2$ and $P0_3$ valid waveform/ level selection bit	0	Falling waveform/“L” level	
		1	Rising waveform/“H” level	
$K1_1$	Ports $P0_1$ and $P0_0$ return condition selection bit	0	Return by level	
		1	Return by edge	
$K1_0$	Ports $P0_1$ and $P0_0$ valid waveform/ level selection bit	0	Falling waveform/“L” level	
		1	Rising waveform/“H” level	
Key-on wakeup control register K2		at reset : 0000_2	at RAM back-up : state retained	R/W TAK2/TK2A
$K2_3$	INT1 pin return condition selection bit	0	Return by level	
		1	Return by edge	
$K2_2$	INT1 pin key-on wakeup contro bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K2_1$	INT0 pin return condition selection bit	0	Return by level	
		1	Return by edge	
$K2_0$	INT0 pin key-on wakeup contro bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	

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

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

Pull-up control register PU0		at reset : 0000_2		at RAM back-up : state retained	R/W TAPU0/ TPU0A
$PU0_3$	$P0_3$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU0_2$	$P0_2$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU0_1$	$P0_1$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU0_0$	$P0_0$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
Pull-up control register PU1		at reset : 0000_2		at RAM back-up : state retained	R/W TAPU1/ TPU1A
$PU1_3$	$P1_3$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU1_2$	$P1_2$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU1_1$	$P1_1$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU1_0$	$P1_0$ pin 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.

CLOCK CONTROL

The clock control circuit consists of the following circuits.

- On-chip oscillator (internal oscillator)
- Ceramic resonator
- RC oscillation circuit
- Quartz-crystal oscillation circuit
- Multi-plexer (clock selection circuit)
- Frequency divider
- Internal clock generating circuit

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

Figure 54 shows the structure of the clock control circuit.

The 4519 Group operates by the on-chip oscillator clock (f(RING)) which is the internal oscillator after system is released from reset.

Also, the ceramic resonator, the RC oscillation or quartz-crystal oscillator can be used for the main clock (f(XIN)) of the 4519 Group.

The CMCK instruction, CRCK instruction or CYCK instruction is executed to select the ceramic resonator, RC oscillator or quartz-crystal oscillator respectively.

The CMCK, CRCK, and CYCK instructions can be used only to select main clock (f(XIN)). In this time, the start of oscillation and the switch of system clock are not performed.

The oscillation start/stop of main clock f(XIN) is controlled by bit 1 of register MR. The system clock is selected by bit 0 of register MR. The oscillation start/stop of on-chip oscillator is controlled by register RG.

The oscillation circuit by the CMCK, CRCK or CYCK instruction can be selected only at once.

The oscillation circuit corresponding to the first executed one of these instructions is valid.

Execute the main clock (f(XIN)) selection instruction (CMCK, CRCK or CYCK instruction) in the initial setting routine of program (executing it in address 0 in page 0 is recommended).

When the CMCK, CRCK, and CYCK instructions are never executed, main clock (f(XIN)) cannot be used and system can be operated only by on-chip oscillator.

The no operated clock source (f(RING)) or (f(XIN)) cannot be used for the system clock. Also, the clock source (f(RING) or f(XIN)) selected for the system clock cannot be stopped.

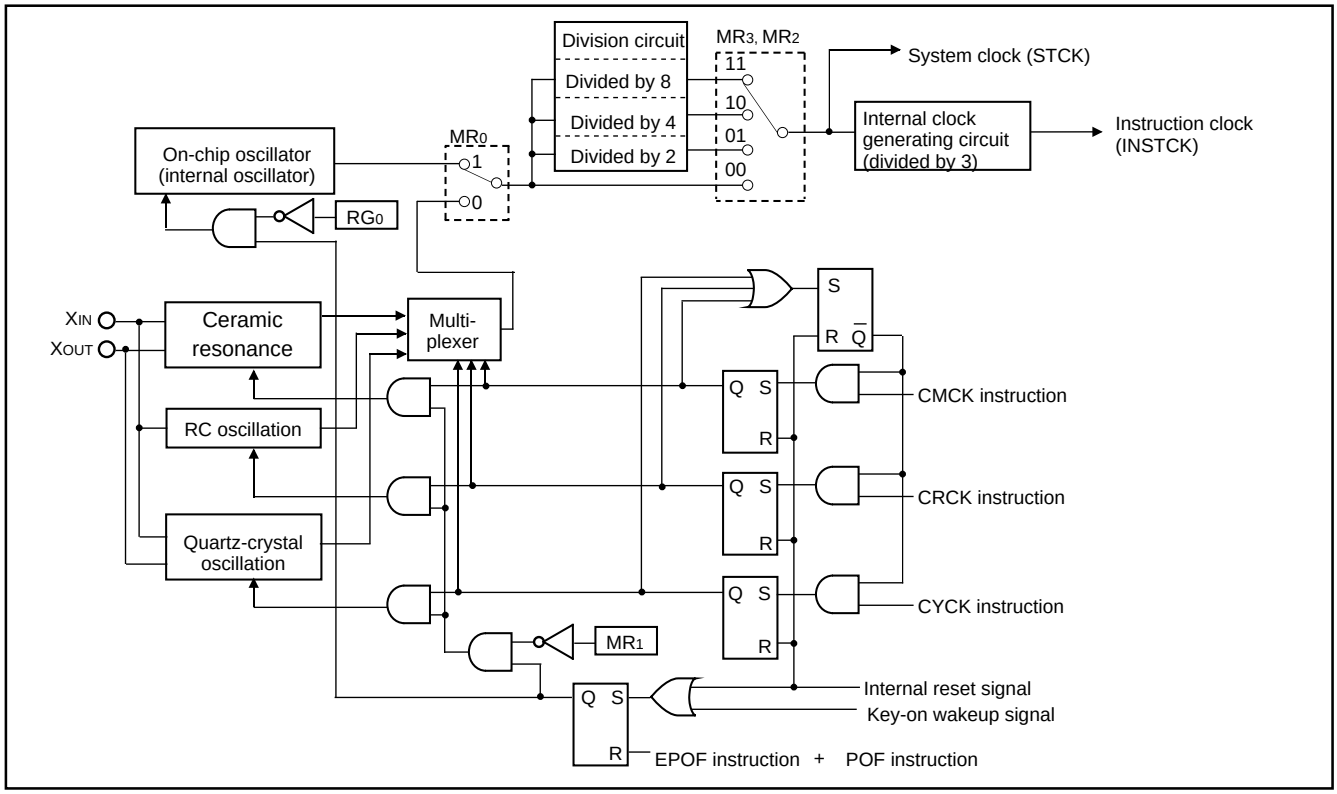


Fig. 54 Clock control circuit structure

(1) Main clock generating circuit (f(XIN))

The ceramic resonator, RC oscillation or quartz-crystal oscillator can be used for the main clock of this MCU.

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

When the ceramic resonator is used, execute the CMCK instruction. When the RC oscillation is used, execute the CRCK instruction. When the quartz-crystal oscillator is used, execute the CYCK instruction. The oscillation start/stop of main clock f(XIN) is controlled by bit 1 of register MR. The system clock is selected by bit 0 of register MR. The oscillation circuit by the CMCK, CRCK or CYCK instruction can be selected only at once. The oscillation circuit corresponding to the first executed one of these instructions is valid.

Execute the CMCK, CRCK or CYCK instruction in the initial setting routine of program (executing it in address 0 in page 0 is recommended). Also, when the CMCK, CRCK or CYCK instruction is not executed in program, this MCU operates by the on-chip oscillator.

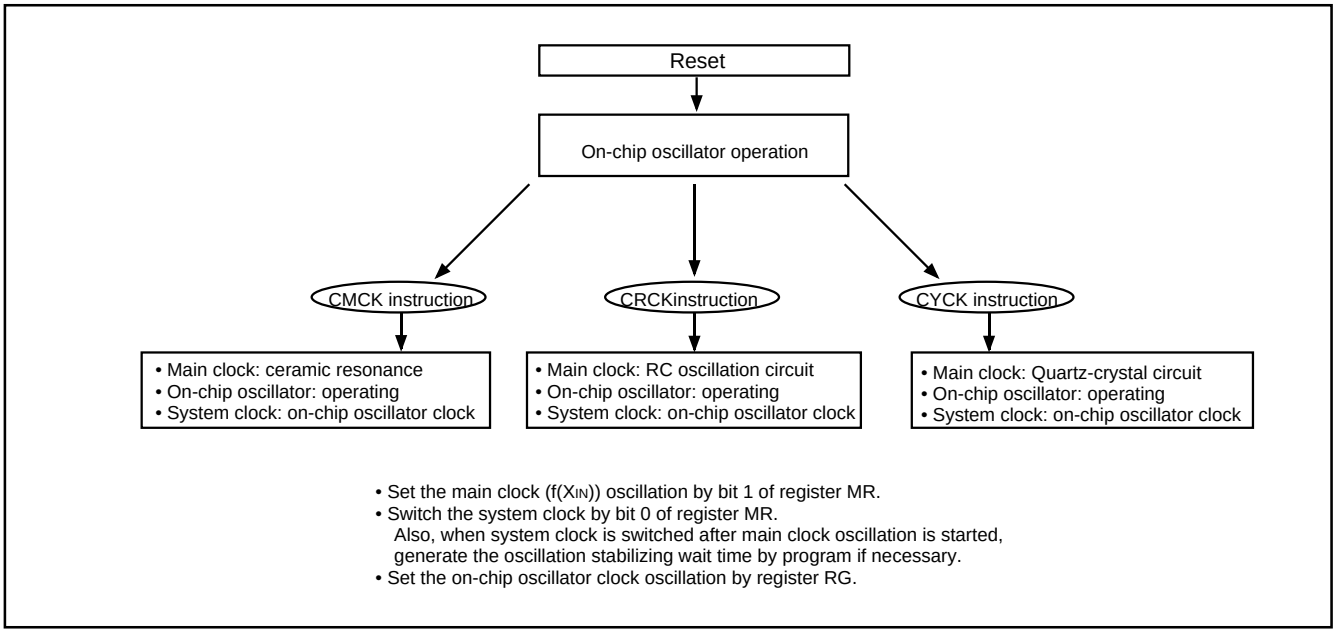


Fig. 55 Switch to ceramic resonance/RC oscillation/quartz-crystal oscillation

(2) On-chip oscillator operation

When the MCU operates by the on-chip oscillator as the main clock (f(XIN)) without using the ceramic resonator, RC oscillator or quartz-crystal oscillation, leave XIN pin and XOUT pin open (Figure 56).

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

Be careful that the margin of frequencies when designing application products.

(3) Ceramic resonator

When the ceramic resonator is used as the main clock (f(XIN)), connect the ceramic resonator and the external circuit to pins XIN and XOUT at the shortest distance. Then, execute the CMCK instruction. A feedback resistor is built in between pins XIN and XOUT (Figure 57).

(4) RC oscillation

When the RC oscillation is used as the main clock (f(XIN)), connect the XIN pin to the external circuit of resistor R and the capacitor C at the shortest distance and leave XOUT pin open. Then, execute the CRCK instruction (Figure 58).

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

(5) Quartz-crystal oscillator

When a quartz-crystal oscillator is used as the main clock (f(XIN)), connect this external circuit and a quartz-crystal oscillator to pins XIN and XOUT at the shortest distance. Then, execute the CYCK instruction. A feedback resistor is built in between pins XIN and XOUT (Figure 59).

(6) External clock

When the external clock signal for the main clock (f(XIN)) is used, connect the clock source to XIN pin and XOUT pin open. In program, after the CMCK instruction is executed, set main clock (f(XIN)) oscillation start to be enabled (MR1=0).

For this product, when RAM back-up mode and main clock (f(XIN)) stop (MR1=1), XIN pin is fixed to “H” in order to avoid the through current by floating of internal logic. The XIN pin is fixed to “H” until main clock (f(XIN)) oscillation starts to be valid (MR1=0) by the CMCK instruction from reset state. Accordingly, when an external clock is used, connect a 1 kΩ or more resistor to XIN pin in series to limit of current by competitive signal.

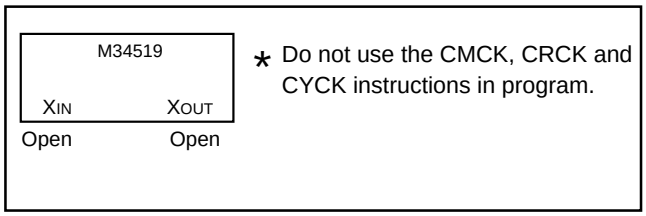


Fig. 56 Handling of XIN and XOUT when operating on-chip oscillator

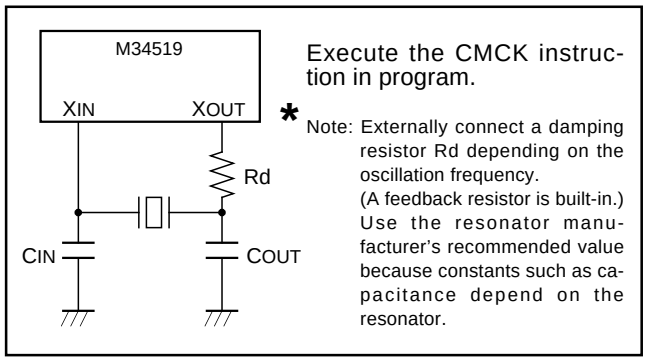


Fig. 57 Ceramic resonator external circuit

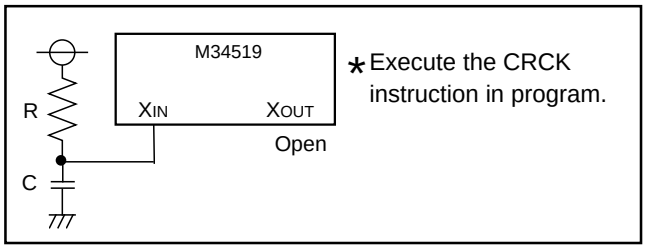


Fig. 58 External RC oscillation circuit

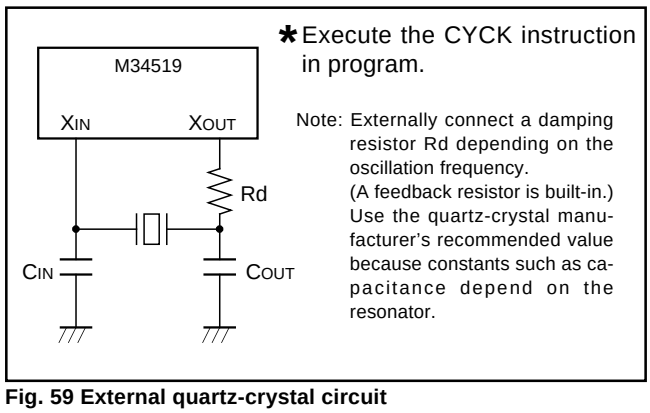


Fig. 59 External quartz-crystal circuit

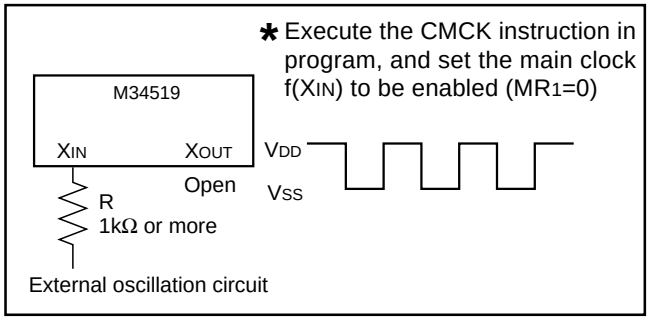


Fig. 60 External clock input circuit

(7) Clock control register MR

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

(8) Clock control register RG

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

Table 23 Clock control registers

Clock control register MR		at reset : 1111_2		at RAM back-up : 1111_2	R/W TAMR/ TMRA
MR_3	Operation mode selection bits	MR_3	MR_2	Operation mode	
		0	0	Through mode (frequency not divided)	
MR_2		0	1	Frequency divided by 2 mode	
		1	0	Frequency divided by 4 mode	
		1	1	Frequency divided by 8 mode	
MR_1	Main clock f(X_{IN}) oscillation circuit control bit	0		Main clock (f(X_{IN})) oscillation enabled	
		1		Main clock (f(X_{IN})) oscillation stop	
MR_0	System clock oscillation source selection bit	0		Main clock (f(X_{IN}))	
		1		Main clock (f(RING))	

Clock control register RG		at reset : 0_2	at RAM back-up : 0_2	W TRGA
RG_0	On-chip oscillator (f(RING)) control bit	0	On-chip oscillator (f(RING)) oscillation enabled	
		1	On-chip oscillator (f(RING)) oscillation stop	

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

ROM ORDERING METHOD

1.Mask ROM Order Confirmation Form*
2.Mark Specification Form*
3.Data to be written to ROM ..... one floppy disk.

*For the mask ROM confirmation and the mark specifications, refer to the “Renesas Technology Corp.” Homepage (http://www.renesas.com/en/rom).

LIST OF PRECAUTIONS

①Noise and latch-up prevention

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

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

In the One Time PROM version, CNV_{SS} pin is also used as V_{PP} pin. Accordingly, when using this pin, connect this pin to V_{SS} through a resistor about 5 kΩ (connect this resistor to CNV_{SS}/V_{PP} pin as close as possible).

②Register initial values 1

The initial value of the following registers are undefined after system is released from reset. After system is released from reset, set initial values.

- Register Z (2 bits)
- Register D (3 bits)
- Register E (8 bits)

③Register initial values 2

The initial value of the following registers are undefined at RAM back-up. After system is returned from RAM back-up, set initial values.

- Register Z (2 bits)
- Register X (4 bits)
- Register Y (4 bits)
- Register D (3 bits)
- Register E (8 bits)

④ Stack registers (SKs)

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.

⑤Multifunction

- The input/output of $P3_0$ and $P3_1$ can be used even when INT0 and INT1 are selected.
- The input of ports $P2_0$–$P2_2$ can be used even when S_{IN}, S_{OUT} and S_{CK} are selected.
- The input/output of D_6 can be used even when CNTR0 (input) is selected.
- The input of D_6 can be used even when CNTR0 (output) is selected.
- The input/output of D_7 can be used even when CNTR1 (input) is selected.
- The input of D_7 can be used even when CNTR1 (output) is selected.

⑥Prescaler

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

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

⑦Timer count source

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

⑧Reading the count value

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

⑨Writing to the timer

Stop timer 1, 2, 3 or 4 counting and then execute the data write instruction (T1AB, T2AB, T3AB, T4AB) to write its data.

⑩Writing to reload register R1, R3, R4H

When writing data to reload register R1, reload register R3 or reload regiser R4H while timer 1, timer 3 or timer 4 is operating, avoid a timing when timer 1, timer 3 or timer 4 underflows.

⑪Timer 4

In order to stop timer 4 while the PWM output function is used, avoid a timing when timer 4 underflows.

When “H” interval extension function of the PWM signal is set to be “valid”, set “1” or more to reload register R4H.

⑫Watchdog timer

- The watchdog timer function is valid after system is released from reset. When not using the watchdog timer function, execute the DWDT instruction and the WRST instruction continuously, and clear the WEF flag to “0” to stop the watchdog timer function.
- The watchdog timer function is valid after system is returned from the RAM back-up state. When not using the watchdog timer function, execute the DWDT instruction and the WRST instruction continuously every system is returned from the RAM back-up state, and stop the watchdog timer function.
- When the watchdog timer function and RAM back-up function are used at the same time, execute the WRST instruction before system enters into the RAM back-up state and initialize the flag WDF1.

⑬ Period measurement circuit

When a period measurement circuit is used, clear bit 0 of register I1 to “0”, and set a timer 1 count start synchronous circuit to be “not selected”.

Start timer operation immediately after operation of a period measurement circuit is started.

When the edge for measurement is input until timer operation is started from the operation of period measurement circuit is started, the count operation is not executed until the timer operation becomes valid. Accordingly, be careful of count data.

When data is read from timer, stop the timer and clear bit 2 of register W5 to “0” to stop the period measurement circuit, and then execute the data read instruction.

Depending on the state of timer 1, the timer 1 interrupt request flag (T1F) may be set to “1” when the period measurement circuit is stopped by clearing bit 2 of register W5 to “0”. In order to avoid the occurrence of an unexpected interrupt, clear the bit 2 of register V1 to “0” (refer to Figure 61①) and then, stop the bit 2 of register W5 to “0” to stop the period measurement circuit.

In addition, execute the SNZT1 instruction to clear the T1F flag after executing at least one instruction (refer to Figure 61②).

Also, set the NOP instruction for the case when a skip is performed with the SNZT1 instruction (refer to Figure 61③).

While a period measurement circuit is operating, the timer 1 interrupt request flag (T1F) is not set by the timer 1 underflow signal, it is the flag for detecting the completion of period measurement.

When a period measurement circuit is used, select the sufficiently higher-speed frequency than the signal for measurement for the count source of a timer 1.

When the signal for period measurement is D_6/CNTR0 pin input, do not select D_6/CNTR0 pin input as timer 1 count source.

(The X_{IN} input is recommended as timer 1 count source at the time of period measurement circuit use.)

When the input of $P3_0$/INT0 pin is selected for measurement, set the bit 3 of a register I1 to “1”, and set the input of INT0 pin to be enabled.

```
    ⋮
LA    0       ; (X0XX2)
TV1A          ; The SNZT1 instruction is valid ..... ①
LA    0       ; (X0XX2)
TW5A          ; Period measurement circuit stop
NOP           ..................................................... ②
SNZT1         ; The SNZT1 instruction is executed
                (T1F flag cleared)
NOP           ..................................................... ③
    ⋮
```

X : these bits are not used here.

Fig. 61 Period measurement circuit program example

⑭ P30/INT0 pin

❶ Note [1] on bit 3 of register I1

When the input of the INT0 pin is controlled with the bit 3 of register I1 in software, be careful about the following notes.

• Depending on the input state of the P30/INT0 pin, the external 0 interrupt request flag (EXF0) may be set when the bit 3 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 61 ①) and then, change the bit 3 of register I1.
In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 62 ②).
Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 62 ③).

```
    ⋮
LA    4      ; (XXX0₂)
TV1A         ; The SNZ0 instruction is valid ..... ①
LA    8      ; (1XXX₂)
TI1A         ; Control of INT0 pin input is changed
NOP          ..... ②
SNZ0         ; The SNZ0 instruction is executed
               (EXF0 flag cleared)
NOP          ..... ③
    ⋮
        X : these bits are not used here.
```

Fig. 62 External 0 interrupt program example-1

❷ Note [2] on bit 3 of register I1

When the bit 3 of register I1 is cleared to "0", the RAM back-up mode is selected and the input of INT0 pin is disabled, be careful about the following notes.

• When the input of INT0 pin is disabled (register $I1_3$ = "0"), set the key-on wakeup function to be invalid (register $K2_0$ = "0") before system enters to the RAM back-up mode. (refer to Figure 63①).

```
    ⋮
LA    0      ; (XXX0₂)
TK2A         ; Input of INT0 key-on wakeup invalid .. ①
DI
EPOF
POF          ; RAM back-up
    ⋮
      X : these bits are not used here.
```

Fig. 63 External 0 interrupt program example-2

❸ Note on bit 2 of register I1

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.

• Depending on the input state of the P30/INT0 pin, the external 0 interrupt request flag (EXF0) may be set when the bit 2 of register I1 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 0 of register V1 to "0" (refer to Figure 64①) and then, change the bit 2 of register I1.
In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to Figure 64②).
Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 64③).

```
    ⋮
LA    4      ; (XXX0₂)
TV1A         ; The SNZ0 instruction is valid ..... ①
LA    12     ; (X1XX₂)
TI1A         ; Interrupt valid waveform is changed
NOP          ..... ②
SNZ0         ; The SNZ0 instruction is executed
               (EXF0 flag cleared)
NOP          ..... ③
    ⋮
        X : these bits are not used here.
```

Fig. 64 External 0 interrupt program example-3

⑮P31/INT1 pin

❶ Note [1] on bit 3 of register I2

When the input of the INT1 pin is controlled with the bit 3 of register I2 in software, be careful about the following notes.

• Depending on the input state of the P31/INT1 pin, the external 1 interrupt request flag (EXF1) may be set when the bit 3 of register I2 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 1 of register V1 to "0" (refer to Figure 65①) and then, change the bit 3 of register I2.
In addition, execute the SNZ1 instruction to clear the EXF1 flag to "0" after executing at least one instruction (refer to Figure 65②).
Also, set the NOP instruction for the case when a skip is performed with the SNZ1 instruction (refer to Figure 65③).

```
    ⋮
LA    4       ; (XX0X2)
TV1A          ; The SNZ1 instruction is valid ..... ①
LA    8       ; (1XXX2)
TI2A          ; Control of INT1 pin input is changed
NOP           ..... ②
SNZ1          ; The SNZ1 instruction is executed
                (EXF1 flag cleared)
NOP           ..... ③
    ⋮
   X : these bits are not used here.
```

Fig. 65 External 1 interrupt program example-1

❷ Note [2] on bit 3 of register I2

When the bit 3 of register I2 is cleared to "0", the RAM back-up mode is selected and the input of INT1 pin is disabled, be careful about the following notes.

• When the input of INT1 pin is disabled (register I23 = "0"), set the key-on wakeup function to be invalid (register K22 = "0") before system enters to the RAM back-up mode. (refer to Figure 66①).

```
    ⋮
LA    0       ; (X0XX2)
TK2A          ; Input of INT1 key-on wakeup invalid .. ①
DI
EPOF
POF           ; RAM back-up
    ⋮
   X : these bits are not used here.
```

Fig. 66 External 1 interrupt program example-2

❸ Note on bit 2 of register I2

When the interrupt valid waveform of the P31/INT1 pin is changed with the bit 2 of register I2 in software, be careful about the following notes.

• Depending on the input state of the P31/INT1 pin, the external 1 interrupt request flag (EXF1) may be set when the bit 2 of register I2 is changed. In order to avoid the occurrence of an unexpected interrupt, clear the bit 1 of register V1 to "0" (refer to Figure 67①) and then, change the bit 2 of register I2.
In addition, execute the SNZ1 instruction to clear the EXF1 flag to "0" after executing at least one instruction (refer to Figure 67②).
Also, set the NOP instruction for the case when a skip is performed with the SNZ1 instruction (refer to Figure 67③).

```
    ⋮
LA    4       ; (XX0X2)
TV1A          ; The SNZ1 instruction is valid ..... ①
LA    12      ; (X1XX2)
TI2A          ; Interrupt valid waveform is changed
NOP           ..... ②
SNZ1          ; The SNZ1 instruction is executed
                (EXF1 flag cleared)
NOP           ..... ③
    ⋮
   X : these bits are not used here.
```

Fig. 67 External 1 interrupt program example-3

⑯ A/D converter-1

- When the TALA instruction is executed, the low-order 2 bits of register AD is transferred to the high-order 2 bits of register A, simultaneously, the low-order 2 bits of register A is "0."
- Do not change the operating mode (both A/D conversion mode and comparator mode) of A/D converter with the bit 3 of register Q1 while the A/D converter is operating.
- Clear the bit 2 of register V2 to "0" to change the operating mode of the A/D converter from the comparator mode to A/D conversion mode.
- The A/D conversion completion flag (ADF) may be set when the operating mode of the A/D converter is changed from the comparator mode to the A/D conversion mode. Accordingly, set a value to the register Q1, and execute the SNZAD instruction to clear the ADF flag.

```
      ⋮
LA    8        ; (X0XX2)
TV2A           ; The SNZAD instruction is valid ..... ①
LA    0        ; (0XXX2)
TQ1A           ; Operation mode of A/D converter is
                 changed from comparator mode to A/D
                 conversion mode.
SNZAD
NOP
      ⋮        X : these bits are not used here.
```

Fig. 68 A/D converter program example-3

⑰ 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 69).
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 70. In addition, test the application products sufficiently.

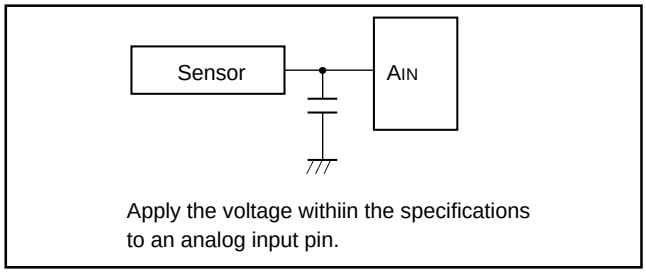


Fig. 69 Analog input external circuit example-1

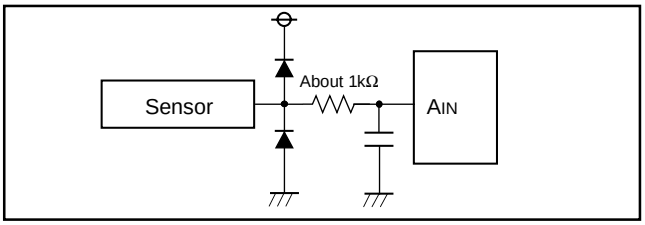


Fig. 70 Analog input external circuit example-2

⑱ POF instruction

When the POF instruction is executed continuously after the EPOF instruction, system enters the RAM back-up state.
Note that system cannot enter the RAM back-up state when executing only the POF instruction.
Be sure to disable interrupts by executing the DI instruction before executing the EPOF instruction and the POF instruction continuously.

⑲ Program counter

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

⑳ Power-on reset

When the built-in power-on reset circuit is used, the time for the supply voltage to rise from 0 V to the value of supply voltage or more must be set to 100 µs or less. If the rising time exceeds 100 µs, connect a capacitor between the $\overline{\text{RESET}}$ pin and V_{SS} at the shortest distance, and input "L" level to $\overline{\text{RESET}}$ pin until the value of supply voltage reaches the minimum operating voltage.

㉑ Clock control

Execute the main clock (f(X_{IN})) selection instruction (CMCK, CRCK or CYCK instruction) in the initial setting routine of program (executing it in address 0 in page 0 is recommended).
The oscillation circuit by the CMCK, CRCK or CYCK instruction can be selected only at once. The oscillation circuit corresponding to the first executed one of these instructions is valid.
The CMCK, CRCK, and CYCK instructions can be used only to select main clock (f(X_{IN})). In this time, the start of oscillation and the switch of system clock are not performed.
When the CMCK, CRCK, and CYCK instructions are never executed, main clock (f(X_{IN})) cannot be used and system can be operated only by on-chip oscillator.
The no operated clock source (f(RING)) or (f(X_{IN})) cannot be used for the system clock. Also, the clock source (f(RING) or f(X_{IN})) selected for the system clock cannot be stopped.

㉒ On-chip oscillator

The clock frequency of the on-chip oscillator depends on the supply voltage and the operation temperature range.
Be careful that variable frequencies when designing application products.
When considering the oscillation stabilize wait time at the switch of clock, be careful that the margin of frequencies of the on-chip oscillator clock.

㉓ External clock

When the external clock signal for the main clock (f(XIN)) is used, connect the clock source to XIN pin and XOUT pin open. In program, after the CMCK instruction is executed, set main clock (f(XIN)) oscillation start to be enabled (MR1=0).

For this product, when RAM back-up mode and main clock (f(XIN)) stop (MR1=1), XIN pin is fixed to "H" in order to avoid the through current by floating of internal logic. The XIN pin is fixed to "H" until main clock (f(XIN)) oscillation start to be valid (MR1=0) by the CMCK instruction from reset state. Accordingly, when an external clock is used, connect a 1 kΩ or more resistor to XIN pin in series to limit of current by competitive signal.

㉔ Electric Characteristic Differences Between Mask ROM and One Time PROM Version MCU

There are differences in electric characteristics, operation margin, noise immunity, and noise radiation between Mask ROM and One Time PROM version MCUs due to the difference in the manufacturing processes.

When manufacturing an application system with the One time PROM version and then switching to use of the Mask ROM version, please perform sufficient evaluations for the commercial samples of the Mask ROM version.

㉕ Note on Power Source Voltage

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

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

CONTROL REGISTERS

Interrupt control register V1		at reset : 0000_2		at RAM back-up : 0000_2	R/W TAV1/TV1A
$V1_3$	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 TAV2/TV2A
$V2_3$	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 TAI1/TI1A
$I1_3$	INT0 pin input control bit (Note 2)	0	INT0 pin input disabled		
		1	INT0 pin input enabled		
$I1_2$	Interrupt valid waveform for INT0 pin/ return level selection bit (Note 2)	0	Falling waveform/“L” level (“L” level is recognized with the SNZI0 instruction)		
		1	Rising waveform/“H” level (“H” level is recognized with the SNZI0 instruction)		
$I1_1$	INT0 pin edge detection circuit control bit	0	One-sided edge detected		
		1	Both edges detected		
$I1_0$	INT0 pin Timer 1 count start synchronous circuit selection bit	0	Timer 1 count start synchronous circuit not selected		
		1	Timer 1 count start synchronous circuit selected		

Interrupt control register I2		at reset : 0000_2		at RAM back-up : state retained	R/W TAI2/TI2A
$I2_3$	INT1 pin input control bit (Note 2)	0	INT1 pin input disabled		
		1	INT1 pin input enabled		
$I2_2$	Interrupt valid waveform for INT1 pin/ return level selection bit (Note 2)	0	Falling waveform/“L” level (“L” level is recognized with the SNZI1 instruction)		
		1	Rising waveform/“H” level (“H” level is recognized with the SNZI1 instruction)		
$I2_1$	INT1 pin edge detection circuit control bit	0	One-sided edge detected		
		1	Both edges detected		
$I2_0$	INT1 pin Timer 3 count start synchronous circuit selection bit	0	Timer 3 count start synchronous circuit not selected		
		1	Timer 3 count start synchronous circuit selected		

Notes 1: “R” represents read enabled, and “W” represents write enabled.
2: When the contents of $I1_2$, $I1_3$ $I2_2$ and $I2_3$ are changed, the external interrupt request flag (EXF0, EXF1) may be set to “1”.

Clock control register MR		at reset : 1111_2		at RAM back-up : 1111_2	R/W TAMR/ TMRA
MR_3	Operation mode selection bits	MR_3	MR_2	Operation mode	
		0	0	Through mode (frequency not divided)	
MR_2		0	1	Frequency divided by 2 mode	
		1	0	Frequency divided by 4 mode	
		1	1	Frequency divided by 8 mode	
MR_1	Main clock f(X_{IN}) oscillation circuit control bit	0		Main clock (f(X_{IN})) oscillation enabled	
		1		Main clock (f(X_{IN})) oscillation stop	
MR_0	System clock oscillation source selection bit	0		Main clock (f(X_{IN}))	
		1		Main clock (f(RING))	

Clock control register RG		at reset : 0_2	at RAM back-up : 0_2	W TRGA
RG_0	On-chip oscillator (f(RING)) control bit	0	On-chip oscillator (f(RING)) oscillation enabled	
		1	On-chip oscillator (f(RING)) oscillation stop	

Timer control register PA		at reset : 0_2	at RAM back-up : 0_2	W TPAA
PA_0	Prescaler control bit	0	Stop (state initialized)	
		1	Operating	

Timer control register W1		at reset : 0000_2		at RAM back-up : state retained	R/W TAW1/TW1A
$W1_3$	Timer 1 count auto-stop circuit selection bit (Note 2)	0		Timer 1 count auto-stop circuit not selected	
		1		Timer 1 count auto-stop circuit selected	
$W1_2$	Timer 1 control bit	0		Stop (state retained)	
		1		Operating	
$W1_1$	Timer 1 count source selection bits	$W1_1$	$W1_0$	Count source	
		0	0	Instruction clock (INSTCK)	
$W1_0$		0	1	Prescaler output (ORCLK)	
		1	0	X_{IN} input	
		1	1	CNTR0 input	

Timer control register W2		at reset : 0000_2		at RAM back-up : state retained	R/W TAW2/TW2A
$W2_3$	CNTR0 output signal selection bit	0		Timer 1 underflow signal divided by 2 output	
		1		Timer 2 underflow signal divided by 2 output	
$W2_2$	Timer 2 control bit	0		Stop (state retained)	
		1		Operating	
$W2_1$	Timer 2 count source selection bits	$W2_1$	$W2_0$	Count source	
		0	0	System clock (STCK)	
$W2_0$		0	1	Prescaler output (ORCLK)	
		1	0	Timer 1 underflow signal (T1UDF)	
		1	1	PWM signal (PWMOUT)	

Notes 1: “R” represents read enabled, and “W” represents write enabled.
2: This function is valid only when the timer 1 count start synchronous circuit is selected ($I1_0$=“1”).

Timer control register W3		at reset : 0000_2		at RAM back-up : state retained	R/W TAW3/TW3A
$W3_3$	Timer 3 count auto-stop circuit selection bit (Note 2)	0		Timer 3 count auto-stop circuit not selected	
		1		Timer 3 count auto-stop circuit selected	
$W3_2$	Timer 3 control bit	0		Stop (state retained)	
		1		Operating	
$W3_1$	Timer 3 count source selection bits	$W3_1$	$W3_0$	Count source	
		0	0	PWM signal (PWMOUT)	
$W3_0$		0	1	Prescaler output (ORCLK)	
		1	0	Timer 2 underflow signal (T2UDF)	
		1	1	CNTR1 input	

Timer control register W4		at reset : 0000_2		at RAM back-up : 0000_2	R/W TAW4/TW4A
$W4_3$	D_7/CNTR1 pin function selection bit	0		D_7 (I/O) / CNTR1 (input)	
		1		CNTR1 (I/O) / D_7 (input)	
$W4_2$	PWM signal “H” interval expansion function control bit	0		PWM signal “H” interval expansion function invalid	
		1		PWM signal “H” interval expansion function valid	
$W4_1$	Timer 4 control bit	0		Stop (state retained)	
		1		Operating	
$W4_0$	Timer 4 count source selection bit	0		X_{IN} input	
		1		Prescaler output (ORCLK) divided by 2	

Timer control register W5		at reset : 0000_2		at RAM back-up : state retained	R/W TAW5/TW5A
$W5_3$	Not used	0		This bit has no function, but read/write is enabled.	
		1			
$W5_2$	Period measurement circuit control bit	0		Stop	
		1		Operating	
$W5_1$	Signal for period measurement selection bits	$W5_1$	$W5_0$	Count source	
		0	0	On-chip oscillator (f(RING/16))	
$W5_0$		0	1	$CNTR_0$ pin input	
		1	0	INT0 pin input	
		1	1	Not available	

Timer control register W6		at reset : 0000_2	at RAM back-up : state retained	R/W TAW6/TW6A
$W6_3$	CNTR1 pin input count edge selection bit	0	Falling edge	
		1	Rising edge	
$W6_2$	CNTR0 pin input count edge selection bit	0	Falling edge	
		1	Rising edge	
$W6_1$	CNTR1 output auto-control circuit selection bit	0	CNTR1 output auto-control circuit not selected	
		1	CNTR1 output auto-control circuit selected	
$W6_0$	D_6/CNTR0 pin 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: This function is valid only when the timer 3 count start synchronous circuit is selected ($I2_0$=“1”).

Serial I/O control register J1		at reset : 0000$_2$		at RAM back-up : state retained	R/W TAJ1/TJ1A
J13	Serial I/O synchronous clock selection bits	J13	J12	Synchronous clock	
J12		0	0	Instruction clock (INSTCK) divided by 8	
		0	1	Instruction clock (INSTCK) divided by 4	
		1	0	Instruction clock (INSTCK) divided by 2	
		1	1	External clock (SCK input)	
J11	Serial I/O port function selection bits	J11	J10	Port function	
J10		0	0	P20, P21,P22 selected/SCK, SOUT, SIN not selected	
		0	1	SCK, SOUT, P22 selected/P20, P21, SIN not selected	
		1	0	SCK, P21, SIN selected/P20, SOUT, P22 not selected	
		1	1	SCK, SOUT, SIN selected/P20, P21,P22 not selected	

A/D control register Q1		at reset : 0000$_2$			at RAM back-up : state retained	R/W TAQ1/TQ1A
Q13	A/D operation mode selection bit	A/D conversion mode				
		Comparator mode				
Q12	Analog input pin selection bits	Q12	Q11	Q10	Analog input pins	
Q11		0	0	0	AIN0	
		0	0	1	AIN1	
Q10		0	1	0	AIN2	
		0	1	1	AIN3	
		1	0	0	AIN4	
		1	0	1	AIN5	
		1	1	0	AIN6	
		1	1	1	AIN7	

A/D control register Q2		at reset : 0000$_2$	at RAM back-up : state retained	R/W TAQ2/TQ2A
Q23	P40/AIN4, P41/AIN5, P42/AIN6, P43/AIN7 pin function selection bit	0	P40, P41, P42, P43	
		1	AIN4, AIN5, AIN6, AIN7	
Q22	P62/AIN2, P63/AIN3 pin function selection bit	0	P62, P63	
		1	AIN2, AIN3	
Q21	P61/AIN1 pin function selection bit	0	P61	
		1	AIN1	
Q20	P60/AIN0 pin function selection bit	0	P60	
		1	AIN0	

A/D control register Q3		at reset : 0000$_2$		at RAM back-up : state retained	R/W TAQ3/TQ3A
Q33	Not used	0	This bit has no function, but read/write is enabled.		
		1			
Q32	A/D converter operation clock selection bit	0	Instruction clock (INSTCK)		
		1	On-chip oscillator (f(RING))		
Q31	A/D converter operation clock division ratio selection bits	Q31	Q30	Division ratio	
Q30		0	0	Frequency divided by 6	
		0	1	Frequency divided by 12	
		1	0	Frequency divided by 24	
		1	1	Frequency divided by 48	

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

Key-on wakeup control register K0		at reset : 0000_2		at RAM back-up : state retained	R/W TAK0/TK0A
$K0_3$	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		
Key-on wakeup control register K1		at reset : 0000_2		at RAM back-up : state retained	R/W TAK1/TK1A
$K1_3$	Ports $P0_2$ and $P0_3$ return condition selection bit	0	Return by level		
		1	Return by edge		
$K1_2$	Ports $P0_2$ and $P0_3$ valid waveform/ level selection bit	0	Falling waveform/“L” level		
		1	Rising waveform/“H” level		
$K1_1$	Ports $P0_1$ and $P0_0$ return condition selection bit	0	Return by level		
		1	Return by edge		
$K1_0$	Ports $P0_1$ and $P0_0$ valid waveform/ level selection bit	0	Falling waveform/“L” level		
		1	Rising waveform/“H” level		
Key-on wakeup control register K2		at reset : 0000_2		at RAM back-up : state retained	R/W TAK2/TK2A
$K2_3$	INT1 pin return condition selection bit	0	Return by level		
		1	Return by edge		
$K2_2$	INT1 pin key-on wakeup contro bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K2_1$	INT0 pin return condition selection bit	0	Return by level		
		1	Return by edge		
$K2_0$	INT0 pin key-on wakeup contro bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

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

Pull-up control register PU0		at reset : 0000_2		at RAM back-up : state retained	R/W TAPU0/ TPU0A
$PU0_3$	$P0_3$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU0_2$	$P0_2$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU0_1$	$P0_1$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU0_0$	$P0_0$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
Pull-up control register PU1		at reset : 0000_2		at RAM back-up : state retained	R/W TAPU1/ TPU1A
$PU1_3$	$P1_3$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU1_2$	$P1_2$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU1_1$	$P1_1$ pin pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU1_0$	$P1_0$ pin 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.

Port output structure control register FR0		at reset : 0000_2		at RAM back-up : state retained	W TFR0A
$FR0_3$	Ports $P1_2$, $P1_3$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR0_2$	Ports $P1_0$, $P1_1$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR0_1$	Ports $P0_2$, $P0_3$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR0_0$	Ports $P0_0$, $P0_1$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		

Port output structure control register FR1		at reset : 0000_2		at RAM back-up : state retained	W TFR1A
$FR1_3$	Port D_3 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR1_2$	Port D_2 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR1_1$	Port D_1 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR1_0$	Port D_0 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		

Port output structure control register FR2		at reset : 0000_2		at RAM back-up : state retained	W TFR2A
$FR2_3$	Port D_7/CNTR1 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR2_2$	Port D_6/CNTR0 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR2_1$	Port D_5 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR2_0$	Port D_4 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		

Port output structure control register FR3		at reset : 0000_2		at RAM back-up : state retained	W TFR3A
$FR3_3$	Port $P5_3$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR3_2$	Port $P5_2$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR3_1$	Port $P5_1$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR3_0$	Port $P5_0$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		

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

INSTRUCTIONS

The 4519 Group has the 153 instructions. Each instruction is described as follows;

(1) Index list of instruction function
(2) Machine instructions (index by alphabet)
(3) Machine instructions (index by function)
(4) Instruction code table

SYMBOL

The symbols shown below are used in the following list of instruction function and the machine instructions.

Symbol	Contents	Symbol	Contents
A	Register A (4 bits)	PS	Prescaler
B	Register B (4 bits)	T1	Timer 1
DR	Register DR (3 bits)	T2	Timer 2
E	Register E (8 bits)	T3	Timer 3
V1	Interrupt control register V1 (4 bits)	T4	Timer 4
V2	Interrupt control register V2 (4 bits)	T1F	Timer 1 interrupt request flag
I1	Interrupt control register I1 (4 bits)	T2F	Timer 2 interrupt request flag
I2	Interrupt control register I2 (4 bits)	T3F	Timer 3 interrupt request flag
MR	Clock control register MR (4 bits)	T4F	Timer 4 interrupt request flag
RG	Clock control register RG (1 bit)	WDF1	Watchdog timer flag
PA	Timer control register PA (1 bit)	WEF	Watchdog timer enable flag
W1	Timer control register W1 (4 bits)	INTE	Interrupt enable flag
W2	Timer control register W2 (4 bits)	EXF0	External 0 interrupt request flag
W3	Timer control register W3 (4 bits)	EXF1	External 1 interrupt request flag
W4	Timer control register W4 (4 bits)	P	Power down flag
W5	Timer control register W5 (4 bits)	ADF	A/D conversion completion flag
W6	Timer control register W6 (4 bits)	SIOF	Serial I/O transmit/receive completion flag
J1	Serial I/O control register J1 (4 bits)		
Q1	A/D control register Q1 (4 bits)	D	Port D (8 bits)
Q2	A/D control register Q2 (4 bits)	P0	Port P0 (4 bits)
Q3	A/D control register Q3 (4 bits)	P1	Port P1 (4 bits)
PU0	Pull-up control register PU0 (4 bits)	P2	Port P2 (3 bits)
PU1	Pull-up control register PU1 (4 bits)	P3	Port P3 (4 bits)
FR0	Port output format control register FR0 (4 bits)	P4	Port P4 (4 bits)
FR1	Port output format control register FR1 (4 bits)	P5	Port P5 (4 bits)
FR2	Port output format control register FR2 (4 bits)	P6	Port P6 (4 bits)
FR3	Port output format control register FR3 (4 bits)		
K0	Key-on wakeup control register K0 (4 bits)	x	Hexadecimal variable
K1	Key-on wakeup control register K1 (4 bits)	y	Hexadecimal variable
K2	Key-on wakeup control register K2 (4 bits)	z	Hexadecimal variable
X	Register X (4 bits)	p	Hexadecimal variable
Y	Register Y (4 bits)	n	Hexadecimal constant
Z	Register Z (2 bits)	i	Hexadecimal constant
DP	Data pointer (10 bits)	j	Hexadecimal constant
	(It consists of registers X, Y, and Z)	$A_3A_2A_1A_0$	Binary notation of hexadecimal variable A
PC	Program counter (14 bits)		(same for others)
PC_H	High-order 7 bits of program counter		
PC_L	Low-order 7 bits of program counter	←	Direction of data movement
SK	Stack register (14 bits × 8)	↔	Data exchange between a register and memory
SP	Stack pointer (3 bits)	?	Decision of state shown before “?”
CY	Carry flag	( )	Contents of registers and memories
RPS	Prescaler reload register (8 bits)	—	Negate, Flag unchanged after executing instruction
R1	Timer 1 reload register (8 bits)	M(DP)	RAM address pointed by the data pointer
R2	Timer 2 reload register (8 bits)	a	Label indicating address a_6 a_5 a_4 a_3 a_2 a_1 a_0
R3	Timer 3 reload register (8 bits)	p, a	Label indicating address a_6 a_5 a_4 a_3 a_2 a_1 a_0
R4L	Timer 4 reload register (8 bits)		in page p_5 p_4 p_3 p_2 p_1 p_0
R4H	Timer 4 reload register (8 bits)	C + x	Hex. C + Hex. number x

Note : Some instructions of the 4519 Group has the skip function to unexecute the next described instruction. The 4519 Group just invalidates the next instruction when a skip is performed. The contents of program counter is not increased by 2. Accordingly, the number of cycles does not change even if skip is not performed. However, the cycle count becomes “1” if the TABP p, RT, or RTS instruction is skipped.

INDEX LIST OF INSTRUCTION FUNCTION

Group-ing	Mnemonic	Function	Page
Register to register transfer	TAB	(A) ← (B)	110, 130
	TBA	(B) ← (A)	119, 130
	TAY	(A) ← (Y)	119, 130
	TYA	(Y) ← (A)	128, 130
	TEAB	(E_7–E_4) ← (B) (E_3–E_0) ← (A)	120, 130
	TABE	(B) ← (E_7–E_4) (A) ← (E_3–E_0)	111, 130
	TDA	(DR_2–DR_0) ← (A_2–A_0)	119, 130
	TAD	(A_2–A_0) ← (DR_2–DR_0) (A_3) ← 0	112, 130
	TAZ	(A_1, A_0) ← (Z_1, Z_0) (A_3, A_2) ← 0	119, 130
	TAX	(A) ← (X)	118, 130
	TASP	(A_2–A_0) ← (SP_2–SP_0) (A_3) ← 0	116, 130
RAM addresses	LXY x, y	(X) ← x x = 0 to 15 (Y) ← y y = 0 to 15	98, 130
	LZ z	(Z) ← z z = 0 to 3	98, 130
	INY	(Y) ← (Y) + 1	98, 130
	DEY	(Y) ← (Y) – 1	95, 130
RAM to register transfer	TAM j	(A) ← (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15	114, 130
	XAM j	(A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15	128, 130
	XAMD j	(A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) – 1	128, 130
RAM to register transfer	XAMI j	(A) ← → (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) + 1	129, 130
	TMA j	(M(DP)) ← (A) (X) ← (X)EXOR(j) j = 0 to 15	122, 130
Arithmetic operation	LA n	(A) ← n n = 0 to 15	98, 132
	TABP p	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p (PC_L) ← (DR_2–DR_0, A_3–A_0) (DR_2) ← 0 (DR_1, DR_0) ← (ROM(PC))$_{9, 8}$ (B) ← (ROM(PC))$_{7–4}$ (A) ← (ROM(PC))$_{3–0}$ (PC) ← (SK(SP)) (SP) ← (SP) – 1	111, 132
	AM	(A) ← (A) + (M(DP))	91, 132
	AMC	(A) ← (A) + (M(DP)) + (CY) (CY) ← Carry	91, 132
	A n	(A) ← (A) + n n = 0 to 15	91, 132
	AND	(A) ← (A) AND (M(DP))	92, 132
	OR	(A) ← (A) OR (M(DP))	101, 132
	SC	(CY) ← 1	103, 132
	RC	(CY) ← 0	102, 132
	SZC	(CY) = 0 ?	108, 132
	CMA	(A) ← ($\overline{A}$)	94, 132
	RAR	→ CY → $A_3A_2A_1A_0$	101, 132

Note: p is 0 to 47 for M34519M6,
p is 0 to 63 for M34519M8/E8.

