DATA SHEET

NEC

MOS INTEGRATED CIRCUIT

μPD17072,17073

4-BIT SINGLE-CHIP MICROCONTROLLER WITH HARDWARE FOR DIGITAL TUNING SYSTEM

DESCRIPTION

μPD17072 and 17073 are low-voltage 4-bit single-chip CMOS microcontrollers containing hardware ideal for organizing a digital tuning system.

The CPU employs 17K architecture and can manipulate the data memory directly, perform arithmetic operations, and control peripheral hardware with a single instruction. All the instructions are 16-bit one-word instructions.

As peripheral hardware, a prescaler that can operate at up to 230 MHz for a digital tuning system, a PLL frequency synthesizer, and an intermediate frequency (IF) counter are integrated in addition to I/O ports, an LCD controller/driver, A/D converter, and BEEP.

Therefore, a high-performance, multi-function digital tuning system can be configured with a single chip of μPD17072 or 17073.

Because the μPD17072 and 17073 can operate at low voltage (V_{DD} = 1.8 to 3.6 V), they are ideal for controlling battery-cell driven portable devices such as portable radio equipment, headphone stereos, or radio cassette recorders.

FEATURES

- 17K architecture: general-purpose register system
- Program memory (ROM)
  - 6 KB (3072 × 16 bits): μPD17072
  - 8 KB (4096 × 16 bits): μPD17073
- General-purpose data memory (RAM)
  - 176 × 4 bits
- Instruction execution time
  - 53.3 μs (with 75-kHz crystal resonator: normal operation)
  - 106.6 μs (with 75-kHz crystal resonator: low-speed mode)
- Decimal operation
- Table reference
- Hardware for PLL frequency synthesizer
  Dual modulus prescaler (230 MHz max.), programmable divider, phase comparator, charge pump
- Various peripheral hardware
  General-purpose I/O ports, LCD controller/driver, serial interface, A/D converter, BEEP, intermediate frequency (IF) counter
- Many interrupts
  - External: 1 channel
  - Internal: 2 channels
- Power-ON reset, CE reset, and power failure detector
- CMOS low power consumption
- Supply voltage: V_{DD} = 1.8 to 3.6 V

Unless otherwise stated, the μPD17073 is taken as a representative product in this document.

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

Document No. U11450EJ1V0DS00 (1st edition)
Date Published September 1996 P
Printed in Japan

ORDERING INFORMATION

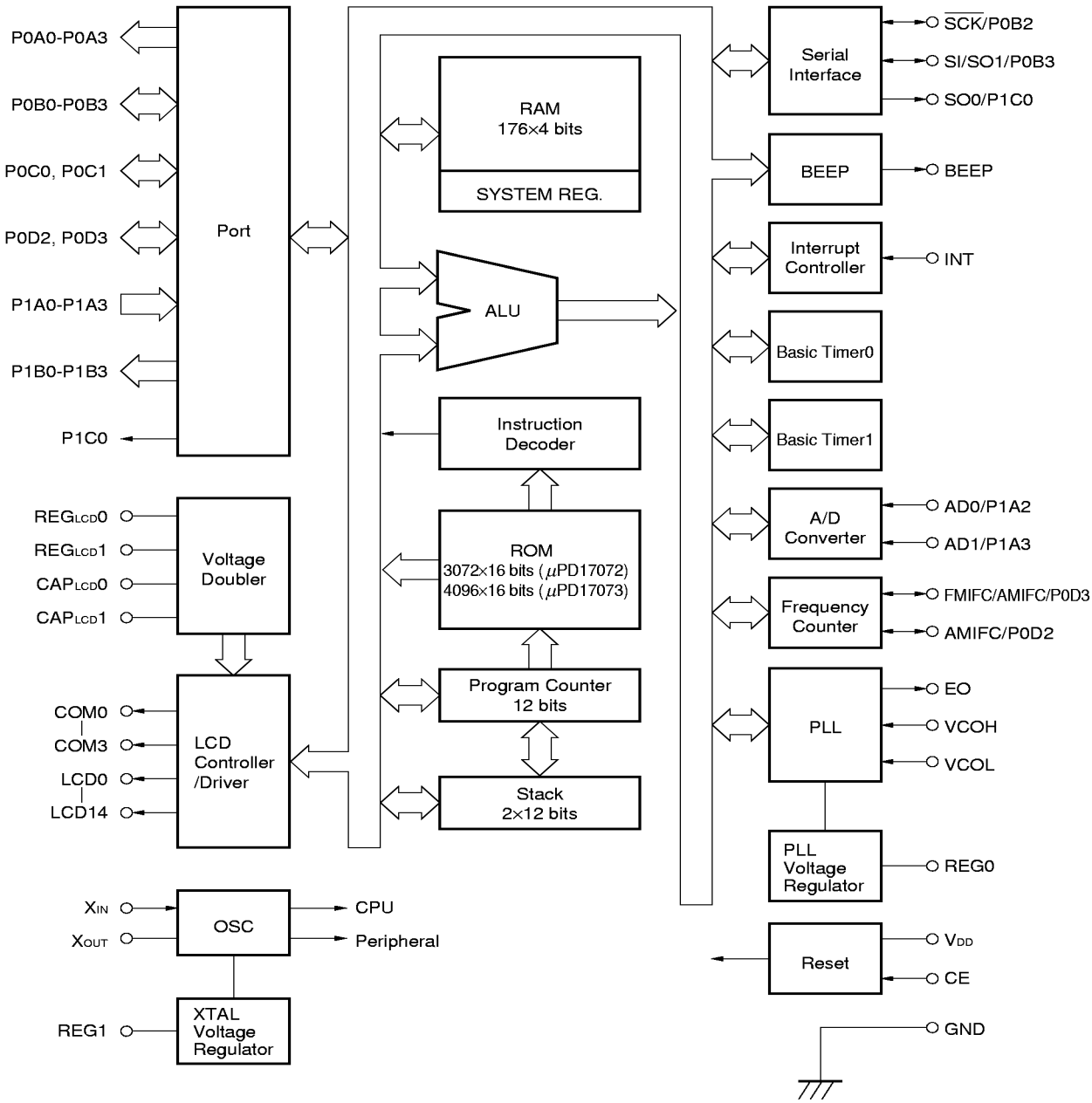
Part Number	Package
μPD17072GB-×××-1A7	56-pin plastic QFP (10 × 10 mm, 0.65-mm pitch)
μPD17072GB-×××-9EU	64-pin plastic TQFP (fine pitch) (10 × 10 mm, 0.5-mm pitch)
μPD17073GB-×××-1A7	56-pin plastic QFP (10 × 10 mm, 0.65-mm pitch)
μPD17073GB-×××-9EU	64-pin plastic TQFP (fine pitch) (10 × 10 mm, 0.5-mm pitch)

Remark ××× is a ROM code number.

FUNCTION OUTLINE

Item		Function
Program memory (ROM)		• 6K bytes (3072 × 16 bits): μPD17072 • 8K bytes (4096 × 16 bits): μPD17073 • Table reference area: 4096 × 16 bits
General-purpose data memory (RAM)		• 176 × 4 bits General-purpose register: 16 × 4 bits (fixed at 00H through 0FH of BANK0, shared with data buffers.)
LCD segment register		15 × 4 bits
Peripheral control register		32 × 4 bits
Instruction execution time		• 53.3 μs (with 75-kHz crystal resonator: normal operation) • 106.6 μs (with 75-kHz crystal resonator: low-speed mode) Selectable by software
Stack level		• Address stack: 2 levels (stack can be manipulated) • Interrupt stack: 1 level (stack cannot be manipulated)
General-purpose port		• I/O port: 8 • Input port: 4 • Output port: 9
BEEP		• 1 type • Selectable frequency (1.5 kHz, 3 kHz)
LCD controller/driver		• 15 segments, 4 commons 1/4 duty, 1/2 bias, frame frequency of 62.5 Hz, drive voltage V_{LCD1} = 3.1 V (TYP.)
Serial interface		• 1 channel (Serial I/O mode) 3-wire/2-wire mode selectable
A/D converter		4 bits × 2 channels (successive approximation via software)
Interrupt		• 3 channels (maskable interrupt) External interrupt: 1 (INT pin) Internal interrupt: 2 (basic timer 1, serial interface)
Timer		• 2 channels Basic timer 0: 125 ms Basic timer 1: 8 ms, 32 ms
Reset		• Power-ON reset (on power application) • Reset by CE pin (CE pin: low level → high level) • Power failure detection function
PLL frequency synthesizer	Division method	• Direct division method (VCOL pin: 8 MHz MAX.) • Pulse swallow method (VCOL pin: 55 MHz MAX.) (VCOH pin: 230 MHz MAX.)
	Reference frequency	• 6 types selectable by program 1, 3, 5, 6.25, 12.5, 25 kHz
	Charge pump	Error out output: 1 line (EO pin)
	Phase comparator	Unlock detectable by program
Frequency counter		• Frequency measurement P0D3/FMIFC/AMIFC pin: FMIF mode, 10 to 11 MHz P0D3/FMIFC/AMIFC pin: AMIF mode P0D2/AMIFC pin } 400 to 500 kHz
Supply voltage		V_{DD} = 1.8 to 3.6 V
Package		• 56-pin plastic QFP (10 × 10 mm, 0.65-mm pitch) • 64-pin plastic TQFP (10 × 10 mm, 0.5-mm pitch)

BLOCK DIAGRAM

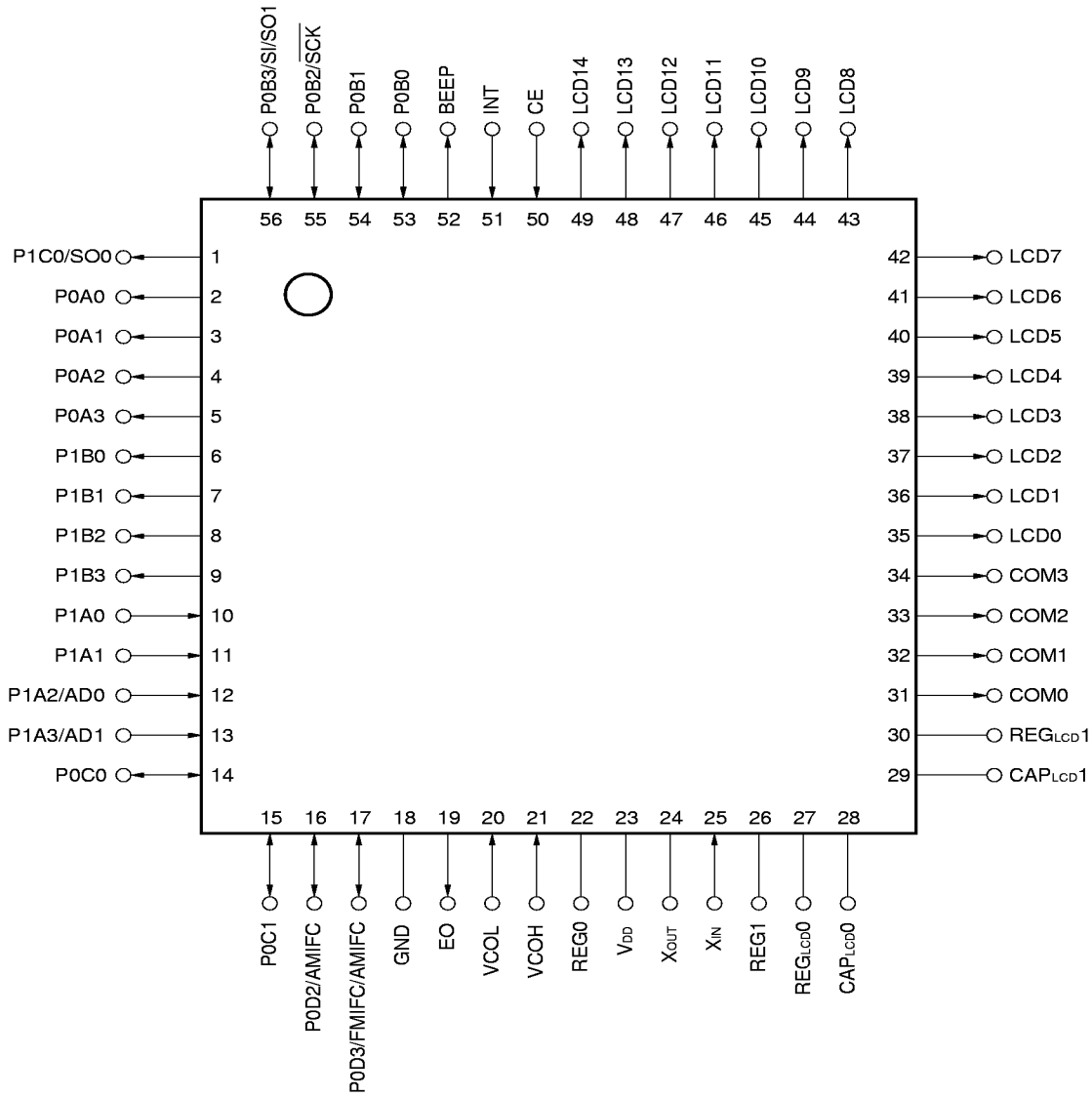


PIN CONFIGURATION (Top View)

