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.

RENESAS

4559 Group

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

REJ03B0188-0104
Rev.1.04
Aug 23, 2007

DESCRIPTION

The 4559 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 two 8-bit timers (each timer has one or two reload registers), a 16-bit timer for clock count, interrupts, and oscillation circuit switch function.
The various microcomputers in the 4559 Group include variations of type as shown in the table below.

FEATURES

- Minimum instruction execution time.............................0.5 µs (at 6 MHz oscillation frequency, in high-speed through-mode)
- Supply voltage ....1.8 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 and carrier wave output auto-control function
  Timer 2 ....8-bit timer with two reload registers and carrier wave generation circuit
  Timer 3 .... 16-bit timer (fixed dividing frequency)
- Interrupt ....4 sources
- Key-on wakeup function pins 17
- I/O port 22
- Output port 3
- LCD control circuit
  Segment output 32
  Common output 4
- Voltage drop detection circuit
  Reset occurrence.................................Typ. 1.7 V (Ta = 25 °C)
  Reset release ....Typ. 1.8 V (Ta = 25 °C)
  Skip occurrence ....Typ. 2.0 V (Ta = 25 °C)
- Power-on reset circuit
- Watchdog timer
- Clock generating circuit
  Built-in clock (on-chip oscillator)
  Main clock (ceramic resonator/RC oscillation)
  Sub-clock (quartz-crystal oscillation)
- LED drive directly enabled (port D)

APPLICATION

Remote control transmitter

Table 1 Support Product

Part number	ROM size (× 10 bits)	RAM size (× 4 bits)	Package	ROM type
M34559G6FP (Note 1)	6144 words	288 words	PLQP0052JA-A	QzROM
M34559G6-XXXFP	6144 words	288 words	PLQP0052JA-A	QzROM

Note 1: Shipped in blank

PIN CONFIGURATION

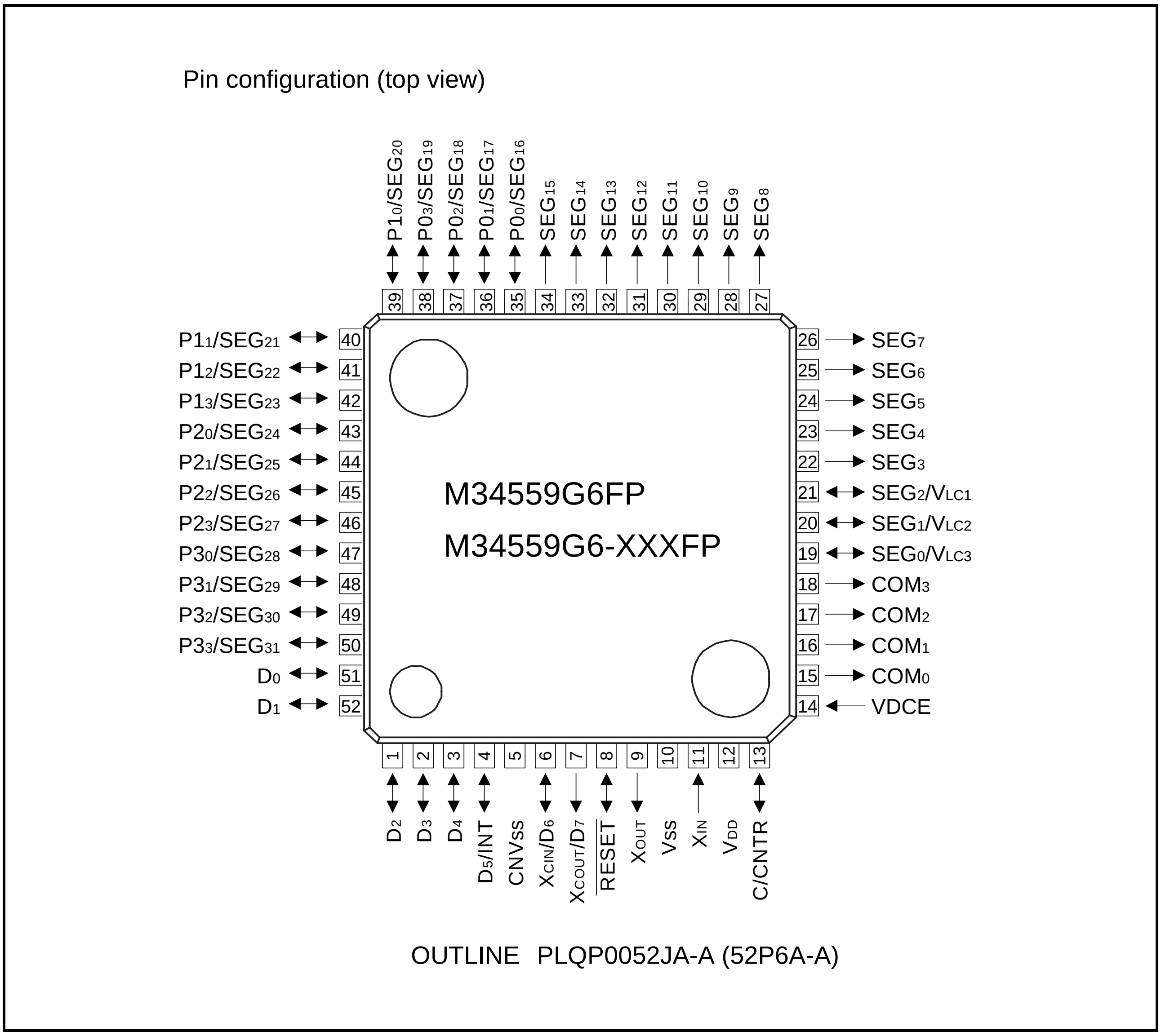


Fig 1. Pin configuration (PLQP0052JA-A type)

FUNCTIONAL BLOCK DIAGRAM

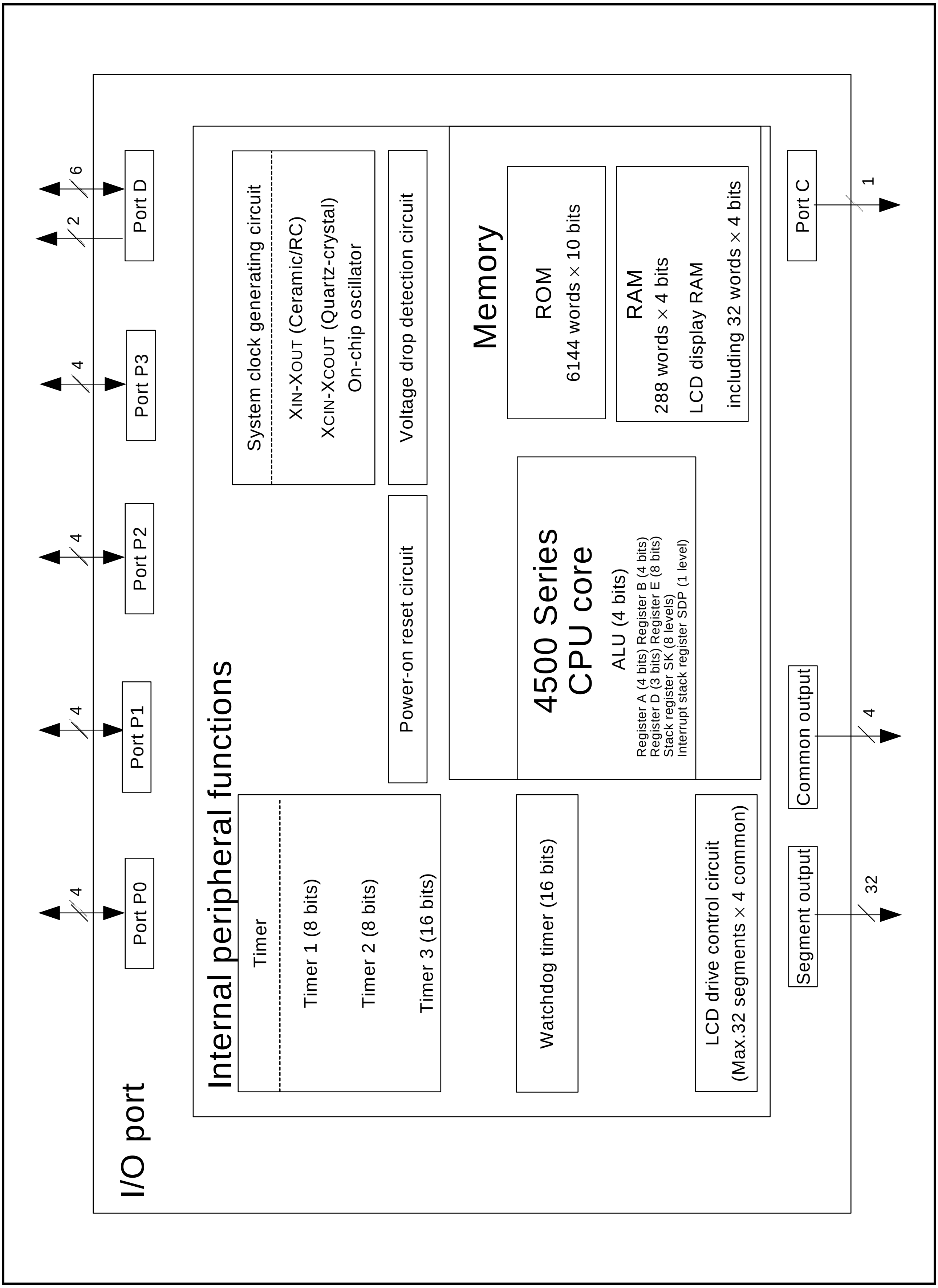


Fig 2. Functional block diagram (PLQP0052JA-A type)

PERFORMANCE OVERVIEW

Table 2 Performance overview

Parameter			Function
Number of basic instructions			135
Minimum instruction execution time			0.5 µs (Oscillation frequency 6 MHz: high-speed through mode)
Memory sizes	ROM		6144 words × 10 bits
	RAM		288 words × 4 bits (including LCD display RAM 32 words × 4 bits)
I/O port	D_0–D_5	I/O (Input is examined by skip decision.)	Six independent I/O ports. The output structure can be switched by software. Port D_5 is also used as INT pin.
	D_6, D_7	Output	Two independent output ports. Ports D_6 and D_7 are also used as X_{CIN} and X_{COUT}, respectively.
	$P0_0$–$P0_3$	I/O	4-bit I/O port; A pull-up function, a key-on wakeup function and output structure can be switched by software. Ports $P0_0$–$P0_3$ are also used as SEG_{16}–SEG_{19}, respectively.
	$P1_0$–$P1_3$	I/O	4-bit I/O port; A pull-up function, a key-on wakeup function and output structure can be switched by software. Ports $P1_0$–$P1_3$ are also used as SEG_{20}–SEG_{23}, respectively.
	$P2_0$–$P2_3$	I/O	4-bit I/O port; A pull-up function, a key-on wakeup function and output structure can be switched by software. Ports $P2_0$–$P2_3$ are also used as SEG_{24}–SEG_{27}, respectively.
	$P3_0$–$P3_3$	I/O	4-bit I/O port; A pull-up function, a key-on wakeup function and output structure can be switched by software. Ports $P3_0$–$P3_3$ are also used as SEG_{28}–SEG_{31}, respectively.
	C	Output	1-bit output; Port C is also used as CNTR pin.
Timer	Timer 1		8-bit timer with a reload register and carrier wave output auto-control function, and has an event counter.
	Timer 2		8-bit timer with two reload registers and carrier wave generation function.
	Timer 3		16-bit timer, fixed dividing frequency (timer for clock count)
	Timer LC		4-bit programmable timer with a reload register (for LCD clock generating)
	Watchdog timer		16-bit timer, fixed dividing frequency (timer for monitor)
LCD control circuit	Selective bias value		1/2, 1/3 bias
	Selective duty value		2, 3, 4 duty
	Common output		4
	Segment output		32
	Internal resistor for power supply		2r × 3, 2r × 2, r × 3, r × 2 (r = 100 kΩ, (Ta = 25 °C, Typical value))
Voltage drop detection circuit	Reset occurrence		Typ. 1.7 V (Ta=25 °C)
	Reset release		Typ. 1.8 V (Ta=25 °C)
	Skip occurrence		Typ. 2.0 V (Ta=25 °C)
Power-on reset circuit			Built-in
Interrupt	Source		4 sources (one for external, three for timers)
	Nesting		1 level
Subroutine nesting			8 levels
Device structure			CMOS silicon gate
Package			52-pin plastic molded LQFP (PLQP0052JA-A)
Operating temperature range			-20 to 85 °C
Power source voltage			1.8 to 5.5 V (It depends on operation source clock, oscillation frequency and operation mode)
Power dissipation (Typ. value)	At active mode		0.3 mA (Ta = 25 °C, V_{DD} = 3 V, f(X_{IN}) = 4 MHz, f(X_{CIN}) = stop, f(RING) = stop, f(STCK) = f(X_{IN})/8
	At clock operating mode		5 µA (Ta = 25 °C, V_{DD} = 3 V, f(X_{CIN}) = 32 kHz)
	At RAM back-up		0.1 µA (Ta = 25 °C, V_{DD} = 5 V, output transistor is cut-off state)

PIN DESCRIPTION

Table 3 Pin description

Pin	Name	Input/Output	Function
V_{DD}	Power source	–	Connected to a plus power supply.
V_{SS}	Power source	–	Connected to a 0 V power supply.
CNV_{SS}	CNV_{SS}	–	This pin is shared with the V_{PP} pin which is the power source input pin for programming the built-in QzROM. Connect to V_{SS} through a resistor about 5 kΩ.
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.
X_{IN}	Main clock input	Input	I/O pins of the main clock generating circuit. When using a ceramic resonator, connect it between pins X_{IN} and X_{OUT}. A feedback resistor is built-in between them. When using the RC oscillation, connect a resistor and a capacitor to X_{IN}, and leave X_{OUT} pin open.
X_{OUT}	Main clock output	Output	
X_{CIN}	Sub clock input	Input	I/O pins of the sub-clock generating circuit. Connect a 32.768 kHz quartz-crystal oscillator between pins X_{CIN} and X_{COUT}. A feedback resistor is built-in between them. X_{CIN} and X_{COUT} pins are also used as ports D_6 and D_7, respectively.
X_{COUT}	Sub clock output	Output	
$\overline{RESET}$	Reset I/O	I/O	An N-channel open-drain I/O pin for a system reset. When the SRST instruction, watchdog timer, the built-in power-on reset or the voltage drop detection circuit causes the system to be reset, the $\overline{RESET}$ pin outputs “L” level.
D_0–D_5	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. Port D_5 is also used as INT pin.
D_6, D_7	Output port D	Output	Each pin of port D has an independent 1-bit wide output function. The output structure is N-channel open-drain. Ports D_6 and D_7 are also used as X_{CIN} pin and X_{COUT} pin, respectively.
$P0_0$–$P0_3$	I/O port P0	I/O	Port P0 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to “1” and select the N-channel open-drain. Port P0 has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports $P0_0$–$P0_3$ are also used as SEG_{16}–SEG_{19}, respectively.
$P1_0$–$P1_3$	I/O port P1	I/O	Port P1 serves as a 4-bit I/O port. The output structure can be switched to N-channel open-drain or CMOS by software. For input use, set the latch of the specified bit to “1” and select the N-channel open-drain. Port P1 has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports $P1_0$–$P1_3$ are also used as SEG_{20}–SEG_{23}, respectively.
$P2_0$–$P2_3$	I/O port P2	I/O	Port P2 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 P2 has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports $P2_0$–$P2_3$ are also used as SEG_{24}–SEG_{27}, respectively.
$P3_0$–$P3_3$	I/O port P3	I/O	Port P3 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 P3 has a key-on wakeup function and a pull-up function. Both functions can be switched by software. Ports $P3_0$–$P3_3$ are also used as SEG_{28}–SEG_{31}, respectively.
C	Output port C	Output	1-bit output port. The output structure is CMOS. Port C is also used as CNTR pin.
COM_0–COM_3	Common output	Output	LCD common output pins. Pins COM_0 and COM_1 are used at 1/2 duty, pins COM_0–COM_2 are used at 1/3 duty and pins COM_0–COM_3 are used at 1/4 duty.
SEG_0–SEG_{31}	Segment output	Output	LCD segment output pins. SEG_0–SEG_2 pins are used as V_{LC3}–V_{LC1} pins, respectively. SEG_{16}–SEG_{31} pins are used as Ports $P0_0$–$P0_3$, Ports $P1_0$–$P1_3$, Ports $P2_0$–$P2_3$, and Ports $P3_0$–$P3_3$, respectively.
CNTR	Timer I/O	I/O	CNTR pin has the function to input the clock for the timer 1 event counter and to output the PWM signal generated by timer 2. CNTR pin is also used as Port C.
INT	Interrupt input	Input	INT pin accepts external interrupts. They have the key-on wakeup function which can be switched by software. INT pin is also used as Port D_5.
V_{LC3}–V_{LC1}	LCD power source	–	These are the LCD power supply pins. If an internal resistor is used, connect the V_{LC3} pin to the V_{DD} pin. (If brightness adjustment is required, connect via a resistor.) When using an external power supply, apply voltage such that $V_{SS} \leq V_{LC1} \leq V_{LC2} \leq V_{LC3} \leq V_{DD}$. Pins V_{LC3} to V_{LC1} also function as pins SEG_0 to SEG_2.

MULTIFUNCTION

Table 4 Pin description

Pin	Multifunction	Pin	Multifunction	Pin	Multifunction	Pin	Multifunction
$P0_0$	SEG_{16}	SEG_{16}	$P0_0$	$P3_0$	SEG_{28}	SEG_{28}	$P3_0$
$P0_1$	SEG_{17}	SEG_{17}	$P0_1$	$P3_1$	SEG_{29}	SEG_{29}	$P3_1$
$P0_2$	SEG_{18}	SEG_{18}	$P0_2$	$P3_2$	SEG_{30}	SEG_{30}	$P3_2$
$P0_3$	SEG_{19}	SEG_{19}	$P0_3$	$P3_3$	SEG_{31}	SEG_{31}	$P3_3$
$P1_0$	SEG_{20}	SEG_{20}	$P1_0$	D_5	INT	INT	D_5
$P1_1$	SEG_{21}	SEG_{21}	$P1_1$	D_6	X_{CIN}	X_{CIN}	D_6
$P1_2$	SEG_{22}	SEG_{22}	$P1_2$	D_7	X_{COUT}	X_{COUT}	D_7
$P1_3$	SEG_{23}	SEG_{23}	$P1_3$	C	CNTR	CNTR	C
$P2_0$	SEG_{24}	SEG_{24}	$P2_0$	SEG_0	V_{LC3}	V_{LC3}	SEG_0
$P2_1$	SEG_{25}	SEG_{25}	$P2_1$	SEG_1	V_{LC2}	V_{LC2}	SEG_1
$P2_2$	SEG_{26}	SEG_{26}	$P2_2$	SEG_2	V_{LC1}	V_{LC1}	SEG_2
$P2_3$	SEG_{27}	SEG_{27}	$P2_3$				

Note 1. Pins except above have just single function.
Note 2. The input/output of D_5 can be used even when INT is selected.
Be careful when using inputs of both INT and D_5 since the input threshold value of INT pin is different from that of port D_5.
Note 3. “H“ output function of port C can be used even when the CNTR (output) is used.

PORT FUNCTION

Table 5 Port function

Port	Pin	Input Output	Output structure	I/O unit	Control instructions	Control registers	Remark
Port D	D_0–D_4, D_5/INT	I/O (6)	N-channel open-drain/ CMOS	1 bit	SD, RD SZD, CLD	FR1, FR2, I1, K2	Programmable output structure selection function
	D_6/X_{CIN}, D_7/X_{COUT}	Output (2)	N-channel open-drain			RG	–
Port P0	$P0_0/SEG_{16}$, $P0_1/SEG_{17}$, $P0_2/SEG_{18}$, $P0_3/SEG_{19}$	I/O (4)	N-channel open-drain/ CMOS	4 bits	OP0A IAP0	PU0, K0, FR0, C1	Programmable pull-up, key-on wakeup and output structure selection function
Port P1	$P1_0/SEG_{20}$, $P1_1/SEG_{21}$, $P1_2/SEG_{22}$, $P1_3/SEG_{23}$	I/O (4)	N-channel open-drain/ CMOS	4 bits	OP1A IAP1	PU1, K0, FR0, C2	Programmable pull-up, key-on wakeup and output structure selection function
Port P2	$P2_0/SEG_{24}$, $P2_1/SEG_{25}$, $P2_2/SEG_{26}$, $P2_3/SEG_{27}$,	I/O (4)	N-channel open-drain/ CMOS	4 bits	OP2A IAP2	PU2, K1, FR3, L3	Programmable pull-up, key-on wakeup and output structure selection function
Port P3	$P3_0/SEG_{28}$, $P3_1/SEG_{29}$, $P3_2/SEG_{30}$, $P3_3/SEG_{31}$	I/O (4)	N-channel open-drain/ CMOS	4 bits	OP3A IAP3	PU3, K2, K3, FR2, C3	Programmable pull-up, key-on wakeup and output structure selection function
Port C	C/CNTR	Output (1)	CMOS	1 bit	RCP SCP	W1, W2, W4	–

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 ($f(X_{IN})$) by the external ceramic resonator
- Clock ($f(X_{IN})$) by the external RC oscillation
- Clock ($f(X_{IN})$) by the external input
- Clock (f(RING)) of the on-chip oscillator which is the internal oscillator
- Clock ($f(X_{CIN})$) 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.

• Machine cycle

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

• 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.

Table 6 Table Selection of system clock

Register MR				System clock	Operation mode
MR_3	MR_2	MR_1	MR_0		
1	1	0	0	f(STCK) = f(RING)/8	Internal frequency divided by 8 mode
1	0	0	0	f(STCK) = f(RING)/4	Internal frequency divided by 4 mode
0	1	0	0	f(STCK) = f(RING)/2	Internal frequency divided by 2 mode
0	0	0	0	f(STCK) = f(RING)	Internal frequency through mode
1	1	0	1	$f(STCK) = f(X_{IN})/8$	High-speed frequency divided by 8 mode
1	0	0	1	$f(STCK) = f(X_{IN})/4$	High-speed frequency divided by 4 mode
0	1	0	1	$f(STCK) = f(X_{IN})/2$	High-speed frequency divided by 2 mode
0	0	0	1	$f(STCK) = f(X_{IN})$	High-speed through mode
1	1	1	0	$f(STCK) = f(X_{CIN})/8$	Low-speed frequency divided by 8 mode
1	0	1	0	$f(STCK) = f(X_{CIN})/4$	Low-speed frequency divided by 4 mode
0	1	1	0	$f(STCK) = f(X_{CIN})/2$	Low-speed frequency divided by 2 mode
0	0	1	0	$f(STCK) = f(X_{CIN})$	Low-speed through mode

Note 1. The f(RING)/8 is selected after system is released from reset

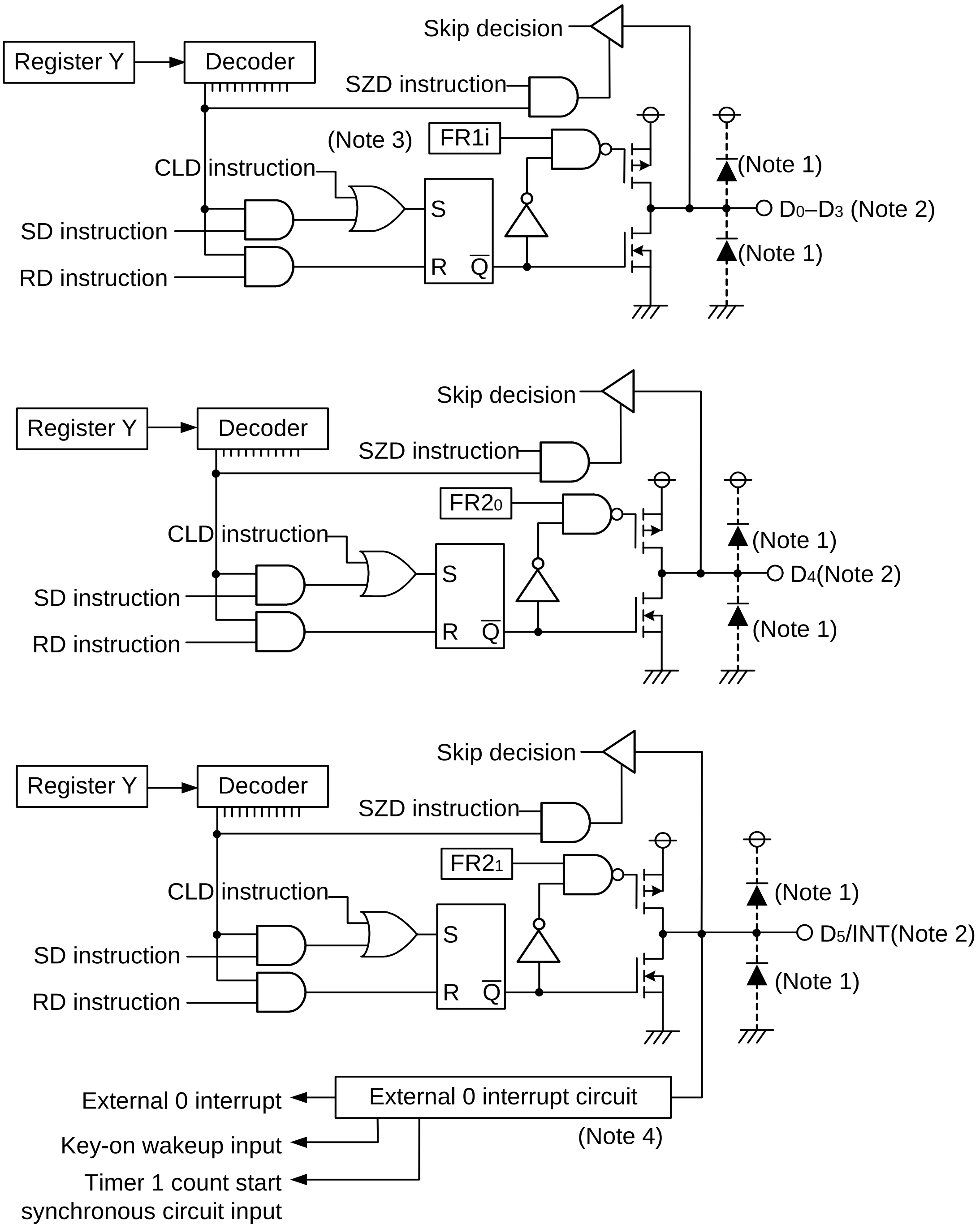
CONNECTIONS OF UNUSED PINS

Table 7 Port function

Pin	Connection	Usage condition
X_{IN}	Connect to V_{SS}.	RC oscillator is not selected
X_{OUT}	Open.	–
X_{CIN}/D_6	Connect to V_{SS}.	–
X_{COUT}/D_7	Open.	–
D_0–D_4	Open.	–
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.
D_5/INT	Open.	INT pin input is disabled.
	Connect to V_{SS}.	N-channel open-drain is selected for the output structure.
$P0_0/SEG_{16}$– $P0_3/SEG_{19}$	Open.	The key-on wakeup function is invalid.
	Connect to V_{SS}.	Segment output is not selected. N-channel open-drain is selected for the output structure. Pull-up transistor is OFF. The key-on wakeup function is invalid.
$P1_0/SEG_{20}$– $P1_3/SEG_{23}$	Open.	The key-on wakeup function is invalid.
	Connect to V_{SS}.	Segment output is not selected. N-channel open-drain is selected for the output structure. Pull-up transistor is OFF. The key-on wakeup function is invalid.
$P2_0/SEG_{24}$– $P2_3/SEG_{27}$	Open.	The key-on wakeup function is invalid.
	Connect to V_{SS}.	Segment output is not selected. N-channel open-drain is selected for the output structure. Pull-up transistor is OFF. The key-on wakeup function is invalid.
$P3_0/SEG_{28}$– $P3_3/SEG_{31}$	Open.	The key-on wakeup function is invalid.
	Connect to V_{SS}.	Segment output is not selected. N-channel open-drain is selected for the output structure. Pull-up transistor is OFF. The key-on wakeup function is invalid.
C/CNTR	Open.	CNTR input is not selected for timer 1 count source.
COM_0–COM_3	Open.	–
SEG_0/V_{LC3}	Open.	SEG_0 pin is selected.
SEG_1/V_{LC2}	Open.	SEG_1 pin is selected.
SEG_2/V_{LC1}	Open.	SEG_2 pin is selected.
SEG_3–SEG_{15}	Open.	–

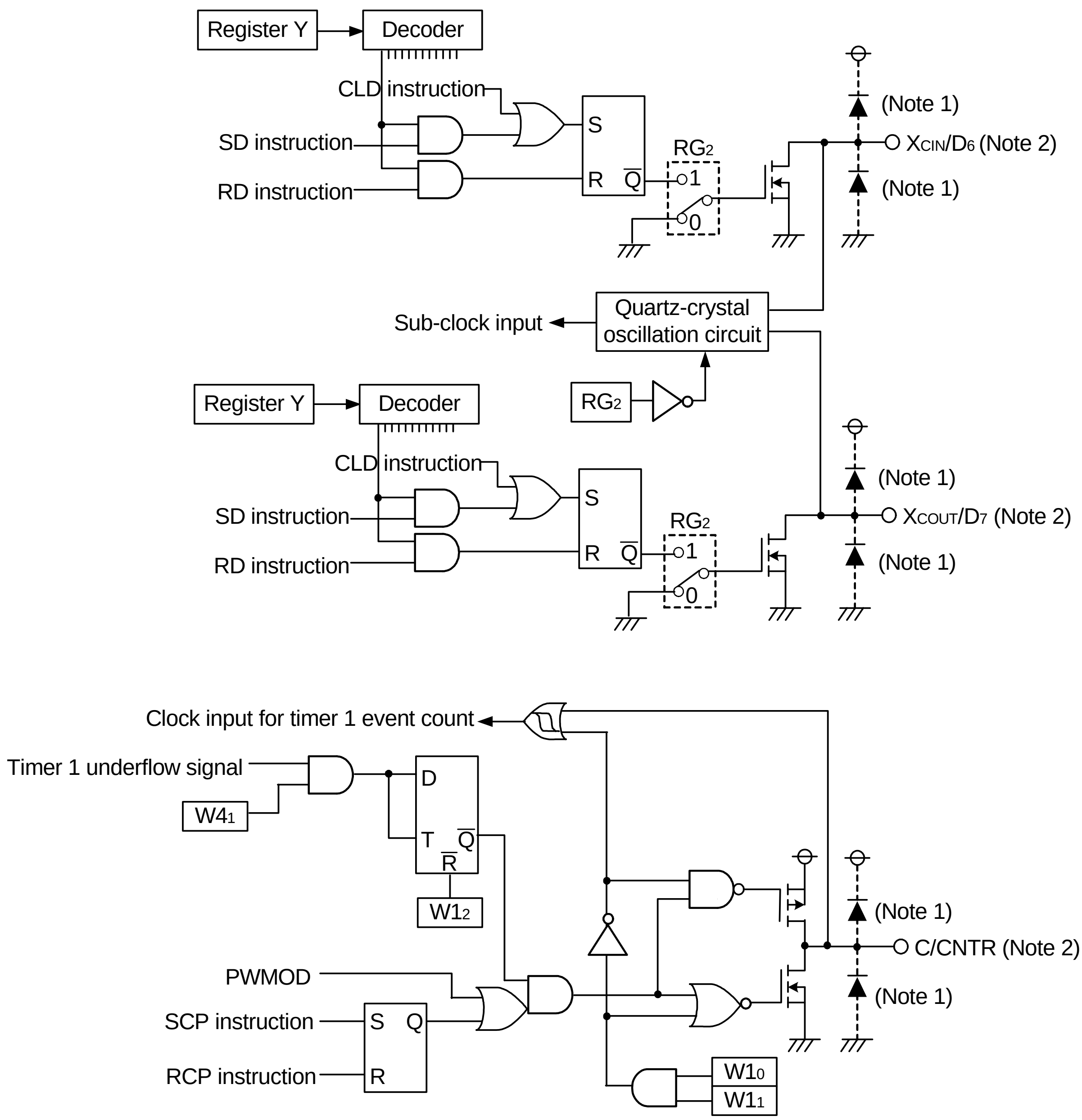
(Note when connecting to V_{SS} or V_{DD})
Connect the unused pins to V_{SS} using the thickest wire at the shortest distance against noise.

PORT BLOCK DIAGRAM



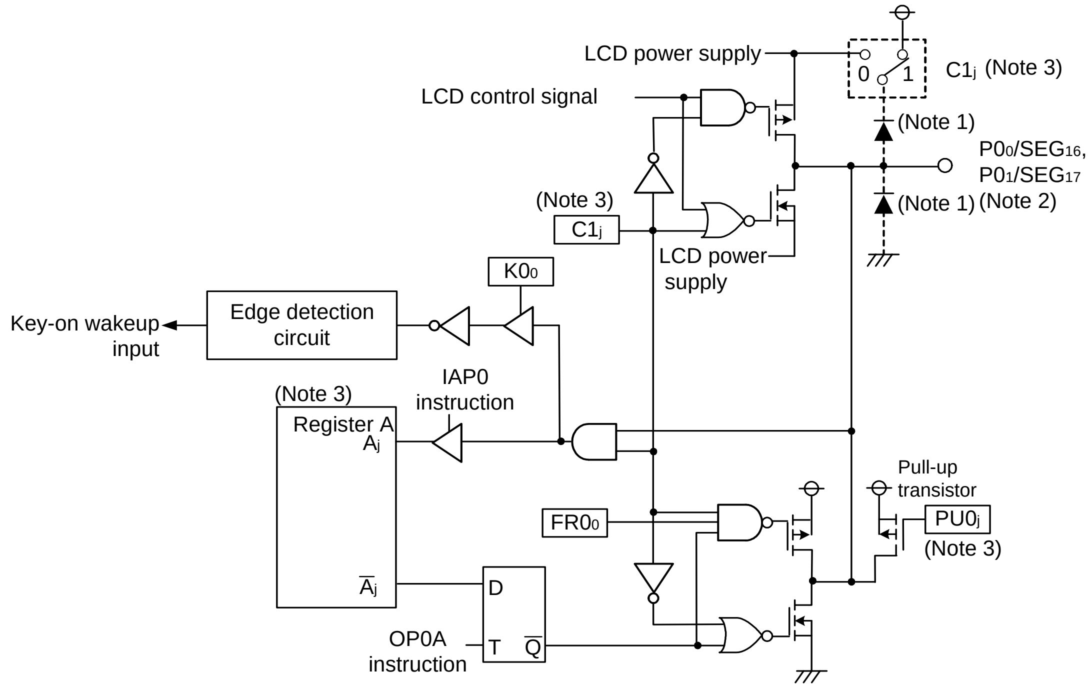
Notes 1. This symbol represents a parasitic diode on the port.
2. Applied potential to these ports must be V_{DD} or less.
3. i represents bits 0 to 3.
4. As for details, refer to the external interrupt structure.

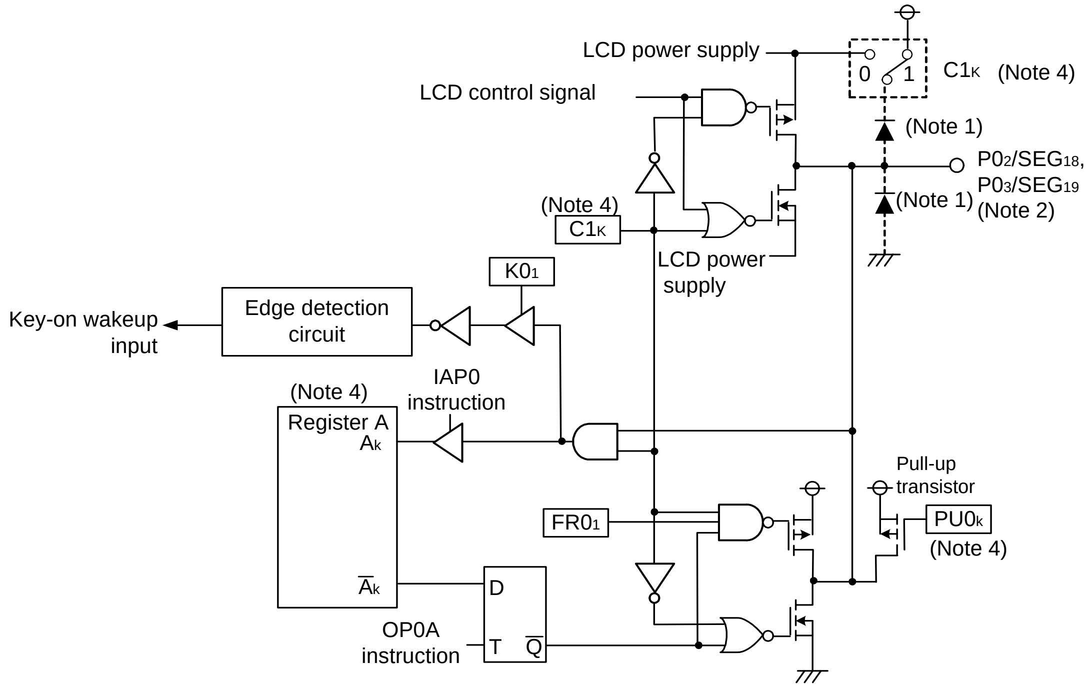
Fig 3. Port block diagram (1)



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

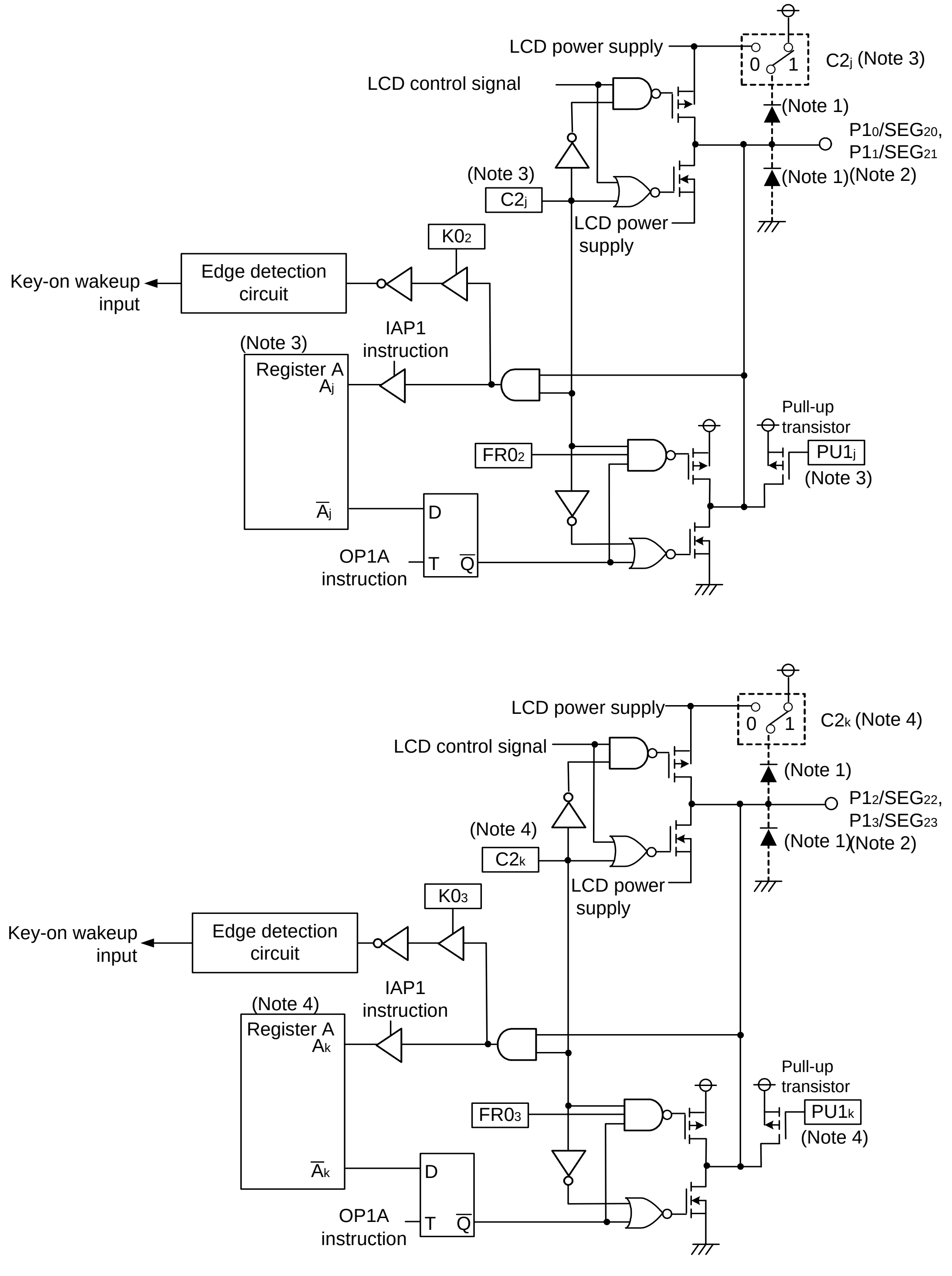
Fig 4. Port block diagram (2)





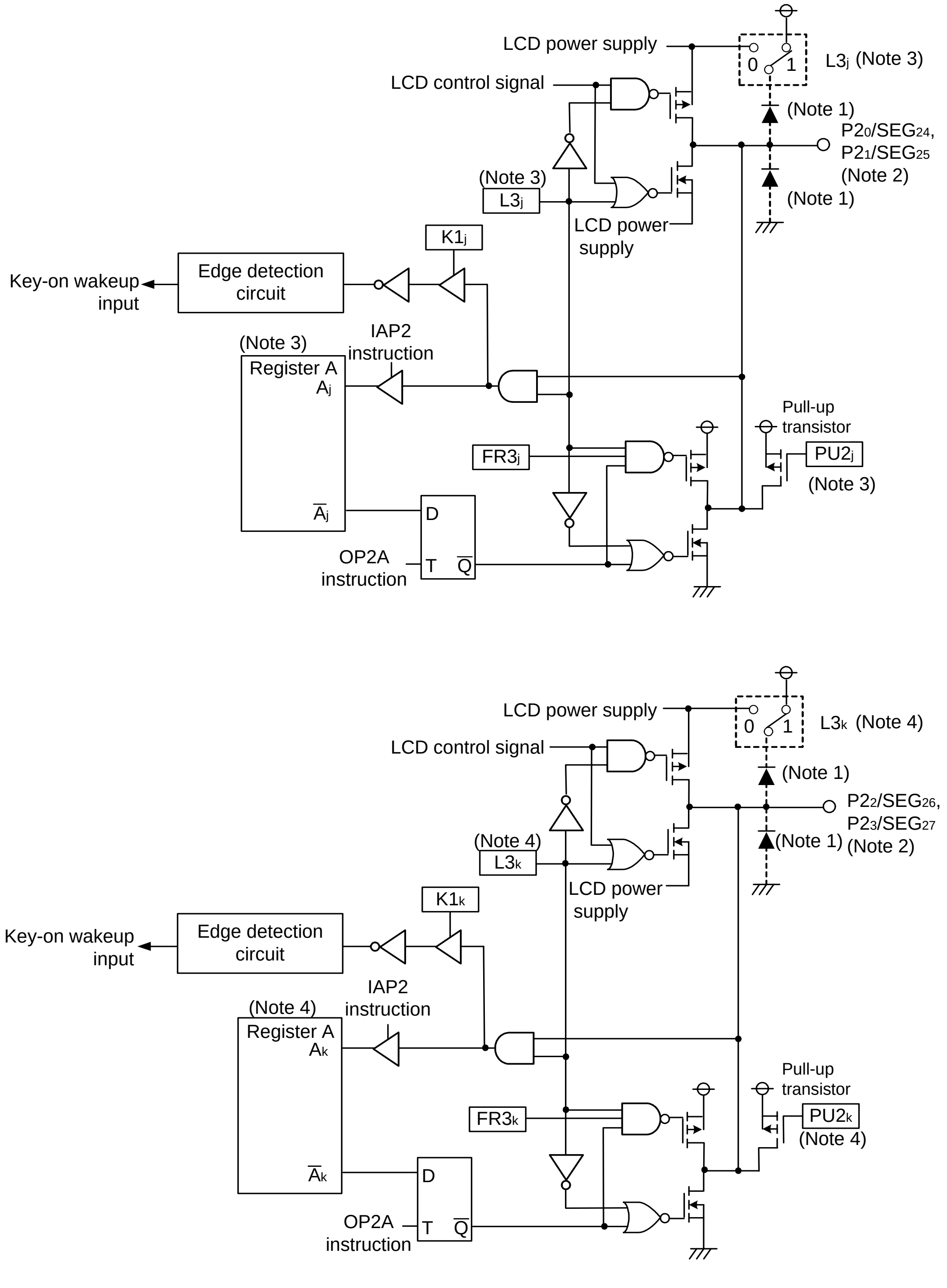
Notes 1. This symbol represents a parasitic diode on the port.
2. Applied potential to these ports must be V_{DD} or less.
3. j represents bits 0, 1.
4. k represents bits 2, 3.

