To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

“Standard”:	Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
“High Quality”:	Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
“Specific”:	Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

4519 Group

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

REJ03B0007-0301
Rev.3.01
2005.06.15

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 interface, 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 interface .... 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)

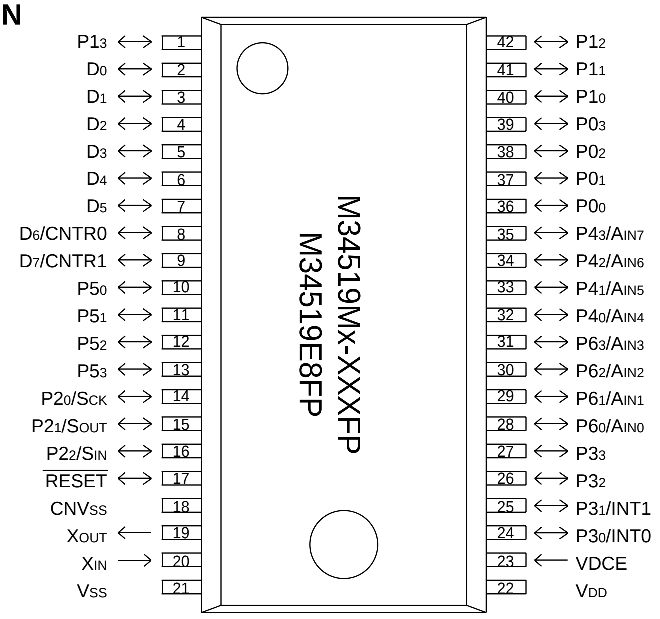
APPLICATION

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

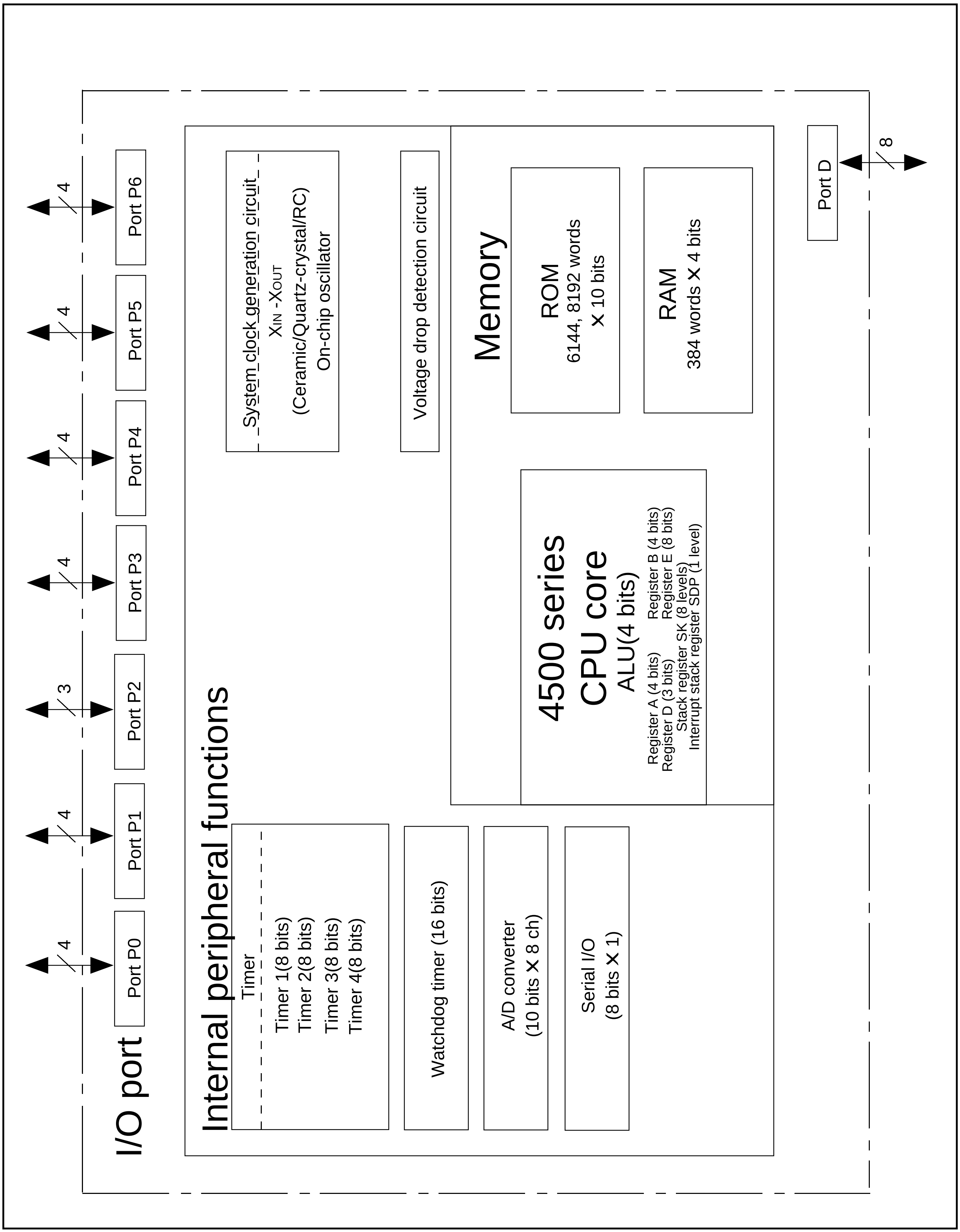
Note: Shipped in blank.

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 X_{IN} 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	D_0–D_7	I/O (Input is examined by skip decision)	Eight independent I/O ports; Ports D_6 and D_7 are also used as CNTR0 and CNTR1, respectively. The output structure is switched by software.
	$P0_0$–$P0_3$	I/O	4-bit I/O port; a pull-up function, a key-on wakeup function and output structure can be switched by software.
	$P1_0$–$P1_3$	I/O	4-bit I/O port; a pull-up function, a key-on wakeup function and output structure can be switched by software.
	$P2_0$–$P2_2$	I/O	3-bit I/O port; ports $P2_0$, $P2_1$ and $P2_2$ are also used as S_{CK}, S_{OUT} and S_{IN}, respectively.
	$P3_0$–$P3_3$	I/O	4-bit I/O port ; ports $P3_0$ and $P3_1$ are also used as INT0 and INT1, respectively.
	$P4_0$–$P4_3$	I/O	4-bit I/O port ; ports $P4_0$–$P4_3$ are also used as A_{IN4}–A_{IN7}, respectively.
	$P5_0$–$P5_3$	I/O	4-bit I/O port ; the output structure is switched by software.
	$P6_0$–$P6_3$	I/O	4-bit I/O port ; ports $P6_0$–$P6_3$ are also used as A_{IN0}–A_{IN3}, 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, V_{DD}=5V, f(X_{IN})=6 MHz, f(STCK)=f(X_{IN}), on-chip oscillator stop)
			70 µA (Ta=25 °C, V_{DD}=5V, f(X_{IN})=32 kHz, f(STCK)=f(X_{IN}), on-chip oscillator stop)
			150 µA (Ta=25 °C, V_{DD}=5V, on-chip oscillator is used, f(STCK)=f(RING), f(X_{IN}) stop)
	RAM back-up mode		0.1 µA (Ta=25 °C, V_{DD} = 5 V, output transistors in the cut-off state)

PIN DESCRIPTION

Pin	Name	Input/Output	Function
VDD	Power supply	—	Connected to a plus power supply.
VSS	Ground	—	Connected to a 0 V power supply.
CNVSS	CNVSS	—	Connect CNVSS to VSS and apply “L” (0V) to CNVSS certainly.
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.
$\overline{\text{RESET}}$	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 $\overline{\text{RESET}}$ pin outputs “L” level.
XIN	Main clock input	Input	I/O pins of the main clock generating circuit. When using a ceramic resonator, connect it between pins XIN and XOUT. When using a 32 kHz quartz-crystal oscillator, connect it between pins XIN and XOUT. A feedback resistor is built-in between them. When using the RC oscillation, connect a resistor and a capacitor to XIN, and leave XOUT pin open.
XOUT	Main clock output	Output	
D0–D7	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 D6, D7 is also used as CNTR0 pin and CNTR1 pin, respectively.
P00–P03	I/O port P0	I/O	Port P0 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to “1” and select the N-channel open-drain. Port P0 has a key-on wakeup function and a pull-up function. Both functions can be switched by software.
P10–P13	I/O port P1	I/O	Port P1 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to “1” and select the N-channel open-drain. Port P1 has a key-on wakeup function and a pull-up function. Both functions can be switched by software.
P20–P23	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 P20–P22 are also used as SCK, SOUT, SIN, respectively.
P30–P33	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 P30 and P31 are also used as INT0 pin and INT1 pin, respectively.
P40–P43	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 P40–P43 are also used as AIN4–AIN7, respectively.
P50–P53	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.
P60–P63	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 P60–P63 are also used as AIN0–AIN3, 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 D6 and D7, 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 P30 and P31, respectively.
AIN0–AIN7	Analog input	Input	A/D converter analog input pins. AIN0–AIN7 are also used as ports P60–P63 and P40–P43, respectively.
SCK	Serial I/O data I/O	I/O	Serial I/O data transfer synchronous clock I/O pin. SCK pin is also used as port P20..
SOUT	Serial I/O data output	Output	Serial I/O data output pin. SOUT pin is also used as port P21.
SIN	Serial I/O clock input	Input	Serial I/O data input pin. SIN pin is also used as port P22.

MULTIFUNCTION

Pin	Multifunction	Pin	Multifunction	Pin	Multifunction	Pin	Multifunction
D_6	CNTR0	CNTR0	D_6	$P6_0$	AIN0	AIN0	$P6_0$
D_7	CNTR1	CNTR1	D_7	$P6_1$	AIN1	AIN1	$P6_1$
$P2_0$	SCK	SCK	$P2_0$	$P6_2$	AIN2	AIN2	$P6_2$
$P2_1$	SOUT	SOUT	$P2_1$	$P6_3$	AIN3	AIN3	$P6_3$
$P2_2$	SIN	SIN	$P2_2$	$P4_0$	AIN4	AIN4	$P4_0$
$P3_0$	INT0	INT0	$P3_0$	$P4_1$	AIN5	AIN5	$P4_1$
$P3_1$	INT1	INT1	$P3_1$	$P4_2$	AIN6	AIN6	$P4_2$
				$P4_3$	AIN7	AIN7	$P4_3$

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.

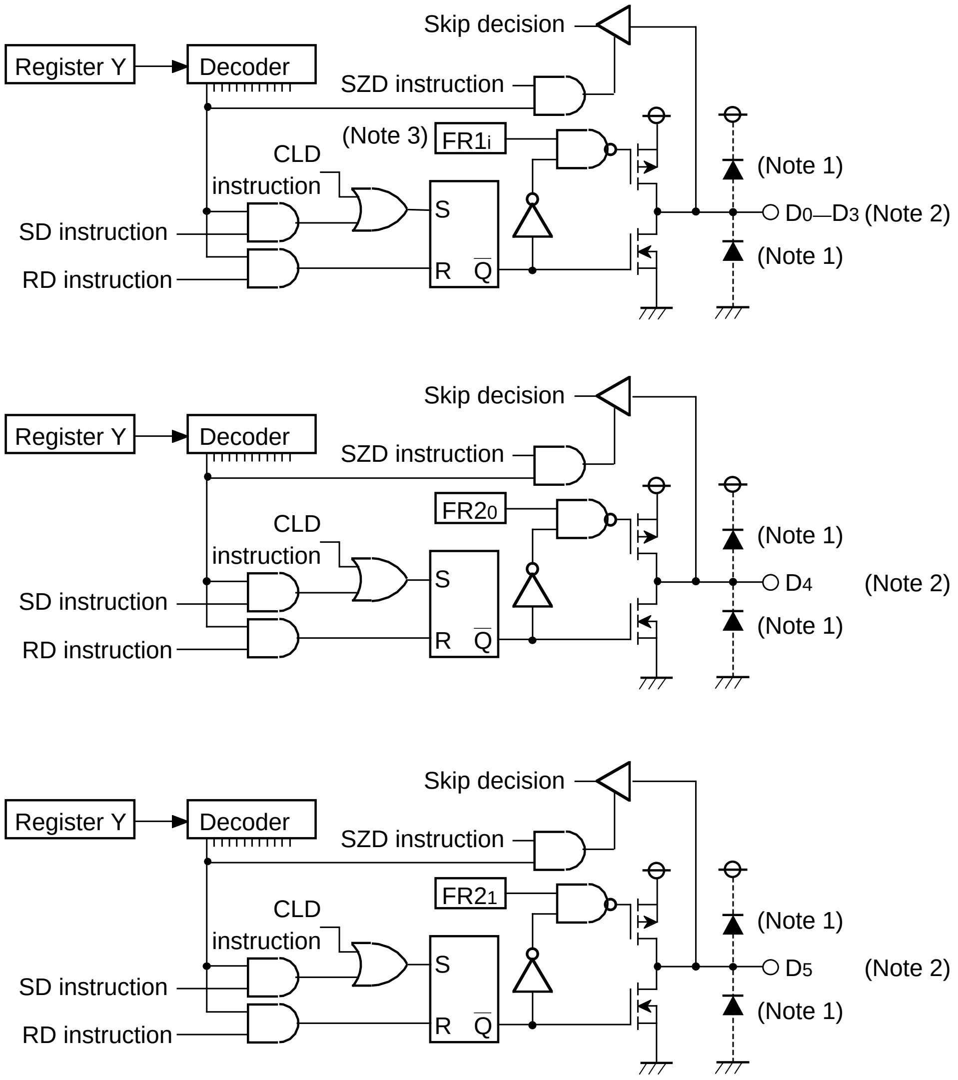
6: The key-on wakeup function is selected with every two bits. When only one of key-on wakeup function is used, considering that the value of key-on wake-up control register K1, set the unused 1-bit to “H” input (turn pull-up transistor ON and open) or “L” input (connect to V_{SS}, or open and set the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

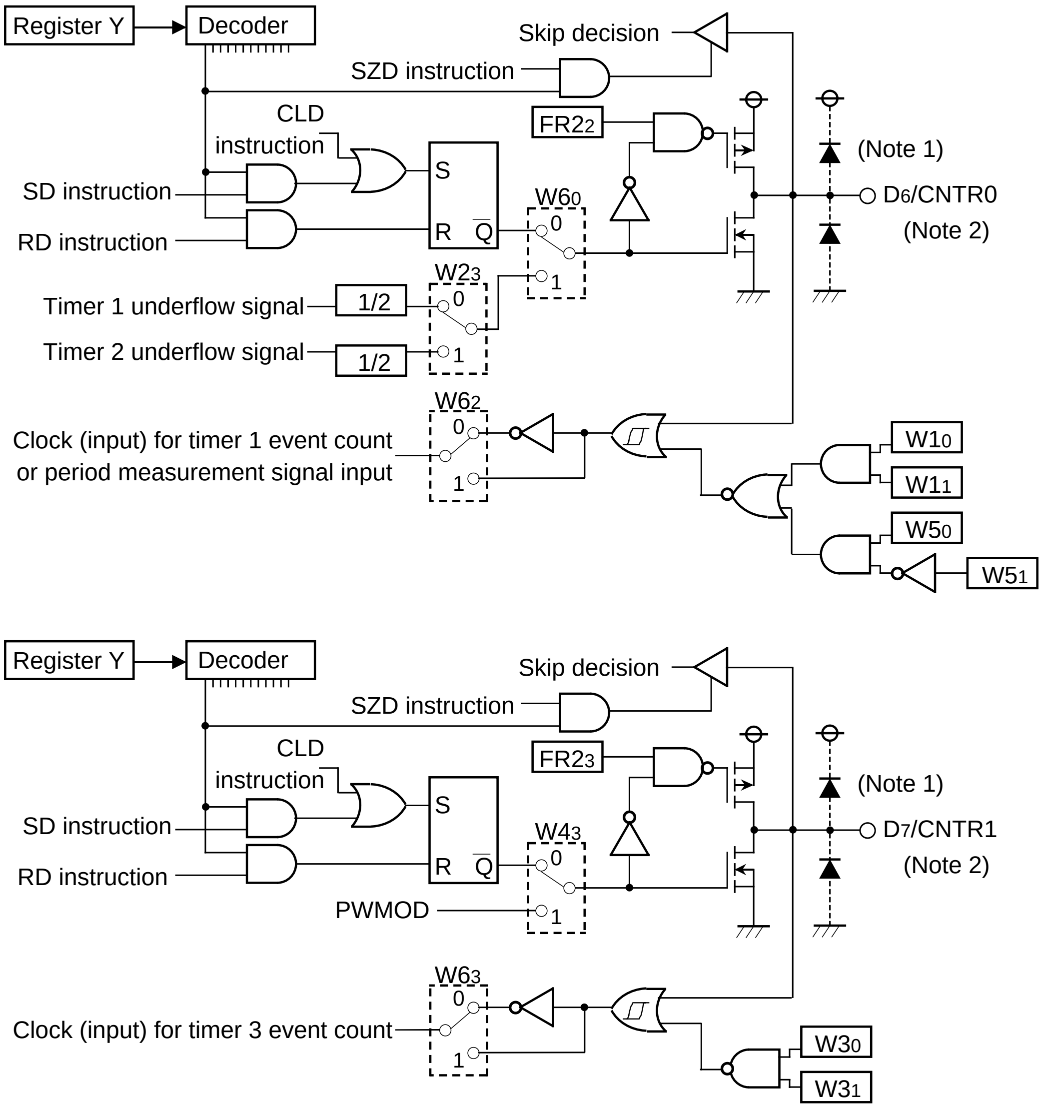
● Connect the unused pins to V_{SS} and V_{DD} using the thickest wire at the shortest distance against noise.

PORT BLOCK DIAGRAMS



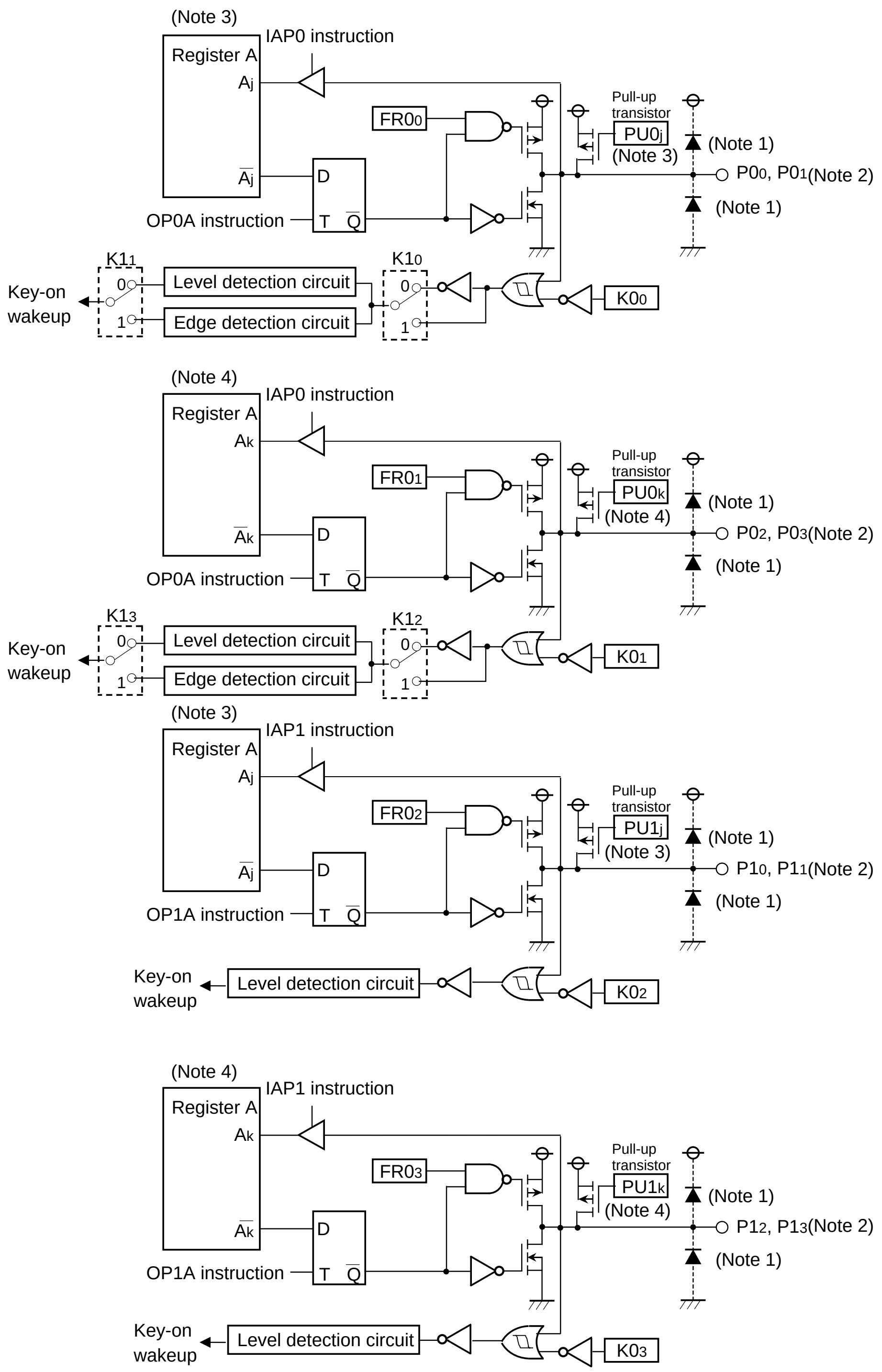
Notes 1: ----|◀---- This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be VDD or less.
3: i represents bits 0 to 3.

Port block diagram (1)



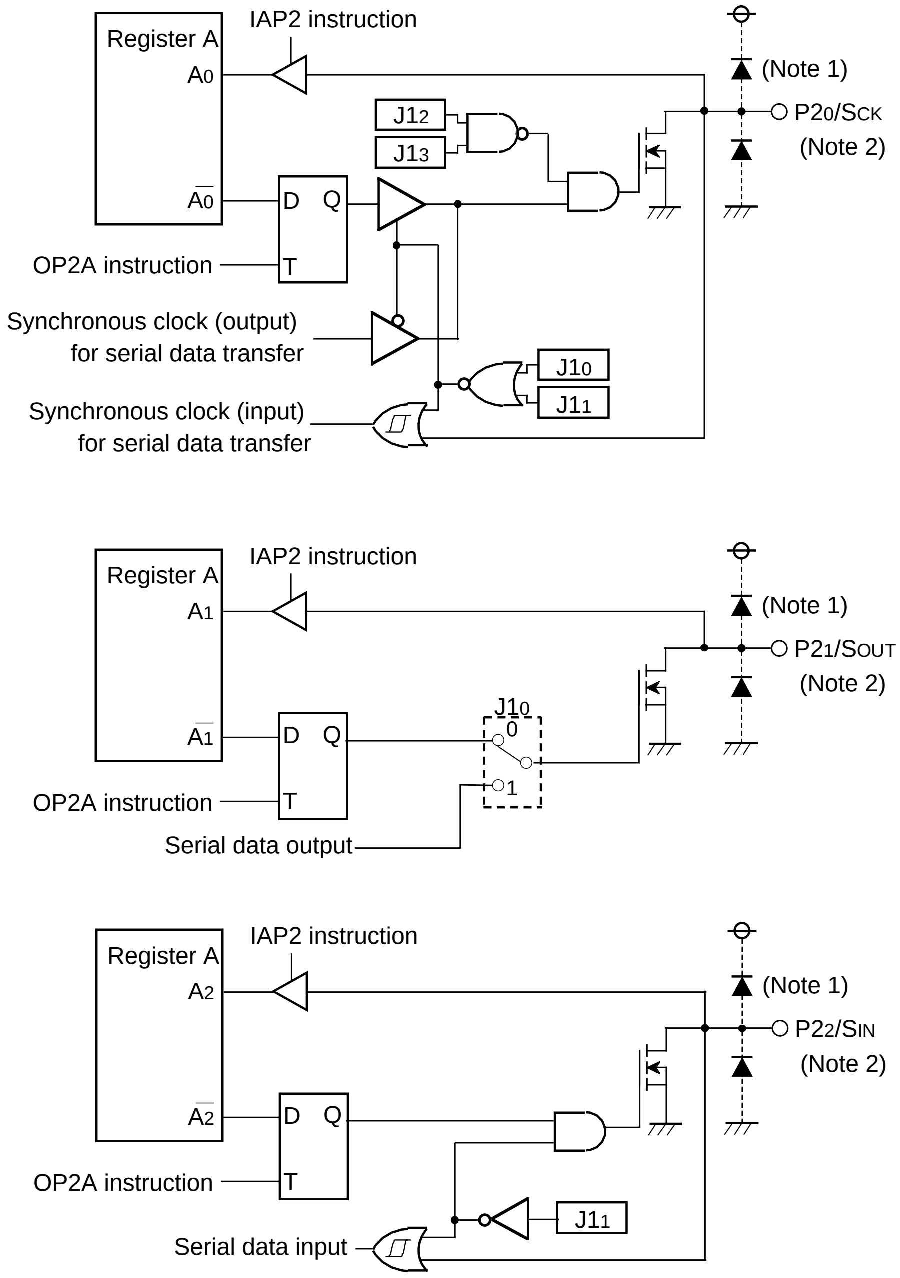
Notes 1: ----▶|---- This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be VDD or less.

Port block diagram (2)



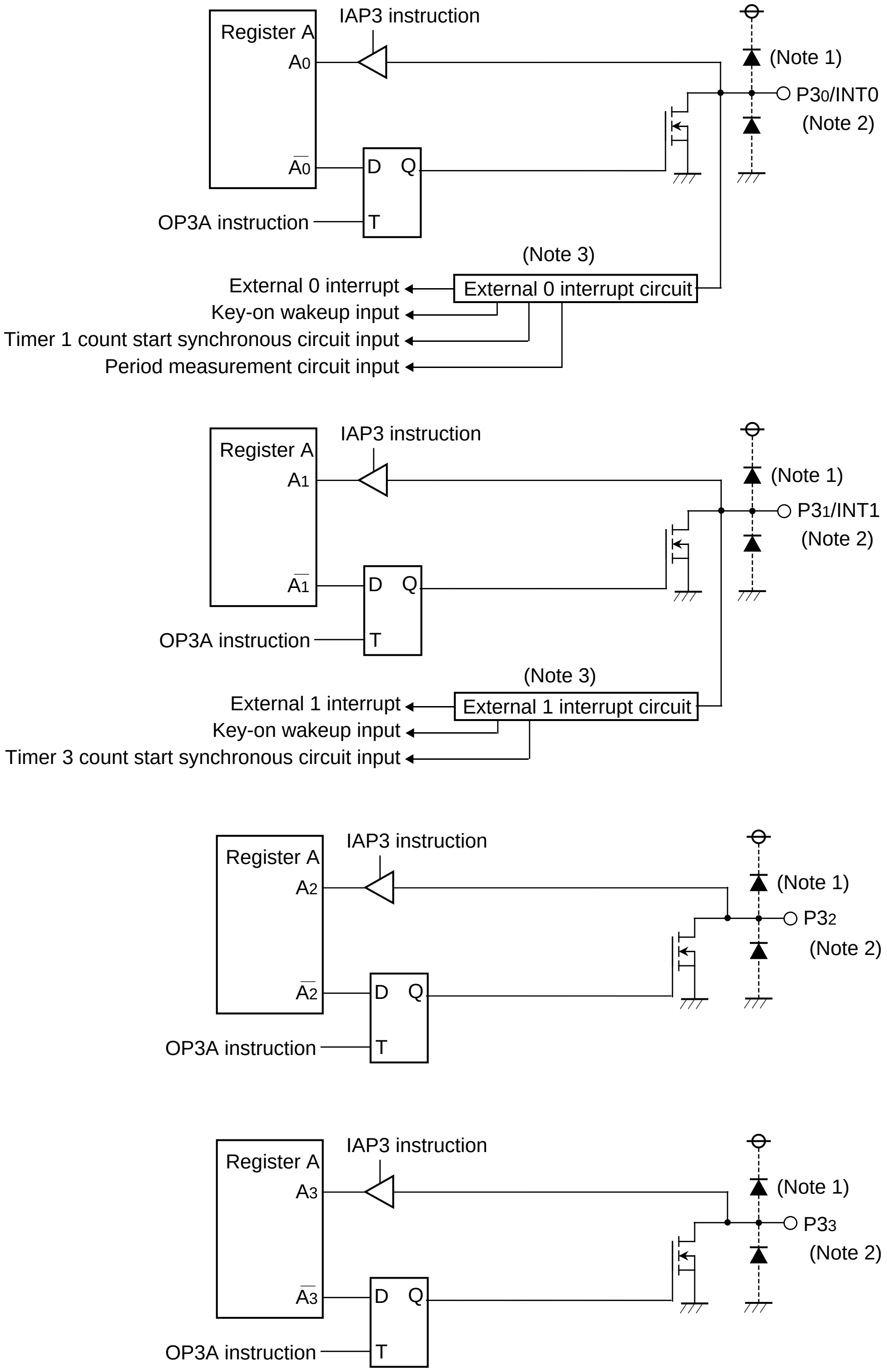
Notes 1: -----|◀----- This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be V_{DD} or less.
3: j represents bits 0 and 1.
4: k represents bits 2 and 3.