56-pin plastic QFP (10 × 10 mm)

μPD17072GB-×××-1A7

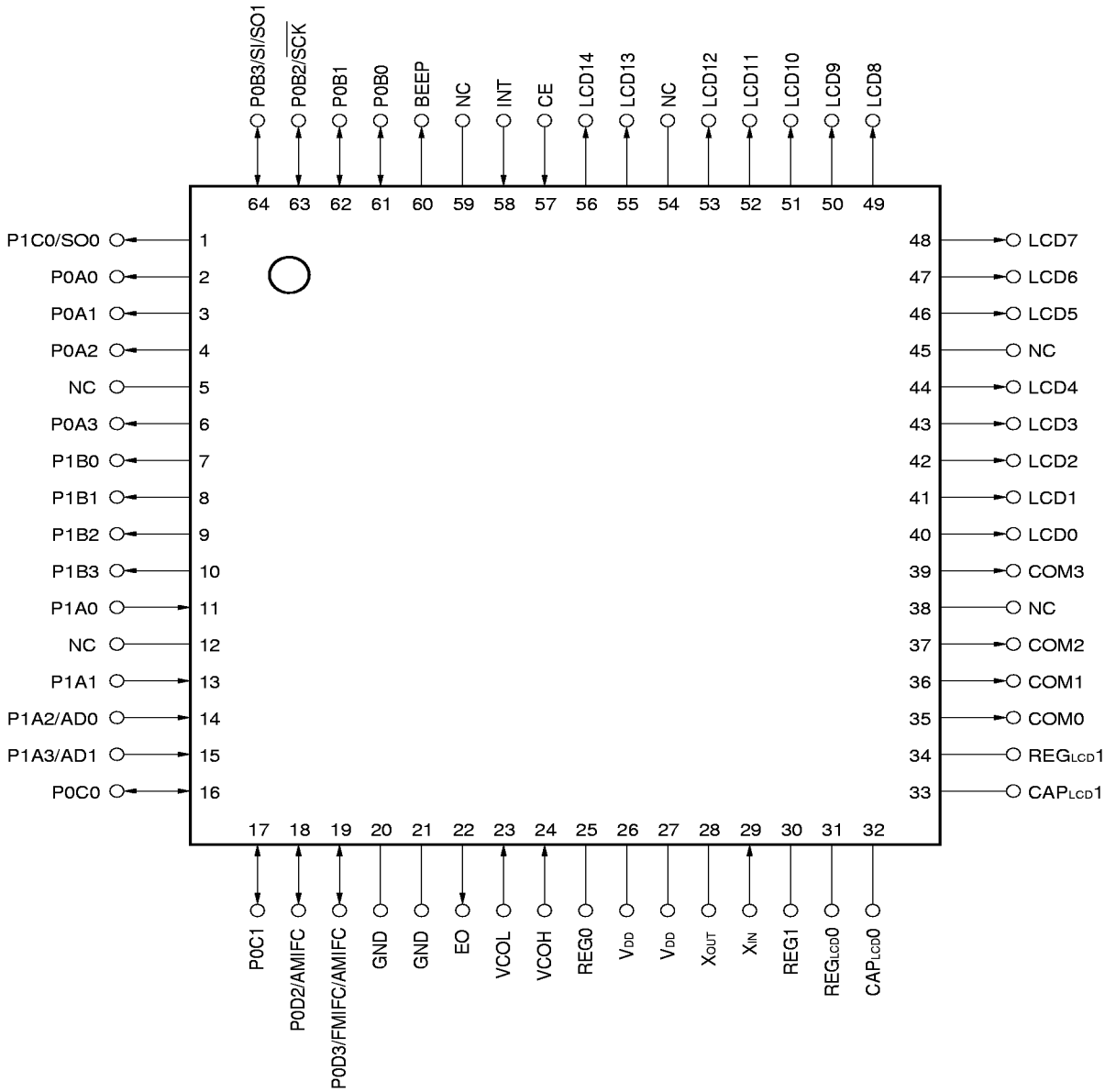
μPD17073GB-×××-1A7



64-pin plastic TQFP (fine pitch) (10 × 10 mm)

μPD17072GB-×××-9EU

μPD17073GB-×××-9EU



PIN IDENTIFICATION

AD0, AD1	:	A/D converter input
AMIFC	:	Intermediate frequency (IF) counter input
BEEP	:	BEEP output
$CAP_{CLD}0$, $CAP_{LCD}1$	:	Capacitor connection for LCD drive voltage
CE	:	Chip enable
COM0-COM2	:	LCD common signal output
EO	:	Error out
FMIFC	:	Intermediate frequency (IF) counter input
GND	:	Ground
INT	:	External interrupt request signal input
LCD0-LCD14	:	LCD segment signal output
NC	:	No connection
P0A0-P0A3	:	Port 0A
P0B0-P0B3	:	Port 0B
P0C0, P0C1	:	Port 0C
P0D2, P0D3	:	Port 0D
P1A0-P1A3	:	Port 1A
P1B0-P1B3	:	Port 1B
P1C0	:	Port 1C
$REG_{LCD}0$, $REG_{LCD}1$	:	LCD drive voltage
REG0	:	PLL voltage regulator
REG1	:	Oscillation circuit voltage regulator
$\overline{SCK}$	:	Serial clock I/O
SI	:	Serial data input
SO0, SO1	:	Serial data output
VCOL	:	Local oscillator input
VCOH	:	Local oscillator input
V_{DD}	:	Positive power supply
X_{IN}, X_{OUT}	:	Crystal resonator connection pins

CONTENTS

1. PIN FUNCTION

1.1 Pin Function List

Pin No. QFP	Pin No. TQFP	Symbol	Function	Output format	At power-ON reset

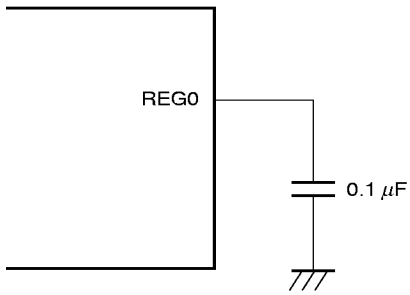
1	1	P1C0/SO0	Port 1C and output of serial interface.
			• P1C0
			  • 1-bit output port
			• SO0
			  • Serial data output
				CMOS push-pull	Low-level output

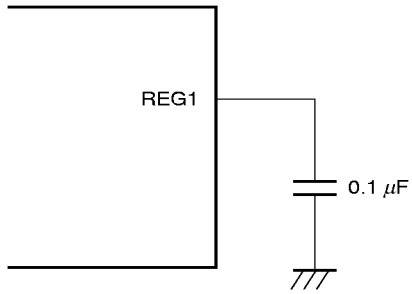
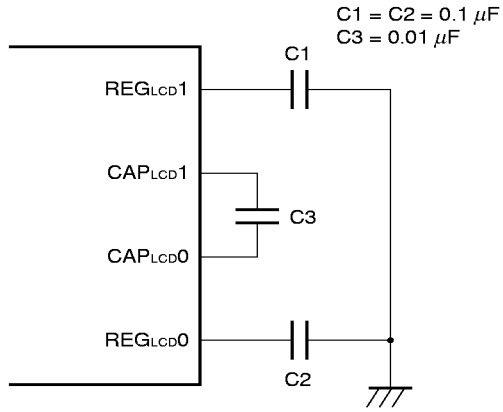
2 3 4 5	2 3 4 6	P0A0 P0A1 P0A2 P0A3	4-bit output port (port 0A).	CMOS push-pull	Low-level output
6 7 8 9	7 8 9 10	P1B0 P1B1 P1B2 P1B3	4-bit output port (port 1B).	CMOS push-pull	Low-level output

10
11
12
13
	11
	13
	14
	15
		P1A0
		P1A1
		P1A2/AD0
		P1A3/AD1
			Port 1A and analog inputs to A/D converter.
			• P1A3-P1A0
			  • 4-bit input port
			• AD1, AD0
			  • Analog inputs to A/D converter
				—	Inputs with pull-down resistor

14 15	16 17	P0C0 P0C1	2-bit I/O port (port 0C). Input/output mode can be set in 1-bit units.	CMOS push-pull	Input

16
17
	18
	19
		P0D2/AMIFC
		P0D3/FMIFC/
		AMIFC
			Port 0D and IF counter inputs.
			• P0D3, P0D2
			  • 2-bit I/O port
			  • Can be set in input/output mode in 1-bit units.
			• FMIFC, AMIFC
			  • IF counter inputs
				CMOS push-pull	Input

18	20 21	GND	Ground	—	—
19	22	EO	Output from charge pump of PLL frequency synthesizer	CMOS 3-state	Floating
20 21	23 24	VCOL VCOH	Input local oscillation frequency of PLL.	—	Floating
22	25	REG0	Output of PLL voltage regulator. Connect this pin to GND via 0.1-μF capacitor. 	—	Low-level output

Pin No. QFP	Pin No. TQFP	Symbol	Function	Output format	At power-ON reset
23	26 27	V_{DD}	Positive power supply. Supply 1.8 to 3.6 V (T_A = –20 to +70 °C) to operate all functions. Do not apply voltage higher than that of V_{DD} pin to any pin other than V_{DD}.	—	—
24	28	X_{OUT}	Pins for connecting crystal resonator for system clock oscillation.	CMOS push-pull	—
25	29	X_{IN}		—	
26	30	REG1	Output of voltage regulator for oscillation circuit. Connect this pin to GND via 0.1-μF capacitor. 	—	—
27 28 29 30	31 32 33 34	$REG_{LCD}0$ $CAP_{LCD}0$ $CAP_{LCD}1$ $REG_{LCD}1$	• $REG_{LCD}1$, $REG_{LCD}0$ LCD drive power pins. • $CAP_{LCD}1$, $CAP_{LCD}0$ Connect capacitors for doubler circuit to generate LCD drive voltage, across these pins. To configure doubler circuit, connect capacitors as shown below.  **Caution The value of the LCD drive voltage differs if the values of C1, C2, and C3 are changed because of the configuration of the doubler circuit.**	—	—

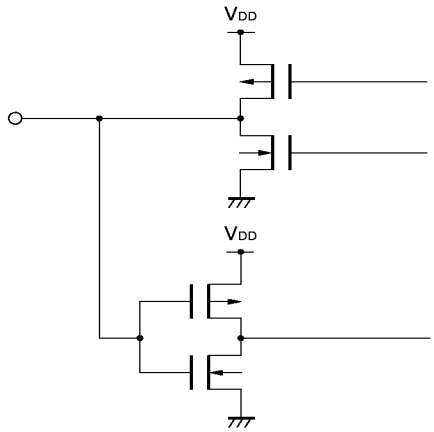
Pin No. QFP	Pin No. TQFP	Symbol	Function	Output format	At power-ON reset
31 32 33 34	35 36 37 39	COM0 COM1 COM2 COM3	Common signal outputs of LCD controller/driver.	CMOS ternary output	Low-level output
35 | 49	40 | 56	LCD0 | LCD14	Segment signal outputs of LCD controller/driver.	CMOS push-pull	Low-level output
50	57	CE	Device operation select and reset signal input.	—	Input
51	58	INT	External interrupt request signal input. Interrupt request is issued at rising or falling edge of signal input to this pin.	—	Input
52	60	BEEP	BEEP signal output pin. BEEP output of 1.5 kHz or 3 kHz can be selected.	CMOS push-pull	Low-level output

53
54
55
56
	61
	62
	63
	64
		P0B0
		P0B1
		P0B2/$\overline{SCK}$
		P0B3/SI/SO1
			Port 0B and serial interface I/O.
			• P0B3-P0B0
			  • 4-bit I/O port
			  • Can be set in input or output mode in 1-bit units.
			• $\overline{SCK}$
			  • Serial clock I/O
			• SO1
			  • Serial data output
			• SI
			  • Serial data input
				CMOS push-pull	Input

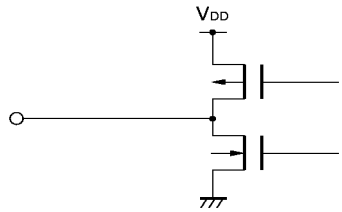
—	5 12 38 45 54 59	NC	No connection	—	—

1.2 Equivalent Circuits of Pins

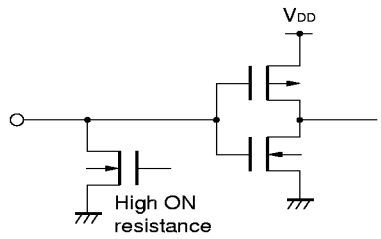
(1) P0B (P0B3/SI/SO1, P0B2/$\overline{SCK}$, P0B1, P0B0)
P0C (P0C1, P0C0)
P0D (P0D3/FMIFC/AMIFC, P0D2/AMIFC) } (I/O)