INDEX LIST OF INSTRUCTION FUNCTION (continued)

Group-ing	Mnemonic	Function	Page
Bit operation	SB j	(Mj(DP)) ← 1 j = 0 to 3	103, 132
	RB j	(Mj(DP)) ← 0 j = 0 to 3	101, 132
	SZB j	(Mj(DP)) = 0 ? j = 0 to 3	107, 132
Comparison operation	SEAM	(A) = (M(DP)) ?	104, 132
	SEA n	(A) = n ? n = 0 to 15	104, 132
Branch operation	B a	(PCL) ← a6–a0	92, 134
	BL p, a	(PCH) ← p (PCL) ← a6–a0	92, 134
	BLA p	(PCH) ← p (PCL) ← (DR2–DR0, A3–A0)	92, 134
Subroutine operation	BM a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← 2 (PCL) ← a6–a0	93, 134
	BML p, a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← p (PCL) ← a6–a0	93, 134
	BMLA p	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← p (PCL) ← (DR2–DR0, A3–A0)	93, 134
Return operation	RTI	(PC) ← (SK(SP)) (SP) ← (SP) – 1	102, 134
	RT	(PC) ← (SK(SP)) (SP) ← (SP) – 1	102, 134
	RTS	(PC) ← (SK(SP)) (SP) ← (SP) – 1	103, 134

Group-ing	Mnemonic	Function	Page
Interrupt operation	DI	(INTE) ← 0	95, 136
	EI	(INTE) ← 1	95, 136
	SNZ0	V10 = 0: (EXF0) = 1 ? After skipping, (EXF0) ← 0 V10 = 1: NOP	104, 136
	SNZ1	V11 = 0: (EXF1) = 1 ? After skipping, (EXF1) ← 0 V11 = 1: NOP	104, 136
	SNZI0	I12 = 1 : (INT0) = “H” ? I12 = 0 : (INT0) = “L” ?	105, 136
	SNZI1	I22 = 1 : (INT1) = “H” ? I22 = 0 : (INT1) = “L” ?	105, 136
	TAV1	(A) ← (V1)	116, 136
	TV1A	(V1) ← (A)	126, 136
	TAV2	(A) ← (V2)	117, 136
	TV2A	(V2) ← (A)	126, 136
	TAI1	(A) ← (I1)	113, 136
	TI1A	(I1) ← (A)	121, 136
	TAI2	(A) ← (I2)	113, 136
	TI2A	(I2) ← (A)	121, 136
Timer operation	TPAA	(PA0) ← (A0)	123, 136
	TAW1	(A) ← (W1)	117, 136
	TW1A	(W1) ← (A)	126, 136
	TAW2	(A) ← (W2)	117, 136
	TW2A	(W2) ← (A)	126, 136
	TAW3	(A) ← (W3)	117, 136
	TW3A	(W3) ← (A)	127, 136

Note: p is 0 to 47 for M34519M6,
p is 0 to 63 for M34519M8/E8.

INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function	Page
Timer operation	TAW4	(A) ← (W4)	118, 136
	TW4A	(W4) ← (A)	127, 136
	TAW5	(A) ← (W5)	118, 138
	TW5A	(W5) ← (A)	127, 138
	TAW6	(A) ← (W6)	118, 138
	TW6A	(W6) ← (A)	127, 138
	TABPS	(B) ← (TPS_7–TPS_4) (A) ← (TPS_3–TPS_0)	112, 138
	TPSAB	(RPS_7–RPS_4) ← (B) (TPS_7–TPS_4) ← (B) (RPS_3–RPS_0) ← (A) (TPS_3–TPS_0) ← (A)	123, 138
	TAB1	(B) ← ($T1_7$–$T1_4$) (A) ← ($T1_3$–$T1_0$)	110, 138
	T1AB	($R1_7$–$R1_4$) ← (B) ($T1_7$–$T1_4$) ← (B) ($R1_3$–$R1_0$) ← (A) ($T1_3$–$T1_0$) ← (A)	108, 138
	TAB2	(B) ← ($T2_7$–$T2_4$) (A) ← ($T2_3$–$T2_0$)	110, 138
	T2AB	($R2_7$–$R2_4$) ← (B) ($T2_7$–$T2_4$) ← (B) ($R2_3$–$R2_0$) ← (A) ($T2_3$–$T2_0$) ← (A)	108, 138
	TAB3	(B) ← ($T3_7$–$T3_4$) (A) ← ($T3_3$–$T3_0$)	110, 138
	T3AB	($R3_7$–$R3_4$) ← (B) ($T3_7$–$T3_4$) ← (B) ($R3_3$–$R3_0$) ← (A) ($T3_3$–$T3_0$) ← (A)	109, 138
	TAB4	(B) ← ($T4_7$–$T4_4$) (A) ← ($T4_3$–$T4_0$)	111, 138
	T4AB	($R4L_7$–$R4L_4$) ← (B) ($T4_7$–$T4_4$) ← (B) ($R4L_3$–$R4L_0$) ← (A) ($T4_3$–$T4_0$) ← (A)	109, 138
Timer operation	T4HAB	($R4H_7$–$R4H_4$) ← (B) ($R4H_3$–$R4H_0$) ← (A)	109, 138
	TR1AB	($R1_7$–$R1_4$) ← (B) ($R1_3$–$R1_0$) ← (A)	125, 138
	TR3AB	($R3_7$–$R3_4$) ← (B) ($R3_3$–$R3_0$) ← (A)	125, 138
	T4R4L	($T4_7$–$T4_4$) ← ($R4L_7$–$R4L_4$)	109, 140
	SNZT1	$V1_2$ = 0: (T1F) = 1 ? After skipping, (T1F) ← 0 $V1_2$ = 1: NOP	106, 140
	SNZT2	$V1_3$ = 0: (T2F) = 1 ? After skipping, (T2F) ← 0 $V1_3$ = 1: NOP	106, 140
	SNZT3	$V2_0$ = 0: (T3F) = 1 ? After skipping, (T3F) ← 0 $V2_0$ = 1: NOP	106, 140
	SNZT4	$V2_1$ = 0: (T4F) = 1 ? After skipping, (T4F) ← 0 $V2_1$ = 1: NOP	107, 140
Input/Output operation	IAP0	(A) ← (P0)	96, 140
	OP0A	(P0) ← (A)	99, 140
	IAP1	(A) ← (P1)	96, 140
	OP1A	(P1) ← (A)	99, 140
	IAP2	(A_2–A_0) ← ($P2_2$–$P2_0$) (A_3) ← 0	96, 140
	OP2A	($P2_2$–$P2_0$) ← (A_2–A_0)	99, 140
	IAP3	(A) ← (P3)	97, 140
	OP3A	(P3) ← (A)	100, 140
	IAP4	(A) ← (P4)	97, 140
	OP4A	(P4) ← (A)	100, 140
	IAP5	(A) ← (P5)	97, 140
	OP5A	(P5) ← (A)	100, 140
	IAP6	(A) ← (P6)	97, 140
	OP6A	(P6) ← (A)	100, 140

INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function	Page
Input/Output operation	CLD	(D) ← 1	93, 140
	RD	(D(Y)) ← 0 (Y) = 0 to 7	102, 140
	SD	(D(Y)) ← 1 (Y) = 0 to 7	103, 140
	SZD	(D(Y)) = 0 ? (Y) = 0 to 7	108, 140
	TAPU0	(A) ← (PU0)	115, 140
	TPU0A	(PU0) ← (A)	123, 140
	TAPU1	(A) ← (PU1)	115, 140
	TPU1A	(PU1) ← (A)	124, 140
	TAK0	(A) ← (K0)	113, 142
	TK0A	(K0) ← (A)	122, 142
	TAK1	(A) ← (K1)	114, 142
	TK1A	(K1) ← (A)	122, 142
	TAK2	(A) ← (K2)	114, 142
	TK2A	(K2) ← (A)	122, 142
	TFR0A	(FR0) ← (A)	120, 142
	TFR1A	(FR1) ← (A)	120, 142
	TFR2A	(FR2) ← (A)	120, 142
	TFR3A	(FR3) ← (A)	121, 142
Clock operation	CMCK	Ceramic resonator selected	94, 142
	CRCK	RC oscillator selected	94, 142
	CYCK	Quartz-crystal oscillator selected	94, 142
	TRGA	(RG_0) ← (A_0)	125, 142
	TAMR	(A) ← (MR)	110, 142
	TMRA	(MR) ← (A)	123, 142
Serial I/O operation	TABSI	(B) ← (SI_7–SI_4) (A) ← (SI_3–SI_0)	112, 142
	TSIAB	(SI_7–SI_4) ← (B) (SI_3–SI_0) ← (A)	125, 142
	SST	(SIOF) ← 0 Serial I/O starting	107, 142
	SNZSI	$V2_3$=0: (SIOF)=1? After skipping, (SIOF) ← 0 $V2_3$=1: NOP	106, 142
	TAJ1	(A) ← (J1)	113, 142
	TJ1A	(J1) ← (A)	121, 142
A/D operation	TABAD	In A/D conversion mode , (B) ← (AD_9–AD_6) (A) ← (AD_5–AD_2) In comparator mode, (B) ← (AD_7–AD_4) (A) ← (AD_3–AD_0)	111, 144
	TALA	(A_3, A_2) ← (AD_1, AD_0) (A_1, A_0) ← 0	114, 144
	TADAB	(AD_7–AD_4) ← (B) (AD_3–AD_0) ← (A)	112, 144
	ADST	(ADF) ← 0 A/D conversion starting	91, 144
	SNZAD	$V2_1$ = 0: (ADF) = 1 ? After skipping, (ADF) ← 0 $V2_1$=1: NOP	105, 144
	TAQ1	(A) ← (Q1)	115, 144
	TQ1A	(Q1) ← (A)	124, 144
	TAQ2	(A) ← (Q2)	116, 144
	TQ2A	(Q2) ← (A)	124, 144
	TAQ3	(A) ← (Q3)	116, 144
	TQ3A	(Q3) ← (A)	124, 144

INDEX LIST OF INSTRUCTION FUNCTION (continued)

Group-ing	Mnemonic	Function	Page
Other operation	NOP	(PC) ← (PC) + 1	99, 144
	POF	Transition to RAM back-up mode	101, 144
	EPOF	POF instruction valid	96, 144
	SNZP	(P) = 1 ?	105, 144
	DWDT	Stop of watchdog timer function enabled	95, 144
	WRST	(WDF1) = 1 ? After skipping, (WDF1) ← 0	128, 144
	SRST	System reset occurrence	107, 144

MACHINE INSTRUCTIONS (INDEX BY ALPHABET)

A n (Add n and accumulator)

Instruction code

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

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

Operation: (A) ← (A) + n
n = 0 to 15

Grouping: Arithmetic operation

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

ADST (A/D conversion STart)

Instruction code

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

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

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

Grouping: A/D conversion operation

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

AM (Add accumulator and Memory)

Instruction code

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

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

Operation: (A) ← (A) + (M(DP))

Grouping: Arithmetic operation

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

AMC (Add accumulator, Memory and Carry)

Instruction code

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

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

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

Grouping: Arithmetic operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

AND (logical AND between accumulator and memory)

Instruction code

D9									D0				
0	0	0	0	0	1	1	0	0	0	$_2$	0	1	8	$_{16}$

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

Operation: (A) ← (A) AND (M(DP))

Grouping: Arithmetic operation

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

B a (Branch to address a)

Instruction code

D9									D0				
0	1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$	1	8+a	a	$_{16}$

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

Operation: (PC_L) ← a_6 to a_0

Grouping: Branch operation

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

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

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

Instruction code

D9									D0				
0	0	1	1	1	p_4	p_3	p_2	p_1	p_0	$_2$	0	E+p	p	$_{16}$
1	0	p_5	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$	2	p+a	a	$_{16}$

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

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

Grouping: Branch operation

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

Note: p is 0 to 47 for M34519M6 and p is 0 to 63 for M34519M8E8.

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

Instruction code

D9									D0				
0	0	0	0	0	1	0	0	0	0	$_2$	0	1	0	$_{16}$
1	0	p_5	p_4	0	0	p_3	p_2	p_1	p_0	$_2$	2	p	p	$_{16}$

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

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

Grouping: Branch operation

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

Note: p is 0 to 47 for M34519M6 and p is 0 to 63 for M34519M8E8.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

BM a (Branch and Mark to address a in page 2)

Instruction code

D9									D0				
0	1	0	a6	a5	a4	a3	a2	a1	a0	$_2$	1	a	a	$_{16}$

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

Operation:
(SP) ← (SP) + 1
(SK(SP)) ← (PC)
(PCH) ← 2
(PCL) ← a6–a0

Grouping: Subroutine call operation

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

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

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

Instruction code

D9									D0				
0	0	1	1	0	p4	p3	p2	p1	p0	$_2$	0	C +p	p	$_{16}$
1	0	p5	a6	a5	a4	a3	a2	a1	a0	$_2$	2	p +a	a	$_{16}$

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

Operation:
(SP) ← (SP) + 1
(SK(SP)) ← (PC)
(PCH) ← p
(PCL) ← a6–a0

Grouping: Subroutine call operation

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

Note: p is 0 to 47 for M34519M6 and p is 0 to 63 for M34519M8E8.
Be careful not to over the stack because the maximum level of subroutine nesting is 8.

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

Instruction code

D9									D0				
0	0	0	0	1	1	0	0	0	0	$_2$	0	3	0	$_{16}$
1	0	p5	p4	0	0	p3	p2	p1	p0	$_2$	2	p	p	$_{16}$

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

Operation:
(SP) ← (SP) + 1
(SK(SP)) ← (PC)
(PCH) ← p
(PCL) ← (DR2–DR0, A3–A0)

Grouping: Subroutine call operation

Description: Call the subroutine : Calls the subroutine at address (DR2 DR1 DR0 A3 A2 A1 A0)2 specified by registers D and A in page p.

Note: p is 0 to 47 for M34519M6 and p is 0 to 63 for M34519M8E8.
Be careful not to over the stack because the maximum level of subroutine nesting is 8.

CLD (CLear port D)

Instruction code

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

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

Operation: (D) ← 1

Grouping: Input/Output operation

Description: Sets (1) to port D.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

CMA (CoMplement of Accumulator)

Instruction code

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

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

Operation: (A) ← $\overline{(A)}$

Grouping: Arithmetic operation

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

CMCK (Clock select: ceraMic oscillation ClocK)

Instruction code

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

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

Operation: Ceramic oscillation circuit selected

Grouping: Clock control operation

Description: Selects the ceramic oscillation circuit for main clock f(XIN).

CRCK (Clock select: Rc oscillation ClocK)

Instruction code

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

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

Operation: RC oscillation circuit selected

Grouping: Clock control operation

Description: Selects the RC oscillation circuit for main clock f(XIN).

CYCK (Clock select: crYstal oscillation ClocK)

Instruction code

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

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

Operation: Quartz-crystal oscillation circuit selected

Grouping: Clock control operation

Description: Selects the quartz-crystal oscillation circuit for main clock f(XIN).

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

DEY (DEcrement register Y)

Instruction code

D9									D0			
0	0	0	0	0	1	0	1	1	1	0	1	7
$_2$										$_{16}$		

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

Operation: (Y) ← (Y) – 1

Grouping: RAM addresses

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

DI (Disable Interrupt)

Instruction code

D9									D0			
0	0	0	0	0	0	0	1	0	0	0	0	4
$_2$										$_{16}$		

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

Operation: (INTE) ← 0

Grouping: Interrupt control operation

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

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

DWDT (Disable WatchDog Timer)

Instruction code

D9									D0			
1	0	1	0	0	1	1	1	0	0	2	9	C
$_2$										$_{16}$		

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

Operation: Stop of watchdog timer function enabled

Grouping: Other operation

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

EI (Enable Interrupt)

Instruction code

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

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

Operation: (INTE) ← 1

Grouping: Interrupt control operation

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

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

EPOF (Enable POF instruction)

Instruction code

D9									D0				
0	0	0	1	0	1	1	0	1	1	2	0 5 B		16

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

Operation: POF instruction valid

Grouping: Other operation

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

IAP0 (Input Accumulator from port P0)

Instruction code

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

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

Operation: (A) ← (P0)

Grouping: Input/Output operation

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

IAP1 (Input Accumulator from port P1)

Instruction code

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

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

Operation: (A) ← (P1)

Grouping: Input/Output operation

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

IAP2 (Input Accumulator from port P2)

Instruction code

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

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

Operation: (A_2–A_0) ← ($P2_2$–$P2_0$)
(A_3) ← 0

Grouping: Input/Output operation

Description: Transfers the input of port P2 to register A.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

IAP3 (Input Accumulator from port P3)

Instruction code

D9									D0				
1	0	0	1	1	0	0	0	1	1	$_2$	2 6 3	$_{16}$	

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

Operation: (A) ← (P3)

Grouping: Input/Output operation

Description: Transfers the input of port P3 to register A.

IAP4 (Input Accumulator from port P4)

Instruction code

D9									D0				
1	0	0	1	1	0	0	1	0	0	$_2$	2 6 4	$_{16}$	

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

Operation: (A) ← (P4)

Grouping: Input/Output operation

Description: Transfers the input of port P4 to register A.

IAP5 (Input Accumulator from port P5)

Instruction code

D9									D0				
1	0	0	1	1	0	0	1	0	1	$_2$	2 6 5	$_{16}$	

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

Operation: (A) ← (P5)

Grouping: Input/Output operation

Description: Transfers the input of port P5 to register A.

IAP6 (Input Accumulator from port P6)

Instruction code

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

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

Operation: (A) ← (P6)

Grouping: Input/Output operation

Description: Transfers the input of port P6 to register A.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

INY (INcrement register Y)

Instruction code

D9									D0				
0	0	0	0	0	1	0	0	1	1	$_2$	0 1 3	$_{16}$	

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

Operation: (Y) ← (Y) + 1

Grouping: RAM addresses

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

LA n (Load n in Accumulator)

Instruction code

D9									D0				
0	0	0	1	1	1	n	n	n	n	$_2$	0 7 n	$_{16}$	

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

Operation: (A) ← n
n = 0 to 15

Grouping: Arithmetic operation

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

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

Instruction code

D9									D0				
1	1	x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0	$_2$	3 x y	$_{16}$	

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

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

Grouping: RAM addresses

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

LZ z (Load register Z with z)

Instruction code

D9									D0				
0	0	0	1	0	0	1	0	z_1	z_0	$_2$	0 4 8+z	$_{16}$	

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

Operation: (Z) ← z z = 0 to 3

Grouping: RAM addresses

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

NOP (No OPeration)

Instruction code

D9									D0				
0	0	0	0	0	0	0	0	0	0	$_2$	0	0	0 $_{16}$

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

Operation: (PC) ← (PC) + 1

Grouping: Other operation

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

OP0A (Output port P0 from Accumulator)

Instruction code

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

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

Operation: (P0) ← (A)

Grouping: Input/Output operation

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

OP1A (Output port P1 from Accumulator)

Instruction code

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

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

Operation: (P1) ← (A)

Grouping: Input/Output operation

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

OP2A (Output port P2 from Accumulator)

Instruction code

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

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

Operation: (P2) ← (A)

Grouping: Input/Output operation

Description: Outputs the contents of register A to port P2.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

OP3A (Output port P3 from Accumulator)

Instruction code

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

$_2$ $_{16}$

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

Operation: (P3) ← (A)

Grouping: Input/Output operation

Description: Outputs the contents of register A to port P3.

OP4A (Output port P4 from Accumulator)

Instruction code

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

$_2$ $_{16}$

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

Operation: (P4) ← (A)

Grouping: Input/Output operation

Description: Outputs the contents of register A to port P4.

OP5A (Output port P5 from Accumulator)

Instruction code

D9									D0			
1	0	0	0	1	0	0	1	0	1	2	2	5

$_2$ $_{16}$

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

Operation: (P5) ← (A)

Grouping: Input/Output operation

Description: Outputs the contents of register A to port P5.

OP6A (Output port P6 from Accumulator)

Instruction code

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

$_2$ $_{16}$

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

Operation: (P6) ← (A)

Grouping: Input/Output operation

Description: Outputs the contents of register A to port P6.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

OR (logical OR between accumulator and memory)

Instruction code

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

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

Operation: (A) ← (A) OR (M(DP))

Grouping: Arithmetic operation

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

POF (Power OFf)

Instruction code

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

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

Operation: Transition to RAM back-up mode

Grouping: Other operation

Description: Puts the system in RAM back-up state by executing the POF instruction after executing the EPOF instruction.

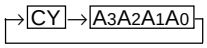
Note: If the EPOF instruction is not executed before executing this instruction, this instruction is equivalent to the NOP instruction.

RAR (Rotate Accumulator Right)

Instruction code

D9									D0				
0	0	0	0	0	1	1	1	0	1	2	0 1 D		16

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

Operation: 

Grouping: Arithmetic operation

Description: Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right.

RB j (Reset Bit)

Instruction code

D9									D0				
0	0	0	1	0	0	1	1	j	j	2	0 4 C+j		16

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

Operation: (Mj(DP)) ← 0
j = 0 to 3

Grouping: Bit operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

RC (Reset Carry flag)

Instruction code

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

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

Operation: (CY) ← 0

Grouping: Arithmetic operation

Description: Clears (0) to carry flag CY.

RD (Reset port D specified by register Y)

Instruction code

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

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

Operation: (D(Y)) ← 0
However,
(Y) = 0 to 7

Grouping: Input/Output operation

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

RT (ReTurn from subroutine)

Instruction code

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

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

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

Grouping: Return operation

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

RTI (ReTurn from Interrupt)

Instruction code

D9									D0				
0	0	0	1	0	0	0	1	1	0	2	0	4	6	16

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

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

Grouping: Return operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

RTS (ReTurn from subroutine and Skip)

Instruction code

D9									D0				
0	0	0	1	0	0	0	1	0	1	$_2$	0 4 5	$_{16}$	

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

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

Grouping: Return operation

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

SB j (Set Bit)

Instruction code

D9									D0				
0	0	0	1	0	1	1	1	j	j	$_2$	0 5 C+j	$_{16}$	

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

Operation: (Mj(DP)) ← 1
j = 0 to 3

Grouping: Bit operation

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

SC (Set Carry flag)

Instruction code

D9									D0				
0	0	0	0	0	0	0	1	1	1	$_2$	0 0 7	$_{16}$	

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

Operation: (CY) ← 1

Grouping: Arithmetic operation

Description: Sets (1) to carry flag CY.

SD (Set port D specified by register Y)

Instruction code

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

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

Operation: (D(Y)) ← 1
(Y) = 0 to 7

Grouping: Input/Output operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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

Instruction code

D9									D0				
0	0	0	0	1	0	0	1	0	1	$_2$	0 2 5	$_{16}$	
0	0	0	1	1	1	n	n	n	n	$_2$	0 7 n	$_{16}$	

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

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

Grouping: Comparison operation

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

SEAM (Skip Equal, Accumulator with Memory)

Instruction code

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

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

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

Grouping: Comparison operation

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

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

Instruction code

D9									D0				
0	0	0	0	1	1	1	0	0	0	$_2$	0 3 8	$_{16}$	

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

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

Grouping: Interrupt operation

Description: When $V1_0$ = 0 : Skips the next instruction when external 0 interrupt request flag EXF0 is “1.” After skipping, clears (0) to the EXF0 flag. When the EXF0 flag is “0,” executes the next instruction.
When $V1_0$ = 1 : This instruction is equivalent to the NOP instruction.

SNZ1 (Skip if Non Zero condition of external 1 interrupt request flag)

Instruction code

D9									D0				
0	0	0	0	1	1	1	0	0	1	$_2$	0 3 9	$_{16}$	

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	$V1_1$ = 0: (EXF1) = 1

Operation: $V1_1$ = 0: (EXF1) = 1 ?
After skipping, (EXF1) ← 0
$V1_1$ = 1: SNZ1 = NOP
($V1_1$: bit 1 of the interrupt control register V1)

Grouping: Interrupt operation

Description: When $V1_1$ = 0 : Skips the next instruction when external 1 interrupt request flag EXF1 is “1.” After skipping, clears (0) to the EXF1 flag. When the EXF1 flag is “0,” executes the next instruction.
When $V1_1$ = 1 : This instruction is equivalent to the NOP instruction.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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

Instruction code

D9									D0			
1	0	1	0	0	0	0	1	1	1	2	8	7

$_2$ $_{16}$

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

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

Grouping: A/D conversion operation

Description: When $V2_2$ = 0 : Skips the next instruction when A/D conversion completion flag ADF is “1.” After skipping, clears (0) to the ADF flag. When the ADF flag is “0,” executes the next instruction.
When $V2_2$ = 1 : This instruction is equivalent to the NOP instruction.

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

Instruction code

D9									D0			
0	0	0	0	1	1	1	0	1	0	0	3	A

$_2$ $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	$I1_2$ = 0 : (INT0) = “L” $I1_2$ = 1 : (INT0) = “H”

Operation:
$I1_2$ = 0 : (INT0) = “L” ?
$I1_2$ = 1 : (INT0) = “H” ?
($I1_2$: bit 2 of the interrupt control register I1)

Grouping: Interrupt operation

Description: When $I1_2$ = 0 : Skips the next instruction when the level of INT0 pin is “L.” Executes the next instruction when the level of INT0 pin is “H.”
When $I1_2$ = 1 : Skips the next instruction when the level of INT0 pin is “H.” Executes the next instruction when the level of INT0 pin is “L.”

SNZI1 (Skip if Non Zero condition of external 1 Interrupt input pin)

Instruction code

D9									D0			
0	0	0	0	1	1	1	0	1	1	0	3	B

$_2$ $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	$I2_2$ = 0 : (INT1) = “L” $I2_2$ = 1 : (INT1) = “H”

Operation:
$I2_2$ = 0 : (INT1) = “L” ?
$I2_2$ = 1 : (INT1) = “H” ?
($I2_2$: bit 2 of the interrupt control register I2)

Grouping: Interrupt operation

Description: When $I2_2$ = 0 : Skips the next instruction when the level of INT1 pin is “L.” Executes the next instruction when the level of INT1 pin is “H.”
When $I2_2$ = 1 : Skips the next instruction when the level of INT1 pin is “H.” Executes the next instruction when the level of INT1 pin is “L.”

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

Instruction code

D9									D0			
0	0	0	0	0	0	0	0	1	1	0	0	3

$_2$ $_{16}$

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

Operation:
(P) = 1 ?

Grouping: Other operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

SNZSI (Skip if Non Zero condition of Serial I/o interrupt request flag)

Instruction code

D9									D0				
1	0	1	0	0	0	1	0	0	0	2	8	8	
									$_2$				$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	$V2_3$ = 0: (SIOF) = 1

Operation:
$V2_3$ = 0: (SIOF) = 1 ?
After skipping, (SIOF) ← 0
$V2_3$ = 1: SNZSI = NOP
($V2_3$ = bit 3 of interrupt control register V2)

Grouping: Serial I/O operation

Description: When $V2_3$ = 0 : Skips the next instruction when serial I/O interrupt request flag SIOF is "1." After skipping, clears (0) to the SIOF flag. When the SIOF flag is "0," executes the next instruction.
When $V2_3$ = 1 : This instruction is equivalent to the NOP instruction.

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

Instruction code

D9									D0				
1	0	1	0	0	0	0	0	0	0	2	8	0	
									$_2$				$_{16}$

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

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

Grouping: Timer operation

Description: When $V1_2$ = 0 : Skips the next instruction when timer 1 interrupt request flag T1F is "1." After skipping, clears (0) to the T1F flag. When the T1F flag is "0," executes the next instruction.
When $V1_2$ = 1 : This instruction is equivalent to the NOP instruction.

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

Instruction code

D9									D0				
1	0	1	0	0	0	0	0	0	1	2	8	1	
									$_2$				$_{16}$

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

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

Grouping: Timer operation

Description: When $V1_3$ = 0 : Skips the next instruction when timer 2 interrupt request flag T2F is "1." After skipping, clears (0) to the T2F flag. When the T2F flag is "0," executes the next instruction.
When $V1_3$ = 1 : This instruction is equivalent to the NOP instruction.

SNZT3 (Skip if Non Zero condition of Timer 3 interrupt request flag)

Instruction code

D9									D0				
1	0	1	0	0	0	0	0	1	0	2	8	2	
									$_2$				$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	$V2_0$ = 0: (T3F) = 1

Operation:
$V2_0$ = 0: (T3F) = 1 ?
After skipping, (T3F) ← 0
$V2_0$ = 1: SNZT3 = NOP
($V2_0$ = bit 0 of interrupt control register V2)

Grouping: Timer operation

Description: When $V2_0$ = 0 : Skips the next instruction when timer 3 interrupt request flag T3F is "1." After skipping, clears (0) to the T3F flag. When the T3F flag is "0," executes the next instruction.
When $V2_0$ = 1 : This instruction is equivalent to the NOP instruction.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

SNZT4 (Skip if Non Zero condition of Timer 4 inerrupt request flag)

Instruction code

D9									D0			
1	0	1	0	0	0	0	0	1	1	2	8	3

$_2$ $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	$V2_1$ = 0: (T4F) = 1

Operation: $V2_1$ = 0: (T4F) = 1 ?
After skipping, (T4F) ← 0
$V2_1$ = 1: SNZT4 = NOP
($V2_1$ = bit 1 of interrupt control register V2)

Grouping: Timer operation

Description: When $V2_1$ = 0 : Skips the next instruction when timer 4 interrupt request flag T4F is "1." After skipping, clears (0) to the T4F flag. When the T4F flag is "0," executes the next instruction.
When $V2_1$ = 1 : This instruction is equivalent to the NOP instruction.

SRST (System ReSeT)

Instruction code

D9									D0			
0	0	0	0	0	0	0	0	0	1	0	0	1

$_2$ $_{16}$

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

Operation: System reset occurrence

Grouping: Other operation

Description: System reset occurs.

SST (Serial i/o transmission/reception STart)

Instruction code

D9									D0			
1	0	1	0	0	1	1	1	1	0	2	9	E

$_2$ $_{16}$

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

Operation: (SIOF) ← 0
Serial I/O transmission/reception start

Grouping: Serial I/O operation

Description: Clears (0) to SIOF flag and starts serial I/O.

SZB j (Skip if Zero, Bit)

Instruction code

D9									D0			
0	0	0	0	1	0	0	0	j	j	0	2	j

$_2$ $_{16}$

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

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

Grouping: Bit operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

SZC (Skip if Zero, Carry flag)

Instruction code

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

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

Operation: (CY) = 0 ?

Grouping: Arithmetic operation

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

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

Instruction code

D9									D0				
0	0	0	0	1	0	0	1	0	0	$_2$	0 2 4	$_{16}$	
0	0	0	0	1	0	1	0	1	1	$_2$	0 2 B	$_{16}$	

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	(D(Y)) = 0 (Y) = 0 to 7

Operation: (D(Y)) = 0 ?
(Y) = 0 to 7

Grouping: Input/Output operation

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

T1AB (Transfer data to timer 1 and register R1 from Accumulator and register B)

Instruction code

D9									D0				
1	0	0	0	1	1	0	0	0	0	$_2$	2 3 0	$_{16}$	

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

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

Grouping: Timer operation

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

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

Instruction code

D9									D0				
1	0	0	0	1	1	0	0	0	1	$_2$	2 3 1	$_{16}$	

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

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

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits of timer 2 and timer 2 reload register R2. Transfers the contents of register A to the low-order 4 bits of timer 2 and timer 2 reload register R2.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

T3AB (Transfer data to timer 3 and register R3 from Accumulator and register B)

Instruction code

D9									D0				
1	0	0	0	1	1	0	0	1	0	$_2$	2	3	2	$_{16}$

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

Operation:
(T3$_7$–T3$_4$) ← (B)
(R3$_7$–R3$_4$) ← (B)
(T3$_3$–T3$_0$) ← (A)
(R3$_3$–R3$_0$) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits of timer 3 and timer 3 reload register R3. Transfers the contents of register A to the low-order 4 bits of timer 3 and timer 3 reload register R3.

T4AB (Transfer data to timer 4 and register R4L from Accumulator and register B)

Instruction code

D9									D0				
1	0	0	0	1	1	0	0	1	1	$_2$	2	3	3	$_{16}$

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

Operation:
(T4$_7$–T4$_4$) ← (B)
(R4L$_7$–R4L$_4$) ← (B)
(T4$_3$–T4$_0$) ← (A)
(R4L$_3$–R4L$_0$) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits of timer 4 and timer 4 reload register R4L. Transfers the contents of register A to the low-order 4 bits of timer 4 and timer 4 reload register R4L.

T4HAB (Transfer data to register R4H from Accumulator and register B)

Instruction code

D9									D0				
1	0	0	0	1	1	0	1	1	1	$_2$	2	3	7	$_{16}$

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

Operation:
(R4H$_7$–R4H$_4$) ← (B)
(R4H$_3$–R4H$_0$) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits of timer 4 and timer 4 reload register R4H. Transfers the contents of register A to the low-order 4 bits of timer 4 and timer 4 reload register R4H.

T4R4L (Transfer data to timer 4 from register R4L)

Instruction code

D9									D0				
1	0	1	0	0	1	0	1	1	1	$_2$	2	9	7	$_{16}$

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

Operation:
(T4$_7$–T4$_4$) ← (R4L$_7$–R4L$_4$)
(T4$_3$–T4$_0$) ← (R4L$_3$–R4L$_0$)

Grouping: Timer operation

Description: Transfers the contents of reload register R4L to timer 4.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAB (Transfer data to Accumulator from register B)

Instruction code

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

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

Operation: (A) ← (B)

Grouping: Register to register transfer

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

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

Instruction code

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

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

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

Grouping: Timer operation

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

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

Instruction code

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

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

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

Grouping: Timer operation

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

TAB3 (Transfer data to Accumulator and register B from timer 3)

Instruction code

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

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

Operation: (B) ← ($T3_7$–$T3_4$)
(A) ← ($T3_3$–$T3_0$)

Grouping: Timer operation

Description: Transfers the high-order 4 bits ($T3_7$–$T3_4$) of timer 3 to register B.
Transfers the low-order 4 bits ($T3_3$–$T3_0$) of timer 3 to register A.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAB4 (Transfer data to Accumulator and register B from timer 4)

Instruction code

D9									D0			
1	0	0	1	1	1	0	0	1	1	2	7	3

$_2$ $_{16}$

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

Operation:
(B) ← ($T4_7$–$T4_4$)
(A) ← ($T4_3$–$T4_0$)

Grouping: Timer operation

Description: Transfers the high-order 4 bits ($T4_7$–$T4_4$) of timer 4 to register B.
Transfers the low-order 4 bits ($T4_3$–$T4_0$) of timer 4 to register A.

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

Instruction code

D9									D0			
1	0	0	1	1	1	1	0	0	1	2	7	9

$_2$ $_{16}$

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

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

Grouping: A/D conversion operation

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

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

Instruction code

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

$_2$ $_{16}$

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

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

Grouping: Register to register transfer

Description: Transfers the high-order 4 bits (E_7–E_4) of register E to register B, and low-order 4 bits of register E to register A.

TABP p (Transfer data to Accumulator and register B from Program memory in page p)

Instruction code

D9									D0			
0	0	1	0	p_5	p_4	p_3	p_2	p_1	p_0	0	8 +p	p

$_2$ $_{16}$

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

Operation:
(SP) ← (SP) + 1
(SK(SP)) ← (PC)
(PC_H) ← p
(PC_L) ← (DR_2–DR_0, A_3–A_0)
(DR_2) ← 0
(DR_1, DR_0) ← (ROM(PC))$_{9, 8}$
(B) ← (ROM(PC))$_{7–4}$
(A) ← (ROM(PC))$_{3–0}$
(PC) ← (SK(SP))
(SP) ← (SP) – 1

Grouping: Arithmetic operation

Description: Transfers bits 9 and 8 to register D, bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 7 to 0 are the ROM pattern in address (DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0)$_2$ specified by registers A and D in page p.

Note: p is 0 to 47 for M34519M6, and p is 0 to 63 for M34519M8E8.
When this instruction is executed, be careful not to over the stack because 1 stage of stack register is used.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TABPS (Transfer data to Accumulator and register B from PreScaler)

Instruction code

D9								D0				
1	0	0	1	1	1	0	1	0	1	$_2$	2 7 5	$_{16}$

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

Operation:
(B) ← (TPS7–TPS4)
(A) ← (TPS3–TPS0)

Grouping: Timer operation

Description: Transfers the high-order 4 bits (TPS7–TPS4) of prescaler to register B, and transfers the low-order 4 bits (TPS3–TPS0) of prescaler to register A.

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

Instruction code

D9								D0				
1	0	0	1	1	1	1	0	0	0	$_2$	2 7 8	$_{16}$

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

Operation:
(B) ← (SI7–SI4)
(A) ← (SI3–SI0)

Grouping: Serial I/O operation

Description: Transfers the high-order 4 bits (SI7–SI4) of serial I/O register SI to register B, and transfers the low-order 4 bits (SI3–SI0) of serial I/O register SI to register A.

TAD (Transfer data to Accumulator from register D)

Instruction code

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

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

Operation:
(A2–A0) ← (DR2–DR0)
(A3) ← 0

Grouping: Register to register transfer

Description: Transfers the contents of register D to the low-order 3 bits (A2–A0) of register A.

Note: When this instruction is executed, “0” is stored to the bit 3 (A3) of register A.

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

Instruction code

D9								D0				
1	0	0	0	1	1	1	0	0	1	$_2$	2 3 9	$_{16}$

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

Operation:
(AD7–AD4) ← (B)
(AD3–AD0) ← (A)

Grouping: A/D conversion operation

Description: In the A/D conversion mode (Q13 = 0), this instruction is equivalent to the NOP instruction. In the comparator mode (Q13 = 1), transfers the contents of register B to the high-order 4 bits (AD7–AD4) of comparator register, and the contents of register A to the low-order 4 bits (AD3–AD0) of comparator register.
(Q13 = bit 3 of A/D control register Q1)

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAI1 (Transfer data to Accumulator from register I1)

Instruction code

D9								D0				
1	0	0	1	0	1	0	0	1	1	$_2$	2 5 3	$_{16}$

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

Operation: (A) ← (I1)

Grouping: Interrupt operation

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

TAI2 (Transfer data to Accumulator from register I2)

Instruction code

D9								D0				
1	0	0	1	0	1	0	1	0	0	$_2$	2 5 4	$_{16}$

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

Operation: (A) ← (I2)

Grouping: Interrupt operation

Description: Transfers the contents of interrupt control register I2 to register A.

TAJ1 (Transfer data to Accumulator from register J1)

Instruction code

D9								D0				
1	0	0	1	0	0	0	0	1	0	$_2$	2 4 2	$_{16}$

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

Operation: (A) ← (J1)

Grouping: Serial I/O operation

Description: Transfers the contents of serial I/O control register J1 to register A.

TAK0 (Transfer data to Accumulator from register K0)

Instruction code

D9								D0				
1	0	0	1	0	1	0	1	1	0	$_2$	2 5 6	$_{16}$

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

Operation: (A) ← (K0)

Grouping: Input/Output operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAK1 (Transfer data to Accumulator from register K1)

Instruction code

D9									D0				
1	0	0	1	0	1	1	0	0	1	$_2$	2	5	9	$_{16}$

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

Operation: (A) ← (K1)

Grouping: Input/Output operation

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

TAK2 (Transfer data to Accumulator from register K2)

Instruction code

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

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

Operation: (A) ← (K2)

Grouping: Input/Output operation

Description: Transfers the contents of key-on wakeup control register K2 to register A.

TALA (Transfer data to Accumulator from register LA)

Instruction code

D9									D0				
1	0	0	1	0	0	1	0	0	1	$_2$	2	4	9	$_{16}$

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

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

Grouping: A/D conversion operation

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

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

TAM j (Transfer data to Accumulator from Memory)

Instruction code

D9									D0				
1	0	1	1	0	0	j	j	j	j	$_2$	2	C	j	$_{16}$

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

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

Grouping: RAM to register transfer

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAMR (Transfer data to Accumulator from register MR)

Instruction code

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

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

Operation: (A) ← (MR)

Grouping: Clock operation

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

TAPU0 (Transfer data to Accumulator from register PU0)

Instruction code

D9									D0				
1	0	0	1	0	1	0	1	1	1	2	5	7	
									$_2$				$_{16}$

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

Operation: (A) ← (PU0)

Grouping: Input/Output operation

Description: Transfers the contents of pull-up control register PU0 to register A.

TAPU1 (Transfer data to Accumulator from register PU1)

Instruction code

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

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

Operation: (A) ← (PU1)

Grouping: Input/Output operation

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

TAQ1 (Transfer data to Accumulator from register Q1)

Instruction code

D9									D0				
1	0	0	1	0	0	0	1	0	0	2	4	4	
									$_2$				$_{16}$

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

Operation: (A) ← (Q1)

Grouping: A/D conversion operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAQ2 (Transfer data to Accumulator from register Q2)

Instruction code

D9									D0				
1	0	0	1	0	0	0	1	0	1	$_2$	2	4	5	$_{16}$

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

Operation: (A) ← (Q2)

Grouping: A/D conversion operation

Description: Transfers the contents of A/D control register Q2 to register A.

TAQ3 (Transfer data to Accumulator from register Q3)

Instruction code

D9									D0				
1	0	0	1	0	0	0	1	1	0	$_2$	2	4	6	$_{16}$

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

Operation: (A) ← (Q3)

Grouping: A/D conversion operation

Description: Transfers the contents of A/D control register Q3 to register A.

TASP (Transfer data to Accumulator from Stack Pointer)

Instruction code

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

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

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

Grouping: Register to register transfer

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

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

TAV1 (Transfer data to Accumulator from register V1)

Instruction code

D9									D0				
0	0	0	1	0	1	0	1	0	0	$_2$	0	5	4	$_{16}$

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

Operation: (A) ← (V1)

Grouping: Interrupt operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAV2 (Transfer data to Accumulator from register V2)

Instruction code

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

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

Operation: (A) ← (V2)

Grouping: Interrupt operation

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

TAW1 (Transfer data to Accumulator from register W1)

Instruction code

D9									D0				
1	0	0	1	0	0	1	0	1	1	$_2$	2	4	B	$_{16}$

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

Operation: (A) ← (W1)

Grouping: Timer operation

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

TAW2 (Transfer data to Accumulator from register W2)

Instruction code

D9									D0				
1	0	0	1	0	0	1	1	0	0	$_2$	2	4	C	$_{16}$

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

Operation: (A) ← (W2)

Grouping: Timer operation

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

TAW3 (Transfer data to Accumulator from register W3)

Instruction code

D9									D0				
1	0	0	1	0	0	1	1	0	1	$_2$	2	4	D	$_{16}$

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

Operation: (A) ← (W3)

Grouping: Timer operation

Description: Transfers the contents of timer control register W3 to register A.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAW4 (Transfer data to Accumulator from register W4)

Instruction code

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

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

Operation: (A) ← (W4)

Grouping: Timer operation

Description: Transfers the contents of timer control register W4 to register A.

TAW5 (Transfer data to Accumulator from register W5)

Instruction code

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

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

Operation: (A) ← (W5)

Grouping: Timer operation

Description: Transfers the contents of timer control register W5 to register A.

TAW6 (Transfer data to Accumulator from register W6)

Instruction code

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

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

Operation: (A) ← (W6)

Grouping: Timer operation

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

TAX (Transfer data to Accumulator from register X)

Instruction code

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

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

Operation: (A) ← (X)

Grouping: Register to register transfer

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAY (Transfer data to Accumulator from register Y)

Instruction code

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

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

Operation: (A) ← (Y)

Grouping: Register to register transfer

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

TAZ (Transfer data to Accumulator from register Z)

Instruction code

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

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

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

Grouping: Register to register transfer

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

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

TBA (Transfer data to register B from Accumulator)

Instruction code

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

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

Operation: (B) ← (A)

Grouping: Register to register transfer

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

TDA (Transfer data to register D from Accumulator)

Instruction code

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

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

Operation: (DR_2–DR_0) ← (A_2–A_0)

Grouping: Register to register transfer

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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

Instruction code

D9									D0				
0	0	0	0	0	1	1	0	1	0	0	1	A	
$_2$										$_{16}$			

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

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

Grouping: Register to register transfer

Description: Transfers the contents of register B to the high-order 4 bits (E_7–E_4) of register E, and the contents of register A to the low-order 4 bits (E_3–E_0) of register E.

TFR0A (Transfer data to register FR0 from Accumulator)

Instruction code

D9									D0				
1	0	0	0	1	0	1	0	0	0	2	2	8	
$_2$										$_{16}$			

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

Operation: (FR0) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to the port output structure control register FR0.

TFR1A (Transfer data to register FR1 from Accumulator)

Instruction code

D9									D0				
1	0	0	0	1	0	1	0	0	1	2	2	9	
$_2$										$_{16}$			

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

Operation: (FR1) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to the port output structure control register FR1.

TFR2A (Transfer data to register FR2 from Accumulator)

Instruction code

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

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

Operation: (FR2) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to the port output structure control register FR2.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TFR3A (Transfer data to register FR3 from Accumulator)

Instruction code

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

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

Operation: (FR3) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to the port output structure control register FR3.

TI1A (Transfer data to register I1 from Accumulator)

Instruction code

D9								D0				
1	0	0	0	0	1	0	1	1	1	2	1	7
									$_2$			$_{16}$

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

Operation: (I1) ← (A)

Grouping: Interrupt operation

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

TI2A (Transfer data to register I2 from Accumulator)

Instruction code

D9								D0				
1	0	0	0	0	1	1	0	0	0	2	1	8
									$_2$			$_{16}$

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

Operation: (I2) ← (A)

Grouping: Interrupt operation

Description: Transfers the contents of register A to interrupt control register I2.

TJ1A (Transfer data to register J1 from Accumulator)

Instruction code

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

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

Operation: (J1) ← (A)

Grouping: Serial I/O operation

Description: Transfers the contents of register A to serial I/O control register J1.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TK0A (Transfer data to register K0 from Accumulator)

Instruction code

D9									D0			
1	0	0	0	0	1	1	0	1	1	2	1	B

$_2$ $_{16}$

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

Operation: (K0) ← (A)

Grouping: Input/Output operation

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

TK1A (Transfer data to register K1 from Accumulator)

Instruction code

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

$_2$ $_{16}$

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

Operation: (K1) ← (A)

Grouping: Input/Output operation

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

TK2A (Transfer data to register K2 from Accumulator)

Instruction code

D9									D0			
1	0	0	0	0	1	0	1	0	1	2	1	5

$_2$ $_{16}$

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

Operation: (K2) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to key-on wakeup control register K2.

TMA j (Transfer data to Memory from Accumulator)

Instruction code

D9									D0			
1	0	1	0	1	1	j	j	j	j	2	B	j

$_2$ $_{16}$

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

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

Grouping: RAM to register transfer

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TMRA (Transfer data to register MR from Accumulator)

Instruction code

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

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

Operation: (MR) ← (A)

Grouping: Other operation

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

TPAA (Transfer data to register PA from Accumulator)

Instruction code

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

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

Operation: (PA_0) ← (A_0)

Grouping: Timer operation

Description: Transfers the contents of lowermost bit (A_0) register A to timer control register PA.

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

Instruction code

D9									D0			
1	0	0	0	1	1	0	1	0	1	2	3	5
$_2$										$_{16}$		

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

Operation:
(RPS_7–RPS_4) ← (B)
(TPS_7–TPS_4) ← (B)
(RPS_3–RPS_0) ← (A)
(TPS_3–TPS_0) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits of prescaler and prescaler reload register RPS, and transfers the contents of register A to the low-order 4 bits of prescaler and prescaler reload register RPS.

TPU0A (Transfer data to register PU0 from Accumulator)

Instruction code

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

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

Operation: (PU0) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to pull-up control register PU0.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TPU1A (Transfer data to register PU1 from Accumulator)

Instruction code

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

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

Operation: (PU1) ← (A)

Grouping: Input/Output operation

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

TQ1A (Transfer data to register Q1 from Accumulator)

Instruction code

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

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

Operation: (Q1) ← (A)

Grouping: A/D conversion operation

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

TQ2A (Transfer data to register Q2 from Accumulator)

Instruction code

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

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

Operation: (Q2) ← (A)

Grouping: A/D conversion operation

Description: Transfers the contents of register A to A/D control register Q2.

TQ3A (Transfer data to register Q3 from Accumulator)

Instruction code

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

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

Operation: (Q3) ← (A)

Grouping: A/D conversion operation

Description: Transfers the contents of register A to A/D control register Q3.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TR1AB (Transfer data to register R1 from Accumulator and register B)

Instruction code

D9									D0				
1	0	0	0	1	1	1	1	1	1	$_2$	2	3	F	$_{16}$

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

Operation: (R17–R14) ← (B)
(R13–R10) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits (R17–R14) of reload register R1, and the contents of register A to the low-order 4 bits (R13–R10) of reload register R1.

TR3AB (Transfer data to register R3 from Accumulator and register B)

Instruction code

D9									D0				
1	0	0	0	1	1	1	0	1	1	$_2$	2	3	B	$_{16}$

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

Operation: (R37–R34) ← (B)
(R33–R30) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits (R37–R34) of reload register R3, and the contents of register A to the low-order 4 bits (R33–R30) of reload register R3.

TRGA (Transfer data to register RG from Accumulator)

Instruction code

D9									D0				
1	0	0	0	0	0	1	0	0	1	$_2$	2	0	9	$_{16}$

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

Operation: (RG0) ← (A0)

Grouping: Clock control operation

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

TSIAB (Transfer data to register SI from Accumulator and register B)

Instruction code

D9									D0				
1	0	0	0	1	1	1	0	0	0	$_2$	2	3	8	$_{16}$

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

Operation: (SI7–SI4) ← (B)
(SI3–SI0) ← (A)

Grouping: Serial I/O operation

Description: Transfers the contents of register B to the high-order 4 bits (SI7–SI4) of serial I/O register SI, and transfers the contents of register A to the low-order 4 bits (SI3–SI0) of serial I/O register SI.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TV1A (Transfer data to register V1 from Accumulator)

Instruction code

D9									D0				
0	0	0	0	1	1	1	1	1	1	$_2$	0 3 F	$_{16}$	

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

Operation: (V1) ← (A)

Grouping: Interrupt operation

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

TV2A (Transfer data to register V2 from Accumulator)

Instruction code

D9									D0				
0	0	0	0	1	1	1	1	1	0	$_2$	0 3 E	$_{16}$	

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

Operation: (V2) ← (A)

Grouping: Interrupt operation

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

TW1A (Transfer data to register W1 from Accumulator)

Instruction code

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

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

Operation: (W1) ← (A)

Grouping: Timer operation

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

TW2A (Transfer data to register W2 from Accumulator)

Instruction code

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

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

Operation: (W2) ← (A)

Grouping: Timer operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TW3A (Transfer data to register W3 from Accumulator)

Instruction code

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

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

Operation: (W3) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register A to timer control register W3.

TW4A (Transfer data to register W4 from Accumulator)

Instruction code

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

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

Operation: (W4) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register A to timer control register W4.

TW5A (Transfer data to register W5 from Accumulator)

Instruction code

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

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

Operation: (W5) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register A to timer control register W5.