Port block diagram (3)



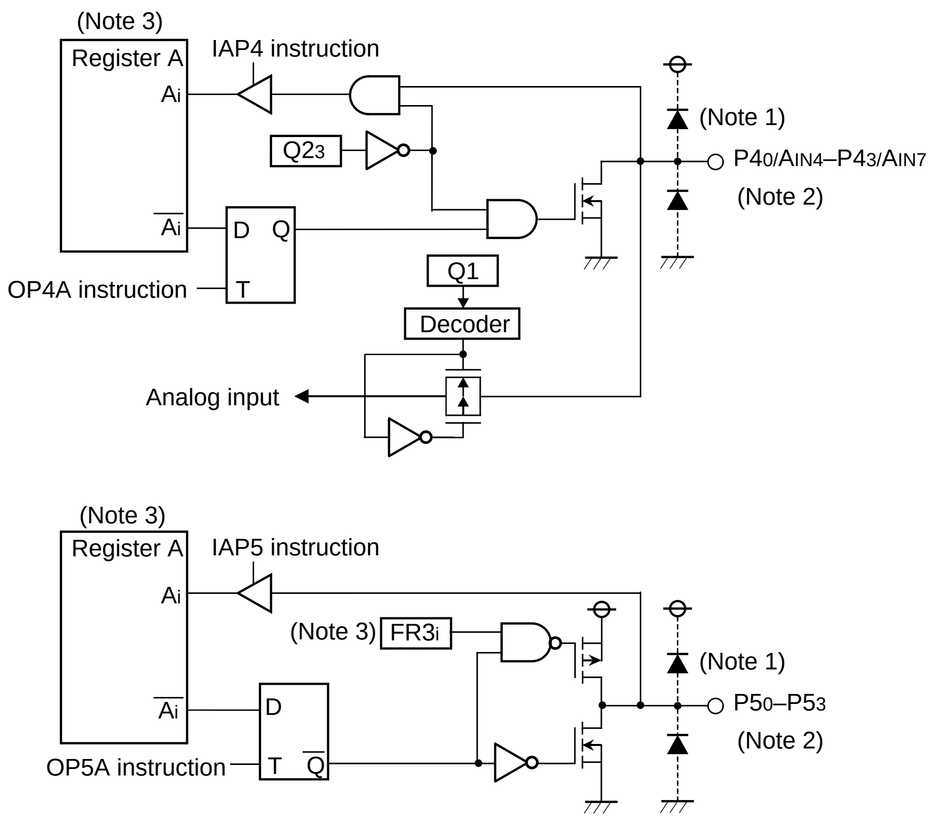
Notes 1: ----|◀---- This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be V_{DD} or less.

Port block diagram (4)



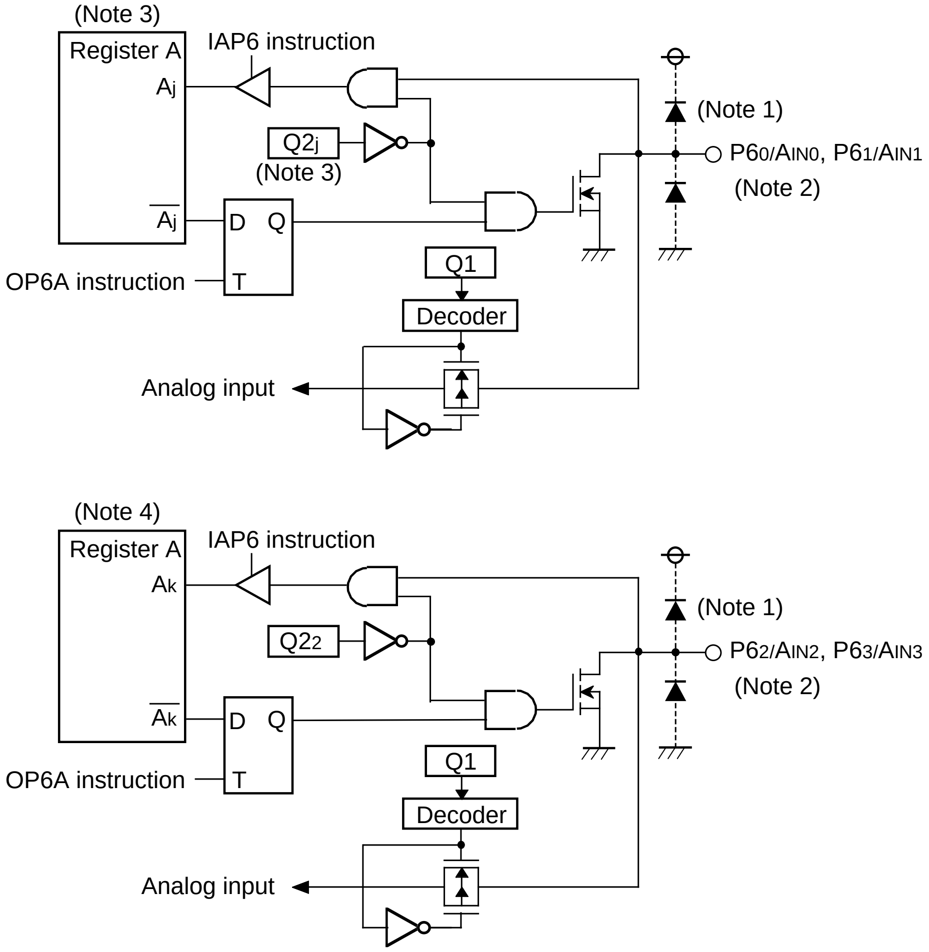
Notes 1: ----◀---- This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be V_{DD} or less.
3: As for details, refer to the external interrupt circuit structure.

Port block diagram (5)



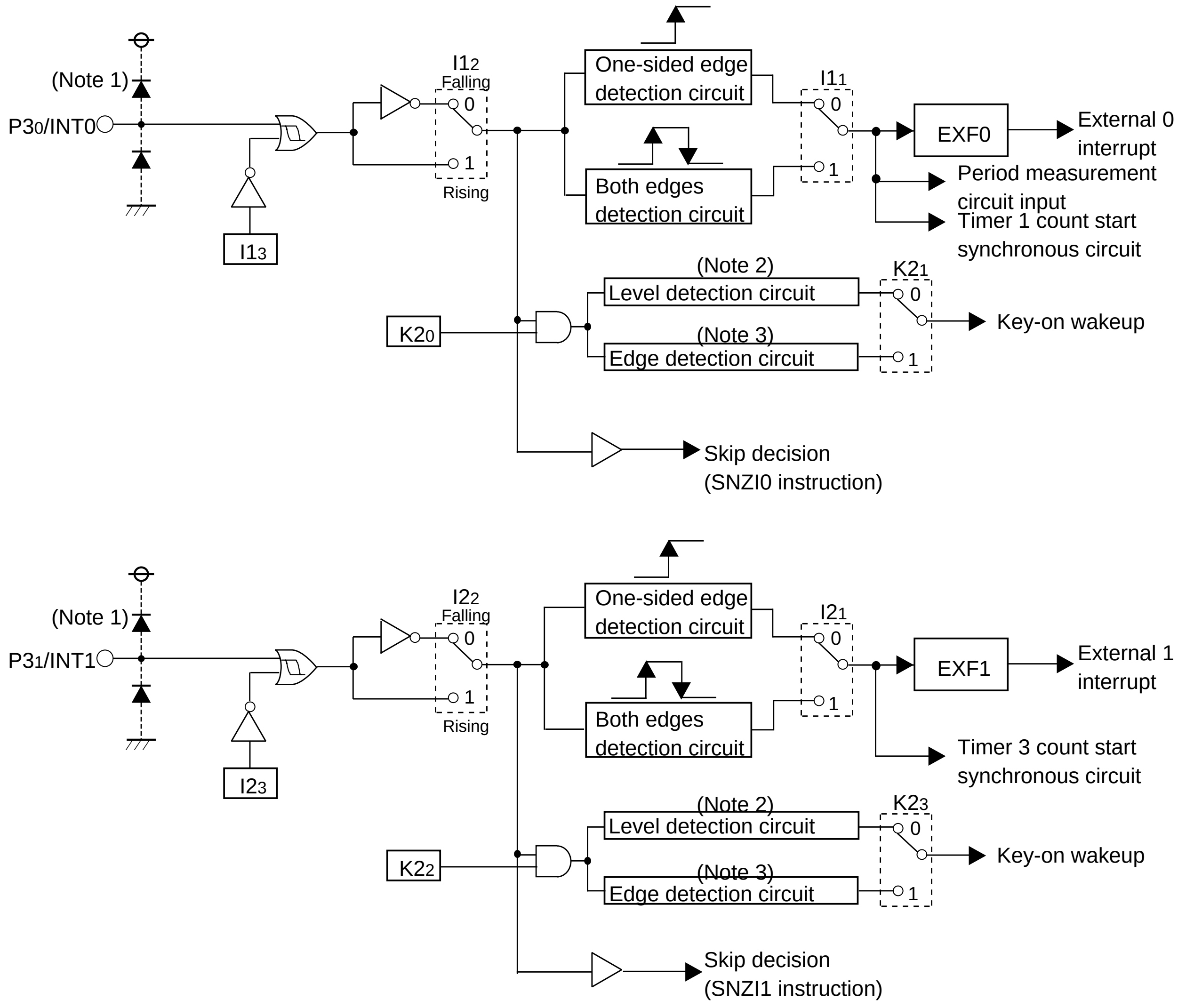
Notes 1:-----|◀----- This symbol represents a parasitic diode on the port.
2: Applied potential to these ports must be V_{DD} or less.
3: i represents bits 0 to 3.

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.

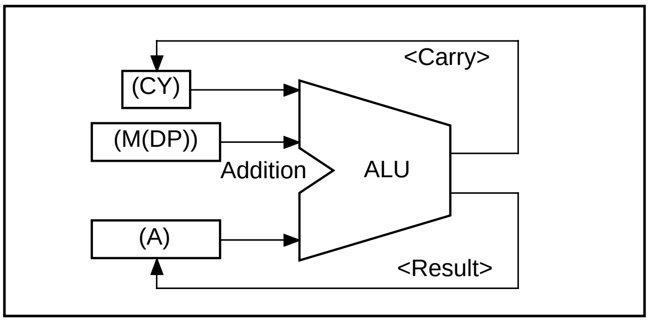


Fig. 1 AMC instruction execution example

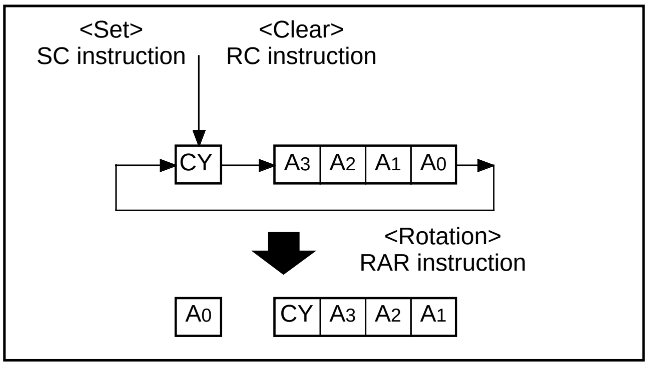


Fig. 2 RAR instruction execution example

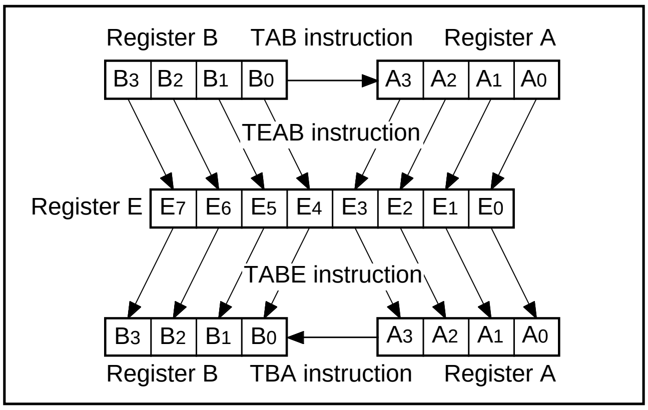


Fig. 3 Registers A, B and register E

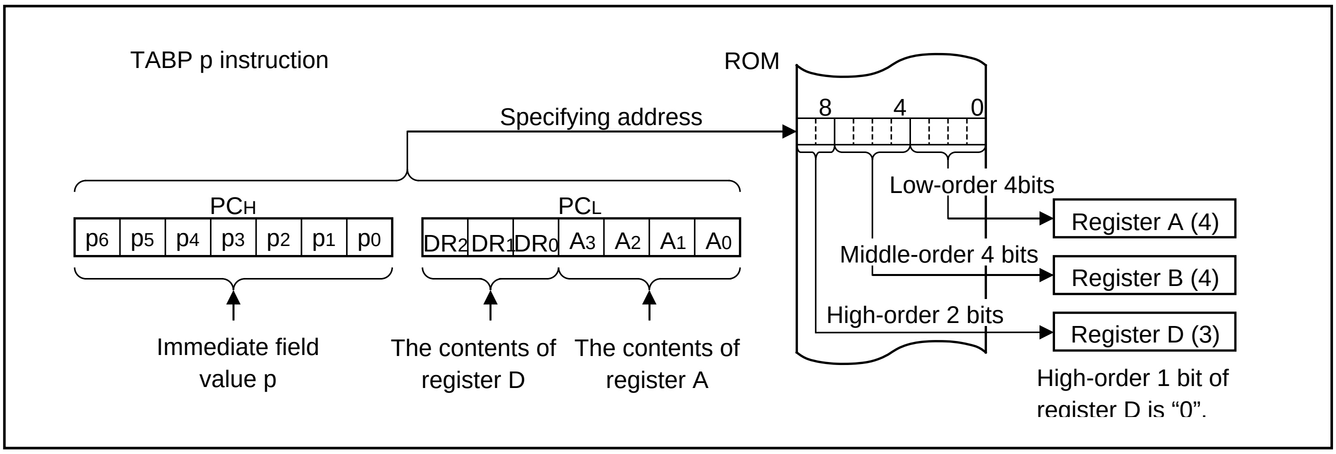


Fig. 4 TABP p instruction execution example

(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.

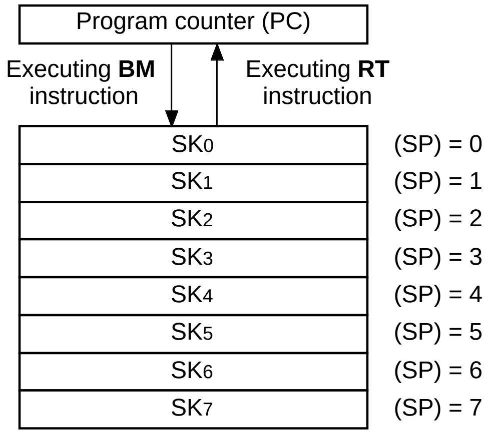
(6) Interrupt stack register (SDP)

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.

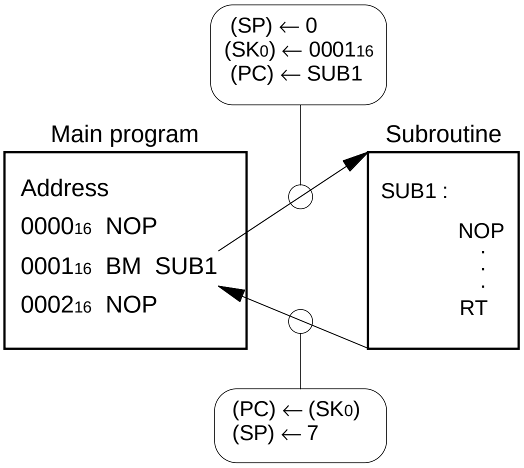
(7) Skip flag

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.



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

Fig. 5 Stack registers (SKs) structure



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

Fig. 6 Example of operation at subroutine call

(8) Program counter (PC)

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

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

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

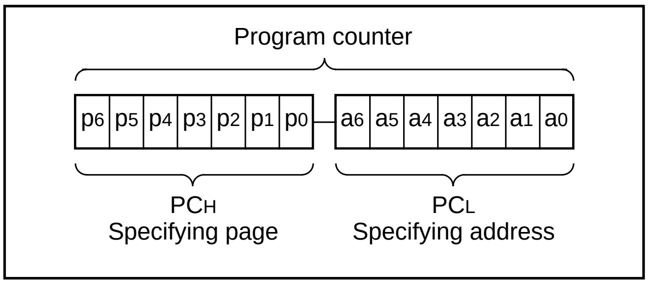


Fig. 7 Program counter (PC) structure

(9) Data pointer (DP)

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

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

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

• 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.

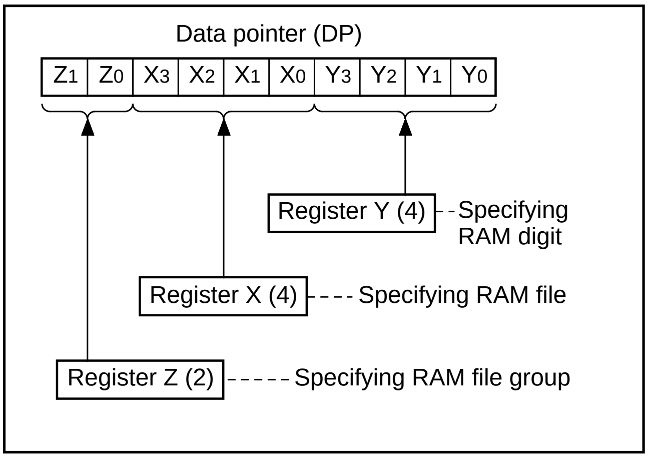


Fig. 8 Data pointer (DP) structure

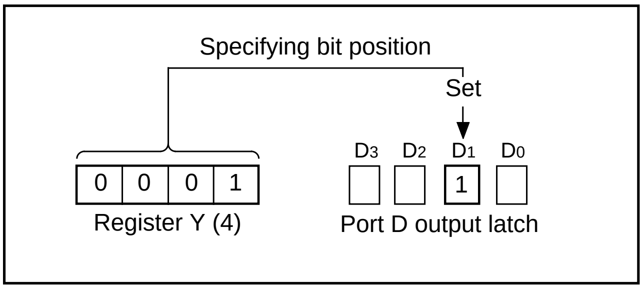


Fig. 9 SD instruction execution example

PROGRAM MEMORY (ROM)

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

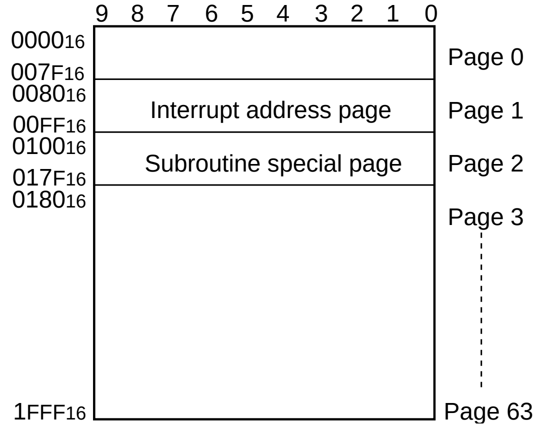


Fig. 10 ROM map of M34519M8/E8

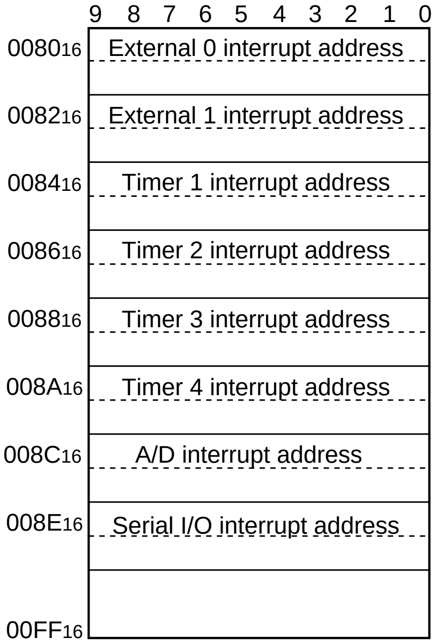


Fig. 11 Page 1 (addresses 0080_{16} to $00FF_{16}$) structure

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	

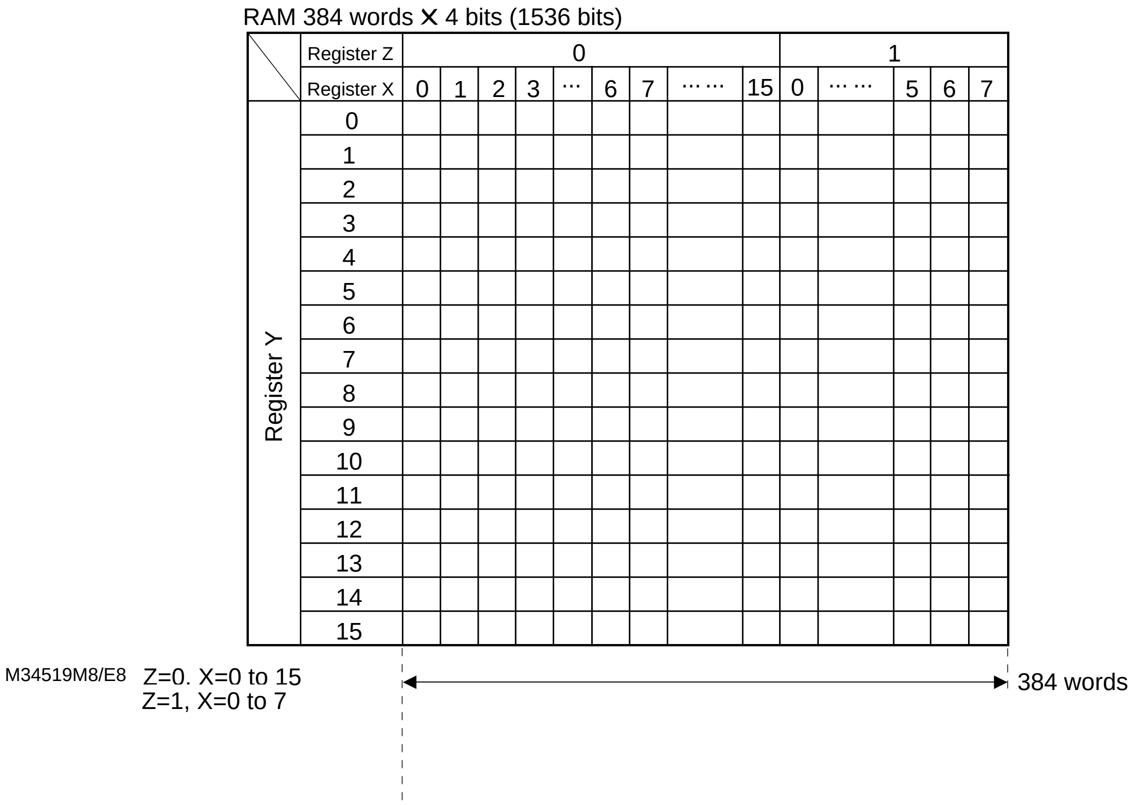


Fig. 12 RAM map

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).

(5) Interrupt processing

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

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

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

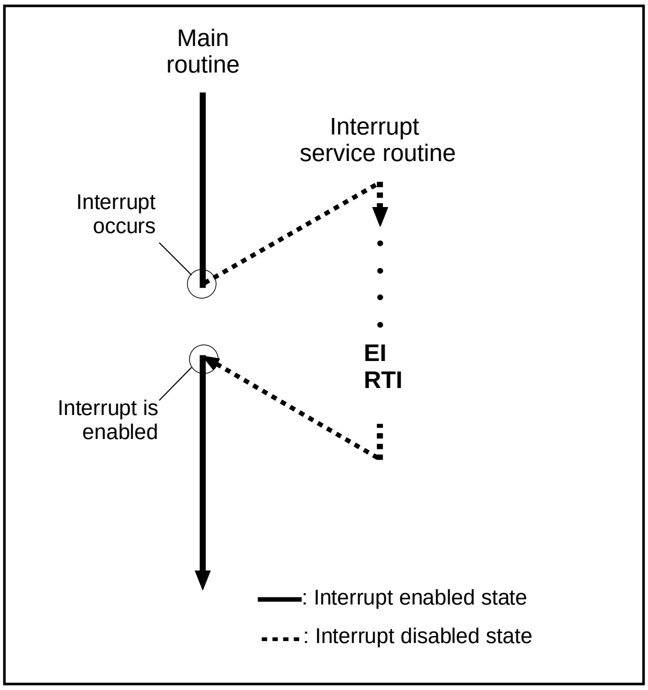


Fig. 13 Program example of interrupt processing

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

Fig. 14 Internal state when interrupt occurs

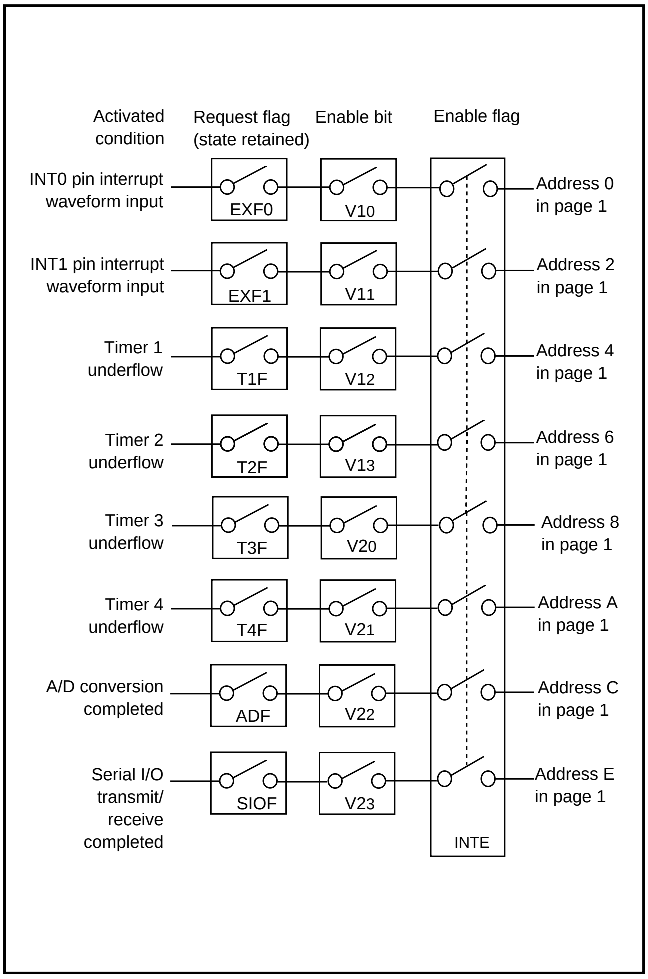


Fig. 15 Interrupt system diagram

(6) Interrupt control registers

- Interrupt control register V1
  Interrupt enable bits of external 0, external 1, timer 1 and timer 2 are assigned to register V1. Set the contents of this register through register A with the TV1A instruction. The TAV1 instruction can be used to transfer the contents of register V1 to register A.

