

MITSUBISHI 4-BIT SINGLE-CHIP MICROCOMPUTER  
4500 SERIES

4551  
Group

User's Manual



MITSUBISHI  
ELECTRIC

— keep safety first in your circuit designs ! —

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

— Notes regarding these materials —

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

## **Preface**

This user's manual describes the hardware and instructions of Mitsubishi's 4551 Group CMOS 4-bit microcomputer.

After reading this manual, the user should have a thorough knowledge of the functions and features of the 4551 Group and should be able to fully utilize the product. The manual starts with specifications and ends with application examples.

## **BEFORE USING THIS USER'S MANUAL**

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

### **1. Organization**

- **CHAPTER 1 HARDWARE**

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

- **CHAPTER 2 APPLICATION**

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

- **CHAPTER 3 APPENDIX**

This chapter includes precautions for systems development using the microcomputer, the mask ROM confirmation forms (mask ROM version), ROM programming confirmation forms (One Time PROM version) and mark specification forms which are to be submitted when ordering.

Be sure to refer to this chapter because this chapter also includes necessary information for systems development.

# Table of contents

## CHAPTER 1 HARDWARE

DESCRIPTION .....	1-3
FEATURES .....	1-3
PIN CONFIGURATION (TOP VIEW) .....	1-3
APPLICATION .....	1-3
BLOCK DIAGRAM .....	1-4
PERFORMANCE OVERVIEW .....	1-5
DEFINITION OF CLOCK AND CYCLE .....	1-5
PIN DESCRIPTION .....	1-6
MULTIFUNCTION .....	1-7
CONNECTIONS OF UNUSED PINS .....	1-7
PORT FUNCTION .....	1-7
PORT BLOCK DIAGRAMS .....	1-8
FUNCTIONAL BLOCK OPERATIONS .....	1-10
CPU .....	1-10
PROGRAM MEMORY (ROM) .....	1-13
DATA MEMORY (RAM) .....	1-14
INTERRUPT FUNCTION .....	1-15
EXTERNAL INTERRUPTS .....	1-18
TIMERS .....	1-20
WATCHDOG TIMER .....	1-24
CARRIER GENERATING CIRCUIT .....	1-25
LCD FUNCTION .....	1-28
RESET FUNCTION .....	1-32
VOLTAGE DROP DETECTION CIRCUIT .....	1-34
POWER DOWN FUNCTION .....	1-35
CLOCK CONTROL .....	1-38
ROM ORDERING METHOD .....	1-39
LIST OF PRECAUTIONS .....	1-40
SYMBOL .....	1-41
LIST OF INSTRUCTION FUNCTION .....	1-42
INSTRUCTION CODE TABLE .....	1-44
MACHINE INSTRUCTIONS .....	1-46
CONTROL REGISTERS .....	1-60
BUILT-IN PROM VERSION .....	1-64
PIN CONFIGURATION (TOP VIEW) .....	1-64

**CHAPTER 2 APPLICATION**

<b>2.1 I/O pins .....</b>	<b>2-2</b>
2.1.1 I/O ports .....	2-2
2.1.2 Related registers .....	2-4
2.1.3 Port application examples .....	2-6
2.1.4 Notes on use .....	2-8
<b>2.2 Interrupts .....</b>	<b>2-9</b>
2.2.1 Interrupt functions .....	2-9
2.2.2 Related registers .....	2-10
2.2.3 Interrupt application examples .....	2-12
2.2.4 Notes on use .....	2-16
<b>2.3 Timers .....</b>	<b>2-17</b>
2.3.1 Timer functions .....	2-17
2.3.2 Related registers .....	2-17
2.3.3 Timer application examples .....	2-20
2.3.4 Notes on use .....	2-24
<b>2.4 Carrier generating circuit .....</b>	<b>2-25</b>
2.4.1 Carrier functions .....	2-25
2.4.2 Related registers .....	2-26
2.4.3 Carrier wave output application examples .....	2-28
2.4.4 Notes on use .....	2-32
<b>2.5 LCD function .....</b>	<b>2-33</b>
2.5.1 Operation description .....	2-33
2.5.2 Related registers .....	2-34
2.5.3 LCD application examples .....	2-36
2.5.4 Notes on use .....	2-38
<b>2.6 Power down function .....</b>	<b>2-39</b>
2.6.1 Clock control function .....	2-41
2.6.2 Power down function .....	2-41
2.6.3 Related register .....	2-44
2.6.4 Power down function application example .....	2-45
2.6.5 Notes on use .....	2-45
<b>2.7 Reset .....</b>	<b>2-46</b>
2.7.1 Reset circuit .....	2-46
2.7.2 Internal state at reset .....	2-48
2.7.3 Voltage drop detection circuit .....	2-49
<b>2.8 Oscillation circuit .....</b>	<b>2-50</b>
2.8.1 Oscillation circuit .....	2-50
2.8.2 Oscillation operation .....	2-51
2.8.3 Notes on use .....	2-51

---

**CHAPTER 3 APPENDIX**

<b>3.1 Electrical characteristics .....</b>	<b>3-2</b>
3.1.1 Absolute maximum ratings .....	3-2
3.1.2 Recommended operating conditions .....	3-3
3.1.3 Electrical characteristics .....	3-4
3.1.4 Basic timing diagram .....	3-5
<b>3.2 Typical characteristics .....</b>	<b>3-6</b>
3.2.1 VDD–IDD characteristics .....	3-6
3.2.2 VOH–IOH characteristics (port CARR) .....	3-9
3.2.3 VOL–IOL characteristics (Ports P0, P1, D0–D7, CARR, $\overline{\text{RESET}}$ ) .....	3-10
3.2.4 Voltage drop detection circuit temperature characteristics .....	3-11
<b>3.3 List of precautions .....</b>	<b>3-12</b>
<b>3.4 Notes on noise .....</b>	<b>3-13</b>
3.4.1 Shortest wiring length .....	3-13
3.4.2 Connection of bypass capacitor across Vss line and Vcc line .....	3-15
3.4.3 Oscillator concerns .....	3-16
3.4.4 setup for I/O ports .....	3-17
3.4.5 providing of watchdog timer function by software .....	3-17
<b>3.5 Mask ROM order confirmation form .....</b>	<b>3-18</b>
<b>3.6 ROM programming order confirmation form .....</b>	<b>3-20</b>
<b>3.7 Mark specification form .....</b>	<b>3-21</b>
<b>3.8 Package outline .....</b>	<b>3-22</b>

# List of figures

## CHAPTER 1 HARDWARE

---

Fig. 1 AMC instruction execution example .....	1-10
Fig. 2 RAR instruction execution example .....	1-10
Fig. 3 Registers A, B and register E .....	1-10
Fig. 4 TABP p instruction execution example .....	1-10
Fig. 5 Stack registers (SKs) structure .....	1-11
Fig. 6 Example of operation at subroutine call .....	1-11
Fig. 7 Program counter (PC) structure .....	1-12
Fig. 8 Data pointer (DP) structure .....	1-12
Fig. 9 SD instruction execution example .....	1-12
Fig. 10 ROM map of M34551E8 .....	1-13
Fig. 11 Interrupt address page (addresses 0080 <sub>16</sub> to 00FF <sub>16</sub> ) structure .....	1-13
Fig. 12 RAM map .....	1-14
Fig. 13 Program example of interrupt processing .....	1-16
Fig. 14 Internal state when interrupt occurs .....	1-16
Fig. 15 Interrupt system diagram .....	1-16
Fig. 16 Interrupt sequence .....	1-17
Fig. 17 External interrupt circuit structure .....	1-18
Fig. 18 Auto-reload function .....	1-20
Fig. 19 Timers structure .....	1-21
Fig. 20 Watchdog timer function .....	1-24
Fig. 21 Program example to enter the RAM back-up mode when using the watchdog timer ....	1-24
Fig. 22 Carrier wave selection register .....	1-25
Fig. 23 Carrier wave output auto-control by timer 1 .....	1-27
Fig. 24 LCD clock control circuit structure .....	1-28
Fig. 25 LCD controller/driver structure .....	1-29
Fig. 26 LCD RAM map .....	1-30
Fig. 27 LCD controller/driver structure .....	1-31
Fig. 28 Reset release timing .....	1-32
Fig. 29 $\overline{\text{RESET}}$ pin input waveform and reset operation .....	1-32
Fig. 30 Power-on reset circuit example .....	1-32
Fig. 31 Internal state at reset .....	1-33
Fig. 32 Voltage drop detection reset circuit .....	1-34
Fig. 33 Voltage drop detection circuit operation waveform .....	1-34
Fig. 34 Set source and clear source of the P flag .....	1-36
Fig. 35 Start condition identified example using the SNZP instruction .....	1-36
Fig. 36 State transition .....	1-37
Fig. 37 Clock control circuit structure .....	1-38
Fig. 38 Ceramic resonator external circuit .....	1-39
Fig. 39 Quartz-crystal oscillator external circuit .....	1-39
Fig. 40 External 0 interrupt program example .....	1-40
Fig. 41 Pin configuration of built-in PROM version .....	1-64
Fig. 42 PROM memory map .....	1-65
Fig. 43 Flow of writing and test of the product shipped in blank .....	1-65



**CHAPTER 2 APPLICATION**

Fig. 2.1.1 Key input by key scan .....	2-6
Fig. 2.1.2 Input timing of key scan .....	2-7
Fig. 2.2.1 External interrupt operation example .....	2-12
Fig. 2.2.2 External interrupt setting example .....	2-13
Fig. 2.2.3 Timer 1 constant period interrupt setting example .....	2-14
Fig. 2.2.4 Timer 2 constant period interrupt setting example .....	2-15
Fig. 2.3.1 Watchdog timer function .....	2-20
Fig. 2.3.2 Constant period measurement setting example .....	2-21
Fig. 2.3.3 Constant period counter by timer 2 setting example .....	2-22
Fig. 2.3.4 Watchdog timer setting example .....	2-23
Fig. 2.4.1 Carrier wave selection register .....	2-26
Fig. 2.4.2 Carrier wave auto-control setting example 1 .....	2-28
Fig. 2.4.3 Carrier wave auto-control setting example 2 .....	2-29
Fig. 2.4.4 Carrier wave output interval setting example .....	2-30
Fig. 2.4.5 Carrier wave by software generating example .....	2-31
Fig. 2.5.1 LCD clock control circuit structure .....	2-33
Fig. 2.5.2 LCD RAM map .....	2-34
Fig. 2.5.3 LCD display panel example .....	2-36
Fig. 2.5.4 Segment assignment example .....	2-36
Fig. 2.5.5 LCD RAM assignment example .....	2-36
Fig. 2.5.6 Initial setting example .....	2-37
Fig. 2.6.1 State transition .....	2-39
Fig. 2.6.2 Oscillation stabilizing time in each mode .....	2-40
Fig. 2.6.3 Start condition identified example .....	2-43
Fig. 2.6.4 Software setting example .....	2-45
Fig. 2.7.1 Power-on reset circuit example .....	2-46
Fig. 2.7.2 Reset circuit example when the supply voltage rising time exceeds 100 ms ..	2-47
Fig. 2.7.3 Oscillation stabilizing time after system is released from reset .....	2-47
Fig. 2.7.4 Internal state at reset .....	2-48
Fig. 2.7.5 Voltage drop detection reset circuit .....	2-49
Fig. 2.7.6 Voltage drop detection circuit operation waveform .....	2-49
Fig. 2.8.1 Oscillation circuit example connecting ceramic resonator externally .....	2-50
Fig. 2.8.2 Oscillation circuit example connecting quartz-crystal externally .....	2-50
Fig. 2.8.3 Structure of clock control circuit .....	2-51

**CHAPTER 3 APPENDIX**

Fig. 40 External 0 interrupt program example .....	3-12
Fig. 3.4.1 Selection of packages .....	3-13
Fig. 3.4.2 Wiring for the $\overline{\text{RESET}}$ input pin .....	3-13
Fig. 3.4.3 Wiring for clock I/O pins .....	3-14
Fig. 3.4.4 Wiring for CNVss pin .....	3-14
Fig. 3.4.5 Wiring for the VPP pin of the One Time PROM version .....	3-15
Fig. 3.4.6 Bypass capacitor across the VSS line and the VCC line .....	3-15
Fig. 3.4.7 Wiring for a large current signal line .....	3-16
Fig. 3.4.8 Wiring to a signal line where potential levels change frequently .....	3-16
Fig. 3.4.9 Vss pattern on the underside of an oscillator .....	3-16
Fig. 3.4.10 Watchdog timer by software .....	3-17

# List of tables

## CHAPTER 1 HARDWARE

Table Selection of system clock .....	1-5
Table 1 ROM size and pages .....	1-13
Table 2 RAM size .....	1-14
Table 3 Interrupt sources .....	1-15
Table 4 Interrupt request flag, interrupt enable bit and skip instruction .....	1-15
Table 5 Interrupt enable bit function .....	1-15
Table 6 Interrupt control register .....	1-17
Table 7 External interrupt activated condition .....	1-18
Table 8 External interrupt control register .....	1-19
Table 9 Function related timers .....	1-20
Table 10 Timer control registers .....	1-22
Table 11 Carrier generating circuit control register and control flag .....	1-26
Table 12 Duty and maximum number of displayed pixels .....	1-28
Table 13 LCD control registers .....	1-30
Table 14 Port state at reset .....	1-33
Table 15 Functions and states retained at power down .....	1-35
Table 16 Return source and return condition .....	1-36
Table 17 Pull-up control register .....	1-36
Table 18 Clock control register .....	1-38
Table 19 Product of built-in PROM version .....	1-64
Table 20 Programming adapter .....	1-65

## CHAPTER 2 APPLICATION

Table 2.1.1 Pull-up control register PU0 .....	2-4
Table 2.1.2 LCD control register L2 .....	2-4
Table 2.1.3 Clock control register MR .....	2-5
Table 2.1.4 connections of unused pins .....	2-8
Table 2.2.1 Interrupt control register V1 .....	2-10
Table 2.2.2 Interrupt control register I1 .....	2-11
Table 2.3.1 Interrupt control register V1 .....	2-17
Table 2.3.3 Timer control register W2 .....	2-18
Table 2.3.2 Timer control register W1 .....	2-18
Table 2.3.4 Timer control register W3 .....	2-19
Table 2.4.1 Carrier wave output control register C2 .....	2-27
Table 2.4.2 Carrier wave generating control flag CR .....	2-27
Table 2.5.1 Duty and maximum number of displayed pixels .....	2-33
Table 2.5.2 LCD control register L1 .....	2-34
Table 2.5.3 LCD control register L2 .....	2-35
Table 2.5.4 Timer control register W3 .....	2-35
Table 2.5.5 Frame frequency .....	2-38
Table 2.6.1 Functions and states retained at RAM back-up mode and the clock operating mode ....	2-42
Table 2.6.2 Return source and return condition .....	2-43
Table 2.6.3 Start condition identification .....	2-43
Table 2.6.4 Clock control register MR .....	2-44
Table 2.6.5 Pull-up control register PU0 .....	2-44
Table 2.8.1 Maximum value of oscillation frequency and supply voltage .....	2-50

**CHAPTER 3 APPENDIX**

---

Table 3.1.1 Absolute maximum ratings .....	3-2
Table 3.1.2 Recommended operating conditions .....	3-3
Table 3.1.3 Electrical characteristics .....	3-4



# CHAPTER 1

## **HARDWARE**

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

# HARDWARE

---

## *MEMO*

### DESCRIPTION

The 4551 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 an 8-bit timer with a reload register, a 14-bit timer which is also used as a watchdog timer, a 4-bit timer with a reload register, a carrier wave output circuit and an LCD control circuit.

The mask ROM version and built-in PROM version of 4551 Group are produced as shown in the table below.

### FEATURES

- Minimum instruction execution time ..... 1.5  $\mu$ s  
( $f(X_{IN})=8.0$  MHz,  $V_{DD}=5.0$  V, system clock =  $f(X_{IN})/4$ )
- Supply voltage  
..... 2.5 V to 5.5 V (One Time PROM version)  
..... 2.2 V to 5.5 V (Mask ROM version)
- System clock switch function  
..... Clock divided by 4 or not divided

- LCD control circuit  
Segment output ..... 20  
Common output ..... 4
- Carrier wave frequency switch function  
System clock, system clock/2, system clock/8,  
system clock/12, system clock/16, system clock/24, "H" fixed
- Timers  
Timer 1 ..... 8-bit timer with a reload register  
Timer 2 ..... 14-bit timer also used as a watchdog timer  
Timer LC ..... 4-bit timer with a reload register
- Interrupt ..... 3 sources
- Voltage drop detection circuit ..... 1
- Clock generating circuit (ceramic resonance and quartz-crystal oscillation)

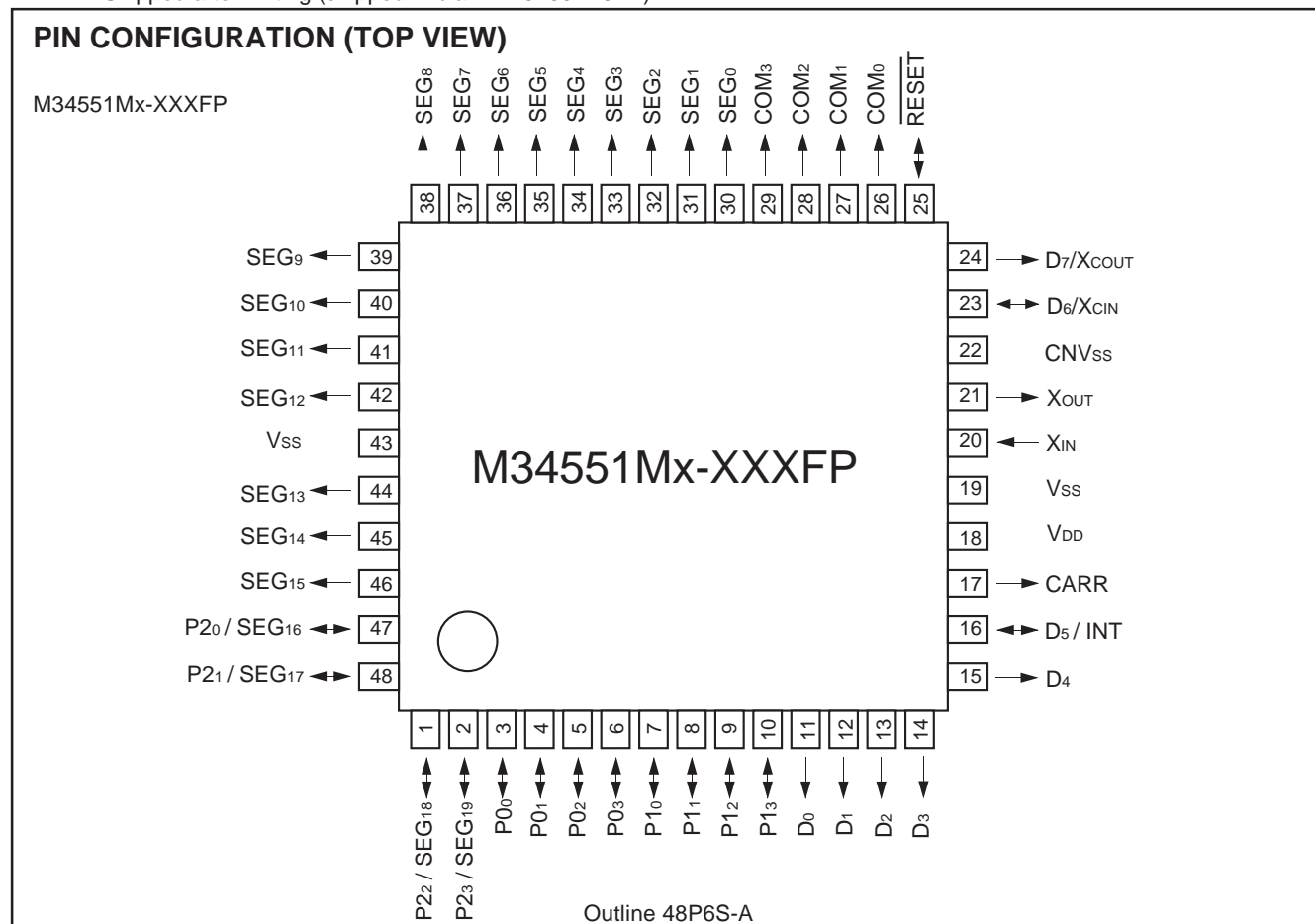
### APPLICATION

Remote control transmitter

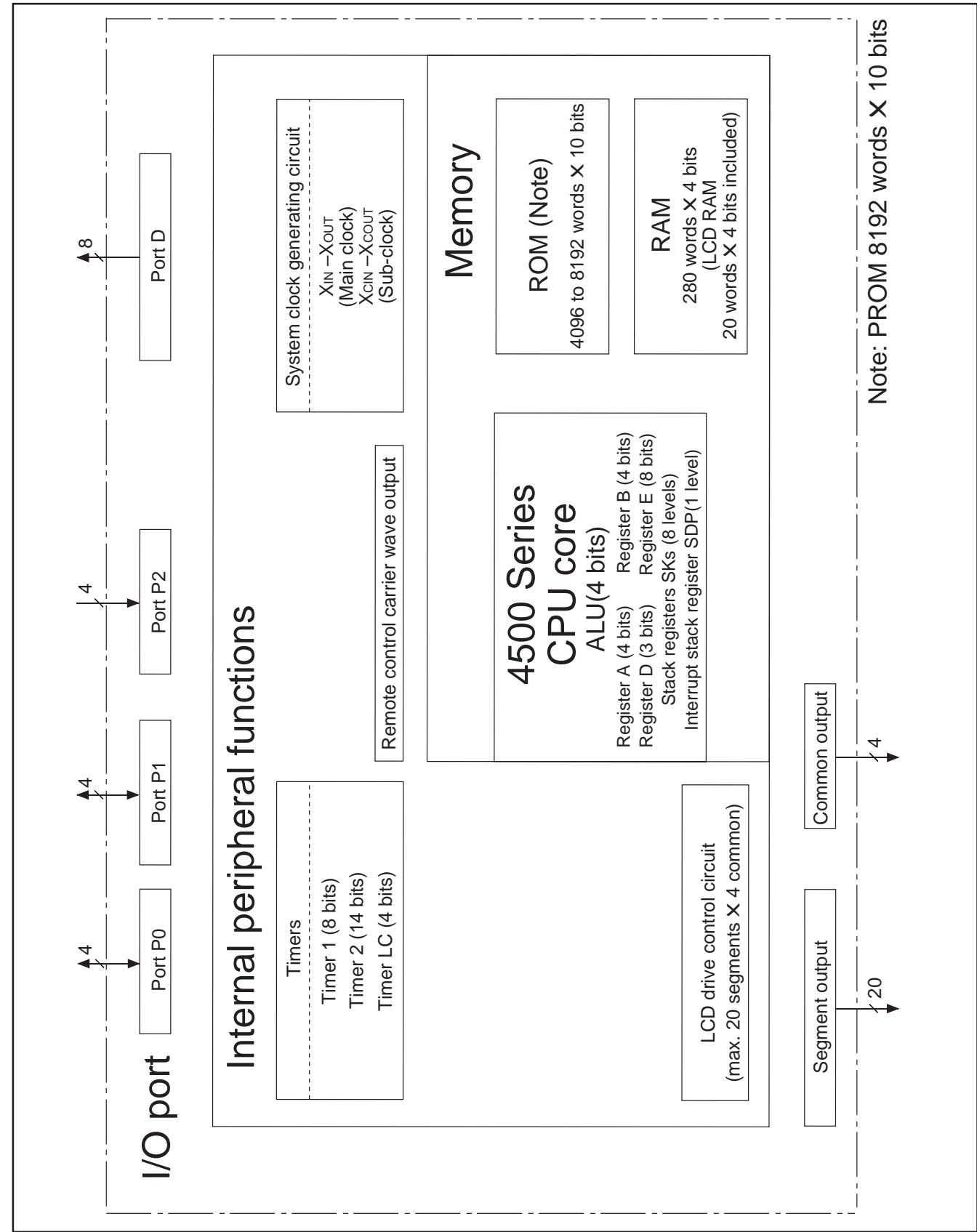
Product	ROM (PROM) size (X 10 bits)	RAM size (X 4 bits)	Package	ROM type
M34551M4-XXXXFP	4096 words	280 words	48P6S-A	Mask ROM
M34551M8-XXXXFP (Note 1)	8192 words	280 words	48P6S-A	Mask ROM
M34551E8-XXXXFP (Note 2)	8192 words	280 words	48P6S-A	One Time PROM

Notes 1: Under development (Aug. 1998)

2: Shipped after writing (shipped in blank: M34551E8FP)



BLOCK DIAGRAM



## PERFORMANCE OVERVIEW

Parameter			Function
Number of basic instructions			92
Minimum instruction execution time			1.5 $\mu$ s ( $f(X_{IN}) = 8.0$ MHz:system clock = $f(X_{IN})/4$ ; $V_{DD} = 5.0$ V)
Memory sizes	ROM	M34551M4	4096 words $\times$ 10 bits
		M34551M8	8192 words $\times$ 10 bits
		M34551E8	8192 words $\times$ 10 bits
	RAM		280 words $\times$ 4 bits (LCD RAM 20 words $\times$ 4 bits included)
Input/Output ports	D0–D7	Output	Eight independent output ports
	P00–P03	I/O	4-bit I/O port; each pin is equipped with a pull-up function.
	P10–P13	I/O	4-bit I/O port; each pin is equipped with a pull-up function.
	P20–P23	Input	4-bit input port
	CARR	Output	1-bit output port (CMOS output)
Timers	Timer 1		8-bit timer with a reload register
	Timer 2/ Watchdog timer		14-bit timer/ Fixed dividing frequency timer
	Timer LC		4-bit timer with a reload register
Interrupt	Sources		3 (one for external and two for timer)
	Nesting		1 level
Subroutine nesting			8 levels (however, only 7 levels can be used when an interrupt is used or the TABP p instruction is executed)
LCD	Selective bias value		1/2, 1/3 bias
	Selective duty value		2, 3, 4 duty
	Common output		4
	Segment output		20
	Internal resistor for power supply		200 k $\Omega$ $\times$ 3
Device structure			CMOS silicon gate
Package			48-pin plastic molded QFP
Operating temperature range			–20 $^{\circ}$ C to 70 $^{\circ}$ C
Supply voltage			2.2 V to 5.5 V (One Time PROM version: 2.5 V to 5.5 V)
Power dissipation (typical value)	at active		2.5 mA ( $f(X_{IN}) = 8.0$ MHz system clock = $f(X_{IN})/4$ , $V_{DD}$ =5 V)
	at clock operating		27.5 $\mu$ A (at main clock oscillation stop, sub-clock oscillation frequency: 32.0 kHz, $T_a$ =25 $^{\circ}$ C, $V_{DD}$ =5 V)
	at RAM back-up		0.1 $\mu$ A (at main clock oscillation stop, sub-clock oscillation stop, $T_a$ =25 $^{\circ}$ C, $V_{DD}$ =5V)

## DEFINITION OF CLOCK AND CYCLE

## ● System clock (STCK)

The system clock is the basic clock for controlling this product.  
The system clock can be selected by bits 0 and 3 of the clock control register MR as shown in the table below.

Table Selection of system clock

Register MR		System clock (STCK)
MR <sub>3</sub>	MR <sub>0</sub>	
0	0	$f(X_{IN})$
0	1	$f(X_{CIN})$
1	0	$f(X_{IN})/4$
1	1	$f(X_{CIN})/4$

Note:  $f(X_{IN})/4$  is selected immediately after system is released from reset.

## ● Instruction clock (INSTK)

The instruction clock is the standard clock for controlling CPU.  
The instruction clock is a signal derived from dividing the system clock by 3. The one cycle of the instruction clock is equivalent to the one machine cycle.

## ● Machine cycle

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



# HARDWARE

## PIN DESCRIPTION

### PIN DESCRIPTION

Pin	Name	Input/Output	Function
V <sub>DD</sub>	Power supply	—	Connected to a plus power supply.
V <sub>SS</sub>	Ground	—	Connected to a 0 V power supply.
CNV <sub>SS</sub>	CNV <sub>SS</sub>	Input	Connect CNV <sub>SS</sub> to V <sub>SS</sub> and apply “L” (0V) to CNV <sub>SS</sub> certainly.
RESET	Reset input	I/O	An N-channel open-drain I/O pin for a system reset. A pull-up resistor is built-in this pin. When the watchdog timer causes the system to be reset or the low-supply voltage is detected, the RESET pin outputs “L” level.
X <sub>IN</sub>	Main clock input	Input	I/O pins of the main clock generating circuit. A ceramic resonator can be connected between X <sub>IN</sub> pin and X <sub>OUT</sub> pin. A feedback resistor is built-in between them.
X <sub>OUT</sub>	Main clock output	Output	
D <sub>0</sub> –D <sub>4</sub>	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.
D <sub>5</sub> /INT	Output port D/ Interrupt input	I/O	1-bit output port. Port D <sub>5</sub> is also used as an INT input pin. When D <sub>5</sub> /INT pin is used as the INT input pin, set the output latch to “1.” The output structure is N-channel open-drain.
D <sub>6</sub> /X <sub>CIN</sub>	Output port D/ Sub-clock input	I/O	Each pin of port D has an independent 1-bit output function. Ports D <sub>6</sub> and D <sub>7</sub> are also used as pins X <sub>CIN</sub> and X <sub>COUT</sub> for the sub-clock generating circuit, respectively. When pins D <sub>6</sub> /X <sub>CIN</sub> and D <sub>7</sub> /X <sub>COUT</sub> are used as the pins for the sub-clock generating circuit, a 32.0 kHz quartz-crystal oscillator can be connected between X <sub>CIN</sub> pin and X <sub>COUT</sub> pin. A feedback resistor is built-in between them.
D <sub>7</sub> /X <sub>COUT</sub>	Output port D/ Sub-clock output	Output	
P <sub>00</sub> –P <sub>03</sub>	I/O port P0	I/O	4-bit I/O port. It can be used as an input port when the output latch is set to “1.” The output structure is N-channel open-drain. Every pin of the ports has a key-on wakeup function and a pull-up function.
P <sub>10</sub> –P <sub>13</sub>	I/O port P1	I/O	4-bit I/O port. It can be used as an input port when the output latch is set to “1.” The output structure is N-channel open-drain. Every pin of the ports has a key-on wakeup function and a pull-up function. Both functions can be switched by software.
P <sub>20</sub> /SEG <sub>16</sub> – P <sub>23</sub> /SEG <sub>19</sub>	Input port P2/ Segment output	I/O	4-bit input port. Ports P <sub>20</sub> –P <sub>23</sub> are also used as the segment output pins SEG <sub>16</sub> –SEG <sub>19</sub> , respectively.
CARR	Carrier wave output for remote control	Output	Carrier wave output pin for remote control transmit. The output structure is the CMOS circuit.
SEG <sub>0</sub> –SEG <sub>15</sub>	Segment output	Output	LCD segment output pins.
COM <sub>0</sub> –COM <sub>3</sub>	Common output	Output	LCD common output pins. Pins COM <sub>0</sub> and COM <sub>1</sub> are used at 1/2 duty, pins COM <sub>0</sub> –COM <sub>2</sub> are used at 1/3 duty and pins COM <sub>0</sub> –COM <sub>3</sub> are used at 1/4 duty.

## MULTIFUNCTION

Pin	Multifunction	Pin	Multifunction
D5	INT	INT	D5
D6	X <sub>CIN</sub>	X <sub>CIN</sub>	D6
D7	X <sub>COUT</sub>	X <sub>COUT</sub>	D7
P20	SEG <sub>16</sub>	SEG <sub>16</sub>	P20
P21	SEG <sub>17</sub>	SEG <sub>17</sub>	P21
P22	SEG <sub>18</sub>	SEG <sub>18</sub>	P22
P23	SEG <sub>19</sub>	SEG <sub>19</sub>	P23

Notes 1: Pins except above have just single function.

2: The ports D5–D7 are the output port and ports P20–P23 are the input ports.

## CONNECTIONS OF UNUSED PINS

Pin	Connection	Pin	Connection
D0–D4	Connect to V <sub>SS</sub> , or set the output latch to “0” and open.	CARR	Open
D5/INT		SEG <sub>0</sub> –SEG <sub>15</sub>	Open
D6/X <sub>CIN</sub>	Select ports D6 and D7 and connect to V <sub>SS</sub> , or set the output latch to “0” and open.	COM <sub>0</sub> –COM <sub>3</sub>	Open
D7/X <sub>COUT</sub>		P0 <sub>0</sub> –P0 <sub>3</sub>	Set the output latch to “1” and open.
P20/SEG <sub>16</sub> –P23/SEG <sub>19</sub>	Select port P2 and connect to V <sub>SS</sub> , or select the segment output function and open.	P10–P13	Open or connect to V <sub>SS</sub> (Note)

Note: In order to connect ports P10–P13 to V<sub>SS</sub>, turn off their pull-up transistors (Pull-up control register PU0i=“0”) by software. In order to make these pins open, turn on their pull-up transistors (register PU0i=“1”) by software, or turn off their pull-up transistors (register PU0i=“0”) and set the output latch to “0” (i = 0, 1, 2, or 3).

Be sure to select the key-on wakeup function and the pull-up function with every one port.

(Note in order to set the output latch to “0” and make pins open)

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

(Note in order to connect unused pins to V<sub>SS</sub> or V<sub>DD</sub>)

- To avoid noise, connect the unused pins to V<sub>SS</sub> or V<sub>DD</sub> at the shortest distance using a thick wire.

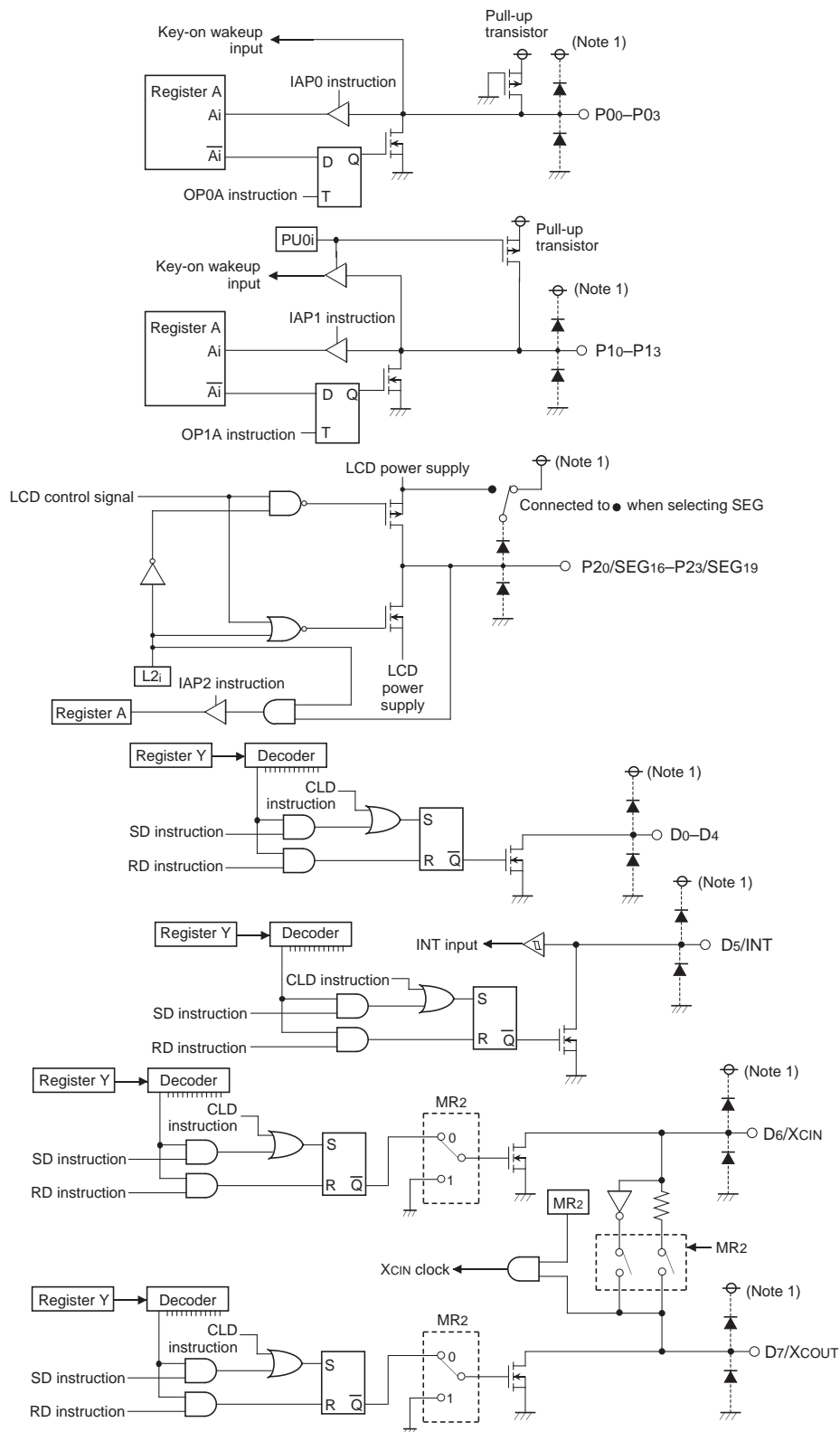
## PORT FUNCTION

Port	Pin	Input/Output	Output structure	Control bits	Control instructions	Control registers	Remark
Port D	D0–D4, D5/INT, D6/X <sub>CIN</sub> , D7/X <sub>COUT</sub>	Output (8)	N-channel open-drain	1	SD RD CLD	MR	
Port P0	P0 <sub>0</sub> –P0 <sub>3</sub>	I/O (4)	N-channel open-drain	4	OP0A IAP0		Pull-up functions Key-on wakeup functions
Port P1	P10–P13	I/O (4)	N-channel open-drain	4	OP1A IAP1	PU0	Pull-up functions (programmable) Key-on wakeup functions (programmable)
Port P2	P20/SEG <sub>16</sub> –P23/SEG <sub>19</sub>	Input (4)		4	IAP2		

# HARDWARE

## PIN DESCRIPTION

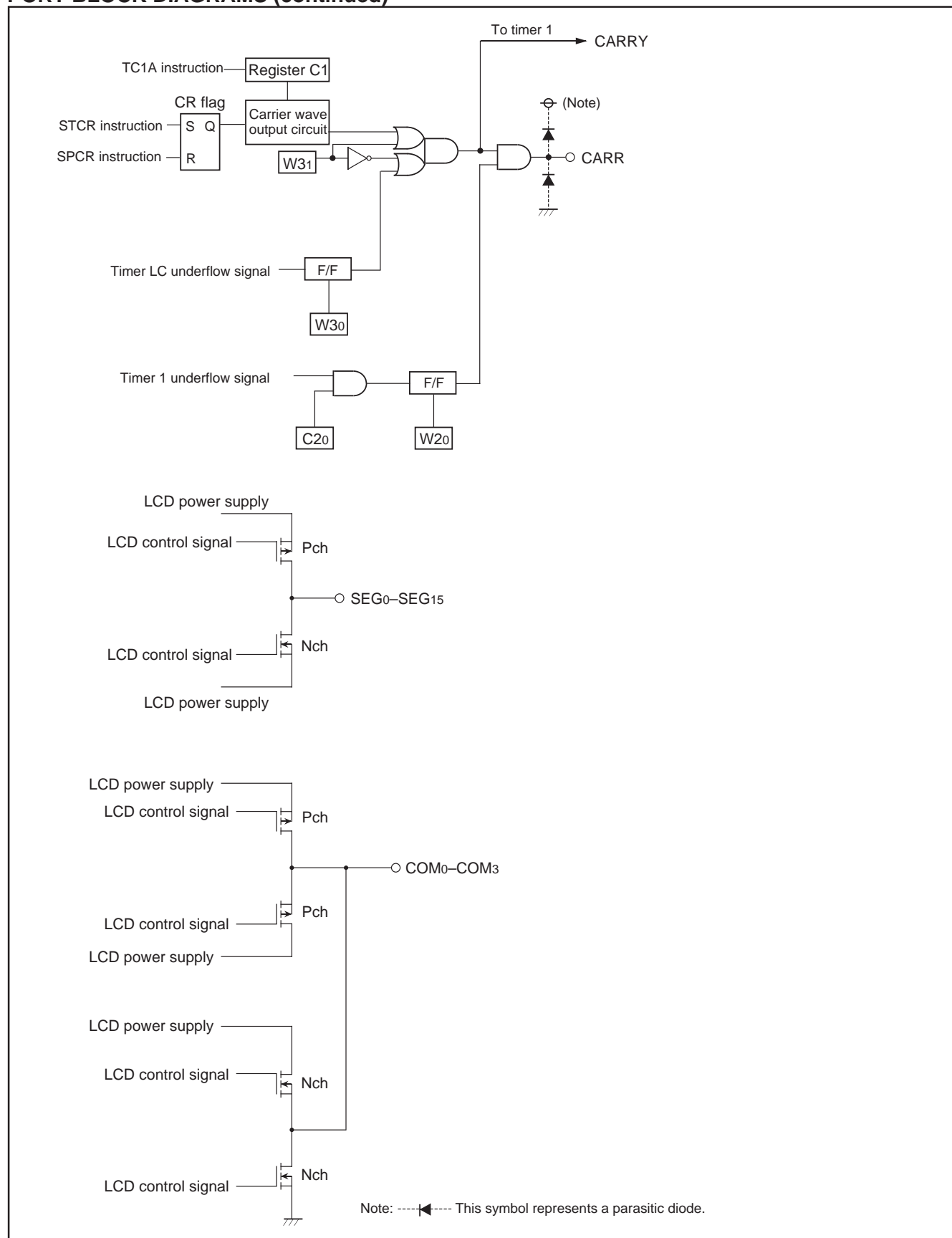
### PORT BLOCK DIAGRAMS



Notes 1: ----- This symbol represents a parasitic diode.

2: i represents bit 0, 1, 2 or 3.

### PORT BLOCK DIAGRAMS (continued)



# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### FUNCTIONAL 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 (CY)

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

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

It is unchanged with both A n instruction and AM instruction. The value of A<sub>0</sub> is stored in carry flag CY with the RAR instruction (Figure 2).

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

#### (3) Registers B and E

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

#### (4) Register D

Register D is a 3-bit register.

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

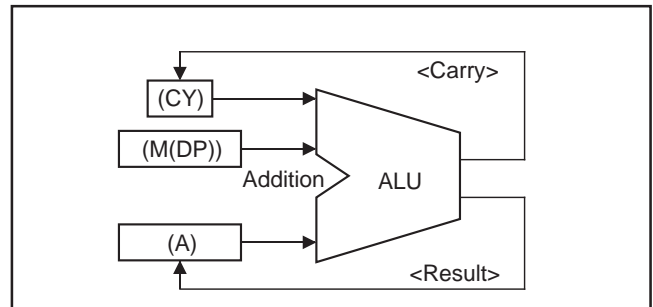


Fig. 1 AMC instruction execution example

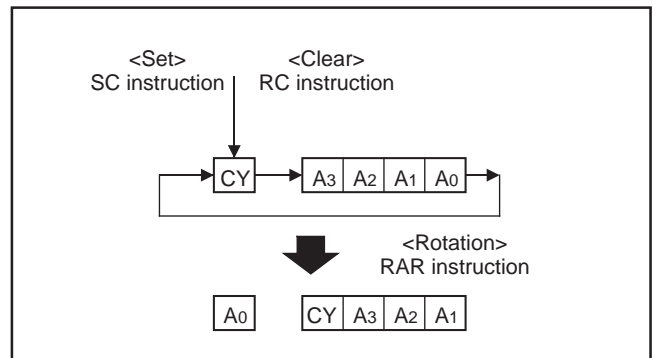


Fig. 2 RAR instruction execution example

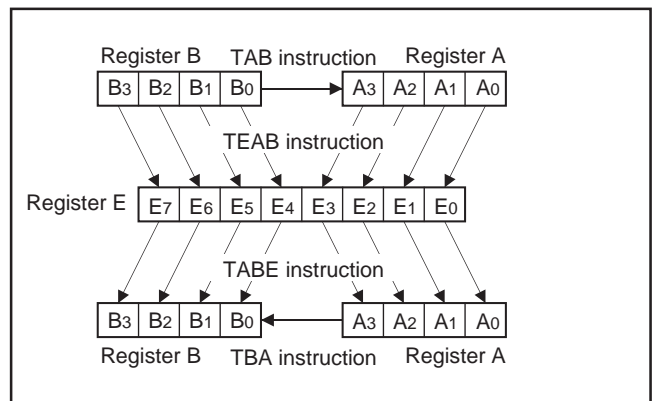


Fig. 3 Registers A, B and register E

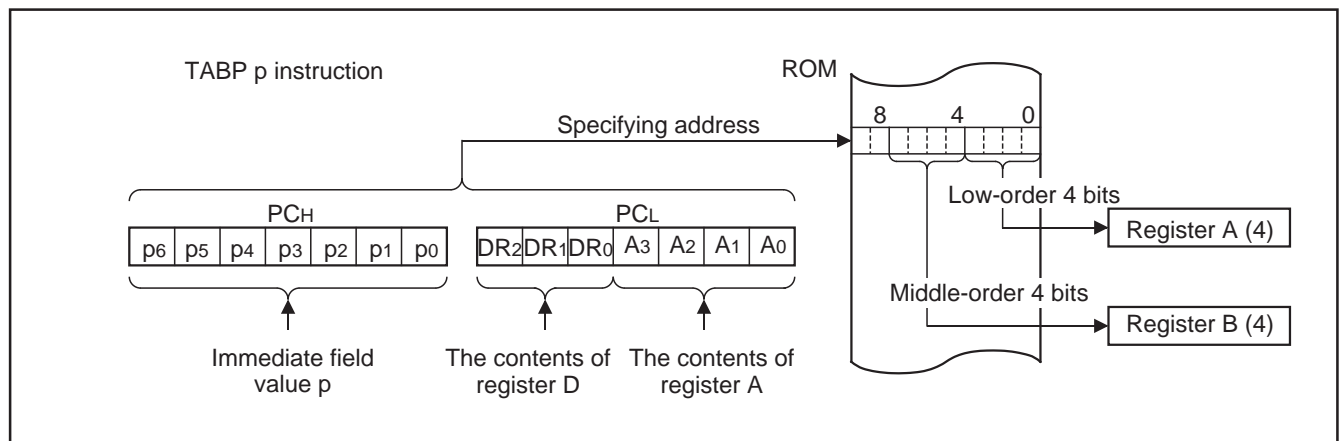


Fig. 4 TABP p instruction execution example

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

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

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

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

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

Figure 5 shows the stack registers (SKs) structure.

Figure 6 shows the example of operation at subroutine call.

### (6) Interrupt stack register (SDP)

Interrupt stack register (SDP) is a 1-stage register. When an interrupt occurs, this register (SDP) is used to temporarily store the contents of data pointer, carry flag, skip flag, register A, and register B just before an interrupt until returning to the original routine. Multiple interrupts cannot be used.

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

### (7) Skip flag

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

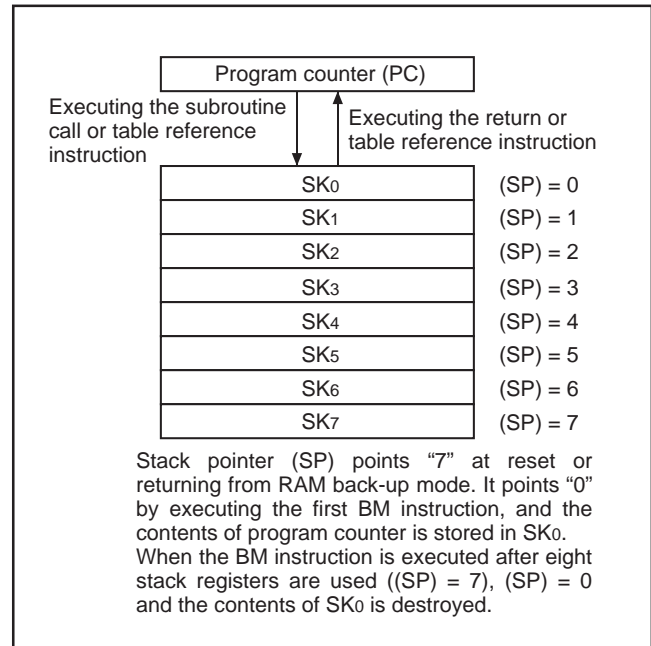


Fig. 5 Stack registers (SKs) structure

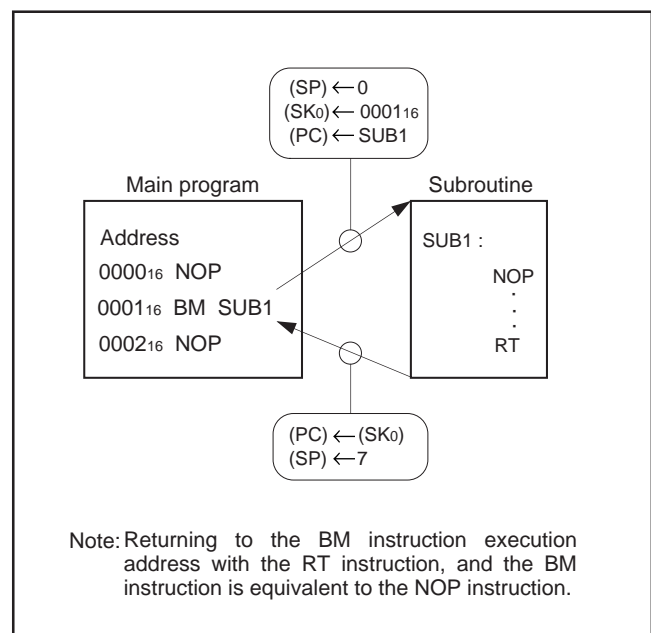


Fig. 6 Example of operation at subroutine call

# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### (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<sub>H</sub> (most significant bit to bit 7) which specifies to a ROM page and PC<sub>L</sub> (bits 6 to 0) which specifies an address within a page. After it reaches the last address (address 127) of a page, it specifies address 0 of the next page (Figure 7).

Make sure that the PC<sub>H</sub> does not specify after the last page of the built-in ROM.

### (9) Data pointer (DP)

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

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

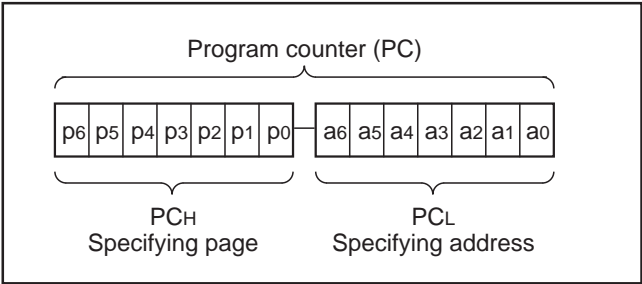


Fig. 7 Program counter (PC) structure

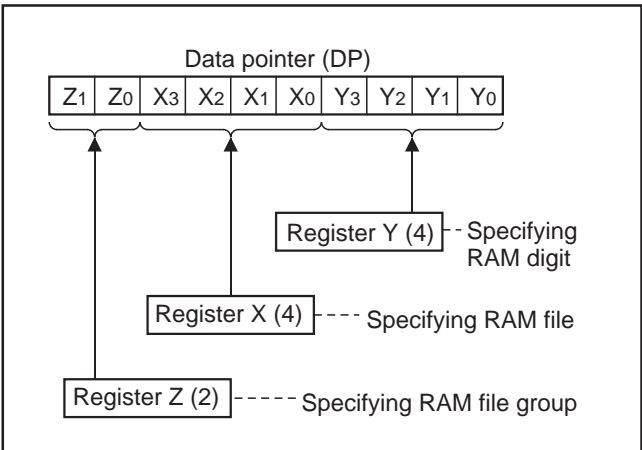


Fig. 8 Data pointer (DP) structure

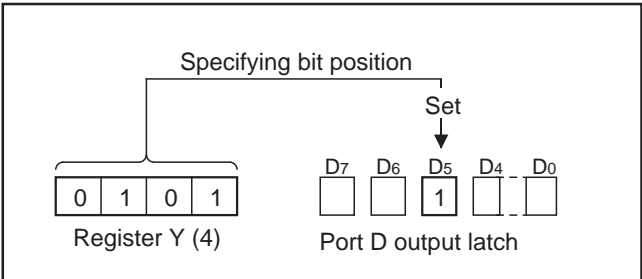


Fig. 9 SD instruction execution example

PROGRAM MEMORY (ROM)

1 word of ROM is composed of 10 bits. ROM is separated every 128 words by the unit of page (addresses 0 to 127). Table 1 shows the ROM size and pages. Figure 10 shows the ROM map of M34551E8.

Table 1 ROM size and pages

Product	ROM size (X 10 bits)	Pages
M34551M4	4096 words	32 (0 to 31)
M34551M8	8192 words	64 (0 to 63)
M34551E8	8192 words	64 (0 to 63)

A top part of page 1 (addresses 0080<sub>16</sub> to 00FF<sub>16</sub>) is reserved for interrupt addresses (Figure 11). When an interrupt occurs, the address (interrupt address) corresponding to each interrupt is set in the program counter, and the instruction at the interrupt address is executed. When using an interrupt service routine, write the instruction generating the branch to that routine at an interrupt address.