TW6A (Transfer data to register W6 from Accumulator)

Instruction code

D9									D0				
1	0	0	0	0	1	0	0	1	1	$_2$	2	1	3	$_{16}$

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

Operation: (W6) ← (A)

Grouping: Timer operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TYA (Transfer data to register Y from Accumulator)

Instruction code

D9									D0				
0	0	0	0	0	0	1	1	0	0	0	0	C
$_2$										$_{16}$			

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

Operation: (Y) ← (A)

Grouping: Register to register transfer

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

WRST (Watchdog timer ReSeT)

Instruction code

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

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

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

Grouping: Other operation

Description: Skips the next instruction when watchdog timer flag WDF1 is "1." After skipping, clears (0) to the WDF1 flag. When the WDF1 flag is "0," executes the next instruction. Also, stops the watchdog timer function when executing the WRST instruction immediately after the DWDT instruction.

XAM j (eXchange Accumulator and Memory data)

Instruction code

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

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

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

Grouping: RAM to register transfer

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

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

Instruction code

D9									D0				
1	0	1	1	1	1	j	j	j	j	2	F	j
$_2$										$_{16}$			

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

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

Grouping: RAM to register transfer

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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

Instruction code

D9									D0				
1	0	1	1	1	0	j	j	j	j	2	E	j	

$_2$ $_{16}$

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

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

Grouping: RAM to register transfer

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

MACHINE INSTRUCTIONS (INDEX BY TYPES)

Type of instructions \ Parameter	Mnemonic	Instruction code D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	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 register B to the high-order 4 bits (E_7–E_4) of register E, and the contents of register A to the low-order 4 bits (E_3–E_0) of register E.
–	–	Transfers the high-order 4 bits (E_7–E_4) of register E to register B, and low-order 4 bits (E_3–E_0) of register E to register A.
–	–	Transfers the contents of the low-order 3 bits (A_2–A_0) of register A to register D.
–	–	Transfers the contents of register D to the low-order 3 bits (A_2–A_0) of register A.
–	–	Transfers the contents of register Z to the low-order 2 bits (A_1, A_0) of register A.
–	–	Transfers the contents of register X to register A.
–	–	Transfers the contents of stack pointer (SP) to the low-order 3 bits (A_2–A_0) of register A.
Continuous description	–	Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y. When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped.
–	–	Loads the value z in the immediate field to register Z.
(Y) = 0	–	Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. When the contents of register Y is not 0, the next instruction is executed.
(Y) = 15	–	Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped. When the contents of register Y is not 15, the next instruction is executed.
–	–	After transferring the contents of M(DP) to register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
–	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
(Y) = 15	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped. When the contents of register Y is not 15, the next instruction is executed.
(Y) = 0	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped. When the contents of register Y is not 0, the next instruction is executed.
–	–	After transferring the contents of register A to M(DP), an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

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
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 (Note) (PC_L) ← (DR_2–DR_0, A_3–A_0) (DR_2) ← 0 (DR_1, DR_0) ← (ROM(PC))$_{9, 8}$ (B) ← (ROM(PC))$_{7–4}$ (A) ← (ROM(PC))$_{3–0}$ (SK(SP)) ← (PC) (SP) ← (SP) – 1
	AM	0	0	0	0	0	0	1	0	1	0	0 0 A	1	1	(A) ← (A) + (M(DP))
	AMC	0	0	0	0	0	0	1	0	1	1	0 0 B	1	1	(A) ← (A) + (M(DP)) +(CY) (CY) ← Carry
	A n	0	0	0	1	1	0	n	n	n	n	0 6 n	1	1	(A) ← (A) + n n = 0 to 15
	AND	0	0	0	0	0	1	1	0	0	0	0 1 8	1	1	(A) ← (A) AND (M(DP))
	OR	0	0	0	0	0	1	1	0	0	1	0 1 9	1	1	(A) ← (A) OR (M(DP))
	SC	0	0	0	0	0	0	0	1	1	1	0 0 7	1	1	(CY) ← 1
	RC	0	0	0	0	0	0	0	1	1	0	0 0 6	1	1	(CY) ← 0
	SZC	0	0	0	0	1	0	1	1	1	1	0 2 F	1	1	(CY) = 0 ?
	CMA	0	0	0	0	0	1	1	1	0	0	0 1 C	1	1	(A) ← ($\overline{A}$)
	RAR	0	0	0	0	0	1	1	1	0	1	0 1 D	1	1	→ CY → $A_3A_2A_1A_0$ →
Bit operation	SB j	0	0	0	1	0	1	1	1	j	j	0 5 C +j	1	1	(Mj(DP)) ← 1 j = 0 to 3
	RB j	0	0	0	1	0	0	1	1	j	j	0 4 C +j	1	1	(Mj(DP)) ← 0 j = 0 to 3
	SZB j	0	0	0	0	1	0	0	0	j	j	0 2 j	1	1	(Mj(DP)) = 0 ? j = 0 to 3
Comparison operation	SEAM	0	0	0	0	1	0	0	1	1	0	0 2 6	1	1	(A) = (M(DP)) ?
	SEA n	0	0	0	0	1	0	0	1	0	1	0 2 5	2	2	(A) = n ? n = 0 to 15
		0	0	0	1	1	1	n	n	n	n	0 7 n			

Note: p is 0 to 47 for M34519M6,
p is 0 to 63 for M34519M8/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 9 and 8 to register D, 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 ad-dress (DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0)$_2$ specified by registers A and D in page p. When this instruction is executed, be careful not to over the stack because 1 stage of stack register is used.
–	–	Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged.
–	0/1	Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY.
Overflow = 0	–	Adds the value n in the immediate field to register A, and stores a result in register A. The contents of carry flag CY remains unchanged. Skips the next instruction when there is no overflow as the result of operation. Executes the next instruction when there is overflow as the result of operation.
–	–	Takes the AND operation between the contents of register A and the contents of M(DP), and stores the result in register A.
–	–	Takes the OR operation between the contents of register A and the contents of M(DP), and stores the result in register A.
–	1	Sets (1) to carry flag CY.
–	0	Clears (0) to carry flag CY.
(CY) = 0	–	Skips the next instruction when the contents of carry flag CY is “0.”
–	–	Stores the one’s complement for register A’s contents in register A.
–	0/1	Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right.
–	–	Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).
–	–	Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).
(Mj(DP)) = 0 j = 0 to 3	–	Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is “0.” Executes the next instruction when the contents of bit j of M(DP) is “1.”
(A) = (M(DP))	–	Skips the next instruction when the contents of register A is equal to the contents of M(DP). Executes the next instruction when the contents of register A is not equal to the contents of M(DP).
(A) = n	–	Skips the next instruction when the contents of register A is equal to the value n in the immediate field. Executes the next instruction when the contents of register A is not equal to the value n in the immediate field.

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Type of instructions / Parameter	Mnemonic	Instruction code D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
Branch operation	B a	0	1	1	a6	a5	a4	a3	a2	a1	a0	1 8 a +a	1	1	(PCL) ← a6–a0
	BL p, a	0	0	1	1	1	p4	p3	p2	p1	p0	0 E p +p	2	2	(PCH) ← p (Note) (PCL) ← a6–a0
		1	0	p5	a6	a5	a4	a3	a2	a1	a0	2 p a +a			
	BLA p	0	0	0	0	0	1	0	0	0	0	0 1 0	2	2	(PCH) ← p (Note) (PCL) ← (DR2–DR0, A3–A0)
		1	0	p5	p4	0	0	p3	p2	p1	p0	2 p p			
Subroutine operation	BM a	0	1	0	a6	a5	a4	a3	a2	a1	a0	1 a a	1	1	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← 2 (PCL) ← a6–a0
	BML p, a	0	0	1	1	0	p4	p3	p2	p1	p0	0 C p +p	2	2	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PCH) ← p (Note)
		1	0	p5	a6	a5	a4	a3	a2	a1	a0	2 p a +a			(PCL) ← a6–a0
	BMLA p	0	0	0	0	1	1	0	0	0	0	0 3 0	2	2	(SP) ← (SP) + 1 (SK(SP)) ← (PC)
		1	0	p5	p4	0	0	p3	p2	p1	p0	2 p p			(PCH) ← p (Note) (PCL) ← (DR2–DR0,A3–A0)
Return operation	RTI	0	0	0	1	0	0	0	1	1	0	0 4 6	1	1	(PC) ← (SK(SP)) (SP) ← (SP) – 1
	RT	0	0	0	1	0	0	0	1	0	0	0 4 4	1	2	(PC) ← (SK(SP)) (SP) ← (SP) – 1
	RTS	0	0	0	1	0	0	0	1	0	1	0 4 5	1	2	(PC) ← (SK(SP)) (SP) ← (SP) – 1

Note: p is 0 to 47 for M34519M6,
p is 0 to 63 for M34519M8/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.

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Parameter / Type of instructions	Mnemonic	Instruction code D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
Interrupt operation	DI	0	0	0	0	0	0	0	1	0	0	0 0 4	1	1	(INTE) ← 0
	EI	0	0	0	0	0	0	0	1	0	1	0 0 5	1	1	(INTE) ← 1
	SNZ0	0	0	0	0	1	1	1	0	0	0	0 3 8	1	1	$V1_0$ = 0: (EXF0) = 1 ? After skipping, (EXF0) ← 0 $V1_0$ = 1: SNZ0 = NOP
	SNZ1	0	0	0	0	1	1	1	0	0	1	0 3 9	1	1	$V1_1$ = 0: (EXF1) = 1 ? After skipping, (EXF1) ← 0 $V1_1$ = 1: SNZ1 = NOP
	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	TPAA	1	0	1	0	1	0	1	0	1	0	2 A A	1	1	(PA_0) ← (A_0)
	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)

Skip condition	Carry flag CY	Datailed description
–	–	Clears (0) to interrupt enable flag INTE, and disables the interrupt.
–	–	Sets (1) to interrupt enable flag INTE, and enables the interrupt.
$V1_0$ = 0: (EXF0) = 1	–	When $V1_0$ = 0 : Skips the next instruction when external 0 interrupt request flag EXF0 is “1.” After skipping, clears (0) to the EXF0 flag. When the EXF0 flag is “0,” executes the next instruction. When $V1_0$ = 1 : This instruction is equivalent to the NOP instruction. ($V1_0$: bit 0 of interrupt control register V1)
$V1_1$ = 0: (EXF1) = 1	–	When $V1_1$ = 0 : Skips the next instruction when external 1 interrupt request flag EXF1 is “1.” After skipping, clears (0) to the EXF1 flag. When the EXF1 flag is “0,” executes the next instruction. When $V1_1$ = 1 : This instruction is equivalent to the NOP instruction. ($V1_1$: bit 1 of interrupt control register V1)
(INT0) = “H” However, $I1_2$ = 1	–	When $I1_2$ = 1 : Skips the next instruction when the level of INT0 pin is “H.” ($I1_2$: bit 2 of interrupt control register I1)
(INT0) = “L” However, $I1_2$ = 0	–	When $I1_2$ = 0 : Skips the next instruction when the level of INT0 pin is “L.”
(INT1) = “H” However, $I2_2$ = 1	–	When $I2_2$ = 1 : Skips the next instruction when the level of INT1 pin is “H.” ($I2_2$: bit 2 of interrupt control register I2)
(INT1) = “L” However, $I2_2$ = 0	–	When $I2_2$ = 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 register A to timer control register PA.
–	–	Transfers the contents of timer control register W1 to register A.
–	–	Transfers the contents of register A to timer control register W1.
–	–	Transfers the contents of timer control register W2 to register A.
–	–	Transfers the contents of register A to timer control register W2.
–	–	Transfers the contents of timer control register 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.

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Parameter / Type of instructions	Mnemonic	Instruction code D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
Timer operation	TAW5	1	0	0	1	0	0	1	1	1	1	2 4 F	1	1	(A) ← (W5)
	TW5A	1	0	0	0	0	1	0	0	1	0	2 1 2	1	1	(W5) ← (A)
	TAW6	1	0	0	1	0	1	0	0	0	0	2 5 0	1	1	(A) ← (W6)
	TW6A	1	0	0	0	0	1	0	0	1	1	2 1 3	1	1	(W6) ← (A)
	TABPS	1	0	0	1	1	1	0	1	0	1	2 7 5	1	1	(B) ← (TPS7–TPS4) (A) ← (TPS3–TPS0)
	TPSAB	1	0	0	0	1	1	0	1	0	1	2 3 5	1	1	(RPS7–RPS4) ← (B) (TPS7–TPS4) ← (B) (RPS3–RPS0) ← (A) (TPS3–TPS0) ← (A)
	TAB1	1	0	0	1	1	1	0	0	0	0	2 7 0	1	1	(B) ← (T17–T14) (A) ← (T13–T10)
	T1AB	1	0	0	0	1	1	0	0	0	0	2 3 0	1	1	(R17–R14) ← (B) (T17–T14) ← (B) (R13–R10) ← (A) (T13–T10) ← (A)
	TAB2	1	0	0	1	1	1	0	0	0	1	2 7 1	1	1	(B) ← (T27–T24) (A) ← (T23–T20)
	T2AB	1	0	0	0	1	1	0	0	0	1	2 3 1	1	1	(R27–R24) ← (B) (T27–T24) ← (B) (R23–R20) ← (A) (T23–T20) ← (A)
	TAB3	1	0	0	1	1	1	0	0	1	0	2 7 2	1	1	(B) ← (T37–T34) (A) ← (T33–T30)
	T3AB	1	0	0	0	1	1	0	0	1	0	2 3 2	1	1	(R37–R34) ← (B) (T37–T34) ← (B) (R33–R30) ← (A) (T33–T30) ← (A)
	TAB4	1	0	0	1	1	1	0	0	1	1	2 7 3	1	1	(B) ← (T47–T44) (A) ← (T43–T40)
	T4AB	1	0	0	0	1	1	0	0	1	1	2 3 3	1	1	(R4L7–R4L4) ← (B) (T47–T44) ← (B) (R4L3–R4L0) ← (A) (T43–T40) ← (A)
	T4HAB	1	0	0	0	1	1	0	1	1	1	2 3 7	1	1	(R4H7–R4H4) ← (B) (R4H3–R4H0) ← (A)
	TR1AB	1	0	0	0	1	1	1	1	1	1	2 3 F	1	1	(R17–R14) ← (B) (R13–R10) ← (A)
	TR3AB	1	0	0	0	1	1	1	0	1	1	2 3 B	1	1	(R37–R34) ← (B) (R33–R30) ← (A)
	T4R4L	1	0	1	0	0	1	0	1	1	1	2 9 7	1	1	(T47–T40) ← (R4L7–R4L0)

Skip condition	Carry flag CY	Datailed description
–	–	Transfers the contents of timer control register W5 to register A.
–	–	Transfers the contents of register A to timer control register W5.
–	–	Transfers the contents of timer control register W6 to register A.
–	–	Transfers the contents of register A to timer control register W6.
–	–	Transfers the high-order 4 bits of prescaler to register B, and transfers the low-order 4 bits of prescaler to register A.
–	–	Transfers the contents of register B to the high-order 4 bits of prescaler and prescaler reload register RPS, and transfers the contents of register A to the low-order 4 bits of prescaler and prescaler reload register RPS.
–	–	Transfers the high-order 4 bits of timer 1 to register B, and transfers the low-order 4 bits of timer 1 to register A.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 1 and timer 1 reload register R1, and transfers the contents of register A to the low-order 4 bits of timer 1 and timer 1 reload register R1.
–	–	Transfers the high-order 4 bits of timer 2 to register B, and transfers the low-order 4 bits of timer 2 to register A.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 2 and timer 2 reload register R2, and transfers the contents of register A to the low-order 4 bits of timer 2 and timer 2 reload register R2.
–	–	Transfers the high-order 4 bits of timer 3 to register B, and transfers the low-order 4 bits of timer 3 to register A.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 3 and timer 3 reload register R3, and transfers the contents of register A to the low-order 4 bits of timer 3 and timer 3 reload register R3.
–	–	Transfers the high-order 4 bits of timer 4 to register B, and transfers the low-order 4 bits of timer 4 to register A.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 4 and timer 4 reload register R4L, and transfers the contents of register A to the low-order 4 bits of timer 4 and timer 4 reload register R4L.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 4 reload register R4H, and transfers the contents of register A to the low-order 4 bits of timer 4 reload register R4H.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 1 reload register R1, and transfers the contents of register A to the low-order 4 bits of timer 1 reload register R1.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 3 reload register R3, and transfers the contents of register A to the low-order 4 bits of timer 3 reload register R3.
–	–	Transfers the contents of timer 4 reload register R4L to timer 4.
–	–	

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Parameter / Type of instructions	Mnemonic	Instruction code D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
Timer operation	SNZT1	1	0	1	0	0	0	0	0	0	0	2 8 0	1	1	$V1_2$ = 0: (T1F) = 1 ? After skipping, (T1F) ← 0 $V1_2$ = 0: NOP
	SNZT2	1	0	1	0	0	0	0	0	0	1	2 8 1	1	1	$V1_3$ = 0: (T2F) = 1 ? After skipping, (T2F) ← 0 $V1_3$ = 0: NOP
	SNZT3	1	0	1	0	0	0	0	0	1	0	2 8 2	1	1	$V2_0$ = 0: (T3F) = 1 ? After skipping, (T3F) ← 0 $V2_0$ = 0: NOP
	SNZT4	1	0	1	0	0	0	0	0	1	1	2 8 3	1	1	$V2_1$ = 0: (T4F) = 1 ? After skipping, (T4F) ← 0 $V2_1$ = 0: NOP
Input/Output operation	IAP0	1	0	0	1	1	0	0	0	0	0	2 6 0	1	1	(A) ← (P0)
	OP0A	1	0	0	0	1	0	0	0	0	0	2 2 0	1	1	(P0) ← (A)
	IAP1	1	0	0	1	1	0	0	0	0	1	2 6 1	1	1	(A) ← (P1)
	OP1A	1	0	0	0	1	0	0	0	0	1	2 2 1	1	1	(P1) ← (A)
	IAP2	1	0	0	1	1	0	0	0	1	0	2 6 2	1	1	(A_2–A_0) ← ($P2_2$–$P2_0$) (A_3) ← 0
	OP2A	1	0	0	0	1	0	0	0	1	0	2 2 2	1	1	($P2_2$–$P2_0$) ← (A_2–A_0)
	IAP3	1	0	0	1	1	0	0	0	1	1	2 6 3	1	1	(A_1, A_0) ← ($P3_1$, $P3_0$)
	OP3A	1	0	0	0	1	0	0	0	1	1	2 2 3	1	1	($P3_1$, $P3_0$) ← (A_1, A_0)
	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)
	IAP6	1	0	0	1	1	0	0	1	1	0	2 6 6	1	1	(A) ← (P6)
	OP6A	1	0	0	0	1	0	0	1	1	0	2 2 6	1	1	(P6) ← (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	1	1	(D(Y)) = 0 ? (Y) = 0 to 7
		0	0	0	0	1	0	1	0	1	1	0 2 B	1	1	
	TAPU0	1	0	0	1	0	1	0	1	1	1	2 5 7	1	1	(A) ← (PU0)
	TPU0A	1	0	0	0	1	0	1	1	0	1	2 2 D	1	1	(PU0) ← (A)
	TAPU1	1	0	0	1	0	1	1	1	1	0	2 5 E	1	1	(A) ← (PU1)
	TPU1A	1	0	0	0	1	0	1	1	1	0	2 2 E	1	1	(PU1) ← (A)

Skip condition	Carry flag CY	Datailed description
$V1_2$ = 0: (T1F) = 1	–	Skips the next instruction when the contents of bit 2 ($V1_2$) of interrupt control register V1 is “0” and the contents of T1F flag is “1.” After skipping, clears (0) to T1F flag.
$V1_3$ = 0: (T2F) =1	–	Skips the next instruction when the contents of bit 3 ($V1_3$) of interrupt control register V1 is “0” and the contents of T2F flag is “1.” After skipping, clears (0) to T2F flag.
$V2_0$ = 0: (T3F) = 1	–	Skips the next instruction when the contents of bit 0 ($V2_0$) of interrupt control register V2 is “0” and the contents of T3F flag is “1.” After skipping, clears (0) to T3F flag.
$V2_1$ = 0: (T4F) =1	–	Skips the next instruction when the contents of bit 1 ($V2_1$) of interrupt control register V2 is “0” and the contents of T4F flag is “1.” After skipping, clears (0) to T4F flag.
–	–	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.
–	–	Outputs the contents of register A to port P2.
–	–	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.
–	–	Transfers the input of port P6 to register A.
–	–	Outputs the contents of register A to port P6.
–	–	Sets (1) to all 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 However, (Y)=0 to 7	–	Skips the next instruction when a bit of port D specified by register Y is “0.” Executes the next instruction when a bit of port D specified by register Y is “1.”
–	–	Transfers the contents of pull-up control register PU0 to register A.
–	–	Transfers the contents of register A to pull-up control register PU0.
–	–	Transfers the contents of pull-up control register PU1 to register A.
–	–	Transfers the contents of register A to pull-up control register PU1.

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Parameter / Type of instructions	Mnemonic	Instruction code D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
Input/Output operation	TAK0	1	0	0	1	0	1	0	1	1	0	2 5 6	1	1	(A) ← (K0)
	TK0A	1	0	0	0	0	1	1	0	1	1	2 1 B	1	1	(K0) ← (A)
	TAK1	1	0	0	1	0	1	1	0	0	1	2 5 9	1	1	(A) ← (K1)
	TK1A	1	0	0	0	0	1	0	1	0	0	2 1 4	1	1	(K1) ← (A)
	TAK2	1	0	0	1	0	1	1	0	1	0	2 5 A	1	1	(A) ← (K2)
	TK2A	1	0	0	0	0	1	0	1	0	1	2 1 5	1	1	(K2) ← (A)
	TFR0A	1	0	0	0	1	0	1	0	0	0	2 2 8	1	1	(FR0) ← (A)
	TFR1A	1	0	0	0	1	0	1	0	0	1	2 2 9	1	1	(FR1) ← (A)
	TFR2A	1	0	0	0	1	0	1	0	1	0	2 2 A	1	1	(FR2) ← (A)
	TFR3A	1	0	0	0	1	0	1	0	1	1	2 2 B	1	1	(FR3) ← (A)
Serial I/O operation	TABSI	1	0	0	1	1	1	1	0	0	0	2 7 8	1	1	(B) ← (SI_7–SI_4) (A) ← (SI_3–SI_0)
	TSIAB	1	0	0	0	1	1	1	0	0	0	2 3 8	1	1	(SI_7–SI_4) ← (B) (SI_3–SI_0) ← (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	$V2_3$=0: (SIOF)=1? After skipping, (SIOF) ← 0 $V2_3$ = 1: NOP
	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)
Clock operation	CMCK	1	0	1	0	0	1	1	0	1	0	2 9 A	1	1	Ceramic resonator selected
	CRCK	1	0	1	0	0	1	1	0	1	1	2 9 B	1	1	RC oscillator selected
	CYCK	1	0	1	0	0	1	1	1	0	1	2 9 D	1	1	Quartz-crystal oscillator selected
	TRGA	1	0	0	0	0	0	1	0	0	1	2 0 9	1	1	(RG_0) ← (A_0)
	TAMR	1	0	0	1	0	1	0	0	1	0	2 5 2	1	1	(A) ← (MR)
	TMRA	1	0	0	0	0	1	0	1	1	0	2 1 6	1	1	(MR) ← (A)

Skip condition	Carry flag CY	Datailed description
–	–	Transfers the contents of key-on wakeup control register K0 to register A.
–	–	Transfers the contents of register A to key-on wakeup control register K0 .
–	–	Transfers the contents of key-on wakeup control register K1 to register A.
–	–	Transfers the contents of register A to key-on wakeup control register K1.
–	–	Transfers the contents of key-on wakeup control register K2 to register A.
–	–	Transfers the contents of register A to key-on wakeup control register K2.
–	–	Transferts the contents of register A to port output format control register FR0.
–	–	Transferts the contents of register A to port output format control register FR1.
–	–	Transferts the contents of register A to port output format control register FR2.
–	–	Transferts the contents of register A to port output format control register FR3.
–	–	Transfers the high-order 4 bits of serial I/O register SI to register B, and transfers the low-order 4 bits of serial I/O register SI to register A.
–	–	Transfers the contents of register B to the high-order 4 bits of serial I/O register SI, and transfers the contents of register A to the low-order 4 bits of serial I/O register SI.
–	–	Clears (0) to SIOF flag and starts serial I/O.
$V2_3$ = 0: (SIOF) = 1	–	Skips the next instruction when the contents of bit 3 ($V2_3$) of interrupt control register V2 is “0” and contents of SIOF flag is “1.” After skipping, clears (0) to SIOF flag.
–	–	Transfers the contents of serial I/O control register J1 to register A.
–	–	Transfers the contents of register A to serial I/O control register J1.
–	–	Selects the ceramic resonator for main clock f(X_{IN}).
–	–	Selects the RC oscillation circuit for main clock f(X_{IN}).
–	–	Selects the quartz-crystal oscillation circuit for main clock f(X_{IN}).
–	–	Transfers the contents of clock control regiser RG to register A.
–	–	Transfers the contents of clock control regiser MR to register A.
–	–	Transfers the contents of register A to clock control register MR.

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Parameter / Type of instructions	Mnemonic	Instruction code D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Hexadecimal notation	Number of words	Number of cycles	Function
A/D conversion operation	TABAD	1	0	0	1	1	1	1	0	0	1	2 7 9	1	1	$Q1_3$ = 0: (B) ← (AD_9–AD_6) (A) ← (AD_5–AD_2) $Q1_3$ = 1: (B) ← (AD_7–AD_4) (A) ← (AD_3–AD_0)
	TALA	1	0	0	1	0	0	1	0	0	1	2 4 9	1	1	(A_3, A_2) ← (AD_1, AD_0) (A_1, A_0) ← 0
	TADAB	1	0	0	0	1	1	1	0	0	1	2 3 9	1	1	(AD_7–AD_4) ← (B) (AD_3–AD_0) ← (A)
	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	$V2_1$ = 0: (ADF) = 1 ? After skipping, (ADF) ← 0 $V2_2$ = 1: NOP
	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)
	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)
	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) ← (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	Transition to RAM back-up mode
	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) = 1 ? After skipping, (WDF1) ← 0
	DWDT	1	0	1	0	0	1	1	1	0	0	2 9 C	1	1	Stop of watchdog timer function enabled
	SRST	0	0	0	0	0	0	0	0	0	1	0 0 1	1	1	System reset occurrence

Skip condition	Carry flag CY	Datailed description
–	–	In the A/D conversion mode ($Q1_3$ = 0), transfers the high-order 4 bits (AD_9–AD_6) of register AD to register B, and the middle-order 4 bits (AD_5–AD_2) of register AD to register A. In the comparator mode ($Q1_3$ = 1), transfers the middle-order 4 bits (AD_7–AD_4) of register AD to register B, and the low-order 4 bits (AD_3–AD_0) of register AD to register A. ($Q1_3$: bit 3 of A/D control register Q1)
–	–	Transfers the low-order 2 bits (AD_1, AD_0) of register AD to the high-order 2 bits (AD_3, AD_2) of register A.
–	–	In the comparator mode ($Q1_3$ = 1), transfers the contents of register B to the high-order 4 bits (AD_7–AD_4) of comparator register, and the contents of register A to the low-order 4 bits (AD_3–AD_0) of comparator register. ($Q1_3$ = bit 3 of A/D control register Q1)
–	–	Clears (0) to A/D conversion completion flag ADF, and the A/D conversion at the A/D conversion mode ($Q1_3$ = 0) or the comparator operation at the comparator mode ($Q1_3$ = 1) is started. ($Q1_3$ = bit 3 of A/D control register Q1)
$V2_2$ = 0: (ADF) = 1	–	When $V2_2$ = 0 : Skips the next instruction when A/D conversion completion flag ADF is “1.” After skipping, clears (0) to the ADF flag. When the ADF flag is “0,” executes the next instruction. ($V2_2$: bit 2 of interrupt control register V2)
–	–	Transfers the contents of A/D control register Q1 to register A.
–	–	Transfers the contents of register A to A/D control register Q1.
–	–	Transfers the contents of A/D control register Q2 to register A.
–	–	Transfers the contents of register A to A/D control register Q2.
–	–	Transfers the contents of A/D control register Q3 to register A.
–	–	Transfers the contents of register A to A/D control register Q3.
–	–	No operation; Adds 1 to program counter value, and others remain unchanged.
–	–	Puts the system in RAM back-up state by executing the POF instruction after executing the EPOF instruction.
–	–	Makes the immediate after POF instruction valid by executing the EPOF instruction.
(P) = 1	–	Skips the next instruction when the P flag is “1”. After skipping, the P flag remains unchanged.
(WDF1) = 1	–	Skips the next instruction when watchdog timer flag WDF1 is “1.” After skipping, clears (0) to the WDF1 flag. Also, stops the watchdog timer function when executing the WRST instruction immediately after the DWDT instruction.
–	–	Stops the watchdog timer function by the WRST instruction after executing the DWDT instruction.
–	–	System reset occurs.

INSTRUCTION CODE TABLE

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

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

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

	The second word
BL	1p paaa aaaa
BML	1p paaa aaaa
BLA	1p pp00 pppp
BMLA	1p pp00 pppp
SEA	00 0111 nnnn
SZD	00 0010 1011

• * cannot be used in the M34519M6.

INSTRUCTION CODE TABLE (continued)

D9–D4 / D3–D0	Hex. notation	100000	100001	100010	100011	100100	100101	100110	100111	101000	101001	101010	101011	101100	101101	101110	101111	110000 111111
		20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30–3F
0000	0	–	TW3A	OP0A	T1AB	–	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	TW5A	OP2A	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	TK1A	OP4A	–	TAQ1	TAI2	IAP4	–	–	–	–	TMA 4	TAM 4	XAM 4	XAMI 4	XAMD 4	LXY
0101	5	TQ2A	TK2A	OP5A	TPSAB	TAQ2	–	IAP5	TABPS	–	–	–	TMA 5	TAM 5	XAM 5	XAMI 5	XAMD 5	LXY
0110	6	TQ3A	TMRA	OP6A	–	TAQ3	TAK0	IAP6	–	–	–	–	TMA 6	TAM 6	XAM 6	XAMI 6	XAMD 6	LXY
0111	7	–	TI1A	–	T4HAB	–	TAPU0	–	–	SNZAD	T4R4L	–	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	TRGA	–	TFR1A	TADAB	TALA	TAK1	–	TABAD	–	–	–	TMA 9	TAM 9	XAM 9	XAMI 9	XAMD 9	LXY
1010	A	–	–	TFR2A	–	–	TAK2	–	–	–	CMCK	TPAA	TMA 10	TAM 10	XAM 10	XAMI 10	XAMD 10	LXY
1011	B	–	TK0A	TFR3A	TR3AB	TAW1	–	–	–	–	CRCK	–	TMA 11	TAM 11	XAM 11	XAMI 11	XAMD 11	LXY
1100	C	–	–	–	–	TAW2	–	–	–	–	DWDT	–	TMA 12	TAM 12	XAM 12	XAMI 12	XAMD 12	LXY
1101	D	–	–	TPU0A	–	TAW3	–	–	–	–	CYCK	–	TMA 13	TAM 13	XAM 13	XAMI 13	XAMD 13	LXY
1110	E	TW1A	–	TPU1A	–	TAW4	TAPU1	–	–	–	SST	–	TMA 14	TAM 14	XAM 14	XAMI 14	XAMD 14	LXY
1111	F	TW2A	–	–	TR1AB	TAW5	–	–	–	–	ADST	–	TMA 15	TAM 15	XAM 15	XAMI 15	XAMD 15	LXY

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

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

	The second word
BL	1p paaa aaaa
BML	1p paaa aaaa
BLA	1p pp00 pppp
BMLA	1p pp00 pppp
SEA	00 0111 nnnn
SZD	00 0010 1011

BUILT-IN PROM VERSION

In addition to the mask ROM versions, the 4519 Group has the One Time PROM versions whose PROMs can only be written to and not be erased.

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

Table 23 shows the product of built-in PROM version. Figure 73 shows the pin configurations of built-in PROM versions.

The One Time PROM version has pin-compatibility with the mask ROM version.

Table 24 Product of built-in PROM version

Part number	PROM size (× 10 bits)	RAM size (× 4 bits)	Package	ROM type
M34519E8FP	8192 words	384 words	42P2R-A	One Time PROM [shipped in blank]

PIN CONFIGURATION (TOP VIEW)

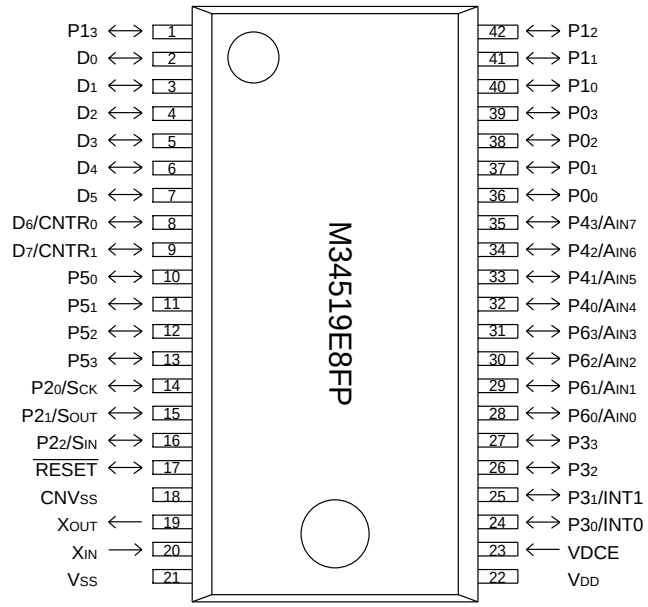


Fig. 71 Pin configuration of built-in PROM version

(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 adapter is listed in Table 24. Contact addresses at the end of this data 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 73.

(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, Renesas Technology 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 74 before using is recommended (Products shipped in blank: PROM contents is not written in factory when shipped).

(3) Electric Characteristic Differences Between Mask ROM and One TIme PROM Version MCU

There are differences in electric characteristics, operation margin, noise immunity, and noise radiation between Mask ROM and One Time PROM version MCUs due to the difference in the manufacturing processes.

When manufacturing an application system with the One Time PROM version and then switching to use of the Mask ROM version, please perform sufficient evaluations for the commercial samples of the Mask ROM version.

Table 25 Programming adapter

Microcomputer	Name of Programming Adapter
M34519E8FP	PCA7441

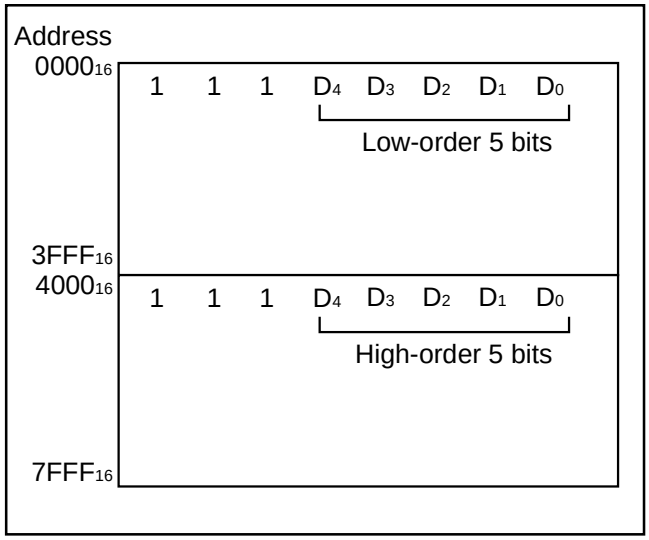


Fig. 72 PROM memory map

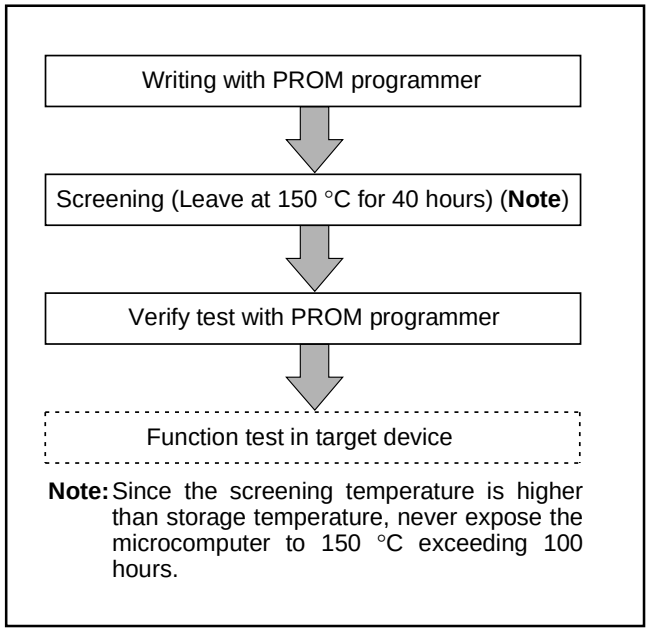


Fig. 73 Flow of writing and test of the product shipped in blank

CHAPTER 2

APPLICATION

2.1 I/O pins

The 4519 Group has thirty-five I/O pins.
Port P2 is also used as Serial I/O pins SCK, SOUT, SIN.
Port $P3_0$ is also used as INT0 input pin.
Port $P3_1$ is also used as INT1 input pin.
Port P4 is also used as analog input pins A_{IN4}–A_{IN7}.
Port P6 is also used as analog input pins A_{IN0}–A_{IN3}.
Port D_6 is also used as CNTR0 I/O pin.
Port D_7 is also used as CNTR1 I/O pin.
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**

In the following conditions, the pin state of port P0 is transferred as input data to register A when the **IAP0** instruction is executed.
• Set bit $FR0_0$ or bit $FR0_1$ of register FR0 to "0" according to the port to be used.
• Set the output latch of specified port P0i (i=0, 1, 2 or 3) to "1" with the **OP0A** instruction.
If $FR0_0$ or $FR0_1$ is "0" and the output latch is "0", "0" is output to specified port P0.
If $FR0_0$ or $FR0_1$ is "1", the output latch value is output to specified port P0.

● **Output**

The contents of register A is set to the output latch with the **OP0A** instruction, and is output to port P0.
N-channel open-drain or CMOS can be selected as the output structure of port P0 in 2 bits unit by setting $FR0_0$ or $FR0_1$.

(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 PU1.

● **Input**

In the following conditions, the pin state of port P1 is transferred as input data to register A when the **IAP1** instruction is executed.
• Set bit $FR0_2$ or bit $FR0_3$ of register FR0 to "0" according to the port to be used.
• Set the output latch of specified port P1i (i=0, 1, 2 or 3) to "1" with the **OP1A** instruction.
If $FR0_2$ or $FR0_3$ is "0" and the output latch is "0", "0" is output to specified port P1.
If $FR0_2$ or $FR0_3$ is "1", the output latch value is output to specified port P1.

● **Output**

The contents of register A is set to the output latch with the **OP1A** instruction, and is output to port P1.
N-channel open-drain or CMOS can be selected as the output structure of port P1 in 2 bits unit by setting $FR0_2$ or $FR0_3$.

(3) Port P2

Port P2 is a 3-bit I/O port.
$P2_0$–$P2_3$ are also used as serial I/O pins SCK, SOUT, SIN.

● **Input**

In the following condition, the pin state of port P2 is transferred as input data to register A when the **IAP2** instruction is executed.
• Set the output latch of specified port P2i (i=0, 1 or 2) to "1" with the **OP2A** instruction.
If the output latch is "0", "0" is output to specified port P2.

● **Output**

The contents of register A is set to the output latch with the **OP2A** instruction, and is output to port P2.
The output structure is an N-channel open-drain.

Notes 1: Port $P2_0$ is also used as the serial I/O pin SCK. Accordingly, when port $P2_0$ is used as an input/output port, set bits $J1_1$ and $J1_0$ of register J1 to "00_2". Also, set bits $J1_3$ and $J1_2$ of register J1 to "00_2", "01_2" or "10_2".
2: Port $P2_1$ is also used as the serial I/O pin SOUT. Accordingly, when port $P2_1$ is used as an input/output port, set bits $J1_1$ and $J1_0$ of register J1 to "00_2" or "10_2".
3: Port $P2_2$ is also used as the serial I/O pin SIN. Accordingly, when port $P2_2$ is used as an input/output port, set bits $J1_1$ and $J1_0$ of register J1 to "00_2" or "10_2".

(4) Port P3

Port P3 is a 4-bit I/O port.
$P3_0$ is also used as INT0 input pin and $P3_1$ is also used as INT1 input pin.
Also, the key-on wakeup function of INT0 and INT1 can be turned ON/OFF by setting bits $K2_0$ and $K2_2$ of register K2.

● **Input**

In the following condition, the pin state of port P3 is transferred as input data to register A when the **IAP3** instruction is executed.
• Set the output latch of specified port P3i (i=0, 1, 2 or 3) to "1" with the **OP3A** instruction.
If the output latch is "0", "0" is output to specified port P3.

● **Output**

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

(5) Port P4

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

Port P4$_0$–P4$_3$ are also used as analog input pins A_{IN4}–A_{IN7}.

● **Input**

In the following conditions, the pin state of port P4 is transferred as input data to register A when the **IAP4** instruction is executed.

• Set the output latch of specified port P4i (i=0, 1, 2 or 3) to "1" with the **OP4A** instruction.

If the output latch is "0", "0" is output to specified port P4.

● **Output**

The contents of register A is set to the output latch with the **OP4A** instruction, and is output to port P4.

The output structure is an N-channel open-drain.

(6) Port P5

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

● **Input**

In the following conditions, the pin state of port P5 is transferred as input data to register A when the **IAP5** instruction is executed.

• Set bit FR3$_i$ (i=0, 1, 2 or 3) of register FR3 to "0" according to the port to be used.

• Set the output latch of specified port P5i (i=0, 1, 2 or 3) to "1" with the **OP5A** instruction.

If FR3$_i$ is "0" and the output latch is "0", "0" is output to specified port P5.

If FR3$_i$ is "1", the output latch value is output to specified port P5.

● **Output**

The contents of register A is set to the output latch with the **OP5A** instruction, and is output to port P5.

N-channel open-drain or CMOS can be selected as the output structure of port P5 in 2 bits unit by setting FR3$_i$.

(7) Port P6

Port P6 is a 4-bit I/O port.

Port P6$_0$–P6$_3$ are also used as analog input pins A_{IN0}–A_{IN3}.

● **Input**

In the following conditions, the pin state of port P6 is transferred as input data to register A when the **IAP6** instruction is executed.

• Set the output latch of specified port P6i (i=0, 1, 2 or 3) to "1" with the **OP6A** instruction.

If the output latch is "0", "0" is output to specified port P6.

● **Output**

The contents of register A is set to the output latch with the **OP6A** instruction, and is output to port P6.

The output structure is an N-channel open-drain.

(8) Port D

Ports D_0–D_7 are eight independent I/O ports. Port D_6 is also used as CNTR0 I/O pin. Port D_7 is also used as CNTR1 I/O pin.

■ 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.

● Input

The pin state of port D can be obtained with the **SZD** instruction.
In the following conditions, if the pin state of port Dj (j=0, 1, 2, 3, 4, 5, 6 or 7) is "0" when the **SZD** instruction is executed, the next instruction is skipped. If it is "1" when the **SZD** instruction is executed, the next instruction is executed.

- Set bit i (i=0,1,2 or 3) of register FR1 or FR2 to "0" according to the port to be used.
- Set the output latch of specified port Dj to "1" with the **SD** instruction.

If $FR1_i$ or $FR2_i$ is "0" and the output latch is "0", "0" is output to specified port D.
If $FR1_i$ or $FR2_i$ is "1", the output latch value is output to specified port D.

● Output

Set the output level to the output latch with the **SD**, **CLD** and **RD** instructions.
The state of pin enters the high-impedance state when the **SD** instruction is executed.
All port D enter the high-impedance state or "H" level state when the **CLD** instruction is executed.
The state of pin becomes "L" level when the **RD** instruction is executed.
N-channel open-drain or CMOS can be selected as the output structure of ports D_0–D_7 in 1 bit unit by setting registers FR1, FR2.

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 pin. Accordingly, when using port D_6, set bit 0 ($W6_0$) of register W6 to "0."
3: Port D_7 is also used as CNTR1 pin. Accordingly, when using port D_7, set bit 3 ($W4_3$) of register W4 to "0."

2.1.2 Related registers

(1) Timer control register W4

Table 2.1.1 shows the timer control register W4.
Set the contents of this register through register A with the **TW4A** instruction.
The contents of register W4 is transferred to register A with the **TAW4** instruction.

Table 2.1.1 Timer control register W4

Timer control register W4		at reset : 0000_2	at RAM back-up : state retained	R/W
$W4_3$	D_7/CNTR1 pin function selection bit	0	D_7 (I/O) / CNTR1 (input)	
		1	CNTR1 (I/O) / D_7 (input)	
$W4_2$	PWM signal "H" interval expansion function control bit	0	PWM signal "H" interval expansion function invalid	
		1	PWM signal "H" interval expansion function valid	
$W4_1$	Timer 4 control bit	0	Stop (state retained)	
		1	Operating	
$W4_0$	Timer 4 count source selection bit	0	X_{IN} input	
		1	Prescaler output (ORCLK) divided by 2	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: When setting the port, $W4_2$–$W4_0$ are not used.

(2) Timer control register W6

Table 2.1.2 shows the timer control register W6.
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 2.1.2 Timer control register W6

Timer control register W6		at reset : 0000_2	at RAM back-up : state retained	R/W
$W6_3$	CNTR1 pin input count edge selection bit	0	Falling edge	
		1	Rising edge	
$W6_2$	CNTR0 pin input count edge selection bit	0	Falling edge	
		1	Rising edge	
$W6_1$	CNTR1 output auto-control circuit selection bit	0	CNTR1 output auto-control circuit not selected	
		1	CNTR1 output auto-control circuit selected	
$W6_0$	D_6/CNTR0 pin function selection bit (**Note 2**)	0	D_6(I/O)/CNTR0 input	
		1	CNTR0 input/output/D_6 (input)	

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

(3) Serial I/O control register J1

Table 2.1.3 shows the serial I/O control register J1.
Set the contents of this register through register A with the **TJ1A** instruction.
The contents of register J1 is transferred to register A with the **TAJ1** instruction.

Table 2.1.3 Serial I/O control register J1

Serial I/O control register J1		at reset : 0000_2		at RAM back-up : state retained	R/W
$J13$	Serial I/O synchronous clock selection bits	$J13$	$J12$	Synchronous clock	
		0	0	Instruction clock (INSTCK) divided by 8	
$J12$		0	1	Instruction clock (INSTCK) divided by 4	
		1	0	Instruction clock (INSTCK) divided by 2	
		1	1	External clock (SCK input)	
$J11$	Serial I/O port function selection bits	$J11$	$J10$	Port function	
		0	0	$P20$, $P21$, $P22$ selected/SCK, SOUT, SIN not selected	
$J10$		0	1	SCK, SOUT, $P22$ selected/$P20$, $P21$, SIN not selected	
		1	0	SCK, $P21$, SIN selected/$P20$, SOUT, $P22$ not selected	
		1	1	SCK, SOUT, SIN selected/$P20$, $P21$, $P22$ not selected	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: When setting the port, $J13$–$J12$ are not used.

(4) A/D control register Q2

Table 2.1.4 shows the A/D control register Q2.
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 2.1.4 A/D control register Q2

A/D control register Q2		at reset : 0000_2	at RAM back-up : state retained	R/W
$Q23$	$P40$/AIN4, $P41$/AIN5, $P42$/AIN6, $P43$/AIN7 pin function selection bit	0	$P40$, $P41$, $P42$, $P43$	
		1	AIN4, AIN5, AIN6, AIN7	
$Q22$	$P62$/AIN2, $P63$/AIN3 pin function selection bit	0	$P62$, $P63$	
		1	AIN2, AIN3	
$Q21$	$P61$/AIN1 pin function selection bit	0	$P61$	
		1	AIN1	
$Q20$	$P60$/AIN0 pin function selection bit	0	$P60$	
		1	AIN0	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: In order to select AIN3–AIN0, set register Q1 after setting register Q2.

(5) Pull-up control register PU0

Table 2.1.5 shows the pull-up control register PU0.
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 2.1.5 Pull-up control register PU0

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

(6) Pull-up control register PU1

Table 2.1.6 shows the pull-up control register PU1.
Set the contents of this register through register A with the **TPU1A** instruction.
The contents of register PU1 is transferred to register A with the **TAPU1** instruction.

Table 2.1.6 Pull-up control register PU1

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

(7) Port output structure control register FR0

Table 2.1.7 shows the port output structure control register FR0.
Set the contents of this register through register A with the **TFR0A** instruction.

Table 2.1.7 Port output structure control register FR0

Port output structure control register FR0		at reset : 0000_2	at RAM back-up : state retained	W
$FR0_3$	Ports $P1_2$, $P1_3$ output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR0_2$	Ports $P1_0$, $P1_1$ output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR0_1$	Ports $P0_2$, $P0_3$ output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR0_0$	Ports $P0_1$, $P0_0$ output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	

Note: "W" represents write enabled.

(8) Port output structure control register FR1

Table 2.1.8 shows the port output structure control register FR1.
Set the contents of this register through register A with the **TFR1A** instruction.

Table 2.1.8 Port output structure control register FR1

Port output structure control register FR1		at reset : 0000_2	at RAM back-up : state retained	W
$FR1_3$	Port D_3 output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR1_2$	Port D_2 output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR1_1$	Port D_1 output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR1_0$	Port D_0 output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	

Note: "W" represents write enabled.

(9) Port output structure control register FR2

Table 2.1.9 shows the port output structure control register FR2.
Set the contents of this register through register A with the **TFR2A** instruction.

Table 2.1.9 Port output structure control register FR2

Port output structure control register FR2		at reset : 0000_2	at RAM back-up : state retained	W
$FR2_3$	Port D_7/CNTR1 output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR2_2$	Port D_6/CNTR0 output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR2_1$	Port D_5 output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR2_0$	Port D_4 output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	

Note: "W" represents write enabled.

(10) Port output structure control register FR3

Table 2.1.10 shows the port output structure control register FR3.
Set the contents of this register through register A with the **TFR3A** instruction.

Table 2.1.10 Port output structure control register FR3

Port output structure control register FR3		at reset : 0000_2	at RAM back-up : state retained	W
$FR3_3$	Port $P5_3$ output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR3_2$	Port $P5_2$ output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR3_1$	Port $P5_1$ output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	
$FR3_0$	Port $P5_0$ output structure selection bit	0	N-channel open-drain output	
		1	CMOS output	

Note: "W" represents write enabled.

(11) Key-on wakeup control register K0

Table 2.1.11 shows the key-on wakeup control register K0.
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 2.1.11 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 $P1_2$, $P1_3$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_2$	Pins $P1_0$, $P1_1$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_1$	Pins $P0_2$, $P0_3$ key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K0_0$	Pins $P0_0$, $P0_1$ 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.

(12) Key-on wakeup control register K2

Table 2.1.12 shows the key-on wakeup control register K2.
Set the contents of this register through register A with the **TK2A** instruction.
The contents of register K2 is transferred to register A with the **TAK2** instruction.

Table 2.1.12 Key-on wakeup control register K2

Key-on wakeup control register K2		at reset : 0000_2	at RAM back-up : state retained	R/W
$K2_3$	INT1 pin return condition selection bit	0	Return by level	
		1	Return by edge	
$K2_2$	INT1 pin key-on wakeup control bit	0	Key-on wakeup invalid	
		1	Key-on wakeup valid	
$K2_1$	INT0 pin return condition selection bit	0	Returned by level	
		1	Returned by edge	
$K2_0$	INT0 pin key-on wakeup control bit	0	Key-on wakeup invalid	
		1	Key-on wakeup valid	

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

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.