- Interrupt control register V2
  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

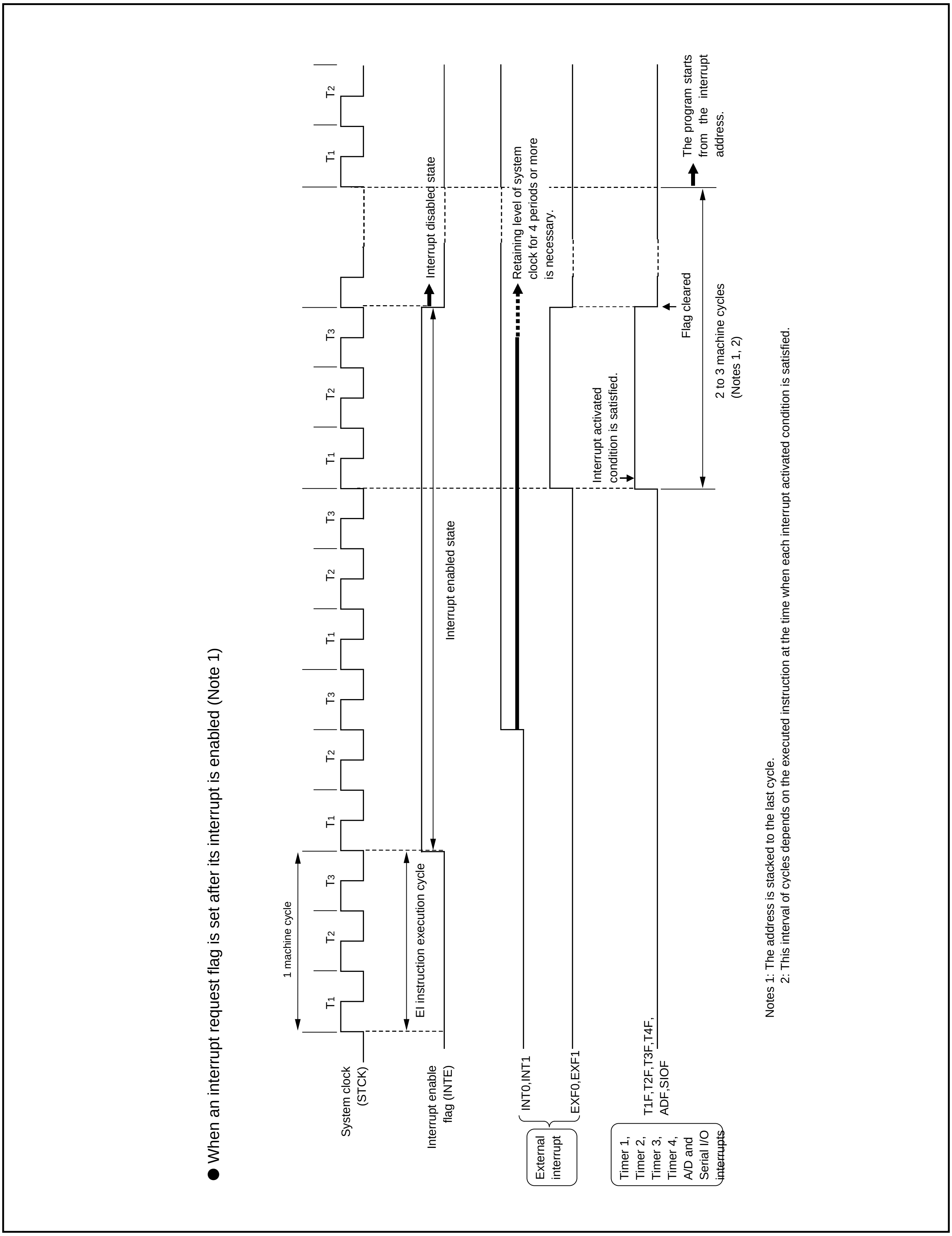
Interrupt control register V1		at reset : 0000_2	at RAM back-up : 0000_2 R/W TAV1/TV1A
$V1_3$	Timer 2 interrupt enable bit	0	Interrupt disabled (SNZT2 instruction is valid)
		1	Interrupt enabled (SNZT2 instruction is invalid)
$V1_2$	Timer 1 interrupt enable bit	0	Interrupt disabled (SNZT1 instruction 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).



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

Fig. 16 Interrupt sequence

EXTERNAL INTERRUPTS

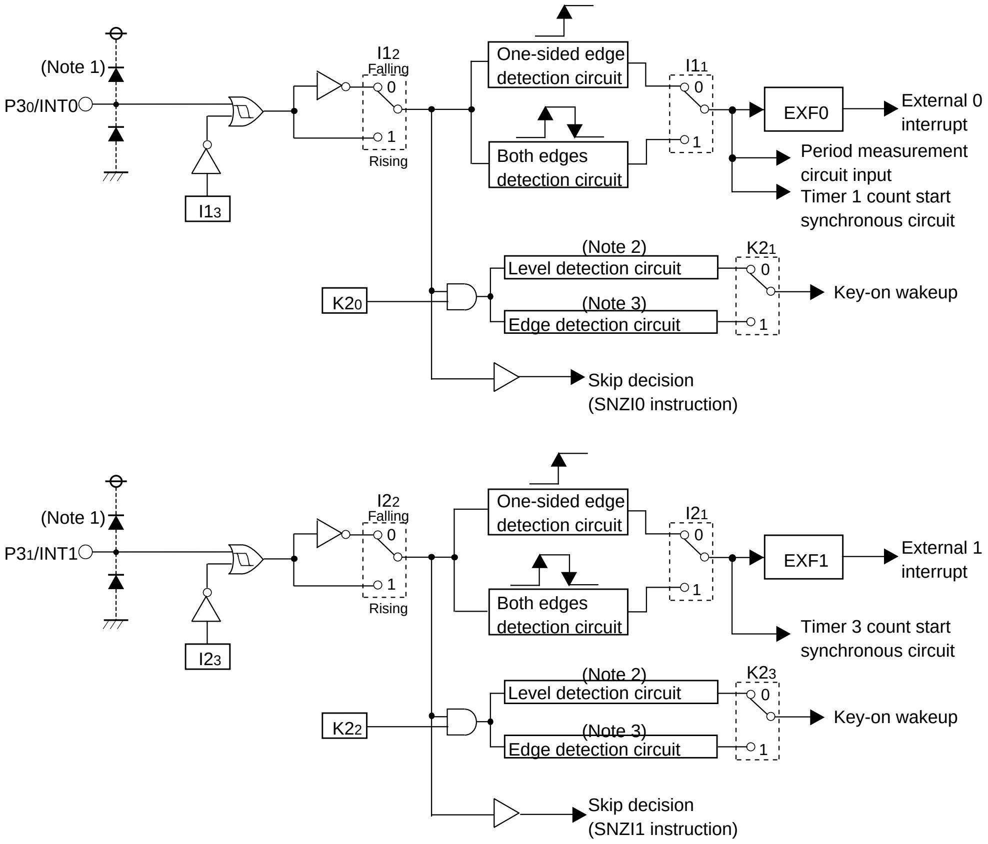
The 4519 Group has the external 0 interrupt and external 1 interrupt.

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



Notes 1: ----|◀---- This symbol represents a parasitic diode on the port.
2: $I1_2$ ($I2_2$) = 0: “L” level detected
$I1_2$ ($I2_2$) = 1: “H” level detected
3: $I1_2$ ($I2_2$) = 0: Falling edge detected
$I1_2$ ($I2_2$) = 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 (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.

(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       ; (×××02)
  TV1A            ; The SNZ0 instruction is valid ........... ①
  LA      8       ; (1×××2)
  TI1A            ; Control of INT0 pin input is changed
  NOP             .......................................................... ②
  SNZ0            ; The SNZ0 instruction is executed
                   (EXF0 flag cleared)
  NOP             .......................................................... ③
        :

     × : 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       ; (×××02)
  TK2A            ; Input of INT0 key-on wakeup invalid .. ①
  DI
  EPOF
  POF             ; RAM back-up
        :

     × : 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       ; (×××02)
  TV1A            ; The SNZ0 instruction is valid ........... ①
  LA      12      ; (×1××2)
  TI1A            ; Interrupt valid waveform is changed
  NOP             .......................................................... ②
  SNZ0            ; The SNZ0 instruction is executed
                   (EXF0 flag cleared)
  NOP             .......................................................... ③
        :

     × : 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        ; (××0×2)
   TV1A             ; The SNZ1 instruction is valid ...........①
   LA      8        ; (1×××2)
   TI2A             ; Control of INT1 pin input is changed
   NOP              .......................................................... ②
   SNZ1             ; The SNZ1 instruction is executed
                     (EXF1 flag cleared)
   NOP              .......................................................... ③
        ⋮

        × : 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        ; (×0××2)
   TK2A             ; Input of INT1 key-on wakeup invalid .. ①
   DI
   EPOF
   POF              ; RAM back-up
        ⋮

        × : 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        ; (××0×2)
   TV1A             ; The SNZ1 instruction is valid ...........①
   LA      12       ; (×1××2)
   TI2A             ; Interrupt valid waveform is changed
   NOP              .......................................................... ②
   SNZ1             ; The SNZ1 instruction is executed
                     (EXF1 flag cleared)
   NOP              .......................................................... ③
        ⋮

        × : 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.

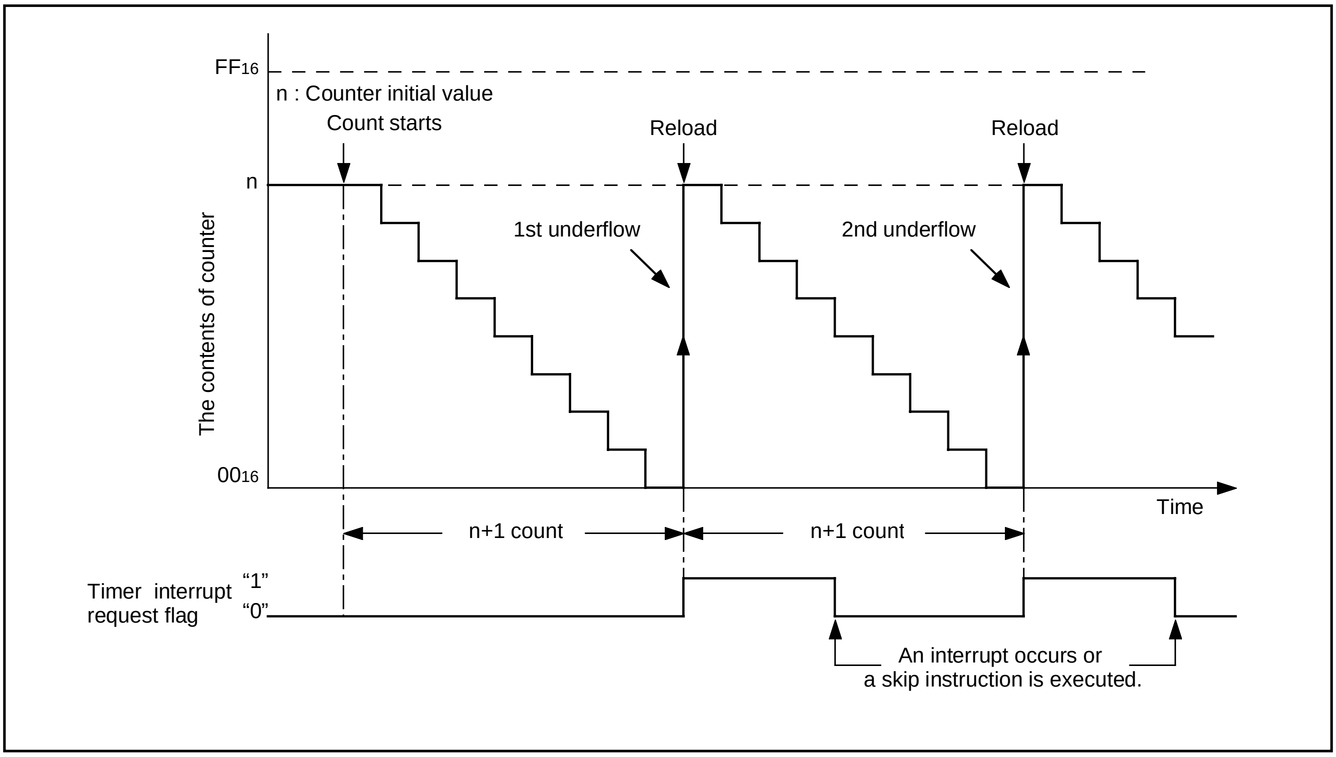


Fig. 24 Auto-reload function

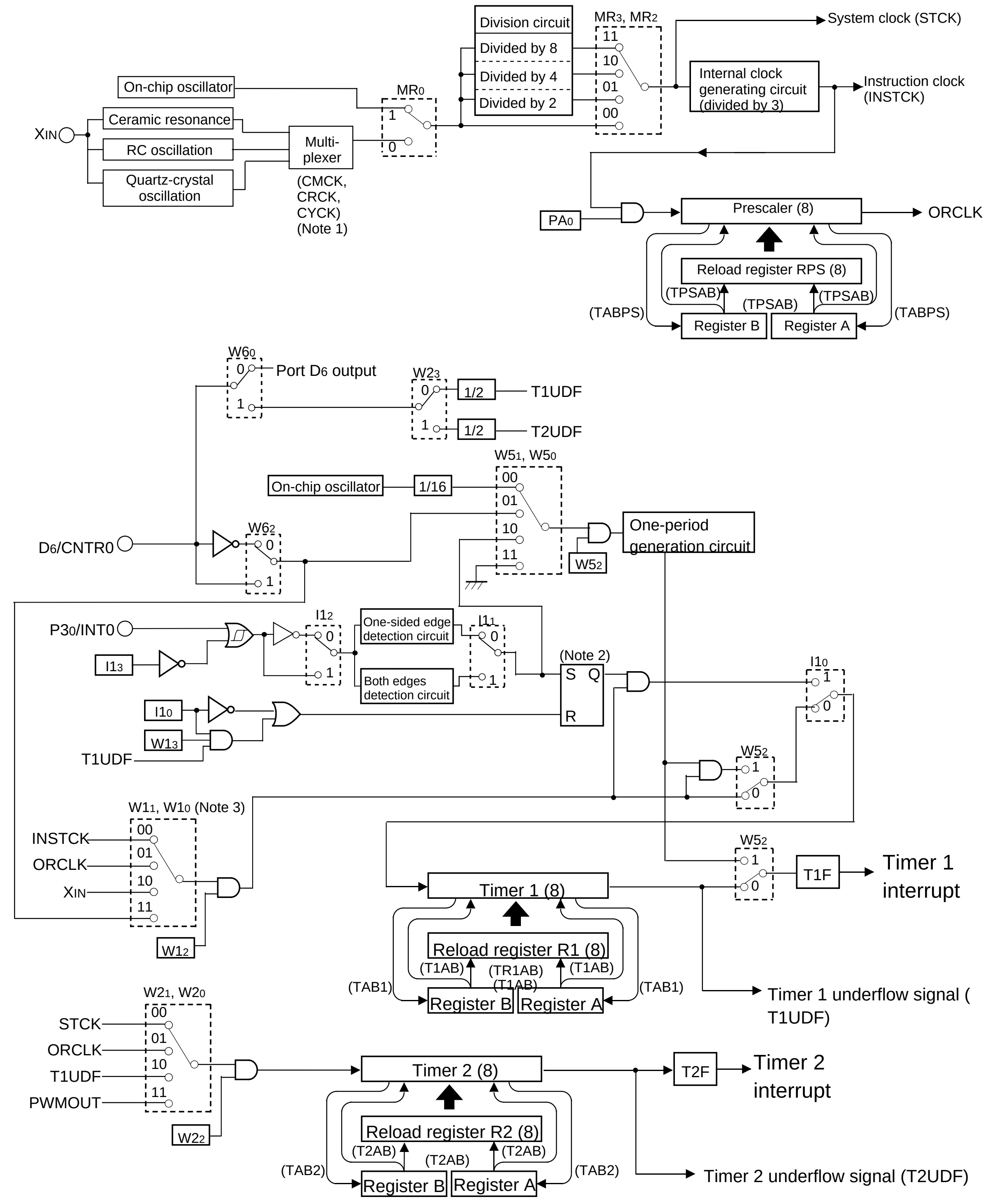
The 4519 Group timer consists of the following circuits.

- 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)

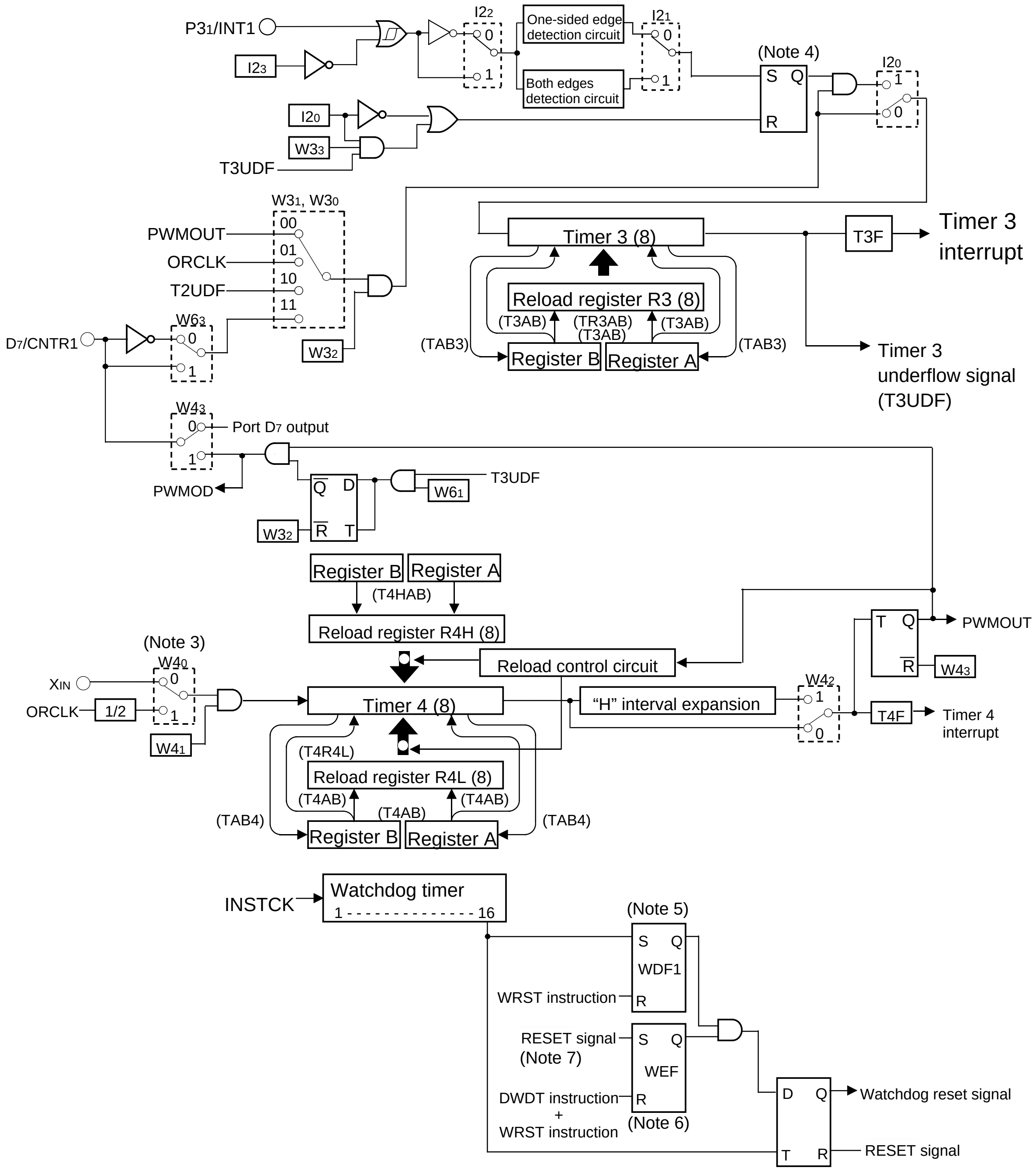
⬆ Data is set automatically from each reload register when timer underflows (auto-reload function).

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 P3$_0$/INT0 pin selected by bits 1 (I1$_1$) and 2 (I1$_2$) of register I1.

3: X$_{IN}$ cannot be used for the count source when bit 1 (MR$_1$) of register MR is set to “1” and f(X$_{IN}$) 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: X_{IN} cannot be used for the count source when bit 1 (MR_1) of register MR is set to “1” and f(X_{IN}) oscillation is stopped.
4: Timer 3 count start synchronous circuit is set by the valid edge of $P3_1$/INT1 pin selected by bits 1 ($I2_1$) and 2 ($I2_2$) 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	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))	
		0	1	$CNTR_0$ pin input	
$W5_0$		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, D_6/CNTR0 pin input, or $P3_0$/INT0 pin input (one cycle, “H”, or “L” pulse width at the case of a $P3_0$/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 X_{IN} 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 D_6/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 $P3_0$/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        ; (✕0✕✕2)
TV1A            ; The SNZT1 instruction is valid ........①
LA     0        ; (✕0✕✕2)
TW5A            ; Period measurement circuit stop
NOP             ......................................................... ②
SNZT1           ; The SNZT1 instruction is executed
                 (T1F flag cleared)
NOP             ......................................................... ③
     ⋮

   ✕ : 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 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.)

(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 $P3_0$/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: 3_0/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 $P3_0$/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 $P3_0$/INT0 pin is “H” or “H” pulse width (from rising to falling) when its level is “L” is measured.

When the input of $P3_0$/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 (D6/CNTR0 pin, D7/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 (D7/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.

When clearing bit 1 of register W4 to “0” to stop timer 4 while the PWM output function is used, avoid a timing when timer 4 underflows.

(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 register 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
  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.

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

     × : 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.

- Prescaler, Timer 1, Timer 2 and Timer 3 count start timing and count time when operation starts

  Count starts from the first rising edge of the count source (2) after Prescaler, Timer 1, Timer 2 and Timer 3 operations start (1).

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

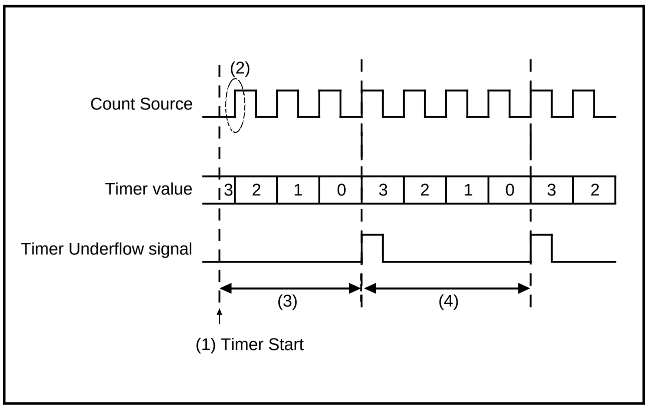


Fig. 29 Timer count start timing and count time when operation starts (Prescaler, Timer 1, Timer 2 and Timer 3)

- Timer 4 count start timing and count time when operation starts

  Count starts from the rising edge (2) after the first falling edge of the count source, after Timer 4 operations start (1).

  Time to first underflow (3) is different from time among next underflow (4) by the timing to start the timer and count source operations after count starts.