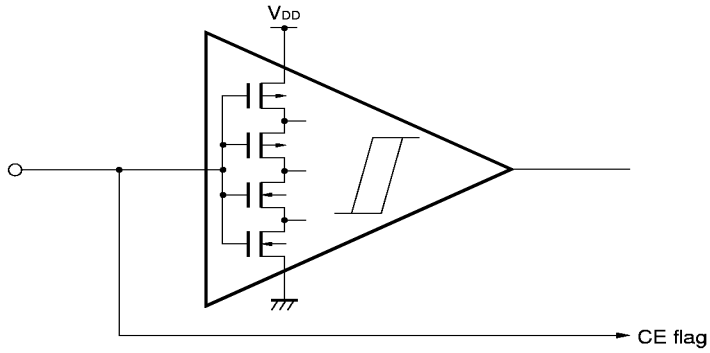
(2) P0A (P0A3, P0A2, P0A1, P0A0)
P1B (P1B3, P1B2, P1B1, P1B0)
P1C (P1C0/SO0)
LCD14-LCD0
BEEP
EO } (Output)



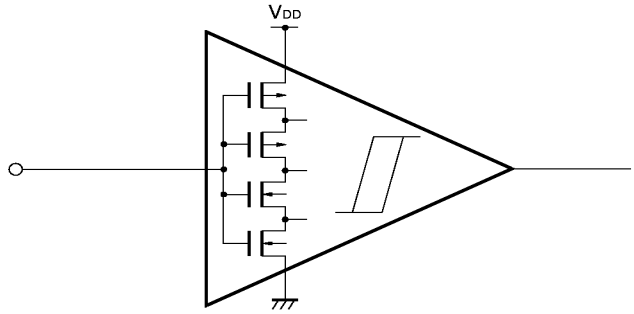
(3) P1A (P1A3/AD1, P1A2/AD0, P1A1, P1A0) (Input)



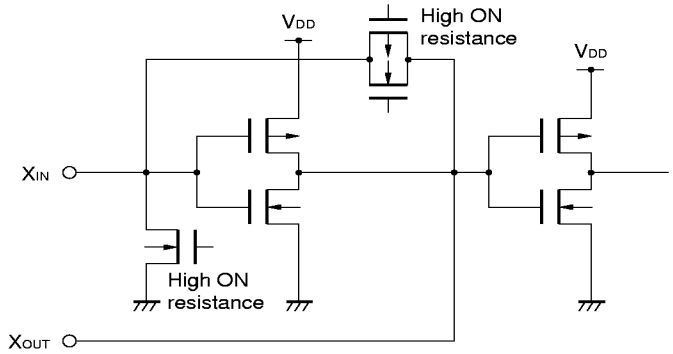
(4) CE (Schmitt trigger input)



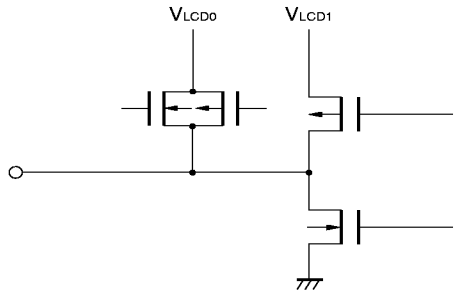
(5) INT (Schmitt trigger input)



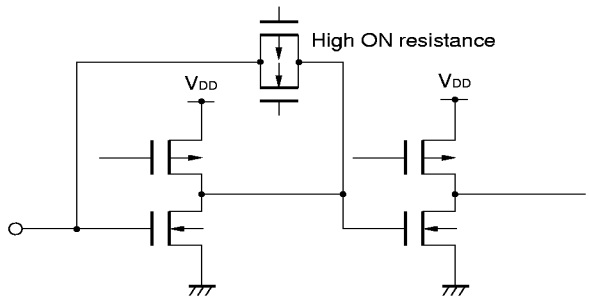
(6) XOUT (output), XIN (input)



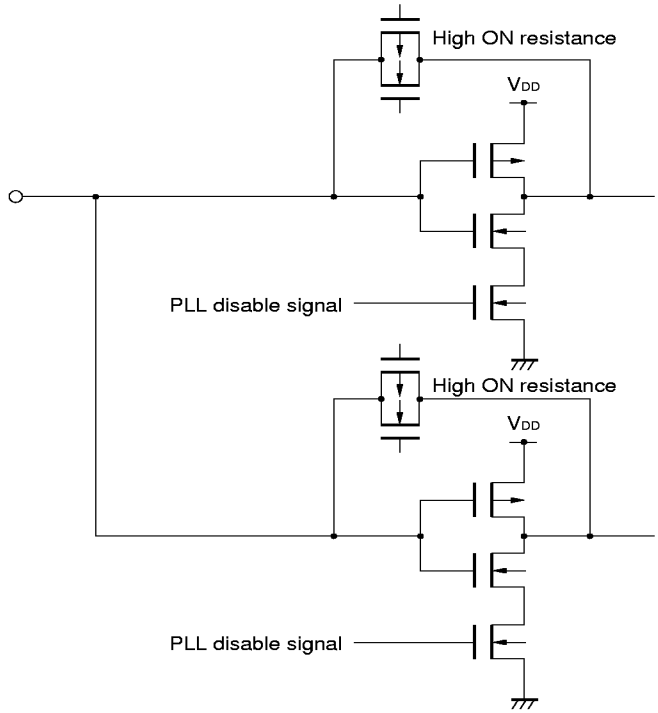
(7) COM3 through COM0 (output)



(8) VCOH (input)



(9) VCOL (input)



1.3 Processing of Unused Pins

It is recommended that the unused pins be connected as follows:

Table 1-1. Processing of Unused Pins

Pin name		I/O mode	Recommended processing of unused pins
Port pin	P0A0-P0A3	CMOS push-pull output	Open
	P0B0, P0B1	I/O[Note 1]	Set by software to output low level and open
	P0B2/$\overline{SCK}$		
	P0B3/SI/SO1		
	P0C0, P0C1		
	P0D2/AMIFC		
	P0D3/FMIFC/AMIFC		
	P1A0, P1A1	Input	Connect each of these pins to V_{DD} or GND via resistor[Note 2].
	P1A2/AD0		
	P1A3/AD1		
	P1B0-P1B3	CMOS push-pull output	Open
	P1C0/SO0		
Pins other than port pins	BEEP	CMOS push-pull output	Open
	CE	Input	Connect to V_{DD} via resistor[Note 2].
	COM0-COM3	Output	Open
	EO	Output	
	INT	Input	Connect to GND via resistor[Note 2].
	LCD0-LCD14	CMOS push-pull output	Open
	VCOH, VCOL	Input	Connect each of these pins to GND via resistor[Note 2].

Notes 1. The I/O ports are set in the input mode on power application, on clock stop, and on CE reset.

2. When pulling up (connecting to V_{DD} via resistor) or pulling down (connecting to GND via resistor) a pin externally with high resistance, the pin almost goes into a high-impedance state, and consequently, the current consumption (through current) of the port increases. Generally, the pull-up or pull-down resistance is several 10 kΩ, though it varies depending on the application circuit.

1.4 Notes on Using CE Pin

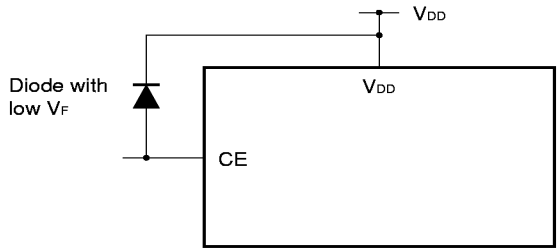
The CE pin has a function to set a test mode in which the internal operations of the μPD17073 are tested (dedicated to IC test), in addition to the functions listed in 1.1 Pin Function List.

When a voltage higher than V_{DD} is applied to the CE pin, the test mode is set. This means that if noise exceeding V_{DD} is applied to the CE pin even during normal operation, the test mode is set, affecting the normal operation.

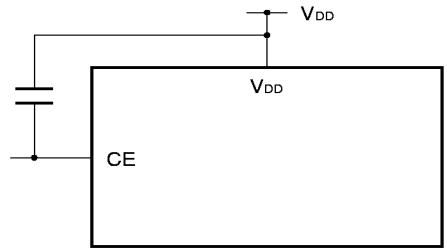
If the wiring of the CE pin is too long, the above problem occurs because wiring noise is superimposed on the CE pin.

Therefore, wire the CE pin with as short a wiring length as possible to suppress noise. If noise cannot be avoided, use external components as shown below to suppress noise.

- Connect a diode with low V_F between CE and V_{DD}



- Connect a capacitor between CE and V_{DD}



2. PROGRAM MEMORY (ROM)

2.1 General

Figure 2-1 shows the configuration of the program memory.

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

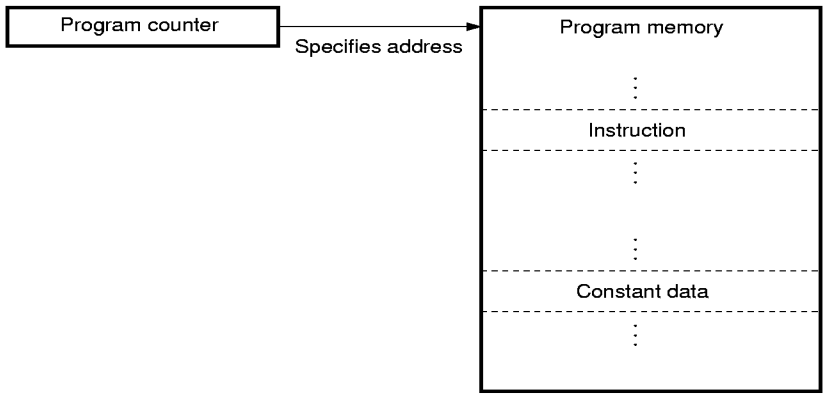
The addresses of the program memory are specified by the program counter.

The program memory has the following two major functions:

(1) Stores program

(2) Stores constant data

Figure 2-1. Outline of Program Memory



2.2 Program Memory

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

As shown in this figure, the program memory is configured as follows:

μPD17072: 3072 × 16 bits (0000H-0BFFH)

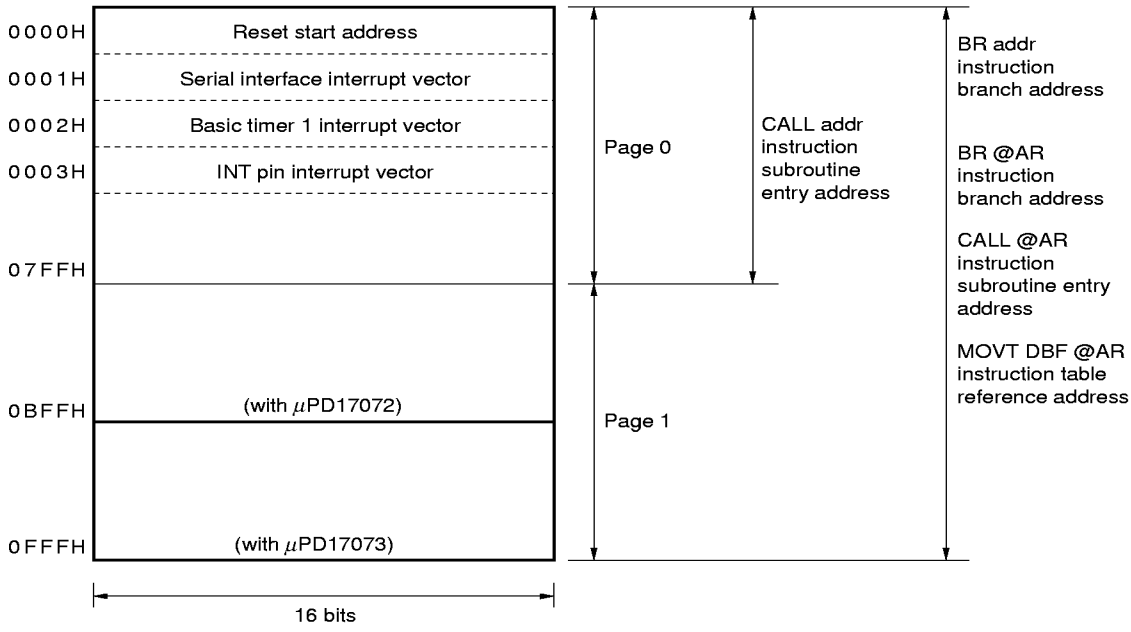
μPD17073: 4096 × 16 bits (0000H-0FFFH)

Therefore, the addresses of the program memory range from 0000H to 0FFFH.

All the "instructions" are "one-word instructions" each of which is 16 bits long. Consequently, one instruction can be stored in one address of the program memory.

As constant data, the contents of the program memory are read to the data buffer by using a table reference instruction.

Figure 2-2. Configuration of Program Memory



Caution With the μPD17072, the range of addresses that can be called by each instruction is 0000H to 0BFFH. The area from addresses 0C00H through 0FFFH is an undefined area.

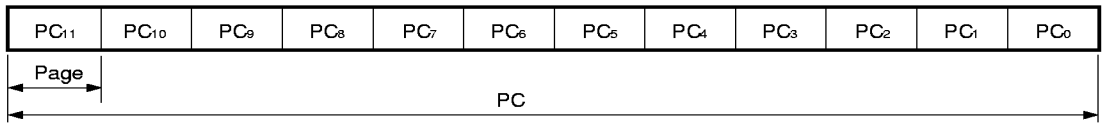
2.3 Program Counter

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

The program counter specifies an address of the program memory.

As shown in this figure, the program counter is a 12-bit binary counter. The most significant bit b_{11} indicates a page.