Page 2 (addresses 0100<sub>16</sub> to 017F<sub>16</sub>) is the special page for subroutine calls. Subroutines written in this page can be called from any page with the 1-word instruction (BM). Subroutines extending from page 2 to another page can also be called with the BM instruction when it starts on page 2.

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

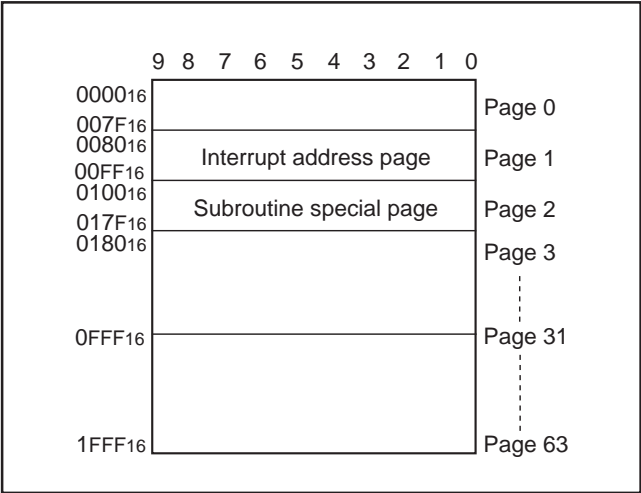


Fig. 10 ROM map of M34551E8

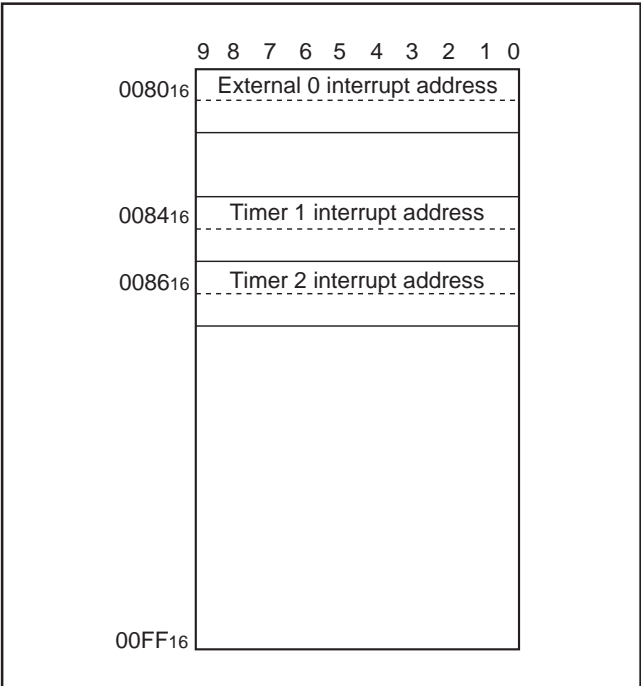


Fig. 11 Interrupt address page (addresses 0080<sub>16</sub> to 00FF<sub>16</sub>) structure



# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### 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, be sure to set a value to the data pointer certainly when returning from power down.

RAM includes the area corresponding to the LCD. A segment is turned on automatically when “1” is written in the bit corresponding to the segment.

Table 2 shows the RAM size. Figure 12 shows the RAM map.

Table 2 RAM size

Product	RAM size
M34551M4	280 words X 4 bits (1120 bits)
M34551M8	
M34551E8	

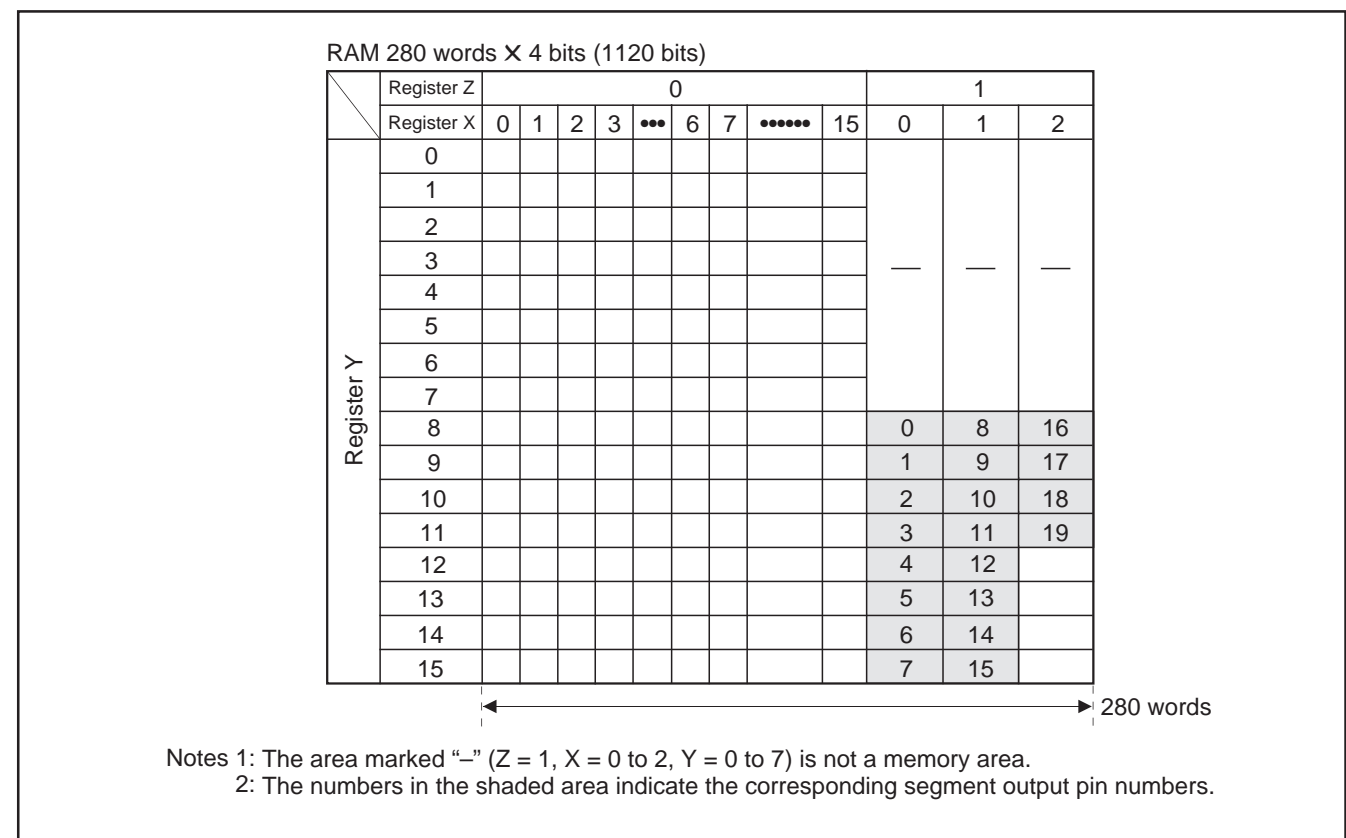


Fig. 12 RAM map

## INTERRUPT FUNCTION

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

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

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

### (1) Interrupt enable flag (INTE)

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

### (2) Interrupt enable bits (V10–V13)

Use an interrupt enable bit of interrupt control register V1 to select the corresponding interrupt request or skip instruction. Table 4 shows the interrupt request flag, interrupt enable bit and skip instruction.

Table 5 shows the interrupt enable bit function.

### (3) Interrupt request flag

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

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

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

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

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

**Table 3 Interrupt sources**

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

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

Interrupt name	Request flag	Enable bit	Skip instruction
External 0 interrupt	EXF0	V1 <sub>0</sub>	SNZ0
Timer 1 interrupt	T1F	V1 <sub>2</sub>	SNZT1
Timer 2 interrupt	T2F	V1 <sub>3</sub>	SNZT2

**Table 5 Interrupt enable bit function**

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

# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### (4) Internal state during an interrupt

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

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

### (5) Interrupt processing

When an interrupt occurs, a program at an interrupt address is executed after a branch to a sequence for storing data into stack register is performed. Write the branch instruction to an interrupt service routine at an interrupt address. Use the RTI instruction to return to main 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 to the main routine. (Refer to Figure 13)

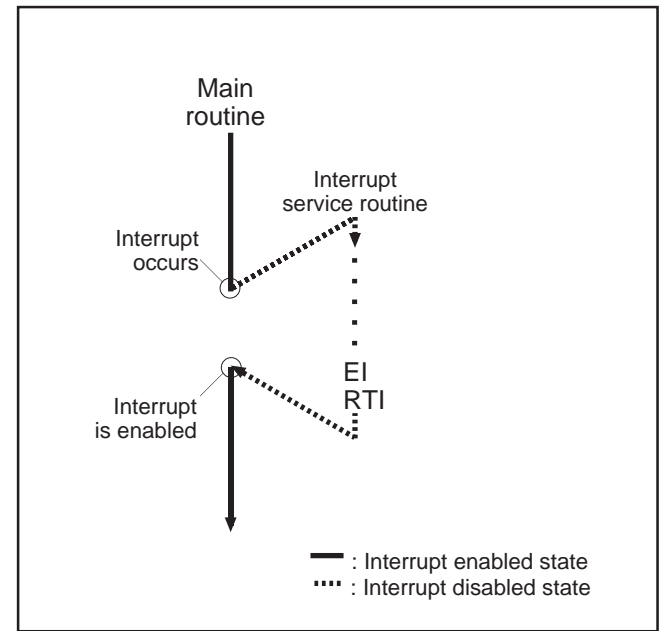


Fig. 13 Program example of interrupt processing

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

Fig. 14 Internal state when interrupt occurs

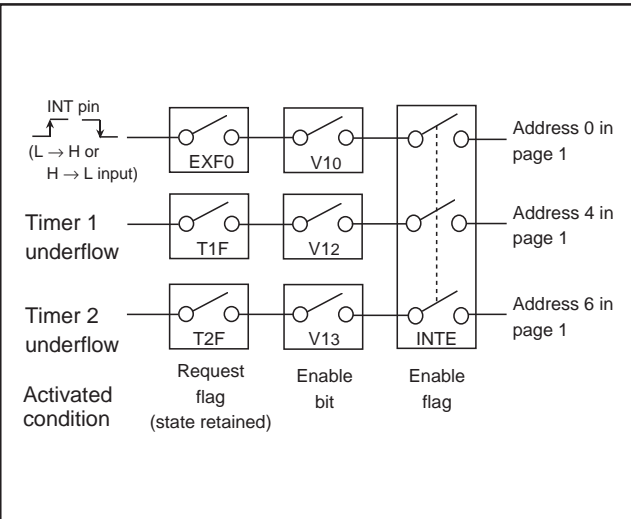


Fig. 15 Interrupt system diagram

### (6) Interrupt control register

#### ● 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.

**Table 6 Interrupt control register**

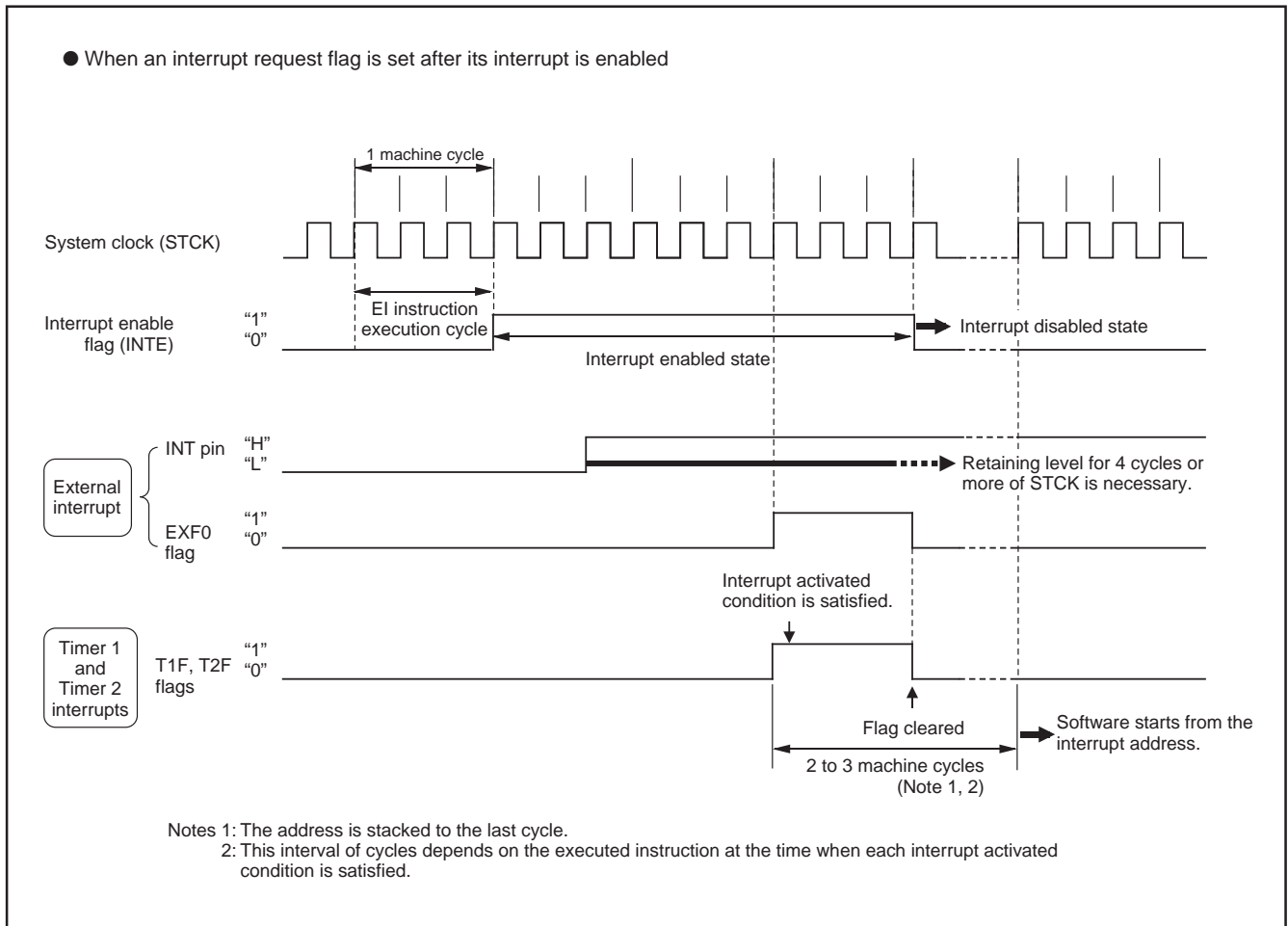
Interrupt control register V1		at reset : 0000 <sub>2</sub>		at power down : 0000 <sub>2</sub>	R/W
V1 <sub>3</sub>	Timer 2 interrupt enable bit	0	Interrupt disabled (SNZT2 instruction is valid)		
		1	Interrupt enabled (SNZT2 instruction is invalid)		
V1 <sub>2</sub>	Timer 1 interrupt enable bit	0	Interrupt disabled (SNZT1 instruction is valid)		
		1	Interrupt enabled (SNZT1 instruction is invalid)		
V1 <sub>1</sub>	Not used	0	This bit has no function, but read/write is enabled.		
		1			
V1 <sub>0</sub>	External 0 interrupt enable bit	0	Interrupt disabled (SNZ0 instruction is valid)		
		1	Interrupt enabled (SNZ0 instruction is invalid)		

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

### (7) Interrupt sequence

Interrupts occur only when the respective INTE flag, interrupt enable bits (V1<sub>0</sub>–V1<sub>3</sub>), and interrupt request flags (EXF0, T1F, T2F) are "1." The interrupt actually occurs 2 to 3 machine cycles after the cycle in which all three conditions are satisfied.

The interrupt occurs after 3 machine cycles only when the three interrupt conditions are satisfied on execution of other than one-cycle instructions (Refer to Figure 16).



**Fig. 16 Interrupt sequence**

# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### EXTERNAL INTERRUPTS

An external interrupt request occurs when a valid waveform (= waveform causing the external 0 interrupt) is input to an interrupt input pin (edge detection).

The external 0 interrupt can be controlled with the interrupt control register I1.

Table 7 External interrupt activated condition

Name	Input pin	Valid waveform	Valid waveform selection bit (I12)
External 0 interrupt	D5/INT	Falling waveform ("H"→"L")	0
		Rising waveform ("L"→"H")	1

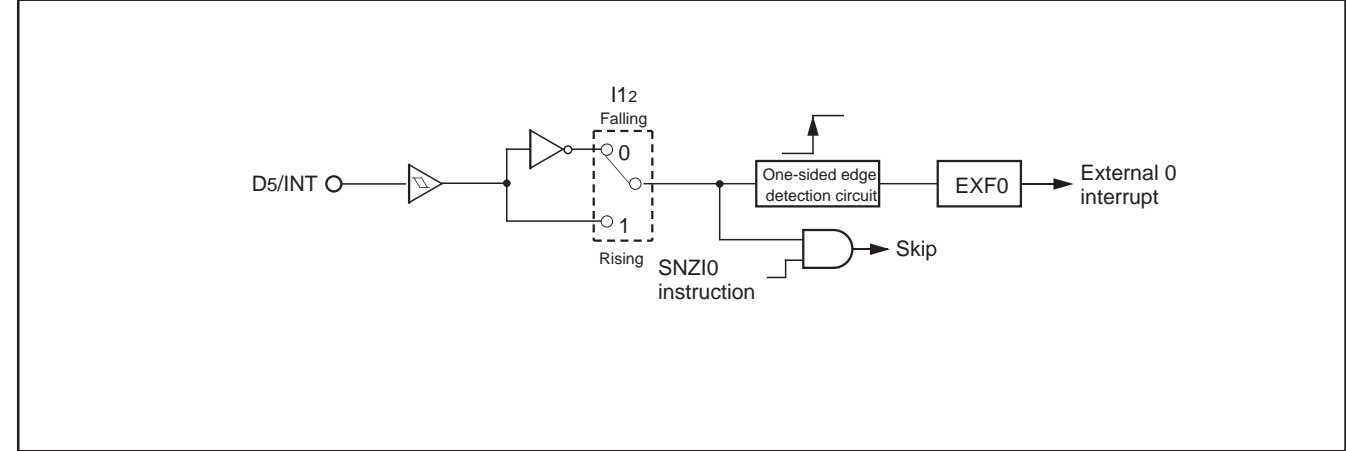


Fig. 17 External interrupt circuit structure

### (1) External 0 interrupt request flag (EXF0)

External 0 interrupt request flag (EXF0) is set to "1" when a valid waveform is input to D5/INT pin.

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

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

The D5/INT pin need not be selected the external interrupt input INT function or the normal output port D5 function. However, the EXF0 flag is set to "1" when a valid waveform output from port D5 is input to INT pin even if it is used as an output port D5.

#### ● External 0 interrupt activated condition

External 0 interrupt activated condition is satisfied when a valid waveform is input to D5/INT pin.

The valid waveform can be selected from rising waveform or falling waveform. An example of how to use the external 0 interrupt is as follows.

- ① Select the valid waveform with the bit 2 of register I1.
- ② Clear the EXF0 flag to "0" with the SNZ0 instruction.
- ③ Set the NOP instruction for the case when a skip is performed with the SNZ0 instruction.
- ④ Set both the external 0 interrupt enable bit (V10) and the INTE flag to "1."

The external 0 interrupt is now enabled. Now when a valid waveform is input to the D5/INT pin, the EXF0 flag is set to "1" and the external 0 interrupt occurs.

### (2) External interrupt control register

#### ● 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 T11A instruction. The TAI1 instruction can be used to transfer the contents of register I1 to register A.

**Table 8 External interrupt control register**

Interrupt control register I1		at reset : 0000 <sub>2</sub>	at power down : state retained	R/W
I1 <sub>3</sub>	Not used	0	This bit has no function, but read/write is enabled.	
		1		
I1 <sub>2</sub>	Interrupt valid waveform for INT pin selection bit (Note 2)	0	Falling waveform ("L" level of INT pin is recognized with the SNZI0 instruction)	
		1	Rising waveform ("H" level of INT pin is recognized with the SNZI0 instruction)	
I1 <sub>1</sub>	Not used	0	This bit has no function, but read/write is enabled.	
		1		
I1 <sub>0</sub>	Not used	0	This bit has no function, but read/write is enabled.	
		1		

Notes 1: "R" represents read enabled, and "W" represents write enabled.

- 2: Depending on the input state of D5/INT pin, the external interrupt request flag EXF0 may be set to "1" when the contents of I1<sub>2</sub> is changed. Accordingly, set a value to bit 2 of register I1 and execute the SNZ0 instruction to clear the EXF0 flag after executing at least one instruction.

# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### TIMERS

The 4551 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 set 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" every  $n$  count of a count pulse.

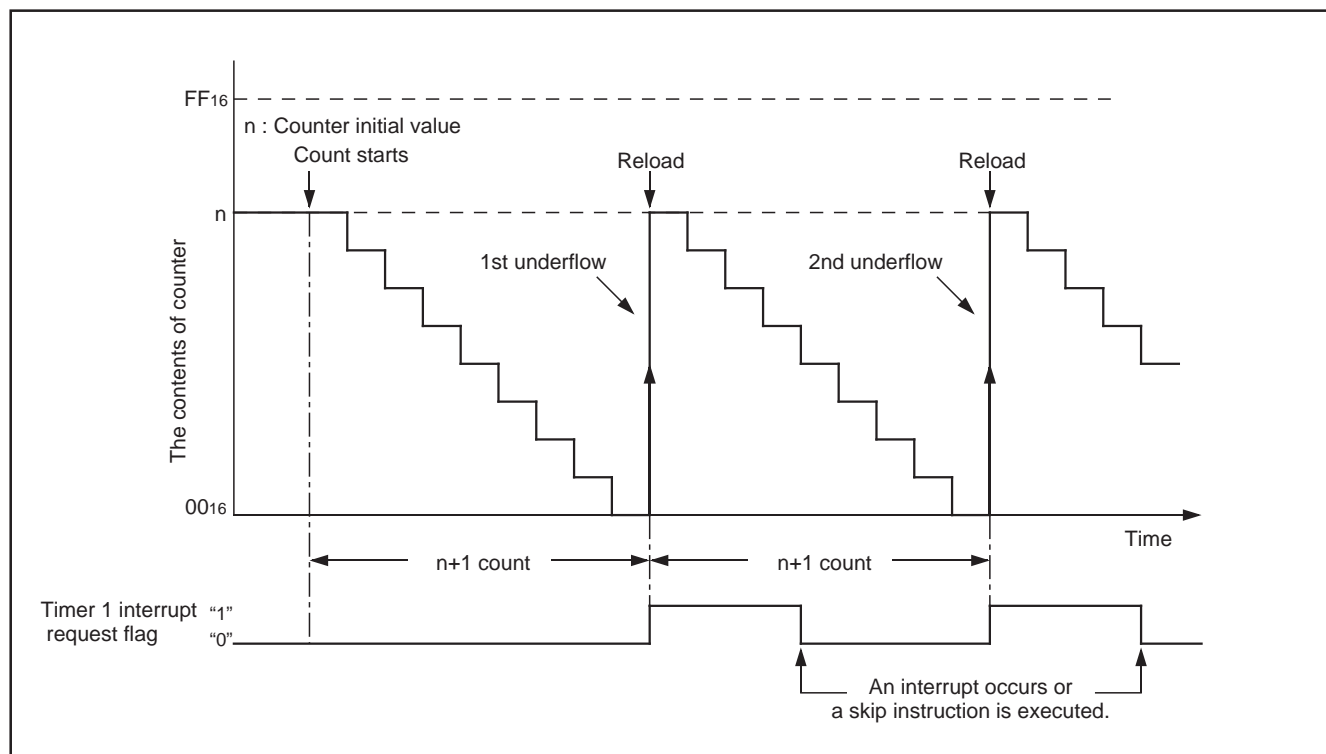


Fig. 18 Auto-reload function

The 4551 Group timer consists of the following circuits.

- Prescaler : frequency divider
  - Timer 1 : 8-bit programmable timer
  - Timer 2 : 14-bit fixed dividing frequency timer
  - Timer LC : 4-bit programmable timer
- (Timers 1 and 2 have the interrupt function, respectively)

Prescaler, timer 1, timer 2 and timer LC can be controlled with the timer control registers W1, W2 and W3. Each function is described below.

Table 9 Function related timers

Circuit	Structure	Count source	Frequency dividing ratio	Use of output signal	Control register
Prescaler	Frequency divider	• Instruction clock (INSTCK)	4, 8	• Timer 1 and 2 count sources	W1
Timer 1	8-bit programmable binary down counter	• Prescaler output (ORCLK) • Carrier generating circuit output (CARRY, CARRY/2)	1 to 256	• Timer 1 interrupt • Port CARR output control	W1 W2
Timer 2	14-bit fixed dividing frequency	• Prescaler output (ORCLK) • $f(X_{CIN})$	16384	• Timer 2 interrupt • Divider for LCD • Watchdog timer	W2
Timer LC	4-bit programmable binary down counter	• Bit 3 of timer 2 • System clock (STCK)	1 to 16	• Divider for LCD • Carrier output	W3

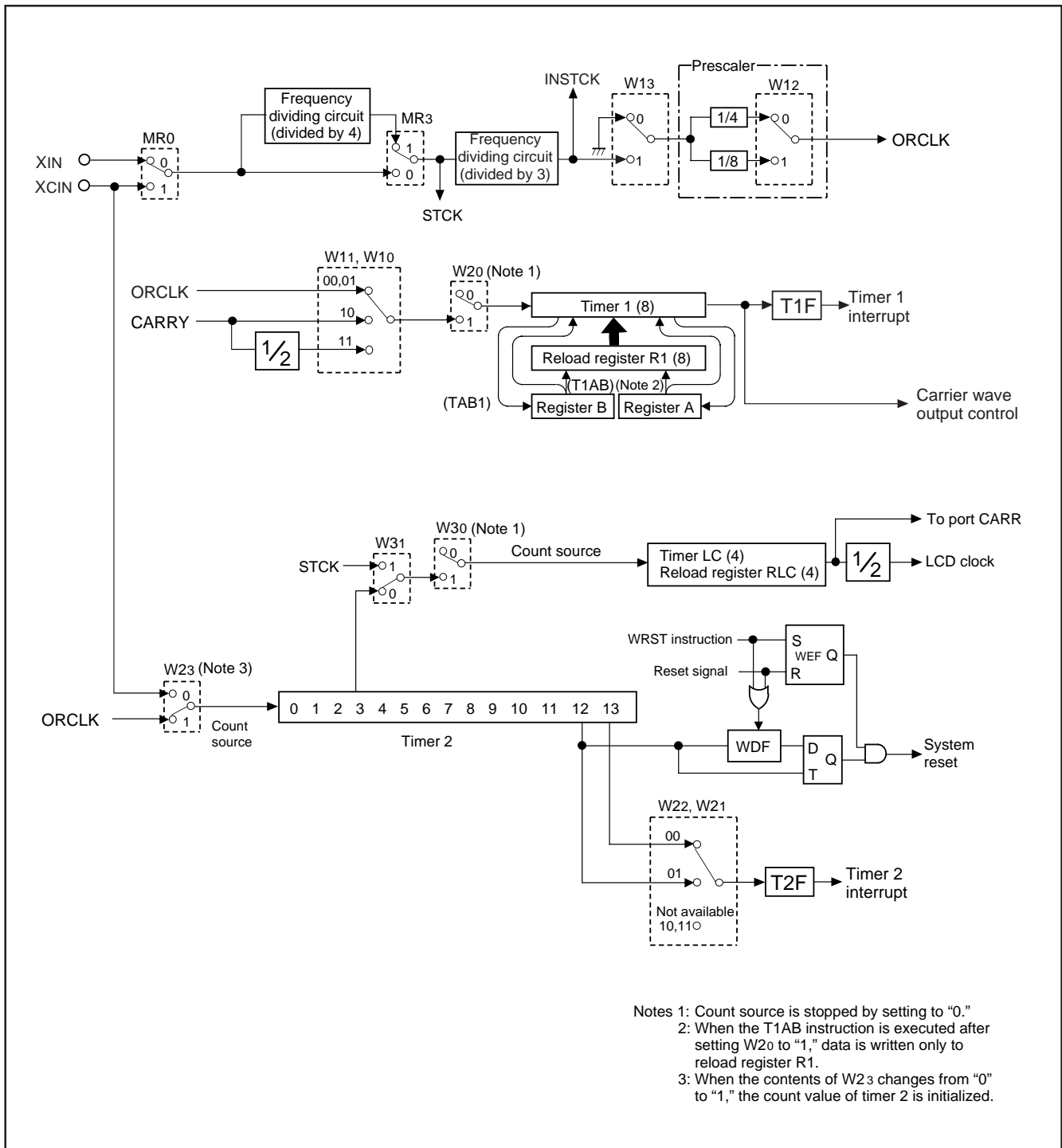


Fig. 19 Timers structure



# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

**Table 10 Timer control registers**

Timer control register W1		at reset : 0000 <sub>2</sub>		at power down : 0000 <sub>2</sub>	R/W
W1 <sub>3</sub>	Prescaler control bit	0	Stop (prescaler state initialized)		
		1	Operating		
W1 <sub>2</sub>	Prescaler dividing ratio selection bit	0	Instruction clock (INSTCK) divided by 4		
		1	Instruction clock (INSTCK) divided by 8		
W1 <sub>1</sub>	Timer 1 count source selection bits	W1 <sub>1</sub>	W1 <sub>0</sub>	Count source	
		0	0	Prescaler output (ORCLK)	
0		1			
W1 <sub>0</sub>		1	0	Carrier output (CARRY)	
		1	1	Carrier output divided by 2 (CARRY/2)	

Timer control register W2		at reset : 1000 <sub>2</sub>		at power down : --- 0 <sub>2</sub>	R/W
W2 <sub>3</sub>	Timer 2 count source selection bit	0	f(X <sub>CIN</sub> )		
		1	Prescaler output (ORCLK)		
W2 <sub>2</sub>	Timer 2 count value selection bits	W2 <sub>2</sub>	W2 <sub>1</sub>	Count source	
		0	0	Underflow occur every 2 <sup>14</sup> count	
0		1	Underflow occur every 2 <sup>13</sup> count		
W2 <sub>1</sub>		1	0	Not available	
		1	1	Not available	
W2 <sub>0</sub>	Timer 1 control bit	0	Stop (timer 1 state retained)		
		1	Operating		

Timer control register W3		at reset : 00 <sub>2</sub>		at power down : state retained	R/W
W3 <sub>1</sub>	Timer LC count source selection bit	0	Bit 3 of timer 2 is output (timer 2 count source divided by 16)		
		1	System clock (STCK)		
W3 <sub>0</sub>	Timer LC control bit	0	Stop (timer LC state retained)		
		1	Operating		

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

"—" represents state retained.

### (1) Timer control registers

#### ● Timer control register W1

Register W1 controls the count source of timer 1, the frequency dividing ratio and count operation of prescaler. Set the contents of this register through register A with the TW1A instruction. The TAW1 instruction can be used to transfer the contents of register W1 to register A.

#### ● Timer control register W2

Register W2 controls the count operation of timer 1 and count operation and count source of timer 2. Set the contents of this register through register A with the TW2A instruction. The TAW2 instruction can be used to transfer the contents of register W2 to register A.

#### ● Timer control register W3

Register W3 controls the count operation and count source of timer LC. 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.

**(2) Precautions**

Note the following for the use of timers.

- Prescaler

Stop the prescaler operation to change its frequency dividing ratio.

- Count source

Stop timer 1 or timer LC counting to change its count source. When timer 2 count source changes from  $f(X_{CIN})$  to ORCLK ( $W2_3 = "0" \rightarrow W2_3 = "1"$ ), the count value of timer 2 is initialized. However, when timer 2 count source changes from ORCLK to  $f(X_{CIN})$  ( $W2_3 = "1" \rightarrow W2_3 = "0"$ ) or the same count source is set again ( $W2_3 = "0" \rightarrow W2_3 = "0"$  or  $W2_3 = "1" \rightarrow W2_3 = "1"$ ), the count value of timer 2 is not initialized.

- Timer 2

Timer 2 has the watchdog timer function (WDT). When timer 2 is used as the WDT, note that the processing to initialize the count value and the execution of the WRST instruction.

- Reading the count value

Stop the prescaler and then execute the TAB1 instruction to read timer 1 data.

- Writing to reload register R1

When writing data to reload register R1 while timer 1 is operating, avoid a timing when timer 1 underflows.

**(3) Prescaler**

Prescaler is a frequency divider. Its frequency dividing ratio can be selected. The count source of prescaler is the instruction clock (INSTCK).

Use the bit 2 of register W1 to select the prescaler dividing ratio and the bit 3 to start and stop its operation. When the bit 3 of register W1 is cleared to "0," prescaler is initialized, and the output signal (ORCLK) stops.

**(4) Timer 1 (interrupt function)**

Timer 1 is an 8-bit binary down counter with the timer 1 reload register (R1). When timer 1 stops, data can be set simultaneously in timer 1 and the reload register (R1) with the T1AB instruction. When timer 1 is operating, data can be set only in the reload register (R1) with the T1AB instruction. When setting the next count data to reload register R1 while timer 1 is operating, be sure to set data before timer 1 underflows.

Timer 1 starts counting after the following process;

- ① set data in timer 1,
- ② select the count source with bits 0 and 1 of register W1,
- ③ set the bit 0 of register W2 to "1."

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).

When a value set in reload register R1 is  $n$ , timer 1 divides the count source signal by  $n + 1$  ( $n = 0$  to 255).

Data can be read from timer 1 to registers A and B. Stop counting and then execute the TAB1 instruction to read its data.

**(5) Timer 2 (interrupt function)**

Timer 2 is a 14-bit binary down counter.

Timer 2 starts counting after the following process;

- ① select the count source with the bit 3 of register W2, and
- ② the clock as a count source is supplied.

Timer 2 stops counting and its count value is retained when supply of a clock as a count source stops. Timer 2 is initialized at reset and when the count source changes from  $f(X_{CIN})$  ( $W2_3 = "0"$ ) to ORCLK ( $W2_3 = "1"$ ).

The count value to set the timer 2 interrupt request flag (T2F) to "1" can be selected from every 8192 count or every 16384 count with bits 1 and 2 of register W2. The count source signal divided by 16 is output from timer 2.

Timer 2 can be used as a counter for clock in the clock operating mode (POF instruction executed).

**(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.

Timer LC starts counting after the following process;

- ① set data in timer LC,
- ② select the count source with the bit 1 of register W3,
- ③ set the bit 0 of register W3 to "1."

Timer LC is the timer for LCD clock generating. Also, it can be used as the multi-carrier generator by setting the bit 1 of register W3 to "1" and selecting the system clock (STCK) as a count source. When the multi-carrier generator is selected, the waveform which is the timer LC underflow signal divided by 2 can be output as a carrier wave from port CARR. At this time, stop the carrier generating circuit and LCD control circuit. When the multi-carrier generator (duty ratio: 1/2 fixed) is used, the enable/stop of the carrier wave output from port CARR can be set by the stop of timer LC or the carrier wave output auto-control function by timer 1.

**(7) Timer interrupt request flags (T1F and T2F)**

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 and SNZT2).

Use the interrupt control register V1 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.

# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### WATCHDOG TIMER

Watchdog timer provides a method to reset the system when a program runs wild. Watchdog timer consists of timer 2, watchdog timer enable flag (WEF), and watchdog timer flag (WDF). When the WRST instruction is executed after system is released from reset, the WEF flag is set to "1." At this time, the watchdog timer starts operating. When the WEF flag is set to "1," it cannot be cleared to "0" until system reset is performed. Also, when the WRST instruction is not executed once, watchdog timer does not operate because the WEF flag retains "0."

When the watchdog timer is operating, the WDF flag is set to "1" every time the bit 12 of timer 2 is cleared from "1" to "0." This means that count is performed 8192 times. When the bit 12 of

timer 2 is cleared from "1" to "0" while the WDF flag is set to "1," the internal reset signal is generated and system reset is performed.

The WDF flag can be cleared to "0" with the WRST instruction. In the RAM back-up mode, though timer 2 count operation stops, its count value is retained and the WDF flag is initialized. In the clock operating mode, timer 2 count operation is continued and the WDF flag is initialized.

When using the watchdog timer, execute the WRST instruction at a certain cycle which consists of timer 2's 8191 counts or less to keep the microcomputer operation normal.

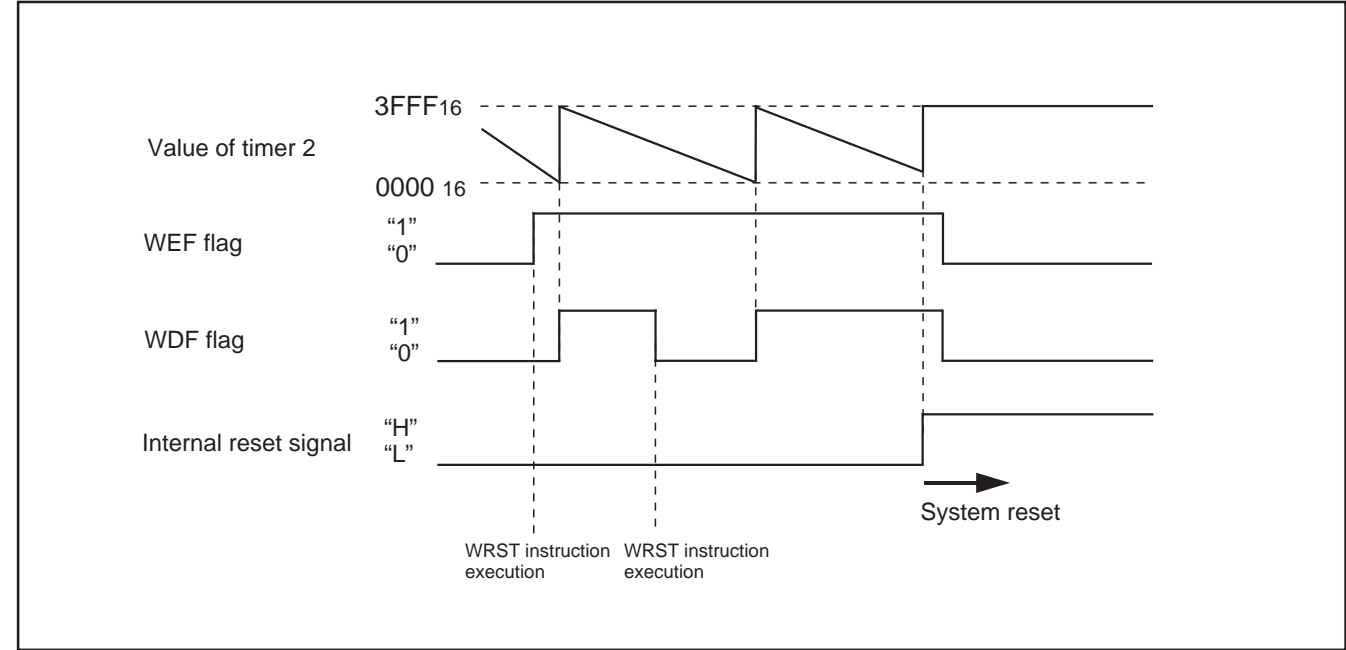


Fig. 20 Watchdog timer function

The contents of the WDF flag are initialized in the RAM back-up mode.

If the WDF flag is set to "1" at the same time that the microcomputer enters the RAM back-up mode, system reset may be performed.

When using the watchdog timer and the RAM back-up mode, initialize the WDF flag with the WRST instruction just before the microcomputer enters the RAM back-up mode (refer to Figure 21).

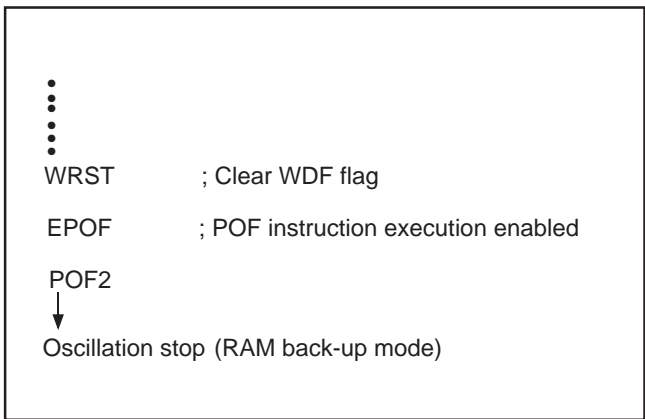


Fig. 21 Program example to enter the RAM back-up mode when using the watchdog timer

### CARRIER GENERATING CIRCUIT

The 4551 Group has a carrier generating circuit that generates the transfer waveform by dividing the system clock (STCK) for each remote control carrier wave. Each carrier waveform can be output by setting the carrier wave selection register (C1).

Also, timer 1 can auto-control the carrier wave output from port CARR by setting the carrier wave output control register (C2).

Carrier wave selection register C1													
(at reset: 0 1 1 1 2, at power down: 0 1 1 1 2, W)													
Register C1 Setting value				Output waveform						Carrier wave			
C13	C12	C11	C10	STCR instruction	Output waveform						SPCR instruction	Frequency	Duty
0	0	0	0	"H" "L"								STCK/24	1/3
0	0	0	1	"H" "L"									1/2
0	0	1	0	"H" "L"								STCK/16	1/4
0	0	1	1	"H" "L"									1/2
0	1	0	0	"H" "L"								STCK/2	1/2
0	1	0	1	"H" "L"								No carrier wave	
1	1	0	1	"H" "L"									
0	1	1	0	"H" "L"								No available	
1	1	1	0	"H" "L"									
0	1	1	1	"H" "L"								'L' fixed	
1	1	1	1	"H" "L"									
1	0	0	0	"H" "L"								STCK/12	1/3
1	0	0	1	"H" "L"									1/2
1	0	1	0	"H" "L"								STCK/8	1/4
1	0	1	1	"H" "L"									1/2
1	1	0	0	"H" "L"								STCK	1/2

Note: "W" represents write enabled.

Fig. 22 Carrier wave selection register

# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

Table 11 Carrier generating circuit control register and control flag

Carrier wave output control register C2		at reset : 0 <sub>2</sub>		at power down : 0 <sub>2</sub>	W
C2 <sub>0</sub>	Carrier wave output auto-control bit	0	Auto-control output by timer 1 is invalid		
		1	Auto-control output by timer 1 is valid		

Carrier wave generating control flag CR		at reset : 0 <sub>2</sub>		at power down : 0 <sub>2</sub>	W
CR	Carrier wave generating control	0	Carrier wave generating stop (SPCR instruction)		
		1	Carrier wave generating start (STCR instruction)		

Note: "W" represents write enabled.

### (1) Carrier generating circuit related registers

- Carrier wave selection register C1  
Each carrier waveform can be selected by setting the register C1. Set the contents of this register through register A with the TC1A instruction.
- Carrier wave output control register C2  
Timer 1 can auto-control the output enable interval and the output disable interval of the carrier wave output from port CARR by setting the register C2. Set the contents of this register through register A with the TC2A instruction. The setting of the output enable/disable interval is described below.

- ① Validate the carrier wave output auto-control function (C2<sub>0</sub>=“1”).
- ② Select the carrier wave or the carrier wave divided by 2 as the timer 1 count source.
- ③ Set the count value (the output enable interval of carrier wave from port CARR) to timer 1.
- ④ Operate timer 1 (W2<sub>0</sub>=“1”).
- ⑤ Operate the carrier generating circuit (STCR instruction executed).
- ⑥ Set the next count value (the output disable interval of carrier wave from port CARR) to reload register R1 before timer 1 underflow occurs.

The carrier wave is output from port CARR until the first timer 1 underflow occurs. The output of the carrier wave from port CARR is disabled and the next count value is loaded from reload register R1 to timer 1 by the first timer 1 underflow. Then, the output of carrier wave is disabled until the second timer 1 underflow. Also, the next enable interval of the carrier wave output can be set by setting the third count value to timer 1 reload register before the second timer 1 underflow occurs. If the carrier wave output auto-control function is invalidated (C2<sub>0</sub>=“0”) while the carrier wave output is auto-controlled, the output of port CARR retains the state when the auto-control is invalidated regardless of timer 1 underflow. This state can be terminated by timer 1 stop (W2<sub>0</sub>=“0”). When the carrier wave output auto-control function is validated (C2<sub>0</sub>=“1”) again after it is invalidated (C2<sub>0</sub>=“0”), the auto-control of carrier wave output is started again when the next timer 1 underflow occurs.

### (2) Carrier wave generating control flag (CR)

The CR flag is used to control the carrier wave generating operation of the carrier generating circuit. The CR flag is “1” and the carrier wave generating is started by executing the STCR instruction. The CR flag is “0” and the carrier wave generating is stopped by executing the SPCR instruction. The CR flag is “0” at system reset.

### (3) Note on the carrier generating circuit stop

In order to stop the carrier wave which has the cycle longer than that of the instruction clock with the SPCR instruction, stop it at the point when the carrier wave outputs “L” level in the SPCR instruction execution cycle. If this condition is not satisfied, the last “H” output interval of carrier wave is shortened.

### (4) Notes when using the carrier wave output auto-control function

- Execute the STCR instruction after setting the timer 1 and register C2 in order to start the carrier generating circuit operation.
- Stop the timer 1 (W2<sub>0</sub>=“0”) after stopping the carrier generating circuit (SPCR instruction executed) while the carrier wave output is disabled in order to stop the carrier wave output auto-control operation.
- If the carrier wave output auto-control function is invalidated (C2<sub>0</sub>=“0”) while the carrier wave output is auto-controlled, the output of port CARR retains the state when the auto-control is invalidated regardless of timer 1 underflow. This state can be terminated by timer 1 stop (W2<sub>0</sub>=“0”). When the carrier wave output auto-control function is validated (C2<sub>0</sub>=“1”) again after it is invalidated (C2<sub>0</sub>=“0”), the auto-control of carrier wave output is started again when the next timer 1 underflow occurs. However, when the carrier wave output auto-control bit is changed during timer 1 underflow, the error-operation may occur.
- Use the carrier wave or the carrier wave divided by 2 as the timer 1 count source when the carrier wave output auto-control function is selected. If the ORCLK is used as the count source, a hazard may occur in port CARR output because ORCLK is not synchronized with the carrier wave.
- When “no carrier wave” is selected with register C1 ((C1<sub>3</sub>C1<sub>2</sub>C1<sub>1</sub>C1<sub>0</sub>) = (0101), (1101)), the enable/disable of the carrier wave output cannot be controlled by the carrier wave output auto-control function.

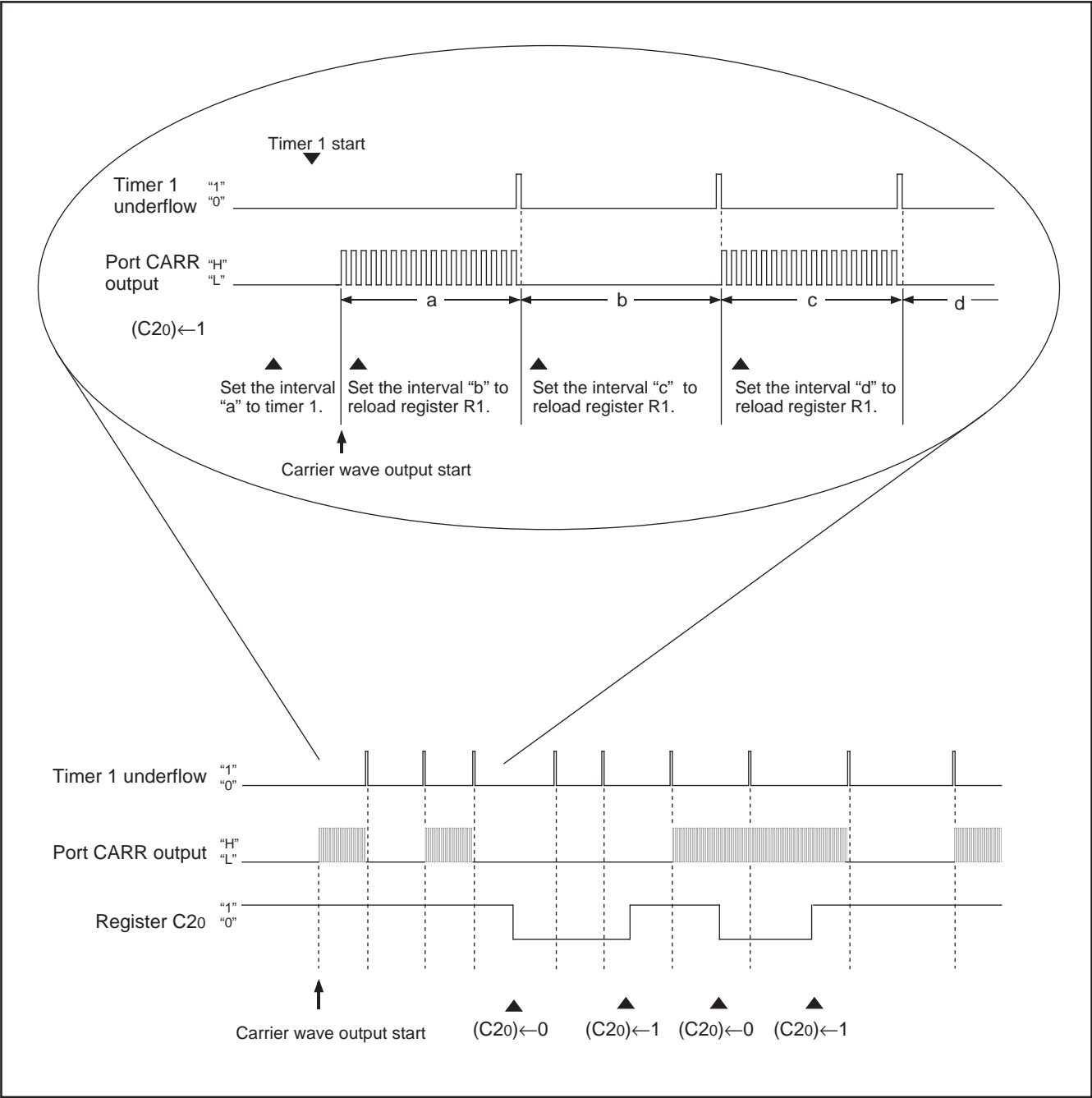


Fig. 23 Carrier wave output auto-control by timer 1

# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### LCD FUNCTION

The 4551 Group has an LCD (Liquid Crystal Display) controller/driver. When data are set in timer control registers (W2, W3), timer LC, LCD control registers (L1, L2), and LCD RAM, 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 20 segment signal output pins can be used to drive the LCD. By using these pins, up to 80 segments (when 1/4 duty and 1/3 bias are selected) can be controlled to display. When the required number of segment pins is 19 or less, pins SEG16–SEG19 (4) can be used as input ports P20–P23.

#### (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 12 Duty and maximum number of displayed pixels**  
**(2) LCD clock control**

Duty	Maximum number of displayed pixels	Used COM pins
1/2	40 segments	COM0, COM1 (Note)
1/3	60 segments	COM0–COM2 (Note)
1/4	80 segments	COM0–COM3

Note: Leave unused COM pins open.

The LCD clock is determined by the timer 2 count source selection bit (W23), timer LC control bit (W30), and timer LC. Accordingly, the frequency (F) of the LCD clock is obtained by the following formula. Numbers ① to ⑤ shown below the formula correspond to numbers in Figure 24, respectively.

- When using the prescaler output (ORCLK) as timer 2 count source (W23="1")

$$F = \text{ORCLK} \times \frac{1}{16} \times \frac{1}{LC+1} \times \frac{1}{2}$$

①                      ②③                      ④                      ⑤

- When using the f(XCIN) as timer 2 count source (W23="0")

$$F = f(\text{XCIN}) \times \frac{1}{16} \times \frac{1}{LC+1} \times \frac{1}{2}$$

①                      ②③                      ④                      ⑤

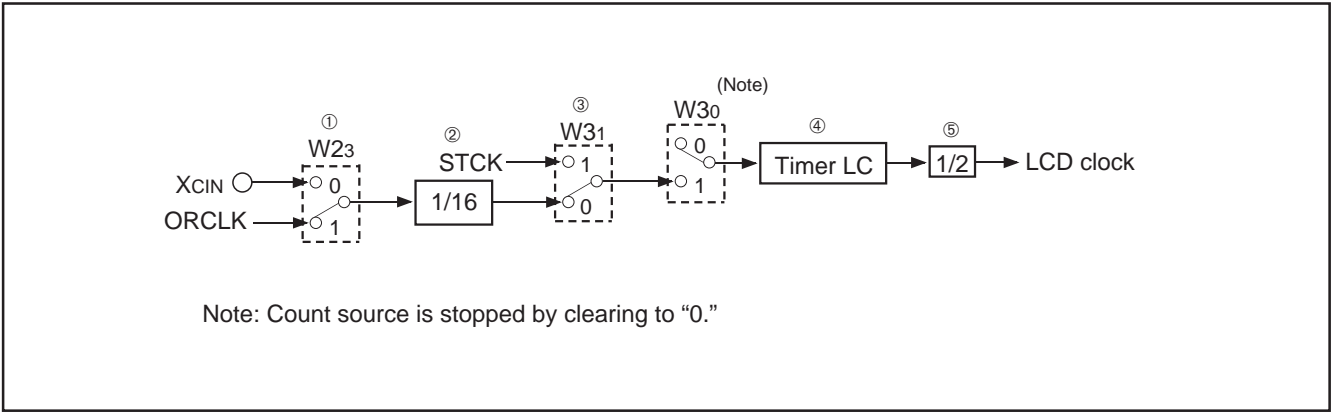
[LC: 0 to 15]

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

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

$$\text{Frame period} = \frac{n}{F} \text{ (s)}$$

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



**Fig. 24 LCD clock control circuit structure**





# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### (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.

### (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 ( $=V_{DD}$ ).

Z	1											
X	0				1				2			
Y \ Bit	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
9	SEG1	SEG1	SEG1	SEG1	SEG9	SEG9	SEG9	SEG9	SEG17	SEG17	SEG17	SEG17
10	SEG2	SEG2	SEG2	SEG2	SEG10	SEG10	SEG10	SEG10	SEG18	SEG18	SEG18	SEG18
11	SEG3	SEG3	SEG3	SEG3	SEG11	SEG11	SEG11	SEG11	SEG19	SEG19	SEG19	SEG19
12	SEG4	SEG4	SEG4	SEG4	SEG12	SEG12	SEG12	SEG12				
13	SEG5	SEG5	SEG5	SEG5	SEG13	SEG13	SEG13	SEG13				
14	SEG6	SEG6	SEG6	SEG6	SEG14	SEG14	SEG14	SEG14				
15	SEG7	SEG7	SEG7	SEG7	SEG15	SEG15	SEG15	SEG15				
COM	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0

Note: The area marked "—" is not the LCD display RAM.

Fig. 26 LCD RAM map

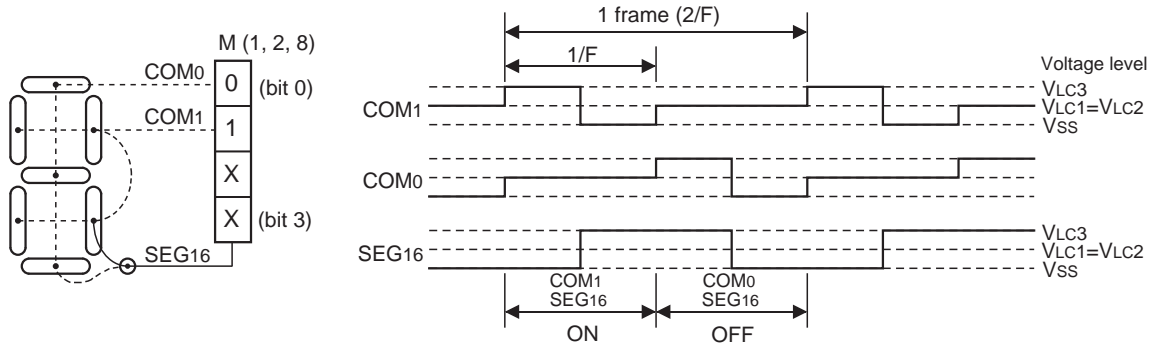
Table 13 LCD control registers

LCD control register L1		at reset : 0000 <sub>2</sub>		at power down : state retained	R/W
L13	Not used	0	This bit has no function, but read/write is enabled		
		1			
L12	LCD on/off bit	0	Off		
		1	On		
L11	LCD duty and bias selection bits	L11	L10	Duty	Bias
		0	0	Not available	
		0	1	1/2	1/2
		1	0	1/3	1/3
L10		1	1	1/4	1/3

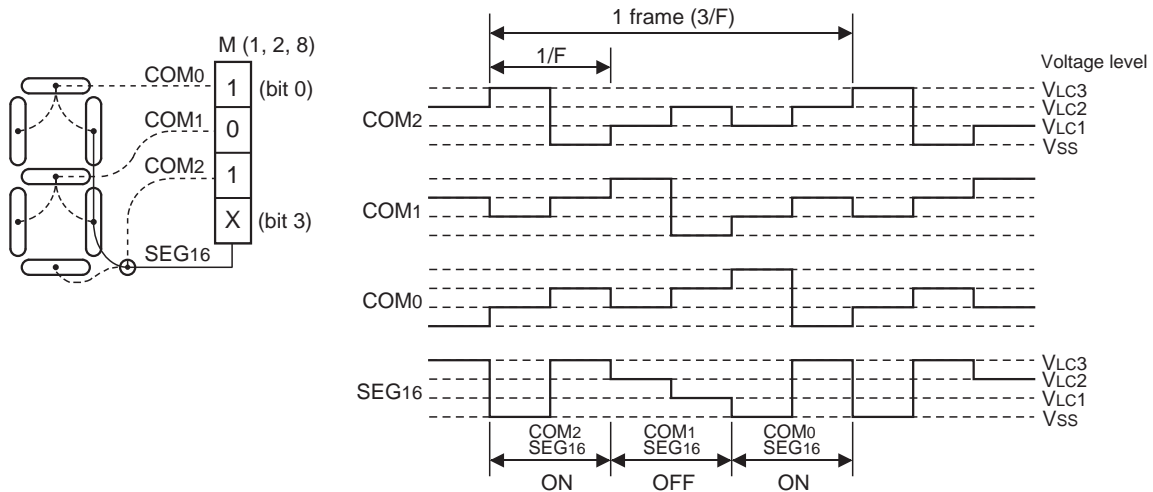
LCD control register L2		at reset : 1111 <sub>2</sub>		at power down : state retained	W
L23	P23/SEG19 pin function switch bit	0	SEG19		
		1	P23		
L22	P22/SEG18 pin function switch bit	0	SEG18		
		1	P22		
L21	P21/SEG17 pin function switch bit	0	SEG17		
		1	P21		
L20	P20/SEG16 pin function switch bit	0	SEG16		
		1	P20		

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

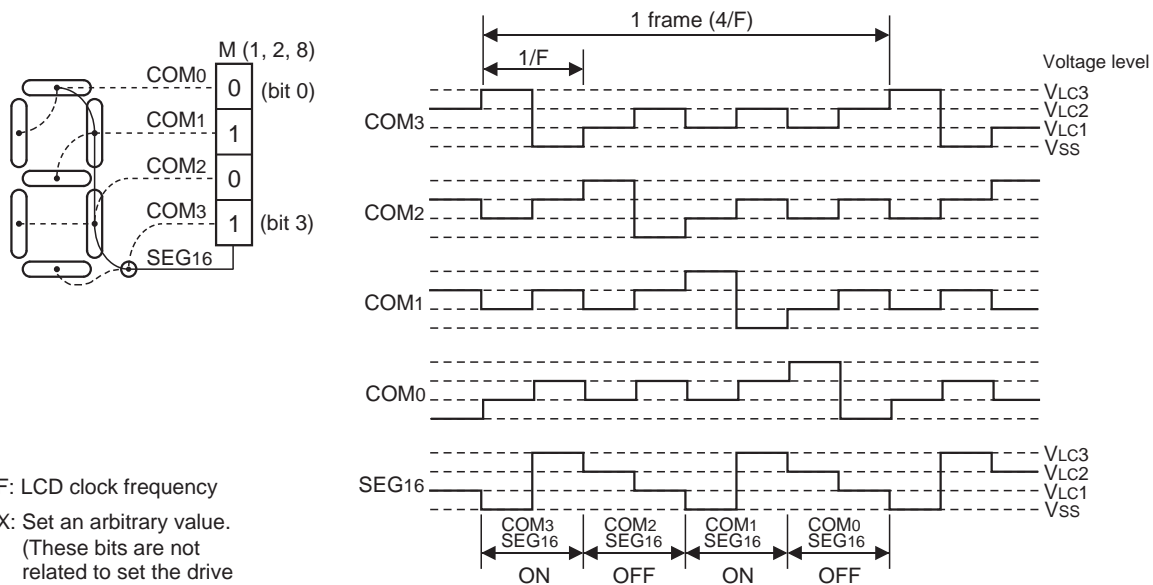
1/2 Duty, 1/2 Bias: When writing (XX10)<sub>2</sub> to address M (1, 2, 8) in RAM.



1/3 Duty, 1/3 Bias: When writing (X101)<sub>2</sub> to address M (1, 2, 8) in RAM.



1/4 Duty, 1/3 Bias: When writing (1010)<sub>2</sub> to address M (1, 2, 8) in RAM.



F: LCD clock frequency  
X: Set an arbitrary value.  
(These bits are not related to set the drive waveform at each duty.)

Fig. 27 LCD controller/driver structure

# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### RESET FUNCTION

System reset is performed by applying "L" level to  $\overline{\text{RESET}}$  pin for 1 machine cycle or more when the following condition is satisfied;

- the value of supply voltage is the minimum value or more of the recommended operating conditions.
- Then when "H" level is applied to  $\overline{\text{RESET}}$  pin, software starts (Address 0 in page 0).

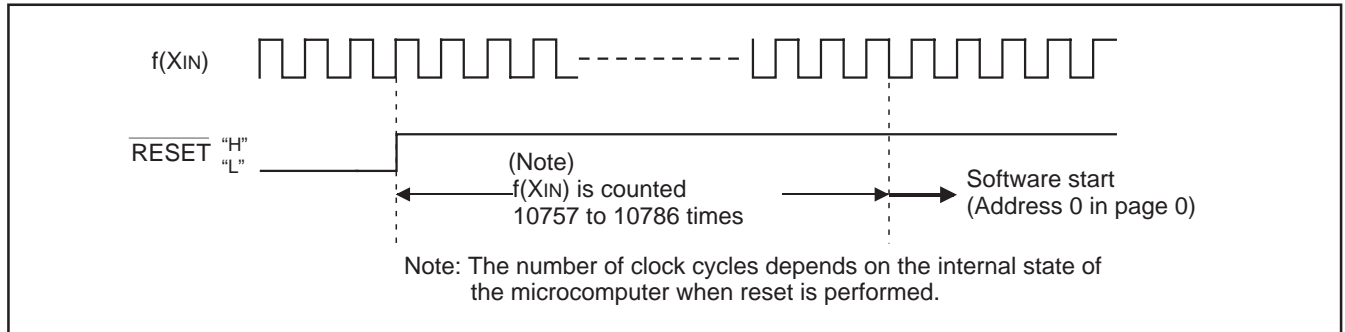


Fig. 28 Reset release timing

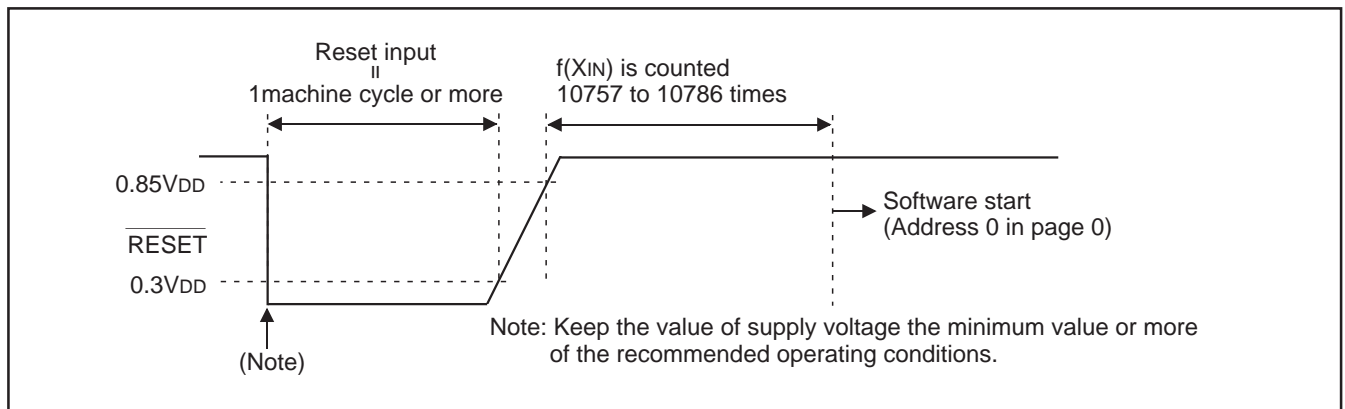


Fig. 29  $\overline{\text{RESET}}$  pin input waveform and reset operation

#### (1) Power-on reset

Reset can be automatically performed at power on (power-on reset) by the built-in power-on reset circuit. When the built-in power-on reset circuit is used, the time for the supply voltage to reach the minimum operating voltage must be set to 100

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

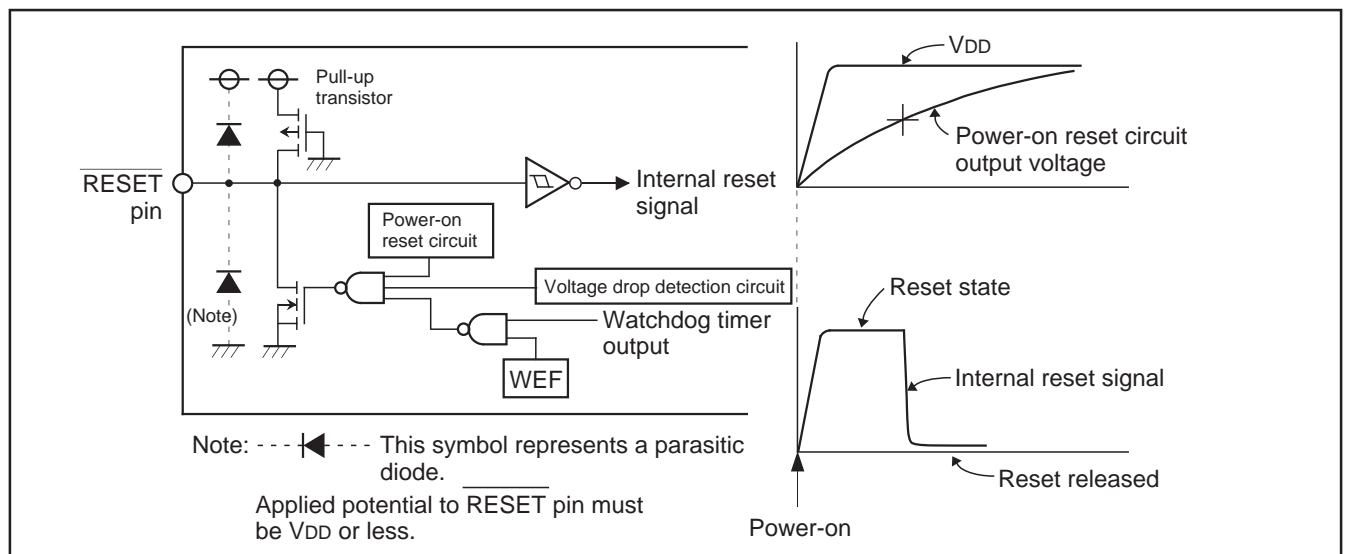


Fig. 30 Power-on reset circuit example

### (2) Internal state at reset

Table 14 shows port state at reset, and Figure 31 shows internal state at reset (they are retained after system is released from reset).

The contents of timers, registers, flags and RAM except those shown in Figure 31 are undefined, so set the initial values to them.

**Table 14 Port state at reset**

Name	Function	State
D0–D4, D5/INT	D0–D4, D5	High impedance (Note 1)
D6/XCIN, D7/XCOUT	D6, D7	
P00–P03	P00–P03	“H” (VDD) level (Note 1)
P10–P13	P10–P13	(Notes 1, 2)
P20/SEG16–P23/SEG19	P20–P23	High impedance
SEG0–SEG15	SEG0–SEG15	VLC3 (VDD) level
COM0–COM3	COM0–COM3	
CARR	CARR	“L” (VSS) level

Notes 1: Output latch is set to “1.”

2: The pull-up transistor is turned off.

• Program counter (PC) .....	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
Address 0 in page 0 is set to program counter.		
• Interrupt enable flag (INTE) .....	0	(Interrupt disabled)
• Power down flag (P) .....	0	
• External 0 interrupt request flag (EXF0) .....	0	
• Interrupt control register V1 .....	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	
• Watchdog timer flag (WDF) .....	0	
• Watchdog timer enable flag (WEF) .....	0	
• Timer control register W1 .....	0 0 0 0	(Prescaler stopped)
• Timer control register W2 .....	0 0 0 0	(Timer 1 stopped)
• Timer control register W3 .....	0 0	(Timer LC stopped)
• Clock control register MR .....	1 0 0 0	
• Carrier wave selection register C1 .....	0 1 1 1	
• Carrier wave output control register C2 .....	0	
• Carrier wave generating control flag CR .....	0	(Carrier wave output disabled)
• LCD control register L1 .....	0 0 0 0	(LCD off)
• LCD control register L2 .....	1 1 1 1	(Port P2 selected)
• Pull-up control register PU0 .....	0 0 0 0	
• General-purpose register V2 .....	0 0 0 0	
• Carry flag (CY) .....	0	
• Register A .....	0 0 0 0	
• Register B .....	0 0 0 0	
• Register D .....	X X X	
• Register E .....	X X X X X X X X	
• Data pointer X .....	0 0 0 0	
• Data pointer Y .....	0 0 0 0	
• Data pointer Z .....	X X	
• Stack pointer (SP) .....	1 1 1	

“X” represents undefined.

**Fig. 31 Internal state at reset**

# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### VOLTAGE DROP DETECTION CIRCUIT

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

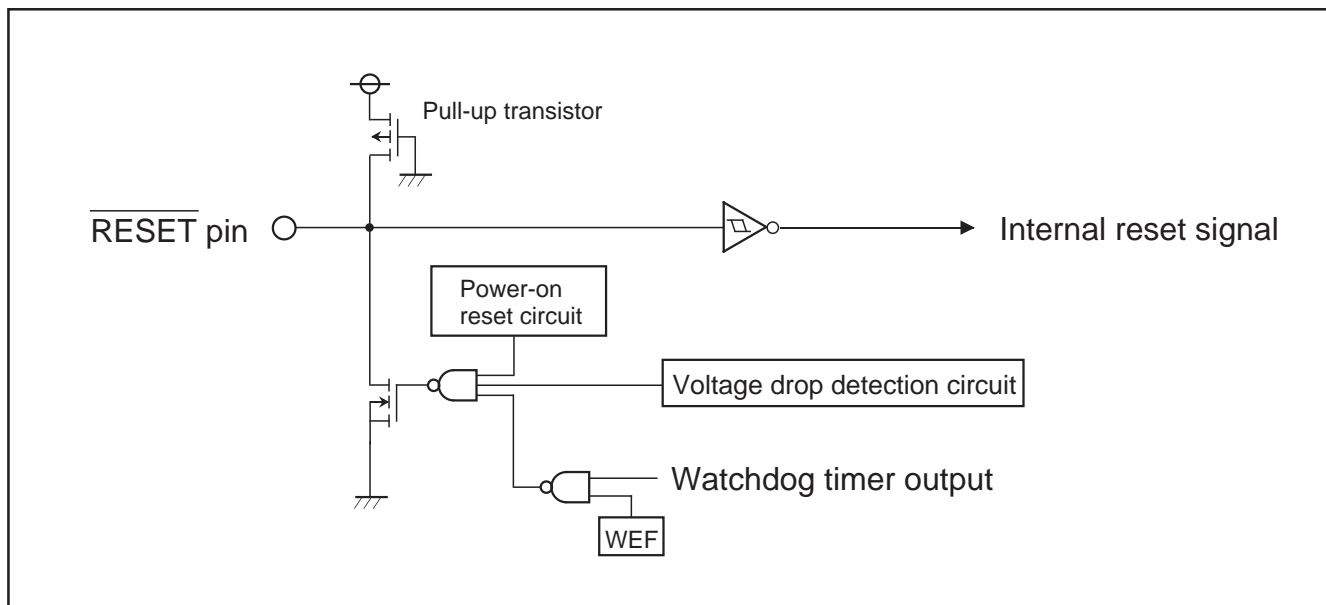


Fig. 32 Voltage drop detection reset circuit

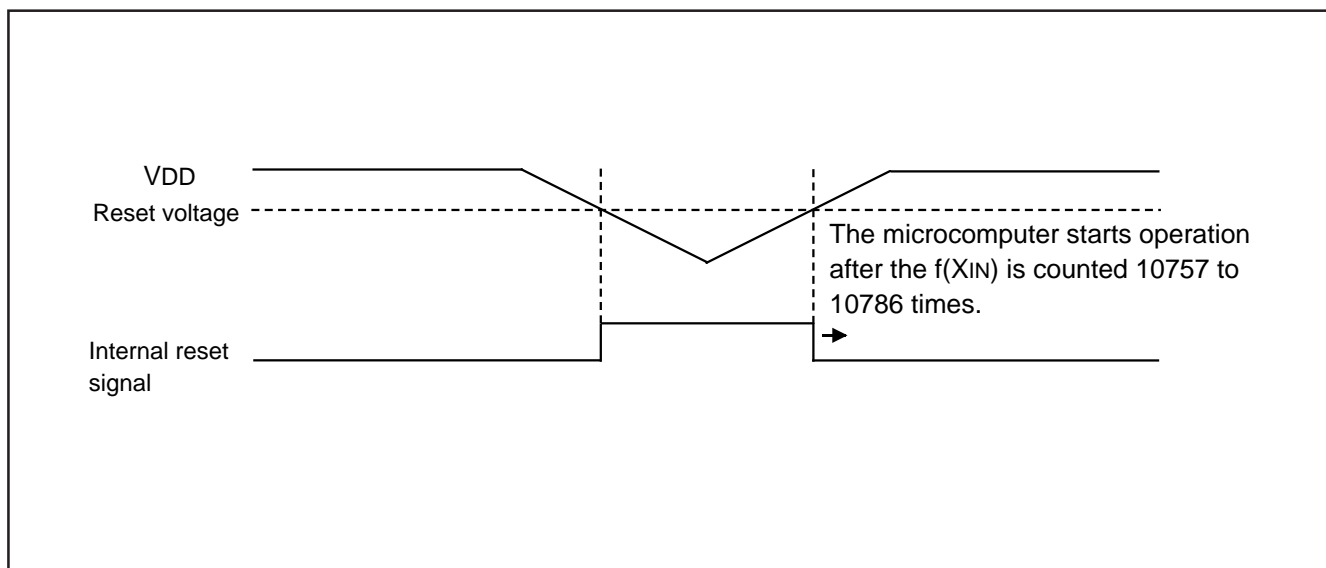


Fig. 33 Voltage drop detection circuit operation waveform

## POWER DOWN FUNCTION

The 4551 Group has 2-type power down functions.

- Clock operating mode ..... POF instruction
- RAM back-up mode ..... POF2 instruction

Power down is performed by executing each instruction. Above power down functions are different from reset in start conditions. Table 15 shows the function and states retained at power down. Figure 36 shows the state transition.

- Return from power down state ..... Warm start condition
- Return from reset state ..... Cold start condition

### (1) Clock operating mode

The following functions and states are retained.

- RAM
- Reset circuit
- XCIN–XCOUT oscillation
- LCD display
- Timer 2

### (2) RAM back-up mode

The following functions and states are retained.

- RAM
  - Reset circuit
- Unlike the clock operating mode, all oscillations stop in the RAM back-up mode.

### (3) Warm start condition

The system returns from the power down state when;

- the external wakeup signal is input or the timer 2 underflow occurs in the clock operating mode, or when;
  - the external wakeup signal is input in the RAM back-up 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;

- reset pulse is input to  $\overline{\text{RESET}}$  pin,
  - reset by watchdog timer is performed, or
  - reset by the voltage drop detection circuit is performed.
- In this case, the P flag is “0.”

Table 15 Functions and states retained at power down

Function	Power down	
	Clock operating	RAM back-up
Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) (Note 2)	X	X
Contents of RAM	O	O
Port level	O	O
Clock control register MR	O	O
Timer control register W1	X	X
Timer control registers W2, W3	O	O
Interrupt control register V1	X	X
Interrupt control register I1	O	O
Carrier wave control registers and flag (C1, C2, CR)	X	X
LCD display function	O	(Note 3)
LCD control registers L1, L2	O	O
Timer LC	O	(Note 4)
Timer 1 function	X	X
Timer 2 function	O	O
External 0 interrupt request flag (EXF0)	X	X
Timer 1 interrupt request flag (T1F)	X	X
Timer 2 interrupt request flag (T2F)	O	O
Watchdog timer flag (WDF)	O	X
Watchdog timer enable flag (WEF)	O	O
Interrupt enable flag (INTE)	X	X
General-purpose register V2	X	X

Notes 1: “O” represents that the function can be retained, and “X” represents that the function is initialized.

Registers and flags other than the above are undefined at power down, and set an initial value after returning.

2: The stack pointer (SP) points the level of the stack register and is initialized to “1112” at power down.

3: LCD is turned off.

4: The state of the timer is undefined.

# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### (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 (timer 2 or external wakeup signal) can be identified by examining the state of T2F flag.

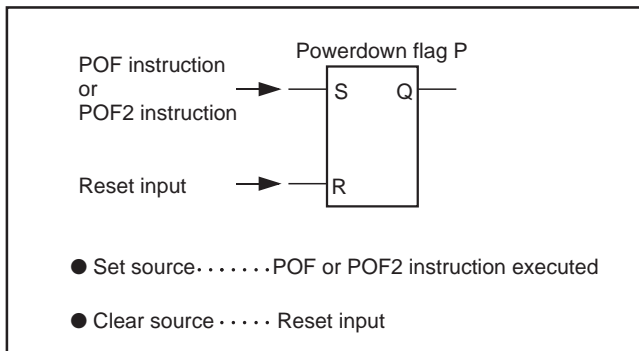


Fig. 34 Set source and clear source of the P flag

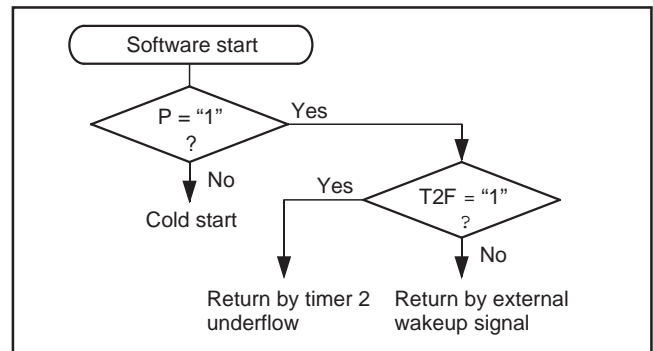


Fig. 35 Start condition identified example using the SNZP instruction

### (6) Return signal

An external wakeup signal or timer 2 interrupt request flag 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 16 shows the return condition for each return source.

### (7) Port P1 control register

#### ● Pull-up control register PU0

Register PU0 controls the ON/OFF of the port P1 pull-up transistor and the ON/OFF of the key-on wakeup function. Set the contents of this register through register A with the TPU0A instruction. In addition, the TAPU0 instruction can be used to transfer the contents of register PU0 to register A.

Table 16 Return source and return condition

Return source		Return condition	Remarks
External wakeup signal	Ports P0, P1	Returns by an external falling edge input ("H"→"L").	Port P0 shares the falling edge detection circuit with port P1. The key-on wakeup function of port P0 is always valid. The only key-on wakeup function of the port P1 bit of which the pull-up transistor is turned on is valid. Set all the port using the key-on wakeup function to "H" level before going into the power down state.
	Timer 2 interrupt request flag	Returns by timer 2 underflow and setting T2F to "1."	The timer 2 interrupt request flag (T2F) can be used only when system returns from the clock operating mode (POF instruction execution). However, if the POF and POF2 instructions are executed while the T2F = "1", its operation is recognized as the return condition and system returns from the clock operating mode.

Note: P1 pin has the pull-up transistor which can be turned on/off by software.

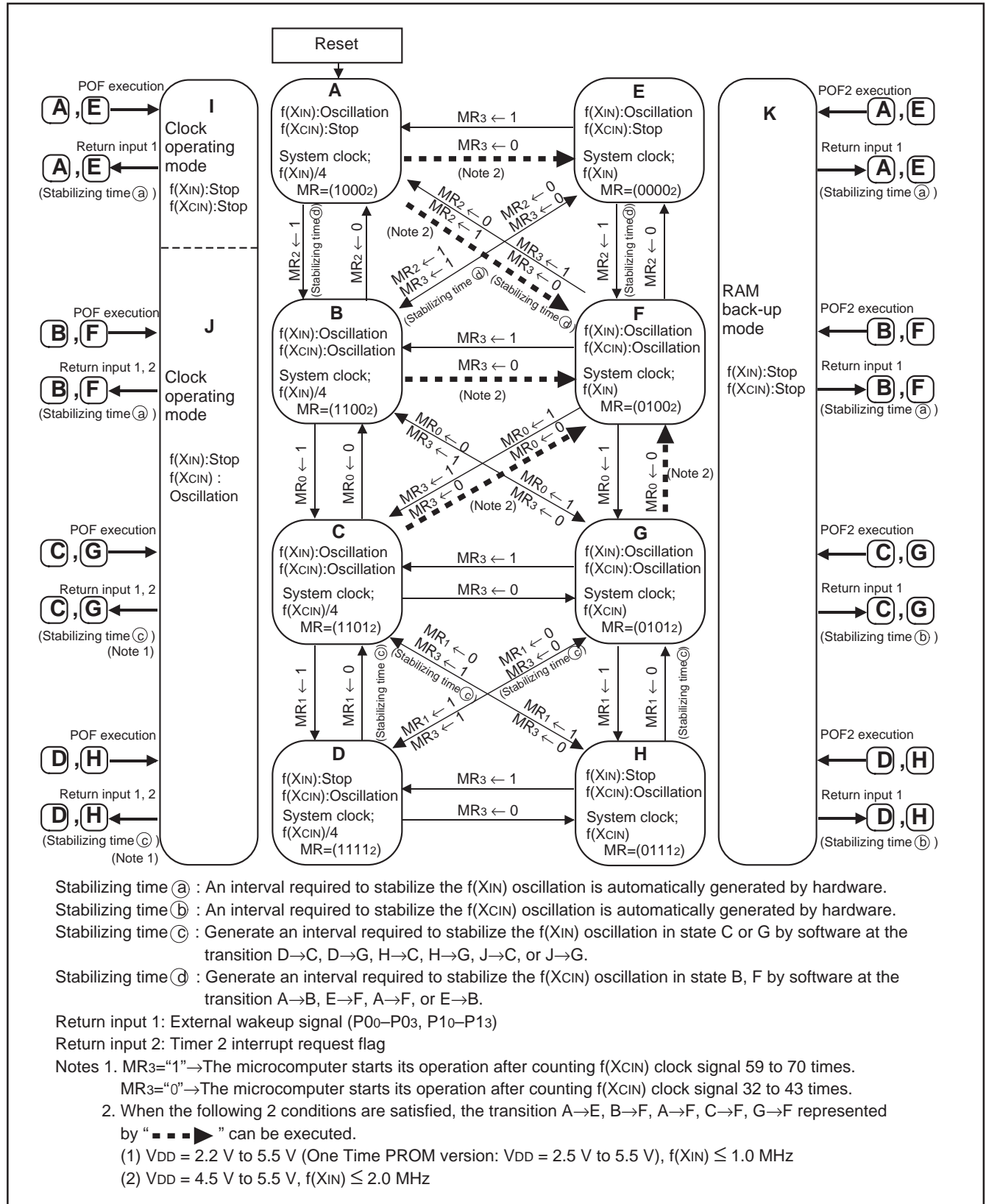
Table 17 Pull-up control register

Pull-up control register PU0		at reset : 0000 <sub>2</sub>		at power down : state retained	R/W
PU0 <sub>3</sub>	Port P1 <sub>3</sub> pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup		
		1	Pull-up transistor ON, key-on wakeup		
PU0 <sub>2</sub>	Port P1 <sub>2</sub> pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup		
		1	Pull-up transistor ON, key-on wakeup		
PU0 <sub>1</sub>	Port P1 <sub>1</sub> pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup		
		1	Pull-up transistor ON, key-on wakeup		
PU0 <sub>0</sub>	Port P1 <sub>0</sub> pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup		
		1	Pull-up transistor ON, key-on wakeup		

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

### (8) State transition

State transition is described using Figure 36.



**Fig. 36 State transition**



# HARDWARE

## FUNCTIONAL BLOCK OPERATIONS

### CLOCK CONTROL

The clock control circuit consists of the following circuits.

- Clock generating circuit
- Control circuit to stop the clock oscillation
- System clock (STCK) selection circuit
- Instruction clock (INSTCK) generating circuit
- Control circuit to return from the power down state

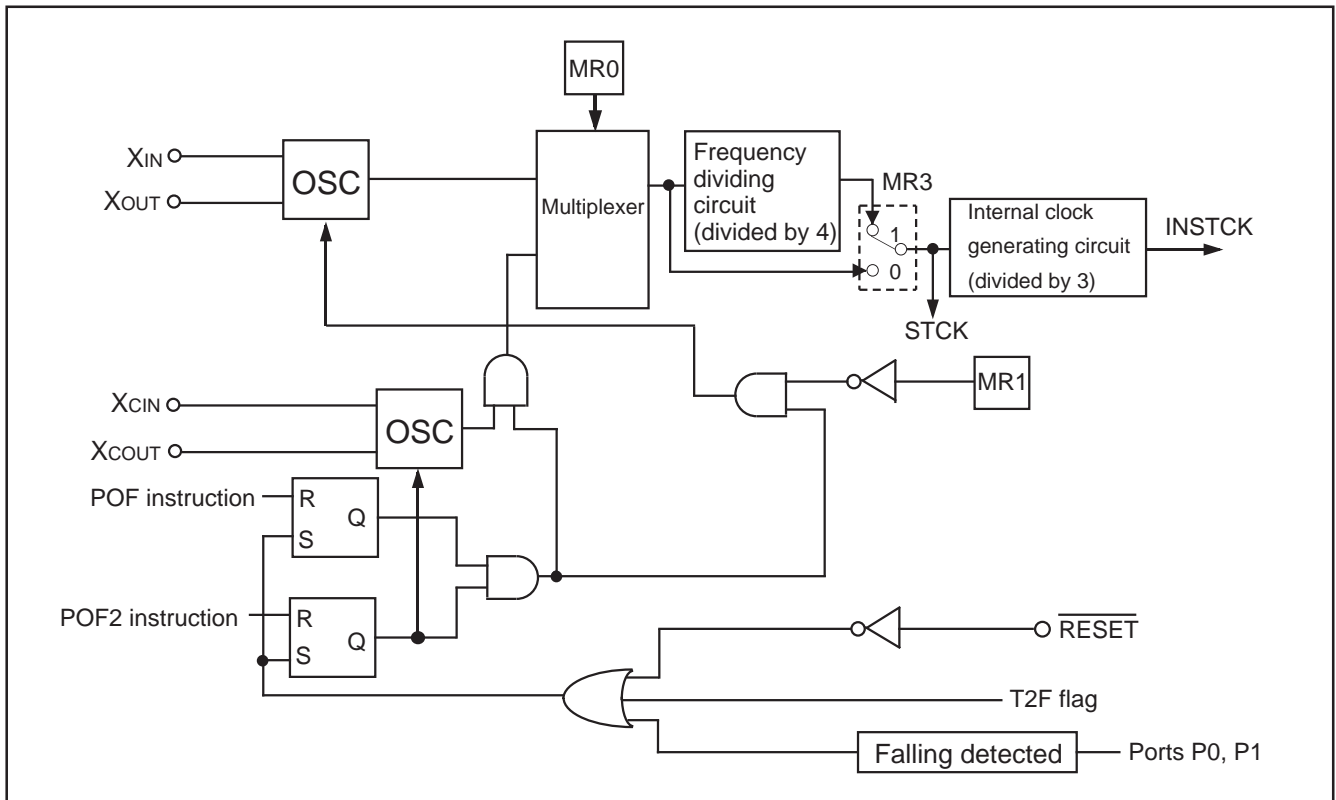


Fig. 37 Clock control circuit structure

#### (1) Clock control register

- Clock control register MR

Register MR controls the system clock. Set the contents of this register through register A with the TMRA

instruction. In addition, the TAMR instruction can be used to transfer the contents of register MR to register A.

Table 18 Clock control register

Clock control register MR		at reset : 1000 <sub>2</sub>		at power down : state retained	R/W
MR3	System clock (STCK) selection bit	0	MR0=0	$f(X_{IN})$	
			MR0=1	$f(X_{CIN})$	
		1	MR0=0	$f(X_{IN})/4$	
			MR0=1	$f(X_{CIN})/4$	
MR2	$f(X_{CIN})$ oscillation circuit control bit	0	$f(X_{CIN})$ oscillation stop, ports D <sub>6</sub> and D <sub>7</sub> selected		
		1	$f(X_{CIN})$ oscillation enabled, ports D <sub>6</sub> and D <sub>7</sub> not selected		
MR1	$f(X_{IN})$ oscillation circuit control bit	0	Oscillation enabled		
		1	Oscillation stop		
MR0	Clock selection bit	0	$f(X_{IN})$		
		1	$f(X_{CIN})$		

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

### (2) $f(X_{IN})$ clock generating circuit

Clock signal  $f(X_{IN})$  is obtained by externally connecting a ceramic resonator. Connect this external circuit to pins  $X_{IN}$  and  $X_{OUT}$  at the shortest distance. A feedback resistor is built in between pins  $X_{IN}$  and  $X_{OUT}$ .

### (3) $f(X_{CIN})$ clock generating circuit

Clock signal  $f(X_{CIN})$  is obtained by externally connecting a quartz-crystal oscillator. Connect this external circuit to pins  $X_{CIN}$  and  $X_{COUT}$  at the shortest distance. A feedback resistor is built in between pins  $X_{CIN}$  and  $X_{COUT}$ .

## ROM ORDERING METHOD

Please submit the information described below when ordering Mask ROM.

- |   |   |
|---|---|
| (1) M34551M4-XXXFP Mask ROM Order Confirmation Form ..... | 1 |
| (2) Data to be written into mask ROM                      |   |
| • EPROM (three sets containing the identical data)        |   |
| (3) Mark Specification Form .....                         | 1 |

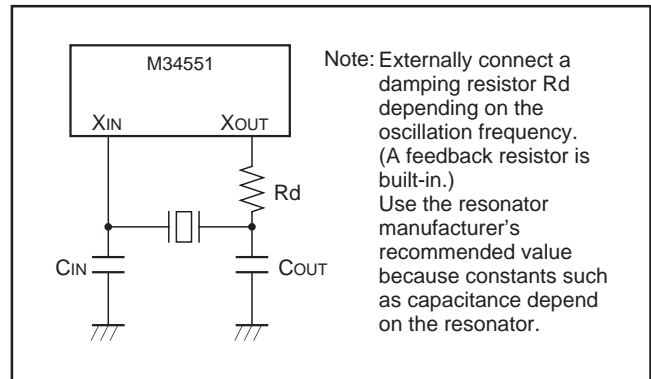


Fig. 38 Ceramic resonator external circuit

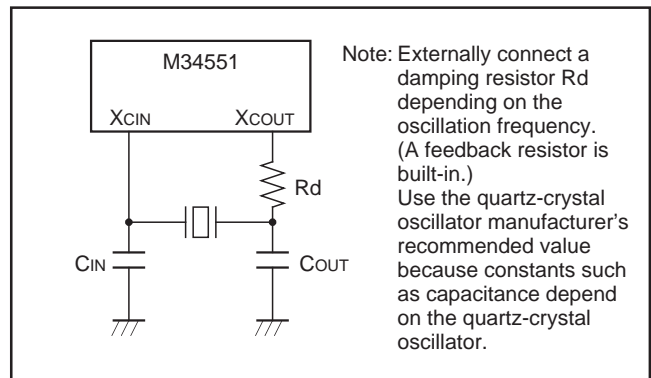


Fig. 39 Quartz-crystal oscillator external circuit

### 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  $\mu$ F) between pins VDD and VSS at the shortest distance,
- equalize its wiring in width and length, and
- use the thickest wire.

In the built-in PROM version, CNVSS pin is also used as VPP pin. Accordingly, when using this pin, connect this pin to VSS through a resistor about 5 k $\Omega$  (connect this resistor to CNVSS/VPP pin as close as possible).

#### ② Prescaler

Stop the prescaler operation to change its frequency dividing ratio.

#### ③ Count source

Stop timer 1 or timer LC counting to change its count source. When timer 2 count source changes from f(XCIN) to ORCLK (W23 = "0"  $\rightarrow$  W23 = "1"), the count value of timer 2 is initialized. However, when timer 2 count source changes from ORCLK to f(XCIN) (W23 = "1"  $\rightarrow$  W23 = "0") or the same count source is set again (W23 = "0"  $\rightarrow$  W23 = "0" or W23 = "1"  $\rightarrow$  W23 = "1"), the count value of timer 2 is not initialized.

#### ④ Timer 2

Timer 2 has the watchdog timer function (WDT). When timer 2 is used as the WDT, note that the processing to initialize the count value and the execution of the WRST instruction.

#### ⑤ Reading the count value

Stop the prescaler and then execute the TAB1 instruction to read timer 1 data.

#### ⑥ Writing to reload register R1

Write the data to reload register R1 while timer 1 is operating, avoid a timing when timer 1 underflows.

#### ⑦ Notes when using the carrier wave output auto-control function

- Execute the STCR instruction after setting the timer 1 and register C2 in order to start the carrier generating circuit operation.
- Stop the timer 1 (W20="0") after stopping the carrier generating circuit (SPCR instruction executed) while the carrier wave output is disabled in order to stop the carrier wave output auto-control operation.
- If the carrier wave output auto-control function is invalidated (C20="0") while the carrier wave output is auto-controlled, the output of port CARR retains the state when the auto-control is invalidated regardless of timer 1 underflow. This state is released by timer 1 stop (W20="0").  
When the carrier wave output auto-control function is validated (C20="1") again after it is invalidated (C20="0"), the auto-control of carrier wave output is started again when the next timer 1 underflow occurs. However, when the carrier wave output auto-control bit is changed during timer 1 underflow, the error-operation may occur.

- Use the carrier wave or the carrier wave divided by 2 as the timer 1 count source when the carrier wave output auto-control function is selected.  
If the ORCLK is used as the count source, a hazard may occur in port CARR output because ORCLK is not synchronized with the carrier wave.
- When "no carrier wave" is selected with register C1 ((C13C12C11C10) = (0101), (1101)), the enable/disable of the carrier wave output cannot be controlled by the carrier wave output auto-control function.

#### ⑧ D5/INT pin

When the interrupt valid waveform of D5/INT pin is changed with the bit 2 of register I1 in software, be careful about the following notes.

- Clear the bit 0 of register V1 to "0" and then change the interrupt valid waveform of D5/INT pin with the bit 2 of register I1 (refer to Figure 40①).
- Clear the bit 2 of register I1 to "0" and execute the SNZ0 instruction to clear the EXF0 flag after executing at least one instruction (refer to Figure 40②). Depending on the input state of the D5/INT pin, the external 0 interrupt request flag (EXF0) may be set to "1" when the interrupt valid waveform is changed.

```

:
LA      4      ; (XXX02)
TV1A    ; The SNZ0 instruction is valid ①
LA      4
TI1A    ; Change of the interrupt valid waveform
NOP                                           ②
SNZ0    ; The SNZ0 instruction is executed
NOP
:
X : this bit is not related to the setting of INT.
```

**Fig. 40 External 0 interrupt program example**

#### ⑨ One Time PROM version

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

#### ⑩ Multifunction

Note that the port D5 output function can be used even when INT function is selected.

#### ⑪ Power down instruction (POF instruction, POF2 instruction)

Execute the POF or POF2 instruction immediately after executing the EPOF instruction to enter the power down state. Note that system cannot enter the power down state when executing only the POF or POF2 instruction.  
Be sure to disable interrupts by executing the DI instruction before executing the EPOF instruction.

#### ⑫ Program counter

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

## SYMBOL

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

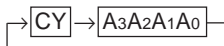
Symbol	Contents	Symbol	Contents
A	Register A (4 bits)	WDF	Watchdog timer flag
B	Register B (4 bits)	INTE	Interrupt enable flag
DR	Register D (3 bits)	EXF0	External 0 interrupt request flag
E	Register E (8 bits)	P	Power down flag
V1	Interrupt control register V1 (4 bits)	D	Port D (8 bits)
V2	General-purpose register V2 (4 bits)	P0	Port P0 (4 bits)
I1	Interrupt control register I1 (4 bits)	P1	Port P1 (4 bits)
W1	Timer control register W1 (4 bits)	P2	Port P2 (4 bits)
W2	Timer control register W2 (4 bits)	x	Hexadecimal variable
W3	Timer control register W3 (2 bits)	y	Hexadecimal variable
C1	Carrier wave selection register C1 (4 bits)	z	Hexadecimal variable
C2	Carrier wave output control register C2 (1 bit)	p	Hexadecimal variable
CR	Carrier wave generating control flag	n	Hexadecimal constant which represents the immediate value
L1	LCD control register L1	i	Hexadecimal constant which represents the immediate value
L2	LCD control register L2	j	Hexadecimal constant which represents the immediate value
PU0	Pull-up control register PU0 (4 bits)	A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	Binary notation of hexadecimal variable A (same for others)
MR	Clock control register MR (4 bits)	←	Direction of data movement
X	Register X (4 bits)	↔	Data exchange between a register and memory
Y	Register Y (4 bits)	?	Decision of state shown before “?”
Z	Register Z (2 bits)	( )	Contents of registers and memories
DP	Data pointer (10 bits) (It consists of registers X, Y, and Z)	—	Negate, Flag unchanged after executing instruction
PC	Program counter (14 bits)	M(DP)	RAM address pointed by the data pointer
PC <sub>H</sub>	High-order 7 bits of program counter	a	Label indicating address a <sub>6</sub> a <sub>5</sub> a <sub>4</sub> a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub>
PC <sub>L</sub>	Low-order 7 bits of program counter	p, a	Label indicating address a <sub>6</sub> a <sub>5</sub> a <sub>4</sub> a <sub>3</sub> a <sub>2</sub> a <sub>1</sub> a <sub>0</sub> in page p <sub>5</sub> p <sub>4</sub> p <sub>3</sub> p <sub>2</sub> p <sub>1</sub> p <sub>0</sub>
SK	Stack register (14 bits X 8)	C	Hex. C + Hex. number x (also same for others)
SP	Stack pointer (3 bits)	+	
CY	Carry flag	x	
R1	Timer 1 reload register		
R2	Timer 2 reload register		
RLC	Timer LC reload register		
STCK	System clock		
INSTK	Instruction clock		
T1	Timer 1		
T2	Timer 2		
TLC	Timer LC		
T1F	Timer 1 interrupt request flag		
T2F	Timer 2 interrupt request flag		

Note : The 4551 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.

# HARDWARE

## LIST OF INSTRUCTION FUNCTION

### LIST OF INSTRUCTION FUNCTION

Grouping	Mnemonic	Function	Grouping	Mnemonic	Function	Grouping	Mnemonic	Function
Register to register transfer	TAB	$(A) \leftarrow (B)$	RAM to register transfer	XAMI j	$(A) \leftarrow \rightarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15 $(Y) \leftarrow (Y) + 1$	Bit operation	SB j	$(Mj(DP)) \leftarrow 1$ j = 0 to 3
	TBA	$(B) \leftarrow (A)$		TMA j	$(M(DP)) \leftarrow (A)$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15		RB j	$(Mj(DP)) \leftarrow 0$ j = 0 to 3
	TAY	$(A) \leftarrow (Y)$					SZB j	$(Mj(DP)) = 0 ?$ j = 0 to 3
	TYA	$(Y) \leftarrow (A)$				Comparison operation	SEAM	$(A) = (M(DP)) ?$
	TEAB	$(E7-E4) \leftarrow (B)$ $(E3-E0) \leftarrow (A)$		LA n	$(A) \leftarrow n$ n = 0 to 15		SEA n	$(A) = n ?$ n = 0 to 15
	TABE	$(B) \leftarrow (E7-E4)$ $(A) \leftarrow (E3-E0)$		TABP p	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ $(PCL) \leftarrow (DR2-DR0,$ A3-A0) $(B) \leftarrow (ROM(PC))_{7 \text{ to } 4}$ $(A) \leftarrow (ROM(PC))_{3 \text{ to } 0}$ $(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$	Branch operation	B a	$(PCL) \leftarrow a6-a0$
	TDA	$(DR2-DR0) \leftarrow (A2-A0)$					BL p, a	$(PCH) \leftarrow p$ $(PCL) \leftarrow a6-a0$
	TAD	$(A2-A0) \leftarrow (DR2-DR0)$ $(A3) \leftarrow 0$		AM	$(A) \leftarrow (A) + (M(DP))$		BLA p	$(PCH) \leftarrow p$ $(PCL) \leftarrow (DR2-DR0,$ A3-A0)
	TAZ	$(A1, A0) \leftarrow (Z1, Z0)$ $(A3, A2) \leftarrow 0$	Arithmetic operation	AMC	$(A) \leftarrow (A) + (M(DP))$ + (CY) $(CY) \leftarrow \text{Carry}$	Subroutine operation	BM a	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow 2$ $(PCL) \leftarrow a6-a0$
	TAX	$(A) \leftarrow (X)$		A n	$(A) \leftarrow (A) + n$ n = 0 to 15		BML p, a	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ $(PCL) \leftarrow a6-a0$
	TASP	$(A2-A0) \leftarrow (SP2-SP0)$ $(A3) \leftarrow 0$		AND	$(A) \leftarrow (A)AND(M(DP))$		BMLA p	$(SP) \leftarrow (SP) + 1$ $(SK(SP)) \leftarrow (PC)$ $(PCH) \leftarrow p$ $(PCL) \leftarrow (DR2-DR0,$ A3-A0)
RAM addresses	LXY x, y	$(X) \leftarrow x, x = 0 \text{ to } 15$ $(Y) \leftarrow y, y = 0 \text{ to } 15$		OR	$(A) \leftarrow (A)OR(M(DP))$	Return operation	RTI	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$
	LZ z	$(Z) \leftarrow z, z = 0 \text{ to } 3$		SC	$(CY) \leftarrow 1$		RT	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$
	INY	$(Y) \leftarrow (Y) + 1$		RC	$(CY) \leftarrow 0$		RTS	$(PC) \leftarrow (SK(SP))$ $(SP) \leftarrow (SP) - 1$
	DEY	$(Y) \leftarrow (Y) - 1$		SZC	$(CY) = 0 ?$			
RAM to register transfer	TAM j	$(A) \leftarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15		CMA	$(A) \leftarrow \bar{A}$			
	XAM j	$(A) \leftarrow \rightarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15		RAR				
	XAMD j	$(A) \leftarrow \rightarrow (M(DP))$ $(X) \leftarrow (X)EXOR(j)$ j = 0 to 15 $(Y) \leftarrow (Y) - 1$						

## LIST OF INSTRUCTION FUNCTION (CONTINUED)

Grouping	Mnemonic	Function	Grouping	Mnemonic	Function	Grouping	Mnemonic	Function
Interrupt operation	DI	(INTE) $\leftarrow$ 0	Timer operation	TLCA	(TLC) $\leftarrow$ (A) (RLC) $\leftarrow$ (A)	Carrier wave generating operation	TC1A	(C1) $\leftarrow$ (A)
	EI	(INTE) $\leftarrow$ 1		SNZT1	(T1F) = 1 ? After skipping the next instruction, (T1F) $\leftarrow$ 0		STCR	Carrier wave generating start
	SNZ0	(EXF0) = 1 ? After skipping the next instruction, (EXF0) $\leftarrow$ 0		SNZT2	(T2F) = 1 ? After skipping the next instruction, (T2F) $\leftarrow$ 0		SPCR	Carrier wave generating stop
	SNZI0	I12 = 1 : (INT0) = "H" ? I12 = 0 : (INT0) = "L" ?					TC2A	(C20) $\leftarrow$ (A0)
	TAV1	(A) $\leftarrow$ (V1)	Input/Output operation	IAP0	(A) $\leftarrow$ (P0)	Other operation	NOP	(PC) $\leftarrow$ (PC) + 1
	TV1A	(V1) $\leftarrow$ (A)		OP0A	(P0) $\leftarrow$ (A)		POF	Transition to clock operating mode
	TAI1	(A) $\leftarrow$ (I1)		IAP1	(A) $\leftarrow$ (P1)		POF2	Transition to RAM back-up mode
	TI1A	(I1) $\leftarrow$ (A)		OP1A	(P1) $\leftarrow$ (A)		EPOF	Power down instruction (POF, POF2) valid
Timer operation	TAW1	(A) $\leftarrow$ (W1)		IAP2	(A) $\leftarrow$ (P2)		SNZP	(P) = 1 ?
	TW1A	(W1) $\leftarrow$ (A)		CLD	(D) $\leftarrow$ 1		WRST	(WDF) $\leftarrow$ 0, (WEF) $\leftarrow$ 1
	TAW2	(A) $\leftarrow$ (W2)		RD	(D(Y)) $\leftarrow$ 0 (Y) = 0 to 7		TAMR	(A) $\leftarrow$ (MR)
	TW2A	(W2) $\leftarrow$ (A)		SD	(D(Y)) $\leftarrow$ 1 (Y) = 0 to 7		TMRA	(MR) $\leftarrow$ (A)
	TAW3	(A1, A0) $\leftarrow$ (W31, W30)		TPU0A	(PU0) $\leftarrow$ (A)		TAV2	(A) $\leftarrow$ (V2)
	TW3A	(W31, W30) $\leftarrow$ (A1, A0)		TAPU0	(A) $\leftarrow$ (PU0)		TV2A	(V2) $\leftarrow$ (A)
	TAB1	(B) $\leftarrow$ (T17–T14) (A) $\leftarrow$ (T13–T10)						
	T1AB	at timer 1 stop (W20=0) (R17–R14) $\leftarrow$ (B) (T17–T14) $\leftarrow$ (B) (R13–R10) $\leftarrow$ (A) (T13–T10) $\leftarrow$ (A) At timer 1 operating (W20=1), (R17–R14) $\leftarrow$ (B) (R13–R10) $\leftarrow$ (A)	LCD control operation	TL1A	(L1) $\leftarrow$ (A)			
				TAL1	(A) $\leftarrow$ (L1)			
				TL2A	(L2) $\leftarrow$ (A)			

# HARDWARE

## INSTRUCTION CODE TABLE

INSTRUCTION CODE TABLE

D3-D0	Hex. notation	D9-D4																		010000   010111	011000   011111
		00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10-17	18-1F		
0000	0	NOP	BLA	SZB 0	BMLA	—	TASP	A 0	LA 0	TABP 0	TABP 16	TABP 32*	TABP 48*	BML	BML	BL	BL	BM	B		
0001	1	—	CLD	SZB 1	—	—	TAD	A 1	LA 1	TABP 1	TABP 17	TABP 33*	TABP 49*	BML	BML	BL	BL	BM	B		
0010	2	POF	—	SZB 2	—	—	TAX	A 2	LA 2	TABP 2	TABP 18	TABP 34*	TABP 50*	BML	BML	BL	BL	BM	B		
0011	3	SNZP	INY	SZB 3	—	—	TAZ	A 3	LA 3	TABP 3	TABP 19	TABP 35*	TABP 51*	BML	BML	BL	BL	BM	B		
0100	4	DI	RD	—	—	RT	TAV1	A 4	LA 4	TABP 4	TABP 20	TABP 36*	TABP 52*	BML	BML	BL	BL	BM	B		
0101	5	EI	SD	SEAn	—	RTS	TAV2	A 5	LA 5	TABP 5	TABP 21	TABP 37*	TABP 53*	BML	BML	BL	BL	BM	B		
0110	6	RC	—	SEAM	—	RTI	—	A 6	LA 6	TABP 6	TABP 22	TABP 38*	TABP 54*	BML	BML	BL	BL	BM	B		
0111	7	SC	DEY	—	—	—	—	A 7	LA 7	TABP 7	TABP 23	TABP 39*	TABP 55*	BML	BML	BL	BL	BM	B		
1000	8	POF2	AND	—	SNZ0	LZ 0	—	A 8	LA 8	TABP 8	TABP 24	TABP 40*	TABP 56*	BML	BML	BL	BL	BM	B		
1001	9	—	OR	TDA	—	LZ 1	—	A 9	LA 9	TABP 9	TABP 25	TABP 41*	TABP 57*	BML	BML	BL	BL	BM	B		
1010	A	AM	TEAB	TABE	SNZI0	LZ 2	—	A 10	LA 10	TABP 10	TABP 26	TABP 42*	TABP 58*	BML	BML	BL	BL	BM	B		
1011	B	AMC	—	—	—	LZ 3	EPOF	A 11	LA 11	TABP 11	TABP 27	TABP 43*	TABP 59*	BML	BML	BL	BL	BM	B		
1100	C	TYA	CMA	—	—	RB 0	SB 0	A 12	LA 12	TABP 12	TABP 28	TABP 44*	TABP 60*	BML	BML	BL	BL	BM	B		
1101	D	—	RAR	—	—	RB 1	SB 1	A 13	LA 13	TABP 13	TABP 29	TABP 45*	TABP 61*	BML	BML	BL	BL	BM	B		
1110	E	TBA	TAB	—	TV2A	RB 2	SB 2	A 14	LA 14	TABP 14	TABP 30	TABP 46*	TABP 62*	BML	BML	BL	BL	BM	B		
1111	F	—	TAY	SZC	TV1A	RB 3	SB 3	A 15	LA 15	TABP 15	TABP 31	TABP 47*	TABP 63*	BML	BML	BL	BL	BM	B		

The above table shows the relationship between machine language codes and machine language instructions. D 3–D0 show the low-order 4 bits of the machine language code, and D 9–D4 show the high-order 6 bits of the machine language code. The hexadecimal representation of the code is also provided. There are one-word instructions and two-word instructions, but only the first word of each instruction is shown. Do not use code marked “—.”

The codes for the second word of a two-word instruction are described below. \* cannot be used at M34551M4.

The second word	
BL	1 0 p a a a a a a a
BML	1 0 p a a a a a a a
BLA	1 0 p p 0 0 p p p p
BMLA	1 0 p p 0 0 p p p p
SEA	0 0 0 1 1 1 n n n n

INSTRUCTION CODE TABLE (CONTINUED)

D3-D0	D9-D4	100000	100001	100010	100011	100100	100101	100110	100111	101000	101001	101010	101011	101100	101101	101110	101111	110000 111111
	Hex. notation	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30-3F
0000	0	–	TW3A	OP0A	T1AB	–	–	IAP0	TAB1	SNZT1	–	WRST	TMA 0	TAM 0	XAM 0	XAMI 0	XAMD 0	LXY
0001	1	–	–	OP1A	–	–	–	IAP1	–	SNZT2	–	–	TMA 1	TAM 1	XAM 1	XAMI 1	XAMD 1	LXY
0010	2	–	–	–	–	–	TAMR	IAP2	–	–	–	–	TMA 2	TAM 2	XAM 2	XAMI 2	XAMD 2	LXY
0011	3	–	–	–	–	–	TAI1	–	–	–	–	–	TMA 3	TAM 3	XAM 3	XAMI 3	XAMD 3	LXY
0100	4	–	–	–	–	–	–	–	–	–	–	–	TMA 4	TAM 4	XAM 4	XAMI 4	XAMD 4	LXY
0101	5	–	–	–	–	–	–	–	–	–	–	–	TMA 5	TAM 5	XAM 5	XAMI 5	XAMD 5	LXY
0110	6	–	TMRA	–	–	–	–	–	–	–	–	–	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	–	–	–	–	–	–	–	–	–	STCR	TC1A	TMA 8	TAM 8	XAM 8	XAMI 8	XAMD 8	LXY
1001	9	–	–	–	–	–	–	–	–	–	SPCR	TC2A	TMA 9	TAM 9	XAM 9	XAMI 9	XAMD 9	LXY
1010	A	TL1A	–	–	–	TAL1	–	–	–	–	–	–	TMA 10	TAM 10	XAM 10	XAMI 10	XAMD 10	LXY
1011	B	TL2A	–	–	–	TAW1	–	–	–	–	–	–	TMA 11	TAM 11	XAM 11	XAMI 11	XAMD 11	LXY
1100	C	–	–	–	–	TAW2	–	–	–	–	–	–	TMA 12	TAM 12	XAM 12	XAMI 12	XAMD 12	LXY
1101	D	TLCA	–	TPU0A	–	TAW3	–	–	–	–	–	–	TMA 13	TAM 13	XAM 13	XAMI 13	XAMD 13	LXY
1110	E	TW1A	–	–	–	–	–	–	–	–	–	–	TMA 14	TAM 14	XAM 14	XAMI 14	XAMD 14	LXY
1111	F	TW2A	–	–	–	–	–	–	–	–	–	–	TMA 15	TAM 15	XAM 15	XAMI 15	XAMD 15	LXY

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

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

The second word	
BL	1 0 p a a a a a a a
BML	1 0 p a a a a a a a
BLA	1 0 p p 0 0 p p p p
BMLA	1 0 p p 0 0 p p p p
SEA	0 0 0 1 1 1 n n n n



# HARDWARE

## MACHINE INSTRUCTIONS

### MACHINE INSTRUCTIONS

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D <sub>9</sub>	D <sub>8</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	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 <sub>7</sub> –E <sub>4</sub> ) ← (B) (E <sub>3</sub> –E <sub>0</sub> ) ← (A)
	TABE	0	0	0	0	1	0	1	0	1	0	0 2 A	1	1	(B) ← (E <sub>7</sub> –E <sub>4</sub> ) (A) ← (E <sub>3</sub> –E <sub>0</sub> )
	TDA	0	0	0	0	1	0	1	0	0	1	0 2 9	1	1	(DR <sub>2</sub> –DR <sub>0</sub> ) ← (A <sub>2</sub> –A <sub>0</sub> )
	TAD	0	0	0	1	0	1	0	0	0	1	0 5 1	1	1	(A <sub>2</sub> –A <sub>0</sub> ) ← (DR <sub>2</sub> –DR <sub>0</sub> ) (A <sub>3</sub> ) ← 0
	TAZ	0	0	0	1	0	1	0	0	1	1	0 5 3	1	1	(A <sub>1</sub> , A <sub>0</sub> ) ← (Z <sub>1</sub> , Z <sub>0</sub> ) (A <sub>3</sub> , A <sub>2</sub> ) ← 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 <sub>2</sub> –A <sub>0</sub> ) ← (SP <sub>2</sub> –SP <sub>0</sub> ) (A <sub>3</sub> ) ← 0
RAM addresses	LXY x, y	1	1	x <sub>3</sub>	x <sub>2</sub>	x <sub>1</sub>	x <sub>0</sub>	y <sub>3</sub>	y <sub>2</sub>	y <sub>1</sub>	y <sub>0</sub>	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 <sub>1</sub>	z <sub>0</sub>	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

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 registers A and B to register E.
–	–	Transfers the contents of register E to registers A and B.
–	–	Transfers the contents of register A to register D.
–	–	Transfers the contents of register D to register A.
–	–	Transfers the contents of register Z to register A.
–	–	Transfers the contents of register X to register A.
–	–	Transfers the contents of stack pointer (SP) to register A.
Continuous description	–	<p>Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y.</p> <p>When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped.</p>
–	–	Loads the value z in the immediate field to register Z.
(Y) = 0	–	Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped.
(Y) = 15	–	Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped.

# HARDWARE

## MACHINE INSTRUCTIONS

### MACHINE INSTRUCTIONS (CONTINUED)

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D <sub>9</sub>	D <sub>8</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Hexadecimal notation			
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
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 <sub>5</sub>	p <sub>4</sub>	p <sub>3</sub>	p <sub>2</sub>	p <sub>1</sub>	p <sub>0</sub>	0 8 p +p	1	3	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC <sub>H</sub> ) ← p (PC <sub>L</sub> ) ← (DR <sub>2</sub> –DR <sub>0</sub> , A <sub>3</sub> –A <sub>0</sub> ) (B) ← (ROM(PC)) <sub>7 to 4</sub> (A) ← (ROM(PC)) <sub>3 to 0</sub> (PC) ← (SK(SP)) (SP) ← (SP) – 1 (Note)

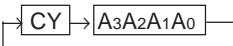
Note: p is 0 to 31 for M34551M4 and p is 0 to 63 for M34551M8 and M34551E8.

Skip condition	Carry flag CY	Detailed 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.
–	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
(Y) = 15	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped.
(Y) = 0	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped.
–	–	After transferring the contents of register A to M(DP), an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
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 <sub>2</sub> DR <sub>1</sub> DR <sub>0</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub> ) <sub>2</sub> specified by registers A and D in page p. When this instruction is executed, 1 stage of stack register is used.

# HARDWARE

## MACHINE INSTRUCTIONS

### MACHINE INSTRUCTIONS (CONTINUED)

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D <sub>9</sub>	D <sub>8</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Hexadecimal notation			
Arithmetic operation	AM	0	0	0	0	0	0	1	0	1	0	0 0 A	1	1	$(A) \leftarrow (A) + (M(DP))$
	AMC	0	0	0	0	0	0	1	0	1	1	0 0 B	1	1	$(A) \leftarrow (A) + (M(DP)) + (CY)$ $(CY) \leftarrow \text{Carry}$
	A n	0	0	0	1	1	0	n	n	n	n	0 6 n	1	1	$(A) \leftarrow (A) + n$ $n = 0 \text{ to } 15$
	AND	0	0	0	0	0	1	1	0	0	0	0 1 8	1	1	$(A) \leftarrow (A) \text{AND} (M(DP))$
	OR	0	0	0	0	0	1	1	0	0	1	0 1 9	1	1	$(A) \leftarrow (A) \text{OR} (M(DP))$
	SC	0	0	0	0	0	0	0	1	1	1	0 0 7	1	1	$(CY) \leftarrow 1$
	RC	0	0	0	0	0	0	0	1	1	0	0 0 6	1	1	$(CY) \leftarrow 0$
	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	$(A) \leftarrow \overline{(A)}$
	RAR	0	0	0	0	0	1	1	1	0	1	0 1 D	1	1	
Bit operation	SB j	0	0	0	1	0	1	1	1	j	j	0 5 C +j	1	1	$(M_j(DP)) \leftarrow 1$ $j = 0 \text{ to } 3$
	RB j	0	0	0	1	0	0	1	1	j	j	0 4 C +j	1	1	$(M_j(DP)) \leftarrow 0$ $j = 0 \text{ to } 3$
	SZB j	0	0	0	0	1	0	0	0	j	j	0 2 j	1	1	$(M_j(DP)) = 0 ?$ $j = 0 \text{ to } 3$
Comparison operation	SEAM	0	0	0	0	1	0	0	1	1	0	0 2 6	1	1	$(A) = (M(DP)) ?$
	SEA n	0	0	0	0	1	0	0	1	0	1	0 2 5	2	2	$(A) = n ?$ $n = 0 \text{ to } 15$
		0	0	0	1	1	1	n	n	n	n	0 7 n			

Skip condition	Carry flag CY	Detailed description
–	–	Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged.
–	0/1	Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY.
Overflow = 0	–	Adds the value n in the immediate field to register A. The contents of carry flag CY remains unchanged. Skips the next instruction when there is no overflow as the result of operation.
–	–	Performs the AND operation between the contents of register A and the contents of M(DP), and stores the result in register A.
–	–	Performs the OR operation between the contents of register A and the contents of M(DP), and stores the result in register A.
–	1	Sets carry flag CY to “1.”
–	0	Clears carry flag CY to “0.”
(CY) = 0	–	Skips the next instruction when the contents of carry flag CY is “0.”
–	–	Stores the one’s complement for register A’s contents in register A.
–	0/1	Rotates the contents of register A including the contents of carry flag CY to the right by 1 bit.
–	–	Sets the contents of bit j (bit specified by the value j in the immediate field) of M(DP) to “1.”
–	–	Clears the contents of bit j (bit specified by the value j in the immediate field) of M(DP) to “0.”
(M <sub>j</sub> (DP)) = 0 j = 0 to 3	–	Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is “0.”
(A) = (M(DP))	–	Skips the next instruction when the contents of register A is equal to the contents of M(DP).
(A) = n	–	Skips the next instruction when the contents of register A is equal to the value n in the immediate field.

# HARDWARE

## MACHINE INSTRUCTIONS

### MACHINE INSTRUCTIONS (CONTINUED)

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D <sub>9</sub>	D <sub>8</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Hexadecimal notation			
Branch operation	B a	0	1	1	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	1 8 a +a	1	1	(PC <sub>L</sub> ) ← a <sub>6</sub> –a <sub>0</sub>
	BL p, a	0	0	1	1	1	p <sub>4</sub>	p <sub>3</sub>	p <sub>2</sub>	p <sub>1</sub>	p <sub>0</sub>	0 E p +p	2	2	(PC <sub>H</sub> ) ← p (PC <sub>L</sub> ) ← a <sub>6</sub> –a <sub>0</sub> (Note)
		1	0	p <sub>5</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	2 p a +a			
	BLA p	0	0	0	0	0	1	0	0	0	0	0 1 0	2	2	(PC <sub>H</sub> ) ← p (PC <sub>L</sub> ) ← (DR <sub>2</sub> –DR <sub>0</sub> , A <sub>3</sub> –A <sub>0</sub> ) (Note)
		1	0	p <sub>5</sub>	p <sub>4</sub>	0	0	p <sub>3</sub>	p <sub>2</sub>	p <sub>1</sub>	p <sub>0</sub>	2 p p			
Subroutine operation	BM a	0	1	0	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	1 a a	1	1	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC <sub>H</sub> ) ← 2 (PC <sub>L</sub> ) ← a <sub>6</sub> –a <sub>0</sub>
	BML p, a	0	0	1	1	0	p <sub>4</sub>	p <sub>3</sub>	p <sub>2</sub>	p <sub>1</sub>	p <sub>0</sub>	0 C p +p	2	2	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC <sub>H</sub> ) ← p (PC <sub>L</sub> ) ← a <sub>6</sub> –a <sub>0</sub> (Note)
		1	0	p <sub>5</sub>	a <sub>6</sub>	a <sub>5</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	2 p a +a			
	BMLA p	0	0	0	0	1	1	0	0	0	0	0 3 0	2	2	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC <sub>H</sub> ) ← p (PC <sub>L</sub> ) ← (DR <sub>2</sub> –DR <sub>0</sub> , A <sub>3</sub> –A <sub>0</sub> ) (Note)
		1	0	p <sub>5</sub>	p <sub>4</sub>	0	0	p <sub>3</sub>	p <sub>2</sub>	p <sub>1</sub>	p <sub>0</sub>	2 p p			
Return operation	RTI	0	0	0	1	0	0	0	1	1	0	0 4 6	1	1	(PC) ← (SK(SP)) (SP) ← (SP) – 1
	RT	0	0	0	1	0	0	0	1	0	0	0 4 4	1	2	(PC) ← (SK(SP)) (SP) ← (SP) – 1
	RTS	0	0	0	1	0	0	0	1	0	1	0 4 5	1	2	(PC) ← (SK(SP)) (SP) ← (SP) – 1

Note: p is 0 to 31 for M34551M4 and p is 0 to 63 for M34551M8 and M34551E8.

Skip condition	Carry flag CY	Detailed description
–	–	Branch within a page : Branches to address a in the identical page.
–	–	Branch out of a page : Branches to address a in page p.
–	–	Branch out of a page : Branches to address $(DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0)_2$ specified by registers D and A in page p.
–	–	Call the subroutine in page 2 : Calls the subroutine at address a in page 2.
–	–	Call the subroutine : Calls the subroutine at address a in page p.
–	–	Call the subroutine : Calls the subroutine at address $(DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0)_2$ specified by registers D and A in page p.
–	–	Returns from interrupt service routine to main routine. Returns each value of data pointer (X, Y, Z), carry flag, skip status, NOP mode status by the continuous description of the LA/LXY instruction, register A and register B to the states just before interrupt.
–	–	Returns from subroutine to the routine called the subroutine.
Skip unconditionally	–	Returns from subroutine to the routine called the subroutine, and skips the next instruction unconditionally.



# HARDWARE

## MACHINE INSTRUCTIONS

### MACHINE INSTRUCTIONS (CONTINUED)

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D <sub>9</sub>	D <sub>8</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	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	(EXF0) = 1 ? After skipping the next instruction, (EXF0) ← 0
	SNZI0	0	0	0	0	1	1	1	0	1	0	0 3 A	1	1	I1 <sub>2</sub> = 1 : (INT) = "H" ?  I1 <sub>2</sub> = 0 : (INT) = "L" ?
	TAV1	0	0	0	1	0	1	0	1	0	0	0 5 4	1	1	(A) ← (V1)
	TV1A	0	0	0	0	1	1	1	1	1	1	0 3 F	1	1	(V1) ← (A)
	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	SNZT1	1	0	1	0	0	0	0	0	0	0	2 8 0	1	1	(T1F) = 1 ? After skipping the next instruction (T1F) ← 0
	SNZT2	1	0	1	0	0	0	0	0	0	1	2 8 1	1	1	(T2F) = 1 ? After skipping the next instruction (T2F) ← 0
	TAW1	1	0	0	1	0	0	1	0	1	1	2 4 B	1	1	(A) ← (W1)
	TW1A	1	0	0	0	0	0	1	1	1	0	2 0 E	1	1	(W1) ← (A)
	TAW2	1	0	0	1	0	0	1	1	0	0	2 4 C	1	1	(A) ← (W2)
	TW2A	1	0	0	0	0	0	1	1	1	1	2 0 F	1	1	(W2) ← (A)
	TAW3	1	0	0	1	0	0	1	1	0	1	2 4 D	1	1	(A <sub>1</sub> , A <sub>0</sub> ) ← (W3 <sub>1</sub> , W3 <sub>0</sub> )
	TW3A	1	0	0	0	0	1	0	0	0	0	2 1 0	1	1	(W3 <sub>1</sub> , W3 <sub>0</sub> ) ← (A <sub>1</sub> , A <sub>0</sub> )

Skip condition	Carry flag CY	Detailed description
–	–	Clears the interrupt enable flag INTE to “0,” and disables the interrupt.
–	–	Sets the interrupt enable flag INTE to “1,” and enables the interrupt.
(EXF0) = 1	–	Skips the next instruction when the contents of EXF0 flag is “1.” After skipping, clears the EXF0 flag to “0.”
(INT) = “H” However, I12 = 1	–	When bit 2 (I12) of register I1 is “1” : Skips the next instruction when the level of INT pin is “H.”
(INT) = “L” However, I12 = 0	–	When bit 2 (I12) of register I1 is “0” : Skips the next instruction when the level of INT pin is “L.”
–	–	Transfers the contents of interrupt control register V1 to register A.
–	–	Transfers the contents of register A to interrupt control register V1.
–	–	Transfers the contents of interrupt control register I1 to register A.
–	–	Transfers the contents of register A to interrupt control register I1.
(T1F) = 1	–	Skips the next instruction when the contents of T1F flag is “1.” After skipping, clears T1F flag.
(T2F) =1	–	Skips the next instruction when the contents of T2F flag is “1.” After skipping, clears T2F flag.
–	–	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.

# HARDWARE

## MACHINE INSTRUCTIONS

### MACHINE INSTRUCTIONS (CONTINUED)

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D <sub>9</sub>	D <sub>8</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Hexadecimal notation			
Timer operation	TAB1	1	0	0	1	1	1	0	0	0	0	2 7 0	1	1	(B) ← (T17–T14) (A) ← (T13–T10)
	T1AB	1	0	0	0	1	1	0	0	0	0	2 3 0	1	1	At timer 1 stop (W20=0), (R17–R14) ← (B) (T17–T14) ← (B) (R13–R10) ← (A) (T13–T10) ← (A) At timer 1 operating (W20=1), (R17–R14) ← (B) (R13–R10) ← (A)
	TLCA	1	0	0	0	0	0	1	1	0	1	2 0 D	1	1	(TLC) ← (A) (RLC) ← (A)
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)
	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
	TPU0A	1	0	0	0	1	0	1	1	0	1	2 2 D	1	1	(PU0) ← (A)
	TAPU0	1	0	0	1	0	1	0	1	1	1	2 5 7	1	1	(A) ← (PU0)

Skip condition	Carry flag CY	Detailed description
–	–	Transfers the contents of timer 1 to registers A and B.
–	–	When stopping ( $W2_0=0$ ), transfers the contents of registers A and B to timer 1 and timer 1 reload register. When operating ( $W2_0=1$ ), transfers the contents of registers A and B only to timer 1 reload register.
–	–	Transfers the contents of register A to timer LC and timer LC reload register.
–	–	Transfers the input of port P0 to register A.
–	–	Outputs the contents of register A to port P0.
–	–	Transfers the input of port P1 to register A.
–	–	Outputs the contents of register A to port P1.
–	–	Transfers the input of port P2 to register A.
–	–	Sets port D to “1.”
–	–	Clears a bit of port D specified by register Y to “0.”
–	–	Sets a bit of port D specified by register Y to “1.”
–	–	Transfers the contents of register A to pull-up control register PU0.
–	–	Transfers the contents of pull-up control register PU0 to register A.

# HARDWARE

## MACHINE INSTRUCTIONS

### MACHINE INSTRUCTIONS (CONTINUED)

Parameter Type of instructions	Mnemonic	Instruction code											Number of words	Number of cycles	Function
		D <sub>9</sub>	D <sub>8</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Hexadecimal notation			
LCD control operation	TL1A	1	0	0	0	0	0	1	0	1	0	2 0 A	1	1	(L1) ← (A)
	TAL1	1	0	0	1	0	0	1	0	1	0	2 4 A	1	1	(A) ← (L1)
	TL2A	1	0	0	0	0	0	1	0	1	1	2 0 B	1	1	(L2) ← (A)
Carrier generating circuit operation	TC1A	1	0	1	0	1	0	1	0	0	0	2 A 8	1	1	(C1) ← (A)
	STCR	1	0	1	0	0	1	1	0	0	0	2 9 8	1	1	Carrier wave generating start
	SPCR	1	0	1	0	0	1	1	0	0	1	2 9 9	1	1	Carrier wave generating stop
	TC2A	1	0	1	0	1	0	1	0	0	1	2 A 9	1	1	(C2 <sub>0</sub> ) ← (A <sub>0</sub> )
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	Power down instruction (POF, POF2) 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	(WDF) ← 0, (WEF) ← 1
	TAMR	1	0	0	1	0	1	0	0	1	0	2 5 2	1	1	(A) ← (MR)
	TMRA	1	0	0	0	0	1	0	1	1	0	2 1 6	1	1	(MR) ← (A)
	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)

Skip condition	Carry flag CY	Detailed description
–	–	Transfers the contents of register A to LCD control register L1.
–	–	Transfers the contents of register L1 to register A.
–	–	Transfers the contents of register A to LCD control register L2.
–	–	Transfers the contents of register A to carrier wave selection register C1.
–	–	Starts generating carrier wave.
–	–	Stops generating carrier wave.
–	–	Transfers the contents of register A to carrier wave output control register C2.
–	–	No operation
–	–	Puts the system in clock operating mode state by executing the POF instruction after executing the EPOF instruction. f(XCIN) oscillation, LCD, timer LC and timer 2 are operated.
–	–	Puts the system in RAM back-up mode state by executing the POF2 instruction after executing the EPOF instruction. Oscillation is stopped.
–	–	Validates the power down instruction (POF, POF2) which is executed after the EPOF instruction by executing the EPOF instruction.
(P) = 1	–	Skips the next instruction when P flag is "1." After skipping, P flag remains unchanged.
–	–	Operates the watchdog timer and initializes the watchdog timer flag (WDF).
–	–	Transfers the contents of clock control register MR to register A.
–	–	Transfers the contents of register A to clock control register MR.
–	–	Transfers the contents of general-purpose register V2 to register A.
–	–	Transfers the contents of register A to general-purpose register V2.

# HARDWARE

## CONTROL REGISTERS

### CONTROL REGISTERS

Interrupt control register V1		at reset : 0000 <sub>2</sub>		at power down : 0000 <sub>2</sub>	R/W
V1 <sub>3</sub>	Timer 2 interrupt enable bit	0	Interrupt disabled (SNZT2 instruction is valid)		
		1	Interrupt enabled (SNZT2 instruction is invalid)		
V1 <sub>2</sub>	Timer 1 interrupt enable bit	0	Interrupt disabled (SNZT1 instruction is valid)		
		1	Interrupt enabled (SNZT1 instruction is invalid)		
V1 <sub>1</sub>	Not used	0	This bit has no function, but read/write is enabled.		
		1			
V1 <sub>0</sub>	External 0 interrupt enable bit	0	Interrupt disabled (SNZ0 instruction is valid)		
		1	Interrupt enabled (SNZ0 instruction is invalid)		

Timer control register W1		at reset : 0000 <sub>2</sub>		at power down : 0000 <sub>2</sub>	R/W
W1 <sub>3</sub>	Prescaler control bit	0	Stop (prescaler state initialized)		
		1	Operating		
W1 <sub>2</sub>	Prescaler dividing ratio selection bit	0	Instruction clock (INSTCK) divided by 4		
		1	Instruction clock (INSTCK) divided by 8		
W1 <sub>1</sub>	Timer 1 count source selection bits	W1 <sub>1</sub>	W1 <sub>0</sub>	Count source	
		0	0	Prescaler output (ORCLK)	
		0	1		
W1 <sub>0</sub>		1	0	Carrier output (CARRY)	
		1	1	Carrier output divided by 2 (CARRY/2)	

Timer control register W2		at reset : 1000 <sub>2</sub>		at power down : — — — 0 <sub>2</sub>	R/W
W2 <sub>3</sub>	Timer 2 count source selection bit	0	f(X <sub>CIN</sub> )		
		1	Prescaler output (ORCLK)		
W2 <sub>2</sub>	Timer 2 count value selection bits	W2 <sub>2</sub>	W2 <sub>1</sub>	Count source	
		0	0	Underflow occur every 2 <sup>14</sup> count	
0		1	Underflow occur every 2 <sup>13</sup> count		
W2 <sub>1</sub>		1	0	Not available	
		1	1	Not available	
W2 <sub>0</sub>	Timer 1 control bit	0	Stop (timer 1 state retained)		
		1	Operating		

Timer control register W3		at reset : 00 <sub>2</sub>		at power down : state retained	R/W
W3 <sub>1</sub>	Timer LC count source selection bit	0	Bit 3 of timer 2 is output (timer 2 count source divided by 16)		
		1	System clock (STCK)		
W3 <sub>0</sub>	Timer LC control bit	0	Stop (timer LC state retained)		
		1	Operating		

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

"—" represents state retained.

### CONTROL REGISTERS (CONTINUED)

Interrupt control register I1		at reset : 0000 <sub>2</sub>		at power down : state retained	R/W
I1 <sub>3</sub>	Not used	0	This bit has no function, but read/write is enabled.		
		1			
I1 <sub>2</sub>	Interrupt valid waveform for INT pin selection bit (Note 2)	0	Falling waveform ("L" level of INT pin is recognized with the SNZI0 instruction)		
		1	Rising waveform ("H" level of INT pin is recognized with the SNZI0 instruction)		
I1 <sub>1</sub>	Not used	0	This bit has no function, but read/write is enabled.		
		1			
I1 <sub>0</sub>	Not used	0	This bit has no function, but read/write is enabled.		
		1			

Pull-up control register PU0		at reset : 0000 <sub>2</sub>		at power down : state retained	R/W
PU0 <sub>3</sub>	Port P1 <sub>3</sub> pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup		
		1	Pull-up transistor ON, key-on wakeup		
PU0 <sub>2</sub>	Port P1 <sub>2</sub> pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup		
		1	Pull-up transistor ON, key-on wakeup		
PU0 <sub>1</sub>	Port P1 <sub>1</sub> pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup		
		1	Pull-up transistor ON, key-on wakeup		
PU0 <sub>0</sub>	Port P1 <sub>0</sub> pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup		
		1	Pull-up transistor ON, key-on wakeup		

Clock control register MR		at reset : 1000 <sub>2</sub>		at power down : state retained	R/W
MR <sub>3</sub>	System clock (STCK) selection bit	0	MR <sub>0</sub> =0	f(X <sub>IN</sub> )	
			MR <sub>0</sub> =1	f(X <sub>CIN</sub> )	
		1	MR <sub>0</sub> =0	f(X <sub>IN</sub> )/4	
			MR <sub>0</sub> =1	f(X <sub>CIN</sub> )/4	
MR <sub>2</sub>	f(X <sub>CIN</sub> ) oscillation circuit control bit	0	f(X <sub>CIN</sub> ) oscillation stop, ports D <sub>6</sub> and D <sub>7</sub> selected		
		1	f(X <sub>CIN</sub> ) oscillation enabled, ports D <sub>6</sub> and D <sub>7</sub> not selected		
MR <sub>1</sub>	f(X <sub>IN</sub> ) oscillation circuit control bit	0	Oscillation enabled		
		1	Oscillation stop		
MR <sub>0</sub>	Clock selection bit	0	f(X <sub>IN</sub> )		
		1	f(X <sub>CIN</sub> )		

Notes 1: "R" represents read enabled, and "W" represents write enabled.

- 2: Depending on the input state of D<sub>5</sub>/INT pin, the external interrupt request flag EXF0 may be set to "1" when the contents of I1<sub>2</sub> is changed. Accordingly, set a value to bit 2 of register I1 and execute the SNZ0 instruction to clear the EXF0 flag after executing at least one instruction.



# HARDWARE

## CONTROL REGISTERS

### CONTROL REGISTERS (CONTINUED)

Carrier wave selection register C1		at reset : 0111 <sub>2</sub>				at power down : 0111 <sub>2</sub>		W
		C1 <sub>3</sub>	C1 <sub>2</sub>	C1 <sub>1</sub>	C1 <sub>0</sub>	Carrier wave frequency		Duty
Carrier wave selection bits		0	0	0	0	STCK/24		1/3
		0	0	0	1	STCK/24		1/2
		0	0	1	0	STCK/16		1/4
		0	0	1	1	STCK/16		1/2
		0	1	0	0	STCK/2		1/2
		0	1	0	1	No carrier wave		
		0	1	1	0	Not available		
		0	1	1	1	"L" fixed		
		1	0	0	0	STCK/12		1/3
		1	0	0	1	STCK/12		1/2
		1	0	1	0	STCK/8		1/4
		1	0	1	1	STCK/8		1/2
		1	1	0	0	STCK		1/2
		1	1	0	1	No carrier wave		
		1	1	1	0	Not available		
		1	1	1	1	"L" fixed		

Carrier wave output control register C2		at reset : 0 <sub>2</sub>		at power down : 0 <sub>2</sub>		W
C2 <sub>0</sub>	Carrier wave output auto-control bit	0	Auto-control output by timer 1 is invalid			
		1	Auto-control output by timer 1 is valid			

Carrier wave generating control flag CR		at reset : 0 <sub>2</sub>		at power down : 0 <sub>2</sub>		W
CR	Carrier wave generating control	0	Carrier wave generating stop (SPCR instruction)			
		1	Carrier wave generating start (STCR instruction)			

Note: "W" represents write enabled.

### CONTROL REGISTERS (CONTINUED)

LCD control register L1		at reset : 0000 <sub>2</sub>		at power down : state retained		R/W	
L1 <sub>3</sub>	Not used	0	This bit has no function, but read/write is enabled				
		1					
L1 <sub>2</sub>	LCD on/off bit	0	Off				
		1	On				
L1 <sub>1</sub>	LCD duty and bias selection bits	L1 <sub>1</sub>	L1 <sub>0</sub>	Duty		Bias	
		0	0	Not available			
0		1	1/2		1/2		
L1 <sub>0</sub>		1	0	1/3		1/3	
		1	1	1/4		1/3	

LCD control register L2		at reset : 1111 <sub>2</sub>		at power down : state retained	W
L23	P23/SEG <sub>19</sub> pin function switch bit	0	SEG <sub>19</sub>		
		1	P2 <sub>3</sub>		
L22	P22/SEG <sub>18</sub> pin function switch bit	0	SEG <sub>18</sub>		
		1	P2 <sub>2</sub>		
L21	P21/SEG <sub>17</sub> pin function switch bit	0	SEG <sub>17</sub>		
		1	P2 <sub>1</sub>		
L20	P20/SEG <sub>16</sub> pin function switch bit	0	SEG <sub>16</sub>		
		1	P2 <sub>0</sub>		

General-purpose register V2	at reset : 0000 <sub>2</sub>	at power down : 0000 <sub>2</sub>	R/W
4-bit general-purpose register.			
The data transfer between register A and this register is performed with the TV2A and TAV2 instructions.			

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

# HARDWARE

## BUILT-IN PROM VERSION

### BUILT-IN PROM VERSION

In addition to the mask ROM version, the 4551 Group has the programmable ROM version software compatible with mask ROM. The One Time PROM version has PROM which can only be written to and not be erased.

The built-in PROM version has functions similar to those of the mask ROM version, but it has a PROM mode that enables writing to built-in PROM.

Table 19 shows the product of built-in PROM version. Figure 41 shows the pin configurations of built-in PROM version. The One Time PROM version has pin-compatibility with the mask ROM version.

Table 19 Product of built-in PROM version

Product	PROM size (X 10 bits)	RAM size (X 4 bits)	Package	ROM type
M34551E8-XXXFP	8192 words	280 words	48P6S-A	One Time PROM [shipped after writing] (shipped after writing and test in factory)
M34551E8FP				One Time PROM [shipped in blank]

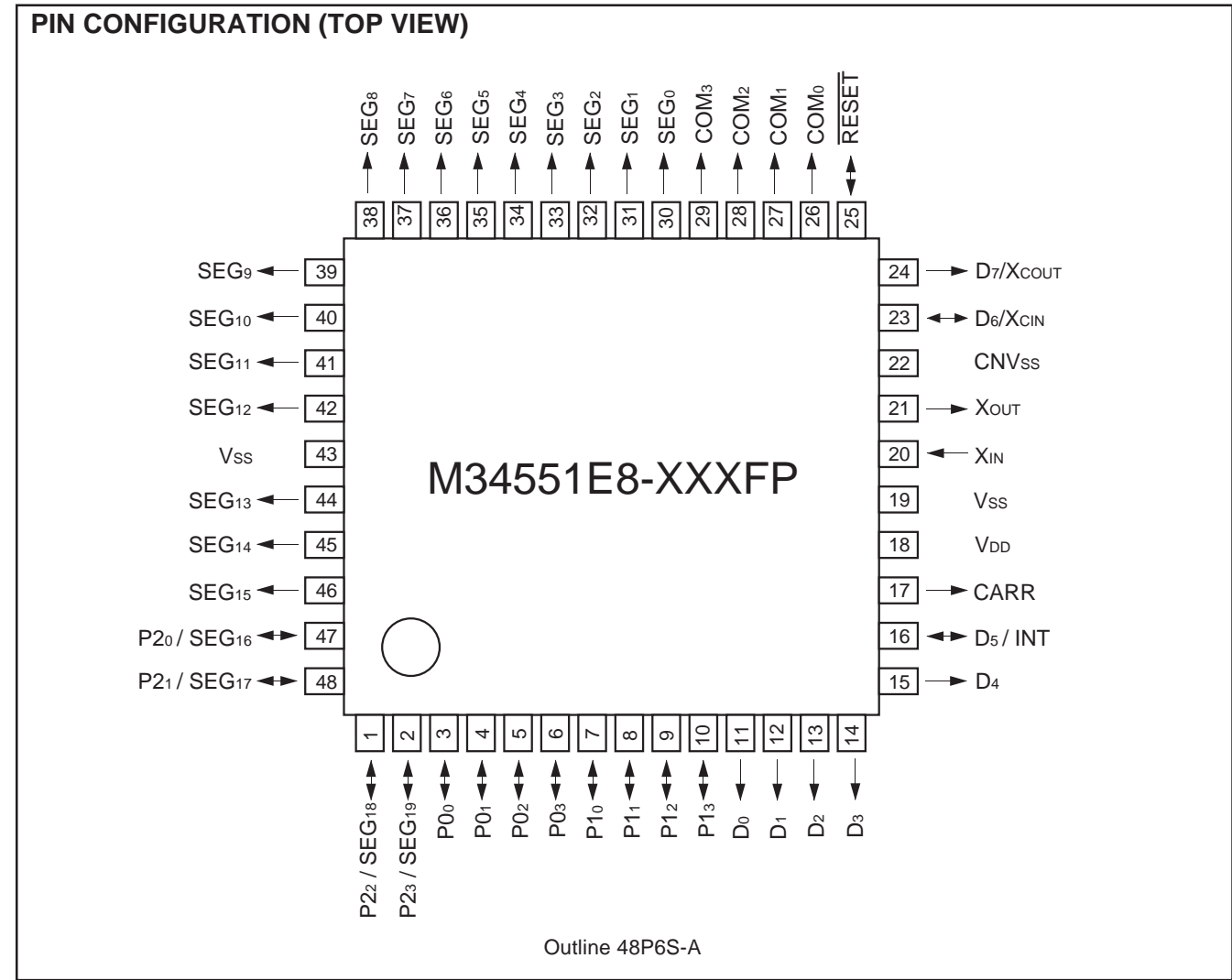


Fig. 41 Pin configuration of built-in PROM version

### (1) PROM mode

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

In the PROM mode, the programming adapter can be used with a general-purpose PROM programmer to write to or read from the built-in PROM as if it were M5M27C256K. Programming adapter is listed in Table 20. Contact addresses at the end of this book for the appropriate PROM programmer.

- Writing and reading of built-in PROM

Programming voltage is 12.5 V. Write the program in the PROM of the built-in PROM version as shown in Figure 42.

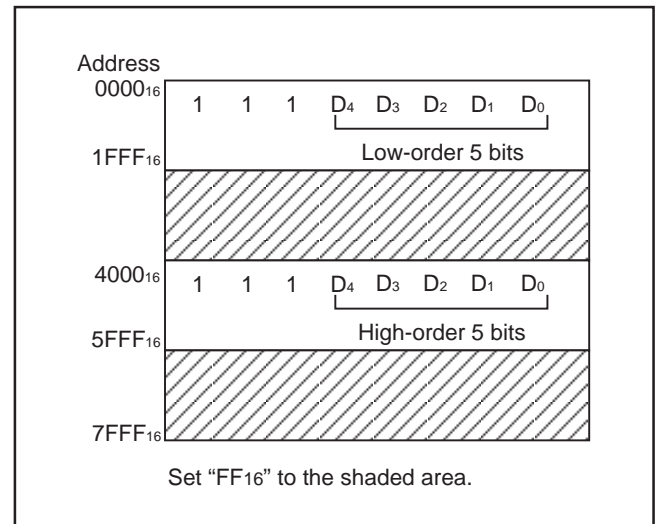
### (2) Notes on handling

- ① A high-voltage is used for writing. Take care that overvoltage is not applied. Take care especially at turning on the power.
- ② For the One Time PROM version shipped in blank, Mitsubishi Electric corp. does not perform PROM writing test and screening in the assembly process and following processes. In order to improve reliability after writing, performing writing and test according to the flow shown in Figure 43 before using is recommended.

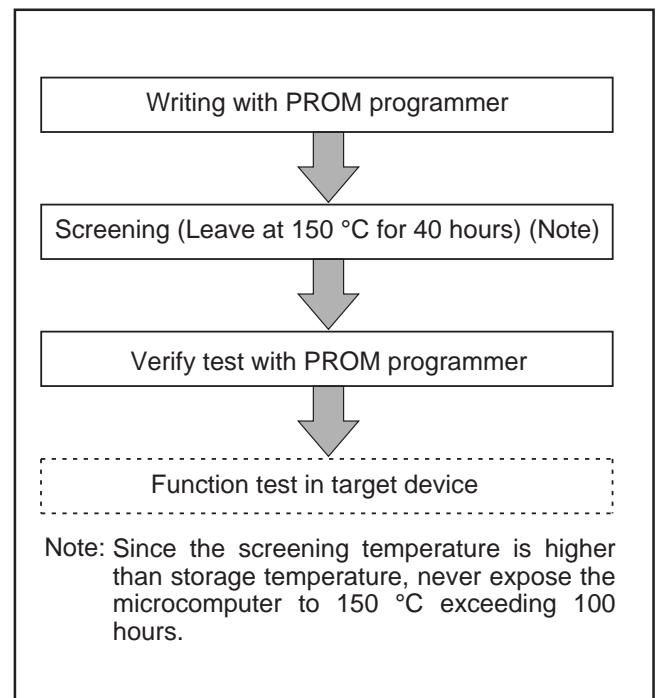
( Products shipped in blank: PROM contents is not written in factory when shipped)

**Table 20 Programming adapter**

Microcomputer	Programming adapter
M34551E8-XXXFP, M34551E8FP	PCA7414



**Fig. 42 PROM memory map**



**Fig. 43 Flow of writing and test of the product shipped in blank**

# HARDWARE

## BUILT-IN PROM VERSION

---

### *MEMO*



# CHAPTER 2

## **APPLICATION**

- 2.1 I/O pins
- 2.2 Interrupts
- 2.3 Timers
- 2.4 Carrier generating circuit
- 2.5 Liquid crystal display
- 2.6 Power down function
- 2.7 Reset
- 2.8 Oscillation circuit

# APPLICATION

## 2.1 I/O pins

---

### 2.1 I/O pins

The 4551 Group has the eight I/O pins, four input pins and eight output pins. (Ports P20–P23, D5–D7 are also used as segment output pins SEG16–SEG19, INT input pin, XCIN, and XCOUT, respectively).

This section describes each port I/O function, related registers, application example using each port function and notes.

#### 2.1.1 I/O ports

##### (1) Port P0

Port P0 is a 4-bit I/O port.

Port P0 has the key-on wakeup function and pull-up transistor.

##### ■ Input/output of port P0

- Data input to port P0

Set the output latch of specified port P0i (i=0 to 3) to “1” with the **OP0A** instruction. If the output latch is set to “0,” “L” level can be input.

The state of port P0 is transferred to register A when the **IAP0** instruction is executed.

- Data output from port P0

The contents of register A is output to port P0 with the **OP0A** instruction.

The output structure is an N-channel open-drain (built-in pull-up resistor).

##### (2) Port P1

Port P1 is a 4-bit I/O port.

Port P1 has the key-on wakeup function and pull-up transistor which turn ON/OFF by setting register PU0.

##### ■ Input/output of port P1

- Data input to port P1

Set the output latch of specified port P1i (i=0 to 3) to “1” with the **OP1A** instruction. If the output latch is set to “0,” “L” level can be input.

The state of port P1 is transferred to register A when the **IAP1** instruction is executed.

- Data output from port P1

The contents of register A is output to port P1 with the **OP1A** instruction.

The output structure is an N-channel open-drain.

**Note:** When the pull-up function becomes valid, simultaneously, the key-on wakeup function becomes valid.

Accordingly, be careful when using the key-on wakeup function. (Refer to the Table 2.1.1 and notes for the power down function.)

### (3) Port P2

Port P2 is a 4-bit input port.

#### ■ Input of port P2

Port P2 is also used as SEG16–SEG19. Accordingly, when ports P20/SEG16–P23/SEG19 are used as port P2, set the corresponding bits of the LCD control register L2 to “1.”

#### ● Data input to port P2

The state of port P2 is transferred to register A when the **IAP2** instruction is executed.

### (4) Port D

D0–D7 are eight independent output ports.

#### ■ Output of port D

Each pin of port D has an independent 1-bit wide output function. For output of ports D0–D7, select one of port D with the register Y of the data pointer first.

#### ● Data output from port D

Set the output level to the output latch with the **SD** and **RD** instructions.

The state of pin enters the high-impedance state when the **SD** instruction is executed.

The states of all port D enter the high-impedance state when the **CLD** instruction is executed.

The state of pin becomes “L” level when the **RD** instruction is executed.

The output structure is an N-channel open-drain.

**Notes 1:** When the **SD** and **RD** instructions are used, do not set “10002” or more to register Y.

**2:** Port D6 is also used as XCIN, and port D7 is also used as XCOUT. Accordingly, when using port D6 and D7 functions, set the clock control register MR2 to “0.”



# APPLICATION

## 2.1 I/O pins

### 2.1.2 Related registers

#### (1) Pull-up control register PU0

Register PU0 controls the ON/OFF of the port P10–P13 pull-up transistor and the ON/OFF of the key-on wakeup function.

Set the contents of this register through register A with the **TPU0A** instruction.

The contents of register PU0 is transferred to register A with the **TAPU0** instruction.

Table 2.1.1 shows the pull-up control register PU0.

**Table 2.1.1 Pull-up control register PU0**

Pull-up control register PU0		at reset : 00002		at power down : state retained	R/W
PU03	Port P13	0	Pull-up transistor OFF, no key-on wakeup function		
	pull-up transistor control bit	1	Pull-up transistor ON, key-on wakeup function		
PU02	Port P12	0	Pull-up transistor OFF, no key-on wakeup function		
	pull-up transistor control bit	1	Pull-up transistor ON, key-on wakeup function		
PU01	Port P11	0	Pull-up transistor OFF, no key-on wakeup function		
	pull-up transistor control bit	1	Pull-up transistor ON, key-on wakeup function		
PU00	Port P10	0	Pull-up transistor OFF, no key-on wakeup function		
	pull-up transistor control bit	1	Pull-up transistor ON, key-on wakeup function		

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

#### (2) LCD control register L2

Register L2 is used to select the port P2 function or segment output pin function.

Set the contents of this register through register A with the **TL2A** instruction.

Table 2.1.2 shows the LCD control register L2.

**Table 2.1.2 LCD control register L2**

LCD control register L2		at reset : 11112		at power down : state retained	W
L23	P23/SEG19 function switch bit	0	SEG19		
		1	P23		
L22	P22/SEG18 function switch bit	0	SEG18		
		1	P22		
L21	P21/SEG17 function switch bit	0	SEG17		
		1	P21		
L20	P20/SEG16 function switch bit	0	SEG16		
		1	P20		

**Note:** “W” represents write enabled.

**(3) Clock control register MR**

The oscillation circuit control bit is assigned to the bit 2 of the clock control register MR.

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

The contents of register MR is transferred to register A with the **TAMR** instruction.

Table 2.1.3 shows the clock control register MR.

**Table 2.1.3 Clock control register MR**

Clock control register MR		at reset : 1000 <sub>2</sub>	at power down : state retained	R/W
MR3	System clock (STCK) selection bit	0	MR0 = 0   f(XIN)	
			MR0 = 1   f(XCIN)	
		1	MR0 = 0   f(XIN)/4	
			MR0 = 1   f(XCIN)/4	
MR2	f(XCIN) oscillation circuit control bit	0	(XCIN) oscillation stop, ports D6, D7 selected	
		1	(XCIN) oscillation enabled, ports D6, D7 not selected	
MR1	f(XIN) oscillation circuit control bit	0	Oscillation enabled	
		1	Oscillation stop	
MR0	Clock selection bit	0	f(XIN)	
		1	f(XCIN)	

**Notes** 1: "R" represents read enabled, and "W" represents write enabled.

2: When setting ports, MR3, MR1, MR0 are not used.

3: Do not stop the oscillation circuit selected with the clock selection bit (MR0).

Note the stop of the oscillation circuit selected with the clock selection bit (MR0) if the following setting is performed.

Example 1: (MR3MR2MR1MR0) = X0X1 (f(XCIN) selected, f(XCIN) oscillation stop)

Example 2: (MR3MR2MR1MR0) = XX10 (f(XIN) selected, f(XIN) oscillation stop)

X: "0" or "1."

# APPLICATION

## 2.1 I/O pins

### 2.1.3 Port application examples

#### (1) Key input by key scan

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

**Outline:** The connecting required external part is just keys.

**Specifications:** Port D is used to output “L” level and port P0 is used to input 16 keys. Multiple key inputs are not detected.

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

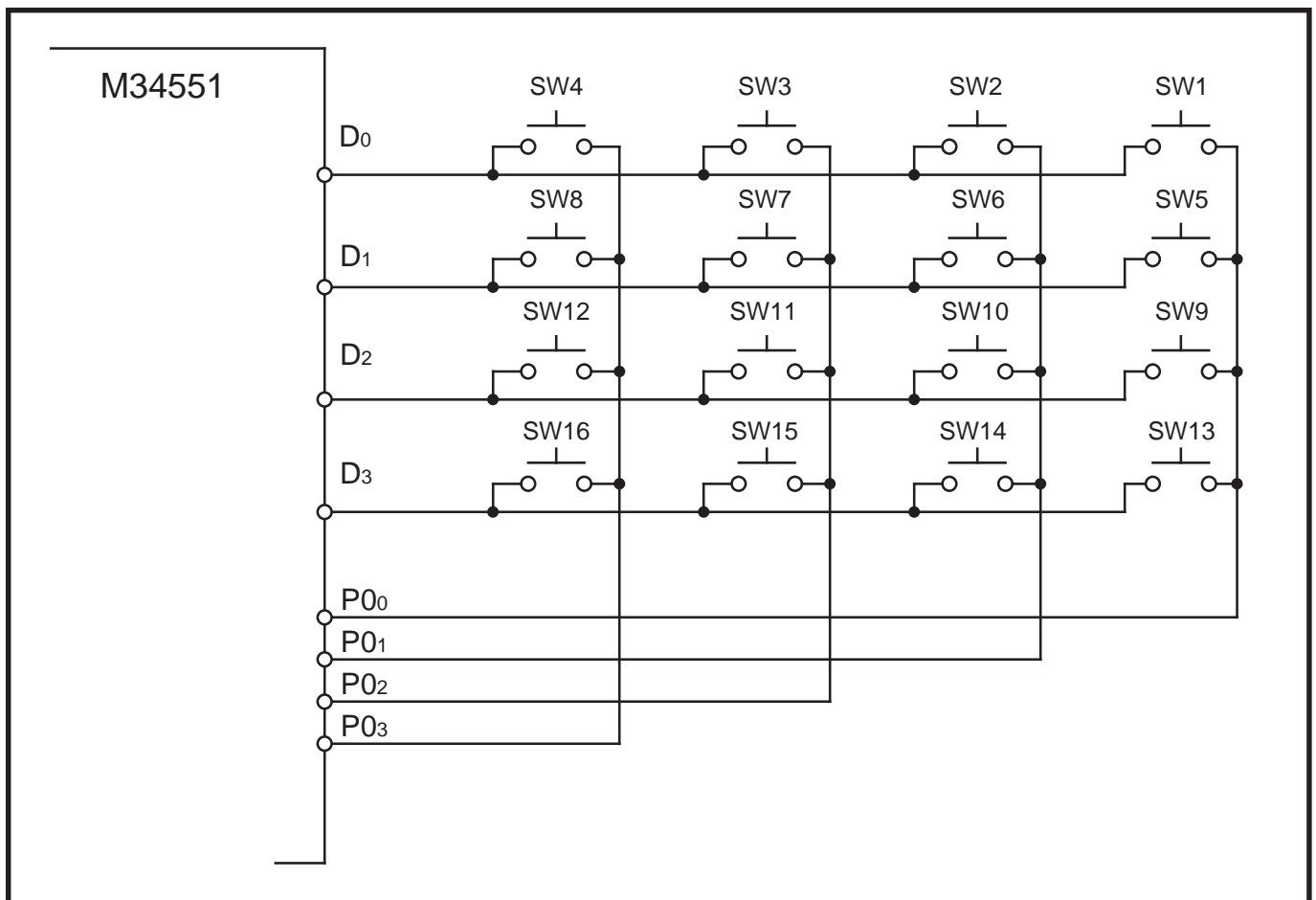


Fig. 2.1.1 Key input by key scan

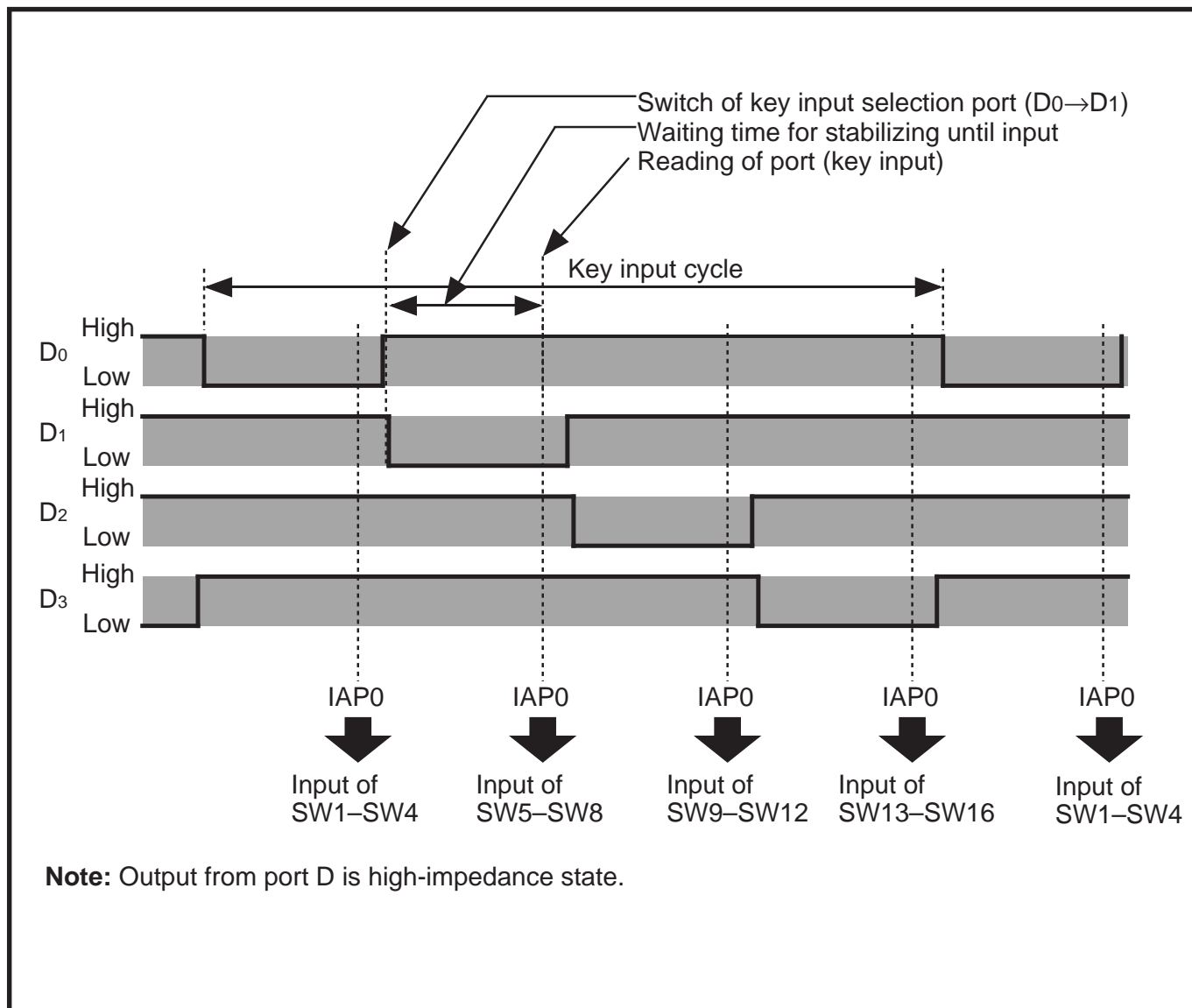


Fig. 2.1.2 Input timing of key scan

# APPLICATION

## 2.1 I/O pins

### 2.1.4 Notes on use

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

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

#### (2) Noise and latch-up prevention

Connect an approximate 0.1  $\mu$ F bypass capacitor directly to the VSS line and the VDD line with the thickest possible wire at the shortest distance, and equalize its wiring in width and length.

The CNVSS pin is also used as the VPP pin (programming voltage = 12.5 V) at the One Time PROM version.

Connect the CNVSS/VPP pin to VSS through an approximate 5 k $\Omega$  resistor which is connected to the CNVSS/VPP pin at the shortest distance.

#### (3) Note on multifunction

Port D5 is also used as the INT pin.

Note that the port D5 output function can be used even when INT pin function is selected.

#### (4) Connection of unused pins

Table 2.1.4 shows the connections of unused pins.

#### (5) SD, RD instructions

When the SD and RD instructions are used, do not set “10002” or more to register Y.

**Table 2.1.4 connections of unused pins**

Pin	Connection
D0–D4, D5/INT	Connect to VSS pin, or set the output latch to “0.”
D6/XCIN, D7/XCOUT	Select D6 and D7 and connect to VSS, or set the output latch to “0” and open.
P20/SEG16–P23/SEG19	Select P2 and connect to VSS, or select segment output function and open.
CARR	Open.
SEG0–SEG15	Open.
COM0–COM3	Open.
P00–P03	Set the output latch to “1” and open.
P10–P13	Open or connect to VSS (Note).

**Note:** In order to connect ports P10–P13 to VSS, turn off their pull-up transistors (Pull-up control register PU0i=“0”) by software. In order to make these pins open, turn on their pull-up transistors (register PU0i=“1”) by software, or turn off their pull-up transistors (register PU0i=“0”) and set the output latch to “0” (i = 0, 1, 2, or 3).

Be sure to select the key-on wakeup function and the pull-up function with every one port.

(Note in order to set the output latch to “0” and make pins open)

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

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

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

## 2.2 Interrupts

The 4551 Group has three interrupt sources : external interrupt (INT), timer 1 interrupt and timer 2 interrupt. This section describes individual types of interrupts, related registers, application examples using interrupts and notes.

### 2.2.1 Interrupt functions

#### (1) External interrupt (INT)

The interrupt request occurs by the change of input level of INT pin.

The interrupt valid waveform can be selected by the bit 2 of the interrupt control register I1.

##### ■ External interrupt processing

- When the interrupt is used

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

- When the interrupt is not used

The interrupt is disabled and the **SNZ0** instruction is valid when the bit 0 of register V1 is set to "0."

#### (2) Timer 1 interrupt

The interrupt request occurs by the timer 1 underflow.

##### ■ Timer 1 interrupt processing

- When the interrupt is used

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

- When the interrupt is not used

The interrupt is disabled and the **SNZT1** instruction is valid when the bit 2 of register V1 is set to "0."

#### (3) Timer 2 interrupt

The interrupt request occurs by the timer 2 underflow.

##### ■ Timer 2 interrupt processing

- When the interrupt is used

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

- When the interrupt is not used

The interrupt is disabled and the **SNZT2** instruction is valid when the bit 3 of register V1 is set to "0."

# APPLICATION

## 2.2 Interrupts

### 2.2.2 Related registers

#### (1) Interrupt enable flag (INTE)

The interrupt enable flag (INTE) controls whether the every interrupt enable/disable.

Interrupts are enabled when INTE flag is set to “1” with the **EI** instruction and disabled when INTE flag is cleared to “0” with the **DI** instruction.

When any interrupt occurs, the INTE flag is automatically cleared to “0,” so that other interrupts are disabled until the **EI** instruction is executed.

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

#### (2) Interrupt control register V1

Interrupt enable bits of external 0, timer 1 and timer 2 are assigned to register V1.

Set the contents of this register through register A with the **TV1A** instruction.

In addition, the **TAV1** instruction can be used to transfer the contents of register V1 to register A.

Table 2.2.1 shows the interrupt control register V1.

**Table 2.2.1 Interrupt control register V1**

Interrupt control register V1		at reset : 00002		at power down : 00002	R/W
V13	Timer 2 interrupt enable bit	0	Interrupt disabled ( <b>SNZT2</b> instruction is valid)		
		1	Interrupt enabled ( <b>SNZT2</b> instruction is invalid)		
V12	Timer 1 interrupt enable bit	0	Interrupt disabled ( <b>SNZT1</b> instruction is valid)		
		1	Interrupt enabled ( <b>SNZT1</b> instruction is invalid)		
V11	Not used	0	This bit has no function, but read/write is enabled.		
		1			
V10	External 0 interrupt enable bit	0	Interrupt disabled ( <b>SNZ0</b> instruction is valid)		
		1	Interrupt enabled ( <b>SNZ0</b> instruction is invalid)		

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

#### (3) Interrupt request flag

The activated condition for each interrupt is examined. Each interrupt request flag is set to “1” when the activated condition is satisfied, even if the interrupt is disabled by the INTE flag or its interrupt enable bit.

Each interrupt request flag is cleared to “0” when either;

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

**(4) Interrupt control register I1**

The interrupt valid waveform for INT pin is assigned to the bit 2 of register I1.

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

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

Table 2.2.2 shows the interrupt control register I1.

**Table 2.2.2 Interrupt control register I1**

Interrupt control register I1		at reset : 0000 <sub>2</sub>	at power down : state retained	R/W
I13	Not used	0	This bit has no function, but read/write is enabled.	
		1		
I12	Interrupt valid waveform for INT pin selection bit( <b>Note 2</b> )	0	Falling waveform ("L" level of INT pin is recognized with the <b>SNZIO</b> instruction)	
		1	Rising waveform ("H" level of INT pin is recognized with the <b>SNZIO</b> instruction)	
I11	Not used	0	This bit has no function, but read/write is enabled.	
		1		
I10	Not used	0	This bit has no function, but read/write is enabled.	
		1		

**Notes 1:** "R" represents read enabled, and "W" represents write enabled.

**2:** Depending on the input state of D5/INT pin, the external interrupt request flag (EXF0) may be set to "1" when the interrupt valid waveform is changed. Accordingly, set a value to the bit 2 of register I1, and execute the **SNZ0** instruction to clear the EXF0 flag to "0" after executing at least one instruction.



# APPLICATION

## 2.2 Interrupts

### 2.2.3 Interrupt application examples

#### (1) External interrupt

The INT pin is used for external interrupts, of which valid waveforms can be chosen, which can recognize the change of both edges ("H"→"L" or "L"→"H").

**Outline:** An external interrupt can be used by dealing with the change of edge ("H"→"L" or "L"→"H") in both directions as a trigger.

**Specifications:** An interrupt occurs by the change of an external signals edge ("H"→"L" or "L"→"H").

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

#### (2) Timer 1 interrupt

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

**Outline:** The constant period interrupts by the timer 1 underflow signal can be used.

**Specifications:** Prescaler and timer 1 divide the system clock frequency  $f(XIN) = 3.6 \text{ MHz}$ , and the timer 1 interrupt occurs every 1 ms.

Figure 2.2.3 shows a setting example of the timer 1 constant period interrupt.

#### (3) Timer 2 interrupt

Timer 2 is the fixed dividing frequency, and the constant period interrupts which the count source is divided by  $2^{13}$  or  $2^{14}$  can be used.

**Outline:** The constant period interrupts by the timer 2 underflow signal can be used.

**Specifications:** Timer 2 divides the sub-clock frequency  $f(XCIN) = 32.768 \text{ kHz}$ , and the timer 2 interrupt occurs every 0.5 sec.

Figure 2.2.4 shows a setting example of the timer 2 constant period interrupt.

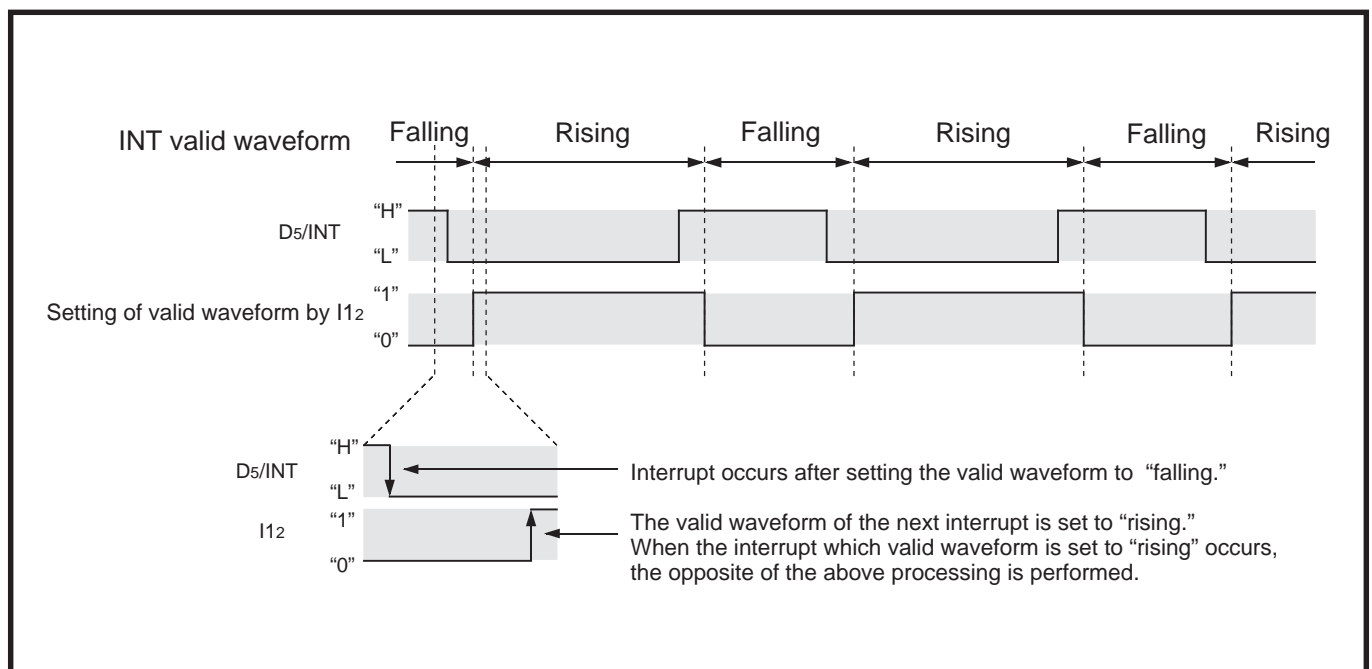


Fig. 2.2.1 External interrupt operation example

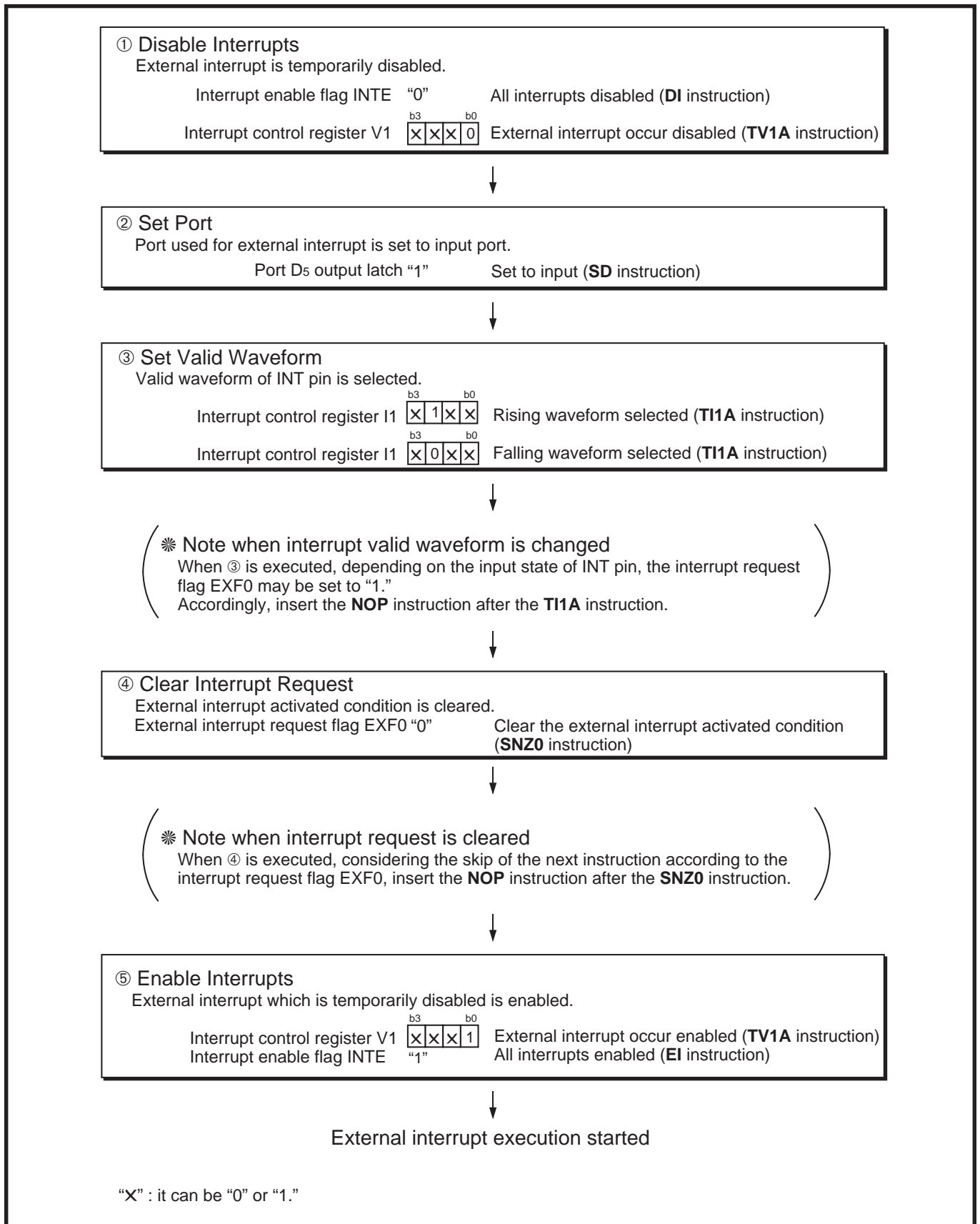


Fig. 2.2.2 External interrupt setting example

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

# APPLICATION

## 2.2 Interrupts

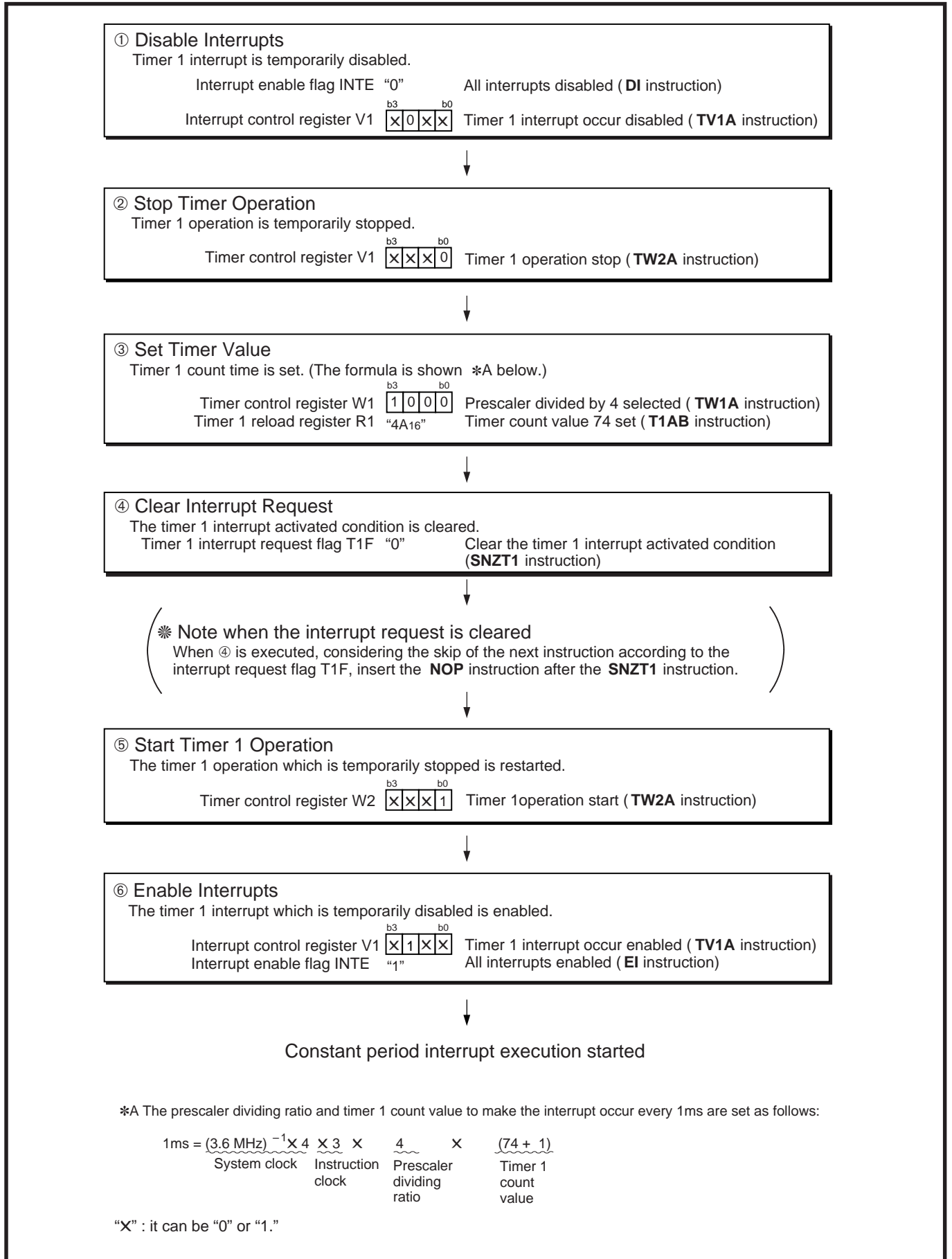
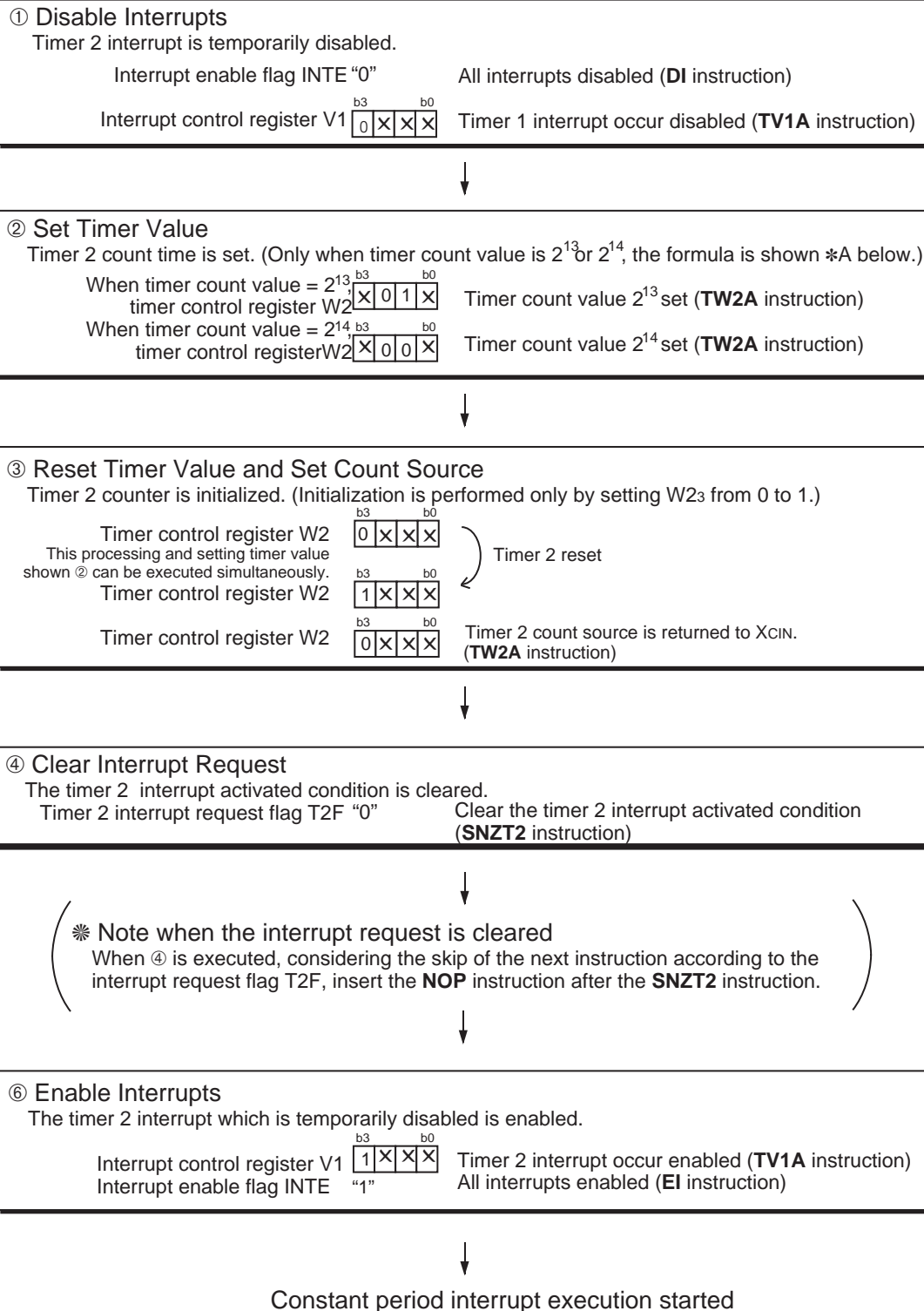


Fig. 2.2.3 Timer 1 constant period interrupt setting example



\* A The timer 2 count value to make the interrupt occur every 0.5s is set as follows:

$$0.5s = \underbrace{(32.768\text{kHz})^{-1}}_{\text{Sub-clock}} \times \underbrace{2^{14}}_{\text{Timer 2 count value}}$$

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

Fig. 2.2.4 Timer 2 constant period interrupt setting example

# APPLICATION

## 2.2 Interrupts

---

### 2.2.4 Notes on use

(1) **Setting of external interrupt valid waveform**

Depending on the input state of D5/INT pin, the external interrupt request flag (EXF0) may be set to “1” when the interrupt valid waveform is changed. Accordingly, set a value to the bit 2 of register I1, and execute the **SNZ0** instruction to clear the EXF0 flag to “0” after executing at least one instruction.

(2) **Multiple interrupts**

Multiple interrupts cannot be used in the 4551 Group.

(3) **Notes on interrupt processing**

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

(4) **D5/INT pin**

The D5/INT pin need not be selected the external interrupt input INT function or the normal output port D5 function. However, the EXF0 flag is set to “1” when a valid waveform output from port D5 is input to INT pin even if it is used as an output port D5.

(5) **Power down instruction**

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

## 2.3 Timers

The 4551 Group has an 8-bit timer with a reload register, a 4-bit timer and the 14-bit fixed dividing frequency timer which has the watchdog timer function.

This section describes individual types of timers, related registers, application examples using timers and notes.

### 2.3.1 Timer functions

#### (1) Timer 1

##### ■ Timer operation

##### ■ Carrier wave output auto-control function

(Refer to section “2.4 Carrier generating circuit” for details.)

#### (2) Timer 2

##### ■ Timer operation

(Timer 2 has the function to return from clock operation mode (**POF** instruction execution))

##### ■ Watchdog function

Watchdog timer provides a method to reset the system when a program runs wild.

When the **WRST** instruction is executed after system is released from reset, in this time, the watchdog timer starts operating. System reset is performed if the **WRST** instruction is not performed while timer 2 counts  $2^{13}$ .

#### (3) Timer LC

##### ■ LCD frame clock generating

### 2.3.2 Related registers

#### (1) Interrupt control register V1

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

Set the contents of this register through register A with the **TV1A** instruction. The **TAV1** instruction can be used to transfer the contents of register V1 to register A.

Table 2.3.1 shows the interrupt control register V1.

**Table 2.3.1 Interrupt control register V1**

Interrupt control register V1		at reset : 00002		at power down : 00002	R/W
V13	Timer 2 interrupt enable bit	0	Interrupt disabled ( <b>SNZT2</b> instruction is valid)		
		1	Interrupt enabled ( <b>SNZT2</b> instruction is invalid)		
V12	Timer 1 interrupt enable bit	0	Interrupt disabled ( <b>SNZT1</b> instruction is valid)		
		1	Interrupt enabled ( <b>SNZT1</b> instruction is invalid)		
V11	Not used	0	This bit has no function, but read/write is enabled.		
		1			
V10	External 0 interrupt enable bit	0	Interrupt disabled ( <b>SNZ0</b> instruction is valid)		
		1	Interrupt enabled ( <b>SNZ0</b> instruction is invalid)		

**Notes 1:** “R” represents read enabled, and “W” represents write enabled.

**2:** When timer is used, V10 is not used.

# APPLICATION

## 2.3 Timers

### (2) Timer control register W1

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

Set the contents of this register through register A with the **TW1A** instruction. The **TAW1** instruction can be used to transfer the contents of register W1 to register A.

Table 2.3.2 shows the timer control register W1.

**Table 2.3.2 Timer control register W1**

Timer control register W1		at reset : 0000 <sub>2</sub>		at power down : 0000 <sub>2</sub>	R/W
W13	Prescaler control bit	0	Stop (prescaler state initialized)		
		1	Operating		
W12	Prescaler dividing ratio selection bit	0	Instruction clock (INSTCK) divided by 4		
		1	Instruction clock (INSTCK) divided by 8		
W11	Timer 1 count source selection bits	W11	W10	Count source	
		0	0	Prescaler output (ORCLK)	
0		1			
1		0	Carrier output (CARRY)		
W10		1	1	Carrier output/2 (CARRY/2)	

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

### (3) Timer control register W2

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

Set the contents of this register through register A with the **TW2A** instruction. The **TAW2** instruction can be used to transfer the contents of register W2 to register A.

Table 2.3.3 shows the timer control register W2.

**Table 2.3.3 Timer control register W2**

Timer control register W2		at reset : 1000 <sub>2</sub>		at power down : – – – 0 <sub>2</sub>	R/W
W23	Timer 2 count source selection bit	0	f(XCIN)		
		1	Prescaler output (ORCLK)		
W22	Timer 2 count value selection bits	W22	W21	Count source	
		0	0	Underflow occur every 2 <sup>14</sup> count	
W21		0	1	Underflow occur every 2 <sup>13</sup> count	
		1	0	Not available	
		1	1	Not available	
		W20	0	Stop (timer 1 state retained)	
1			Operating		

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

“–” represents state retained.

**(4) Timer control register W3**

The timer LC control bit is assigned to the bit 0, and timer LC count source selection bit is assigned to the bit 1.

Set the contents of this register through register A with the **TW3A** instruction. The **TAW3** instruction can be used to transfer the contents of register W3 to register A.

Table 2.3.4 shows the timer control register W3.

**Table 2.3.4 Timer control register W3**

Timer control register W3		at reset : 00 <sub>2</sub>	at power down : state retained	R/W
W3 <sub>1</sub>	Timer LC count source selection bit	0	Bit 3 of timer 2 is output (timer 2 count source divided by 16)	
		1	System clock (STCK)	
W3 <sub>0</sub>	Timer LC control bit	0	Stop (timer LC state retained)	
		1	Operating	

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



# APPLICATION

## 2.3 Timers

### 2.3.3 Timer application examples

#### (1) Timer operation: measurement of constant period

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

**Outline:** The constant period by the timer 1 underflow signal can be measured.

**Specifications:** Timer 1 and prescaler divides the system clock frequency  $f(XIN) = 3.6 \text{ MHz}$ , and the timer 1 interrupt request occurs every 4 ms.

Figure 2.3.2 shows the setting example of the constant period measurement.

#### (2) Timer operation: constant period counter by timer 2

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

**Outline:** The correct time can be measured and the clock which has high-accuracy can be set up by using a 32.768 kHz quartz-crystal oscillator.

**Specifications:** Timer 2 divides the sub-clock frequency  $f(XCIN) = 32.768 \text{ kHz}$ , and the timer 2 interrupt request occurs every 250 ms.

Figure 2.3.3 shows the setting example of constant period counter by timer 2.

#### (3) Watchdog timer

Watchdog timer provides a method to reset the system when a program run-away occurs.

In the 4551 Group, the bit 12 of timer 2 is used for the watchdog timer.

Accordingly, when the watchdog timer function is set to be valid, execute the **WRST** instruction at a certain cycle which consists of timer 2's 8191 counts or less.

**Outline:** Execute the **WRST** instruction in timer 2's 8192 count at the normal operation. If program runs wild, the **WRST** instruction is never executed and system reset occurs.

**Specifications:** System clock frequency  $f(XIN) = 3.6 \text{ MHz}$ , sub-clock frequency  $f(XCIN) = 32.768 \text{ kHz}$  are used, and program run-away is detected by executing the **WRST** instruction in 250 ms.

Figure 2.3.1 shows the watchdog timer function, and Figure 2.3.4 shows the example of watchdog timer.

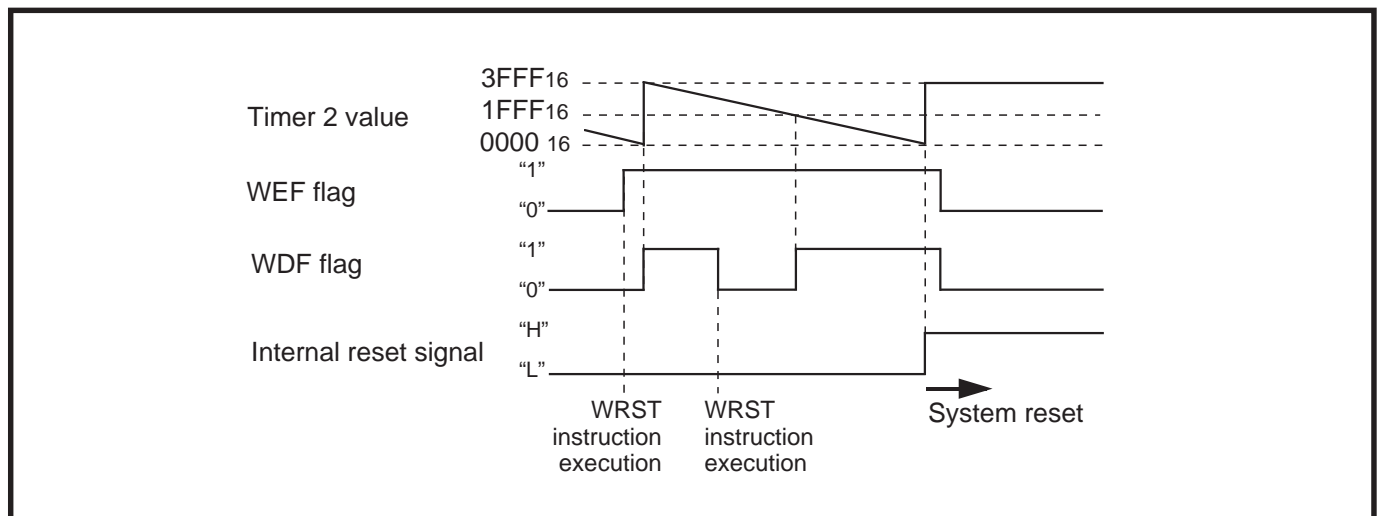
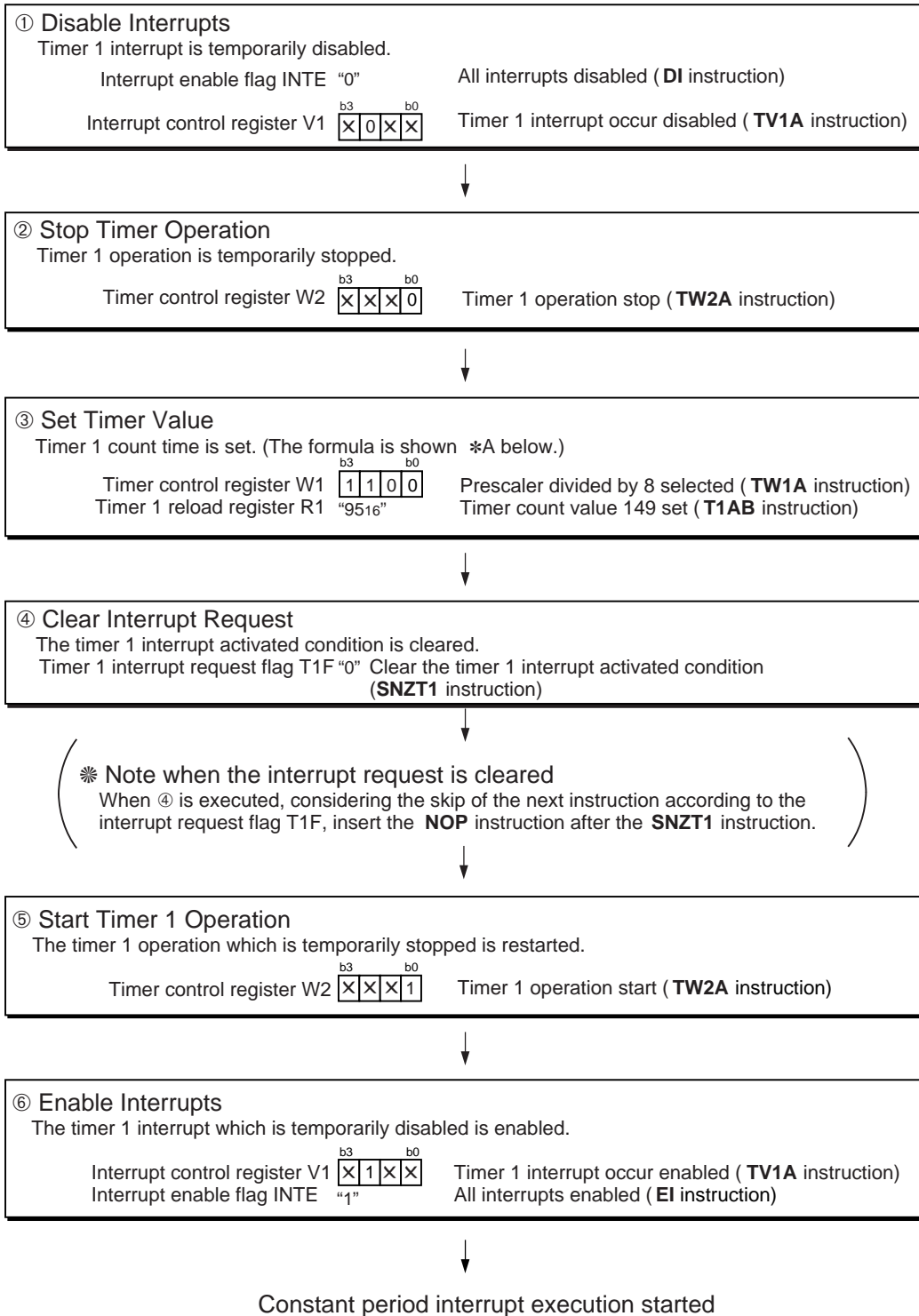


Fig. 2.3.1 Watchdog timer function



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

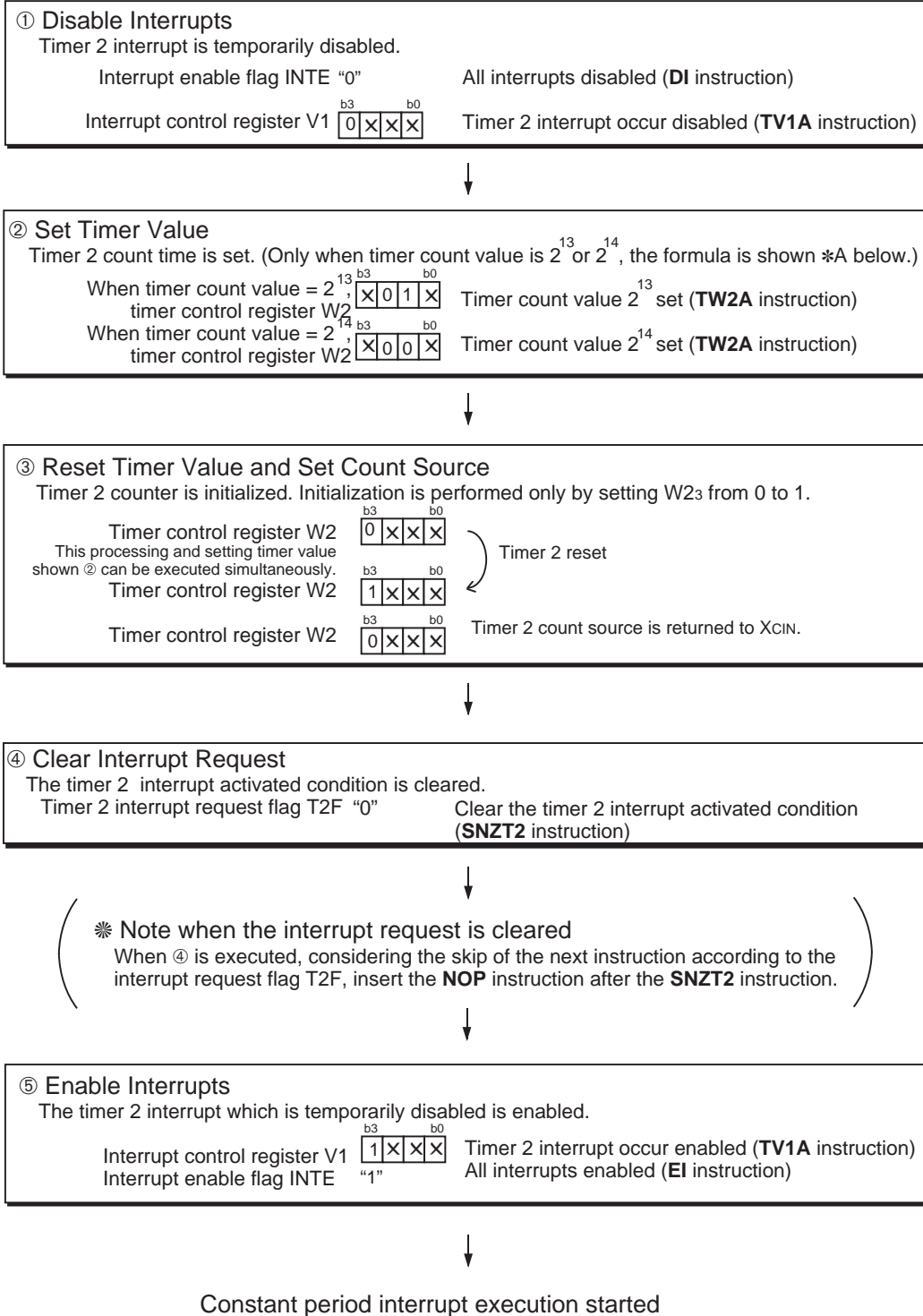
$$4\text{ms} = \underbrace{(3.6\text{ MHz})^{-1}}_{\text{System clock}} \times \underbrace{4 \times 3}_{\text{Instruction clock}} \times \underbrace{8}_{\text{Prescaler dividing ratio}} \times \underbrace{(149 + 1)}_{\text{Timer 1 count value}}$$

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

Fig. 2.3.2 Constant period measurement setting example

# APPLICATION

## 2.3 Timers



\* A The timer 2 count value to make the interrupt occur every 0.25s is set as follows:

$$0.25s = \underbrace{(32.768kHz)^{-1}}_{\text{Sub-clock}} \times \underbrace{2^{13}}_{\text{Timer 2 count value}}$$

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

Fig. 2.3.3 Constant period counter by timer 2 setting example

## ① Set Timer 2 Count Source

Timer 2 countsource is set.

Timer control register W2 

b3	b0
0	x x x

 XCIN selected (**TW2A** instruction)



## ② Activate Watchdog Timer

Watchdog timer is activated.

Watchdog timer enable flag WEF "1"

Watchdog timer enable flag WEF set  
(**WRST** instruction)



Main routine (every 20 ms)

## WDF Flag Reset

Watchdog timer flag WDF is reset. "0"

Watchdog timer flag WDF cleared  
(**WRST** instruction)



Main routine execution



Repeat

Do not perform the interrupt processing for reset of watchdog timer flag WDF.  
Interrupt may keep operating even when a program runs wild.

When going into RAM back-up mode

⋮  
WRST;    WDF flag cleared  
  
EPOF;    POF instruction enabled  
  
POF2  
↓  
Oscillation stop    (RAM back-up mode)

In the RAM back-up mode, the value of WDF flag is initialized.  
However, when the WDF flag is "1," at the same time, a microcomputer may be reset.  
When the watchdog timer and RAM back-up mode are used, execute the **WRST** instruction  
before system enters into the RAM back-up mode in order to initialize the WDF flag.

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

Fig. 2.3.4 Watchdog timer setting example

# APPLICATION

## 2.3 Timers

---

### 2.3.4 Notes on use

(1) **Prescaler**

Stop the prescaler operation to change its frequency dividing ratio.

(2) **Count source**

- Stop timer 1 or timer LC counting to change its count source.
- When timer 2 count source changes from f(XCIN) to ORCLK ( $W23 = "0" \rightarrow W23 = "1"$ ), the count value of timer 2 is initialized. However, when timer 2 count source changes from ORCLK to f(XCIN) ( $W23 = "1" \rightarrow W23 = "0"$ ) or the same count source is set again ( $W23 = "0" \rightarrow W23 = "0"$  or  $W23 = "1" \rightarrow W23 = "1"$ ), the count value of timer 2 is not initialized.

(3) **Timer 2**

Timer 2 has the watchdog timer function (WDT). When timer 2 is used as the WDT, note that the processing to initialize the count value and the execution of the **WRST** instruction.

(4) **Reading the count value**

Stop the prescaler and then execute the **TAB1** instruction to read timer 1 data.

(5) **Writing to reload register R1**

When writing data to reload register R1 while timer 1 is operating, avoid a timing when timer 1 underflows.

### 2.4 Carrier generating circuit

The 4551 Group has a carrier generating circuit that generates the transfer waveform by dividing the system clock (STCK) for each remote control carrier wave.

Also, the 4551 Group has the function to control the carrier wave output from port CARR by using timer 1. This section describes carrier functions, related registers, application examples using each carrier output and notes.

#### 2.4.1 Carrier functions

##### (1) Carrier wave output

Carrier wave is selected by the carrier wave selection register C1.

Carrier output is started with the **STCR** instruction, and carrier output is stopped with the **SPCR** instruction.

##### (2) Carrier wave output auto-control function

Timer 1 can auto-control the output enable/disable interval of port CARR carrier wave by setting register C2.

# APPLICATION

## 2.4 Carrier generating circuit

### 2.4.2 Related registers

#### (1) Carrier wave selection register C1

The output waveform of carrier wave is selected.

Set the contents of this register through register A with the **TC1A** instruction.

Figure 2.4.1 shows the relationship between register C1 and carrier wave.

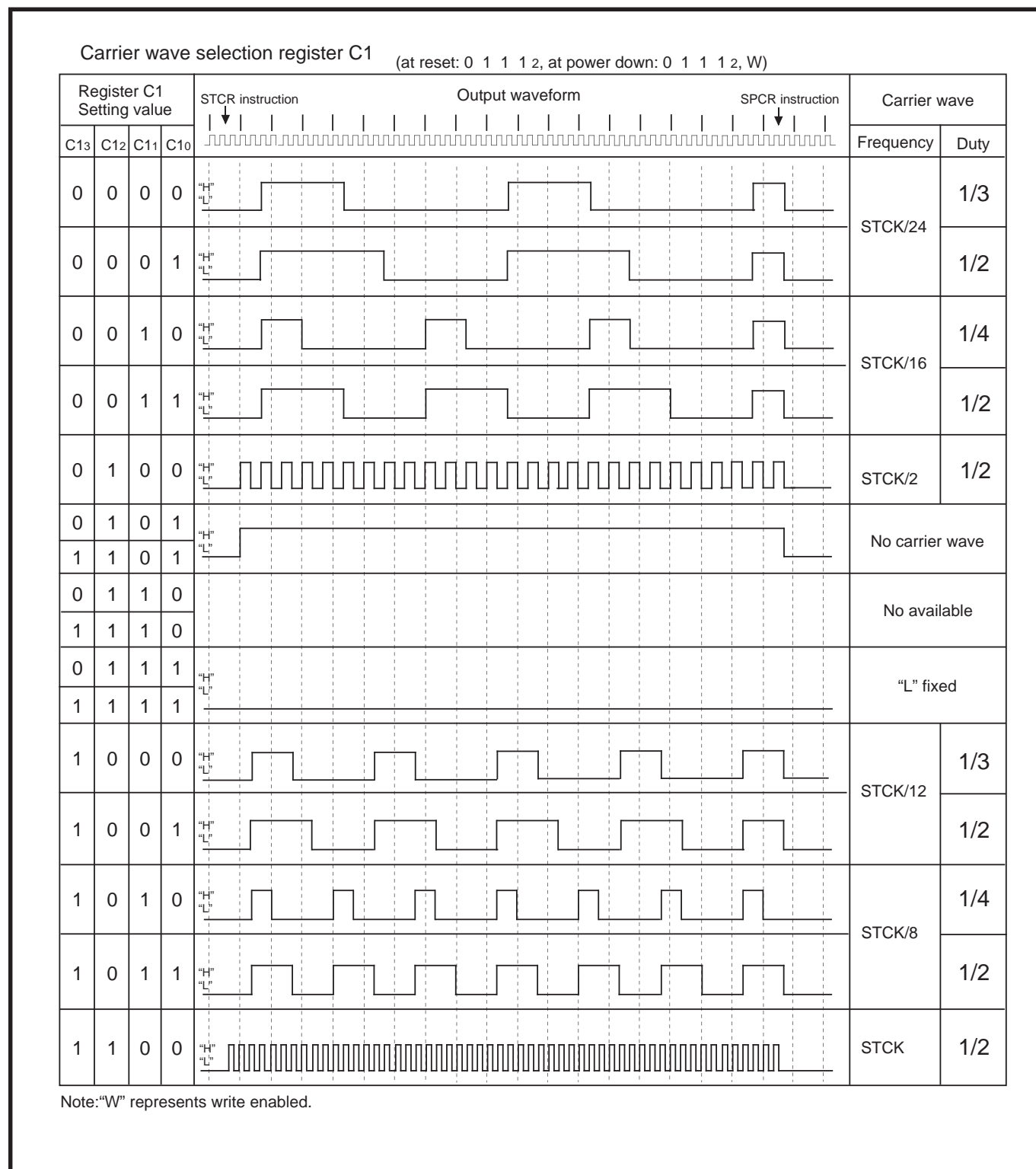


Fig. 2.4.1 Carrier wave selection register

**(2) Carrier wave output control register C2**

The carrier wave output auto-control bit is assigned to the bit.

Set the contents of this register through register A with the **TC2A** instruction.

Table 2.4.1 shows the carrier wave output control register C2.

**Table 2.4.1 Carrier wave output control register C2**

Carrier wave output control register C2		at reset : 02		at power down : 02		W
C20	Carrier wave output auto-control bit	0	Auto-control output by timer 1 is invalid			
		1	Auto-control output by timer 1 is valid			

**Note:** “W” represents write enabled.

**(3) Carrier wave generating control flag CR**

Execute the **SPCR** instruction to set the carrier wave generating control flag CR to “0.”

Execute the **STCR** instruction to set the carrier wave generating control flag CR to “1.”

Table 2.4.2 shows the carrier wave generating control flag CR.

**Table 2.4.2 Carrier wave generating control flag CR**

Carrier wave generating control flag CR		at reset : 02		at power down : 02		W
CR	Carrier wave generation control	0	Carrier wave generating stop ( <b>SPCR</b> instruction)			
		1	Carrier wave generating start ( <b>STCR</b> instruction)			

**Note:** “W” represents write enabled.



# APPLICATION

## 2.4 Carrier generating circuit

### 2.4.3 Carrier wave output application examples

#### (1) Remote control waveform output by carrier wave output auto-control function

The carrier wave output auto-control function can be used to turn carrier wave ON/OFF automatically by counting carrier waveform.

**Outline:** ① CARRY is selected as the timer 1 count source and port CARR can be controlled.

② Output of waveform can be controlled by the bit C20.

③ OFF interval can be output.

**Specifications:** The 37.9 kHz carrier wave is output from port CARR by using system clock frequency  $f(XIN) = 3.64 \text{ MHz}$ .

Also, the timer 1 interrupt occurs, and at the same time, setting the next output interval is performed.

Figure 2.4.2 and Figure 2.4.3 show the setting example of carrier wave auto-control, Figure 2.4.4 shows the setting example of carrier wave output interval.

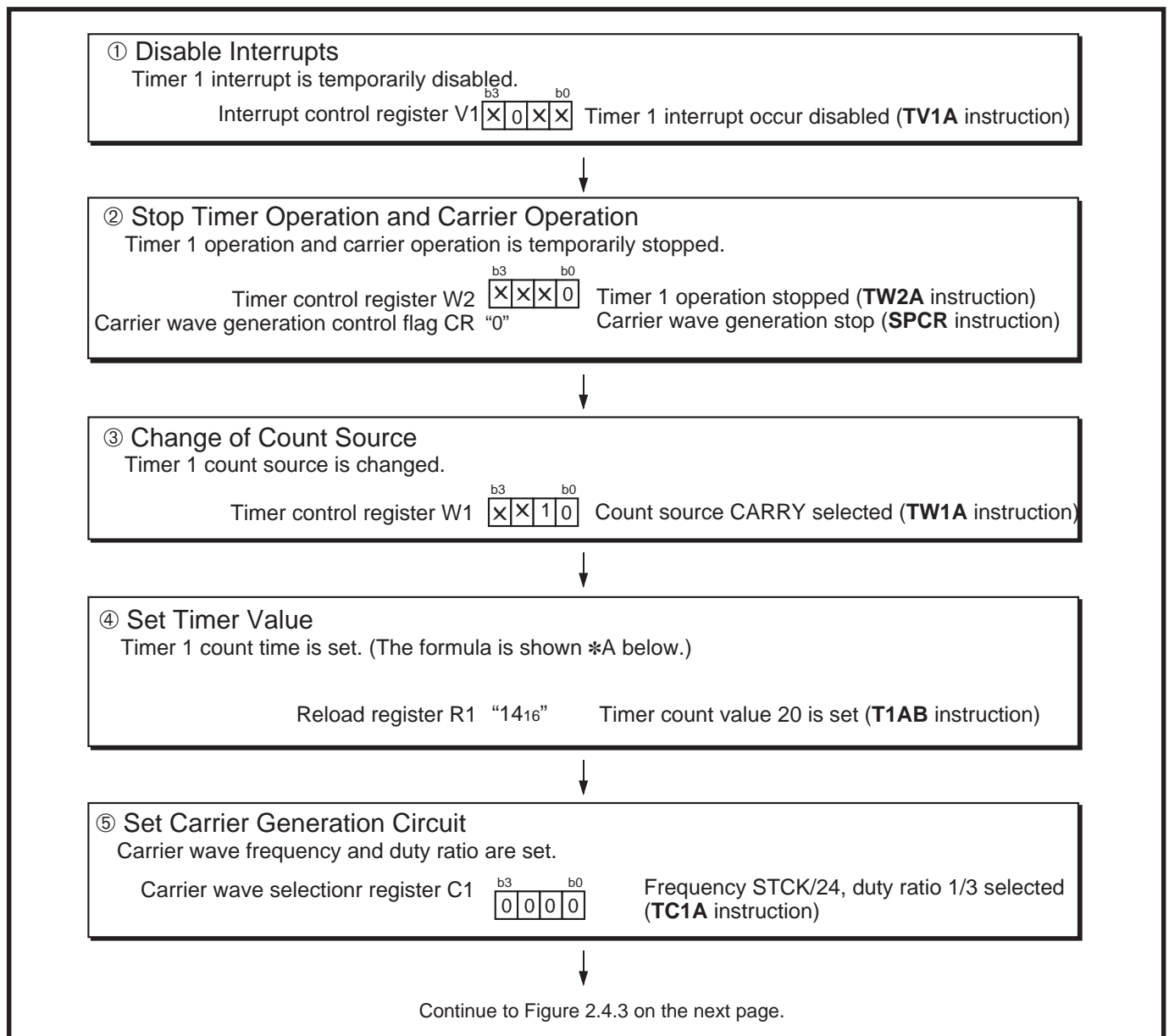


Fig. 2.4.2 Carrier wave auto-control setting example 1

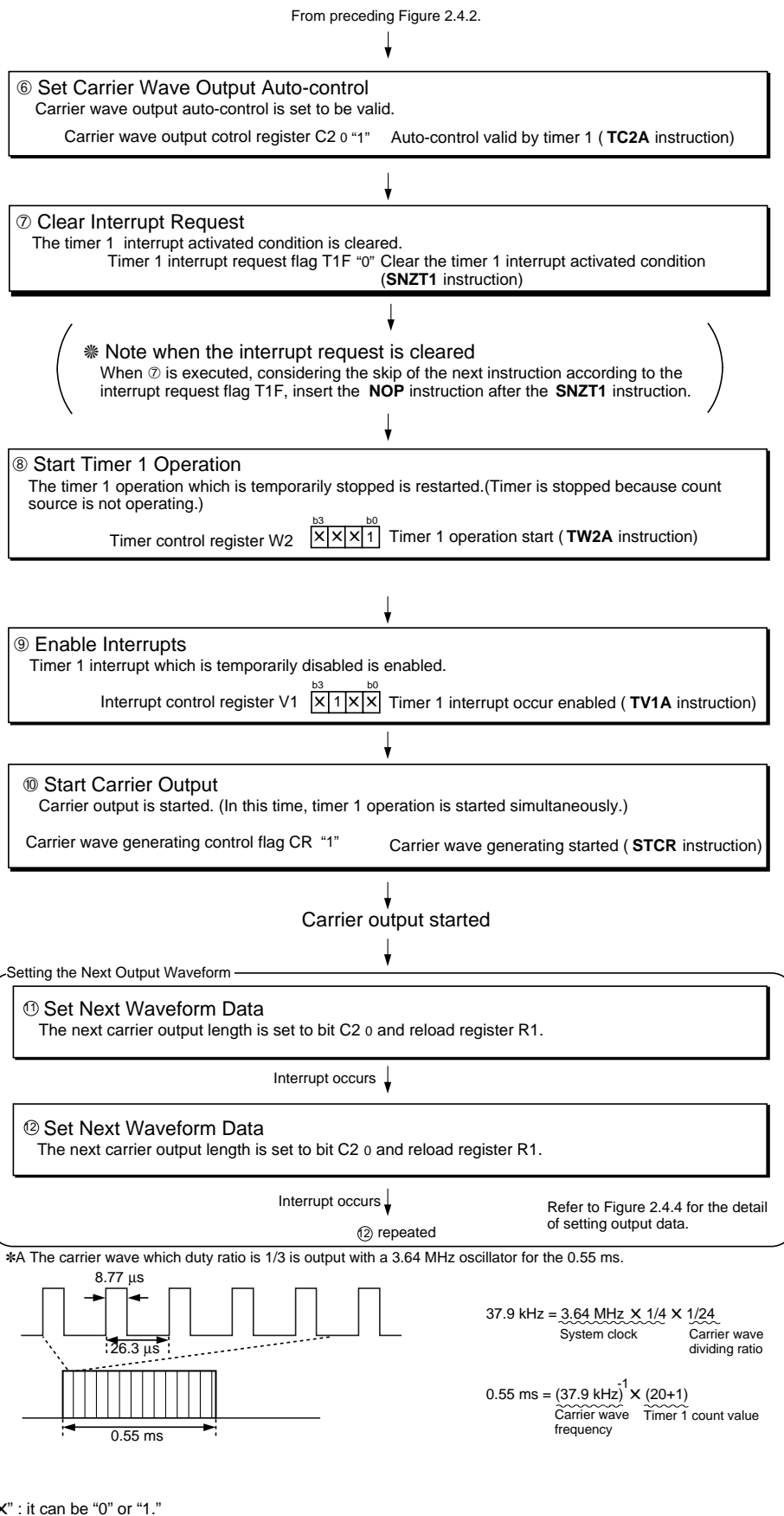


Fig. 2.4.3 Carrier wave auto-control setting example 2

# APPLICATION

## 2.4 Carrier generating circuit

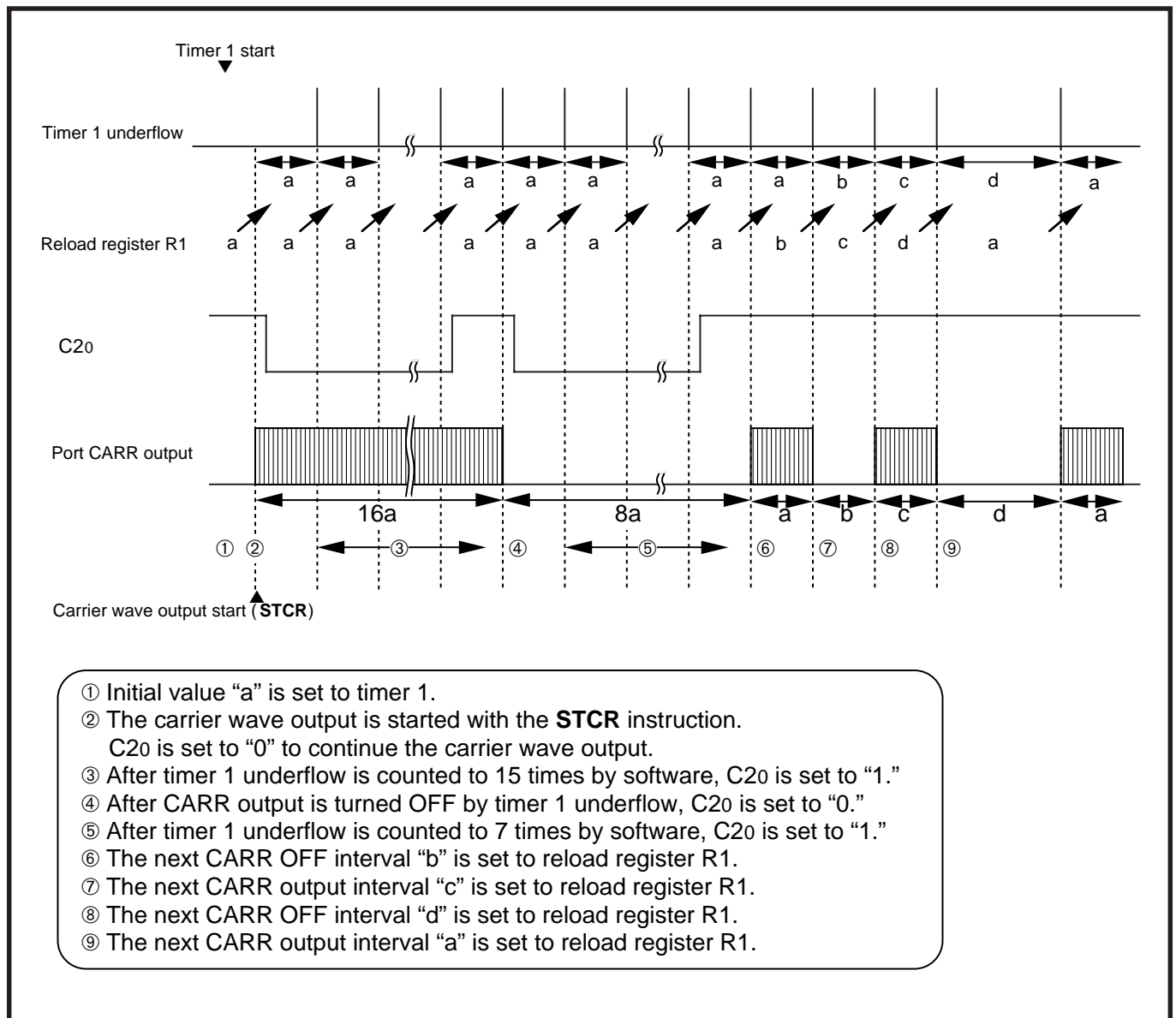


Fig. 2.4.4 Carrier wave output interval setting example

(2) Carrier wave generating by software

Carrier wave generating can be output by software count.

**Outline:** The carrier wave generating circuit is set to “no carrier wave” and carrier wave is generated by software.

**Specifications:** The 37.9 kHz carrier wave is generated by using main clock frequency  $f(XIN) = 3.64$  MHz.

Figure 2.4.5 shows the generating example of carrier wave by software.

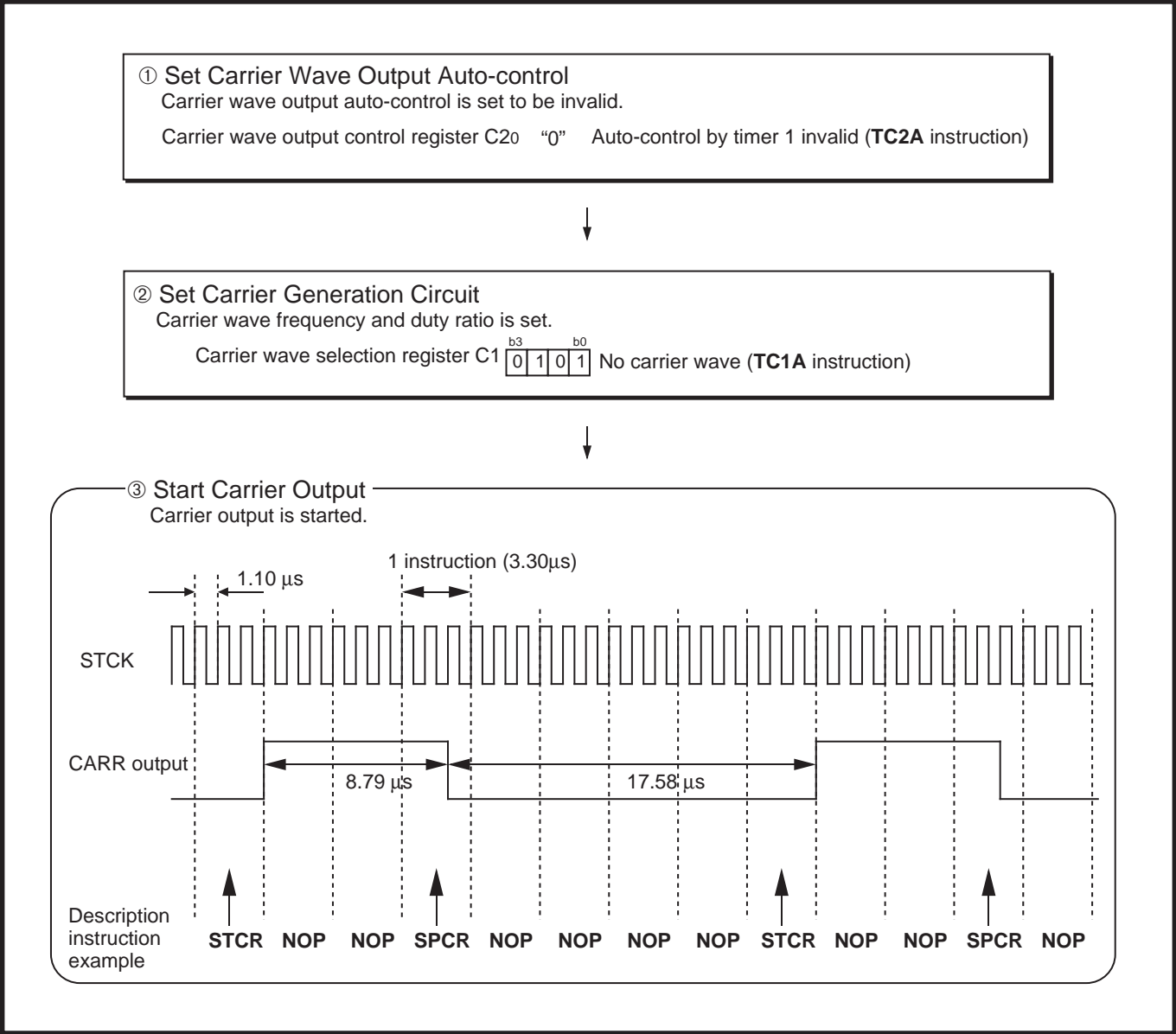


Fig. 2.4.5 Carrier wave by software generating example

# APPLICATION

## 2.4 Carrier generating circuit

---

### 2.4.4 Notes on use

#### (1) Note on the carrier generating circuit stop

In order to stop the carrier wave which has the cycle longer than that of the instruction clock with the **SPCR** instruction, stop it at the point when the carrier wave outputs "L" level in the **SPCR** instruction execution cycle.

If this condition is not satisfied, the last "H" output interval of carrier wave is shortened.

#### (2) Notes when using the carrier wave output auto-control function

- Execute the **STCR** instruction after setting the timer 1 and register C2 in order to start the carrier generating circuit operation.
- Stop the timer 1 (W20="0") after stopping the carrier generating circuit (**SPCR** instruction executed) while the carrier wave output is disabled in order to stop the carrier wave output auto-control operation.
- If the carrier wave output auto-control function is invalidated (C20="0") while the carrier wave output is auto-controlled, the output of port CARR retains the state when the auto-control is invalidated regardless of timer 1 underflow. This state can be terminated by timer 1 stop (W20="0"). When the carrier wave output auto-control function is validated (C20="1") again after it is invalidated (C20="0"), the auto-control of carrier wave output is started again when the next timer 1 underflow occurs. However, when the carrier wave output auto-control bit is changed during timer 1 underflow, the error-operation may occur.
- Use the carrier wave or the carrier wave divided by 2 as the timer 1 count source when the carrier wave output auto-control function is selected.  
If the ORCLK is used as the count source, a hazard wave may occur in port CARR output because ORCLK is not synchronized with the carrier wave.
- When "no carrier wave" is selected with register C1 ((C13C12C11C10) = (0101), (1101)), the disable/enable of the carrier wave output cannot be controlled by the carrier wave output auto-control function.

## 2.5 LCD function

The 4551 Group has an LCD (Liquid Crystal Display) controller/driver.

4 common signal output pins and 20 segment signal output pins can be used to drive the LCD. By using these pins, up to 80 segments (when 1/4 duty and 1/3 bias are selected) can be controlled to display.

This section describes the LCD operation description, related registers, application examples using the LCD and notes.

### 2.5.1 Operation description

#### (1) LCD duty and bias control

Table 2.5.1 shows the duty and maximum number of displayed pixels. Use bits 0 and 1 of LCD control register (L1) to select the proper display method for the LCD panel being used. Also, when using segment pins are 19 or less, SEG16–SEG19 can be used as an I/O port with register L2.

**Table 2.5.1 Duty and maximum number of displayed pixels**

Duty	Bias	Maximum number of displayed pixels	Used COM pins
1/2	1/2	40 segments	COM <sub>0</sub> , COM <sub>1</sub> ( <b>Note</b> )
1/3	1/3	60 segments	COM <sub>0</sub> –COM <sub>2</sub> ( <b>Note</b> )
1/4	1/3	80 segments	COM <sub>0</sub> –COM <sub>3</sub> ( <b>Note</b> )

**Note:** Leave unused COM pins open.

#### (2) LCD drive timing

The frequency (F) of the LCD clock generating the LCD drive timing and frame frequency are shown below. Figure 2.5.1 shows the structure of the LCD clock circuit.

- When the prescaler output (ORCLK) is used for the timer 2 count source (W23 = "1")

$$F = \underbrace{\text{ORCLK}}_{\text{①}} \times \underbrace{\frac{1}{16}}_{\text{②③}} \times \underbrace{\frac{1}{\text{LC} + 1}}_{\text{④}} \times \underbrace{\frac{1}{2}}_{\text{⑤}}$$

- When f(XCIN) is used for the timer 2 count source (W23 = "0")

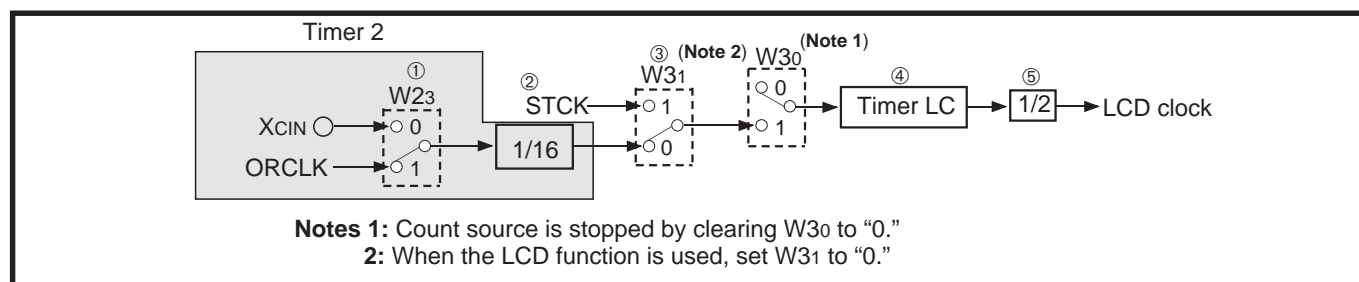
$$F = \underbrace{f(\text{XCIN})}_{\text{①}} \times \underbrace{\frac{1}{16}}_{\text{②③}} \times \underbrace{\frac{1}{\text{LC} + 1}}_{\text{④}} \times \underbrace{\frac{1}{2}}_{\text{⑤}}$$

The frame frequency for each display method can be obtained by the following formula.

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

$$\text{Frame period} = \frac{n}{F} \text{ (s)}$$

[F: Frame frequency, 1/n: Duty]



**Fig. 2.5.1 LCD clock control circuit structure**

# APPLICATION

## 2.5 LCD function

### (3) LCD display method

The 4551 Group has the LCD RAM area for the LCD display.

When “1” is written to a bit in the LCD RAM data, the display pixel which correspond to the bit automatically turns on.

Figure 2.5.2 shows the LCD RAM map.

Z X Y Bit	1											
	0				1				2			
	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
9	SEG1	SEG1	SEG1	SEG1	SEG9	SEG9	SEG9	SEG9	SEG17	SEG17	SEG17	SEG17
10	SEG2	SEG2	SEG2	SEG2	SEG10	SEG10	SEG10	SEG10	SEG18	SEG18	SEG18	SEG18
11	SEG3	SEG3	SEG3	SEG3	SEG11	SEG11	SEG11	SEG11	SEG19	SEG19	SEG19	SEG19
12	SEG4	SEG4	SEG4	SEG4	SEG12	SEG12	SEG12	SEG12				
13	SEG5	SEG5	SEG5	SEG5	SEG13	SEG13	SEG13	SEG13				
14	SEG6	SEG6	SEG6	SEG6	SEG14	SEG14	SEG14	SEG14				
15	SEG7	SEG7	SEG7	SEG7	SEG15	SEG15	SEG15	SEG15				
COM	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0

Note: — LCD display RAM is not assigned.

Fig. 2.5.2 LCD RAM map

### 2.5.2 Related registers

#### (1) LCD control register L1

The LCD duty and bias selection bits are assigned to bits 0 and 1. The LCD on/off bit is assigned to the bit 2.

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 to register A.

Table 2.5.2 shows the LCD control register L1.

Table 2.5.2 LCD control register L1

LCD control register L1		at reset : 00002		at power down : state retained	R/W
L13	Not used	0	This bit has no function, but read/write is enabled.		
		1			
L12	LCD on/off bit	0	Off		
		1	On		
L11	LCD duty and bias selection bits	L11	L10	Duty	Bias
		0	0	Not available	
		0	1	1/2	1/2
		1	0	1/3	1/3
L10		1	1	1/4	1/3

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

**(2) LCD control register L2**

Port P2 function and segment pin function of pins P20/SEG16–P23/SEG19 can be switched by setting register L2.

Set the contents of this register through register A with the **TL2A** instruction.

Table 2.5.3 shows the LCD control register L2.

**Table 2.5.3 LCD control register L2**

LCD control register L2		at reset : 11112		at power down : state retained	W
L23	P23/SEG19 function switch bit	0	SEG19		
		1	P23		
L22	P22/SEG18 function switch bit	0	SEG18		
		1	P22		
L21	P21/SEG17 function switch bit	0	SEG17		
		1	P21		
L20	P20/SEG16 function switch bit	0	SEG16		
		1	P20		

**Note:** “W” represents write enabled.

**(3) Timer control register W3**

The timer LC control bit is assigned to the bit 0, and the timer LC count source selection bit is assigned to the bit 1.

When the LCD display function is used, set the bit 1 to “0.”

Set the contents of this register through register A with the **TW3A** instruction. The **TAW3** instruction can be used to transfer the contents of register W3 to register A.

Table 2.5.4 shows the timer control register W3.

**Table 2.5.4 Timer control register W3**

Timer control register W3		at reset : 002		at power down : state retained	R/W
W31	Timer LC count source selection bit	0	Bit 3 of timer 2 is output (timer 2 count source divided by 16)		
		1	System clock (STCK)		
W30	Timer LC control bit	0	Stop (timer LC state retained)		
		1	Operating		

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



# APPLICATION

## 2.5 LCD function

### 2.5.3 LCD application examples

#### (1) LCD display

LCD display function can be used to display 80 pixels (maximum 4 common X 20 segment).

**Outline:** LCD can be displayed easily by using the LCD display function.

**Specifications:** 1/4 duty and 1/3 bias LCD is displayed by using LCD display panel example. Timer 2 is used for the LCD clock source, the sub-clock  $f(XCIN) = 32.768\text{ kHz}$  is used for the timer 2 clock source, and the frame frequency is set to 85 Hz.

Figure 2.5.3 shows the LCD display panel example, Figure 2.5.4 shows the segment assignment example, Figure 2.5.5 shows the LCD RAM assignment example, and Table 2.5.5 shows the frame frequency.

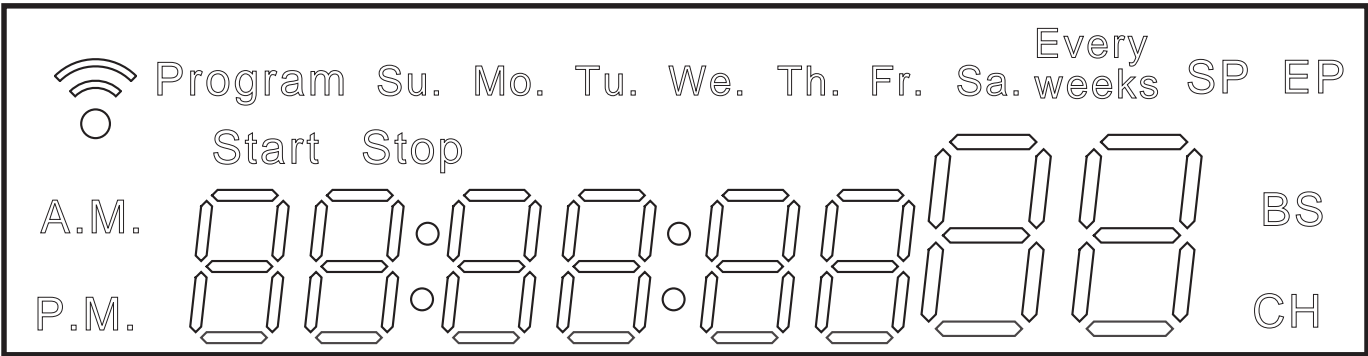


Fig. 2.5.3 LCD display panel example

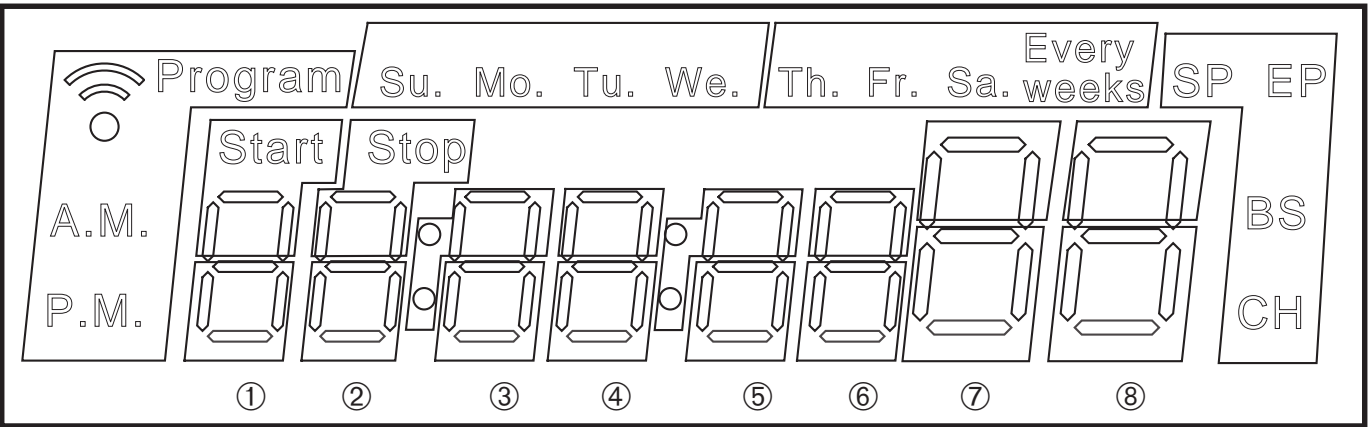


Fig. 2.5.4 Segment assignment example

Z	1											
X	0				1				2			
Y Bit	3	2	1	0	3	2	1	0	3	2	1	0
8	①-g	①-e	①-d	①-c	Start	①-f	①-b	①-a	We.	Tu.	Mo.	Su.
9	②-g	②-e	②-d	②-c	Stop	②-f	②-b	②-a	Sa.	Fr.	Tu.	SP
10	③-g	③-f	③-d	③-c	:	③-f	③-b	③-a	BS	CH	EP	Program
11	④-g	④-e	④-d	④-c	Unused	④-f	④-b	④-a	Every weeks	P.M.	A.M.	
12	⑤-g	⑤-e	⑤-d	⑤-c	:	⑤-f	⑤-b	⑤-a				
13	⑥-g	⑥-e	⑥-d	⑥-c	Unused	⑥-f	⑥-b	⑥-a				
14	⑦-g	⑦-e	⑦-d	⑦-c	Unused	⑦-f	⑦-b	⑦-a				
15	⑧-g	⑧-e	⑧-d	⑧-c	Unused	⑧-f	⑧-b	⑧-a				
COM	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0

Note: — LCD display RAM is not assigned.

Fig. 2.5.5 LCD RAM assignment example

## Initialization

## ① Operate Timer 2

Timer 2 count source is set to f(XCIN).

(Refer to section "2.3 Timer.")

## ② Set SEG16–SEG19

P20/SEG16–P23/SEG19 are set to output.

LCD control register L2 

b3				b0
0	0	0	0	0

 SEG16–SEG19 selected (**TL2A** instruction)

## ③ Stop Timer LC

Timer LC operation is stopped.

Timer control register W3 

b1	b0
X	0

 Timer LC stopped (**TW3A** instruction)

## ④ Set Timer LC

Timer LC value is set. (The formula is shown \*A below.)

Timer LC reload register RLC

Timer LC TLC "216" Timer count value 2 is set. (**TLCA** instruction)

## ⑤ Initialization of LCD Display RAM

LCD display RAM is initialized.

LCD display RAM

Initial data is set.

## ⑥ Set LCD Display Method

LCD duty and bias are set.

LCD control register L1 

b3				b0
X	X	1	1	1

 1/4 duty and 1/3 bias set. (**TL1A** instruction)

## ⑦ Set Timer LC Count Source

Timer LC count source is changed.

Timer control register W3 

b1	b0
0	1

 Count source is set to bit 3 of timer 2 .  
Timer LC start (**TW3A** instruction)

## ⑧ Display LCD

LCD display function is set to be valid.

LCD control register LC 

b3				b0
X	1	X	X	X

 LCD turned ON (**TL1A** instruction)

↓ to normal program

Display changed by rewriting LCD display RAM

\* A The timer LC count value when the frame frequency is set to 85.3 Hz is set as follows:

$$85.3 \text{ Hz} = \underbrace{(32.768 \text{ kHz})}_{\text{Sub-clock}} \times \underbrace{\frac{1}{16}}_{\text{Bit 3 of timer 2}} \times \underbrace{\frac{1}{(2+1)}}_{\text{Timer LC}} \times \frac{1}{2} \times \underbrace{\frac{1}{4}}_{\text{Duty ratio}}$$

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

Fig. 2.5.6 Initial setting example

# APPLICATION

## 2.5 LCD function

**Table 2.5.5 Frame frequency**

LCD clock = XcIN: 32.768 kHz				LCD clock = ORCLK: 910 kHz/12 or = ORCLK:3.64 MHz/48				LCD clock = ORCLK: 910 kHz/24 or = ORCLK:3.64 MHz/96			
LCD timer value	Duty			LCD timer value	Duty			LCD timer value	Duty		
	1/2	1/3	1/4		1/2	1/3	1/4		1/2	1/3	1/4
0	512 Hz	341 Hz	256 Hz	0	1185 Hz	790 Hz	592 Hz	0	592 Hz	395 Hz	296 Hz
1	256 Hz	170 Hz	128 Hz	1	592 Hz	395 Hz	296 Hz	1	296 Hz	197 Hz	148 Hz
2	171 Hz	114 Hz	85 Hz	2	395 Hz	263 Hz	197 Hz	2	197 Hz	132 Hz	99 Hz
3	128 Hz	85 Hz	64 Hz	3	296 Hz	197 Hz	148 Hz	3	148 Hz	99 Hz	74 Hz
4	102 Hz	68 Hz	51 Hz	4	237 Hz	158 Hz	118 Hz	4	118 Hz	79 Hz	59 Hz
5	85 Hz	57 Hz	43 Hz	5	197 Hz	132 Hz	99 Hz	5	99 Hz	66 Hz	49 Hz
6	73 Hz	49 Hz	37 Hz	6	169 Hz	113 Hz	85 Hz	6	85 Hz	56 Hz	42 Hz
7	64 Hz	42 Hz	32 Hz	7	148 Hz	99 Hz	74 Hz	7	74 Hz	49 Hz	37 Hz
8	57 Hz	38 Hz	28 Hz	8	131 Hz	88 Hz	66 Hz	8	66 Hz	44 Hz	33 Hz
9	51 Hz	34 Hz	27 Hz	9	118 Hz	79 Hz	59 Hz	9	59 Hz	39 Hz	30 Hz
10	47 Hz	31 Hz	23 Hz	10	108 Hz	72 Hz	54 Hz	10	54 Hz	36 Hz	27 Hz
11	43 Hz	28 Hz	21 Hz	11	99 Hz	66 Hz	49 Hz	11	49 Hz	33 Hz	25 Hz
12	39 Hz	26 Hz	20 Hz	12	91 Hz	61 Hz	46 Hz	12	46 Hz	30 Hz	23 Hz
13	37 Hz	24 Hz	18 Hz	13	85 Hz	56 Hz	42 Hz	13	42 Hz	28 Hz	21 Hz
14	34 Hz	23 Hz	17 Hz	14	79 Hz	53 Hz	39 Hz	14	39 Hz	26 Hz	20 Hz
15	32 Hz	21 Hz	16 Hz	15	74 Hz	49 Hz	37 Hz	15	37 Hz	25 Hz	18 Hz

**Note:** Values in the table shows the frame frequency (however, the values are rounded off to the decimal points).

### 2.5.4 Notes on use

#### (1) Timer LC count source

Stop each timer counting to change timer LC count source.

## 2.6 Power down function

The 4551 Group has the clock operation mode and RAM back-up mode for the power down function. The 4551 Group enters 12 kinds of state which includes the reset state to reduce the power dissipation. Figure 2.6.1 shows the state transition, and Figure 2.6.2 shows the oscillation stabilizing time.

In this section, the clock control function, each power down function, related register and application example for the power down function are described.

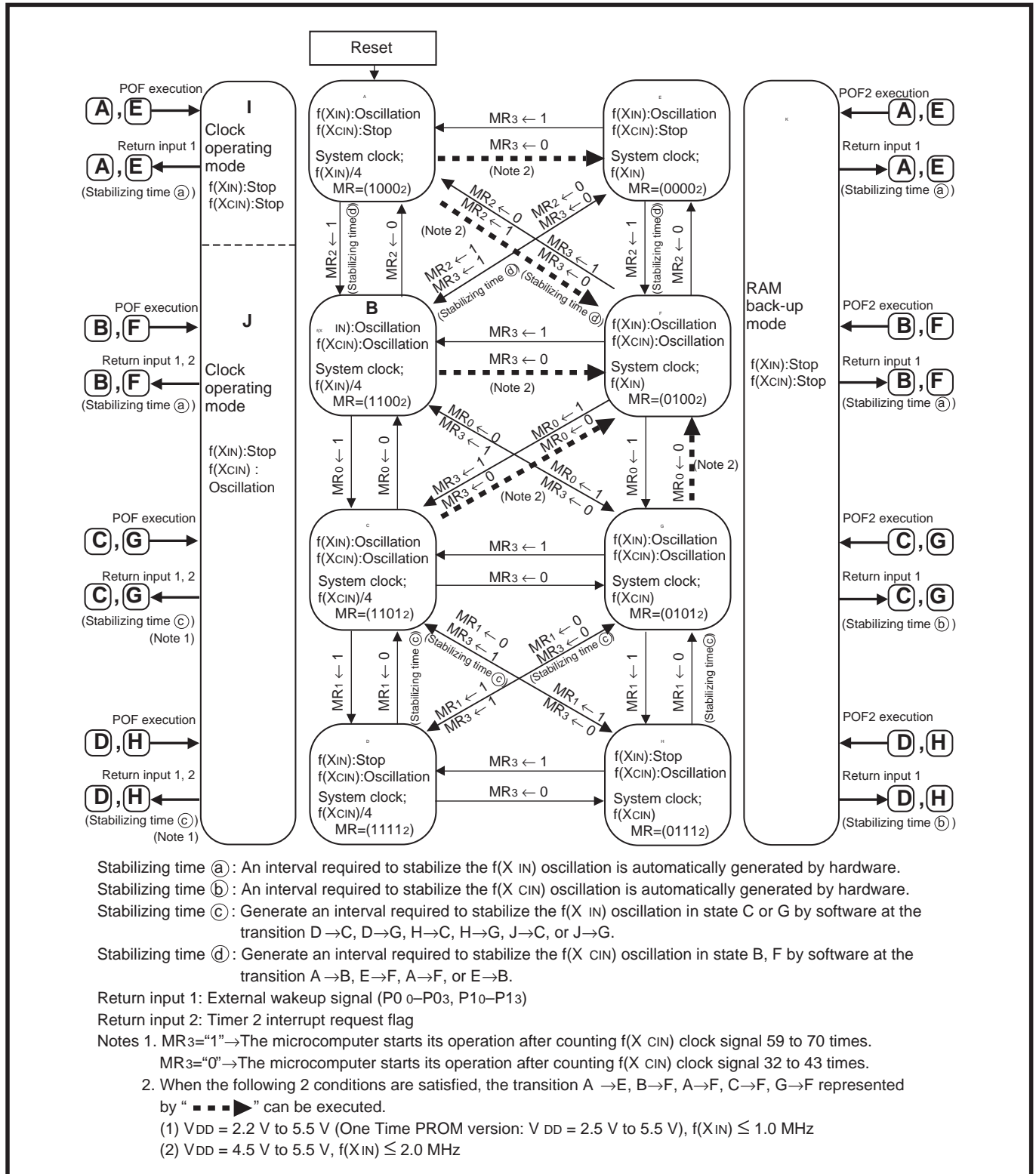


Fig. 2.6.1 State transition

# APPLICATION

## 2.6 Power down function

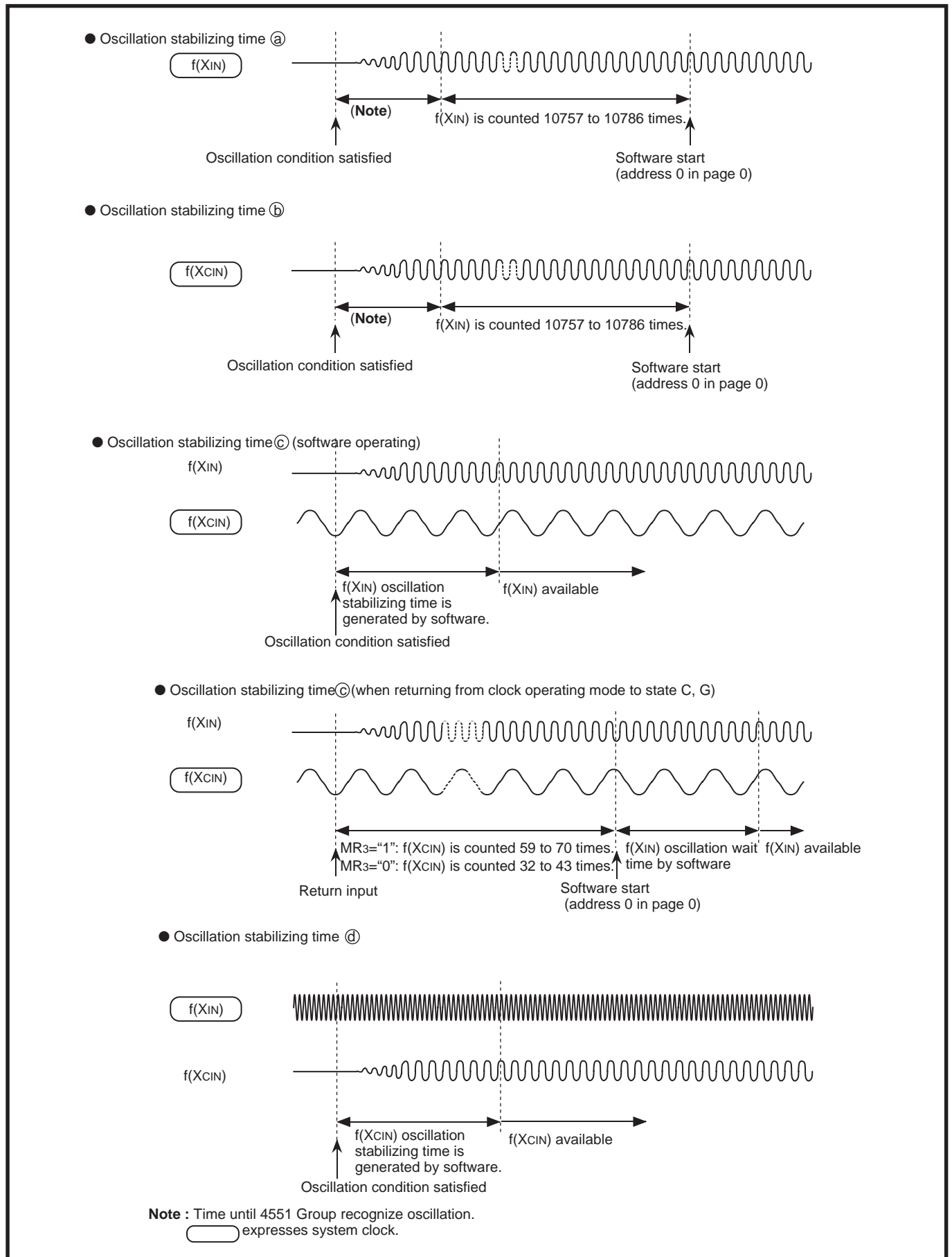


Fig. 2.6.2 Oscillation stabilizing time in each mode

**2.6.1 Clock control function**

The 4551 Group can reduce the power dissipation by controlling oscillation with the clock control register MR. Refer to section “3.1 Electrical characteristics” for the oscillation frequency and power dissipation.

In Figure 2.6.1, directions in which state transition can be executed are expressed by the arrow. For example, execute the transition A → B → ((XCIN) oscillation stabilizing wait) → G → H to execute the transition from the state A (MR3MR2MR1MR0 = 1000) after reset to the state H (MR3MR2MR1MR0 = 0111).

**Note:** Do not stop the oscillation circuit selected by clock selection bit (MR0).

Note the stop of the oscillation circuit selected with the clock selection bit (MR0) if the following setting is performed.

Example 1: (MR3MR2MR1MR0) = (X0X1) (f(XCIN) selected, f(XCIN) oscillation stop)

Example 2: (MR3MR2MR1MR0) = (XX10) (f(XIN) selected, f(XIN) oscillation stop)

X: “0” or “1.”

**2.6.2 Power down function**

When the **POF** instruction or **POF2** instruction is executed just after the **EPOF** instruction, system enters the power down state. Table 2.6.1 shows the internal state at each mode. Also, Table 2.6.2 shows the return source from this state.

**(1) Clock operating mode**

In this mode, current dissipation can be reduced by stopping XIN-XOUT oscillation and system clock with the states of RAM, reset circuit, XCIN-XCOUT oscillation, LCD display and timer 2 retained.

**(2) RAM back-up mode**

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

# APPLICATION

## 2.6 Power down function

**Table 2.6.1 Functions and states retained at RAM back-up mode and the clock operating mode**

Function	Clock operating	RAM back-up
Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) ( <b>Note 2</b> )	X	X
Contents of RAM	O	O
Port level	O	O
Clock control register MR	O	O
Timer control register W1	X	X
Timer control registers W2, W3	O	O
Interrupt control register V1	X	X
Interrupt control register I1	O	O
Carrier wave control registers and flag (C1, C2, CR)	X	X
LCD display function	O	( <b>Note 3</b> )
LCD control registers L1, L2	O	O
Timer LC	O	( <b>Note 4</b> )
Timer 1 function	X	X
Timer 2 function	O	O
External 0 interrupt request flag (EXF0)	X	X
Timer 1 interrupt request flag (T1F)	X	X
Timer 2 interrupt request flag (T2F)	O	O
Watchdog timer flag (WDF)	O	X
Watchdog timer enable flag (WEF)	O	O
Interrupt enable flag (INTE)	X	X
General-purpose register V2	X	X

**Notes 1:** "O" represents that the function can be retained, and "X" represents that the function is initialized.

Registers and flags other than the above are undefined at power down, and set an initial value after returning from power down state.

**2:** The stack pointer (SP) points the level of the stack register and is initialized to "1112" at power down.

**3:** LCD is turned off.

**4:** The state of timer is undefined.

Table 2.6.2 Return source and return condition

Return source		Return condition	Remarks
External wakeup signal	Ports P0, P1	Returns by an external falling edge input ("H"→"L").	Port P0 shares the falling edge detection circuit with port P1. The key-on wakeup function of port P0 is always valid. The only key-on wakeup function of the port P1 bit of which the pull-up transistor is turned on is valid. Set all the port using the key-on wakeup function to "H" level before going into the power down state.
	Timer 2 interrupt request flag	Returns by timer 2 underflow and setting T2F flag to "1."	The timer 2 interrupt request flag (T2F) can be used only when system returns from the clock operating mode ( <b>POF</b> instruction execution). However, if the <b>POF</b> and <b>POF2</b> instructions are executed while the T2F flag = "1", its operation is recognized as the return condition and system returns from the clock operating mode.

**Note:** P1 pin has the pull-up transistor which can be turned on/off by software.

### (3) Start condition identification

When system returns from both power down and reset, software is started from address 0 in page 0.

The start condition (warm start or cold start) can be identified by examining the state of the power down flag (P) with the **SNZP** instruction. Also, warm start condition (timer 2 or external wakeup signal) can be identified by the state of the T2F flag. Table 2.6.3 shows the start condition identification, and Figure 2.6.3 shows the start condition identified example.

Table 2.6.3 Start condition identification

Return condition	P flag	T2F flag
External wakeup signal input	1	1
Timer 2 interrupt request flag	1	1
Reset	0	0

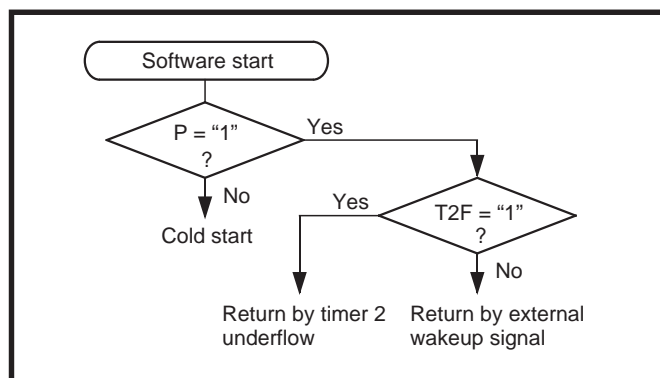


Fig. 2.6.3 Start condition identified example



# APPLICATION

## 2.6 Power down function

### 2.6.3 Related register

#### (1) Clock control register MR

Clock control register MR controls the system clock.

Set the contents of this register through register A with the **TMRA** instruction. The **TAMR** instruction can be used to transfer the contents of register MR to register A.

Table 2.6.4 shows the clock control register MR.

**Table 2.6.4 Clock control register MR**

Clock control register MR		at reset : 10002	at power down : state retained	R/W
MR3	System clock (STCK) selection bit	0	MR0=0    f(XIN)	
			MR0=1    f(XCIN)	
		1	MR0=0    f(XIN)/4	
			MR0=1    f(XCIN)/4	
MR2	f(XCIN) oscillation circuit control bit	0	f(XCIN) oscillation stop, ports D6 and D7 selected	
		1	f(XCIN) oscillation enabled, ports D6 and D7 not selected	
MR1	f(XIN) oscillation circuit control bit	0	Oscillation enabled	
		1	Oscillation stop	
MR0	Clock selection bit	0	f(XIN)	
		1	f(XCIN)	

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

#### (2) Pull-up control register PU0

Pull-up control register PU0 controls the pull-up function and key-on wakeup function.

Set the contents of this register through register A with the **TPU0A** instruction. The **TAPU0** instruction can be used to transfer the contents of register PU0 to register A.

Table 2.6.5 shows the pull-up control register PU0.

**Table 2.6.5 Pull-up control register PU0**

Pull-up control register PU0		at reset : 00002	at power down : state retained	R/W
PU03	Port P13 pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup	
		1	Pull-up transistor ON, key-on wakeup	
PU02	Port P12 pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup	
		1	Pull-up transistor ON, key-on wakeup	
PU01	Port P11 pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup	
		1	Pull-up transistor ON, key-on wakeup	
PU00	Port P10 pull-up transistor control bit	0	Pull-up transistor OFF, no key-on wakeup	
		1	Pull-up transistor ON, key-on wakeup	

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

## 2.6.4 Power down function application example

## (1) Clock display

A clock which is high-accuracy and low-power dissipation can be set up by using a 32.768 kHz quartz-crystal as a sub-clock and executing the **POF** instruction.

**Outline:** The power dissipation can be reduced by using the **POF** instruction.

**Specifications:** Time is displayed by the LCD and a 32.768 kHz quartz-crystal oscillator. The main routine is executed by key input.

Figure 2.6.4 shows the software setting example.