Fig 5. Port block diagram (3)



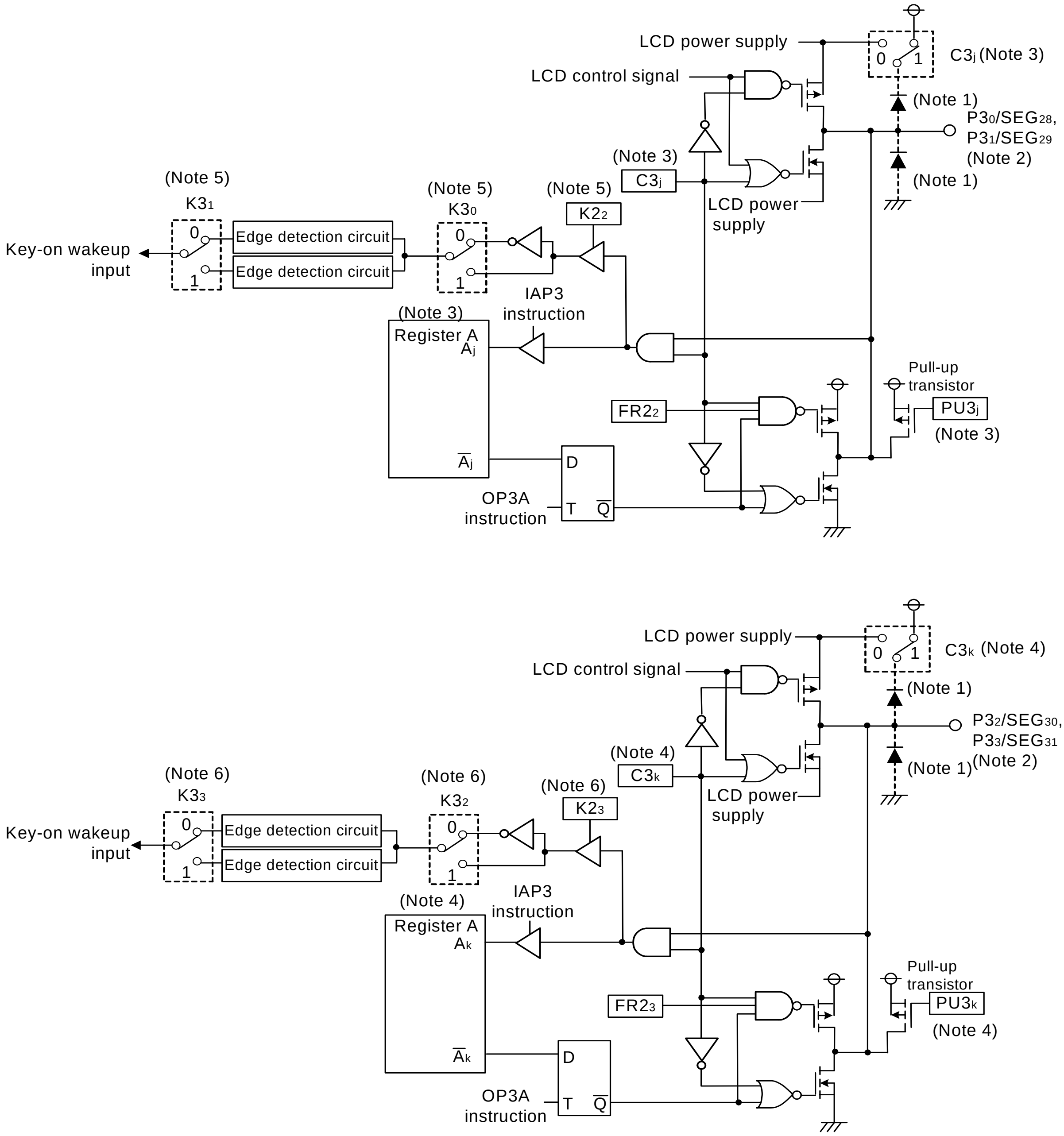
Notes 1. This symbol represents a parasitic diode on the port.
2. Applied potential to these ports must be V_{DD} or less.
3. j represents bits 0, 1.
4. k represents bits 2, 3.

Fig 6. Port block diagram (4)



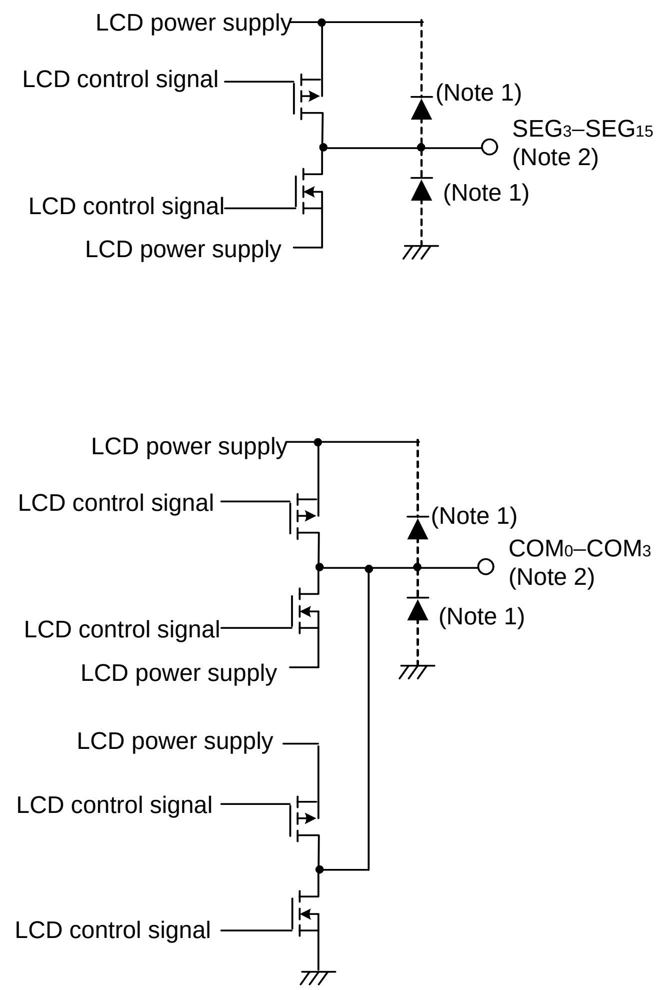
Notes 1. This symbol represents a parasitic diode on the port.
2. Applied potential to these ports must be V_{DD} or less.
3. j represents bits 0, 1.
4. k represents bits 2, 3.

Fig 7. Port block diagram (5)



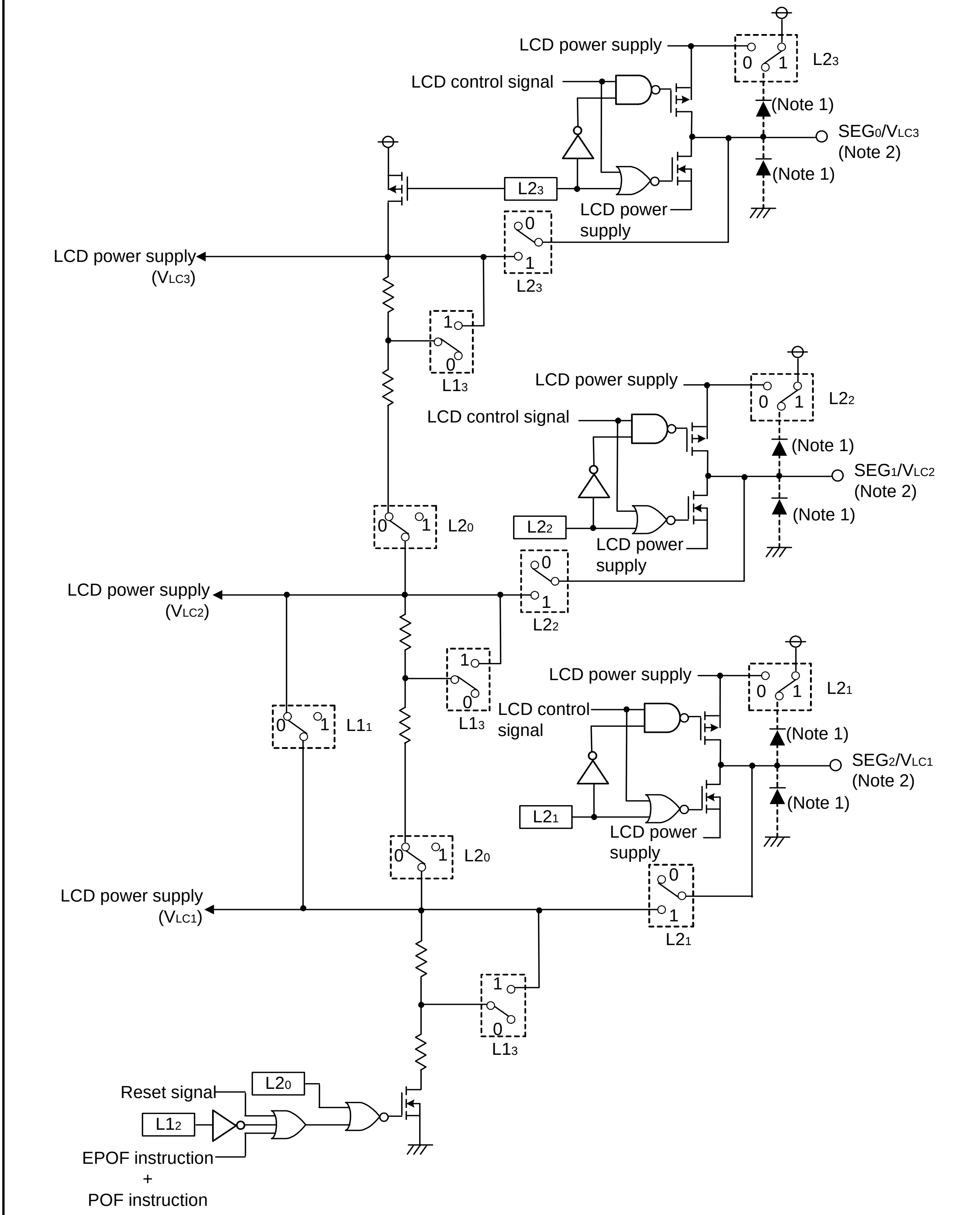
Notes 1. This symbol represents a parasitic diode on the port.
2. Applied potential to these ports must be V_{DD} or less.
3. j represents bits 0, 1.
4. k represents bits 2, 3.
5. For setting key-on wakeup of ports $P3_0$ and $P3_1$ to be invalid ($K2_2$ = “0”) set registers $K3_0$ and $K3_1$ to “0.”
6. For setting key-on wakeup of ports $P3_2$ and $P3_3$ to be invalid ($K2_3$ = “0”) set registers $K3_2$ and $K3_3$ to “0.”

Fig 8. Port block diagram (6)



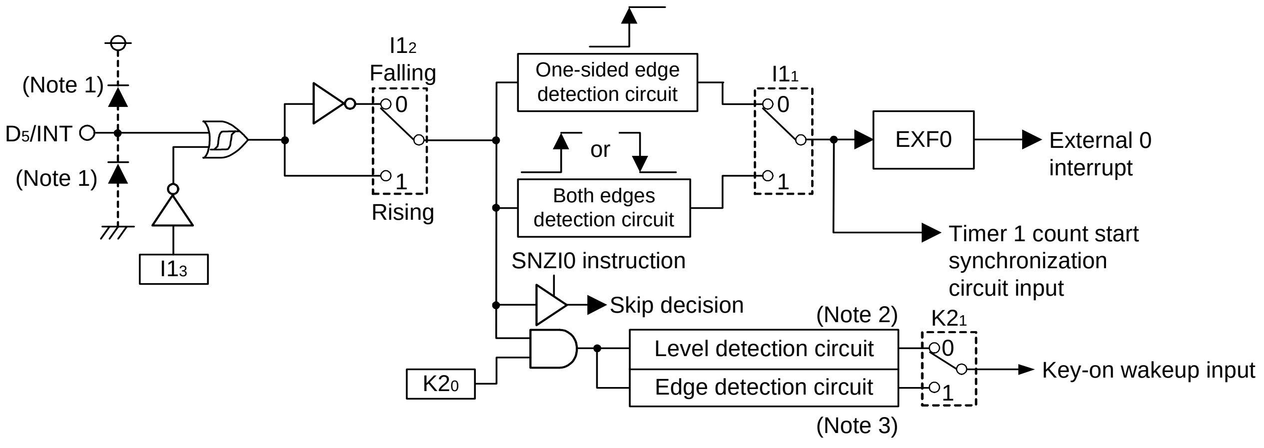
Notes 1. This symbol represents a parasitic diode on the port.
2. Applied potential to these ports must be V_{DD} or less.

Fig 9. Port block diagram (7)



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

Fig 10. Port block diagram (8)



Notes 1: This symbol represents a parasitic diode on the port.
2: When $I1_2$ is 0, “L” level is detected.
When $I1_2$ is 1, “H” level is detected.
3: When $I1_2$ is 0, falling edge is detected.
When $I1_2$ is 1, rising edge is detected.

Fig 11. External interrupt circuit structure

FUNCTION BLOCK OPERATIONS

CPU

(1) Arithmetic logic unit (ALU)

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

(2) Register A and carry flag

Register A is a 4-bit register used for arithmetic, transfer, exchange, and I/O operation.
Carry flag CY is a 1-bit flag that is set to “1” when there is a carry with the AMC instruction (Figure 12).
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 13).
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 14).
Register E is undefined after system is released from reset and returned from the power down mode. 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 (Figure 15).
Also, when the TABP p instruction is executed at UPTF flag = “1”, the high-order 2 bits of ROM reference data is stored to the low-order 2 bits of register D, the high-order 1 bit of register D is “0”.
When the TABP p instruction is executed at UPTF flag = “0”, the contents of register D remains unchanged. The UPTF flag is set to “1” with the SUPT instruction and cleared to “0” with the RUPT instruction.
The initial value of UPTF flag is “0”.
Register D is undefined after system is released from reset and returned from the power down mode. Accordingly, set the initial value.

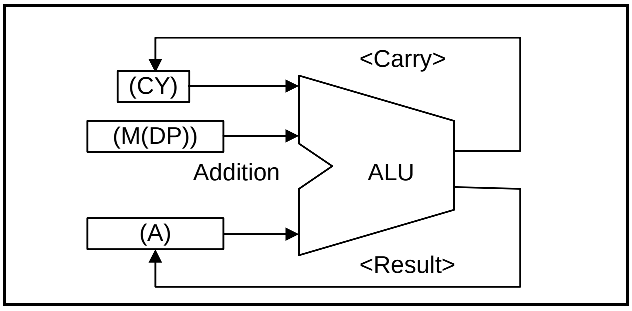


Fig 12. AMC instruction execution example

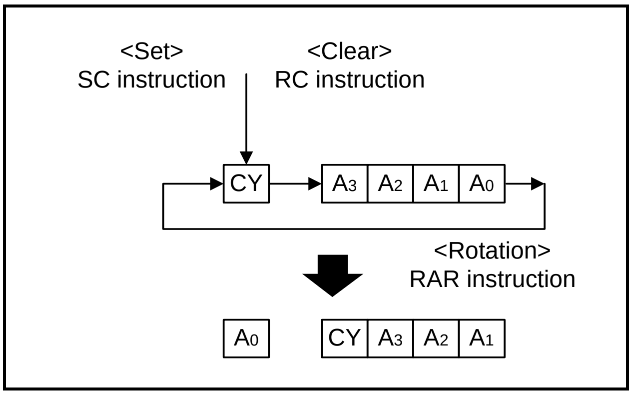


Fig 13. RAR instruction execution example

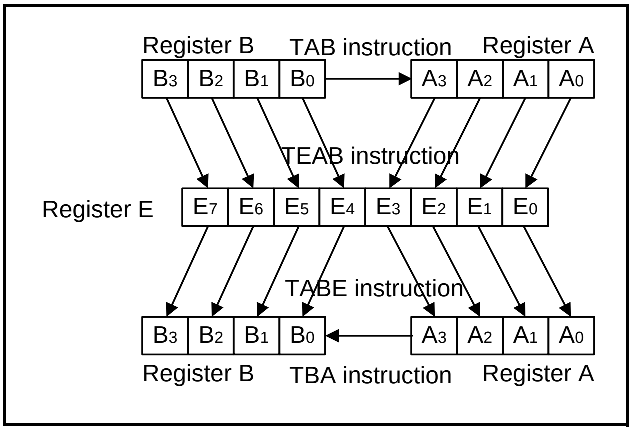


Fig 14. Registers A, B and register E

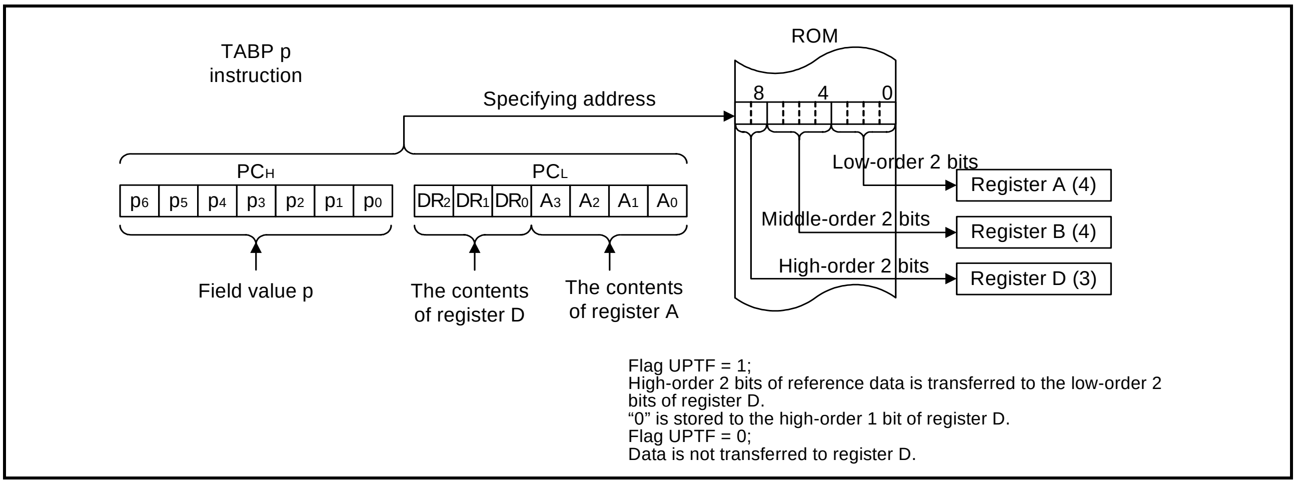


Fig 15. TABP p instruction execution example

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

Stack registers are 14-bit registers.
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).

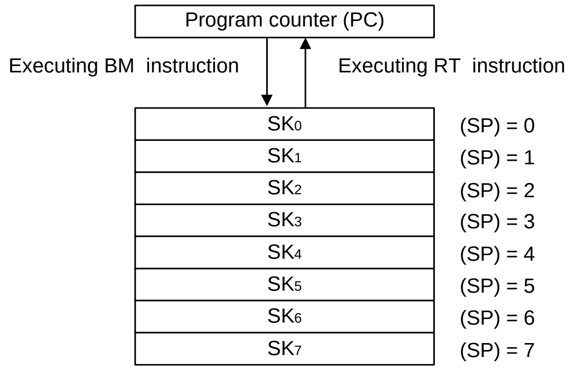
Stack registers (SKs) are eight identical registers, so that subroutines can be nested up to 8 levels. However, one of stack registers is used respectively when using an interrupt service routine and when executing a table reference instruction. Accordingly, be careful not to over the stack when performing 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 16 shows the stack registers (SKs) structure.
Figure 17 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.

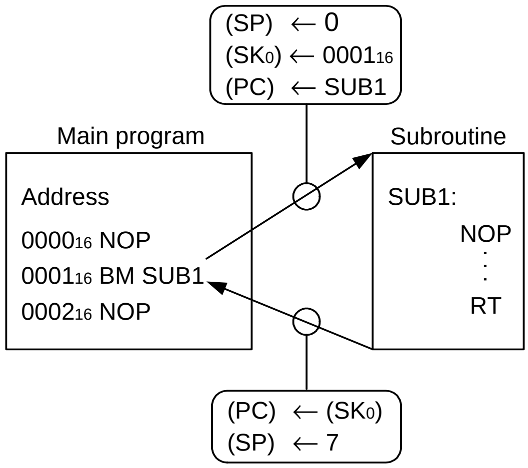
(7) Skip flag

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 power down 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 16. 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 17. 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 18).
Make sure that the PC_H does not specify after the last page of the built-in ROM.

(9) Data pointer (DP)

Data pointer (DP) is used to specify a RAM address and consists of registers Z, X, and Y. Register Z specifies a RAM file group, register X specifies a file, and register Y specifies a RAM digit (Figure 19).
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 20).

- **Note**

Register Z of data pointer is undefined after system is released from reset.
Also, registers Z, X and Y are undefined in the power down mode. After system is returned from the power down mode, set these registers.

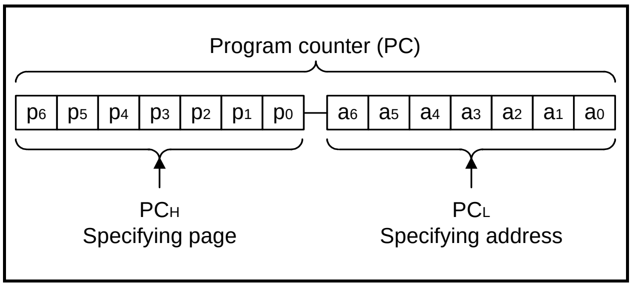


Fig 18. Program counter (PC) structure

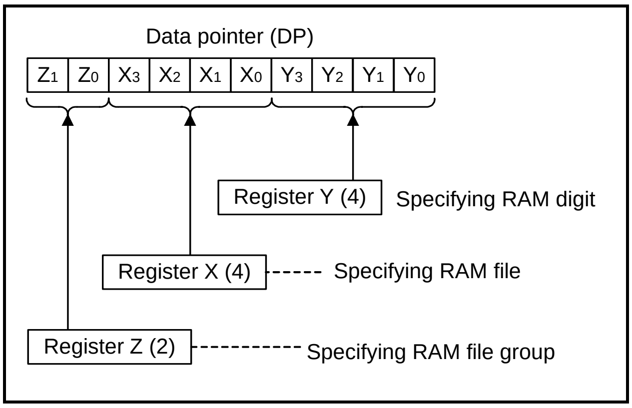


Fig 19. Data pointer (DP) structure

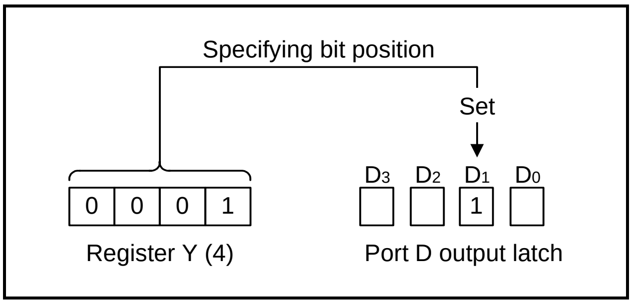


Fig 20. SD instruction execution example

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 21 shows the ROM map of M34559G6.

Table 8 ROM size and pages

Part number	ROM (PROM) size (× 10 bits)	Pages
M34559G6	6144 words	48 (0 to 47)

A part of page 1 (addresses 0080$_{16}$ to 00FF$_{16}$) is reserved for interrupt addresses (Figure 22). 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.

ROM Code Protect Address

When selecting the protect bit write by using a serial programmer or selecting protect enabled for writing shipment by Renesas Technology corp., reading or writing from/to QzROM is disabled by a serial programmer.
As for the QzROM product in blank, the ROM code is protected by selecting the protect bit write at ROM writing with a serial programmer.
As for the QzROM product shipped after writing, whether the ROM code protect is used or not can be selected as ROM option setup ("MASK option" written in the mask file converter) when ordering.

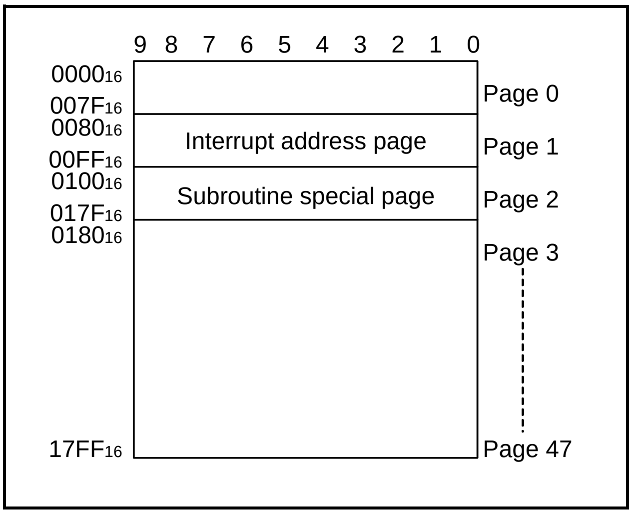


Fig 21. ROM map of M34559G6

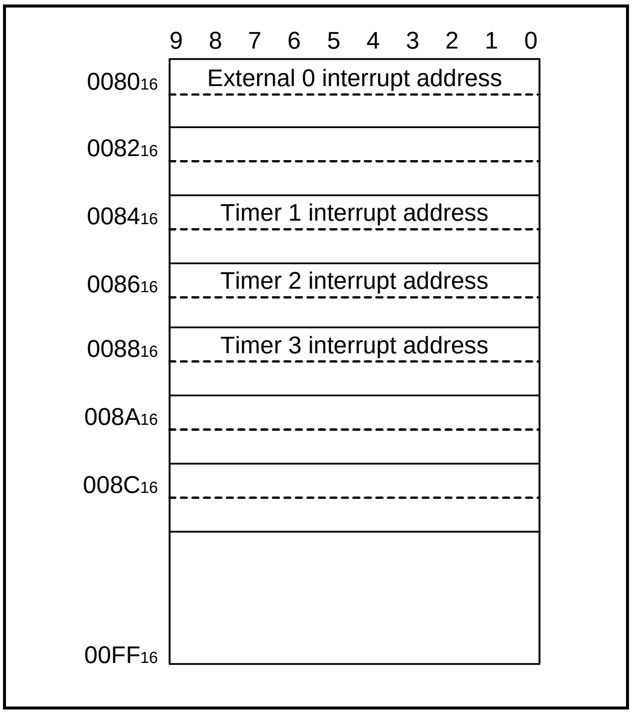


Fig 22. Page 1 (addresses 0080$_{16}$ to 00FF$_{16}$) structure

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 power down mode).
RAM includes the area for LCD.
When writing “1” to a bit corresponding to displayed segment, the segment is turned on.
Table 9 shows the RAM size. Figure 23 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 power down mode. After system is returned from the power down mode, set these registers.

Table 9 RAM size and pages

Part number	RAM size
M34559G6	288 words × 4 bits (1152 bits)

RAM 288 words × 4 bits (1152 bits)

	Register Z	0									1			
	Register X	0	1	2	3	···	12	13	14	15	0	1	2	3
Register Y	0													
	1													
	2													
	3													
	4													
	5													
	6													
	7													
	8										0	8	16	24
	9										1	9	17	25
	10										2	10	18	26
	11										3	11	19	27
	12										4	12	20	28
	13										5	13	21	29
	14										6	14	22	30
	15										7	15	23	31

Note: The numbers in the shaded area indicate the corresponding segment output pin numbers.

Fig 23. 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 10 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 11 shows the interrupt request flag, interrupt enable bit and skip instruction.
Table 12 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 except the voltage drop detection circuit interrupt request flag is cleared to “0” when either;

- an interrupt occurs, or
- a skip instruction is executed.

The voltage drop detection circuit interrupt request flag cannot be cleared to “0” at the state that the activated condition is satisfied.
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 10.

Table 10 Interrupt sources

Priority level	Interrupt source		Interrupt address
	Interrupt name	Activated condition	
1	External 0 interrupt	Level change of INT0 pin	Address 0 in page 1
2	Timer 1 interrupt	Timer 1 underflow	Address 4 in page 1
3	Timer 2 interrupt	Timer 2 underflow	Address 6 in page 1
4	Timer 3 interrupt	Timer 3 underflow	Address 8 in page 1

Table 11 Interrupt request flag, interrupt enable bit and skip instruction

Interrupt name	Interrupt request flag	Skip instruction	Interrupt enable bit
External 0 interrupt	EXF0	SNZ0	$V1_0$
Timer 1 interrupt	T1F	SNZT1	$V1_2$
Timer 2 interrupt	T2F	SNZT2	$V1_3$
Timer 3 interrupt	T3F	SNZT3	$V2_0$

Table 12 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 25).

- 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 24)

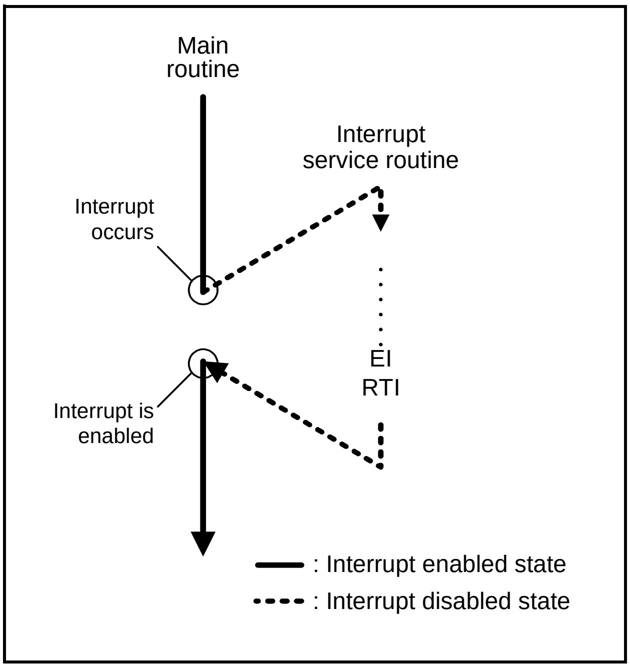


Fig 24. Program example of interrupt processing

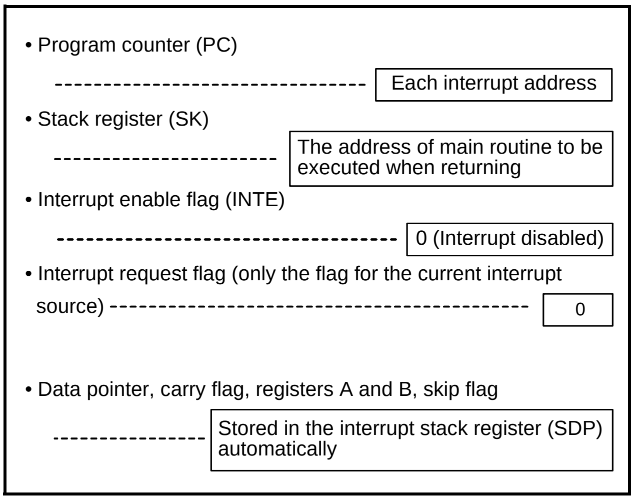


Fig 25. Internal state when interrupt occurs

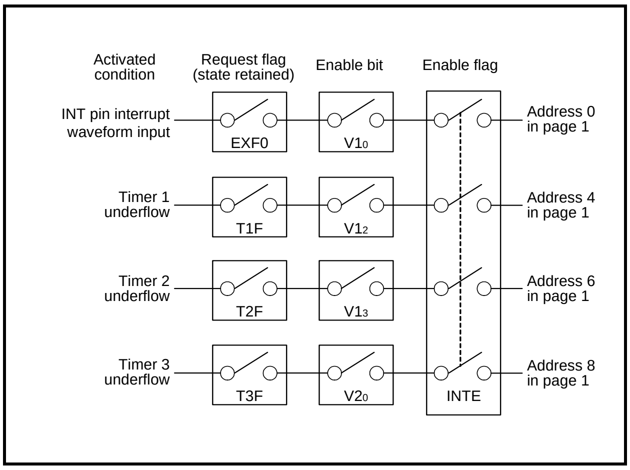


Fig 26. Interrupt system diagram

(6) Interrupt control registers

• Interrupt control register V1
Interrupt enable bits of external 0, 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 interrupt enable bit are assigned to register V2. Set the contents of this register through register A with the TV2A instruction. The TAV2 instruction can be used to transfer the contents of register V2 to register A.

Table 13 Interrupt control registers

Interrupt control register V1		at reset : 0000_2	at power down : 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)	
$V1_1$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$V1_0$	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 power down : 0000_2	R/W TAV2/TV2A
$V2_3$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$V2_2$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$V2_1$	Not used	0	This bit has no function, but read/write is enabled.	
		1		
$V2_0$	Timer 3 interrupt enable bit	0	Interrupt disabled (SNZT3 instruction is valid)	
		1	Interrupt enabled (SNZT3 instruction is invalid)	

Note 1."R" represents read enabled, and "W" represents write enabled.

(7) Interrupt sequence

Interrupts occur only when the respective INTE flag, interrupt enable bits ($V1_0$, $V1_2$, $V1_3$, $V3_0$), and interrupt request flag are set to "1." The interrupt occurs two or three cycles after the cycle where all the above three conditions are satisfied.
The interrupt occurs after three machine cycles if instructions other than one-cycle instruction are executed when the conditions are satisfied (Refer to Figure 27).

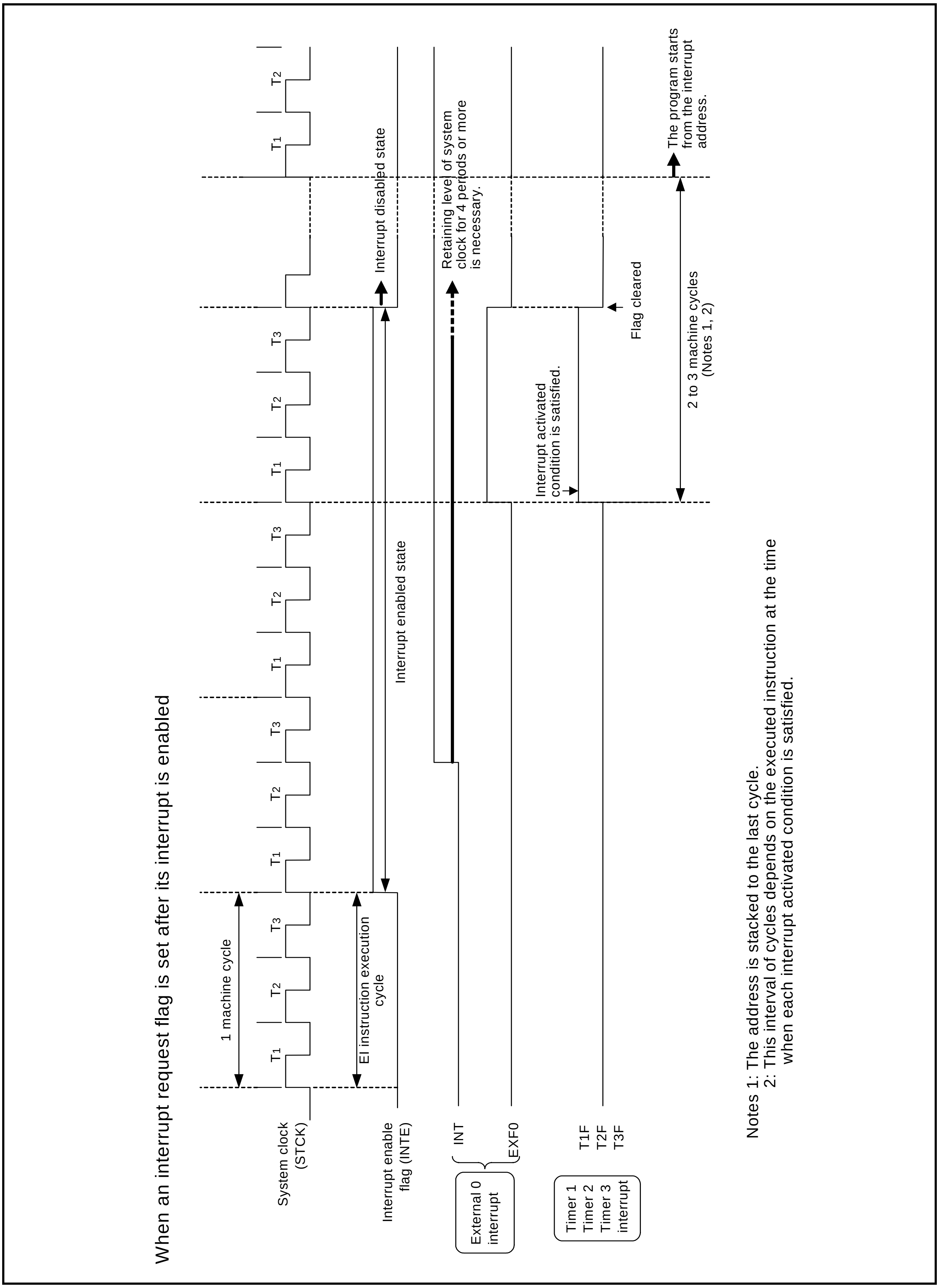


Fig 27. Interrupt sequence

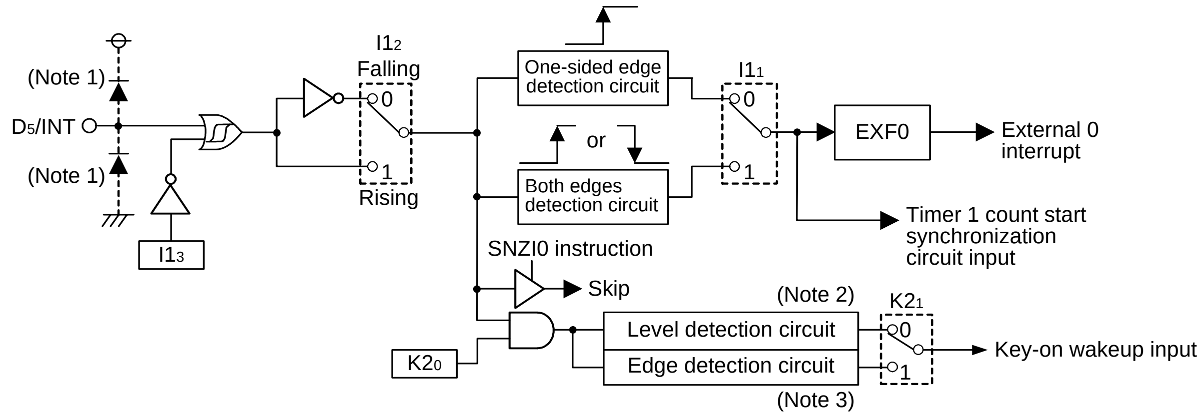
EXTERNAL INTERRUPTS

The 4559 Group has the external 0 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 register I1.