Figure 2-3. Configuration of Program Counter



2.4 Execution Flow of Program Memory

Execution of the program is controlled by the program counter which specifies an address of the program memory.
Figure 2-4 shows the values to be set to the program counter when each instruction is executed.
Table 2-1 shows the vector addresses that are to be set to the program counter when each interrupt occurs.

Figure 2-4. Specification by Program Counter On Execution of Each Instruction

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

Table 2-1. Interrupt Vector Address

Priority	Internal/external	Interrupt source	Vector address
1	External	INT pin	0003H
2	External	Basic timer 1	0002H
3	External	Serial interface	0001H

2.5 Notes on Using Program Memory

(1) μPD17072

The program memory addresses of the μPD17072 are 0000H through 0BFFH. However, because the addresses that can be specified by the program counter (PC) are 0000H through 0FFFH, keep the following points in mind when specifying a program memory address:

- Be sure to write a branch instruction to address 0BFFH, when writing an instruction to this address.
- Do not write an instruction to addresses 0C00H through 0FFFH.
- Do not branch to addresses 0C00H through 0FFFH.

(2) With μPD17073

The program memory addresses of the μPD17073 are 0000H through 0FFFH. Keep the following point in mind:

- Be sure to write a branch instruction to address 0FFFH, when writing an instruction to this address.

3. ADDRESS STACK (ASK)

3.1 General

Figure 3-1 outlines the address stack.

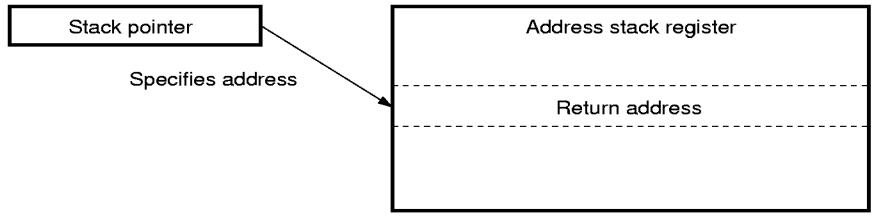
The address stack consists of a stack pointer and an address stack register.

The address of the address stack register is specified by the stack pointer.

The address stack saves return addresses when a subroutine call instruction has been executed and when an interrupt has been accepted.

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

Figure 3-1. Outline of Address Stack



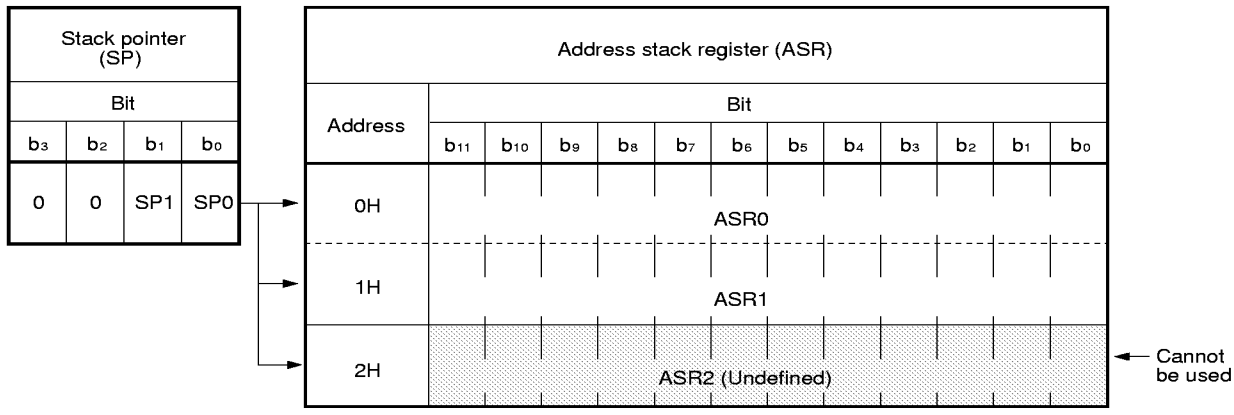
3.2 Address Stack Register (ASR)

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

The address stack register consists of three 12-bit registers ASR0-ASR2. Actually, however, no register is assigned to ASR2, and the address stack register therefore consists of two 12-bit registers (ASR0 and ASR1).

The address stack saves return addresses when a subroutine call instruction has been executed, when an interrupt has been accepted, and when a table reference instruction is executed.

Figure 3-2. Configuration of Address Stack Register



3.3 Stack Pointer (SP)

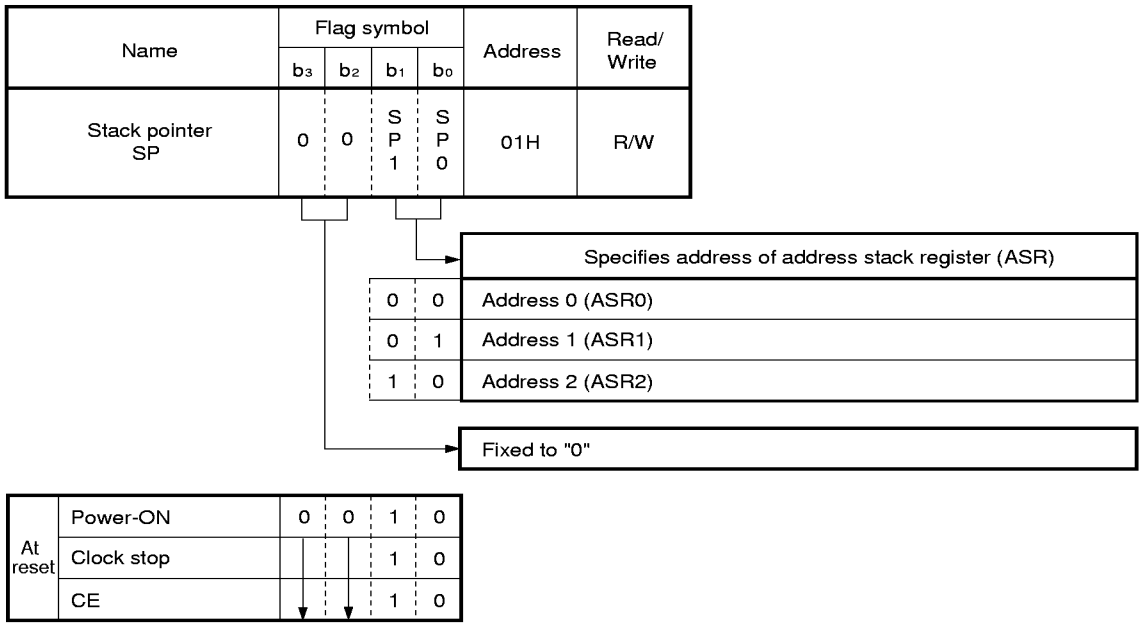
Figure 3-3 shows the configuration and functions of the stack pointer.

The stack pointer is a 4-bit binary counter.

The stack pointer specifies the addresses of the address stack registers.

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

Figure 3-3. Configuration and Functions of Stack Pointer



3.4 Operations of Address Stack

3.4.1 Subroutine call ("CALL addr" or "CALL @AR") and return ("RET" or "RETSK") instructions

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

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

3.4.2 Table reference instruction ("MOVT DBF, @AR")

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

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

3.4.3 On acceptance of interrupt and execution of return instruction ("RETI" instruction)

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

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

3.4.4 Address stack manipulation instructions ("PUSH AR" and "POP AR")

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

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

3.5 Notes on Using Address Stack

The nesting level of the address stack is two, and the value of the address stack register ASR2 is "undefined" when the value of the stack pointer is 2H.

Consequently, if a subroutine is called or an interrupt is used exceeding 2 levels without manipulating the stack, program execution returns to an "undefined" address.

4. DATA MEMORY (RAM)

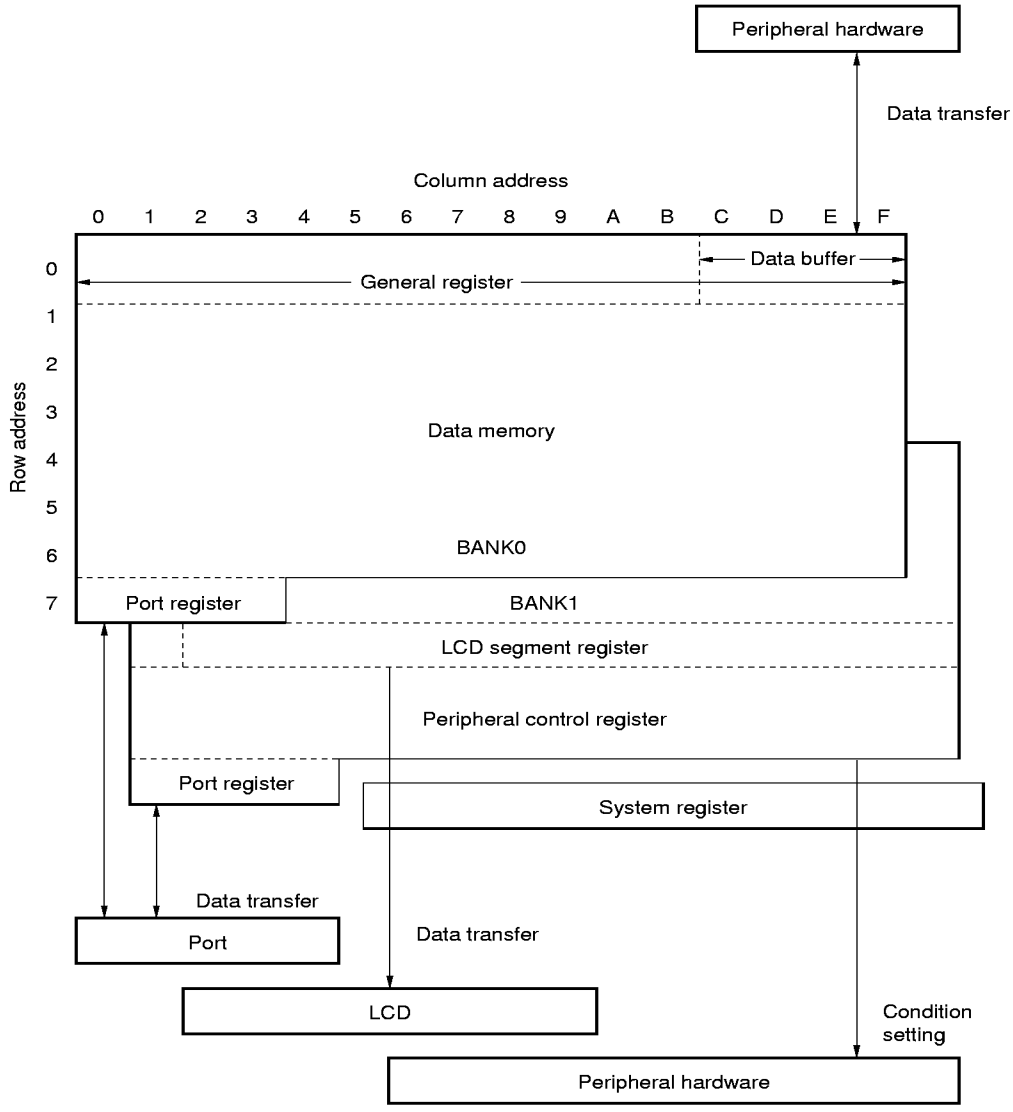
4.1 General

Figure 4-1 outlines the data memory.

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

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

Figure 4-1. Outline of Data Memory



4.2 Configuration and Function of Data Memory

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

As shown in this figure, the data memory is divided into three banks, and each bank consists of 128 nibbles with 7H row addresses and 0FH column addresses.

In terms of function, the data memory can be divided into six blocks each of which is described in the following paragraphs 4.2.1 through 4.2.8.

The contents of the data memory can be operated, compared, judged, and transferred in 4-bit units by data memory manipulation instructions.

Table 4-1 lists the data memory manipulation instructions.

4.2.1 System registers (SYSREG)

The system registers are allocated to addresses 74H through 7FH.

These registers are allocated independently of the bank and directly control the CPU. The same system registers exist at addresses 74H through 7FH of each bank.

With the μPD17073, only AR (address register: addresses 75H through 77H), BANK (bank register: address 79H), and PSWORD (program status word: addresses 7EH and 7FH) can be manipulated.

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

4.2.2 Data buffer (DBF)

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

The data buffer reads the constant data in the program memory (table reference), and transfers data with peripheral hardware.

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

4.2.3 General registers

With the μPD17073, the general registers are fixed at row address 0 of BANK0, i.e., addresses 00H through 0FH, and cannot be moved.

Operations and data transfer between the general registers and data memory can be executed with a single instruction.

The general registers can be controlled by data memory manipulation instructions, like the other data memory areas.

For details, refer to **6. GENERAL REGISTER (GR)**.

4.2.4 LCD segment registers

The LCD segment registers are allocated to addresses 41H through 4FH of BANK1 of the data memory, and are used to set the display data of the LCD controller/driver.

For details, refer to **18. LCD CONTROLLER/DRIVER**.

4.2.5 Port registers

The port registers are allocated to addresses 70H through 73H of BANK0 and addresses 70H through 73H of BANK1, and are used to set the output data of each general-purpose port and read the data of the input ports.

