NEC

MOS INTEGRATED CIRCUIT

μPD17010

4-BIT SINGLE-CHIP MICROCONTROLLER WITH HARDWARE DEDICATED TO DIGITAL TUNING SYSTEM

The μPD17010 is a 4-bit single-chip CMOS microcontroller containing hardware for digital tuning systems.

The CPU uses a 17K architecture and can directly manipulate the data memory and control various operations and peripheral hardware with a single instruction. All instructions are 16-bit 1-word instructions.

As the peripheral hardware, a prescaler for digital tuning that operates at up to 150 MHz, PLL frequency synthesizer, and frequency counter, as well as many I/O ports, LCD controller/driver, 12-bit timer, A/D converter, D/A converter (PWM output), clock generator port are provided.

Therefore, a high-performance digital tuning system with sophisticated functions can be configured with a single chip. The μPD17P010 is available as a one-time PROM model of the μPD17010. This one-time PROM model can be used for evaluation of the program of the μPD17010 and small-scale production of the application system.

FEATURES

- 17K architecture: General-purpose register system
- Program memory (ROM)
  16K bytes (7932 × 16 bits)
- General-purpose data memory (RAM)
  432 × 4 bits
- Instruction execution time
  4.44 μs (with 4.5-MHz crystal resonator)
- Decimal operation
- Table reference
- Hardware for PLL frequency synthesizer
  Dual modulus prescaler (150 MHz MAX.), programmable divider, phase comparator, charge pump
- A variety of peripheral hardware
  General-purpose I/O ports, LCD controller/driver, serial interface, 12-bit timer, A/D converter, D/A converter (PWM output), clock generator port, frequency counter
- Many interrupts
  External: 1
  Internal: 4
  External/internal (multiplexed): 1
- Power-ON reset, reset by CE pin, and power failure detection circuit
- Low power-dissipation CMOS
- Supply voltage: 5 V±10 %

Document No. U10340EJ2V0DS00 (2nd edition)
(Previous No. ID-2878)
Date Published October 1995 P
Printed in Japan

ORDERING INFORMATION

Part Number	Package
μPD17010GF-×××-3B9	80-pin plastic QFP (14 × 20 mm)
μPD17010GF-E××-3B9[Note]	80-pin plastic QFP (14 × 20 mm)

Note Model supporting I²C bus. To use the I²C bus (including when the function is implemented by program without using the peripheral hardware), consult NEC when ordering mask.

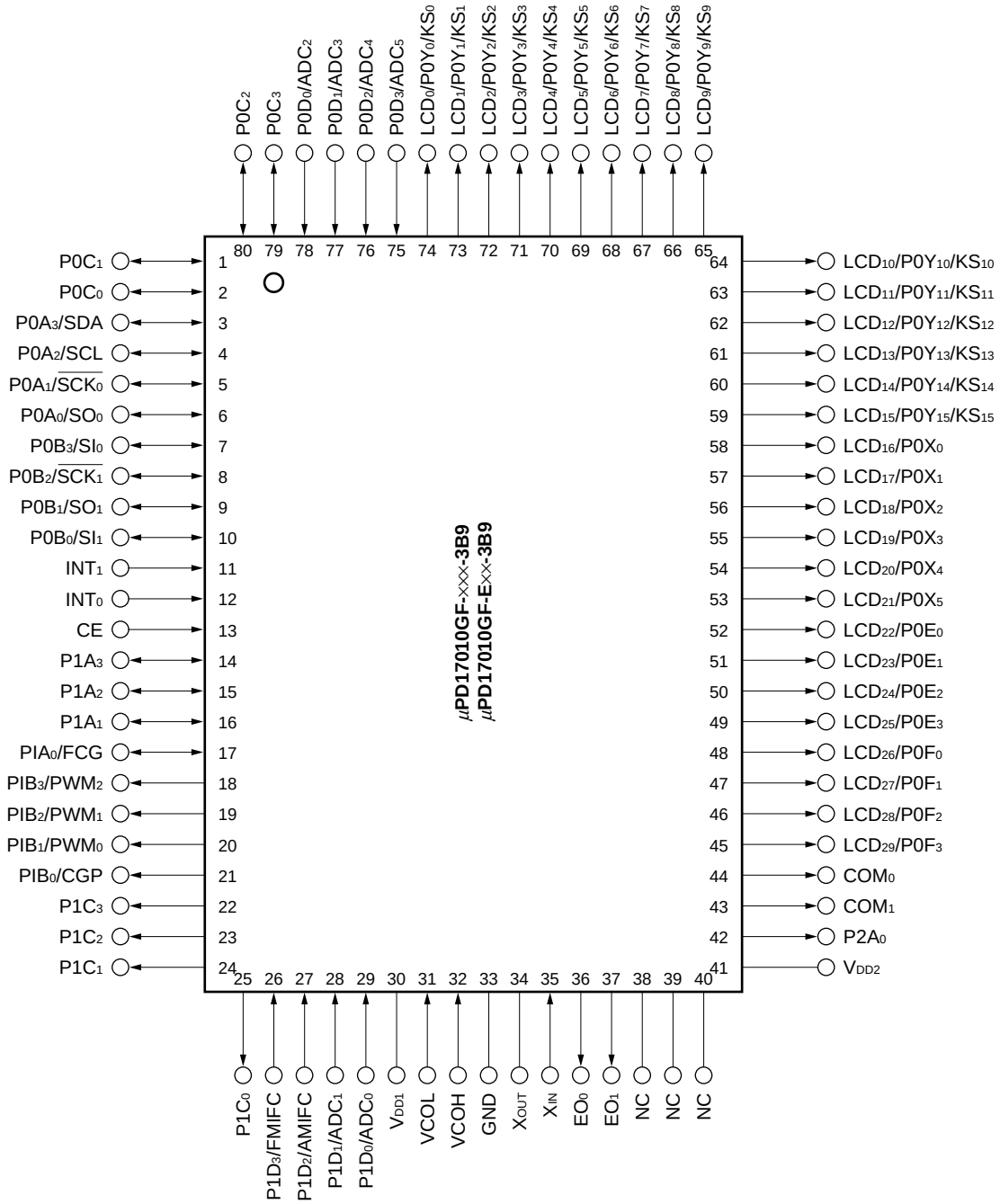
Remark ××× indicates a ROM code.

FUNCTIONAL OUTLINE

Item	Function
Program memory (ROM)	• 16K bytes (7932 × 16 bits) All internal ROM areas can be referenced through table
General-purpose data memory (RAM)	• 432 × 4 bits Data buffer : 4 × 4 bits, general register : 16 × 4 bits
System register	• 12 × 4 bits
Register file	• 41 × 4 bits (control register)
General-purpose port register (including LCD dot data register)	• 24 × 4 bits
Instruction execution time	• 4.44 μs (with 4.5-MHz crystal resonator)
Stack level	• 9 levels (stack can be manipulated)
General-purpose ports	• I/O ports : 16 • Input ports : 8 • Output ports : 9 (+30: LCD segment pin)
Clock generator port (CGP)	• 1 VDP (Variable Duty Pulse) and SG (Signal Generator) functions
LCD controller/driver	• 30 segments, 2 commons 1/2 duty, 1/2 bias, frame frequency: 250 MHz, drive voltage: V_{DD} Segment pins multiplexed with key source: 16 All 30 pins can be used as output port pins (4, 4, 6, and 16 pins can be independently set)
Serial interface	• 2 systems (3 channels) Serial interface 0 : 2-line (I²C bus, serial I/O) 3-line (serial I/O) Serial interface 1 : 3-line (serial I/O)
D/A converter	• 8 bits × 3 (PWM output, output voltage: 16 V MAX.)
A/D converter	• 6 bits × 6 (successive approximation via software)

Item		Function
Interrupt		• 6 (maskable interrupts) External : 1 (INT_0 pin) Internal : 4 (12-bit timer, basic timer 1, serial interface 0, frequency counter) External/internal (multiplexed) : 1 (INT_1 pin or overflow of timer/counter)
Timer		• 3 channels 12-bit timer (1, 50 μs) Basic timer 0 carry (1, 5, 100, 250 ms) Basic timer 1 interrupt (1, 5, 100, 250 ms)
Reset		• Power-ON reset (on power up) • Reset by CE pin (CE pin low level → high level) • Power failure detection function
PLL frequency synthesizer	Division method	• 2 types Direct division (VCOL pin: 30 MHz MAX.) Pulse swallow (VCOL pin: 40 MHz MAX.) (VCOH pin: 150 MHz MAX.)
	Reference frequency	• 12 types selectable by program 1, 1.25, 2.5, 3, 5, 6.25, 9, 10, 12.5, 25, 50, 100 kHz
	Charge pump	• Two independent error out outputs
	Phase comparator	• Unlock detection programmable Delay time of unlock F/F selectable
Frequency counter		• Frequency measurement $P1D_3$/FMIFC pin : 5 to 15 MHz $P1D_2$/AMIFC pin : 0.1 to 1 MHz • External gate width measurement $P1A_0$/FCG pin
Supply voltage		5 V ± 10 %
Package		80-pin plastic QFP (14 × 20 mm)

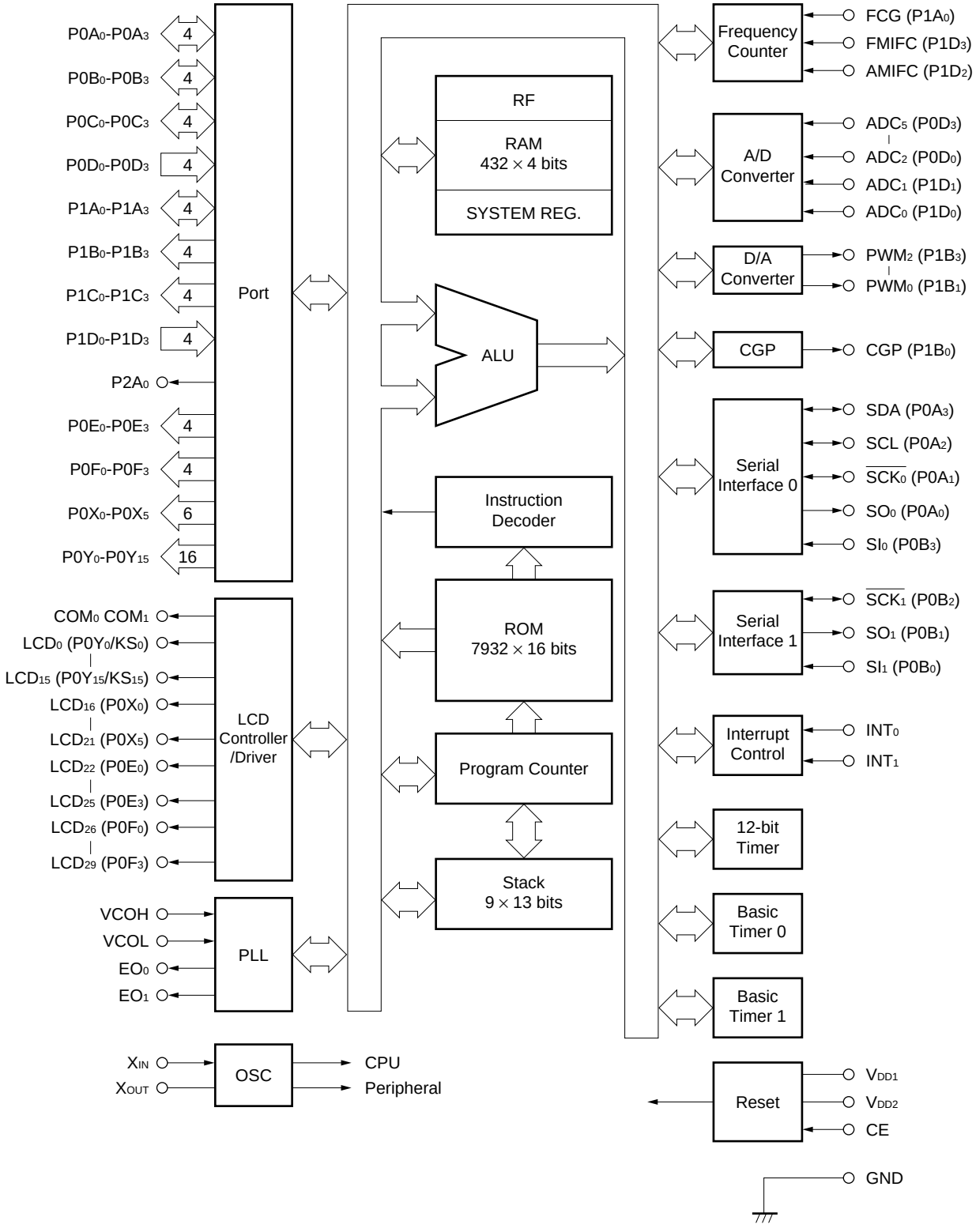
PIN CONFIGURATION (Top View)



PIN NAME

ADC_0-ADC_5	:	A/D converter input	$P0F_0$-$P0F_3$	:	Port 0F
AMIFC	:	AM intermediate frequency counter input	$P0X_0$-$P0X_5$	:	Port 0X
CE	:	Chip enable input	$P0Y_0$-$P0Y_{15}$	:	Port 0Y
CGP	:	Clock generator port	$P1A_0$-$P1A_3$	:	Port 1A
COM_0, COM_1	:	LCD common signal output	$P1B_0$-$P1B_3$	:	Port 1B
EO_0, EO_1	:	Error out output	$P1C_0$-$P1C_3$	:	Port 1C
FCG	:	Frequency count input for external gate	$P1D_0$-$P1D_3$	:	Port 1D
FMIFC	:	FM intermediate frequency counter input	$P2A_0$	:	Port 2A
GND	:	Ground	PWM_0-PWM_2	:	D/A converter output
INT_0, INT_1	:	External interrupt input	$\overline{SCK_0}$, $\overline{SCK_1}$	:	Serial clock I/O
KS_0-KS_{15}	:	Key source signal output	SCL	:	Serial clock I/O
LCD_0-LCD_{29}	:	LCD segment signal output	SDA	:	Serial data I/O
NC	:	No connection	SI_0, SI_1	:	Serial data input
$P0A_0$-$P0A_3$	:	Port 0A	SO_0, SO_1	:	Serial data output
$P0B_0$-$P0B_3$	:	Port 0B	VCOH	:	Local oscillation input
$P0C_0$-$P0C_3$	:	Port 0C	VCOL	:	Local oscillation input
$P0D_0$-$P0D_3$	:	Port 0D	V_{DD1}, V_{DD2}	:	Power supply
$P0E_0$-$P0E_3$	:	Port 0E	X_{IN}, X_{OUT}	:	Crystal resonator connection

BLOCK DIAGRAM



CONTENTS

1. PIN FUNCTIONS

1.1 Pin Function List

Pin No.	Symbol	Function	Output Format	Power-ON Reset
79 80 1 2	$P0C_3$ $P0C_2$ $P0C_1$ $P0C_0$	4-bit I/O port. Can be set in input or output mode in 4-bit units.	CMOS push-pull	Input
3 4 5 6 7 8 9 10	$P0A_3$/SDA[Note] $P0A_2$/SCL[Note] $P0A_1/\overline{SCK_0}$ $P0A_0/SO_0$ $P0B_3/SI_0$ $P0B_2/\overline{SCK_1}$ $P0B_1/SO_1$ $P0B_0/SI_1$	Port 0A, port 0B, or serial interface I/O. • $P0A_3$-$P0A_0$ • 4-bit I/O port • Can be set in input or output mode in 1-bit units • $P0B_3$-$P0B_0$ • 4-bit CMOS I/O port • Can be set in input or output mode in 1-bit units • SDA, SCL • SDA : Serial data I/O • SCL : serial clock I/O • $\overline{SCK_0}$, SO_0, SI_0 • $\overline{SCK_0}$: Serial clock I/O • SO_0 : Serial data output • SI_0 : Serial data input (SDA and SCL cannot be used simultaneously with $\overline{SCK_0}$, SI_0, and SO_0.) • $\overline{SCK_1}$, SO_1, SI_1 output • $\overline{SCK_1}$: Serial clock I/O • SO_1 : Serial data output • SI_1 : Serial data input SDA, SCL, $\overline{SCK_0}$, SI_0, $\overline{SCK_1}$, and SI_1 are Schmitt trigger input pins with hysteresis.	N-ch open-drain Withstanding 5 V ($P0A_3$/SDA, $P0A_2$/SCL) CMOS push-pull [$P0A_1/\overline{SCK_0}$, $P0A_0/SO_0$, $P0B_3$, $P0B_2/\overline{SCK_1}$, $P0B_1/SO_1$, $P0B_0$]	Input ($P0A_3$-$P0A_0$, $P0B_3$-$P0B_0$)
11 12	INT_1 INT_0	Edge-detectable vector interrupts. Either rising edge or falling edge can be selected. These pins are Schmitt trigger input pins with hysteresis. **Exercise care that voltage higher than V_{DD} is not applied to $INTP_0$ pin on power application. If voltage higher than V_{DD} is applied, μPD17010 may not operate correctly.**	—	Input

Note The $P0A_3$/SDA and $P0A_2$/SCL are N-ch open-drain output pins and must be connected with external pull-up resistors.

Pin No.	Symbol	Function	Output Format	Power-ON Reset
13	CE	Selects operation of μPD17010 and inputs reset signal. (1) Device operation selection The PLL frequency synthesizer can operate while CE pin is high. When CE pin is low, the PLL frequency synthesizer is automatically disabled (operation inhibited) internally. (2) Reset signal input When CE pin goes high, device is reset in synchronization with internal basic timer 0 carry FF (CE reset). This pin does not accept high or low level of less than 110 to 165 μs to prevent malfunctioning due to noise. Input signal level of this pin can be detected by CEJDG register (address 07H) of register file. At this time, contents of CEJDG register are not changed by low or high level of less than 110 to 165 μs. This pin is Schmitt trigger input pin with hysteresis. **Exercise care that voltage higher than V_{DD} is not applied to this pin on power application. If voltage higher than V_{DD} is applied, μPD17010 may not operate correctly.**	—	Input
14 | 16 17	$P1A_3$ | $P1A_1$ $P1A_0$/FCG	I/O of port 1A and input of external gate counter. • $P1A_3$-$P1A_0$ • 4-bit CMOS I/O port • Can be set in input or output mode in 1-bit units • FCG • Input to frequency counter for external gate	CMOS push-pull ($P1A_3$-$P1A_0$)	Input ($P1A_3$-$P1A_0$)
18 19 20 21	$P1B_3$/PWM_2[Note] $P1B_2$/PWM_1[Note] $P1B_1$/PWM_0[Note] $P1B_0$/CGP	Port 1B, and output of D/A converter and clock generator port. • $P1B_3$-$P1B_0$ • 4-bit output port • PWM_2-PWM_0 • Output of D/A converter with 8-bit resolution • CGP • Clock generator port output	N-ch open-drain Withstanding 16 V [$P1B_3$-PWM_2 | $P1B_1$-PWM_0] CMOS push-pull ($P1B_0$-CGP)	Outputs undefined data ($P1B_3$-$P1B_0$)
22 | 25	$P1C_3$ | $P1C_0$	4-bit CMOS output port	CMOS push-pull	Outputs undefined data

Note The $P1B_3$/PWM_2 through $P1B_1$/PWM_0 are N-ch open-drain output pins and must be connected with external pull-up resistors.

<table>
<tr><th>Pin No.</th><th>Symbol</th><th>Function</th><th>Output Format</th><th>Power-ON Reset</th></tr>
<tr><td>26
27
28
29</td><td>P1D3/FMIFC
P1D2/AMIFC
P1D1/ADC1
P1D0/ADC0</td><td>Port 1D, input to frequency counter, and analog input to A/D converter
• P1D3-P1D0
• 4-bit input port
• FMIFC, AMIFC
• Frequency measurable with FM and AM intermediate frequency counters
<table>
<tr><th>Input Pin</th><th>Input Frequency (MHz)</th><th>Input Amplitude (V_{P-P})</th></tr>
<tr><td rowspan="2">P1D3/FMIFC</td><td>5-15</td><td>0.3</td></tr>
<tr><td>10.5-10.9</td><td>0.06</td></tr>
<tr><td rowspan="2">P1D2/AMIFC</td><td>0.1-1</td><td>0.3</td></tr>
<tr><td>0.44-0.46</td><td>0.05</td></tr>
</table>
These pins are input pins to AC amplifier. Cut off DC components of input signals with capacitor.
• ADC1, ADC0
• Analog inputs to 6-bit resolution A/D converter</td><td>—</td><td>Input (P1D3-P1D0)</td></tr>
<tr><td>30
41</td><td>VDD1
VDD2</td><td>Positive power supply. Supplies 5 V ±10% when CPU and peripheral functions operate. When clock is stopped, data can be retained at 2.2 V. When VDD rises, internal power-ON reset circuit resets μPD17010.
Do not apply voltage higher than VDD pin to any pin other than VDD pins (VDD1 and VDD2 pins). Especially exercise care when raising both VDD and CE pins simultaneously as it may cause latch up.
Be sure to connect VDD1 and VDD2 pins to the same voltage level.
VDD2 pin supplies power to crystal oscillation circuit (XIN and XOUT pins) and error out circuit (EO0 and EO1 pins), and VDD1 pin supplies power to the other circuits.</td><td>—</td><td>—</td></tr>
</table>

<table>
<tr><th>Pin No.</th><th>Symbol</th><th>Function</th><th>Output Format</th><th>Power-ON Reset</th></tr>
<tr><td>31
32</td><td>VCOL
VCOH</td><td>Inputs local oscillation frequency to PLL. Two types of division modes are selectable: direct division (MF mode) and pulse swallow division (HF and VHF modes).
<table><tr><th>Division Mode</th><th>Input Pin</th><th>Input Frequency (MHz)</th><th>Input Voltage (V_{P-P})</th></tr><tr><td>Direct division (MF)</td><td>VCOL</td><td>0.5-30</td><td>0.3</td></tr><tr><td>Pulse swallow (HF)</td><td>VCOL</td><td>5-40</td><td>0.3</td></tr><tr><td>Pulse swallow (VHF)</td><td>VCOH</td><td>9-150</td><td>0.3</td></tr></table>These pins are input pins to AC amplifier. Cut off DC components of input signals with capacitor.</td><td>—</td><td>Input</td></tr>
<tr><td>33</td><td>GND</td><td>Ground</td><td>—</td><td>—</td></tr>
<tr><td>34</td><td>X_{OUT}Note</td><td>Connects crystal resonator.</td><td>CMOS push-pull</td><td rowspan="2">—</td></tr>
<tr><td>35</td><td>X_{IN}Note</td><td>Connect 4.5-MHz crystal resonator to these pins.</td><td>—</td></tr>
<tr><td>36
37</td><td>EO_0
EO_1</td><td>Output from charge pump of PLL frequency synthesizer.
If the value resulting from dividing local oscillation (VCO) frequency input to VCOL pin (pin 31) or VCOH pin (pin 32) is higher than reference frequency, EO_0 and EO_1 pins output high level; if it is lower than reference frequency, EO_0 and EO_1 pins output low level. If it coincides with reference frequency, EO_0 and EO_1 pins float.
Because the same signal is output to EO_0 and EO_1 pins, either pin can be used.</td><td>CMOS 3-state</td><td>High impedance</td></tr>
<tr><td>38
|
40</td><td>NC</td><td>No connection</td><td>—</td><td>—</td></tr>
<tr><td>42</td><td>$P2A_0$</td><td>1-bit CMOS output port</td><td>CMOS push-pull</td><td>Outputs undefined data</td></tr>
<tr><td>43
44</td><td>COM_1
COM_0</td><td>Outputs common signal of LCD controller/driver. These pins output low level in display off mode, at power-ON reset, and on execution of clock stop instruction.</td><td>CMOS ternary output</td><td>Low-level output</td></tr>
</table>

Note Refer to **APPENDIX A. NOTES ON CONNECTING CRYSTAL RESONATOR**.

Pin No.	Symbol	Function	Output Format	Power-ON Reset
45 | 48 49 | 52 53 | 58 59 | 74	$LCD_{29}/P0F_3$ | $LCD_{26}/P0F_0$ $LCD_{25}/P0E_3$ | $LCD_{22}/P0E_0$ $LCD_{21}/P0X_5$ | $LCD_{16}/P0X_0$ $LCD_{15}/P0Y_{15}/KS_{15}$ | $LCD_0/P0Y_{15}/KS_0$	Output of ports 0F, 0E, 0X, 0Y, segment signal output of LCD controller/driver, and key source signal output of key matrix. • $P0F_3$-$P0F_0$ • 4-bit CMOS output port • $P0E_3$-$P0E_0$ • 4-bit CMOS output port • $P0X_5$-$P0X_0$ • 6-bit CMOS output port • $P0Y_{15}$-$P0Y_0$ • 16-bit CMOS output port • LCD_{29}-LCD_0 • Segment signal output of LCD controller/driver • KS_{15}-KS_0 • Key source signal output of key matrix	CMOS push-pull	Low-level output (LCD_{29}-LCD_0)
75 | 78	$P0D_3/ADC_5$ | $P0D_0/ADC_2$	Port 0D, analog input to A/D converter, and key source signal return input to LCD segment. • $P0D_3$-$P0D_0$ • 4-bit input port • Internal pull-down resistor is always on. • ADC_5-ADC_2 • Analog input to 6-bit resolution A/D converter • Internal pull-down resistor is off. • Key source signal return input • Internal pull-down resistor is on only during key source output (220 μs) when LCD segment pin is used as key source, and is off during LCD segment signal output.	—	Input with pull-down resistor ($P0D_3$-$P0D_0$)

1.2 Notes on Using General-Purpose Ports

1.2.1 Data bits of port register

To read the input data of and to set output data to ports 0A, 0B, 0C, 0D, 1A, 1B, 1C, 1D, and 2A, the corresponding port register (P0A through P2A registers) in the data memory is used.

At this time, the $P0A_3$ pin of port 0A corresponds to the most significant bit of port register P0A, and $P0A_0$ pin corresponds to the least significant bit.

The same applies to ports 0B, 0C, 0D, 1A, 1B, 1C, 1D, and 2A.

Output data is set to ports 0E, 0F, 0X, and 0Y by the LCD group register via the LCD segment register on the data memory or the data buffer.

1.2.2 I/O ports (ports 0A, 0B, 0C, and 1A)

(1) When each port is set in input mode

When an instruction that reads the contents of each port register on the data memory (when the address of the port register is specified as m of SKT m, #n or ADD r, m) is executed, the status of each port pin is used as the value of the port register.

When an instruction that writes data to a port register (specified by m of MOV m, #n4 or r of ADD r, m) is executed, the value of that data is written to that output data latch circuit.

(2) When each port is set in output mode

When an instruction that writes data to each port register is executed, the value of that data is written to the output data latch circuit, and is output from each pin.

When an instruction that reads the contents of each port register is executed, the contents of the output data latch are used as the value of the port register. However, if an instruction that reads the contents of a port register is executed to the $P0A_3$/SDA and $P0A_2$/SCL pins, the pin status which is different from the output data may be read.

All the above port pins are set in the input mode at power-ON reset, CE reset, and on execution of the clock stop instruction.

At power-ON reset, the contents of the output data latch circuit are undefined. Unless data is written to the port register before a port is set in the output mode, therefore, undefined data is output. At CE reset and on execution of the clock stop instruction, the contents of the output data latch circuit remain unchanged.

1.2.3 Output ports (ports 1B, 1C, 0F, 0E, 0X, and 0Y)

An output port writes the value of a port register to the output data latch circuit and outputs this value from each output pin when an instruction that writes data to the port register is executed.

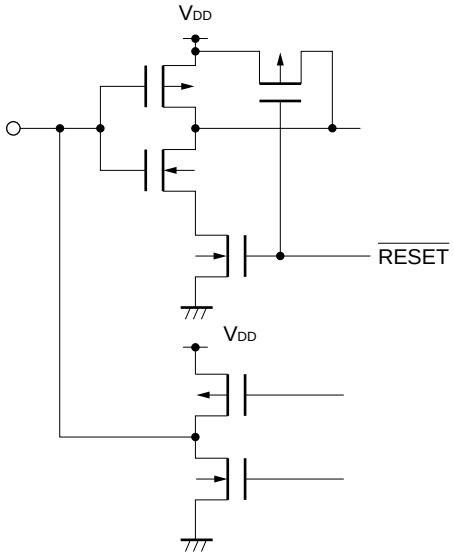
When an instruction that reads the value of the port register is executed, the status of the output data latch circuit is set to the port register.

At power-ON reset, undefined data is output.

At CE reset and on execution of the clock stop instruction, the previously output data is retained. However, ports 0E, 0F, 0X, and 0Y automatically output a low level at power-ON reset, and also on execution of the clock stop instruction.

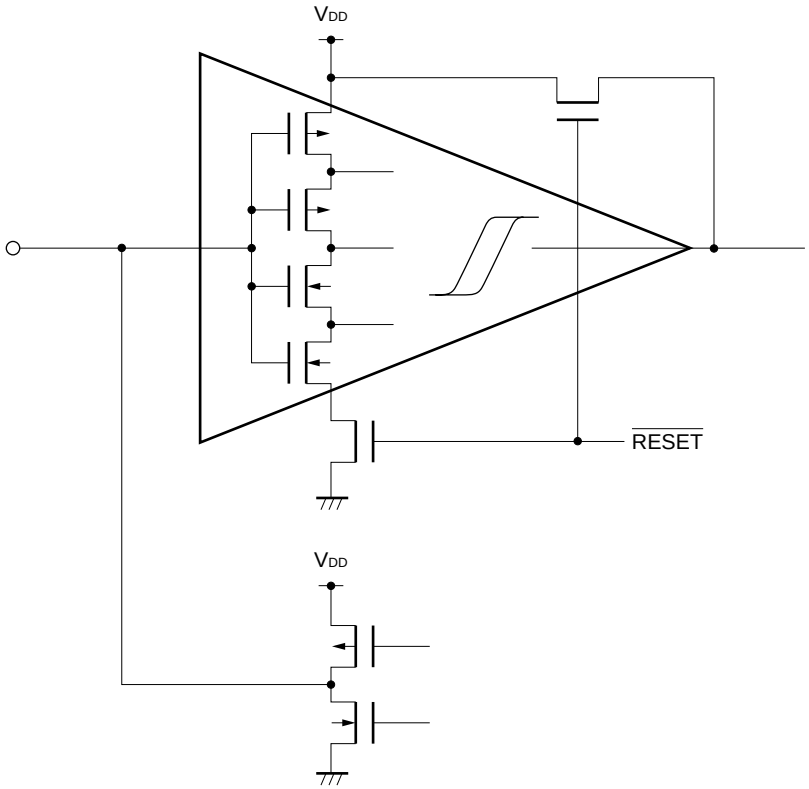
1.3 Equivalent Circuits of Pins

(1) P0A (P0A$_0$/SO$_0$)
P0B (P0B$_1$/SO$_1$)
P0C (P0C$_3$, P0C$_2$, P0C$_1$, P0C$_0$)[Note]
P1A (P1A$_3$, P1A$_2$, P1A$_1$, P1A$_0$/FCG)
(I/O)

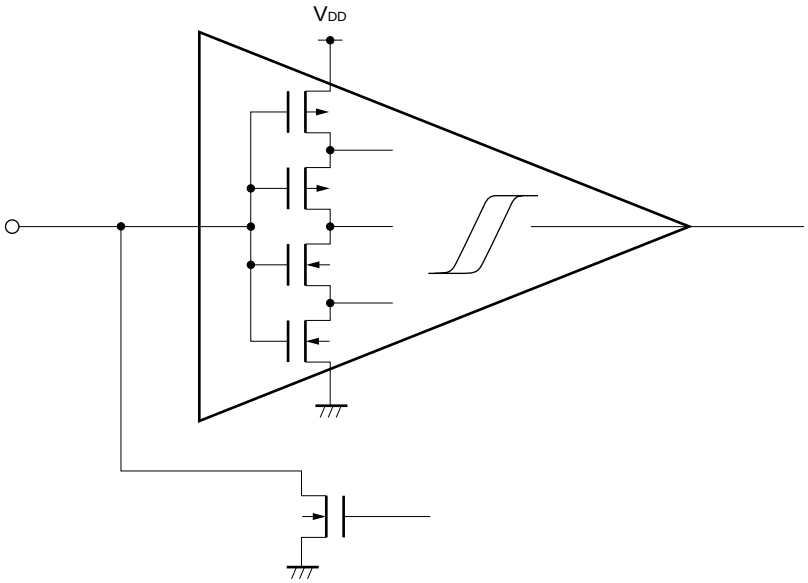


Note The $\overline{\text{RESET}}$ signal is not supplied to P0C.

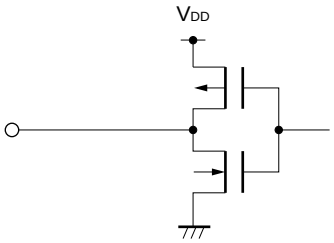
(2) P0A (P0A$_1$/$\overline{\text{SCK}}_0$),
P0B (P0B$_3$/SI$_0$, P0B$_2$/$\overline{\text{SCK}}_1$, P0B$_0$/SI$_1$)
(hysteresis input or output)



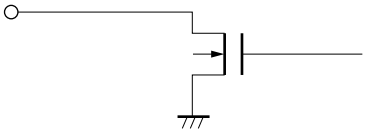
(3) P0A (P0A$_3$/SDA, P0A$_2$/SCL) (hysteresis input or output)



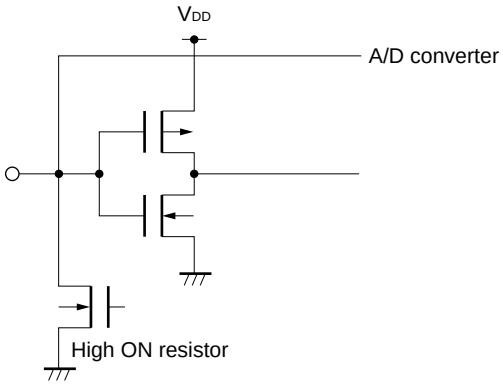
(4) P1B (P1B$_0$/CGP)
P1C (P1C$_3$, P1C$_2$, P1C$_1$, P1C$_0$)
P2A (P2A$_0$)
LCD$_0$/P0Y$_0$/KS$_0$-LCD$_{29}$/P0F$_3$ } **(output)**



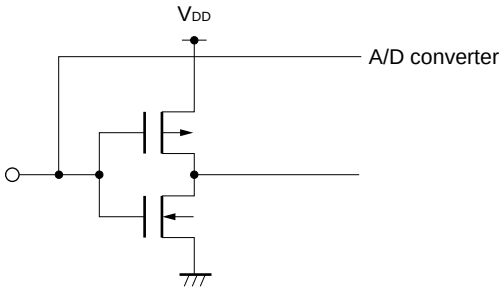
(5) P1B (P1B$_3$/PWM$_2$, P1B$_2$/PWM$_1$, P1B$_1$/PWM$_0$) (output)



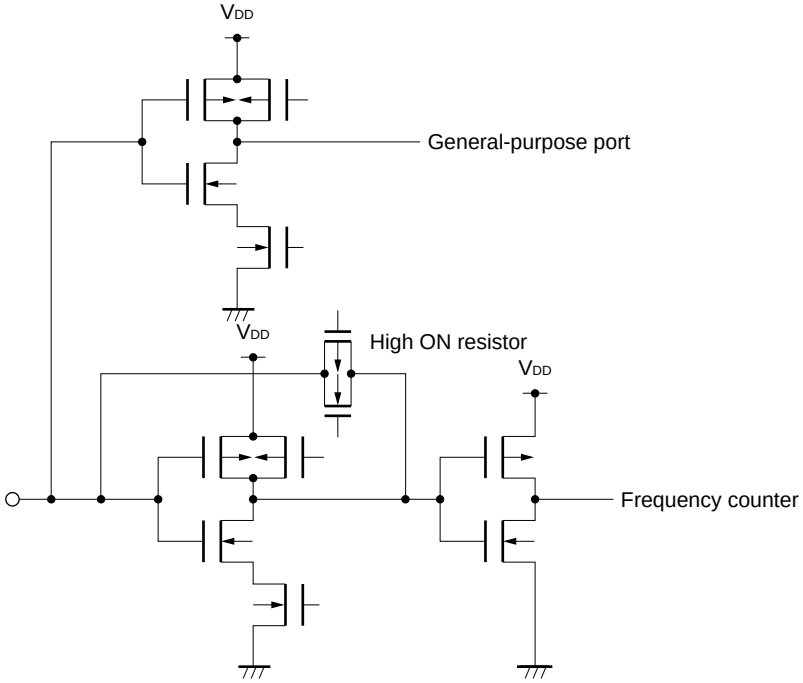
(6) P0D (P0D3/ADC5, P0D2/ADC4, P0D1/ADC3, P0D0/ADC2) (input)



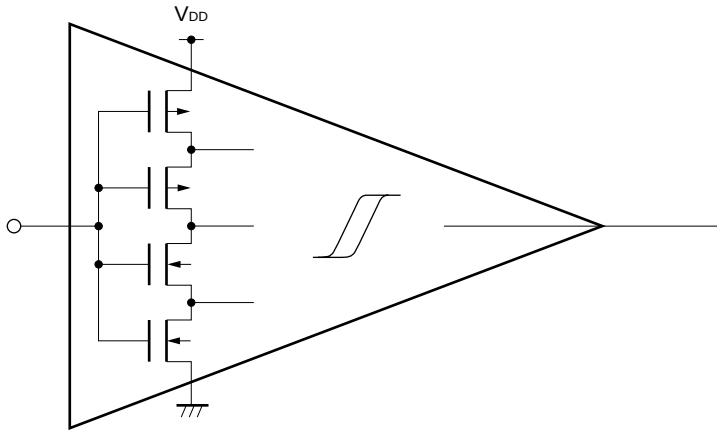
(7) P1D (P1D1/ADC1, P1D0/ADC0) (input)



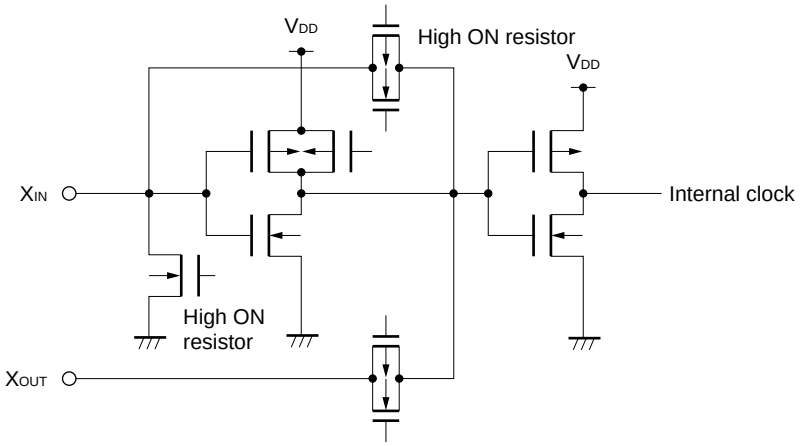
(8) P1D (P1D3/FMIFC, P1D2/AMIFC) (input)



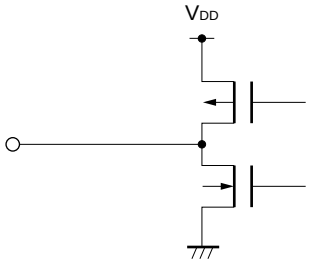
(9) CE, INT_1, INT_0 (Schmitt trigger input)



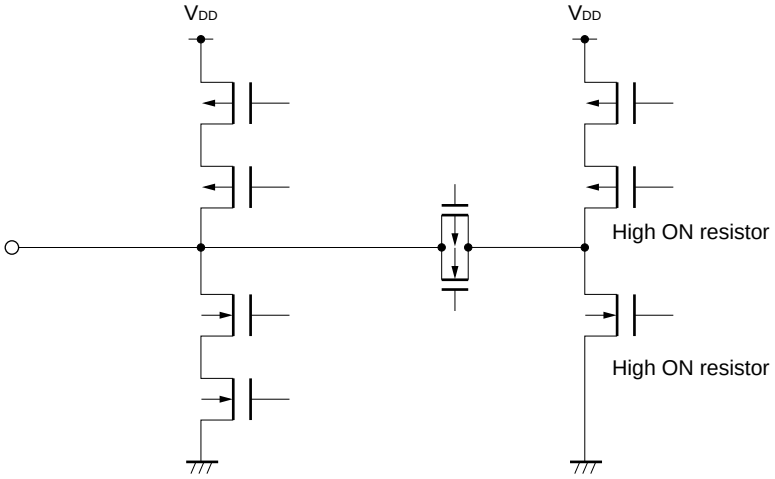
(10) X_{OUT} (output), X_{IN} (input)



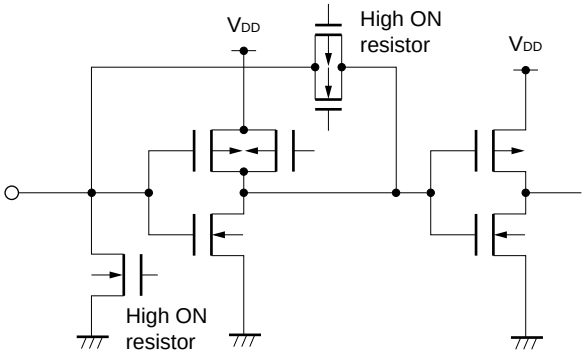
(11) EO_1, EO_0 (output)



(12) COM_1 / COM_0 (output)



(13) VCOH / VCOL (input)



1.4 Processing of Unused Pins

It is recommended that the unused pins be processed as shown below.

Table 1-1. Processing of Unused Pins

Pin Name		I/O	Recommended Processing
Port pins	$P0D_0/ADC_2$-$P0D_3/ADC_5$	Input	Individually connect to GND via resistor[Note 1]
	$P1D_0/ADC_0$[Note 2]		Individually connect to V_{DD} or GND via resistor[Note 1]
	$P1D_1/ADC_1$[Note 2]		
	$P1D_2$/AMIFC[Notes 2, 3]		Set as $P1D_2$ and connect to V_{DD} or GND via resistor[Note 1]
	$P1D_3$/FMIFC[Notes 2, 3]		Set as $P1D_3$ and connect to V_{DD} or GND via resistor [Note 1]
	$P0E_0/LCD_{22}$-$P0E_3/LCD_{25}$	CMOS push-pull output	Open
	$P0F_0/LCD_{26}$-$P0F_3/LCD_{29}$		
	$P0X_0/LCD_{16}$-$P0X_5/LCD_{21}$		
	$P0Y_0/LCD_0/KS_0$- $P0Y_{15}/LCD_{15}/KS_{15}$		
	$P1B_0$/CGP		
	$P1C_0$-$P1C_3$		
	$P2A_0$		
	$P1B_1/PWM_0$-$P1B_3/PWM_2$	N-ch open-drain output	Set to low level output via software, and open
	$P0A_0/SO_0$	I/O[Note 4]	Set as general-purpose input port via software, and connect each pin to V_{DD} or GND via resistor[Note 1]
	$P0A_1/\overline{SCK_0}$		
	$P0A_2$/SCL		
	$P0A_3$/SDA		
	$P0B_0/SI_1$		
	$P0B_1/SO_1$		
	$P0B_2/\overline{SCK_1}$		
	$P0B_3/SI_0$		
	$P0C_1/P0C_3$		
	$P1A_0$/FCG[Note 2]		
	$P1A_1/P1A_3$[Note 2]		
Pins other than port pins	CE	Input	Connect to V_{DD} via resistor[Note1]
	INT_0, INT_1		Connect each pin to GND via resistor[Note1]
	VCOH, VCOL		Set disable via software, and open
	COM_0, COM_1	Output	Open
	EO_0, EO_1		

Notes 1. If a pin is externally pulled up (connecting to V_{DD} via resistor) or down (connecting to GND via resistor) with a high resistance, the pin almost goes into a high impedance state. This increases the (inrush) current dissipation of the port. The value of the pull-up or pull-down resistor is generally several 10 kilohms, though this varies and depends on the application circuit.

2. The current dissipation of the general-purpose input port does not increase even in the high-impedance state.

3. Do not set these pins as AMIFC and FMIFC; otherwise, the current dissipation increases.

4. The I/O ports serve as general-purpose input ports at power application, clock stop, and CE reset.

1.5 Notes on Using CE, INT_0, and INT_1 Pins

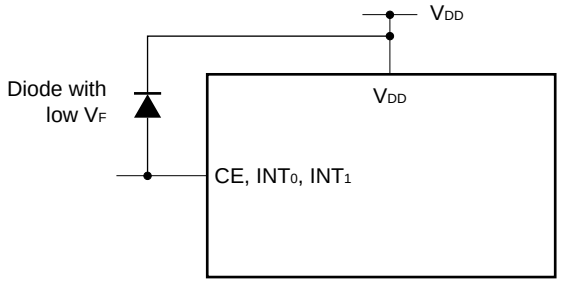
The CE, INT_0, and INT_1 pins have a function to set a test mode (for IC test) in which the internal operations of the μPD17010 are tested, in addition to the functions indicated in **1.1 Pin Function List**.

If a voltage higher than V_{DD} is applied to any of these pins, the test mode is set. If a noise higher than V_{DD} is added on any of these pins in the normal operation mode, therefore, the test mode is set by mistake.

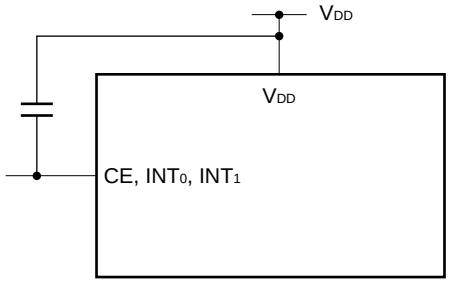
This may happen if the wiring length of the CE, INT_0, and INT_1 pins is too long and as a result, noise is added to the circuitry.

Therefore, keep the wiring of these pins as short as possible to suppress the noise. If necessary, use an external component as shown below to suppress the noise.

- **Connect diode with low V_F between V_{DD}**



- **Connect capacitor between V_{DD}**



2. PROGRAM MEMORY (ROM)

2.1 Outline of Program Memory

Figure 2-1 outlines the program memory.

As shown in this figure, the program memory consists of a program memory and a program counter.

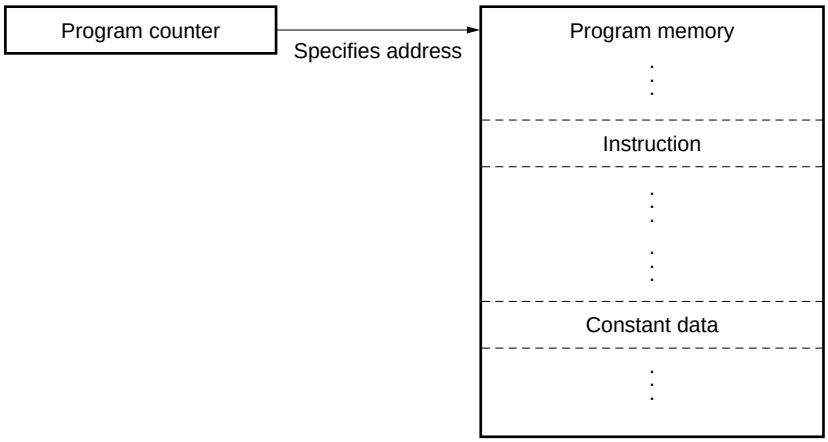
The addresses of the program memory are specified by the program counter.

The program memory has the following two major functions:

(1) Stores executed instructions

(2) Stores constant data

Figure 2-1. Outline of Program Memory



2.2 Program Memory

Figure 2-2 shows the configuration of the program memory.

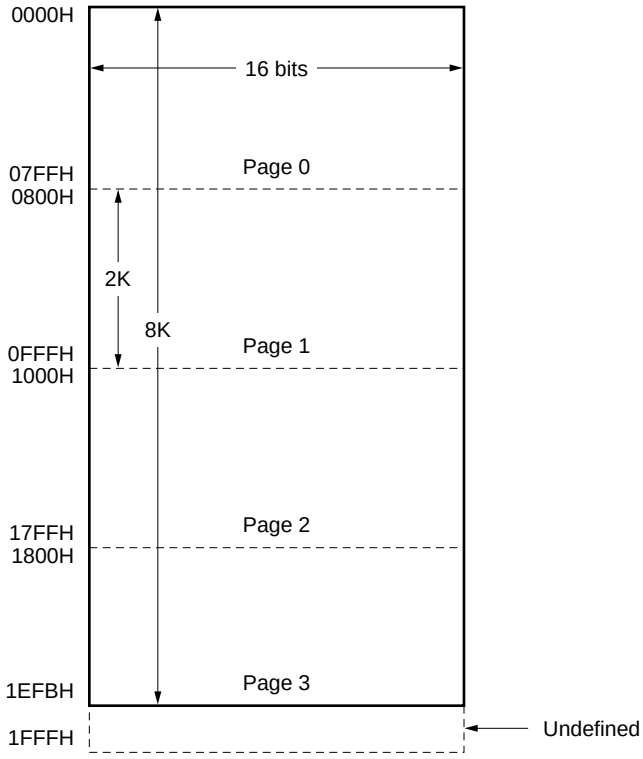
As shown, the program memory consists of 7932 steps by 16 bits.

Therefore, the program memory address ranges from 0000H to 1EFBH.

Because all “instructions” are 16-bit long “1-word instructions”, one instruction can be stored in one program memory address.

Constant data reads the contents of the program memory to the data buffer by using a table reference instruction.

Figure 2-2. Program Memory Configuration



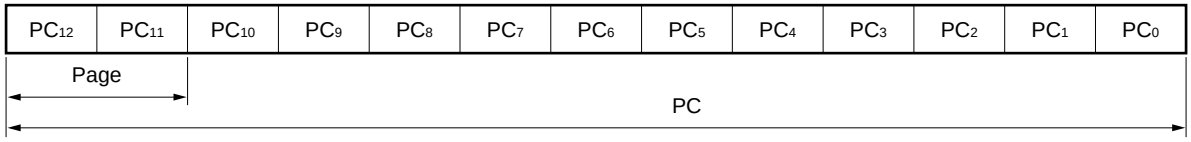
2.3 Program Counter

Figure 2-3 shows the configuration of the program counter.

As shown, the program counter is a 13-bit binary counter. The highest 2 bits, bits b_{11} and b_{12}, indicate a page.

The program counter specifies an address of the program memory.

Figure 2-3. Program Counter Configuration



2.4 Program Flow

The program flow is controlled by the program counter that specifies an address of the program memory.

The operation to be performed when each instruction is executed is described below.

Figure 2-4 shows the value set to the program counter when each instruction is executed.

Table 2-1 shows the vector address to be used when an interrupt is accepted.

2.4.1 Branch instruction

(1) Direct branch (“BR addr”)

The branch destination address of the direct branch instruction ranges from 0000H through 1EFBH, i.e., any address of the program memory.

(2) Indirect branch (“BR @AR”)

The branch destination address of the indirect branch instruction ranges from 0000H through 1EFBH, i.e., any address of the program memory.

For more information, refer to **5.3 Address Register (AR)**.

2.4.2 Subroutine

(1) Direct subroutine call (“CALL addr”)

The first address of a subroutine that can be called by the direct subroutine call instruction is within page 0 (address 0000H to 07FFH) of the program memory.

(2) Indirect subroutine call (“CALL @AR”)

The first address of a subroutine that can be called by the indirect subroutine call instruction ranges from 0000H to 1EFBH, i.e., any address of the program memory.

For more information, refer to **5.3 Address Register (AR)**.

2.4.3 Table reference

The address that can be referenced by the table reference instruction (“MOVT DBF, @AR”) ranges from 0000H to 1EFBH, i.e., any address of the program memory.

For more information, refer to **5.3 Address Register (AR)** and **9.2.2 Table reference instruction (“MOVT DBF, @AR”)**.

Figure 2-4. Specification by Program Counter for Each Instruction

Instruction \ Program Counter		Contents of Program Counter (PC)												
		b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
BR addr	Page 0	0	0	Instruction operand (addr)										
	Page 1	0	1											
	Page 2	1	0											
	Page 3	1	1											
CALL addr		0	0	Instruction operand (addr)										
BR @AR CALL @AR MOVT DBF, @AR		Contents of address register												
RET RETSK RETI		Contents of address stack register (ASR) specified by stack pointer (SP) (return address)												
When interrupt is accepted		Vector address of each interrupt												
At power-ON or CE reset		0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2-1. Interrupt Vector Address

Order	Internal/External	Interrupt Source	Vector Address
1	External	INT_0 pin	0006H
2	Internal/external	INT_1 pin or timer/counter overflow	0005H
3	Internal	12-bit timer	0004H
4	Internal	Basic timer 1	0003H
5	Internal	Serial interface 0	0002H
6	Internal	Frequency counter	0001H

2.5 Notes on Using Program Memory

The address that can be specified by the program counter ranges from 0000H to 1FFFH. By contrast, the program memory exist at addresses 0000H through 1EFBH.

Therefore, do not use an instruction that sets the value of the program counter to 1EFCH to 1FFFH.

The addresses 1EFCH through 1FFFH of the program memory are “undefined” values.

3. ADDRESS STACK (ASK)

3.1 Outline of Address Stack

Figure 3-1 outlines the address stack.

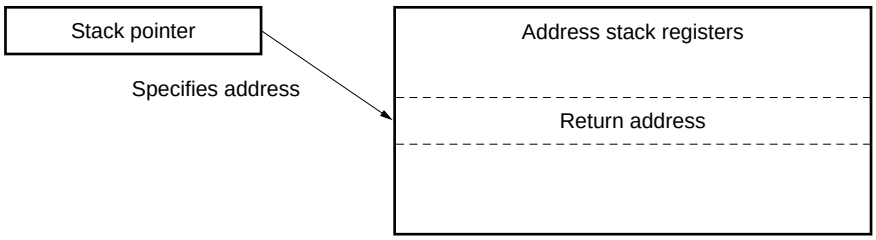
The address stack consists of a stack pointer and address stack registers.

The addresses of the address stack registers are specified by the stack pointer.

The address stack saves a return address when a subroutine call instruction is executed or when an interrupt is accepted.

The address stack is also used when the table reference instruction is executed.

Figure 3-1. Outline of Address Stack



3.2 Address Stack Register (ASR)

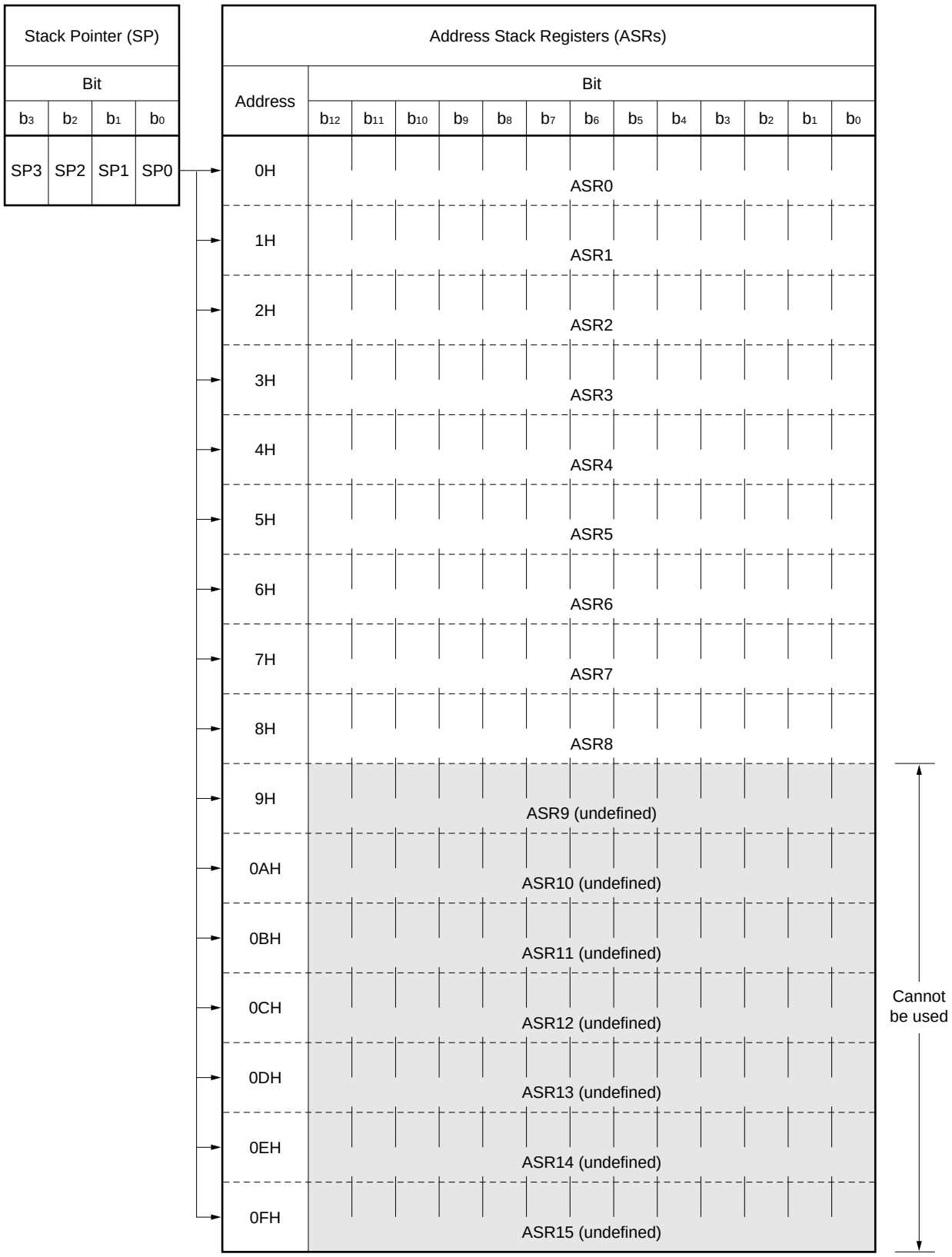
Figure 3-2 shows the configuration of the address stack registers.

Sixteen 16-bit address stack registers, ASR0 through ASR15, are available. However, registers are not allocated to ASR9 through ASR15. Actually, therefore, nine 16-bit registers (ASR0 through ASR8) can be used.

The higher 3 bits of ASR0 through ASR8 are fixed to “0”.

The address stack stores a return address when a subroutine call instruction or table reference instruction is executed, or when an interrupt is accepted.

Figure 3-2. Address Stack Registers Configuration



3.3 Stack Pointer (SP)

3.3.1 Configuration and function of stack pointer

Figure 3-3 shows the configuration and function of the stack pointer.

The stack pointer is a 4-bit binary counter.

It specifies the addresses of the address stack registers.

The value of the stack pointer can be directly read or written by using a register manipulation instruction.

Figure 3-3. Configuration and Function of Stack Pointer

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Stack pointer (SP)	SP3	SP2	SP1	SP0	01H	R/W

b_3	b_2	b_1	b_0	Specifies addresses of address stack registers (ASRs)
0	0	0	0	Address 0 (ASR0)
0	0	0	1	Address 1 (ASR1)
0	0	1	0	Address 2 (ASR2)
0	0	1	1	Address 3 (ASR3)
0	1	0	0	Address 4 (ASR4)
0	1	0	1	Address 5 (ASR5)
0	1	1	0	Address 6 (ASR6)
0	1	1	1	Address 7 (ASR7)
1	0	0	0	Address 8 (ASR8)
1	0	0	1	Address 9 (ASR9)
1	0	1	0	Address 10 (ASR10)
1	0	1	1	Address 11 (ASR11)
1	1	0	0	Address 12 (ASR12)
1	1	0	1	Address 13 (ASR13)
1	1	1	0	Address 14 (ASR14)
1	1	1	1	Address 15 (ASR15)

On reset	Power-ON	1	0	0	1
	Clock stop	1	0	0	1
	CE	1	0	0	1

3.4 Operation of Address Stack

3.4.1 Subroutine call instructions (“CALL addr”, “CALL @AR”) and return instructions (“RET”, “RETSK”)

When a subroutine call instruction is executed, the value of the stack pointer is decremented by one, and a return address is stored to the address stack register specified by the stack pointer.

When a return instruction is executed, the contents of the address stack register (return address) specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

3.4.2 Table reference instructions (“MOVT DBF, @AR”)

When a table reference instruction is executed, the value of the stack pointer is decremented by one, and a return address is stored to the address stack register specified by the stack pointer.

Next, the contents of the program memory specified by the address register are read to the data buffer, the contents of the address stack register (return address) specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

3.4.3 When interrupt is accepted or when return instruction (“RETI”) is executed

When an interrupt is accepted, the value of the stack pointer is decremented by one, and a return address is stored to the address stack register specified by the stack pointer.

When the return instruction is executed, the contents of the address stack register (return address) specified by the stack pointer is restored to the program counter, and the value of the stack pointer is incremented by one.

3.4.4 Address stack manipulation instructions (“PUSH AR”, “POP AR”)

When the “PUSH” instruction is executed, the value of the stack pointer is decremented by one, and the contents of the address register are transferred to the address stack register specified by the stack pointer.

When the “POP” instruction is executed, the contents of the address stack register specified by the stack pointer are transferred to the address register, and the value of the stack pointer is incremented by one.

3.5 Notes on Using Address Stack

3.5.1 Nesting level

The values of the address stack registers (ASR9 through ASR15) are “undefined” when the value of the stack pointer is 09H through 0FH.

If a subroutine call instruction or interrupt that exceeds level 9 is used without manipulating the stack, execution returns to an “undefined” address.

4. DATA MEMORY (RAM)

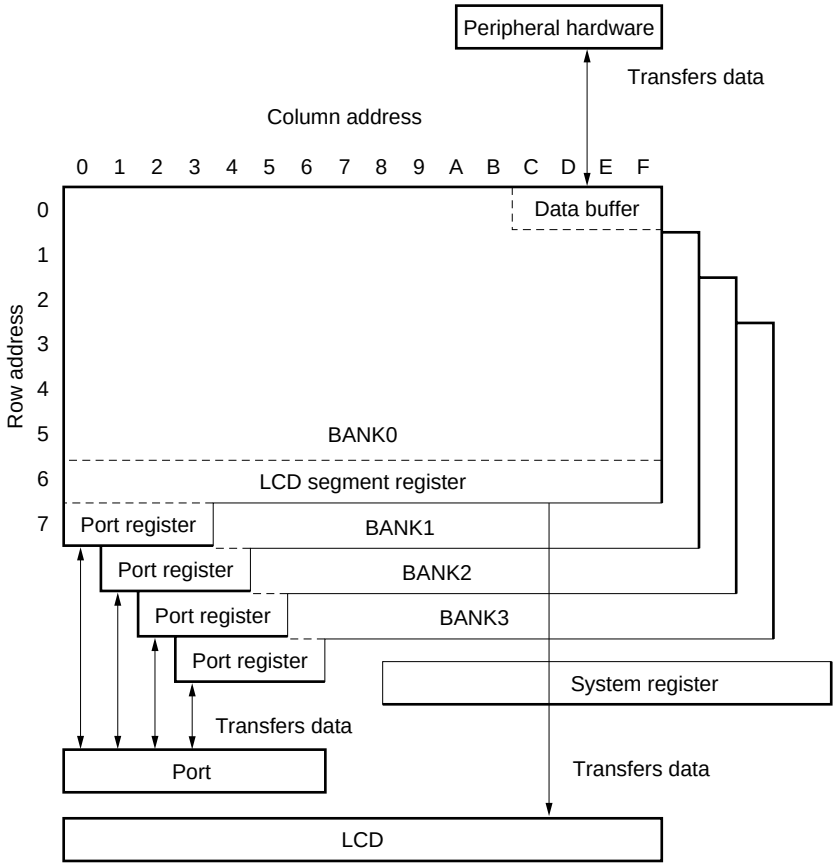
4.1 Outline of Data Memory

Figure 4-1 outlines the data memory.

As shown in this figure, the data memory consists of a general-purpose data memory, system register, data buffer, LCD segment register, and port register.

The data memory stores data, transfers data with peripheral hardware, sets display data, transfers data with ports, and controls the CPU.

Figure 4-1. Outline of Data Memory



4.2 Configuration and Function of Data Memory

Figure 4-2 shows the configuration of the data memory.

As shown in this figure, the data memory is divided into four banks with each bank consisting of a total of 128 nibbles (row address 7H and column address 0FH).

The data memory is divided by function into the six blocks as described in **4.2.1** through **4.2.6** below.

The contents of the data memory can be manipulated by using data memory manipulation instructions, and 4-bit data can be operated, compared, judged, and transferred with a single instruction.

Table 4-1 shows the data memory manipulation instructions.

4.2.1 System register (SYSREG)

The system register is allocated to addresses 74H through 7FH.

Because the system register is allocated regardless of banks, the same system register exist at addresses 74H through 7FH of any bank.

For details, refer to **5. SYSTEM REGISTER (SYSREG)**.

4.2.2 Data buffer (DBF)

The data buffer is allocated to addresses 0CH through 0FH of BANK0.

For details, refer to **9. DATA BUFFER (DBF)**.

4.2.3 LCD segment data register (LCD segment register)

The LCD segment register is allocated to addresses 60H through 6FH of BANK0 of the data memory.

For details, refer to **21. LCD CONTROLLER/DRIVER**.

4.2.4 Port data register (port register)

The port register is allocated to addresses 70H through 73H of each bank.

For details, refer to **15. GENERAL-PURPOSE PORTS**.

4.2.5 General-purpose data memory

The general-purpose data memory is allocated to an area of the data memory excluding the system register, LCD segment register, and port register.

This consists of a total of 432 nibbles (432 x 4 bits) of 96 nibbles of BANK0 and 112 words each of BANK1 through BANK3.

4.2.6 Not provided data memory

As a part of the LCD segment register and port register, a data memory area to which nothing is actually allocated exists.

For this data memory area, refer to **4.4.2 Notes on not provided data memory**, **15. GENERAL-PURPOSE PORTS**, and **21. LCD CONTROLLER/DRIVER**.

Figure 4-2. Data Memory Configuration

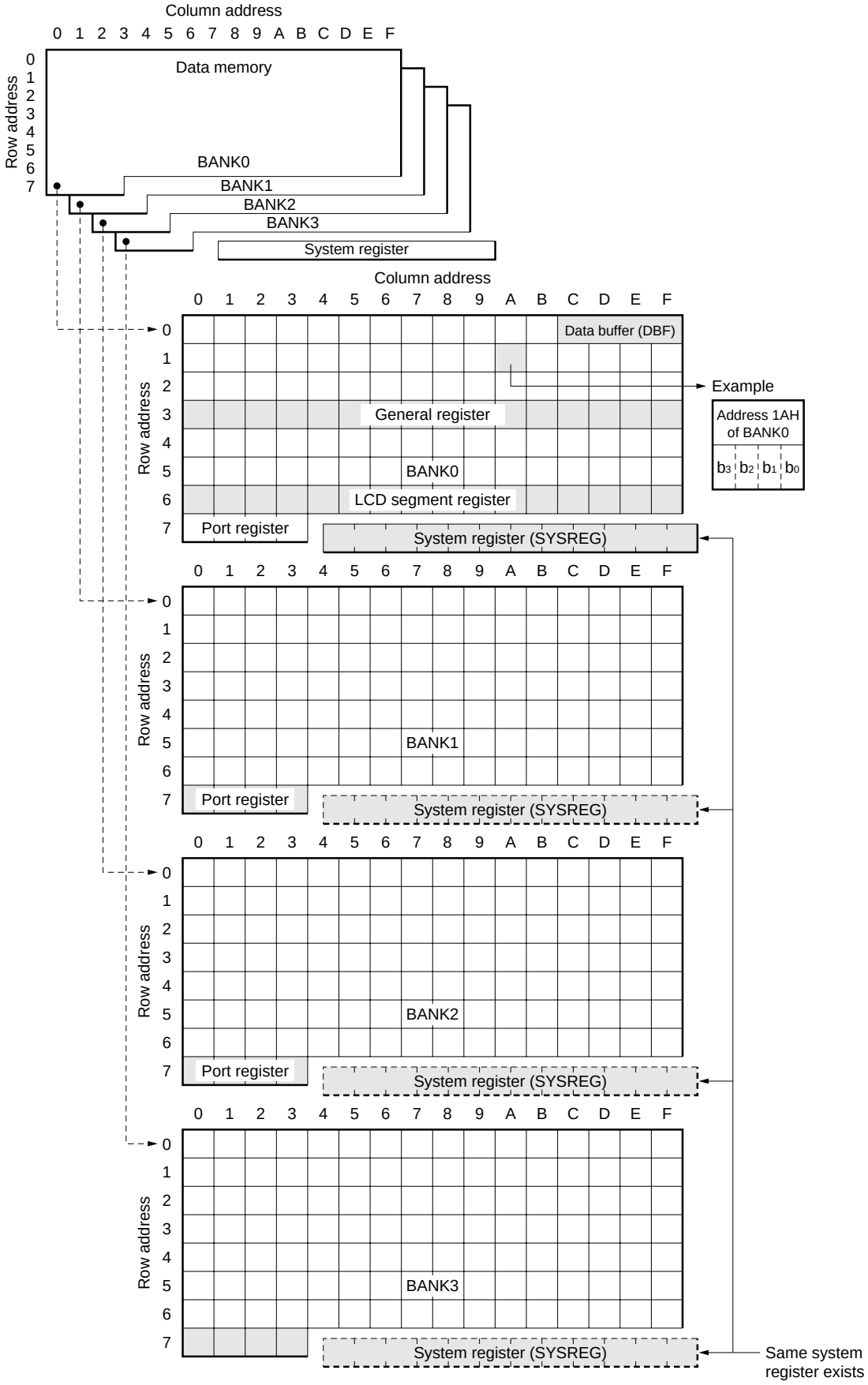


Table 4-1. Data Memory Manipulation Instruction List

Function		Instruction
Operation	Addition	ADD ADDC
	Subtraction	SUB SUBC
	Logical	AND OR XOR
Comparison		SKE SKGE SKLT SKNE
Transfer		MOV LD ST
Judgment		SKT SKF

4.3 Addressing of Data Memory

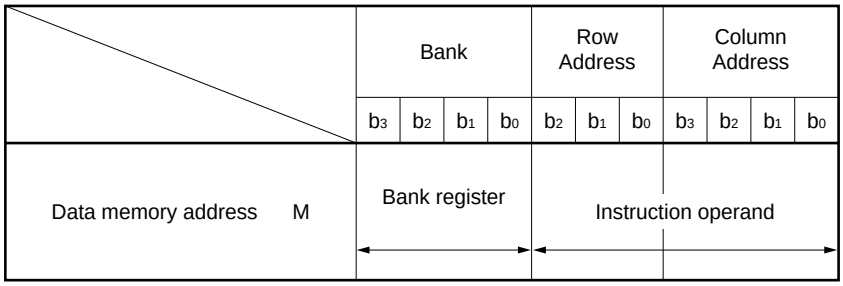
Figure 4-3 shows addressing of the data memory.

A data memory address is specified by bank, row and column addresses.

The row and column addresses are directly specified by using a data memory manipulation instruction, but the bank is specified by the contents of the bank register.

For the details of the bank register, refer to **5. SYSTEM REGISTER (SYSREG)**.

Figure 4-3. Addressing of Data Memory



4.4 Notes on Using Data Memory

4.4.1 On power-ON reset

At power-ON reset, the contents of the general-purpose data memory are “undefined”.

Initialize the general-purpose data memory as necessary.

4.4.2 Notes on not provided data memory

If a data memory manipulation instruction is executed to manipulate the address of a data memory area to which nothing has been allocated, the following operations are performed:

(1) Device operation

When a read instruction is executed, “0” is read.

Nothing is changed if a write instruction is executed.

(2) Assembler (AS17K) operation

Assembly is executed normally.

An error does not occur.

(3) Emulator (IE-17K) operation

When a read instruction is executed, “0” is read.

Nothing is changed if a write instruction is executed.

An error does not occur.

5. SYSTEM REGISTER (SYSREG)

5.1 Outline of System Register

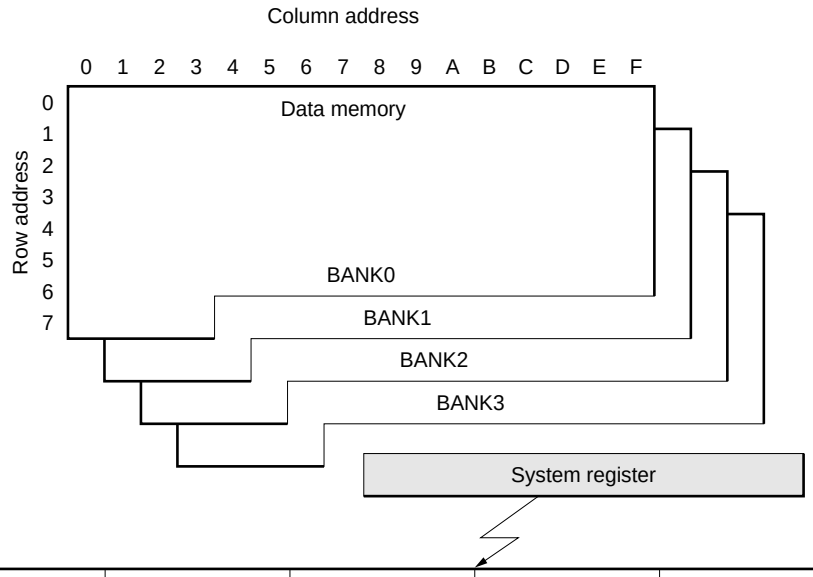
Figure 5-1 shows the location of the system register on the data memory and its outline.

As shown in this figure, the system register is located at addresses 74H through 7FH of the data memory independently of the bank. Therefore, the same system register exists at addresses 74H through 7FH of any bank.

Because the system register is located on the data memory, it can be manipulated by any data memory manipulation instruction.

The system register consists of seven types of registers by function.

Figure 5-1. Location on Data Memory and Outline of System Register



Address	74H	75H	76H	77H	78H	79H
Name	Address register (AR)				Window register (WR)	Bank register (BANK)
Outline	Controls program memory address				Transfers data with register file	Specifies bank of data memory

Address	7AH	7BH	7CH	7DH	7EH	7FH
Name	Index register (IX)			General register pointer (RP)		Program status word (PSWORD)
	Data memory row address pointer (MP)					
Outline	Modifies address of data memory			Specifies address of general register		Controls operation

5.2 System Register List

Figure 5-2 shows the configuration of the system register.

Figure 5-2. System Register Configuration

Address	74H				75H				76H				77H				78H				79H			
Name	System register																							
	Address register (AR)																Window register (WR)				Bank register (BANK)			
Symbol	AR3				AR2				AR1				AR0				WR				BANK			
Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data	0	0	0	←	→				→				→				↔				0	0	↔	

Address	7AH				7BH				7CH				7DH				7EH				7FH			
Name	System register																							
	Index register (IX)												General register pointer (RP)								Program status word (PSWORD)			
	Data memory row address pointer (MP)																							
Symbol	IXH				IXM				IXL				RPH				RPL				PSW			
	MPH				MPL																			
Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data	MPE	0	0	←	(MP) →				(IX) →				0	0	←		(RP) →				BCD	CMP	CY	Z	IXE

5.3 Address Register (AR)

5.3.1 Configuration of address register

Figure 5-3 shows the configuration of the address register.

As shown in this figure, the address register consists of 16 bits, 74H through 77H (AR3 through AR0), of the system register. Actually, however, it operates as a 13-bit register because the highest 3 bits are always fixed to “0”.

Figure 5-3. Address Register Configuration

Address		74H				75H				76H				77H			
Name		Address Register (AR)															
Symbol		AR3				AR2				AR1				AR0			
Bit		b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data		0	0	0	^ MSB v												^ LSB v
On reset	Power-ON	0				0				0				0			
	Clock stop	0				0				0				0			
	CE	0				0				0				0			

Remark Power-ON : on power-ON reset
Clock stop : on execution of clock stop instruction
CE : on CE reset

5.3.2 Function of address register

The address register specifies a program memory address when the table reference instruction (“MOVT DBF, @AR”), stack manipulation instruction (“PUSH AR”, “POP AR”), indirect branch instruction (“BR @AR”), or indirect subroutine call instruction “CALL @AR”) is executed.

A dedicated instruction (“INC AR”) that can increment the contents of the address register by one at a time is provided.

The following paragraphs (1) through (5) describe the operation of the address register when each instruction is executed.

(1) Table reference instruction (“MOVT DBF, @AR”)

This instruction reads the constant data (16 bits) of the program memory address specified by the contents of the address register to the data buffer.

The constant data storing address that can be specified by the address register is 0000H to 1EFBH.

(2) Stack manipulation instruction (“PUSH AR”, “POP AR”)

When the “PUSH AR” instruction is executed, the contents of the stack pointer are decremented by one, and the contents of the address register (AR) are stored to the address stack register specified by the stack pointer.

When the “POP AR” instruction is executed, the contents of the address stack register specified by the stack pointer are transferred to the address register, and the contents of the stack pointer are incremented by one.

(3) Indirect branch instruction (“BR @AR”)

This instruction branches execution to the program memory address specified by the contents of the address register.

The branch address that can be specified by the address register is 0000H to 1EFBH.

(4) Indirect subroutine call instruction (“CALL @AR”)

This instruction calls the subroutine at the program memory address specified by the contents of the address register.

The first address of the subroutine that is specified by the address register is 0000H to 1EFBH.

(5) Address register increment instruction (“INC AR”)

This instruction increments the contents of the address register by one.

Because the address register consists of 13 bits, if “INC AR” instruction is executed when the contents of the address register are “1FFFH”, the address register contents are cleared to “0000H”.

5.3.3 Address register and data buffer

The address register can transfer data via data buffer as a part of the peripheral hardware.

For details, refer to **9. DATA BUFFER (DBF)**.

5.3.4 Notes on using address register

Because the address register consists of 13 bits, its contents can be up to 1FFFH theoretically.

However, the highest address of the program memory is 1EFBH.

Therefore, the maximum value that can be set to the address register is 1EFBH.

5.4 Window Register (WR)

5.4.1 Configuration of window register

Figure 5-4 shows the configuration of the window register.

As shown in this figure, the window register consists of 4 bits of address 78H of the system register.

Figure 5-4. Window Register Configuration

Address		78H			
Name		Window Register (WR)			
Symbol		WR			
Bit		b_3	b_2	b_1	b_0
Data		^ MSB v			^ LSB v
On reset	Power-ON	Undefined			
	Clock stop	Retains previous status			
	CE				

5.4.2 Function of window register

The window register transfers data with the register file (RF) described later.

To transfer data between the window register and register file, dedicated instructions “PEEK WR, rf” and “POKE rf, WR” are used (where rf is the address of the register file).

The following paragraphs (1) and (2) describe the operation to be performed when each instruction is executed.

For more information, refer to **8. REGISTER FILE (RF)**.

(1) “PEEK WR, rf” instruction

This instruction transfers the contents of the register file addressed by “rf” to the window register.

(2) “POKE rf, WR” instruction

This instruction transfers the contents of the window register to the register file addressed by “rf”.

5.5 Bank Register (BANK)

5.5.1 Configuration of bank register

Figure 5-5 shows the configuration of the bank register.

As shown in this figure, the bank register consists of 4 bits of address 79H (BANK) of the system register.

Actually, however, the bank register operates as a 2-bit register because its highest 2 bits are always fixed to “0”.

Figure 5-5. Bank Register Configuration

Address		79H			
Name		Bank Register (BANK)			
Symbol		BANK			
Bit		b_3	b_2	b_1	b_0
Data		0	0	^MSB v	^LSB v
On reset	Power-ON	0			
	Clock stop	0			
	CE	0			

5.5.2 Function of bank register

The bank register specifies a bank of the data memory.

Table 5-1 shows the relation between the value of the bank register and the bank of the data memory.

Because the bank register exists on the system register, its contents can be rewritten regardless of the bank currently specified.

Therefore, the bank register can be manipulated independently of the current bank status.

Table 5-1. Specifying Bank of Data Memory

Bank Register (BANK)				Bank of Data Memory
b_3	b_2	b_1	b_0	
0	0	0	0	BANK0
0	0	0	1	BANK1
0	0	1	0	BANK2
0	0	1	1	BANK3

5.6 Index Register (IX) and Data Memory Row Address Pointer (MP: Memory Pointer)

5.6.1 Configuration of index register and data memory row address pointer

Figure 5-6 shows the configuration of the index register and data memory row address pointer.

As shown in this figure, the index register consists of an index register (IX) made up of a total of 11 bits (the lower 3 bits (IXH) of the address 7AH, and addresses 7BH and 7CH (IXM and IXL) of the system register) and an index enable flag (IXE) that is the least significant bit of address 7FH (PSW).

The data memory row address pointer (memory pointer) consists of a data memory row address pointer made up of a total of 7 bits (the lower 3 bits of 7AH (MPH) and 7BH (MPL)) and a data memory row address pointer enable flag (memory pointer enable flag: MPE) that is the most significant bit of 7AH (MPH).

This means that the higher 7 bits of the index register are shared by the data memory row address pointer.

However, the highest 2 bits of the index register and data memory row address pointer (bits b_2 and b_1 of 7AH) are always fixed to “0”.

Figure 5-6. Configuration of Index Register and Data Memory Row Address Pointer

Address		7AH				7BH				7CH				7EH				7FH			
Name		Index Register (IX) / Memory Pointer (MP)																Program Status Word (PSWORD)			
Symbol		IXH / MPH				IXM / MPL				IXL								PSW			
Bit		b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data		MPE	0	0	^MSB v				IX ^LSB v												IXE
					^MSB v				^LSB v												
							MP														
On reset	Power-ON	0				0				0											0
	Clock stop	0				0				0											0
	CE	0				0				0											0

5.6.2 Function of index register and data memory row address pointer

The index register and data memory row address pointer modify the addresses of the data memory.

The following paragraphs (1) and (2) describe the functions of the index register and data memory row address pointer.

A dedicated instruction ("INC IX") that can increment the contents of the index register by one at a time is provided.

For the details of address modification, refer to **7. ALU (ARITHMETIC LOGIC UNIT) BLOCK**.

(1) Index register

A data memory address is modified according to the contents of the index register when a data memory manipulation instruction is executed.

However, this modification is valid only when the IXE flag is set to "1".

The address is modified by ORing the bank, row address, and column address of the data memory with the contents of the index register, and an instruction is executed to the data memory specified by the result of the OR operation (called actual address).

Address modification by the index register is subjected to all data memory manipulation instructions.

The following instructions are not subject to modification.

INC	AR	RORC	r
INC	IX	CALL	addr
MOVT	DBF, @AR	CALL	@AR
PUSH	AR	RET	
POP	AR	RETSK	
PEEK	WR, rf	RETI	
POKE	rf, WR	EI	
GET	DBF, p	DI	
PUT	p, DBF	STOP	s
BR	addr	HALT	h
BR	@AR	NOP	

(2) Data memory row address pointer

When a general register indirect transfer instruction ("MOV @r, m", "MOV m, @r") is executed, the address of the indirect transfer destination is modified.

This modification, however, is valid only when the MPE flag is set to "1".

To modify the address, the bank and row address at the indirect transfer destination are replaced with the contents of the data memory row address pointer.

Instructions other than the general register indirect transfer instruction is not subject to address modification.

(3) Index register increment instruction ("INC IX")

This instruction increments the contents of the index register by one at a time.

Because the index register consists of 9 bits, if the "INC IX" instruction is executed when the contents of the index register are "1FFH", the index register is cleared to "000H".

5.7 General Register Pointer (RP)

5.7.1 Configuration of general register pointer

Figure 5-7 shows the configuration of the general register pointer.

As shown in this figure, the general register pointer consists of a total of 7 bits: 4 bits of the address 7DH (RPH) of the system register and the higher 3 bits of address 7EH (RPL). Actually, however, only the lower 5 bits (the lower 2 bits of address 7DH and the higher 3 bits of address 7EH) are valid because the higher 2 bits of address 7DH are always fixed to 0.

Figure 5-7. General Register Pointer Configuration

Address		7DH				7EH			
Name		General Register Pointer (RP)							
Symbol		RPH				RPL			
Bit		b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data		0	0	^ MSB v				^ LSB v	BCD
On reset	Power-ON	0				0			
	Clock stop	0				0			
	CE	0				0			

5.7.2 Function of general register pointer

The general register pointer specifies a general register on the data memory.

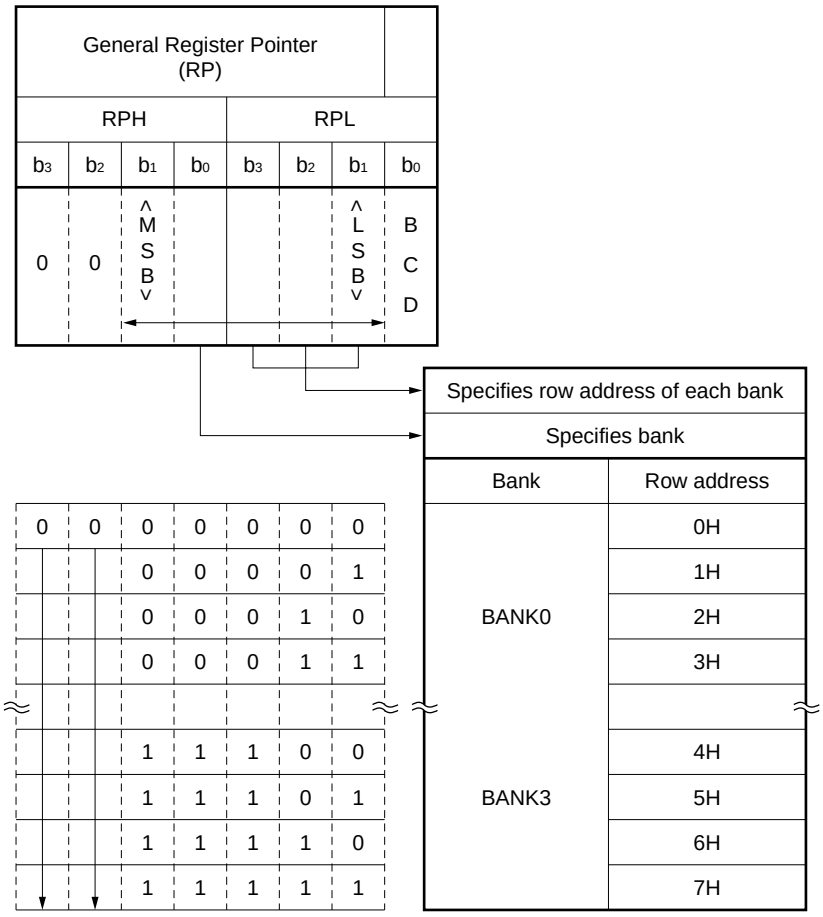
Figure 5-8 shows the addresses of the general register specified by the general register pointer.

As shown in this figure, the higher 4 bits of the general register pointer (RPH: address 7DH) specify a bank, and the lower 3 bits (RPL: address 7EH) specify a row address.

Because the number of valid bits of the general register pointer is 5, the row addresses (0H through 7H) of all the banks (BANK0 through BANK3) can be specified as a general register.

For the details of the operation of the general register, refer to **6. GENERAL REGISTER (GR)**.

Figure 5-8. Address of General Register Specified by General Register Pointer



5.7.3 Notes on using general register pointer

The least significant bit of address 7EH (RPL) of the general register pointer is allocated as the BCD flag of the program status word.

When rewriting RPL, therefore, pay attention to the value of the BCD flag.

5.8 Program Status Word (PSWORD)

5.8.1 Configuration of program status word

Figure 5-9 shows the configuration of the program status word.

As shown in this figure, the program status word consists of a total of 5 bits: the least significant bit of address 7EH (RPL) of the system register and 4 bits of address 7FH (PSW).

Each bit of the program status word has its own function, and the program status word consists of BCD flag (BCD), compare flag (CMP), carry flag (CY), zero flag (Z), and index enable flag (IXE).

Figure 5-9. Program Status Word Configuration

Address		7EH				7EH			
Name		(RP)				Program Status Word (PSWORD)			
Symbol		RPL				PSW			
Bit		b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data					BCD	CMP	CY	Z	IXE
On reset	Power-ON	0				0			
	Clock stop	0				0			
	CE	0				0			

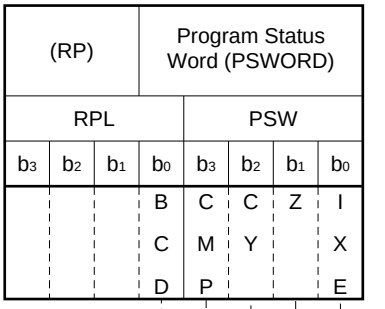
5.8.2 Function of program status word

The program status word is a register that sets the condition for the operation of the ALU (Arithmetic Logic Unit) or transfer instruction or indicates the result of an operation.

Table 5-2 outlines the function of each flag of the program status word.

For details of the operation, refer to **7. ALU (ARITHMETIC LOGIC UNIT) BLOCK**.

Table 5-2. Functional Outline on Each Flag of Program Status Word



Flag Name	Function
Index enable flag (IXE)	Modifies address of data memory when data memory manipulation instruction is executed 0: Does not modify 1: Modifies
Zero flag (Z)	Indicates that resultof arithmeticoperation is zero. Note that status of 0 and 1 of this flag differs depending on contents of compare flag.
Carry flag (CY)	Indicates occurrence of carry or borrow as result of executing addition or subtraction instruction. Reset to 0 if carry or borrow does not occur. Set to 1 if carry or borrow occurs. This flag is also used as shift bit of "RORC r" instruction.
Compare flag (CMP)	This flag specifies whether or not resultof arithmetic operation isstored to data memory or general regizster. 0: Stores result 1: Does not store result
BCD flag (BCD)	This flag specifies whether arithmetic operation is performed in binary or decimal 1: Binary operation 0: Decimal operation

5.8.3 Notes on using program status word

If an arithmetic operation (addition or subtraction) instruction is executed to the program status word, the result of the arithmetic operation is stored to the program status word.

For example, when an operation that generates a carry is executed and if the result of the operation is 0000B, 0000B is stored to PSW.

5.9 Notes on Using System Register

Data of the system register that is fixed to “0” is not affected in any way even if a write instruction is executed to it.

If this data is read, “0” is always read.

6. GENERAL REGISTER (GR)

6.1 Outline of General Register

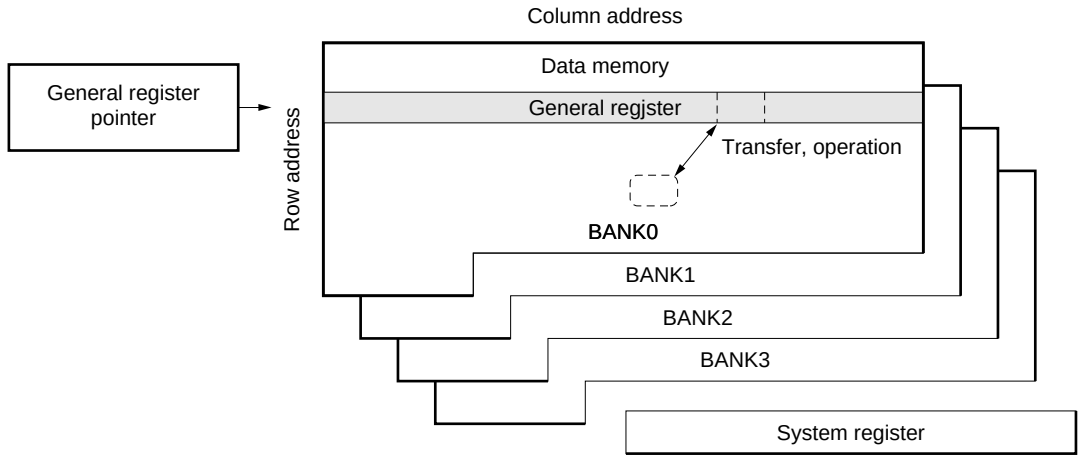
Figure 6-1 outlines the general register.

As shown in this figure, the general register consists of a general register pointer and a general register.

The bank and row address of the general register are specified by the general register pointer.

The general register is used to transfer data or execute operations between data memory areas.

Figure 6-1. Outline of General Register



6.2 General Register

The general register consists of 16 nibbles (16 × 4 bits) which are at the same row addresses on the data memory.

For the ranges of banks and row addresses that can be specified for the general register pointer and general register, refer to **5.7 General Register Pointer (RP)**.

The 16 nibbles at the same row addresses specified as the general register can execute operations and transfer data with a data memory area with a single instruction.

This means that operation and data transfer between data memory areas can be executed with a single instruction.

The general register can be controlled by using data memory manipulation instructions in the same manner as the other data memory areas.

6.3 General Register Address Generation by Each Instruction

6.3.1 and **6.3.2** describe how the addresses of the general register are generated by each instruction.

For the details of the operation of each instruction, refer to **7. ALU (ARITHMETIC LOGIC UNIT) BLOCK**.

6.3.1 Addition (“ADD r, m”, “ADDC r, m”), Subtraction (“SUB r, m”, “SUBC r, m”), Logical operation (“AND r, m”, “OR r, m”, “XOR r, m”), Direct transfer (“LD r, m”, “ST m, r”), Rotation processing (“RORC r”) instruction

Table 6-1 shows the address of general register “R” specified by the operand “r” of an instruction. Instruction operand “r” only specifies a column address.

Table 6-1. General Register Address Generation

	Bank				Row Address			Column Address			
	b_3	b_2	b_1	b_0	b_2	b_1	b_0	b_3	b_2	b_1	b_0
General register address R	Contents of general register pointer							r			

6.3.2 Indirect transfer (“MOV @r, m”, “MOV m, @r”) instructions

Table 6-2 shows the address of the general register “R” specified by instruction operand “r” and indirect transfer address specified by “@R”.

Table 6-2. General Register Address Generation

	Bank				Row Address			Column Address			
	b_3	b_2	b_1	b_0	b_2	b_1	b_0	b_3	b_2	b_1	b_0
General register address R	Contents of general register pointer							r			
Indirect transfer address @R	Same as data memory							Contents of R			

6.4 Notes on Using General Register

6.4.1 Row address of general register

Because the row address of the general register is specified by the general register pointer, the currently specified bank may be different from the bank of the general register.

6.4.2 Operation between general register and immediate data

There is no instruction povided to execute an operation between the general register and immediate data.

To execute an operation instruction between the general register and immediate data, the general register must be treated as a data memory area.

7. ALU (ARITHMETIC LOGIC UNIT) BLOCK

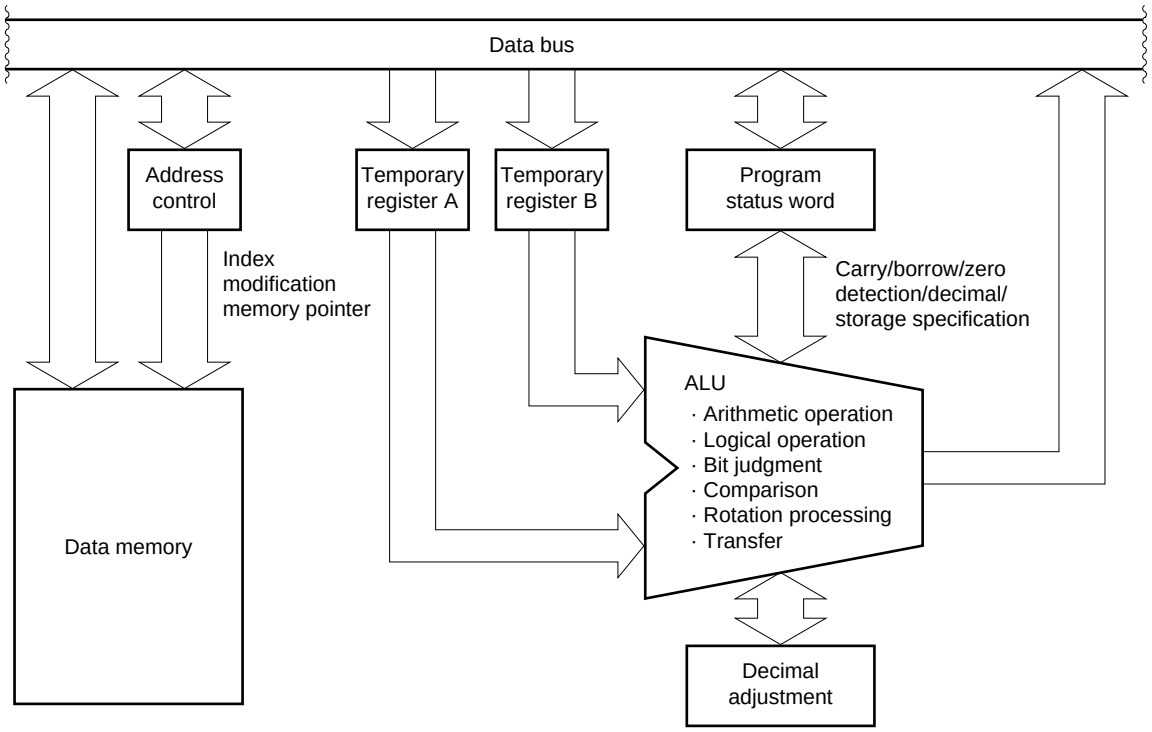
7.1 Outline of ALU Block

Figure 7-1 outlines the ALU block.

As shown in this figure, the ALU block consists of an ALU, temporary registers A and B, program status word, decimal adjustment circuit, and data memory address control circuit.

The ALU executes operation, judgment, comparison, rotation, and transfer of 4-bit data on the data memory.

Figure 7-1. Outline of ALU Block



7.2 Configuration and Function of Each Block

7.2.1 ALU

The ALU executes arithmetic operation, logical operation, bit judgment, comparison, rotation processing, and transfer of 4-bit data by using an instruction specified by the program.

7.2.2 Temporary registers A and B

Temporary registers A and B temporarily store 4-bit data.

These registers are automatically used when an instruction is executed, and cannot be controlled by program.

7.2.3 Program status word

The program status word controls the operation and stores the status of the ALU.

For the details of the program status word, refer to **5.8 Program Status Word (PSWORD)**.

7.2.4 Decimal adjustment circuit

The decimal adjustment circuit converts the result of an arithmetic operation into a decimal number if the BCD flag of the program status word is set to “1” when the arithmetic operation is executed.

7.2.5 Address control circuit

The address control circuit specifies an address of the data memory.

At this time, it also controls address modification by the index register and data memory row address pointer.

7.3 ALU Processing Instruction List

Table 7-1 lists the operations of the ALU when each instruction is executed.

Table 7-2 shows modification of data memory addresses by the index register and data memory row address pointer.

Table 7-3 shows the decimal adjustment data when a decimal operation is performed.

Table 7-1. ALU Processing Instruction Operation List

ALU Function	Instruction		Difference in Operation Depending on Program Status Word (PSWORD)					Address Modification	
			Value of BCD Flag	Value of CMP Flag	Operation	Operation of CY Flag	Operation of Z Flag	Index	Memory Pointer
Addition	ADD	r, m	0	0	Stores result of binary operation	Set if carry or borrow occurs; otherwise, reset	Set if result of operation is 0000B; otherwise, reset	Modified	Not modified
		m, #n4							
	ADDC	r, m	0	1	Does not store result of binary operation		Retains status if result of operation is 0000B; otherwise, reset		
		m, #n4							
Sub-traction	SUB	r, m	1	0	Stores result of decimal operation		Set if result of operations is 0000B; otherwise, reset		
		m, #n4							
	SUBC	r, m	1	1	Does not store result of decimal operation		Retains status if result of operation is 0000B; otherwise, reset		
		m, #n4							
Logical operation	OR	r, m	Any (retained)	Any (retained)	Not affected	Retains previous status	Retains previous status	Modified	Not modified
		m, #n4							
	AND	r, m							
		m, #n4							
	XOR	r, m							
		m, #n4							
Judgment	SKT	m, #n	Any (retained)	Any (reset)	Not affected	Retains previous status	Retains previous status	Modified	Not modified
	SKF	m, #n							
Compar-ison	SKE	m, #n4	Any (retained)	Any (retained)	Not affected	Retains previous status	Retains previous status	Modified	Not modified
	SKNE	m, #n4							
	SKGE	m, #n4							
	SKLT	m, #n4							
Transfer	LD	r, m	Any (retained)	Any (retained)	Not affected	Retains previous status	Retains previous status	Modified	Not modified
	ST	m, r							
	MOV	m, #n4							
		@r, m							Modified
		m, @r							
Rotation	RORC	r	Any (retained)	Any (retained)	Not affected	Value of b_0 of general register	Retains previous status	Not modified	Not modified

Table 7-2. Modification of Data Memory Address and Modification of Indirect Transfer Address by Index Register and Data Memory Row Address Pointer

IXE	MPE	General Register Address Specified by r			Data Memory Address Specified by m			Indirect Transfer Address Specified by @r		
		Bank	Row Address	Column Address	Bank	Row Address	Column Address	Bank	Row Address	Column Address
		b_3 b_2 b_1 b_0	b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_2 b_1 b_0	b_3 b_2 b_1 b_0
0	0	RP		r	BANK		m	BANK	m_R	(r)
0	1		ditto			ditto		MP		(r)
1	0		ditto		BANK Logical IX		m OR	BANK Logical IXH, IXM	m_R OR	(r)
1	1		ditto			ditto		MP		(r)

BANK : bank register
IX : index register
IXE : index enable flag
IXH : bits 10 through 8 of index register
IXM : bits 7 through 4 of index register
IXL : bits 3 through 0 of index register
m : data memory address indicated by m_R, m_C
m_R : data memory row address (high)
m_C : data memory column address (low)
MP : data memory row address pointer
MPE : memory pointer enable flag
r : general register column address
RP : general register pointer
(×) : contents addressed by ×
× : direct address such as m and r

Table 7-3. Decimal Adjustment Data

Operation Result	Hexadecimal Addition		Decimal Addition	
	CY	Operation Result	CY	Operation Result
0	0	0000B	0	0000B
1	0	0001B	0	0001B
2	0	0010B	0	0010B
3	0	0011B	0	0011B
4	0	0100B	0	0100B
5	0	0101B	0	0101B
6	0	0110B	0	0110B
7	0	0111B	0	0111B
8	0	1000B	0	1000B
9	0	1001B	0	1001B
10	0	1010B	1	0000B
11	0	1011B	1	0001B
12	0	1100B	1	0010B
13	0	1101B	1	0011B
14	0	1110B	1	0100B
15	0	1111B	1	0101B
16	1	0000B	1	0110B
17	1	0001B	1	0111B
18	1	0010B	1	1000B
19	1	0011B	1	1001B
20	1	0100B	1	1110B
21	1	0101B	1	1111B
22	1	0110B	1	1100B
23	1	0111B	1	1101B
24	1	1000B	1	1110B
25	1	1001B	1	1111B
26	1	1010B	1	1100B
27	1	1011B	1	1101B
28	1	1100B	1	1010B
29	1	1101B	1	1011B
30	1	1110B	1	1100B
31	1	1111B	1	1101B

Operation Result	Hexadecimal Addition		Decimal Addition	
	CY	Operation Result	CY	Operation Result
0	0	0000B	0	0000B
1	0	0001B	0	0001B
2	0	0010B	0	0010B
3	0	0011B	0	0011B
4	0	0100B	0	0100B
5	0	0101B	0	0101B
6	0	0110B	0	0110B
7	0	0111B	0	0111B
8	0	1000B	0	1000B
9	0	1001B	0	1001B
10	0	1010B	1	1100B
11	0	1011B	1	1101B
12	0	1100B	1	1110B
13	0	1101B	1	1111B
14	0	1110B	1	1100B
15	0	1111B	1	1101B
–16	1	0000B	1	1110B
–15	1	0001B	1	1111B
–14	1	0010B	1	1100B
–13	1	0011B	1	1101B
–12	1	0100B	1	1110B
–11	1	0101B	1	1111B
–10	1	0110B	1	0000B
–9	1	0111B	1	0001B
–8	1	1000B	1	0010B
–7	1	1001B	1	0011B
–6	1	1010B	1	0100B
–5	1	1011B	1	0101B
–4	1	1100B	1	0110B
–3	1	1101B	1	0111B
–2	1	1110B	1	1000B
–1	1	1111B	1	1001B

Remark Decimal adjustment is not carried out correctly in the portion ▭ in the above table.

7.4 Notes on Using ALU

7.4.1 Notes on operation to program status word

When an arithmetic operation is executed to the program status word, the result of the operation is stored to the program status word.

The CY and Z flags of the program status word are normally set or reset depending on the result of an arithmetic operation. If an arithmetic operation is executed to the program status word itself, however, the result of the operation is stored to the program status word, and occurrence of a carry or borrow, and whether the result of the operation is zero cannot be identified.

If the CMP flag is set, the result of the operation is not stored to the program status word, and therefore, the CY and Z flags are set or reset as usual.

7.4.2 Notes on using decimal operation

A decimal operation can be executed as long as the result falls within the following ranges:

(1) Result of addition must be 0 to 19 in decimal.
(2) Result of subtraction must be 0 to 9 or –10 to –1 in decimal.

If these ranges are exceeded in decimal operation, the CY flag is set, and the result of the operation is greater than 1010B (0AH).

8. REGISTER FILE (RF)

8.1 Outline of Register File

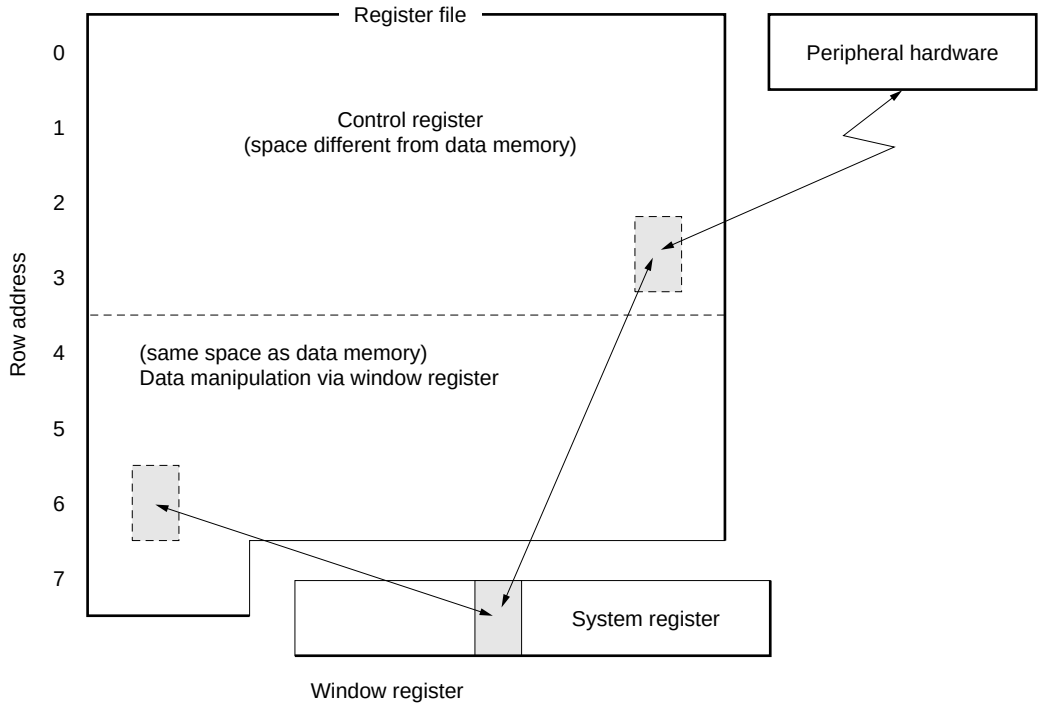
Figure 8-1 outlines the register file.

As shown in this figure, the register file consists of a control register that exists on a space different from the data memory, and a portion overlapping the data memory.

The control register sets the conditions of the peripheral hardware.

The data on the register file is read or data is written to the register file via window register.

Figure 8-1. Outline of Register File



8.2 Configuration and Function of Register File

Figure 8-2 shows the configuration of the register file and the relation between the data memory and register file.

The register file is allocated addresses in 4-bit units in the same manner as the data memory, and has a total of 128 nibbles with row addresses 0H through 7H and column addresses 0H through 0FH.

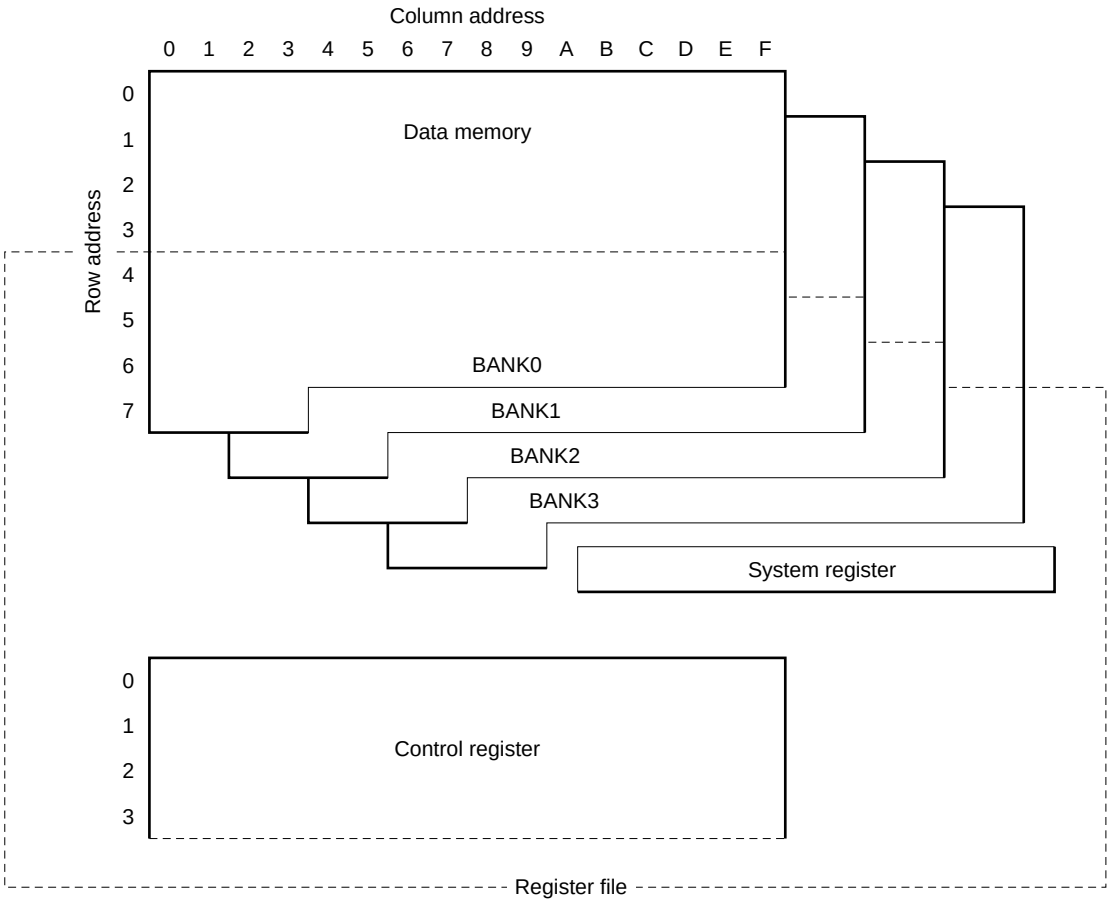
An area consisting of addresses 00H through 3FH is called a control register. This register sets the condition of the peripheral hardware.

Addresses 40H through 7FH overlap the data memory.

This means that the memory addresses 40H through 7FH of the bank of the data memory selected at that time exist at the addresses 40H through 7FH of the register file.

Therefore, because addresses 40H through 7FH overlap the data memory, they can be treated in the same manner as an data memory area, except that they can be manipulated by using a register file manipulation instruction “PEEK WR, rf” or “POKE rf, WR”).

Figure 8-2. Configuration of Register File and Relation with Data Memory



8.2.1 Register file manipulation instructions (“PEEK WR, rf”, “POKE rf, WR”)

Data is read from or written to the register file via the window register of the system register, by using the following instructions:

(1) “PEEK WR, rf”

This instruction reads data from the register file addressed by “rf” to the window register.

(2) “POKE rf, WR”

This instruction writes data of the window register to the register file addressed by “rf”.

8.3 Control Registers

Figure 8-3 shows the configuration of the control registers.

As shown in this figure, the control registers consist of a total of 64 nibbles (64 words × 4 bits) of addresses 00H through 3FH of the register file.

Of these 64 nibbles, however, only 41 nibbles are actually used, and the remaining 23 nibbles are unused registers which are prohibited from reading or writing.

Each nibble of a control register has an attribute which may be read/write (R/W), read-only (R), write-only (W), or reset on read (R & Reset).

Nothing is affected even if data is written to read-only (R and R & Reset) registers.

If a write-only register (W) is read, an “undefined” value is read.

The bits fixed to “0” of the 4-bit data in 1 nibble are always “0” when they are read, and retain “0” even when data is written to them.

If an attempt is made to read the contents of the unused 23 nibbles, an undefined value is read. Nothing is changed even if data is written to these nibbles.

Figure 8-3. Configuration of Control Register (1/2)

Column Address / Row Address	Item	0	1	2	3	4	5	6	7
0 (8)[Note]	Name		Stack pointer SP	Serial I/O1 mode select (SIO1MODE)		IF counter open status judge (IFCGOSTR)	PLL unlock FF judge (PLLULJDG)	A/D converter compare judge (ADCJDG)	CE pin level judge (CEJDG)
	Symbol		^SP3v, ^SP2v, ^SP1v, ^SP0v	SIO1TS, SIO1HIZ, SIO1CK1, SIO1CK0		0, 0, 0, IFCGOSTT	0, 0, 0, PLLUL	0, 0, 0, ADCCMP	0, 0, 0, CE
	Read/ Write		R/W	R/W		R	R & Reset	R	R
1 (9)[Note]	Name	LCD mode select (LCDMODE)	LCD port select (LCDPORT)	IF counter mode select (IFCMODE)	PWM mode select (PWMMODE)	A/D converter channel select (ADCCH)	PLL unlock FF sensibility select (PLLULSEN)	Key input judge (KEYJDG)	Basic timer 0 carry FF judge (BTM0CYJG)
	Symbol	0, 0, KSEN, LCDEN	P0YSEL, P0XSEL, P0ESEL, P0FSEL	IFCMD1, IFCMD0, IFCCK1, IFCCK0	PWM2SEL, PWM1SEL, PWM0SEL, CGPSEL	ADCCH3, ADCCH2, ADCCH1, ADCCH0	PLULSEN3, PLULSEN2, PLULSEN1, PLULSEN0	0, 0, 0, KEYJ	0, 0, 0, BTM0CY
	Read/ Write	R/W	R/W	R/W	R/W	R/W	R/W	R & Reset	R & Reset
2 (A)[Note]	Name		PLL mode select (PLLMODE)		IF counter control (IFCCONT)				Port 0C group I/O select (P0CGPIO)
	Symbol		PLLMD3, PLLMD2, PLLMD1, PLLMD0		0, 0, IFCSTRT, IFCRES				0, 0, 0, P0CGIO
	Read/ Write		R/W		R/W				R/W
3 (B)[Note]	Name		PLL reference clock select (PLLRFCLK)				Port 1A bit I/O select (P1ABIO)	Port 0B bit I/O select (P0BBIO)	Port 0A bit I/O select (P0ABIO)
	Symbol		PLLRFCK3, PLLRFCK2, PLLRFCK1, PLLRFCK0				P1ABIO3, P1ABIO2, P1ABIO1, P1ABIO0	P0BBIO3, P0BBIO2, P0BBIO1, P0BBIO0	P0ABIO3, P0ABIO2, P0ABIO1, P0ABIO0
	Read/ Write		R/W				R/W	R/W	R/W

Note ( ) indicates the address when the assembler (AS17K) is used.