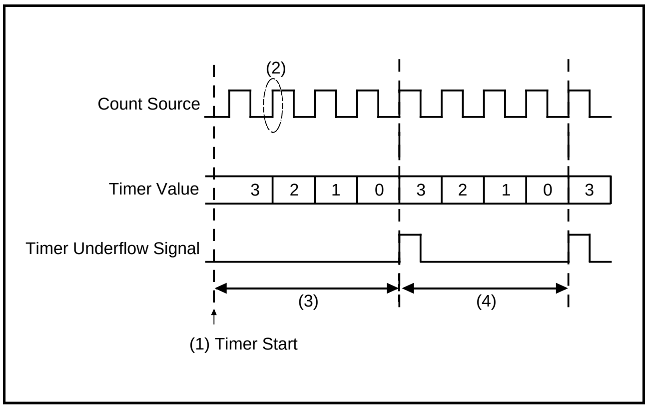
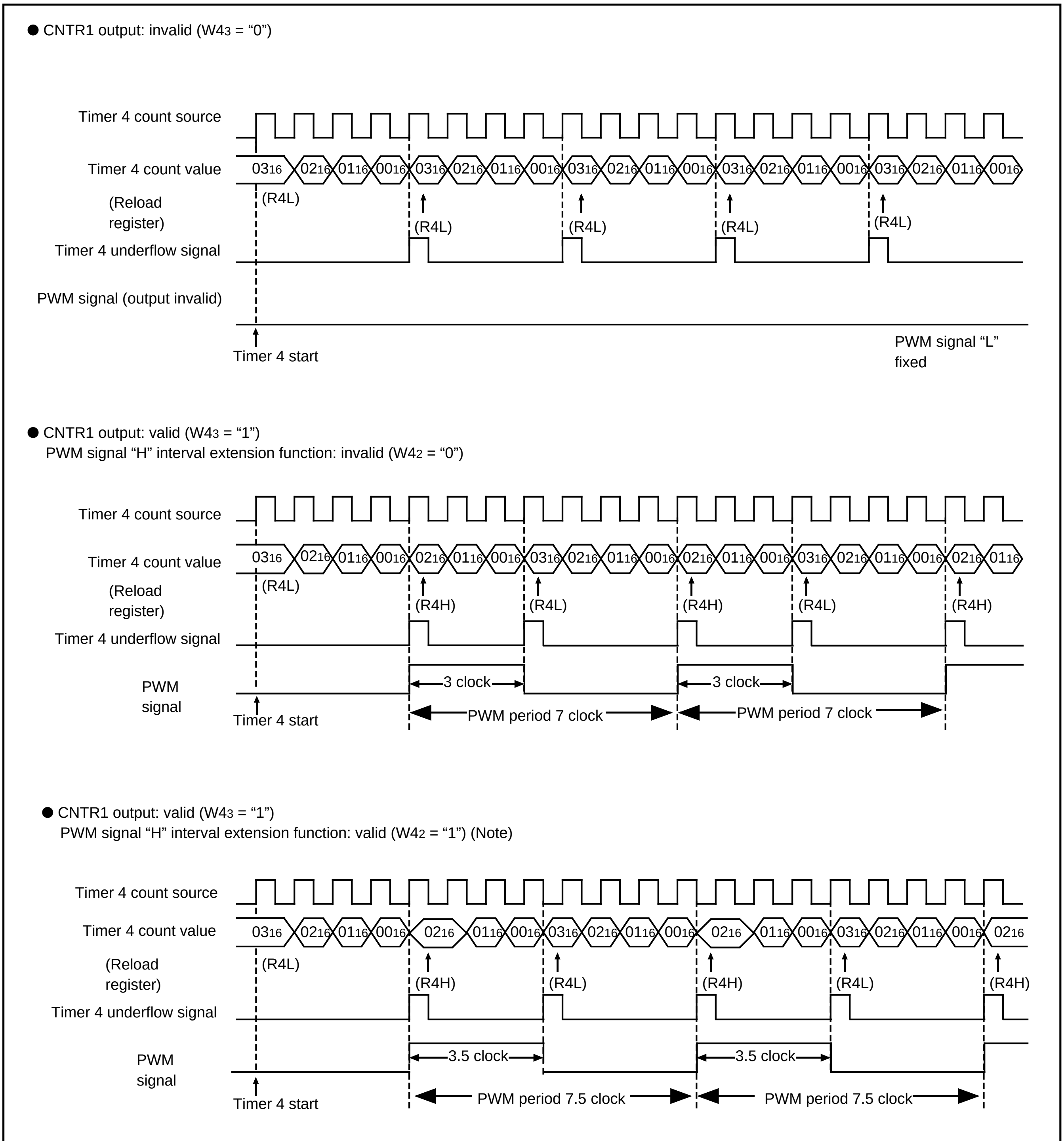
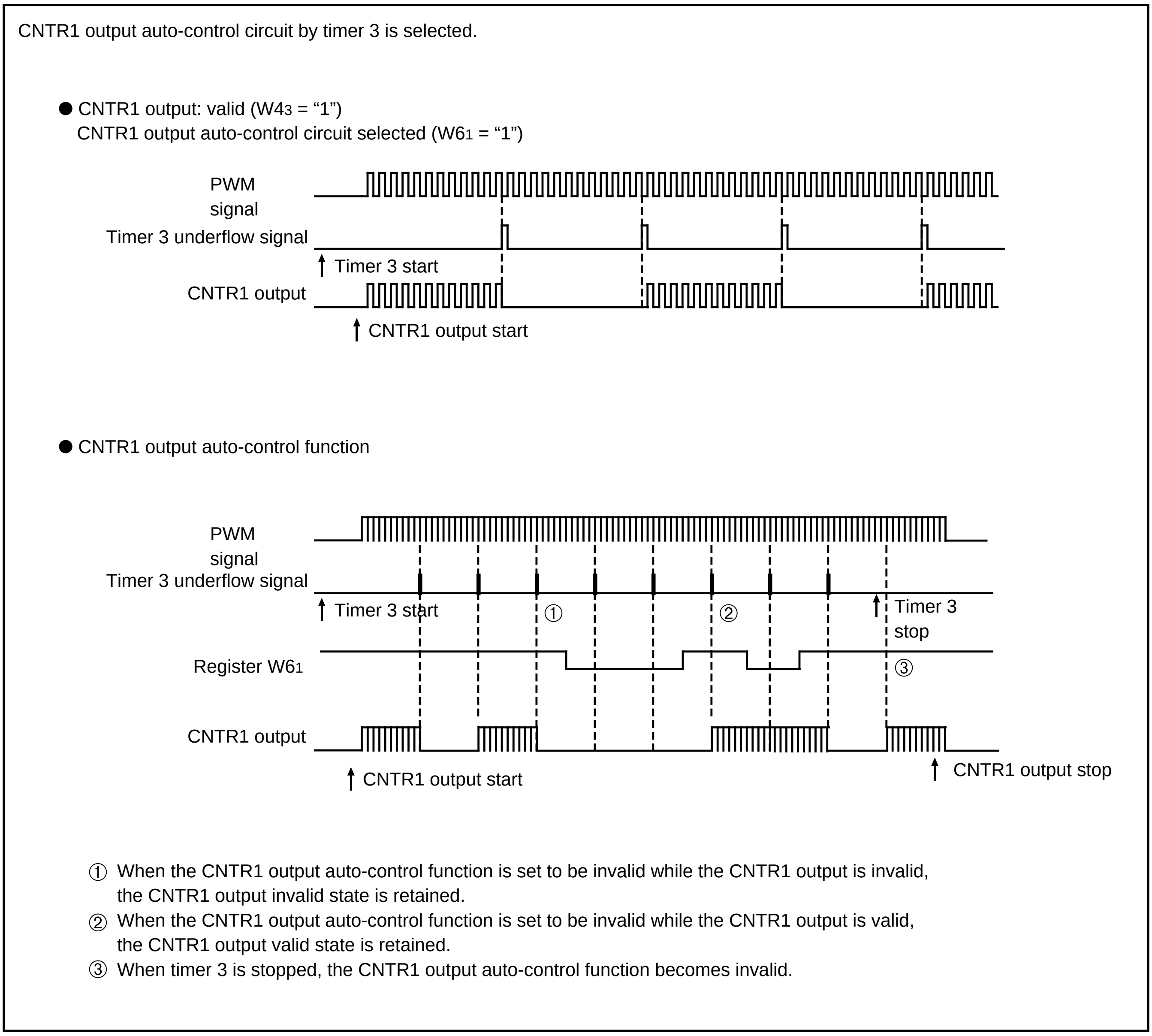


Fig. 30 Timer count start timing and count time when operation starts (Timer 4)



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

Fig. 31 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”)

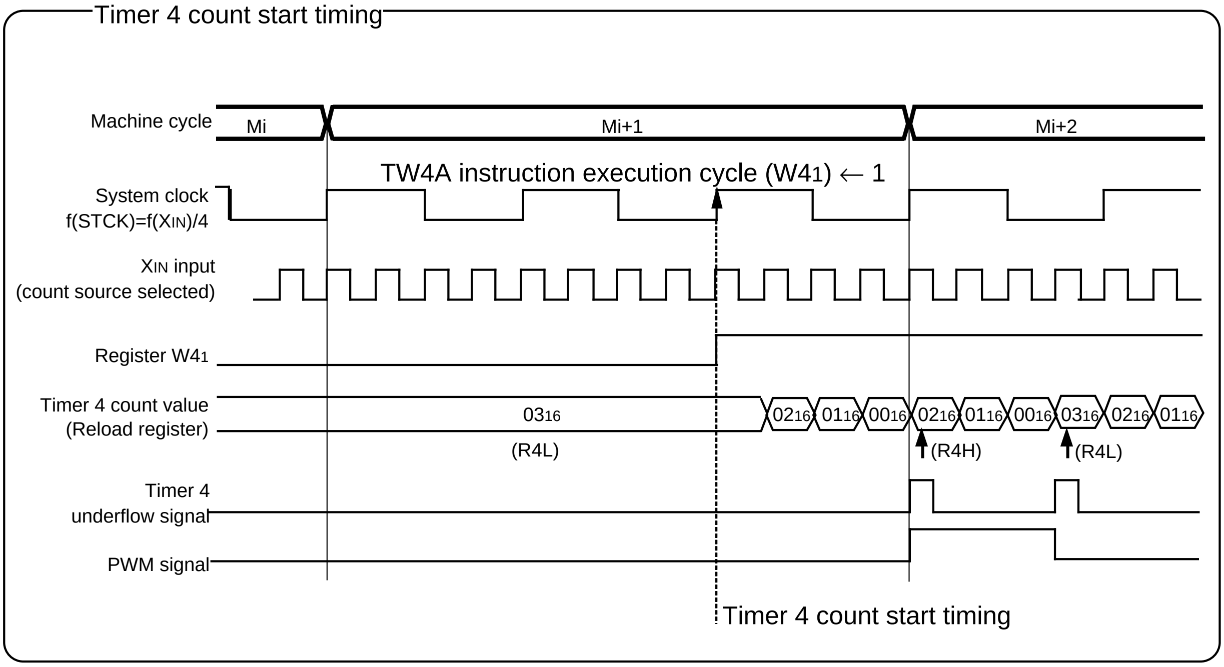
- CNTR1 output auto-control function

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.

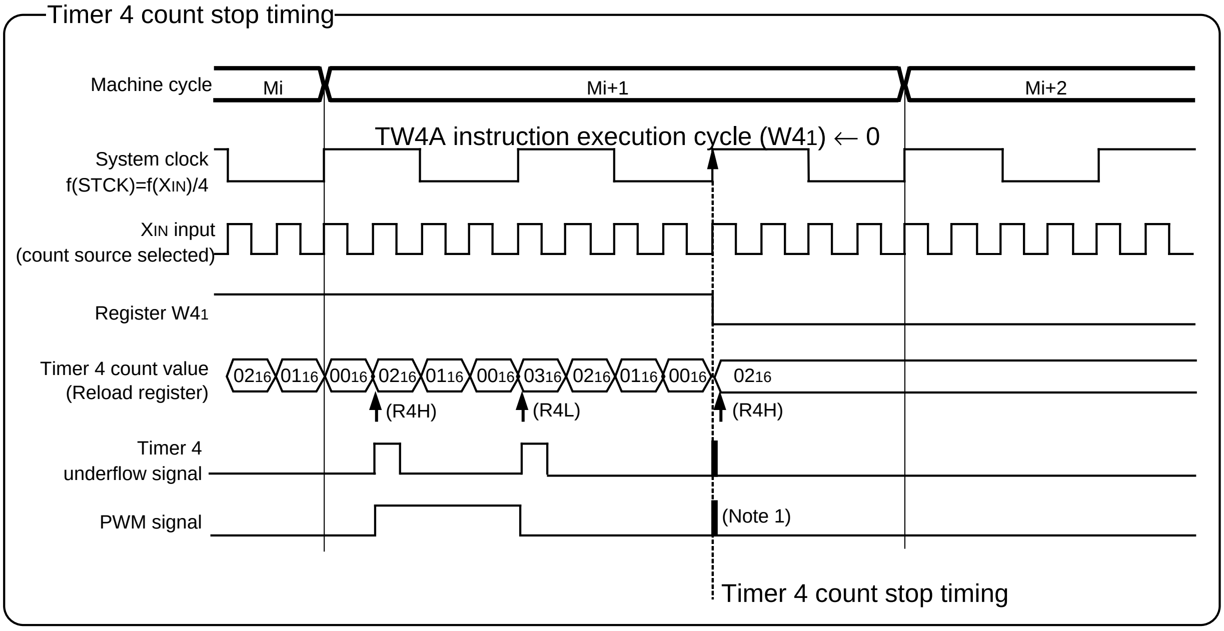
Fig. 32 CNTR1 output auto-control function by timer 3

●Waveform extension function of CNTR1 output "H" interval: Invalid ($W4_2$ = "0"),
CNTR1 output: valid ($W4_3$ = "1"),
Count source: X_{IN} input selected ($W4_0$ = "0"),
Reload register R4L: "03_{16}"
Reload register R4H: "02_{16}"

Timer 4 count start timing



Timer 4 count stop timing



Notes 1: In order to stop timer 4 at CNTR1 output valid ($W4_3$ = "1"), avoid a timing when timer 4 underflows.
If these timings overlap, a hazard may occur in a CNTR1 output waveform.
2: At CNTR1 output valid, timer 4 stops after "H" interval of PWM signal set by reload register R4H is output.

Fig. 33 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 “FFFF16” after system is released from reset.

After the count is started, when the timer WDT underflow occurs (after the count value of timer WDT reaches “000016,” 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.

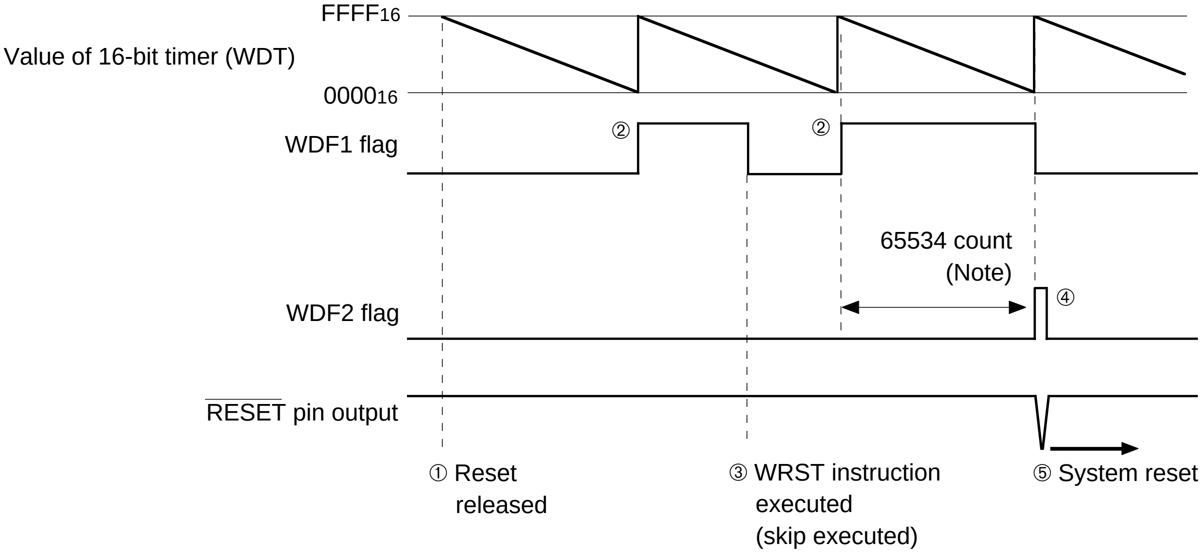
When the DWDT instruction and the WRST instruction are executed continuously, the WEF flag is cleared to “0” and the watchdog timer function is invalid.

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. 34 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 35).
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 36).
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. 35 Program example to start/stop watchdog timer

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

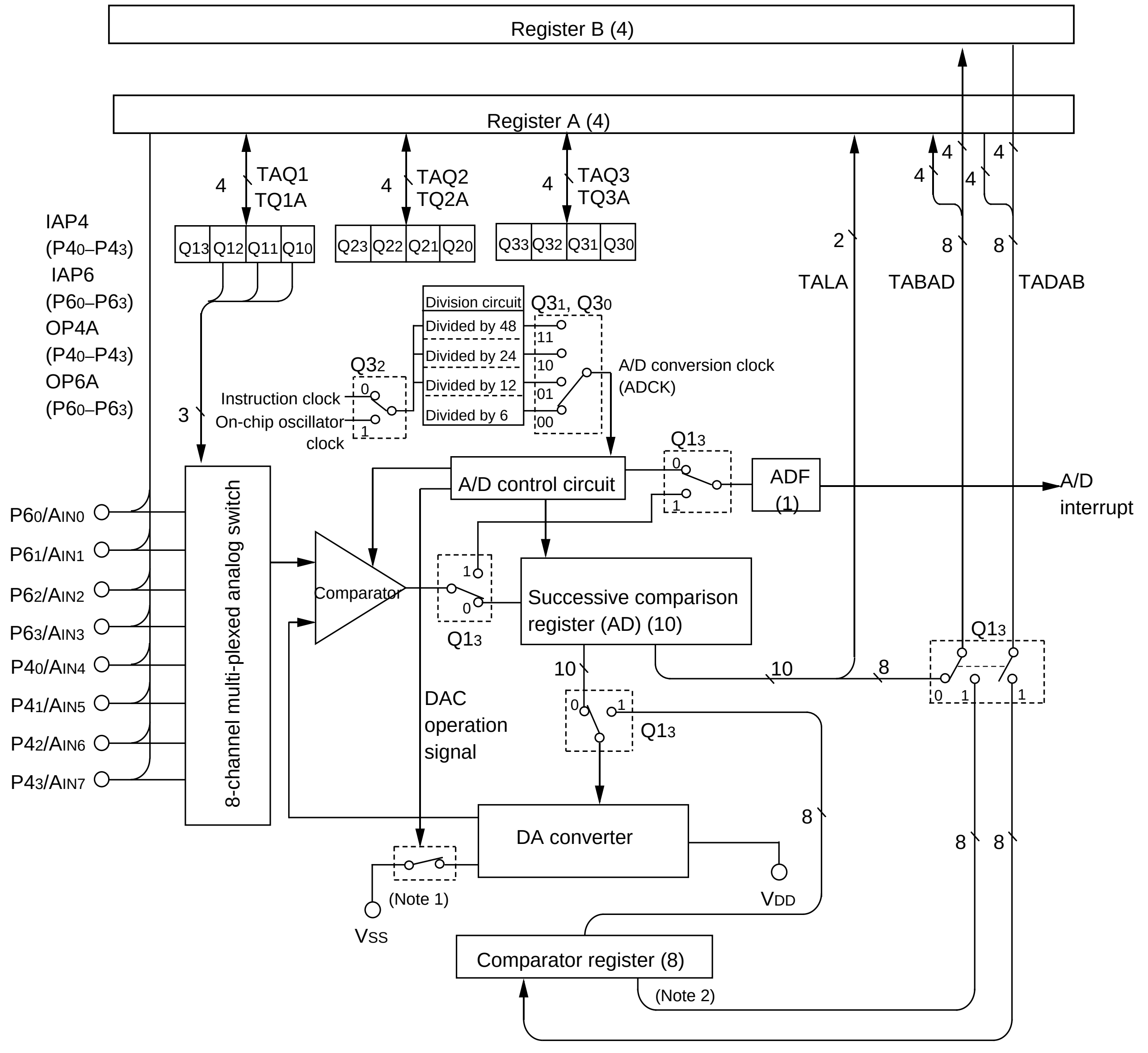
Fig. 36 Program example to enter the mode when using the watchdog timer

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 ≤ VDD ≤ 5.5V)
	Differential non-linearity error: ±0.9LSB (2.2 V ≤ VDD ≤ 5.5V)
Conversion speed	31 μs (f(XIN) = 6 MHz, STCK = f(XIN) (XIN 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. 37 A/D conversion circuit structure

Table 12 A/D control registers

<table>
<tr><td colspan="2">A/D control register Q1</td><td colspan="3">at reset : 0000_2</td><td>at RAM back-up : state retained</td><td>R/W
TAQ1/TQ1A</td></tr>
<tr><td rowspan="2">$Q1_3$</td><td rowspan="2">A/D operation mode selection bit</td><td colspan="5">A/D conversion mode</td></tr>
<tr><td colspan="5">Comparator mode</td></tr>
<tr><td rowspan="3">$Q1_2$</td><td rowspan="9">Analog input pin selection bits</td><td>$Q1_2$</td><td>$Q1_1$</td><td>$Q1_0$</td><td colspan="2">Analog input pins</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td colspan="2">AIN0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="2">AIN1</td></tr>
<tr><td rowspan="3">$Q1_1$</td><td>0</td><td>1</td><td>0</td><td colspan="2">AIN2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td colspan="2">AIN3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td colspan="2">AIN4</td></tr>
<tr><td rowspan="3">$Q1_0$</td><td>1</td><td>0</td><td>1</td><td colspan="2">AIN5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td colspan="2">AIN6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td colspan="2">AIN7</td></tr>
</table>

<table>
<tr><td colspan="2">A/D control register Q2</td><td>at reset : 0000_2</td><td>at RAM back-up : state retained</td><td>R/W
TAQ2/TQ2A</td></tr>
<tr><td rowspan="2">$Q2_3$</td><td rowspan="2">$P4_0$/AIN4, $P4_1$/AIN5, $P4_2$/AIN6, $P4_3$/AIN7 pin function selection bit</td><td>0</td><td colspan="2">$P4_0$, $P4_1$, $P4_2$, $P4_3$</td></tr>
<tr><td>1</td><td colspan="2">AIN4, AIN5, AIN6, AIN7</td></tr>
<tr><td rowspan="2">$Q2_2$</td><td rowspan="2">$P6_2$/AIN2, $P6_3$/AIN3 pin function selection bit</td><td>0</td><td colspan="2">$P6_2$, $P6_3$</td></tr>
<tr><td>1</td><td colspan="2">AIN2, AIN3</td></tr>
<tr><td rowspan="2">$Q2_1$</td><td rowspan="2">$P6_1$/AIN1 pin function selection bit</td><td>0</td><td colspan="2">$P6_1$</td></tr>
<tr><td>1</td><td colspan="2">AIN1</td></tr>
<tr><td rowspan="2">$Q2_0$</td><td rowspan="2">$P6_0$/AIN0 pin function selection bit</td><td>0</td><td colspan="2">$P6_0$</td></tr>
<tr><td>1</td><td colspan="2">AIN0</td></tr>
</table>

<table>
<tr><td colspan="2">A/D control register Q3</td><td colspan="2">at reset : 0000_2</td><td>at RAM back-up : state retained</td><td>R/W
TAQ3/TQ3A</td></tr>
<tr><td rowspan="2">$Q3_3$</td><td rowspan="2">Not used</td><td colspan="2">0</td><td rowspan="2" colspan="2">This bit has no function, but read/write is enabled.</td></tr>
<tr><td colspan="2">1</td></tr>
<tr><td rowspan="2">$Q3_2$</td><td rowspan="2">A/D converter operation clock selection bit</td><td colspan="2">0</td><td colspan="2">Instruction clock (INSTCK)</td></tr>
<tr><td colspan="2">1</td><td colspan="2">On-chip oscillator (f(RING))</td></tr>
<tr><td rowspan="2">$Q3_1$</td><td rowspan="5">A/D converter operation clock division ratio selection bits</td><td>$Q3_1$</td><td>$Q3_0$</td><td colspan="2">Division ratio</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">Frequency divided by 6</td></tr>
<tr><td rowspan="3">$Q3_0$</td><td>0</td><td>1</td><td colspan="2">Frequency divided by 12</td></tr>
<tr><td>1</td><td>0</td><td colspan="2">Frequency divided by 24</td></tr>
<tr><td>1</td><td>1</td><td colspan="2">Frequency divided by 48</td></tr>
</table>

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 $P4_0/A_{IN4}$–$P4_3/A_{IN7}$, $P6_0/A_{IN0}$–$P6_3/A_{IN3}$. 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 V_{ref} generated from the built-in D/A converter can be obtained with the reference voltage V_{DD} by the following formula:

Logic value of comparison voltage V_{ref}

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

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

(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 V_{ref} is compared with the analog input voltage V_{IN}.

③ When the comparison result is $V_{ref} < V_{IN}$, the topmost bit of the register AD remains set to “1.” When the comparison result is $V_{ref} > V_{IN}$, it is cleared to “0.”

The 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 $f(X_{IN})$ = 6.0 MHz in X_{IN} 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 38).

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

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

✻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 38 shows the A/D conversion timing chart.

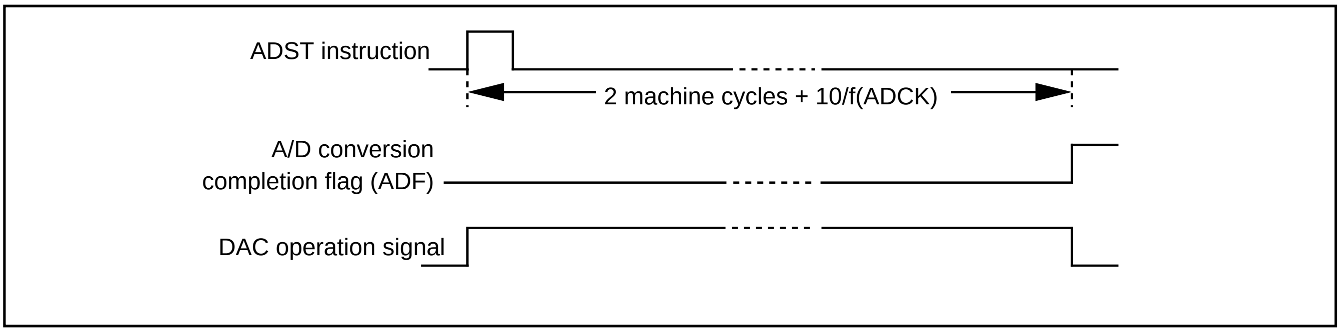


Fig. 38 A/D conversion timing chart

(8) How to use A/D conversion

How to use A/D conversion is explained using as example in which the analog input from P60/AIN0 pin is A/D converted, and the high-order 4 bits of the converted data are stored in address M(Z, X, Y) = (0, 0, 0), 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 AIN0 pin function with the bit 0 of the register Q2. Select the AIN0 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 39)
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).

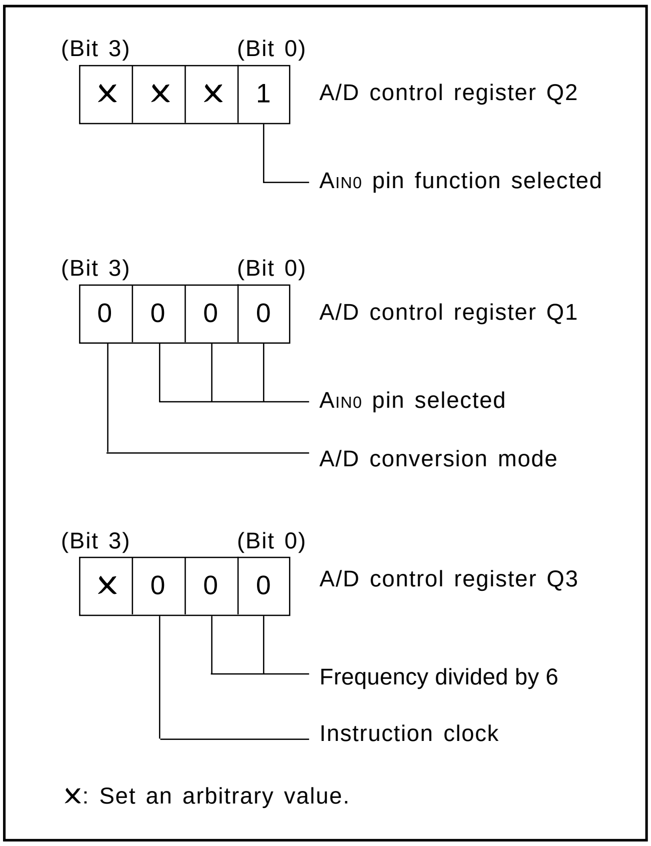


Fig. 39 Setting registers

(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(X_{IN}) = 6.0 MHz in X_{IN} 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.

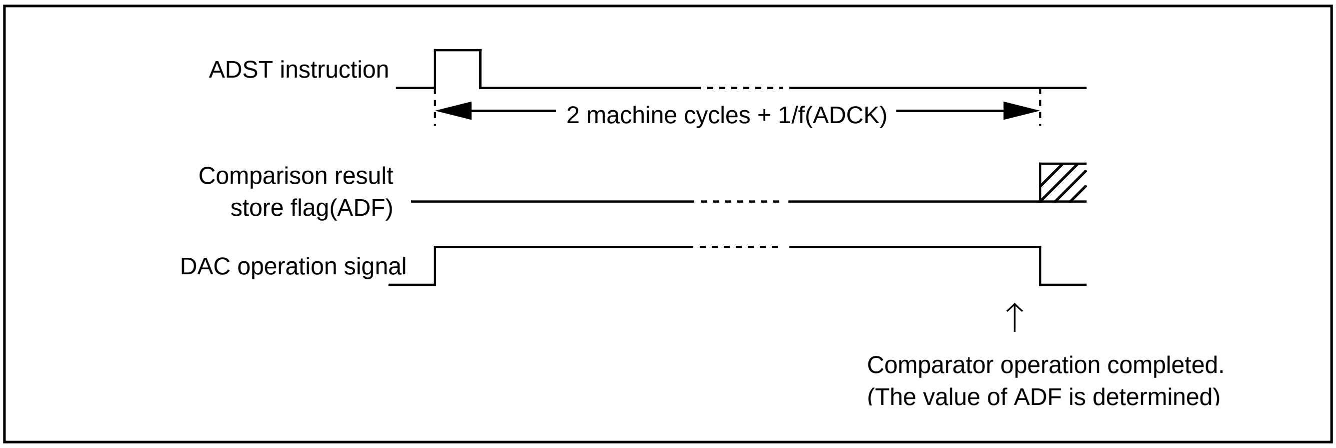


Fig. 40 Comparator operation timing chart

(14) Definition of A/D converter accuracy

The A/D conversion accuracy is defined below (refer to Figure 41).