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

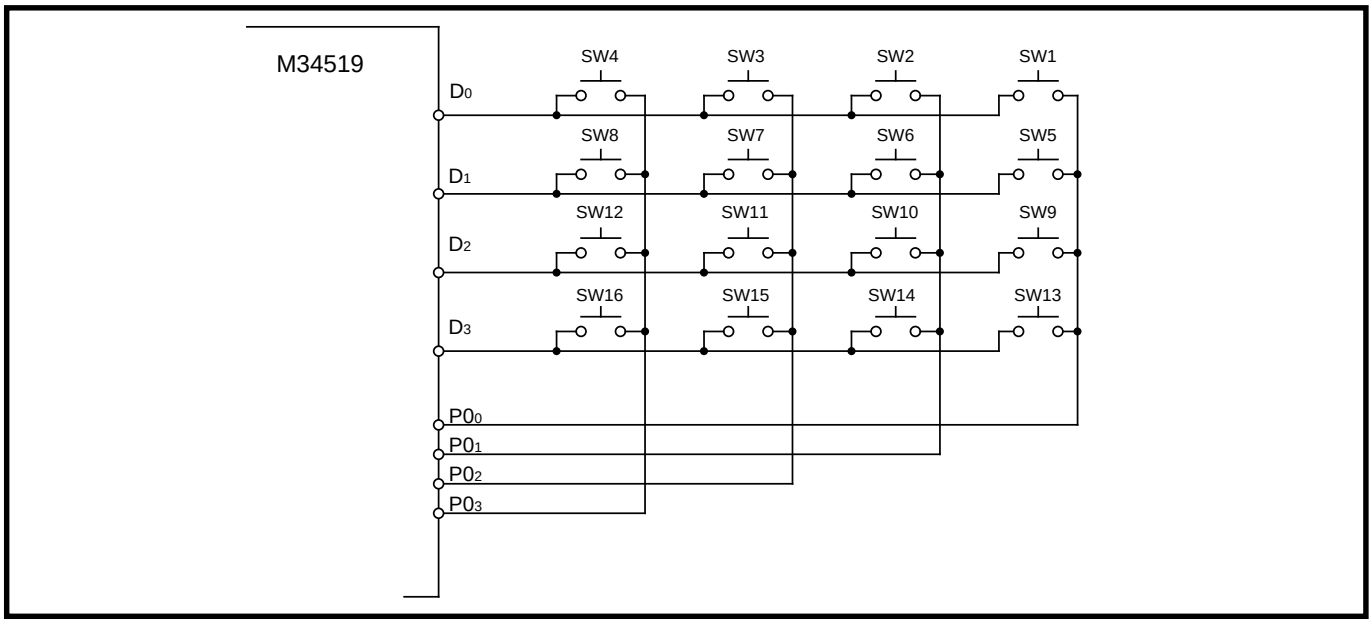
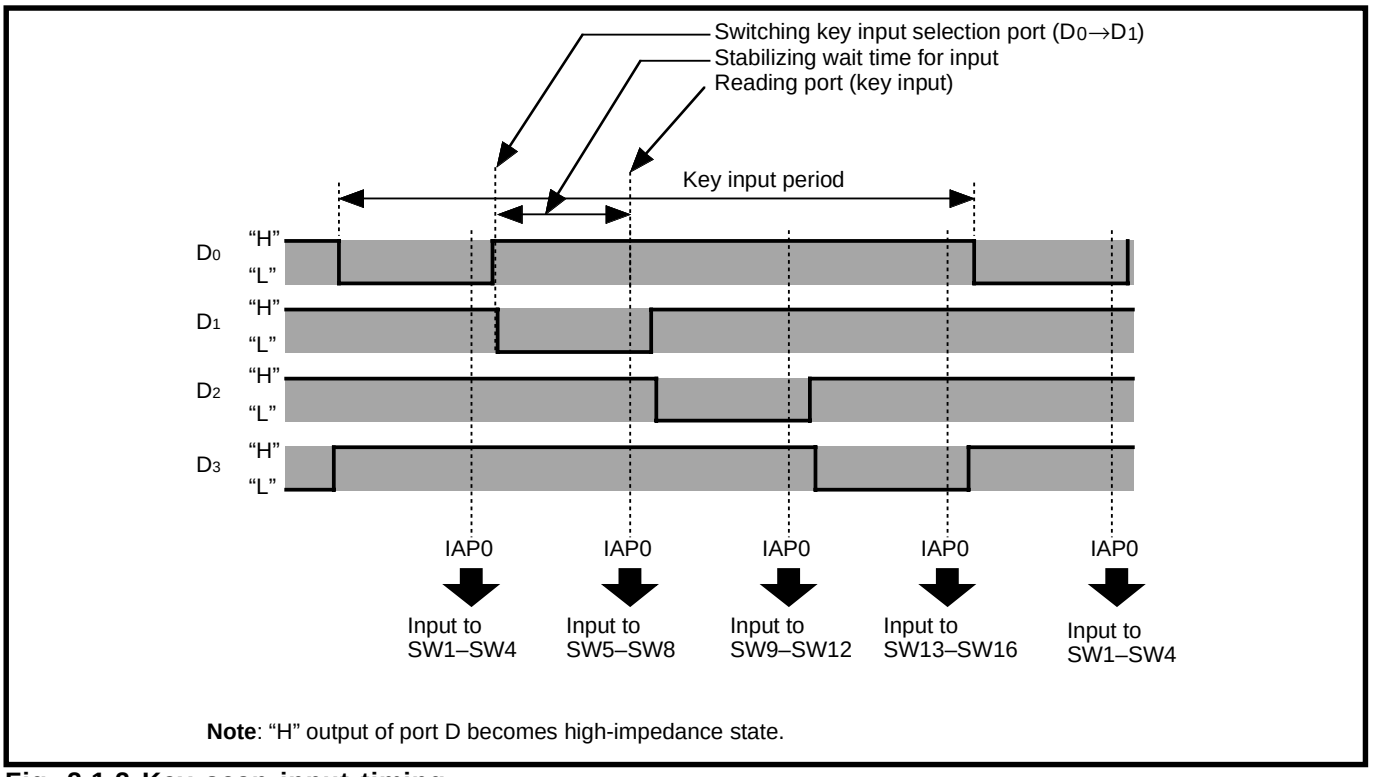


Fig. 2.1.1 Key input by key scan



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 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.

As for the port which has the output structure selection function, select the N-channel open-drain output structure.

(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 One Time 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) Multifunction

- Be careful that the output of ports $P3_0$ and $P3_1$ can be used even when INT0 and INT1 pins are selected.
- Be careful that the input of ports $P2_0$–$P2_2$ can be used even when S_{IN}, S_{OUT} and S_{CK} pins are selected.
- Be careful that the input/output of port D_6 can be used even when input of CNTR0 pin is selected.
- Be careful that the input of port D_6 can be used even when output of CNTR0 pin is selected.
- Be careful that the input/output of port D_7 can be used even when input of CNTR1 pin is selected.
- Be careful that the input of port D_7 can be used even when output of CNTR1 pin is selected.

(4) Connection of unused pins

Table 2.1.13 shows the connections of unused pins.

(5) SD, RD, SZD instructions

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

(6) Port $P3_0$/INT0 pin

When the RAM back-up mode is used by clearing the bit 3 of register I1 to "0" and setting the input of INT0 pin to be disabled, be careful about the following note.

- When the input of INT0 pin is disabled (register $I1_3$ = "0"), clear bit 0 of register K2 to "0" to invalidate the key-on wakeup before system goes into the RAM back-up mode.

(7) Port $P3_1$/INT1 pin

When the RAM back-up mode is used by clearing the bit 3 of register I2 to "0" and setting the input of INT1 pin to be disabled, be careful about the following note.

- When the input of INT1 pin is disabled (register $I2_3$ = "0"), clear bit 2 of register K2 to "0" to invalidate the key-on wakeup before system goes into the RAM back-up mode.

Table 2.1.13 Connections of unused pins

Pin	Connection	Usage condition	
XIN	Open.	Internal oscillator is selected.	(**Note 1**)
XOUT	Open.	Internal oscillator is selected.	(**Note 1**)
		RC oscillator is selected.	(**Note 2**)
		External clock input is selected for main clock.	(**Note 3**)
D0–D5	Open.	———	
	Connect to VSS.	N-channel open-drain is selected for the output structure.	(**Note 4**)
D6/CNTR0	Open.	CNTR0 input is not selected for timer 1 count source.	
	Connect to VSS.	N-channel open-drain is selected for the output structure.	(**Note 4**)
D7/CNTR1	Open.	CNTR1 input is not selected for timer 3 count source.	
	Connect to VSS.	N-channel open-drain is selected for the output structure.	(**Note 4**)
P00–P03	Open.	The key-on wakeup function is not selected.	(**Note 6**)
	Connect to VSS.	N-channel open-drain is selected for the output structure.	(**Note 5**)
		The pull-up function is not selected.	(**Note 4**)
		The key-on wakeup function is not selected.	(**Note 6**)
P10–P13	Open.	The key-on wakeup function is not selected.	(**Note 7**)
	Connect to VSS.	N-channel open-drain is selected for the output structure.	(**Note 5**)
		The pull-up function is not selected.	(**Note 4**)
		The key-on wakeup function is not selected.	(**Note 7**)
P20/SCK	Open.	SCK pin is not selected.	
	Connect to VSS.	———	
P21/SOUT	Open.	———	
	Connect to VSS.	———	
P22/SIN	Open.	SIN pin is not selected.	
	Connect to VSS.	———	
P30/INT0	Open.	"0" is set to output latch.	
	Connect to Vss.	———	
P31/INT1	Open.	"0" is set to output latch.	
	Connect to Vss.	———	
P32, P33	Open.	———	
	Connect to Vss.	———	
P40/AIN4–P43/AIN7	Open.	———	
	Connect to Vss.	———	
P50–P53	Open.	———	
	Connect to Vss.	N-channel open-drain is selected for the output structure.	
P60/AIN0–P63/AIN3	Open.	———	
	Connect to Vss.	———	

Notes 1: After system is released from reset, the internal oscillation (on-chip oscillator) is selected for system clock (RG0=0, MR0=1).

2: When the CRCK instruction is executed, the RC oscillation circuit becomes valid. Be careful that the swich of system clock is not executed at oscillation start only by the CRCK instruction execution.
In order to start oscillation, setting the main clock f(XIN) oscillation to be valid (MR1=0) is required. (If necessary, generate the oscillation stabilizing wait time by software.)
Also, when the main clock (f(XIN)) is selected as system clock, set the main clock f(XIN) oscillation (MR1=0) to be valid, and select main clock f(XIN) (MR0=0). Be careful that the switch of system clock cannot be executed at the same time when main clock oscillation is started.

3: In order to use the external clock input for the main clock, select the ceramic resonance by executing the CMCK instruction at the beggining of software, and then set the main clock (f(XIN)) oscillation to be valid (MR1=0). Until the main clock (f(XIN)) oscillation becomes valid (MR1=0) after ceramic resonance becomes valid, XIN pin is fixed to "H". When an external clock is used, insert a 1 kΩ resistor to XIN pin in series for limits of current.

4: Be sure to select the output structure of ports D0–D5 and the pull-up function of P00–P03 and P10–P13 with every one port. Set the corresponding bits of registers for each port.

5: Be sure to select the output structure of ports P00–P03 and P10–P13 with every two ports. If only one of the two pins is used, leave another one open.

6: The key-on wakeup function is selected with every two bits. When only one of key-on wakeup function is used, considering that the value of key-on wake-up control register K1, set the unused 1-bit to "H" input (turn pull-up transistor ON and open) or "L" input (connect to VSS, or open and set the output latch to "0").

7: The key-on wakeup function is selected with every two bits. When one of key-on wakeup function is used, turn pull-up transistor of unused one ON and open.

(Note when connecting to VSS and VDD)

● Connect the unused pins to VSS and VDD using the thickest wire at the shortest distance against noise.

2.2 Interrupts

The 4519 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, and the INT0 pin input is controlled by the bit 3 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, and the INT1 pin input is controlled by the bit 3 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 completion of 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 completion of 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 while the INTE flag is "1", 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 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.

(3) Interrupt control register V1

Table 2.2.1 shows the interrupt control 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 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) (**Note 2**)	
$V1_2$	Timer 1 interrupt enable bit	0	Interrupt disabled (**SNZT1** instruction is valid)	
		1	Interrupt enabled (**SNZT1** instruction is invalid) (**Note 2**)	
$V1_1$	External 1 interrupt enable bit	0	Interrupt disabled (**SNZ1** instruction is valid)	
		1	Interrupt enabled (**SNZ1** instruction is invalid) (**Note 2**)	
$V1_0$	External 0 interrupt enable bit	0	Interrupt disabled (**SNZ0** instruction is valid)	
		1	Interrupt enabled (**SNZ0** instruction is invalid) (**Note 2**)	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: These instructions are equivalent to the **NOP** instruction.

(4) Interrupt control register V2

Table 2.2.2 shows the interrupt control 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 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 (**Note 2**)	0	Interrupt disabled (**SNZSI** instruction is valid)	
		1	Interrupt enabled (**SNZSI** instruction is invalid) (**Note 2**)	
$V2_2$	A/D interrupt enable bit	0	Interrupt disabled (**SNZAD** instruction is valid)	
		1	Interrupt enabled (**SNZAD** instruction is invalid) (**Note 2**)	
$V2_1$	Timer 4 interrupt enable bit	0	Interrupt disabled (**SNZT4** instruction is valid)	
		1	Interrupt enabled (**SNZT4** instruction is invalid) (**Note 2**)	
$V2_0$	Timer 3 interrupt enable bit	0	Interrupt disabled (**SNZT3** instruction is valid)	
		1	Interrupt enabled (**SNZT3** instruction is invalid) (**Note 2**)	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: These instructions are equivalent to the **NOP** instruction.

(5) Interrupt control register I1

Table 2.2.3 shows the interrupt control register I1.
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 Interrupt control register I1

Interrupt control register I1		at reset : 0000_2	at RAM back-up : state retained	R/W
$I1_3$	INT0 pin input control bit (**Note 2**)	0	INT0 pin input disabled	
		1	INT0 pin input enabled	
$I1_2$	Interrupt valid waveform for INT0 pin/return level selection bit (**Note 2**)	0	Falling waveform /"L" level ("L" level is recognized with the **SNZI0** instruction)	
		1	Rising waveform /"H" level ("H" level is recognized with the **SNZI0** instruction)	
$I1_1$	INT0 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I1_0$	INT0 pin Timer 1 count start synchronous circuit selection bit	0	Timer 1 count start synchronous circuit not selected	
		1	Timer 1 count start synchronous circuit selected	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: When the contents of $I1_2$ and $I1_3$ are changed, the external interrupt request flag EXF0 may be set to "1". Accordingly, clear EXF0 flag with the **SNZ0** instruction when the bit 0 ($V1_0$) of register V1 to "0". In this time, set the **NOP** instruction after the **SNZ0** instruction, for the case when a skip is performed with the **SNZ0** instruction.

(6) Interrupt control register I2

Table 2.2.4 shows the interrupt control register I2.

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 Interrupt control register I2

Interrupt control register I2		at reset : 0000_2	at RAM back-up : state retained	R/W
$I2_3$	INT1 pin input control bit (**Note 2**)	0	INT1 pin input disabled	
		1	INT1 pin input enabled	
$I2_2$	Interrupt valid waveform for INT1 pin/return level selection bit (**Note 2**)	0	Falling waveform /"L" level ("L" level is recognized with the **SNZI1** instruction)	
		1	Rising waveform /"H" level ("H" level is recognized with the **SNZI1** instruction)	
$I2_1$	INT1 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I2_0$	INT1 pin Timer 3 count start synchronous circuit selection bit	0	Timer 3 count start synchronous circuit not selected	
		1	Timer 3 count start synchronous circuit selected	

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

2: When the contents of $I2_2$ and $I2_3$ are changed, the external interrupt request flag EXF1 may be set to "1". Accordingly, clear EXF1 flag with the **SNZ1** instruction when the bit 1 ($V1_1$) of register V1 to "0". In this time, set the **NOP** instruction after the **SNZ1** instruction, for the case when a skip is performed 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 falling edge (“H”→“L”), rising edge (“L”→“H”) and both edges (“H”→“L” or “L”→“H”).

Outline: An external 0 interrupt can be used by dealing with the falling edge (“H”→“L”), rising edge (“L”→“H”) and both edges (“H”→“L” or “L”→“H”) as a trigger.
Specifications: An interrupt occurs by the change of an external signal edge (both edges: “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 falling edge (“H”→“L”), rising edge (“L”→“H”) and both edges (“H”→“L” or “L”→“H”).

Outline: An external 1 interrupt can be used by dealing with the falling edge (“H”→“L”), rising edge (“L”→“H”) and both edges (“H”→“L” or “L”→“H”) as a trigger.
Specifications: An interrupt occurs by the change of an external signal edge (falling edge: “H”→“L”).

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: Timer 1 divides the system clock frequency = 2.0 MHz, and the timer 1 interrupt occurs every 0.25 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 and prescaler divide the system clock frequency (= 4.0 MHz), and the timer 2 interrupt occurs every 1 ms.

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 = 6.0 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: Timer 4 and prescaler divide the system clock frequency (= 4.0 MHz), and the timer 4 interrupt occurs every 50 ms.

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

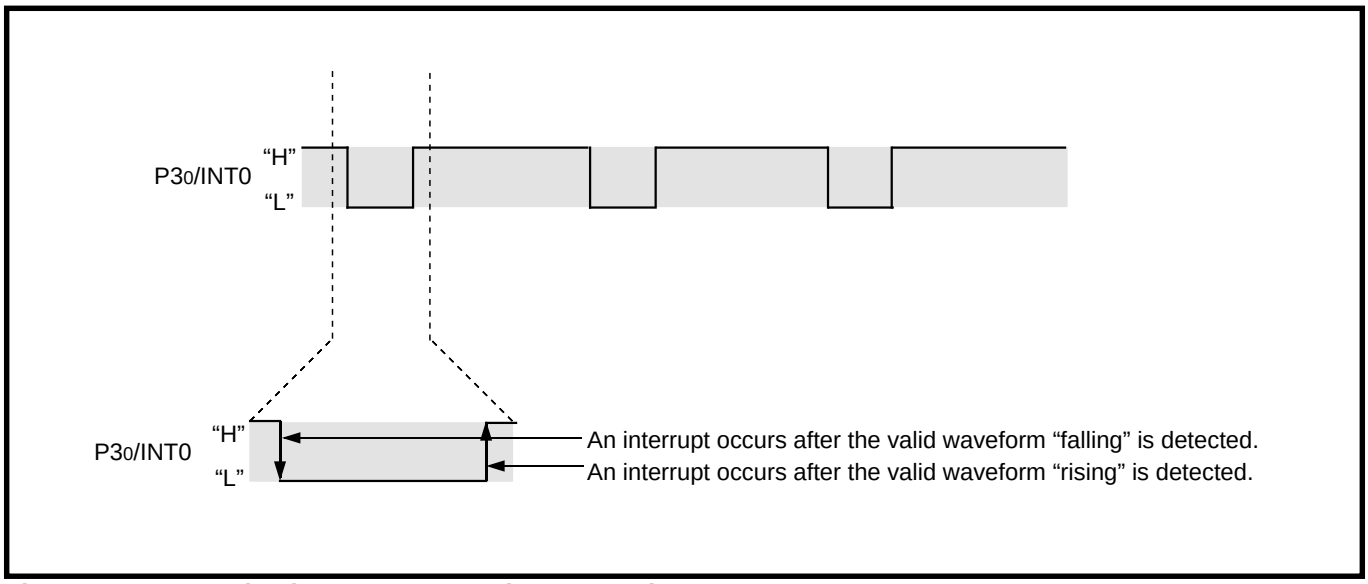


Fig. 2.2.1 External 0 interrupt operation example

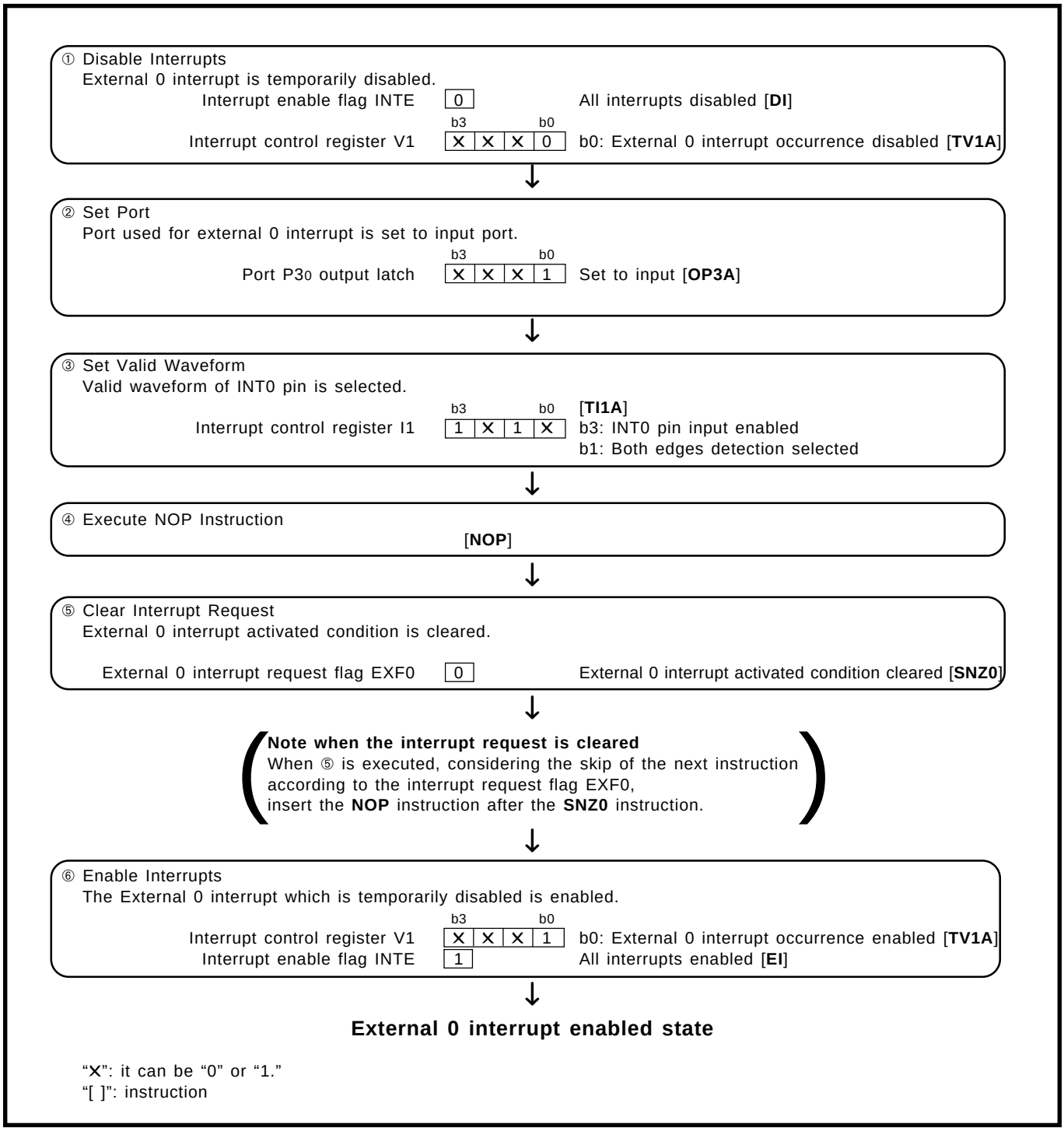


Fig. 2.2.2 External 0 interrupt setting example

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

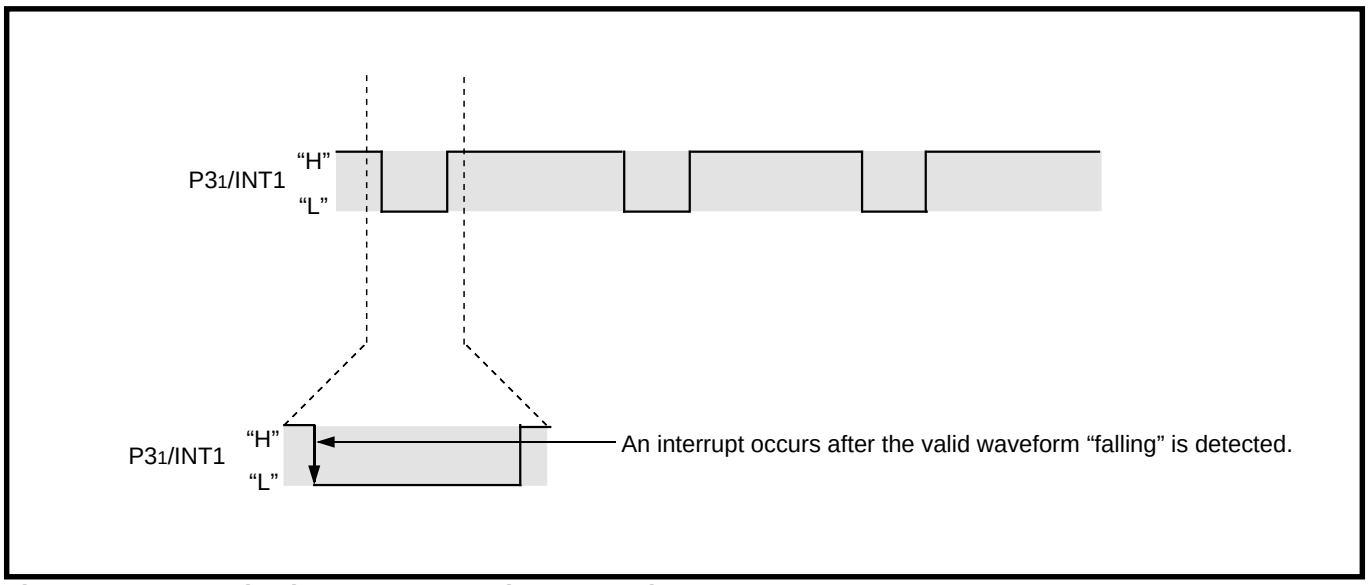


Fig. 2.2.3 External 1 interrupt operation example

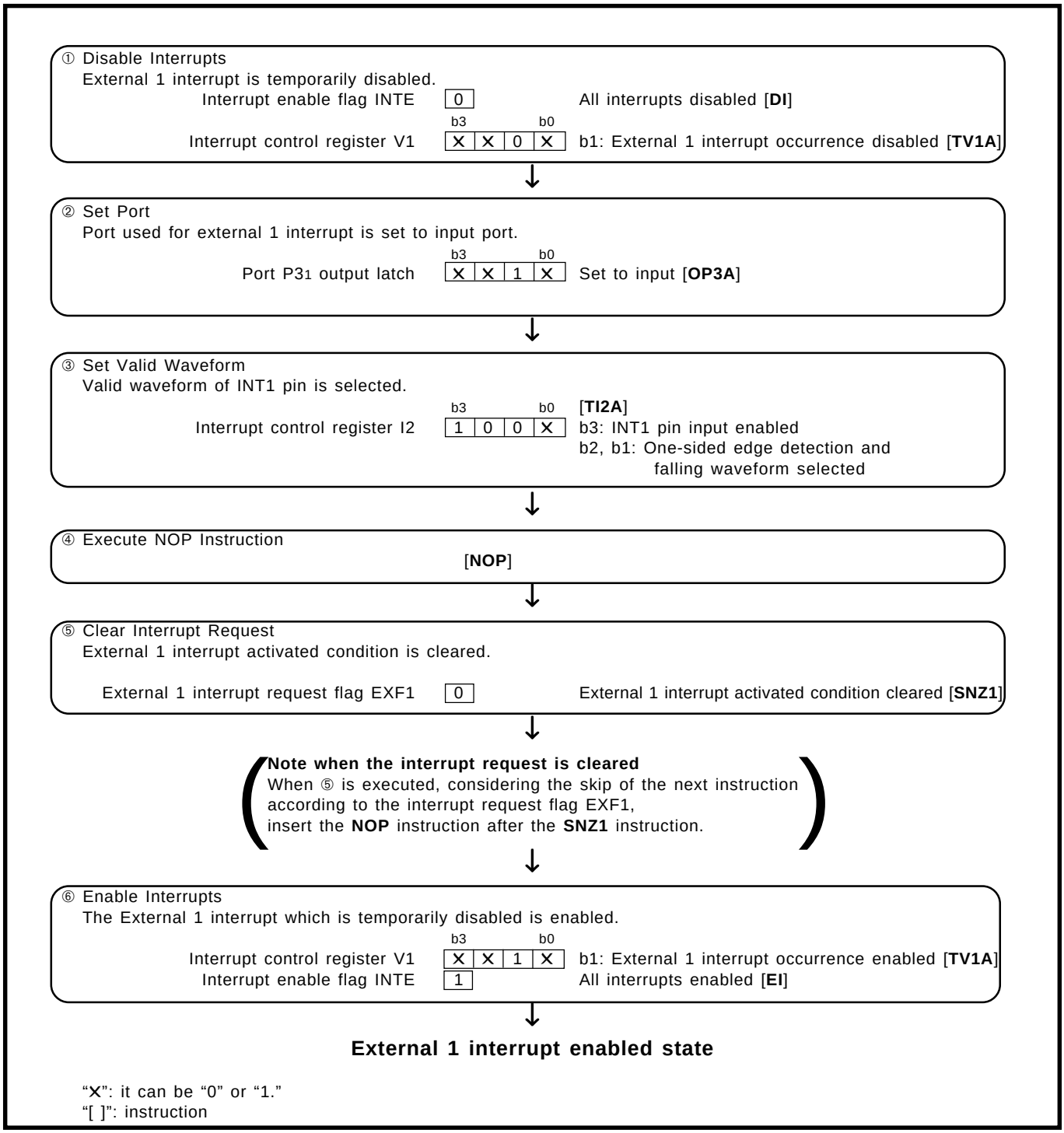


Fig. 2.2.4 External 1 interrupt setting example

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

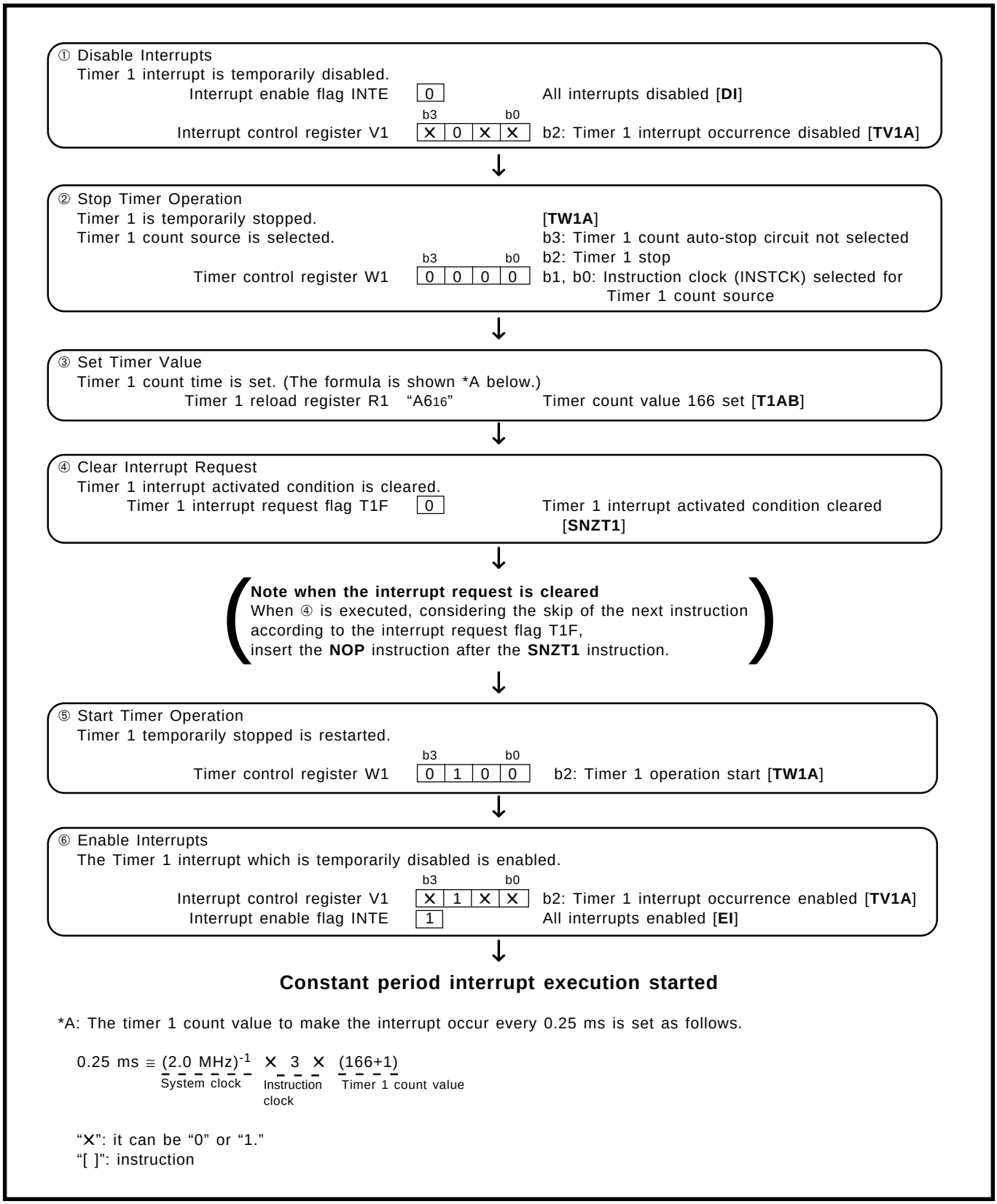
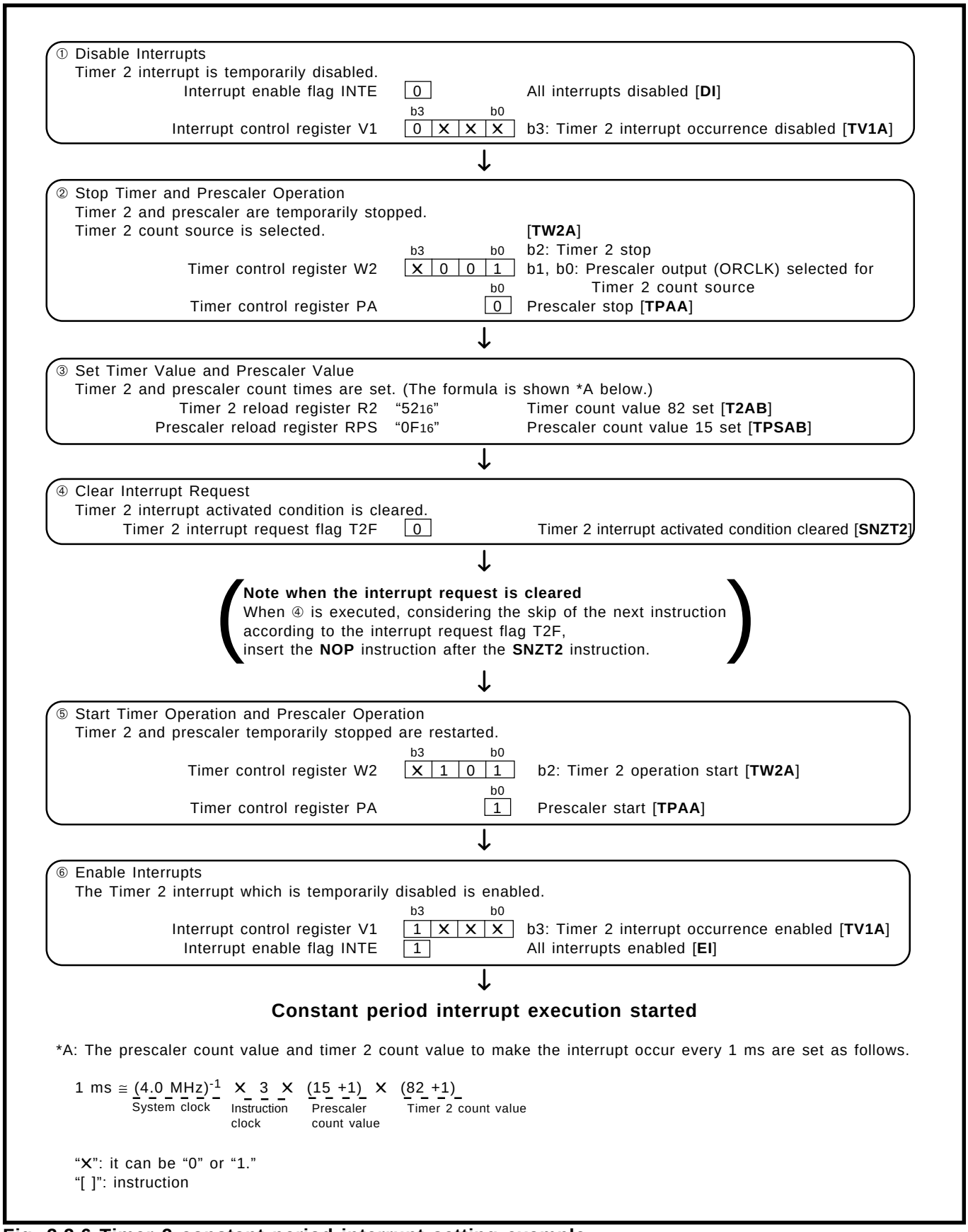


Fig. 2.2.5 Timer 1 constant period interrupt setting example



*A: The prescaler count value and timer 2 count value to make the interrupt occur every 1 ms are set as follows.

$$1\ \text{ms} \cong \underbrace{(4.0\ \text{MHz})^{-1}}_{\text{System clock}} \times \underbrace{3}_{\substack{\text{Instruction}\\ \text{clock}}} \times \underbrace{(15+1)}_{\substack{\text{Prescaler}\\ \text{count value}}} \times \underbrace{(82+1)}_{\text{Timer 2 count value}}$$

"X": it can be "0" or "1."
"[ ]": instruction

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**]
Interrupt control register V2	b3 X X X b0 = X X X 0	b0: Timer 3 interrupt occurrence disabled [**TV2A**]

↓

② Stop Timer Operation
Timer 3 and prescaler are temporarily stopped.
Timer 3 count source is selected.

Timer control register W3	b3 b0 = 0 0 0 1	[**TW3A**] b3: Timer 3 count auto-stop circuit not selected b2: Timer 3 stop b1, b0: Prescaler output (ORCLK) selected for Timer 3 count source
Timer control register PA	b0 = 0	Prescaler stop [**TPAA**]

↓

③ Set Timer Value and Prescaler Value
Timer 3 and prescaler count times are set. (The formula is shown *A below.)

Timer 3 reload register R3	"EF_{16}"	Timer count value 239 set [**T3AB**]
Prescaler reload register RPS	"$F9_{16}$"	Prescaler count value 249 set [**TPSAB**]

↓

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**]

↓

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 Operation and Prescaler Operation
Timer 3 and prescaler temporarily stopped are restarted.

Timer control register W3	b3 b0 = 0 1 0 1	b2: Timer 3 operation start [**TW3A**]
Timer control register PA	b0 = 1	Prescaler start [**TPAA**]

↓

⑥ Enable Interrupts
The Timer 3 interrupt which is temporarily disabled is enabled.

Interrupt control register V2	b3 b0 = X X X 1	b0: Timer 3 interrupt occurrence enabled [**TV2A**]
Interrupt enable flag INTE	1	All interrupts enabled [**EI**]

↓

Constant period interrupt execution started

*A: The prescaler count value and timer 3 count value to make the interrupt occur every 30 ms are set as follows.

$$30\ \text{ms} \cong (6.0\ \text{MHz})^{-1} \times 3 \times (249+1) \times (239+1)$$

(6.0 MHz)$^{-1}$: System clock; 3: Instruction clock; (249+1): Prescaler count value; (239+1): Timer 3 count value

"X": it can be "0" or "1."
"[ ]": instruction

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**]
Interrupt control register V2 [X X 0 X] (b3 b0) b1: Timer 4 interrupt occurrence disabled [**TV2A**]

↓

② Stop Timer and Prescaler Operation [**TW4A**]
Timer 4 and prescaler are temporarily stopped.
Timer 4 count source is selected.
Timer control register W4 [0 0 0 1] (b3 b0)
b3: CNTR1 input
b2: PWM signal "H" interval expansion function invalid
b1: Timer 4 stop
b0: Prescaler output (ORCLK) divided by 2 selected for Timer 4 count source
Timer control register PA [0] (b0) Prescaler stop [**TPAA**]

↓

③ Set Timer Value and Prescaler Value
Timer 4 and prescaler count times are set. (The formula is shown *A below.)
Timer 4 reload register R4L "DD_{16}" Timer count value 221 set [**T4AB**]
Prescaler reload register RPS "95_{16}" Prescaler count value 149 set [**TPSAB**]

↓

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**]

↓

Note when the interrupt request is cleared
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 Operation and Prescaler Operation
Timer 4 and prescaler temporarily stopped are restarted.
Timer control register W4 [0 0 1 1] (b3 b0) b1: Timer 4 operation start [**TW4A**]
Timer control register PA [1] (b0) Prescaler start [**TPAA**]

↓

⑥ Enable Interrupts
The Timer 4 interrupt which is temporarily disabled is enabled.
Interrupt control register V2 [X X 1 X] (b3 b0) b1: Timer 4 interrupt occurrence enabled [**TV2A**]
Interrupt enable flag INTE [1] All interrupts enabled [**EI**]

↓

Constant period interrupt execution started

*A: The prescaler count value and timer 4 count value to make the interrupt occur every 50 ms are set as follows.

$$50\ \text{ms} \cong \underbrace{(4.0\ \text{MHz})^{-1}}_{\text{System clock}} \times \underbrace{3}_{\text{Instruction clock}} \times \underbrace{(149+1)}_{\text{Prescaler count value}} \times \underbrace{2}_{\text{Timer 4 count source}} \times \underbrace{(221+1)}_{\text{Timer 4 count value}}$$

"X": it can be "0" or "1."
"[ ]": instruction

Fig. 2.2.8 Timer 4 constant period interrupt setting example

2.2.4 Notes on use

(1) Setting of INT0 interrupt valid waveform

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.
Depending on the input state of $P3_0$/INT0 pin, the external interrupt request flag (EXF0) may be set to "1" when the bit 2 of register I1 is changed.

(2) Setting of INT0 pin input control

Set a value to the bit 3 of register I1, and execute the **SNZ0** instruction to clear the EXF0 flag to "0" after executing at least one instruction.
Depending on the input state of $P3_0$/INT0 pin, the external interrupt request flag (EXF0) may be set to "1" when the bit 3 of register I1 is changed.

(3) Setting of INT1 interrupt valid waveform

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.
Depending on the input state of $P3_1$/INT1 pin, the external interrupt request flag (EXF1) may be set to "1" when the bit 2 of register I2 is changed.

(4) Setting of INT1 pin input control

Set a value to the bit 3 of register I2, and execute the **SNZ1** instruction to clear the EXF1 flag to "0" after executing at least one instruction.
Depending on the input state of $P3_1$/INT1 pin, the external interrupt request flag (EXF1) may be set to "1" when the bit 3 of register I2 is changed.

(5) Multiple interrupts

Multiple interrupts cannot be used in the 4519 Group.

(6) 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.

(7) $P3_0$/INT0 pin

When the external interrupt input pin INT0 is used, set the bit 3 of register I1 to "1".
Even in this case, port $P3_0$ I/O function is valid.
Also, the EXF0 flag is set to "1" when bit 3 of register I1 is set to "1" by input of a valid waveform (valid waveform causing external 0 interrupt) even if it is used as an I/O port $P3_0$.
The input threshold characteristics (V_{IH}/V_{IL}) are different between INT0 pin input and port $P3_0$ input. Accordingly, note this difference when INT0 pin input and port $P3_0$ input are used at the same time.

(8) $P3_1$/INT1 pin

When the external interrupt input pin INT1 is used, set the bit 3 of register I2 to "1".
Even in this case, port $P3_1$ I/O function is valid.
Also, the EXF1 flag is set to "1" when bit 3 of register I2 is set to "1" by input of a valid waveform (valid waveform causing external 1 interrupt) even if it is used as an I/O port $P3_1$.
The input threshold characteristics (V_{IH}/V_{IL}) are different between INT1 pin input and port $P3_1$ input. Accordingly, note this difference when INT1 pin input and port $P3_1$ input are used at the same time.

(9) POF instruction

When the **POF** instruction is executed continuously after the **EPOF** instruction, system enters the RAM back-up state.
Note that system cannot enter the RAM back-up state when executing only the **POF** instruction.
Be sure to disable interrupts by executing the **DI** instruction before executing the **EPOF** instruction and the **POF** instruction continuously.

2.3 Timers

The 4519 Group has four 8-bit timers (each has a reload register) and 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**
(Timer 4 has the PWM output function)

(5) Watchdog timer

■ **Watchdog function**
Watchdog timer provides a method to reset the system when a program run-away occurs.
System operates after it is released from reset. When the timer count value underflows, the WDF1 flag is set to “1.” Then, if the **WRST** instruction is never executed until timer WDT counts 65534, WDF2 flag is set to “1,” and system reset occurs.
When the **DWDT** instruction and the **WRST** instruction are executed continuously, the watchdog timer function is invalid.
The **WRST** instruction has the skip function. When the **WRST** instruction is executed while the WDF1 flag is “1”, the next instruction is skipped and then, the WDF1 flag is cleared to “0”.

2.3.2 Related registers

(1) Interrupt control register V1

Table 2.3.1 shows the interrupt control 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 2.3.1 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) (**Note 2**)	
$V1_2$	Timer 1 interrupt enable bit	0	Interrupt disabled (**SNZT1** instruction is valid)	
		1	Interrupt enabled (**SNZT1** instruction is invalid) (**Note 2**)	
$V1_1$	External 1 interrupt enable bit	0	Interrupt disabled (**SNZ1** instruction is valid)	
		1	Interrupt enabled (**SNZ1** instruction is invalid) (**Note 2**)	
$V1_0$	External 0 interrupt enable bit	0	Interrupt disabled (**SNZ0** instruction is valid)	
		1	Interrupt enabled (**SNZ0** instruction is invalid) (**Note 2**)	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: These instructions are equivalent to the **NOP** instruction.
3: When timer is used, $V1_1$ and $V1_0$ are not used.

(2) Interrupt control register V2

Table 2.3.2 shows the interrupt control 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 2.3.2 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) (**Note 2**)	
$V2_2$	A/D interrupt enable bit	0	Interrupt disabled (**SNZAD** instruction is valid)	
		1	Interrupt enabled (**SNZTAD** instruction is invalid) (**Note 2**)	
$V2_1$	Timer 4 interrupt enable bit	0	Interrupt disabled (**SNZT4** instruction is valid)	
		1	Interrupt enabled (**SNZT4** instruction is invalid) (**Note 2**)	
$V2_0$	Timer 3 interrupt enable bit	0	Interrupt disabled (**SNZT3** instruction is valid)	
		1	Interrupt enabled (**SNZT3** instruction is invalid) (**Note 2**)	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: These instructions are equivalent to the **NOP** instruction.
3: When timer is used, $V2_3$ and $V2_2$ is not used.

(3) Interrupt control register I1

Table 2.3.3 shows the interrupt control register I1.
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 2.3.3 Interrupt control register I1

Interrupt control register I1		at reset : 0000_2	at RAM back-up : state retained	R/W
$I1_3$	INT0 pin input control bit (**Note 2**)	0	INT0 pin input disabled	
		1	INT0 pin input enabled	
$I1_2$	Interrupt valid waveform for INT0 pin/return level selection bit (**Note 2**)	0	Falling waveform/"L" level ("L" level is recognized with the **SNZI0** instruction)	
		1	Rising waveform/"H" level ("H" level is recognized with the **SNZI0** instruction)	
$I1_1$	INT0 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I1_0$	INT0 pin Timer 1 count start synchronous circuit selection bit	0	Timer 1 count start synchronous circuit not selected	
		1	Timer 1 count start synchronous circuit selected	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: When the contents of $I1_2$ and $I1_3$ are changed, the external interrupt request flag EXF0 may be set. Accordingly, clear EXF0 flag with the **SNZ0** instruction when the bit 0 ($V1_0$) of register V1 to "0". In this time, set the **NOP** instruction after the **SNZ0** instruction, for the case when a skip is performed with the **SNZ0** instruction.

(4) Interrupt control register I2

Table 2.3.4 shows the interrupt control register I2.
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 2.3.4 Interrupt control register I2

Interrupt control register I2		at reset : 0000_2	at RAM back-up : state retained	R/W
$I2_3$	INT1 pin input control bit (**Note 2**)	0	INT1 pin input disabled	
		1	INT1 pin input enabled	
$I2_2$	Interrupt valid waveform for INT1 pin/return level selection bit (**Note 2**)	0	Falling waveform/"L" level ("L" level is recognized with the **SNZI1** instruction)	
		1	Rising waveform/"H" level ("H" level is recognized with the **SNZI1** instruction)	
$I2_1$	INT1 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I2_0$	INT1 pin Timer 3 count start synchronous circuit selection bit	0	Timer 3 count start synchronous circuit not selected	
		1	Timer 3 count start synchronous circuit selected	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: When the contents of $I2_2$ and $I2_3$ are changed, the external interrupt request flag EXF1 may be set. Accordingly, clear EXF1 flag with the **SNZ1** instruction when the bit 1 ($V1_1$) of register V1 to "0". In this time, set the **NOP** instruction after the **SNZ1** instruction, for the case when a skip is performed with the **SNZ1** instruction.

(5) Timer control register PA

Table 2.3.5 shows the timer control register PA.
Set the contents of this register through register A with the **TPAA** instruction.

Table 2.3.5 Timer control register PA

Timer control register PA		at reset : 0_2		at RAM back-up : state retained	W
PA_0	Prescaler control bit	0	Stop (state initialized)		
		1	Operating		

Note: "W" represents write enabled.

(6) Timer control register W1

Table 2.3.6 shows the timer control register W1.
Set the contents of this register through register A with the **TW1A** instruction.
In addition, the **TAW1** instruction can be used to transfer the contents of register W1 to register A.

Table 2.3.6 Timer control register W1

Timer control register W1		at reset : 0000_2		at RAM back-up : state retained	R/W
$W1_3$	Timer 1 count auto-stop circuit control bit (**Note 2**)	0	Timer 1 count auto-stop circuit not selected		
		1	Timer 1 count auto-stop circuit selected		
$W1_2$	Timer 1 control bit	0	Stop (state retained)		
		1	Operating		
$W1_1$	Timer 1 count source selection bits	$W1_1$	$W1_0$	Count source	
		0	0	Instruction clock (INSTCK)	
$W1_0$		0	1	Prescaler output (ORCLK)	
		1	0	X_{IN} input	
		1	1	CNTR0 input	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: This function is valid only when the timer 1 count start synchronous circuit is selected ($I1_0$="1").

(7) Timer control register W2

Table 2.3.7 shows the timer control register W2.
Set the contents of this register through register A with the **TW2A** instruction.
In addition, the **TAW2** instruction can be used to transfer the contents of register W2 to register A.