Table 14 External interrupt activated conditions

Name	Input pin	Activated condition	Valid waveform selection bit
External 0 interrupt	D_5/INT	When the next waveform is input to D_5/INT pin • Falling waveform ("H" → "L") • Rising waveform ("L" → "H") • Both rising and falling waveforms	$I1_1$ $I1_2$



Note 1: This symbol represents a parasitic diode on the port.
2: When $I1_2$= 0(X=0 or 1) is 0, "L" level is detected.
When $I1_2$ is 1, "H" level is detected.
3: When $I1_2$ is 0, falling edge is detected.
When $I1_2$ is 1, rising edge is detected.

Fig 28. External interrupt circuit structure

(1) External 0 interrupt request flag (EXF0)

External 0 interrupt request flag (EXF0) is set to “1” when a valid waveform is input to D_5/INT 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 27).

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 D_5/INT 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 INT 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 D_5/INT pin, the EXF0 flag is set to “1” and the external 0 interrupt occurs.

(2) External interrupt control registers

(1) 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.

Table 15 External interrupt control register

Interrupt control register I1		at reset : 0000_2	at power down : state retained	R/W TAI1/TI1A
$I1_3$	INT pin input control bit (Note 2)	0	INT pin input disabled	
		1	INT pin input enabled	
$I1_2$	Interrupt valid waveform for INT pin/ return level selection bit (Note 2)	0	Falling waveform (“L” level of INT pin is recognized with the SNZI0 instruction)/“L” level	
		1	Rising waveform (“H” level of INT pin is recognized with the SNZI0 instruction)/“H” level	
$I1_1$	INT pin edge detection circuit control bit	0	One-sided edge detected	
		1	Both edges detected	
$I1_0$	INT 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	

Note 1.“R” represents read enabled, and “W” represents write enabled.
Note 2.When the contents of $I1_2$ and $I1_3$ are changed, the external interrupt request flag EXF0 may be set.

(3) Notes on interrupts

(1) Bit 3 of register I1
When the input of the INT 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 D_5/INT 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 (1) in Figure 29.) 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 (2) in Figure 29.).
Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to (3) in Figure 29.).

```
    ⋮
LA 4        ; (×××02)
TV1A        ; The SNZ0 instruction is valid ...... (1)
LA 8        ; (1×××2)
TI1A        ; Control of INT pin input is changed
NOP         ................................................... (2)
SNZ0        ; The SNZ0 instruction is executed
             (EXF0 flag cleared)
NOP         ................................................... (3)
    ⋮

×: these bits are not used here.
```

Fig 29. External 0 interrupt program example-1

(2) Bit 3 of register I1
When the bit 3 of register I1 is cleared to "0", the power down mode is selected and the input of INT pin is disabled, be careful about the following notes.

- When the INT pin input is disabled (register $I1_3$ = "0"), set the key-on wakeup of INT pin to be invalid (register $K2_0$ = "0") before system enters to power down mode. (refer to (1) in Figure 30.).

```
    ⋮
LA 0        ; (×××02)
TK2A        ; INT0 key-on wakeup disabled .....(1)
DI
EPOF
POF2        ; RAM back-up
    ⋮

×: these bits are not used here.
```

Fig 30. External 0 interrupt program example-2

(3) Bit 2 of register I1
When the interrupt valid waveform of the INT 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 D_5/INT 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 (1) in Figure 31.) and then, change the bit 2 of register I1 is changed.
In addition, execute the SNZ0 instruction to clear the EXF0 flag to "0" after executing at least one instruction (refer to (2) in Figure 31.).
Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to (3) in Figure 31.).

```
    ⋮
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 31. External 0 interrupt program example-3

TIMERS

The 4559 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.

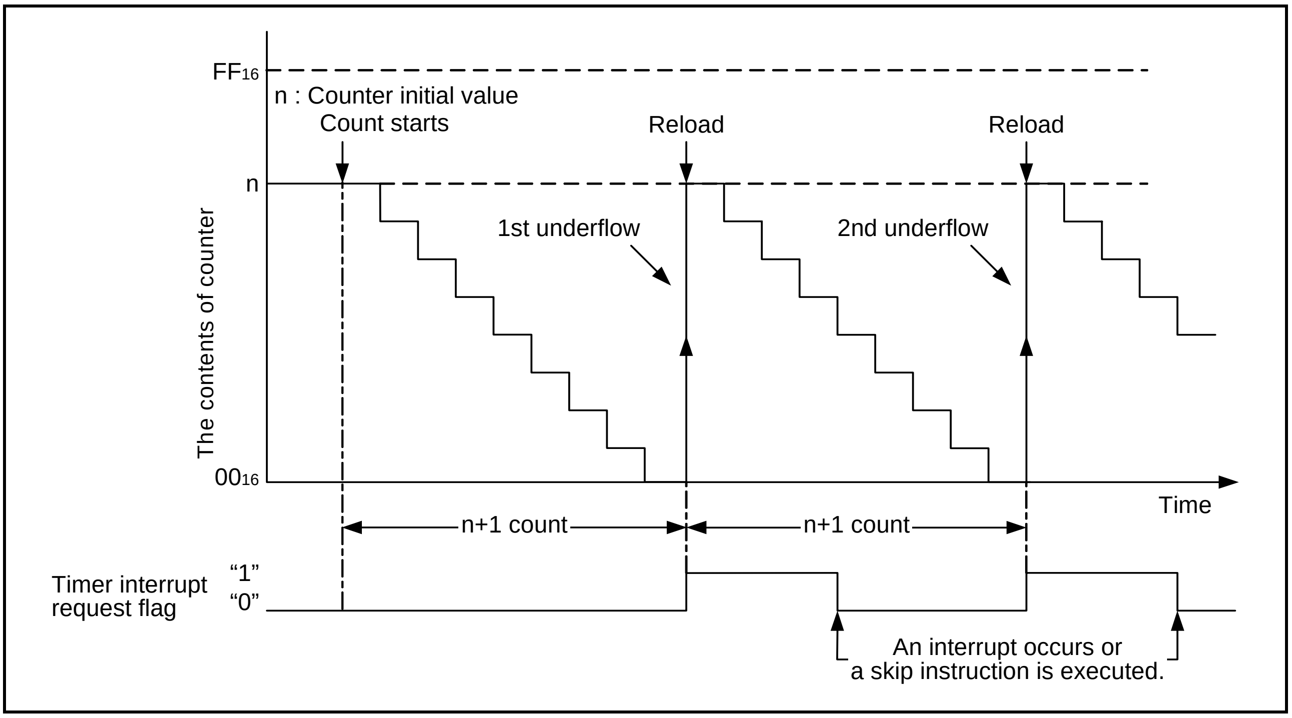


Fig 32. Auto-reload function

The 4559 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 : 16-bit fixed frequency timer
- Timer LC : 4-bit programmable timer
- Watchdog timer: 16-bit fixed frequency timer

(Timers 1, 2 and 3 have the interrupt function, respectively)

Prescaler, timer 1, timer 2, timer 3 and timer LC can be controlled with the timer control registers PA and W1 to W4. The watchdog timer is a free counter which is not controlled with the control register.
Each function is described below.

Table 16 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 count source • Timer 2 count source • Timer 3 count source	PA
Timer 1	8-bit programmable binary down counter (link to INT input) (carrier wave output auto-control function)	• PWM signal (PWMOUT) • Prescaler output (ORCLK) • Timer 3 underflow (T3UDF) • CNTR input	1 to 256	• CNTR output control • Timer 1 interrupt	W1 W4
Timer 2	8-bit programmable binary down counter (with carrier wave generation function)	• XIN input • Prescaler output divided by 2 (ORCLK/2)	1 to 256	• Timer 1 count source • CNTR output • Timer 2 interrupt	W2 W4
Timer 3	16-bit fixed dividing frequency	• XIN input • Prescaler output (ORCLK)	8192 16384 32768 65536	• Timer 1 count source • Timer LC count source • Timer 3 interrupt	W3
Timer LC	4-bit programmable binary down counter	• Bit 4 of timer 3 ($T3_4$) • System clock (STCK)	1 to 16	• LCD clock	W4
Watchdog timer	16-bit fixed dividing frequency	• Instruction clock (INSTCK)	65536	• System reset (counting twice) • Decision of flag WDF1	-

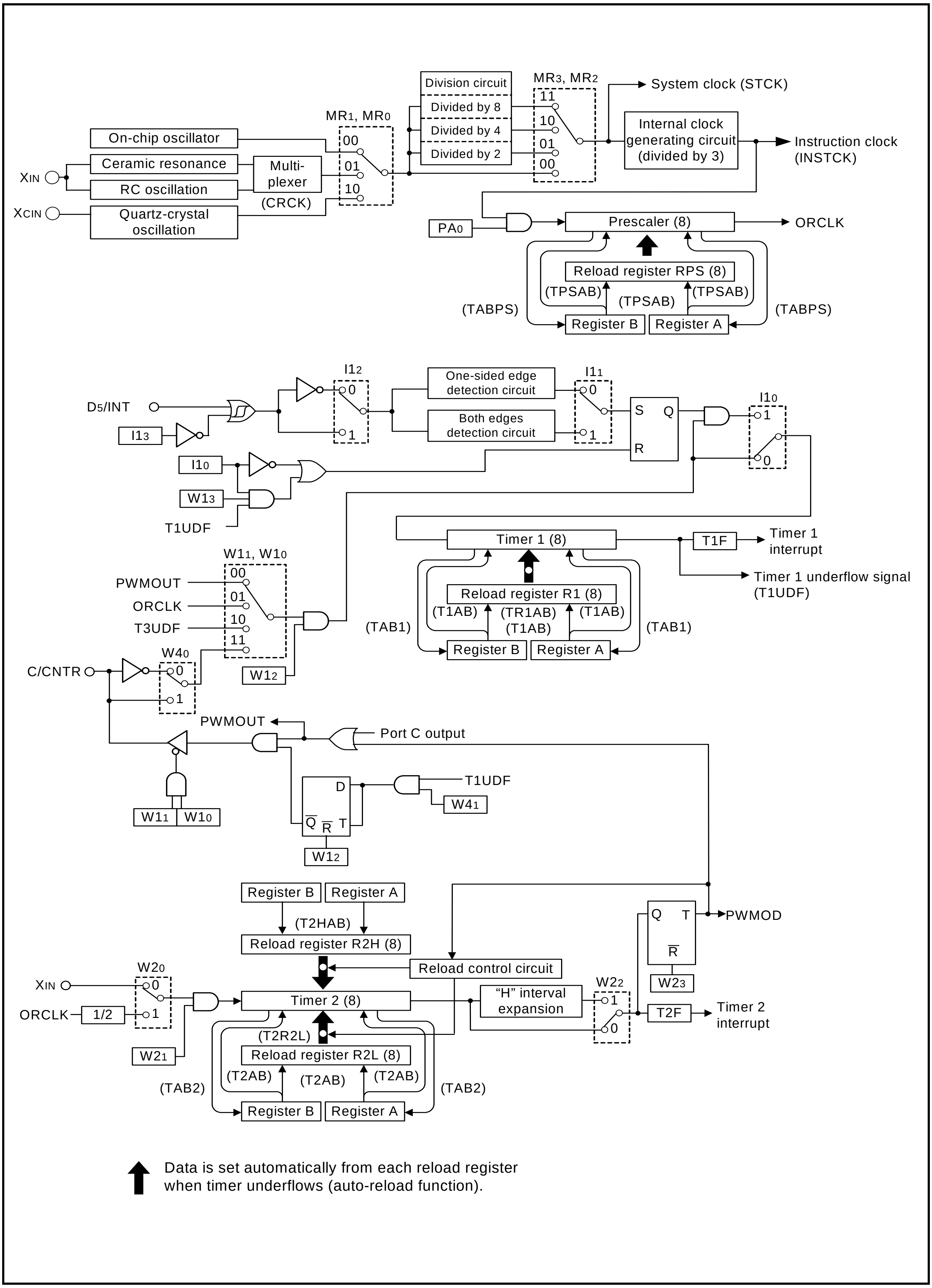
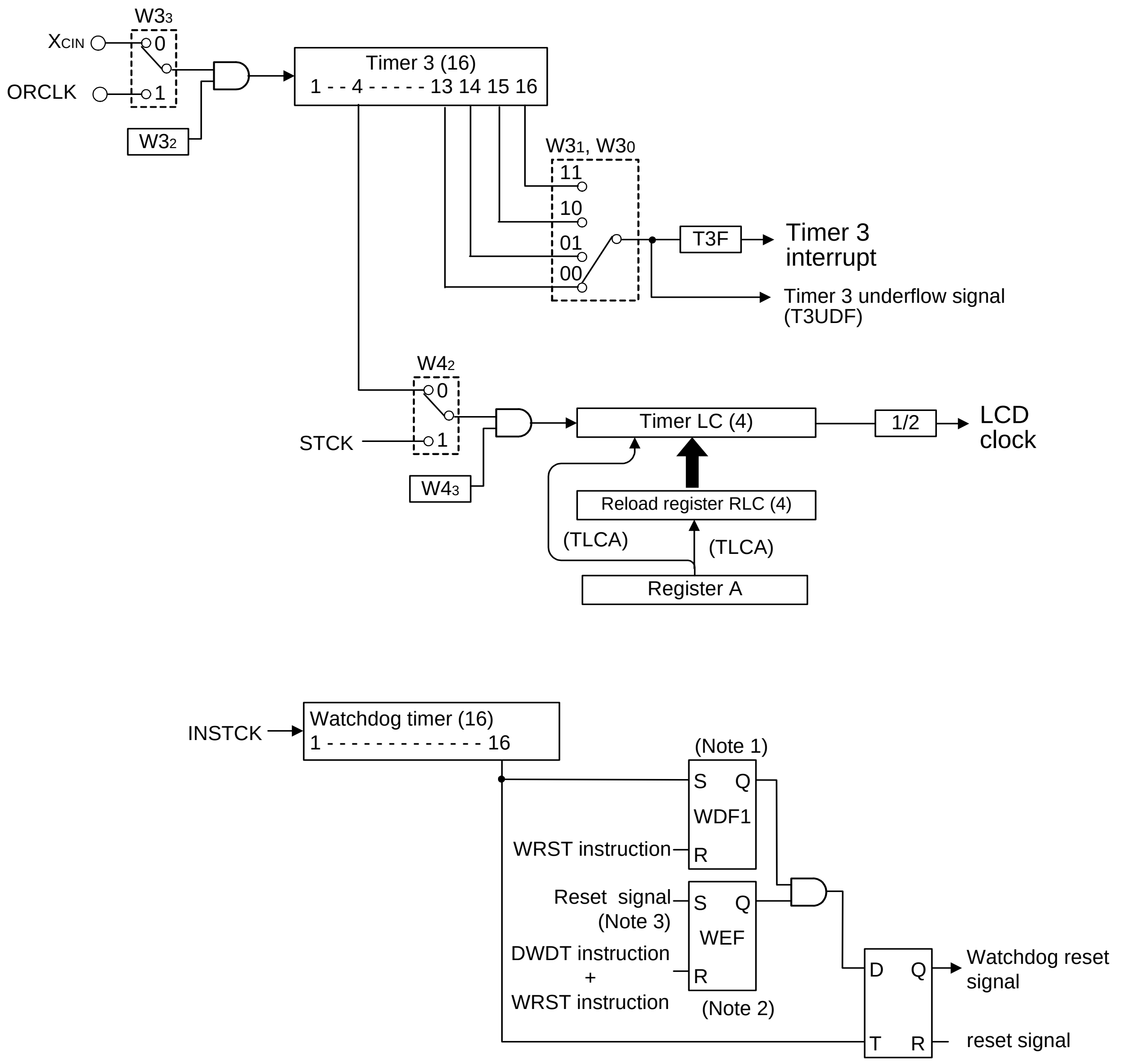


Fig 33. Timers structure (1)



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

Note 1: Flag WDF1 is cleared to “0” and the next instruction is skipped when the WRST instruction is executed while flag WDF1 = “1”.
The WRST instruction is equivalent to the NOP instruction while flag WDF1 = “0”.
2: Flag WEF is cleared to “0” and watchdog timer reset does not occur when the DWDT instruction and WRST instruction are executed continuously.
3: The WEF flag is set to “1” at system reset or RAM back-up mode.

Fig 34. Timers structure (2)

Table 17 Timer control registers

Timer control register PA		at reset : 0_2		at power down : 0_2	W TPAA
PA_0	Prescaler control bit	0	Stop (state retained)		
		1	Operating		

Timer control register W1		at reset : 0000_2			at power down : 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 (Note 3)	$W1_1$	$W1_0$	Count source		
		0	0	PWM signal (PWMOUT)		
$W1_0$		0	1	Prescaler output (ORCLK)		
		1	0	Timer 3 underflow signal (T3UDF)		
		1	1	CNTR input		

Timer control register W2		at reset : 0000_2		at power down : 0000_2	R/W TAW2/TW2A
$W2_3$	CNTR pin function control bit	0	CNTR pin output invalid		
		1	CNTR pin output valid		
$W2_2$	PWM signal “H” interval expansion function control bit	0	PWM signal “H” interval expansion function invalid		
		1	PWM signal “H” interval expansion function valid		
$W2_1$	Timer 2 control bit	0	Stop (state retained)		
		1	Operating		
$W2_0$	Timer 2 count source selection bit	0	X_{IN} input		
		1	Prescaler output (ORCLK)/2		

Timer control register W3		at reset : 0000_2			at power down : state retained	R/W TAW3/TW3A
$W3_3$	Timer 3 count source selection bit	0	X_{CIN} input			
		1	Prescaler output (ORCLK)			
$W3_2$	Timer 3 control bit	0	Stop (initial state)			
		1	Operating			
$W3_1$	Timer 3 count value selection bits	$W3_1$	$W3_0$	Count value		
		0	0	Underflow every 8192 count		
$W3_0$		0	1	Underflow every 16384 count		
		1	0	Underflow every 32768 count		
		1	1	Underflow every 65536 count		

Timer control register W4		at reset : 0000_2		at power down : state retained	R/W TAW4/TW4A
$W4_3$	Timer LC control bit	0	Stop (state retained)		
		1	Operating		
$W4_2$	Timer LC count source selection bit	0	Bit 4 ($T3_4$) of timer 3		
		1	System clock (STCK)		
$W4_1$	CNTR pin output auto-control circuit selection bit	0	CNTR output auto-control circuit not selected		
		1	CNTR output auto-control circuit selected		
$W4_0$	CNTR pin input count edge selection bit	0	Falling edge		
		1	Rising edge		

Note 1. “R” represents read enabled, and “W” represents write enabled.
Note 2. This function is valid only when the timer 1 control start synchronous circuit is selected ($I1_0$ =“1”).
Note 3. Port C output is invalid when CNTR input is selected for the timer 1 count source.

(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 count operation and count source of timer 1, and timer 1 count auto-stop circuit. Set the contents of this register through register A with the TW1A instruction. The TAW1 instruction can be used to transfer the contents of register W1 to register A.
- Timer control register W2
  Register W2 controls the count operation and count source of timer 2, CNTR pin output, and extension function of PWM signal "H" interval. Set the contents of this register through register A with the TW2A instruction. The TAW2 instruction can be used to transfer the contents of register W2 to register A.
- Timer control register W3
  Register W3 controls the count operation and count source of timer 3. 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 input count edge of CNTR pin, CNTR1 pin output auto-control circuit. Set the contents of this register through register A with the TW4A instruction. The TAW4 instruction can be used to transfer the contents of register W4 to register A.

(2) 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 can be selected 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 and 3 count sources.

(3) Timer 1 (interrupt function)

Timer 1 is an 8-bit binary down counter with a 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 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 bit 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).
INT 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."

(4) Timer 2 (interrupt function)

Timer 2 is an 8-bit binary down counter with two timer 2 reload register (R2L, R2H). Data can be set simultaneously in timer 2 and the reload register R2L with the T2AB instruction. Data can be set in the reload register R2H with the T2HAB instruction. The contents of reload register R2L set with the T2AB instruction can be set to timer 2 again with the T2R2L 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.
When executing the T2HAB instruction to set data to reload register R2H while timer 2 is operating, avoid a timing when timer 2 underflows.
Timer 2 starts counting after the following process;
(1) set data in timer 2
(2) set count source by bit 0 of register W2, and
(3) set the bit 1 of register W2 to "1."

When a value set in reload register R2L is n and R2H is m, timer 2 divides the count source signal by n + 1 or m + 1 (n = 0 to 255, m = 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 R2L, and count continues (auto-reload function).
When bit 3 of register W2 is set to "1", timer 2 reloads data from reload register R2L and R2H alternately each underflow.
Timer 2 generates the PWM signal (PWMOUT) of the "L" interval set as reload register R2L, and the "H" interval set as reload registerR2H. The PWM signal (PWMOUT) is output from CNTR pin. When bit 2 of register W2 is set to "1" at this time, the interval (PWM signal "H" interval) set to reload register R2H for the counter of timer 2 is extended for a half period of count source.
In this case, when a value set in reload register R2H is m, timer 2 divides the count source signal by n + 1.5 (m = 1 to 255).
When this function is used, set "1" or more to reload register R2H.

When bit 1 of register W4 is set to “1”, the PWM signal output to CNTR pin is switched to valid/invalid each timer 1 underflow. However, when timer 1 is stopped (bit 2 of register W1 is cleared to “0”), this function is canceled.
Even when bit 1 of a register W2 is cleared to “0” in the “H” interval of PWM signal, timer 2 does not stop until it next timer 2 underflow.
When clearing bit 1 of register W2 to “0” to stop timer 2, avoid a timing when timer 2 underflows.

(5) Timer 3 (interrupt function)

Timer 3 is a 16-bit binary down counter.
Timer 3 starts counting after the following process;
(1) set count value by bits 0 and 1 of register W3,
(2) set count source by bit 3 of register W3, and
(3) set the bit 2 of register W3 to “1.”

Once count is started, when timer 3 underflows (the set count value is counted), the timer 3 interrupt request flag (T3F) is set to “1,” and count continues.
Bit 4 of timer 3 can be used as the timer LC count source for the LCD clock generating.
When bit 2 of register W3 is cleared to “0”, timer 3 is initialized to “$FFFF_{16}$” and count is stopped.
Timer 3 can be used as the counter for clock because it can be operated at clock operating mode (POF instruction execution). When timer 3 underflow occurs at clock operating mode, system returns from the power down state.
When operating timer 3 during clock operating mode, set 1 cycle or more of count source to the following period; from setting bit 2 of register W3 to “1” till executing the POF instruction.

(6) Timer LC

Timer LC is a 4-bit binary down counter with the timer LC reload register (RLC). Data can be set simultaneously in timer LC and the reload register (RLC) with the TLCA instruction. Data cannot be read from timer LC. Stop counting and then execute the TLCA instruction to set timer LC data.
Timer LC starts counting after the following process;
(1) set data in timer LC,
(2) select the count source with the bit 2 of register W4, and
(3) set the bit 3 of register W4 to “1.”

When a value set in reload register RLC is n, timer LC divides the count source signal by n + 1 (n = 0 to 15).
Once count is started, when timer LC underflows (the next count pulse is input after the contents of timer LC becomes “0”), new data is loaded from reload register RLC, and count continues (auto-reload function).
Timer LC underflow signal divided by 2 can be used for the LCD clock.

(7) Timer input/output pin (C/CNTR pin)

CNTR pin is used to input the timer 1 count source and output the PWM signal generated by timer 2. The selection of CNTR output signal can be controlled by bit 3 of register W2.
When the PWM signal is output from C/CNTR pin, set “0” to the output latch of port C.
When the CNTR input is selected for timer 1 count source, timer 1 counts the waveform of CNTR input selected by bit 0 of register W4. Also, when the CNTR input is selected, the output of port C is invalid (high-impedance state).

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

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).
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.

(9) Count start synchronization circuit (timer 1)

Timer 1 has the count start synchronous circuit which synchronizes the input of INT 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 INT pin input can be performed.
When timer 1 count start synchronous circuit is used, the count start synchronous circuit is set, the count source is input to timer by inputting valid waveform to INT pin.
The valid waveform of INT 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 $I1_0$ to “0” or system reset.
However, when the count auto-stop circuit is selected, the count start synchronous circuit is cleared (auto-stop) at the timer 1 underflow.

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

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 circuit 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.

(11) Precautions

- Prescaler
  Stop prescaler counting and then execute the TABPS instruction to read its data.
  Stop prescaler counting and then execute the TPSAB instruction to write data to prescaler.
- Timer count source
  Stop timer 1, 2, 3 or LC counting to change its count source.
- Reading the count value
  Stop timer 1 or 2 counting and then execute the TAB1 or TAB2 instruction to read its data.
- Writing to the timer
  Stop timer 1, 2 or LC counting and then execute the T1AB, T2AB, T2R2L or TLCA instruction to write data to timer.
- Writing to reload register
  In order to write a data to the reload register R1 while the timer 1 is operating, execute the TR1AB instruction except a timing of the timer 1 underflow.
  In order to write a data to the reload register R2H while the timer 2 is operating, execute the T2HAB instruction except a timing of the timer 3 underflow.
- PWM signal
  If the timer 2 count stop timing and the timer 2 underflow timing overlap during output of the PWM signal, a hazard may occur in the PWM output waveform.
  When "H" interval expansion function of the PWM signal is used, set "1" or more to reload register R2H.
  Set the port C output latch to "0" to output the PWM signal from C/CNTR pin.
- Timer 3
  Stop timer 3 counting to change its count source.
  When operating timer 3 during clock operating mode, set 1 cycle or more of count source to the following period; from setting bit 2 of register W3 to "1" till executing the POF instruction.
- Prescaler and timer 1 count start timing and count time when operation starts
  Count starts from the first rising edge of the count source (2) in Figure 35 after prescaler and timer operations start (1) in Figure 35.
  Time to first underflow (3) in Figure 35 is shorter (for up to 1 period of the count source) than time among next underflow (4) in Figure 35 by the timing to start the timer and count source operations after count starts.
  When selecting CNTR input as the count source of timer 1, timer 1 operates synchronizing with the falling edge of CNTR input.

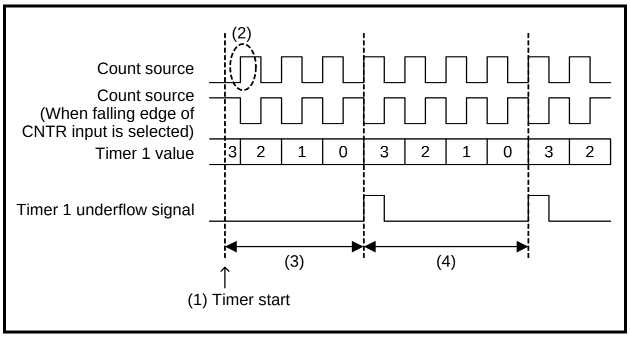


Fig 35. Timer count start timing and count time when operation starts

- Timer 2 and Timer LC 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 2 and Timer LC 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.

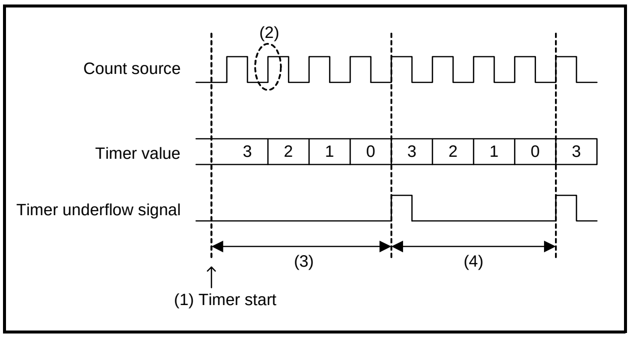


Fig 36. Timer count start timing and count time when operation starts (Timer 2 and Timer LC)

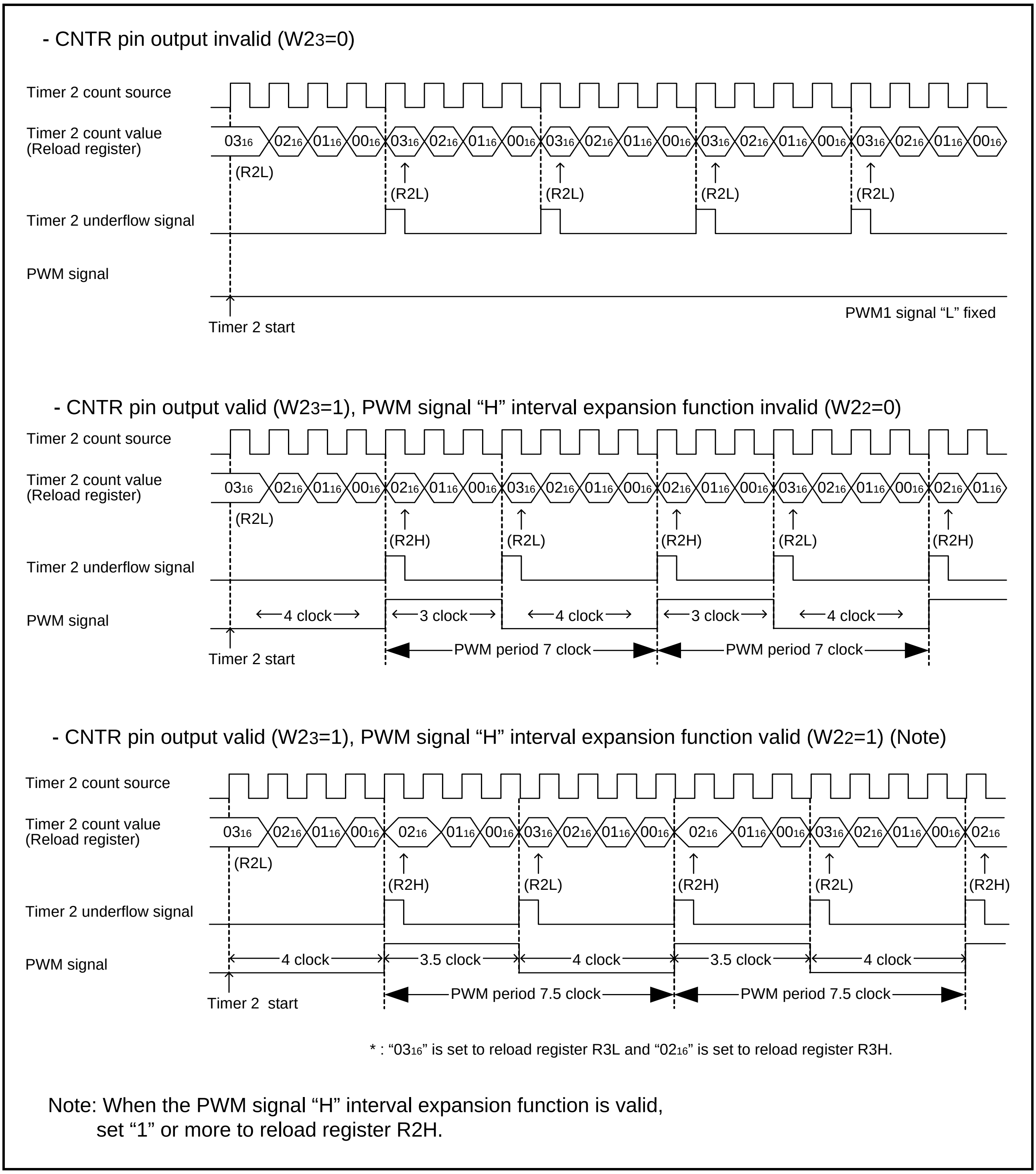
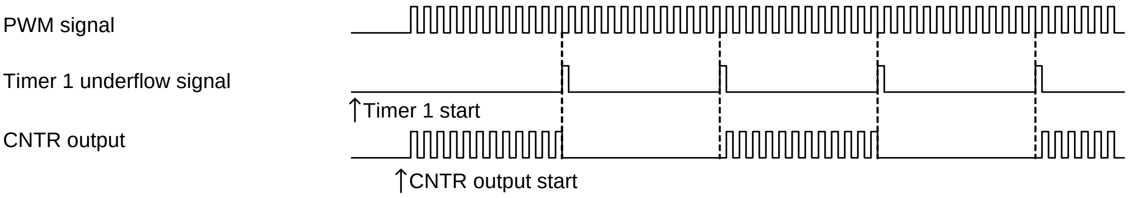


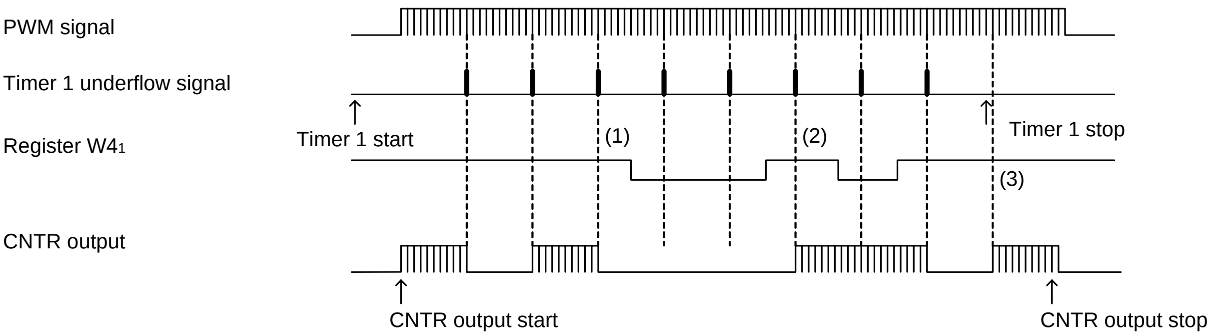
Fig 37. Timer 2 operation example

• CNTR output auto-control circuit operation example 1 ($W2_3$ = “1”, $W4_1$ = “1”)



* When the CNTR1 output auto-control circuit is selected, valid/invalid of CNTR output is repeated every timer 1 underflows.

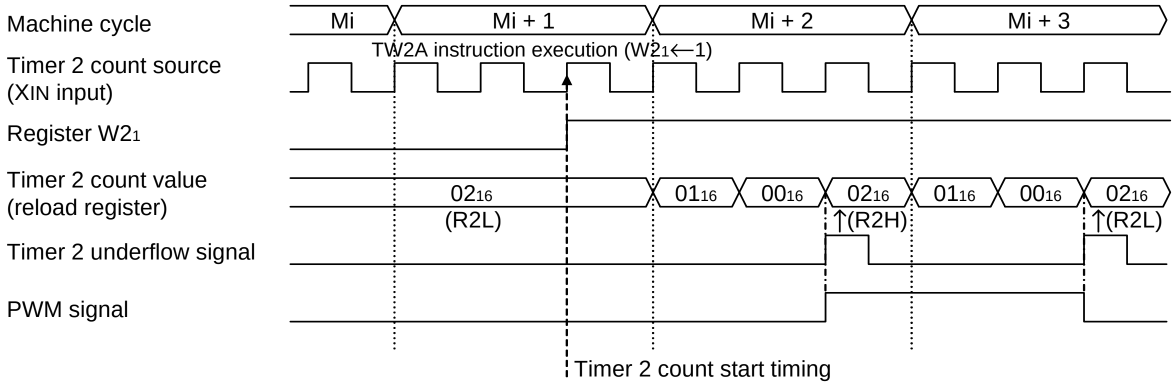
• CNTR output auto-control circuit operation example 2 ($W2_3$ = “1”)



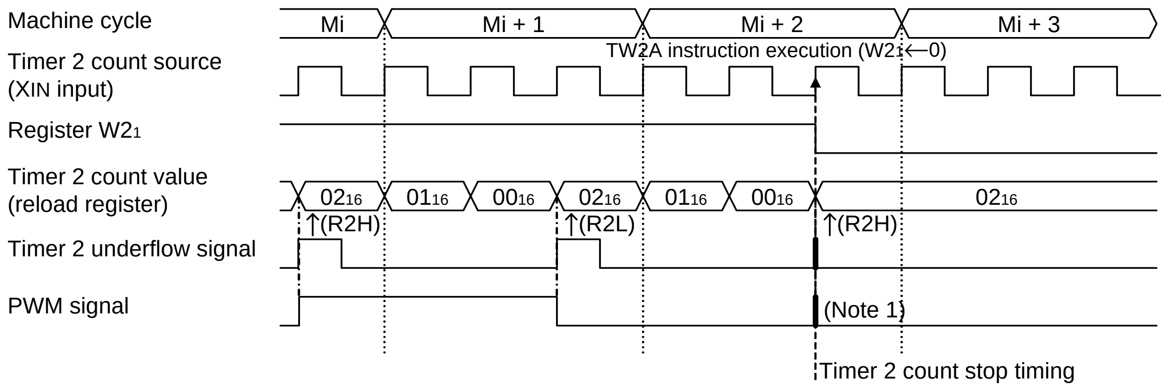
(1) When the CNTR output auto-control function is not selected while the CNTR output is invalid, CNTR output invalid state is retained.
(2) When the CNTR output auto-control function is not selected while the CNTR output is valid, CNTR output valid state is retained.
(3) When the timer 1 is stopped, the CNTR output auto-control function becomes invalid.

Fig 38. CNTR output auto-control function by timer 1

Timer 2 count start timing (R2L = “02_{16}”, R2H = “02_{16}”, $W2_3$ = “1”)



Timer 2 count stop timing (R2L = “02_{16}”, R2H = “02_{16}”, $W2_3$ = “1”)



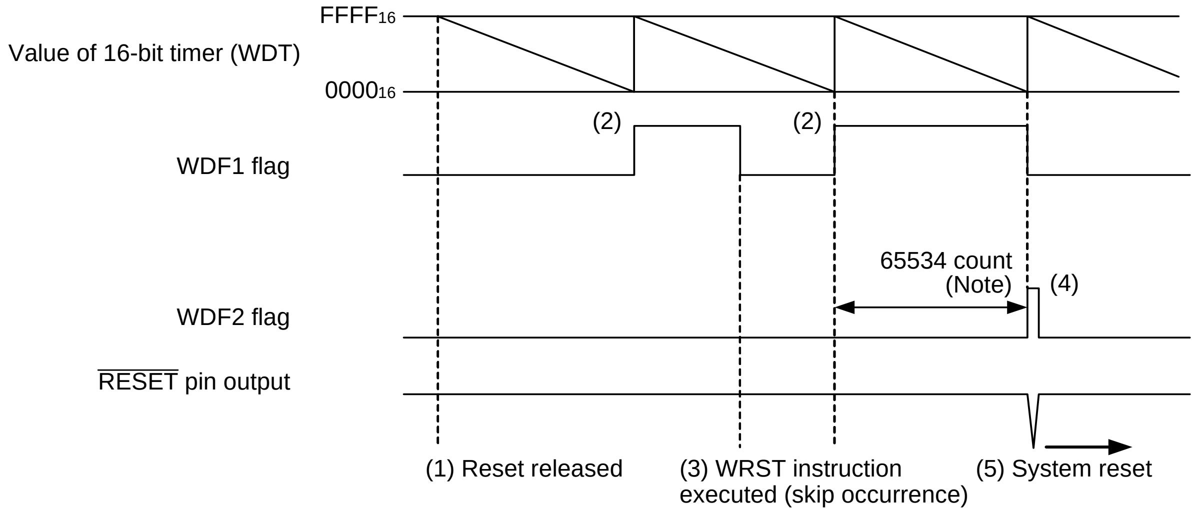
Notes 1: If the timer count stop timing and the timer underflow timing overlap while the CNTR pin output is valid ($W2_3$=“1”), a hazard may occur in the PWM signal waveform.

2: When timer count is stopped during “H” interval of the PWM signal, timer is stopped after the end of the “H” output interval.

Fig 39. Timer count start/stop timing

WATCHDOG TIMER

Watchdog timer provides a method to reset the system when a program run-away occurs. Watchdog timer consists of timer WDT(16-bit binary counter), watchdog timer enable flag (WEF), and watchdog timer flags (WDF1, WDF2).
The timer WDT downcounts the instruction clocks as the count source from "FFFF$_{16}$" after system is released from reset.
After the count is started, when the timer WDT underflow occurs (after the count value of timer WDT reaches "0000$_{16}$," the next count pulse is input), the WDF1 flag is set to "1." If the WRST instruction is never executed until the timer WDT underflow occurs (until timer WDT counts 65534), WDF2 flag is set to "1," and the $\overline{\text{RESET}}$ pin outputs "L" level to reset the microcomputer.
Execute the WRST instruction at each period of 65534 machine cycle or less by software when using watchdog timer to keep the microcomputer operating normally.
When the WEF flag is set to "1" after system is released from reset, the watchdog timer function is valid.
When the DWDT instruction and the WRST instruction are executed continuously, the WEF flag is cleared to "0" and the watchdog timer function is invalid.
The WEF flag is set to "1" at system reset or RAM back-up mode.
The WRST instruction has the skip function. When the WRST instruction is executed while the WDF1 flag is "1", the WDF1 flag is cleared to "0" and the next instruction is skipped.
When the WRST instruction is executed while the WDF1 flag is "0", the next instruction is not skipped.
The skip function of the WRST instruction can be used even when the watchdog timer function is invalid.



(1) After system is released from reset (= after program is started), timer WDT starts count down.
(2) When timer WDT underflow occurs, WDF1 flag is set to "1."
(3) When the WRST instruction is executed while the WDF1 flag is "1", 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 machine cycle because the count source of watchdog timer is the instruction clock.

Fig 40. 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 41).
The watchdog timer is not stopped with only the DWDT instruction.
The contents of WDF1 flag and timer WDT are initialized at the power down mode.
When using the watchdog timer and the power down mode, initialize the WDF1 flag with the WRST instruction just before the microcomputer enters the power down mode. Also, set the NOP instruction after the WRST instruction, for the case when a skip is performed with the WRST instruction (refer to Figure 42).

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

Fig 41. Program example to start/stop watchdog timer