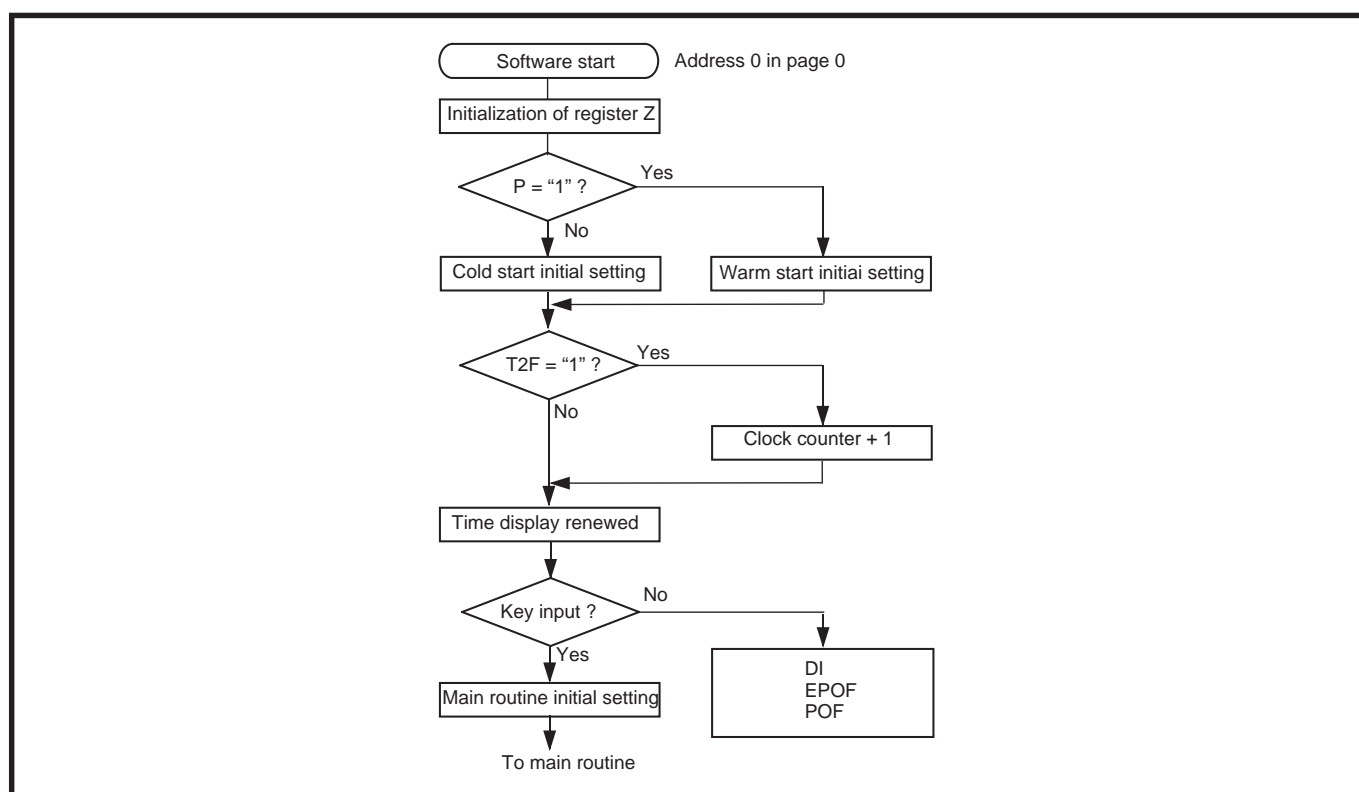


Fig. 2.6.4 Software setting example

## 2.6.5 Notes on use

## (1) Key-on wakeup function

After setting ports (P1 specified with register PU0 and P0) which key-on wakeup function is valid to "H," execute the **POF** or **POF2** instruction.

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

## (2) Power down instruction

Execute the **POF** or **POF2** instruction immediately after executing the **EPOF** instruction to enter the power down state.

Note that system cannot enter the power down state when executing only the **POF** or **POF2** instruction. Be sure to disable interrupts by executing the **DI** instruction before executing the **EPOF** instruction.

## (3) POF2 instruction

If the **POF** and **POF2** instructions are executed while the T2F flag = "1," its operation is recognized as the return condition and system returns from power down state.

# APPLICATION

## 2.7 Reset

### 2.7 Reset

System reset is performed by applying “L” level to the  $\overline{\text{RESET}}$  pin for 1 machine cycle or more when the following conditions are satisfied;

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

Then when “H” level is applied to the  $\overline{\text{RESET}}$  pin, the software starts from address 0 in page 0 after elapsing the internal oscillation stabilizing time ( $f(\text{XIN})$  is counted 10757 to 10786 machine cycles). Figure 2.7.3 shows the oscillation stabilizing time.

#### 2.7.1 Reset circuit

The 4551Group has the power-on reset circuit and voltage drop detection circuit.

##### (1) Power-on reset

Reset can be performed automatically at power on (power-on reset) by the built-in power-on reset circuit. When the built-in power-on reset circuit is used, set the time until the supply voltage rises to the minimum operation voltage to 100  $\mu\text{s}$  or less. When the rising time exceeds 100  $\mu\text{s}$ , connect a capacitor between the  $\overline{\text{RESET}}$  pin and Vss at the shortest distance, input “L” level to  $\overline{\text{RESET}}$  pin until the supply voltage reaches the minimum operation voltage.

Figure 2.7.1 shows the power-on reset example and Figure 2.7.2 shows the reset circuit example when the supply voltage rising time exceeds 100  $\mu\text{s}$ .

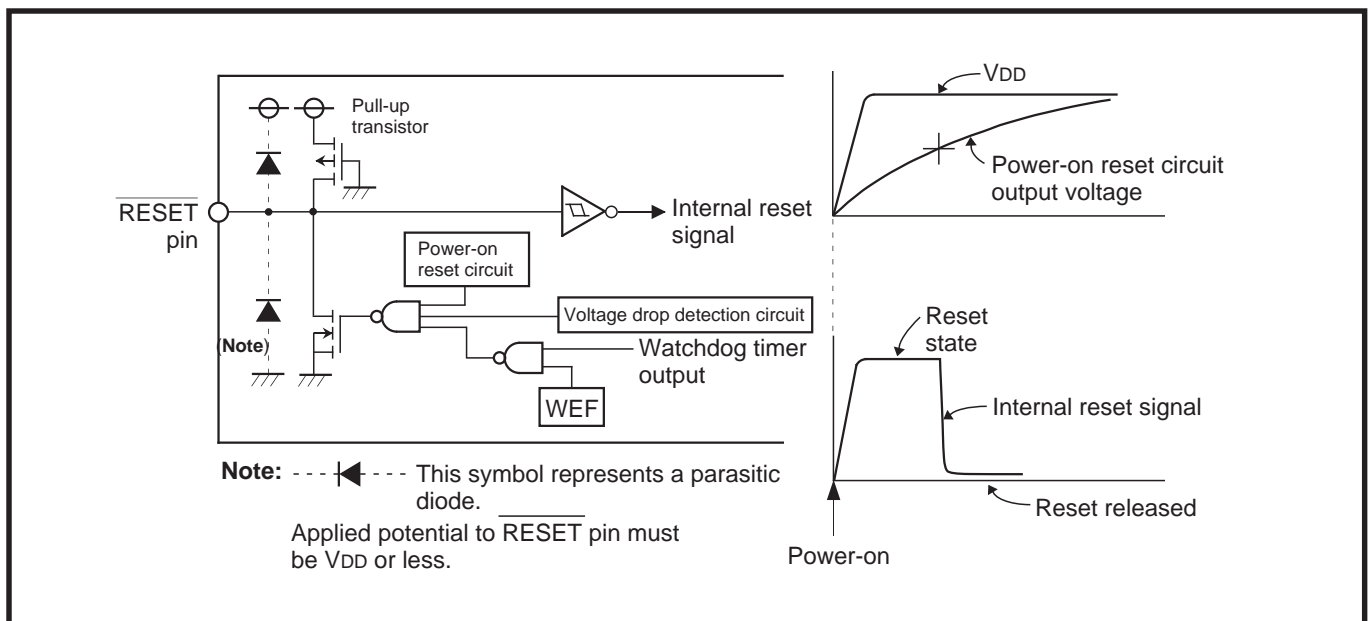


Fig. 2.7.1 Power-on reset circuit example

Calculate the capacitor value to input "L" level until the supply voltage reaches the minimum operation voltage.

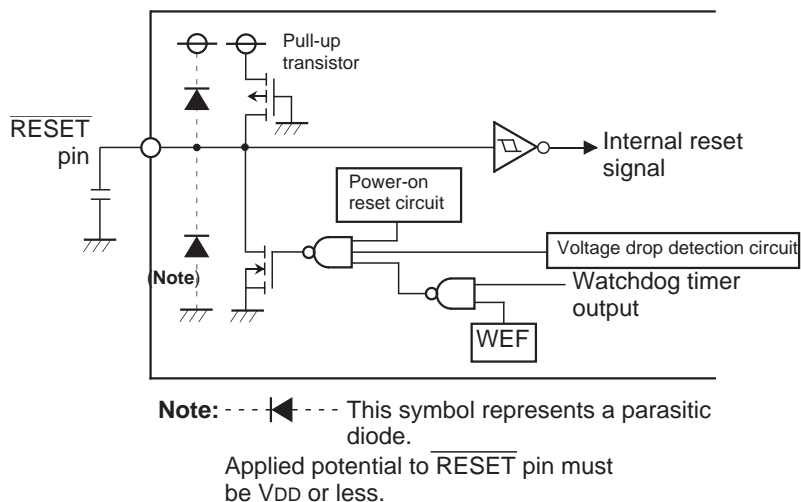


Fig. 2.7.2 Reset circuit example when the supply voltage rising time exceeds 100  $\mu\text{s}$

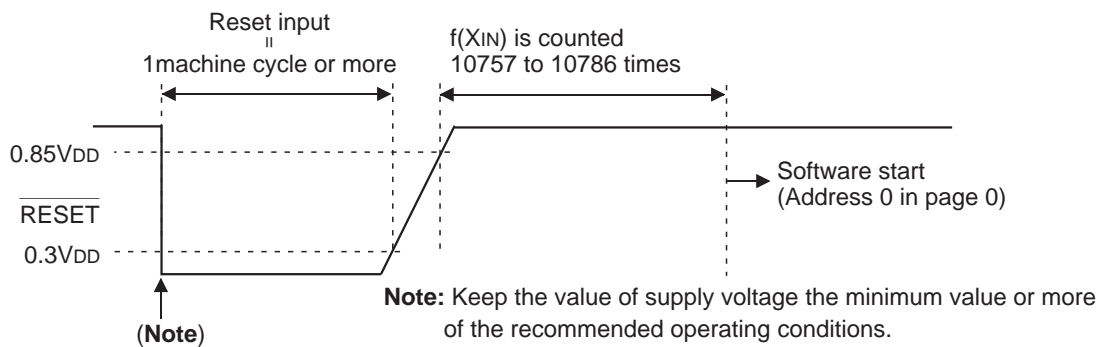


Fig. 2.7.3 Oscillation stabilizing time after system is released from reset

# APPLICATION

## 2.7 Reset

### 2.7.2 Internal state at reset

Figure 2.7.4 shows the internal state at reset. The contents of timers, registers, flags and RAM other than shown in Figure 2.7.4 are undefined, so set them to initial values.

• Program counter (PC) .....	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
Address 0 in page 0 is set to program counter.		
• Interrupt enable flag (INTE) .....	0	(Interrupt disabled)
• Power down flag (P) .....	0	
• External 0 interrupt request flag (EXF0) .....	0	
• Interrupt control register V1 .....	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	
• Watchdog timer flag (WDF) .....	0	
• Watchdog timer enable flag (WEF) .....	0	
• Timer control register W1 .....	0 0 0 0	(Prescaler stopped)
• Timer control register W2 .....	0 0 0 0	(Timer 1 stopped)
• Timer control register W3 .....	0 0	(Timer LC stopped)
• Clock control register MR .....	1 0 0 0	
• Carrier wave selection register C1 .....	0 1 1 1	
• Carrier wave output control register C2 .....	0	
• Carrier wave generating control flag CR .....	0	(Carrier wave output disabled)
• LCD control register L1 .....	0 0 0 0	(LCD off)
• LCD control register L2 .....	1 1 1 1	(Port P2 selected)
• Pull-up control register PU0 .....	0 0 0 0	
• General-purpose register V2 .....	0 0 0 0	
• Carry flag (CY) .....	0	
• Register A .....	0 0 0 0	
• Register B .....	0 0 0 0	
• Register D .....	X X X	
• Register E .....	X X X X X X X X	
• Data pointer X .....	0 0 0 0	
• Data pointer Y .....	0 0 0 0	
• Data pointer Z .....	X X	
• Stack pointer (SP) .....	1 1 1	

"X" represents undefined.

Fig. 2.7.4 Internal state at reset

### 2.7.3 Voltage drop detection circuit

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

Figure 2.7.5 shows the voltage drop detection reset circuit, and Figure 2.7.6 shows the operation waveform example of the voltage drop detection circuit.

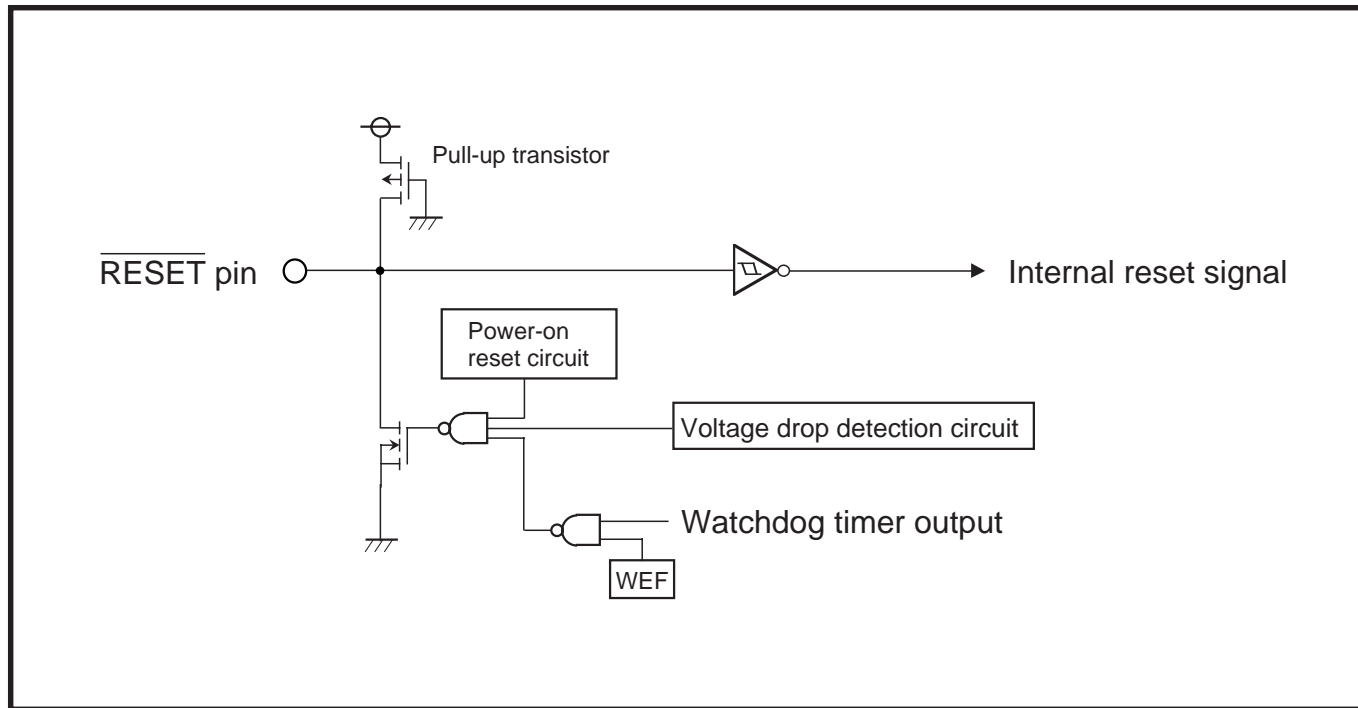


Fig. 2.7.5 Voltage drop detection reset circuit

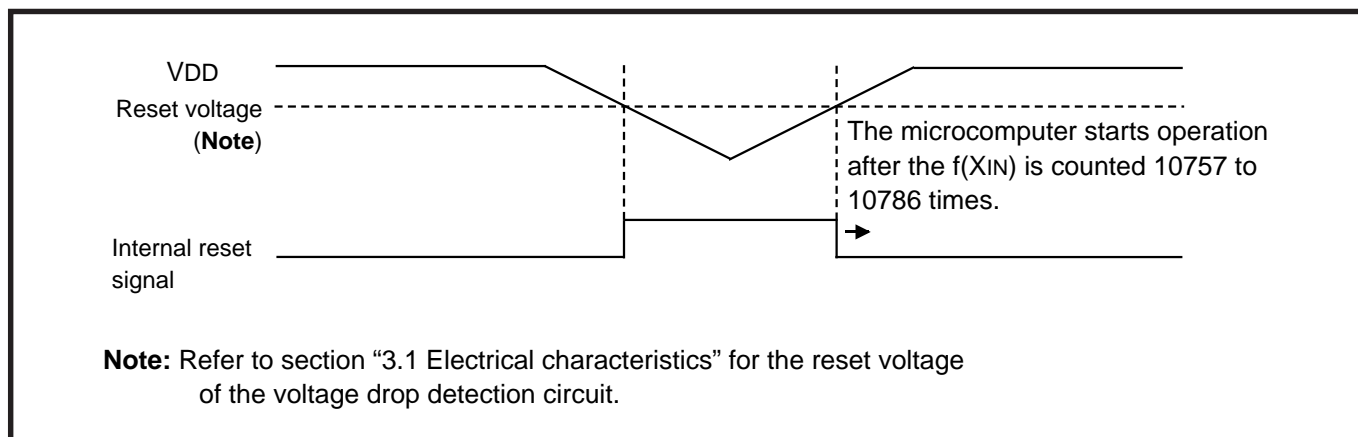


Fig. 2.7.6 Voltage drop detection circuit operation waveform

# APPLICATION

## 2.8 Oscillation circuit

### 2.8 Oscillation circuit

The 4551 Group has an internal oscillation circuit to produce the clock required for microcomputer operation. The clock signal  $f(XIN)$  is obtained by connecting a ceramic resonator to  $XIN$  pin and  $XOUT$  pin. The clock signal  $f(XCIN)$  is obtained by connecting a quartz-crystal oscillator to  $XCIN$  pin and  $XCOUT$  pin.

#### 2.8.1 Oscillation circuit

##### (1) $f(XIN)$ clock generating circuit

The clock signal  $f(XIN)$  is obtained by connecting a ceramic resonator externally.

Connect this external circuit to pins  $XIN$  and  $XOUT$  at the shortest distance. A feed-back resistor is built-in between  $XIN$  pin and  $XOUT$  pin.

Figure 2.8.1 shows an example of an oscillation circuit connecting a ceramic resonator externally. Keep the maximum value of oscillation frequency within the range listed Table 2.8.1.

##### (2) $f(XCIN)$ clock generating circuit

The clock signal  $f(XCIN)$  is obtained by connecting a quartz-crystal externally.

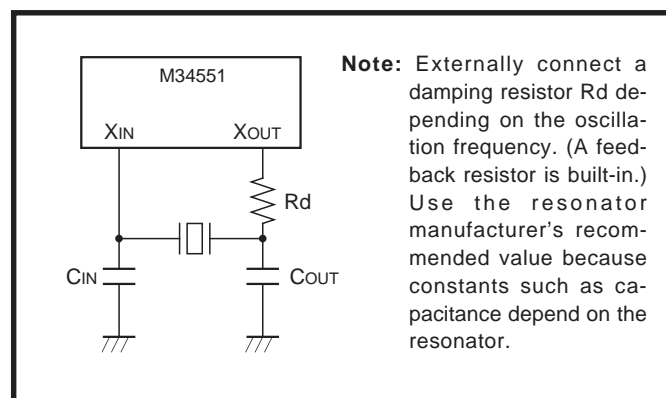
Connect this external circuit to pins  $XCIN$  and  $XCOUT$  at the shortest distance. A feed-back resistor is built-in between  $XCIN$  pin and  $XCOUT$  pin.

Figure 2.8.2 shows an example of an oscillation circuit connecting a quartz-crystal externally.

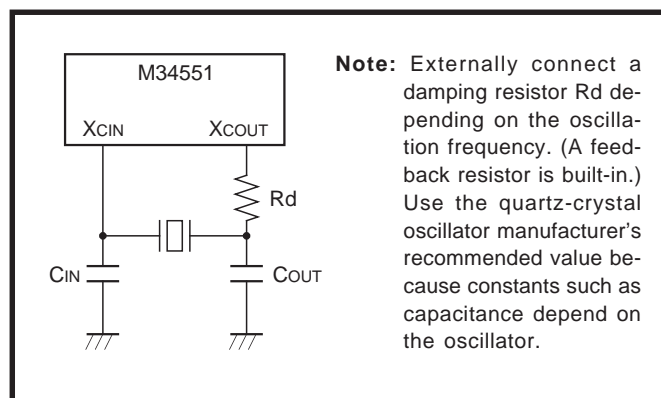
**Table 2.8.1 Maximum value of oscillation frequency and supply voltage**

Supply voltage	(System clock)	Oscillation frequency
4.5 V to 5.5 V	$(f(XIN)/4)$	8.0 MHz
4.5 V to 5.5 V	$(f(XIN))$	2.0 MHz
2.2 V to 5.5 V ( <b>Note</b> )	$(f(XIN)/4)$	4.0 MHz
2.5 V to 5.5 V ( <b>Note</b> )	$(f(XIN))$	1.0 MHz

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



**Fig. 2.8.1 Oscillation circuit example connecting ceramic resonator externally**



**Fig. 2.8.2 Oscillation circuit example connecting quartz-crystal externally**

### 2.8.2 Oscillation operation

System clock is supplied to CPU and peripheral device as the standard clock for the microcomputer operation. For the 4551 Group, the clock ( $f(XIN)$ ),  $(f(XIN)/4)$ ,  $(f(XCIN))$ , or  $(f(XCIN))/4$  which is supplied from the oscillation circuit is selected with the register MR.

Figure 2.8.3 shows the structure of the clock control circuit.

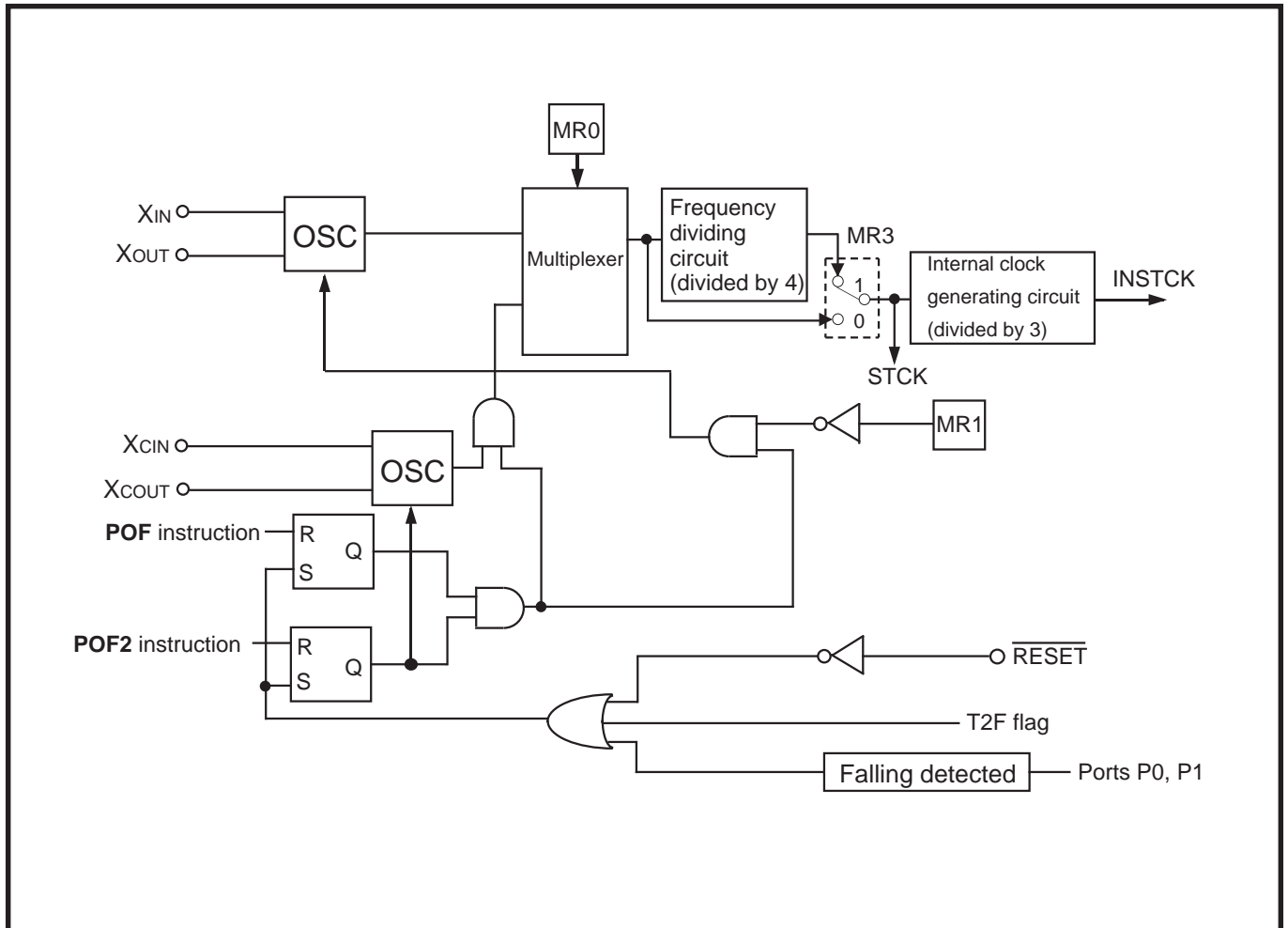


Fig. 2.8.3 Structure of clock control circuit

### 2.8.3 Notes on use

#### (1) Value of a part connected to an oscillator

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



# APPLICATION

## 2.8 Oscillation circuit

---

### *MEMO*



## CHAPTER 3

# APPENDIX

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

# APPENDIX

## 3.1 Electrical characteristics

---

### 3.1 Electrical characteristics

#### 3.1.1 Absolute maximum ratings

**Table 3.1.1 Absolute maximum ratings**

Symbol	Parameter	Conditions	Ratings	Unit
V <sub>DD</sub>	Supply voltage		−0.3 to 7.0	V
V <sub>I</sub>	Input voltage P0, P1, P2, $\overline{\text{RESET}}$ , X <sub>IN</sub> , X <sub>CIN</sub>		−0.3 to V <sub>DD</sub> +0.3	V
V <sub>O</sub>	Output voltage P0, P1, D	Output transistors in cut-off state	−0.3 to V <sub>DD</sub> +0.3	V
V <sub>O</sub>	Output voltage CARR, X <sub>OUT</sub> , X <sub>COUT</sub>		−0.3 to V <sub>DD</sub> +0.3	V
V <sub>O</sub>	Output voltage SEG, COM		−0.3 to V <sub>DD</sub> +0.3	V
P <sub>d</sub>	Power dissipation		300	mW
T <sub>opr</sub>	Operating temperature range		−20 to 70	°C
T <sub>stg</sub>	Storage temperature range		−40 to 125	°C

## 3.1.2 Recommended operating conditions

Table 3.1.2 Recommended operating conditions

(Mask ROM version:  $T_a = -20\text{ }^{\circ}\text{C}$  to  $70\text{ }^{\circ}\text{C}$ ,  $V_{DD} = 2.2\text{ V}$  to  $5.5\text{ V}$ , unless otherwise noted)(One Time PROM version:  $T_a = -20\text{ }^{\circ}\text{C}$  to  $70\text{ }^{\circ}\text{C}$ ,  $V_{DD} = 2.5\text{ V}$  to  $5.5\text{ V}$ , unless otherwise noted)

Symbol	Parameter		Conditions	Limits			Unit
				Min.	Typ.	Max.	
V <sub>DD</sub>	Supply voltage	Mask ROM version	f(X <sub>IN</sub> ) ≤ 4.0 MHz, ceramic resonator, STCK=f(X <sub>IN</sub> )/4	2.2		5.5	V
			f(X <sub>IN</sub> ) ≤ 1.0 MHz, ceramic resonator, STCK=f(X <sub>IN</sub> )				
		One Time PROM version	f(X <sub>IN</sub> ) ≤ 4.0 MHz, ceramic resonator, STCK=f(X <sub>IN</sub> )/4	2.5		5.5	
			f(X <sub>IN</sub> ) ≤ 1.0 MHz, ceramic resonator, STCK=f(X <sub>IN</sub> )				
			f(X <sub>IN</sub> ) ≤ 8.0 MHz, ceramic resonator, STCK=f(X <sub>IN</sub> )/4	4.5		5.5	
			f(X <sub>IN</sub> ) ≤ 2.0 MHz, ceramic resonator, STCK=f(X <sub>IN</sub> )				
V <sub>RAM</sub>	RAM back-up voltage	RAM back-up	2.0		5.5	V	
V <sub>SS</sub>	Supply voltage			0		V	
V <sub>IH</sub>	“H” level input voltage P0, P1, P2		0.8V <sub>DD</sub>		V <sub>DD</sub>	V	
V <sub>IH</sub>	“H” level input voltage X <sub>IN</sub>		0.7V <sub>DD</sub>		V <sub>DD</sub>	V	
V <sub>IH</sub>	“H” level input voltage RESET		0.85V <sub>DD</sub>		V <sub>DD</sub>	V	
V <sub>IH</sub>	“H” level input voltage INT		0.8V <sub>DD</sub>		V <sub>DD</sub>	V	
V <sub>IL</sub>	“L” level input voltage P0, P1, P2		0		0.3V <sub>DD</sub>	V	
V <sub>IL</sub>	“L” level input voltage X <sub>IN</sub>		0		0.3V <sub>DD</sub>	V	
V <sub>IL</sub>	“L” level input voltage RESET		0		0.3V <sub>DD</sub>	V	
V <sub>IL</sub>	“L” level input voltage INT		0		0.2V <sub>DD</sub>	V	
I <sub>OL</sub> (peak)	“L” level peak output current P0, P1, D <sub>0</sub> –D <sub>7</sub> , CARR	V <sub>DD</sub> =5.0 V			10	mA	
		V <sub>DD</sub> =3.0 V			4		
I <sub>OL</sub> (avg)	“L” level average output current P0, P1, D <sub>0</sub> –D <sub>7</sub> , CARR ( <b>Note</b> )	V <sub>DD</sub> =5.0 V			5	mA	
		V <sub>DD</sub> =3.0 V			2		
I <sub>OH</sub> (peak)	“H” level peak output current CARR	V <sub>DD</sub> =5.0 V			–30	mA	
		V <sub>DD</sub> =3.0 V			–15		
I <sub>OH</sub> (avg)	“H” level average output current CARR ( <b>Note</b> )	V <sub>DD</sub> =5.0 V			–15	mA	
		V <sub>DD</sub> =3.0 V			–7		
f(X <sub>CIN</sub> )	f(X <sub>CIN</sub> ) clock frequency	Quartz-crystal oscillator			50	kHz	
V <sub>DET</sub>	Voltage drop detection circuit	Mask ROM version	1.15		2.15	V	
			1.30	1.65	2.00		
		One Time PROM version	1.00		2.00		
			1.15	1.50	1.85		
T <sub>PON</sub>	Valid power supply rising time for power-on reset circuit	Mask ROM version V <sub>DD</sub> = 0 to 2.2 V One Time PROM version V <sub>DD</sub> = 0 to 2.5 V			100	μs	

**Note:** The average output current is the average current value at the 100 ms interval.

# APPENDIX

## 3.1 Electrical characteristics

### 3.1.3 Electrical characteristics

**Table 3.1.3 Electrical characteristics**

(Mask ROM version: Ta = -20 °C to 70 °C, VDD = 2.2 V to 5.5 V, unless otherwise noted)

(One Time PROM version: Ta = -20 °C to 70 °C, VDD = 2.5 V to 5.5 V, unless otherwise noted)

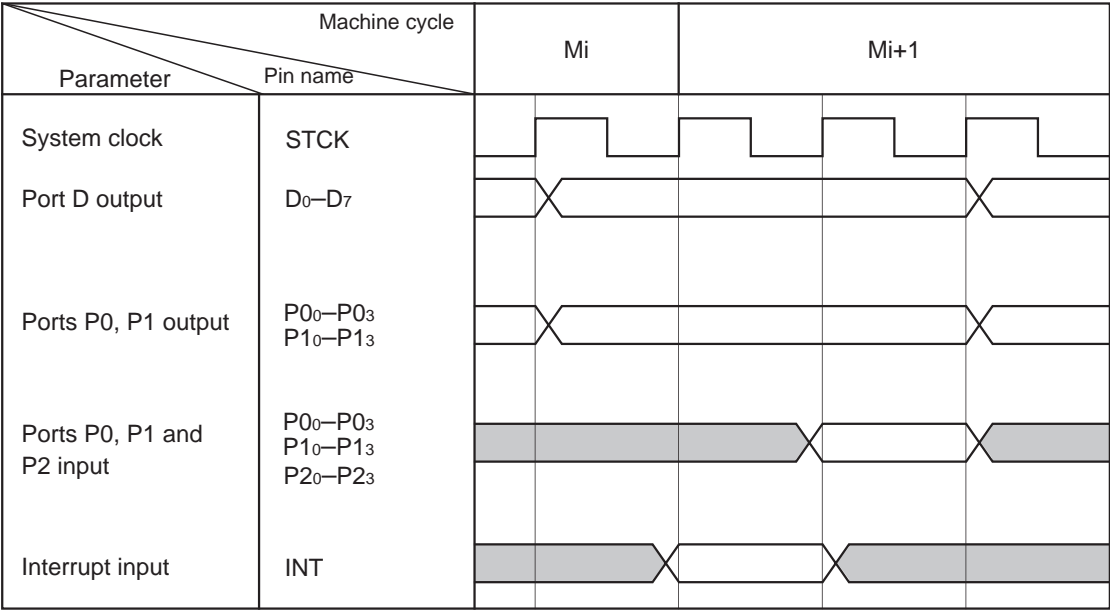
Symbol	Parameter		Test conditions		Limits			Unit
					Min.	Typ.	Max.	
VOL	“L” level output voltage P0, P1, D0–D7, CARR, RESET		IOL = 5 mA	VDD = 5.0 V			0.9	V
			IOL = 2 mA	VDD = 3.0 V			0.9	
VOH	“H” level output voltage CARR		IOH = -15 mA	VDD = 5.0 V	2.4			V
			IOH = -7 mA	VDD = 3.0 V	1.0			
IiH	“H” level input current P0, P1, P2, RESET		VI = VDD (Note 1)				1	μA
IiL	“L” level input current P1, P2		VI = 0 V (Note 1)		-1			μA
Ioz	Output current at off-state D0–D7		VO = VDD				1	μA
IDD	Supply current (Note 2)	at active high-speed mode while LCD is operating (SEG0–SEG15)	VDD = 5.0 V, f(XCIN) = 32 kHz, f(XIN) = 8 MHz STCK = f(XIN)/4			2.5	5.0	mA
			VDD = 5.0 V f(XCIN) = 32 kHz STCK = f(XIN)	f(XIN) = 2 MHz		2.3	4.6	
				f(XIN) = 1 MHz		1.4	2.8	
			VDD = 3.0 V, f(XCIN) = 32 kHz, f(XIN) = 4 MHz STCK = f(XIN)/4			0.7	1.4	
			VDD = 3.0 V f(XCIN) = 32 kHz STCK = f(XIN)	f(XIN) = 1 MHz		0.6	1.2	
				f(XIN) = 500 kHz		0.4	0.8	
		at active low-speed mode while LCD is operating (SEG0–SEG15)	VDD = 5.0 V f(XIN) = stop f(XCIN) = 32 kHz	STCK = f(XCIN)/4		70	140	μA
				STCK = f(XCIN)		90	180	
			VDD = 3.0 V f(XIN) = stop f(XCIN) = 32 kHz	STCK = f(XCIN)/4		30	60	
				STCK = f(XCIN)		40	80	
		at clock operating mode while LCD is operating (SEG0–SEG15)	f(XIN) = stop f(XCIN) = 32 kHz Ta=25 °C	VDD = 5.0 V		27.5	60	μA
				VDD = 3.0 V		10	17.5	
			f(XIN) = stop f(XCIN) = 32 kHz	VDD = 5.0 V			65	
				VDD = 3.0 V			20	
		at RAM back-up mode	f(XIN) = stop, f(XCIN) = stop, Ta = 25 °C			0.1	1.0	μA
			f(XIN) = stop, f(XCIN) = stop				10	
RPH	Pull-up resistor value	P0, P1	VDD = 5.0 V, VI = 0 V		20	50	125	kΩ
			VDD = 3.0 V, VI = 0 V		40	100	250	
		RESET	VDD = 5.0 V, VI = 0 V		12	30	70	kΩ
			VDD = 3.0 V, VI = 0 V		25	60	130	
VT+ – VT–	Hysteresis	INT	VDD = 5.0 V			0.5		V
			VDD = 3.0 V			0.4		
		RESET	VDD = 5.0 V			1.5		V
			VDD = 3.0 V			0.6		
RCOM	COM output impedance		VDD = 5.0 V			1.3	6.5	kΩ
			VDD = 3.0 V			1.6	8	
RSEG	SEG output impedance		VDD = 5.0 V			1.8	9	kΩ
			VDD = 3.0 V			2.2	11	
RVLC	LCD power supply internal resistor value (Note 3)		Impedance between V <sub>LC3</sub> and V <sub>SS</sub> Ta=25 °C		300	600	1200	kΩ

**Notes 1:** In this case, the pull-up transistor of port P1 is turned off and the port P2 function is selected by software.

**2:** The current value includes the current dissipation of the LCD power supply internal resistor (RVLC).

**3:** V<sub>LC3</sub>=V<sub>DD</sub>.

3.1.4 Basic timing diagram



# APPENDIX

## 3.2 Typical characteristics

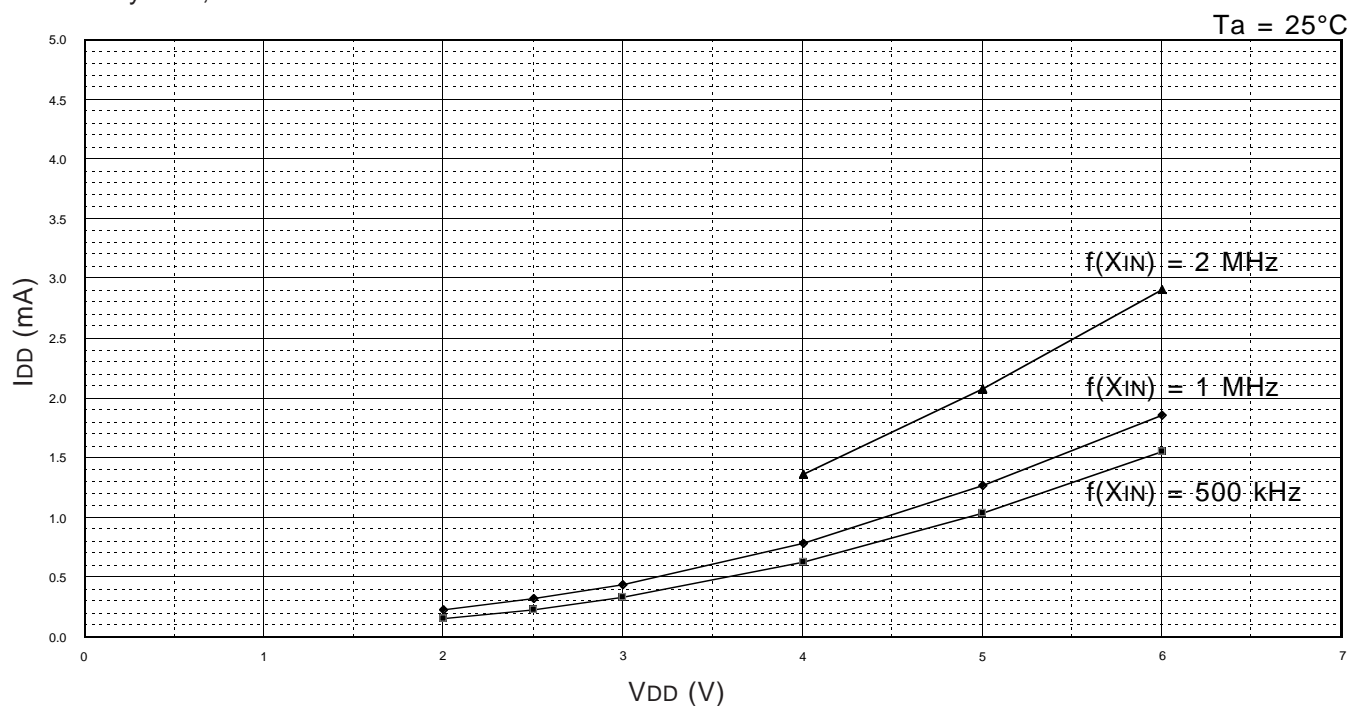
## 3.2 Typical characteristics

### 3.2.1 VDD–IDD characteristics

#### (1) CPU high-speed operating (system clock: $f(X_{IN})$ )

[measurement condition]

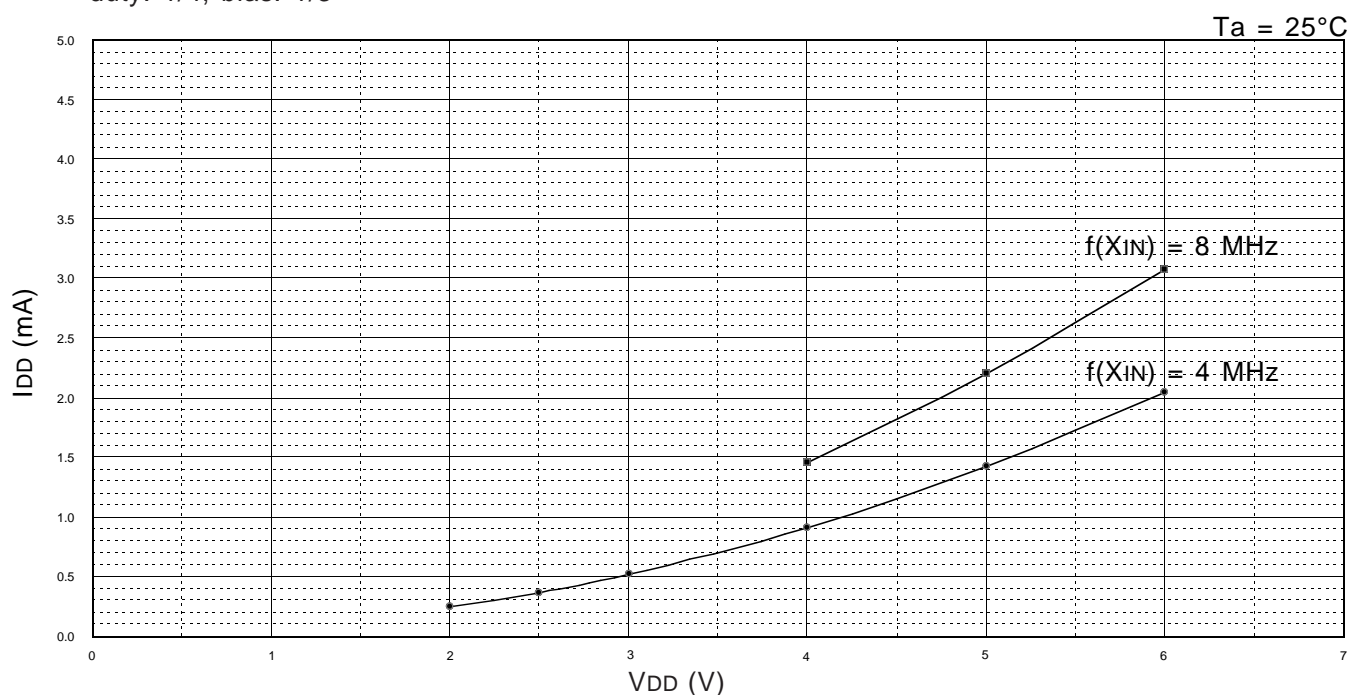
$X_{IN}$ : operating,  $X_{CIN}$ : 32 kHz, timer 2 count source:  $X_{CIN}$ , system clock:  $f(X_{IN})$ , LCD clock: 1 kHz, duty: 1/4, bias: 1/3



#### (2) CPU high-speed operating (system clock: $f(X_{IN})/4$ )

[measurement condition]

$X_{IN}$ : operating,  $X_{CIN}$ : 32 kHz, timer 2 count source:  $X_{CIN}$ , system clock:  $f(X_{IN})/4$ , LCD clock: 1 kHz, duty: 1/4, bias: 1/3

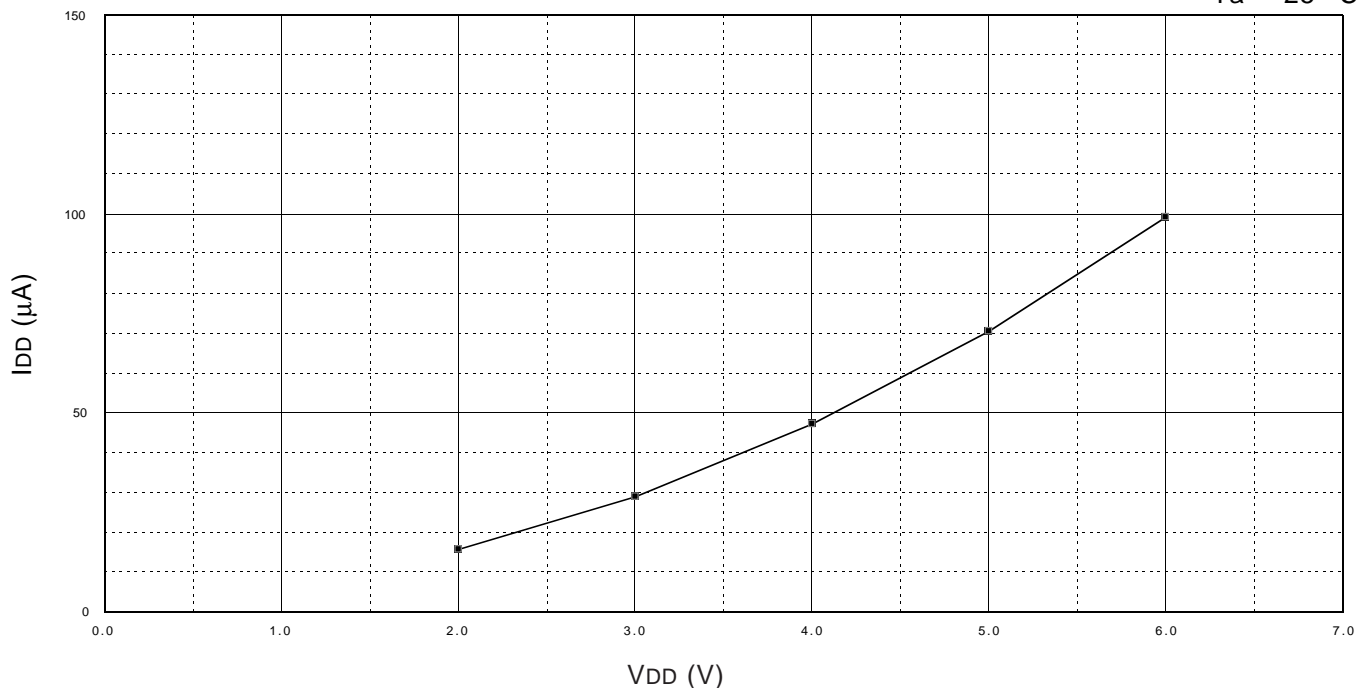


### (3) CPU low-speed operating (system clock: $f(X_{CIN})$ )

[measurement condition]

XIN: stop, XCIN: 32 kHz, timer 2 count source: XCIN, system clock:  $f(X_{CIN})$ , LCD clock: 1 kHz, duty: 1/4, bias: 1/3

Ta = 25 °C

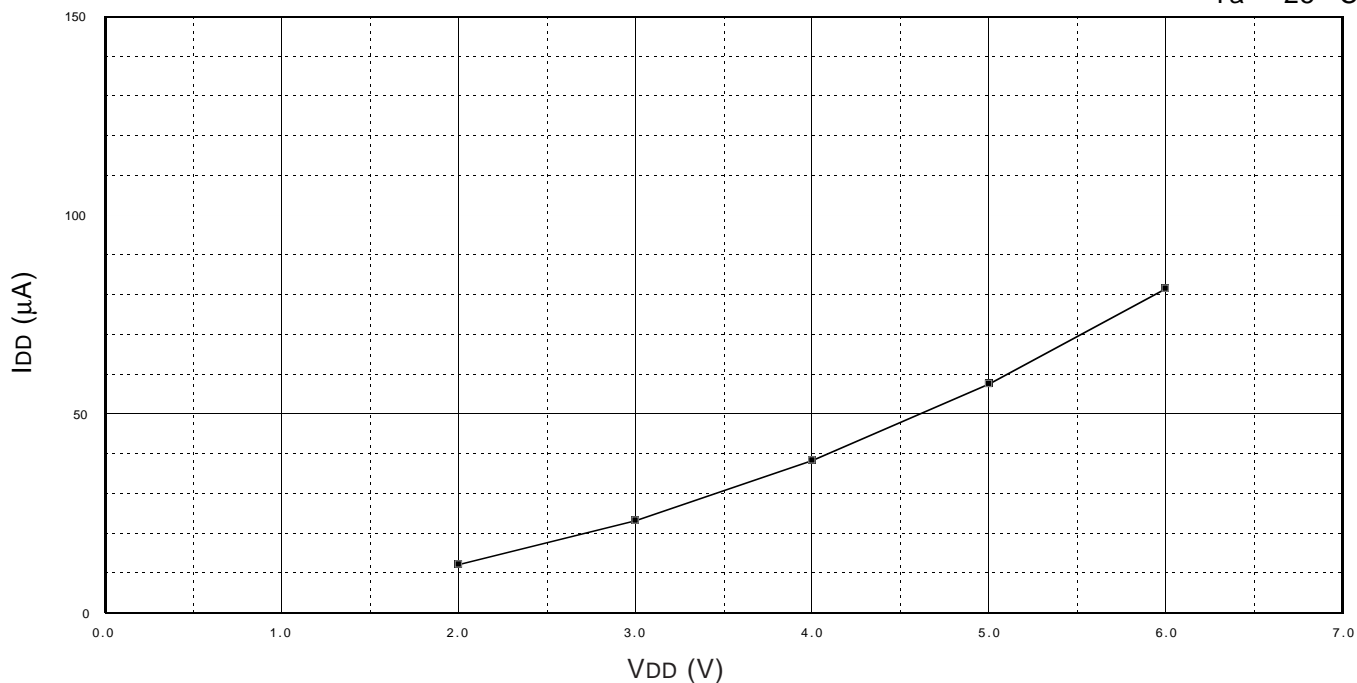


### (4) CPU low-speed operating (system clock: $f(X_{CIN})/4$ )

[measurement condition]

XIN: stop, XCIN: 32 kHz, timer 2 count source: XCIN, system clock:  $f(X_{CIN})/4$ , LCD clock: 1 kHz, duty: 1/4, bias: 1/3

Ta = 25 °C





# APPENDIX

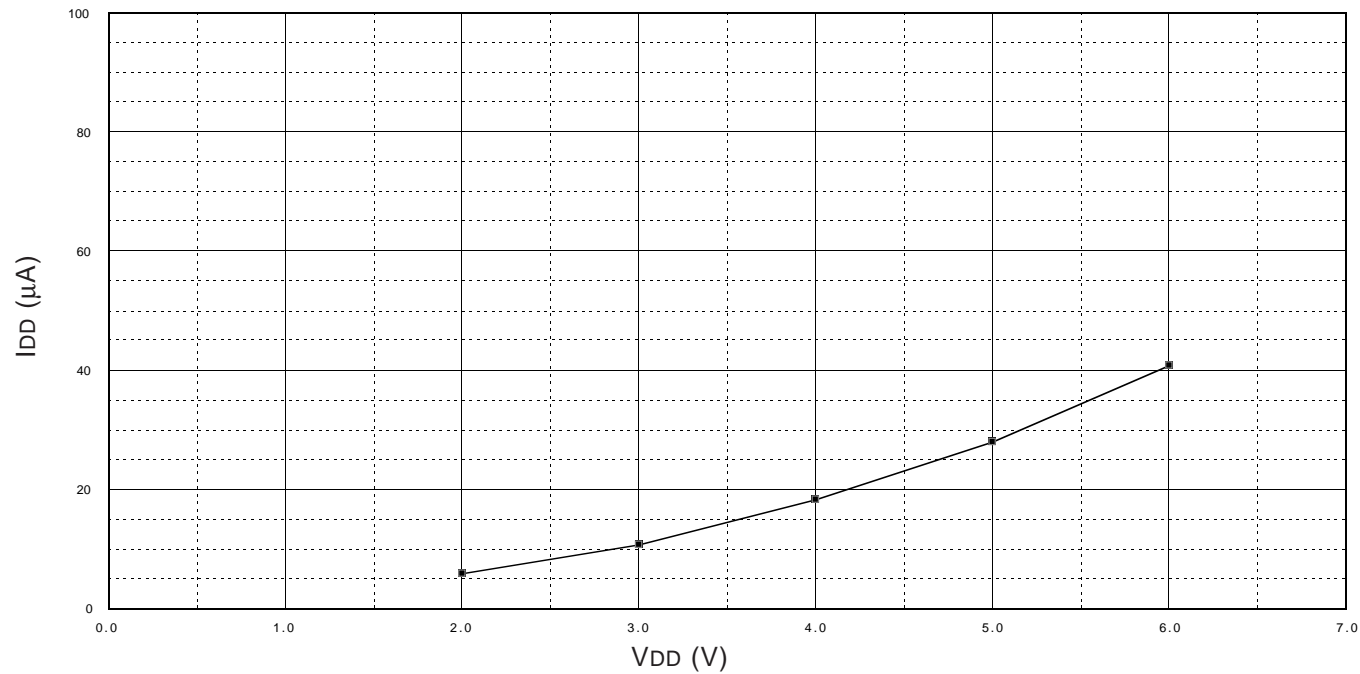
## 3.2 Typical characteristics

### (5) Clock operating

[measurement condition]