Table 2.3.7 Timer control register W2

Timer control register W2		at reset : 0000_2		at RAM back-up : state retained	R/W
$W2_3$	CNTR0 output selection bit	0	Timer 1 underflow signal divided by 2 output		
		1	Timer 2 underflow signal divided by 2 output		
$W2_2$	Timer 2 control bit	0	Stop (state retained)		
		1	Operating		
$W2_1$	Timer 2 count source selection bits	$W2_1$	$W2_0$	Count source	
		0	0	System clock (STCK)	
$W2_0$		0	1	Prescaler output (ORCLK)	
		1	0	Timer 1 underflow signal (T1UDF)	
		1	1	PWM signal (PWMOUT)	

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

(8) Timer control register W3

Table 2.3.8 shows the timer control register W3.
Set the contents of this register through register A with the **TW3A** instruction.
In addition, the **TAW3** instruction can be used to transfer the contents of register W3 to register A.

Table 2.3.8 Timer control register W3

Timer control register W3		at reset : 0000_2		at RAM back-up : state retained	R/W
$W3_3$	Timer 3 count auto-stop circuit control bit (**Note 2**)	0		Timer 3 count auto-stop circuit not selected	
		1		Timer 3 count auto-stop circuit selected	
$W3_2$	Timer 3 control bit	0		Stop (state retained)	
		1		Operating	
$W3_1$	Timer 3 count source selection bits	$W3_1$	$W3_0$	Count source	
		0	0	PWM signal (PWMOUT)	
		0	1	Prescaler output (ORCLK)	
$W3_0$		1	0	Timer 2 underflow signal (T2UDF)	
		1	1	CNTR1 input	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: This function is valid only when the timer 3 count start synchronous circuit is selected ($I2_0$="1").

(9) Timer control register W4

Table 2.3.9 shows the timer control register W4.
Set the contents of this register through register A with the **TW4A** instruction.
In addition, the **TAW4** instruction can be used to transfer the contents of register W4 to register A.

Table 2.3.9 Timer control register W4

Timer control register W4		at reset : 0000_2	at RAM back-up : 0000_2	R/W
$W4_3$	D_7/CNTR1 pin function selection bit	0	D_7 (I/O) / CNTR1 (input)	
		1	CNTR1 (I/O) / D_7 (input)	
$W4_2$	PWM signal "H" interval expansion function control bit	0	PWM signal "H" interval expansion function invalid	
		1	PWM signal "H" interval expansion function valid	
$W4_1$	Timer 4 control bit	0	Stop (state retained)	
		1	Operating	
$W4_0$	Timer 4 count source selection bit	0	X_{IN} input	
		1	Prescaler output (ORCLK) divided by 2	

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

(10) Timer control register W5

Table 2.3.10 shows the timer control register W5.
Set the contents of this register through register A with the **TW5A** instruction.
In addition, the **TAW5** instruction can be used to transfer the contents of register W5 to register A.

Table 2.3.10 Timer control register W5

Timer control register W5		at reset : 0000_2		at RAM back-up : state retained	R/W
$W5_3$	Not used	0		This bit has no function, but read/write is enabled.	
		1			
$W5_2$	Period measurement circuit control bit	0		Stop	
		1		Operating	
$W5_1$	Signal for period measurement selection bits	$W5_1$	$W5_0$	Count source	
		0	0	On-chip oscillator (f(RING/16))	
$W5_0$		0	1	$CNTR_0$ pin input	
		1	0	INT0 pin input	
		1	1	Not available	

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

(11) Timer control register W6

Table 2.3.11 shows the timer control register W6.
Set the contents of this register through register A with the **TW6A** instruction.
In addition, the **TAW6** instruction can be used to transfer the contents of register W6 to register A.

Table 2.3.11 Timer control register W6

Timer control register W6		at reset : 0000_2	at RAM back-up : state retained	R/W
$W6_3$	CNTR1 pin input count edge selection bit	0	Falling edge	
		1	Rising edge	
$W6_2$	CNTR0 pin input count edge selection bit	0	Falling edge	
		1	Rising edge	
$W6_1$	CNTR1 output auto-control circuit selection bit	0	CNTR1 output auto-control circuit not selected	
		1	CNTR1 output auto-control circuit selected	
$W6_0$	D_6/CNTR0 pin function selection bit	0	D_6(I/O)/CNTR0 input	
		1	CNTR0 input/output/D_6 (input)	

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 divide the system clock frequency f(XIN) = 4.0 MHz, and the timer 1 interrupt occurs every 3 ms.

Figure 2.3.4 shows the setting example of the constant period measurement.

(2) CNTR0 output operation: buzzer output

Outline: Square wave output from timer 2 can be used for buzzer output.
Specifications: 4 kHz square wave is output from the CNTR0 pin at system clock frequency f(XIN) = 4.0 MHz. Also, timer 2 interrupt occurs simultaneously.

Figure 2.3.1 shows the peripheral circuit example, and Figure 2.3.5 shows the setting example of CNTR0 output.

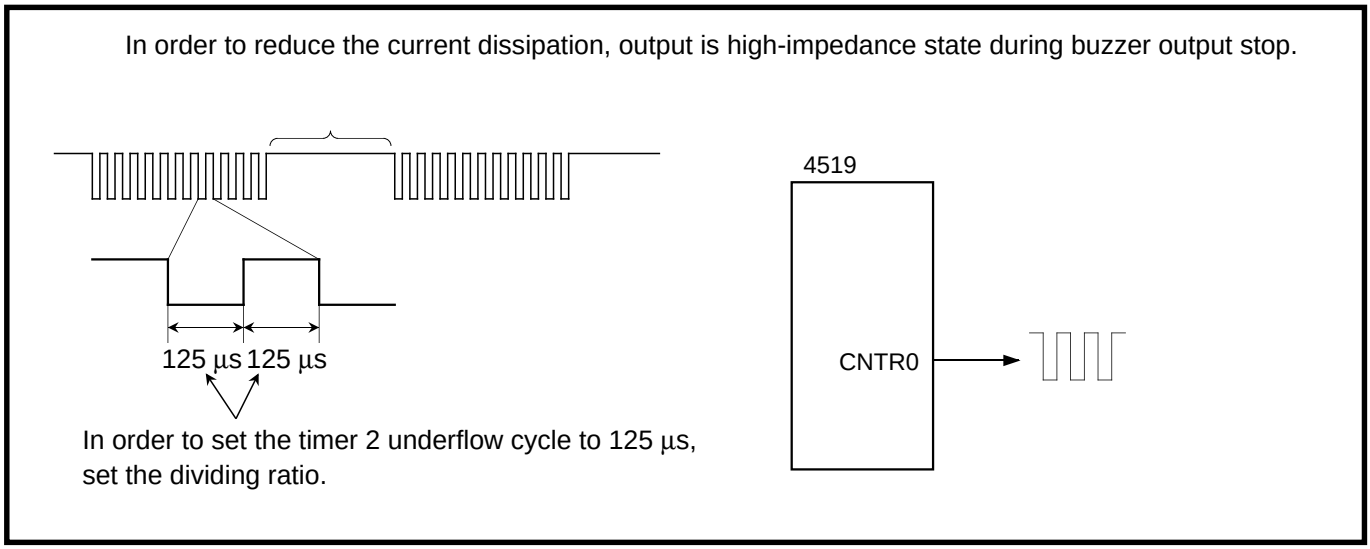


Fig. 2.3.1 Peripheral circuit example

(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 1 count source is input to CNTR0 pin, and the timer 1 interrupt occurs every 100 counts.

Figure 2.3.6 shows the setting example of CNTR0 input.

(4) Timer operation: timer start by external input

Outline: The constant period can be measured by external input.
Specifications: Timer 3 operates by INT1 input as a trigger and an interrupt occurs after 1 ms.

Figure 2.3.7 shows the setting example of timer start.

(5) CNTR1 output control: PWM output control

Outline: The PWM output from CNTR1 pin can be performed by timer 4.
Specifications: Timer 4 divides the main clock frequency f(XIN) = 4.0 MHz and the waveform, which "H" period is 0.875 µs of the 1.875 µs PWM periods, is output from CNTR1 pin.

Figure 2.3.2 shows the timer 4 operation and Figure 2.3.8 shows the setting example of PWM output control.

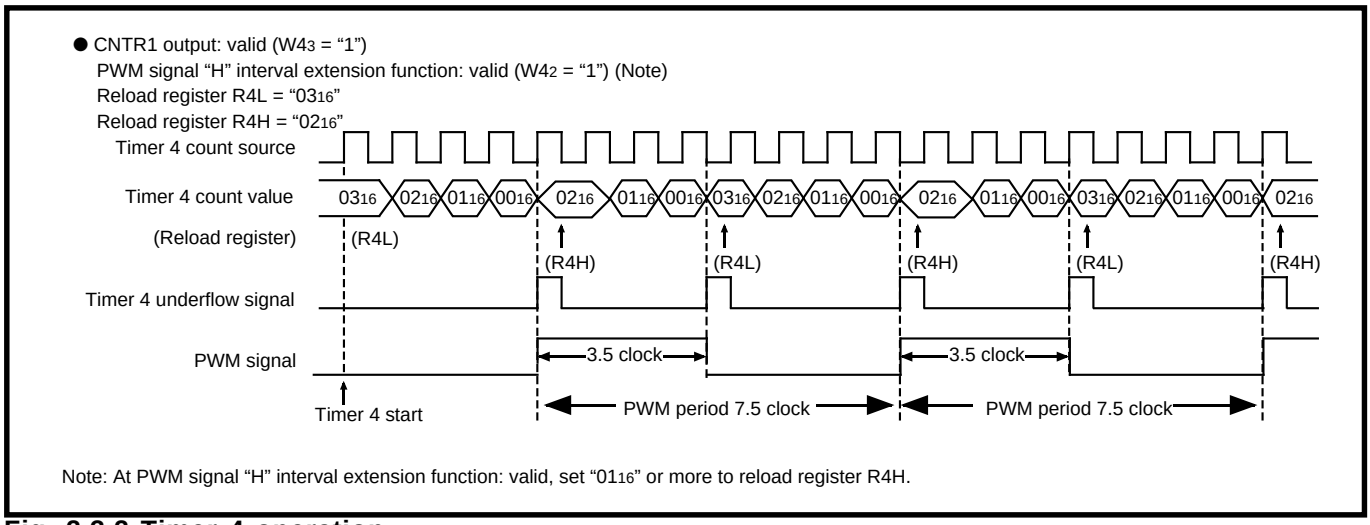


Fig. 2.3.2 Timer 4 operation

(6) Period measurement

Outline: The period of the followings can be measured by timer 1.
- on-chip oscillator divided by 16
- CNTR0 pin input
- INT0 pin input

Specifications: Timer 1 count is performed during one period from the rise of a CNTR0 input to the next rise.
Timer 1 count source is XIN input.

Figure 2.3.9 and Figure 2.3.10 show the setting example of period measurement of a CNTR0 pin input.

(7) Pulse width measurement

Outline: "H" pulse width or "L" pulse width of INT0 pin input can be measured by Timer 1.
Specifications: Timer 1 count is performed during "H" pulse input from the rise of an INT0 input to the next rise.
Timer 1 count source is XIN input.

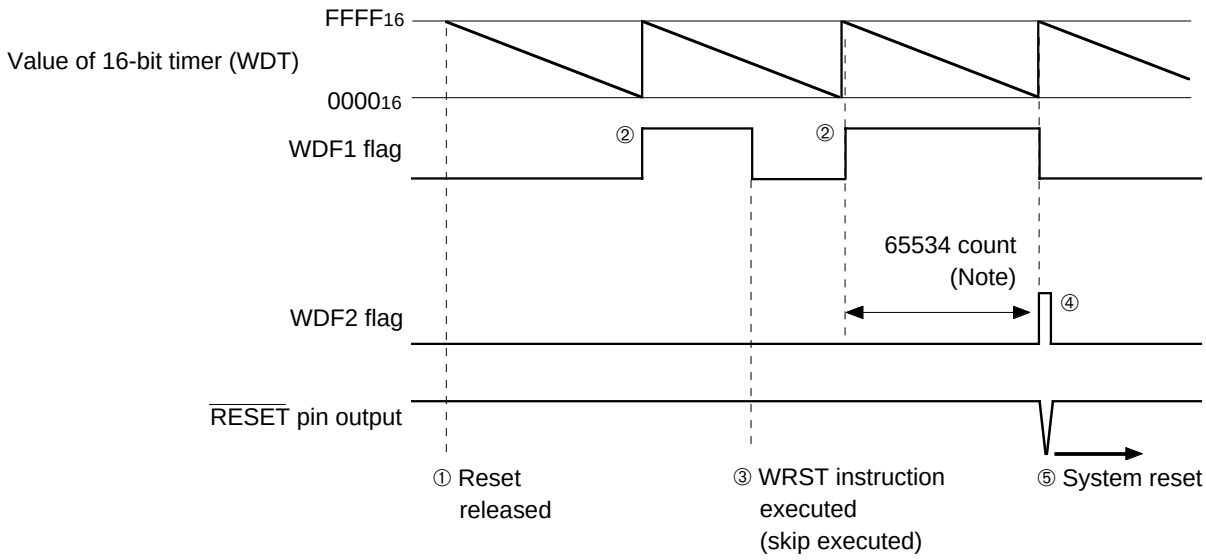
Figure 2.3.11 and Figure 2.3.12 show the setting example of pulse width measurement of an INT0 pin input.

(8) Watchdog timer

Watchdog timer provides a method to reset the system when a program run-away occurs. Accordingly, when the watchdog timer function is set to be valid, execute the **WRST** instruction at a certain period which consists of 16-bit timers' 65534 counts or less (execute **WRST** instruction at less than 65534 machine cycles).

Outline: Execute the **WRST** instruction in 16-bit timer's 65534 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 49 ms.

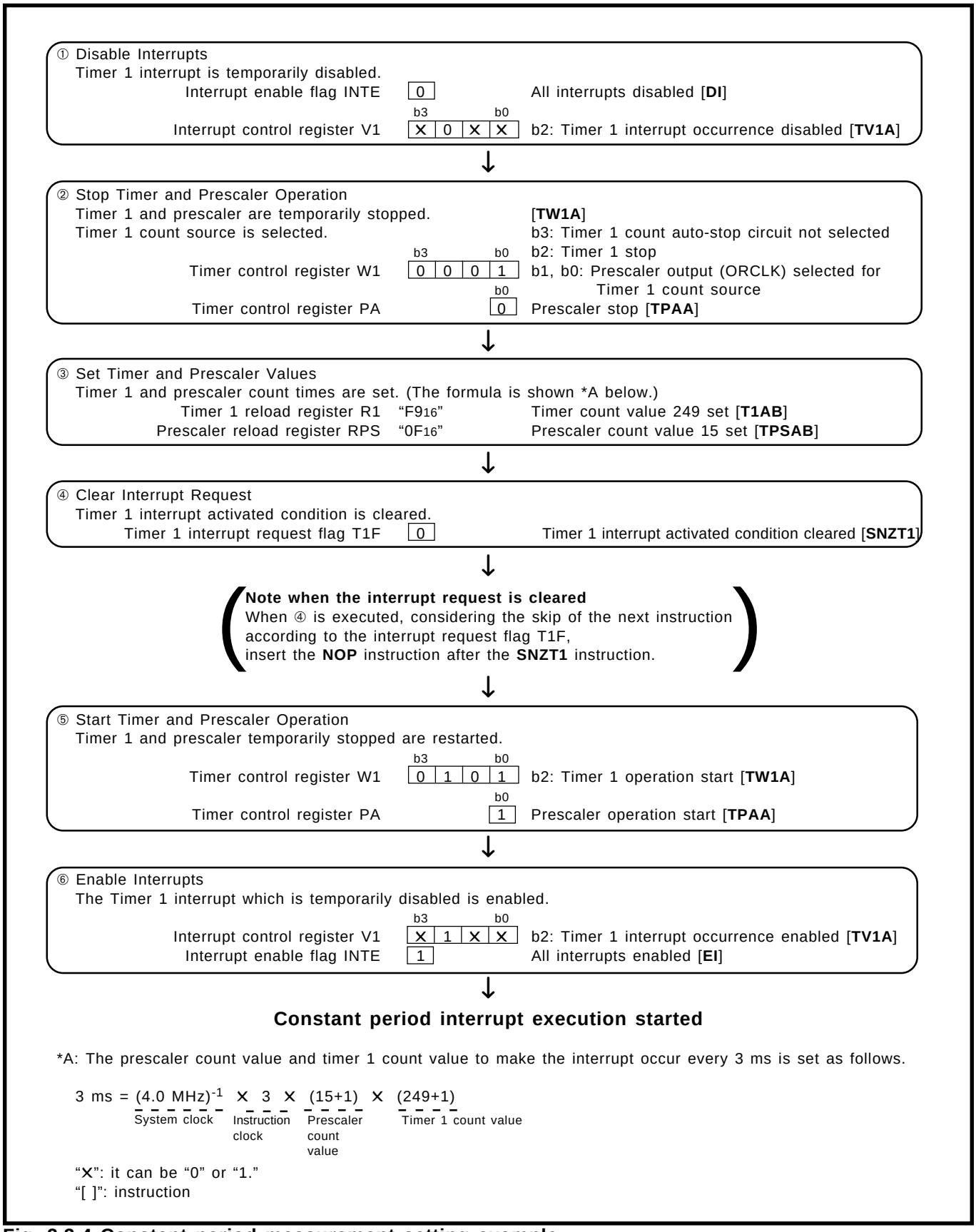
Figure 2.3.3 shows the watchdog timer function, and Figure 2.3.13 shows the example of watchdog timer.



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

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

Fig. 2.3.3 Watchdog timer function



*A: The prescaler count value and timer 1 count value to make the interrupt occur every 3 ms is set as follows.

$$3\ \text{ms} = \underbrace{(4.0\ \text{MHz})^{-1}}_{\text{System clock}} \times \underbrace{3}_{\substack{\text{Instruction}\\ \text{clock}}} \times \underbrace{(15+1)}_{\substack{\text{Prescaler}\\ \text{count}\\ \text{value}}} \times \underbrace{(249+1)}_{\text{Timer 1 count value}}$$

"X": it can be "0" or "1."
"[ ]": instruction

Fig. 2.3.4 Constant period measurement setting example

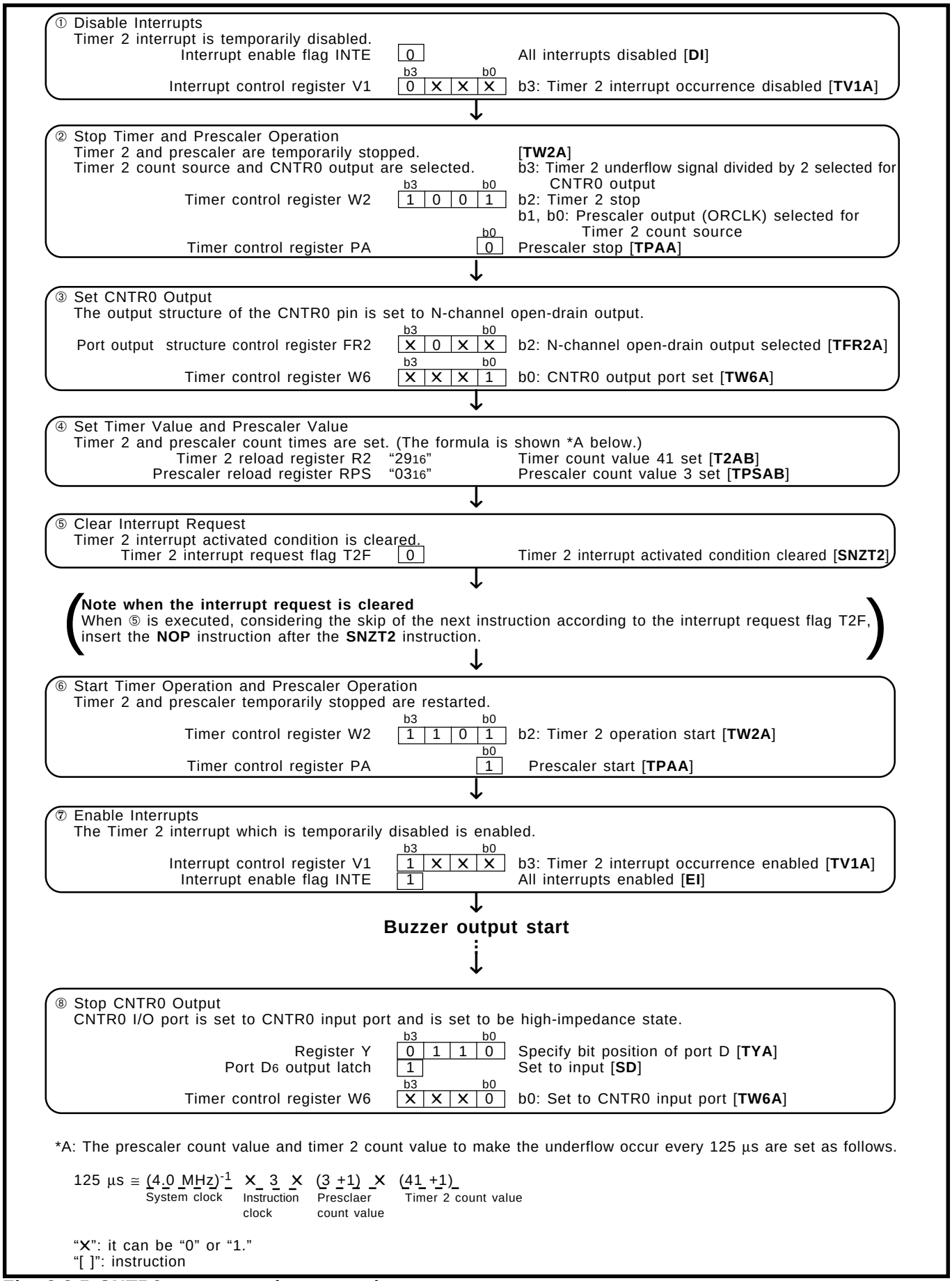


Fig. 2.3.5 CNTR0 output setting example

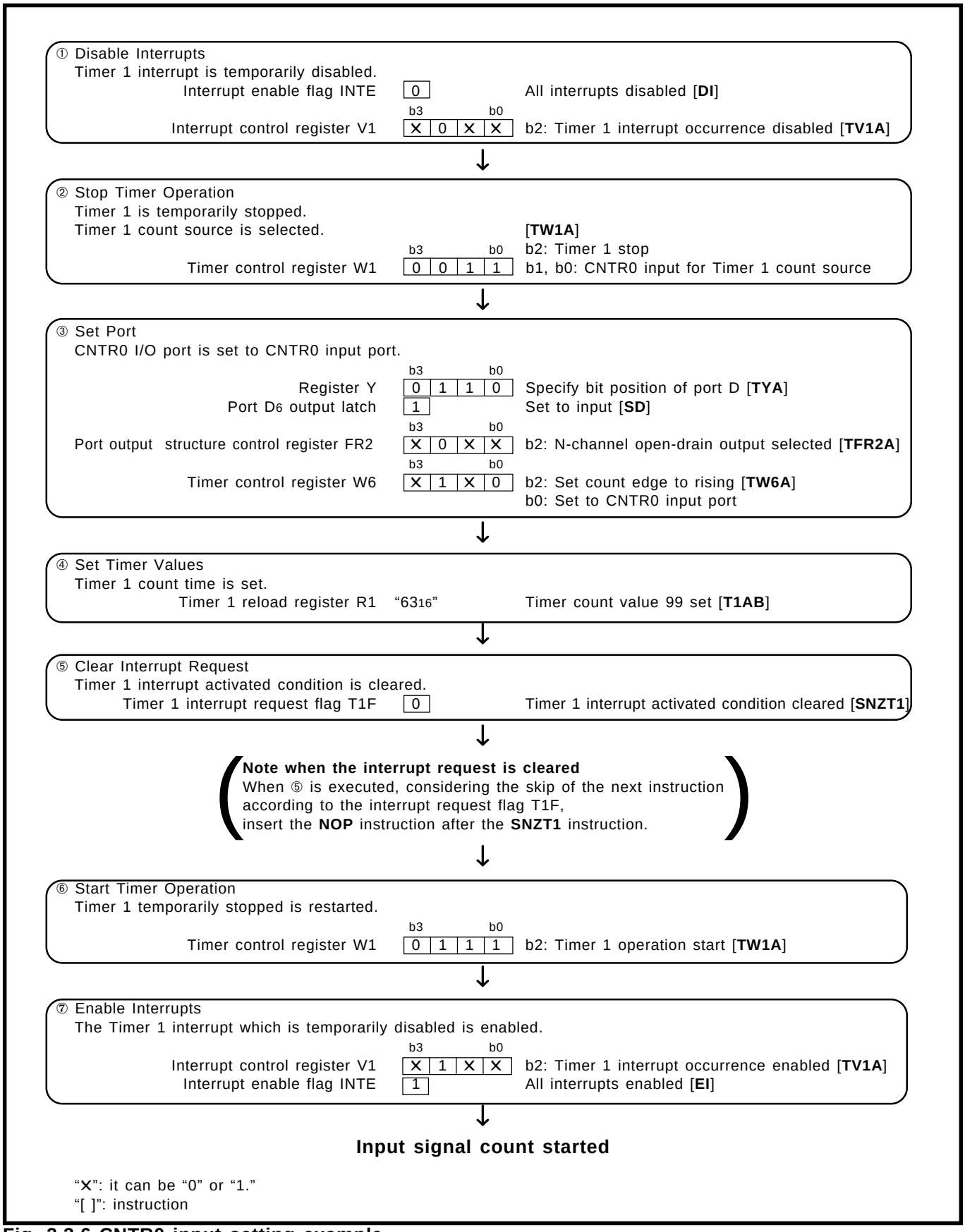


Fig. 2.3.6 CNTR0 input setting example

However, specify the pulse width input to CNTR0 pin, CNTR1 pin. Refer to section "**3.1 Electrical characteristics**" for the timer external input period condition.

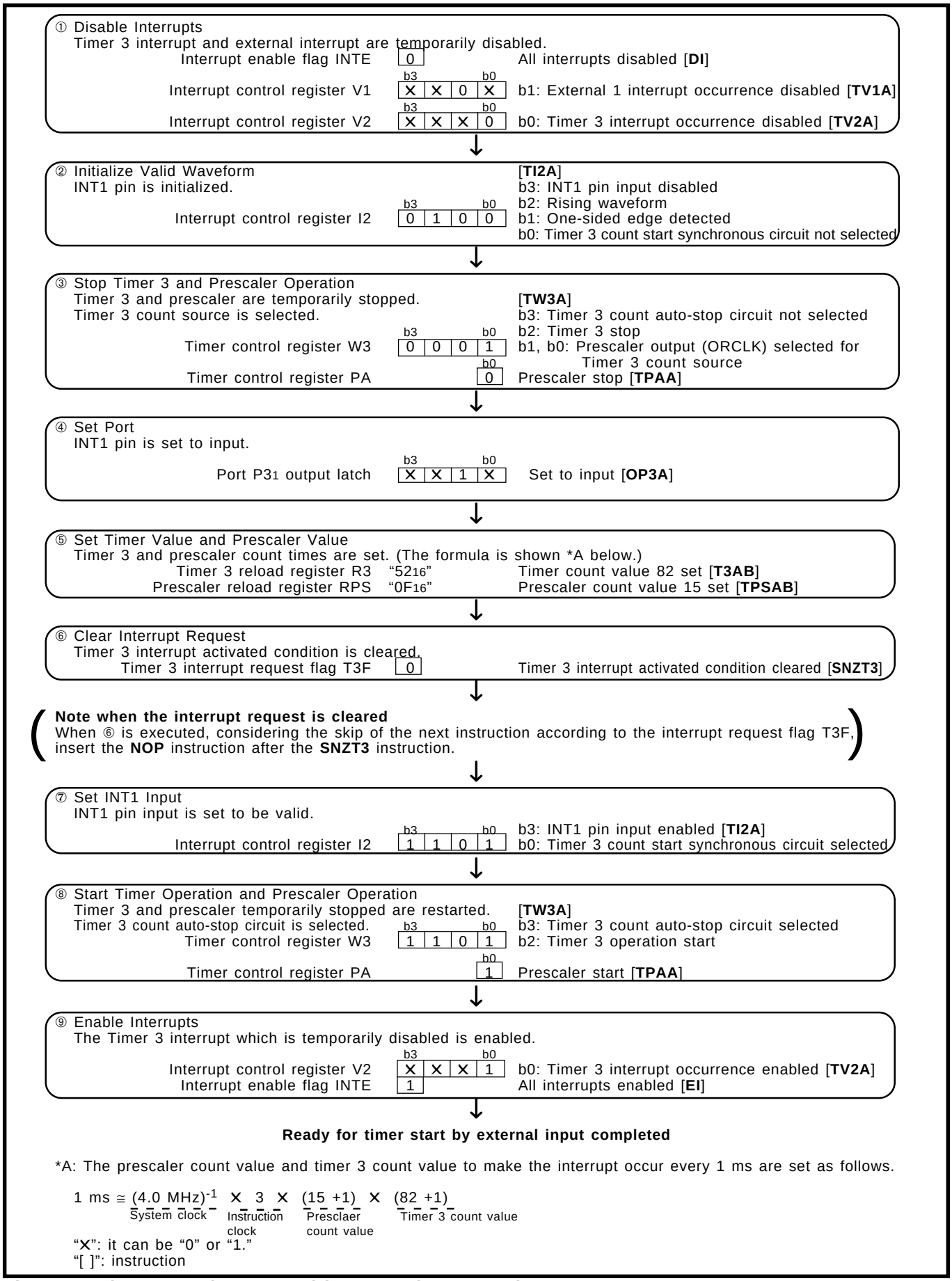
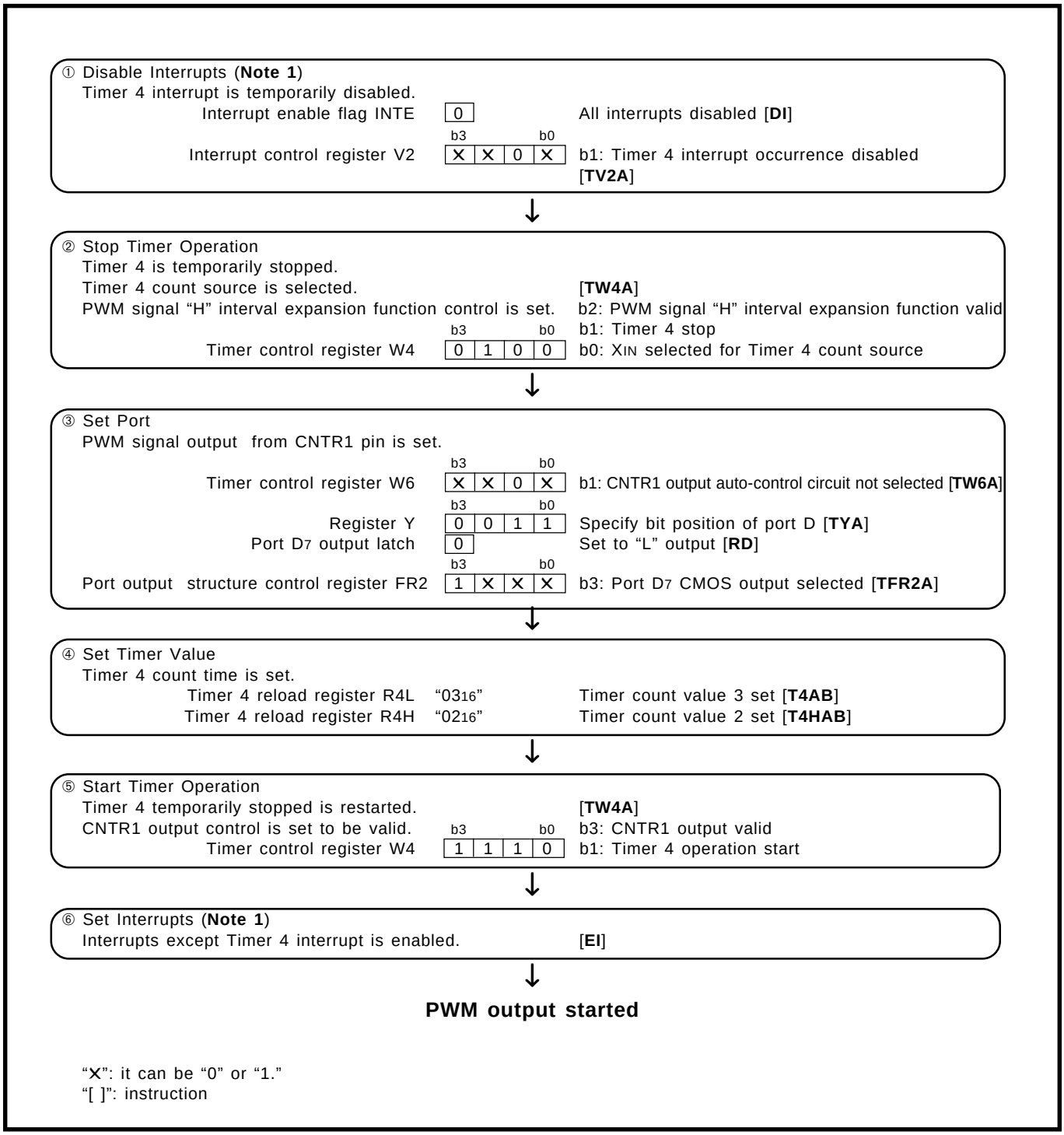


Fig. 2.3.7 Timer start by external input setting example



"X": it can be "0" or "1."
"[ ]": instruction

Fig. 2.3.8 PWM output control setting example

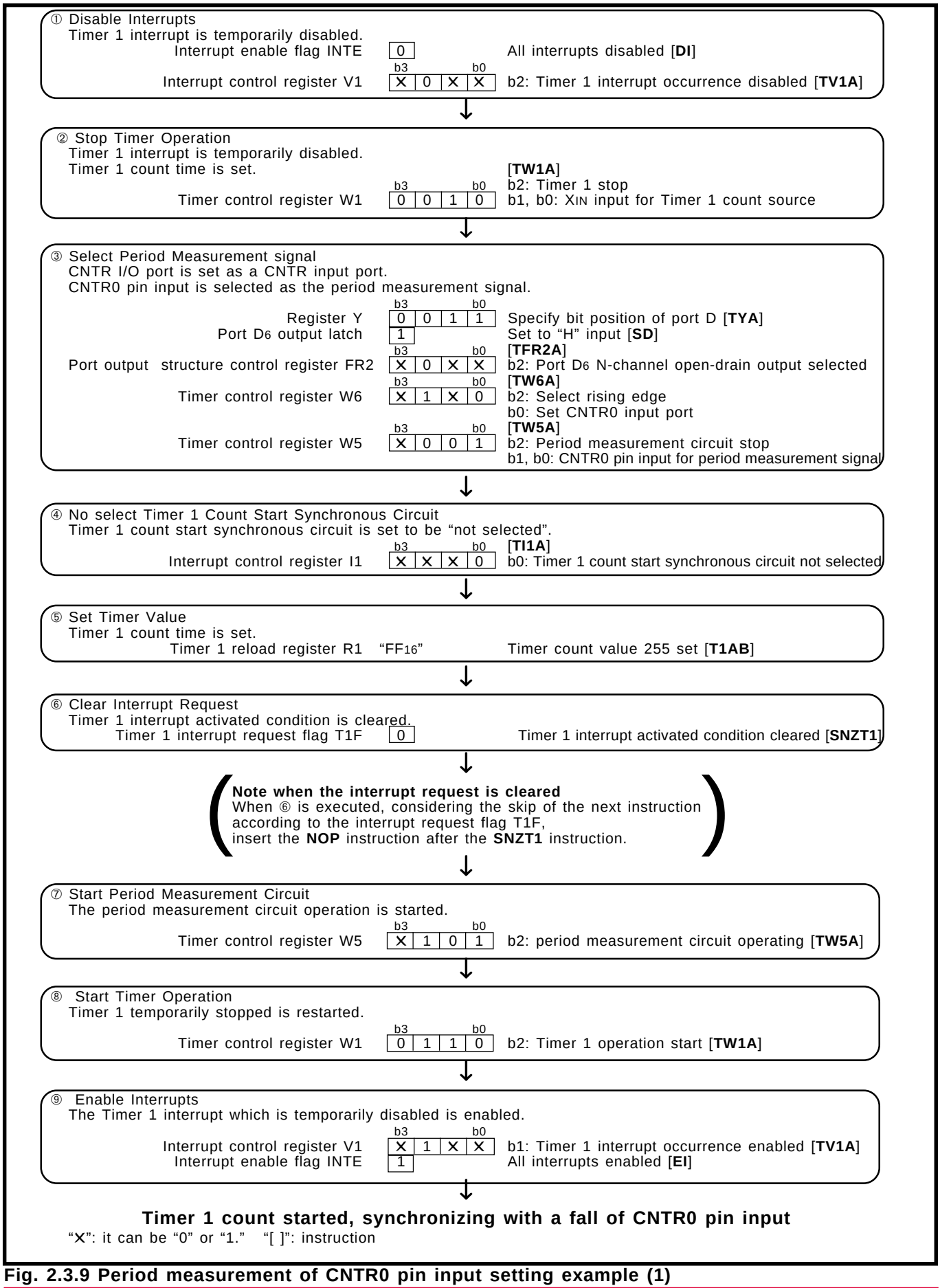


Fig. 2.3.9 Period measurement of CNTR0 pin input setting example (1)

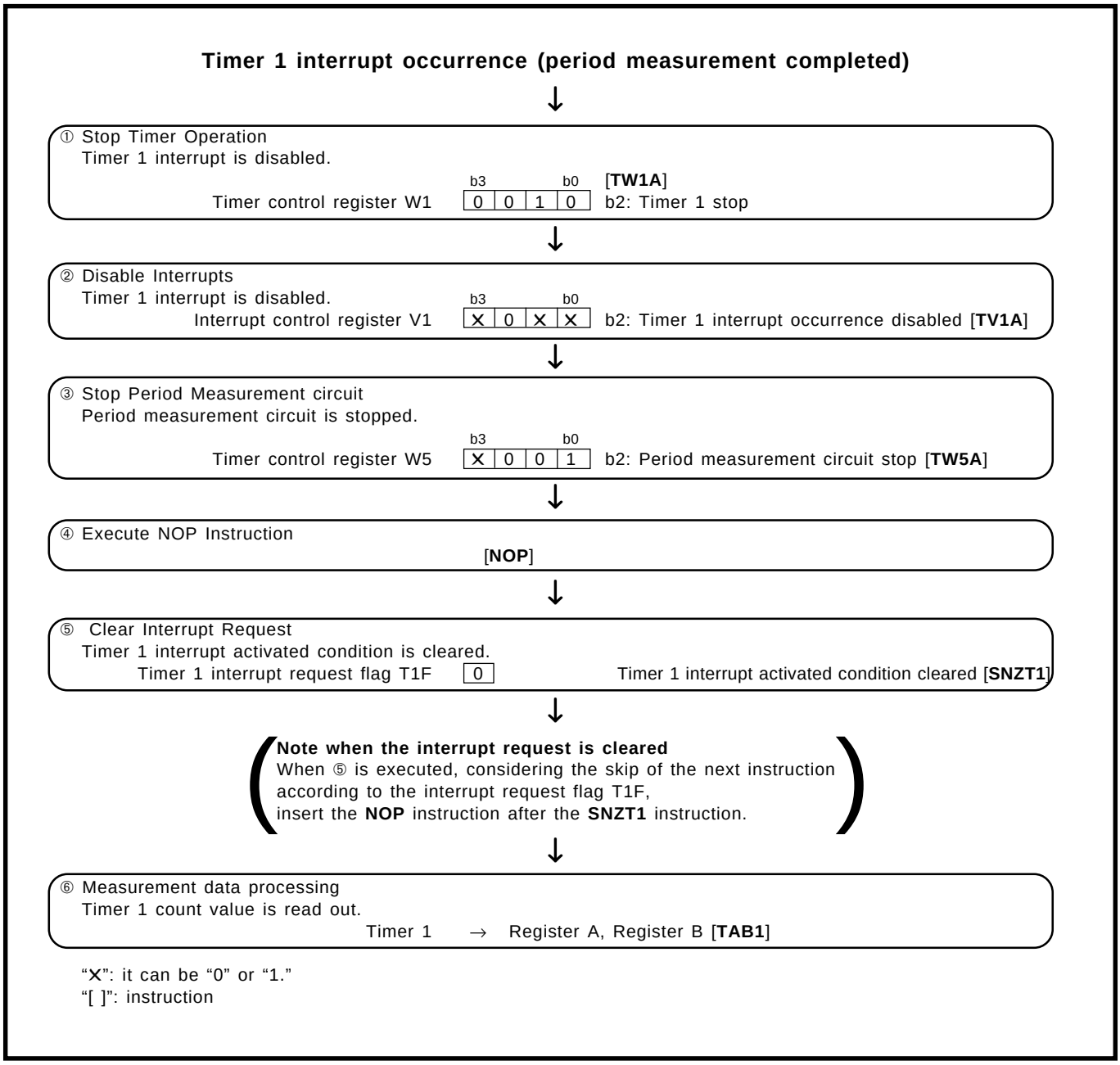


Fig. 2.3.10 Period measurement of CNTR0 pin input setting example (2)

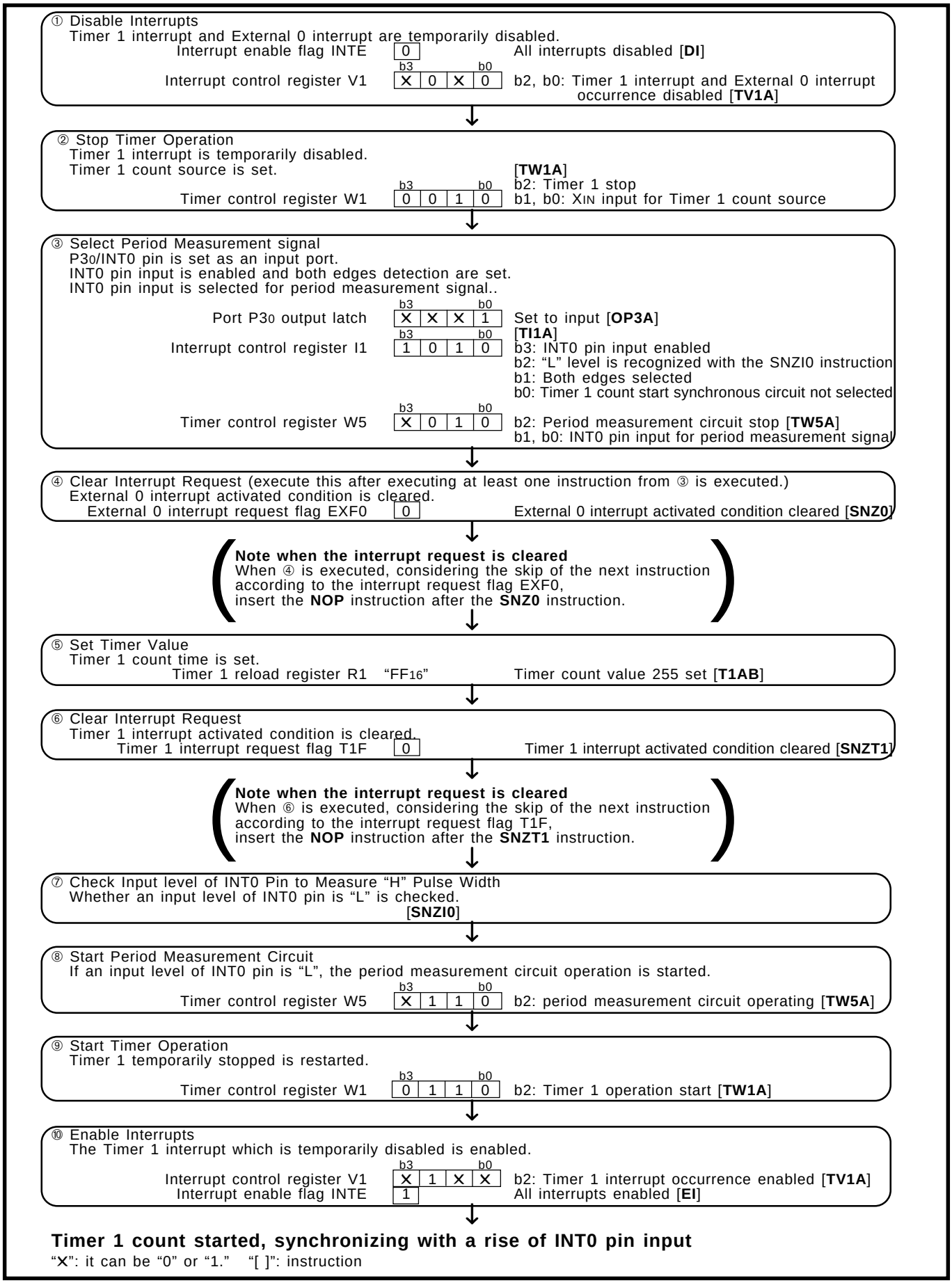


Fig. 2.3.11 Pulse width measurement of INT0 pin input setting example (1)

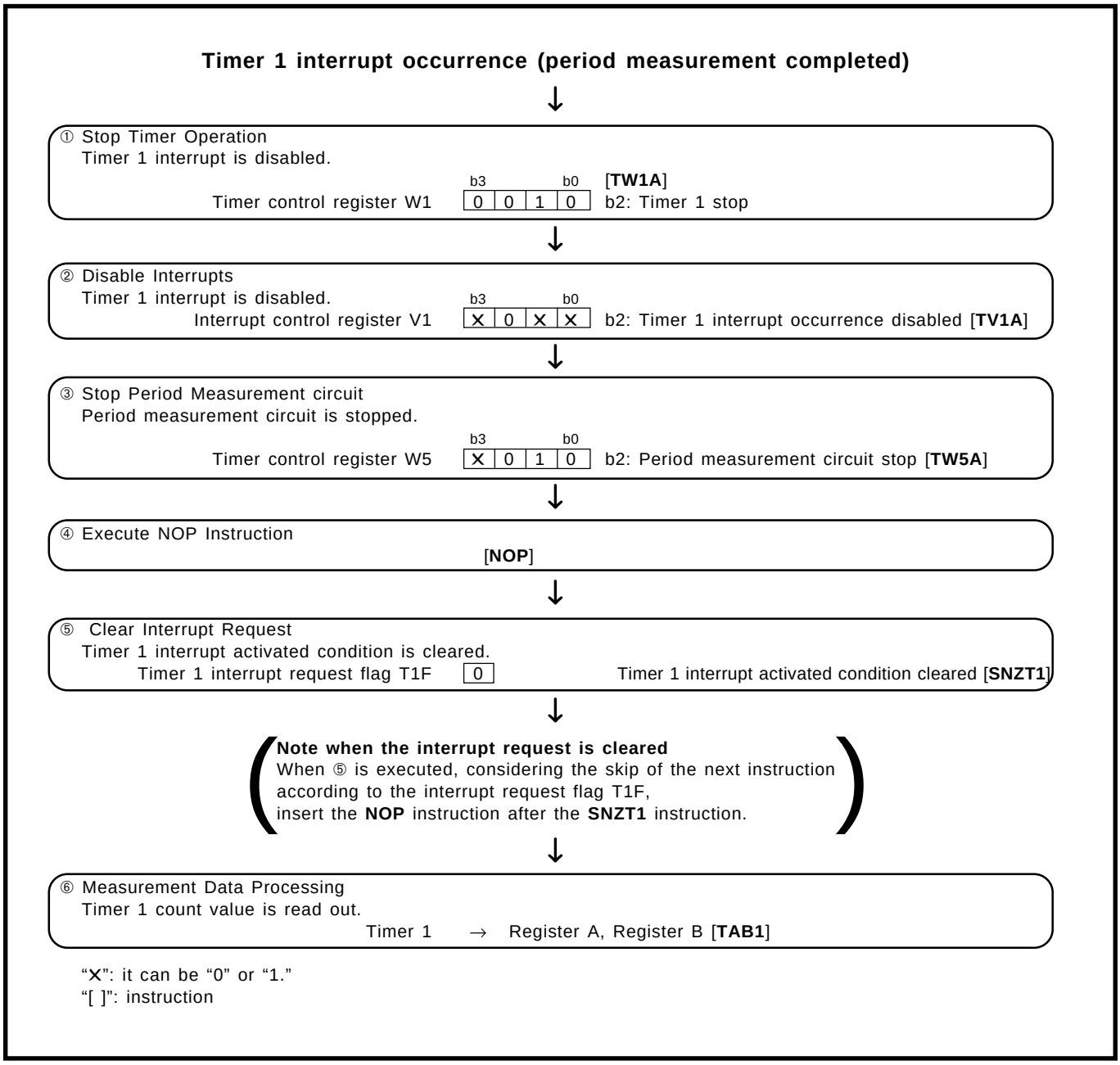


Fig. 2.3.12 Pulse width measurement of INT0 pin input setting example (2)

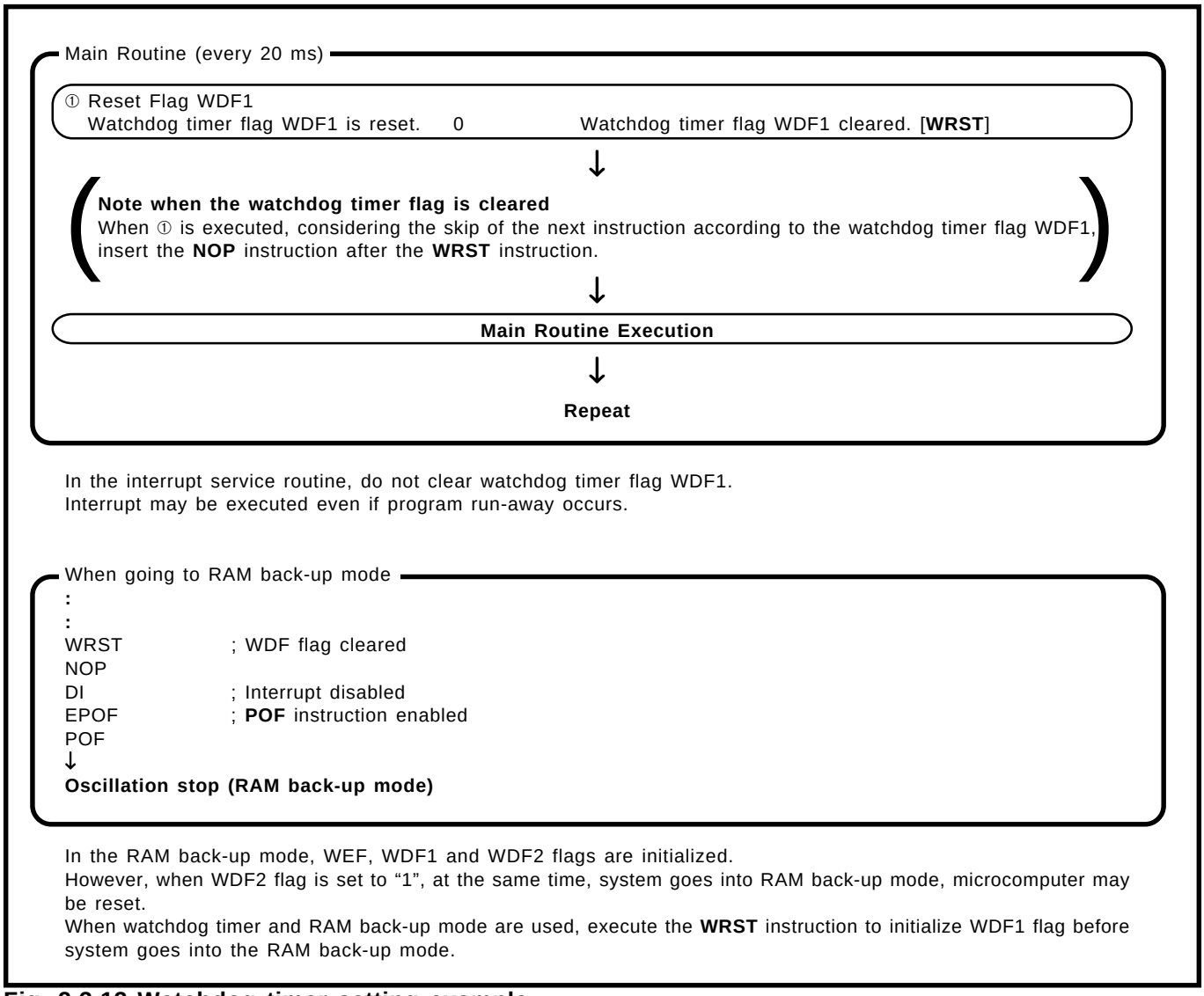


Fig. 2.3.13 Watchdog timer setting example

2.3.4 Notes on use

(1) Prescaler

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

(2) Count source

Stop timer 1, 2, 3, 4 or LC 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 the timer

Stop timer 1, 2, 3, 4 or LC counting and then execute the **T1AB**, **T2AB**, **T3AB**, **T4AB** or **TLCA** instruction to write its data.