For details, refer to **10. GENERAL-PURPOSE PORT**.

4.2.6 Peripheral control registers

The peripheral control registers are allocated to addresses 50H through 6FH of BANK1 and are used to set the conditions of the peripheral hardware (such as PLL, serial interface, A/D converter, IF counter, and timer).

For details, refer to **8. PERIPHERAL CONTROL REGISTER**.

4.2.7 General-purpose data memory

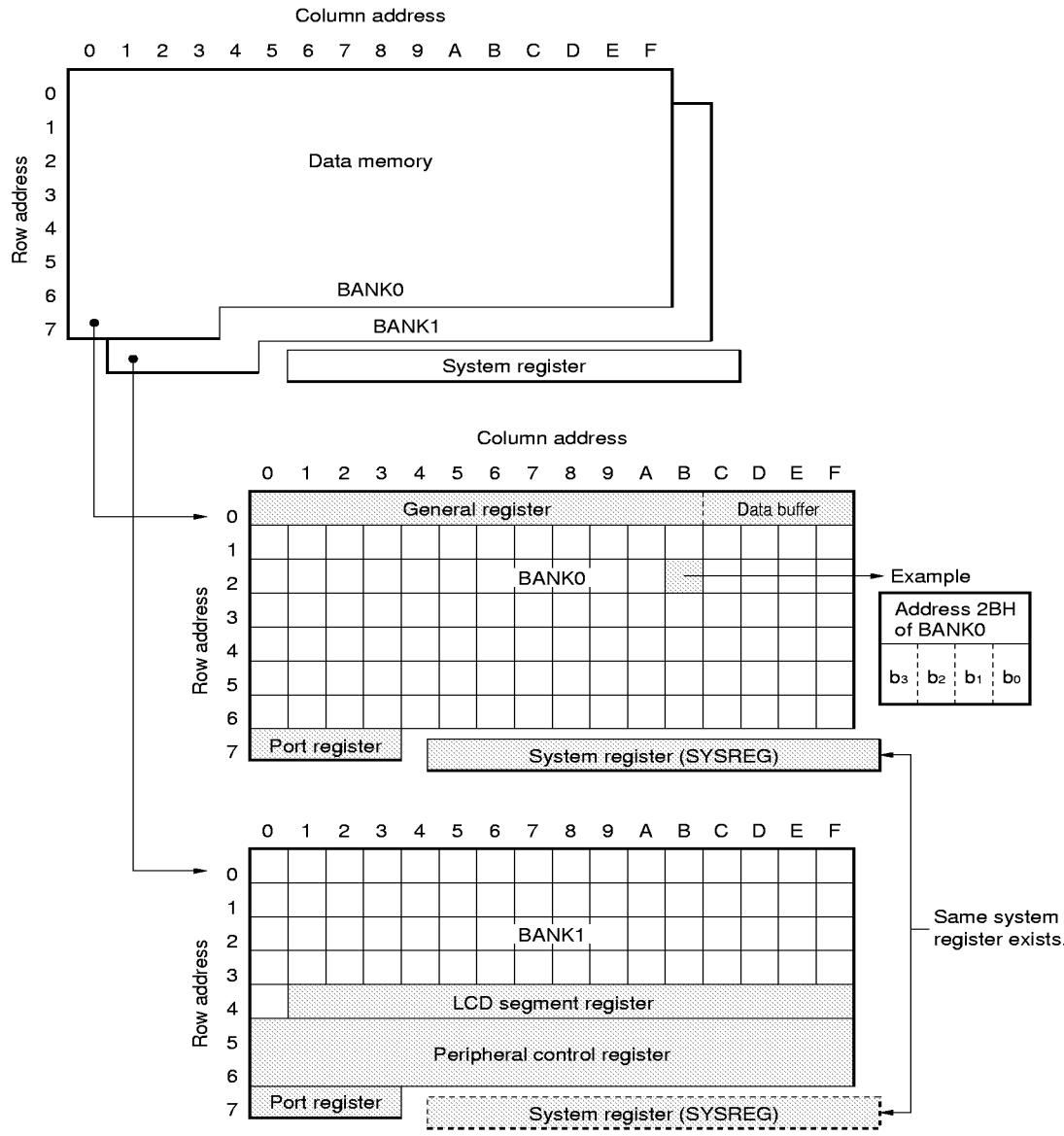
The general-purpose data memory is allocated to the area of the data memory excluding the system register, LCD segment register, port register, and peripheral control register.

With the μPD17073, a total of 176 nibbles (176 × 4 bits), 112 nibbles of BANK0 and 64 nibbles of BANK1, can be used as the general-purpose data memory.

4.2.8 Data memory areas not provided

For these data memory areas, refer to **4.4.2 Notes on data memory areas not provided, 8.2 Configuration and Function of Peripheral Control Registers**, and **Table 10-1 Relation between Each Port (Pin) and Port Register**.

Figure 4-2. Configuration of Data Memory



Caution Address 40H of BANK1, bit 3 of address 50H, and address 73H are test mode areas. Do not write "1" to these areas.

Table 4-1. Data Memory Manipulation Instructions

Function		Instruction
Operation	Add	ADD ADDC
	Subtract	SUB SUBC
	Logical	AND OR XOR
Compare		SKE SKGE SKLT SKNE
Transfer		MOV LD ST
Judge		SKT SKF

4.3 Addressing Data Memory

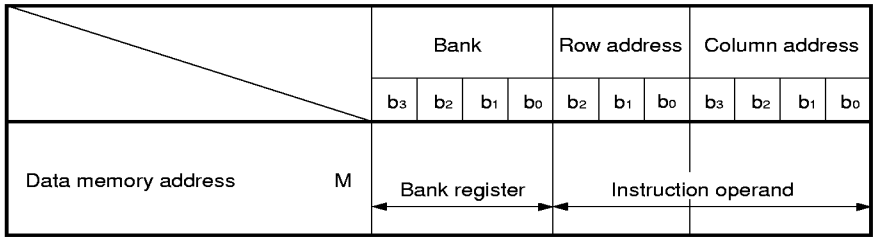
Figure 4-3 shows how to address the data memory.

An address of the data memory is specified by using a bank, row address, and column address.

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

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

Figure 4-3. Addressing Data Memory



4.4 Notes on Using Data Memory

4.4.1 On power-ON reset

On power-ON reset, the contents of the general-purpose data memory are “undefined”.
Initialize the memory if necessary.

4.4.2 Notes on data memory not provided

If a data memory manipulation instruction is executed to manipulate an address where no data memory is assigned, the following operations are performed:

(1) Device operation

When a read instruction is executed, “0” is read.
Nothing is changed even when a write instruction is executed.
Address 40H of BANK1, bit 3 of address 50H, and address 73H are test mode areas. Do not write “1” to these areas.

(2) Assembler operation

The program is assembled normally. No “error” occurs.

(3) In-circuit emulator operation

“0” is read when a read instruction is executed.
Nothing is changed when a write instruction is executed.
No “error” occurs.

5. SYSTEM REGISTER (SYSREG)

5.1 General

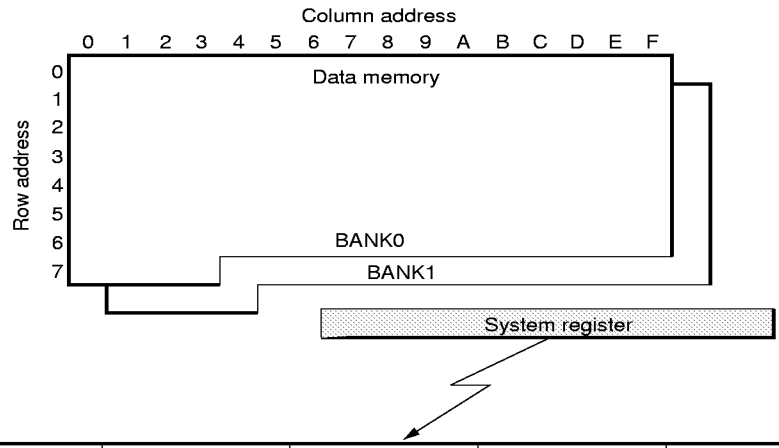
Figure 5-1 shows the location of the system register on the data memory and outline.

As shown, the system register is assigned to addresses 74H-7FH of the data memory, regardless of bank. In other words, the same system register is assigned to addresses 74H-7FH of any bank.

Since the system register is located on the data memory, it can be manipulated by all the data memory manipulation instructions.

With the μPD17073, only the address register (AR: 74H through 77H), bank register (BANK: 79H), and program status word (PSWORD: 7EH, 7FH) of addresses 74H through 7FH can be manipulated.

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



Address	74H	75H	76H	77H	78H	79H
Name	Address register (AR)				Fixed to 0	Bank register (BANK)
Outline	Controls program memory address					Specifies data memory bank

Address	7AH	7BH	7CH	7DH	7EH	7FH
Name	Fixed to 0					Program status word (PSWORD)
Outline						Controls operation

5.2 Address Register (AR)

5.2.1 Configuration of address register

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

As shown in this figure, the address register consists of 16 bits of the system register: 74H through 77H (AR3 through AR0). However, the higher 4 bits are always fixed to 0, and therefore, the address register actually functions as a 12-bit register.

Figure 5-2. Address Register Configuration

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

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

5.2.2 Functions of address register

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

A dedicated instruction ("INC AR") that can increment the value of the address register by one is available.

The following paragraphs (1) through (5) describe the operations of the address register when each of these instructions has been executed.

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

When the "MOVT DBF, @AR" instruction is executed, the constant data (16 bits) of the program memory address specified by the contents of the address register are read to the data buffer.

The addresses of the constant data which can be specified by the address register are 0000H-0FFFH.

(2) Stack manipulation instruction ("PUSH AR", "POP AR")

By executing the "PUSH AR" instruction, the stack pointer is decremented by one and the contents of the address register (AR) are stored to the address stack register specified by the stack pointer.

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

(3) Indirect branch instruction ("BR @AR")

When the "BR @AR" instruction is executed, the program execution branches to a program memory address specified by the contents of the address register.

The branch addresses that can be specified by the address register are 0000H-0FFFH.

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

When the "CALL @AR" instruction is executed, the subroutine at the program memory address specified by the contents of the address register can be called.

The first addresses of the subroutine that can be specified by the address register are 0000H-0FFFH.

(5) Address register increment instruction ("INC AR")

This instruction increments the contents of the address register by one each time it is executed.

Since the address register is configured of 12 bits, its contents become "0000H" when the "INC AR" instruction is executed with the contents of the address register being "0FFFH".

5.2.3 Address register and data buffer

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

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

5.3 Bank Register (BANK)

5.3.1 Configuration of bank register

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

As shown in this figure, the bank register consists of 4 bits of address 79H (BANK) of the system register. Note, however, that the higher 3 bits are always fixed to "0"; therefore, this register actually serves as a 1-bit register.

Figure 5-3. Bank Register Configuration

Address		79H			
Name		Bank register (BANK)			
Symbol		BANK			
Bit		b_3	b_2	b_1	b_0
Data		0	0	0	←
At reset	Power-ON	0			
	Clock stop	0			
	CE	0			

5.3.2 Function of bank register

The bank register selects a bank of the data memory.

Table 5-1 shows the value of the bank register and how a bank of the data memory is specified.

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

In other words, the current bank status has nothing to do with manipulation of the bank register.

Table 5-1. Specifying Bank of Data Memory

Bank register (BANK)				Bank of data memory
b_3	b_2	b_1	b_0	
0	0	0	0	BANK0
0	0	0	1	BANK1

5.4 Program Status Word (PSWORD)

5.4.1 Configuration of program status word

Figure 5-4 shows the configuration of the program status word.

As shown in this figure, the program status word consists of a total of 5 bits: the least significant bit of address 7EH (RPL) and 4 bits of 7FH (PSW) of the system register. However, bit 0 of 7FH is always fixed to 0.

Each of the 5 bits in the program status word has its own function as a BCD flag (BCD), compare flag (CMP), carry flag (CY), zero flag (Z), respectively.

Figure 5-4. Program Status Word Configuration

Address		7EH				7FH			
Name		(RP)			Program status word (PSWORD)				
Symbol		RPL				PSW			
Bit		b_0	b_2	b_2	b_0	b_0	b_2	b_2	b_0
Data					BCD	CMP	CY	Z	0
At reset	Power-ON	0				0			
	Clock stop	0				0			
	CE	0				0			

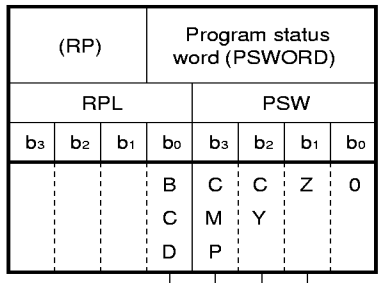
5.4.2 Functions of program status word

The program status word sets conditions, under which the ALU (Arithmetic Logic Unit) performs arithmetic or transfer operations, and indicates the results of the operations.

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

For details, refer to **7. ALU (Arithmetic Logic Unit) BLOCK**.

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



Flag name	Function
Zero flag (Z)	Indicates that the result of arithmetic operation is 0. Condition under which this flag is set differs depending on contents of compare flag.
Carry flag (CY)	Indicates occurrence of carry or borrow as a result of executing addition or subtraction instruction. Reset (0) when carry or borrow does not occur. Set (1) when carry or borrow occurs. Also used as shift bit of "RORC r" instruction.
Compare flag (CMP)	Stores or does not store result of arithmetic operation in data memory or general register. 0: Stores result 1: Does not store result
BCD flag (BCD)	Executes arithmetic operation in decimal. 0: Executes binary operation 1: Executes decimal operation

5.4.3 Notes on using program status word

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

Even if an operation that generates a carry has been executed, for example, if the result of the operation is 0000B, 0000B is stored in PSW.

5.5 Notes on Using System Register

The data in the system register which are fixed to “0” are not influenced even when a write instruction is executed.

When these data are read, “0” is always read.

6. GENERAL REGISTERS (GR)

6.1 Outline of General Registers

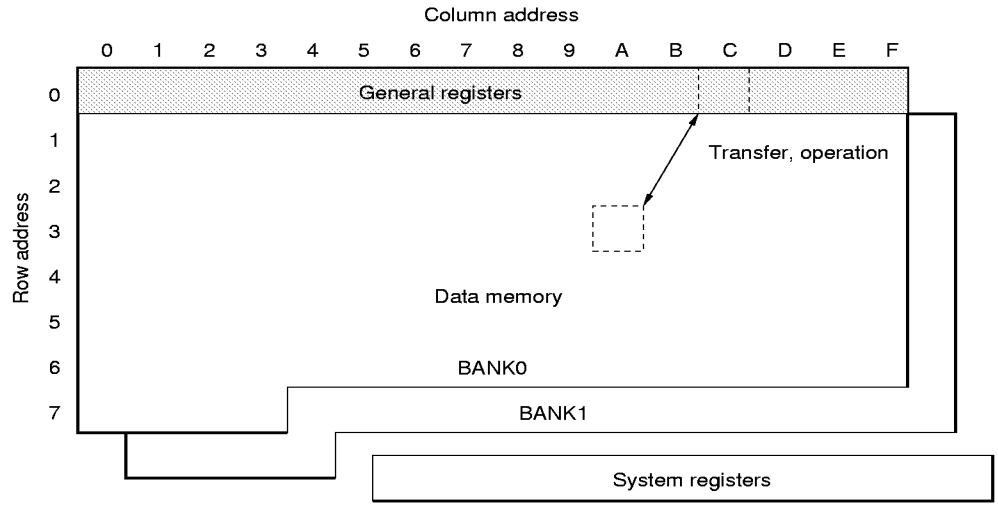
With the μPD17073, the general registers are fixed at row address 0 of BANK0 on the data memory, and consist of 16 nibbles (16 × 4 bits) of 00H through 0FH.

The 16 nibbles of the row address 0 specified as the general registers can perform operations and data transfer with the data memory with a single instruction.

In other words, operations and data transfer between data memory areas can be executed with a single instruction.

The general registers can be controlled by data memory manipulation instructions, like the other data memory areas.

Figure 6-1. Outline of General Registers



6.2 Address Creation of General Register with Each Instruction

The following paragraphs 6.2.1 and 6.2.2 describe how the address of the general register is created when each instruction is executed.

For details of the operation of each instruction, refer to **7. ALU (Arithmetic Logic Unit) BLOCK**.

6.2.1 Addition ("ADD r, m", "ADDC r, m"), subtraction ("SUB r, m", "SUBC r, m"), logical operation ("AND r, m", "OR r, m", "XOR r, m"), direct transfer ("LD r, m", "ST m, r"), rotate processing ("RORC r") instructions

Table 6-1 shows the address of general register "R" specified by an instruction operand "r". The operand "r" specifies only the column address.

Table 6-1. Address Creation of General Register

		Bank				Row address			Column address			
		b_3	b_2	b_1	b_0	b_2	b_1	b_0	b_3	b_2	b_1	b_0
General register address	R	Fixed to 0				Fixed to 1			r			

6.2.2 Indirect transfer ("MOV @r, m", "MOV m, @r") instructions

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

Table 6-2. Address Creation of General Register

		Bank				Row address			Column address			
		b_3	b_2	b_1	b_0	b_2	b_1	b_0	b_3	b_2	b_1	b_0
General register address	R	Fixed to 0				Fixed to 0			r			
Indirect transfer address	@R	Fixed to 0				Fixed to 0			Contents of R			

6.3 Notes on Using General Register

There is no instruction available that performs an operation between the general register and immediate data.

To perform an operation between the data memory specified as the general register and immediate data, the data memory must be treated as data memory instead of as the general register.

7. ALU (ARITHMETIC LOGIC UNIT) BLOCK

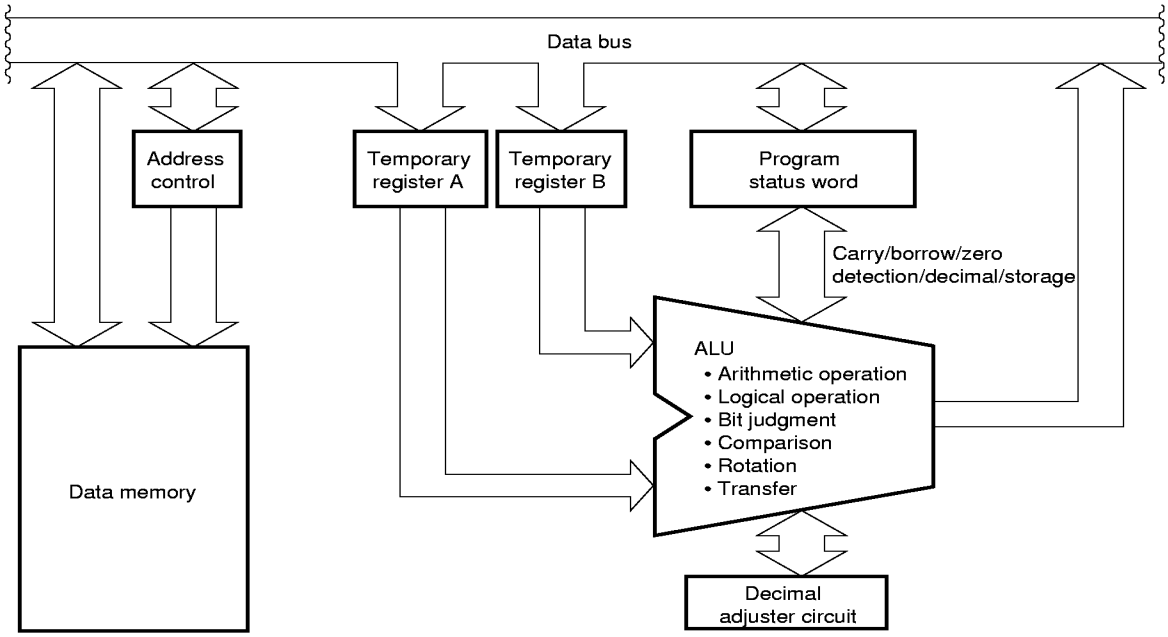
7.1 General

Figure 7-1 shows the configuration of the ALU block.

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

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

Figure 7-1. Outline of ALU Block



7.2 Configuration and Function of Each Block

7.2.1 Functions of ALU

The ALU performs arithmetic operation, logical operation, bit judgment, comparison, rotation, and transfer of 4-bit data as the instruction specified by the program.

7.2.2 Temporary registers A and B

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

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

7.2.3 Program status word

The program status word controls the operations of the ALU and stores the status of the ALU. For details, refer to **5.4 Program Status Word (PSWORD)**.

7.2.4 Decimal adjuster circuit

When the BCD flag of the program status word is set to 1 during an arithmetic operation, the result of the operation is converted into decimal numbers by the decimal adjuster circuit.

7.2.5 Address control circuit

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

7.3 ALU Processing Instructions

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

Table 7-2 shows the decimal adjusted data when a decimal operation is performed.

Table 7-1. ALU Processing Instruction List

ALU function	Instruction		Difference of operation due to program status word (PSWORD)				
			Value of BCD flag	Value of CMP flag	Operation	Operation of CY flag	Operation of Z flag
Addition	ADD	r, m	0	0	Stores result of binary addition	Set if carry or borrow occurs; otherwise, reset	Set if result of operation is 0000B; otherwise, reset
		m, #n4					
	ADDC	r, m	0	1	Does not store result of binary operation		Retains status if result of operation is 0000B; otherwise, reset
		m, #n4					
Subtraction	SUB	r, m	1	0	Stores result of decimal operation		Set if result of operation is 0000B; otherwise, reset
		m, #n4					
	SUBC	r, m	1	1	Does not store result of decimal operation		Retains status if result of operation is 0000B; otherwise, reset
		m, #n4					
Logical operation	OR	r, m	Any (retained)	Any (retained)	No change	Retains previous status	Retains previous status
		m, #n4					
	AND	r, m					
		m, #n4					
	XOR	r, m					
		m, #n4					
Judgment	SKT	m, #n	Any (retained)	Any (reset)	No change	Retains previous status	Retains previous status
	SKF	m, #n					
Comparison	SKE	m, #n4	Any (retained)	Any (retained)	No change	Retains previous status	Retains previous status
	SKNE	m, #n4					
	SKGE	m, #n4					
	SKLT	m, #n4					
Transfer	LD	r, m	Any (retained)	Any (retained)	No change	Retains previous status	Retains previous status
	ST	m, r					
	MOV	m, #n4					
		@r, m					
		m, @r					
Rotation	RORC	r	Any (retained)	Any (retained)	No change	Value of b_0 of general register	Retains previous status

Table 7-2. Decimal Adjusted Data

Result of operation	Hexadecimal addition		Decimal addition	
	CY	Result of operation	CY	Result of operation
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

Result of operation	Hexadecimal subtraction		Decimal subtraction	
	CY	Result of operation	CY	Result of operation
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 The shaded part indicates that decimal adjustment is not made correctly.

7.4 Notes on Using ALU

7.4.1 Notes on executing operation to program status word

When an arithmetic operation is performed to the program status word, the result of the operation is stored 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 an arithmetic operation executed. However, if the program status word itself is used for an operation, the result of the operation is stored in the program status word, making it impossible to judge whether a carry or borrow occurs, or the result of the operation is zero.

However, if the CMP flag is set, the result of the operation is not stored in the program status word; consequently, the CY and Z flags are set (1) or cleared (0) normally.

7.4.2 Notes on using decimal operation

A decimal operation can be executed only if the result of the operation falls within the following range:

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

If a decimal operation is executed exceeding this range, the CY flag is set, and the result is 1010B (0AH) or higher.

8. PERIPHERAL CONTROL REGISTERS

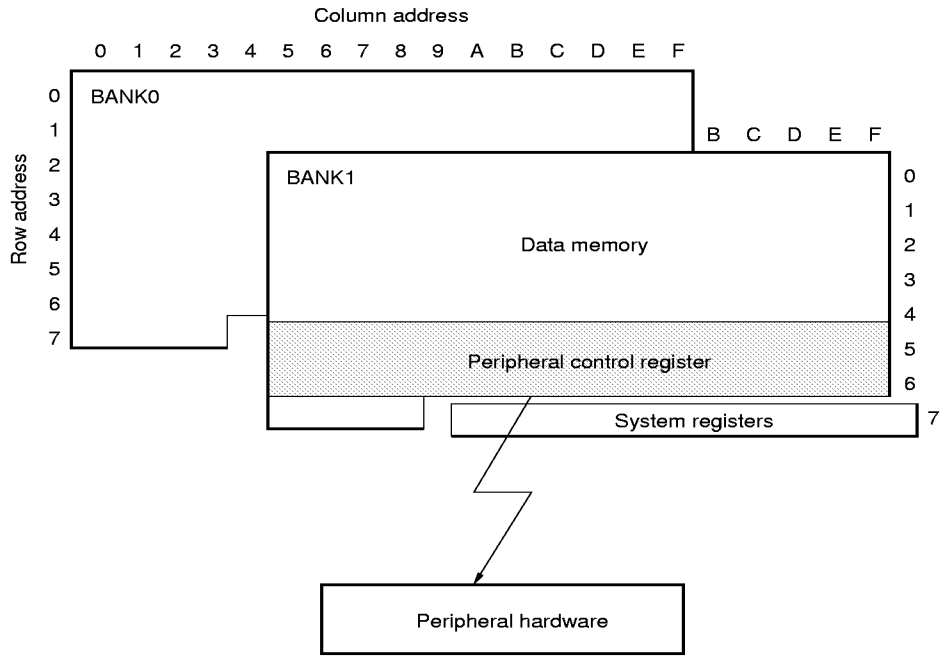
8.1 Outline of Peripheral Control Registers

Figure 8-1 outlines the peripheral control registers.

Thirty-two 4-bit peripheral registers are available that control the peripheral hardware such as the PLL frequency synthesizer, serial interface, and intermediate frequency counter (IF).

Because the peripheral control registers are located on the data memory, they can be manipulated by all the data memory manipulation instructions.

Figure 8-1. Outline of Peripheral Control Registers



8.2 Configuration and Function of Peripheral Control Registers

Figure 8-2 shows the configuration of the peripheral control registers.

Table 8-1 lists the peripheral hardware control functions of the peripheral control registers.

As shown in Figure 8-2, the peripheral control registers consist of a total of 32 nibbles (32 × 4 bits) of addresses 50H through 6FH of BANK1.

Each peripheral control register has an attribute of 1 nibble, and is classified into four types: read/write (R/W), read-only (R), write-only (W), and read-and-reset (R&Reset) registers.

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

If a write-only (W) register is read, the value is undefined.

Of the 4-bit data of 1 nibble, the bits fixed to "0" are always "0" when read, and also "0" when data is written to these bits.

Caution Bit 3 of address 50H of BANK1 (bit 3 of the LCD driver display start register) is allocated to a test mode area. Therefore, do not write "1" to this bit.

[MEMO]

Figure 8-2. Configuration of Peripheral Control Registers (1/2)

(BANK1) Column address / Row address / Item		0	1	2	3	4	5	6	7
5	Name	LCD driver display start register	Basic timer 0 carry register	CE pin status detection register	Port 1A pull-down resistor select register	Stack pointer	System clock select register	Interrupt edge select register	Interrupt enable register
	Symbol	Note 0 / 0 / ADCON / LCDEN	0 / 0 / 0 / BTM0CY	0 / 0 / 0 / CE	P1APLD3 / P1APLD2 / P1APLD1 / P1APLD0	0 / 0 / SP1 / SP0	0 / 0 / 0 / SYSCK	0 / INT / BTM1CK / IEG	0 / IPSIO / IPBTM1 / IP
	Read/ Write	R/W	R&Reset	R	R/W	R/W	R/W	R/W	R/W
6	Name	Serial I/O mode select register	Serial I/O clock select register	IF counter mode select register	IF counter gate status detection register	IF counter control register	PLL mode select register	PLL reference frequency select register	PLL data register
	Symbol	0 / SIOSEL / SIOHIZ / SIOTS	0 / 0 / SIOCK1 / SIOCK0	IFCMD1 / IFCMD0 / IFCCK1 / IFCCK0	0 / 0 / 0 / IFCG	0 / 0 / IFCSTRT / IFCRES	0 / 0 / PLLMD1 / PLLMD0	0 / PLLRFCK2 / PLLRFCK1 / PLLRFCK0	PLLR17 / PLLR16 / PLLR15 / PLLR14
	Read/ Write	R/W	R/W	R/W	R	W	R/W	R/W	R/W

Note This is a test mode area. Do not write “1” to this area.

Figure 8-2. Configuration of Peripheral Control Registers (2/2)

8	9	A	B	C	D	E	F
INT pin interrupt request register	Basic timer 1 interrupt request register	Serial interface interrupt request register	BEEP clock select register	A/D converter channel select register	A/D converter reference voltage setting register	A/D converter compare start register	A/D converter compare result detection register
0 0 0 IRQ	0 0 0 IRQBTM1	0 0 0 IRQSIO	0 0 BEEP0CK1 BEEP0CK0	0 0 ADCCH1 ADCCH0	ADCRFSEL3 ADCRFSEL2 ADCRFSEL1 ADCRFSEL0	0 0 0 ADCSTRT	0 0 0 ADCCMP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
PLL data register				PLL data set register	PLL unlock FF register	Port 0B bit I/O select register	Port 0C bit I/O select register
PLLR13 PLLR12 PLLR11 PLLR10	PLLR9 PLLR8 PLLR7 PLLR6	PLLR5 PLLR4 PLLR3 PLLR2	PLLR1 0 0 0	0 0 0 PLLPUT	0 0 0 PLLUL	P0BBIO3 P0BBIO2 P0BBIO1 P0BBIO0	P0DBIO3 P0DBIO2 P0CBIO1 P0CBIO0
R/W				W	R&Reset	R/W	R/W

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

Peripheral hardware	Control register				Peripheral hardware control function			At reset		
	Name	Address	Read/ Write	b_3 b_2 Symbol b_1 b_0	Functional outline	Set value 0	Set value 1	Power-ON	Clock stop	CE
Stack	Stack pointer (SP)	(BANK1) 54H	R/W	0	Fixed to 0			2	2	2
				0						
				SP1	Stack pointer					
				SP0						
Timer	Basic timer 0 carry register	(BANK1) 51H	R& Reset	0	Fixed to 0			0	1	1
				0						
				0						
				BTM0CY	Detects status of carry FF	Reset	Set			
Interrupt	Interrupt edge select register	(BANK1) 56H	R/W	0	Fixed to 0			0	0	0
				INT	Detects status of INT Pin	Low level	High level			
				BTM1CK	Sets set time interval of IRQBTM1 flag	32 ms (31.25 Hz)	8 ms (125 Hz)			
				IEG	Sets interrupt issuing edge of INT pin	Rising edge	Falling edge			
	Interrupt enable register	(BANK1) 57H	R/W	0	Fixed to 0			0	0	0
				IPSIO	Serial interface } Enables interrupt	Disables interrupt	Enables interrupt			
				IPBTM1	Basic timer 1 } Enables interrupt					
				IP	INT pin } Enables interrupt					
	INT pin interrupt request register	(BANK1) 58H	R/W	0	Fixed to 0			0	0	0
				0						
				0						
				IRQ	Detects interrupt request of INT pin	Not requested	Requested			
	Basic timer 1 interrupt request register	(BANK1) 59H	R/W	0	Fixed to 0			0	0	0
				0						
				0						
				IRQBTM1	Detects interrupt request of basic timer 1	Not requested	Requested			
	Serial interface interrupt request register	(BANK1) 5AH	R/W	0	Fixed to 0			0	0	0
				0						
				0						
				IRQSIO	Detects interrupt of serial interface	Not requested	Requested			
Pin	CE pin status detection register	(BANK1) 52H	R	0	Fixed to 0			–	–	–
				0						
				0						
				CE	Detects status of CE pin	Low level	High level			
	Port 1A pull-down select register	(BANK1) 53H	R/W	P1APLD3	P1A3 } Selects pull-down resistor of these pins	Resistor ON	Resistor OFF	0	R	R
				P1APLD2	P1A2 } Selects pull-down resistor of these pins					
				P1APLD1	P1A1 } Selects pull-down resistor of these pins					
				P1APLD0	P1A0 } Selects pull-down resistor of these pins					

Remark –: Determined by status of pin, R: Previous status is retained.

Table 8-1. Peripheral Hardware Control Functions of Peripheral Control Registers (2/4)

Peripheral hardware	Control register				Peripheral hardware control function			At reset		
	Name	Address	Read/ Write	b_3 b_2 Symbol b_1 b_0	Functional outline	Set value		Power-ON	Clock stop	CE
						0	1			
PLL frequency synthesizer	PLL mode select register	(BANK1) 65H	R/W	0	Fixed to 0			0	0	R
				0						
				PLLMD1	Sets division mode of PLL	0 0 1 1 Disable MF VHF HF				
				PLLMD0		0 1 0 1				
	PLL reference frequency select register	(BANK1) 66H	R/W	0	Fixed to 0			0	0	R
				PLLRFCK2	Sets reference frequency of PLL	0:1 kHz 1:3 kHz 2:5 kHz 3:6.25 kHz 4:12.5 kHz 5:25 kHz 6, 7: PLL disable				
				PLLRFCK1						
				PLLRFCK0						
	PLL data register	(BANK1) 67H	R/W	PLLR17	Sets division ratio of PLL	• In direct division mode PLLR6-PLLR17: Valid data PLLR1-PLLR5: don't care 0-15 (000H-00FH): Setting prohibited 16-$2^{12}-1$ (010H-FFFH): Can be set[Note] • In pulse swallow mode PLLR1-PLLR17: Valid data 0-1023 (0000H-03FFH): Setting prohibited 1024-$2^{17}-1$ (0400H-1FFFFH): Can be set[Note]		U	R	R
				PLLR16						
				PLLR15						
				PLLR14						
		(BANK1) 68H		PLLR13						
				PLLR12						
				PLLR11						
				PLLR10						
		(BANK1) 69H		PLLR9						
				PLLR8						
				PLLR7						
				PLLR6						
		(BANK1) 6AH		PLLR5						
				PLLR4						
				PLLR3						
				PLLR2						
		(BANK1) 6BH		PLLR1						
				0	Fixed to 0					
				0						
				0						
	PLL data set register	(BANK1) 6CH	W	0	Fixed to 0			0	0	0
				0						
				0						
				PLLPUT	Data transfer to programmable counter	Does not transfer	Transfers			
	PLL unlock FF register	(BANK1) 6DH	R& Reset	0	Fixed to 0			U	R	R
				0						
				0						
				PLLUL	Detects status of unlock FF	Locked status	Unlocked status			

Note For the details of the set value, refer to **Figure 15-4 Configuration of PLL Data Register**.

Remark U: Undefined, R: Previous status is retained.

Table 8-1. Peripheral Hardware Control Functions of Peripheral Control Registers (3/4)

Peripheral hardware	Control register: Name	Address	Read/ Write	b_3 b_2 b_1 b_0 Symbol	Peripheral hardware control function: Functional outline	Set value: 0	Set value: 1	At reset: Power-ON	At reset: Clock stop	At reset: CE
A/D converter	A/D converter channel select register	(BANK1) 5CH	R/W	0	Fixed to 0			0	0	0
				0						
				ADCCH1	Selects pin used for A/D converter	0 0: Not used; 0 1: AD0	1 0: AD1; 1 1: AD1			
				ADCCH0						
	A/D converter reference voltage setting register	(BANK1) 5DH	R/W	ADCRFSEL3	Sets compare voltage	$V_{REF} = \frac{x + 0.5}{16} \times V_{DD}$ (V) (0 ≤ x ≤ 0FH)		0	0	0
				ADCRFSEL2						
				ADCRFSEL1						
				ADCRFSEL0						
	A/D converter compare start register	(BANK1) 5EH	R/W	0	Fixed to 0			0	0	0
				0						
				0						
				ADCSTRT	Starts A/D converter operation/checks comparator operation	Invalid/Stop	Starts/operates			
	A/D converter compare result detection register	(BANK1) 5FH	R	0	Fixed to 0			0	0	0
				0						
				0						
				ADCCMP	Detects compare result	$V_{ADCIN} < V_{REF}$	$V_{ADCIN} > V_{REF}$			
General-purpose port	Port 0B bit I/O select register	(BANK1) 6EH	R/W	P0BBIO3	P0B3 pin	Input	Output	0	0	0
				P0BBIO2	P0B2 pin					
				P0BBIO1	P0B1 pin					
				P0BBIO0	P0B0 pin					
	Port 0C bit I/O select register	(BANK1) 6FH	R/W	P0DBIO3	P0D3 pin					
				P0DBIO2	P0D2 pin					
				P0CBIO1	P0C1 pin					
				P0CBIO0	P0C0 pin					
					Sets I/O mode of these pins (bit I/O)					
Serial interface	Serial I/O mode select register	(BANK1) 60H	R/W	0	Fixed to 0			0	0	0
				SIOSEL	Selects serial I/O mode of P0B3/SI/SO1 pin	Serial input	Serial output			
				SIOHIZ	Sets P1C0/SO0 pin in serial output mode	General-purpose output port	Serial output			
				SIOTS	Sets start or stop of operation	Stops operation	Starts operation			
	Serial I/O clock select register	(BANK1) 61H	R/W	0	Fixed to 0			0	0	0
				0						
				SIOCK1	Sets clock of serial interface	0 0: External clock; 0 1: 12.5 kHz	1 0: 18.75 kHz; 1 1: 37.5 kHz			
				SIOCK0						

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

Peripheral hardware	Control register			b_3 b_2 b_1 b_0 Symbol	Peripheral hardware control function			At reset		
	Name	Address	Read/ Write		Functional outline	Set value		Power-ON	Clock stop	CE
						0	1			
IF counter	IF counter mode select register	(BANK1) 62H	R/W	IFCMD1	Sets mode of IF counter	0 0 1 1 IF counter OFF (General I/O port) FMIFC pin FMIF mode AMIFC pin AMIF mode FMIFC pin AMIF mode 0 1 0 1		0	0	0
				IFCMD0						
				IFCCK1	Sets gate time of IF counter	0 0 1 1 1 ms 4 ms 8 ms Open 0 1 0 1				
				IFCCK0						
	IF counter gate status detection register	(BANK1) 63H	R	0	Fixed to 0			0	0	0
				0						
				0						
				IFCG	Detects opening/closing gate of IF counter	Closed	Open			
	IF counter control register	(BANK1) 64H	W	0	Fixed to 0			0	0	0
				0						
				IFCSTRT	Starts counting of IF counter	Does not start	Starts			
				IFCRES	Resets IF counter	Does not reset	Resets			
BEEP	BEEP clock select register	(BANK1) 5BH	R/W	0	Fixed to 0			0	0	R
				0						
				BEEP0CK1	Sets output status of BEEP pin	0 0 1 1 General output port (low-level output) General output port (high-level output) BEEP (1.5 kHz) BEEP (3 kHz) 0 1 0 1				
				BEEP0CK0						
LCD controller/ driver	LCD driver display start register	(BANK1) 50H	R/W	0	Fixed to 0					
				0						
				ADCON[Note]	Sets A/D converter power supply and	0 0 1 1 Power OFF Power ON Power ON Power ON Display OFF Display ON Display OFF Display ON 0 1 0 1		0	0	0
				LCDEN	ON/OFF of all LCD display			0	0	R
Standby	System clock select register	(BANK1) 55H	R/W	0	Fixed to 0			0	R	R
				0						
				0						
				SYSCK	Selects system clock (1 instruction execution time)	53.3 μs	106.6 μs			

Note When LCDEN= 1, the power supply for the A/D converter is ON even if ADCON = 0.