```
     ⋮
WRST        ; WDF1 flag cleared
NOP
DI          ; Interrupt disabled
EPOF        ; POF instruction enabled
POF2        ; RAM back-up mode
     ↓
Oscillation stop
     ⋮
```

Fig 42. Program example when using the watchdog timer

LCD FUNCTION

The 4559 Group has an LCD (Liquid Crystal Display) controller/driver. When data are set in LCD RAM and timer LC, LCD control registers (L1, L2, L3, C1, C2, C3), and timer control registers (W3, W4), the LCD controller/driver automatically reads the display data and controls the LCD display by setting duty and bias.

4 common signal output pins and 32 segment signal output pins can be used to drive the LCD. By using these pins, up to 128 pixels (when internal power, 1/4 duty and 1/3 bias are selected) can be controlled to display. When using the external input, set necessary pins with the LCD control register 2 and apply the proper voltage to the pins .

The LCD power input pins (V_{LC3}–V_{LC1}) are also used as pins SEG$_0$–SEG$_2$. When SEG$_0$ is selected, the internal power (V_{DD}) is used for the LCD power.

(1) Duty and bias

There are 3 combinations of duty and bias for displaying data on the LCD. Use bits 0 and 1 of LCD control register (L1) to select the proper display method for the LCD panel being used.

- 1/2 duty, 1/2 bias
- 1/3 duty, 1/3 bias
- 1/4 duty, 1/3 bias

Table 18 Table 11 Duty and maximum number of displayed pixels

Duty	Maximum number of displayed pixels	Used COM pins
1/2	64 pixels	COM$_0$, COM$_1$ (Note)
1/3	96 pixels	COM$_0$–COM$_2$ (Note)
1/4	128 pixels	COM$_0$–COM$_3$

Note. Leave unused COM pins open.

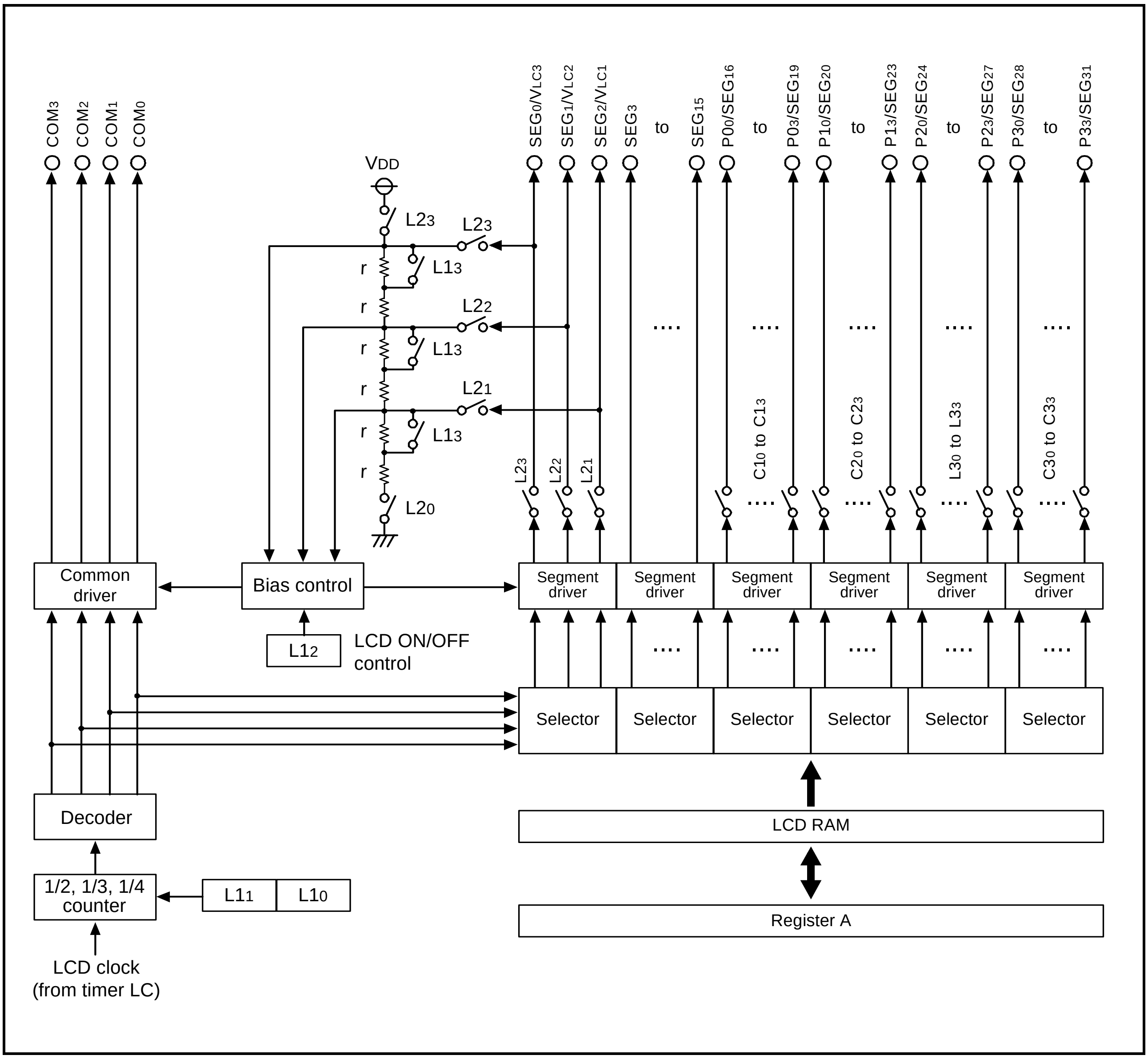


Fig 43. LCD controller/driver

(2) LCD clock control

The LCD clock is determined by the timer LC setting value and timer LC count source.
After setting data to timer LC, timer LC starts counting by setting count source with bit 2 of register W4 and setting bit 3 of register W4 to "1."
Accordingly, the frequency (F) of the LCD clock is obtained by the following formula. Numbers ((1) to (3)) shown below the formula correspond to numbers in Figure 44, respectively.

• When using the system clock (STCK) as timer LC count source ($W4_2$="1")

$$F = \underbrace{STCK}_{(1)} \times \underbrace{\frac{1}{LC+1}}_{(2)} \times \underbrace{\frac{1}{2}}_{(3)}$$

[LC: 0 to 15]

• When using the bit 4 of timer 3 as timer LC count source ($W4_2$="0")

$$F = \underbrace{T3_4}_{(1)} \times \underbrace{\frac{1}{LC+1}}_{(2)} \times \underbrace{\frac{1}{2}}_{(3)}$$

[LC: 0 to 15]

The frame frequency and frame period for each display method can be obtained by the following formula:

$$\text{Frame frequency} = \frac{F}{n} \text{ (Hz)}$$

$$\text{Frame frequency} = \frac{n}{F} \text{ (Hz)}$$

[F: LCD clock frequency
1/n: Duty]

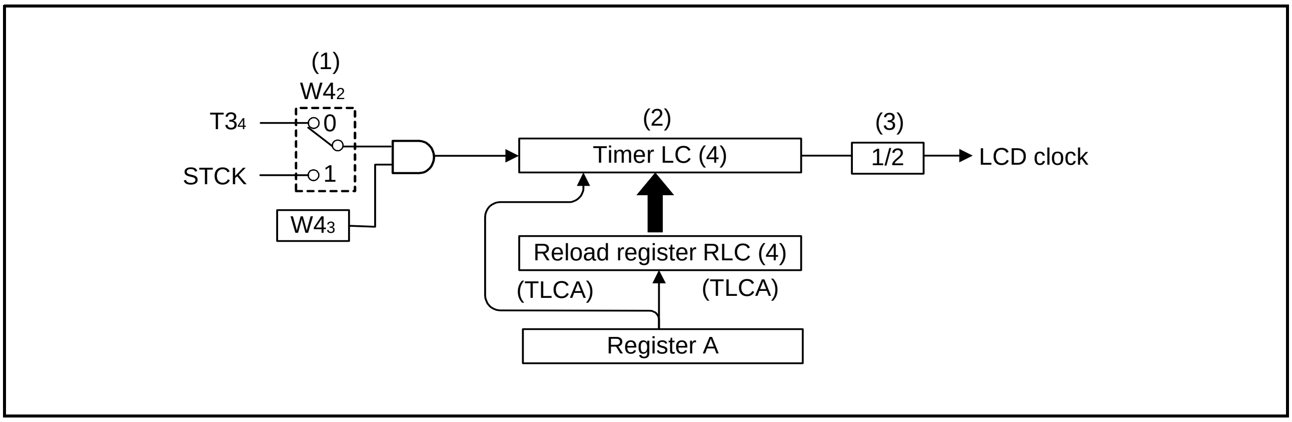


Fig 44. LCD clock control circuit structure

(3) LCD RAM

RAM contains areas corresponding to the liquid crystal display. When "1" is written to this LCD RAM, the display pixel corresponding to the bit is automatically displayed.

Z	1															
X	0				1				2				3			
Y \ bit	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0
8	SEG0	SEG0	SEG0	SEG0	SEG8	SEG8	SEG8	SEG8	SEG16	SEG16	SEG16	SEG16	SEG24	SEG24	SEG24	SEG24
9	SEG1	SEG1	SEG1	SEG1	SEG9	SEG9	SEG9	SEG9	SEG17	SEG17	SEG17	SEG17	SEG25	SEG25	SEG25	SEG25
10	SEG2	SEG2	SEG2	SEG2	SEG10	SEG10	SEG10	SEG10	SEG18	SEG18	SEG18	SEG18	SEG26	SEG26	SEG26	SEG26
11	SEG3	SEG3	SEG3	SEG3	SEG11	SEG11	SEG11	SEG11	SEG19	SEG19	SEG19	SEG19	SEG27	SEG27	SEG27	SEG27
12	SEG4	SEG4	SEG4	SEG4	SEG12	SEG12	SEG12	SEG12	SEG20	SEG20	SEG20	SEG20	SEG28	SEG28	SEG28	SEG28
13	SEG5	SEG5	SEG5	SEG5	SEG13	SEG13	SEG13	SEG13	SEG21	SEG21	SEG21	SEG21	SEG29	SEG29	SEG29	SEG29
14	SEG6	SEG6	SEG6	SEG6	SEG14	SEG14	SEG14	SEG14	SEG22	SEG22	SEG22	SEG22	SEG30	SEG30	SEG30	SEG30
15	SEG7	SEG7	SEG7	SEG7	SEG15	SEG15	SEG15	SEG15	SEG23	SEG23	SEG23	SEG23	SEG31	SEG31	SEG31	SEG31
COM	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0

Fig 45. LCD RAM map

(4) LCD drive waveform

When "1" is written to a bit in the LCD RAM data, the voltage difference between common pin and segment pin which correspond to the bit automatically becomes $|V_{LC3}|$ and the display pixel at the cross section turns on.

When returning from reset, and in the RAM back-up mode, a display pixel turns off because every segment output pin and common output pin becomes V_{LC3} level.

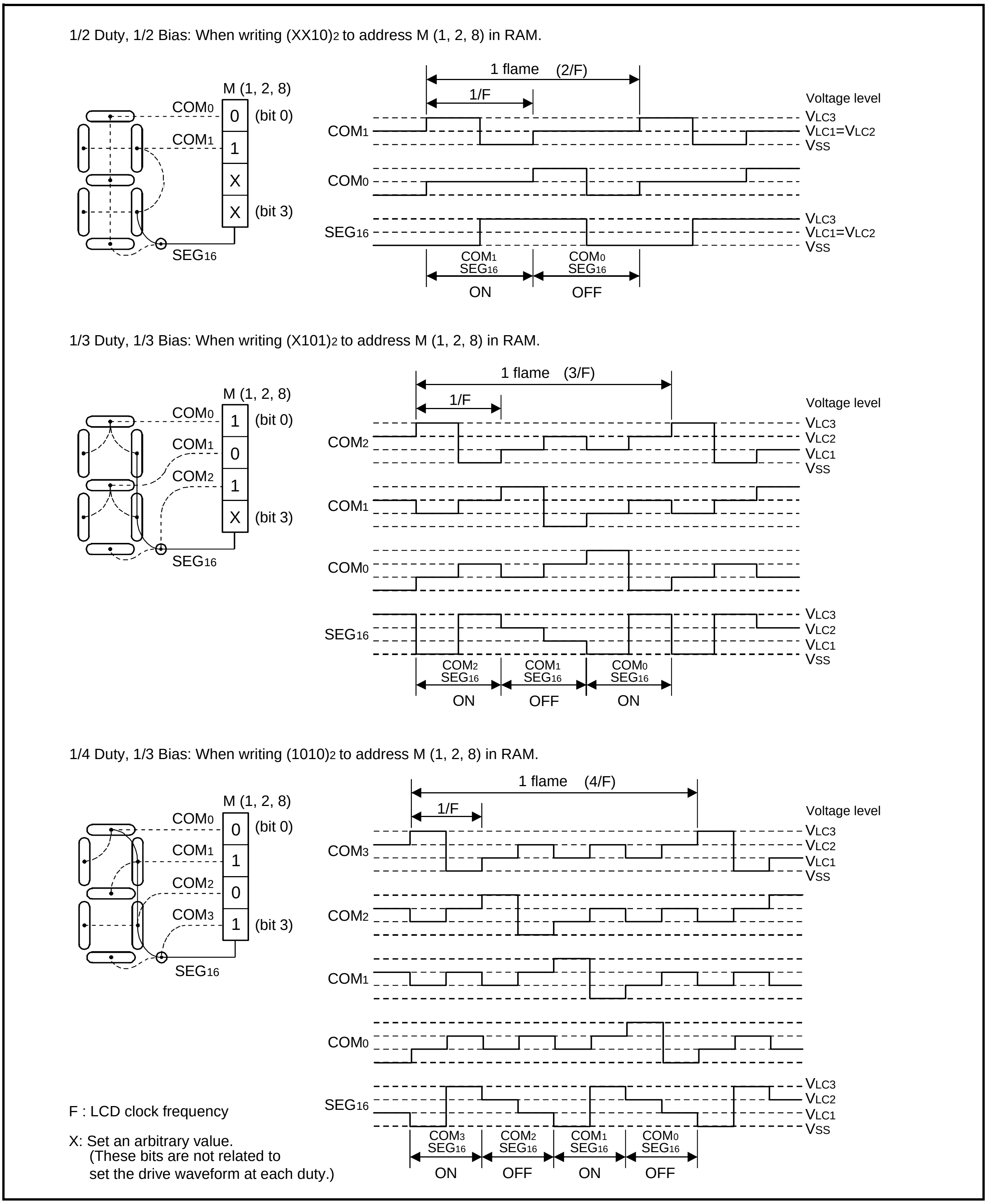


Fig 46. LCD controller/driver structure

(5) LCD power supply circuit

Select the LCD power supply circuit suitable for the using LCD panel.

The LCD power supply circuit is fixed by the followings;

- The internal dividing resistor is controlled by bit 0 of register L2.
- The internal dividing resistor is selected by bit 3 of register L1.
- The bias condition is selected by bits 0 and 1 of register L1.

- Internal dividing resistor

The 4553 Group has the internal dividing resistor for LCD power supply.

When bit 0 of register L2 is set to ì0î, the internal dividing resistor is valid. However, when the LCD is turned off by setting bit 2 of register L1 to ì0î, the internal dividing resistor is turned off.

The same six resistor (r) is prepared for the internal dividing resistor.

According to the setting value of bit 3 of register L1 and using bias condition, the resistor is prepared as follows;

- $L1_3$ = “0”, 1/3 bias used: 2r × 3 = 6r
- $L1_3$ = “0”, 1/2 bias used: 2r × 2 = 4r
- $L1_3$ = “1”, 1/3 bias used: r × 3 = 3r
- $L1_3$ = “1”, 1/2 bias used: r × 2 = 2r

- SEG_0/V_{LC3} pin

The selection of SEG_0/V_{LC3} pin function is controlled with the bit 3 of register L2.

When the V_{LC3} pin function is selected, apply voltage of $V_{LC3} < V_{DD}$ to the pin externally.

When the SEG_0 pin function is selected, V_{LC3} is connected to V_{DD} internally.

- SEG_1/V_{LC2}, SEG_2/V_{LC1} pin

The selection of SEG_1/V_{LC2} pin function is controlled with the bit 2 of register L2.

The selection of SEG_2/V_{LC1} pin function is controlled with the bit 1 of register L2.

When the V_{LC2} pin and V_{LC1} pin functions are selected and the internal dividing resistor is not used, apply voltage of $0 < V_{LC1} < V_{LC2} < V_{LC3}$ to these pins. Short the V_{LC2} pin and V_{LC1} pin at 1/2 bias.

When the V_{LC2} pin and V_{LC1} pin functions are selected and the internal dividing resistor is used, the dividing voltage value generated internally is output from the V_{LC1} pin and V_{LC2} pin. The V_{LC2} pin and V_{LC1} pin have the same electric potential at 1/2 bias.

When SEG1 and SEG2 pin functions are selected, use the internal dividing resistor ($L2_0$ = ”0”). In this time, V_{LC2} and V_{LC1} are connected to the generated dividing voltage.

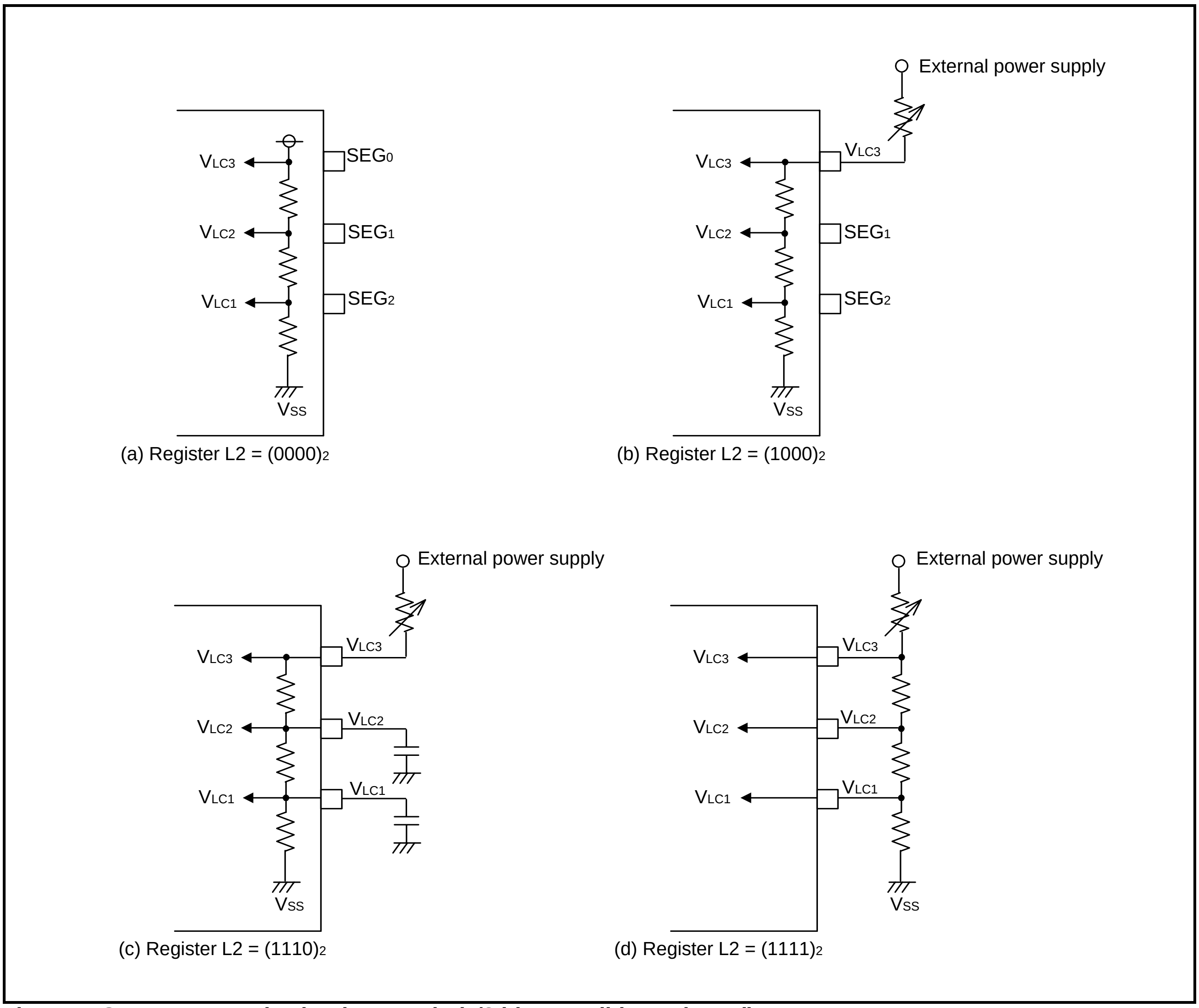


Fig 47. LCD power supply circuit example (1/3 bias condition selected)

(6) LCD control register

• LCD control register L1

Register L1 controls duty/bias selection, LCD operation, internal dividing resistor selection. Set the contents of this register through register A with the TL1A instruction. The TAL1 instruction can be used to transfer the contents of register L1.

• LCD control register L2

Register L2 controls internal dividing resistor operation, selection of pin functions; SEG_0/V_{LC3}, SEG_1/V_{LC2}, SEG_2/V_{LC1}. Set the contents of this register through register A with the TL2A instruction.

• LCD control register L3

Register L3 controls selection of pin functions; $P2_0/SEG_{24}$ to $P2_3/SEG_{27}$. Set the contents of this register through register A with the TL3A instruction.

• LCD control register C1

Register C1 controls selection of pin functions; $P0_0/SEG_{16}$ to $P0_3/SEG_{19}$. Set the contents of this register through register A with the TC1A instruction.

• LCD control register C2

Register C2 controls selection of pin functions; $P1_0/SEG_{20}$ to $P1_3/SEG_{23}$. Set the contents of this register through register A with the TC2A instruction.

• LCD control register C3

Register C3 controls selection of pin functions; $P3_0/SEG_{28}$ to $P3_3/SEG_{31}$. The contents of this register through register A with the TC3A instruction.

Table 19 LCD control registers (1)

LCD control register L1		at reset : 0000_2		at power down : state retained		R/W TAL1/TL1A
$L1_3$	Internal dividing resistor for LCD power supply selection bit (Note 2)	0	2r × 3, 2r × 2			
		1	r × 3, r × 2			
$L1_2$	LCD control bit	0	Stop (OFF)			
		1	Operating			
$L1_1$	LCD duty and bias selection bits	$L1_1$	$L1_0$	Duty	Bias	
		0	0	Not available	Not available	
		0	1	1/2	1/2	
$L1_0$		1	0	1/3	1/3	
		1	1	1/4	1/3	

LCD control register L2		at reset : 0000_2		at power down : state retained	W TL2A
$L2_3$	SEG_0/V_{LC3} pin function switch bit (Note 3)	0	SEG_0		
		1	V_{LC3}		
$L2_2$	SEG_1/V_{LC2} pin function switch bit (Note 4)	0	SEG_1		
		1	V_{LC2}		
$L2_1$	SEG_2/V_{LC1} pin function switch bit (Note 4)	0	SEG_2		
		1	V_{LC1}		
$L2_0$	Internal dividing resistor for LCD power supply control bit	0	Internal dividing resistor valid		
		1	Internal dividing resistor invalid		

LCD control register L3		at reset : 1111_2		at power down : state retained	W TL3A
$L3_3$	$P2_3/SEG_{27}$ pin function switch bit	0	SEG_{27}		
		1	$P2_3$		
$L3_2$	$P2_2/SEG_{26}$ pin function switch bit	0	SEG_{26}		
		1	$P2_2$		
$L3_1$	$P2_1/SEG_{25}$ pin function switch bit	0	SEG_{25}		
		1	$P2_1$		
$L3_0$	$P2_0/SEG_{24}$ pin function switch bit	0	SEG_{24}		
		1	$P2_0$		

Note 1.“R” represents read enabled, and “W” represents write enabled.
Note 2.“r (resistor) multiplied by 3” is used at 1/3 bias, and “r multiplied by 2” is used at 1/2 bias.
Note 3.V_{LC3} is connected to V_{DD} internally when SEG_0 pin is selected.
Note 4.Use internal dividing resistor when SEG_1 and SEG_2 pins are selected.

Table 20 LCD control registers (2)

LCD control register C1		at reset : 1111_2	at power down : state retained	W TC1A
$C1_3$	$P0_3/SEG_{19}$ pin function switch bit	0	SEG_{19}	
		1	$P0_3$	
$C1_2$	$P0_2/SEG_{18}$ pin function switch bit	0	SEG_{18}	
		1	$P0_2$	
$C1_1$	$P0_1/SEG_{17}$ pin function switch bit	0	SEG_{17}	
		1	$P0_1$	
$C1_0$	$P0_0/SEG_{16}$ pin function switch bit	0	SEG_{16}	
		1	$P0_0$	

LCD control register C2		at reset : 1111_2	at power down : state retained	W TC2A
$C2_3$	$P1_3/SEG_{23}$ pin function switch bit	0	SEG_{23}	
		1	$P1_3$	
$C2_2$	$P1_2/SEG_{22}$ pin function switch bit	0	SEG_{22}	
		1	$P1_2$	
$C2_1$	$P1_1/SEG_{21}$ pin function switch bit	0	SEG_{21}	
		1	$P1_1$	
$C2_0$	$P1_0/SEG_{20}$ pin function switch bit	0	SEG_{20}	
		1	$P0_0$	

LCD control register C3		at reset : 1111_2	at power down : state retained	W TC3A
$C3_3$	$P3_3/SEG_{31}$ pin function switch bit	0	SEG_{31}	
		1	$P3_3$	
$C3_2$	$P3_2/SEG_{30}$ pin function switch bit	0	SEG_{30}	
		1	$P3_2$	
$C3_1$	$P3_1/SEG_{29}$ pin function switch bit	0	SEG_{29}	
		1	$P3_1$	
$C3_0$	$P3_0/SEG_{28}$ pin function switch bit	0	SEG_{28}	
		1	$P3_0$	

Note 1."R" represents read enabled, and "W" represents write enabled.

RESET FUNCTION

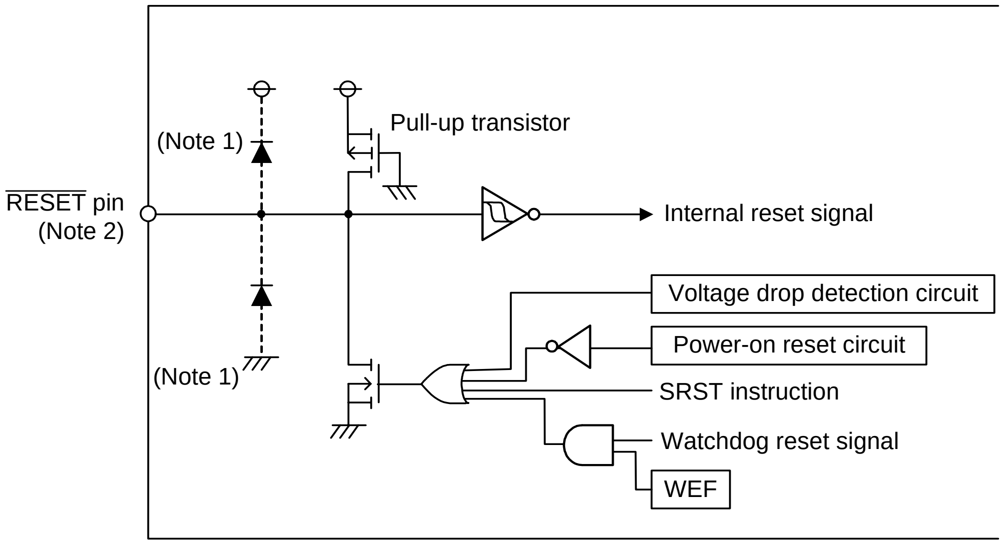
System reset is performed by the followings:

- "L" level is applied to the RESET pin externally,
- System reset instruction (SRST) is executed,
- Reset occurs by watchdog timer,
- Reset occurs by built-in power-on reset
- Reset occurs by voltage drop detection circuit

Then when "H" level is applied to RESET pin, software starts from address 0 in page 0.

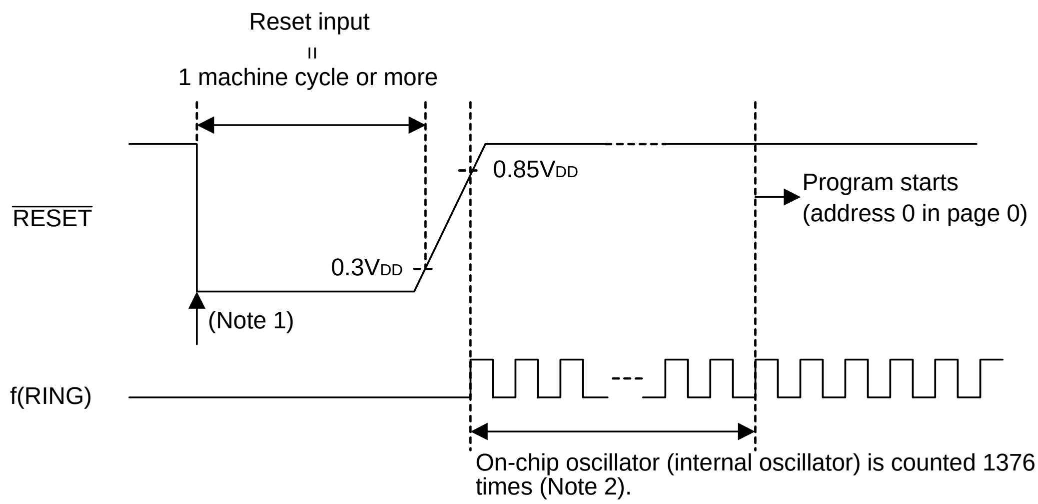
(1) RESET pin input

System reset is performed certainly by applying "L" level to 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.



Notes 1: This symbol represents a parasitic diode.
2: Applied potential to RESET pin must be V_{DD} or less.

Fig 48. Structure of RESET pin and its peripherals



Notes 1: Keep the value of supply voltage to the minimum value or more of the recommended operating conditions.
2: It depends on the internal state at reset.

Fig 49. RESET pin input waveform and reset release timing

(2) 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, set the time for the supply voltage to rise from 0 V to the minimum voltage of recommended operating conditions 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.

(3) System reset instruction (SRST)

By executing the SRST instruction, "L" level is output to $\overline{\text{RESET}}$ pin and system reset is performed.

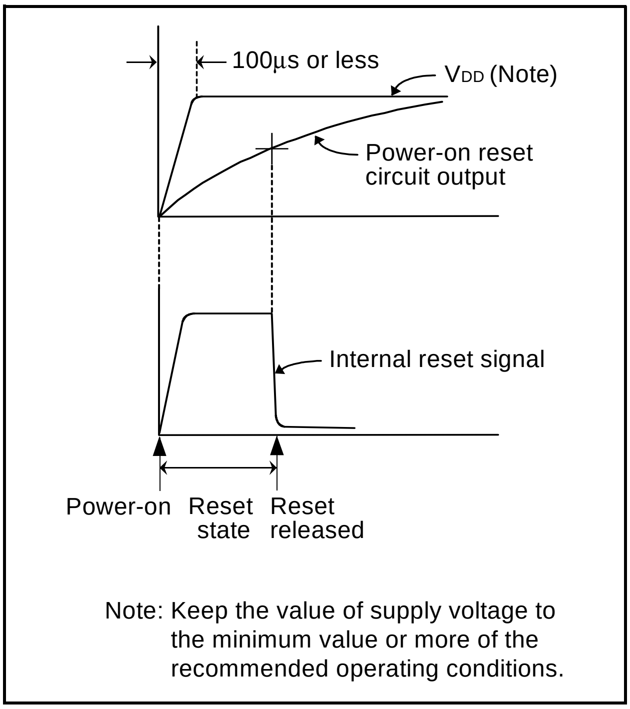


Fig 50. Power-on reset operation

Table 21 Port state at reset

Name	Function	State
D_0–D_4	D_0–D_4	High-impedance (Notes 1, 2)
D_5/INT	D_5	High-impedance (Notes 1, 2)
X_{CIN}/D_6, X_{COUT}/D_7	X_{CIN}, X_{COUT}	Sub-clock input
$P0_0$/SEG_{16}–$P0_3$/SEG_{19}	$P0_0$–$P0_3$	High-impedance (Notes 1, 2, 3)
$P1_0$/SEG_{20}–$P1_3$/SEG_{23}	$P1_0$–$P1_3$	High-impedance (Notes 1, 2, 3)
$P2_0$/SEG_{24}–$P2_3$/SEG_{27}	$P2_0$–$P2_3$	High-impedance (Notes 1, 2, 3)
$P3_0$/SEG_{28}–$P3_3$/SEG_{31}	$P3_0$–$P3_3$	High-impedance (Notes 1, 2, 3)
SEG_0/V_{LC3}–SEG_2/V_{LC1}	SEG_0–SEG_2	V_{LC3} (V_{DD}) level
SEG_3–SEG_{15}	SEG_3–SEG_{15}	V_{LC3} (V_{DD}) level
COM_0–COM_3	COM_0–COM_3	V_{LC3} (V_{DD}) level
C/CNTR	C/CNTR	"L" (V_{SS}) level

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

(4) Internal state at reset

Figure 51 and 52 shows internal state at reset (they are the same after system is released from reset). The contents of timers, registers, flags and RAM except shown in Figure 51 and 52 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	
• 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	
• Timer 1 interrupt request flag (T1F)	0	
• Timer 2 interrupt request flag (T2F)	0	
• Timer 3 interrupt request flag (T3F)	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 LC stopped)
• Clock control register MR	1 1 0 0	
• Clock control register RG	0 0 0	
• LCD control register L1	0 0 0 0	
• LCD control register L2	0 0 0 0	
• LCD control register L3	1 1 1 1	
• LCD control register C1	1 1 1 1	
• LCD control register C2	1 1 1 1	
• LCD control register C3	1 1 1 1	

Fig 51. Internal state at reset (1)

• 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
• Key-on wakeup control register K3	0 0 0 0
• Pull-up control register PU0	0 0 0 0
• Pull-up control register PU1	0 0 0 0
• Pull-up control register PU2	0 0 0 0
• Pull-up control register PU3	0 0 0 0
• Port output structure control register FR0	0 0 0 0
• Port output structure control register FR1	0 0 0 0
• Port output structure control register FR2	0 0 0 0
• Port output structure control register FR3	0 0 0 0
• High-order bit reference enable flag (UPTF)	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 (oeprating)
• Ceramic resonator circuit	Oeprating
• RC oscillation circuit	Stop
• Quartz-crystal oscillator	Oeprating

"X" represents undefined.

Fig 52. Internal state at reset (2)

VOLTAGE DROP DETECTION CIRCUIT (WITH SKIP JUDGMENT)

The built-in voltage drop detection circuit is used to set the voltage drop detection circuit flag (VDF) or to perform system reset.

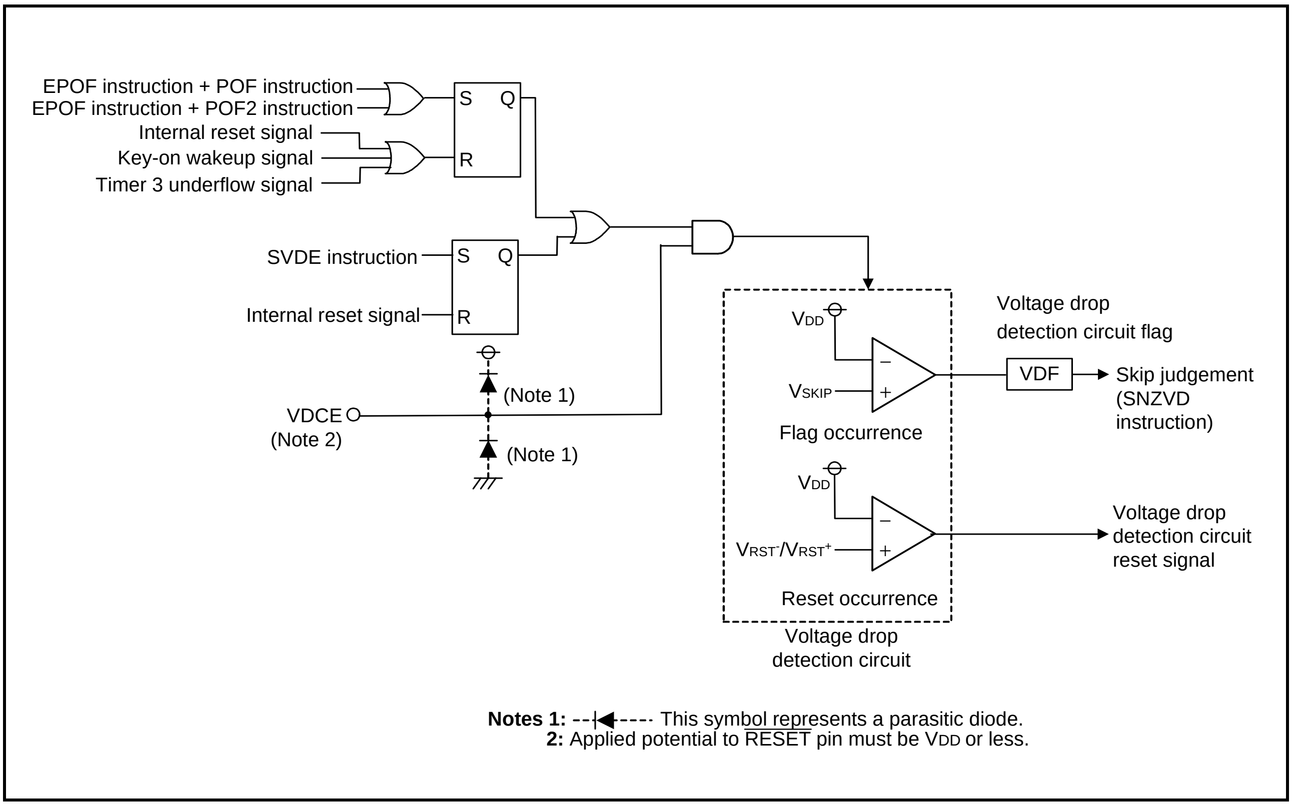


Fig 53. Voltage drop detection reset circuit

(1) Operating state of voltage drop detection circuit

The voltage drop detection circuit becomes valid by inputting "H" to the VDCE pin and it becomes invalid by inputting "L."
When not executing the SVDE instruction under "H" level of the VDCE pin, the voltage drop detection circuit become invalid in power down state (RAM back-up, clock operating mode). As for this, the voltage drop detection circuit becomes valid at returning from power down, again.
When executing the SVDE instruction under "H" level of the VDCE pin, the voltage drop detection circuit becomes valid in power down state (RAM back-up, clock operating mode).
The state of executing SVDE instruction can be cleared by system reset.

Table 22 Operating state of voltage drop detection circuit

VDCE pin	SVDE instruction	at CPU operating	at power down
"L"	No execute	×	×
	Execute	×	×
"H"	No execute	O	×
	Execute	O	O

Note. "O" indicates valid, "×" indicates invalid.

(2) Voltage drop detection circuit flag (VDF)

Voltage drop detection circuit flag (VDF) is set to “1” when the supply voltage goes the skip occurrence voltage (V_{SKIP}) or less. Moreover, voltage drop detection circuit flag (VDF) is cleared to “0” when the supply voltage goes the skip occurrence voltage (V_{SKIP}) or more. The state of the voltage drop detection circuit flag (VDF) can be examined with the skip instruction (SNZVD). Even when the skip instruction is executed, the voltage drop detection circuit flag is not cleared to “0”.
Refer to the electrical characteristics for skip occurrence voltage value.

(3) Voltage drop detection circuit reset

System reset is performed when the supply voltage goes the reset occurrence voltage (V_{RST}^-) or less.
When the supply voltage goes reset release voltage (V_{RST}^+) or more, the oscillation circuit goes to be in the operating enabled state and system reset is released .
Refer to the electrical characteristics for reset occurrence value and reset release voltage value.

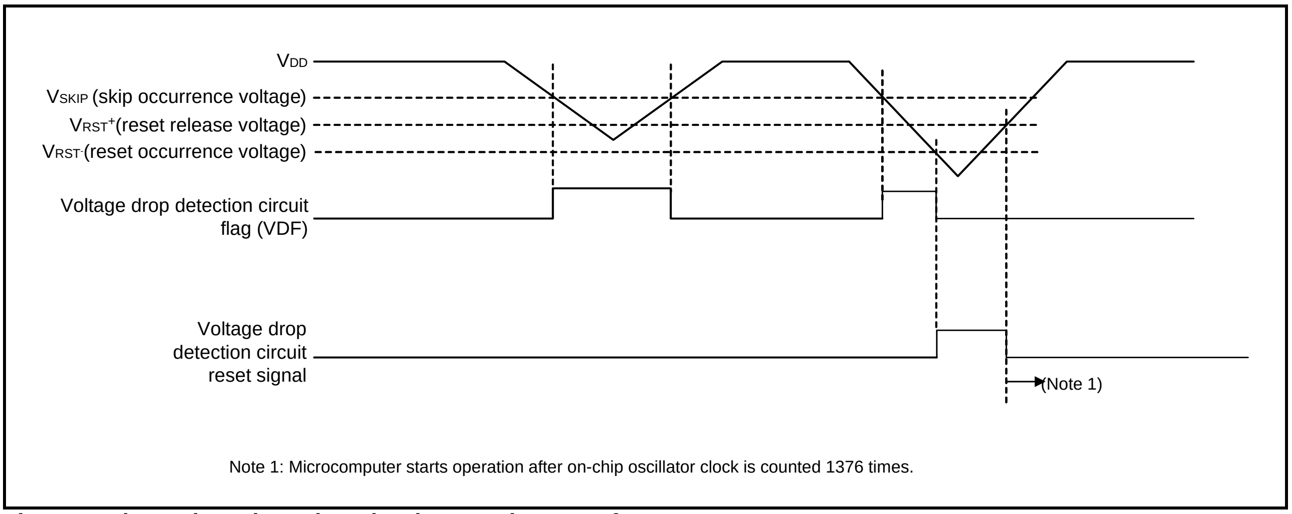


Fig 54. Voltage drop detection circuit operation waveform

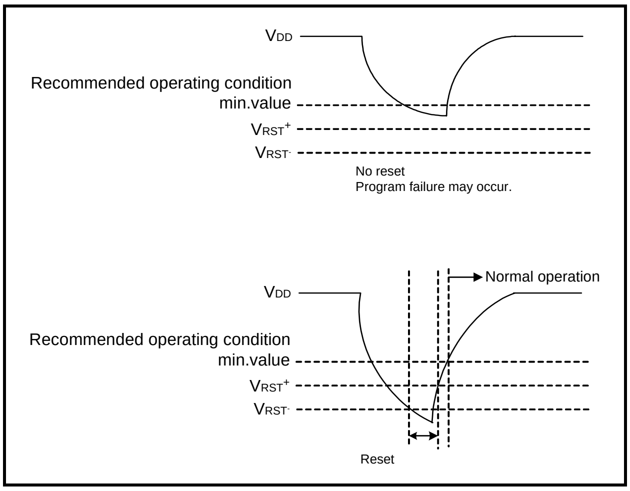


Fig 55. V_{DD} and V_{RST}^-

(4) Note on voltage drop detection circuit

The voltage drop detection circuit detection voltage of this product is set up lower than the minimum value of the supply voltage of the recommended operating conditions.
When the supply voltage of a microcomputer falls below to the minimum value of recommended operating conditions and regoes up, depending on the capacity value of the bypass capacitor added to the power supply pin, the following case may cause program failure (Figure 55);
supply voltage does not fall below to V_{RST}^-, and its voltage re-goes up with no reset.
In such a case, please design a system which supply voltage is once reduced below to V_{RST}^- and re-goes up after that.

POWER DOWN FUNCTION

The 4559 Group has 2-type power down functions.
System enters into each power down state by executing the following instructions.

- Clock operating mode .... EPOF and POF instructions
- RAM back-up mode .... EPOF and POF2 instructions

When the EPOF instruction is not executed before the POF or POF2 instruction is executed, these instructions are equivalent to the NOP instruction.

(1) Clock operating mode

The following functions and states are retained.
- RAM
- Reset circuit
- XCIN–XCOUT oscillation
- LCD display
- Timer 3

(2) RAM back-up mode

The following functions and states are retained.
- RAM
- Reset circuit

(3) Warm start condition

The system returns from the power down state when;
- External wakeup signal is input
- Timer 3 underflow occurs

in the power down mode.
In either case, the CPU starts executing the software from address 0 in page 0. In this case, the P flag is "1."

(4) Cold start condition

The CPU starts executing the software from address 0 in page 0 when;
- external "L" level is input to $\overline{\text{RESET}}$ pin,
- execute system reset instruction (SRST instruction)
- reset by watchdog timer is performed
- reset by internal power-on reset, or
- reset by the voltage drop detection circuit is performed.

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

(5) Identification of the start condition

Warm start or cold start can be identified by examining the state of the power down flag (P) with the SNZP instruction. The warm start condition from the clock operating mode can be identified by examining the state of T3F flag.

Table 23 Functions and states retained at power down mode

Function	Power down mode	
	Clock operating	RAM back-up
Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) (Note 2)	×	×
Contents of RAM	O	O
Interrupt control registers V1, V2	×	×
Interrupt control registers I1	O	O
Selected oscillation circuit	O	O
Clock control register MR, RG	O	O
Timer 1, Timer 2 functions	(Note 3)	(Note 3)
Timer 3 function	O	(Note 3)
Timer LC function	O	(Note 3)
Watchdog timer function	× (Note 4)	× (Note 4)
Timer control registers PA, WA	×	×
Timer control registers W1, W3, W4	O	O
LCD display function	O	(Note 5)
LCD control registers L1 to L3, C1 to C3	O	O
Voltage drop detection circuit	(Note 6)	(Note 6)
Port level	(Note 7)	(Note 7)
Key-on wakeup control registers K0 to K2	O	O
Pull-up control registers PU0, PU1	O	O
Port output structure control registers FR0 to FR2	O	O
External interrupt request flags (EXF0)	×	×
Timer interrupt request flags (T1F, T2F)	(Note 3)	(Note 3)
Timer interrupt request flag (T3F)	O	(Note 3)
Interrupt enable flag (INTE)	×	×
Voltage drop detection circuit flag (VDF)	×	×
Watchdog timer flags (WDF1, WDF2)	× (Note 4)	× (Note 4)
Watchdog timer enable flag (WEF)	× (Note 4)	× (Note 4)

Note 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 power down mode, and set an initial value after returning.
Note 2. The stack pointer (SP) points the level of the stack register and is initialized to "7" at power down mode.
Note 3. The state of the timer is undefined.