(5) Writing to reload register R1, reload register R3 and reload register R4H

When writing data to reload register R1 while timer 1 is operating respectively, avoid a timing when timer 1 underflows.
When writing data to reload register R3 while timer 3 is operating respectively, avoid a timing when timer 3 underflows.
When writing data to reload register R4H while timer 4 is operating respectively, avoid a timing when timer 4 underflows.

(6) Timer 4

- At CNTR1 output vaild, if a timing of timer 4 underflow overlaps with a timing to stop timer 4, a hazard may be generated in a CNTR1 output waveform. Please review sufficiently.
- When "H" interval extension function of the PWM signal is set to be "valid", set "01_{16}" or more to reload register R4H.

(7) Watchdog timer

- The watchdog timer function is valid after system is released from reset. When not using the watchdog timer function, stop the watchdog timer function and execute the **DWDT** instruction, the **WRST** instruction continuously, and clear the WEF flag to "0".
- The watchdog timer function is valid after system is returned from the RAM back-up state. When not using the watchdog timer function, stop the watchdog timer function and execute the **DWDT** instruction and the **WRST** instruction continuously every system is returned from the RAM back-up state.
- When the watchdog timer function and RAM back-up function are used at the same time, initialize the flag WDF1 with the **WRST** instruction before system enters into the RAM back-up state.

(8) Pulse width input to CNTR0 pin, CNTR1 pin

Refer to section "3.1 Electrical characteristics" for rating value of pulse width input to CNTR0 pin, CNTR1 pin.

(9) Period measurement circuit

- When a period measurement circuit is used, clear bit 0 of register I1 to "0", and set a timer 1 count start synchronous circuit to be "not selected".

- While a period measurement circuit is operating, the timer 1 interrupt request flag (T1F) is not set by the timer 1 underflow signal, it is the flag for detecting the completion of period measurement.

- When a period measurement circuit is used, select the sufficiently higher-speed frequency than the signal for measurement for the count source of a timer 1.

● When the signal for period measurement is D6/CNTR0 pin input, do not select D6/CNTR0 pin input as timer 1 count source.
(The XIN input is recommended as timer 1 count source at the time of period measurement circuit use.)

●When the input of P30/INT0 pin is selected for measurement, set the bit 3 of a register I1 to "1", and set the input of INT0 pin to be enabled.

● Start timer operation immediately after operation of a period measurement circuit is started.

● Even when the edge for measurement is input by timer operation is started from the operation of period measurement circuit is started, timer 1 is not operated.

● When data is read from timer 1, stop the timer 1 and the period measurement circuit, and then execute the data read instruction. Depending on the state of timer 1, the timer 1 interrupt request flag (T1F) may be set to "1" when the period measurement circuit is stopped by clearing bit 2 of register W5 to "0". In order to avoid the occurrence of an unexpected interrupt, disable the timer 1 interrupt, and then, stop the period measurement circuit. Figure 2.3.14 shows the setting example to read measurement data of period measurement circuit.

(10) Prescaler, timer 1, timer 2 and timer 3 count start time and count time when operation starts

Count starts from the first rising edge of the count source ② in Fig.2.3.15 after prescaler, timer 1, timer 2 and timer 3 operations start ① in Fig.2.3.15.

Time to first underflow ③ in Fig.2.3.15 is shorter (for up to 1 period of the count source) than time among next underflow ④ in Fig.2.3.15 by the timing to start the timer and count source operations after count starts.

(11) Timer 4 count start time and count time when operation starts

Count starts from the rising edge ② in Fig.2.3.16 after the first falling edge of the count source, after timer 4 operation starts ① in Fig.2.3.16. Time to first underflow ③ in Fig.2.3.16 is different from time among next underflow ④ in Fig.2.3.16 by the timing to start the timer and count source operations after count starts.

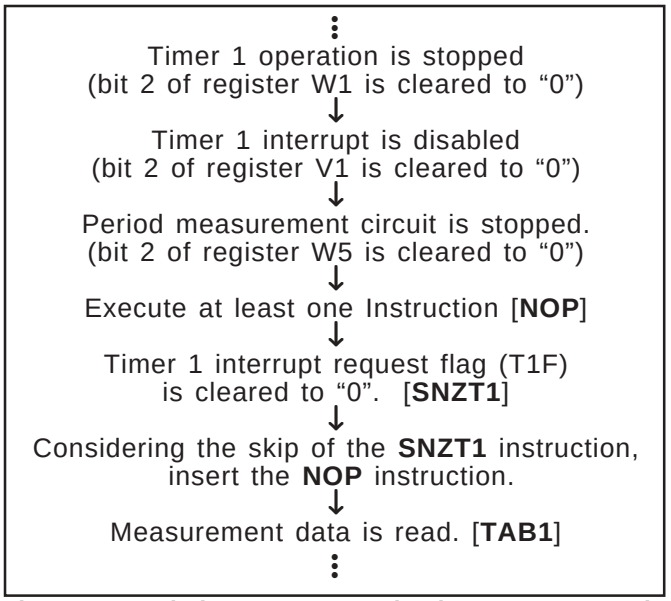


Fig. 2.3.14 Period measurement circuit program example

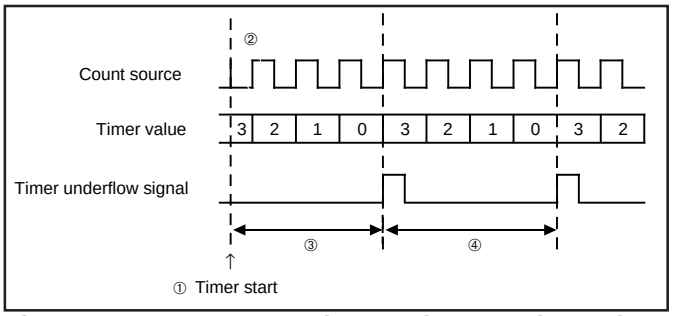


Fig. 2.3.15 Count start time and count time when operation starts (PS, T1, T2 and T3)

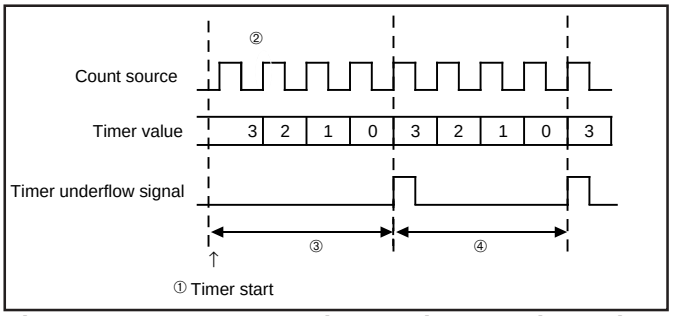
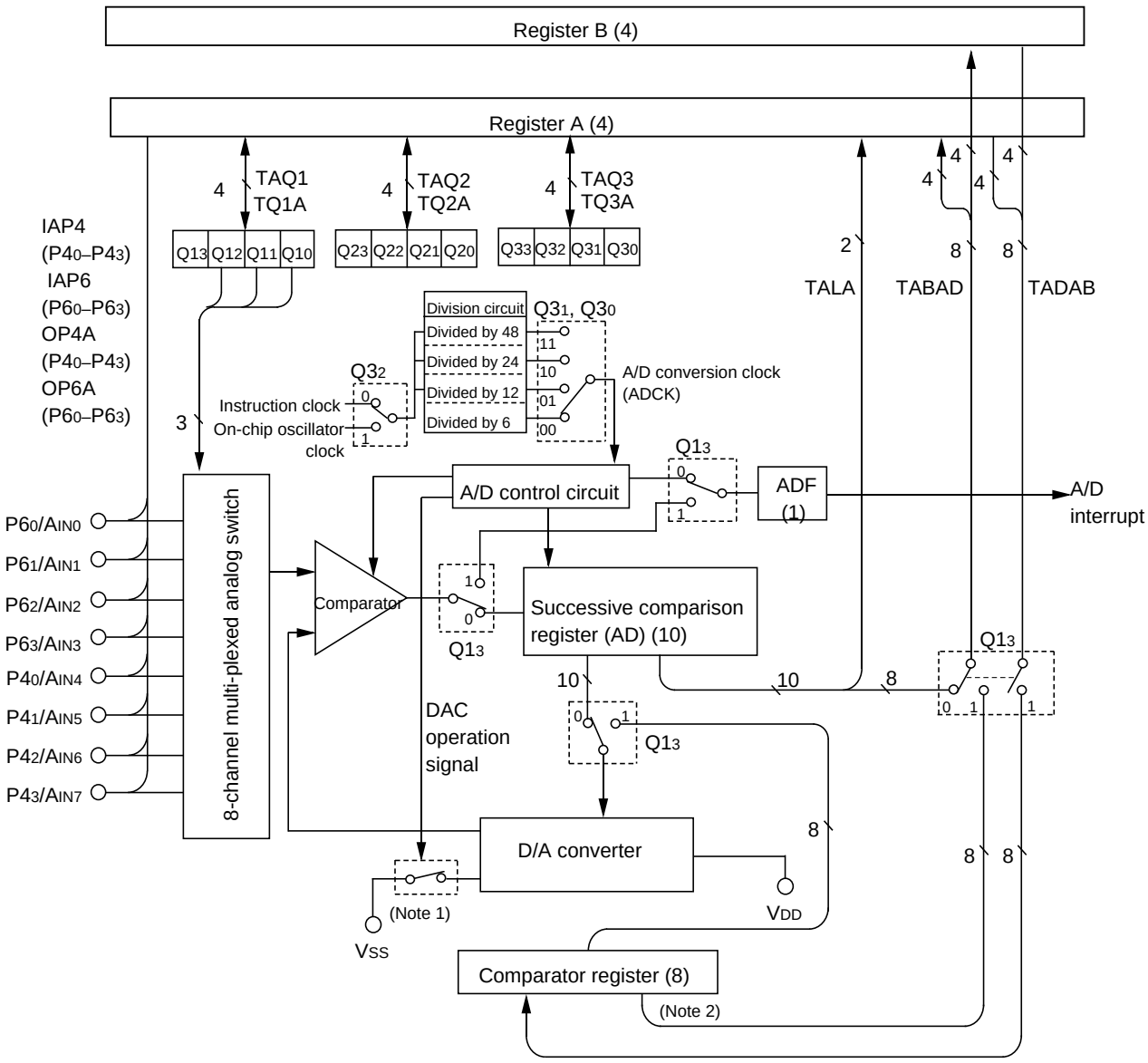


Fig. 2.3.16 Count start time and count time when operation starts (T4)

2.4 A/D converter

The 4519 Group has an 8-channel A/D converter with the 10-bit successive comparison method.
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 2.4.1 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 (Q13=1).
The value of the comparator register is retained even when the mode is switched to the A/D conversion mode (Q13=0) because it is separated from the successive comparison register (AD). Also, the resolution in the comparator mode is 8 bits because the comparator register consists of 8 bits.

Fig. 2.4.1 A/D converter structure

2.4.1 Related registers

(1) Interrupt control register V2

Table 2.4.1 shows the interrupt control 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 2.4.1 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 (**Note 2**)	0	Interrupt disabled (**SNZSI** instruction is valid)	
		1	Interrupt enabled (**SNZSI** instruction is invalid) (**Note 2**)	
$V2_2$	A/D interrupt enable bit	0	Interrupt disabled (**SNZAD** instruction is valid)	
		1	Interrupt enabled (**SNZAD** instruction is invalid) (**Note 2**)	
$V2_1$	Timer 4 interrupt enable bit	0	Interrupt disabled (**SNZT4** instruction is valid)	
		1	Interrupt enabled (**SNZT4** instruction is invalid) (**Note 2**)	
$V2_0$	Timer 3 interrupt enable bit	0	Interrupt disabled (**SNZT3** instruction is valid)	
		1	Interrupt enabled (**SNZT3** instruction is invalid) (**Note 2**)	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: These instructions are equivalent to the **NOP** instruction.
3: When setting the A/D converter, $V2_3$, $V2_1$ and $V2_0$ are not used.

(2) A/D control register Q1

Table 2.4.2 shows the A/D control register Q1.
Set the contents of this register through register A with the **TQ1A** instruction.
In addition, the **TAQ1** instruction can be used to transfer the contents of register Q1 to register A.

Table 2.4.2 A/D control register Q1

A/D control register Q1		at reset : 0000_2			at RAM back-up : state retained	R/W
$Q1_3$	A/D operation mode control bit	0			A/D conversion mode	
		1			Comparator mode	
$Q1_2$	Analog input pin selection bits	$Q1_2$	$Q1_1$	$Q1_0$	Analog input 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}	
$Q1_0$		1	0	1	A_{IN5}	
		1	1	0	A_{IN6}	
		1	1	1	A_{IN7}	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: In order to select A_{IN7}–A_{IN0}, set register Q1 after setting regsiter Q2.

(3) A/D control register Q2

Table 2.4.3 shows the A/D control register Q2.
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 2.4.3 A/D control register Q2

A/D control register Q2		at reset : 0000_2		at RAM back-up : state retained	R/W
$Q2_3$	$P2_3/AIN_3$ pin function selection bit	0	$P4_0$, $P4_1$, $P4_2$, $P4_3$		
		1	AIN_4, AIN_5, AIN_6, AIN_7		
$Q2_2$	$P6_2/AIN_2$, $P6_3/AIN_3$ pin function selection bit	0	$P6_2$, $P6_3$		
		1	AIN_2, AIN_3		
$Q2_1$	$P6_1/AIN_1$ pin function selection bit	0	$P6_1$		
		1	AIN_1		
$Q2_0$	$P6_0/AIN_0$ pin function selection bit	0	$P6_0$		
		1	AIN_0		

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

(4) A/D control register Q3

Table 2.4.4 shows the A/D control register Q3.
Set the contents of this register through register A with the **TQ3A** instruction.
The contents of register Q3 is transferred to register A with the **TAQ3** instruction.

Table 2.4.4 A/D control register Q3

A/D control register Q3		at reset : 0000_2		at RAM back-up : state retained	R/W
$Q3_3$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$Q3_2$	A/D converter operation clock selection bit	0	Instruction clock (INSTCK)		
		1	On-chip oscillator (f(RING))		
$Q3_1$	A/D converter operation clock division ratio selection bits	$Q3_1$	$Q3_0$	Division ratio	
$Q3_0$		0	0	Frequency divided by 6	
		0	1	Frequency divided by 12	
		1	0	Frequency divided by 24	
		1	1	Frequency divided by 48	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: In order to select AIN_7–AIN_4, set register Q1 after setting regsiter Q3.

2.4.2 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 AIN_0 pin for this analog input.

Figure 2.4.2 shows the A/D conversion mode setting example.

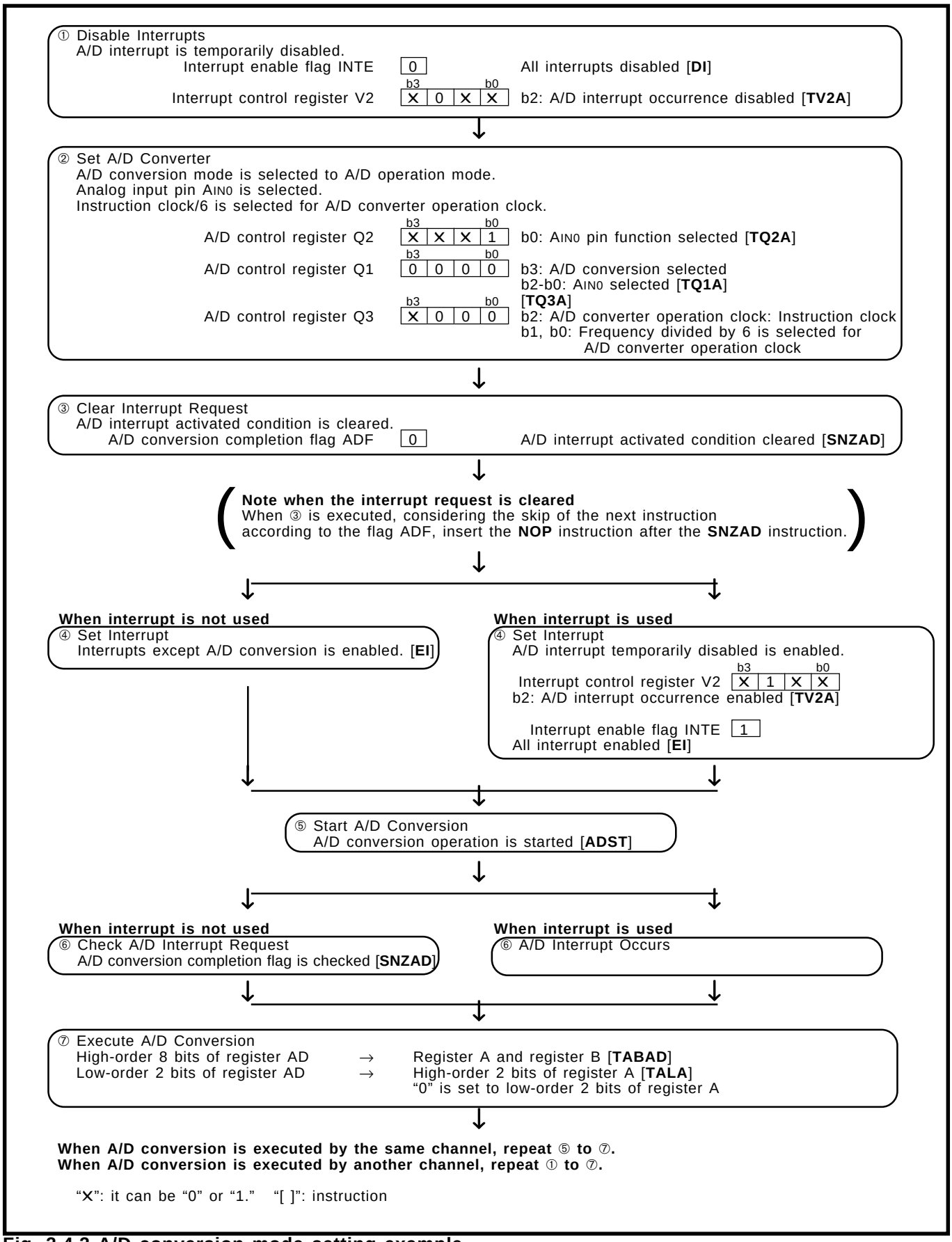


Fig. 2.4.2 A/D conversion mode setting example

2.4.3 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 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 2.4.3 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 2.4.4. In addition, test the application products sufficiently.

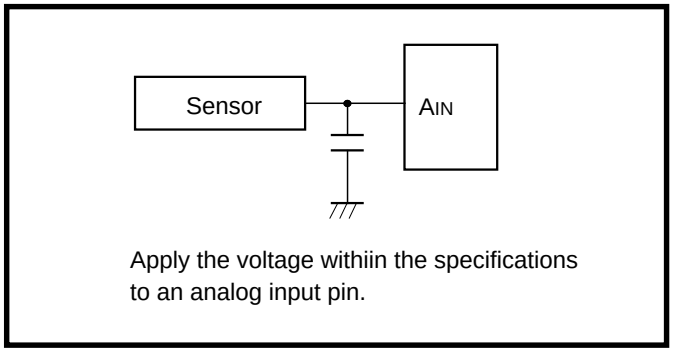


Fig. 2.4.3 Analog input external circuit example-1

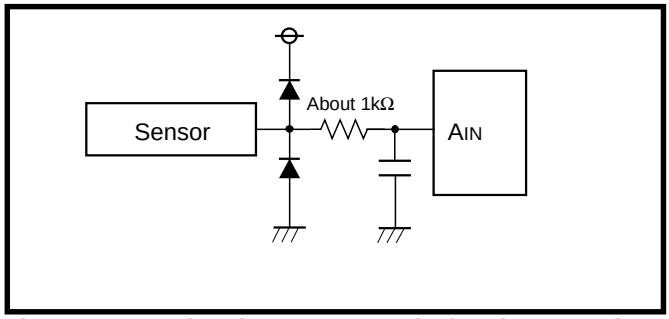


Fig. 2.4.4 Analog input external circuit example-2

(3) Notes for the use of A/D conversion 2

Do not change the operating mode of the A/D converter by bit 3 of register Q1 during A/D conversion (A/D conversion mode and comparator mode).

(4) Notes for the use of A/D conversion 3

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 Q1 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 (refer to Figure 2.4.5①).
- 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 bit 3 of register Q1, and execute the **SNZAD** instruction to clear the ADF flag to “0”.

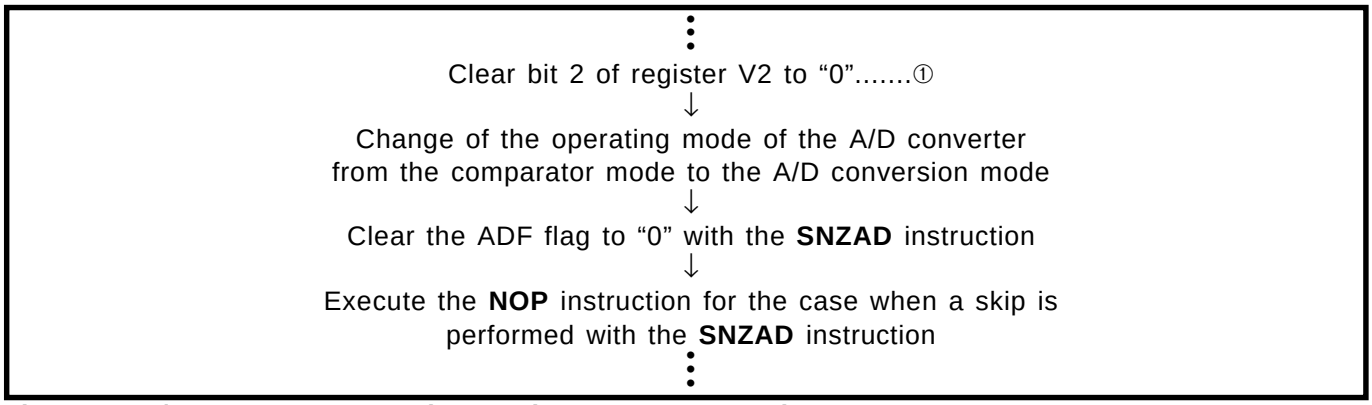


Fig. 2.4.5 A/D converter operating mode program example

(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 2 machine cycles + A/D conversion clock (ADCK) 1 clock.

(6) Analog input pins

When $P4_0/AIN_4$–$P4_3/AIN_7$, $P6_0/AIN_0$–$P6_3/AIN_3$ are set to pins for analog input, they cannot be used as I/O ports P4 and P6.

(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

As for the supply voltage when A/D converter is used and the recommended operating condition of the A/D convesion clock frequency, refer to the "3.1 Electrical characteristics".

2.5 Serial I/O

The 4519 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.

2.5.1 Serial I/O functions

Serial I/O consists of the serial I/O register SI, serial I/O control 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: 4519 Group only supports LSB-first transmit and receive.

■ **Shift clock**

When using the internal clock of 4519 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 2, 4 or 8.

When using an external clock, the clock frequency input to the SCK pin determines the data transfer rate.

Figure 2.5.1 shows the serial I/O block diagram.

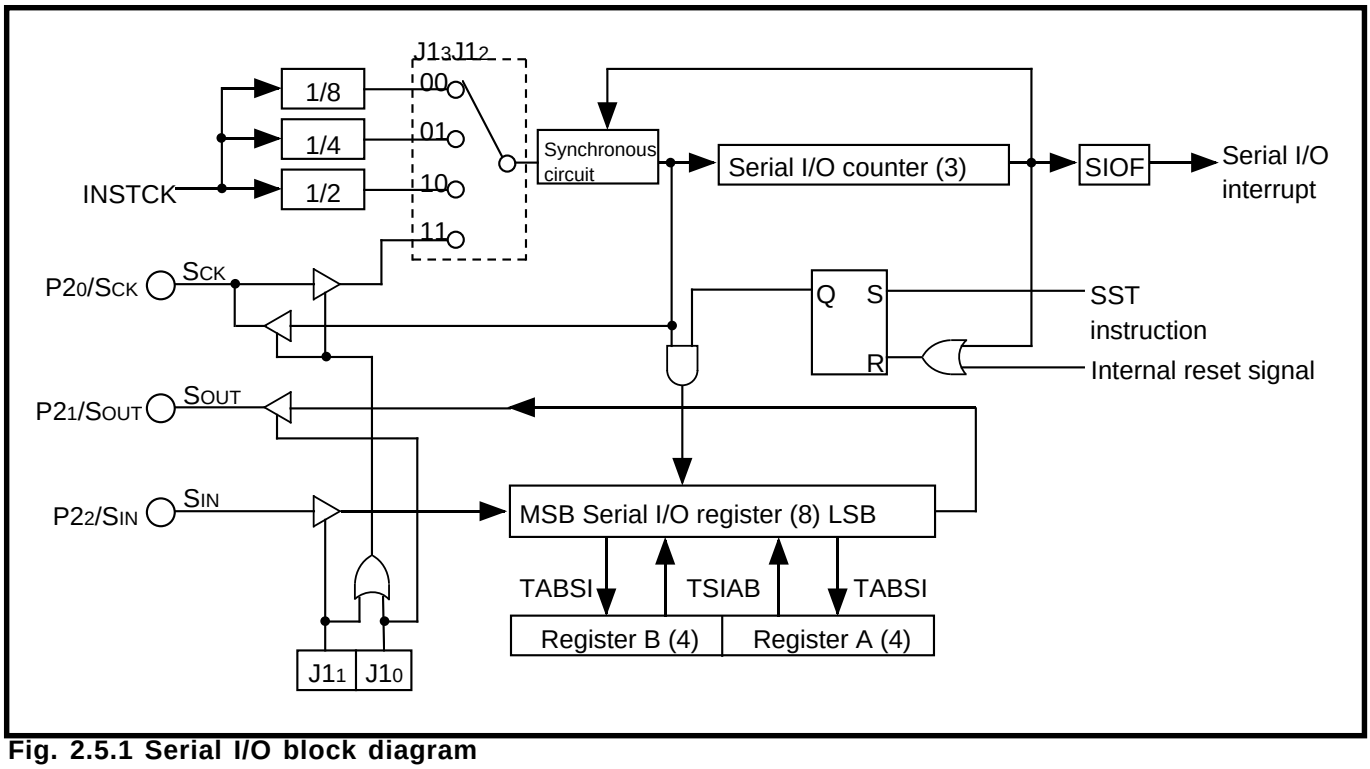


Fig. 2.5.1 Serial I/O block diagram

2.5.2 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.

Also, the low-order 4 bits of register SI is transferred to register A, and the high-order 4 bits of register SI is transferred to register B with the **TABSI** instruction.

(2) Serial I/O transmit/receive completion flag (SIOF)

Serial I/O transmit/receive completion flag (SIOF) is set to "1" when serial data transmit or receive operation completes. The state of SIOF flag can be examined with the skip instruction (**SNZSI**).

(3) Interrupt control register V2

Table 2.5.1 shows the interrupt control 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 2.5.1 Interrupt control register V2

Interrupt control register V2		at reset : 0000_2	at RAM back-up : 0000_2	R/W
$V2_3$	Timer 4, serial I/O interrupt enable bit	0	Interrupt disabled (**SNZSI** instruction is valid)	
		1	Interrupt enabled (**SNZSI** instruction is invalid) (**Note 2**)	
$V2_2$	A/D interrupt enable bit	0	Interrupt disabled (**SNZAD** instruction is valid)	
		1	Interrupt enabled (**SNZAD** instruction is invalid) (**Note 2**)	
$V2_1$	Timer 4 interrupt enable bit	0	Interrupt disabled (**SNZT4** instruction is valid)	
		1	Interrupt enabled (**SNZT4** instruction is invalid) (**Note 2**)	
$V2_0$	Timer 3 interrupt enable bit	0	Interrupt disabled (**SNZT3** instruction is valid)	
		1	Interrupt enabled (**SNZT3** instruction is invalid) (**Note 2**)	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: These instructions are equivalent to the **NOP** instruction.
3: When setting the serial I/O, $V2_2$, $V2_1$ and $V2_0$ are not used.

(4) Serial I/O mode register J1

Table 2.5.2 shows the serial I/O mode register J1.

Set the contents of this register through register A with the **TJ1A** instruction.

In addition, the **TAJ1** instruction can be used to transfer the contents of register J1 to register A.

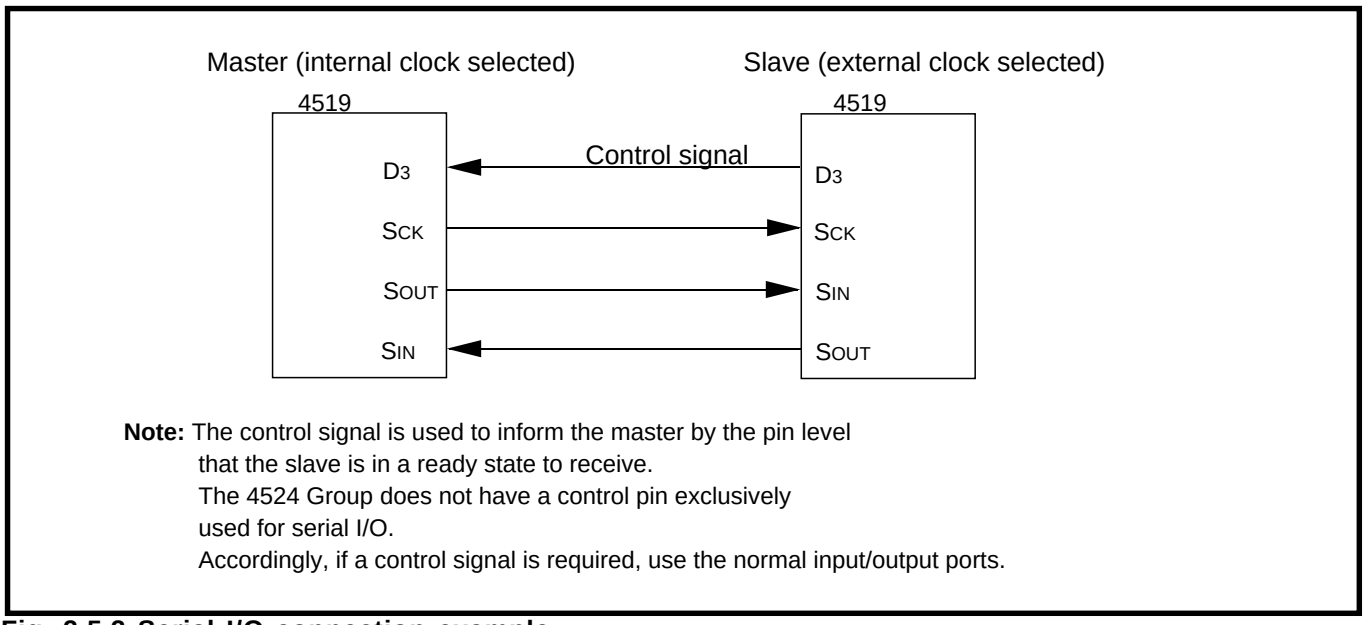
Table 2.5.2 Serial I/O mode register J1

Serial I/O control register J1		at reset : 0000_2		at RAM back-up : state retained	R/W
$J1_3$	Serial I/O synchronous clock selection bits	$J1_3$	$J1_2$	Synchronous clock	
		0	0	Instruction clock (INSTCK) divided by 8	
$J1_2$		0	1	Instruction clock (INSTCK) divided by 4	
		1	0	Instruction clock (INSTCK) divided by 2	
		1	1	External clock (S_{CK} input)	
$J1_1$	Serial I/O port function selection bits	$J1_1$	$J1_0$	Port function	
		0	0	$P2_0$, $P2_1$, $P2_2$ selected/S_{CK}, S_{OUT}, S_{IN} not selected	
$J1_0$		0	1	S_{CK}, S_{OUT}, $P2_2$ selected/$P2_0$, $P2_1$, S_{IN} not selected	
		1	0	S_{CK}, $P2_1$, S_{IN} selected/$P2_0$, S_{OUT}, $P2_2$ not selected	
		1	1	S_{CK}, S_{OUT}, S_{IN} selected/$P2_0$, $P2_1$, $P2_2$ not selected	

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

2.5.3 Operation description

Figure 2.5.2 shows the serial I/O connection example, Figure 2.5.3 shows the serial I/O register state, and Figure 2.5.4 shows the serial I/O transfer timing.



Note: The control signal is used to inform the master by the pin level that the slave is in a ready state to receive.
The 4524 Group does not have a control pin exclusively used for serial I/O.
Accordingly, if a control signal is required, use the normal input/output ports.

Fig. 2.5.2 Serial I/O connection example

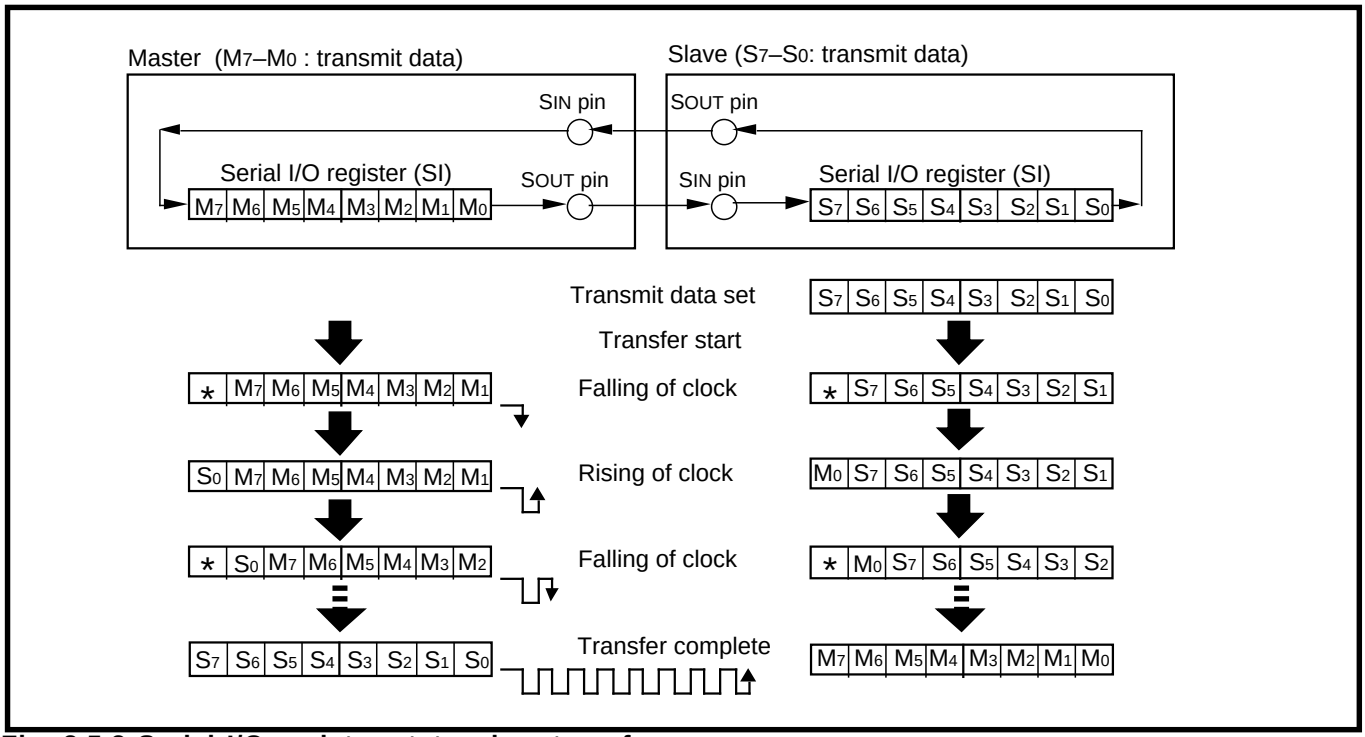
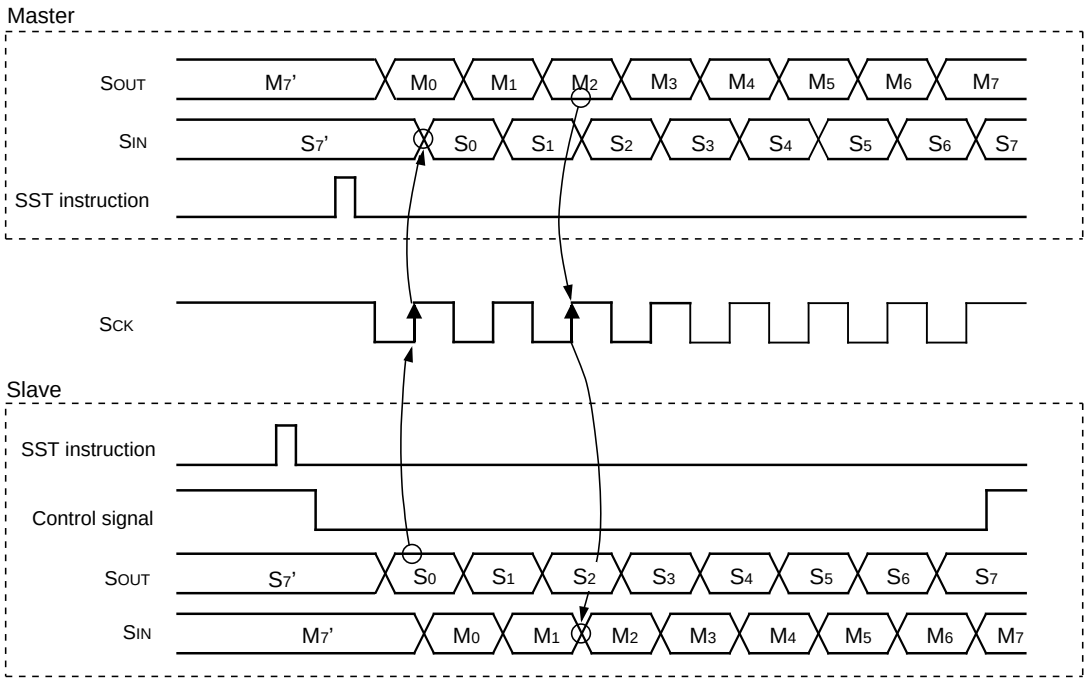


Fig. 2.5.3 Serial I/O register state when transfer



M0–M7: Contents of master serial I/O register
S0–S7: Contents of slave serial I/O register
Rising of SCK: Serial input
Falling of SCK: Serial output
M7', S7': Contents of previous master, slave MSB

Fig. 2.5.4 Serial I/O transfer timing

The full duplex communication of master and slave is described using the connection example shown in Figure 2.5.2.

(1) Transmit/receive operation of master

① Set the transmit data to 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 Check whether the microcomputer on the slave side is ready to transmit/receive or not.
In the connection example in Figure 2.5.2, check that the input level of control signal is “L” level.

③ Start serial transmit/receive 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 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 of register SI.

⑧ A serial I/O interrupt request occurs when the transmit/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/receive multiple data in succession.
2: For the program on the master side, start to transmit the next data at the next timing (control signal turns “L”). Do not start to transmit the next data during the previous data transfer (control signal = “L”).

(2) Transmit/receive operation of slave

① Set the transmit data 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 Start serial transmit/receive with the **SST** instruction. However, in Figure 2.5.2 where an external clock is selected, transmit/receive 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 master side is informed that the receiving side is ready to receive. In the connection example in Figure 2.5.2, the control signal "L" level is output.

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 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 of register SI.

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

⑨ Read the receive data within the serial I/O interrupt service routine; or read the data 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.

⑩ Set the control signal pin level to "H" after the receive operation is completed.

Note: Repeat steps ① through ⑩ to transmit/receive multiple data in succession.

2.5.4 Serial I/O application example

(1) Serial I/O

Outline: The 4519 Group can communicate with peripheral ICs.
Specifications: Figure 2.5.2 Serial I/O connection example.

Figure 2.5.5 shows the setting example when a serial I/O interrupt of master side is not used, and Figure 2.5.6 shows the slave serial I/O setting example.

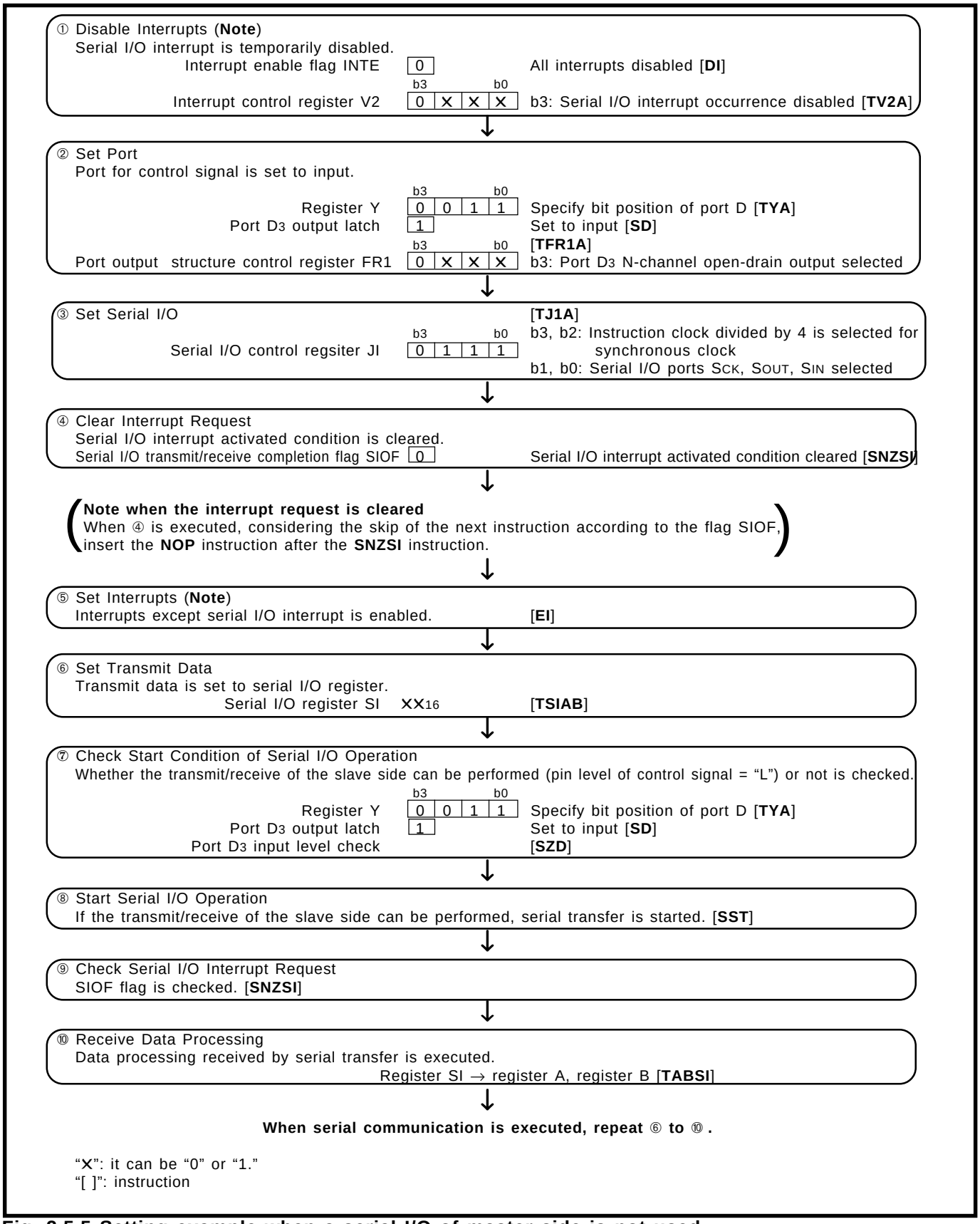


Fig. 2.5.5 Setting example when a serial I/O of master side is not used

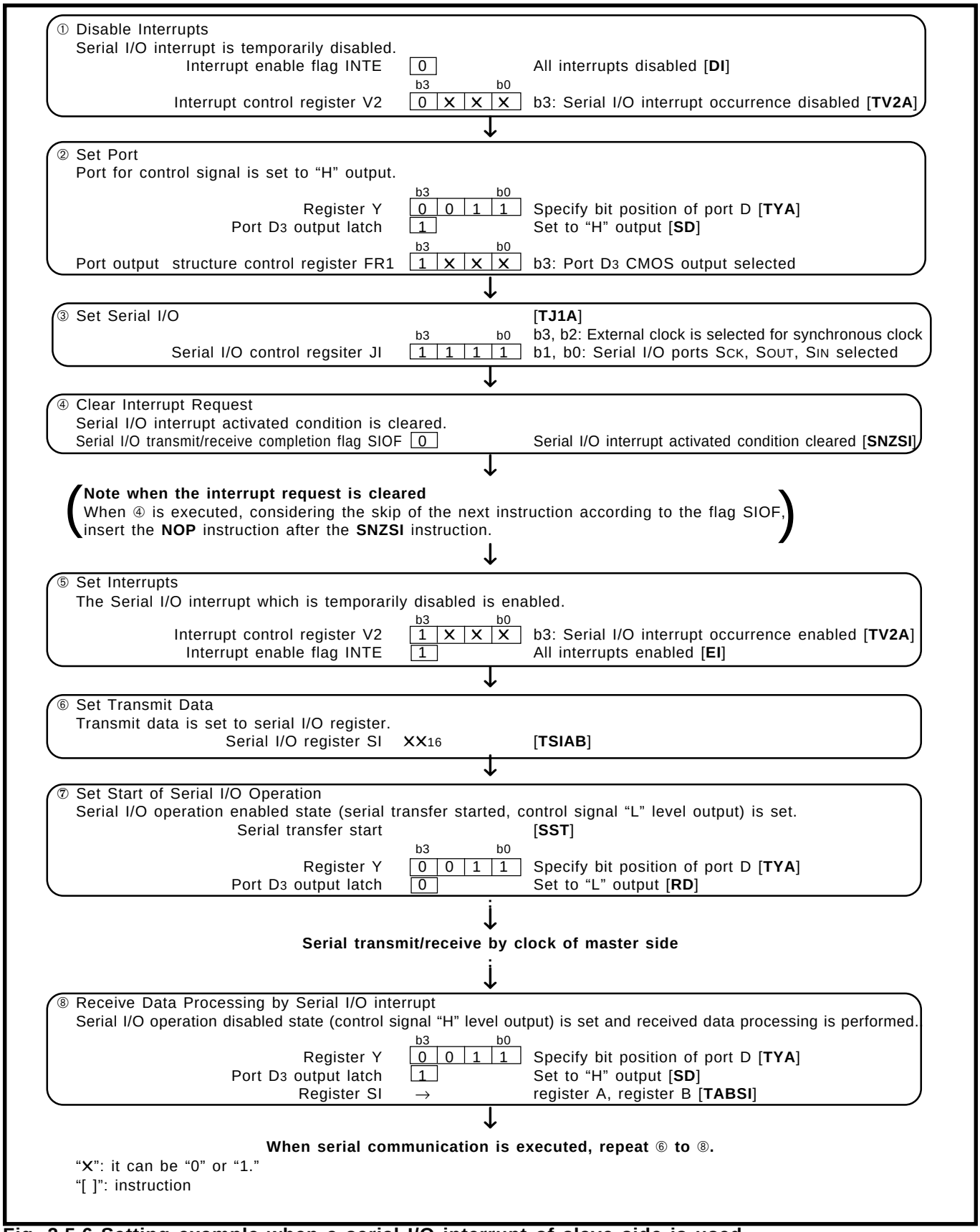


Fig. 2.5.6 Setting example when a serial I/O interrupt of slave side is used

2.5.5 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 transmit/receive is continued as long as an external clock is input. If an external clock is input 9 times or more and serial transmit/receive 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 to “1” when a clock is counted 8 times.
- Be sure to set the initial input level on the external clock pin to “H” level.
- Refer to section “3.1 Electrical characteristics” when using serial I/O with an external clock.

2.6 Reset

System reset is performed by applying “L” level to the $\overline{\text{RESET}}$ 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.

Then when “H” level is applied to $\overline{\text{RESET}}$ pin, the program starts from address 0 in page 0 after elapsing of the internal oscillation stabilizing time (On-chip oscillator (internal oscillator) clock is counted for 120 to 144 times). Figure 2.6.2 shows the structure of reset pin and its peripherals, and power-on reset operation.

2.6.1 Reset circuit

The 4519 Group has the voltage drop detection circuit.

(1) Power-on reset

Reset can be automatically performed at power on (power-on reset) by the built-in power-on reset circuit. When the built-in power-on reset circuit is used, the time for the supply voltage to rise from 0 V to the minimum rating value of the recommended operating conditions must be set to 100 µs or less. If the rising time exceeds 100 µs, connect a capacitor between the $\overline{\text{RESET}}$ pin and V_{SS} at the shortest distance, and input “L” level to $\overline{\text{RESET}}$ pin until the value of supply voltage reaches the minimum rating value of the recommended operating conditions.