Figure 8-3. Configuration of Control Register (2/2)

	8	9	A	B	C	D	E	F
	Serial I/O0 mode select (SIO0MODE)	Basic timer clock select (BTMCLK)			Timer counter clock select (TMCLK)	Timer counter overflow detect (TMOVDET)	12-bit timer mode control (TMMDCONT)	Interrupt group select (IGRPSELR)
3	SIO0CH	BTM1CK1			TMCK3	0	0	TMRPT	0	0	IGRPSL
2	SB	BTM1CK0			TMCK2	0	0	TMRES	0	0	
1	SIO0MS	BTM0CK1			TMCK1	0	0	TMEN	0	0	
0	SIO0TX	BTM0CK0			TMCK0	TMOVF					
	R/W	R/W			R/W	R	R/W	R/W
	Serial I/O0 wait control (SIO0WT)	Serial I/O0 wait status judge (SIO0WSTR)						Interrupt edge select (INTEDGE)
3	SBACK	0	0	0					0	0	IEG1	IEG0
2	SIO0NWT	0	0	0								
1	SIO0WRQ1	0	0	0								
0	SIO0WRQ0	SIO0WSTT										
	R/W	R						R/W
	Serial I/O0 status judge (SIO0STUS)						Interrupt permission 1 (INTPM1)	Interrupt permission 2 (INTPM2)
3	SIO0SF8	SBSTT							0	0	IPBTM1	IPTM	IPGRP	IP0
2	SIO0SF9	SBBSY							0	0	IPIFC	IPSIO0				
1														
0														
	R/W						R/W	R/W
	Serial I/O0 interrupt mode (SIO0INT)	Serial I/O0 clock select (SIO0CLK)	IF counter interrupt request (IREQIFC)	Serial I/O0 interrupt request (IREQSIO0)	Basic timer 1 interrupt request (IREQBTM1)	12-bit timer interrupt request (IREQTM)	Group interrupt request (IREQGRP)	INT0 interrupt request (IREQINT)
3	SIO0IMD3	SIO0CK3	0	0	0	IRQIFC	0	0	0	IRQSIO0	0	0	0	IRQBTM1	0	0	0	IRQTM	INT1	0	0	IRQGRP	INT0	0	0	IRQ0
2	SIO0IMD2	SIO0CK2																								
1	SIO0IMD1	SIO0CK1																								
0	SIO0IMD0	SIO0CK0																								
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-1. Outline of Peripheral Hardware Control Function of Control Register (1/5)

Peripheral Hardware	Control Register: Name	Address	Read/Write	b3 b2 Symbol b1 b0	Peripheral Hardware Control Function: Functional Outline	Set Value	On Reset: Power-ON	Clock Stop	CE
Stack	Stack pointer SP	01H	R/W	(SP3)	Stack pointer		7	7	7
				(SP2)					
				(SP1)					
				(SP0)					
Timer	Basic timer clock select (BTMCLK)	09H	R/W	BTM1CK1	Sets basic timer 1 interrupt time	0 0 1 1 100 ms 250 ms 5 ms 1 ms 0 1 0 1	0	0	Retained
				BTM1CK0					
				BTM0CK1	Sets basic timer 0 carry FF time	0 0 1 1 100 ms 250 ms 5 ms 1 ms 0 1 0 1			
				BTM0CK0					
	Timer counter clock select (TMCLK)	0CH	R/W	TMCK3	0				
				TMCK2	0				
				TMCK1	Sets clock of 12-bit timer	0 0 1 1 1 kHz 3 kHz 100 kHz 90 kHz 0 1 0 1	0	0	Retained
				TMCK0					
	Timer counter overflow detect (TMOVDET)	0DH	R	0					
				0					
				0					
				TMOVF	Detects overflow of timer counter	Set on occurrence of overflow. Reset by reset signal.	0	0	Retained
	12-bit timer mode control (TMMDCONT)	0EH	R/W	0					
				TMRPT	Sets operation mode of 12-bit timer.	0: Free-run count mode 1: Modulo count mode	0	0	Retained
				TMRES	Resets data of timer/counter.	0: NOP instruction 1: Reset			
				TMEN	Sets operation of timer/counter.	0: Does not operate 1: Operates			
	Basic timer 0 carry FF judge (BTM0CYJG)	17H	Read & Reset	0					
				0					
				0					
				BTM0CY	Detects basic timer 0 carry FF	0: Resets FF 1: Sets FF	0	1	1
Pin	CE pin level judge (CEJDG)	07H	R	0					
				0					
				0					
				CE	Detects status of CE pin	0: Low level 1: high level	—	—	—

Table 8-1. Outline of Peripheral Hardware Control Function of Control Register (2/5)

Peripheral Hardware	Control Register				Peripheral Hardware Control Function		On Reset		
	Name	Address	Read/ Write	b3 b2 Symbol b1 b0	Functional Outline	Set Value	Power-ON	Clock Stop	CE
Interrupt	Interrupt group select (IGRPSELR)	0FH	R/W	0 0 0 IGRPSL	Sets set condition of IRQGRP	0: Edge of INT_1 pin 1: Timer overflow	0	0	0
	Interrupt edge select (INTEDGE)	1FH	R/W	0 0 IEG1 IEG2	Sets interrupt issuance edge of INT_1 and INT_0 pins	0: Rising edge 1: Falling edge	0	0	0
	Interrupt permission 1 (INTPM1)	2EH	R/W	0 0 IPIFC IPSIO0	Enables IF counter, serial interface 0, basic timer 1, 12-bit timer, INT_1 pin or overflow of timer/counter, and INT_0 pin interrupts	0: Disabled 1: Enabled	0	0	0
	Interrupt permission 2 (INTPM2)	2FH	R/W	IPBTM1 IPTM IRGRP IP0					
	IF counter interrupt request (IREQIFC)	3AH	R/W	0 0 0 IRQIFC	Detects interrupt request of IF counter	0: No request 1: Request	0	0	0
	Serial I/O0 interrupt request (IREQSIO0)	3BH	R/W	0 0 0 IRQSIO0	Detects interrupt request of serial interface 0	0: No request 1: Request	0	0	0
	Basic timer 1 interrupt request (IREQBTM1)	3CH	R/W	0 0 0 IRQBTM1	Detects interrupt request of basic timer 1	0: No request 1: Request	0	0	0
	12-bit timer interrupt request (IREQTM)	3DH	R/W	0 0 0 IRQTM	Detects interrupt request of 12-bit timer	0: No request 1: Request	0	0	0
	Group interrupt request (IREQGRP)	3EH	R/W	INT1 0 0 IRQGRP	Detects status of INT_1 pin Detects INT_1 pin or timer/counter overflow interrupt request	0: Low level 1: High level 0: No request 1: Request	0	0	0
	INT0 interrupt request (IREQINT0)	3FH	R/W	INT0 0 0 IRQ0	Detects status of INT_0 pin Detects interrupt request of INT_0 pin	0: Low level 1: High level 0: No request 1: Request	0	0	0

Table 8-1. Outline of Peripheral Hardware Control Function of Control Register (3/5)

Peripheral Hardware	Control Register: Name	Address	Read/Write	b3 b2 Symbol b1 b0	Peripheral Hardware Control Function: Functional Outline	Set Value	On Reset: Power-ON	Clock Stop	CE
PLL frequency synthesizer	PLL unlock FF judge (PLLULJDG)	05H	Read & Reset	0 0 0 PLLUL	Detects status of unlock FF	0: Locked 1: Unlocked	Undefined	Retained	Retained
	PLL unlock FF sensibility select (PLLULSEN)	15H	R/W	PLULSEN3 PLULSEN2 PLULSEN1 PLULSEN0	0 0 Sets set delay time of unlock FF	0 0 1 1 1 μs 2 μs 0.5 μs Disable 0 1 0 1	0	0	Retained
	PLL mode select (PLLMODE)	21H	R/W	PLLMD3 PLLMD2 PLLMD1 PLLMD0	0 0 Sets division method of PLL	0 0 1 1 Disable MF VHF HF 0 1 0 1	0	0	Retained
	PLL reference clock select (PLLRFCLK)	31H	R/W	PLLRFCK3 PLLRFCK2 PLLRFCK1 PLLRFCK0	Sets reference frequency of PLL	0:1.25 1:2.5 2:5 3:10 4:6.25 5:12.5 6:25 7:50 8:3 9:A:B:Setting prohibited C:1 D:9 E:100 F:Off	F	F	Retained
A/D converter	A/D converter channel select (ADCCH)	14H	R/W	ADCCH3 ADCCH2 ADCCH1 ADCCH0	0 Selects pin used as A/D converter	0:AD_0 1:AD_1 2:AD_2 3:AD_3 4:AD_4 5:AD_5 6:7:Input port	7	7	7
	A/D converter compare judge (ADCJDG)	06H	R	0 0 0 ADCCMP	Detects comparison result of A/D converter	0:V_{REF}>V_{ADCIN} 1: V_{REF}<V_{ADCIN}	Undefined	Retained	Retained
General-purpose port	Port 0C group I/O select (P0CGPIO)	27H	R/W	0 0 0 P0CGIO	Sets I/O mode of $P0C_3$-$P0C_0$ pins (in 4-bit units)	0: Input 1: Output	0	0	0
	Port 1A bit I/O select (P1ABIO)	35H	R/W	P1ABIO3 P1ABIO2 P1ABIO1 P1ABIO0	Sets I/O mode of $P1A_3$, $P1A_2$, $P1A_1$, $P1A_0$, $P0B_3$, $P0B_2$, $P0B_1$, $P0B_0$, $P0A_3$, $P0A_2$, $P0A_1$, and $P0A_0$ pins (in 1-bit units)	0: Input 1: Output	0	0	0
	Port 0B bit I/O select (P0BBIO)	36H	R/W	P0BBIO3 P0BBIO2 P0BBIO1 P0BBIO0					
	Port 0A bit I/O select (P0ABIO)	37H	R/W	P0ABIO3 P0ABIO2 P0ABIO1 P0ABIO0					
D/A converter	PWM mode select (PWMMODE)	13H	R/W	PWM2SEL PWM1SEL PWM0SEL CGPSEL	Sets PWM_2, PWM_1, and PWM_0 pins as D/A converter. Sets CGP pin as CGP	0: General-purpose output port 1: D/A converter 0: General-purpose output port 1: CGP	0	0	Retained

Table 8-1. Outline of Peripheral Hardware Control Function of Control Register (4/5)

Peripheral Hardware	Control Register: Name	Address	Read/Write	b3 / b2 Symbol / b1 / b0	Peripheral Hardware Control Function: Functional Outline	Set Value	On Reset: Power-ON	Clock Stop	CE
Serial interface	Serial I/O1 mode select (PLLULJDG)	02H	R/W	SIO1TS	Starts serial interface 1.	0: Does not operate 1: Starts	0	0	0
				SIO1HIZ	Sets P0B1/SO1 pin.	0: General-purpose port 1: Serial out			
				SIO1CK1	Sets clock of serial interface 1	0 0: External 0 1: 37.5 kHz 1 0: 75 kHz 1 1: 450 kHz			
				SIO1CK0					
	Serial I/O0 mode select (SIO0MODE)	08H	R/W	SIO0CH	Sets 2-line or 3-line mode.	0 0: 2-line serial I/O 0 1: I²C bus 1 0: 3-line serial I/O 1 1: Setting prohibited	0	0	0
				SB	Sets I²C bus/serial I/O mode.				
				SIO0MS	Sets direction of clock.	0: External clock 1: Internal clock			
				SIO0TX	Sets I/O mode.	0: Input 1: Output			
	Serial I/O0 wait control (SIO0WT)	18H	R/W	SBACK	Sets and detects acknowledge	Setting and detecting of 0, 1	0	0	0
				SIO0NWT	in I²C bus mode.	0: Enabled 1: Disabled			
				SIO0WRQ1	Enables wait.	0 0: No wait 0 1: 8 clocks 1 0: 9 clocks 1 1: SB8 clocks			
				SIO0WRQ0	Sets wait timing of serial interface 0				
	Serial I/O0 wait status judge (SIO0WSTR)	19H	R	0					
				0					
				0					
				SIO0WSTT	Detects wait status of serial interface 0	0: Wait 1: Serial communication	0	0	0
	Serial I/O0 status judge (SIO0STUS)	28H	R/W	SIO0SF8	Detects clock counter of serial interface 0	Set when clock counter is 8; reset when clock counter is 0 or 1	0	0	0
				SIO0SF9		Set when clock counter is 9; reset when clock counter is 0 or 1			
				SBSTT	Detects number of clocks in I²C bus mode	Set when start condition - 9 th clock			
				SBBSY	Detects start and stop conditions in I²C bus mode	Sets when start condition - stop condition			
	Serial I/O0 interrupt mode (SIO0INT)	38H	R/W	SIO0IMD3	0	0 0: 7th clock 0 1: 8th clock	Undefined	Retained	Retained
				SIO0IMD2	0				
				SIO0IMD1	Sets interrupt condition of serial interface 0	0 0: 7th clock after start condition 0 1: Stop condition			
				SIO0IMD0					
	Serial I/O0 clock select (SIO0CLK)	39H	R/W	SIO0CK3	0		Undefined	Retained	Retained
				SIO0CK2	0				
				SIO0CK1	Sets internal clock of serial interface 0	0 0: 37.5 kHz 0 1: 75 kHz 1 0: 112.5 kHz 1 1: 225 kHz			
				SIO0CK0					
Frequency counter	IF counter open status judge (IFCGOSTR)	04H	R	0					
				0					
				0					
				IFCGOSTT	Detects gate opening/closing of frequency counter	0: Close 1: Open	0	—	—
	IF counter mode select (IFCMODE)	12H	R/W	IFCMD1	Sets mode of frequency counter	0 0: CGP 0 1: FMIF 1 0: AMIF 1 1: FCG	0	0	Retained
				IFCMD0					
				IFCCK1	Sets gate time of frequency counter	0 0: 1ms 0 1: 4 ms 1 0: 8 ms 1 1: Open 0 0: 1kHz 0 1: 100 kHz 1 0: 900 kHz 1 1:			
				IFCCK0					
	IF counter control (IFCCONT)	23H	R/W	0			0	0	Retained
				0					
				IFCSTRT	Specifies count start of frequency counter.	0: NOP instruction 1: Start			
				IFCRES	Specifies data reset of frequency counter.	0: NOP Instruction 1: Reset			

Table 8-1. Outline of Peripheral Hardware Control Function of Control Register (5/5)

Peripheral Hardware	Control Register Name	Address	Read/Write	b3 b2 Symbol b1 b0	Peripheral Hardware Control Function: Functional Outline	Set Value	On Reset: Power-ON	Clock Stop	CE
LCD driver	LCD mode select (LCDMODE)	10H	R/W	0			0	0	Retained
				0					
				KSEN	Sets key source signal output	0: Key source off 1: Key source on			
				LCDEN	Sets LCD display output	0: Display off 1: Display on			
	LCD port select (LCDPORT)	11H	R/W	R0YSEL	Sets $P0Y_0$-$P0Y_{15}$, $P0X_0$-$P0X_5$, $P0E_0$-$P0E_3$, and $P0F_0$-$P0F_3$ pins as general-purpose output port	0: LCD segment 1: General-purpose output port	0	0	Retained
				R0XSEL					
				R0ESEL					
				R0FSEL					
	Key input judge (KEYJDG)	16H	Read & Reset	0			0	0	0
				0					
				0					
				KEYJ	Detects key input latch of LCD key source	0: Not latched 1: Latched			

8.4 Notes on Using Register File

Remember the following three points, (1) through (3), when manipulating the write-only registers (W), read-only registers (R), and unused registers of the control registers (addresses 00H through 3FH of the register file):

(1) When a write-only register is read, an “undefined value” is read.
(2) Nothing is changed even if data is written to a read-only register.
(3) An “undefined value” is read if an unused portion is read. Nothing is changed even if data is written to this portion.

9. DATA BUFFER (DBF)

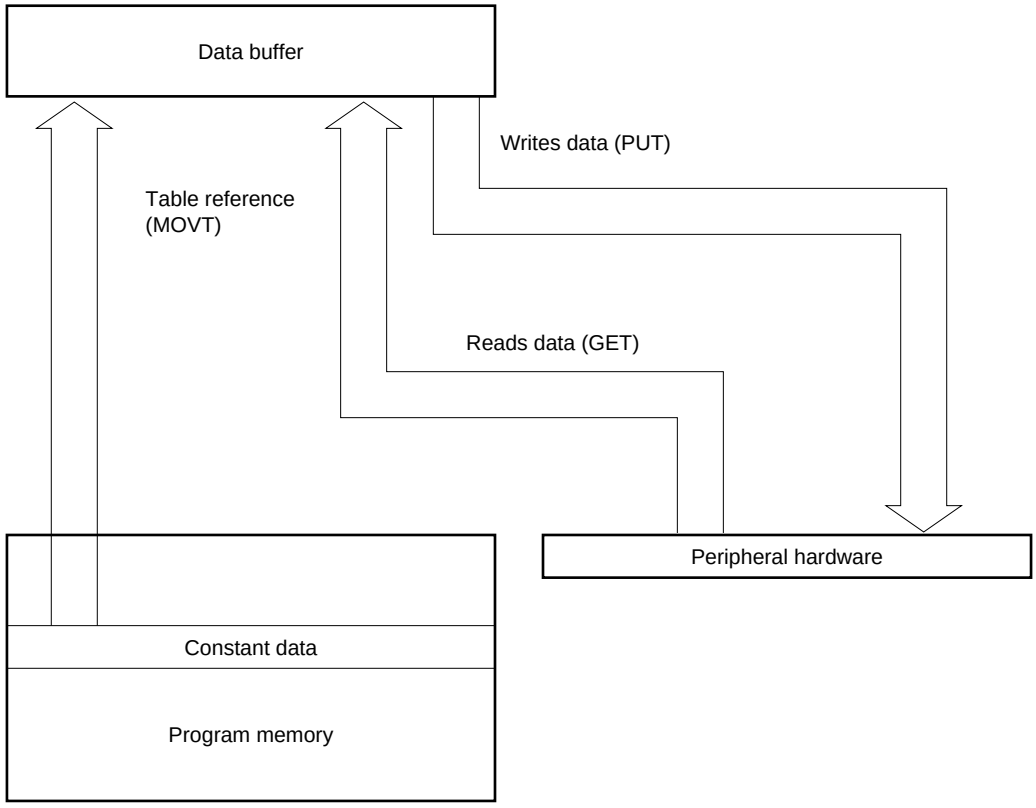
9.1 Outline of Data Buffer

Figure 9-1 outlines the data buffer.

The data buffer is located on the data memory and has the following two functions:

(1) Reads constant data on program memory (table reference)
(2) Transfers data with peripheral hardware

Figure 9-1. Outline of Data Buffer



9.2 Data Buffer

9.2.1 Configuration of data buffer

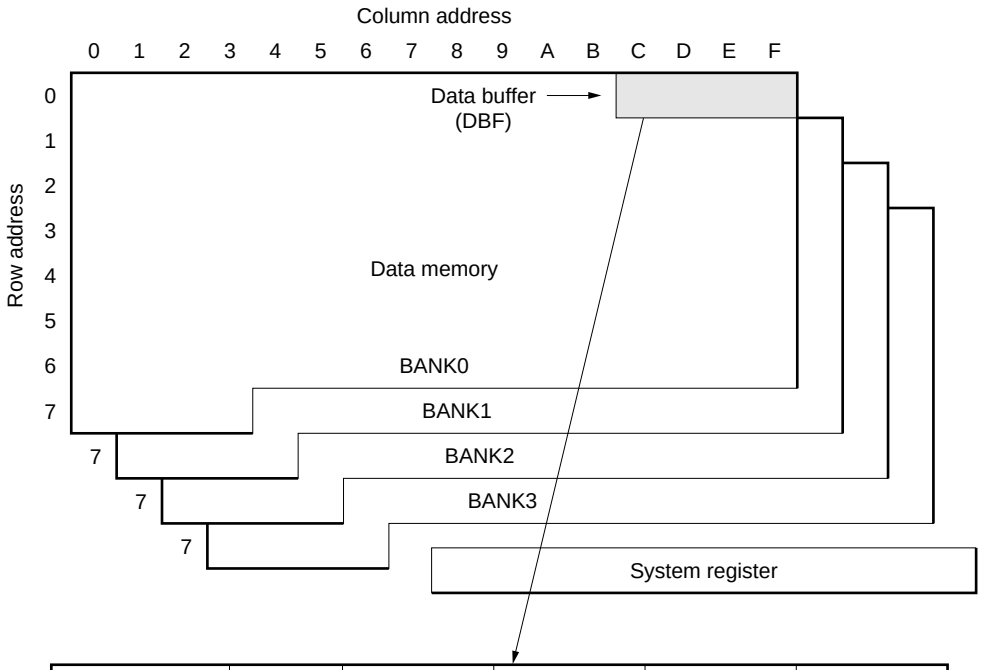
Figure 9-2 shows the configuration of the data buffer.

As shown in this figure, the data buffer consists of a total of 16 bits at addresses 0CH through 0FH of BANK0 on the data memory.

The MSB of the 16-bit data is the bit b_3 of address 0CH, and the LSB is the bit b_0 of address 0FH.

Because the data buffer is located on the data memory, it can be manipulated by any data memory manipulation instruction.

Figure 9-2. Configuration of Data Buffer



Data Memory	Address	0 CH				0 DH				0 ED				0 FH			
	Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data buffer	Bit	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
	Symbol	DBF 3				DBF 2				DBF 1				DBF 0			
	Data	^ M S B V															^ L S B V

Data

9.2.2 Table reference instruction (“MOVT DBF, @AR”)

The operation of the “MOVT DBF, @AR” instruction is described next.

When the table reference instruction is executed, one stack level is used.

All the program memory addresses, 0000H through 1EFBH, can be referenced.

```
MOVT DBF, @AR
```

This instruction reads the contents of the program memory addressed by the contents of the address register to the data buffer.

9.2.3 Peripheral hardware control instructions (“PUT”, “GET”)

The operations of the “PUT” and “GET” instructions are described next.

(1) GET DBF, p

Reads the data of the peripheral register addressed by p to the data buffer.

(2) PUT p, DBF

Sets the data of the data buffer to the peripheral register addressed by p.

9.3 List of Peripheral Hardware and Data Buffer Functions

Table 9-1 lists the functions of the peripheral hardware and data buffer.

Table 9-1. List of Peripheral Hardware and Data Buffer Functions (1/2)

Peripheral Hardware		Peripheral Register Transferring Data with Data Buffer			
		Name	Symbol	Peripheral Address	Execution of PUT/GET Instruction
A/D converter		A/D converter data register	ADCR	02H	PUT/GET
Serial interface	Serial interface 1 (SIO1)	Presettable shift register 1	SIO1SFR	03H	PUT/GET
	Serial interface 0 (I^2C, SBI, SIO0)	Presettable shift register 0	SIO0SFR	04H	
D/A converter (PWM output)	PWM_0 pin	PWM data register 0	PWMR0	05H	PUT/GET
	PWM_1 pin	PWM data register 1	PWMR1	06H	
	PWM_2 pin	PWM data register 2	PWMR2	07H	
LCD controller/driver	LCD segment group 0	LCD segment group register 0	LCDR0	08H	PUT
	LCD segment group 1	LCD segment group register 1	LCDR1	09H	
	LCD segment group 2	LCD segment group register 2	LCDR2	0AH	
	LCD segment group 3	LCD segment group register 3	LCDR3	0BH	
	LCD segment group 4	LCD segment group register 4	LCDR4	0CH	
	LCD segment group 5	LCD segment group register 5	LCDR5	0DH	
	LCD segment group 6	LCD segment group register 6	LCDR6	0EH	
	LCD segment group 7	LCD segment group register 7	LCDR7	0FH	
Output port	Port 0X	P0X group register	P0X	0CH	PUT
	Port 0Y	P0Y group register	P0Y	42H	PUT/GET
Clock generator port (CGP)		CGP data register	CGPR	20H	PUT/GET
Address register (AR)		Address register	AR	40H	PUT/GET
PLL frequency synthesizer		PLL data register	PLLR	41H	PUT/GET
Key source controller/decoder		Key source data register	KSR	42H	PUT/GET
Frequency counter		IF counter data register	IFC	43H	GET
12-bit timer	Timer modulo	Timer modulo register	TMM	46H	PUT/GET
	Timer counter	Timer counter	TMC	47H	GET

Table 9-1. List of Peripheral Hardware and Data Buffer Functions (2/2)

Function		
Number of I/O Bits of Data Buffer	Actual Number of Bits	Outline
8	6	Sets compare voltage V_{REF} data of A/D converter $V_{REF} = \frac{x-0.5}{64} \times V_{DD}$, $1 \le x \le 63$
8	8	Sets serial out data and reads serial in data
8	8	Sets duty factor of output signal of D/A converter Duty D = $\frac{x+0.25}{256} \times 100\%$, $0 \le x \le 1125$ Frequency f = 4349.5 Hz
8	7	LCD segment group 0
	4	LCD segment group 1
	7	LCD segment group 2
	7	LCD segment group 3 — Sets display data of each group
	7	LCD segment group 4 — 0: Display on
	3	LCD segment group 5 — 1: Display off
	7	LCD segment group 6
	7	LCD segment group 7
8	8	Sets output data of port 0X 0: low level, 1: high level
16	16	Sets output data of port 0Y 0: low level, 1: high level
8	7	Sets frequency of SG function Frequency f = $\frac{18}{2\ (2 \times x)}$ kHz Sets duty factor of VDP function Duty D = $\frac{x+2}{67}$, $0 \le x \le 63$
16	16	Transfers data with address register
16	16	Sets division value (N value) of PLL
16	16	Sets output data of key source signal
16	16	Reads measured value of frequency counter
16	12	Sets reference data of timer modulo
16	12	Sets data of up counter

9.4 Notes on Using Data Buffer

Remember the following three points when transferring data with the peripheral hardware via data buffer by executing the PUT instruction to access the unused peripheral address or write-only peripheral register or the GET instruction to access the read-only peripheral register:

(1) An “undefined value” is read when a write-only register is read.
(2) Nothing is changed even if data is written to a read-only register.
(3) An “undefined value” is read if an unused address is read. Nothing is changed even if data is written to this address.

10. INTERRUPT

10.1 Outline of Interrupt Block

Figure 10-1 outlines the interrupt block.

As shown in this figure, the interrupt block temporarily stops the program under execution and branches execution to an interrupt vector address when an interrupt request is output by peripheral hardware.

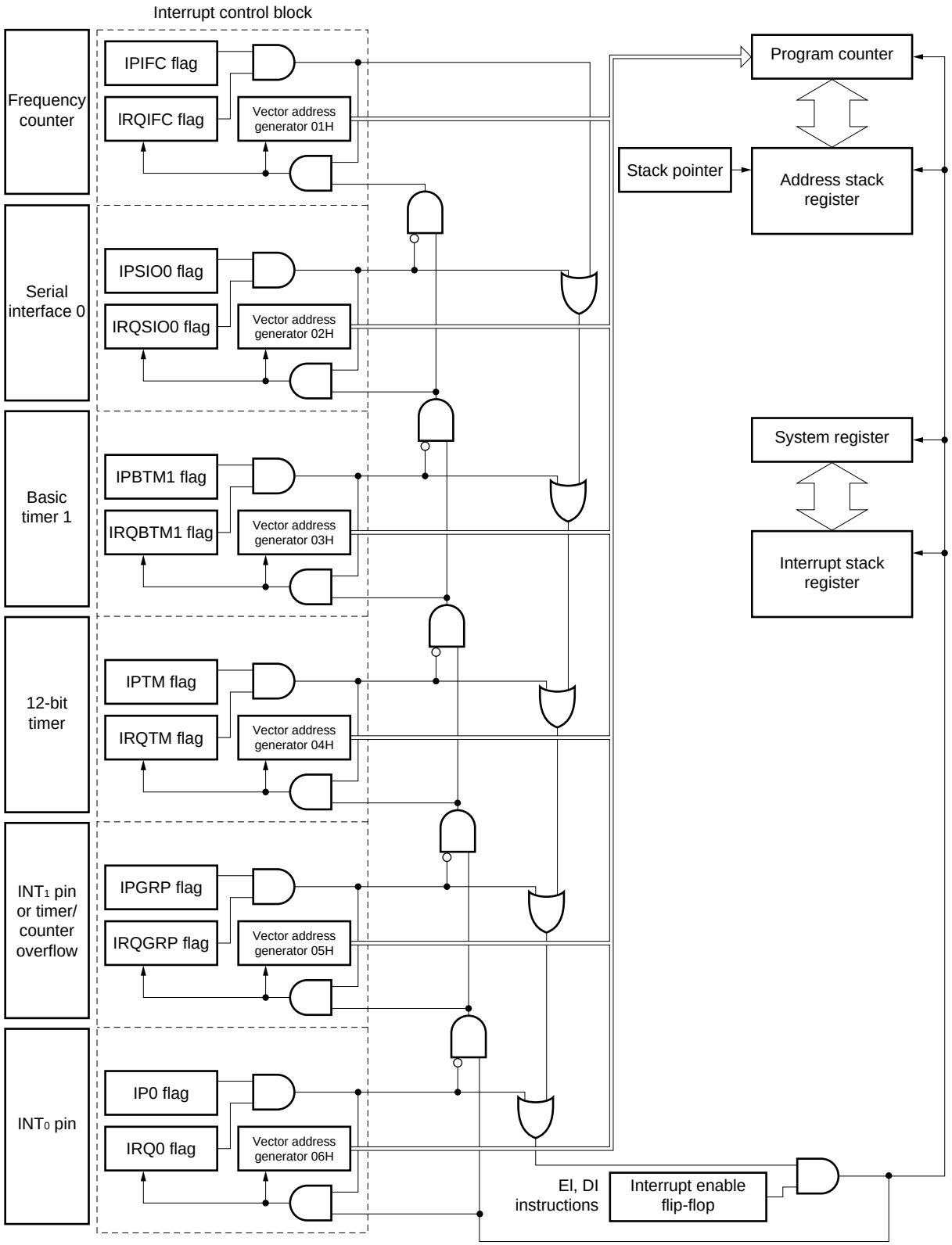
The interrupt block consists of an “interrupt control block” for each peripheral hardware, “interrupt enable flip-flop” that enables all interrupts, “stack pointer” that is controlled when an interrupt has been accepted, “address stack register”, “program counter”, and “system register stack”.

The “interrupt control block” of each peripheral hardware consists of an “interrupt request flag (IRQxxx) that detects each interrupt request”, “interrupt permission flag (IPxxx) that enables each interrupt”, and “vector address generator (VAG)” that specifies each vector address when an interrupt has been accepted.

The following peripheral hardware has an interrupt function:

- INT_0 pin
- Group (INT_1 pin or timer/counter overflow)
- 12-bit timer
- Basic timer 1
- Serial interface 0
- Frequency counter

Figure 10-1. Outline of Interrupt Block



10.2 Interrupt Control Block

The interrupt control block is provided to each peripheral hardware and detects an interrupt request, enables the interrupt, and generates a vector address when the interrupt has been accepted.

10.2.1 Configuration and function of interrupt request flag (IRQ×××)

The interrupt request flag (IRQ×××) is set to "1" when an interrupt request is issued from the corresponding peripheral hardware, and is reset to "0" when the interrupt has been accepted.

If an interrupt is not enabled, the status of issuance of each interrupt request can be detected by detecting the interrupt request flag (IRQ×××).

When "1" is directly written to an interrupt request flag via window register, the operation is equivalent to issuance of an interrupt request.

Once this flag has been set to "1", it is not reset until the corresponding interrupt is accepted or "0" is written to it via the window register.

If two or more interrupt requests are issued at the same time, the interrupt request flag corresponding to the interrupt that is not accepted is not reset.

The configuration and function of the interrupt request flag are illustrated below.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
IF counter interrupt request register (IREQIFC)	0	0	0	IRQIFC	3AH	R/W

Sets status of issuance of frequency counter interrupt request

0	Interrupt request not issued
1	Interrupt request issued

Fixed to "0"

On reset	Power-ON	0	0	0	0
	Clock stop				0
	CE				0

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Serial I/O0 interrupt request register (IREQSIO0)	0	0	0	IRQSIO0	3BH	R/W

IRQSIO0	Sets status of issuance of serial interface 0 interrupt request
0	Interrupt request not issued
1	Interrupt request issued

b_3	Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	↓	↓	↓	0
	CE	↓	↓	↓	0

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Basic timer 1 interrupt request register (IREQBTM1)	0	0	0	IRQBTM1	3CH	R/W

IRQBTM1	Sets status of issuance of basic timer 1 interrupt request
0	Interrupt request not issued
1	Interrupt request issued

b_3	Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	↓	↓	↓	0
	CE	↓	↓	↓	0

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
12-bit timer interrupt request register (IREQTM)	0	0	0	IRQTM	3DH	R/W

IRQTM — Sets status of issuance of 12-bit timer interrupt request

0	Interrupt request not issued
1	Interrupt request issued

Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop				0
	CE				0

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Group interrupt request register (IREQGRP)	INT1	0	0	IRQGRP	3EH	R/W

IRQGRP — Sets status of issuance of INT_1 pin or timer/counter overflow interrupt request

0	Interrupt request not issued
1	Interrupt request issued

Fixed to "0"

INT1 — Detects status of INT_1 pin

0	Low level is input
1	High level is input

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	0			0
	CE	0			0

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
INT0 interrupt request register (IREQINT0)	INT0	0	0	IRQ0	3FH	R/W

b_0	Sets status of INT_0 pin interrupt request issuance
0	Interrupt request not issued
1	Interrupt request issued

b_1	Fixed to "0"

b_2	Detects status of INT_1 pin
0	Low level is input
1	High level is input

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	0	↓	↓	0
	CE	0	↓	↓	0

10.2.2 Configuration and function of interrupt permission flag (IPxxx)

Each interrupt permission flag enables the interrupt of each peripheral hardware.
So that an interrupt is accepted, all the following three conditions must be satisfied:

- Interrupt is enabled by corresponding interrupt permission flag.
- Interrupt request is issued by corresponding interrupt request flag.
- “EI” instruction (that enables all interrupts) is executed.

The configuration and function of the interrupt permission flag are illustrated below.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Interrupt permission 1 register (INTPM1)	0	0	IPIFC	IPSIO0	2EH	R/W

b0 (IPSIO0):

	Enables serial I/O0 interrupt
0	Disables
1	Enables

b1 (IPIFC):

	Enables IF counter interrupt
0	Disables
1	Enables

b3, b2: Fixed to "0"

On reset		b_3	b_2	b_1	b_0
	Power-ON	0	0	0	0
	Clock stop	↓	↓	0	0
	CE	↓	↓	0	0

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Interrupt permission 2 register (INTPM2)	IPBTM1	IPTM	IPGRP	IP0	2FH	R/W

IPBTM1	Enables basic timer 1 interrupt
0	Disables
1	Enables

IPTM	Enables 12-bit timer interrupt
0	Disables
1	Enables

IPGRP	Enables INT_1 pin or timer/counter overflow interrupt
0	Disables
1	Enables

IP0	Enables INT_0 pin interrupt
0	Disables
1	Enables

On reset		b_3	b_2	b_1	b_0
	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

10.2.3 Vector address generator (VAG)

The vector address generator generates a branch address (vector address) of the program memory corresponding to an interrupt source when each peripheral hardware interrupt has been accepted.

Table 10-1 shows the vector addresses corresponding to the respective interrupt sources.

Table 10-1. Vector Addresses Corresponding to Respective Interrupt Sources

Interrupt Source	Vector Address
INT_0 pin	06H
INT_1 pin or timer/counter overflow	05H
12-bit timer	04H
Basic timer 1	03H
Serial interface 0	02H
Frequency counter	01H

10.3 Interrupt Stack

10.3.1 Configuration and function of interrupt stack register

Figure 10-2 shows the configuration of the interrupt stack register and the system registers that are saved to the interrupt stack register.

To the interrupt stack register, the contents of the following system registers are saved when an interrupt has been accepted.

- Window register (WR)
- Bank register (BANK)
- General register pointer (RP)
- Program status word (PSWORD)

When an interrupt has been accepted and the contents of the above system registers have been saved to the interrupt stack register, the contents of the system registers, except the window register, are reset to “0”.

The interrupt stack register can save up to three levels of the contents of the above system registers.

Therefore, interrupts can be nested up to three levels.

The contents of the interrupt stack register are restored to the system registers when the interrupt return instruction (“RETI”) is executed.

Figure 10-2. Configuration of Interrupt Stack Register

Interrupt Stack Register (INTSK)																				
Name	Window stack (WRSK)				Bank stack (BANKSK)				Register pointer stack H (RPHSK)				Register pointer stack L (RPLSK)				Status stack (PSWSK)			
Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
0H					–	–			–	–										
1H					–	–			–	–										
2H					–	–			–	–										

Remark –: Bit not saved

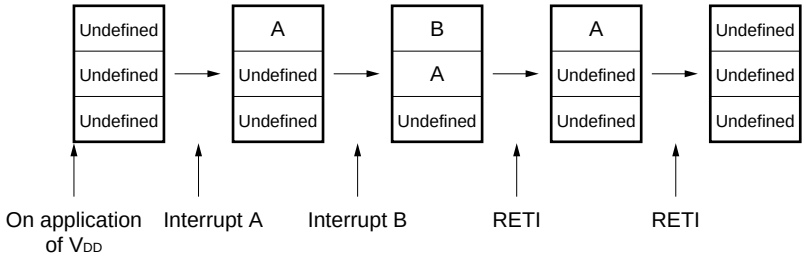
10.3.2 Operation of interrupt stack register

Figure 10-3 shows the operation of the interrupt stack register.

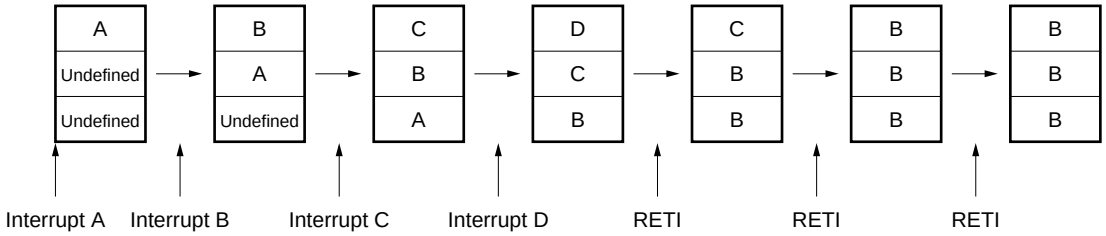
When interrupts are nested exceeding four levels, the contents saved first are dumped and therefore, must be saved by program.

Figure 10-3. Operation of Interrupt Stack Register

(a) When interrupts are nested within 3 levels



(b) If interrupts are nested exceeding 3 levels



10.4 Stack Pointer, Address Stack Register, and Program Counter

The address stack register saves a return address from which program execution is resumed when execution has returned from an interrupt processing routine.

The stack pointer specifies the address of an address stack register.

When an interrupt has been accepted, the value of the stack pointer is decremented by one, and the value of the program counter at that time is saved to the address stack register specified by the stack pointer.

When a dedicated return instruction “RETI” is executed after the processing of the interrupt processing routine has been completed, the contents of the address stack register specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

For more information, refer to **3. ADDRESS STACK (ASK)**.

10.5 Interrupt Enable Flip-Flop (INTE)

The interrupt enable flip-flop enables all the interrupts.

When this flip-flop is set, all the interrupts are enabled. When it is reset, all the interrupts are disabled.

To set or reset this flip-flop, a dedicated instruction, “EI (to set)” or “DI (to reset)”, is used.

The “EI” instruction sets this flip-flop when the next instruction is executed, and the “DI” instruction resets the flip-flop during its execution.

When an interrupt is accepted, this flip-flop is automatically reset.

Even if the “DI” instruction is executed in the DI status, or if the “DI” instruction is executed in the EI status, nothing is affected.

This flip-flop is reset at power-ON reset or CE reset, and when the clock stop instruction is executed.

10.6 Accepting Interrupt

10.6.1 Accepting Interrupt and Priority

An interrupt is accepted in the following sequence:

(1) Each peripheral hardware outputs an interrupt request signal to the interrupt control block when an interrupt condition (for example, input of a falling edge to the INT_0 pin) is satisfied.

(2) When the interrupt control block accepts the interrupt request signal from the peripheral hardware, it sets the corresponding interrupt request flag (for example, IRQ0 flag for the INT_0 pin) to "1".

(3) If the interrupt permission flag (for example, IP0 flag for the IRQ0 flag) corresponding to the interrupt request flag that has been set to "1" when each interrupt request flag is set to "1", the interrupt control block outputs "1".

(4) The signal output from the interrupt control block is ORed with the output of the interrupt enable flip-flop, and an interrupt accept signal is output.
This interrupt enable flip-flop is set to "1" by the "EI" instruction and reset to "0" by the "DI" instruction.
If the interrupt control block outputs "1" while the interrupt enable flip-flop is set to "1", the interrupt enable flip-flop outputs "1", and the interrupt is accepted.

As shown in Figure 10-1, the output of the interrupt enable flip-flop is input to the interrupt control block via an AND circuit when the interrupt is accepted.

The interrupt request flag is reset to "0" by the signal input to the interrupt control block, and a vector address for the interrupt is output.

If the interrupt control block outputs "1" at this time, the interrupt accept signal is not transferred to the next stage. If two or more interrupt requests are issued at the same time, therefore, the interrupts are accepted according to the following priority:

INT_0 pin > INT_1 pin or timer/counter overflow > timer > serial interface 1 > frequency counter

If the interrupt permission flag is not set to "1", the interrupt generated by the corresponding source is not accepted.
If the interrupt permission flag is reset to "0", therefore, the interrupt with a high hardware priority can be disabled.

10.6.2 Timing chart illustrating acceptance of interrupt

Figure 10-4 shows the timing chart illustrating how an interrupt is accepted.

(1) in Figure 10-4 is the chart illustrating how one interrupt is accepted.

(a) in (1) indicates the case where the interrupt request flag is set to “1” last, and (b) indicates the case where the interrupt permission flag is set to “1” last.

In either case, the interrupt is accepted when all the interrupt request flag, interrupt enable flip-flop, and interrupt permission flag have been set to “1”.

If the flag or flip-flop that is set to 1 last in the first instruction cycle of the “MOVT DBF, @AR” instruction or when an instruction that satisfies a given skip condition is executed, the interrupt is accepted in the second instruction cycle of the “MOVT DBF, @AR” instruction or after the skipped instruction (treated as NOP) has been executed.

The interrupt enable flip-flop is set in the instruction cycle next to the one in which the “EI” instruction was executed.

(2) in Figure 10-4 is the timing chart illustrating how two or more interrupts are used.

To use two or more interrupts, the interrupts are accepted according to the priority determined by hardware if all the interrupt permission flags are set. The hardware priority can be changed by manipulating the interrupt permission flag by program.

The “interrupt cycle” shown in Figure 10-4 is a special cycle in which the interrupt request flag is reset after the interrupt has been accepted, a vector address is specified, and the contents of the program counter are saved. This cycle requires 4.44 μs, which is equivalent to one instruction execution time.

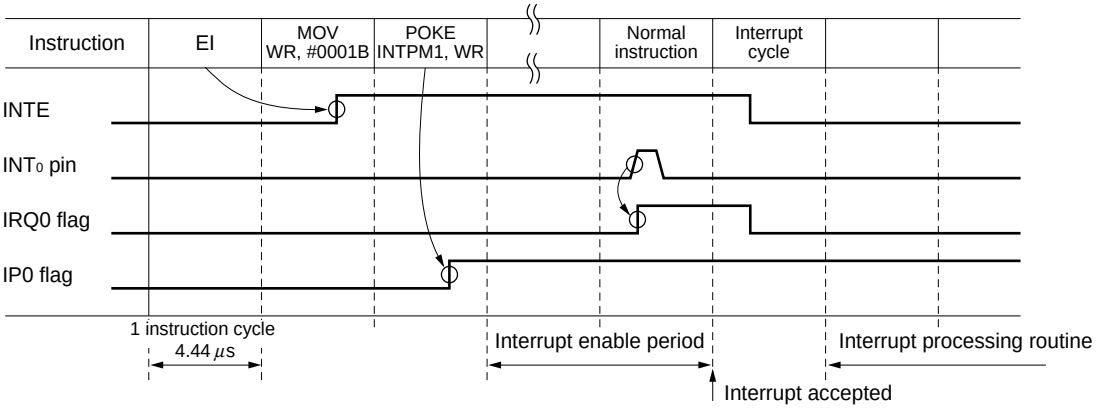
For details, refer to **10.7 Operation after Accepting Interrupt**.

Figure 10-4. Timing Charts Illustrating Acceptance of Interrupts (1/2)

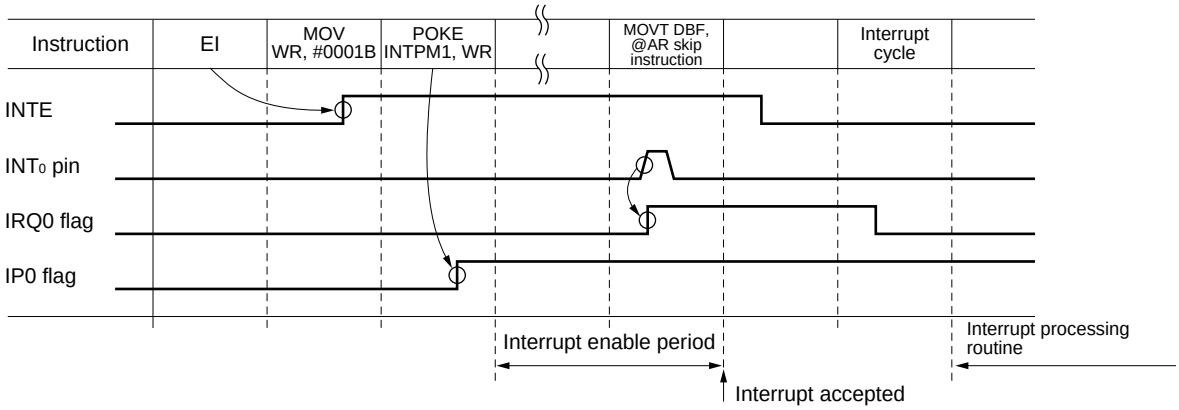
(1) When one interrupt (e.g., rising of INT_0 pin) is used

(a) If interrupt mask time is not specified by interrupt permission flag (IPxxx)

<1> If a normal instruction other than "MOVT" and instruction that satisfies a skip condition is executed when interrupt is accepted



<2> If "MOVT" or instruction that satisfies a skip condition is executed when interrupt is accepted



(b) If interrupt pending period is specified by interrupt permission flag

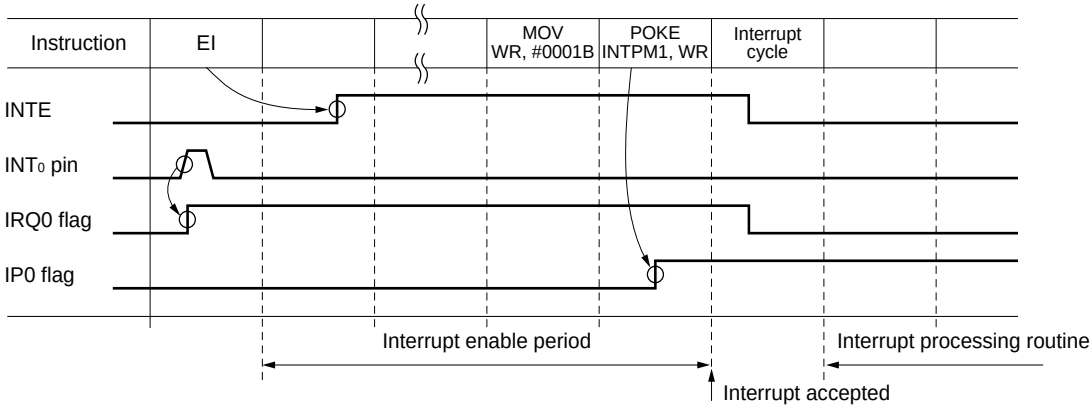
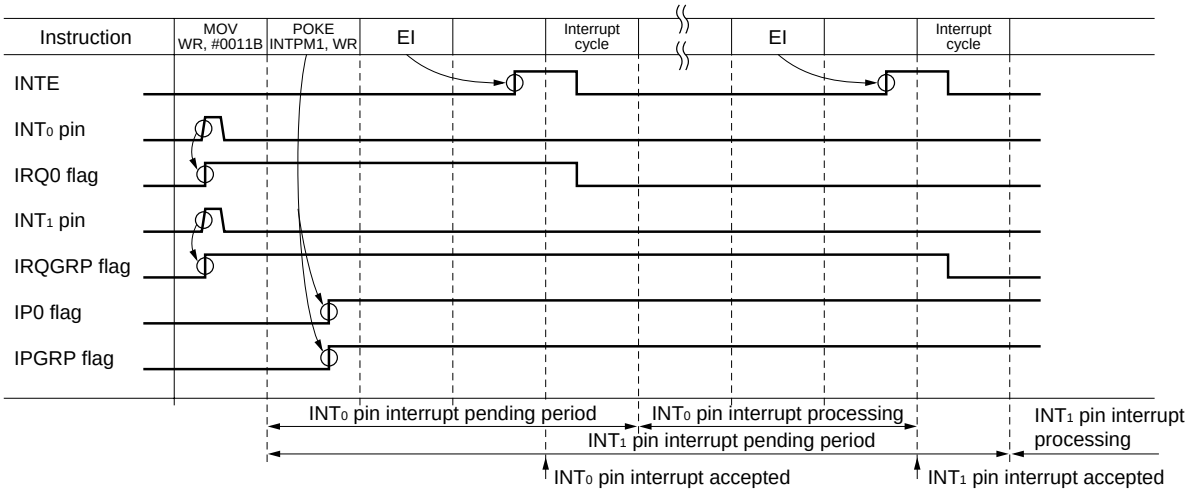


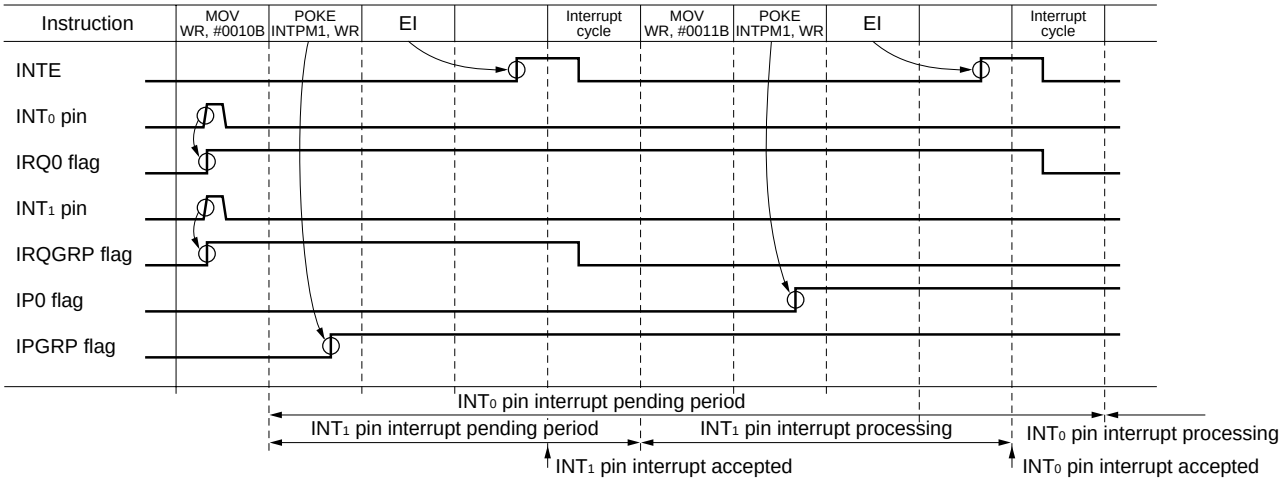
Figure 10-4. Timing Charts Illustrating Acceptance of Interrupts (2/2)

(2) When two or more interrupts (e.g., INT_0 pin and INT_1 pin) are used

(a) Hardware priority



(b) Software priority



10.7 Operation after Accepting Interrupt

When an interrupt has been accepted, the following processing is sequentially performed automatically:

(1) The interrupt enable flip-flop and the interrupt request flag corresponding to the accepted interrupt request are reset to “0”. The result is that the interrupt is disabled.

(2) The contents of the stack pointer are decremented by one.

(3) The contents of the program counter are saved to the address stack register specified by the stack pointer. The contents of the program counter are the program memory address to be executed next when the interrupt is accepted.
For example, if a branch instruction is executed, the contents of the program counter are the branch destination address. If a subroutine call instruction is executed, they are the called address. If the skip condition of a skip instruction is satisfied, the next instruction is executed as “NOP” instruction, and then the interrupt is accepted. The contents of the program counter are the skipped address.

(4) The contents of the window register (WR), bank register (BANK), general register pointer (RP), and program status word (PSWORD) are saved to the interrupt stack.

(5) The contents of the vector address generator corresponding to the accepted interrupt are transferred to the program counter. The result is that execution branches to an interrupt processing routine.

Processing (1) through (5) above is executed in one special instruction cycle (4.44 μs) not accompanied by normal instruction execution.

This instruction cycle is called an “interrupt cycle”.

Therefore, one instruction cycle time is required after an interrupt has been accepted until execution branches to the corresponding vector address.

10.8 Returning from Interrupt Processing Routine

To return execution from an interrupt processing routine to the processing during which the interrupt was accepted, a dedicated instruction, “RETI”, is used.

When this instruction is executed, the following processing is sequentially executed automatically:

(1) The contents of the address stack register specified by the stack pointer are restored to the program counter.

(2) The contents of the interrupt stack are restored to the window register (WR), bank register (BANK), general register pointer (RP), and program status word (PSWORD).

(3) The contents of the stack pointer are incremented by one.

Processing (1) through (3) above is performed in one instruction cycle in which the “RETI” instruction is executed.

The difference between the “RETI” instruction and the subroutine return instructions “RET” and “RETSK” is how the contents of the window register, bank register, general register pointer, and program status word are restored, as in step (2) above.

10.9 External (INT_0 and INT_1 Pins) Interrupts

10.9.1 Outline of external interrupt

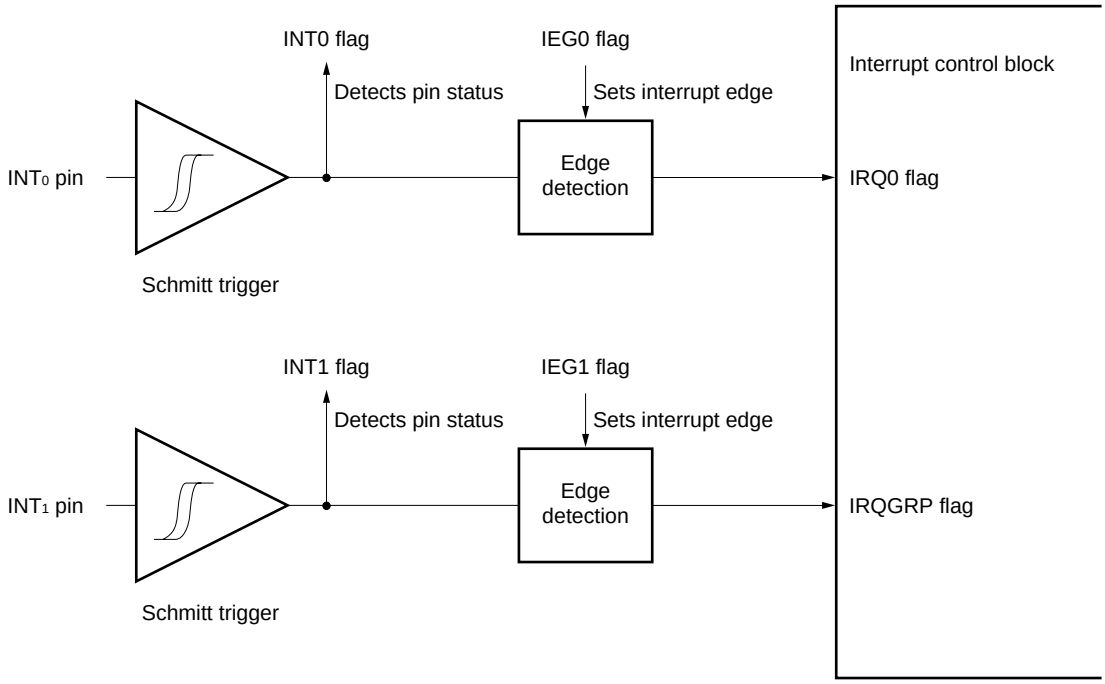
Figure 10-5 outlines external interrupts.

As shown in this figure, an external interrupt request is issued when a rising edge or falling edge is input to the INT_0 or INT_1 pin.

Whether the interrupt request is issued at the rising or falling edge is set independently by program.

The INT_0 and INT_1 pins are Schmit trigger input pins to prevent malfunctioning due to noise. These pins do not accept a pulse input of less than 1 μs.

Figure 10-5. Outline of External Interrupts



10.9.2 Edge detection block

The edge detection block sets the input signal edge (rising or falling) at which interrupt requests are issued from the INT_0 and INT_1 pins, and detects the set edge.

The edge is set by the IEG0 and IEG1 flags.

The configuration and function of each flag are described next.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Interrupt edge select register (INTEDGE)	0	0	IEG1	IEG0	1FH	R/W

IEG0:
- Sets input edge at which interrupt request of INT_0 pin is issued
- 0: Rising edge
- 1: Falling edge

IEG1:
- Sets input edge at which interrupt request of INT_1 pin is issued
- 0: Rising edge
- 1: Falling edge

b_2: Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop			0	0
	CE			0	0

When the edge at which the interrupt request is issued is changed by the IEG0 and IEG1 flags, the interrupt request signal may be issued as soon as the edge has been changed.

For example, suppose, as shown in Table 10-2, that the IEG0 flag is now set to “1” (specifying the falling edge) and that a high level is input from the INT_0 pin. If the IEG0 flag is reset at this time, the edge detection circuit assumes that the rising edge has been input, and issues the interrupt request.

Table 10-2. Issuance of Interrupt Request by Changing IEG0 and IEG1 Flags

Changes in IEG0 and IEG1 Flags	Status of INT_0 and INT_1 Pins	Issuance of Interrupt Request	Status of IRQ0 and IRQGRP Flags
$1 \rightarrow 0$ (falling)	Low level	Not issued	Retains previous status
	High level	Issued	Set to “1”
$0 \rightarrow 1$ (rising)	Low level	Issued	Set to “1”
	High level	Not issued	Retains previous status

10.9.3 Pin status detection block

The level of the signals input to the INT_0 and INT_1 pins can be detected by using the INT0 and INT1 flags.

The INT0 and INT1 flags can be set to “1” or reset to “0” via window register, regardless of whether an interrupt request is issued or not. Therefore, these pins can be used as a 2-bit general-purpose input port when the interrupt function is not used.

If interrupts are not enabled, these flags can be used as a general-purpose input port that can detect the rising edge or falling edge by reading the contents of the corresponding interrupt request flags (IRQ0 and IRQGRP flags). In this case, however, the interrupt request flags are not automatically reset to “0” and must be reset by program.

For the details of the configuration and function of the INT0 and INT1 flags, refer to **10.2 Interrupt Control Block**.

10.10 Internal Interrupts

Five internal interrupt sources are available: timer/counter overflow, 12-bit timer, basic timer 1, serial interface 0, and frequency counter.

10.10.1 Timer/counter overflow interrupt

The timer/counter overflow interrupt issues an interrupt request when the 12-bit timer/counter overflows.

The timer/counter overflow interrupt or the interrupt caused by the INT_1 pin can be selected by using the IGRPSL flag.

The configuration and function of this flag are shown below.

For details, refer to **10.9 External (INT_0 and INT_1 Pins) Interrupts** and **11. TIMER FUNCTION**.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Group interrupt select register (IGRPSELR)	0	0	0	IGRPSL	0FH	R/W

IGRPSL	
	Sets interrupt source of IRQGRP flag
0	Issues interrupt request by INT_1 pin
1	Issues interrupt request by overflow of timer/counter

Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	↓	↓	↓	0
	CE	↓	↓	↓	0

10.10.2 12-bit timer interrupt

The 12-bit timer interrupt request can be issued at fixed time intervals.

For details, refer to **11. TIMER FUNCTION**.

10.10.3 Basic timer 1 interrupt

The basic timer 1 interrupt request can be issued at fixed time intervals.

For details, refer to **11. TIMER FUNCTION**.

10.10.4 Serial interface 0 interrupt

The serial interface 0 interrupt request can be issued at completion of the serial out or serial in operation.

For details, refer to **19. SERIAL INTERFACE**.

10.10.5 Frequency counter

The frequency counter interrupt request can be issued at completion of the count operation.

For details, refer to **20. FREQUENCY COUNTER (FC)**.

11. TIMER FUNCTION

The timer function is used to control program execution time.

11.1 Configuration of Timer

Figure 11-1 shows the configuration of the timer.

As shown in this figure, the timer block consists of a basic timer 0 carry block, basic timer 1 interrupt block, and 12-bit timer block.

The clock generation circuit that sets time to each timer consists of a clock select blocks A, B, and C, basic timer clock select register (BTMCLK: RF address 09H) of the control register, and timer counter clock select register (TMCLK: RF address 0CH).

The clock of each timer is generated by dividing the system clock (4.5 MHz). If the crystal oscillator is not 4.5 MHz, the clock of each timer changes accordingly.

11.1.1 Configuration of basic timer 0 carry block

The basic timer 0 carry block consists of a clock select block A and basic timer 0 carry FF block.

11.1.2 Configuration of basic timer 1 interrupt block

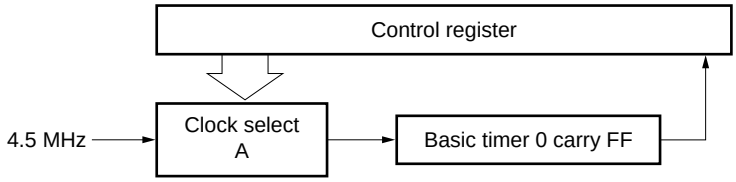
The basic timer 1 interrupt block consists of a clock select block B and interrupt control block.

11.1.3 Configuration of 12-bit timer block

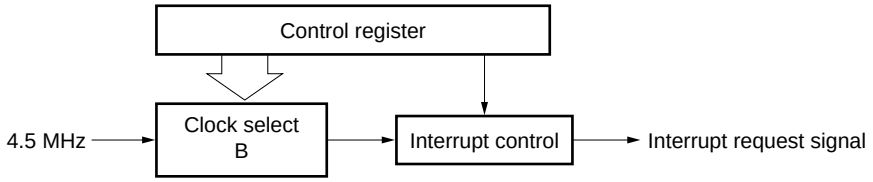
The 12-bit timer block consists of a clock select block C, 12-bit timer mode control block, count block, and interrupt control block.

Figure 11-1. Configuration of Timer Block

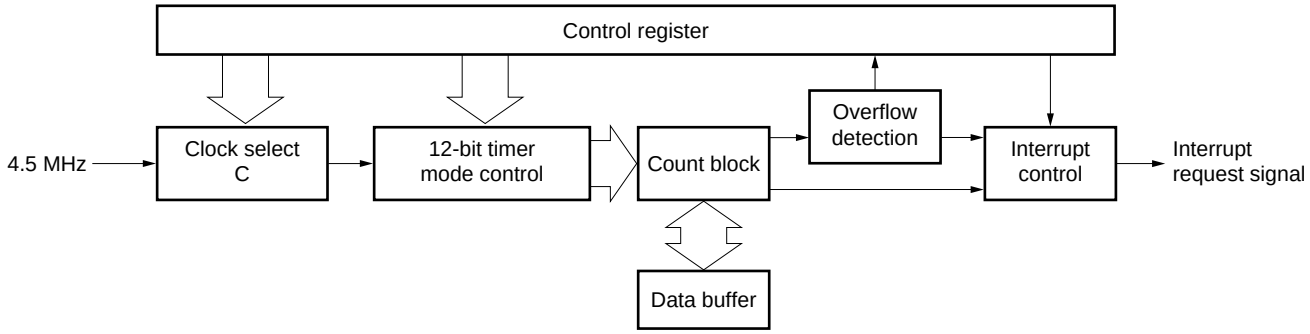
- **Basic timer 0 carry block**



- **Basic timer 1 interrupt block**



- **12-bit timer block**



11.2 Functional Outline of Timer

The timer can be used in three ways: to detect the carry FF of the basic timer 0 carry, to use the interrupt of basic timer 1, and to use the interrupt of the 12-bit timer.

11.2.1 Functional outline of basic timer 0 carry

The basic timer 0 carry controls time by detecting via program the status of the basic timer 0 carry FF that is set at fixed intervals.

For details, refer to **11.3 Basic Timer 0 Carry**.

11.2.2 Functional outline of basic timer 1 interrupt

The basic timer 1 interrupt controls time by generating an interrupt at fixed time intervals.

For details, refer to **11.4 Basic Timer 1 Interrupt**.

11.2.3 Functional outline of 12-bit timer

The 12-bit timer counts up the basic clocks with a 12-bit counter. When the count value coincides with the data set by program, it generates an interrupt to control time.

For details, refer to **11.5 12-Bit Timer**.

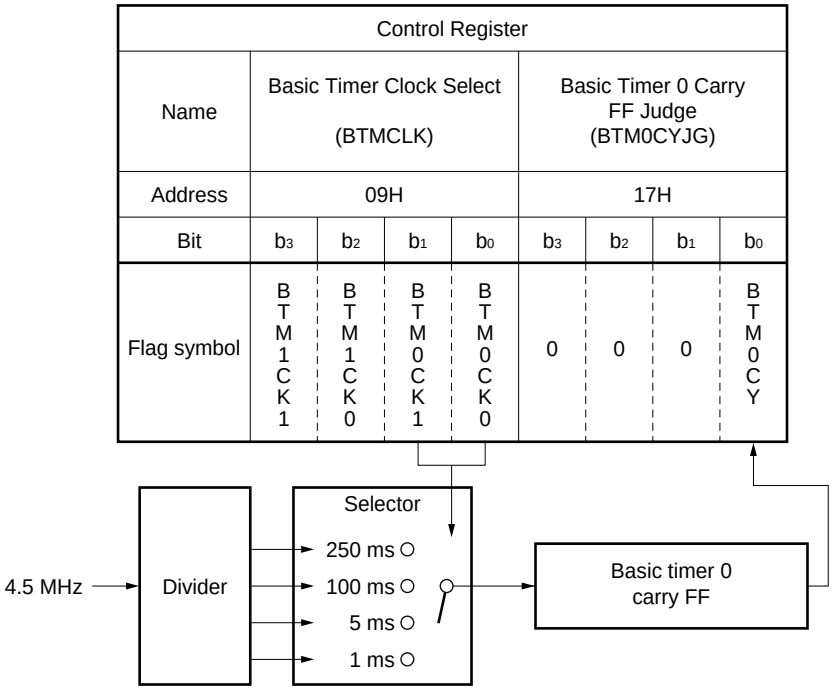
11.3 Basic Timer 0 Carry

11.3.1 Configuration of basic timer 0 carry

Figure 11-2 shows the configuration of the basic timer 0 carry.

As shown in this figure, the basic timer 0 carry consists of a divider, selector, and basic timer 0 carry FF block.

Figure 11-2. Configuration of Basic Timer 0 Carry



11.3.2 Function of basic timer 0 carry

The basic timer 0 carry is set to 1 at the rising edge of the basic timer 0 carry FF setting pulse set by the lower 2 bits (BTM0CK1 and BTM0CK0 flags) of the basic timer clock select register.

The content of the basic timer 0 carry FF corresponds to the least significant bit (BTM0CY flag) of the basic timer 0 carry FF judge register (BTM0CYJG: RF address 17H) on a one-to-one basis. When the basic timer 0 carry FF is set to 1, the BTM0CY flag is simultaneously set to 1.

The BTM0CY flag is reset to 0 on reading its content to the window register by the "PEEK"instruction (Read & Reset).

When the BTM0CY flag is reset to 0, the basic timer 0 carry FF is simultaneously reset to 0.

By reading the BTM0CY flag by program, therefore, a timer with the time set via the basic timer clock select register can be created.

When using the basic timer 0 carry, bear in mind the following point:

Caution The basic timer 0 carry is disabled from being set on power application (at V_{DD} reset) and is not set until the content of the BTM0CY flag is once read by the "PEEK" instruction.

Consequently, when the BTM0CY flag is read for the first time after power-ON reset, "0" is always read. After that, the flag is set to 1 at time intervals set by the basic timer clock select register.

The basic timer 0 carry also controls the timing of reset by the CE pin (CE reset).

When the CE pin goes high, CE reset is effected in synchronization with the timing at which the basic timer 0 carry FF is set next.

Therefore, a power failure can be detected by reading the content of the BTM0CY flag at system reset (power-ON reset or CE reset). For details, refer to **11.3.7 Notes on using basic timer 0 carry** and **13. RESET**.

Because the BTM0CY flag is a read-only flag, the device operation is not affected in any way even if data is written to this flag by using the "POKE" instruction. However, an error occurs when the 17K series assembler (AS17K) is used. For details, refer to **8.4 Notes on Using Register File**.

11.3.3 Configuration and function of basic timer clock select register (BTMCLK)

The basic timer clock select register sets two time intervals of the internal basic timer 0 carry and basic timer 1 interrupt.

The time intervals of the basic timer 0 carry and basic timer 1 interrupt can be independently set.

The configuration and function are shown next.

Figure 11-3 shows the waveform of the timer time setting pulse.

Name	Flag Symbol b_3	b_2	b_1	b_0	Address	Read/ Write
Basic timer clock select register (BTMCLK)	BTM1CK1	BTM1CK0	BTM0CK1	BTM0CK0	09H	R/W

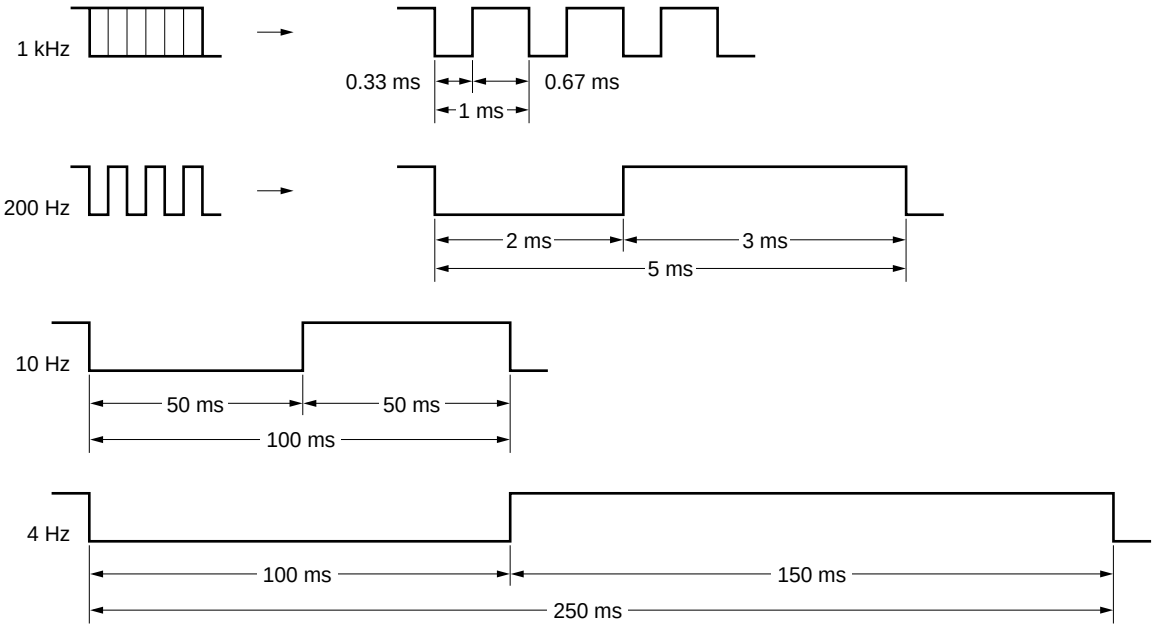
BTM0CK1	BTM0CK0	Sets time interval at which basic timer 0 carry is set
0	0	100 ms
0	1	250 ms
1	0	5 ms
1	1	1 ms

BTM1CK1	BTM1CK0	Sets time interval at which basic timer 1 interrupt is set[Note]
0	0	100 ms
0	1	250 ms
1	0	5 ms
1	1	1 ms

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	Retained			

Note Refer to **11.4** for the basic timer 1 interrupt.

Figure 11-3. Waveform of Timer Time Setting Pulse



11.3.4 Configuration and function of basic timer 0 carry flip-flop (FF) judge register (BTM0CYJG)

The basic timer 0 carry flip-flop (FF) judge register detects the status of the basic timer 0 carry flip-flop (FF) of the internal timer.

The configuration and function of BTM0CYJG are illustrated below.

Name	Flag Symbol b_3	b_2	b_1	b_0	Address	Read/ Write
Basic timer 0 carry FF judge register (BTM0CYJG)	0	0	0	BTM0CY	17H	R & Reset

	Detects status of basic timer 0 carry FF
0	Basic timer 0 carry FF is not set
1	Basic timer 0 carry FF is set

Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop				1
	CE				1

The BTM0CY flag is set at time intervals set by the basic timer clock select register (BTMCLK).

The status of this flag is detected by the “PEEK” instruction via the window register.

If the BTM0CY flag is set at this time, its value is transferred to the window register and then the BTM0CY flag is reset (Read & Reset).

Because the BTM0CY flag is reset to “0” at power-ON reset and is set to “1” at CE reset and at CE reset after execution of the clock stop instruction, it can be used to detect a power failure.

The BTM0CY flag is not set once V_{DD} has been applied until the “PEEK” instruction is executed. Once the “PEEK” instruction has been executed, it is set at time intervals set by the basic timer clock select register.

11.3.5 Example of use of timer with BTM0CY flag

Here is a program example:

Example

```
M1          MEM         0.10H          ; 1-second counter
            INITFLG NOT BTM0CK1, BTM0CK0
                                       ; Embedded macro
                                       ; Sets basic timer 0 carry FF setting time to 250 ms
LOOP:
  SKT1      BTM0CY                     ; Embedded macro
                                       ; Tests BTM0CY flag.  If it is “0”, branches to NEXT
  BR        NEXT
  ADD       M1, #0100B                 ; Adds 4 to data memory M1
  SKT1      CY                         ; Embedded macro
                                       ; Tests CY flag
  BR        NEXT                       ; If it is “0”, branches to NEXT
  Processing A                         ; If it is “1”, executes processing A
NEXT:
  Processing B                         ; Executes processing B and branches to LOOP
  BR        LOOP
```

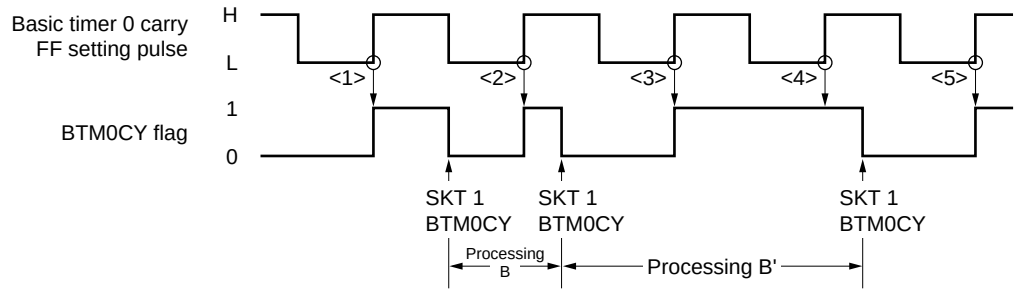
This program executes processing A every 1 second.

When creating this program, the following point must be noted.

Caution The time interval at which the BTM0CY flag is detected must be shorter than the time interval at which the basic timer 0 carry FF is set to 1.

In the above example, if processing B requires 250 ms or longer as shown in Figure 11-4, the basic timer 0 carry FF is not set.

Figure 11-4. Detection of BTM0CY Flag and Basic Timer 0 Carry FF



Status of BTM0CY flag set in <3> is not detected because time of processing B' is too long after BTM0CY flag that was set in <2> has been detected.

11.3.6 Timer error due to BTM0CY flag

Timer errors due to the BTM0CY flag include an error due to the detection time of the BTM0CY flag and an error that occurs when the basic timer 0 carry FF setting time is changed.

The following paragraphs (1) and (2) describe the respective errors.

(1) Error due to detection time of BTM0CY flag

As described in **11.3.5**, the time interval at which the BTM0CY flag is detected must be shorter than the time interval at which the basic timer 0 carry FF is set to 1.

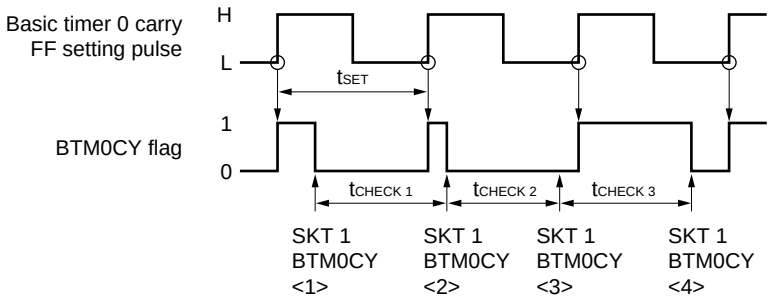
Where the time interval at which the BTM0CY flag is detected is t_{CHECK} and the time interval at which the basic timer 0 carry FF is set is t_{SET} (250 ms, 100 ms, 5 ms, or 1 ms), the relation between the two must be as follows:

$t_{CHECK} < t_{SET}$

The timer error when the BTM0CY flag is detected is as shown in Figure 11-5.

$0 < \text{error} < t_{CHECK}$

Figure 11-5. Error due to BTM0CY Flag Detection Time Interval



As shown in Figure 11-5, when the BTM0CY flag is detected in <2>, the timer is updated because the flag is “1”.

When the BTM0CY flag is detected next time in <3>, the timer is not updated until the flag is detected again in <4> because the flag is “0”.

Consequently, the time of the timer at this time is extended by the time of t_{CHECK3}.

(2) Error when basic timer 0 carry FF setting time is changed

The basic timer 0 carry FF setting time is set by the BTM0CK1 and BTM0CK0 flags of the basic timer clock select register.

As shown in Figures 11-2 and 11-3, the timer time setting pulse can be selected from the four types: 1 kHz, 200 Hz, 10 Hz, and 4 Hz.

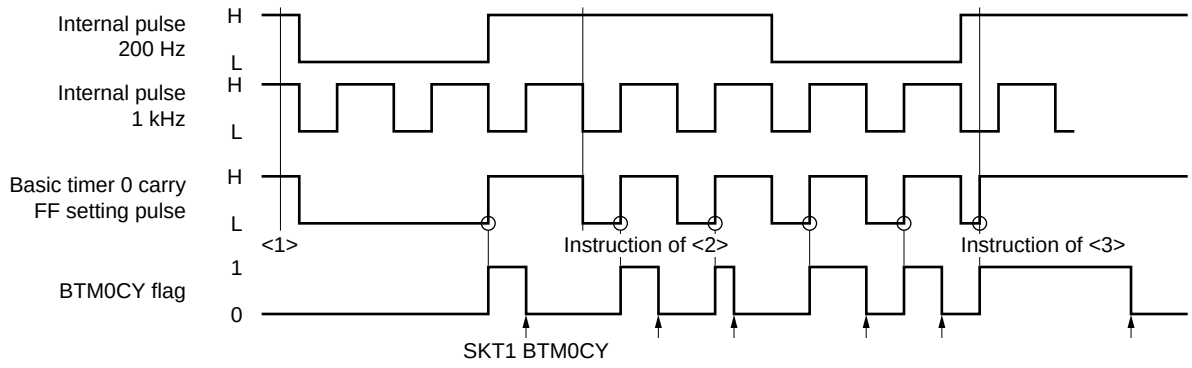
These four types of pulses operate independently of each other.

Therefore, if the timer time setting pulse is changed by the BTM0CK1 and BTM0CK0 flags, an error occurs as shown in the example below.

Example

```
; <1>
        INITFLG BTM0CK1, NOT BTM0CK0
                                ; Embedded macro
                                ; Sets basic timer 0 carry FF setting pulse to 200 Hz (5 ms)
        Processing A
; <2>
        SET2 BTM0CK1, BTM0CK0
                                ; Embedded macro
                                ; Sets basic timer 0 carry FF setting pulse to 1 kHz (1 ms)
        Processing A
; <3>
        INITFLG BTM0CK1, NOT BTM0CK0
                                ; Embedded macro
                                ; Sets the basic timer 0 carry FF setting pulse to 200 Hz (5 ms)
```

At this time, the basic timer 0 carry FF setting pulse is changed as follows:



As shown above, by changing the setting time of the basic timer 0 carry FF, the BTM0CY flag holds the previous status when the new pulse falls (<2> in the above figure). If the pulse rises, however, the BTM0CY flag is set to 1 (<3> in the figure).

In the above example, the pulse frequency is changed between 200 Hz (5 ms) and 1 kHz (1 ms). The same applies to change between 4 Hz (250 ms) and 10 Hz (100 ms).

Therefore, as shown in Figure 11-6, the error that may occur until the BTM0CY flag is set first after the basic timer 0 carry FF setting time has been changed is as follows:

$-t_{SET} < \text{error} < t_{CHECK}$

where,

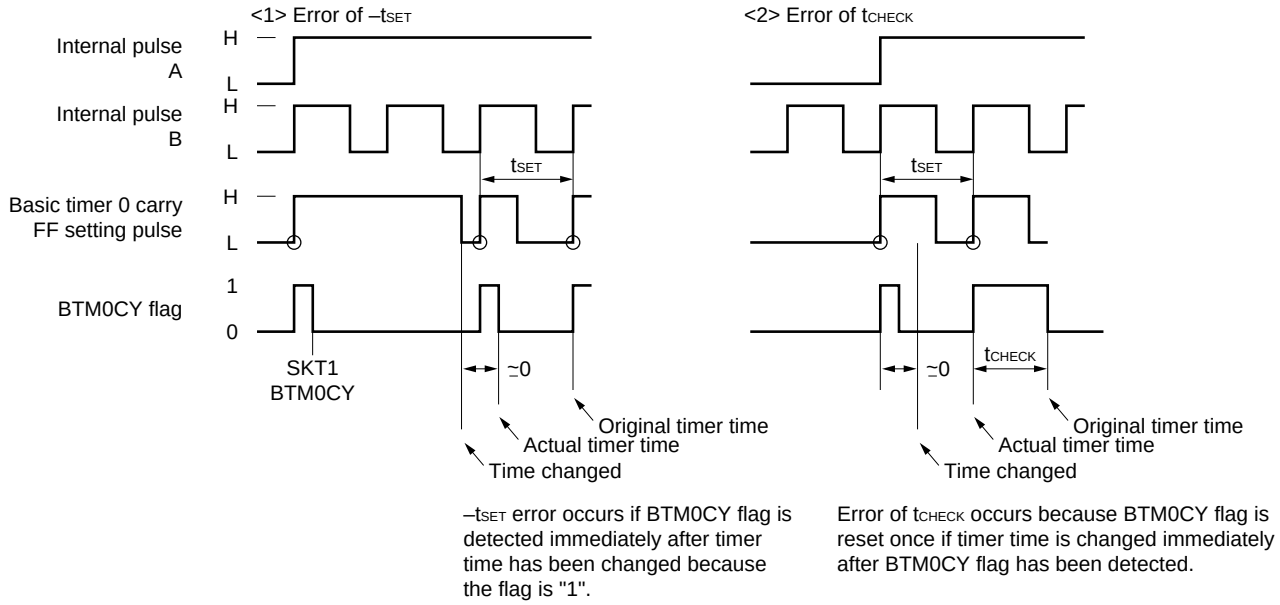
t_{SET} : new basic timer 0 carry FF setting time

t_{CHECK} : time required to detect BTM0CY flag

A phase difference is provided among the internal pulses of 4 Hz, 10 Hz, 200 Hz, and 1 kHz. Because this phase difference is shorter than the new pulse time, it is included in the above error.

For the phase difference of each pulse, refer to **11.4.5 Notes on using basic timer 1 interrupt**.

Figure 11-6. Errors When Basic Timer 0 Carry FF Setting Time Is Changed from A to B



–t_{SET} error occurs if BTM0CY flag is detected immediately after timer time has been changed because the flag is "1".

Error of t_{CHECK} occurs because BTM0CY flag is reset once if timer time is changed immediately after BTM0CY flag has been detected.

11.3.7 Notes on using basic timer 0 carry

The basic timer 0 carry is used not only as a timer but also as a reset synchronization signal when reset is effected by using the CE pin (CE reset).

If the next basic timer 0 carry FF setting pulse rises after the CE pin has gone high, CE reset is effected.

At this time, the following points must be noted.

(1) The sum of the timer updating processing time and the BTM0CY flag detection time interval must be shorter than the basic timer 0 carry FF setting time.

(2) When a program in which the timer always operates after power application (power-ON reset) regardless of CE reset is created, the timer must be adjusted each time the CE reset is effected.

(3) Detection of the BTM0CY flag takes precedence over the reset synchronization signal at CE reset. Therefore, if the two contend, CE reset is delayed once.

Above (1) through (3) are described in (a) through (c) below.

(a) Timer updating processing time and BTM0CY flag detection time interval

As described in **11.3.6**, the time interval t_{SET} at which the BTM0CY flag is detected must be shorter than the time interval at which the basic timer 0 carry FF is set.

Even if the time interval at which the BTM0CY flag is detected is short, if the timer updating processing time is long, the timer processing may not be correctly performed if CE reset is effected.

Therefore, the following condition must be satisfied:

$t_{CHECK} + t_{TIMER} < t_{SET}$

where,

t_{CHECK} : time interval at which BTM0CY flag is detected

t_{TIMER} : timer updating processing time

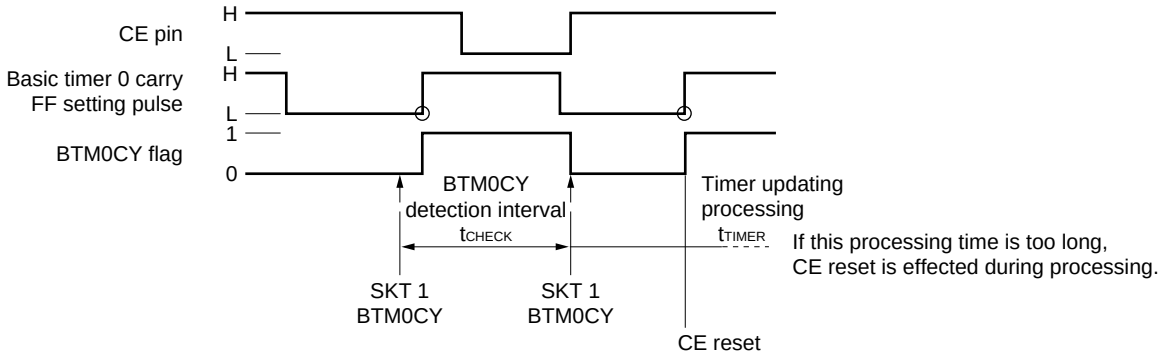
t_{SET} : time interval at which basic timer 0 carry FF is set

Here is an example:

Example Example of timer updating processing and BTM0CY flag detection time interval

```
START :                          ; Program address 0000H
        CLR2  BTM0CK1, BTM0CK0
                                 ; Embedded macro
                                 ; Sets basic timer 0 carry FF setting time to 100 ms
BTIMER :
        ; <1>
          SKT1 BTM0CY            ; Embedded macro
                                 ; Tests BTM0CY flag.
          BR     AAA             ; If it is “0”, branches to AAA.
          Timer updating
          BR     BTIMER
AAA :
          Processing A
          BR     BTIMER
```

Here is the timing chart of the above program:



(b) Adjusting basic timer 0 carry on CE reset

An example of adjusting the timer at CE reset is given below.

As shown in this example, the timer must be adjusted at CE reset “when the basic timer 0 carry FF is used for power failure detection and the basic timer 0 carry FF is used as a watch timer”.

The basic timer 0 carry FF is reset to 0 on the first power application (power-ON reset) and is disabled from being set until the BTM0CY flag is once read by using the “PEEK” instruction.

When the CE pin goes high, CE reset is effected in synchronization with the rising edge of the basic timer 0 carry FF setting pulse. At this time, the BTM0CY flag is set to 1 and the timer operation is started.

Therefore, by detecting the status of the BTM0CY flag on system reset (power-ON reset or CE reset), whether power-ON reset or CE reset has been effected can be judged (The BTM0CY flag is “0” when power-ON reset has been effected. It is “1” when CE reset has been effected) (power failure detection).

At this time a watch timer should continue its operation even at CE reset.

However, when the BTM0CY flag has been read to detect a power failure, the flag is reset to 0. Consequently, the set (1) status of the flag is overlooked once.

For this reason, the watch timer must be updated if CE reset has been detected as a result of power failure detection.

For further information on power failure detection, also refer to **13.6 Power Failure Detection**.

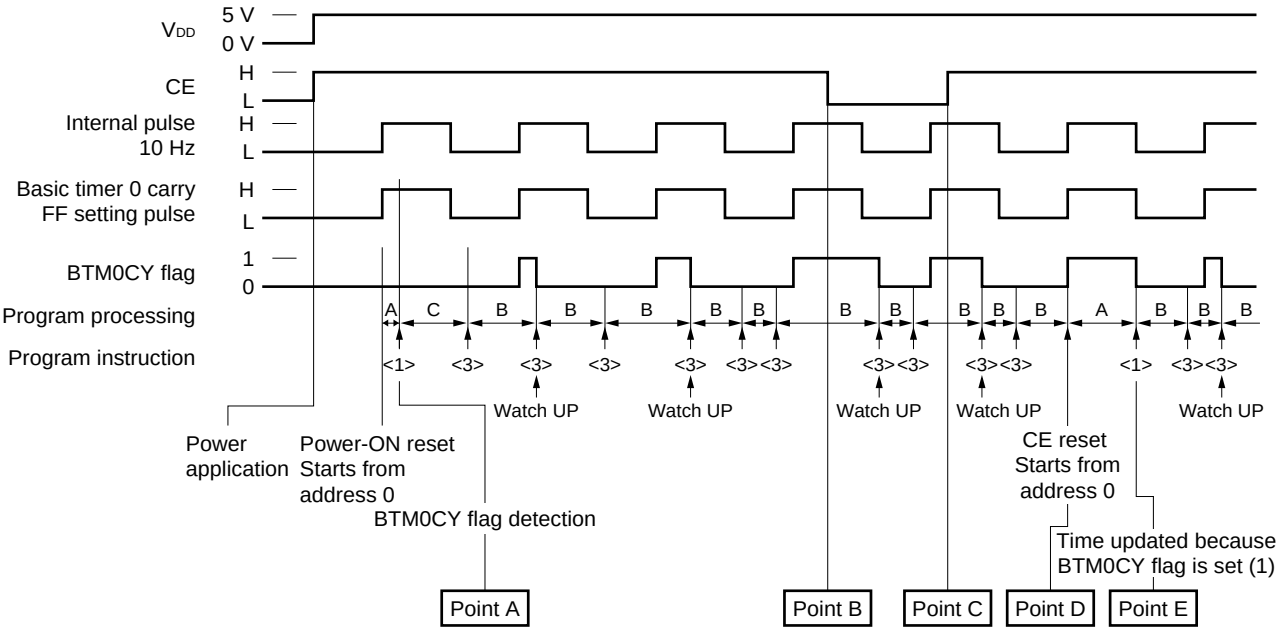
Example Adjusting timer on CE reset

To detect power failure and update watch by using basic timer 0 carry

```
START  :                                  ; Program address 0000H
           Processing A
       ; <1>
           SKT1    BTM0CY                 ; Embedded macro
                                          ; Tests BTM0CY flag
           BR      INITIAL                  If it is “0”, branches to INITIAL (power failure detection)
BACKUP :
       ; <2>
           Updates 100-ms watch           ; Adjusts timer because backup (CE reset) has been effected
LOOP   :
       ; <3>
           Processing B                   ; Updates watch by testing BTM0CY flag,
           SKF1    BTM0CY                 ; while executing processing B
           BR      BACKUP
           BR      LOOP
INITIAL :
           CLR2    BTM0CK1, BTM0CK0
                                          ; Embedded macro
                                          ; Because power failure (power-ON reset) occurs,
                                          ; sets basic timer 0 carry FF setting time to 100 ms,
                                          ; and executes processing C.
           Processing C
           BR      LOOP
```

Figure 11-7 is a timing chart illustrating the above program.

Figure 11-7. Timing Chart



As shown in this figure, the program is started from address 0000H at the rising edge of the internal 10-Hz pulse on application of supply voltage V_{DD} at first.

When the BTM0CY flag is detected next at point A, a power failure (power-ON reset) is detected because the BTM0CY flag is reset to 0 on power application.

Therefore, “processing C” is executed and the basic timer 0 carry FF setting pulse is set to 100 ms.

Because the contents of the BTM0CY flag have been read once at point A, the BTM0CY flag is set to 1 every 100 ms.

If the CE pin goes low at point B and then high at point C, the program counts up the watch while executing “processing B”, unless the clock stop instruction is executed.

Because the CE pin goes high at point C, CE reset is effected at point D where the next basic timer 0 carry FF setting pulse rises. Consequently, the program starts from address 0000H.

If the BTM0CY flag is detected at point E at this time, backup (CE reset) is assumed because the flag is set to 1.

As is evident from the figure, unless the watch is updated by 100 ms at point E, the watch is delayed by 100 ms each time CE reset is effected.

If processing A takes 100 ms or longer when a power failure is detected at point E, setting of the BTM0CY flag is overlooked two times. Therefore, processing A must be executed shorter than 100 ms.

The above description also applies when 250 ms, 5 ms, or 1 ms is selected as the basic timer 0 carry FF setting pulse.

Therefore, the BTM0CY flag must be detected in order to detect a power failure less than the basic timer 0 carry FF setting time after the program has been started from address 0000H.

(c) If detection of BTM0CY flag collides with CE reset

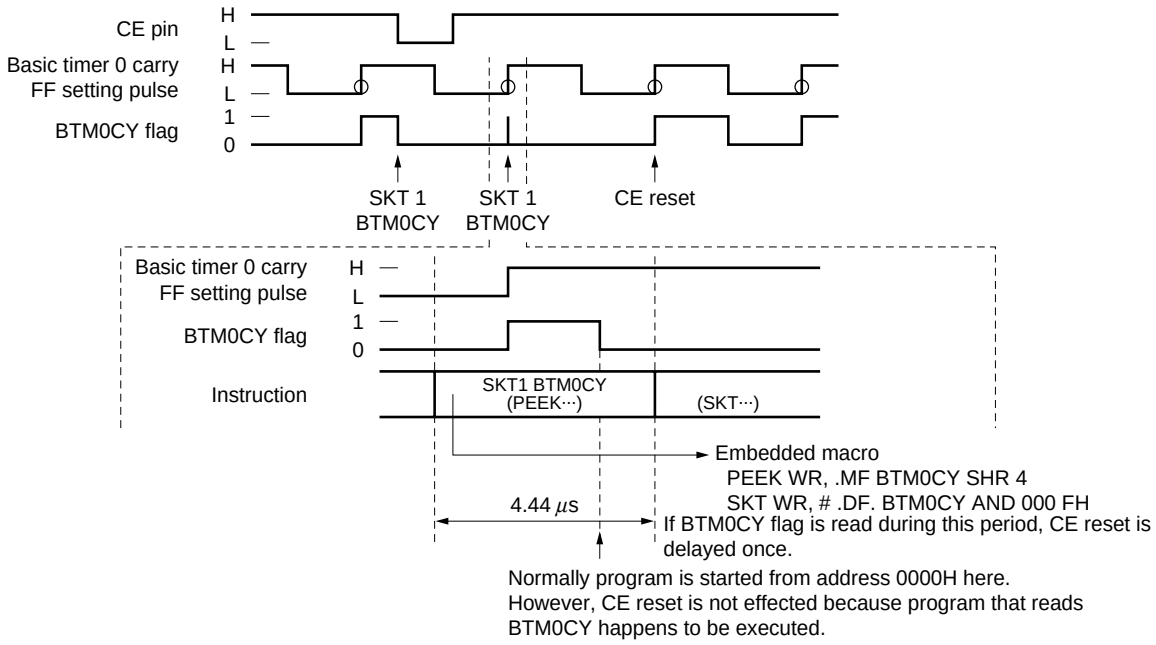
As described in **(b)**, CE reset is effected as soon as the BTM0CY flag is set to 1.

At this time if an instruction that reads the BTM0CY flag happens to be executed at the same time as CE reset, the BTM0CY flag read instruction takes precedence.

Therefore, if setting of the BTM0CY flag (rising of the basic timer 0 carry FF setting pulse) after the CE pin has gone high collides with the BTM0CY flag read instruction, CE reset is effected "when the BTM0CY flag is set next time".

This operation is illustrated in Figure 11-8.

Figure 11-8. Operation If CE Reset Collides with BTM0CY Flag Read Instruction



Therefore, if the program that cyclically detects the BTM0CY flag and in which the detection time interval of the BTM0CY flag coincides with the BTM0CY flag setting time, CE reset is never effected.

Remember the following point:

Because one instruction cycle is 4.44 μs (1/225 kHz), a program, for example, that detects the BTM0CY flag once each time 255 instructions have been executed reads the BTM0CY flag every 1 ms (= 4.44 μs × 225).

At this time, once setting and detection of the BTM0CY flag have coincided, CE reset is never effected, regardless of whether the 1-, 5-, 100-, or 250-ms timer time setting pulse is selected.

Therefore, **do not create a program with a cycle that satisfies the following condition:**

$$\frac{t_{SET} \times 225}{X} = n \text{ (n: natural number)}$$

where,

t_{SET}: BTM0CY flag setting time

X : step X of instruction in which BTM0CY flag is read

Here is an example of a program that satisfies the above condition. Do not create such a program.

Example

```
            ┌──────────────┐
            │ Processing A │
            └──────────────┘
        SET2 BTM0CK1, BTM0CK0
                                    ; Embedded macro
                                    ; Sets basic timer 0 carry FF setting pulse to 1 ms
LOOP :
        ;<1>
            SKT1  BTM0CY            ; Embedded macro
            BR    BBB
AAA :
            ┌──────────────┐
            │  221 steps   │
            └──────────────┘
            BR    LOOP
BBB :
            ┌──────────────┐
            │  221 steps   │
            └──────────────┘
            BR    LOOP
```

In this example, the BTM0CY flag read instruction in <1> is repeated each time 225 instructions have been executed. If the BTM0CY flag happens to be set when instruction <1> is executed, CE reset is not effected after that.

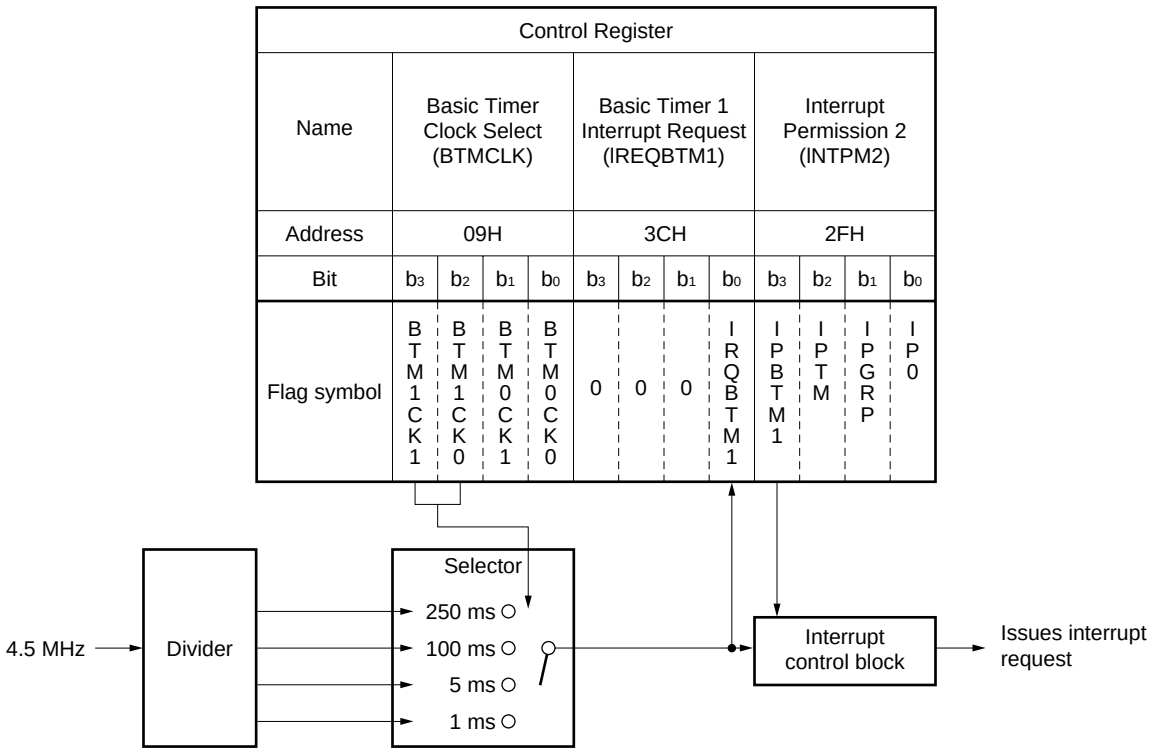
11.4 Basic Timer 1 Interrupt

11.4.1 Configuration of basic timer 1 interrupt block

Figure 11-9 shows the configuration of the basic timer 1 interrupt block.

As shown in this figure, the basic timer 1 interrupt block consists of a divider, a selector, and an interrupt control block.

Figure 11-9. Configuration of Basic Timer 1 Interrupt Block



11.4.2 Function of basic timer 1 interrupt block

The basic timer 1 interrupt block issues an interrupt request at the falling edge of the basic timer 1 interrupt pulse set by the higher 2 bits (BTM1CK1 and BTM1CK0 flags) of the basic timer clock select register.

The basic timer 1 interrupt request corresponds to the IRQBTM1 flag of the basic timer 1 interrupt request register (IREQBTM1: RF address 3CH) on a one-to-one basis, and the IRQBTM1 flag is set to 1 when the basic timer 1 interrupt request is issued. When the basic timer 1 interrupt pulse falls, therefore, the IRQBTM1 flag is set to 1.

So that the basic timer 1 interrupt may occur, the interrupt request must be issued, the "EI" instruction which enables all the interrupts must be issued, and the basic timer 1 interrupt must be enabled, as described in **10. INTERRUPT**.

To enable the basic timer 1 interrupt, set the IPBTM1 flag of the interrupt permission 2 register (INTPM2: RF address 2FH) to 1.

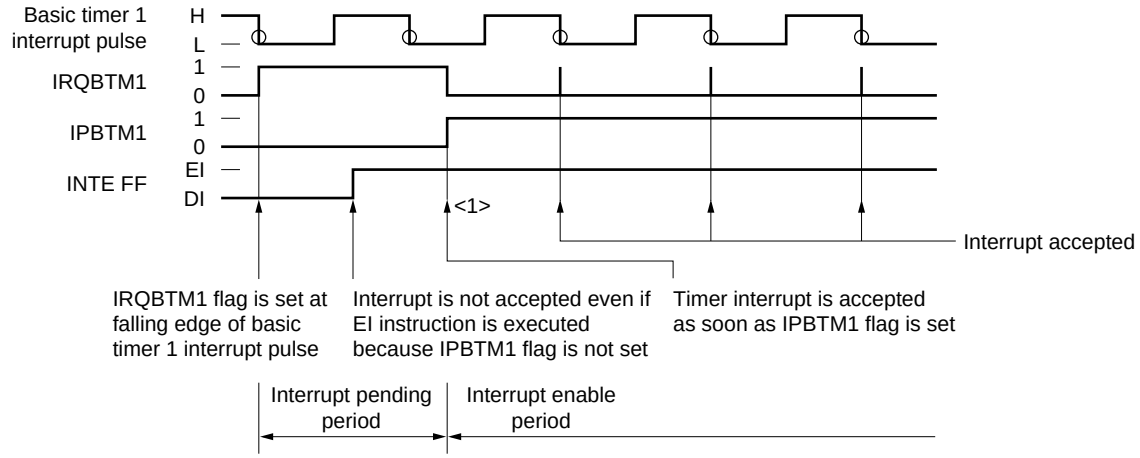
Therefore, the basic timer 1 interrupt is accepted if the IRQBTM1 flag is set to 1 when the "EI" instruction has been executed and the IPBTM1 flag has been set to 1.

When the basic timer 1 interrupt has been accepted, the program flow is transferred to program memory address 0003H.

The IRQBTM1 flag is reset to 0 when the interrupt has been accepted.

Figure 11-10 shows the relation between the basic timer 1 interrupt pulse and IRQBTM1 flag.

Figure 11-10. Relation between Basic Timer 1 Interrupt Pulse and IRQBTM1 Flag



The point that must be remembered here is that the basic timer 1 interrupt is accepted when the "EI" instruction is executed and the IPBTM1 flag is set, as shown in <1> in Figure 11-10, once the IRQBTM1 flag is set when the timer interrupt is disabled by the "DI" instruction or IPBTM1 flag.

In this case, the interrupt request is cleared if "0" is written to the IRQBTM1 flag.

If "1" is written to the IRQBTM1 flag, the operation is equivalent to issuance of the interrupt request.

When the basic timer 1 interrupt is accepted, one level of the stack is used.

The contents of the window register (WR), bank register (BANK), general register pointer (RP), and program status word (PSWORD) are automatically saved.

To return from the interrupt processing routine, use the dedicated instruction "RETI".

For details, refer to **3. ADDRESS STACK (ASK)** and **10. INTERRUPT**.

For the configuration and function of the basic timer clock select register, refer to **11.3.3**.

11.4.3 and **11.4.4** below describe an example of using the basic timer 1 interrupt and an error of the basic timer 1 interrupt.

For the relation between the basic timer 1 interrupt and other interrupts (such as INT_0 pin, INT_1 pin, 12-bit timer, serial interface 0, and frequency counter interrupts), refer to **10. INTERRUPT**.

11.4.3 Example of timer using basic timer 1 interrupt

Example

```
M1          MEM         0.10H          ; 80-ms counter
BTIMER1     DAT         0003H          ; Defines symbol of basic timer interrupt vector address

            BR          START          ; Branches to START
ORG         BTIMER1                    ; Program address (0003H)
            ADD         M1, #0001B     ; Adds 1 to M1
            SKT1        CY             ; Tests CY flag
            BR          EI_RETI        ; Returns if carry does not occur
            Processing A
EI_RETI:
            EI
            RETI
START:
            INITFLG BTM1CK1, NOT BTM1CK0
                                       ; Embedded macro
                                       ; Sets basic timer 1 interrupt pulse to 5 ms
            MOV         M1, #0000B     ; Clears contents of M1 to 0
            SET1        IPBTM1         ; Enables basic timer 1 interrupt
            EI                         ; Enables all interrupts
LOOP:
            Processing B
            BR          LOOP
```

This program executes processing A every 80 ms.

The points to be noted in this case are that the DI status is automatically set when the interrupt has been accepted, and that the IRQBTM1 flag is set to 1 even in the DI status.

If processing A takes 5 ms or longer, therefore, the interrupt is accepted as soon as execution is returned by the “RETI” instruction, and as a result, processing B is not executed.

11.4.4 Error of basic timer 1 interrupt

As described in **11.4.2**, the interrupt is accepted each time the basic timer 1 interrupt pulse falls if the EI instruction has been executed and the basic timer 1 interrupt has been enabled.

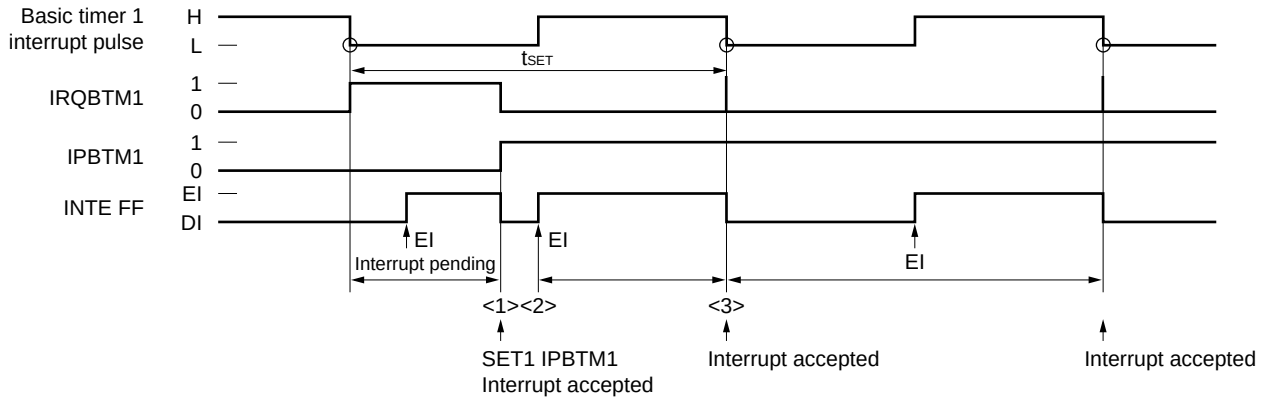
Therefore, a timer error only occurs when the basic timer 1 interrupt is used in the following cases:

(1) When the first interrupt is accepted after the basic timer 1 interrupt has been enabled
(2) When the first interrupt is accepted after the time of the basic timer 1 interrupt pulse is changed
(3) When the IRQBTM1 flag is written

Figure 11-11 shows an error that may occur in each of the above cases.

Figure 11-11. Error of Basic Timer 1 Interrupt (1/2)

(a) When basic timer 1 interrupt is enabled



When basic timer 1 interrupt is enabled by setting the IPBTM1 flag in point <1> above, the interrupt is immediately accepted.

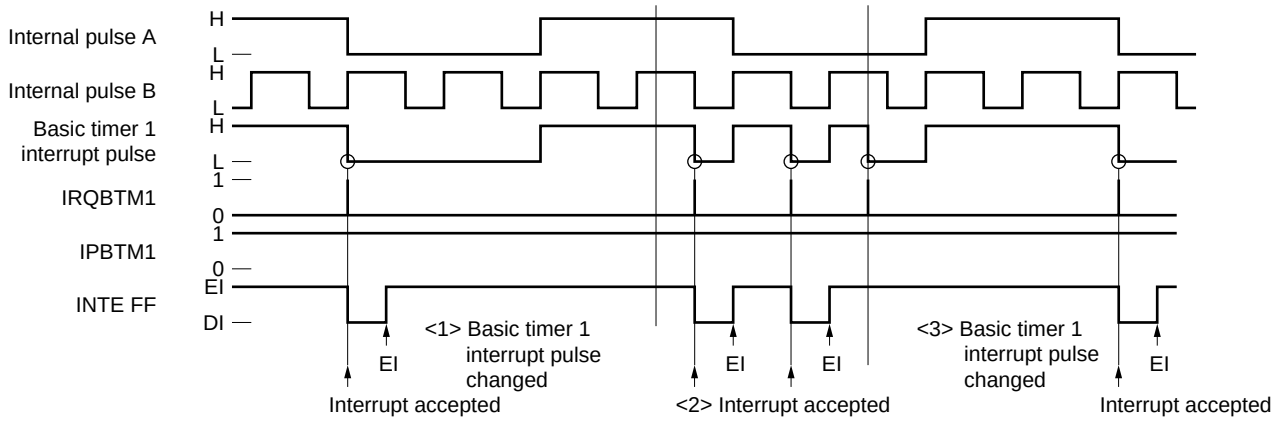
The error at this time is $-t_{SET}$.

If the interrupt is subsequently enabled by the “EI” instruction at point <2>, the interrupt occurs at the falling edge of the basic timer 1 interrupt pulse at point <3>.

At this time, the relation between $-t_{SET}$ and error is as follows: $-t_{SET} < \text{error} < 0$

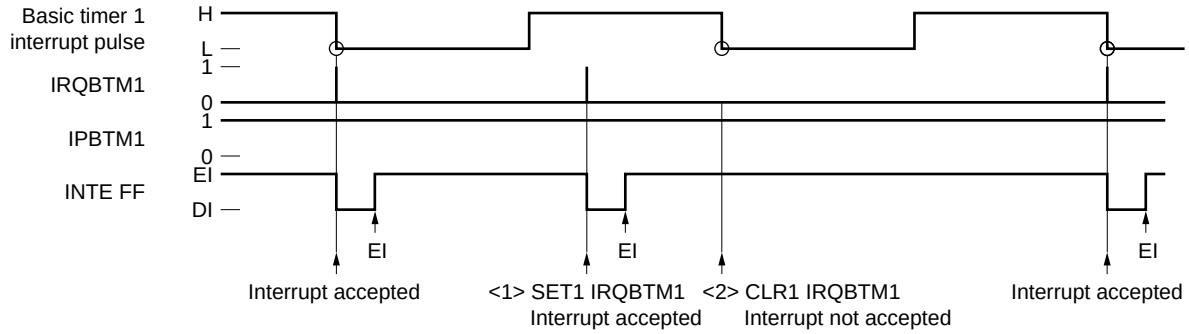
Figure 11-11. Error of Basic Timer 1 Interrupt (2/2)

(b) When basic timer 1 interrupt pulse is changed



Because the basic timer 1 interrupt pulse does not fall even if basic timer 1 interrupt pulse is changed to B in <1>, the interrupt is accepted in <2>. Because the basic timer 1 interrupt pulse falls if the basic timer 1 interrupt pulse is changed to A in <3>, the interrupt is immediately accepted.

(c) When IRQBTM1 flag is manipulated



If the IRQBTM1 flag is set in <1>, the interrupt is immediately accepted.
If resetting the IRQBTM1 flag collides with the falling of the basic timer 1 interrupt pulse in <2>, the interrupt is not accepted.

11.4.5 Notes on using basic timer 1 interrupt

When creating a program, in which the basic timer 1 always operates for a specific time after once power has been applied (power-ON reset) such as a watch program, using the basic timer 1 interrupt the basic timer 1 interrupt processing time must be completed in a specific time.

This is described by taking the following example.

Example

```
M1          MEM         0.10H          ; 1-ms counter
BTIMER1     DAT         0003H          ; Symbol definition of basic timer interrupt vector address

            BR          START          ; Branches to START
ORG         BTIMER1                    ; Program address (0003H)
            ADD         M1, #0100B     ; Adds 0100B to M1
            SKT1        CY             ; Watch processing if carry occurs
            BR          EI_RETI        ; Returns if carry does not occur
            ; <1>
            Watch processing
EI_RETI:
            EI
            RETI
START:
            INITFLG NOT BTM1CK1, BTM1CK0, NOT BTM0CK1, NOT BTM0CK0
                                       ; Embedded macro
                                       ; Sets basic timer 1 interrupt time to 250 ms and basic timer 0 carry
                                       ; FF setting time to 100 ms
            SET1        IPBTM1         ; Embedded macro
                                       ; Enables basic timer 1 interrupt
            EI                         ; Enables all interrupts
LOOP:
            Processing A
            BR          LOOP
```

In this example, watch processing <1> is executed every 1 second while processing A is executed.

If the CE pin goes high as shown in Figure 11-12 (a), CE reset is effected in synchronization with the rising of the basic timer 0 carry FF setting pulse.

If the basic timer 1 interrupt request is issued at the same time as the setting of the basic timer 0 carry FF, CE reset takes precedence.

When CE reset is effected, the basic timer 1 interrupt request (IRQBTM1 flag) is reset. Consequently, timer processing is not performed once.

To prevent this, actually there is a delay between the “rising of the basic timer 0 carry FF setting pulse” and “falling of the basic timer 1 interrupt pulse”, as shown in Figure 11-12 (b).

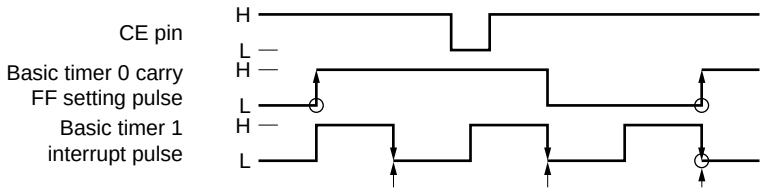
Therefore, as shown in Figure 11-12 (b), the basic timer 1 interrupt occurs without fail even if CE reset is effected, if the watch processing is performed within 10 ms.

Because four types of basic timer 0 carry FF and basic timer 1 interrupt time setting pulses, 4 Hz (250ms), 10 Hz (100 ms), 200 Hz (5 ms), and 1 kHz (1 ms), can be set separately, a time difference is provided as shown in Figure 11-13 and Table 11-1.

If it is necessary to enable the basic timer 1 interrupt even at CE reset, the basic timer 1 interrupt processing must be completed within the delay time of the pulse as shown in Figure 11-13.

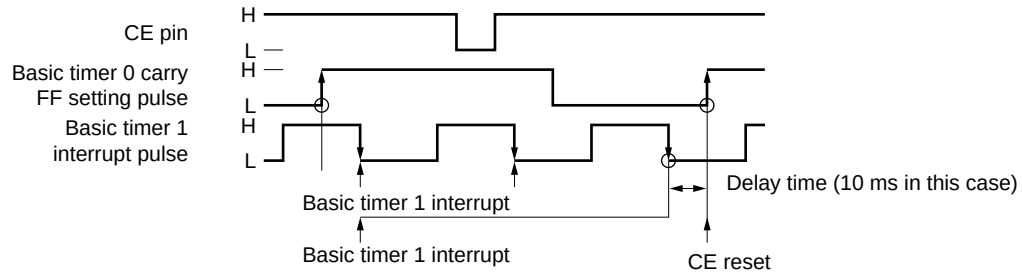
Figure 11-12. Timing Chart

(a)



Because basic timer 0 carry FF setting pulse rises, CE reset is effected here, and basic timer 1 interrupt is missed once.

(b)



CE reset

Because there is a delay of 10 ms between the falling of the basic timer 1 interrupt pulse and the rising of the basic timer 0 carry FF setting pulse, if the basic timer 1 interrupt processing is performed within 10 ms, the timer processing is normally executed even if CE reset is effected.

Figure 11-13. Time Difference between Basic Timer 0 Carry FF Setting Pulse and Basic Timer 1 Interrupt Pulse

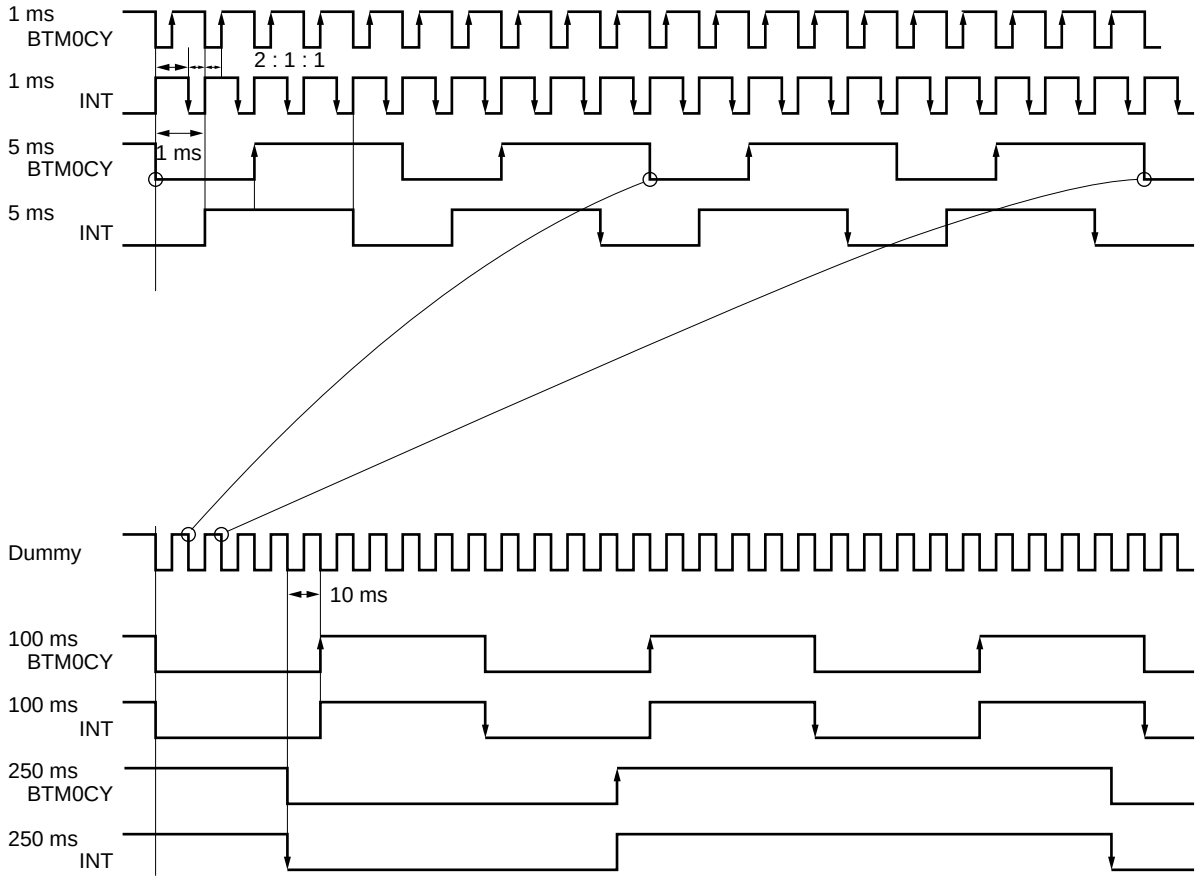
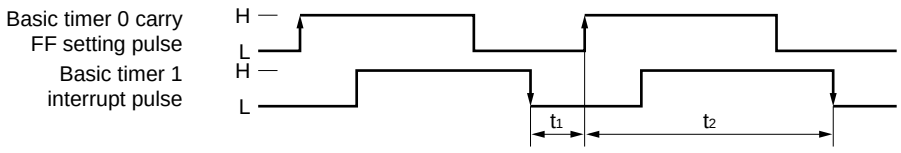


Table 11-1. Time Difference between Rising Edge of Basic Timer 0 Carry FF Pulse and Falling Edge of Basic Timer 1 Interrupt Pulse

Internal Pulse		Minimum Value of Time Difference (See Figure below)	
Basic Timer 0 Carry	Basic Timer 1 Interrupt	t_1	t_2
1 ms	1 ms	666 μs	333 μs
1 ms	5 ms	333 μs	666 μs
1 ms	100 ms	333 μs	666 μs
1 ms	250 ms	333 μs	666 μs
5 ms	1 ms	333 μs	666 μs
5 ms	5 ms	3 ms	2 ms
5 ms	100 ms	2 ms	3 ms
5 ms	250 ms	2 ms	3 ms
100 ms	1 ms	333 μs	666 μs
100 ms	5 ms	1 ms	4 ms
100 ms	100 ms	50 ms	50 ms
100 ms	250 ms	10 ms	40 ms
250 ms	1 ms	333 μs	666 μs
250 ms	5 ms	1 ms	4 ms
250 ms	100 ms	40 ms	10 ms
250 ms	250 ms	100 ms	150 ms



11.5 12-Bit Timer

11.5.1 Configuration of 12-bit timer

The 12-bit timer consists of a clock select block, 12-bit timer mode control block, count block, overflow detection block, and interrupt control block, as shown in Figure 11-1.

11.5.2 Functional outline of 12-bit timer

The count block of the 12-bit timer performs counting operation each time the time selected by the clock select block.

If the count value of the count block reaches a specific value, an interrupt request is issued.

The function of each block is outlined below.

(1) Clock select block

This block generates the count clock of the 12-bit timer.

The count clock is selected by the timer/counter clock select register (TMCLK: RF address 0CH).

This block consists of a divider and selector.

(2) 12-bit timer mode control block

This block controls the mode of the 12-bit timer.

It can control starting and resetting the timer/counter, and select a modulo count mode or free-run count mode.

These control operations are performed by using the 12-bit timer mode control register (TMMDCONT: RF address 0EH).

(3) Count block

The count block counts the count clocks of the timer counter (TMC: peripheral address 47H) and issues an interrupt request when the value of the timer/counter coincides with a predetermined value of the timer modulo register (TMM: peripheral address 46H).

(4) Overflow detection block

The overflow detection block detects an overflow in the timer/counter in the free-run count mode.

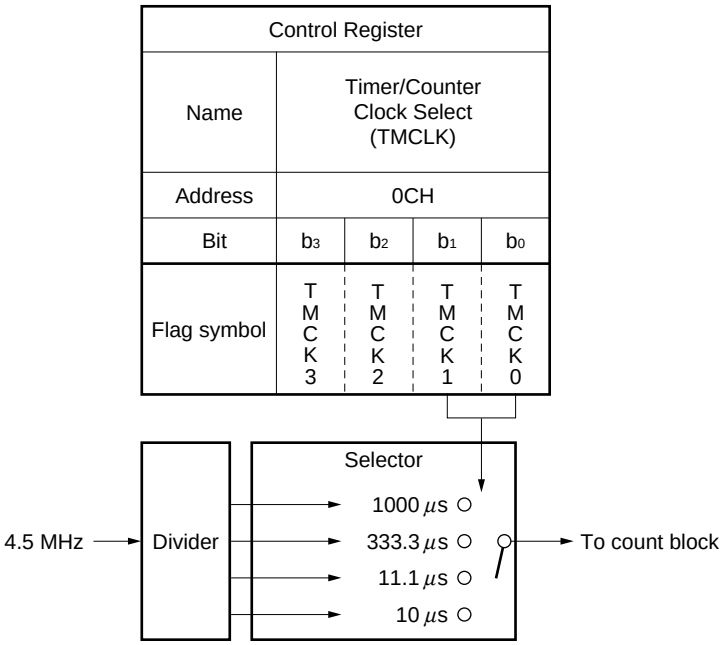
To detect the overflow, the timer/counter overflow detect register (TMOVDET: RF address 0DH) is used.

11.5.3 Divider and selector

(1) Configuration of divider and selector

Figure 11-14 shows the configuration of the divider and selector.

Figure 11-14. Divider and Selector Configuration



(2) Functions of divider and selector

The divider and selector divides the system clock (4.5 MHz) and generates the count clock of the 12-bit timer.

Four types of the count clock can be selected for different clock frequencies by the timer/counter clock select register.

The configuration and function of the timer/counter clock select register are shown below.

Configuration and function of timer/count clock select register (TMCLK)

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Timer/counter clock select register (TMCLK)	TMCK3	TMCK2	TMCK1	TMCK0	0CH	R/W

TMCK1	TMCK0	Timer Clock Cycle (Frequency)	Measurable Time Range
0	0	1 ms (1 kHz)	1 ms - 4095 ms
0	1	333.3 μs (3 kHz)	333.3 μs - 1365 ms
1	0	10 μs (100 kHz)	10 μs - 40.95 ms
1	1	11.1 μs (90 kHz)	11.1 μs - 45.5 ms

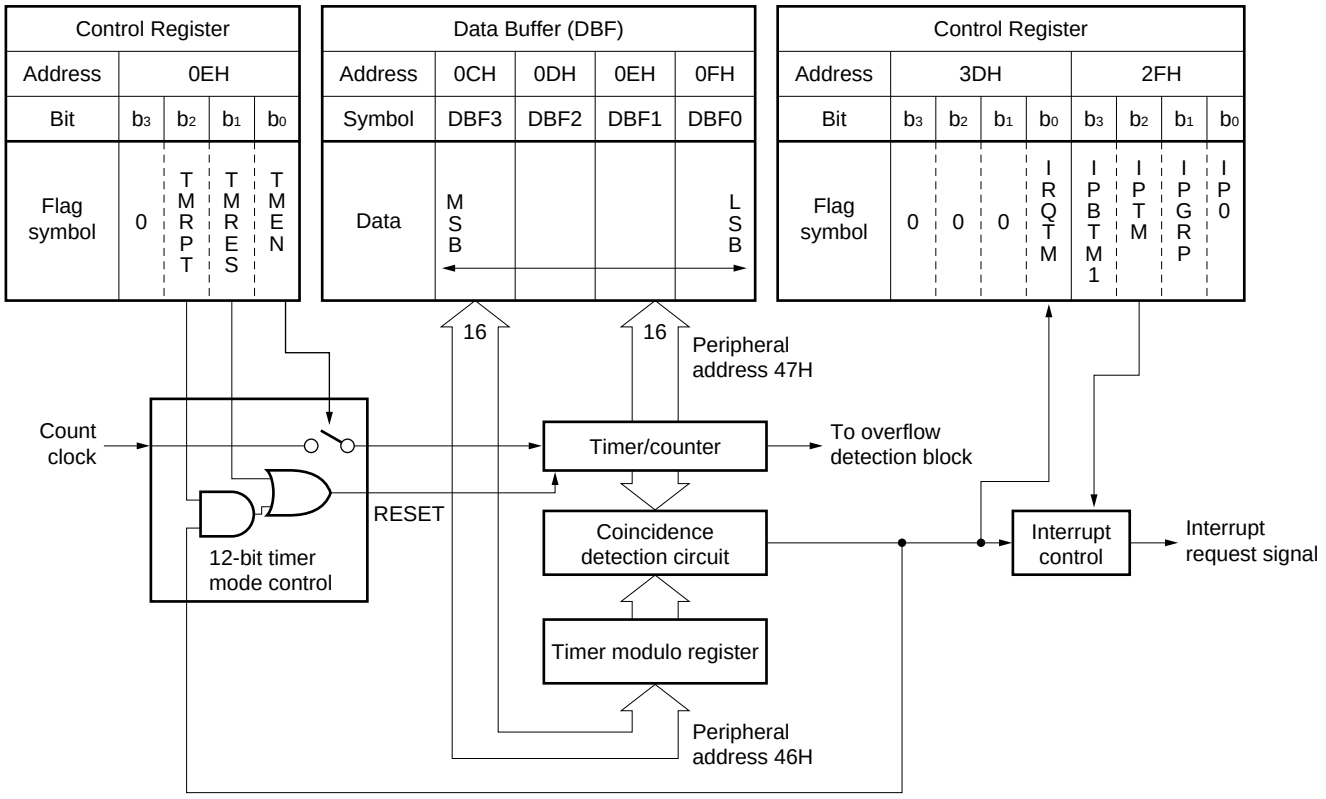
Fixed to "0"

On reset	TMCK3	TMCK2	TMCK1	TMCK0
Power-ON	0	0	0	0
Clock stop			0	0
CE			Retained	

11.5.4 12-bit timer mode control block and count block

Figure 11-15 shows the configuration of the 12-bit timer mode control block and count block.

Figure 11-15. Configuration of 12-Bit Timer Mode Control Block and Count Block



(1) Function of 12-bit timer mode control block

The 12-bit timer mode control block controls starting and resetting of the timer/counter and selects an operation mode of the 12-bit timer.

The mode is controlled by the 12-bit timer mode control register.

How each mode control operation is performed is described below.

(a) Start control

The timer/counter is started by using the TMEN flag.

(b) Reset control

The timer/counter is reset by using the TMRES flag.

The timer/counter is also reset if a coincidence is detected by the coincidence detection circuit of the count block in the modulo count mode.

(c) Mode control

The operation mode of the 12-bit timer is set by the TMRPT flag.

This flag selects two types of modes: free-run count mode and modulo count mode.

In the free-run count mode, the contents of the timer/counter is not reset but continues counting even after the value of the timer/counter has coincided with the value of the timer modulo register.

In the modulo count mode, the contents of the timer/counter is reset and then continues counting after the value of the timer/counter has coincided with the value of the modulo register.

The function and configuration of the 12-bit timer mode control register are shown below.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
12-bit timer mode control register (TMMDCONT)	0	TMRPT	TMRES	TMEN	0EH	R/W

TMEN — Starts or stops timer/counter
- 0: Stops timer/counter
- 1: Operates timer/counter

TMRES — Controls count value of timer/counter[Note]
- 0: Not affected
- 1: Resets timer/counter

TMRPT — Sets mode of 12-bit timer
- 0: Free-run count mode
- 1: Modulo count mode

b_3 — Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop		0	0	0
	CE		Retained		

Note The TMRES flag is always “0” when read.

(2) Count block

When the count clock is supplied to the timer/counter as shown in Figure 11-15, the timer/counter starts counting. When the value of the timer/counter coincides with the contents of the timer modulo register, an interrupt request signal is output.

In the modulo count mode, the timer/counter is reset and then continues counting.

The configuration and function of the timer counter and timer modulo register are shown below.

(a) Configuration and function of timer/counter

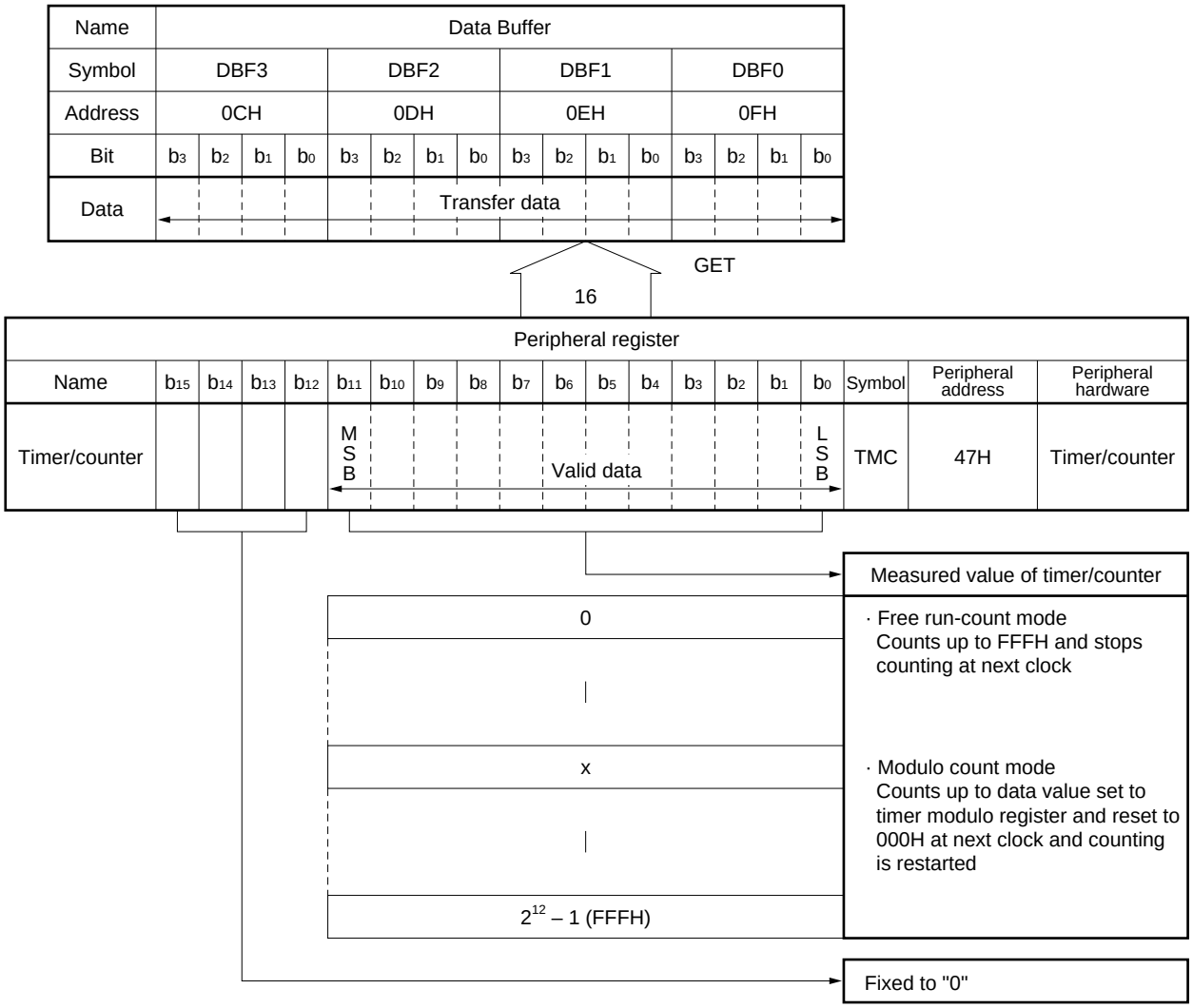
The timer/counter counts the count clock.

In the free-run count mode, the timer/counter counts up to FFFH, sets the timer counter overflow detect flag to 1 at the next clock, and stops counting.

The configuration and function of the timer/counter are shown below.

Because the timer/counter is of 12-bit configuration, the lower 12 bits of the data buffer are valid. The higher 4 bits are always "0" when they are read.

The timer/counter can be read even during the counting operation. However, the data read at this time may not be accurate. For details, refer to **11.5.7 Error of 12-bit timer**.



(b) Configuration and function of timer modulo register

The timer modulo register sets reference data to issue an interrupt request when the count value of the timer/counter coincides with its contents.

Because this register is a 12-bit register, a value of 1 to 4095 can be set.

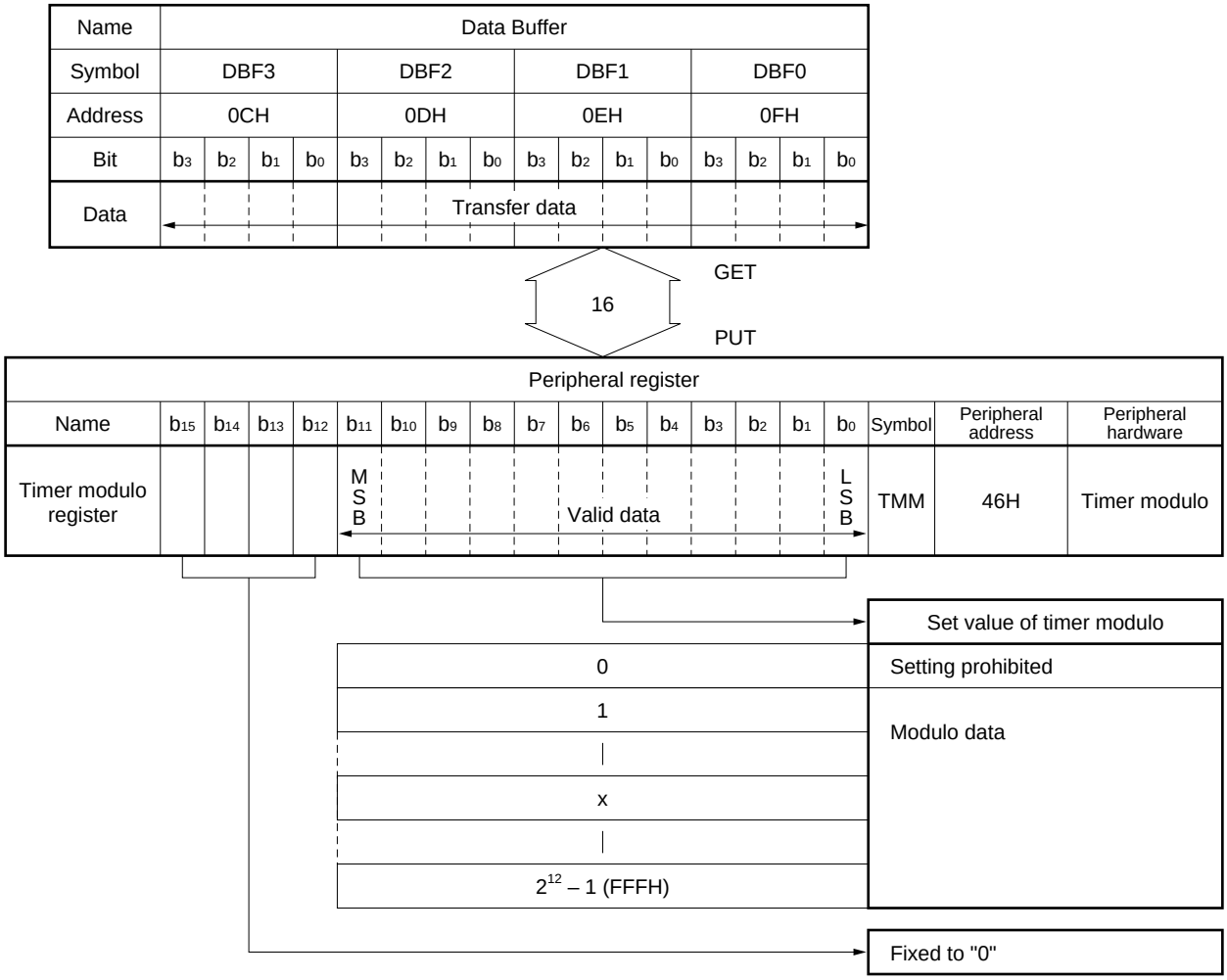
The coincidence detection circuit detects coincidence between the value set to the timer modulo register and the count value of the timer/counter, and issues an interrupt request.

When the interrupt request is issued, the IRQTM flag is set. If the IPTM flag is set in the EI status, the interrupt is accepted, and the program flow is transferred to interrupt vector address 0004H.

If data coincidence is detected in the modulo count mode (TMRPT flag = 1), the contents of the timer/counter are reset.

The configuration and function of the timer modulo register are illustrated below.

Because the timer modulo register is of 12-bit configuration, the lower 12 bits of the data buffer are valid. The higher 4 bits can be any value when they are written. These bits are always “0” when read.



11.5.5 Overflow detection block

Figure 11-16 shows the configuration of the overflow detection block.

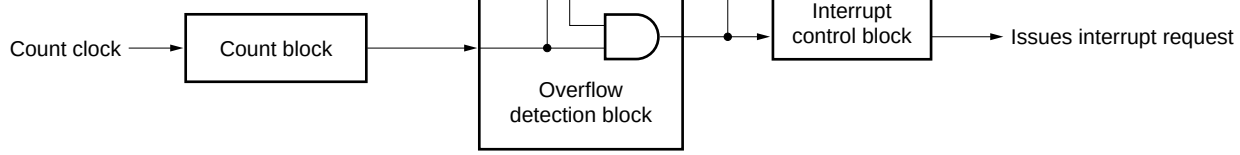
The overflow detection block detects an overflow in the timer/counter.

When an overflow is detected, the TMOVF flag of the timer/counter overflow detect register is set.

When this flag has been set, the counting operation is stopped.

Figure 11-16. Configuration of Overflow Detection Block

Control Register																
Name	Timer/Counter Overflow Detect (TMOVDET)				Interrupt Group Select (IGRPSELR)				Group Interrupt Request (IREQGRP)				Interrupt Permission 2 (INTPM2)			
Address	0DH				0FH				3EH				2FH			
Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Flag symbol	0	0	0	TMOVF	0	0	0	TGRPSL	INT1	0	0	IRQGRP	IPBTM1	IPTM	IPGRP	IP0



As shown in Figure 11-16, the overflow of the timer/counter can be used as an interrupt source by using the interrupt group select register (IGRPSELR: RF address 0FH).

The configuration and function of the timer/counter overflow detect register and interrupt group select register are shown below.

(1) Configuration and function of timer/counter overflow detect register

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Timer/counter overflow detect register (TMOVDET)	0	0	0	TMOVF	0DH	R

Detects overflow of timer/counter	
0	Does not overflow
1	Overflows
Fixed to "0"	

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	↓	↓	↓	0
	CE	↓	↓	↓	Re-tained

(2) Configuration and function of interrupt group select register

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Interrupt group select register (IGRPSELR)	0	0	0	IGRPSL	0FH	R/W

Sets interrupt source of IRQGRP	
0	Issues interrupt request at rising or falling edge of INT_1 pin
1	Issues interrupt request on overflow of timer/counter
Fixed to "0"	

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	↓	↓	↓	0
	CE	↓	↓	↓	0

11.5.6 Example of using 12-bit timer

Here are examples of using the 12-bit timer:

Example 1. Modulo count mode

```
TMINT   DAT     0004H       ; Symbol definition of 12-bit timer/counter interrupt vector address
        BR      START
ORG     TMINT               ; Program address (0004H)
        Processing A
        EI
        RETI
START:
        INITFLG TMCK1, NOT TMCK0
                            ; Sets count clock to 10 μs
        MOV     DBF2, #50 SHR 8 AND 0FH
        MOV     DBF1, #50 SHR 4 AND 0FH
        MOV     DBF0, #50 AND 0FH
        PUT     TMM, DBF
        SET1    IPTM
        EI
        SET3    TMRPT, TMRES, TMEN
LOOP:
        Main processing
        BR      LOOP
```

This program executes processing A every 500 μs.

However, processing A must be completed shorter than 500 μs.

Example 2. Free-run count mode

```
            BR      Start
                ⋮
Start
            INITFLG TMCK1, NOT TMCK0
                                ; Sets count clock to 10 μs
            INITFLG NOT TMRPT, TMRES, TMEN
            Processing A
            SKF1    TMOVF
            BR      Overflows
            GET     DBF, TMC
                ⋮
Overflows       :
                ⋮
```

This program measures the time required to perform processing A. The time can be set from 10 μs to 40950 μs (in the above example, the time 40950 μs or longer cannot be measured, and therefore, execution branches by software to another routine).

This program is used to measure the pulse width of a remote controller signal.

To issue an interrupt request at fixed time intervals, the modulo count mode is convenient. To measure a total time, however, the free-running count mode is convenient.

11.5.7 Error of 12-bit timer

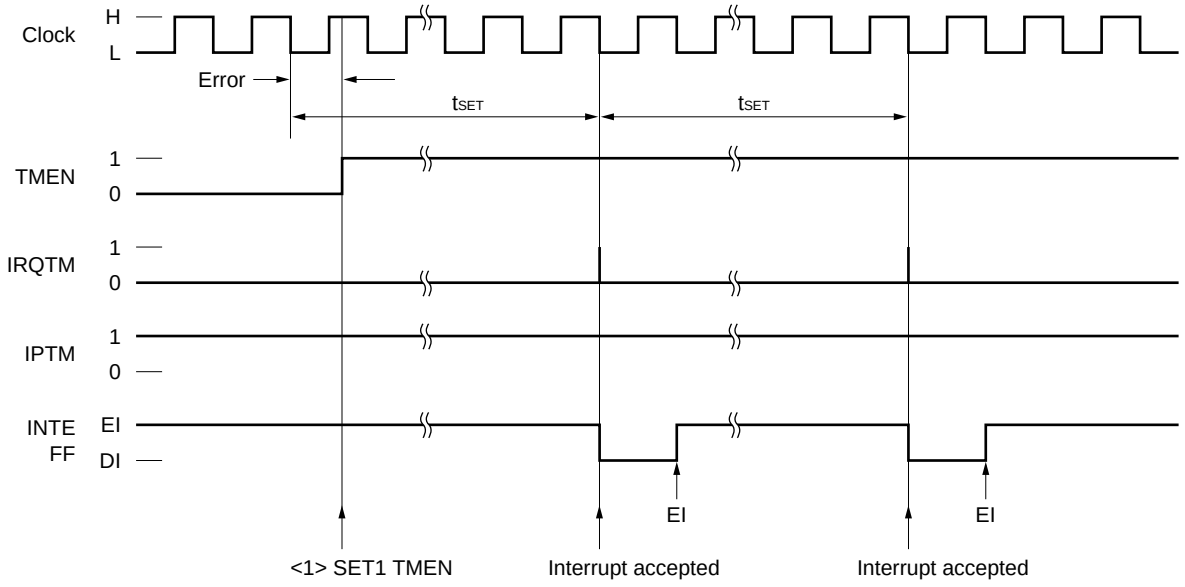
An error of the interrupt by the timer occurs in the following cases:

(1) When the TMEN flag is set to 1
(2) When the timer/counter is reset
(3) When the data of the timer/counter is read during counting operation

Figure 11-17 illustrates the error that may occur during operation.

Figure 11-17. Error of 12-Bit Timer (1/2)

(1) When TMEN flag is set to 1



Remark f_{SET} = (Data set to timer modulo register) × (Count clock)

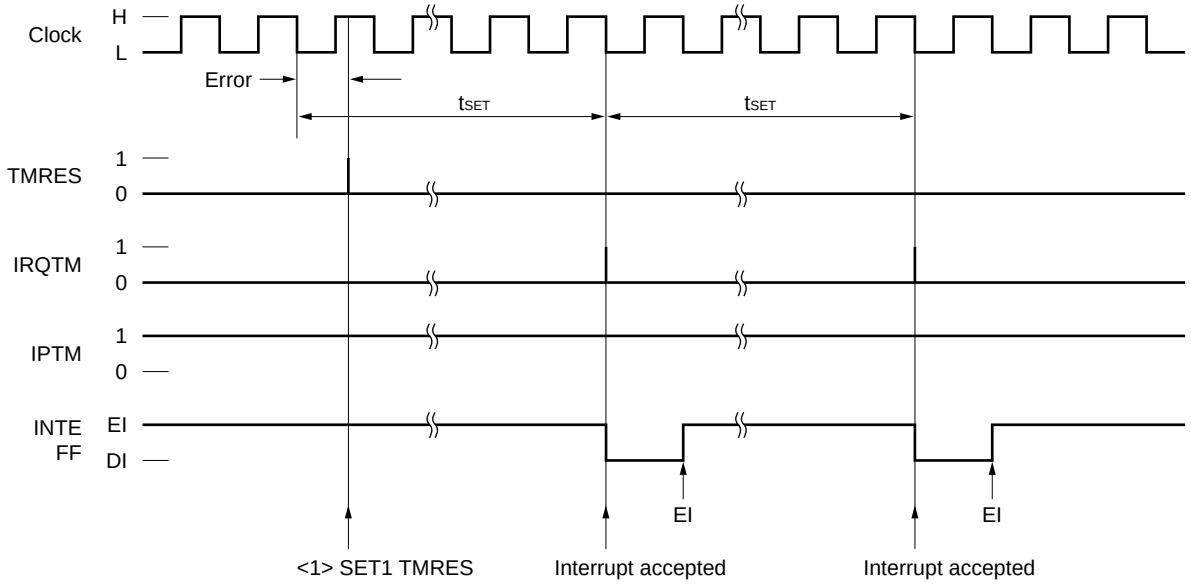
If the timer modulo register operates with the TMEN flag set in<1> above, the timer/counter is incremented at the falling edge of the clock, and an interrupt request is issued when the contents of the timer/counter coincide with those of the timer modulo register.

Depending on the timing at which the TMEN flag is set to 1, the error of the interrupt request issuance varies as follows:

$$0 \leq (\text{error}) < (\text{one cycle of count clock})$$

Figure 11-17. Error of 12-Bit Timer (2/2)

(2) When timer/counter is reset



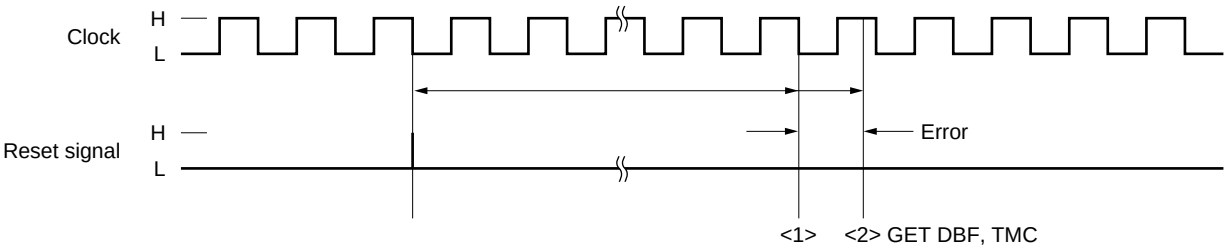
Remark t_{SET} = (Data set to timer modulo register) × (Count clock)

If the TMRES flag is set in <1> above, the contents of the timer/counter are reset, and the timer/counter is incremented at the falling edge of the next clock. When the contents of the timer/counter coincide with those of the timer modulo register, an interrupt request is issued.

Depending on the timing at which the TMRES flag is set to 1 at this time, the error of interrupt request issuance varies as follows:

$$0 \leq (\text{error}) < (\text{one cycle of count clock})$$

(3) When data of timer/counter is read during counting operation



When the data of the timer/counter is read in <1> and <2> above, the result is the same data in both the cases. Therefore, the error when the data of the timer/counter is read is as follows:

$$0 \leq (\text{error}) \leq (\text{one cycle of count clock})$$

11.5.8 Notes on using 12-bit timer

The interrupt by the 12-bit timer may occur at the same time as the CE reset or basic timer 1 interrupt.

Therefore, if time control, such as watch processing, is necessary at CE reset, do not use the 12-bit timer but use the basic timer 0 carry or basic timer 1 interrupt.

When using the 12-bit timer in combination with the basic timer 1 interrupt, pay attention to the priorities.

12. STANDBY

The standby function is used to reduce the current dissipation of the device during back up.

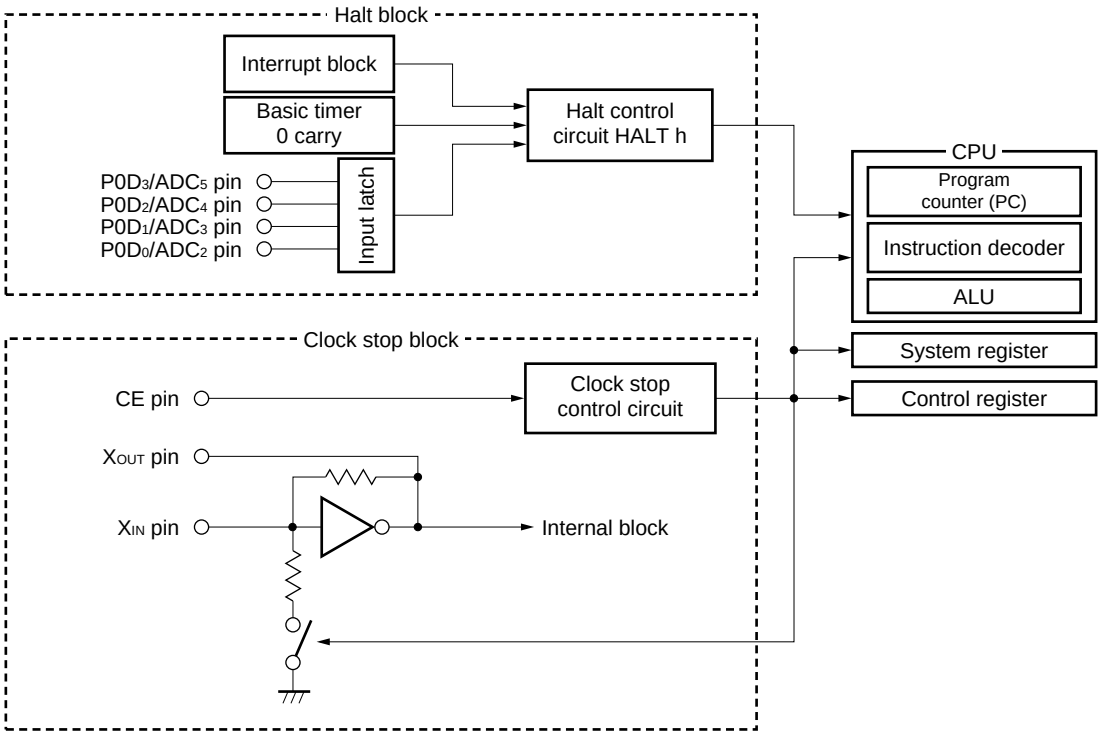
12.1 Configuration of Standby Block

Figure 12-1 shows the configuration of the standby block.

As shown in this figure, the standby block is divided into two subblocks: halt control block and clock stop control block.

The halt control block consists of a halt control circuit, interrupt control block, basic timer 0 carry, and the $P0D_0$/ADC_2 through $P0D_3$/ADC_5 key input pins, and controls the operation of the CPU (program counter, instruction decoder, and ALU block). The clock stop control block has a clock stop control circuit that controls the 4.5-MHz crystal oscillator circuit, CPU, system registers, and control registers.

Figure 12-1. Configuration of Standby Block



12.2 Standby Function

The standby function reduces the current dissipation of the device by stopping part of or entire device operation.

The standby function is divided into a halt function and clock stop function.

The halt function reduces the current dissipation of the device by stopping the operation of the CPU by using a dedicated instruction “HALT h”.

The clock stop function reduces the current dissipation of the device by stopping the 4.5-MHz crystal oscillation circuit by using a dedicated instruction “STOP s”. In addition to the halt and clock stop functions, the CE pin is also used to set an operation mode of the device.

The CE pin is used to control the operation of the PLL frequency synthesizer and to reset the device, and therefore, can be said to be one of the standby functions. **12.3** below describes how device operation modes are set by the CE pin.

12.4 and **12.5** respectively describe the halt function and clock stop function.

12.3 Device Operation Mode Set by CE Pin

The CE pin controls the following functions (1) through (3) depending on the input level of and the rising edge of an externally input signal.

(1) Operation control of PLL frequency synthesizer
(2) Validation control of clock stop instruction
(3) Device reset

The following **12.3.1** through **12.3.3** describe (1) through (3) above.

12.3.1 Operation control of PLL frequency synthesizer

The PLL frequency synthesizer can operate only while the CE pin is high.

While the CE pin is low, the frequency synthesizer is automatically disabled.

In the PLL disable status, the VCOH and VCOL pins are internally pulled down, and the EO_0 and EO_1 pins are floated.

The PLL frequency synthesizer can be disabled by program even when the CE pin is high.

12.3.2 Validation control of clock stop instruction

The clock stop instruction “STOP s” is valid only when the CE pin is low.

The “STOP s” instruction executed when the CE pin is high is excuted as a no-operation (NOP) instruction.

12.3.3 Device reset

By asserting the CE pin high, the device can be reset (CE reset).

In addition to CE reset, power-ON reset can be also performed on application of supply voltage V_{DD}.

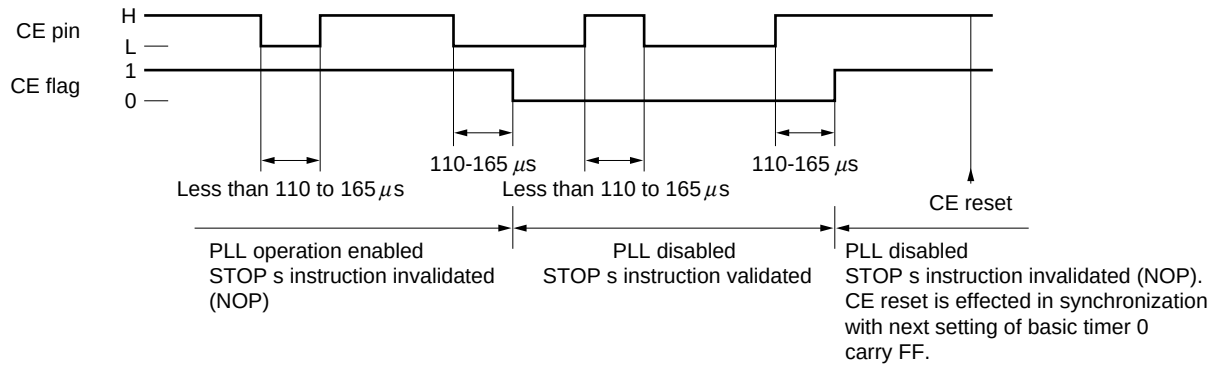
For details, refer to **13. RESET**.

12.3.4 Signal input to CE pin

The CE pin does not accept a low level or high level of less than 110 to 165 μs to prevent malfunctioning due to noise.

The level of the signal input to the CE pin can be detected by the CE flag of the CE pin level judge register (RF address 07H). Figure 12-2 shows the relation between the input signal and CE flag.

Figure 12-2. Relation between CE Pin Input Signal and CE Flag



12.3.5 Configuration and function of CE pin level judge register

The CE pin level judge register detects the input signal level of the CE pin.

The configuration and function of this register are shown below.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
CE pin level judge register (CEJDG)	0	0	0	CE	07H	R

CE	Detects level input to CE pin
0	Low level is input
1	High level is input
b_3, b_2, b_1	Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	–
	Clock stop	↓	↓	↓	–
	CE	↓	↓	↓	–

The CE flag is not affected by a low or high level of less than 110 to 165 μs.

12.4 Halt Function

The halt function stops the operation clock of the CPU by executing the “HALT h” instruction.

When this instruction is executed, the program is stopped, until the halt status is later released. The power dissipation of the device in the halt status is reduced by the operation current of the CPU. The halt status is released by key input, basic timer 0 carry, and interrupt.

The release condition of the key input, basic timer 0 carry, and interrupt is specified by the operand “h” of the “HALT h” instruction. The “HALT h” instruction is valid regardless of the input level of the CE pin.

The following **12.4.1** through **12.4.6** describe the halt status and halt release conditions.

12.4.1 Halt status

In the halt status, all the operations of the CPU are stopped.

In other words, program execution is stopped at the “HALT h” instruction.

However, the peripheral hardware continues the operation set before the “HALT h” instruction is executed.

For the operation of the peripheral hardware, refer to **12.6 Device Operation in Halt and Clock Stop Status**.

12.4.2 Halt releasing condition

Figure 12-3 shows the halt release conditions.

As shown in this figure, the halt release conditions are set by the 4-bit data specified by the operand "h" of the "HALT h" instruction.

The halt status is released when a condition specified by "1" in operand "h" is satisfied.

When the halt status is released, execution is started from the instruction next to the "HALT h" instruction.

If two or more release conditions are set, the halt status is released if one of the set conditions is satisfied.

When the device is reset (power-ON reset or CE reset), the halt status is released, and the reset operation is performed.

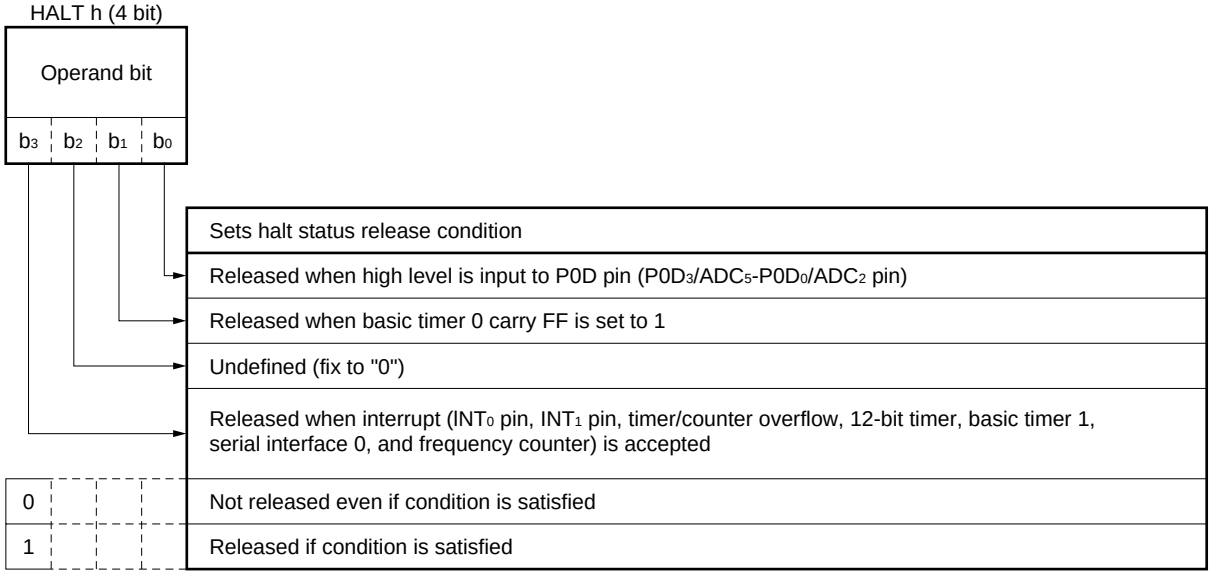
If 0000B is set as halt release condition "h", no release condition is set.

At this time, the halt status is released when the device is reset (power-ON reset or CE reset).

The following **12.4.3** through **12.4.5** describe the halt release conditions by key input, basic timer 0 carry, and interrupt, respectively.

12.4.6 shows an example where two or more release conditions are set.

Figure 12-3. Halt Release Condition



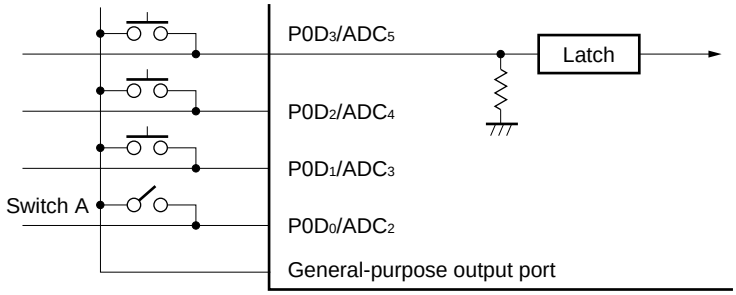
12.4.3 Releasing halt by key input

Releasing the halt status by key input is set by the “HALT 0001B” instruction.

When releasing the HALT condition by key input is set, the halt status is released if a high level is input to any one of the P0D0/ADC2 to P0D3/ADC5 pins.

The following paragraphs (1) through (4) describe the points to be noted when a general-purpose output port is used as a key source signal, when LCD segment signal output is multiplexed with key source signal output, and when the P0D0/ADC2 through P0D3/ADC5 pins are used as A/D converter pins.

(1) Notes on using general-purpose output port as key source signal



After the general-purpose output port for key source signal is asserted high level, the “HALT 0001B” instruction is executed.

If an alternate switch such as A in the above figure is used, a high level is always applied to the P0D0/ADC2 pin while switch A is closed.

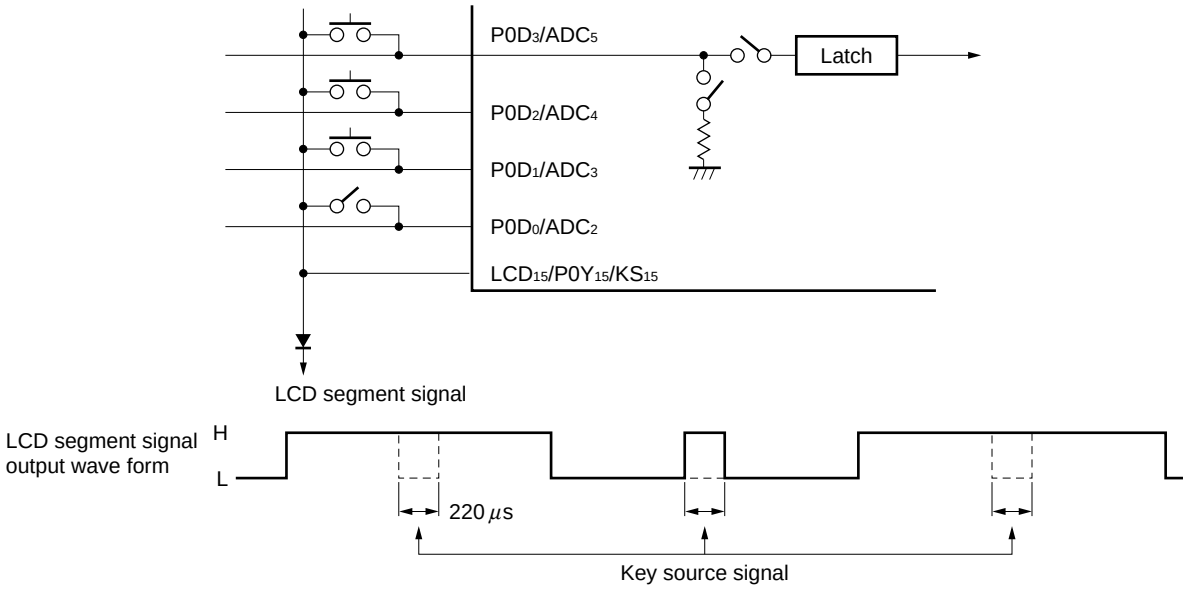
Consequently, the halt status is released immediately.

Therefore, exercise care when using an alternate switch.

To use a general-purpose output port as a key source signal, reset the KSEN flag of the LCD mode select register (LCDMODE: RF address 10H) to “0”.

At this time, the P0D0/ADC2 through P0D3/ADC5 pins are automatically pulled down internally.

(2) Notes on multiplexing LCD segment signal output with key source signal output



After setting the key source signal output data, execute the “HALT 0001B” instruction.

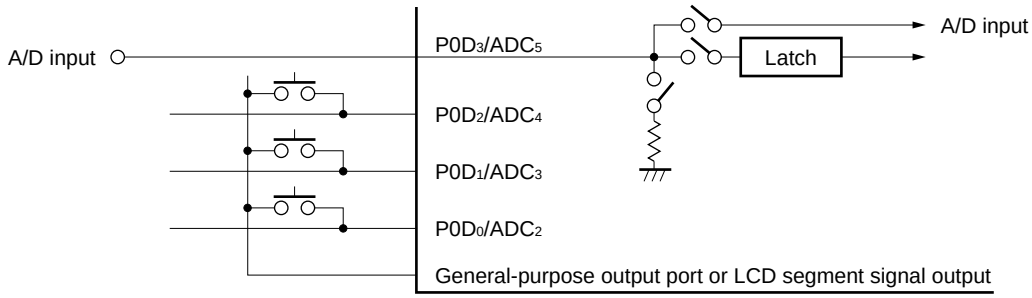
If the key source signal output data is “0” at this time, the halt status is not released even if the high level of an LCD segment signal is input to the pin.

To multiplex LCD segment signal output with key source signal output, set the KSEN flag of the LCD mode select register to 1.

The key source signal data (setting a pin that outputs the key source) is set by the key source data register (KSR: peripheral address 42H) via data buffer.

When the LCD segment signal output is multiplexed with key source signal output, the internal key latch circuit latches data only while the key source signal is output, and is disconnected from the external source while the LCD segment signal is output. The internal pull-down resistor is ON only while the key source signal is output.

(3) Notes on using P0D0/ADC2 through P0D3/ADC5 pins as A/D converter pins



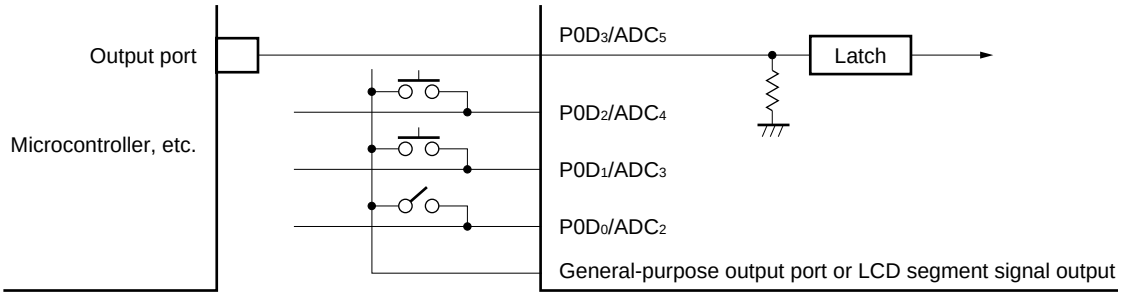
When any of the P0D0/ADC2 through P0D3/ADC5 pins is selected as the A/D converter pins, the selected pin (only one pin can be selected at a time) is disconnected from the input latch and connected to the internal A/D converter.

If a high level happens to be input to the pin when the pin is selected to the A/D converter, the latch circuit retains the high level

If the “HALT 0001B” instruction is executed in this status, the halt status is immediately released because the input latch is high.

To avoid this, set the input port mode before executing the “HALT 0001B” instruction, and inputs a low level to the A/D converter.

(4) Others



The P0D0/ADC2 through P0D3/ADC5 pins can be used as general-purpose input port pins with pull-down resistor. Therefore, the halt status can also be released by another microcontroller as shown above.

12.4.4 Releasing halt status by basic timer 0 carry

Releasing the halt status by using the basic timer 0 carry is set by the “HALT 0010B” instruction.

When releasing the halt status by using the basic timer 0 carry is set, the halt status is released as soon as the basic timer 0 carry FF has been set to 1.

The basic timer 0 carry FF corresponds to the BTM0CY flag of the basic timer 0 carry FF judge register on a one-to-one basis, as described in **11. TIMER FUNCTION**, and is set to 1 at fixed time intervals (1, 5, 100, or 250 ms). Therefore, the halt status can be released at fixed time intervals.

Here is an example:

Example

```
        M1          MEM  0.10H          ; 1-second counter
        HLTBTMR DAT 0010B               ; Symbol definition
        INITFLG NOT BTM0CK1, BTM0CK0
                                        ; Embedded macro
                                        ; Sets basic timer 0 carry FF setting time to 250 ms
LOOP:
        HALT        HLTBTME             ; Setting basic timer 0 carry FF as halt release condition
        SKT1        BTM0CY              ; Embedded macro
        BR          LOOP                ; Branches to LOOP if BTM0CY flag is not set
        ADD         M1, #0100B          ; Adds 0100B to contents of M1
        SKT1        CY                  ; Embedded macro
        BR          LOOP                ; Executes processing A if carry occurs
        ┌──────────────┐
        │ Processing A │
        └──────────────┘
        BR          LOOP
```

In this example, the halt status is released every 250 ms, and processing A is executed every 1 second.

12.4.5 Releasing halt status by interrupt

Releasing the halt status by an interrupt is set by the “HALT 1000B” instruction.

When releasing the halt status by an interrupt is set, the halt status is released as soon as the interrupt is accepted.

There are the following six interrupt sources available (refer to **10. INTERRUPT**):

- INT_0 pin
- INT_1 pin or timer/counter overflow
- 12-bit timer
- Basic timer 1
- Serial interface 0
- Frequency counter

Therefore, which interrupt source is used to release the halt status must be specified in advance by program.

To accept an interrupt, the interrupt request must be issued from each interrupt source, all the interrupts must be enabled (by the EI instruction), and each interrupt must be enabled (the corresponding interrupt permission flag must be set).

Even if an interrupt request is issued, therefore, if the interrupt is not enabled, the interrupt is not accepted, and the halt status is not released.

When the halt status is released by accepting an interrupt, the program flow is transferred to the vector address of the interrupt.

When the “RETI” instruction is executed after the interrupt processing, the program flow is restored to the instruction next to the “HALT” instruction.

An example is given below.

Example

```
HLTINT   DAT     1000B      ; Symbol definition of halt condition
INTBTM1  DAT     0003H      ; Interrupt vector address symbol definition
INT0PIN  DAT     0006H      ; Interrupt vector address symbol definition

START:                      ; Program address 0000H
         BR      MAIN
ORG      INTBTM1            ; Timer interrupt vector address (0003H)
         BR      INTBTIMER1
ORG      INT0PIN            ; INT0 pin interrupt vector address (0006H)
         Processing A       ; Interrupt processing by INT0 pin
         BR      EI_RETI
INTBTIMER1:
         Processing B       ; Interrupt processing by timer
EI_RETI:
         EI
         RETI
MAIN:
         SET2    IPBTM1, IP0  ; Embedded macro
         SET2    BTMCK3, BTMCK2
                              ; Embedded macro
                              ; Sets time interval of timer interrupt to 1 ms
LOOP:
         Processing C       ; Main routine processing
         EI                 ; Enables all interrupts
         HALT    HLTINT     ; Sets releasing halt status by interrupt
       ; <1>
         BR      LOOP
```

In the above example, the halt status is released when the interrupt by the basic timer 1 is accepted, and processing B is executed. When the interrupt by the INT_0 pin is accepted, processing A is executed. Each time the halt status is released, processing C is executed.

If the interrupt request by the INT_0 pin and the interrupt request by the basic timer 1 are issued exactly at the same time in the halt status, the processing A of the INT_0 pin which has the higher hardware priority is executed.

If the “RETI” instruction is executed after processing A has been executed, execution is restored to the “BR LOOP” instruction in <1>, but the “BR LOOP” instruction is not executed, and the basic timer 1 interrupt is immediately accepted.

If the “RETI” instruction is executed after the processing B of the basic timer 1 interrupt processing has been executed, the “BR LOOP” instruction is executed.

Caution To execute the HALT instruction whose release condition is setting the interrupt request flag (IRQxxx) with the corresponding interrupt permission flag (IPxxx) set, describe a NOP instruction immediately before the HALT instruction.

If the NOP instruction is described immediately before the HALT instruction, time of one instruction is generated between the IRQxxx manipulation instruction and HALT instruction. When the CLR1 IRQxxx instruction is executed, for example, clearing IRQxxx is correctly reflected on the HALT instruction (Example 1). If the NOP instruction is not described immediately before the HALT instruction, the CLR1 IRQxxx instruction is not reflected on the HALT instruction, and the HALT mode is not set (Example 2).

Example 1. Program that correctly executes HALT instruction

```
        ⋮
        ⋮              ; Setting of IRQ×××
        ⋮
CLR1    IRQ×××
NOP                    ; Describe NOP instruction immediately before HALT instruction
                       ; (clearing IRQxxx is correctly reflected on HALT instruction
HALT    1000B          ; Correctly executes HALT instruction (HALT mode is set)
        ⋮
        ⋮
```

2. Program that does not set HALT mode

```
        ⋮
        ⋮              ; Setting of IRQ×××
        ⋮
CLR1    IRQ×××         ; Clearing IRQxxx is not reflected on HALT instruction
                       ; (reflected on instruction next to HALT instruction)
HALT    1000B          ; HALT instruction is ignored (HALT mode is not set)
        ⋮
        ⋮
```

12.4.6 If two or more release conditions are set simultaneously

If two or more halt release conditions are set at the same time, the halt status is released if even one of the set conditions is satisfied. The following examples indicate how release conditions is identified when two or more release conditions are satisfied at once.

Example 1.

```
            HLTINT      DAT 1000B
            HLTBTMR     DAT 0010B
            HLTKEY      DAT 0001B
            INT0PIN     DAT 0006H       ; INT0 pin interrupt vector address symbol definition
START:
            BR          MAIN
ORG:        INT0PIN
            Processing A                ; INT0 pin interrupt processing
            EI
            RETI
BTMRUP:                                 ; Basic timer 0 carry processing
            Processing B
            RET
KEYDEC:                                 ; Key input processing
            Processing C
            RET
MAIN
            MOVT        DBF, @AR        ; Sets key source output data (table reference) to
                                        ; key source data register (KSR)
            PUT         KSR, DBF
            SET2        KSEN, LCDEN     ; Embedded macro
                                        ; Multiplexes LCD segment signal output with key source
                                        ; signal output
            SET2        BTM0CK1, BTM0CK0 ; Embedded macro
                                        ; Sets basic timer 0 carry FF setting time to 1 ms
            SET1        IP0             ; Embedded macro
                                        ; Enables INT0 pin interrupt
            EI
LOOP:
            HALT        HLTINT OR HLTBTMR OR HLTKEY
                                        ; Sets interrupt, basic timer 0 carry, and key in
                                        ; put as halt releasing conditions
            SKF1        BTM0CY          ; Embedded macro
                                        ; Detects BTM0CY flag
            CALL        BTMRUP          ; Basic timer 0 carry processing if BTM0CY flag is set
            SKF1        KEYJ            ; Embedded macro
                                        ; Detects key input latch
            CALL        KEYDEC          ; Key input processing if latched
            BR          LOOP
```

In example 1 above, the INT_0 pin interrupt, 1-ms basic timer 0 carry, and key input are set as the halt status releasing conditions.

To identify the condition responsible for releasing the halt status, a vector address is detected if the halt status is released by an interrupt, the BTM0CY flag is detected if the halt status is released by the basic timer 0 carry, and the KEYJ flag is detected if the halt status is released by key input.

When using two or more releasing conditions, the following two points must be noted:

(1) All the set release conditions must be detected when the halt status is released.

(2) The release conditions are detected according to their priorities.

Care must be exercised if the program after "MAIN" in Example 1 above is as shown in Example 2 below. Do not create the following program if the priority of the timer by timer carry is high.

Example 2.

```
MAIN:
        SET4    P1C3, P1C2, P1C1, P1C0    ; Uses general-purpose output port as key source signal
        SET2    BTM0CK1, BTM0CK0
        SET1    IP0
        EI
LOOP:
        HALT    HLTINT OR HLTBTMR OR HLTKEY
        SKF4    P0D3, P0D2, P0D1, P0D0    ; Detects key input
        BR      KEYDEC
        SKF1    BTM0CY
        CALL    BTMRUP
        BR      LOOP
KEYDEC:                                   ; Key input processing
        Processing C
        BR      LOOP
```

Suppose that the halt status is released by key input in Example 2 above, and that the basic timer 0 carry FF is set to 1 immediately after that.

Then the program executes the "HALT" instruction again after executing the key input processing.

The halt status is immediately released because the basic timer 0 carry FF is set.

However, because a high level of about 100 ms is usually input as key input, execution branches to the key input processing.

Consequently, the basic timer 0 carry FF is not correctly detected.

12.5 Clock Stop Function

The clock stop function stops the 4.5-MHz crystal oscillation circuit by executing the “STOP s” instruction (clock stop status).

Therefore, the current dissipation of the device is reduced down to 5 μA MAX.

For the details of the current dissipation, refer to **12.7 Current Dissipation in Halt and Clock Stop Status**.

Specify “000B” as the operand “s” of the “STOP s” instruction.

The “STOP s” instruction is valid only when the CE pin is low, and is executed as a no-operation (NOP) instruction when the CE pin is high.

Therefore, the “STOP s” instruction must be executed when the CE pin is low.

The clock stop status is released by asserting the CE pin high (CE reset).

The following **12.5.1** through **12.5.3** describe the clock stop status, how to release the clock stop status, and notes on using the clock stop instruction.

12.5.1 Clock stop status

Because the crystal oscillation circuit is stopped in the clock stop status, all the device operations, such as the CPU and peripheral hardware, are stopped.

For the operations of the CPU and peripheral hardware, refer to **12.6 Device Operation in Halt and Clock Stop Status**.

In the clock stop status, the power failure detection circuit does not operate even if the supply voltage V_{DD} of the device is lowered to 2.2 V. Therefore, the data memory can be backed up at a low voltage. For the details of the power failure detection circuit, refer to **13. RESET**.

12.5.2 Releasing clock stop status

The clock stop status is released by asserting the CE pin high (CE reset) or lowering the device supply voltage V_{DD} to 2.2 V once and then increasing it to 4.5 V (power-ON reset).

Figures 12-4 and 12-5 show the releasing operation at CE reset and power-ON reset, respectively.

When the clock stop status is released by power-ON reset, the power failure detection circuit operates.

For the details of the power-ON reset, refer to **13.4 Power-ON Reset**.

Figure 12-4. Releasing Clock Stop Status by CE Reset

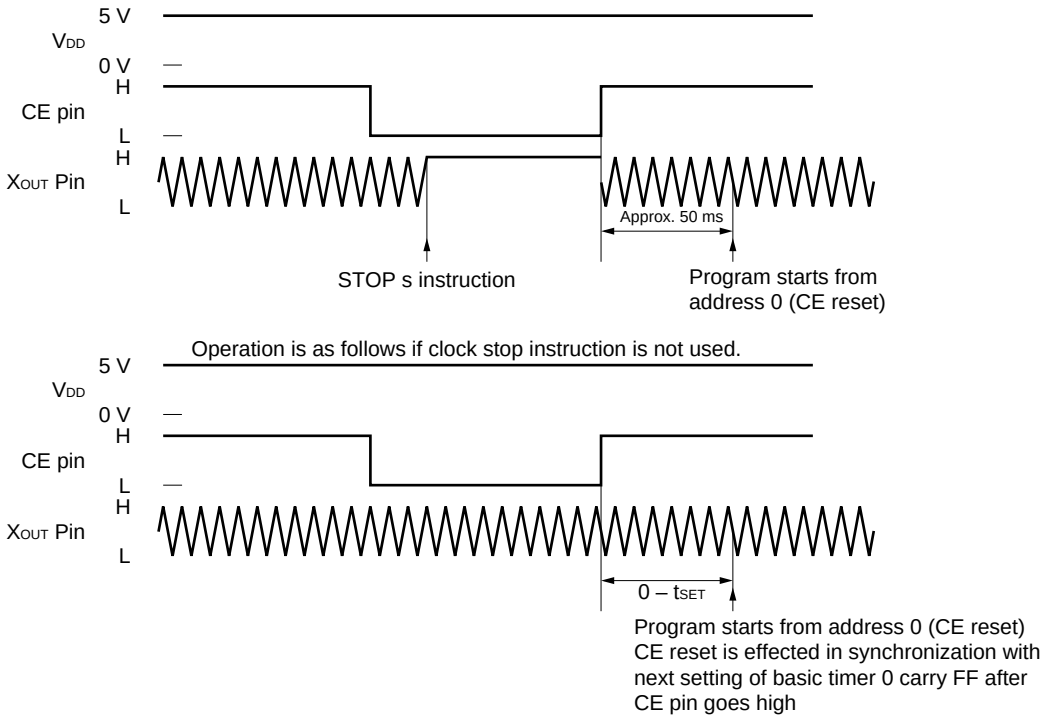
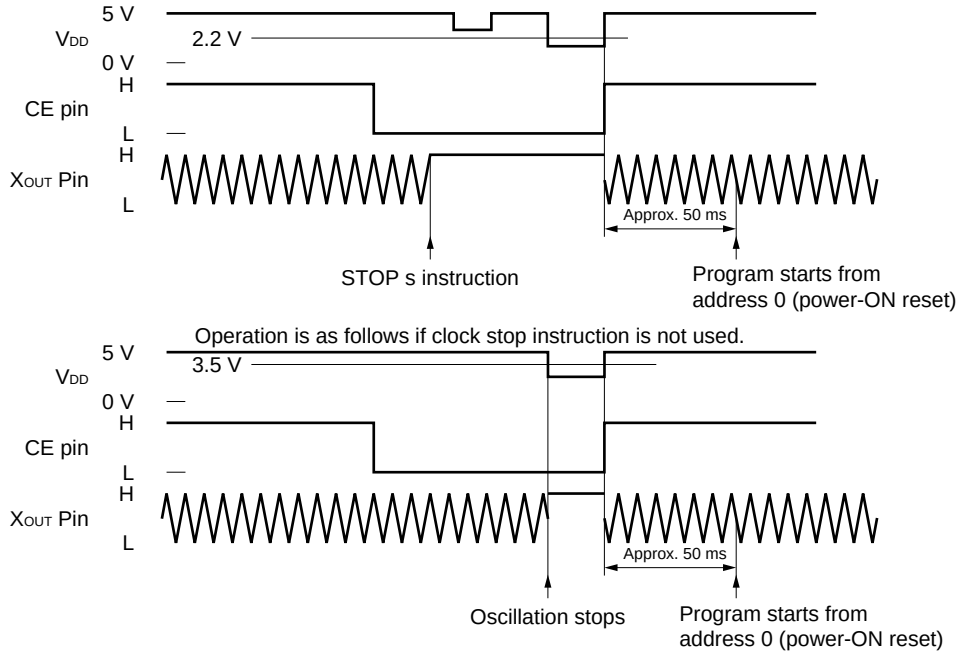


Figure 12-5. Releasing Clock Stop Status by Power-ON Reset



12.5.3 Notes on using clock stop instruction

The clock stop instruction (STOP s) is valid only when the CE pin is low.

Therefore, processing to be performed when the CE pin happens to be high must be considered in the program.

Here is an example:

Example

```
            XTAL    DAT     0000B       ; Symbol definition of clock stop condition
CEJDG:
  ; <1>
            SKF1    CE                  ; Embedded macro
                                        ; Detects input level of CE pin
            BR      MAIN                ; If CE = high, branches to main processing
            Processing A                ; Processing when CE = low
  ; <2>
            STOP    XTAL                ; Clock stop
  ; <3>
            BR      $–1
MAIN:
            Main processing
            BR      CEJDG
```

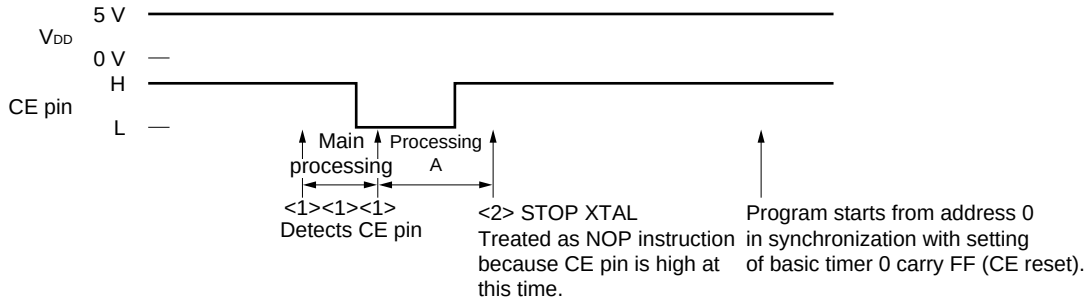
In the above example, the status of the CE pin is detected in <1>. If the CE pin is low, processing A is executed, and then the clock stop instruction in <2>, “STOP XTAL”, is executed.

If the CE pin goes high while the “STOP XTAL” instruction in <2> is executed as shown in the figure below, the “STOP XTAL” instruction operates as a no-operation (NOP) instruction.

At this time, without a branch instruction of <3> “BR $–1”, the program might malfunction by branching to the main processing.

It is therefore necessary to insert a branch instruction as in <3> or to create a program that does not malfunction even if execution branches to the main processing.

If a branch instruction is used as in <3>, CE reset is effected in synchronization with the next setting of the basic timer 0 carry FF even if the CE pin remains high.



12.6 Device Operation in Halt and Clock Stop Status

Table 12-1 shows the operations of the CPU and peripheral hardware in the halt and clock stop statuses.

As shown in this table, all the peripheral hardware continues the normal operation in the halt status, but instruction execution is stopped.

In the clock stop status, all the peripheral hardware stops operation.

The control registers that control the operations of the peripheral hardware operate normally in the halt status (not initialized) but are initialized to predetermined values in the clock stop status (when the STOP s instruction is executed).

In other words, the peripheral hardware continues the operation set by the control registers in the halt status. In the clock stop status, the operations of the peripheral hardware are determined by the control registers that have been initialized to predetermined values.

For the values to which the control registers are to be initialized, refer to **8. REGISTER FILE (RF)**.

Here is an example:

Example To set $P0A_3$/SDA and $P0A_2$/SCL pins of port 0A in output port mode and use $P0A_1/\overline{SCK_0}$ and $P0A_0/SO_0$ pins as serial interface lines

```
  HLTINT    DAT       1000B       ; Symbol definition
  XTAL      DAT       0000B       ;
  INITFLG P0ABIO3, P0ABIO2, P0ABIO1, P0ABIO0
                                  ; Embedded macro
; <1>
  SET2      P0A3, P0A2            ;
  INITFLG SIO0CH, NOT SB, SIO0MS, SIO0TX
                                  ;
  SET2      SIO0CK1, SIO0CK0
; <2>
  INITFLG NOT SIO0IMD1, SIO0IMD0
  CLR1      IRQSIO0
  SET1      IPSIO0
  EI
; <3>
  SET1      SIO0NWT
; <4>
  HALT      HLTINT
; <5>
  STOP      XTAL
```

In the above example, the $P0A_3$ and $P0A_2$ pins output a high level in <1>, the conditions of the serial interface 0 are set in <2>, and serial communication is started in <3>.

If the "HALT" instruction is executed in <4> at this time, serial communication is continued. The halt status is released when the interrupt by the serial interface 0 is accepted.

If the "STOP" instruction is executed in <5> instead of the "HALT" instruction in <4>, all the flags of the control registers set in <1>, <2>, and <3> are initialized when the "STOP" instruction is executed. Consequently, serial communication is stopped, and all the pins of port 0A are set in the general-purpose input port mode.

Table 12-1. Device Operation in Halt and Clock Stop Status

Peripheral Hardware	Status			
	CE Pin = High Level		CE Pin = Low Level	
	Halt	Clock Stop	Halt	Clock Stop
Program counter	Stops at address of HALT instruction	STOP instruction is invalid (NOP)	Stops at address of HALT instruction	Initialized to 0000H and stops
System register	Retained		Retained	Initialized[Note]
Peripheral register	Retained		Retained	Retained
Control register	Retained		Retained	Initialized[Note]
12-bit timer	Normal operation		Normal operation	Operation stopped
Basic timer	Normal operation		Normal operation	Operation stopped
PLL frequency synthesizer	Normal operation		Disabled	Operation stopped
A/D converter	Normal operation		Normal operation	Operation stopped
D/A converter	Normal operation		Normal operation	Operation stopped
Clock generator port	Normal operation		Normal operation	Operation stopped
Serial interface	Normal operation		Normal operation	Operation stopped
Frequency counter	Normal operation		Normal operation	Operation stopped
LCD controller/driver	Normal operation		Normal operation	Operation stopped
Key source controller/decoder	Normal operation		Normal operation	Operation stopped
General-purpose I/O port	Normal operation		Normal operation	Input port
General-purpose input port	Normal operation		Normal operation	Input port
General-purpose output port	Normal operation		Normal operation	Retained

Note For the values to which the control registers are to be initialized, refer to **5. SYSTEM REGISTER (SYSREG)** and **8. REGISTER FILE (RF)**.

12.7 Current Dissipation in Halt and Clock Stop Status

12.7.1 Device current dissipation in halt status

Figure 12-6 shows the device current dissipation I_{DD} in the halt status.

The numbers (1) through (4) in this figure indicate current dissipation when each of the four types of programs below is executed.

As shown in Figure 12-6, the less often the halt status is released, the lower the current dissipation is.

(1) Program 1

The HALT instruction is not used.

Example

```
NOP
BR    $–1
```

(2) Program 2

The 5-ms basic timer 1 interrupt is set as the halt release condition, and 20 instructions (about 90 μs) are executed each time the halt status is released.

Example

```
HLTINT   DAT      1000B
INTBTM1  DAT      0003H

         BR       LOOP
ORG      BTM1INT
REPT     17
         NOP
ENDR
         EI
         RETI
LOOP:
         INITFLG  BTM1CK1, NOT BTM1CK0
         SET1     IPBTM1
         EI
         HALT     HLTINT
         BR       $–1
```

(3) Program 3

The 100-ms basic timer 1 interrupt is set as the halt rlease condition, and 20 instructions are executed each time the halt status is released.

Example

```
HLTINT   DAT      1000B
INTBTM1  DAT      0003H

         BR       LOOP
ORG      BTM1INT
REPT     17
         NOP
ENDR
         EI
         RETI
LOOP:
         CLR2     BTM1CK1, BTM1CK0
         SET1     IPBTM1
         EI
         HALT     HLTINT
         BR       $–1
```

(4) Program 4

Nothing is set as the halt release condition.

Example

```
HLTNORLS DAT 0000B
         HALT HLTNORLS
```

The device current dissipation I_{DD} shown in Figure 12-6 is measured under the following conditions:

- PLL is disabled.
- Frequency counter is disabled.
- Sine wave with a frequency f_{IN} = 4.5 MHz and input amplitude $V_{IN} = V_{DD}$ to the X_{IN} pin from a standard signal generator.
- All the pins set in the output mode are open.
- All the pins set in the input port (except the X_{IN} pin) are pulled down with a 47-KΩ resistor.

12.7.2 Device current dissipation in clock stop status

Figure 12-7 shows the device current dissipation IDD in the clock stop status.

The current dissipation shown in Figure 12-7 is measured under the following conditions:

- All the pins set in the output mode are open.
- All the pins set in the input mode (except the X_{IN} pin) are pulled down with a 47-KΩ resistor.
- A crystal resonator is connected (oscillation is stopped, however).

Figure 12-6. Device Current Dissipation in Halt Status (reference)

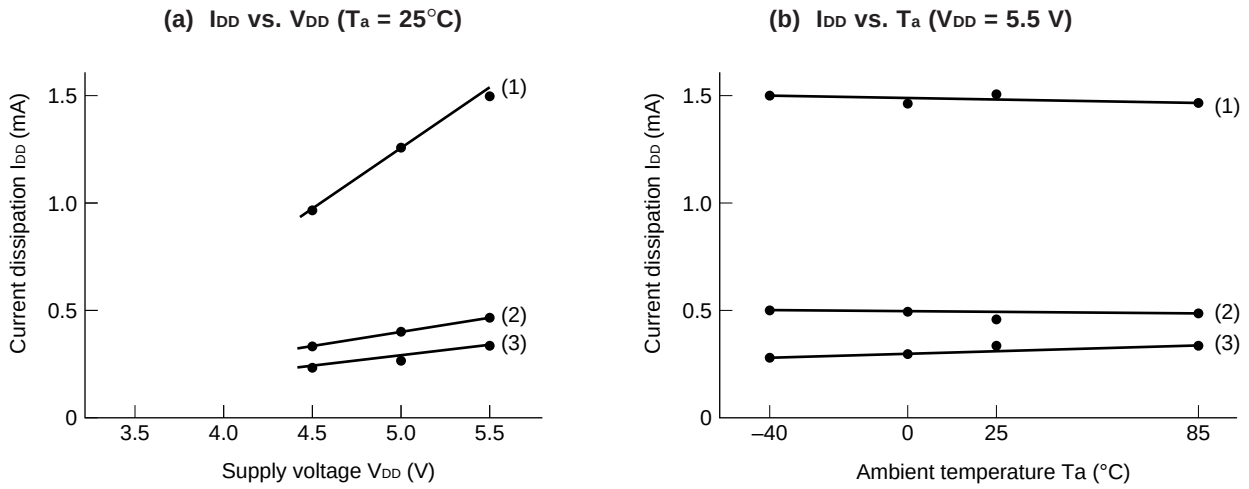
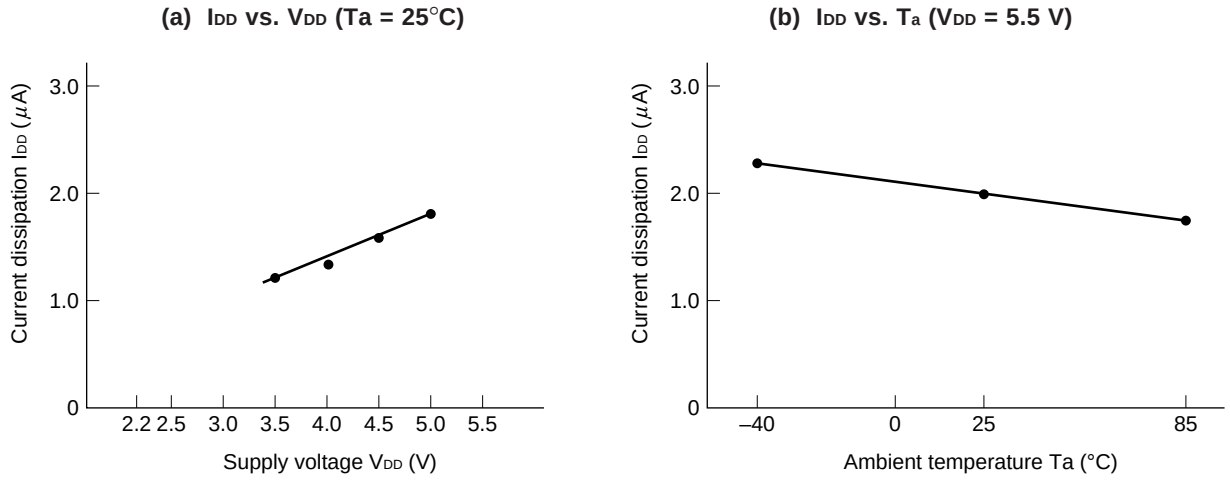


Figure 12-7. Device Current Dissipation in Clock Stop Status (reference)



12.7.3 Notes on processing of each pin in halt and clock stop statuses

The halt status is used to reduce the current dissipation when, for example, only the watch operates.

The clock stop function is used to reduce the current dissipation when only the contents of the data memory are to be retained.

Therefore, the current dissipation must be minimized in the halt and clock stop statuses.

The current dissipation substantially changes depending on the status of each pin.

Therefore, remember the points indicated in Table 12-2.

Table 12-2. Notes on Status of Each Pin in Halt and Clock Stop Statuses (1/2)

Pin Function		Pin Symbol	Pin Status and Notes on Processing	
			Halt Status	Clock Stop Status
General-purpose I/O port	Port 0A	$P0A_3$/SDA $P0A_2$/SCL $P0A_1/\overline{SCK_0}$ $P0A_0/SO_0$	Status before halt status is set is retained. **(1) When specified as output pin** Current dissipation increases if these pins are externally pulled down while they output high level, or externally pulled up while they output low level. Exercise care in using N-ch open-drain output pins ($P0A_3$, $P0A_2$, $P1B_3$-$P1B_0$). **(2) When specified as input pin** (except ports 1A and 1D) Current dissipation increases due to noise if these pins are floated. **(3) Port 0D ($P0D_3/ADC_5$-$P0D_0/ADC_2$)** Current dissipation increases if these pins are externally pulled up because they are connected to internal pull-down resistor. However, pull-down resistor is disconnected from pins selected as A/D converter pins. **(4) Ports 1D ($P1D_3$/FMIFC-$P1D_0$/ADC_0) and 1A ($P1A_3$-$P1A_0$/FCG)** When $P1D_3$/FMIFC and $P1D_2$/AMIFC pins are used as IF counter pins, internal amplifier operates and current dissipation increases. Because IF counter is not automatically disabled even if CE pin goes low, initialize it by program as necessary. Power dissipation of ports 1D and 1A does not increase due to noise even if they are floated as general-purpose input port.	All pins are specified as general-purpose input port. Current dissipation of all input port pins, except port 0C ($P0C_3$-$P0C_0$), does not increase due to noise even if they are floated externally. Port 0C ($P0C_3$-$P0C_0$) must be externally pulled down or up so that current dissipation does not increase due to noise. Port 0D ($P0D_3/ADC_5$-$P0D_0/ADC_2$) is internally pulled down.
	Port 0B	$P0B_3/SI_0$ $P0B_2/\overline{SCK_1}$ $P0B_1/SO_1$ $P0B_0/SI_1$		
	Port 0C	$P0C_3$ $P0C_2$ $P0C_1$ $P0C_0$		
	Port 1A	$P1A_3$ $P1A_2$ $P1A_1$ $P1A_0$/FCG		
General-purpose input port	Port 0D	$P0D_3/ACD_5$ $P0D_2/ASC_4$ $P0D_1/ADC_3$ $P0D_0/ACD_2$		
	Port 1D	$P1D_3$/FMIFC $P1D_2$/AMIFC $P1D_1/ADC_1$ $P1D_0/ADC_0$		
General-purpose output port	Port 1B	$P1B_3/PWM_2$ $P1B_2/PWM_1$ $P1B_1/PWM_0$ $P1B_0$/CGP		Specified as general-purpose output port. Output contents are retained as is. Therefore, current dissipation increases if these pins are externally pulled down while high level is output or pulled up while low level is output.
	Port 1C	$P1C_3$ $P1C_2$ $P1C_1$ $P1C_0$		
	Port 2A	$P2A_0$		
Interrupt		INT_1 INT_0	Current dissipation increases due to external noise when these pins are floated.	

Table 12-2. Notes on Status of Each Pin in Halt and Clock Stop Statuses (2/2)

Pin Function	Pin Symbol	Pin Status and Notes on Processing	
		Halt Status	Clock Stop Status
LCD segment	$LCD_{29}/P0F_3$ | $LCD_{26}/P0F_0$ $LCD_{25}/P0E_3$ | $LCD_{22}/P0E_0$ $LCD_{21}/P0X_5$ | $LCD_{16}/P0X_0$ $LCD_{15}/P0Y_{15}/KS_{15}$ | $LCD_0/POY_0/KS_0$	When these pins are used as general-purpose output port pins, bear in mind same points as general-purpose port pins described earlier. Current dissipation increases via port 0D (connected with pull-down resistor) when key source signals are output and if there is switch that is always ON, such as transistor switch and if "1" is output as key source data.	All pins are specified as LCD segment signal output pins and output low level (display off).
PLL frequency synthesizer	VCOL VCOH EO_0 EO_1	Current dissipation increases while PLL operates. When PLL is disabled, status of each pin is as follows: VCOL, VCOH: Internally pulled down EO_0, EO_1: Floated When CE pin goes low, PLL is automatically disabled.	PLL is disabled. Status of each pin is as follows: VCOL, VCOH: Internally pulled down EO_0, EO_1: Floated
Crystal oscillation circuit	X_{IN} X_{OUT}	Current dissipation changes depending on oscillation waveform of crystal oscillation circuit. The greater the oscillation amplitude, the lower the current dissipation. Oscillation amplitude is affected by crystal oscillator to be used or load capacitor, and must be evaluated.	X_{IN} pin is internally pulled down, and X_{OUT} pin outputs high level.

13. RESET

The reset function is used to initialize the device operation.

13.1 Configuration of Reset Block

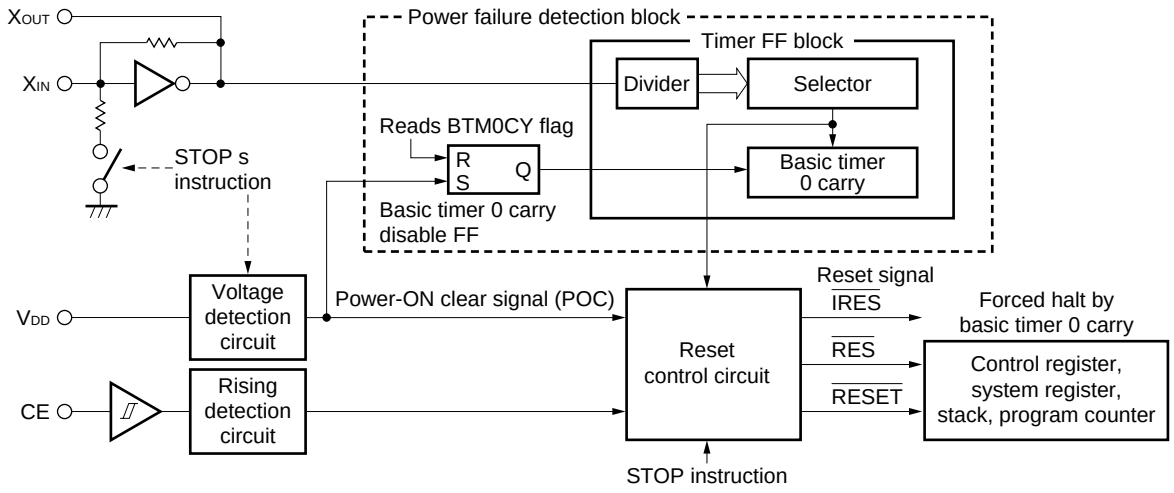
Figure 13-1 shows the configuration of the reset block.

The device is reset in two ways: power-ON reset or V_{DD} reset that is executed by applying supply voltage V_{DD}, and CE reset that is executed by using the CE pin.

The power-ON reset block consists of a voltage detection circuit that detects the voltage input to the V_{DD} pin, a power failure detection circuit, and a reset control circuit.

The CE reset block consists of a circuit that detects the rising of the signal input to the CE pin and a reset control circuit.

Figure 13-1. Configuration of Reset Block



13.2 Reset Function

Power-ON reset is executed when the supply voltage V_{DD} rises from a specific level, and CE reset is executed when the CE pin goes high.

Power-ON reset initializes the program counter, stack, system registers, and control registers, and executes the program from address 0000H.

CE reset initializes the program counter, stack, system registers, and some control registers, and executes the program from address 0000H.

The difference between power-ON reset and CE reset lies in the control registers that are initialized, and the operation of the power failure detection circuit described in **13.6**.

Power-ON reset and CE reset are controlled by the reset signals $\overline{IRES}$, $\overline{RES}$, and $\overline{RESET}$ that are output by the reset control circuit shown in Figure 13-1.

Table 13-1 shows the relations among the $\overline{IRES}$, $\overline{RES}$, and $\overline{RESET}$ signals, and power-ON reset and CE reset.

The reset control circuit also operates when the clock stop instruction (STOP s) described in **12. STANDBY** is executed.

The following **13.3** and **13.4** respectively describe CE reset and power-ON reset.

13.5 describes the relation between CE reset and power-ON reset.

Table 13-1. Relations among Internal Reset Signals and Reset Operations

Internal Reset Signal	Output Signal			Control Operation by Each Reset Signal
	On CE Reset	On Power-ON Reset	On Clock Stop	
$\overline{IRES}$	×	○	○	Forcibly sets device in halt status. Halt status is released by setting basic timer 0 carry FF
$\overline{RES}$	×	○	○	Initializes some control registers
$\overline{RESET}$	○	○	○	Initializes program counter, stack, system registers, and some control registers.

13.3 CE Reset

CE reset is executed when the CE pin goes high.

When the CE pin goes high, the $\overline{\text{RESET}}$ signal is output in synchronization with the rising edge of the next basic timer 0 carry FF setting pulse, and the device is reset.

When CE reset is executed, the program counter, stack, system registers, and some control registers are initialized by the $\overline{\text{RESET}}$ signal, and the program is started from address 0000H.

For the values to which the program counter, stack, system registers, and control registers are to be initialized, refer to the description of each register.

The operation of CE reset differs depending on whether the clock stop instruction is used or not.

This is described in details in **13.3.1** and **13.3.2** below.

13.3.3 describes points to be noted in executing CE reset.

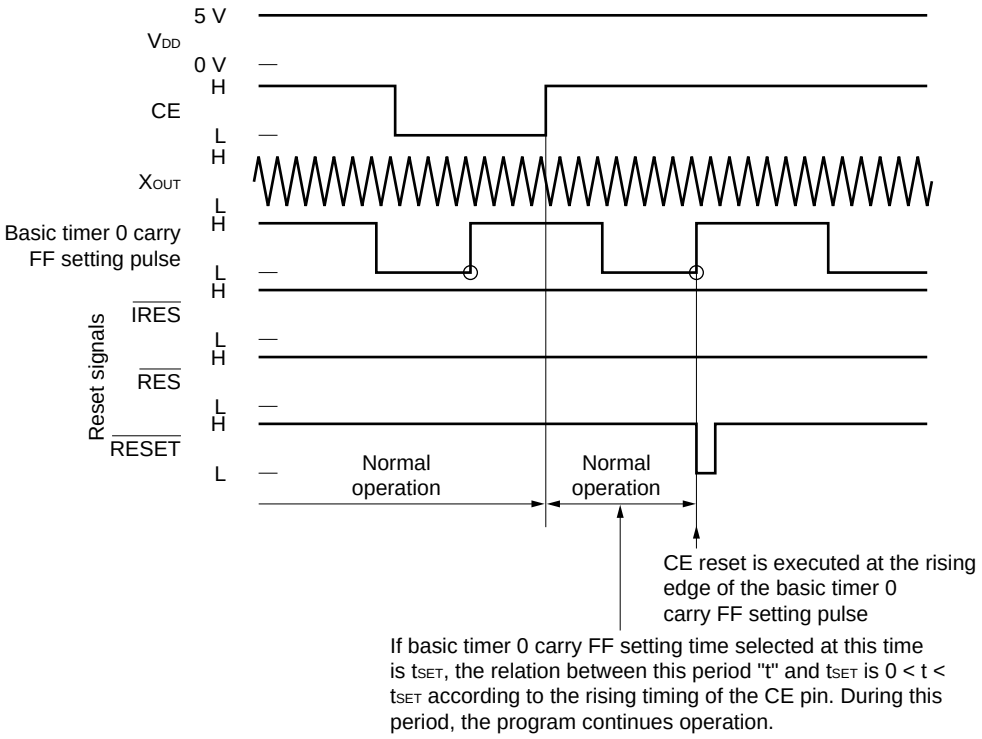
13.3.1 CE reset when clock stop instruction (STOP s) is not used

Figure 13-2 shows the operation.

When the clock stop instruction (STOP s) is not used, the basic timer clock select register of the control registers is not initialized.

Therefore, after the CE pin has gone high, the $\overline{\text{RESET}}$ signal is output at the rising edge of the basic timer 0 carry FF setting pulse selected at that time (1 ms, 5 ms, 100 ms, or 250 ms), and reset is effected.

Figure 13-2. CE Reset Operation When Clock Stop Instruction Is Not Used



13.3.2 CE reset when clock stop instruction (STOP s) is used

Figure 13-3 shows the operation.

When the clock stop instruction is used, the $\overline{\text{IRES}}$, $\overline{\text{RES}}$, and $\overline{\text{RESET}}$ signals are output as soon as the “STOP s” instruction has been executed.

At this time, the basic timer 0 carry FF setting signal is specified to 100 ms because the basic timer clock select register of the control registers is initialized to 0000B by the $\overline{\text{RES}}$ signal.

While the CE pin is low, output of the $\overline{\text{IRES}}$ signal continues, and the device is set in the forced halt status that is released by the basic timer 0 carry.

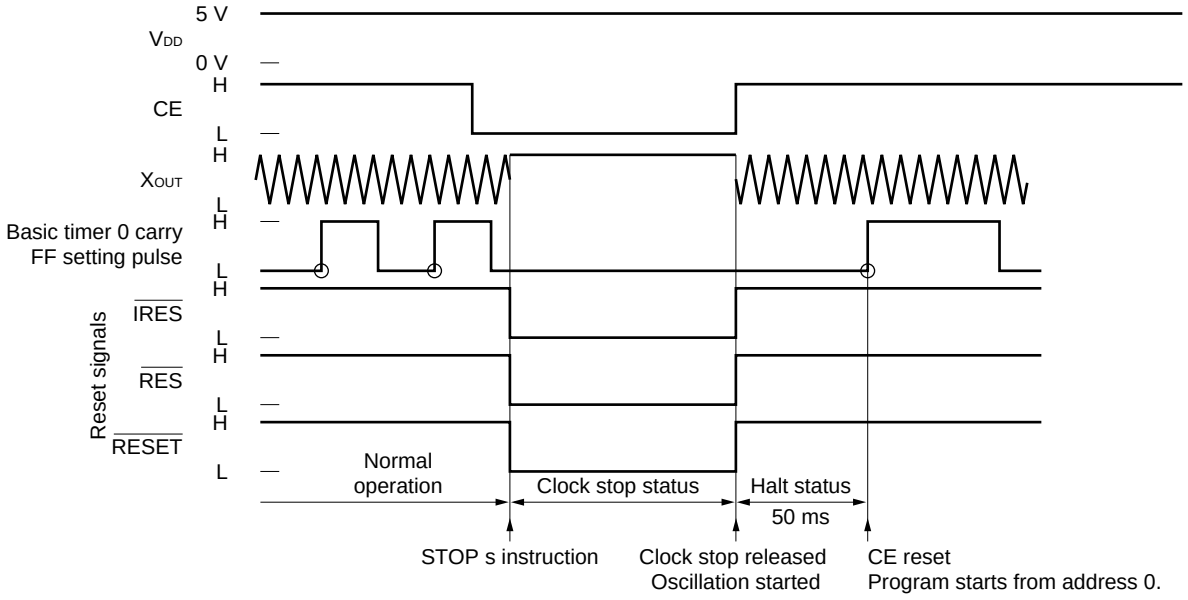
However, the device stops its operation because the clock is stopped.

When the CE pin goes high, the clock stop status is released, and oscillation starts.

Because the halt status is released by the basic timer 0 carry with the $\overline{\text{IRES}}$ signal at this time, the halt status is released and the program is started from address 0 when the basic timer 0 carry FF setting pulse rises after the CE pin has gone high.

Because the basic timer 0 carry FF setting pulse has been initialized to 100 ms, CE reset is executed 50 ms after the CE pin has gone high.

Figure 13-3. CE Reset Operation When Clock Stop Instruction Is Used



13.3.3 Notes on executing CE reset

Because CE reset is executed independently of the instruction under execution, remember the following two points (1) and (2):

(1) Time required to execute timer processing such as watch

When creating a watch program by using the basic timer 0 carry and basic timer 1 interrupt, the processing of the program must be completed within specific time.

For details, refer to **11.3.7 Notes on using basic timer 0 carry** and **11.4.5 Notes on using basic timer 1 interrupt**.

(2) Processing of data and flags used for program

Care must be exercised in rewriting the contents of data and flags that cannot be processed with one instruction and that must not change in contents even if CE reset is effected, such as security code.

Examples of this are given below.

Example 1.

```
R1      MEM     0.01H       ; First digit of key input data of security code
R2      MEM     0.02H       ; Second digit of key input data of security code
R3      MEM     0.03H       ; Data of first digit when security code is changed
R4      MEM     0.04H       ; Data of second digit when security code is changed
M1      MEM     0.11H       ; First digit of current security code
M2      MEM     0.12H       ; Second digit of current security code

START:
        ┌──────────────────────────────┐
        │Key input processing          │
        │R1 ← contents of key A        │ ; Security code input wait mode
        │R2 ← contents of key B        │ ; Substitutes contents of pressed key for R1 and R2
        └──────────────────────────────┘
        SET2    CMP, Z  ;<1>  ; Compares security code with input data
        SUB     R1, M1
        SUB     R2, R2
        SKT1    Z
        BR      ERROR         ; Input data is different from security code
MAIN:
        ┌──────────────────────────────┐
        │Key input processing          │
        │R3 ← contents of key C        │ ; Security code rewriting mode
        │R4 ← contents of key D        │ ; Substitutes contents of pressed key for R3 and R4
        └──────────────────────────────┘
        ST      M1, R3  ;<2>  ; Rewrites security code
        ST      M2, R4  ;<3>
        BR      MAIN
ERROR:
        ┌──────────────────────────────┐
        │        Do not operate        │
        └──────────────────────────────┘
```

In the above program, if the security code is “12H”, the contents of data memory areas M1 and M2 are “1H” and “2H”, respectively.

If CE reset is executed at this time, the contents of the key input are compared with security code “12H” in <1>. If they match, the normal processing is performed.

If the security code is changed in the main processing, the changed code is written to M1 and M2 in <2> and <3>.

If the security code is changed to “34H”, therefore, “3H” and “4H” are written to M1 and M2 in <2> and <3>.

If CE reset is executed when <2> has been executed, however, the program is started from address 0000H without <3> executed.

Therefore, the security code is “32H”. This makes it impossible to clear the security system.

In this case, use the program shown in Example 2.

Example 2.

```
R1          MEM     0.01H          ; First digit of key input data of security code
R2          MEM     0.02H          ; Second digit of key input data of security code
R3          MEM     0.03H          ; Data of first digit when security code is changed
R4          MEM     0.04H          ; Data of second digit when security code is changed
M1          MEM     0.11H          ; First digit of current security code
M2          MEM     0.12H          ; Second digit of current security code
CHANGE      FLG     0.13H.0        ; "1" while security code is changed

START:
            ┌──────────────────────┐
            │Key input processing  │
            │R1 ← contents of key A│  ; Security code input wait mode
            │R2 ← contents of key B│  ; Substitutes contents of pressed key for R1 and R2
            └──────────────────────┘
            SKT1    CHANGE  ;<4>   ; If CHANGE flag is "1"
            BR      SECURITY_CHK
            ST      M1, R3         ; Writes M1 and M2 again
            ST      M2, R4
            CLR1    CHANGE
SECURITY_CHK:
            SET2    CMP, Z  ;<1>   ; Compares security code with input data
            SUB     R1, M1
            SUB     R2, M2
            SKT1    Z
            BR      ERROR          ; Input data is different from security code
MAIN:
            ┌──────────────────────┐
            │Key input processing  │
            │R3 ← contents of key C│  ; Security code rewriting mode
            │R4 ← contents of key D│  ; Substitutes contents of pressed key for R3 and R4
            └──────────────────────┘
            SET1    CHANGE  ;<5>   ; Sets CHANGE flag to "1" until security code is
                                   ; completely changed
            ST      M1, R3  ;<2>   ; Rewrites security code
            ST      M2, R4  ;<3>
            CLR1    CHANGE         ; Resets CHANGE flag to "0" after security code
                                   ; has been changed
            BR      MAIN
ERROR:
            ┌──────────────────────┐
            │   Do not operate     │
            └──────────────────────┘
```

In the above program, the CHANGE flag is set to "1" in <5> before the security code is rewritten in <2> and <3>. Even if CE reset is executed before executing <3>, therefore, it is written again in <4>.

13.4 Power-ON Reset

Power-ON reset is executed when the supply voltage V_{DD} of the device rises from a specific level (power-ON clear voltage).

If the supply voltage V_{DD} is the same as the power-ON clear voltage or lower, the voltage detection circuit shown in Figure 13-1 outputs a power-ON clear signal (POC).

When the power-ON clear signal is output, the crystal oscillation circuit is stopped, and the device operation is stopped.

During the output of power-ON clear signal, $\overline{IRES}$, $\overline{RES}$ and $\overline{RESET}$ signals are output.

If the supply voltage V_{DD} exceeds the power-ON clear voltage, the power-ON clear signal is deasserted, and crystal oscillation is started. The $\overline{IRES}$, $\overline{RES}$, and $\overline{RESET}$ signals are also deasserted at the same time.

At this time, the halt status that is released by the basic timer 0 carry is set by the $\overline{IRES}$ signal. Therefore, power-ON reset is effected at the rising edge of the next basic timer 0 carry FF setting signal.

Because the basic timer 0 carry FF setting signal is initialized to 100 ms by the $\overline{RESET}$ signal, reset is effected 50 ms after the supply voltage V_{DD} has exceeded the power-ON clear voltage, and the program is started from address 0.

This operation is illustrated in Figure 13-4.

The program counter, stack, system registers, and control registers are initialized as soon as the power-ON clear signal is output.

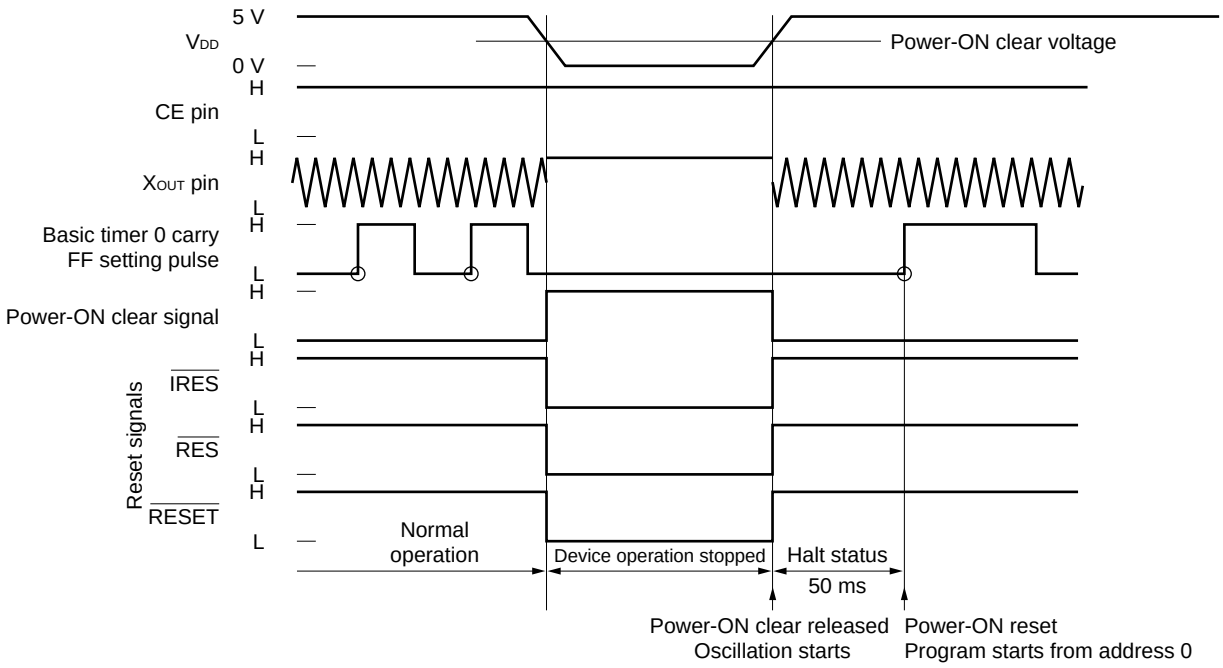
For the values to which the program counter, stack, system registers, and control registers are to be initialized, refer to the description of each register.

The power-ON clear voltage is 3.5 V (rated value) during normal operation, and 2.2 V (rated value) in the clock stop status.

The power-ON reset operations during normal operation and in the clock stop status are described in **13.4.1** and **13.4.2**.

13.4.3 describes the operation when the supply voltage V_{DD} rises from 0 V.

Figure 13-4. Power-ON Reset Operation



13.4.1 Power-ON reset during normal operation

Figure 13-5 (a) shows the operation.

As shown in this figure, the power-ON clear signal is output and the device operation is stopped regardless of the input level of the CE pin when the supply voltage V_{DD} drops below 3.5 V.

If the supply voltage V_{DD} rises above 3.5 V again, the program is started from address 0000H after a halt status of 50 ms.

The normal operation means the operation performed when the clock stop instruction is not used, and includes the halt status set by the halt instruction.

13.4.2 Power-ON reset in clock stop status

Figure 13-5 (b) shows the operation.

As shown in this figure, the power-ON clear signal is output and the device operation is stopped when the supply voltage V_{DD} drops below 2.2 V.

However, it does not seem that the device operation has been changed because the clock stop status is set.

When the supply voltage V_{DD} rises to 3.5 V or higher, the program is started from address 0000H after a halt status of 50 ms.

13.4.3 Power-ON reset when supply voltage V_{DD} rises from 0 V

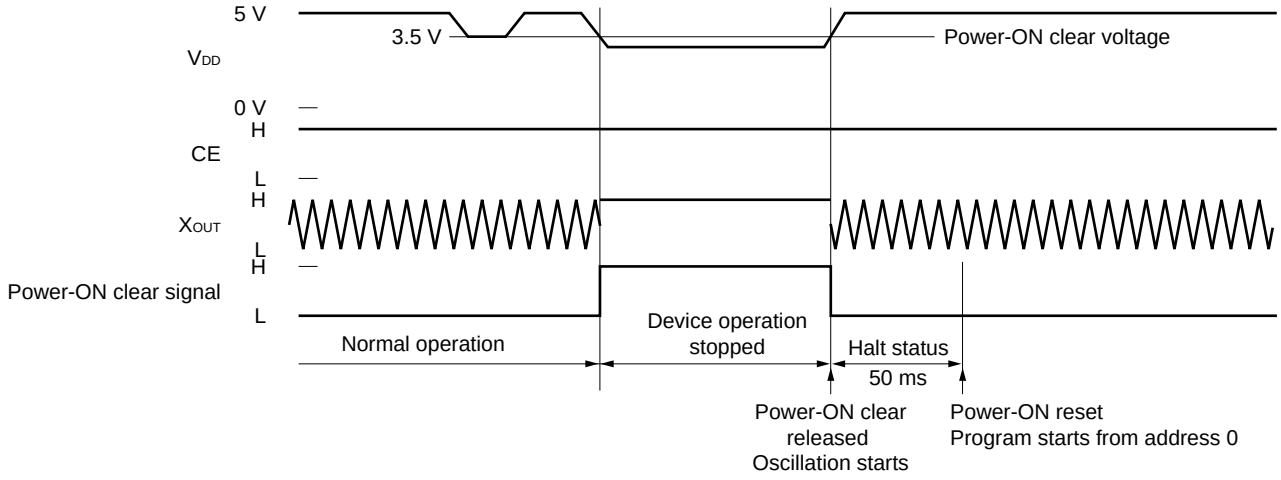
Figure 13-5 (c) shows the operation.

As shown in this figure, the power-ON clear signal is output before the supply voltage V_{DD} rises from 0 V to 3.5 V.

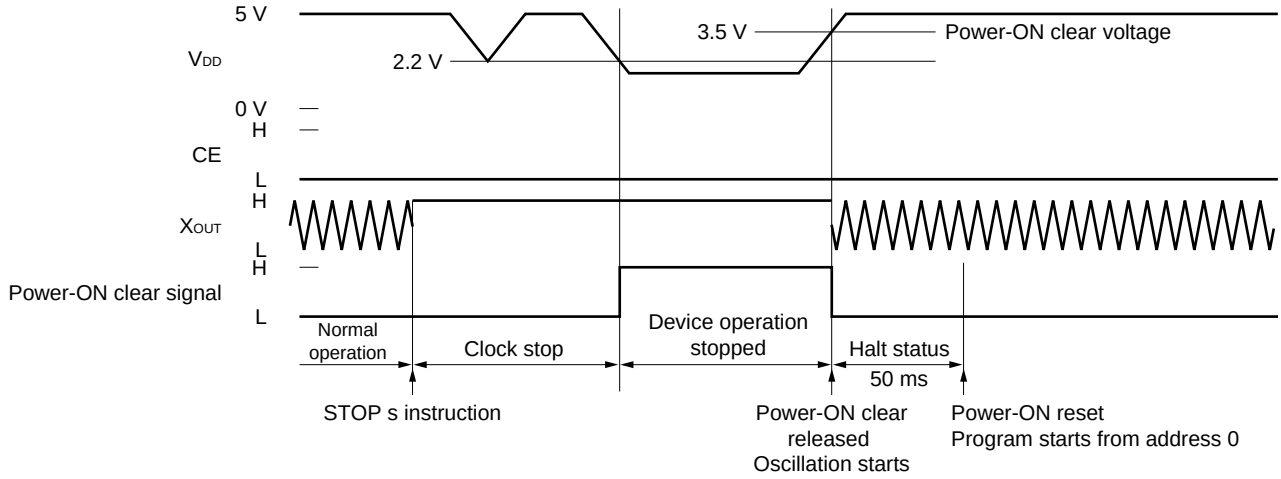
If the supply voltage V_{DD} exceeds the power-ON clear voltage, the crystal oscillation circuit starts operating, and the program is started from address 0000H after a halt status of 50 ms.

Figure 13-5. Power-ON Reset and Supply Voltage V_{DD}

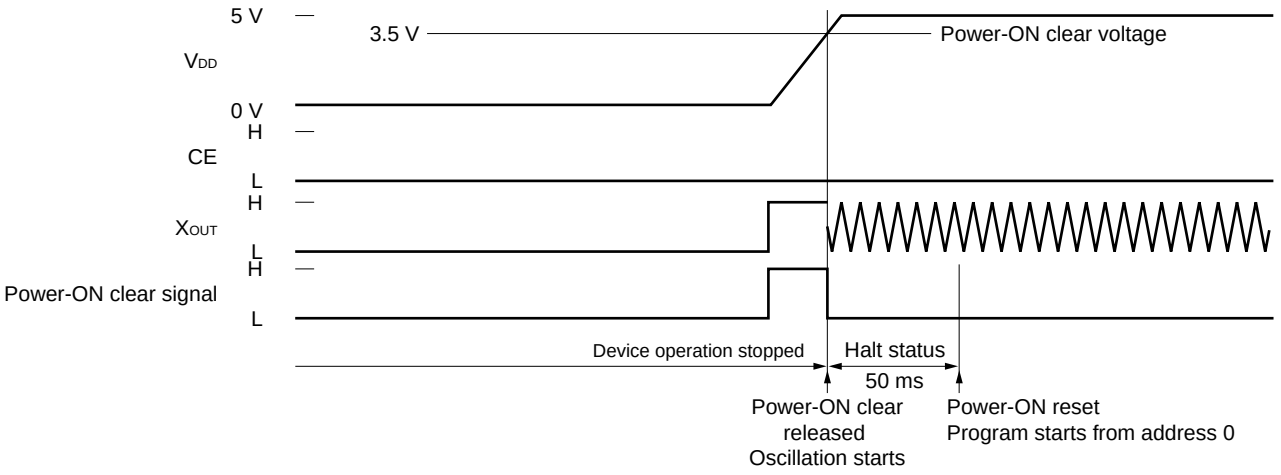
(a) Normal operation (including halt status)



(b) In clock stop status



(c) When supply voltage V_{DD} rises from 0 V



13.5 Relation between CE Reset and Power-ON Reset

On the first application of the supply voltage V_{DD}, power-ON reset and CE reset may be executed simultaneously.

The following **13.5.1** through **13.5.3** describe the reset operations performed at this time.

13.5.4 describes the points to be noted when raising the supply voltage V_{DD}.

13.5.1 If V_{DD} and CE pins simultaneously go high

Figure 13-6 (a) shows the operation.

At this time, the program is started from address 0000H by power-ON reset.

13.5.2 If CE pin goes high in forced halt status set by power-ON reset

Figure 13-6 (b) shows the operation.

At this time, the program is started from address 0000H by power-ON reset in the same manner as **13.5.1**.

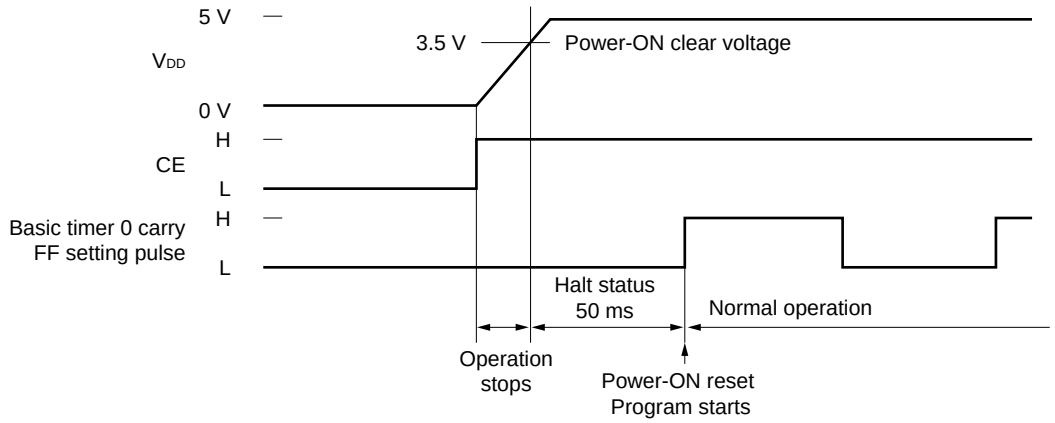
13.5.3 If CE pin goes high after power-ON reset

Figure 13-6 (c) shows the operation.

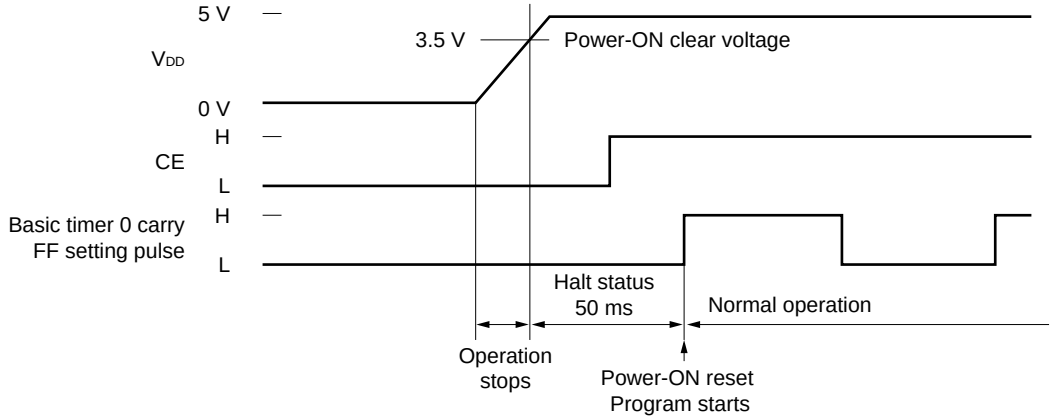
At this time, the program is started from address 0000H by power-ON reset, and is started again from address 0000H at the rise of the next basic timer 0 carry FF setting signal due to CE reset.

Figure 13-6. Relation between Power-ON Reset and CE Reset

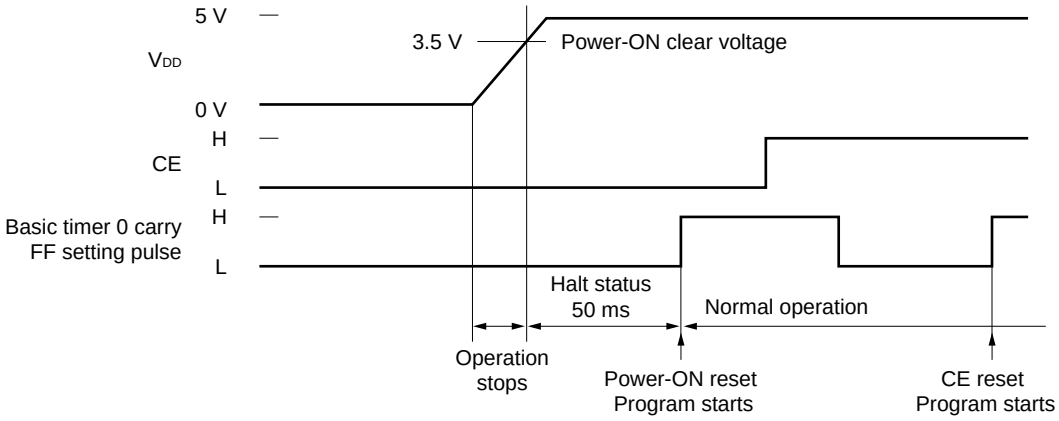
(a) If V_{DD} and CE pins simultaneously go high



(b) If CE pin goes high in halt status



(c) If CE pin goes high after power-ON reset



13.5.4 Notes on raising supply voltage V_{DD}

When raising the supply voltage V_{DD}, the following points (1) and (2) must be noted.

(1) To raise supply voltage V_{DD} from level below power-ON clear voltage

When raising the supply voltage V_{DD}, it must be raised to 3.5 V or higher once.

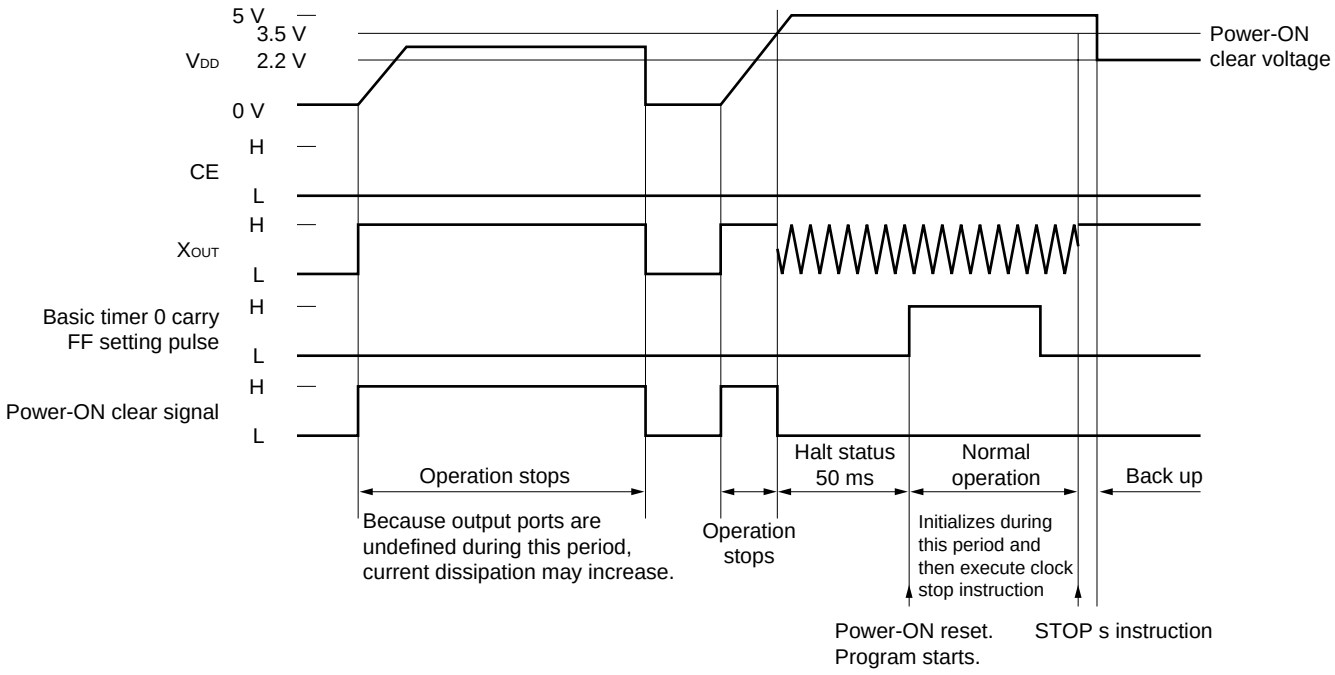
Figure 13-7 illustrates this.

As shown in this figure, if a voltage less than 3.5 V is applied on application of V_{DD} in a program, for example, that backs up V_{DD} at 2.2 V by using the clock stop instruction, the power-ON clear signal is continuously output and the program is not executed.

At this time, the output ports of the device output undefined values and, in some cases, the current dissipation increases.

This means that, if the device is backed up by batteries, the back-up time is substantially shortened.

Figure 13-7. Notes on Raising V_{DD}



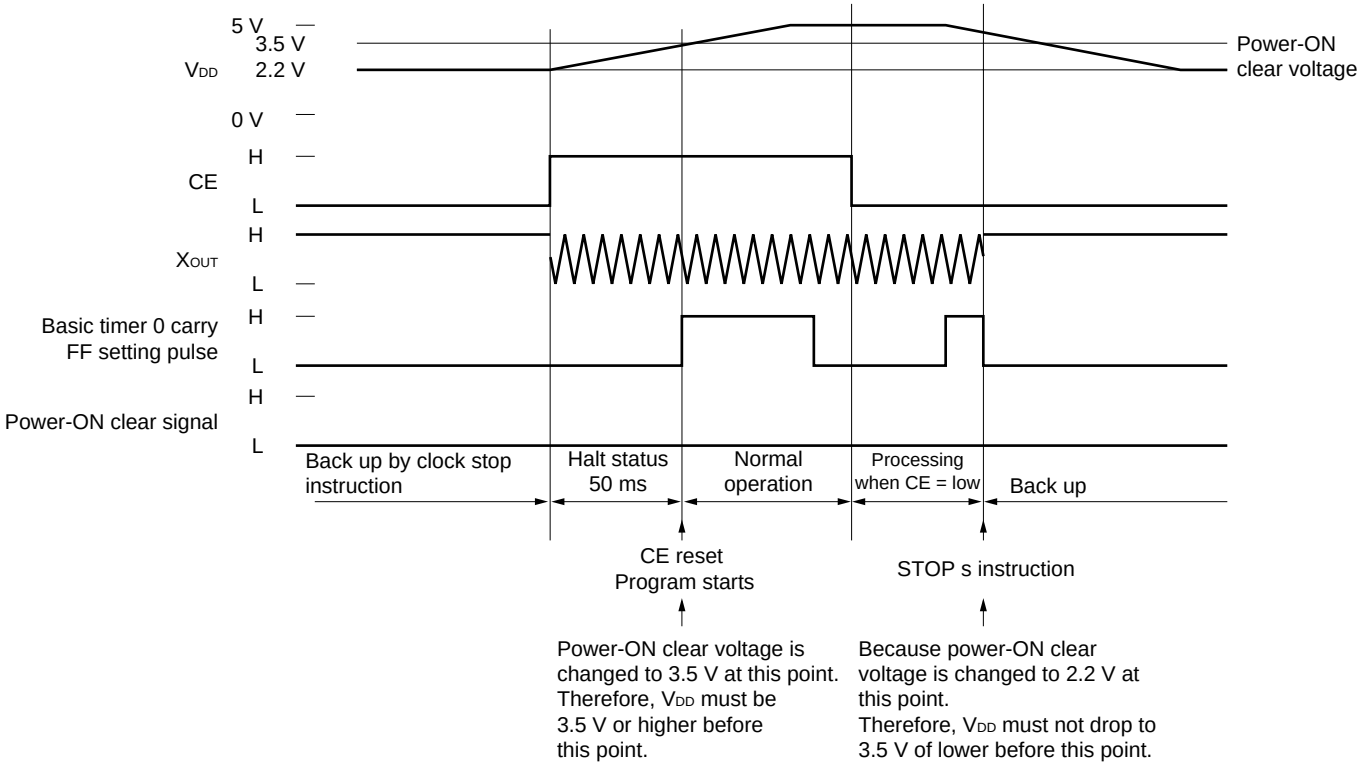
(2) Releasing clock stop status

To restore from the back-up status while the supply voltage V_{DD} is backed up at 2.2 V by using the clock stop instruction, V_{DD} must be raised to 3.5 V or higher within 50 ms after the CE pin has gone high.

As shown in Figure 13-8, CE reset is executed to release the clock stop status. Because the power-ON clear voltage is changed to 3.5 V, 50 ms after the CE pin has gone high, power-ON reset is executed unless V_{DD} is 3.5 V or higher at this point.

The same applies when lowering V_{DD}.

Figure 13-8. Releasing Clock Stop Status



13.6 Power Failure Detection

The power failure detection function is used to identify, when the device has been reset as shown in Figure 13-9, whether the device has been reset by application of supply voltage V_{DD} or by the CE pin.

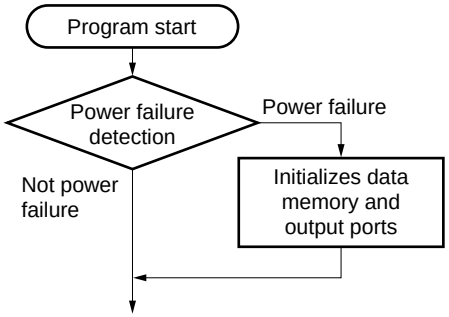
On power application, the contents of the data memory and output ports are “undefined”. These contents are initialized by using the power failure detection function.

A power failure is detected in two ways: by detecting the BTM0CY flag using the power failure detection circuit or by detecting the contents of the data memory (RAM judge).

The following **13.6.1** and **13.6.2** describe the power failure detection circuit and the method to detect a power failure by using the BTM0CY flag.

13.6.3 and **13.6.4** describe the RAM judge method to detect a power failure.

Figure 13-9. Power Failure Detection Flowchart



13.6.1 Power failure detection circuit

The power failure detection circuit consists of a voltage detection circuit, basic timer 0 carry disable flip-flop that is reset by the output (power-ON clear signal) of the power failure detection circuit, and basic timer 0 carry, as shown in Figure 13-1.

The basic timer 0 carry disable FF is set to 1 by the power-ON clear signal, and is reset to 0 when an instruction that reads the BTM0CY flag is executed.

When the basic timer 0 carry disable FF is set to 1, the BTM0CY flag is not set to 1.

If the power-ON clear signal is output (at power-ON reset), the program is started with the BTM0CY flag reset, and is disabled from being set until an instruction that reads the BTM0CY flag is executed.

Once this instruction has been executed, the BTM0CY flag is set each time the basic timer 0 carry FF setting pulse rises. The content of the BTM0CY flag is detected each time the device has been reset. If the flag is reset to 0, power-ON reset (power failure) has been executed. If it is set to 1, CE reset (not power failure) has been executed.

Because the voltage at which a power failure is detected is the same as the voltage at which power-ON reset is executed, V_{DD} = 3.5 V when the crystal oscillates, and V_{DD} = 2.2 V in the clock stop status.

Figure 13-10 shows the transition of the BTM0CY flag status.

Figure 13-11 shows the timing chart of Figure 13-10 and operation of the BTM0CY flag.

Figure 13-10. Status Transition of BTM0CY Flag

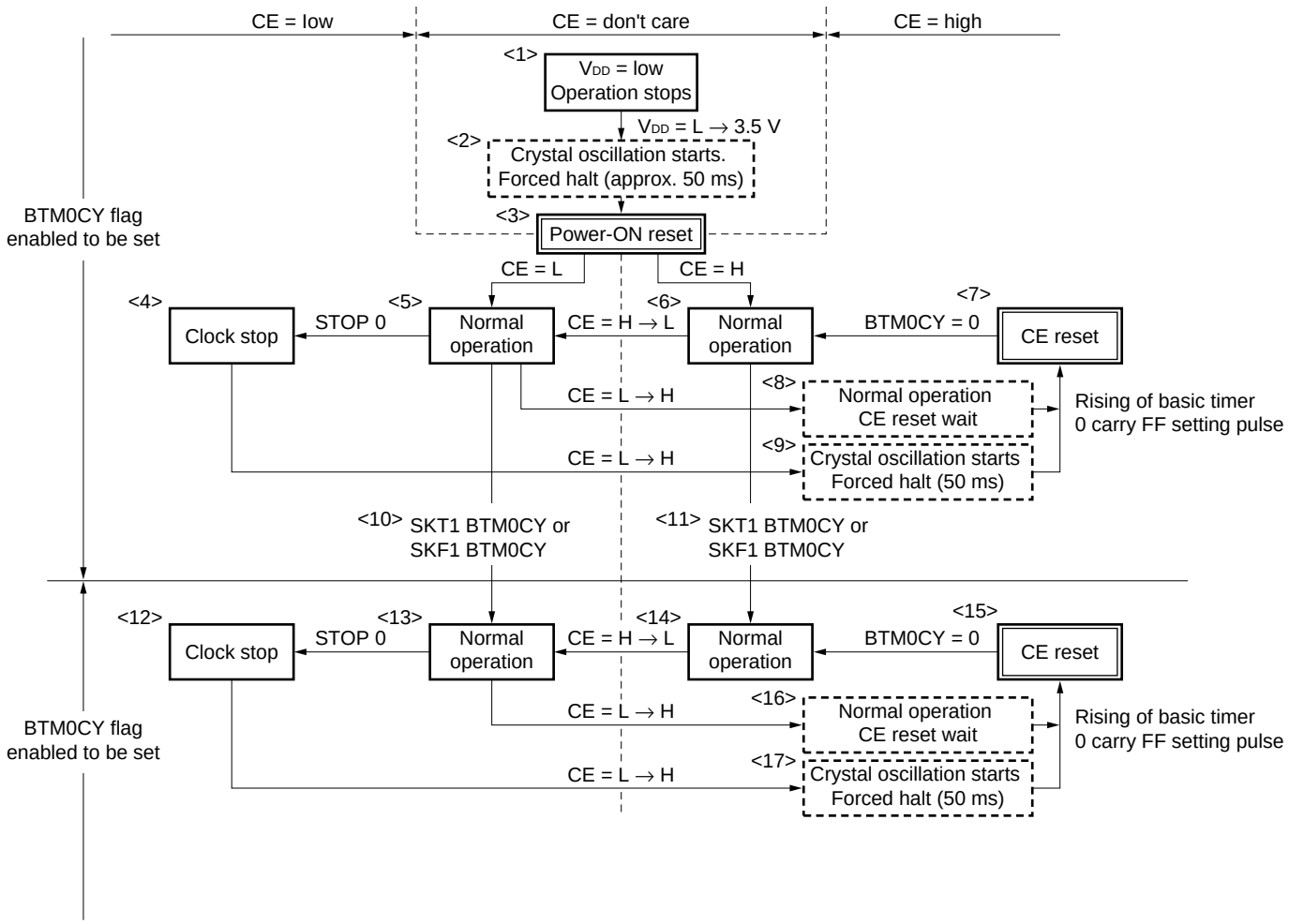
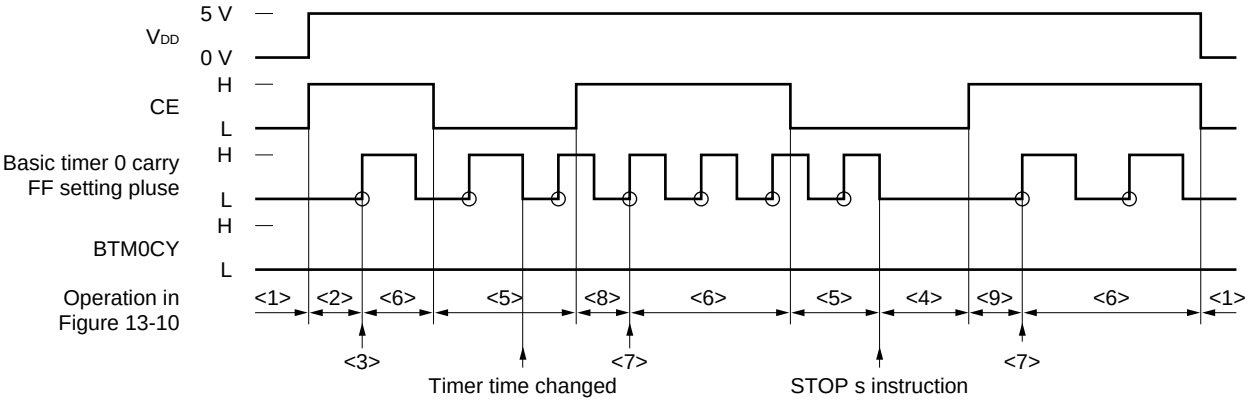
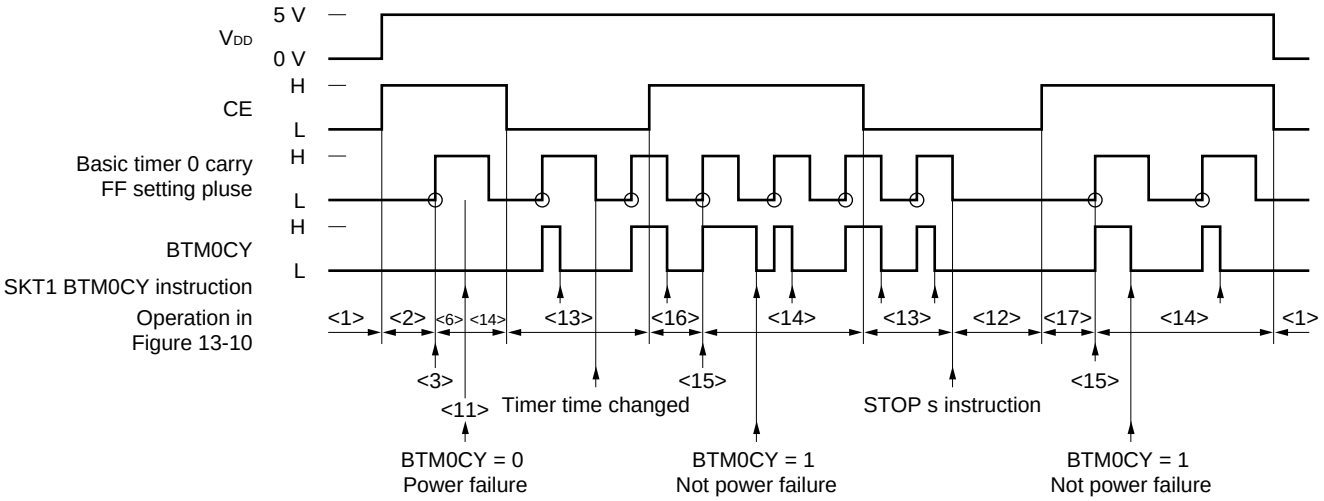


Figure 13-11. Operation of BTM0CY Flag

(a) If BTM0CY flag is never detected (SKT1 BTM0CY or SKF1 BTM0CY is not executed)



(b) To detect power failure by using BTM0CY flag



13.6.2 Notes on detecting power failure with BTM0CY flag

The following points must be remembered when counting the watch by using the BTM0CY flag.

(1) Updating watch

When creating a watch program by using the basic timer 0 carry, the watch must be updated after a power failure has been detected.

This is because the BTM0CY flag is reset to 0 for the BTM0CY flag to be read at power failure detection, and consequently, one watch count is missed.

(2) Watch updating processing time

Updating the watch must be completed until the next basic timer 0 carry FF setting pulse rises.

This is because if the CE pin goes high during the watch updating processing, CE reset is executed without the processing completed.

For the details of (1) and (2) above, refer to **11.3.7 (b) Adjusting basic timer 0 carry on CE reset**.

When performing processing on power failure, the following points must be noted.

(3) Timing of power failure detection

To count the watch by using the BTM0CY flag, it must be carried out since the BTM0CY flag has been read to detect a power failure and the program has been started from address 0000H until the basic timer 0 carry FF setting pulse rises next time.

This is because if the basic timer 0 carry FF setting time is set, say, to 5 ms and a power failure is detected 6 ms after the program was started, the BTM0CY flag is skipped once.

For more information, refer to **11.3.7 (b) Adjusting basic timer 0 carry on CE reset**.

As shown in the example below, power failure detection and initialization processing must be performed within the basic timer 0 carry FF setting time.

This is because if the CE pin goes high and CE reset is executed during power failure processing and initialization processing, these processing is aborted, and troubles may occur.

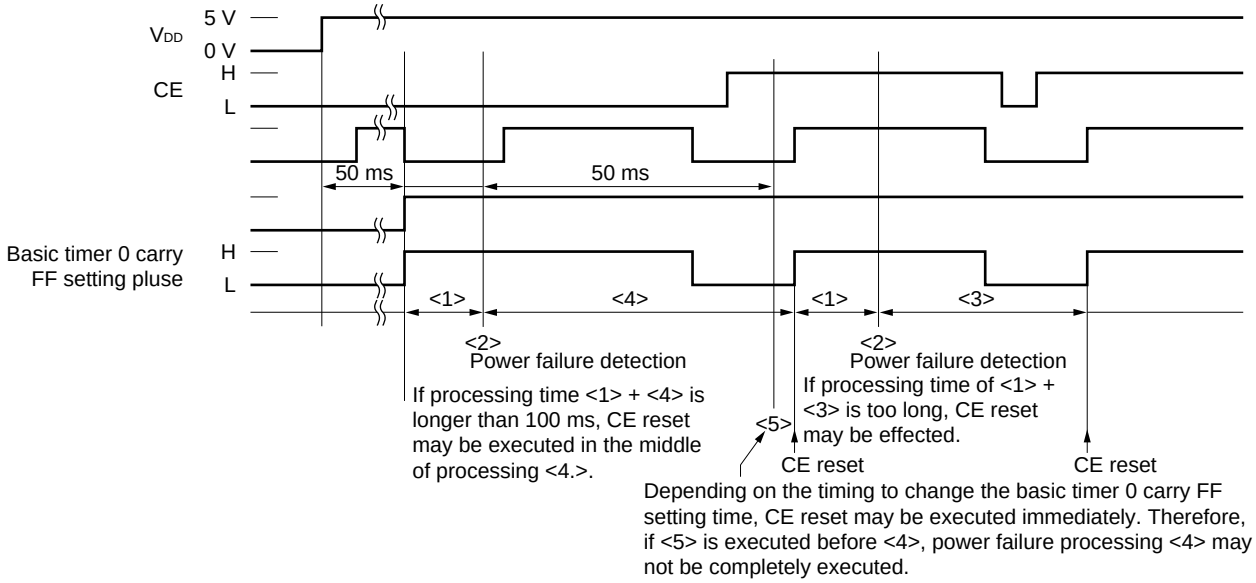
To change the basic timer 0 carry FF setting time during initialization processing, the instruction that changes the time must be executed at the end of the initialization processing, and the instruction must be one instruction.

This is because if the setting time of the basic timer 0 carry FF is changed before initialization processing, the initialization processing may not be completely executed because CE reset may be effected, as shown in the example below.

Example Program example

```
START:                          ; Program address 0000H
  ;<1>
      Processing on reset
  ;<2>
      SKT1    BTM0CY    ; Power failure detection
      BR      INITIAL
BACKUP:
  ;<3>
      Updating watch
      BR      MAIN
INITIAL:
  ;<4>
      Initialization processing
  ;<5>
      INITFLG BTM0CK1, NOT BTM0CK0 ; Embedded macro
                                   ;  Sets basic timer 0 carry FF setting time to 5 ms
MAIN:
      Main processing
      SKT1    BTM0CY
      BR      MAIN
      Updating watch
      BR      MAIN
```

Operation example



13.6.3 Power failure detection by RAM judge method

The RAM judge method detects a power failure by making a judgment whether the contents of the data memory at specified addresses are as specified when the device is reset.

An example of a program that detects a power failure by the RAM judge method is shown below.

The contents of the data memory are “undefined” on power supply voltage V_{DD} application. A power failure is detected by comparing the “undefined” value with a “specified” value.

In some cases, a wrong judgment on power failure detection may be made as described in **13.6.4 Notes on detecting power failure by RAM judge method**.

The advantage of using the RAM judge method is that a lower supply voltage can be backed up than the level at which the power failure detection circuit detects a power failure, as shown in Table 13-2.

Table 13-2. Comparison between Power Failure Detection by Power Failure Detection Circuit and RAM Judge Method

	Power Failure Detection Circuit		RAM Judge	
Data retention voltage (at clock stop)	Effective value	Rated value	Effective value	Rated value
	1–2 V	2.2 V	0–1 V	2.0 V
Operating status	No miss-operation		Miss-operation possible	

Example Program that detects power failure by RAM judge method

```
M012        MEM     0.12H
M034        MEM     0.34H
M056        MEM     0.56H
M107        MEM     1.07H
M128        MEM     1.28H
M16F        MEM     1.6FH
DATA0       DAT     1010B
DATA1       DAT     0101B
DATA2       DAT     0110B
DATA3       DAT     1001B
DATA4       DAT     1100B
DATA5       DAT     0011B

START:
            SET2    CMP, Z
            SUB     When M012, #DATA0   ; M012 = DATA0 and
            SUB     when M034, #DATA1   ; M034 = DATA1 and
            SUB     when M056, #DATA2   ; M056 = DATA2 and
            BANK1
            SUB     when M107, #DATA3   ; M107 = DATA3 and
            SUB     when M128, #DATA4   ; M128 = DATA4 and
            SUB     when M16F, #DATA5   ; M16F = DATA5,
            BANK0
            SKF1    Z
            BR      BACKUP          ; branches to BACKUP
; INITIAL:
            Initialization processing
            MOV     M012, #DATA0
            MOV     M034, #DATA1
            MOV     M056, #DATA2
            BANK1
            MOV     M107, #DATA3
            MOV     M128, #DATA4
            MOV     M16F, #DATA5
            BR      MAIN
BACKUP:
            Backup processing
MAIN:
            Main processing
```

13.6.4 Notes on detecting power failure by RAM judge method

Because the value of the data memory on application of supply voltage V_{DD} is basically "undefined", the following points (1) and (2) must be noted.

(1) Data to be compared

Where the number of bits of the data memory to be compared by the RAM judge method is "n", the probability at which the value of the data memory on application of V_{DD} happens to coincide with the value to be compared is $(1/2)^n$.

To detect a power failure by the RAM judge method, therefore, back up is judged at a probability of $(1/2)^n$. To reduce this probability, as many bits as possible must be compared.

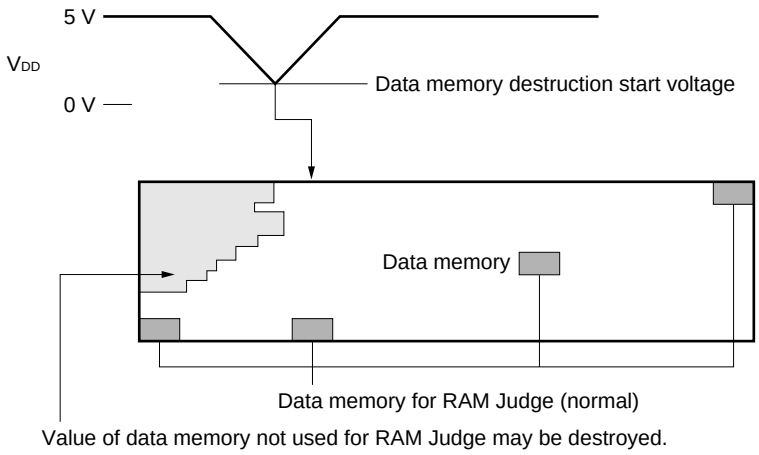
The contents of the data memory on application of V_{DD} are likely to be the same value such as "0000B" and "1111B". Therefore, the data with which the data memory contents are to be compared should be a mix of "0" and "1" such as "1010B" and "0110B" to reduce the possibility of misjudgement.

(2) Notes on program

As shown in Figure 13-12, if a voltage V_{DD} rises from the level at which destruction of the data memory starts, even if the value of the data memory to be compared is normal, the other portions may be destroyed.

At this time, back up is judged by the RAM judge method. It is therefore necessary to take measures to prevent a program hang-up even if the data memory is destroyed.

Figure 13-12. V_{DD} and Data Memory Destruction



14. PLL FREQUENCY SYNTHESIZER

The PLL (Phase Locked Loop) frequency synthesizer is used to lock a frequency in the MF (Medium Frequency), HF (High Frequency), and VHF (Very High Frequency) bands at a specific frequency by means of phase comparison.

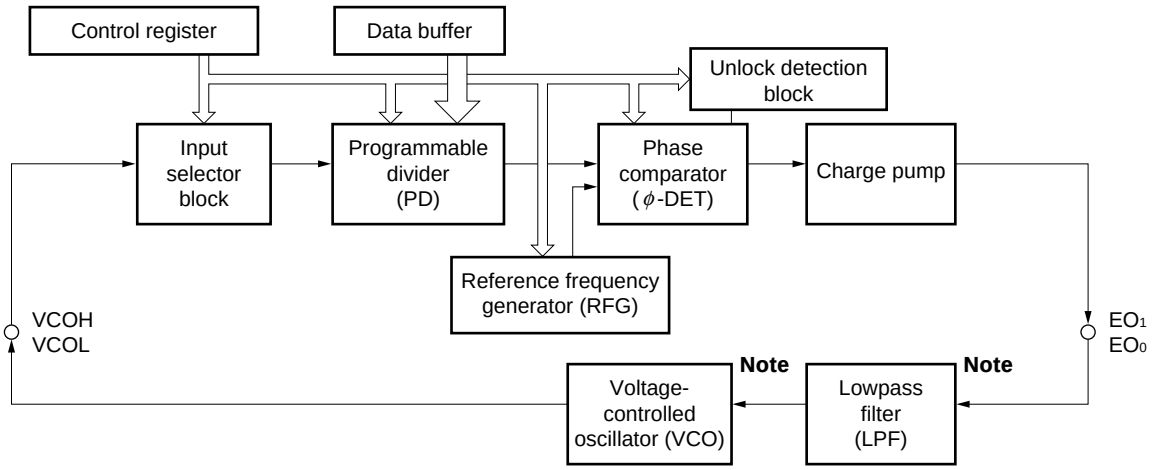
14.1 Configuration of PLL Frequency Synthesizer

Figure 14-1 shows the block diagram of the PLL frequency synthesizer.

As shown in this figure, the PLL frequency synthesizer consists of an input selector block, a programmable divider (PD), a phase comparator (ϕ-DET), a reference frequency generator (RFG), and a charge pump.

By connecting these blocks with an external lowpass filter (LPF) and voltage-controlled oscillator (VCO), a PLL frequency synthesizer can be configured.

Figure 14-1. Block Diagram of PLL Frequency Synthesizer



Note External circuits

14.2 Functional Outline of PLL Frequency Synthesizer

The PLL frequency synthesizer divides the signal input from the VCOH (pin 32) or VCOL (pin 31) pin by using the programmable divider and outputs a phase difference between the input signal and a reference frequency from the EO_1 and EO_0 pins.

The PLL frequency synthesizer operates only when the CE pin is high. It is disabled when the CE pin is low.

For the details of the PLL disabled status, refer to **14.6**.

The following **14.2.1** through **14.2.5** outline the functions of the each block of the PLL frequency synthesizer.

14.2.1 Input selector block

This block selects a pin from which a signal output by an external voltage-controlled oscillator is input.

As the input pin, either the VCOH or VCOL pin is selected by the PLL mode select register (PLLMODE: RF address 21H).

For the details, refer to **14.3**.

14.2.2 Programmable divider

The programmable divider divides the signal input from the VCOH or VCOL pin by a ratio set by the program.

As the division mode, direct division or pulse swallow mode can be selected by using the PLL mode select register.

The division ratio is set by the PLL data register (PLLR: peripheral address 41H) via data buffer.

For the details, refer to **14.3**.

14.2.3 Reference frequency generator

The reference frequency generator generates a reference frequency against which the signal input to the PLL frequency synthesizer is to be compared by the phase comparator.

Twelve reference frequencies can be selected by the PLL reference clock select register (PLLRFCLK: RF address 31H).

For the details, refer to **14.4**.

14.2.4 Phase comparator and unlock detection block

The phase comparator compares the division signal output by the programmable divider with the signal from the reference frequency generator and outputs a phase difference between the two signals.

The unlock detection block detects the unlock status of the PLL.

The unlock status of the PLL is detected by the PLL unlock FF sensibility select register (PLLULSEN: RF address 15H) and PLL unlock FF judge register (PLLULJDG: RF address 05H).

For the details, refer to **14.5**.

14.2.5 Charge pump

The charge pump outputs the signal output by the phase comparator from the EO_1 and EO_0 pins as a high-level, low-level, or floating output.

For the details, refer to **14.5**.

14.3 Input Selector Block and Programmable Divider

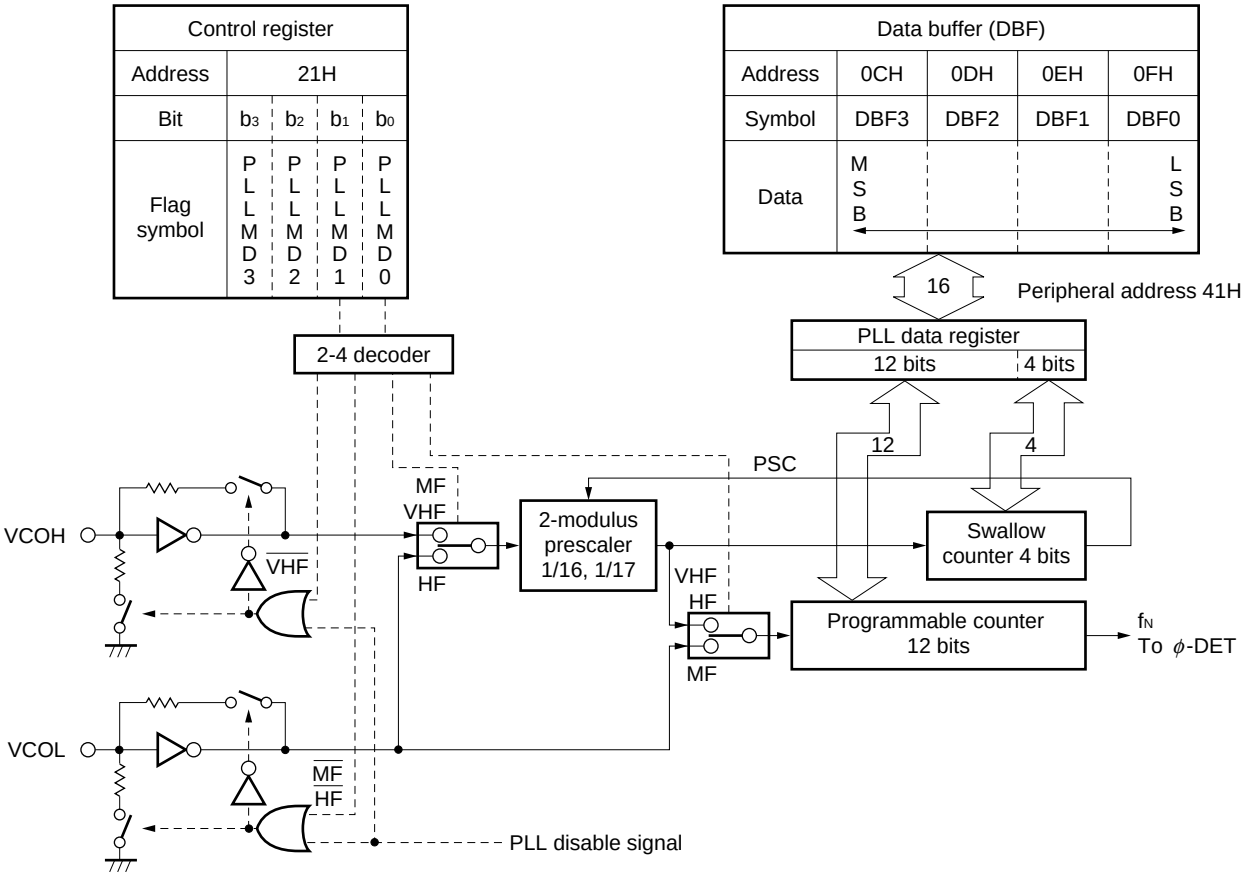
14.3.1 Configuration of input selector block and programmable divider

Figure 14-2 shows the configuration of the input selector block and programmable divider.

As shown in this figure, the input selector block consists of the VCOH and VCOL pins, and the input amplifiers of the respective pins.

The programmable divider consists of a 2-modulus prescaler, a swallow counter, a programmable counter, and a division mode selector switch.

Figure 14-2. Configuration of Input Selector Block and Programmable Divider



14.3.2 Function of input selector block and programmable divider

The input selector block and programmable divider selects the input pin and division mode of the PLL frequency synthesizer.

As the input pin, the VCOH or VCOL pin can be selected.

The voltage of the selected pin is at the intermediate level (about 1/2 V_{DD}). The pin not selected is internally pulled down.

Signals are input to these pins via an AC amplifier. Connect a capacitor in series to the pin to cut off the DC component of the input signal.

As the division mode, direct division or pulse swallow mode can be selected.

The programmable divider divides the input frequency in division mode according to the value set to the swallow counter or programmable counter.

Table 14-1 shows the input pins (VCOH and VCOL) and division modes.

The input pin and division mode to be used are selected by the PLL mode select register.

14.3.3 describes the configuration and function of the PLL mode select register.

The division ratio is set to the programmable divider by the PLL data register via data buffer.

14.3.4 describes the programmable divider and PLL data register.

Table 14-1. Input Pins and Division Methods

Division Method	Pin Used	Input Frequency (MHz)	Input Amplitude (V_{P-P})	Division Ratio Set	Division Ratio Set to Data Buffer
Direct division (MF)	VCOL	0.5 - 30	0.3	16 to 2^{12}–1	010×H-FFF×H (×: lower 4 bits are don't care)
Pulse swallow (HF)	VCOL	5 - 40	0.3	256 to 2^{16}–1	0100H-FFFFH
Pulse swallow (VHF)	VCOH	9 - 150	0.3	256 to 2^{16}–1	0100H-FFFFH

14.3.3 Configuration and function of PLL mode select register (PLLMODE)

The PLL mode select register sets the division mode of the PLL frequency synthesizer and the pin to be used.

The configuration and function of the PLL mode select register are illustrated below.

The following paragraphs (1) through (4) outlines the respective division modes.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
PLL mode select register (PLLMODE)	PLLMD3	PLLMD2	PLLMD1	PLLMD0	21H	R/W

		Sets division mode of PLL frequency synthesizer
0	0	Disables VCOL and VCOH pins
0	1	Direct division (VCOL pin MF mode)
1	0	Pulse swallow (VCOH pin VHF mode)
1	1	Pulse swallow (VCOL pin HF mode)

Fixed to "0"

On reset	Power-ON	0	0	0	0
	Clock stop			0	0
	CE			Retained	

(1) Direct division mode (MF)

In this mode, the VCOL pin is used.

The VCOH pin is pulled down.

In the direct division mode, the frequency is divided only by using the programmable counter.

(2) Pulse swallow mode (HF)

In this mode, the VCOL pin is used.

The VCOH pin is pulled down.

In the pulse swallow mode, the frequency is divided by using the swallow counter and programmable counter.

(3) Pulse swallow mode (VHF)

In this mode, the VCOH pin is used.

The VCOL pin is pulled down.

In the pulse swallow mode, the frequency is divided by using the swallow counter and programmable counter.

(4) VCOL and VCOH pin disabled mode

In this mode, both the VCOH and VCOL pins are internally pulled down.

However, the phase comparator, reference frequency generator, and charge pump operate.

Therefore, the operation in this mode is different from that in the PLL disabled status described later.

14.3.4 Programmable divider and PLL data register

The programmable divider divides the signal input from the VCOH or VCOL pin by the value set to the swallow counter or programmable counter.

The swallow counter and programmable counter are 4-bit and 12-bit binary down-counters, respectively.

A division ratio is set to the swallow counter and programmable counter by the PLL data register (PLLR: peripheral address 41H) via data buffer.

Data is set to or read from the PLL data register by using the “PUT PLLR, DBF” or “GET DBF, PLLR” instruction.

The division ratio is called “N value”.

For setting the division ratio (N value) in each division mode, refer to **14.7**.

(1) PLL data register and data buffer

The relation between the PLL data register and data buffer is described next.

In the direct division mode, the higher 12 bits of the PLL data register are valid, and all the 16 bits are valid in the pulse swallow mode.

In the direct division mode, all the 12 bits of the PLL data register are set to the programmable counter.

In the pulse swallow mode, the higher 12 bits are set to the programmable counter, and the lower 4 bits are set to the swallow counter.

(2) Relation between division ratio N and divided output frequency of programmable divider

The relation between the value “N” set to the PLL data register and the frequency “f_N” of the signal divided and output by the programmable divider is as shown below.

For details, refer to **14.7**.

(a) Direct division mode (MF)

$$f_N = \frac{f_{IN}}{N} \quad \text{N: 12 bits}$$

(b) Pulse swallow mode (HF, VHF)

$$f_N = \frac{f_{IN}}{N} \quad \text{N: 16 bits}$$

Name	Data Buffer															
Symbol	DBF3				DBF2				DBF1				DBF0			
Address	0CH				0DH				0EH				0FH			
Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data	Transfer data															

16

GET can be executed

PUT can be executed

Peripheral Register																		
Name	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Symbol	Peripheral address	Peripheral hardware
PLL data register	Valid data																PLLR	41H	PLL frequency synthesizer

			Sets division ratio of PLL frequency synthesizer
Direct division mode	0	Don't care	Setting prohibited
	15 (00FH)	Don't care	
	16 (010H)	Don't care	Division ratio N: N = x
	x	Don't care	
	$12^{12} - 1$ (FFFH)	Don't care	
Pulse swallow mode	0		Setting prohibited
	255 (00FFH)		
	256 (0100H)		Division ratio N: N = x
	x		
	$12^{16} - 1$ (FFFFH)		

14.4 Reference Frequency Generator

14.4.1 Configuration and function of reference frequency generator

Figure 14-3 shows the configuration of the reference frequency generator.

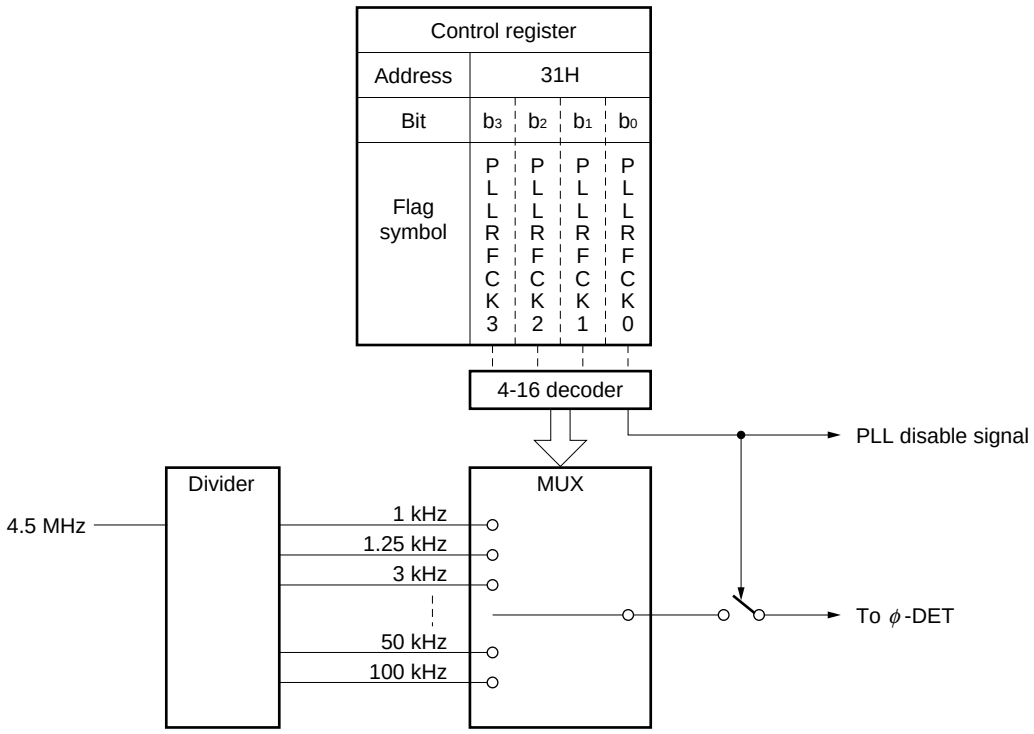
As shown in this figure, the reference frequency generator divides 4.5 MHz crystal oscillation to generate the reference frequency "f_r" of the PLL frequency synthesizer.

Twelve types of reference frequency f_r can be selected: 1, 1.25, 2.5, 3, 5, 6.25, 9, 10, 12.5, 25, 50, and 100 kHz.

The reference frequency f_r is selected by the PLL reference clock select register.

The following **14.4.2** describes the configuration and function of the PLL reference clock select register.

Figure 14-3. Configuration of Reference Frequency Generator (RFG)



14.4.2 Configuration and function of PLL reference clock select register (PLLRFCLK)

Name	Flag Symbol b_3	b_2	b_1	b_0	Address	Read/ Write
PLL reference clock select register (PLLRFCLK)	PLLRFCK3	PLLRFCK2	PLLRFCK1	PLLRFCK0	31H	R/W

b_3	b_2	b_1	b_0	Sets reference frequency f_r of PLL frequency synthesizer
0	0	0	0	1.25 kHz
0	0	0	1	2.5 kHz
0	0	1	0	5 kHz
0	0	1	1	10 kHz
0	1	0	0	6.25 kHz
0	1	0	1	12.5 kHz
0	1	1	0	25 kHz
0	1	1	1	50 kHz
1	0	0	0	3 kHz
1	0	0	1	Setting prohibited
1	0	1	0	Setting prohibited
1	0	1	1	Setting prohibited
1	1	0	0	1 kHz
1	1	0	1	9 kHz
1	1	1	0	100 kHz
1	1	1	1	PLL disabled

On reset	Power-ON	1	1	1	1	
	Clock stop	1	1	1	1	
	CE	Retained				

When the PLL disabled status is selected by the PLL reference clock select register, the VCOH and VCOL pins are internally pulled down.

The EO_1 and EO_0 pins are floated.

For the details of the PLL disabled status, refer to **14.6**.

14.5 Phase Comparator (φ-DET), Charge Pump, and Unlock Detection Block

14.5.1 Configuration of phase comparator, charge pump, and unlock detection block

Figure 14-4 shows the configuration of the phase comparator, charge pump, and unlock detection block.

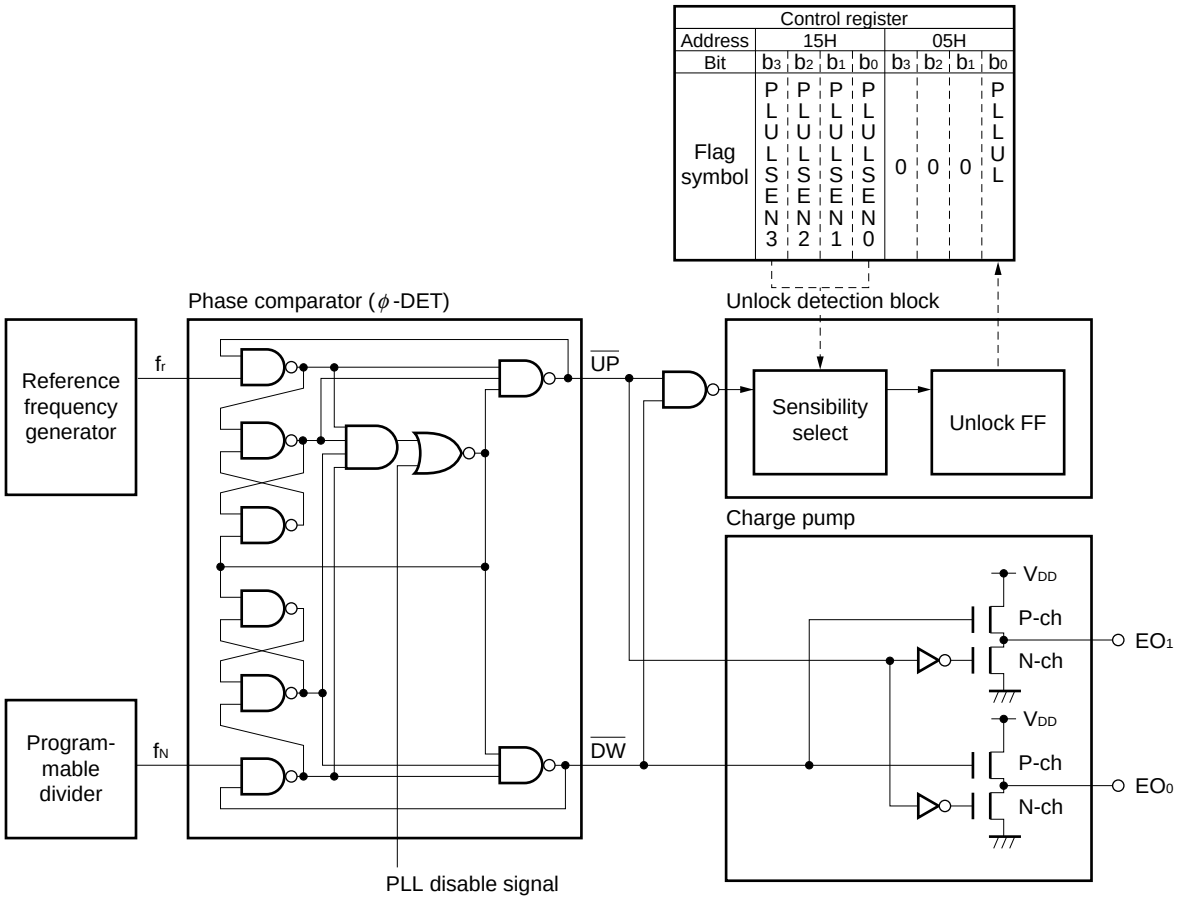
The phase comparator compares the phase of the divided frequency "f_N" output by the programmable divider with that of the reference frequency "f_r" output by the reference frequency generator, and outputs an up request signal ($\overline{UP}$) and down request signal ($\overline{DW}$).

The charge pump outputs the signal output by the phase comparator from the error out pins (EO_1 and EO_0 pins).

The unlock detection block consists of a sensibility select circuit and an unlock FF, and detects the unlock status of the PLL frequency synthesizer.

The following **14.5.2**, **14.5.3**, and **14.5.4** respectively describe the operations of the phase comparator, charge pump, and unlock detection block.

Figure 14-4. Configuration of Phase Comparator, Charge Pump, and Unlock Detection Block



14.5.2 Function of phase comparator

As shown in Figure 14-4, the phase comparator compares the phase of the divided frequency "f_N" output by the programmable divider with the phase of the reference frequency "f_r" and outputs an up request or a down request signal.

If the divided frequency f_N is lower than the reference frequency f_r, it outputs an up request signal; if f_N is higher than f_r, it outputs a down signal.

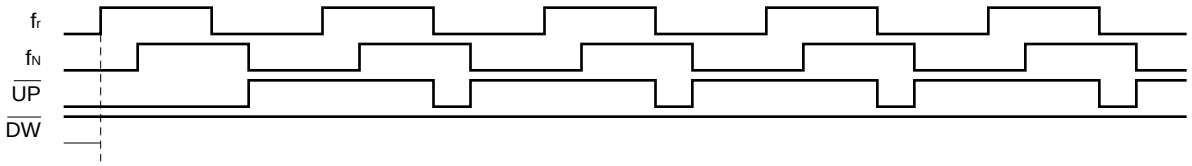
Figure 14-5 shows the relations among reference frequency f_r, divided frequency f_N, up request signal, and down request signal.

In the PLL disabled status, neither the up nor down request signal is output.

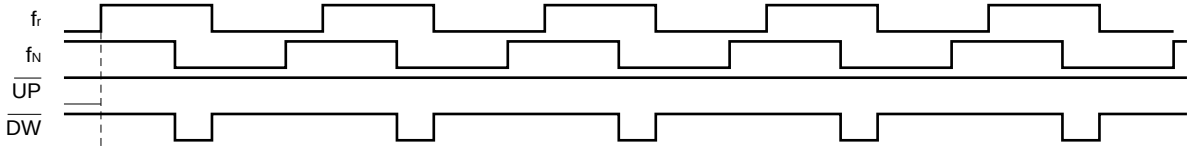
The up and down request signals are input to the charge pump and unlock detection block.

Figure 14-5. Relation among f_r, f_N, and $\overline{UP}$ and $\overline{DW}$ Signals

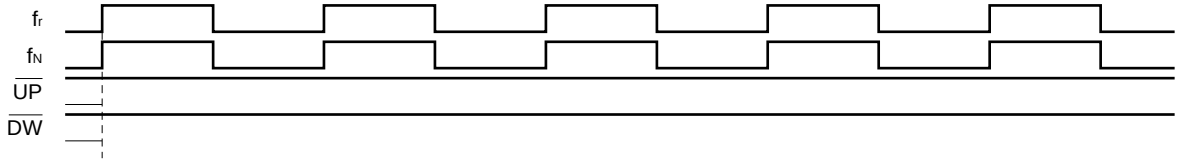
(a) If f_N lags behind f_r in phase



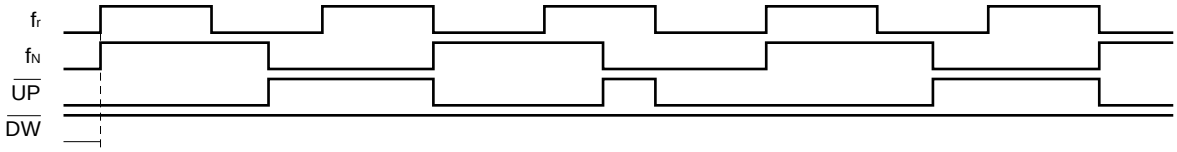
(b) If f_N advances f_r in phase



(c) If f_N and f_r are in the same phase



(d) If f_N is lower than f_r



14.5.3 Charge pump

As shown in Figure 14-4, the charge pump outputs the up or down request signal from the phase comparator, from to the error out pins (EO_1 and EO_0 pins).

Therefore, the relation among the output of the error out pins, divided frequency f_N, and reference frequency f_r is as follows:

When reference frequency f_r > divided frequency f_N: low level output

When reference frequency f_r < divided frequency f_N: high level output

When reference frequency f_r = divided frequency f_N: floating

14.5.4 Unlock detection block

As shown in Figure 14-4, the unlock detection block detects the unlock status of the PLL frequency synthesizer from the up request or down request signal of the phase comparator.

In the unlock status, either the up request or down request signal outputs low level, and the unlock status is detected by this low-level signal.

In the unlock status, the unlock flip-flop (FF) is set to 1.

The status of the unlock FF is detected by the PLL unlock FF judge register (refer to **14.5.5**).

The unlock FF is set at the cycle of reference frequency f_r selected at that time.

It is reset when the contents of the PLL unlock FF judge register are read by using the PEEK instruction (Read & Reset).

Therefore, the unlock FF must be detected in a cycle longer than the cycle $1/f_r$ of the reference frequency fr.

The unlock sensibility select circuit controls the status in which the unlock FF is set by delaying the up request or down request signal of the phase comparator.

If the signal is delayed for the longer time, the unlock FF is not set even if there is a large phase difference between the divided frequency f_N and reference frequency fr.

The delay time of the unlock sensibility select circuit is set by the PLL unlock FF sensibility select register (refer to **14.5.6**).

14.5.5 Configuration and function of PLL unlock FF judge register (PLLULJDG)

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
PLL unlock FF Judge register (PLLULJDG)	0	0	0	PLLUL	05H	R & Reset

Detects status of unlock FF	
0	Unlock FF = 0: PLL lock status
1	Unlock FF = 1: PLL unlock status

Fixed to "0"

On reset	Power-ON	0	0	0	Un-de-fined
	Clock stop				Re-tained
	CE				Re-tained

This register is a read-only register, and is reset when its contents are read to the window register by the “PEEK” instruction.

Because the unlock FF is set in the cycle of the reference frequency f_r, the contents of this register must be read to the window register in a cycle longer than reference frequency cycle $1/f_r$.

14.5.6 Configuration and function of PLL unlock FF sensibility select register (PLULSEN)

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
PLL unlock FF sensibility select register (PLULSEN)	PLULSEN3	PLULSEN2	PLULSEN1	PLULSEN0	15H	R/W

Sets delay time between reference frequency f_r and divided frequency f_N necessary for setting unlock FF

0	0	0.9 to 1.0 μs or longer
0	1	1.9 to 2.0 μs or longer
1	0	0.45 to 0.55 μs or longer
1	1	Unlock FF disable (always PLLUL = 1)

Fixed to "0"

On reset	Power-ON	0	0	0	0
	Clock stop			0	0
	CE			Retained	

When the unlock FF disable status is set, the unlock FF is always set. If the lock status of the PLL is detected by the PLL unlock FF judge register, it is always in the unlock status (PLLUL flag = 1).

14.6 PLL Disabled Status

The PLL frequency synthesizer stops operating (disabled) while the CE pin (pin 13) is low.

It also stops when the PLL disabled status is selected by the PLL reference mode select register.

Table 14-2 shows the operations of the respective blocks in the PLL disabled status.

When the VCOL and VCOH pins are disabled by the PLL mode select register, only the VCOL and VCOH pins are internally pulled down, and the other blocks operate.

The PLL reference mode select register and PLL mode select register are not initialized (retain the previous status) at CE reset. Therefore, when the CE pin goes high after the CE pin has gone low once and the PLL has been disabled, these registers return to the previous status.

If it is necessary to disable the PLL at CE reset, initialize the PLL by program.

The PLL is disabled at power-ON reset.

Table 14-2. Operations of Respective Block in PLL Disabled Status

Condition / Block	CE Pin = Low Level (PLL Disabled)	CE Pin = High Level	
		PLLRFCLK = 1111B (PLL disabled)	PLLMODE = 0000B (VCOH and VCOL disabled)
VCOL and VCOH pins	Internally pulled down	Internally pulled down	Internally pulled down
Programmable counter	Division stopped	Division stopped	Operates
Reference frequency generator	Output stopped	Output stopped	Operates
Phase comparator	Output stopped	Output stopped	Operates
Charge pump	Error out pins floated	Error out pins floated	Operates, but normally outputs low level because no signal is input

14.7 Using PLL Frequency Synthesizer

To control the PLL frequency synthesizer, the following data is necessary:

(1) Division mode : direct division (MF), pulse swallow (HF, VHF)
(2) Pins used : VCOL or VCOH pin
(3) Reference frequency : f_r
(4) Division ratio : N

The following **14.7.1** through **14.7.3** describe how to set the PLL data in the respective division modes (MF, HF, and VHF).

14.7.1 Direct division mode

(1) Selecting division mode

Select the direct division mode by the PLL mode select register.

(2) Pins used

When the direct division mode is selected, the VCOL pin is enabled to operate.

(3) Setting reference frequency f_r

Set a reference frequency by using the PLL reference clock select register.

(4) Calculating division ratio N

Calculate as follows:

$$N = \frac{f_{VCOL}}{f_r}$$

where,
f_{VCOL} : input frequency of VCOL pin
f_r : reference frequency

(5) Example of setting PLL data

Setting the data to receive the broadcasting in the following MW band is described.

Reception frequency : 1422 kHz (MW band)
Reference frequency : 9 kHz
Intermediate frequency : +450 kHz

Division ratio N is

$$N = \frac{f_{VCOL}}{f_r} = \frac{1422 + 450}{9} = 208 \text{ (decimal)} = 0D0H \text{ (hexadecimal)}$$

Set data to the PLL data register (PLLR: peripheral address 41H), PLL mode select register (PLLMODE: RF address 21H), and PLL reference clock select register (PLLRFCLK: RF address 31H) as follows:

PLLR			
0000	1101	0000	Don't care
0	D	0	

PLLMODE	PLLRFCLK
0001	1101
MF	9kHz

14.7.2 Pulse swallow mode (HF)

(1) Selecting division mode

Select the pulse swallow mode by the PLL mode select register.

(2) Pins used

When the pulse swallow mode is selected, the VCOL pin is enabled to operate.

(3) Setting reference frequency f_r

Set a reference frequency by using the PLL reference clock select register.

(4) Calculating division ratio N

Calculate as follows:

$$N = \frac{f_{VCOL}}{f_r}$$

where,

f_{VCOL} : input frequency of VCOL pin

f_r : reference frequency

(5) Example of setting PLL data

Setting the data to receive the broadcasting in the following SW band is described.

Reception frequency : 25.50 MHz (SW band)

Reference frequency : 5 kHz

Intermediate frequency : +450 kHz

Division ratio N is

$$N = \frac{f_{VCOL}}{f_r} = \frac{25500 + 450}{5} = 5190 \text{ (decimal)} = 1446\text{H (hexadecimal)}$$

Set data to the PLL data register (PLLR: peripheral address 41H), PLL mode select register (PLLMODE: RF address 21H), and PLL reference clock select register (PLLRFCLK: RF address 31H) as follows:

PLLR			
0001	0100	0100	0110
1	4	4	6

PLLMODE	PLLRFCLK
0011	0010
MF	5 kHz

14.7.3 Pulse swallow mode (VHF)

(1) Selecting division mode

Select the pulse swallow mode by the PLL mode select register.

(2) Pins used

When the pulse swallow mode is selected, the VCOH pin is enabled to operate.

(3) Setting reference frequency f_r

Set a reference frequency by using the PLL reference clock select register.

(4) Calculating division ratio N

Calculate as follows:

$$N = \frac{f_{VCOH}}{f_r}$$

where,
f_{VCOH} : input frequency of VCOH pin
f_r : reference frequency

(5) Example of setting PLL data

Setting the data to receive the broadcasting in the following FM band is described.

Reception frequency : 100.0 MHz (FM band)
Reference frequency : 25 kHz
Intermediate frequency : +10.7 MHz

Division ratio N is

$$N = \frac{f_{VCOH}}{f_r} = \frac{100.0 + 10.7}{0.025} = 4428 \text{ (decimal)} = 114CH \text{ (hexadecimal)}$$

Set data to the PLL data register (PLLR: peripheral address 41H), PLL mode select register (PLLMODE: RF address 21H), and PLL reference clock select register (PLLRFCLK: RF address 31H) as follows:

PLLR			
0001	0001	0100	1100
1	1	4	C

PLLMODE	PLLRFCLK
0010	0110
VHF	25 kHz

14.8 Status on Reset

14.8.1 On power-ON reset

The PLL is disabled because the PLL reference clock select register is initialized to 1111B.

14.8.2 On execution of clock stop instruction

The PLL is disabled when the CE pin goes low.

14.8.3 On CE reset

(1) CE reset after execution of clock stop instruction

The PLL is disabled because the PLL reference clock select register is initialized to 1111B by the clock stop instruction.

(2) CE reset without clock stop instruction executed

The PLL reference clock select register restores the previous status when the CE pin goes high because the register holds the previous status.

14.8.4 In halt status

The set status is retained as long as the CE pin is high.

15. GENERAL-PURPOSE PORTS

The general-purpose ports output high-level, low-level, and floating signals to external circuits, and read high-level and low-level signals from the external circuits.

15.1 Configuration and Classification of General-Purpose Ports

Figure 15-1 shows the block diagram of the general-purpose ports.

Table 15-1 classifies the general-purpose ports.

As shown in Figure 15-1, the general-purpose ports include port 0A (P0A) through port 2A (P2A) that set data from addresses 70H through 73H (port registers) of each bank of the data memory, ports 0E (P0E), 0F (P0F), and 0X (P0X) that set data from addresses 68H, 69H, 6BH, and 6DH of bank 0 of the data memory, and ports 0Y (P0Y) and 0X (P0X) that set data via data buffer (DBF) (data can be set to P0X via port register and peripheral register).

Each port consists of general-purpose port pins (for example, $P0A_3$ to $P0A_0$ pins for port 0A).

The general-purpose ports are classified into I/O ports, input ports, and output ports, as shown in Table 15-1.

The I/O ports are further subdivided into bit I/O ports which can be specified in the input or output mode in 1-bit (1-pin) units, and a group I/O ports which can be specified in the input or output mode in 4-bit (4-pin) units.

Figure 15-1. Block Diagram of General-Purpose Ports

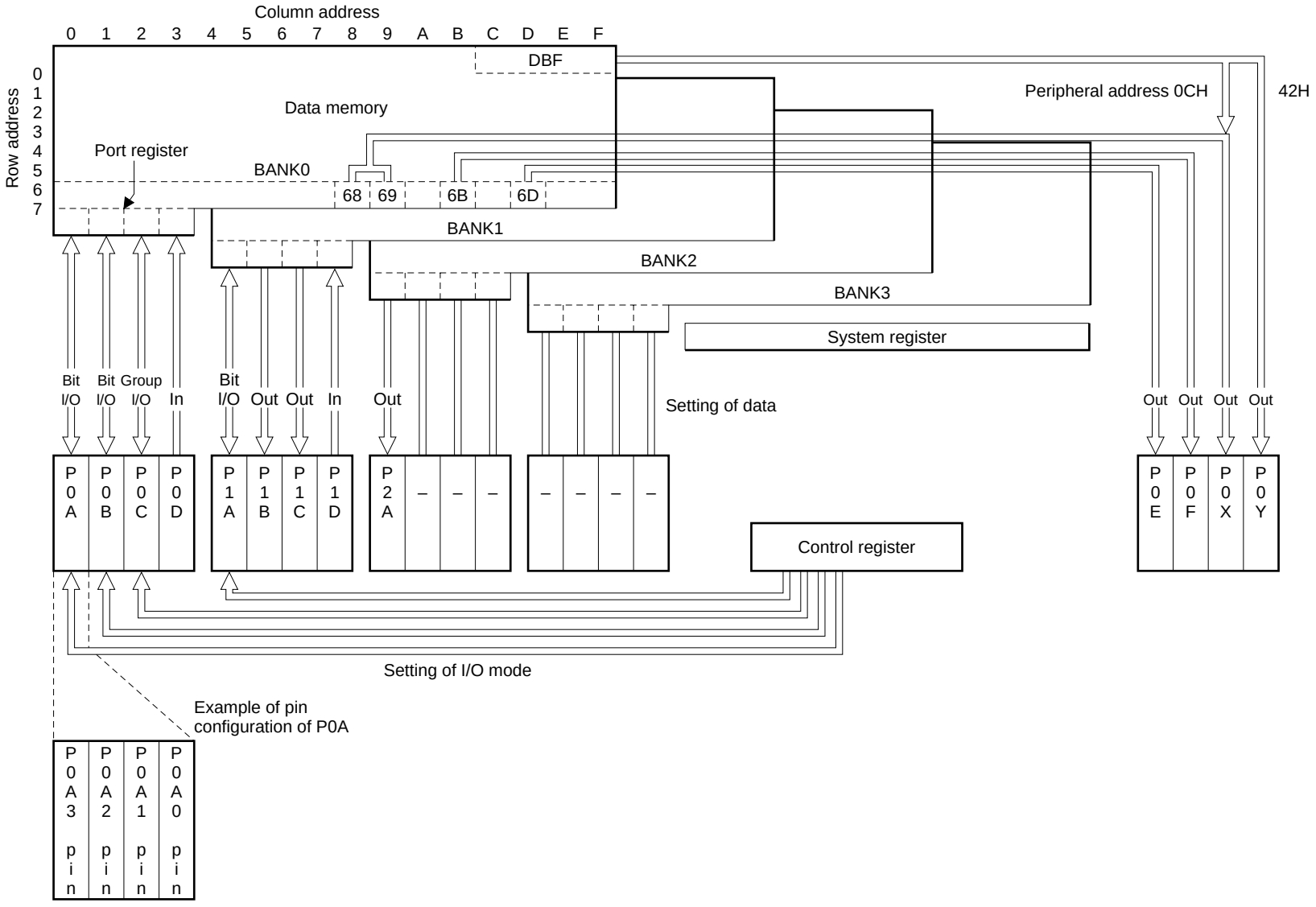


Table 15-1. Classification of General-Purpose Ports

Classification			Port	Data Set by:
General-purpose ports	I/O ports	Bit I/O	Port 0A Port 0B Port 1A	Port register
		Group I/O	Port 0C	Port register
	Input port		Port 0D Port 1D	Port register
	Output port		Port 1B Port 1C Port 2A	Port register
			Port 0E Port 0F	Port register (shared with LCD segment register)
			Port 0X	
			Port 0Y	Peripheral register

15.2 Functional Outline of General-Purpose Ports

A general-purpose output port or a general-purpose I/O port set in the output mode outputs a high or low level from the corresponding pin when data is set to the port register or port group register.

A general-purpose input port or a general-purpose I/O port set in the input mode detects the input signal level applied to the corresponding pin by reading the contents of the port register.

A general-purpose I/O port can be set in the input or output mode by using the corresponding control register.

In other words, the input or output mode can be changed by program.

P0A through P0D, P1A through P1D, and P2A are set as general-purpose ports at power-ON reset. The mode of the pins multiplexed with the other hardware is independently set by the corresponding control register.

P0E, P0F, P0X, and P0Y are set as LCD segment signal output pins at power-ON reset. These ports can be independently specified as general-purpose output ports by using the corresponding control register.

The following **15.2.1** through **15.2.5** describe the functions of the port register, port group register, and the functional outline of the respective ports.

15.2.1 General-purpose port data registers (port registers)

A port register sets the output data of the corresponding general-purpose port or reads the input data of the port.

Since the port register is located on the data memory, it can be operated by all the data memory manipulation instructions.

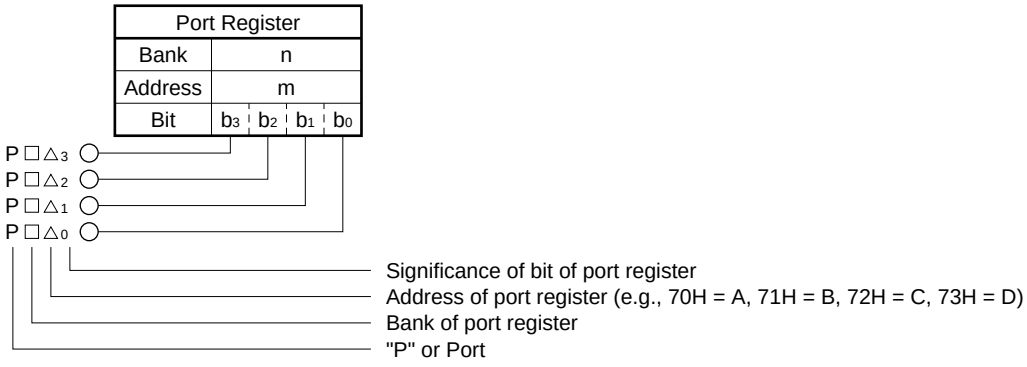
Figure 15-2 shows the relation between each port register and the corresponding pin.

The output data of each port pin is set by setting the data to the port register corresponding to the pin set as a general-purpose output port.

The input status of each port pin is detected by reading the port register corresponding to the port set as a general-purpose input port.

Table 15-2 shows the relation between each port (pin) and port register.

Figure 15-2. Relation between Port Register and Pin



Reserved words are defined by the assembler (AS17K) for port registers.

Because these reserved words are defined in flag (bit) units, assembler embedded macro instructions can be used.

Note that no reserved word of data memory type is defined for the port register.

P0E, P0F, P0X, and P0Y are shared by LCD segment signal output pins. The port registers of P0E, P0F, and P0X are also shared by LCD segment registers.

Because the LCD segment registers are also located on the data memory, they can be used in the same manner as the port registers.

15.2.2 Port 0X (P0X) and port 0Y (P0Y) group registers

The port 0X (P0X) group register sets the output data of P0X. This register is shared by an LCD group register, and is allocated to peripheral address 0CH.

The port 0Y (P0Y) group register sets the output data of P0Y. This register is shared by a key source data register, and is allocated to peripheral address 42H.

For details, refer to **15.6.7**.

15.2.3. General-purpose I/O ports (P0A, P0B, P0C, and P1A)

P0A, P0B, P0C, and P1A are set in the input or output mode by the P0A bit I/O select register (P0ABIO: RF address 37H), P0B bit I/O select register (P0BBIO: RF address 36H), P0C group I/O select register (P0CBIO: RF address 27H), and P1A bit I/O select register (P1ABIO: RF address 35H), respectively.

The input/output data of the P0A, P0B, P0C, and P1A are set by port registers P0A (address 70H of BANK0), P0B (address 71H of BANK0), P0C (address 72H of BANK0), and P1A (address 70H of BANK1), respectively.

Refer to Table 15-2.

For details, refer to **15.3**.

15.2.4 General-purpose input ports (P0D and P1D)

The input data of P0D and P1D are read by using port registers P0D (address 73H of BANK0) and P1D (address 73H of BANK1), respectively.

Refer to Table 15-2.

For details, refer to **15.4**.

15.2.5 General-purpose output ports (P1B, P1C, P2A, P0E, P0F, P0X, and P0Y)

(1) P1B, P1C, and P2A

The output data of P1B, P1C, and P2A are set by using port registers P1B (address 71H of BANK1), P1C (address 72H of BANK1), and P2A (address 70H of BANK2), respectively.

Refer to Table 15-2.

For details, refer to **15.5**.

(2) P0E, P0F, P0X, and P0Y

P0E, P0F, P0X, and P0Y usually operates as LCD segment signal output pins. These ports are used as output ports if so specified by the LCD port select register (LCDPORT: RF address 11H).

The output data of P0E and P0F are set by the P0E register (shared by LCD segment register LCDD13, address 6DH of BANK0) and P0F register (shared with LCDD11, address 6BH of BANK0).

The output data of P0X is set by the P0XL register (shared by LCDD8, address 68H of BANK0) and P0XH (shared by LCDD11, address 69H of BANK0), or by the port 0X (P0X) group register via data buffer.

The output data of P0Y is set by the port 0Y (P0Y) group register via data buffer.

Refer to Table 15-2.

For details, refer to **15.6**.

Table 15-2. Relation between Port Pins and Port Registers (1/2)

Port	Pin			Data Setting Method					
	Number	Symbol	I/O	Port register (Data Memory)					Remark
				Bank	Address	Symbol	Bit Symbol (Reserved Word)		
Port 0A (P0A)	3	$P0A_3$	I/O (bit I/O)	BANK0	70H	P0A	b_3	P0A3	
	4	$P0A_2$					b_2	P0A2	
	5	$P0A_1$					b_1	P0A1	
	6	$P0A_0$					b_0	P0A0	
Port 0B (P0B)	7	$P0B_3$	I/O (bit I/O)		71H	P0B	b_3	P0B3	
	8	$P0B_2$					b_2	P0B2	
	9	$P0B_1$					b_1	P0B1	
	10	$P0B_0$					b_0	P0B0	
Port 0C (P0C)	79	$P0C_3$	I/O (group I/O)		72H	P0C	b_3	P0C3	
	80	$P0C_2$					b_2	P0C2	
	1	$P0C_1$					b_1	P0C1	
	2	$P0C_0$					b_0	P0C0	
Port 0D (P0D)	75	$P0D_3$	Input		73H	P0D	b_3	P0D3	
	76	$P0D_2$					b_2	P0D2	
	77	$P0D_1$					b_1	P0D1	
	78	$P0D_0$					b_0	P0D0	
Port 1A (P1A)	14	$P1A_3$	I/O (bit I/O)	BANK1	70H	P1A	b_3	P1A3	
	15	$P1A_2$					b_2	P1A2	
	16	$P1A_1$					b_1	P1A1	
	17	$P1A_0$					b_0	P1A0	
Port 1B (P1B)	18	$P1B_3$	Output		71H	P1B	b_3	P1B3	
	19	$P1B_2$					b_2	P1B2	
	20	$P1B_1$					b_1	P1B1	
	21	$P1B_0$					b_0	P1B0	
Port 1C (P1C)	22	$P1C_3$	Output		72H	P1C	b_3	P1C3	
	23	$P1C_2$					b_2	P1C2	
	24	$P1C_1$					b_1	P1C1	
	25	$P1C_0$					b_0	P1C0	
Port 1D (P1D)	26	$P1D_3$	Input		73H	P1D	b_3	P1D3	
	27	$P1D_2$					b_2	P1D2	
	28	$P1D_1$					b_1	P1D1	
	29	$P1D_0$					b_0	P1D0	
Port 2A (P2A)	No pins			BANK2	70H	P2A	b_3	P2A3	Nothing is allocated. Cannot be used as data memory.
							b_2	P2A2	
							b_1	P2A1	
	42	$P2A_0$	Output				b_0	P2A0	
					71H	---	b_3	---	Nothing is allocated. Cannot be used as data memory.
							b_2		
							b_1		
							b_0		
					72H	---	b_3	---	
							b_2		
							b_1		
							b_0		
					73H	---	b_3	---	
							b_2		
							b_1		
							b_0		

Table 15-2. Relation between Port Pins and Port Registers (2/2)

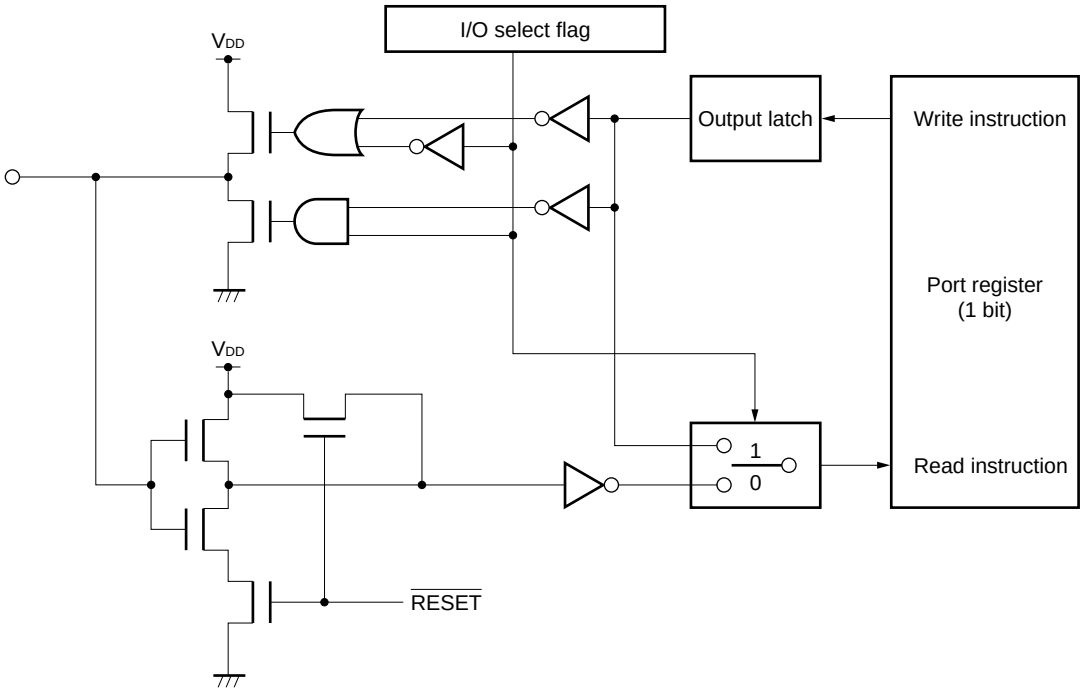
Port	Pin			Data Setting Method								
	Number	Symbol	I/O	Port register (Data Memory)					P0X, P0Y Group Registers (Peripheral Registers)		
				Bank	Address	Symbol	Bit Symbol (Reserved Word)		Peripheral Address	Symbol (Reserved Word)	Bit
				BANK3	70H	---	b_3	---	Nothing is allocated. Cannot be used as data memory.		
							b_2				
							b_1				
							b_0				
					71H	---	b_3	---			
							b_2				
							b_1				
							b_0				
					72H	---	b_3	---			
							b_2				
							b_1				
							b_0				
					73H	---	b_3	---			
							b_2				
							b_1				
							b_0				
Port 0E (P0E)	49	$P0E_3$	Output	Bank1	6BH	P0E	b_3	P0E3			
	50	$P0E_2$			(Shared by LCDD11)		b_2	P0E2			
	51	$P0E_1$					b_1	P0E1			
	52	$P0E_0$					b_0	P0E0			
Port 0F (P0F)	45	$P0F_3$	Output		6DH	P0F	b_3	P0F3			
	46	$P0F_2$			(Shared by LCDD13)		b_2	P0F2			
	47	$P0F_1$					b_1	P0F1			
	48	$P0F_0$					b_0	P0F0			
Port 0X (P0X)	53	$P0X_5$	Output		69H	P0XH	b_3	P0XH3	0CH	P0X	b_7
	54	$P0X_4$			(Shared by LCDD9)		b_2	P0XH2	(Shared by LCDR4)		b_6
	55	$P0X_3$					b_1	P0XH1			b_5
	56	$P0X_2$					b_0	P0XH0			b_4
	57	$P0X_1$			68H	P0XL	b_3	P0XL3			b_3
	58	$P0X_0$			(Shared by LCDD8)		b_2	P0XL2			b_2
	No pins						b_1	P0XL1			Don't care
							b_0	P0XL0			
Port 0Y (P0Y)	59	$P0Y_{15}$	Output						42H	P0Y	b_{15}
	60	$P0Y_{14}$							(Shared by KSR)		b_{14}
	61	$P0Y_{13}$									b_{13}
	|	|								|
	72	$P0Y_2$									b_2
	73	$P0Y_1$									b_1
	74	$P0Y_0$									b_0

15.3 General-Purpose I/O Ports (P0A, P0B, P0C, and P1A)

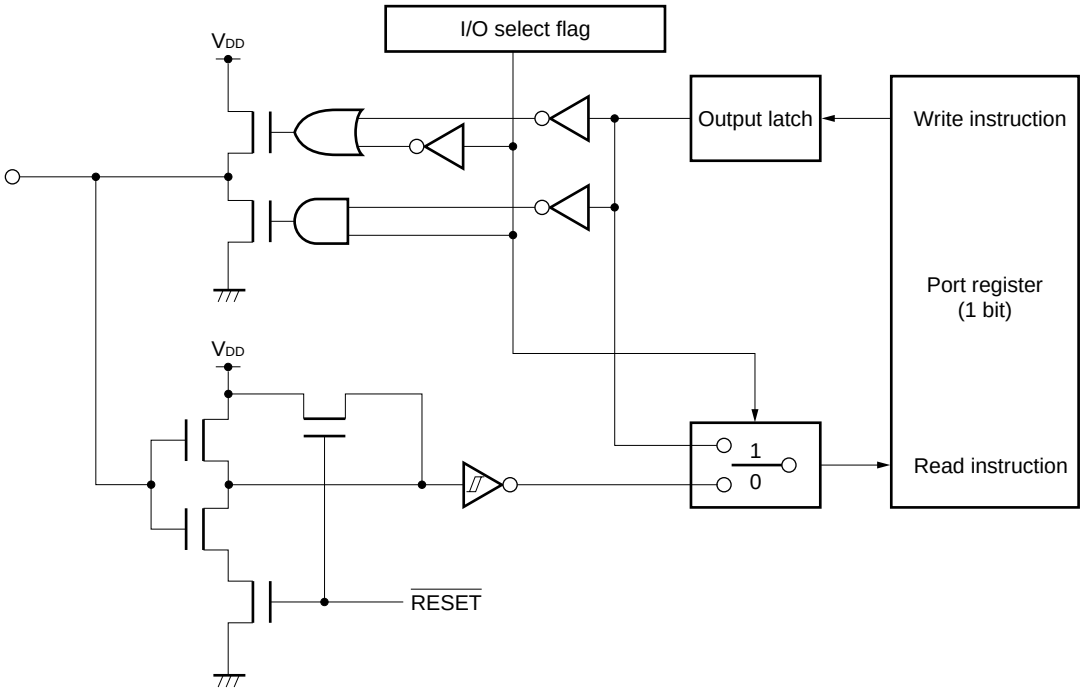
15.3.1 Configuration of I/O ports

(1) through (4) below show the configuration of the I/O ports.

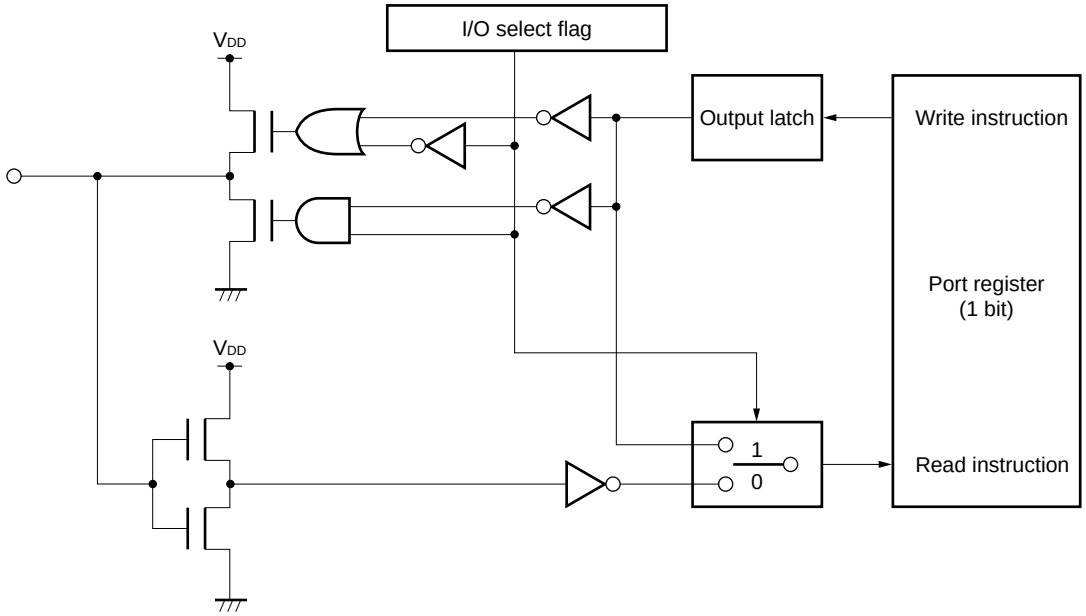
(1) P0A (P0A$_0$ pin),
P0B (P0B$_1$ pin),
P1A (P1A$_3$, P1A$_2$, P1A$_1$, and P1A$_0$ pins)



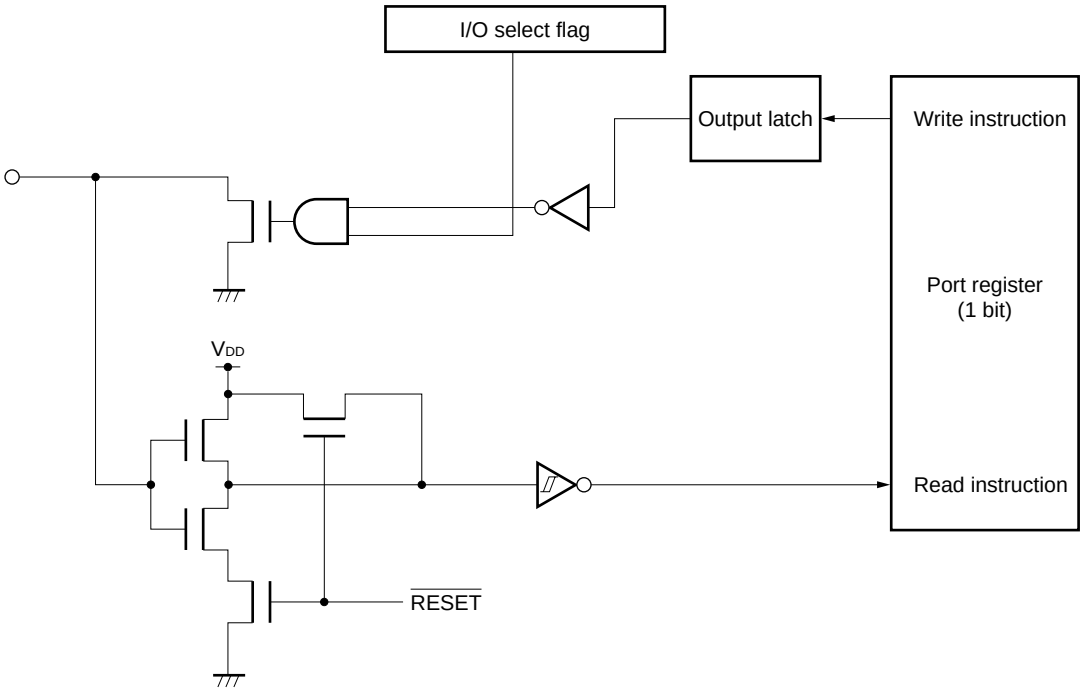
(2) P0A (P0A$_1$ pin)
P0B (P0B$_3$, P0B$_2$, and P0B$_0$ pins)



(3) P0C (P0C$_3$, P0C$_2$, P0C$_1$, and P0C$_0$ pins)



(4) P0A (P0A$_3$, and P0A$_2$ pins)



15.3.2 Using I/O ports

The I/O ports are set in the input or output mode by the control register P0A, P0B, P0C, and the I/O select registers of P1A, respectively.

The bit I/O ports (P0A, P0B, and P1A) can be set in the input or output mode in 1-bit units, and the group I/O port (P0C) can be set in the input or output mode in 4-bit units.

To set the output data or to read the input data, execute an instruction that writes data to the corresponding port register or that reads data from the corresponding port register.

The following **15.3.3** describes the I/O select register of each port.

15.3.4 and **15.3.5** describe the input and output modes of the I/O ports.

15.3.6 describes the points to be noted in using the I/O ports.

15.3.3 Port 0A bit I/O select register (P0ABIO)
Port 0B bit I/O select register (P0BBIO)
Port 1A bit I/O select register (P1ABIO)
Port 0C group I/O select register (P0CGPIO)

Port 0A bit I/O, port 0B bit I/O, port 1A bit I/O, and port 0C group I/O select registers sets the input or output mode of each pin of P0A, P0B, P1A, and P0C.

(1) through (4) shows the configuration and functions of these registers.

(1) Port 0A bit I/O select register (P0ABIO)

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Port 0A bit I/O select register (P0ABIO)	P0ABIO3	P0ABIO2	P0ABIO1	P0ABIO0	37H	R/W

P0ABIO0	Sets input/output mode
0	Sets $P0A_0$ pin in input mode
1	Sets $P0A_0$ pin in output mode

P0ABIO1	Sets input/output mode
0	Sets $P0A_1$ pin in input mode
1	Sets $P0A_1$ pin in output mode

P0ABIO2	Sets input/output mode
0	Sets $P0A_2$ pin in input mode
1	Sets $P0A_2$ pin in output mode

P0ABIO3	Sets input/output mode
0	Sets $P0A_3$ pin in input mode
1	Sets $P0A_3$ pin in output mode

On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

(2) Port 0B bit I/O select register (P0BBIO)

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Port 0B bit I/O select register (P0BBIO)	P0BBIO3	P0BBIO2	P0BBIO1	P0BBIO0	36H	R/W

P0BBIO0

	Sets input/output mode
0	Sets $P0B_0$ pin in input mode
1	Sets $P0B_0$ pin in output mode

P0BBIO1

	Sets input/output mode
0	Sets $P0B_1$ pin in input mode
1	Sets $P0B_1$ pin in output mode

P0BBIO2

	Sets input/output mode
0	Sets $P0B_2$ pin in input mode
1	Sets $P0B_2$ pin in output mode

P0BBIO3

	Sets input/output mode
0	Sets $P0B_3$ pin in input mode
1	Sets $P0B_3$ pin in output mode

		P0BBIO3	P0BBIO2	P0BBIO1	P0BBIO0
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

(3) Port 1A bit I/O select register (P1ABIO)

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Port 1A bit I/O select register (P1ABIO)	P1ABIO3	P1ABIO2	P1ABIO1	P1ABIO0	35H	R/W

P1ABIO0

	Sets input/output mode
0	Sets $P1A_0$ pin in input mode
1	Sets $P1A_0$ pin in output mode

P1ABIO1

	Sets input/output mode
0	Sets $P1A_1$ pin in input mode
1	Sets $P1A_1$ pin in output mode

P1ABIO2

	Sets input/output mode
0	Sets $P1A_2$ pin in input mode
1	Sets $P1A_2$ pin in output mode

P1ABIO3

	Sets input/output mode
0	Sets $P1A_3$ pin in input mode
1	Sets $P1A_3$ pin in output mode

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

(4) Port 0C group I/O select register (P0CGPIO)

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Port 0C group I/O select register (P0CGPIO)	0	0	0	P0CGIO	27H	R/W

P0CGIO

- Sets input/output mode
  - 0: Sets $P0C_3$-$P0C_0$ pins in input mode
  - 1: Sets $P0C_3$-$P0C_0$ pins in output mode
- Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop				0
	CE				0

15.3.4 To use I/O ports (P0A, P0B, P0C, or P1A) as input port

The pin to be set in the input mode is selected by the I/O select register of each port.

Note that P0C can be set in the input or output mode in 4-bit units only.

The pin set in the input mode is floated (Hi-Z) and waits for input of an external signal.

To read the input data of a pin, execute an instruction that reads the contents of the port register corresponding to the pin, such as “SKT” instruction.

When a high level is input to each pin, “1” is read to the port register; when a low level is input, “0” is read to the register.

If an instruction that writes the port register of the port set in the input mode, such as “MOV” instruction, is executed, the contents of the output latch are rewritten.

15.3.5 To use I/O ports (P0A, P0B, P0C, or P1A) as output port

The pin to be set in the output mode is selected by the I/O select register of each port.

Note that P0C can be set in the input or output mode in 4-bit units only.

The pin set in the output mode outputs the contents of the output latch from each pin.

To set the output data, execute an instruction that writes data to the port register corresponding to the pin, such as “MOV” instruction.

To output a high level to each pin, write “1” to the corresponding port register. To output a low level, write “0” to the register.

A port pin can be floated by setting it in the input mode.

When an instruction that reads the port register set in the output mode, such as “SKT” instruction, is executed, the contents of the output latch are read.

Note, however, that the status of the $P0A_3$ and $P0A_2$ pins are read as is, the contents of the output latch and read data may differ. For further information, refer to **15.3.6**.

15.3.6 Notes on using I/O ports ($P0A_3$ and $P0A_2$ pins)

When using the $P0A_3$ and $P0A_2$ pins as output pins as shown in the example below, the contents of the output latch may be rewritten.

Example To specify $P0A_3$ and $P0A_2$ pins as output port pins

```
INITFLG P0ABIO3, P0ABIO2, NOT P0ABIO1, NOT P0ABIO0
                ; Sets P0A3 and P0A2 pins in output mode
INITFLG P0A3, P0A2, NOT P0A1, NOT P0A0
                ; Outputs high level to P0A3 and P0A2 pins
; <1>
CLR1   P0A3  ; Outputs low level to P0A3 pin
Macro expansion
AND    .  MF. P0A3 SHR 4, #. DF. (NOT P0A3 AND 0FH)
```

If the $P0A_2$ pin happens to be made low externally when the instruction <1> in the above example is executed, the contents of the output latch of the $P0A_2$ pin are written to “0” by the “CLR1” instruction.

In other words, if an operation instruction (such as “ADD” or “OR”) is executed to the P0A port register when the $P0A_3$ or $P0A_2$ pin is set in the output mode, the contents of the output latch are written to the pin level at that time, regardless of the previous status.

15.3.7 Reset status of I/O ports (P0A, P0B, P0C, and P1A)

(1) On power-ON reset

All the I/O ports are set in the input mode.

The contents of the output latch are “undefined”; therefore, the output latch must be initialized by program as necessary when setting the ports in the output mode.

(2) On CE reset

All the I/O ports are set in the input mode.

The contents of the output latch are retained.

(3) On execution of clock stop instruction

All the I/O ports are set in the input mode.

The contents of the output latch are retained.

The I/O ports, except P0C, prevent an increase in the current dissipation due to noise superimposed on the input buffer, by using the $\overline{\text{RESET}}$ signal output, as described in **15.3.1**, when the clock stop instruction is executed.

P0C must be externally pulled down or up as necessary because, if it is floated when the clock stop instruction is executed, its current dissipation may increase due to external noise.

(4) In halt status

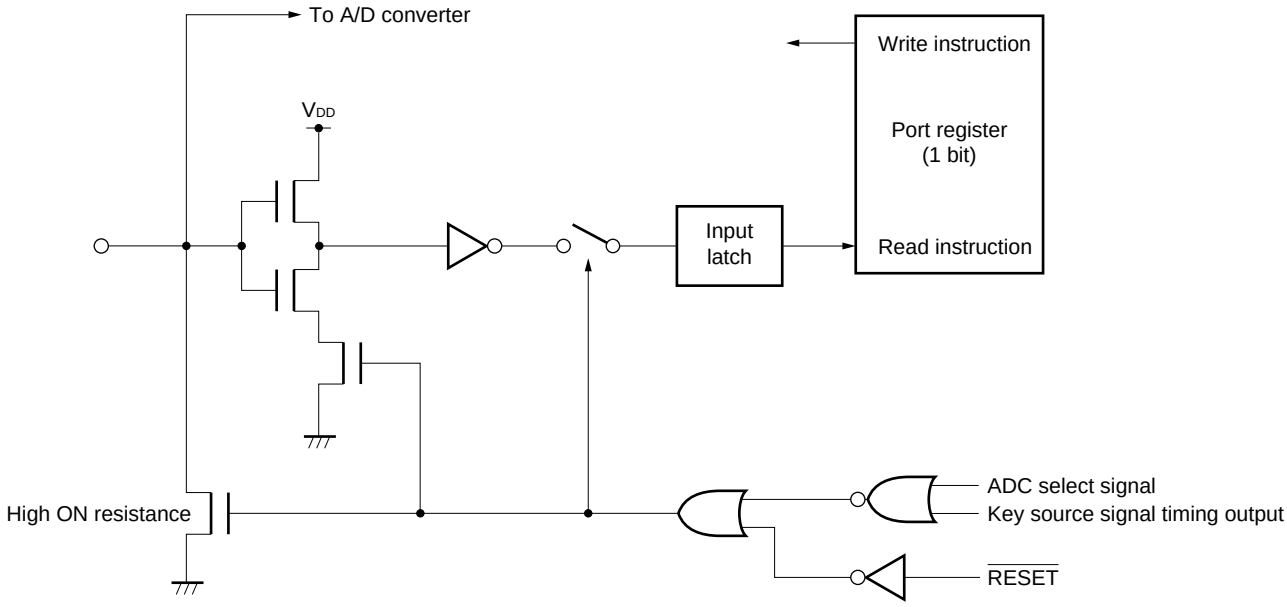
The I/O ports retain the previous status.

15.4 General-Purpose Input Ports (P0D and P1D)

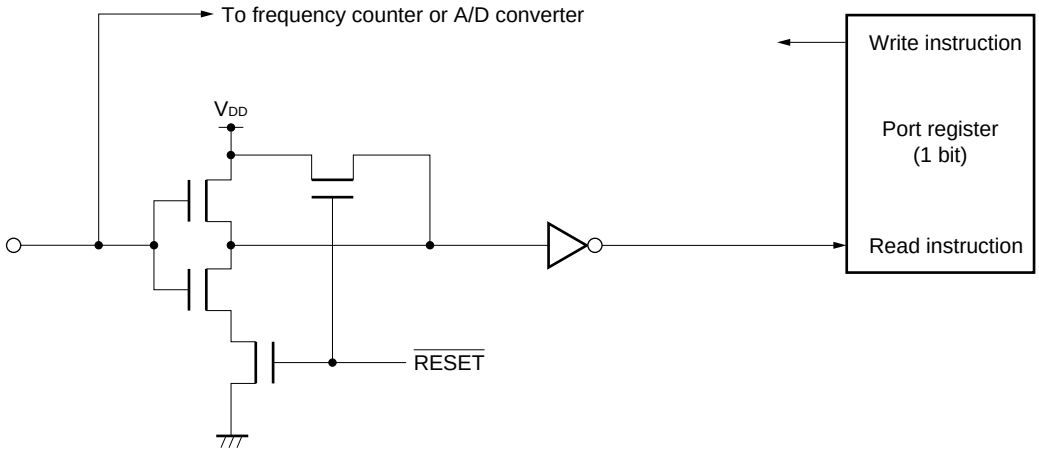
15.4.1 Configuration of input ports

(1) and (2) below show the configuration of the input ports.

(1) P0D (P0D$_3$, P0D$_2$, P0D$_1$, and P0D$_0$ pins)



(2) P1D (P1D$_3$, P1D$_2$, P1D$_1$, and P1D$_0$ pins)



15.4.2 Example of using input ports (P0D and P1D)

The input data is read by executing an instruction that reads the contents of the port register corresponding to each pin, such as “SKT” instruction.

When a high level is input to the pin, “1” is read to the port register; when a low level is input, “0” is read to the register.

The contents of the port register are not changed by executing a write instruction, such as “MOV”.

15.4.3 Notes on using input port (P0D)

P0D is internally pulled down when used as a general-purpose port.

15.4.4 Reset status of input ports (P0D and P1D)

(1) On power-ON reset

All the input ports are specified as general-purpose input ports.

(2) On CE reset

All the input ports are specified as general-purpose input ports.

(3) On execution of clock stop instruction

All the input ports are specified as general-purpose input ports.

P1D prevents an increase in the current dissipation due to noise superimposed on the input buffer as described in **15.4.1** because the $\overline{\text{RESET}}$ signal is output when the clock stop instruction is executed.

P0D is internally pulled down.

(4) In halt status

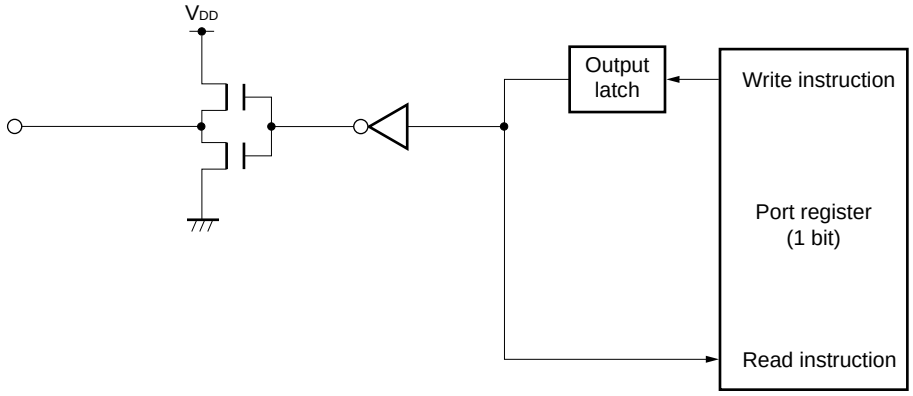
The input ports retain the previous status.

15.5 General-Purpose Output Ports (P1B, P1C, and P2A)

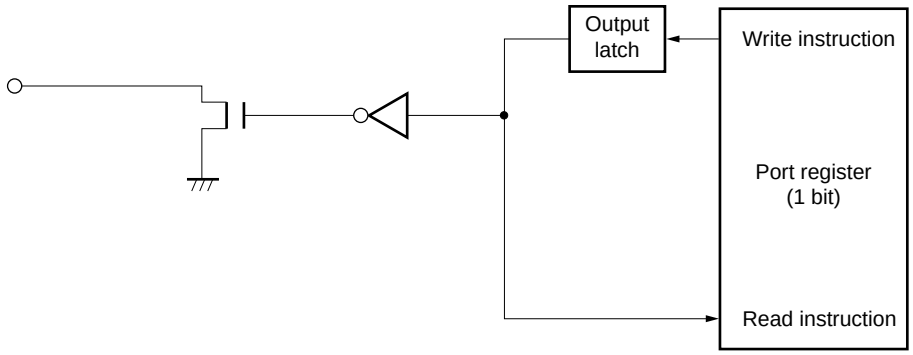
15.5.1 Configuration of output ports (P1B, P1C, and P2A)

(1) and (2) below show the configuration of the output ports.

(1) P1B (P1B$_0$ pin)
P1C (P1C$_3$, P1C$_2$, P1C$_1$, and P1C$_0$ pins)
P2A (P2A$_0$ pin)



(2) P1B (P1B$_3$, P1B$_2$, and P1B$_1$ pins)



15.5.2 Example of using output ports (P1B, P1C, and P2A)

The output ports output the contents of the output latch from each pin.

The output data is set by executing an instruction that writes data to the port register corresponding to each pin, such as “MOV” instruction.

Write “1” to the port register to output a high level to the port pin; write “0” to the register to output a low level.

Note, that the $P1B_3$, $P1B_2$, and $P1B_1$ pins float when they output a high level, because they are open-drain output ports.

When an instruction that reads the port register, such as “SKT” instruction, is executed, the contents of the output latch are read.

15.5.3 Reset status of output ports (P1B, P1C, and P2A)

(1) On power-ON reset

The contents of the output latch are output.

Because the contents of the output latch are “undefined”, an “undefined” value is output for a fixed period (until the output latch is initialized by program).

(2) On CE reset

The contents of the output latch are output.

Because the contents of the output latch are retained, the output data is not affected by CE reset.

(3) On execution of clock stop instruction

The contents of the output latch are output.

Because the contents of the output latch are retained, the output data is not affected by execution of the clock stop instruction.

Therefore, initialize the output latch by program as necessary.

(4) In halt status

The contents of the output latch are output.

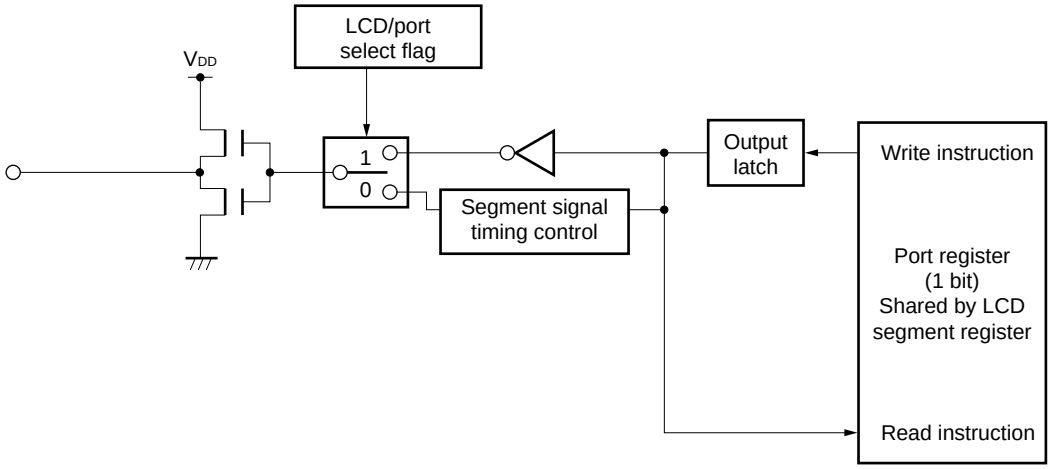
Because the contents of the output latch are retained, the output data is not affected in the halt status.

15.6 General-Purpose Output Ports (P0E, P0F, P0X, and P0Y)

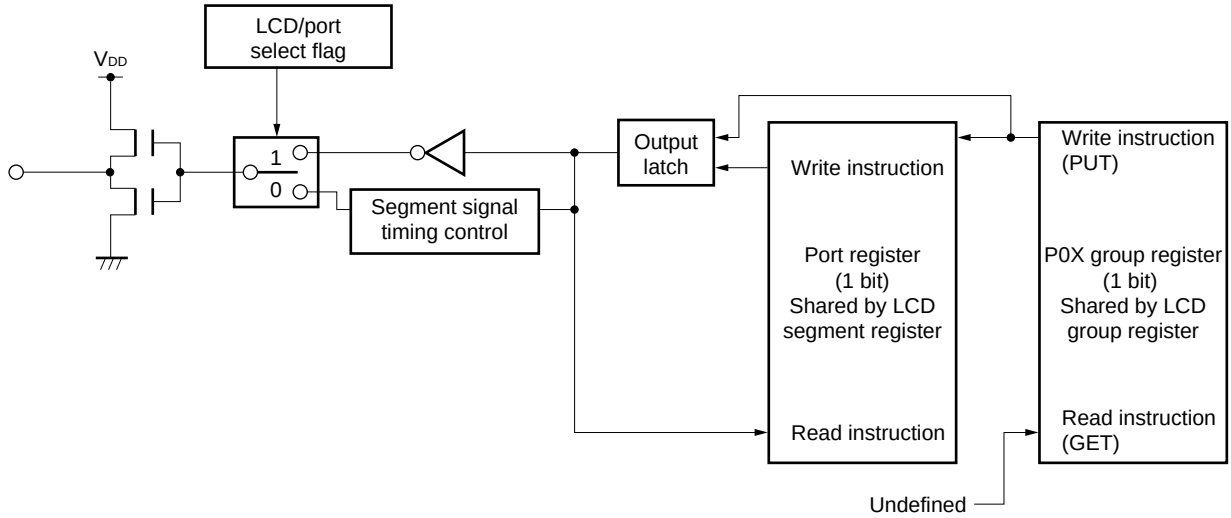
15.6.1 Configuration of output ports (P0E, P0F, P0X, and P0Y)

(1) through (3) show the configuration of the output ports (P0E, P0F, P0X, and P0Y).

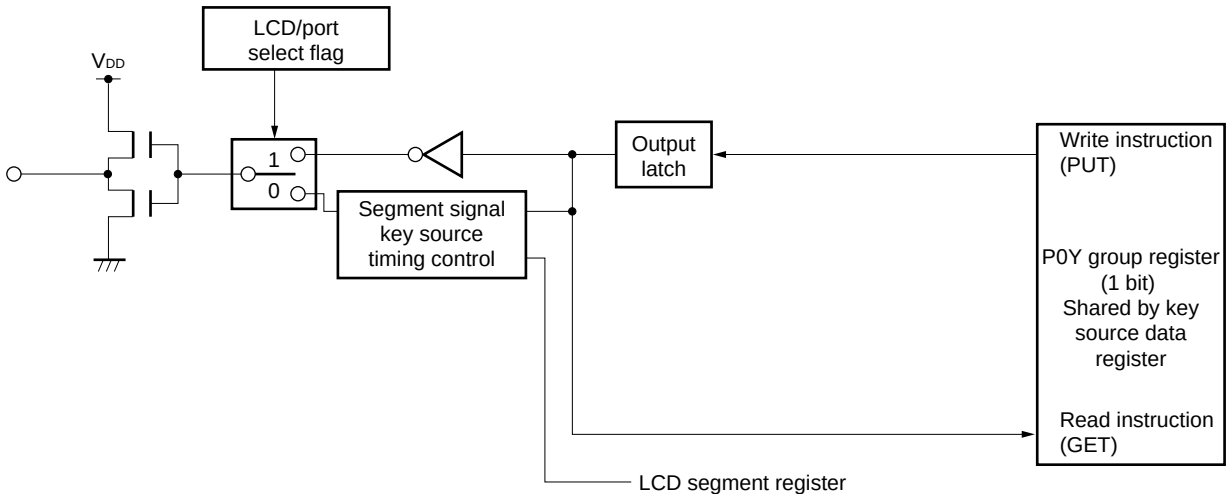
(1) P0E ($P0E_3$, $P0E_2$, $P0E_1$, and $P0E_0$ pins)
P0F ($P0F_3$, $P0F_2$, $P0F_1$, and $P0F_0$ pins)



(2) P0X ($P0X_5$ through $P0X_0$ pins)



(3) P0Y ($P0Y_{15}$ through $P0Y_0$ pins)



15.6.2 Example of using output ports (P0E, P0F, P0X, and P0Y)

The pins of P0E, P0F, P0X, and P0Y are set as LCD segment signal output pins on power-ON reset.

To use these pins as output port pins, therefore, the port pins to be used must be selected by the P0ESEN, P0FSEN, P0XSEN, or P0YSEN flag of the LCD port select register (LCDPORT: RF address 11H).

The port to be used can be selected regardless of P0E, P0F, P0X and P0Y.

The pins not set by the LCD port select register as output port pins can be used as LCD segment signal output points.

The following **15.6.3** through **15.6.5** describe how to set the output data of P0E, P0F, P0X, and P0Y.

15.6.6 and **15.6.7** describe the configuration and function of the LCD port select register and P0X and P0Y group registers.

15.6.3 Setting data to P0E and P0F

To set output data to P0E and P0F, an instruction that writes data to the port register corresponding to the port pin, such as “MOV” instruction, is executed.

To output a high level to each pin, “1” is written to the port register. To output a low level, “0” is written to the register.

When an instruction that reads the contents of the port register, such as “SKT”, is executed, the contents of the output latch are read.

Figure 15-3 shows the relation among the P0F port register, LCD segment register, and LCD group register.

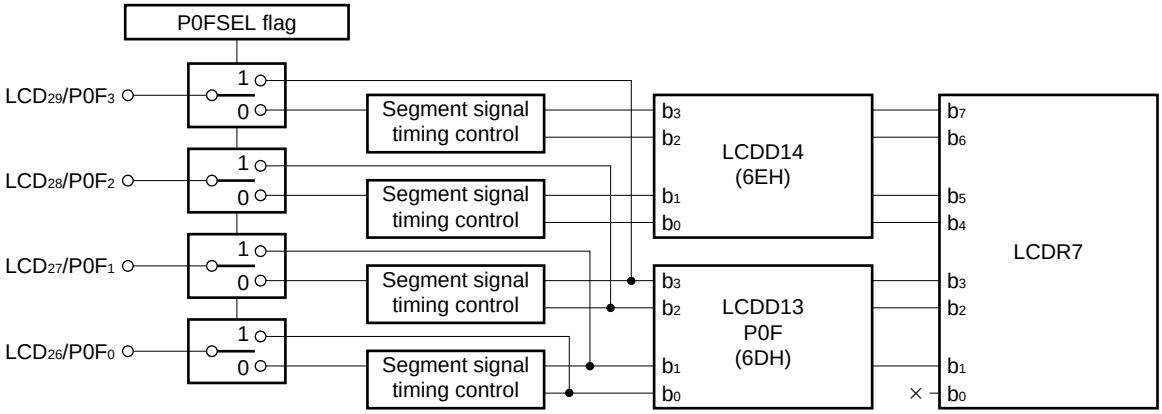
As shown in this figure, the LCD segment register LCDD14 can be used as a general-purpose data memory when P0F is used.

If data is set to the LCD group register LCDR7, the higher 3 bits of P0F are changed.

The same applies to P0F.

For details, refer to **Figure 21-7. Relation among LCD Display Dot, Ports 0E Through 0Y, Key Source Output, and Data Setting Registers** in **21. LCD CONTROLLER/DRIVER**.

Figure 15-3. Relation among P0F Port Register, LCD Segment Register, and LCD Group Register



15.6.4 Setting data to P0X

To set output data to P0X, the port register or port 0X (P0X) group register may be used.

To use the port register, execute an instruction that writes data to the port registers (P0XH and P0XL) corresponding to the port pins, such as “MOV” instruction.

To output a high level to each pin, “1” is written to the port register. To output a low level, “0” is written to the register.

When an instruction that reads the contents of the port register, such as “SKT”, is executed, the contents of the output latch are read.

To use the P0X group register, execute the “PUT P0X, DBF” instruction that writes data to the P0X group register (P0X) corresponding to the port pin.

When the “GET DBF, P0X” that reads the contents of the P0X group register (P0X) is executed, an “undefined value” is read out.

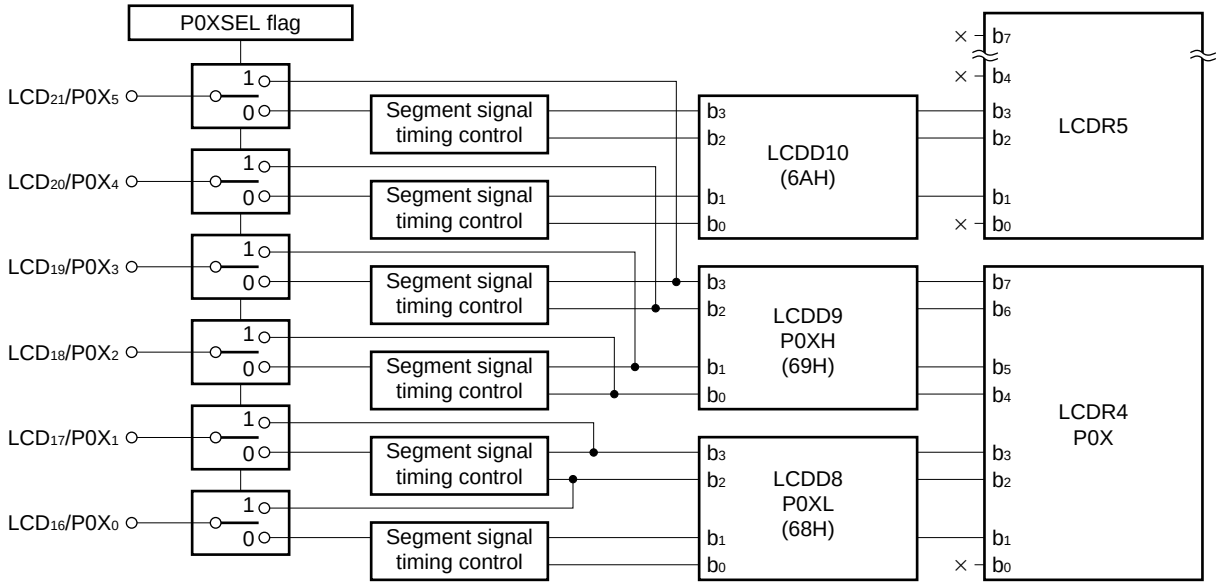
To set data by using the P0X group register, write “1” to output a high level to the port pin, and write “0” to output a low level.

Figure 15-4 shows the relation among the P0X port register, P0X group register, LCD segment register, and LCD group register.

As shown in this figure, the LCD segment register LCDD10 can be used as a general-purpose data memory when P0X is used.

For details, refer to **Figure 21-7. Relation among LCD Display Dot, Ports 0E Through 0Y, Key Source Output, and Data Setting Registers** in **21. LCD CONTROLLER/DRIVER**.

Figure 15-4. Relation among P0X Port Register, P0X Group Register, LCD Segment Register, and LCD Group Register



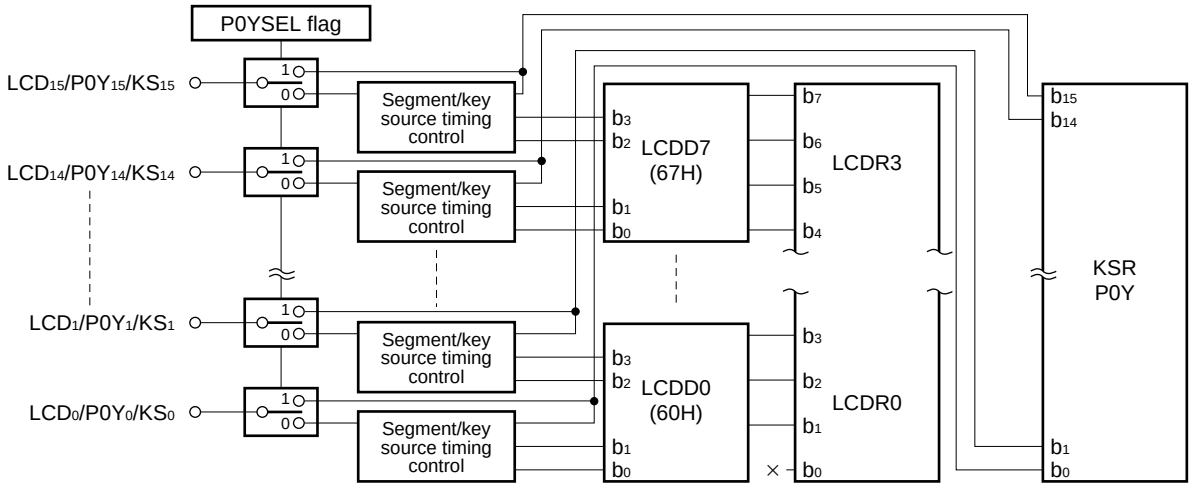
15.6.5 Setting data to P0Y

To set output data to P0Y, execute the “PUT P0Y, DBF” instruction that writes data to the port 0Y (P0Y) group register corresponding to the port pin.

When the “GET DBF, P0Y” instruction that reads the contents of the P0Y group register is executed, the contents of the output latch are read.

To output a high level to the port pin, write “1” to the register. To output a low level, write “0”.

Figure 15-5. Relation among P0Y Port Register, P0Y Group Register, LCD Segment Register, and LCD Group Register



15.6.6 Configuration and function of LCD port select register (LCDPORT)

The LCD port select register selects whether P0E, P0F, P0X, and P0Y pins are used as LCD segment signal output pins or general-purpose output port pins.

The configuration and function of this register are shown below.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
LCD port select register (LCDPORT)	P0YSEL	P0XSEL	P0ESEL	P0FSEL	11H	R/W

P0YSEL — Selects LCD segment signal output pins or general-purpose output port pins

0	Uses $LCD_{26}/P0F_0$-$LCD_{29}/P0F_3$ pins as LCD segment signal output pins
1	Uses $LCD_{26}/P0F_0$-$LCD_{29}/P0F_3$ pins as general-purpose output port pins

P0XSEL — Selects LCD segment signal output pins or general-purpose output port pins

0	Uses $LCD_{22}/P0E_0$-$LCD_{25}/P0E_3$ pins as LCD segment signal output pins
1	Uses $LCD_{22}/P0E_0$-$LCD_{25}/P0E_3$ pins as general-purpose output port pins

P0ESEL — Selects LCD segment signal output pins or general-purpose output port pins

0	Uses $LCD_{16}/P0X_0$-$LCD_{21}/P0X_5$ pins as LCD segment signal output pins
1	Uses $LCD_{16}/P0X_0$-$LCD_{21}/P0X_5$ pins as general-purpose output port pins

P0FSEL — Selects LCD segment signal output pins or general-purpose output port pins

0	Uses $LCD_0/P0Y_0/KS_0$-$LCD_{15}/P0Y_{15}/KS_{15}$ pins as LCD segment signal output port pins
1	Uses $LCD_0/P0Y_0/KS_0$-$LCD_{15}/P0Y_{15}/KS_{15}$ pins as general-purpose output port pins

		P0YSEL	P0XSEL	P0ESEL	P0FSEL
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	Retained			

Ports 0F, 0E, 0X and 0Y can be independently set as general-purpose output ports.

The pins not set as general-purpose output port pins can be used as LCD segment signal output pins.

The 16 pins $LCD_0/P0Y_0/KS_0$ through $LCD_{15}/P0Y_{15}/KS_{15}$ pins multiplex LCD segment signal output and key source signal output. When these pins are set as general-purpose output port pins, the LCD segment signals and key source signals are not output.

15.6.7 Port 0X (P0X) group register and port 0Y (P0Y) group register

(1) and (2) below show the functions of the P0X and P0Y group registers.

The P0X and P0Y group registers set the output data of P0X ($P0X_0$ through $P0X_5$ pins) and P0Y ($P0Y_0$ through $P0Y_{15}$ pins).

P0X and P0Y can respectively set 6-bit and 16-bit output data at one time.

(1) Function of P0X group register

Name	Data Buffer			
Symbol	DBF3	DBF2	DBF1	DBF0
Address	0CH	0DH	0EH	0FH
Bit	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0
Data	Don't care	Don't care	Transfer data	

8

GET sets undefined data

PUT can be executed

Peripheral Register										
Name	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Symbol	Peripheral address	Peripheral hardware
P0X group register	Valid data						–	–	P0X	0CH	Port 0X

Don't care

Sets output data of port 0X

- $LCD_{16}/P0X_0$ pin
- $LCD_{17}/P0X_1$ pin
- $LCD_{18}/P0X_2$ pin
- $LCD_{19}/P0X_3$ pin
- $LCD_{20}/P0X_4$ pin
- $LCD_{21}/P0X_5$ pin

0	Outputs low level
1	Outputs high level

The output data of port 0X can be set not only by the P0X group register (peripheral address 0CH) but also port registers P0XH and P0XL (69H and 68H of BANK0).

If data is set to the P0X group register (peripheral register), the data of the P0XH and P0XL registers (port registers) corresponding to the overlapping bit data are changed to the same value.

(2) Function of P0Y group register

Name	Data Buffer			
Symbol	DBF3	DBF2	DBF1	DBF0
Address	0CH	0DH	0EH	0FH
Bit	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0
Data	Transfer data			

16

GET can be executed

PUT can be executed

Peripheral Register																		
Name	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Symbol	Peripheral address	Peripheral hardware
P0Y group register	Valid data																P0Y	42H	Port 0Y

	Sets output data of port 0Y
	$LCD_0/P0Y_0/KS_0$ pin
	$LCD_1/P0Y_1/KS_1$ pin
	$LCD_2/P0Y_2/KS_2$ pin
	$LCD_3/P0Y_3/KS_3$ pin
	$LCD_4/P0Y_4/KS_4$ pin
	$LCD_5/P0Y_5/KS_5$ pin
	$LCD_6/P0Y_6/KS_6$ pin
	$LCD_7/P0Y_7/KS_7$ pin
	$LCD_8/P0Y_8/KS_8$ pin
	$LCD_9/P0Y_9/KS_9$ pin
	$LCD_{10}/P0Y_{10}/KS_{10}$ pin
	$LCD_{11}/P0Y_{11}/KS_{11}$ pin
	$LCD_{12}/P0Y_{12}/KS_{12}$ pin
	$LCD_{13}/P0Y_{13}/KS_{13}$ pin
	$LCD_{14}/P0Y_{14}/KS_{14}$ pin
	$LCD_{15}/P0Y_{15}/KS_{15}$ pin
0	Outputs low level
1	Outputs high level

Port 0Y is shared with key source signal output pins.

Therefore, the P0Y group register (peripheral address 42H) is shared with the key source data register (peripheral address 42H) described later.

Therefore, the output data of the port 0Y is set to this register when the $LCD_0/P0Y_0/KS_0$ through $LCD_{15}/P0Y_{15}/KS_{15}$ pins are specified as output port pins, and key source signal output data is set to the register when these pins are specified as key source signal output pins.

15.6.8 Reset status of output ports (P0E, P0F, P0X, and P0Y)

(1) On power-ON reset

These output port pins are set as LCD segment signal output pins, and output a low level.

Because the contents of the output latch are undefined, undefined data is output if these pins are set as output port pins. Initialize the contents of the output latch by program as necessary.

(2) On CE reset

These pins are set as LCD segment signal output pins, and output a low level.

The contents of the output latch are retained. Therefore, previous values can be retained if the pins are set in the output port mode.

(3) On execution of clock stop instruction

These pins are set as LCD segment signal output pins, and output a low level.

The contents of the output latch are retained. Therefore, previous values can be retained if the pins are set in the output port mode.

(4) In halt status

The contents of the output latch are output.

Because the contents of the output latch are retained, the output data is not changed in the halt status.

16. A/D CONVERTER (ADC)

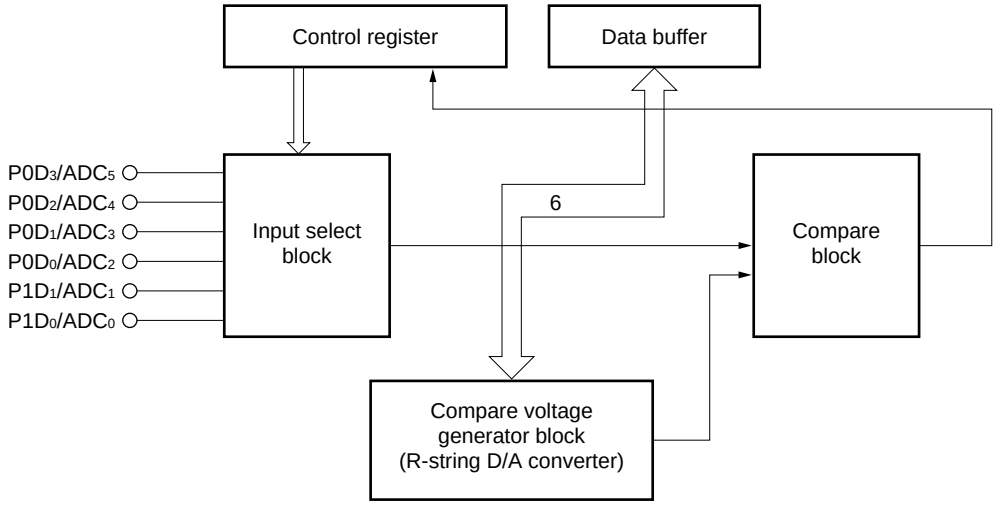
The A/D converter is used to input an external analog signal as a digital signal.

16.1 Configuration of A/D Converter

Figure 16-1 shows the block diagram of the A/D converter.

As shown in this figure, the A/D converter consists of an input select block, a compare voltage generator block, and a compare block.

Figure 16-1. Block Diagram of A/D Converter



16.2 Functional Outline of A/D Converter

The A/D converter compares the voltage input to the P0D3/ADC5 through P1D0/ADC0 pins with an internal compare voltage, and outputs the result of the comparison as “True (1)” or “False (0)”.

This comparison result is judged by software. In this way, the A/D converter is used as a successive approximation converter.

The following **16.2.1** through **16.2.3** outlines the functions of each block.

For details, refer to **16.3** through **16.5**.

16.2.1 Input select block

This block selects which of the P0D3/ADC5 through P1D0/ADC0 pins is used.

The pin to be used is selected by the A/D converter channel select register (ADCCH: RF address 14H).

Only one pin can be used at a time.

For details, refer to **16.3**.

16.2.2 Compare voltage generator block

This block generates a compare voltage against which the input voltage is to be compared.

The compare voltage is generated by an R-string D/A converter.

For details, refer to **16.4**.

16.2.3 Compare block

This block compares the input voltage with the internal compare voltage.

The result of the comparison is detected by the A/D converter compare judge register (ADCJDG: RF address 06H).

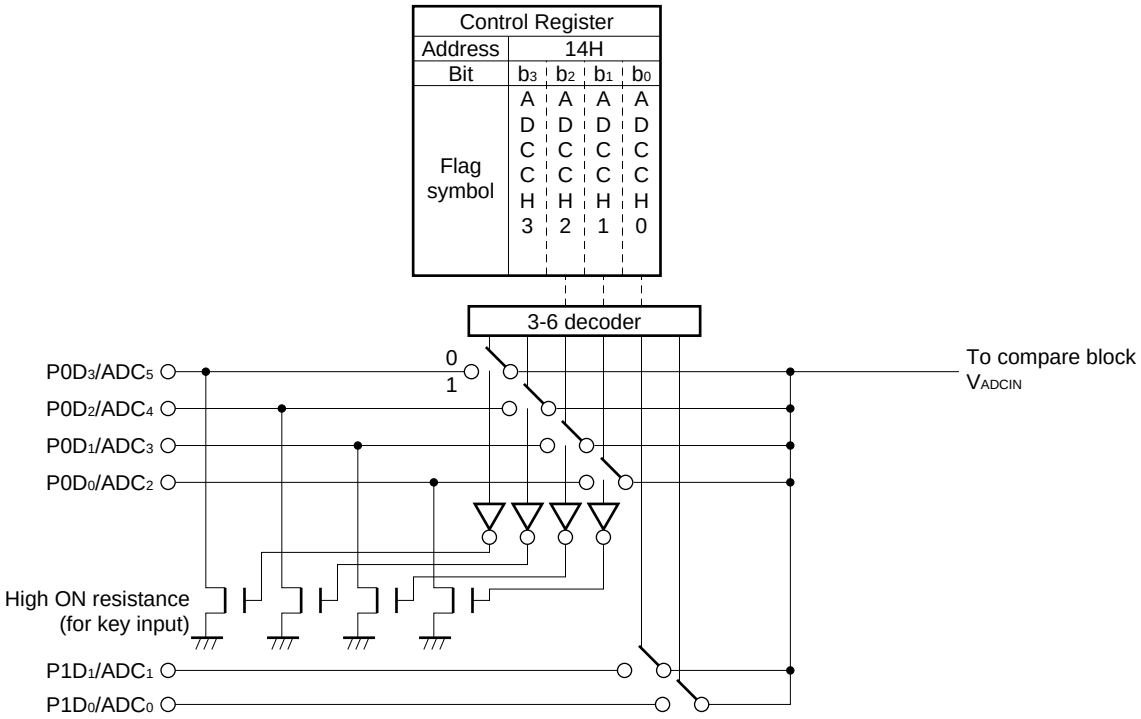
For details, refer to **16.5**.

16.3 Input Select Block

16.3.1 Configuration of input select block

Figure 16-2 shows the configuration of the input select block.

Figure 16-2. Configuration of Input Select Block



16.3.2 Function of input select block

The input select block selects the pin to be used by using the A/D converter channel select register.

Only one pin can be used at a time as an A/D converter pin.

The pins not used as A/D converter pins can be used as general-purpose input port pins.

Although port 0D (P0D$_3$/ADC$_5$ through P0D$_0$/ADC$_2$ pins) are internally connected with a pull-down resistor, the pin selected by the A/D converter channel select register (refer to **16.3.3**) is disconnected from the pull-down resistor.

The pins not selected remains connected to the pull-down resistor.

16.3.3 Configuration and function of A/D converter channel select register (ADCCH)

The A/D converter channel select register selects a pin to be used as the A/D converter pin.

The configuration and function of this register are illustrated below.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
A/D converter channel select register (ADCCH)	ADCCH3	ADCCH2	ADCCH1	ADCCH0	14H	R/W

b_3	b_2	b_1	Selects pin to be used as A/D converter pin
0	0	0	P1D$_0$/ADC$_0$ pin
0	0	1	P1D$_1$/ADC$_1$ pin
0	1	0	P0D$_0$/ADC$_2$ pin
0	1	1	P0D$_1$/ADC$_3$ pin
1	0	0	P0D$_2$/ADC$_4$ pin
1	0	1	P0D$_3$/ADC$_5$ pin
1	1	0	A/D converter is not used (general-purpose input port)
1	1	1	A/D converter is not used (general-purpose input port)

Fixed to "0"

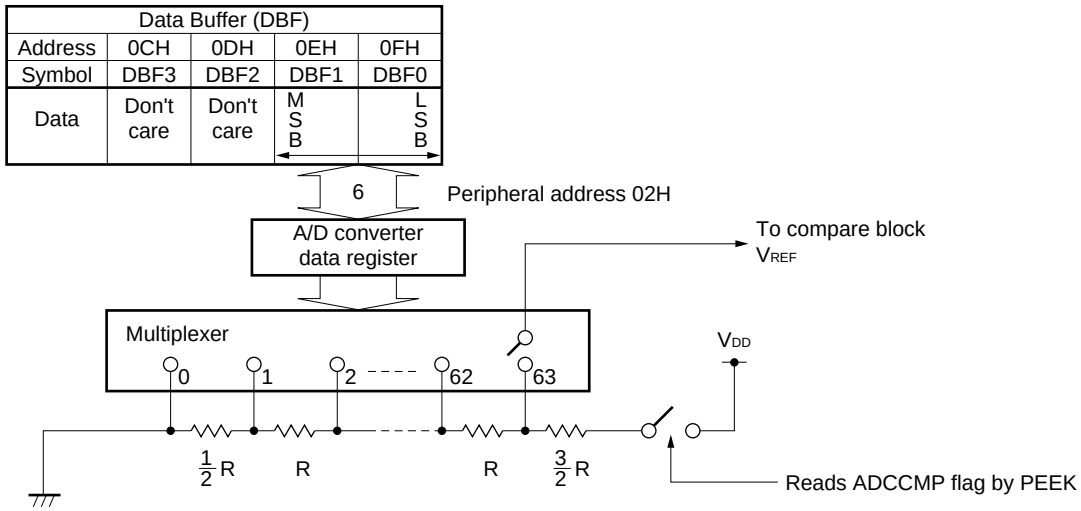
		b_3	b_2	b_1	b_0
On reset	Power-ON	0	1	1	1
	Clock stop		1	1	1
	CE		Retained		

16.4 Compare Voltage Generator Block

16.4.1 Configuration of compare voltage generator block

Figure 16-3 shows the configuration of the compare voltage generator block.

Figure 16-3. Configuration of Compare Voltage Generator Block



16.4.2 Function of compare voltage generator block

The compare voltage generator block switches the multiplexer by using the 6-bit data set to the A/D converter data register (ADCR: peripheral address 02H) and generates a compare voltage.

This means that this block is an R-string D/A converter.

The compare voltage can be set in 64 steps by the R string (resistance division).

The supply voltage of the R-string is the same as the supply voltage V_{DD} of the device.

A voltage is supplied to the R-string resistor only when the A/D converter compare judge register described later is detected.

The compare voltage is compared with the voltage input to the compare block.

The following **16.4.3** describe the configuration and function of the A/D converter data register.

Table 16-1 lists the compare voltages.

16.4.3 Configuration and function of A/D converter data register (ADCR)

The A/D converter data register sets the compare voltage of the A/D converter.

Because this register is 6 bits long, the lower 6 bits of the data buffer are valid.

Name	Data Buffer			
Symbol	DBF3	DBF2	DBF1	DBF0
Address	0CH	0DH	0EH	0FH
Bit	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0
Data	Don't care	Don't care	Transfer data	

8

GET can be executed

PUT can be executed

Peripheral Register										
Name	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Symbol	Peripheral address	Peripheral hardware
A/D converter data register	0	0	Valid data						ADCR	02H	A/D converter

Sets compare voltage V_{REF} of A/D converter

0	V_{REF} = 0 V
1	$V_{REF} = \frac{x - 0.5}{64} \times V_{DD}$ V
x	
63	

Fixed to "0"

Table 16-1. Set Values of A/D Converter Data Register and Compare Voltages

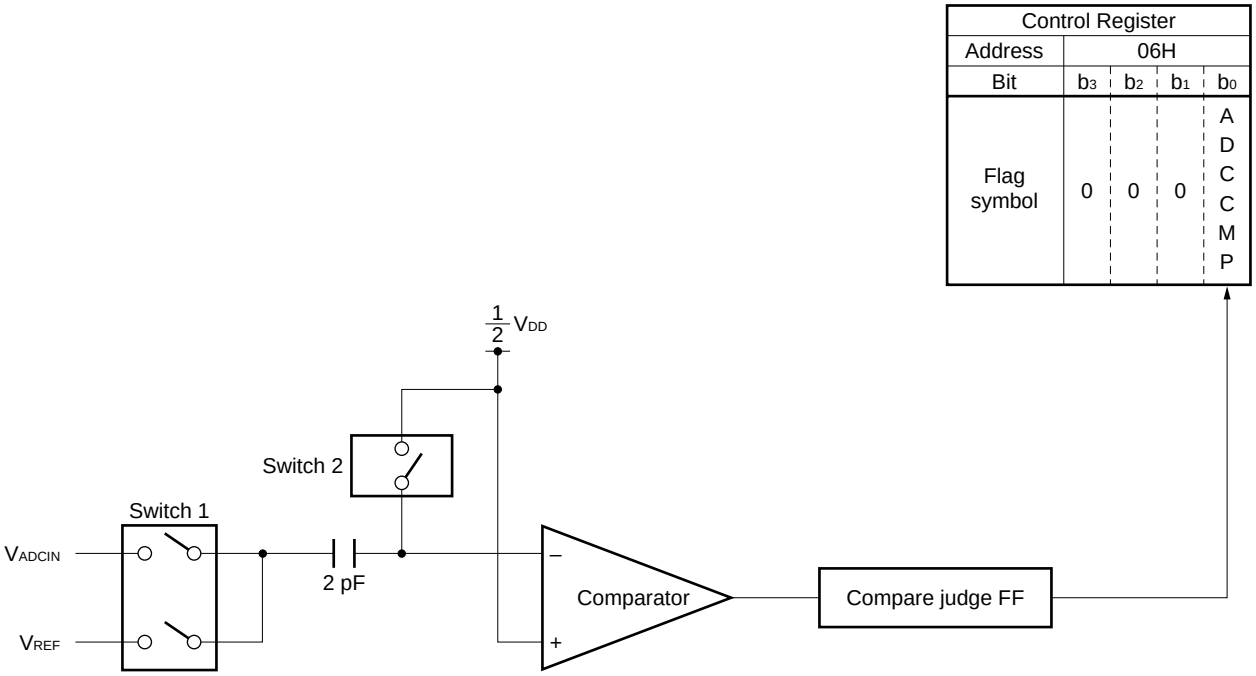
ADCR Set Data		Compare Voltage		ADCR Set Data		Compare Voltage	
DEC	HEX	Logic voltage Unit: × V_{DD} V	When V_{DD} = 5 V Unit: V	DEC	HEX	Logic voltage Unit: × V_{DD} V	When V_{DD} = 5 V Unit: V
0	00H	0	0	32	20H	31.5/64	2.461
1	01H	0.5/64	0.039	33	21H	32.5/64	2.539
2	02H	1.5/64	0.117	34	22H	33.5/64	2.617
3	03H	2.5/64	0.195	35	23H	34.5/64	2.695
4	04H	3.5/64	0.273	36	24H	35.5/64	2.773
5	05H	4.5/64	0.352	37	25H	36.5/64	2.852
6	06H	5.5/64	0.430	38	26H	37.5/64	2.930
7	07H	6.5/64	0.508	39	27H	38.5/64	3.008
8	08H	7.5/64	0.586	40	28H	39.5/64	3.086
9	09H	8.5/64	0.664	41	29H	40.5/64	3.164
10	0AH	9.5/64	0.742	42	2AH	41.5/64	3.242
11	0BH	10.5/64	0.820	43	2BH	42.5/64	3.320
12	0CH	11.5/64	0.898	44	2CH	43.5/64	3.398
13	0DH	12.5/64	0.977	45	2DH	44.5/64	3.477
14	0EH	13.5/64	1.055	46	2EH	45.5/64	3.555
15	0FH	14.5/64	1.133	47	2FH	46.5/64	3.633
16	10H	15.5/64	1.211	48	30H	47.5/64	3.711
17	11H	16.5/64	1.289	49	31H	48.5/64	3.789
18	12H	17.5/64	1.367	50	32H	49.5/64	3.867
19	13H	18.5/64	1.445	51	33H	50.5/64	3.945
20	14H	19.5/64	1.523	52	34H	51.5/64	4.023
21	15H	20.5/64	1.602	53	35H	52.5/64	4.102
22	16H	21.5/64	1.680	54	36H	53.5/64	4.180
23	17H	22.5/64	1.758	55	37H	54.5/64	4.258
24	18H	23.5/64	1.836	56	38H	55.5/64	4.336
25	19H	24.5/64	1.914	57	39H	56.5/64	4.414
26	1AH	25.5/64	1.992	58	3AH	57.5/64	4.492
27	1BH	26.5/64	2.070	59	3BH	58.5/64	4.570
28	1CH	27.5/64	2.148	60	3CH	59.5/64	4.648
29	1DH	28.5/64	2.227	61	3DH	60.5/64	4.727
30	1EH	29.5/64	2.305	62	3EH	61.5/64	4.805
31	1FH	30.5/64	2.383	63	3FH	62.5/64	4.883

16.5 Compare Block

16.5.1 Configuration of compare block

Figure 16-4 shows the configuration of the compare block.

Figure 16-4. Configuration of Compare Block



16.5.2 Function of compare block

The compare block compares voltage V_{ADCIN} input from a pin with internal compare voltage V_{REF} by using a comparator and outputs the result to the compare judge FF.

The compare judge FF can be detected by reading the ADCCMP flag of the A/D converter compare judge register.

The ADCCMP flag is set when $V_{ADCIN} > V_{REF}$, and is reset when $V_{ADCIN} < V_{REF}$.

The comparator compares the voltage when the ADCCMP flag is read.

In other words, when the ADCCMP flag is read by executing the “PEEK” instruction, switches 1 and 2 are operated to make the comparison.

Therefore, the time the A/D converter takes to make comparison once is equivalent to one instruction execution time (4.44 μs).

The following **16.5.3** describes the configuration and function of the A/D converter compare judge register.

16.5.3 Configuration and function of A/D converter compare judge register (ADCJDG)

The A/D converter compare judge register compares the input voltage V_{ADCIN} and compare voltage V_{REF} of the A/D converter.

The configuration and function of this register are illustrated below.

Name	Flag Symbol b_3	b_2	b_1	b_0	Address	Read/ Write
A/D converter compare judge register (ADCJDG)	0	0	0	ADCCMP	06H	R

b_0 (ADCCMP): Detects comparison result of A/D converter

0	$V_{ADCIN} < V_{REF}$
1	$V_{ADCIN} > V_{REF}$

b_3, b_2, b_1: Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	Undefined
	Clock stop	↓	↓	↓	Retained
	CE	↓	↓	↓	Retained

16.6 Performance of A/D Converter

The performance of the A/D converter is as follows:

Parameter	Performance
Resolution	1LSB
Input voltage range	0 – V_{DD}
Quantized error	$\pm \frac{1}{2}$ LSB
Over range	$\frac{62.5}{64} \times V_{DD}$
Error of offset, gain, non-linearity	$\pm \frac{3}{2}$ LSB[Note]

Note Including quantized error

16.7 Using A/D Converter

16.7.1 Comparing one compare voltage

Here is a program example:

Example To compare the input voltage V_{ADCIN} of the ADC_0 pin with compare voltage V_{REF} (31.5/64 V_{DD}) and branch to AAA if $V_{ADCIN} > V_{REF}$ and branch to BBB if $V_{ADCIN} < V_{REF}$

```
INIT:
    ADCR7   FLG   0.0EH.3      ; Dummy
    ADCR6   FLG   0.0EH.2      ; Dummy
    ADCR5   FLG   0.0EH.1      ; Defines each bit of data buffer as data setting flag of ADCR
    ADCR4   FLG   0.0EH.0
    ADCR3   FLG   0.0FH.3
    ADCR2   FLG   0.0FH.2
    ADCR1   FLG   0.0FH.1
    ADCR0   FLG   0.0FH.0

    CLR3    ADCCH2, ADCCH1, ADCCH0
                               ; Sets P1D0/ADC0 pin as A/D converter pin
START:
    INITFLG   NOT ADCR3,  NOT ADCR2,  NOT ADCR1,  NOT ADCR0
    INITFLG   NOT ADCR7,  NOT ADCR6,       ADCR5,  NOT ACCR4
    PUT       ADCR, DBF   ; Sets compare voltage VREF to 31.5/64 VDD
    SKT1      ADCCMP      ; Detects ADCCMP flag
    BR        AAA         ; Branches to AAA if ADCCMP flag is False (0)
    BR        BBB         ; Branches to BBB if ADCCMP flag is True (1)
```

16.7.2 Successive approximation by binary search

The A/D converter can compare only one compare voltage at a time.

To convert an input analog voltage into a digital signal, therefore, successive approximation must be executed by program.

If the processing time of the successive approximation program differs depending on the input voltage, it is not desirable in some cases because of the relation with the other programs.

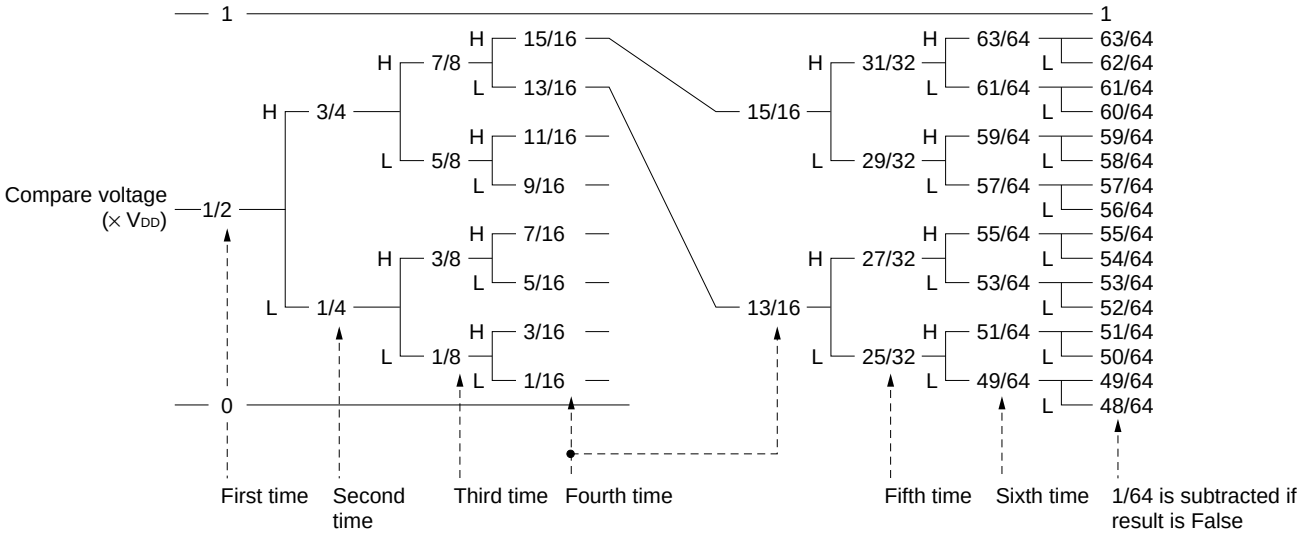
In these cases, executing binary search as described in (1) through (3) below is convenient.

(1) Concept of binary search

The concept of binary search is described below.

First, the compare voltage is set to 1/2 V_{DD}, and a voltage of 1/4 V_{DD} is added if the result of the comparison is True (if a high level is input) and a voltage of 1/4 V_{DD} is subtracted if the result is False (if a low level is input).

In the same way, comparison is sequentially performed by changing the voltage to be added or subtracted to 1/8 V_{DD} and then to 1/16 V_{DD} to 1/64 V_{DD}. If the result of the sixth comparison is False, 1/64 V_{DD} is subtracted and the operation is completed.



(2) Flowchart of binary search

: Selects pin to be used

: Sets compare voltage to 1/2 V_{DD}

: Detects comparison result

: Subtracts 1/2 V_{DD} if result is "0"

: Adds 1/4 V_{DD} to compare voltage regardless of whether result is "0" or "1"

: Detects comparison result

: Subtracts 1/4 V_{DD} if result is "0"

: Adds 1/8 V_{DD} to compare voltage regardless of whether result is "0" or "1"

: Detects comparison result

: Subtracts 1/8 V_{DD} if result is "0"

: Adds 1/16 V_{DD} to compare voltage regardless of whether result is "0" or "1"

: Detects comparison result

: Subtracts 1/16 V_{DD} if result is "0"

: Adds 1/32 V_{DD} to compare voltage regardless of whether result is "0" or "1"

: Detects comparison result

: Subtracts 1/32 V_{DD} if result is "0"

: Adds 1/64 V_{DD} to compare voltage regardless of whether result is "0" or "1"

: Detects comparison result

: Subtracts 1/64 V_{DD} if result is "0"

: Completes conversion if result is "1"

(3) Program example of binary search

(a) Where conversion time is short

```
INIT:
        ADCR7   FLG   0.0EH.3      ; Dummy
        ADCR6   FLG   0.0EH.2      ; Dummy
        ADCR5   FLG   0.0EH.1      ; Defines each bit of data buffer as data setting flag of ADCR
        ADCR4   FLG   0.0EH.0
        ADCR3   FLG   0.0FH.3
        ADCR2   FLG   0.0FH.2
        ADCR1   FLG   0.0FH.1
        ADCR0   FLG   0.0FH.0

        CLR3    ADCCH2, ADCCH1, ADCCH0
                                   ; Sets P1D0/ADC0 pin as A/D converter pin
START:
        INITFLG  NOT ADCR3,  NOT ADCR2,  NOT ADCR1,  NOT ADCR0
        INITFLG  NOT ADCR7,  NOT ADCR6,       ADCR5,  NOT ADCR4
        PUT    ADCR, DBF  ; Sets compare voltage to 31.5/64 VDD
        SKT1   ADCCMP     ; Detects ADCCMP
        CLR1   ADCR5      ; Subtracts 32/64 VDD if ADCCMP is "0" and,
        SET1   ADCR4      ; adds 16/64 VDD
        PUT    ADCR, DBF
        SKT1   ADCCMP     ; Detects ADCCMP
        CLR1   ADCR4      ; Subtracts 16/64 VDD if ADCCMP is "0" and,
        SET1   ADCR3      ; adds 8/64 VDD
        PUT    ADCR, DBF
        SKT1   ADCCMP     ; Detects ADCCMP
        CLR1   ADCR3      ; Subtracts 8/64 VDD if ADCCMP is "0" and,
        SET1   ADCR       ; adds 4/64 VDD
        PUT    ADCR, DBF
        SKT1   ADCCMP     ; Detects ADCCMP
        CLR1   ADCR2      ; Subtracts 4/64 VDD if ADCCMP is "0" and,
        SET1   ADCR1      ; adds 2/64 VDD
        PUT    ADCR, DBF
        SKT1   ADCCMP     ; Detects ADCCMP
        CLR1   ADCR1      ; Subtracts 2/64 VDD if ADCCMP is "0" and,
        SET1   ADCR0      ; adds 1/64 VDD
        PUT    ADCR, DBF
        SKT1   ADCCMP     ; Detects ADCCMP
        CLR1   ADCR0      ; Subtracts 1/64 VDD if ADCCMP is "0"
END:
```

A/D conversion

Number of program steps : 31
Number of execution steps : 31
A/D conversion time : 137.8 μs

(b) Where number of program steps is small

```
ADWORK1   MEM    0.00H      ; Work area for changing compare voltage
ADWORK0   MEM    0.01H

          INITFLG  NOT ADCCH2,  NOT ADCCH1,  NOT ADCCH0
                             ; Set P1D0/ADC0 pin as A/D converter pin
START:                                                          ┐
          MOV    DBF1, #0010B      ; Sets initial value of compare │
                                   ; voltage 31.5/64 VDD        │
          MOV    DBF0, #0000B                                  │
          MOV    ADWORK1, #0001B                               │
          MOV    ADWORK0, #0000B                               │
AD_CHECK:                                                       │
          PUT    ADCR, DBF         ; Sets compare voltage VREF │
          SKT1   ADCCMP            ; Detects ADCCMP flag       │
          BR     ADIN_L                                        │
          ADD    DBF0, ADWORK0     ; Increases compare voltage │
                                   ; if ADCCMP flag is "1"     │
          ADDC   DBF1, ADWORK1                                 │
          BR     NEXT_AD                                       │
ADIN_L                                                          │ A/D conversion
          SUB    DBF0, ADWORK0     ; Decreases compare voltage │
                                   ; if ADCCMP flag is "0"     │
          SUBC   DBF1, ADWORK1                                 │
;         NOP                      ; Described to keep A/D     │
                                   ; conversion time constant  │
NEXT_AD:                                                        │
          RORC   ADWORK1                                       │
          RORC   ADWORK0                                       │
          SKT1   CY                ; 6 bits have been compared?│
          BR     AD_CHECK                                      │
          PUT    ADCR, DBF                                     │
          SKT1   ADCCMP                                        │
          AND    DBF0, #1110B                                  │
          :                                                    ┘
          Number of program steps      : 22
          Number of execution steps    : 58 to 63
          A/D conversion time          : 257.8 to 280 μs

     After keeping A/D conversion time constant,

          Number of program steps      : 23
          Number of execution steps    : 63
          A/D conversion time          : 280 μs
```

16.8 Notes on Using A/D Converter

When the $P0D_3/ADC_5$ to $P0D_0/ADC_2$ pin is used as the A/D converter pin and when it is specified that the halt status be released by key input, the halt status may not be set.

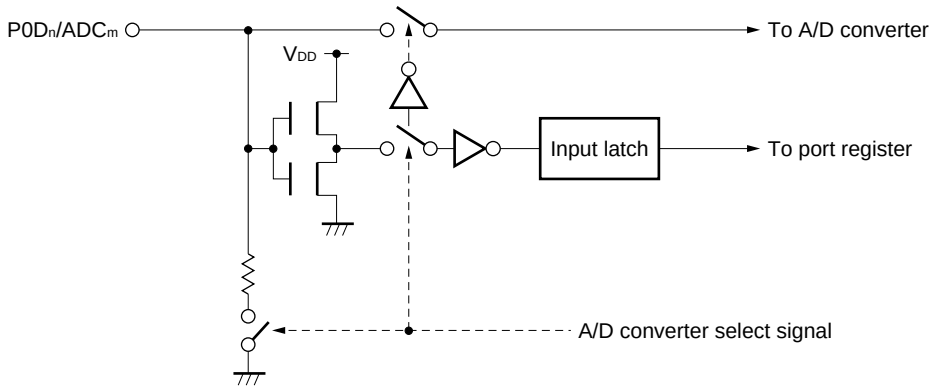
This is because the pin set as the A/D converter pin is disconnected from the latch of the input port as described in **12.4 Halt Function**.

Figure 16-5 below shows the relation between the $P0D_3/ADC_5$ through $P0D_0/ADC_2$ pins and the input latch.

As shown in this figure, if a high level happens to be input to an A/D converter pin set by the A/D converter select signal, the input latch retains “1”.

Therefore, even if it is specified that the halt status be released by key input, it is judged that a high level is input to this pin, and the halt status is released as soon as it has been set.

Figure 16-5. Relation between $P0D_3/ADC_5$ through $P0D_0/ADC_2$ Pins and Input Latch



16.9 Reset Status

16.9.1 On power-ON reset

All the $P0D_3/ADC_5$ through $P0D_0/ADC_2$ pins and $P1D_1/ADC_1$ and $P1D_0/ADC_0$ pins are set in the general-purpose input port mode.

16.9.2 On execution of clock stop instruction

All the $P0D_3/ADC_5$ through $P0D_0/ADC_2$ pins and $P1D_1/ADC_1$ and $P1D_0/ADC_0$ pins are set in the general-purpose input port mode.

16.9.3 On CE reset

The pin selected as the A/D converter pin is retained as is.

17. D/A CONVERTER (DAC)

The D/A converter (DAC) outputs signals by means of variable-duty PWM (Pulse Width Modulation).

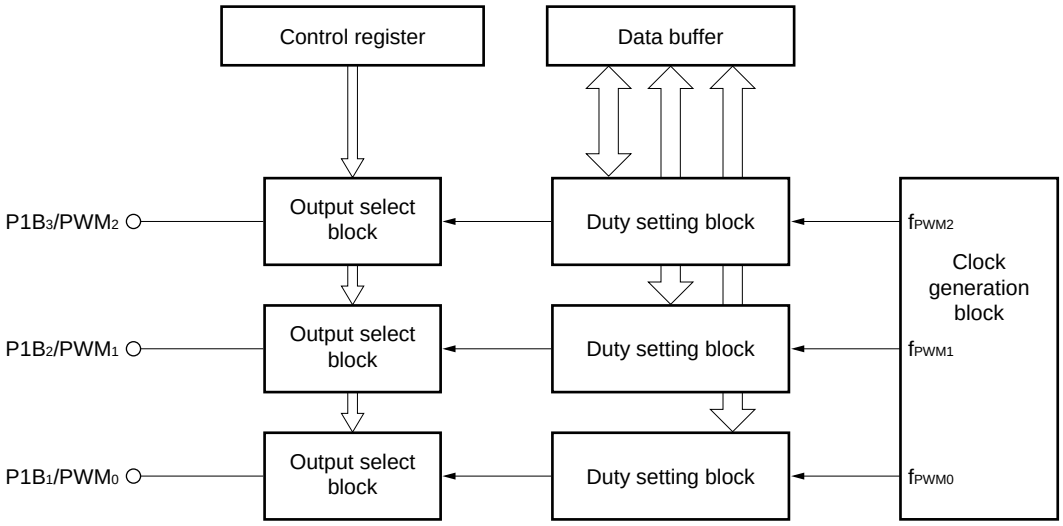
By connecting an external lowpass filter to the D/A converter, digital signals can be converted into analog signals.

17.1 Configuration of D/A Converter

Figure 17-1 shows the block diagram of the D/A converter.

As shown in this figure, the D/A converter consists of an output select block and a duty setting block for each pin, and a clock generation block.

Figure 17-1. Block Diagram of D/A Converter



17.2 Functional Outline of D/A Converter

Each pin of the D/A converter outputs a variable-duty signal independently of the other pins.

The output frequency is 4394.5 Hz, and the duty factor can be changed in 256 steps.

The following **17.2.1** through **17.2.3** outline the functions of the respective blocks.

17.2.1 Output select block

An output select block specifies whether each pin is used as a general-purpose output port pin or D/A converter pin.

This selection is made by using the PWM mode select register (PWMMODE: RF address 13H) (refer to **17.3**).

17.2.2 Duty setting block

A duty setting block outputs a variable-duty signal whose duty factor can be changed in 256 steps.

The duty factor of each pin is independently set by the PWM data register (PWMR0, PWMR1, or PWMR2: peripheral address 05H, 06H, or 07H) via data buffer (refer to **17.4**).

17.2.3 Clock generation block

The clock generation block generates the basic clock that is used to set a duty factor (refer to **17.4**).

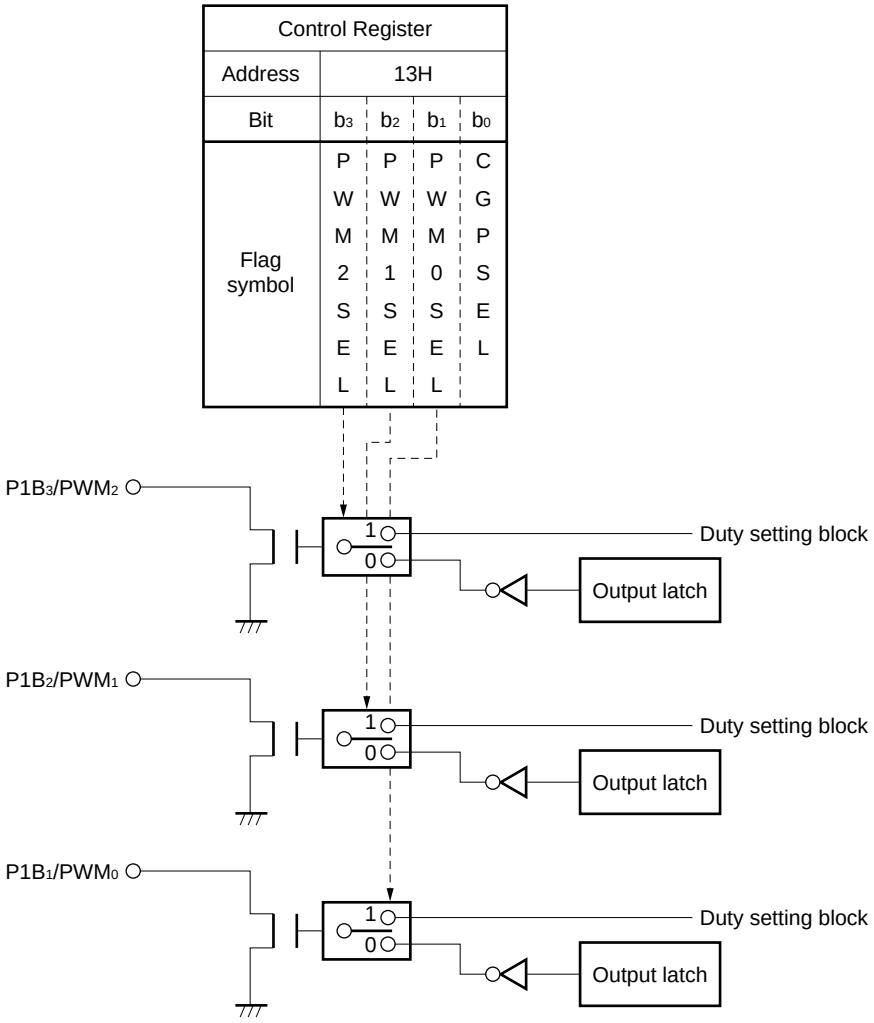
The frequency f_{PWM} of the generated clock is 1125 kHz.

17.3 Output Select Block

17.3.1 Configuration of output select block

Figure 17-2 shows the configuration of the output select block.

Figure 17-2. Configuration of Output Select Block



17.3.2 Function of output select block

The output select block selects whether the P1B3/PWM2 through P1B1/PWM0 pins are used as general-purpose output port pins or D/A converter pins.

This selection is made by the PWM2SEL, PWM1SEL, and PWM0SEL flags of the PWM mode select register. Each pin is selected independently of the others.

The P1B3/PWM2 through P1B1/PWM0 pins are N-ch open-drain output pins and therefore, must be connected with external pull-up resistors.

The following **17.3.3** describe the configuration and function of the PWM mode select register.

17.3.3 Configuration and function of PWM mode select register (PWMMODE)

The PWM mode select register selects a pin that is used for the D/A converter (PWM output) or clock generator port (CGP).

The configuration and function of this register are illustrated below.

For the details of the CGP, refer to **18. CLOCK GENERATOR PORT (CGP)**.

Name	Flag Symbol b3	b2	b1	b0	Address	Read/Write
PWM mode select register (PWMMODE)	PWM2SEL	PWM1SEL	PWM0SEL	CGPSEL	13H	R/W

CGPSEL

	Selects general-purpose output port or clock generator port
0	P1B0/CGP pin is used as general-purpose output port
1	P1B0/CGP pin is used as clock generator port

PWM0SEL

	Selects general-purpose output port or D/A converter (PWM output)
0	P1B1/PWM0 pin is used as general-purpose output port
1	P1B1/PWM0 pin is used as D/A converter

PWM1SEL

	Selects general-purpose output port or D/A converter (PWM output)
0	P1B2/PWM1 pin is used as general-purpose output port
1	P1B2/PWM1 pin is used as D/A converter

PWM2SEL

	Selects general-purpose output port or D/A converter (PWM output)
0	P1B3/PWM2 pin is used as general-purpose output port
1	P1B3/PWM2 pin is used as D/A converter

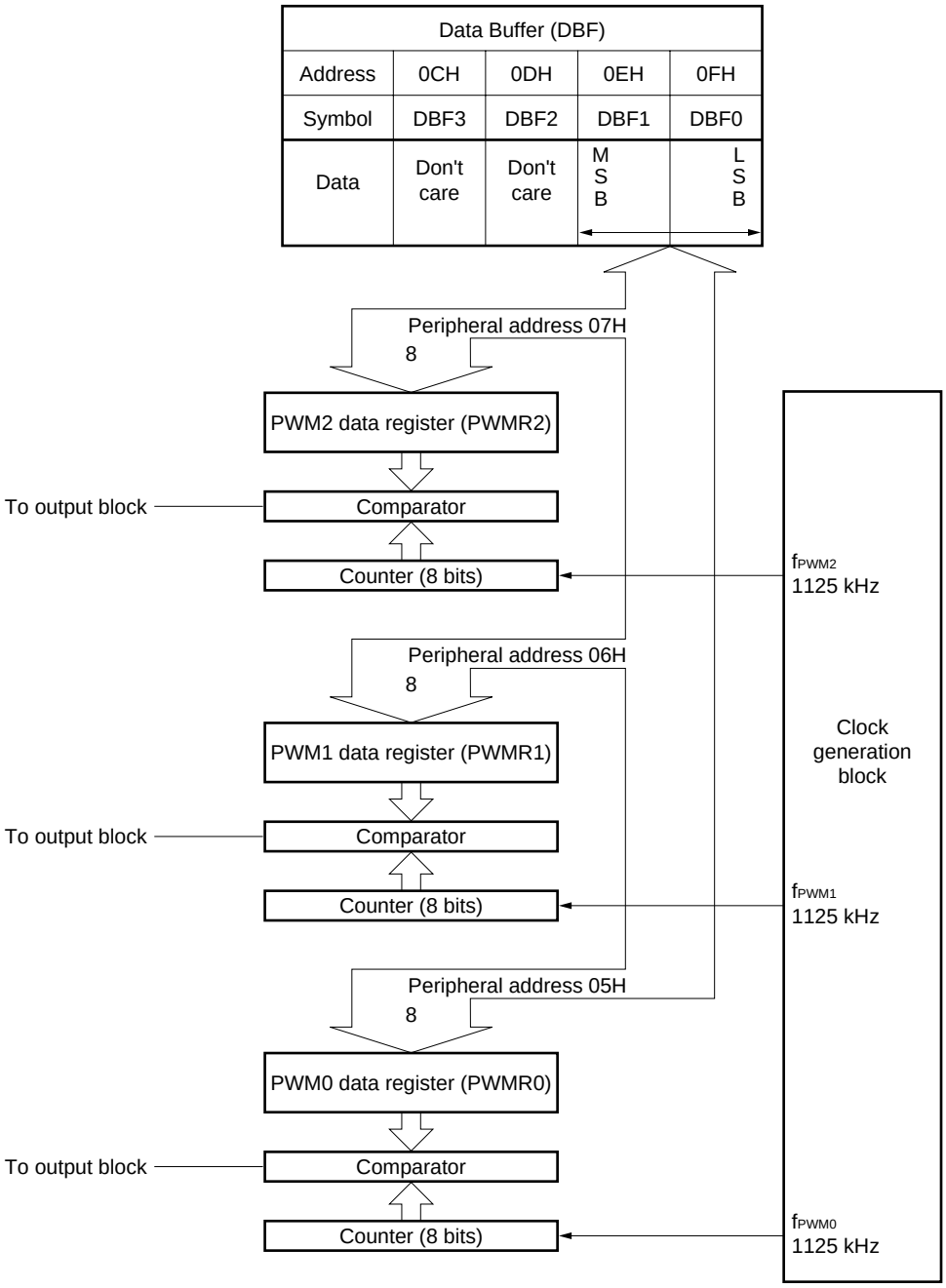
		b3	b2	b1	b0
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	Retained			

17.4 Duty Setting Block and Clock Generation Block

17.4.1 Configuration of duty setting block and clock generation block

Figure 17-3 shows the configuration of the duty setting block and clock generation block.

Figure 17-3. Configuration of Duty Setting Block and Clock Generation Block

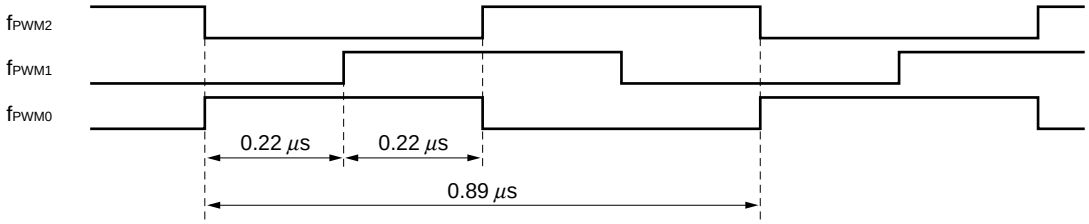


17.4.2 Function and operation of clock generation block

The clock generation block outputs the basic clocks (f_{PWM2}, f_{PWM1}, and f_{PWM0}) to set the duty factors of the output signals (PWM_2, PWM_1, and PWM_0 pins).

The output frequencies of all f_{PWM2}, f_{PWM1}, and f_{PWM0} are 1125 kHz (0.89 μs).

However, there are the following phase differences among f_{PWM2}, f_{PWM1}, and f_{PWM0}.



17.4.3 Function and operation of duty setting block

The duty setting block compares the values set to the respective PWM data registers (PWM2, PWM1, and PWM0) with the values of the basic clocks (f_{PWM2}, f_{PWM1}, and f_{PWM0}) counted by the respective 8-bit counters, and outputs a high level if the value of the PWM register is greater or a low level if the value of the PWM register is smaller.

Where the value set to the PWM register is "x", therefore, the duty factor is as follows:

$$\text{Duty: } D = \frac{x + 0.25}{256} \times 100\%$$

0.25 is an offset, and a high level is output even when x = 0.

Because the frequency of the basic clock is 1125 kHz, the frequency and cycle of the output signal are as follows:

$$\text{Frequency: } f = \frac{1125\ \text{kHz}}{256} = 4394.5\ \text{Hz}$$

$$\text{Cycle: } t = \frac{256}{1125\ \text{kHz}} = 227.6\ \mu\text{s}$$

Data is set to each PWM data register independently via data buffer.

Therefore, a signal with a different duty factor can be output by each pin.

The following **17.4.4** and **17.4.5** describe the configuration and function of the PWM data register, and relation between the output waveform and duty factor of each pin.

17.4.4 Configuration and function of each PWM data register

The function of each PWM data register is illustrated below.

The PWM data register sets the duty factor of the output signal (PWM output) of the D/A converter.

Name	Data Buffer															
Symbol	DBF3				DBF2				DBF1				DBF0			
Address	0CH				0DH				0EH				0FH			
Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data	Don't care				Don't care				Transfer data							

8

GET can be executed

PUT can be executed

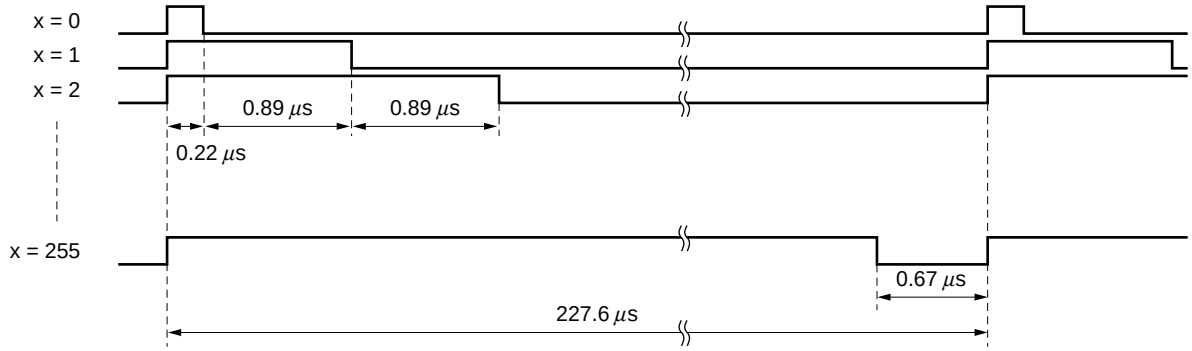
Peripheral Register											
Name	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Symbol	Peripheral address	Peripheral hardware
PWM0 data register	Valid data								PWMR0	05H	PWM_0 pin
PWM1 data register									PWMR1	06H	PWM_1 pin
PWM2 data register									PWMR2	07H	PWM_2 pin

	Sets PWM output duty of each pin
0	Duty: $D = \frac{x + 0.25}{256} \times 100\%$
x	Frequency: $f = \frac{1125}{256}$ kHz
255	= 4394.5 Hz

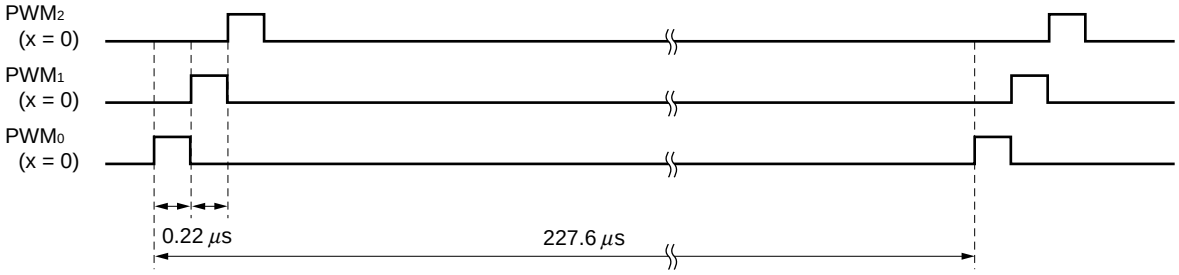
17.4.5 Relation between output waveform of D/A converter and each pin

(1) shows the relation between the duty factor and output waveform. (2) shows the relation of the output waveform of each pin.

(1) Duty and output waveform



(2) Output waveform of each pin



17.5 Reset Status

17.5.1 On power-ON reset

The P1B3/PWM2 through P1B1/PWM0 pins are specified as the general-purpose output port pins.

The output value is “undefined”.

The value of each PWM data register is “undefined”.

17.5.2 On execution of clock stop instruction

The P1B3/PWM2 through P1B1/PWM0 pins are specified as the general-purpose output port pins.

The output value is the “previous contents of the output latch”.

Each PWM data register retains the previous value.

17.5.3 On CE reset

The P1B3/PWM2 through P1B1/PWM0 pins retain the previous output status.

Therefore, the pin used as a D/A converter pin retains the PWM output.

17.5.4 In halt status

The P1B3/PWM2 through P1B1/PWM0 pins retain the previous output status.

Therefore, the pin used as a D/A converter pin retains the PWM output.

18. CLOCK GENERATOR PORT (CGP)

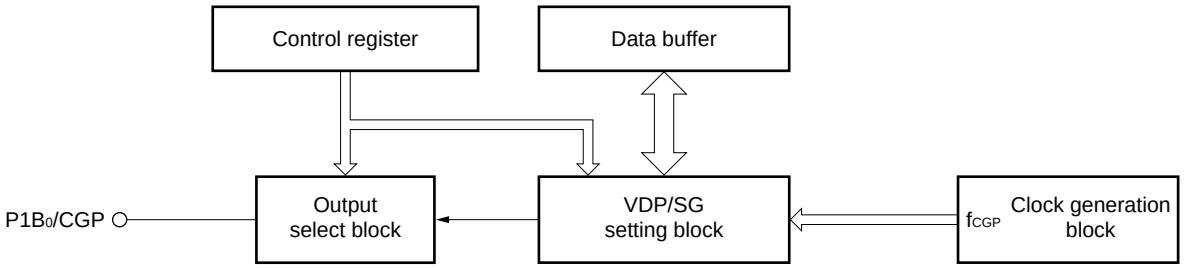
The clock generator port outputs signals in two modes: VDP (Variable Duty Pulse) mode in which the duty factor is changed, and SG (Signal Generator) mode in which the frequency is changed.

18.1 Configuration of Clock Generator Port

Figure 18-1 shows the block diagram of the clock generator port.

As shown in this figure, the clock generator port consists of an output select block, a VDP/SG setting block, and a clock generation block.

Figure 18-1. Block Diagram of Clock Generator Port



18.2 Functional Outline of Clock Generator Port

The clock generator port outputs a variable-duty signal (VDP function) or variable-frequency signal (SG function) from the P1B0/CGP pin.

The VDP function can changes the duty factor in 64 steps.

The SG function can changes the frequency in 64 steps.

The following **18.2.1** through **18.2.3** outline the functions of the respective blocks.

Because the clock generator port is shares the hardware with the frequency counter that is described later, the clock generator and frequency counter cannot be used at the same time. For details, refer to **18.7**.

18.2.1 Output select block

The output select block selects whether the P1B0/CGP pin is used as a general-purpose output port pin or the clock generator port.

This selection is made by the PWM mode select register (PWMMODE: RF address 13H).

For details, refer to **18.3**.

18.2.2 VDP/SG setting block

The VDP/SG setting block selects the VDP or SG function, and outputs a variable-duty signal when the VDP function is selected and a variable-frequency signal when the SG function is selected.

The duty factor when the VDP function is selected and the frequency when the SG function is selected is selected by the CGP data register (CGPR: peripheral address 20H) via data buffer.

For details, refer to **18.4**.

18.2.3 Clock generation block

The clock generation block generates a basic clock that is used to set a duty factor for the VDP function or a frequency for the SG function.

The frequency f_{CGP} of the generated clock is 18 kHz.

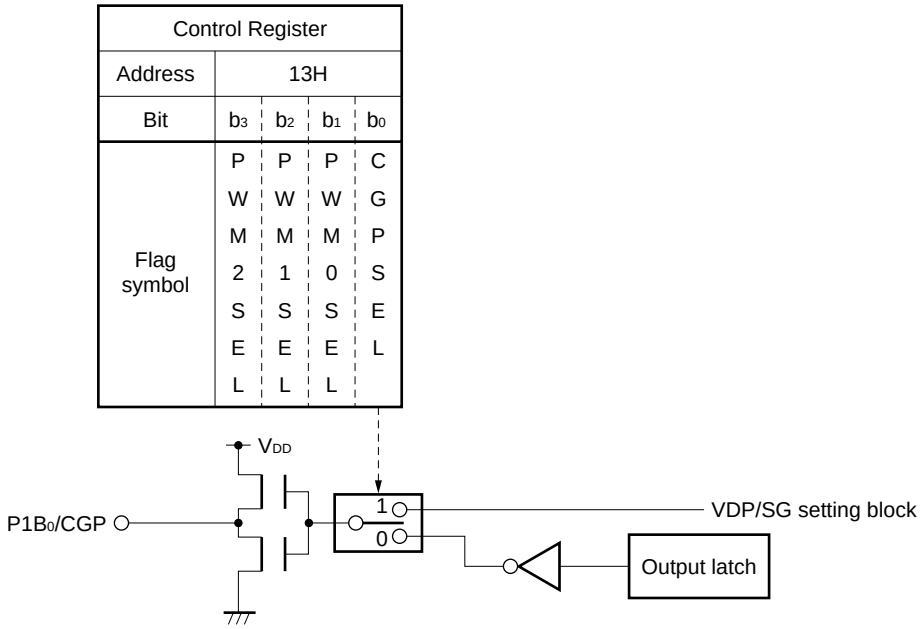
For details, refer to **18.4**.

18.3 Output Select Block

18.3.1 Configuration of output select block

Figure 18-2 shows the configuration of the output select block.

Figure 18-2. Configuration of Output Select Block



18.3.2 Function of output select block

The output select block selects whether the P1B0/CGP pin is used as a general-purpose output port pin or the clock generator port.

This selection is made by the CGPSEL flag of the PWM mode select register.

The following **18.3.3** describes the configuration and function of the PWM mode select register.

18.3.3 Configuration and function of PWM mode select register (PWMMODE)

The PWM mode select register selects pins that are used as a D/A converter pins and clock generator port.

The configuration and function of this register are shown below.

For the details of the D/A converter, refer to **17. D/A CONVERTER (DAC)**.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
PWM mode select register (PWMMODE)	PWM2SEL	PWM1SEL	PWM0SEL	CGPSEL	13H	R/W

CGPSEL — Selects general-purpose output port or clock generator port

0	$P1B_0$/CGP pin is used as general-purpose output port
1	$P1B_0$/CGP pin is used as clock generator port

PWM0SEL — Selects general-purpose output port or D/A converter (PWM output)

0	$P1B_1$/PWM_0 pin is used as general-purpose output port
1	$P1B_1$/PWM_0 pin is used as D/A converter

PWM1SEL — Selects general-purpose output port or D/A converter (PWM output)

0	$P1B_2$/PWM_1 pin is used as general-purpose output port
1	$P1B_2$/PWM_1 pin is used as D/A converter

PWM2SEL — Selects general-purpose output port or D/A converter (PWM output)

0	$P1B_3$/PWM_2 pin is used as general-purpose output port
1	$P1B_3$/PWM_2 pin is used as D/A converter

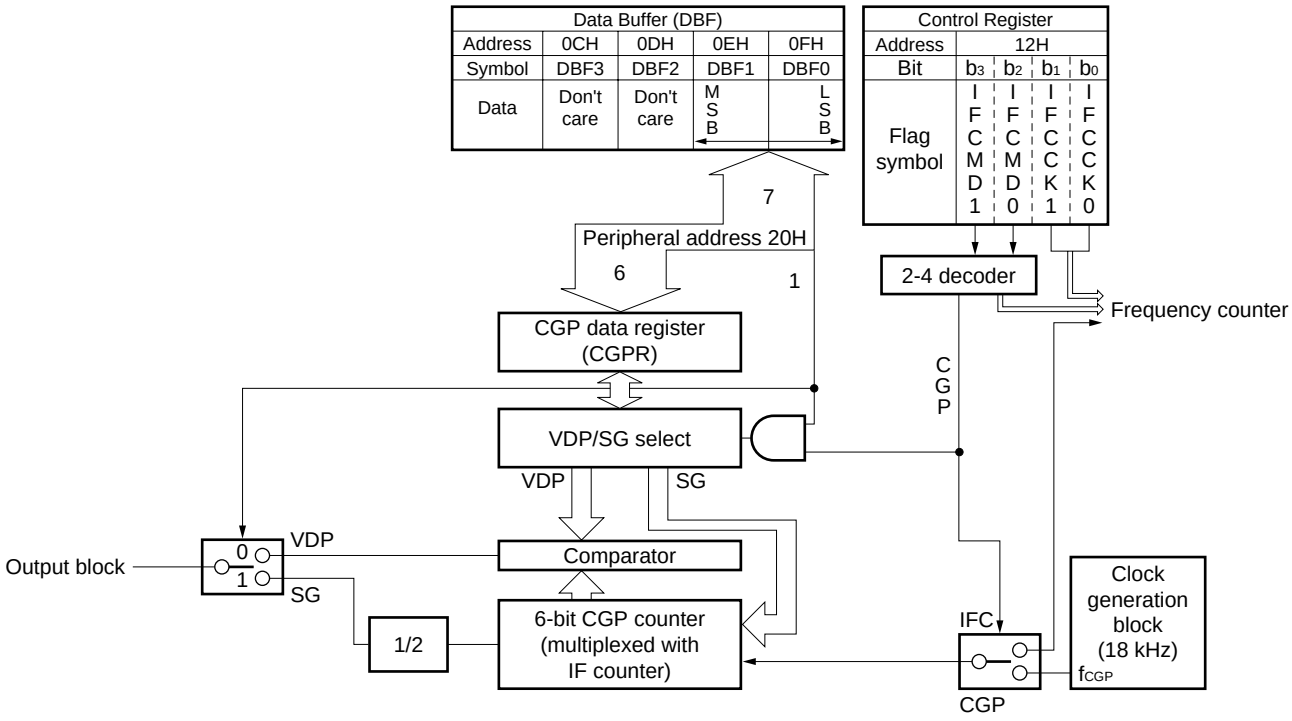
		PWM2SEL	PWM1SEL	PWM0SEL	CGPSEL
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	Retained			

18.4 VDP/SG Setting Block and Clock Generation Block

18.4.1 Configuration of VDP/SG setting block and clock generation block

Figure 18-3 shows the configuration of the VDP/SG setting block and clock generation block.

Figure 18-3. Configuration of VDP/SG Setting Block and Clock Generation Block



18.4.2 Function and operation of clock generation block

The clock generation block outputs a basic clock (f_{CGP}) that is used to set a duty factor for the VDP function and a frequency for the SG function.

The output frequency is 18 kHz.

18.4.3 Function and operation of VDP/SG setting block

The VDP/SG setting block selects the VDP or SG function, and sets a duty factor for the VDP function or a frequency for the SG function.

The 6-bit counter (CGP counter) of the VDP/SG setting block is multiplexed with an IF counter that is described later, either the CGP function or frequency counter can be selected by the IF counter mode select register (IFCMODE: RF address 12H). (Refer to **18.4.4**)

(1) and (2) below describe the operations of the VDP function and SG function.

Data is set to the CGP data register (refer to **18.4.5**) via data buffer.

The following **18.4.6** shows the output waveforms of the VDP and SG functions.

18.4.7 lists the set values of the CGP data register, duty factors of the VDP function, and frequencies of the SG function.

(1) VDP function

When the VDP function is selected, the value set to the higher 6 bits of the CGP data register is compared with the value of the basic clock (f_{CGP}) counted by the CGP counter. If the value of the CGP data register is greater, a high level is output; if the value of the CGP data register is smaller, a low level is output.

Where the value "x" set to the CGP register is "x", the duty factor D_{VDP} is as follows:

Duty: $$D_{VDP} = \frac{x + 2}{67} \times 100\%$$

"2" is an offset and a pulse is output even when x = 0.

Because the frequency of the basic clock is 18 kHz, the frequency f_{VDP} and cycle t_{VDP} of the output signal are as follows:

Frequency: $$f_{VDP} = \frac{18\ \text{kHz}}{67} = 268.7\ \text{Hz}$$

Cycle: $$t_{VDP} = \frac{67}{18\ \text{kHz}} = 3722.2\ \mu\text{s}$$

(2) SG function

The SG function compares the value set to the higher 6 bits of the CGP data register with the basic clock (f_{CGP}) counted by the CGP counter, and outputs a signal when the clock counts reaches "0".

Where the value set to the CGP register is "x", the output frequency f_{SG} is as follows:

Frequency: $$f_{SG} = \frac{18}{2(x + 2)}\ \text{kHz}$$

"2" is an offset and a pulse is output even when x = 0.

The duty factor D_{SG} is 50% as follows because a 1/2 divider is used:

Duty: $D_{SG} = 50\%$

18.4.4 Configuration and function of IF counter mode select register (IFCMODE)

The IF counter mode select register sets the functions of the frequency counter (IF counter and external gate counter) and clock generator port.

The configuration and function of this register are illustrated below.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
IF counter mode select register (IFCMODE)	IFCMD1	IFCMD0	IFCCK	IFCCK	12H	R/W

IFCCK	IFCMD	Sets gate time of IF counter and reference frequency of external gate counter	
		Gate time of IF counter	Reference frequency of external gate counter
0	0	1 ms	1 kHz
0	1	4 ms	100 kHz
1	0	8 ms	900 kHz
1	1	Open	0 kHz

IFCMD	IFCMD	Selects functions of IF counter, external gate counter (FCG), and clock generator port (CGP)
0	0	Clock generator port (CGP)
0	1	IF counter (FMIFC)
1	0	IF counter (AMIFC)
1	1	External gate counter (FCG)

		IFCMD1	IFCMD0	IFCCK	IFCCK
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	Retained			

The frequency counter and clock generator port cannot be used at the same time.

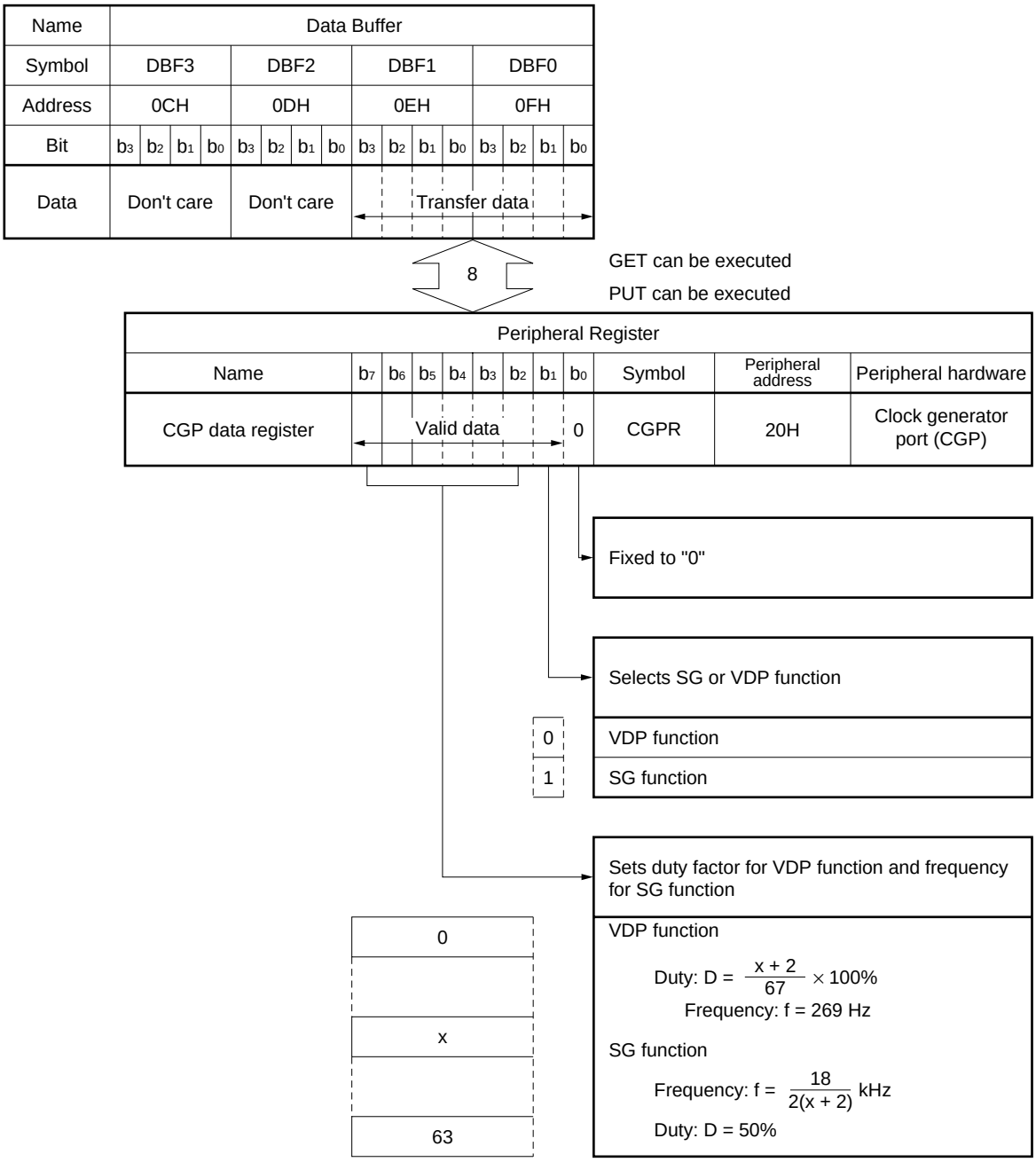
To use the clock generator port, reset the IFCMD1 and IFCMD0 flags to “0”.

After resetting these flags to “0”, the CGPSEL flag of the output select block must be set to “1”.

18.4.5 Configuration and function of CGP data register

The configuration and function of the CGP data register are illustrated below.

The CGP data register selects the VDP or SG function, and sets a duty factor for the VDP function and a frequency for the SG function.



The CGP counter is multiplexed with the higher 6 bits of the IF counter that is described in **20. FREQUENCY COUNTER (FC)**.

Therefore, the frequency counter and clock generator port cannot be used at the same time.

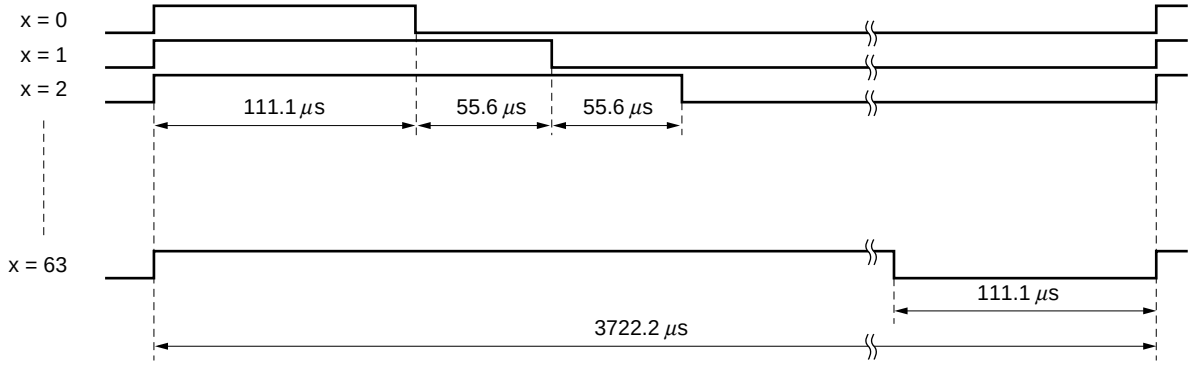
For details, refer to **18.7**.

18.4.6 Output waveforms of VDP and SG functions

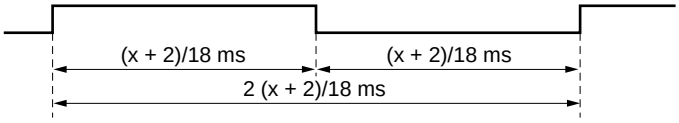
(1) shows the relation between the duty factor and output waveform of the VDP function.

(2) shows the output waveform of the SG function.

(1) Duty factor and output waveform of VDP function

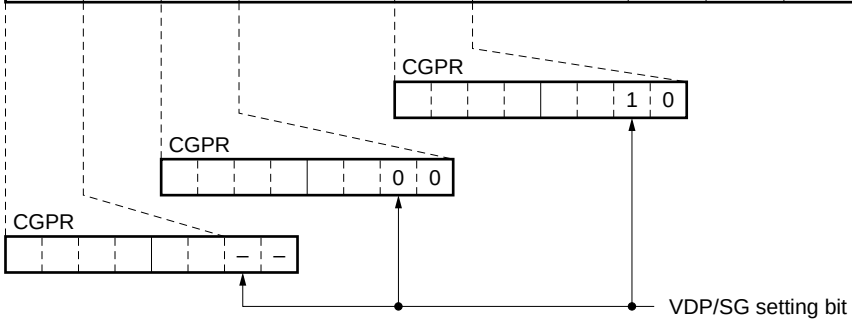


(2) Output waveform of SG function



18.4.7 List of set values of CGP data register (CGPR) and duty factors of VDP and frequencies of SG

CGPR Set Data (higher 6 bits)		Duty Factor of VDP Function		Frequency of SG Function		CGPR Set Data (higher 6 bits)		Duty Factor of VDP Function		Frequency of SG Function	
DEC	HEX	HEX		HEX	(Hz)	DEC	HEX	HEX		HEX	(Hz)
0	00H	00H	2/67	02H	4500.000	32	20H	80H	34/67	82H	264.706
1	01H	04H	3/67	06H	3000.000	33	21H	84H	35/67	86H	257.143
2	02H	08H	4/67	0AH	2250.000	34	22H	88H	36/67	8AH	250.000
3	03H	0CH	5/67	0EH	1800.000	35	23H	8CH	37/67	8EH	243.243
4	04H	10H	6/67	12H	1500.000	36	24H	90H	38/67	92H	236.842
5	05H	14H	7/67	16H	1285.714	37	25H	94H	39/67	96H	230.769
6	06H	18H	8/67	1AH	1125.000	38	26H	98H	40/67	9AH	225.000
7	07H	1CH	9/67	1EH	1000.000	39	27H	9CH	41/67	9EH	219.512
8	08H	20H	10/67	22H	900.000	40	28H	A0H	42/67	A2H	214.286
9	09H	24H	11/67	26H	818.182	41	29H	A4H	43/67	A6H	209.302
10	0AH	28H	12/67	2AH	750.000	42	2AH	A8H	44/67	AAH	204.545
11	0BH	2CH	13/67	2EH	692.308	43	2BH	ACH	45/67	AEH	200.000
12	0CH	30H	14/67	32H	642.857	44	2CH	B0H	46/67	B2H	195.652
13	0DH	34H	15/67	36H	600.000	45	2DH	B4H	47/67	B6H	191.489
14	0EH	38H	16/67	3AH	562.500	46	2EH	B8H	48/67	BAH	187.500
15	0FH	3CH	17/67	3EH	529.412	47	2FH	BCH	49/67	BEH	183.673
16	10H	40H	18/67	42H	500.000	48	30H	C0H	50/67	C2H	180.000
17	11H	44H	19/67	46H	473.684	49	31H	C4H	51/67	C6H	176.471
18	12H	48H	20/67	4AH	450.000	50	32H	C8H	52/67	CAH	173.077
19	13H	4CH	21/67	4EH	428.571	51	33H	CCH	53/67	CEH	169.811
20	14H	50H	22/67	52H	409.091	52	34H	D0H	54/67	D2H	166.667
21	15H	54H	23/67	56H	391.304	53	35H	D4H	55/67	D6H	163.636
22	16H	58H	24/67	5AH	375.000	54	36H	D8H	56/67	DAH	160.714
23	17H	5CH	25/67	5EH	360.000	55	37H	DCH	57/67	DEH	157.895
24	18H	60H	26/67	62H	346.154	56	38H	E0H	58/67	E2H	155.172
25	19H	64H	27/67	66H	333.333	57	39H	E4H	59/67	E6H	152.542
26	1AH	68H	28/67	6AH	321.429	58	3AH	E8H	60/67	EAH	150.000
27	1BH	6CH	29/67	6EH	310.345	59	3BH	ECH	61/67	EEH	147.541
28	1CH	70H	30/67	72H	300.000	60	3CH	F0H	62/67	F2H	145.161
29	1DH	74H	31/67	76H	290.323	61	3DH	F4H	63/67	F6H	142.857
30	1EH	78H	32/67	7AH	281.250	62	3EH	F8H	64/67	FAH	140.625
31	1FH	7CH	33/67	7EH	272.727	63	3FH	FCH	65/67	FEH	138.462



18.5 Using Clock Generator

The following **18.5.1** and **18.5.2** describe how to use the VDP and SG functions.

18.5.1 VDP function

An program example of using the VDP function is shown below.

As shown in this example, execute the "SET1 CGPSEL" instruction that sets the $P1B_0$/CGP pin as the CGP output pin after setting data to the CGP data register.

This is because if the contents of the CGP data register happens to be undefined (especially, at power-ON reset) when the "SET1 CGPSEL" instruction has been executed, an undefined signal is output.

Example To output a signal with a duty factor of 10/67

```
VDPDUTY   DAT  20H              ; Defines VDP function and data with duty factor = 10/67
CLR2      IFCMD1, IFCMD0        ; Sets 6-bit counter as CGP
MOV       DBF1, #VDPDUTY SHR 4 AND 0FH
MOV       DBF0, #VDPDUTY AND 0FH
PUT       CGPR, DBF             ; Sets VDP function and duty to CGP data register
SET1      CGPSEL                ; Sets P1B0/CGP pin as CGP output pin
                                ; Execute this instruction after setting data to CGP data register.
```

18.5.2 SG function

A program example of using the SG function is shown below.

When using the SG function, execute the "SET1 CGPSEL" instruction that sets the $P1B_0$/CGP as the CGP output pin after setting data to the CGP data register, in the same manner as in the example in **18.5.1**.

This is because if the contents of the CGP data register happens to be undefined (especially, at power-ON reset) when the "SET1 CGPSEL" instruction has been executed, an undefined signal is output.

Example To output a signal with a frequency of 900 Hz

```
SGFRQ     DAT  22H              ; Defines SG function and data with frequency = 900 Hz
CLR2      IFCMD1, IFCMD0        ; Sets 6-bit counter as CGP
MOV       DBF1, #SGFRQ SHR 4 AND 0FH
MOV       DBF0, #SGFRQ AND 0FH
PUT       CGPR, DBF             ; Sets SG function and frequency to CGP data register
SET1      CGPSEL                ; Sets P1B0/CGP pin as CGP output pin
                                ; Execute this instruction after setting data to CGP data register.
```

18.6 Reset Status

18.6.1 On power-ON reset

The P1B0/CGP pin is specified as a general-purpose output port pin because the CGPSEL flag is reset.

Because the value of the latch of the output port is “undefined”, undefined data is output.

The value of the CGP data register is “undefined”.

18.6.2 On execution of clock stop instruction

The P1B0/CGP pin is specified as a general-purpose output port pin because the CGPSEL flag is reset.

Because the value of the latch of the output port is the “previous contents of the output latch”, the value of the latch is output.

The value of the CGP data register retains the previous value.

18.6.3 On CE reset

The P1B0/CGP pin retains the previous output status.

18.6.4 In halt status

The P1B0/CGP pin retains the previous output status.

18.7 Notes on Using Clock Generator Port

The 6-bit CGP counter that sets the duty factor (for the VDP function) and frequency (for the SG function) of the clock generator port is multiplexed with the IF counter described in **20. FREQUENCY COUNTER (FC)**.

Therefore, the clock generator port and frequency counter cannot be used at the same time.

If the data of the IF counter mode select register and IF counter data register (IFC: peripheral address 43H) are manipulated when the clock generator port is used, the operation described in **18.7.1** is performed.

If the data of the IF counter mode select register and CGP data register are manipulated when the frequency counter is used, the operation described in **18.7.2** is performed.

18.7.1 When clock generator port is used

(1) If IFCMD1 and IFCMD0 flags of IF counter mode select register are manipulated

If a value other than “0” is written to the IFCMD1 and IFCMD0 flags, the P1B0/CGP pin retains the current output level when the data has been set, and stops the CGP operation.

If the flags are reset to “0”, the CGP operation is started.

(2) If IF counter data register is manipulated

The CGP operation is not affected even if the IF counter data register is read (by the GET instruction) or written (by the PUT instruction).

When the register is read, an “undefined” value is read. Nothing is changed even if the register is written. However, because the IF counter data register is a read-only peripheral register, do not write anything to this register.

18.7.2 When frequency counter is used

(1) If IFCMD1 and IFCMD0 flags of IF counter mode select register are manipulated

If "0" is written to the IFCMD1 and IFCMD0 flags, the $P1B_0$/CGP pin performs the operation specified by the CGP data register at that time when the data has been set.

To perform the CGP operation, however, the CGPSEL flag of the PWM mode select register must also be set.

If the IFCMD1 and IFCMD0 flags are set to the previous value, the frequency counter continues operation, but the count value is not accurate.

In other words, frequency counting is not performed while the CGP operation is selected.

(2) When CGP data register is manipulated

The frequency counter is not affected even if it is read (by the GET instruction) or written (by the PUT instruction).

When the counter is read, the value set when the CGP function was previously used ("undefined value" if the CGP function was not used) is read.

When it is written, the contents of bits 3 through 1 of DBF1 and DBF0 are written to the CGP data register.

19. SERIAL INTERFACE

The serial interface is used to transfer serial data in 8-bit units with an external device.

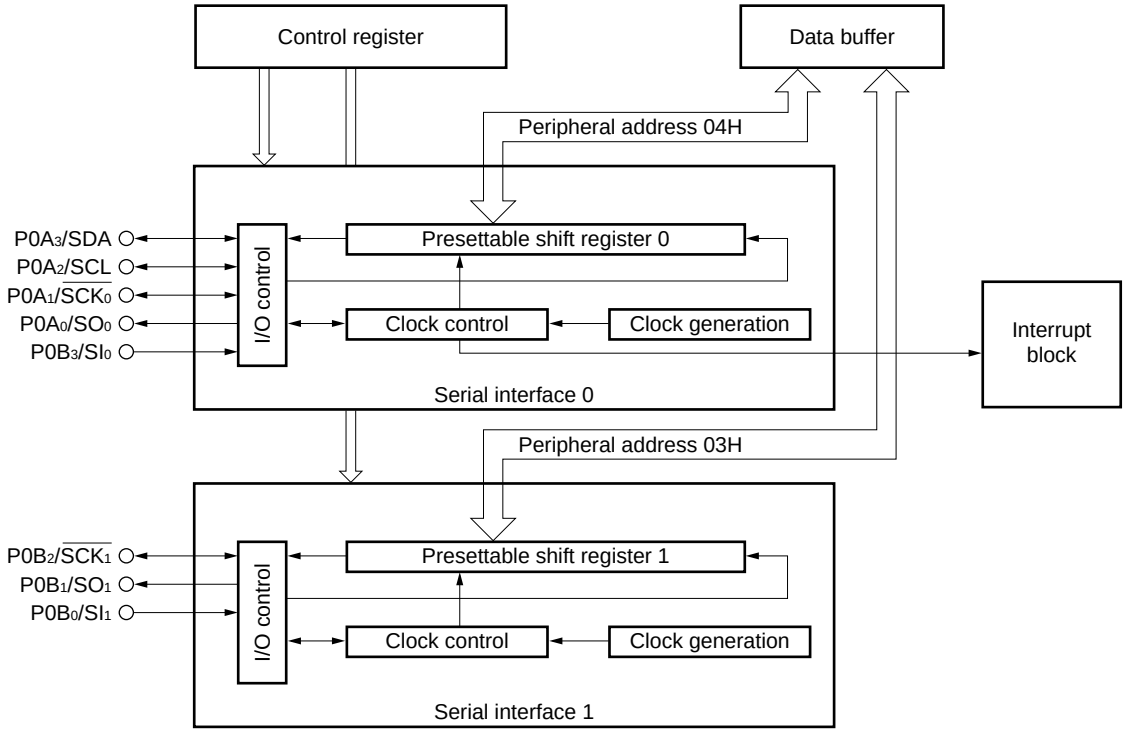
19.1 Configuration of Serial Interface

Figure 19-1 shows the block diagram of the serial interface.

As shown in this figure, the serial interface consists of two channels: serial interface 0 (SIO0) and serial interface 1 (SIO1).

Serial interfaces 0 and 1 respectively consist of an I/O control circuit, a presettable shift register, a clock control block, a clock generation block, and an interrupt block.

Figure 19-1. Block Diagram of Serial Interface



19.2 Functional Outline of Serial Interface

Table 19-1 shows the classification and communication mode of the serial interface.

As shown in this table, two serial interface channels, 0 (SIO0) and 1 (SIO1), are provided.

Serial interfaces 0 and 1 can be used simultaneously.

Serial interface 0 can be used in two-line or three-line mode. In the two-line mode, the $P0A_3$/SDA and $P0A_2$/SCL pins are used, and the $P0A_1/\overline{SCK_0}$, $P0A_0/SO_0$, and $P0B_3/SI_0$ pins are used in the three-line mode.

Moreover, the I^2C bus[Note] and serial I/O mode can be selected in the two-line mode.

Serial interface 1 can be used only in the three-line mode, in which the $P0B_2/\overline{SCK_1}$, $P0B_1/SO_1$, and $P0B_0/SI_1$ pins are used. The communication mode in this mode is the serial I/O mode.

Serial interface 0 is controlled by the following control registers:

- Serial I/O0 mode select register (SIO0MODE : RF address 08H)
- Serial I/O0 wait control register (SIO0WT : RF address 18H)
- Serial I/O0 wait status judge register (SIO0WSTR : RF address 19H)
- Serial I/O0 status judge register (SIO0STUS : RF address 28H)
- Serial I/O0 interrupt mode register (SIO0INT : RF address 38H)
- Serial I/O0 clock select register (SIO0CLK : RF address 39H)

Serial interface 1 is controlled by the serial I/O1 mode select register (SIO1MODE: RF address 02H) of the control registers.

Serial out data is set to and serial in data is read from serial interfaces 0 and 1 by the presettable shift registers 0 (SIO0SFR: peripheral address 04H) and 1 (SIO1SFR: peripheral address 03H) via data buffer.

The following **19.3** through **19.12** describe serial interface 0. **19.13** through **19.21** describe serial interface 1.

Note When using the I^2C bus mode (including when it is realized by program without using the peripheral hardware), advise NEC when you place an order for mask.

Table 19-1. Classification and Communication Modes of Serial Interface

Serial interface	Classification by Hardware	Number of Lines	Communication Mode	Pins Used
	Serial interface 0 (SIO0)	2 lines	I^2C bus mode	$P0A_3$/SDA
			Serial I/O mode	$P0A_2$/SCL
		3 lines	Serial I/O mode	$P0A_1/\overline{SCK_0}$ $P0A_0/SO_0$ $P0B_3/SI_0$
	Serial interface 1 (SIO1)	3 lines	Serial I/O mode	$P0B_2/\overline{SCK_1}$ $P0B_1/SO_1$ $P0B_0/SI_1$

19.3 Configuration of Serial Interface 0 (SIO0)

Figure 19-2 shows the block diagram of serial interface 0.

As shown in this figure, the shift clock control block of the serial interface 0 consists of a clock I/O pin block, a clock generation block, a wait control block, a clock count block, a start/stop detection block, and an interrupt control block.

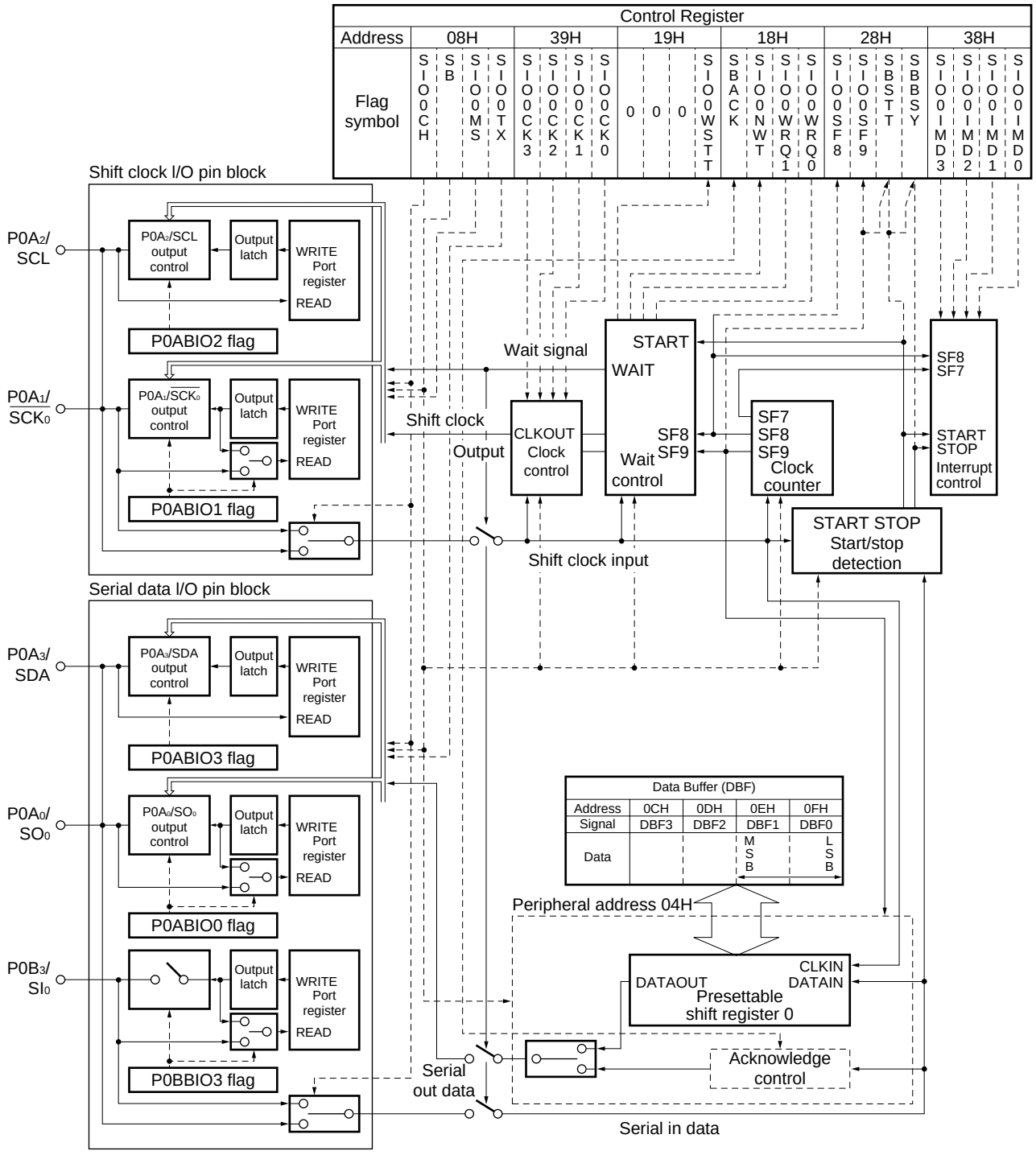
The serial data control block consists of a serial data I/O pin block, a presettable shift register 0, and an acknowledge block.

These blocks are controlled by the flags of control registers.

Data is written to or read from the presettable shift register 0 via data buffer.

19.4 outlines the functions of the respective blocks.

Figure 19-2. Block Diagram of Serial Interface 0



19.4 Functional Outline of Serial Interface 0

Serial interface 0 can be used in two modes in terms of the number of pins as shown in Table 19-1: three-line and two-line modes.

In the two-line mode, the $P0A_3$/SDA and $P0A_2$/SCL pins are used, and the $P0A_1/\overline{SCK_0}$, $P0A_0/SO_0$, and $P0B_3/SI_0$ pins are used in the three-line mode.

In the two-line mode, two communication modes, I^2C bus and serial I/O modes can be selected. Only the serial I/O mode can be used in the three-line mode.

In the I^2C bus and serial I/O modes, an internal clock (master) or external clock (slave) operation can be selected. Moreover, reception (RX) or transmission (TX) operation can be selected.

In the I^2C bus mode, serial communication between two or more devices can be executed with two lines.

The following **19.4.1** through **19.4.9** outline the functions of the respective blocks shown in Figure 19-2.

For the details of the respective blocks, refer to **19.5** through **19.10**.

19.4.1 Shift clock I/O pin block

This block selects a shift clock I/O pin.

The shift clock I/O pin is selected by the serial I/O0 mode register.

For details, refer to **19.5**.

19.4.2 Serial data I/O pin block

This block selects a serial data I/O pin.

The serial data I/O pin is selected by the serial I/O0 mode select register.

For details, refer to **19.5**.

19.4.3 Clock generation block

This block selects the clock frequency of the shift clock and controls the shift clock output timing.

The clock frequency is selected by the serial I/O0 clock select register.

For details, refer to **19.6**.

19.4.4 Clock counter

This counter counts the rising edges of the clock output by the shift clock output pin and outputs a signal at the seventh clock (SF7 signal), eighth clock (SF8 signal), and ninth clock (SF9 signal).

These signals are used to control wait (pause) cycle of serial communication and interrupt.

The eighth clock (SF8) and ninth clock (SF9) signals can be detected by the serial I/O0 status judge register.

For details, refer to **19.7**.

19.4.5 Start/stop detection block

This block detects a start and stop conditions in the I^2C bus mode.

It does not operate in the serial I/O mode.

The start and stop conditions can be detected by the SBSTT and SBBSY flags of the serial I/O0 status judge register.

For details, refer to **19.7**.

19.4.6 Presettable shift register 0 (SIO0SFR)

This shift register sets serial out data and stores serial in data.

It performs a shift operation in response to the clock input or output to the shift clock input pin, and inputs or outputs data.

The output data is set and the input data is read via data buffer.

For details, refer to **19.8**.

19.4.7 Wait control block

This block controls the wait (pause) and wait release (communication operation) states of serial communication.

The wait condition is set by the serial I/O0 wait control register SIO0WRQ1 and SIO0WRQ0 flag, and the wait state is released by the SIO0NWT flag.

For details, refer to **19.9**.

19.4.8 Acknowledge block

This block controls the acknowledge signal when the I^2C bus mode is used.

It does not operate in the serial I/O mode.

The acknowledge signal is set or read by the SBACK flag of the serial I/O0 wait control register.

For details, refer to **19.9**.

19.4.9 Interrupt control block

This block issues an interrupt request signal in response to signals from the clock counter and start/stop detection block.

The condition under which the interrupt request is issued is specified by the SIO0IMD3 through SIO0IMD0 flags of the serial I/O0 interrupt mode register.

For details, refer to **19.10**.

19.5 Shift Clock and Serial Data I/O Pin Control Block

The shift clock and data I/O pin control block controls the communication mode (I^2C bus or serial I/O mode), the number of pins used (two-line or three-line mode), and transmission or reception operation of serial interface 0.

These control operations are performed by the serial I/O0 mode select register (refer to **19.5.1**).

19.5.2 below shows the status of each pin set by the serial I/O0 mode select register.

19.5.1 Configuration and function of serial I/O0 mode select register (SIO0MODE)

Name	Flag Symbol b_3	b_2	b_1	b_0	Address	Read/ Write
Serial I/O0 mode select register (SIO1MODE)	SIO0CH	SB	SIO0MS	SIO0TX	08H	R/W

SIO0TX — Sets serial I/O of $P0A_3$/SDA pin in 2-line mode or $P0A_0/SO_0$ in 3-line mode (Selects "TX" or "RX")

	2-line ($P0A_3$/SDA pin)	3-line ($P0A_0/SO_0$ pin)
0	Serial input (Hi-Z): RX operation	General-purpose port
1	Serial output : TX operation	Serial output : TX operation

SIO0MS — Sets direction of shift clock

	I^2C bus mode	Serial I/O mode
0	Slave operation (external clock input)	External clock input
1	Master operation (internal clock output)	Internal clock output

SB, SIO0CH — Selects pin used and communication mode

SB	SIO0CH	
0	0	2-line serial I/O mode
0	1	I^2C bus mode
1	0	3-line serial I/O mode
1	1	Setting prohibited

		SB	SIO0CH	SIO0MS	SIO0TX
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

19.5.2 Pin status set by serial I/O0 mode select register

Table 19-2 shows the pin status set by the serial I/O0 mode select register.

As shown in this table, the I/O select flag must be manipulated to set each pin.

For the details of the I/O select flag, refer to **15. GENERAL-PURPOSE PORTS**.

Table 19-2. Pin Status Set by Serial I/O0 Mode Select Register

SIO0MODE							Pin						
b_3	b_2	Communi-cation mode	b_1	Clock direction	b_0	Serial I/O	Pin symbol	I/O select flag of each pin					Set pin status
SIO0CH	SB		SIO0MS		SIO0TX			P0ABIO3	P0ABIO2	P0ABIO1	P0ABIO0	P0BBIO3	
0	0	2-line serial I/O			0	Input (reception)	$P0A_3$/SDA	0					Serial input (Hi-Z)
								1					General-purpose output port
					1	Output (transmission)		0					Serial output
								1					
			0	External			$P0A_2$/SCL		0				External clock (Hi-Z)
									1				General-purpose output port
			1	Internal					0				Internal clock
									1				
							$P0A_1/\overline{SCK_0}$						General-purpose I/O port
							$P0A_0/SO_0$						General-purpose I/O port
							$P0B_3/SI_0$						General-purpose I/O port
0	1	I^2C bus			0	Input (reception)	$P0A_3$/SDA	0					Serial input (reception: Hi-Z)
								1					General-purpose output port
					1	Output (transmission)		0					Serial output (transmission)
								1					
			0	External (slave)			$P0A_2$/SCL		0				External clock (slave)
									1				General-purpose output port
			1	Internal (master)					0				Internal clock (master)
									1				
							$P0A_1/\overline{SCK_0}$						General-purpose I/O port
							$P0A_0/SO_0$						General-purpose I/O port
							$P0B_3/SI_0$						General-purpose I/O port
1	0	3-line serial I/O					$P0A_3$/SDA						General-purpose I/O port
							$P0A_2$/SCL						General-purpose I/O port
			0	External			$P0A_1/\overline{SCK_0}$			0			External clock
										1			General-purpose output port
			1	Internal						0			Internal clock
										1			
					0	Input (reception)	$P0A_0/SO_0$				0		General-purpose input port
											1		General-purpose output port
					1	Output (transmission)					0		Serial output
											1		
							$P0B_3/SI_0$					0	Serial input
												1	General-purpose output port
1	1						Setting prohibited						

19.6 Clock Generation Block

The clock generation block generates a clock when the internal clock is used (master operation) and controls the clock output timing.

The internal clock frequency f_{SC} is set by the serial I/O0 clock select register (refer to **19.6.1**).

The shift clock output by the clock generation block is valid only when the master operation (SIO0MS = 1) is performed.

The shift clock is successively output, until serial communication is placed in the wait status because the wait condition described later is satisfied.

19.6.2 and **19.6.3** below describe the clock output waveform and generation timing in each communication mode.

19.6.1 Configuration and function of serial I/O0 clock select register (SIO0CLK)

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Serial I/O0 clock select register (SIO0CLK)	SIO0CK3	SIO0CK2	SIO0CK1	SIO0CK0	39H	R/W

Sets internal shift clock frequency f_{SC} of serial interface 0		
0	0	37.5 kHz
0	1	75 kHz
1	0	112.5 kHz
1	1	225 kHz

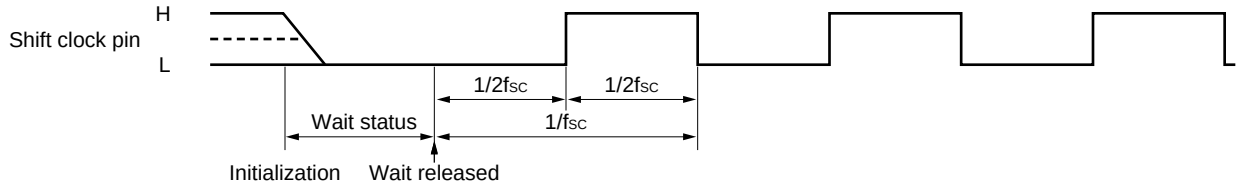
Fixed to "0"

On reset				
	Power-ON	0	0	Undefined
	Clock stop			Retained
	CE			Retained

19.6.2 Shift clock generation timing in I²C bus mode

(1) When wait status is released from initial status

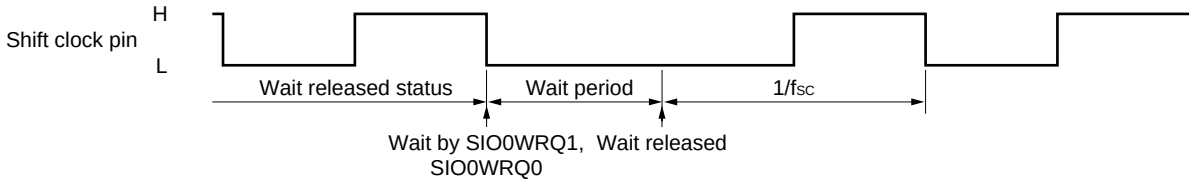
The "initial status" is the point at which the master operation in the I²C bus mode has been selected. In the wait status, a low level is output to the shift clock pin (P0A2/SCL pin).



(2) When wait operation is performed

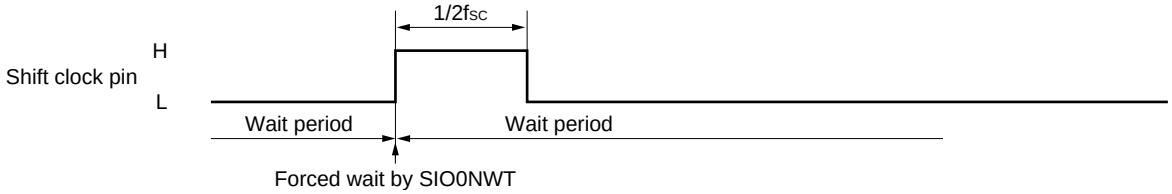
For the details of the wait operation, refer to **19.9**.

(a) When wait status is set under condition of SIO0WRQ0 and SIO0WRQ1 flags (normal operation)



(b) When forced wait status is set during wait status

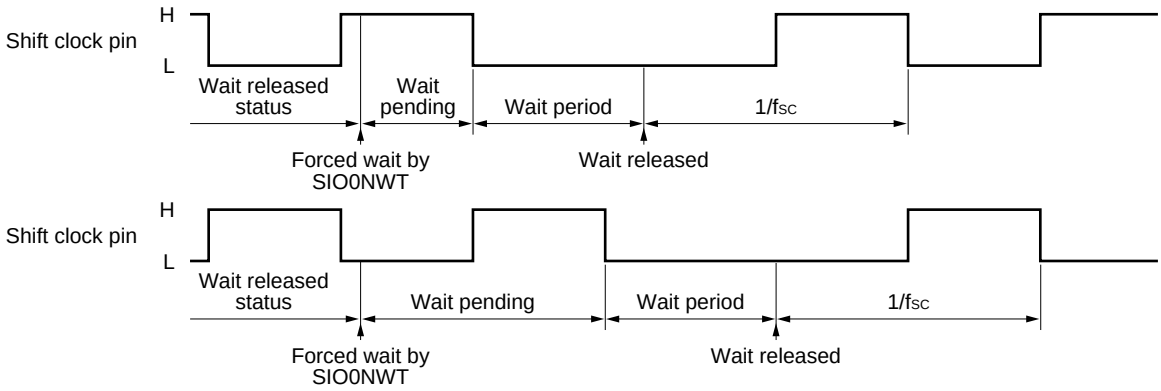
At this time, one pulse of the clock is output (however, the clock counter and presettable shift register 0 do not operate).



(c) When forced wait status is set during wait release

The wait status is set at the falling edge of the clock next to the one that has set the forced wait status. However, the clock counter and presettable shift register 0 stop operation when the forced wait status is set.

If the forced wait status is set while the shift clock pin is low, the clock counter and presettable shift register operates for the duration of one pulse.



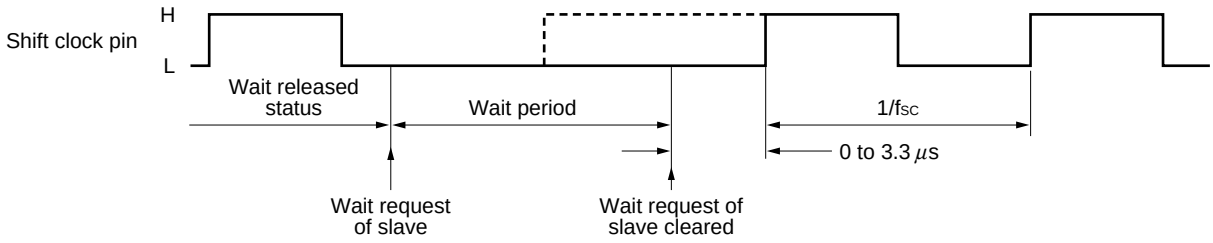
(d) If wait status is released during wait release

Nothing is changed.

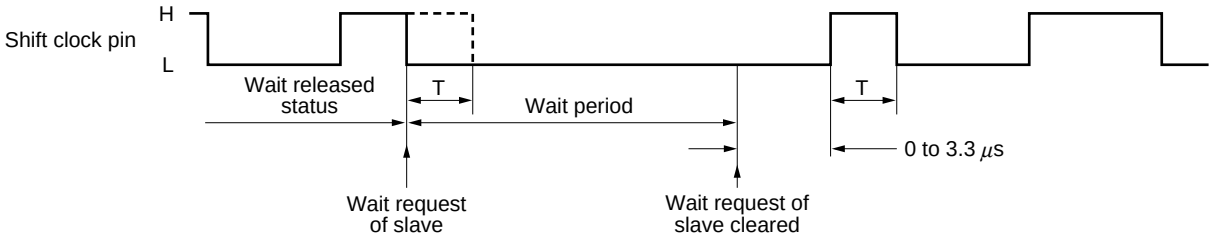
(e) If wait request is issued by slave during wait release

The clock is output 0 to 3.3 μs after the wait request of the slave is cleared.

- **When master outputs low level**



- **When master outputs high level**

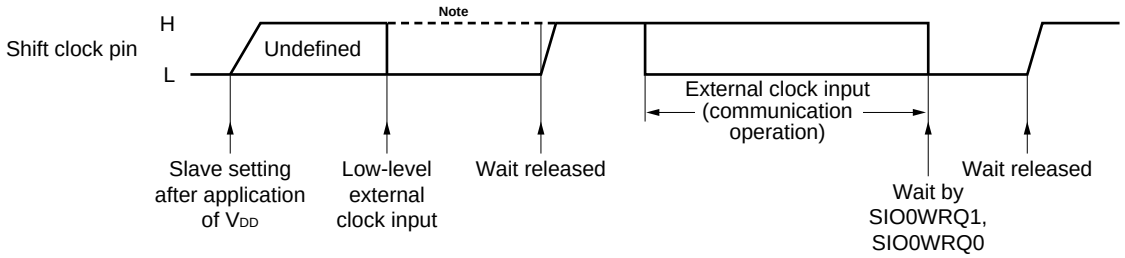


(3) Slave (external clock) operation

When the slave operation is selected for the first time after application of supply voltage V_{DD}, the output of the shift clock pin is undefined.

If a low-level external clock is input when the shift clock pin is off (the actual pin level is high because the pin is externally pulled up), the shift clock pin retains the low level until the wait status is released.

If a low-level external clock is not input, the shift clock pin retains the high level.

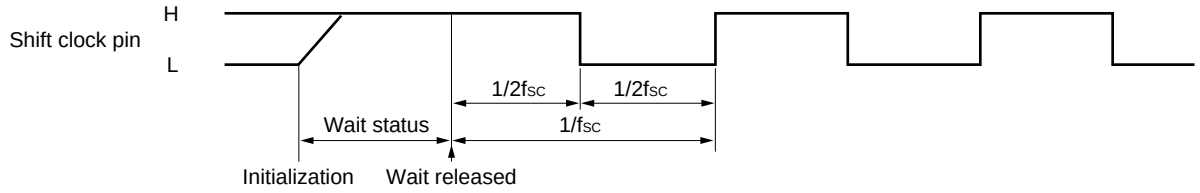


Note When a low-level external clock is not input

19.6.3 Shift clock generation timing in serial I/O mode

(1) When wait status is released from initial status

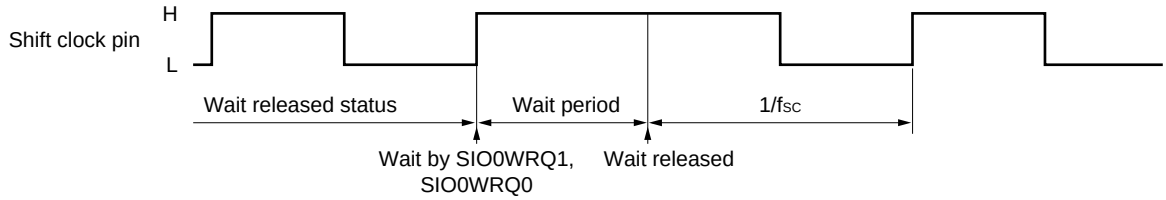
The "initial status" is the point at which the internal clock operation in the serial I/O mode has been selected. In the wait status, a high level is output to the shift clock pin ($P0A_2$/SCL pin in the two-line mode and $P0A_1$/$\overline{SCK_0}$ pin in the three-line mode).



(2) When wait operation is performed

For the details of the wait operation, refer to **19.9**.

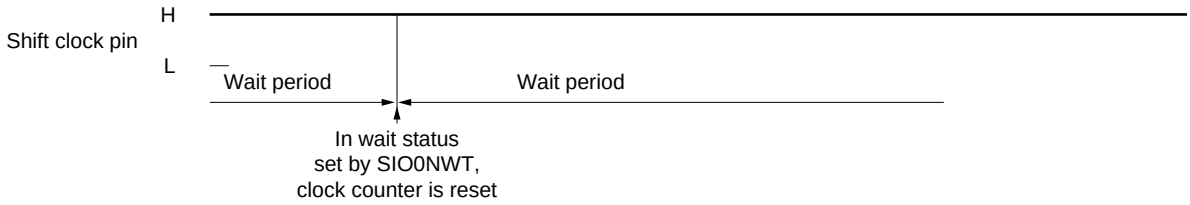
(a) When wait status is set under condition of SIO0WRQ0 and SIO0WRQ1 flags (normal operation)



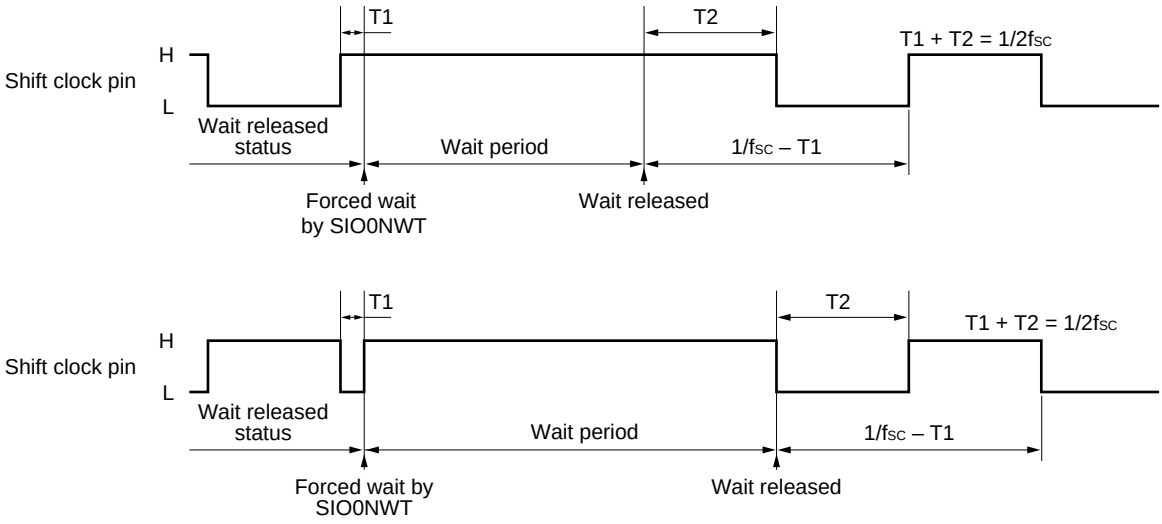
(b) When forced wait status is set during wait status

The shift clock pin retains the high level.

However, note that the clock counter is reset.



(c) When forced wait status is set during wait release



(d) If wait status is released during wait release

The clock output waveform is not changed.

However, note that the clock counter is reset.

19.7 Clock Counter and Start/Stop Detection Block

The clock counter is a wrap-around counter that counts the number of clocks input to the shift clock pin ($P0A_2$/SCL pin in the two-line mode and $P0A_1/\overline{SCK_0}$ pin in the three-line mode) selected at that time.

The clock counter directly reads the status of the shift clock pin. At this time, whether the clock is the internal clock or external clock is not judged.

The clock counter does not operate in the wait status of serial communication.

The contents of the clock counter can be detected via the SIO0SF8 and SIO0SF9 flags of the serial I/O0 status judge register, but cannot be directly read by program.

The following **19.7.1** through **19.7.4** describe the configuration and function of the serial I/O0 status judge register, the operation of the clock counter, and how the clock counter is reset.

The start/stop detection block detects the start/stop condition in the I^2C bus mode.

The start condition and stop condition can be detected by the SBSTT and SBBSY flags of the serial I/O0 status judge register.

19.7.5 describes the operations of the SBSTT and SBBSY flags.

19.7.1 Configuration and function of serial I/O0 status judge register (SIO0STUS)

The serial I/O0 status judge register detects the clock counter of serial interface 0 and the start/stop conditions in the I^2C bus mode.

The configuration and function of this register are illustrated below.

Name	Flag Symbol b_3	b_2	b_1	b_0	Address	Read/ Write
Serial I/O0 status judge register (SIO0STUS)	SIO0SF8	SIO0SF9	SBTT	SBBSY	28H	R

SIO0SF8 — Detects clock counter of serial interface 0

0	Reset to 0 when clock counter is "0" or "1"
1	Set to 1 when clock counter is "8"

SIO0SF9 — Detects clock counter of serial interface 0

0	Reset to 0 when clock counter is "0" or "1"
1	Set to 1 when clock counter is "9"

SBTT — Detects start condition in I^2C bus mode and clock counter

	I^2C bus mode	Serial I/O mode
0	Reset to 0 at falling edge of clock when clock counter is "9"	Retains previous value
1	Set to 1 when start condition is detected	

SBBSY — Detects start/stop conditions in I^2C bus mode

	I^2C bus mode	Serial I/O mode
0	Reset to 0 when stop condition is detected	Retains "0"
1	Set to 1 when start condition is detected	

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

19.7.2 Operation of clock counter in I²C bus mode

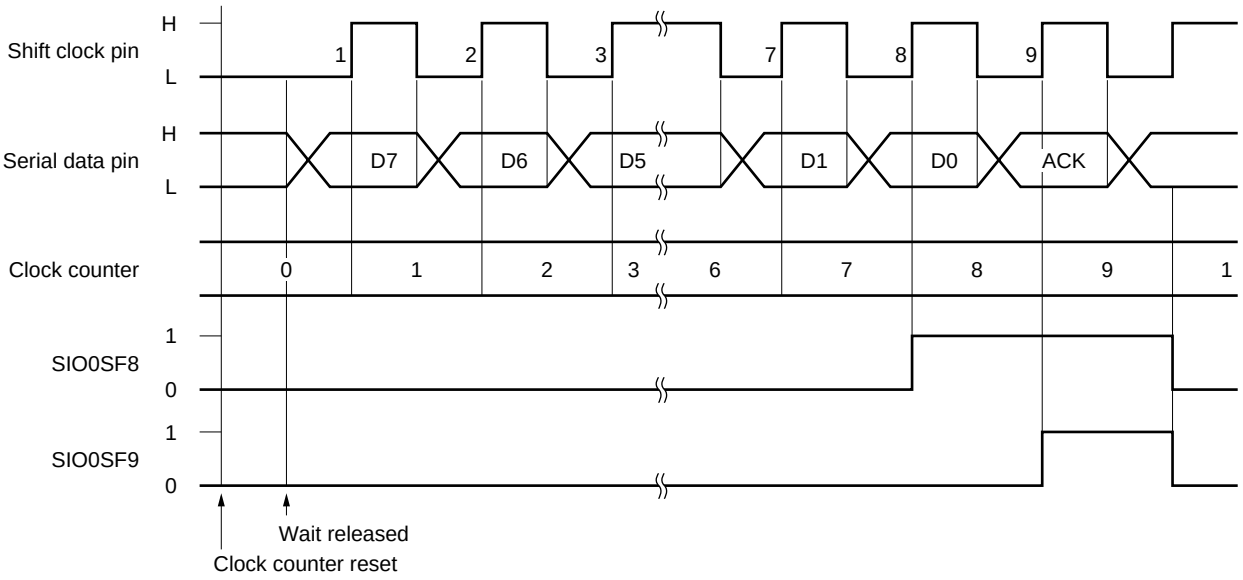
Figure 19-3 shows the operation of the clock counter.

The initial value of the clock counter is “0”. The clock counter is incremented each time the rising edge of the P0A2/SCL pin has been detected. After its value has been incremented to “9”, it is returned to “1” and the counter continues counting.

The SIO0SF8 and SIO0SF9 flags detect the status in which the value of the clock counter reaches “8” and “9”.

These flags operate regardless of the master (internal clock) or slave (external clock), or the reception or transmission operation.

Figure 19-3. Operation of Clock Counter in I²C Bus Mode



19.7.3 Operation of clock counter in serial I/O mode

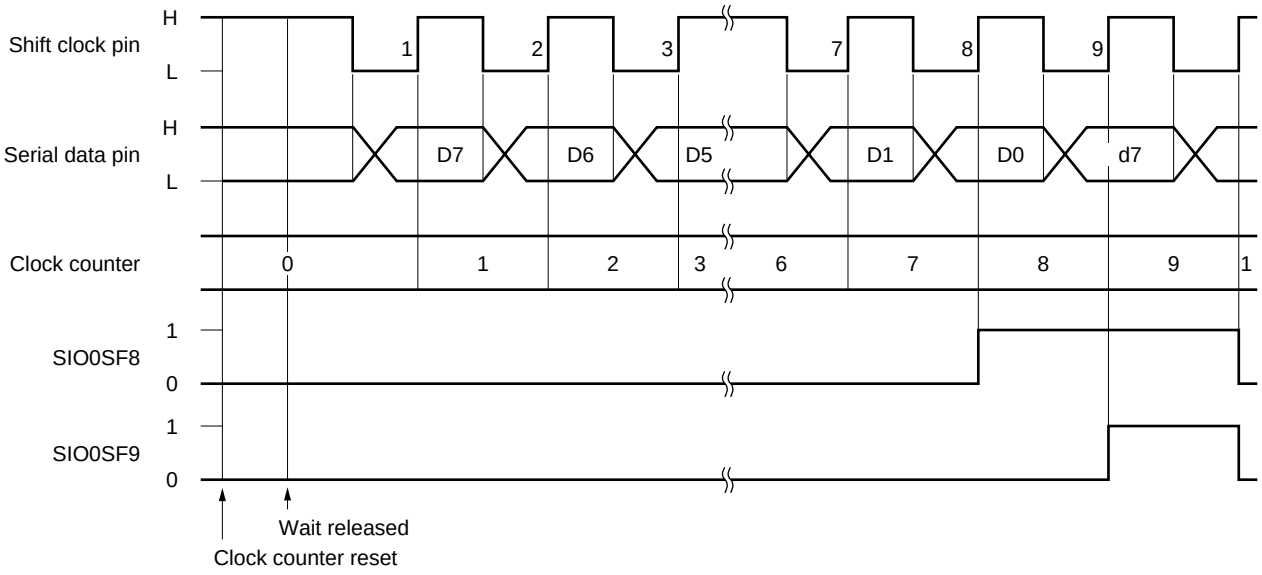
Figure 19-4 shows the operation of the clock counter.

The initial value of the clock counter is “0”. The clock counter is incremented each time the rising edge of the shift clock pin has been detected. After its value has been incremented to “9”, it is returned to “1” and the counter continues counting.

The SIO0SF8 and SIO0SF9 flags detect the status in which the value of the clock counter reaches “8” and “9”.

These flags operate regardless of the master or slave, or the reception or transmission operation.

Figure 19-4. Operation of Clock Counter in Serial I/O Mode



19.7.4 Reset (0) condition of clock counter

(1) In I^2C bus mode

(a) On power-ON reset
(b) On execution of clock stop instruction
(c) On detection of start condition
(d) If communication mode is changed from I^2C bus to 2- or 3-line serial I/O
(e) On CE reset

(2) In 2- or 3-line serial I/O mode

(a) On power-ON reset
(b) On execution of clock stop instruction
(c) When data is written to serial I/O0 wait control register
(d) If communication mode is changed from 2- or 3-line serial I/O to I^2C
(e) On CE reset

19.7.5 Operations of SBSTT and SBSSY flags

Figure 19-5 shows the operations of the SBSTT and SBBSY flags.

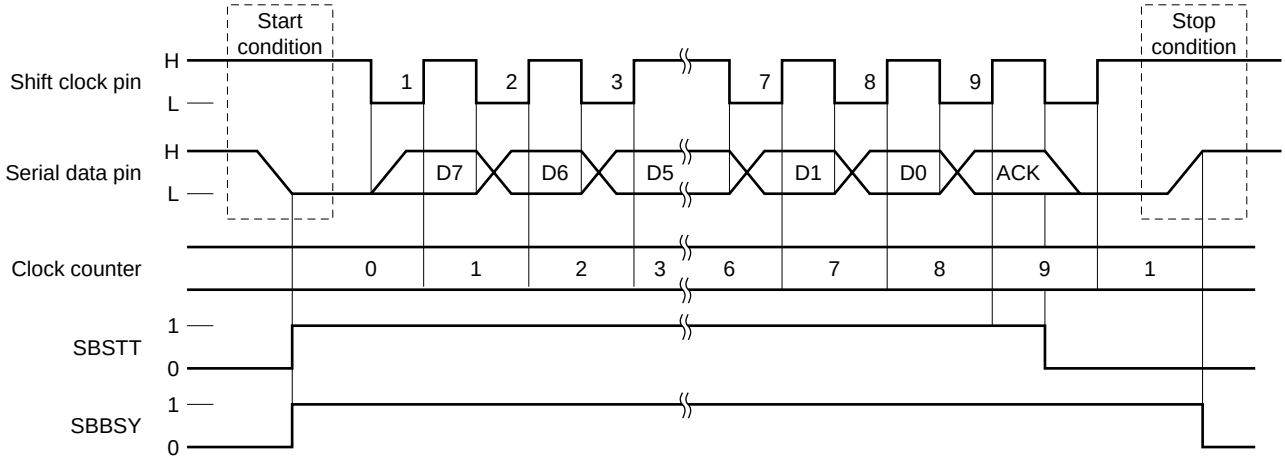
These flags operate only in the I^2C bus mode.

By detecting these flags, communication status of other stations can be detected.

These flags operate regardless of whether the master or slave, or reception or transmission operation is performed, or whether the wait status is set or released.

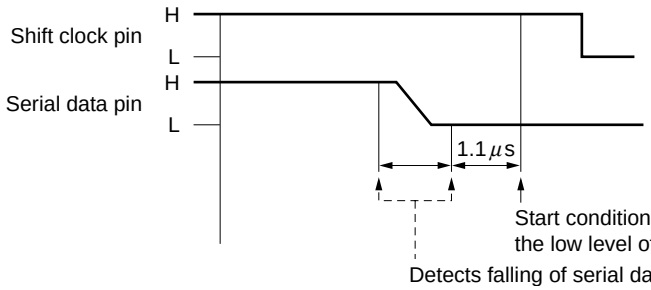
In the serial I/O mode, these flags retain “0”.

Figure 19-5. Operations of SBSTT and SBBSY Flags



The start and stop conditions are detected in the following timing (1) and (2).

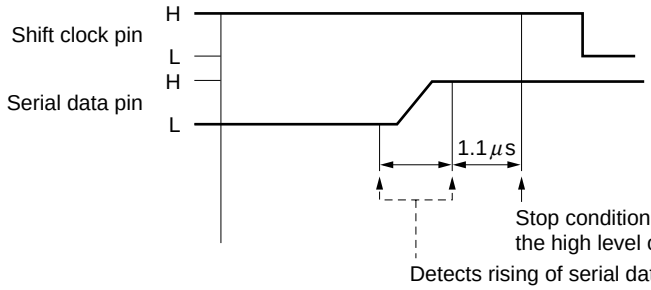
(1) Start condition detection timing



Start condition is assumed if the shift clock pin is high 1.1 μs after the low level of the serial data pin has been detected.

Detects falling of serial data pin

(2) Stop condition detection timing



Stop condition is assumed if the shift clock pin is high 1.1 μs after the high level of the serial data pin has been detected.

Detects rising of serial data pin

19.8 Presettable Shift Register 0 (SIO0SFR)

The presettable shift register 0 is an 8-bit shift register that writes serial out data and reads serial in data.

Data is written to or read from the presettable shift register 0 by the “PUT” or “GET” instruction via data buffer.

19.8.1 describes the configuration of the presettable shift register 0 and its relation with the data buffer.

The data of the presettable shift register 0 is shifted in synchronization with the clock applied to the shift clock pin ($P0A_2$/SCL pin in the two-line mode and $P0A_1/\overline{SCK_0}$ pin in the three-line mode) selected at that time.

In the I^2C bus mode, the most significant bit (MSB) of the presettable shift register 0 is output to the serial data pin ($P0A_3$/SDA pin) in synchronization with the falling edge of the shift clock, and the data of the serial data pin is read to the least significant bit (LSB) of the presettable shift register 0 in synchronization with the rising edge of the clock.

19.8.2 and **19.8.3** below describe the operation of the presettable shift register 0 in the I^2C bus mode and serial I/O mode, and the points to be noted.

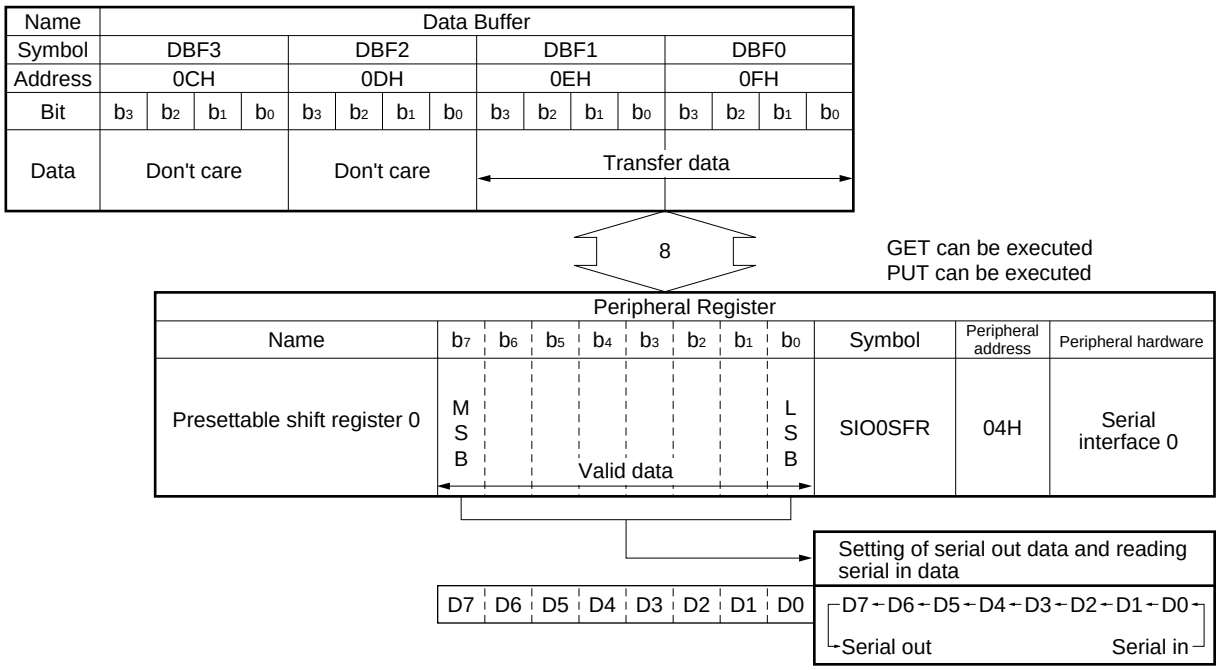
19.8.4 describes the points to be noted in writing or read data to or from the presettable shift register 0.

The presettable shift register 0 does not shift data in the wait status.

For the details of the operations of the register in the respective serial communication modes, refer to **19.11**.

19.8.1 Configuration of presettable shift register 0 and its relation with data buffer

The configuration of the presettable shift register 0 and its relation with the data buffer are illustrated below.



19.8.2 Operation of presettable shift register in I²C bus mode

Figure 19-6 shows the data shift operation in the I²C bus mode.

Table 19-3 shows the data shift operation during reception or transmission in the I²C bus mode.

Figure 19-6. Data Shift Operation in I²C Bus Mode

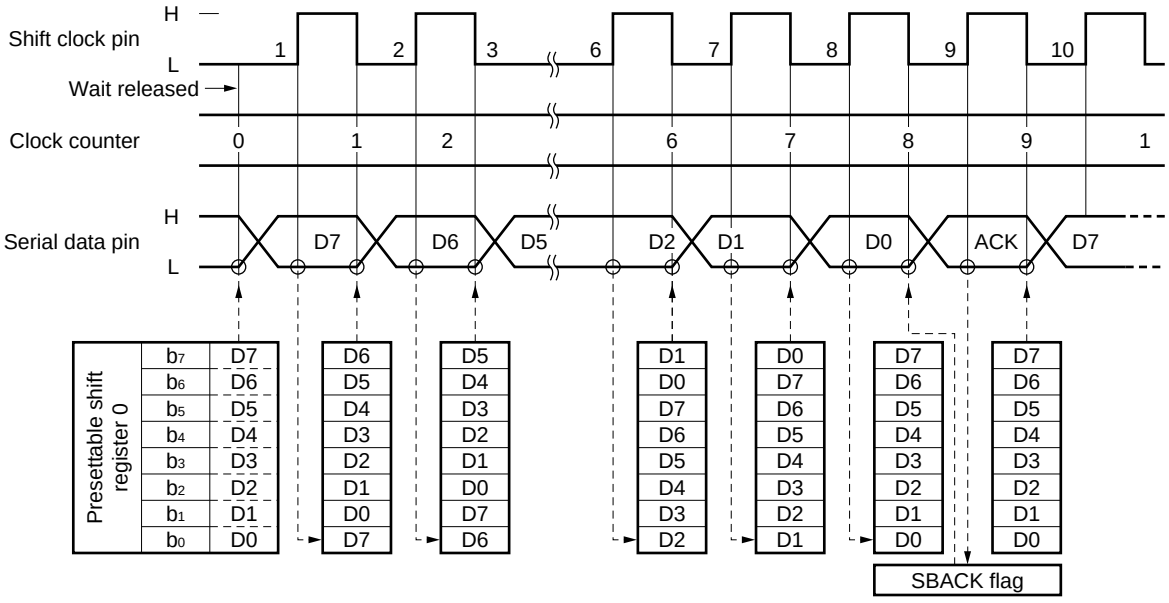


Table 19-3. Data Shift Operation during Reception and Transmission

I²C Bus Mode	
Reception	Transmission
Status of P0A3/SDA pin is input shifted from LSB at rising edge of P0A2/SCL pin. No output is produced. Content of SBACK flag is output at falling edge of shift clock when clock counter is “8”. Does not operate in wait status.	Data is shifted from MSB and output to P0A3/SDA pin at falling edge of P0A2/SCL pin. Status of P0A3/SDA pin is input from LSB at rising edge of P0A2/SCL pin. Status of P0A3/SDA pin is read to SBACK flag at rising edge of shift clock when clock counter reaches “9”. Does not operate in wait status.

19.8.3 Operation in serial I/O mode

Figure 19-7 shows the data shift operation in the serial I/O mode.

Table 19-4 shows the data shift operation during reception or transmission in the serial I/O mode.

Figure 19-7. Data Shift Operation in Serial I/O Mode

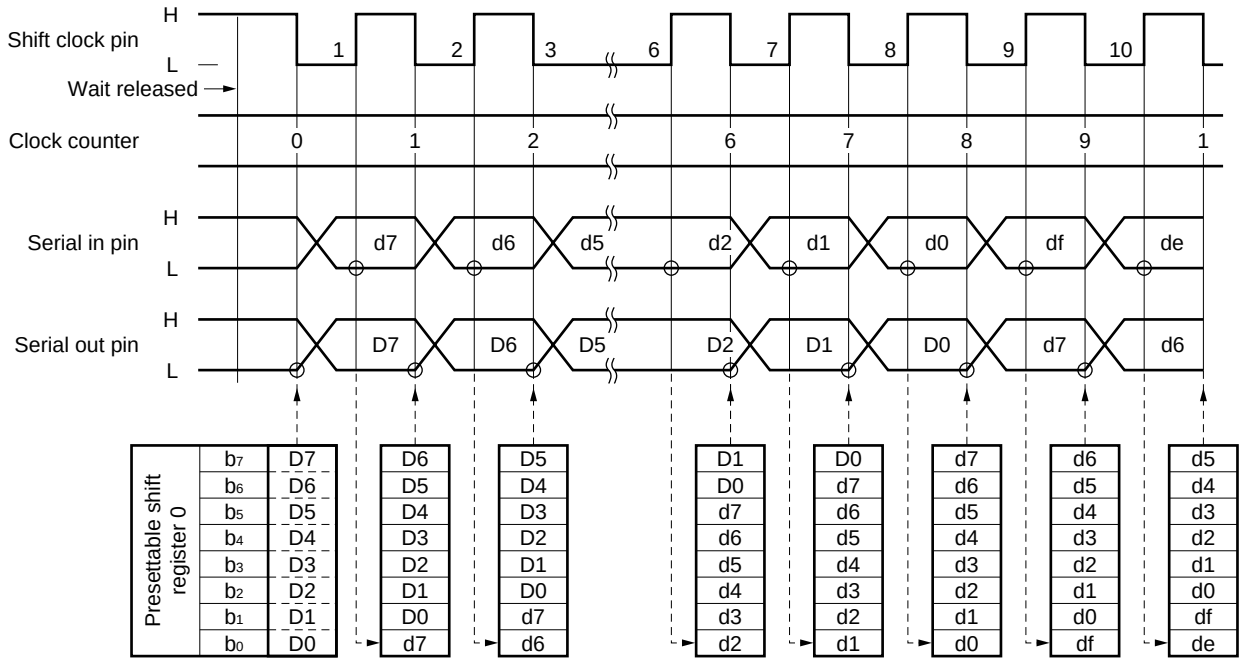


Table 19-4. Data Shift Operation during Reception or Transmission

Serial I/O Mode	
Reception	Transmission
Status of P0A3/SDA pin (P0B3/SI0 pin in 3-line mode) is shifted from LSB and input at rising edge of shift clock pin. No output is produced. Does not operate in wait status.	Data is shifted from MSB and output to P0A3/SDA pin (P0A0/SO0 pin in 3-line mode) at falling edge of shift clock. Status of P0A3/SDA (P0B3/SI0 in 3-line mode) pin is input from LSB at rising edge of shift clock. Does not operate in wait status.

19.8.4 Notes on setting and reading data

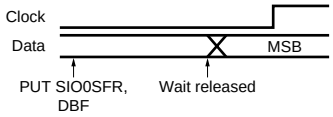
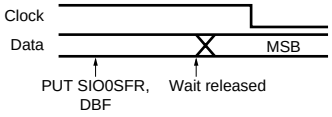
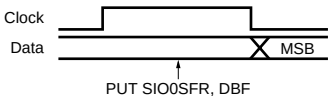
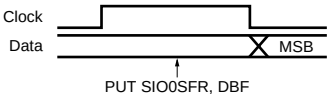
To set data to the presettable shift register 0, use the "PUT SIO0SFR, DBF" instruction.

To read data, use the "GET DBF, SIO0SFR" instruction.

Set or read data in the wait status. While the wait status is released, data may not be correctly set or read depending on the status of the shift clock pin.

Table 19-5 shows the timing of setting and reading data, and points to be noted.

Table 19-5. Reading (GET) and Writing (PUT) Data of Presettable Shift Register 0 and Notes

<table>
<tr><th colspan="2">Status on Execution of PUT/GET</th><th>Status of Shift Clock Pin</th><th>I²C Bus Mode</th><th>Serial I/O mode</th></tr>
<tr><td rowspan="2">Wait status</td><td>Read (GET)</td><td>I²C bus mode
Fixed to low</td><td>Normal read</td><td>Normal read</td></tr>
<tr><td>Write (PUT)</td><td>Serial I/O mode
Fixed to high</td><td>Normal write
Outputs MSB contents as data when wait status is released next time (during transmission)
 
</td><td>Normal write
Outputs MSB contents as data when wait status is released next time (during transmission)
 
</td></tr>
<tr><td rowspan="4">Wait released status</td><td rowspan="2">Read (GET)</td><td>When low</td><td>Normal read</td><td>Normal read</td></tr>
<tr><td>When high</td><td>Normal read
(Set value is shifted 1 bit (MSB is shifted to LSB) and is read when internal clock is used)</td><td>Normal read
(Set value is shifted 1 bit (MSB is shifted to LSB) and is read when internal clock is used)</td></tr>
<tr><td rowspan="2">Write (PUT)</td><td>When high</td><td>Normal write
Outputs MSB contents at falling edge of shift clock. Clock counter is not reset.
 
</td><td>Normal write
Outputs MSB contents at falling edge of shift clock. Clock counter is not reset.
 
</td></tr>
<tr><td>When low</td><td>Cannot be written normally.
Contents of SIO0SFR are lost</td><td>Cannot be written normally.
Contents of SIO0SFR are lost</td></tr>
</table>

19.9 Wait Block and Acknowledge

The wait block places communication of serial interface 0 in the wait status or releases the wait status.

The acknowledge block outputs and detects an acknowledge signal in the I²C bus mode.

The wait block and acknowledge block are controlled by the serial I/O0 wait control register (refer to **19.9.1**). The wait status is detected by the serial I/O0 wait status judge register (refer to **19.9.2**).

19.9.3 through **19.9.5** describe the outline of the wait operation, wait operations in the respective communication modes, and points to be noted, and **19.9.6** describes the acknowledge block.

19.9.1 Configuration and function of serial I/O0 wait control register (SIO0WT)

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Serial I/O0 wait control register (SIO0WT)	SBACK	SIO0NWT	SIO0WRQ1	SIO0WRQ0	18H	R/W

Sets wait condition

SIO0WRQ1	SIO0WRQ0	Name	I^2C bus mode	Serial I/O mode
0	0	No wait	Setting prohibited	No wait
0	1	Data wait	Wait at falling edge of shift clock when clock counter is "8"	Wait at rising edge of shift clock when clock counter is "8"
1	0	Acknowl-edge wait	Wait at falling edge of shift clock when clock counter is "9"	Wait at rising edge of shift clock when clock counter is "9"
1	1	Address wait	Wait at falling edge of shift clock when clock counter is "8" after start condition has been detected	Setting prohibited

Sets wait and detects wait status

SIO0NWT	When writing to flag	When flag is read
0	Forced wait	Wait in progress under condition of SIO0WRQ1 and 0 flags
1	Releases wait (serial communication start)	Serial communication in progress

Sets and detects acknowledge signal in I^2C bus mode

SBACK	I^2C bus mode		Serial I/O mode
	Reception (SIO0TX = 0)	Transmission (SIO0TX = 1)	
0	Outputs "0" as acknowledge	Detects acknowledge of slave. Acknowledge is "0"	Does not operate. Therefore, can be used as 1-bit flag
1	Outputs "1" as acknowledge	Detects acknowledge of slave. Acknowledge is "1"	

		SBACK	SIO0NWT	SIO0WRQ1	SIO0WRQ0
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

19.9.2 Configuration and function of serial I/O0 wait status judge register (SIOWSTR)

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Serial I/O0 wait status judge register (SIO0WSTR)	0	0	0	SIO0WSTT	19H	R/W

b_0	Detects wait status of serial interface 0
0	• Wait under condition of SIO0WRQ1 and 0 flags • Wait by slave (master mode)
1	Serial communication in progress

Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop				0
	CE				0

19.9.3 Outline of wait operation

In the wait status, the clock generation block and presettable shift register 0 stop operation, and therefore, serial communication stops.

Serial communication can be executed by releasing the wait status.

To release the wait status, write “1” to the SIO0NWT flag.

When “1” is written to the SIO0NWT flag, the internal clock is output to the shift clock output pin (when the device is operating as the master), and the presettable shift register 0 and clock counter start operating.

If the condition set by the SIO0WRQ0 and SIO0WRQ1 flags is satisfied, the wait status is set. At this time, the SIO0NWT flag is automatically reset to 0.

By detecting the content of the SIO0NWT flag when the wait status has been released, the operation status of serial communication can be checked.

Therefore, by writing “1” to the SIO0NWT flag and then detecting “0” of the SIO0NWT flag after serial communication has been started, data is read or set.

Note that there is a time lag since the SIO0NWT flag has been cleared to “0” until the wait status is actually set.

If data is set to the presettable shift register 0 (by using the PUT instruction) or data is read (by using the GET instruction) while the wait status is released, the correct data may not be set or read. For details, refer to **19.8**.

If “0” is written to the SIO0NWT flag while the wait status is released, the wait status is set. This is called “forced wait status”.

Note that there is a time lag for the forced wait status in the I^2C bus mode since “0” has been written to the SIO0NWT flag until the wait status is actually set.

If “0” is written to the SIO0NWT flag when the device operates as the master in the I^2C bus mode, one pulse of the shift clock is output. Note that the clock counter and presettable shift register 0 stop operating at this time.

If “1” is written to the SIO0NWT flag in the serial I/O mode, the clock counter is reset to 0.

The wait status is detected by reading the contents of the SIO0WSTT flag.

The SIO0WSTT flag is set to “1” also when the shift clock pin of the slave outputs a low level while the device is operating as the master.

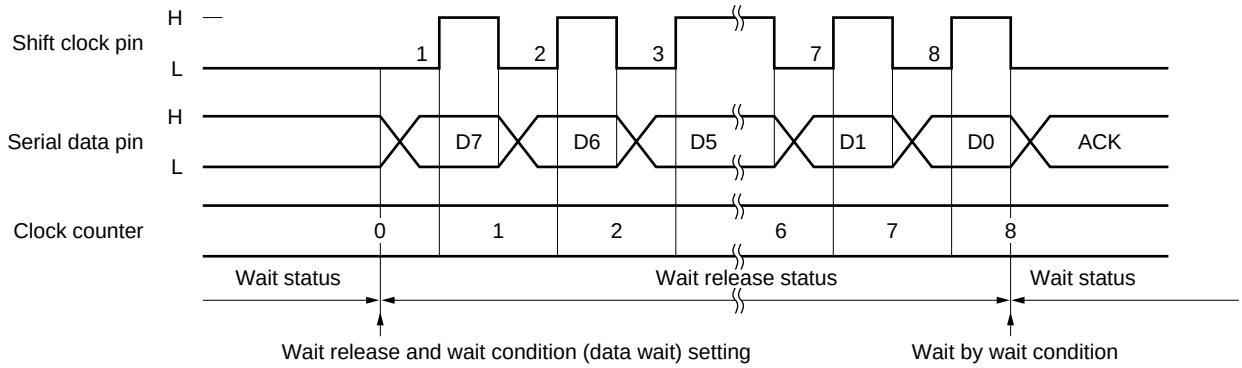
Because the SIO0WSTT flag is a read-only flag, it cannot be used to set or release the wait status, unlike the SIO0NWT flag.

19.9.4 Wait operation and notes in I²C bus mode

(1) Wait operation in I²C bus mode

Figure 19-8 shows an example of the data wait (SIO0WRQ1 = 0, SIO0WRQ0 = 1) operation in the I²C bus mode.

Figure 19-8. Data Wait Operation in I²C Bus Mode



When the wait status is released, serial data is output (during transmission), and the wait status remains released until the condition set by the SIO0WRQ1 and SIO0WRQ0 flags is satisfied.

When the wait condition is satisfied, the serial clock pin is made low, and the clock counter and presettable shift register 0 stop operation.

If data is written to the presettable shift register 0 while the wait status is released and the shift clock pin is low, the correct data may not be set.

If data is written to the presettable shift register 0 while the wait is released and the shift clock pin is high, the content of the MSB is output to the serial data output pin at the falling edge of the shift clock next to the one at which the “PUT” instruction was executed.

If the forced wait status is set while the wait status is released, the wait status is set at the falling edge of the clock next to the one at which “0” was written to the SIO0NWT flag.

Nothing is changed even if the forced wait status is released while the wait status is released.

If the forced wait status is set in the wait status, one pulse of the shift clock is output.

Do not set the data wait condition (SIO0WRQ1 = 0, SIO0WRQ0 = 1) successively in the I²C bus mode. If the data wait condition is set two times in succession and then the wait status is released, the wait status is set immediately when the wait status is released the second time.

Therefore, a different wait condition must be set after the wait status of the data wait condition.

When the I²C bus mode is used, there is a period in which the status of the SIO0NWT flag and the actual communication operation differ as described in (2) and (3) below.

(2) Normal wait operation in I²C bus mode

In the I²C bus mode, communication is placed in the wait status at the falling edge of the shift clock if the wait condition set by the SIO0WRQ1 and SIO0WRQ0 flags is satisfied.

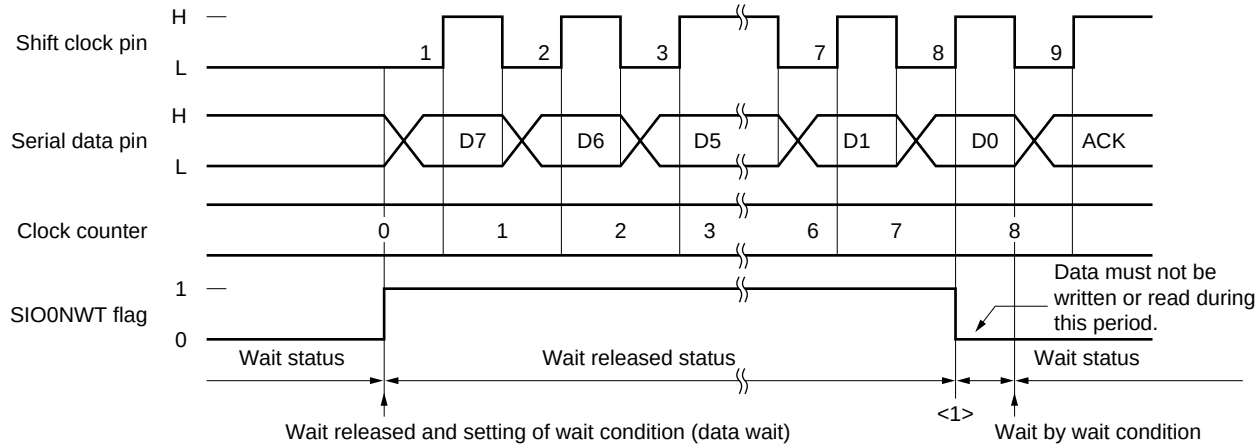
Figure 19-9 shows the data wait operation and SIO0NWT flag operation in the I²C bus mode.

As shown in <1> in this figure, the SIO0NWT flag is reset to "0" at the rising edge half the clock before the wait status is set.

Therefore, if data is written (PUT) or read (GET) immediately after the SIO0NWT flag has been cleared to "0", the data may be lost.

Consequently, write or read data after the low level of the shift clock pin has been detected after "0" was read from the SIO0NWT flag.

Figure 19-9. Data Wait and SIO0NWT Flag Operations in I²C Bus Mode



(3) Forced wait operation in I²C bus mode

If "0" is written to the SIO0NWT flag while the wait status is released in the I²C bus mode, the forced wait status is set at the falling edge of the next clock.

Therefore, data must be written or read after the negative transition of the shift clock pin (from high to low) has been detected in the same manner as (1) above.

If the forced wait status is set in the wait status while the device is operating as the master and receiving data, one pulse of the shift clock is output.

This must be noted when setting the acknowledge signal described in **19.9.6**.

(4) Wait request by slave

If the shift clock pin is forcibly made low by an external source (this is called wait request by a slave) while the pin is outputting a high level and while the device is operating as the master, the SIO0NWT flag is reset to "0".

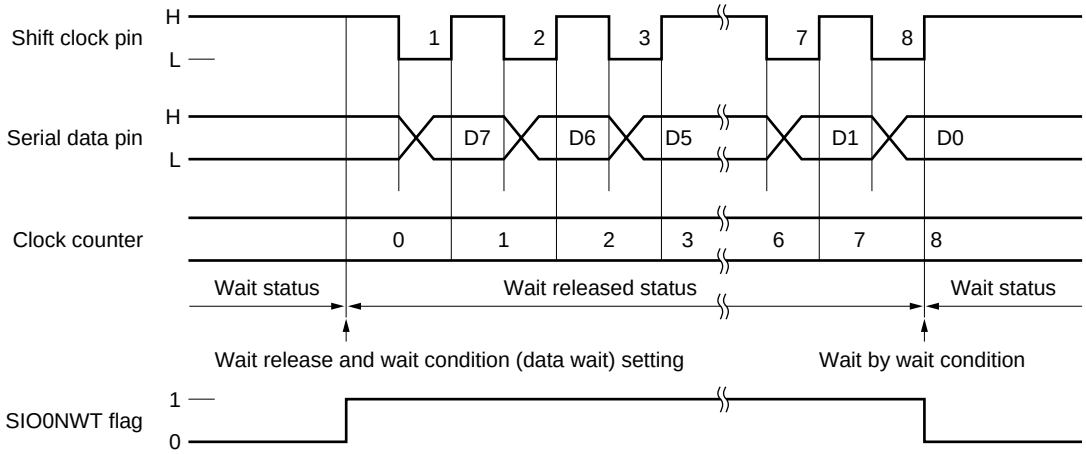
At this time, the SIO0NWT flag is set to 1 when the wait request by the slave has been released, and the device continues operation.

19.9.5 Wait operation and note in serial I/O mode

(1) Wait operation in serial I/O mode

Figure 19-10 shows an example of the data wait (SIO0WRQ1 = 0, SIO0WRQ0 = 1) operation in the serial I/O mode.

Figure 19-10. Data Wait Operation in Serial I/O Mode



When the wait status is released, serial data is output at the next falling edge of the clock (during transmission operation), and the wait status is released until the condition set by the SIO0WRQ1 and SIO0WRQ0 flags is satisfied.

When the wait condition is satisfied, the shift clock pin is made high, and the operations of the clock counter and presettable shift register 0 are stopped.

If data is written to the presettable shift register 0 while the wait status is released and the shift clock pin is low, the correct data may not be set.

If data is written to the presettable shift register 0 while the wait status is released and the shift clock pin is high, the content of the MSB is output to the serial data output pin at the next falling edge of the shift clock after the “PUT” instruction was executed.

If the forced wait status is set in the wait status, the wait status is set immediately when “0” has been written to the SIO0NWT flag.

If the wait status is released again while the wait status is released, the clock counter may be reset.

19.9.6 Acknowledge block and its operation

The acknowledge block operates only in the I²C bus mode.

This block is used to output an acknowledge signal during the reception operation in the I²C bus mode, and to detect the acknowledge signal during transmission operation.

During reception, the content of the SBACK flag is output to the serial data pin at the falling edge of the shift clock when the current value of the clock counter is “8” (the serial data pin automatically enters the output port).

Once data has been set to the SBACK flag during reception, the value of the data is retained.

During transmission, the status of the serial data pin is read to the SBACK flag at the rising edge of the shift clock when the current value of the clock counter is “9” (the serial data pin automatically enters the input port).

Figure 19-11 illustrates the acknowledge signal output and input operations.

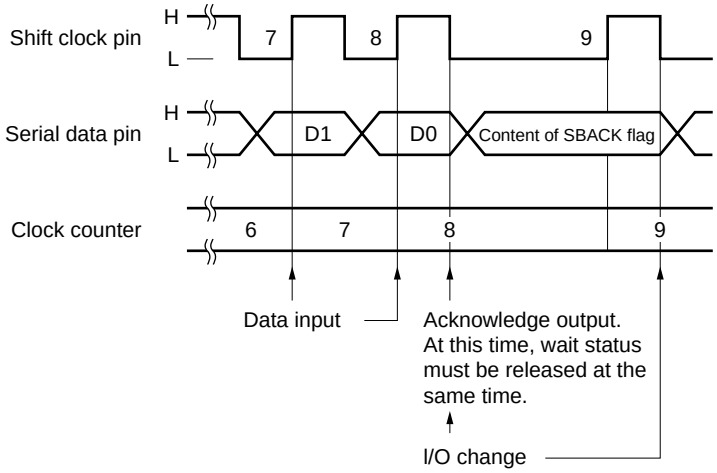
Set the acknowledge signal (setting of the SBACK flag) during reception as soon as the wait status has been released (setting of SIO0NWT flag).

This is because, even if an attempt is made to set the SBACK flag alone, the SIO0NWT flag is also set because the SIO0NWT flag is in the register at the same address. If the wait status is set at this time, the wait status is released and one pulse of the shift clock is output.

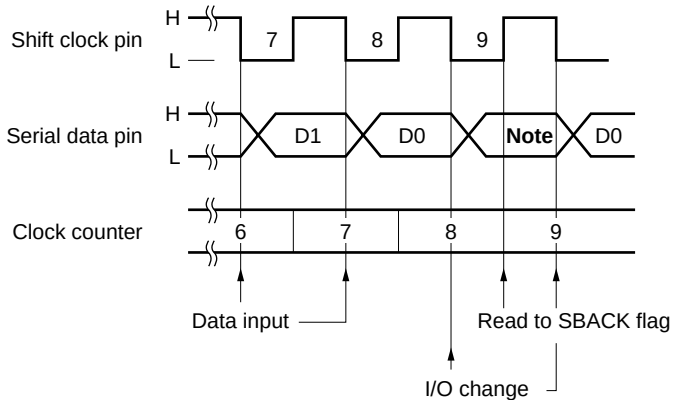
In the serial I/O mode, the SBACK flag can be used as 1-bit general-purpose flag.

Figure 19-11. Acknowledge Signal Output and Input Operations

(1) Output operation (during reception)



(2) Input operation (during transmission)



Note Acknowledge signal from reception side

Caution When the acknowledge signal is output or input, be sure to set the acknowledge wait status at the falling edge of the eighth clock.

19.10 Interrupt Control Block

The interrupt control block issues the interrupt request of serial interface 0 and sets the condition under which the interrupt request is to be issued by using the serial I/O0 interrupt mode register.

When the interrupt request issuance condition is satisfied, the IRQSIO0 flag of the serial I/O0 interrupt request register (IREQSIO0: RF address 3BH) is set to 1.

The following **19.10.1** describes the configuration and function of the serial I/O0 interrupt mode register.

19.10.2 and **19.10.3** indicate the interrupt request issuance timing in the respective communication modes.

19.10.1 Configuration and function of serial interface 0 interrupt mode register (SIO0INT)

The functions of the respective flags of the serial interface 0 interrupt mode register is shown below.

Do not change the contents of these flags during serial communication (when the SIO0NWT flag is “1”).

Change these flags after “0” has been written to the SIO0NWT flag or when the SIO0NWT flag is “0”. If the contents of these flags are changed during serial communication, an interrupt request may be issued as soon as the flag contents have been changed.

Name	Flag Symbol b_3	b_2	b_1	b_0	Address	Read/ Write
Serial I/O0 interrupt mode register (SIO0INT)	SIO0IMD3	SIO0IMD2	SIO0IMD1	SIO0IMD0	38H	R/W

Sets interrupt request issuance condition

SIO0IMD1	SIO0IMD0	I^2C bus mode	Serial I/O mode
0	0	Rising of shift clock when clock counter reaches "7"	Rising of shift clock when clock counter reaches "7"[Note1]
0	1	Rising of shift clock when clock counter reaches "8"	Rising of shift clock when clock counter reaches "8"[Note2]
1	0	Rising of shift clock when clock counter reaches "7" after start condition has been detected[Note3]	Interrupt request is not issued
1	1	When stop condition is detected[Note4]	Interrupt request is not issued

Fixed to "0"

		SIO0IMD1	SIO0IMD0	
On reset	Power-ON	0	0	Undefined
	Clock stop			Retained
	CE			Retained

Notes 1. If this mode is set when the current value of the clock counter is “7”, the interrupt request is issued.

2. If this mode is set when the current value of the clock counter is “8”, the interrupt request is issued.

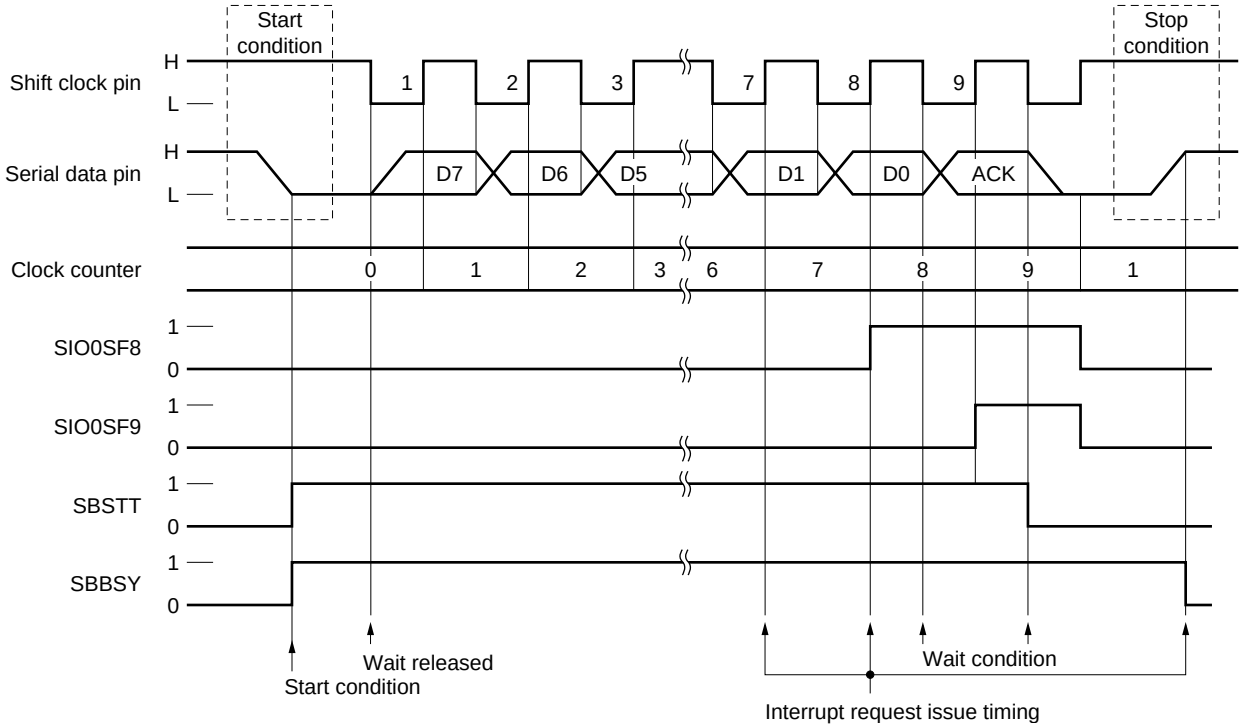
3. If this mode is selected when the SBSTT flag is “1” and the current value of the clock counter is “7”, the interrupt request is issued.

4. When this mode is selected after the stop condition has been detected, the interrupt request is issued.

19.10.2 Interrupt request issuance timing in I²C bus mode

Figure 19-12 shows the interrupt request issuance timing in the I²C bus mode.

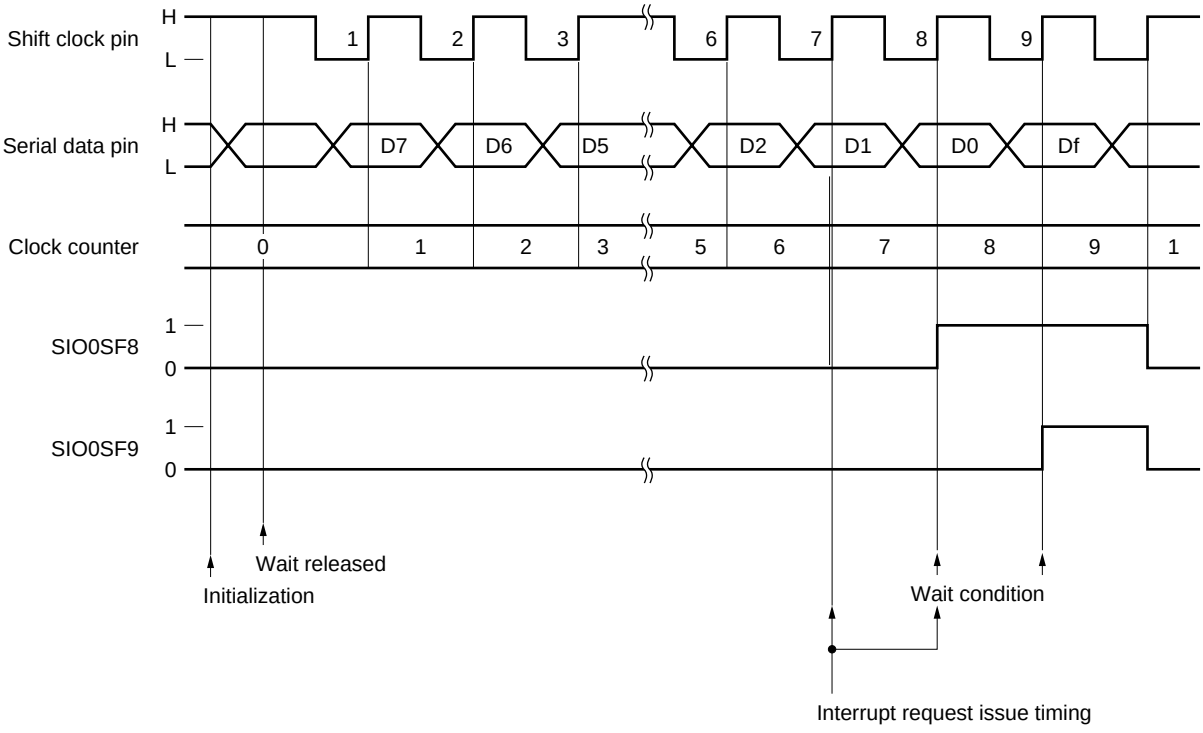
Figure 19-12. Interrupt Request Issuance timing in I²C Bus Mode



19.10.3 Interrupt request issuance timing in serial I/O mode

Figure 19-13 shows the interrupt request issuance timing in the serial I/O mode.

Figure 19-13. Interrupt Request Issuance Timing in Serial I/O Mode



19.11 Using Serial Interface 0

19.11.1 Using I²C bus mode

The I²C bus mode is selected by resetting the SIO0CH flag to “0” and setting the SB flag to “1”.

In this mode, the P0A3/SDA and P0A2/SCL pins are used.

Figure 19-14 shows the I/O block and communication method in the I²C bus mode.

Table 19-6 shows the pins used in the I²C bus mode and the function and operation of the control register.

As shown in Figure 19-14 and Table 19-6, a master or slave operation can be performed in the I²C bus mode. Data can be transmitted (TX) or received (RX) during master and slave operations.

The master or slave operation is selected by the SIO0MS flag, and the reception or transmission is selected by the SIO0TX flag.

During the master operation, the internal shift clock is output from the P0A2/SCL pin. If transmission is carried out at this time, data is output from the P0A3/SDA pin at the falling edge of the shift clock. During reception, the status of the P0A3/SDA pin is input to the presettable shift register 0 at the rising edge of the shift clock.

During master or slave operation, the start and stop conditions of serial communication can be detected by the SBSTT and SBBSY flags.

The start and stop conditions are usually output by the master. This output is made by program (by controlling each pin as a general-purpose output port pin).

During the slave operation, the P0A2/SCL pin is floated and the device waits for an external clock. If transmission is performed at this time, data is output from the P0A3/SDA pin at the falling edge of the shift clock. If reception is performed, the status of the P0A3/SDA pin is input to the presettable shift register 0 at the rising edge of the clock applied to the P0A2/SCL pin.

During reception by the master or slave, an acknowledge signal is output each time 8-bit data has been communicated.

During transmission by the master or slave, an acknowledge signal is detected each time 8-bit data has been communicated.

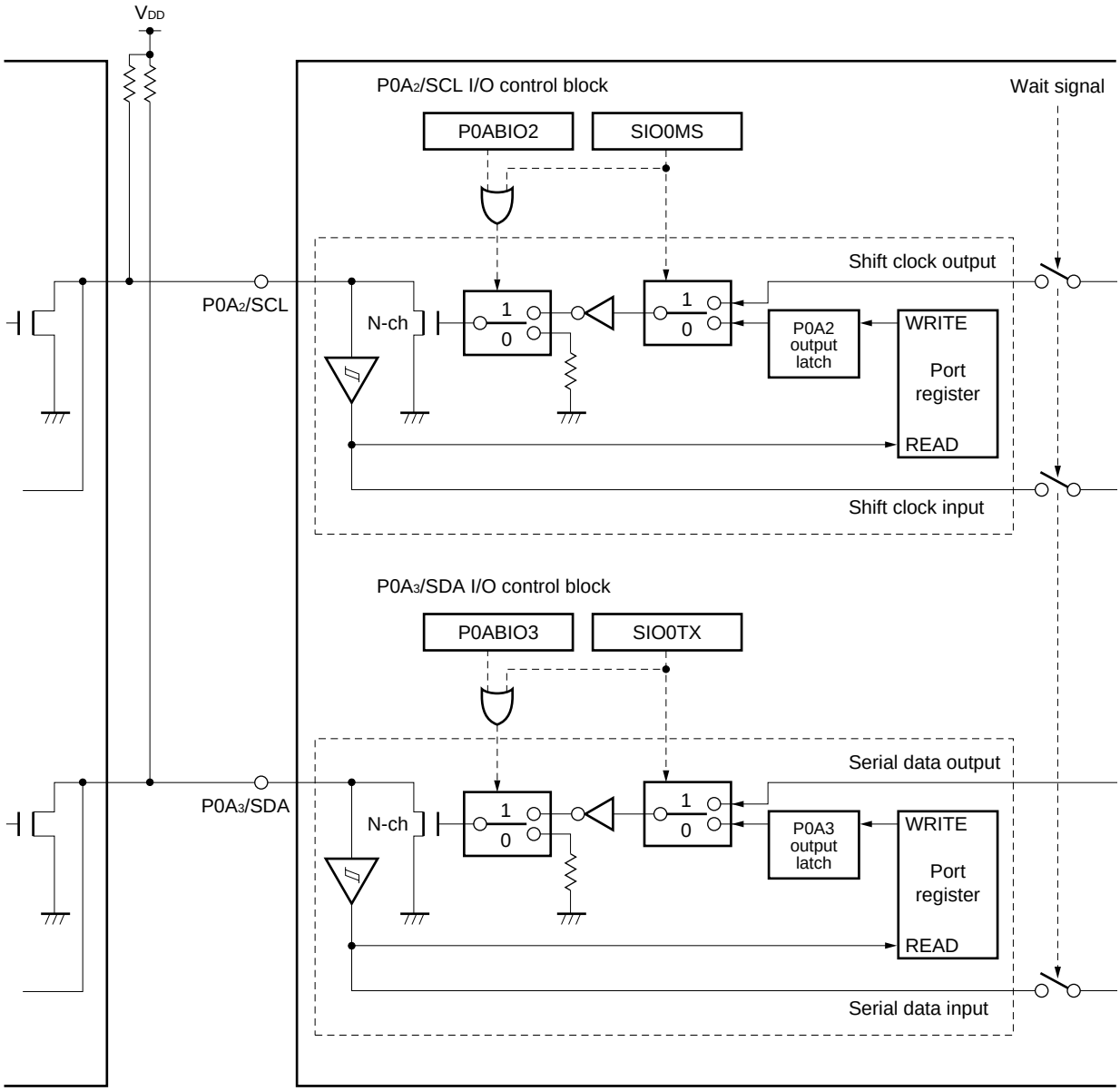
The P0A3/SDA and P0A2/SCL pins are N-ch open-drain output pins; therefore, the communication line goes low if either the master or slave outputs a low level.

When the values output to the P0A3/SDA and P0A2/SCL pins are read, the “status of pin at that time” is read.

Paragraphs (1) through (4) below Table 19-6 show program examples for transmission and reception during master and slave operations.

Figure 19-14. I/O Block and Communication Method in I²C Bus Mode

I/O block



Communication method

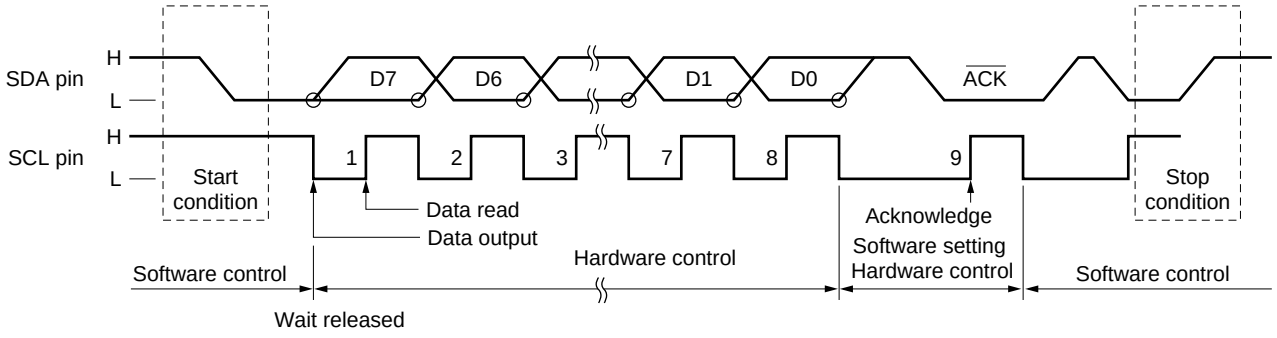


Table 19-6. Outline of Operation in I²C Bus Mode

Item \ Operation Mode		I²C Bus Mode SIO0CH=0, SB=1			
		Slave operation SIO0MS=0		Master operation SIO0MS=1	
		Reception (RX) SIO0TX=0	Transmission (TX) SIO0TX=1	Reception (RX) SIO0TX=0	Transmission (TX) SIO0TX=1
Setting status of each pin	$P0A_3$/SDA	**When P0ABIO3 = 0** Floating External data input wait **When P0ABIO3 = 1** General-purpose output port Outputs contents of output latch. Normally, P0ABIO3 is reset to 0.	Outputs contents of SIO0SFR at falling edge of external clock regardless of P0ABIO3	**When P0ABIO3 = 0** Floating External data input wait **When P0ABIO3 = 1** General-purpose output port Outputs contents of output latch. Normally, P0ABIO3 is reset to 0.	Outputs contents of SIO0SFR at falling edge of internal shift clock regardless of P0ABIO3
	$P0A_2$/SCL	**When P0ABIO2 = 0** Floating External clock input wait **When P0ABIO2 = 1** General-purpose output port Outputs contents of output latch. Normally, P0ABIO2 is reset to 0.	**When P0ABIO2 = 0** Floating External clock input wait **When P0ABIO2 = 1** General-purpose output port Outputs contents of output latch. Normally, P0ABIO2 is reset to 0.	Outputs internal shift clock regardless of P0ABIO2	Outputs internal shift clock regardless of P0ABIO2
Clock counter operation		Incremented at rising edge of SCL pin			
Operation of presettable shift register 0 (SIO0SFR)		**Output** Not output **Input** Shifts data of SDA pin from LSB and inputs it each time SCL pin rises	**Output** Shifts data from MSB and outputs it to SDA each time SCL pin falls **Input** Shifts data of SDA pin from LSB and inputs it each time SCL pin rises	**Output** Not output **Input** Shifts data of SDA pin from LSB and inputs it each time SCL pin rises	**Output** Shifts data from MSB and outputs it to SDA each time SCL pin falls **Input** Shifts data of SDA pin from LSB and inputs it each time SCL pin rises
Wait operation		Serial communication is started when "1" is written to SIO0NWT. SIO0NWT is reset to "0" under condition set by SIO0WRQ1 and SIO0WRQ0			
		When SIO0NWT = 0 Forcibly outputs low level from SCL pin. SDA pin is floated. **When SIO0NWT = 1** Floats SCL pin and waits for external clock input. SDA pin is floated and data of SDA pin is input to SIO0SFR at rising edge of SCL pin.	**When SIO0NWT = 0** Forcibly outputs low level from SCL pin. SDA pin retains its status. **When SIO0NWT = 1** Floats SCL pin and waits for external clock input. Outputs contents of SIO0SFR to SDA pin at falling edge of SCL pin.	**When SIO0NWT = 0** Forcibly outputs low level from SCL pin. SDA pin is floated. **When SIO0NWT = 1** Outputs internal shift clock from SCL pin. SDA pin is floated and data of SDA pin is input to SIO0SFR at rising edge of SCL pin.	**When SIO0NWT = 0** Forcibly outputs low level from SCL pin. SDA pin retains its status. **When SIO0NWT = 1** Outputs internal shift clock from SCK pin. Outputs ccontents of SIO0SFR to SDA pin at falling edge of SCL pin.
Acknowledge		Outputs contents of SBACK flag are from SDA pin at falling edge of SCL pin when clock counter is 8	Status of SDA pin is written to SBCK flag at rising edge of SCL pin when clock counter reaches 9	Outputs contents of SBACK flag are from SDA pin at falling edge of SCL pin when clock counter is 8	Status of SDA pin is written to SBCK flag at rising edge of SCL pin when clock counter reaches 9

(1) Program example in I²C bus mode (in master transmission mode)

Example To transmit 1-byte data "96H" to device with slave address of "1010010B"

```
        SDA     FLG     P0A3
        SCL     FLG     P0A2
        SDABIO  FLG     P0ABIO3
        SCLBIO  FLG     P0ABIO2
INIT:
        CLR4    SIO0CH, SB, SIO0MS, SIO0TX
        SET2    SDABIO, SCLBIO          ; Issues start bit by program
        SET2    SDA, SCL
        CLR1    SDA
        CLR1    SCL
        MOV     DBF1, #0AH              ; Sets slave address
        MOV     DBF0, #4                ; "0" of bit b0 indicates transmission
        PUT     SIO0SFR, DBF
        INITFLG NOT SIO0CH, SB, SIO0MS, SIO0TX
                                        ; I²C bus, master, transmission
        CLR2    SIO0CK1, SIO0CK0        ; Clock cycle = 37.5 kHz (≤ 100 kHz)
        INITFLG SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling of shift clock when clock counter is "8"
LOOP1:
        SKT1    SIO0SF8                 ; Waits for data
        BR      LOOP1
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        MOV     DBF1, #9                ; Sets transmit data
        MOV     DBF0, #6
        PUT     SIO0SFR, DBF
        INITFLG SBACK, SIO0NWT, SIO0WRQ1, NOT SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling of shift clock when clock counter is "9"
LOOP2:
        SKT1    SIO0SF9                 ; Waits for acknowledge
        BR      LOOP2
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        SKF1    SBACK                   ; Detects acknowledge
        BR      INIT                    ; Redoes if NACK
        INITFLG SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is "8"
```

```
LOOP3:
        SKT1    SIO0SF8                         ; Waits for data
        BR      LOOP3
        CALL    CLK_WAIT                        ; Waits until SCL pin goes low
        INITFLG SBACK, SIO0NWT, SIO0WRQ1, NOT SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “9”
LOOP4:
        SKT1    SIO0SF9                         ; Waits for acknowledge
        BR      LOOP4
        CALL    CLK_WAIT                        ; Waits until SCL pin goes low
        SKF1    SBACK                           ; Detects acknowledge
        BR      INIT                            ; Redoes if NACK
        CLR4    SIO0CH, SB, SIO0MS, SIO0TX
LOOP5:
        SET1    SCL                             ; Issues stop bit by program
        SKT1    SCL
        BR      LOOP5
        SET1    SDA
        ⋮
CLK_WAIT:                                       ; Subroutine
        SKF1    SCL
        BR      CLK_WAIT
        RET
        ⋮
```

(2) Program example in I²C bus mode (master reception mode)

Example To receive 2-byte data from device with slave address “1010010B” and stores the data at addresses 00H through 03H of BANK0

```
SDA     FLG     P0A3
SCL     FLG     P0A2
SDABIO  FLG     P0ABIO3
SCLBIO  FLG     P0ABIO2

DATA1H  MEM     0.00H       ; Stores higher 4 bits of first byte
DATA1L  MEM     0.01H       ; Stores lower 4 bits of first byte
DATA2H  MEM     0.02H       ; Stores higher 4 bits of second byte
DATA2L  MEM     0.03H       ; Stores lower 4 bits of second byte
```

```
INIT:
        CLR4    SIO0CH, SB, SIO0MS, SIO0TX
        SET2    SDABIO, SCLBIO          ; Issues start bit by program
        SET2    SDA, SCL
        CLR1    SDA
        CLR1    SCL
        MOV     DBF1, #0AH              ; Sets slave address
        MOV     DBF0, #5                ; "1" of bit b0 indicates reception
        PUT     SIO1SFR, DBF
        INITFLG NOT SIO0CH, SB, SIO0MS, SIO0TX
                                        ; I2C bus, master, transmission
        CLR2    SIO0CK1, SIO0CK0        ; Clock cycle = 75 kHz (≤ 100 kHz)
        INITFLG SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is "8"
LOOP1:
        SKT1    SIO0SF8                 ; Waits for data
        BR      LOOP1
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        INITFLG SBACK, SIO0NWT, SIO0WRQ1, NOT SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is "9"
LOOP2:
        SKT1    SIO0SF9                 ; Waits for acknowledge
        BR      LOOP2
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        SKF1    SBACK                   ; Detects acknowledge
        BR      INIT
        CLR1    SDABIO                  ; Sets SDA pin in input (reception) mode
        INITFLG NOT SIO0CH, SB, SIO0MS, NOT SIO0TX
                                        ; I2C bus, master, reception
        INITFLG SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter "8"
```

```
LOOP3:
        SKT1    SIO0SF8                 ; Waits for data
        BR      LOOP3
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        SET1    SDABIO                  ; Sets  SDA pin in output (acknowledge output) mode
        GET     DBF, SIO0SFR            ; Reads receive data
        ST      DATA1H, DBF1            ; Stores read data
        ST      DATA1L, DBF0
        INITFLG NOT SBACK, SIO0NWT, SIO0WRQ1, NOT SIO0WRQ0
        ; Outputs ACK (low level)
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “9”
LOOP4:
        SKT1    SIO0SF9                 ; Waits for acknowledge
        BR      LOOP4
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        CLR1    SDABIO                  ; Sets SDA pin in input (data reception) mode
        INITFLG SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “8”
LOOP5:
        SKT1    SIO0SF8                 ; Waits for data
        BR      LOOP5
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        SET1    SDABIO                  ; Sets SDA pin in output (acknowledge output) mode
        GET     DBF, SIO0SFR            ; Reads receive data
        ST      DATA2H, DBF1            ; Stores read data
        ST      DATA2L, DBF0
        INITFLG NOT SBACK, SIO0NWT, SIO0WRQ1, NOT SIO0WRQ0
        ; Outputs ACK (low level)
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “9”
LOOP6:
        SKT1    SIO0SF9                 ; Waits for acknowledge
        BR      LOOP6
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        CLR4    SIO0CH, SB, SIO0MS, SIO0TX
```

```
LOOP7:
        SET1    SCL                     ; Issues stop bit by program
        SKT1    SCL
        BR      LOOP7
        SET1    SDA
        ⋮
CLK_WAIT:                               ; Subroutine
        SKF1    SCL
        BR      CLK_WAIT
        RET
        ⋮
```

(3) Program example in I²C bus mode (in slave transmission mode)

Example To transmit 2-byte data "96C3H" with slave address of "1010010B"

```
        SDA     FLG     P0A3
        SCL     FLG     P0A2
        SDABIO  FLG     P0ABIO3
        SCLBIO  FLG     P0ABIO2

        NG      FLG     0.00H.0         ; "1" if data cannot be received

INIT:
        CLR2    SCLBIO, SDABIO
        INITFLG NOT SIO0CH, SB, NOT SIO0MS, NOT SIO0TX
                                        ; I²C bus, slave, reception
        INITFLG SBACK, SIO0NWT, SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is "8" after start
        ; condition has been detected
LOOP1:
        SKT1    SBSTT                   ; Waits until start bit is detected
        BR      LOOP1
```

```
LOOP2:
        SKT1    SIO0SF8                 ; Waits for data
        BR      LOOP2
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        SET1    SDABIO                  ; Sets SDA pin in output (acknowledge output) mode
        GET     DBF, SIO0SFR            ; Reads slave address
        ST      R1, DBF1
        ST      R0, DBF0
        SET2    CMP, Z                  ; Detects coincidence of slave address
        SUB     DBF1, #0AH              ; If transmission/reception mode is set,
        SUB     DBF0, #5                ; does not judge bit b0
        SKT1    Z
        BR      NACK0                   ; Slave address does not coincide
        MOV     DBF1, #9                ; Sets transmit data
        MOV     DBF0, #6
        PUT     SIO0SFR, DBF
        CLR1    NG                      ; Slave address coincides
        CLR1    SDA                     ; Sends ACK signal
        INITFLG NOT SBACK, SIO0NWT, SIO0WRQ1, NOT SIO0WRQ0
        ; Outputs ACK (low level)
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “9”
        BR      LOOP3
NACK0:
        SET1    NG                      ; Slave address does not coincide
        SET1    SDA                     ; Sends NACK signal
        INITFLG SBACK, SIO0NWT, SIO0WRQ1,  NOT SIO0WRQ0
        ; Outputs NACK (high level)
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “9”
LOOP3:
        SKT1    SIO0SF9                 ; Waits for acknowledge
        BR      LOOP3
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        CLR1    SDABIO                  ; Sets SDA pin in input (data reception) mode
        SKF1    NG
        BR      INIT                    ; Redoes if slave address does not coincide
        INITFLG NOT SIO0CH, SB, NOT SIO0MS, SIO0TX
                                        ; I2C bus, slave, transmission
        INITFLG SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “8”
```

```
LOOP4:
        SKT1    SIO0SF8                 ; Waits for data
        BR      LOOP4
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        MOV     DBF1, #0CH              ; Sets transmit data
        MOV     DBF0, #3
        PUT     SIO0SFR, DBF
        INITFLG NOT SBACK, SIO0NWT, SIO0WRQ1, NOT SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “9”
LOOP5:
        SKT1    SIO0SF9                 ; Waits for acknowledge
        BR      LOOP5
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        SKF1    SBACK
        BR      INIT
        INITFLG SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “8”
LOOP6:
        SKT1    SIO0SF8                 ; Waits for data
        BR      LOOP6
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        INITFLG NOT SBACK, SIO0NWT, SIO0WRQ1, NOT SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “9”
LOOP7:
        SKT1    SIO0SF9                 ; Waits for acknowledge
        BR      LOOP7
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        SKF1    SBACK
        BR      INIT
        INITFLG NOT SIO0CH, SB, NOT SIO0MS, NOT SIO0TX
                                        ; I2C bus, slave, reception
        INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “8”
```

```
LOOP8:
        SKF1    SBBSY                   ; Waits until stop bit is detected
        BR      LOOP8
        CLR4    SIO0CH, SB, SIO0MS, SIO0TX
        ⋮
CLK_WAIT:                               ; Subroutine
        SKF1    SCL
        BR      CLK_WAIT
        RET
        ⋮
```

(4) Program example in I²C bus mode (slave reception mode)

Example To receive 1-byte data from master and store it addresses 00H and 01H of BANK0. Slave address is "1010010B".

```
        SDA     FLG     P0A3
        SCL     FLG     P0A2
        SDABIO  FLG     P0ABIO3
        SCLBIO  FLG     P0ABIO2

        DATAH   MEM     0.00H           ; Stores higher 4 bits
        DATAL   MEM     0.01H           ; Stores lower 4 bits
        NG      FLG     0.02H.0         ; "0" if data is not received

INIT:
        INITFLG NOT SIO0CH, SB, NOT SIO0MS, NOT SIO0TX
                                        ; I²C bus, slave, reception
        INITFLG SBACK, SIO0NWT, SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is "8"
LOOP1:
        SKT1    SBSTT                   ; Waits until start bit is detected
        BR      LOOP1
```

```
LOOP2:
        SKT1    SIO0SF8                 ; Waits for data
        BR      LOOP2
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        SET1    SDABIO                  ; Sets SDA pin in output (acknowledge output) mode
        GET     DBF, SIO0SFR            ; Reads slave address
        SKNE    DBF1, #0AH              ; Detects coincidence of slave address
        SKE     DBF0, #4                ; CMP flag may be used
        BR      NACK0                   ; Slave address does not coincide
        CLR1    NG                      ; Slave address coincides
        CLR1    SDA                     ; Sends ACK signal
        INITFLG NOT SBACK, SIO0NWT, SIO0WRQ1, NOT SIO0WRQ0
        ; Outputs ACK (low level)
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “9”
        BR      LOOP3
NACK0:
        SET1    NG                      ; Slave address does not coincide
        SET1    SDA                     ; Sends NACK signal
        INITFLG SBACK, SIO0NWT, SIO0WRQ1, NOT SIO0WRQ0
        ; Outputs NACK (high level)
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “9”
LOOP3:
        SKT1    SIO0SF9                 ; Waits for acknowledge
        BR      LOOP3
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        CLR1    SDABIO                  ; Sets SDA pin in input (data reception) mode
        SKF1    NG
        BR      INIT                    ; Redoes if slave address does not coincide
        INITFLG SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “8”
```

```
LOOP4:
        SKT1    SIO0SF8                 ; Waits for data
        BR      LOOP4
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        SET1    SDABIO                  ; Sets SDA pin in output (acknowledge output) mode
        GET     DBF, SIO0SFR            ; Reads receive data
        ST      DATAH, DBF1             ; Stores read data
        ST      DATAL, DBF0
        CLR1    SDA                     ; Sends ACK signal
        INITFLG NOT SBACK, SIO0NWT, SIO0WRQ1, NOT SIO0WRQ0
        ; Outputs ACK (low level)
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “9”
LOOP5:
        SKT1    SIO0SF9                 ; Waits for acknowledge
        BR      LOOP5
        CALL    CLK_WAIT                ; Waits until SCL pin goes low
        CLR1    SDABIO
        INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is falling edge of shift clock when clock counter is “8”
LOOP6:
        SKF1    SBBSY                   ; Waits until stop bit is detected
        BR      LOOP6
        CLR4    SIO0CH, SB, SIO0MS, SIO0TX
        ⋮
CLK_WAIT:                               ; Subroutine
        SKF1    SCL
        BR      CLK_WAIT
        RET
        ⋮
```

19.11.2 Using 2-line serial I/O mode

The two-line serial I/O mode is selected by resetting both the SIO0CH and SB flags to “0”.

In this mode, the $P0A_3$/SDA and $P0A_2$/SCL pins are used.

Figure 19-15 shows the I/O block and communication method in the two-line serial I/O mode.

Table 19-7 shows the functions and operations of the respective pins and control register in the two-line serial I/O mode.

As shown in Figure 19-15 and Table 19-7, an internal clock (master) and external clock (slave) operation may be performed in the two-line serial I/O mode. Data can be transmitted (TX) or received (RX) in both the master and slave modes.

The master or slave operation is selected by the SIO0MS flag, and reception or transmission is selected by the SIO0TX flag.

During the master operation, the internal shift clock is output from the $P0A_2$/SCL pin. If transmission is performed at this time, data is output from the $P0A_3$/SDA pin at the falling edge of the shift clock. If reception is performed, the status of the $P0A_3$/SDA pin is input to the presettable shift register 0 at the rising edge of the shift clock.

During the slave operation, the $P0A_2$/SCL pin is floated (Hi-Z state), and the device waits for an external clock. If transmission is performed at this time, data is output from the $P0A_3$/SDA pin at the falling edge of the shift clock. If reception is performed, the status of the $P0A_3$/SDA pin is input to the presettable shift register 0 at the rising edge of the clock applied to the $P0A_2$/SCL pin.

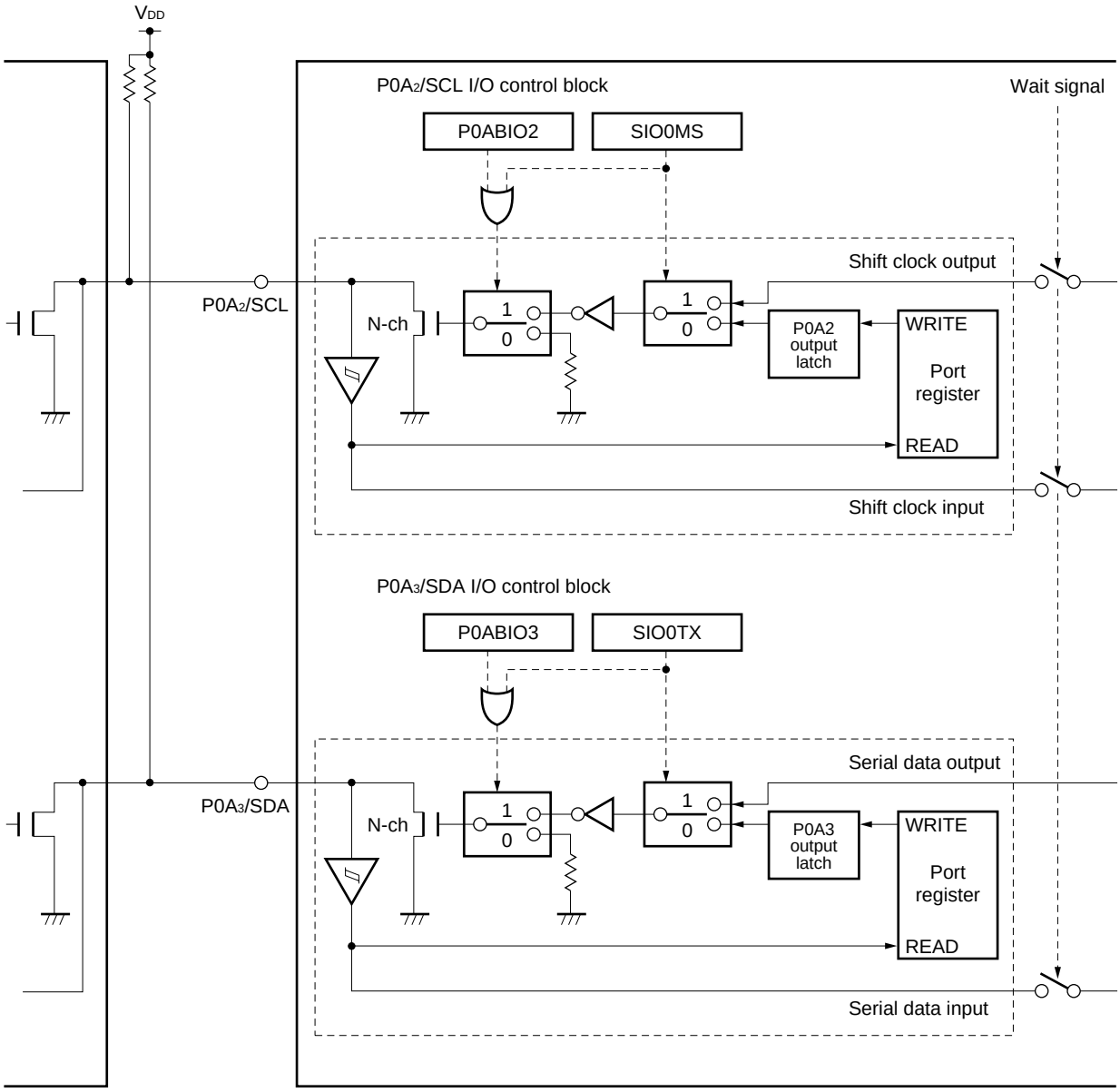
The $P0A_3$/SDA and $P0A_2$/SCL pins are N-ch open-drain output pins; therefore, the communication line goes low if either the master or slave outputs a low level.

When the values output to the $P0A_3$/SDA and $P0A_2$/SCL pins are read, the “status of pin at that time” is read.

Paragraphs (1) through (4) below Table 19-7 show program examples for transmission and reception during master and slave operations.

Figure 19-15. I/O Block and Communication Method in 2-Line Serial Mode

I/O block



Communication method

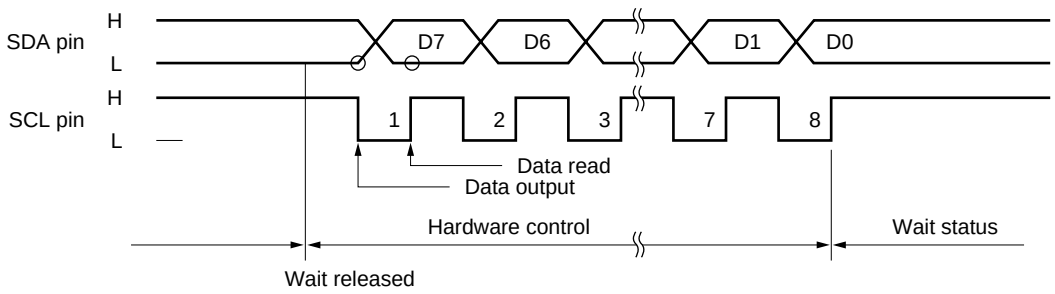


Table 19-7. Outline of Operation in 2-Line Serial I/O Mode

Operation Mode / Item		2-line Serial I/O Mode SIO0CH=0, SB=0			
		Slave operation SIO0MS=0		Master operation SIO0MS=1	
		Reception (RX) SIO0TX=0	Transmission (TX) SIO0TX=1	Reception (RX) SIO0TX=0	Transmission (TX) SIO0TX=1
Setting status of each pin	$P0A_3$/SDA	**When P0ABIO3 = 0** Floating External data input wait **When P0ABIO3 = 1** General-purpose output port Outputs contents of output latch. Normally, P0ABIO3 is reset to 0.	Outputs contents of SIO0SFR at falling edge of external clock regardless of P0ABIO3	**When P0ABIO3 = 0** Floating External data input wait **When P0ABIO3 = 1** General-purpose output port Outputs contents of output latch. Normally, P0ABIO3 is reset to 0.	Outputs contents of SIO0SFR at falling edge of internal shift clock regardless of P0ABIO3
	$P0A_2$/SCL	**When P0ABIO2 = 0** Floating External clock input wait **When P0ABIO2 = 1** General-purpose output port Outputs contents of output latch. Normally, P0ABIO2 is reset to 0.	**When P0ABIO2 = 0** Floating External clock input wait **When P0ABIO2 = 1** General-purpose output port Outputs contents of output latch. Normally, P0ABIO2 is reset to 0.	Outputs internal shift clock regardless of P0ABIO2	Outputs internal shift clock regardless of P0ABIO2
Clock counter operation		Incremented at rising edge of SCL pin			
Operation of presettable shift register 0 (SIO0SFR)		**Output** Not output **Input** Shifts data of SDA pin from LSB and inputs it each time SCL pin rises	**Output** Shifts data from MSB and outputs it to SDA each time SCL pin falls **Input** Shifts data of SDA pin from LSB and inputs it each time SCL pin rises	**Output** Not output **Input** Shifts data of SDA pin from LSB and inputs it each time SCL pin rises	**Output** Shifts data from MSB and outputs it to SDA each time SCL pin falls **Input** Shifts data of SDA pin from LSB and inputs it each time SCL pin rises
Wait operation		Serial communication is started when “1” is written to SIO0NWT. SIO0NWT is reset to “0” under condition set by SIO0WRQ1 and SIO0WRQ0			
		When SIO0NWT = 0 SCL pin is floated. SDA pin is floated. **When SIO0NWT = 1** Floats SCL pin and waits for external clock input. SDA pin is floated and data of SDA pin is input to SIO0SFR at rising edge of SCL pin.	**When SIO0NWT = 0** SCL pin is floated. SDA pin retains its status. **When SIO0NWT=1** Floats SCL pin and waits for external clock input. SDA pin is floated and data of SDA pin is input to SIO0SFR at falling edge of SCL pin.	**When SIO0NWT = 0** SCL pin is floated. SDA pin is floated. **When SIO0NWT = 1** Outputs internal shift clock from SCL pin. SDA pin is floated and data of SDA pin is input to SIO0SFR at rising edge of SCL pin.	**When SIO0NWT = 0** SCL pin is floated. SDA pin retains its status. **When SIO0NWT = 1** Outputs internal shift clock from SCL pin. Outputs contents of SIO0SFR to SDA pin at falling edge of SCL pin.

(1) Program example in 2-line serial I/O mode (master transmission mode)

Example To transmit 2-byte data “A596H”

```
        INITFLG  NOT SIO0CH, NOT SB, SIO0MS, SIO0TX
                                          ; 2-line serial I/O, master, transmission
        CLR2     SIO0CK1, SIO0CK0         ; Clock cycle = 37.5 kHz
        MOV      DBF1, #0AH               ; Sets first byte of transmit data
        MOV      DBF0, #5
        PUT      SIO0SFR, DBF
        INITFLG  NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP1:
        SKF1     SIO0NWT                  ; Waits until wait status is released
        BR       LOOP1
        MOV      DBF1, #9                 ; Sets second byte of transmit data
        MOV      DBF0, #6
        PUT      SIO0SFR, DBF
        INITFLG  NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
        LOOP2:
        SKF1     SIO0NWT                  ; Wait until wait status is released
        BR       LOOP2
        ⋮
```

(2) Program example in 2-line serial I/O mode (master reception mode)

Example To receive and store 2-byte data to addresses 00H through 03H of BANK0

```
        SDABIO  FLG         P0ABIO3

        DATA1H  MEM         0.00H           ; Stores higher 4 bits of first byte
        DATA1L  MEM         0.01H           ; Stores lower 4 bits of first byte
        DATA2H  MEM         0.02H           ; Stores higher 4 bits of second byte
        DATA2L  MEM         0.03H           ; Stores lower 4 bits of second byte

        CLR1    SDABIO
        INITFLG NOT SIO0CH, NOT SB, SIO0MS, NOT SIO0TX
                                            ; 2-line serial I/O, master, transmission
        INITFLG NOT SIO0CK1, SIO0CK0        ; Clock cycle = 75 kHz
        INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP1:
        SKF1    SIO0NWT             ; Wait until wait status is released
        BR      LOOP1
        GET     DBF, SIO0SFR        ; Reads receive data
        ST      DATA1H, DBF1        ; Stores read data
        ST      DATA1L, DBF0
        INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP2:
        SKF1    SIO0NWT             ; Waits until wait status is released
        BR      LOOP2
        GET     DBF, SIO0SFR        ; Reads receive data
        ST      DATA2H, DBF1        ; Stores read data
        ST      DATA2L, DBF0
        ⋮
```

(3) Program example in 2-line serial I/O mode (slave transmission mode)

Example To transmit 2-byte data “A596H”

```
            SCLBIO  FLG         P0ABIO2

            CLR1    SCLBIO
            INITFLG NOT SIO0CH, NOT SB, NOT SIO0MS, SIO0TX
                                              ; 2-line serial I/O, slave, transmission
            MOV     DBF1, #0AH                ; Sets transmit data
            MOV     DBF0, #5
            PUT     SIO0SFR, DBF
            INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
            ;  Releases wait
            ;  Wait condition is rising edge of shift clock when clock counter is “8”
LOOP1:
            SKF1    SIO0NWT                   ; Waits until wait status is released
            BR      LOOP1
            MOV     DBF1, #9                  ; Sets transmit data
            MOV     DBF0, #6
            PUT     SIO0SFR, DBF
            INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
            ;  Releases wait
            ;  Wait condition is rising edge of shift clock when clock counter is “8”
LOOP2:
            SKF1    SIO0NWT                   ; Waits until wait status is released
            BR      LOOP2
            ⋮
```

(4) Program example in 2-line serial I/O mode (slave reception mode)

Example To receive and store 2-byte data to addresses 00H through 03H of BANK0

```
        SDABIO  FLG     P0ABIO3
        SCLBIO  FLG     P0ABIO2

        DATA1H  MEM     0.00H       ; Stores higher 4 bits of first byte
        DATA1L  MEM     0.01H       ; Stores lower 4 bits of first byte
        DATA2H  MEM     0.02H       ; Stores higher 4 bits of second byte
        DATA2L  MEM     0.03H       ; Stores lower 4 bits of second byte

        CLR2    SCLBIO, SDABIO
        CLR4    SIO0CH, SB, SIO0MS, SIO0TX
                                    ; 2-line serial I/O, slave, reception
        INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP1:
        SKF1    SIO0NWT             ; Waits until wait status is released
        BR      LOOP1
        GET     DBF, SIO0SFR        ; Reads receive data
        ST      DATA1H, DBF1        ; Stores read data
        ST      DATA1L, DBF0
        INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP2:
        SKF1    SIO0NWT             ; Waits until wait status is released
        BR      LOOP2
        GET     DBF, SIO0SFR        ; Reads receive data
        ST      DATA2H, DBF1        ; Stores read data
        ST      DATA2L, DBF0
        ⋮
```

19.11.3 Using three-line serial I/O mode

The three-line serial I/O mode is selected by setting the SIO0CH flag to "1" and resetting the SB flags to "0".

In this mode, the $P0A_1/\overline{SCK_0}$, $P0A_0/SO_0$, and $P0B_3/SI_0$ pins are used.

Figure 19-16 shows the I/O block and communication method in the three-line serial I/O mode.

Table 19-8 shows the functions and operations of the respective pins and control register in the three-line serial I/O mode.

As shown in Figure 19-16 and Table 19-8, an internal clock (master) and external clock (slave) operation may be performed in the three-line serial I/O mode. Data can be transmitted (TX) or received (RX) in both the master and slave modes.

The master or slave operation is selected by the SIO0MS flag, and reception or transmission is selected by the SIO0TX flag.

During the master operation, the internal shift clock is output from the $P0A_1/\overline{SCK_0}$ pin. If transmission is performed at this time, data is output from the $P0A_0/SO_0$ pin at the falling edge of the shift clock.

During master operation, the status of the $P0B_3/SI_0$ pin is input to the presettable shift register 0 at the rising edge of the shift clock, regardless of whether transmission or reception is performed. At this time, however, the $P0B_3/SI_0$ pin must be set in this input mode.

During the slave operation, the $P0A_1/\overline{SCK_0}$ pin is floated (Hi-Z state), and the device waits for an external clock. If transmission is performed at this time, data is output from the $P0A_3$/SDA pin at the falling edge of the shift clock.

During slave operation, the status of the $P0B_3/SI_0$ pin is input to the presettable shift register 0 at the rising edge of the shift clock, regardless of whether transmission or reception is performed. However, the $P0B_3/SI_0$ pin must be set in the input mode.

The "status of the output latch at that time" is read when the contents of the port register corresponding to the $P0A_1/\overline{SCK_0}$ or $P0A_0/SO_0$ pin are read.

Paragraphs (1) through (4) below Table 19-8 show program examples for transmission and reception during master and slave operations.

Figure 19-16. I/O Block and Communication Method in 3-Line Serial Mode (1/2)

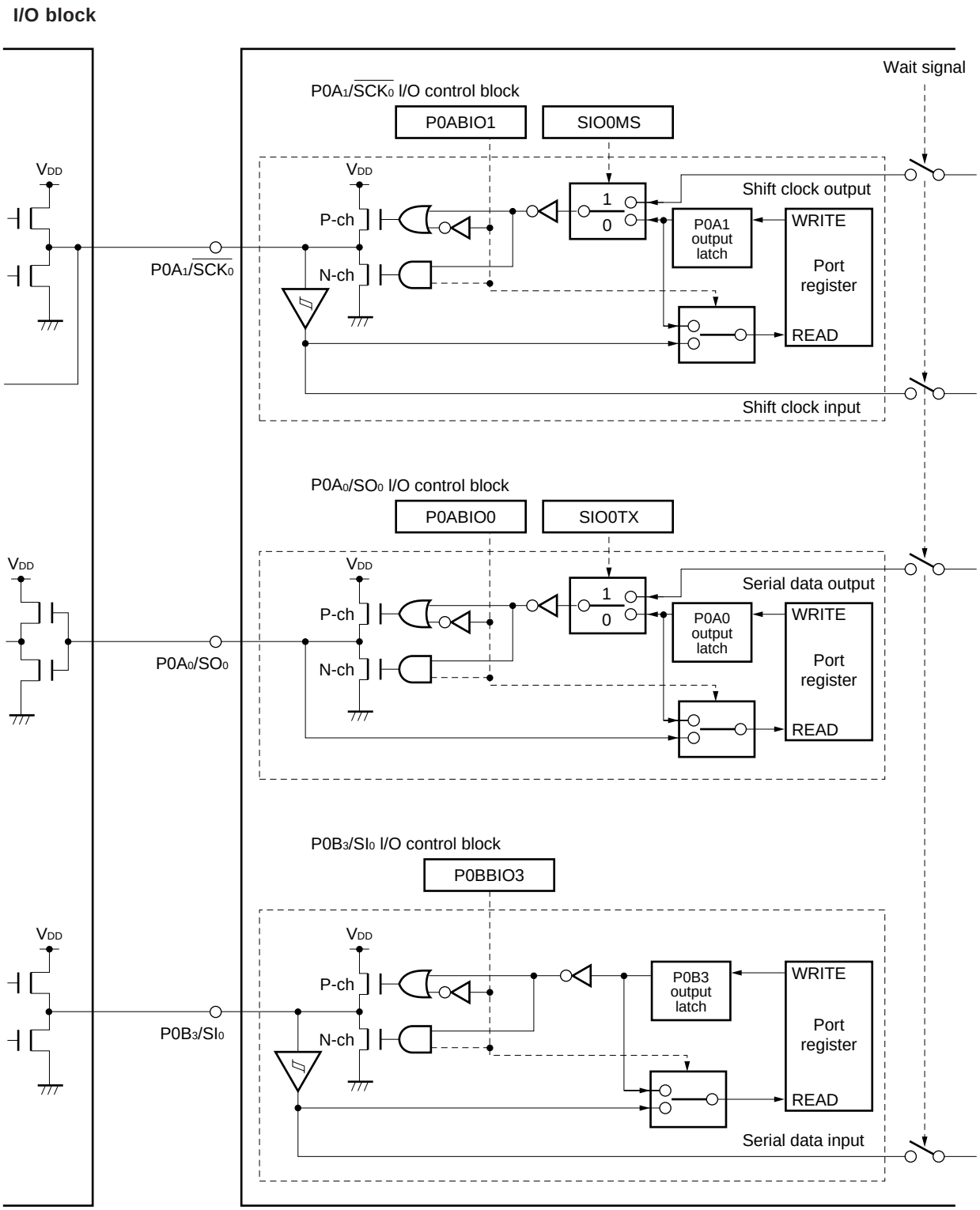


Figure 19-16. I/O Block and Communication Method in 3-Line Serial Mode (2/2)

Communication method

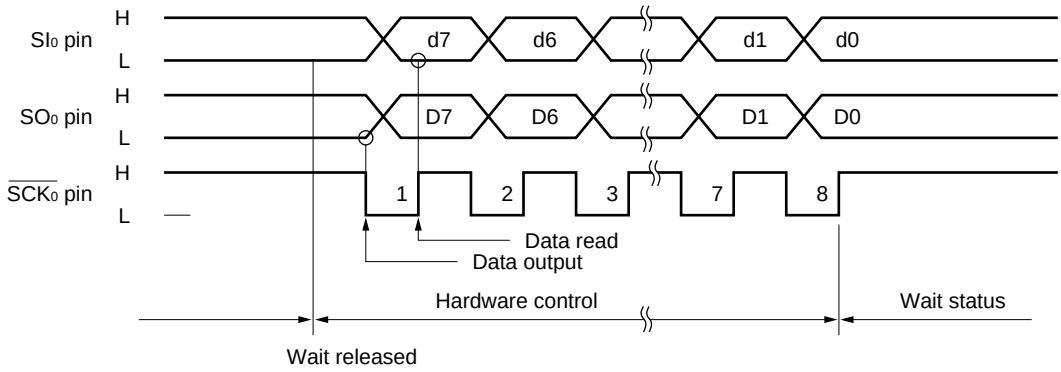


Table 19-8. Outline of Operation in 3-Line Serial I/O Mode

Item \ Operation Mode		3-line Serial I/O Mode SIO0CH=1, SB=0			
		Slave operation SIO0MS=0		Master operation SIO0MS=1	
		Reception (RX) SIO0TX=0	Transmission (TX) SIO0TX=1	Reception (RX) SIO0TX=0	Transmission (TX) SIO0TX=1
Setting status of each pin	$P0A_1/\overline{SCK_0}$	**When P0ABIO1 = 0** Floating External clock input wait **When P0ABIO1 = 1** General-purpose output port Outputs contents of output latch. Normally, P0ABIO1 is reset to 0.	**When P0ABIO1 = 0** Floating External clock input wait **When P0ABIO1 = 1** General-purpose output port Outputs contents of output latch. Normally, P0ABIO1 is reset to 0.	Internal shift clock is output regardless of P0ABIO1	Internal shift clock is output regardless of P0ABIO1
	$P0A_0/SO_0$	**When P0ABIO0 = 0** General-purpose input port Floating **When P0ABIO0 = 1** General-purpose output port Outputs contents of output latch.	Outputs contents of SIO0SFR at falling edge of external clock regardless of P0ABIO0	**When P0ABIO0 = 0** General-purpose input port Floating **When P0ABIO0 = 1** General-purpose output port Outputs contents of output latch.	Outputs contents of SIO0SFR at falling edge of internal shift clock regardless of P0ABIO0
	$P0B_3/SI_0$	**When P0BBIO3 = 0** Floating External data input wait **When P0BBIO3 = 1** General-purpose output port Outputs contents of output latch. Normally, P0BBIO3 is reset to 0.	**When P0BBIO3 = 0** Floating External data input wait **When P0BBIO3 = 1** General-purpose output port Outputs contents of output latch. Normally, P0BBIO3 is reset to 0.	**When P0BBIO3 = 0** Floating External data input wait **When P0BBIO3 = 1** General-purpose output port Outputs contents of output latch. Normally, P0BBIO3 is reset to 0.	**When P0BBIO3 = 0** Floating External data input wait **When P0BBIO3 = 1** General-purpose output port Outputs contents of output latch. Normally, P0BBIO3 is reset to 0.
Clock counter operation		Incremented at rising edge of $\overline{SCK_0}$ pin			
Operation of presettable shift register 0 (SIO0SFR)		**Output** Not output **Input** Shifts data of SI_0 pin from LSB and inputs it each time $\overline{SCK_0}$ pin rises	**Output** Shifts data from MSB and outputs it to SO_0 pin each time $\overline{SCK_0}$ pin falls **Input** Shifts data of SI_0 pin from LSB and inputs it each time $\overline{SCK_0}$ pin rises	**Output** Not output **Input** Shifts data of SI_0 pin from LSB and inputs it each time $\overline{SCK_0}$ pin rises	**Output** Shifts data from MSB and outputs it to SO_0 pin each time $\overline{SCK_0}$ pin falls **Input** Shifts data of SI_0 pin from LSB and inputs it each time $\overline{SCK_0}$ pin rises
Wait operation		Serial communication is started when "1" is written to SIO0NWT. SIO0NWT is reset to "0" under condition set by SIO0WRQ1 and SIO0WRQ0			
		When SIO0NWT = 0 $\overline{SCK_0}$ pin is floated. SO_0 pin is general-purpose port. SI_0 pin is floated. **When SIO0NWT = 1** $\overline{SCK_0}$ pin waits for input of external clock. Inputs data of SI_0 pin to SIO0SFR at rising edge of $\overline{SCK_0}$ pin.	**When SIO0NWT = 0** $\overline{SCK_0}$ pin is floated. SO_0 pin retains its status. SI_0 pin is floated. **When SIO0NWT = 1** $\overline{SCK_0}$ pin waits for input of external clock. Outputs contents of SIO0SFR to SO_0 pin at falling edge of $\overline{SCK_0}$ pin. Inputs data of SI_0 pin to SIO0SFR at rising edge of $\overline{SCK_0}$ pin.	**When SIO0NWT = 0** $\overline{SCK_0}$ pin outputs high level. SO_0 pin is general-purpose port. SI_0 pin is floated. **When SIO0NWT = 1** $\overline{SCK_0}$ pin outputs internal shift clock. Inputs data of SI_0 pin to SIO0SFR at rising edge of $\overline{SCK_0}$ pin.	**When SIO0NWT = 0** $\overline{SCK_0}$ pin outputs high level. SO_0 pin retains its status. SI_0 pin is floated. **When SIO0NWT = 1** Outputs internal shift clock from $\overline{SCK_0}$ pin. Outputs contents of SIO0SFR to SO_0 pin at falling edge of $\overline{SCK_0}$ pin. Inputs data of SI_0 pin to SIO0SFR at rising edge of $\overline{SCK_0}$ pin.

(1) Program example in 3-line serial I/O mode (master transmission mode)

Example To transmit 2-byte data “A596H”

```
        INITFLG  SIO0CH, NOT SB, SIO0MS, SIO0TX
                                          ; 3-line serial I/O, master, transmission
        INITFLG  SIO0CK1, NOT SIO0CK0     ; Clock cycle = 112.5 kHz
        MOV      DBF1, #0AH               ; Sets transmit data
        MOV      DBF0, #5
        PUT      SIO0SFR, DBF
        INITFLG  NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP1:
        SKF1     SIO0NWT                  ; Wait until wait status is released
        BR       LOOP1
        MOV      DBF1, #9                 ; Sets transmit data
        MOV      DBF0, #6
        PUT      SIO0SFR, DBF
        INITFLG  NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP2:
        SKF1     SIO0NWT                  ; Waits until wait status is released
        BR       LOOP2
        ⋮
```

(2) Program example in 3-line serial I/O mode (master reception mode)

Example To receive and store 2-byte data to addresses 00H through 03H of BANK0

```
        SI0BIO   FLG        P0BBIO3

        DATA1H   MEM        0.00H          ; Stores higher 4 bits of first byte
        DATA1L   MEM        0.01H          ; Stores lower 4 bits of first byte
        DATA2H   MEM        0.02H          ; Stores higher 4 bits of second byte
        DATA2L   MEM        0.03H          ; Stores lower 4 bits of second byte

        CLR1     SI0BIO
        INITFLG  SIO0CH, NOT SB, SIO0MS, NOT SIO0TX
                                            ; 3-line serial I/O, master, reception
        SET2     SIO0CK1, SIO0CK0           ; Clock cycle = 225 kHz
        INITFLG  NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP1:
        SKF1     SIO0NWT                    ; Waits until wait status is released
        BR       LOOP1
        GET      DBF, SIO0SFR               ; Reads receive data
        ST       DATA1H, DBF1               ; Stores read data
        ST       DATA1L, DBF0
        INITFLG  NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP2:
        SKF1     SIO0NWT                    ; Waits until wait status is released
        BR       LOOP2
        GET      DBF, SIO0SFR               ; Reads receive data
        ST       DATA2H, DBF1               ; Stores read data
        ST       DATA2L, DBF0
        ⋮
```

(3) Program example in 3-line serial I/O mode (slave transmission mode)

Example To transmit 2-byte data “A596H”

```
            SCK0BIO FLG         P0ABIO1

            CLR1    SCK0BIO
            INITFLG SIO0CH, NOT SB, NOT SIO0MS, SIO0TX
                                            ; 3-line serial I/O, slave, transmission
            MOV     DBF1, #0AH              ; Sets transmit data
            MOV     DBF0, #5
            PUT     SIO1SFR, DBF
            INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
            ; Releases wait
            ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP1:
            SKF1    SIO0NWT             ; Waits until wait status is released
            BR      LOOP1
            MOV     DBF1, #9            ; Sets transmit data
            MOV     DBF0, #6
            PUT     SIO0SFR, DBF
            INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
            ; Releases wait
            ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP2:
            SKF1    SIO0NWT             ; Waits until wait status is released
            BR      LOOP2
            ⋮
```

(4) Program example in 3-line serial I/O mode (slave reception mode)

Example To receive and store 2-byte data to addresses 00H through 03H of BANK0

```
        SCK0BIO FLG     P0ABIO1
        SI0BIO  FLG     P0BBIO3

        DATA1H  MEM     0.00H       ; Stores higher 4 bits of first byte
        DATA1L  MEM     0.01H       ; Stores lower 4 bits of first byte
        DATA2H  MEM     0.02H       ; Stores higher 4 bits of second byte
        DATA2L  MEM     0.03H       ; Stores lower 4 bits of second byte

        CLR2    SCK0BIO, SI0BIO
        INITFLG SIO0CH, NOT SB, NOT SIO0MS, NOT SIO0TX
                                    ; 3-line serial I/O, slave, reception
        INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP1:
        SKF1    SIO0NWT             ; Waits until wait status is released
        BR      LOOP1
        GET     DBF, SIO0SFR        ; Reads receive data
        ST      DATA1H, DBF1        ; Stores read data
        ST      DATA1L, DBF0
        INITFLG NOT SBACK, SIO0NWT, NOT SIO0WRQ1, SIO0WRQ0
        ; Releases wait
        ; Wait condition is rising edge of shift clock when clock counter is “8”
LOOP2:
        SKF1    SIO0NWT             ; Waits until wait status is released
        BR      LOOP2
        GET     DBF, SIO0SFR        ; Reads receive data
        ST      DATA2H, DBF1        ; Stores read data
        ST      DATA2L, DBF0
        ⋮
```

19.12 Reset Status of Serial Interface 0

19.12.1 At power-ON reset

All the $P0A_3$/SDA through $P0A_0/SO_0$ and $P0B_3/SI_0$ pins are set in the general-purpose input port mode (floating output).

The value of the presettable shift register 0 is undefined.

19.12.2 On execution of clock stop instruction

All the $P0A_3$/SDA through $P0A_0/SO_0$ and $P0B_3/SI_0$ pins are set in the general-purpose input port mode (floating output).

The presettable shift register 0 retains the previous value.

19.12.3 At CE reset

All the $P0A_3$/SDA through $P0A_0/SO_0$ and $P0B_3/SI_0$ pins are set in the general-purpose input port mode (floating output).

The presettable shift register 0 retains the previous value.

19.12.4 In halt status

The I/O pins retain the current status.

If the internal clock is used (master operation) at this time, the clock is not output when the “HALT” instruction has been executed.

Therefore, the “HALT” instruction must be executed after communication has been completed when the internal clock is used.

If an external clock is forcibly input, the serial interface 0 operates even when the internal clock is set.

When the external clock is used (slave operation), the operation continues even when the “HALT” instruction is executed.

To release the halt status by using the interrupt of serial interface 0, the internal clock cannot be used as described above.

19.13 Configuration of Serial Interface 1 (SIO1)

Figure 19-17 shows the block diagram of serial interface 1.

As shown in this figure, the shift clock control block of the serial interface 1 consists of a clock I/O pin block, a clock generation block, a wait control block, and a clock count block.

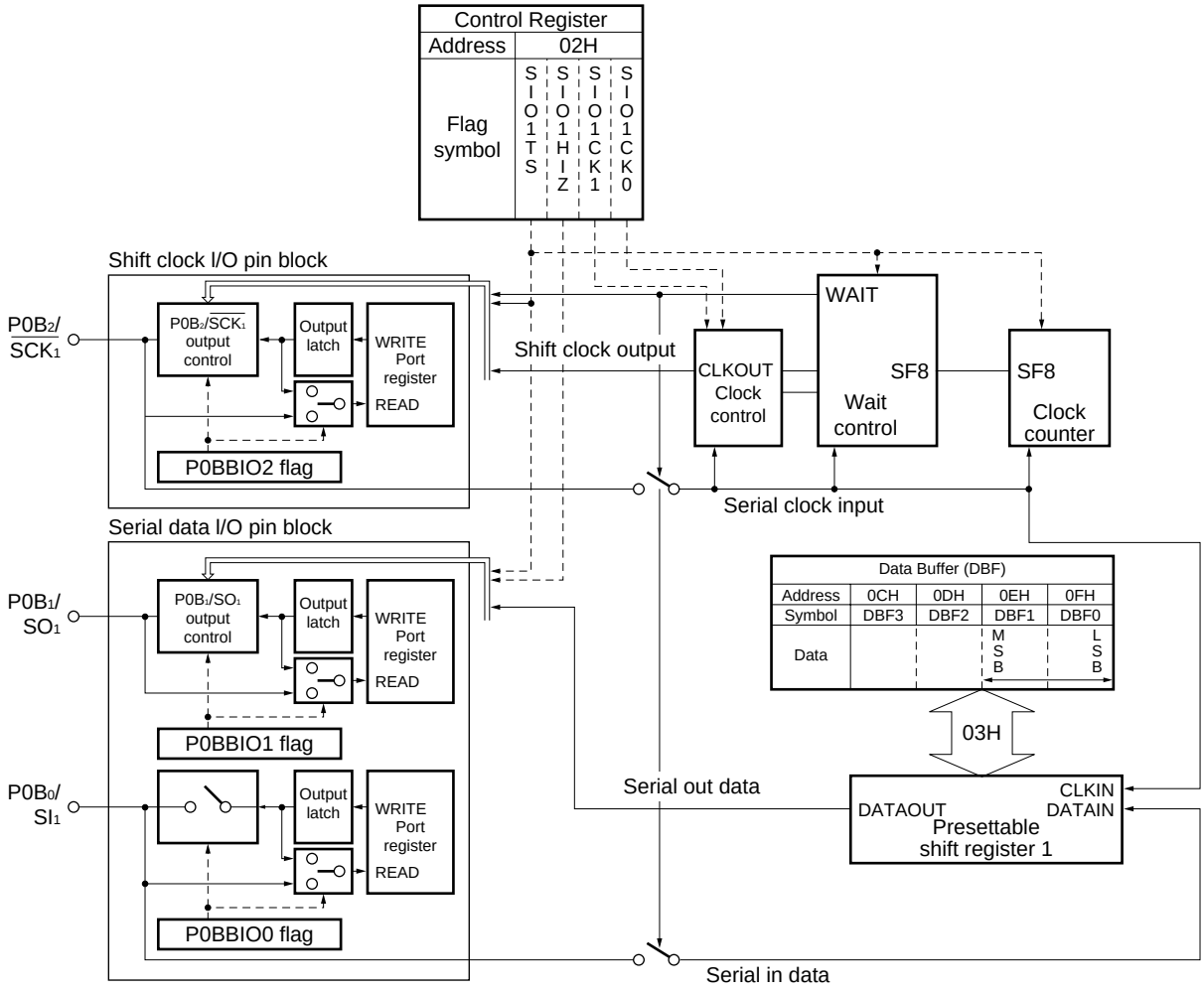
The serial data control block consists of a serial data I/O pin block and a presettable shift register 1.

These blocks are controlled by the flags of control registers.

Data is written to or read from the presettable shift register 1 via data buffer.

19.14 outlines the functions of the respective blocks.

Figure 19-17. Block Diagram of Serial Interface 1



19.14 Functional Outline of Serial Interface 1

Serial interface 1 can be used in three-line serial I/O mode as indicated in Table 19-1.

This interface uses the P0B2/$\overline{SCK_1}$, P0B1/SO1, and P0B0/SI1 pins.

Serial interface 1 can operate with an internal clock or external clock. Moreover, reception or transmission can be selected.

The following **19.14.1** through **19.14.6** outline the functions of the respective blocks of serial interface 1.

For the details of the respective blocks, refer to **19.15** through **19.9**.

19.14.1 Shift clock I/O pin block

This block selects a shift clock I/O pin.

The shift clock I/O pin is selected by the serial I/O1 mode register.

For details, refer to **19.15**.

19.14.2 Serial data I/O pin block

This block selects a serial data I/O pin.

The serial data I/O pin is selected by the serial I/O1 mode select register.

For details, refer to **19.15**.

19.14.3 Clock generation block

This block selects the clock frequency of the shift clock and controls the shift clock output timing.

The clock frequency is selected by the serial I/O1 mode select register.

For details, refer to **19.16**.

19.14.4 Clock counter

This counter counts the rising edges of the clock output by the shift clock output pin and outputs a signal at the eighth clock (SF8 signal).

The SF8 signal is used to place serial communication in the wait (pause) status.

For details, refer to **19.17**.

19.14.5 Presettable shift register 1 (SIO1SFR)

This shift register sets serial out data and stores serial in data.

It performs a shift operation in response to the clock input to the shift clock I/O pin, and inputs or outputs data.

The output data is set and the input data is read via data buffer.

For details, refer to **19.18**.

19.14.6 Wait control block

This block controls the wait (pause) and wait release (communication operation) states of serial communication.

The wait status of serial communication is set or released by the serial I/O1 mode select register.

For details, refer to **19.19**.

19.15 Shift Clock and Serial Data I/O Pin Control Block

The shift clock and serial data I/O pin control block controls the setting of the respective pins and transmission or reception operation of serial interface 1.

These control operations are performed by the serial I/O1 mode select register.

19.15.1 describe the configuration and function of the serial I/O1 mode select register.

19.15.2 shows the status of each pin set by the serial I/O1 mode select register.

19.5.1 Configuration and function of serial I/O1 mode select register (SIO1MODE)

The configuration and function of the serial I/O1 mode select register are illustrated below.

The SIO1CK1 and SIO1CK0 flags select the internal or external clock, and sets the frequency of the internal clock.

For the details of the clock, refer to **19.16**.

The SIO1TS flag sets or releases the wait status of serial interface 1.

For the details of the wait operation, refer to **19.19**.

Name	Flag Symbol b_3	b_2	b_1	b_0	Address	Read/ Write
Serial I/O1 mode select register (SIO1MODE)	SIO1TS	SIO1HIZ	SIO1CK1	SIO1CK0	02H	R/W

Sets I/O clock frequency of serial interface 1	SIO1CK1	SIO1CK0
External clock (slave)	0	0
37.5 kHz (master)	0	1
75 kHz (master)	1	0
450 kHz (master)	1	1

Sets $P0B_1/SO_1$ pin as serial out pin	SIO1HIZ
General-purpose I/O port	0
Serial out	1

Starts serial communication of serial interface 1	SIO1TS
Does not perform serial communication (wait)	0
Performs serial communication (releases wait)	1

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

19.15.2 Pin status set by serial I/O1 mode select register

Table 19-9 shows the pin status set by the serial I/O1 mode select register.

As shown in this table, the I/O select flag must be manipulated to set each pin.

For the details of the I/O select flag, refer to **15. GENERAL-PURPOSE PORTS**.

Table 19-9. Pin Status Set by Serial I/O1 Mode Select Register

SIO1MODE						Pin				
Communication mode	b_2	Setting of serial output	b_1	b_0	Clock direction	Pin symbol	I/O select flag of each pin			Set pin status
	SIO1HIZ		SIO1CK1	SIO1CK0			P0BBIO2	P0BBIO1	P0BBIO0	
3-line serial I/O			0	0	External clock	$P0B_2/\overline{SCK_1}$	0			In wait status: General-purpose input port On release of wait: External clock input
							1			In wait status: General-purpose output port On release of wait: General-purpose output port
			0	1	Internal clock		0			In wait status: General-purpose input port On release of wait: General-purpose input port
			1	0						
			1	1			1			In wait status: High-level output On release of wait: Internal clock output
	0	General-purpose output port				$P0B_1/SO_1$		0		General-purpose input port
								1		General-purpose output port
	1	Serial output						0		General-purpose input port
								1		Serial output
						$P0B_0/SI_1$			0	Serial input
									1	General-purpose output port

19.16 Clock Generation Block

The clock generation block generates a clock when the internal clock is used (master operation) and controls the clock output timing.

The internal clock frequency f_{SC} is set by the SIO1CK1 and SIO1CK0 flags of the serial I/O1 mode select register.

The shift clock is successively output, until the current value of the clock counter described in **19.17** reaches "8".

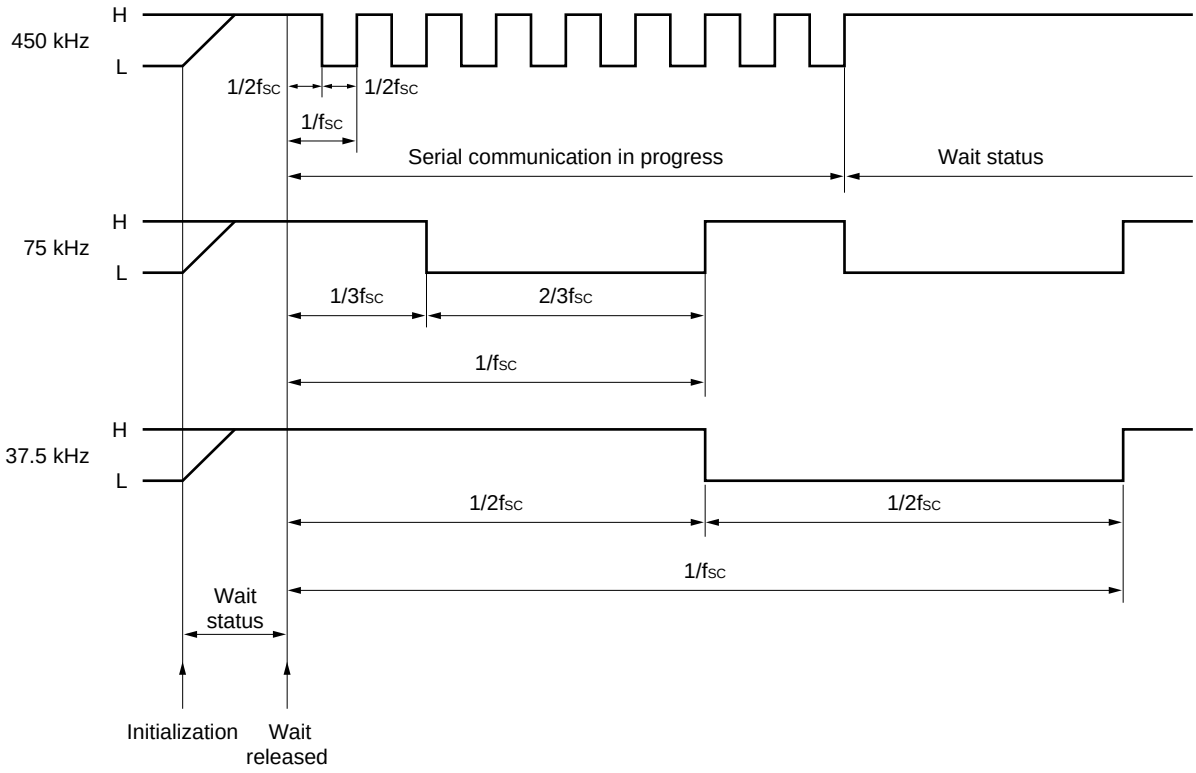
19.6.1 below describes the clock output waveform and generation timing.

19.16.1 Internal shift clock generation timing

(1) When wait status is released from initial status

The "initial status" is the point at which the internal clock is selected, and the $P0B_2/\overline{SCK_1}$ pin is set in the output mode to output a high level.

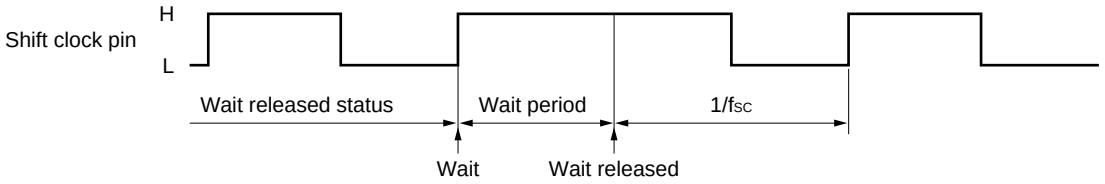
In the wait status, a high level is output to the $P0B_2/\overline{SCK_1}$ pin.



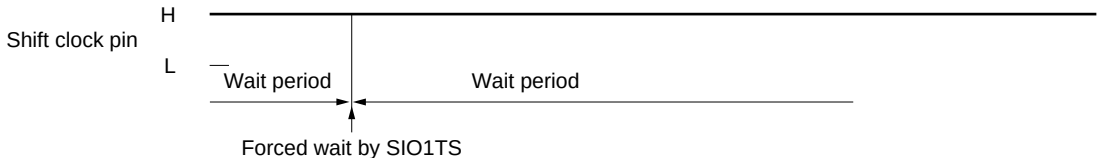
(2) When wait operation is performed

For the details of the wait operation, refer to **19.19**.

(a) When wait status is set because value of clock counter has reached "8" (normal operation)

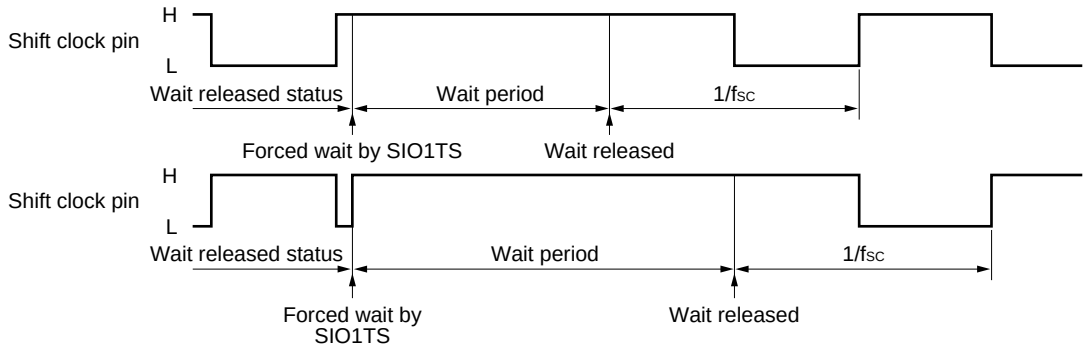


(b) When forced wait status is set during wait status



(c) When forced wait status is set during wait release

At this time, the clock counter is reset.



(d) If wait status is released during wait release

In this case, the clock output waveform is not changed.

The clock counter is not reset, either.

(e) If clock frequency is changed and wait status is released at the same time

The clock frequency can be changed and the wait status can be released by using the serial I/O1 mode select register of the control registers.

Therefore, the clock frequency can be changed and the wait status can be released by one instruction. When this is done, the operation is the same as releasing the wait status from the initial status as described in (1) above.

19.17 Clock Counter

The clock counter is a wrap-around counter that counts the number of clocks input to or output from the shift clock pin (P0B2/$\overline{SCK_1}$ pin).

The clock counter directly reads the status of the shift clock pin. At this time, whether the clock is the internal clock or external clock is not judged.

The clock counter does not operate in the wait status of serial communication.

Serial communication is placed in the wait status at the rising edge of the shift clock when the current value of the clock counter is “8”.

The contents of the clock counter can not be read directly by the program.

The following **19.17.1** and **19.17.2** describe the operation and reset condition of the clock counter.

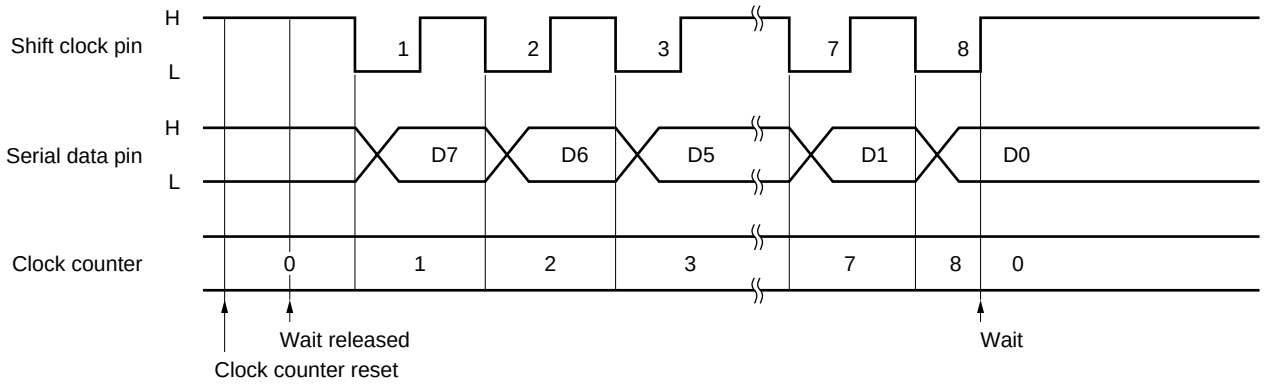
19.7.1 Operation of clock counter

Figure 19-18 shows the operation of the clock counter.

The initial value of the clock counter is “0”. The clock counter is incremented each time the falling edge of the shift clock pin has been detected. After its value has been incremented to “8”, it is reset to “0” at the next rising edge of the shift clock pin.

When the clock counter has been reset to 0, serial communication is placed in the wait status.

Figure 19-18. Operation of Clock Counter



19.17.2 Reset (0) condition of clock counter

The clock counter is reset to 0 under the conditions (1) through (5) below.

(1) On power-ON reset
(2) On execution of clock stop instruction
(3) When “0” has been written to the SIO1TS flag (forced wait)
(4) When the wait status is released and the shift clock rises when the current value of the clock counter is “8”
(5) On CE reset

19.18 Presettable Shift Register 1 (SIO1SFR)

The presettable shift register 1 (SIO1SFR) is an 8-bit shift register that writes serial out data and reads serial in data.

Data is written to or read from the presettable shift register 1 by the “PUT” or “GET” instruction via data buffer.

19.18.1 describes the configuration of the presettable shift register 1 and its relation with the data buffer.

The data of the presettable shift register 1 is shifted in synchronization with the clock applied to the shift clock pin ($P0B_2/\overline{SCK_1}$ pin).

At this time, the most significant bit (MSB) of the presettable shift register 1 is output to the serial data output pin ($P0B_1/SO_1$ pin) in synchronization with the falling edge of the shift clock, and the data of the serial data input pin ($P0B_0/SI_1$ pin) is read to the least significant bit (LSB) of the presettable shift register 1 in synchronization with the rising edge of the clock.

19.8.2 describes the operation.

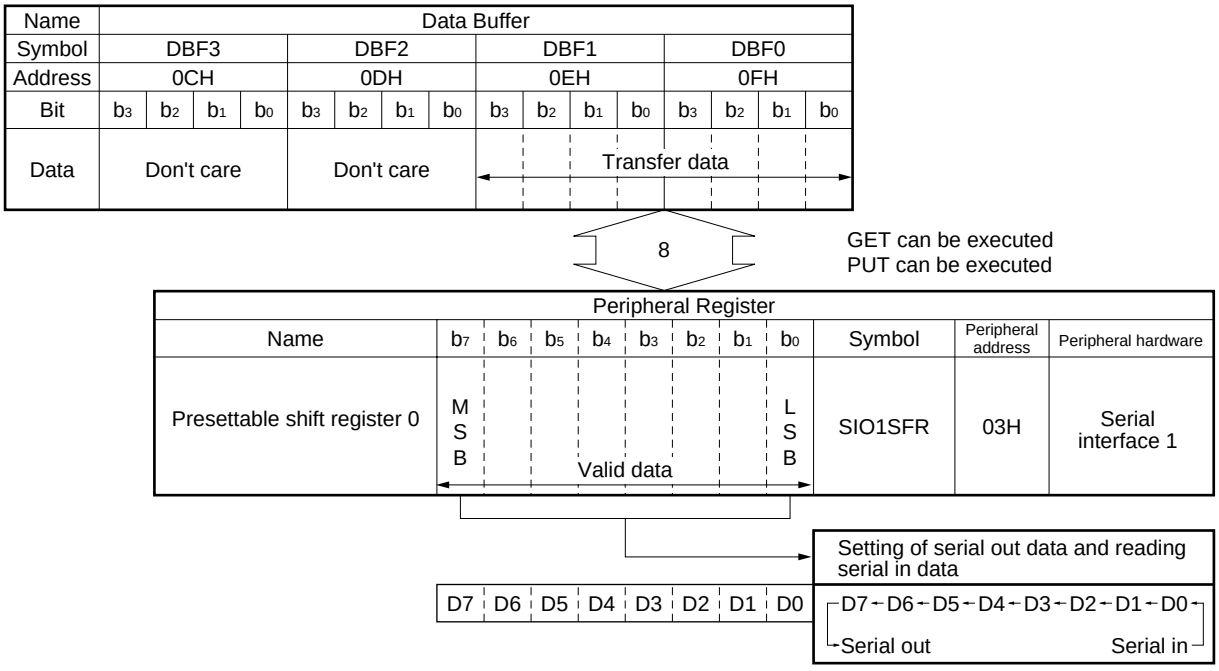
19.18.3 describes the points to be noted in writing or read data to or from the presettable shift register 1.

The presettable shift register 1 does not shift data in the wait status.

For the details of the operations of the register in the respective serial communication modes, refer to **19.20**.

19.18.1 Configuration of presettable shift register 1 and its relation with data buffer

The configuration of the presettable shift register 1 and its relation with the data buffer are illustrated below.



19.18.2 Operation of presettable shift register 1

Figure 19-19 shows the data shift operation of the presettable shift register 1.

Table 19-10 shows the data shift operation during reception or transmission.

Figure 19-19. Data Shift Operation of Presettable Shift Register 1

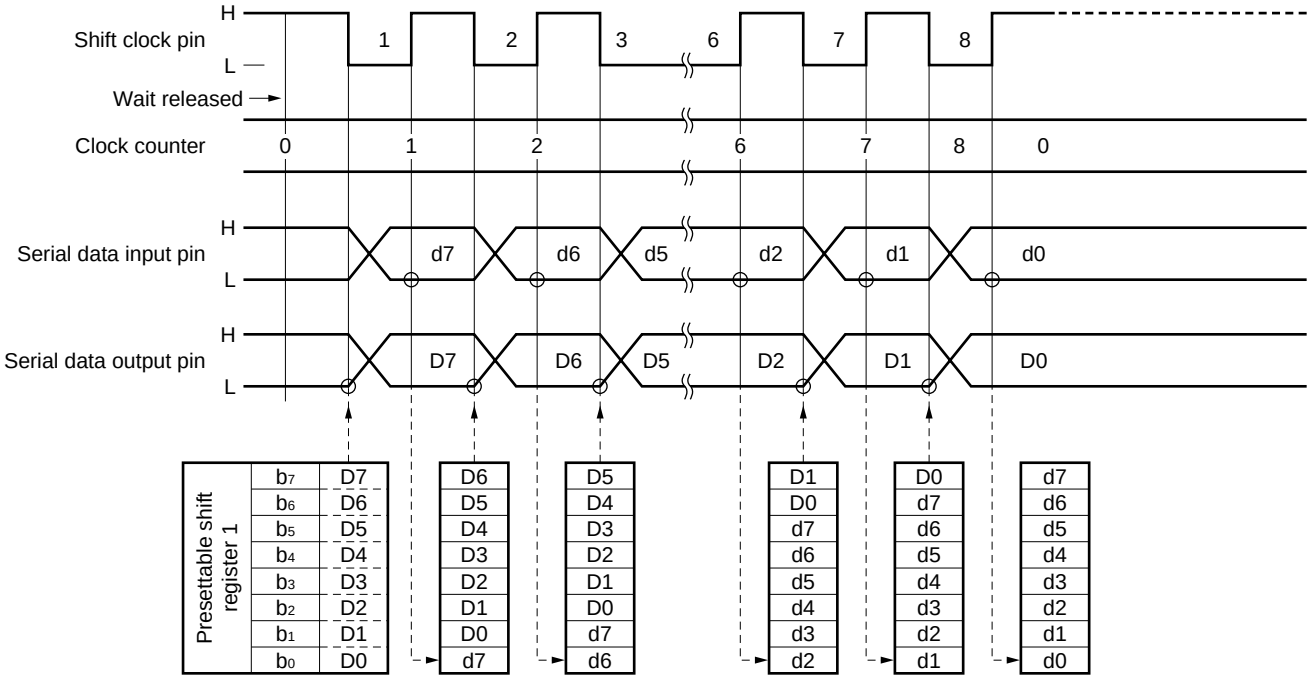


Table 19-10. Data Shift Operation during Reception and Transmission

Serial I/O Mode	
Serial input operation	Serial output operation
Status of $P0B_0/SI_1$ pin is shifted from LSB and input at rising edge of shift clock pin.	Data is shifted from MSB and output to $P0B_1/SO_1$ pin at falling edge of shift clock pin.
Contents of output latch are input when P0BBIO0 flag is “0”.	Data is not output if P0BBIO1 flag is “1” or SIO1HIZ flag is “0”.
Does not operate in wait status.	Does not operate in wait status.

19.18.3 Notes on setting and reading data

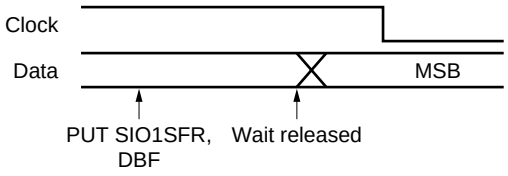
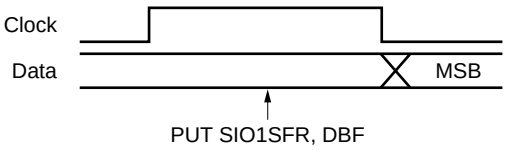
To set data to the presettable shift register 1, use the “PUT SIO1SFR, DBF” instruction.

To read data, use the “GET DBF, SIO1SFR” instruction.

Set or read data in the wait status. While the wait status is released, data may not be correctly set or read depending on the status of the shift clock pin.

Table 19-11 shows the timing of setting and reading data, and points to be noted.

Table 19-11. Reading (GET) and Writing (PUT) Data of Presettable Shift Register 1 and Notes

<table>
<tr><th colspan="2">Status on Execution of PUT/GET</th><th>Status of Shift Clock Pin</th><th>Presettable Shift Register 1 (SIO1SFR)</th></tr>
<tr><td rowspan="2">Wait status</td><td>Read (GET)</td><td>External clock
Floating
Internal clock
High level</td><td>Normal read</td></tr>
<tr><td>Write (PUT)</td><td>Normal write
Outputs MSB contents as data when wait status is released next time (during transmission) (However, if shift clock is low in wait status when external clock is used, data cannot be correctly written and contents of SIO1SFR are lost.)
 
</td></tr>
<tr><td rowspan="3">Wait released status</td><td rowspan="2">Read (GET)</td><td>Low level</td><td>Normal read</td></tr>
<tr><td>High level</td><td>Normal read
(When internal clock is selected, set value is shifted 1 bit and is read (MSB is shifted to LSB).)</td></tr>
<tr><td rowspan="2">Write (PUT)</td><td>High level</td><td>Normal write
Outputs MSB contents when shift clock falls. Clock counter is not reset.
 
</td></tr>
<tr><td>Low level</td><td>Cannot be written normally.
Contents of SIO1SFR are lost.</td></tr>
</table>

19.19 Wait Block

The wait block places communication of serial interface 1 in the wait status or releases the wait status.

The wait block is controlled by the SIO1TS flag of the serial I/O1 mode select register.

19.19.1 below describes the wait operation and points to be noted.

19.19.1 Wait operation and notes

In the wait status, the clock generation block and presettable shift register 1 stop operation, and therefore, serial communication stops.

Serial communication can be executed by releasing the wait status.

To release the wait status, write “1” to the SIO1TS flag.

When “1” is written to the SIO1TS flag, the internal clock is output to the shift clock output pin (when the device is operating as the master), and the presettable shift register 1 and clock counter start operating.

If the shift clock rises when the current value of the clock counter is “8”, the wait status is set. At this time, the SIO1TS flag is automatically reset to 0.

By detecting the contents of the SIO1TS flag when the wait status has been released, the operation status of serial communication can be checked.

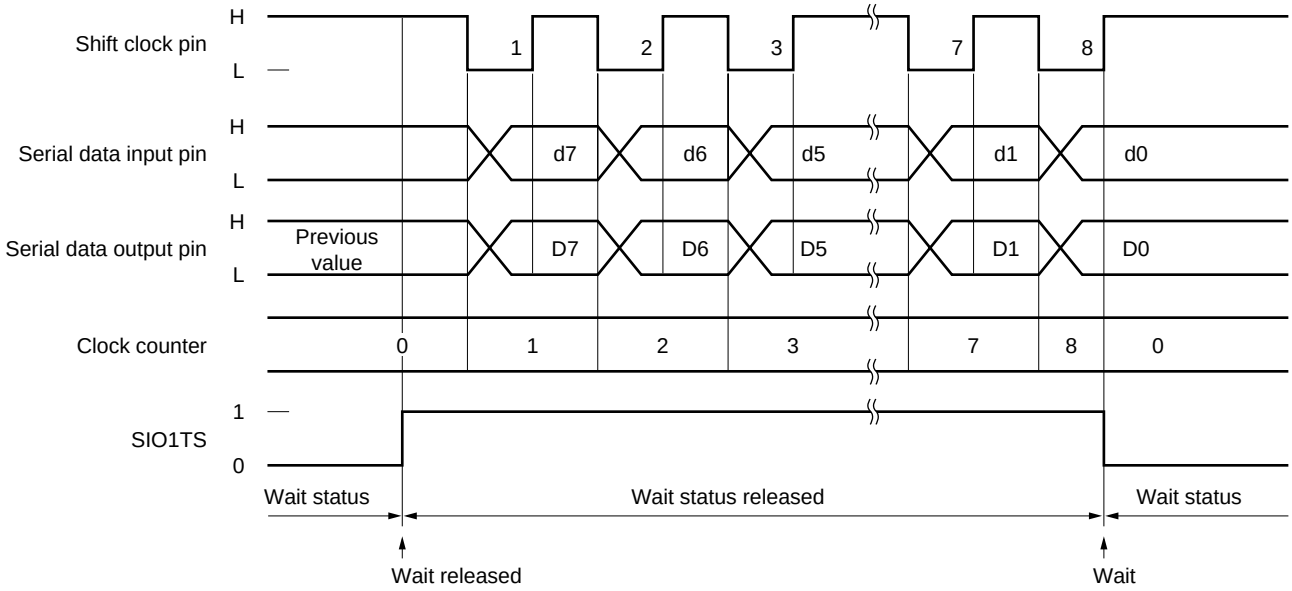
Therefore, by writing “1” to the SIO1TS flag and then detecting “0” of the SIO1TS flag after serial communication has been started, data is read or set.

If data is set to the presettable shift register 1 (by using the PUT instruction) or data is read (by using the GET instruction) while the wait status is released, the correct data may not be set or read. For details, refer to **19.18.3 Notes on setting and reading data**.

If “0” is written to the SIO1TS flag while the wait status is released, the wait status is set. This is called “forced wait status”. If the forced wait status is set, the clock counter is reset to “0”.

Figure 19-20 shows an example of the wait operation.

Figure 19-20. Example of Wait Operation



When the wait status is released, serial data is output at the falling edge of the next clock, and the wait status is set.

When eight pulses of the serial clock have been input, the shift clock pin outputs a high level, and the clock counter and presettable shift register 1 stop operating.

If data is written to or read from the presettable shift register 1 while the wait status is released and the shift clock pin is high, the correct data is not set.

If data is written to the presettable shift register while the wait status is released and the shift clock pin is low, the contents of the MSB are output to the serial data output pin as soon as the “PUT” instruction has been executed.

If the forced wait status is set while the wait status is released, the wait status is set immediately when “0” has been written to the SIO1TS flag, and the clock counter is reset to “0”.

19.20 Using Serial Interface 1

Figure 19-21 shows the I/O block and communication method of serial interface 1.

Table 19-12 shows the operation in each mode of serial interface 1.

As shown in Figure 19-21 and Table 19-12, serial interface 1 can operate on an internal clock (master) or external clock (slave), and can perform reception or transmission.

The master or slave operation is selected by the SIO1CK1 and SIO1CK0 flags, and the reception or transmission is selected by the SIO1HIZ flag.

During the master operation, the internal shift clock is output from the $P0B_2/\overline{SCK_1}$ pin. However, the $P0B_2/\overline{SCK_1}$ pin must be set in the output port mode (P0BBIO2 flag = 1).

During the slave operation, the $P0B_2/\overline{SCK_1}$ pin is floated, and the device waits for the external clock. However, the $P0B_2/\overline{SCK_1}$ pin must be set in the input port mode (P0BBIO2 flag = 0).

Serial data is output from the $P0B_1/SO_1$ pin at the falling edge of the shift clock, regardless of whether the internal clock or external clock is selected, when serial data is output. However, the $P0B_1/SO_1$ pin must be set in the output mode (P0BBIO1 flag = 1), and the SIO1HIZ flag must be set.

The status of the $P0B_0/SI_1$ pin is input to the presettable shift register 1 at the rising edge of the shift clock, regardless of whether the internal clock or external clock is selected, when serial data is input. However, the $P0B_0/SI_1$ pin must be set in the input port mode (P0BBIO0 flag = 0).

If the value output to the $P0B_2/\overline{SCK_1}$ pin is read, the “status of the output latch at that time” is read in the wait status, and the “status of the pin at that time” is read when the wait status is released.

When the value output to the $P0B_1/SO_1$ pin is read, the “status of the output latch at that time” is read.

Paragraphs (1) through (4) below Table 19-12 show program examples for transmission and reception during master and slave operations.

Figure 19-21. I/O Block and Communication Method of Serial Interface 1 (1/2)

I/O block

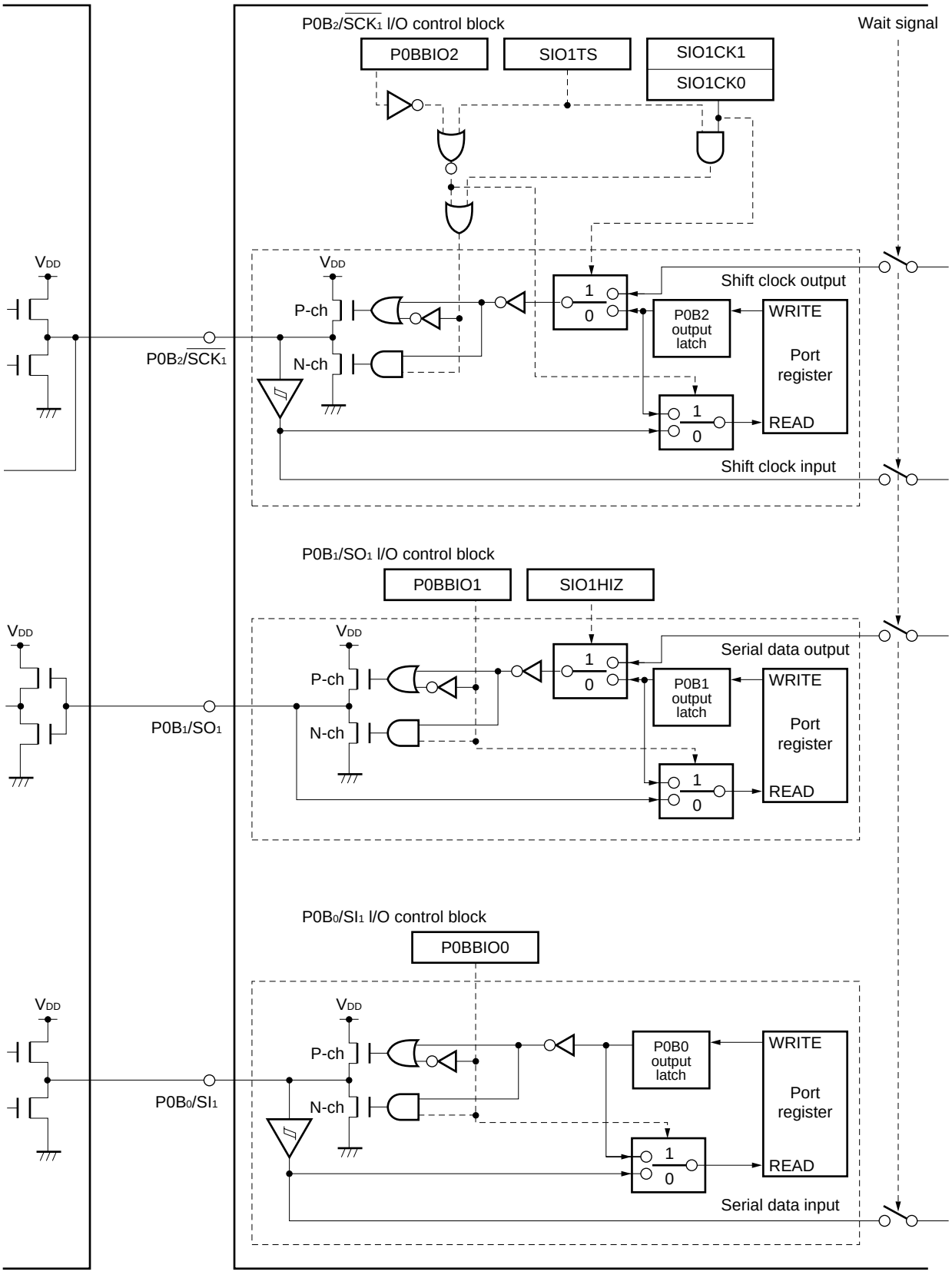


Figure 19-21. I/O Block and Communication Method of Serial Interface 1 (2/2)

Communication method

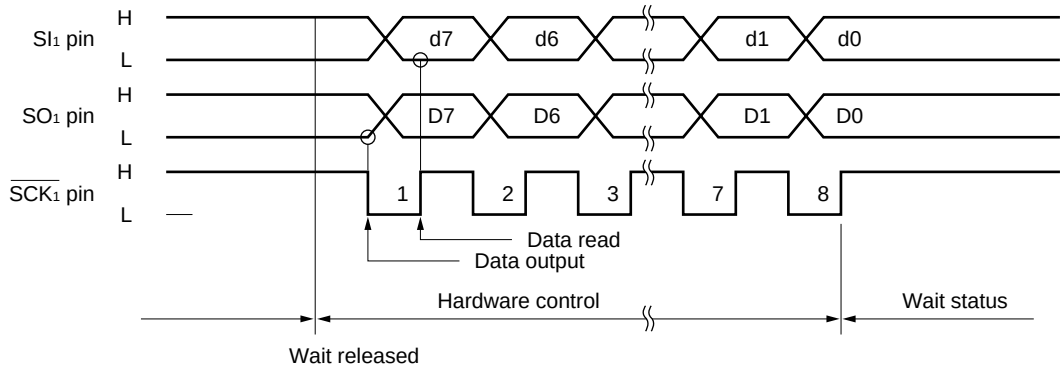


Table 19-12. Operation of Serial Interface 1 in Each Mode

Item \ Operation Mode		3-line Serial I/O Mode			
		Slave operation SIO1CK1=SIO1CK0=0		Master operation Other than SIO1CK1=SIO1CK0=0	
Setting status of each pin	P0B2/$\overline{SCK_1}$	In wait status	When wait released	In wait status	When wait released
		When P0BBIO2 = 0 Floating General-purpose input port **When P0BBIO2 = 1** General-purpose output port Outputs contents of output latch. Normally, P0BBIO2 is reset to 0.	**When P0BBIO2 = 0** Floating External clock input **When P0BBIO2 = 1** General-purpose output port Outputs contents of output latch.	**When P0BBIO2 = 0** Floating General-purpose input port **When P0BBIO2 = 1** Outputs high level. Normally, P0BBIO2 is set to 1.	**When P0BBIO2 = 0** Floating General-purpose input port **When P0BBIO2 = 1** Outputs internal clock
	P0B1/SO1	**When SIO1HIZ = 0**	**When SIO1HIZ = 1**	**When SIO1HIZ = 0**	**When SIO1HIZ = 1**
		When P0BBIO1 = 0 General-purpose input port Floating **When P0BBIO1 = 1** General-purpose output port Outputs contents of output latch.	**When P0BBIO1 = 0** General-purpose input port Floating **When P0BBIO1 = 1** Outputs serial data.	**When P0BBIO1 = 0** General-purpose input port Floating **When P0BBIO1 = 1** General-purpose output port Outputs contents of output latch.	**When P0BBIO1 = 0** General-purpose input port Floating **When P0BBIO1 = 1** Outputs serial data.
	P0B0/SI1	**When P0BBIO0 = 0** Floating Waits for input of external data **When P0BBIO0 = 1** General-purpose output port Outputs contents of output latch. Normally, P0BBIO0 is reset to 0.			
Clock counter operation		Incremented at falling edge of $\overline{SCK_1}$ pin			
Operation of presettable shift register 1 (SIO1SFR)		**Output** When SIO1HIZ = 1 Shifts data from MSB at falling edge of $\overline{SCK_1}$ pin and outputs it from SO_1 pin. When SIO1HIZ = 0 Does not output data. **Input** Shifts data of SI_1 pin from LSB and inputs it at rising edge of $\overline{SCK_1}$ pin regardless of P0BBIO0. However, contents of output latch are output to SI_1 pin when P0BBIO0 = 1.			
Wait operation		Serial communication is started when “1” is written to SIO1TS. SIO1TS is reset to “0” at rising edge of shift clock when clock counter value is “8”. Refer to above for operation of each pin.			

(1) Program example of serial interface 1 (master transmission mode)

Example To transmit 2-byte data "A596H"

```
SCK1BIO FLG      P0BBIO2
SO1BIO  FLG      P0BBIO1

        MOV      DBF1, #0AH              ; Sets transmit data
        MOV      DBF0, #5
        PUT      SIO1SFR, DBF
        SET2     SCK1BIO, SO1BIO
        INITFLG  SIO1TS, SIO1HIZ, NOT SIO1CK1, SIO1CK0
                                         ; Releases wait, serial output
                                         ; Master (fSC = 37.5 kHz)
LOOP1:
        SKF1     SIO1TS                  ; Waits until wait status is released
        BR       LOOP1
        MOV      DBF1, #9                ; Sets transmit data
        MOV      DBF0, #6
        PUT      SIO1SFR, DBF
        SET1     SIO1TS                  ; Releases wait
LOOP2:
        SKF1     SIO1TS                  ; Waits until wait status is released
        BR       LOOP2
        ⋮
```

(2) Program example of serial interface 1 (master reception mode)

Example To receive and store 2-byte data to addresses 00H through 03H of BANK0

```
SCK1BIO FLG      P0BBIO2
SI1BIO  FLG      P0BBIO0

DATA1H  MEM      0.00H       ; Stores higher 4 bits of first byte
DATA1L  MEM      0.01H       ; Stores lower 4 bits of first byte
DATA2H  MEM      0.02H       ; Stores higher 4 bits of second byte
DATA2L  MEM      0.03H       ; Stores lower 4 bits of second byte

        INITFLG  SCK1BIO, NOT SI1BIO
        INITFLG  SIO1TS, NOT SIO1HIZ, SIO1CK1, SIO1CK0
                                 ; Releases wait, no serial output
                                 ; Master (fSC = 450 kHz)
```

```
LOOP1:
        SKF1    SIO1TS                  ; Waits until wait status is released
        BR      LOOP1
        GET     DBF, SIO1SFR            ; Reads receive data
        ST      DATA1H, DBF1            ; Stores read data
        ST      DATA1L, DBF0
        SET1    SIO1TS                  ; Releases wait
LOOP2:
        SKF1    SIO1TS                  ; Wait until wait status is released
        BR      LOOP2
        GET     DBF, SIO1SFR            ; Reads receive data
        ST      DATA2H, DBF1            ; Stores read data
        ST      DATA2L, DBF0
        ⋮
```

(3) Program example of serial interface 1 (slave transmission mode)

Example To transmit 2-byte data “A596H”

```
        SCK1BIO FLG     P0BBIO2
        SO1BIO  FLG     P0BBIO1

        INITFLG NOT SCK1BIO, SO1BIO
        MOV     DBF1, #0AH              ; Sets transmit data
        MOV     DBF0, #5
        PUT     SIO1SFR, DBF
        INITFLG SIO1TS, SIO1HIZ, NOT SIO1CK1, NOT SIO1CK0
                                        ; Releases wait, serial output, slave
LOOP1:
        SKF1    SIO1TS                  ; Waits until wait status is released
        BR      LOOP1
        MOV     DBF1, #9                ; Sets transmit data
        MOV     DBF0, #6
        PUT     SIO1SFR, DBF
        SET1    SIO1TS                  ; Releases wait
LOOP2
        SKF1    SIO1TS                  ; Waits until wait status is released
        BR      LOOP2
        ⋮
```

(4) Program example of serial interface 1 (slave reception mode)

Example To receive and store 2-byte data to addresses 00H through 03H of BANK0

```
        SCK1BIO FLG     P0BBIO2
        SI1BIO  FLG     P0BBIO0

        DATA1H  MEM     0.00H       ; Stores higher 4 bits of first byte
        DATA1L  MEM     0.01H       ; Stores lower 4 bits of first byte
        DATA2H  MEM     0.02H       ; Stores higher 4 bits of second byte
        DATA2L  MEM     0.03H       ; Stores lower 4 bits of second byte

        CLR2    SCK1BIO, SI1BIO
        INITFLG SIO1TS, NOT SIO1HIZ, NOT SIO1CK1, NOT SIO1CK0
                                    ; Releases wait, no serial output, slave
LOOP1:
        SKF1    SIO1TS              ; Waits until wait status is released
        BR      LOOP1
        GET     DBF, SIO1SFR        ; Reads receive data
        ST      DATA1H, DBF1        ; Stores read data
        ST      DATA1L, DBF0
        SET1    SIO2TS              ; Releases wait
LOOP2
        SKF1    SIO1TS              ; Waits until wait status is released
        BR      LOOP2
        GET     DBF, SIO1SFR        ; Reads receive data
        ST      DATA2H, DBF1        ; Stores read data
        ST      DATA2L, DBF0
        ⋮
```

19.21 Reset Status of Serial Interface 1

19.21.1 At power-ON reset

All the $P0B_2/\overline{SCK_1}$ through $P0B_0/SI_1$ pins are set in the general-purpose input port mode (floating output).

The value of the presettable shift register 1 is undefined.

19.21.2 On execution of clock stop instruction

All the $P0B_2/\overline{SCK_1}$ through $P0B_0/SI_1$ pins are set in the general-purpose input port mode (floating output).

The presettable shift register 1 retains the previous value.

19.21.3 At CE reset

All the $P0B_2/\overline{SCK_1}$ through $P0B_0/SI_1$ pins are set in the general-purpose input port mode (floating output).

The presettable shift register 1 retains the previous value.

19.21.4 In halt status

The I/O pins retain the current status.

If the internal clock is used (master operation) at this time, the clock is not output when the “HALT” instruction has been executed.

Therefore, the “HALT” instruction must be executed after communication has been completed when the internal clock is used.

If an external clock is forcibly input, the serial interface 1 operates even when the internal clock is set.

When the external clock is used (slave operation), the operation continues even when the “HALT” instruction is executed.

20. FREQUENCY COUNTER (FC)

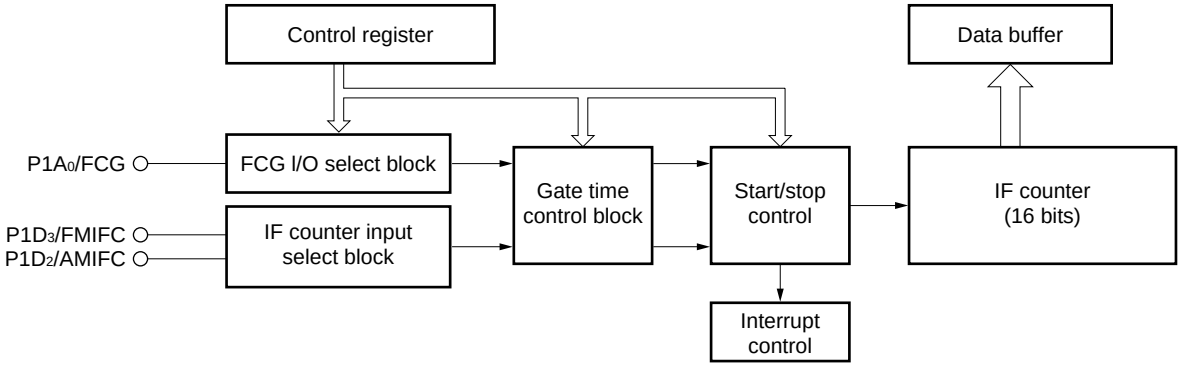
The frequency counter (FC) is used to measure the intermediate frequency (IF) of a tuner or to detect the pulse width of an external signal.

20.1 Configuration of Frequency Counter

Figure 20-1 shows the block diagram of the frequency counter.

As shown in this figure, the frequency counter consists of an FCG I/O select block, an IF counter input select block, a gate time control block, a start/stop control block, and a count block.

Figure 20-1. Block Diagram of Frequency Counter



20.2 Functional Outline of IF Counter

The frequency counter has an IF count function to count the frequency of an externally input signal and an external gate counter (FCG: Frequency for external Gate signal) to detect the pulse width of an externally input signal.

The IF counter function counts the frequency input to the $P1D_3$/FMIFC or $P1D_2$/AMIFC pin for a fixed time (1 ms, 4 ms, 8 ms, or open) with a 16-bit counter.

The external gate counter (FCG) function counts the frequency of the internal clock (1 kHz, 100 kHz, or 900 kHz) from a rising edge of the signal applied to the $P1A_0$/FCG pin to the next rising edge by using a 16-bit counter.

For the details of the IF counter and external gate functions, refer to **20.5** and **20.6**, respectively.

Because the frequency counter shares the hardware with the clock generator port described in **18. CLOCK GENERATOR PORT (CGP)**, the frequency counter and clock generator port cannot be used at the same time. For details, refer to **20.8 Notes on Using Frequency Counter**.

20.2.1 IF counter input select block and FCG I/O select block

The IF counter input select block selects whether the $P1D_3$/FMIFC and $P1D_2$/AMIFC pin is used as general-purpose input port pins or IF counter pins.

The FCG I/O select block selects whether the $P1A_0$/FCG pin is used as a general-purpose I/O port pin or external gate counter pin.

Selection of the general-purpose port function, IF counter function, or external gate function is made by using the IF counter mode select register (IFCMODE: RF address 12H).

For details, refer to **20.3**.

20.2.2 Gate time control block

The gate time control block controls the time during which the frequency is counted, by using the IF counter mode select register.

The following paragraphs (1) and (2) outline the operations of the IF counter function an external gate counter function.

For details, refer to **20.3**.

(1) IF counter function

This function is to set the internal gate time (1 ms, 4 ms, 8 ms, or open) to count the frequency applied to the $P1D_3$/FMIFC or $P1D_2$/AMIFC pin, by using the IF counter mode select register.

(2) External gate counter function

This function is to count the internal frequency (1 kHz, 100 kHz, or 900 kHz) during the external gate time (the time from a rising edge of the signal applied to the $P1A_0$/FCG pin to the next rising edge), by using the IF counter mode select register.

20.2.3 Start/stop control block

The start/stop control block starts or stops the frequency counter by using the IF counter control register (IFCCONT: RF address 23H), IF counter gate open status register (IFCGOSTR: RF address 04H), and IF counter interrupt request register (IREQIFC: RF address 3AH).

When the IF counter function is used, the start/stop control block issues an interrupt request when the internal gate is closed.

For details, refer to **20.4**.

20.2.4 IF counter

The IF counter counts the input frequency when the IF counter function or external gate counter function is used, by using a 16-bit binary counter.

The count value is read by the IF counter data register (IFC: peripheral address 43H) via data buffer.

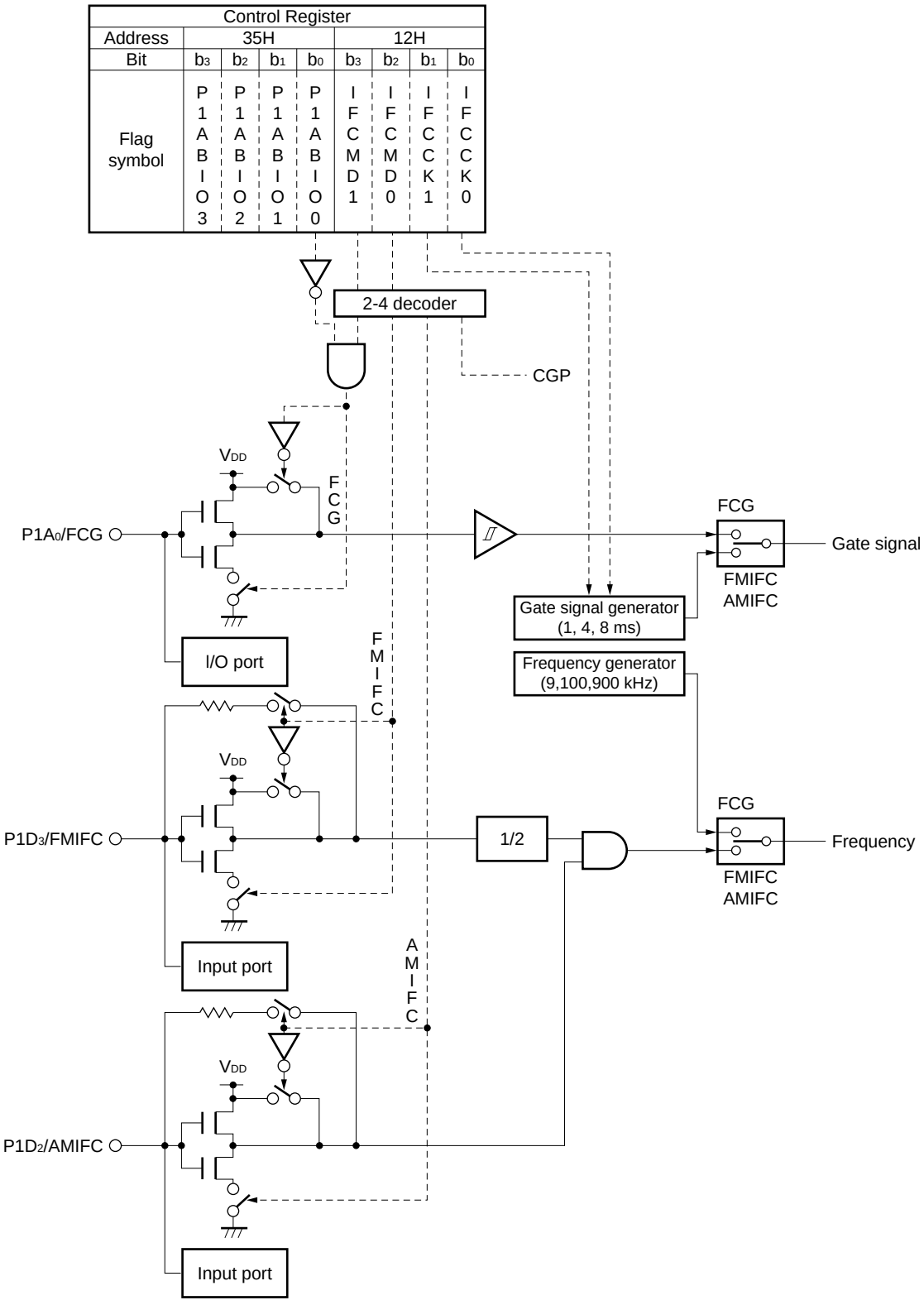
For details, refer to **20.4**.

20.3 I/O Select Block and Gate Time Control Block

20.3.1 Configuration of I/O select block and gate time control block

Figure 20-2 shows the configuration of the IF counter input select, external gate counter I/O select, and gate time control blocks.

Figure 20-2. Configuration of I/O Select Block and Gate Time Control Block



20.3.2 Function of I/O select block

The I/O select block selects whether each pin is used as a general-purpose I/O port pin or frequency counter pin.

The selection is made by using the IFCMD1 and IFCMD0 flags of the IF counter mode select register (refer to **20.3.4**).

To use the $P1A_0$/FCG pin as an external gate counter pin, the P1ABIO0 flag of the port 1A bit I/O register must be reset to “0”.

This is because the $P1A_0$/FCG pin functions as a general-purpose output port pin if the P1ABIO0 flag is set to “1”, even when the external gate counter function is selected by the IFCMD1 and IFCMD0 flags.

20.3.3 Function of gate time control block

The gate time control block sets the gate time (count time) when the IF counter function is used and the count frequency when the external gate counter function is used.

The gate time and count frequency are set by the IFCCK1 and IFCCK0 flags of the IF counter mode select register (refer to **20.3.4**).

20.3.4 Configuration and function of IF counter mode select register (IFCMODE)

The IF counter mode select register selects the IF counter function or external gate counter function.

The configuration and function of this register are illustrated below.

Because the frequency counter is multiplexed with the clock generator port, this register can also select the clock generator port function.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
IF counter mode select register (IFCMODE)	IFCMD1	IFCMD0	IFCCK1	IFCCK0	12H	R/W

Sets gate time of IF counter and reference frequency of external gate counter		
IFCCK1	IFCCK0	Gate time of IF counter	Reference frequency of external gate counter
0	0	1 ms	1 kHz
0	1	4 ms	100 kHz
1	0	8 ms	900 kHz
1	1	Open	0 kHz

Selects functions of IF counter, external gate counter (FCG), or clock generator port (CGP)	
IFCMD1	IFCMD0	
0	0	Clock generator port (CGP)
0	1	IF counter (FMIFC)
1	0	IF counter (AMIFC)
1	1	External gate counter (FCG)

		IFCMD1	IFCMD0	IFCCK1	IFCCK0
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	Retained			

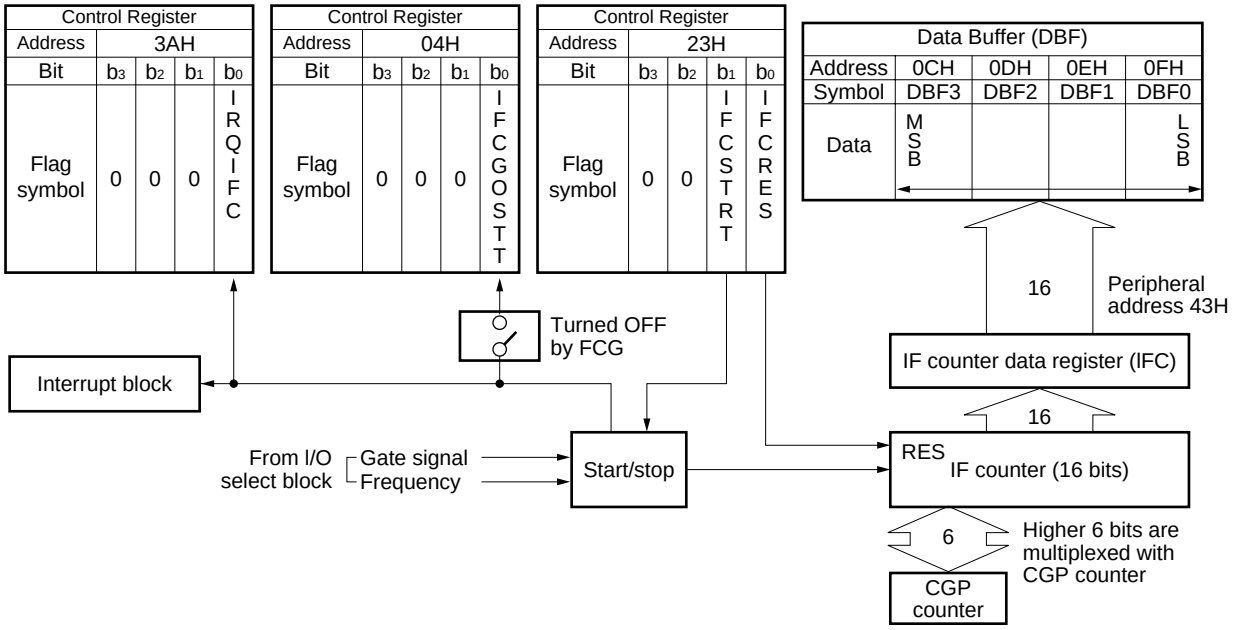
The IF counter, external gate counter, and clock generator port functions cannot be used at the same time.

20.4 Start/Stop Control Block and IF Counter

20.4.1 Configuration of start/stop control block and counter

Figure 20-3 shows the configuration of the start/stop control block and counter.

Figure 20-3. Configuration of Start/Stop Control Block and Counter



20.4.2 Function of start/stop control block

The start/stop control block starts or stops counting of the frequency counter.

The counter is started by the IFCSTRT flag of the IF counter control register.

It is stopped by the IFCGOSTT flag of the IF counter gate open status register or the IRQIFC flag of the IF counter interrupt request register.

Note, however, that the stop of the counter cannot be detected by the IFCGOSTT flag when the the external gate counter function is used.

The following **20.4.3** and **20.4.4** describe the operations when the IF counter function and external gate function are selected.

20.4.7 and **20.4.8** describe the configuration and function of the IF counter control register and IF counter gate open status register.

20.4.3 Gate operation of IF counter function

(1) When 1, 4, or 8 ms of gate time is selected

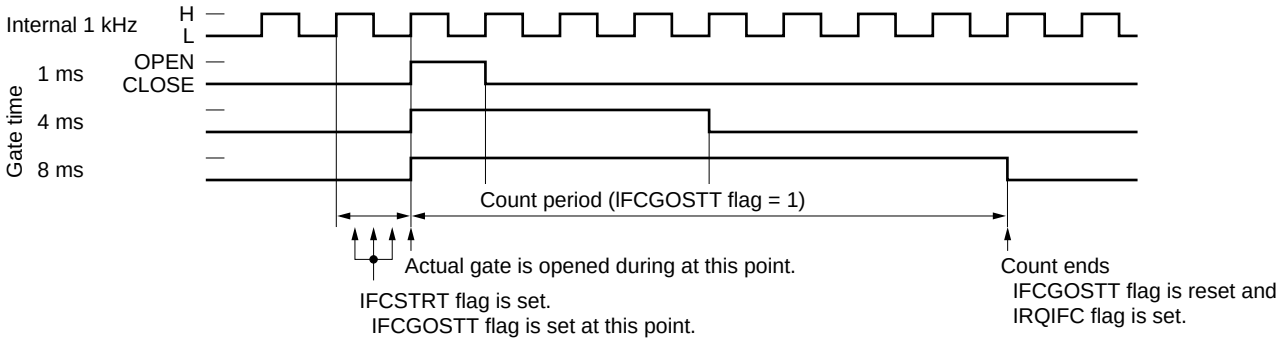
The gate is opened for 1, 4, or 8 ms starting from the rising edge of the internal 1-kHz signal after the IFCSTRT flag has been set to 1, as illustrated below.

While this gate is open, the frequency input from the specified pin is counted by the 16-bit counter.

When the gate is closed, the IFCGOSTT flag is reset, and IRQIFC flag is set.

The IFCGOSTT flag is automatically set to 1 when the IFCSTRT flag is set.

The IRQIFC flag is reset when an interrupt has been accepted or when “0” has been written to it.

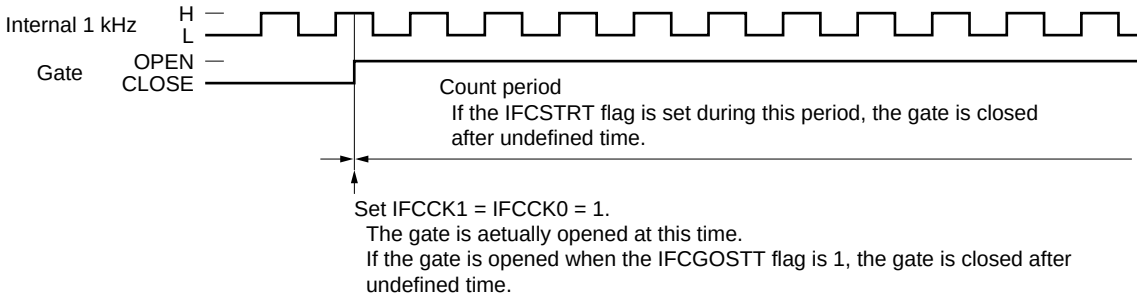


(2) When "open" is selected as gate time

When "open" is selected as the gate time by the IFCCK1 and IFCCK0 flags, the gate is opened as illustrated below.

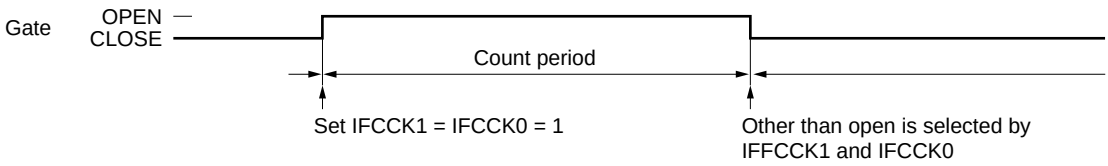
If the counter is started by the IFCSTRT flag while the gate is open, the gate is closed after undefined time. Therefore, do not set the IFCSTRT flag to 1 when the gate is opened.

However, the counter can be reset by the IFCRES flag.



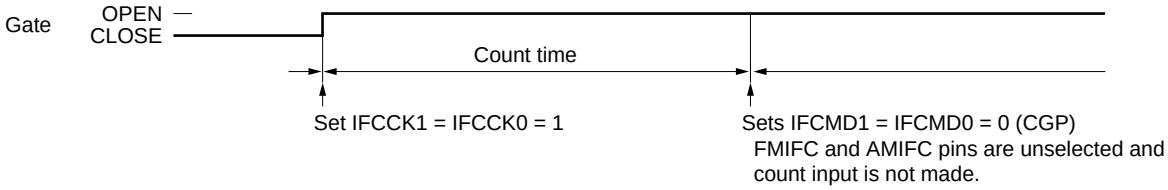
If "open" is selected as the gate time, the gate is opened or closed in two ways as illustrated in (a) and (b) below.

(a) Gate time is set to other than open by IFCCK1 and IFCCK0 flags



(b) Pin selected by IFCMD1 and IFCMD0 flags is unselected

The gate remains open, and counting is stopped by disabling input from the pin.



20.4.4 Gate operation of external gate counter (FCG)

The gate is opened starting from the rising edge of the signal input to the pin to the next rising edge after the IFCSTRT flag has been set, as shown below.

While the gate is open, the internal frequency (1 kHz, 100 kHz, or 900 kHz) is counted by the 16-bit counter.

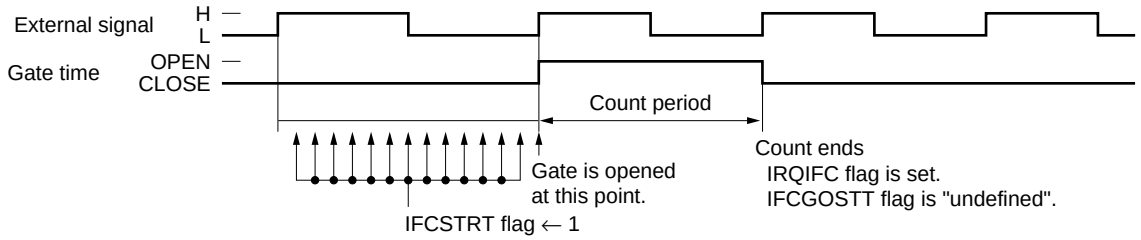
When the gate is closed, the IRQIFC flag is set.

The IRQIFC flag is reset when an interrupt is accepted or when “0” is written to it.

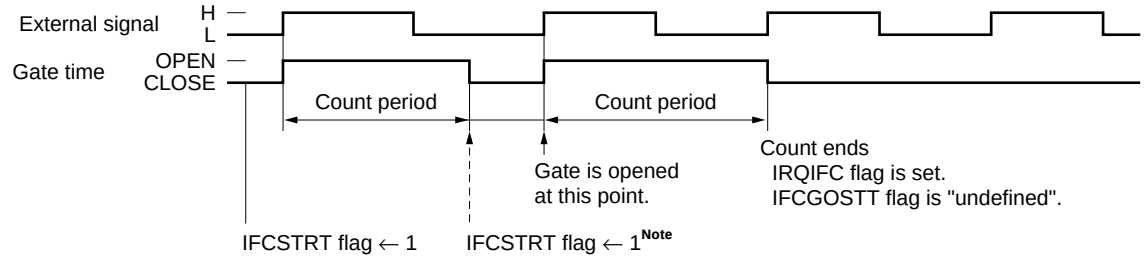
Even if the IFCSTRT flag is set, the IRQIFC flag is not automatically reset. This flag therefore must be reset by program when the counter is started.

The IFCGOSTT flag is automatically set to 1 when the IFCSTRT flag is set, but is not reset when the gate is closed.

In other words, opening or closing of the gate cannot be detected by the IFCGOSTT flag when the external gate counter function is used.



When counter is reset and started while gate is open



Note If the IFCRES flag is set at this point, the IRQIFC flag is reset to “0”.

20.4.5 Function and operation of 16-bit counter

The 16-bit counter counts up the frequency input during gate time.

This counter is reset by writing "1" to the IFCRES flag of the IF counter control register.

When the 16-bit counter counts up to FFFFH, it is reset to 0000H and continues counting.

Because the higher 6 bits of this counter are multiplexed with the clock generator port function, the frequency counter and clock generator port cannot be used at the same time.

The following paragraphs (1) and (2) describe the operations of the IF counter function and external gate counter function.

The value of the IF counter data register is read via data buffer.

20.4.6 describes the configuration and function of the IF counter data register.

(1) When IF counter function is used

The 16-bit counter counts the frequency input to the $P1D_3$/FMIFC or $P1D_2$/AMIFC pin while the gate is open.

Note, however, that the frequency input to the $P1D_3$/FMIFC pin is divided by two.

The relation between count value "x (HEX)" and input frequencies (f_{FMIFC} and f_{AMIFC}) is shown below.

FMIFC

$$f_{FMIFC} = \frac{x(DEC)}{t_{GATE}} \times 2 \text{ (kHz)}$$

t_{GATE}: gate time (1 ms, 4 ms, 8 ms)

AMIFC

$$f_{AMIFC} = \frac{x(DEC)}{t_{GATE}} \text{ (kHz)}$$

t_{GATE}: gate time (1 ms, 4 ms, 8 ms)

(2) When external gate function is used

The 16-bit counter counts the internal frequency while the gate is opened by the signal input to the $P1A_0$/FCG pin.

The relation between count value "x (HEX)" and gate width t_{GATE} of the input signal is shown below.

$$t_{GATE} = \frac{x(DEC)}{f_r} \text{ (ms)}$$

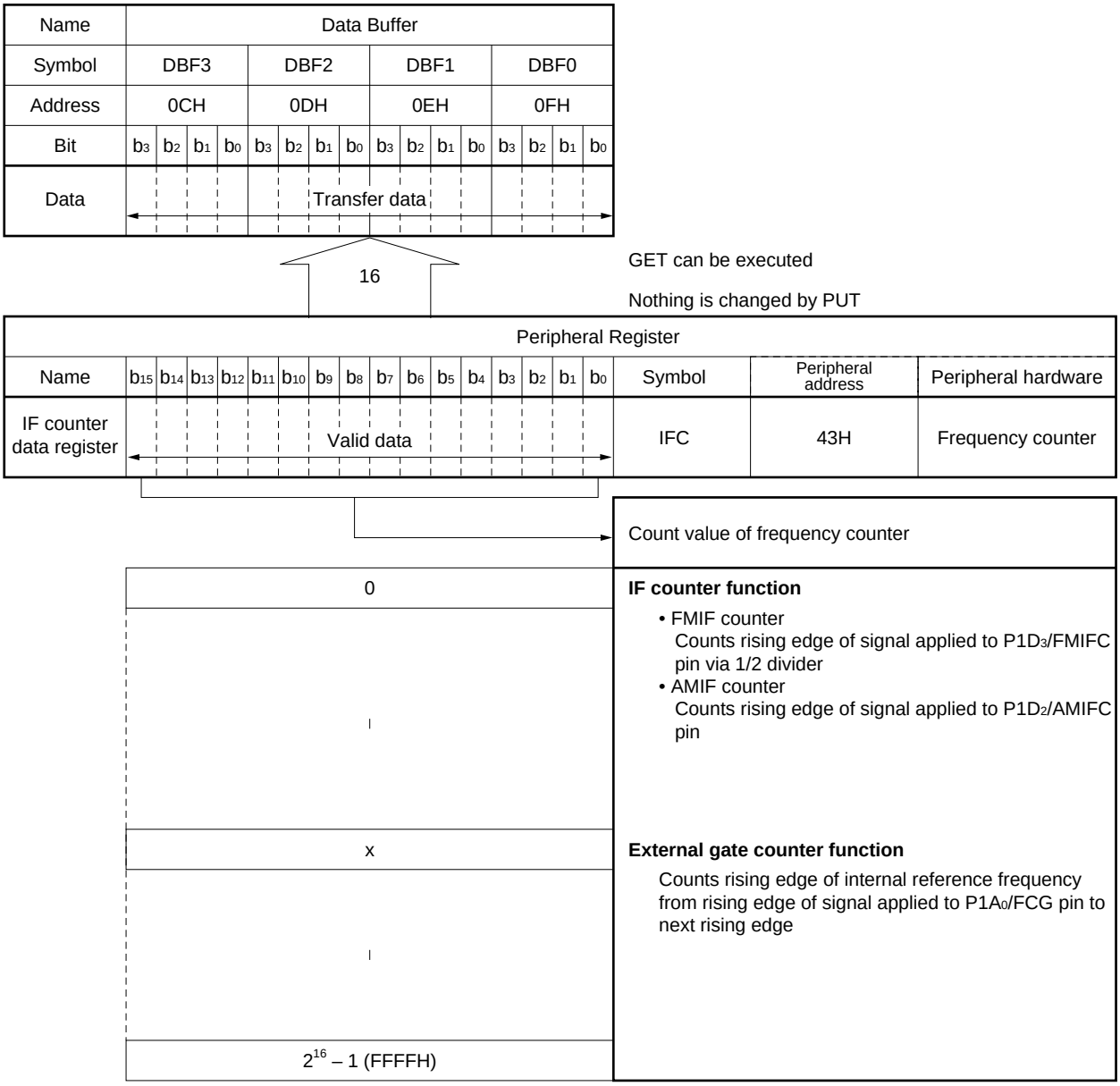
f_r: internal frequency (1 kHz, 100 kHz, 900 kHz)

20.4.6 Configuration and function of IF counter data register (IFC)

The configuration and function of the IF counter data register are illustrated below.

The IF counter data register reads the count value of the frequency counter.

The IF counter data register counts up to FFFFH, then is reset to 0000H on the next input and continues counting.



The higher 6 bits of the IF counter are multiplexed with the CGP counter.

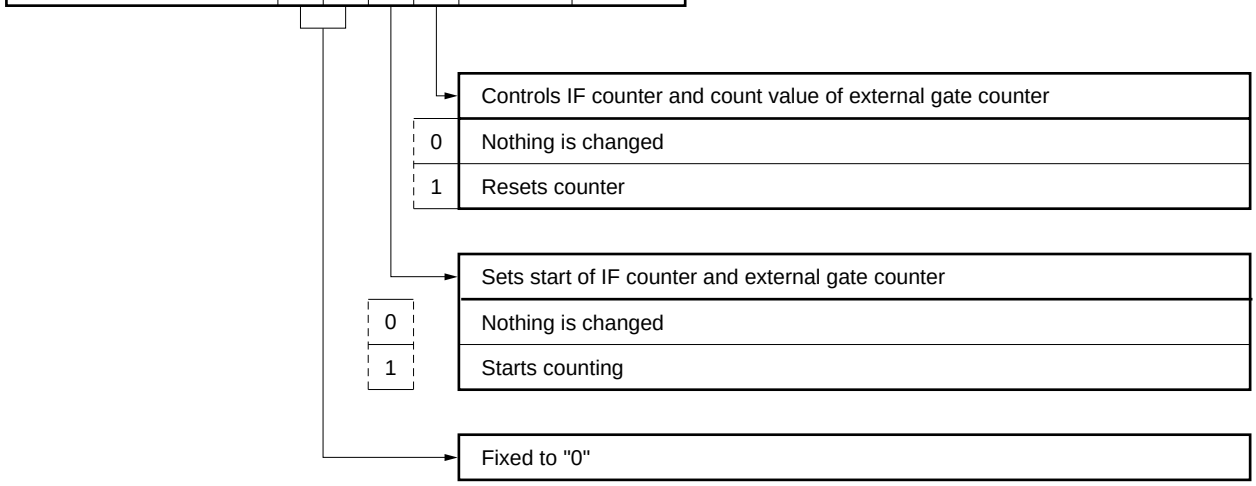
Therefore, the frequency counter function and clock generator port function cannot be used at the same time.

For details, refer to **20.8 Notes on Using Frequency Counter**.

20.4.7 Configuration and function of IF counter control register (IFCCONT)

The IF counter control register starts the frequency counter function (IF counter and external gate counter) and resets the 16-bit counter.

Name	Flag Symbol b_3	b_2	b_1	b_0	Address	Read/ Write
IF counter control register (IFCCONT)	0	0	IFCSTRT	IFCRES	23H	R/W



		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop			0	0
	CE			Retained	

The IF counter is controlled by writing the contents of the window register to it by using the “POKE” instruction. When the contents of this register are read to the window register by the “PEEK” instruction, “0” is read.

20.4.8 Configuration and function of IF counter gate open status register (IFCGOSTR)

This register detects the opening and closing of the gate when the IF counter function is used.

The closing of the gate cannot be detected when the external gate counter function is used.

The configuration and function of this register are illustrated below.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
IF counter gate open status register (IFCGOSTR)	0	0	0	IFCGOSTT	04H	R

IFCGOSTT	Detects opening/closing of gate of reference counter	
	When IF counter function is used	When external gate counter function is used
0	Set to 1 since "1" is set to the IFCSTRT flag until gate time is out	Reset to 0 after fixed time (gate time of IF counter function) regardless of input to $P1A_0$/FCG pin
1		

Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	↓	↓	↓	–
	CE	↓	↓	↓	–

When the IFCGOSTT flag is set to 1 (when the gate is open), do not read the contents of the IF counter data register (IFC) to the data buffer.

The gate of the IF external gate counter function cannot be opened or closed by the IFCGOSTT flag. Open or close the gate of the external gate counter by using the IRQIFC flag.

20.5 Using IF Counter Function

The following subsections **20.5.1** through **20.5.3** describe how to use the hardware of the IF counter, program example, and count error.

20.5.1 Using hardware of IF counter

Figure 20-4 shows the block diagram when the $P1D_3$/FMIFC and $P1D_2$/AMIFC pins are used.

Table 20-1 shows the range of the frequencies that can be input to the $P1D_3$/FMIFC and $P1D_2$/AMIFC pins.

As shown in figure 20-4, the IF counter has an input pin provided with an AC amplifier. Cut off the DC component of the input signal by using capacitor C.

When the $P1D_3$/FMIFC and $P1D_2$/AMIFC pins are used for the IF counter function, switch SW turns ON, and the voltage on each pin drops to about 1/2 V_{DD}.

If the voltage has not risen to the sufficient intermediate level at this time, the AC amplifier does not operate correctly, and IF counting cannot correctly be performed.

Therefore, make sure that a sufficient wait time elapses after each pin has been specified to be used for the IF counter until the counter is started.

Figure 20-4. IF Count Function of Each Pin

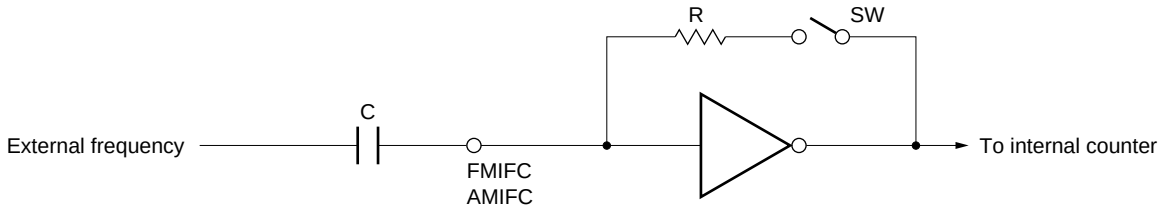


Table 20-1. IF Counter Input Frequency Range

Input Pin	Input Frequency (MHz)	Input Amplitude (V_{P-P})
$P1D_3$/FMIFC	5-15	0.3
	10.5-10.9	0.06
$P1D_2$/AMIFC	0.1-1	0.3
	0.44-0.46	0.05

20.5.2 Program example of IF counter function

This subsection presents a program example of the IF counter function.

As shown in the example below, a wait time must elapse since an instruction that specifies the P1D3/FMICF or P1D2/AMIFC pin as the IF counter pin until the counter is started.

This is because the internal AC amplifier does not operate normally as soon as the pin has been selected for the IF counter function, as described in **20.5.1**.

Example To count frequency of P1D3/FMIFC pin (gate time: 8 ms)

```
        INITFLG NOT IFCMD1, IFCMD0, IFCCK1, NOT IFCCK0
                                    ; Selects FMIFC pin and sets gate time to 8 ms
            Wait                    ; Internal AC amplifier stabilization time
        SET2  IFCRES, IFCSTRT       ; Resets and starts counter
    LOOP
        SKT1      IFCGOSTT          ; Detects opening/closing of gate
        BR        READ              ; Branches to READ: if gate is closed
            Processing A            ; Do not read data of IF counter during this processing A
                                    ; Do not select CGP function
        BR        LOOP
     READ:
        GET       DBF, IFC          ; Reads value of IF counter data to data buffer.
```

20.5.3 Error of IF counter

Errors of the IF counter include an error of the gate time and a count error.

Each error is described in (1) and (2) below.

(1) Error of gate time

The gate time of the IF counter is created by dividing the system clock frequency of 4.5 MHz.

If 4.5 MHz shifts by “+x” ppm, therefore, the gate time shifts by “–x” ppm.

(2) Count error

The IF counter counts the frequency at the rising edge of the input signal.

If a high level is input to the pin when the gate is open, one excess pulse is counted.

When the gate is closed, however, this excess pulse is not counted depending on the status of the pin.

The counter error, therefore, is “+1, –0”.

20.6 Using External Gate Counter Function

The following **20.6.1** through **20.6.3** describe how to use the hardware of the external gate counter, program example, and count error.

20.6.1 Using external gate counter

A program example of the external gate counter function is shown below.

The external gate counter function is to open or close the gate by using the IRQIFC flag.

An interrupt can be generated by the IRQIFC flag. To not use the interrupt, the contents of the IRQIFC flag can be detected by program.

Example To set internal frequency to 100 kHz (with interrupt used)

```
INTIFC  DAT     0001H           ; Symbol definition of IF counter interrupt vector address
        BR      MAIN
ORG     INTIFC
        GET     DBF, IFC        ; Reads value of IF counter data register to data buffer
        EI
        RETI
MAIN:
        INITFLG IFCMD1, IFCMD0, NOT IFCCK1, IFCCK0
                                ; Selects FCG function and sets internal frequency to 100 kHz
        IFC_RES_AND_START       ; Resets and starts counter
        CLR1    IRQIFC
        SET     IPIFC           ; Enables interrupt by IRQIFC flag
        EI
```

20.6.2 Error of external gate counter

Errors of the the external gate counter include an error of the internal frequency and a count error.

Each error is described in (1) and (2) below.

(1) Error of internal frequency

The internal frequency of the external gate counter is created by dividing the system clock frequency of 4.5 MHz.

If 4.5 MHz shifts by “+x” ppm, therefore, the internal frequency shifts by “–x” ppm.

(2) Count error

The external gate counter counts frequency at the falling edge of the internal frequency.

Therefore, if the internal frequency is low when the gate is open (when the input of the pin rises), one excess pulse is counted.

However, when the gate is closed (when the next input of the pin rises), the frequency is not counted due to the count level of the internal frequency.

Therefore, the count error is “+1, –0”.

20.7 Reset Status

20.7.1 On power-ON reset

The P1D3/FMICF and P1D2/AMIFC pins are set as general-purpose input port pins.

The P1A0/FCG pin is set as a general-purpose I/O port pin.

20.7.2 On execution of clock stop instruction

The P1D3/FMICF and P1D2/AMIFC pins are set as general-purpose input port pins.

The P1A0/FCG pin is set as a general-purpose I/O port pin.

20.7.3 On CE reset

The P1D3/FMIFC, P1D2/AMIFC, and P1A0/FCG pins retain the previous status.

20.7.4 In halt status

The P1D3/FMIFC, P1D2/AMIFC, and P1A0/FCG pins retain the status immediately before the halt status.

When releasing the halt status by using the interrupt of the frequency counter at this time, the following point must be noted.

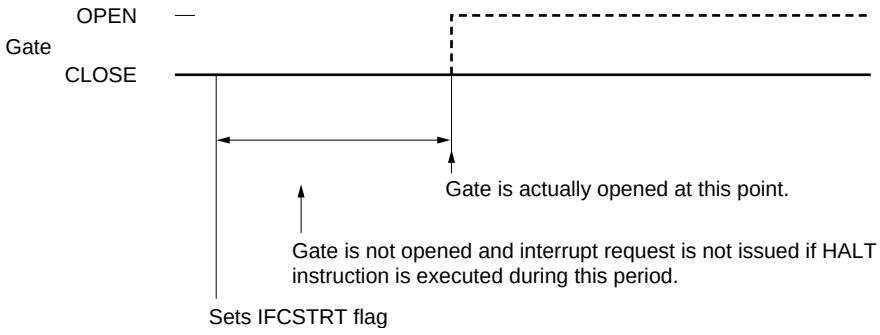
Caution If the "HALT" instruction is executed after counting has been started by the IFCSTRT flag and before the gate is actually opened, the gate is not opened.
When using the IF counter function, therefore, wait for at least 1 ms before executing the "HALT" instruction. When the external gate counter function is used, execute the "HALT" instruction after the P1A0/FCG pin has gone high.

Figure 20-5 illustrates the gate operation when the "HALT" instruction is used.

As shown in this figure, closing of the gate cannot be detected if the gate is not opened. Consequently, the interrupt request is not issued.

If halt release conditions other than the interrupt are not set and if the interrupts other than that of the IF counter are not enabled, the HALT status is not released.

Figure 20-5. Gate Operation When "HALT" Instruction Is Used



20.8 Notes on Using Frequency Counter

The frequency counter shares the hardware with the clock generator port described in the preceding chapter.

Therefore, the clock generator port and frequency counter cannot be used at the same time.

If the data of the IF counter mode select register and IF counter data register are manipulated when the clock generator port is used, the operation described in **20.8.1** is performed.

If the data of the IF counter mode select register and CGP data register (peripheral address 20H) is manipulated when the frequency counter is used, the operation described in **20.8.2** is performed.

20.8.1 When clock generator port is used

(1) When IFCMD1 and IFCMD0 flags of IF counter mode select register are manipulated

If a value other than "0" is written to the IFCMD1 and IFCMD0 flags, the $P1B_0$/CGP pin retains the output level at that time when data is set, and stops the CGP operation.

If the IFCMD1 and IFCMD0 flags are reset to "0" again, the CGP operation is started.

(2) When IF counter data register is manipulated

The CGP operation is not affected even if the IF counter data register is read (GET) or written (PUT).

An "undefined" value is read when the register is read, and nothing is changed when the register is written.

Because the IF counter data register is a read-only peripheral register, do not write data to this register.

If the write instruction "PUT IFC, DBF" is executed, the 17K series assembler (AS17K) generates an error.

20.8.2 When frequency counter is used

(1) When IFCMD1 and IFCMD0 flags of IF counter mode select register are manipulated

If "0" is written to the IFCMD1 and IFCMD0 flags, the $P1B_0$/CGP performs the operation of the CGP data register at that time when the data has been set.

To perform the CGP operation, however, the CGPSEL flag of the PWM mode select register must be set.

If the previous values are set again to the IFCMD1 and IFCMD0 flags, the frequency counter continues operating, but the count value is not accurate.

In other words, the frequency is not counted while the CGP operation is selected.

(2) When CGP data register is manipulated

The frequency counter is not affected even when the CGP data register is read (GET) or written (PUT).

When this register is read, the value set when the CGP function was previously used (if the CGP function was not used, an "undefined value") is read.

When the register is written, the contents of the bits 3 through 1 of the DBF1 and DBF0 are written to the CGP data register.

21. LCD CONTROLLER/DRIVER

The LCD (Liquid Crystal Display) controller/driver can display an LCD of up to 60 dots MAX. by using a combination of segment signals and common signals.

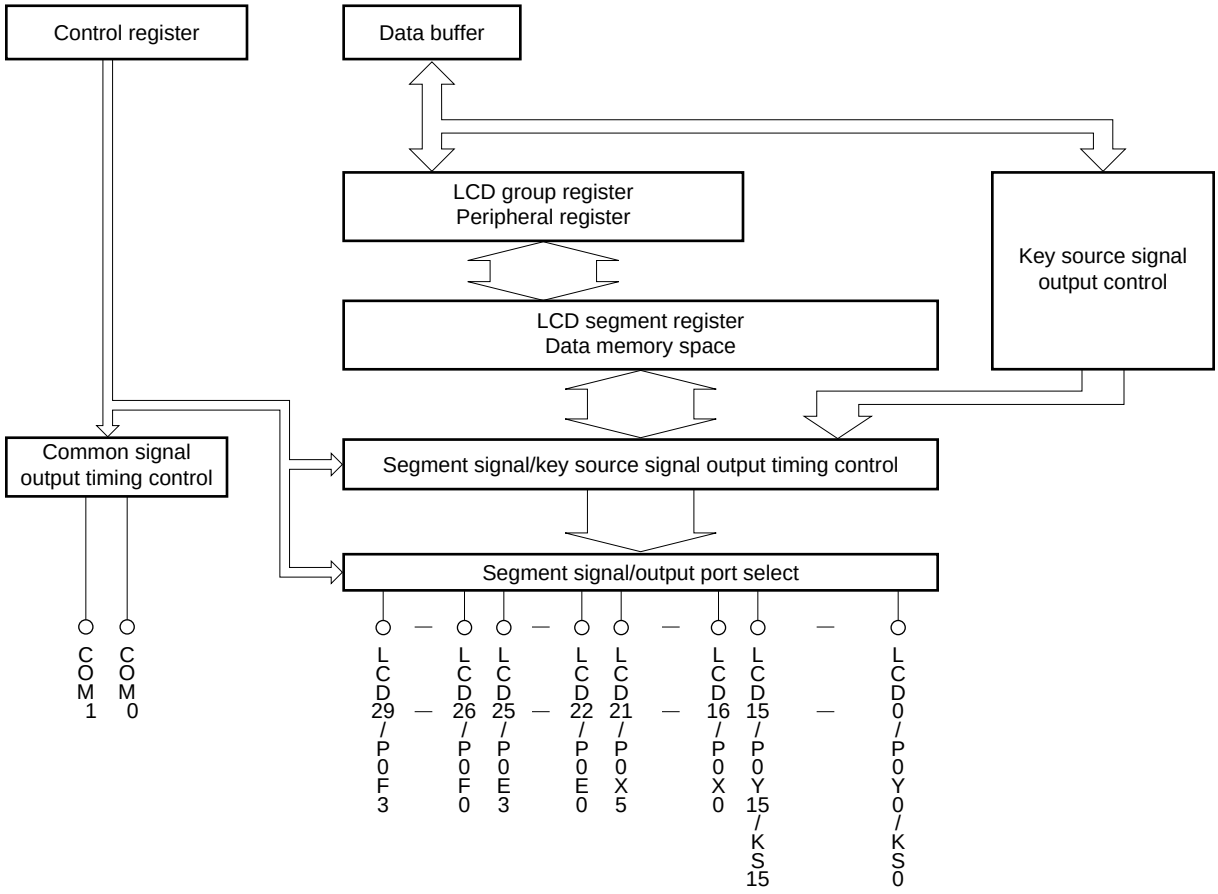
21.1 Configuration of LCD Controller/Driver

Figure 21-1 shows the block diagram of the LCD controller/driver.

As shown in this figure, the LCD controller/driver consists of a common signal output timing control block, a segment signal/key source signal output timing control block, a segment signal/output port select block, an LCD segment register, an LCD group register, and a key source signal output control block.

21.2 outlines the function of each block.

Figure 21-1. Block Diagram of LCD Controller/Driver



21.2 Functional Outline of LCD Controller/Driver

The LCD controller/driver can display an LCD of up to 60 dots MAX. by using a combination of common signal output pins (COM_1 and COM_0 pins) and segment signal output pins (LCD_{29}/$P0F_3$ through LCD_0/$P0Y_0$/KS_0 pins).

Figure 21-2 shows the relation among the common signal output pins, segment signal output pins, and display dots.

As shown in this figure, two dots, which are the intersections with the COM_1 and COM_0 pins, can be displayed per segment line.

The drive mode is 1/2 duty, 1/2 bias, and the drive voltage is supply voltage V_{DD}.

The segment signal output pins (LCD_{29}/$P0F_3$ through LCD_0/$P0Y_0$/KS_0) can be also used as general-purpose output port pins.

When they are used as general-purpose port pins, port 0F (LCD_{29}/$P0F_3$ through LCD_{26}/$P0F_0$ pins), port 0E (LCD_{25}/$P0E_3$ through LCD_{22}/$P0E_0$ pins), port 0X (LCD_{21}/$P0X_5$ through LCD_{16}/$P0X_0$ pins), and port 0Y (LCD_{15}/$P0Y_{15}$/KS_{15} through LCD_0/$P0Y_0$/KS_0 pins) can be independently used.

Of the segment signal output pins, the LCD_{15}/$P0Y_{15}$/KS_{15} through LCD_0/$P0Y_0$/KS_0 pins can be used as key source signal output pins.

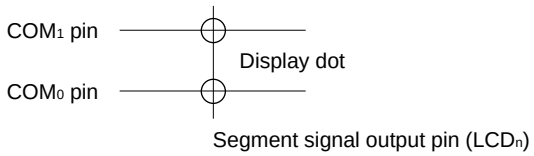
The key source signal output pins are multiplexed with the LCD segment output pins by means of time division.

For the details of the general-purpose output ports, refer to **15. GENERAL-PURPOSE PORTS**.

For the details of the key source signal output, refer to **22. KEY SOURCE CONTROLLER/DECODER**.

The following **21.2.1** through **21.2.6** outline the function of each block of the LCD controller/driver.

Figure 21-2. Common Signal Output, Segment Signal Output, and Display Dot



21.2.1 LCD segment register

The LCD segment register sets the dot data on the LCD that is illuminated or extinguished.

Because this register is located on the data memory, it can be controlled by any data memory manipulation instruction.

When the segment signal output pins are used as general-purpose output port pins, this register sets output data.

For details, refer to **21.3**.

21.2.2 LCD group register

The LCD group register sets the dot data of the LCD that is illuminated or extinguished.

Data is set to this register via data buffer.

When data is set to the LCD group register, the value of the corresponding LCD segment register changes at the same time.

When the segment signal output pins are used as general-purpose output port pins, this register sets the output data.

If data is set to the LCD group register at this time, the value of the corresponding LCD segment register is changed at the same time.

For details, refer to **21.3**.

21.2.3 Common signal output timing control block

The common signal output timing control block controls the common signal output timing of the COM_1 and COM_0 pins.

These pins output low level when LCD display is not performed.

Whether LCD display is performed or not is selected by the LCD mode select register (LCDMODE: RF address 10H).

For details, refer to **21.4**.

21.2.4 Segment signal/key source signal output timing control block

The segment signal/key source signal output timing block controls the segment signal output timing of the LCD_{29}/$P0F_3$ through LCD_0/$P0Y_0$/KS_0 pins.

These pins output low level when LCD display is not performed.

Whether LCD display is performed or not is selected by the LCD mode select register.

The segment signal/key source signal output timing control block also controls the timing of the segment signals and key source signals output by the LCD_{15}/$P0Y_{15}$/KS_{15} through LCD_0/$P0Y_0$/KS_0 pins.

Whether the key source signals are used or not is selected by the LCD mode select register.

For details, refer to **21.4**.

21.2.5 Segment signal/general-purpose port select block

The segment signal/general-purpose port select block selects whether each segment signal output pin is used for LCD display (segment signal output) or as a general-purpose output port pin.

This selection is made by using the LCD port select register (LCDPORT: RF address 11H).

For details, refer to **21.4**.

21.2.6 Key source signal output control block

The key source signal output control block sets the key source signal output data output by the LCD_{15}/$P0Y_{15}$/KS_{15} through LCD_0/$P0Y_0$/KS_0 pins and detects the key input timing.

The key source signal output data is set by the key source data register (KSR: peripheral address 42H) via the data buffer.

The key source data register also sets the output data of port 0Y.

To use the key source signal, use the $P0D_3$/ADC_5 through $P0D_0$/ADC_2 pins as key input pins.

For details, refer to **22. KEY SOURCE CONTROLLER/DECODER**.

21.3 LCD Segment Register and LCD Group Register

The LCD segment register and LCD group register sets the display dot on an LCD to be illuminated or extinguished.

21.3.1 Configuration of LCD segment register

Figure 21-3 shows the location of the LCD segment register on the data memory.

Figure 21-4 shows the configuration of the LCD segment register.

Figure 21-3. Location of LCD Segment Register on Data Memory

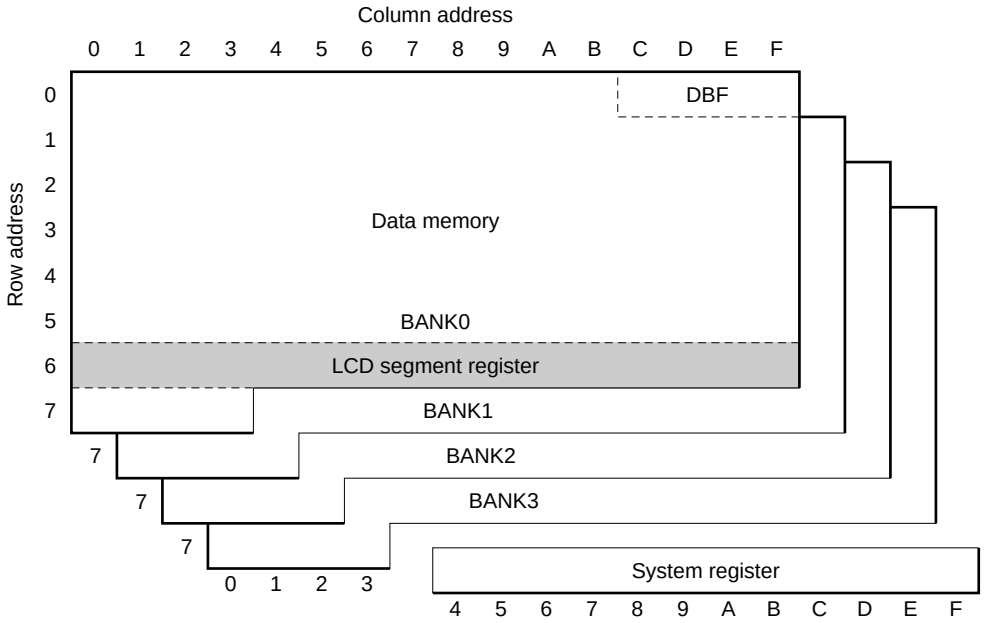


Figure 21-4. Configuration of LCD Segment Register

LCD segment register															
Address	60H	61H	62H	63H	64H	65H	66H	67H	68H	69H	6AH	6BH	6CH	6DH	6EH	6FH
Symbol	LCDD0	LCDD1	LCDD2	LCDD3	LCDD4	LCDD5	LCDD6	LCDD7	LCDD8	LCDD9	LCDD10	LCDD11	LCDD12	LCDD13	LCDD14	–

Nothing is allocated.
Cannot be used as data memory.

LCDD0			
b_3	b_2	b_1	b_0

21.3.2 Function of LCD segment register

Figure 21-5 shows the relation between 1 nibble (4 bits) of the LCD segment register and an LCD display dot.

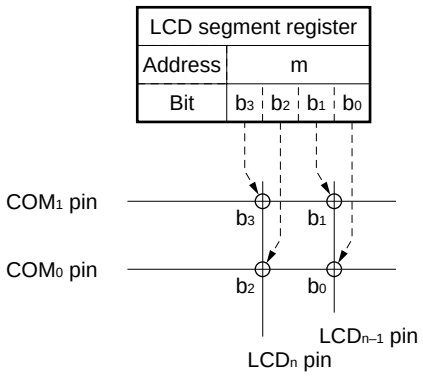
As shown in this figure, one nibble of the LCD segment register can set 4 dots of display data (data to be illuminated or extinguished).

An LCD display dot corresponding to the LCD segment register bit that is set to “1” lights, and a dot corresponding to the register bit that is reset to “0” remains dark.

The LCD segment register sets output data when the segment signal output pin is used as an output port pin.

Figure 21-7 shows the relation between the LCD segment register and LCD display dots that are illuminated or extinguished.

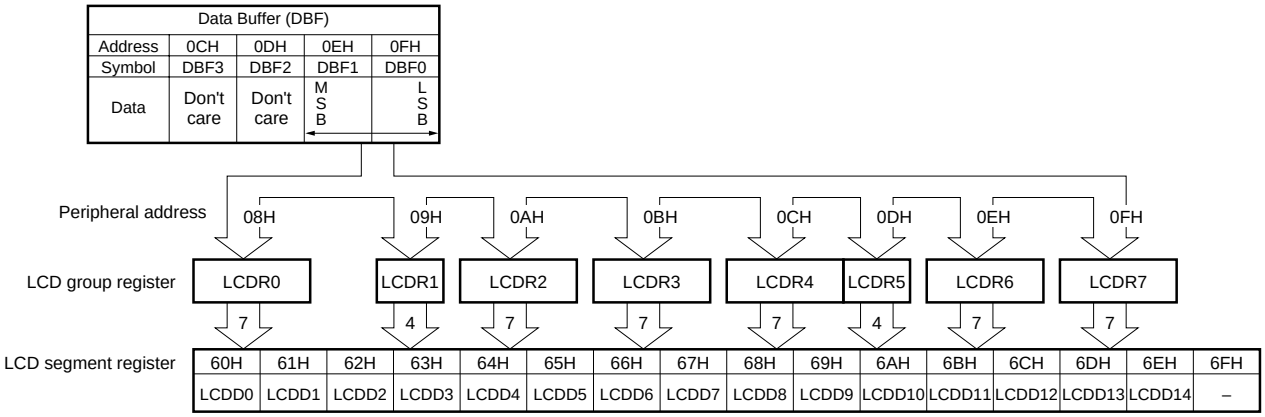
Figure 21-5. Relation between 1 Nibble of LCD Segment Register and LCD Display Dot



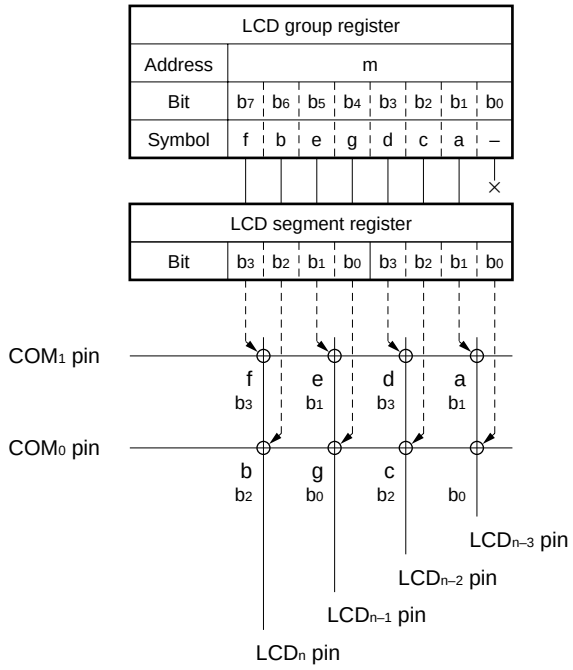
21.3.3 Configuration of LCD group register

Figure 21-6 shows the configuration of the LCD group register and its relation with the LCD segment register.

Figure 21-6. Configuration of LCD Group Register and Its Relation with LCD Segment Register



Relation between LCD group register and LCD display dot



21.3.4 Function of LCD group register

The LCD group register sets the data of the LCD display dot that is to be illuminated or extinguished, like the LCD segment register.

As shown in Figure 21-6, data is set to the LCD group register in 7-dot or 4-dot units via data buffer.

By executing the “PUT LCDRn, DBF” instruction, therefore, the LCD display data of a group specified by “n” ($0 \leq n \leq 7$) is set.

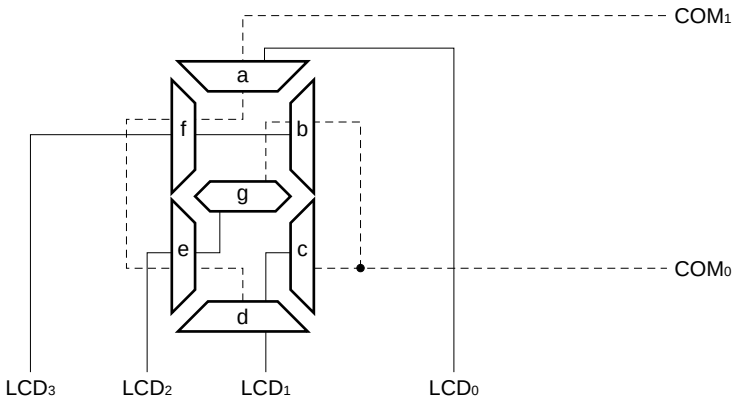
If the “PUT LCDRn, DBF” instruction is executed at this time, the corresponding value of the LCD segment register is changed accordingly.

In other words, display data of 7 dots can be set with a single instruction by using the LCD group register.

The LCD segment register sets output data when the segment signal output pin is used as an output port pin.

The following **21.3.5** describes the relation between the LCD group register and data buffer.

Because the LCD group register can set display data of 7 dots with one instruction, it can be used to display a 7-segment LCD wired as shown below.



The configuration and function of each LCD group register are described next.

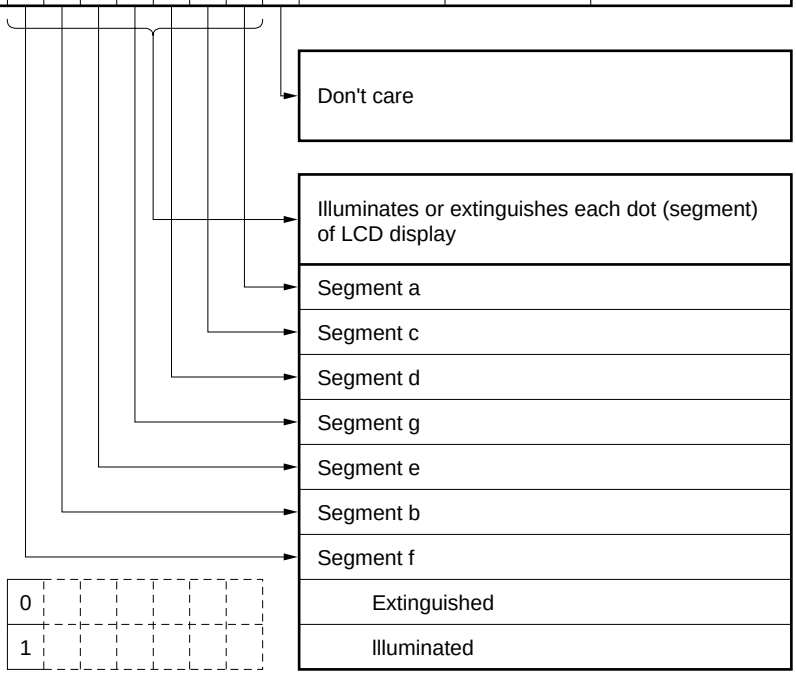
Name	Data Buffer															
Symbol	DBF3				DBF2				DBF1				DBF0			
Address	0CH				0DH				0EH				0FH			
Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data	Don't care				Don't care				Transfer data							

8

GET reads undefined data

PUT can be executed

Peripheral Register											
Name	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Symbol	Peripheral address	Peripheral hardware
LCD group register 0	Valid data							–	LCDR0	08H	LCD segment group 0
LCD group register 1	–				–		–	–	LCDR1	09H	LCD segment group 1
LCD group register 2								–	LCDR2	0AH	LCD segment group 2
LCD group register 3								–	LCDR3	0BH	LCD segment group 3
LCD group register 4								–	LCDR4	0CH	LCD segment group 4
LCD group register 5	–	–	–	–				–	LCDR5	0DH	LCD segment group 5
LCD group register 6								–	LCDR6	0EH	LCD segment group 6
LCD group register 7								–	LCDR7	0FH	LCD segment group 7



For the relation among segments a through g and each dot, refer to Figure 21-7.

Figure 21-7. Relation among LCD Display Dot, Ports 0E Through 0Y, Key Source Output, and Data Setting Registers (1/2)

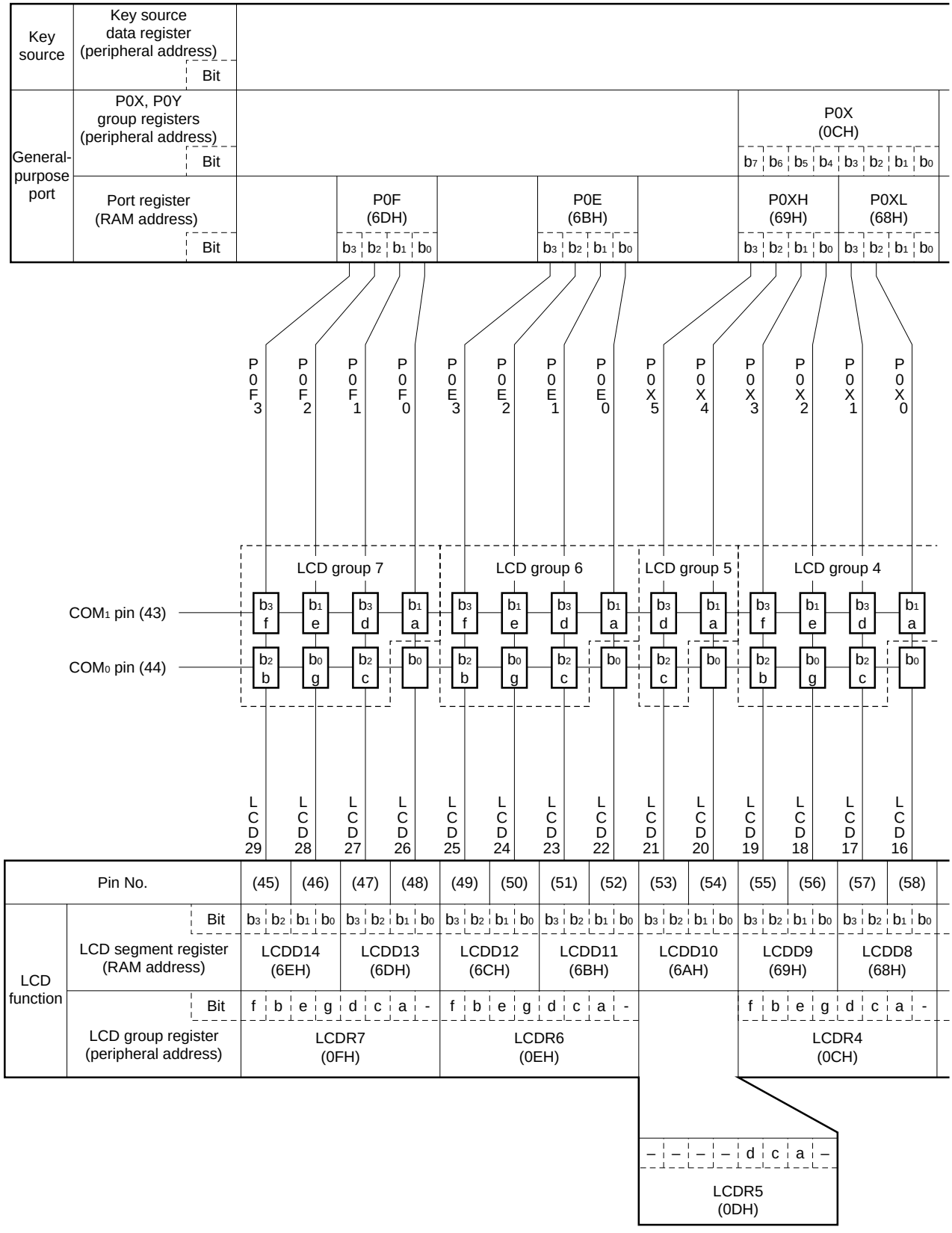
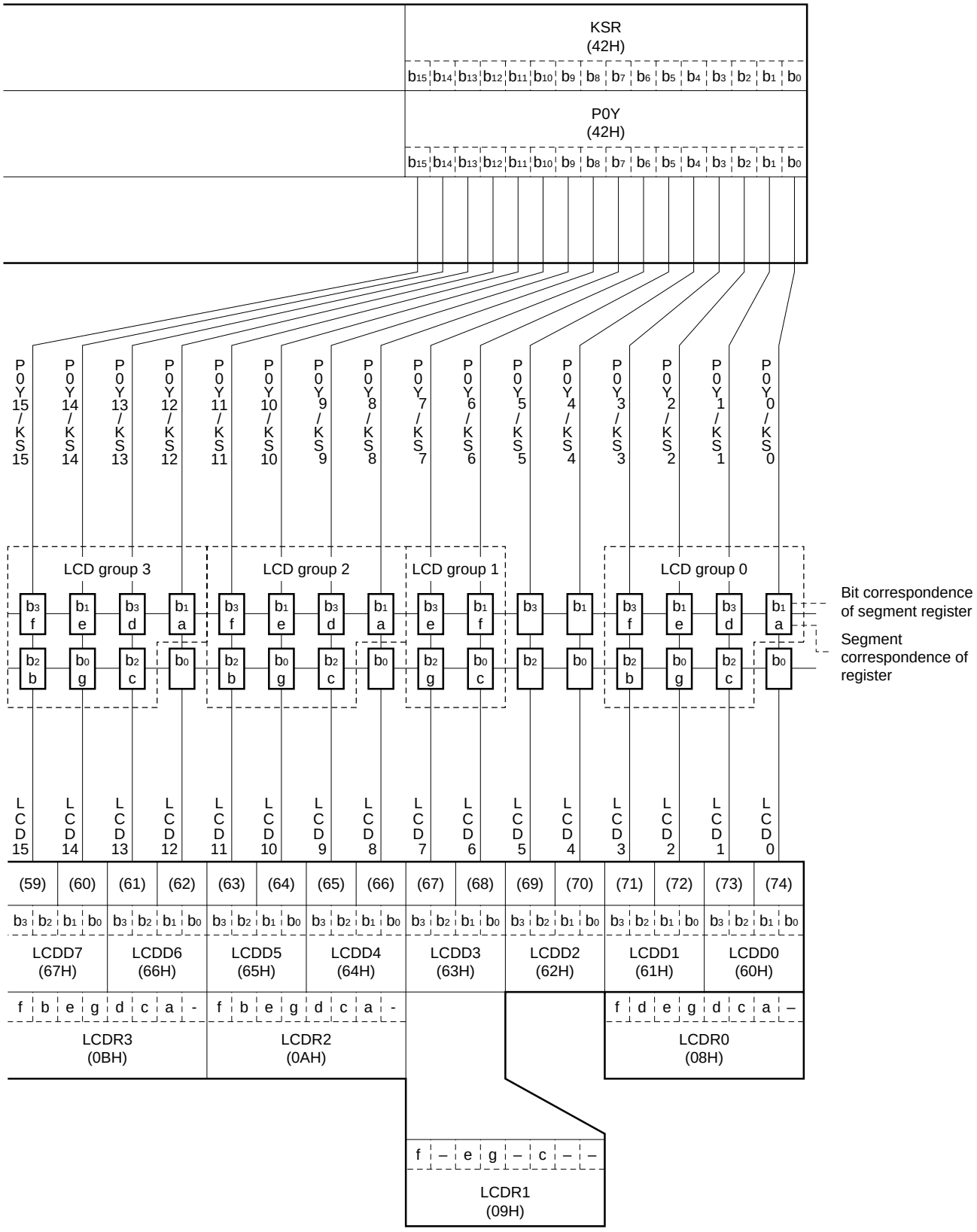


Figure 21-7. Relation among LCD Display Dot, Ports 0E Through 0Y, Key Source Output, and Data Setting Registers (2/2)

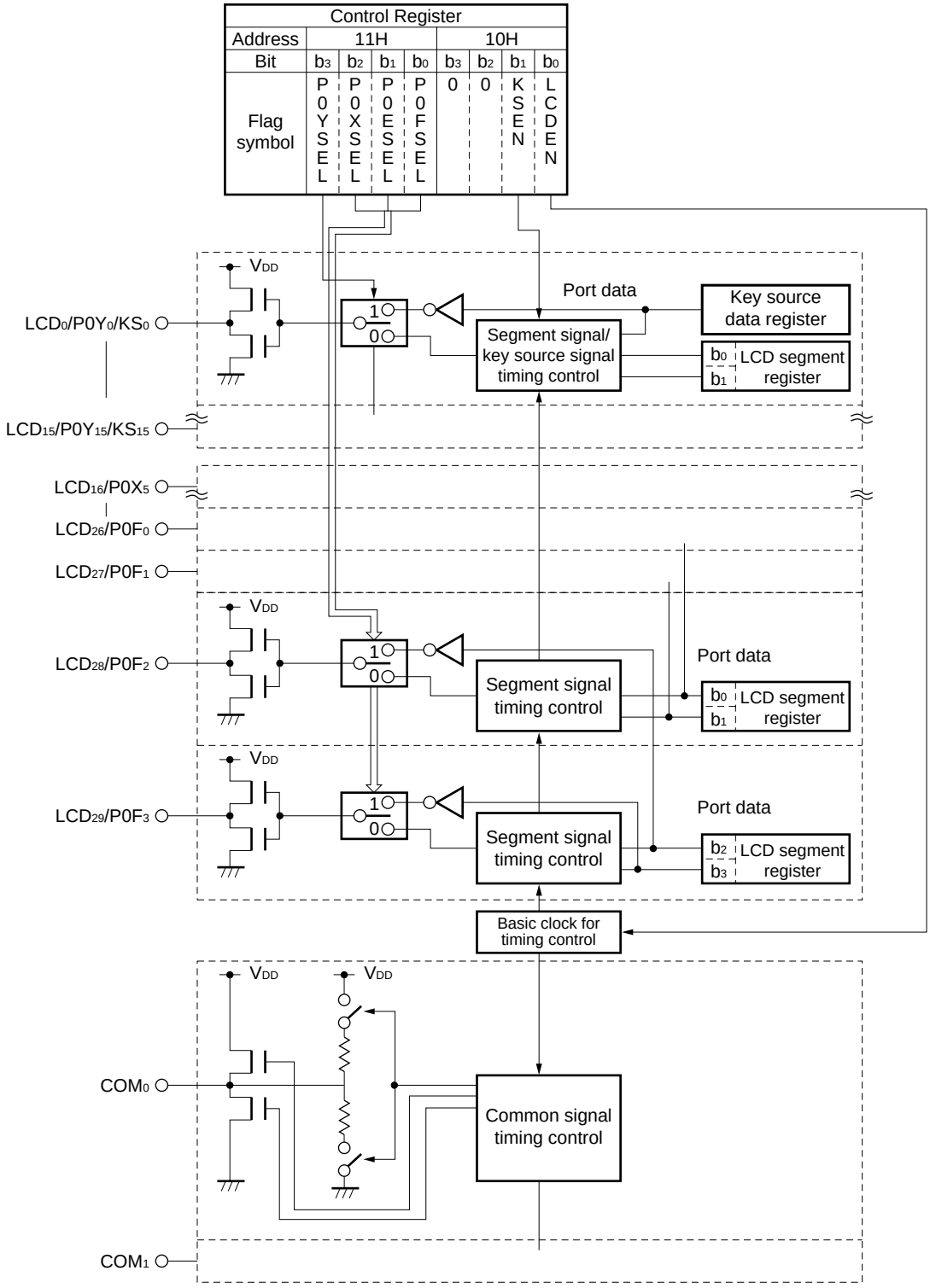


21.4 Output Timing Control Block and Segment/Port Select Block

21.4.1 Configuration of output timing control block and segment/port select block

Figure 21-8 shows the configuration of the common and segment signal/key source signal output timing control blocks and segment signal/general-purpose output port select block.

Figure 21-8. Configuration of Timing Control Block and Port Select Block



21.4.2 Function of segment signal/general-purpose output port select block

The segment signal/general-purpose output port select block specifies whether each pin is used as a segment signal output pin or a general-purpose output port pin, by using the P0YSEL through P0FSEL flags of the LCD port select register.

When each flag is “1”, the corresponding pin is specified as a general-purpose output port pin.

Segment pins that are not used as general-purpose output ports can be used to perform LCD display.

Although the LCD_{15}/$P0Y_{15}$/KS_{15} through LCD_0/$P0Y_0$/KS_0 pins can output segment signals and key source signals at the same time, port output takes precedence when port 0Y is selected.

For the details of the general-purpose output port, refer to **15. GENERAL-PURPOSE PORTS**.

The following **21.4.4** describes the configuration and function of the LCD port select register.

21.4.3 Function of output timing control block

The output timing control block controls the timing of the common and segment signals for LCD display and the timing of the key source and segment signals when the key source controller/decoder is used.

The common and segment signals are output when the LCDEN flag of the LCD mode select register is set to “1”.

In other words, the LCD display can be turned off by the LCDEN flag.

When the LCD display is turned off, the common and segment signals output low level.

The key source signal is output when the KSEN flag of the LCD mode select register is “1”.

Therefore, use of the key source signal can be specified by the KSEN flag.

21.4.5 describes the configuration and function of the LCD mode select register.

21.4.6 describes the output waveforms of the common and segment signals.

For the details of the key source controller/decoder, refer to **22. KEY SOURCE CONTROLLER/DECODER**.

21.4.4 Configuration and function of LCD port select register

The LCD port select register specifies whether the LCD segment signal output pins are used as general-purpose output port pins.

The configuration and function of this register are illustrated below.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
LCD port select register (LCDPORT)	P0YSEL	P0XSEL	P0ESEL	P0FSEL	11H	R/W

P0FSEL

Selects LCD segment signal output pin or general-purpose output port pin	
0	Uses $LCD_{26}/P0F_0$-$LCD_{29}/P0F_3$ pins as LCD segment pins
1	Uses $LCD_{26}/P0F_0$-$LCD_{29}/P0F_3$ pins as general-purpose port pins

P0ESEL

Selects LCD segment signal output pin or general-purpose output port pin	
0	Uses $LCD_{22}/P0E_0$-$LCD_{25}/P0E_3$ pins as LCD segment pins
1	Uses $LCD_{22}/P0E_0$-$LCD_{25}/P0E_3$ pins as general-purpose port pins

P0XSEL

Selects LCD segment signal output pin or general-purpose output port pin	
0	Uses $LCD_{16}/P0X_0$-$LCD_{21}/P0X_5$ pins as LCD segment pins
1	Uses $LCD_{16}/P0X_0$-$LCD_{21}/P0X_5$ pins as general-purpose port pins

P0YSEL

Selects LCD segment signal output pin or general-purpose output port pin	
0	Uses $LCD_0/P0Y_0/KS_0$-$LCD_{15}/P0Y_{15}/KS_{15}$ pins as LCD segment pins
1	Uses $LCD_0/P0Y_0/KS_0$-$LCD_{15}/P0Y_{15}/KS_{15}$ pins as general-purpose port pins

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	Retained			

21.4.5 Configuration and function of LCD mode select register (LCDMODE)

The LCD mode select register turns ON/OFF LCD display and specifies output of the key source signals.

The configuration and function of this register are illustrated below.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
LCD mode select register (LCDMODE)	0	0	KSEN	LCDEN	10H	R/W

Turns ON/OFF all LCD display	
0	Display OFF (segment and common output pins output low level)
1	Display ON

Sets output of key source signal	
0	Key source OFF
1	Key source ON

Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop			0	0
	CE			Retained	

21.4.6 Output waveforms of common and segment signals

Figures 21-9 and 21-10 show the output waveforms of the common and segment signals.

Figure 21-9 shows the waveform when the key source signals are not output, and Figure 21-10 shows the waveform when the key source signals are output.

As shown in Figure 21-9, the LCD driver outputs a 1/2-duty, 1/2-bias signal (voltage average method) with a frame frequency of 250 Hz.

As the common signals, the COM_1 and COM_0 pins output three levels of voltages (0, $1/2V_{DD}$, V_{DD}) each having a phase different of 1/4 from the others.

Therefore, voltages in a range of $\pm 1/2V_{DD}$ are output as the common signal. This method is called 1/2 bias driving method.

As the segment signals, two levels of voltages (0, V_{DD}) having a phase corresponding to the display dot are output from the segment signal output pins.

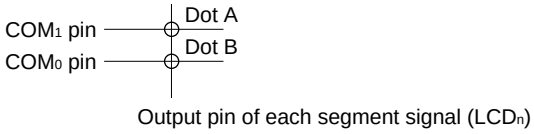
Because two display dots are illuminated or extinguished by one segment pin as shown in Figure 21-9, four types of phase differences, <1> through <4> in Figure 21-9, are output to dots A and B, by using combination of ON and OFF statuses.

Dots A and B light when the phase difference between the common and segment signals reaches V_{DD}.

The duty factor at which each of dots A and B lights is 1/2 and the frequency is 250 Hz.

This display method is called 1/2 duty driving method, and the frequency is called frame frequency.

Figure 21-9. Output Waveforms of Common and Segment Signals (when key source signals are not output)



Common signal

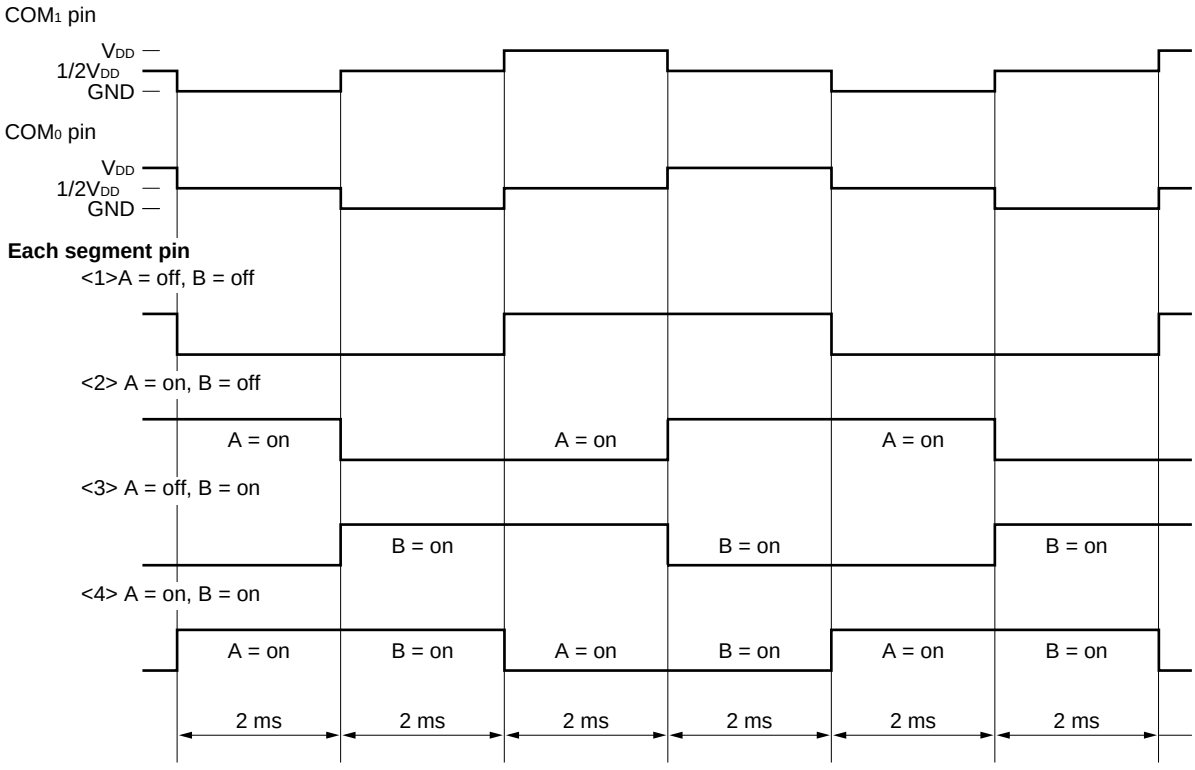
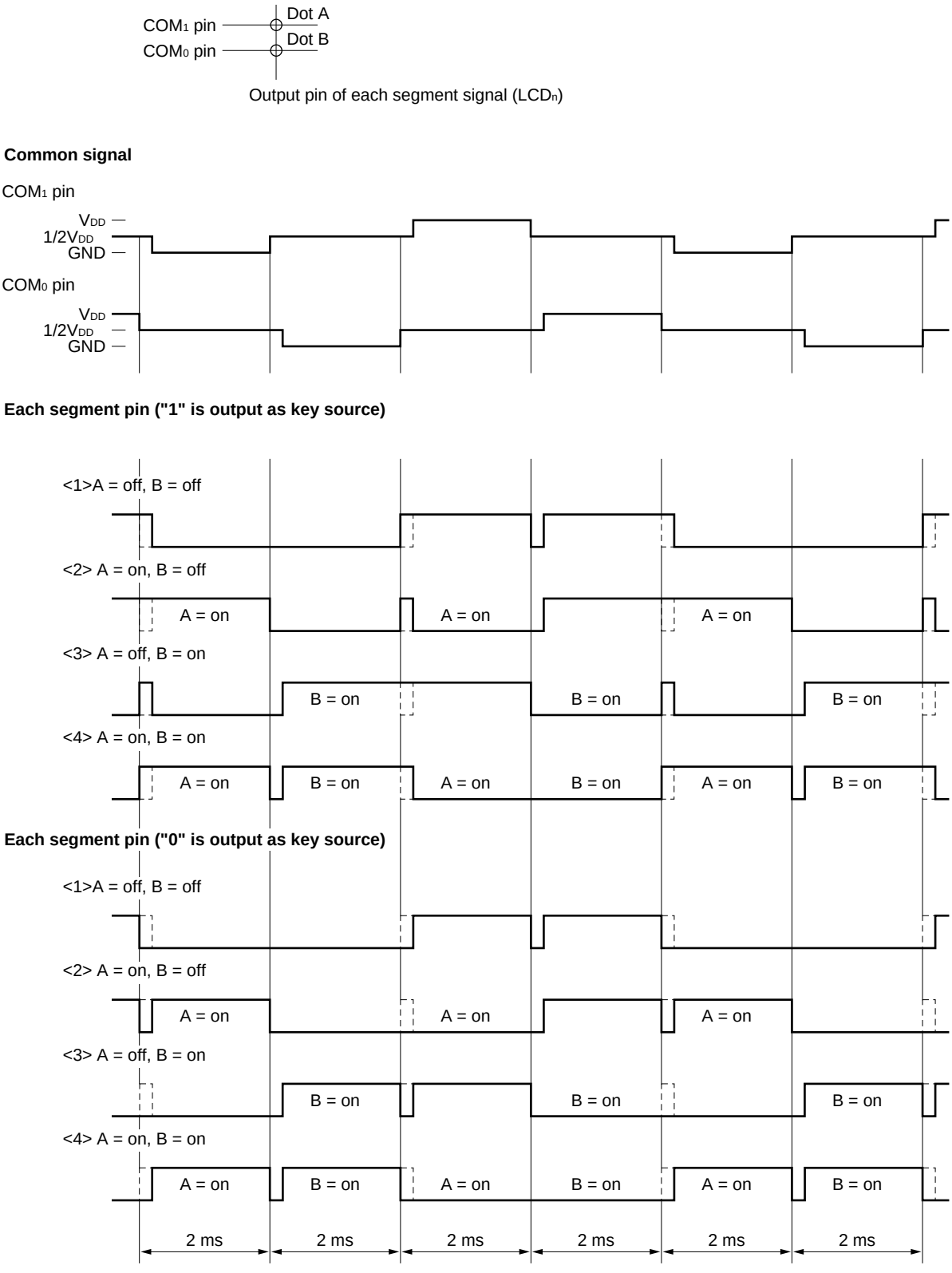


Figure 21-10. Output Waveforms of Common and Segment Signals (when key source signals are output)



21.5 Using LCD Controller/Driver

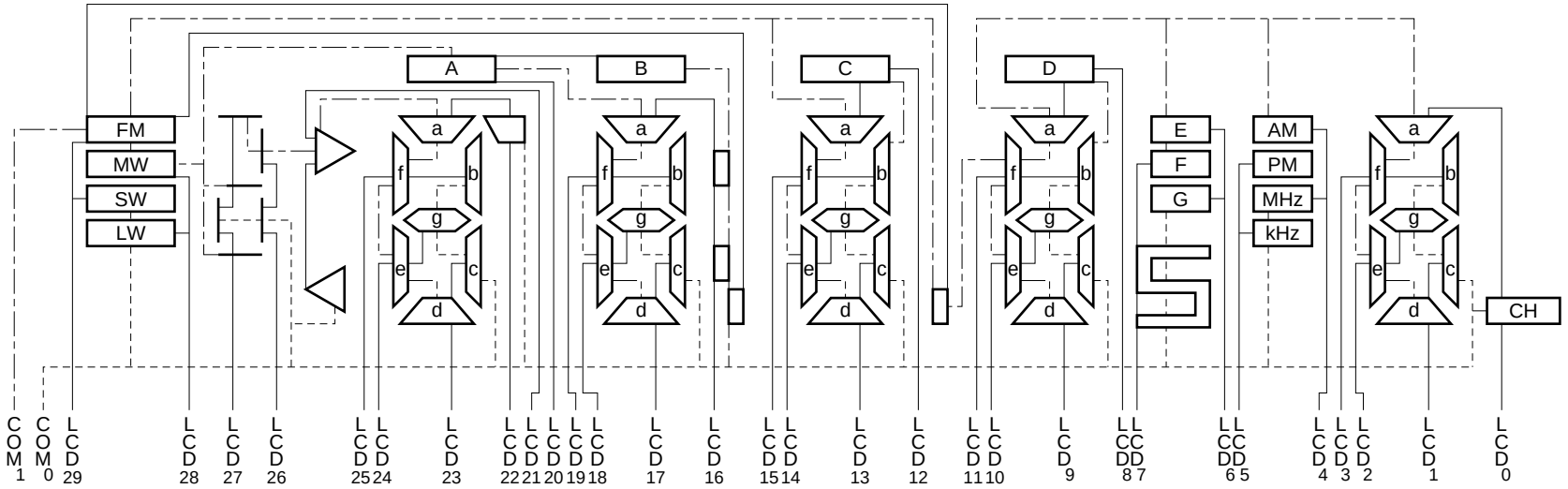
Figure 21-11 shows an example of wiring of an LCD panel.

An example of a program that turns on a 7-segment display by using the LCD_0 through LCD_3 pins and the wiring shown in Figure 21-11 is given below.

Example

```
    PMNO   MEM    0.01H       ; Preset memory number and BK data storage area
    CH     FLAG   DBF0.1      ; Symbol definition of least significant bit of DBF as "CH" display flag
LCDDATA:                      ; Table data for display
;                             b3b2b1b0b3b2b1b0  ; Corresponds to LCD segment register
;                             f  b  e g d c a –  ; Corresponds to LCD group register
    DW   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B ; BLANK
    DW   0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 B ; 1
    DW   0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 B ; 2
    DW   0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 B ; 3
    DW   0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 B ; 4
    DW   0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 0 B ; 5
    DW   0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 B ; 6
    DW   0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0 B ; 7
    DW   0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 B ; 8
    DW   0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0 B ; 9
    DW   0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 0 B ; A
    DW   0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 B ; B
    DW   0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 B ; C
    DW   0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 B ; D
    DW   0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 B ; E
    DW   0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 0 B ; F

    CLR4  P0YSEL, P0XSEL, P0ESEL, P0FSEL
    MOV   RPL, #1110B
    MOV   AR3, #.DL. LCDDATA SHR 12  AND 0FH
    MOV   AR2, #.DL. LCDDATA SHR 8   AND 0FH
    MOV   AR1, #.DL. LCDDATA SHR 4   AND 0FH
    MOV   AR0, #.DL. LCDDATA         AND 0FH
    ADD   AR0, PMNO
    ADDC  AR1,    #0
    ADDC  AR2,    #0
    ADDC  AR3,    #0
    MOVT  DBF,    @AR
    MOV   RPH,   #0000B
    MOV   RPL,   #0000B
    SKGE  PMNO, #0AH
    SET1  CH
    LD    LCDD1, DBF1
    LD    LCDD0, DBF0
    SET1  LCDEN
```

Figure 21-11. Example of Wiring of LCD Panel



Correspondence among Segment Pins, Common Pins, and LCD Panel Display

Segment pin / Common pin		LCD29	LCD28	LCD27	LCD26	LCD25	LCD24	LCD23	LCD22	LCD21	LCD20	LCD19	LCD18	LCD17	LCD16	LCD15	LCD14	LCD13	LCD12	LCD11	LCD10	LCD9	LCD8	LCD7	LCD6	LCD5	LCD4	LCD3	LCD2	LCD1	LCD0
		45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
COM_1	43	FM	MW			f	e	d	a	▷	A	f	e	d	a	f	e	d	a	f	e	d	a	F	E	PM	AM	f	e	d	a
COM_0	44	SW	LW			b	g	c		◁	B	b	g	c	:	b	g	c	C	b	g	c	D	G	G	kHz	MHz	b	g	c	CH

21.6 Reset Status

21.6.1 On power-ON reset

The $LCD_{29}/P0F_3$ through $LCD_0/P0Y_0/KS_0$ pins are specified as the LCD segment signal output pins, and output a low level.

The COM_1 and COM_0 pins output a low level.

Therefore, the LCD display is turned off.

21.6.2 On execution of clock stop instruction

The $LCD_{29}/P0F_3$ through $LCD_0/P0Y_0/KS_0$ pins are specified as the LCD segment signal output pins, and output a low level.

The COM_1 and COM_0 pins output a low level.

Therefore, the LCD display is turned off.

21.6.3 On CE reset

Of the $LCD_{29}/P0F_3$ through $LCD_0/P0Y_0/KS_0$ pins, those that are specified as segment signal output pins output segment signals and those that are specified as general-purpose output port pins retain the output values.

The COM_1 and COM_0 pins output the common signals.

21.6.4 In halt status

Of the $LCD_{29}/P0F_3$ through $LCD_0/P0Y_0/KS_0$ pins, those that are specified as segment signal output pins output segment signals and those that are specified as general-purpose output port pins retain the output values.

The COM_1 and COM_0 pins output the common signals.

22. KEY SOURCE CONTROLLER/DECODER

The key source controller/decoder can configure a key matrix consisting of up to 64 keys by outputting LCD segment signals and key source signals by means of time division.

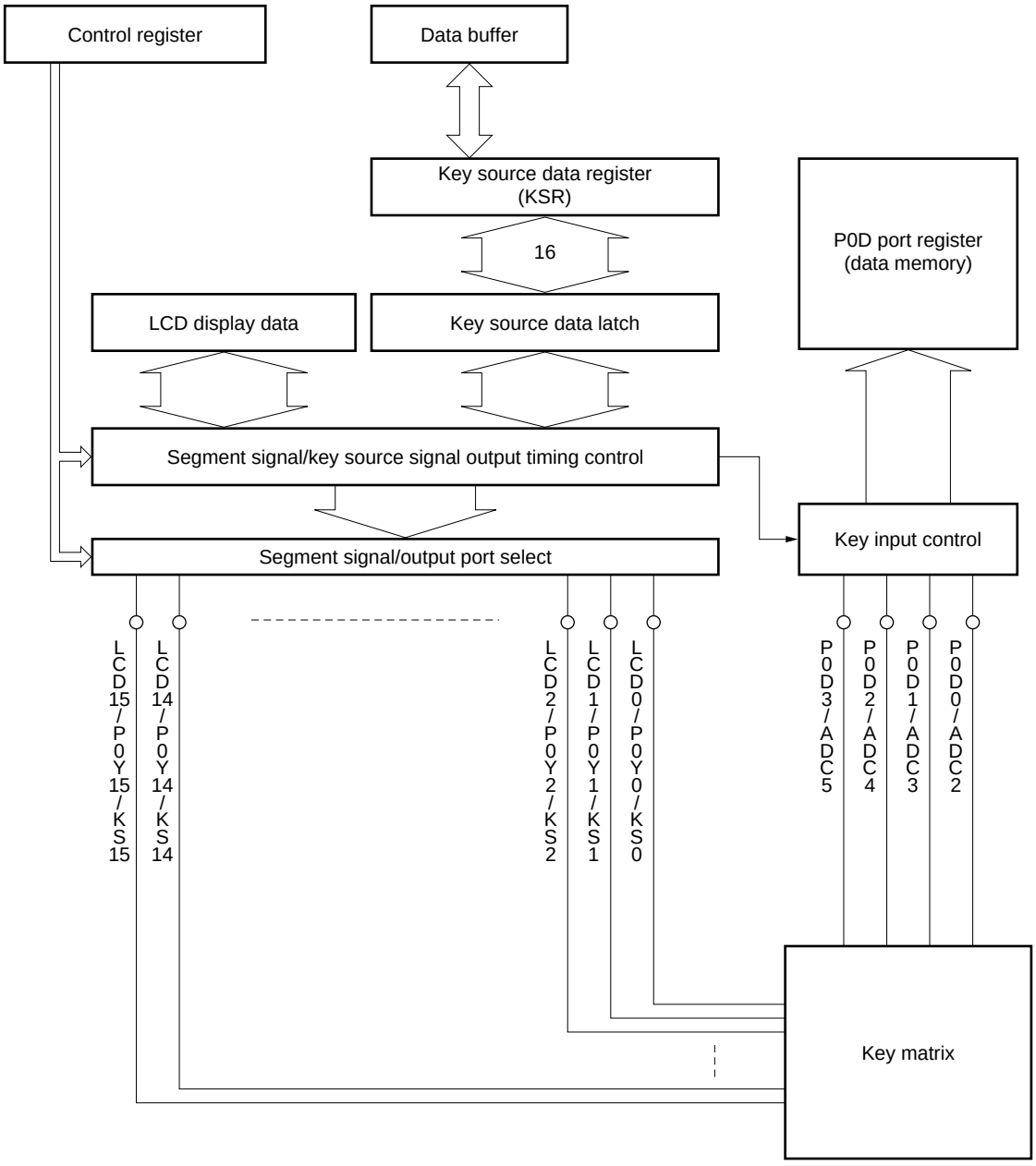
22.1 Configuration of Key Source Controller/Decoder

Figure 22-1 shows the configuration of the key source controller/decoder.

As shown in this figure, the key source controller/decoder consists of a segment signal/key source signal timing output control block, a segment signal/output port select block, a key source data register, a key input control block, and a P0D port register.

22.2 outlines the function of each block.

Figure 22-1. Block Diagram of Key Source Controller/Decoder



22.2 Functional Outline of Key Source Controller/Decoder

The key source controller/decoder can configure a key matrix of up to 64 keys by using the key source signal output pins ($LCD_{15}/P0Y_{15}/KS_{15}$ through $LCD_0/P0Y_0/KS_0$ pins) and key input pins ($P0D_3/ADC_5$ through $P0D_0/ADC_2$ pins).

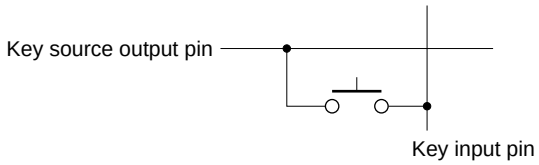
Figure 22-2 shows an example of configuration of a key matrix.

The $LCD_{15}/P0Y_{15}/KS_{15}$ through $LCD_0/P0Y_0/KS_0$ pins are shared with LCD segment signal output pins.

Therefore, these pins output key source signals and LCD segment signals by means of time division.

The following **22.2.1** through **22.2.4** outline the function of each block of the key source controller/decoder.

Figure 22-2. Example of Key Matrix Configuration



22.2.1 Key source data register (KSR)

The key source data register sets the key source output data of the pin that outputs a key source signal.

Data are set to this register via data buffer.

When data are set to this register, the key source data are output from the $LCD_{15}/P0Y_{15}/KS_{15}$ through $LCD_0/P0Y_0/KS_0$ pins.

The key source data register also sets output data when the $LCD_{15}/P0Y_{15}/KS_{15}$ through $LCD_0/P0Y_0/KS_0$ pins are used as general-purpose output port pins.

When data are set to the key source data register, the port output data are output from the corresponding pins.

For details, refer to **22.3**.

22.2.2 Segment signal/key source signal output timing control block

The segment signal/key source signal output timing control block controls the timing of the key source and segment signals output from the $LCD_{29}/P0F_3$ through $LCD_0/P0Y_0/KS_0$ pins.

Whether the key source signals are used or not is specified by the LCD mode select register.

The key source signals are not output when the LCD display is not used. At this time, these pins output a low level.

Whether the LCD display is used or not is specified by the LCD mode select register.

For details, refer to **22.4**.

22.2.3 Segment signal/general-purpose port select block

The segment signal/general-purpose port select block selects whether the $LCD_{29}/P0F_3$ through $LCD_0/P0Y_0/KS_0$ pins are used for LCD display (segment signal output) or as general-purpose output port pins.

Whether the segment signal output or general-purpose output port pin is used is specified by the LCD port select register.

To output the key source signals, the $LCD_{29}/P0F_3$ through $LCD_0/P0Y_0/KS_0$ pins must be specified as the LCD signal output pins.

For details, refer to **22.4**.

22.2.4 Key input control block and P0D port register

The key input control block detects the key signals input to the $P0D_3/ADC_5$ through $P0D_0/ADC_2$ pins in synchronization with the key source signal output timing.

Therefore, to output the key source signals from the $LCD_{15}/P0Y_{15}/KS_{15}$ through $LCD_0/P0Y_0/KS_0$ pins, the $P0D_3/ADC_5$ through $P0D_0/ADC_2$ pins are used as key input pins.

The key input data are read by the P0D port register (address 73H of BANK0) on the data memory.

Because the $P0D_3/ADC_5$ through $P0D_0/ADC_2$ pins are multiplexed with the A/D converter pins, care must be exercised when using these pins as the A/D converter pins.

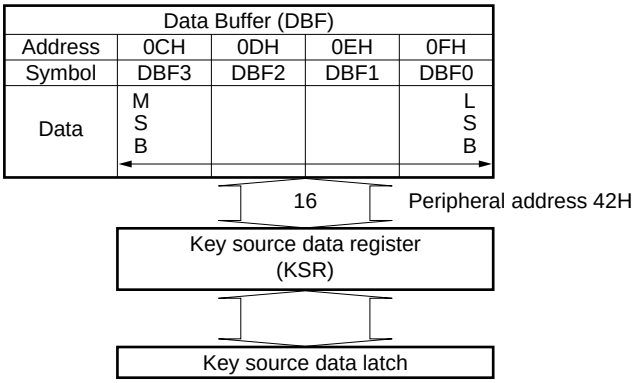
For details, refer to **22.5**.

22.3 Key Source Data Setting Block

22.3.1 Configuration of key source data setting block

Figure 22-3 shows the configuration of the key source data setting block.

Figure 22-3. Configuration of Key Source Data Setting Block



22.3.2 Function of key source data setting block

The key source data setting block sets the key source data output from the LCD_{15}/$P0Y_{15}$/KS_{15} through LCD_0/$P0Y_0$/KS_0 pins.

The key source data is set by the key source data register (KSR: peripheral address 42H) via data buffer.

Each bit of the key source data register corresponds to each of the LCD_{15}/$P0Y_{15}$/KS_{15} through LCD_0/$P0Y_0$/KS_0 pins, and sets the key source data of each pin.

The pin that is set to “1” by the key source data register outputs a high level as the key source signal. The pin that is reset to “0” outputs a low level.

For the output timing, refer to **22.4**.

When the LCD_{15}/$P0Y_{15}$/KS_{15} through LCD_0/$P0Y_0$/KS_0 pins are used as general-purpose output port pins, this block sets the output data.

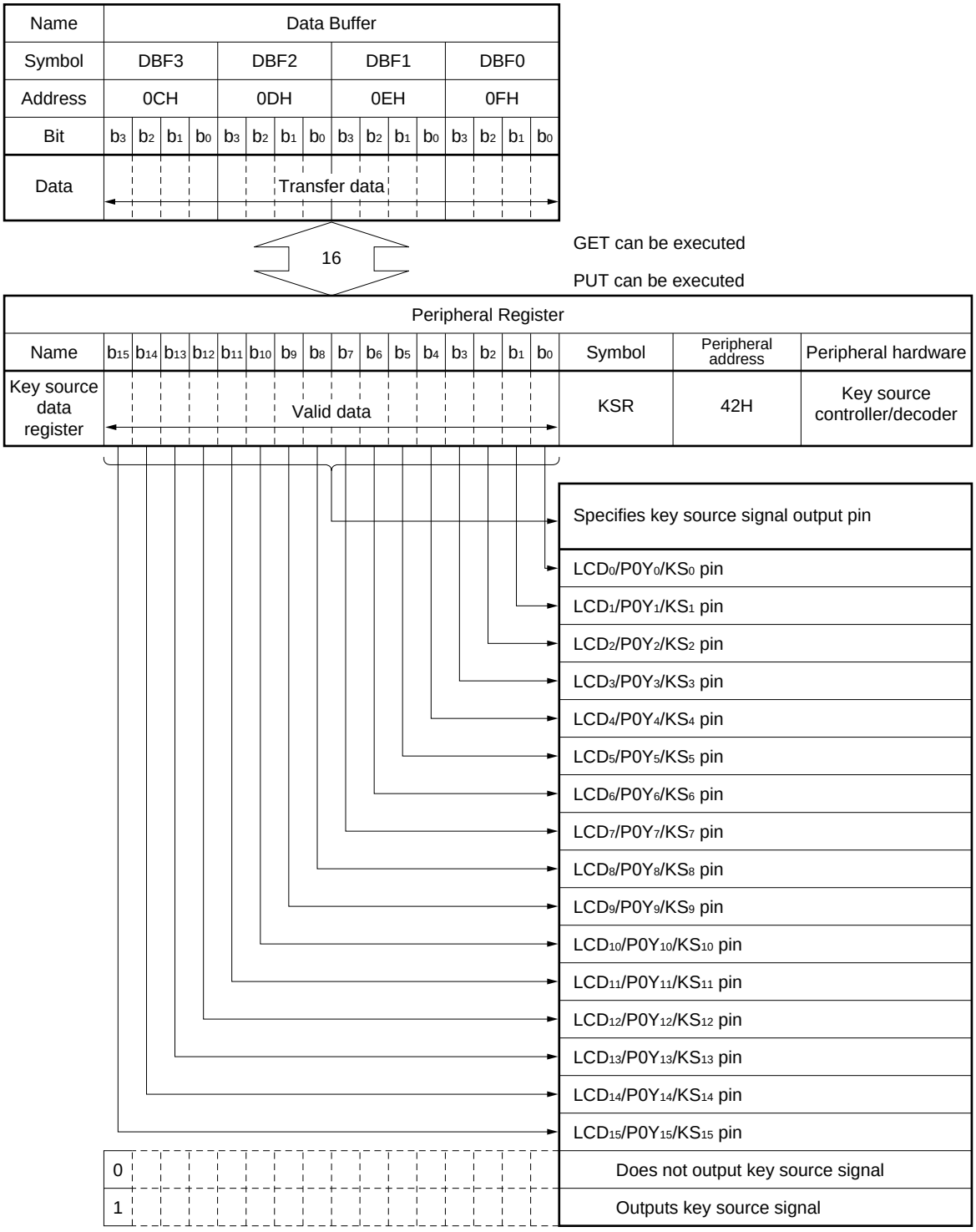
The register that sets the data at this time is called the P0Y group register (P0Y: peripheral address 42H). The peripheral address of this register is the same as that of the key source data register. The only difference is the name.

The following **22.3.3** describes the configuration and function of the key source data register.

Also refer to Figure 21-7 in **21. LCD CONTROLLER/DRIVER**.

22.3.3 Configuration and function of key source data register (KSR)

The configuration and function of the key source data register are illustrated below.

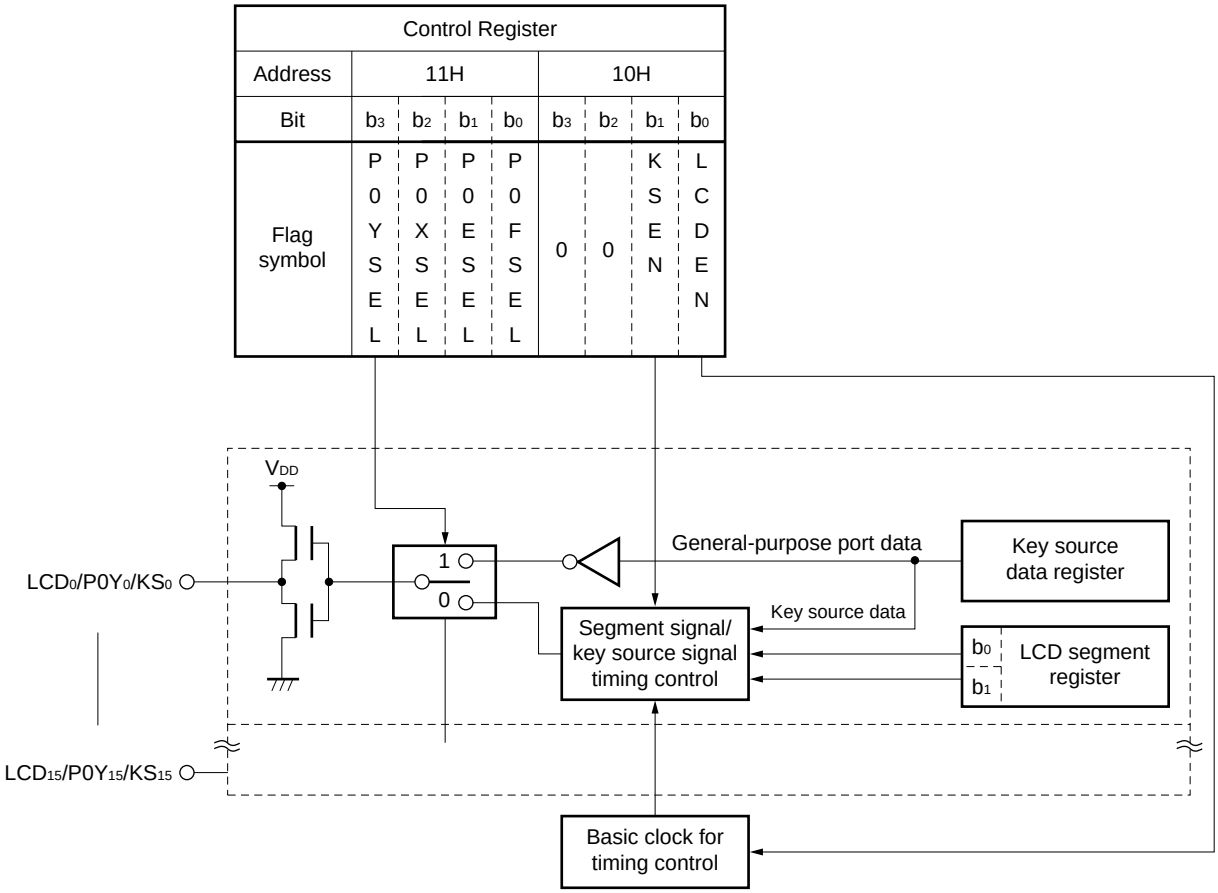


22.4 Output Timing Control Block and Segment/Port Select Block

22.4.1 Configuration of output timing control block and segment/port select block

Figure 22-4 shows the common signal and segment signal/key source signal output timing control blocks and segment signal/general-purpose output port select block.

Figure 22-4. Configuration of Timing Control Block and Port Select Block



22.4.2 Function of segment signal/general-purpose output port select block

The segment signal/general-purpose output port select block specifies whether the $LCD_{15}/P0Y_{15}/KS_{15}$ through $LCD_0/P0Y_0/KS_0$ pins are used as segment signal output pins or general-purpose output port (port 0Y) pins, by using the P0YSEL flag of the LCD port select register.

When the P0YSEL flag is "1", these pins are used as general-purpose output port pins.

To output key source signals from the $LCD_{15}/P0Y_{15}/KS_{15}$ through $LCD_0/P0Y_0/KS_0$ pins, the P0YSEL flag must be reset to "0".

When port 0Y is selected, the port output takes precedence.

For the details of the general-purpose output port, refer to **15. GENERAL-PURPOSE PORTS**.

22.4.3 Function of output timing control block

The output timing control block controls the timing of the key source and segment signals.

The LCD segment signals are output when the LCDEN flag of the LCD mode select register is "1".

The LCD display is turned off when the LCDEN flag is reset to "0". At this time, the segment signal pins output a low level, and the key source signals are not output.

To output the key source signals, therefore, the LCDEN flag must be "1".

The key source signals are output when the KSEN flag of the LCD mode select register is "1".

Therefore, whether the key source signals are used or not is specified by the KSEN flag.

To output the key source signals, therefore, the P0YSEL flag must be "0" and, at the same time, the LCDEN and KSEN flags must be "1".

The following **22.4.4** describes the configuration and function of the LCD mode select register.

22.4.5 describes the output waveforms of the key source signals and segment signals.

For the relation among the common and segment signals of the LCD, and key source signals, refer to **21. LCD CONTROLLER/DRIVER**.

22.4.4 Configuration and function of LCD mode select register (LCDMODE)

The LCD mode select register turns ON/OFF LCD display and specifies output of the key source signals.

The configuration and function of this register are illustrated below.

Name	Flag Symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
LCD mode select register (LCDMODE)	0	0	KSEN	LCDEN	10H	R/W

LCDEN — Turns ON/OFF all LCD display

0	Display OFF (segment and common output pins output low level)
1	Display ON

KSEN — Sets output of key source signal

0	Key source OFF
1	Key source ON

Fixed to "0"

On reset	Power-ON	0	0	0	0
	Clock stop			0	0
	CE			Retained	

22.4.5 Output waveforms of segment and key source signals

Figure 22-5 shows the output waveforms of the key source and segment signals.

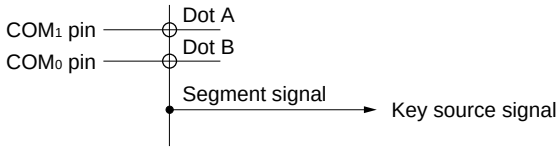
As shown in this figure, the key source and segment signals are output by means of time division.

The key source signal is output for 220 μs at intervals of 4 ms.

The pin which is set to “1” by the key source register outputs a high level for 220 μs every 4 ms, and the pin which is reset to “0” by the key source register outputs a low level for 220 μs every 4 ms.

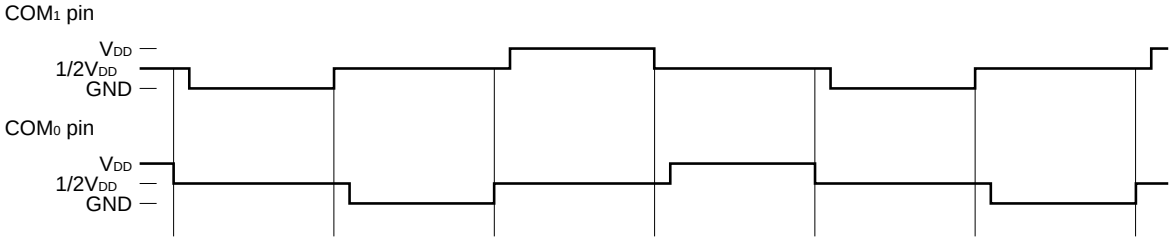
When output of the key source signal is specified (KSEN flag = 1), the pins that do not output key sources (LCD_{29}/$P0F_3$ through LCD_{16}/$P0X_0$ pins) output the waveform shown in Figure 22-5. However, waveform of “0” is output as the key source data.

Figure 22-5. Output Waveforms of Key Source and Segment Signals

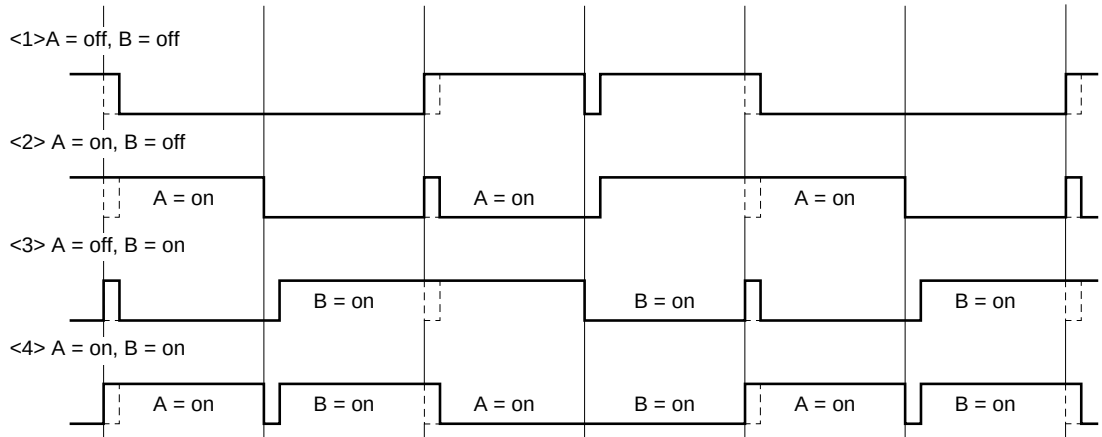


Output pin of each segment/key source signal (LCD_n/KS_n)

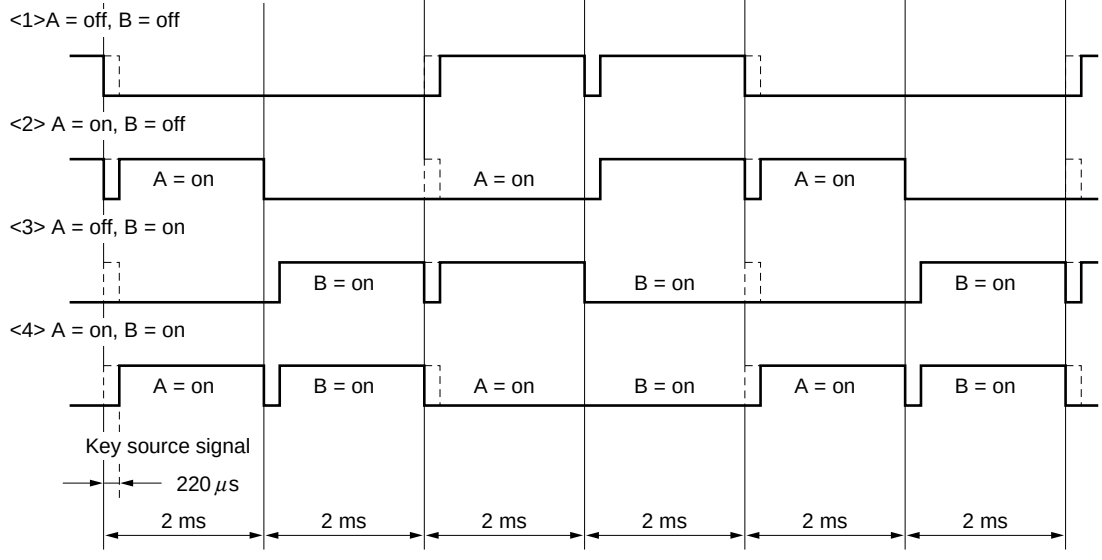
Common signal



Each segment pin ("1" is output as key source)



Each segment pin ("0" is output as key source)

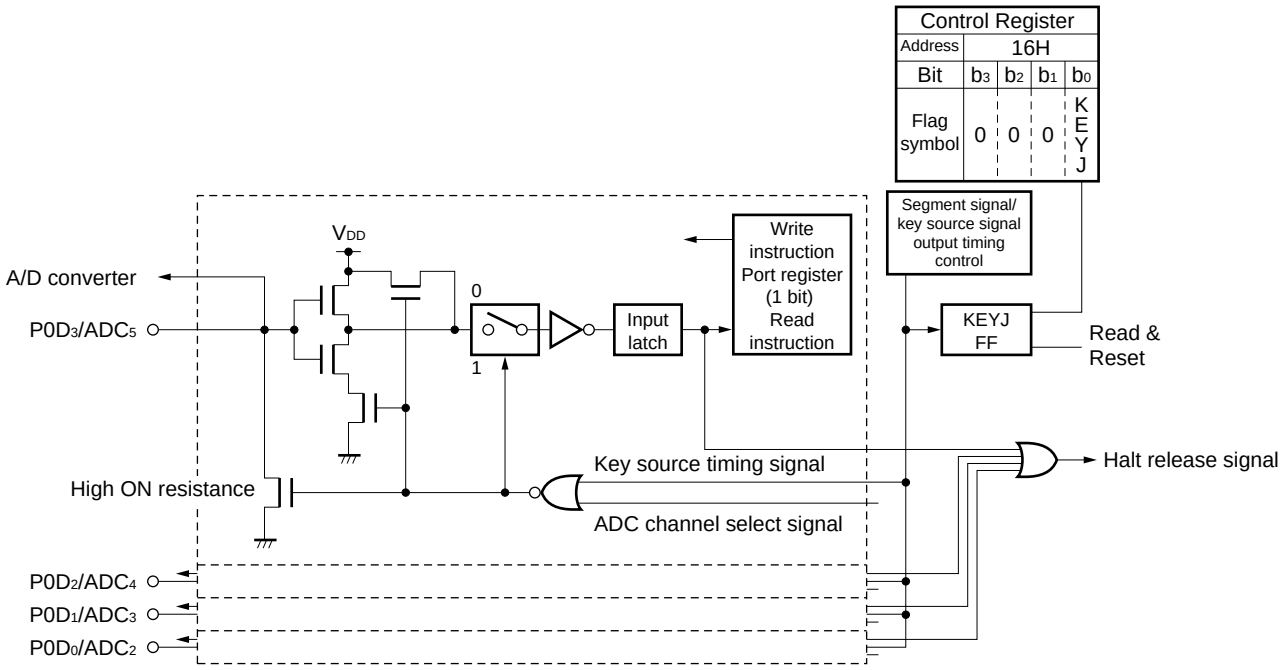


22.5 Key Input Block

22.5.1 Configuration of key input block

Figure 22-6 shows the configuration of the key input block.

Figure 22-6. Configuration of Key Input Control Block



22.5.2 Function of key input control block

The key input control block controls the timing to read the key input signals from the P0D3/ADC5 through P0D0/ADC2 pins and reads key input data.

Figure 22-7 shows the key source signals and key input timing.

As shown in this figure, the internal pull-down resistor of the P0D3/ADC5 through P0D0/ADC2 pins are off while the display data of the LCD segment is output, and on for only 220 μs while the key source signals are output.

The signal input to each key input pin is connected to the input latch for 220 μs during which the key source signals are output.

Therefore, the signal input to each key input pin can be detected during the period of 220 μs in which the key source signals are output.

Figure 22-8 shows the timing chart of the key source signals, key input signals, and key input data (P0D port register).

Whether the key source signals are output or not is detected by the KEYJ flag of the key input judge register (KEYJDG: RF address 16H).

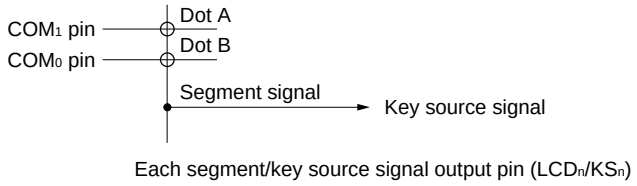
The KEYJ flag is set after the key source signals have been output for 220 μs, and is reset when data has been set to the key source data register or the content of the KEYJ flag has been read.

Therefore, the key input can be loaded by detecting the KEYJ flag

after the key source signal data has been output to the key source data register, and detecting the status of each key input pin after the KEYJ flag has been set to “1”.

The following subsection **22.5.3** describes the configuration and function of the key input judge register.

Figure 22-7. Key Source Signal and Key Input Timing



Each segment/key source signal output pin (LCD_n/KS_n)

Each segment pin (pin outputting "1" to key source. A = on, B = off)

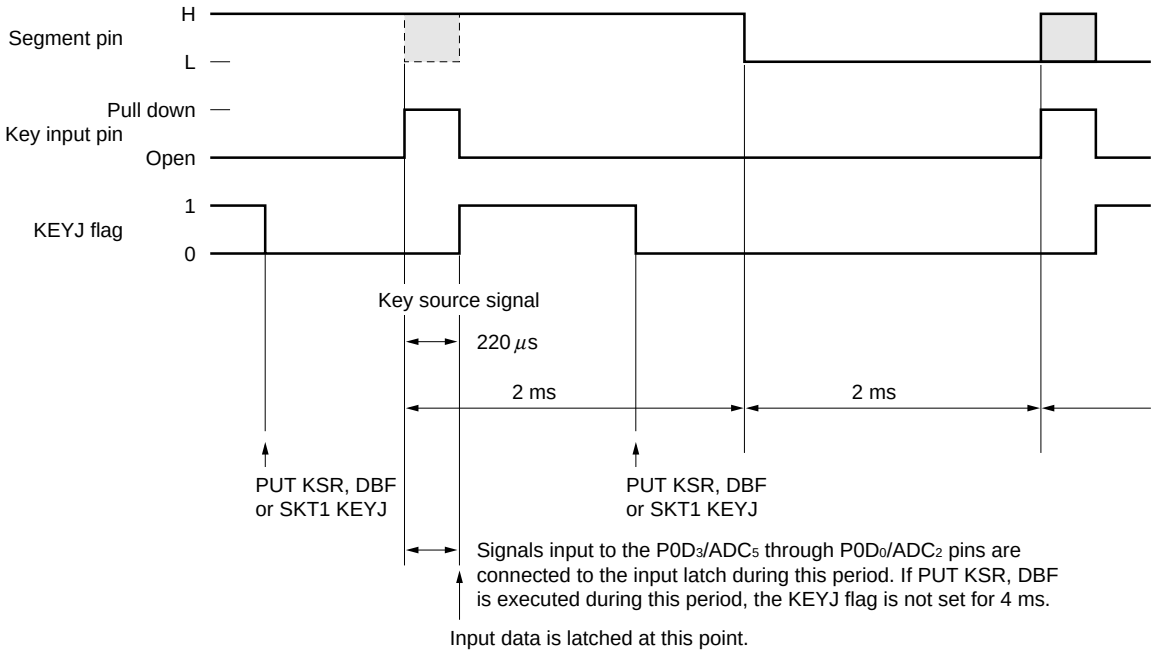
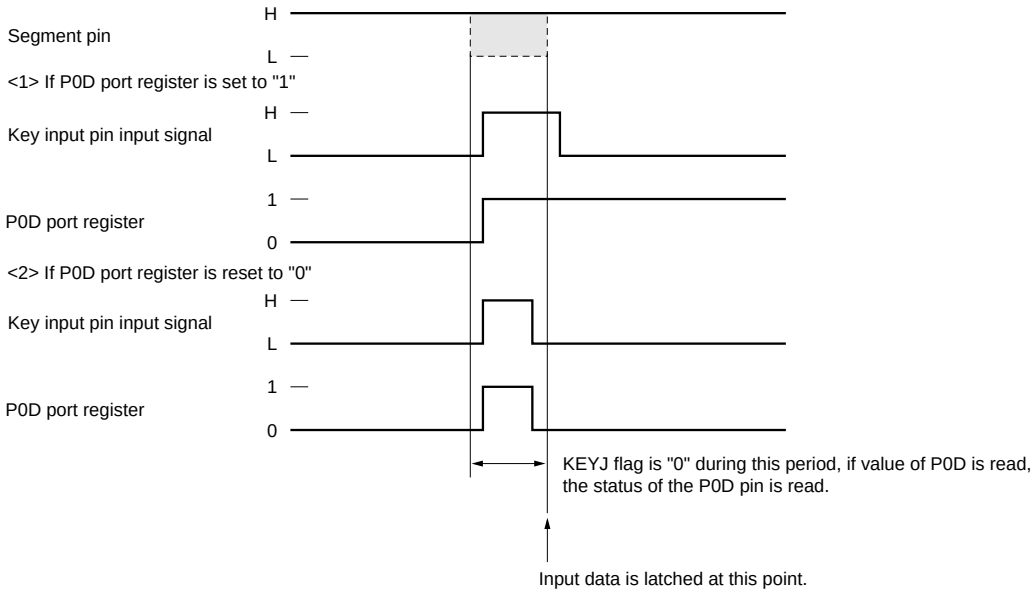


Figure 22-8. Timing Chart of Key Source Signal, Key Input Signal, and Key Input Data (P0D Port Register)



22.5.3 Configuration and function of key input judge register

The key input judge register detects the presence or absence of a latched key input signal when the LCD segment signal output pins are multiplexed with the key source signal output pins.

The configuration and function of this register are illustrated below.

Name	Flag Symbol b_3	b_2	b_1	b_0	Address	Read/ Write
Key input judge register (KEYJDG)	0	0	0	KEYJ	16H	R & Reset

Detects presence/absence of latched key input signal	
0	Key not latched
1	Key latched

Fixed to "0"

		b_3	b_2	b_1	b_0
On reset	Power-ON	0	0	0	0
	Clock stop	↓	↓	↓	0
	CE	↓	↓	↓	0

The key source signal data is set by setting the contents of the data buffer to the key source data register by using the “PUT” instruction.

When the key source signal output data is set by the “PUT” instruction via data buffer, the KEYJ flag is reset to 0.

The KEYJ flag is also reset to 0 when it is read to the window register by the “PEEK” instruction (Read & Reset).

22.6 Using Key Source Controller/Decoder

22.6.1 Configuration of key matrix

Figure 22-9 shows an example of key matrix configuration.

As shown in this figure, a key matrix can consist of up to 64 keys.

Because the key source signal output pins also output LCD segment signals at the same time, diode “A” must be connected to prevent the flowing back of the LCD segment signals when a momentary switch is used.

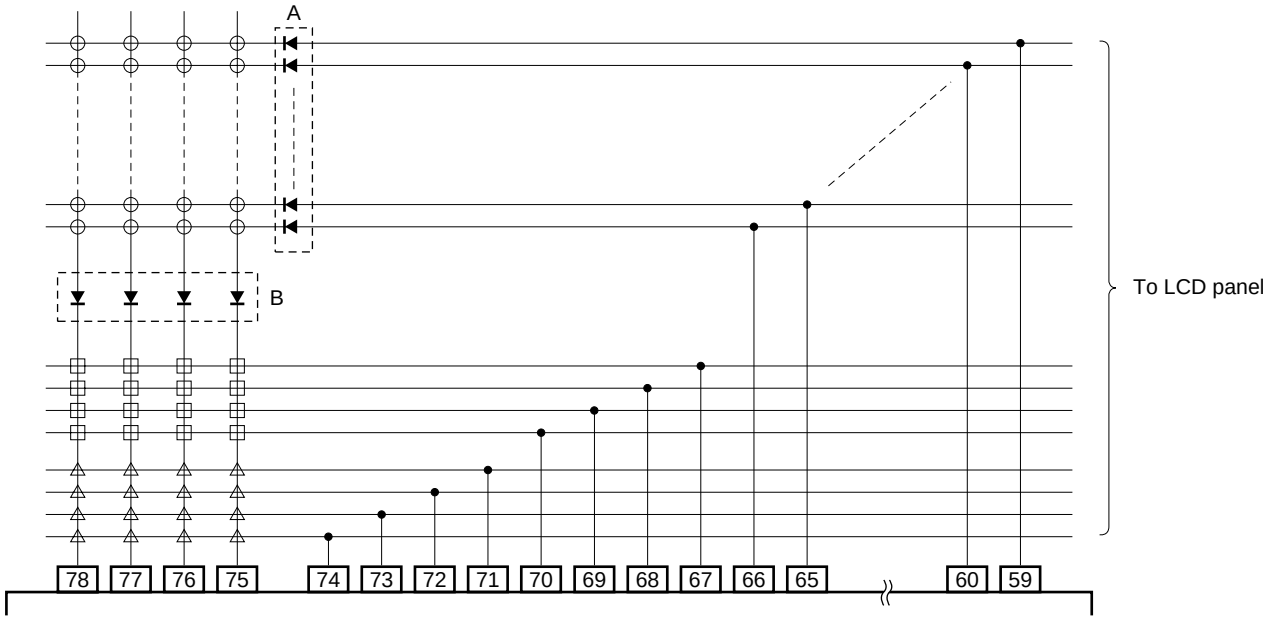
Diodes “B” and “C” are used to prevent sneaking of the key source signal.

Use PNP transistors as the transistor switches.

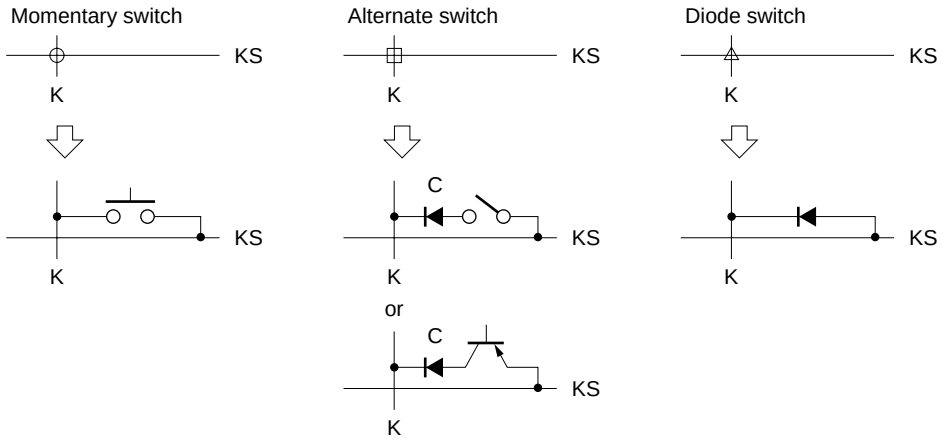
The following paragraph (1) describes the points to be noted when NPN transistors are used.

Paragraphs (2) through (4) describe the points to be noted when diodes A, B, and C are not used.

Figure 22-9. Example of Key Matrix Configuration

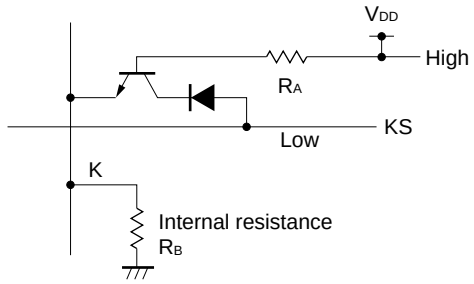


Configuration of each switch



(1) Note on using a NPN transistor switch

If an NPN transistor is used as a transistor switch, a low level may not be correctly read as shown in the example below.



In the figure on the left, if KS is low and a high level is input to the base of the transistor, the voltage V_K input to K is as follows:

$$V_K = \frac{R_B}{R_A + R_B} \times (V_{DD} - V_{BE})$$

Because KS is low at this time, a low level must be input to K.

However, the voltage input to K changes with R_A and R_B as shown in the above expression.

Therefore, a low level may not be input to K depending on the values of R_A and R_B.

(2) If diode A is not used

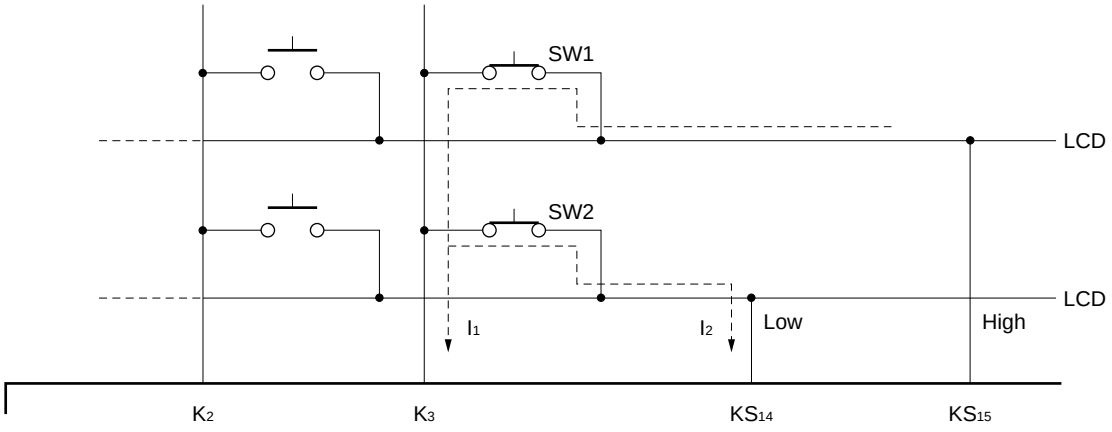
A circuit example where diode A is missing is shown below.

Suppose that switches SW1 and SW2 are on, that a high level is output from KS_{15}, and that a low level is output from KS_{14}, as shown below.

If diode A is missing at this time, the currents I_1 and I_2 shown by the dotted line will start to flow.

Therefore, the high level of KS_{15} and low level of KS_{14} are not correctly output because of I_2. The result is that the key data of K_3 cannot be accurately read.

If KS_{15} and KS_{14} are used as LCD segment signal output pins, the LCD display does not correctly turn on or off.



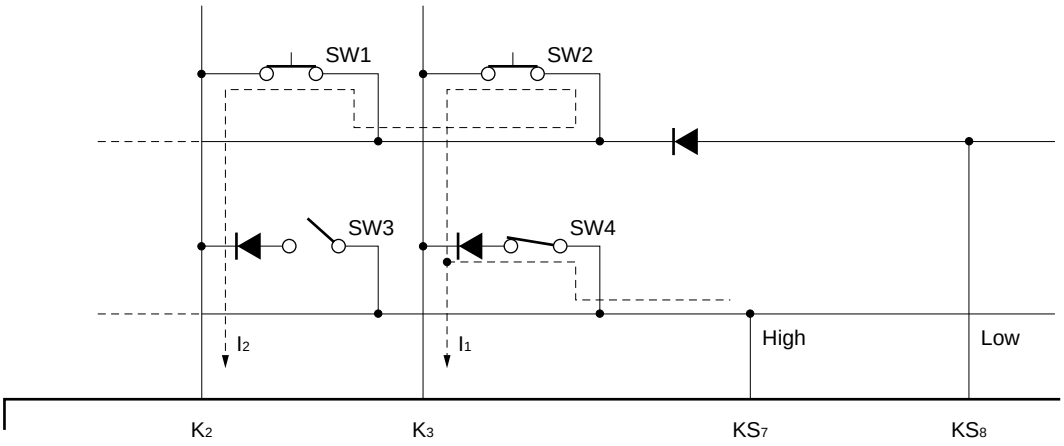
(3) If diode B is not used

A circuit example where diode B is missing is shown below.

Suppose that switches SW1, SW2, and SW4 are on, and that a high level is output from KS_7, as shown below.

If diode B is missing at this time, currents I_1 and I_2 shown by the dotted line will start to flow.

Therefore, a high level is input to K_2 because of I_2 even when switch SW3 is turned off. Consequently, it judges that SW3 is on.



(4) If diode C is missing

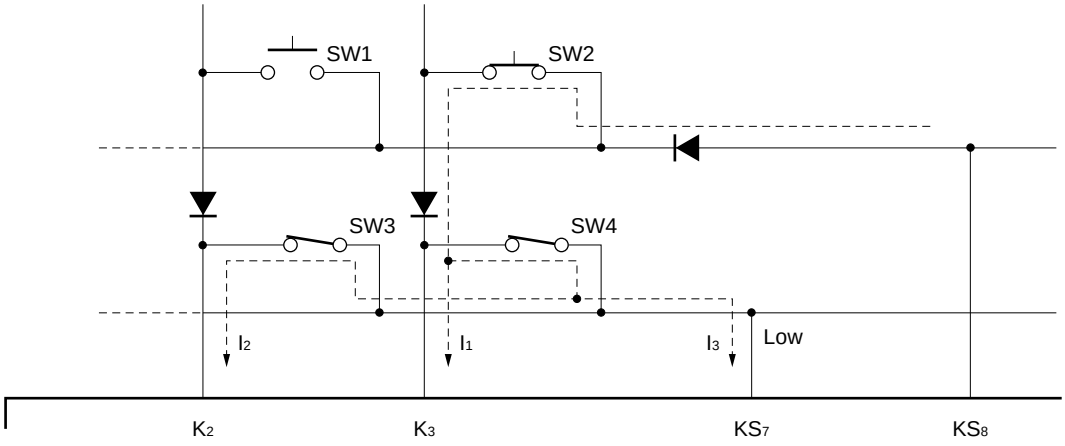
A circuit example where diode C is missing is shown below.

Suppose that switches SW2, SW3, and SW4 are on, and that a high level is output from KS_8, as shown below.

If diode C is missing at this tie, currents I_1, I_2, and I_3 shown by the dotted line will start to flow.

Therefore, a high level is input to K_2 because of I_2 even when switch SW1 is off, and it judges that SW1 is on.

Moreover, the high level of KS_8 is not correctly output because of I_3.



22.6.2 Inputting from alternate switch and diode switch

A program example is given below.

Example To read the statuses of the alternate and diode switches of the $LCD_{15}/P0Y_{15}/KS_{15}$ through $LCD_8/P0Y_8/KS_8$ pins to addresses 20H through 27H of BANK0 of the data memory.

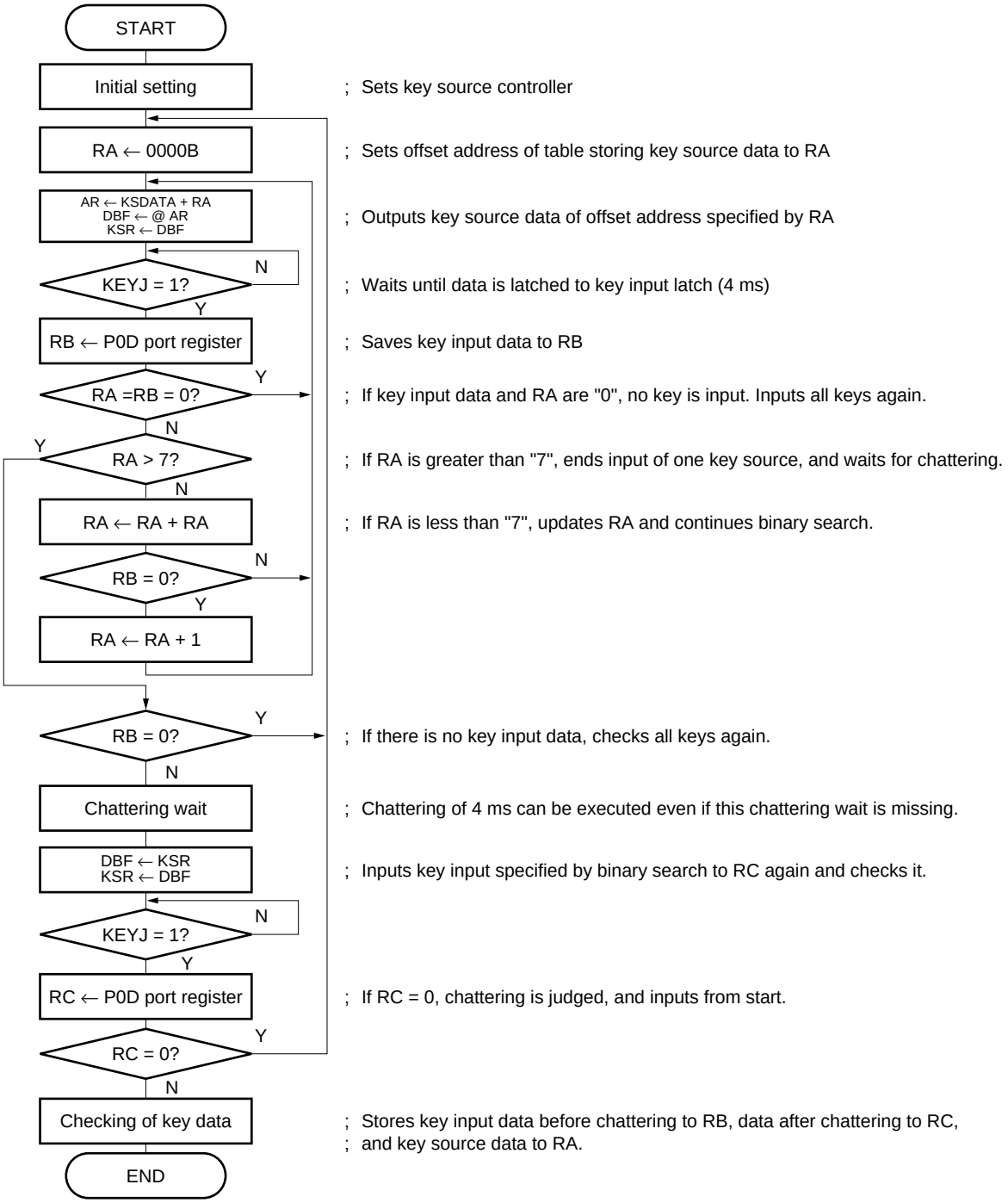
```
KS8         NIBBLE8  0.20H
KEY_IN      MEM      0.73H          ; P0D port register

KEY_LOAD:
            CLR1     P0YON          ; Sets LCD15/P0Y15/KS15-LCD8/P0Y8/KS8 pins
                                    ; as LCD segment pins
            SET2     LCDEN, KSEN    ; Outputs LCD segment and key source signals
            MOV      DBF3, #0000B   ; Sets key source data
            MOV      DBF2, #0001B   ; Outputs low level from KS8
            MOV      DBF1, #0000B
            MOV      DBF0, #0000B
            MOV      IXM, #0000B
            MOV      IXL, #0000B
            MOV      RPH, #0000B
            MOV      RPL, #0000B
KSCAN:
            PUT      KSR, DBF       ; Outputs signal of key source data
LOOP:
            SKF1     KEYJ           ; Judges if key input is latched
            BR       KCHECK
            Processing A            ; Waits until key input is latched
            BR       LOOP
KCHECK:
            MOV      RPL#.DM.KEY_IN SHR 3 AND 0EH
            SET1     IXE
            ST       KS8, KEY_IN    ; Stores key input data to data memory
            CLR1     IXE
            MOV      RPL, #0000B
            INC      IX
            ADD      DBF2, DBF2     ; Updates value of key source data and,
            ADD      DBF3, DBF3     ; scans key again
            SKT1     CY             ; Judges if all key source lines are input
            BR       KSCAN
KEY_END:                            ; End of input
```

22.6.3 Inputting momentary switch by binary search

The key source controller/decoder requires 4 ms to input the key of one key source signal line.
To input the keys of 16 key source signals, therefore, it takes 64 ms.
Therefore, the binary search method described in (1) and (2) are convenient.

(1) Flowchart

When KS_7 through KS_0 are used as key source signals of momentary switch



Example of table data for binary search

Shift address (RA)	Table data (key source data)															
	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
0000B									■	■	■	■	■	■	■	■
0001B									■	■	■	■				
0010B													■	■		
0011B									■	■						
0100B															■	
0101B													■			
0110B											■					
0111B									■							
1000B																■
1001B															■	
1010B														■		
1011B													■			
1100B												■				
1101B											■					
1110B										■						
1111B									■							

(2) Program example

```
RA      MEM     0.1AH                   ; General-purpose work register
RB      MEM     0.1BH                   ; General-purpose work register
RC      MEM     0.1CH                   ; General-purpose work register
KEY_IN  MEM     0.73H                   ; P0D port register
KSDATA:
        ;       KKKKKKKKKKKKKKKK
        ;       SSSSSSSSSSSSSSSS
        ;       1111119876543210
        ;       543210
        DW      0000000011111111B       ; RA=0
        DW      0000000011110000B       ; RA=1
        DW      0000000000001100B       ; RA=2
        DW      0000000011000000B       ; RA=3
        DW      0000000000000010B       ; RA=4
        DW      0000000000001000B       ; RA=5
        DW      0000000000100000B       ; RA=6
        DW      0000000010000000B       ; RA=7
        DW      0000000000000001B       ; RA=8
        DW      0000000000000010B       ; RA=9
        DW      0000000000000100B       ; RA=10
        DW      0000000000001000B       ; RA=11
        DW      0000000000010000B       ; RA=12
        DW      0000000000100000B       ; RA=13
        DW      0000000001000000B       ; RA=14
        DW      0000000010000000B       ; RA=15
```

```
KEY_LOAD:
        CLR1    P0YON                       ; Sets LCD15/P0Y15/KS15-LCD8/P0Y8/KS8 pins
                                            ; as LCD segment pins
        SET2    LCDEN, KSEN                 ; Outputs LCD segment and key source signals
START:
        MOV     RA, #0000B
KSCAN:
        MOV     AR3, #.DL.KSDATA SHR 0CH AND 0FH
        MOV     AR2, #.DL.KSDATA SHR 8 AND 0FH
        MOV     AR1, #.DL.KSDATA SHR 4 AND 0FH
        MOV     AR0, #.DL.KSDATA AND 0FH
        MOV     RPL, #.DL. AR0 SHR 3 AND 0EH
        ADD     AR0, RA
        ADDC    AR1, #0
        ADDC    AR2, #0
        ADDC    AR3, #0
        MOV     RPL, #0
        MOVT    DBF, @AR                    ; Reads table data

        PUT     KSR, DBF                    ; Outputs signal of key source data
LOOP1:
        SKF1    KEYJ                        ; Judges if key input is latched
        BR      KCHECK
        Processing A                        ; Waits until key input is latched
        BR      LOOP1
KCHECK:
        MOV     PRL, #.DM.RB SHR 3 AND 0EH
        LD      RB, KEY_IN                  ; Stores key input data to RB
        SKNE    RA, #0000B                  ; All keys are checked?
        SKE     RB, #0000B
        BR      Key input
        BR      START                       ; No key input at all
Key input:
        SKLT    RA, #1000B                  ; One key source selected?
        BR      LASTCHK

                                            ; Unless one key source selected,
        ADD     RA, RA                      ; updates value of RA and scans key again
        SKE     RB, #0000B
        ADD     RA, #0001B
        BR      KSCAN
```

```
LASTCHK:
        MOV     RPL, #0
        SKNE    RB, #0000B              ; Key input of one key source?
        BR      START                   ; If not, it is judged as chattering
        ┌──────────────┐
        │Chattering wait│
        └──────────────┘
LOOP2:
        SKF1    KEYJ                    ; Judges if key input is latched
        BR      KEYDEC
        ┌──────────────┐
        │ Processing B │                ; Waits until key input is latched
        └──────────────┘
        BR      LOOP2
KEYDEC:
        MOV     RPL, #.DM.RC SHR 3 AND 0EH
        LD      RC, KEY_IN              ; Stores key input data to latch
        SET2    CAP, Z                  ; Compares key input data after
        SUB     RC, RB                  ; chattering wait with key input data
        SKT1    Z                       ; before chattering wait.
        BR      START                   ; If they differ
KEY_END:
                                        ; stores key source data to RA,
                                        ; key input data before chattering to RB,
                                        ; and key input data after chattering to RC,
                                        ; respectively.
```

22.7 Reset Status

22.7.1 At power-ON reset

The LCD_{29}/$P0F_3$ through LCD_0/$P0Y_0$/KS_0 pins are specified as LCD segment signal output pins, and output a low level (display off). Therefore, low-level key source signals are output.

22.7.2 On execution of clock stop instruction

The LCD_{29}/$P0F_3$ through LCD_0/$P0Y_0$/KS_0 pins are specified as LCD segment signal output pins, and output a low level (display off). Therefore, low-level key source signals are output.

22.7.3 At CE reset

If the key source signals are output, the output data are retained.

22.7.4 In halt status

If the key source signals are output, the output data are retained as is.

If key input is specified as the condition under which the halt status is released, the halt status is released when a high level is input to the $P0D_3$/ADC_5 through $P0D_0$/ADC_2 pins.

However, when the key source controller is used, the halt status is released only by the high level input during 220 μs in which the key source data is output.

To release the halt status by key input by using the key source controller, do not use the $P0D_3$/ADC_5 through $P0D_0$/ADC_2 pins for the A/D converter.

For how to release the halt status by key input, refer to **12.4 Halt Function**.

23. μPD17010 INSTRUCTION

23.1 Instruction Set

b_{14}-b_{11} BIN.				b_{15} HEX.	0		1	
0	0	0	0	0	ADD	r, m	ADD	m, #n4
0	0	0	1	1	SUB	r, m	SUB	m, #n4
0	0	1	0	2	ADDC	r, m	ADDC	m, #n4
0	0	1	1	3	SUBC	r, m	SUBC	m, #n4
0	1	0	0	4	AND	r, m	AND	m, #n4
0	1	0	1	5	XOR	r, m	XOR	m. #n4
0	1	1	0	6	OR	r, m	OR	m, #n4
0	1	1	1	7	INC	AR		
					INC	IX		
					MOVT	DBF, @AR		
					BR	@AR		
					CALL	@AR		
					RET			
					RETSK			
					EI			
					DI			
					RETI			
					PUSH	AR		
					POP	AR		
					GET	DBF, p		
					PUT	p, DBF		
					PEEK	WR, rf		
					POKE	rf, WR		
					RORC	r		
					STOP	s		
					HALT	h		
					NOP			
1	0	0	0	8	LD	r, m	ST	m, r
1	0	0	1	9	SKE	m, #n4	SKGE	m, #n4
1	0	1	0	A	MOV	@r, m	MOV	m, @r
1	0	1	1	B	SKNE	m, #n4	SKLT	m, #n4
1	1	0	0	C	BR	addr (page 0)	CALL	addr (page 0)
1	1	0	1	D	BR	addr (page 1)	MOV	m, #n4
1	1	1	0	E	BR	addr (page 2)	SKT	m, #n
1	1	1	1	F	BR	addr (page 3)	SKF	m, #n

23.2 Instruction List

Legend

AR	:	Address register
ASR	:	Address stack register indicated by stack pointer
addr	:	Program memory address (lower 11 bits)
BANK	:	Bank register
CMP	:	Compare flag
CY	:	Carry flag
DBF	:	Data buffer
h	:	Halt release condition
INTEF	:	Interrupt enable flag
INTR	:	Register automatically saved to stack when interrupt occurs
INTSK	:	Interrupt stack register
IX	:	Index register
MP	:	Data memory row address pointer
MPE	:	Memory pointer enable flag
m	:	Data memory address indicated by m_R, m_C
m_R	:	Data memory row address (higher)
m_C	:	Data memory column address (lower)
n	:	Bit position (4 bits)
n4	:	Immediate data (4 bits)
PAGE	:	Page (bits 12 and 11 of program counter)
PC	:	Program counter
p	:	Peripheral address
p_H	:	Peripheral address (higher 3 bits)
p_L	:	Peripheral address (lower 4 bits)
r	:	General register column address
rf	:	Register file address
rf_R	:	Register file row address (higher 3 bits)
rf_C	:	Register file column address (lower 4 bits)
SP	:	Stack pointer
s	:	Stop release condition
WR	:	Window register
(×)	:	Contents addressed by ×

Instruction	Mnemonic	Operand	Operation	Instruction Code			
				op code	Operand		
Addition	ADD	r, m	(r) ← (r) + (m)	00000	m_R	m_C	r
		m, #n4	(m) ← (m) + n4	10000	m_R	m_C	n4
	ADDC	r, m	(r) ← (r) + (m) + CY	00010	m_R	m_C	r
		m, #n4	(m) ← (m) + n4 + CY	10010	m_R	m_C	n4
	INC	AR	AR ← AR + 1	00111	000	1001	0000
		IX	IX ← IX + 1	00111	000	1000	0000
Subtraction	SUB	r, m	(r) ← (r) – (m)	00001	m_R	m_C	r
		m, #n4	(m) ← (m) – n4	10001	m_R	m_C	n4
	SUBC	r, m	(r) ← (r) – (m) – CY	00011	m_R	m_C	r
		m, #n4	(m) ← (m) – n4 – CY	10011	m_R	m_C	n4
Logical operation	OR	r, m	(r) ← (r) ∨ (m)	00110	m_R	m_C	r
		m, #n4	(m) ← (m) ∨ n4	10110	m_R	m_C	n4
	AND	r, m	(r) ← (r) ∧ (m)	00100	m_R	m_C	r
		m, #n4	(m) ← (m) ∧ n4	10100	m_R	m_C	n4
	XOR	r, m	(r) ← (r) ⊻ (m)	00101	m_R	m_C	r
		m, #n4	(m) ← (m) ⊻ n4	10101	m_R	m_C	n4
Judgment	SKT	m, #n	CMP ← 0, if (m) ∧ n=n, then skip	11110	m_R	m_C	n
	SKF	m, #n	CMP ← 0, if (m) ∧ n=0, then skip	11111	m_R	m_C	n
Comparison	SKE	m, #n4	(m) – n4, skip if zero	01001	m_R	m_C	n4
	SKNE	m, #n4	(m) – n4, skip if not zero	01011	m_R	m_C	n4
	SKGE	m, #n4	(m) – n4, skip if not borrow	11001	m_R	m_C	n4
	SKLT	m, #n4	(m) – n4, skip if borrow	11011	m_R	m_C	n4
Rotation	RORC	r	CY → $(r)_{b3}$ → $(r)_{b2}$ → $(r)_{b1}$ → $(r)_{b0}$	00111	000	0111	r
Transfer	LD	r, m	(r) ← (m)	01000	m_R	m_C	r
	ST	m, r	(m) ← (r)	11000	m_R	m_C	r
	MOV	@r, m	if MPE=1: (MP, (r))← (m) if MPE=0: (BANK, m_R, (r))← (m)	01010	m_R	m_C	r
		m, @r	if MPE=1: (m)← (MP, (r)) if MPE=0: (m)← (BANK, m_R, (r))	11010	m_R	m_C	r
		m, #n4	(m) ← n4	11101	m_R	m_C	n4
	MOVT	DBF, @AR	SP ← SP–1, ASR ← PC, PC ← AR, DBF ← (PC), PC ← ASR, SP ← SP+1	00111	000	0001	0000

Instruction	Mnemonic	Operand	Operation	Instruction Code			
				op code	Operand		
Transfer	PUSH	AR	SP ← SP–1, ASR ← AR	00111	000	1101	0000
	POP	AR	AR ← ASR, SP ← SP + 1	00111	000	1100	0000
	PEEK	WR, rf	WR ← (rf)	00111	rf_R	0011	rf_C
	POKE	rf, WR	(rf) ← WR	00111	rf_R	0010	rf_C
	GET	DBF, p	DBF ← (p)	00111	p_H	1011	p_L
	PUT	p, DBF	(p) ← DBF	00111	p_H	1010	p_L
Branch	BR	addr	PC_{10-0} ← addr, PAGE ← 0	01100	addr		
			PC_{10-0} ← addr, PAGE ← 1	01101			
			PC_{10-0} ← addr, PAGE ← 2	01110			
			PC_{10-0} ← addr, PAGE ← 3	01111			
		@AR	PC ← AR	00111	000	0100	0000
Subroutine	CALL	addr	SP ← SP–1, ASR ← PC, PC_{10-0} ← addr, PAGE ← 0	11100	addr		
		@AR	SP ← SP–1, ASR ← PC, PC ← AR	00111	000	0101	0000
	RET		PC ← ASR, SP ← SP + 1	00111	000	1110	0000
	RETSK		PC ← ASR, SP ← SP + 1 and skip	00111	001	1110	0000
	RETI		PC ← ASR, INTR ← INTSK, SP ← SP + 1	00111	100	1110	0000
Interrupt	EI		INTEF ← 1	00111	000	1111	0000
	DI		INTEF ← 0	00111	001	1111	0000
Others	STOP	s	STOP	00111	010	1111	s
	HALT	h	HALT	00111	011	1111	h
	NOP		No operation	00111	100	1111	0000

23.3 Assembler (AS17K) Embedded Macro Instructions

Legend

flag n : FLG symbol
n : Bit number
< > : Can be omitted

	Mnemonic	Operand	Operation	n
Embedded	SKTn	flag 1, … flag n	if (flag 1) to (flag n) = all “1”, then skip	1 ≤ n ≤ 4
macro	SKFn	flag 1, … flag n	if (flag 1) to (flag n) = all “0”, then skip	1 ≤ n ≤ 4
	SETn	flag 1, … flag n	(flag 1) to (flag n) ← 1	1 ≤ n ≤ 4
	CLRn	flag 1, … flag n	(flag 1) to (flag n) ← 0	1 ≤ n ≤ 4
	NOTn	flag 1, … flag n	if (flag n) = “0”, then (flag n) ← 1 if (flag n) = “1”, then (flag n) ← 0	1 ≤ n ≤ 4
	INITFLG	<NOT>flag 1, …<<NOT>flag n>	if description = NOT flag n, then (flag n) ← 0 if description = flag n, then (flag n) ← 1	1 ≤ n ≤ 4
	BANKn		(BANK) ← n	0 ≤ n ≤ 3

24. μPD17010 RESERVED WORDS

24.1 Reserved Word List

24.1.1 System register (SYSREG)

Symbol Name	Attribute	Value	R/W	Description
AR3	MEM	0.74H	R/W	Bits 15-12 of address register
AR2	MEM	0.75H	R/W	Bits 11-8 of address register
AR1	MEM	0.76H	R/W	Bits 7-4 of address register
AR0	MEM	0.77H	R/W	Bits 3-0 of address register
WR	MEM	0.78H	R/W	Window register
BANK	MEM	0.79H	R/W	Bank register
IXH	MEM	0.7AH	R/W	Index register, high
MPH	MEM	0.7AH	R/W	Memory pointer, high
MPE	FLG	0.7AH.3	R/W	Memory pointer enable flag
IXM	MEM	0.7BH	R/W	Index register, middle
MPL	MEM	0.7BH	R/W	Memory pointer, low
IXL	MEM	0.7CH	R/W	Index register, low
RPH	MEM	0.7DH	R/W	General register pointer, high
RPL	MEM	0.7EH	R/W	General register pointer, low
PSW	MEM	0.7FH	R/W	Program status word
BCD	FLG	0.7EH.0	R/W	BCD flag
CMP	FLG	0.7FH.3	R/W	Compare flag
CY	FLG	0.7FH.2	R/W	Carry flag
Z	FLG	0.7FH.1	R/W	Zero flag
IXE	FLG	0.7FH.0	R/W	Index enable flag

24.1.2 Data buffer (DBF)

Symbol Name	Attribute	Value	R/W	Description
DBF3	MEM	0.0CH	R/W	Bits 15-12 of DBF
DBF2	MEM	0.0DH	R/W	Bits 11-8 of DBF
DBF1	MEM	0.0EH	R/W	Bits 7-4 of DBF
DBF0	MEM	0.0FH	R/W	Bits 3-0 of DBF

24.1.3 LCD segment register

Symbol Name	Attribute	Value	R/W	Description
LCDD0	MEM	0.60H	R/W	LCD segment register
LCDD1	MEM	0.61H	R/W	LCD segment register
LCDD2	MEM	0.62H	R/W	LCD segment register
LCDD3	MEM	0.63H	R/W	LCD segment register
LCDD4	MEM	0.64H	R/W	LCD segment register
LCDD5	MEM	0.65H	R/W	LCD segment register
LCDD6	MEM	0.66H	R/W	LCD segment register
LCDD7	MEM	0.67H	R/W	LCD segment register
LCDD8	MEM	0.68H	R/W	LCD segment register
LCDD9	MEM	0.69H	R/W	LCD segment register
LCDD10	MEM	0.6AH	R/W	LCD segment register
LCDD11	MEM	0.6BH	R/W	LCD segment register
LCDD12	MEM	0.6CH	R/W	LCD segment register
LCDD13	MEM	0.6DH	R/W	LCD segment register
LCDD14	MEM	0.6EH	R/W	LCD segment register

24.1.4 Port register

Symbol Name	Attribute	Value	R/W	Description
P0A3	FLG	0.70H.3	R/W	Bit 3 of port 0A
P0A2	FLG	0.70H.2	R/W	Bit 2 of port 0A
P0A1	FLG	0.70H.1	R/W	Bit 1 of port 0A
P0A0	FLG	0.70H.0	R/W	Bit 0 of port 0A
P0B3	FLG	0.71H.3	R/W	Bit 3 of port 0B
P0B2	FLG	0.71H.2	R/W	Bit 2 of port 0B
P0B1	FLG	0.71H.1	R/W	Bit 1 of port 0B
P0B0	FLG	0.71H.0	R/W	Bit 0 of port 0B
P0C3	FLG	0.72H.3	R/W	Bit 3 of port 0C
P0C2	FLG	0.72H.2	R/W	Bit 2 of port 0C
P0C1	FLG	0.72H.1	R/W	Bit 1 of port 0C
P0C0	FLG	0.72H.0	R/W	Bit 0 of port 0C
P0D3	FLG	0.73H.3	R	Bit 3 of port 0D
P0D2	FLG	0.73H.2	R	Bit 2 of port 0D
P0D1	FLG	0.73H.1	R	Bit 1 of port 0D
P0D0	FLG	0.73H.0	R	Bit 0 of port 0D
P0XL3	FLG	0.68H.3	R/W	Bit 1 of port 0X
P0XL2	FLG	0.68H.2	R/W	Bit 0 of port 0X
P0XL1	FLG	0.68H.1	R/W	Dummy
P0XL0	FLG	0.68H.0	R/W	Dummy

Symbol Name	Attribute	Value	R/W	Description
P0XH3	FLG	0.69H.3	R/W	Bit 5 of port 0X
P0XH2	FLG	0.69H.2	R/W	Bit 4 of port 0X
P0XH1	FLG	0.69H.1	R/W	Bit 3 of port 0X
P0XH0	FLG	0.69H.0	R/W	Bit 2 of port 0X
P0E3	FLG	0.6BH.3	R/W	Bit 3 of port 0E
P0E2	FLG	0.6BH.2	R/W	Bit 2 of port 0E
P0E1	FLG	0.6BH.1	R/W	Bit 1 of port 0E
P0E0	FLG	0.6BH.0	R/W	Bit 0 of port 0E
P0F3	FLG	0.6DH.3	R/W	Bit 3 of port 0F
P0F2	FLG	0.6DH.2	R/W	Bit 2 of port 0F
P0F1	FLG	0.6DH.1	R/W	Bit 1 of port 0F
P0F0	FLG	0.6DH.0	R/W	Bit 0 of port 0F
P1A3	FLG	1.70H.3	R/W	Bit 3 of port 1A
P1A2	FLG	1.70H.2	R/W	Bit 2 of port 1A
P1A1	FLG	1.70H.1	R/W	Bit 1 of port 1A
P1A0	FLG	1.70H.0	R/W	Bit 0 of port 1A
P1B3	FLG	1.71H.3	R/W	Bit 3 of port 1B
P1B2	FLG	1.71H.2	R/W	Bit 2 of port 1B
P1B1	FLG	1.71H.1	R/W	Bit 1 of port 1B
P1B0	FLG	1.71H.0	R/W	Bit 0 of port 1B
P1C3	FLG	1.72H.3	R/W	Bit 3 of port 1C
P1C2	FLG	1.72H.2	R/W	Bit 2 of port 1C
P1C1	FLG	1.72H.1	R/W	Bit 1 of port 1C
P1C0	FLG	1.72H.0	R/W	Bit 0 of port 1C
P1D3	FLG	1.73H.3	R	Bit 3 of port 1D
P1D2	FLG	1.73H.2	R	Bit 2 of port 1D
P1D1	FLG	1.73H.1	R	Bit 1 of port 1D
P1D0	FLG	1.73H.0	R	Bit 0 of port 1D
P2A3	FLG	2.70H.3	R/W	Bit 3 of port 2A
P2A2	FLG	2.70H.2	R/W	Bit 2 of port 2A
P2A1	FLG	2.70H.1	R/W	Bit 1 of port 2A
P2A0	FLG	2.70H.0	R/W	Bit 0 of port 2A

24.1.5 Register file (control register)

Symbol Name	Attribute	Value	R/W	Description
SP	MEM	0.81H	R/W	Stack pointer
SIO1TS	FLG	0.82H.3	R/W	SIO1 start flag
SIO1HIZ	FLG	0.82H.2	R/W	SIO1/P0B1 select flag
SIO1CK1	FLG	0.82H.1	R/W	Bit 1 of SIO1 clock select
SIO1CK0	FLG	0.82H.0	R/W	Bit 0 of SIO1 clock select
IFCGOSTT	FLG	0.84H.0	R	IF counter gate open status flag
PLLUL	FLG	0.85H.0	R	PLL unlock FF flag
ADCCMP	FLG	0.86H.0	R	ADC judge flag
CE	FLG	0.87H.0	R	CE pin status flag
SIO0CH	FLG	0.88H.3	R/W	SIO0 mode select flag
SB	FLG	0.88H.2	R/W	I^2C bus/serial I/O mode select flag
SIO0MS	FLG	0.88H.1	R/W	SIO0 clock mode select flag
SIO0TX	FLG	0.88H.0	R/W	SIO0 TX/RX select flag
BTM1CK1	FLG	0.89H.3	R/W	Basic timer 1 clock select flag
BTM1CK0	FLG	0.89H.2	R/W	Basic timer 1 clock select flag
BTM0CK1	FLG	0.89H.1	R/W	Basic timer 0 clock select flag
BTM0CK0	FLG	0.89H.0	R/W	Basic timer 0 clock select flag
TMCK3	FLG	0.8CH.3	R/W	Timer/counter clock select flag (dummy: 0)
TMCK2	FLG	0.8CH.2	R/W	Timer/counter clock select flag (dummy: 0)
TMCK1	FLG	0.8CH.1	R/W	Timer/counter clock select flag
TMCK0	FLG	0.8CH.0	R/W	Timer/counter clock select flag
TMOVF	FLG	0.8DH.0	R	Timer/counter overflow detect flag
TMRPT	FLG	0.8EH.2	R/W	12-bit timer repeat select flag
TMRES	FLG	0.8EH.1	R/W	Timer/counter reset flag
TMEN	FLG	0.8EH.0	R/W	Timer/counter enable flag
IGRPSL	FLG	0.8FH.0	R/W	Interrupt group select flag
KSEN	FLG	0.90H.1	R/W	Key source decoder enable flag
LCDEN	FLG	0.90H.0	R/W	LCD driver enable flag
R0YSEL	FLG	0.91H.3	R/W	Port 0Y select flag
P0XSEL	FLG	0.91H.2	R/W	Port 0X select flag
P0ESEL	FLG	0.91H.1	R/W	Port 0E select flag
P0FSEL	FLG	0.91H.0	R/W	Port 0F select flag
IFCMD1	FLG	0.92H.3	R/W	IF counter mode select flag
IFCMD0	FLG	0.92H.2	R/W	IF counter mode select flag
IFCCK1	FLG	0.92H.1	R/W	IF counter clock select flag
IFCCK0	FLG	0.92H.0	R/W	IF counter clock select flag
PWM2SEL	FLG	0.93H.3	R/W	PWM2 select flag
PWM1SEL	FLG	0.93H.2	R/W	PWM1 select flag
PWM0SEL	FLG	0.93H.1	R/W	PWM0 select flag
CGPSEL	FLG	0.93H.0	R/W	CGP select flag

Symbol Name	Attribute	Value	R/W	Description
ADCCH3	FLG	0.94H.3	R/W	AD mode select flag (dummy: 0)
ADCCH2	FLG	0.94H.2	R/W	AD mode select flag
ADCCH1	FLG	0.94H.1	R/W	AD mode select flag
ADCCH0	FLG	0.94H.0	R/W	AD mode select flag
PLULSEN3	FLG	0.95H.3	R/W	PLL unlock sensibility select flag (dummy: 0)
PLULSEN2	FLG	0.95H.2	R/W	PLL unlock sensibility select flag (dummy: 0)
PLULSEN1	FLG	0.95H.1	R/W	PLL unlock sensibility select flag
PLULSEN0	FLG	0.95H.0	R/W	PLL unlock sensibility select flag
KEYJ	FLG	0.96H.0	R	Key input judge flag
BTM0CY	FLG	0.97H.0	R	Basic timer 0 carry FF status flag
SBACK	FLG	0.98H.3	R/W	I^2C bus acknowledge flag
SIO0NWT	FLG	0.98H.2	R/W	SIO0 not wait flag
SIO0WRQ1	FLG	0.98H.1	R/W	SIO0 wait mode flag
SIO0WRQ0	FLG	0.98H.0	R/W	SIO0 wait mode flag
SIO0WSTT	FLG	0.99H.0	R	SIO0 wait status judge flag
IEG1	FLG	0.9FH.1	R/W	INT1 interrupt edge select flag
IEG0	FLG	0.9FH.0	R/W	INT0 interrupt edge select flag
PLLMD3	FLG	0.0A1H.3	R/W	PLL mode select flag (dummy: 0)
PLLMD2	FLG	0.0A1H.2	R/W	PLL mode select flag (dummy: 0)
PLLMD1	FLG	0.0A1H.1	R/W	PLL mode select flag
PLLMD0	FLG	0.0A1H.0	R/W	PLL mode select flag
IFCSTRT	FLG	0.0A3H.1	R/W	IF counter start flag
IFCRES	FLG	0.0A3H.0	R/W	IF counter reset flag
P0CGIO	FLG	0.0A7H.0	R/W	Port 0C group I/O select flag
SIO0SF8	FLG	0.0A8H.3	R	SIO0 clock counter status flag
SIO0SF9	FLG	0.0A8H.2	R	SIO0 clock counter status flag
SBSTT	FLG	0.0A8H.1	R	I^2C bus start condition status flag
SBBSY	FLG	0.0A8H.0	R	I^2C bus start/stop condition status flag
IPIFC	FLG	0.0AEH.1	R/W	IF counter interrupt permission flag
IPSIO0	FLG	0.0AEH.0	R/W	SIO0 interrupt permission flag
IPBTM1	FLG	0.0AFH.3	R/W	Basic timer 1 interrupt permission flag
IPTM	FLG	0.0AFH.2	R/W	12-bit timer interrupt permission flag
IPGRP	FLG	0.0AFH.1	R/W	Group interrupt permission flag
IP0	FLG	0.0AFH.0	R/W	INT0 interrupt permission flag
PLLRFCK3	FLG	0.0B1H.3	R/W	PLL reference clock select flag
PLLRFCK2	FLG	0.0B1H.2	R/W	PLL reference clock select flag
PLLRFCK1	FLG	0.0B1H.1	R/W	PLL reference clock select flag
PLLRFCK0	FLG	0.0B1H.0	R/W	PLL reference clock select flag

Symbol Name	Attribute	Value	R/W	Description
P1ABIO3	FLG	0.0B5H.3	R/W	P1A3 I/O select flag
P1ABIO2	FLG	0.0B5H.2	R/W	P1A2 I/O select flag
P1ABIO1	FLG	0.0B5H.1	R/W	P1A1 I/O select flag
P1ABIO0	FLG	0.0B5H.0	R/W	P1A0 I/O select flag
P0BBIO3	FLG	0.0B6H.3	R/W	P0B3 I/O select flag
P0BBIO2	FLG	0.0B6H.2	R/W	P0B2 I/O select flag
P0BBIO1	FLG	0.0B6H.1	R/W	P0B1 I/O select flag
P0BBIO0	FLG	0.0B6H.0	R/W	P0B0 I/O select flag
P0ABIO3	FLG	0.0B7H.3	R/W	P0A3 I/O select flag
P0ABIO2	FLG	0.0B7H.2	R/W	P0A2 I/O select flag
P0ABIO1	FLG	0.0B7H.1	R/W	P0A1 I/O select flag
P0ABIO0	FLG	0.0B7H.0	R/W	P0A0 I/O select flag
SIO0IMD3	FLG	0.0B8H.3	R/W	SIO0 interrupt mode select flag (dummy: 0)
SIO0IMD2	FLG	0.0B8H.2	R/W	SIO0 interrupt mode select flag (dummy: 0)
SIO0IMD1	FLG	0.0B8H.1	R/W	SIO0 interrupt mode select flag
SIO0IMD0	FLG	0.0B8H.0	R/W	SIO0 interrupt mode select flag
SIO0CK3	FLG	0.0B9H.3	R/W	SIO0 shift clock select flag (dummy: 0)
SIO0CK2	FLG	0.0B9H.2	R/W	SIO0 shift clock select flag (dummy: 0)
SIO0CK1	FLG	0.0B9H.1	R/W	SIO0 shift clock select flag
SIO0CK0	FLG	0.0B9H.0	R/W	SIO0 shift clock select flag
IRQIFC	FLG	0.0BAH.0	R/W	IF counter interrupt request flag
IRQSIO0	FLG	0.0BBH.0	R/W	SIO0 interrupt request flag
IRQBTM1	FLG	0.0BCH.0	R/W	Basic timer 1 interrupt request flag
IRQTM	FLG	0.0BDH.0	R/W	12-bit timer interrupt request flag
INT1	FLG	0.0BEH.3	R/W	INT1 pin status flag
IRQGRP	FLG	0.0BEH.0	R/W	Group interrupt request flag
INT0	FLG	0.0BFH.3	R/W	INT0 pin status flag
IRQ0	FLG	0.0BFH.0	R/W	INT0 interrupt request flag

24.1.6 Peripheral hardware register

Symbol Name	Attribute	Value	R/W	Description
ADCR	DAT	02H	R/W	A/D converter V_{REF} data register
SIO1SFR	DAT	03H	R/W	SIO1 presettable shift register
SIO0SFR	DAT	04H	R/W	SIO0 presettable shift register
PWMR0	DAT	05H	R/W	PWM0 data register
PWMR1	DAT	06H	R/W	PWM1 data register
PWMR2	DAT	07H	R/W	PWM2 data register
LCDR0	DAT	08H	W	LCD group register 0
LCDR1	DAT	09H	W	LCD group register 1
LCDR2	DAT	0AH	W	LCD group register 2
LCDR3	DAT	0BH	W	LCD group register 3
LCDR4	DAT	0CH	W	LCD group register 4
P0X	DAT	0CH	W	Port 0X group register
LCDR5	DAT	0DH	W	LCD group register 5
LCDR6	DAT	0EH	W	LCD group register 6
LCDR7	DAT	0FH	W	LCD group register 7
CGPR	DAT	20H	R/W	CGP data register
AR	DAT	40H	R/W	Address register of GET/PUT/PUSH/POP/CALL/BR/ MOVT/INC instruction
PLLR	DAT	41H	R/W	PLL data register
KSR	DAT	42H	R/W	Key source data register
P0Y	DAT	42H	R/W	Port 0Y group register
IFC	DAT	43H	R	IF counter data register
TMM	DAT	46H	R/W	Timer modulo register
TMC	DAT	47H	R	Timer/counter

24.1.7 Others

Symbol Name	Attribute	Value	Description
DBF	DAT	0FH	Fixed operand value of PUT, GET, and MOVT instructions
IX	DAT	01H	Fixed operand value of INC instruction

25. ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings (T_A = 25 ± 2 °C)

Parameter	Symbol	Condition	Rating	Unit
Supply voltage	V_{DD}		–0.3 to +6.0	V
Input voltage	V_I		–0.3 to V_{DD} +0.3	V
Output voltage	V_O	Except $P1B_1$-$P1B_3$, $P0A_2$, $P0A_3$	–0.3 to V_{DD} +0.3	V
Output withstand voltage	V_{BDS1}	$P1B_1$-$P1B_3$	18.0	V
	V_{BDS2}	$P0A_2$, $P0A_3$	V_{DD}+0.3	V
High-level output current	I_{OH}	1 pin	–12	mA
		Total of $P2A_0$, LCD_0-LCD_{29} pins	–25	mA
		Total of all pins except above	–40	mA
Low-level output current	I_{OL}	1 pin of $P0A_0$-$P0A_3$, $P1A_1$-$P1A_3$, $P2A_0$	15	mA
		1 pin other than above	10	mA
		Total of $P0A_0$-$P0A_3$, $P1A_1$-$P1A_3$, $P2A_0$	50	mA
		Total of all pins other than above	20	mA
Total power dissipation[Note]	Pt		450	mW
Operating ambient temperature	T_A		–40 to +85	°C
Storage temperature	T_{stg}		–55 to +125	°C

Note Refer to **Calculation of Total Dissipation** on next page.

Caution If the absolute maximum rating of even one of the above parameters is exceeded even momentarily, the product may be damaged. The absolute maximum ratings specify the values which if exceeded may cause the product to be physically damaged. Be sure not to exceed these ratings when using the product.

Recommended Operating Range

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Supply voltage	V_{DD1}	When PLL and CPU operate	4.5	5.0	5.5	V
	V_{DD2}	When PLL stops and CPU operates	3.5	5.0	5.5	V
Data retention voltage	V_{DDR}	When crystal resonator stops	2.2		5.5	V
Supply voltage rise time	t_{rise}	V_{DD} = 0 → 4.5 V			500	ms
Input amplitude	V_{IN1}	VCOL, VCOH	0.5		V_{DD}	V_{p-p}
	V_{IN2}	AMIFC, FMIFC	0.5		V_{DD}	V_{p-p}
Output withstand voltage	V_{BDS}	$P1B_1$-$P1B_3$			16.0	V
Operating ambient temperature	T_A		–40		+85	°C

Calculation of Total Dissipation

The μPD17010 dissipates the following three types of powers, and the sum of these three types of powers must be lower than total dissipation Pt (use at lower than about 80% of the rated dissipation is recommended).

<1> CPU dissipation : Calculated as V_{DD} (MAX.) × I_{DD} (MAX.)
<2> Output pin dissipation : Total of dissipation when maximum current is allowed to flow into each output pin
<3> Dissipation by pull-down resistor : Power dissipation by internal pull-down resistor

Here is an example:

Example Assume that the following currents flow into the output pins:

- High-level output : $P2A_0$ pin : 12 mA
  LCD_0 pin : 12 mA
  LCD_1 pin : 1 mA
  $P0B_0$-$P0B_2$ pins : 12 mA
  $P0B_3$ pin : 4 mA
- Low-level output : $P0A_0$-$P0A_2$ pins : 15 mA
  $P0A_3$ pin : 5 mA
  $P0C_0$, $P0C_1$ pins : 10 mA

Also assume that a current of 0.3 mA flows into the $P0D_0$ through $P0D_3$ pins with the internal resistor on.

<1> CPU dissipation: 5.5 V × 15 mA = 82.5 mW
<2> Output pin dissipation: $P2A_0$ pin ... 2.4 V × 12 mA = 28.8 mW
LCD_0 pin ... 3 V × 12 mA = 36 mW
LCD_1 pin ... 1 V × 1 mA = 1 mW
Total of $P0B_0$-$P0B_2$ pins ... 2.4 V × 12 mA × 3 = 86.4 mW
$P0B_3$ pin ... 1 V × 4 mA = 4 mW
Total of $P0A_0$-$P0A_2$ pins ... 2 V × 15 mA × 3 = 90 mW
$P0A_3$ pin ... 2 V × 5 mA = 10 mA
Total of $P0C_0$, $P0C_1$ pins ... 2 V × 10 mA × 2 = 40 mW
<3> Pull-down resistor dissipation: total of $P0D_0$-$P0D_3$ pins ... 5.5 V × 0.3 mA × 4 = 6.6 mW

Pt = <1> + <2> + <3> = 82.5 + (28.8 + 36 + 1 + 86.4 + 4 + 90 + 10 + 40) + 6.6 = 385.3 mW

Because the absolute maximum value of the total dissipation is 450 mW, it is considered that this rating is not exceeded in the above example.

However, design your system taking into consideration the above description.

DC Characteristics (T_A = –40 to +85 °C, V_{DD} = 4.5 to 5.5 V)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Supply voltage	V_{DD1}	When CPU and PLL operate	4.5	5.0	5.5	V
	V_{DD2}	When CPU operates and PLL stops	3.5	5.0	5.5	V
Supply current	I_{DD1}	When CPU and PLL operate X_{IN} pin Sine wave input (f_{IN} = 4.5 MHz, V_{IN} = V_{DD}), T_A = 25 °C		1.2	2.4	mA
	I_{DD2}	When CPU operates and PLL stops When HALT instruction is used (20 instructions are executed every 1 ms) Sine wave input to X_{IN} pin (f_{IN} = 4.5 MHz, V_{IN} = V_{DD}), T_A = 25 °C		0.45	0.90	mA
Data retention voltage	V_{DDR1}	Power failure detection by timer FF. When crystal resonator is used	3.5		5.5	V
	V_{DDR2}	Power failure detection by timer FF. When crystal resonator stops	2.2		5.5	V
	V_{DDR3}	Retention of data memory (RAM)	2.0		5.5	V
Data retention current	I_{DDR1}	When crystal resonator stops T_A = 25 °C		2	5	μA
	I_{DDR2}	When crystal resonator stops V_{DD} = 5.0 V, T_A = 25 °C		2	3	μA
Middle-level output voltage	V_{OM1}	COM_0, COM_1 V_{DD} = 5 V	2.3	2.5	2.7	V
High-level input voltage	V_{IH1}	$P0A_0$-$P0A_3$, $P0B_0$-$P0B_3$, $P0C_0$-$P0C_3$, $P1A_0$-$P1A_3$, $P1D_0$-$P1D_3$ CE, INT_0, INT_1	0.8 V_{DD}		V_{DD}	V
	V_{IH2}	$P0D_0$-$P0D_3$	0.6 V_{DD}		V_{DD}	V
Low-level input voltage	V_{IL}	$P0A_0$-$P0A_3$, $P0B_0$-$P0B_3$, $P0C_0$-$P0C_3$, $P0D_0$-$P0D_3$, $P1A_0$-$P1A_3$, $P1D_0$-$P1D_3$, CE, INT_0, INT_1	0		0.2 V_{DD}	V
High-level output current	I_{OH1}	$P0A_0$, $P0A_1$, $P1A_1$-$P1A_3$, $P2A_0$ V_{OH} = V_{DD} – 2 V, V_{DD} = 5 V, T_A = 25 °C	–2.0	–10.0		mA
	I_{OH2}	$P0B_0$-$P0B_3$, $P0C_0$-$P0C_3$, $P1A_0$, $P1B_0$, $P1C_0$-$P1C_3$ V_{OH} = V_{DD} –1 V	–1.0	–5.0		mA
	I_{OH3}	LCD_0-LCD_{29}, EO_0, EO_1 V_{OH} = V_{DD} –1 V	–1.0	–4.0		mA
Low-level output current	I_{OL1}	$P0A_0$-$P0A_3$, $P1A_1$-$P1A_3$, $P2A_0$ V_{OL} = 2 V, V_{DD} = 5 V, T_A = 25 °C	5.0	15.0		mA
	I_{OL2}	$P0B_0$-$P0B_3$, $P0C_0$-$P0C_3$, $P1B_0$, $P1C_0$-$P1C_3$ V_{OL} = 1 V	1.0	7.0		mA
	I_{OL3}	LCD_0-LCD_{29}, EO_0, EO_1 V_{OL} = 1 V	1.0	3.5		mA
	I_{OL4}	$P1B_1$-$P1B_3$ V_{OL} = 1 V	1.0	2.0		mA
High-level input current	I_{IH1}	When VCOH pulled down V_{IH} = V_{DD}	0.1	0.8		mA
	I_{IH2}	When VCOL pulled down V_{IH} = V_{DD}	0.1	0.8		mA
	I_{IH3}	When X_{IN} pulled down V_{IH} = V_{DD}	0.1	1.3		mA
	I_{IH4}	When $P0D_0$-$P0D_3$ pulled down V_{IH} = V_{DD}	0.05	0.13	0.30	mA
Output off leakage current	I_{L1}	$P0A_2$, $P0A_3$ V_{OH} = V_{DD}			500	nA
	I_{L2}	$P1B_1$-$P1B_3$ V_{OH} = 16 V			500	nA
	I_{L3}	EO_0, EO_1 V_{OH} = V_{DD}, V_{OL} = 0 V			±100	nA

AC Characteristics (T_A = –40 to + 85 °C, V_{DD} = 4.5 to 5.5 V)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{IN1}	VCOL MF mode, sine wave input, V_{IN} = 0.2 V_{p-p}	0.5		30	MHz
	f_{IN2}	VCOL HF mode, sine wave input, V_{IN} = 0.2 V_{p-p}	5		40	MHz
	f_{IN3}	VCOH, sine wave input, V_{IN} = 0.2 V_{p-p}	9		150	MHz
	f_{IN4}	AMIFC, sine wave input, V_{IN} = 0.5 V_{p-p}	0.1		1	MHz
	f_{IN5}	AMIFC, sine wave input, V_{IN} = 0.05 V_{p-p}	0.44		0.46	MHz
	f_{IN6}	FMIFC, sine wave input, V_{IN} = 0.5 V_{p-p}	5		15	MHz
	f_{IN7}	FMIFC, sine wave input, V_{IN} = 0.06 V_{p-p}	10.5		10.9	MHz
AD conversion resolution					6	bit
AD conversion total error		T_A = –10 to + 50 °C		±1	±1.5	LSB

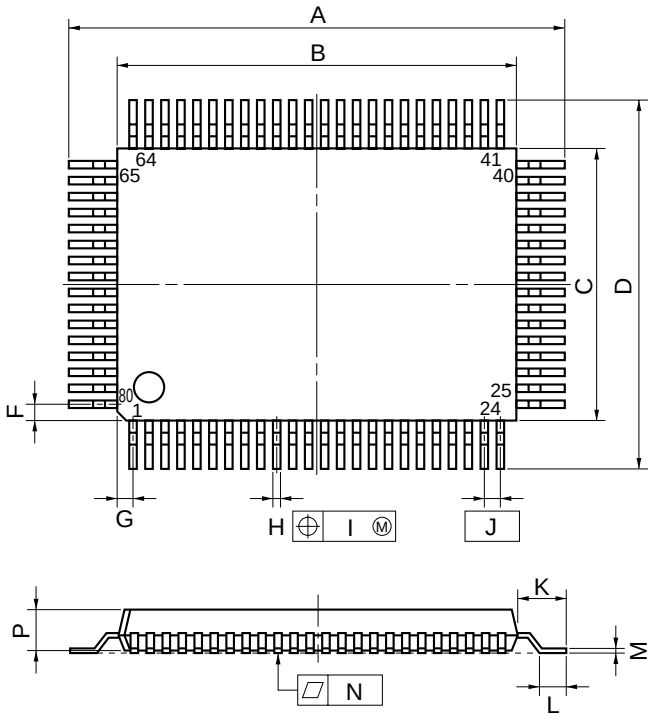
Reference Characteristics

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Supply current	I_{DD3}	When CPU and PLL operate VCOH sine wave input, f_{IN} = 150 MHz, V_{IN} = 0.3 V_{p-p} V_{DD} = 5 V, T_A = 25 °C		15		mA
High-level output current	I_{OH4}	COM_0, COM_1 V_{OH} = V_{DD} –1 V		–0.2		mA
Middle-level output current	I_{OM1}	COM_0, COM_1 V_{OM} = V_{DD} –1 V		–20		μA
	I_{OM2}	COM_0, COM_1 V_{OM} = 1 V		20		μA
Low-level output current	I_{OL5}	COM_0, COM_1 V_{OL} = 1 V		0.2		mA

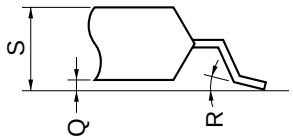
26. PACKAGE

(a) Package for mass production

80 PIN PLASTIC QFP (14×20)



detail of lead end



NOTE

Each lead centerline is located within 0.15 mm (0.006 inch) of its true position (T.P.) at maximum material condition.

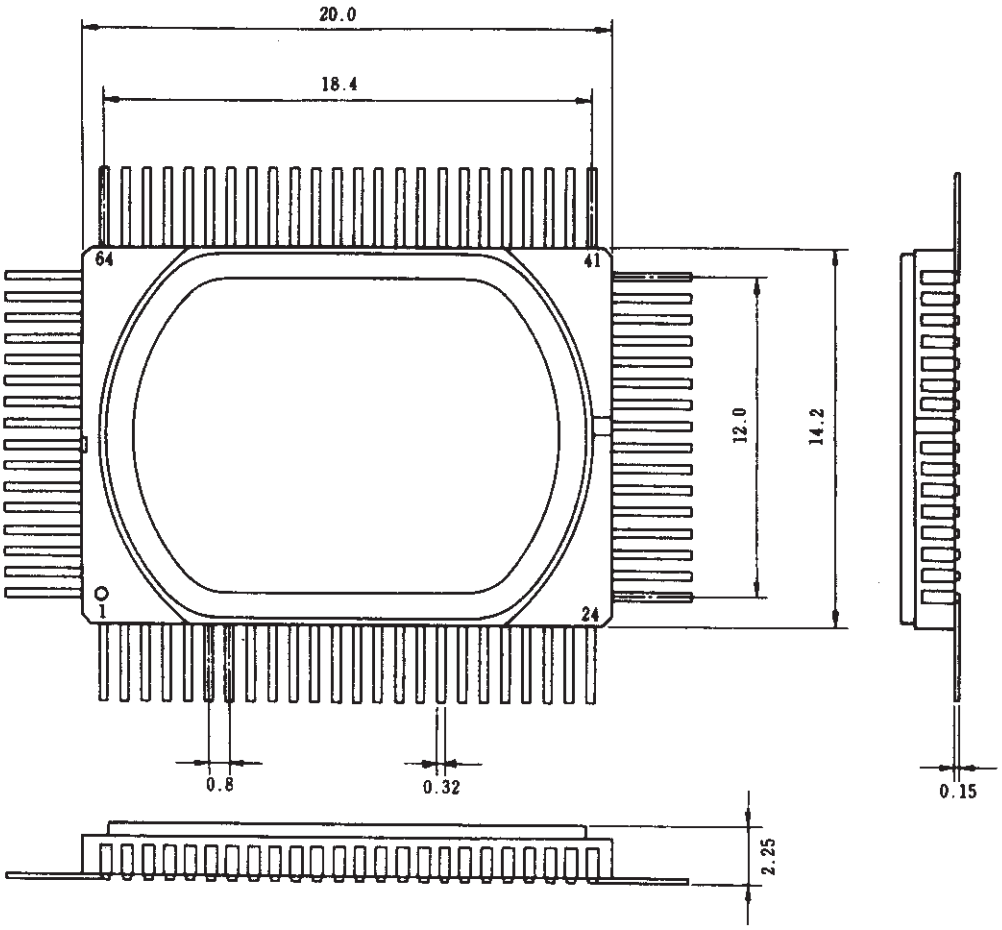
Caution The ES model is different from the mass-production model in package and materials. Refer to (b) Package of ES model.

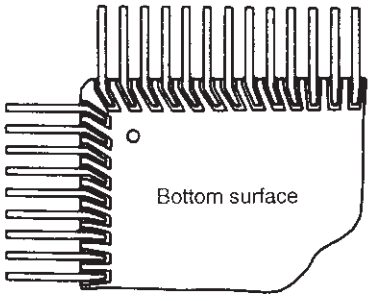
ITEM	MILLIMETERS	INCHES
A	23.2±0.2	$0.913^{+0.009}_{-0.008}$
B	20.0±0.2	$0.787^{+0.009}_{-0.008}$
C	14.0±0.2	$0.551^{+0.009}_{-0.008}$
D	17.2±0.2	0.677±0.008
F	1.0	0.039
G	1.8	0.031
H	0.35±0.10	$0.014^{+0.004}_{-0.005}$
I	0.15	0.006
J	0.8 (T.P.)	0.031 (T.P.)
K	1.6±0.2	0.063±0.008
L	0.8±0.2	$0.031^{+0.009}_{-0.008}$
M	$0.15^{+0.10}_{-0.05}$	$0.006^{+0.004}_{-0.003}$
N	0.12	0.005
P	2.7	0.106
Q	0.125±0.075	0.005±0.003
R	5°±5°	5°±5°
S	3.0 MAX.	0.119 MAX.

S80GF-80-3B9-2

(b) Package of ES model

80 PIN CERAMIC QFP FOR ES (REFERENCE) (UNIT: mm)





Cautions 1. The leads are molded diagonally at the bottom.
2. Cutting the tip of the leads is not a quality control target. Therefore, the lead length is not specified.

27. RECOMMENDED SOLDERING CONDITIONS

Solder this product under the following recommended conditions.

For the details of the recommended soldering conditions, refer to Information document **Semiconductor Device Mounting Technology Manual (IEI-1207)**.

For the soldering methods and conditions other than those recommended, consult NEC.

Table 27-1. Soldering Conditions of Surface Mount Type

μPD17010GF-×××-3B9: 80-pin plastic QFP (14 × 20 mm)
μPD17010GF-E××-3B9: 80-pin plastic QFP (14 × 20 mm)

Soldering Method	Soldering Condition	Symbol of Recommended Condition
Infrared reflow	Package peak temperature: 235 °C, Time: 30 seconds MAX. (210 °C MIN.), Number of times: 2 MAX., Number of days: 7[Note] (After this, 20 hours of prebaking is necessary at 125 °C.) <**Precaution**> (1) Start second reflow after the device temperature that has risen due to the first reflow has dropped to room temperature. (2) Do not clean flux with water after the first reflow.	IR35-207-2
VPS	Package peak temperature: 215 °C, Time: 40 seconds MAX. (200 °C MIN.), Number of times: 2 MAX., Number of days: 7[Note] (After this, 20 hours of prebaking is necessary at 125 °C.) <**Precaution**> (1) Start second reflow after the device temperature that has risen due to the first reflow has dropped to room temperature. (2) Do not clean flux with water after the first reflow.	VP15-207-2
Wave soldering	Soldering bath temperature: 260 °C MAX., Time: 10 seconds MAX., Number of times: 1, Preheating temperature: 120 °C MAX. (package surface temperature), Number of days: 7[Note] (After this, 20 hours of prebaking is necessary at 125 °C.)	WS60-207-1
Pin partial heating	Pin temperature: 300 °C MAX., Time: 3 seconds MAX. (per side of device)	—

Note Number of days for storage after the dry pack was opened. The storage conditions are at 25 °C, 65% RH MAX.

Caution Do not use two or more soldering methods in combination (except pin partial heating).

APPENDIX A. NOTES ON CONNECTING CRYSTAL RESONATOR

When connecting a crystal resonator, wire the portion enclosed by a dotted line in Figure A-1 below as follows to prevent the adverse influence of the circuit capacitance:

- Keep the wiring length as short as possible.
- Do not cross the wiring with any other signal lines. Do not route the wiring in the vicinity of lines through which a high alternating current flows.
- Keep the ground point of the capacitor of the oscillation circuit at the same potential as GND. Do not ground the circuit to a ground pattern through which a high current flows.
- Do not extract signals from the oscillation circuit.

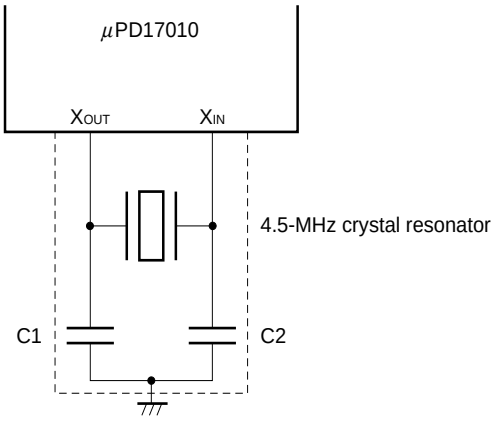
When connecting the capacitor or adjusting the oscillation frequency, keep in mind the following points (1) through (3):

(1) If the capacitances C1 and C2 are too high, the oscillation characteristics may be degraded and the current dissipation may increase.

(2) Generally, connect the trimmer capacitor for oscillation frequency adjustment to the X_{IN} pin. However, depending on the crystal resonator to be used, the oscillation stability may be degraded if the trimmer capacitor is connected to the X_{IN} pin (in this case, connect the trimmer capacitor to the X_{OUT} pin). Therefore, evaluate oscillation by using the crystal resonator actually used.

(3) Adjust the oscillation frequency by measuring the LCD drive waveform (125 Hz) or VCO oscillation frequency. If a probe is connected to the X_{OUT} or X_{IN} pin, accurate measurement cannot be made due to the capacitance of the probe.

Figure A-1. Connecting Crystal Resonator



APPENDIX B. DIFFERENCES AMONG μPD17010, μPD17003A, AND μPD17005

(1) Function List

Item	μPD17003A	μPD17005	μPD17010
ROM	8K bytes (3836 × 16 bits)	16K bytes (7932 × 16 bits)	
RAM	320 × 4 bits	432 × 4 bits	
Output port	9 (+30: LCD segment pins)		13 (+30: LCD segment pins)
Control register	33 × 4 bits		41 × 4 bits
General register pointer	4 bits		5 bits
Stack level	7 bits		9 bits
Serial interface	• SIO1 clock 75, 150, 225, 450 kHz • SIO2 clock External, 75, 150, 450 kHz		• SIO0 clock 37.5, 75, 112.5, 225 kHz • SIO1 clock External, 37.5, 75, 450 kHz Hysteresis characteristics of SCL, SDA, $\overline{SCK_0}$, $\overline{SCK_1}$, SI_0, SI_1 pins
D/A converter frequency	878.9 Hz		4394.5 Hz
Interrupt	• 5 External : 2 (INT_0 and INT_1 pins) Internal : 3 (TM, SIO1, IFC) • Interrupt priority (vector address) 1. (5H) INT_0 pin 2. (4H) INT_1 pin 3. (3H) Timer 4. (2H) Serial interface 1 5. (1H) Frequency counter • System register automatic saving (4 levels) (BANK, IXE)		• 6 External : 1 (INT_0 pin) Internal : 4 (TM, BTM1, SIO0, IFC) External/internal: 1 (INT_1 pin or timer/counter overflow) • Interrupt priority (vector address) 1. (6H) INT_0 pin 2. (5H) INT_1 pin (shared with timer/counter overflow) 3. (4H) 12-bit timer 4. (3H) Basic timer 1 5. (2H) Serial interface 0 6. (1H) Frequency counter • System register automatic saving (3 levels) (WR, BANK, RP, PSWORD) • Address modification of IRQ××× flag
Timer	• Timer carry (Clock: 4, 10, 200, 1000 Hz) • Timer interrupt (Clock: 4, 10, 200, 1000 Hz)		• Basic timer 0 carry (Clock: 4, 10, 200, 1000 Hz) • Basic timer 1 interrupt (Clock: 4, 10, 200, 1000 Hz) • 12-bit timer (Clock: 1, 3, 90, 100 kHz)
Operational amplifier for PLL frequency synthesizer low-pass filter	Provided		Not provided
One-time PROM model	μPD17P005		μPD17P010

(2) Development Tools

Item		μPD17003A	μPD17005	μPD17010
Hardware	SE board	SE-17010		
	Emulation probe	EP-17003GF		
Software	Device file	AS17003	AS17005	AS17010
	Macro library	• IFCSET. LIB • IRQ. MAC		None

(3) Notes on names of reserved words

Some reserved words of the control registers of the μPD17010 are different from those of the μPD17003A and 17005.

The following table shows the difference among the μPD17010, μPD17003A, and 17005 in reserved words.

Item	μPD17003A	μPD17005	μPD17010
Timer	TMMD3		BTM1CK1
	TMMD2		BTM1CK0
	TMMD1		BTM0CK1
	TMMD0		BTM0CK0
	TMCY		BTM0CY
			TMCK3
			TMCK2
			TMCK1
			TMCK0
			TMOVF
			TMRPT
			TMRES
			TMEN
PLL frequency synthesizer	PLULDYL3		PLULSEN3
	PLULDLY2		PLULSEN2
	PLULDLY1		PLULSEN1
	PLULDLY0		PLULSEN0
	PLLRFMD3		PLLRFCK3
	PLLRFMD2		PLLRFCK2
	PLLRFMD1		PLLRFCK1
	PLLRFMD0		PLLRFCK0
D/A converter	PWM2ON		PWM2SEL
	PWM1ON		PWM1SEL
	PWM0ON		PWM0SEL
	CGPON		CGPSEL
LCD driver	P0YON		P0YSEL
	P0XON		P0XSEL
	P0EON		P0ESEL
	P0FON		P0FSEL
IF	IFCG		IFCGOSTT

Item	μPD17003A	μPD17005	μPD17010
Serial interface	SIO2TS		SIO1TS
	SIO2HIZ		SIO1HIZ
	SIO2CK1		SIO1CK1
	SIO2CK0		SIO1CK0
	SIO1CH		SIO0CH
	SIO1MS		SIO0MS
	SIO1TX		SIO0TX
	SIO1NWT		SIO0NWT
	SIO1WRQ1		SIO0WRQ1
	SIO1WRQ0		SIO0WRQ0
			SIO0WSTT
	SIO1SF8		SIO0SF8
	SIO1SF9		SIO0SF9
	SIO1IMD3		SIO0IMD3
	SIO1IMD2		SIO0IMD2
	SIO1IMD1		SIO0IMD1
	SIO1IMD0		SIO0IMD0
	SIO1CK3		SIO0CK3
	SIO1CK2		SIO0CK2
	SIO1CK1		SIO0CK1
	SIO1CK0		SIO0CK0
Interrupt			IGRPSL
	IPSIO1		IPSIO0
	IPTM		IPBTM1
	IP1		IPGRP
			IPTM
	IRQSIO1		IRQSIO0
	IRQTM		IRQBTM1
			IRQTM
	IRQ1		IRQGRP

APPENDIX C. DEVELOPMENT TOOLS

The following tools are available to support development of the program of the μPD17010.

Hardware

Name	Function
In-circuit emulator (IE-17K IE-17K-ET[Note 1] EMU-17K[Note 2])	IE-17K, IE-17K-ET, and EMU-17K are in-circuit emulators that can be commonly used with 17K series. IE-17K and IE-17K-ET are connected to host machine, PC-9800 series or IBM PC/AT™, via RS-232C. EMU-17K is mounted to the expansion slot of host machine, PC-9800 series. When these in-circuit emulators are used with system evaluation board (SE board) dedicated to each model, they operate as emulators supporting that model. More sophisticated debugging environment can be created when man-machine interface software *SIMPLEHOST*™ is used. EMU-17K has function to check data memory contents real-time.
SE board (SE-17010)	SE-17010 is SE board for μPD17010 and 17P010. This board can be used alone for system evaluation or with in-circuit emulator for debugging.
Emulation probe (EP-17003GF)	EP-17003GF is emulation probe for μPD17010 and 17P010. When used with EV-9200G-80[Note 3], it connects SE board and target system.
Conversion socket (EV-9200G-80[Note 3])	EV-9200G-80 is conversion socket for 80-pin plastic QFP (14 × 20 mm). This is used to connect EP-17003GF and target system.
PROM programmer (AF-9703[Note 4] AF-9704[Note 4] AF-9705[Note 4] AF-9706[Note 4])	AF-9703, AF-9704, AF-9705, and AF-9706 are PROM programmers supporting μPD17P010. When connected with programmer adapter AF-9803, they can program μPD17P010.
Program adapter (AF-9803[Note 4])	AF-9803 is adapter for programming μPD17P010. This is used with AF-9703, AF-9704, AF-9705, or AF-9706.

Notes 1. Low-price model: external power supply type

2. Product of I.C. For details, consult I.C.

3. One EV-9200G-80 is provided to the EP-17003GF. Five EV-9200G-80s are separately available as a set.

4. These are products of Ando Electric. For details, consult Ando Electric.

Software

Name	Remark	Host Machine	OS		Distribution Media	Order Code
17K series assembler (AS17K)	AS17K is assembler that can be commonly used with 17K series. To develop program of μPD17010, this assembler and device file (AS17010) are used in combination.	PC-9800 series	MS-DOS™		5"2HD	μS5A10AS17K
					3.5"2HD	μS5A13AS17K
		IBM PC/AT	PC DOS™		5"2HC	μS7B10AS17K
					3.5"2HC	μS7B13AS17K
Device file (AS17010)	AS17010 is device file for μPD17010 and μPD17P010. This is used in combination with assembler (AS17K) for 17K series.	PC-9800 series	MS-DOS		5"2HD	μS5A10AS17010
					3.5"2HD	μS5A13AS17010
		IBM PC/AT	PC DOS		5"2HC	μS7B10AS17010
					3.5"2HC	μS7B13AS17010
Support software (*SIMPLEHOST*)	*SIMPLEHOST* is man-machine interface software that runs on Windows™ when program is developed by using in-circuit emulator and personal computer.	PC-9800 series	MS-DOS	Windows	5"2HD	μS5A10IE17K
					3.5"2HD	μS5A13IE17K
		IBM PC/AT	PC DOS		5"2HC	μS7B10IE17K
					3.5"2HC	μS7B13IE17K

Remark The version of the supported OS is as follows:

OS	Version
MS-DOS	Ver. 3.30 to Ver. 5.00A[Note]
PC DOS	Ver. 3.1 to Ver. 5.0[Note]
Windows	Ver. 3.0 to Ver. 3.1

Note Although MS-DOS Ver. 5.00/5.00A and PC DOS Ver.5.0 have a task swap function, this function cannot be used with this software.

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

Purchase of NEC I^2C components conveys a license under the Philips I^2C Patent Rights to use these components in an I^2C system, provided that the system conforms to the I^2C Standard Specification as defined by Philips.

***SIMPLHOST* is a trademark of NEC Corporation.**

MS-DOS and Windows are trademarks of Microsoft Corporation.

PC/AT and PC DOS are trademarks of IBM Corporation.

The export of this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.