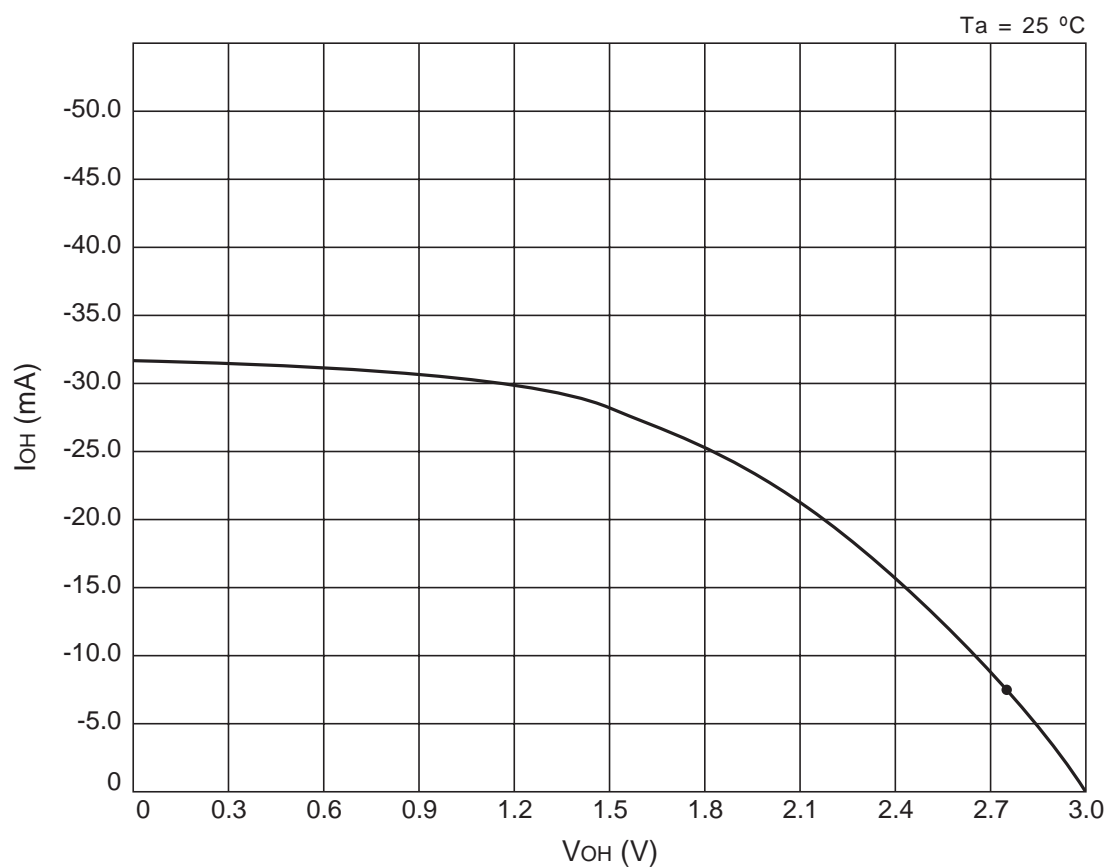
XIN: stop, XCIN: 32 kHz, timer 2 count source: XCIN, system clock: stop, LCD clock: 1 kHz,  
duty: 1/4, bias: 1/3

Ta = 25 °C

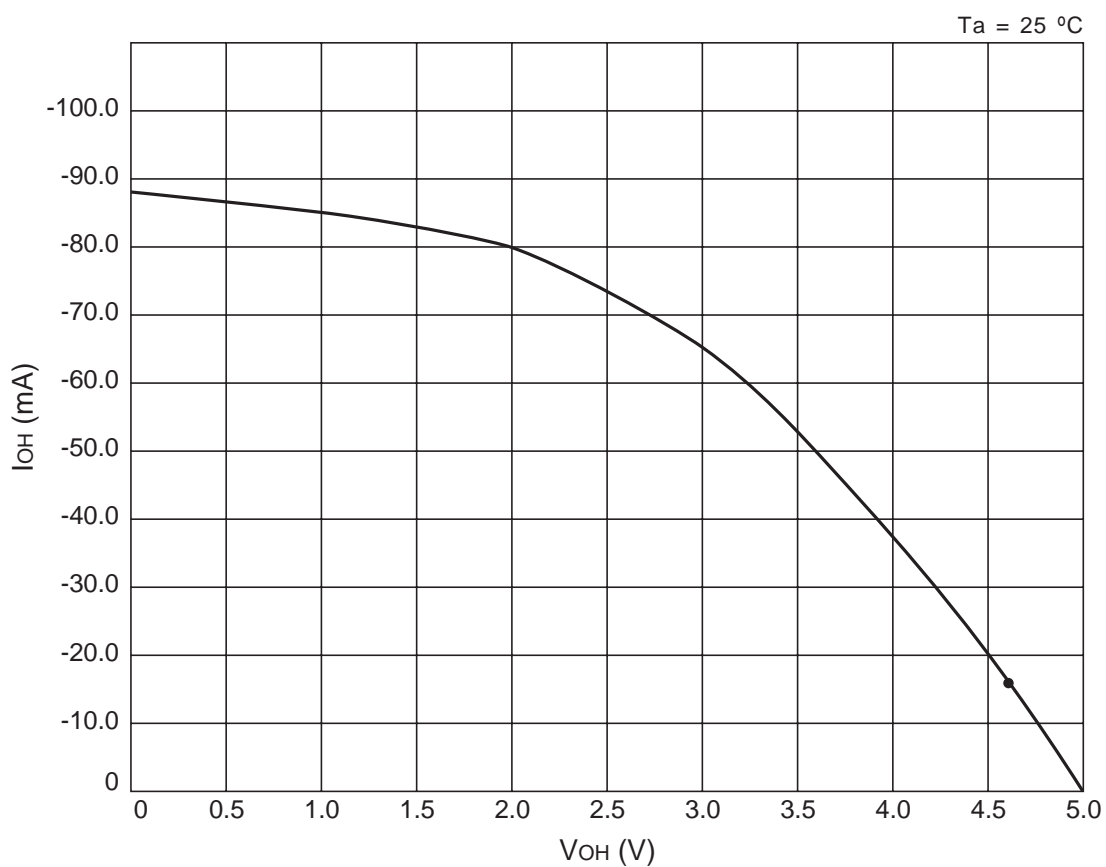


### 3.2.2 $V_{OH}$ – $I_{OH}$ characteristics (port CARR)

(1)  $V_{DD} = 3.0\text{ V}$



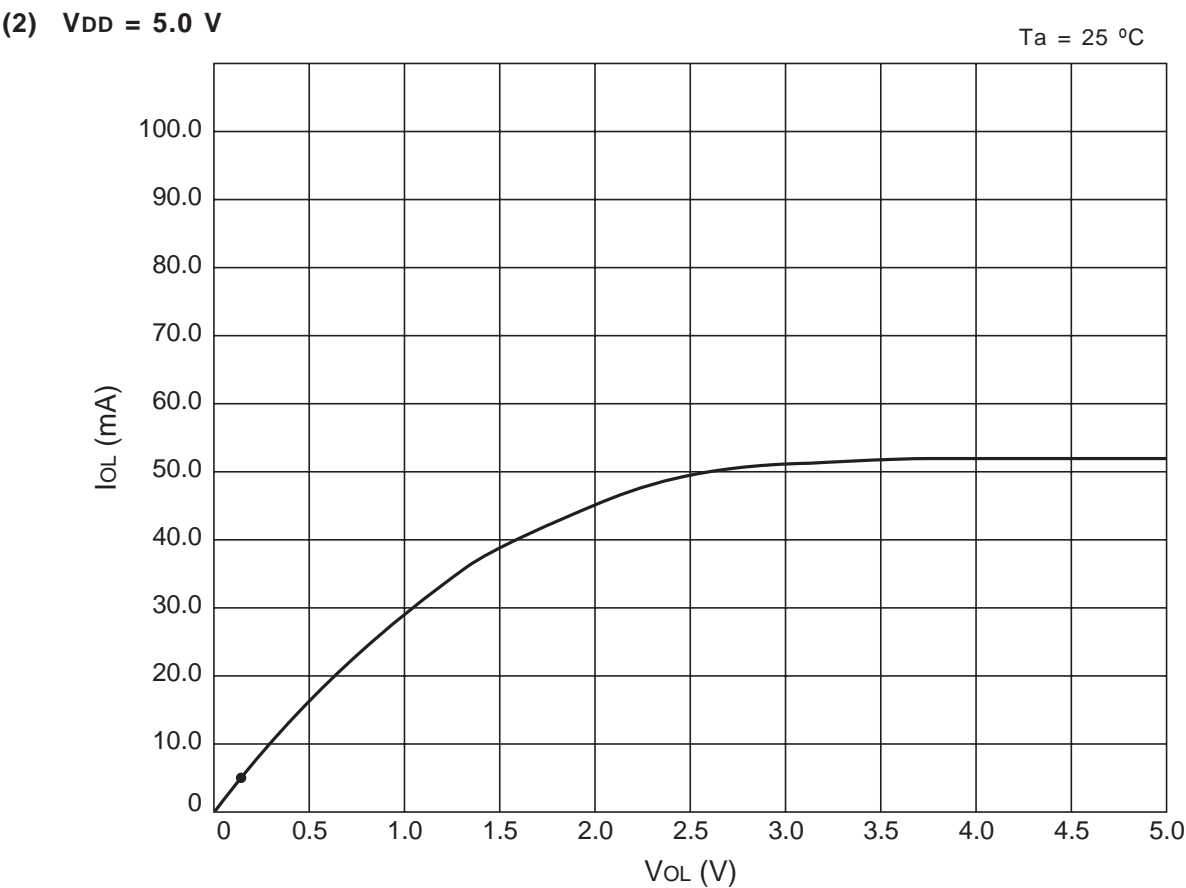
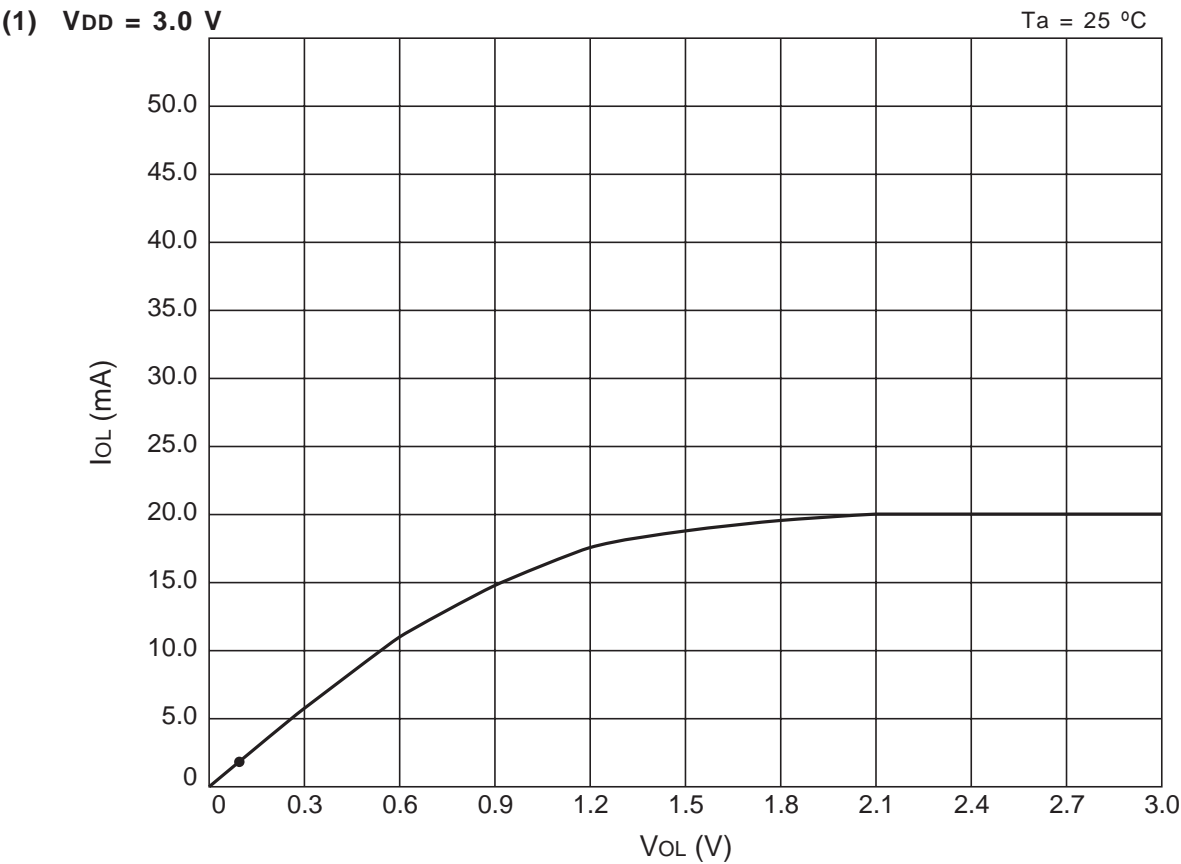
(2)  $V_{DD} = 5.0\text{ V}$



# APPENDIX

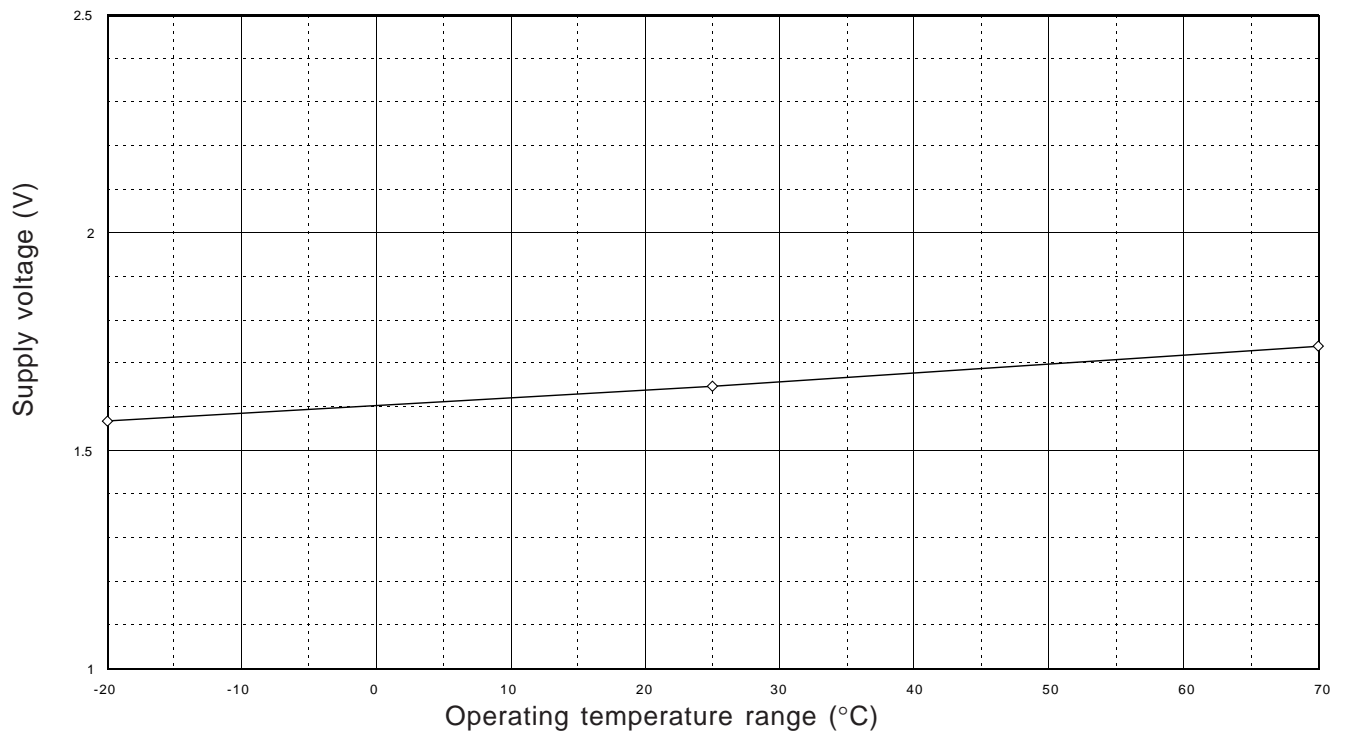
## 3.2 Typical characteristics

### 3.2.3 VOL–IOL characteristics (Ports P0, P1, D0–D7, CARR, $\overline{\text{RESET}}$ )

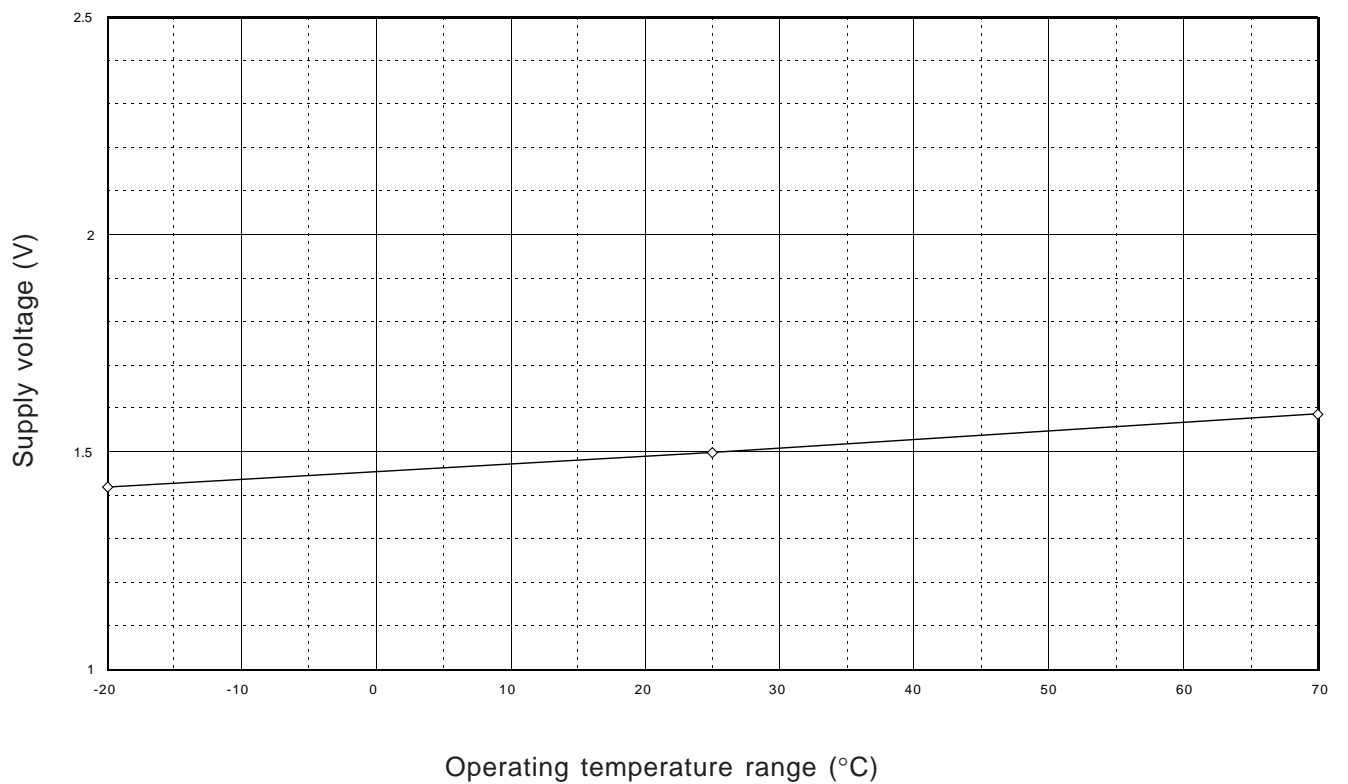


### 3.2.4 Voltage drop detection circuit temperature characteristics

#### (1) M34551Mx-XXXFP



#### (2) M34551E8-XXXFP



# APPENDIX

## 3.3 List of precautions

### 3.3 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  $\mu$ F) between pins  $V_{DD}$  and  $V_{SS}$  at the shortest distance,
- equalize its wiring in width and length, and
- use the thickest wire.

In the built-in PROM version,  $CNV_{SS}$  pin is also used as  $V_{PP}$  pin. Accordingly, when using this pin, connect this pin to  $V_{SS}$  through a resistor about 5 k $\Omega$  (connect this resistor to  $CNV_{SS}/V_{PP}$  pin as close as possible).

#### ② Prescaler

Stop the prescaler operation to change its frequency dividing ratio.

#### ③ Count source

Stop timer 1 or timer LC counting to change its count source. When timer 2 count source changes from  $f(X_{CIN})$  to ORCLK ( $W23 = "0" \rightarrow W23 = "1"$ ), the count value of timer 2 is initialized. However, when timer 2 count source changes from ORCLK to  $f(X_{CIN})$  ( $W23 = "1" \rightarrow W23 = "0"$ ) or the same count source is set again ( $W23 = "0" \rightarrow W23 = "0"$  or  $W23 = "1" \rightarrow W23 = "1"$ ), the count value of timer 2 is not initialized.

#### ④ Timer 2

Timer 2 has the watchdog timer function (WDT). When timer 2 is used as the WDT, note that the processing to initialize the count value and the execution of the WRST instruction.

#### ⑤ Reading the count value

Stop the prescaler and then execute the TAB1 instruction to read timer 1 data.

#### ⑥ Writing to reload register R1

Write the data to reload register R1 while timer 1 is operating, avoid a timing when timer 1 underflows.

#### ⑦ Notes when using the carrier wave output auto-control function

- Execute the STCR instruction after setting the timer 1 and register C2 in order to start the carrier generating circuit operation.
- Stop the timer 1 ( $W20 = "0"$ ) after stopping the carrier generating circuit (SPCR instruction executed) while the carrier wave output is disabled in order to stop the carrier wave output auto-control operation.
- If the carrier wave output auto-control function is invalidated ( $C20 = "0"$ ) while the carrier wave output is auto-controlled, the output of port CARR retains the state when the auto-control is invalidated regardless of timer 1 underflow. This state is released by timer 1 stop ( $W20 = "0"$ ).  
When the carrier wave output auto-control function is validated ( $C20 = "1"$ ) again after it is invalidated ( $C20 = "0"$ ), the auto-control of carrier wave output is started again when the next timer 1 underflow occurs. However, when the carrier wave output auto-control bit is changed during timer 1 underflow, the error-operation may occur.

- Use the carrier wave or the carrier wave divided by 2 as the timer 1 count source when the carrier wave output auto-control function is selected.  
If the ORCLK is used as the count source, a hazard may occur in port CARR output because ORCLK is not synchronized with the carrier wave.
- When "no carrier wave" is selected with register C1 ( $(C13C12C11C10) = (0101), (1101)$ ), the enable/disable of the carrier wave output cannot be controlled by the carrier wave output auto-control function.

#### ⑧ D5/INT pin

When the interrupt valid waveform of D5/INT pin is changed with the bit 2 of register I1 in software, be careful about the following notes.

- Clear the bit 0 of register V1 to "0" and then change the interrupt valid waveform of D5/INT pin with the bit 2 of register I1 (refer to Figure 40①).
- Clear the bit 2 of register I1 to "0" and execute the SNZ0 instruction to clear the EXF0 flag after executing at least one instruction (refer to Figure 40②). Depending on the input state of the D5/INT pin, the external 0 interrupt request flag (EXF0) may be set to "1" when the interrupt valid waveform is changed.

```
:  
LA      4      ; (XXX02)  
TV1A    ; The SNZ0 instruction is valid ①  
LA      4  
TI1A    ; Change of the interrupt valid waveform  
NOP                                           ②  
SNZ0    ; The SNZ0 instruction is executed  
NOP  
:  
X : this bit is not related to the setting of INT.
```

Fig. 40 External 0 interrupt program example

#### ⑨ One Time PROM version

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

#### ⑩ Multifunction

Note that the port D5 output function can be used even when INT function is selected.

#### ⑪ Power down instruction (POF instruction, POF2 instruction)

Execute the POF or POF2 instruction immediately after executing the EPOF instruction to enter the power down state. Note that system cannot enter the power down state when executing only the POF or POF2 instruction. Be sure to disable interrupts by executing the DI instruction before executing the EPOF instruction.

#### ⑫ Program counter

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

### 3.4 Notes on noise

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

#### 3.4.1 Shortest wiring length

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

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

#### (1) Package

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

##### ● Reason

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

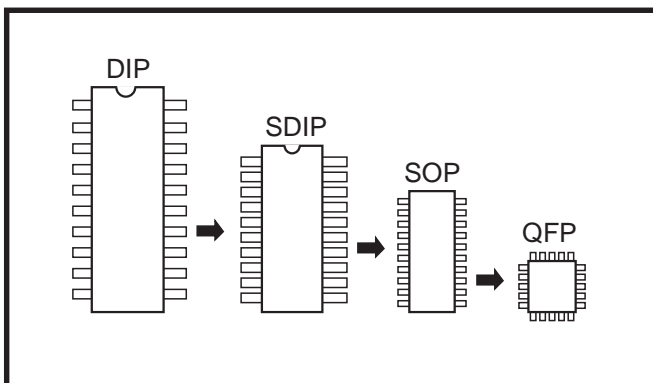


Fig. 3.4.1 Selection of packages

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

Make the length of wiring which is connected to the  $\overline{\text{RESET}}$  input pin as short as possible. Especially, connect a capacitor across the  $\overline{\text{RESET}}$  input pin and the  $V_{\text{ss}}$  pin with the shortest possible wiring (within 20mm).

##### ● Reason

The width of a pulse input into the  $\overline{\text{RESET}}$  pin is determined by the timing necessary conditions. If noise having a shorter pulse width than the standard is input to the  $\overline{\text{RESET}}$  input pin, the reset is released before the internal state of the microcomputer is completely initialized. This may cause a program runaway.

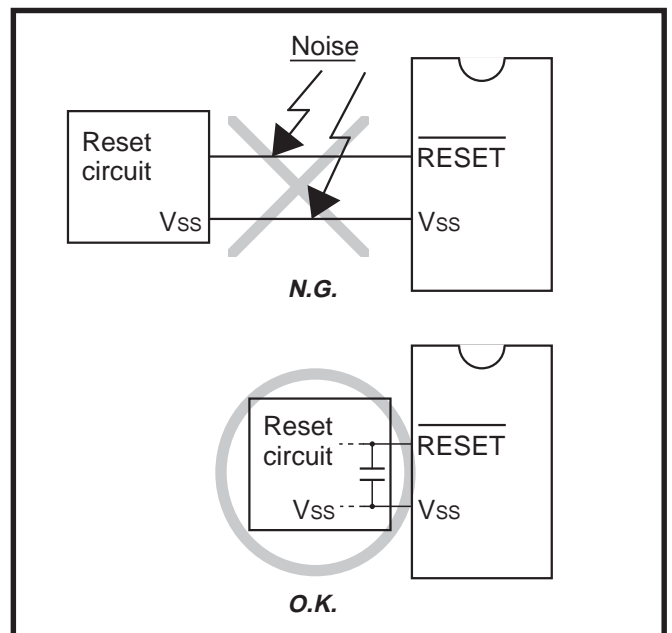


Fig. 3.4.2 Wiring for the  $\overline{\text{RESET}}$  input pin

# APPENDIX

## 3.4 Notes on noise

### (3) Wiring for clock input/output pins

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

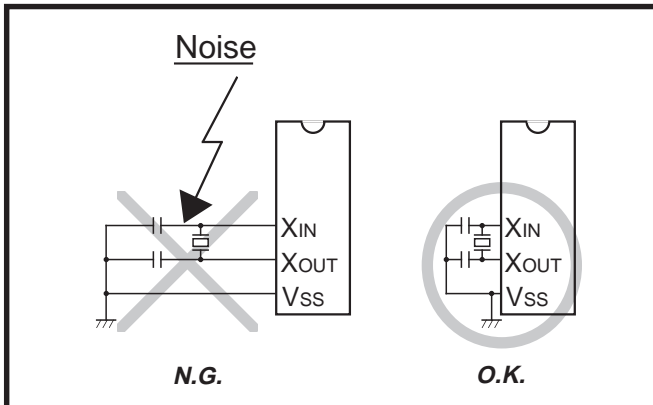


Fig. 3.4.3 Wiring for clock I/O pins

#### ● Reason

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

### (4) Wiring to CNVss pin

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

#### ● Reason

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

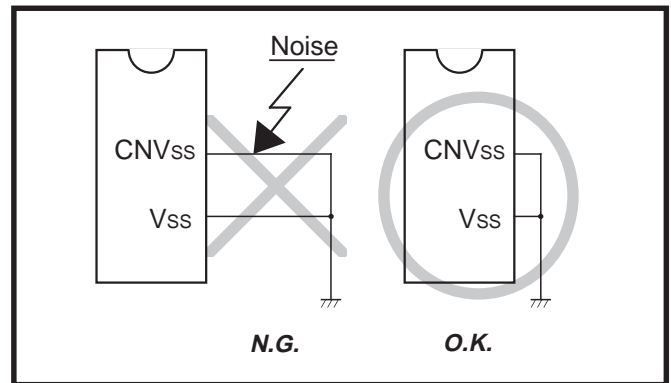


Fig. 3.4.4 Wiring for CNVss pin

### (5) Wiring to VPP pin of One Time PROM version

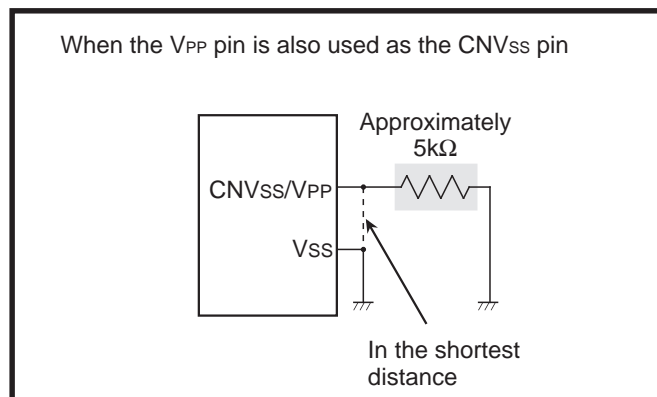
#### ● When the VPP pin is also used as the CNVss pin

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

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

#### ● Reason

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

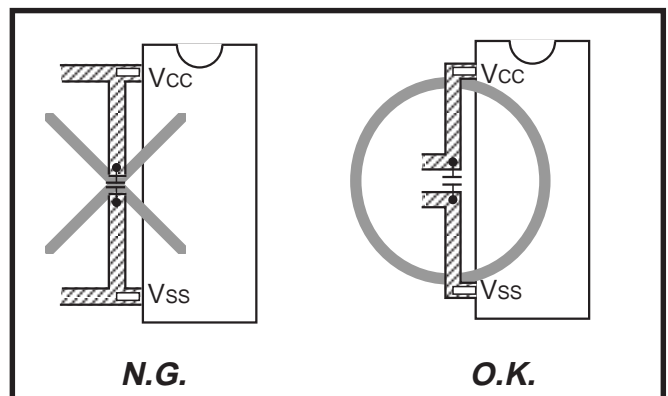


**Fig. 3.4.5 Wiring for the VPP pin of the One Time PROM version**

### 3.4.2 Connection of bypass capacitor across Vss line and Vcc line

Connect an approximately 0.1 μF bypass capacitor across the Vss line and the Vcc line as follows:

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



**Fig. 3.4.6 Bypass capacitor across the Vss line and the Vcc line**



# APPENDIX

## 3.4 Notes on noise

### 3.4.3 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.

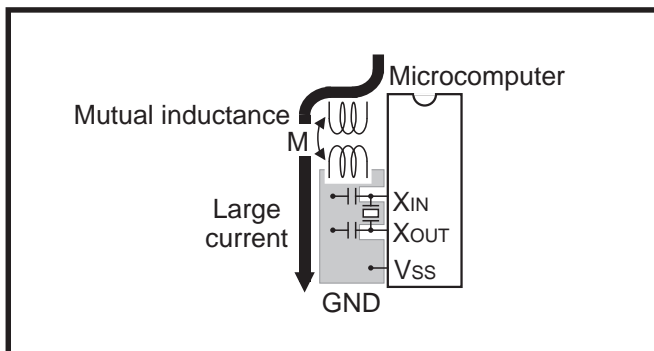


Fig. 3.4.7 Wiring for a large current signal line

#### (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 CARR pin signal line) may affect other lines at signal rising edge or falling edge. If such lines cross over a clock line, clock waveforms may be deformed, which causes a microcomputer failure or a program runaway.

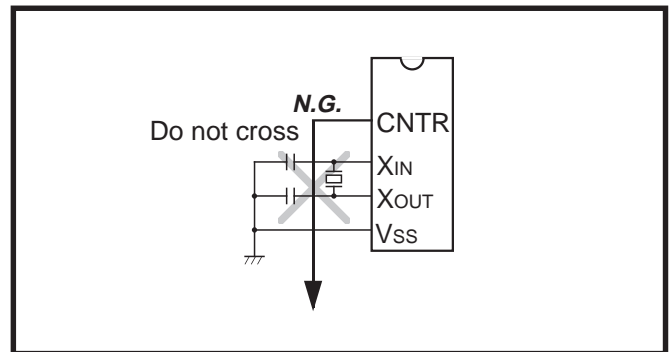


Fig. 3.4.8 Wiring to a signal line where potential levels change frequently

#### (3) Oscillator protection using Vss pattern

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

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

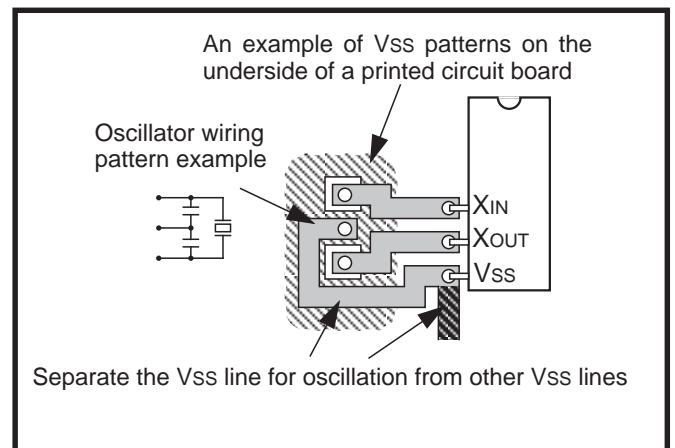


Fig. 3.4.9 Vss pattern on the underside of an oscillator

### 3.4.4 Setup for I/O ports

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

<Hardware>

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

<Software>

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

### 3.4.5 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 byte 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:

$$N+1 \geq (\text{Counts of interrupt processing executed in each main routine})$$

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

- Watches the operation of the interrupt processing routine by comparing the SWDT contents with counts of interrupt processing after the initial value N has been set.
- Detects that the interrupt processing routine has failed and determines to branch to the program

initialization routine for recovery processing in the following case:

If the SWDT contents do not change after interrupt processing.

<The interrupt processing routine>

- Decrements the SWDT contents by 1 at each interrupt processing.
- Determines that the main routine operates normally when the SWDT contents are reset to the initial value N at almost fixed cycles (at the fixed interrupt processing count).
- Detects that the main routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:

If the SWDT contents are not initialized to the initial value N but continued to decrement and if they reach 0 or less.

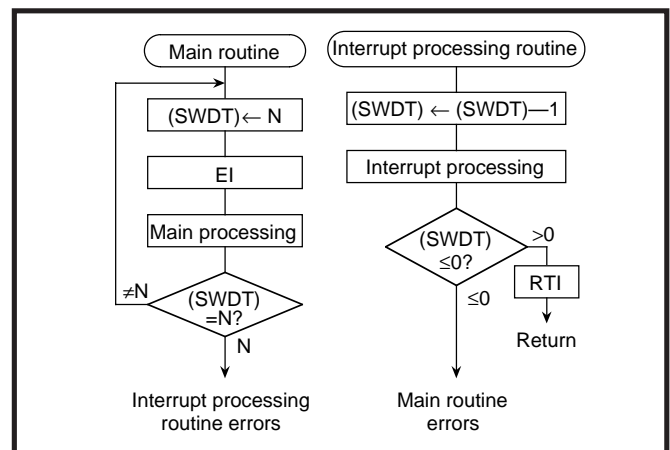


Fig. 3.4.10 Watchdog timer by software

# APPENDIX

## 3.5 Mask ROM order confirmation form

### 3.5 Mask ROM order confirmation form

GZZ-SH10-66B <63A0>

**4500 SERIES MASK ROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP MICROCOMPUTER M34551M4-XXXFP**  
**mitsubishi electric**

Please fill in all items marked \*.

* Customer	Company name	TEL ( )	Receipt	Date:	Section head signature	Supervisor signature
	Date issued	Date:				
			Issuance signature	Responsible officer	Supervisor	

\*1. Confirmation

Specify the type of EPROMs submitted.

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

If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM area  (hexadecimal notation)

EPROM Type:

<input type="checkbox"/> 27C256	<input type="checkbox"/> 27C512																														
<table><tr><td>Low-order 5-bit data</td><td>0000<sub>16</sub></td><td>4.00K</td></tr><tr><td></td><td>0FFF<sub>16</sub></td><td></td></tr><tr><td>High-order 5-bit data</td><td>4000<sub>16</sub></td><td>4.00K</td></tr><tr><td></td><td>4FFF<sub>16</sub></td><td></td></tr><tr><td></td><td>7FFF<sub>16</sub></td><td></td></tr></table>	Low-order 5-bit data	0000 <sub>16</sub>	4.00K		0FFF <sub>16</sub>		High-order 5-bit data	4000 <sub>16</sub>	4.00K		4FFF <sub>16</sub>			7FFF <sub>16</sub>		<table><tr><td>Low-order 5-bit data</td><td>0000<sub>16</sub></td><td>4.00K</td></tr><tr><td></td><td>0FFF<sub>16</sub></td><td></td></tr><tr><td>High-order 5-bit data</td><td>4000<sub>16</sub></td><td>4.00K</td></tr><tr><td></td><td>4FFF<sub>16</sub></td><td></td></tr><tr><td></td><td>FFFF<sub>16</sub></td><td></td></tr></table>	Low-order 5-bit data	0000 <sub>16</sub>	4.00K		0FFF <sub>16</sub>		High-order 5-bit data	4000 <sub>16</sub>	4.00K		4FFF <sub>16</sub>			FFFF <sub>16</sub>	
Low-order 5-bit data	0000 <sub>16</sub>	4.00K																													
	0FFF <sub>16</sub>																														
High-order 5-bit data	4000 <sub>16</sub>	4.00K																													
	4FFF <sub>16</sub>																														
	7FFF <sub>16</sub>																														
Low-order 5-bit data	0000 <sub>16</sub>	4.00K																													
	0FFF <sub>16</sub>																														
High-order 5-bit data	4000 <sub>16</sub>	4.00K																													
	4FFF <sub>16</sub>																														
	FFFF <sub>16</sub>																														

Set "FF<sub>16</sub>" in the shaded area.

Set "1112" in the area  of low-order and high-order 5-bit data.

\* 2. Mark Specification

Mark specification must be submitted using the correct form for the type of package being ordered. Fill out the approximate Mark Specification Form (48P6S-A for M34551M4-XXXFP) and attach to the Mask ROM Order Confirmation Form.

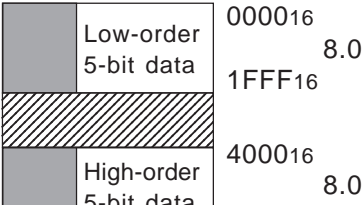
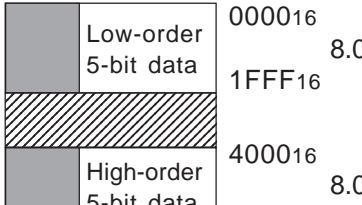
\* 3. Comments

**4500 SERIES MASK ROM ORDER CONFIRMATION FORM**  
**SINGLE-CHIP MICROCOMPUTER M34551M8-XXXFP**  
**MITSUBISHI ELECTRIC**

* Customer	Company name	TEL ( )	F		
	Date issued				
	Issuance signature				

If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

--	--	--	--

27C256	27C512
 <p>Diagram illustrating the memory map for the 27C256 device. The memory is divided into four 8.00K blocks. The first block (0000<sub>16</sub> to 1FFF<sub>16</sub>) is labeled 'Low-order 5-bit data'. The second block (4000<sub>16</sub> to 5FFF<sub>16</sub>) is labeled 'High-order 5-bit data'. The third and fourth blocks (6000<sub>16</sub> to 7FFF<sub>16</sub>) are shaded with diagonal lines.</p>	 <p>Diagram illustrating the memory map for the 27C512 device. The memory is divided into four 8.00K blocks. The first block (0000<sub>16</sub> to 1FFF<sub>16</sub>) is labeled 'Low-order 5-bit data'. The second block (4000<sub>16</sub> to 5FFF<sub>16</sub>) is labeled 'High-order 5-bit data'. The third and fourth blocks (6000<sub>16</sub> to 7FFF<sub>16</sub>) are shaded with diagonal lines.</p>

Set "1112" in the area  of low-order and high-order 5-bit data.

### \* 3. Comments

# APPENDIX

## 3.6 ROM programming order confirmation form

### 3.6 ROM programming order confirmation form

GZZ-SH10-69B <64A0>

**4500 SERIES ROM PROGRAMMING ORDER CONFIRMATION FORM**  
**SINGLE-CHIP MICROCOMPUTER M34551E8-XXXFP**  
**mitsubishi electric**

Please fill in all items marked \*.

* Customer	Company name	TEL ( )	Receipt	Date:	
	Date issued	Date:		Section head signature	Supervisor signature
			Issuance signature	Responsible officer	Supervisor

\* 1. Confirmation

Specify the type of EPROMs submitted.

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

If at least two of the three sets of EPROMs submitted contain the identical data, we will produce programming based on this data. We shall assume the responsibility for errors only if the ROM data on the products we produce differ from this data. Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM area  (hexadecimal notation)

EPROM Type:

<input type="checkbox"/> 27C256	<input type="checkbox"/> 27C512																										
<table><tr><td>Low-order 5-bit data</td><td>0000<sub>16</sub></td><td rowspan="2">8.00K</td></tr><tr><td></td><td>1FFF<sub>16</sub></td></tr><tr><td>High-order 5-bit data</td><td>4000<sub>16</sub></td><td rowspan="2">8.00K</td></tr><tr><td></td><td>5FFF<sub>16</sub></td></tr><tr><td></td><td>7FFF<sub>16</sub></td><td></td></tr></table>	Low-order 5-bit data	0000 <sub>16</sub>	8.00K		1FFF <sub>16</sub>	High-order 5-bit data	4000 <sub>16</sub>	8.00K		5FFF <sub>16</sub>		7FFF <sub>16</sub>		<table><tr><td>Low-order 5-bit data</td><td>0000<sub>16</sub></td><td rowspan="2">8.00K</td></tr><tr><td></td><td>1FFF<sub>16</sub></td></tr><tr><td>High-order 5-bit data</td><td>4000<sub>16</sub></td><td rowspan="2">8.00K</td></tr><tr><td></td><td>5FFF<sub>16</sub></td></tr><tr><td></td><td>FFFF<sub>16</sub></td><td></td></tr></table>	Low-order 5-bit data	0000 <sub>16</sub>	8.00K		1FFF <sub>16</sub>	High-order 5-bit data	4000 <sub>16</sub>	8.00K		5FFF <sub>16</sub>		FFFF <sub>16</sub>	
Low-order 5-bit data	0000 <sub>16</sub>	8.00K																									
	1FFF <sub>16</sub>																										
High-order 5-bit data	4000 <sub>16</sub>	8.00K																									
	5FFF <sub>16</sub>																										
	7FFF <sub>16</sub>																										
Low-order 5-bit data	0000 <sub>16</sub>	8.00K																									
	1FFF <sub>16</sub>																										
High-order 5-bit data	4000 <sub>16</sub>	8.00K																									
	5FFF <sub>16</sub>																										
	FFFF <sub>16</sub>																										

Set "FF<sub>16</sub>" in the shaded area.

Set "1112" in the area ☐ of low-order and high-order 5-bit data.

\* 2. Mark Specification

Mark specification must be submitted using the correct form for the type of package being ordered. Fill out the approximate Mark Specification Form (48P6S-A for M34551E8-XXXFP) and attach to the ROM Programming Order Confirmation Form.

\* 3. Comments

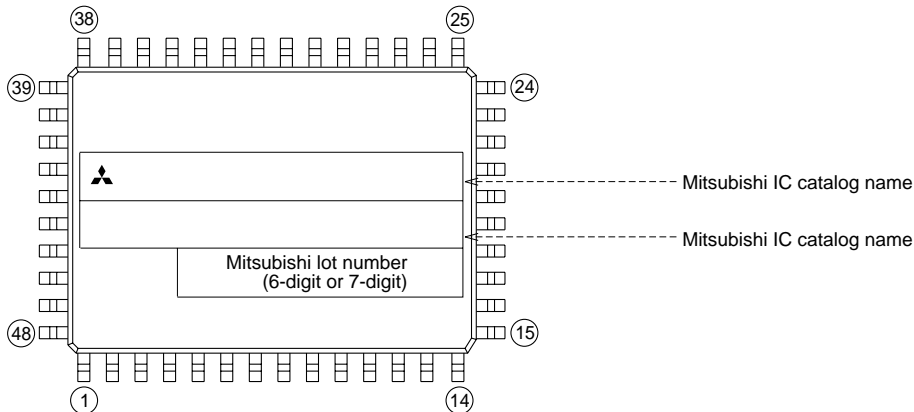
### 3.7 Mark specification form

#### 48P6S-A (48-PIN QFP) MARK SPECIFICATION FORM

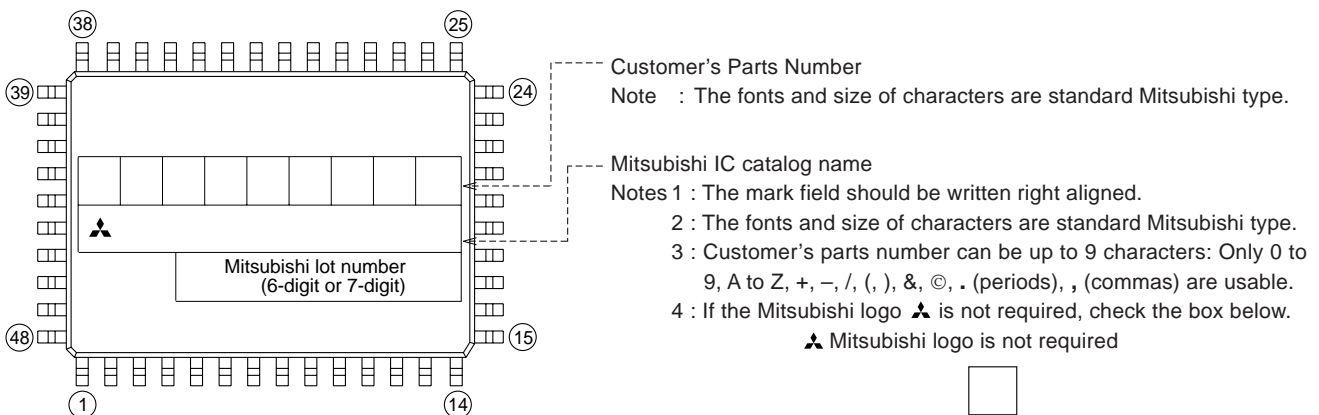
Mitsubishi IC catalog name

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

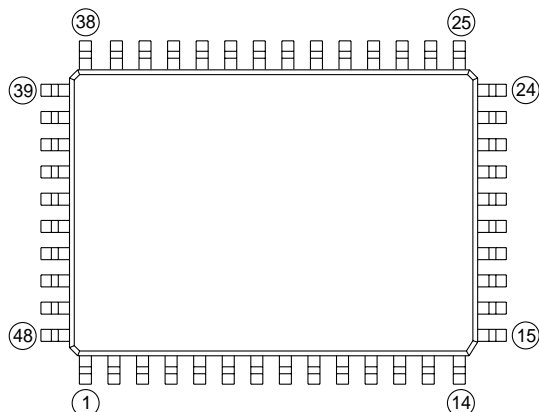
##### A. Standard Mitsubishi Mark



##### B. Customer's Parts Number + Mitsubishi IC Catalog Name



##### C. Special Mark Required



Notes1 : If Special Mark is to be Printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated technically as close as possible.

Mitsubishi lot number (6-digit, or 7-digit) and Mask ROM number (3-digit) are always marked.

2 : If the customer's trade mark logo must be used in the Special Mark, check the box below.

Please submit a clean original of the logo.

For the new special character fonts, a clean font original (ideally logo drawing) must be submitted.

Special logo required

☐

# APPENDIX

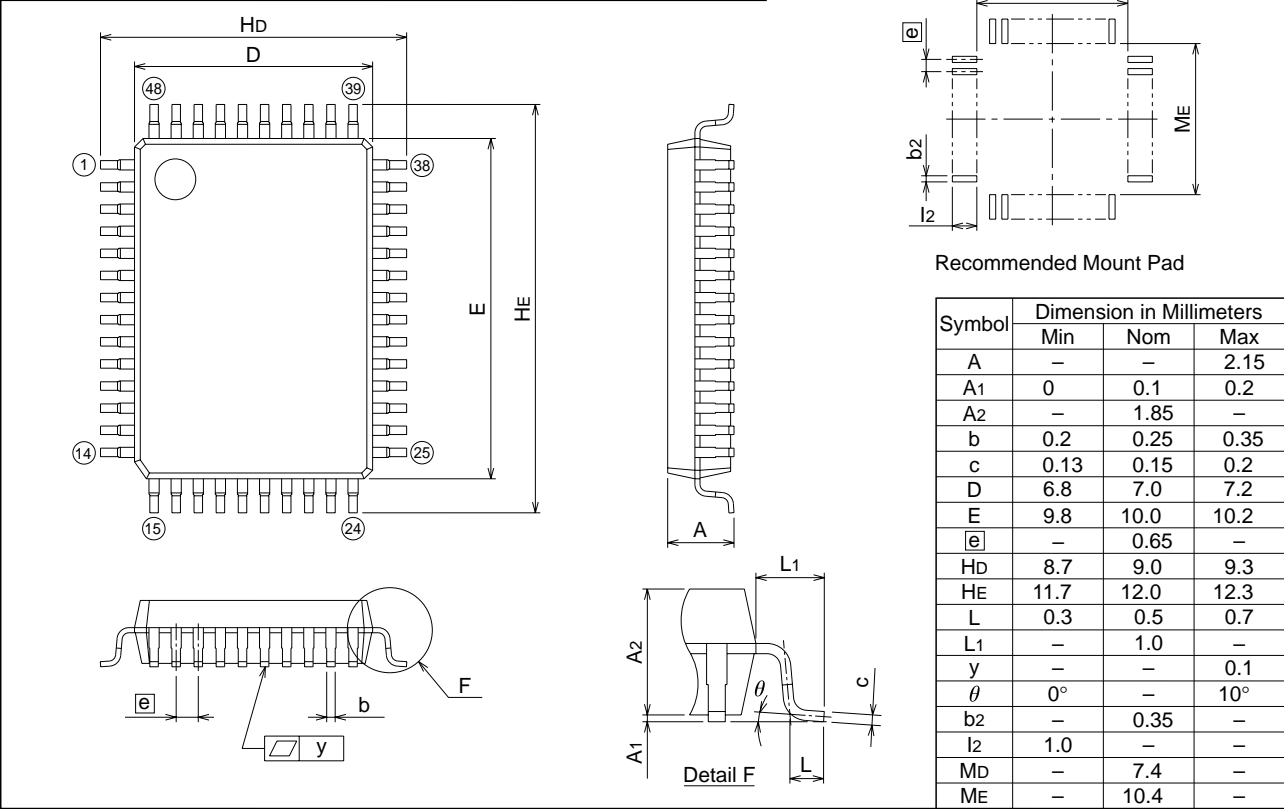
## 3.8 Package outline

### 3.8 Package outline

#### 48P6S-A

Plastic 48pin 7X10mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP48-P-710-0.65	—	0.29	Alloy 42



**MITSUBISHI SEMICONDUCTORS  
USER'S MANUAL  
4551 Group**

---

Sep. First Edition 1998

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

Published by  
Mitsubishi Electric Corp., Semiconductor Marketing Division

---

This book, or parts thereof, may not be reproduced in any form without permission of Mitsubishi Electric Corporation.

**©1998 MITSUBISHI ELECTRIC CORPORATION**



# User's Manual

## 4551 Group



**mitsubishi electric corporation**

HEAD OFFICE: MITSUBISHI DENKI BLDG., MARUNOUCHI, TOKYO 100. TELEX: J24532 CABLE: MELCO TOKYO