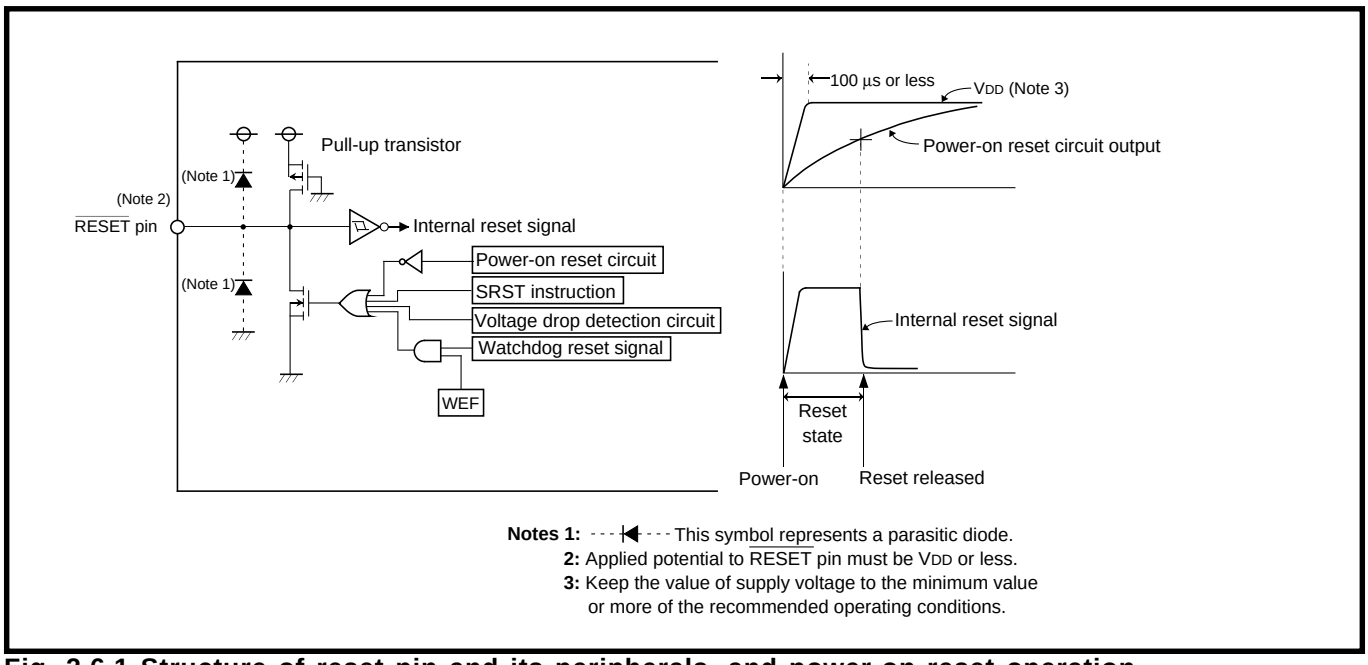


Fig. 2.6.1 Structure of reset pin and its peripherals, and power-on reset operation

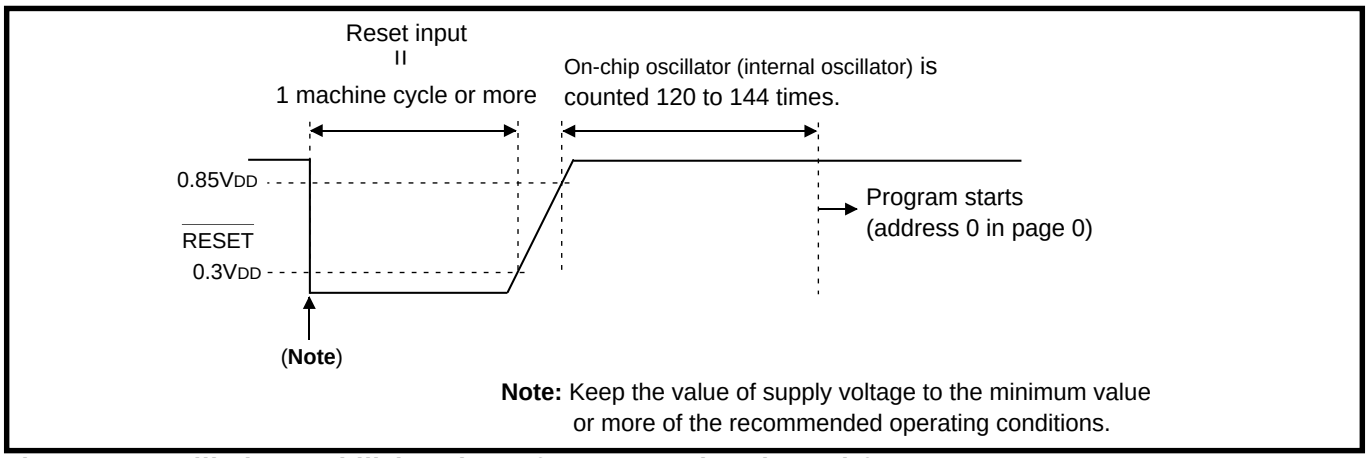


Fig. 2.6.2 Oscillation stabilizing time after system is released from reset

2.6.2 Internal state at reset

Figure 2.6.3 and Figure 2.6.4 show the internal state at reset. The contents of timers, registers, flags and RAM other than shown in Figure 2.6.3 and Figure 2.6.4 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 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
- Interrupt control register I3 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) 1
- Timer control register PA ..... 0 (Prescaler stopped)
- Timer control register W1 ..... 0 0 0 0 (Timer 1 stopped)
- Timer control register W2 ..... 0 0 0 0 (Timer 2 stopped)
- Timer control register W3 ..... 0 0 0 0 (Timer 3 stopped)
- Timer control register W4 ..... 0 0 0 0 (Timer 4 stopped)
- Timer control register W5 ..... 0 0 0 0
- Timer control register W6 ..... 0 0 0 0 (Period measurement circuit)
- Clock control register MR 1 1 1 1
- Serial I/O transmit/receive completion flag (SIOF) 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
- A/D 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

"X" represents undefined.

Fig. 2.6.3 Internal state at reset

- Key-on wakeup control register K0 ..... 0 0 0 0
- Key-on wakeup control register K1 ..... 0 0 0 0
- Key-on wakeup control register K2 ..... 0 0 0 0
- Pull-up control register PU0 ..... 0 0 0 0
- Pull-up control register PU1 ..... 0 0 0 0
- Port output structure control register FR0 ..... 0 0 0 0
- Port output structure control register FR1 ..... 0 0 0 0
- Port output structure control register FR2 ..... 0 0 0 0
- Port output structure control register FR3 ..... 0 0 0 0
- 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
- Operation source clock ..... On-chip oscillator (operating)
- Ceramic resonator circuit ..... Operating
- Quartz-crystal oscillation circuit ..... Stop
- RC oscillation circuit ..... Stop

"X" represents undefined.

Fig. 2.6.4 Internal state at reset

2.6.3 Notes on use

(1) Register initial value

The initial value of the following registers are undefined after system is released from reset. After system is released from reset, set initial values.

- Register Z (2 bits)
- Register D (3 bits)
- Register E (8 bits)

(2) Power-on reset

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

Refer to section "**3.1 Electrical characteristics**" for the reset voltage of the recommended operating conditions.

2.7 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 2.7.1 shows the voltage drop detection circuit, and Figure 2.7.2 shows the operation waveform example of the voltage drop detection circuit. Table 2.7.1 shows the voltage drop detection circuit operation state.

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

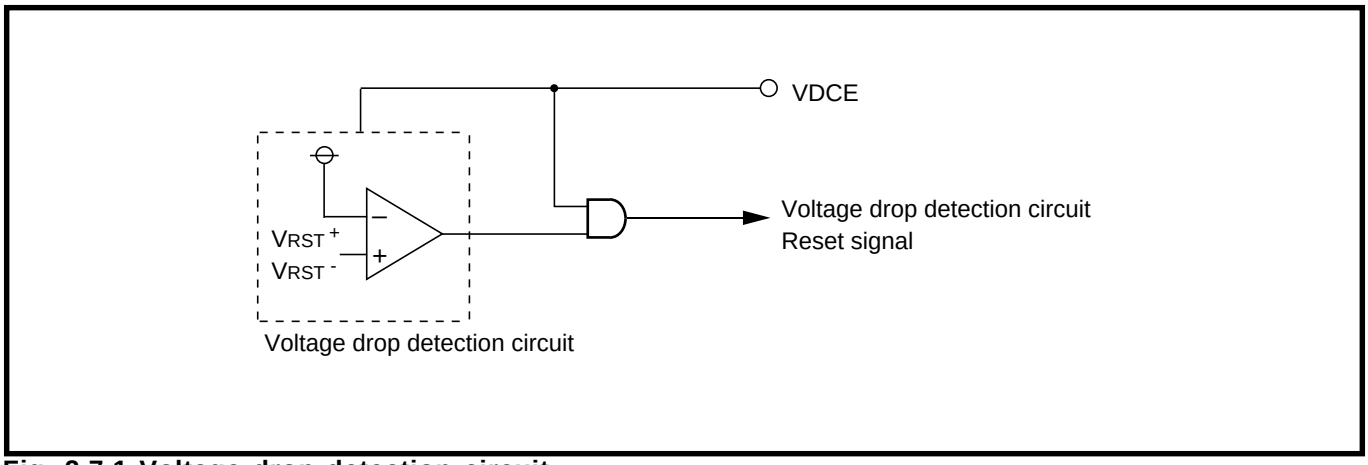


Fig. 2.7.1 Voltage drop detection circuit

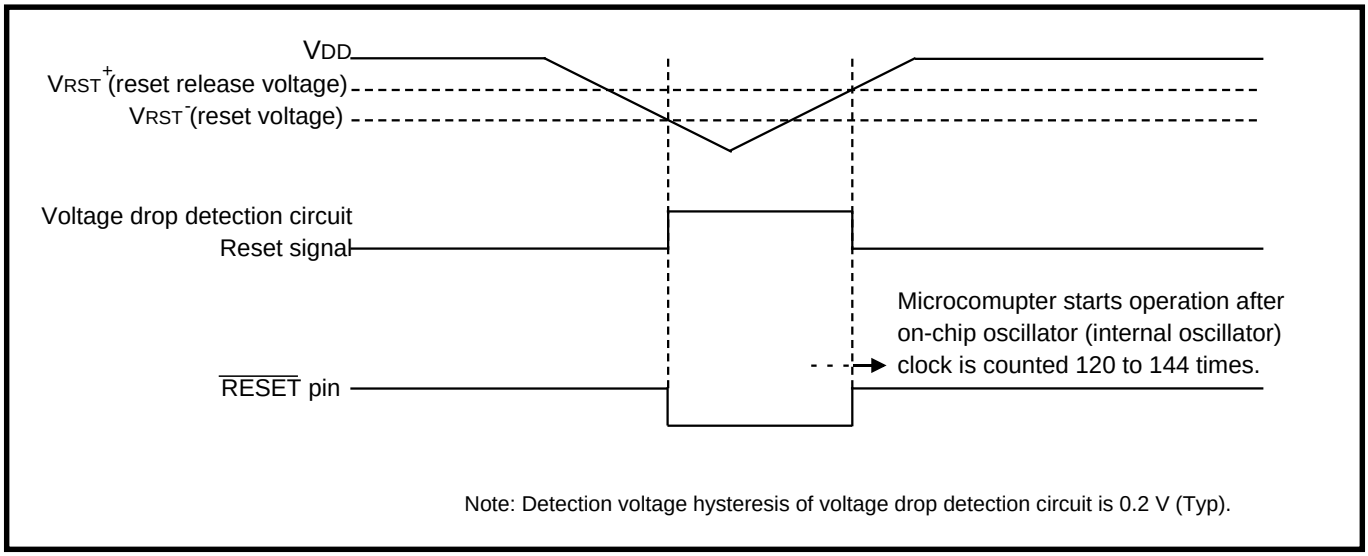


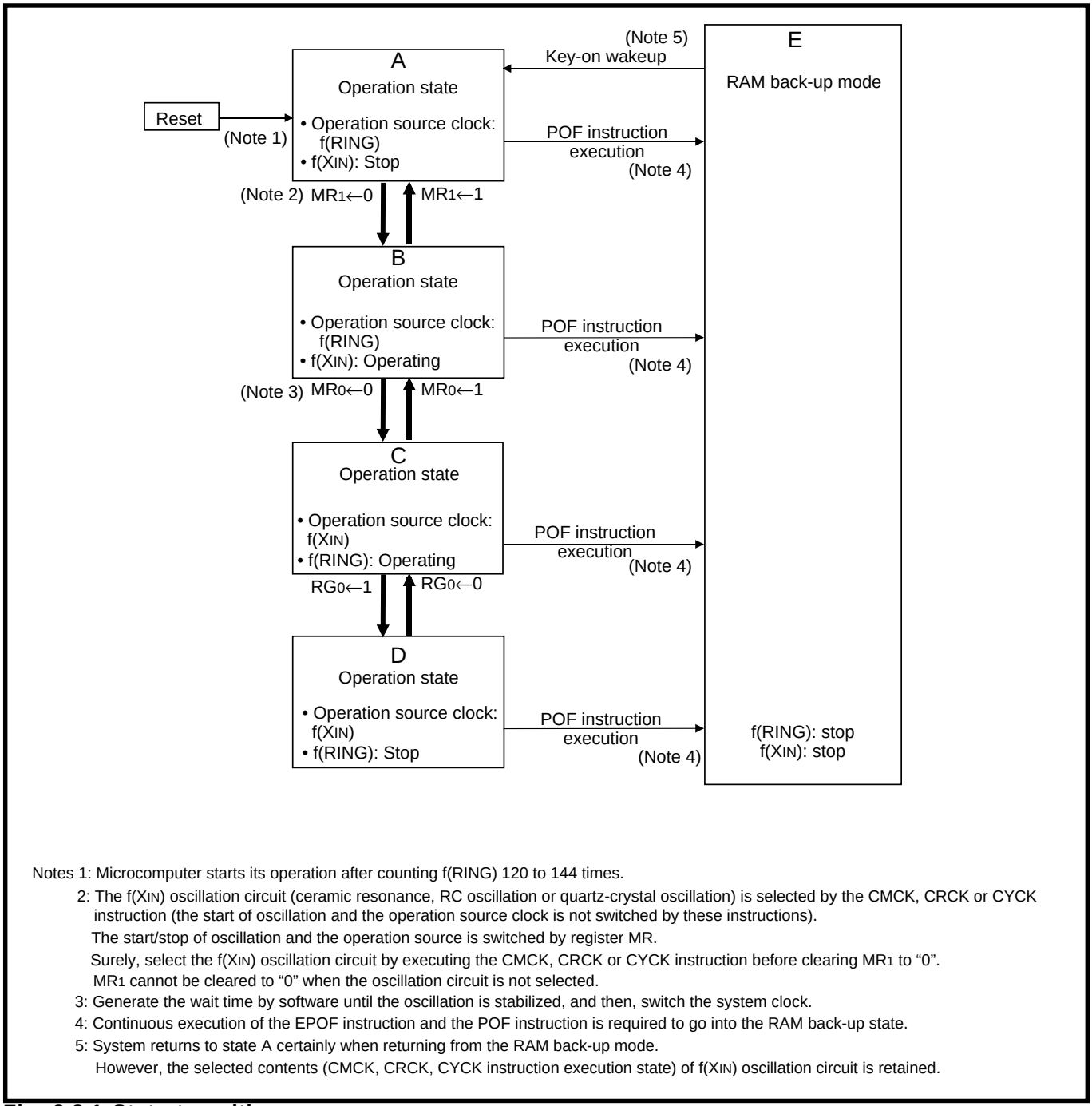
Fig. 2.7.2 Voltage drop detection circuit operation waveform example

Table 2.7.1 Voltage drop detection circuit operation state

VDCE pin	At CPU operating	At RAM back-up
"L"	Invalid	Invalid
"H"	Valid	Valid

2.8 RAM back-up

The 4519 Group has the RAM back-up mode.
Figure 2.8.1 shows the state transition.



Notes 1: Microcomputer starts its operation after counting f(RING) 120 to 144 times.
2: The f(XIN) oscillation circuit (ceramic resonance, RC oscillation or quartz-crystal oscillation) is selected by the CMCK, CRCK or CYCK instruction (the start of oscillation and the operation source clock is not switched by these instructions).
The start/stop of oscillation and the operation source is switched by register MR.
Surely, select the f(XIN) oscillation circuit by executing the CMCK, CRCK or CYCK instruction before clearing MR1 to "0".
MR1 cannot be cleared to "0" when the oscillation circuit is not selected.
3: Generate the wait time by software until the oscillation is stabilized, and then, switch the system clock.
4: Continuous execution of the EPOF instruction and the POF instruction is required to go into the RAM back-up state.
5: System returns to state A certainly when returning from the RAM back-up mode.
However, the selected contents (CMCK, CRCK, CYCK instruction execution state) of f(XIN) oscillation circuit is retained.

Fig. 2.8.1 State transition

2.8.1 RAM back-up mode

The system goes into RAM back-up mode when the **POF** instruction is executed immediately after the **EPOF** instruction is executed. Table 2.8.1 shows the function and state retained at RAM back-up mode. Also, Table 2.8.2 shows the return source from this state.

(1) RAM back-up mode

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

Table 2.8.1 Functions and states retained at RAM back-up mode

Function	RAM back-up
Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) (Note 2)	X
Contents of RAM	O
Interrupt control registers V1, V2	X
Interrupt control registers I1, I2	O
Selected oscillation circuit	O
Clock control register MR	O
Timer 1 to timer 4 functions	(**Note 3**)
Watchdog timer function	X (**Note 4**)
Timer control registers PA, W4	X
Timer control registers W1 to W3, W5, W6	O
Serial I/O function	X
Serial I/O control register J1	O
A/D function	X
A/D control registers Q1 to Q3	O
Voltage drop detection circuit	O (**Note 5**)
Port level	O
Pull-up control registers PU0, PU1	O
Key-on wakeup control registers K0 to K2	O
Port output format control registers FR0 to FR3	O
External interrupt request flags (EXF0, EXF1)	X
Timer interrupt request flags (T1F to T4F)	(**Note 3**)
A/D conversion completion flag (ADF)	X
Serial I/O transmit/receive completion flag SIOF	X
Interrupt enable flag (INTE)	X
Watchdog timer flags (WDF1, WDF2)	X (**Note 4**)
Watchdog timer enable flag (WEF)	X (**Note 4**)

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

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

3: The state of the timer is undefined.

4: Initialize the watchdog timer flag WDF1 with the **WRST** instruction, and then go into the RAM back-up state.

5: The valid/invalid of the voltage drop detection circuit can be controlled only by VDCE pin.

Table 2.8.2 Return source and return condition

Return source		Return condition	Remarks
External wakeup signal	Ports $P0_0$–$P0_3$	Return by an external "H" level or "L" level input, or rising edge ("L"→"H") or falling edge ("H"→"L").	The key-on wakeup function can be selected with 2 port units. Select the return level ("L" level or "H" level), and return condition (return by level or edge) with the register K1 according to the external state before going into the RAM back-up state.
	Ports $P1_0$–$P1_3$	Return by an external "L" level input.	The key-on wakeup function can be selected with 2 port units. Set the port using the key-on wakeup function to "H" level before going into the RAM back-up state.
	INT0 INT1	Return by an external "H" level or "L" level input, or rising edge ("L"→"H") or falling edge ("H"→"L"). The external interrupt request flags (EXF0, EXF1) are not set.	Select the return level ("L" level or "H" level) with the registers I1 and I2 according to the external state, and return condition (return by level or edge) with the register K2 before going into the RAM back-up state.

(3) Start condition identification

When system returns from both RAM back-up mode and reset, program 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 2.8.3 shows the start condition identification, and Figure 2.8.4 shows the start condition identified example.

Table 2.8.3 Start condition identification

	Start condition	P flag	Timer 5 interrupt request flag
Warm start	External wakeup signal input	1	0
Cold start (Reset)	Reset pulse input to $\overline{\text{RESET}}$ pin	0	0
	Reset by watchdog timer		
	Reset by voltage drop detection circuit		
	SRST instruction execution		

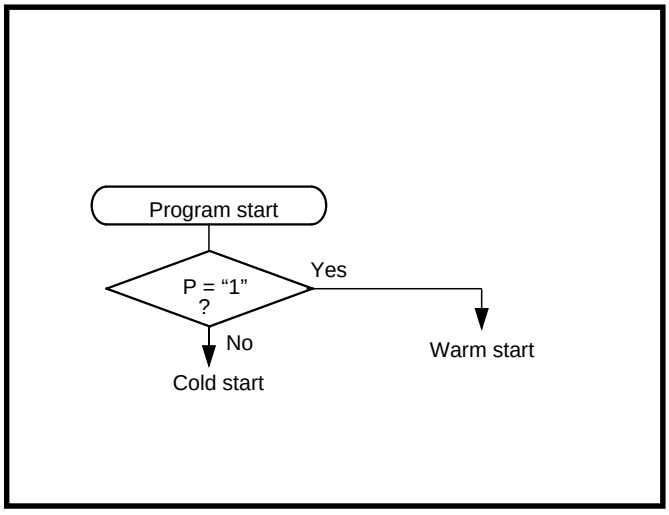


Fig. 2.8.2 Start condition identified example

2.8.2 Related registers

(1) Interrupt control register I1

Table 2.8.4 shows the interrupt control register I1.
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 2.8.4 Interrupt control register I1

Interrupt control register I1		at reset : 0000_2	at RAM back-up : state retained	R/W
$I1_3$	INT0 pin input control bit (**Note 2**)	0	INT0 pin input disabled	
		1	INT0 pin input enabled	
$I1_2$	Interrupt valid waveform for INT0 pin/return level selection bit (**Note 2**)	0	Falling waveform/"L" level ("L" level is recognized with the **SNZI0** instruction)	
		1	Rising waveform/"H" level ("H" level is recognized with the **SNZI0** instruction)	
$I1_1$	INT0 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I1_0$	INT0 pin Timer 1 count start synchronous circuit selection bit	0	Timer 1 count start synchronous circuit not selected	
		1	Timer 1 count start synchronous circuit selected	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: When the contents of $I1_2$ and $I1_3$ are changed, the external interrupt request flag EXF0 may be set to "1". Accordingly, clear EXF0 flag with the **SNZ0** instruction when the bit 0 ($V1_0$) of register V1 to "0". In this time, set the **NOP** instruction after the **SNZ0** instruction, for the case when a skip is performed with the **SNZ0** instruction.
3: When setting the RAM back-up, $I1_1$–$I1_0$ are not used.

(2) Interrupt control register I2

Table 2.8.5 shows the interrupt control register I2.
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 2.8.5 Interrupt control register I2

Interrupt control register I2		at reset : 0000_2	at RAM back-up : state retained	R/W
$I2_3$	INT1 pin input control bit (**Note 2**)	0	INT1 pin input disabled	
		1	INT1 pin input enabled	
$I2_2$	Interrupt valid waveform for INT1 pin/return level selection bit (**Note 2**)	0	Falling waveform/"L" level ("L" level is recognized with the **SNZI1** instruction)	
		1	Rising waveform/"H" level ("H" level is recognized with the **SNZI1** instruction)	
$I2_1$	INT1 pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I2_0$	INT1 pin Timer 3 count start synchronous circuit selection bit	0	Timer 3 count start synchronous circuit not selected	
		1	Timer 3 count start synchronous circuit selected	

Notes 1: "R" represents read enabled, and "W" represents write enabled.
2: When the contents of $I2_2$ and $I2_3$ are changed, the external interrupt request flag EXF1 may be set to "1". Accordingly, clear EXF1 flag with the **SNZ1** instruction when the bit 1 ($V1_1$) of register V1 to "0". In this time, set the **NOP** instruction after the **SNZ1** instruction, for the case when a skip is performed with the **SNZ1** instruction.
3: When setting the RAM back-up, $I2_1$–$I2_0$ are not used.

(3) Pull-up control register PU0

Table 2.8.6 shows the pull-up control register PU0.
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 2.8.6 Pull-up control register PU0

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

(4) Pull-up control register PU1

Table 2.8.7 shows the pull-up control register PU1.
Set the contents of this register through register A with the **TPU1A** instruction.
The contents of register PU1 is transferred to register A with the **TAPU1** instruction.

Table 2.8.7 Pull-up control register PU1

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

(5) Key-on wakeup control register K0

Table 2.8.8 shows the key-on wakeup control register K0.
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 2.8.8 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.

(6) Key-on wakeup control register K1

Table 2.8.9 shows the key-on wakeup control register K1.
Set the contents of this register through register A with the **TK1A** instruction.
The contents of register K1 is transferred to register A with the **TAK1** instruction.

Table 2.8.9 Key-on wakeup control register K1

Key-on wakeup control register K1		at reset : 0000_2	at RAM back-up : state retained	R/W
$K1_3$	Ports $P0_2$ and $P0_3$ return condition selection bit	0	Return by level	
		1	Return by edge	
$K1_2$	Ports $P0_2$ and $P0_3$ valid waveform/level selection bit	0	Falling waveform/“L” level	
		1	Rising waveform/“H” level	
$K1_1$	Ports $P0_1$ and $P0_0$ return condition selection bit	0	Return by level	
		1	Return by edge	
$K1_0$	Ports $P0_1$ and $P0_0$ valid waveform/level selection bit	0	Falling waveform/“L” level	
		1	Rising waveform/“H” level	

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

(7) Key-on wakeup control register K2

Table 2.8.10 shows the key-on wakeup control register K2.
Set the contents of this register through register A with the **TK2A** instruction.
The contents of register K2 is transferred to register A with the **TAK2** instruction.

Table 2.8.10 Key-on wakeup control register K2

Key-on wakeup control register K2		at reset : 0000_2	at RAM back-up : state retained	R/W
$K2_3$	INT1 pin return condition selection bit	0	Return by level	
		1	Return by edge	
$K2_2$	INT1 pin key-on wakeup control bit	0	Key-on wakeup not used	
		1	Key-on wakeup used	
$K2_1$	INT0 pin return condition selection bit	0	Returned by level	
		1	Returned by edge	
$K2_0$	INT0 pin 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.

2.8.3 Notes on use

(1) 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 and the **POF** instruction.

(2) Key-on wakeup function

After checking none of the return condition for ports (P0, P1, INT0 and INT1 specified with register K0–K2) with valid key-on wakeup function is satisfied, execute the **POF** instruction.

If at least one of return condition for ports with valid key-on wakeup function is satisfied, system returns from the RAM back-upn state immediately after the **POF** instruction is executed.

(3) Return from RAM back-up mode

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

The initial value of the following registers are undefined at RAM back-up. After system is returned from RAM back-up mode, set initial values.

- Register Z (2 bits)
- Register X (4 bits)
- Register Y (4 bits)
- Register D (3 bits)
- Register E (8 bits)

(4) Watchdog timer

- The watchdog timer function is valid after system is returned from the RAM back-up state. When not using the watchdog timer function, stop the watchdog timer function with the **DWDT** instruction and the **WRST** instruction continuously every system is returned from the RAM back-up.
- When the watchdog timer function and RAM back-up function are used at the same time, initialize the flag WDF1 with the **WRST** instruction before system goes into the RAM back-up state.

(5) Port $P3_0$/INT0 pin

When the RAM back-up mode is used by clearing the bit 3 of register I1 to “0” and setting the input of INT0 pin to be disabled, be careful about the following note.

- When the input of INT0 pin is disabled (register $I1_3$ = “0”), clear bit 0 of register K2 to “0” to invalidate the key-on wakeup before system goes into the RAM back-up mode.

(6) Port $P3_1$/INT1 pin

When the RAM back-up mode is used by clearing the bit 3 of register I2 to “0” and setting the input of INT1 pin to be disabled, be careful about the following note.

- When the input of INT1 pin is disabled (register $I2_3$ = “0”), clear bit 2 of register K2 to “0” to invalidate the key-on wakeup before system goes into the RAM back-up mode.

2.9 Oscillation circuit

The 4519 Group has an internal oscillation circuit to produce the clock required for microcomputer operation. The 4519 Group operates by the on-chip oscillator clock (f(RING)) which is the internal oscillator after system is released from reset.
Also, the ceramic resonator, the RC oscillation or quartz-crystal oscillator can be used for the main clock (f(XIN)) of the 4519 Group. The CMCK instruction, CRCK instruction or CYCK instruction is executed to select the ceramic resonator, RC oscillator or quartz-crystal oscillator respectively.

2.9.1 Oscillation operation

System clock is supplied to CPU and peripheral device as the base clock for the microcomputer operation. The system clock f(XIN) or f(RING) is selected by bit 0 of register MR.
The oscillation start/stop of main clock f(XIN) is controlled by bit 1 of register MR.
Also, an operation mode of a selected clock is selected from the followings by bits 3 and 2 of register MR.

- through mode (f(XIN)) (not divided),
- frequency divided by 2 mode (f(XIN)/2),
- frequency divided by 4 mode (f(XIN)/4), or
- frequency divided by 8 mode (f(XIN)/8)

Figure 2.9.1 shows the structure of the clock control circuit.

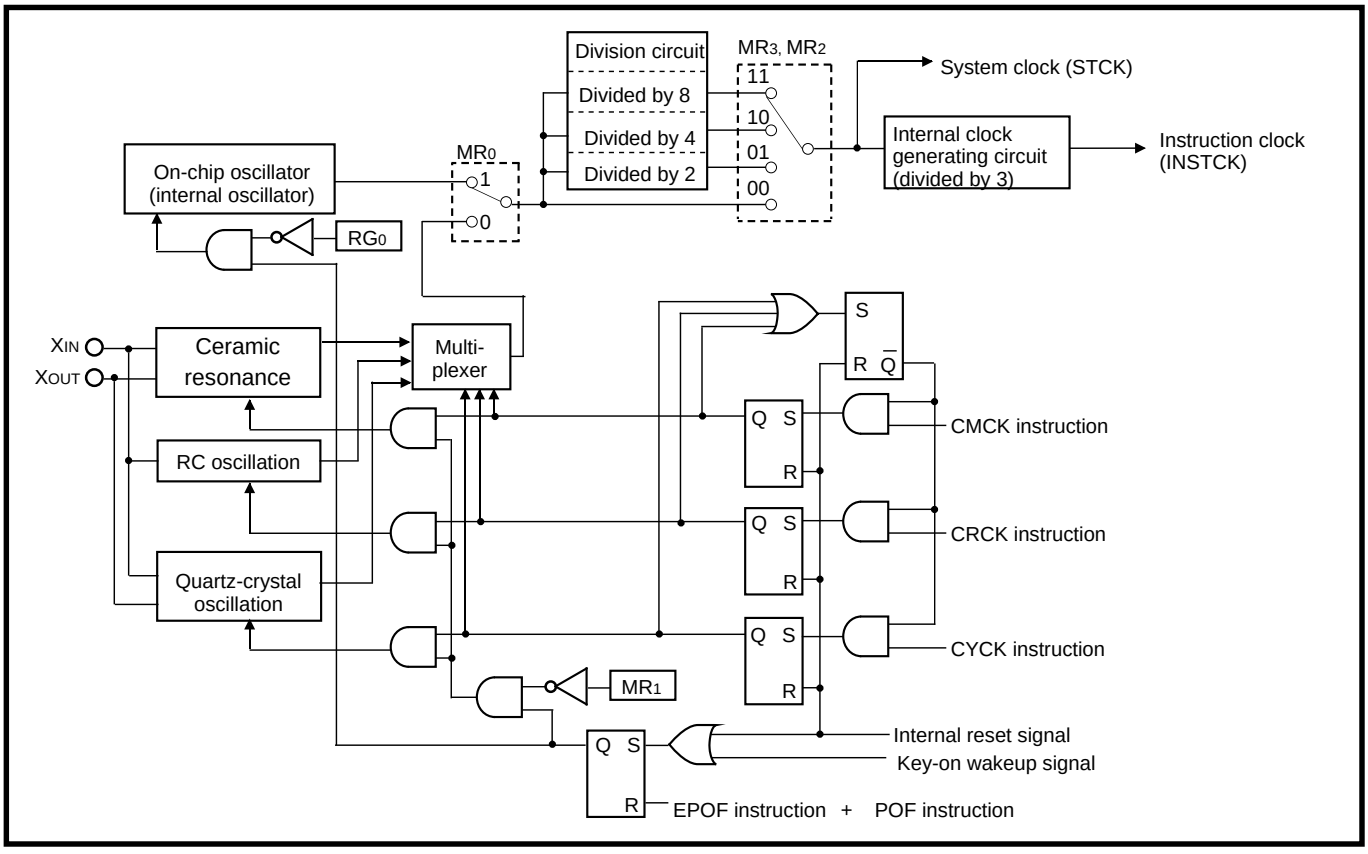


Fig. 2.9.1 Structure of clock control circuit

2.9.2 Related register

(1) Clock control register MR

Table 2.9.1 shows the clock control register MR.
Set the contents of this register through register A with the **TMRA** instruction.
The contents of register MR is transferred to register A with the **TAMR** instruction.

Table 2.9.1 Clock control register MR

Clock control register MR		at reset : 1111_2		at RAM back-up : state retained	R/W
		MR_3	MR_2	Operation mode	
MR_3	Operation mode selection bits	0	0	Through-mode (frequency not divided)	
MR_2		0	1	Frequency divided by 2 mode	
		1	0	Frequency divided by 4 mode	
		1	1	Frequency divided by 8 mode	
MR_1	Main clock f(X_{IN}) oscillation circuit control bit	0		Main clock oscillation enabled	
		1		Main clock oscillation stop	
MR_0	System clock oscillation source selection bit	0		Main clock (f(X_{IN})	
		1		Sub-clock (f(X_{CIN}))	

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

(2) Clock control register RG

Table 2.9.2 shows the clock control register RG.
Set the contents of this register through register A with the **TRGA** instruction.

Table 2.9.2 Clock control register RG

Clock control register RG		at reset : 0_2	at RAM back-up : state retained	W
RG_0	On-chip oscillator (f(RING)) control bit	0	On-chip oscillator (f(RING)) oscillation enabled	
		1	On-chip oscillator (f(RING)) oscillation stop	

2.9.3 Notes on use

(1) Clock control

Execute the main clock (f(XIN)) selection instruction (**CMCK**, **CRCK** or **CYCK** instruction) in the initial setting routine of program (executing it in address 0 in page 0 is recommended).
The oscillation circuit by the **CMCK**, **CRCK** or **CYCK** instruction can be selected only at once. The oscillation circuit corresponding to the first executed one of these instructions is valid.
The **CMCK**, **CRCK** or **CYCK** instructions can be used only to select main clock (f(XIN)). In this time, the start of oscillation and the switch of system clock are not performed.
When the **CMCK**, **CRCK** or **CYCK** instructions are never executed, main clock (f(XIN)) cannot be used and system can be operated only by on-chip oscillator.
The no operated clock source (f(RING)) or (f(XIN)) cannot be used for the system clock. Also, the clock source (f(RING) or f(XIN)) selected for the system clock cannot be stopped.

(2) On-chip oscillator

The clock frequency of the on-chip oscillator depends on the supply voltage and the operation temperature range.
Be careful that margin of frequencies when designing application products.
When considering the oscillation stabilize wait time at the switch of clock, be careful that the margin of frequencies of the on-chip oscillator clock.

(3) External clock

When the external clock signal for the main clock (f(XIN)) is used, connect the clock source to XIN pin and XOUT pin open. In program, after the CMCK instruction is executed, set main clock (f(XIN)) oscillation start to be enabled (MR1=0).
For this product, when RAM back-up mode and main clock (f(XIN)) stop (MR1=1), XIN pin is fixed to “H” in order to avoid the through current by floating of internal logic. The XIN pin is fixed to “H” until main clock (f(XIN)) oscillation start to be valid (MR1=0) by the CMCK instruction from reset state. Accordingly, when an external clock is used, connect a 1 kΩ or more resistor to XIN pin in series to limit of current by competitive signal.

(4) 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

3.1.1 Absolute maximum ratings

Table 3.1.1 Absolute maximum ratings

Symbol	Parameter	Conditions		Ratings	Unit
V_{DD}	Supply voltage			–0.3 to 6.5	V
V_I	Input voltage P0, P1, P2, P3, P4, P5, P6, D_0–D_7, $\overline{RESET}$, X_{IN}, VDCE			–0.3 to V_{DD}+0.3	V
V_I	Input voltage SCK, SIN, CNTR0, CNTR1, INT0, INT1			–0.3 to V_{DD}+0.3	V
V_I	Input voltage A_{IN0}–A_{IN7}			–0.3 to V_{DD}+0.3	V
V_O	Output voltage P0, P1, P2, P3, P4, P5, P6, D_0–D_7, $\overline{RESET}$	Output transistors in cut-off state		–0.3 to V_{DD}+0.3	V
V_O	Output voltage SCK, SOUT, CNTR0, CNTR1	Output transistors in cut-off state		–0.3 to V_{DD}+0.3	V
V_O	Output voltage X_{OUT}			–0.3 to V_{DD}+0.3	V
P_d	Power dissipation	Ta = 25 °C	42P2R-A	300	mW
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} = 1.8 to 5.5 V, unless otherwise noted)
(One Time PROM version: Ta = –20 °C to 85 °C, V_{DD} = 2.5 to 5.5 V, unless otherwise noted)

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{DD}	Supply voltage (when ceramic resonator/on-chip oscillator is used)	Mask ROM version	f(STCK) ≤ 6 MHz	4.0		5.5	V
			f(STCK) ≤ 4.4 MHz	2.7		5.5	
			f(STCK) ≤ 2.2 MHz	2.0		5.5	
			f(STCK) ≤ 1.1 MHz	1.8		5.5	
		One Time PROM version	f(STCK) ≤ 6 MHz	4.0		5.5	
			f(STCK) ≤ 4.4 MHz	2.7		5.5	
			f(STCK) ≤ 2.2 MHz	2.5		5.5	
V_{DD}	Supply voltage (when RC oscillation is used)	f(STCK) ≤ 4.4 MHz		2.7		5.5	V
V_{DD}	Supply voltage (when quartz-crystal oscillator is used)	Mask ROM version	$f(X_{IN})$ ≤ 50 kHz	2.0		5.5	V
		One Time PROM version	$f(X_{IN})$ ≤ 50 kHz	2.5		5.5	V
V_{RAM}	RAM back-up voltage	Mask ROM version	at RAM back-up mode	1.6			V
		One Time PROM version	at RAM back-up mode	2.0			V
V_{SS}	Supply voltage				0		V
V_{IH}	"H" level input voltage	P0, P1, P2, P3, P4, P5, P6, D_0–D_7, VDCE, X_{IN}		0.8V_{DD}		V_{DD}	V
V_{IH}	"H" level input voltage	$\overline{RESET}$		0.85V_{DD}		V_{DD}	V
V_{IH}	"H" level input voltage	S_{CK}, S_{IN}, CNTR0, CNTR1, INT0, INT1		0.85V_{DD}		V_{DD}	V
V_{IL}	"L" level input voltage	P0, P1, P2, P3, P4, P5, P6, D_0–D_7, VDCE, X_{IN}		0		0.2V_{DD}	V
V_{IL}	"L" level input voltage	$\overline{RESET}$		0		0.3V_{DD}	V
V_{IL}	"L" level input voltage	S_{CK}, S_{IN}, CNTR0, CNTR1, INT0, INT1		0		0.15V_{DD}	V
I_{OH}(peak)	"H" level peak output current	P0, P1, P5, D_0–D_7	V_{DD} = 5 V			–20	mA
		CNTR0, CNTR1	V_{DD} = 3 V			–10	
I_{OH}(avg)	"H" level average output current (Note)	P0, P1, P5, D_0–D_7	V_{DD} = 5 V			–10	mA
		CNTR0, CNTR1	V_{DD} = 3 V			–5	
I_{OL}(peak)	"L" level peak output current	P0, P1, P2, P4, P5, P6	V_{DD} = 5 V			24	mA
		S_{CK}, S_{OUT}	V_{DD} = 3 V			12	
I_{OL}(peak)	"L" level peak output current	P3, $\overline{RESET}$	V_{DD} = 5 V			10	mA
			V_{DD} = 3 V			4	
I_{OL}(peak)	"L" level peak output current	D_0–D_5	V_{DD} = 5 V			24	mA
			V_{DD} = 3 V			12	
I_{OL}(peak)	"L" level peak output current	D_6, D_7	V_{DD} = 5 V			40	mA
		CNTR0, CNTR1	V_{DD} = 3 V			30	
I_{OL}(avg)	"L" level average output current (Note)	P0, P1, P2, P4, P5, P6	V_{DD} = 5 V			12	mA
		S_{CK}, S_{OUT}	V_{DD} = 3 V			6	
I_{OL}(avg)	"L" level average output current (Note)	P3, $\overline{RESET}$	V_{DD} = 5 V			5	mA
			V_{DD} = 3 V			2	
I_{OL}(avg)	"L" level average output current (Note)	D_0–D_5	V_{DD} = 5 V			15	mA
			V_{DD} = 3 V			7	
I_{OL}(avg)	"L" level average output current (Note)	D_6, D_7	V_{DD} = 5 V			30	mA
		CNTR0, CNTR1	V_{DD} = 3 V			15	
ΣI_{OH}(avg)	"H" level total average current	P5, D_0–D_7, CNTR0, CNTR1				–60	mA
		P0, P1				–60	
ΣI_{OL}(avg)	"L" level total average current	P2, P5, D_0–D_7, $\overline{RESET}$, CNTR0, CNTR1				80	mA
		P0, P1, P3, P4, P6				80	

Note: The average output current is the average value during 100 ms.

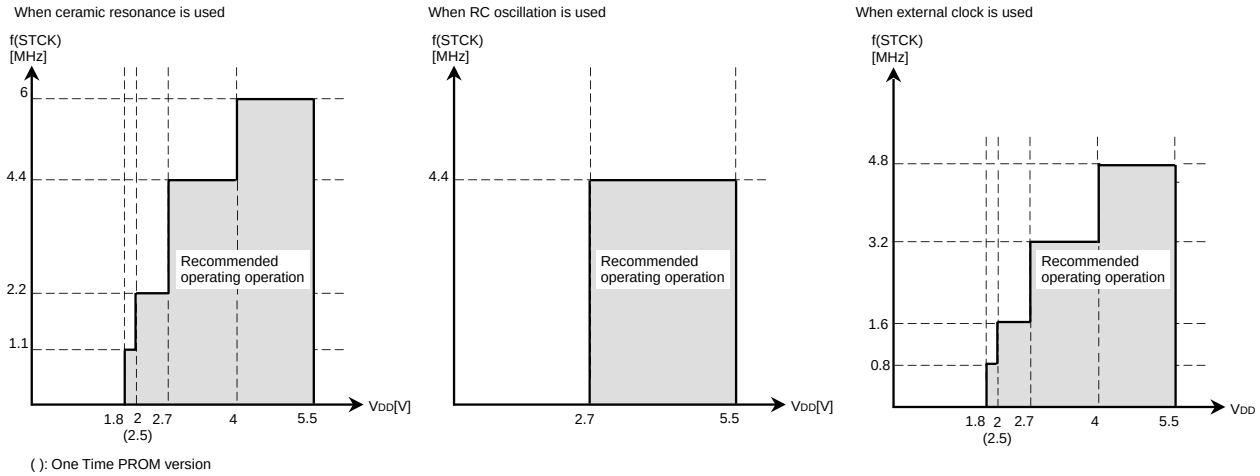
Table 3.1.3 Recommended operating conditions 2

(Mask ROM version: Ta = –20 °C to 85 °C, VDD = 1.8 to 5.5 V, unless otherwise noted)
(One Time PROM version: Ta = –20 °C to 85 °C, VDD = 2.5 to 5.5 V, unless otherwise noted)

Symbol	Parameter	Conditions			Limits			Unit
					Min.	Typ.	Max.	
f(XIN)	Oscillation frequency (with a ceramic resonator)	Mask ROM version	Through mode	VDD = 4.0 to 5.5 V			6.0	MHz
				VDD = 2.7 to 5.5 V			4.4	
				VDD = 2.0 to 5.5 V			2.2	
				VDD = 1.8 to 5.5 V			1.1	
			Frequency/2 mode	VDD = 2.7 to 5.5 V			6.0	
				VDD = 2.0 to 5.5 V			4.4	
				VDD = 1.8 to 5.5 V			2.2	
			Frequency/4, 8 mode	VDD = 2.0 to 5.5 V			6.0	
				VDD = 1.8 to 5.5 V			4.4	
		One Time PROM version	Through mode	VDD = 4.0 to 5.5 V			6.0	
				VDD = 2.7 to 5.5 V			4.4	
				VDD = 2.5 to 5.5 V			2.2	
			Frequency/2 mode	VDD = 2.7 to 5.5 V			6.0	
				VDD = 2.5 to 5.5 V			4.4	
			Frequency/4, 8 mode	VDD = 2.5 to 5.5 V			6.0	
f(XIN)	Oscillation frequency (at RC oscillation) (Note)	VDD = 2.7 to 5.5 V					4.4	MHz
f(XIN)	Oscillation frequency (with a ceramic resonator selected, external clock input)	Mask ROM version	Through mode	VDD = 4.0 to 5.5 V			4.8	MHz
				VDD = 2.7 to 5.5 V			3.2	
				VDD = 2.0 to 5.5 V			1.6	
				VDD = 1.8 to 5.5 V			0.8	
			Frequency/2 mode	VDD = 2.7 to 5.5 V			4.8	
				VDD = 2.0 to 5.5 V			3.2	
				VDD = 1.8 to 5.5 V			1.6	
			Frequency/4, 8 mode	VDD = 2.0 to 5.5 V			4.8	
				VDD = 1.8 to 5.5 V			3.2	
		One Time PROM version	Through mode	VDD = 4.0 to 5.5 V			4.8	
				VDD = 2.7 to 5.5 V			3.2	
				VDD = 2.5 to 5.5 V			1.6	
			Frequency/2 mode	VDD = 2.7 to 5.5 V			4.8	
				VDD = 2.5 to 5.5 V			3.2	
			Frequency/4, 8 mode	VDD = 2.5 to 5.5 V			4.8	

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

<System clock (STCK) operating condition map>



( ): One Time PROM version

Table 3.1.4 Recommended operating conditions 3

(Mask ROM version: Ta = –20 °C to 85 °C, V_{DD} = 1.8 to 5.5 V, unless otherwise noted)
(One Time PROM version: Ta = –20 °C to 85 °C, V_{DD} = 2.5 to 5.5 V, unless otherwise noted)

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
f(X_{IN})	Oscillation frequency (with a quartz-crystal oscillator)	Mask ROM version	V_{DD} = 2.0 to 5.5 V			50	kHz
		One Time PROM version	V_{DD} = 2.5 to 5.5 V			50	
f(CNTR)	Timer external input frequency	CNTR0, CNTR1				f(STCK)/6	Hz
tw(CNTR)	Timer external input period (“H” and “L” pulse width)	CNTR0, CNTR1		3/f(STCK)			s
f(S_{CK})	Serial I/O external input frequency	S_{CK}				f(STCK)/6	Hz
tw(S_{CK})	Serial I/O external input frequency (“H” and “L“ pulse width)	S_{CK}		3/f(STCK)			s
TPON	Power-on reset circuit valid supply voltage rising time	Mask ROM version	V_{DD} = 0 → 1.8 V			100	µs
		One Time PROM version	V_{DD} = 0 → 2.5 V			100	

3.1.3 Electrical characteristics

Table 3.1.5 Electrical characteristics 1

(Mask ROM version: Ta = –20 °C to 85 °C, V_{DD} = 1.8 to 5.5 V, unless otherwise noted)
(One Time PROM version: Ta = –20 °C to 85 °C, V_{DD} = 2.5 to 5.5 V, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	"H" level output voltage P0, P1, P5, D_0–D_7, CNTR0, CNTR1	V_{DD} = 5 V	I_{OH} = –10 mA	3			V
			I_{OH} = –3 mA	4.1			
		V_{DD} = 3 V	I_{OH} = –5 mA	2.1			
			I_{OH} = –1 mA	2.4			
V_{OL}	"L" level output voltage P0, P1, P2, P4, P5, P6 S_{CK}, S_{OUT}	V_{DD} = 5 V	I_{OL} = 12 mA			2	V
			I_{OL} = 4 mA			0.9	
		V_{DD} = 3 V	I_{OL} = 6 mA			0.9	
			I_{OL} = 2 mA			0.6	
V_{OL}	"L" level output voltage P3, $\overline{RESET}$	V_{DD} = 5 V	I_{OL} = 5 mA			2	V
			I_{OL} = 1 mA			0.9	
		V_{DD} = 3 V	I_{OL} = 2 mA			0.9	
V_{OL}	"L" level output voltage D_0–D_5	V_{DD} = 5 V	I_{OL} = 15 mA			2	V
			I_{OL} = 5 mA			0.9	
		V_{DD} = 3 V	I_{OL} = 9 mA			1.4	
			I_{OL} = 3 mA			0.9	
V_{OL}	"L" level output voltage D_6, D_7, CNTR0, CNTR1	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	
			I_{OL} = 5 mA			0.9	
I_{IH}	"H" level input current P0, P1, P2, P3, P4, P5, P6, D_0–D_7, VDCE, $\overline{RESET}$, S_{CK}, S_{IN}, CNTR0, CNTR1, INT0, INT1	V_I = V_{DD} Ports P4, P6 selected				2	μA
I_{IL}	"L" level input current P0, P1, P2, P3, P4, P5, P6, D_0–D_7, VDCE, S_{CK}, S_{IN}, CNTR0, CNTR1, INT0, INT1	V_I = 0 V P0, P1 No pull-up Ports P4, P6 selected				–2	μA
R_{PU}	Pull-up resistor value P0, P1, $\overline{RESET}$	V_I = 0 V	V_{DD} = 5 V	30	60	125	kΩ
			V_{DD} = 3 V	50	120	250	
$V_{T+} – V_{T–}$	Hysteresis S_{CK}, S_{IN}, CNTR0, CNTR1, INT0, INT1	V_{DD} = 5 V			0.2		V
		V_{DD} = 3 V			0.2		
$V_{T+} – V_{T–}$	Hysteresis $\overline{RESET}$	V_{DD} = 5 V			1		V
		V_{DD} = 3 V			0.4		
f(RING)	On-chip oscillator clock frequency	V_{DD} = 5 V		200	500	700	kHz
		V_{DD} = 3 V		100	250	400	
		Mask ROM version	V_{DD} = 1.8 V	30	120	200	
$\Delta f(X_{IN})$	Frequency error (with RC oscillation, error of external R, C not included) (Note)	V_{DD} = 5 V ± 10 %, Ta = 25 °C				±17	%
		V_{DD} = 3 V ± 10 %, Ta = 25 °C				±17	%

Note: When RC oscillation is used, use the external 30 pF or 33 pF capacitor (C).