Note 4. Initialize the WDF1 flag with the WRST instruction, and then go into the power down state.
Note 5. LCD is turned off.
Note 6. When the SVDE instruction is executed, this function is valid at power down.
Note 7. In the power down mode, C/CNTR pin outputs "L" level. However, when the CNTR input is selected ($W1_1$, $W1_0$="11"), C/CNTR pin is in an input enabled state (output = high-impedance).
Other ports retain their respective output levels.

(6) Return signal

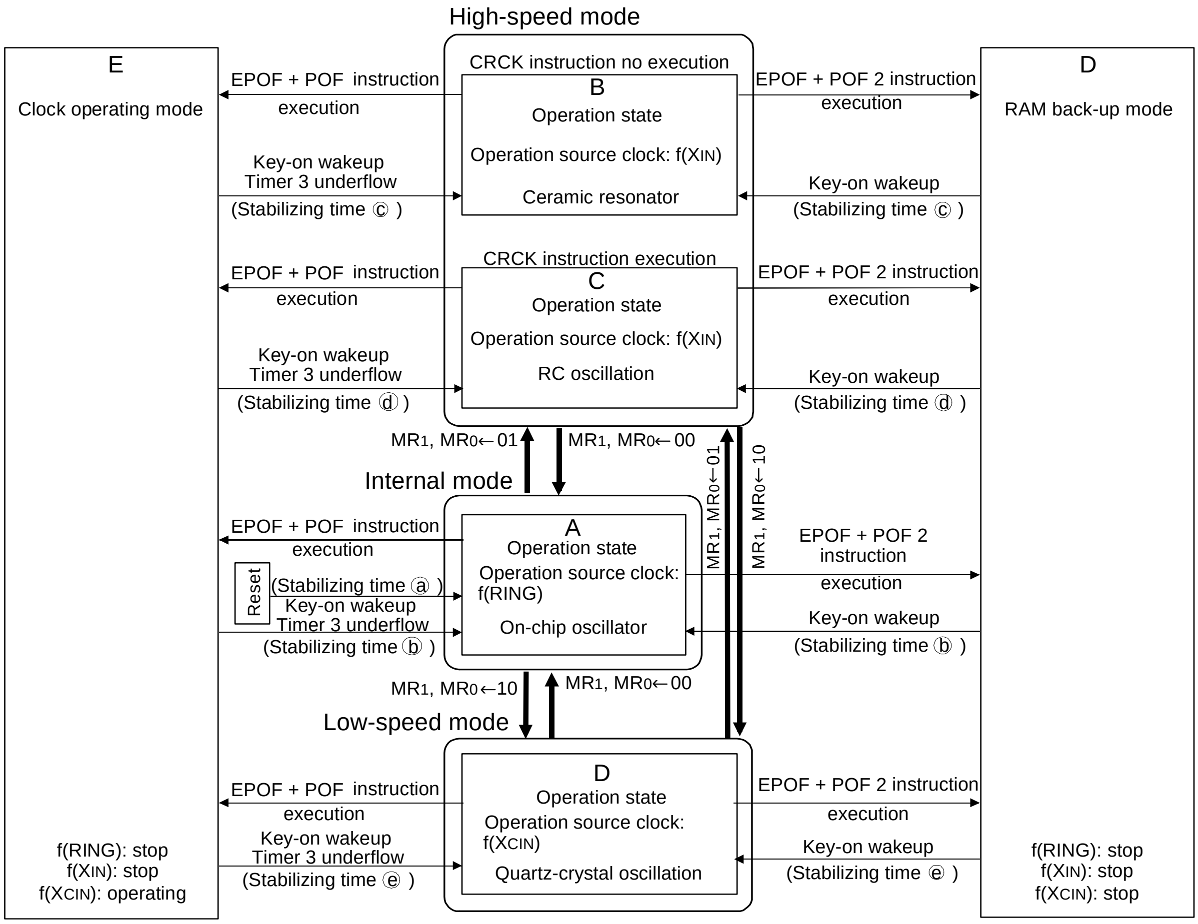
An external wakeup signal or timer 3 interrupt request flag (T3F) is used to return from the clock operating mode.
An external wakeup signal is used to return from the RAM back-up mode because the oscillation is stopped.
Table 24 shows the return condition for each return source.

(7) Control registers

- Key-on wakeup control register K0
  Register K0 controls the ports P0 and P1 key-on wakeup function. Set the contents of this register through register A with the TK0A instruction. In addition, the TAK0 instruction can be used to transfer the contents of register K0 to register A.
- Key-on wakeup control register K1
  Register K1 controls the return condition and the selection of valid waveform/level of port P1. 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 K0 to register A.
- Key-on wakeup control register K2
  Register K2 controls the port P3 and INT pin key-on wakeup function and the selection of return condition of INT pin. 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.
- Key-on wakeup control register K3
  Register K3 controls the selection of return condition and valid waveform/level of port P3. Set the contents of this register through register A with the TK3A instruction. In addition, the TAK3 instruction can be used to transfer the contents of register K3 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 PU1 to register A.
- Pull-up control register PU2
  Register PU2 controls the ON/OFF of the ports P2 pull-up transistor. Set the contents of this register through register A with the TPU2A instruction. In addition, the TAPU2 instruction can be used to transfer the contents of register PU2 to register A.
- Pull-up control register PU3
  Register PU3 controls the ON/OFF of the ports P3 pull-up transistor. Set the contents of this register through register A with the TPU3A instruction. In addition, the TAPU3 instruction can be used to transfer the contents of register PU3 to register A.
- External interrupt control register I1
  Register I1 controls the input control and the selection of valid waveform/level of INT pin. Set the contents of this register through register A with the TI1A instruction. In addition, the TAI1 instruction can be used to transfer the contents of register I1 to register A.

Table 24 Return source and return condition

Return source		Return condition	Remarks
External wakeup signal	Ports $P0_0$–$P0_3$ Ports $P1_0$–$P1_3$ Ports $P2_0$–$P2_3$	Return by an external falling edge (“H” → “L”).	For ports P0 and P1, the key-on wakeup function can be selected by two port unit, for port P2, it can be selected by a unit.
External wakeup signal	Ports $P3_0$–$P3_3$	Return by an external “H” level or “L” level input, or rising edge (“L” → “H”) or falling edge (“H” → “L”). Return by an external “L” level input,	The key-on wakeup function can be selected by two port unit. Select the return level (“L” level or “H” level) and return condition (return by level or edge) with register K3 according to the external state before going into the power down state.
External wakeup signal	INT pin	Return by an external “H” level or “L” level input, or rising edge (“L” → “H”) or falling edge (“H” → “L”). When the return level is input, the interrupt request flag (EXF0) is not set.	Select the return level (“L” level or “H” level) with register I1 and return condition (return by level or edge) with register K2 according to the external state before going into the power down state.
Timer 3 interrupt request flag (T3F)		Return by timer 3 underflow or by setting T3F to “1”. It can be used in the clock operating mode.	Clear T3F with the SNZT3 instruction before system enters into the power down state. When system enters into the power down state while T3F is “1”, system returns from the state immediately because it is recognized as return condition.



Stabilizing time a: Microcomputer starts its operation after counting the f(RING) to 1376 times.
Stabilizing time b: Microcomputer starts its operation after counting the f(RING) to (system clock division ratio X 171) times.
Stabilizing time c: Microcomputer starts its operation after counting the f(XIN) to (system clock division ratio X 171) times.
Stabilizing time d: Microcomputer starts its operation after counting the f(XIN) to (system clock division ratio X 171) times.
Stabilizing time e: Microcomputer starts its operation after counting the f(XCIN) to (system clock division ratio X 171) times.

Notes 1: Selection of the system clock by the clock control registers MR and RG is state retained at power down. The waiting time to stabilize oscillation at return can be adjustment by setting the clock control registers MR and RG before transition to the power down state.
2: Continuous execution of the EPOF instruction and the POF instruction is required to go into the clock operating state.
3: Continuous execution of the EPOF instruction and the POF2 instruction is required to go into the RAM back-up state.
4: The state after system is released from reset;
• A ceramic resonator is selected as the main clock (f(XIN)).
• Main clock (f(XIN)) and Sub-clock (f(XCIN)) are valid.
5: When the RC oscillation circuit is used, executing the CRCK instruction is required. If the CRCK instruction is not executed, the ceramic resonator is selected as the main clock f(XIN).
6: When the unoperating clock is selected as the system clock, turn it on by the clock control register RG, and generate the wait time until the oscillation is stabilized, and then, switch the system clock.
7: The Sub-clock (quartz-crystal oscillation) is operating except in state D.

Fig 56. State transition

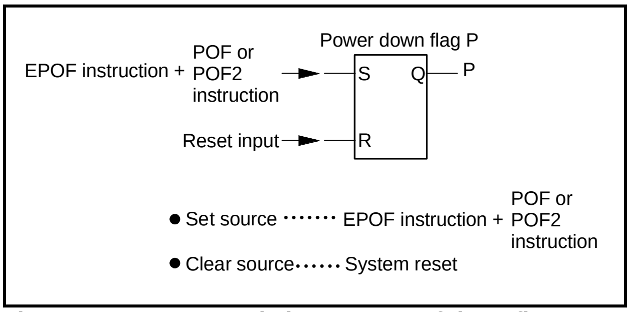


Fig 57. Set source and clear source of the P flag

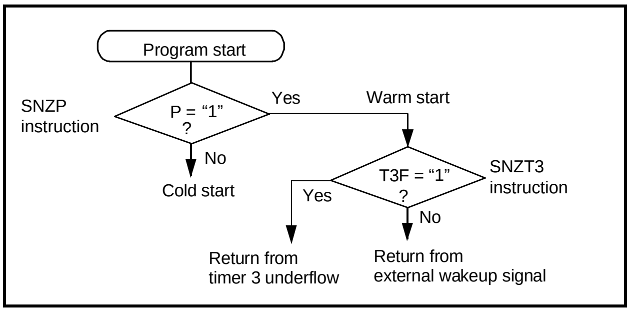


Fig 58. Start condition identified example using the SNZP instruction

Table 25 Key-on wakeup control register

Key-on wakeup control register K0		at reset : 0000_2		at power down : state retained	R/W TAK0/TK0A
$K0_3$	Ports $P1_2$, $P1_3$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_2$	Ports $P1_0$, $P1_1$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_1$	Ports $P0_2$, $P0_3$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_0$	Ports $P0_0$, $P0_1$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K1		at reset : 0000_2		at power down : state retained	R/W TAK1/TK1A
$K1_3$	Port $P2_3$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K1_2$	Port $P2_2$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K1_1$	Port $P2_1$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K1_0$	Port $P2_0$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K2		at reset : 0000_2		at power down : state retained	R/W TAK2/TK2A
$K2_3$	Ports $P3_2$, $P3_3$ key-on wakeup control bit (Note 3)	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K2_2$	Ports $P3_0$, $P3_1$ key-on wakeup control bit (Note 2)	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K2_1$	INT pin return condition selection bit	0	Return by level		
		1	Return by edge		
$K2_0$	INT pin key-on wakeup control bit	0	Key-on wakeup invalid		
		1	Key-on wakeup valid		

Key-on wakeup control register K3		at reset : 0000_2		at power down : state retained	R/W TAK3/TK3A
$K3_3$	Ports $P3_2$, $P3_3$ return condition selection bit (Note 3)	0	Return by level		
		1	Return by edge		
$K3_2$	Ports $P3_2$, $P3_3$ valid waveform/level selection bit (Note 3)	0	Falling waveform/"L" level		
		1	Rising waveform/"H" level		
$K3_1$	Ports $P3_0$, $P3_1$ return condition selection bit (Note 2)	0	Return by level		
		1	Return by edge		
$K3_0$	Ports $P3_0$, $P3_1$ valid waveform/level selection bit (Note 2)	0	Falling waveform/"L" level		
		1	Rising waveform/"H" level		

Note 1. "R" represents read enabled, and "W" represents write enabled.
Note 2. To be invalid ($K2_2$ = "0") key-on wakeup of ports $P3_0$ and $P3_1$, set the registers $K3_0$ and $K3_1$ to "0."
Note 3. To be invalid ($K2_3$ = "0") key-on wakeup of ports $P3_2$ and $P3_3$, set the registers $K3_2$ and $K3_3$ to "0."

Table 26 Pull-up control register

Pull-up control register PU0		at reset : 0000$_2$	at power down : state retained	R/W TAPU0/TPU0A
PU0$_3$	Port P0$_3$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU0$_2$	Port P0$_2$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU0$_1$	Port P0$_1$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU0$_0$	Port P0$_0$ 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 power down : state retained	R/W TAPU1/TPU1A
PU1$_3$	Port P1$_3$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU1$_2$	Port P1$_2$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU1$_1$	Port P1$_1$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU1$_0$	Port P1$_0$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	

Pull-up control register PU2		at reset : 0000$_2$	at power down : state retained	R/W TAPU2/TPU2A
PU2$_3$	Port P2$_3$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU2$_2$	Port P2$_2$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU2$_1$	Port P2$_1$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU2$_0$	Port P2$_0$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	

Pull-up control register PU3		at reset : 0000$_2$	at power down : state retained	R/W TAPU3/TPU3A
PU3$_3$	Port P3$_3$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU3$_2$	Port P3$_2$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU3$_1$	Port P3$_1$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	
PU3$_0$	Port P3$_0$ pull-up transistor control bit	0	Pull-up transistor OFF	
		1	Pull-up transistor ON	

Note 1.“R” represents read enabled, and “W” represents write enabled.

Table 27 Interrupt control register

Interrupt control register I1		at reset : 0000_2		at power down : state retained	R/W TAI1/TI1A
$I1_3$	INT pin input control bit (Note 2)	0	INT pin input disabled		
		1	INT pin input enabled		
$I1_2$	Interrupt valid waveform for INT pin/ return level selection bit (Note 2)	0	Falling waveform (“L” level of INT pin is recognized with the SNZI0 instruction)/“L” level		
		1	Rising waveform (“H” level of INT pin is recognized with the SNZI0 instruction)/“H” level		
$I1_1$	INT pin edge detection circuit control bit	0	One-sided edge detected		
		1	Both edges detected		
$I1_0$	INT 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		

Note 1. “R” represents read enabled, and “W” represents write enabled.
Note 2. When the contents of $I1_2$ and $I1_3$ are changed, the external interrupt request flag EXF0 may be set.

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 59 shows the structure of the clock control circuit.
The 4559 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 or the RC oscillation can be used for the main clock (f(XIN)) of the 4559 Group.
The quartz-crystal oscillator can be used for sub-clock (f(XCIN)).

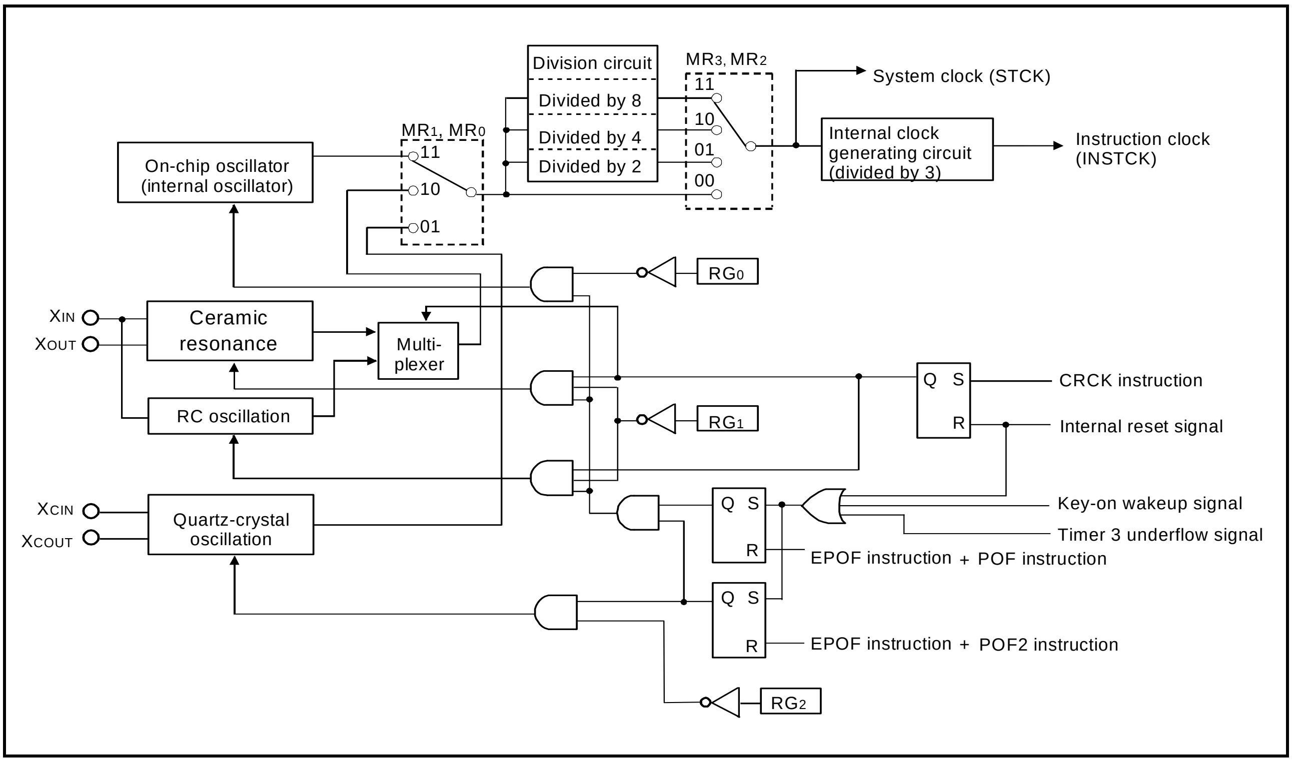


Fig 59. Clock control circuit structure

(1) On-chip oscillator operation

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

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

When the MCU operates by the ceramic resonator or the RC oscillator as the main clock (f(XIN)).
After system is released from reset, the ceramic oscillation is valid for main clock.
The ceramic oscillation is invalid and the RC oscillation circuit is valid with the CRCK instruction.
The CRCK instruction can be executed only once.
Execute the CRCK instruction in the initial setting routine (executing it in address 0 in page 0 is recommended).
When the main clock (f(XIN)) is not used, connect XIN pin to VSS and leave XOUT pin open, and do not execute the CRCK instruction (Figure 61).

(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.
A feedback resistor is built in between pins XIN and XOUT (Figure 62). Do not execute the CRCK instruction in program.

(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 63).
To select RC oscillation as the system clock, select the main clock (f(XIN) as the system clock by bits 0 and 1 of the clock control register MR.
The frequency is affected by a capacitor, a resistor and a microcomputer. So, set the constants within the range of the frequency limits.

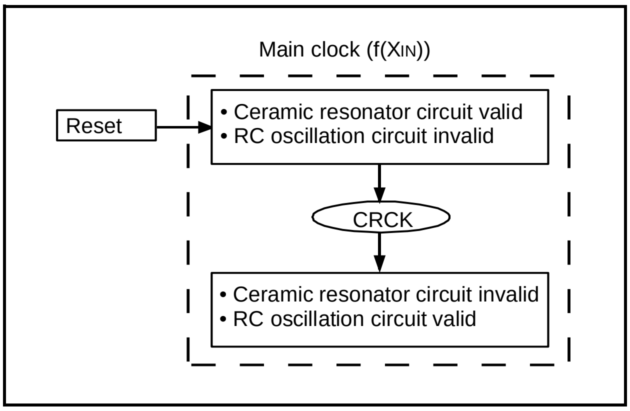


Fig 60. Switch to ceramic resonance/RC oscillation

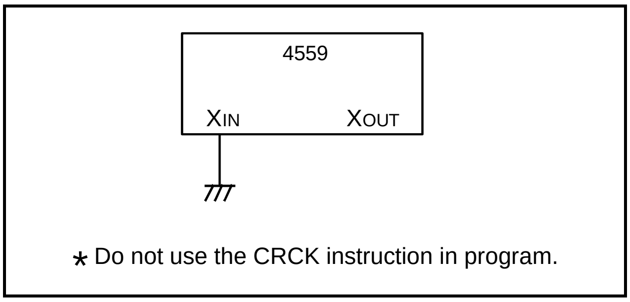


Fig 61. Handling of XIN and XOUT when operating on-chip oscillator

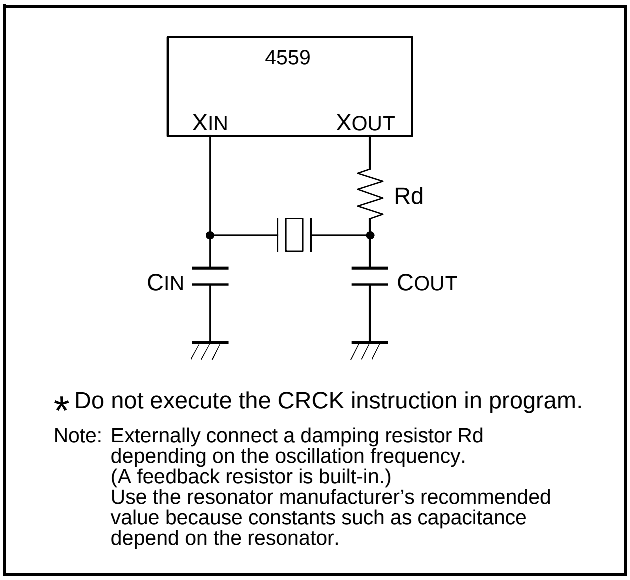


Fig 62. Ceramic resonator external circuit

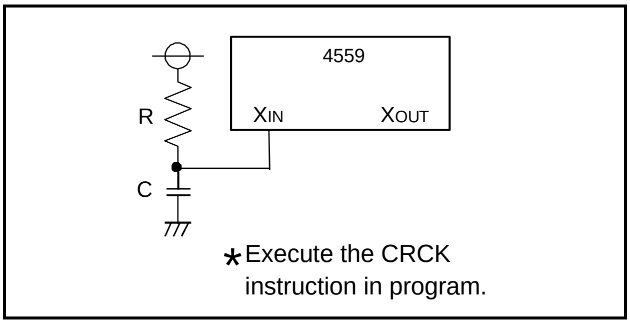


Fig 63. External RC oscillation circuit

(5) External clock

When the external clock signal is used as the main clock ($f(X_{IN})$), connect the X_{IN} pin to the clock source and leave X_{OUT} pin open (Figure 64). Do not execute the CRCK instruction.
Be careful that the maximum value of the oscillation frequency when using the external clock differs from the value when using the ceramic resonator (refer to the recommended operating condition). Also, note that the power down mode (POF and POF2 instructions) cannot be used when using the external clock.

(6) Sub-clock generating circuit $f(X_{CIN})$

Sub-clock signal $f(X_{CIN})$ is obtained by externally connecting a quartz-crystal oscillator. Connect this external circuit and a quartz-crystal oscillator to pins X_{CIN} and X_{COUT} at the shortest distance. A feedback resistor is built in between pins X_{CIN} and X_{COUT} (Figure 65). X_{CIN} pin and X_{COUT} pin are also used as ports D_6 and D_7, respectively. The sub-clock oscillation circuit is invalid and the function of ports D_6 and D_7 are valid by setting bit 2 of register RG to "1".
When sub-clock, ports D_6 and D_7 are not used, connect X_{CIN}/D_6 to V_{SS} and leave X_{COUT}/D_7 open.

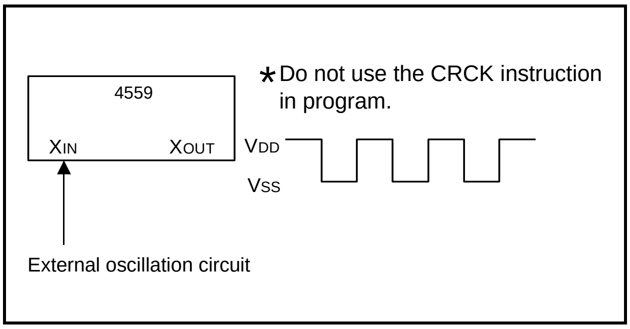


Fig 64. External clock input circuit

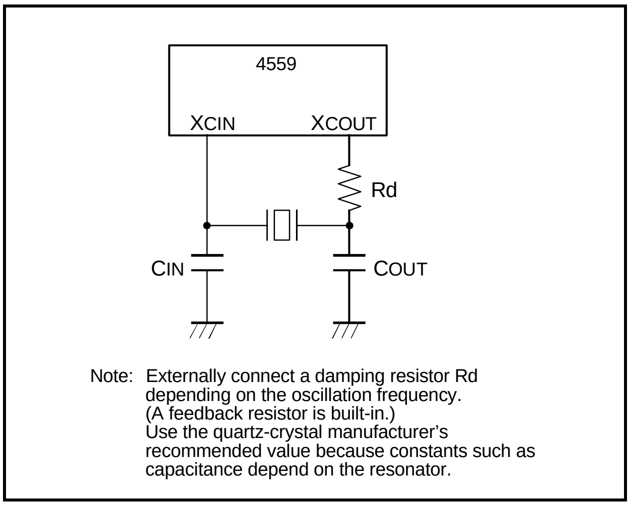


Fig 65. External quarts-crystal circuit

(7) Clock control register MR

Register MR controls system clock and operation mode (frequency division of 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 the start/stop of each oscillation circuit. Set the contents of this register through register A with the TRGA instruction.

Table 28 Clock control registers

Clock control register MR		at reset : 1100_2		at power down : state retained	R/W TAMR/TMRA
MR_3	Operation mode selection bits	MR_3	MR_2	Operation mode	
		0	0	Through mode	
		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	System clock selection bits (Note 2)	MR_1	MR_0	System clock	
		0	0	f(RING)	
		0	1	$f(X_{IN})$	
MR_0		1	0	$f(X_{CIN})$	
		1	1	Not available (Note 3)	

Clock control register RG		at reset : 000_2	at power down : state retained	W TRGA
RG_2	Sub-clock ($f(X_{CIN})$) control bit (Note 4)	0	Sub-clock ($f(X_{CIN})$) oscillation available, ports D_6 and D_7 not selected	
		1	Sub-clock ($f(X_{CIN})$) oscillation stop, ports D_6 and D_7 selected	
RG_1	Main-clock ($f(X_{IN})$) control bit (Note 4)	0	Main clock ($f(X_{IN})$) oscillation available	
		1	Main clock ($f(X_{IN})$) oscillation stop	
RG_0	On-chip oscillator (f(RING)) control bit (Note 4)	0	On-chip oscillator (f(RING)) oscillation available	
		1	On-chip oscillator (f(RING)) oscillation stop	

Note 1. R” represents read enabled, and “W” represents write enabled.
Note 2. The stopped clock cannot be selected for system clock.
Note 3. “11” cannot be set to the low-order 2 bits (MR_1, MR_0) of register MR.
Note 4. The oscillation circuit selected for system clock cannot be stopped.

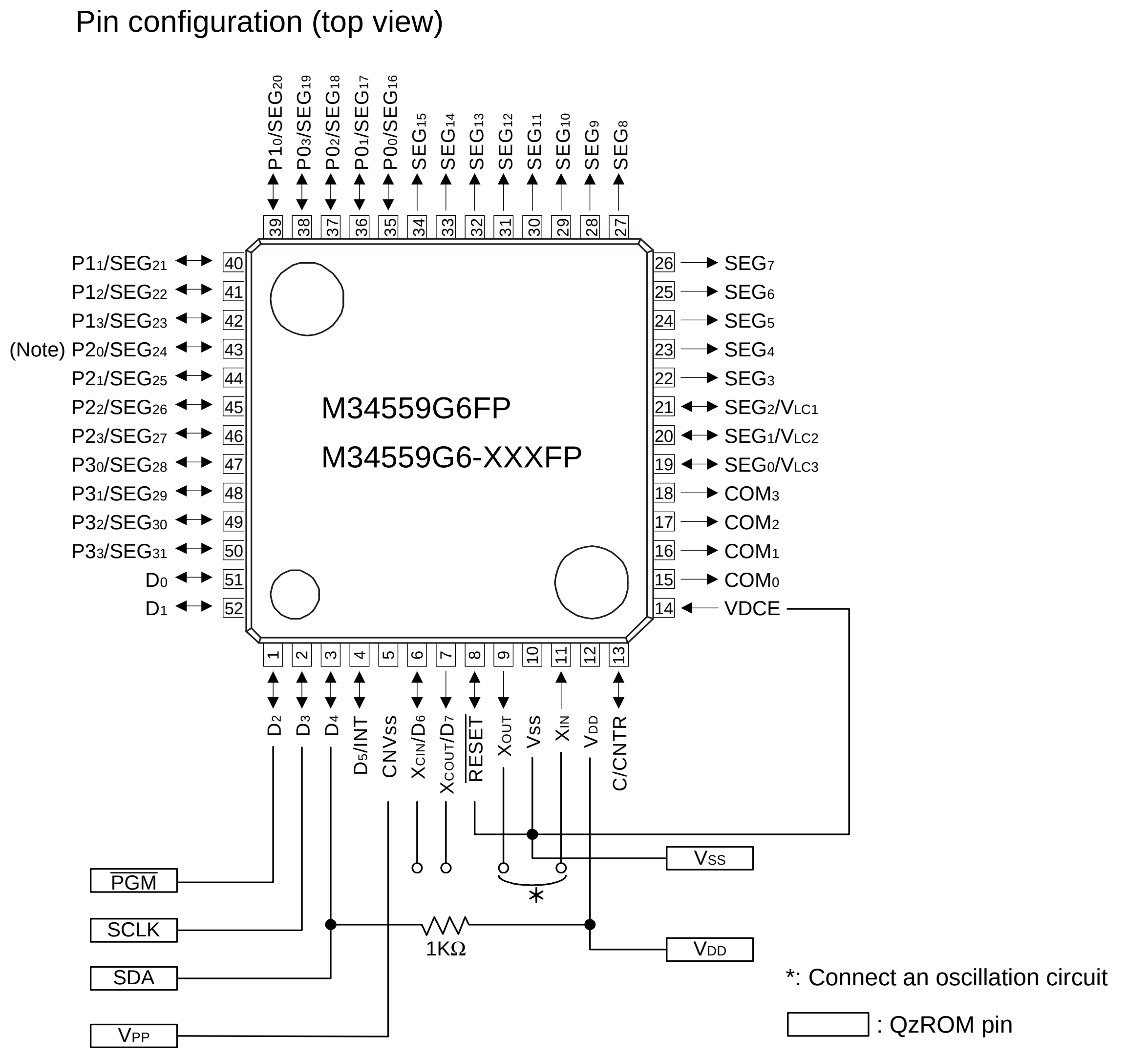
QzROM Writing Mode

In the QzROM writing mode, the user ROM area can be rewritten while the microcomputer is mounted on-board by using a serial pro-grammer which is applicable for this microcomputer. Table 29 lists the pin description (QzROM writing mode) and Figure 66 shows the pin connections.
Refer to Figure 67 for examples of a connection with a serial pro-grammer.
Contact the manufacturer of your serial programmer for serial pro-grammer. Refer to the user's manual of your serial programmer for details on how to use it.

Table 29 Pin description (QzROM writing mode)

Pin	Name	I/O	Function
V_{DD}, V_{SS}	Power source, GND		Apply 2.7 to 4.7V to V_{CC}, and 0V to V_{SS}.
$\overline{RESET}$	Reset input	input	Reset input pin for active "L". Reset occurs when RESET pin is hold at an "L" level for 16 cycles or more of X_{IN}.
X_{IN}, X_{CIN}	Clock input	input	Either connect an oscillator circuit or connect X_{IN} and X_{CIN} to V_{SS} and leave X_{OUT} and X_{COUT} open.
X_{OUT}, X_{COUT}	Clock output	output	
D_0 – D_5 $P0_0/SEG_{16}$ – $P0_3/SEG_{19}$ $P1_0/SEG_{20}$ – $P1_3/SEG_{23}$ $P2_0/SEG_{24}$ (Note 1) – $P2_3/SEG_{27}$ $P3_0/SEG_{28}$ – $P3_3/SEG_{31}$	I/O port	I/O	Input "H" or "L" level signal or leave the pin open.
CNV_{SS}	V_{PP} input	input	QzROM programmable power source pin.
D_4	SDA input/output	I/O	Serial data I/O pin.
D_3	SCLK input	input	Serial clock input pin.
D_2	$\overline{PGM}$ input	input	Read/program pulse input pin.
VDCE	Voltage drop detection circuit enable	input	Input "H" or "L" level signal
SEG_0/V_{LC3} – SEG_2/V_{LC1} SEG_3 – SEG_{15} COM_0 – COM_3	Segment output/ LCD power source/ Common output	output	Either connect to an LCD panel or leave open.
C/CNTR	Output port C/ Timer I/O	output	C/CNTR pin outputs "L" level.

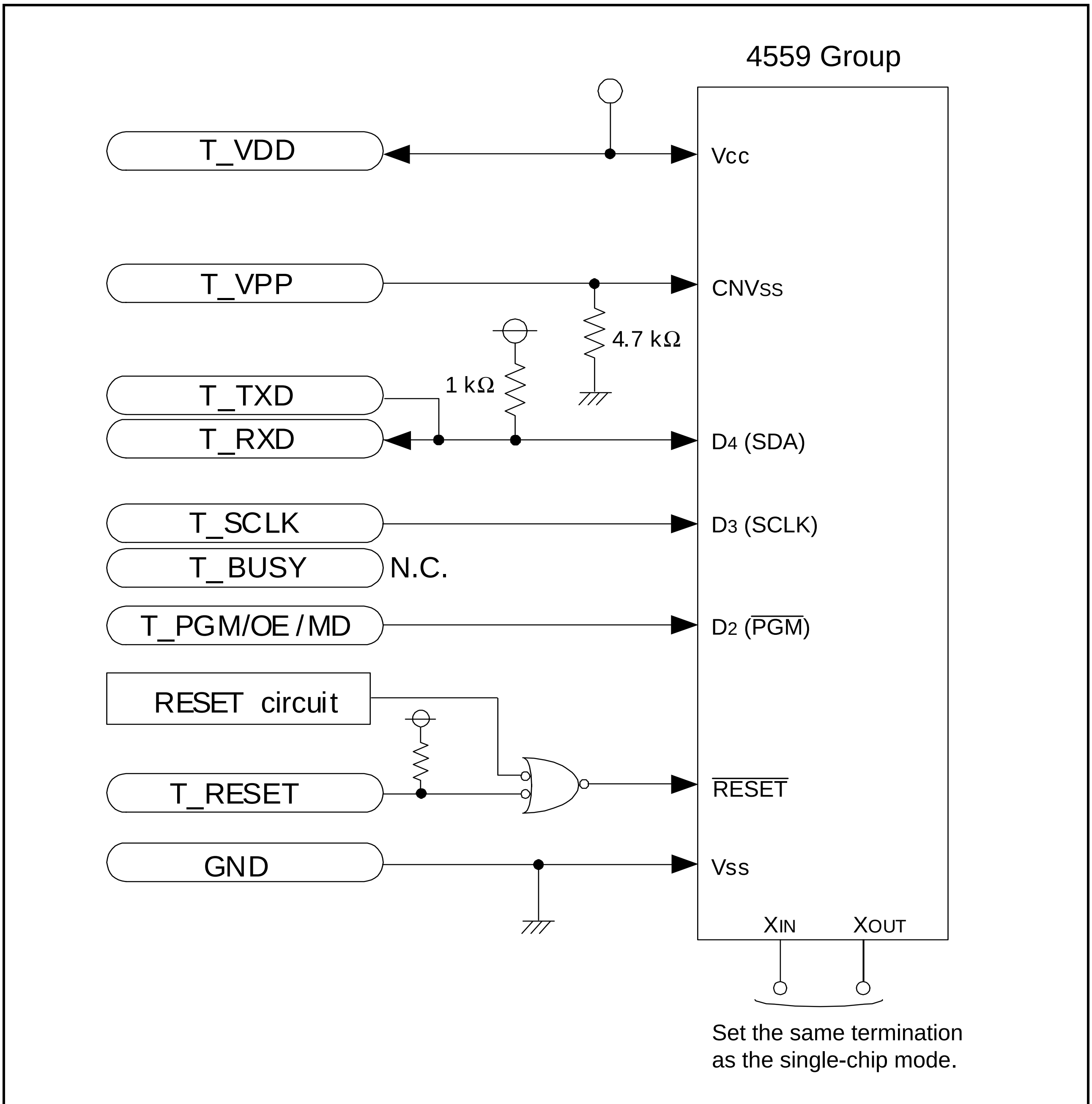
Note 1. Note that the $P2_0/SEG_{24}$ pin is pulled down internally by the MCU during the transition period (the period when V_{PP} is approximately 0.5 V_{DD} to 1.3 V_{DD}) when the programming power supply (V_{PP}) is applied to the CNV_{SS} pin. In addition, the $P2_0/SEG_{24}$ pin is high inpedance when V_{PP} is approximately 1.3 V_{DD} or grater.



OUTLINE PLQP0052JA-A (52P6A-A)

Note: Note that the $P2_0/SEG_{24}$ pin is pulled down internally by the MCU during the transition period (that period when V_{PP} is approximately 0.5 V_{DD} to 1.3 V_{DD}) when the programming power supply (V_{PP}) is applied to the CNV_{SS} pin. In addition, the $P2_0/SEG_{24}$ pin is high impedance when V_{PP} is approximately 1.3 V_{DD} or greater.

Fig 66. Pin connection diagram



Note: For the programming circuit, the wiring capacity of each signal pin must not exceed 47 pF.

Fig 67. When using programmer of Suisei Electronics System Co., LTD, connection example

LIST OF PRECAUTIONS

(1) Noise and latch-up prevention

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

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

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

(2) Note on Power Source Voltage

When the power source voltage value of a microcomputer is less than the value which is indicated as the recommended operating conditions, the microcomputer does not operate normally and may perform unstable operation.
In a system where the power source voltage drops slowly when the power source voltage drops or the power supply is turned off, reset a microcomputer when the supply voltage is less than the recommended operating conditions and design a system not to cause errors to the system by this unstable operation.

(3) 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)

(4) 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)

(5) Program counter

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

(6) 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.

(7) Multifunction

- The input/output of D_5 can be used even when INT is used. Be careful when using inputs of both INT and D_5 since the input threshold value of INT pin is different from that of port D_5.
- “H“ output function of port C can be used even when the CNTR (output) is used.

(8) Power-on reset

When the built-in power-on reset circuit is used, set the time for the supply voltage to rise from 0 V to the minimum voltage of recommended operating conditions 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.

(9) POF, POF2 instruction

When the POF or POF2 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 or POF2 instruction.
Be sure to disable interrupts by executing the DI instruction before executing the EPOF instruction and the POF/POF2 instruction continuously.

(10)D5/INT pin

(1) Bit 3 of register I1

When the input of the D5/INT 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 D5/INT 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 (1) in Figure 68.) 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 (2) in Figure 68.).
  Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to (3) in Figure 68.).