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

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

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

- 1LSB at relative accuracy → $\frac{V_{FST}-V_{0T}}{1022}$ (V)
- 1LSB at absolute accuracy → $\frac{V_{DD}}{1024}$ (V)

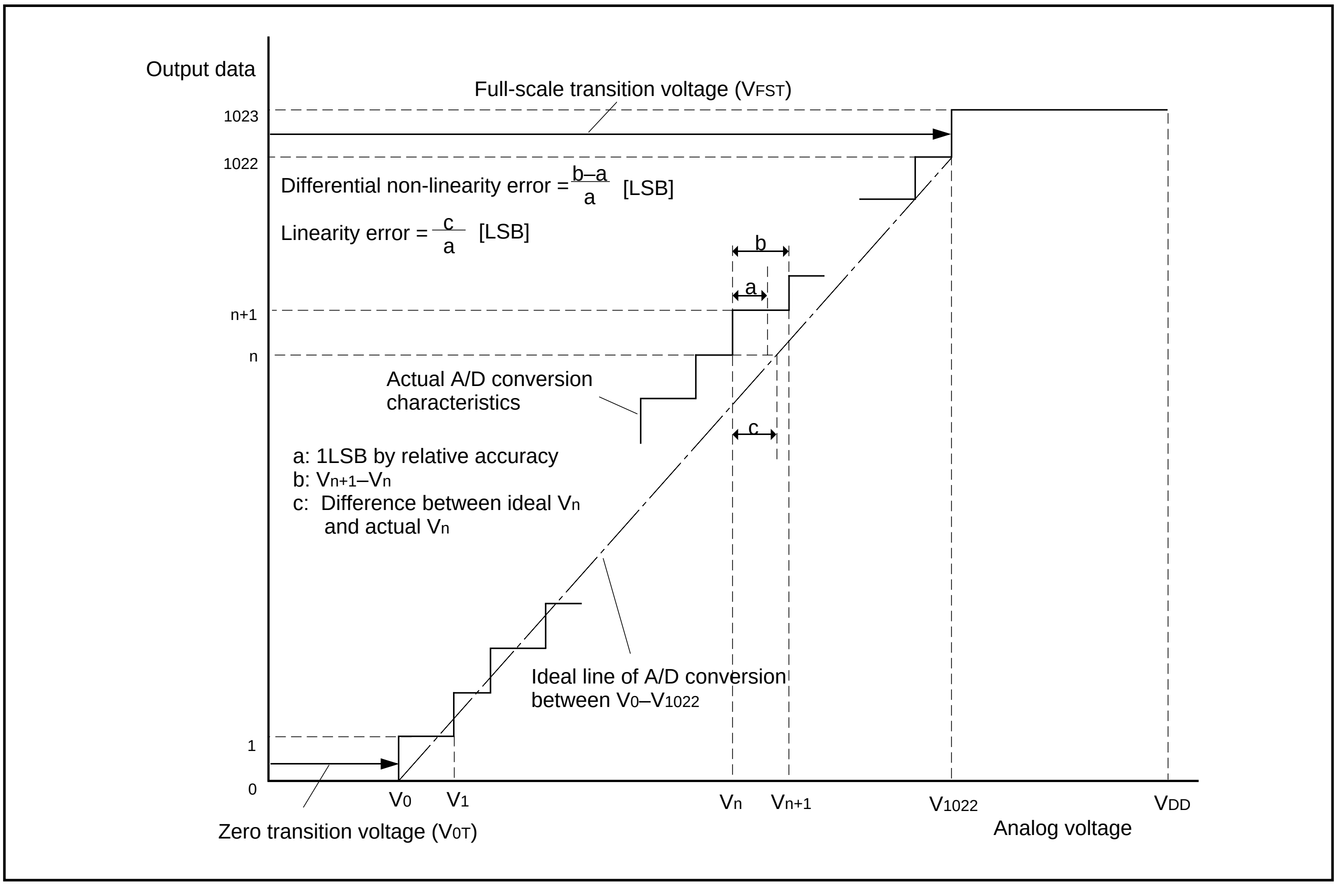


Fig. 41 Definition of A/D conversion accuracy

SERIAL INTERFACE

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.

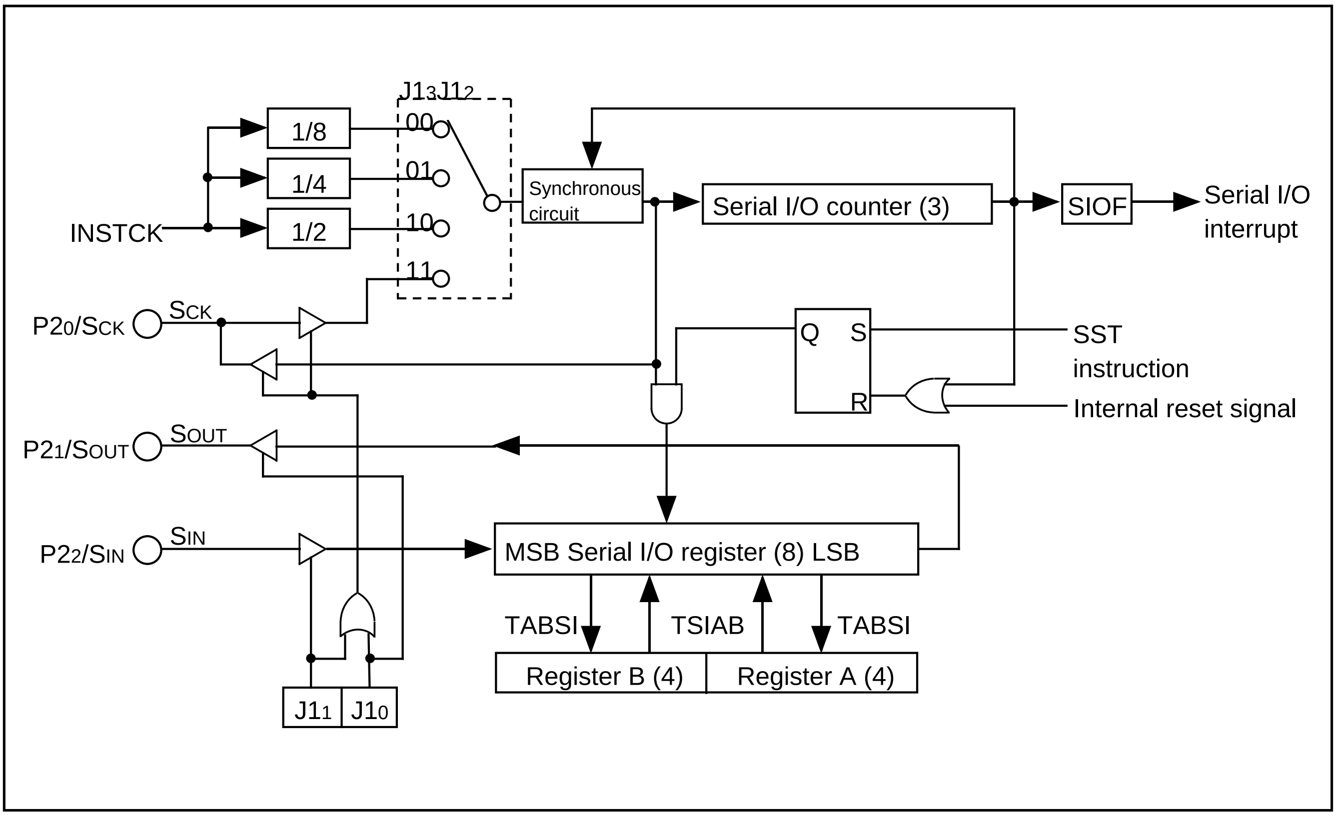


Fig. 42 Serial I/O structure

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
		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 (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.

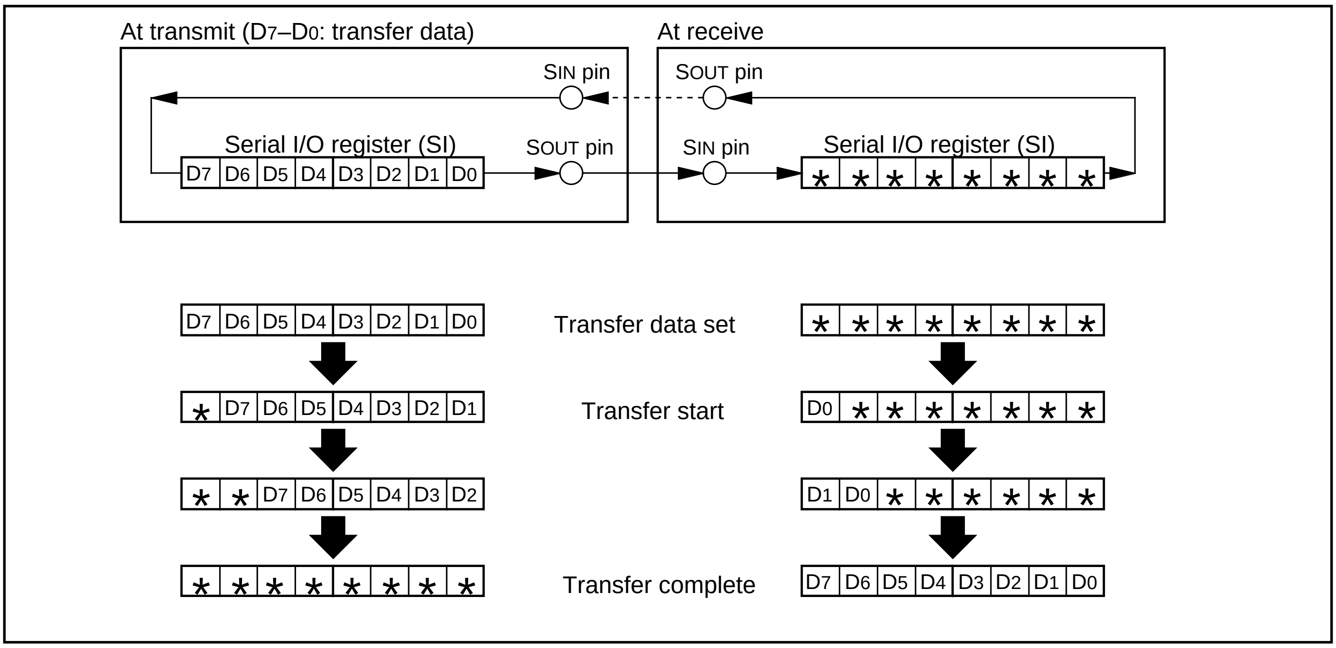


Fig. 43 Serial I/O register state when transferring

(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 44 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 44 shows the data transfer timing and Table 16 shows the data transfer sequence.

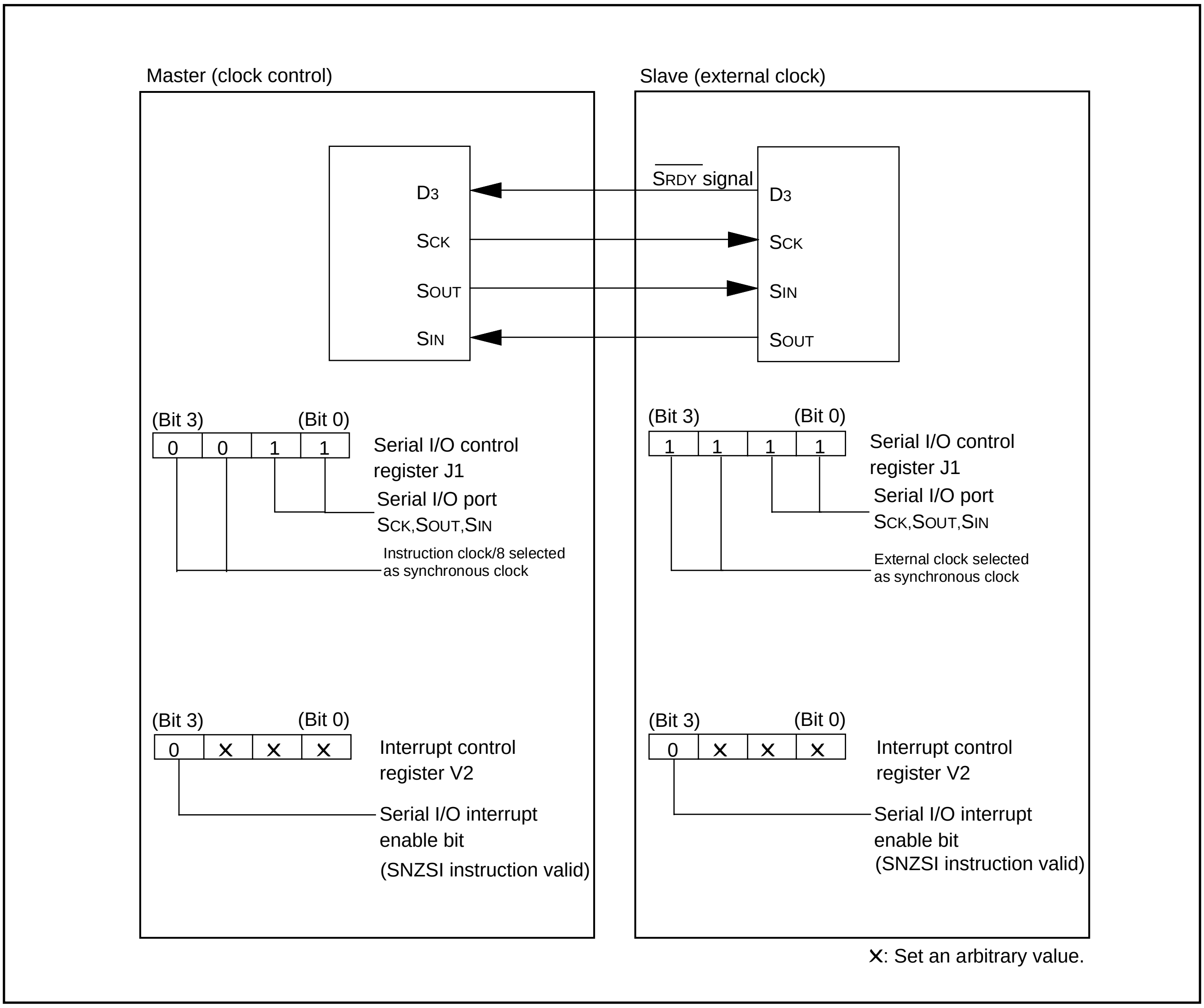
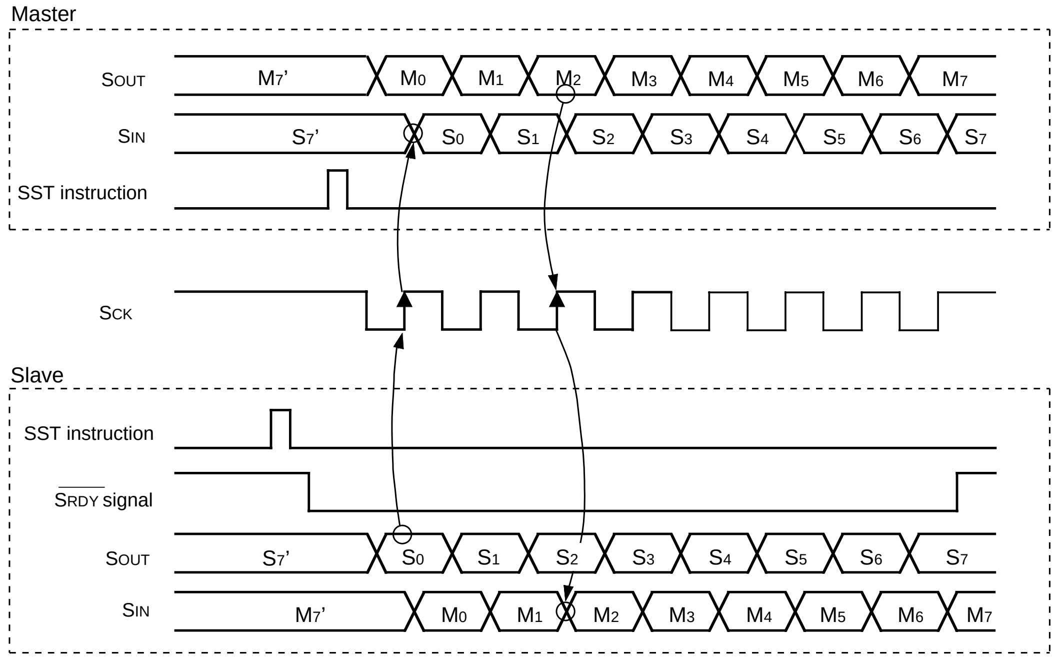


Fig. 44 Serial I/O connection example



M_0–M_7: Contents of master serial I/O register
S_0–S_7: Contents of slave serial I/O register
Rising of SCK: Serial input
Falling of SCK: Serial output

Fig. 45 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 44.	[Initial setting] • Setting serial I/O mode register J1, and interrupt control register V2 shown in Figure 44.
TJ1A and TV2A instructions	TJ1A and TV2A instructions
• Setting the port received the reception enable signal ($\overline{S_{RDY}}$) to the input mode. (Port D_3 is used in this example)	• Setting the port transmitted the reception enable signal (S_{RDY}) 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.

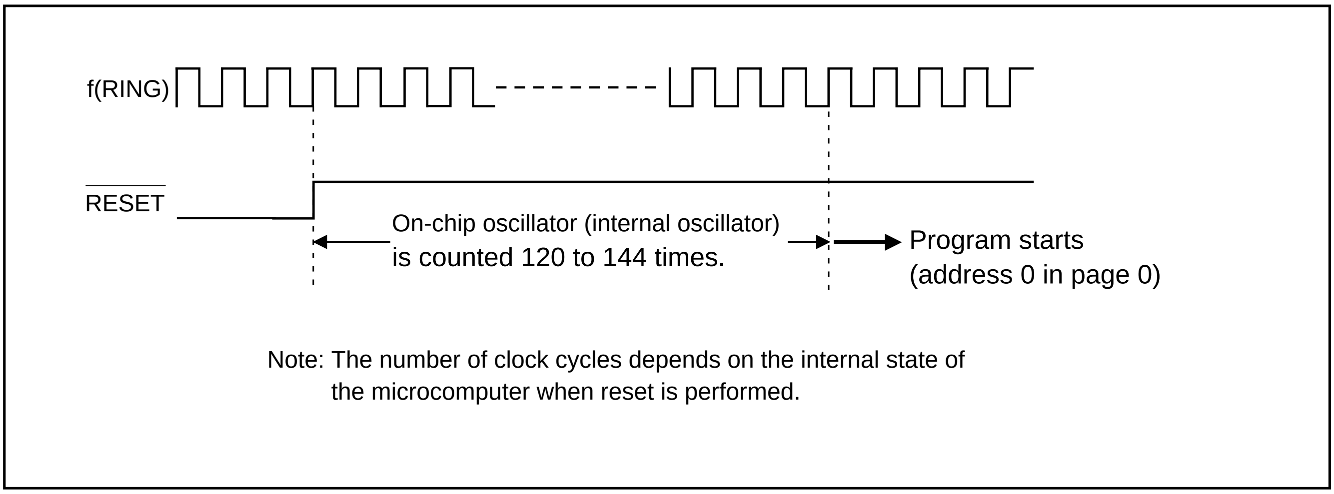


Fig. 46 Reset release timing

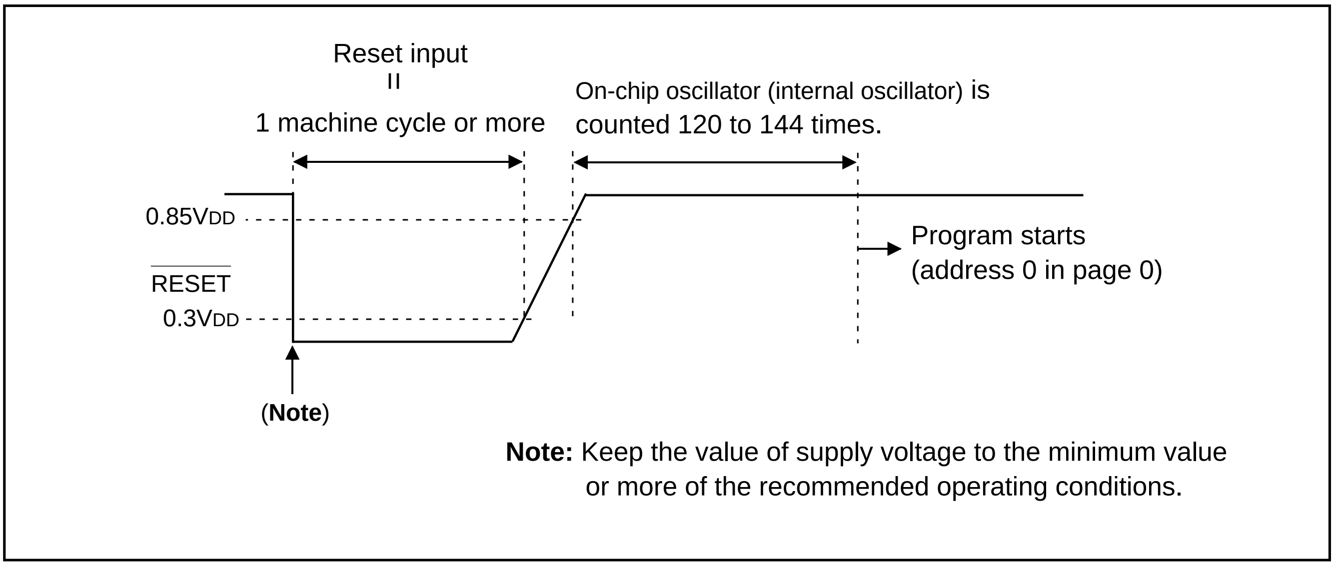
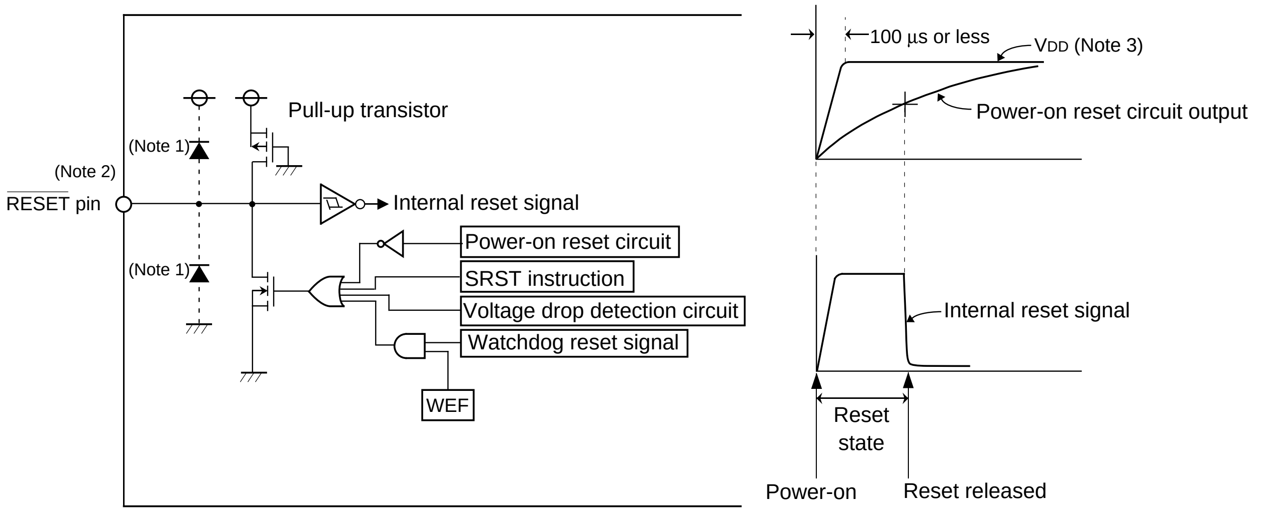


Fig. 47 $\overline{\text{RESET}}$ pin input waveform and reset operation

(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 VSS 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 VDD or less.
3: Keep the value of supply voltage to the minimum value or more of the recommended operating conditions.

Fig. 48 Structure of reset pin and its peripherals,, and power-on reset operation

Table 1 Port state at reset

Name	Function	State
D0–D5	D0–D5	High-impedance (Notes 1, 2)
D6/CNTR0	D6	High-impedance (Notes 1, 2)
D7/CNTR1	D7	High-impedance (Notes 1, 2)
P00–P03	P00–P03	High-impedance (Notes 1, 2, 3)
P10–P13	P10–P13	High-impedance (Notes 1, 2, 3)
P20/SCK, P21/SOUT, P22/SIN	P20–P22	High-impedance (Note 1)
P30/INT0, P31/INT1, P32, P33	P30–P33	High-impedance (Note 1)
P40/AIN4–P43/AIN7	P40–P43	High-impedance (Note 1)
P50–P53	P50–P53	High-impedance (Notes 1, 2)
P60/AIN0–P63/AIN3	P60–P63	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 49 and 50 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) Address 0 in page 0 is set to program counter.	0 0 0 0 0 0 0 0 0 0 0 0 0 0	
• 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	✕ ✕ ✕ ✕ ✕ ✕ ✕ ✕	
• 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	✕ ✕ ✕ ✕ ✕ ✕ ✕ ✕ ✕ ✕	
• Comparator register	✕ ✕ ✕ ✕ ✕ ✕ ✕ ✕	
• 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	

"✕" represents undefined.

Fig. 49 Internal state at reset 1

• 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	× × ×
• Register E	× × × × × × × ×
• Register X	0 0 0 0
• Register Y	0 0 0 0
• Register Z	× ×
• Stack pointer (SP)	1 1 1
• Operation source clock	On-chip oscillator (operating)
• Ceramic resonator circuit	Stop
• RC oscillation circuit	Stop
• Quartz-crystal oscillation circuit	Stop

"×" represents undefined.

Fig. 50 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.

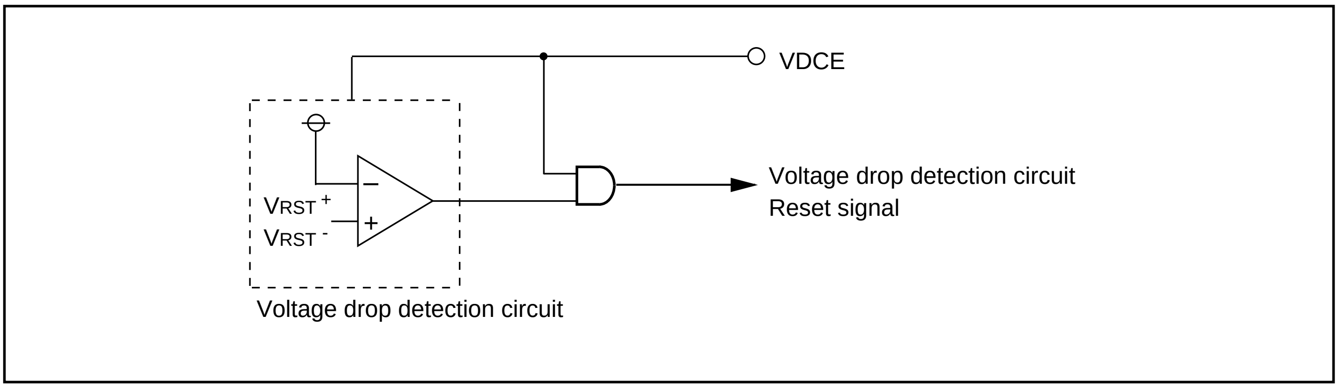


Fig. 51 Voltage drop detection reset circuit

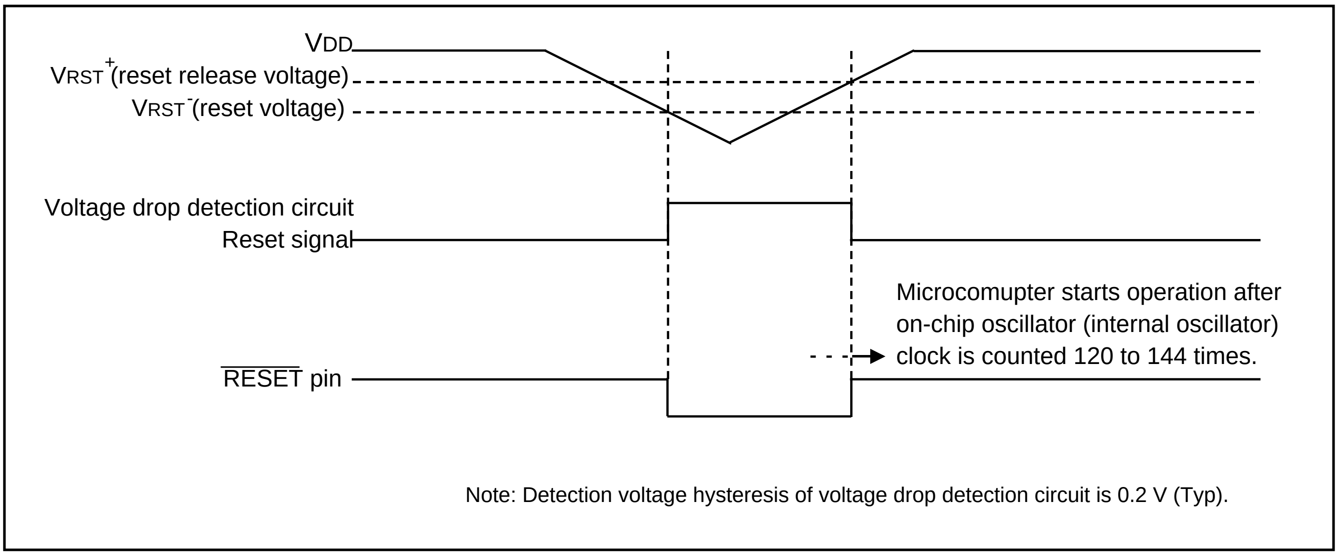


Fig. 52 Voltage drop detection circuit operation waveform

Table 17 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 53 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 18 Functions and states retained at RAM back-up

Function	RAM back-up
Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) (Note 2)	×
Contents of RAM	O
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

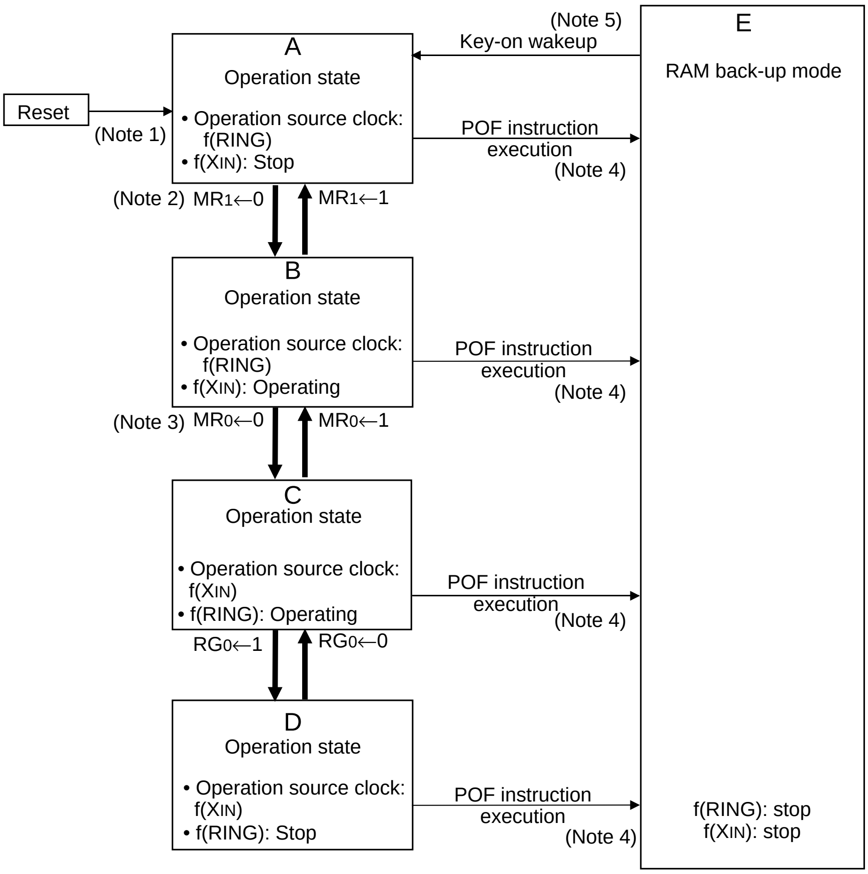
An external wakeup signal is used to return from the RAM back-up mode because the oscillation is stopped. Table 19 shows the return condition for each return source.

(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.

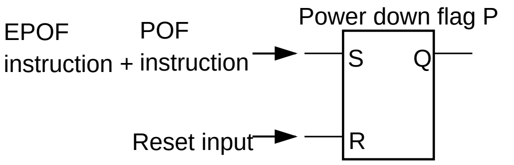
Table 19 Return source and return condition

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(X$_{IN}$) 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(X$_{IN}$) oscillation circuit by executing the CMCK, CRCK or CYCK instruction before clearing MR$_1$ to “0”.
MR$_1$ 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(X$_{IN}$) oscillation circuit is retained.

Fig. 53 State transition



● Set source ······· EPOF instruction + POF instruction

● Clear source······Reset input

Fig. 54 Set source and clear source of the P flag

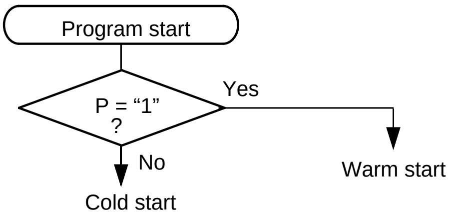


Fig. 55 Start condition identified example using the SNZP instruction

Table 20 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 21 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 56 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.

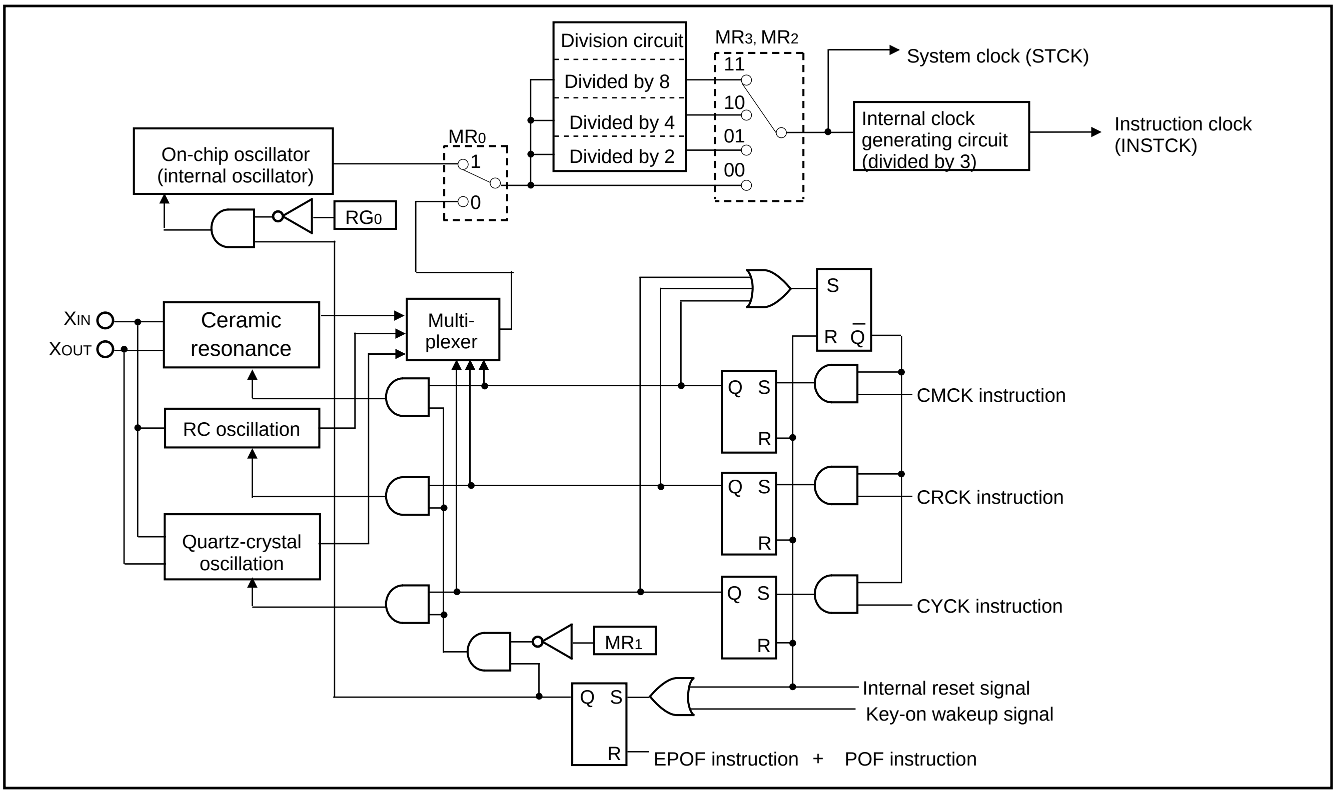


Fig. 56 Clock control circuit structure

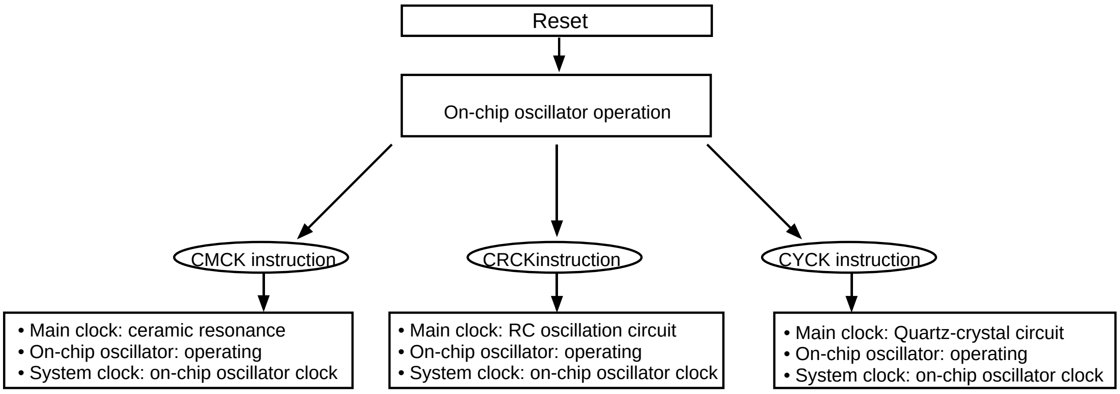
(1) Main clock generating circuit (f(XIN))

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.



- Set the main clock (f(XIN)) oscillation by bit 1 of register MR.
- Switch the system clock by bit 0 of register MR.
  Also, when system clock is switched after main clock oscillation is started, generate the oscillation stabilizing wait time by program if necessary.
- Set the on-chip oscillator clock oscillation by register RG.

Fig. 57 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 58).

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 59).

(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 60).

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

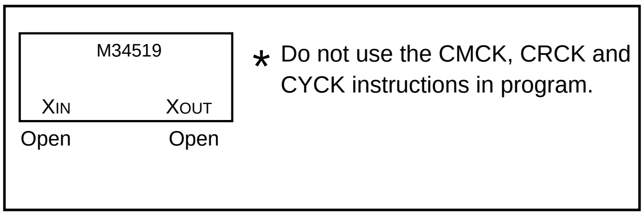
(5) Quartz-crystal oscillator

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 61).

(6) External clock

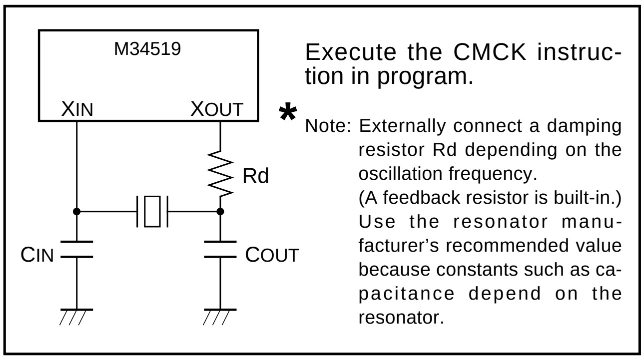
When the external clock signal for the main clock (f(XIN)) is used, connect the clock source to XIN pin and XOUT pin open. In program, after the CMCK instruction is executed, set main clock (f(XIN)) oscillation start to be enabled (MR1=0).

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.



* Do not use the CMCK, CRCK and CYCK instructions in program.

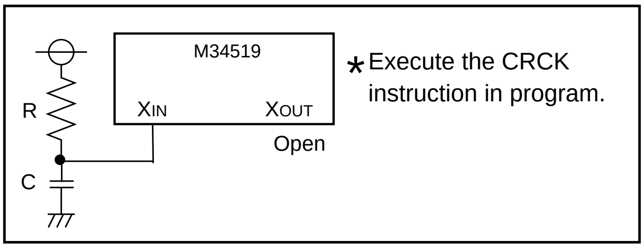
Fig. 58 Handling of XIN and XOUT when operating on-chip oscillator



* Execute the CMCK instruction in program.

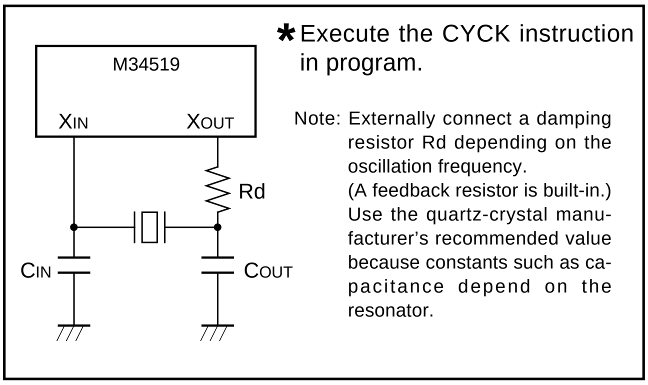
Note: Externally connect a damping resistor Rd depending on the oscillation frequency.
(A feedback resistor is built-in.)
Use the resonator manufacturer's recommended value because constants such as capacitance depend on the resonator.

Fig. 59 Ceramic resonator external circuit



* Execute the CRCK instruction in program.

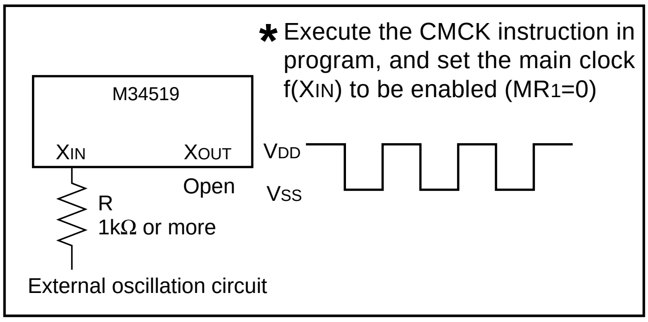
Fig. 60 External RC oscillation circuit



* Execute the CYCK instruction in program.

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

Fig. 61 External quartz-crystal circuit



* Execute the CMCK instruction in program, and set the main clock f(XIN) to be enabled (MR1=0)

Fig. 62 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 22 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)	
		0	1	Frequency divided by 2 mode	
MR_2		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

① Noise and latch-up prevention

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

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

In the One Time PROM version, CNVSS pin is also used as VPP pin. Accordingly, when using this pin, connect this pin to VSS through a resistor about 5 kΩ (connect this resistor to CNVSS/VPP pin as close as possible).

② Register initial values 1

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

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

③ Register initial values 2

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

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

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.

⑤ Multifunction

- The input/output of P30 and P31 can be used even when INT0 and INT1 are selected.
- The input of ports P20–P22 can be used even when SIN, SOUT and SCK are selected.
- The input/output of D6 can be used even when CNTR0 (input) is selected.
- The input of D6 can be used even when CNTR0 (output) is selected.
- The input/output of D7 can be used even when CNTR1 (input) is selected.
- The input of D7 can be used even when CNTR1 (output) is selected.

⑥ 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.

⑦ Timer count source

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

⑧ 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.

⑨ 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.

⑩ Writing to reload register R1, R3, R4H

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

⑪ Timer 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.

⑫ Watchdog timer

- The watchdog timer function is valid after system is released from reset. When not using the watchdog timer function, execute the DWDT instruction and the WRST instruction continuously, and clear the WEF flag to “0” to stop the watchdog timer function.
- The 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 Prescaler, Timer 1, Timer 2 and Timer 3 count start timing and count time when operation starts

Count starts from the first rising edge of the count source (2) after Prescaler, Timer 1, Timer 2 and Timer 3 operations start (1).

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

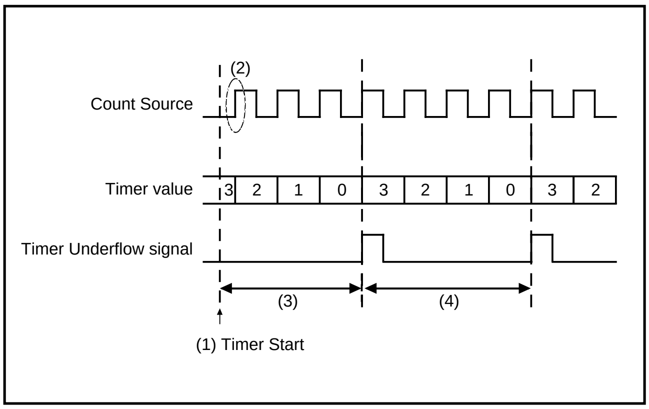


Fig. 63 Timer count start timing and count time when operation starts (Prescaler, Timer 1, Timer 2 and Timer 3)

⑭ Timer 4 count start timing and count time when operation starts

Count starts from the rising edge (2) after the first falling edge of the count source, after Timer 4 operations start (1).

Time to first underflow (3) is different from time among next underflow (4) by the timing to start the timer and count source operations after count starts.

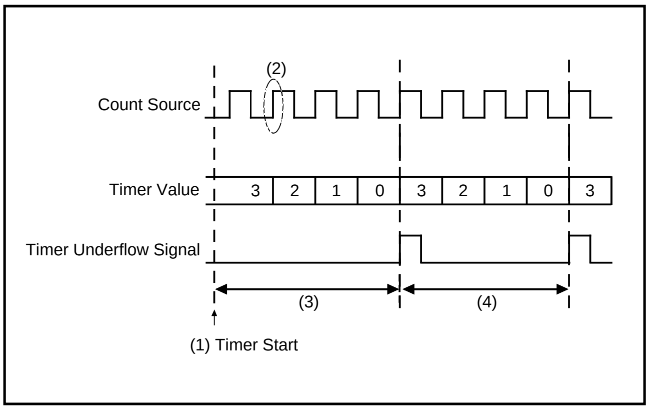


Fig. 64 Timer count start timing and count time when operation starts (Timer 4)

⑮ 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 65①) 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 65②).

Also, set the NOP instruction for the case when a skip is performed with the SNZT1 instruction (refer to Figure 65③).

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

   × : these bits are not used here.
```

Fig. 65 Period measurement circuit program example

⑯ $P3_0$/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 $P3_0$/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 66 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 66 ②).

  Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 66 ③).

```
      ⋮
LA      4       ; (×××02)
TV1A            ; The SNZ0 instruction is valid ...........①
LA      8       ; (1×××2)
TI1A            ; Control of INT0 pin input is changed
NOP             .......................................................... ②
SNZ0            ; The SNZ0 instruction is executed
                 (EXF0 flag cleared)
NOP             .......................................................... ③
      ⋮
                × : these bits are not used here.
```

Fig. 66 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 67①).

```
      ⋮
LA      0       ; (×××02)
TK2A            ; Input of INT0 key-on wakeup invalid ..①
DI
EPOF
POF             ; RAM back-up
      ⋮
      × : these bits are not used here.
```

Fig. 67 External 0 interrupt program example-2

❸ 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 68①) 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 68②).

  Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to Figure 68③).

```
      ⋮
LA      4       ; (×××02)
TV1A            ; The SNZ0 instruction is valid ...........①
LA      12      ; (×1××2)
TI1A            ; Interrupt valid waveform is changed
NOP             .......................................................... ②
SNZ0            ; The SNZ0 instruction is executed
                 (EXF0 flag cleared)
NOP             .......................................................... ③
      ⋮

      × : these bits are not used here.
```

Fig. 68 External 0 interrupt program example-3

⑰ $P3_1$/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 $P3_1$/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 69①) 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 69②).
  Also, set the NOP instruction for the case when a skip is performed with the SNZ1 instruction (refer to Figure 69③).

```
        ⋮
  LA     4      ; (✕✕0✕2)
  TV1A          ; The SNZ1 instruction is valid ...........①
  LA     8      ; (1✕✕✕2)
  TI2A          ; Control of INT1 pin input is changed
  NOP           .......................................................... ②
  SNZ1          ; The SNZ1 instruction is executed
                  (EXF1 flag cleared)
  NOP           .......................................................... ③
        ⋮

           ✕ : these bits are not used here.
```

Fig. 69 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 $I2_3$ = “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 70①).

```
        ⋮
  LA     0      ; (✕0✕✕2)
  TK2A          ; Input of INT1 key-on wakeup invalid .. ①
  DI
  EPOF
  POF           ; RAM back-up
        ⋮

           ✕ : these bits are not used here.
```

Fig. 70 External 1 interrupt program example-2

❸ 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 71①) 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 71②).
  Also, set the NOP instruction for the case when a skip is performed with the SNZ1 instruction (refer to Figure 71③).

```
        ⋮
  LA     4      ; (✕✕0✕2)
  TV1A          ; The SNZ1 instruction is valid ...........①
  LA     12     ; (✕1✕✕2)
  TI2A          ; Interrupt valid waveform is changed
  NOP           .......................................................... ②
  SNZ1          ; The SNZ1 instruction is executed
                  (EXF1 flag cleared)
  NOP           .......................................................... ③
        ⋮

           ✕ : these bits are not used here.
```

Fig. 71 External 1 interrupt program example-3

⑱ 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.

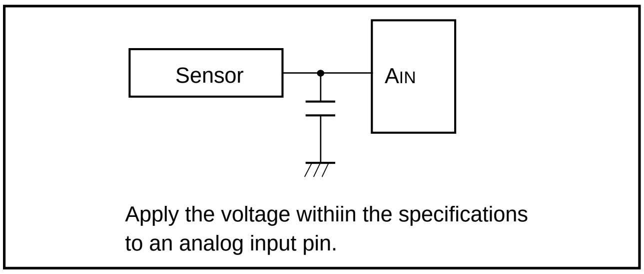
```
        ⋮
LA    8        ; (X0XX2)
TV2A           ; The SNZAD instruction is valid ........①
LA    0        ; (0XXX2)
TQ1A           ; Operation mode of A/D converter is
                 changed from comparator mode to A/D
                 conversion mode.
SNZAD
NOP
        ⋮          X : these bits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

Fig. 72 A/D converter program example-3

⑲ 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 73).

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 74. In addition, test the application products sufficiently.



Apply the voltage withiin the specifications to an analog input pin.

Fig. 73 Analog input external circuit example-1

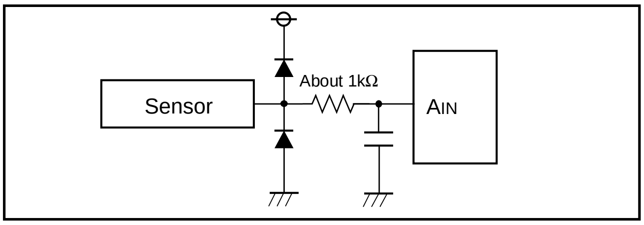


Fig. 74 Analog input external circuit example-2

⑳ 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.

㉑ Program counter

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

㉒ 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.

㉓ 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.

㉔ 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.

㉕ 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 (MR_1=0).

For this product, when RAM back-up mode and main clock (f(XIN)) stop (MR_1=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 (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 XIN pin in series to limit of current by competitive signal.

㉖ 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.

㉗ 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)	
		0	1	Frequency divided by 2 mode	
MR$_2$		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)	
		0	1	Prescaler output (ORCLK)	
W1$_0$		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)	
		0	1	Prescaler output (ORCLK)	
W2$_0$		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	
$W3_1$ $W3_0$	Timer 3 count source selection bits	$W3_1$	$W3_0$	Count source	
		0	0	PWM signal (PWMOUT)	
		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	
$W5_1$ $W5_0$	Signal for period measurement selection bits	$W5_1$	$W5_0$	Count source	
		0	0	On-chip oscillator (f(RING/16))	
		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
$J1_3$	Serial I/O synchronous clock selection bits	$J1_3$	$J1_2$	Synchronous clock	
		0	0	Instruction clock (INSTCK) divided by 8	
		0	1	Instruction clock (INSTCK) divided by 4	
$J1_2$		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	
		0	1	SCK, SOUT, $P2_2$ selected/$P2_0$, $P2_1$, SIN not selected	
$J1_0$		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	

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	AIN_0	
		0	0	1	AIN_1	
$Q1_1$		0	1	0	AIN_2	
		0	1	1	AIN_3	
		1	0	0	AIN_4	
$Q1_0$		1	0	1	AIN_5	
		1	1	0	AIN_6	
		1	1	1	AIN_7	

A/D control register Q2		at reset : 0000_2	at RAM back-up : state retained	R/W TAQ2/TQ2A
$Q2_3$	$P4_0$/AIN_4, $P4_1$/AIN_5, $P4_2$/AIN_6, $P4_3$/AIN_7 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	

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.

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)	$\overset{C}{\underset{x}{+}}$	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

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

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

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

Grouping	Mnemonic	Function
Interrupt operation	DI	(INTE) ← 0
	EI	(INTE) ← 1
	SNZ0	$V1_0$ = 0: (EXF0) = 1 ? After skipping, (EXF0) ← 0 $V1_0$ = 1: NOP
	SNZ1	$V1_1$ = 0: (EXF1) = 1 ? After skipping, (EXF1) ← 0 $V1_1$ = 1: NOP
	SNZI0	$I1_2$ = 1 : (INT0) = “H” ? $I1_2$ = 0 : (INT0) = “L” ?
	SNZI1	$I2_2$ = 1 : (INT1) = “H” ? $I2_2$ = 0 : (INT1) = “L” ?
	TAV1	(A) ← (V1)
	TV1A	(V1) ← (A)
	TAV2	(A) ← (V2)
	TV2A	(V2) ← (A)
	TAI1	(A) ← (I1)
	TI1A	(I1) ← (A)
	TAI2	(A) ← (I2)
	TI2A	(I2) ← (A)
Timer operation	TPAA	(PA_0) ← (A_0)
	TAW1	(A) ← (W1)
	TW1A	(W1) ← (A)
	TAW2	(A) ← (W2)
	TW2A	(W2) ← (A)
	TAW3	(A) ← (W3)
	TW3A	(W3) ← (A)

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
Timer operation	TAW4	(A) ← (W4)
	TW4A	(W4) ← (A)
	TAW5	(A) ← (W5)
	TW5A	(W5) ← (A)
	TAW6	(A) ← (W6)
	TW6A	(W6) ← (A)
	TABPS	(B) ← (TPS$_7$–TPS$_4$) (A) ← (TPS$_3$–TPS$_0$)
	TPSAB	(RPS$_7$–RPS$_4$) ← (B) (TPS$_7$–TPS$_4$) ← (B) (RPS$_3$–RPS$_0$) ← (A) (TPS$_3$–TPS$_0$) ← (A)
	TAB1	(B) ← (T1$_7$–T1$_4$) (A) ← (T1$_3$–T1$_0$)
	T1AB	(R1$_7$–R1$_4$) ← (B) (T1$_7$–T1$_4$) ← (B) (R1$_3$–R1$_0$) ← (A) (T1$_3$–T1$_0$) ← (A)
	TAB2	(B) ← (T2$_7$–T2$_4$) (A) ← (T2$_3$–T2$_0$)
	T2AB	(R2$_7$–R2$_4$) ← (B) (T2$_7$–T2$_4$) ← (B) (R2$_3$–R2$_0$) ← (A) (T2$_3$–T2$_0$) ← (A)
	TAB3	(B) ← (T3$_7$–T3$_4$) (A) ← (T3$_3$–T3$_0$)
	T3AB	(R3$_7$–R3$_4$) ← (B) (T3$_7$–T3$_4$) ← (B) (R3$_3$–R3$_0$) ← (A) (T3$_3$–T3$_0$) ← (A)
	TAB4	(B) ← (T4$_7$–T4$_4$) (A) ← (T4$_3$–T4$_0$)
	T4AB	(R4L$_7$–R4L$_4$) ← (B) (T4$_7$–T4$_4$) ← (B) (R4L$_3$–R4L$_0$) ← (A) (T4$_3$–T4$_0$) ← (A)

Grouping	Mnemonic	Function
Timer operation	T4HAB	(R4H$_7$–R4H$_4$) ← (B) (R4H$_3$–R4H$_0$) ← (A)
	TR1AB	(R1$_7$–R1$_4$) ← (B) (R1$_3$–R1$_0$) ← (A)
	TR3AB	(R3$_7$–R3$_4$) ← (B) (R3$_3$–R3$_0$) ← (A)
	T4R4L	(T4$_7$–T4$_4$) ← (R4L$_7$–R4L$_4$)
	SNZT1	V1$_2$ = 0: (T1F) = 1 ? After skipping, (T1F) ← 0 V1$_2$ = 1: NOP
	SNZT2	V1$_3$ = 0: (T2F) = 1 ? After skipping, (T2F) ← 0 V1$_3$ = 1: NOP
	SNZT3	V2$_0$ = 0: (T3F) = 1 ? After skipping, (T3F) ← 0 V2$_0$ = 1: NOP
	SNZT4	V2$_1$ = 0: (T4F) = 1 ? After skipping, (T4F) ← 0 V2$_1$ = 1: NOP
Input/Output operation	IAP0	(A) ← (P0)
	OP0A	(P0) ← (A)
	IAP1	(A) ← (P1)
	OP1A	(P1) ← (A)
	IAP2	(A$_2$–A$_0$) ← (P2$_2$–P2$_0$) (A$_3$) ← 0
	OP2A	(P2$_2$–P2$_0$) ← (A$_2$–A$_0$)
	IAP3	(A) ← (P3)
	OP3A	(P3) ← (A)
	IAP4	(A) ← (P4)
	OP4A	(P4) ← (A)
	IAP5	(A) ← (P5)
	OP5A	(P5) ← (A)
	IAP6	(A) ← (P6)
	OP6A	(P6) ← (A)

INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function
Input/Output operation	CLD	(D) ← 1
	RD	(D(Y)) ← 0 (Y) = 0 to 7
	SD	(D(Y)) ← 1 (Y) = 0 to 7
	SZD	(D(Y)) = 0 ? (Y) = 0 to 7
	TAPU0	(A) ← (PU0)
	TPU0A	(PU0) ← (A)
	TAPU1	(A) ← (PU1)
	TPU1A	(PU1) ← (A)
	TAK0	(A) ← (K0)
	TK0A	(K0) ← (A)
	TAK1	(A) ← (K1)
	TK1A	(K1) ← (A)
	TAK2	(A) ← (K2)
	TK2A	(K2) ← (A)
	TFR0A	(FR0) ← (A)
	TFR1A	(FR1) ← (A)
	TFR2A	(FR2) ← (A)
	TFR3A	(FR3) ← (A)
Clock operation	CMCK	Ceramic resonator selected
	CRCK	RC oscillator selected
	CYCK	Quartz-crystal oscillator selected
	TRGA	(RG_0) ← (A_0)
	TAMR	(A) ← (MR)
	TMRA	(MR) ← (A)

Grouping	Mnemonic	Function
Serial I/O operation	TABSI	(B) ← (SI_7–SI_4) (A) ← (SI_3–SI_0)
	TSIAB	(SI_7–SI_4) ← (B) (SI_3–SI_0) ← (A)
	SST	(SIOF) ← 0 Serial I/O starting
	SNZSI	$V2_3$=0: (SIOF)=1? After skipping, (SIOF) ← 0 $V2_3$=1: NOP
	TAJ1	(A) ← (J1)
	TJ1A	(J1) ← (A)
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)
	TALA	(A_3, A_2) ← (AD_1, AD_0) (A_1, A_0) ← 0
	TADAB	(AD_7–AD_4) ← (B) (AD_3–AD_0) ← (A)
	ADST	(ADF) ← 0 A/D conversion starting
	SNZAD	$V2_1$ = 0: (ADF) = 1 ? After skipping, (ADF) ← 0 $V2_1$=1: NOP
	TAQ1	(A) ← (Q1)
	TQ1A	(Q1) ← (A)
	TAQ2	(A) ← (Q2)
	TQ2A	(Q2) ← (A)
	TAQ3	(A) ← (Q3)
	TQ3A	(Q3) ← (A)

INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function
Other operation	NOP	(PC) ← (PC) + 1
	POF	Transition to RAM back-up mode
	EPOF	POF instruction valid
	SNZP	(P) = 1 ?
	DWDT	Stop of watchdog timer function enabled
	WRST	(WDF1) = 1 ? After skipping, (WDF1) ← 0
	SRST	System reset occurrence

MACHINE INSTRUCTIONS (INDEX BY ALPHABET)

A n (Add n and accumulator)

Instruction code	D_9									D_0						Number of words	Number of cycles	Flag CY	Skip condition
	0	0	0	1	1	0	n	n	n	n	$_2$	0	6	n	$_{16}$	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	D_9									D_0						Number of words	Number of cycles	Flag CY	Skip condition
	1	0	1	0	0	1	1	1	1	1	$_2$	2	9	F	$_{16}$	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	D_9									D_0						Number of words	Number of cycles	Flag CY	Skip condition
	0	0	0	0	0	0	1	0	1	0	$_2$	0	0	A	$_{16}$	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	D_9									D_0						Number of words	Number of cycles	Flag CY	Skip condition
	0	0	0	0	0	0	1	0	1	1	$_2$	0	0	B	$_{16}$	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

D_9									D_0					
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

D_9									D_0					
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

D_9									D_0					
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

D_9									D_0					
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

D_9									D_0					
0	1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$	1	a	a	$_{16}$

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

Operation:
$(SP) \leftarrow (SP) + 1$
$(SK(SP)) \leftarrow (PC)$
$(PC_H) \leftarrow 2$
$(PC_L) \leftarrow a_6–a_0$

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

D_9									D_0					
0	0	1	1	0	p_4	p_3	p_2	p_1	p_0	$_2$	0	C +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:
$(SP) \leftarrow (SP) + 1$
$(SK(SP)) \leftarrow (PC)$
$(PC_H) \leftarrow p$
$(PC_L) \leftarrow a_6–a_0$

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

D_9									D_0					
0	0	0	0	1	1	0	0	0	0	$_2$	0	3	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:
$(SP) \leftarrow (SP) + 1$
$(SK(SP)) \leftarrow (PC)$
$(PC_H) \leftarrow p$
$(PC_L) \leftarrow (DR_2–DR_0, A_3–A_0)$

Grouping: Subroutine call operation

Description: 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.

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

D_9									D_0					
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: $(D) \leftarrow 1$

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: $(A) \leftarrow \overline{(A)}$

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(X_{IN}).

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(X_{IN}).

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(X_{IN}).

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

DEY (DEcrement register Y)

Instruction code

D_9									D_0						
0	0	0	0	0	1	0	1	1	1	$_2$	0	1	7	$_{16}$	

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

D_9									D_0					
0	0	0	0	0	0	0	1	0	0	$_2$	0	0	4	$_{16}$

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

D_9									D_0					
1	0	1	0	0	1	1	1	0	0	$_2$	2	9	C	$_{16}$

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

D_9									D_0					
0	0	0	0	0	0	0	1	0	1	$_2$	0	0	5	$_{16}$

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						Number of words	Number of cycles	Flag CY	Skip condition
	0	0	0	1	0	1	1	0	1	1	2	0	5	B	16	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						Number of words	Number of cycles	Flag CY	Skip condition
	1	0	0	1	1	0	0	0	0	0	2	2	6	0	16	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						Number of words	Number of cycles	Flag CY	Skip condition
	1	0	0	1	1	0	0	0	0	1	2	2	6	1	16	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						Number of words	Number of cycles	Flag CY	Skip condition
	1	0	0	1	1	0	0	0	1	0	2	2	6	2	16	1	1	–	–

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

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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 0 0 0 0 0 1 0 0 1 1 $_2$ 0 1 3 $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 0 0 0 1 1 1 n n n n $_2$ 0 7 n $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 1 1 x_3 x_2 x_1 x_0 y_3 y_2 y_1 y_0 $_2$ 3 x y $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 0 0 0 1 0 0 1 0 z_1 z_0 $_2$ 0 4 8+z $_{16}$	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 2 3 $_{16}$

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 2 4 $_{16}$

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 2 5 $_{16}$

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 2 6 $_{16}$

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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 $0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1_2$ $0\ 1\ 9_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 $0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0_2$ $0\ 0\ 2_{16}$	1	1	–	–

Operation: Transition to RAM back-up mode

Grouping: Other operation

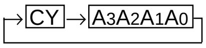
Description: Puts the system in RAM back-up state by executing the POF instruction after executing the EPOF instruction.

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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 $0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 1_2$ $0\ 1\ D_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 $0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ j\ j_2$ $0\ 4\ \begin{matrix}C\\+j\end{matrix}_{16}$	1	1	–	–

Operation: $(M_j(DP)) \leftarrow 0$
$j = 0$ 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						Number of words	Number of cycles	Flag CY	Skip condition
	0	0	0	0	0	0	0	1	1	0	$_2$	0	0	6	$_{16}$	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						Number of words	Number of cycles	Flag CY	Skip condition
	0	0	0	0	0	1	0	1	0	0	$_2$	0	1	4	$_{16}$	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						Number of words	Number of cycles	Flag CY	Skip condition
	0	0	0	1	0	0	0	1	0	0	$_2$	0	4	4	$_{16}$	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						Number of words	Number of cycles	Flag CY	Skip condition
	0	0	0	1	0	0	0	1	1	0	$_2$	0	4	6	$_{16}$	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 D_9 … D_0

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

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.

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

SEAM (Skip Equal, Accumulator with Memory)

Instruction code D_9 … D_0

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 D_9 … D_0

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 D_9 … D_0

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	D_9 … D_0		Number of words	Number of cycles	Flag CY	Skip condition
	$1\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1_2$	$2\ 8\ 7_{16}$	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	D_9 … D_0		Number of words	Number of cycles	Flag CY	Skip condition
	$0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0_2$	$0\ 3\ A_{16}$	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	D_9 … D_0		Number of words	Number of cycles	Flag CY	Skip condition
	$0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 1_2$	$0\ 3\ B_{16}$	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	D_9 … D_0		Number of words	Number of cycles	Flag CY	Skip condition
	$0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1_2$	$0\ 0\ 3_{16}$	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 (D_9 … D_0)	Number of words	Number of cycles	Flag CY	Skip condition
$1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0_2$ $2\ 8\ 8_{16}$	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 (D_9 … D_0)	Number of words	Number of cycles	Flag CY	Skip condition
$1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0_2$ $2\ 8\ 0_{16}$	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 (D_9 … D_0)	Number of words	Number of cycles	Flag CY	Skip condition
$1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1_2$ $2\ 8\ 1_{16}$	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 (D_9 … D_0)	Number of words	Number of cycles	Flag CY	Skip condition
$1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0_2$ $2\ 8\ 2_{16}$	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

D_9									D_0					
1	0	1	0	0	0	0	0	1	1	$_2$	2	8	3	$_{16}$

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) $\leftarrow$ 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

D_9									D_0					
0	0	0	0	0	0	0	0	0	1	$_2$	0	0	1	$_{16}$

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

D_9									D_0					
1	0	1	0	0	1	1	1	1	0	$_2$	2	9	E	$_{16}$

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

Operation:
(SIOF) $\leftarrow$ 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

D_9									D_0					
0	0	0	0	1	0	0	0	j	j	$_2$	0	2	j	$_{16}$

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	D_9 — D_0		Number of words	Number of cycles	Flag CY	Skip condition
	0 0 0 0 1 0 1 1 1 1 $_2$	0 2 F $_{16}$	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	D_9 — D_0		Number of words	Number of cycles	Flag CY	Skip condition
	0 0 0 0 1 0 0 1 0 0 $_2$	0 2 4 $_{16}$	2	2	–	(D(Y)) = 0 (Y) = 0 to 7
	0 0 0 0 1 0 1 0 1 1 $_2$	0 2 B $_{16}$				

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	D_9 — D_0		Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 1 1 0 0 0 0 $_2$	2 3 0 $_{16}$	1	1	–	–

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

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	D_9 — D_0		Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 1 1 0 0 0 1 $_2$	2 3 1 $_{16}$	1	1	–	–

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

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

D_9									D_0	
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:

$(T3_7–T3_4) \leftarrow (B)$
$(R3_7–R3_4) \leftarrow (B)$
$(T3_3–T3_0) \leftarrow (A)$
$(R3_3–R3_0) \leftarrow (A)$

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

D_9									D_0	
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:

$(T4_7–T4_4) \leftarrow (B)$
$(R4L_7–R4L_4) \leftarrow (B)$
$(T4_3–T4_0) \leftarrow (A)$
$(R4L_3–R4L_0) \leftarrow (A)$

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

D_9									D_0	
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:

$(R4H_7–R4H_4) \leftarrow (B)$
$(R4H_3–R4H_0) \leftarrow (A)$

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

D_9									D_0	
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:

$(T4_7–T4_4) \leftarrow (R4L_7–R4L_4)$
$(T4_3–T4_0) \leftarrow (R4L_3–R4L_0)$

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

D_9									D_0					
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

D_9									D_0					
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

D_9									D_0					
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

D_9									D_0					
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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 1 0 0 1 1 1 0 0 1 1$_2$ 2 7 3$_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 1 0 0 1 1 1 1 0 0 1$_2$ 2 7 9$_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 0 0 0 0 1 0 1 0 1 0$_2$ 0 2 A$_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 0 0 1 0 p_5 p_4 p_3 p_2 p_1 $p_0$$_2$ 0 8+p p$_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 D_0 $1\,0\,0\,1\,1\,1\,0\,1\,0\,1_2$ $2\,7\,5_{16}$	1	1	–	–

Operation: $(B) \leftarrow (TPS_7–TPS_4)$
$(A) \leftarrow (TPS_3–TPS_0)$

Grouping: Timer operation

Description: Transfers the high-order 4 bits (TPS_7–TPS_4) of prescaler to register B, and transfers the low-order 4 bits (TPS_3–TPS_0) of prescaler to register A.

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

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 D_0 $1\,0\,0\,1\,1\,1\,1\,0\,0\,0_2$ $2\,7\,8_{16}$	1	1	–	–

Operation: $(B) \leftarrow (SI_7–SI_4)$
$(A) \leftarrow (SI_3–SI_0)$

Grouping: Serial I/O operation

Description: Transfers the high-order 4 bits (SI_7–SI_4) of serial I/O register SI to register B, and transfers the low-order 4 bits (SI_3–SI_0) of serial I/O register SI to register A.

TAD (Transfer data to Accumulator from register D)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 D_0 $0\,0\,0\,1\,0\,1\,0\,0\,0\,1_2$ $0\,5\,1_{16}$	1	1	–	–

Operation: $(A_2–A_0) \leftarrow (DR_2–DR_0)$
$(A_3) \leftarrow 0$

Grouping: Register to register transfer

Description: Transfers the contents of register D to the low-order 3 bits (A_2–A_0) of register A.

Note: When this instruction is executed, “0” is stored to the bit 3 (A_3) of register A.

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

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 D_0 $1\,0\,0\,0\,1\,1\,1\,0\,0\,1_2$ $2\,3\,9_{16}$	1	1	–	–

Operation: $(AD_7–AD_4) \leftarrow (B)$
$(AD_3–AD_0) \leftarrow (A)$

Grouping: A/D conversion operation

Description: In the A/D conversion mode ($Q1_3 = 0$), this instruction is equivalent to the NOP instruction.
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)

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	D_9									D_0					
	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	D_9									D_0					
	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	D_9									D_0					
	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:
$(A_3, A_2) \leftarrow (AD_1, AD_0)$
$(A_1, A_0) \leftarrow 0$

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	D_9									D_0					
	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 1 0 0 1 0 1 0 0 1 0$_2$ 2 5 2$_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 1 0 0 1 0 1 0 1 1 1$_2$ 2 5 7$_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 1 0 0 1 0 1 1 1 1 0$_2$ 2 5 E$_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0 1 0 0 1 0 0 0 1 0 0$_2$ 2 4 4$_{16}$	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

D_9									D_0					
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

D_9									D_0					
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

D_9									D_0					
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

D_9									D_0					
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	D_9 … D_0		Number of words	Number of cycles	Flag CY	Skip condition
	0 0 0 1 0 1 0 1 0 1 $_2$	0 5 5 $_{16}$	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	D_9 … D_0		Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 0 1 0 1 1 $_2$	2 4 B $_{16}$	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	D_9 … D_0		Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 0 1 1 0 0 $_2$	2 4 C $_{16}$	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	D_9 … D_0		Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 1 0 0 1 1 0 1 $_2$	2 4 D $_{16}$	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	$_2$	2	4	E	$_{16}$

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	$_2$	2	4	F	$_{16}$

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	$_2$	2	5	0	$_{16}$

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	$_2$	0	5	2	$_{16}$

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	D_9									D_0			
	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	D_9									D_0			
	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: $(A_1, A_0) \leftarrow (Z_1, Z_0)$
$(A_3, A_2) \leftarrow 0$

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	D_9									D_0			
	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	D_9									D_0			
	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: $(DR_2–DR_0) \leftarrow (A_2–A_0)$

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

D_9									D_0					
0	0	0	0	0	1	1	0	1	0	$_2$	0	1	A	$_{16}$

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

Operation: $(E_7–E_4) \leftarrow (B)$
$(E_3–E_0) \leftarrow (A)$

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

D_9									D_0					
1	0	0	0	1	0	1	0	0	0	$_2$	2	2	8	$_{16}$

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

Operation: $(FR0) \leftarrow (A)$

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

D_9									D_0					
1	0	0	0	1	0	1	0	0	1	$_2$	2	2	9	$_{16}$

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

Operation: $(FR1) \leftarrow (A)$

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

D_9									D_0					
1	0	0	0	1	0	1	0	1	0	$_2$	2	2	A	$_{16}$

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

Operation: $(FR2) \leftarrow (A)$

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	D_9									D_0					
	1	0	0	0	1	0	1	0	1	1	$_2$	2	2	B	$_{16}$

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	D_9									D_0					
	1	0	0	0	0	1	0	1	1	1	$_2$	2	1	7	$_{16}$

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	D_9									D_0					
	1	0	0	0	0	1	1	0	0	0	$_2$	2	1	8	$_{16}$

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	D_9									D_0					
	1	0	0	0	0	0	0	0	1	0	$_2$	2	0	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	$_2$	2	1	B	$_{16}$

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	$_2$	2	1	4	$_{16}$

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	$_2$	2	1	5	$_{16}$

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	$_2$	2	B	j	$_{16}$

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	D_9 … D_0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 0 1 0 1 1 0$_2$ 2 1 6$_{16}$	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	D_9 … D_0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 1 0 1 0 1 0 1 0$_2$ 2 A A$_{16}$	1	1	–	–

Operation: $(PA_0) \leftarrow (A_0)$

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	D_9 … D_0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 1 1 0 1 0 1$_2$ 2 3 5$_{16}$	1	1	–	–

Operation:
$(RPS_7–RPS_4) \leftarrow (B)$
$(TPS_7–TPS_4) \leftarrow (B)$
$(RPS_3–RPS_0) \leftarrow (A)$
$(TPS_3–TPS_0) \leftarrow (A)$

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	D_9 … D_0	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 0 0 1 0 1 1 0 1$_2$ 2 2 D$_{16}$	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	2	E $_{16}$

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	$_2$	2	0	4 $_{16}$

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	$_2$	2	0	5 $_{16}$

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	$_2$	2	0	6 $_{16}$

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

D_9									D_0			
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:
($R1_7$–$R1_4$) ← (B)
($R1_3$–$R1_0$) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits ($R1_7$–$R1_4$) of reload register R1, and the contents of register A to the low-order 4 bits ($R1_3$–$R1_0$) of reload register R1.

TR3AB (Transfer data to register R3 from Accumulator and register B)

Instruction code

D_9									D_0			
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:
($R3_7$–$R3_4$) ← (B)
($R3_3$–$R3_0$) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits ($R3_7$–$R3_4$) of reload register R3, and the contents of register A to the low-order 4 bits ($R3_3$–$R3_0$) of reload register R3.

TRGA (Transfer data to register RG from Accumulator)

Instruction code

D_9									D_0			
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: (RG_0) ← (A_0)

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

D_9									D_0			
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:
(SI_7–SI_4) ← (B)
(SI_3–SI_0) ← (A)

Grouping: Serial I/O operation

Description: Transfers the contents of register B to the high-order 4 bits (SI_7–SI_4) of serial I/O register SI, and transfers the contents of register A to the low-order 4 bits (SI_3–SI_0) 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	$_2$	0	0	C	$_{16}$

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	$_2$	2	A	0	$_{16}$

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	$_2$	2	D	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 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	$_2$	2	F	j	$_{16}$

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	Number of words	Number of cycles	Flag CY	Skip condition
	1 0 1 1 1 0 j j j j $_2$ 2 E j $_{16}$	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)

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

Skip condition	Carry flag CY	Datailed description
–	–	Transfers the contents of register B to register A.
–	–	Transfers the contents of register A to register B.
–	–	Transfers the contents of register Y to register A.
–	–	Transfers the contents of register A to register Y.
–	–	Transfers the contents of 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)

Parameter / Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			
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	$\rightarrow \boxed{CY} \rightarrow \boxed{A_3A_2A_1A_0} \rightarrow$
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 (continued)

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			
Branch operation	B a	0	1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 8 a +a	1	1	$(PC_L) \leftarrow a_6–a_0$
	BL p, a	0	0	1	1	1	p_4	p_3	p_2	p_1	p_0	0 E p +p	2	2	$(PC_H) \leftarrow p$ (Note) $(PC_L) \leftarrow a_6–a_0$
		1	0	p_5	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2 p a +a			
	BLA p	0	0	0	0	0	1	0	0	0	0	0 1 0	2	2	$(PC_H) \leftarrow p$ (Note) $(PC_L) \leftarrow (DR_2–DR_0, A_3–A_0)$
		1	0	p_5	p_4	0	0	p_3	p_2	p_1	p_0	2 p p			
Subroutine operation	BM a	0	1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 a a	1	1	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow 2$ $(PC_L) \leftarrow a_6–a_0$
	BML p, a	0	0	1	1	0	p_4	p_3	p_2	p_1	p_0	0 C p +p	2	2	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow p$ (Note) $(PC_L) \leftarrow a_6–a_0$
		1	0	p_5	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2 p a +a			
	BMLA p	0	0	0	0	1	1	0	0	0	0	0 3 0	2	2	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PC_H) \leftarrow p$ (Note) $(PC_L) \leftarrow (DR_2–DR_0,A_3–A_0)$
		1	0	p_5	p_4	0	0	p_3	p_2	p_1	p_0	2 p p			
Return operation	RTI	0	0	0	1	0	0	0	1	1	0	0 4 6	1	1	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) – 1$
	RT	0	0	0	1	0	0	0	1	0	0	0 4 4	1	2	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) – 1$
	RTS	0	0	0	1	0	0	0	1	0	1	0 4 5	1	2	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) – 1$

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											Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			
Interrupt operation	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.

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D$_9$	D$_8$	D$_7$	D$_6$	D$_5$	D$_4$	D$_3$	D$_2$	D$_1$	D$_0$	Hexadecimal notation			
Timer operation	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) ← (TPS$_7$–TPS$_4$) (A) ← (TPS$_3$–TPS$_0$)
	TPSAB	1	0	0	0	1	1	0	1	0	1	2 3 5	1	1	(RPS$_7$–RPS$_4$) ← (B) (TPS$_7$–TPS$_4$) ← (B) (RPS$_3$–RPS$_0$) ← (A) (TPS$_3$–TPS$_0$) ← (A)
	TAB1	1	0	0	1	1	1	0	0	0	0	2 7 0	1	1	(B) ← (T1$_7$–T1$_4$) (A) ← (T1$_3$–T1$_0$)
	T1AB	1	0	0	0	1	1	0	0	0	0	2 3 0	1	1	(R1$_7$–R1$_4$) ← (B) (T1$_7$–T1$_4$) ← (B) (R1$_3$–R1$_0$) ← (A) (T1$_3$–T1$_0$) ← (A)
	TAB2	1	0	0	1	1	1	0	0	0	1	2 7 1	1	1	(B) ← (T2$_7$–T2$_4$) (A) ← (T2$_3$–T2$_0$)
	T2AB	1	0	0	0	1	1	0	0	0	1	2 3 1	1	1	(R2$_7$–R2$_4$) ← (B) (T2$_7$–T2$_4$) ← (B) (R2$_3$–R2$_0$) ← (A) (T2$_3$–T2$_0$) ← (A)
	TAB3	1	0	0	1	1	1	0	0	1	0	2 7 2	1	1	(B) ← (T3$_7$–T3$_4$) (A) ← (T3$_3$–T3$_0$)
	T3AB	1	0	0	0	1	1	0	0	1	0	2 3 2	1	1	(R3$_7$–R3$_4$) ← (B) (T3$_7$–T3$_4$) ← (B) (R3$_3$–R3$_0$) ← (A) (T3$_3$–T3$_0$) ← (A)
	TAB4	1	0	0	1	1	1	0	0	1	1	2 7 3	1	1	(B) ← (T4$_7$–T4$_4$) (A) ← (T4$_3$–T4$_0$)
	T4AB	1	0	0	0	1	1	0	0	1	1	2 3 3	1	1	(R4L$_7$–R4L$_4$) ← (B) (T4$_7$–T4$_4$) ← (B) (R4L$_3$–R4L$_0$) ← (A) (T4$_3$–T4$_0$) ← (A)
	T4HAB	1	0	0	0	1	1	0	1	1	1	2 3 7	1	1	(R4H$_7$–R4H$_4$) ← (B) (R4H$_3$–R4H$_0$) ← (A)
	TR1AB	1	0	0	0	1	1	1	1	1	1	2 3 F	1	1	(R1$_7$–R1$_4$) ← (B) (R1$_3$–R1$_0$) ← (A)
	TR3AB	1	0	0	0	1	1	1	0	1	1	2 3 B	1	1	(R3$_7$–R3$_4$) ← (B) (R3$_3$–R3$_0$) ← (A)
	T4R4L	1	0	1	0	0	1	0	1	1	1	2 9 7	1	1	(T4$_7$–T4$_0$) ← (R4L$_7$–R4L$_0$)

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.
–	–	

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			
Timer operation	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											Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			
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											Number of words	Number of cycles	Function
		D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			
A/D conversion operation	TABAD	1	0	0	1	1	1	1	0	0	1	2 7 9	1	1	$Q1_3 = 0$: $(B) \leftarrow (AD_9–AD_6)$ $(A) \leftarrow (AD_5–AD_2)$ $Q1_3 = 1$: $(B) \leftarrow (AD_7–AD_4)$ $(A) \leftarrow (AD_3–AD_0)$
	TALA	1	0	0	1	0	0	1	0	0	1	2 4 9	1	1	$(A_3, A_2) \leftarrow (AD_1, AD_0)$ $(A_1, A_0) \leftarrow 0$
	TADAB	1	0	0	0	1	1	1	0	0	1	2 3 9	1	1	$(AD_7–AD_4) \leftarrow (B)$ $(AD_3–AD_0) \leftarrow (A)$
	ADST	1	0	1	0	0	1	1	1	1	1	2 9 F	1	1	$(ADF) \leftarrow 0$ 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, $(ADF) \leftarrow 0$ $V2_2 = 1$: NOP
	TAQ1	1	0	0	1	0	0	0	1	0	0	2 4 4	1	1	$(A) \leftarrow (Q1)$
	TQ1A	1	0	0	0	0	0	0	1	0	0	2 0 4	1	1	$(Q1) \leftarrow (A)$
	TAQ2	1	0	0	1	0	0	0	1	0	1	2 4 5	1	1	$(A) \leftarrow (Q2)$
	TQ2A	1	0	0	0	0	0	0	1	0	1	2 0 5	1	1	$(Q2) \leftarrow (A)$
	TAQ3	1	0	0	1	0	0	0	1	1	0	2 4 6	1	1	$(A) \leftarrow (Q3)$
	TQ3A	1	0	0	0	0	0	0	1	1	0	2 0 6	1	1	$(Q3) \leftarrow (A)$
Other operation	NOP	0	0	0	0	0	0	0	0	0	0	0 0 0	1	1	$(PC) \leftarrow (PC) + 1$
	POF	0	0	0	0	0	0	0	0	1	0	0 0 2	1	1	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, $(WDF1) \leftarrow 0$
	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

	D_9–D_4	000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111	010000 010111	011000 011111
D_3–D_0	Hex. notation	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10–17	18–1F
0000	0	NOP	BLA	SZB 0	BMLA	–	TASP	A 0	LA 0	TABP 0	TABP 16	TABP 32	TABP 48*	BML	BML	BL	BL	BM	B
0001	1	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. D_3–D_0 show the low-order 4 bits of the machine language code, and D_9–D_4 show the high-order 6 bits of the machine language code. The hexadecimal representation of the code is also provided. There are one-word instructions and two-word instructions, but only the first word of each instruction is shown. Do not use code marked "–."