Table 3.1.6 Electrical characteristics 2

(Mask ROM version: Ta = –20 °C to 85 °C, V_{DD} = 1.8 to 5.5 V, unless otherwise noted)
(One Time PROM version: Ta = –20 °C to 85 °C, V_{DD} = 2.5 to 5.5 V, unless otherwise noted)

Symbol	Parameter		Test conditions		Limits			Unit
					Min.	Typ.	Max.	
I_{DD}	Supply current	at active mode (with a ceramic resonator, on-chip oscillator stop)	V_{DD} = 5 V, f(X_{IN}) = 6 MHz	f(STCK) = f(X_{IN})/8		1.4	2.8	mA
				f(STCK) = f(X_{IN})/4		1.6	3.2	
				f(STCK) = f(X_{IN})/2		2.0	4.0	
				f(STCK) = f(X_{IN})		2.8	5.6	
			V_{DD} = 5 V, f(X_{IN}) = 4 MHz	f(STCK) = f(X_{IN})/8		1.1	2.2	mA
				f(STCK) = f(X_{IN})/4		1.2	2.4	
				f(STCK) = f(X_{IN})/2		1.5	3.0	
				f(STCK) = f(X_{IN})		2.0	4.0	
			V_{DD} = 3 V, f(X_{IN}) = 4 MHz	f(STCK) = f(X_{IN})/8		0.4	0.8	mA
				f(STCK) = f(X_{IN})/4		0.5	1.0	
				f(STCK) = f(X_{IN})/2		0.6	1.2	
				f(STCK) = f(X_{IN})		0.8	1.6	
		at active mode (with a quartz-crystal oscillator, on-chip oscillator stop)	V_{DD} = 5 V, f(X_{IN}) = 32 kHz	f(STCK) = f(X_{IN})/8		55	110	μA
				f(STCK) = f(X_{IN})/4		60	120	
				f(STCK) = f(X_{IN})/2		65	130	
				f(STCK) = f(X_{IN})		70	140	
			V_{DD} = 3 V, f(X_{IN}) = 32 kHz	f(STCK) = f(X_{IN})/8		12	24	μA
				f(STCK) = f(X_{IN})/4		13	26	
				f(STCK) = f(X_{IN})/2		14	28	
				f(STCK) = f(X_{IN})		15	30	
		at active mode (with an on-chip oscillator, f(X_{IN}) stop)	V_{DD} = 5 V	f(STCK) = f(RING)/8		50	100	μA
				f(STCK) = f(RING)/4		70	140	
				f(STCK) = f(RING)/2		100	200	
				f(STCK) = f(RING)		150	300	
			V_{DD} = 3 V	f(STCK) = f(RING)/8		10	20	μA
				f(STCK) = f(RING)/4		15	30	
				f(STCK) = f(RING)/2		20	40	
				f(STCK) = f(RING)		35	70	
		at RAM back-up mode (POF instruction execution)	Ta = 25 °C			0.1	3	μA
			V_{DD} = 5 V				10	
			V_{DD} = 3 V				6	

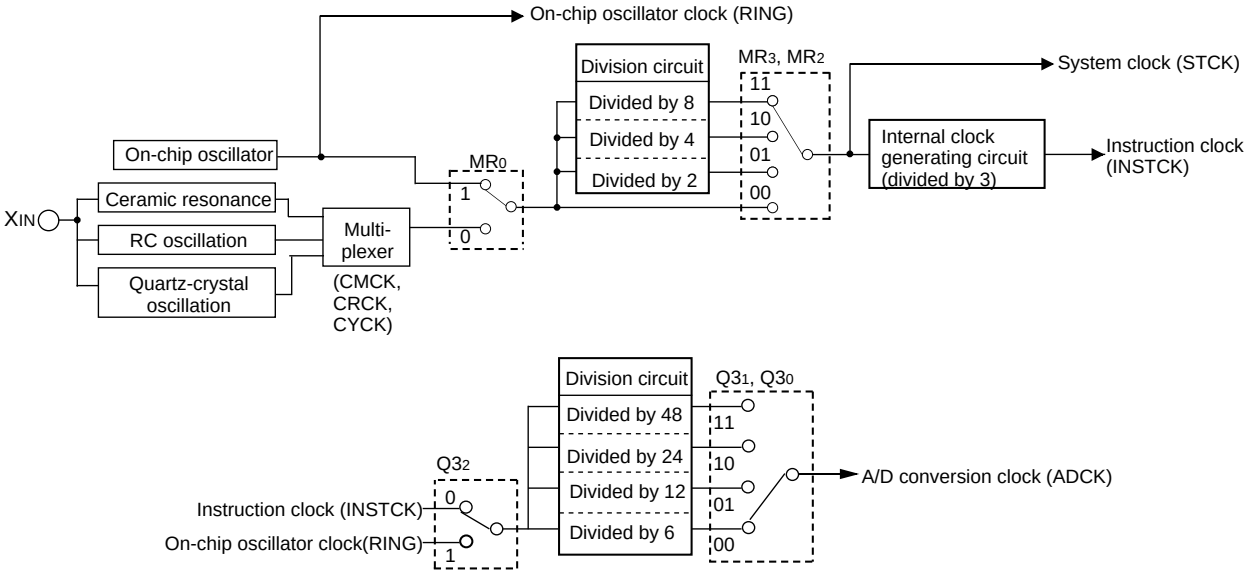
3.1.4 A/D converter recommended operating conditions

Table 3.1.7 A/D converter recommended operating conditions

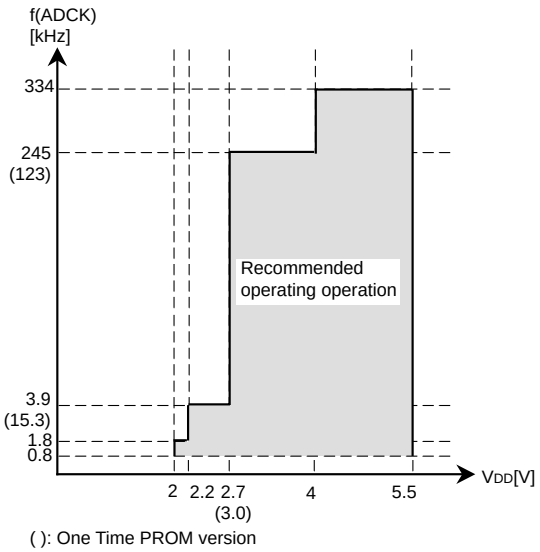
(Comparator mode included, Ta = –20 °C to 85 °C, unless otherwise noted)

Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{DD}	Supply voltage	Mask ROM version		2.0		5.5	V
		One Time PROM version		3.0		5.5	
V_{IA}	Analog input voltage			0		V_{DD}	V
f(ADCK)	A/D conversion clock frequency (Note)	Mask ROM version	V_{DD} = 4.0 to 5.5 V	0.8		334	kHz
			V_{DD} = 2.7 to 5.5 V	0.8		245	
			V_{DD} = 2.2 to 5.5 V	0.8		3.9	
			V_{DD} = 2.0 to 5.5 V	0.8		1.8	
		One Time PROM version	V_{DD} = 4.0 to 5.5 V	0.8		334	
			V_{DD} = 3.0 to 5.5 V	0.8		123	

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



<Operating condition map of A/D conversion clock (ADCK) >



( ): One Time PROM version

Table 3.1.8 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	2.7 (3.0) V ≤ V_{DD} ≤ 5.5 V((): One Time PROM version)				±2	LSB
		Mask ROM version	2.2 V ≤ V_{DD} < 2.7 V			±4	
–	Differential non-linearity error	2.2 (3.0) V ≤ V_{DD} ≤ 5.5 V (():One Time PROM version)				±0.9	LSB
V_{0T}	Zero transition voltage	Mask ROM version	V_{DD} = 5.12 V	0	10	20	mV
			V_{DD} = 3.072 V	0	7.5	15	
			V_{DD} = 2.56 V	0	7.5	15	
		One Time PROM version	V_{DD} = 5.12 V	0	15	30	
			V_{DD} = 3.072 V	3	13	23	
V_{FST}	Full-scale transition voltage	Mask ROM version	V_{DD} = 5.12 V	5105	5115	5125	mV
			V_{DD} = 3.072 V	3064.5	3072	3079.5	
			V_{DD} = 2.56 V	2552.5	2560	2567.5	
		One Time PROM version	V_{DD} = 5.12 V	5100	5115	5130	
			V_{DD} = 3.072 V	3065	3075	3085	
–	Absolute accuracy (Quantization error excluded)	Mask ROM version	2.0 V ≤ V_{DD} < 2.2 V			±8	LSB
I_{ADD}	A/D operating current (Note 1)	V_{DD} = 5 V			150	450	µA
		V_{DD} = 3 V			75	225	
T_{CONV}	A/D conversion time	f(X_{IN}) = 6 MHz f(STCK) = f(X_{IN}) (X_{IN} through mode) ADCK=INSTCK/6				31	µs
–	Comparator resolution					8	bits
–	Comparator error (Note 2)	Mask ROM version	V_{DD} = 5.12 V			±20	mV
			V_{DD} = 3.072 V			±15	
			V_{DD} = 2.56 V			±15	
		One Time PROM version	V_{DD} = 5.12 V			±30	
			V_{DD} = 3.072 V			±23	
–	Comparator comparison time	f(X_{IN}) = 6 MHz f(STCK) = f(X_{IN}) (X_{IN} through mode) ADCK=INSTCK/6				4	µs

Notes 1: When the A/D converter is used, I_{ADD} is added to I_{DD} (supply current).
2: 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 D/A 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

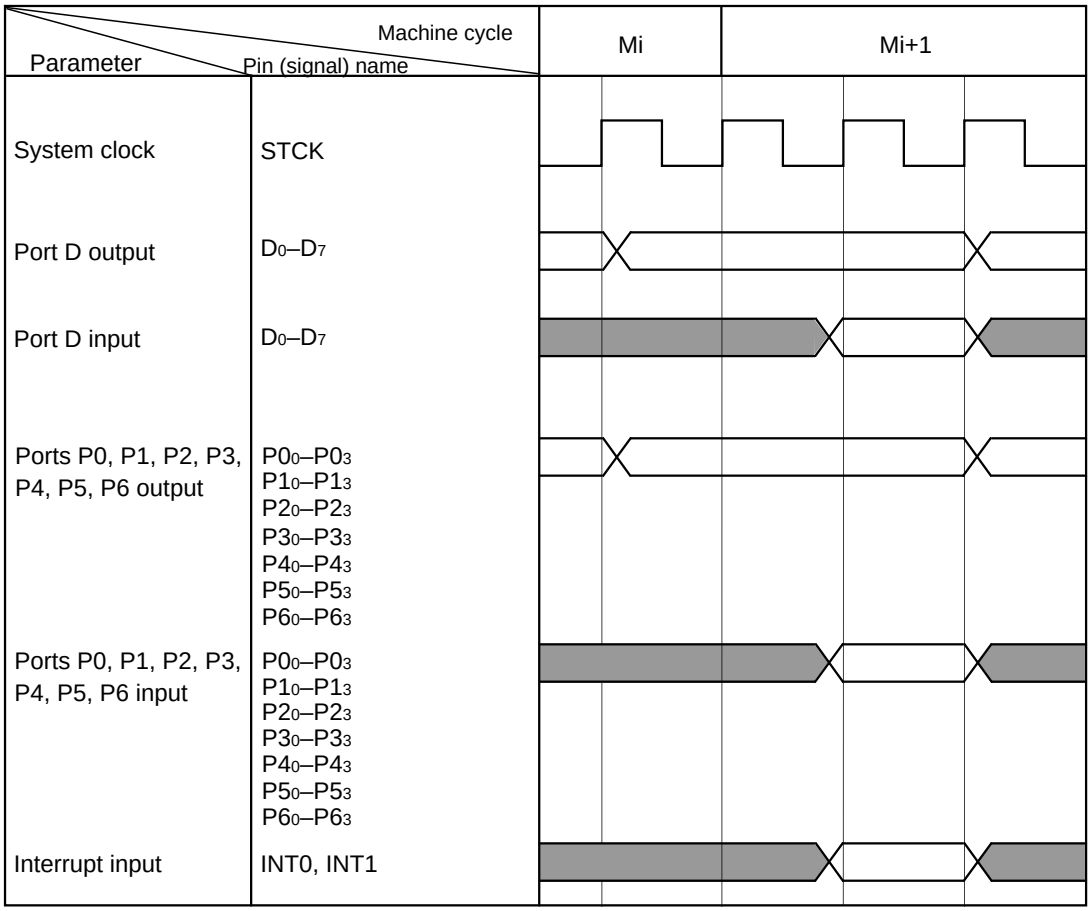
Table 3.1.9 Voltage drop detection circuit characteristics

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

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{RST-}	Detection voltage (reset occurs) (Note 1)	Ta = 25 °C	3.3	3.5	3.7	V
			2.7		4.2	
			2.6		4.2	
V_{RST+}	Detection voltage (reset release) (Note 2)	Ta = 25 °C	3.5	3.7	3.9	V
			2.9		4.4	
			2.8		4.4	
$V_{RST+} - V_{RST-}$	Detection voltage hysteresis			0.2		V
I_{RST}	Operation current (Note 3)	V_{DD} = 5 V		50	100	μA
		V_{DD} = 3 V		30	60	
T_{RST}	Detection time	$V_{DD} \rightarrow (V_{RST-} - 0.1\ V)$ (Note 4)		0.2	1.2	ms

Notes 1: The detected voltage (V_{RST-}) is defined as the voltage when reset occurs when the supply voltage (V_{DD}) is falling.
2: The detected voltage (V_{RST+}) is defined as the voltage when reset is released when the supply voltage (V_{DD}) is rising from reset occurs.
3: When the voltage drop detection circuit is used (VDCE pin = "H"), I_{RST} is added to I_{DD} (power current).
4: The detection time (T_{RST}) is defined as the time until reset occurs when the supply voltage (V_{DD}) is falling to [V_{RST-} – 0.1 V].

3.1.6 Basic timing diagram



3.2 Typical characteristics

As for the standard characteristics, refer to “Renesas Technology Corp.” Homepage.

http://www.renesas.com/en/720

3.3 List of precautions

3.3.1 Program counter

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

3.3.2 Stack registers (SKs)

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.

3.3.3 Notes on I/O port

(1) Note when an I/O port 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.

As for the port which has the output structure selection function, select the N-channel open-drain output structure.

(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 One Time 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) Multifunction

- Be careful that the output of ports $P3_0$ and $P3_1$ can be used even when INT0 and INT1 pins are selected.
- Be careful that the input of ports $P2_0$–$P2_2$ can be used even when S_{IN}, S_{OUT} and S_{CK} pins are selected.
- Be careful that the input/output of port D_6 can be used even when input of CNTR0 pin is selected.
- Be careful that the input of port D_6 can be used even when output of CNTR0 pin is selected.
- Be careful that the input/output of port D_7 can be used even when input of CNTR1 pin is selected.
- Be careful that the input of port D_7 can be used even when output of CNTR1 pin is selected.

(4) Connection of unused pins

Table 3.3.1 shows the connections of unused pins.

(5) SD, RD, SZD instructions

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

(6) Port $P3_0$/INT0 pin

When the RAM back-up mode is used by clearing the bit 3 of register I1 to "0" and setting the input of INT0 pin to be disabled, be careful about the following note.

- When the input of INT0 pin is disabled (register $I1_3$ = "0"), clear bit 0 of register K2 to "0" to invalidate the key-on wakeup before system goes into the RAM back-up mode.

(7) Port $P3_1$/INT1 pin

When the RAM back-up mode is used by clearing the bit 3 of register I2 to "0" and setting the input of INT1 pin to be disabled, be careful about the following note.

- When the input of INT1 pin is disabled (register $I2_3$ = "0"), clear bit 2 of register K2 to "0" to invalidate the key-on wakeup before system goes into the RAM back-up mode.

Table 3.3.1 Connections of unused pins

Pin	Connection	Usage condition	
X_{IN}	Open.	Internal oscillator is selected.	(**Note 1**)
X_{OUT}	Open.	Internal oscillator is selected.	(**Note 1**)
		RC oscillator is selected.	(**Note 2**)
		External clock input is selected for main clock.	(**Note 3**)
D_0–D_5	Open.	———	
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.	(**Note 4**)
D_6/CNTR0	Open.	CNTR0 input is not selected for timer 1 count source.	
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.	(**Note 4**)
D_7/CNTR1	Open.	CNTR1 input is not selected for timer 3 count source.	
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.	(**Note 4**)
$P0_0$–$P0_3$	Open.	The key-on wakeup function is not selected.	(**Note 6**)
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.	(**Note 5**)
		The pull-up function is not selected.	(**Note 4**)
		The key-on wakeup function is not selected.	(**Note 6**)
$P1_0$–$P1_3$	Open.	The key-on wakeup function is not selected.	(**Note 7**)
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.	(**Note 5**)
		The pull-up function is not selected.	(**Note 4**)
		The key-on wakeup function is not selected.	(**Note 7**)
$P2_0$/S_{CK}	Open.	S_{CK} pin is not selected.	
	Connect to V_{SS}.	———	
$P2_1$/S_{OUT}	Open.	———	
	Connect to V_{SS}.	———	
$P2_2$/S_{IN}	Open.	S_{IN} pin is not selected.	
	Connect to V_{SS}.	———	
$P3_0$/INT0	Open.	“0” is set to output latch.	
	Connect to Vss.	———	
$P3_1$/INT1	Open.	“0” is set to output latch.	
	Connect to Vss.	———	
$P3_2$, $P3_3$	Open.	———	
	Connect to Vss.	———	
$P4_0$/A_{IN4}–$P4_3$/	Open.	———	
A_{IN7}	Connect to Vss.	———	
$P5_0$–$P5_3$	Open.	———	
	Connect to Vss.	N-channel open-drain is selected for the output structure.	
$P6_0$/A_{IN0}–$P6_3$/	Open.	———	
A_{IN3}	Connect to Vss.	———	

Notes 1: After system is released from reset, the internal oscillation (on-chip oscillator) is selected for system clock (RG_0=0, MR_0=1).

2: When the CRCK instruction is executed, the RC oscillation circuit becomes valid. Be careful that the swich of system clock is not executed at oscillation start only by the CRCK instruction execution.
In order to start oscillation, setting the main clock f(X_{IN}) oscillation to be valid (MR_1=0) is required. (If necessary, generate the oscillation stabilizing wait time by software.)
Also, when the main clock (f(X_{IN})) is selected as system clock, set the main clock f(X_{IN}) oscillation (MR_1=0) to be valid, and select main clock f(X_{IN}) (MR_0=0). Be careful that the switch of system clock cannot be executed at the same time when main clock oscillation is started.

3: In order to use the external clock input for the main clock, select the ceramic resonance by executing the CMCK instruction at the beggining of software, and then set the main clock (f(X_{IN})) oscillation to be valid (MR_1=0). Until the main clock (f(X_{IN})) oscillation becomes valid (MR_1=0) after ceramic resonance becomes valid, X_{IN} pin is fixed to “H”. When an external clock is used, insert a 1 kΩ resistor to X_{IN} pin in series for limits of current.

4: Be sure to select the output structure of ports D_0–D_5 and the pull-up function of $P0_0$–$P0_3$ and $P1_0$–$P1_3$ with every one port. Set the corresponding bits of registers for each port.

5: Be sure to select the output structure of ports $P0_0$–$P0_3$ and $P1_0$–$P1_3$ with every two ports. If only one of the two pins is used, leave another one open.

6: The key-on wakeup function is selected with every two bits. When only one of key-on wakeup function is used, considering that the value of key-on wake-up control register K1, set the unused 1-bit to “H” input (turn pull-up transistor ON and open) or “L” input (connect to V_{SS}, or open and set the output latch to “0”).

7: The key-on wakeup function is selected with every two bits. When one of key-on wakeup function is used, turn pull-up transistor of unused one ON and open.

(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.

3.3.4 Notes on interrupt

(1) Setting of INT0 interrupt valid waveform

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.

Depending on the input state of $P3_0$/INT0 pin, the external interrupt request flag (EXF0) may be set to "1" when the bit 2 of register I1 is changed.

(2) Setting of INT0 pin input control

Set a value to the bit 3 of register I1, and execute the **SNZ0** instruction to clear the EXF0 flag to "0" after executing at least one instruction.

Depending on the input state of $P3_0$/INT0 pin, the external interrupt request flag (EXF0) may be set to "1" when the bit 3 of register I1 is changed.

(3) Setting of INT1 interrupt valid waveform

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.

Depending on the input state of $P3_1$/INT1 pin, the external interrupt request flag (EXF1) may be set to "1" when the bit 2 of register I2 is changed.

(4) Setting of INT1 pin input control

Set a value to the bit 3 of register I2, and execute the **SNZ1** instruction to clear the EXF1 flag to "0" after executing at least one instruction.

Depending on the input state of $P3_1$/INT1 pin, the external interrupt request flag (EXF1) may be set to "1" when the bit 3 of register I2 is changed.

(5) Multiple interrupts

Multiple interrupts cannot be used in the 4519 Group.

(6) 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.

(7) $P3_0$/INT0 pin

When the external interrupt input pin INT0 is used, set the bit 3 of register I1 to "1".

Even in this case, port $P3_0$ I/O function is valid.

Also, the EXF0 flag is set to "1" when bit 3 of register I1 is set to "1" by input of a valid waveform (valid waveform causing external 0 interrupt) even if it is used as an I/O port $P3_0$.

The input threshold characteristics (V_{IH}/V_{IL}) are different between INT0 pin input and port $P3_0$ input. Accordingly, note this difference when INT0 pin input and port $P3_0$ input are used at the same time.

(8) $P3_1$/INT1 pin

When the external interrupt input pin INT1 is used, set the bit 3 of register I2 to "1".

Even in this case, port $P3_1$ I/O function is valid.

Also, the EXF1 flag is set to "1" when bit 3 of register I2 is set to "1" by input of a valid waveform (valid waveform causing external 1 interrupt) even if it is used as an I/O port $P3_1$.

The input threshold characteristics (V_{IH}/V_{IL}) are different between INT1 pin input and port $P3_1$ input. Accordingly, note this difference when INT1 pin input and port $P3_1$ input are used at the same time.

(9) POF instruction

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

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

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

3.3.5 Notes on timer

(1) Prescaler

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

(2) Count source

Stop timer 1, 2, 3, 4 or LC 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 the timer

Stop timer 1, 2, 3, 4 or LC counting and then execute the **T1AB**, **T2AB**, **T3AB**, **T4AB** or **TLCA** instruction to write its data.

(5) Writing to reload register R1, reload register R3 and reload register R4H

When writing data to reload register R1 while timer 1 is operating respectively, avoid a timing when timer 1 underflows.
When writing data to reload register R3 while timer 3 is operating respectively, avoid a timing when timer 3 underflows.
When writing data to reload register R4H while timer 4 is operating respectively, avoid a timing when timer 4 underflows.

(6) Timer 4

- At CNTR1 output vaild, if a timing of timer 4 underflow overlaps with a timing to stop timer 4, a hazard may be generated in a CNTR1 output waveform. Please review sufficiently.
- When "H" interval extension function of the PWM signal is set to be "valid", set "01_{16}" or more to reload register R4H.

(7) Watchdog timer

- The watchdog timer function is valid after system is released from reset. When not using the watchdog timer function, stop the watchdog timer function and execute the **DWDT** instruction, the **WRST** instruction continuously, and clear the WEF flag to "0".
- The watchdog timer function is valid after system is returned from the RAM back-up state. When not using the watchdog timer function, stop the watchdog timer function and execute the **DWDT** instruction and the **WRST** instruction continuously every system is returned from the RAM back-up state.
- When the watchdog timer function and RAM back-up function are used at the same time, initialize the flag WDF1 with the **WRST** instruction before system enters into the RAM back-up state.

(8) Pulse width input to CNTR0 pin, CNTR1 pin

Refer to section "3.1 Electrical characteristics" for rating value of pulse width input to CNTR0 pin, CNTR1 pin.

(9) Period measurement circuit

- When a period measurement circuit is used, clear bit 0 of register I1 to "0", and set a timer 1 count start synchronous circuit to be "not selected".

- While a period measurement circuit is operating, the timer 1 interrupt request flag (T1F) is not set by the timer 1 underflow signal, it is the flag for detecting the completion of period measurement.

- When a period measurement circuit is used, select the sufficiently higher-speed frequency than the signal for measurement for the count source of a timer 1.

- When the signal for period measurement is D_6/CNTR0 pin input, do not select D_6/CNTR0 pin input as timer 1 count source.
  (The X_{IN} input is recommended as timer 1 count source at the time of period measurement circuit use.)

- When the input of $P3_0$/INT0 pin is selected for measurement, set the bit 3 of a register I1 to "1", and set the input of INT0 pin to be enabled.

- Start timer operation immediately after operation of a period measurement circuit is started.

- Even when the edge for measurement is input by timer operation is started from the operation of period measurement circuit is started, timer 1 is not operated.

- When data is read from timer 1, stop the timer 1 and the period measurement circuit, and then execute the data read instruction. Depending on the state of timer 1, the timer 1 interrupt request flag (T1F) may be set to "1" when the period measurement circuit is stopped by clearing bit 2 of register W5 to "0". In order to avoid the occurrence of an unexpected interrupt, disable the timer 1 interrupt, and then, stop the period measurement circuit. Figure 3.3.1 shows the setting example to read measurement data of period measurement circuit.

(10) Prescaler, timer 1, timer 2 and timer 3 count start time and count time when operation starts

Count starts from the first rising edge of the count source ② in Fig. 3.3.2 after prescaler, timer 1, timer 2 and timer 3 operations start ① in Fig. 3.3.2.

Time to first underflow ③ in Fig. 3.3.2 is shorter (for up to 1 period of the count source) than time among next underflow ④ in Fig. 3.3.2 by the timing to start the timer and count source operations after count starts.

(11) Timer 4 count start time and count time when operation starts

Count starts from the rising edge ② in Fig. 3.3.3 after the first falling edge of the count source, after timer 4 operation starts ① in Fig. 3.3.3.

Time to first underflow ③ in Fig. 3.3.3 is different from time among next underflow ④ in Fig. 3.3.3 by the timing to start the timer and count source operations after count starts.

```
⋮
Timer 1 operation is stopped
(bit 2 of register W1 is cleared to "0")
↓
Timer 1 interrupt is disabled
(bit 2 of register V1 is cleared to "0")
↓
Period measurement circuit is stopped.
(bit 2 of register W5 is cleared to "0")
↓
Execute at least one Instruction [NOP]
↓
Timer 1 interrupt request flag (T1F)
is cleared to "0".  [SNZT1]
↓
Considering the skip of the SNZT1 instruction,
insert the NOP instruction.
↓
Measurement data is read. [TAB1]
⋮
```

Fig. 3.3.1 Period measurement circuit program example

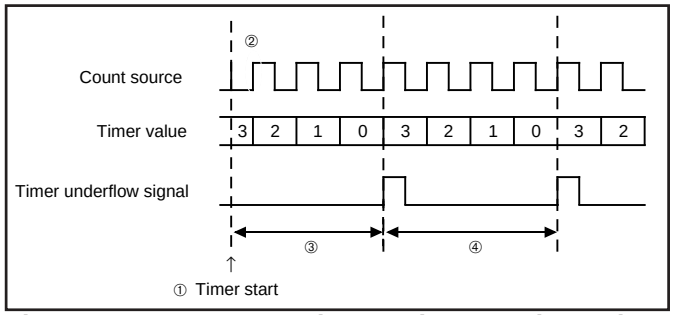


Fig. 3.3.2 Count start time and count time when operation starts (PS, T1, T2 and T3)

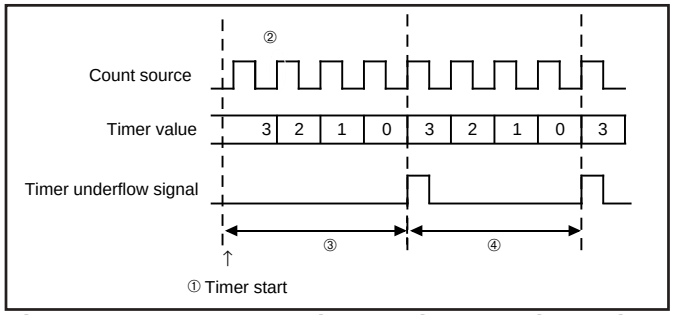


Fig. 3.3.3 Count start time and count time when operation starts (T4)

3.3.6 Notes on A/D conversion

(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 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 3.3.4 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 3.3.5. In addition, test the application products sufficiently.

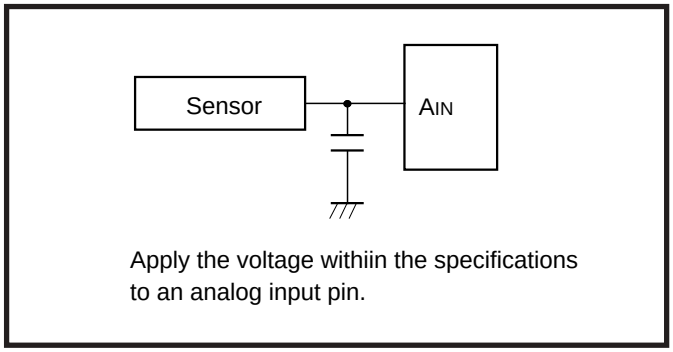


Fig. 3.3.4 Analog input external circuit example-1

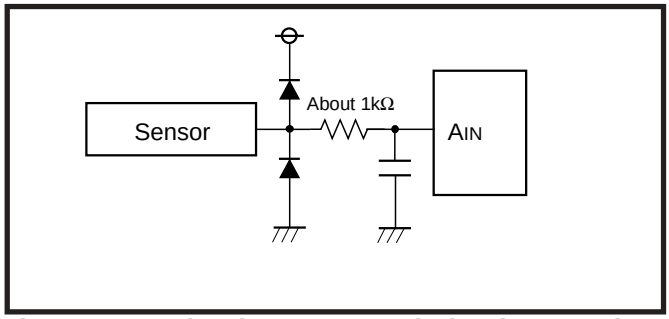


Fig. 3.3.5 Analog input external circuit example-2

(3) Notes for the use of A/D conversion 2

Do not change the operating mode of the A/D converter by bit 3 of register Q1 during A/D conversion (A/D conversion mode and comparator mode).

(4) Notes for the use of A/D conversion 3

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 Q1 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 (refer to Figure 3.3.6①).
- 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 bit 3 of register Q1, and execute the **SNZAD** instruction to clear the ADF flag to “0”.

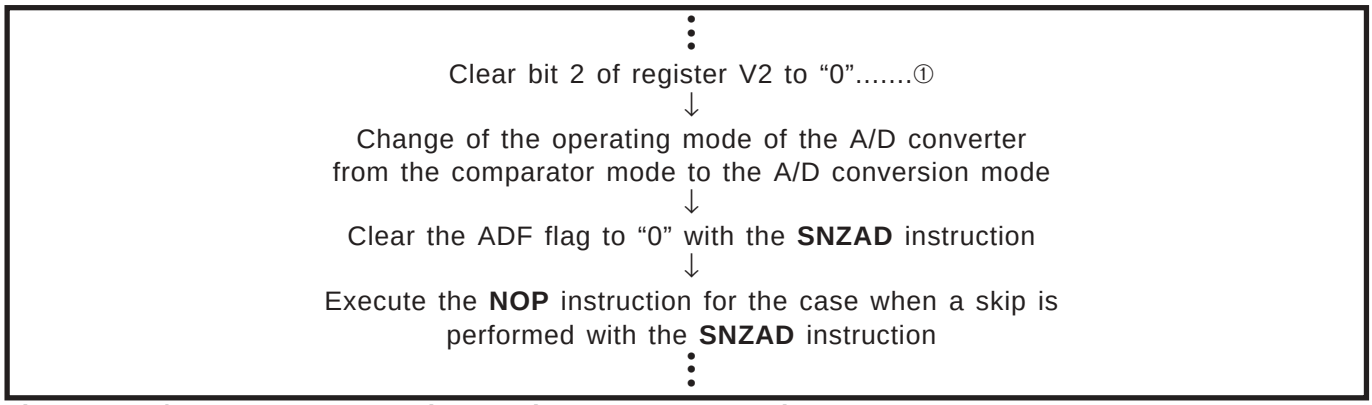


Fig. 3.3.6 A/D converter operating mode program example

(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 2 machine cycles + A/D conversion clock (ADCK) 1 clock.

(6) Analog input pins

When $P4_0/AIN_4$–$P4_3/AIN_7$, $P6_0/AIN_0$–$P6_3/AIN_3$ are set to pins for analog input, they cannot be used as I/O ports P4 and P6.

(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

As for the supply voltage when A/D converter is used and the recommended operating condition of the A/D convesion clock frequency, refer to the "3.1 Electrical characteristics".

3.3.7 Notes on serial I/O

(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 transmit/receive is continued as long as an external clock is input. If an external clock is input 9 times or more and serial transmit/receive 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 to "1" when a clock is counted 8 times.
- Be sure to set the initial input level on the external clock pin to "H" level.
- Refer to section "3.1 Electrical characteristics" when using serial I/O with an external clock.

3.3.8 Notes on reset

(1) Register initial value

The initial value of the following registers are undefined after system is released from reset. After system is released from reset, set initial values.

- Register Z (2 bits)
- Register D (3 bits)
- Register E (8 bits)

(2) Power-on reset

When the built-in power-on reset circuit is used, the time for the supply voltage to rise from 0 V to the minimum rating value of the recommended operating conditions must be set to 100 µs or less. If the rising time exceeds 100 µs, connect a capacitor between the $\overline{RESET}$ pin and V_{SS} at the shortest distance, and input "L" level to $\overline{RESET}$ pin until the value of supply voltage reaches the minimum rating value of the recommended operating conditions.
Refer to section "**3.1 Electrical characteristics**" for the reset voltage of the recommended operating conditions.

3.3.9 Notes on RAM back-up

(1) 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 and the **POF** instruction.

(2) Key-on wakeup function

After checking none of the return condition for ports (P0, P1, INT0 and INT1 specified with register K0–K2) with valid key-on wakeup function is satisfied, execute the **POF** instruction.

If at least one of return condition for ports with valid key-on wakeup function is satisfied, system returns from the RAM back-upn state immediately after the **POF** instruction is executed.

(3) Return from RAM back-up mode

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

The initial value of the following registers are undefined at RAM back-up. After system is returned from RAM back-up mode, set initial values.

- Register Z (2 bits)
- Register X (4 bits)
- Register Y (4 bits)
- Register D (3 bits)
- Register E (8 bits)

(4) Watchdog timer

- The watchdog timer function is valid after system is returned from the RAM back-up state. When not using the watchdog timer function, stop the watchdog timer function with the **DWDT** instruction and the **WRST** instruction continuously every system is returned from the RAM back-up.
- When the watchdog timer function and RAM back-up function are used at the same time, initialize the flag WDF1 with the **WRST** instruction before system goes into the RAM back-up state.

(5) Port $P3_0$/INT0 pin

When the RAM back-up mode is used by clearing the bit 3 of register I1 to “0” and setting the input of INT0 pin to be disabled, be careful about the following note.

- When the input of INT0 pin is disabled (register $I1_3$ = “0”), clear bit 0 of register K2 to “0” to invalidate the key-on wakeup before system goes into the RAM back-up mode.

(6) Port $P3_1$/INT1 pin

When the RAM back-up mode is used by clearing the bit 3 of register I2 to “0” and setting the input of INT1 pin to be disabled, be careful about the following note.

- When the input of INT1 pin is disabled (register $I2_3$ = “0”), clear bit 2 of register K2 to “0” to invalidate the key-on wakeup before system goes into the RAM back-up mode.

3.3.10 Notes on clock control

(1) Clock control

Execute the main clock ($f(X_{IN})$) selection instruction (**CMCK**, **CRCK** or **CYCK** instruction) in the initial setting routine of program (executing it in address 0 in page 0 is recommended).

The oscillation circuit by the **CMCK**, **CRCK** or **CYCK** instruction can be selected only at once. The oscillation circuit corresponding to the first executed one of these instructions is valid.

The **CMCK**, **CRCK** or **CYCK** instructions can be used only to select main clock ($f(X_{IN})$). In this time, the start of oscillation and the switch of system clock are not performed.

When the **CMCK**, **CRCK** or **CYCK** instructions are never executed, main clock ($f(X_{IN})$) cannot be used and system can be operated only by on-chip oscillator.

The no operated clock source (f(RING)) or ($f(X_{IN})$) cannot be used for the system clock. Also, the clock source (f(RING) or $f(X_{IN})$) selected for the system clock cannot be stopped.

(2) On-chip oscillator

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

Be careful that margin of frequencies when designing application products.

When considering the oscillation stabilize wait time at the switch of clock, be careful that the margin of frequencies of the on-chip oscillator clock.

(3) External clock

When the external clock signal for the main clock ($f(X_{IN})$) is used, connect the clock source to X_{IN} pin and X_{OUT} pin open. In program, after the CMCK instruction is executed, set main clock ($f(X_{IN})$) oscillation start to be enabled (MR_1=0).

For this product, when RAM back-up mode and main clock ($f(X_{IN})$) stop (MR_1=1), X_{IN} pin is fixed to “H” in order to avoid the through current by floating of internal logic. The X_{IN} pin is fixed to “H” until main clock ($f(X_{IN})$) oscillation start to be valid (MR_1=0) by the CMCK instruction from reset state. Accordingly, when an external clock is used, connect a 1 kΩ or more resistor to X_{IN} pin in series to limit of current by competitive signal.

(4) 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.

3.3.11 Electric characteristic differences between Mask ROM and One Time PROM version MCU

There are differences in electric characteristics, operation margin, noise immunity, and noise radiation between Mask ROM and One Time PROM version MCUs due to the difference in the manufacturing processes.

When manufacturing an application system with the One time PROM version and then switching to use of the Mask ROM version, please perform sufficient evaluations for the commercial samples of the Mask ROM version.

3.3.12 Note on Power Source Voltage

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

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

3.4 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.

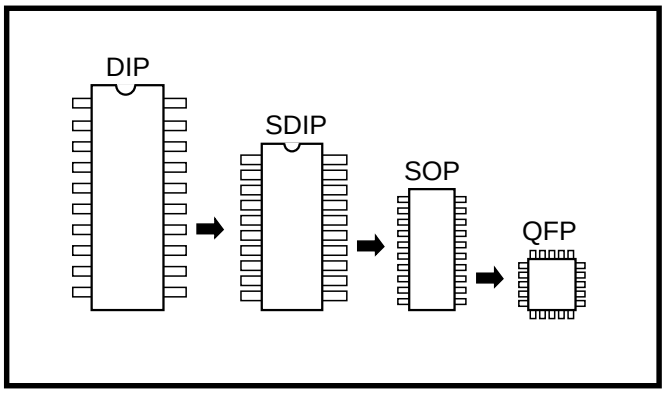


Fig. 3.4.1 Selection of packages

(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.

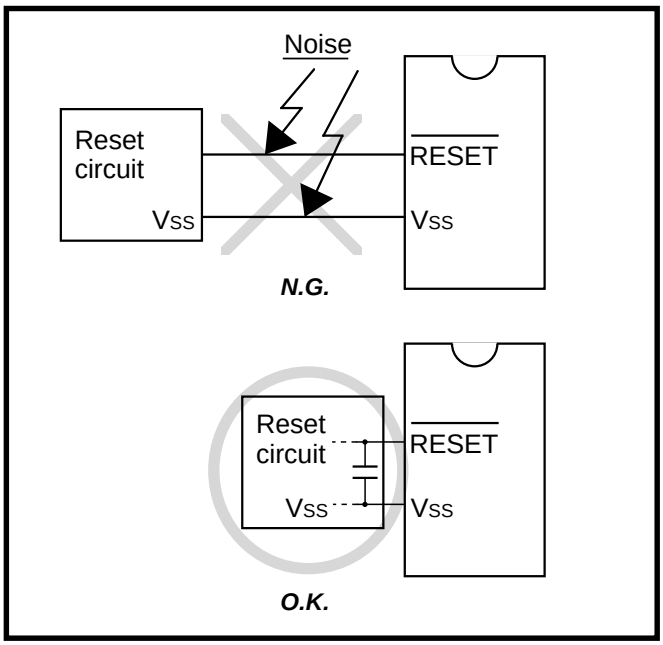


Fig. 3.4.2 Wiring for the $\overline{\text{RESET}}$ input pin

(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.

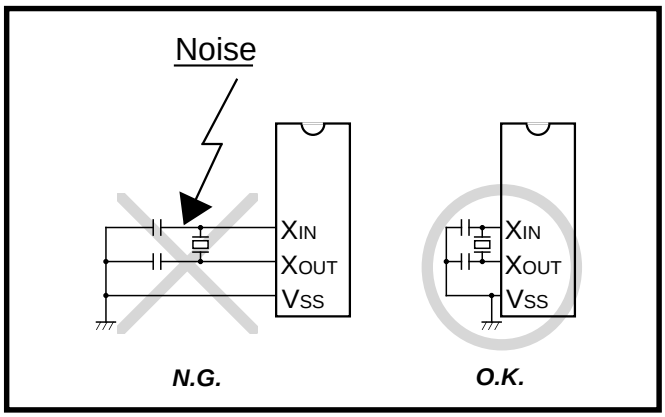


Fig. 3.4.3 Wiring for clock I/O pins

● **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.

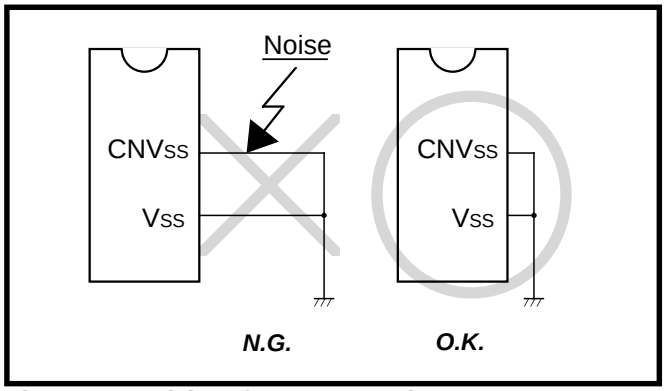


Fig. 3.4.4 Wiring for CNVss pin

(5) Wiring to V_{PP} pin of built-in PROM version

In the built-in PROM version of the 4524 Group, the CNV_{SS} pin is also used as the built-in PROM power supply input pin V_{PP}.

● **When the V_{PP} pin is also used as the CNV_{SS} pin**

Connect an approximately 5 kΩ resistor to the V_{PP} pin the shortest possible in series and also to the V_{SS} pin. When not connecting the resistor, make the length of wiring between the V_{PP} pin and the V_{SS} 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 V_{PP} pin of the built-in PROM version is the power source input pin for the built-in PROM. When programming in the built-in PROM, the impedance of the V_{PP} 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 V_{PP} pin, abnormal instruction codes or data are read from the built-in PROM, which may cause a program runaway.

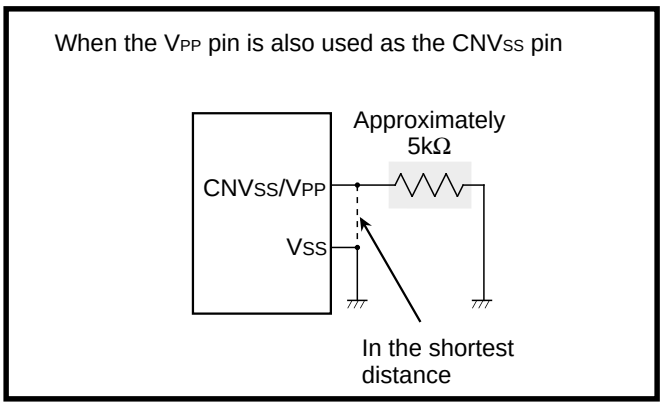


Fig. 3.4.5 Wiring for the V_{PP} pin of the built-in PROM version

3.4.2 Connection of bypass capacitor across V_{SS} line and V_{DD} line

Connect an approximately 0.1 μF bypass capacitor across the V_{SS} line and the V_{DD} line as follows:

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

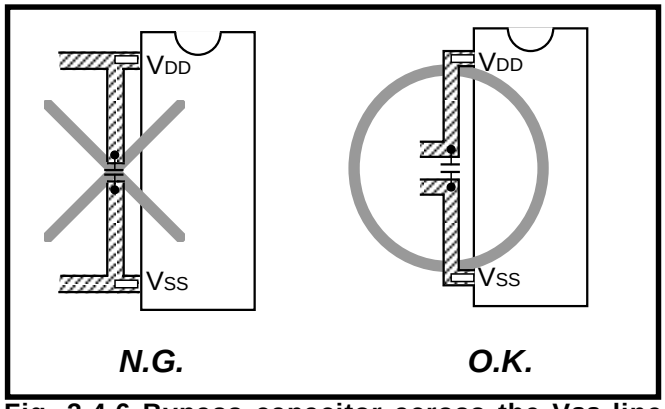


Fig. 3.4.6 Bypass capacitor across the V_{SS} line and the V_{DD} line

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.

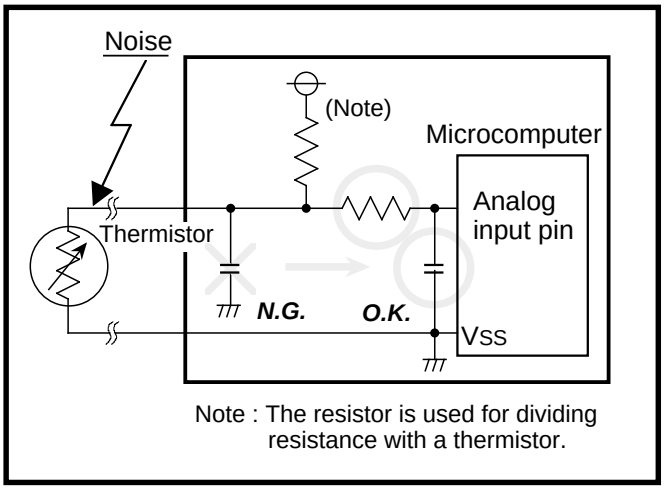


Fig. 3.4.7 Analog signal line and a resistor and a capacitor

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.

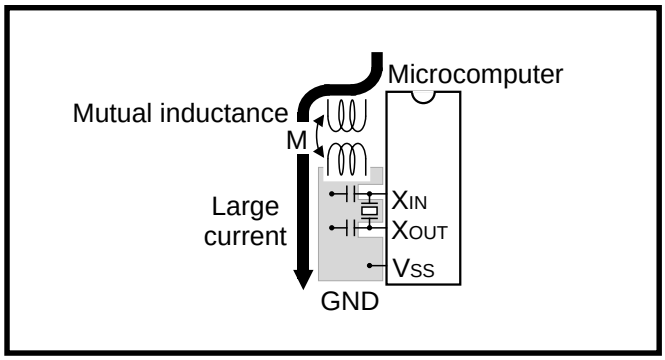


Fig. 3.4.8 Wiring for a large current signal line

(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.

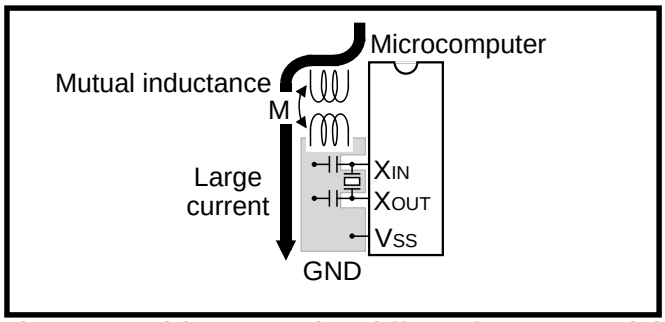


Fig. 3.4.9 Wiring to a signal line where potential levels change frequently

(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.

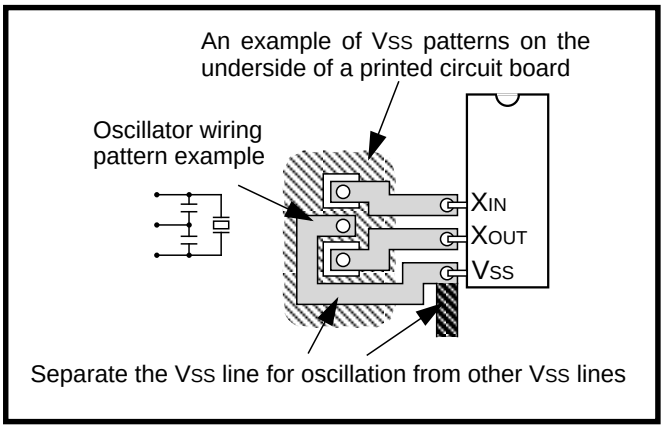


Fig. 3.4.10 Vss pattern on the underside of an oscillator

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.

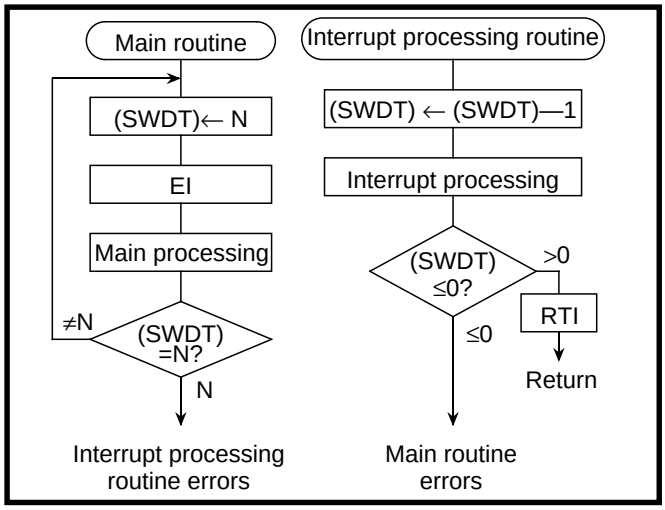


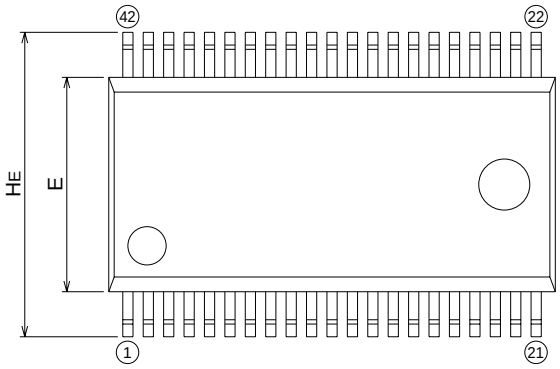
Fig. 3.4.11 Watchdog timer by software

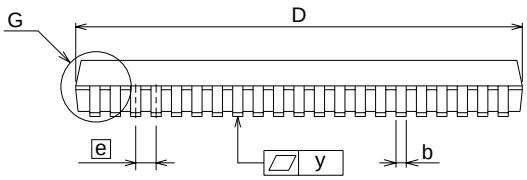
3.5 Package outline

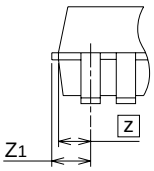
42P2R-A Recommended

Plastic 42pin 450mil SSOP

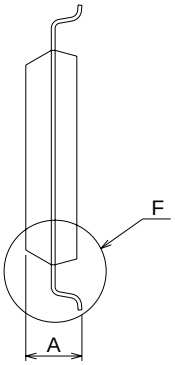
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SSOP42-P-450-0.80	–	0.63	Alloy 42/Cu Alloy

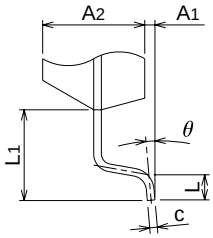




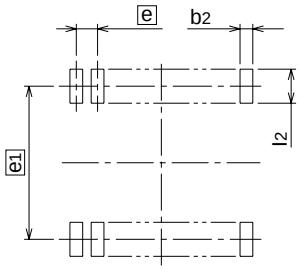


Detail G





Detail F



Recommended Mount Pad

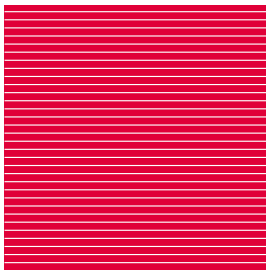
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	2.4
A1	0.05	–	–
A2	–	2.0	–
b	0.35	0.4	0.5
c	0.13	0.15	0.2
D	17.3	17.5	17.7
E	8.2	8.4	8.6
e	–	0.8	–
HE	11.63	11.93	12.23
L	0.3	0.5	0.7
L1	–	1.765	–
z	–	0.75	–
Z1	–	–	0.9
y	–	–	0.15
θ	0°	–	10°
b2	–	0.5	–
e1	–	11.43	–
l2	1.27	–	–

RENESAS 4-BIT CISC SINGLE-CHIP MICROCOMPUTER
USER'S MANUAL
4519 Group

Publication Data : Rev.1.00 Aug 08, 2004

Published by : Sales Strategic Planning Div.
Renesas Technology Corp.

4519 Group
User's Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan