DATA SHEET

NEC

MOS INTEGRATED CIRCUIT
μPD17068

4-BIT SINGLE-CHIP MICROCONTROLLER CONTAINING IMAGE DISPLAY CONTROLLER AND PLL FREQUENCY SYNTHESIZER FOR DIGITAL TUNING SYSTEMS

The μPD17068 is a 4-bit single-chip microcontroller for digital tuning systems. It contains an image display controller (IDC) that supports many types of display, and a PLL synthesizer.

The CPU of the μPD17068 is capable of 4-bit parallel addition, logical operations, bit tests, setting/resetting of a carry flag, and supports a powerful interrupt function and timer function.

The image display controller for on-screen display is user-programmable, allowing a range of displays to be programmed.

The peripheral hardware includes a full complement of I/O ports, controlled with powerful I/O instructions, as well as a serial interface, a 6-bit A/D converter, and an 8-bit D/A converter (PWM output).

FEATURES

- Program memory (ROM) : 24K bytes (12032 × 16 bits)
- Character ROM (CROM) : 4086 × 24 bits (255 characters)
- Data memory (RAM) : 1007 × 4 bits
- Video RAM (VRAM) : 672 × 4 bits (can be used for data memory)
- Address stack : 12 levels
- Interrupt stack : 2 levels
- Instruction execution time : 2 μs (when an 8 MHz crystal is used)
- PLL frequency synthesizer
- 8-bit serial interface
  (2 channels: One for two-wire or three-wire mode, compatible with I^2C bus, and one for three-wire mode only)
- D/A converter: 8 bits × 9 lines (PWM output)
- A/D converter: 6 bits × 8 lines
- Horizontal synchronizing signal counter
- Commercial power supply frequency counter
- Power-failure detection circuit and power-on reset circuit
- Interrupt input for remote-controller signal (with noise canceler)
- User-programmable image display controller (IDC)
  - Displayed characters : Up to 192 per screen (more characters can be displayed when the use of the entire screen is specified with a program)
  - Display mode : 16 × 16 dots in 15 lines × 24 columns; 14 × 16 dots in 17 lines × 24 columns
  - Character patterns : 255
  - Character format : 16 × 16 dots or 14 × 16 dots
  - Colors : 15
  - Character sizes : 16 sizes for height (can be specified per line); 24 sizes for width (can be specified per character)
- Many I/O ports
  - I/O : 19 ports
  - Input only : 4 ports
  - Output only : 21 ports
- Operating supply voltage: 5 V ±10 %
- Low power dissipation by use of CMOS technology

The information in this document is subject to change without notice.

Document No. IC-3525
(O.D. No. IC-8999)
Date Published November 1994 P
Printed in Japan

© NEC Corporation 1994

ORDERING INFORMATION

Part number	Package	Quality grade
μPD17068GF-×××-3BA	100-pin plastic QFP (14 × 20 mm)	Standard
μPD17068GF-E××-3BA[Note]	100-pin plastic QFP (14 × 20 mm)	Standard

Note Product supporting an I^2C bus interface. When using the I^2C bus interface (including implementation with a program that does not use peripheral hardware), make this point clear to your NEC sales representative when ordering mask options.

Remark ××× is a ROM code.

Please refer to “Quality grade on NEC Semiconductor Devices” (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

FUNCTION OVERVIEW

Item		Function
Program memory (ROM)		• 24K bytes (12032 × 16 bits) Table reference area: 12032 × 16 bits
Character ROM (CROM)		• 4086 × 24 bits (255 characters)
Data memory	RAM	• 1007 × 4 bits (including area also used for VRAM) Data buffers: 4 × 4 bits, general-purpose registers: 16 × 4 bits
Video RAM (VRAM)		• 672 × 4 bits (can be used for data memory (RAM))
System registers		• 12 × 4 bits
Register files		• 12 × 4 bits
General-purpose port registers		• 12 × 4 bits
Instruction execution time		• 2 μs (when 8 MHz crystal is used)
Stack levels		• 12 levels (stack manipulation possible)
General-purpose ports		• I/O : 19 ports Input only : 4 ports Output only : 21 ports
IDC (Image Display Controller)		• Displayed characters : Up to 192 per screen (more characters can be displayed when the use of the entire screen is specified with a program) • Display mode : 16 × 16 dots in 15 lines × 24 columns 14 × 16 dots in 17 lines × 24 columns • Character patterns : 255 (user-programmable) • Character format : 16 × 16 dots or 14 × 16 dots (2-dot interval can be specified between characters.) • Colors : 15 • Character sizes : 16 different heights (can be specified per line) 24 different widths (can be specified per character)
PLL frequency synthesizer		• Frequency division method : Pulse swallow • Reference frequency : 5, 6.25, 10, 12.5, and 25 kHz • Contains a charge pump for an external low-pass filter • Phase comparator : Unlock can be detected with a program. The delay for the unlock flip-flop is selectable.
Serial interface		• 2 channels Serial interface 0 (two-wire or three-wire mode, compatible with I^2C bus) Serial interface 1 (three-wire mode only)
D/A converter		• 8 bits × 9 lines (PWM output with withstand voltage of 12.5 V max.)
A/D converter		• 6 bits × 8 lines (successive approximation system with software)
Interrupts		• 10 channels (maskable interrupts) External interrupts : 3 channels (INT_0, INT_{NC}, and $\overline{V_{SYNC}}/\overline{H_{SYNC}}$) Internal interrupts : 7 channels (timers 0 and 1, serial interfaces 0 and 1, basic timer 2, VRAM pointer, and timer 0 overflow)

Item	Function
Timers	Timer 0 : 10 μs to 204.75 ms (interrupt) Timer 1 : 1 μs to 256 ms (interrupt) Basic timer 0 : 1, 5, and 100 ms (carry) Basic timer 1 : 125 μs, 1 ms, 5 ms, 100 ms, and external (carry) Basic timer 2 : 125 μs, 1 ms, 5 ms, 100 ms, and external (interrupt) Watch timer : Day, hour, minute, and second (count value)
Reset	• Power-on reset • Reset with the CE pin (by switching the CE pin from low to high) • Power-failure detection function
Supply voltage	5 V±10%
Package	100-pin plastic QFP (14 × 20 mm)

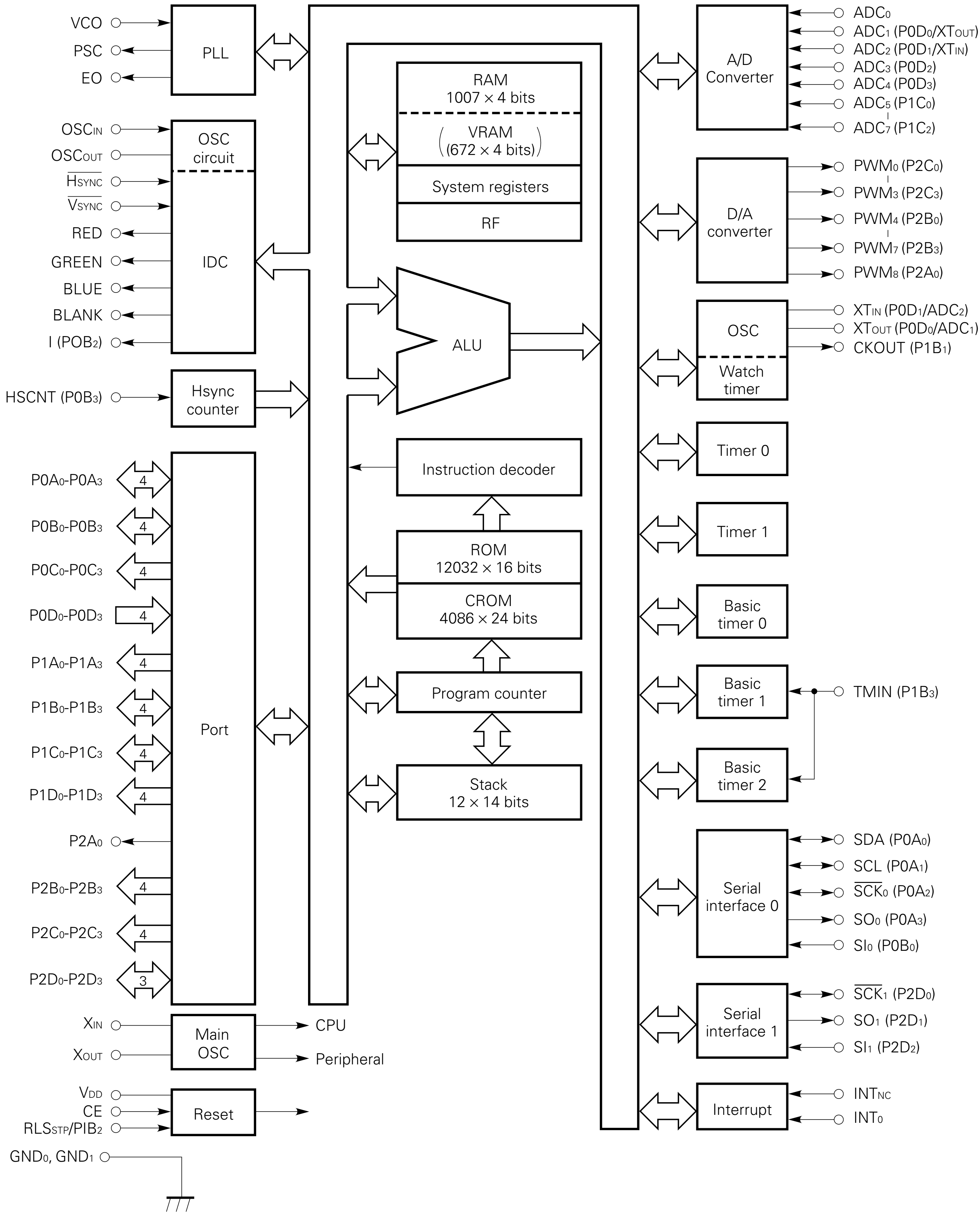
Remark Parentheses for timers indicate how to obtain the elapsed time for each timer.

Interrupt : Receiving an interrupt

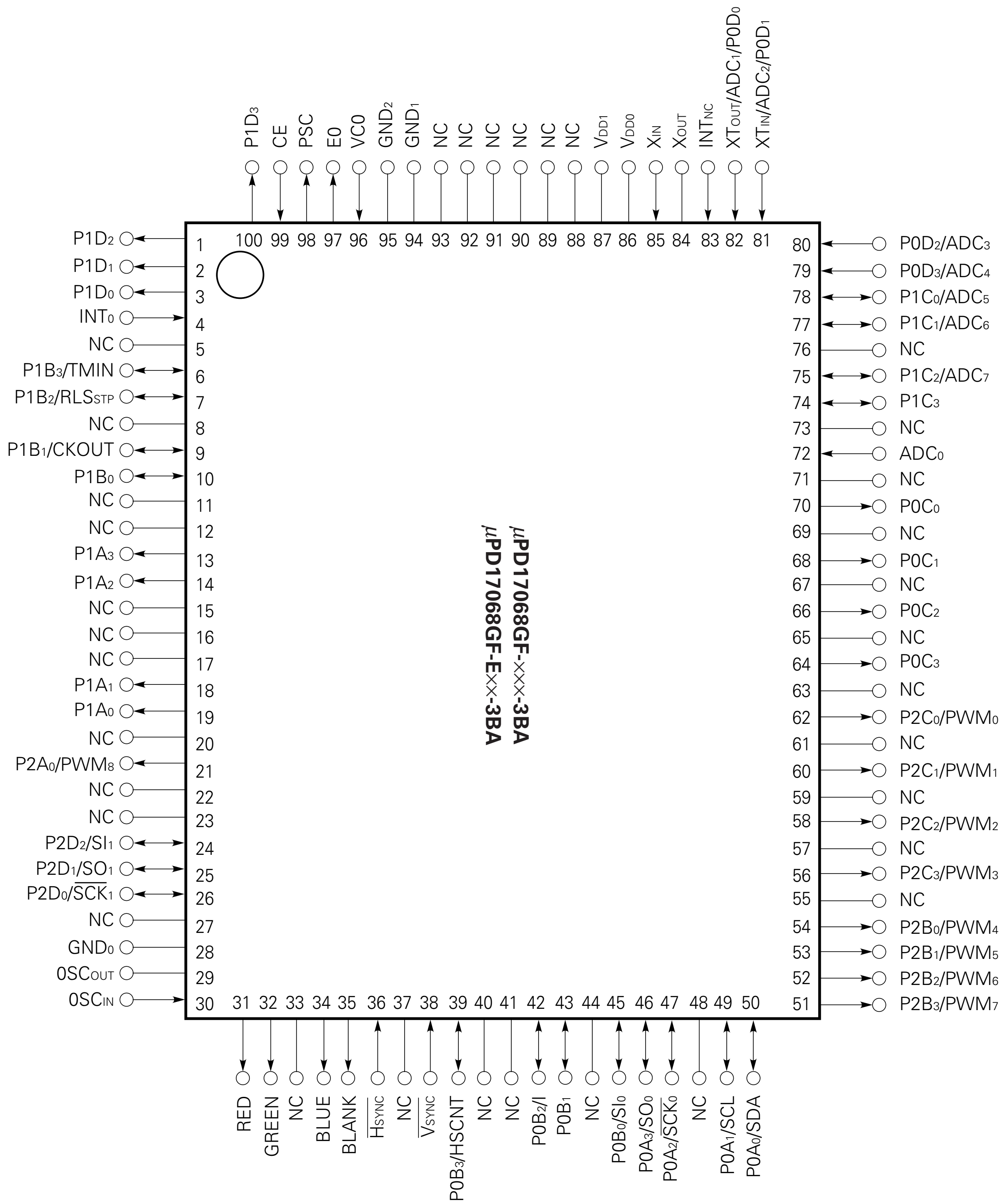
Carry : Detecting the state of the carry flip-flop

Count value : Reading the count value

BLOCK DIAGRAM



PIN CONFIGURATION (TOP VIEW)



PINS

ADC_0 – ADC_7	: A/D converter input
BLANK	: Blanking signal output
BLUE	: Character signal output
CE	: Chip enable
CKOUT	: Watch timer adjustment output
EO	: Error output
GND_0, GND_1, GND_2	: Ground
GREEN	: Character signal output
HSCNT	: Input for horizontal synchronizing signal counter
$\overline{H_{SYNC}}$	: Horizontal synchronizing signal input
I	: Character signal output
INT_0, INT_{NC}	: Input for external interrupt request signal
OSC_{IN}, OSC_{OUT}	: LC oscillation I/O for IDC
$P0A_0$ - $P0A_3$	: Port 0A
$P0B_0$ - $P0B_3$	: Port 0B
$P0C_0$ - $P0C_3$	: Port 0C
$P0D_0$ - $P0D_3$	: Port 0D
$P1A_0$ - $P1A_3$	: Port 1A
$P1B_0$ - $P1B_3$	: Port 1B
$P1C_0$ - $P1C_3$	: Port 1C
$P1D_0$ - $P1D_3$	: Port 1D
$P2A_0$	: Port 2A
$P2B_0$ - $P2B_3$	: Port 2B
$P2C_0$ - $P2C_3$	: Port 2C
$P2D_0$ - $P2D_2$	: Port 2D
PSC	: Pulse swallow control output
PWM_0 - PWM_8	: Pulse width modulation output
RED	: Character signal output
RLS_{STP}	: Input for clock stop release signal
SCL	: Shift clock I/O
$\overline{SCK_0}$, $\overline{SCK_1}$	: Shift clock I/O
SDA	: Serial data I/O
SI_0, SI_1	: Serial data input
SO_0, SO_1	: Serial data output
TMIN	: Event input for basic timer 1 or 2
VCO	: Local oscillation input
V_{DD0}, V_{DD1}	: Main power supply
$\overline{V_{SYNC}}$	: Vertical synchronizing signal input
X_{IN}, X_{OUT}	: Main clock oscillation I/O
XT_{IN}, XT_{OUT}	: Watch timer oscillation I/O

CONTENTS

1. PIN FUNCTIONS

1.1 LIST OF PIN FUNCTIONS

(1) Port pins

Pin	Function	I/O	Output type	At reset	Also used as:
P0A$_0$	4-bit I/O port. Input or output mode can be specified in bit units. The pins are automatically set to input mode when the power (V_{DD}) is turned on, the clock is stopped, or the device is reset with the CE pin.	I/O	N-ch open drain	Input	SDA
P0A$_1$					SCL
P0A$_2$			CMOS push-pull		$\overline{SCK_0}$
P0A$_3$					SO$_0$
P0B$_0$	4-bit I/O port. Input or output mode can be specified in bit units. The pins are automatically set to input mode when the power (V_{DD}) is turned on, the clock is stopped, or the device is reset with the CE pin.	I/O	CMOS push-pull	Input	SI$_0$
P0B$_1$					–
P0B$_2$					I
P0B$_3$					HSCNT
P0C$_0$ – P0C$_3$	4-bit output port. Undefined data is output when the power (V_{DD}) is turned on.	Output	CMOS push-pull	Outputs undefined data.	–
P0D$_0$	4-bit input port	Input	–	Input with a pull-down registor	ADC$_1$/XT$_{OUT}$
P0D$_1$					ADC$_2$/XT$_{IN}$
P0D$_2$					ADC$_3$
P0D$_3$					ADC$_4$
P1A$_0$ – P1A$_3$	4-bit output port	Output	N-ch open drain with intermediate with-stand voltage and high current	Outputs undefined data.	–
P1B$_0$	4-bit I/O port. Input or output mode can be specified in bit units.	I/O	CMOS push-pull	Input	–
P1B$_1$					CKOUT
P1B$_2$					RLS$_{STP}$
P1B$_3$					TMIN
P1C$_0$ – P1C$_2$	4-bit I/O port. Input or output mode can be specified in 4-bit units.	I/O	CMOS push-pull	Input	ADC$_5$ – ADC$_7$
P1C$_3$					–
P1D$_0$ – P1D$_3$	4-bit output port	Output	CMOS push-pull	Outputs undefined data.	–
P2A$_0$	1-bit output port	Output	N-ch open drain with intermediate with-stand voltage	Outputs undefined data.	PWM$_3$
P2B$_0$ – P2B$_3$	4-bit output port	Output	N-ch open drain with intermediate withstand voltage	Outputs undefined data.	PWM$_4$ – PWM$_7$
P2C$_0$ – P2C$_3$	4-bit output port	Output	N-ch open drain with intermediate withstand voltage	Outputs undefined data.	PWM$_0$ – PWM$_3$
P2D$_0$	3-bit I/O port. Input or output mode can be specified in bit units. The pins are automatically set to input mode when the power (VDD) is turned on, the clock is stopped, or the device is reset with the CE pin.	I/O	CMOS push-pull	Input	$\overline{SCK_1}$
P2D$_1$					SO$_1$
P2D$_2$					SI$_1$

(2) Non-port pins

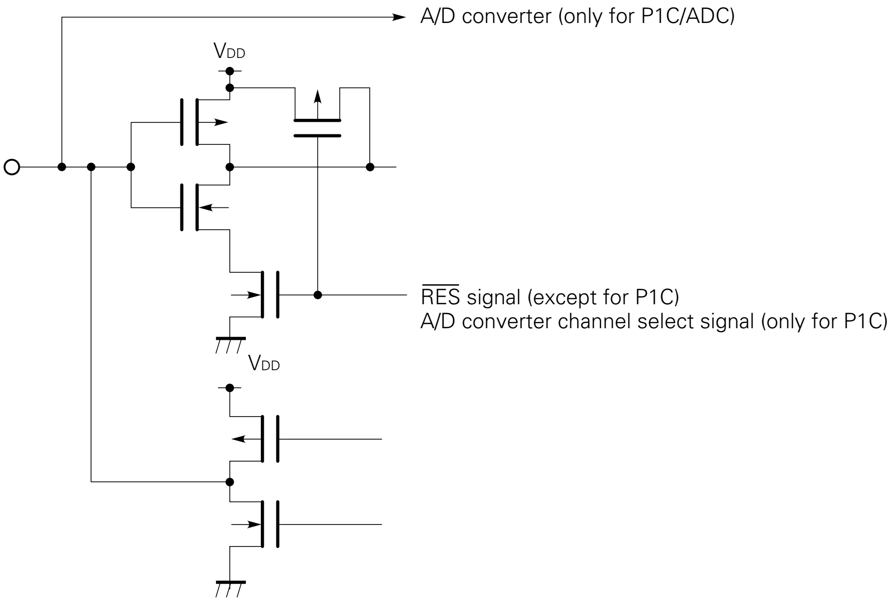
Pin	Function	I/O	Output type	At reset	Also used as:
INT$_0$	Input pin for an external interrupt request signal. An interrupt request is triggered by the rising or falling edge of the signal input to this pin.	Input	–	Input	–
INT$_{NC}$	Input pin for an interrupt request signal with a noise canceler. When inputting a signal subject to much noise, such as a remote-controller signal, use this pin to facilitate programming. Whether the rising or falling edge of the input signal is used to trigger an interrupt request can be specified with a program.	Input	–	Input	–
TMIN	Event input pin for basic timer 1 or 2	Input	–	Input	PIB$_3$
XT$_{IN}$	Pins for connecting the crystal (32.768 kHz) for the watch timer	–	–	–	P0D$_1$/ADC$_2$
XT$_{OUT}$					P0D$_0$/ADC$_1$
CKOUT	Output pin for adjusting the watch timer	Output	CMOS push-pull	Input	P1B$_1$
$\overline{SCK_0}$	Shift clock I/O pins	I/O	CMOS push-pull	Input	P0A$_2$
$\overline{SCK_1}$					P2D$_0$
SI$_0$	Serial data input pins	Input	–	Input	P0B$_0$
SI$_1$					P2D$_2$
SO$_0$	Serial data output pins	Output	CMOS push-pull	Input	P0A$_3$
SO$_1$					P2D$_1$
SCL	Shift clock I/O pin	I/O	N-ch open drain	Input	P0A$_1$
SDA	Serial data I/O pin	I/O	N-ch open drain	Input	P0A$_0$
ADC$_0$	Analog input pins for the A/D converter with 6-bit resolution	Input	–	Input	–
ADC$_1$					P0D$_0$/XT$_{OUT}$
ADC$_2$					P0D$_1$/XT$_{IN}$
ADC$_3$					P0D$_2$
ADC$_4$					P0D$_3$
ADC$_5$ | ADC$_7$	Analog input pins for the A/D converter with 6-bit resolution	Input	–	Input	P1C$_0$ | P1C$_2$
PWM$_0$ | PWM$_3$	Output pins for the D/A converter with 8-bit resolution	Output	N-ch open drain with intermediate withstand voltage	Low-level output or high imped-ance	P2C$_0$ | P2C$_3$
PWM$_4$ | PWM$_7$					P2B$_0$ | P2B$_3$
PWM$_8$					P2A$_0$

Pin	Function	I/O	Output type	At reset	Also used as:
EO	Output pin for the charge pump of the PLL frequency synthesizer. When the divided local oscillation (VCO) frequency is higher than the reference frequency, the output of this pin goes to high level. When the divided frequency is lower than the reference frequency, the output of this pin goes to low level. When the frequencies are the same, this pin enters floating status.	Output	CMOS tristate	High impedance	–
PSC	Output pin for pulse swallow control. This pin is used to output a signal to change the frequency division ratio to the μPB595 dedicated prescaler.	Output	CMOS push-pull	Output	–
VCO	Local oscillation input pin. The local oscillation (VCO) output from the tuner is frequency-divided by the μPB595 dedicated prescaler and input to this pin (the μPB595 is a two-modulus prescaler for up to 1 GHz).	Input	–	Internally pulled-down	–
HSCNT	Input pin for horizontal synchronizing signal counter.	Input	–	Input	$P0B_3$
BLANK	Output pin for the blanking signal for cutting the video signal. The signal is high active.	Output	CMOS push-pull	Low-level output	–
RED	Output pin for the R signal for character data received from the IDC. The signal is high active.	Output	CMOS push-pull	Low-level output	–
GREEN	Output pin for the G signal for character data from the IDC. The signal is high active.	Output	CMOS push-pull	Low-level output	–
BLUE	Output pin for the B signal for character data from the IDC. The signal is high active.	Output	CMOS push-pull	Low-level output	–
I	Output pin for the I signal for character data from the IDC.	Output	CMOS push-pull	Input	$P0B_2$
$\overline{H_{SYNC}}$	Input pin for the horizontal synchronizing signal for the IDC. Used to input the active-low horizontal synchronizing signal.	Input	–	Input	–
$\overline{V_{SYNC}}$	Input pin for the vertical synchronizing signal for the IDC. Used to input the active-low vertical synchronizing signal.	Input	–	Input	–
OSC_{IN} OSC_{OUT}	Pins for connecting the LC oscillation circuit for the IDC. Used to connect a 10 MHz LC oscillation circuit.	–	–	–	–

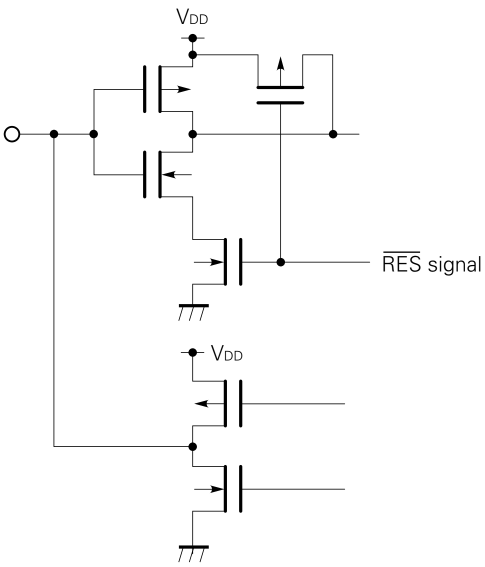
Pin	Function	I/O	Output type	At reset	Also used as:
CE	Input pin for the device operation selection signal and reset signal. **(1) Device operation selection signal** When the device operation selection signal at the CE pin is high, the PLL frequency synthesizer and IDC are enabled. When the signal is low, the PLL frequency synthesizer and IDC are disabled. **(2) Reset signal** When the reset signal at the CE pin is changed from low to high, the device is reset in synchronization with the internal carry flip-flop for basic interval timer 0.	Input	–	Input	–
RLS_{STP}	Input pin for the clock stop release signal	Input	–	Input	$P1B_2$
X_{IN} X_{OUT}	Pins for connecting a crystal (8 kHz) for the main clock	–	–	–	–
V_{DD0} V_{DD1}	Main power supply pins. Supply 5 V±10% when operating the entire device. Supply 4.0 to 5.5 V when the IDC is not being used. The minimum supply voltage is 2.5 V in the clock-stop state. A power-on reset circuit is provided. When the supply voltage increases from 0 to 4.0 V, the system is reset and the program restarts at address 0. The voltage must increase from 0 to 4.0 V within 500 ms to enable correct operation of the power-on reset circuit.	–	–	–	–
GND_0 | GND_2	Ground pins	–	–	–	–
NC	No connection	–	–	–	–

1.2 EQUIVALENT CIRCUIT OF EACH PIN

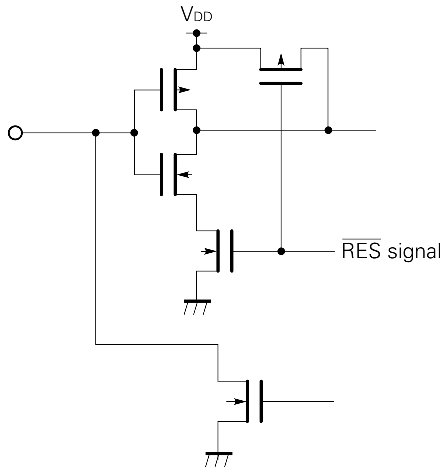
(1) P0A (P0A$_3$/SO$_0$, P0A$_2$/$\overline{SCK_0}$)
P0B (P0B$_2$/I, P0B$_1$, P0B$_0$/SI$_0$)
P1B (P1B$_2$/RLS$_{STP}$, P1B$_1$/CKOUT, P1B$_0$)
P1C (P1C$_3$, P1C$_2$/ADC$_7$, P1C$_1$/ADC$_6$, P1C$_0$/ADC$_5$)
(I/O)



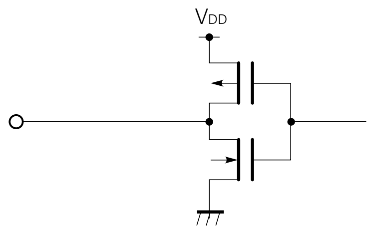
(2) P2D (P2D$_2$/SI$_1$, P2D$_1$/SO$_1$, P2D$_0$/$\overline{SCK_1}$) : (I/O)



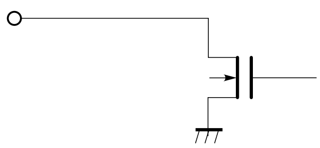
(3) P0A (P0A$_1$/SCL, P0A$_0$/SDA) : (I/O)



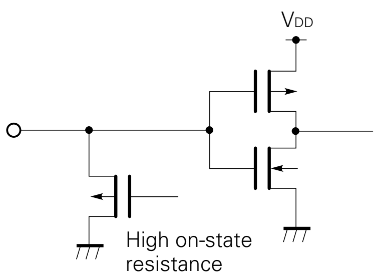
(4) P0C (P0C$_3$, P0C$_2$, P0C$_1$, P0C$_0$)
P1D (P1D$_3$, P1D$_2$, P1D$_1$, P1D$_0$)
RED, GREEN, BLUE, BLANK
PSC
(Output)



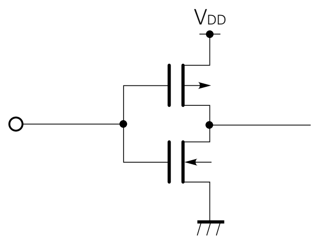
(5) P1A (P1A$_3$, P1A$_2$, P1A$_1$, P1A$_0$)
P2A (P2A$_0$/PWM$_8$)
P2B (P2B$_3$/PWM$_7$, P2B$_2$/PWM$_6$, P2B$_1$/PWM$_5$, P2B$_0$/PWM$_4$)
P2C (P2C$_3$/PWM$_3$, P2C$_2$/PWM$_2$, P2C$_1$/PWM$_1$, P2C$_0$/PWM$_0$)
(Output)



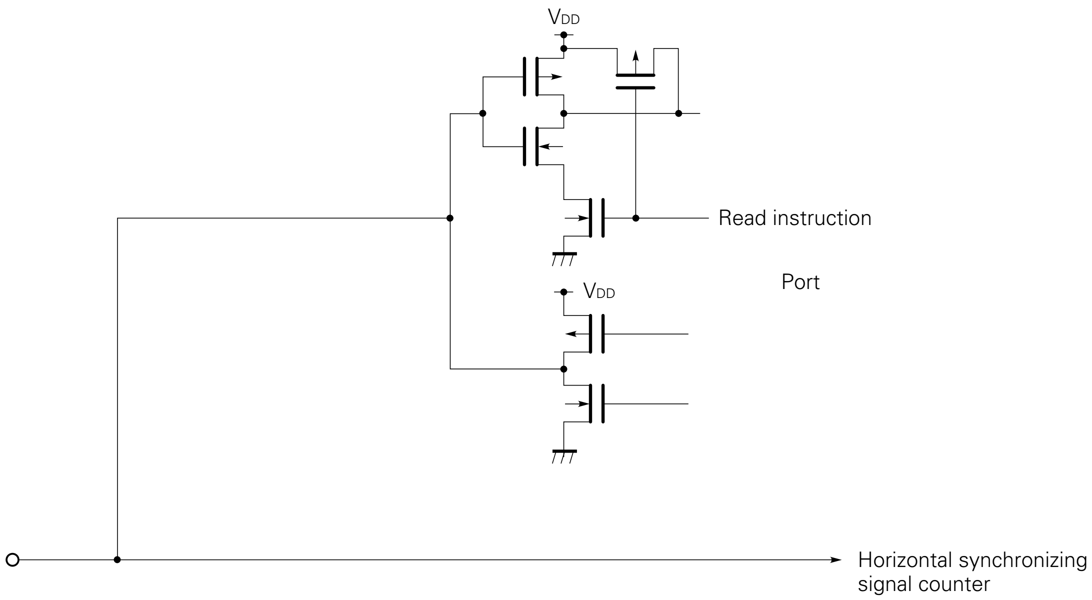
(6) P0D (P0D$_3$/ADC$_4$, P0D$_2$/ADC$_3$, P0D$_1$/ADC$_2$/XT$_{IN}$, P0D$_0$/ADC$_1$/XT$_{OUT}$) : (Input)



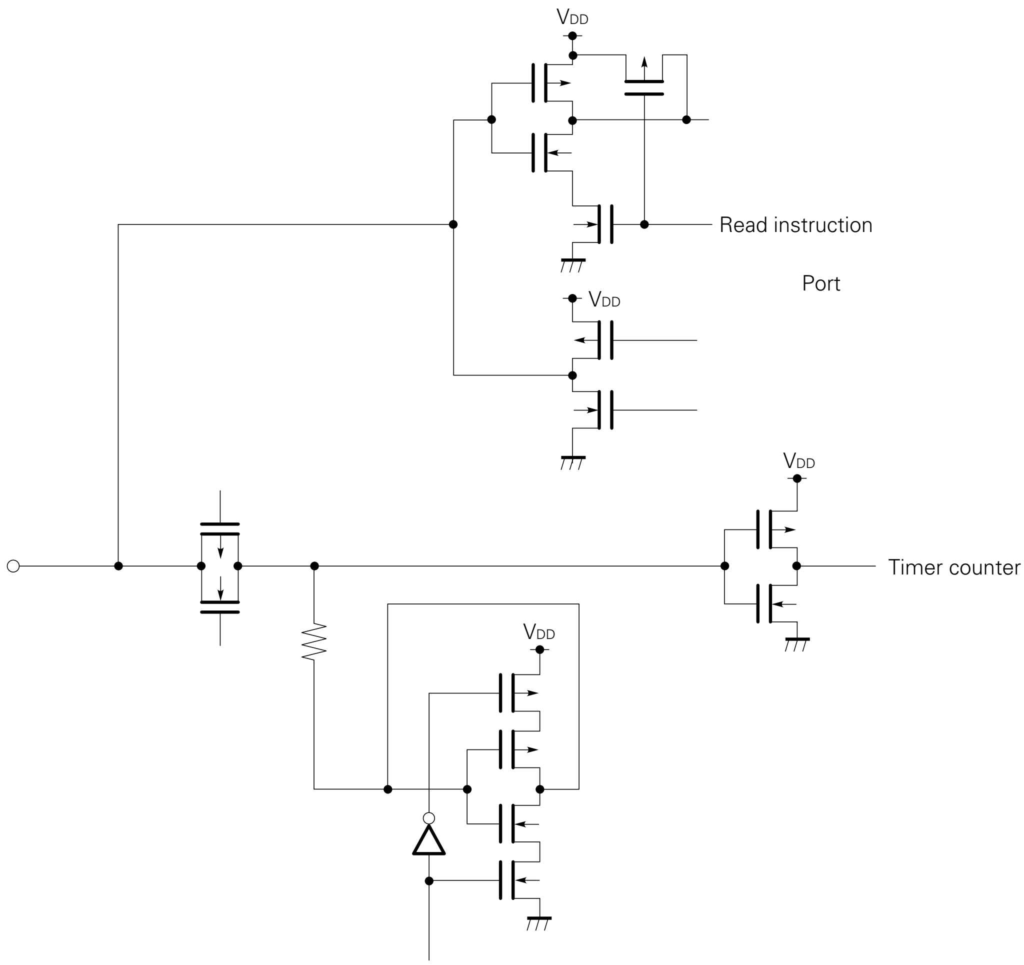
(7) ADC0 : (Input)



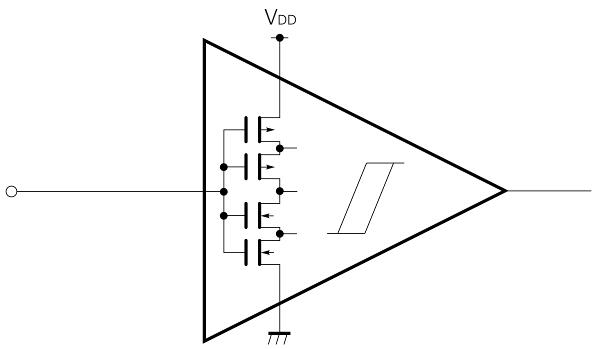
(8) P0B3/HSCNT : (I/O)



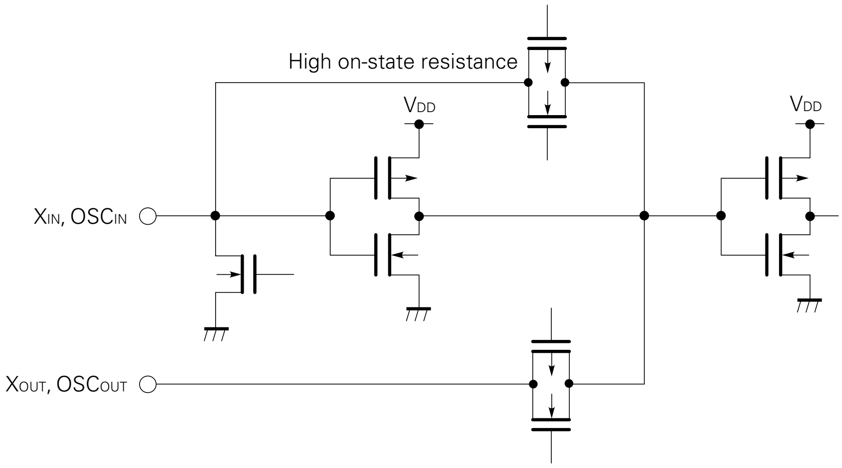
(9) P1B$_3$/TMIN : (I/O)



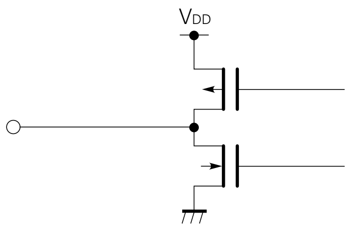
(10) $\overline{\text{H}_{\text{SYNC}}}$, $\overline{\text{V}_{\text{SYNC}}}$, CE, INT$_0$, INT$_{\text{NC}}$: (Schmitt-triggered input)



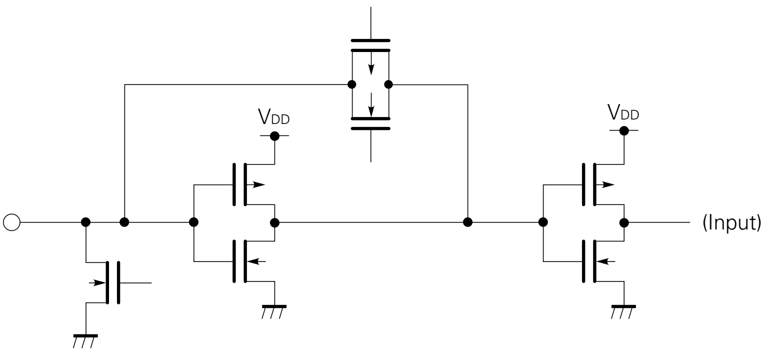
(11) X_{IN}, OSC_{IN} : (Input)
X_{OUT}, OSC_{OUT} :



(12) EO : (Output)



(13) VCO : (Input)



1.3 HANDLING UNUSED PINS

Connect unused pins as follows:

Table 1-1 Handling Unused Pins (1/2)

Pin	I/O	Recommended connection when not used
P0A$_0$/SDA	I/O	Input: Connect to V$_{DD}$ or V$_{SS}$. Output: Output a low and leave open.
P0A$_1$/SCL		
P0A$_2$/$\overline{\text{SCK}_0}$	I/O	Input: Connect to V$_{DD}$ or V$_{SS}$. Output: Leave open.
P0A$_3$/SO$_0$		
P0B$_0$/SI$_0$		
P0B$_1$		
P0B$_2$/I		
P0B$_3$/HSCNT		
P0C$_0$ – P0C$_3$	Output	Leave open.
P0D$_0$/ADC$_1$/XT$_{OUT}$	Input with a pull-down resistor.	Leave open or connect to V$_{SS}$.
P0D$_1$/ADC$_2$/XT$_{IN}$		
P0D$_2$/ADC$_3$		
P0D$_3$/ADC$_4$		
P1A$_0$ – P1A$_3$	N-ch open-drain output	Output a low and leave open.
P1B$_0$	I/O	Input: Connect to V$_{DD}$ or V$_{SS}$. Output: Leave open.
P1B$_1$/CKOUT		
P1B$_2$/RLS$_{STP}$		
P1B$_3$/TMIN		
P1C$_0$/ADC$_5$ – P1C$_2$/ADC$_7$		
P1C$_3$		
P1D$_0$ – P1D$_3$	Output	Leave open.
P2A$_0$/PWM$_8$	N-ch open-drain output	Output a low and leave open.
P2B$_0$/PWM$_4$ – P2B$_3$/PWM$_7$		
P2C$_0$/PWM$_0$ – P2C$_3$/PWM$_3$		
P2D$_0$/$\overline{\text{SCK}_1}$	I/O	Input: Connect to V$_{DD}$ or V$_{SS}$. Output: Leave open.
P2D$_1$/SO$_1$		
P2D$_2$/SI$_1$		

Table 1-1 Handling Unused Pins (2/2)

Pin	I/O	Recommended connection when not used
EO	Output	Leave open.
PSC		
VCO	Input with a pull-down resistor	Leave open or connect to V_{SS}.
BLANK	Output	Leave open.
RED		
GREEN		
BLUE		
$\overline{H_{SYNC}}$	Input	Connect to V_{DD} or V_{SS}.
$\overline{V_{SYNC}}$		
OSC_{IN}	Input with a pull-down resistor	Leave open or connect to V_{SS}.
OSC_{OUT}	Output	Leave open.
ADC_0	Input	Connect to V_{DD} or V_{SS}.
INT_0		
INT_{NC}		

1.4 NOTES ON USE OF THE CE AND INT_{NC} PINS

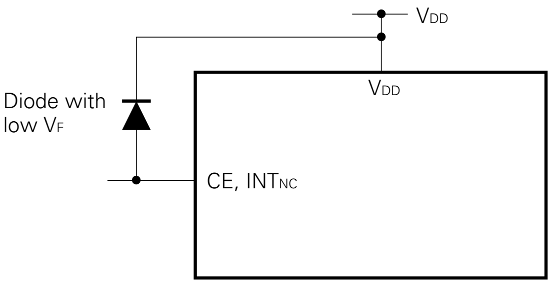
The CE and INT_{NC} pins support a test mode for selecting the function for testing the internal operation of the μPD17068 (IC test), in addition to the functions described in **Section 1.1**.

Applying a voltage exceeding V_{DD} to the CE or INT_{NC} pin causes the μPD17068 to enter test mode. If noise exceeding V_{DD} is encountered during normal operation, the device will be switched to test mode.

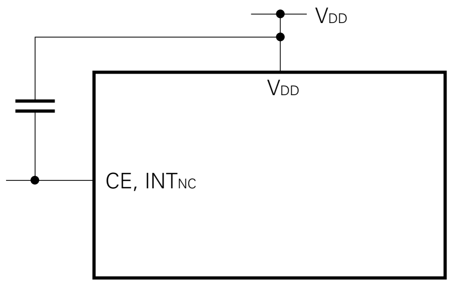
For example, if the wiring from the CE or INT_{NC} pin is too long, noise may be induced in the wiring, thus resulting in this mode switching.

When installing wiring, route the wiring such that noise is suppressed as much as possible. If, however, noise arises, use an external part to suppress it as shown below.

○ **Connect a diode with low V_F between the pin and V_{DD}.**



○ **Connect a capacitor between the pin and V_{DD}.**



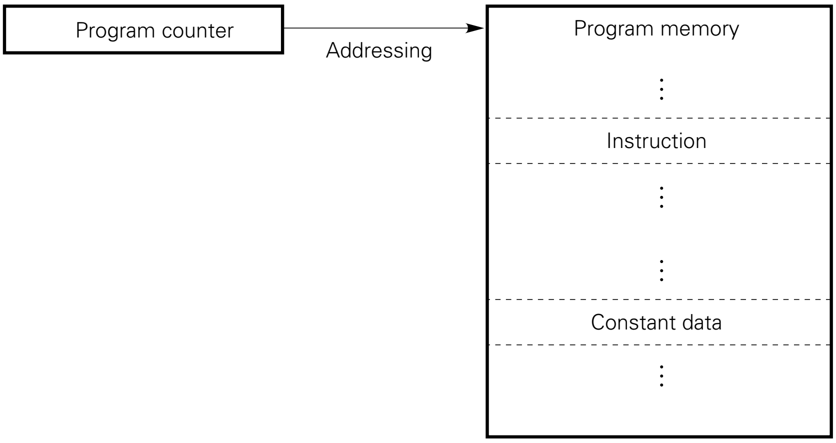
2. PROGRAM MEMORY (ROM)

2.1 OUTLINE OF PROGRAM MEMORY

Fig. 2-1 outlines program memory. As shown, program memory is addressed with a program counter. Program memory has the following functions:

(1) Storing programs
(2) Storing constant data

Fig. 2-1 Outline of Program Memory



2.2 PROGRAM MEMORY CONFIGURATION

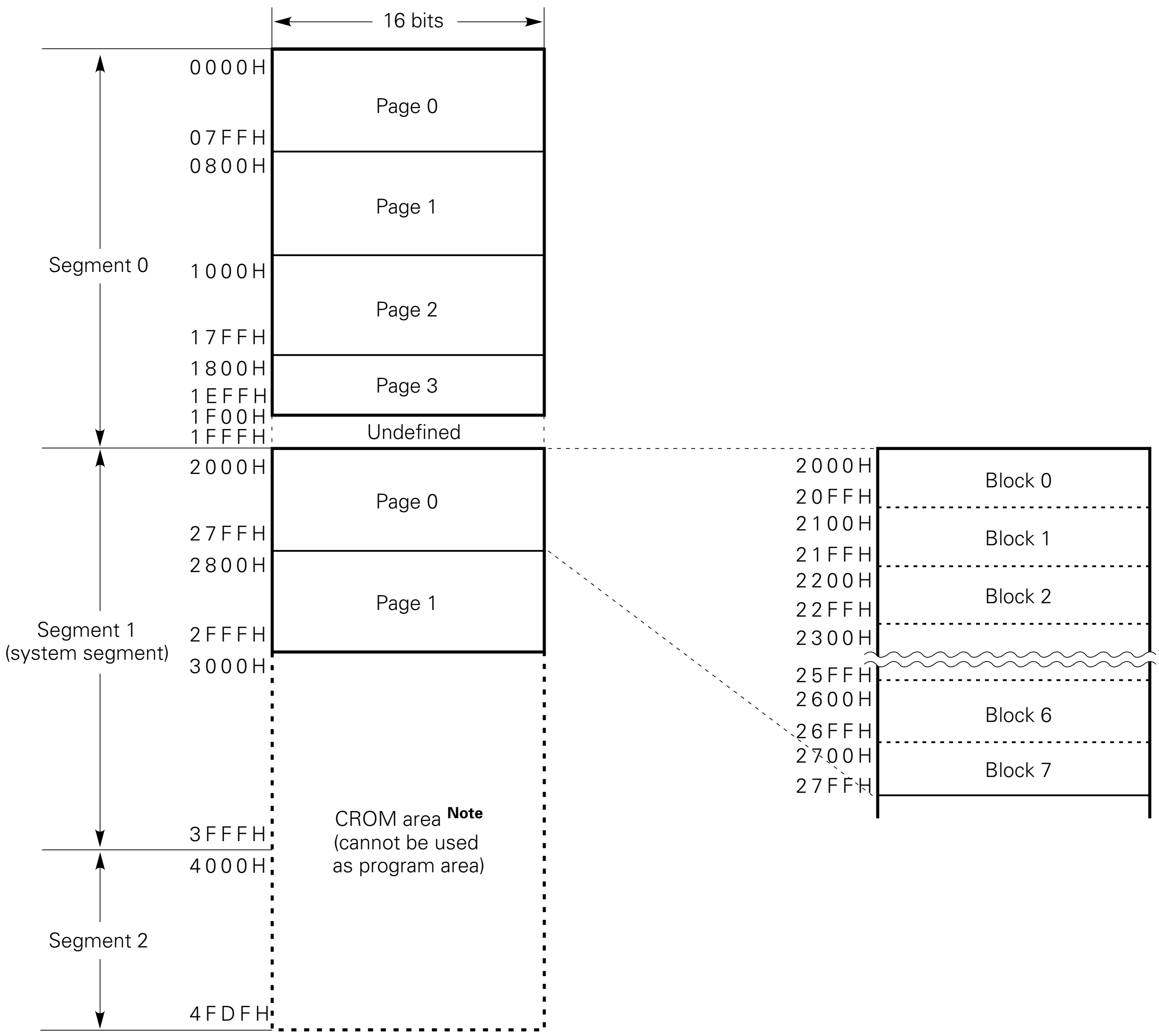
Fig. 2-2 shows the configuration of program memory. As shown, program memory consists of 24K bytes (12032 × 16 bits). Program memory therefore has addresses 0000H to 2FFFH.

Program memory addresses 3000H to 4FDFH are assigned to the CROM (character ROM) area. This area cannot be used as an ordinary program area.

All μPD17068 instructions are 16-bit one-word instructions. Each instruction can be stored at a single address of program memory.

Constant data stored in program memory is read into the data buffer by executing a table reference instruction.

Fig. 2-2 Program Memory Configuration



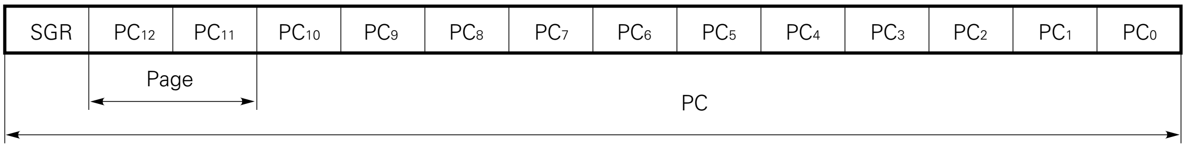
Note Addresses in the CROM area are specified with VRAM. Addresses that can be specified with VRAM are 3000H to 3FEFH. An address in the CROM area, however, has 32 bits because CROM data is represented in 24-bit units. Therefore, the CROM addresses which are actually used (actual addresses) are 3000H to 4FDFH (see **Chapter 16**).

2.3 PROGRAM COUNTER

2.3.1 Program Counter Configuration

Fig. 2-3 shows the configuration of the program counter. As shown, the program counter consists of a 13-bit binary counter and 1-bit segment register (SGR). Bits 11 and 12 indicate a page. The program counter is used to specify an address in program memory.

Fig. 2-3 Program Counter Configuration



2.3.2 Segment Register (SGR)

The segment register specifies a segment of program memory. Table 2-1 lists the correspondence between segment register values and program memory segments. The segment register is set when a SYSCAL entry instruction is executed.

Table 2-1 Correspondence between Segment Register Values and Program Memory Segments

Segment register value	Program memory segment
0	Segment 0
1	Segment 1

2.4 PROGRAM FLOW

The execution flow of a program is controlled with the program counter, which specifies an address in program memory. This section describes the operation of several types of instructions.

Fig. 2-4 shows the value set in the program counter when each instruction is executed. Table 2-2 lists the vector addresses when interrupts are received.

2.4.1 Branch Instructions

(1) Direct branch ("BR addr")

A direct branch instruction can branch only within the same segment of program memory.

(2) Indirect branch ("BR @AR")

An indirect branch instruction can branch to all addresses of program memory, 0000H to 2FFFH. See also **Section 5.3**.

2.4.2 Subroutines

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

A direct subroutine call instruction can call a subroutine starting at an address in page 0 in program memory.

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

An indirect subroutine call instruction can call a subroutine starting at any address in program memory, 0000H to 2FFFH. See also **Section 5.3**.

2.4.3 Table Reference

A table reference instruction (“MOVT DBF, @AR”) can reference all addresses in program memory, 0000H to 2FFFH. See also **Sections 5.3** and **9.2.2**.

2.4.4 System Call

A system call instruction (SYSCAL entry) can call a subroutine starting at any of the first 16 addresses of a block (0 to 7) in page 0 of segment 1.

Fig. 2-4 Program Counter Value for Each Instruction

Instruction \ Program Counter		Value of program counter (PC)													
		SGR	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	Hold	0	0	Instruction operand (addr)										
	Page 1		0	1											
	Page 2		1	0											
	Page 3		1	1											
CALL addr		Hold	0	0	Instruction operand (addr)										
SYSCAL entry		1	0	0	entry$_H$			0	0	0	0	entry$_L$			
BR @AR CALL @AR MOVT DBF, @AR		Contents of address register													
RET RETSK RETI		Return address: Contents of address stack register (ASR) specified with stack pointer (SP)													
When an interrupt is received		0	Vector address for the interrupt												
At power-on reset or CE reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0

Remark entry$_H$: Three high-order bits of entry
entry$_L$: Four low-order bits of entry

Table 2-2 Interrupt Vector Addresses

Priority	Internal/external	Interrupt source	Vector address
1	External	INT_{NC} pin	000AH
2	External	INT_0 pin	0009H
3	Internal	Timer 0	0008H
4	Internal	Timer 1	0007H
5	Internal	Basic timer 2	0006H
6	Internal	VRAM pointer	0005H
7	External	Interrupt group 1[Note 1]	0004H
8	Internal	Serial interface 0	0003H
9	Internal	Serial interface 1	0002H
10	Internal	Interrupt group 0[Note 2]	0001H

Notes 1. Interrupt group 1 : $\overline{V_{SYNC}}$ or $\overline{H_{SYNC}}$ pin
2. Interrupt group 0 : Timer 0 overflow

2.5 NOTES ON USE OF PROGRAM MEMORY

2.5.1 Program Counter and Program Memory Size

The program counter can specify addresses 0000H to 3FFFH, while the valid program memory addresses are 0000H to 1EFFH and 2000H to 2FFFH.

Therefore, do not use an instruction specifying addresses 1F00H to 1FFFH or 3000H to 3FFFH.

Program memory addresses 1F00H to 1FFFH contain undefined values. Addresses 3000H to 3FFFH constitute the CROM area, which cannot be specified with the program counter.

2.5.2 Last Address of Each Segment

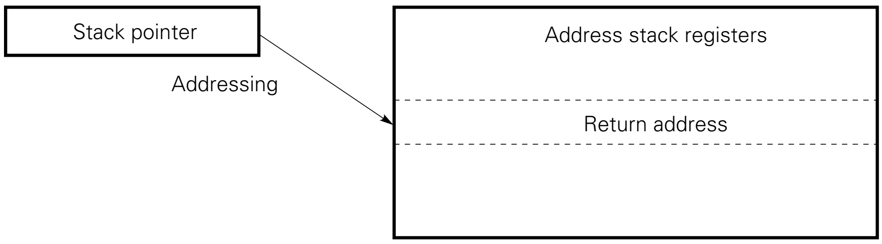
The segment register is not connected the binary counter. The last address of segment 0, 1FFFH, is followed by address 0000H of the same segment. Use instructions such as indirect branch, indirect subroutine call, and system call to specify another segment.

3. ADDRESS STACK (ASK)

3.1 OUTLINE OF ADDRESS STACK

Fig. 3-1 outlines the address stack. The address stack consists of the stack pointer and address stack registers. The stack pointer is used to specify one of the address stack registers. The address stack is used to store the return address when a subroutine call instruction is executed or an interrupt is received. The address stack is also used when a table reference instruction is executed.

Fig. 3-1 Outline of Address Stack



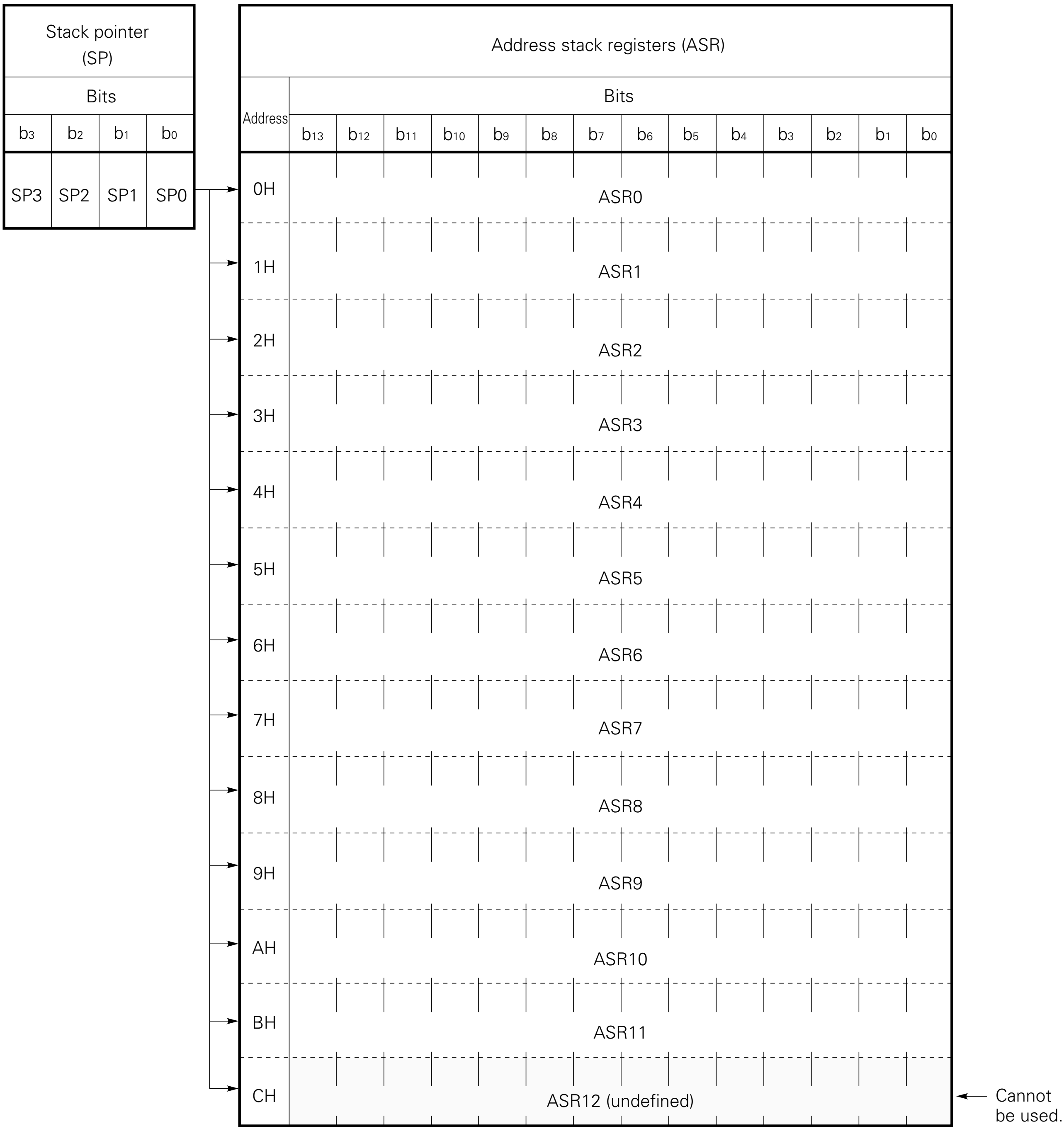
3.2 ADDRESS STACK REGISTERS (ASR)

Fig. 3-2 shows the configuration of the address stack registers. There are 13 address stack registers, ASR0 to ASR12, each consisting of 14 bits. ASR12, however, cannot be used, the twelve 14-bit registers (ASR0 to ASR11) actually being used.

The most significant bit of each address stack register is the segment register stack (SGRSK), the other 13 bits being used as the program counter stack (PCSK).

The address stack is used to store the return address when a subroutine call or table reference instruction is executed or an interrupt is received.

Fig. 3-2 Configuration of Address Stack Registers



3.3 STACK POINTER (SP)

3.3.1 Configuration and Function of Stack Pointer

Fig. 3-3 shows the configuration and function of the stack pointer. The stack pointer is a 4-bit binary counter, used for specifying an address stack register. The value of the stack pointer can be directly read or written with a register manipulation instruction.

Fig. 3-3 Configuration and Function of Stack Pointer

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

b_3	b_2	b_1	b_0	Specifies an address stack register (ASR).
0	0	0	0	ASR0
0	0	0	1	ASR1
0	0	1	0	ASR2
0	0	1	1	ASR3
0	1	0	0	ASR4
0	1	0	1	ASR5
0	1	1	0	ASR6
0	1	1	1	ASR7
1	0	0	0	ASR8
1	0	0	1	ASR9
1	0	1	0	ASR10
1	0	1	1	ASR11
1	1	0	0	ASR12

Upon reset	Power-on	1	1	0	0
	Clock stop	1	1	0	0
	CE	1	1	0	0

3.4 ADDRESS STACK OPERATION

3.4.1 Subroutine Call Instruction (“CALL addr” or “CALL @AR”) and Return Instruction (“RET” or “RETSK”)

When a subroutine call instruction is executed, the value of the stack pointer is decremented by one, after which the return address is stored in the address stack register specified with the stack pointer.

When a return instruction is executed, the contents (return address) of the address stack register specified with the stack pointer are read back into the program counter, after which the value of the stack pointer is incremented by one.

3.4.2 Table Reference Instruction (“MOVT DBF, @AR”)

When a table reference instruction is executed, the value of the stack pointer is decremented by one, after which the return address is stored in the address stack register specified with the stack pointer.

Next, the contents of the program memory address specified with the address register are read into the data buffer. Finally, the contents (return address) of the address stack register specified with the stack pointer are read back into the program counter, after which the value of the stack pointer is incremented by one.

3.4.3 Interrupt Reception and Return Instruction (“RETI”)

When an interrupt is received, the value of the stack pointer is decremented by one, after which the return address is stored in the address stack register specified with the stack pointer.

When a return instruction is executed, the contents (return address) of the address stack register specified with the stack pointer are read back into the program counter, after which the value of the stack pointer is incremented by one.

3.4.4 Address Stack Manipulation Instructions (“PUSH AR”, “POP AR”)

When a PUSH instruction is executed, the value of the stack pointer is decremented by one, after which the contents of the address register are transferred to the address stack register specified with the stack pointer.

When a POP instruction is executed, the contents of the address stack register specified with the stack pointer are transferred to the address register, after which the value of the stack pointer is incremented by one.

3.4.5 System Call Instruction (“SYSCAL entry”) and Return Instruction (“RET” or “RETSK”)

When a “SYSCAL entry” instruction is executed, the value of the stack pointer is decremented by one, after which the return address and the value of the segment register are stored in the address stack register specified with the stack pointer.

When a return instruction is executed, the contents of the address stack register specified with the stack pointer are restored into the program counter and segment register, after which the value of the stack pointer is incremented by one.

3.5 NOTES ON USE OF ADDRESS STACK

3.5.1 Nesting Level

When the stack pointer contains 0CH, it specifies address stack register ASR12, whose value is undefined. If the user attempts to use subroutine calls or interrupts that exceed 12 levels, without stack manipulation, the program will resume from an undefined address. Therefore, do not attempt such an operation.

4. DATA MEMORY (RAM)

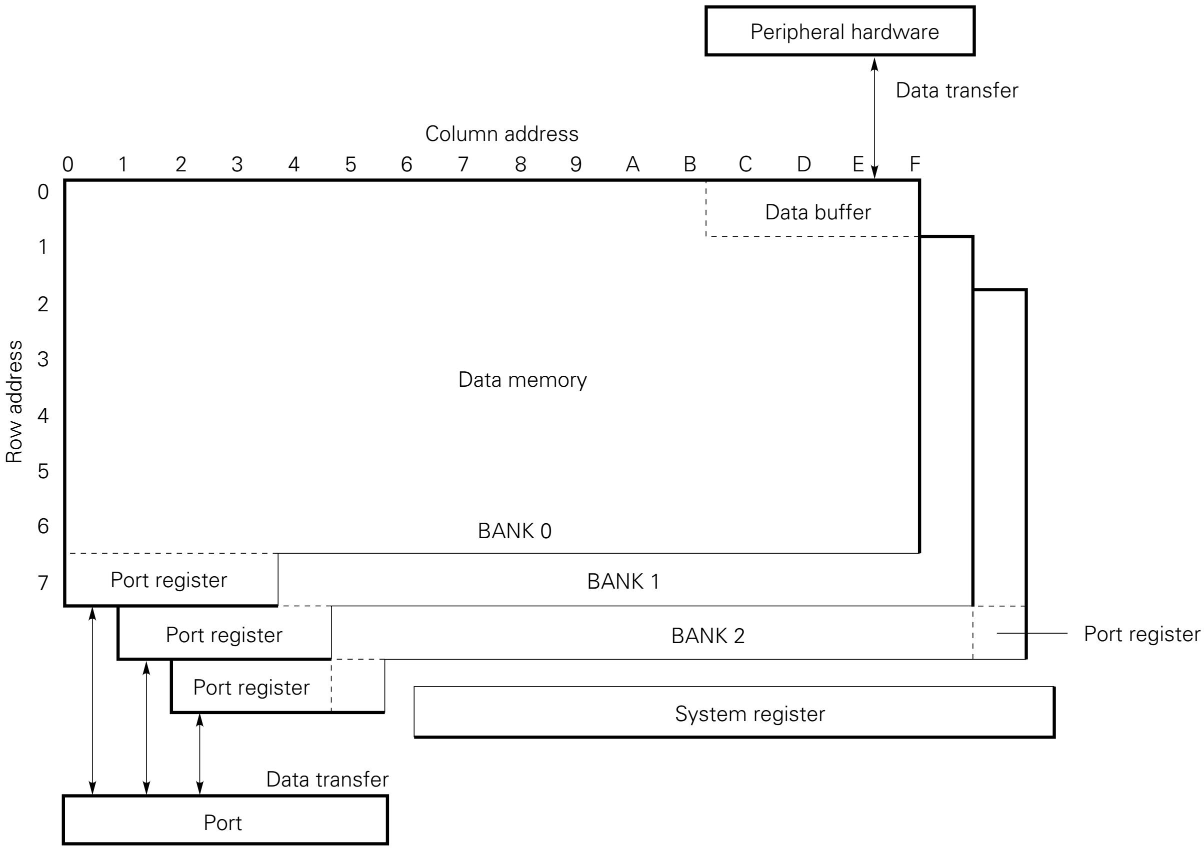
4.1 OUTLINE OF DATA MEMORY

Fig. 4-1 outlines the data memory.

As shown in Fig. 4-1, the data memory consists of a general-purpose data memory, system registers, data buffer, and port registers.

The data memory is used to store data, transfer data to and from peripheral hardware, set display data, transfer data to and from ports, and control the CPU.

Fig. 4-1 Outline of Data Memory



4.2 CONFIGURATION AND FUNCTIONS OF DATA MEMORY

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

As shown in Fig. 4-2, the data memory is divided into banks. Each bank consists of 128 nibbles made up of row addresses 0H to 7H by column addresses 0H to FH.

The data memory is divided into the functional blocks described in **Sections 4.2.1** through **4.2.6**.

By using data memory manipulation instructions, 4-bit operations, comparison, decision, and transfer operation can be performed for the data memory.

Table 4-1 indicates the data memory manipulation instructions.

4.2.1 System Register (SYSREG)

A system register is allocated at addresses 74H-7FH.

A system register is allocated, independently of the banks; each bank contains the same system register at addresses 74H-7FH.

See **Chapter 5** for details.

4.2.2 Data Buffer (DBF)

A data buffer is allocated at addresses 0CH-0FH of BANK0.

See **Chapter 9** for details.

4.2.3 VRAM (Video RAM) for the IDC

Addresses 00H-3FH of BANK2 of the data memory can also be used as a VRAM for the IDC.

Fig. 4-3 shows the configuration of the VRAM. The VRAM consists of VRAMBANK0-VRAMBANKD, that is, 672×16 bits. A VRAMBANK can be specified using the VRAM select register at addresses 73H of BANK2.

This area is used when the VRAMSEL flag (RF: address 33H, bit 3) is set to 1. When this area is not used as the VRAM, this area can be used as an ordinary RAM.

See **Chapter 16** for details.

4.2.4 Port Register

A port register is allocated at addresses 70H-73H of BANK0 and BANK1, and at addresses 6FH and 70H-72H of BANK2.

See **Chapter 10** for details.

4.2.5 General-Purpose Data memory

The general-purpose data memory consists of the data memory other than the system registers and port registers.

The general-purpose data memory is made up of 335 nibbles; 112 nibbles of each of BANK0 and BANK1, and 111 nibbles of BANK2.

4.2.6 Unmounted Data Memory

The data memory at addresses 30H-3FH of BANK2 and some portion of the port registers are not allocated for any purpose.

For details of the unmounted data memory area, see **Section 4.4.2** and **Chapter 10**.

Fig. 4-2 Configuration of Data Memory

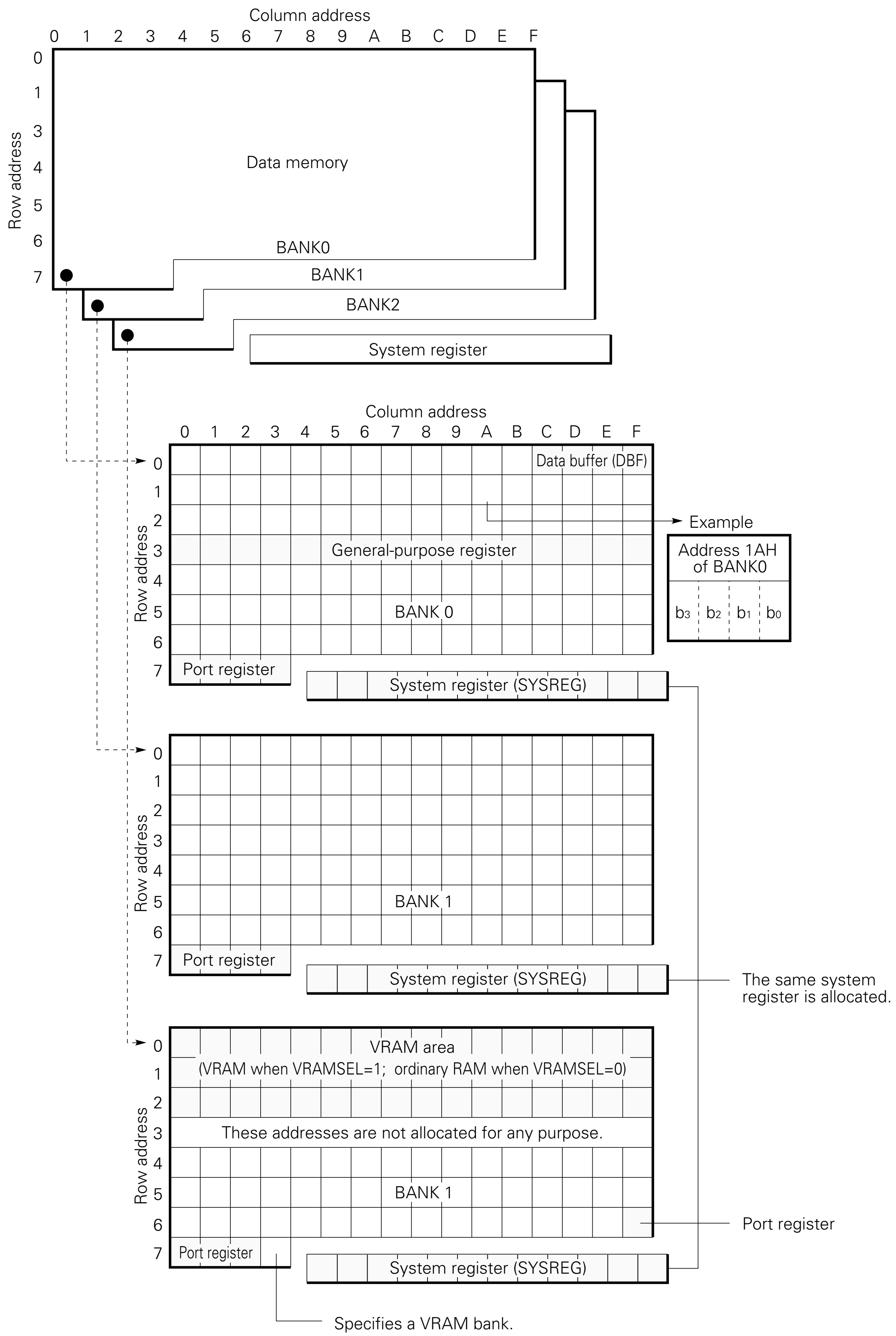


Fig. 4-3 Configuration of VRAM

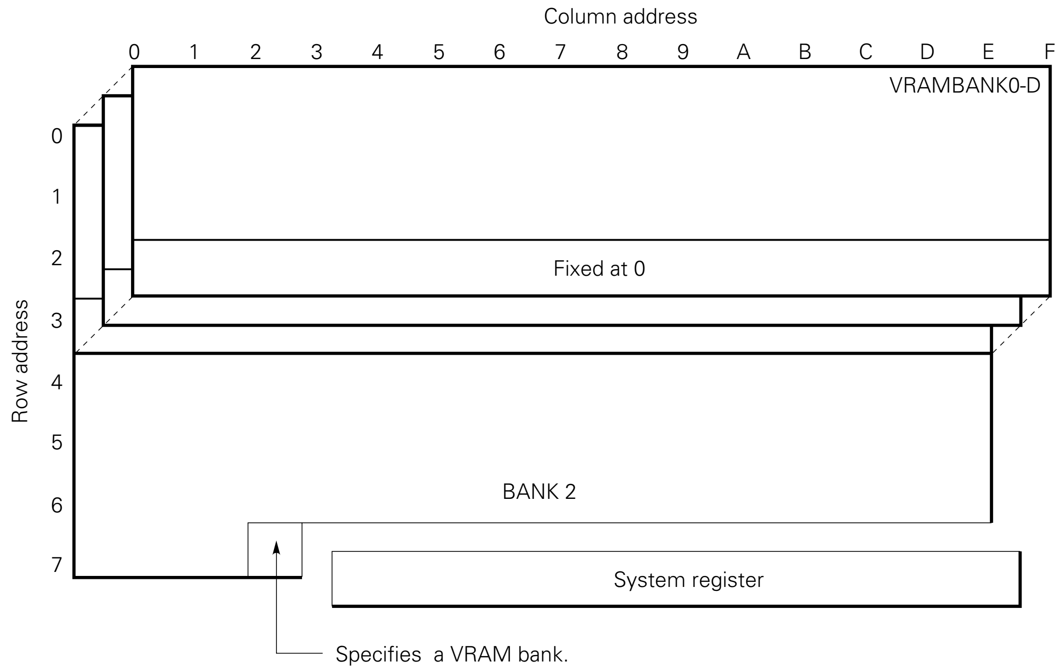


Table 4-1 List of Data Memory Manipulation Instructions

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

4.3 DATA MEMORY ADDRESSING

Fig. 4-4 shows how a data memory address is specified.

A data memory address is specified with a bank, row address, and column address.

A row address and column address are directly specified with a data memory manipulation instruction. A bank is specified with a bank register.

See **Chapter 5** for details of a bank register.

Fig. 4-4 Data Memory Addressing

	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
Data memory address	Bank register				Instruction operand						

4.4 NOTES ON USING DATA MEMORY

4.4.1 Power-On Reset

Upon power-on reset, the contents of the general-purpose data memory are undefined.

Initialize the general-purpose data memory as required.

4.4.2 Notes on Unmounted Data Memory

If a data memory manipulation instruction is executed for an address in the unmounted data memory, the operations below are performed.

(1) Device operation

When a read instruction is executed, 0 is read.

When a write instruction is executed, no change is made.

(2) Assembler (AS17K) operation

Normal assembly operation is performed.

No error occurs.

(3) Emulator (IE-17K) operation

When a read instruction is executed, 0 is read.

When a write instruction is executed, no change is made.

No error occurs.

5. SYSTEM REGISTER (SYSREG)

5.1 OUTLINE OF SYSTEM REGISTER

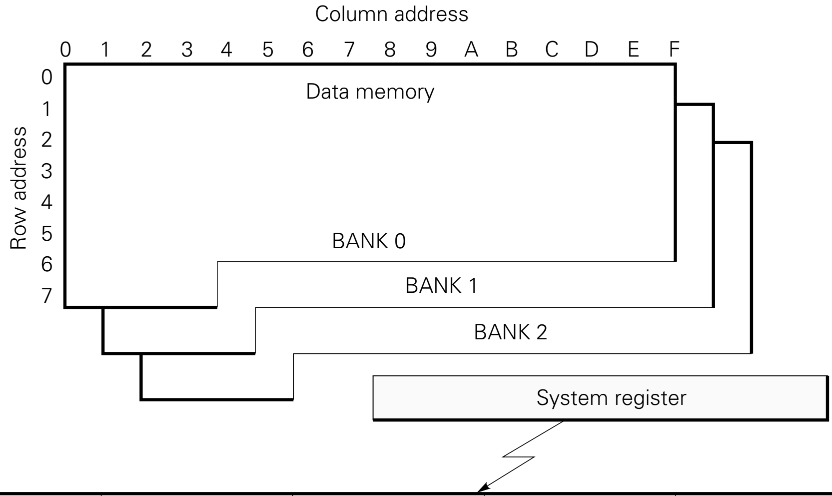
Fig. 5-1 shows where the system registers are located in the data memory, and also outlines the system register.

As shown in Fig. 5-1, a system register is allocated, independently of the banks; each bank contains the same system register at data memory addresses 74H-7FH.

The system registers are allocated in the data memory, so that the system registers can be manipulated using any manipulation instructions.

A system register consists of seven types of registers for different functions.

Fig. 5-1 Location on Data Memory and Outline of System Registers



Address	74H	75H	76H	77H	78H	79H
Register	System register (AR)				Window register (WR)	Bank register (BANK)
Outline	Program memory address control				Data transfer to and from register files	Data memory bank specification

Address	7AH	7BH	7CH	7DH	7EH	7FH
Register	Index register (IX)			General-purpose register pointer (RP)		Program status word (PSWORD)
	Data memory row address pointer (MP)					
Outline	Data memory address modification			General-purpose register addressing		Operation control

5.2 FORMAT OF SYSTEM REGISTER

Fig. 5-2 shows the format of the system register.

Fig. 5-2 Format of System Register

<table>
<tr><td>Address</td><td colspan="4">74H</td><td colspan="4">75H</td><td colspan="4">76H</td><td colspan="4">77H</td><td colspan="4">78H</td><td colspan="4">79H</td></tr>
<tr><td rowspan="2">Register</td><td colspan="24">System register</td></tr>
<tr><td colspan="16">Address register
(AR)</td><td colspan="4">Window register
(WR)</td><td colspan="4">Bank register
(BANK)</td></tr>
<tr><td>Symbol</td><td colspan="4">AR3</td><td colspan="4">AR2</td><td colspan="4">AR1</td><td colspan="4">AR0</td><td colspan="4">WR</td><td colspan="4">BANK</td></tr>
<tr><td>Bit</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td>Data</td><td>0</td><td>0</td><td colspan="14"></td><td colspan="4"></td><td>0</td><td>0</td><td colspan="2"></td></tr>
</table>

<table>
<tr><td>Address</td><td colspan="4">7AH</td><td colspan="4">7BH</td><td colspan="4">7CH</td><td colspan="4">7DH</td><td colspan="4">7EH</td><td colspan="4">7FH</td></tr>
<tr><td rowspan="3">Register</td><td colspan="24">System register</td></tr>
<tr><td colspan="12">Index register
(IX)</td><td colspan="7" rowspan="2">General-purpose
register pointer
(RP)</td><td colspan="5" rowspan="2">Program status word
(PSWORD)</td></tr>
<tr><td colspan="8">Data memory row
address pointer
(MP)</td><td colspan="4"></td></tr>
<tr><td rowspan="2">Symbol</td><td colspan="4">IXH</td><td colspan="4">IXM</td><td colspan="4" rowspan="2">IXL</td><td colspan="4" rowspan="2">RPH</td><td colspan="4" rowspan="2">RPL</td><td colspan="4" rowspan="2">PSW</td></tr>
<tr><td colspan="4">MPH</td><td colspan="4">MPL</td></tr>
<tr><td>Bit</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td>Data</td><td>M
P
E</td><td>0</td><td>0</td><td colspan="9">(IX)
(MP)</td><td>0</td><td>0</td><td colspan="5">(RP)</td><td>B
C
D</td><td>C
M
P</td><td>C
Y</td><td>Z</td><td>I
X
E</td></tr>
</table>

5.3 ADDRESS REGISTER (AR)

5.3.1 Format of Address Register

Fig. 5-3 shows the format of the address register.

As shown in Fig. 5-3, the address register consists of the 16 bits of 74H-77H (AR3-AR0) of a system register. However, the higher 2 bits are always set to 0, so that the address register actually operates as a 14-bit register.

Fig. 5-3 Format of Address Register

<table>
<tr><td colspan="2">Address</td><td colspan="4">74H</td><td colspan="4">75H</td><td colspan="4">76H</td><td colspan="4">77H</td></tr>
<tr><td colspan="2">Register</td><td colspan="16">Address register (AR)</td></tr>
<tr><td colspan="2">Symbol</td><td colspan="4">AR3</td><td colspan="4">AR2</td><td colspan="4">AR1</td><td colspan="4">AR0</td></tr>
<tr><td colspan="2">Bit</td><td>b_3</td><td>b_2</td><td>b_1</td><td>b_0</td><td>b_3</td><td>b_2</td><td>b_1</td><td>b_0</td><td>b_3</td><td>b_2</td><td>b_1</td><td>b_0</td><td>b_3</td><td>b_2</td><td>b_1</td><td>b_0</td></tr>
<tr><td colspan="2">Data</td><td>0</td><td>0</td><td>⟨MSB⟩</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>⟨LSB⟩</td></tr>
<tr><td rowspan="3">Upon reset</td><td>Power-on</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td></tr>
<tr><td>Clock stop</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td></tr>
<tr><td>CE</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td></tr>
</table>

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

5.3.2 Address Register Functions

The address register is used to specify program memory addresses for execution of a table reference instruction (MOVT DBF, @AR), stack manipulation instructions (PUSH AR and POP AR), indirect branch instruction (BR @AR), and indirect subroutine call instruction (CALL @AR).

For the address register, a dedicated instruction (INC AR) is available which can increment the address register by 1 at a time.

The operation performed when each instruction is executed is described (1) through (5) below.

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

The instruction loads the constant data (16 bits) held at the program memory address specified in the address register into the data buffer.

Constant data stored at addresses 0000H-2FFFH can be specified using the address register.

(2) Stack manipulation instructions (PUSH AR, POP AR)

When the PUSH AR instruction is executed, the stack pointer is decremented by 1, then the contents of the address register (AR) are stored in the address stack register pointed to by the decremented stack pointer.

When the POP AR is executed, the contents of the address stack register pointed to by the stack pointer are transferred to the address register, then the stack pointer is incremented by 1.

(3) Indirect branch instruction (BR @AR)

The instruction causes a branch to the program memory address specified by the address register.

A branch address from 0000H to 2FFFH can be specified using the address register.

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

The subroutine at the program memory address specified by the address register can be called.

A subroutine start address from 0000H to 2FFFH can be specified by the address register.

(5) Address register increment instruction (INC AR)

The instruction increments the address register by 1.

The address register consists of 14 bits. When the INC AR instruction is executed, however, the address register operates on a 13-bit basis. This means that the address specified after 1FFFH is not 2000H but 0000H. To increment the address register to 2000H, segment register switching is required. Note, however, that the address specified after 3FFFH is not 2000H but 0000H; in this case, segment register switching is not required.

5.3.3 Address Register and Data Buffer

The address register allows data transfer through the data buffer as part of peripheral hardware.

See **Chapter 9** for details.

5.3.4 Notes on Using Address Register

The address register consists of 14 bits, so that it can specify up to 3FFFH.

However, the program memory area consists of addresses 0000H-1EFFH and addresses 2000H-2FFFH.

Accordingly, a value from 0000H to 1EFFH or from 2000H to 2FFFH must be specified in the address register.

5.4 WINDOW REGISTER (WR)

5.4.1 Format of Window Register

Fig. 5-4 shows the format of the window register.

As shown in Fig. 5-4, the window register consists of the 4 bits of 78H of a system register.

Fig. 5-4 Format of Window Register

Address		78H			
Register		Window register (WR)			
Symbol		WR			
Bit		b_3	b_2	b_1	b_0
Data		(MSB)			(LSB)
Upon reset	Power-on	Undefined			
	Clock stop	The previous state is held.			
	CE				

5.4.2 Window Register Functions

The window register is used to transfer data to and from a register file (RF) described later.

For data transfer to and from a register file, the dedicated instructions PEEK WR, rf and POKE rf, WR are used (rf: register file address).

The operation performed when each instruction is executed is described in (1) and (2) below.

See also **Chapter 8**.

(1) PEEK WR, rf instruction

The instruction transfers the contents of the register file addressed by rf to the window register.

(2) POKE rf, WR instruction

The instruction transfers the contents of the window register to the register file addressed by rf.

5.5 BANK REGISTER (BANK)

5.5.1 Format of Bank Register

Fig. 5-5 shows the format of the bank register.

As shown in Fig. 5-5, the bank register consists of the 4 bits of 79H (BANK) of a system register. However, the higher 2 bits are always set to 0, so that the bank register actually operates as a 2-bit register.

Fig. 5-5 Format of Bank Register

Address		79H			
Register		Bank register (BANK)			
Symbol		BANK			
Bit		b_3	b_2	b_1	b_0
Data		0	0	(MSB)	(LSB)
Upon reset	Power-on	0			
	Clock stop	0			
	CE	0			

5.5.2 Bank Register Functions

The bank register specifies a data memory bank.

Table 5-1 indicates the bank register values and specified data memory banks.

A bank register is contained in each system register, so that it can be rewritten regardless of the bank currently specified.

This means that bank register manipulation is independent of the state of the currently specified bank.

Table 5-1 Data Memory Bank Specification

Bank register (BANK)				Data memory bank
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	Not to be set

5.6 INDEX REGISTER (IX) AND DATA MEMORY ROW ADDRESS POINTER (MP: MEMORY POINTER)

5.6.1 Index Register (IX)

(1) Format of Index Register

Fig. 5-6 shows the format of the index register.

The index register consists of 11 bits: the lower 3 bits (IXH) of 7AH of the system register, 7BH (IXM), and 7CH (IXL). The lower 2 bits (bits 2 and 1 of 7AH) are always set to 0. In operation to access the VRAM, however, only the higher 1 bit (bit 2 of 7AH) is always set to 0. For the method of VRAM access, see **Section 16.5.7**.

Fig. 5-6 Format of Index Register

Address	7AH				7BH				7CH				7EH				7FH			
Register	Index register (IX)																Program status word (PSWORD)			
	Memory pointer (MP)																			
Symbol	IXH				IXM				IXL								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
Data memory access	MPE	Fixed at 0	Fixed at 0																	IXE
		IX																		
				BANK		Row address			Column address											
VRAM access	MPE	Fixed at 0																		IXE
		IX																		
			VRAMBANK				Row address		Column address											
Upon reset: Power-on	0				0				0											0
Upon reset: Clock stop	0				0				0											0
Upon reset: CE	0				0				0											0

(2) Index register functions

The index register is used to modify data memory addresses when a data memory manipulation instruction is executed. That is, the data memory bank, row address, and column address specified by a data memory manipulation instruction are ORed with the contents of the index register, and the instruction is executed for the data memory location specified by the result of OR operation.

Note, however, that address modification is enabled only when the IXE flag (bit 0 of 7FH of the system register) is set to 1.

A dedicated instruction (INC IX) for incrementing the index register allows easy access to a data memory location.

Address modification using the index register can be performed with all data memory manipulation instructions.

With the instructions listed below, address modification using the index register is impossible.

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

For details of address modification, see **Chapter 7**.

5.6.2 Data Memory Row Address Pointer (MP)

(1) Format of data memory row pointer

Fig. 5-7 shows the format of the data memory row address pointer (referred to as the memory pointer).

The memory pointer consists of 7 bits: the lower 3 bits (MPH) of 7AH of the system register, and 7BH (MPL) of the system register. The higher 2 bits (bits 2 and 1 of 7AH) are always set to 0. In operation to access the VRAM, however, only the higher 1 bit (bit 2 of 7AH) is always set to 0. For the method of VRAM access, see **Section 16.5.7**.

(2) Memory pointer functions

When the general-purpose register indirect transfer instructions (MOV @r,m and MOV m,@r) are executed, the memory pointer is used to modify the indirect transfer destination address @r. That is, the bank and row address of the indirect transfer destination specified by an instruction is replaced with the contents of the memory pointer.

Note, however, that address modification is enabled only when the MPE flag (bit 3 of 7AH of the system register) is set to 1.

Address modification using the memory pointer can be performed only with the general-purpose register indirect transfer instructions.

Fig. 5-7 Format of Data Memory Row Address Pointer

Address	7AH				7BH				7CH				7EH				7FH			
Register	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 memory access	MPE	Fixed at 0	Fixed at 0																	IXE
		MP		BANK		Row address														
VRAM access	MPE	Fixed at 0																		IXE
		MP	VRAMBANK				Row address													
Upon reset: Power-on	0				0				0											0
Upon reset: Clock stop	0				0				0											0
Upon reset: CE	0				0				0											0

5.7 GENERAL-PURPOSE REGISTER POINTER (RP)

5.7.1 Format of General-Purpose Register Pointer

Fig. 5-8 shows the format of the general-purpose register pointer.

As shown in Fig. 5-8, the general-purpose register pointer consists of 7 bits: the 4 bits of address 7DH (RPH) of the system register, and the higher 3 bits of address 7EH (RPL) of the system register. However, the higher 2 bits of address 7DH are always set to 0, so that the lower 5 bits (lower 2 bits of address 7DH and higher 3 bits of address 7EH) are usable.

Fig. 5-8 Format of General-Purpose Register Pointer

<table>
<tr><td colspan="2">Address</td><td colspan="4">7DH</td><td colspan="4">7EH</td></tr>
<tr><td colspan="2">Register</td><td colspan="7">General-purpose register pointer (RP)</td><td></td></tr>
<tr><td colspan="2">Symbol</td><td colspan="4">RPH</td><td colspan="4">RPL</td></tr>
<tr><td colspan="2">Bit</td><td>b_3</td><td>b_2</td><td>b_1</td><td>b_0</td><td>b_3</td><td>b_2</td><td>b_1</td><td>b_0</td></tr>
<tr><td colspan="2">Data</td><td>0</td><td>0</td><td>⟨MSB⟩</td><td></td><td></td><td></td><td>⟨LSB⟩</td><td>BCD</td></tr>
<tr><td rowspan="3">Upon reset</td><td>Power-on</td><td colspan="4">0</td><td colspan="4">0</td></tr>
<tr><td>Clock stop</td><td colspan="4">0</td><td colspan="4">0</td></tr>
<tr><td>CE</td><td colspan="4">0</td><td colspan="4">0</td></tr>
</table>

5.7.2 General-Purpose Register Pointer Functions

The general-purpose register pointer specifies a general-purpose register in the data memory.

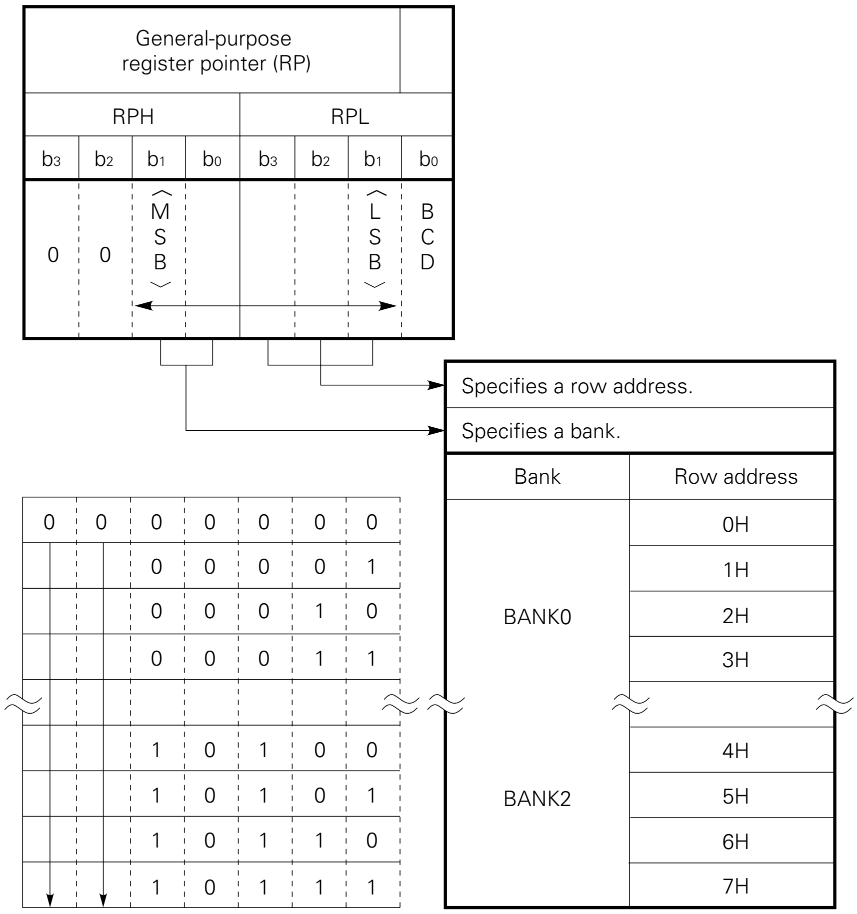
Fig. 5-9 shows the address of a general-purpose register specified with the general-purpose register pointer.

As shown in Fig. 5-9, the higher 4 bits (RPH: address 7DH) of the general-purpose register pointer specify a bank, and the lower 3 bits (RPL: address 7EH) of the general-purpose register pointer specify a row address.

The effective bits of the general-purpose register pointer are the five bits, so that any row address (0H-7H) of any bank can be specified as a general-purpose register. However, when the VRAMSEL flag (RF: 33H, bit 3) is set to 1, the VRAM area and 40H-6FH of BANK2 cannot be specified as general-purpose registers.

See **Chapter 6** for details of general-purpose register operation.

Fig. 5-9 General-Purpose Register Addresses Specified by General-Purpose Register Pointer



5.7.3 Notes on Using General-Purpose Register Pointer

The low-order bit of address 7EH (RPL) of the general-purpose register pointer is used as the BCD flag of the program status word.

Pay attention to the value of the BCD flag when rewriting RPL.

5.8 PROGRAM STATUS WORD (PSWORD)

5.8.1 Format of Program Status Word

Fig. 5-10 shows the format of the program status word.

As shown in Fig. 5-10, the program status word consists of 5 bits: the low-order bit of 7EH (RPL) of the system register, and the 4 bits of address 7FH (PSW) of the system register.

A different function is assigned to each bit of the program status word; the program status word consists of a BCD flag (BCD), compare flag (CMP), carry flag (CY), zero flag (Z), and index enable flag (IXE).

Fig. 5-10 Format of Program Status Word

Address		7EH				7FH			
Register		(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
Upon reset	Power-on	0				0			
	Clock stop	0				0			
	CE	0				0			

5.8.2 Program Status Word Functions

The program status word is used to set conditions for transfer instructions and operations by the arithmetic logic unit (ALU), and also to indicate the states of the results of operations.

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

See **Chapter 7** for details.

Table 5-2 Outline of Function of Each Flag of Program Status Word

(RP)			Program status word (PSWORD)				
RPL				PSW			
b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
			BCD	CMP	CY	Z	IXE

Flag name	Function
Index enable flag (IXE)	Used to specify whether a data memory address is to be modified when a data memory manipulation instruction is executed. 0 : Not modified 1 : Modified
Zero flag (Z)	Used to indicate that the result of an arithmetic operation is 0. Note that the states of 0 and 1 differ, depending on the value of the compare flag.
Carry flag (CY)	Used to indicate the occurrence of a carry or borrow as the result of an addition or subtraction instruction executed. This flag is reset to 0 when neither a carry nor a borrow is produced. This flag is set to 1 when a carry or borrow is produced. This flag is used also as a shift bit for the RORC r instruction.
Compare flag (CMP)	Used to specify whether to store the result of an arithmetic operation in a data memory area or general-purpose register. 0 : Stores the result. 1 : Does not store the result.
BCD flag (BCD)	Used to specify whether to perform an arithmetic operation in decimal. 0 : Performs a binary operation. 1 : Performs a decimal operation.

5.8.3 Notes on Using Program Status Word

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

If an operation is performed which produces the result 0000B with a carry, for example, 0000B is stored in the PSW.

5.9 NOTES ON USING SYSTEM REGISTER

Those data items in the program status word that are always set to 0 are not affected by an attempt to execute a write instruction.

When those data items in the program status word that are always set to 0 are read, 0 is read.

6. GENERAL-PURPOSE REGISTER (GR)

6.1 OUTLINE OF GENERAL-PURPOSE REGISTER

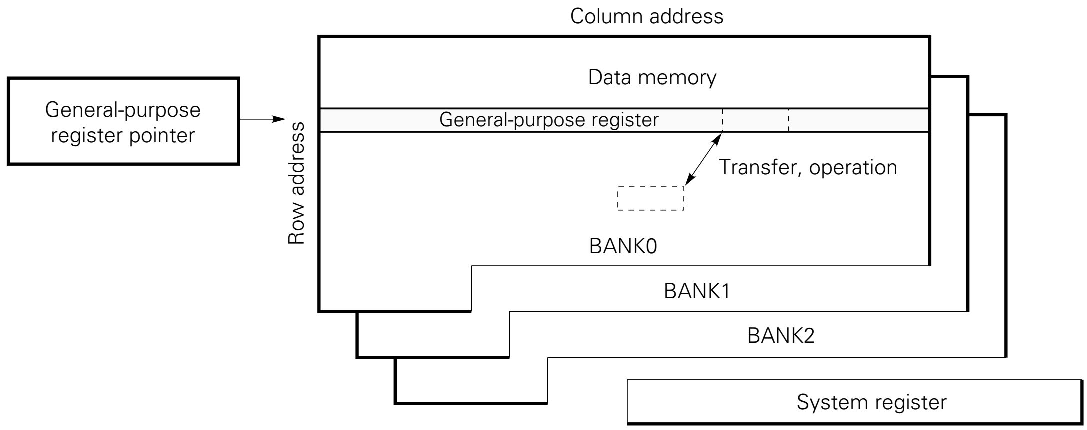
Fig. 6-1 outlines the general-purpose register.

As shown in Fig. 6-1, the general-purpose register consists of a general-purpose register pointer and general-purpose register body.

The bank and row address of a general-purpose register body is specified with the general-purpose register pointer.

A general-purpose register pointer body is used to perform an operation with or transfer data to and from a data memory area.

Fig. 6-1 Outline of General-Purpose Register



6.2 GENERAL-PURPOSE REGISTER BODY

The general-purpose register body consists of a row on the data memory, which is 16 nibbles (16 × 4 bits) long.

See **Section 5.7** for information about the general-purpose register pointer, and a bank and row address specifiable as a general-purpose register.

One instruction can be used to perform an operation with or transfer data to and from a 16-nibble row specified as a general-purpose register.

This means that an operation or data transfer between data memory areas can be performed with one instruction.

As with other data memory areas, a general-purpose register can be controlled using data memory manipulation instructions.

6.3 GENERAL-PURPOSE REGISTER ADDRESS GENERATION WITH INSTRUCTIONS

Sections 6.3.1 and **6.3.2** below describe general-purpose register address generation when each instruction is executed.

For the detailed operation of each instruction, see **Chapter 7**.

6.3.1 Addition Instructions (ADD r,m, ADDC r,m)
Subtraction Instructions (SUB r,m, SUBC r,m)
Logical Operation Instructions (AND r,m, OR r,m, XOR r,m)
Direct Transfer Instructions (LD r,m, ST m,r), and
Rotate Instruction (RORC r)

Table 6-1 indicates a general-purpose register address specified by operand r of an instruction. Operand r specifies only a column address.

Table 6-1 General-Purpose 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-purpose register address	Contents of general-purpose register pointer							r			

6.3.2 Indirect Transfer Instructions (MOV @r,m, MOV m,@r)

Table 6-2 indicates a general-purpose register address specified by operand r of an instruction, and an indirect transfer address specified by @r.

Table 6-2 General-Purpose 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-purpose register address	Contents of general-purpose register pointer							r			
Indirect transfer address	Same as data memory							Contents of r			

6.4 NOTES ON USING GENERAL-PURPOSE REGISTER

6.4.1 Row Address of General-Purpose Register

The row address of a general-purpose register is specified by the general-purpose register pointer. Accordingly, note that the currently specified bank may differ from the bank of the general-purpose register specified.

6.4.2 Operation between General-Purpose Register and Immediate Data

No instruction is available for operation between a general-purpose register and immediate data.

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

7. ARITHMETIC LOGIC UNIT (ALU) BLOCK

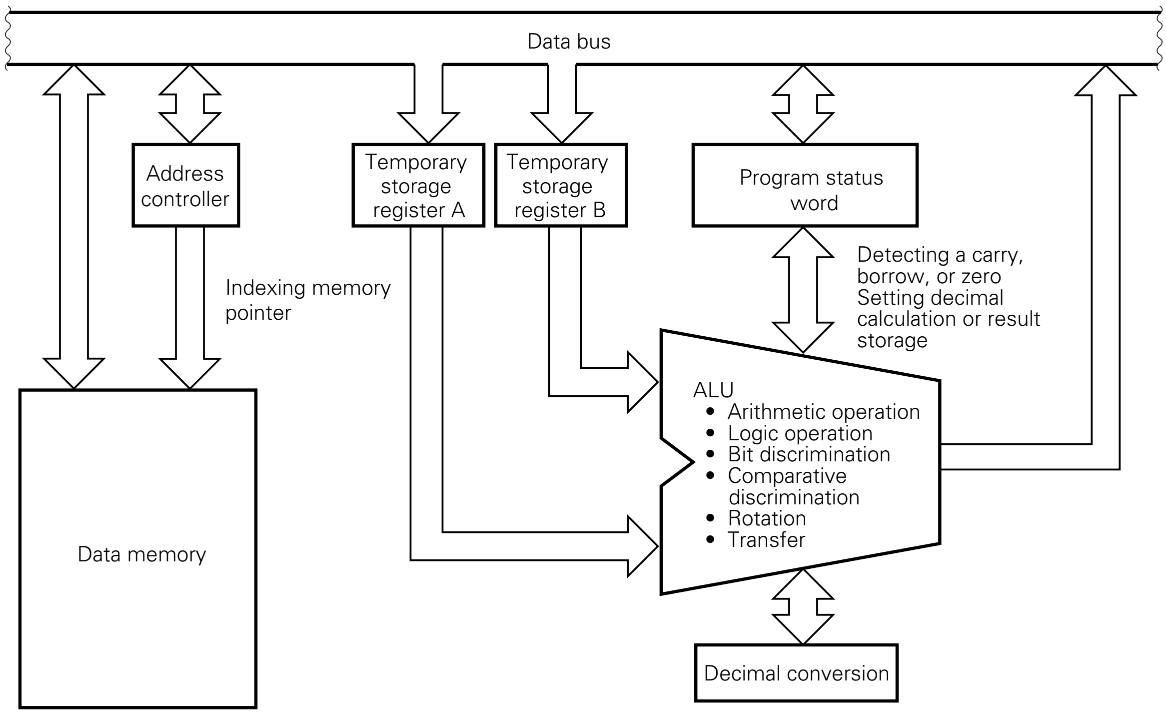
7.1 OVERVIEW

Fig. 7-1 is an overview of the ALU block.

As shown in Fig. 7-1, the ALU block consists of the ALU, temporary storage registers A and B, program status word, decimal conversion circuit, and data memory address controller.

The ALU performs arithmetic and logic operations on the 4-bit data in the data memory and performs discrimination, comparison, rotation, and transfer.

Fig. 7-1 Overview of the ALU Block



7.2 CONFIGURATION AND FUNCTIONS OF THE COMPONENTS OF THE ALU BLOCK

7.2.1 ALU

In response to a programmed instruction, the ALU performs 4-bit arithmetic or logic processing, bit discrimination, comparative discrimination, rotation, or transfer.

7.2.2 Temporary Storage Registers A and B

Temporary storage registers A and B temporarily hold the 4-bit data.

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

7.2.3 Program Status Word

A program status word controls the operation of the ALU and holds the status of the ALU.

For details of the program status word, see **Section 5.8**.

7.2.4 Decimal Conversion Circuit

If the BCD flag of the program status word is set to 1 when an arithmetic operation is executed, the decimal conversion circuit converts the results of the arithmetic operation to a decimal number.

7.2.5 Address Controller

The address controller specifies an address in data memory.

At the same time, the circuit also controls address modification by the index register or data memory row address pointer.

7.3 ALU OPERATIONS

Table 7-1 lists the operations performed by the ALU when instructions are executed.

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

Table 7-3 lists the converted decimal data used in decimal operations.

Table 7-1 ALU Operations

ALU function	Instruction		Operation difference due to program status word (PSWORD)					Address modification	
			Value of the BCD flag	Value of the CMP flag	Operation	Operation of the CY flag	Operation of the Z flag	Index	Memory pointer
Addition	ADD	r, m	0	0	Binary operation The result is stored.	Set by a carry or borrow. Otherwise, the flag is reset.	Set if the operation result is 0000B. Otherwise, the flag is reset.	Provided	Not provided
		m, #n4							
	ADDC	r, m	0	1	Binary operation The result is not stored.		Retains the status if the operation result is 0000B. Otherwise, the flag is reset.		
		m, #n4							
Subtraction	SUB	r, m	1	0	Decimal operation The result is stored.		Set if the operation result is 0000B. Otherwise, the flag is reset.		
		m, #n4							
	SUBC	r, m	1	1	Decimal operation The result is not stored.		Retains the status if the operation result is 0000B. Otherwise, the flag is reset.		
		m, #n4							
Logic operation	OR	r, m	Optional (hold)	Optional (hold)	Not changed	Retains the previous state.	Retains the previous state.	Provided	Not provided
		m, #n4							
	AND	r, m							
		m, #n4							
	XOR	r, m							
		m, #n4							
Discrimi-nation	SKT	m, #n	Optional (hold)	Optional (reset)	Not changed	Retains the previous state.	Retains the previous state.	Provided	Not provided
	SKF	m, #n							
Comparison	SKE	m, #n4	Optional (hold)	Optional (hold)	Not changed	Retains the previous state.	Retains the previous state.	Provided	Not provided
	SKNE	m, #n4							
	SKGE	m, #n4							
	SKLT	m, #n4							
Transfer	LD	r, m	Optional (hold)	Optional (hold)	Not changed	Retains the previous state.	Retains the previous state.	Provided	Not provided
	ST	m, r							
	MOV	m, #n4							
		@r, m							Provided
		m, @r							
Rotation	RORC	r	Optional (hold)	Optional (hold)	Not changed	Value of b_0 of the general-purpose register	Retains the previous state.	Not provided	Not provided

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

IXE	MPE	General-purpose register address specified with r			Data memory address specified with m			Indirect transfer address specified with @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		Same as above			Same as above		MP		(r)
1	0		Same as above		BANK Logical OR IX	m		BANK Logical OR IXH, IXM	m_R	(r)
1	1		Same as above			Same as above		MP		(r)

BANK : Bank register
IX : Index register
IXE : Index enable flag
IXH : Bits 10 to 8 of the index register
IXM : Bits 7 to 4 of the index register
IXL : Bits 3 to 0 of the index register
m : Data memory address specified with m_R and m_C
m_R : Data memory row address (high order)
m_C : Data memory column address (low order)
MP : Data memory row address pointer
MPE : Memory pointer enable flag
r : General-purpose register column address
RP : General-purpose register pointer
(x) : Contents addressed by x
x : m, r, and other direct address

Table 7-3 Converted Decimal 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 subtraction		Decimal subtraction	
	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 Correct decimal conversion is not possible in the shaded area.

7.4 NOTES ON USING THE ALU

7.4.1 Notes on Using the Program Status Word for Operations

After an arithmetic operation has been performed on the program status word, the operation result is held in the program status word.

The CY and Z flags of the program status word are usually set or reset according to the result of the arithmetic operation. If the arithmetic operation is performed on the program status word itself, the result of the operation is stored and a carry, borrow, or zero cannot be discriminated.

If the CMP flag is set, the result of the arithmetic operation is not stored and the CY and Z flags are set or reset as usual.

7.4.2 Notes on Performing Decimal Operations

A decimal operation can be carried out only when the operation result is within the following ranges:

(1) The result of addition is between 0 and 19 in decimal.
(2) The result of subtraction is between 0 and 9 or –10 and –1 in decimal.

If a decimal operation exceeding the above ranges is performed, the CY flag is set, resulting in a value greater than or equal to 1010B (0AH).

8. REGISTER FILE (RF)

8.1 OVERVIEW

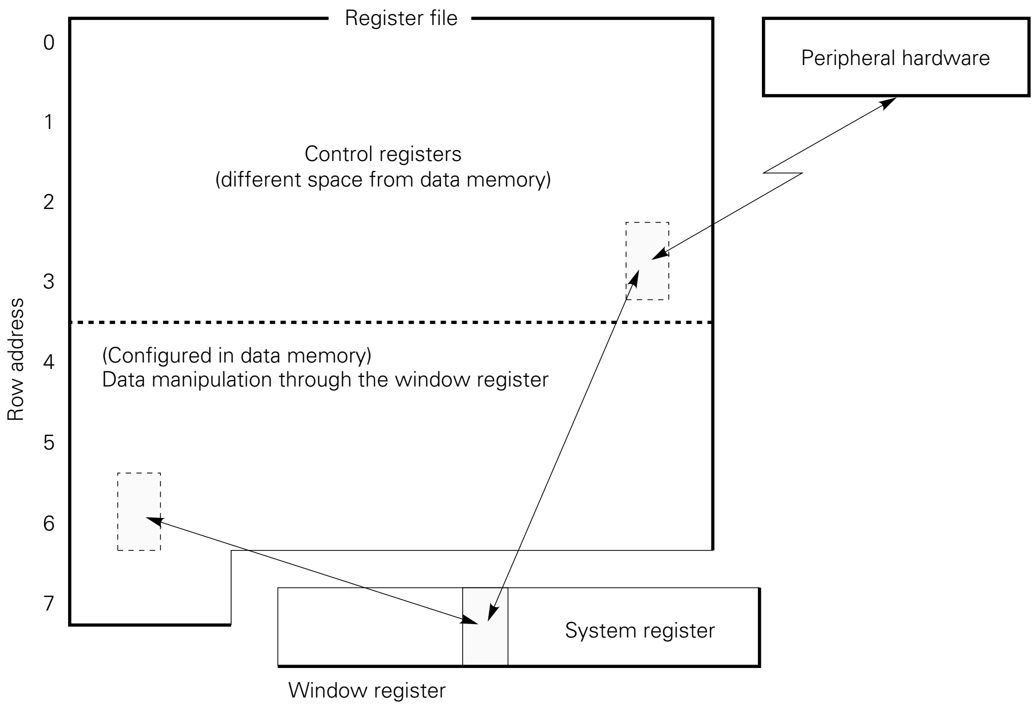
Fig. 8-1 shows an overview of the register file.

As shown in Fig. 8-1, the register file consists of control registers in a different space from data memory, and a part of data memory.

The control register sets the conditions of the peripheral hardware.

The data in the register file is read and written through a window register.

Fig. 8-1 Overview of the Register File



8.2 CONFIGURATION AND FUNCTIONS OF THE REGISTER FILE

Fig. 8-2 shows the configuration of the register file and the relationship between the register file and data memory.

Like data memory, the register file is assigned addresses in units of four bits. The row addresses range from 0H to 7H and the column addresses from 0H to 0FH, that is, 128 nibbles in total.

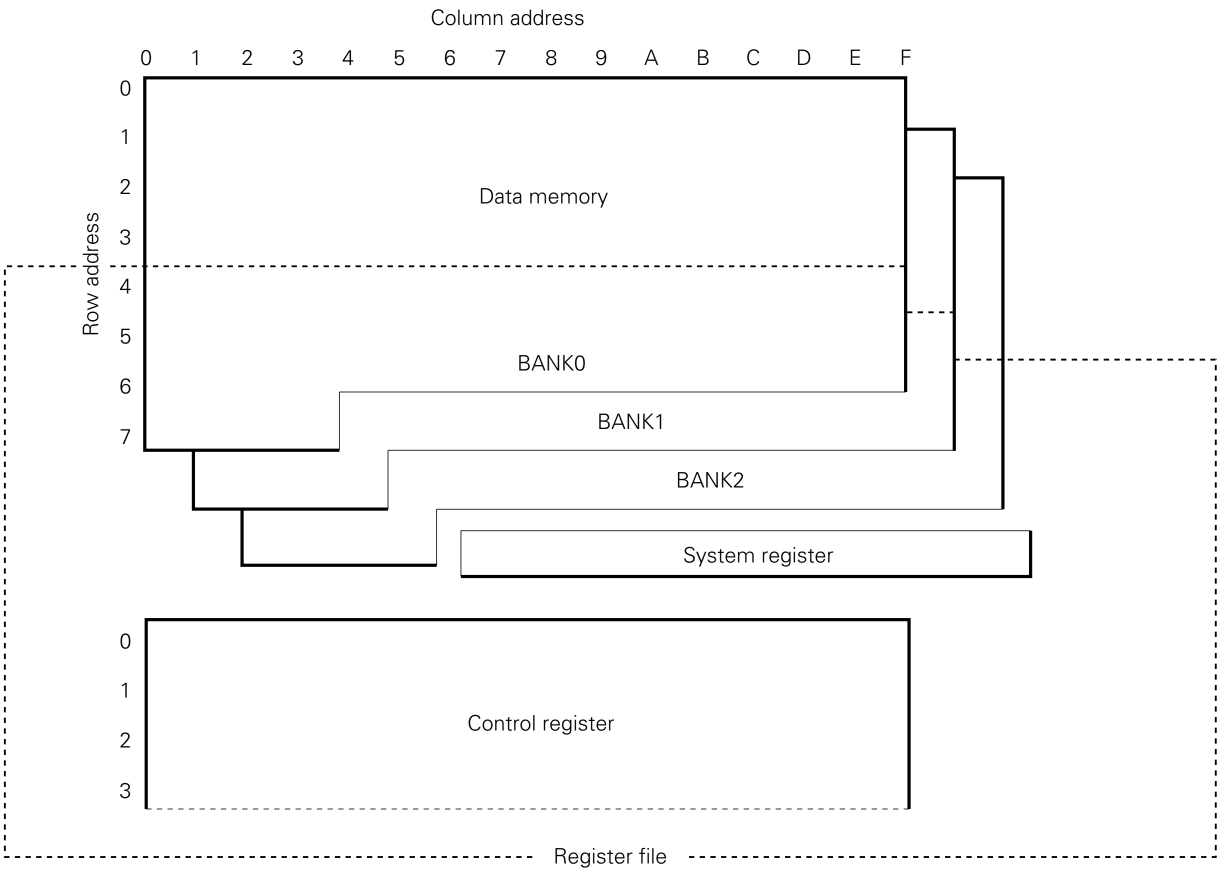
The address locations from 00H to 3FH are referred to as control registers and are used to set the conditions for the peripheral hardware.

Address locations 40H to 7FH and data memory overlap.

The data at addresses 40H to 7FH in the register file is identical to that at addresses 40H to 7FH in the current bank of data memory.

Because address locations 40H to 7FH and data memory overlap, they are the same as the ordinary address locations in data memory, except that they can be manipulated by the register file manipulation instructions (PEEK WR, rf and POKE rf, WR).

Fig. 8-2 Configuration of the Register File and the Relationship between the Register File and Data Memory



8.2.1 Register File Manipulation Instructions (PEEK WR, rf and POKE rf, WR)

Data in the register file is read and written through the window register of the system register. The following instructions are used:

(1) PEEK WR, rf

Reads the data at address rf of the register file into the window register.

(2) POKE rf, WR

Writes the data of the window register at address rf into the register file.

8.3 CONTROL REGISTERS

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

As shown in Fig. 8-3, the control registers consist of 64 nibbles (64 x 4 bits) of addresses 00H to 3FH in the register file.

However, only 61 nibbles are actually used. The remaining three nibbles are not used, reading and writing for these nibbles being inhibited.

Each nibble of each control register has an attribute. Each nibble has one of the following four attributes: read/write (R/W), read only (R), write only (W), and reset at reading (R & Reset).

If writing to a read-only (R, or R & Reset) register is attempted, nothing changes.

If reading from a write-only (W) register is attempted, an undefined value is read.

Of the four bits in a single nibble, a bit that is always set to 0 is always read as 0. Even if writing is attempted, the bit remains set to 0.

If an attempt is made to read the contents of the three unused nibbles, an undefined value is read. If writing to the unused part is attempted, nothing changes.

[MEMO]

Fig. 8-3 Configuration of the Control Registers (1/2)

Row Address	Item (Column Address →)	0	1	2	3	4	5	6	7
0 (8)[Note]	Register		Stack pointer (SP)	CE pin edge detection register	PWM mode select register 3	PWM mode select register 2	PWM mode select register 1	Watch timer mode select register	CE pin level judge register
	Symbol		SP3, SP2, SP1, SP0	0, 0, 0, CEDET	0, 0, 0, PWM8SEL	PWM7SEL, PWM6SEL, PWM5SEL, PWM4SEL	PWM3SEL, PWM2SEL, PWM1SEL, PWM0SEL	WTMHLD, 0, CKOSEL, XTSEL	0, 0, 0, CE
	Read/Write		R/W	R & Reset	R/W	R/W	R/W	W (WTMHLD, 0, CKOSEL); R/W (XTSEL)	R
1 (9)[Note]	Register		HSYNC-counter-gate control register	HSYNC-counter-gate judge register	PLL reference clock select register	Watch timer reset register	INTNC mode select register	Basic timer 1 carry flip-flop judge register	Basic timer 0 carry flip-flop judge register
	Symbol		0, 0, HSCGT1, HSCGT0	HSCGOSTT, 0, 0, 0	PLLRFCK3, PLLRFCK2, PLLRFCK1, PLLRFCK0	WTMRES3, WTMRES2, WTMRES1, WTMRES0	0, INTNCMD2, INTNCMD1, INTNCMD0	0, 0, 0, BTM1CY	0, 0, 0, BTM0CY
	Read/Write		R/W	R	R/W	R/W	R/W	R & Reset	R & Reset
2 (A)[Note]	Register	Clock-stop release enable register	A/D converter channel select register	PLL-unlock-flip-flop judge register		A/D converter control register		Port 2D bit I/O select register	Port 1C group I/O select register
	Symbol	0, 0, 0, RLSEN	0, ADCCH2, ADCCH1, ADCCH0	0, 0, 0, PLLUL		ADCEN, 0, 0, ADCCMP		0, P2DBIO2, P2DBIO1, P2DBIO0	0, 0, 0, P1CGIO
	Read/Write	R/W	R/W	R & Reset		R/W (ADCEN, 0, 0); R (ADCCMP)		R/W	R/W
3 (B)[Note]	Register	IDC background select register	IDC enable register	PLL-unlock-flip-flop sensibility select register	IDC mode select register	P1B2 pin edge detection register	Port 1B bit I/O select register	Port 0B bit I/O select register	Port 0A bit I/O select register
	Symbol	IDCBKEN, IDCBKR, IDCBKG, IDCBKB	0, 0, 0, IDCEN	0, 0, PLULSEN1, PLULSEN0	VRAMSEL, IDCSEL, IDCD14SL, IDCCPCH	0, 0, 0, P1B2EDET	P1BBIO3, P1BBIO2, P1BBIO1, P1BBIO0	P0BBIO3, P0BBIO2, P0BBIO1, P0BBIO0	P0ABIO3, P0ABIO2, P0ABIO1, P0ABIO0
	Read/Write	R/W	R/W	R /W	R/W	R & Reset	R/W	R/W	R/W

Note An address used with the assembler (AS17K) is indicated in parentheses.

Fig. 8-3 Configuration of the Control Registers (2/2)

8	9	A	B	C	D	E	F
Serial I/O 0 mode select register	Timer 0 clock select register	Basic timer 2 mode select register	Basic timer 1 mode select register	Basic timer 0 clock select register	Timer 0 control register	Timer 0 overflow register	Interrupt group selection register
SIO0CH SB SIO0MS SIO0TX	0 0 0 TM0CK	BTM2EXCK BTM2ZX BTM2CK1 BTM2CK1	BTM1EXCK BTM1ZX BTM1CK1 BTM1CK0	0 0 BTM0CK1 BTM0CK0	0 TM0RPT TM0RES TM0EN	0 0 0 TM0VF	0 0 IGRP1SL IGRP0SL
R/W	R/W	R/W	R/W	R/W	R/W, W, R/W	R	R/W
Serial I/O 0 wait control register		Timer 1 clock select register	Timer 1 control register	Serial I/O 1 mode select register	Watch timer 8-Hz carry register	Watch timer 128-Hz carry register	Interrupt edge selection register
SBACK SIO0NWT SIO0WRQ1 SIO0WRQ0		0 0 TM1CK1 TM1CK0	0 0 TM1RES TM1EN	SIO1TS SIO1HIZ SIO1CK1 SIO1CK0	0 0 0 WTM8HZ	0 0 0 WTM128HZ	0 IEGGRP1 IEG0 IEGNC
R/W		R/W	R/W, W, R/W	R/W	R & Reset	R & Reset	R/W
Serial I/O 0 status judge register	Interrupt request register 10	Interrupt request register 9	Interrupt request register 8	Interrupt request register 7	Interrupt enable register 3	Interrupt enable register 2	Interrupt enable register 1
SIO0SF8 SIO0SF9 SBSTT SBBSY	0 0 0 IRQGRP0	0 0 0 IRQSIO1	0 0 0 IRQSIO0	INTGRP1 0 0 IRQGRP1	0 0 IPGRP0 IPSIO1	IPSIO0 IPGRP1 IPIDCVP IPBTM2	IPTM1 IPTM0 IP0 IPNC
R	R/W	R/W	R/W	R, R/W	R/W	R/W	R/W
Serial I/O 0 interrupt mode register	Serial I/O 0 clock select register	Interrupt request register 6	Interrupt request register 5	Interrupt request register 4	Interrupt request register 3	Interrupt request register 2	Interrupt request register 1
0 0 SIO0IMD1 SIO0IMD0	0 0 SIO0CK1 SIO0CK0	0 0 0 IRQIDCVP	0 0 0 IRQBTM2	0 0 0 IRQTM1	0 0 0 IRQTM0	INT0 0 0 IRQ0	INTNC 0 0 IRQNC
R/W	R/W	R/W	R/W	R/W	R/W	R, R/W	R, R/W

Table 8-1 Peripheral Hardware Control Functions of the Control Registers (1/7)

Peripheral hardware	Register	Address	Read/Write	Symbol (b3 / b2 / b1 / b0)	Function overview	Set value 0	Set value 1	Upon reset: Power on	Upon reset: STOP	Upon reset: CE
Stack	Stack pointer (SP)	01H	R/W	(SP3)	Stack pointer			12	12	12
				(SP2)						
				(SP1)						
				(SP0)						
Timer	Watch timer mode register	06H	W	WTMHLD	Selects whether to hold the watch timer for 500 ms.	Does not hold.	Holds.	0	H	H
			W	0	Always set to 0.			0	0	0
			W	CKOSEL	Selects whether to output an oscillation frequency of 32.768 kHz.	Does not output.	Outputs.	0	0	0
			R/W	XTSEL	Selects the function of the P0D1 and P0D0 pins.	Operates as a port.	Connects an oscillator.	0	H	H
	Timer 0 clock select register	09H	R/W	0	Always set to 0.			0	0	H
				0						
				0						
				TM0CK	Sets the clock of timer 0.	10 μs	50 μs			
	Basic timer 2 mode select register	0AH	R/W	BTM2EXCK	Selects the base clock (internal/ external).	0, 4: 100 ms 1,5: 5 ms 2,6: 1 ms 3,7: 125 μs 8, 9: Divides the external clock by 5. A, B: Divides the external clock by 6. C, D: Divides the external clock by 5 (with zero-cross on). E, F: Divides the external clock by 6 (with zero-cross on).		0	0	H
				BTM2ZX	Turns the zero-cross circuit on or off.					
				BTM2CK1	Sets an interrupt time.					
				BTM2CK0						
	Basic timer 1 mode select register	0BH	R/W	BTM1EXCK	Selects the base clock (internal/ external).	0, 4: 100 ms 1,5: 5 ms 2,6: 1 ms 3,7: 125 μs 8, 9: Divides the external clock by 5. A, B: Divides the external clock by 6. C, D: Divides the external clock by 5 (with zero-cross on). E, F: Divides the external clock by 6 (with zero-cross on).		0	0	H
				BTM1ZX	Turns the zero-cross circuit on or off.					
				BTM1CK1	Sets a carry flip-flop time.					
				BTM1CK0						
	Basic timer 0 mode select register	0CH	R/W	0	Always set to 0.			0	0	H
				0						
				BTM0CK1	Sets a carry flip-flop time.	0 0 0 0 100 ms 5 ms 1 ms 1 ms 0 0 0 0				
				BTM0CK0						
	Timer 0 control register	0DH	R/W	0	Always set to 0.			0	0	H
			R/W	TM0RPT	Selects the operation mode of timer 0.	Free-run count mode	Modulo count mode			
			W	TM0RES	Selects whether to reset the timer 0 counter.	Does not perform reset.	Resets.			
			R/W	TM0EN	Selects whether to start the timer 0 counter.	Does not start.	Starts.			
	Timer 0 overflow register	0EH	R	0	Always set to 0.			0	0	H
				0						
				0						
				TM0OVF	Detects whether the timer 0 counter overflows.	Does not overflow.	Overflows.			
	Watch timer reset register	14H	R/W	WTMRES3	Selects whether to reset the watch timer.	Does not reset.	Resets.	0	H	H
				WTMRES2						
				WTMRES1						
				WTMRES0						

Remark H: Holds the previous state.

Table 8-1 Peripheral Hardware Control Functions of the Control Registers (2/7)

Peripheral hardware	Register	Address	Read/Write	Symbol (b3/b2/b1/b0)	Function overview	Set value 0	Set value 1	Upon reset: Power on	Upon reset: STOP	Upon reset: CE
Timer	Basic timer 1 carry flip-flop judge register	16H	R & RES	0	Always set to 0.			0	1	1
				0						
				0						
				BTM1CY	Detects the status of the carry flip-flop.	Reset	Set			
	Basic timer 0 carry flip-flop judge register	17H	R & RES	0	Always set to 0.			0	1	1
				0						
				0						
				BTM0CY	Detects the status of the carry flip-flop.	Reset	Set			
	Timer 1 clock select register	1AH	R/W	0	Always set to 0.			0	0	H
				0						
				TMCK1	Sets the clock of timer 1.	TMCK1, TMCK0 = 0, 0: 1 ms; 0, 1: 100 μs; 1, 0: 50 μs; 1, 1: 10 μs				
				TMCK0						
	Timer 1 control register	1BH	R/W	0	Always set to 0.			0	0	H
				0						
			W	TM1RES	Selects whether to reset the timer 1 counter.	Does not perform reset.	Resets.			
			R/W	TM1EN	Selects whether to operate the timer 1 counter.	Does not start.	Starts.			
	Watch timer 8-Hz carry register	1DH	R & RES	0	Always set to 0.			0	H	H
				0						
				0						
				WTM8HZ	Detects the status of the carry flip-flop.	Reset	Set			
	Watch timer 128-Hz carry register	1EH	R & RES	0	Always set to 0.			0	H	H
				0						
				0						
				WTM128HZ	Detects the status of the carry flip-flop.	Reset	Set			
Interrupt	Interrupt group selection register	0FH	R/W	0	Always set to 0.			0	0	0
				0						
				IGRP1SL	Selects an interrupt source (group 1).	V_{SYNC} signal	H_{SYNC} signal			
				IGRP0SL	Timer 0 overflow interrupt (group 0)	Is not used.	Is used.			
	INT_{NC} mode select register	15H	R/W	0	Always set to 0.			0	0	0
				INTNCMD2	Selects the pulse width used for accepting the interrupt of the INT_{NC} pin.	0 : Accepts at the edge. 1 : 200 μs 2 : 400 μs 3 : 2 ms 4 : 4 ms				
				INTNCMD1						
				INTNCMD0						
	Interrupt edge selection register	1FH	R/W	0	Always set to 0.			0	0	0
				IEGGRP1	Sets the edge where an interrupt is issued (group 1).	Rising edge	Falling edge			
				IEG0	Sets the edge where an interrupt is issued (INT_0).	Rising edge	Falling edge			
				IEGNC	Sets the edge where an interrupt is issued (INT_{NC}).	Rising edge	Falling edge			

Remark H: Holds the previous state.

Table 8-1 Peripheral Hardware Control Functions of the Control Registers (3/7)

Peripheral hardware	Register	Address	Read/Write	Symbol (b3 / b2 / b1 / b0)	Function overview	Set value 0	Set value 1	Upon reset: Power on	Upon reset: STOP	Upon reset: CE
Interrupt	Interrupt request register 10	29H	R/W	0	Always set to 0.	Low level	High level	0	0	0
				0						
				0						
				IRQGRP0	Detects an interrupt request (group 0).	Detects no interrupt request or interrupt handling is in progress.	Detects an interrupt request.			
	Interrupt request register 9	2AH	R/W	0	Always set to 0.			0	0	0
				0						
				0						
				IRQSIO1	Detects an interrupt request (SIO_1).	Detects no interrupt request or interrupt handling is in progress.	Detects an interrupt request.			
	Interrupt request register 8	2BH	R/W	0	Always set to 0.			0	0	0
				0						
				0						
				IRQSIO0	Detects an interrupt request (SIO_0).	Detects no interrupt request or interrupt handling is in progress.	Detects an interrupt request.			
	Interrupt request register 7	2CH	R	INTGRP1	Displays the level of the H_{SYNC}/V_{SYNC} signal.	Low level	High level	0	0	0
			R/W	0	Always set to 0.					
				0						
				IRQGRP1	Displays an interrupt request (group 1).	Detects no interrupt request or interrupt handling is in progress.	Detects an interrupt request.			
	Interrupt request register 6	3AH	R/W	0	Always set to 0.			0	0	0
				0						
				0						
				IRQIDCVP	Detects an interrupt request (VRAM pointer).	Detects no interrupt request or interrupt handling is in progress.	Detects an interrupt request.			
	Interrupt request register 5	3BH	R/W	0	Always set to 0.			0	0	0
				0						
				0						
				IRQBTM2	Detects an interrupt request (BTM2).	Detects no interrupt request or interrupt handling is in progress.	Detects an interrupt request.			
	Interrupt request register 4	3CH	R/W	0	Always set to 0.			0	0	0
				0						
				0						
				IRQTM1	Detects an interrupt request (TM1).	Detects no interrupt request or interrupt handling is in progress.	Detects an interrupt request.			
	Interrupt request register 3	3DH	R/W	0	Always set to 0.			0	0	0
				0						
				0						
				IRQTM0	Detects an interrupt request (TM0).	Detects no interrupt request or interrupt handling is in progress.	Detects an interrupt request.			
	Interrupt request register 2	3EH	R	INT0	Displays the level of the signal input to the INT_0 pin.	Low level	High level	0	0	0
			R/W	0	Always set to 0.					
				0						
				IRQ0	Detects an interrupt request (INT_0 pin).	Detects no interrupt request or interrupt handling is in progress.	Detects an interrupt request.			

Table 8-1 Peripheral Hardware Control Functions of the Control Registers (4/7)

Peripheral hardware	Register	Address	Read/Write	Symbol (b3 b2 b1 b0)	Function overview	Set value 0	Set value 1	Upon reset: Power on	Upon reset: STOP	Upon reset: CE
Interrupt	Interrupt request register 1	3FH	R	INTNC	Displays the level of the signal input to the INT_{NC} pin.	Low level	High level	0	0	0
			R/W	0	Always set to 0.					
				0						
				IRQNC	Detects an interrupt request (INT_{NC} pin).	Detects no interrupt request or interrupt handling is in progress.	Detects an interrupt request.			
	Interrupt enable register 3	2DH	R/W	0	Always set to 0.			0	0	0
				0						
				IPGRP0	Group 0 — Selects whether to enable an interrupt.	Inhibits an interrupt.	Enables an interrupt.			
				IPSI01	Serial interface 1					
	Interrupt enable register 2	2EH	R/W	IPSIO0	Serial interface 0			0	0	0
				IPGRP1	Group 1					
				IPIDCVP	VRAM pointer					
				IPBTM2	Basic timer 2					
	Interrupt enable register 1	2FH	R/W	IPTM1	Timer 1			0	0	0
				IPTM0	Timer 0					
				IP0	INT0 pin					
				IPNC	INT_{NC} pin					
Pin	CE pin edge detection register	02H	R & RES	0	Always set to 0.			0	–	–
				0						
				0						
				CEEDET	Detects the input of a rising edge to the CE pin.	Does not detect the input.	Detects the input.			
	CE pin level judge register	07H	R	0	Always set to 0.			0	–	–
				0						
				0						
				CE	Detects the status of the CE pin.	Low level	High level			
	Clock-stop release enable register	20H	R/W	0	Always set to 0.			0	H	H
				0						
				0						
				RLSEN	Selects whether to release the clock-stop by the $P1B_2$ pin.	Does not release.	Releases.			
	$P1B_2$ pin edge detection register	34H	R & RES	0	Always set to 0.			0	–	–
				0						
				0						
				P1B2EDET	Detects the input of a rising edge to the $P1B_2$ pin.	Does not detect the input.	Detects the input.			
PLL frequency synthesizer	PLL reference clock select register	13H	R/W	PLLRFCK3	Sets the PLL reference frequency.	2: 5 kHz 3: 10 kHz 4: 6.25 kHz 5: 12.5 kHz 6: 25 kHz F: Operation stop (disable status) 0, 1, 7 to E: Setting inhibited		F	F	H
				PLLRFCK2						
				PLLRFCK1						
				PLLRFCK0						

Remark H: Holds the previous state.

Table 8-1 Peripheral Hardware Control Functions of the Control Registers (5/7)

Peripheral hardware	Register	Address	Read/Write	b3 b2 b1 b0 Symbol	Function overview	Set value 0	Set value 1	Upon reset: Power on	STOP	CE
PLL frequency synthesizer	PLL-unlock-flip-flop judge register	22H	R & RES	0	Always set to 0.			U	H	H
				0						
				0						
				PLLUL	Detects the status of the unlock flip-flop.	Lock status	Unlock status			
	PLL-unlock-flip-flop sensibility select register	32H	R/W	0	Always set to 0.			0	0	H
				0						
				PLULSEN1	Specifies a delay for setting the unlock flip-flop.	0 1.25 0 3.5	0 0.25 1 Disable			
				PLULSEN0		0–1.5 μs 1–3.75 μs	1–0.5 μs 1 Disable			
A/D converter	A/D converter channel select register	21H	R/W	0	Always set to 0.			0	0	0
				ADCCH2	Selects the pins to be used as the A/D converter.	0 : ADC_0 2 : $P0D_1/ADC_2/XT_{IN}$ 4 : $P0D_3/ADC_4$ 6 : $P0C_1/ADC_6$	1 : $P0D_0/ADC_1/XT_{OUT}$ 3 : $P0D_2/ADC_3$ 5 : $P0D_0/ADC_5$ 6 : $P0D_2/ADC_7$			
				ADCCH1						
				ADCCH0						
	A/D converter control register	24H	R/W	ADCEN	Sets the operation of the A/D converter.	Stop	Start	0	0	0
				0	Always set to 0.					
				0						
			R	ADCCMP	Detects the result of comparison.	$V_{ADCIN} < V_{REF}$	$V_{ADCIN} > V_{REF}$	U	H	H
General-purpose port	Port 2D bit I/O select register	26H	R/W	0	Always set to 0.			0	0	0
				P2DBIO2	$P2D_2$ pin: Sets the pin for input or output (bit I/O).	Input	Output			
				P2DBIO1	$P2D_1$ pin					
				P2DBIO0	$P2D_0$ pin					
	Port 1C group I/O select register	27H	R/W	0	Always set to 0.			0	0	0
				0						
				0						
				P1CGIO	Sets port 1C for input or output (group I/O).	Input	Output			
	Port 1B bit I/O select register	35H	R/W	P1BBIO3	$P1B_3$ pin: Sets the pin for input or output (bit I/O).	Input	Output	0	0	0
				P1BBIO2	$P1B_2$ pin					
				P1BBIO1	$P1B_1$ pin					
				P1BBIO0	$P1B_0$ pin					
	Port 0B bit I/O select register	36H	R/W	P0BBIO3	$P0B_3$ pin					
				P0BBIO2	$P0B_2$ pin					
				P0BBIO1	$P0B_1$ pin					
				P0BBIO0	$P0B_0$ pin					
	Port 0A bit I/O select register	37H	R/W	P0ABIO3	$P0A_3$ pin					
				P0ABIO2	$P0A_2$ pin					
				P0ABIO1	$P0A_1$ pin					
				P0ABIO0	$P0A_0$ pin					

Remark H: Holds the previous state.
U: Undefined

Table 8-1 Peripheral Hardware Control Functions of the Control Registers (6/7)

Peripheral hardware	Register	Address	Read/Write	Symbol (b3 / b2 / b1 / b0)	Function overview	Set value 0	Set value 1	Upon reset: Power on	Upon reset: STOP	Upon reset: CE
D/A converter	PWM mode select register 3	03H	R/W	0	Always set to 0.			0	0	U
				0						
				0						
				PWM8SEL	$P2A_0/PWM_8$ pin — Selects whether to set these pins as a D/A converter.	General-purpose output port	DA converter			
	PWM mode select register 2	04H	R/W	PWM7SEL	$P2B_3/PWM_7$ pin					
				PWM6SEL	$P2B_2/PWM_6$ pin					
				PWM5SEL	$P2B_1/PWM_5$ pin					
				PWM4SEL	$P2B_0/PWM_4$ pin					
	PWM mode select register 1	05H	R/W	PWM3SEL	$P2C_3/PWM_3$ pin					
				PWM2SEL	$P2C_2/PWM_2$ pin					
				PWM1SEL	$P2C_1/PWM_1$ pin					
				PWM0SEL	$P2C_0/PWM_0$ pin					
Serial interface	Serial I/O 0 mode select register	08H	R/W	SIO0CH	Sets the number of communication wires.	Two-wire system	Three-wire system	0	0	0
				SB	Sets the communication method.	Serial I/O method	I^2C bus method (two-wire system only)			
				SIO0MS	Sets master or slave operation.	Master operation	Slave operation			
				SIO0TX	Sets the direction of transfer.	Reception	Transmission			
	Serial I/O 0 wait control register	18H	R/W	SBACK	Sets and detects the acknowledgment (I^2C bus method).	Sets and detects 0 or 1.		0	0	0
				SIO0NWT	Selects whether to enable a wait.	Enables.	Releases.			
				SIO0WRQ1 / SIO0WRQ0	Sets a wait mode.	0 0: No wait. 0 1: Data wait.	1 0: Acknowledge wait. 1 1: Address wait.			
	Serial I/O 1 mode select register	1CH	R/W	SIO1TS	Selects whether to start or stop the operation.	Stops the operation.	Starts the operation	0	0	0
				SIO1HIZ	Sets the status of the $P2D_1/SO_1$ pin.	General-purpose I/O port	Serial data output pin			
				SIO1CK1 / SIO1CK0	Sets the I/O clock.	0 0: External clock 0 1: 100 kHz	1 0: 500 kHz 1 1: 1 kHz			
	Serial I/O 0 status judge register	28H	R	SIO0SF8	Detects the contents of the clock counter.	0 or 1	8	0	0	0
				SIO0SF9		0 or 1	9			
				SBSTT	Detects the number of clocks (I^2C bus method).	Up to nine clocks are set, beginning with the start condition.				
				SBBSY	Detects the starting conditions (I^2C bus method).	All clocks are set, from the start condition up to the stop condition.				
	Serial I/O 0 interrupt mode register	38H	R/W	0	Always set to 0.			U	H	H
				0						
				SIO0IMD1 / SIO0IMD0	Sets the interrupt condition of serial interface 0.	0 0: Seventh clock 0 1: Eighth clock	1 0: 7th clock after the start condition 1 1: Stop condition			
	Serial I/O 0 clock select register	39H	R/W	0	Always set to 0.			U	H	H
				0						
				SIO0CK1 / SIO0CK0	Sets the internal clock of serial interface 0.	0 0: 100 kHz 0 1: 50 kHz	1 0: 500 kHz 1 1: 1 kHz			

Remark H: Holds the previous state.
U: Undefined

Table 8-1 Peripheral Hardware Control Functions of the Control Registers (7/7)

Peripheral hardware	Register	Address	Read/Write	b3 / b2 / b1 / b0 Symbol	Function overview	Set value 0	Set value 1	Upon reset: Power on	Upon reset: STOP	Upon reset: CE
Horizontal synchronizing signal counter	HSYNC-counter -gate control register	11H	R/W	0	Always set to 0.			0	0	0
				0						
				HSCGT1	Controls the gate of the HSYNC counter.	0 0 Gate closed; 0 1 Gate open	1 0 1.69 ms open; 1 1 Setting inhibited			
				HSCGT0						
	HSYNC-counter-gate judge register	12H	R	HSCGOSTT	Detects whether the gate of the HSYNC counter is open or closed.	Closed	Open	0	–	–
				0	Always set to 0.					
				0						
				0						
IDC	IDC back-ground select register	30H	R/W	IDCBKEN	Specifies the screen background.	Does not display the screen background color.	Displays the screen background color.	0	0	0
				IDCBKR	Background color R (Eight colors)	0: Black, 1: Blue, 2: Green, 3: Cyan, 4: Red, 5: Magenta, 6: Yellow, 7: White				
				IDCBKG	Background color G					
				IDCBKB	Background color B					
	IDC enable register	31H	R/W	0	Always set to 0.			0	0	0
				0						
				0						
				IDCEN	Turns the IDC display on or off.	Display on	Display off			
	IDC mode select register	33H	R/W	VRAMSEL	Turns VRAM on or off.	VRAM off	VRAM on	0	0	0
				IDCISEL	Selects the function of the P0B2/I pin.	General-purpose port	I pin			
				IDCD14SEL	Sets the number of dots in the vertical direction for the character to be displayed.	16 dots	14 dots			
				IDCCPCH	Sets the display interval between characters.	No interval	Interval of 2 dots			

8.4 NOTES ON USING THE REGISTER FILE

When operating a write-only (W) register, read-only (R) register, or unused register of the control registers (address locations 00H to 3FH of the register file), note (1), (2), and (3) below:

(1) If reading from a write-only register is attempted, an undefined value is read.
(2) If writing to a read-only register is attempted, nothing changes.
(3) If an attempt is made to read the contents of an unused part, an undefined value will be read. If writing to an unused part is attempted, nothing changes.

9. DATA BUFFER (DBF)

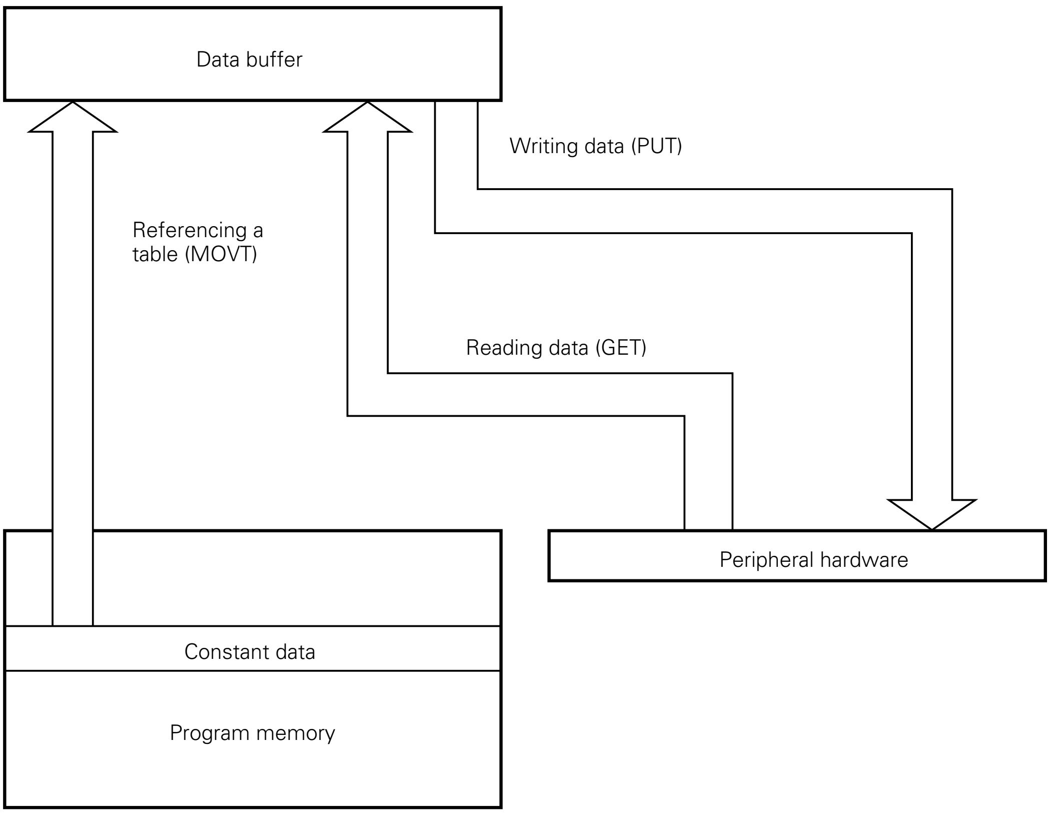
9.1 OVERVIEW

Fig. 9-1 shows an overview of the data buffer.

The data buffer is configured in data memory. It provides the following two functions:

(1) Function to read constant data related to program memory (refer to the table)
(2) Function to transfer data to or from the peripheral hardware

Fig. 9-1 Overview of the Data Buffer



9.2 DATA BUFFER MAIN BODY

9.2.1 Configuration of the Data Buffer Main Body

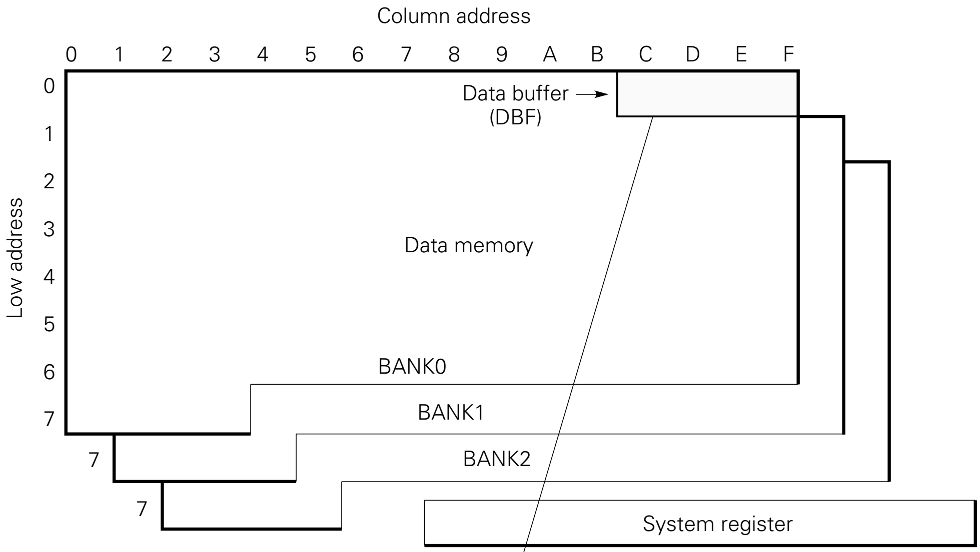
Fig. 9-2 shows the configuration of the data buffer.

As shown in Fig. 9-2, the data buffer consists of 16 bits of addresses 0CH to 0FH of BANK0 in data memory.

The most significant bit (MSB) of the 16-bit data is bit 3 at address 0CH. The least significant bit (LSB) is bit 0 at address 0FH.

The data buffer is configured in data memory and can thus be manipulated by any data memory manipulation instruction.

Fig. 9-2 Configuration of the Data Buffer



Data memory	Address	0CH				0DH				0ED				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 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	MSB								Data							LSB

9.2.2 Instruction to Reference a Table (MOVT DBF, @AR)

The MOVT DBF, @AR instruction functions as described below:

When an instruction to reference a table is executed, a stack of a single level is used.

All program memory addresses, 0000H to 2FFFH, allow table reference.

MOVT DBF, @AR

Data at the address specified by the address register is read from program memory and placed in the data buffer.

9.2.3 Instructions for Controlling the Peripheral Hardware (PUT, GET)

The PUT and GET instructions operate as described below:

(1) GET DBF, p

Data in the peripheral register at address p is read and written into the data buffer.

(2) PUT p, DBF

Data in the data buffer is set in the peripheral register at address p.

9.3 PERIPHERAL HARDWARE AND DATA BUFFER

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

[MEMO]

Table 9-1 Relationship between the Peripheral Hardware and Data Buffer (1/2)

Peripheral hardware		Peripheral register used to transfer data to or from the data buffer			
		Name	Symbol	Peripheral address	Instruction that can be used
Image display controller (IDC)	IDC	IDC start position setting register	IDCORG	01H	PUT/GET
	VRAM	VRAM pointer buffer	IDCVP	42H	GET
		VRAM pointer register	IDCVPR	43H	PUT/GET
A/D converter		A/D converter data register	ADCR	02H	PUT/GET
Serial interface	Serial interface 0 (SIO_0)	SIO0 shift register	SIO0SFR	03H	PUT/GET
	Serial interface 1 (SIO_1)	SIO1 shift register	SIO1SFR	07H	
Horizontal synchronizing signal counter		H_{SYNC} counter data register	HSC	04H	GET
Timer 1	Timer 1 modulo	Timer 1 modulo register	TM1M	05H	PUT/GET
	Timer 1 counter	Timer 1 counter	TM1C	06H	GET
D/A converter (PWM output)	$P2C_0/PWM_0$ pin	PWM data register 0	PWMR0	0CH	PUT/GET
	$P2C_1/PWM_1$ pin	PWM data register 1	PWMR1	0DH	
	$P2C_2/PWM_2$ pin	PWM data register 2	PWMR2	0EH	
	$P2C_3/PWM_3$ pin	PWM data register 3	PWMR3	0FH	
	$P2B_0/PWM_4$ pin	PWM data register 4	PWMR4	10H	
	$P2B_1/PWM_5$ pin	PWM data register 5	PWMR5	11H	
	$P2B_2/PWM_6$ pin	PWM data register 6	PWMR6	12H	
	$P2B_3/PWM_7$ pin	PWM data register 7	PWMR7	13H	
	$P2A_0/PWM_8$ pin	PWM data register 8	PWMR8	14H	
Watch timer	Seconds counter	Seconds setting register	WTMSEC	1AH	PUT/GET
	Minutes counter	Minutes setting register	WTMMIN	1BH	
	Hours counter	Hours setting register	WTMHR	1CH	
	Days counter	Days setting register	WTMDAY	1DH	
Address register (AR)		Address register	AR	40H	PUT/GET
PLL frequency synthesizer		PLL data register	PLLR	41H	PUT/GET
Timer 0	Timer 0 modulo	Timer 0 modulo register	TMOM	46H	PUT/GET
	Timer 0 counter	Timer 0 counter	TMOC	47H	GET

Table 9-1 Relationship between the Peripheral Hardware and Data Buffer (2/2)

Function		
Number of I/O bits of the data buffer	Number of bits actually used	Description
8	8	Sets the display start position of the image display controller.
16	10	Specifies a VRAM address.
16	10	Reads the value of the VRAM pointer.
8	6	Sets the reference voltage (VREF) of the A/D converter. $V_{REF} = \frac{x - 0.5}{64} \times V_{DD}$, $1 \leqq x \leqq 63$
8	8	Sets the serial out data and reads the serial in data.
8	6	Reads the value of the horizontal synchronizing signal counter.
8	8	Sets the reference data for timer 1.
8	8	Reads the count for timer 1.
8	8	Sets the duty cycle of the output signal of the D/A converter. Duty cycle : $D = \frac{x}{256} \times 100\%$, $0 \leqq x \leqq 255$ Frequency : f = 1.953 kHz
8	6	Writes and reads the data of the seconds counter, minutes counter, hours counter, and days counter.
	5	
	3	
16	14	Transfers data to or from the address register.
16	16	Sets N, by which the PLL frequency is divided.
16	12	Sets the reference data for timer 0.
16	12	Reads the data of the timer 0 counter.

9.4 NOTES ON USING THE DATA BUFFER

When transferring data, through the data buffer, to or from the peripheral hardware, note the following three points on unused peripheral addresses, write-only peripheral registers (PUT only), and read-only peripheral registers (GET only):

(1) If reading of a write-only register is attempted, an undefined value will be read.
(2) If writing to a read-only register is attempted, nothing changes.
(3) If an attempt is made to read the data at an unused address, an undefined value will be read. If writing to the unused address is attempted, nothing changes.

10. GENERAL-PURPOSE PORTS

A general-purpose port outputs high, low, and floating signals to external circuits and reads high and low signals from the external circuits.

10.1 OVERVIEW

Table 10-1 indicates the relationship between the ports and port registers.

General-purpose ports are classified into three types: I/O ports, input ports, and output ports.

I/O ports can be divided into two types: In a bit I/O port, each bit (each pin) can be set to input or output mode. In a group I/O port, the bits can set to input or output mode in units of four bits (four pins).

Table 10-1 Relationship between Ports (Pins) and Port Registers (1/2)

Port	Pin			Data setting method					
	No.	Symbol	I/O	Port register (data memory)					Remarks
				Bank	Address	Symbol	Bit symbol (reserved word)		
Port 0A	36	$P0A_3$	I/O (bit I/O)	BANK0	70H	P0A	b_3	P0A3	
	37	$P0A_2$					b_2	P0A2	
	38	$P0A_1$					b_1	P0A1	
	39	$P0A_0$					b_0	P0A0	
Port 0B	32	$P0B_3$	I/O (bit I/O)		71H	P0B	b_3	P0B3	
	33	$P0B_2$					b_2	P0B2	
	34	$P0B_1$					b_1	P0B1	
	35	$P0B_0$					b_0	P0B0	
Port 0C	48	$P0C_3$	Output		72H	P0C	b_3	P0C3	
	49	$P0C_2$					b_2	P0C2	
	50	$P0C_1$					b_1	P0C1	
	51	$P0C_0$					b_0	P0C0	
Port 0D	57	$P0D_3$	Input		73H	P0D	b_3	P0D3	
	58	$P0D_2$					b_2	P0D2	
	59	$P0D_1$					b_1	P0D1	
	60	$P0D_0$					b_0	P0D0	

Table 10-1 Relationship between Ports (Pins) and Port Registers (2/2)

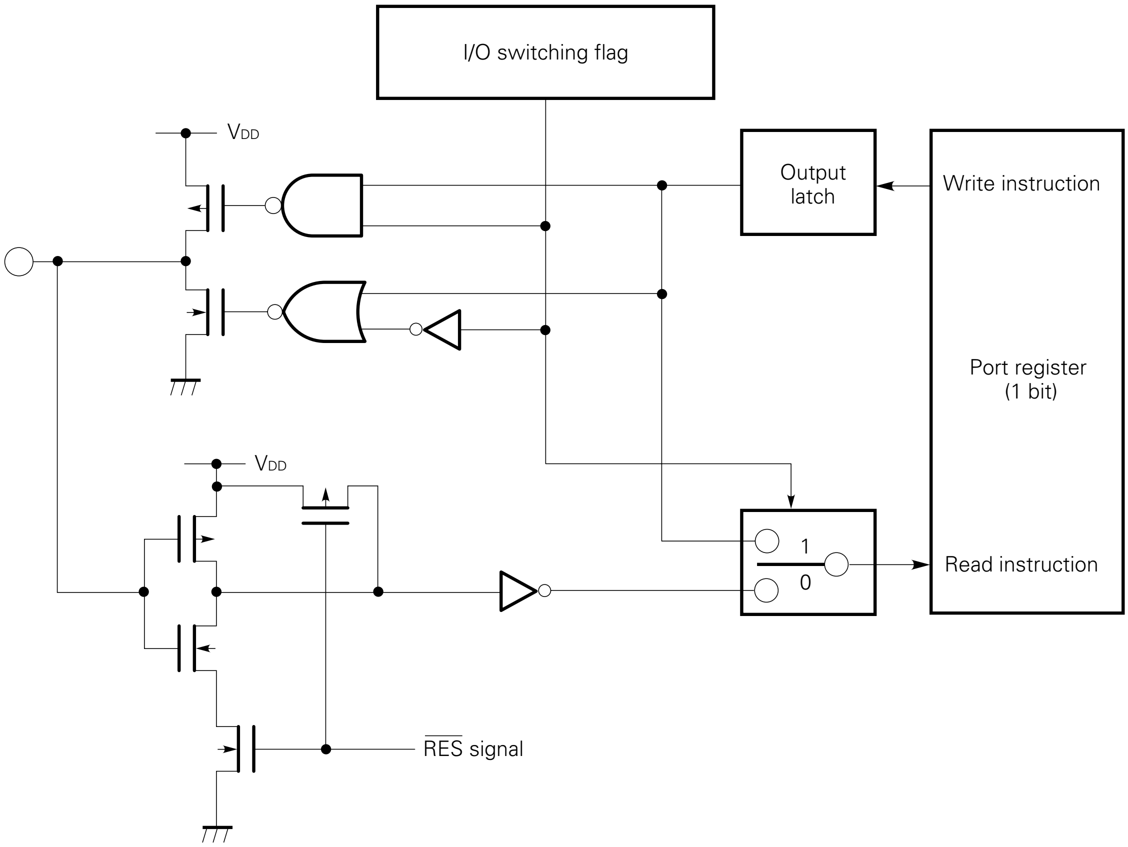
Port	Pin			Data setting method					
	No.	Symbol	I/O	Port register (data memory) Bank	Address	Symbol	Bit symbol (reserved word)		Remarks
Port 1A	15	$P1A_3$	Output	BANK1	70H	P1A	b_3	P1A3	
	16	$P1A_2$					b_2	P1A2	
	17	$P1A_1$					b_1	P1A1	
	18	$P1A_0$					b_0	P1A0	
Port 1B	11	$P1B_3$	I/O (bit I/O)		71H	P1B	b_3	P1B3	
	12	$P1B_2$					b_2	P1B2	
	13	$P1B_1$					b_1	P1B1	
	14	$P1B_0$					b_0	P1B0	
Port 1C	53	$P1C_3$	I/O (group I/O)		72H	P1C	b_3	P1C3	
	54	$P1C_2$					b_2	P1C2	
	55	$P1C_1$					b_1	P1C1	
	56	$P1C_0$					b_0	P1C0	
Port 0D	6	$P1D_3$	Output		73H	P1D	b_3	P1D3	
	7	$P1D_2$					b_2	P1D2	
	8	$P1D_1$					b_1	P1D1	
	9	$P1D_0$					b_0	P1D0	
Port 2A	No relevant pins			BANK2	70H	P2A	b_3	—	Always set to 0
							b_2	—	
							b_1	—	
	19	$P2A_0$	Output				b_0	P2A0	
Port 2B	40	$P2B_3$	Output		71H	P2B	b_3	P2B3	
	41	$P2B_2$					b_2	P2B2	
	42	$P2B_1$					b_1	P2B1	
	43	$P2B_0$					b_0	P2B0	
Port 2C	44	$P2C_3$	Output		72H	P2C	b_3	P2C3	
	45	$P2C_2$					b_2	P2C2	
	46	$P2C_1$					b_1	P2C1	
	47	$P2C_0$					b_0	P2C0	
Port 2D	No relevant pins				6FH	P2D	b_3	—	Always set to 0
	20	$P2D_2$	I/O (bit I/O)				b_2	P2D2	
	21	$P2D_1$					b_1	P2D1	
	22	$P2D_0$					b_0	P2D0	

10.2 GENERAL-PURPOSE I/O PORTS (P0A, P0B, P1B, P1C, P2D)

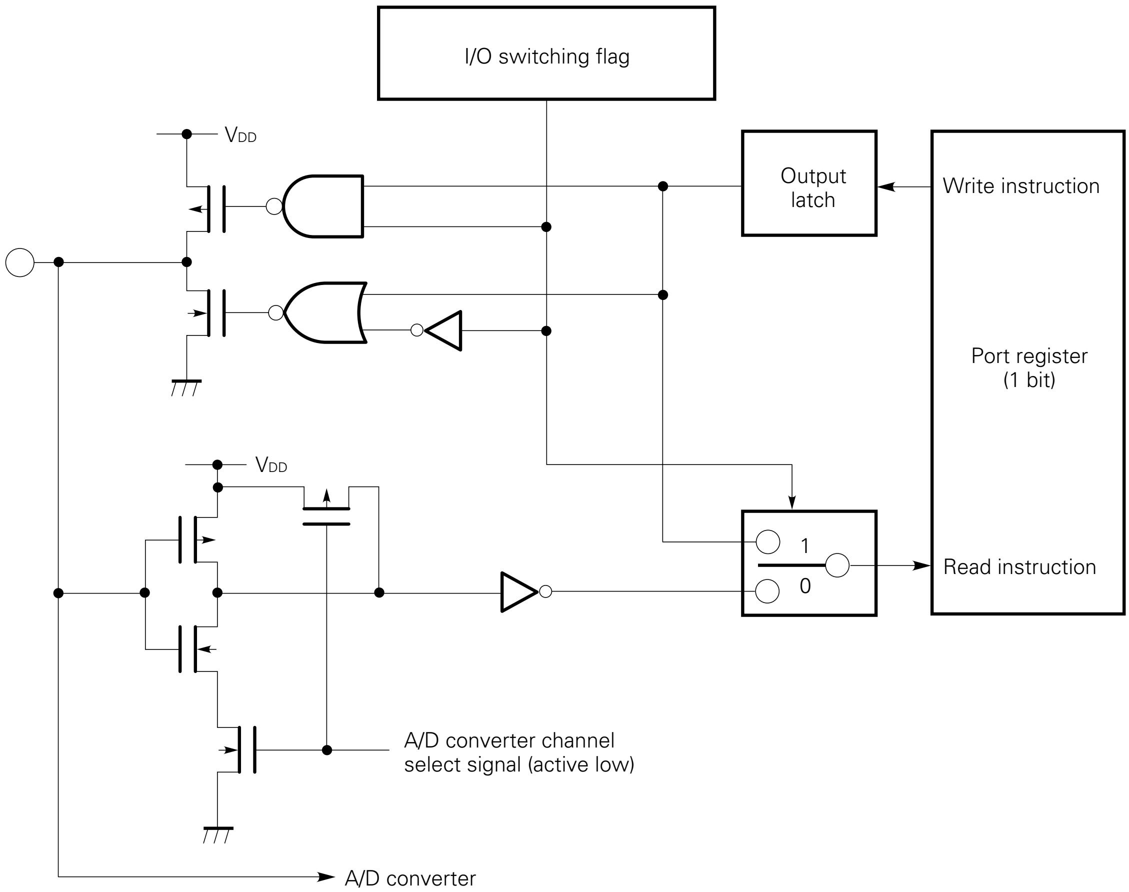
10.2.1 Configurations of the I/O ports

(1) to (5) below describe the configurations of the I/O ports:

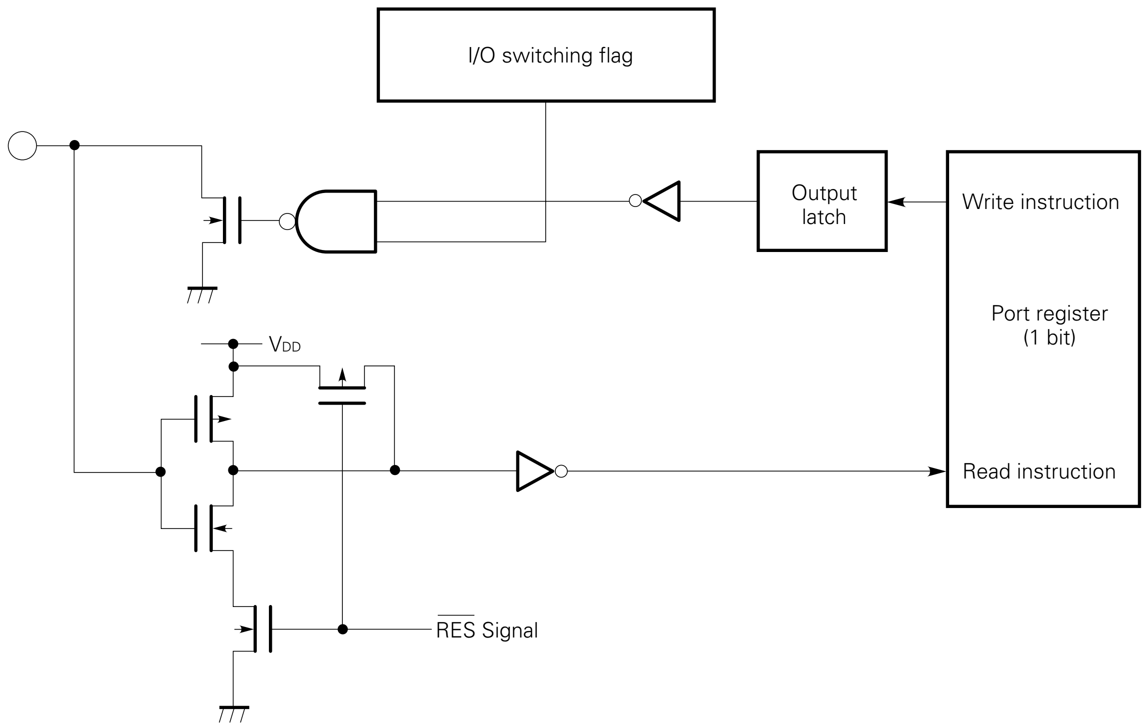
(1) P0A ($P0A_3$, $P0A_2$),
P0B ($P0B_2$, $P0B_0$),
P1B ($P1B_2$, $P1B_1$, $P1B_0$),
P2D ($P2D_2$, $P2D_1$, $P2D_0$)



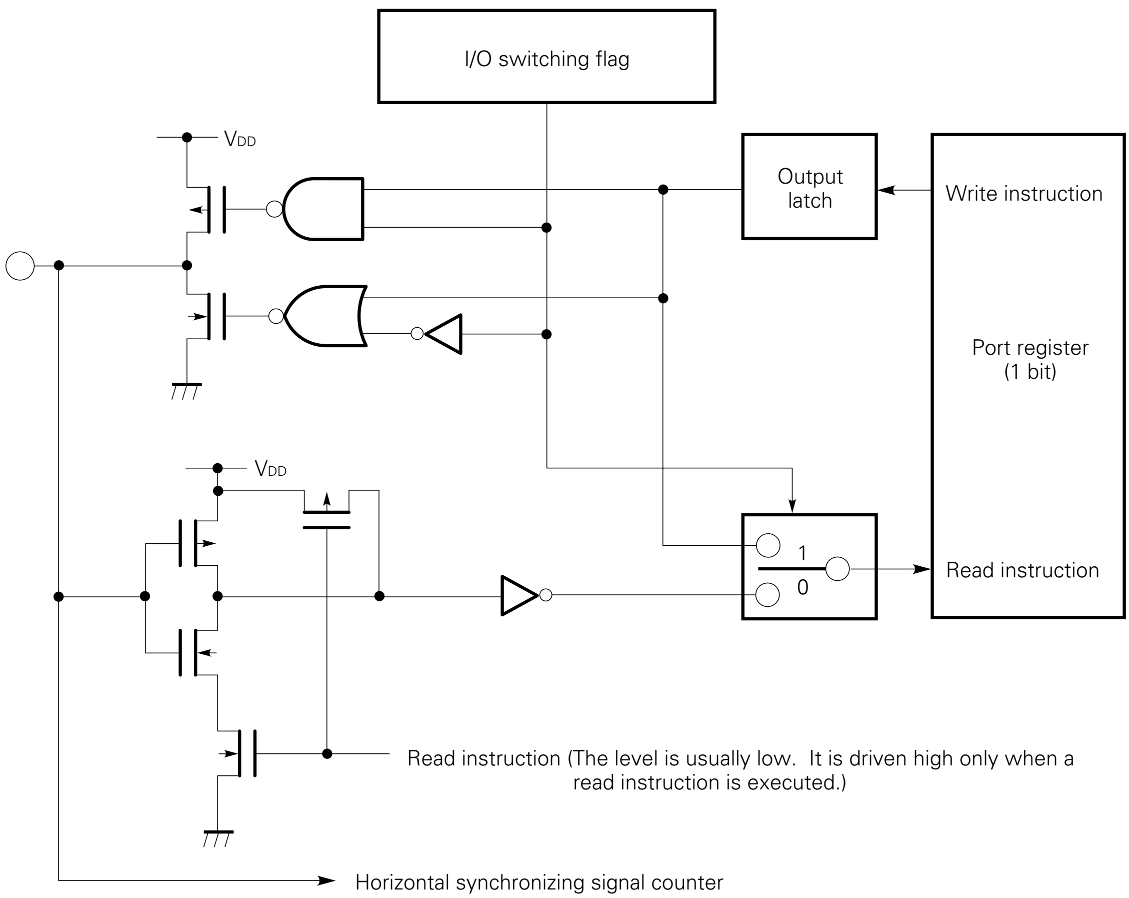
(2) P1C ($P1C_3$, $P1C_2/ADC_7$, $P1C_1/ADC_6$, $P1C_0/ADC_5$)



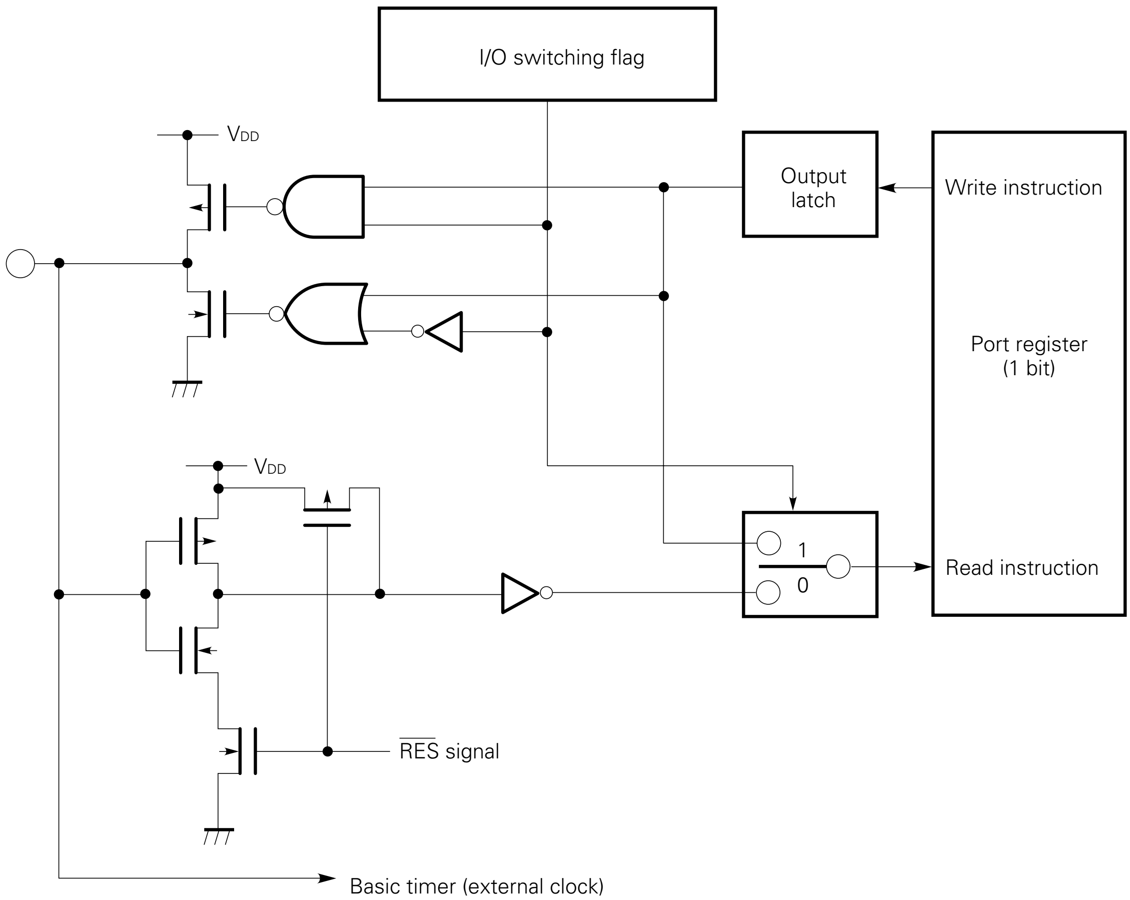
(3) P0A ($P0A_1$, $P0A_0$)



(4) $P0B_3$/HSCNT



(5) P1B3/TMIN



10.2.2 Using the I/O Port

The I/O select register of control register P0A, P0B, P1B, P1C, or P2D sets the I/O port to input or output mode.

P0A, P0B, P1B, and P2D are bit I/O ports, each bit of which (each pin) can be set to input or output mode.

To set the output data, write the data to the corresponding port register. To read the input data, execute an instruction to read the data.

Section 10.2.3 describes the configurations of the I/O select registers of the ports.

Sections 10.2.4 and **10.2.5** describe the use of the I/O port as an input port and/or output port.

Section 10.2.6 provides notes on using the I/O port.

10.2.3 Control Registers of the I/O Ports

Port 0A bit I/O select register, port 0B bit I/O select register, port 1B bit I/O select register, port 1C group I/O select register, and port 2D bit I/O select register set pins P0A, P0B, P1B, P1C, and P2D in input or output mode, respectively.

(1) to (5) below describe the configurations and functions of the control registers:

(1) Port 0A bit I/O select register

Register	Flag symbol b_3	Flag symbol b_2	Flag symbol b_1	Flag symbol b_0	Address	Read/write
Port 0A bit I/O select register	P0ABIO3	P0ABIO2	P0ABIO1	P0ABIO0	37H	R/W

P0ABIO0: Sets the port to input or output mode.

P0ABIO0	
0	Sets pin $P0A_0$ to input mode.
1	Sets pin $P0A_0$ to output mode.

P0ABIO1: Sets the port to input or output mode.

P0ABIO1	
0	Sets pin $P0A_1$ to input mode.
1	Sets pin $P0A_1$ to output mode.

P0ABIO2: Sets the port to input or output mode.

P0ABIO2	
0	Sets pin $P0A_2$ to input mode.
1	Sets pin $P0A_2$ to output mode.

P0ABIO3: Sets the port to input or output mode.

P0ABIO3	
0	Sets pin $P0A_3$ to input mode.
1	Sets pin $P0A_3$ to output mode.

Upon 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

(2) Port 0B bit I/O select register

Register	Flag symbol b3	Flag symbol b2	Flag symbol b1	Flag symbol b0	Address	Read/write
Port 0B bit I/O select register	P0BBIO3	P0BBIO2	P0BBIO1	P0BBIO0	36H	R/W

P0BBIO0	Sets the port to input or output mode.
0	Sets pin $P0B_0$ to input mode.
1	Sets pin $P0B_0$ to output mode.

P0BBIO1	Sets the port to input or output mode.
0	Sets pin $P0B_1$ to input mode.
1	Sets pin $P0B_1$ to output mode.

P0BBIO2	Sets the port to input or output mode.
0	Sets pin $P0B_2$ to input mode.
1	Sets pin $P0B_2$ to output mode.

P0BBIO3	Sets the port to input or output mode.
0	Sets pin $P0B_3$ to input mode.
1	Sets pin $P0B_3$ to output mode.

Upon reset		b3	b2	b1	b0
	Power-on	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

(3) Port 1B bit I/O select register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Port 1B bit I/O select register	P1BBIO3	P1BBIO2	P1BBIO1	P1BBIO0	35H	R/W

P1BBIO0: Sets the port to input or output mode.

0	Sets pin $P1B_0$ to input mode.
1	Sets pin $P1B_0$ to output mode.

P1BBIO1: Sets the port to input or output mode.

0	Sets pin $P1B_1$ to input mode.
1	Sets pin $P1B_1$ to output mode.

P1BBIO2: Sets the port to input or output mode.

0	Sets pin $P1B_2$ to input mode.
1	Sets pin $P1B_2$ to output mode.

P1BBIO3: Sets the port to input or output mode.

0	Sets pin $P1B_3$ to input mode.
1	Sets pin $P1B_3$ to output mode.

Upon reset		b3	b2	b1	b0
	Power-on	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

(4) Port 1C group I/O select register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Port 1C group I/O select register	0	0	0	P1CGIO	27H	R/W

b_0 (P1CGIO): Sets the port to input or output mode.

0	Sets pins $P1C_3$ to $P1C_0$ to input mode.
1	Sets pins $P1C_3$ to $P1C_0$ to output mode.

b_3 to b_1: Always set to 0.

Upon reset		b_3	b_2	b_1	b_0
	Power-on	0	0	0	0
	Clock stop	↓	↓	↓	0
	CE	↓	↓	↓	0

(5) Port 2D bit I/O select register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Port 2D bit I/O select register	0	P2DBIO2	P2DBIO1	P2DBIO0	26H	R/W

P2DBIO0: Sets the port to input or output mode.

P2DBIO0	
0	Sets pin $P2D_0$ to input mode.
1	Sets pin $P2D_0$ to output mode.

P2DBIO1: Sets the port to input or output mode.

P2DBIO1	
0	Sets pin $P2D_1$ to input mode.
1	Sets pin $P2D_1$ to output mode.

P2DBIO2: Sets the port to input or output mode.

P2DBIO2	
0	Sets pin $P2D_2$ to input mode.
1	Sets pin $P2D_2$ to output mode.

b_3: Always set to 0.

Upon reset		b_3	b_2	b_1	b_0
	Power-on	0	0	0	0
	Clock stop	↓	0	0	0
	CE	↓	0	0	0

10.2.4 Using an I/O Port as an Input Port

Select the pin to be set to input mode, using the I/O select register of each port.

The pins of port 1C can be set to input mode only in units of four bits.

The specified input pin enters the floating (Hi-Z) status and waits for the input of an external signal.

To read the input data, execute a read instruction (such as SKT) for the port register corresponding to the pin.

If the signal input to the pin is high, 1 is read from the corresponding port register. If the input signal is low, 0 is read from the port register.

If a write instruction (such as MOV) is executed for the port register corresponding to an input port, the contents of the output latch are rewritten.

10.2.5 Using an I/O Port as an Output Port

Select the pin to be set to output mode, using the I/O select register of each port.

The pins of port 1C can be set in the output mode only in units of four bits.

The specified output pin outputs the contents of the output latch.

To set the output data, execute a write instruction (such as MOV) for the port register corresponding to the pin.

To output a high signal to a pin, write 1. To output a low signal, write 0.

To set a port to the floating state, set the port to input mode.

If a read instruction (such as SKT) is executed for the port register corresponding to an output port, the contents of the output latch are read.

For the $P0A_0$ and $P0A_1$ pins, the status of the pin is read as is. The contents of the output latch and the read data may differ (see **Section 10.2.6**).

10.2.6 Notes on Using the I/O Port

If the $P0A_0$ and $P0A_1$ pins are used for output as described below, the contents of the output latches may be rewritten.

Example Setting the $P0A_0$ and $P0A_1$ pins as output ports

```
INITFLG NOT P0ABIO3, NOT P0ABIO2, P0ABIO1, P0ABIO0
                    ; Sets the P0A1 and P0A0 pins to output mode.
INITFLG NOT P0A3, NOT P0A2, POA1, POA0
                    ; Outputs a high signal to the P0A1 and P0A0 pins.
; ①
CLR1    P0A1        ; Outputs a low signal to the P0A1 pin.
Macro expansion
AND     . MF. P0A1 SHR 4, #. DF. (NOT P0A1 AND 0FH)
```

If the signal on pin $P0A_0$ is driven low by the execution of instruction ① above, the CLR1 instruction rewrites the contents of the output latch of pin $P0A_0$ to 0.

If an instruction to read, the contents of port register P0A are executed when the $P0A_0$ or $P0A_1$ pin is set to output mode, the contents of the output latch are rewritten to the current signal level of the pin, even though the actual contents of the output latch are not changed.

10.2.7 Statuses of the I/O Ports upon Reset

(1) At power-on reset

All pins are set to input mode.
The contents of the output latches are set to 0.

(2) At CE reset

All pins are set to input mode.
The contents of the output latches are retained.

(3) At a clock-stop

All pins are set to input mode.
The contents of the output latches are retained.

(4) In the halt state

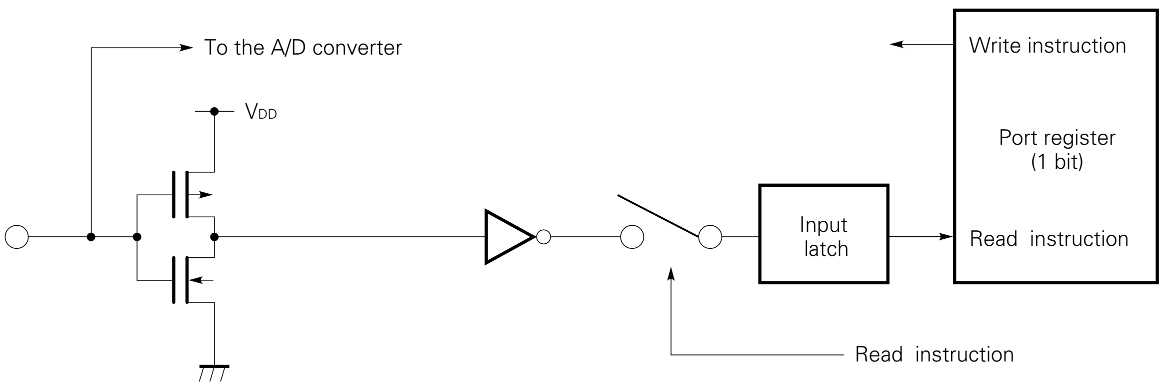
The previous statuses are retained.

10.3 GENERAL-PURPOSE INPUT PORT (P0D)

10.3.1 Configuration of the Input Port

The configuration of the input port is shown below:

- **P0D ($P0D_3$ to $P0D_0$)**



10.3.2 Using the Input Port

To read the input data, execute an instruction to read the contents of port register P0D (such as SKT).

If the signal input to a pin is high, 1 is read from the corresponding port register. If the input signal is low, 0 is read from the port register.

If a write instruction (such as MOV) is executed for a port register, nothing changes.

10.3.3 Notes on Using the Input Port

P0D is internally pulled down if it is used as a general-purpose port.

10.3.4 Statuses of the Input Port upon Reset

(1) At power-on reset

All pins are set to input mode.

(2) At CE reset

All pins are set to input mode.

(3) At a clock-stop

All pins are set to input mode.
They are internally pulled down.

(4) In the halt state

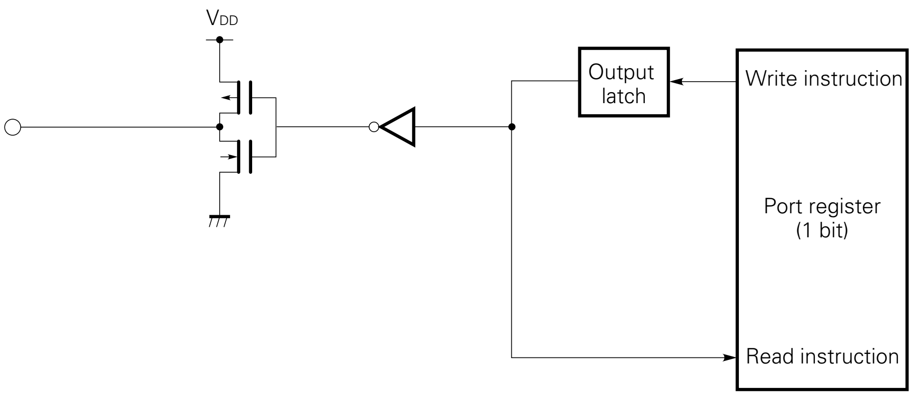
The previous statuses are retained.

10.4 GENERAL-PURPOSE OUTPUT PORTS (P0C, P1A, P1D, P2A, P2B, P2C)

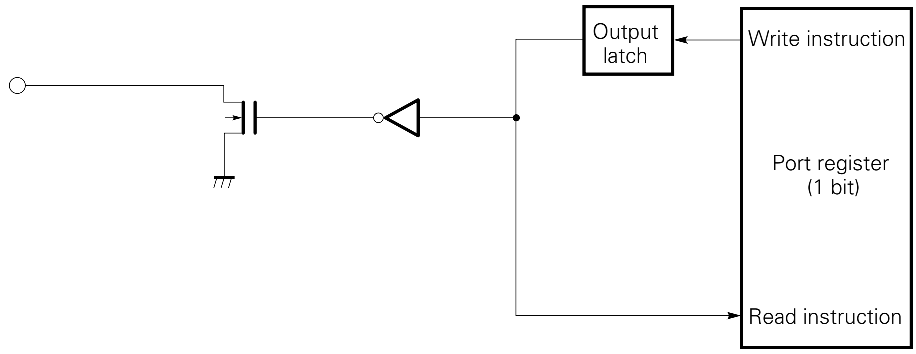
10.4.1 Configurations of the Output Ports

The configurations of the output ports are shown in (1) and (2) below:

**(1) P0C ($P0C_3$, $P0C_2$, $P0C_1$, $P0C_0$),
P1D ($P1D_3$, $P1D_2$, $P1D_1$, $P1D_0$)**



(2) P1A ($P1A_3$, $P1A_2$, $P1A_1$, $P1A_0$),
P2A ($P2A_0$),
P2B ($P2B_3$, $P2B_2$, $P2B_1$, $P2B_0$),
P2C ($P2C_3$, $P2C_2$, $P2C_1$, $P2C_0$)



10.4.2 Using the Output Port

The output port outputs the contents of the output latch from each pin.

To set the output data, execute a write instruction (such as MOV) for the port register corresponding to each pin.

To output a high signal to a pin, write 1. To output a low signal, write 0.

The pins of P1A, P2A, P2B, and P2C are N-ch open-drain output. The pins enter the floating status if a high signal is output.

If a read instruction (such as SKT) is executed for a port register, the contents of the output latch are read.

10.4.3 Statuses of the Output Port upon Reset

(1) At power-on reset

The contents of the output latch are output.

The contents of the output latch are undefined. If required, initialize them by a program before setting a pin to output mode.

(2) Upon CE reset

The contents of the output latch are output.

The output latch retains the data existing immediately before the reset. If a pin is directly set to output mode, the previous contents are output.

(3) Upon a clock stop

The contents of the output latch are output.

The output latch retains the last data existing immediately before the reset. If a pin is directly set to output mode, the previous contents are output.

(4) In the halt state

The previous statuses are retained.

11. INTERRUPT

11.1 OUTLINE OF THE INTERRUPT BLOCK

Fig. 11.1 is an outline of the interrupt block.

As shown in the figure, when an interrupt is requested by peripheral hardware, the interrupt block suspends the program currently being executed and causes a branch to a vector address.

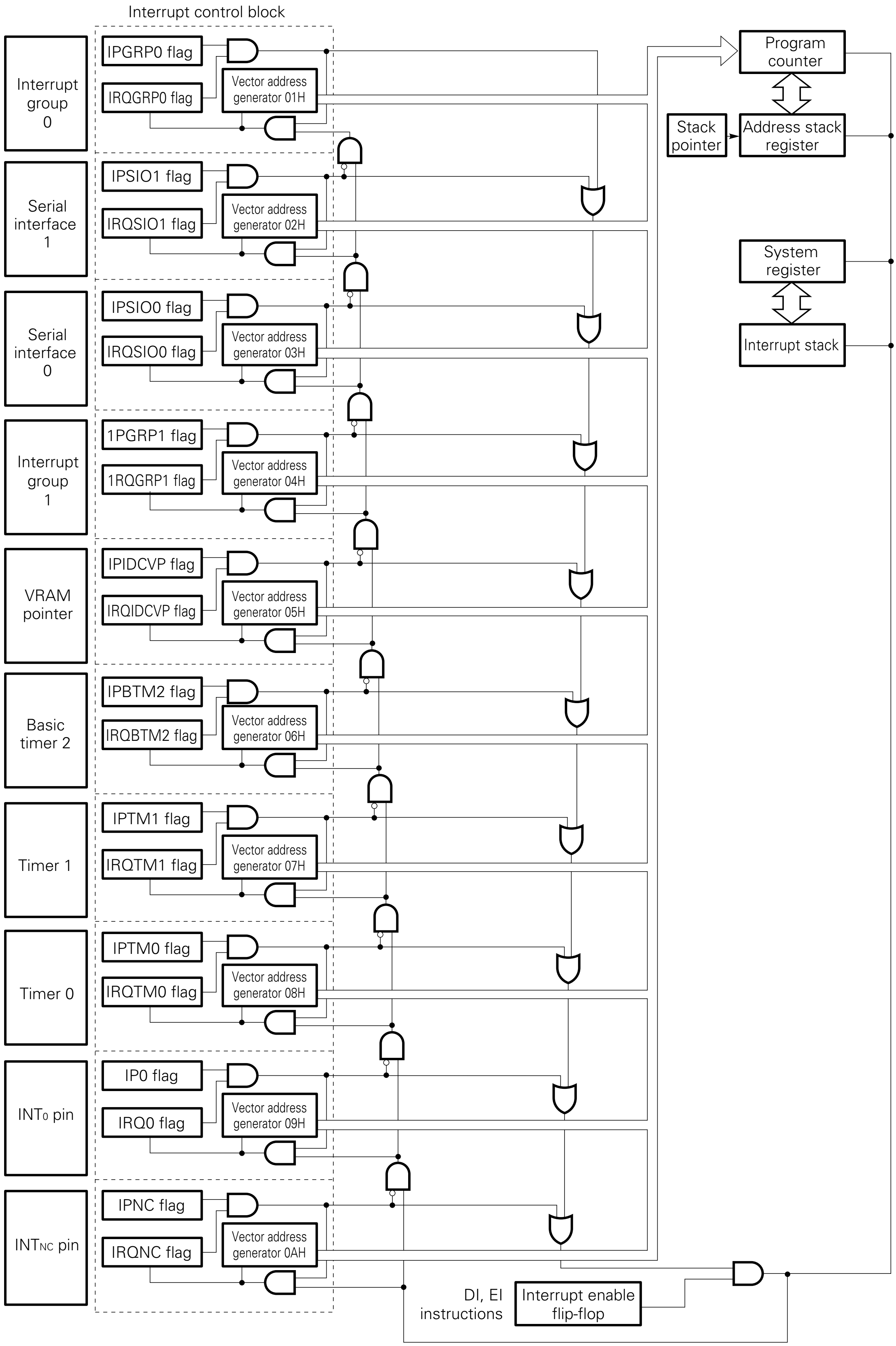
The interrupt block contains interrupt control blocks, provided for each item of peripheral hardware, and an interrupt enable flip-flop that enables all interrupts. The interrupt block also contains a stack pointer, an address stack register, a program counter, and an interrupt stack, all of which are controlled when an interrupt is accepted.

The interrupt control block for each item of peripheral hardware consists of an interrupt request flag (IRQ×××) that detects an interrupt request, an interrupt enable flag (IP×××) that enables the interrupt, and a vector address generator (VAG) that specifies a vector address when the interrupt is accepted.

The following lists the peripheral hardware that supports the interrupt function:

- INT_0 pin
- INT_{NC} pin
- Timer 0
- Timer 1
- Basic timer 2
- VRAM pointer
- Interrupt group 0 (timer 0 overflow)
- Interrupt group 1 ($\overline{V_{SYNC}}$ or $\overline{H_{SYNC}}$ signal)
- Serial interface 0
- Serial interface 1

Fig. 11-1 Schematic Diagram of Interrupt Block



11.2 INTERRUPT CONTROL BLOCKS

An interrupt control block is provided for each item of peripheral hardware. The block indicates whether an interrupt has been requested by the associated peripheral hardware, enables the interrupt, and generates a vector address when the interrupt is accepted.

11.2.1 Formats and Functions of Interrupt Request Flags (IRQ×××)

When an interrupt request is received from an item of peripheral hardware, the corresponding interrupt request flag is set to 1. It is reset to 0 once the interrupt has been accepted.

When "1" is written into an interrupt request flag, via the window register, the effect is the same as when the corresponding interrupt request is generated.

When interrupts are not enabled, for example, the interrupt request state can be detected by reading these interrupt request flags.

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

When more than one interrupt request occurs at any one time, and one of these interrupt requests is accepted, the interrupt request flags for the other interrupt requests are not reset.

The interrupt request flags are set in interrupt request registers in the register file.

Figs. 11-2 to 11-11 illustrate the formats and functions of the interrupt request registers.

Fig. 11-2 Format of Interrupt Request Register 1

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Interrupt request register 1	INTNC	0	0	IRQNC	3FH	R/W R for bit 3 only

IRQNC (b0): Sets the interrupt request state for the INT_{NC} pin.

0	Interrupt not requested
1	Interrupt requested

b2, b1: Fixed to 0

INTNC (b3): Detects the input level on the INT_{NC} pin.

0	Low Level
1	High Level

Upon reset	b_3	b_2	b_1	b_0
Power-on	0	0	0	0
Clock stop	0	↓	↓	0
CE	0	↓	↓	0

Fig. 11-3 Format of Interrupt Request Register 2

Register	Flag symbol				Address	Read/write
	b3	b2	b1	b0		
Interrupt request register 2	INT0	0	0	IRQ0	3EH	R/W R for bit 3 only

IRQ0 (b0): Sets the interrupt request state for the INT_0 pin.

0	Interrupt not requested
1	Interrupt requested

b2, b1: Fixed to 0

INT0 (b3): Detects the input level on the INT_0 pin.

0	Low level
1	High level

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	0	↓	↓	0
CE	0	↓	↓	0

Fig. 11-4 Format of Interrupt Request Register 3

Register	Flag symbol				Address	Read/write
	b3	b2	b1	b0		
Interrupt request register 3	0	0	0	IRQTM0	3DH	R/W

IRQTM0 (b0): Sets the timer 0 interrupt request state.

0	Interrupt not requested
1	Interrupt requested

b3, b2, b1: Fixed to 0

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	0
CE	↓	↓	↓	0

Fig. 11-5 Format of Interrupt Request Register 4

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Interrupt request register 4	0	0	0	IRQTM1	3CH	R/W

IRQTM1: Sets the timer 1 interrupt request state.

0	Interrupt not requested
1	Interrupt requested

b3 to b1: Fixed to 0

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	0
CE	↓	↓	↓	0

Fig. 11-6 Format of Interrupt Request Register 5

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Interrupt request register 5	0	0	0	IRQBTM2	3BH	R/W

IRQBTM2: Sets the basic timer 2 interrupt request state.

0	Interrupt not requested
1	Interrupt requested

b3 to b1: Fixed to 0

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	0
CE	↓	↓	↓	0

Fig. 11-7 Format of Interrupt Request Register 6

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Interrupt request register 6	0	0	0	IRQDCVP	3AH	R/W

IRQDCVP: Sets the VRAM pointer interrupt request state.

IRQDCVP	Sets the VRAM pointer interrupt request state.
0	Interrupt not requested
1	Interrupt requested

b3, b2, b1: Fixed to 0

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	0
CE	↓	↓	↓	0

Fig. 11-8 Format of Interrupt Request Register 7

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Interrupt request register 7	INTGRP1	0	0	IRQGRP1	2CH	R/W R for bit 3 only

IRQGRP1	Sets the interrupt group 1 interrupt request state.
0	Interrupt not requested
1	Interrupt requested

b2, b1: Fixed to 0

INTGRP1	Detects the input level of the V_{SYNC} or H_{SYNC} signal.
0	Low Level
1	High Level

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	0	↓	↓	0
CE	0	↓	↓	0

Fig. 11-9 Format of Interrupt Request Register 8

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Interrupt request register 8	0	0	0	IRQSIO0	2BH	R/W

IRQSIO0: Sets the serial interface 0 interrupt request state.

IRQSIO0	
0	Interrupt not requested
1	Interrupt requested

b3, b2, b1: Fixed to 0

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	0
CE	↓	↓	↓	0

Fig. 11-10 Format of Interrupt Request Register 9

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Interrupt request register 9	0	0	0	IRQSIO1	2AH	R/W

IRQSIO1: Sets the serial interface 1 interrupt request state.

IRQSIO1	
0	Interrupt not requested
1	Interrupt requested

b3, b2, b1: Fixed to 0

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	0
CE	↓	↓	↓	0

Fig. 11-11 Format of Interrupt Request Register 10

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Interrupt request register 10	0	0	0	IRQGRP0	29H	R/W

	Sets the interrupt group 0 interrupt request state.
0	Interrupt not requested
1	Interrupt requested

b_3, b_2, b_1: Fixed to 0

Upon reset		b_3	b_2	b_1	b_0
	Power-on	0	0	0	0
	Clock stop	↓	↓	↓	0
	CE	↓	↓	↓	0

11.2.2 Interrupt Enable Flags (IP×××)

The interrupt enable flags enable the interrupts requested from the corresponding peripheral hardware. An interrupt can be accepted only when all of the following conditions are satisfied:

- The interrupt is enabled by the setting of the corresponding interrupt enable flag.
- The corresponding interrupt request flag indicates that an interrupt request has occurred.
- The EI instruction (enabling all interrupts) has been executed.

The interrupt enable flags are arranged in interrupt enable registers on the register file. Figs. 11-12 to 11-14 show the formats and functions of the interrupt enable registers.

Fig. 11-12 Format of Interrupt Enable Register 1

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Interrupt enable register 1	IPTM1	IPTM0	IP0	IPNC	2FH	R/W

IPNC: Specifies whether to enable interrupts for the INT_{NC} pin.

0	Disables interrupts.
1	Enables interrupts.

IP0: Specifies whether to enable interrupts for the INT_0 pin.

0	Disables interrupts.
1	Enables interrupts.

IPTM0: Specifies whether to enable interrupts for timer 0.

0	Disables interrupts.
1	Enables interrupts.

IPTM1: Specifies whether to enable interrupts for timer 1.

0	Disables interrupts.
1	Enables interrupts.

Upon 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

Fig. 11-13 Format of Interrupt Enable Register 2

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Interrupt enable register 2	IPSIO0	IPGRP1	IPIDCVP	IPBTM2	2EH	R/W

IPBTM2: Specifies whether to enable interrupts for basic timer 2.

IPBTM2	
0	Disables interrupts.
1	Enables interrupts.

IPIDCVP: Specifies whether to enable interrupts for the VRAM pointer.

IPIDCVP	
0	Disables interrupts.
1	Enables interrupts.

IPGRP1: Specifies whether to enable interrupts for interrupt group 1.

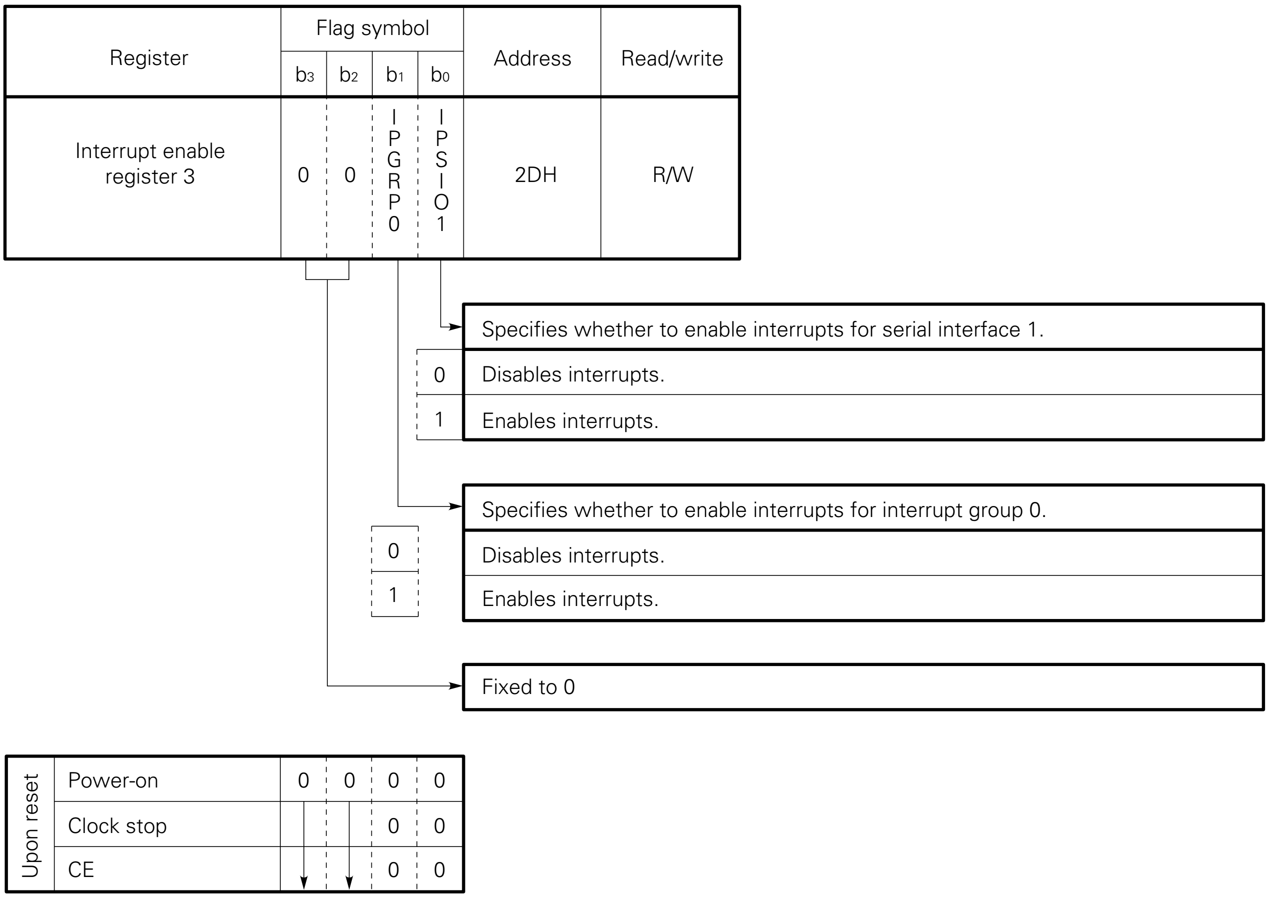
IPGRP1	
0	Disables interrupts.
1	Enables interrupts.

IPSIO0: Specifies whether to enable interrupts for serial interface 0.

IPSIO0	
0	Disables interrupts.
1	Enables interrupts.

Upon reset		b3	b2	b1	b0
	Power-on	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

Fig. 11-14 Format of Interrupt Enable Register 3



11.2.3 Vector Address Generator (VAG)

When an interrupt requested from an item of peripheral hardware has been accepted, the vector address generator generates the branch address (vector address) of a program memory location for the accepted interrupt source.

Table 11-1 lists the vector addresses generated for different interrupt sources.

Table 11-1 Vector Addresses for Different Interrupt Sources

Interrupt source	Vector address
INT_{NC} pin	000AH
INT_0 pin	0009H
Timer 0	0008H
Timer 1	0007H
Basic timer 2	0006H
VRAM pointer	0005H
Interrupt group 1 ($\overline{V_{SYNC}}$ or $\overline{H_{SYNC}}$ pin)	0004H
Serial interface 0	0003H
Serial interface 1	0002H
Interrupt group 0 (timer 0 overflow)	0001H

11.3 INTERRUPT STACK REGISTER

11.3.1 Format and Functions of the Interrupt Stack Register

Fig. 11-15 shows the format of the interrupt stack register.

When an interrupt is accepted, the contents of the following system registers are saved in the interrupt stack register:

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

When an interrupt is accepted, the contents of the above system registers are saved in the interrupt stack register. Then, the contents of all system registers, except the window register, are reset to 0.

Up to two levels of the system register contents can be saved in the interrupt stack register.

Therefore, up to two levels of interrupt are possible.

The contents of the system registers are restored from the interrupt stack register once the interrupt return instruction (RETI instruction) has been executed.

Fig. 11-15 Format of the Interrupt Stack Register

Interrupt stack register (INTSK)																				
Name	Window stack (WRSK)				Bank stack (BANKSK)				Register pointer stack, high (RPHSK)				Register pointer stack, low (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					–	–			–	–										

Remark – : Bit not saved

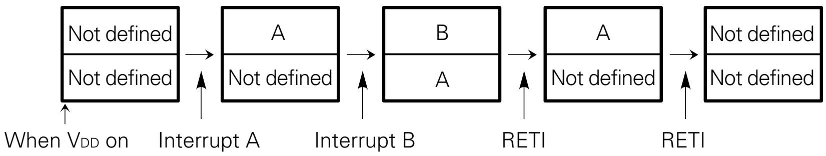
11.3.2 Interrupt Stack Operation

Fig. 11-16 illustrates the operation of the interrupt stack.

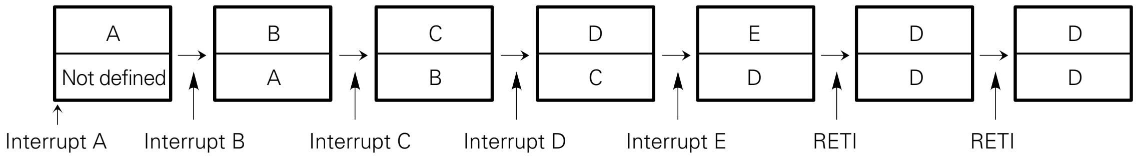
If more than two interrupt levels are accepted, the data saved first is removed from the stack, and so must be saved by software.

Fig. 11-16 Interrupt Stack Operation

(a) When the number of interrupt levels does not exceed 2



(b) When the number of interrupt levels exceeds 2



11.4 STACK POINTER, ADDRESS STACK REGISTER, AND PROGRAM COUNTER

The address stack register holds the return address to which control returns from an interrupt handling routine.

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

When an interrupt is accepted, the value of the stack pointer is decremented by one, and the current value of the program counter is saved in the address stack register location specified by the stack pointer.

When the interrupt return instruction (RETI instruction) is executed after execution of the interrupt handling routine, the contents of the address stack register location, specified by the stack pointer, are restored to the program counter. The stack pointer is then incremented by one.

See also **Chapter 3**.

11.5 INTERRUPT ENABLE FLIP-FLOP (INTE)

The interrupt enable flip-flop enables all interrupts.

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

The flip-flop is set and reset by using dedicated instructions: The EI instruction (for setting) and the DI instruction (for resetting).

The EI instruction causes the flip-flop to be set once the instruction immediately after the EI instruction has been executed. The DI instruction resets the flip-flop upon execution of the DI instruction.

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

Nothing occurs when a DI instruction is executed while interrupts are disabled (DI state), or when an EI instruction is executed while interrupts are enabled (EI state).

When a power-on reset occurs, when a clock-stop instruction is executed, or when a CE reset occurs, the flip-flop is reset.

11.6 ACCEPTING INTERRUPTS

11.6.1 Operation for Accepting Interrupts and Priorities

Interrupts are accepted in the following sequence:

(1) When an interrupt condition (for example, a high-to-low input signal transition occurs on the INT_0 pin) is satisfied, the peripheral hardware outputs an interrupt request signal to the associated interrupt request block.

(2) Upon receiving of the interrupt request signal from the peripheral hardware, the interrupt request block sets its interrupt request flag (IRQ0 flag for the INT_0 pin, for example) to 1.

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

(4) The signal output from the interrupt request block is ANDed with the output of the interrupt enable flip-flop, and the interrupt acceptance signal is output.
The interrupt enable flip-flop is set to 1 with the EI instruction, and is reset to 0 with the DI instruction.
If 1 is output from an interrupt request block while the interrupt enable flip-flop is set to 1, an interrupt is accepted.

When an interrupt is accepted, the output of the interrupt enable flip-flop is applied to each interrupt request block via an AND circuit, as shown in Fig. 11-1.

The signal applied to the interrupt request block for the accepted interrupt resets the corresponding interrupt request flag to 0, and causes the vector address corresponding to the interrupt to be output.

If an interrupt request block outputs 1 at this time, the interrupt acceptance signal is not transferred to the subsequent interrupt request blocks. When more than one interrupt request is generated at any one time, the interrupts are accepted according to the priorities shown below.

If the interrupt enable flag for an interrupt source is not set to 1, the interrupt for that interrupt source is not accepted.

Therefore, an interrupt with a high hardware priority can be disabled by resetting the corresponding interrupt enable flag to 0.

Table 11-2 Interrupt Priorities

Interrupt source	Priority
INT_{NC} pin	1
INT_0 pin	2
Timer 0	3
Timer 1	4
Basic timer 2	5
VRAM pointer	6
Interrupt group 1	7
Serial interface 0	8
Serial interface 1	9
Interrupt group 0	10

11.6.2 Timing Charts for Accepting Interrupts

Fig. 11-17 shows the timing charts for accepting interrupts.

The timing charts in (1) of Fig. 11-17 apply to the use of one interrupt.

Timing chart (a) in (1) shows how an interrupt is accepted when the interrupt request flag is set to 1. Timing chart (b) in (1) shows how an interrupt is accepted when the interrupt enable flag is set to 1.

In both cases, the interrupt is accepted when the interrupt request flag, interrupt enable flip-flop, and interrupt enable flag have all been set to 1.

If the last flag or flip-flop is set to 1 during the execution of MOVT DBF, the first instruction cycle of the @AR instruction, or an instruction that satisfies the skip conditions, the interrupt is accepted after the execution of MOVT DBF, the second cycle of the @AR instruction, or the skipped instruction (NOP instruction).

The interrupt enable flip-flop is set in the instruction cycle immediately after the EI instruction.

This means that when the interrupt request flag is set during the execution cycle of the EI instruction, the instruction immediately after the EI instruction is executed, after which the interrupt is accepted.

The timing charts in (2) of Fig. 11-17 apply to the use of multiple interrupts.

When multiple interrupts are used, and their interrupt enable flags are all set, they are accepted according to the hardware priorities. The hardware priorities, however, can be changed by programming the settings of the interrupt enable flags.

The interrupt cycle shown in Fig. 11-17 is applied to operations performed after an interrupt has been accepted; these operations include the resetting of the interrupt request flag, specification of a vector address, and saving of the program counter contents. This cycle requires 2 μs (when an 8-MHz crystal is used), which is equal to the time required for executing one instruction.

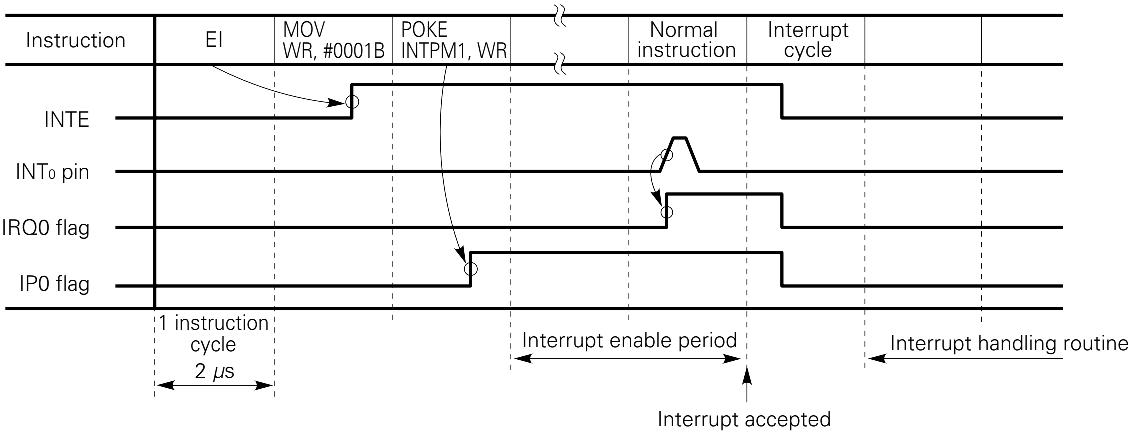
For details, see **Section 11.7**.

Fig. 11-17 Timing Charts for Accepting Interrupts (1/2)

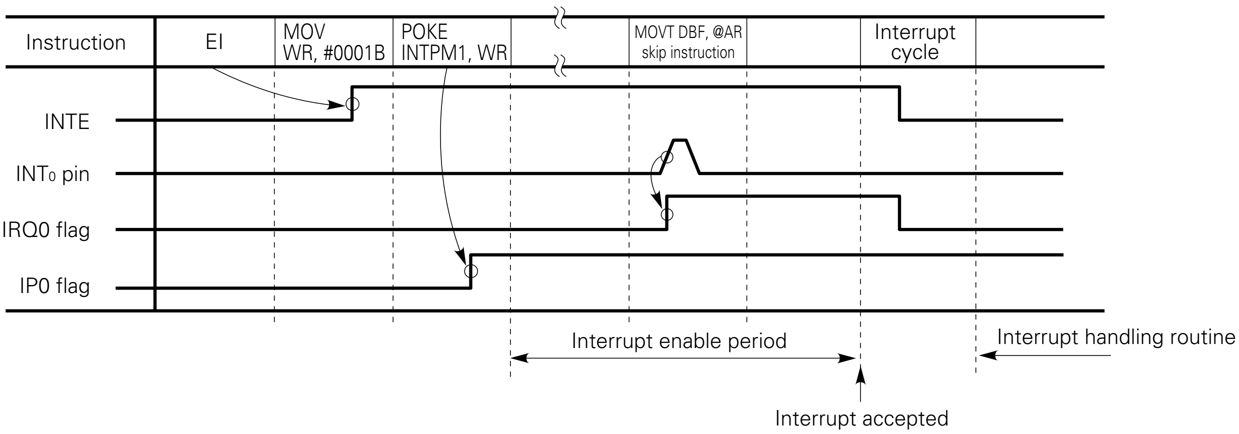
(1) When one interrupt type (example: Low-to-high transition on the INT0 pin) is used

(a) With no interrupt mask time set with interrupt enable flag (IP×××)

① When an interrupt is accepted while a normal instruction is being executed (the instruction is not a MOVT instruction or an instruction that satisfies the skip conditions)



② When an interrupt is accepted while a MOVT instruction or an instruction that satisfies the skip conditions is being executed



(b) With an interrupt pending period set with the interrupt enable flag

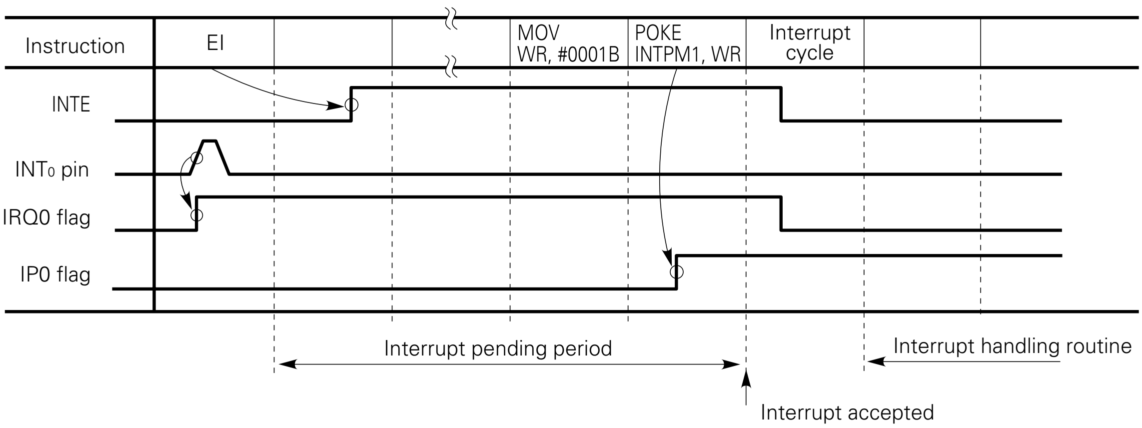
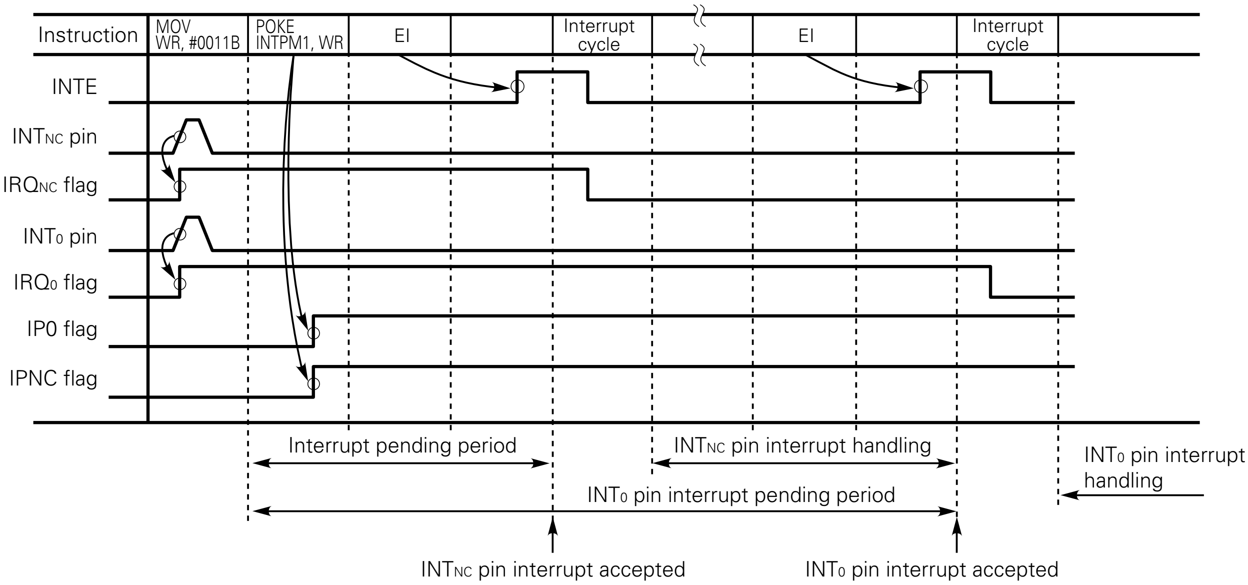


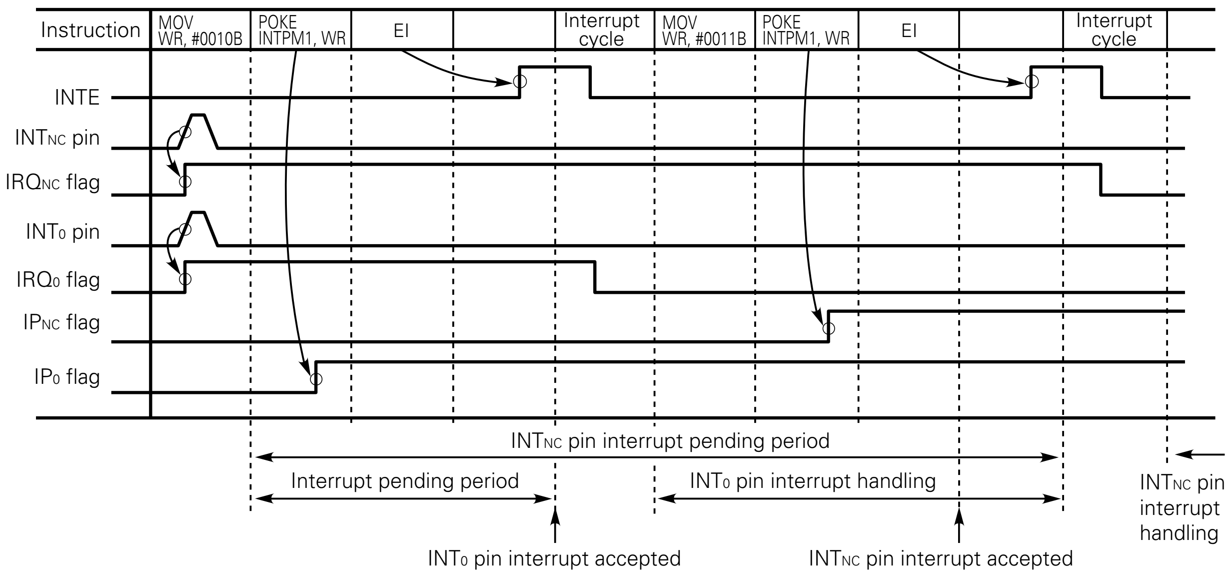
Fig. 11-17 Timing Charts for Accepting Interrupts (2/2)

(2) When multiple interrupts (example: INT_0 and INT_{NC} pins) are used

(a) With hardware priority



(b) With software priority



11.7 OPERATION AFTER AN INTERRUPT IS ACCEPTED

When an interrupt is accepted, the following operations are performed, automatically and in the order shown:

(1) The interrupt enable flip-flop and the interrupt request flag for the accepted interrupt request are reset to 0. This indicates that the interrupt disable state is entered.

(2) The value in the stack pointer is decremented by one.

(3) The program counter contents are saved to the address stack register location specified by the stack pointer.
The saved program counter contents indicate the program memory address subsequent to the address at which the interrupt was accepted.
If the interrupt was accepted during the execution of a branch instruction, for example, the branch destination address is indicated in the program counter. If the interrupt was accepted during the execution of a subroutine call instruction, the called address is indicated. If the skip conditions are satisfied in a skip instruction, the next instruction is executed as an NOP instruction, after which the interrupt is accepted. In this case, the program counter indicates the address subsequent to the skipped instruction.

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

(5) The contents of the vector address generator for the accepted interrupt are transferred to the program counter to branch to the interrupt handling routine.

For operations (1) to (5) above, a special one-instruction cycle (2 μs), that does not involve the execution of a normal instruction, is required.

Such an instruction cycle is called an interrupt cycle.

This means that one instruction cycle (2 μs) is required to branch to the corresponding vector address after the interrupt has been accepted.

11.8 RETURN FROM THE INTERRUPT HANDLING ROUTINE

To return control from the interrupt handling routine to the processing being performed when the interrupt was accepted, the interrupt return instruction (RETI instruction) is used.

When the RETI instruction is executed, the following operations are performed, automatically and in the order shown:

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

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

(3) The stack pointer value is incremented by one.

Operations (1) to (3) above are performed during the one instruction cycle (2 μs) in which the RETI instruction is executed.

The RETI instruction differs from the RET and RETSK instructions, which are subroutine return instructions, only in operation (2) above (system register restoration).

11.9 EXTERNAL INTERRUPTS (INT_0 PIN, INT_{NC} PIN, $\overline{V_{SYNC}}$ PIN, $\overline{H_{SYNC}}$ PIN)

11.9.1 Outline of External Interrupts

Fig. 11-18 is an outline of the external interrupts.

As shown in the figure, an external interrupt request occurs on a rising or falling edge of a signal applied to the INT_0 pin, INT_{NC} pin, $\overline{V_{SYNC}}$ pin, or $\overline{H_{SYNC}}$ pin.

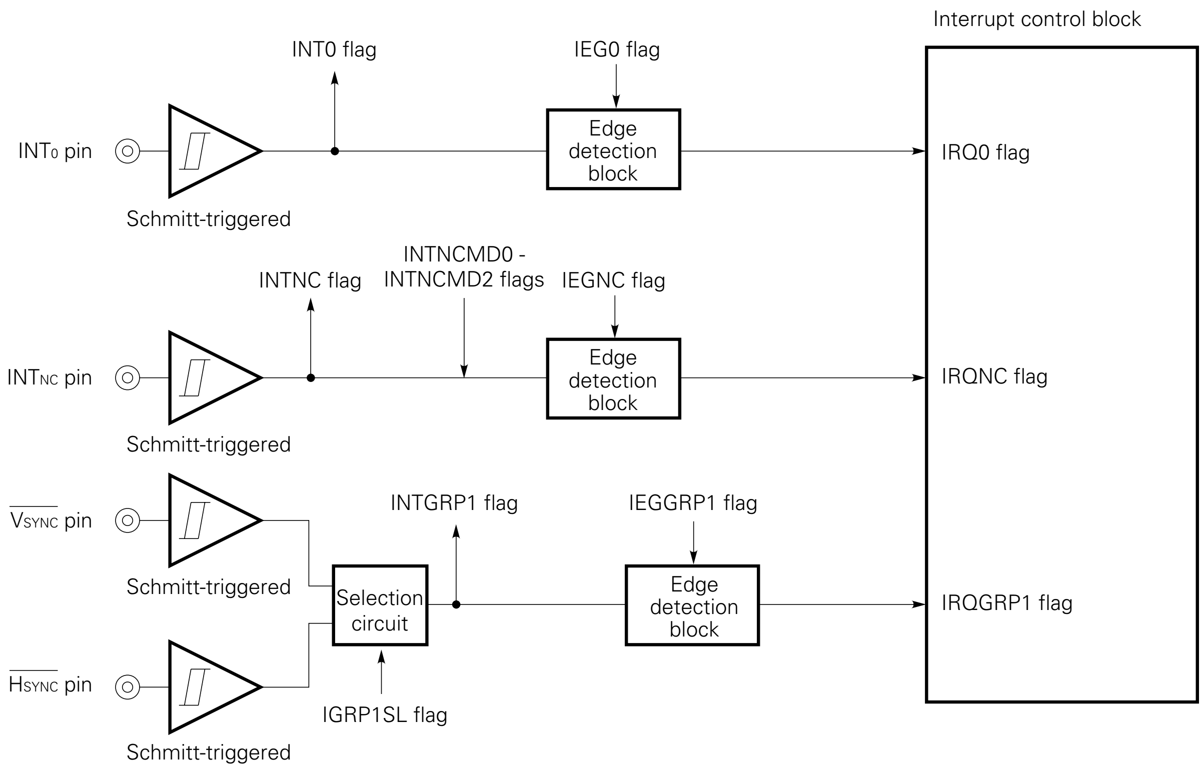
Whether the rising or falling edge is to be used for requesting interrupts can be programmed separately for each pin.

The INT_{NC} pin can be used as a special input pin for remote control by selecting an appropriate pulse width for accepting an interrupt.

Each pin is a Schmitt-triggered input. This can prevent malfunctions due to noise. Pulse widths of less than 1 μs are ignored.

As the interrupt source, either the $\overline{V_{SYNC}}$ or $\overline{H_{SYNC}}$ pin (interrupt group 1) can be specified with the IGRP1SL flag. See **Section 11.10.7**.

Fig. 11-18 Schematic Diagram of External Interrupts



Remark

INT0 INTNC INTGRP1	Detect pin states.
IEG0 IEGNC IEGGRP1	Select the interrupt edge.
IGRP1SL	: Selects the interrupt source for interrupt group 1.
INTNCMD0 to INTNCMD2	: Select the pulse width for accepting interrupts.

11.9.2 Edge Detection Blocks

The edge detection blocks detect the input signal edge (rising or falling edge) that causes an interrupt request to be generated for the INT_0 pin, INT_{NC} pin, and $\overline{V_{SYNC}}$ and $\overline{H_{SYNC}}$ pins. Each input signal edge is selected with the interrupt edge selection register.

Fig. 11-19 shows the format and functions of the interrupt edge selection register.

Fig. 11-19 Format of the Interrupt Edge Selection Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Interrupt edge selection register	0	IEGGRP1	IEG0	IEGNC	1FH	R/W

IEGNC: Specifies the input edge at which an interrupt request is generated. (INT_{NC} pin)

IEGNC	
0	Rising edge
1	Falling edge

IEG0: Specifies the input edge at which an interrupt request is generated. (INT_0 pin)

IEG0	
0	Rising edge
1	Falling edge

IEGGRP1: Specifies the input edge at which an interrupt request is generated. ($\overline{H_{SYNC}}$ or $\overline{V_{SYNC}}$ pin: Selected by the IGRP1SL flag)

IEGGRP1	
0	Rising edge
1	Falling edge

b_3: Fixed to 0

Upon reset	b_3	b_2	b_1	b_0
Power-on	0	0	0	0
Clock stop	↓	0	0	0
CE	↓	0	0	0

Note that when the edge used for requesting interrupts is changed by changing the setting of the interrupt edge selection flag, an interrupt request signal may be issued immediately.

Suppose that the IEG0 flag is set to 1 (falling edge), and that a high is applied to the INT_0 pin, as shown in Table 11-3. Here, note that if the IEG0 flag is reset to 0, the edge detection circuit assumes that a rising edge has been input, and generates an interrupt request.

Table 11-3 Interrupt Requests Generated by Changing the IEG0, IEGNC, and IEGGRP1 Flag Settings

Change in interrupt edge selection flag setting	Pin state	Generation of interrupt request	Interrupt request flag state
1 → 0 (Falling edge) (Rising edge)	Low	Not generated	Previous state is held.
	High	Generated	Set to 1.
0 → 1 (Rising edge) (Falling edge)	Low	Generated	Set to 1.
	High	Not generated	Previous state is held.

11.9.3 Interrupt Control Block

The levels of the signals applied to the pins can be detected with the INTNC, INT0, and INTGRP1 flags, respectively.

These flags are set and reset regardless of the interrupts. When the interrupt function is not being used, these flags can be used as a 3-bit general-purpose input port.

When interrupts are not enabled, these flags can be used as general-purpose ports that can detect a rising or falling edge by reading the interrupt request flags.

In this case, the interrupt request flags are not reset automatically; they must be reset by the program.

Also see **Section 11.2.1**.

11.9.4 Input Pin for Remote Control (INT_{NC})

The input pin for remote control (INT_{NC} pin) differs from normal external interrupt input pins in that the pulse width for accepting interrupts can be selected from among several width options.

The pulse width is specified with the INTNC mode select register.

Fig. 11-20 shows the format and functions of the INTNC mode select register.

Fig. 11-20 Format of the INTNC Mode Select Register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
INT_{NC} mode selector register	0	INTNCMD2	INTNCMD1	INTNCMD0	15H	R/W

b2–b0: Specifies the pulse width for accepting INT_{NC} pin interrupts.

b2	b1	b0	
0	0	0	Accepts interrupts on the pulse edge.
0	0	1	200 μs
0	1	0	400 μs
0	1	1	2 ms
1	0	0	4 ms
Other than above			Not to be set

b3: Fixed to 0

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	0	0	0
CE	↓	0	0	0

11.10 INTERNAL INTERRUPTS

The following seven internal interrupt types are provided.

- Timer 0
- Timer 1
- Basic timer 2
- VRAM pointer
- Serial interface 0
- Serial interface 1
- Interrupt group 0 (timer 0 overflow)

11.10.1 Timer 0 Interrupt

An interrupt request is generated at regular intervals.
See **Chapter 12** for details.

11.10.2 Timer 1 Interrupt

An interrupt request is generated at regular intervals.
See **Chapter 12** for details.

11.10.3 Basic Timer 2 Interrupt

An interrupt request is generated at regular intervals.
See **Chapter 12** for details.

11.10.4 VRAM Pointer Interrupt

An interrupt request is generated at regular intervals.
See **Chapter 16** for details.

11.10.5 Serial Interface 0 Interrupt

Upon the completion of serial output or serial input, an interrupt request can be generated.
See **Chapter 15** for details.

11.10.6 Serial Interface 1 Interrupt

An interrupt request can be generated on the rising edge of the eighth clock pulse, counting from the start of serial interface 1.
See **Chapter 15** for details.

11.10.7 Interrupts by Interrupt Group 0 and Interrupt Group Selection Register

Interrupt group 0 generates an interrupt request when timer 0 overflows.

For details of the conditions governing the request of interrupts, see **Chapter 12.**

Whether to use interrupts caused by the overflow of timer 0 is specified with the IGRP0SL flag of the interrupt group selection register.

In addition, the interrupt group selection register selects the interrupt source for interrupt group 1 (external interrupts). Fig. 11-21 shows the format and functions of the interrupt group selection register.

Caution To use interrupts caused by the overflow of timer 0, both the interrupt enable flag (IPGRP0) and the IGRP0SL flag of the interrupt group selection register must be set to 1.

Fig. 11-21 Format of the Interrupt Group Selection Register

Register	Flag symbol b_3	b_2	b_1	b_0	Address	Read/write
Interrupt group selection register	0	0	IGRP1SL	IGRP0SL	0FH	R/W

IGRP0SL: Specifies whether to use timer 0 overflow interrupts (group 0).

IGRP0SL	
0	Does not use interrupts.
1	Uses interrupts.

IGRP1SL: Selects the interrupt source (group 1).

IGRP1SL	
0	$\overline{V_{SYNC}}$ signal
1	$\overline{H_{SYNC}}$ signal

b_3, b_2: Fixed to 0

Upon reset	b_3	b_2	b_1	b_0
Power-on	0	0	0	0
Clock stop	↓	↓	0	0
CE	↓	↓	0	0

12. TIMERS

The timers in the μPD17068 are used to manage the time required to execute programs.

12.1 OVERVIEW

Fig. 12-1 shows the block diagrams of the timers.

The μPD17068 contains the following six different timers.

- Basic timer 0
- Basic timer 1
- Basic timer 2
- Timer 0 (modulo scheme)
- Timer 1 (modulo scheme)
- Clock timer

Basic timers 0 and 1 are realized by detecting the state of a flip-flops that is set at constant intervals, using software.

Basic timer 2 issues an interrupt request at constant intervals.

Timers 0 and 1 are modulo timers. They issue an interrupt request at constant intervals.

The clock timer counts seconds, minutes, hours, and days.

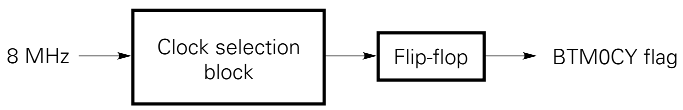
Basic timer 0 is used also to detect a power failure.

The clock pulses for the timers except the clock timer are generated by dividing the frequency of the system clock (8 MHz).

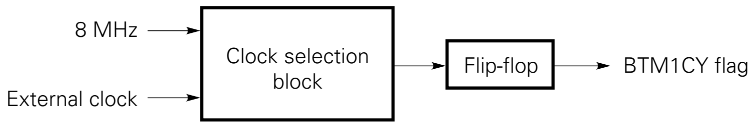
The clock pulse for the clock timer is generated by dividing a frequency of 32 kHz.

Fig. 12-1 Overview of Timers (1/2)

(1) Basic timer 0



(2) Basic timer 1



(3) Basic timer 2

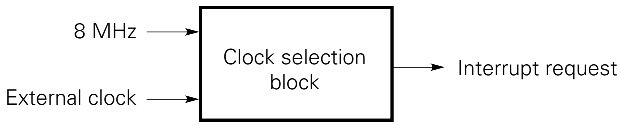
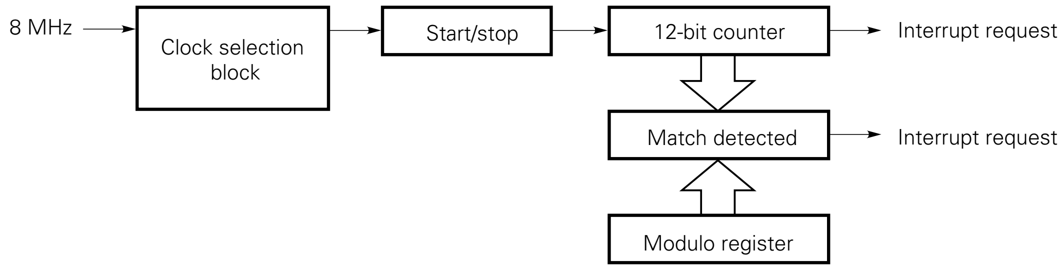
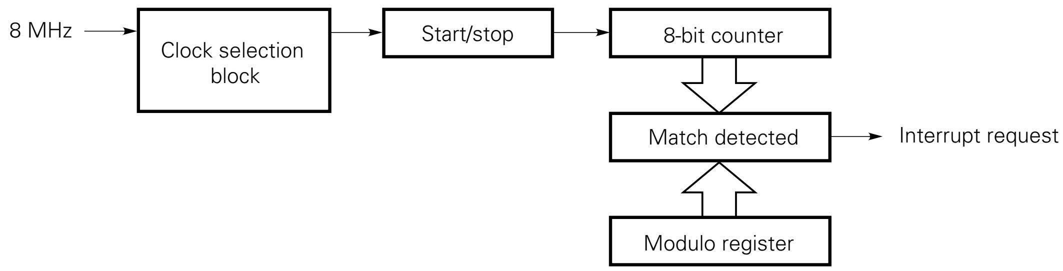


Fig. 12-1 Overview of Timers (2/2)

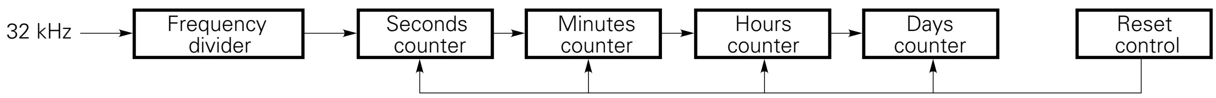
(4) Timer 0



(5) Timer 1



(6) Clock timer



12.2 BASIC TIMER 0

12.2.1 Overview of Basic Timer 0

Fig. 12-2 shows the block diagram of basic timer 0.

Basic timer 0 is realized by detecting the state of a flip-flop that is set at constant intervals, using the BTM0CY flag (bit 0 at address RF:17H).

The contents of the flip-flop correspond to the states of the BTM0CY flag on a one-to-one basis.

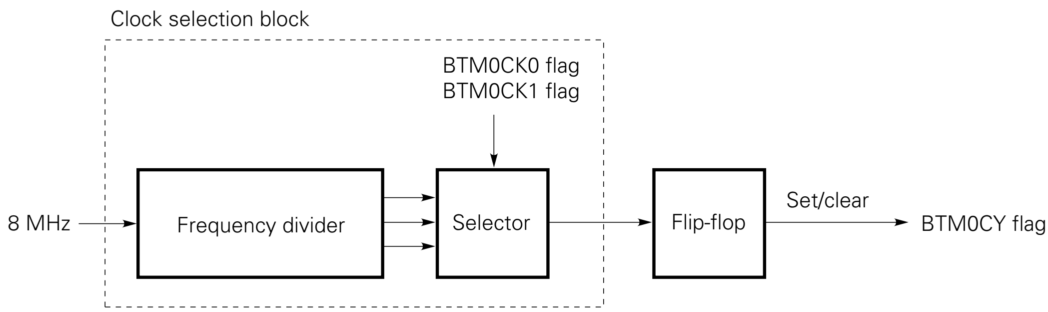
When the BTM0CY flag is read-accessed for the first time after a power-on reset, it is read as “0”. From then on, the flag is set to “1” at constant intervals.

After the CE pin goes from a low level to a high level, a CE reset occurs simultaneously when the BTM0CY flag is set next time.

A power failure can therefore be detected by reading the BTM0CY flag when a system reset (power-on or CE reset) occurs.

See **Chapter 20** for power failure detection.

Fig. 12-2 Block Diagram of Basic Timer 0



Remarks 1. BTM0CK1 and BTM0CK0 (bits 1 and 0 of the basic timer 0 clock select register, respectively; see **Fig. 12-3**) specify the time interval at which the BTM0CY flag is set.

2. BTM0CY (bit 0 of the basic timer 0 carry flip-flop judge register; see **Fig. 12-4**) reflects the state of the flip-flop.

12.2.2 Clock Selection Block

The clock selection block divides the frequency of the system clock (8 MHz) and specifies the time interval at which the BTM0CY flag is set, using the basic timer 0 clock select register.

Fig. 12-3 shows the configuration and function of the basic timer 0 clock select register.

Fig. 12-3 Configuration of the Basic Timer 0 Clock Select Register

Register	Flag symbol				Address	Read/write
	b3	b2	b1	b0		
Basic timer 0 clock select register	0	0	BTM0CK1	BTM0CK0	0CH	R/W

BTM0CK1	BTM0CK0	Specify the time interval at which the BTM0CY flag is set.
0	0	100 ms
0	1	5 ms
1	0	1 ms
1	1	1 ms

b3, b2: Fixed to “0”

Upon reset		b3	b2	b1	b0
	Power-on	0	0	0	0
	Clock stop	↓	↓	0	0
	CE	↓	↓	Hold	

12.2.3 Flip-Flop and BTM0CY Flag

The flip-flop is set at constant intervals, and its state is detected using the BTM0CY flag of the basic timer 0 carry flip-flop judge register.

The BTM0CY flag is reset to “0” by reading its contents into a window register using the PEEK instruction (Read & Reset).

The BTM0CY flag is “0” at a power-on reset. It becomes “1” at a CE reset. So, it can be used as a power failure detection flag.

Even when power is supplied, the BTM0CY flag will not be set until a read instruction is executed. Once a read instruction is executed, the flag is set at constant intervals.

Fig. 12-4 shows the configuration and function of the basic timer 0 carry flip-flop judge register.

Fig. 12-4 Configuration of the Basic Timer 0 Carry Flip-Flop Judge Register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Basic timer 0 carry flip-flop judge register	0	0	0	BTM0CY	17H	R & Reset

BTM0CY: Detects the state of the flip-flop.

0	The flip-flop is not set.
1	The flip-flop is set.

b3, b2, b1: Fixed to "0"

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	1
CE	↓	↓	↓	1

12.2.4 Example of Using Basic Timer 0

A sample program follows.

The following program performs process A at every one second.

Example

```
        CLR2    BTM0CK1, BTM0CK0    ; Specifies that the BTM0CY flag be set at intervals of 100 ms.
LOOP:
        SKT1    BTM0CY              ; Branches to NEXT if BTM0CY is "0".
        BR      NEXT
        ADD     M1, #1              ; Adds 1 to M1.
        SKGE    M1, #0AH            ; Performs process A if M1 is 10 or greater (1 second).
        BR      NEXT
        MOV     M1, #0
        Process A
NEXT:
        Process B                   ; Performs process B and branches to LOOP.
        BR      LOOP
```

12.2.5 Time Interval Error in Basic Timer 0

There are two types of errors in basic timer 0; one type is related to the time interval at which the BTM0CY flag is detected, and the other type can occur when the time interval at which the BTM0CY flag is set is changed.

Each type of error is described under (1) and (2).

(1) Error related to the detection time of the BTM0CY flag

The time interval at which the BTM0CY flag is detected must be shorter than the time interval at which the BTM0CY flag is set. (See **Section 12.2.6.**)

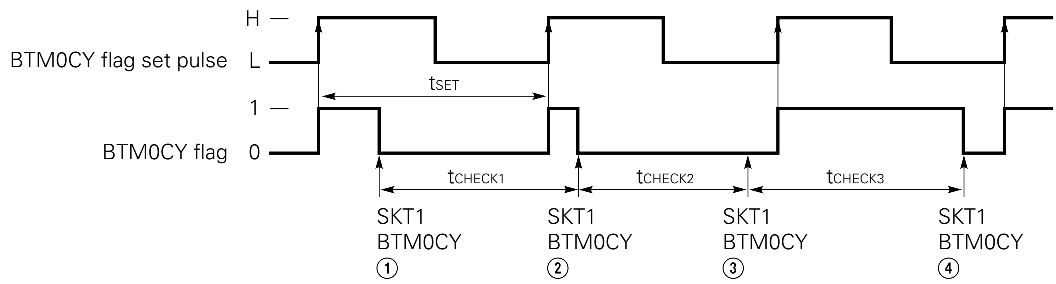
In other words, assuming that the BTM0CY flag is detected at intervals of t_{CHECK} and is set at intervals of t_{SET} (100, 5, or 1 ms), the relationship between t_{CHECK} and t_{SET} must be as follows:

$t_{CHECK} < t_{SET}$

Under this condition, the time interval error encountered when the BTM0CY flag is detected is as follows:

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

Fig. 12-5 Basic Timer 0 Error Related to the Detection Time of the BTM0CY Flag



As shown in Fig. 12-5, when the BTM0CY flag is detected at ②, the timer is updated because the flag is "1". When the flag is detected at ③, because it is "0", the timer is not updated until it is detected again at ④. Therefore, the timer is incremented by t_{CHECK3}.

(2) Error due to a change to the time interval at which the BTM0CY flag is set

The BTM0CK1 and BTM0CK0 flags specify the time interval at which the BTM0CY flag is set.

As shown in **Section 12.2.2**, the time interval set pulse can be selected from three types, 1 kHz, 200 Hz, and 10 Hz.

These three pulses operate independently. When the time interval set pulse is switched using the BTM0CK1 and BTM0CK0 flags, therefore, an error occurs as explained in the following example.

Example

```
; ①
INITFLG NOT BTM0CK1, BTM0CK0 ; Selects 200 Hz (5 ms) as the BTM0CY flag set pulse.
```

Process A

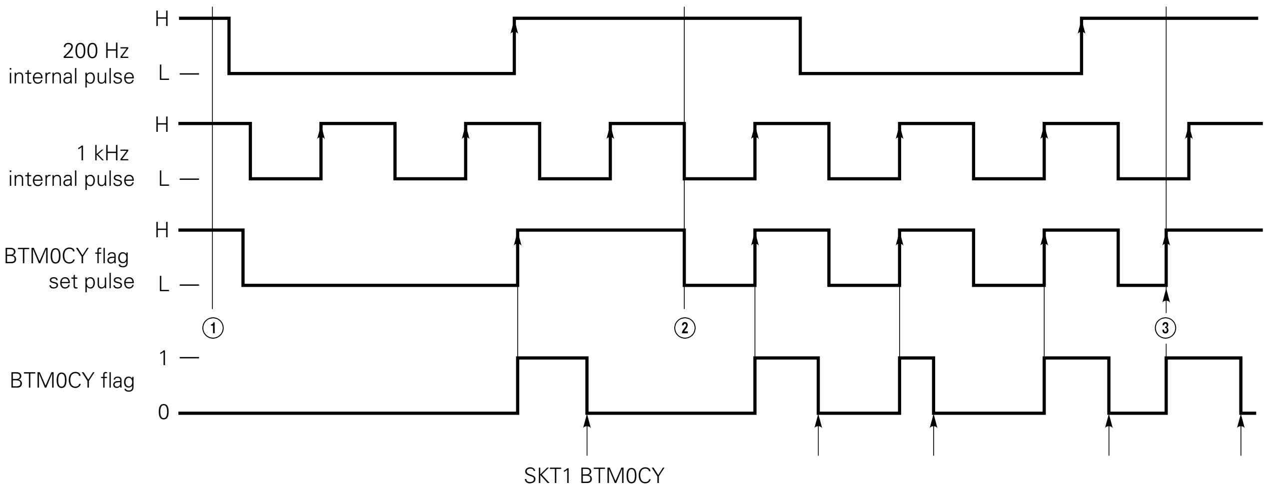
```
; ②
INITFLG BTM0CK1, NOT BTM0CK0 ; Selects 1 kHz (1 ms) as the BTM0CY flag set pulse.
```

Process A

```
; ③
INITFLG NOT BTM0CK1, BTM0CK0 ; Selects 200 Hz (5 ms) as the BTM0CY flag set pulse.
```

With this coding, the BTM0CY flag set pulse is switched as shown in Fig. 12-6.

Fig. 12-6 BTM0CY Flag Set Pulse Switching



As shown in Fig. 12-6, when the BTM0CY flag set time interval is switched, if the selected pulse falls, the BTM0CY flag maintains its previous state (② in the figure). If the selected pulse rises, the BTM0CY flag is set to “1” (③ in the figure).

This example illustrates switching between 200 Hz (5 ms) and 1 kHz (1 ms). This explanation also applies to switching between 200 Hz and 10 Hz (100 ms) and between 1 kHz and 10 Hz.

Consequently, if the BTM0CY flag set time interval is switched as shown in Fig. 12-7, a time interval error observed before the BTM0CY flag is set for the first time is as given below.

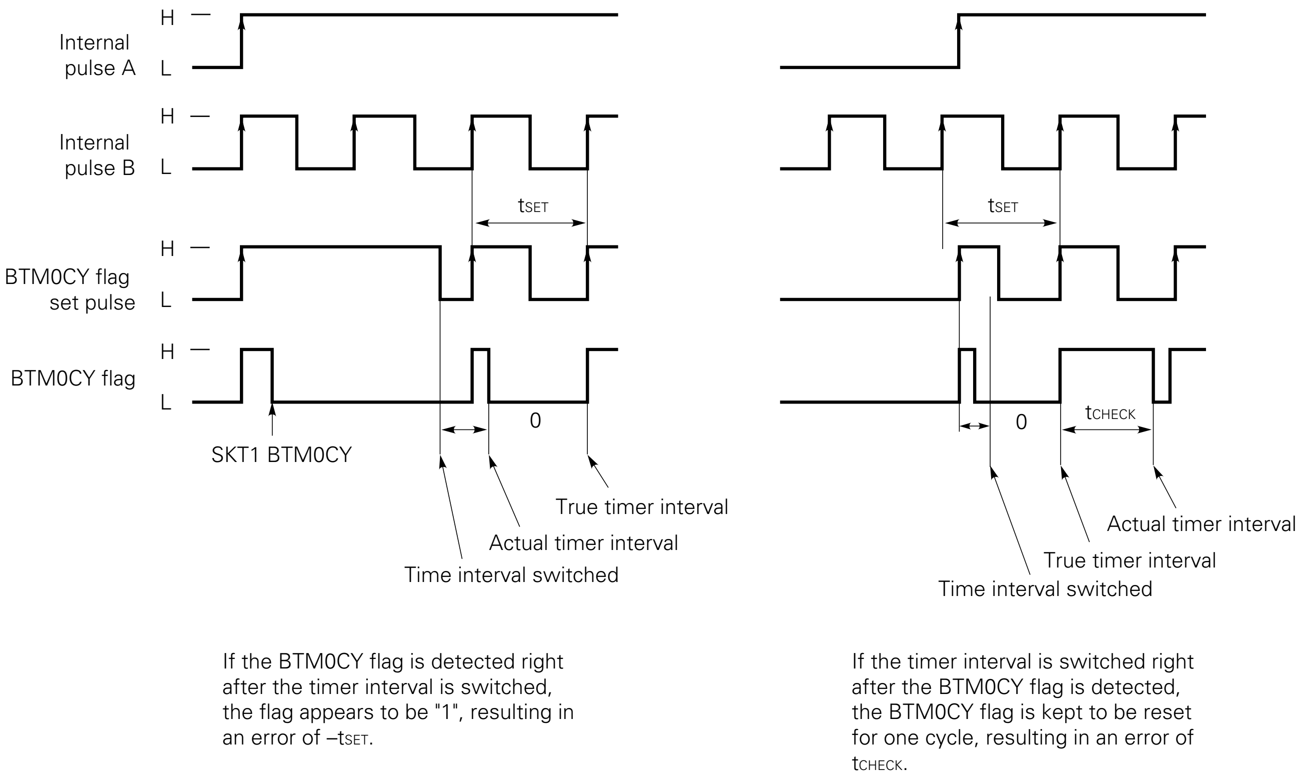
– t_{SET} < error < t_{CHECK}

t_{SET} : Newly selected BTM0CY flag set time interval
t_{CHECK} : BTM0CY flag detection time interval

The 10 Hz, 200 Hz, and 1 kHz internal pulses have a phase difference from one another. These phase differences are shorter than the pulse interval and included in the error described above.

See **Section 12.4.5** for each pulse phase difference.

Fig. 12-7 Timer Error That Occurs When the BTM0CY Flag Set Time Interval Is Switched from A to B



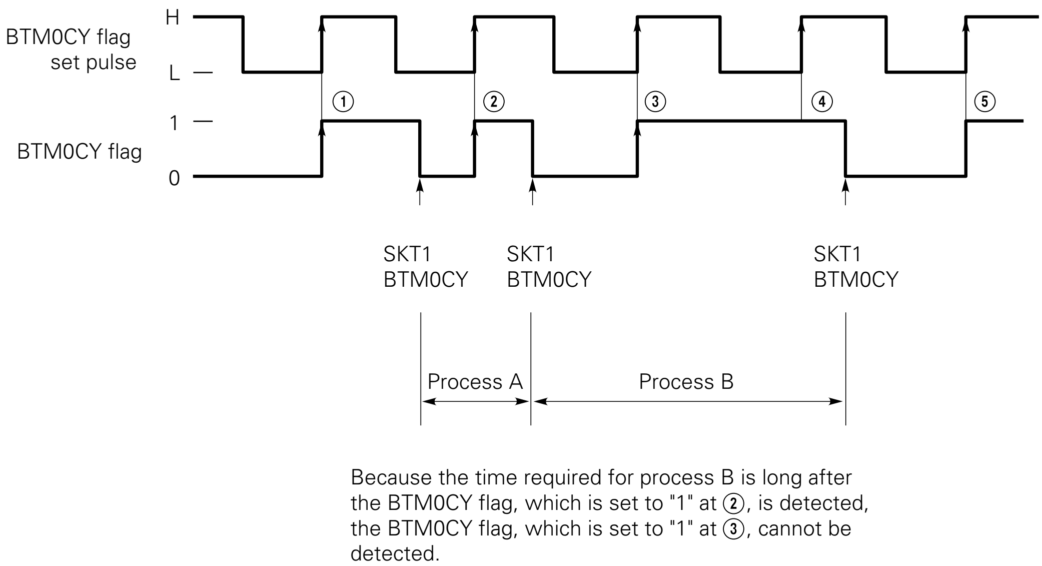
12.2.6 Cautions for Using Basic Timer 0

(1) BTM0CY flag detection time interval

Keep the BTM0CY flag detection time interval shorter than the BTM0CY flag set time interval.

Otherwise, the BTM0CY flag cannot be set if the time required for process B is longer than the BTM0CY flag set time interval, as shown in Fig. 12-8.

Fig. 12-8 BTM0CY Flag and Its Detection



(2) Sum of the timer update process time and the BTM0CY flag detection time interval

As explained in (1), the BTM0CY flag detection time interval (t_{SET}) must be kept shorter than the BTM0CY flag set time interval.

Even when the BTM0CY flag detection time interval is short, however, if the timer update process time is long, a CE reset may prevent a normal timer update process.

The following conditions must therefore be satisfied.

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

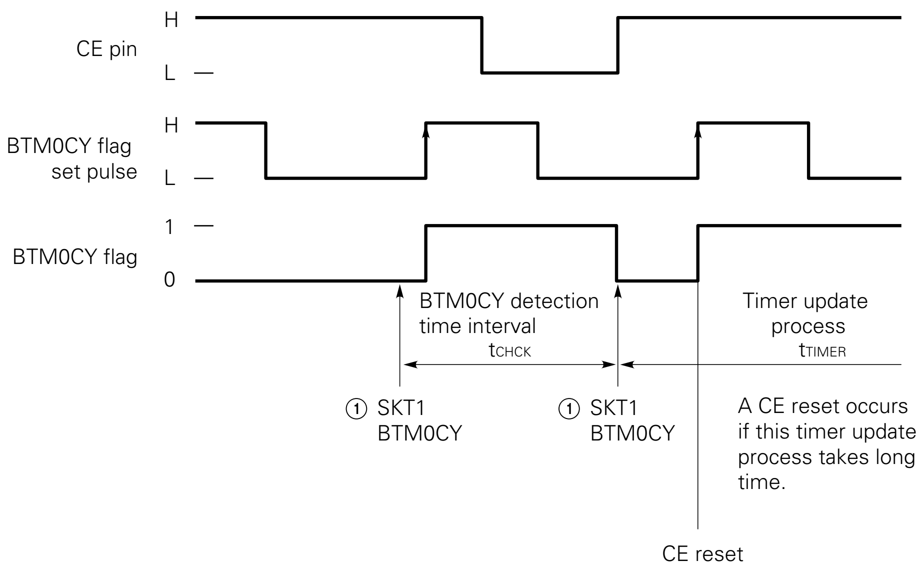
where t_{CHECK} : BTM0CY flag detection time interval
t_{TIMER} : Timer update process time
t_{SET} : BTM0CY flag set time interval

The coding that meets these conditions is given below.

Example Timer update process and BTM0CY flag detection time interval

```
START:
          CLR2    BTM0CK1, BTM0CK0   ; Specifies 100 ms as the BTM0CY flag set time
                                     ; interval.
BTIMER:
          ; ①
          SKT1    BTM0CY             ; Updates the timer if the BTM0CY flag is "1".
          BR      AAA
          [Timer update]
          BR      BTIMER
AAA:
          [Process A]
          BR      BTIMER
```

The timing chart for this coding is as follows:



(3) Correcting the basic timer 0 carry at a CE reset

The following paragraphs describe an example of correcting the timer at a CE reset.

As explained in the example, it is necessary to correct the timer at a CE reset, if the BTM0CY flag is used both to detect a power failure and to control the clock timer.

The BTM0CY flag is reset (0) when the supply voltage is first applied (at a power-on reset), and kept disabled from being set until it is read-accessed using the PEEK instruction.

When the CE pin goes from a low level to a high level, a CE reset occurs in synchronization with the rising edge of the BTM0CY flag set pulse, setting the BTM0CY flag to "1" and making it active.

Detecting the state of the BTM0CY flag at a system reset (power-on or CE reset) can therefore check for a power failure. If the flag is "0", it means that a power-on reset has occurred. If it is "1", it means that a CE reset has occurred (power failure detection).

In this case, a clock timer must keep operating even at a CE reset.

However, reading the BTM0CY flag for power failure detection resets the flag (0) and makes it impossible to detect the set (1) state of the flag for one cycle.

To solve this problem, it is necessary to update the clock timer if an attempt to detect a power failure detects a CE reset. See **Section 20.6** for power failure detection.

Example Correcting the timer at a CE reset (when the BTM0CY flag is used for both power failure detection and timer update)

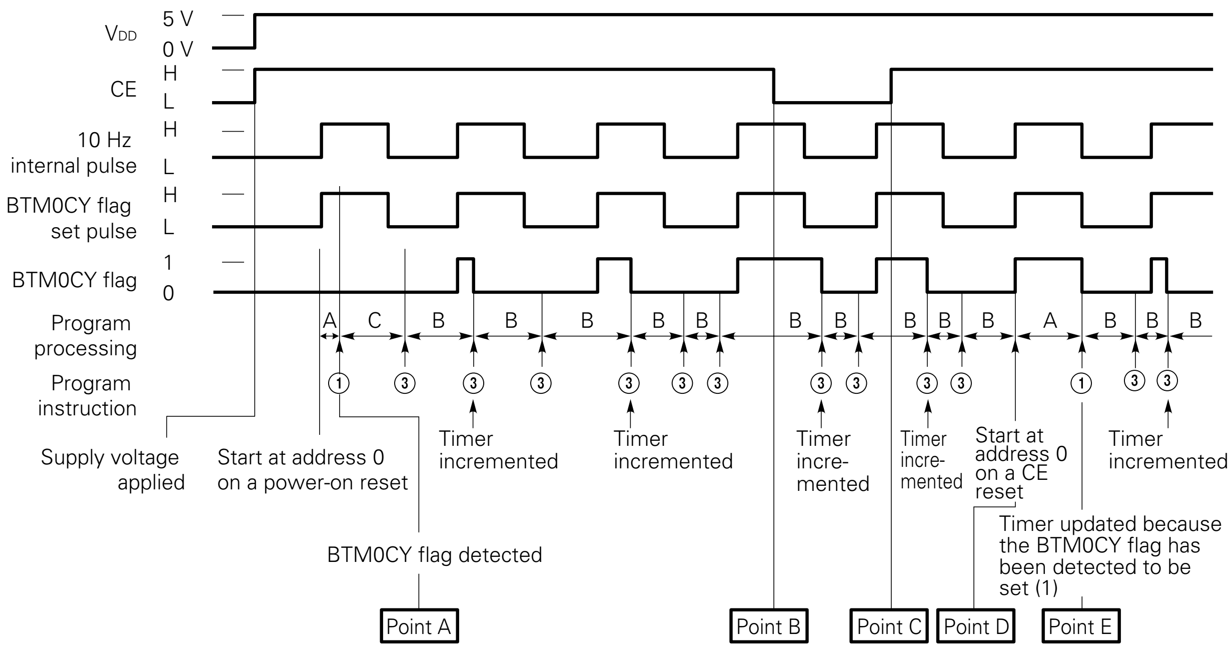
```
START :                                  ; Program address 0000H

              [Process A]
         ; ①
              SKT1    BTM0CY             ; Built-in macro
                                         ; Checks the BTM0CY flag and branches to INITIAL
              BR      INITIAL            ;  if the flag is "0" (power failure detected).

BACKUP :
         ; ②
              [Timer update by 100 ms]   ; Timer correction because of backup (CE reset)
LOOP:
         ; ③
              [Process B]                ; While performing process B,
              SKF1    BTM0CY  ;tests the BTM0CY flag and updates the timer.
              BR      BACKUP
              BR      LOOP
INITIAL :
              CLR2    BTM0CK1, BTM0CK0
                                         ; Built-in macro
                                         ; The BTM0CY flag set time interval is set to
                                         ; 100 ms and process C is performed because a
                                         ; power failure (power-on reset) has occurred.
              [Process C]
              BR      LOOP
```

Fig. 12-9 shows the timing chart for the above program.

Fig. 12-9 Timing Chart



As shown in Fig. 12-9, when supply voltage V_{DD} is applied, the 10 Hz internal pulse rises to make the program start at address 0000H.

When the BTM0CY flag is detected at point A, the BTM0CY flag appears to be reset (0) because it is just after power is supplied. Consequently, it is determined that a power failure (power-on reset) has occurred.

So, process C is performed to select 100 ms as the BTM0CY flag set pulse.

Because the BTM0CY flag is once read-accessed at point A, the BTM0CY flag will be set (1) at intervals of 100 ms.

Even when the CE pin goes to a low at point B and goes to a high at point C, the program continues to update the clock while performing process B, unless the clock stop instruction is executed.

Because the CE pin goes from a low to a high at point C, a CE reset occurs at point D, where the BTM0CY flag set pulse rises, to start the program at address 0000H.

When the BTM0CY flag is detected at point E, it is determined that a backup (CE reset) has occurred, because the flag appears to be set (1).

Also, as easily seen from the figure, if the clock is not updated by 100 ms at point E, the clock loses 100 ms every time a CE reset occurs.

If process A takes more than 100 ms to detect for a power failure at point E, it is impossible to detect the BTM0CY flag for two cycles. Therefore, process A must be performed within 100 ms.

The above description applies also when either 5 ms or 1 ms is selected as the BTM0CY flag set pulse.

It is necessary, therefore, to detect the BTM0CY flag for power failure detection after the program starts at address 0000H and before the BTM0CY flag is set.

(4) When the BTM0CY flag is detected simultaneously with a CE reset

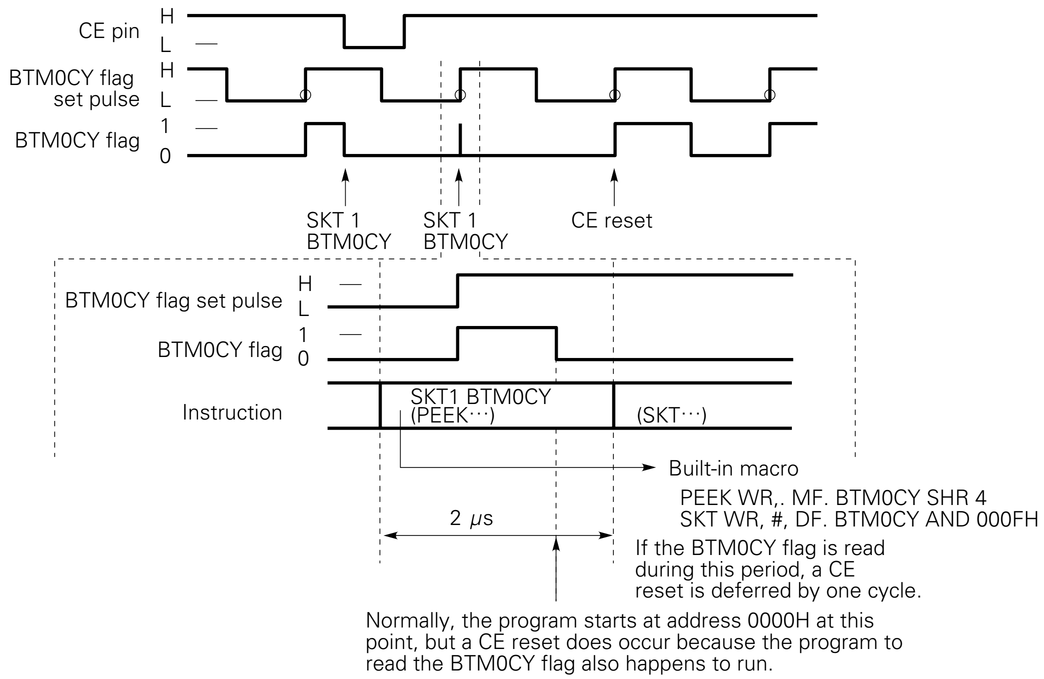
As described in (3), a CE reset occurs at the same time the BTM0CY flag is set (1).

Under this condition, if a read instruction is executed for the BTM0CY flag simultaneously with a CE reset, the read instruction takes preference.

Therefore, if a BTM0CY flag read instruction occurs simultaneously with the rising edge of the BTM0CY flag set pulse that occurs after the CE pin goes from a low to a high, a CE reset occurs when the BTM0CY flag is set on the next cycle.

This operation is illustrated in Fig. 12-10.

Fig. 12-10 Operation That Occurs If a CE Reset Occurs Simultaneously with a BTM0CY Flag Read Instruction



Therefore, if a program is designed to detect the BTM0CY flag cyclically and has the BTM0CY flag detection time interval that matches the BTM0CY flag set time interval, a CE reset may not occur for ever.

Observe the following cautions.

Because one instruction cycle is 2 μs (1/500 kHz), the BTM0CY flag is read at every 1 ms (2 μs × 500) if the program is designed to detect the BTM0CY flag once every 500 instructions.

In this case, regardless of which of 1, 5, and 100 ms is selected as the timer interval set pulse, a CE reset will not occur for ever once the BTM0CY flag set and detection time intervals coincide with each other.

To solve this problem, do not create a program with a cycle that meets the following condition.

$$\frac{(t_{SET} \times 500)}{X} = n \text{ (where n is a natural number)}$$

t_{SET} : BTM0CY flag set time interval

X : Number of steps in a cycle that a BTM0CY flag read instruction is issued

An example of a program that meets the above condition is given below. Do not create such a program.

Example

```
            Process A
            SET2 BTM0CK1, BTM0CK0
                                  ; Built-in macro
                                  ; Specifies 1 ms as the BTM0CY flag set pulse.
LOOP :
        ; ①
            SKT1   BTM0CY         ; Built-in macro
            BR     BBB

AAA    :
            496 steps
        BR         LOOP

BBB    :
            496 steps
        BR         LOOP
```

In this example, a CE reset will not occur for ever if the BTM0CY flag happens to be set at a timing of a BTM0CY flag read instruction at ①, because the read instruction is repeated at every 500 instructions.

12.3 BASIC TIMER 1

12.3.1 Overview of Basic Timer 1

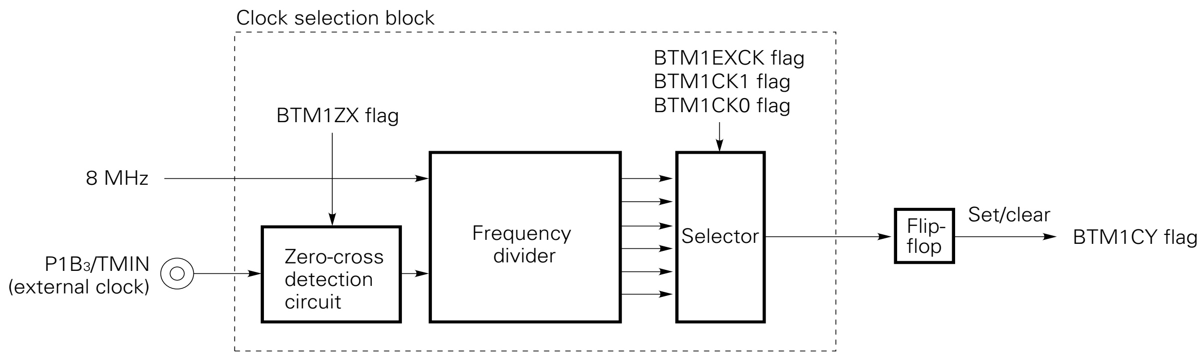
Fig. 12-11 shows the block diagram of basic timer 1.

Basic timer 1 is realized by detecting the state of a flip-flop that is set at constant intervals, using the BTM1CY flag (bit 0 at address RF:16H).

The contents of the flip-flop corresponds the states of the BTM1CY flag on a one-to-one basis.

Basic timer 1 can use an external clock input at the $P1B_3$/TMIN pin as its base clock. It can also detect the zero cross of the external clock.

Fig. 12-11 Block Diagram of Basic Timer 1



Remarks 1. BTM1EXCK (bit 3 of basic timer 1 mode select register; see **Fig. 12-12**) specifies the base clock (internal or external clock).

2. BTM1ZX (bit 2 of basic timer 1 mode select register; see **Fig. 12-12**) specifies whether the zero-cross detection circuit is on or off.

3. BTM1CK1 and BTM1CK0 (bits 1 and 0 of the basic timer 1 mode select register, respectively; see **Fig. 12-12**) specify the time interval at which the BTM1CY flag is set.

4. BTM1CY (bit 0 of the basic timer 1 carry flip-flop judge register; see **Fig. 12-13**) reflects the state of the flip-flop.

12.3.2 Clock Selection Block

The clock selection block divides the frequency of the system clock (8 MHz) or an external clock and specifies the time interval at which the BTM1CY flag is set, using the basic timer 1 mode select register.

Either the system clock (8 MHz) or an external clock input at the $P1B_3$/TMIN pin can be selected as the base clock of basic timer 1. When an external clock is selected, it is possible to select whether the zero-cross detection circuit is turned on or off.

Fig. 12-12 shows the configuration and function of the basic timer 1 mode select register.

Fig. 12-12 Configuration of the Basic Timer 1 Mode Select Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Basic timer 1 mode select register	BTM1EXCK	BTM1ZX	BTM1CK1	BTM1CK0	0BH	R/W

b_3	b_2	b_1	b_0	Specify the time interval at which the BTM1CY flag is set and whether the zero-cross detection circuit is turned on or off.
0	×	0	0	100 ms
0	×	0	1	5 ms
0	×	1	0	1 ms
0	×	1	1	125 μs
1	0	0	×	Divides the frequency of an external clock (at the $P1B_3$/TMIN pin) by 5.
1	0	1	×	Divides the frequency of an external clock (at the $P1B_3$/TMIN pin) by 6.
1	1	0	×	Divides the frequency of an external clock (at the $P1B_3$/TMIN pin) by 5 (with the zero-cross detection circuit on).
1	1	1	×	Divides the frequency of an external clock (at the $P1B_3$/TMIN pin) by 6 (with the zero-cross detection circuit on).

Upon reset					
	Power-on	0	0	0	0
	Clock stop	0	0	0	0
	CE	Hold			

Remark ×: Don't care

12.3.3 Flip-Flop and BTM1CY Flag

The flip-flop is set at constant intervals, and its state is detected using the BTM1CY flag of the basic timer 1 carry flip-flop judge register.

The BTM1CY flag is reset to “0” by reading its contents into a window register using the PEEK instruction (Read & Reset).

Even when power is supplied, the BTM1CY flag will not be set until a read instruction is executed. Once a read instruction is executed, the flag is set at constant intervals.

Fig. 12-13 shows the configuration and function of the basic timer 1 carry flip-flop judge register.

Fig. 12-13 Configuration of the Basic Timer 1 Carry Flip-Flop Judge Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Basic timer 1 carry flip-flop judge register	0	0	0	BTM1CY	16H	R & Reset

BTM1CY:

	Detects the state of the flip-flop.
0	The flip-flop is not set.
1	The flip-flop is set.

b_3–b_1: Fixed to "0"

Upon reset		b_3	b_2	b_1	b_0
	Power-on	0	0	0	0
	Clock stop	↓	↓	↓	1
	CE	↓	↓	↓	1

12.3.4 Time Interval Error in Basic Timer 1

Similarly to basic timer 0, there are two types of errors in basic timer 1; one type is related to the detection time interval of the BTM1CY flag, and the other type can occur when the time interval at which the BTM1CY flag is set is changed.

See **Section 12.2.5.**

12.4 BASIC TIMER 2

12.4.1 Overview of Basic Timer 2

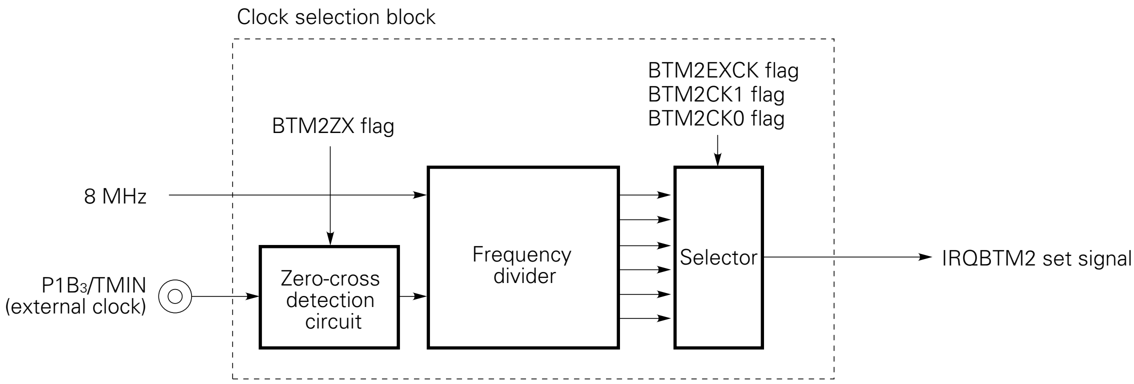
Fig. 12-14 shows the block diagram of basic timer 2.

Basic timer 2 issues interrupt requests at constant intervals and sets (1) the IRQBTM2 flag.

If an EI instruction has been executed and the IPBTM2 flag has been set, the basic timer 2 interrupt requests are accepted when the IRQBTM2 flag is set. (See **Chapter 11.**)

Basic timer 2 can use an external clock input at the $P1B_3$/TMIN pin as its base clock. It can also detect the zero cross of the external clock.

Fig. 12-14 Block Diagram of Basic Timer 2



Remarks 1. BTM2EXCK (bit 3 of the basic timer 2 mode select register; see **Fig. 12-15**) specifies the base clock (internal or external clock).

2. BTM2ZX (bit 2 of the basic timer 2 mode select register; see **Fig. 12-15**) specifies whether the zero-cross detection circuit is on or off.

3. BTM2CK1 and BTM2CK0 (bits 1 and 0 of the basic timer 2 mode select register, respectively; see **Fig. 12-15**) specify the time interval at which the IRQBTM2 flag is set.

12.4.2 Clock Selection Block

The clock selection block divides the frequency of the system clock (8 MHz) or an external clock and specifies the time interval at which the IRQBTM2 flag is set, using the basic timer 2 mode select register.

Either the system clock (8 MHz) or an external clock input at the $P1B_3$/TMIN pin can be selected as the base clock of basic timer 2.

When an external clock is selected, it is possible to select whether the zero-cross detection circuit is turned on or off. With an external clock (8 MHz), it is also possible to select that the zero-cross detection circuits is on or off.

Fig. 12-15 shows the configuration and function of the basic timer 2 mode select register.

Fig. 12-15 Configuration of the Basic Timer 2 Mode Select Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Basic timer 2 mode select register	BTM2EXCK	BTM2ZK	BTM2CK1	BTM2CK2	0AH	R/W

b_3	b_2	b_1	b_0	Specify the time interval at which the IRQBTM2 flag is set and whether the zero-cross detection circuit is turned on or off.
0	×	0	0	100 ms
0	×	0	1	5 ms
0	×	1	0	1 ms
0	×	1	1	125 μs
1	0	0	×	Divides the frequency of an external clock (at the $P1B_3$/TMIN pin) by 5.
1	0	1	×	Divides the frequency of an external clock (at the $P1B_3$/TMIN pin) by 6.
1	1	0	×	Divides the frequency of an external clock (at the $P1B_3$/TMIN pin) by 5 (with the zero-cross detection circuit on).
1	1	1	×	Divides the frequency of an external clock (at the $P1B_3$/TMIN pin) by 6 (with the zero-cross detection circuit on).

Upon reset	Power-on	0	0	0	0
	Clock stop	0	0	0	0
	CE	Hold			

Remark ×: Don't care

12.4.3 Example of Using Basic Timer 2

A sample program follows.

Example

```
        BR      AAA                     ; Branches to AAA.
BTIMER:                                 ; Program address 0006H
        ADD     M1, #0001B              ; Adds 1 to M1.
        SKT1    CY                      ; Tests the CY flag.
        BR      BBB                     ; Returns to the main routine if there is no carry.
        [Process A]
BBB:
        EI
        RETI
AAA:
        INITFLG NOT BTM2EXCK, NOT BTM2ZX, BTM2CK1, NOT BTM2CK0
                                        ; Selects 1 ms as the basic timer 2 interrupt pulse.
        MOV     M1, #000B               ; Clears the M1 contents to 0.
        SET1    IPBTM2                  ; Enables the basic timer 2 interrupt.
        EI                              ; Enables all interrupts.
LOOP:
        [Process B]
        BR      LOOP
```

The above program performs process A at intervals of 1 ms.

Note that when an interrupt request is accepted, subsequent interrupts are disabled. Also note that even if interrupts are disabled, the IRQBTM2 flag can be set to “1”.

In other words, if process A takes 1 ms or more, an interrupt request is accepted when a return is made by a RETI instruction, thus disabling process B from being performed.

12.4.4 Time Interval Error in Basic Timer 2

As explained in **Section 12.4.3**, if an EI instruction has been executed and basic timer 2 interrupts are enabled, an interrupt request is accepted each time the basic timer interrupt pulse rises.

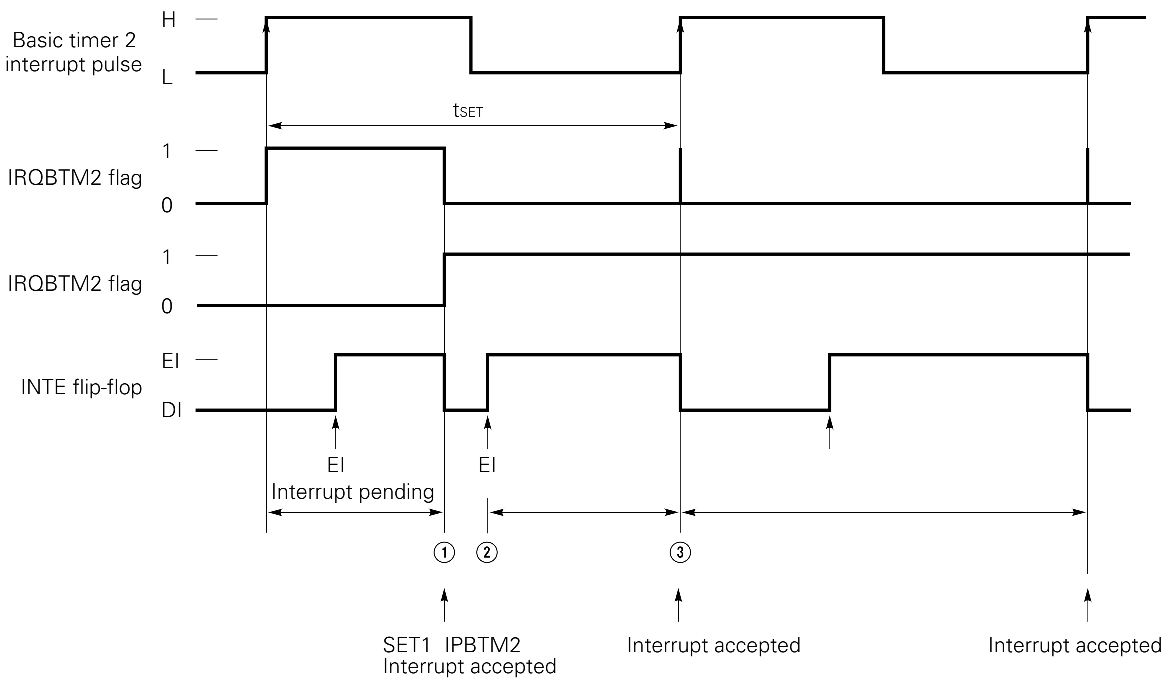
Therefore, a basic timer 2 error occurs only when:

(1) An interrupt request is accepted for the first time after basic timer 2 interrupts are enabled.
(2) An interrupt request is accepted for the first time after a time interval at which the IRQBTM2 flag is set is changed, that is after an interrupt pulse is changed, or
(3) The IRQBTM2 is write-accessed.

Fig. 12-16 shows the timing charts for errors that occur under the above conditions.

Fig. 12-16 Basic Time 2 Error (1/2)

(a) When basic timer 2 interrupts are enabled



When the IPBTM2 flag is set at point ① in the above chart to enable basic timer 2 interrupts, an interrupt request is accepted immediately.

A basic timer 2 error for this case is as follows:

$0 \leq \text{(error)} \leq t_{SET}$

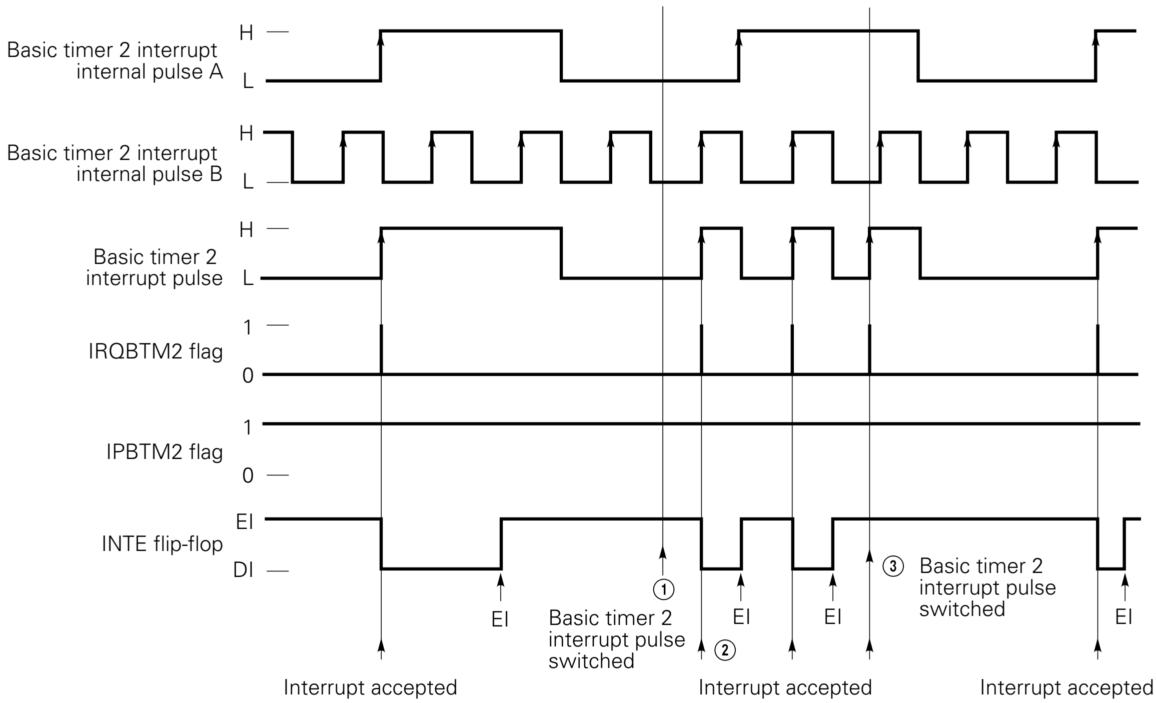
When an EI instruction is issued at point ② to enable interrupts, an interrupt occurs at point ③, where the basic timer 2 interrupt pulse rises.

A basic timer 2 error for this case is as follows:

$0 \leq \text{(error)} \leq t_{SET}$

Fig. 12-16 Basic Time 2 Error (2/2)

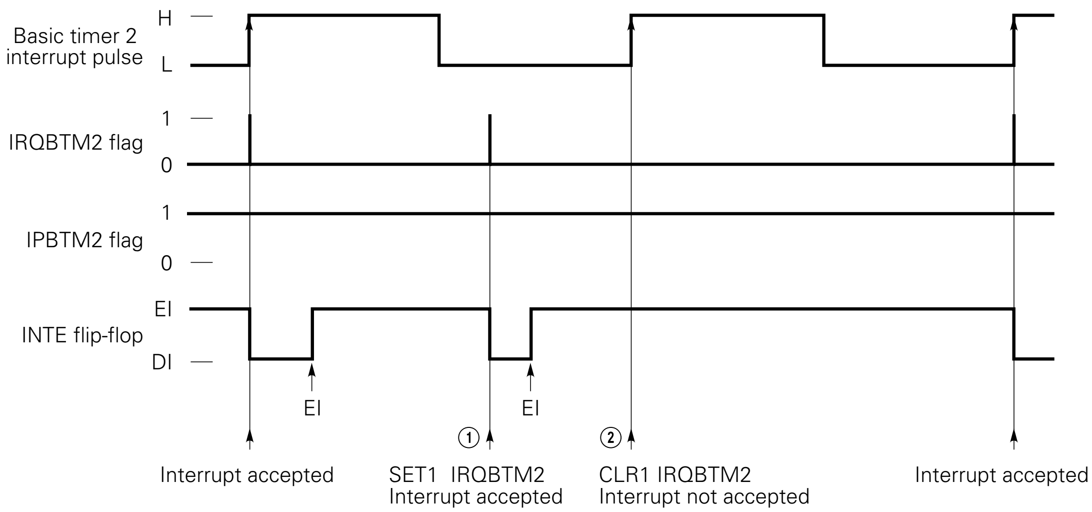
(b) When the basic timer 2 interrupt pulse is switched



Even when the basic timer 2 interrupt pulse is switched from A to B at point ① in the above chart, an interrupt request is not accepted at point ② because the basic timer 2 interrupt pulse does not rise.

When the basic timer 2 interrupt pulse is switched from B to A at point ③, an interrupt request is accepted immediately, because the basic timer 2 interrupt pulse rises.

(c) When the IRQBTM2 flag is manipulated



When the IRQBTM2 flag is set to "1" at point ① in the above chart, an interrupt request is accepted immediately.

If the IRQBTM2 flag is reset to "0" at point ② at the same time the basic timer 2 interrupt pulse rises, an interrupt is not accepted.

12.4.5 Cautions for Using Basic Timer 2

When basic timer 2 is used in a program that runs at constant intervals once power is supplied (after power-on reset), such as a clock program, the program must complete interrupt handling related to basic timer 2 within a certain period of time.

This is explained below, using an example.

Example

```
M1        MEM     0.10H                      ; 1 ms counter
TIMER     DAT     0006H                      ; Interrupt vector address symbol definition

          BR      START                      ; Branches to START.
ORG       TIMER                              ; Program address (0006H)
          ADD     M1, #1010B                 ; Add 1010B to the M1 contents.
          SKT1    CY                         ; Clock processing if a carry occurs
          BR      EI_RETI                    ; Makes a return if there is no carry.
        ; ①
          [Clock processing]
EI_RETI :
          EI
          RETI
START   :
          CLR2    BTM0CK1, BTM0CK0
                                             ; Built-in macro
                                             ; Specifies that the BTM0CY flag is set at
                                             ; intervals of 100 ms.
          CLR4    BTM2EXCK, BTM2ZX, BTM2CK1, BTM2CK0
                                             ; Built-in macro
                                             ; Set the basic timer 2 interrupt time to
                                             ; 100 ms.
          SET1    IPBTM2                     ; Enables basic timer 2 interrupts.
          EI                                 ; Enables all interrupts.

LOOP:
          [Process A]
          BR      LOOP
```

In the above example, the clock processing is repeated at intervals of 1 second, while process A is performed.

As shown in Fig. 12-17 (a), when the CE pin goes from a low to a high, a CE reset occurs at the same time the BTM0CY flag set pulse rises.

If a basic timer 2 interrupt is requested at the same time the BTM0CY flag is set, a CE reset takes preference.

When a CE reset occurs, it automatically resets a basic timer 2 interrupt request (IRQBTM2), hence failing to perform timer processing for one cycle.

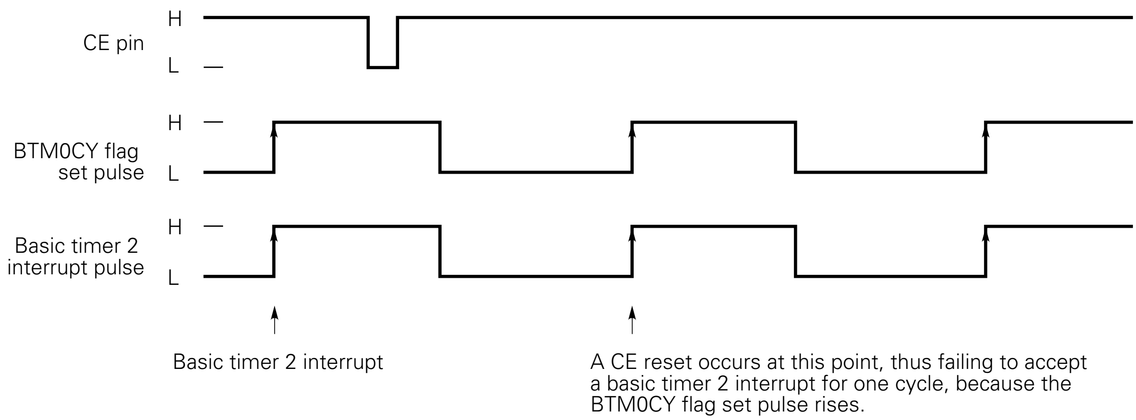
To solve this problem, as shown in Fig. 12-17 (b), a gap in time is provided between when the BTM0CY flag set pulse rises and when the basic timer 2 interrupt pulse rises. In this example, the clock processing is completed with 550 μs in order to eliminate a possibility that the occurrence of a CE reset prohibits acceptance of a basic timer 2 interrupt request.

In other words, you should complete your basic timer 2 interrupt handling within 550 μs, if you want to enable basic timer 2 interrupts even at a CE reset. If 125 μs has been selected as the basic timer 2 interrupt time interval, however, the interrupt handling must be completed within 75 μs.

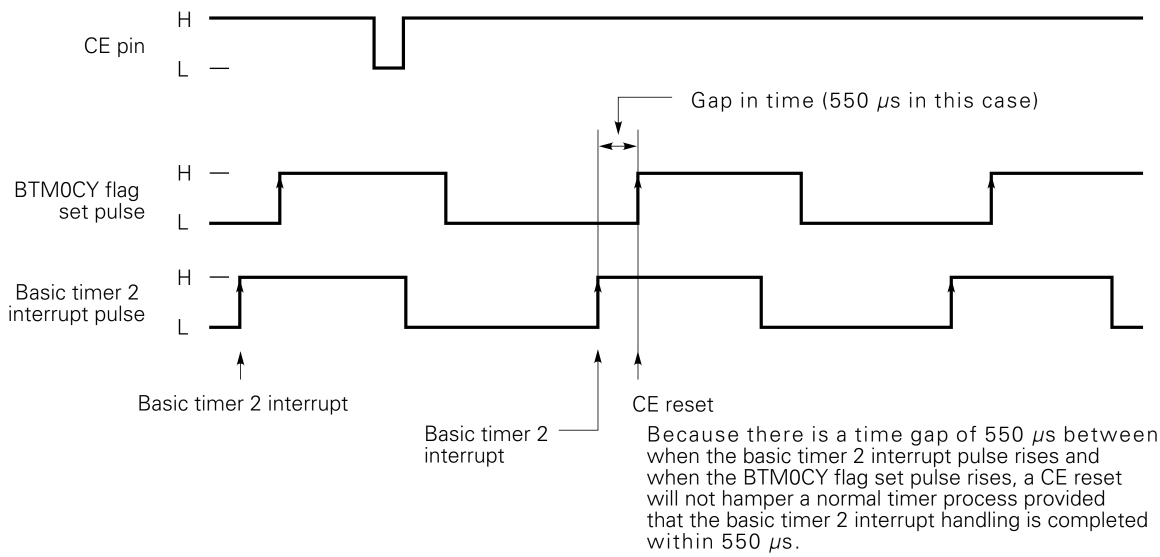
A gap in time is also provided between the BTM0CY flag set pulse and the BTM1CY flag set pulse.

Fig. 12-17 Timing Chart

(a)



(b)



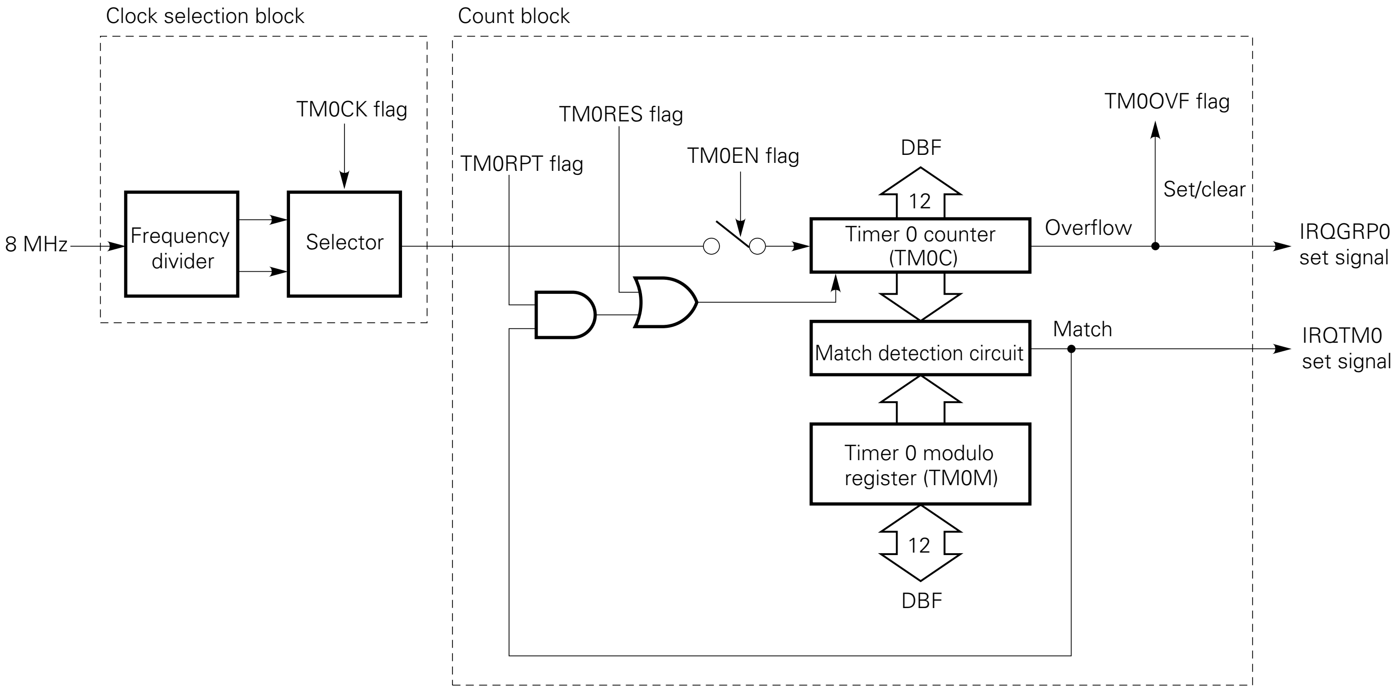
12.5 TIMER 0

12.5.1 Overview of Timer 0

Fig. 12-18 shows the block diagram of timer 0.

Timer 0 is realized by dividing the frequency of the basic clock (100 kHz or 20 kHz) using the 12-bit counter and by comparing the count in the 12-bit counter with a previously set value.

Fig. 12-18 Block Diagram of Timer 0



Remarks 1. TM0CK (bit 0 of timer 0 clock select register; see **Fig. 12-19**) specifies the basic clock frequency.

2. TM0EN (bit 0 of timer 0 control register; see **Fig. 12-20**) specifies whether to start or stop timer 0.

3. TM0RES (bit 1 of timer 0 control register; see **Fig. 12-20**) specifies whether to reset the timer 0 counter.

4. TM0RPT (bit 2 of timer 0 control register; see **Fig. 12-20**) selects the modulo count mode or free-run count mode.

5. TM0OVF (bit 0 of timer 0 overflow register; see **Fig. 12-21**) detects when the timer 0 counter overflows.

12.5.2 Clock Selection Block

The clock selection block specifies the basic clock pulse used to run the timer 0 counter.
Two types of pulses can be selected as the basic clock pulse using the TM0CK flag.
Fig. 12-19 shows the configuration and function of the timer 0 clock select register.

Fig. 12-19 Configuration of the Timer 0 Clock Select Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Timer 0 clock select register	0	0	0	TM0CK	09H	R/W

TM0CK (b_0):

	Specifies the basic clock for timer 0.
0	100 kHz (10 μs)
0	20 kHz (50 μs)

b_3, b_2, b_1: Fixed to "0"

Upon reset		b_3	b_2	b_1	b_0
	Power-on	0	0	0	0
	Clock stop	↓	↓	↓	0
	CE	↓	↓	↓	Hold

12.5.3 Count Block

The count block counts the basic clock pulses using the 12-bit timer 0 counter, and outputs the count. It also issues an interrupt request if the count matches the value in the timer 0 modulo register.

The TM0EN flag specifies whether to start or stop the basic clock pulse supplied to the timer 0 counter.

The TM0RES flag can reset the timer 0 counter.

The timer 0 counter is not automatically reset when its content matches the value in the timer modulo register.

The TM0RPT flag selects the modulo count or free-run count mode.

In the free-run count mode, when the content of the timer 0 counter matches the content of the timer modulo register, timer 0 counter continues to be incremented without being reset.

In the module count mode, when the content of the timer 0 counter matches the content of the timer modulo register, timer 0 counter continues to be incremented after reset.

The TM0OVF flag can be used to detect when the counter overflows. It issues an interrupt request when an overflow occurs.

The timer 0 counter can be read- and write-accessed through the data buffer.

Fig. 12-20 shows the configuration and function of the timer 0 control register.

Fig. 12-21 shows the configuration and function of the timer 0 overflow register.

Figs. 12-22 and 12-23 show the configuration of the timer 0 counter and timer 0 modulo register, respectively.

Fig. 12-20 Configuration of the Timer 0 Control Register

Register	Flag symbol b_3	b_2	b_1	b_0	Address	Read/write
Timer 0 control register	0	TM0RPT	TM0RES	TM0EN	0DH	R/W (for bit 0, W only)

TM0EN	Specifies whether to start or stop timer 0.
0	Stop.
1	Start.

TM0RES	Resets the timer 0 counter.
0	Do not reset.
1	Reset.

TM0RPT	Selects a timer 0 mode.
0	Free-run count mode
1	Modulo count mode

b_3	Fixed to "0"

Upon reset		b_3	b_2	b_1	b_0
	Power-on	0	0	0	0
	Clock stop	↓	0	0	0
	CE	↓	Hold		

Fig. 12-21 Configuration of the Timer 0 Overflow Register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Timer 0 overflow register	0	0	0	TM0OVF	0EH	R

b0 (TM0OVF): Detects when the timer 0 counter overflows.

0	No overflow
1	Overflow

b3 to b1: Fixed to "0"

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	0
CE	↓	↓	↓	Hold

Fig. 12-22 Configuration of the Timer 0 Counter

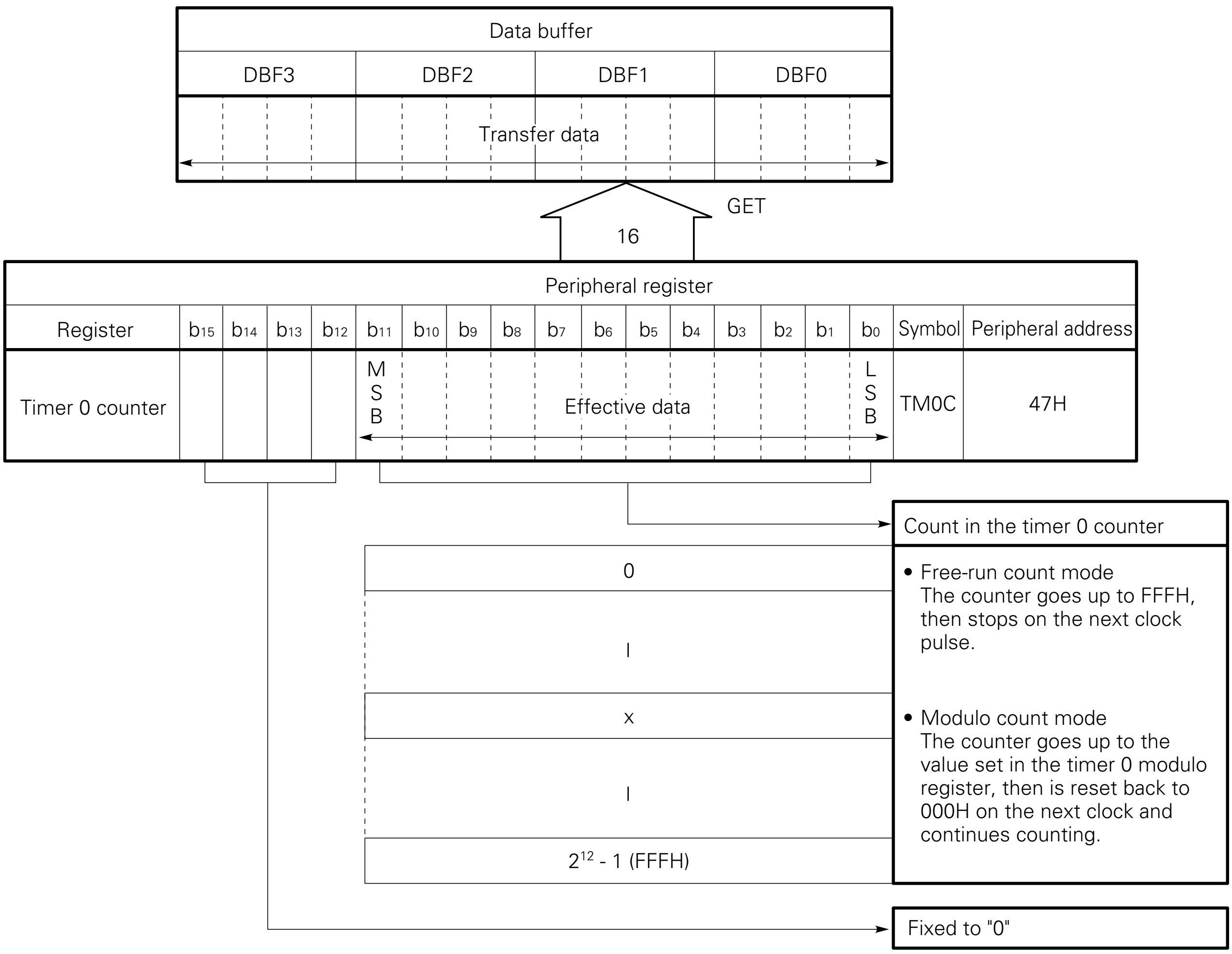
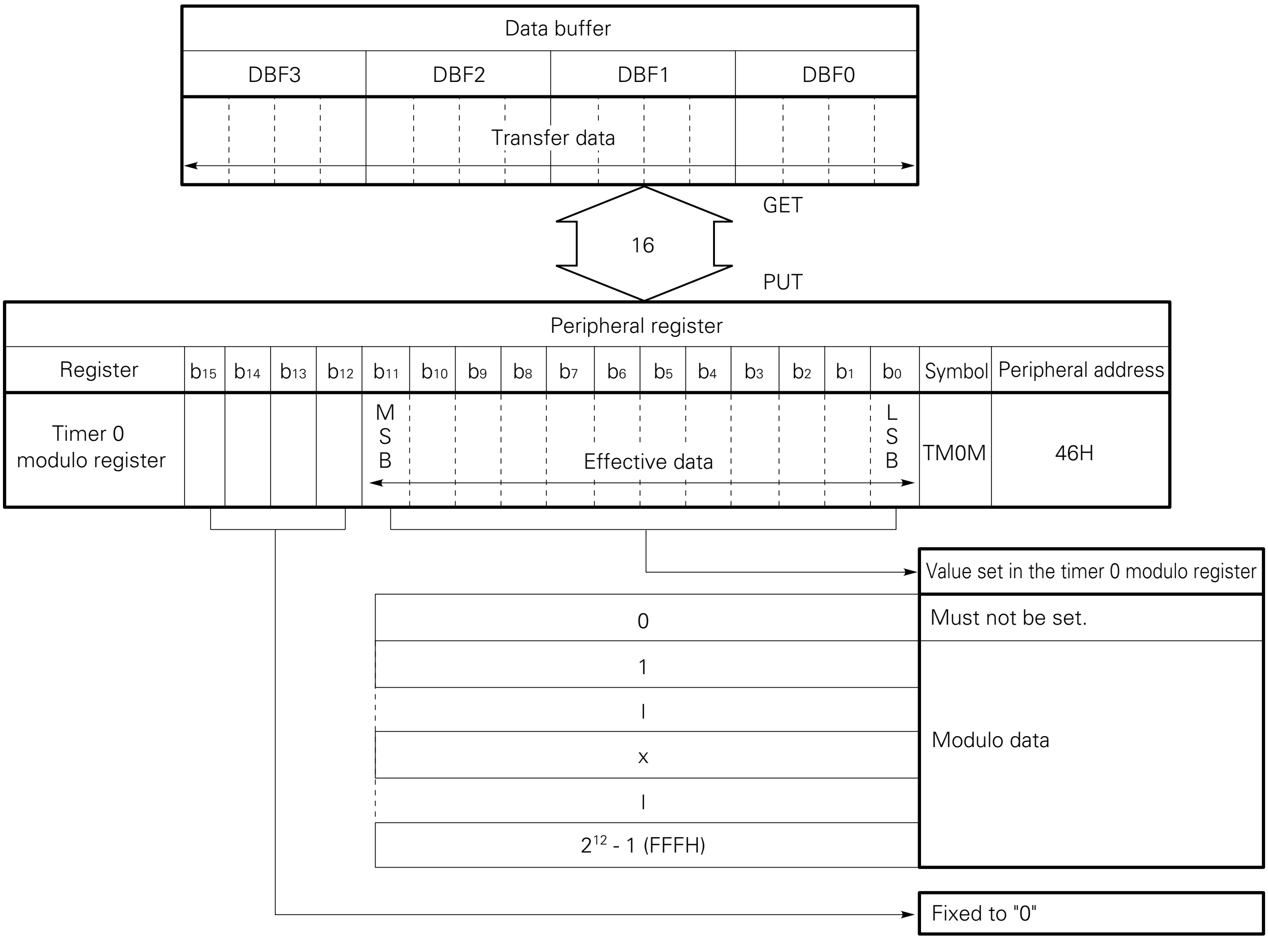


Fig. 12-23 Configuration of the Timer 0 Modulo Register



12.5.4 Example of Using Timer 0

Example 1. Module count mode

```
        BR      START
ORG     0004H
TMINT:
        [Process A]
        EI
        RETI
START:
        CLR1    TMCK0 ;Specifies 10 μs as the count clock.
        MOV     DBF2, #50 SHR 8 AND 0FH
        MOV     DBF1, #50 SHR 4 AND 0FH
        MOV     DBF0, #50 AND 0FH
        PUT     TM0M, DBF
        SET1    IPTM0
        EI
        SET3    TM0RPT, TM0RES, TM0EN
LOOP:
        [Main process]
        BR      LOOP
```

This program performs process A at intervals of 500 μs. Process A must be completed within 500 μs.

Example 2. Free-run count mode

```
                BR      START
                     ⋮
     START:
                CLR1    TMCK0 ;Specifies 10 μs as the count clock.
                CLR1    TM0RPT
                SET2    TM0RES, TM0EN
                [Process A]
                SKF1    TM0OVF
                BR      Overflow detected
                GET     DBF, TM0C

                     ⋮
     Overflow detected:
                     ⋮
```

This program measures the time required to perform process A. It can measure a period of time ranging from 10 μs to 40950 μs. (This program cannot measure a period longer than 40950 μs. A branch should be made to another routine.)

The above sample program can be used to measure the pulse width of a remote control signal.

The modulo count mode is convenient in issuing interrupt requests at constant intervals, while the free-run count mode is useful to measure total time.

12.5.5 Time Interval Error in Timer 0

Timer 0 encounters up to one basic pulse's worth of time interval error under the following conditions.

(1) When the TM0EN flag is manipulated

When the TM0EN flag is set, 0 to +1 pulse's worth of error occurs.
When the TM0EN flag is reset, 0 to –1 pulse's worth of error occurs.

(2) When the counter is reset during operation

When the counter is reset, 0 to +1 pulse's worth of error occurs.

(3) When the basic clock is switched during operation of the counter

0 to +1 newly selected pulse's worth of error occurs.

12.5.6 Cautions for Using Timer 0

A timer 0 interrupt request may occur simultaneously with other timer interrupt requests or a CE reset.
If it is necessary to update the timer even at a CE reset, do not use timer 0. Use basic timer 0, instead.

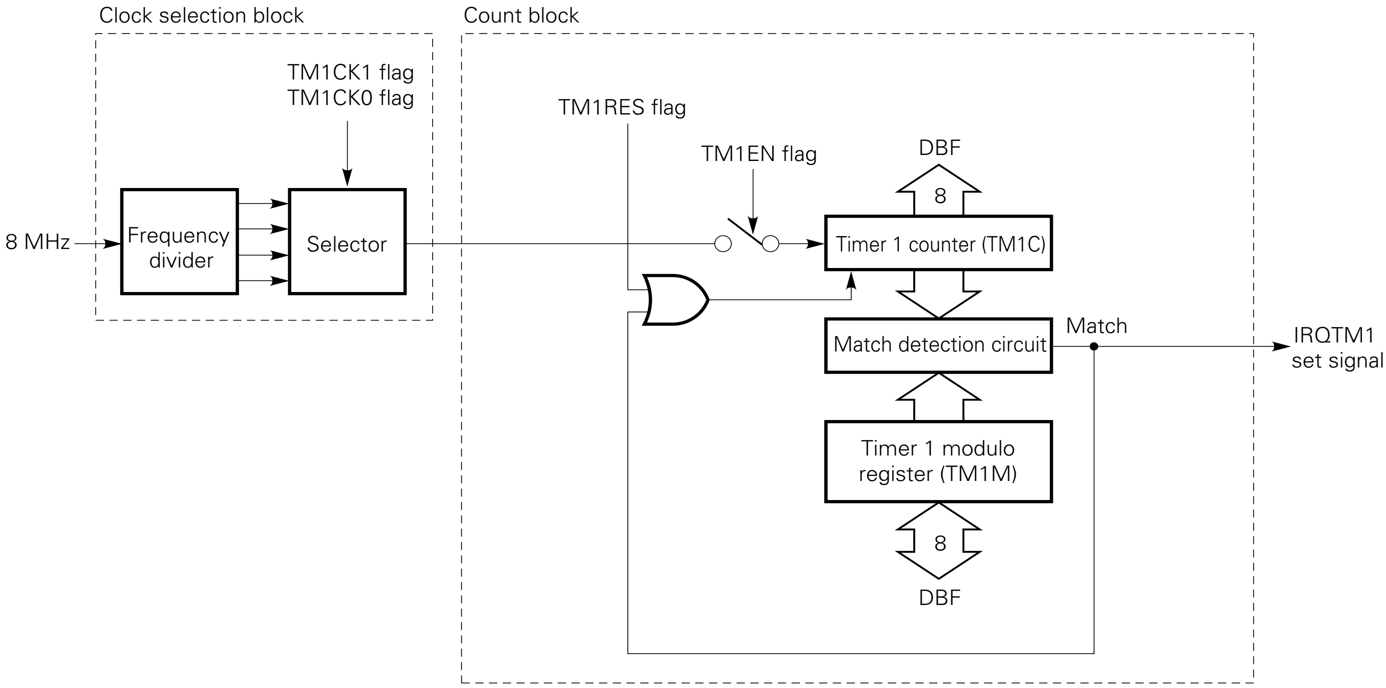
12.6 TIMER 1

12.6.1 Overview of Timer 1

Fig. 12-24 shows the block diagram of timer 1.

Timer 1 is realized by dividing the frequency of the basic clock (100 kHz, 10 kHz, 20 kHz, or 1 kHz) using the 8-bit counter and by comparing the count in the 8-bit counter with a previously set value.

Fig. 12-24 Block Diagram of Timer 1



Remarks 1. TM1CK1 and TM1CK0 (bits 1 and 0 of timer 1 clock select register; **Fig. 12-25**) specify the basic clock frequency.

2. TM1EN (bit 0 of timer 1 control register; **Fig. 12-26**) specifies whether to start or stop timer 1.

3. TM1RES (bit 1 of timer 1 control register; **Fig. 12-26**) specifies whether to reset the timer 1 counter.

12.6.2 Clock Selection Block

The clock selection block specifies the basic clock pulse used to run the timer 1 counter.

Four types of pulses can be selected as the basic clock pulse using the TM1CK0 and TM1CK1 flags.

Fig. 12-25 shows the configuration and function of the timer 1 clock select register.

Fig. 12-25 Configuration of the Timer 0 Clock Select Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Timer 1 clock select register	0	0	TM1CK1	TM1CK0	1AH	R/W

TM1CK1, TM1CK0: Specifies the basic clock for timer 1.

TM1CK1	TM1CK0	Specifies the basic clock for timer 1.
0	0	1 kHz (1 ms)
0	1	10 kHz (100 μs)
1	0	20 kHz (50 μs)
1	1	100 kHz (10 μs)

b_3, b_2: Fixed to "0"

Upon reset		b_3	b_2	b_1	b_0
Upon reset	Power-on	0	0	0	0
	Clock stop	↓	↓	0	0
	CE	↓	↓	Hold	

12.6.3 Count Block

The count block counts the basic clock pulses using the 8-bit timer 1 counter, and outputs the count. It also issues an interrupt request if the count matches the value in the timer 1 modulo register.

The TM1EN flag can start or stop the basic clock pulse supplied to the timer 1 counter.

The TM1RES flag can reset the timer 1 counter.

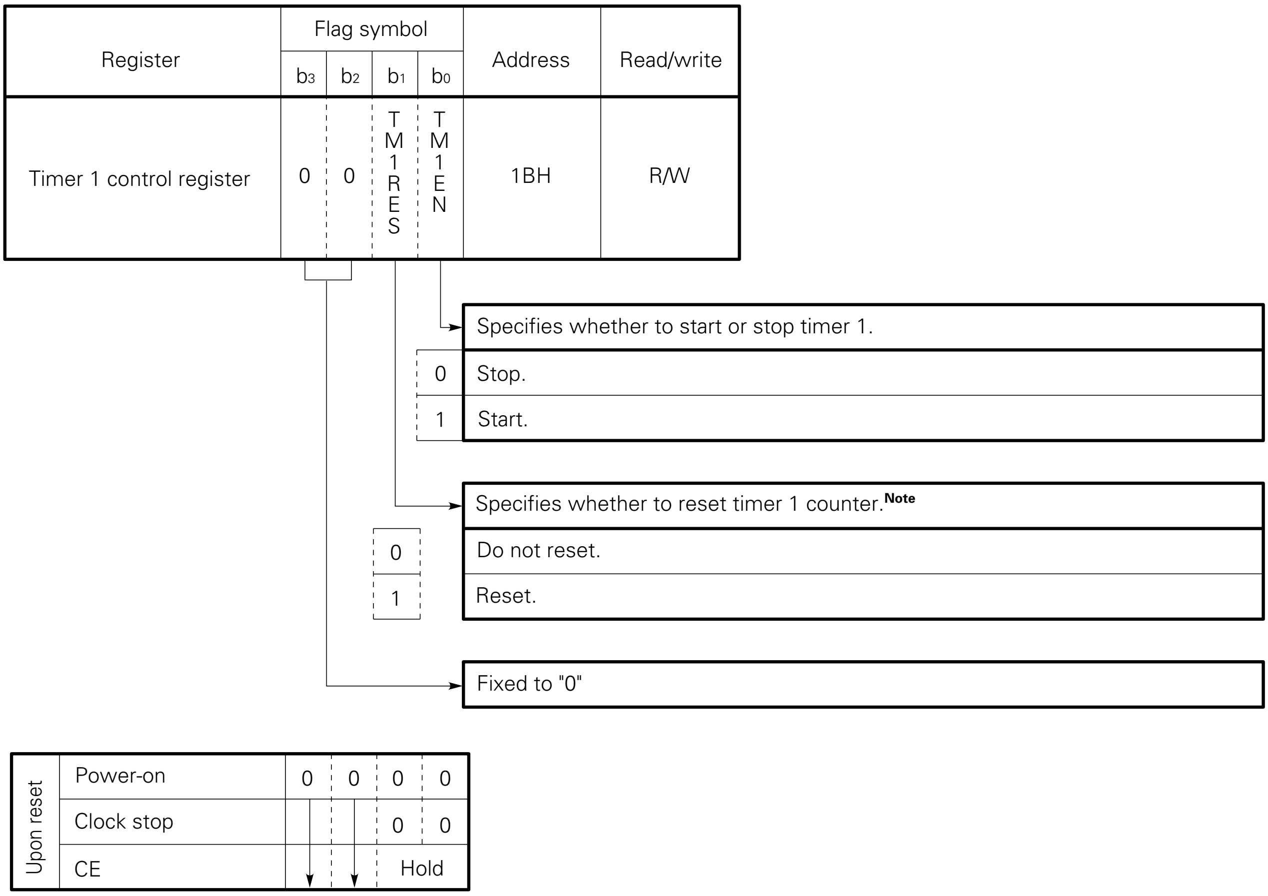
The timer 1 counter is not automatically reset when its content matches the value in the timer 1 modulo register.

The timer 1 counter can be read- and write-accessed through the data buffer.

Fig. 12-26 shows the configuration and function of the timer 1 control register.

Figs. 12-27 and 12-28 show the configuration and function of the timer 1 counter and timer 1 modulo register, respectively.

Fig. 12-26 Configuration of the Timer 1 Control Register



Note This is effective only at write access. "0" is always read out.

Fig. 12-27 Configuration of the Timer 1 Counter

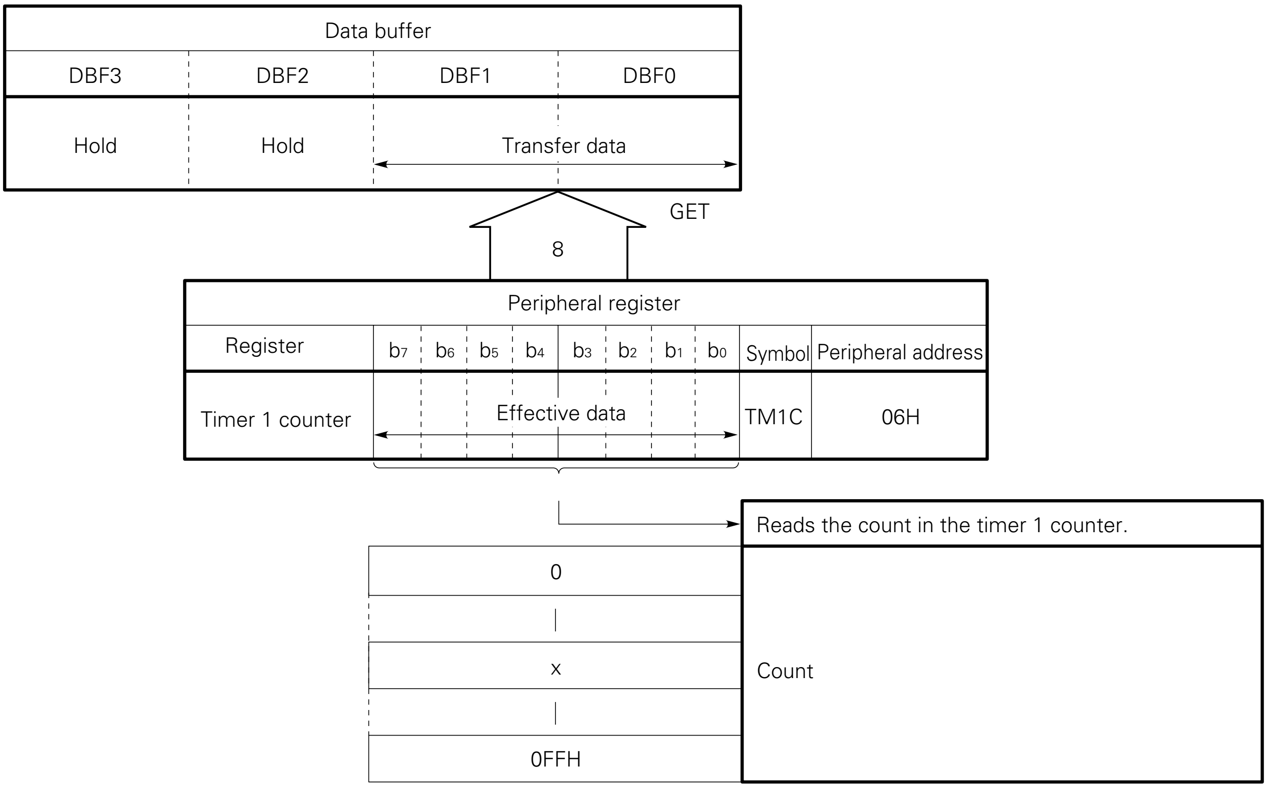
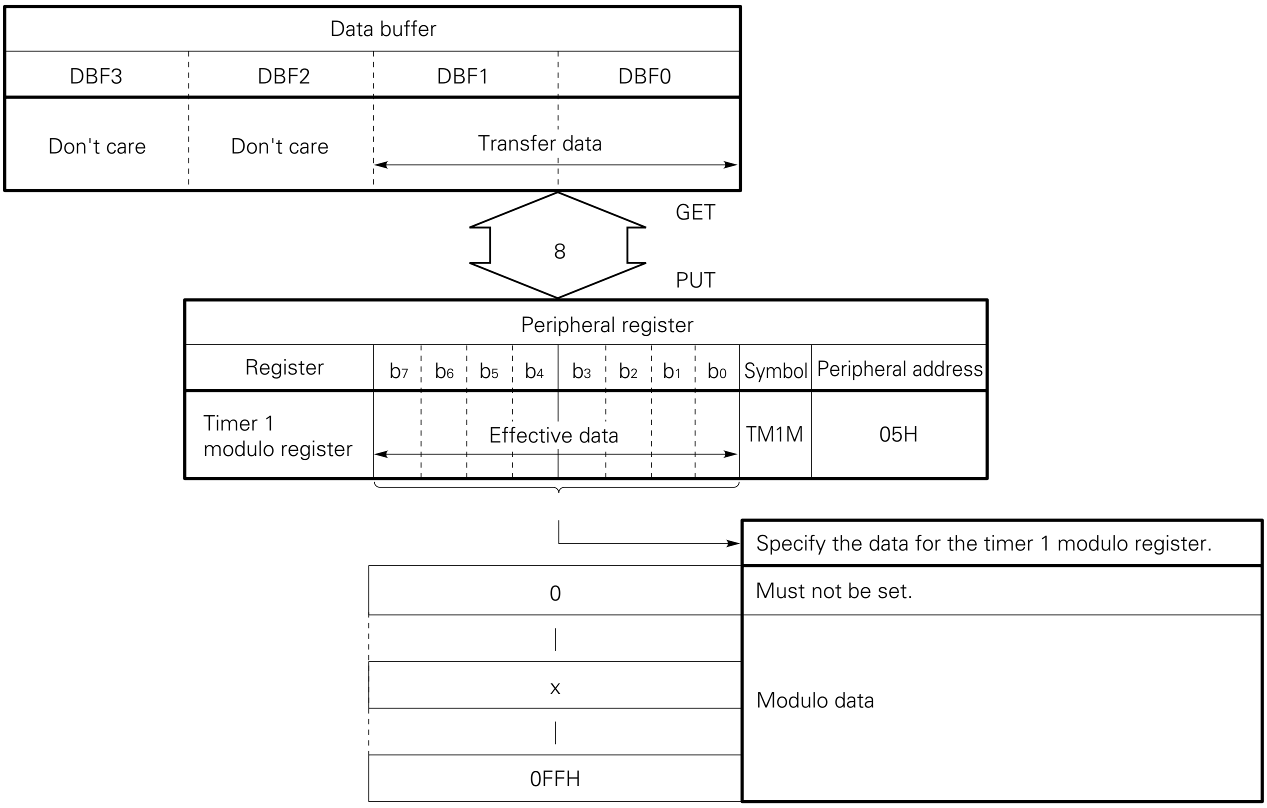


Fig. 12-28 Configuration of the Timer 1 Modulo Register



12.6.4 Time Interval Error in Timer 1

Timer 1 encounters up to one basic pulse's worth of time interval error under the following conditions.

(1) When the TM1EN flag is manipulated

When the TM1EN flag is set, 0 to +1 pulse's worth of error occurs.
When the TM1EN flag is reset, 0 to -1 pulse's worth of error occurs.

(2) When the counter is reset during operation

When the counter is reset, 0 to +1 pulse's worth of error occurs.

(3) When the basic clock is switched during operation of the counter

0 to +1 newly selected pulse's worth of error occurs.

12.6.5 Cautions for Using Timer 1

A timer 1 interrupt request may occur simultaneously with other timer interrupt requests or a CE reset. If it is necessary to update the timer even at a CE reset, do not use timer 1. Use basic timer 0, instead.

12.7 CLOCK TIMER

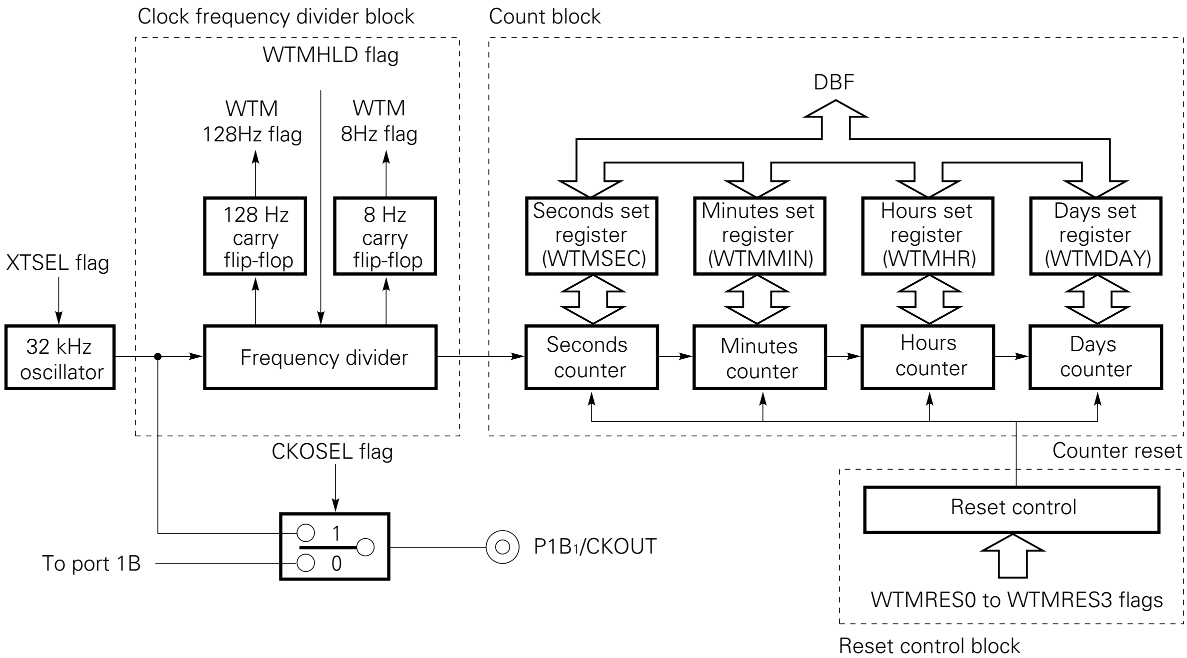
12.7.1 Overview of the Clock Timer

Fig. 12-29 shows the block diagram of the clock timer.

The clock timer is realized by counting seconds, minutes, hours, and days using a clock pulse and by reading out the counts. The clock pulse is generated by dividing a frequency of 32 kHz.

The clock timer can also be used as an ordinary timer by detecting a flip-flop that is set at 128 or 8 Hz by software.

Fig. 12-29 Block Diagram of the Clock Timer



Remarks 1. XTSEL (bit 0 of the clock timer mode register; see **Fig. 12-32**) selects the function of the P0D$_0$/ADC$_1$/XT$_{OUT}$ and P0D$_1$/ADC$_2$/XT$_{IN}$ pins.

2. CKOSEL (bit 1 of the clock timer mode register; see **Fig. 12-32**) specifies whether to output the clock timer adjustment oscillator.

3. WTM128HZ (bit 0 of the clock timer 128 Hz carry register; see **Fig. 12-30**) detects the state of the 128 Hz carry flip-flop.

4. WTM8HZ (bit 0 of the clock timer 8 Hz carry register; see **Fig. 12-31**) detects the state of the 8 Hz carry flip-flop.

5. WTMHLD (bit 3 of the clock timer mode register; see **Fig. 12-32**) specifies whether to hold the clock timer.

6. WTMRES0 to WTMRES3 (bits 0 to 3 of the clock timer reset register; see **Fig. 12-36**) specifies whether to reset the clock timer.

12.7.2 Clock Frequency Divider Block

The clock frequency block divides a frequency of 32 kHz to generate a clock pulse used for the clock timer.

The clock frequency block can also detect the WTM128HZ flag (bit 0 of the clock timer 128 Hz carry register) and the WTM8HZ flag (bit 0 of the clock timer 8 Hz carry register).

Setting (1) the WTMHLD flag of the clock timer mode register can hold the clock output at a point where 500 ms worth of frequency division is completed.

Figs. 12-30 and 12-31 show the configuration and function of the clock timer 128 Hz carry register and clock timer 8 Hz carry register.

Fig. 12-32 shows the configuration and function of the clock timer mode register.

Fig. 12-30 Configuration of the Clock Timer 128 Hz Carry Register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Clock timer 128 Hz carry register	0	0	0	WTM128HZ	1EH	R & Reset

b0: Detects the state of the 128 Hz carry flip-flop.

0	Reset.
1	Set.

b3, b2, b1: Fixed to "0"

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	Hold
CE	↓	↓	↓	Hold

Fig. 12-31 Configuration of the Clock Timer 8 Hz Carry Register

Register	Flag symbol				Address	Read/write
	b3	b2	b1	b0		
Watch timer 8 Hz carry register	0	0	0	WTM8HZ	1DH	R & Reset

WTM8HZ: Detects the state of the 8 Hz carry flip-flop.

0	Reset.
1	Set.

b3, b2, b1: Fixed to "0"

Upon reset		b3	b2	b1	b0
	Power-on	0	0	0	0
	Clock stop	↓	↓	↓	Hold
	CE				Hold

Fig. 12-32 Configuration of the Clock Timer Mode Register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Clock timer mode register	WTMHLD	0	CKOSEL	XTSEL	06H	W (R/W for bit 0 only)

XTSEL — Selects the function of the $P0D_0/ADC_1/XT_{OUT}$ and $P0D_1/ADC_2/XT_{IN}$ pins.

XTSEL	
0	Operation as a port
1	Operation as a connection pin for the clock timer oscillator

CKOSEL — Specifies whether to output the clock timer adjustment oscillator.

CKOSEL	
0	No output.
1	Output from the $P1B_1$/CKOUT.

b2 — Fixed to "0"

WTMHLD — Specifies whether to hold the clock timer.

WTMHLD	
0	Do not hold.
1	Hold.

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	Hold	↓	0	Hold
CE	Hold	↓	0	Hold

12.7.3 Count Block

The count block counts clock pulses, which are generated by the clock frequency division block, using four 8-bit counters.

The clock timer consists of a seconds counter (sexagesimal), minutes counter (sexagesimal), hours counter (24-ary), days counter (septinary).

These counters are reset by the reset control block (see **Section 12.7.4**).

Each counter can be write- and read-accessed through the data buffer.

Figs. 12-33 to 12-35 show the configuration and function of the registers to control the count block.

Fig. 12-33 Configuration of the Seconds and Minutes Set Registers

Data buffer			
DBF3	DBF2	DBF1	DBF0
Don't care	Don't care	Transfer data	

GET
8
PUT

Peripheral register									
Register	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Symbol	Peripheral address
Seconds set register	0	0	Effective data						WTMSEC	1AH
Minutes set register	0	0	Effective data						WTMMIN	1BH

	Counts in the seconds and minutes counters
0 | x | 59 (3BH)	Incremented at every second (or every minute for the minutes counter) with a carry generated at every 60 seconds (or every 60 minutes for the minutes counter)--sexagesimal counters. • At read access Current count in each counter • At write access Each counter is initialized. The counters must be reset before a write access.
60 (3CH) | 63 (3FH)	Values which do not exist at a read access. If any of these values is written to the counters, their contents become undefined.

b_7, b_6: Fixed to "0"

Fig. 12-34 Configuration of the Hours Set Register

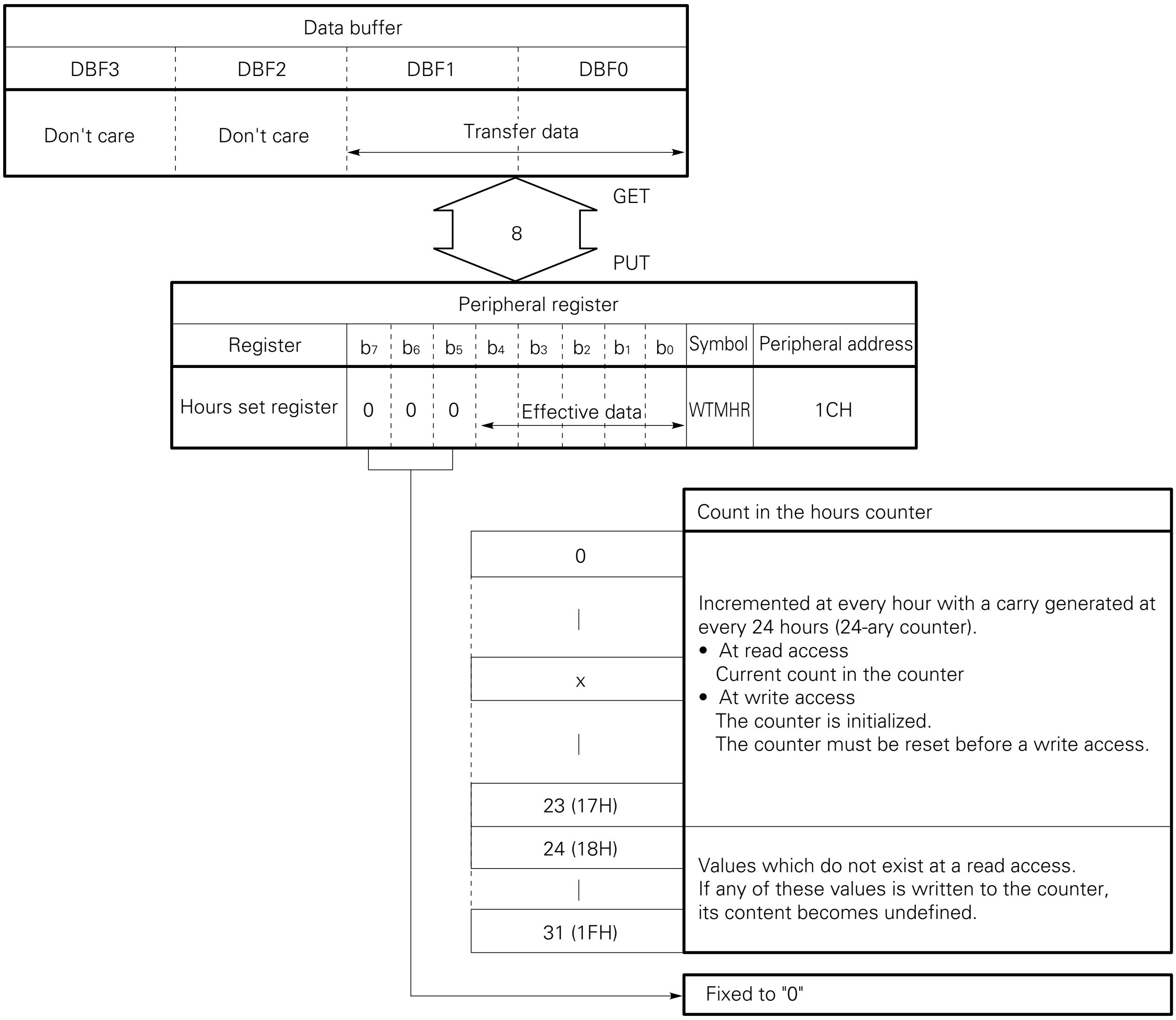
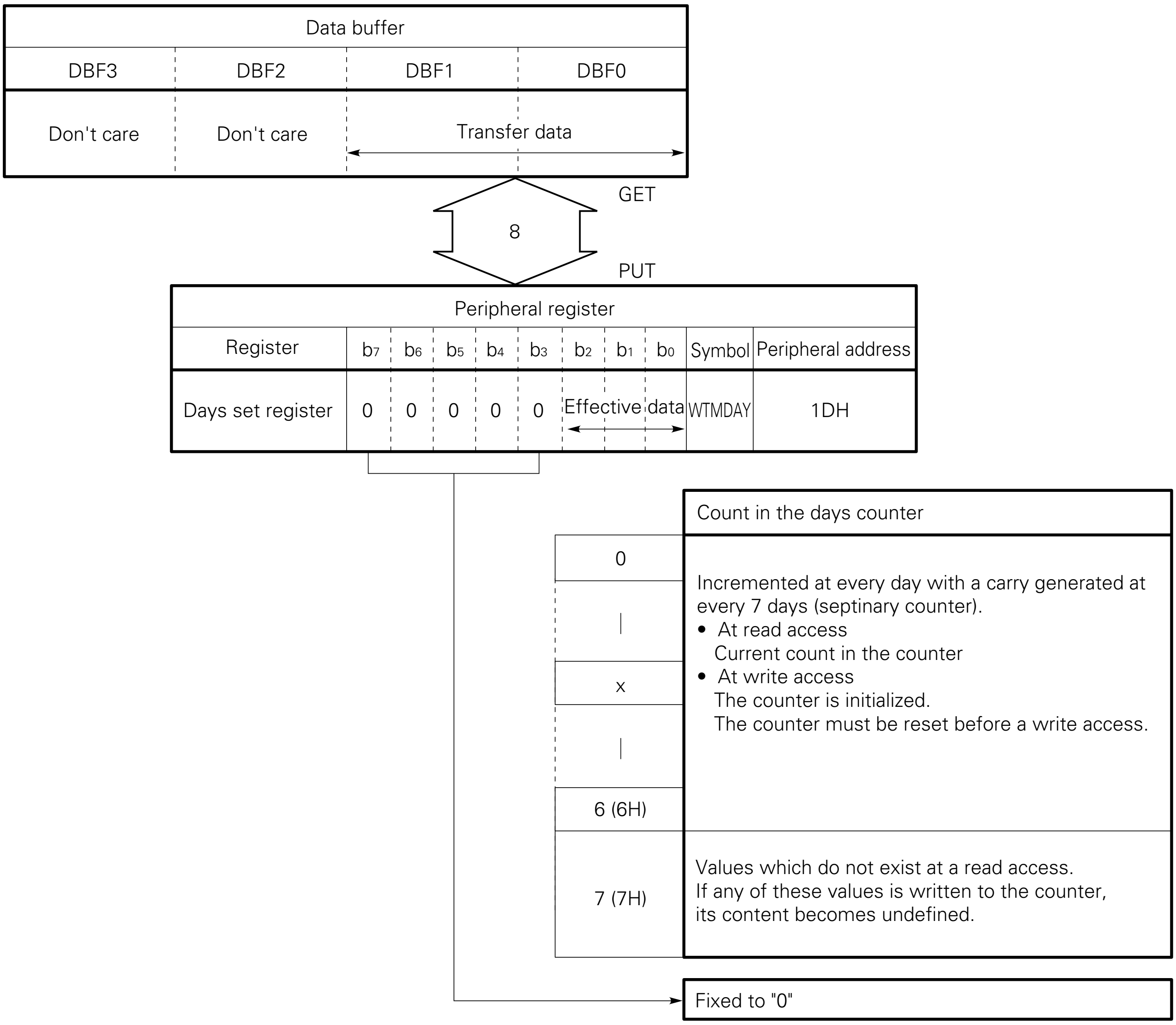


Fig. 12-35 Configuration of the Days Set Register



12.7.4 Reset Control Block

The clock timer reset register specifies whether to set or reset the clock timer.

Fig. 12-36 shows the configuration and function of the clock timer reset register.

Fig. 12-36 Configuration of the Clock Timer Reset Register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Clock timer reset register	WTMRES3	WTMRES2	WTMRES1	WTMRES0	14H	R/W

WTMRES0: Specifies whether to reset the clock timer basic clock, seconds, minutes, hours, and days counters.

WTMRES0	
0	Do not reset.
1	Reset.

WTMRES1: Specifies whether to reset the clock timer basic clock.

WTMRES1	
0	Do not reset.
1	Reset.

WTMRES2: Specifies whether to reset the seconds, minutes, and hours counters.

WTMRES2	
0	Do not reset.
1	Reset.

WTMRES3: Specifies whether to reset the days counter.

WTMRES3	
0	Do not reset.
1	Reset.

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	Hold			
CE	Hold			

12.7.5 32 kHz Oscillator and Oscillation Frequency Adjustment

The 32 kHz oscillator generates a 32 kHz clock pulse for the clock timer.

When using the clock timer, set the XTSEL flag of the clock timer mode register to specify the $P0D_0/ADC_1/XT_{OUT}$ and $P0D_1/ADC_2/XT_{IN}$ pins as the clock timer oscillator connection pins.

The 32 kHz clock pulse is supplied at the $P1B_1$/CKOUT pin for oscillation frequency adjustment. To use this pin for oscillation frequency adjustment output, set the CKOSEL flag of the clock timer mode register.

See Fig. 12-32 for the configuration and function of the clock timer mode register.

12.7.6 Cautions for Using the Clock Timer

(1) Rewriting the counts in the counters

When you want to rewrite the contents of the seconds, minutes, hours, and days counters, previously reset them.

If the contents of these counters have not been reset, a normal value cannot be written to the counters when the count, such as seconds, minutes, hours, or days, is a non-zero or a nonexisting value (for example, 61 in the seconds counter).

(2) Reset control

While the WTMRESn (n = 0 to 3) is “1”, the counter corresponding to the flag will not work.

If you want to reset any of these counters, first set the corresponding WTMRES flag to “1” and reset the counter, then reset the flag to “0” again.

13. A/D CONVERTER

13.1 OUTLINE OF A/D CONVERTER

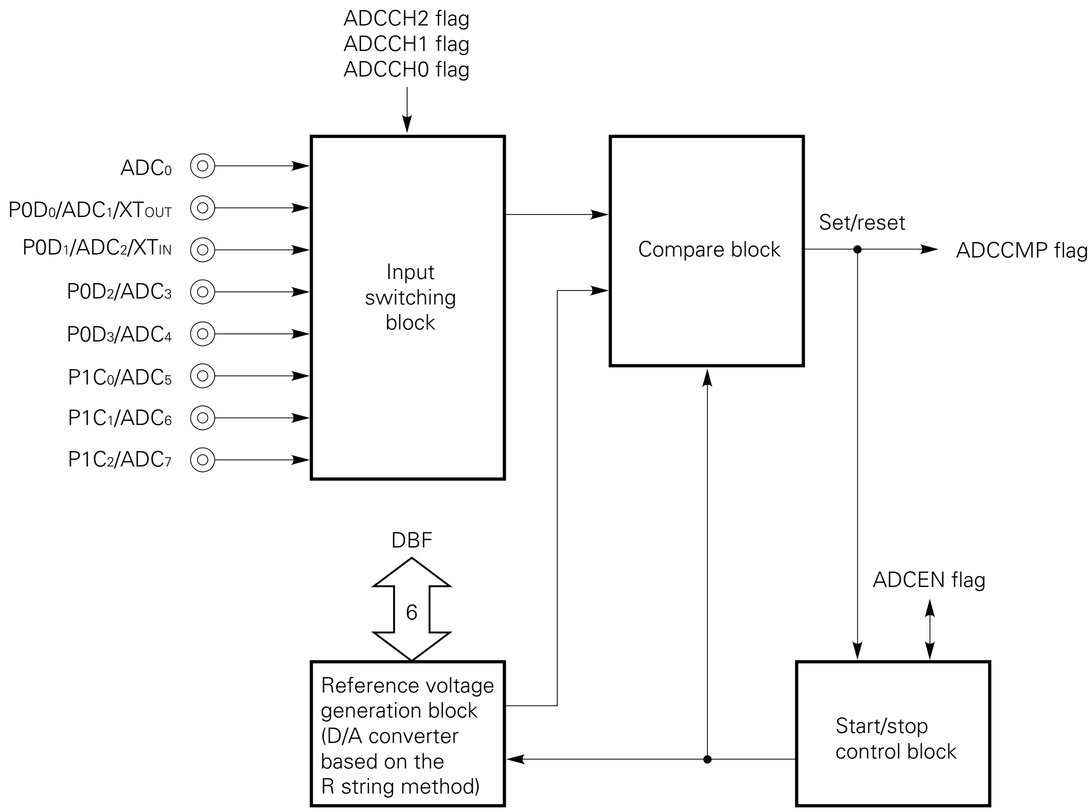
Fig. 13-1 outlines the A/D converter.

The A/D converter compares an analog voltage applied to the ADC_0-ADC_7 pins with an internal reference voltage, then converts the analog voltage to a 6-bit digital signal by evaluating the result of comparison with software.

The result of comparison is detected using the ADCCMP flag.

The successive approximation system is used for comparison.

Fig. 13-1 Schematic Diagram of A/D Converter



Remarks 1. ADCCH0-2 (A/D converter channel select register bits 0 to 2): Select a pin used for the A/D converter. (See **Fig. 13-3.**)

2. ADCCMP (A/D converter control register bit 0): Detects the result of comparison. (See **Fig. 13-5.**)

3. ADCEN (A/D converter control register bit 3): Detects comparison execution and comparison status. (See **Fig. 13-5.**)

13.2 INPUT SWITCHING BLOCK

Fig. 13-2 shows the configuration of the input switching block.

The input switching block selects a pin according to the setting of the A/D converter channel select register.

If the $P1C_0/ADC_5$-$P1C_2/ADC_7$ pins are set as general-purpose output ports at this time, output signals are output on these pins and the $P1C_3$ pin. When any of these pins is to be used for the A/D converter, the pin must be set as an input port by using the port 1C group I/O select register. In this case, the pins not used for the A/D converter and the $P1C_3$ pin are set as general-purpose input ports.

Only one pin can be used for the A/D converter at a time.

Port 0D contains a pull-down resistor. However, the pull-down resistor is turned off when the port is selected for the A/D converter. At this time, the pull-down resistors of the pins not selected remain on.

Fig. 13-3 shows the format and function of the A/D converter channel select register.

Fig. 13-2 Configuration of Input Switching Block

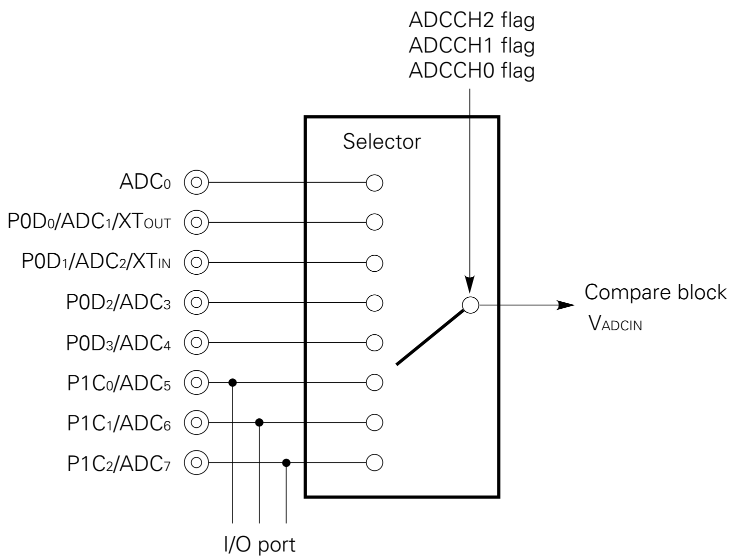


Fig. 13-3 Format of A/D Converter Channel Select Register

Register	Flag symbol b3	Flag symbol b2	Flag symbol b1	Flag symbol b0	Address	Read/write
A/D converter channel select register	0	ADCCH2	ADCCH1	ADCCH0	21H	R/W

b3: Fixed to 0

ADCCH2	ADCCH1	ADCCH0	Selects a pin used for the A/D converter.
0	0	0	ADC_0 pin
0	0	1	$P0D_0/ADC_1/XT_{OUT}$ pin
0	1	0	$P0D_1/ADC_2/XT_{IN}$ pin
0	1	1	$P0D_2/ADC_3$ pin
1	0	0	$P0D_3/ADC_4$ pin
1	0	1	$P1C_0/ADC_5$ pin
1	1	0	$P1C_1/ADC_6$ pin
1	1	1	$P1C_2/ADC_7$ pin

Upon reset		b3	b2	b1	b0
	Power-on	0	0	0	0
	Clock stop		0	0	0
	CE		*	*	*

* = Hold

13.3 COMPARE VOLTAGE GENERATION BLOCK AND COMPARE BLOCK

Fig. 13-4 shows the configuration of the reference voltage generation block and compare block.

According to the 6-bit data set in the A/D converter data register, the reference voltage generation block switches between tap decoders to generate 64 types of reference voltage V_{REF}.

This means that the reference voltage generation block is a D/A converter based on the R string method.

The power supply for the R string method has the same potential as V_{DD} of the device.

The compare block compares the voltage V_{ADCIN} applied to a pin with the reference voltage V_{REF} to determine which is larger.

Fig. 13-5 and Fig. 13-6 show the formats and functions of the A/D converter control register flags and A/D converter data register. Table 13-1 provides a list of reference voltages.

Fig. 13-4 Configuration of Reference Voltage Generation Block and Compare Block

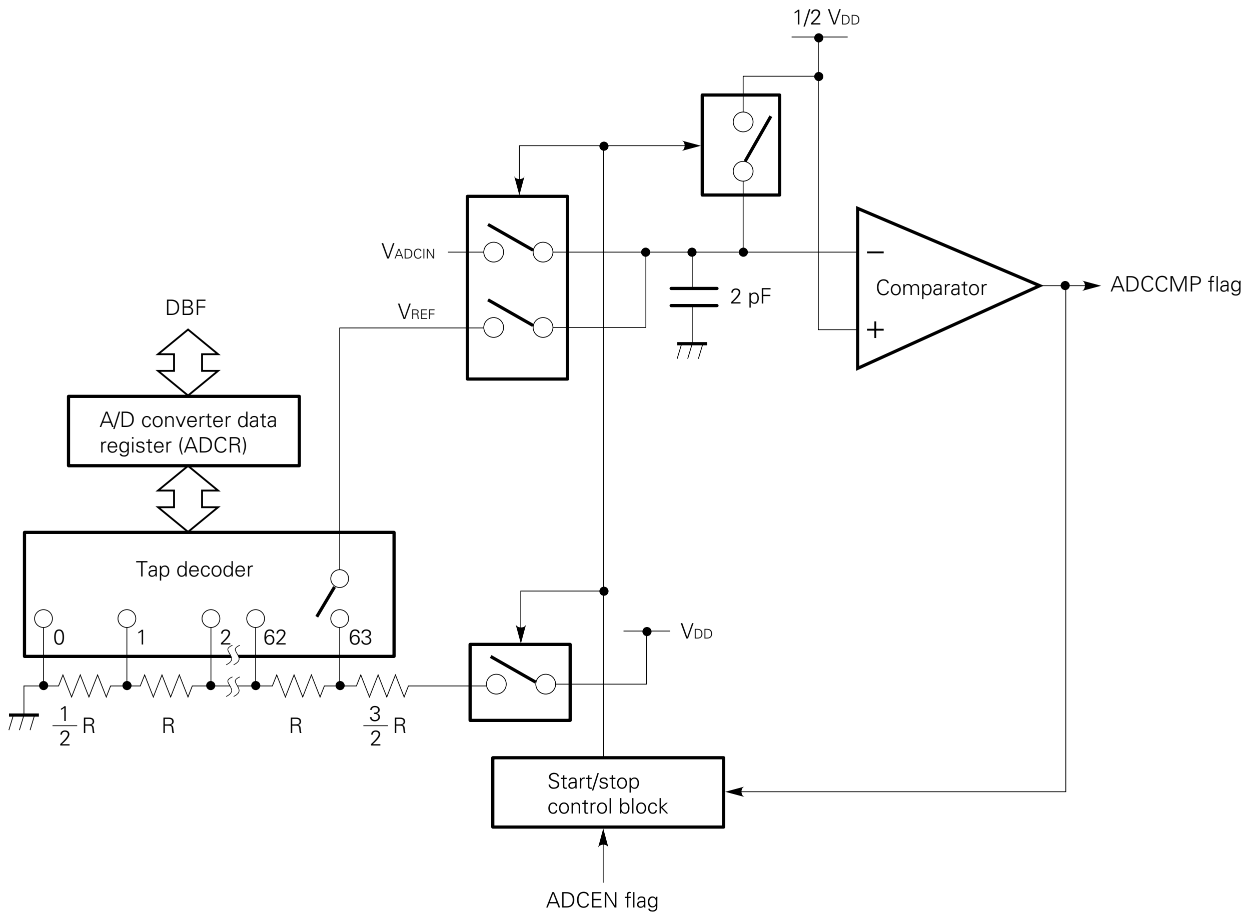


Fig. 13-5 Format of A/D Converter Control Register

Register	Flag symbol b3	Flag symbol b2	Flag symbol b1	Flag symbol b0	Address	Read/write
A/D converter control register	ADCEN	0	0	ADCCMP	24H	R/W R for bit 0 only

ADCCMP (b0): Detects the result of comparison by the A/D converter.

ADCCMP	
0	$V_{ADCIN} < V_{REF}$
1	$V_{ADCIN} > V_{REF}$

b2, b1: Fixed to 0

ADCEN (b3): Detects the comparison execution and comparison status of the A/D converter.

ADCEN	In write operation	In read operation
0	No change	Comparison completed
1	Comparison executed	Comparison in progress

Upon reset		b3	b2	b1	b0
	Power-on	0	0	0	*
	Clock stop	0	↓	↓	**
	CE	0	↓	↓	**

* = Undefined
** = Hold

Fig. 13-6 Format of A/D Converter Data Register

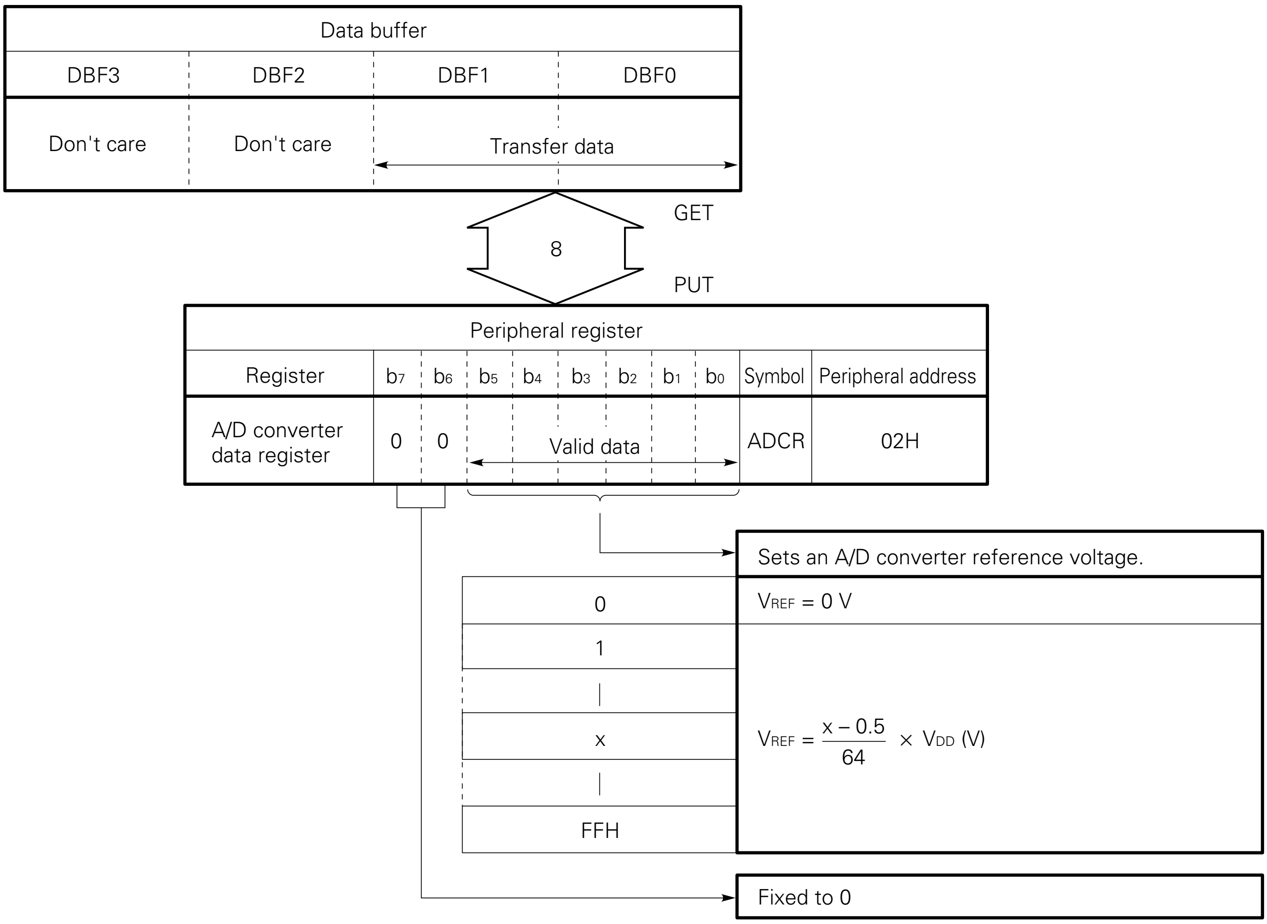


Table 13-1 Values in A/D Converter Data Register and Reference Voltages

Data set in ADCR		Reference voltage		Data set in ADCR		Reference voltage	
Decimal (DEC)	Hexadecimal (HEX)	Logical voltage Unit: $\times V_{DD}$ V	When V_{DD} = 5 V Unit: V	Decimal (DEC)	Hexadecimal (HEX)	Logical voltage Unit: $\times V_{DD}$ 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	14.5/65	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

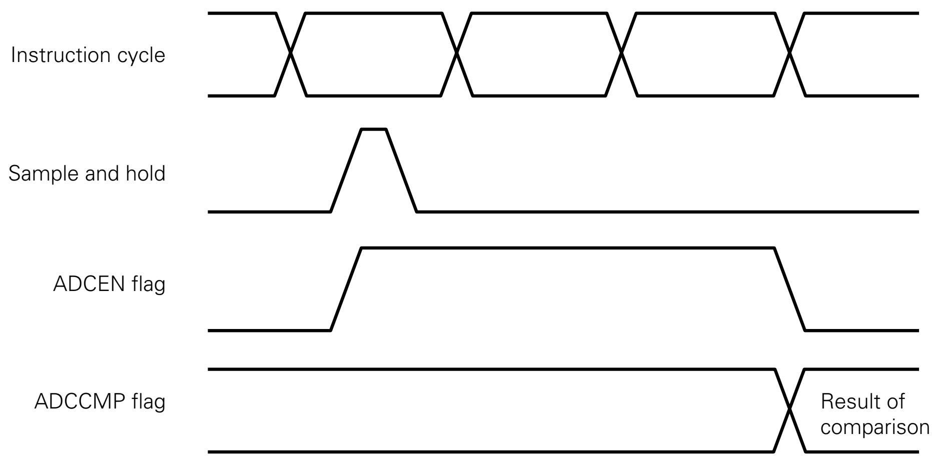
13.4 COMPARE TIMING CHART

When a compare operation is completed, the ADCEN flag is automatically reset to 0.

This means that the ADCEN flag is detected after the ADCEN flag is set, and the result of comparison (ADCCMP flag) is read when the ADCEN flag is reset. Accordingly, the time required for one compare operation is three-instruction execution time (6 μs).

Fig. 13-7 indicates the timing chart.

Fig. 13-7 Timing of A/D Converter Compare Operation



13.5 A/D CONVERTER PERFORMANCE

Table 13-2 indicates the performance of the A/D converter.

Table 13-2 Performance of A/D Converter

Item	Performance
Resolution	6 bits
Input voltage range	0-V_{DD}
Quantization error	$\pm\frac{1}{2}$ LSB
Over-range	$\frac{62.5}{64} \times V_{DD}$
Errors associated with offset, gain, nonlinearity, etc.	$\pm\frac{3}{2}$ LSB Note

Note A quantization error is included.

13.6 USING A/D CONVERTER

13.6.1 Comparison with One Reference Voltage

An example of program is described below.

Example A comparison is made between the input voltage V_{ADCIN} applied to the ADC_0 pin and the reference voltage V_{REF} ((31.5/64) × V_{DD}).

When $V_{ADCIN} > V_{REF}$, a branch to AAA occurs. When $V_{ADCIN} < V_{REF}$, a branch to BBB occurs.

```
INIT:
          ADCR7    FLG    0.0EH.3        ; Dummy
          ADCR6    FLG    0.0EH.2        ; Dummy
          ADCR5    FLG    0.0EH.1        ; Defines each bit of the data buffer as an ADCR data setting flag.
          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

          INITFLG  NOT ADCCH3, NOT ADCCH2, NOT ADCCH1, NOT ADCCH0
                                         ; Sets the ADC0 pin for the A/D converter.
START:
          INITFLG  NOT ADCR3, NOT ADCR2, NOT ADCR1, NOT ADCR0
          INITFLG  NOT ADCR7, NOT ADCR6, NOT ADCR5, NOT ADCR4
          PUT      ADCR, DBF             ; Sets (31.5/64) × VDD as the reference voltage VREF.
          SET1     ADCEN                 ; Starts comparison.
          SKF1     ADCEN                 ; Detects compare operation in progress.
          BR       $ – 1
          SKT1     ADCCMP                ; Detects the ADCCMP flag, then
          BR       AAA                   ; Branches to AAA when the result of comparison is false.
          BR       BBB                   ; Branches to BBB when the result of comparison is true.
```

13.6.2 Successive Approximation Based on the Binary Search Method

In one compare operation, the A/D converter can make a comparison with only one reference voltage.

This means that successive approximation needs to be programmed for conversion of an analog voltage to a digital signal.

If the processing time of a successive approximation program varies from one input voltage to another, processing of other programs may be adversely affected.

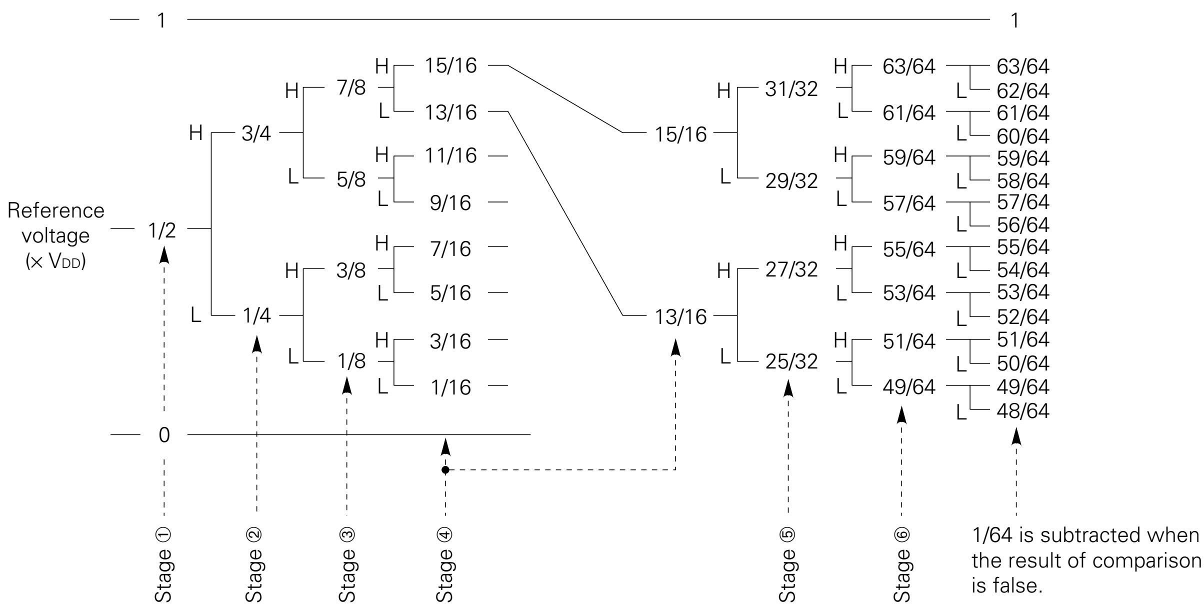
In such a case, the binary search method described in (1) to (3) below is useful.

(1) Concept of binary search

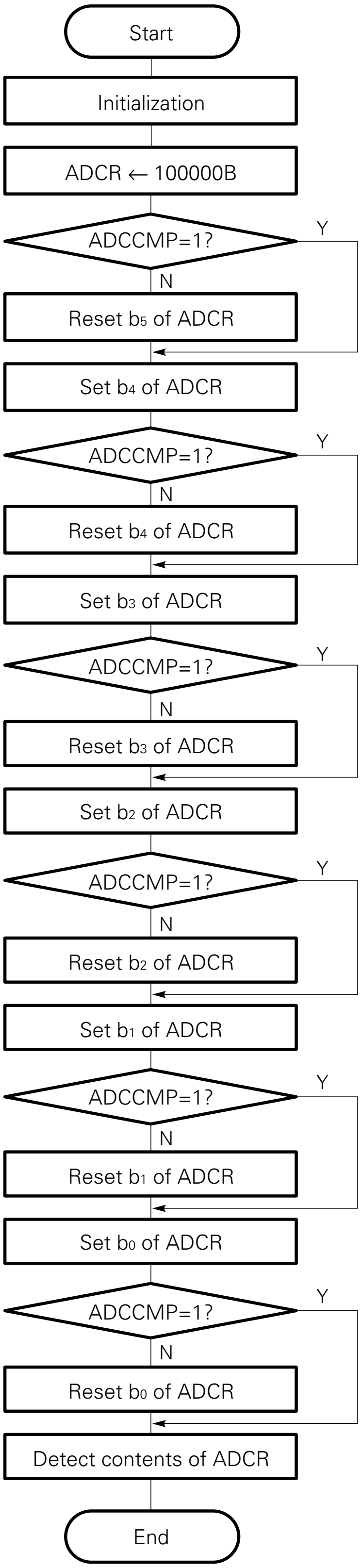
The concept of binary search is described below.

First, $V_{DD}/2$ is set as a reference voltage. Then, a comparison is made by adding a voltage of $V_{DD}/4$ when the result of comparison is true (when a higher level is applied), or subtracting a voltage of $V_{DD}/4$ when the result of comparison is false (when a lower level is applied).

Similarly, comparisons are made sequentially adding or subtracting $V_{DD}/8$, $V_{DD}/16$, $V_{DD}/32$, and $V_{DD}/64$ in this order. If the result of comparison is false at the sixth stage, $V_{DD}/64$ is subtracted finally.



(2) Flowchart of the binary search method



: Sets a pin to be used.

: Sets the reference voltage to $V_{DD}/2$.

: Detects the result of comparison, then

: Subtracts $V_{DD}/2$ from the reference voltage when the flag is set to 0.

: Adds $V_{DD}/4$ to the reference voltage when the flag is set to either 0 or 1.

: Detects the result of comparison, then

: Subtracts $V_{DD}/4$ from the reference voltage when the flag is set to 0.

: Adds $V_{DD}/8$ to the reference voltage when the flag is set to either 0 or 1.

: Detects the result of comparison, then

: Subtracts $V_{DD}/8$ from the reference voltage when the flag is set to 0.

: Adds $V_{DD}/16$ to the reference voltage when the flag is set to either 0 or 1.

: Detects the result of comparison, then

: Subtracts $V_{DD}/16$ from the reference voltage when the flag is set to 0.

: Adds $V_{DD}/32$ to the reference voltage when the flag is set to either 0 or 1.

: Detects the result of comparison, then

: Subtracts $V_{DD}/32$ from the reference voltage when the flag is set to 0.

: Adds $V_{DD}/64$ to the reference voltage when the flag is set to either 0 or 1.

: Detects the result of comparison, then

: Subtracts $V_{DD}/64$ from the reference voltage when the flag is set to 0.

: Ends conversion when the flag is set to 1.

(3) Example of program based on the binary search method

(a) Method with a short conversion time

```
INIT:
        ADCR7   FLG   0.0EH.3      ; Dummy
        ADCR6   FLG   0.0EH.2      ; Dummy
        ADCR5   FLG   0.0EH.1      ; Defines each bit of the data buffer as an ADCR data setting
        ADCR4   FLG   0.0EH.0      ; flag.
        ADCR3   FLG   0.0FH.3      ;
        ADCR2   FLG   0.0FH.2      ;
        ADCR1   FLG   0.0FH.1      ;
        ADCR0   FLG   0.0FH.0      ;

        INITFLG  NOT ADCCH3, NOT ADCCH2, NOT ADCCH1, NOT ADCCH0
                                   ; Sets the ADC0 pin for the A/D converter.
START:
        INITFLG  NOT ADCR3, NOT ADCR2, NOT ADCR1, NOT ADCR0
        INITFLG  NOT ADCR7, NOT ADCR6, NOT ADCR5, NOT ADCR4
        PUT     ADCR, DBF          ; Sets (31.5/64) × VDD as the reference voltage VREF.
        SET1    ADCEN              ; Starts comparison.
        SKF1    ADCEN              ; Detects compare operation in progress.
        BR      $ – 1
        SKT1    ADCCMP             ; Detects the ADCCMP flag, then
        CLR1    ADCR5              ; Subtracts (32/64) × VDD when the flag is set to 0.
        SET1    ADCR4              ; Adds (16/64) × VDD.
        PUT     ADCR, DBF
        SET1    ADCEN              ; Starts comparison.
        SKF1    ADCEN              ; Detects compare operation in progress.
        BR      $ – 1
        SKT1    ADCCMP             ; Detects the ADCCMP flag, then
        CLR1    ADCR4              ; Subtracts (16/64) × VDD when the flag is set to 0.
        SET1    ADCR3              ; Adds (8/64) × VDD.
        PUT     ADCR, DBF
        SET1    ADCEN              ; Starts comparison.
        SKF1    ADCEN              ; Detects compare operation in progress.
        BR      $ – 1
        SKT1    ADCCMP             ; Detects the ADCCMP flag, then
        CLR1    ADCR3              ; Subtracts (8/64) × VDD when the flag is set to 0.
        SET1    ADCR2              ; Adds (4/64) × VDD.
        PUT     ADCR, DBF
        SET1    ADCEN              ; Starts comparison.
        SKF1    ADCEN              ; Detects compare operation in progress.
        BR      $ – 1
```

```
        SKT1    ADCCMP          ; Detects the ADCCMP flag, then
        CLR1    ADCR2           ; Subtracts (4/64) × VDD when the flag is set to 0.
        SET1    ADCR1           ; Adds (2/64) × VDD.
        PUT     ADCR, DBF
        SET1    ADCEN           ; Starts comparison.
        SKF1    ADCEN           ; Detects compare operation in progress.
        BR      $ – 1
        SKT1    ADCCMP          ; Detects the ADCCMP flag, then
        CLR1    ADCR1           ; Subtracts (2/64) × VDD when the flag is set to 0.
        SET1    ADCR0           ; Adds (1/64) × VDD.
        PUT     ADCR, DBF
        SET1    ADCEN           ; Starts comparison.
        SKF1    ADCEN           ; Detects compare operation in progress.
        BR      $ – 1
        SKT1    ADCCMP          ; Detects the ADCCMP flag, then
        CLR1    ADCR1           ; Subtracts (1/64) × VDD when the flag is set to 0.
END:
```

(b) Method with a smaller number of program steps

```
INIT:
        WORKR1  MEM  0.01H      ;
        WORKR0  MEM  0.00H      ;
        INITFLG  NOT ADCCH3, NOT ADCCH2, NOT ADCCH1, NOT ADCCH0
                                ; Sets the ADC0 pin for the A/D converter.
START:
        MOV     DBF1, #0010B
        MOV     DBF0, #0000B
        MOV     WORKR1, #0110B
        MOV     WORKR0, #0000B
        CLR1    CY
LOOP:
        RORC    WORK1
        RORC    WORK0
        SKF1    CY
        BR      END
        PUT     ADCR, DBF       ; Sets (31.5/64) × VDD as the reference voltage VREF.
        SET1    ADCEN           ; Starts comparison.
        SKF1    ADCEN           ; Detects compare operation in progress.
        BR      $ – 1
        SKT1    ADCCMP
        BR      BBB
AAA:
        OR      DBF1, WORK1
```

```
        OR      DBF0, WORK0
        BR      LOOP
BBB:
        EOR     DBF1, WORKR1
        EOR     DBF0, WORKR0
        BR      LOOP
END:
```

13.7 NOTES ON USING A/D CONVERTER

When the $P1C_0/ADC_5$ to $P1C_2/ADC_7$ pins are used for the A/D converter, the pins need to be set as general-purpose input ports with the P1CGIO flag. Port 1C is a group I/O, so that the $P1C_3$ pin can be used only as a general-purpose input port when the A/D converter is used.

13.8 STATES UPON RESET

13.8.1 Power-On Reset

The $P0D_3/ADC_4$, $P0D_2/ADC_3$, $P0D_1/ADC_2/XT_{IN}$, $P0D_0/ADC_1/XT_{OUT}$ pins, and $P1C_2/ADC_7$ to $P1C_0/ADC_5$ pins are set as general-purpose input ports.

13.8.2 Clock Stop

The $P0D_3/ADC_4$, $P0D_2/ADC_3$, $P0D_1/ADC_2/XT_{IN}$, $P0D_0/ADC_1/XT_{OUT}$ pins, and $P1C_2/ADC_7$ to $P1C_0/ADC_5$ pins are set as general-purpose input ports.

13.8.3 CE Reset

The $P0D_3/ADC_4$, $P0D_2/ADC_3$, $P0D_1/ADC_2/XT_{IN}$, $P0D_0/ADC_1/XT_{OUT}$ pins, and $P1C_2/ADC_7$ to $P1C_0/ADC_5$ pins are set as general-purpose input ports.

14. D/A CONVERTER (PWM METHOD)

14.1 OUTLINE OF D/A CONVERTER

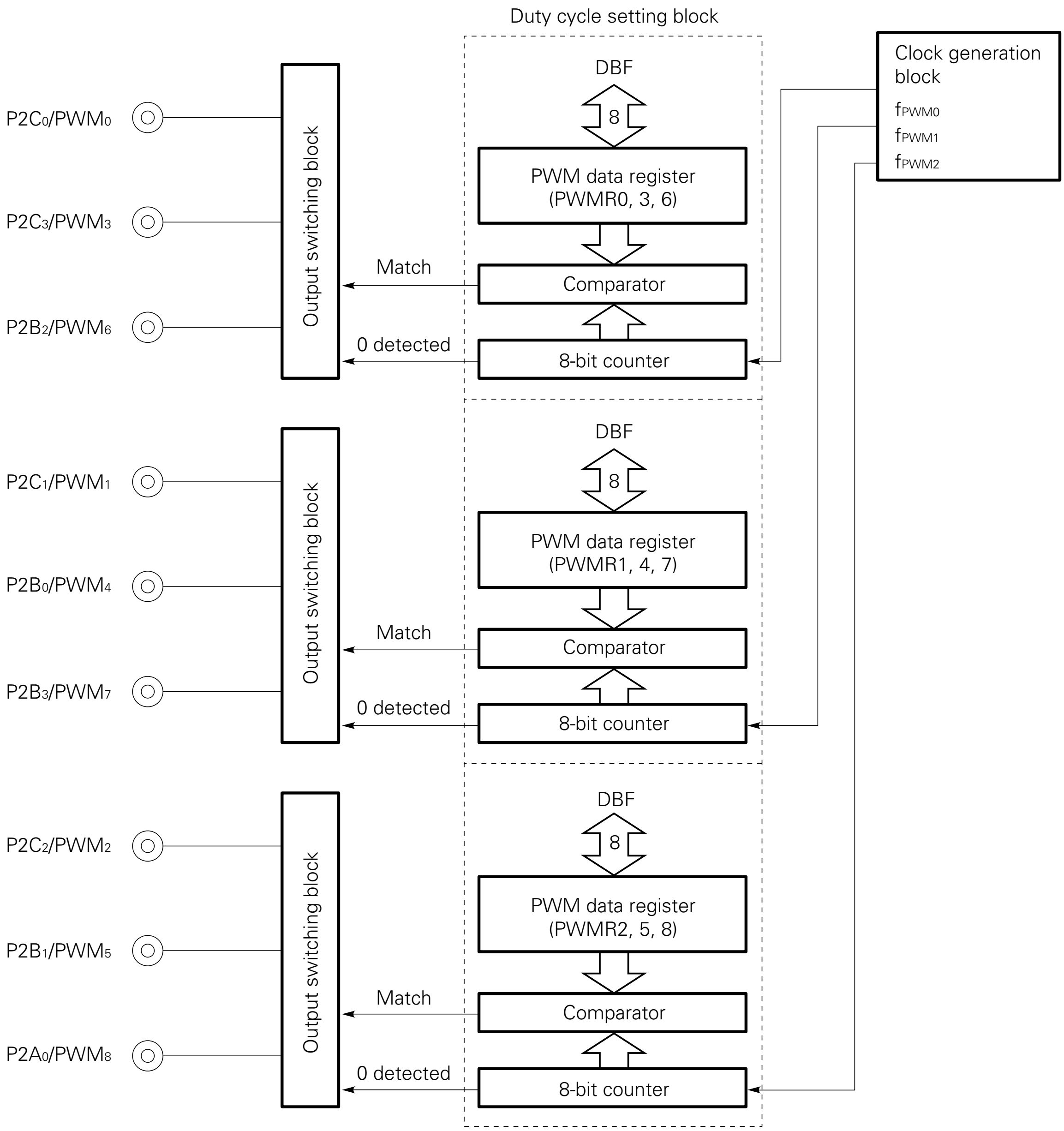
Fig. 14-1 outlines the D/A converter.

The D/A converter outputs a signal according to the pulse width modulation (PWM) method, which allows a variable duty cycle. By attaching an external low-pass filter, a digital signal can be converted to an analog signal.

A signal with a variable duty cycle is output on each pin independently.

The output frequency is 1953 Hz, and the duty cycle can be changed in 256 steps.

Fig. 14-1 Schematic Diagram of D/A Converter



14.2 OUTPUT SWITCHING BLOCK

According to the PWM0SEL-PWM8SEL flags of PWM mode select registers 1 to 3, the output switching block determines whether each output pin of the D/A converter is to be used for the D/A converter or as a general-purpose output port.

Fig. 14-2 shows the configuration of the output switching block. Fig. 14-3 through Fig. 14-5 show the formats and functions of PWM mode select registers 1 to 3.

Whether to use a pin for the D/A converter or as a general-purpose output port can be set independently of other pins.

Each output pin is an N-ch open drain output, so that a pull-up resistor is externally required.

Fig. 14-2 Configuration of Output Switching Block

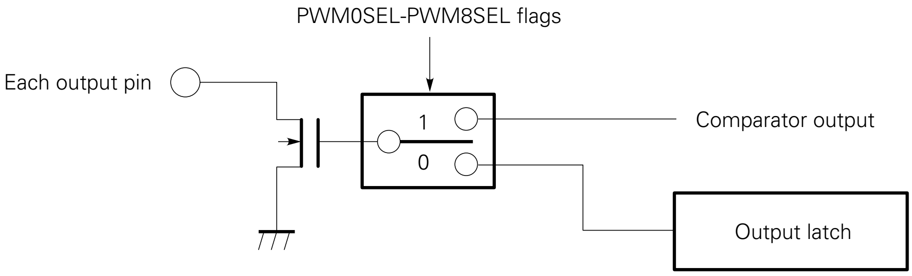


Fig. 14-3 Format of PWM Mode Select Register 3

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
PWM mode select register 3	0	0	0	PWM8SEL	03H	R/W

b0 (PWM8SEL):

	Sets the function of $P2A_0/PWM_8$ pin.
0	General-purpose output port
1	D/A converter

b3–b1: Fixed to 0

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	0
CE	↓	↓	↓	Hold

Fig. 14-4 Format of PWM Mode Select Register 2

Register	Flag symbol b3	Flag symbol b2	Flag symbol b1	Flag symbol b0	Address	Read/write
PWM mode select register 2	PWM7SEL	PWM6SEL	PWM5SEL	PWM4SEL	04H	R/W

PWM4SEL (b0):

PWM4SEL	Sets the function of P2B0/PWM4 pin.
0	General-purpose output port
1	D/A converter

PWM5SEL (b1):

PWM5SEL	Sets the function of P2B1/PWM5 pin.
0	General-purpose output port
1	D/A converter

PWM6SEL (b2):

PWM6SEL	Sets the function of P2B2/PWM6 pin.
0	General-purpose output port
1	D/A converter

PWM7SEL (b3):

PWM7SEL	Sets the function of P2B3/PWM7 pin.
0	General-purpose output port
1	D/A converter

Upon reset		b3	b2	b1	b0
	Power-on	0	0	0	0
	Clock stop	0	0	0	0
	CE	Hold			

Fig. 14-5 Format of PWM Mode Select Register 1

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
PWM mode select register 1	PWM3SEL	PWM2SEL	PWM1SEL	PWM0SEL	05H	R/W

PWM0SEL	Sets the function of $P2C_0/PWM_0$ pin.
0	General-purpose output port
1	D/A converter

PWM1SEL	Sets the function of $P2C_1/PWM_1$ pin.
0	General-purpose output port
1	D/A converter

PWM2SEL	Sets the function of $P2C_2/PWM_2$ pin.
0	General-purpose output port
1	D/A converter

PWM3SEL	Sets the function of $P2C_3/PWM_3$ pin.
0	General-purpose output port
1	D/A converter

Upon reset		b_3	b_2	b_1	b_0
	Power-on	0	0	0	0
	Clock stop	0	0	0	0
	CE	Hold			

14.3 DUTY CYCLE SETTING BLOCK

The duty cycle setting block compares the value set in each PWM data register (PWMR0-PWMR8) with the count value of each 8-bit counter with the timing of each basic clock (f_{PWM0}, f_{PWM1}, f_{PWM2}). The block outputs the low level when a match with the value of a PWM data register is found. When the counter value reaches 0, the block outputs the high level with the timing of a basic clock.

Let x be the value set in a PWM data register. Then, the duty cycle is as follows:

$$\text{Duty cycle:}\quad D = \frac{x}{256} \times 100\%$$

A basic clock of 500 kHz is used, so that the frequency and period of an output signal are as follows:

$$\text{Frequency:}\quad f = \frac{500\ \text{kHz}}{256} \fallingdotseq 1.953\ \text{kHz}$$

Period: $T = \dfrac{256}{500\ \text{kHz}} = 512\ \mu s$

For each pin, a different value can be set in each PWM data register through the data buffer (DBF). This means that a signal with an independent duty cycle can be output on each pin.
Fig. 14-6 shows the format of the PWM data registers.

Fig. 14-6 Format of PWM Data Registers

Data buffer			
DBF3	DBF2	DBF1	DBF0
Don't care	Don't care	← Transfer data →	

GET
8
PUT

Peripheral register										
Register	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Symbol	Peripheral address
PWM0 data register	← Valid data →								PWMR0	0CH
PWM1 data register	← →								PWMR1	0DH
PWM2 data register	← →								PWMR2	0EH
PWM3 data register	← →								PWMR3	0FH
PWM4 data register	← →								PWMR4	10H
PWM5 data register	← →								PWMR5	11H
PWM6 data register	← →								PWMR6	12H
PWM7 data register	← →								PWMR7	13H
PWM8 data register	← →								PWMR8	14H

Value	Sets the PWM output duty cycle for each pin.
0	Duty ratio : $D = \dfrac{x}{256} \times 100\%$
|	
x	Frequency : $f = \dfrac{500}{256}$ kHz
|	$= 1953$ Hz
255	

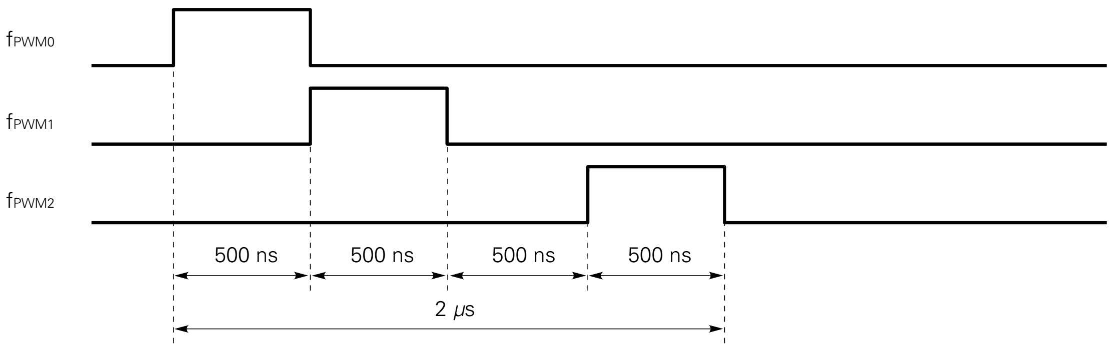
14.4 CLOCK GENERATION BLOCK

The clock generation block outputs the basic clocks (f_{PWM0}, f_{PWM1}, f_{PWM2}) used to set the duty cycle of each output signal.

The output frequency is 500 kHz for all of f_{PWM0}, f_{PWM1}, f_{PWM2}.

Fig. 14-7 shows the phase differences among the clocks.

Fig. 14-7 Phase Differences among Basic Clocks

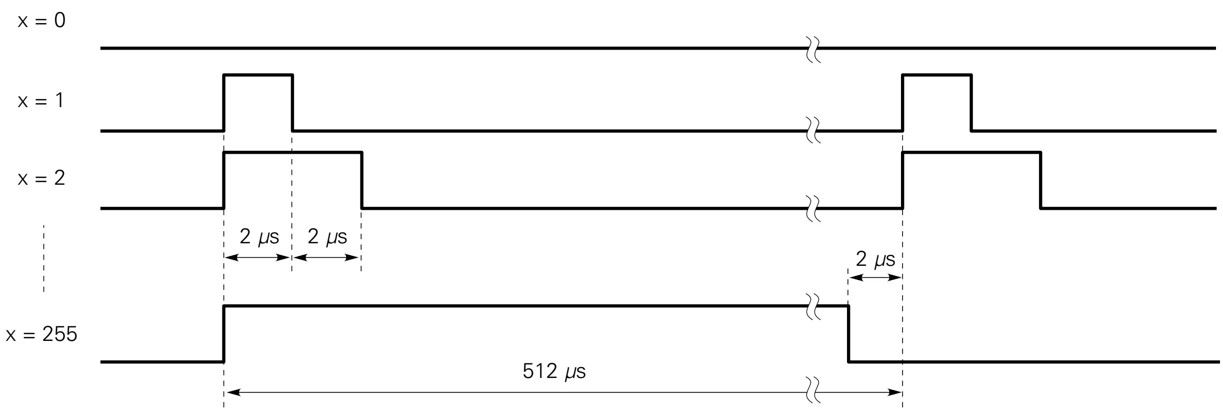


14.5 OUTPUT WAVEFORMS OF D/A CONVERTER

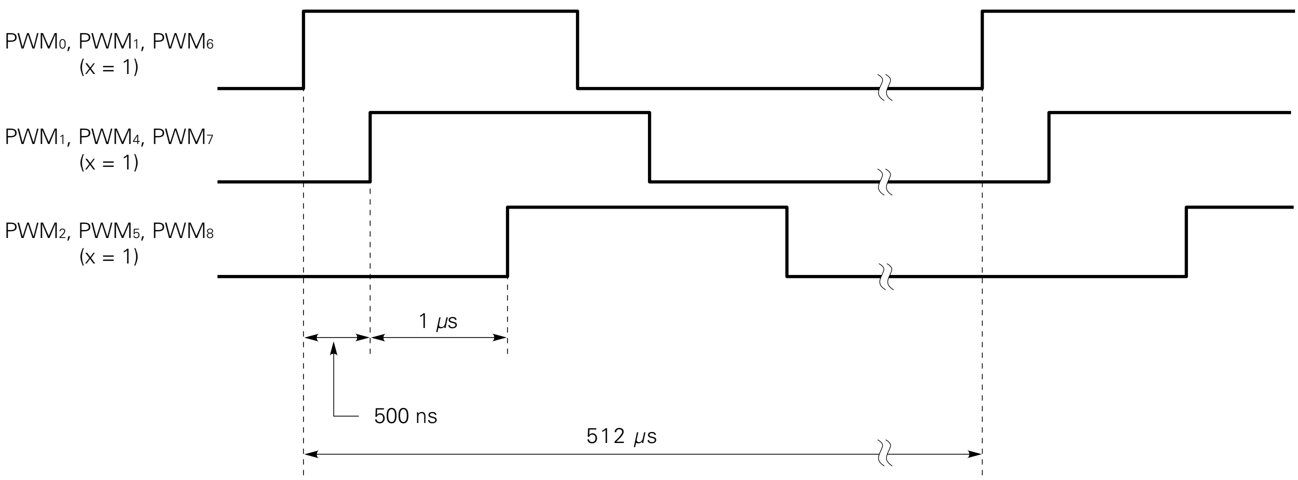
The relationships between duty cycles and output waveforms are shown in (1).

The output waveform on each pin is shown in (2) below. A phase difference results because of a different basic clock supplied.

(1) Duty cycles and output waveforms



(2) Output waveform on each pin



14.6 NOTES ON USING D/A CONVERTER

(1) After turning on the power, perform initial PWM output setting according to the procedure below. The PWM data registers are undefined when the power is turned on, so that this procedure is required to set desired data beforehand.

① Set desired values in the PWM data registers.

② Set the PWMnSEL flags.

(2) Never rewrite the data set in a PWM data register during PWM operation. This is because an output signal with a correct duty cycle cannot be obtained during one period (512 μs).

14.7 STATES UPON RESET

14.7.1 Power-On Reset

The $P2C_0/PWM_0$ to $P2A_0/PWM_8$ pins are set as general-purpose output ports.
All values output at this time are undefined.
The value of each PWM data register is undefined.

14.7.2 Clock Stop

The $P2C_0/PWM_0$ to $P2A_0/PWM_8$ pins are set as general-purpose output ports.
All values output at this time are the previous latch values.
The previous value of each PWM data register is preserved.

14.7.3 CE Reset

The $P2C_0/PWM_0$ to $P2A_0/PWM_8$ pins preserve the previous output states.
This means that those pins that are used for the D/A converter preserve the PWM output states.

14.7.4 Halt State

The $P2C_0/PWM_0$ to $P2A_0/PWM_8$ pins preserve the previous output states.
This means that those pins that are used for the D/A converter preserve the PWM output states.

15. SERIAL INTERFACE

15.1 GENERAL

Fig. 15-1 outlines the serial interface.

Table 15-1 shows the serial interface classes and communication modes.

As shown in Fig. 15-1, the serial interface consists of two systems: serial interface 0 (SIO0) and serial interface 1 (SIO1).

Serial interface 0 and serial interface 1 can be used simultaneously.

Serial interface 0 can use a 2-wire system or a 3-wire system. The 2-wire system uses the SDA and SCL pins and the 3-wire system uses the $\overline{SCK_0}$, SO_0, and SI_0 pins.

With the 2-wire system, I^2C bus or serial I/O can be selected as the communication mode.

Serial interface 1 can only use a 3-wire system. The pins used are $\overline{SCK_1}$, SO_1, and SI_1. The communication mode is serial I/O.

Fig. 15-1 Serial Interface

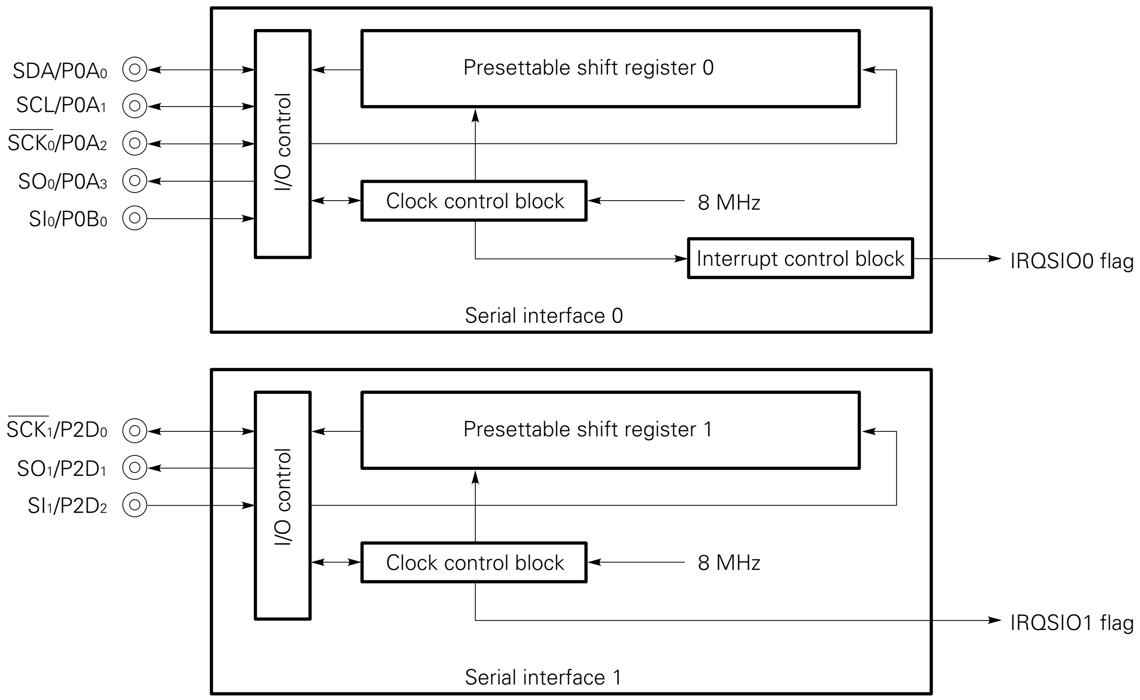


Table 15-1 Serial Interface Classes and Communication Modes

Channel	Number of wires	Communication mode	Pins used
Serial interface 0	2-wire system	I^2C bus	$P0A_0$/SDA
		Serial I/O	$P0A_1$/SCL
	3-wire system	Serial I/O	$P0A_2/\overline{SCK_0}$ $P0A_3/SO_0$ $P0B_0/SI_0$
Serial interface 1	3-wire system	Serial I/O	$P2D_0/\overline{SCK_1}$ $P2D_1/SO_1$ $P2D_2/SI_1$

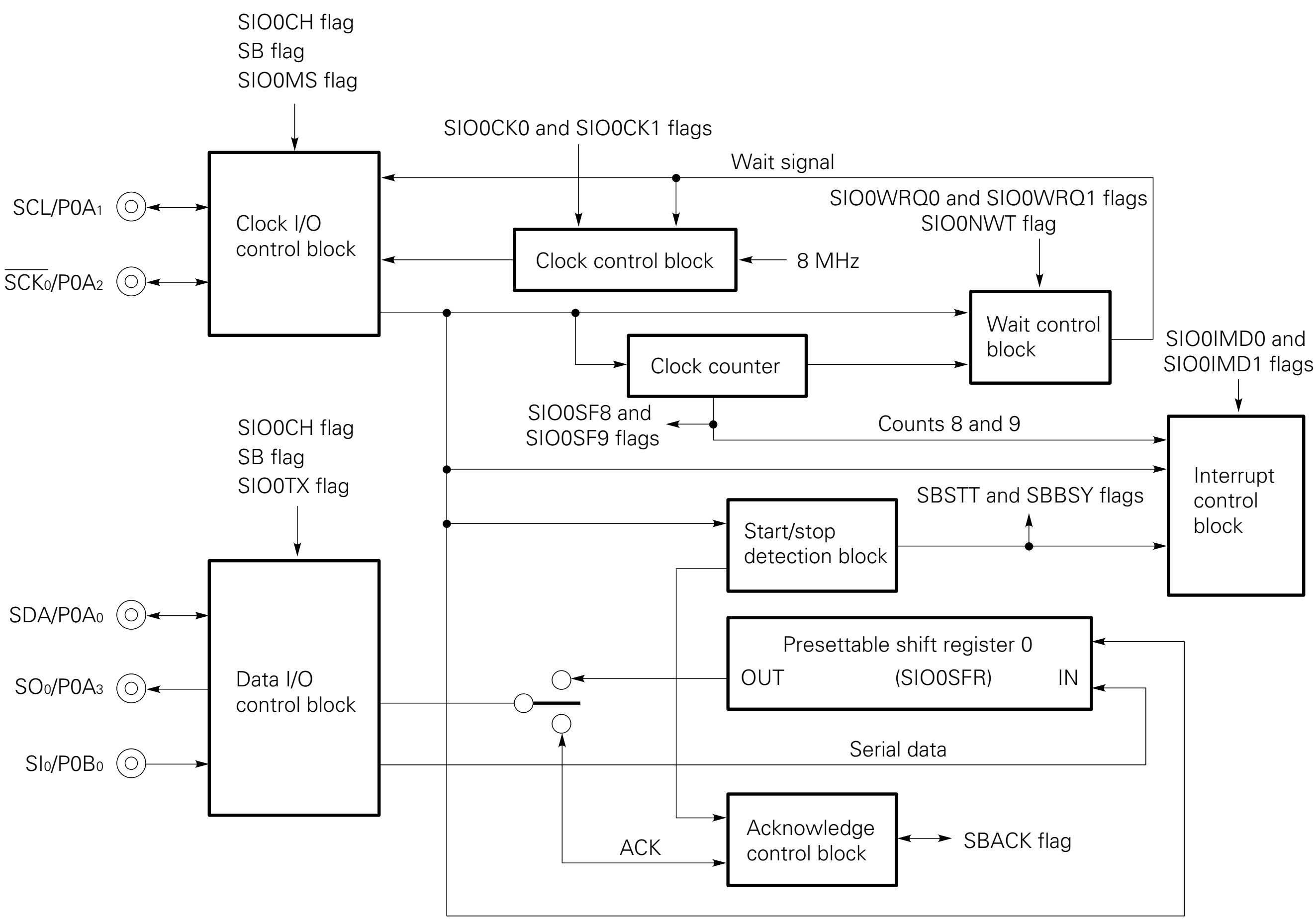
15.2 SERIAL INTERFACE 0

15.2.1 General

Fig. 15-2 outlines serial interface 0.

Serial interface 0 can use a 2-wire I^2C bus or 2-wire/3-wire serial I/O mode.

Fig. 15-2 Serial Interface 0



Remarks

1. SIO0CH (bit 3 of serial I/O-0 mode selection register: see **Fig. 15-3**): 2-wire/3-wire system selection
2. SB (bit 2 of serial I/O-0 mode selection register: see **Fig. 15-3**): I^2C bus/serial I/O selection
3. SIO0MS (bit 1 of serial I/O-0 mode selection register: see **Fig. 15-3**): Master/slave selection
4. SIO0TX (bit 0 of serial I/O-0 mode selection register: see **Fig. 15-3**): Receive/transmit selection
5. SIO0CK0 and SIO0CK1 (bits 0 and 1 of serial I/O-0 clock selection register: see **Fig. 15-4**): Set the internal shift clock frequency.
6. SIO0WRQ0 and SIO0WRQ1 (bits 0 and 1 of serial I/O-0 wait control register: see **Fig. 15-7**): Set the communication wait condition.
7. SIO0NWT (bit 2 of serial I/O-0 wait control register: see **Fig. 15-7**): Sets the start of communication.
8. SIO0SF8 and SIO0SF9 (bits 3 and 2 of serial I/O-0 status judge register: see **Fig. 15-5**): Clock counter detection
9. SBSTT and SBBSY (bits 1 and 0 of serial I/O-0 status judge register: see **Fig. 15-5**): I^2C bus start condition, stop condition, and clock counter detection
10. SIO0IMD0 and SIO0IMD1 (bits 0 and 1 of serial I/O-0 interrupt mode register: see **Fig. 15-8**): Set the interrupt timing.
11. SBACK (bit 3 of serial I/O-0 wait control register: see **Fig. 15-7**): Acknowledge data read/write

15.2.2 Clock I/O Control Block and Data I/O Control Block

The clock I/O control block and data I/O control block control the serial interface 0 communication mode (I^2C bus or serial I/O), the pins used (2-wire system or 3-wire system), and the transmit and receive operations.

The SIO0CH and SB flags select the 2-wire/3-wire system and I^2C bus/serial I/O, respectively.

The SIO0MS flag selects internal clock (master)/external clock (slave) and the SIO0TX flag selects the receive (RX)/transmit (TX) operation.

These flags are located in the serial I/O-0 mode selection register.

Fig. 15-3 shows the organization and functions of the serial I/O-0 mode selection register.

Table 15-2 shows the pin settings.

As shown in Table 15-2, to set the pins, the serial interface control flag and the I/O setting flag of each pin, must be manipulated.

Fig. 15-3 Organization of Serial I/O-0 Mode Selection Register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Serial I/O0 mode selection register	SIO0CH	SB	SIO0MS	SIO0TX	08H	R/W

SIO0TX (b0): Sets the SDA/P0A0 pin (2-wire system) and SO0/P0A3 pin (3-wire system) serial I/O mode. (Receive "RX" and transmit "TX" operation selection)

	2-wire system (SDA/P0A0 pin)	3-wire system (SO0/P0A3 pin)
0	Serial input (Hi-Z) : RX operation	General-purpose port
1	Serial output : TX operation	Serial output : TX operation

SIO0MS (b1): Sets the shift clock to be used.

	I²C 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 (b2): I²C bus or serial I/O mode selection

0	Serial I/O mode
1	I²C bus mode (At this time, do not set the SIO0CH flag to 1.)

SIO0CH (b3): 2-wire system or 3-wire system selection

0	2-wire system
1	3-wire system (At this time, do not set the SB flag to 1.)

Upon reset		b3	b2	b1	b0
	Power-on	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

Table 15-2 Pin Settings by Control Flag

Flag							Pin						
SIO0CH	SB	Communication mode	SIO0MS	Clock to be used	SIO0TX	Serial I/O	Pin name	P0ABIO3	P0ABIO2	P0ABIO1	P0ABIO0	P0BBIO0	Pin setting
0	0	2-wire serial I/O			0	Input (receive)	P0A$_0$/SDA				0		Serial input
											1		General-purpose output port
					1	Output (transmit)					0		Serial output
											1		
			0	External			P0A$_1$/SCL			0			External clock
										1			General-purpose output port
			1	Internal						0			Internal clock
										1			
							P0A$_2$/$\overline{SCK}_0$						General-purpose I/O port
							P0A$_3$/SO$_0$						General-purpose I/O port
							P0B$_0$/SI$_0$						General-purpose I/O port
0	1	2-wire I^{2}C bus			0	Input (receive)	P0A$_0$/SDA				0		Serial input
											1		General-purpose output port
					1	Output (transmit)					0		Serial output
											1		
			0	External (slave)			P0A$_1$/SCL			0			External clock (slave)
										1			General-purpose output port
			1	Internal (master)						0			Internal clock (master)
										1			
							P0A$_2$/$\overline{SCK}_0$						General-purpose I/O port
							P0A$_3$/SO$_0$						General-purpose I/O port
							P0B$_0$/SI$_0$						General-purpose I/O port
1	0	3-wire serial I/O					P0A$_0$/SDA						General-purpose I/O port
							P0A$_1$/SCL						General-purpose I/O port
			0	External			P0A$_2$/$\overline{SCK}_0$		0				External clock
									1				General-purpose output port
			1	Internal					0				Internal clock
									1				
					0	General-purpose port	P0A$_3$/SO$_0$	0					General-purpose input port
								1					General-purpose output port
					1	Output (transmit)		0					Serial output
								1					
							P0B$_0$/SI$_0$					0	Serial input
												1	General-purpose output port
1	1						Not to be set.						

15.2.3 Clock Control Block

The clock control block controls the clock generation and clock output timings when the internal clock is used (master operation).

The SIO0CK0 and SIO0CK1 flags of the serial I/O-0 clock selection register set the internal clock frequency f_{SC}.

Fig. 15-4 shows the organization and functions of the serial I/O-0 clock selection register.

The shift clock output from the clock control block is valid only during master operation (SIO0MS flag = 1).

For the clock generation timing, see the item for each communication mode.

Fig. 15-4 Organization of Serial I/O-0 Clock Selection Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Serial I/O-0 clock selection register	0	0	SIO0CK1	SIO0CK0	39H	R/W

Sets the serial interface 0 internal shift clock frequency f_{SC}.

SIO0CK1	SIO0CK0	f_{SC}
0	0	100 kHz
0	1	50 kHz
1	0	500 kHz
1	1	1 MHz

b_3, b_2: Fixed to 0.

Upon reset		b_3	b_2	b_1, b_0
	Power-on	0	0	Undefined
	Clock stop	↓	↓	Hold
	CE	↓	↓	Hold

15.2.4 Clock Counter and Start/Stop Detection Block

The clock counter is a wrap around counter that counts the rising edge of the clock pulses.

Whether the internal clock or the external clock is used cannot be judged because the clock counter reads the state of the clock pin directly.

The contents of the clock counter can be detected through the SIO0SF8 and SIO0SF9 flags of the serial I/O-0 status judge register, but cannot be read directly by program.

The start/stop detection block detects the start and stop conditions when the I^2C bus mode is used.

The start and stop conditions are detected with the SBSTT and SBBSY flags of the I/O-0 status judge register.

Fig. 15-5 shows the organization and functions of the serial I/O-0 status judge register.

For clock counter operation and timing chart, see the item for each communication mode.

Fig. 15-5 Organization of Serial I/O-0 Status Change Register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Serial I/O0 status judge register	SIO0SF8	SIO0SF9	SBSTT	SBBSY	28H	R

SBBSY (b0): I^2C bus mode start/stop condition detection

	I^2C bus mode	Serial I/O mode
0	Detect stop condition	Hold at "0".
1	Detect start condition	

SBSTT (b1): I^2C bus mode start condition and clock counter detection

	I^2C bus mode	Serial I/O mode
0	Detect falling edge of clock when clock counter reaches "9".	Hold previous value.
1	Detect start condition	

SIO0SF9 (b2): Clock counter detection

	I^2C bus mode	Serial I/O mode
0	Clock counter "0" or "1"	
1	Detect rising edge of clock when clock counter reaches "9".	Clock counter "9".

SIO0SF8 (b3): Clock counter detection

	I^2C bus mode	Serial I/O mode
0	Clock counter "0" or "1"	
1	Detect rising edge of clock when clock counter reaches "8".	Clock counter "8"

Upon reset		b3	b2	b1	b0
	Power-on	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

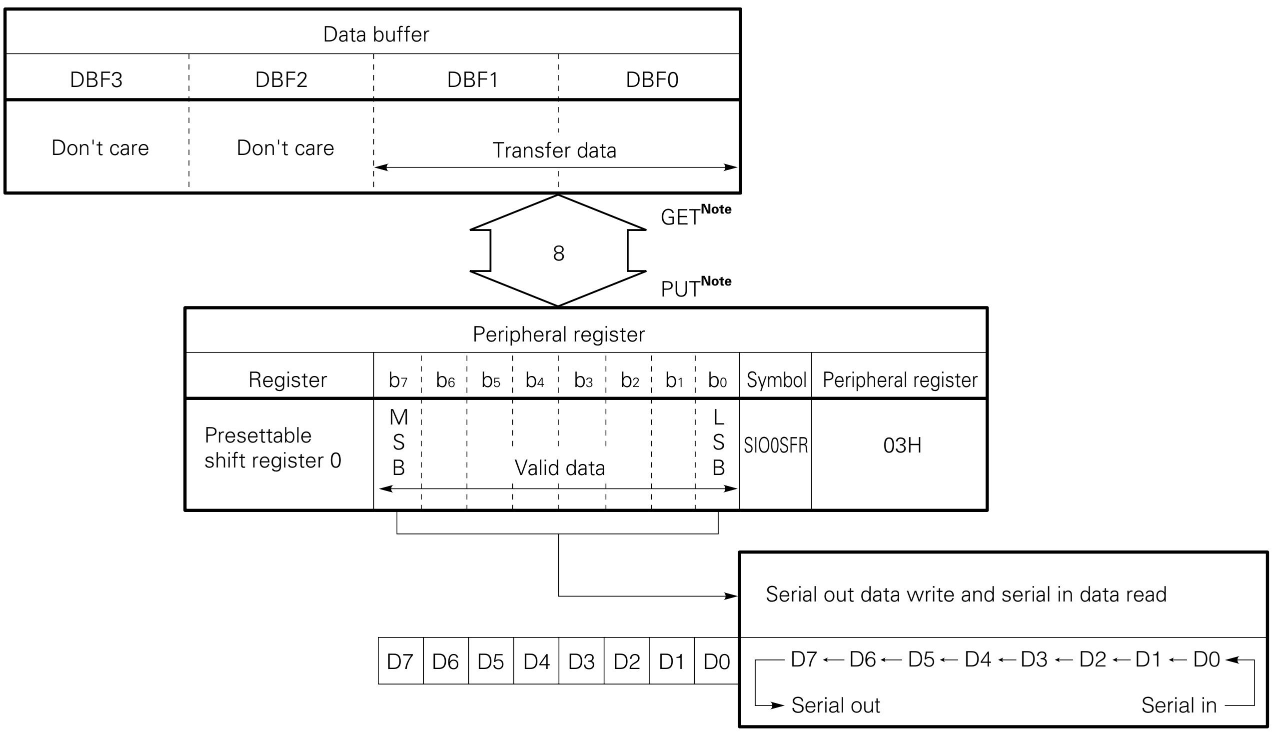
15.2.5 Presettable Shift Register 0

Presettable shift register 0 is an 8-bit shift register for writing serial out data and reading serial in data. It is written and read through a data buffer.

Presettable shift register 0 outputs the contents of the most significant bit (MSB) to the serial data I/O pin in synchronization with the falling edge of the shift clock and reads data at the least significant bit (LSB) in synchronization with the rising edge of the shift clock.

Fig. 15-6 shows the organization and functions of presettable shift register 0.

Fig. 15-6 Organization of Presettable Shift Register 0



Note If a PUT or GET instruction is executed during serial communication, the data may be destroyed. For details, see **Section 15.2.10**.

15.2.6 Wait Control Block and Acknowledge Control Block

The wait control block controls communication wait and release.

The SIO0WRQ0 and SIO0WRQ1 flags (bits 0 and 1 of the serial I/O-0 wait control register) set the wait condition.

Serial communication is started by setting (wait release) the SIO0NWT flag (bit 2 of the serial I/O-0 wait control register).

The communication state can be detected with the SIO0NWT flag.

When "0" is written in the SIO0NWT flag in the wait released state, the wait state is set. This is called "forced wait".

The acknowledge control block outputs and detects the acknowledge signal when the I^2C bus mode is used.

Acknowledge is written and read by the SBACK flag (bit 3 of the serial I/O-0 wait control register).

Fig. 15-7 shows the organization and functions of the serial I/O-0 wait control register.

Fig. 15-7 Organizations of Serial I/O-0 Wait Control Register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
Serial I/O-0 wait control register	SBACK	SIO0NWT	SIO0WRQ1	SIO0WRQ0	18H	R/W

Wait condition setting (SIO0WRQ1, SIO0WRQ0)

b1	b0	Name	I^2C bus mode	Serial I/O mode
0	0	No wait	Does not wait.	
0	1	Data wait	The wait state is set by the falling edge of the shift clock when the clock counter reaches "8".	The wait state is set by the rising edge of the shift clock when the clock counter reaches "8".
1	0	Acknowledge wait	The wait state is set by the falling edge of the shift clock when the clock counter reaches "9".	The wait state is set by the rising edge of the shift clock when the clock counter reaches "9".
1	1	Address wait	Falling edge of clock when clock counter reaches "8" after start condition detected.	Not to be set.

Wait setting and wait state detection (SIO0NWT)

b2	When flag written	When flag read
0	Forced wait	Wait according to the condition specified with the SIO0WRQ0 and SIO0WRQ1 flags
1	Release wait (serial communication start)	In serial communication

I^2C bus mode acknowledge signal setting and detection (SBACK)

b3	I^2C bus mode: At reception (SIO0TX = 0)	I^2C bus mode: At transmission (SIO0TX = 1)	Serial I/O mode
0	Output "0" as acknowledge	Detect slave acknowledge (acknowledge = "0")	No operation. Therefore, can be used for other purposes.
1	Output "1" as acknowledge	Detect slave acknowledge (acknowledge = "1")	

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	0	0	0	0
CE	0	0	0	0

15.2.7 Interrupt Control Block

In the interrupt control block, the serial I/O-0 interrupt mode register specifies the condition at which an interrupt request is issued.

When the interrupt request condition is established, the IRQSIO0 flag is set.

Change the interrupt condition in the wait state. If the interrupt condition is changed in the wait released state, an interrupt request may be issued at the time the condition is changed.

Fig. 15-8 shows the organization and functions of the serial I/O-0 interrupt mode register.

Fig. 15-8 Organization of Serial I/O-0 Interrupt Mode Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Serial I/O-0 interrupt mode register	0	0	SIOIMD1	SIOIMD0	38H	R/W

b_1, b_0: Sets the interrupt request condition.

SIOIMD1	SIOIMD0	I^2C bus mode	Serial I/O mode
0	0	Rising edge of shift clock when clock counter reaches "7".	Rising edge of shift clock when clock counter reaches"7". Note 1
0	1	Rising edge of shift clock when clock counter reaches"8".	Rising edge of shift clock when clock counter reaches "8". Note 2
1	0	Rising edge of shift clock when clock counter reaches"7" after start condition detected. Note 3	Interrupt request not issued.
1	1	When stop condition detected. Note 4	Interrupt request not issued.

b_3, b_2: Fixed to 0.

Upon reset	b_3	b_2	b_1, b_0
Power-on	0	0	Undefined
Clock stop	↓	↓	Hold
CE	↓	↓	Hold

Notes 1. If this mode is set when the clock counter count is "7", an interrupt request is issued.

2. If this mode is set when the clock counter count is "8", an interrupt request is issued.

3. If this mode is set when the SBSTT flag is "1" and the clock counter count is "7", an interrupt request is issued.

4. If this mode is set after the stop condition has been specified, an interrupt request is issued.

15.2.8 I^{2}C Bus Mode

(1) General

The I^{2}C bus mode communicates with 2-wires: SCL pin and SDA pin.

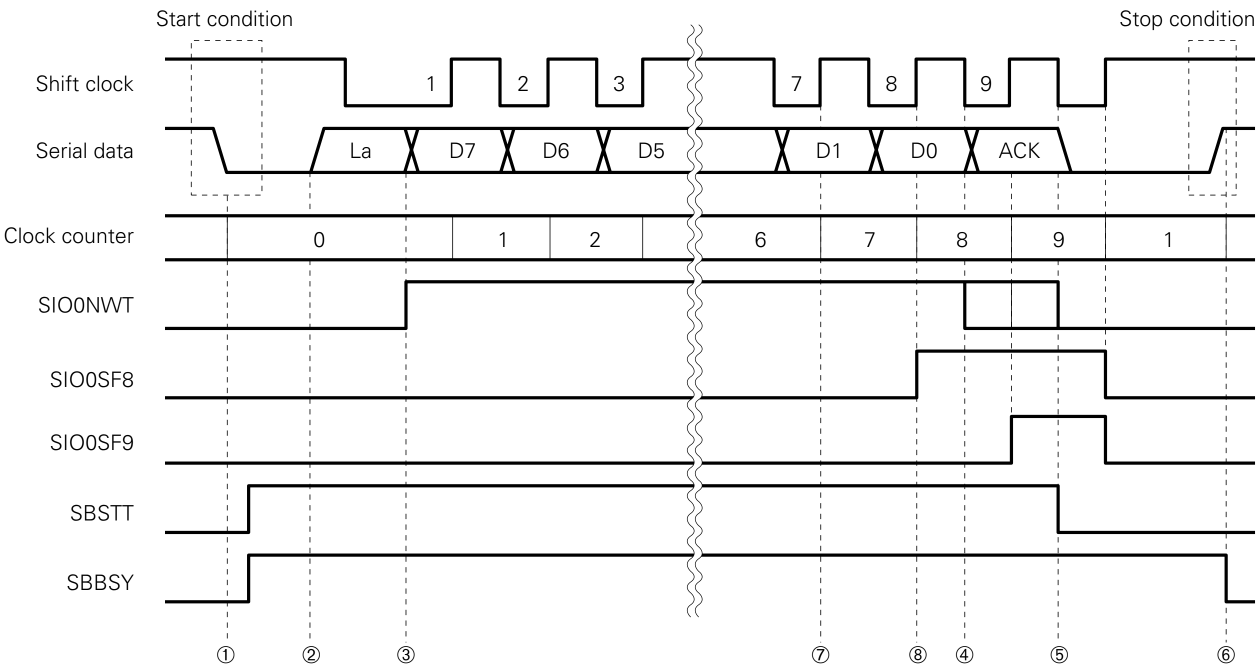
It has the following features:

- Communication can be controlled by start/stop conditions and 9th clock acknowledge.
- The communication wait state can be set by externally fixing the clock at a low level by using the N-ch open drain pin.

(2) Timing chart

Fig. 15-9 is the timing chart.

Fig. 15-9 I^{2}C Bus Mode Timing Chart



Remarks ① Start condition generation by general-purpose I/O port
② Master transmit state setting
③ Wait release
④ Wait timing at address wait and data wait setting
⑤ Wait timing at acknowledge wait setting
⑥ Stop condition generation by general-purpose I/O port
⑥, ⑦, ⑧ Interrupt request timing

(3) Clock counter operation

The clock counter initial value is “0”. Thereafter, the clock counter is incremented each time the rising edge of the clock pin signal is detected. When the clock counter reaches “9”, it returns to “1” and continues counting.

The clock counter reset conditions are:

① Power-on reset
② Clock-stop instruction execution
③ Start condition detection
④ Communication mode switched from I^2C bus mode to 2-wire or 3-wire serial I/O mode
⑤ CE reset

(4) Wait operation and cautions

When the wait state is released, serial data is immediately output (at transmit operation) and the serial interface remains in the wait released state until the condition set by the SIO0WRQ0 and SIO0WRQ1 flags is satisfied.

When the wait condition is satisfied, the shift clock pin is made low level and operation of the clock counter and presettable shift register 0 is stopped.

Note that if data is written into presettable shift register 0 while the serial interface is in the released state and the shift clock pin is low level, the data may not be written correctly.

If forced wait is specified in the wait released state, the forced wait state is set at the falling edge of the next clock pulse after “0” is written in the SIO0NWT flag.

The wait released state is not changed even if wait release is specified again in that state.

Note that when forced wait is specified in the wait state, one shift clock is output.

When using the I^2C bus mode, do not set the data wait condition (SIO0WRQ0 = 1, SIO0WRQ1 = 0) consecutively.

This is because when the data wait condition is set twice in succession to release the wait state, the wait state will be immediately set when it is released for the second time.

When the shift clock output pin is externally forced to low level while it is at high level during master operation (this is called wait request from a slave), master operation is set to the wait state.

In this case, operation restarts at the time the slave wait request is cleared.

(5) Interrupt request timing

The interrupt request timing can be selected with the SIO0IMD0 and SIO0IMD1 flags.

(6) Acknowledge block and its operation

The acknowledge block operates only when the I^2C bus mode is used.

It is used in receive operation acknowledge signal output and transmit operation acknowledge signal detection.

During the receive operation, the contents of the SBACK flag are output from the serial data pin at the falling edge of the shift clock when the clock counter reaches "8".

During the receive operation, once data is set in the SBACK flag, that value is held.

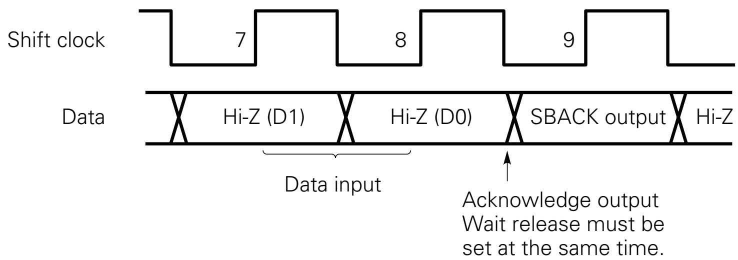
During the transmit operation, the state of the serial data pin is read at the SBACK flag at the rising edge of the shift clock when the clock counter reaches "9".

Fig. 15-10 shows the acknowledge output and input operations.

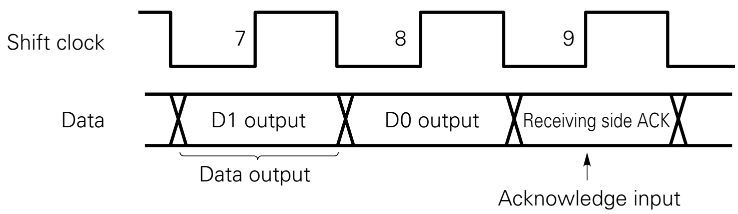
During the receive operation, set acknowledge (SBACK flag setting) simultaneously with release of the wait state (SIO0NWT flag setting). This is because the SBACK and SIO0NWT flags are included in the same register. As a result, if only the SBACK flag is set, the SIO0NWT flag is also set. If the serial interface is in the wait state, the wait state is released in the wait state and one shift clock is output.

Fig. 15-10 Acknowledge Output and Input Operations

● Receive operation



● Transmit operation

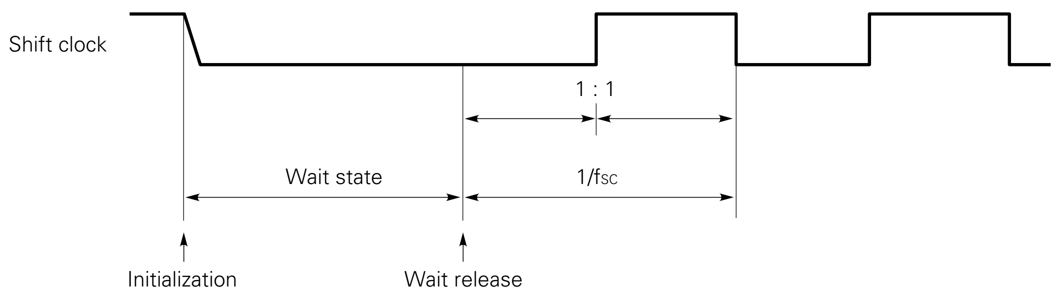


(7) **Shift clock generation timing in I²C bus mode**

(a) **When the initial state is released**

"Initial state" refers to the time that I²C bus method master operation is selected.
While the serial interface is in the wait state, low level is output from the shift clock pin.

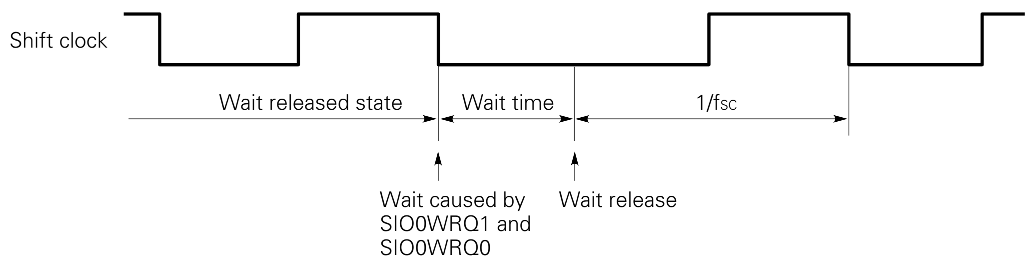
Fig. 15-11 Shift Clock Generation Timing in I²C Bus Mode (1/5)



(b) **When wait operation is performed**

① **When the interface enters the wait state when the condition specified with the SIO0WRQ0 and SIO0WRQ1 flags is satisfied (normal operation)**

Fig. 15-11 Shift Clock Generation Timing in I²C Bus Mode (2/5)



② **When forced wait is set in the wait state**

Nothing changes.

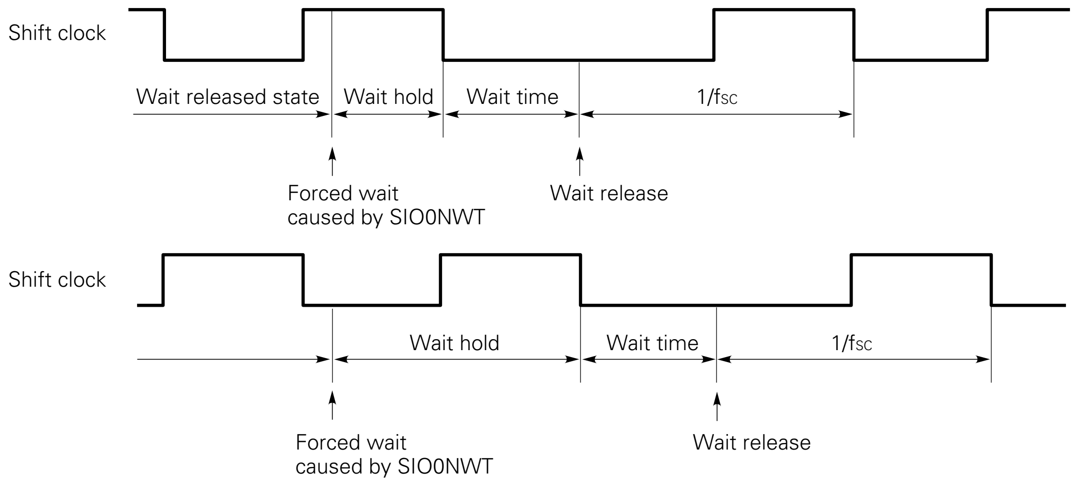
③ **When forced wait is set in the wait released state**

The wait state is set at the falling edge of the next clock after forced wait is set.

However, operation of the clock counter and presettable shift register 0 is stopped at the time force wait is set.

When forced wait is set when the clock pin is low level, the clock counter and presettable shift register 0 operate for one clock.

Fig. 15-11 Shift Clock Generation Timing in I²C Bus Mode (3/5)



④ **When wait release is specified in wait released state**

Nothing changes.

⑤ **When a slave issued a wait request in wait released state**

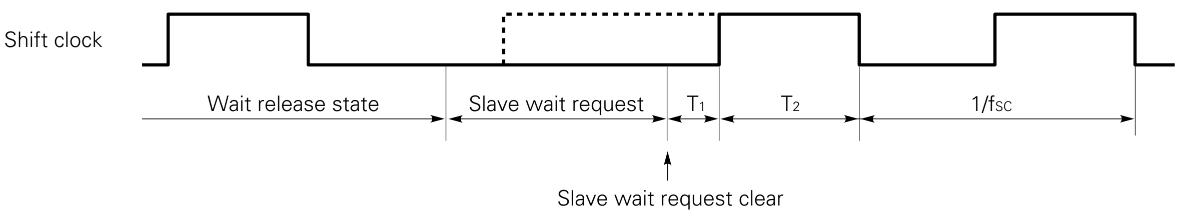
The clock is output at the timing shown in Fig. 15-11 (4/5) after the slave wait request is cleared.

The values of T_1 and T_2 in the table are:

f_{sc}	T_1	T_2
50 kHz	0 to 10 μs	1 to 10 μs
100 kHz	0 to 5 μs	1 to 5 μs
500 kHz	0 to 1 μs	0.5 to 1 μs
1 MHz	0 to 687.5 ns	187.5 to 500 ns

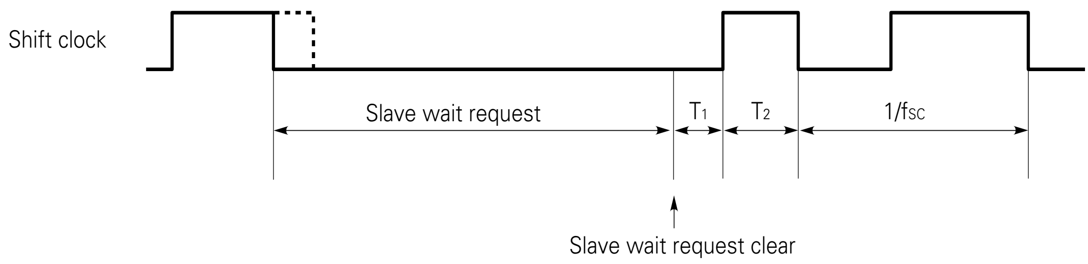
Fig. 15-11 Shift Clock Generation Timing in I²C Bus Mode (4/5)

● If wait request is issued when the SCL pin is at low level



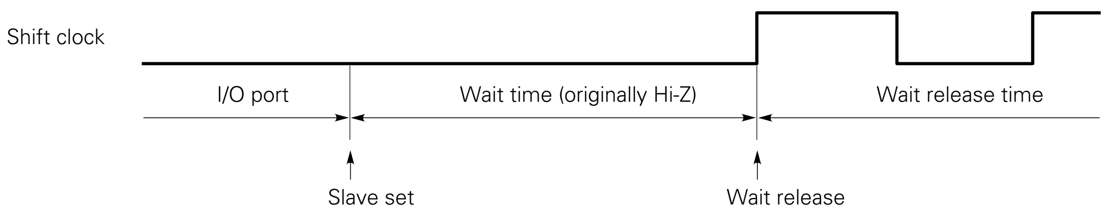
Remark If the slave wait request is cleared before the next rising edge of the SCL pin signal, wait is not recognized and operation continues.

● If wait request is issued when the SCL pin is at high level



(c) Slave (external clock) operation
At the first slave operation setting after the power supply voltage V_{DD} is turned on, the SCL pin output is undefined.
At this time, if the SCL pin is externally set to low level, it outputs low level until the next time the wait state is released.

Fig. 15-11 Shift Clock Generation Timing in I²C Bus Mode (5/5)

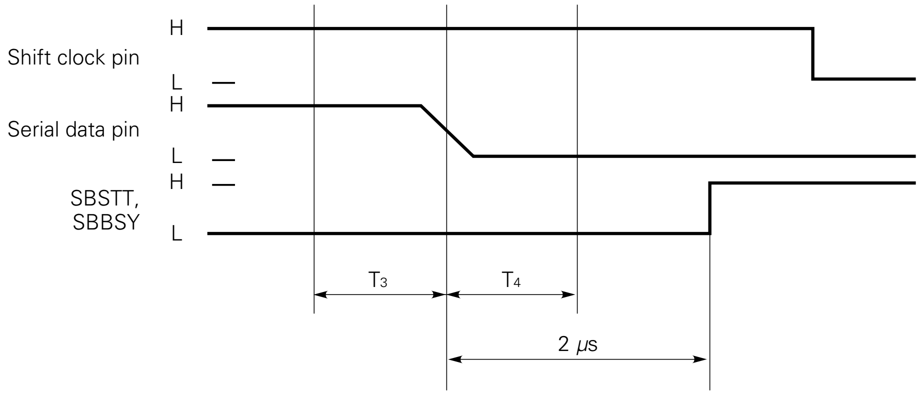


(8) Start/stop conditions and SBSTT and SBBSY flags operation

Fig. 15-12 shows the fetch timing of the start and stop conditions. To fetch the start and stop conditions correctly, the shift clock must satisfy the states indicated in the figure for at least 1 μs (T_3 and T_4) before and after the edges of the serial data. When this condition is satisfied, the SBSTT and SBBSY flags change 2 μs after the edges.
The SBSTT and SBBSY flags operate only when the I²C bus mode is used.
The communication state of the other station can be detected by detecting these flags.
These flags operate without regard to master, slave, receiving, transmitting, waiting, or wait released.
For the serial I/O mode, "0" is held.
For a description of SBSTT flag and SBBSY flag operation, see **Fig. 15-9.**

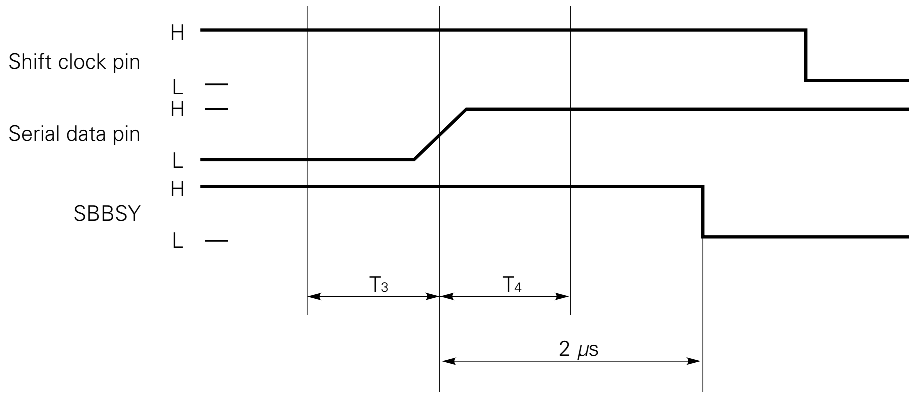
Fig. 15-12 Start/Stop Conditions Fetch Timing

(a) Start condition fetch timing



Note T_3 and T_4 must be at least 1 μs.

(b) Stop condition fetch timing



Note T_3 and T_4 must be at least 1 μs.

Remark Fig. 15-12 (a) and (b) indicate the timings with a clock frequency of 8 MHz.

15.2.9 Serial I/O Mode

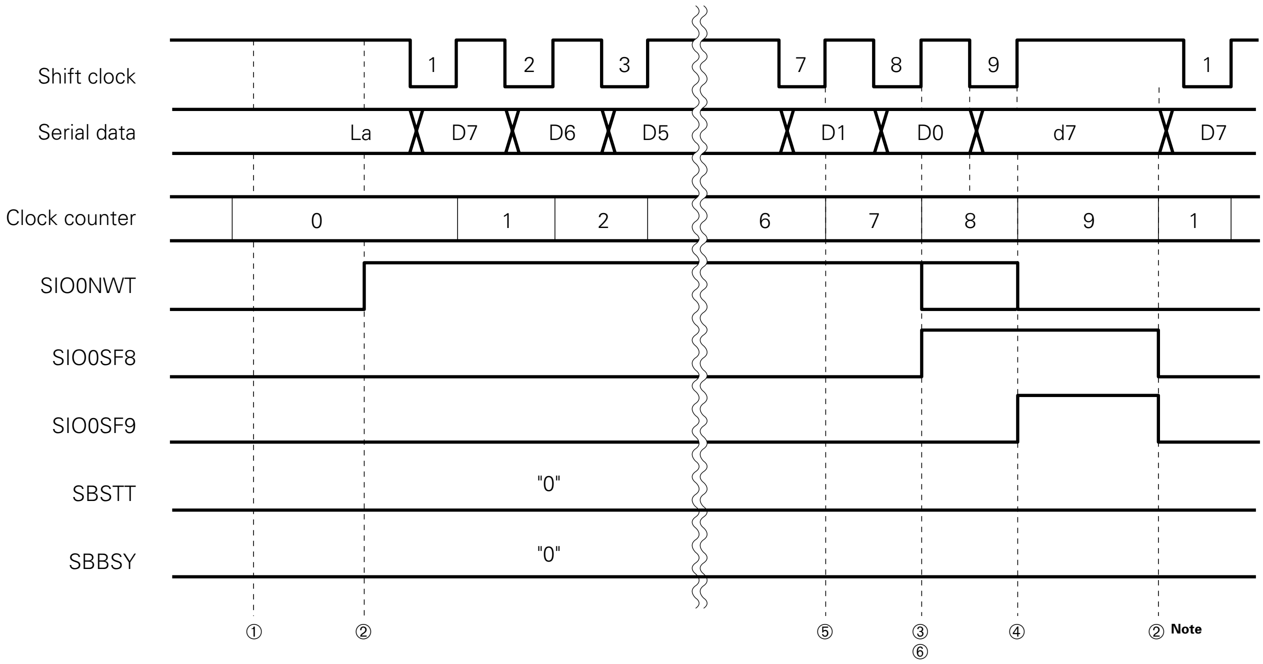
(1) General

In the serial I/O mode, communication is performed with the 2-wire system, which uses the SCL and SDA pins, on with the 3-wire system, which uses the $\overline{SCK_0}$, SO_0, and SI_0 pins.

(2) Timing chart

Fig. 15-13 is the serial I/O mode timing chart.

Fig. 15-13 Serial I/O Mode Timing Chart



Note SIO0SF8 and SIO0SF9 are also reset when the I/O-0 wait control register is written.

Remarks ① Master transmit state setting

② Wait release

③ Wait timing at address wait and data wait setting

④ Wait timing at acknowledge wait setting

⑤, ⑥ Interrupt timing

(3) Clock counter operation

The clock counter initial value is "0". Thereafter, the clock counter is incremented each time the rising edge of the clock pin signal is detected. When the clock counter reaches "9", it returns to "1" and continues counting.

The clock counter reset conditions are:

① Power-on reset
② Clock-stop instruction execution
③ Data written to serial I/O-0 wait control register
④ Communication mode switched from 2-wire or 3-wire serial I/O mode to I^2C bus mode
⑤ CE reset

(4) Wait operation and cautions

If the wait state is released, serial data is output (at transmit operation) at the falling edge of the next clock and the serial interface remains in the wait released state until the condition set with the SIO0WRQ0 and SIO0WRQ1 flag is satisfied.

When the wait condition is satisfied, the shift clock pin is made high level and operation of the clock counter and presettable shift register 0 is stopped.

Note that if data is written and read to and from presettable shift register 0 while the serial interface is in the wait released state and the shift clock pin is high level, the data will not be written correctly.

If data is written into presettable shift register 0 while the serial interface is in the wait released state and the shift clock pin is low level, the contents of MSB is output from the serial data output pin when the PUT instruction is executed.

If forced wait is set in the wait released state, the wait state is set as soon as "0" is written in the SIO0NWT flag.

Note that if wait release is set again in the wait released state, the clock counter will be reset.

(5) Interrupt request timing

The interrupt request timing can be selected with the SIO0IMD0 and SIO0IMD1 flags.

See **Section 15.2.7**.

(6) Acknowledge block and its operation

The acknowledge block operates only when the I^2C bus mode is used.

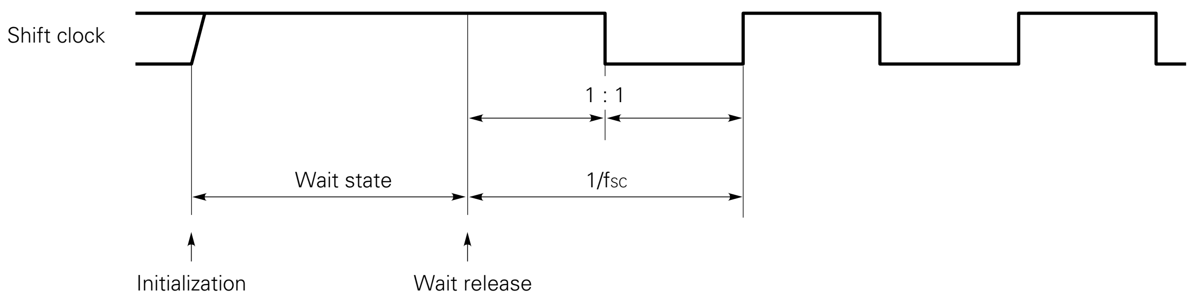
(7) Serial I/O shift clock generation timing

(a) When the initial state is released

"Initial state" refers to the time when serial I/O internal clock operation is selected.

During the wait state, high level is output from the shift clock pin.

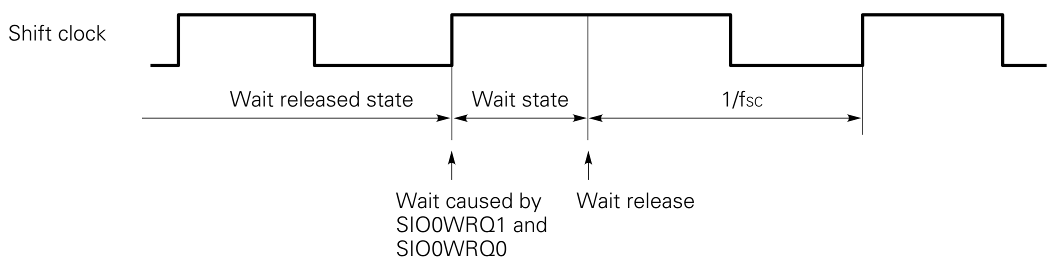
Fig. 15-14 Shift Clock Generation Timing in Serial I/O Mode (1/4)



(b) When wait operation is performed

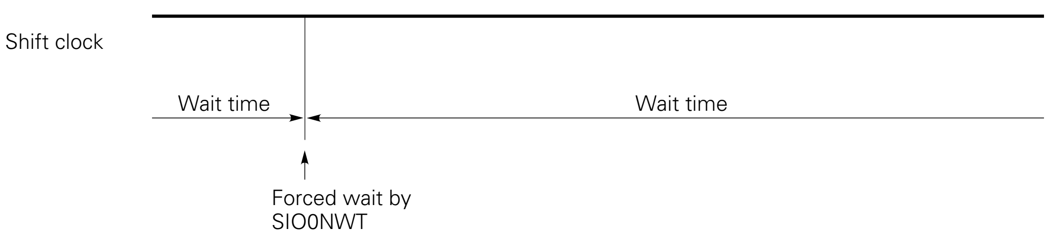
① **When the interface enters the wait state when the condition specified with the SIO0WRQ0 and SIO0WRQ1 flags is satisfied (normal operation)**

Fig. 15-14 Shift Clock Generation Timing in Serial I/O Mode (2/4)



② **When forced wait is set in the wait state**

Fig. 15-14 Shift Clock Generation Timing in Serial I/O Mode (3/4)

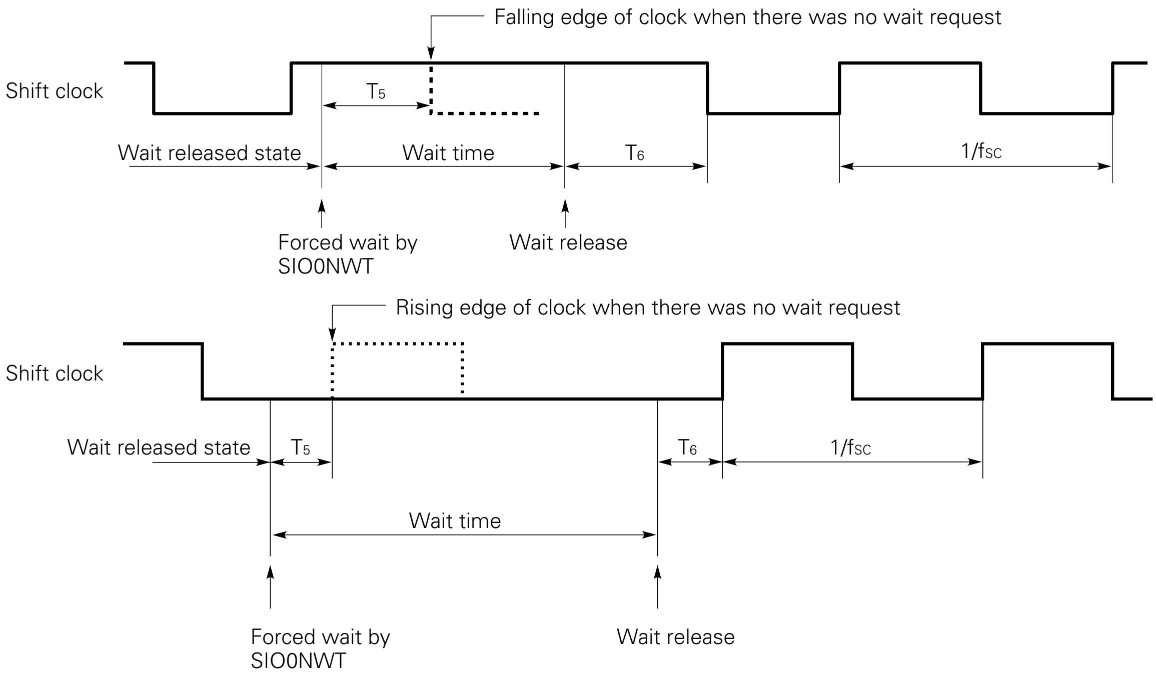


③ **When forced wait is set in the wait released state**

After forced wait release, the clock pulses are output at the specified period after the remaining clock pulses are output. T_5 is equal to T_6.

However, when a clock faster than the shift clock is selected, T_5 does not equal T_6 and becomes $0 \leq T_6 \leq 500$ ns.

Fig. 15-14 Shift Clock Generation Timing in Serial I/O Mode (4/4)



④ **When wait is released in the wait released state**

The clock output waveform does not change. Note that the clock counter is reset.

(8) SBSTT and SBBSY flags operation

The SBSTT and SBBSY flags operate only when the I^2C bus mode is used.

For the serial I/O mode, these flags are held at “0”.

15.2.10 Data Write and Read Cautions

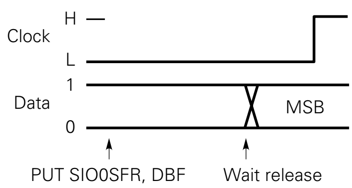
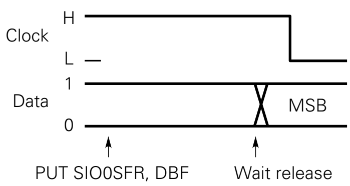
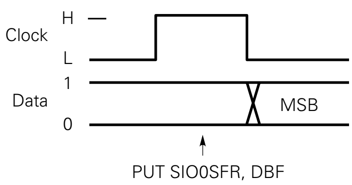
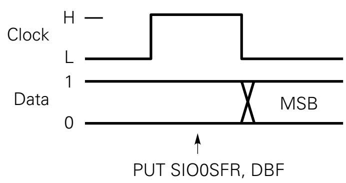
Data is written to presettable shift register 0 with the “PUT SIO0SFR, DBF” instruction.

Data is read with the “GET DBF, SIO0SFR” instruction.

Write and read data in the wait state. If data is written and read in the wait released state, the correct data may not be written and read, depending on the state of the shift clock pin.

The data write and read timing and cautions are given below.

Table 15-3 Presettable Shift Register 0 Data Read and Write Operations and Cautions

<table>
<tr><th colspan="2">State at PUT/
GET execution</th><th>State of shift
clock pin</th><th>I^2C bus mode</th><th>Serial I/O mode</th></tr>
<tr><td rowspan="2">Wait state</td><td>Read (GET)</td><td rowspan="2">● Fixed at low level in the I^2C bus mode
● Fixed at high level in the serial I/O mode</td><td>Normal read</td><td>Normal read</td></tr>
<tr><td>Write (PUT)</td><td>Normal write
The contents of the MSB is output the next time the wait state is released. (At transmit operation)
(At transmit operation)

</td><td>Normal write
The contents of the MSB is output at the falling edge of the shift clock pin signal while the wait state is released next time.

</td></tr>
<tr><td rowspan="4">Wait released state</td><td rowspan="2">Read (GET)</td><td>High level</td><td>Normal read</td><td>Normal read</td></tr>
<tr><td>Low level</td><td>Normal read</td><td>Normal read</td></tr>
<tr><td rowspan="2">Write (PUT)</td><td>High level</td><td>Normal write
The contents of the MSB is output when the clock falls. The clock counter is not reset.

</td><td>Normal write
The contents of the MSB is output when the clock falls. The clock counter is not reset.

</td></tr>
<tr><td>Low level</td><td>Not written normally.
The contents of SIO0SFR are destroyed.</td><td>Not written normally.
The contents of SIO0SFR are destroyed.</td></tr>
</table>

15.2.11 Serial Interface 0 Operation

Tables 15-4 through 15-6 outline operation for each communication mode.

Table 15-4 I²C Bus Mode Operation

Item		Operation mode: I²C bus mode — Slave operation (SIO0MS = 0): Receive (SIO0TX = 0)	Slave operation (SIO0MS = 0): Transit (SIO0TX = 1)	Master operation (SIO0MS = 1): Receive (SIO0TX = 0)	Master operation (SIO0MS = 1): Transit (SIO0TX = 1)
State of each pin	SDA/P0A$_0$	**When POABI00 = 0**, is floating and waiting for external data input **When P0ABIO0 = 1**, works as a general-purpose output port and outputs the contents of the output latch.	Outputs the contents of SIO0SFR at the falling edge of the external clock regardless of P0ABIO0.	**When P0ABIO0 = 0**, is floating and waiting for external data input **When P0ABIO0 = 1**, works as a general-purpose output port and outputs the contents of the output latch.	Outputs the contents of SIO0SFR at the falling edge of the external clock regardless of P0ABIO0.
	SCL/P0A$_1$	**When P0ABIO0 = 0**, is floating and waiting for external data input **When P0ABIO0 = 1**, works as a general-purpose output port and outputs the contents of the output latch.		Outputs the internal clock regardless of P0ABIO1.	
Clock counter		Incremented at the rising edge of the SCL pin signal.			
Presettable shift register 0 operation	Output	Not output	Shifts and outputs the data from the MSB each time the SCL pin signal falls.	Not output	Shifts and outputs the data from the MSB each time the SCL pin signal falls.
	Input	Shifts and inputs the data from the LSB each time the SCL pin signal rises.			
Wait operation	Waiting	Outputs low level from the SCL pin. SDA pin: Floating	Outputs low level from the SCL pin. SDA pin: State held	Outputs low level from the SCL pin. SDA pin: Floating	Outputs low level from the SCL pin. SDA pin: State held
	Wait released	SCL pin: Floating and waiting for external clock input SDA pin: Floating and waiting for external data	SCL pin: Floating and waiting for external clock input SDA pin: Outputs data each time the SCL pin signal falls.	SCL pin: Outputs the internal clock. SDA pin: Floating and waiting for external data	SCL pin: Outputs the internal clock. SDA pin: Outputs data each time the SCL pin signal falls.
Acknowledge		ACK output at the falling edge of the 8th clock.	ACK fetched at the rising edge of the 9th clock	ACK output at the falling edge of the 8th clock.	ACK fetched at the rising edge of the 9th clock.

Table 15-5 2-wire Serial I/O Operation

Item		2-wire serial I/O mode: Slave operation (SIO0MS = 0), Receive (SIO0TX = 0)	Slave operation (SIO0MS = 0), Transit (SIO0TX = 1)	Master operation (SIO0MS = 1), Receive (SIO0TX = 0)	Master operation (SIO0MS = 1), Transit (SIO0TX = 1)
State of each pin	SDA/P0A$_0$	**When P0ABIO0 = 0**, is floating and waiting for external data input **When P0ABIO0 = 1**, works as a general-purpose output port. Outputs the contents of the output latch.	Outputs the contents of SIO0SFR at the falling edge of the external clock regardless of P0ABIO0.	**When P0ABIO0 = 0**, is floating and waiting for external data input **When P0ABIO0 = 1**, works as a general-purpose output port and outputs the contents of the output latch.	Outputs the contents of SIO0SFR at the falling edge of the external clock regardless of P0ABIO0.
	SCL/P0A$_1$	**When P0ABIO0 = 0**, is floating and waiting for external data input **When P0ABIO0 = 1**, works as a general-purpose output port. Outputs the contents of the output latch.		Outputs the internal clock regardless of P0ABIO1.	
Clock counter		Incremented at the rising edge of the SCL pin signal.			
Presettable shift register 0 operation	Output	Not output	Shifts and outputs the data from the MSB each time the SCL pin signal falls.	Not output	Shifts and outputs the data from the MSB each time the SCL pin signal falls.
	Input	Shifts and inputs the data from the LSB each time the SCL pin signal rises.			
Wait operation	Waiting	SCL pin: Floating SDA pin: Floating	SCL pin: Floating SDA pin: State held	SCL pin: Floating SDA pin: Floating	SCL pin: Floating SDA pin: State held
	Wait released	SCL pin: Floating and waiting for external clock input SDA pin: Floating and waiting for external data	SCL pin: Floating and waiting for external clock input SDA pin: Outputs data each time the SCL pin signal falls.	SCL pin: Outputs the internal clock. SDA pin: Floating and waiting for external data	SCL pin: Outputs the internal clock. SDA pin: Outputs data each time the SCL pin signal falls.

Table 15-6 3-wire Serial I/O Operation

Item		3-wire serial I/O mode: Slave operation (SIO0MS = 0), Receive (SIO0TX = 0)	3-wire serial I/O mode: Slave operation (SIO0MS = 0), Transit (SIO0TX = 1)	3-wire serial I/O mode: Master operation (SIO0MS = 1), Receive (SIO0TX = 0)	3-wire serial I/O mode: Master operation (SIO0MS = 1), Transit (SIO0TX = 1)
State of each pin	$\overline{SCK_0}$/$P0A_2$	**When P0ABIO2 = 0**, is floating and waiting for external data input **When P0ABIO2 = 1**, works as a general-purpose output port. Outputs the contents of the output latch.	**When P0ABIO2 = 0**, is floating and waiting for external data input **When P0ABIO2 = 1**, works as a general-purpose output port. Outputs the contents of the output latch.	Outputs the internal clock regardless of P0ABIO2.	Outputs the internal clock regardless of P0ABIO2.
	SO_0/$P0A_3$	**When P0ABIO3 = 0**, works as a general-purpose input port and is floating **When P0ABIO3 = 1**, works as a general-purpose output port and outputs the contents of the output latch.	Outputs the contents of SIO0SFR at the falling edge of the external clock regardless of P0ABIO3.	**When P0ABIO3 = 0**, works as a general-purpose input port and is floating **When P0ABIO3 = 1**, works as a general-purpose output port and outputs the contents of the output latch.	Outputs the contents of SIO0SFR at the falling edge of the external clock regardless of P0ABIO3.
	SI_0/$P0B_0$	**When P0BBIO0 = 0**, is floating and waiting for external data input **When P0BBIO0 = 1**, works as a general-purpose output port. Outputs the contents of the output latch.	**When P0BBIO0 = 0**, is floating and waiting for external data input **When P0BBIO0 = 1**, works as a general-purpose output port. Outputs the contents of the output latch.	**When P0BBIO0 = 0**, is floating and waiting for external data input **When P0BBIO0 = 1**, works as a general-purpose output port. Outputs the contents of the output latch.	**When P0BBIO0 = 0**, is floating and waiting for external data input **When P0BBIO0 = 1**, works as a general-purpose output port. Outputs the contents of the output latch.
Clock counter		Incremented at the rising edge of the $\overline{SCK_0}$ pin signal.	Incremented at the rising edge of the $\overline{SCK_0}$ pin signal.	Incremented at the rising edge of the $\overline{SCK_0}$ pin signal.	Incremented at the rising edge of the $\overline{SCK_0}$ pin signal.
Presettable shift register 0 operation	Output	Not output	Shifts and outputs the data from the MSB each time the $\overline{SCK_0}$ pin signal falls.	Not output	Shifts and outputs the data from the MSB each time the $\overline{SCK_0}$ pin signal falls.
	Input	Shifts and inputs the data from the LSB each time the $\overline{SCK_0}$ pin signal rises.	Shifts and inputs the data from the LSB each time the $\overline{SCK_0}$ pin signal rises.	Shifts and inputs the data from the LSB each time the $\overline{SCK_0}$ pin signal rises.	Shifts and inputs the data from the LSB each time the $\overline{SCK_0}$ pin signal rises.
Wait operation	Waiting	$\overline{SCK_0}$ pin: Floating SO_0 pin: General-purpose port SI_0 pin: Floating	$\overline{SCK_0}$ pin: Floating SO_0 pin: State held SI_0 pin: Floating	$\overline{SCK_0}$ pin: High level output SO_0 pin: General-purpose port SI_0 pin: Floating	$\overline{SCK_0}$ pin: High level output SO_0 pin: State held SI_0 pin: Floating
	Wait released	$\overline{SCK_0}$ pin: Floating and waiting for external clock input SO_0 pin: General-purpose port SI_0 pin: Floating and waiting for external	$\overline{SCK_0}$ pin: Floating and waiting for external clock input SO_0 pin: Data output SI_0 pin: Floating and waiting for external data	$\overline{SCK_0}$ pin: Outputs the internal clock. SO_0 pin: General-purpose port SI_0 pin: Floating and waiting for external data	$\overline{SCK_0}$ pin: Outputs the internal clock. SO_0 pin: Data output SI_0 pin: Floating and waiting for external data

15.2.12 State When Serial Interface 0 Is Reset

(1) Power-on reset

All the pins are set to general-purpose input ports.
The contents of presettable shift register 0 are undefined.

(2) Clock-stop

All the pins are set to general-purpose input ports.
The contents of presettable shift register 0 retain their previous value.

(3) Halt

All the terminals remain in their set states.
The internal clock stops output when a HALT instruction is executed.
The external clock operates.

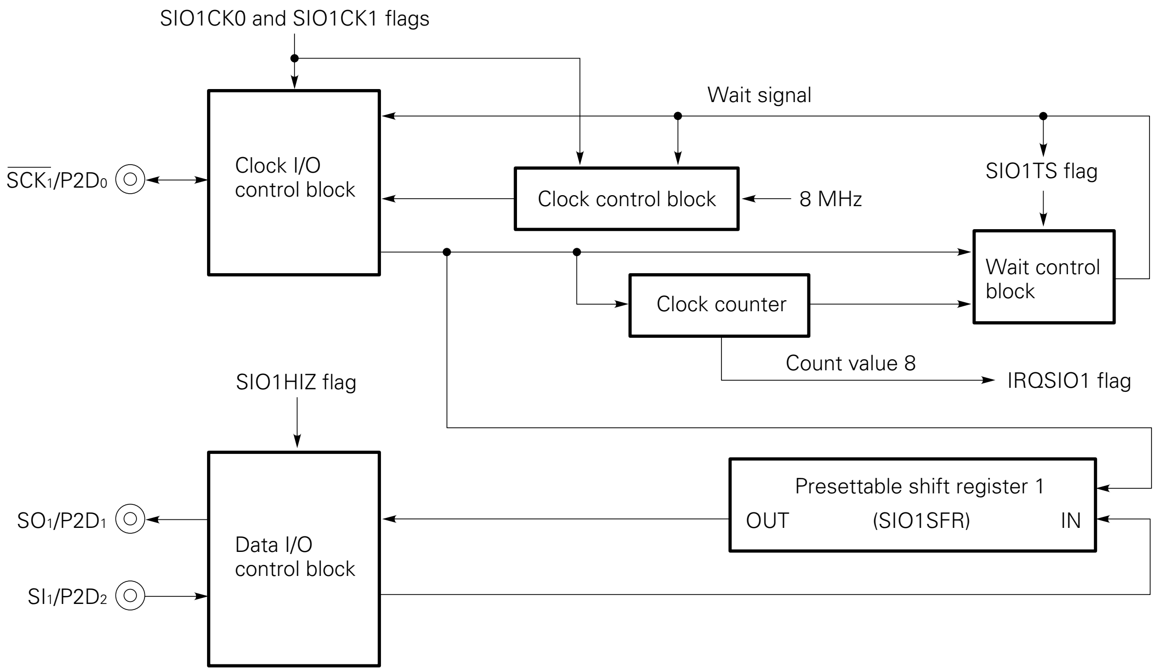
15.3 SERIAL INTERFACE 1

15.3.1 General

Fig. 15-15 outlines serial interface 1.

Serial interface 1 uses the 3-wire serial I/O mode.

Fig. 15-15 Serial Interface 1



Remarks 1. SIO1CK0 and SIO1CK1 (bits 0 and 1 of the serial I/O-1 mode selection register: see **Fig. 15-16**) sets the shift clock.

2. SIO1TS (bit 3 of the serial I/O-1 mode selection register: see **Fig. 15-16**) selects communication operation start/stop.

3. SIO1HIZ (bit 2 of the serial I/O-1 mode selection register: see **Fig. 15-16**) selects the function of the $SO_1/P2D_1$ pins.

15.3.2 Clock I/O Control Block and Data I/O Control Block

The clock I/O control block and data I/O control block control the serial interface 1 transmit and receive operations and select the shift clock.

The SIO1CK0 and SIO1CK1 flags select internal clock (master) or external clock (slave) operation.

The SIO1HIZ flag selects if the SO_1 pin is used as serial data output.

The flags that control the clock I/O control block and data I/O control block are located in the serial I/O-1 mode selection register.

Fig. 15-16 shows the organization and functions of the serial I/O-1 mode selection register. Table 15-7 shows the setting state of each pin.

As shown in Table 15-7, to set each pin, the serial interface control flag and the I/O setting flag of each pin must be manipulated.

Fig. 15-16 Configuration of Serial I/O-1 Mode Selection Register

Register	Flag symbol b3	Flag symbol b2	Flag symbol b1	Flag symbol b0	Address	Read/write
Serial I/O-1 mode selection register	SIO1TS	SIO1HIZ	SIO1CK1	SIO1CK0	1CH	R/W

SIO1CK1, SIO1CK0: Selects the serial interface 1 shift clock.

SIO1CK1	SIO1CK0	Selects the serial interface 1 shift clock.
0	0	External clock input
0	1	100 kHz
1	0	500 kHz
1	1	1 MHz

SIO1HIZ	Selects the function of the P2D1/SO1 pins.
0	General-purpose I/O port
1	Serial data output pin

SIO1TS	Selects serial communication operation start/stop.
0	Operation stop (wait state)
1	Operation start

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	0	0	0	0
CE	0	0	0	0

15.3.3 Clock Counter

The clock counter is a wrap-around counter that counts the rising edge of the clock pulses.

The clock counter reads the state of the clock pin directly. Therefore, whether the clock is internal clock or external clock cannot be judged.

The contents of the clock counter cannot be directly read by program.

Table 15-7 Setting of Each Pin by Control Flag

Flag						Pin				
Communication mode	SIO1HIZ	SIO1 pin setting	SIO1CK1	SIO1CK0	Clock setting	Pin name	P2DBIO2	P2DBIO1	P2DBIO0	Pin setting
3-wire serial I/O			0	0	External clock	$\overline{SCK_1}$/P2D0			0	Waiting : General-purpose input port
										Wait released : External clock input wait
									1	Waiting : General-purpose output port
										Wait released : General-purpose output port
			0	1	Internal clock				0	Waiting : Outputs high level. Wait released : Internal clock output
			1	0						
			1	1					1	
	0	General-purpose port				SO_1/$P2D_1$		0		General-purpose input port
								1		General-purpose output port
	1	Serial output						0		Waiting : Outputs high level. Wait released : Serial data output
								1		
						SI_1/$P2D_2$	0			Serial data input
							1			General-purpose output port

15.3.4 Presettable Shift Register 1

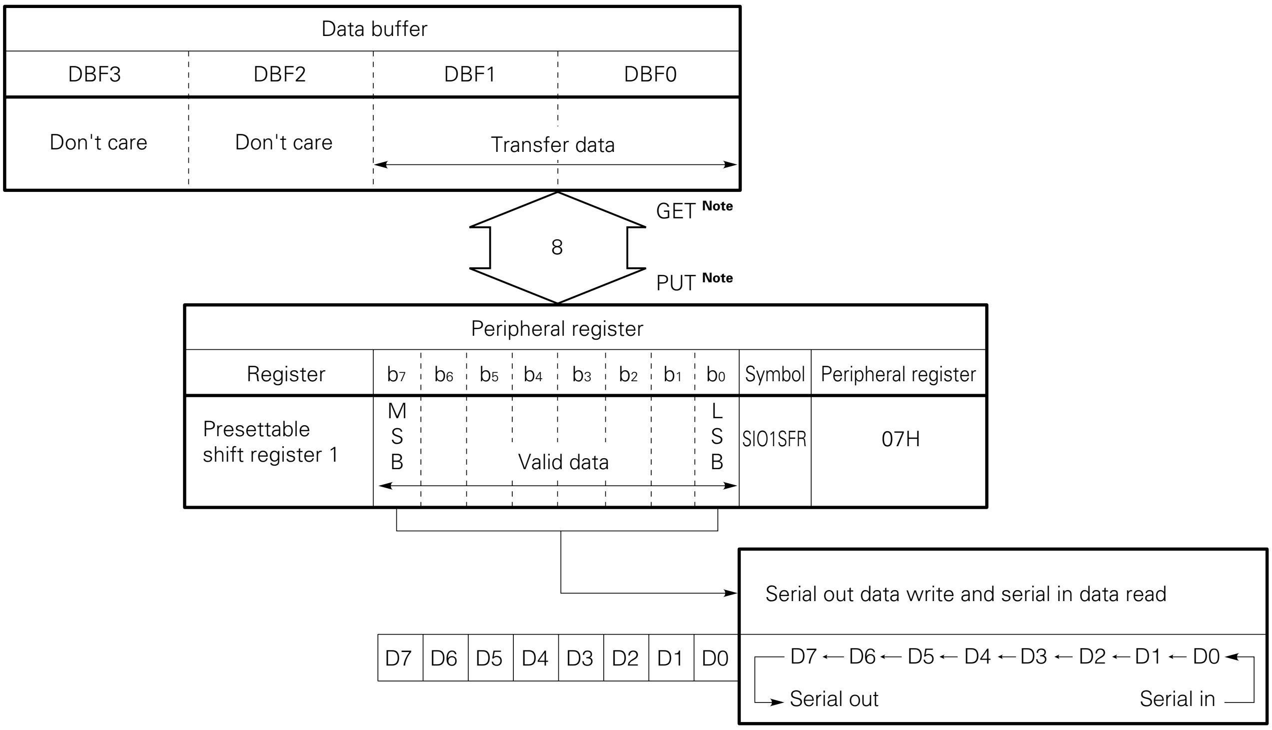
Presettable shift register 1 is an 8-bit shift register for writing serial out data and reading serial in data.

Presettable shift register 1 writes and reads data through a data buffer.

Presettable shift register 1 outputs (at transmit operation) the contents of the most significant bit (MSB) to the serial data I/O pin in synchronization with the falling edge of the shift clock and reads data at the least significant bit (LSB) in synchronization with the rising edge of the shift clock.

Fig. 15-17 shows the organization and functions of presettable shift register 1.

Fig. 15-17 Configuration of Presettable Shift Register 1



Note If a PUT or GET instruction is executed during serial communications, the data may be destroyed. For details, see **Section 15.3.7**.

15.3.5 Wait Control Block

The wait control block controls communication wait and its release.

Serial communication is started by setting wait release at the SIO1TS flag of the serial I/O-1 mode selection register. Then, wait is released. Wait is set again 8 clocks after communication started.

The communication state can be sensed with the SIO1TS flag. In short, the communication state can be sensed by detecting the state of the SIO1TS flag after it is set to "1".

When "0" is written in the SIO1TS flag in the wait released state, the wait state is set. This is called "forced wait".

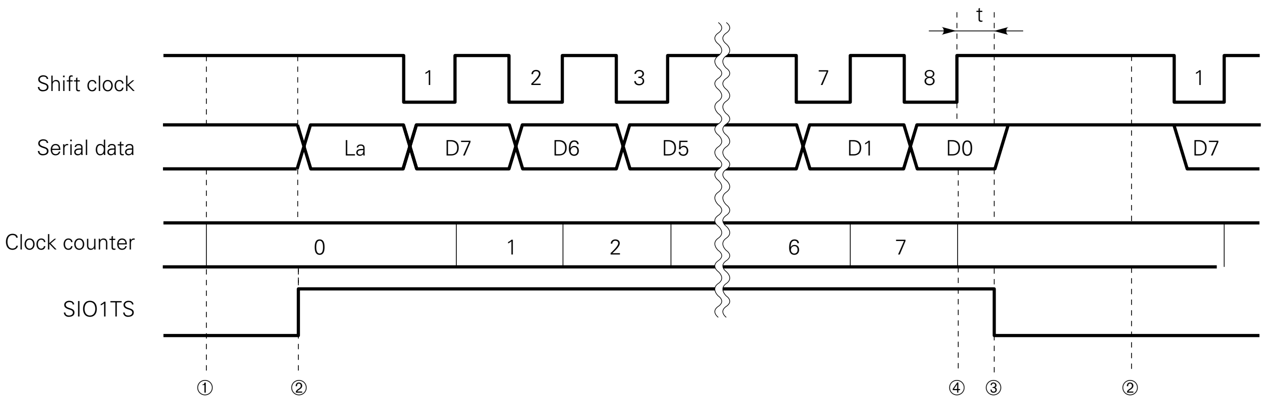
For the organization and functions of the serial I/O-1 mode selection register, see **Fig. 15-16**.

15.3.6 Serial Interface 1 Operation

(1) Timing chart

Fig. 15-18 shows the timing chart.

Fig. 15-18 Serial Interface 1 Timing Chart



Remarks ① Master transmit state setting (SIO1HIZ=1)
② Wait release
③ Wait timing
④ Interrupt timing

Caution As shown in Fig. 15-18, serial data output pin SO_1 outputs high level after the end of serial data transfer. The time until high level is output is (when the main clock is 8 MHz for both ① and ②):

① When internal clock selected as shift clock source
t = 312.5 ns

② When external clock selected as shift clock source
312.5 ≥ t ≥ 125 ns

(2) Clock counter operation

The clock counter initial value is "0", and is incremented each time the rising edge of the clock pin signal is detected thereafter. When the clock counter reaches "8", it returns to "1" and continues counting. The clock counter reset conditions are:

① Power-on reset
② Clock-stop instruction execution
③ "0" was written in the SIO1TS flag
④ Rising edge of shift clock when clock counter reaches "8" in the wait released state
⑤ CE reset

(3) Wait operation and cautions

When the wait state is released, serial data is output (at transmit operation) at the falling edge of the next clock and serial interface 1 remains in the wait released state for 8 clocks.

After 8 clocks are output, the shift clock pin is made high level and clock counter and presettable shift register 1 operation stops.

Note that if presettable shift register 1 is written or read when the serial interface is in the wait released state and the shift clock pin is high level, the data may not be set correctly.

If presettable shift register 1 is written or read when the serial interface is in the wait released state and the shift clock pin is low level, the contents of the MSB are output from the serial data output pin when a “PUT” instruction is executed.

If forced wait is set in the wait released state, serial interface 1 enters the wait state as soon as “0” is written in the SIO1TS flag.

Note that if wait release is set again in the wait released state, the clock counter is reset.

(4) Interrupt request timing

An interrupt request is issued when 8 clocks are transmitted (received).

(5) Shift clock generation timing

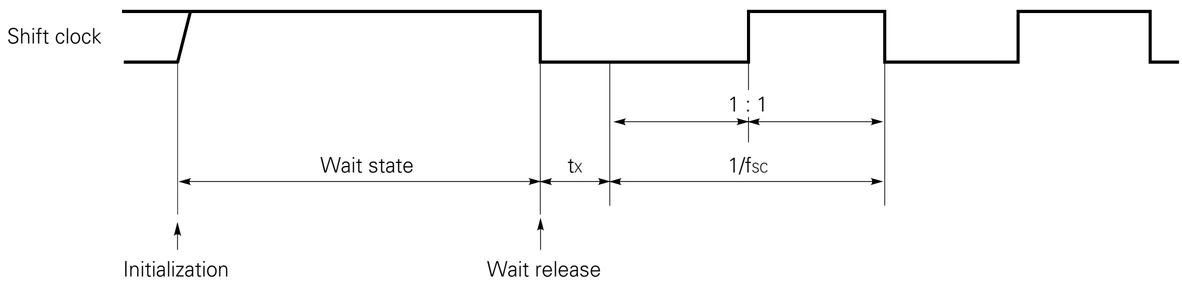
(a) When the initial state is released

“Initial state” refers to the time when internal clock operation is selected and the $P2D_0/\overline{SCK_1}$ pin is set to high level.

During the wait state, high level is output from the shift clock pin.

Wait can be released and the clock can be selected simultaneously.

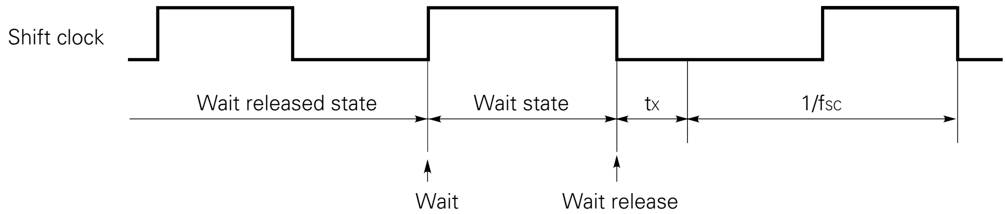
Fig. 15-19 Serial Interface 1 Shift Clock Generation Timing (1/4)



(b) When wait operation is performed

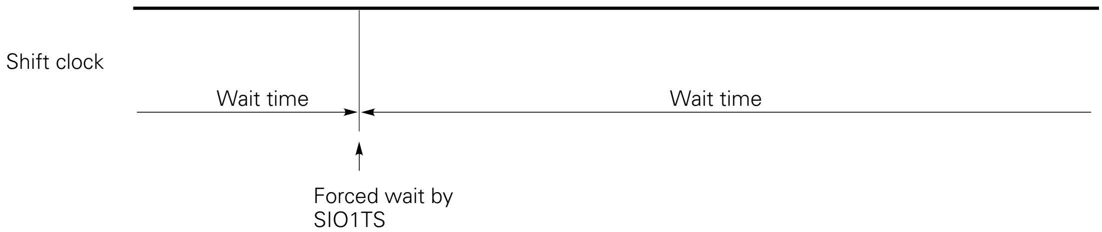
① When wait is set at 8th clock (normal operation)

Fig. 15-19 Serial Interface 1 Shift Clock Generation Timing (2/4)



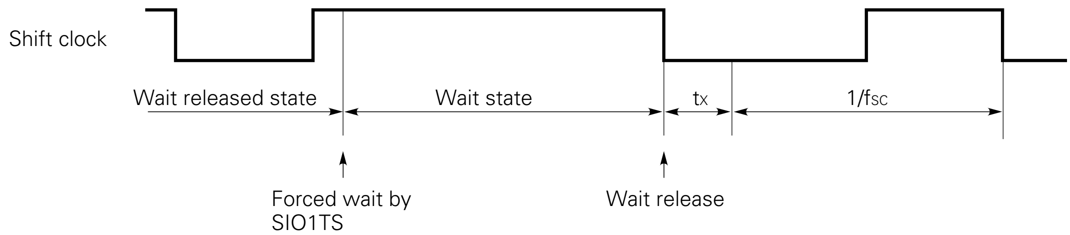
② When forced wait is set in wait state

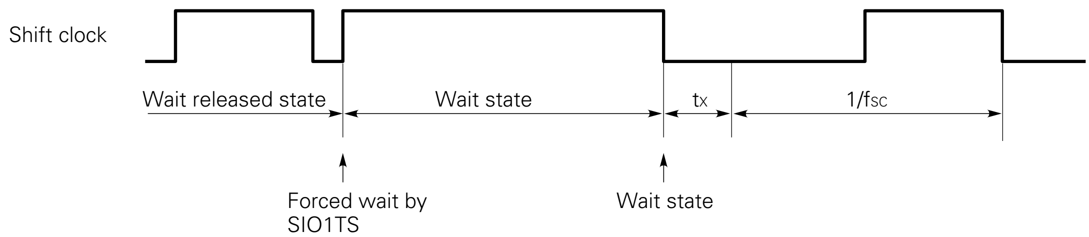
Fig. 15-19 Serial Interface 1 Shift Clock Generation Timing (3/4)



③ When forced wait is set in wait released state

Fig. 15-19 Serial Interface 1 Shift Clock Generation Timing (4/4)





Caution The value of t_X in Fig. 15-19 is normally 187.5 ns.
However, when 1 MHz is selected as the serial clock, t_X becomes 687.5 ns (187.5 + 500 ns).

④ **When wait release is specified in wait released state**

The clock output waveform does not change. The clock counter is not reset either. However, do not change the clock frequency in the wait released state.

15.3.7 Data Write and Data Read Cautions

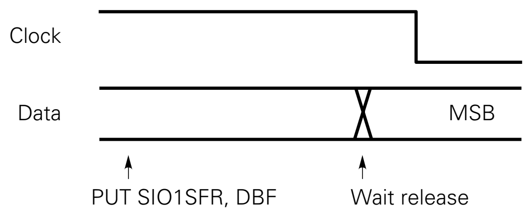
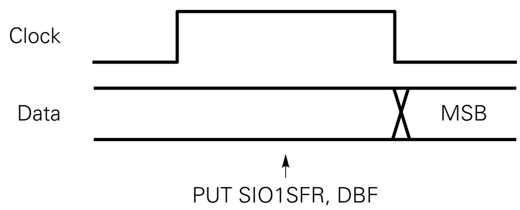
Data is written to presettable shift register 1 with the "PUT SIO1SFR, DBF" instruction.

Data is read from presettable shift register 1 with the "GET DBF, SIO1SFR" instruction.

Write and read data in the wait state. In the wait released state, the data may not be set and read correctly, depending on the state of the shift clock pin.

The data write and read timing and cautions are given below.

Table 15-8 Presettable Shift Register 1 Data Read and Data Write Operations and Cautions

<table>
<tr><th colspan="2">State at PUT/GET execution</th><th>State of shift clock pin</th><th>Operation of presettable shift register 1</th></tr>
<tr><td rowspan="2">Wait state</td><td>Read (GET)</td><td rowspan="2">● Floating when an external clock is used
● High level output when the internal clock is used
SIO1SFR are destroyed.)</td><td>Normal read</td></tr>
<tr><td>Write (PUT)</td><td>Normal write
The contents of the MSB is output the next time the wait state is released. (At transmit operation)
(If the clock pin is low level in the wait state when an external clock is used, data is not written normally. The contents of

</td></tr>
<tr><td rowspan="4">Wait released state</td><td rowspan="2">Read (GET)</td><td>High level</td><td>Normal read</td></tr>
<tr><td>Low level</td><td>Normal read</td></tr>
<tr><td rowspan="2">Write (PUT)</td><td>High level</td><td>Normal write
Outputs the contents of the MSB at the falling edge of the shift clock. The clock counter is not reset.

</td></tr>
<tr><td>Low level</td><td>Not written normally.
The contents of SIO1SFR are destroyed.</td></tr>
</table>

15.3.8 Serial Interface 1 Operation

Table 15-9 summarizes serial interface 1 operation.

Table 15-9 Serial Interface 1 Operation

<table>
<tr><td colspan="2" rowspan="2">Operation mode
Item</td><td colspan="4">Serial interface 1</td></tr>
<tr><td colspan="2">Slave operation
(both SIO1CK1 and SIOICK0 are 0)</td><td colspan="2">Master operation
(both SIO1CK1 and SIO1CK0 are not 0)</td></tr>
<tr><td rowspan="6">State of each pin</td><td rowspan="2">$\overline{SCK}_1$/P2D$_0$</td><td>Waiting
(SIO1TS = 0)</td><td>Wait released
(SIO1TS = 1)</td><td>Waiting
(SIO1TS = 0)</td><td>Wait released
(SIO1TS = 1)</td></tr>
<tr><td>● When P2DBIO0 = 0, works as a general-purpose input port, and is floating.
● When P2DBIO0 = 1, works as a general-purpose output port. Outputs the contents of the output latch.</td><td>● When P2DBIO0 = 0, is floating and waiting for external clock input.
● When P2DBIO0 = 1, works as a general-purpose output port. Outputs the contents of the output latch.</td><td>Outputs high level regardless of P2DIO0. When the state of the pin is read at this time, the contents of the output latch are read.</td><td>Outputs the internal clock regardless of P2DBIO0. When the state of the pin is read at this time, the contents of the output latch are read.</td></tr>
<tr><td rowspan="3">SO$_1$/P2D$_1$</td><td colspan="4">SIO1HIZ = 0
● When P2DBIO1 = 0, works as a general-purpose input port and is floating.
● When P2DBIO1 = 1, works as a general-purpose output port. Outputs the contents of the output latch.</td></tr>
<tr><td colspan="4">SIO1HIZ = 1</td></tr>
<tr><td>Outputs high level regardless of P2DBIO1. When the state of the pin is read at this time, the contents of the output latch are read.</td><td>Outputs serial data regardless of P2DBIO1. When the state of the pin is read at this time, the contents of the output latch are read.</td><td>Outputs high level regardless of P2DBIO1. When the state of the pin is read at this time, the contents of the output latch are read.</td><td>Outputs serial data regardless of P2DBIO1. When the state of the pin is read at this time, the contents of the output latch are read.</td></tr>
<tr><td>SI$_1$, P2D$_2$</td><td colspan="4">● When P2DBIO2 = 0, is floating and waiting for external data input.
● When P2DBIO2 = 1, works as a general-purpose output port. Outputs the contents of the output latch.</td></tr>
<tr><td>Clock counter</td><td></td><td colspan="4">Incremented at the rising edge of the $\overline{SCK}_1$ pin signal.</td></tr>
<tr><td rowspan="2">Presettable shift register 1 operation</td><td>Output</td><td colspan="4">● When P2DBIO2 = 0, not output.
● When P2DBIO2 = 1, shifts and outputs the data from the SO$_1$ pin from the MSB at the falling edge of the $\overline{SCK}_1$ pin signal.</td></tr>
<tr><td>Input</td><td colspan="4">Shifts and inputs the data from the LSB at the rising edge of the $\overline{SCK}_1$ pin signal regardless of P2DBIO2.
However, when P2DBIO2 = 1, outputs the contents of the output latch from the SI$_1$ pin.</td></tr>
</table>

15.3.9 State When Serial Interface 1 Is Reset

(1) At power-on reset

All the pins are set to general-purpose input ports.
The contents of presettable shift register 1 are undefined.

(2) At clock-stop

All the pins are set to general-purpose input ports.
The contents of presettable shift register 1 retain their previous state.

(3) At CE reset

All the pins are set to general-purpose input ports.
The contents of presettable shift register 1 retain their previous state.

(4) At halt

All the pins retain their set states.
The internal clock stops output when a HALT instruction is executed.
If an external clock is used, operation continues even if a HALT instruction is executed.

16. IMAGE DISPLAY CONTROLLER (IDC)

The IDC is used to display the channel No., volume, timer clock, etc. on a TV screen.

16.1 GENERAL

16.1.1 Configuration

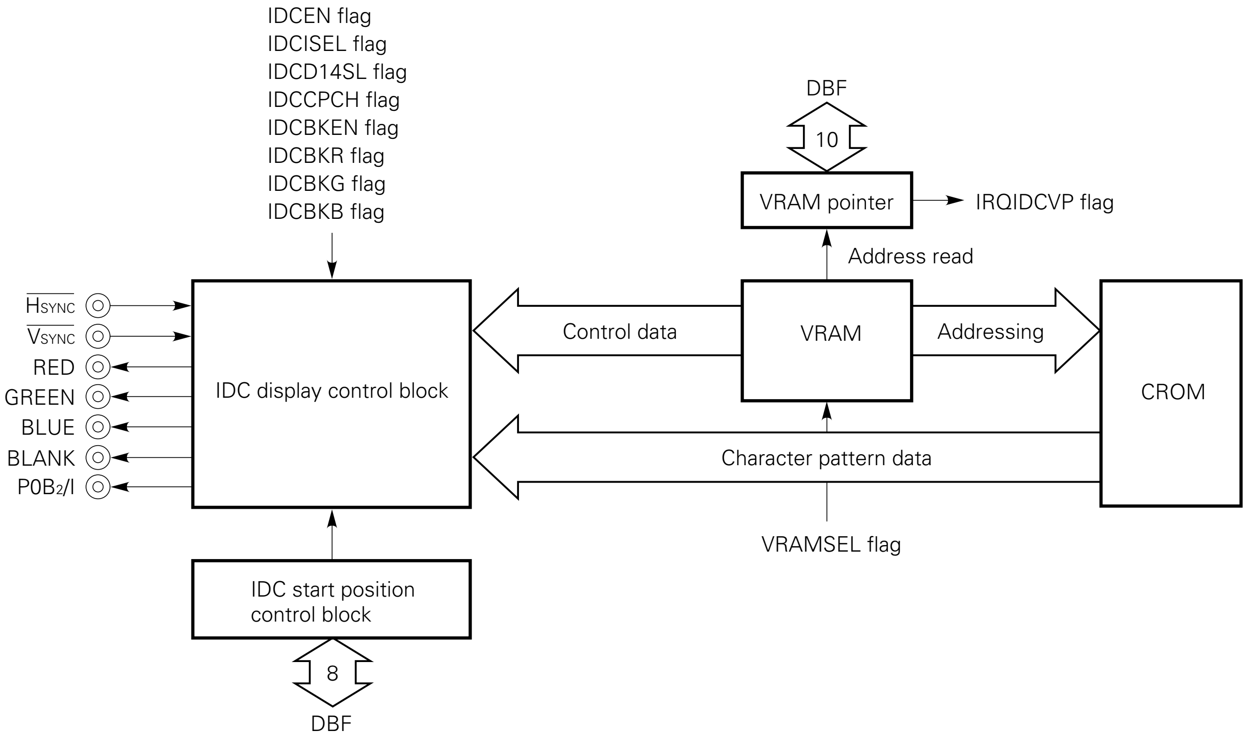
Fig. 16-1 outlines the IDC.

The display pattern is set in the CROM (Character ROM) area by program.

The VRAM (Video RAM) stores the data for selecting the actual display pattern from the CROM.

VRAM is allocated to BANK2 of the data memory. (See **Fig. 4-2**.)

Fig. 16-1 IDC



Remarks
1. IDCEN (bit 0 of IDC enable register: see **Fig. 16-2**) sets IDC display ON/OFF.
2. IDCISEL (bit 2 of IDC mode selection register: see **Fig. 16-3**) selects the POB2/I pin function.
3. IDCD14SL (bit 1 of IDC mode selection register: see **Fig. 16-3**) sets the number of vertical dots of the display character.
4. IDCCPCH (bit 0 of IDC mode selection register: see **Fig. 16-3**) sets whether or not there is a space between display characters.
5. IDCBKEN (bit 3 of IDC background selection register: see **Fig. 16-4**) sets whether or not a screen background is displayed.
6. IDCBKR, IDCBKG, and IDCBKB (bits 2, 1 and 0 of IDC background selection register: see **Fig. 16-4**) set the screen background color.
7. VRAMSEL (bit 3 of IDC mode selection register: see **Fig. 16-3**) sets whether or not there is a VRAM area.

16.1.2 IDC Functions

Table 16-1 summarizes the IDC functions.

Table 16-1 IDC Functions

Item	Function	Operation data
Number of display characters	Maximum 192 characters/screen (full screen possible by program)	—
Display position adjust-ment range	Within horizontal 24 characters, vertical 15 rows (8 lines × 24 columns mode)	Control data
Display format	16 × 16 dot mode: 15 lines × 24 columns	IDCD14SL flag
	14 × 16 dot mode: 17 lines × 24 columns	
Character set (font)	255 kinds (user-programmable)	Character pattern data
Character size	Vertical: 14 sizes (1-14 times, line units)	Control data
	Horizontal: 24 sizes (1-24 times, character units)	
Space between characters	0/2 bit (The size of one dot depends on the character size.)	IDCCPCH flag
Character color	16 kinds (character units)	Character pattern data
Character background color	8 kinds (character units)	Control data
Screen background color	8 kinds (set for 1 screen)	IDCBKEN flag IDCBKR flag IDCBKG flag IDCBKB flag
Character rimming	Rim (character units)	Control data Character pattern data
	Reverse video, rounding (character units)	Character pattern data

16.2 IDC DISPLAY CONTROL BLOCK

The IDC display control block controls IDC display on/off, VRAM, space between display characters, display format, I pin use, and the screen background color.

16.2.1 IDC Display Control Block Control Registers

The IDC display control block is controlled by IDC enable register, IDC mode selection register, and IDC background selection register.

Figs. 16-2 to 16-4 show the organization and functions of each register.

Fig. 16-2 Configuration of IDC Enable Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
IDC enable register	0	0	0	IDCEN	31H	R/W

IDCEN: Sets IDC display on/off.

IDCEN	
0	Display off
1	Display on

b_3 to b_1: Fixed to 0.

Upon reset		b_3	b_2	b_1	b_0
	Power-on	0	0	0	0
	Clock stop	↓	↓	↓	0
	CE	↓	↓	↓	0

Caution When setting the IDCEN flag to "1" (at the start of display), do it while the vertical synchronizing signal is high level (vertical flyback time, $\overline{V_{SYNC}}$, is low).

Fig. 16-3 Configuration of IDC Mode Selection Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
IDC mode selection register	VRAMSEL	IDCISEL	IDCD14SL	IDCCPCH	33H	R/W

IDCCPCH (b_0): Sets whether or not there is a space between display characters.

b_0	
0	No space
1	Space

IDCD14SL (b_1): Sets the number of vertical dots of the display character.

b_1	
0	16 dots
1	14 dots

IDCISEL (b_2): Selects the function of the POB_2/I pin.

b_2	
0	General-purpose I/O port
1	I pin

VRAMSEL (b_3): Enables/disables the VRAM.

b_3	
0	VRAM disable
1	VRAM enable

Upon 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

Fig. 16-4 Configuration of IDC Background Selection Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
IDC background selection register	IDCBKEN	IDCBKR	IDCBKG	IDCBKB	30H	R/W

IDCBKR	IDCBKG	IDCBKB	Sets the screen background color.
0	0	0	No background (black)
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

IDCBKEN	Sets whether or not screen background is displayed.
0	Do not display screen background color.
1	Display screen background color.

Upon 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

16.2.2 Display Format

When the IDCD14SL flag of the IDC mode selection register is set (1), 14 vertical × 16 horizontal dots is selected. When it is reset (0), 16 vertical × 16 horizontal dots is selected.

When you want to display 17 lines on one screen, select the 14 × 16 dots mode.

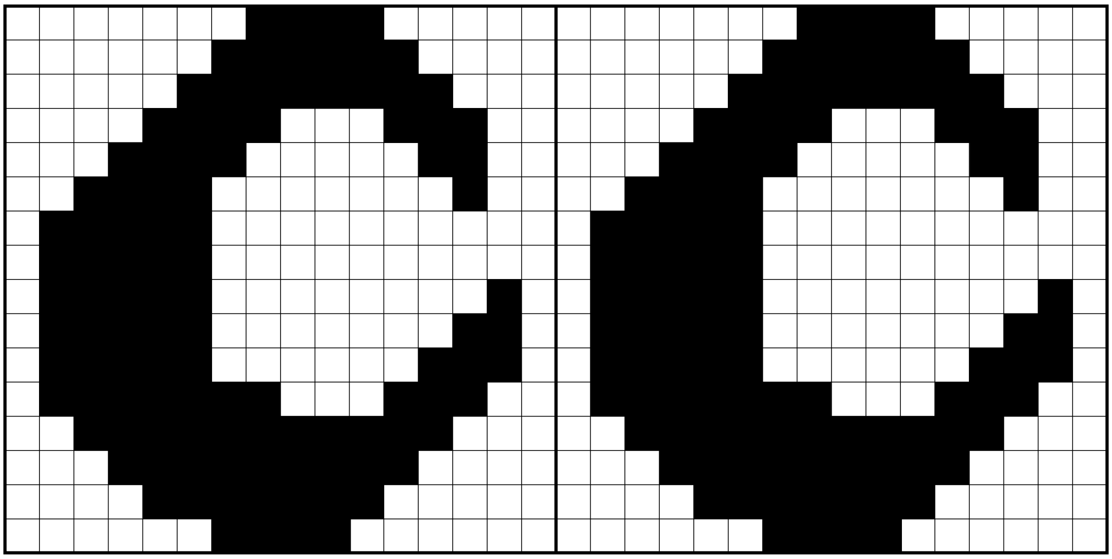
16.2.3 Space between Characters

A 2-dot space can be set in front of a character by setting the IDCCPCH flag of the IDC mode selection register. The size of the space depends on the character size (horizontal).

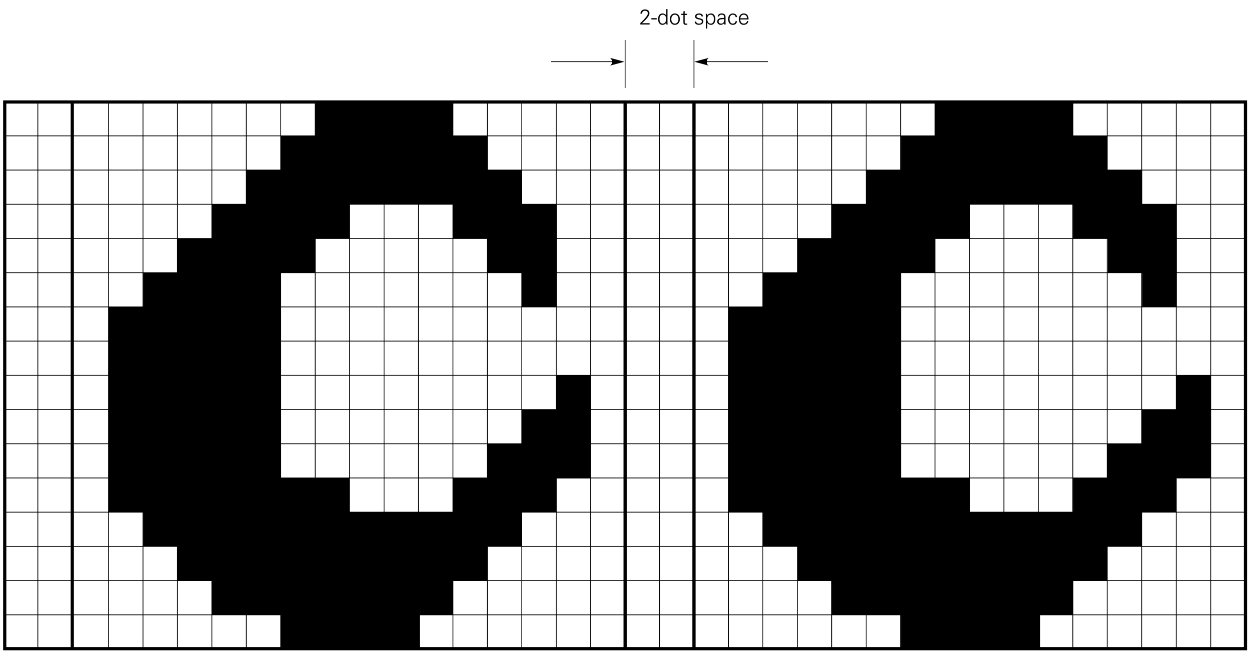
When a space is not set, kanji and other characters and graphics can be displayed by combining two or more characters.

Fig. 16-5 Space between Characters

- **When IDCCPCH flag is "0"**



- **When IDCCPCH flag is "1"**



16.2.4 Screen Background Color

The background color of the entire display screen can be set by manipulating the IDC background selection register flags.

The character color and screen background color can be set simultaneously. The display priority is:

Character color < character background color < screen background color

16.3 IDC START POSITION CONTROL BLOCK

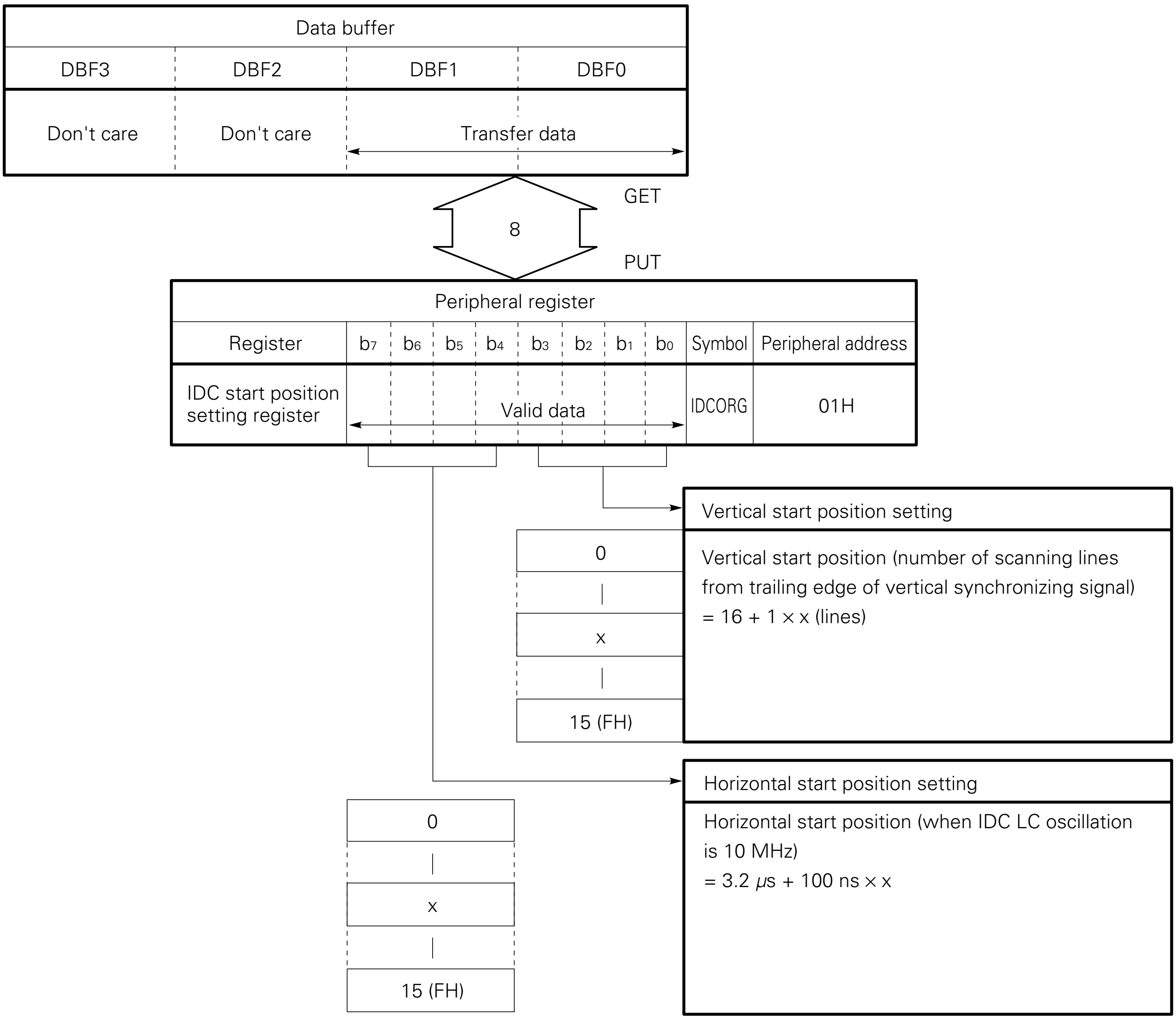
The IDC start position control block can shift the display position of the entire screen by setting data in the IDC start position setting register (IDCORG: peripheral address 01H).

16.3.1 Configuration of IDC Start Position Setting Register

Fig. 16-6 shows the configuration of the IDC start position setting register.

Set data when the VSYNC signal is low level.

Fig. 16-6 Configuration of IDC Start Position Setting Register



16.3.2 Horizontal Start Position Setting

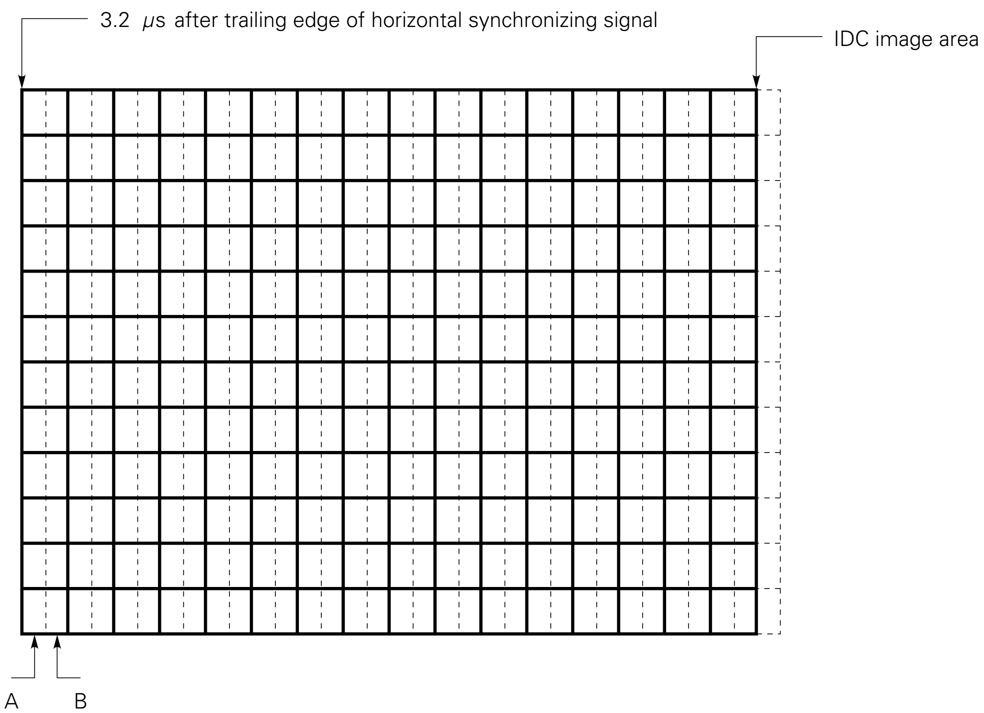
When the data set in the horizontal start position setting register is "0H" and OSC_{IN} = 10 MHz, the horizontal start position is set to 3.2 μs (2 characters) after the trailing edge of the horizontal synchronizing signal.

Each time this data is increased by "1", the horizontal start position is shifted 100 ns (1 dot of minimum size character) to the right. That is, it can be expressed as follows:

Horizontal start position = 3.2 μs + 100 ns × (horizontal start position setting data)

Referring to Fig. 16-7, assume that the position is A when the horizontal start position setting register set value is "0H." When the set value is made "1H", the horizontal start position moves 100 ns to the right and becomes position B.

Fig. 16-7 Horizontal Direction Movement



Remarks  : Image area when horizontal position set data is "0H"

 : Image area when horizontal position set data is "1H"

16.3.3 Vertical Start Position Setting

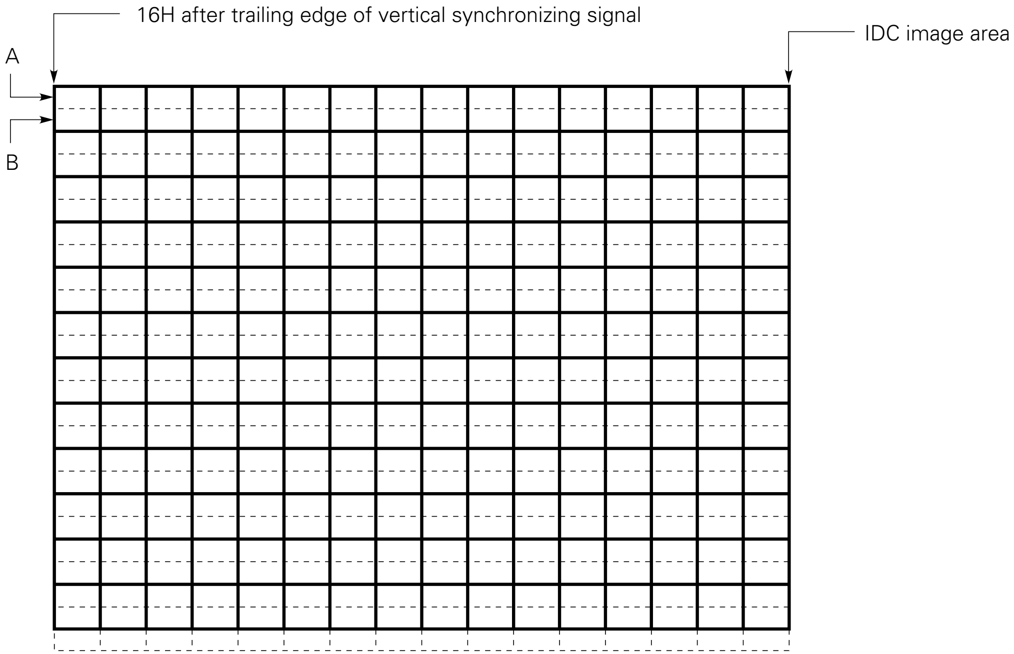
When the data set in the vertical start position setting register is "0H", the vertical start position is set to 16 scanning lines after the trailing edge of the vertical synchronizing signal.

Each time this data is increased by "1", the vertical start position is moved down 1 line. This can be expressed as follows:

Vertical start position = 16 + 1 × (vertical start position setting data)

Referring to Fig. 16-8, assume that the vertical start position is A when the vertical start position setting register set value is "0H." When the set value is made "1H", the vertical start position is moved down 1 line to position B.

Fig. 16-8 Vertical Direction Movement



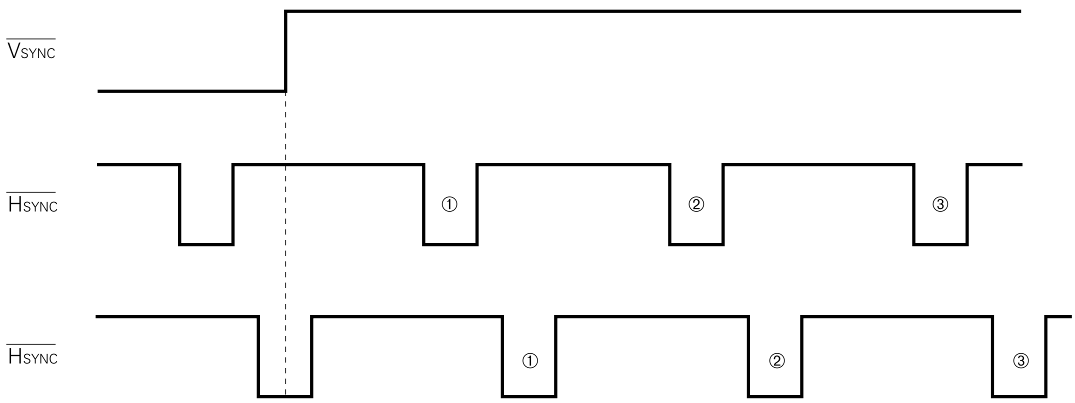
Remarks  : Image area when vertical position setting data is "0H"

 : Image area when vertical position setting data is "1H"

The display character vertical start position is determined by the vertical start position register.

The vertical start position (counted by the number of horizontal scanning lines) at this time is selected as shown in Fig. 16-9, according to the state of the V_{SYNC} and H_{SYNC} signals that are input at the $\overline{V_{SYNC}}$ and $\overline{H_{SYNC}}$ pins. In short, the first H_{SYNC} signal after the rising edge of the V_{SYNC} signal is counted as the first line.

Fig. 16-9 How Vertical Start Position Is Counted



Remark ①, ②, or ③ indicates the number of each scanning line.

16.4 CROM (CHARACTER ROM)

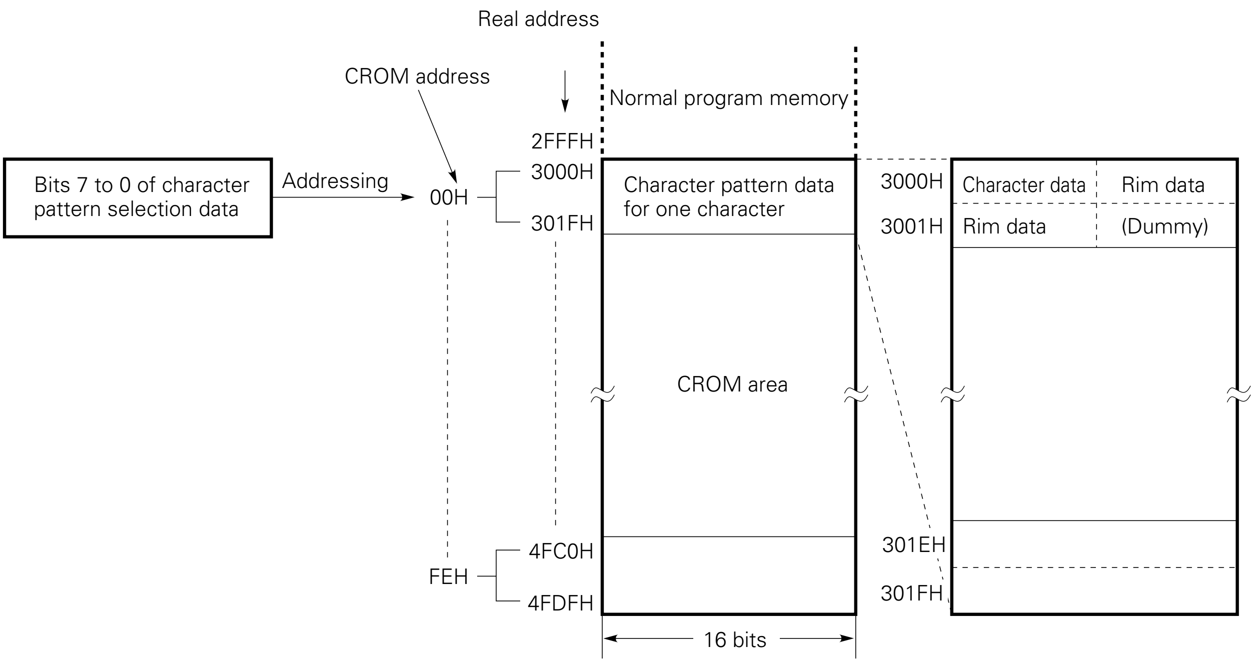
The CROM stores the IDC display pattern data (character pattern data).

Fig. 16-10 shows the configuration of the CROM.

CROM is allocated to CROM area (3000H-4FDFH) in program memory (ROM). The CROM area cannot be used as normal program memory. CROM is addressed with VRAM character pattern selection data.

CROM stores the data of 4080 steps (4080 × 24 bits: 255 characters), but since one address occupies 32 bits, its actual capacity is 8160 × 16 bits. (See **Fig. 2-2**.)

Fig. 16-10 Configuration of CROM



16.4.1 Character Pattern Data Configuration

The character pattern data is used for displaying characters and graphic patterns.

One character consists of 16 horizontal dots by 16 vertical dots. Since the data for 16 horizontal dots corresponds to one step of CROM, the character pattern data for one character consists of 16 steps (16 × 24 bits).

Fig. 16-11 shows the configuration of the character patterns.

The 8 low-order bits of the character pattern data are fixed at "1" (dummy).

The character data that stores the actual display pattern consists of 8 bits. The bits corresponding to the dots that are lit are set to "1" and the bits corresponding to the dots that are not lit are set to "0". Two dots of the actual display pattern correspond to one bit of character pattern data. A character pattern data is formed by dot image by superimposing 16-bit rim data (in dot units) onto the character data.

If a 17K Series assembler (AS17K) is used, data like that shown in Fig. 16-12 can be generated automatically by using a DCP pseudo instruction. A display pattern generation development tool (IDC font editor) is also available. Use this tool, if necessary.

Fig. 16-11 Configuration of Character Data Pattern

b_{31} – b_{24}	b_{23} – b_8	b_7 – b_0
Character data	Rim data	Dummy (Fixed to 1.)

Examples of character pattern data are shown in Fig. 16-12. In this data, "0" corresponds to □ and "1" corresponds to ■. The control data specifies the character size, position, and color.

Fig. 16-12 Character Pattern Data Setting (Character: “N”)

● Character pattern

Real address	b_{31} — b_{24}	b_{23} — b_{16}	b_{15} — b_8	b_7 — b_0
×××0H	00000000	00000100		
×××1H			00000100	11111111
×××2H	00100010	00001010		
×××3H			00001010	11111111
×××4H	00100011	00010010		
×××5H			00010001	11111111
×××6H	01110010	00100001		
×××7H			00010010	11111111
×××8H	01110010	00100001		
×××9H			00010100	11111111
×××AH	00110110	00010000		
×××BH			10100100	11111111
×××CH	00110110	00010000		
×××DH			10100100	11111111
×××EH	00111100	00010000		
×××FH			01101000	11111111
×××0H	01111100	00100000		
×××1H			01001000	11111111
×××2H	01111100	00100100		
×××3H			00001000	11111111
×××4H	01011100	01001100		
×××5H			00010000	11111111
×××6H	01011100	01001010		
×××7H			00010000	11111111
×××8H	11011100	10010010		
×××9H			00010000	11111111
×××AH	11001000	10001001		
×××BH			00100000	11111111
×××CH	11001000	10010001		
×××DH			00100000	11111111
×××EH	00000000	01100000		
×××FH			11000000	11111111

|← Character data →|← Rim data →|← Dummy →|

● Setting when a DCP pseudo instruction is used

```
DCP '          #             #       '
DCP '        # O #         # O #     '
DCP '      # O O #         # O O O # '
DCP '    # O O O O #       # O O #   '
DCP '    # O O O O #       # O #     '
DCP '      # O O O O #   # O O #     '
DCP '      # O O O O #   # O O #     '
DCP '      # O O O O O # # O #       '
DCP '    # O O O O O O # O O #       '
DCP '    # O O # O O O O O O #       '
DCP '  # O O # O O O O O O #         '
DCP '  # O O #   # O O O O #         '
DCP '# O O #     # O O O O #         '
DCP '# O O O #       # O O #         '
DCP '# O O #       # O O #           '
DCP '  # #         # #               '
```

Conversion with a DCP pseudo instruction

```
; Display_N :
DB    04H, 04H, 00H, 0FFH
DB    0AH, 0AH, 22H, 0FFH
DB    11H, 12H, 23H, 0FFH
DB    12H, 21H, 72H, 0FFH
DB    14H, 21H, 72H, 0FFH
DB    0A4H, 10H, 36H, 0FFH
DB    0A4H, 10H, 36H, 0FFH
DB    68H, 10H, 3CH, 0FFH
DB    48H, 20H, 7CH, 0FFH
DB    08H, 24H, 7CH, 0FFH
DB    10H, 4CH, 5CH, 0FFH
DB    10H, 4AH, 5CH, 0FFH
DB    10H, 92H, 0DCH, 0FFH
DB    20H, 89H, 0C8H, 0FFH
DB    20H, 91H, 0C8H, 0FFH
DB    0C0H, 60H, 00H, 0FFH
```

16.4.2 Definition of Character Pattern Data with Assembler

Character data can be easily defined with a 17K Series assembler by using a DCP pseudo instruction. The DCP pseudo instruction description is shown below.

(1) Format

Symbol field	Mnemonic field	Operand field	Comment field
[Label:]	DCP	'display pattern'	[:comment]

(2) Description

The display pattern uses only the three characters "O", "#", and " " (blank). Sixteen characters are described on one line. If a character other than these three characters is described, or if less than 16 characters are described, an error is generated.

Each of these three characters corresponds to one dot of the display pattern and has the following meaning:

"O" : Dot to be displayed (lit)
"#" : Rim
" " : Blank

(3) Assembly method

Before a file describing characters with a DCP pseudo instruction is assembled, it must be converted to a source file. Perform this conversion as follows:

① Create a file defining the character using a DCP pseudo instruction. Make the extension DCP.
② Convert the file created at step ① to a source file by executing program DCP.EXE as follows:

DCP.EXE␣×××.DCP
(␣: space, ×××.DCP: File name of created file)

③ When the program ends, a ×××.ASM file is created. Assemble this file.

16.5 VRAM (VIDEO RAM)

16.5.1 General

Fig. 16-13 shows the configuration of the VRAM. Fig. 16-14 shows VRAM bank specification.

The VRAM stores the following three kinds of data:

- Character pattern selection data
- Carriage return data (C/R)
- Control data 1 and 2

The VRAM is allocated at addresses 00H-3FH of BANK 2 of data memory, and is enabled only when the VRAMSEL flag is "1". When the VRAMSEL flag is set to "1", neither VRAM nor RAM exist at addresses 30H-3FH of data memory and "0" is always read from this area.

VRAM consists of 14 banks designated VRAMBANK0 to VRAMBANKD. (See **Fig. 4-2**.)

The data at address 73H of BANK 2 of data memory specifies the VRAM BANK.

One item of VRAM data consists of data at 3 addresses (12 bits). Each bank consists of 48 nibbles. Two-hundred twenty-three data items can be set as VRAM data.

Fig. 16-13 Configuration of VRAM

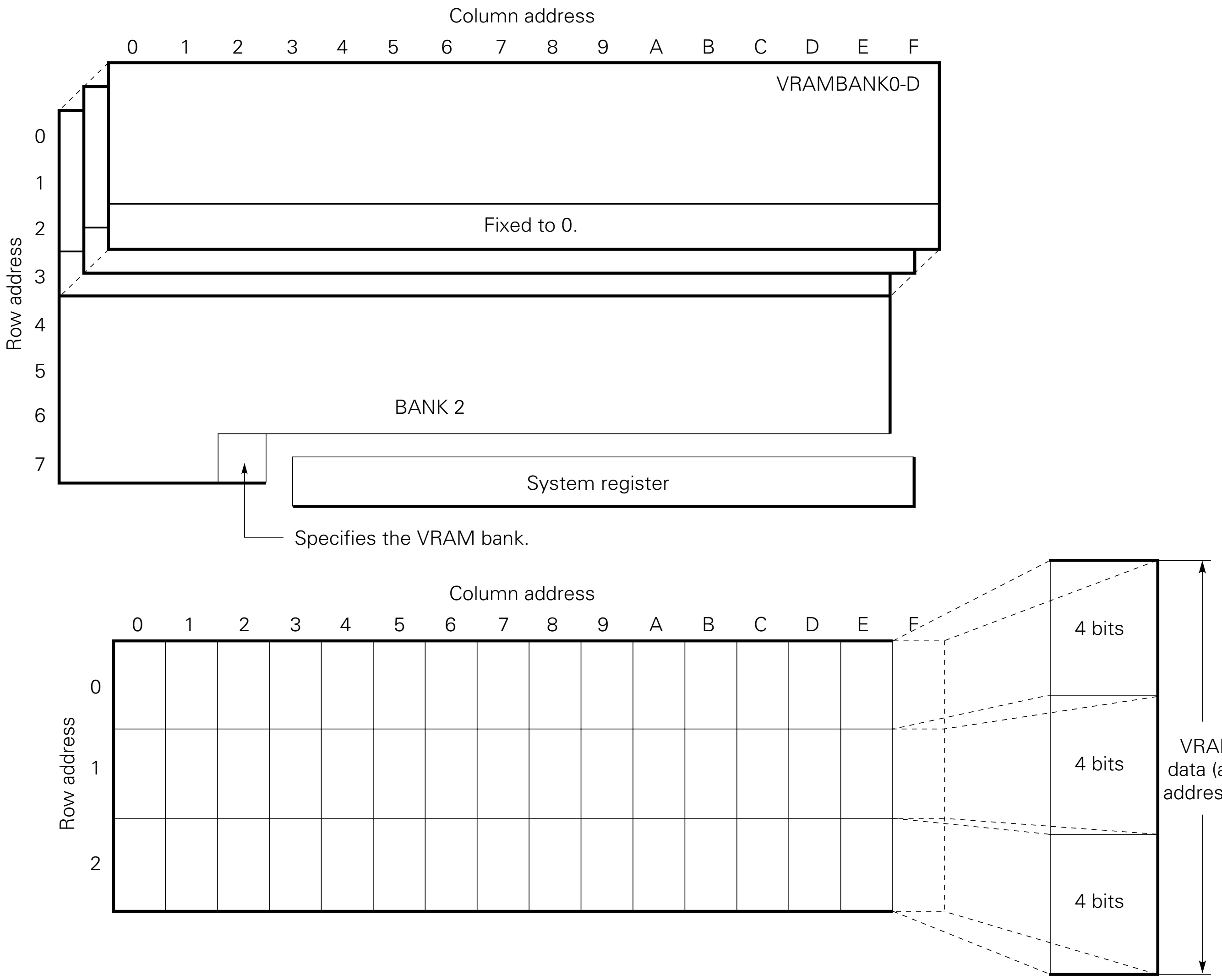
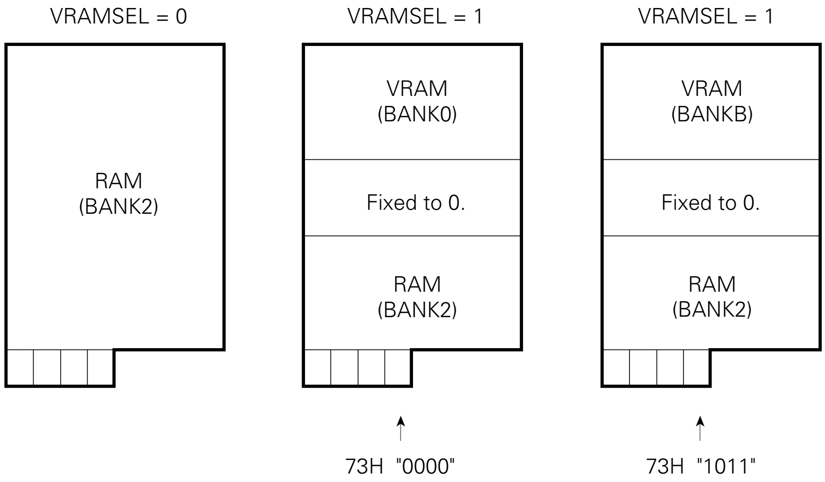


Fig. 16-14 VRAM Bank Specification



16.5.2 Configuration of VRAM Data

Fig. 16-15 shows the configuration of the VRAM data.

One item of VRAM data consists of 12 bits, and is divided into an ID field and a data field.

Fig. 16-15 Configuration of VRAM Data

Address	0 × H				1 × H				2 × H			
	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Name	ID	Data field										

- ID field

  The ID field represents the state of the character pattern selection data. The ID field settings and character pattern selection data states are shown in Table 16-2.

  For details, see **Section 16.5.5.**

Table 16-2 ID Field Settings and Character Pattern Selection Data States

ID field setting	Character pattern selection data state
0 **Note**	When combined with control data 2
0 or 1	When combined with control data 1

Note Becomes carriage return data only when the ID field is 0 and the data field has the following value:

- 10111111111 ··· VRAM pointer reset C/R
- 11011111111 ··· Line C/R
- 11111111111 ··· Screen C/R

16.5.3 Character Pattern Selection Data

Fig. 16-16 shows the configuration of the character pattern selection data.

A “character pattern” is data that specifies the shape and other attributes of the character displayed on a television set or other screen and is stored in the CROM (Character ROM).

Bits 8 to 10 of the character pattern selection data specify the character color. Bits 0 to 7 are the CROM address. Table 16-3 shows the correspondence between the CROM address specified by the character pattern selection data and the real address. For the CROM, see **Section 16.4**.

Fig. 16-16 Configuration of Character Pattern Selection Data

b_{11}	b_{10} – b_8	b_7 – b_0
ID	Character color	CROM address

b_{10}	b_9	b_8	Set the character color.
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

Table 16-3 CROM Address Specified by Character Pattern Selection Data and Real Address (1/2)

CROM address	Real address	CROM address	Real address	CROM address	Real address	CROM address	Real address
00H	3000H-301FH	20H	3400H-341FH	40H	3800H-381FH	60H	3C00H-3C1FH
01H	3020H-303FH	21H	3420H-343FH	41H	3820H-383FH	61H	3C20H-3C3FH
02H	3040H-305FH	22H	3440H-345FH	42H	3840H-385FH	62H	3C40H-3C5FH
03H	3060H-307FH	23H	3460H-347FH	43H	3860H-387FH	63H	3C60H-3C7FH
04H	3080H-309FH	24H	3480H-349FH	44H	3880H-389FH	64H	3C80H-3C9FH
05H	30A0H-30BFH	25H	34A0H-34BFH	45H	38A0H-38BFH	65H	3CA0H-3CBFH
06H	30C0H-30DFH	26H	34C0H-34DFH	46H	38C0H-38DFH	66H	3CC0H-3CDFH
07H	30E0H-30FFH	27H	34E0H-34FFH	47H	38E0H-38FFH	67H	3CE0H-3CFFH
08H	3100H-311FH	28H	3500H-351FH	48H	3900H-391FH	68H	3D00H-3D1FH
09H	3120H-313FH	29H	3520H-353FH	49H	3920H-393FH	69H	3D20H-3D3FH
0AH	3140H-315FH	2AH	3540H-355FH	4AH	3940H-395FH	6AH	3D40H-3D5FH
0BH	3160H-317FH	2BH	3560H-357FH	4BH	3960H-397FH	6BH	3D60H-3D7FH
0CH	3180H-319FH	2CH	3580H-359FH	4CH	3980H-399FH	6CH	3D80H-3D9FH
0DH	31A0H-31BFH	2DH	35A0H-35BFH	4DH	39A0H-39BFH	6DH	3DA0H-3DBFH
0EH	31C0H-31DFH	2EH	35C0H-35DFH	4EH	39C0H-39DFH	6EH	3DC0H-3DDFH
0FH	31E0H-31FFH	2FH	35E0H-35FFH	4FH	39E0H-39FFH	6FH	3DE0H-3DFFH
10H	3200H-321FH	30H	3600H-361FH	50H	3A00H-3A1FH	70H	3E00H-3E1FH
11H	3220H-323FH	31H	3620H-363FH	51H	3A20H-3A3FH	71H	3E20H-3E3FH
12H	3240H-325FH	32H	3640H-365FH	52H	3A40H-3A5FH	72H	3E40H-3E5FH
13H	3260H-327FH	33H	3660H-367FH	53H	3A60H-3A7FH	73H	3E60H-3E7FH
14H	3280H-329FH	34H	3680H-369FH	54H	3A80H-3A9FH	74H	3E80H-3E9FH
15H	32A0H-32BFH	35H	36A0H-36BFH	55H	3AA0H-3ABFH	75H	3EA0H-3EBFH
16H	32C0H-32DFH	36H	36C0H-36DFH	56H	3AC0H-3ADFH	76H	3EC0H-3EDFH
17H	32E0H-32FFH	37H	36E0H-36FFH	57H	3AE0H-3AFFH	77H	3EE0H-3EFFH
18H	3300H-331FH	38H	3700H-371FH	58H	3B00H-3B1FH	78H	3F00H-3F1FH
19H	3320H-333FH	39H	3720H-373FH	59H	3B20H-3B3FH	79H	3F20H-3F3FH
1AH	3340H-335FH	3AH	3740H-375FH	5AH	3B40H-3B5FH	7AH	3F40H-3F5FH
1BH	3360H-337FH	3BH	3760H-377FH	5BH	3B60H-3B7FH	7BH	3F60H-3F7FH
1CH	3380H-339FH	3CH	3780H-379FH	5CH	3B80H-3B9FH	7CH	3F80H-3F9FH
1DH	33A0H-33BFH	3DH	37A0H-37BFH	5DH	3BA0H-3BBFH	7DH	3FA0H-3FBFH
1EH	33C0H-33DFH	3EH	37C0H-37DFH	5EH	3BC0H-3BDFH	7EH	3FC0H-3FDFH
1FH	33E0H-33FFH	3FH	37E0H-37FFH	5FH	3BE0H-3BFFH	7FH	3FE0H-3FFFH

Table 16-3 CROM Address Specified by Character Pattern Selection Data and Real Address (2/2)

CROM address	Real address
80H	4000H-401FH
81H	4020H-403FH
82H	4040H-405FH
83H	4060H-407FH
84H	4080H-409FH
85H	40A0H-40BFH
86H	40C0H-40DFH
87H	40E0H-40FFH
88H	4100H-411FH
89H	4120H-413FH
8AH	4140H-415FH
8BH	4160H-417FH
8CH	4180H-419FH
8DH	41A0H-41BFH
8EH	41C0H-41DFH
8FH	41E0H-41FFH
90H	4200H-421FH
91H	4220H-423FH
92H	4240H-425FH
93H	4260H-427FH
94H	4280H-429FH
95H	42A0H-42BFH
96H	42C0H-42DFH
97H	42E0H-42FFH
98H	4300H-431FH
99H	4320H-433FH
9AH	4340H-435FH
9BH	4360H-437FH
9CH	4380H-439FH
9DH	43A0H-43BFH
9EH	43C0H-43DFH
9FH	43E0H-43FFH

CROM address	Real address
A0H	4400H-441FH
A1H	4420H-443FH
A2H	4440H-445FH
A3H	4460H-447FH
A4H	4480H-449FH
A5H	44A0H-44BFH
A6H	44C0H-44DFH
A7H	44E0H-44FFH
A8H	4500H-451FH
A9H	4520H-453FH
AAH	4540H-455FH
ABH	4560H-457FH
ACH	4580H-459FH
ADH	45A0H-45BFH
AEH	45C0H-45DFH
AFH	45E0H-45FFH
B0H	4600H-461FH
B1H	4620H-463FH
B2H	4640H-465FH
B3H	4660H-467FH
B4H	4680H-469FH
B5H	46A0H-46BFH
B6H	46C0H-46DFH
B7H	46E0H-46FFH
B8H	4700H-471FH
B9H	4720H-473FH
BAH	4740H-475FH
BBH	4760H-477FH
BCH	4780H-479FH
BDH	47A0H-47BFH
BEH	47C0H-47DFH
BFH	47E0H-47FFH

CROM address	Real address
C0H	4800H-481FH
C1H	4820H-483FH
C2H	4840H-485FH
C3H	4860H-487FH
C4H	4880H-489FH
C5H	48A0H-48BFH
C6H	48C0H-48DFH
C7H	48E0H-48FFH
C8H	4900H-491FH
C9H	4920H-493FH
CAH	4940H-495FH
CBH	4960H-497FH
CCH	4980H-499FH
CDH	49A0H-49BFH
CEH	49C0H-49DFH
CFH	49E0H-49FFH
D0H	4A00H-4A1FH
D1H	4A20H-4A3FH
D2H	4A40H-4A5FH
D3H	4A60H-4A7FH
D4H	4A80H-4A9FH
D5H	4AA0H-4ABFH
D6H	4AC0H-4ADFH
D7H	4AE0H-4AFFH
D8H	4B00H-4B1FH
D9H	4B20H-4B3FH
DAH	4B40H-4B5FH
DBH	4B60H-4B7FH
DCH	4B80H-4B9FH
DDH	4BA0H-4BBFH
DEH	4BC0H-4BDFH
DFH	4BE0H-4BFFH

CROM address	Real address
E0H	4C00H-4C1FH
E1H	4C20H-4C3FH
E2H	4C40H-4C5FH
E3H	4C60H-4C7FH
E4H	4C80H-4C9FH
E5H	4CA0H-4CBFH
E6H	4CC0H-4CDFH
E7H	4CE0H-4CFFH
E8H	4D00H-4D1FH
E9H	4D20H-4D3FH
EAH	4D40H-4D5FH
EBH	4D60H-4D7FH
ECH	4D80H-4D9FH
EDH	4DA0H-4DBFH
EEH	4DC0H-4DDFH
EFH	4DE0H-4DFFH
F0H	4E00H-4E1FH
F1H	4E20H-4E3FH
F2H	4E40H-4E5FH
F3H	4E60H-4E7FH
F4H	4E80H-4E9FH
F5H	4EA0H-4EBFH
F6H	4EC0H-4EDFH
F7H	4EE0H-4EFFH
F8H	4F00H-4F1FH
F9H	4F20H-4F3FH
FAH	4F40H-4F5FH
FBH	4F60H-4F7FH
FCH	4F80H-4F9FH
FDH	4FA0H-4FBFH
FEH	4FC0H-4FDFH

16.5.4 Carriage Return Data (C/R)

Fig. 16-17 shows the kinds of carriage return data.

There are the following three kinds of carriage return data. Data other than these functions as character pattern selection data.

- Line C/R
- VRAM pointer reset C/R
- Screen C/R

When displaying data exceeding the VRAM capacity (extended display mode) on one page, set VRAM pointer reset C/R data at the end of the VRAM. For a description of the extended display mode, see **Section 16.8**.

Fig. 16-17 Kinds of Carriage Return Data

	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Line C/R	0	1	1	0	1	1	1	1	1	1	1	1
VRAM pointer reset C/R	0	1	0	1	1	1	1	1	1	1	1	1
Screen C/R	0	1	1	1	1	1	1	1	1	1	1	1

16.5.5 Control Data

Fig. 16-18 shows the configuration of the control data.

"Control data" is used for specifying the character size, display position, and color on the character pattern screen.

There are the following two kinds of control data:

- Control data specified at each line (control data 1)
- Control data specified for each character (control data 2)

Control data 1 is represented by 12 bits following VRAM address 0 (after screen C/R) and the line C/R.

Control data 2 is represented by 12 bits following the character data when the ID field is set to "1". In short, it is used as a pair with the character pattern selection data. Control data 2 modifies the character up to control data 2 directly preceding it or up to the screen C/R.

Specify control data 1 for each line whether or not it has changed.

① Control data 1

b11 – b8	b7 – b4	b3 – b0
Vertical size	Vertical position	Horizontal position

② Control data 2

b11	b10 – b0
1	

← Character pattern selection data →

b11	b10 – b6	b5	b4 – b1	b0
0	Horizontal size	Rim	Character background color	Color density

← Control data 2 →

(1) Functions of control data 1

① **Character vertical size setting (bits 8-11)**
Table 16-4 lists the settings and corresponding character attributes.
Fourteen sizes (1X-14X) can be set for each line.

Table 16-4 Vertical Size Setting

Control data 1				Size	Vertical width of 1 character (in interlace mode)	Maximum number of vertical display characters
b11	b10	b9	b8			
0	0	0	0	1X	16H	8
0	0	0	1	2X	32H	4
0	0	1	0	3X	48H	2
0	0	1	1	4X	64H	2
1	1	0	0	13X	208H	1
1	1	0	1	14X	224H	1

② **Character vertical position setting (bits 4-7)**

Bits 4 to 7 of control data 1 set the vertical position from which display starts for each line. This setting is performed for each line. The line display position is represented by the number of dots from the last line. The set value itself becomes the line spacing. Table 16-5 lists the settings and corresponding vertical positions.

Set the display start position of the entire screen at the IDC start position setting register (IDCORG).

Remark One dot refers to that used when the vertical size is 1X. It does not change even when the vertical size is set to a value other than 1X.

Table 16-5 Vertical Position Setting

Control data 1				Vertical spacing
b_7	b_6	b_5	b_4	
0	0	0	0	Begin display 0 dots from preceding line
0	0	0	1	Begin display 1 dot from preceding line
0	0	1	0	Begin display 3 dots from preceding line
1	1	1	0	Begin display 14 dots from preceding line
1	1	1	1	Begin display 15 dots from preceding line

③ **Character horizontal position setting (bits 0-3)**

Bits 0 to 3 of control data 1 set the horizontal position from which the line is to be displayed. The horizontal position is represented by the number of dots shifted relative to the horizontal start position set by the IDC start position setting register (IDCORG). The set value itself becomes the number of dots the position is shifted. Table 16-6 lists the settings and corresponding horizontal positions.

Remark One dot refers to that used when the horizontal size is 1X.

Table 16-6 Horizontal Position Setting

Control data 1				Horizontal spacing
b_3	b_2	b_1	b_0	
0	0	0	0	Begin display 0 dots from the position set by IDCORG
0	0	0	1	Begin display 1 dot from the position set by IDCORG
0	0	1	0	Begin display 2 dots from the position set by IDCORG
1	1	1	0	Begin display 14 dots from the position set by IDCORG
1	1	1	1	Begin display 15 dots from the position set by IDCORG

(2) Functions of control data 2

① Character horizontal size setting (bits 6-10)
Table 16-7 lists the settings and corresponding character attributes.
Twenty-four sizes (1X-24X) can be set for each character.

Table 16-7 Horizontal Size Setting

Control data 2					Size	Horizontal width of 1 character	Maximum number of display characters in 1 line
b_{10}	b_9	b_8	b_7	b_6			
0	0	0	0	0	1X	1.6 μs	24
0	0	0	0	1	2X	3.2 μs	12
0	0	0	1	0	3X	4.8 μs	8
0	0	1	0	0	4X	6.4 μs	6
0	0	1	0	1	5X	8.0 μs	4
0	0	1	1	0	6X	9.6 μs	4
0	0	1	1	1	7X	11.2 μs	3
1	0	1	1	1	23X	36.8 μs	1
1	1	0	0	0	24X	38.4 μs	1

Remark Since the horizontal size consists of 5 bits, up to 32X can be set as data. Although a value exceeding this may be set, because the number of columns per line is 24, the data will not displayed correctly.

② Rim setting (bit 5)
Bit 5 specifies rimming for the character pattern defined in CROM.
When it is set to “0”, rimming is not executed and when it is set to “1”, rimming is executed. The rim color is black only.

③ Character background color setting (bits 1-4)
Table 16-8 lists the settings and corresponding background colors.
The character background color is set by setting the control data of the first character of the character group to which the background color is applied. Bit 4 enables/disables character background color and bits 0 to 3 set the character background color.
The character background color and screen background color can be set simultaneously.

Table 16-8 Character Background Color Setting

Control data 2				Character background color
b_4	b_3	b_2	b_1	
0	×	×	×	No background (black)
1	0	0	0	
1	0	0	1	Blue
1	0	1	0	Green
1	0	1	1	Cyan
1	1	0	0	Red
1	1	0	1	Magenta
1	1	1	0	Yellow
1	1	1	1	White

Remark ×: Don’t care

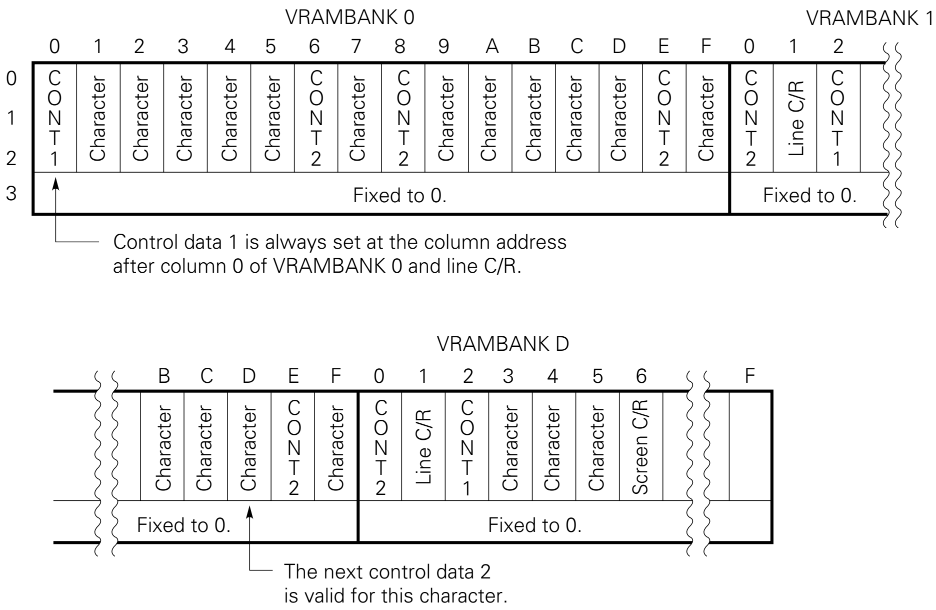
④ Character color density (I output) setting (bit 0)
Bit 0 sets the density of the character color. Up to 8 character colors can be specified. However, 16 colors can be specified by accompanying the specification with the color density.
The density is set by setting “1” in bit 0 of control data 2 of the first character of the character group to which the density is to be set.
Since the I output is also used for $P0B_2$, when using it as the I pin, set the IDCISEL flag to 1.

Remark The I output is output for all the dots regardless of the character dot size. When a space is set between characters, an I signal is output at the space also.

16.5.6 VRAM Data Setting Example

An example of VRAM data setting is shown in Fig. 16-19.

Fig. 16-19 Example of VRAM Data Setting



Remarks CONT1 : Control data 1
CONT2 : Control data 2
Character : Character pattern selection data
Line C/R : C/R that indicates the end of 1 line
Screen C/R : C/R that indicates the end of 1 screen

16.5.7 VRAM Data Setting Cautions

① When setting data at the VRAM, begin from address 00H of VRAMBANK 0 in the state in which "2" is set in the BANK register and the VRAMSEL flag is set.

② VRAM is mapped to addresses 00H to 3FH of RAM bank 2. The area beginning from address 40H is normal RAM. The values at addresses 30H to 3FH are fixed to "0". Therefore, do not set the VRAM data after address 30H.

③ When data is set at the VRAM by using index modification, the index-modified VRAM bank and VRAM row address may not be output for VRAM port only, but also for ports and system registers. The contents of the ports and system registers may be manipulated. (Data memory is not affected.) The hardware that is affected is shown in Table 16-9.

Therefore, when the index modification addressing is used for the addresses shown in Table 16-9, write the data using a direct data memory operation instruction without using index modification (clear the IXE flag).

Table 16-9 Hardware Affected by Index Modification

VRAM address after index modification		Affected hardware		
VRAM bank	VRAM address	Bank	Address	Hardware name
1, 5, 9, D	2FH	2	6FH	Port 2D
	30H to 32H	2	70H to 72H	Ports 2A-2C
	33H	2	73H	VRAM bank specification
	34H to 3FH	—	74H to 7FH	System registers
3, 7, B	34H to 3FH	—	74H to 7FH	System registers

Caution Actually, there is no VRAM at VRAM addresses 30H to 3FH. Do not execute an instruction that operates these addresses.

When an index register increment instruction is executed, in particular, the VRAM addresses may become 30H to 3FH. When setting data over multiple VRAM banks by using an increment instruction, proceed as follows:

(1) Increment the VRAM address. When the address reaches 2FH, stop incrementing and clear the IXE flag.
(2) Switch the VRAM bank and reset the index register.
(3) Set the IXE flag again and start incrementing.

④ When the memory pointer is used to write data to the VRAM, ports and system registers may be operated in the same way as when index modification addressing is used. The hardware that is affected is the same as that shown in Table 16-9. However, since VRAM addresses 30H to 3FH are not VRAM area, do not set them in the memory pointer.
When accessing address 2FH of VRAM banks 1, 5, 9, and D, do not use the memory pointer, but write data with a direct memory operation instruction.

⑤ Always set control data 1 at the beginning of a line and a screen whether or not the data is to be changed.

⑥ Set the character pattern selection data sequentially from the VRAM low address, in the order displayed from the top left of the screen.

⑦ Always set the line carriage return data at the end of a line.

⑧ Always set the VRAM pointer reset carriage return data at the end of the VRAM data when data exceeding the VRAM capacity (extended display mode) is displayed by program.

⑨ Always set the screen carriage return data at the end of the data of a screen.

⑩ When displaying data in the extended display mode, before reading the VRAM pointer reset carriage return data, rewrite the VRAM data at the first line that exceeds the VRAM capacity.

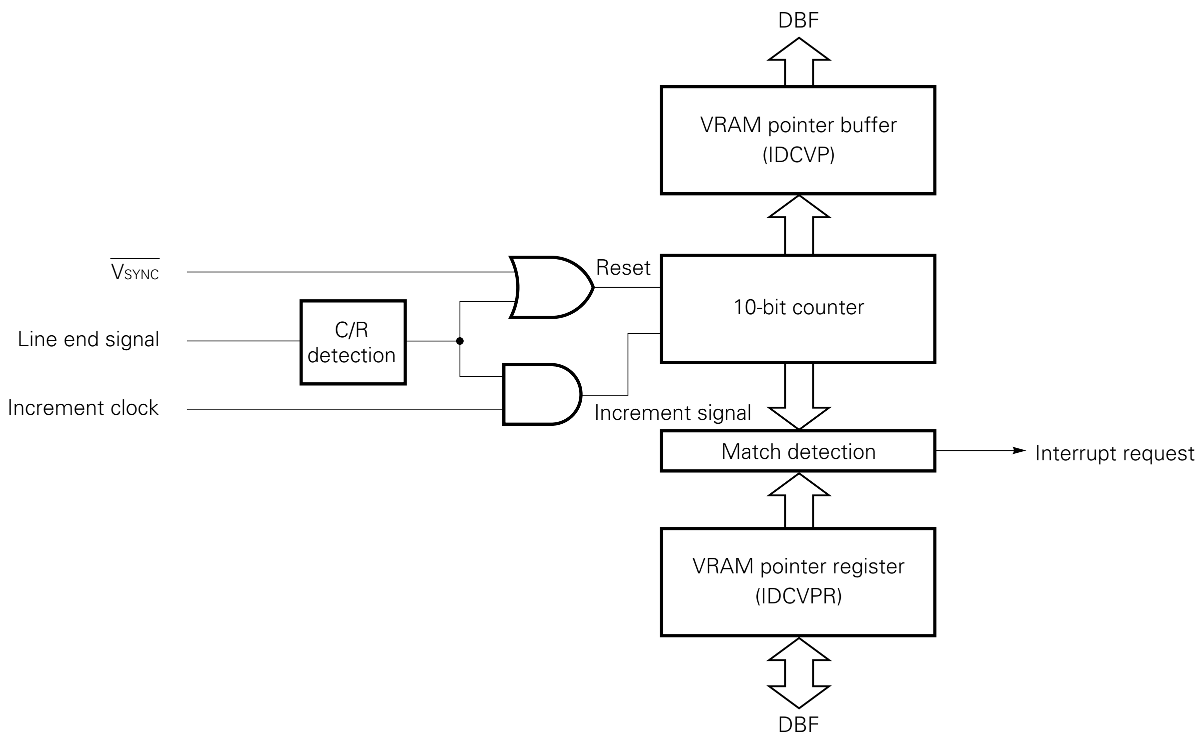
16.6 VRAM POINTER

16.6.1 Configuration of VRAM Pointer

Fig. 16-20 shows the configuration of the VRAM pointer.

The VRAM pointer generates an interrupt request at the specified VRAM address.

Fig. 16-20 Configuration of VRAM Pointer

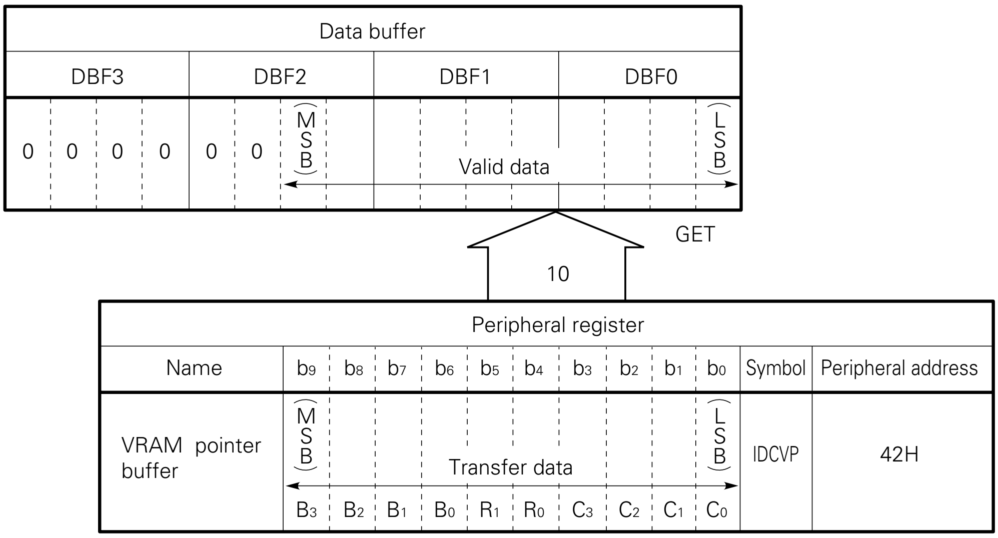


16.6.2 VRAM Pointer Buffer (IDCVP)

Fig. 16-21 shows the configuration of the VRAM pointer buffer.

The VRAM buffer outputs the VRAM pointer value. Since the VRAM addresses at which the data has been already used for display can be identified by reading the VRAM pointer value, the VRAM data before the read address can be rewritten.

Fig. 16-21 Configuration of VRAM Pointer Buffer



Remark B_n : VRAM bank
R_n : VRAM row address
C_n : VRAM column address

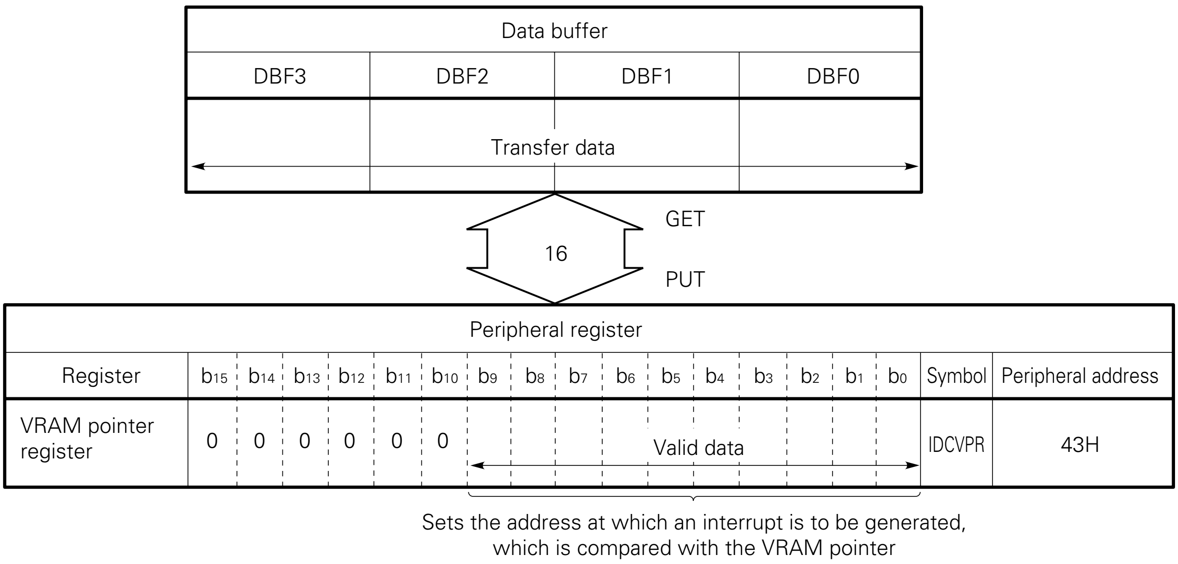
16.6.3 VRAM Pointer Register (IDCVPR)

Fig. 16-22 shows the configuration of the VRAM pointer register.

The VRAM pointer register specifies the VRAM address at which an interrupt is to be generated. When the value set in the VRAM pointer register and the value of the VRAM pointer match, an interrupt request is issued.

Therefore, VRAM data before the VRAM address set in the VRAM pointer register is rewritten in the interrupt routine.

Fig. 16-22 Configuration of VRAM Pointer Register



16.7 IDC OUTPUT PINS (BLANK, RED, GREEN, BLUE, I PINS)

16.7.1 Functions of IDC Output Pins

The IDC output pins (BLANK, RED, GREEN, BLUE, I pins) are CMOS push-pull output pins and output an active high signal.

The signal that blanks the broadcast image (blanking signal) is output from the BLANK pin and the character pattern signal (OR of R, G, B signals) is output from the RED, GREEN, BLUE, and I pins.

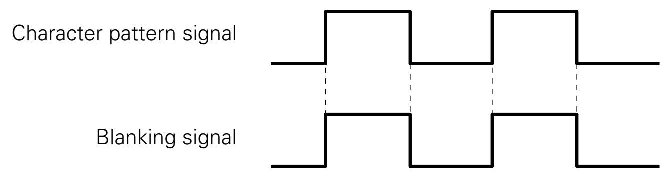
16.7.2 IDC Output Waveforms

Fig. 16-23 shows the IDC output signal waveforms.

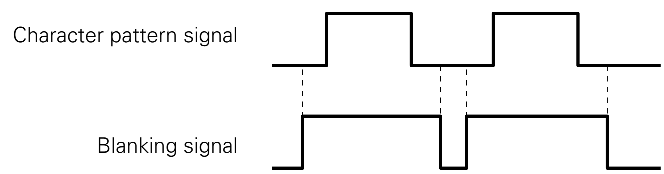
When there is no rim, the blanking signal and character pattern signal output the same signal. When there is a rim, the blanking signal enveloping the character pattern signal is output from the BLANK pin. When the least significant bit of control data 1 is “1”, the character pattern signal output from the I pin outputs high level for the display character only.

Fig. 16-23 IDC Output Waveforms (1 Character)

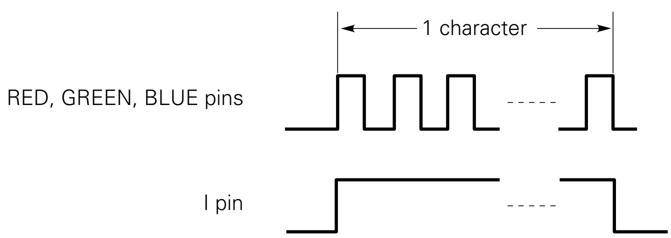
(a) When there is no rim



(b) When there is a rim



(c) I pin output



16.8 SAMPLE PROGRAM

16.8.1 Displaying Data Exceeding VRAM Capacity (Extended Display Mode)

Data exceeding the VRAM capacity can be displayed by applying an interrupt and rewriting the data that has already been displayed by program while VRAM data is being displayed on the screen.

When displaying data in the extended display mode, set the VRAM pointer reset C/R at the end of the VRAM data.

Normal screen display can be performed even when the character group to be displayed exceeds 8 lines as long as the VRAM data does not exceed the VRAM capacity.

(1) Example of normal screen display exceeding 8 lines

	Column 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Column 17
Line 0																		
1													C	H		0	8	
2													0	0	:	0	0	
3																		
4	O	F	F					P	i	n	P					O	N	
5	O	F	F				S	P	E	A	K	E	R			O	N	
6		L							B	A	L					R		
7	L	O	W					B	A	S	S				H	I	G	H
8	L	O	W				T	R	E	B	L	E			H	I	G	H
9	S	O	F	T		P	I	C	T	U	R	E		S	H	A	R	P
Line 10	R	E	D						H	U	E			G	R	E	E	N

(2) Example in extended display mode

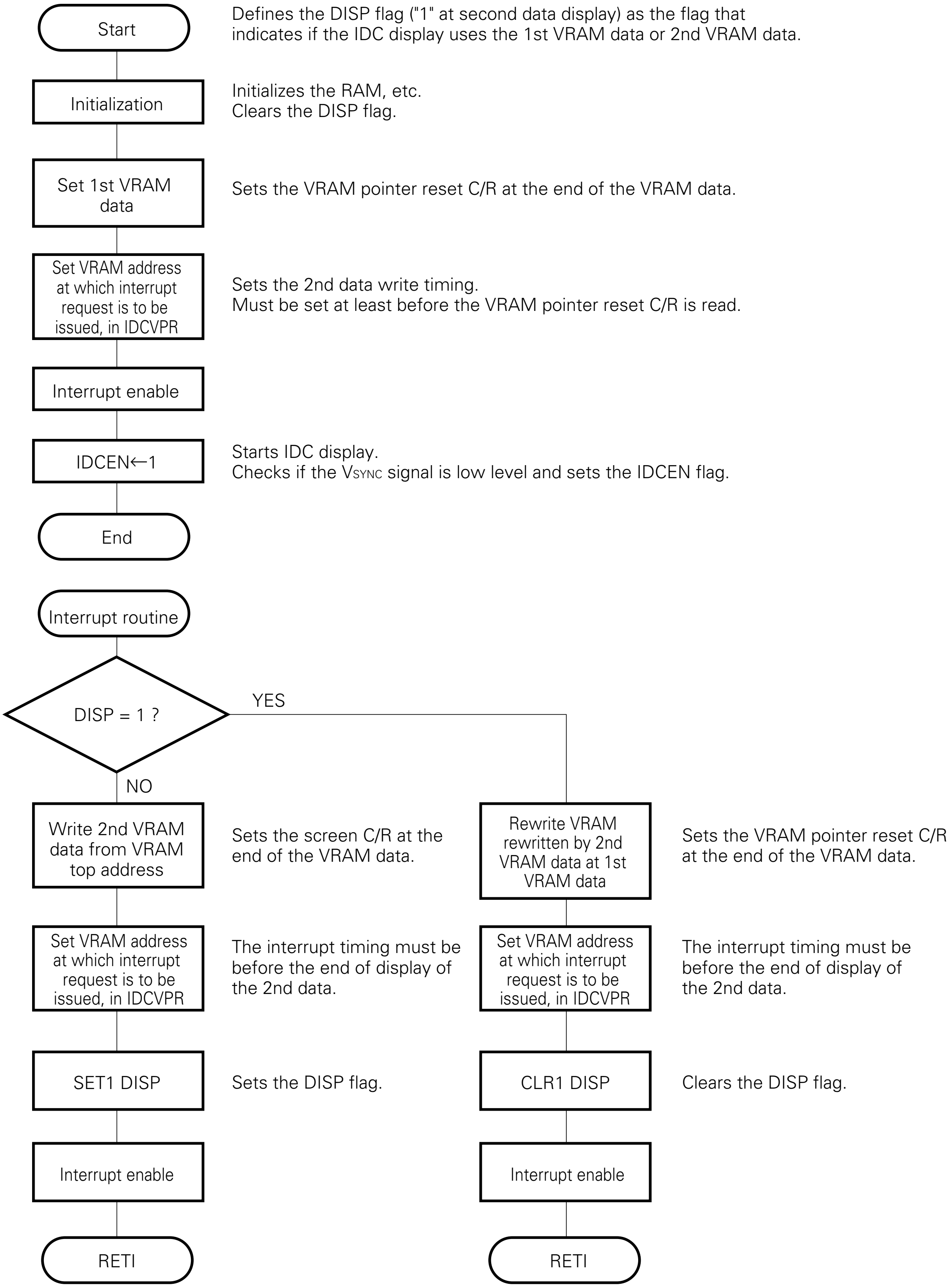
When displaying a screen like the one shown below, use the extended display mode.

	Column 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	Column 22
Line 0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	0
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	0	1
3	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	0	1	2
4	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	0	1	2	3
5	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	0	1	2	3	4
6	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	0	1	2	3	4	5
7	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	0	1	2	3	4	5	6
8	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	0	1	2	3	4	5	6	7
9	9	A	B	C	D	E	F	G	H	I	J	K	L	M	0	1	2	3	4	5	6	7	8
10	A	B	C	D	E	F	G	H	I	J	K	L	M	0	1	2	3	4	5	6	7	8	9
11	B	C	D	E	F	G	H	I	J	K	L	M	0	1	2	3	4	5	6	7	8	9	A
12	C	D	E	F	G	H	I	J	K	L	M	0	1	2	3	4	5	6	7	8	9	A	B
13	D	E	F	G	H	I	J	K	L	M	0	1	2	3	4	5	6	7	8	9	A	B	C
14	E	F	G	H	I	J	K	L	M	0	1	2	3	4	5	6	7	8	9	A	B	C	D
15	F	G	H	I	J	K	L	M	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
Line 16	G	H	I	J	K	L	M	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

The extended display mode flowchart is shown on the next page.

(3) Flowchart

The flowchart when one display data is displayed by rewriting the VRAM data once is shown below.



17. HORIZONTAL SYNCHRONIZING SIGNAL COUNTER

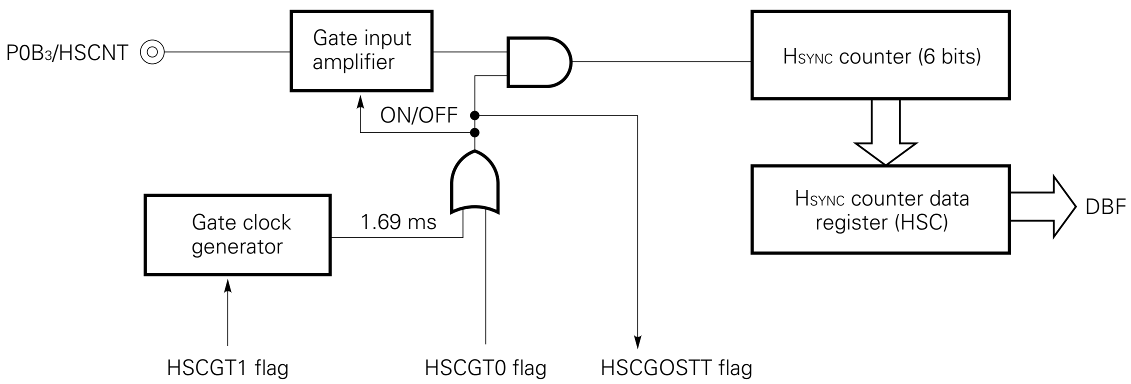
17.1 GENERAL

Fig. 17-1 outlines the horizontal synchronizing signal counter.

The horizontal synchronizing signal counter counts the horizontal synchronizing signal (H_{SYNC} signal) separated from the image signal sent from the broadcast station. It is counted up at the rising edge of the synchronizing signal.

The frequency of the horizontal synchronizing signal can be found, and used to detect which broadcast station is using the frequency currently being received, by dividing the count value by the gate open time (set by H_{SYNC} counter gate control register).

Fig. 17-1 Horizontal Synchronizing Signal Counter



Remarks 1. HSCGOSTT (bit 3 of H_{SYNC} counter gate register: see **Fig. 17-4**): Detects opening and closing of the H_{SYNC} counter gate.

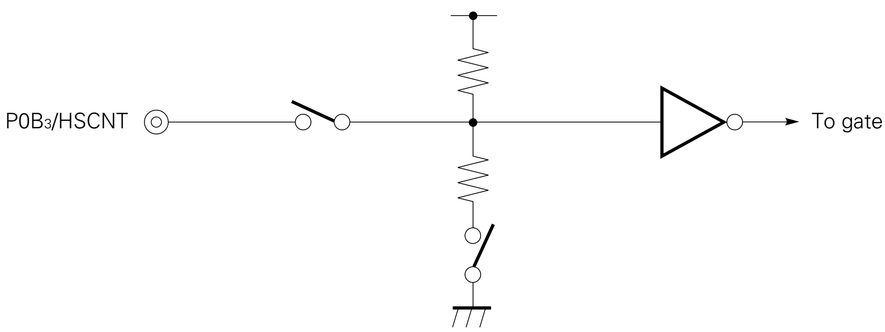
2. HSCGT1 and HSCGT0 (bits 1 and 0 of the H_{SYNC} counter gate control register: see **Fig. 17-3**) : Control opening and closing and the open time of the H_{SYNC} counter gate.

17.2 GATE INPUT AMPLIFIER

Fig. 17-2 shows the configuration of the gate input amplifier.

The gate input amplifier is the self-bias type. To prevent erroneous operation by noise, use it without a coupling capacitor. Input the signal at a full amplitude of low level $0.2V_{DD}$ or less and high level $0.8V_{DD}$ or more.

Fig. 17-2 Gate Input Amplifier



17.3 GATE CONTROL

The HSYNC counter gate is controlled by the HSCGT× flag of the HSYNC counter gate control register and the HSCGOSTT flag of the HSYNC counter gate judge register.

Figs. 17-3 and 17-4 show the organization and functions of the HSYNC counter gate control register and HSYNC counter gate judge register.

The horizontal synchronizing signal counter input pin (HSCNT pin) is also used for $P0B_3$. When using $P0B_3$ as the HSCNT pin, set it to the input mode. If $P0B_3$ is used as the HSCNT pin, when it is read, "0" is always read. When using $P0B_3$ as a port, set the HSYNC counter gate control register to all "0".

Fig. 17-3 Configuration of HSYNC Counter Gate Control Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
HSYNC counter gate control register	0	0	HSCGT1	HSCGT0	11H	R/W

Controls the HSYNC counter gate. (b_1, b_0)

b_1	b_0	Controls the HSYNC counter gate.
0	0	Gate closed mode
0	1	Gate open mode
1	0	1.69 ms gate open mode
1	1	Not to be set.

b_3, b_2: Fixed to 0.

Upon reset		b_3	b_2	b_1	b_0
Upon reset	Power-on	0	0	0	0
	Clock stop	↓	↓	0	0
	CE	↓	↓	0	0

Fig. 17-4 Configuration of HSYNC Counter Gate Judge Register

Register	Flag symbol b3	Flag symbol b2	Flag symbol b1	Flag symbol b0	Address	Read/write
HSYNC counter gate judge register	HSCGOSTT	0	0	0	12H	R

b2, b1, b0: Fixed to 0.

HSCGOSTT: Detects opening and closing of the HSYNC counter gate.

HSCGOSTT	
0	Gate closed
1	Gate open

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	–	↓	↓	↓
CE	–	↓	↓	↓

17.3.1 HSYNC Counter Gate Mode Selection Flag (HSCGT×)

The HSCGT× flag controls the HSYNC counter input gate clock.

The following three modes can be selected:

(a) Gate closed mode

The gate clock generator does not operate and the gate remains closed. Therefore, the HSYNC counter does not operate. Self-biasing of the input pin is also disabled.

When using HSCNT/P0B3 as a port, always select this mode.

(b) Gate open mode

After the gate is opened and the HSYNC counter is reset, counting of the HSYNC signal begins.

The input pin is biased.

(c) 1.69 ms gate open mode

Counting of the HSYNC signal begins after a maximum delay of 8 ms after the gate is opened and the HSYNC counter is reset. The gate clock generator operates and the gate time becomes 1.69 ms.

The input pin is biased.

17.3.2 HSYNC Counter Gate Open Status Flag (HSCGOSTT)

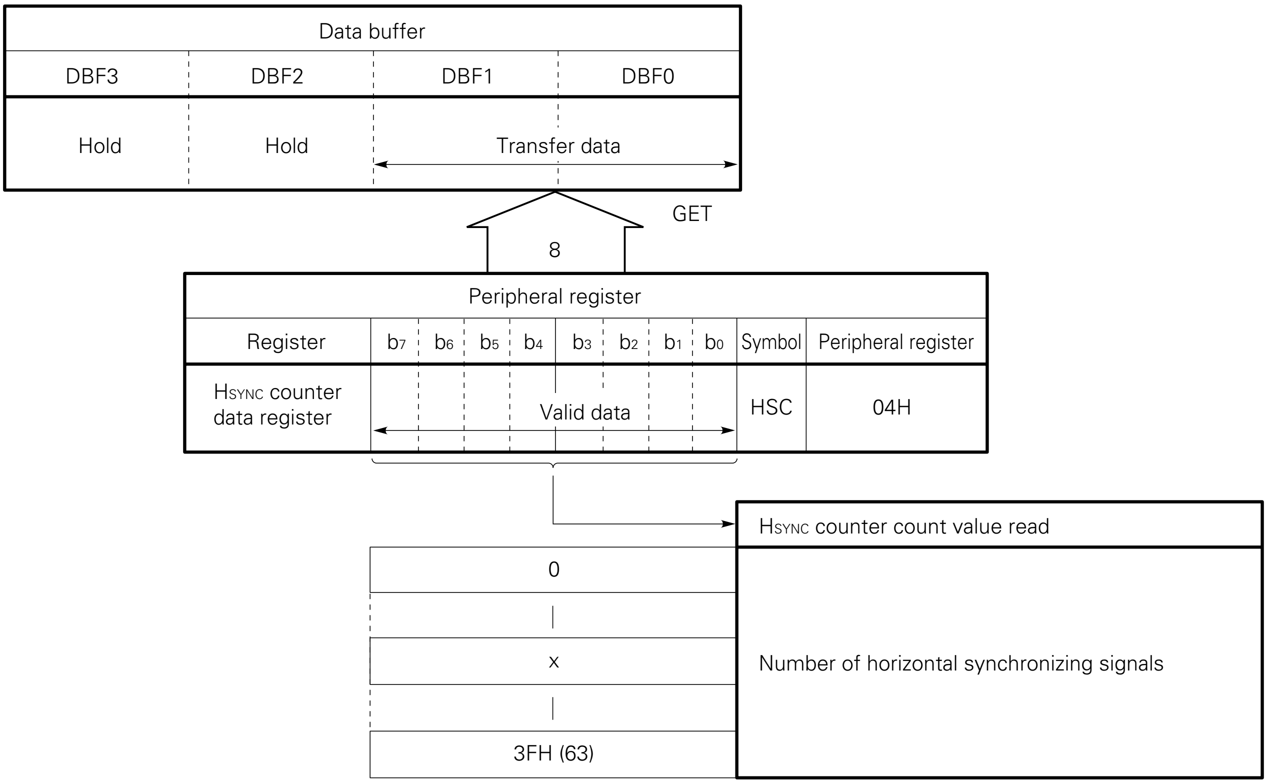
The HSCGOSTT flag detects the status of the HSYNC counter gate. It is normally set (1) while the gate is open.

In the 1.69 ms gate open mode, "1" is read from HSCGOSTT from the time the data was set even if a gate clock does not arrive.

17.4 HSYNC COUNTER DATA REGISTER (HSC)

Fig. 17-5 shows the configuration of the HSYNC counter data register.

Fig. 17-5 Configuration of HSYNC Counter Data Register



17.5 SAMPLE PROGRAM

A sample program for the horizontal synchronizing signal counter is shown below.

Example 1.69 ms gate open mode

```
        INITFLG   HSCGT1, NOT HSCGT0     ; Sets the HSYNC counter to the 1.69 ms gate
                                         ; open mode.
LOOP :
        SLF1      HSCGOSTT               ; Detects the HSCGOSTT flag.
        BR        LOOP                   ; And if the HSCGOSTT flag is "0", reads the count
                                         ; value at the data buffer.
        GET       DBF, HSC               ;
```

17.6 STATE AT RESET

At power-on reset, clock-stop, and CE reset, the gate is set to the gate closed mode and the HSYNC counter is reset.

18. PLL FREQUENCY SYNTHESIZER

The PLL (Phase Locked Loop) frequency synthesizer is used to lock VHF (Very High Frequency) band frequencies to a fixed frequency using a phase error comparison system.

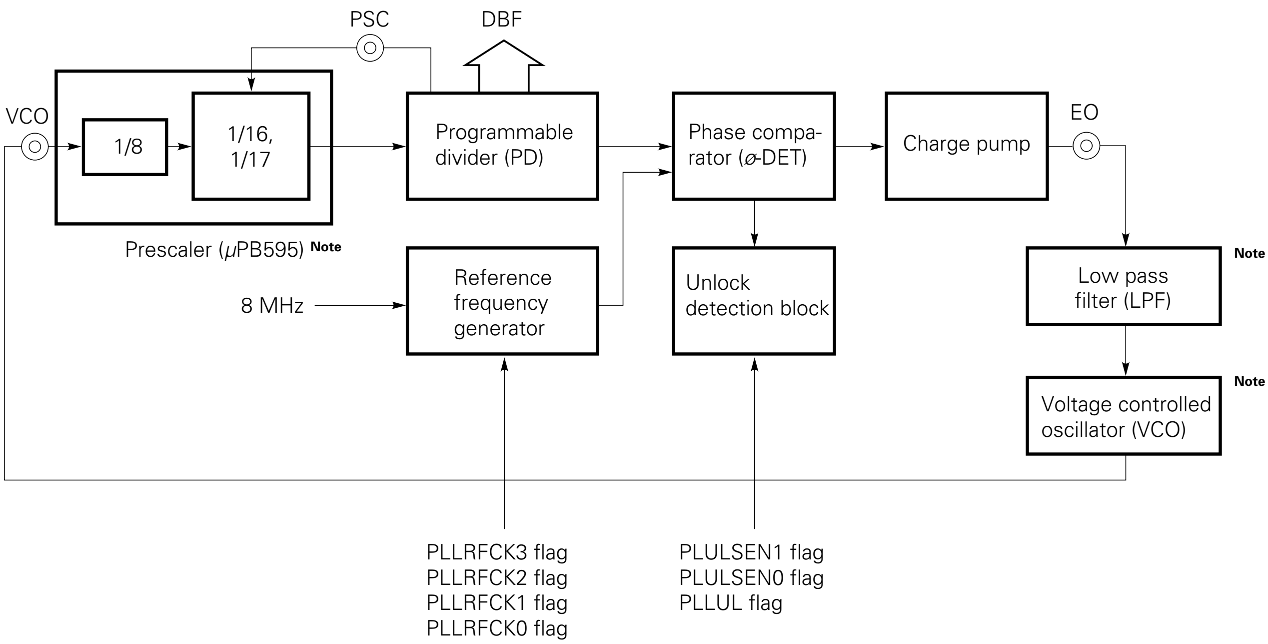
18.1 GENERAL

Fig. 18-1 outlines the PLL frequency synthesizer. A PLL frequency synthesizer can be built by connecting a low pass filter (LPF), voltage controlled oscillator (VCO), and prescaler externally.

The PLL frequency synthesizer divides the signal input from the VCO using a programmable divider and outputs the phase error with the reference frequency to the EO pin.

The PLL frequency synthesizer operates only when the CE pin is high level. When the CE pin is low level, the PLL frequency synthesizer is disabled. For a description of the PLL disabled state, see **Section 18.5.**

Fig. 18-1 PLL Frequency Synthesizer



Remarks 1. PLLRFCK3 to PLLRFCK0 (bits 0-3 of PLL reference clock selection register: see **Fig. 18-5**): Set the PLL frequency synthesizer reference frequency f_r.

2. PLULSEN1 and PLULSEN0 (bits 1 and 0 of PLL unlock flip-flop sensibility selection register: see **Fig. 18-9**): Set the unlock flip-flop set delay time.

3. PLLUL (bit 0 of PLL unlock flip-flop judge register: see **Fig. 18-8**): Detects the state of the unlock flip-flop.

Note External circuit.

18.2 PROGRAMMABLE DIVIDER

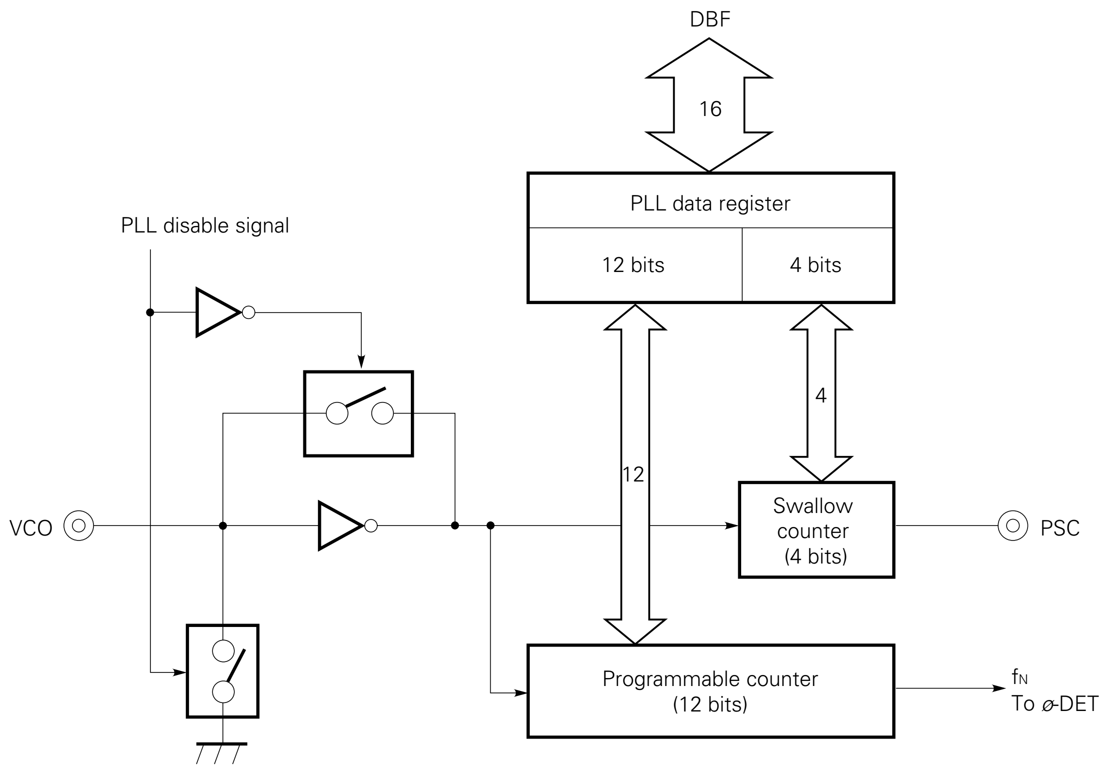
18.2.1 Configuration

Fig. 18-2 shows the configuration of the programmable divider.

The programmable divider divides the signal input from the VCO pin at the division ratio set by program. The division method is the pulse swallow method.

The division value is set by the PLL data register through a data buffer.

Fig. 18-2 Programmable Divider



18.2.2 Programmable Divider and PLL Data Register

The division value is set in the swallow counter and programmable counter by the PLL data register through a data buffer. The swallow counter and programmable counter are 4-bit and 12-bit binary down counters, respectively.

Data is written to the PLL data register with the "PUT PLLR, DBF" instruction, and read from the PLL data register with the "GET DBF, PLLR" instruction.

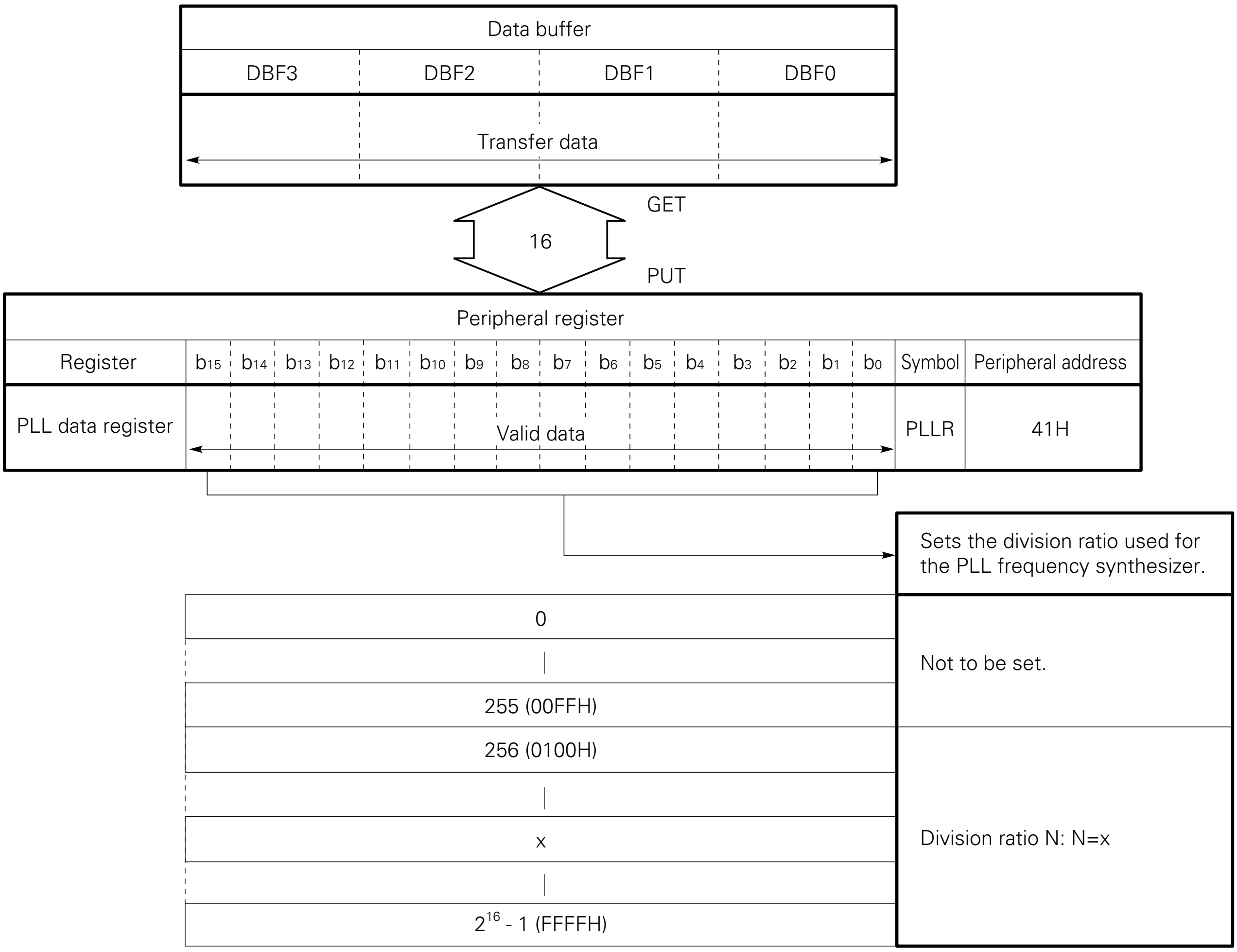
For a description of the division value (N value) setting method, see **Section 18.6.**

(1) PLL data register and data buffer

Fig. 18-3 shows the relationship between the PLL data register and the data buffer.

All 16 bits of the PLL data register are valid. The 12 high-order bits are set in the programmable counter and the 4 low-order bits are set in the swallow counter.

Fig. 18-3 PLL Data Register and Data Buffer



(2) Relationship between programmable divider division value N and divided output frequency

The relationship between the value "N" set in the PLL data register and the frequency "f_N" of the signal divided and output by the programmable divider, is shown below.

For details, see **Section 18.6.**

$$f_N = \frac{f_{IN}}{N} \quad (f_{IN}\text{ : Input frequency})$$

18.3 REFERENCE FREQUENCY GENERATOR

Fig. 18-4 shows the configuration of the reference frequency generator.

The reference frequency generator generates the PLL frequency synthesizer reference frequency "f_r" by dividing the 8-MHz signal of a crystal oscillator.

The reference frequency can be selected from among 5 kHz, 6.25 kHz, 10 kHz, 12.5 kHz, and 25 kHz.

The reference frequency is selected with the PLLRFCK× flags of the PLL reference clock selection register.

Fig. 18-5 shows the organization and functions of the PLL reference clock selection register.

Fig. 18-4 Reference Frequency Generator

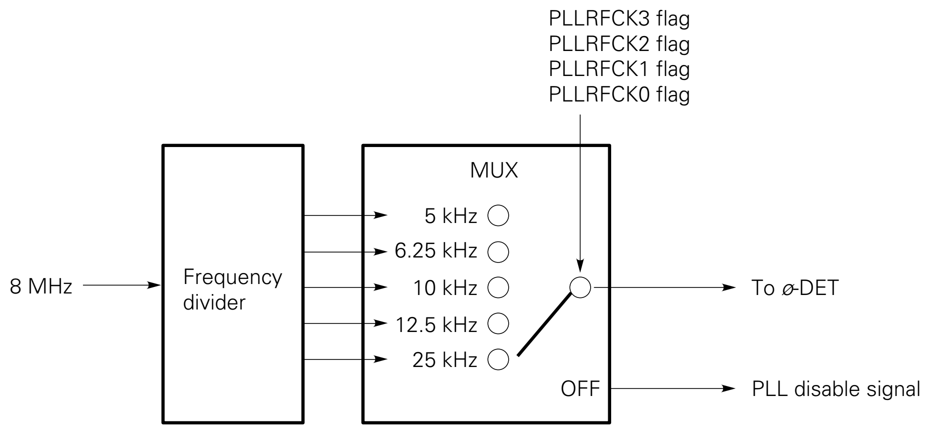


Fig. 18-5 Configuration of PLL Reference Clock Selection Register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
PLL reference clock selection register	PLLRFCK3	PLLRFCK2	PLLRFCK1	PLLRFCK0	13H	R/W

Sets the PLL frequency synthesizer reference frequency f_r.

b3	b2	b1	b0	
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
1	1	1	1	PLL disabled
Others				Not to be set.

Upon reset					
Power-on	1	1	1	1
Clock stop	1	1	1	1
CE	Hold			

Remark If PLL disabled is selected, the VCO pin is pulled down internally. The EO pin is floated.

18.4 PHASE COMPARATOR (ϕ-DET), CHARGE PUMP AND UNLOCK DETECTION BLOCK

18.4.1 Configuration of Phase Comparator, Charge Pump and Unlock Detection Block

Fig. 18-6 shows the configuration of the phase comparator, charge pump and unlock detection block.

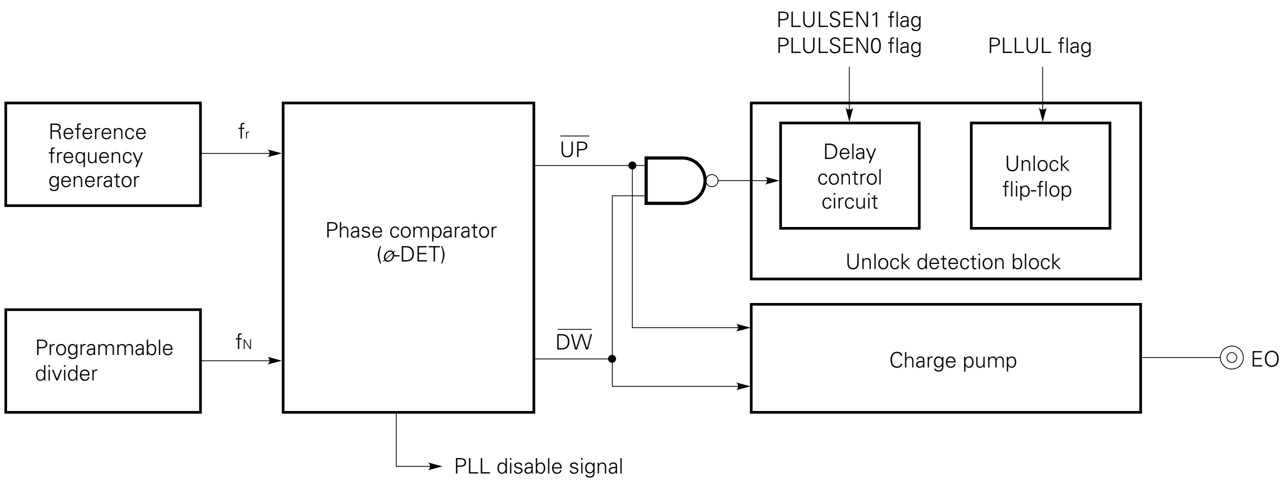
The phase comparator (ϕ-DET) compares the phase of the divided frequency (f_N) signal output from the programmable divider and that of the reference frequency (f_r) signal output from the reference frequency generator and outputs an up request signal ($\overline{UP}$) or down request signal ($\overline{DW}$).

The charge pump outputs the output of the phase comparator from the error out pin (EO pin).

The unlock detection block consists of a delay control circuit and an unlock flip-flop, and detects the PLL frequency synthesizer unlocked state.

Sections 18.4.2 to **18.4.4** describe the operation of the phase comparator, charge pump, and unlock detection block.

Fig. 18-6 Phase Comparator, Charge Pump and Unlock Detection Block



18.4.2 Phase Comparator Functions

As shown in Fig. 18-6, the phase comparator compares the phase of the programmable divider divided (f_N) output and that of the reference frequency (f_r) signal and outputs an up request signal or down request signal.

That is, if divided frequency "f_N" is lower than reference frequency "f_r", an up request signal is output, and if divided frequency "f_N" is higher than reference signal "f_r", a down request signal is output.

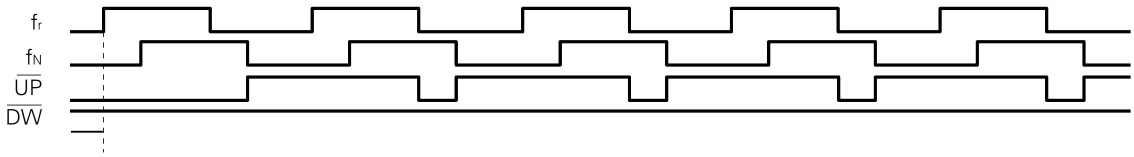
Fig. 18-7 shows the relationship among the reference frequency f_r, division frequency f_N, up request signal, and down request signal.

In the PLL disabled state, neither an up request signal nor a down request signal is output.

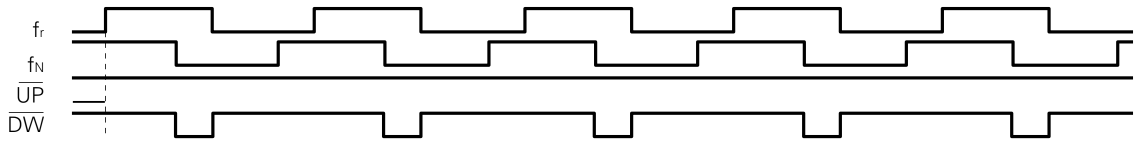
The up request and down request signals are input to the charge pump and unlock detection block.

Fig. 18-7 f_r, f_N, $\overline{UP}$, and $\overline{DW}$ Signal Relationship

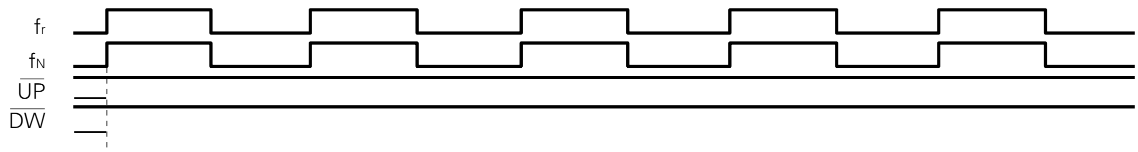
(a) When f_N phase lags f_r phase



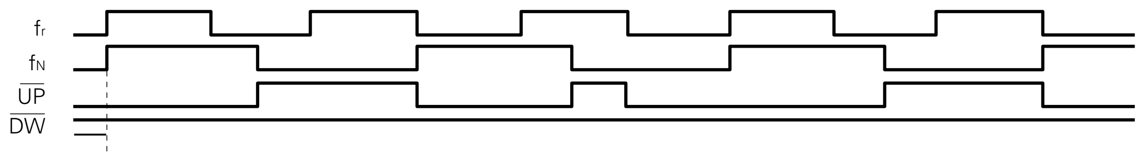
(b) When f_N phase leads f_r phase



(c) When f_N and f_r are same phase



(d) When f_N frequency lower than f_r frequency



18.4.3 Charge Pump

As shown in Fig. 18-6, the charge pump outputs the up request signal or down request signal sent from the phase comparator, from the error out pin (EO pin).

Error output pin output, division frequency f_N, and reference frequency f_r have the following relation:

When reference frequency f_r > division frequency f_N: Low level output
When reference frequency f_r < division frequency f_N: High level output
When reference frequency f_r = division frequency f_N: Floating

18.4.4 Configuration and Functions of Unlock Detection Block

As shown in Fig. 18-6, the unlock detection block detects the PLL frequency synthesizer unlocked state from the phase comparator up request and down request signals.

That is, since the up request signal or down request signal outputs low level while the PLL frequency synthesizer is in the unlocked state, the unlocked state can be detected by monitoring this low level signal.

When the PLL frequency synthesizer is in the unlocked state, the unlock flip-flop is set (1). The state of the unlock flip-flop is detected by the PLLUL flag of PLL unlock flip-flop judge register. The unlock flip-flop is set at the period of the reference frequency f_r selected at the time.

The contents of the PLL unlock flip-flop judge register are read (PEEK instruction) and reset (Read & Reset). The unlock flip-flop must be detected at a period longer than reference frequency f_r period $1/f_r$.

The delay control circuit controls the state that sets the unlock flip-flop by applying a delay to the phase comparator up request signal and down request signal. In other words, if the delay is long, the unlock flip-flop is not set even if the phase deviation between the division frequency (f_N) and reference frequency (f_r) signals is large.

The delay control circuit delay time is set with the PLL unlock flip-flop sensibility selection register.

18.4.5 Organization and Functions of PLL Unlock Flip-Flop Judge Register

Fig. 18-8 shows the organization and functions of the PLL unlock flip-flop judge register.

This register is a read only register, and is reset when the data is read and set to the window register using the "PEEK" instruction.

Since the unlock flip-flop is set at the period of reference frequency f_r, the PLL unlock flip-flop judge register must be read at the window register at a slower period than reference frequency period $1/f_r$.

Fig. 18-8 Configuration of PLL Unlock Flip-Flop Judge Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
PLL unlock flip-flop judge register	0	0	0	PLLUL	22H	R & Reset

PLLUL: Detects the state of the unlock flip-flop.

0	Unlock flip-flop = 0 : PLL locked
1	Unlock flip-flop = 1 : PLL unlocked

b_3–b_1: Fixed to 0.

Upon reset		b_3	b_2	b_1	b_0
	Power-on	0	0	0	*
	Clock stop	↓	↓	↓	Hold
	CE	↓	↓	↓	Hold

* Undefined

18.4.6 Organization and Functions of PLL Unlock Flip-Flop Sensibility Selection Register

Fig. 18-9 shows the organization and functions of the PLL unlock flip-flop sensibility selection register.

When the unlock flip-flop disable state is set by the PLL unlock flip-flop sensibility selection register, the state of the unlock flip-flop is undefined.

Fig. 18-9 Configuration of PLL Unlock Flip-Flop Sensibility Selection Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
PLL unlock flip-flop sensibility selection register	0	0	PLULSEN1	PLULSEN0	32H	R/W

b_1, b_0: Sets the delay time between the reference (f_r) and division frequency (f_N) signals, which is necessary to set the unlock flip-flop.

b_1	b_0	
0	0	1.25-1.5 μs or more
0	1	3.5-37.5 μs or more
1	0	0.25-0.5 μs or more
1	1	Unlock flip-flop disabled

b_3, b_2: Fixed to 0.

Upon reset		b_3	b_2	b_1	b_0
	Power-on	0	0	0	0
	Clock stop	↓	↓	0	0
	CE	↓	↓	Hold	Hold

18.5 PLL DISABLED STATE

The PLL frequency synthesizer is disabled while the CE pin is low level.

The PLL frequency synthesizer is also disabled when PLL disabled is selected by the PLL reference clock selection register.

Table 18-1 shows the state of each block at PLL disabled. Since the PLL reference clock selection register is not initialized (previous state is held) at CE reset, it is reset to its previous state when the CE pin rises to high level after dropping to low level and PLL disabled is set.

Therefore, when PLL disabled must be set at CE reset, the PLL reference clock selection register must be initialized by program.

At power-on reset, PLL disabled is set.

Table 18-1 State of Each Block at PLL Disabled

Block	State	Condition
Reference frequency generator	Output stopped	When PLLRFCK× = 1111B.
	Output not stopped.	CE pin = Low level
Programmable counter	Frequency division stopped	When PLLRFCK× = 1111B (PLL disabled) or CE pin = Low level.
Phase comparator	Output stopped	
Charge pump	Error output pin floated	
VCO pin	Pulled down internally	
PSC pin	Low level output	

18.6 PLL FREQUENCY SYNTHESIZER USE

To control the PLL frequency synthesizer, the following data is necessary:

(1) Reference frequency: f_r
(2) Division value: N

The PLL data setting method is shown below.

(1) Reference frequency f_r setting

The reference frequency is set by the PLL reference clock selection register.

(2) Division value N computation method

Division value N is computed as follows:

$$N = \frac{f_{VCO}}{P \times f_r}$$

f_{VCO} : VCO pin input frequency
f_r : Reference frequency
P : Prescaler division ratio

(3) PLL data setting example

The method of setting the data to receive a VHF band broadcast station is shown below.
A μPB595 is used as the prescaler. Computation is carried out with the fixed division ratio P of 8.

Receiving frequency : 55.25 MHz
Reference frequency : 5 kHz
Intermediate frequency : 45.75 MHz

Division value N is:

$$N = \frac{f_{VCO}}{P \times f_r} = \frac{55250 + 45750}{8 \times 5} = 2525 \text{ (decimal)}$$

$$= 09DDH \text{ (hexadecimal)}$$

Data is written to the PLL data register and PLL reference clock selection register as follows:

PLLR				PLLRF
0 0 0 0	1 0 0 1	1 1 0 1	1 1 0 1	0 0 1 0
0	9	D	D	5 kHz

18.7 SAMPLE PROGRAM

A sample program for controlling the PLL frequency synthesizer is shown below.

Example

```
        MOV     WR,         #00××B    ; Sets the unlock flip-flop set signal delay time.
        POKE    PLLLOCK,    WR        ;
        MOV     WR,         #10××B    ; Sets the reference frequency.
        POKE    PLLRF,      WR
        BANK0
        MOV     DBF3,       #××××B    ; Sets the division value in DBF (bit 0 of DBF0 is the LSB).
        MOV     DBF2,       #××××B    ;
        MOV     DBF1,       #××××B    ;
        MOV     DBF0,       #××××B    ;
        PUT     PLLR,       DBF       ; Sets the division value into the swallow counter and
                                        programmable counter.
UL :                                  ; n steps or more (fr period or more) Note 1
                  :
                  :
                  :
        SKF1    PLLUL
        BR      UL
        MOV     WR,         #0010B    ; Sets the unlock flip-flop set signal delay time. Note 2
        POKE    PLLLOCK,    WR
```

Notes 1. Read the unlock flip-flop at an interval greater than the reference frequency period. If the interval is shorter than this, the unlock flip-flop may not be read correctly, depending on the timing.

2. The first delay time is made maximum and PLL is locked loosely. Next, the delay time is made minimum and the PLL is locked fully. When this is done, viewed overall, the time until the PLL is locked can be shortened and the PLL locking precision can be raised.

18.8 STATE AT RESET

18.8.1 At Power-On Reset

Since the PLL reference clock selection register is initialized to 1111B, the PLL disabled state is set.

18.8.2 At Clock-Stop

The PLL disabled state is set at the time the CE pin drops to low level.

18.8.3 At CE Reset

(1) CE reset caused by clock stop

Since clock-stop initializes the PLL reference clock selection register to 1111B, the PLL disabled state is set.

(2) CE reset when clock not stopped

Since the PLL reference clock selection register retains its previous state, the previous state is set when the CE pin rises to high level.

18.8.4 During the Halt State

If the CE pin is high level, the set state is held.

19. STANDBY

The standby function is used to reduce the supply current during back-up.

19.1 STANDBY FUNCTIONS

Fig. 19-1 outlines the standby block.

The standby block reduces the device current drain by stopping some, or all, operations of the device. The standby block has the following three functions. These functions can be used to suit the application.

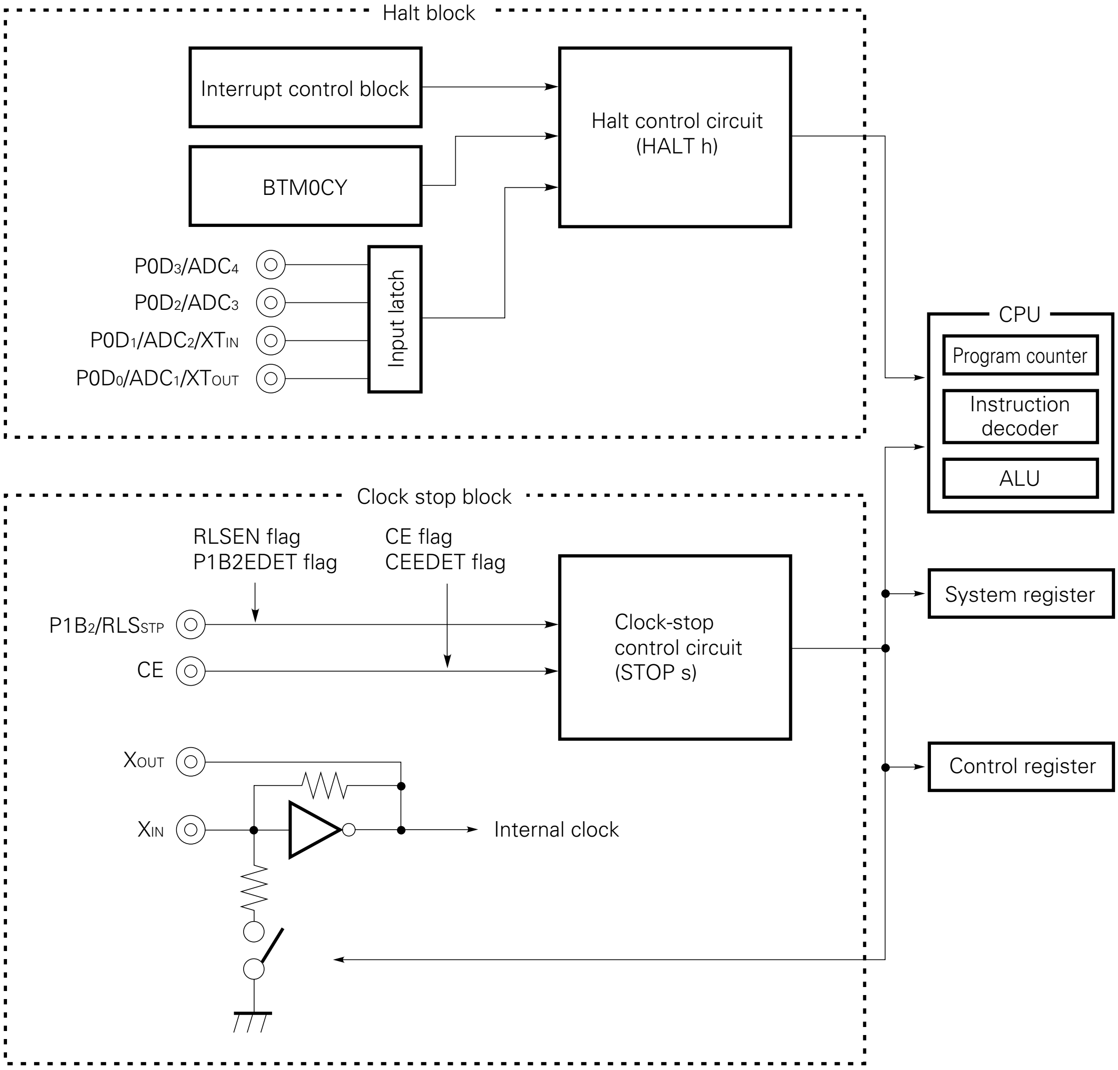
① Halt function
② Clock-stop function
③ Device operation control by CE pin

The halt function reduces the device current drain by stopping CPU operation with a "HALT h" instruction.

The clock-stop function reduces the device current drain by stopping the oscillation circuit with a "STOP s" instruction.

Since the CE pin is used to control operation of the image display controller (IDC) and PLL frequency synthesizer and to reset the device, its operation control function is said to be a standby function.

Fig. 19-1 Standby Block



Remarks 1. RLSEN (bit 0 of clock-stop release enable register: see **Fig. 20-5**): Releases clock-stop.

2. P1B2EDET (bit 0 of $P1B_2$ pin edge detection register: see **Fig. 19-6**): Detects the rising edge input of the $P1B_2$ pin.

3. CE (bit 0 of CE pin level judge register: see **Fig. 19-8**): Detects the status of the CE pin.

4. CEEDET (bit 0 of CE pin edge detection register: see **Fig. 19-9**): Detects the rising edge input of the CE pin.

19.2 HALT FUNCTION

19.2.1 General

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

When a “HALT h” instruction is executed, the program halts and remains stopped until the halt state is released. In the halt state, the device current drain is reduced by the amount of the CPU operating current.

The halt state is released by key input, basic timer 0 and interrupt.

The release conditions are specified with the “h” operand of the HALT h instruction.

The “HALT h” instruction is valid regardless of the CE pin input level.

19.2.2 Halt State

In the halt state, all operations of the CPU are stopped. That is, the “HALT h” instruction stops program execution. However, the peripheral hardware remains in the state set before the “HALT h” is executed.

For an operation description of each hardware device, see **Section 19.4.**

19.2.3 Halt Release Conditions

Fig. 19-2 shows the halt release conditions.

The halt release conditions are set with the 4-bit data specified by the “h” operand of the “HALT h” instruction.

The halt state is released when the condition set to 1 at the “h” operand is satisfied.

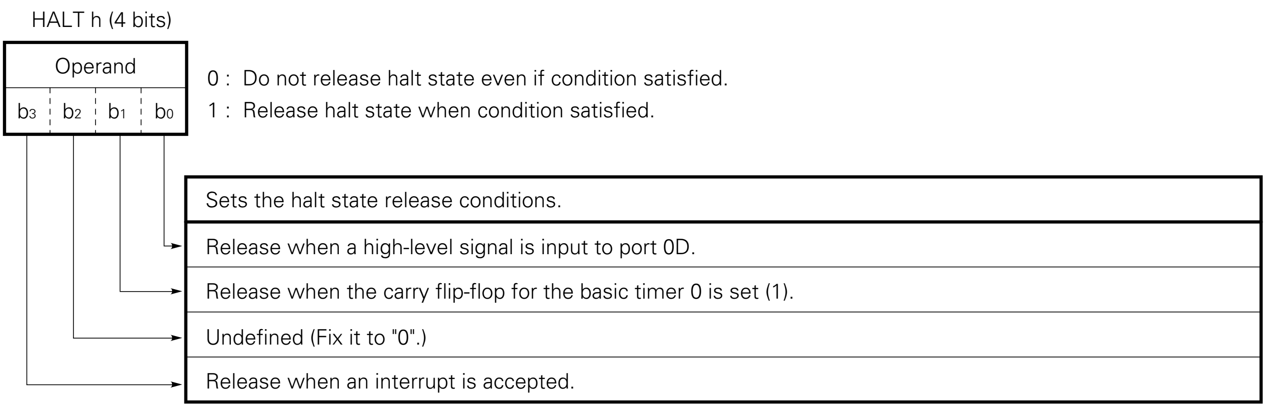
When the halt state is released, the program is executed from the instruction after the “HALT h” instruction.

When multiple release conditions are set, the halt state is released if even one of the set conditions is satisfied.

When reset (power-on reset or CE reset) is applied to the device, the halt state is released and the reset operations are performed.

When 0000B is set at halt release condition “h”, no halt condition is set. If reset (power-on reset or CE reset) is applied to the device at this time, the halt state is released.

Fig. 19-2 Halt Release Conditions



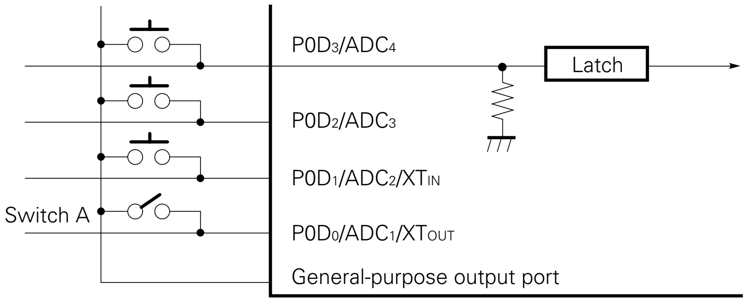
19.2.4 Halt Release by Key Input

Halt release by key input is set by “HALT 0001B” instruction.

When the halt release by key input is set, the halt state is released when high level is input at any one of the 0D port lines ($P0D_3/ADC_4$, $P0D_2/ADC_3$, $P0D_1/ADC_2/XT_{IN}$ and $P0D_0/ADC_1/XT_{OUT}$ pins)

Each 0D port pin has a built-in pull-down resistor.

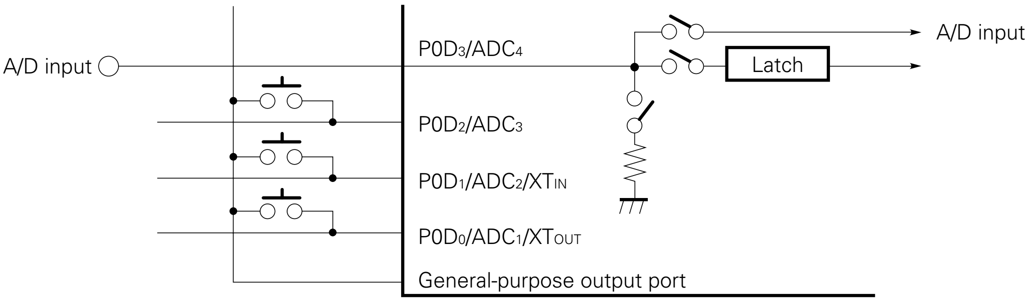
(1) When general-purpose output port is made key source



Execute a “HALT 0001B” instruction after the key source signal general-purpose output port is made high level.

Note that if an alternate switch like switch A in the figure above is used, while switch A is closed, high level is applied to the $P0D_0/ADC_1/XT_{OUT}$ pin and the halt state is immediately released.

(2) When $P0D_0/ADC_1/XT_{OUT}$, $P0D_1/ADC_2/XT_{IN}$, $P0D_2/ADC_3$, or $P0D_3/ADC_4$ pin used as A/D converter input



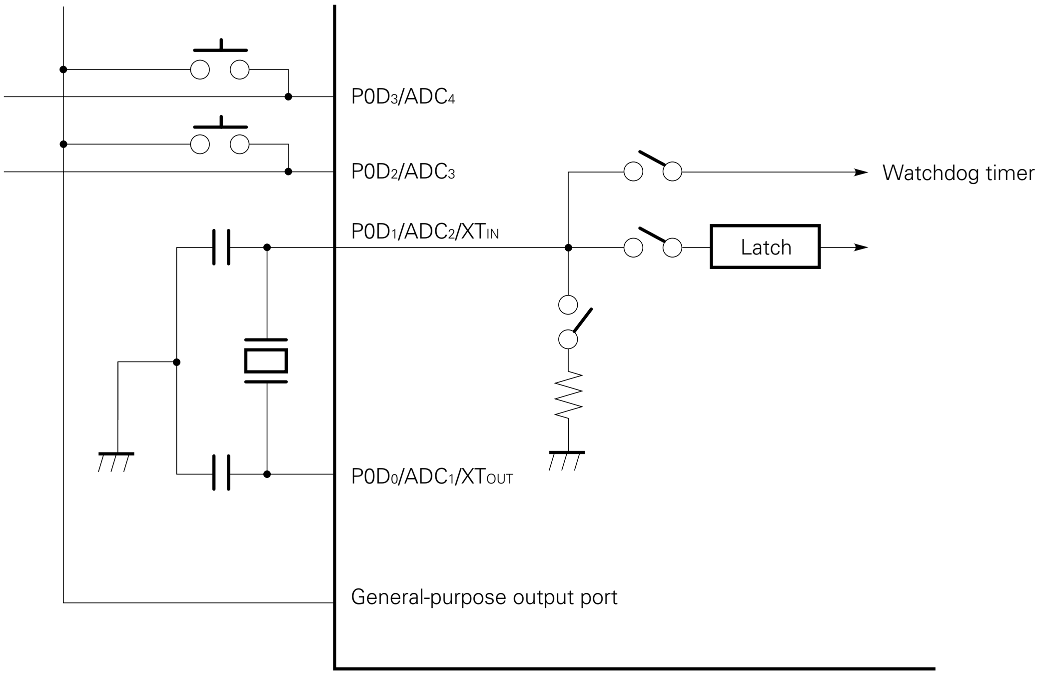
Avoid using the following method as much as possible.

When the $P0D_0/ADC_1/XT_{OUT}$, $P0D_1/ADC_2/XT_{IN}$, $P0D_2/ADC_3$, or $P0D_3/ADC_4$ pin is selected as an A/D converter input, the selected pin (only one pin can be selected at one time) is disconnected from the input latch and connected to the internal A/D converter.

If high level is unexpectedly input to the pin when it is selected as the A/D converter input, the latch circuit is held at high level.

If a “HALT 0001B” instruction is executed in this state, the halt state is immediately released even when an instruction to make the input latch high level is executed because the latch has already been high level.

(3) When used by connecting watchdog timer oscillator to P0D$_0$/ADC$_1$/XT$_{OUT}$ or P0D$_1$/ADC$_2$/XT$_{IN}$ pin



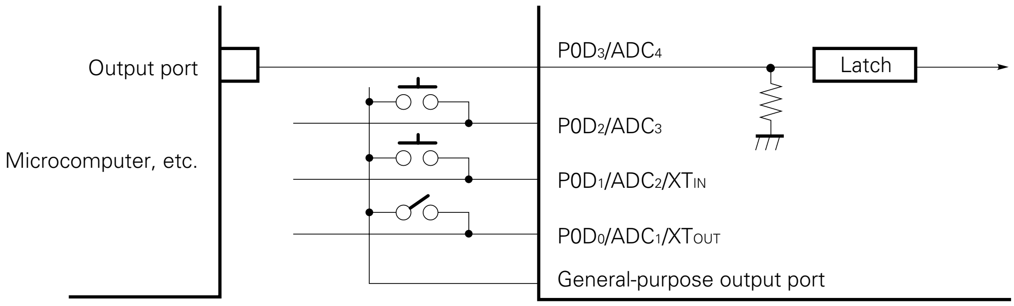
Avoid the using the following method as much as possible.

When the P0D$_0$/ADC$_1$/XT$_{OUT}$ or P0D$_1$/ADC$_2$/XT$_{IN}$ pin is selected as the watchdog timer oscillator connection pin, it is disconnected from the input latch and connected to the internal watchdog timer.

If high level is unexpectedly input to the pin when it is selected as the watchdog timer oscillator connection pin, the latch circuit is held at high level.

When a "HALT 0001B" instruction is executed in this state, the halt state is immediately released even if an instruction to make the input latch high level is executed because the latch has already been high level.

(4) When halt released by other microcomputer, etc.



The P0D$_0$/ADC$_1$/XT$_{OUT}$, P0D$_1$/ADC$_2$/XT$_{IN}$, P0D$_2$/ADC$_3$, and P0D$_3$/ADC$_4$ pins can be used as general-purpose input pins with pull-down resistors.

Halt can be released by another microcomputer, etc. as shown in the figure above.

19.2.5 Halt Release by Basic Timer 0

Halt release by basic timer 0 is set by the “HALT 0010B” instruction.

When halt release by basic timer 0 is set, the halt state is released simultaneously with setting (1) of the carry flip-flop of basic timer 0.

The carry flip-flop of basic timer 0 corresponds to the BTM0CY flag, and is set at a fixed cycle (1 ms, 5 ms or 100 ms). The halt state can be released at a fixed cycle.

Example Program that releases the halt state every 100 ms and executes process A every second.

```
          M1        MEM    0.10H              ; 1 second counter
          HLTTMR    DAT    0010B              ; Symbol definition
                    CLR2   BTMCK1, BTMCK0     ; Built-in macro
                                              ; Sets the cycle of the carry flip-flop of basic timer
                                              ; 0 to 100 ms.
LOOP :
          HALT      HLTTMR                    ; Sets the condition of release caused by the carry
                                              ; flip-flop of basic timer 0 and sets to the halt
                                              ; state.
          SKT1      BTM0CY                    ; Built-in macro
          BR        LOOP                      ; If BTM0CY flag is not set, branches to LOOP.
          ADD       M1, #0100B                ; Adds 0100B to contents of M1.
          SKT1      CY                        ; Built-in macro
          BR        LOOP                      ; If a carry is generated, executes process A.
          | Process A |
          BR        LOOP
```

19.2.6 Halt Release by Interrupt

Halt release by interrupt is set by the “HALT 1000B” instruction.

When halt release by interrupt is set, the halt state is released simultaneously with acceptance of an interrupt. As described in **Chapter 11**, there are 10 interrupt sources. Therefore, which interrupt source releases the halt state must be specified by the program beforehand.

To accept an interrupt, enable all interrupts (EI instruction) and enable each interrupt (interrupt enable flag set) must be satisfied, in addition to issuing an interrupt request from each interrupt source. Even if an interrupt request is issued, if that interrupt is not enabled, the interrupt is not accepted and the halt state is not released.

If the halt state is released by acceptance of an interrupt, the program flow branches to the vector address of the interrupt.

If an RETI instruction is executed after interrupt handling, the program flow returns to the instruction after the HALT instruction.

Example

```
        HLTINT      DAT        1000B          ; Halt condition symbol definition
        INTBTM2     DAT        0006H          ; Interrupt vector address symbol definition
        INT0PIN     DAT        000AH          ; Interrupt vector address symbol definition

        START :                               ; Program address 0000H
                    BR         MAIN
        ORG         INTBTM2                   ; Basic timer 2 interrupt vector address (0006H)
                    BR         INTTIMER
        ORG         INT0PIN                   ; INT0 pin interrupt vector address (000AH)
                    Process A                 ; INT0 pin interrupt handling
                    BR         EI_RETI
        INTTIMER :
                    Process B                 ; Basic timer 2 interrupt handling
        EI_RETI :
                    EI
                    RETI
        MAIN :
                    SET2       IPBTM2, IP0    ; Built-in macro
                    INITFLG    NOT BTM2EXCK, NOT BTM2ZX, BTM2CK1, NOT BTM2CK0
                                              ; Built-in macro
                                              ; Sets basic timer 2 interrupt time interval to 1 ms.
        LOOP :
                    Process C                 ; Main routine processing
                    EI                        ; Enable all interrupts.
                    HALT       HLTINT         ; Sets halt release by interrupt.
                  ; 1
                    BR         LOOP
```

In the example above, when a basic timer 2 interrupt is accepted, the halt state is released and process B is executed. When an INT_0 pin interrupt is accepted, process A is executed. Each time the halt state is released, process C is executed.

When an INT_0 pin interrupt request and a basic timer 2 interrupt request are issued at the same time in the halt state, process A is executed because the INT_0 pin request has higher priority over the basic timer 2 request.

When an “RETI” instruction is executed after process A is executed, the program flow returns to the “BR LOOP” instruction of 1, but the “BR LOOP” instruction is not executed and the basic timer 2 interrupt is accepted.

When a “RETI” instruction is executed after execution of basic timer 2 interrupt handling process B, the “BR LOOP” instruction is executed.

Caution Specify a NOP instruction, immediately before a HALT instruction which is released when an interrupt request flag (IRQ×××) with the corresponding interrupt enable flag (IP×××) set, is set. A NOP instruction specified immediately before a HALT instruction generates one-instruction execution time between the IRQ××× manipulation instruction and HALT instruction. In example 1, clearing IRQ××× by executing the CLR1 IRQ××× instruction affects the HALT instruction correctly. In example 2, however, the CLR1 IRQ××× instruction does not affect the HALT instruction and the system does not enter the HALT mode, because a NOP instruction is not placed immediately before the HALT instruction.

Example 1. Program which correctly executes the HALT instruction

```
                :
                :              ; Sets IRQ×××.
                :

        CLR1    IRQ×××
        NOP                    ; Places a NOP instruction before the HALT instruction.
                               ; (Clearing IRQ××× correctly affects the HALT instruction.)
        HALT    1000B          ; Executes the HALT instruction correctly (enters the HALT
                               ; mode).
                :
                :
```

2. Program which does not correctly execute the HALT instruction

```
                :
                :              ; Sets IRQ×××.
                :

        CLR1    IRQ×××         ; Clearing IRQ××× does not affect the HALT instruction.
                               ; (It affects the instruction after the HALT instruction.)
        HALT    1000B          ; Ignores the HALT instruction (does not enter the HALT
                               ; mode).
                :
                :
```

19.2.7 When Multiple Release Conditions Set Simultaneously

When multiple halt release conditions are set, the halt state is released if even one of the set release conditions is satisfied.

The following example indicates how to judge multiple release conditions when they are satisfied.

Example 1

```
          HLTINT    DAT 1000B
          HLTTMR    DAT 0010B
          HLTKEY    DAT 0001B
          INT0PIN   DAT 000AH          ; INT0 interrupt vector address symbol definition
START :
          BR        MAIN
ORG       INT0PIN
          [Process A]                  ; INT0 interrupt handling
          EI
          RETI
TMRUP :                                ; Timer carry processing
          [Process B]
          RET
KEYDEC :                               ; Key input processing
          [Process C]
          RET
MAIN :
          SET1      P0C0               ; Sets key source output data (high level) at key
                                       ; source pin (P0C0).
          SET2      BTMCK1, BTMCK0     ; Built-in macro
                                       ; Sets the cycle of the carry flip-flop of the basic
                                       ; timer 0 to 1 ms.
          SET1      IP0                ; Built-in macro
                                       ; Enables INT0 pin interrupt.
          EI
LOOP :
          HALT      HLTINT OR HLTTMR OR HLTKEY
                                       ; Sets halt release conditions to interrupt, basic
                                       ; timer 0 carry, and key input.
          SKF1      BTM0CY             ; Built-in macro
                                       ; Detects BTM0CY flag.
          CALL      TMRUP              ; If set (1), executes basic timer 0 carry processing.
          CALL      KEYDEC             ; If latched, executes key input processing. (How-
                                       ; ever, if the interrupt handling and timer process-
                                       ; ing periods are long, key scanning must be
                                       ; repeated.)
          BR        LOOP
```

19.3 CLOCK-STOP FUNCTION

The clock-stop function stops the 8 MHz crystal oscillation circuit (clock stopped state) by executing a "STOP s" instruction.

The supply current is reduced by up to 15 μA.

Specify "0000B" at operand "s" of the "STOP s" instruction.

The "STOP s" instruction is valid only when the CE pin is low level. If a "STOP s" instruction is executed while the CE pin is high level, it is executed as a "NOP" instruction. Always execute a "STOP s" instruction when the CE pin is low level.

The clock-stop state is released by raising the CE pin from low level to high level (CE reset).

19.3.1 Clock-Stop State

Since the crystal oscillation circuit is stopped in the clock-stop state, operation of the CPU, peripheral hardware, and other devices is stopped.

For a description of operation of the CPU and each item of peripheral hardware, see **Section 19.4.**

In the clock-stop state, the power failure detection circuit does not operate even if the power supply voltage V_{DD} drops to 2.2 V. Data memory can be backed up with a low voltage. For a description of the power failure detection circuit, see **Section 20.**

19.3.2 Clock-Stop State Release

The clock-stop state can be released with the three methods described below. For all three methods, after the clock-stop state is released, the program starts from address 0000H.

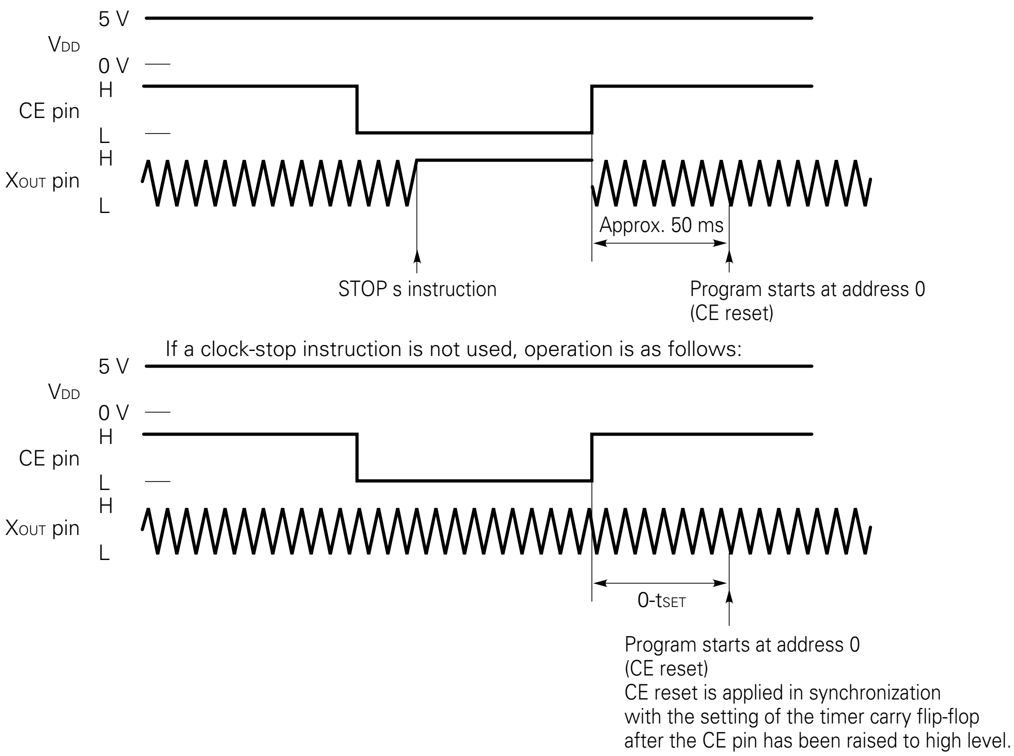
① Raising the CE pin from low level to high level (CE reset)
② Raising the $P1B_2/RLS_{STP}$ pin from low level to high level
③ Dropping V_{DD} to 2.2 V or less, then raising it to 4.5 V (power-on reset)

To use the $P1B_2/RLS_{STP}$ pin to release the clock-stop state, the RLSEN flag of the control register must be set.

19.3.3 Clock-Stop Release by CE Reset

Fig. 19-3 shows the clock-stop release by CE reset.

Fig. 19-3 Clock-Stop Release by CE Reset

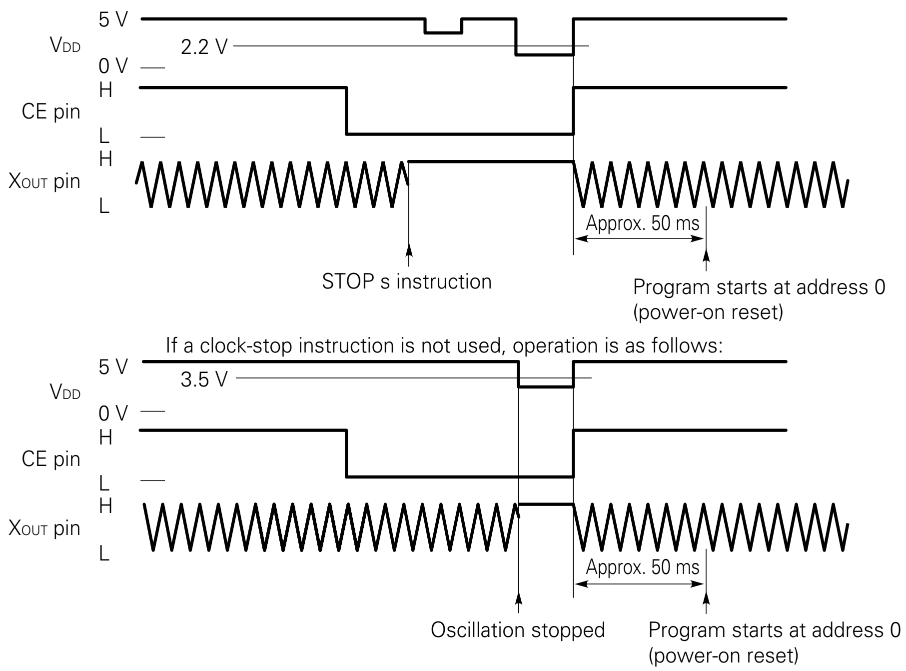


19.3.4 Clock-Stop Release by Power-On Reset

Fig. 19-4 shows the clock-stop release by power-on reset.

If the clock-stop state is released by power-on reset, the power failure detection circuit operates.

Fig. 19-4 Clock-Stop Release by Power-On Reset



19.3.5 Clock-Stop Release by R1B$_2$/RLS$_{STP}$ Pin

When the stop-clock state is released by the P1B$_2$/RLS$_{STP}$ pin, the RLSEN flag of the clock-stop release enable register must be set. The rising edge of the P1B$_2$/RLS$_{STP}$ pin input can be detected by monitoring the P1B2EDET flag of the P1B$_2$ pin edge detection register.

Fig. 19-5 shows the organization and functions of the clock-stop release enable register.

Fig. 19-6 shows the organization and functions of the P1B$_2$ pin edge detection register.

Fig. 19-5 Configuration of Clock-Stop Release Enable Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
Clock-stop release enable register	0	0	0	RLSEN	20H	R/W

RLSEN	Sets clock-stop release by P1B$_2$/RLS$_{STP}$ pin.
0	Do not release.
1	Release at rising edge of signal.

b_3–b_1: Fixed to 0.

Upon reset	b_3	b_2	b_1	b_0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	Hold
CE	↓	↓	↓	Hold

Fig. 19-6 Configuration of P1B2 Pin Edge Detection Register

Register	Flag symbol b3	Flag symbol b2	Flag symbol b1	Flag symbol b0	Address	Read/write
P1B2 pin edge detection register	0	0	0	P1B2EDET	34H	R & Reset

b0 (P1B2EDET): Detects the rising edge input to the P1B2/RLSSTP pin.

0	Rising edge not input.
1	Rising edge input.

b3 to b1: Fixed to 0.

Upon reset		b3	b2	b1	b0
	Power-on	0	0	0	0
	Clock stop	↓	↓	↓	—
	CE	↓	↓	↓	—

19.3.6 Cautions When Using Clock-Stop Instruction

The clock-stop instruction (STOP s instruction) is valid only when the CE pin is low level.
The program must take into account processing when the CE pin is raised unexpectedly to high level.
The description is based on the following example.

Example

```
        XTAL      DAT       0000B        ; Clock-stop condition symbol definition
CEJDG :
   ; ①
        SKF1      CE                     ; Built-in Macro
                                         ; Detects the CE pin input level.
        BR        MAIN                   ; If CE = high level, branches to main processing.
        [Process A]                      ; CE = low level processing
   ; ②
        STOP      XTAL                   ; Clock-stop
   ; ③
        BR        $ – 1
MAIN :
        [Main processing]
        BR        CEJDG
```

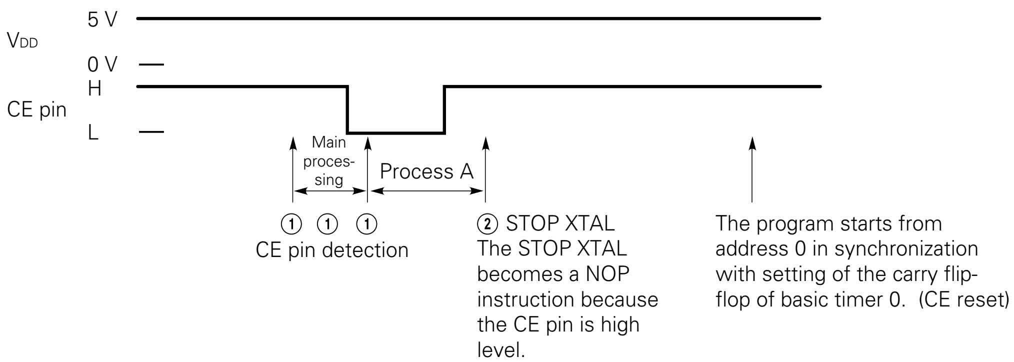
In the example above, the state of the CE pin is detected at ①. If the CE pin is low level, after process A is executed, the clock-stop instruction “STOP XTAL” of ② is executed.

However, if the CE pin becomes high level while the “STOP XTAL” instruction of ② is being executed as shown in the figure below, the “STOP XTAL” instruction operates as a no operation instruction (NOP).

If the branch instruction “BR $-1” of ③ does not exit, the program returns to main processing and erroneous operational occurs.

Therefore, a branch instruction like ③ must be inserted in the program, or the program must be written so that erroneous operational does not occur even if it returns to main processing.

When a branch instruction like ③ is used, CE reset is applied in synchronization with the next setting of the carry flip-flop of basic timer 0, even if the CE pin remains at high level.



19.4 DEVICE OPERATION AT HALT AND CLOCK-STOP

Table 19-1 shows the operation of the CPU and peripheral hardware in the halt state and clock-stop state.

In the halt state, all the peripheral hardware units continue to operate normally except that they stop executing instructions.

In the clock-stop state, all the peripheral hardware units stop operating.

In the halt state, the control register that controls the operating state of the peripheral hardware operates normally (not initialized). However, when a clock-stop instruction is executed, it is initialized to the specified value.

In short, in the halt state, the operation set in the control register continues and in the clock-stop state, the operating state is determined in accordance with the initialized control register value.

For the control register value in the clock-stop state, see **Section 8.**

A sample program is shown below.

Example Program that specifies the P0A$_0$/SDA and P0A$_1$/SCL pins as input ports and uses the P0A$_2$/$\overline{SCK_0}$ and P0A$_3$/SO$_0$ pins as a serial interface.

```
        HLTINT      DAT         1000B
        XTAL        DAT         0000B
        INITFLG     P0ABIO3, P0ABIO2, P0ABIO1, P0ABIO0
    ; ①
        SET2        P0A1, P0A0
        INITFLG     SIO0CH, NOT SB, SIO0MS, SIO0TX
        SET2        SIO0CK1, SIOCK0
    ; ②
        INITFLG     NOT SIO0IMD1, SIO0IMD0
        CLR1        IRQSIO0
        SET1        IPSIO0
        EI
    ; ③
        SET1        SIO0NWT
    ; ④
        HALT        HLTINT
    ; ⑤
        STOP        XTAL
```

In the example, ① outputs high level from the P0A$_1$ and P0A$_0$ pins, ② sets the serial interface 0 conditions, and ③ starts serial communication.

When the HALT instruction is executed at ④ , serial communication continues, and the halt state is released when a serial interface 0 interrupt is received.

If the STOP instruction of ⑤ is executed instead of the HALT instruction of ④, all the flags of the control register set at ①, ② and ③ are initialized. Serial communication is terminated and all the port 0A pins are made general-purpose input ports.

Table 19-1 Device Operation in Halt State and Clock-Stop State

Peripheral hardware	State			
	CE pin: High level		CE pin: Low level	
	At halt	At clock-stop	At halt	At clock-stop
Program counter	Stopped at HALT instruction address.	STOP instruction invalid (NOP)	Stopped at HALT instruction address.	Initialized to 0000H and stopped.
System register	Held		Held	Initialized Note
Peripheral register	Held		Held	Held
Control register	Held		Held	Initialized Note
Timers other than watchdog timer	Normal operation		Normal operation	Operation stopped
Watchdog timer	Normal operation	Normal operation	Normal operation	Normal operation
PLL frequency synthesizer	Normal operation	STOP instruction invalid (NOP)	Disabled	Operation stopped
A/D converter	Normal operation		Normal operation	Operation stopped
D/A converter	Normal operation		Normal operation	Operation stopped
Serial interface	Normal operation		Normal operation	Operation stopped
IDC	Normal operation		Operation stopped	Operation stopped
Horizontal synchronizing signal counter	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	Held

Note For the value that is initialized, see **Sections 5** and **8.**

19.5 PIN PROCESSING CAUTIONS IN HALT STATE AND CLOCK-STOP STATE

The halt state is used to reduce the supply current when only the clock is operating.
The clock-stop function is used to reduce the supply current for holding only the data memory.
Consequently, the supply current must be reduced as much as possible in the halt and clock-stop states.
The supply current depends on the state of each pin and the cautions shown in Table 19-2 must be observed.

Table 19-2 State of Each Pin and Cautions in Halt and Clock-Stop States (1/2)

Pin function		Pin symbol	State of each pin and processing cautions: Halt state	State of each pin and processing cautions: Clock-stop state
General-purpose I/O port	Port 0A	P0A$_3$/SO$_0$ P0A$_2$/$\overline{\text{SCK}_0}$ P0A$_1$/SCL P0A$_0$/SDA	The state before halt is held. **(1) When specified as output pins** If externally pulled down while high level is being output or if externally pulled up while low level is being output, the supply current increases. **(2) When specified as input pins** When floating, noise, etc. increase the drain current. **(3) Port 0D** Since a pull-down resistor is built in, when externally pulled up, the drain current increases.	All pins are specified as general-purpose input pins. Port 0D is internally pulled down.
	Port 0B	P0B$_3$/HSCNT P0B$_2$/I P0B$_1$ P0B$_0$/SI$_0$		
	Port 1B	P1B$_3$/TMIN P1B$_2$/RLS$_{\text{STP}}$ P1B$_1$/CKOUT P1B$_0$		
	Port 1C	P1C$_3$ P1C$_2$/ADC$_7$ P1C$_1$/ADC$_6$ P1C$_0$/ADC$_5$		
	Port 2D	P2D$_2$/SI$_1$ P2D$_1$/SO$_1$ P2D$_0$/$\overline{\text{SCK}_1}$		
General-purpose input port	Port 0D	P0D$_3$/ADC$_4$ P0D$_2$/ADC$_3$ P0D$_1$/ADC$_2$/XT$_{\text{IN}}$ P0D$_0$/ADC$_1$/XT$_{\text{OUT}}$		
General-purpose output port	Port 0C	P0C$_3$ P0C$_2$ P0C$_1$ P0C$_0$		Output ports. The output contents are held. If externally pulled down while high level is being output or if externally pulled up while low level is being output, the supply current increases.
	Port 1A	P1A$_3$ P1A$_2$ P1A$_1$ P1A$_0$		
	Port 1D	P1D$_3$ P1D$_2$ P1D$_1$ P1D$_0$		
	Port 2A	P2A$_0$/PWM$_8$		
	Port 2B	P2B$_3$/PWM$_7$ P2B$_2$/PWM$_6$ P2B$_1$/PWM$_5$ P2B$_0$/PWM$_4$		
	Port 2C	P2C$_3$/PWM$_3$ P2C$_2$/PWM$_2$ P2C$_1$/PWM$_1$ P2C$_0$/PWM$_0$		

Table 19-2 State of Each Pin and Cautions In Halt and Clock-Stop States (2/2)

Pin function	Pin symbol	State of each pin and processing cautions	
		Halt state	Clock-stop state
External interrupt	INT_{NC} INT_0	When floating, noise, etc. increase the supply current.	
PLL frequency synthesizer	VCO EO PSC	At PLL operation, the supply current increases. The state when PLL is disabled is shown below. VCO : Pulled down internally EO : Floating PSC : Low level output When the CE pin becomes low level, the PLL is automatically disabled.	PLL disabled state. VCO : Pulled down internally EO : Floating PSC : Low level output
Image display controller (IDC)	RED GREEN BLUE BLANK $P0B_2$/I	The state before halt is held.	RED, GREEN, BLUE, BLANK: Low level output $P0B_2$/I Specified general-purpose input port.
Crystal oscillation circuit	X_{IN} X_{OUT}	The supply current changes with the oscillation waveform of the crystal oscillation circuit. The larger the oscillation amplitude, the lower the supply current. Since the oscillation amplitude is governed by the crystal and load capacitor used, evaluation is necessary.	The X_{IN} pin is pulled down internally and the X_{OUT} pin outputs high level.

19.6 DEVICE OPERATION CONTROL BY CE PIN

The CE pin controls the following functions by means of the input level and rising edge of a signal received from the outside:

(1) Image display controller (IDC)
(2) PLL frequency synthesizer
(3) Clock-stop instruction disable/enable
(4) Device reset

19.6.1 Image Display Controller (IDC) Operation Control

The IDC can operate only when the CE pin is high level.
When the CE pin is low level, the oscillation circuit stops automatically.

19.6.2 PLL Frequency Synthesizer Operation Control

The PLL frequency synthesizer can operate only when the CE pin is high level.
When the CE pin is low level, the VCO pin is pulled down inside the device and the EO pin is floated. For details, see **Section 18.5.**
The PLL frequency synthesizer can be disabled by program even when the CE pin is high level.

19.6.3 Clock-Stop Instruction Disable/Enable Control

The clock-stop instruction (“STOP s” instruction) is valid only when the CE pin is low level.
If the CE pin is high level, the clock-stop instruction is executed as a no operation instruction (NOP).

19.6.4 Device Reset

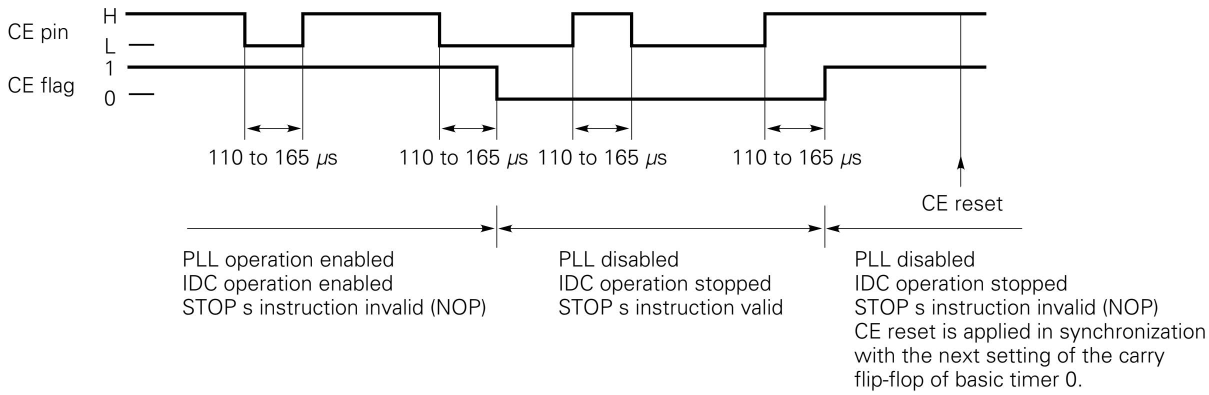
Reset (CE reset) can be applied to the device by raising the CE pin from low level to high level.
Besides CE reset, there is also power-on reset, which is activated when V_{DD} is turned on.
For details, see **Section 20.**

19.6.5 Signal Input to CE Pin

To prevent erroneous operation by noise, the CE pin does not accept signals with a low or high level width of less than 110 to 165 μs. The level of the signal input to the CE pin can be detected with the CE flag of the CE pin level judge register (RF address 07H).

Fig. 19-7 shows the relationship between input signal and CE flag.

Fig. 19-7 Relationship of Signal Input to CE Pin and CE Flag



19.6.6 Organization and Functions of CE Pin Level Judge Register

The CE pin level judge register monitors the CE pin input signal level.

Its organization and functions are shown below.

Fig. 19-8 Configuration of CE Pin Level Judge Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
CE pin level judge register	0	0	0	CE	07H	R

CE: Monitors the level input at the CE pin.

0	Low level input
1	High level input

b_3–b_1: Fixed to 0.

Upon reset	b_3	b_2	b_1	b_0
Power-on	0	0	0	–
Clock stop	↓	↓	↓	–
CE	↓	↓	↓	–

The CE flag also does not change when the CE pin receives signals having a low or high level width of less than 110 to 165 μs.

19.6.7 Organization and Functions of CE Pin Edge Detection Register

The CE pin edge detection register detects the rising edge of the signal applied to the CE pin.
Its organization and functions are shown below.

Fig. 19-9 Configuration of CE Pin Edge Detection Register

Register	Flag symbol b3	b2	b1	b0	Address	Read/write
CE pin detection register	0	0	0	CEEDET	02H	R & Reset

CEEDET: Detects the rising edge input at the CE pin.

0	Rising edge not input
1	Rising edge input

b3–b1: Fixed to 0.

Upon reset	b3	b2	b1	b0
Power-on	0	0	0	0
Clock stop	↓	↓	↓	–
CE	↓	↓	↓	–

Remark The CEEDET flag does not change when the CE pin receives signals having a low or high level width of less than 110 to 165 μs.

20. RESET

The reset function is used to initialize device operation.

20.1 RESET BLOCK CONFIGURATION

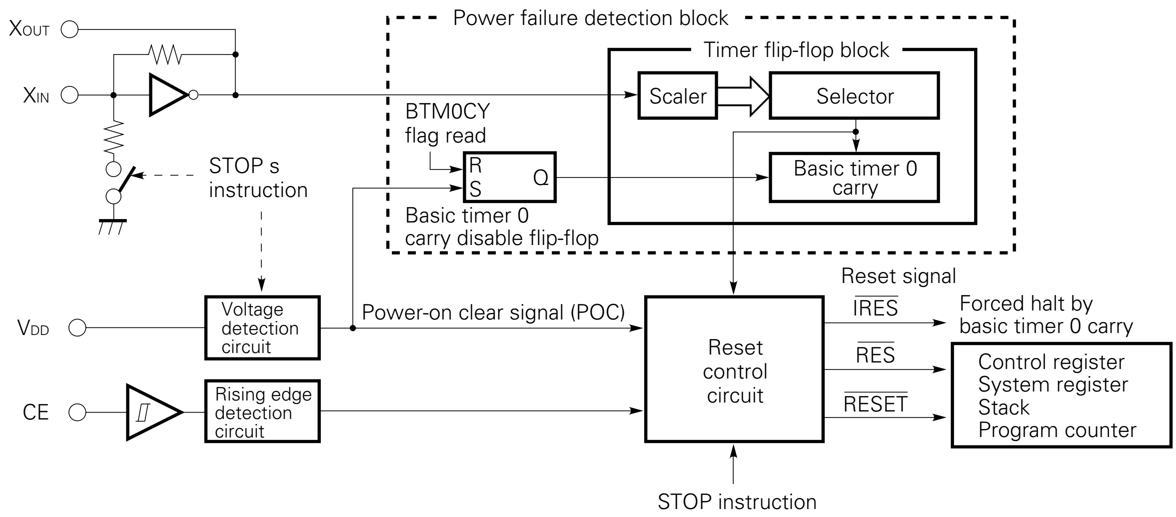
Fig. 20-1 shows the configuration of the reset block.

Device reset is divided into reset by turning on V_{DD} (power-on reset or V_{DD} reset), and reset by CE pin (CE reset).

The power-on reset block consists of a voltage detection circuit that detects the voltage applied 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 edge of the signal input to the CE pin, and a reset control circuit.

Fig. 20-1 Reset Block



20.2 RESET FUNCTION

Power-on reset is applied when V_{DD} rises from a certain voltage. CE reset is applied when the CE pin rises from low level to high level.

Power-on reset initializes the program counter, stack, system register and control registers, and executes the program from address 0000H.

CE reset initializes the program counter, stack, system register and some control registers, and executes the program from address 0000H.

The main differences between power-on reset and CE reset are the operation of the control registers that are initialized and the power failure detection circuit described in **Section 20.6**.

Power-on reset and CE reset are controlled by reset signals $\overline{\text{IRES}}$, $\overline{\text{RES}}$, and $\overline{\text{RESET}}$ output from the reset control circuit in Fig. 20-1.

Table 20-1 shows the $\overline{\text{IRES}}$, $\overline{\text{RES}}$, and $\overline{\text{RESET}}$ signal and power-on reset and CE reset relationship.

The reset control circuit also operates when the clock-stop instruction (STOP s) described in **Section 19** is executed.

Sections 20.3 and **20.4** describe CE reset and power-on reset, respectively.

Section 20.5 describes the relationship between CE reset and power-on reset.

Table 20-1 Relationship Between Internal Reset Signal and Each Reset

Internal reset signal	Output signal			Contents controlled by each reset signal
	At CE reset	At power-on reset	At clock-stop	
$\overline{\text{IRES}}$	×	○	○	Forces the device into the halt state. The halt state is released by the setting of the carry flip-flop of basic timer 0.
$\overline{\text{RES}}$	×	○	○	Initializes some control registers.
$\overline{\text{RESET}}$	○	○	○	Initializes the program counter, stack, system register, and some control registers.

20.3 CE RESET

CE reset is executed by raising the CE pin from low level to high level.

When the CE pin rises to high level, the $\overline{\text{RESET}}$ signal is output and the device is reset in synchronization with the rising edge of the pulse used for the next setting of the carry flip-flop of basic timer 0.

When CE reset is applied, the $\overline{\text{RESET}}$ signal initializes the program counter, stack, system register, and some control registers to their initial value and executes the program from address 0000H.

For the initial values, see the relevant item.

CE reset operation is different when clock-stop is used and when it is not used.

These operations are described in **Sections 20.3.1** and **20.3.2**, respectively.

Section 20.3.3 describes the cautions at CE reset.

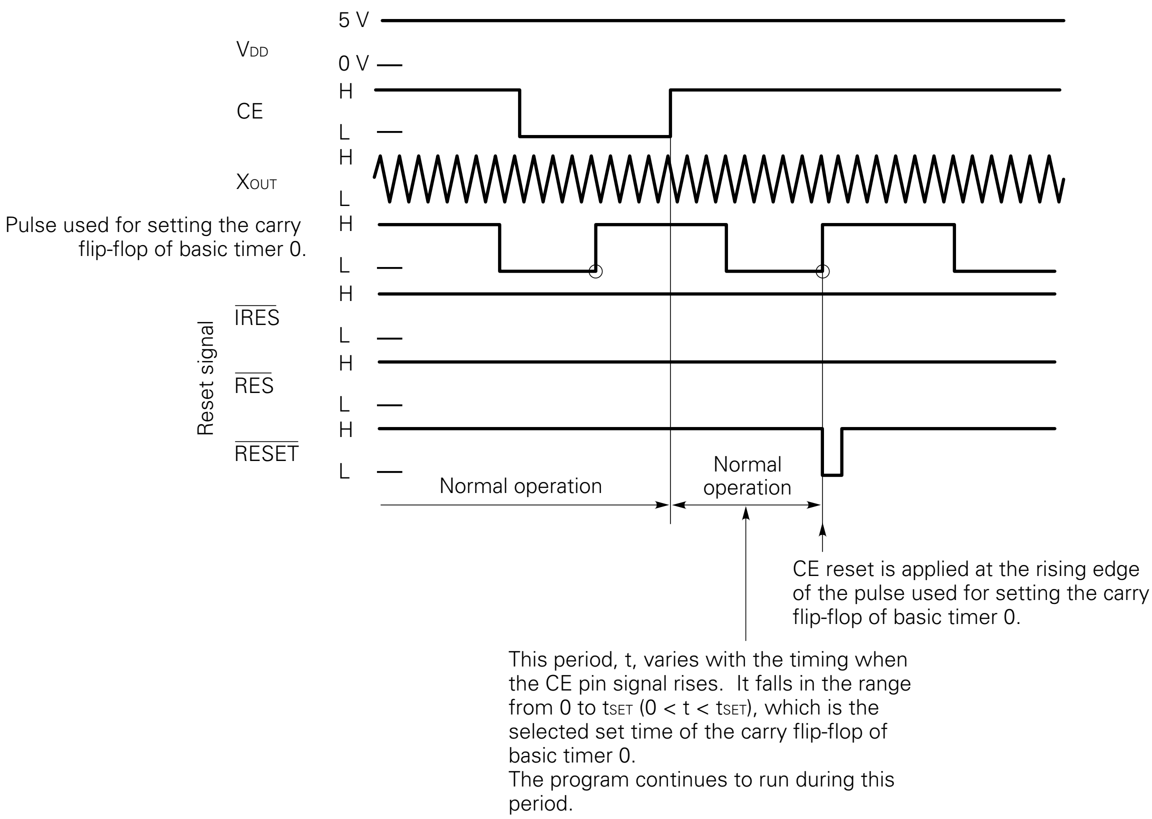
20.3.1 CE Reset When Clock-Stop (STOP s Instruction) Not Used

Fig. 20-2 shows the reset operation.

When clock-stop (STOP s instruction) is not used, the basic timer 0 clock selection register of the control registers is not initialized.

Therefore, after the CE pin becomes high level, the $\overline{\text{RESET}}$ signal is output, and reset is applied at the rising edge of the selected pulse (1 ms, 5 ms, or 100 ms) used for setting the carry flip-flop of basic timer 0.

Fig. 20-2 CE Reset Operation When Clock-Stop Not Used



20.3.2 CE Reset When Clock-Stop (STOP s Instruction) Used

Fig. 20-3 shows the reset operation.

When clock-stop is used, the $\overline{\text{IRES}}$, $\overline{\text{RES}}$ and $\overline{\text{RESET}}$ signals are output at the time the "STOP s" instruction is executed.

At this time, the $\overline{\text{RES}}$ signal initializes the basic timer 0 clock selection register of the control registers to 0000B and sets the basic timer 0 carry flip-flop setting signal to 100 ms.

Since the $\overline{\text{IRES}}$ signal is output continuously while the CE pin is low level, release by basic timer 0 carry is forcibly halted.

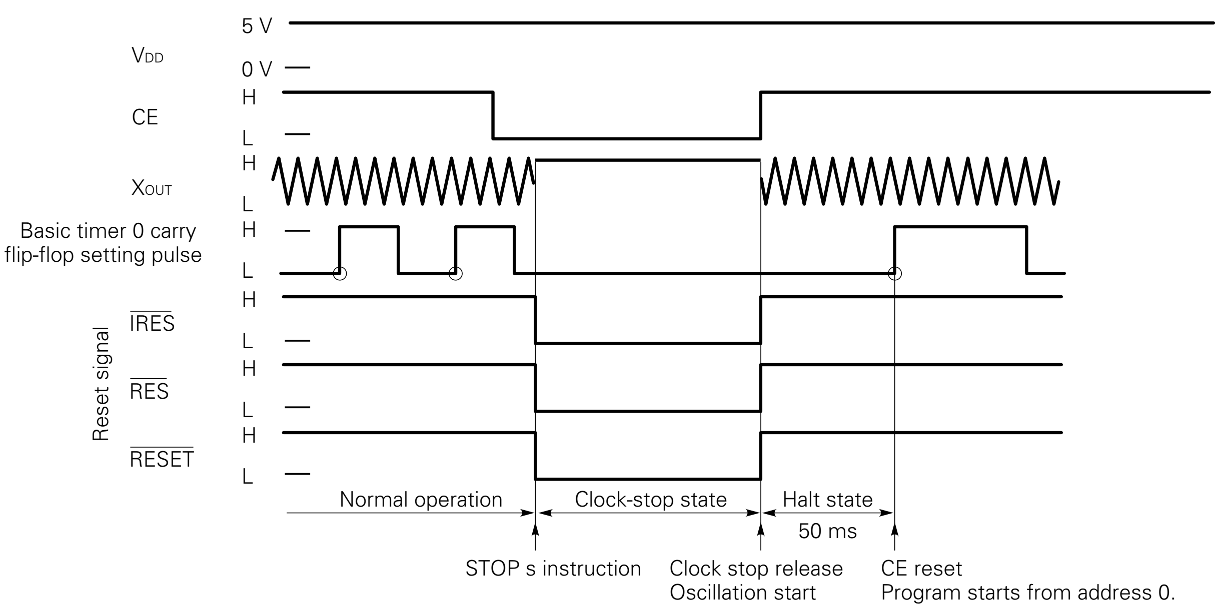
Since the clock itself stops, the device stops operating.

When the CE pin rises to high level, the clock-stop state is released and oscillation begins.

The $\overline{\text{IRES}}$ signal halts release by basic timer 0 carry. When the pulse used for setting the carry flip-flop of basic timer 0 rises after the CE pin rises, the halt state is released and the program starts from address 0.

Since the basic timer 0 carry flip-flop setting pulse is initialized to 100 ms, CE reset is applied 50 ms after the CE pin rises to high level.

Fig. 20-3 CE Reset Operation When Clock-Stop Used



20.3.3 Cautions at CE Reset

When CE reset is used, careful attention must be given to points (1) and (2) below regardless of the instruction being executed.

(1) Time required for clock and other timer processing

When writing a clock program by using basic timer 0 carry and basic timer 2 interrupts, the program must end processing within a certain time.

For details, see **Sections 12.2.6** and **12.4.5.**

(2) Processing of data, flags, etc. used in the program

Care must be exercised when rewriting the contents of data, flags, etc. that cannot be processed by one instruction so that the contents do not change even when CE reset is applied.

20.4 POWER-ON RESET

Power-on reset is executed by raising V_{DD} from a certain voltage (called the power-on clear voltage) or less.

When V_{DD} is less than the power-on clear voltage, the power-on clear signal (POC) is output from the voltage detection circuit shown in Fig. 20-1.

When the power-on clear signal is output, the crystal oscillation circuit stops and the device stops operating.

While the power-on clear signal is being output, the $\overline{\text{IRES}}$, $\overline{\text{RES}}$ and $\overline{\text{RESET}}$ signals are output.

When V_{DD} exceeds the power-on clear voltage, the power-on clear signal is dropped and crystal oscillation starts. At the same time, the $\overline{\text{IRES}}$, $\overline{\text{RES}}$ and $\overline{\text{RESET}}$ signals are also dropped.

Since the $\overline{\text{IRES}}$ signal halts release by basic timer 0 carry, power-on reset is applied at the rising edge of the next basic timer 0 carry flip-flop setting signal.

Since the $\overline{\text{RESET}}$ signal has initialized the basic timer 0 carry flip-flop setting signal to 100 ms, 50 ms after V_{DD} exceeds the power-on clear voltage, reset is applied and the program starts from address 0.

This operation is shown in Fig. 20-4.

At power-on reset, the program counter, stack, system register and control registers are initialized when the power-on clear signal is output.

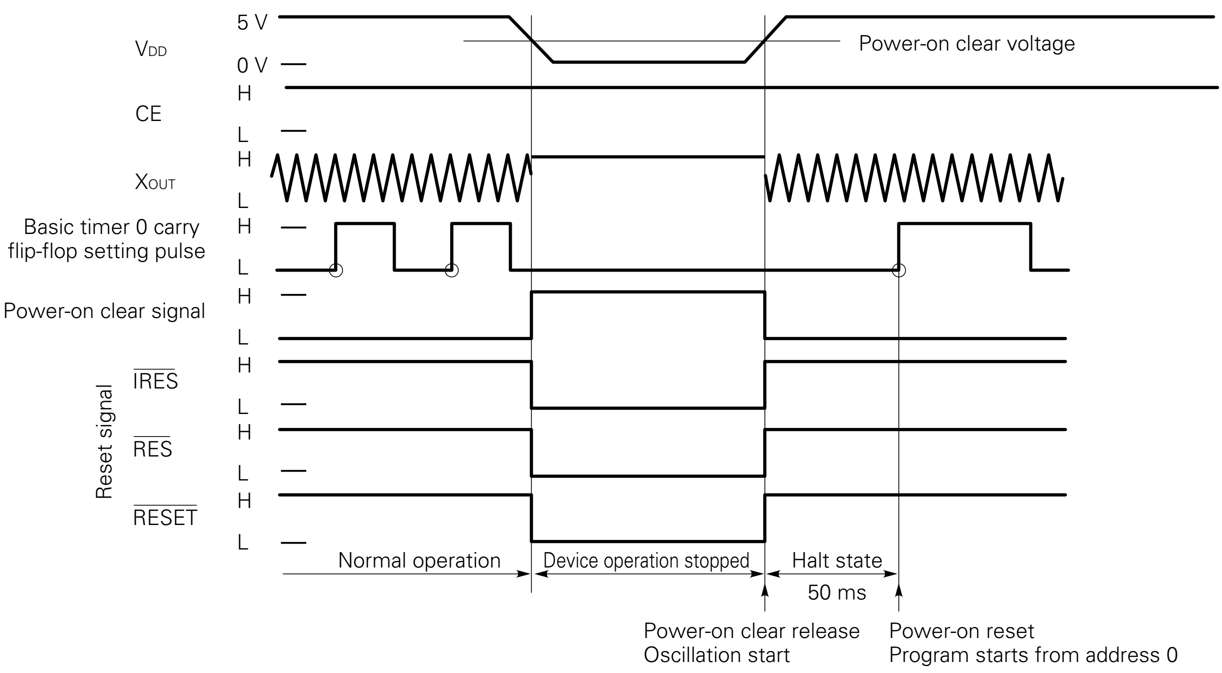
For the initial values, see the relevant items.

During normal operation, the power-on clear voltage is 3.5 V (rated value). In the clock-stop state, the power-on clear voltage becomes 2.2 V (rated value).

Sections 20.4.1 and **20.4.2** describe operation at this time.

Section 20.4.3 describes operation when V_{DD} rises from 0 V,

Fig. 20.4 Power-On Reset Operation



20.4.1 Power-On Reset at Normal Operation

Fig. 20-5 (a) shows power-on reset at normal operation.

As shown in Fig. 20-5 (a), when the V_{DD} drops below 3.5 V, the power-on clear signal is output and operation of the device stops regardless of the input level of the CE pin.

When V_{DD} then rises to 3.5 V or greater, after a 50 ms halt, the program starts from address 0000H.

Normal operation refers to the state in which the clock-stop instruction is not used. This also includes the halt state set by the halt instruction.

20.4.2 Power-On Reset in Clock-Stop State

Fig. 20-5 (b) shows power-on reset in the clock-stop state.

As shown in Fig. 20-5 (b), when V_{DD} drops below 2.2 V, the power-on clear signal is output and device operation stops.

However, since the device is in the clock-stop state, its operation apparently does not change.

When V_{DD} rises to 3.5 V or greater, after a 50 ms halt, the program starts from address 0000H.

20.4.3 Power-On Reset When V_{DD} Rises From 0 V

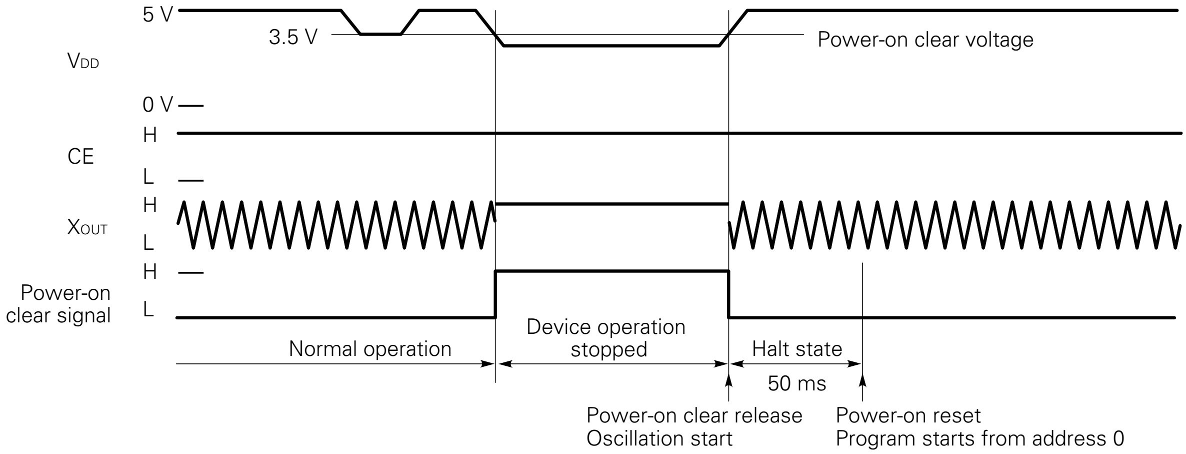
Fig. 20-5 (c) shows power-on reset when V_{DD} rises from 0 V.

As shown in Fig. 20-5 (c), the power-on clear signal is being output while V_{DD} is rising from 0 V to 3.5 V.

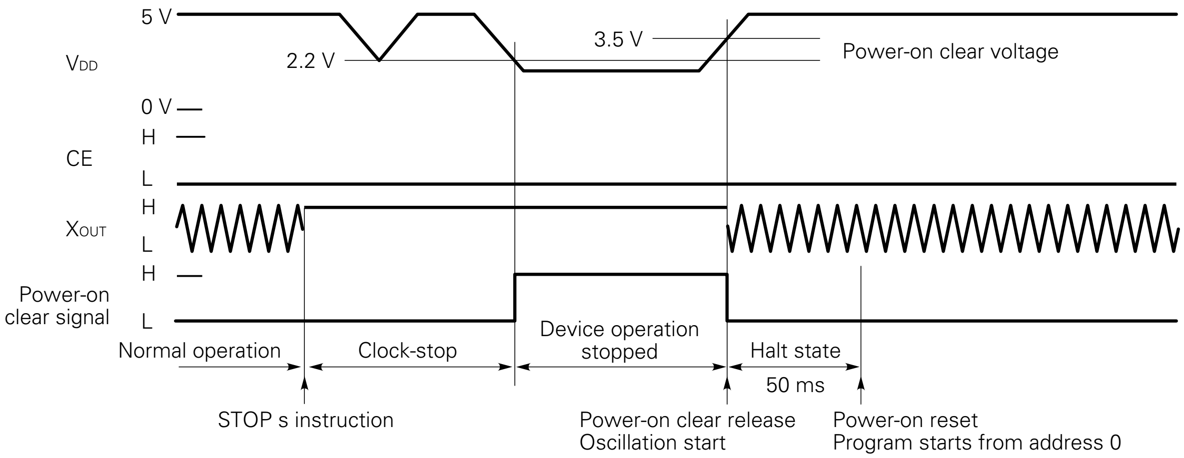
When V_{DD} rises above the power-on clear voltage, the crystal oscillation circuit starts and after a 50 ms halt, the program starts from address 0000H.

Fig. 20-5 Power-On Reset and V_{DD}

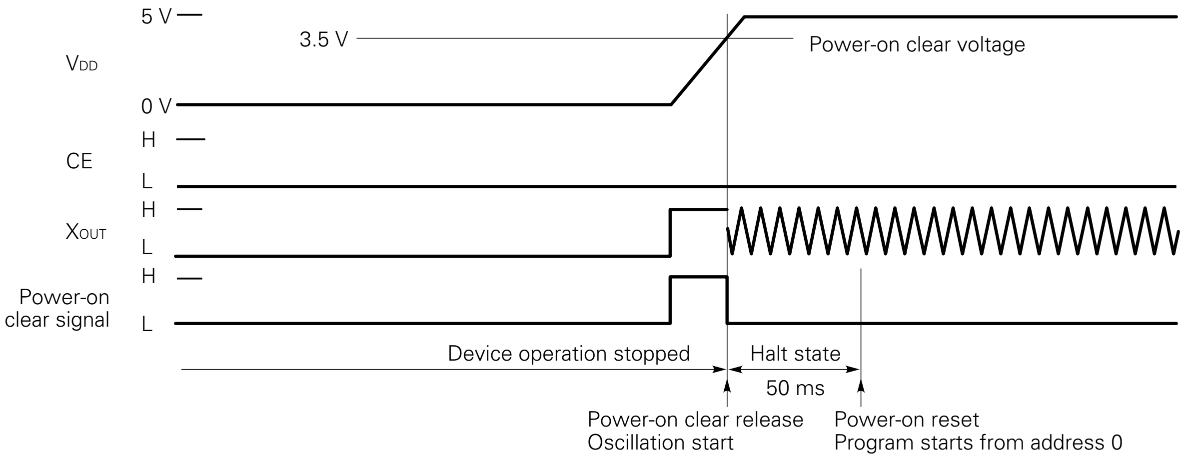
(a) During normal operation (including halt state)



(b) At clock-stop



(c) When V_{DD} rises from 0 V



20.5 RELATIONSHIP BETWEEN CE RESET AND POWER-ON RESET

When V_{DD} is first turned on, power-on reset and CE reset may be applied simultaneously.

Sections 20.5.1 through **20.5.3** describe this reset operation.

Section 20.5.4 describes the cautions when V_{DD} rises.

20.5.1 When V_{DD} Pin and CE Pin Rise Simultaneously

Fig. 20-6 (a) shows the reset operation. Power-on reset starts the program from address 0000H.

20.5.2 When CE Pin Raised in Forced Halt State Caused by Power-On Reset.

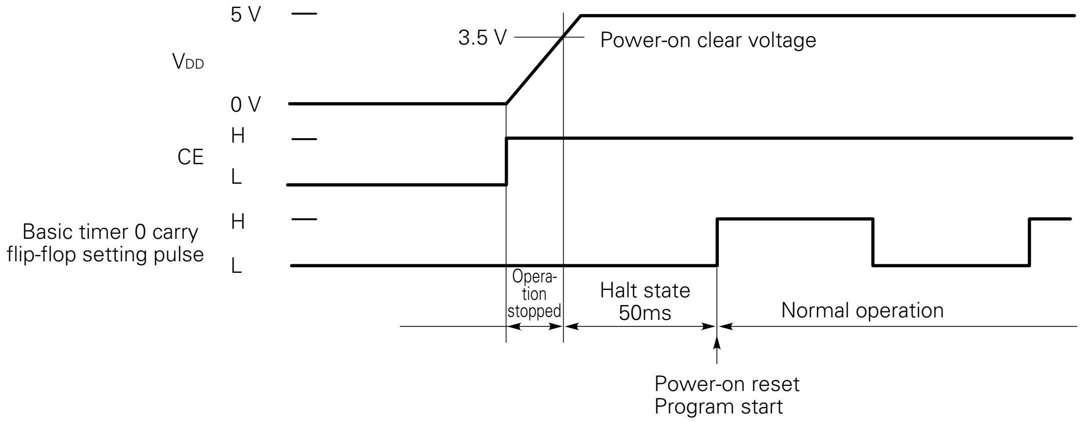
Fig. 20-6 (b) shows the reset operation. Power-on reset starts the program from address 0000H, as in **Section 20.5.1.**

20.5.3 When CE Pin Raised After Power-On Reset

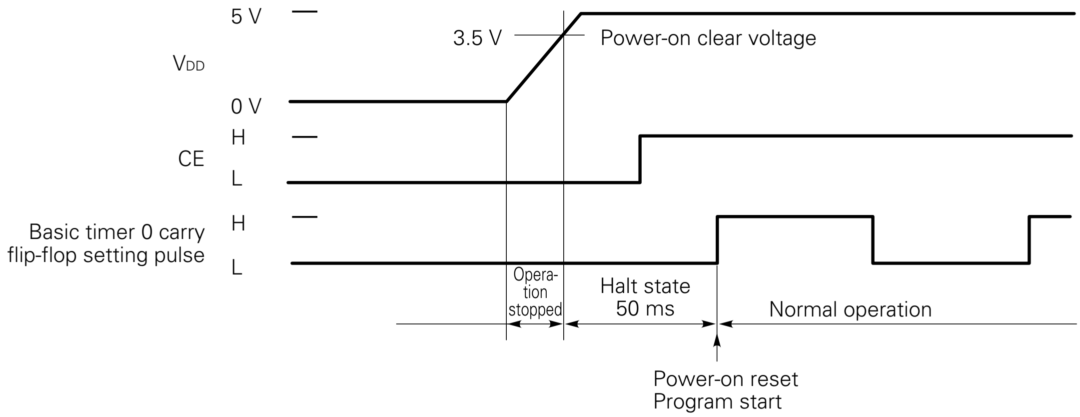
Fig. 20-6 (c) shows the reset operation. Power-on reset starts the program from address 0000H. CE reset restarts the program from address 0000H at the rising edge of the next basic timer 0 carry flip-flop setting signal.

Fig. 20-6 Relationship Between Power-On Reset and CE Reset

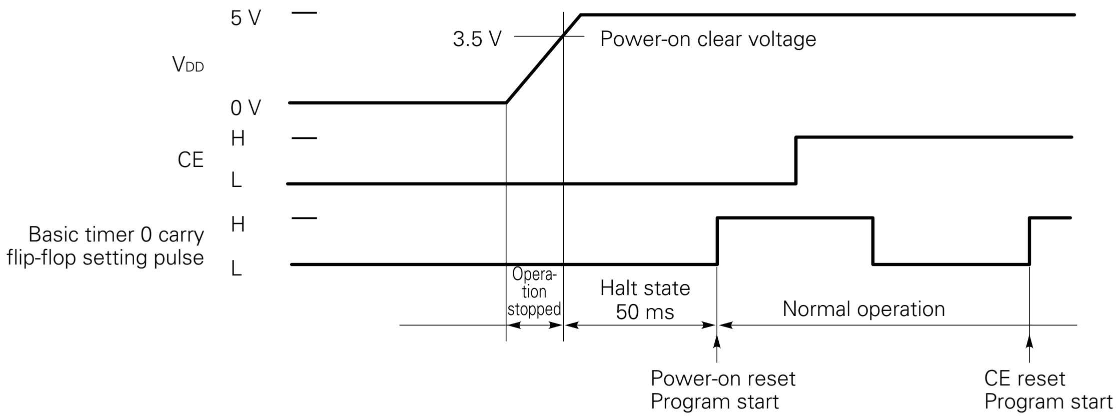
(a) When V_{DD} and CE pin raised simultaneously



(b) When CE Pin Raised in Halt State



(c) When CE Pin Raised After Power-On Reset



20.5.4 Cautions When V_{DD} Raised

When V_{DD} is raised, careful attention must be given to points (1) and (2) below.

(1) When V_{DD} raised from power-on clear voltage

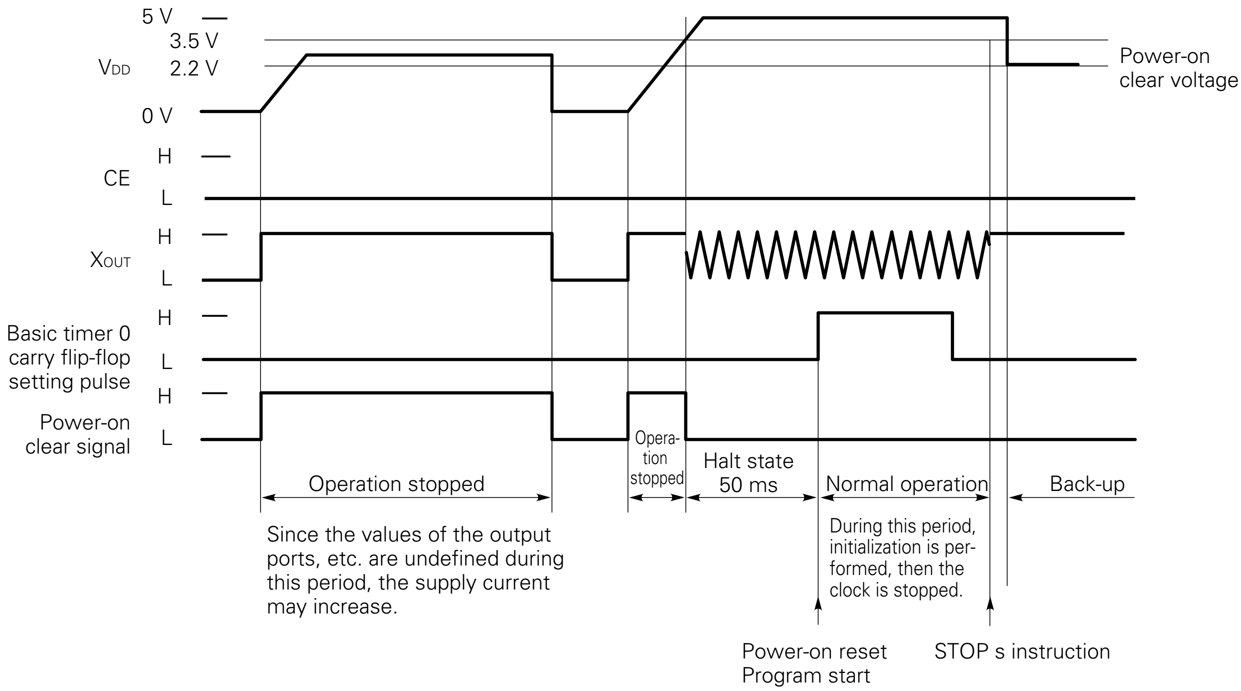
When V_{DD} is raised, it must raised to 3.5 V or greater, once.

This is shown in Fig. 20-7.

As shown in Fig. 20-7, when a voltage under 3.5 V is applied when V_{DD} is turned on in a program that uses clock-stop to back up V_{DD} at 2.2 V, for example, the power-on clear signal continues to be output and the program does not run.

Since the device output port outputs an undefined value, the supply current increases, according to the situation, reducing the back-up time with a battery considerably.

Fig. 20-7 Caution When V_{DD} Raised



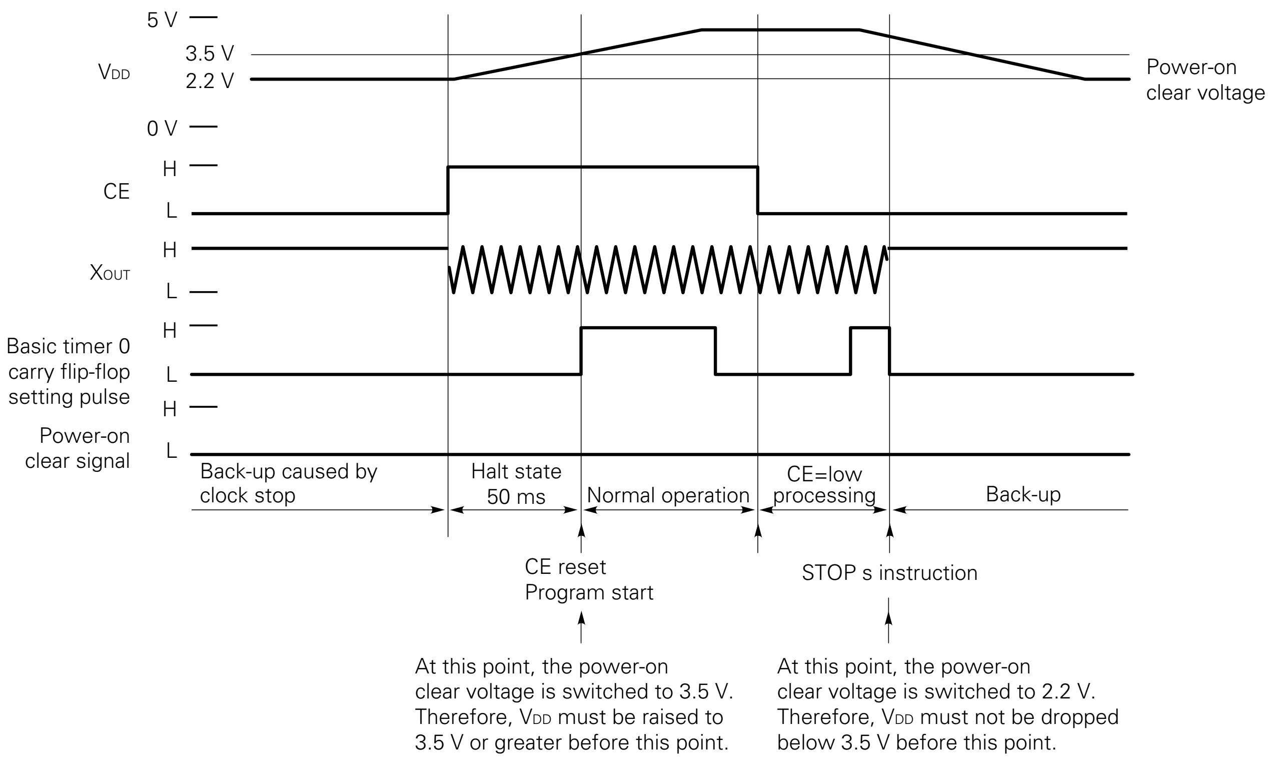
(2) At return from clock-stop state

When returning from the back-up state when clock-stop is used to back-up V_{DD} at 2.2 V, V_{DD} must be raised to 3.5 V or greater within 50 ms after the CE pin becomes high level.

As shown in Fig. 20-8, return from the clock-stop state is performed by CE reset. Since the power-on clear voltage is switched to 3.5 V 50 ms after the CE pin is raised, if V_{DD} is not 3.5 V or greater at this time, power-on reset is applied.

The same caution is necessary when V_{DD} is dropped.

Fig. 20-8 Return From Clock-Stop State



20.6 POWER FAILURE DETECTION

Power failure detection is used to judge whether the device is reset by turning on V_{DD} or by the CE pin, as shown in Fig. 20-9.

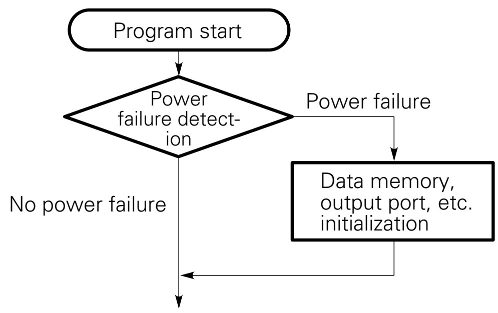
Since the contents of the data memory, output ports, etc. become “undefined” when V_{DD} is turned on, they are initialized by power failure detection.

There are two power failure detection methods. One method detects a power failure by using a power failure detection circuit to detect the BTM0CY flag and the other method detects the contents of the data memory (RAM judgment).

Sections 20.6.1 and **20.6.2** describe the power failure detection circuit, and power failure detection by BTM0CY flag, respectively.

Sections 20.6.3 and **20.6.4** describe power failure detection with the RAM judgment method.

Fig. 20-9 Power Failure Detection Flowchart



20.6.1 Power Failure Detection Circuit

As shown in Fig. 20-1, the power failure detection circuit consists of a voltage detection circuit and timer carry disable flip-flop that is reset by the output (power-on clear signal) of the voltage detection circuit, and basic timer 0 carry.

The basic timer 0 carry disable flip-flop is set (1) by the power-on clear signal and is reset (0) when a BTM0CY flag read instruction is executed.

When the basic timer 0 carry disable flip-flop is set (1), the BTM0CY flag is not set (1).

That is, when the power-on clear signal is output (at power-on reset), the program starts in the state in which the BTM0CY flag is reset and the setting disabled state is set until a BTM0CY read instruction is executed thereafter.

Once a BTM0CY read instruction is executed, the BTM0CY flag is set at each rising edge of the basic timer 0 carry flip-flop setting pulse thereafter. When reset is applied to the device, the contents of the BTM0CY flag are monitored. If the BTM0CY flag has been reset (0), power-on reset (power failure) is judged and if the BTM0CY flag has been set (1), CE reset (no power failure) is judged.

Since the voltage that can detect a power failure is the same as the voltage applied by power-on reset, V_{DD} becomes 3.5 V at crystal oscillation and 2.2 V at clock-stop.

Fig. 20-10 shows the BTM0CY flag state transition.

Fig. 20-11 shows timing chart and BTM0CY flag operation specified in Fig. 20-10.

Fig. 20-10 BTM0CY Flag State Transition

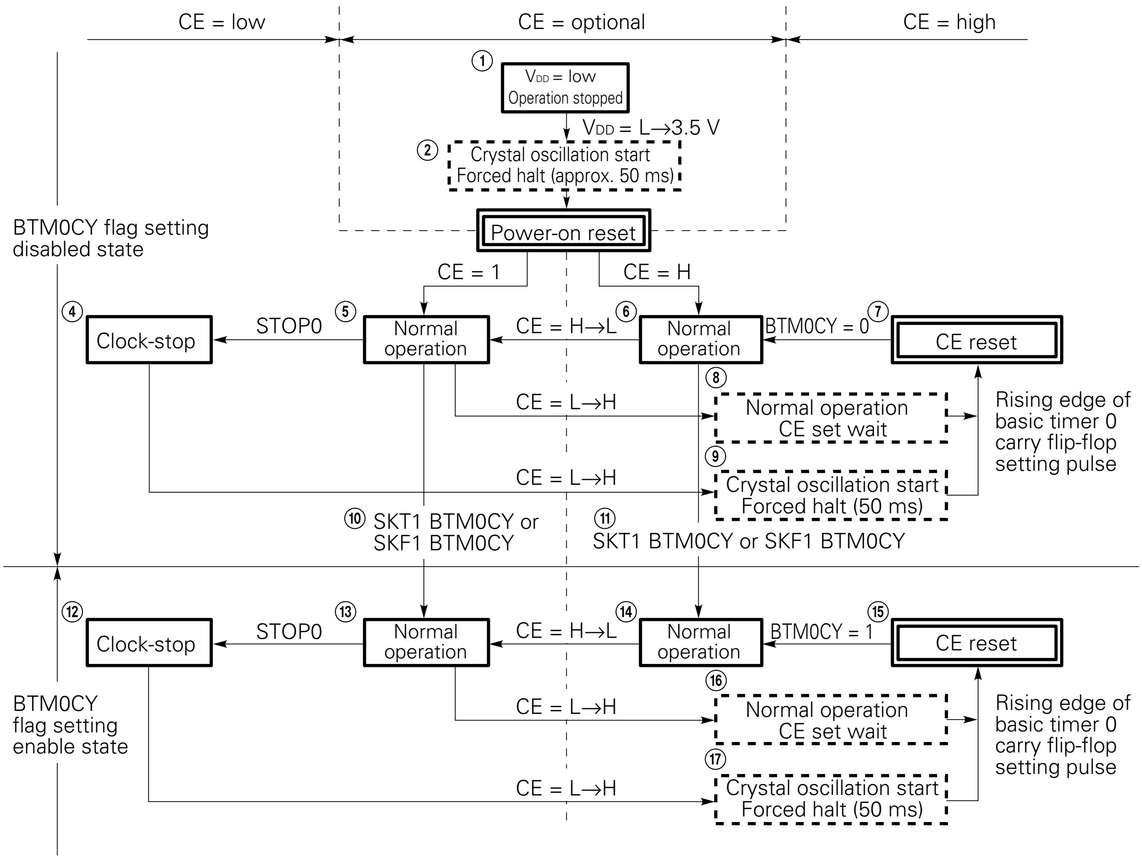
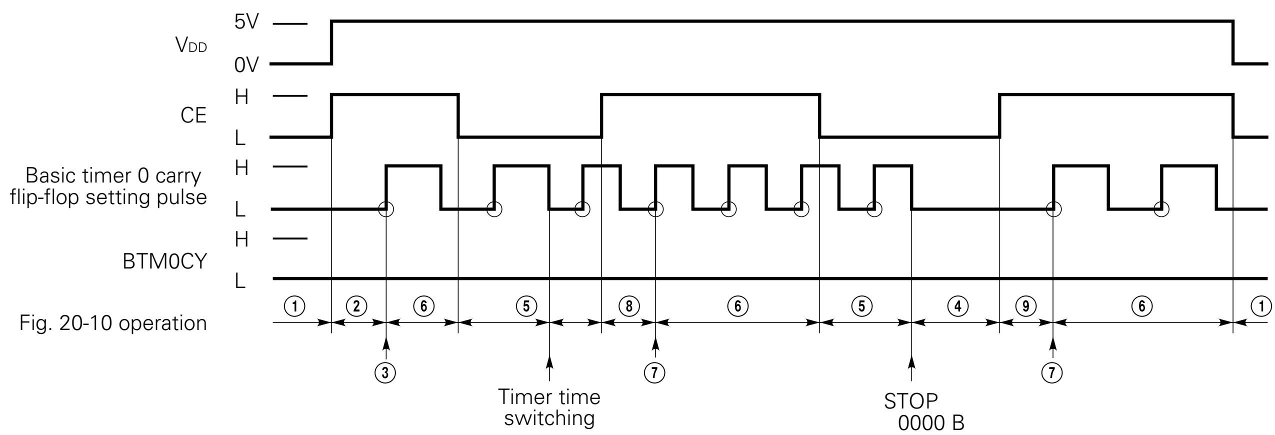
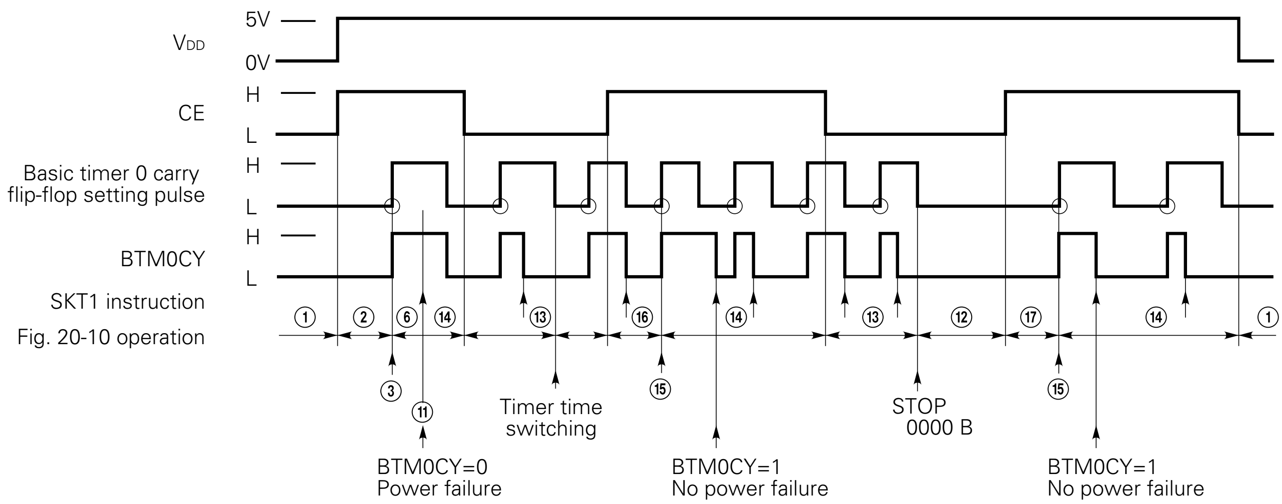


Fig. 20-11 BTM0CY Flag Operation

(a) When BTM0CY flag not detected even once (neither SKT1 BTM0CY nor SKF1 BTM0CY executed)



(b) When power failure detected with BTM0CY flag



20.6.2 Cautions at Power Failure Detection with BTM0CY Flag

When clock counting, etc. is performed with the BTM0CY flag, careful attention must be given to the following points.

(1) Clock updating

When writing a clock program by using basic timer 0, the clock must be updated after a power failure. This is because the BTM0CY flag is reset (0) and one clock count is lost by BTM0CY flag reading when a power failure is detected.

(2) Clock update processing time

When the clock is updated, its processing must end before the next rising edge of the basic timer 0 flip-flop setting pulse.

This is because if the CE pin rises to high level during clock update processing, the clock update processing will not be executed up to the end and a CE reset will be applied.

For (1) and (2) above, see **Section 12.2.6 (3).**

When processing is performed at a power failure, careful attention must be given to the following point.

(3) Power failure detection timing

When clock counting, etc. is performed with the BTM0CY flag, the flag must be read for power-failure detection before the next rising edge of the basic timer 0 carry flip-flop setting pulse, after a program starts from address 0000H.

This is because when the basic timer 0 carry flip-flop setting time is set to 5 ms, for instance, and power failure detection is performed 6 ms after the program starts, one BTM0CY flag is lost.

See **Section "12.2.6 (3).**

As shown in the example on the next page, power failure detection and initialization must be performed within the basic timer 0 carry flip-flop set time.

This is because when the CE pin is raised and CE reset is applied during power failure processing and initialization, these processings are interrupted and a problem may occur.

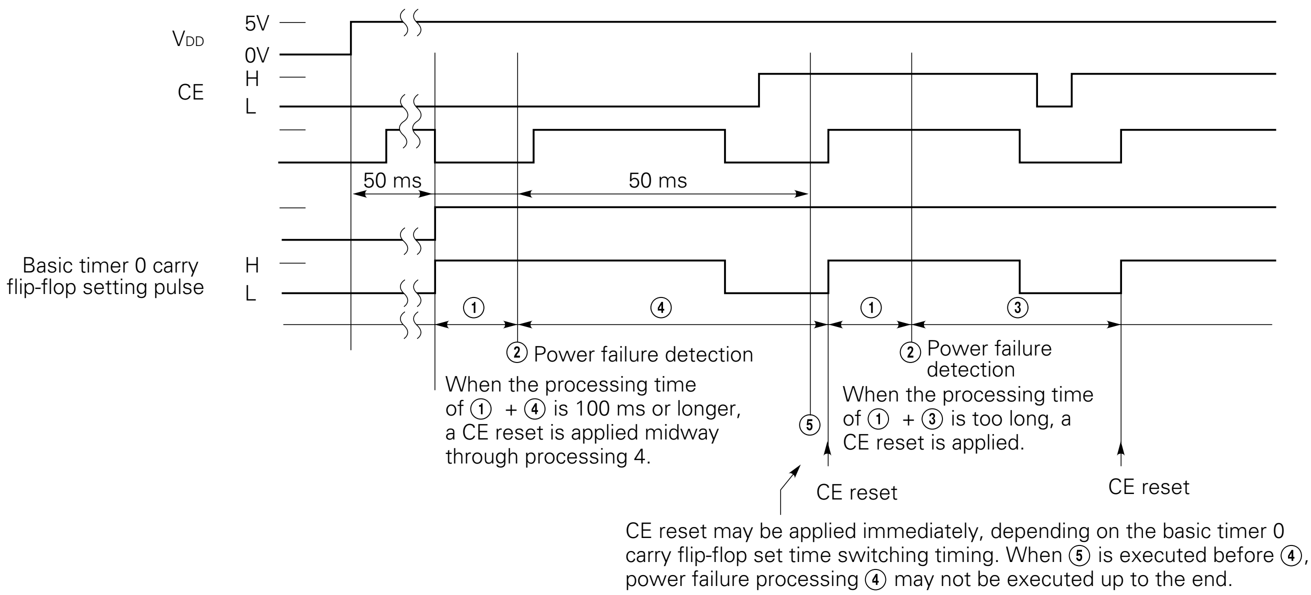
When the basic timer 0 carry flip-flop set time is changed in initialization, an instruction that makes the change must be executed at the end of initialization.

This is also because when the basic timer 0 carry flip-flop set time is switched before initialization as shown in the example on the next page, initialization by CE reset may not be executed up to the end.

Example

```
Sample program
START:                                      ; Program address 0000H
; ①
        Reset processing
; ②
        SKT1    BTM0CY                      ; Power failure detection
        BR      INITIAL
BACKUP:
; ③
        Clock updating
        BR      MAIN
INITIAL:
; ④
        Initialization
; ⑤
        INITFLG NOT BTM0CK1, BTM0CK0        ; Built-in macro
                                            ; Sets basic timer 0 carry flip-flop set time to 5 ms.
MAIN:
        Main process
        SKT1    BTM0CY
        BR      MAIN
        Clock updating
        BR      MAIN
```

Operation example



20.6.3 Power Failure Detection by RAM Judgment

The RAM judgment method detects a power failure by judging if the contents of the data memory at a specified address are the specified value when the device was reset.

A sample program that detects a power failure by RAM judgment is shown on the next page.

The contents of data memory when V_{DD} is turned on are "undefined". The RAM judgment method performs power failure detection by comparing the "undefined value" with the "specified value".

Therefore, power failure detection may be judged erroneously, as described in **Section 20.6.4.**

However, the advantage of using the RAM judgment method is that, as shown in Table 20-2, back-up down to a lower voltage than that detected by power detection circuit is possible.

Table 20-2 Comparison of Power Failure Judgment Circuit and Power Failure Detection by RAM Judgment

	Power failure detection circuit		RAM judgment	
Data hold voltage (at clock-stop)	Actual value	Rating	Actual value	Rating
	2 – 2.2 V	2.2 V	0 – 1 V	2.2 V
Remarks	No erroneous operation		Erroneous operation possible	

Sample program which detects power failure by RAM judgment

```
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       M012, #DATA0   ; If M012 = DATA0 and
          SUB       M034, #DATA1   ; M034 = DATA1 and
          SUB       M056, #DATA2   ; M056 = DATA2 and
          BANK1
          SUB       M107, #DATA3   ; M107 = DATA3 and
          SUB       M128, #DATA4   ; M128 = DATA4 and
          SUB       M16F, #DATA5   ; M16F = DATA5
          BANK0
          SKF1      Z
          BR        BACKUP         ; Branches to BACKUP
; INITIAL:
          [Initialization]
          MOV       M012, #DATA0
          MOV       M034, #DATA1
          MOV       M056, #DATA2
          BANK1
          MOV       M107, #DATA3
          MOV       M128, #DATA4
          MOV       M16F, #DATA5
          BR        MAIN
BACKUP:
          [Back-up processing]
MAIN:
          [Main processing]
```

20.6.4 Cautions at Power Failure Detection by RAM Judgment

Since the data memory value when V_{DD} is turned on is “undefined”, careful attention must be given to points (1) and (2) below.

(1) Comparison data

When n bits of data memory is to be compared by RAM judgment, the probability that the data memory value when V_{DD} is turned on may unexpectedly match the specified value is $(1/2)^n$.

That is, there is a $(1/2)^n$ probability that power failure detection by RAM judgment will be judged back-up.

To reduce this probability, the largest number of bits possible are compared.

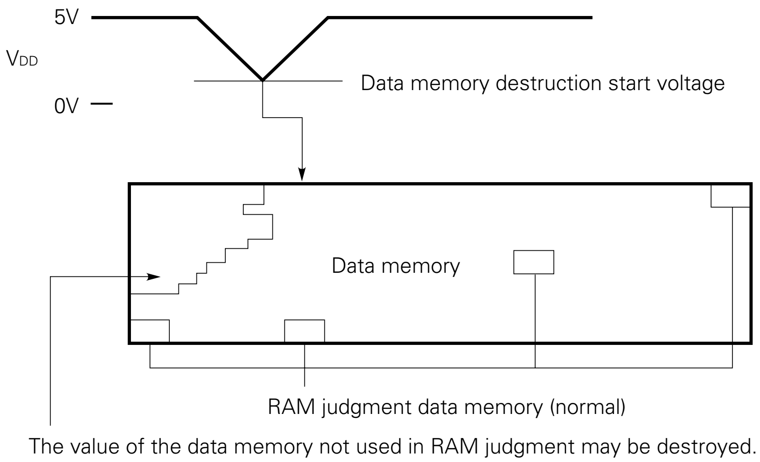
Since, from experience, the contents of data memory when V_{DD} is turned on easily becomes the same value, such as “0000B” and “1111B”, comparison data that mixes “0” and “1”, such as “1010B” and “0110B”, reduces the possibility of erroneous judgment.

(2) Program cautions

When V_{DD} is raised from a voltage that begins to destroy the data memory as shown in Fig. 20-12, even if the value of data memory to be compared is normal, other values may be destroyed.

Since power failure detection by RAM judgment judges it as back-up, the program must be written so that it does not crash even if the data memory is destroyed.

Fig. 20-12 V_{DD} and Destruction of Data Memory



21. INSTRUCTION SET

21.1 LIST OF INSTRUCTION SET

BIN (b_{14}-b_{11})	HEX	b_{15} = 0		b_{15} = 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 INC MOVT BR CALL SYSCAL RET RETSK EI DI RETI PUSH POP GET PUT PEEK POKE RORC STOP HALT NOP	AR IX DBF, @AR @AR @AR entry AR AR DBF, p p, DBF WR, rf rf, WR r s h 		
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

21.2 INSTRUCTIONS

Legend

AR	:	Address register
ASR	:	Address stack register pointed to by the stack pointer
addr	:	Program memory address (11 low-order bits)
BANK	:	Bank register
CMP	:	Compare flag
CY	:	Carry flag
DBF	:	Data buffer
entry	:	Program memory address (bits 10 to 8, bits 3 to 0)
$entry_H$	:	Program memory address (bits 10 to 8)
$entry_L$	:	Program memory address (bits 3 to 0)
h	:	Halt release condition
INTEF	:	Interrupt enable flag
INTR	:	Register automatically saved in the stack when an interrupt occurs
INTSK	:	Interrupt stack register
IX	:	Index register
MP	:	Data memory row address pointer
MPE	:	Memory pointer enable flag
m	:	Data memory address specified by m_R and m_C
m_R	:	Data memory row address (high-order)
m_C	:	Data memory column address (low-order)
n	:	Bit position (four bits)
n4	:	Immediate data (four bits)
PAGE	:	Page (Bits 12 and 11 of the program counter)
PC	:	Program counter
p	:	Peripheral address
p_H	:	Peripheral address (three high-order bits)
p_L	:	Peripheral address (four low-order bits)
r	:	General register column address
rf	:	Register file address
rf_R	:	Register file address (three high-order bits)
rf_C	:	Register file address (four low-order bits)
SGR	:	Segment register (Bit 13 of the program counter)
SP	:	Stack pointer
s	:	Stop release condition
WR	:	Window register
(×)	:	Contents of ×

Instruction set	Mnemonic	Operand	Operation	Instruction code			
				Op code	Operand		
Add	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
Subtract	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
Test	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
Compare	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 set	Mne-monic	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	$PC_{10\text{-}0}$ ← addr, PAGE ← 0	01100	addr		
			$PC_{10\text{-}0}$ ← addr, PAGE ← 1	01101			
			$PC_{10\text{-}0}$ ← addr, PAGE ← 2	01110			
			$PC_{10\text{-}0}$ ← addr, PAGE ← 3	01111			
		@AR	PC ← AR	00111	000	0100	0000
Sub-routine	CALL	addr	SP ← SP − 1, ASR ← PC, $PC_{12,11}$ ← 0, $PC_{10\text{-}0}$ ← addr	11100	addr		
		@AR	SP ← SP − 1, ASR ← PC, PC ← AR	00111	000	0101	0000
	SYSCAL	entry	SP ← SP − 1, ASR ← PC, SGR ← 1, $PC_{12,11}$ ← 0, $PC_{10\text{-}8}$ ← $entry_H$, $PC_{7\text{-}4}$ ← 0, $PC_{3\text{-}0}$ ← $entry_L$	00111	$entry_H$	0000	$entry_L$
	RET		PC ← ASR, SP ← SP + 1	00111	000	0101	0000
	RETSK		PC ← ASR, SP ← SP + 1 and skip	00111	000	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

21.3 ASSEMBLER (AS17K) BUILT-IN MACRO INSTRUCTIONS

Legend

flag n : FLG-type symbol
< > : An operand enclosed in < > is optional.

	Mnemonic	Operand	Operation	n
Built-in macro	SKTn	flag 1, ⋯ flag n	if (flag 1) to (flag n) = all "1", then skip	$1 \leq n \leq 4$
	SKFn	flag 1, ⋯ flag n	if (flag 1) to (flag n) = all "0", then skip	$1 \leq n \leq 4$
	SETn	flag 1, ⋯ flag n	(flag 1) to (flag n) ← 1	$1 \leq n \leq 4$
	CLRn	flag 1, ⋯ flag n	(flag 1) to (flag n) ← 0	$1 \leq n \leq 4$
	NOTn	flag 1, ⋯ flag n	if (flag n) = "0", then (flag n) ← 1 if (flag n) = "1", then (flag n) ← 0	$1 \leq n \leq 4$
	INITFLG	<NOT>flag 1, ⋯ <<NOT>flag n>	if description = NOT flag n, then (flag n) ← 0 if description = flag n, then (flag n) ← 1	$1 \leq n \leq 4$
	BANKn		(BANK) ← n	$0 \leq n \leq 2$

22. RESERVED SYMBOLS

22.1 DATA BUFFER (DBF)

Symbol	Attribute	Value	Read/ write	Description
DBF3	MEM	0.0CH	R/W	DBF bits 15 to 12
DBF2	MEM	0.0DH	R/W	DBF bits 11 to 8
DBF1	MEM	0.0EH	R/W	DBF bits 7 to 4
DBF0	MEM	0.0FH	R/W	DBF bits 3 to 0

22.2 SYSTEM REGISTER (SYSREG)

Symbol	Attribute	Value	Read/ write	Description
AR3	MEM	0.74H	R/W	Bits 15 to 12 of the address register
AR2	MEM	0.75H	R/W	Bits 11 to 8 of the address register
AR1	MEM	0.76H	R/W	Bits 7 to 4 of the address register
AR0	MEM	0.77H	R/W	Bits 3 to 0 of the 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

22.3 VRAM BANK REGISTER

Symbol	Attribute	Value	Read/ write	Description
VRAMBANK	MEM	2.73H	R/W	VRAM bank register

22.4 PORT REGISTER

Symbol	Attribute	Value	Read/ write	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
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/W	Bit 3 of port 1D
P1D2	FLG	1.73H.2	R/W	Bit 2 of port 1D
P1D1	FLG	1.73H.1	R/W	Bit 1 of port 1D
P1D0	FLG	1.73H.0	R/W	Bit 0 of port 1D
P2A0	FLG	2.70H.0	R/W	Bit 0 of port 2A

Symbol	Attribute	Value	Read/ write	Description
P2B3	FLG	2.71H.3	R/W	Bit 3 of port 2B
P2B2	FLG	2.71H.2	R/W	Bit 2 of port 2B
P2B1	FLG	2.71H.1	R/W	Bit 1 of port 2B
P2B0	FLG	2.71H.0	R/W	Bit 0 of port 2B
P2C3	FLG	2.72H.3	R/W	Bit 3 of port 2C
P2C2	FLG	2.72H.2	R/W	Bit 2 of port 2C
P2C1	FLG	2.72H.1	R/W	Bit 1 of port 2C
P2C0	FLG	2.72H.0	R/W	Bit 0 of port 2C
P2D2	FLG	2.6FH.2	R/W	Bit 2 of port 2D
P2D1	FLG	2.6FH.1	R/W	Bit 1 of port 2D
P2D0	FLG	2.6FH.0	R/W	Bit 0 of port 2D

22.5 REGISTER FILES

Symbol	Attribute	Value	Read/ write	Description
SP	MEM	0.81H	R/W	Stack pointer
CEEDET	FLG	0.82H.0	R	CE pin edge detection flag
PWM8SEL	FLG	0.83H.0	R/W	$PWM_8/P2A_0$ pin selection flag
PWM7SEL	FLG	0.84H.3	R/W	$PWM_7/P2B_3$ pin selection flag
PWM6SEL	FLG	0.84H.2	R/W	$PWM_6/P2B_2$ pin selection flag
PWM5SEL	FLG	0.84H.1	R/W	$PWM_5/P2B_1$ pin selection flag
PWM4SEL	FLG	0.84H.0	R/W	$PWM_4/P2B_0$ pin selection flag
PWM3SEL	FLG	0.85H.3	R/W	$PWM_3/P2C_3$ pin selection flag
PWM2SEL	FLG	0.85H.2	R/W	$PWM_2/P2C_2$ pin selection flag
PWM1SEL	FLG	0.85H.1	R/W	$PWM_1/P2C_1$ pin selection flag
PWM0SEL	FLG	0.85H.0	R/W	$PWM_0/P2C_0$ pin selection flag
WTMHLD	FLG	0.86H.3	R/W	Watch timer hold flag
CKOSEL	FLG	0.86H.1	R/W	$P1B_1$/CKOUT pin selection flag
XTSEL	FLG	0.86H.0	R/W	Function selection flag of $P0D_1$ and $P0D_0$ pins
CE	FLG	0.87H.0	R	CE pin status flag
SIO0CH	FLG	0.88H.3	R/W	SIO0 channel selection flag
SB	FLG	0.88H.2	R/W	SIO0 mode selection flag
SIO0MS	FLG	0.88H.1	R/W	SIO0 clock mode selection flag
SIO0TX	FLG	0.88H.0	R/W	SIO0 TX/RX selection flag
TM0CK	FLG	0.89H.0	R/W	Timer 0 clock selection flag
BTM2EXCK	FLG	0.8AH.3	R/W	Bit 3 of basic timer 2 clock selection flag
BTM2ZX	FLG	0.8AH.2	R/W	Bit 2 of basic timer 2 clock selection flag
BTM2CK1	FLG	0.8AH.1	R/W	Bit 1 of basic timer 2 clock selection flag
BTM2CK0	FLG	0.8AH.0	R/W	Bit 0 of basic timer 2 clock selection flag
BTM1EXCK	FLG	0.8BH.3	R/W	Bit 3 of basic timer 1 clock selection flag
BTM1ZX	FLG	0.8BH.2	R/W	Bit 2 of basic timer 1 clock selection flag
BTM1CK1	FLG	0.8BH.1	R/W	Bit 1 of basic timer 1 clock selection flag
BTM1CK0	FLG	0.8BH.0	R/W	Bit 0 of basic timer 1 clock selection flag
BTM0CK1	FLG	0.8CH.1	R/W	Bit 1 of basic timer 0 clock selection flag
BTM0CK0	FLG	0.8CH.0	R/W	Bit 0 of basic timer 0 clock selection flag
TM0RPT	FLG	0.8DH.2	R/W	Timer 0 mode (repeat) selection flag
TM0RES	FLG	0.8DH.1	W	Timer 0 reset flag
TM0EN	FLG	0.8DH.0	R/W	Timer 0 start/stop flag
TM0OVF	FLG	0.8EH.0	R	Timer 0 overflow detection flag

Symbol	Attribute	Value	Read/write	Description
IGRP1SL	FLG	0.8FH.1	R/W	Interrupt request group 1 selection flag
IGRP0SL	FLG	0.8FH.0	R/W	Interrupt request group 0 selection flag
HSCGT1	FLG	0.91H.1	R/W	Bit 1 of H_{SYNC}-counter gate-mode selection flag
HSCGT0	FLG	0.91H.0	R/W	Bit 0 of H_{SYNC}-counter gate-mode selection flag
HSCGOSTT	FLG	0.92H.3	R	H_{SYNC}-counter gate open status flag
PLLRFCK3	FLG	0.93H.3	R/W	Bit 3 of PLL reference clock selection flag
PLLRFCK2	FLG	0.93H.2	R/W	Bit 2 of PLL reference clock selection flag
PLLRFCK1	FLG	0.93H.1	R/W	Bit 1 of PLL reference clock selection flag
PLLRFCK0	FLG	0.93H.0	R/W	Bit 0 of PLL reference clock selection flag
WTMRES3	FLG	0.94H.3	R/W	Watch timer (day setting register) reset flag
WTMRES2	FLG	0.94H.2	R/W	Watch timer (second/minute/hour setting register) reset flag
WTMRES1	FLG	0.94H.1	R/W	Watch timer (basic clock) reset flag
WTMRES0	FLG	0.94H.0	R/W	Watch timer (all) reset flag
INTNCMD2	FLG	0.95H.2	R/W	Bit 2 of INT_{NC} pulse width selection flag
INTNCMD1	FLG	0.95H.1	R/W	Bit 1 of INT_{NC} pulse width selection flag
INTNCMD0	FLG	0.95H.0	R/W	Bit 0 of INT_{NC} pulse width selection flag
BTM1CY	FLG	0.96H.0	R	Basic timer 1 carry flag
BTM0CY	FLG	0.97H.0	R	Basic timer 0 carry flag
SBACK	FLG	0.98H.3	R/W	SIO0 acknowledge flag
SIO0NWT	FLG	0.98H.2	R/W	SIO0 not-wait flag
SIO0WRQ1	FLG	0.98H.1	R/W	Bit 1 of SIO0 wait timing setting flag
SIO0WRQ0	FLG	0.98H.0	R/W	Bit 0 of SIO0 wait timing setting flag
SIO0WSTT	FLG	0.99H.0	R	Judge flag of SIO0 wait status
TM1CK1	FLG	0.9AH.1	R/W	Bit 1 of timer 1 clock selection flag
TM1CK0	FLG	0.9AH.0	R/W	Bit 0 of timer 1 clock selection flag
TM1RES	FLG	0.9BH.1	R/W	Timer 1 reset flag
TM1EN	FLG	0.9BH.0	R/W	Timer 1 enable flag
SIO1TS	FLG	0.9CH.3	R/W	SIO1 start flag
SIO1HIZ	FLG	0.9CH.2	R/W	$P2D_1/SO_1$ pin selection flag
SIO1CK1	FLG	0.9CH.1	R/W	Bit 1 of SIO1 clock selection flag
SIO1CK0	FLG	0.9CH.0	R/W	Bit 0 of SIO1 clock selection flag
WTM8HZ	FLG	0.9DH.0	R	Watch timer 8 Hz carry detection flag
WTM128HZ	FLG	0.9EH.0	R	Watch timer 128 Hz carry detection flag

Symbol	Attribute	Value	Read/ write	Description
IEGGRP1	FLG	0.9FH.2	R/W	Interrupt group 1 edge detection selection flag
IEG0	FLG	0.9FH.1	R/W	INT_0 pin interrupt edge detection selection flag
IEGNC	FLG	0.9FH.0	R/W	INT_{NC} pin interrupt edge detection selection flag
RLSEN	FLG	0.0A0H.0	R/W	Clock stop release setting flag with $P1B_2$ pin
ADCCH2	FLG	0.0A1H.2	R/W	Bit 2 of A/D converter channel selection flag
ADCCH1	FLG	0.0A1H.1	R/W	Bit 1 of A/D converter channel selection flag
ADCCH0	FLG	0.0A1H.0	R/W	Bit 0 of A/D converter channel selection flag
PLLUL	FLG	0.0A2H.0	R	PLL unlock flip-flop flag
ADCEN	FLG	0.0A4H.3	R/W	A/D converter enable flag
ADCCMP	FLG	0.0A4H.0	R/W	A/D converter comparator output
P2DBIO2	FLG	0.0A6H.2	R/W	I/O selection flag of $P2D_2$ pin
P2DBIO1	FLG	0.0A6H.1	R/W	I/O selection flag of $P2D_1$ pin
P2DBIO0	FLG	0.0A6H.0	R/W	I/O selection flag of $P2D_0$ pin
P1CGIO	FLG	0.0A7H.0	R/W	P1C group I/O selection flag
SIO0SF8	FLG	0.0A8H.3	R	SIO0 shift 8 clock flag
SIO0SF9	FLG	0.0A8H.2	R	SIO0 shift 9 clock flag
SBSTT	FLG	0.0A8H.1	R	SIO0 start condition detection flag
SBBSY	FLG	0.0A8H.0	R	SIO0 busy condition detection flag
IRQGRP0	FLG	0.0A9H.0	R/W	Interrupt group 0 (TM0OVF signal) interrupt request flag
IRQSIO1	FLG	0.0AAH.0	R/W	SIO1 interrupt request flag
IRQSIO0	FLG	0.0ABH.0	R/W	SIO0 interrupt request flag
INTGRP1	FLG	0.0ACH.3	R	Interrupt group 1 (H_{SYNC} or V_{SYNC} signal) interrupt status flag
IRQGRP1	FLG	0.0ACH.0	R/W	Interrupt group 1 (H_{SYNC} or V_{SYNC} signal) interrupt request flag
IPGRP0	FLG	0.0ADH.1	R/W	Interrupt group 0 (TM0OVF signal) interrupt enable flag
IPSIO1	FLG	0.0ADH.0	R/W	SIO1 interrupt enable flag
IPSIO0	FLG	0.0AEH.3	R/W	SIO0 interrupt enable flag
IPGRP1	FLG	0.0AEH.2	R/W	Interrupt group 1 (H_{SYNC} or V_{SYNC} signal) interrupt enable flag
IPIDCVP	FLG	0.0AEH.1	R/W	IDC VRAM pointer interrupt enable flag
IPBTM2	FLG	0.0AEH.0	R/W	Basic timer 2 interrupt enable flag
IPTM1	FLG	0.0AFH.3	R/W	Timer 1 interrupt enable flag
IPTM0	FLG	0.0AFH.2	R/W	Timer 0 interrupt enable flag
IP0	FLG	0.0AFH.1	R/W	INT_0 pin interrupt enable flag
IPNC	FLG	0.0AFH.0	R/W	INT_{NC} pin interrupt enable flag

Symbol	Attribute	Value	Read/ write	Description
IDCBKEN	FLG	0.0B0H.3	R/W	IDC background color specification enable flag
IDCBKR	FLG	0.0B0H.2	R/W	Bit 2 of IDC background color specification flag
IDCBKG	FLG	0.0B0H.1	R/W	Bit 1 of IDC background color specification flag
IDCBKB	FLG	0.0B0H.0	R/W	Bit 0 of IDC background color specification flag
IDCEN	FLG	0.0B1H.0	R/W	IDC enable flag
PLULSEN1	FLG	0.0B2H.1	R/W	Bit 1 of PLL unlock flip-flop sensibility selection flag
PLULSEN0	FLG	0.0B2H.0	R/W	Bit 0 of PLL unlock flip-flop sensibility selection flag
VRAMSEL	FLG	0.0B3H.3	R/W	VRAM selection flag
IDCISEL	FLG	0.0B3H.2	R/W	I pin selection flag
IDCD14SL	FLG	0.0B3H.1	R/W	Character dot (vertical) selection flag
IDCCPCH	FLG	0.0B3H.0	R/W	Selection flag for space between displayed characters
P1B2EDET	FLG	0.0B4H.0	R	$P1B_2$ pin edge detection flag
P1BBIO3	FLG	0.0B5H.3	R/W	I/O selection flag of $P1B_3$ pin
P1BBIO2	FLG	0.0B5H.2	R/W	I/O selection flag of $P1B_2$ pin
P1BBIO1	FLG	0.0B5H.1	R/W	I/O selection flag of $P1B_1$ pin
P1BBIO0	FLG	0.0B5H.0	R/W	I/O selection flag of $P1B_0$ pin
P0BBIO3	FLG	0.0B6H.3	R/W	I/O selection flag of $P0B_3$ pin
P0BBIO2	FLG	0.0B6H.2	R/W	I/O selection flag of $P0B_2$ pin
P0BBIO1	FLG	0.0B6H.1	R/W	I/O selection flag of $P0B_1$ pin
P0BBIO0	FLG	0.0B6H.0	R/W	I/O selection flag of $P0B_0$ pin
P0ABIO3	FLG	0.0B7H.3	R/W	I/O selection flag of $P0A_3$ pin
P0ABIO2	FLG	0.0B7H.2	R/W	I/O selection flag of $P0A_2$ pin
P0ABIO1	FLG	0.0B7H.1	R/W	I/O selection flag of $P0A_1$ pin
P0ABIO0	FLG	0.0B7H.0	R/W	I/O selection flag of $P0A_0$ pin
SIO0IMD1	FLG	0.0B8H.1	R/W	Bit 1 of SIO0 interrupt source register
SIO0IMD0	FLG	0.0B8H.0	R/W	Bit 0 of SIO0 interrupt source register
SIO0CK1	FLG	0.0B9H.1	R/W	Bit 1 of SIO0 shift clock frequency selection flag
SIO0CK0	FLG	0.0B9H.0	R/W	Bit 0 of SIO0 shift clock frequency selection flag
IRQIDCVP	FLG	0.0BAH.0	R/W	IDC VRAM pointer interrupt request flag
IRQBTM2	FLG	0.0BBH.0	R/W	Basic timer 2 interrupt request flag
IRQTM1	FLG	0.0BCH.0	R/W	Timer 1 interrupt request flag
IRQTM0	FLG	0.0BDH.0	R/W	Timer 0 interrupt request flag
INT0	FLG	0.0BEH.3	R	INT_0 pin interrupt status flag
IRQ0	FLG	0.0BEH.0	R/W	INT_0 pin interrupt request flag
INTNC	FLG	0.0BFH.3	R	INT_{NC} pin interrupt status flag
IRQNC	FLG	0.0BFH.0	R/W	INT_{NC} pin interrupt request flag

22.6 PERIPHERAL HARDWARE REGISTER

Symbol	Attribute	Value	Read/ write	Description
IDCORG	DAT	01H	R/W	IDC start position setting register
ADCR	DAT	02H	R/W	A/D-converter reference-voltage (V_{REF}) setting register
SIO0SFR	DAT	03H	R/W	SIO0 shift register
HSC	DAT	04H	R	H_{SYNC} counter
TM1M	DAT	05H	R/W	Timer 1 modulo register
TM1C	DAT	06H	R	Timer 1 counter
SIO1SFR	DAT	07H	R/W	SIO1 shift register
PWMR0	DAT	0CH	W	PWM data register 0
PWMR1	DAT	0DH	W	PWM data register 1
PWMR2	DAT	0EH	W	PWM data register 2
PWMR3	DAT	0FH	W	PWM data register 3
PWMR4	DAT	10H	W	PWM data register 4
PWMR5	DAT	11H	W	PWM data register 5
PWMR6	DAT	12H	W	PWM data register 6
PWMR7	DAT	13H	W	PWM data register 7
PWMR8	DAT	14H	W	PWM data register 8
WTMSEC	DAT	1AH	R/W	Register setting seconds of watch timer
WTMMIN	DAT	1BH	R/W	Register setting minutes of watch timer
WTMHR	DAT	1CH	R/W	Register setting hours of watch timer
WTMDAY	DAT	1DH	R/W	Register setting days of watch timer
AR	DAT	40H	R/W	Address register for GET/PUT/PUSH/CALL/BR/MOVT/MOVTH/MOVTL instructions
PLLR	DAT	41H	R/W	PLL data register
IDCVP	DAT	42H	R	IDC VRAM pointer
IDCVPR	DAT	43H	R/W	IDC VRAM pointer reference data register
TM0M	DAT	46H	R/W	Timer 0 modulo register
TM0C	DAT	47H	R	Timer 0 counter

22.7 OTHERS

Symbol	Attribute	Value	Description
DBF	DAT	0FH	Fixed operand value for a PUT/GET/MOVT instruction
IX	DAT	01H	Fixed operand value for an INC instruction

23. ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (T_a = 25 °C)

Parameter	Symbol	Conditions	Rated value	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 P1A, P2B, and P2C	–0.3 to V_{DD} + 0.3	V
Output high current	I_{OH}	One pin	–12	mA
		All pins	–20	mA
Output low current	I_{OL1}	One pin (except P1A)	12	mA
		All pins (except P1A)	20	mA
Output low current	I_{OL2}	One pin (P1A only)	17	mA
		All pins (P1A only)	60	mA
Output withstand voltage	V_{BDS}	P1A, P2A, P2B, P2C	13	V
Operating temperature	T_{opt}		–40 to +85	°C
Storage temperature	T_{stg}		–55 to +125	°C

Caution Absolute maximum ratings are rated values beyond which some physical damages may be caused to the product; if any of the parameters in the table above exceeds its rated value even for a moment, the quality of the product may deteriorate. Be sure to use the product within the rated values.

RECOMMENDED OPERATION RANGE (T_a = –40 to +85 °C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Supply voltage	V_{DD1}		4.5	5.0	5.5	V
	V_{DD2}	When only the CPU is operating	3.5	5.0	5.5	V
	V_{DD3}	When only the watch timer is operating (CPU is stopped)	2.2	5.0	5.5	V
Data hold voltage	V_{DDR}	When clock is stopped (T_a = 25 °C)	2.2		5.5	V
Output withstand voltage	V_{BDS}	P1A, P2A, P2B, P2C			12.5	V
Supply voltage rise time	t_{rise}	$V_{DD} = 0 \rightarrow 4.5$ V			500	ms
Input amplitude	V_{IN}	VCO	0.7		V_{DD}	V_{P-P}

DC CHARACTERISTICS (T_a = –40 to +85 °C, V_{DD} = 5 V ±10 %)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Supply current	I_{DD1}	When all functions are operating, V_{DD} = 5 V, T_a = 25 °C, f_{IN} = 20 MHz, V_{IN} = 0.7 V_{P-P}, When IDC is operating, OSC_{IN} = 10 MHz, When a sine signal is input to the X_{IN} pin, (f_{IN} = 8 MHz, V_{IN} = V_{DD})		20	23	mA
	I_{DD2}	When the CPU and PLL are operating, V_{DD} = 5 V, T_a = 25 °C, f_{IN} = 20 MHz V_{IN} = 0.7 V_{P-P}, When a sine signal is input to the X_{IN} pin, (f_{IN} = 8 MHz, V_{IN} = V_{DD})		9.0	12	mA
	I_{DD3}	When only the CPU is operating, V_{DD} = 5 V, T_a = 25 °C, When a sine signal is input to the X_{IN} pin, (f_{IN} = 8 MHz, V_{IN} = V_{DD})		7.5	9	mA
	I_{DD4}	When the HALT instruction is executed, V_{DD} = 5 V, T_a = 25 °C, When a sine signal is input to the X_{IN} pin, (f_{IN} = 8 MHz, V_{IN} = V_{DD})		2.5	3	mA
Data hold current	I_{DDR1}	When the main clock is stopped and the watch timer is operating V_{DD} = 5 V, T_a = 25 °C		4	15	μA
	I_{DDR2}	When the main clock and watch timer are stopped V_{DD} = 2.5 V, T_a = 25 °C		3	15	μA
Input high voltage	V_{IH1}	P0A, P0B, P1B, P1C, P2D	0.7V_{DD}			V
	V_{IH2}	CE, INT_0, INT_{NC}, $\overline{V_{SYNC}}$, $\overline{H_{SYNC}}$	0.8V_{DD}			V
	V_{IH3}	P0D	0.7V_{DD}			V
Input low voltage	V_{IL1}	P0A, P0B, P0D, P1B, P1C, P2D			0.2V_{DD}	V
	V_{IL2}	CE, INT_0, INT_{NC}, $\overline{V_{SYNC}}$, $\overline{H_{SYNC}}$			0.2V_{DD}	V
Output high current	I_{OH}	$P0A_2$, $P0A_3$, P0B, P0C, P1B, P1C, BLANK, RED, GREEN, BLUE, P1D, P2D, EO V_{OH} = V_{DD} – 1 V	–1	–5		mA
Output low current	I_{OL1}	P0A, P0B, P0C, P1B, P1C, BLANK, RED, GREEN, BLUE, P1D, P2A, P2B, P2C, P2D, EO, PWM V_{OL} = 1 V	1	8.5		mA
	I_{OL2}	P1A V_{OL} = 1 V	15	33		mA
Input high current	I_{IH}	VCO V_{IH} = V_{DD}	0.1	0.85	1.3	mA
Output leakage high current	I_{LOH}	P1A, P2A, P2B, P2C V_O = 12.5 V			0.5	μA
Output off leakage current	I_L	EO V_O = V_{DD} or 0 V		$\pm 10^{-3}$	±1	μA
Built-in pull-down resistor	R_{PD1}	P0D (KEY) V_{IH} = V_{DD}	19	36	69	kΩ
	R_{PD2}	P0D (KEY) V_{IH} = V_{DD} = 5 V	23	36	56	kΩ
	R_{PD3}	P0D (KEY) V_{IH} = V_{DD} = 5 V, T_a = 25 °C	29	36	41	kΩ

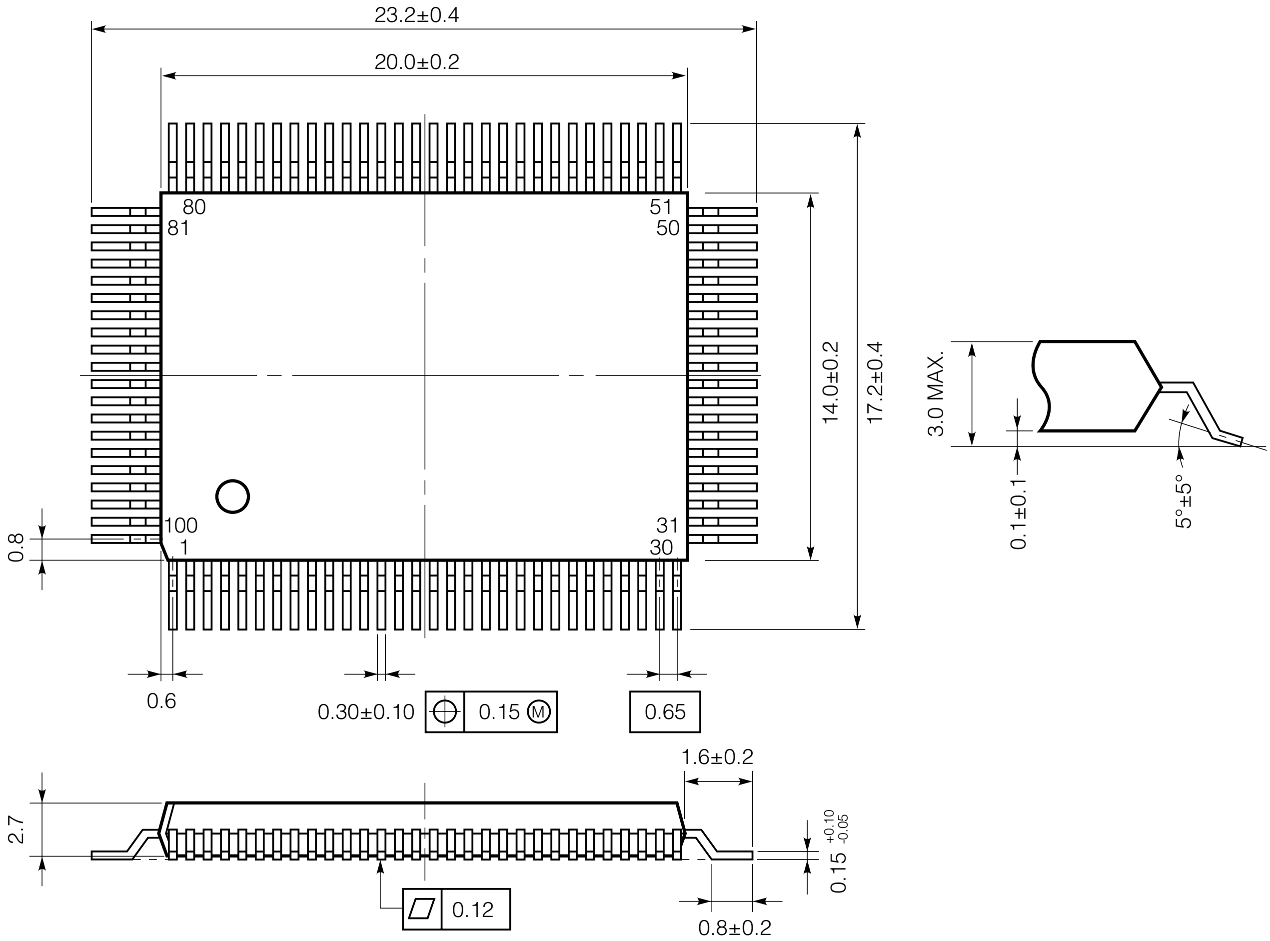
AC CHARACTERISTICS (T_a = –40 to +85 °C, V_{DD} = 5 V ±10 %)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input frequency 1	f_{VCO}	When a sine signal is input to the VCO pin V_{IN} = 0.7V_{P-P}	0.7		20	MHz
Input frequency 2	f_{TMR}	TMIN ($P1B_3$) 50 % duty cycle	45		65	Hz
Input frequency 3	f_{HS}	HSCNT ($P0B_3$)	10		20	kHz

A/D CONVERTER CHARACTERISTICS (T_a = –10 to +50 °C, V_{DD} = 5 V ±10 %)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
A/D conversion absolute accuracy		ADC_0 to ADC_7		±1	±1.5	LSB
A/D conversion resolution		ADC_0 to ADC_7			6	bit
A/D input impedance		ADC_0 to ADC_7	1			MΩ

24. PACKAGE DRAWINGS



S100GF-65-3BA-1

25. RECOMMENDED SOLDERING CONDITIONS

The conditions listed below shall be met when soldering the μPD17068.

For details of the recommended soldering conditions, refer to our document *SMD Surface Mount Technology Manual* (IEI-1207).

Please consult with our sales offices in case any other soldering process is used, or in case soldering is done under different conditions.

Table 25-1 Soldering Conditions for Surface-Mount Devices

μPD17068GF-×××-3BA: 100-pin plastic QFP (14 × 20 mm)
μPD17068GF-E××-3BA: 100-pin plastic QFP (14 × 20 mm)

Soldering process	Soldering conditions	Symbol
Infrared ray reflow	Peak package's surface temperature: 235 °C Reflow time: 30 seconds or less (at 210 °C or more) Maximum allowable number of reflow processes: 2 Exposure limit Note: 7 days (20 hours of pre-baking is required at 125 °C afterward.) <Cautions> (1) Do not start reflow-soldering the device if its temperature is higher than the room temperature because of a previous reflow soldering. (2) Do not use water for flux cleaning before a second reflow soldering.	IR35-207-2
VPS	Peak package's surface temperature: 215 °C Reflow time: 40 seconds or less (at 200 °C or more) Maximum allowable number of reflow processes: 2 Exposure limit Note: 7 days (20 hours of pre-baking is required at 125 °C afterward.) <Cautions> (1) Do not start reflow-soldering the device if its temperature is higher than the room temperature because of a previous reflow soldering. (2) Do not use water for flux cleaning before a second reflow soldering.	VP15-207-2
Wave soldering	Solder temperature: 260 °C or less Flow time: 10 seconds or less Number of flow process: 1 Preheating temperature: 120 °C max. (measured on the package surface) Exposure limit Note: 7 days (20 hours of pre-baking is required at 125 °C afterward.)	WS60-207-1
Partial heating method	Terminal temperature: 300 °C or less Heat time: 3 seconds or less (for each side of device)	–

Note Exposure limit before soldering after dry-pack package is opened.
Storage conditions: Temperature of 25 °C and maximum relative humidity at 65% or less

Caution Do not apply more than a single process at once, except for "Partial heating method."

APPENDIX A. NOTES ON CONNECTING A CRYSTAL

When connecting the crystal, run wires in the portion surrounded by dotted lines in Fig. A-1 according to the following rules to avoid effects such as stray capacitance:

- Minimize the wiring.
- Never cause the wires to cross other signal lines or run near a line carrying a large varying current.
- Cause the grounding point of the capacitor of the oscillator circuit to have the same potential as ground. Never connect the capacitor to a ground pattern carrying a large current.
- Never extract a signal from the oscillator.

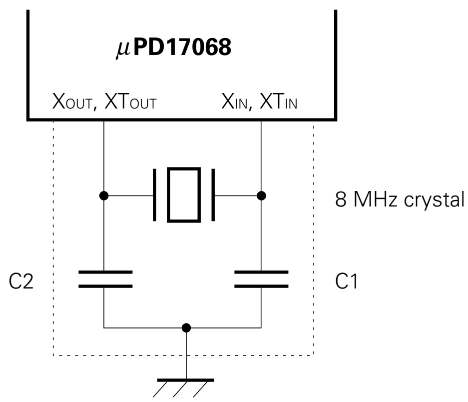
Note the following (1) to (3) when capacitors are connected or the oscillation frequency is adjusted.

(1) When C1 and C2 are too large, the oscillation activation characteristics deteriorate and the supply current increases.

(2) The trimmer capacitor for adjusting the oscillation frequency is usually connected to the X_{IN} (or XT_{IN}) pin. However, this connection may cause deterioration of oscillation stability, depending on the crystal used. (In this case, connect the trimmer capacitor to the X_{OUT} (or XT_{OUT}) pin.) To evaluate the oscillation, use the crystal to be actually used.

(3) Adjust the oscillation frequency while measuring the VCO oscillation frequency. If the probe is connected to pin X_{OUT}, XT_{OUT}, X_{IN}, or XT_{IN}, the oscillation frequency cannot be correctly adjusted because of the probe capacitance.

Fig. A-1 Crystal Connection



APPENDIX B. DEVELOPMENT TOOLS

The following support tools are available for developing programs for the μPD17068.

Hardware

Name	Description
In-circuit emulator IE-17K IE-17K-ET**Note 1** EMU-17K**Note 2**	The IE-17K, IE-17K-ET, and EMU-17K are in-circuit emulators applicable to the 17K series. The IE-17K and IE-17K-ET are connected to the PC-9800 series (host machine) or IBM PC/AT™ through the RS-232-C interface. The EMU-17K is inserted into the extension slot of the PC-9800 series (host machine). Use the system evaluation board (SE board) corresponding to each product together with one of these in-circuit emulators. SIMPLEHOST™, a man machine interface, implements an advanced debug environment. The EMU-17K also enables user to check the contents of the data memory in real time.
SE board (SE-17008)	The SE-17008 is an SE board for the μPD17068, μPD17P068, and μPD17008. It is used solely for evaluating the system. It is also used for debugging in combination with the in-circuit emulator.
Emulation probe (EP-17068GF)	The EP-17068GF is an emulation probe for the μPD17068 and μPD17P068.
Conversion socket (EV-9200GF-100**Note 3**)	The EV-9200GF-100 is a conversion socket for the 100-pin plastic QFP (14 × 20 mm). It is used to connect the EP-17068GF to the target system.
PROM Programmer AF-9703**Note 4** AF-9704**Note 4** AF-9705**Note 4** AF-9706**Note 4**	The AF-9703, AF-9704, AF-9705, and AF-9706 are PROM writers for the μPD17P068. Use one of these PROM writers with the program adapter, AF-9808L, to program the μPD17P068.
Programmer adapter (AF-9808L**Note 4**)	The AF-9808L is a socket unit for the μPD17P068. It is used with the AF-9703, AF-9704, AF-9705, or AF-9706.

Notes 1. Low-end model, operating on an external power supply

2. The EMU-17K is a product of IC Co., Ltd. Contact IC Co., Ltd. (Tokyo, 03-3447-3793) for details.

3. The EP-17068GF is supplied together with one EV-9200GF-100. A set of five EV-9200GF-100s is also available.

4. The AF-9703, AF-9704, AF-9705, AF-9706, and AF-9808L are products of Ando Electric Co., Ltd. Contact Ando Electric Co., Ltd. (Tokyo, 03-3733-1151) for details.

Software

<table>
<tr><th>Name</th><th>Description</th><th>Host machine</th><th colspan="2">OS</th><th>Distribution media</th><th>Part number</th></tr>
<tr><td rowspan="4">17K series assembler (AS17K)</td><td rowspan="4">AS17K is an assembler applicable to the 17K series. In developing μPD17068 programs, AS17K is used in combination with a device file (AS17068).</td><td rowspan="2">PC-9800 series</td><td colspan="2" rowspan="2">MS-DOS™</td><td>5.25-inch, 2HD</td><td>μS5A10AS17K</td></tr>
<tr><td>3.5-inch, 2HD</td><td>μS5A13AS17K</td></tr>
<tr><td rowspan="2">IBM PC/AT</td><td colspan="2" rowspan="2">PC DOS™</td><td>5.25-inch, 2HC</td><td>μS7B10AS17K</td></tr>
<tr><td>3.5-inch, 2HC</td><td>μS7B13AS17K</td></tr>
<tr><td rowspan="4">Device file (AS17068)</td><td rowspan="4">AS17068 is a device file for the μPD17068 and μPD17P068 . It is used together with the assembler (AS17K), which is applicable to the 17K series.</td><td rowspan="2">PC-9800 series</td><td colspan="2" rowspan="2">MS-DOS</td><td>5.25-inch, 2HD</td><td>μS5A10AS17068</td></tr>
<tr><td>3.5-inch, 2HD</td><td>μS5A13AS17068</td></tr>
<tr><td rowspan="2">IBM PC/AT</td><td colspan="2" rowspan="2">PC DOS</td><td>5.25-inch, 2HC</td><td>μS7B10AS17068</td></tr>
<tr><td>3.5-inch, 2HC</td><td>μS7B13AS17068</td></tr>
<tr><td rowspan="4">Support software (SIMPLEHOST)</td><td rowspan="4">SIMPLEHOST, running on the Windows™, provides man-machine-interface in developing programs by using a personal computer and the in-circuit emulator.</td><td rowspan="2">PC-9800 series</td><td rowspan="2">MS-DOS</td><td rowspan="4">Windows</td><td>5.25-inch, 2HD</td><td>μS5A10IE17K</td></tr>
<tr><td>3.5-inch, 2HD</td><td>μS5A13IE17K</td></tr>
<tr><td rowspan="2">IBM PC/AT</td><td rowspan="2">PC DOS</td><td>5.25-inch, 2HC</td><td>μS7B10IE17K</td></tr>
<tr><td>3.5-inch, 2HC</td><td>μS7B13IE17K</td></tr>
</table>

Remark The following table lists the versions of the operating systems described in the above table.

OS	Versions
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 MS-DOS versions 5.00 and 5.00A and PC DOS Ver. 5.0 are provided with a task swap function. This function, however, cannot be used in these software packages.

Cautions on CMOS Devices

① **Countermeasures against static electricity for all MOSs**

Caution **When handling MOS devices, take care so that they are not electrostatically charged.**
Strong static electricity may cause dielectric breakdown in gates. When transporting or storing MOS devices, use conductive trays, magazine cases, shock absorbers, or metal cases that NEC uses for packaging and shipping. Be sure to ground MOS devices during assembling. Do not allow MOS devices to stand on plastic plates or do not touch pins. Also handle boards on which MOS devices are mounted in the same way.

② **CMOS-specific handling of unused input pins**

Caution **Hold CMOS devices at a fixed input level.**
Unlike bipolar or NMOS devices, if a CMOS device is operated with no input, an intermediate-level input may be caused by noise. This allows current to flow in the CMOS device, resulting in a malfunction. Use a pull-up or pull-down resistor to hold a fixed input level. Since unused pins may function as output pins at unexpected times, each unused pin should be separately connected to the V_{DD} or GND pin through a resistor.
If handling of unused pins is documented, follow the instructions in the document.

③ **Statuses of all MOS devices at initialization**

Caution **The initial status of a MOS device is unpredictable when power is turned on.**
Since characteristics of a MOS device are determined by the amount of ions implanted in molecules, the initial status cannot be determined in the manufacture process. NEC has no responsibility for the output statuses of pins, input and output settings, and the contents of registers at power on. However, NEC assures operation after reset and items for mode setting if they are defined.
When you turn on a device having a reset function, be sure to reset the device first.

Caution This product contains an I^2C bus interface circuit.
When using the I^2C bus interface, notify its use to NEC when ordering custom code. NEC can guarantee the following only when the customer informs NEC of the use of the interface:
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.

[MEMO]

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.
NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.
The devices listed in this document are not suitable for use in aerospace equipment, submarine cables, nuclear reactor control systems and life support systems. If customers intend to use NEC devices for above applications or they intend to use "Standard" quality grade NEC devices for applications not intended by NEC, please contact our sales people in advance.
Application examples recommended by NEC Corporation

Standard: Computer, Office equipment, Communication equipment, Test and Measurement equipment, Machine tools, Industrial robots, Audio and Visual equipment, Other consumer products, etc.

Special: Automotive and Transportation equipment, Traffic control systems, Antidisaster systems, Anticrime systems, etc.

M4 92.6

SIMPLEHOST 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.