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

	The second word
BL	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)

D_9–D_4		100000	100001	100010	100011	100100	100101	100110	100111	101000	101001	101010	101011	101100	101101	101110	101111	110000 111111
D_3–D_0	Hex. notation	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30–3F
0000	0	–	TW3A	OP0A	T1AB	–	TAW6	IAP0	TAB1	SNZT1	–	WRST	TMA 0	TAM 0	XAM 0	XAMI 0	XAMD 0	LXY
0001	1	–	TW4A	OP1A	T2AB	–	–	IAP1	TAB2	SNZT2	–	–	TMA 1	TAM 1	XAM 1	XAMI 1	XAMD 1	LXY
0010	2	TJ1A	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. D_3–D_0 show the low-order 4 bits of the machine language code, and D_9–D_4 show the high-order 6 bits of the machine language code. The hexadecimal representation of the code is also provided. There are one-word instructions and two-word instructions, but only the first word of each instruction is shown. Do not use code marked "–."

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

	The second word
BL	1p paaa aaaa
BML	1p paaa aaaa
BLA	1p pp00 pppp
BMLA	1p pp00 pppp
SEA	00 0111 nnnn
SZD	00 0010 1011

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, $\overline{RESET}$, XIN, 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 AIN0–AIN7			–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 XOUT			–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

Recommended operating conditions 1

(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.	
VDD	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	
VDD	Supply voltage (when RC oscillation is used)	f(STCK) ≤ 4.4 MHz		2.7		5.5	V
VDD	Supply voltage (when quartz-crystal oscillator is used)	Mask ROM version	f(XIN) ≤ 50 kHz	2.0		5.5	V
		One Time PROM version	f(XIN) ≤ 50 kHz	2.5		5.5	V
VRAM	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
VSS	Supply voltage				0		V
VIH	“H” level input voltage	P0, P1, P2, P3, P4, P5, P6, D0–D7, VDCE, XIN		0.8VDD		VDD	V
VIH	“H” level input voltage	$\overline{\text{RESET}}$		0.85VDD		VDD	V
VIH	“H” level input voltage	SCK, SIN, CNTR0, CNTR1, INT0, INT1		0.85VDD		VDD	V
VIL	“L” level input voltage	P0, P1, P2, P3, P4, P5, P6, D0–D7, VDCE, XIN		0		0.2VDD	V
VIL	“L” level input voltage	$\overline{\text{RESET}}$		0		0.3VDD	V
VIL	“L” level input voltage	SCK, SIN, CNTR0, CNTR1, INT0, INT1		0		0.15VDD	V
IOH(peak)	“H” level peak output current	P0, P1, P5, D0–D7 CNTR0, CNTR1	VDD = 5 V			–20	mA
			VDD = 3 V			–10	
IOH(avg)	“H” level average output current (Note)	P0, P1, P5, D0–D7 CNTR0, CNTR1	VDD = 5 V			–10	mA
			VDD = 3 V			–5	
IOL(peak)	“L” level peak output current	P0, P1, P2, P4, P5, P6 SCK, SOUT	VDD = 5 V			24	mA
			VDD = 3 V			12	
IOL(peak)	“L” level peak output current	P3, $\overline{\text{RESET}}$	VDD = 5 V			10	mA
			VDD = 3 V			4	
IOL(peak)	“L” level peak output current	D0–D5	VDD = 5 V			24	mA
			VDD = 3 V			12	
IOL(peak)	“L” level peak output current	D6, D7 CNTR0, CNTR1	VDD = 5 V			40	mA
			VDD = 3 V			30	
IOL(avg)	“L” level average output current (Note)	P0, P1, P2, P4, P5, P6 SCK, SOUT	VDD = 5 V			12	mA
			VDD = 3 V			6	
IOL(avg)	“L” level average output current (Note)	P3, $\overline{\text{RESET}}$	VDD = 5 V			5	mA
			VDD = 3 V			2	
IOL(avg)	“L” level average output current (Note)	D0–D5	VDD = 5 V			15	mA
			VDD = 3 V			7	
IOL(avg)	“L” level average output current (Note)	D6, D7 CNTR0, CNTR1	VDD = 5 V			30	mA
			VDD = 3 V			15	
ΣIOH(avg)	“H” level total average current	P5, D0–D7, CNTR0, CNTR1				–60	mA
		P0, P1				–60	
ΣIOL(avg)	“L” level total average current	P2, P5, D0–D7, $\overline{\text{RESET}}$, CNTR0, CNTR1				80	mA
		P0, P1, P3, P4, P6				80	

Note: The average output current is the average value during 100 ms.

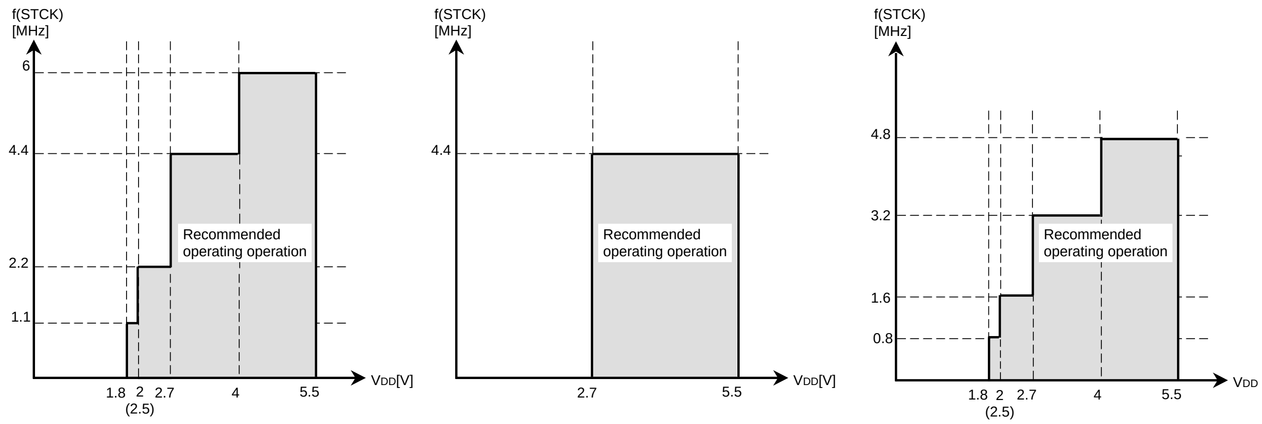
Recommended operating conditions 2

(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	Conditions			Limits			Unit
					Min.	Typ.	Max.	
$f(X_{IN})$	Oscillation frequency (with a ceramic resonator)	Mask ROM version	Through mode	V_{DD} = 4.0 to 5.5 V			6.0	MHz
				V_{DD} = 2.7 to 5.5 V			4.4	
				V_{DD} = 2.0 to 5.5 V			2.2	
				V_{DD} = 1.8 to 5.5 V			1.1	
			Frequency/2 mode	V_{DD} = 2.7 to 5.5 V			6.0	
				V_{DD} = 2.0 to 5.5 V			4.4	
				V_{DD} = 1.8 to 5.5 V			2.2	
			Frequency/4, 8 mode	V_{DD} = 2.0 to 5.5 V			6.0	
				V_{DD} = 1.8 to 5.5 V			4.4	
		One Time PROM version	Through mode	V_{DD} = 4.0 to 5.5 V			6.0	
				V_{DD} = 2.7 to 5.5 V			4.4	
				V_{DD} = 2.5 to 5.5 V			2.2	
			Frequency/2 mode	V_{DD} = 2.7 to 5.5 V			6.0	
				V_{DD} = 2.5 to 5.5 V			4.4	
			Frequency/4, 8 mode	V_{DD} = 2.5 to 5.5 V			6.0	
$f(X_{IN})$	Oscillation frequency (at RC oscillation) (Note)	V_{DD} = 2.7 to 5.5 V					4.4	MHz
$f(X_{IN})$	Oscillation frequency (with a ceramic resonator selected, external clock input)	Mask ROM version	Through mode	V_{DD} = 4.0 to 5.5 V			4.8	MHz
				V_{DD} = 2.7 to 5.5 V			3.2	
				V_{DD} = 2.0 to 5.5 V			1.6	
				V_{DD} = 1.8 to 5.5 V			0.8	
			Frequency/2 mode	V_{DD} = 2.7 to 5.5 V			4.8	
				V_{DD} = 2.0 to 5.5 V			3.2	
				V_{DD} = 1.8 to 5.5 V			1.6	
			Frequency/4, 8 mode	V_{DD} = 2.0 to 5.5 V			4.8	
				V_{DD} = 1.8 to 5.5 V			3.2	
		One Time PROM version	Through mode	V_{DD} = 4.0 to 5.5 V			4.8	
				V_{DD} = 2.7 to 5.5 V			3.2	
				V_{DD} = 2.5 to 5.5 V			1.6	
			Frequency/2 mode	V_{DD} = 2.7 to 5.5 V			4.8	
				V_{DD} = 2.5 to 5.5 V			3.2	
			Frequency/4, 8 mode	V_{DD} = 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

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(SCK)	Serial I/O external input frequency	SCK				f(STCK)/6	Hz
tw(SCK)	Serial I/O external input frequency (“H” and “L“ pulse width)	SCK		3/f(STCK)			s
TPON	Power-on reset circuit valid supply voltage rising time	Mask ROM version	$V_{DD} = 0 \rightarrow 1.8$ V			100	µs
		One Time PROM version	$V_{DD} = 0 \rightarrow 2.5$ V			100	

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, $\overline{\text{RESET}}$	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, $\overline{\text{RESET}}$, 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, $\overline{\text{RESET}}$	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 $\overline{\text{RESET}}$	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	
Δf(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).

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)

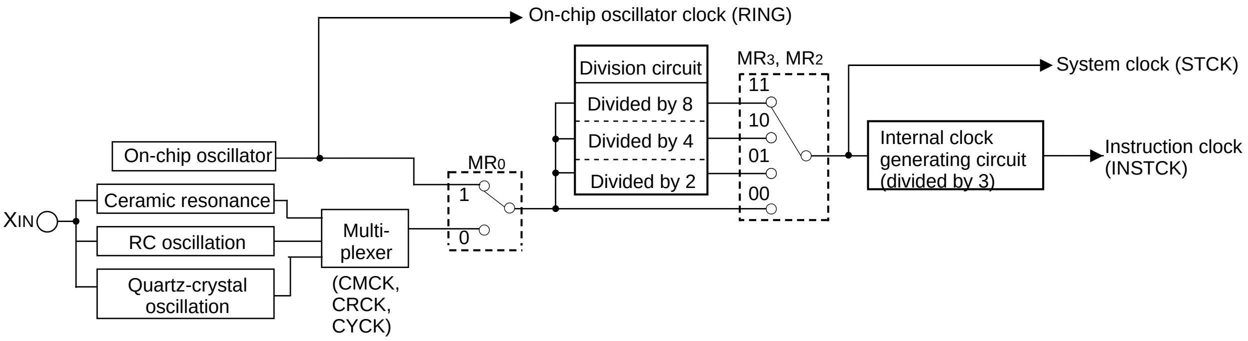
Symbol	Parameter		Test conditions		Limits			Unit
					Min.	Typ.	Max.	
I_{DD}	Supply current	at active mode (with a ceramic resonator, on-chip oscillator stop)	V_{DD} = 5 V f(X_{IN}) = 6 MHz	f(STCK) = f(X_{IN})/8		1.4	2.8	mA
				f(STCK) = f(X_{IN})/4		1.6	3.2	
				f(STCK) = f(X_{IN})/2		2.0	4.0	
				f(STCK) = 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	

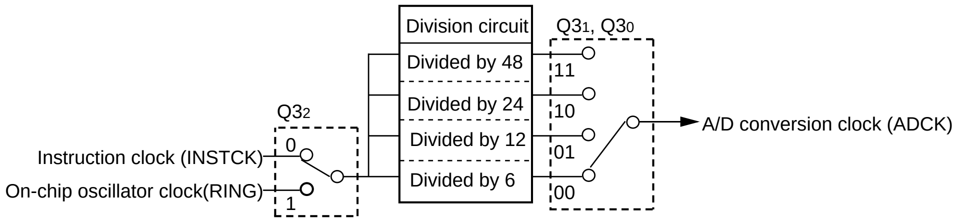
A/D converter recommended operating conditions

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

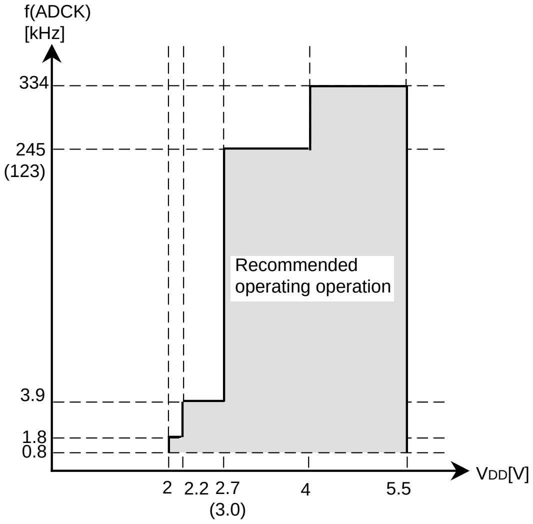
Symbol	Parameter	Conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{DD}	Supply voltage	Mask ROM version		2.0		5.5	V
		One Time PROM version		3.0		5.5	
V_{IA}	Analog input voltage			0		V_{DD}	V
f(ADCK)	A/D conversion clock frequency (Note)	Mask ROM version	V_{DD} = 4.0 to 5.5 V	0.8		334	kHz
			V_{DD} = 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

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)

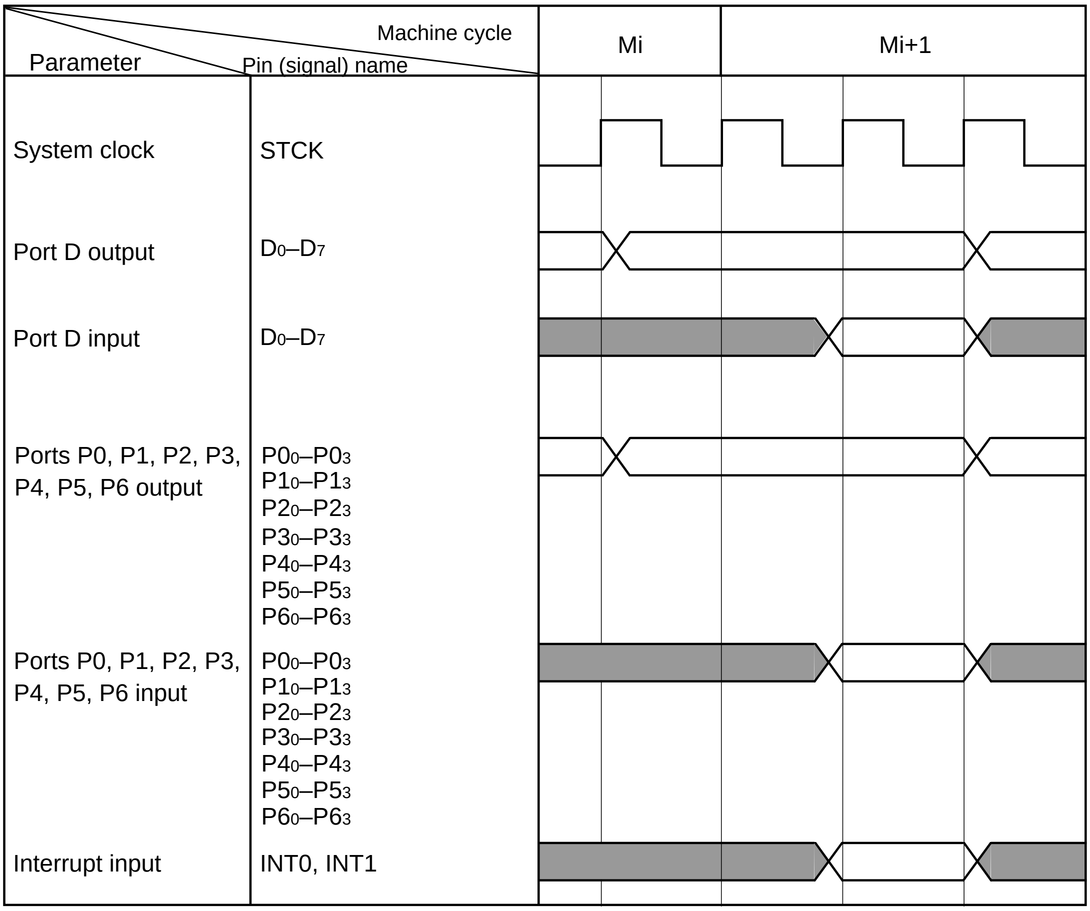
Voltage drop detection circuit characteristics

(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} → (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].

Basic timing diagram



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 75 shows the pin configurations of built-in PROM versions.

The One Time PROM version has pin-compatibility with the mask ROM version.

Table 23 Product of built-in PROM version

Part number	PROM size (X 10 bits)	RAM size (X 4 bits)	Package	ROM type
M34519E8FP	8192 words	384 words	42P2R-A	One Time PROM [shipped in blank]

PIN CONFIGURATION (TOP VIEW)

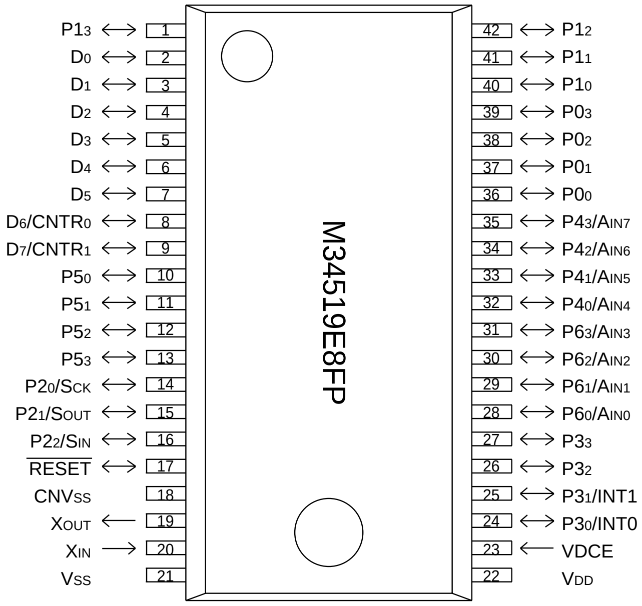


Fig. 75 Pin configuration of built-in PROM version

(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 76.

(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 77 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 24 Programming adapter

Microcomputer	Name of Programming Adapter
M34519E8FP	PCA7441

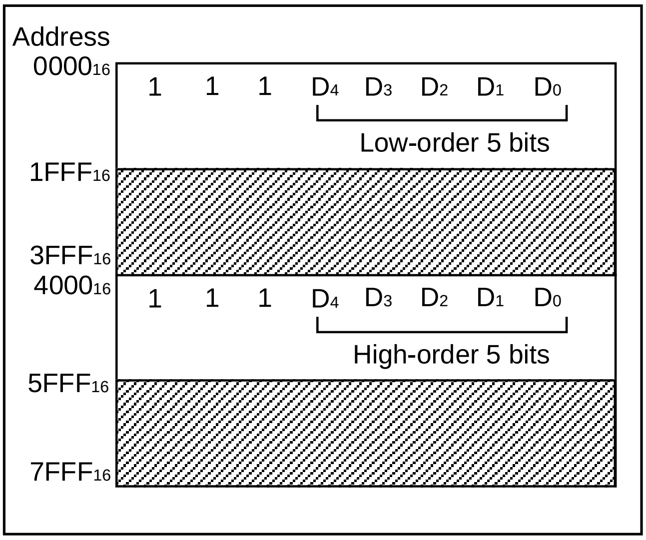
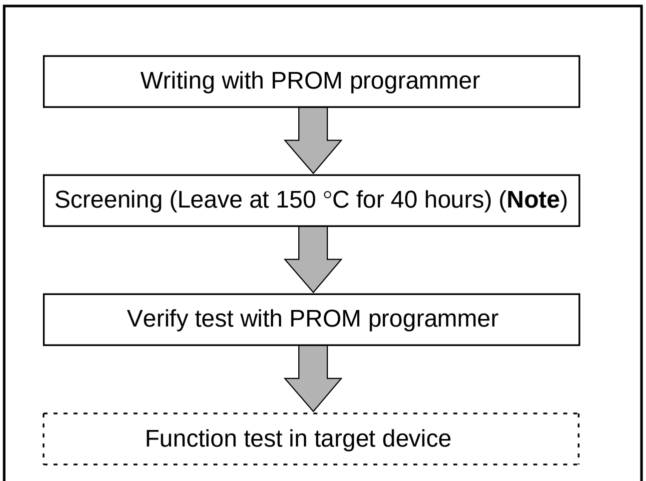


Fig. 76 PROM memory map



Note: Since the screening temperature is higher than storage temperature, never expose the microcomputer to 150 °C exceeding 100 hours.

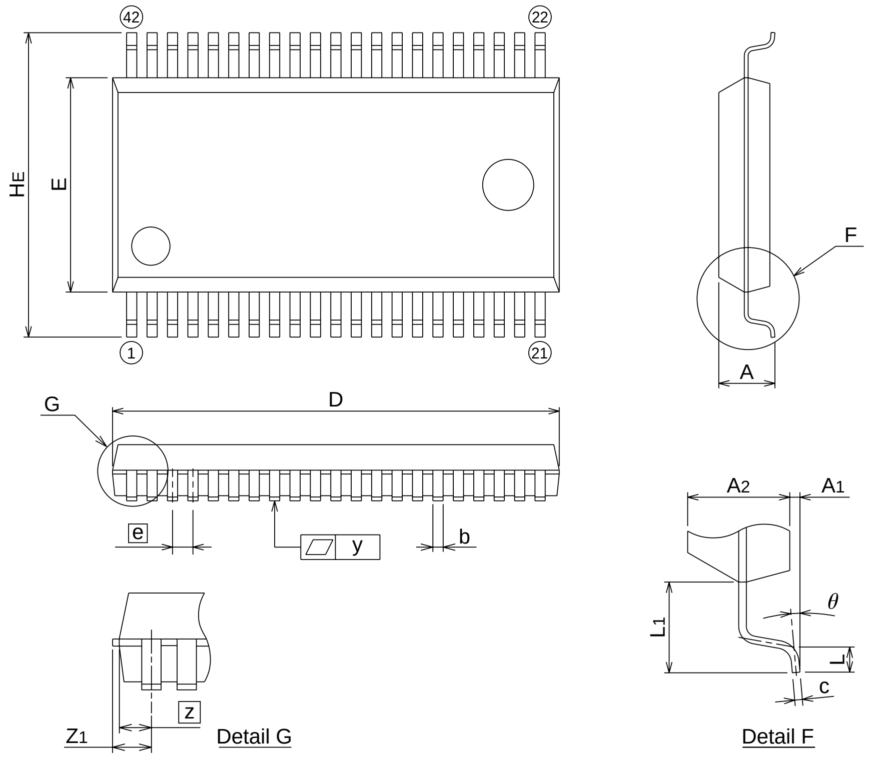
Fig. 77 Flow of writing and test of the product shipped in blank

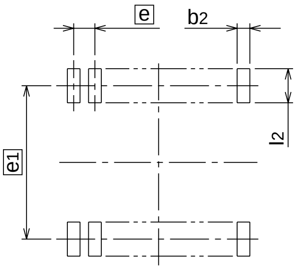
Package outline

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





Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	2.4
A_1	0.05	–	–
A_2	–	2.0	–
b	0.35	0.4	0.5
c	0.13	0.15	0.2
D	17.3	17.5	17.7
E	8.2	8.4	8.6
$\boxed{e}$	–	0.8	–
H_E	11.63	11.93	12.23
L	0.3	0.5	0.7
L_1	–	1.765	–
$\boxed{z}$	–	0.75	–
Z_1	–	–	0.9
y	–	–	0.15
θ	0°	–	10°
b_2	–	0.5	–
$\boxed{e_1}$	–	11.43	–
I_2	1.27	–	–

REVISION HISTORY	4519 Group Data Sheet

Rev.	Date	Description	
		Page	Summary
1.00	Jan. 14, 2003	–	First edition issued
2.00	Apr. 15, 2003		Some values of the following table are revised.
		149	RECOMMENDED OPERATING CONDITIONS 1;
			• Supply voltage (when quartz-crystal oscillator is used)
			• RAM back voltage
		151	RECOMMENDED OPERATING CONDITIONS 3;
			• Oscillation frequency (with a quartz-crystal oscillator)
		154	A/D CONVERTER RECOMMENDED OPERATING CONDITIONS;
			• Supply voltage
			• A/D conversion clock frequency
		155	A/D CONVERTER CHARACTERISTCS;
			• Linearity error
			• Differential non-linearity error
			• Zero transition voltage
			• Full-scale transition voltage
			• Comparator error
		156	VOLTAGE DROP DETECTION CIRCUIT;
			• Detection voltage (reset occurs)
			• Detection voltage (reset release)
3.00	Jul. 27, 2004	All pages	Words standardized: On-chip oscillator, A/D converter
		3	PERFORMANCE OVERVIEW: Power dissipation revised.
		4	PIN DESCRIPTION: Description of RESET pin revised.
		15	Port block diagram (8): Period measurement circuit added.
		25	Fig.17: Period measurement circuit added.
		28	Fig.20 revised.
		29	Fig.23 revised.
		33	Fig.26: Note added.
		34	Table 10 $W1_3$: (Note 2) added, $W2_3$: (Note 2) eliminated.
		39	(12): Some description added.
		40	(14): Some description added.
		44	Some description added.
		45	Fig.33: “DI” instruction added.
		46	Table 11: Relative accuracy revised.
		58	Fig.46: SRST instruction added.
		71	⑪ Timer 4: Some description added.
		73	Fig.64 revised.
		74	Fig.67 revised.
		76	Note on Power Source Voltage added.
		77	$I1_3$, $I1_2$: (Note 2) added.
		78	$W1_3$: (Note 2) added, and Note 2 added.
		86	SNZ0, SNZ1 revised.
		157	Fig.73 revised.
3.01	Jun.15, 2005	All pages	Delete the following: “PRELIMINARY”.
		41	•Prescaler, Timer 1, Timer 2 and Timer 3 count start timing and count time when operation starts, •Timer 4 count start timing and count time when operation starts added.
		73	⑬ Prescaler, Timer 1, Timer 2 and Timer 3 count start timing and count time when operation starts, ⑭ Timer 4 count start timing and count time when operation starts added.

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

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.

RENESAS

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> 2-796-3115, Fax: <82> 2-796-2145

Renesas Technology Malaysia Sdn. Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

© 2005. Renesas Technology Corp., All rights reserved. Printed in Japan.

Colophon .3.0