```
    :
LA 4        ; (×××02)
TV1A        ; The SNZ0 instruction is valid ...... (1)
LA 8        ; (1×××2)
TI1A        ; Control of INT pin input 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-1

(2) Bit 3 of register I1

When the bit 3 of register I1 is cleared to “0”, the power down mode is selected and the input of INT pin is disabled, be careful about the following notes.

- When the INT pin input is disabled (register $I1_3$ = “0”), set the key-on wakeup of INT pin to be invalid (register $K2_0$ = “0”) before system enters to the power down mode. (refer to (1) in Figure 69.).

```
    :
LA 0        ; (×××02)
TK2A        ; INT0 key-on wakeup disabled .....(1)
DI
EPOF
POF2        ; RAM back-up
    :

×: these bits are not used here.
```

Fig 69. External 0 interrupt program example-2

(3) Bit 2 of register I1

When the interrupt valid waveform of the D5/INT 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 D5/INT 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 (1) in Figure 70.) and then, change the bit 2 of register I1 is changed.
  In addition, execute the SNZ0 instruction to clear the EXF0 flag to “0” after executing at least one instruction (refer to (2) in Figure 70.).
  Also, set the NOP instruction for the case when a skip is performed with the SNZ0 instruction (refer to (3) in Figure 70.).

```
    :
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 70. External 0 interrupt program example-3

(11)Prescaler

Stop prescaler counting and then execute the TABPS instruction to read its data.
Stop prescaler counting and then execute the TPSAB instruction to write data to prescaler.

(12)Timer count source

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

(13)Reading the count value

Stop timer 1 or 2 counting and then execute the TAB1 or TAB2 instruction to read its data.

(14)Writing to the timer

Stop timer 1, 2 or LC counting and then execute the T1AB, T2AB, T2R2L or TLCA instruction to write data to timer.

(15)Writing to reload register

In order to write a data to the reload register R1 while the timer 1 is operating, execute the TR1AB instruction except a timing of the timer 1 underflow.
In order to write a data to the reload register R2H while the timer 2 is operating, execute the T3HAB instruction except a timing of the timer 2 underflow.

(16)PWM signal

If the timer 2 count stop timing and the timer 2 underflow timing overlap during output of the PWM signal, a hazard may occur in the PWM output waveform.
When "H" interval expansion function of the PWM signal is used, set "1" or more to reload register R2H.
Set the port C output latch to "0" to output the PWM signal from C/CNTR pin.

(17)Timer 3

Stop timer 3 counting to change its count source.
When operating timer 3 during clock operating mode, set 1 cycle or more of count source to the following period; from setting bit 2 of register W3 to "1" till executing the POF instruction.

(18)Prescaler, timer 1 count start timing and count time when operation starts

Count starts from the first rising edge of the count source (2) in Figure 71 after prescaler and timer operations start (1) in Figure 71.
Time to first underflow (3) in Figure 71 is shorter (for up to 1 period of the count source) than time among next underflow (4) in Figure 71 by the timing to start the timer and count source operations after count starts.
When selecting CNTR input as the count source of timer 1, timer 1 operates synchronizing with the count edge (falling edge or rising edge) of CNTR input selected by software.

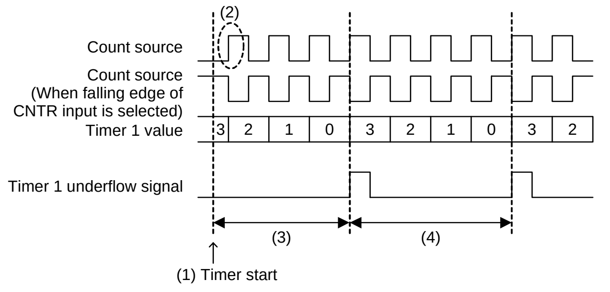


Fig 71. Timer count start timing and count time when operation starts (1)

(19)Timer 2, LC count start timing and count time when operation starts

Count starts from the first edge of the count source (2) in Figure 68 after timer 2 and LC operation start (1) in Figure 72.
Time to first underflow (3) in Figure 68 is different (for up to 1 period of the count source) from time among next underflow (4) in Figure 72 by the timing to start the timer and count source operations after count starts.

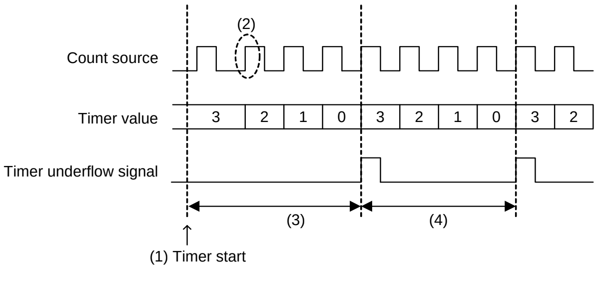


Fig 72. Timer count start timing and count time when operation starts (2)

(20)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 contents of WDF1 flag and timer WDT are initialized at the power down.
- When using the watchdog timer and the power down, initialize the WDF1 flag with the WRST instruction just before the microcomputer enters the power down mode.
  Also, set the NOP instruction after the WRST instruction, for the case when a skip is performed with the WRST instruction.

(21)Voltage drop detection circuit

The voltage drop detection circuit detection voltage of this product is set up lower than the minimum value of the supply voltage of the recommended operating conditions.
When the supply voltage of a microcomputer falls below to the minimum value of recommended operating conditions and regoes up (ex. battery exchange of an application product), depending on the capacity value of the bypass capacitor added to the power supply pin, the following case may cause program failure (Figure 73);
supply voltage does not fall below to V_{RST}^-, and its voltage re-goes up with no reset.
In such a case, please design a system which supply voltage is once reduced below to V_{RST}^- and re-goes up after that.

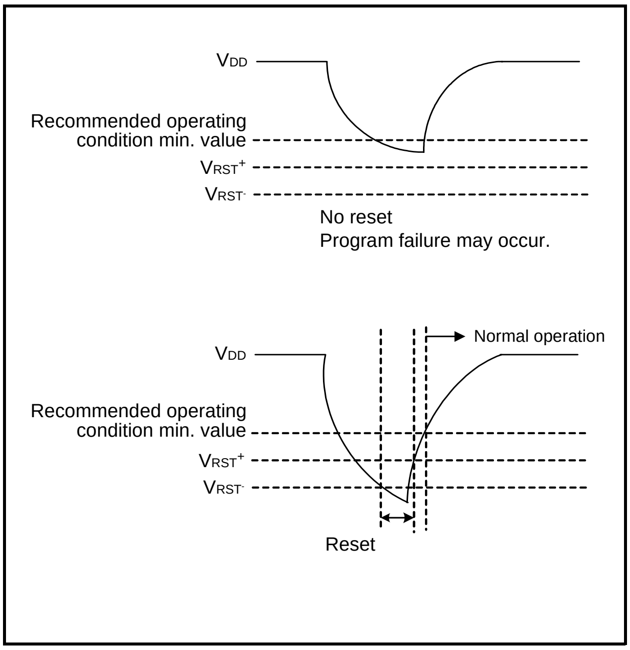


Fig 73. V_{DD} and V_{RST}^-

(22)On-chip oscillator

The clock frequency of the on-chip oscillator depends on the supply voltage and the operation temperature range.
Be careful that variable frequencies when designing application products.
Also, the oscillation stabilize wait time after system is released from reset is generated by the on-chip oscillator clock. When considering the oscillation stabilize wait time after system is released from reset, be careful that the variable frequency of the on-chip oscillator clock.

(23)RC oscillation

The CRCK instruction can be executed only once after reset release.
Execute the CRCK instruction in the initial setting routine (executing it in address 0 in page 0 is recommended).
The frequency is affected by a capacitor, a resistor and a microcomputer.
So, set the constants within the range of the frequency limits.

(24)External clock

Be careful that the maximum value of the oscillation frequency when using the external clock differs from the value when using the ceramic resonator (refer to the recommended operating condition).
Also, note that the power-down mode (POF or POF2 instruction) cannot be used when using the external clock.

(25)QzROM

(1) Be careful not to apply overvoltage to MCU. The contents of QzROM may be overwritten because of overvoltage. Take care especially at turning on the power.
(2) As for the product shipped in blank, Renesas does not perform the writing test to user ROM area after the assembly process though the QzROM writing test is performed enough before the assembly process. Therefore, a writing error of approx. 0.1 % may occur. Moreover, please note the contact of cables and foreign bodies on a socket, etc. because a writing environment may cause some writing errors.

(26)Notes On ROM Code Protect (QzROM product shipped after writing)

As for the QzROM product shipped after writing, the ROM code protect is specified according to the ROM option setup data in the mask file which is submitted at ordering.
The ROM option setup data in the mask file is "00_{16}" for protect enabled or "FF_{16}" for protect disabled.
Note that the mask file which has nothing at the ROM option data or has the data other than "00_{16}" and "FF_{16}" can not be accepted.

(27)Data Required for QzROM Writing Orders

The following are necessary when ordering a QzROM product shipped after writing:
1. QzROM Writing Confirmation Form*
2. Mark Specification Form*
3. ROM data...........Mask file
* For the QzROM writing confirmation form and the mark specification form, refer to the "Renesas Technology Corp." Homepage (http://www.renesas.com/homepage.jsp).
Note that we cannot deal with special font marking (customer's trademark etc.) in QzROM microcomputer.

NOTES ON NOISE

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

(1) Shortest wiring length

The wiring on a printed circuit board can function as an antenna which feeds noise into the microcomputer.
The shorter the total wiring length (by mm unit), the less the possibility of noise insertion into a microcomputer.

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

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

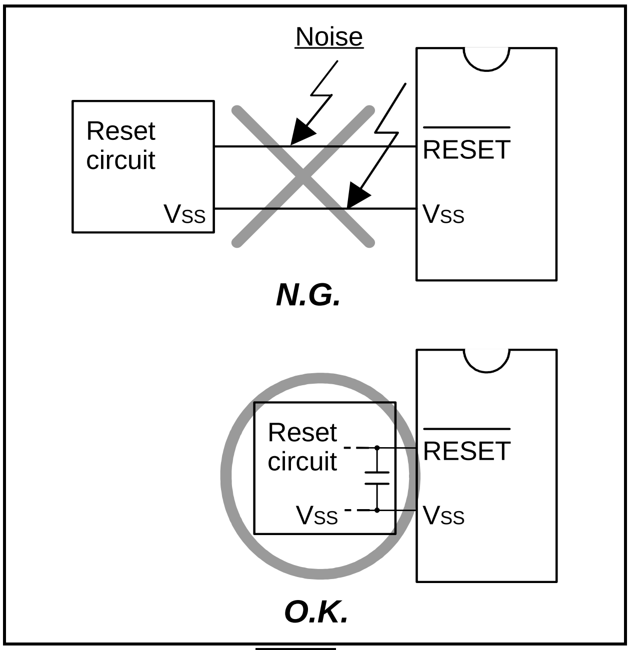


Fig 74. Wiring for the $\overline{\text{RESET}}$ input pin

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

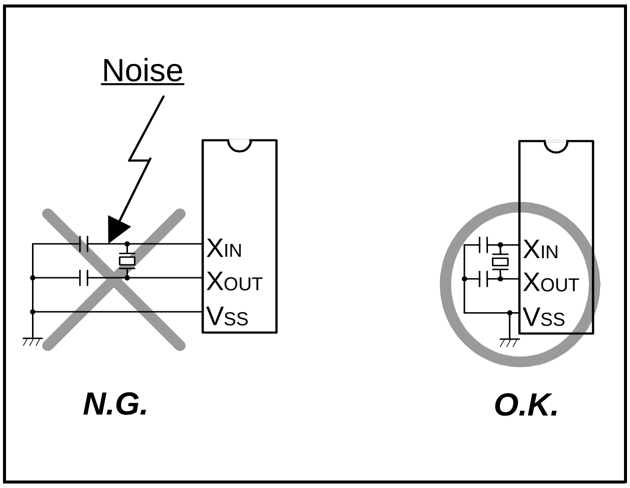


Fig 75. Wiring for clock I/O pins

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

(3) Wiring to CNVSS pin
Connect an approximately 5 kΩ resistor to the VPP pin and also to the GND pattern supplied to the VSS pin with shortest possible wiring.

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

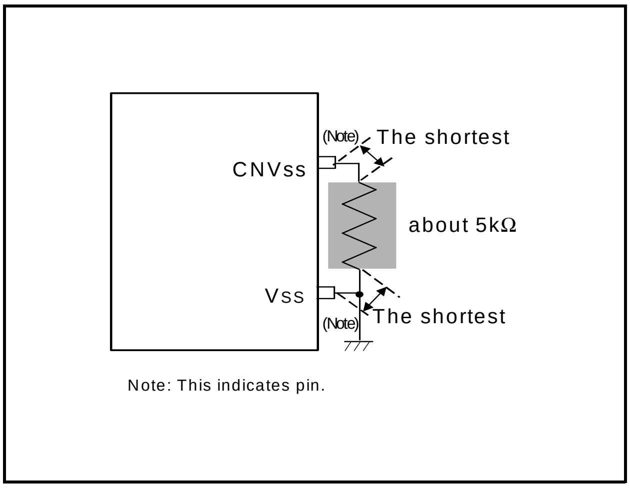


Fig 76. Wiring for CNVSS pin

(2) Connection of bypass capacitor across Vss line and VDD line

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

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

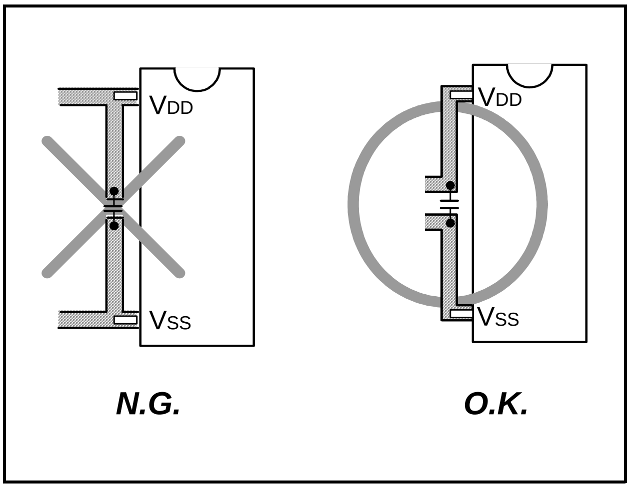


Fig 77. Bypass capacitor across the Vss line and the VDD line

(3) Oscillator concerns

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

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

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

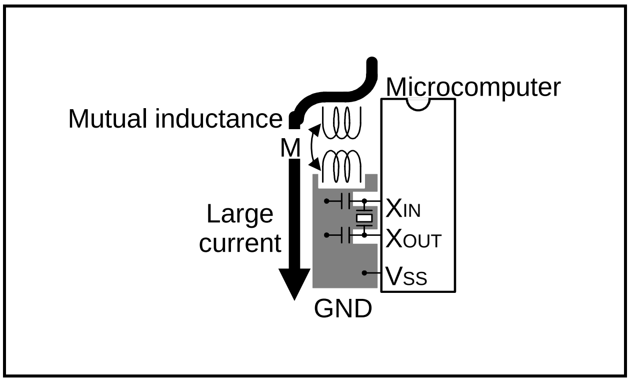


Fig 78. Wiring for a large current signal line

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

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

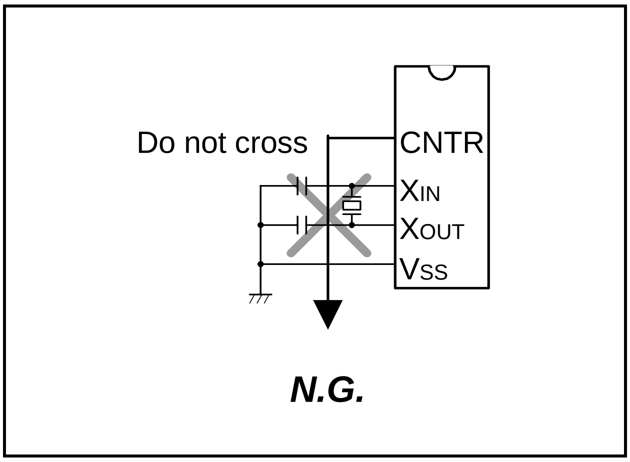


Fig 79. Wiring to a signal line where potential levels change frequently

(3) Oscillator protection using VSS pattern
As for a two-sided printed circuit board, print a VSS pattern on the underside (soldering side) of the position (on the component side) where an oscillator is mounted.
Connect the VSS pattern to the microcomputer VSS pin with the shortest possible wiring.
Besides, separate this VSS pattern from other VSS patterns.

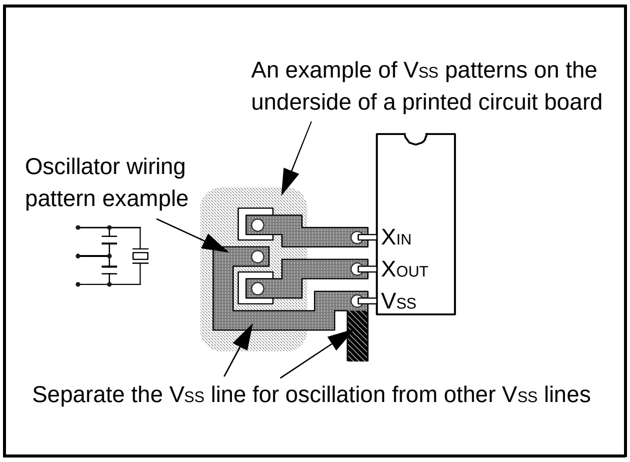


Fig 80. Vss pattern on the underside of an oscillator

(4) Setup for I/O ports
Setup I/O ports using hardware and software as follows:

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

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

(5) Providing of watchdog timer function by software
If a microcomputer runs away because of noise or others, it can be detected by a software watchdog timer and the microcomputer can be reset to normal operation. This is equal to or more effective than program runaway detection by a hardware watchdog timer. The following shows an example of a watchdog timer provided by software.
In the following example, to reset a microcomputer to normal operation, the main routine detects errors of the interrupt processing routine and the interrupt processing routine detects errors of the main routine.
This example assumes that interrupt processing is repeated multiple times in a single main routine processing.

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

  $N + 1 \geq$

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

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

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

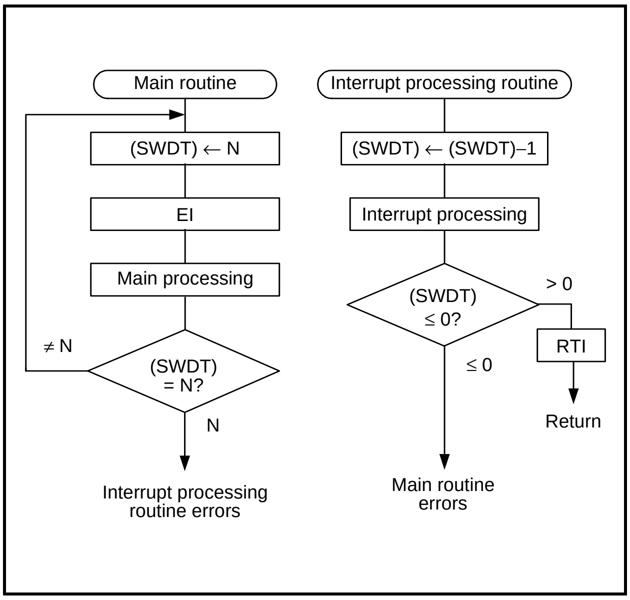


Fig 81. Watchdog timer by software

CONTROL REGISTERS

Interrupt control register V1		at reset : 0000_2		at power down : 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)		
$V1_1$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$V1_0$	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 power down : 0000_2	R/W TAV2/TV2A
$V2_3$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$V2_2$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$V2_1$	Not used	0	This bit has no function, but read/write is enabled.		
		1			
$V2_0$	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 power down : state retained	R/W TAI1/TI1A
$I1_3$	INT pin input control bit (Note 2)	0	INT pin input disabled		
		1	INT pin input enabled		
$I1_2$	Interrupt valid waveform for INT pin/ return level selection bit (Note 2)	0	Falling waveform (“L” level of INT pin is recognized with the SNZI0 instruction)/“L” level		
		1	Rising waveform (“H” level of INT pin is recognized with the SNZI0 instruction)/“H” level		
$I1_1$	INT pin edge detection circuit control bit	0	One-sided edge detected		
		1	Both edges detected		
$I1_0$	INT 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		

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

Clock control register MR		at reset : 1100_2		at power down : state retained	R/W TAMR/TMRA
MR_3	Operation mode selection bits	MR_3	MR_2	Operation mode	
		0	0	Through mode	
MR_2		0	1	Frequency divided by 2 mode	
		1	0	Frequency divided by 4 mode	
		1	1	Frequency divided by 8 mode	
MR_1	System clock selection bits (Note 2)	MR_1	MR_0	System clock	
		0	0	f(RING)	
MR_0		0	1	$f(X_{IN})$	
		1	0	$f(X_{CIN})$	
		1	1	Not available (Note 3)	

Clock control register RG		at reset : 000_2		at power down : state retained	W TRGA
RG_2	Sub-clock ($f(X_{CIN})$) control bit (Note 4)	0	Sub-clock ($f(X_{CIN})$) oscillation available, ports D_6 and D_7 not selected		
		1	Sub-clock ($f(X_{CIN})$) oscillation stop, ports D_6 and D_7 selected		
RG_1	Main-clock ($f(X_{IN})$) control bit (Note 4)	0	Main clock ($f(X_{IN})$) oscillation available		
		1	Main clock ($f(X_{IN})$) oscillation stop		
RG_0	On-chip oscillator (f(RING)) control bit (Note 4)	0	On-chip oscillator (f(RING)) oscillation available		
		1	On-chip oscillator (f(RING)) oscillation stop		

Note 1. R" represents read enabled, and "W" represents write enabled.
Note 2. The stopped clock cannot be selected for system clock.
Note 3. "11" cannot be set to the low-order 2 bits (MR_1, MR_0) of register MR.
Note 4. The oscillation circuit selected for system clock cannot be stopped.

Timer control register PA		at reset : 0_2	at power down : 0_2	W TAPP
PA_0	Prescaler control bit	0	Stop (state retained)	
		1	Operating	

Timer control register W1		at reset : 0000_2		at power down : 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 (Note 3)	$W1_1$	$W1_0$	Count source	
		0	0	PWM signal (PWMOUT)	
		0	1	Prescaler output (ORCLK)	
$W1_0$		1	0	Timer 3 underflow signal (T3UDF)	
		1	1	CNTR input	

Timer control register W2		at reset : 0000_2	at power down : 0000_2	R/W TAW2/TW2A
$W2_3$	CNTR pin function control bit	0	CNTR pin output invalid	
		1	CNTR pin output valid	
$W2_2$	PWM signal “H” interval expansion function control bit	0	PWM signal “H” interval expansion function invalid	
		1	PWM signal “H” interval expansion function valid	
$W2_1$	Timer 2 control bit	0	Stop (state retained)	
		1	Operating	
$W2_0$	Timer 2 count source selection bit	0	X_{IN} input	
		1	Prescaler output (ORCLK)/2	

Timer control register W3		at reset : 0000_2		at power down : state retained	R/W TAW3/TW3A
$W3_3$	Timer 3 count source selection bit	0	X_{IN} input		
		1	Prescaler output (ORCLK)		
$W3_2$	Timer 3 control bit	0	Stop (initial state)		
		1	Operating		
$W3_1$	Timer 3 count value selection bits	$W3_1$	$W3_0$	Count source	
		0	0	Underflow every 8192 count	
		0	1	Underflow every 16384 count	
$W3_0$		1	0	Underflow every 32768 count	
		1	1	Underflow every 65536 count	

Timer control register W4		at reset : 0000_2	at power down : state retained	R/W TAW4/TW4A
$W4_3$	Timer LC control bit	0	Stop (state retained)	
		1	Operating	
$W4_2$	Timer LC count source selection bit	0	Bit 4 ($T3_4$) of timer 3	
		1	System clock (STCK)	
$W4_1$	CNTR pin output auto-control circuit selection bit	0	CNTR output auto-control circuit not selected	
		1	CNTR output auto-control circuit selected	
$W4_0$	CNTR pin input count edge selection bit	0	Falling edge	
		1	Rising edge	

Note 1. “R” represents read enabled, and “W” represents write enabled.
Note 2. This function is valid only when the timer 1 count start synchronous circuit is selected ($I1_0$ =“1”).
Note 3. Port C output is invalid when CNTR input is selected for the timer 1 count source.

LCD control register L1		at reset : 0000_2		at power down : state retained	R/W TAL1/TL1A
$L1_3$	Internal dividing resistor for LCD power supply selection bit (Note 2)	0	2r × 3, 2r × 2		
		1	r × 3, r × 2		
$L1_2$	LCD control bit	0	Stop (OFF)		
		1	Operating		
$L1_1$	LCD duty and bias selection bits	$L1_1$	L1	Duty	Bias
		0	0	Not available	Not available
$L1_0$		0	1	1/2	1/2
		1	0	1/3	1/3
		1	1	1/4	1/3

LCD control register L2		at reset : 0000_2	at power down : state retained	W TL2A
$L2_3$	SEG_0/V_{LC3} pin function switch bit (Note 3)	0	SEG_0	
		1	V_{LC3}	
$L2_2$	SEG_1/V_{LC2} pin function switch bit (Note 4)	0	SEG_1	
		1	V_{LC2}	
$L2_1$	SEG_2/V_{LC1} pin function switch bit (Note 4)	0	SEG_2	
		1	V_{LC1}	
$L2_0$	Internal dividing resistor for LCD power supply control bit	0	Internal dividing resistor valid	
		1	Internal dividing resistor invalid	

LCD control register L3		at reset : 1111_2	at power down : state retained	W TL3A
$L3_3$	$P2_3/SEG_{27}$ pin function switch bit	0	SEG_{27}	
		1	$P2_3$	
$L3_2$	$P2_2/SEG_{26}$ pin function switch bit	0	SEG_{26}	
		1	$P2_2$	
$L3_1$	$P2_1/SEG_{25}$ pin function switch bit	0	SEG_{25}	
		1	$P2_1$	
$L3_0$	$P2_0/SEG_{24}$ pin function switch bit	0	SEG_{24}	
		1	$P2_0$	

Note 1. "R" represents read enabled, and "W" represents write enabled.
Note 2. "r (resistor) multiplied by 3" is used at 1/3 bias, and "r multiplied by 2" is used at 1/2 bias.
Note 3. V_{LC3} is connected to V_{DD} internally when SEG_0 pin is selected.
Note 4. Use internal dividing resistor when SEG_1 and SEG_2 pins are selected.

LCD control register C1		at reset : 1111_2	at power down : state retained	W TC1A
$C1_3$	$P0_3/SEG_{19}$ pin function switch bit	0	SEG_{19}	
		1	$P0_3$	
$C1_2$	$P0_2/SEG_{18}$ pin function switch bit	0	SEG_{18}	
		1	$P0_2$	
$C1_1$	$P0_1/SEG_{17}$ pin function switch bit	0	SEG_{17}	
		1	$P0_1$	
$C1_0$	$P0_0/SEG_{16}$ pin function switch bit	0	SEG_{16}	
		1	$P0_0$	

LCD control register C2		at reset : 1111_2	at power down : state retained	W TC2A
$C2_3$	$P1_3/SEG_{23}$ pin function switch bit	0	SEG_{23}	
		1	$P1_3$	
$C2_2$	$P1_2/SEG_{22}$ pin function switch bit	0	SEG_{22}	
		1	$P1_2$	
$C2_1$	$P1_1/SEG_{21}$ pin function switch bit	0	SEG_{21}	
		1	$P1_1$	
$C2_0$	$P1_0/SEG_{20}$ pin function switch bit	0	SEG_{20}	
		1	$P1_0$	

LCD control register C3		at reset : 1111_2	at power down : state retained	W TC3A
$C3_3$	$P3_3/SEG_{31}$ pin function switch bit	0	SEG_{31}	
		1	$P3_3$	
$C3_2$	$P3_2/SEG_{30}$ pin function switch bit	0	SEG_{30}	
		1	$P3_2$	
$C3_1$	$P3_1/SEG_{29}$ pin function switch bit	0	SEG_{29}	
		1	$P3_1$	
$C3_0$	$P3_0/SEG_{28}$ pin function switch bit	0	SEG_{28}	
		1	$P3_0$	

Note 1.“R” represents read enabled, and “W” represents write enabled. .

Key-on wakeup control register K0		at reset : 0000_2		at power down : state retained	R/W TAK0/TK0A
$K0_3$	Ports $P1_2$, $P1_3$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_2$	Ports $P1_0$, $P1_1$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_1$	Ports $P0_2$, $P0_3$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K0_0$	Ports $P0_0$, $P0_1$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K1		at reset : 0000_2		at power down : state retained	R/W TAK1/TK1A
$K1_3$	Port $P2_3$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K1_2$	Port $P2_2$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K1_1$	Port $P2_1$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K1_0$	Port $P2_0$ key-on wakeup control bit	0	Key-on wakeup not used		
		1	Key-on wakeup used		

Key-on wakeup control register K2		at reset : 0000_2		at power down : state retained	R/W TAK2/TK2A
$K2_3$	Ports $P3_2$, $P3_3$ key-on wakeup control bit (Note 3)	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K2_2$	Ports $P3_0$, $P3_1$ key-on wakeup control bit (Note 2)	0	Key-on wakeup not used		
		1	Key-on wakeup used		
$K2_1$	INT pin return condition selection bit	0	Return by level		
		1	Return by edge		
$K2_0$	INT pin key-on wakeup control bit	0	Key-on wakeup invalid		
		1	Key-on wakeup valid		

Key-on wakeup control register K3		at reset : 0000_2		at power down : state retained	R/W TAK3/TK3A
$K3_3$	Ports $P3_2$, $P3_3$ return condition selection bit (Note 3)	0	Return by level		
		1	Return by edge		
$K3_2$	Ports $P3_2$, $P3_3$ valid waveform/level selection bit (Note 3)	0	Falling waveform/"L" level		
		1	Rising waveform/"H" level		
$K3_1$	Ports $P3_0$, $P3_1$ return condition selection bit (Note 2)	0	Return by level		
		1	Return by edge		
$K3_0$	Ports $P3_0$, $P3_1$ valid waveform/level selection bit (Note 2)	0	Falling waveform/"L" level		
		1	Rising waveform/"H" level		

Note 1. "R" represents read enabled, and "W" represents write enabled.
Note 2. To be invalid ($K2_2$ = "0") key-on wakeup of ports $P3_0$ and $P3_1$, set the registers $K3_0$ and $K3_1$ to "0."
Note 3. To be invalid ($K2_3$ = "0") key-on wakeup of ports $P3_2$ and $P3_3$, set the registers $K3_2$ and $K3_3$ to "0."

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

Pull-up control register PU2		at reset : 0000_2		at power down : state retained	R/W TAPU2/TPU2A
$PU2_3$	Port $P2_3$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU2_2$	Port $P2_2$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU2_1$	Port $P2_1$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU2_0$	Port $P2_0$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		

Pull-up control register PU3		at reset : 0000_2		at power down : state retained	R/W TAPU3/TPU3A
$PU3_3$	Port $P3_3$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU3_2$	Port $P3_2$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU3_1$	Port $P3_1$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		
$PU3_0$	Port $P3_0$ pull-up transistor control bit	0	Pull-up transistor OFF		
		1	Pull-up transistor ON		

Note 1. “R” represents read enabled, and “W” represents write enabled.

Port output structure control register FR0		at reset : 0000_2		at power down : 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 power down : state retained	W TFR1A
$FR1_3$	Ports D_3 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR1_2$	Ports D_2 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR1_1$	Ports D_1 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR1_0$	Ports 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 power down : state retained	W TFR2A
$FR2_3$	Ports $P3_2$, $P3_3$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR2_2$	Ports $P3_0$, $P3_1$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR2_1$	Ports D_5 output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR2_0$	Ports 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 power down : state retained	W TFR3A
$FR3_3$	Ports $P2_3$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR3_2$	Ports $P2_2$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR3_1$	Ports $P2_1$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		
$FR3_0$	Ports $P2_0$ output structure selection bit	0	N-channel open-drain output		
		1	CMOS output		

Note 1. “W” represents write enabled.

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

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

SYMBOL

Symbol	Contents	Symbol	Contents
A	Register A (4 bits)	R2H	Timer 2 reload register (8 bits)
B	Register B (4 bits)	RLC	Timer LC reload register (4 bits)
DR	Register DR (3 bits)	PS	Prescaler
E	Register E (8 bits)	T1	Timer 1
V1	Interrupt control register V1 (4 bits)	T2	Timer 2
V2	Interrupt control register V2 (4 bits)	TLC	Timer LC
I1	Interrupt control register I1 (4 bits)	T1F	Timer 1 interrupt request flag
PA	Timer control register PA (1 bit)	T2F	Timer 2 interrupt request flag
W1	Timer control register W1 (4 bits)	T3F	Timer 3 interrupt request flag
W2	Timer control register W2 (4 bits)	WDF1	Watchdog timer flag
W3	Timer control register W3 (4 bits)	WEF	Watchdog timer enable flag
W4	Timer control register W4 (4 bits)	INTE	Interrupt enable flag
MR	Clock control register MR (4 bits)	EXF0	External 0 interrupt request flag
RG	Clock control register RG (3 bits)	VDF	Voltage drop detection circuit flag
L1	LCD control register L1 (4 bits)	P	Power down flag
L2	LCD control register L2 (4 bits)	D	Port D (8 bits)
L3	LCD control register L3 (4 bits)	P0	Port P0 (4 bits)
C1	LCD control register C1 (4 bits)	P1	Port P1 (4 bits)
C2	LCD control register C2 (4 bits)	P2	Port P2 (4 bits)
C3	LCD control register C3 (4 bits)	P3	Port P3 (4 bits)
K0	Key-on wakeup control register K0 (4 bits)	C	Port C (1 bit)
K1	Key-on wakeup control register K1 (4 bits)	INT	INT pin (1 bit)
K2	Key-on wakeup control register K2 (4 bits)		
K3	Key-on wakeup control register K3 (4 bits)	x	Hexadecimal variable
PU0	Pull-up control register PU0 (4 bits)	y	Hexadecimal variable
PU1	Pull-up control register PU1 (4 bits)	z	Hexadecimal variable
PU2	Pull-up control register PU2 (4 bits)	p	Hexadecimal variable
PU3	Pull-up control register PU3 (4 bits)	n	Hexadecimal constant
FR0	Port output structure control register FR0 (4 bits)	i	Hexadecimal constant
FR1	Port output structure control register FR1 (4 bits)	j	Hexadecimal constant
FR2	Port output structure control register FR2 (4 bits)	$A_3 A_2 A_1 A_0$	Binary notation of hexadecimal variable A (same for others)
FR3	Port output structure control register FR3 (4 bits)		
X	Register X (4 bits)	←	Direction of data movement
Y	Register Y (4 bits)	( )	Contents of registers and memories
Z	Register Z (2 bits)	–	Negate, Flag unchanged after executing instruction
DP	Data pointer (10 bits) (It consists of registers X, Y, and Z)	M (DP)	RAM address pointed by the data pointer
		a	Label indicating address $a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$
PC	Program counter (14 bits)	p, a	Label indicating address $a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$ in page $p_6\ p_5\ p_4\ p_3\ p_2\ p_1\ p_0$
PC_H	High-order 7 bits of program counter		
PC_L	Low-order 7 bits of program counter		
SK	Stack register (14 bits × 8)	C+x	Hex. C + Hex. number x (also same for others)
SP	Stack pointer (3 bits)	?	Decision of state shown before “?”
CY	Carry flag	← →	Data exchange between a register and memory
UPTF	High-order bit reference enable flag		
RPS	Prescaler reload register (8 bits)		
R1	Timer 1 reload register (8 bits)		
R2L	Timer 2 reload register (8 bits)		

Note 1. The 4559 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	Page
Register to register transfer	TAB	(A) ← (B)	103 122
	TBA	(B) ← (A)	110 122
	TAY	(A) ← (Y)	110 122
	TYA	(Y) ← (A)	119 122
	TEAB	(E_7–E_4) ← (B) (E_3–E_0) ← (A)	112 122
	TABE	(B) ← (E_7–E_4) (A) ← (E_3–E_0)	104 122
	TDA	(DR_2–DR_0) ← (A_2–A_0)	111 122
	TAD	(A_2–A_0) ← (DR_2–DR_0) (A_3) ← 0	105 122
	TAZ	(A_1, A_0) ← (Z_1, Z_0) (A_3, A_2) ← 0	110 122
	TAX	(A) ← (X)	110 122
	TASP	(A_2–A_0) ← (SP_2–SP_0) (A_3) ← 0	108 122
RAM addresses	LXY x, y	(X) ← x, x = 0 to 15 (Y) ← y, y = 0 to 15	93 122
	LZ z	(Z) ← z, z = 0 to 3	93 122
	INY	(Y) ← (Y) + 1	92 122
	DEY	(Y) ← (Y) – 1	90 122
RAM to register transfer	TAM j	(A) ← (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15	106 122
	XAM j	(A) ←→ (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15	120 122
	XAMD j	(A) ←→ (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) – 1	120 122
	XAMI j	(A) ←→ (M(DP)) (X) ← (X)EXOR(j) j = 0 to 15 (Y) ← (Y) + 1	120 122
	TMA j	(M(DP)) ← (A) (X) ← (X)EXOR(j) j = 0 to 15	115 122

p=0 to 47

Grouping	Mnemonic	Function	Page
Arithmetic operation	LA n	(A) ← n n = 0 to 15	92 124
	TABP p	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p (PC_L) ← (DR_2–DR_0, A_3–A_0) (UPTF) = 1, (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	104 124
	AM	(A) ← (A) + (M(DP))	87 124
	AMC	(A) ← (A) + (M(DP)) + (CY) (CY) ← Carry	87 124
	A n	(A) ← (A) + n n = 0 to 15	87 124
	AND	(A) ← (A)AND(M(DP))	87 124
	OR	(A) ← (A)OR(M(DP))	94 124
	SC	(CY) ← 1	98 124
	RC	(CY) ← 0	96 124
	SZC	(CY) = 0 ?	102 124
	CMA	(A) ← $\overline{(A)}$	89 124
	RAR	CY → $A_3A_2A_1A_0$	95 124
Bit operation	SB j	(M_j(DP)) ← 1 j = 0 to 3	97 124
	RB j	(M_j(DP)) ← 0 j = 0 to 3	95 124
	SZB j	(M_j(DP)) = 0 ? j = 0 to 3	101 124
Comparison operation	SEAM	(A) = (M(DP)) ?	99 126
	SEA n	(A) = n ? n = 0 to 15	98 126
Branch operation	B a	(PC_L) ← a_6–a_0	88 126
	BL p, a	(PC_H) ← p (PC_L) ← a_6–a_0	88 126
	BLA p	(PC_H) ← p (PC_L) ← (DR_2–DR_0, A_3–A_0)	88 126

INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function	Page
Subroutine operation	BM a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC$_H$) ← 2 (PC$_L$) ← a$_6$–a$_0$	88 126
	BML p, a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC$_H$) ← p (PC$_L$) ← a$_6$–a$_0$	89 126
	BMLA p	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC$_H$) ← p (PC$_L$) ← (DR$_2$–DR$_0$, A$_3$–A$_0$)	89 126
Return operation	RTI	(PC) ← (SK(SP)) (SP) ← (SP) – 1	97 126
	RT	(PC) ← (SK(SP)) (SP) ← (SP) – 1	96 126
	RTS	(PC) ← (SK(SP)) (SP) ← (SP) – 1	97 126
Interrupt operation	DI	(INTE) ← 0	90 128
	EI	(INTE) ← 1	91 128
	SNZ0	V1$_0$ = 0 : (EXF0) = 1 ? (EXF0) ← 0 V1$_0$ = 1 : SNZ0 = NOP	99 128
	SNZI0	I1$_2$ = 0 : (INT) = “L” ? I1$_2$ = 1 : (INT) = “H” ?	99 128
	TAV1	(A) ← (V1)	108 128
	TV1A	(V1) ← (A)	118 128
	TAV2	(A) ← (V2)	108 128
	TV2A	(V2) ← (A)	118 128
	TAI1	(A) ← (I1)	105 128
	TI1A	(I1) ← (A)	113 128

Grouping	Mnemonic	Function	Page
Timer operation	TPAA	(PA) ← (A)	116 128
	TAW1	(A) ← (W1)	109 128
	TW1A	(W1) ← (A)	118 128
	TAW2	(A) ← (W2)	109 128
	TW2A	(W2) ← (A)	118 128
	TAW3	(A) ← (W3)	109 128
	TW3A	(W3) ← (A)	119 128
	TAW4	(A) ← (W4)	109 128
	TW4A	(W4) ← (A)	119 128
	TABPS	(B) ← (TPS$_7$–TPS$_4$) (A) ← (TPS$_3$–TPS$_0$)	104 130
	TPSAB	(RPS$_7$–RPS$_4$) ← (B) (TPS$_7$–TPS$_4$) ← (B) (RPS$_3$–RPS$_0$) ← (A) (TPS$_3$–TPS$_0$) ← (A)	116 130
	TAB1	(B) ← (T1$_7$–T1$_4$) (A) ← (T1$_3$–T1$_0$)	103 130
	T1AB	(R1$_7$–R1$_4$) ← (B) (T1$_7$–T1$_4$) ← (B) (R1$_3$–R1$_0$) ← (A) (T1$_3$–T1$_0$) ← (A)	102 130
	TR1AB	(R1$_7$–R1$_4$) ← (B) (R1$_3$–R1$_0$) ← (A)	117 130
	TAB2	(B) ← (T2$_7$–T2$_4$) (A) ← (T2$_3$–T2$_0$)	104 130
	T2AB	(R2L$_7$–R2L$_4$) ← (B) (T2$_7$–T2$_4$) ← (B) (R2L$_3$–R2L$_0$) ← (A) (T2$_3$–T2$_0$) ← (A)	102 130
	T2R2L	(T2$_7$–T2$_0$) ← (R2L$_7$–R2L$_0$)	103 130
	T2HAB	(R2H$_7$–R2H$_4$) ← (B) (R2H$_3$–R2H$_0$) ← (A)	103 130

p=0 to 47

INDEX LIST OF INSTRUCTION FUNCTION (continued)

Grouping	Mnemonic	Function	Page	
Timer operation	TLCA	(RLC) ← (A) (TLC) ← (A)	115	130
	SNZT1	$V1_2$ = 0 : (T1F) = 1 ? (T1F) ← 0 $V1_2$ = 1 : SNZT1=NOP	100	130
	SNZT2	$V1_3$ = 0 : (T2F) = 1 ? (T2F) ← 0 $V1_3$ = 1 : SNZT2=NOP	100	130
	SNZT3	$V2_0$ = 0 : (T3F) = 1 ? (T3F) ← 0 $V2_0$ = 1 : SNZT3=NOP	100	130
Input/Output operation	IAP0	(A) ← (P0)	91	132
	OP0A	(P0) ← (A)	93	132
	IAP1	(A) ←(P1)	91	132
	OP1A	(P1) ← (A)	94	132
	IAP2	(A) ← (P2)	92	132
	OP2A	(P2) ← (A)	94	132
	IAP3	(A) ← (P3)	92	132
	OP3A	(P3) ← (A)	94	132
	CLD	(D) ← 1	89	132
	RD	(D(Y)) ← 0, (Y) = 0 to 7	96	132
	SD	(D(Y)) ← 1, (Y) = 0 to 7	98	132
	SZD	(D(Y)) = 0 ?, (Y) = 0 to 5	102	132
	RCP	(C) ← 0	96	132
	SCP	(C) ← 1	98	132
	TFR0A	(FR0) ← (A)	112	132
	TFR1A	(FR1) ← (A)	112	132
	TFR2A	(FR2) ← (A)	112	132
	TFR3A	(FR3) ← (A)	113	132
	TAPU0	(A) ← (PU0)	107	132
	TPU0A	(PU0) ← (A)	116	132
	TAPU1	(A) ← (PU1)	107	132
	TPU1A	(PU1) ← (A)	116	132
	TAPU2	(A) ← (PU2)	107	132
	TPU2A	(PU2) ← (A)	117	132
	TAPU3	(A) ← (PU3)	108	132

Grouping	Mnemonic	Function	Page	
Input/Output operation	TPU3A	(PU3) ← (A)	117	132
	TAK0	(A) ← (K0)	105	134
	TK0A	(K0) ← (A)	113	134
	TAK1	(A) ← (K1)	105	134
	TK1A	(K1) ← (A)	113	134
	TAK2	(A) ← (K2)	106	134
	TK2A	(K2) ← (A)	114	134
	TAK3	(A) ← (K3)	106	134
	TK3A	(K3) ← (A)	114	134
LCD operation	TAL1	(A) ← (L1)	106	134
	TL1A	(L1) ← (A)	114	134
	TL2A	(L2) ← (A)	114	134
	TL3A	(L3) ← (A)	115	134
	TC1A	(C1) ← (A)	111	134
	TC2A	(C2) ← (A)	111	134
	TC3A	(C3) ← (A)	111	134
Clock operation	CRCK	RC oscillation selected	90	134
	TAMR	(A) ← (MR)	107	134
	TMRA	(MR) ← (A)	115	134
	TRGA	(RG_2–RG_0) ← (A_2–A_0)	117	134
Other operation	NOP	(PC) ← (PC)+1	93	136
	POF	Transition to clock operating	95	136
	POF2	Transition to RAM back-up	95	136
	EPOF	POF or POF2 instruction	91	136
	SNZP	(P) = 1 ?	99	136
	SNZVD	(VDF) = 1?	100	136
	WRST	(WDF1) = 1 ? (WDF1) ← 0	119	136
	DWDT	Stop of watchdog timer function enabled	90	136
	SRST	System reset	101	136
	RUPT	(UPTF) ←0	97	136
	SUPT	(UPTF) ←1	101	136
	SVDE	At power down mode, voltage drop detection circuit valid	101	136

MACHINE INSTRUCTIONS (INDEX BY ALPHABET)

A n (Add n and accumulator)

Instruction code: D_9 … D_0

0	0	0	1	1	0	n	n	n	n	2	0	6	n	16

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

Operation: (A) ← (A) + n
n = 0 to 15

Grouping: Arithmetic operation

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

AM (Add accumulator and Memory)

Instruction code: D_9 … D_0

0	0	0	0	0	0	1	0	1	0	2	0	0	A	16

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

Operation: (A) ← (A)Å{(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

0	0	0	0	0	0	1	0	1	1	2	0	0	B	16

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

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

Grouping: Arithmetic operation

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

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.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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: $(PC_L) \leftarrow a_6$ 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	a	a	16

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

Operation: $(PC_H) \leftarrow p$
$(PC_L) \leftarrow a_6$ to a_0

Grouping: Branch operation

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

Note: p = 0 to 47

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: $(PC_H) \leftarrow p$
$(PC_L) \leftarrow (DR_2–R_0, A_3–A_0)$

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 = 0 to 47

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.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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

Instruction code:

D9									D0					
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	a	a	16

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

Operation:
(SP) ← (SP) + 1
(SK(SP)) ← (PC)
(PC_H) ← p
(PC_L) ← a_6–a_0

Grouping: Subroutine call operation

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

Note: p = 0 to 47
Be careful not to over the stack because the maximum level of subroutine nesting is 8.

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

Instruction code:

D9									D0					
0	0	0	0	1	1	0	0	0	0	2	0	3	0	16
1	0	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) ← (SP) + 1
(SK(SP)) ← (PC)
(PC_H) ← p
(PC_L) ← (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 = 0 to 47
Be careful not to over the stack because the maximum level of subroutine nesting is 8.

CLD (CLear port D)

Instruction code:

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

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

Operation: (D) ← 1

Grouping: Input/Output operation

Description: Sets (1) to port D.

CMA (CoMplement of Accumulator)

Instruction code:

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

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

Operation: (A) ← $(\overline{A})$

Grouping: Arithmetic operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

CRCK (Clock select: Rc oscillation ClocK)

Instruction code (D9 … D0)	Number of words	Number of cycles	Flag CY	Skip condition
1 0 1 0 0 1 1 0 1 1 (2) = 2 9 B (16)	1	1	-	-

Operation: RC oscillation circuit selected

Grouping: Clock control operation

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

DEY (DEcrement register Y)

Instruction code (D9 … D0)	Number of words	Number of cycles	Flag CY	Skip condition
0 0 0 0 0 1 0 1 1 1 (2) = 0 1 7 (16)	1	1	-	(Y) = 15

Operation: (Y) ← (Y) –1

Grouping: RAM addresses

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

DI (Disable Interrupt)

Instruction code (D9 … D0)	Number of words	Number of cycles	Flag CY	Skip condition
0 0 0 0 0 0 0 1 0 0 (2) = 0 0 4 (16)	1	1	-	-

Operation: (INTE) ← 0

Grouping: Interrupt control operation

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

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

DWDT (Disable WatchDog Timer)

Instruction code (D9 … D0)	Number of words	Number of cycles	Flag CY	Skip condition
1 0 1 0 0 1 1 1 0 0 (2) = 2 9 C (16)	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.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

EI (Enable Interrupt)

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

EPOF (Enable POF instruction)

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

Operation: POF instruction or POF2 instruction valid

Grouping: Other operation

Description: Makes the immediate after POF instruction or POF2 instruction valid by executing the EPOF instruction.

IAP0 (Input Accumulator from port P0)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 1 1 0 0 0 0 0 D_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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 1 1 0 0 0 0 1 D_0 $_2$ 2 6 1 $_{16}$	1	1	-	-

Operation: (A) ← (P1)

Grouping: Input/Output operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

IAP2 (Input Accumulator from port P2)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 1 1 0 0 0 1 0 D_0 $_2$ 2 6 2 $_{16}$	1	1	-	-

Operation: (A) ← (P2)

Grouping: Input/Output operation

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

IAP3 (Input Accumulator from port P3)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 1 1 0 0 0 1 1 D_0 $_2$ 2 6 3 $_{16}$	1	1	-	-

Operation: (A) ← (P3)

Grouping: Input/Output operation

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

INY (INcrement register Y)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 0 1 0 0 1 1 D_0 $_2$ 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 0 0 0 1 1 1 n n n n D_0 $_2$ 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.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D9 1 1 x_3 x_2 x_1 x_0 y_3 y_2 y_1 y_0 D0 $_2$ = 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
D9 0 0 0 1 0 0 1 0 Z_1 Z_0 D0 $_2$ = 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.

NOP (No OPeration)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D9 0 0 0 0 0 0 0 0 0 0 D0 $_2$ = 0 0 0 $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D9 1 0 0 0 1 0 0 0 0 0 D0 $_2$ = 2 2 0 $_{16}$	1	1	-	-

Operation: (P0) ← (A)

Grouping: Input/Output operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

OP1A (Output port P1 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 1 0 0 0 0 1 D_0 ${}_2$ 2 2 1 ${}_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 1 0 0 0 1 0 D_0 ${}_2$ 2 2 2 ${}_{16}$	1	1	-	-

Operation: (P2) ← (A)

Grouping: Input/Output operation

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

OP3A (Output port P3 from Accumulator)

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

Operation: (P3) ← (A)

Grouping: Input/Output operation

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

OR (logical OR between accumulator and memory)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 0 1 1 0 0 1 D_0 ${}_2$ 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.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

POF (Power OFf)

Instruction code: D_9 … D_0

0	0	0	0	0	0	0	0	1	0	2	0	0	2	16

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

Operation: Transition to clock operating mode

Grouping: Other operation

Description: Puts the system in clock operating mode by executing the POF2 instruction after executing the EPOF instruction.

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

POF2 (Power OFf2)

Instruction code: D_9 … D_0

0	0	0	0	0	0	1	0	0	0	2	0	0	8	16

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

Operation: Transition to RAM back-up mode

Grouping: Other operation

Description: Puts the system in RAM back-up state by executing the POF2 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.

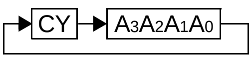
RAR (Rotate Accumulator Right)

Instruction code: D_9 … D_0

0	0	0	0	0	1	1	1	0	1	2	0	1	D	16

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

Operation:



Grouping: Arithmetic operation

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

RB j (Reset Bit)

Instruction code: D_9 … D_0

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

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

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

Operation: (CY) ← 0

Grouping: Arithmetic operation

Description: Clears (0) to carry flag CY.

RCP (Reset Port C)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 1 0 0 0 1 1 0 0 D_0 $_2$ = 2 8 C $_{16}$	1	1	-	-

Operation: (C) ← 0

Grouping: Input/Output operation

Description: Clears (0) to port C.

RD (Reset port D specified by register Y)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 0 1 0 1 0 0 D_0 $_2$ = 0 1 4 $_{16}$	1	1	-	-

Operation: (D(Y)) ← 0
(Y) = 0 to 7

Grouping: Input/Output operation

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

Note: (Y) = 0 to 7.
Do not execute this instruction if values except above are set to register Y.

RT (ReTurn from subroutine)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 1 0 0 0 1 0 0 D_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.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

RTI (ReTurn from Interrupt)

Instruction code (D_9 … D_0)	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.

RTS (ReTurn from subroutine and Skip)

Instruction code (D_9 … D_0)	Number of words	Number of cycles	Flag CY	Skip condition
0 0 0 1 0 0 0 1 0 1 $_2$ 0 4 5 $_{16}$	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.

RUPT (Reset UPT flag)

Instruction code (D_9 … D_0)	Number of words	Number of cycles	Flag CY	Skip condition
0 0 0 1 0 1 1 0 0 0 $_2$ 0 5 8 $_{16}$	1	1	-	-

Operation: (UPTF) ←0

Grouping: Other operation

Description: Clears (0) to the high-order bit reference enable flag UPTF.

Note: Even when the table reference instruction (TABP p) is executed, the high-order 2 bits of ROM reference data is not transferred to register D.

SB j (Set Bit)

Instruction code (D_9 … D_0)	Number of words	Number of cycles	Flag CY	Skip condition
0 0 0 1 0 1 1 1 j j $_2$ 0 5 C+j $_{16}$	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).

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

SC (Set Carry flag)

Instruction code: D_9 … D_0

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.

SCP (Set Port C)

Instruction code: D_9 … D_0

1	0	1	0	0	0	1	1	0	1	2	2	8	D	16

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

Operation: (C) ← 1

Grouping: Input/Output operation

Description: Sets (1) to port C.

SD (Set port D specified by register Y)

Instruction code: D_9 … D_0

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.

Note: (Y) = 0 to 7.
Do not execute this instruction if values except above are set to register Y.

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 n = 0 to 15

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

Grouping: Comparison operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

SEAM (Skip Equal, Accumulator with Memory)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 1 0 0 1 1 0 D_0 $_2$ 0 2 6 $_{16}$	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 interrupt 0 request flag)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 1 1 1 0 0 0 D_0 $_2$ 0 3 8 $_{16}$	1	1	-	$V1_0$ = 0 : (EXF0) = 1

Operation: $V1_0$ = 0 : (EXF0) = 1 ?
(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 : Clears (0) to the EXF0 flag and skips the next instruction when external 0 interrupt request flag EXF0 is “1”. When the EXF0 flag is “0”, executes the next instruction.
When $V1_0$ = 1 : This instruction is equivalent to the NOP instruction.

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

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 1 1 1 0 1 0 D_0 $_2$ 0 3 A $_{16}$	1	1	-	$I1_2$ = 0 : (INT0) = “L” $I1_2$ = 1 : (INT0) = “H”

Operation: $I1_2$ = 0 : (INT) = “L” ?
$I1_2$ = 1 : (INT) = “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 INT pin is “L”. Executes the next instruction when the level of INT pin is “H”.
When $I1_2$ = 1 : Skips the next instruction when the level of INT pin is “H.” Executes the next instruction when the level of INT pin is “L”.

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

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 0 0 0 0 1 1 D_0 $_2$ 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)

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

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 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 ?
(T1F) ← 0
$V1_2 = 1$: SNZT1 = NOP
($V1_2$ = bit 2 of interrupt control register V1)

Grouping: Timer operation

Description: When $V1_2 = 0$: Clears (0) to the T1F flag and skips the next instruction when timer 1 interrupt request flag T1F is “1”. 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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 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 ?
(T2F) ← 0
$V1_3 = 1$: SNZT2 = NOP
($V1_3$ = bit 3 of interrupt control register V1)

Grouping: Timer operation

Description: When $V1_3 = 0$: Clears (0) to the T2F flag and skips the next instruction when timer 2 interrupt request flag T2F is “1”. 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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 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 ?
(T3F) ← 0
$V2_0 = 1$: SNZT3 = NOP

Grouping: Timer operation

Description: When $V2_0 = 0$: Clears (0) to the T3F flag and skips the next instruction when timer 3 interrupt request flag T3F is “1”. When the T3F flag is “0”, executes the next instruction.
When $V2_0 = 1$: This instruction is equivalent to the NOP instruction.

SNZVD (Skip if Non Zero condition of Voltage Detector flag)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 1 0 1 0 0 0 1 0 1 0 $_2$ = 2 8 A $_{16}$	1	1	-	$V2_3 = 0$: (VDF) = 1

Operation: (VDF) = 1?

Grouping: Other operation

Description: Skips the next instruction when voltage drop detection circuit flag VDF is “1”. Execute instruction when VDF is “0”.
After skipping, the contents of VDF remains unchanged.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

SRST (System ReSet)

Instruction code: D9 … D0

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

Grouping: Other operation

Description: System reset occurs.

SUPT (Set UPT flag)

Instruction code: D9 … D0

0	0	0	1	0	1	1	0	0	1	2	0	5	9	16

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

Operation: (UPTF) ←1

Grouping: Other operation

Description: Sets (1) to the high-order bit reference enable flag UPTF. When the table reference instruction (TABP p) is executed, the high-order 2 bits of ROM reference data is transferred to the low-order 2 bits of register D.

SVDE (Set Voltage Detector Enable flag)

Instruction code: D9 … D0

1	0	1	0	0	1	0	0	1	1	2	2	9	3	16

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

Operation: Voltage drop detection circuit valid at powerdown mode.

Grouping: Other operation

Description: Voltage drop detection circuit is valid at powerdown mode (clock operating mode, RAM back-up mode)

Note: This instruction can be used only for H version.

SZB j (Skip if Zero, Bit)

Instruction code: D9 … D0

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

0	0	0	0	1	0	1	1	1	1	2	0	2	F	16

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

Operation: (CY) = 0 ?

Grouping: Arithmetic operation

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

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

Instruction code: D_9 … D_0

0	0	0	0	1	0	0	1	0	0	2	0	2	4	16
0	0	0	0	1	0	1	0	1	1	2	0	2	B	16

Number of words	Number of cycles	Flag CY	Skip condition
2	2	-	(D(Y)) = 0

Operation: (D(Y)) = 0 ?
(Y) = 0 to 5

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”.

Note: (Y) = 0 to 5.
Do not execute this instruction if values except above are set to register Y.

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

Instruction code: D_9 … D_0

1	0	0	0	1	1	0	0	0	0	2	2	3	0	16

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

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

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 R2L from Accumulator and register B)

Instruction code: D_9 … D_0

1	0	0	0	1	1	0	0	0	1	2	2	3	1	16

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

Operation: $(T2_7–T2_4) \leftarrow (B)$
$(R2L_7–R2L_4) \leftarrow (B)$
$(T2_3–T2_0) \leftarrow (A)$
$(R2L_3–R2L_0) \leftarrow (A)$

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits ($T2_7–T2_4$) of timer 2 and the high-order 4 bits ($R2L_7–R2L_4$) of timer 2 reload register R2L. Transfers the contents of register A to the low-order 4 bits ($T2_3–T2_0$) of timer 2 and the low-order 4 bits ($R2L_3–R2L_0$) of timer 2 reload register R2.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

T2HAB (Transfer data to register R2H from Accumulator and register B)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 1 0 0 1 0 1 0 0 D_0 $_2$ 2 9 4 $_{16}$	1	1	-	-

Operation: $(R2H_7–R2H_4) \leftarrow (B)$
$(R2H_3–R2H_0) \leftarrow (A)$

Grouping: Timer operation

Description: Transfers the contents of register B to the high-order 4 bits of timer 2 and timer 2 reload register R2H. Transfers the contents of register A to the low-order 4 bits of timer 2 and timer 2 reload register R2H.

T2R2L (Transfer data to timer 2 from register R2L)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 1 0 0 1 0 1 0 1 D_0 $_2$ 2 9 5 $_{16}$	1	1	-	-

Operation: $(T2_7–T2_0) \leftarrow (R2L_7–R2L_0)$

Grouping: Timer operation

Description: Transfers the contents of reload register R2L to timer 2.

TAB (Transfer data to Accumulator from register B)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 0 1 1 1 1 0 D_0 $_2$ 0 1 E $_{16}$	1	1	-	-

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

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

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

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.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

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

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

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 0 0 0 0 1 0 1 0 1 0 D_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 0 0 1 0 p_5 p_4 p_3 p_2 p_1 p_0 D_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)
(B) ← $(ROM(PC))_{7–4}$
(A) ← $(ROM(PC))_{3–0}$
(UPTF) ← 1
(DR_1, DR_0) ← $(ROM(PC))_{9, 8}$
(DR_2) ← 0
(PC) ← (SK(SP))
(SP) ← (SP) – 1

Grouping: Arithmetic operation

Description: Transfers bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 7 to 0 are the ROM pattern in address $(DR_2\ DR_1\ DR_0\ A_3\ A_2\ A_1\ A_0)_2$ specified by registers A and D in page p. When UPTF is 1, Transfers bits 9, 8 to the low-order 2 bits (DR_1, DR_0) of register D, and “0” is stored to the least significant bit (DR_2) of register D.
When this instruction is executed, 1 stage of stack register (SK) is used.

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

TABPS (Transfer data to Accumulator and register B from Pre-Scaler)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 1 1 1 0 1 0 1 D_0 $_2$ 2 7 5 $_{16}$	1	1	-	-

Operation: (B) ← (TPS_7–TPS_4)
(A) ← (TPS_3–TPS_0)

Grouping: Timer operation

Description: Transfers the high-order 4 bits of prescaler to register B.
Transfers the low-order 4 bits of prescaler to register A.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAD (Transfer data to Accumulator from register D)

Instruction code: D_9 … D_0

0	0	0	1	0	1	0	0	0	1	2	0	5	1	16

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

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

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.
"0" is stored to the bit 3 (A_3) of register A.

TAI1 (Transfer data to Accumulator from register I1)

Instruction code: D_9 … D_0

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: $(A) \leftarrow (I1)$

Grouping: Interrupt operation

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

TAK0 (Transfer data to Accumulator from register K0)

Instruction code: D_9 … D_0

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: $(A) \leftarrow (K0)$

Grouping: Input/Output operation

Description: Transfers the contents of key-on wakeup control register K0 to register A.

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: $(A) \leftarrow (K1)$

Grouping: Input/Output operation

Description: Transfers the contents of key-on wakeup control register K1 to register A.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAK2 (Transfer data to Accumulator from register K2)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 1 0 1 1 0 1 0 D_0 $_2$ 2 5 A $_{16}$	1	1	-	-

Operation: (A) ← (K2)

Grouping: Input/Output operation

Description: Transfers the contents of key-on wakeup control register K2 to register A.

TAK3 (Transfer data to Accumulator from register K3)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 1 0 1 1 0 1 1 D_0 $_2$ 2 5 B $_{16}$	1	1	-	-

Operation: (A) ← (K3)

Grouping: Input/Output operation

Description: Transfers the contents of key-on wakeup control register K3 to register A.

TAL1 (Transfer data to Accumulator from register L1)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 1 0 0 1 0 1 0 D_0 $_2$ 2 4 A $_{16}$	1	1	-	-

Operation: (A) ← (L1)

Grouping: LCD operation

Description: Transfers the contents of LCD control register L1 to register A.

TAM j (Transfer data to Accumulator from Memory)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 1 1 0 0 j j j j D_0 $_2$ 2 C j $_{16}$	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 1 0 0 1 0 1 0 0 1 0 D_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 1 0 0 1 0 1 0 1 1 1 D_0 $_2$ 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 1 0 0 1 0 1 1 1 1 0 D_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.

TAPU2 (Transfer data to Accumulator from register PU2)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 1 0 1 1 1 1 1 D_0 $_2$ 2 5 F $_{16}$	1	1	-	-

Operation: (A) ← (PU2)

Grouping: Input/Output operation

Description: Transfers the contents of pull-up control register PU2 to register A.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAPU3 (Transfer data to Accumulator from register PU3)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 1 0 0 1 0 1 1 1 0 1 ($_2$) = 2 5 D ($_{16}$)	1	1	-	-

Operation: (A) ← (PU3)

Grouping: Input/Output operation

Description: Transfers the contents of pull-up control register PU3 to register A.

TASP (Transfer data to Accumulator from Stack Pointer)

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 0 ($_2$) = 0 5 0 ($_{16}$)	1	1	-	-

Operation: $(A_2–A_0) \leftarrow (SP_2–SP_0)$
$(A_3) \leftarrow 0$

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.
“0” is stored to the bit 3 (A_3) of register A.

TAV1 (Transfer data to Accumulator from register V1)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 0 0 0 1 0 1 0 1 0 0 ($_2$) = 0 5 4 ($_{16}$)	1	1	-	-

Operation: (A) ← (V1)

Grouping: Interrupt operation

Description: Transfers the contents of interrupt control register V1 to register A.

TAV2 (Transfer data to Accumulator from register V2)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 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.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAW1 (Transfer data to Accumulator from register W1)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 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.

TAW4 (Transfer data to Accumulator from register W4)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 1 0 0 1 0 0 1 1 1 0 $_2$ = 2 4 E $_{16}$	1	1	-	-

Operation: (A) ← (W4)

Grouping: Timer operation

Description: Transfers the contents of timer control register W4 to register A.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TAX (Transfer data to Accumulator from register X)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 1 0 1 0 0 1 0 D_0 $_2$ 0 5 2 $_{16}$	1	1	-	-

Operation: (A) ← (X)

Grouping: Register to register transfer

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

TAY (Transfer data to Accumulator from register Y)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 0 1 1 1 1 1 D_0 $_2$ 0 1 F $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 1 0 1 0 0 1 1 D_0 $_2$ 0 5 3 $_{16}$	1	1	-	-

Operation: (A_1, A_0) ← (Z_1, Z_0)
(A_3, A_2) ← 0

Grouping: Register to register transfer

Description: Transfers the contents of register Z to the low-order 2 bits (A_1, A_0) of register A. “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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 0 0 1 1 1 0 D_0 $_2$ 0 0 E $_{16}$	1	1	-	-

Operation: (B) ← (A)

Grouping: Register to register transfer

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TC1A (Transfer data to register C1 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 1 0 1 0 1 0 0 0 D_0 $_2$ = 2 A 8 $_{16}$	1	1	-	-

Operation: (C1) ← (A)

Grouping: LCD control operation

Description: Transfers the contents of register A to the LCD control register C1.

TC2A (Transfer data to register C2 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 1 0 1 0 1 0 0 1 D_0 $_2$ = 2 A 9 $_{16}$	1	1	-	-

Operation: (C2) ← (A)

Grouping: LCD control operation

Description: Transfers the contents of register A to the LCD control register C2.

TC3A (Transfer data to register C3 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 1 0 0 1 1 0 D_0 $_2$ = 2 2 6 $_{16}$	1	1	-	-

Operation: (C3) ← (A)

Grouping: LCD control operation

Description: Transfers the contents of register A to the LCD control register C3.

TDA (Transfer data to register D from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 1 0 1 0 0 1 D_0 $_2$ = 0 2 9 $_{16}$	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		Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0	0 0 0 0 0 1 1 0 1 0 ₂ = 0 1 A ₁₆	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 $(E_3–E_0)$ 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		Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0	1 0 0 0 1 0 1 0 0 0 ₂ = 2 2 8 ₁₆	1	1	-	-

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

Grouping: Input/Output operation

Description: Transfers the contents of register A to port output structure control register FR0.

TFR1A (Transfer data to register FR1 from Accumulator)

Instruction code		Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0	1 0 0 0 1 0 1 0 0 1 ₂ = 2 2 9 ₁₆	1	1	-	-

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

Grouping: Input/Output operation

Description: Transfers the contents of register A to port output structure control register FR1.

TFR2A (Transfer data to register FR2 from Accumulator)

Instruction code		Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0	1 0 0 0 1 0 1 0 1 0 ₂ = 2 2 A ₁₆	1	1	-	-

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

Grouping: Input/Output operation

Description: Transfers the contents of register A to port output structure control register FR2.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TFR3A (Transfer data to register FR3 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 1 0 1 0 1 1 D_0 $_2$ 2 2 B $_{16}$	1	1	-	-

Operation: (FR3) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to port output structure control register FR3.

TI1A (Transfer data to register I1 from Accumulator)

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

Operation: (I1) ← (A)

Grouping: Interrupt operation

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

TK0A (Transfer data to register K0 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 0 1 1 0 1 1 D_0 $_2$ 2 1 B $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 0 1 0 1 0 0 D_0 $_2$ 2 1 4 $_{16}$	1	1	-	-

Operation: (K1) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to key-on wakeup control register K1.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TK2A (Transfer data to register K2 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D9 1 0 0 0 0 1 0 1 0 1 D0 $_2$ 2 1 5 $_{16}$	1	1	-	-

Operation: (K2) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to key-on wakeup control register K2.

TK3A (Transfer data to register K3 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D9 1 0 0 0 1 0 1 1 0 0 D0 $_2$ 2 2 C $_{16}$	1	1	-	-

Operation: (K3) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to key-on wakeup control register K3.

TL1A (Transfer data to register L1 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D9 1 0 0 0 0 0 1 0 1 0 D0 $_2$ 2 0 A $_{16}$	1	1	-	-

Operation: (L1) ← (A)

Grouping: LCD control operation

Description: Transfers the contents of register A to the LCD control register L1.

TL2A (Transfer data to register L2 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D9 1 0 0 0 0 0 1 0 1 1 D0 $_2$ 2 0 B $_{16}$	1	1	-	-

Operation: (L2) ← (A)

Grouping: LCD control operation

Description: Transfers the contents of register A to the LCD control register L2.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TL3A (Transfer data to register L3 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 0 0 1 1 0 0 D_0 ${}_2$ 2 0 C ${}_{16}$	1	1	-	-

Operation: (L3) ← (A)

Grouping: LCD control operation

Description: Transfers the contents of register A to the LCD control register L3.

TLCA (Transfer data to timer LC and register RLC from Accumulator)

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

Operation: (LC) ← (A)
(RLC) ← (A)

Grouping: Timer control operation

Description: Transfers the contents of register A to timer LC and reload register RLC.

TMA j (Transfer data to Memory from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 1 0 1 1 j j j j D_0 ${}_2$ 2 B j ${}_{16}$	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.

TMRA (Transfer data to register MR from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 0 1 0 1 1 0 D_0 ${}_2$ 2 1 6 ${}_{16}$	1	1	-	-

Operation: (MR) ← (A)

Grouping: Clock operation

Description: Transfers the contents of register A to clock control register MR.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TPAA (Transfer data to register PA from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 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 least significant bit of register A (A_0) to timer control register PA.

TPSAB (Transfer data to Pre-Scaler and register RPS from Accumulator and register B)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 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. 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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 1 0 0 0 1 0 1 1 0 1 $_2$ = 2 2 D $_{16}$	1	1	-	-

Operation: $(PU0) \leftarrow (A)$

Grouping: Input/Output operation

Description: Transfers the contents of register A to pull-up control register PU0.

TPU1A (Transfer data to register PU1 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 … D_0: 1 0 0 0 1 0 1 1 1 0 $_2$ = 2 2 E $_{16}$	1	1	-	-

Operation: $(PU1) \leftarrow (A)$

Grouping: Input/Output operation

Description: Transfers the contents of register A to pull-up control register PU1.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TPU2A (Transfer data to register PU2 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 1 0 1 1 1 1 D_0 $_2$ = 2 2 F $_{16}$	1	1	-	-

Operation: (PU2) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to pull-up control register PU2.

TPU3A (Transfer data to register PU3 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 0 0 1 0 0 0 D_0 $_2$ = 2 0 8 $_{16}$	1	1	-	-

Operation: (PU3) ← (A)

Grouping: Input/Output operation

Description: Transfers the contents of register A to pull-up control register PU3.

TR1AB (Transfer data to register R1 from Accumulator and register B)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 1 1 1 1 1 1 D_0 $_2$ = 2 3 F $_{16}$	1	1	-	-

Operation: $(R1_7–R1_4) \leftarrow (B)$
$(R1_3–R1_0) \leftarrow (A)$

Grouping: Timer control operation

Description: Transfers the contents of register B to the high-order 4 bits ($R1_7–R1_4$) of timer 1 reload register R1, and the contents of register A to the low-order 4 bits ($R1_3–R1_0$) of timer 1 reload register R1.

TRGA (Transfer data to register RG from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 0 0 1 0 0 1 D_0 $_2$ = 2 0 9 $_{16}$	1	1	-	-

Operation: $(RG_2–RG_0) \leftarrow (A_2–A_0)$

Grouping: Clock control operation

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

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

TV1A (Transfer data to register V1 from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 1 1 1 1 1 1 D_0 $_2$ 0 3 F $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 1 1 1 1 1 0 D_0 $_2$ 0 3 E $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 0 0 1 1 1 0 D_0 $_2$ 2 0 E $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 0 0 1 1 1 1 D_0 $_2$ 2 0 F $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 0 1 0 0 0 0 D_0 $_2$ 2 1 0 $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 0 0 0 1 0 0 0 1 D_0 $_2$ 2 1 1 $_{16}$	1	1	-	-

Operation: (W4) ← (A)

Grouping: Timer operation

Description: Transfers the contents of register A to timer control register W4.

TYA (Transfer data to register Y from Accumulator)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 0 0 0 0 0 0 1 1 0 0 D_0 $_2$ 0 0 C $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 1 0 1 0 0 0 0 0 D_0 $_2$ 2 A 0 $_{16}$	1	1	-	(WDF1) = 1

Operation: (WDF1) = 1 ?
(WDF1) ← 0

Grouping: Other operation

Description: Clears (0) to the WDF1 flag and skips the next instruction when watchdog timer flag WDF1 is “1”. 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.

MACHINE INSTRUCTIONS (INDEX BY ALPHABET) (continued)

XAM j (eXchange Accumulator and Memory data)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 1 1 0 1 j j j j D_0 $_2$ 2 D j $_{16}$	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	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 1 1 1 1 j j j j D_0 $_2$ 2 F j $_{16}$	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.

XAMI j (eXchange Accumulator and Memory data and Increment register Y and skip)

Instruction code	Number of words	Number of cycles	Flag CY	Skip condition
D_9 1 0 1 1 1 0 j j j j D_0 $_2$ 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	Detailed 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_3–E_0) 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 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. “0” is stored to the bit 3 (A_3) of register A.
–	–	Transfers the contents of register Z to the low-order 2 bits (A_1, A_0) of register A. “0” is stored to the high-order 2 bits (A_3, A_2) 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. “0” is stored to the bit 3 (A_3) 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 1) (PC_L) ← (DR_2–DR_0, A_3–A_0) (B) ← $(ROM(PC))_{7-4}$ (A) ← $(ROM(PC))_{3-0}$ (UPTF) = 1 (DR_1, DR_0) ← $(ROM(PC))_{9,\ 8}$ (DR_2) ← 0 (PC) ← (SK(SP)) (SP) ← (SP) – 1
	AM	0	0	0	0	0	0	1	0	1	0	0 0 A	1	1	(A) ← (A) + (M(DP))
	AMC	0	0	0	0	0	0	1	0	1	1	0 0 B	1	1	(A) ← (A) + (M(DP)) + (CY) (CY) ← Carry
	A n	0	0	0	1	1	0	n	n	n	n	0 6 n	1	1	(A) ← (A) + n n = 0 to 15
	AND	0	0	0	0	0	1	1	0	0	0	0 1 8	1	1	(A) ← (A) AND (M(DP))
	OR	0	0	0	0	0	1	1	0	0	1	0 1 9	1	1	(A) ← (A) OR (M(DP))
	SC	0	0	0	0	0	0	0	1	1	1	0 0 7	1	1	(CY) ← 1
	RC	0	0	0	0	0	0	0	1	1	0	0 0 6	1	1	(CY) ← 0
	SZC	0	0	0	0	1	0	1	1	1	1	0 2 F	1	1	(CY) = 0 ?
	CMA	0	0	0	0	0	1	1	1	0	0	0 1 C	1	1	(A) ← $\overline{(A)}$
	RAR	0	0	0	0	0	1	1	1	0	1	0 1 D	1	1	→ CY → $A_3A_2A_1A_0$
Bit operation	SB j	0	0	0	1	0	1	1	1	j	j	0 5 C+j	1	1	(Mj(DP)) ← 1 j = 0 to 3
	RB j	0	0	0	1	0	0	1	1	j	j	0 4 C+j	1	1	(Mj(DP)) ← 0 j = 0 to 3
	SZB j	0	0	0	0	1	0	0	0	j	j	0 2 j	1	1	(Mj(DP)) = 0 ? j = 0 to 3

Note 1.M34571G4: p=0 to 31, M34571G6: p=0 to 47 and M34571GD: p=0 to 127.

Skip condition	Carry flag CY	Detailed description
Continuous description	–	Loads the value n in the immediate field to register A. When the LA instructions are continuously coded and executed, only the first LA instruction is executed and other LA instructions coded continuously are skipped.
–	–	Transfers bits 7 to 4 to register B and bits 3 to 0 to register A. These bits 7 to 0 are the ROM pattern in address $(DR_2\ DR_1\ DR_0\ A_3\ A_2\ A_1\ A_0)_2$ specified by registers A and D in page p. When UPTF is 1, Transfers bits 9, 8 to the low-order 2 bits (DR_1, DR_0) of register D, and “0” is stored to the least significant bit (DR_2) of register D. When this instruction is executed, 1 stage of stack register (SK) 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”. Executes the next instruction when the contents of carry flag CY is “1”. The contents of carry flag CY remains unchanged.
–	–	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”.

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					
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			
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 1) $(PC_L) \leftarrow a_6–a_0$
		1	0	p_5	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	a	a			
	BLA p	0	0	0	0	0	1	0	0	0	0	0	1	0	2	2	$(PC_H) \leftarrow p$ (Note 1) $(PC_L) \leftarrow (DR_2–DR_0, A_3–A_0)$
		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 1) $(PC_L) \leftarrow a_6–a_0$
		1	0	p_5	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	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 1) $(PC_L) \leftarrow (DR_2–DR_0, A_3–A_0)$
		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 1.M34571G4: p=0 to 31, M34571G6: p=0 to 47 and M34571GD: p=0 to 127.

Skip condition	Carry flag CY	Detailed description
(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 n = 0 to 15	–	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.
–	–	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 ? (EXF0) ← 0 $V1_0$ = 1 : SNZ0 = NOP
	SNZI0	0	0	0	0	1	1	1	0	1	0	0	3	A	1	1	$I1_2$ = 0 : (INT) = “L”? $I1_2$ = 1 : (INT) = “H”?
	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)
Timer operation	TPAA	1	0	1	0	1	0	1	0	1	0	2	A	A	1	1	(PA) ← (A)
	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	Detailed 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 $V1_0 = 0$: Clears (0) to the EXF0 flag and skips the next instruction when external 0 interrupt request flag EXF0 is “1”. 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)
(INT) = “L” However, $I1_2 = 0$ (INT) = “H” However, $I1_2 = 1$	–	When $I1_2 = 0$: Skips the next instruction when the level of INT 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 INT pin is “H”. Executes the next instruction when the level of INT0 pin is “L”. ($I1_2$: bit 2 of interrupt control register I1)
–	–	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 register A (A_0) to timer control register PA.
–	–	Transfers the contents of timer control register W1 to register A.
–	–	Transfers the contents of register A to timer control register W1.
–	–	Transfers the contents of timer control register W2 to register A.
–	–	Transfers the contents of register A to timer control register W2.
–	–	Transfers the contents of timer control register W3 to register A.
–	–	Transfers the contents of register A to timer control register W3.
–	–	Transfers the contents of timer control register W4 to register A.
–	–	Transfers the contents of register A to timer control register W4.

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Parameter / Type of instructions	Mnemonic	Instruction code													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	TABPS	1	0	0	1	1	1	0	1	0	1	2	7	5	1	1	$(B) \leftarrow (TPS_7–TPS_4)$ $(A) \leftarrow (TPS_3–TPS_0)$
	TPSAB	1	0	0	0	1	1	0	1	0	1	2	3	5	1	1	$(RPS_7–RPS_4) \leftarrow (B)$ $(TPS_7–TPS_4) \leftarrow (B)$ $(RPS_3–RPS_0) \leftarrow (A)$ $(TPS_3–TPS_0) \leftarrow (A)$
	TAB1	1	0	0	1	1	1	0	0	0	0	2	7	0	1	1	$(B) \leftarrow (T1_7–T1_4)$ $(A) \leftarrow (T1_3–T1_0)$
	T1AB	1	0	0	0	1	1	0	0	0	0	2	3	0	1	1	$(R1_7–R1_4) \leftarrow (B)$ $(T1_7–T1_4) \leftarrow (B)$ $(R1_3–R1_0) \leftarrow (A)$ $(T1_3–T1_0) \leftarrow (A)$
	TR1AB	1	0	0	0	1	1	1	1	1	1	2	3	F	1	1	$(R1_7–R1_4) \leftarrow (B)$ $(R1_3–R1_0) \leftarrow (A)$
	TAB2	1	0	0	1	1	1	0	0	0	1	2	7	1	1	1	$(B) \leftarrow (T2_7–T2_4)$ $(A) \leftarrow (T2_3–T2_0)$
	T2AB	1	0	0	0	1	1	0	0	0	1	2	3	1	1	1	$(R2L_7–R2L_4) \leftarrow (B)$ $(T2_7–T2_4) \leftarrow (B)$ $(R2L_3–R2L_0) \leftarrow (A)$ $(T2_3–T2_0) \leftarrow (A)$
	T2HAB	1	0	1	0	0	1	0	1	0	0	2	9	4	1	1	$(R2H_7–R2H_4) \leftarrow (B)$ $(R2H_3–R2H_0) \leftarrow (A)$
	T2R2L	1	0	1	0	0	1	0	1	0	1	2	9	5	1	1	$(T2_7) \leftarrow (R2L)$
	TLCA	1	0	0	0	0	0	1	1	0	1	2	0	D	1	1	$(RLC) \leftarrow (A)$ $(TLC) \leftarrow (A)$
	SNZT1	1	0	1	0	0	0	0	0	0	0	2	8	0	1	1	$V1_2 = 0$: (T1F) = 1 ? After skipping, (T1F) $\leftarrow$ 0 $V1_2 = 1$: SNZT1=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) $\leftarrow$ 0 $V1_3 = 1$: SNZT2=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) $\leftarrow$ 0 $V2_0 = 1$: SNZT3=NOP

Skip condition	Carry flag CY	Detailed description
–	–	Transfers the high-order 4 bits of prescaler to register B. 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. 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 ($T1_7$–$T1_4$) of timer 1 to register B. Transfers the low-order 4 bits ($T1_3$–$T1_0$) of timer 1 to register A.
–	–	Transfers the contents of register B to the high-order 4 bits of timer 1 and timer 1 reload register R1L. Transfers the contents of register A to the low-order 4 bits of timer 1 and timer 1 reload register R1L.
–	–	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.
–	–	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.
–	–	Transfers the contents of register B to the high-order 4 bits ($R2L_7$–$R2L_4$) of timer 2 and timer 2 reload register R2L. Transfers the contents of register A to the low-order 4 bits ($R2L_3$–$R2L_0$) of timer 2 and timer 2 reload register R2L.
–	–	Transfers the contents of register B to the high-order 4 bits ($R2H_7$–$R2H_4$) of timer 2 and timer 2 reload register R2H. Transfers the contents of register A to the low-order 4 bits ($R2H_3$–$R2H_0$) of timer 2 and timer 2 reload register R2H.
–	–	Transfers the contents of timer 2 reload register R2L to timer 2.
–	–	Transfers the contents of register A to timer LC and reload register RLC.
$V1_2 = 0$: (T1F) = 1	–	When $V1_2 = 0$: Clears (0) to the T1F flag and skips the next instruction when timer 1 interrupt request flag T1F is “1”. When the T1F flag is “0”, executes the next instruction. When $V1_2 = 1$: This instruction is equivalent to the NOP instruction. ($V1_2$: bit 2 of interrupt control register V1)
$V1_3 = 0$: (T2F) = 1	–	When $V1_3 = 0$: Clears (0) to the T2F flag and skips the next instruction when timer 2 interrupt request flag T2F is “1”. When the T2F flag is “0”, executes the next instruction. When $V1_3 = 1$: This instruction is equivalent to the NOP instruction. ($V1_3$: bit 3 of interrupt control register V1)
$V2_0 = 0$: (T3F) = 1	–	When $V2_0 = 0$: Clears (0) to the T3F flag and skips the next instruction when timer 3 interrupt request flag T3F is “1”. When the T3F flag is “0”, executes the next instruction. When $V2_0 = 1$: This instruction is equivalent to the NOP instruction. ($V2_0$: bit 0 of interrupt control register V2)

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	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) ← (P2)
	OP2A	1	0	0	0	1	0	0	0	1	0	2	2	2	1	1	(P2) ← (A)
	IAP3	1	0	0	1	1	0	0	0	1	1	2	6	3	1	1	(A) ← (P3)
	OP3A	1	0	0	0	1	0	0	0	1	1	2	2	3	1	1	(P3) ← (A)
	CLD	0	0	0	0	0	1	0	0	0	1	0	1	1	1	1	(D) ← 1
	RD	0	0	0	0	0	1	0	1	0	0	0	1	4	1	1	(D(Y)) ← 0 (Y) = 0 to 7
	SD	0	0	0	0	0	1	0	1	0	1	0	1	5	1	1	(D(Y)) ← 1 (Y) = 0 to 7
	SZD	0	0	0	0	1	0	0	1	0	0	0	2	4	2	2	(D(Y)) = 0 ? (Y) = 0 to 5
		0	0	0	0	1	0	1	0	1	1	0	2	B			
	RCP	1	0	1	0	0	0	1	1	0	0	2	8	C	1	1	(C) ← 0
	SCP	1	0	1	0	0	0	1	1	0	1	2	8	D	1	1	(C) ← 1
	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)
	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)
	TAPU2	1	0	0	1	0	1	1	1	1	1	2	5	F	1	1	(A) ← (PU2)
	TPU2A	1	0	0	0	1	0	1	1	1	1	2	2	F	1	1	(PU2) ← (A)
	TAPU3	1	0	0	1	0	1	1	1	0	1	2	5	D	1	1	(A) ← (PU3)
	TPU3A	1	0	0	0	0	0	1	0	0	0	2	0	8	1	1	(PU3) ← (A)

Skip condition	Carry flag CY	Detailed description
–	–	Transfers the input of port P0 to register A.
–	–	Outputs the contents of register A to port P0.
–	–	Transfers the input of port P1 to register A.
–	–	Outputs the contents of register A to port P1.
–	–	Transfers the input of port P2 to the register A.
–	–	Outputs the contents of the register A to port P2.
–	–	Transfers the input of port P3 to the register A.
–	–	Outputs the contents of the register A to port P3.
–	–	Sets (1) to port D.
–	–	Clears (0) to a bit of port D specified by register Y.
–	–	Sets (1) to a bit of port D specified by register Y.
(D(Y)) = 0 Y = 0 to 4	–	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”.
–	–	Clears (0) to port C.
–	–	Sets (1) to port C.
–	–	Transfers the contents of register A to port output structure control register FR0.
–	–	Transfers the contents of register A to port output structure control register FR1.
–	–	Transfers the contents of register A to port output structure control register FR2.
–	–	Transfers the contents of register A to port output structure control register FR3.
–	–	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.
–	–	Transfers the contents of pull-up control register PU2 to register A.
–	–	Transfers the contents of register A to pull-up control register PU2.
–	–	Transfers the contents of pull-up control register PU3 to register A.
–	–	Transfers the contents of register A to pull-up control register PU3.

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)
	TAK3	1	0	0	1	0	1	1	0	1	1	2	5	B	1	1	(A) ← (K3)
	TK3A	1	0	0	0	1	0	1	1	0	0	2	2	C	1	1	(K3) ← (A)
LCD operation	TAL1	1	0	0	1	0	0	1	0	1	0	2	4	A	1	1	(A) ← (L1)
	TL1A	1	0	0	0	0	0	1	0	1	0	2	0	A	1	1	(L1) ← (A)
	TL2A	1	0	0	0	0	0	1	0	1	1	2	0	B	1	1	(L2) ← (A)
	TL3A	1	0	0	0	0	0	1	1	0	0	2	0	C	1	1	(L3) ← (A)
	TC1A	1	0	1	0	1	0	1	0	0	0	2	A	8	1	1	(C1) ← (A)
	TC2A	1	0	1	0	1	0	1	0	0	1	2	A	9	1	1	(C2) ← (A)
	TC3A	1	0	0	0	1	0	0	1	1	0	2	2	6	1	1	(C3) ← (A)
Clock operation	CRCK	1	0	1	0	0	1	1	0	1	1	2	9	B	1	1	RC oscillator selected
	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)
	TRGA	1	0	0	0	0	0	1	0	0	1	2	0	9	1	1	(RG_2–RG_0) ← (A_2–A_0)

Skip condition	Carry flag CY	Detailed 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.
–	–	Transfers the contents of key-on wakeup control register K3 to register A.
–	–	Transfers the contents of register A to key-on wakeup control register K3.
–	–	Transfers the contents of the LCD control register L1 to register A.
–	–	Transfers the contents of register A to the LCD control register L1.
–	–	Transfers the contents of register A to the LCD control register L2.
–	–	Transfers the contents of register A to the LCD control register L3.
–	–	Transfers the contents of register A to the LCD control register C1.
–	–	Transfers the contents of register A to the LCD control register C2.
–	–	Transfers the contents of register A to the LCD control register C3.
–	–	Selects the RC oscillation circuit for main clock, stops the on-chip oscillator (internal oscillator).
–	–	Transfers the contents of clock control regiser MR to register A.
–	–	Transfers the contents of register A to clock control register MR.
–	–	Transfers the contents of register A to clock control register RG.

MACHINE INSTRUCTIONS (INDEX BY TYPES) (continued)

Parameter / Type of instructions	Mnemonic	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			Number of words	Number of cycles	Function
Other operation	NOP	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	(PC) ← (PC) + 1
	POF	0	0	0	0	0	0	0	0	1	0	0	0	2	1	1	Transition to clock operating mode
	POF2	0	0	0	0	0	0	1	0	0	0	0	0	8	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 or POF2 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 ? (WDF1) ← 0
	DWDT	1	0	1	0	0	1	1	1	0	0	2	9	C	1	1	Stop of watchdog timer function enabled
	SRST	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	System reset
	RUPT	0	0	0	1	0	1	1	0	0	0	0	5	8	1	1	(UPTF) ← 0
	SUPT	0	0	0	1	0	1	1	0	0	1	0	5	9	1	1	(UPTF) ← 1
	SVDE	1	0	1	0	0	1	0	0	1	1	2	9	3	1	1	At power down mode, voltage drop detection circuit valid
	SNZVD	1	0	1	0	0	0	1	0	1	0	2	8	A	1	1	(VDF) = 1?

Skip condition	Carry flag CY	Detailed description
–	–	No operation; Adds 1 to program counter value, and others remain unchanged.
–	–	Puts the system in clock operating mode by executing the POF instruction after executing the EPOF instruction.
–	–	Puts the system in RAM back-up state by executing the POF2 instruction after executing the EPOF instruction.
–	–	Makes the immediate after POF or POF2 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. Executes the next instruction when the P flag is “0”.
(WDF1) = 1		Clears (0) to the WDF1 flag and skips the next instruction when watchdog timer flag WDF1 is “1”. 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.
–	–	Stops the watchdog timer function by the WRST instruction after executing the DWDT instruction.
–	–	System reset occurs.
–	–	Clears (0) to the high-order bit reference enable flag UPTF.
–	–	Sets (1) to the high-order bit reference enable flag UPTF.
(VDF) = 1	–	Skips the next instruction when voltage drop detection circuit flag VDF is “1”. Execute instruction when VPF is “0”. After skipping, the contents of VDF remains unchanged.
–	–	Validates the voltage drop detection circuit at power down (clock operating mode and RAM back-up mode).

INSTRUCTION CODE TABLE

D_9–D_4		000000	000001	000010	000011	000100	000101	000110	000111	001000	001001	001010	001011	001100	001101	001110	001111	010000 to 010111	011000 to 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–F
0000	0	NOP	BLA	SZB 0	BMLA	–	TASP	A 0	LA 0	TABP 0	TABP 16	TABP 32	–	BML	BML	BL	BL	BM	B
0001	1	SRST	CLD	SZB 1	–	–	TAD	A 1	LA 1	TABP 1	TABP 17	TABP 33	–	BML	BML	BL	BL	BM	B
0010	2	POF	–	SZB 2	–	–	TAX	A 2	LA 2	TABP 2	TABP 18	TABP 34	–	BML	BML	BL	BL	BM	B
0011	3	SNZP	INY	SZB 3	–	–	TAZ	A 3	LA 3	TABP 3	TABP 19	TABP 35	–	BML	BML	BL	BL	BM	B
0100	4	DI	RD	SZD	–	RT	TAV1	A 4	LA 4	TABP 4	TABP 20	TABP 36	–	BML	BML	BL	BL	BM	B
0101	5	EI	SD	SEAn	–	RTS	TAV2	A 5	LA 5	TABP 5	TABP 21	TABP 37	–	BML	BML	BL	BL	BM	B
0110	6	RC	–	SEAM	–	RTI	–	A 6	LA 6	TABP 6	TABP 22	TABP 38	–	BML	BML	BL	BL	BM	B
0111	7	SC	DEY	–	–	–	–	A 7	LA 7	TABP 7	TABP 23	TABP 39	–	BML	BML	BL	BL	BM	B
1000	8	POF2	AND	–	SNZ0	LZ 0	RUPT	A 8	LA 8	TABP 8	TABP 24	TABP 40	–	BML	BML	BL	BL	BM	B
1001	9	–	OR	TDA	–	LZ 1	SUPT	A 9	LA 9	TABP 9	TABP 25	TABP 41	–	BML	BML	BL	BL	BM	B
1010	A	AM	TEAB	TABE	SNZI 0	LZ 2	–	A 10	LA 10	TABP 10	TABP 26	TABP 42	–	BML	BML	BL	BL	BM	B
1011	B	AMC	–	–	–	LZ 3	EPOF	A 11	LA 11	TABP 11	TABP 27	TABP 43	–	BML	BML	BL	BL	BM	B
1100	C	TYA	CMA	–	–	RB 0	SB 0	A 12	LA 12	TABP 12	TABP 28	TABP 44	–	BML	BML	BL	BL	BM	B
1101	D	–	RAR	–	–	RB 1	SB 1	A 13	LA 13	TABP 13	TABP 29	TABP 45	–	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	–	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	–	BML	BML	BL	BL	BM	B

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

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

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

INSTRUCTION CODE TABLE

D_9–D_4		100000	100001	100010	100011	100100	100101	100110	100111	101000	101001	101010	101011	101100	101101	101110	101111	110000 to 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	–	–	IAP0	TAB1	SNZT 1	–	WRST	TMA 0	TAM 0	XAM 0	XAMI 0	XAMD 0	LXY
0001	1	–	TW4A	OP1A	T2AB	–	–	IAP1	TAB2	SNZT 2	–	–	TMA 1	TAM 1	XAM 1	XAMI 1	XAMD 1	LXY
0010	2	–	–	OP2A	–	–	TAMR	IAP2	–	SNZT 3	–	–	TMA 2	TAM 2	XAM 2	XAMI 2	XAMD 2	LXY
0011	3	–	–	OP3A	–	–	TAI1	IAP3	–	–	SVDE	–	TMA 3	TAM 3	XAM 3	XAMI 3	XAMD 3	LXY
0100	4	–	TK1A	–	–	–	–	–	–	–	T2HA B	–	TMA 4	TAM 4	XAM 4	XAMI 4	XAMD 4	LXY
0101	5	–	TK2A	–	TPSAB	–	–	–	TABPS	–	T2R2 L	–	TMA 5	TAM 5	XAM 5	XAMI 5	XAMD 5	LXY
0110	6	–	TMRA	TC3A	–	–	TAK0	–	–	–	–	–	TMA 6	TAM 6	XAM 6	XAMI 6	XAMD 6	LXY
0111	7	–	TI1A	–	–	–	TAPU0	–	–	–	–	–	TMA 7	TAM 7	XAM 7	XAMI 7	XAMD 7	LXY
1000	8	TPU3A	–	TFR0A	–	–	–	–	–	–	–	TC1A	TMA 8	TAM 8	XAM 8	XAMI 8	XAMD 8	LXY
1001	9	TRGA	–	TFR1A	–	–	TAK1	–	–	–	–	TC2A	TMA 9	TAM 9	XAM 9	XAMI 9	XAMD 9	LXY
1010	A	TL1A	–	TFR2A	–	TAL1	TAK2	–	–	SNZV D	–	TPAA	TMA 10	TAM 10	XAM 10	XAMI 10	XAMD 10	LXY
1011	B	TL2A	TK0A	TFR3A	–	TAW1	TAK3	–	–	–	CRCK	–	TMA 11	TAM 11	XAM 11	XAMI 11	XAMD 11	LXY
1100	C	TL3A	–	TK3A	–	TAW2	–	–	–	RCP	DWDT	–	TMA 12	TAM 12	XAM 12	XAMI 12	XAMD 12	LXY
1101	D	TLCA	–	TPU0A	–	TAW3	TAPU3	–	–	SCP	–	–	TMA 13	TAM 13	XAM 13	XAMI 13	XAMD 13	LXY
1110	E	TW1A	–	TPU1A	–	TAW4	TAPU1	–	–	–	–	–	TMA 14	TAM 14	XAM 14	XAMI 14	XAMD 14	LXY
1111	F	TW2A	–	TPU2A	TR1AB	–	TAPU2	–	–	–	–	–	TMA 15	TAM 15	XAM 15	XAMI 15	XAMD 15	LXY

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

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

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

Electrical characteristics

Absolute maximum ratings

Table 30 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, D_0-D_5, $\overline{RESET}$, X_{IN}, X_{CIN}, INT, CNTR	-	–0.3 to V_{DD}+0.3	V
V_O	Output voltage P0, P1, P2, P3, D_0–D_7, $\overline{RESET}$	Output transistors in cut-off state	–0.3 to V_{DD}+0.3	V
V_O	Output voltage C/CNTR, X_{OUT}, X_{COUT}	-	–0.3 to V_{DD}+0.3	V
V_O	Output voltage SEG_0 to SEG_{31}, COM_0 to COM_3	-	–0.3 to V_{DD}+0.3	V
P_d	Power dissipation	Ta = 25 °C	300	mW
T_{opr}	Operating temperature range	-	–20 to 85	°C
T_{stg}	Storage temperature range	-	–40 to 125	°C

Recommended operating conditions

Table 31 Recommended operating conditions 1 (Ta = –20 °C to 85 °C, V_{DD} = 1.8 to 5.5 V, unless otherwise noted)

Symbol	Parameter	Conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{DD}	Supply voltage (with a ceramic resonator)	f(STCK) ≤ 6MHz		4		5.5	V
		f(STCK) ≤ 4.4MHz		2.7		5.5	
		f(STCK) ≤ 2.2MHz		2		5.5	
		f(STCK) ≤ 1.1MHz		1.8		5.5	
V_{DD}	Supply voltage (when an external clock is used)	f(STCK) ≤ 4.8MHz		4		5.5	V
		f(STCK) ≤ 3.2MHz		2.7		5.5	
		f(STCK) ≤ 1.6MHz		2		5.5	
		f(STCK) ≤ 0.8MHz		1.8		5.5	
V_{DD}	Supply voltage (when RC oscillation is used)	f(STCK) ≤ 4.4 MHz		2.7		5.5	V
V_{DD}	Supply voltage (when quartz-crystal oscillation is used)	f(STCK) ≤ 50 kHz		1.8		5.5	V
V_{DD}	Supply voltage (when on-chip oscillation is used)			1.8		5.5	V
V_{RAM}	RAM back-up voltage	(at RAM back-up)		1.6		5.5	V
V_{SS}	Supply voltage				0		V
V_{LC3}	LCD power supply (Note 1)			1.8		V_{DD}	V
V_{IH}	"H" level input voltage	P0, P1, P2, P3, D_0–D_5		$0.8V_{DD}$		V_{DD}	V
		X_{IN}, X_{CIN}		$0.7V_{DD}$		V_{DD}	
		$\overline{\text{RESET}}$		$0.85V_{DD}$		V_{DD}	
		INT		$0.85V_{DD}$		V_{DD}	
		CNTR		$0.8V_{DD}$		V_{DD}	
V_{IL}	"L" level input voltage	P0, P1, P2, P3, D_0–D_5		0		$0.2V_{DD}$	V
		X_{IN}, X_{CIN}		0		$0.3V_{DD}$	
		$\overline{\text{RESET}}$		0		$0.3V_{DD}$	
		INT		0		$0.15V_{DD}$	
		CNTR		0		$0.15V_{DD}$	
$I_{OH(peak)}$	"H" level peak output current	P0, P1, P2, P3, D_0–D_5	V_{DD} = 5V			–20	mA
			V_{DD} = 3V			–10	
		C/CNTR	V_{DD} = 5V			–30	
			V_{DD} = 3V			–15	
$I_{OH(avg)}$	"H" level average output current (Note 2)	P0, P1, P2, P3, D_0–D_5	V_{DD} = 5V			–10	mA
			V_{DD} = 3V			–5	
		C/CNTR	V_{DD} = 5V			–20	
			V_{DD} = 3V			–10	
$I_{OL(peak)}$	"L" level peak output current	P0, P1, P2, P3, D_0–D_7, C/CNTR	V_{DD} = 5V			24	mA
			V_{DD} = 3V			12	
		$\overline{\text{RESET}}$	V_{DD} = 5V			10	
			V_{DD} = 3V			4	
$I_{OL(avg)}$	"L" level average output current (Note 2)	P0, P1, P2, P3, D_0–D_7, C/CNTR	V_{DD} = 5V			15	mA
			V_{DD} = 3V			7	
		$\overline{\text{RESET}}$	V_{DD} = 5V			5	
			V_{DD} = 3V			2	
$\Sigma I_{OH(avg)}$	"H" level total average current	P0, C/CNTR				–40	mA
		P1, P2, P3, D_0–D_5				–40	
$\Sigma I_{OL(avg)}$	"L" level total average current	P0, C/CNTR				40	mA
		P1, P2, P3, D_0–D_7, $\overline{\text{RESET}}$				40	

Note 1. At 1/2 bias: $V_{LC1} = V_{LC2} = (1/2) \bullet V_{LC3}$
At 1/3 bias: $V_{LC1} = (1/3) \bullet V_{LC3}$, $V_{LC2} = (2/3) \bullet V_{LC3}$
Note 2. The average output current is the average value during 100ms.

Table 32 Recommended operating conditions 2 (Ta = –20 °C to 85 °C, V_{DD} = 1.8 to 5.5 V, unless otherwise noted)

Symbol	Parameter	Conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
$f(X_{IN})$	Oscillation frequency (with a ceramic resonator)	$f(STCK) = f(X_{IN})$	V_{DD} = 4.0 V to 5.5 V			6	MHz
			V_{DD} = 2.7 V to 5.5 V			4.4	
			V_{DD} = 2 V to 5.5 V			2.2	
			V_{DD} = 1.8 V to 5.5 V			1.1	
		$f(STCK) = f(X_{IN})/2$	V_{DD} = 2.7 V to 5.5 V			6	
			V_{DD} = 2 V to 5.5 V			4.4	
			V_{DD} = 1.8 V to 5.5 V			2.2	
		$f(STCK) = f(X_{IN})/4$, $f(X_{IN})/8$	V_{DD} = 2 V to 5.5 V			6	
			V_{DD} = 1.8 V to 5.5 V			4.4	
$f(X_{IN})$	Oscillation frequency (with an external clock input)	$f(STCK) = f(X_{IN})$	V_{DD} = 4 V to 5.5 V			4.8	MHz
			V_{DD} = 2.7 V to 5.5 V			3.2	
			V_{DD} = 2 V to 5.5 V			1.6	
			V_{DD} = 1.8 V to 5.5 V			0.8	
		$f(STCK) = f(X_{IN})/2$	V_{DD} = 2.7 V to 5.5 V			4.8	
			V_{DD} = 2 V to 5.5 V			3.2	
			V_{DD} = 1.8 V to 5.5 V			1.6	
		$f(STCK) = f(X_{IN})/4$, $f(X_{IN})/8$	V_{DD} = 2 V to 5.5 V			4.8	
			V_{DD} = 1.8 V to 5.5 V			3.2	
$f(X_{IN})$	Oscillation frequency (at RC oscillation) (Note 1)	V_{DD} = 2.7 to 5.5 V				4.4	MHz
$f(X_{CIN})$	Oscillation frequency (at quarts-crystal oscillation)	Quartz-crystal oscillator				50	kHz
f(CNTR)	Timer external input frequency	CNTR				f(STCK)/6	Hz
tw(CNTR)	Timer external input period ("H" and "L" pulse width)	CNTR		3/f(STCK)			s
T_{PON}	Power-on reset circuit valid supply voltage rising time (Note 2)	$V_{DD} = 0 \rightarrow 1.8V$				100	µs

Note 1. The frequency is affected by a capacitor, a resistor and a microcomputer. So, set the constants within the range of the frequency limits.

Note 2. If the rising time exceeds the maximum rating value, connect a capacitor between the $\overline{RESET}$ pin and Vss at the shortest distance, and input "L" level to $\overline{RESET}$ pin until the value of supply voltage reaches the minimum operating voltage.

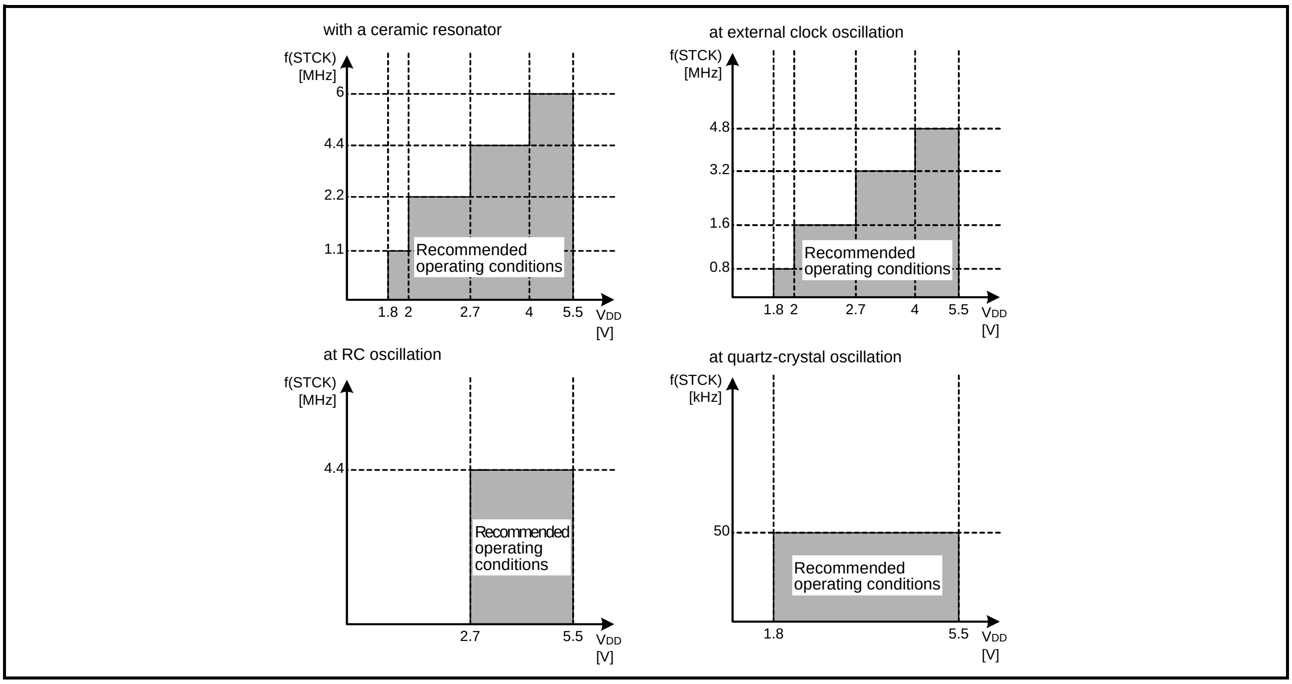


Fig 82. System clock (STCK) operating condition map

Electrical characteristics

Table 33 Electrical characteristics 1 (Ta = –20 °C to 85 °C, V_{DD} = 1.8 to 5.5 V, unless otherwise noted)

Symbol	Parameter		Test conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	“H” level output voltage	P0, P1, P2, P3, D_0–D_5	V_{DD} = 5V	I_{OH} = –10mA	3			V
				I_{OH} = –3mA	4.1			
			V_{DD} = 3V	I_{OH} = –5mA	2.1			
				I_{OH} = –1mA	2.4			
V_{OH}	“H” level output voltage	C/CNTR	V_{DD} = 5V	I_{OH} = –20mA	3			V
				I_{OH} = –6mA	4.1			
			V_{DD} = 3V	I_{OH} =–10mA	2.1			
				I_{OH} = –3mA	2.4			
V_{OL}	“L” level output voltage	P0, P1, P2, P3, D_0–D_7 C/CNTR	V_{DD} = 5V	I_{OL} = 15mA			2	V
				I_{OL} = 5mA			0.9	
			V_{DD} = 3V	I_{OL} = 9mA			1.4	
				I_{OL} = 3mA			0.9	
V_{OL}	“L” level output voltage	$\overline{\text{RESET}}$	V_{DD} = 5V	I_{OL} = 5mA			2	V
				I_{OL} = 1mA			0.6	
			V_{DD} = 3V	I_{OL} = 2mA			0.9	
I_{IH}	“H” level input current	P0, P1, P2, P3, D_0–D_5 $\overline{\text{RESET}}$, X_{IN}, X_{CIN}, INT CNTR	V_I = V_{DD}				2	μA
I_{IL}	“L” level input current	P0, P1, P2, P3, D_0–D_5 $\overline{\text{RESET}}$, X_{IN}, X_{CIN}, INT CNTR	V_I = 0V P0, P1, P2, P3 No pull-up				–2	μA
R_{PU}	Pull-up resistor value	P0, P1, P2, P3 $\overline{\text{RESET}}$	V_I = 0V	V_{DD} = 5V	30	60	125	kΩ
				V_{DD} = 3V	50	120	250	
V_{T+} –V_{T-}	Hysteresis	$\overline{\text{RESET}}$	V_{DD} = 5V			1		V
			V_{DD} = 3V			0.4		
V_{T+} –V_{T-}	Hysteresis	INT	V_{DD} = 5V			0.6		V
			V_{DD} = 3V			0.3		
V_{T+} –V_{T-}	Hysteresis	CNTR	V_{DD} = 5V			0.2		V
			V_{DD} = 3V			0.2		
f(RING)	On-chip oscillator clock frequency		V_{DD} = 5V		200	500	700	kHz
			V_{DD} = 3V		100	250	400	
Δf(X_{IN})	Frequency error (with RC oscillation, error of external RC not included) (Note 1)		V_{DD} = 5V ± 10 %, Ta = center 25 °C				± 17	%
			V_{DD} = 3V ± 10 %, Ta = center 25 °C				± 17	
R_{COM}	COM output impedance (Note 2)		V_{DD} = 5V			1.5	7.5	kΩ
			V_{DD} = 3V			2	10	
R_{SEG}	SEG output impedance (Note 2)		V_{DD} = 5V			1.5	7.5	kΩ
			V_{DD} = 3V			2	10	
R_{VLC}	Internal resistor for LCD power supply		When dividing resistor 2r × 3 selected		300	600	1200	kΩ
			When dividing resistor 2r × 2 selected		200	400	800	
			When dividing resistor r × 3 selected		150	300	600	
			When dividing resistor r × 2 selected		100	200	400	

Note 1. When RC oscillation is used, use the external 33 pF capacitor (C).
Note 2. The impedance state is the resistor value of the output voltage.
at V_{LC3} level output: V_O = 0.8 V_{LC3}
at V_{LC2} level output: V_O = 0.8 V_{LC2}
at V_{LC1} level output: V_O = 0.2 V_{LC2} + V_{LC1}
at V_{SS} level output: V_O = 0.2 V_{LC1}

Table 34 Electrical characteristics 2 (Ta = –20 °C to 85 °C, V_{DD} = 1.8 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 oscillator) (1, 2)	V_{DD} = 5V f(X_{IN}) = 6MHz f(RING) = stop f(X_{CIN}) = stop	f(STCK) = f(X_{IN})/8		1.2	2.4	mA
				f(STCK) = f(X_{IN})/4		1.3	2.6	
				f(STCK) = f(X_{IN})/2		1.6	3.2	
				f(STCK) = f(X_{IN})		2.2	4.4	
			V_{DD} = 5V f(X_{IN}) = 4MHz f(RING) = stop f(X_{CIN}) = stop	f(STCK) = f(X_{IN})/8		0.9	1.8	mA
				f(STCK) = f(X_{IN})/4		1	2	
				f(STCK) = f(X_{IN})/2		1.2	2.4	
				f(STCK) = f(X_{IN})		1.6	3.2	
			V_{DD} = 3V f(X_{IN}) = 4MHz f(RING) = stop f(X_{CIN}) = stop	f(STCK) = f(X_{IN})/8		0.3	0.6	mA
				f(STCK) = f(X_{IN})/4		0.4	0.8	
				f(STCK) = f(X_{IN})/2		0.5	1	
				f(STCK) = f(X_{IN})		0.7	1.4	
		at active mode (with a quartz-crystal oscillator)(1, 2)	V_{DD} = 5V f(X_{IN}) = stop f(RING) = stop f(X_{CIN}) = 32 kHz	f(STCK) = f(X_{CIN})/8		7	14	µA
				f(STCK) = f(X_{CIN})/4		8	16	
				f(STCK) = f(X_{CIN})/2		10	20	
				f(STCK) = f(X_{CIN})		14	28	
			V_{DD} = 3V f(X_{IN}) = stop f(RING) = stop f(X_{CIN}) = 32 kHz	f(STCK) = f(X_{CIN})/8		5	10	µA
				f(STCK) = f(X_{CIN})/4		6	12	
				f(STCK) = f(X_{CIN})/2		7	14	
				f(STCK) = f(X_{CIN})		8	16	
		at active mode (with an on-chip oscillator) (1, 2)	V_{DD} = 5V f(X_{IN}) = stop f(RING) = active f(X_{CIN}) = stop	f(STCK) = f(RING)/8		50	100	µA
				f(STCK) = f(RING)/4		60	120	
				f(STCK) = f(RING)/2		80	160	
				f(STCK) = f(RING)		120	240	
			V_{DD} = 3V f(X_{IN}) = stop f(RING) = active f(X_{CIN}) = stop	f(STCK) = f(RING)/8		10	20	µA
				f(STCK) = f(RING)/4		13	26	
				f(STCK) = f(RING)/2		19	38	
				f(STCK) = f(RING)		31	62	
		at clock operation mode (POF instruction execution)(1, 2)	f(X_{CIN}) = 32 kHz	V_{DD} = 5V		6	12	µA
				V_{DD} = 3V		5	10	
		at RAM back-up mode (POF2 instruction execution)(1)	Ta = 25°C			0.1	3	µA
			V_{DD} = 5V				10	
			V_{DD} = 3V				6	

Note 1. The voltage drop detection circuit operation current (I_{RST}) is added.
Note 2. When the internal dividing resistors for LCD power are used, the current values according to using resistor values are added.

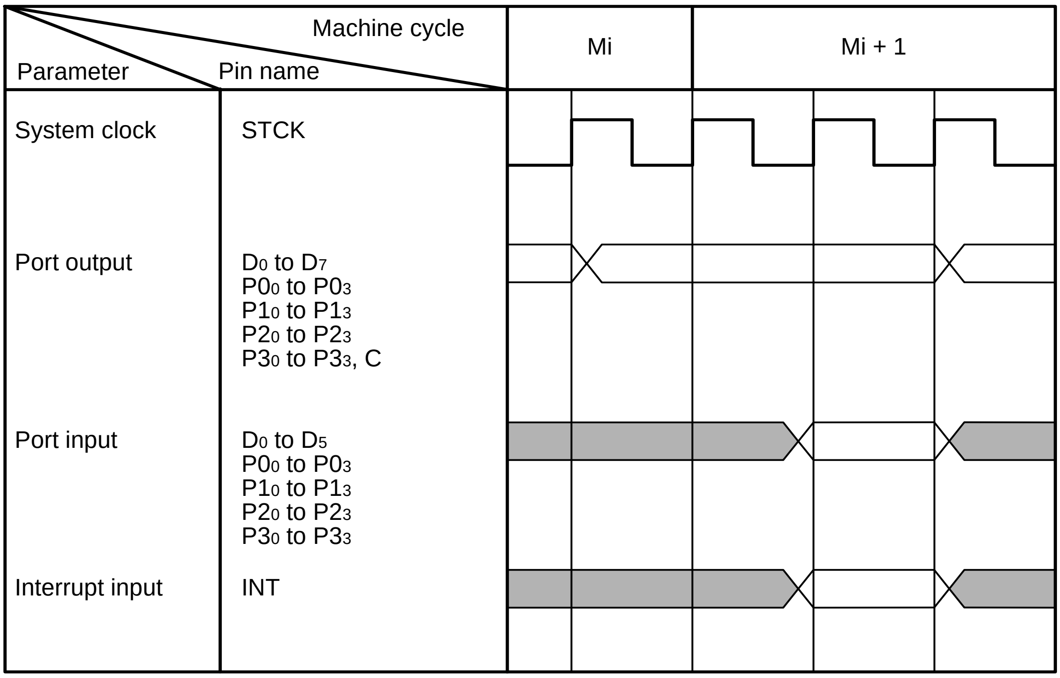
Voltage drop detection circuit characteristics

Table 35 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		1.7		V
		–20°C≤ Ta < 0°C	1.6		2.2	
		0°C≤ Ta < 50°C	1.3		2.1	
		50°C≤ Ta ≤ 85°C	1.1		1.8	
V_{RST+}	Detection voltage (reset release) (Note 2)	Ta = 25°C		1.8		V
		–20°C≤ Ta < 0°C	1.7		2.3	
		0°C≤ Ta < 50°C	1.4		2.2	
		50°C≤ Ta ≤ 85°C	1.2		1.9	
V_{SKIP}	Detection voltage (skip occurs) (Note 3)	Ta = 25°C		2		V
		–20°C≤ Ta < 0°C	1.9		2.5	
		0°C≤ Ta < 50°C	1.6		2.4	
		50°C≤ Ta ≤ 85°C	1.4		2.1	
$V_{RST+} - V_{RST-}$	Detection voltage hysteresis			0.1		V
I_{RST}	Operation current (Note 4)	V_{DD} = 5V		30	60	µA
		V_{DD} = 3V		15	30	
		V_{DD} = 1.8V		6	12	
T_{RST}	Detection time (Note 5)	$V_{DD} \rightarrow (V_{RST-} - 0.1V)$		0.2	1.2	ms

Note 1. The detection voltage (V_{RST-}) is defined as the voltage when reset occurs when the supply voltage (V_{DD}) is falling.
Note 2. The detection voltage (V_{RST+}) is defined as the voltage when reset is released when the supply voltage (V_{DD}) is rising from reset occurs.
Note 3. When the supply voltage goes lower than the detection voltage (V_{SKIP}), the voltage drop detection circuit interrupt request flag (VDF) is set to "1".
Note 4. Voltage drop detection circuit operation current (I_{RST}) is added to I_{DD} (power current) when voltage drop detection circuit is used.
Note 5. The detection time (T_{RST}) is defined as the time until reset occurs when the supply voltage (V_{DD}) is falling to [V_{RST-} –0.1V].

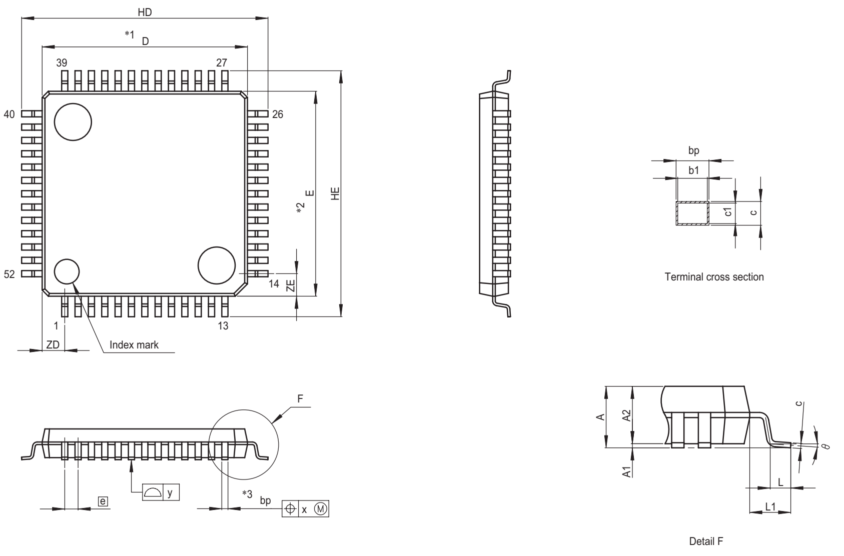
Basic timing diagram



PACKAGE OUTLINE

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP52-10x10-0.65	PLQP0052JA-A	52P6A-A	0.3g

Under development



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.27	0.32	0.37
b_1	—	0.30	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
$\boxed{e}$	—	0.65	—
x	—	—	0.13
y	—	—	0.10
Z_D	—	1.1	—
Z_E	—	1.1	—
L	0.35	0.5	0.65
L_1	—	1.0	—

REVISION HISTORY	4559 Group Datasheet

Rev.	Date	Description	
		Page	Summary
1.00	Jul 27, 2006	-	First edition issued
1.01	Apr 27, 2007	58	Fig56 stabilizing time b, d: (system clock division ratio × <u>15</u>) times. → (system clock division ratio × <u>171</u>) times.
1.02	May 25, 2007	All pages	“PRELIMINARY” deleted
1.03	May 30, 2007	32	Fig33 ORCLK ——→ ORCLK —$\boxed{1/2}$—
		33	Fig34 $W3_0$ → $W3_3$
		34,74	$W3_3$ Timer 3 count source selection bit 1 : Prescaler output (ORCLK)/2 → Prescaler output (ORCLK)
1.04	Aug 23, 2007	4	Timer 1, Timer 2 Explanation of function revised. Segment output “28”→ “32”
		21	Fig. 21 $13FF_{16}$→ $17FF_{16}$
		25	(7)Interrupt sequence revised.
		34	PA0 0 “Stop (state initialized)”→ “Stop (state retained)” W30, W31 “Timer 3 count source selection bits” →“Timer 3 count value selection bits” $W3_0$ 0 “X_{IN} input”→ “X_{CIN} input”
		55	Table 23: Note 4 is revised.
		57	Fig. 56 Note 7 added.
		65, 66, 67	QzROM Writing Mode added.
		69	(2) Bit 3 of register I1 “(register L10=“0”)” →“(register K20=“0”)” (3) Bit 2 of register I1 “the external 1 interrupt request flag (EXF0)” →“the external 0 interrupt request flag (EXF0)”
		71	(27) Data Required for QzROM Writing Orders added.
		72	Fig. 76 Note added.
		73	Fig. 77 “V_{CC}”→”V_{DD}”
		77	PA_0 Prascaler control bit 0 “Stop (state initialized)”→ “Stop (state retained)” $W3_0$, $W3_1$ “Timer 3 count source selection bits” →“Timer 3 count value selection bits”
		84, 85, 86	Index pages added.
		109	TAW4 Operation: “(A) ← (W5)”→“(A) ← (W4)”

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

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 (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

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 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