Remark R: Previous status is retained.

9. DATA BUFFER (DBF)

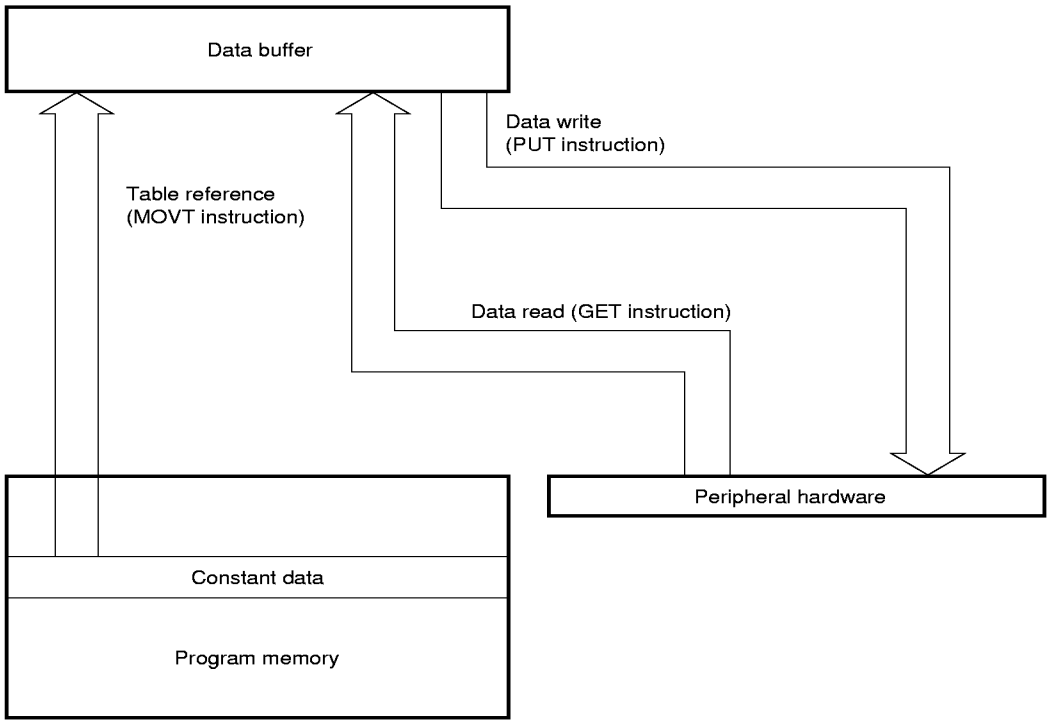
9.1 General

Figure 9-1 outlines the data buffer.

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

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

Figure 9-1. Outline of Data Buffer



9.2 Data Buffer

9.2.1 Configuration of data buffer

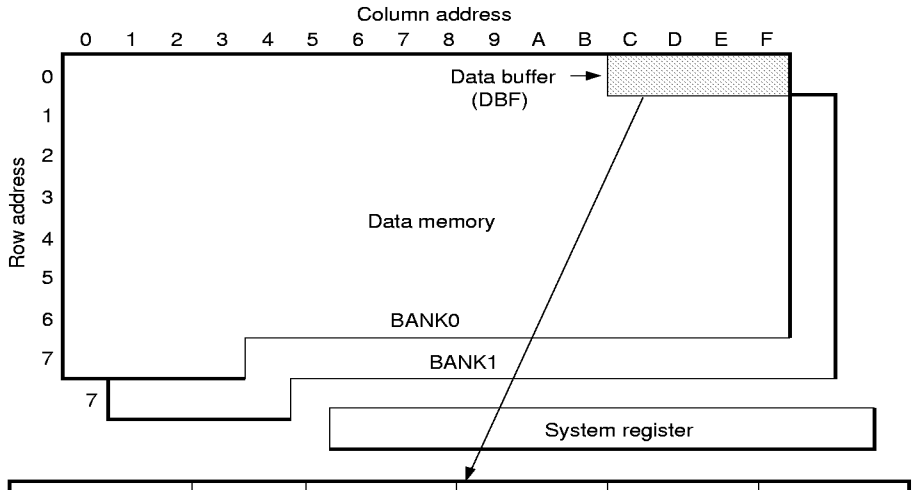
Figure 9-2 shows the configuration of the data buffer.

As shown in this figure, the data buffer is configured of 16 bits of addresses 0CH-0FH of BANK0 on the data memory.

Of these 16 bits, bit 3 of address 0CH is the MSB, while bit 0 of address 0FH is the LSB.

Since the data buffer is on the data memory, it can be manipulated by all the data memory manipulation instructions.

Figure 9-2. Configuration of Data Buffer



Data memory	Address	0CH				0DH				0EH				0FH			
	Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data 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	DBF3				DBF2				DBF1				DBF0			
	Data	M S B															L S B
		Data															

9.2.2 Table reference instruction ("MOVT DBF, @AR")

When this instruction is executed, the contents of the program memory addressed by the contents of the address register are incorporated into the data buffer.

The program memory addresses to which table reference can be executed are addresses 0000H-0FFFH, i.e., all the addresses of the program memory.

9.2.3 Peripheral hardware control instructions ("PUT" and "GET")

The operations of the "PUT" and "GET" instructions are as follows:

(1) GET DBF, p

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

(2) PUT p, DBF

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

9.3 List of Peripheral Hardware and Data Buffer Functions

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

9.4 Notes on Using Data Buffer

When transferring data with the peripheral hardware through the data buffer, keep in mind the following three points in respect with the unused peripheral addresses, write-only peripheral registers (only when using PUT), and read-only peripheral registers (only when using GET):

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

Table 9-1. Relation between Peripheral Hardware and Data Buffer

Peripheral hardware	Peripheral register transferring data with data buffer				Function		
	Name	Symbol	Peripheral address	Execution of PUT/GET instruction	No. of data buffer I/O bits	No. of actual bits	Outline
Serial interface	Presettable shift register	SIOSFR	03H	PUT/ GET	8	8	Sets serial out data and reads serial in data
Address register (AR)	Address register	AR	40H	PUT/ GET	16	12	Transfers data with address register
IF counter	IF counter data register	IFC	43H	GET	16	16	Reads count value of IF counter

10. GENERAL-PURPOSE PORT

The general-purpose ports output high or low floating signals to external circuits, and reads high or low level signals from external circuits.

10.1 General

Table 10-1 shows the relations between each port and port register.

The general-purpose ports are classified into I/O ports, input ports, and output ports.

The I/O port is the bit I/O ports, which can be set in the input or output mode in 1-bit (1-pin) units.

Table 10-1. Relations between Each Port (Pin) and Port Register

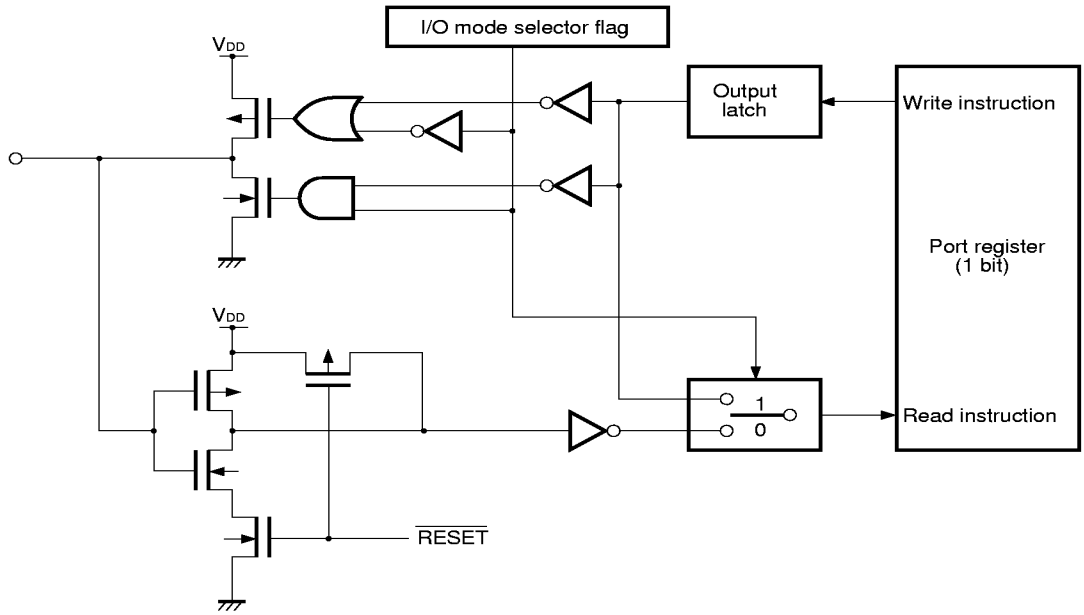
Port	Pin				Data setting method					
	No.		Symbol	I/O	Port register (data memory)					Remarks
	56-pin QFP	64-pin TQFP			Bank	Address	Symbol	Bit symbol (reserved word)		
Port 0A	5	6	P0A3	Output	BANK0	70H	P0A	b_3	P0A3	
	4	4	P0A2					b_2	P0A2	
	3	3	P0A1					b_1	P0A1	
	2	2	P0A0					b_0	P0A0	
Port 0B	56	64	P0B3	I/O (bit I/O)		71H	P0B	b_3	P0B3	
	55	63	P0B2					b_2	P0B2	
	54	62	P0B1					b_1	P0B1	
	53	61	P0B0					b_0	P0B0	
Port 0C	No pin					72H	P0C	b_3	–	Fixed to "0"
								b_2	–	
	15	17	P0C1	I/O (bit I/O)				b_1	P0C1	
	14	16	P0C0					b_0	P0C0	
Port 0D	17	19	P0D3	I/O (bit I/O)		73H	P0D	b_3	P0D3	
	16	18	P0D2					b_2	P0D2	
	No pin							b_1	–	Fixed to "0"
								b_0	–	
Port 1A	13	15	P1A3	Input	BANK1	70H	P1A	b_3	P1A3	
	12	14	P1A2					b_2	P1A2	
	11	13	P1A1					b_1	P1A1	
	10	11	P1A0					b_0	P1A0	
Port 1B	9	10	P1B3	Output		71H	P1B	b_3	P1B3	
	8	9	P1B2					b_2	P1B2	
	7	8	P1B1					b_1	P1B1	
	6	7	P1B0					b_0	P1B0	
Port 1C	No pin					72H	P1C	b_3	–	
								b_2	–	Fixed to "0"
								b_1	–	
	1	1	P1C0	Output				b_0	P1C0	
						73H	–	b_3	–	Test mode area. Do not write "1" to this area.
								b_2		
								b_1		
								b_0		

10.2 General-Purpose I/O Ports (P0B, P0C, P0D)

10.2.1 Configuration of I/O ports

The configurations of the I/O ports are shown below.

P0B (P0B3, P0B2, P0B1, P0B0)
P0C (P0C1, P0C0)
P0D (P0D3, P0D2)



10.2.2 Use of I/O ports

The I/O port is set in the input or output mode by the I/O select registers P0B and P0C of the control register.

P0D, P0C, and P0D are the bit I/O ports. Therefore, these ports can set in input or output mode in 1-bit units.

To set output data or to read input data, data is written to the corresponding port register, or an instruction that reads the data is executed.

10.2.3 describes the configuration of the I/O select register of each port.

10.2.4 describes how to use an I/O port as an input port.

10.2.5 describes how to use an I/O port as an output port.

10.2.3 Control register of I/O port

The port 0B bit I/O select register sets the input or output mode of each pin of P0B. The port 0C bit I/O select register sets the input or output mode of each pin of P0C and P0D.

The following paragraphs (1) and (2) describe the configuration and function.

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

Name	Flag symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Port 0B bit I/O select register	P0BBIO3	P0BBIO2	P0BBIO1	P0BBIO0	(BANK1) 6EH	R/W

P0BBIO0	Sets input or output mode
0	Sets P0B0 in input mode
1	Sets P0B0 in output mode

P0BBIO1	Sets input or output mode
0	Sets P0B1 in input mode
1	Sets P0B1 in output mode

P0BBIO2	Sets input or output mode
0	Sets P0B2/$\overline{SCK}$ in input mode
1	Sets P0B2/$\overline{SCK}$ in output mode

P0BBIO3	Sets input or output mode
0	Sets P0B3/SI/SO1 in input mode
1	Sets P0B3/SI/SO1 in output mode

		P0BBIO3	P0BBIO2	P0BBIO1	P0BBIO0
At reset	Power-ON	0	0	0	0
	Clock stop	0	0	0	0
	CE	0	0	0	0

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

Name	Flag symbol b_3	b_2	b_1	b_0	Address	Read/ Write
Port 0C bit I/O select register	P0DBIO3	P0DBIO2	P0CBIO1	P0CBIO0	(BANK1) 6FH	R/W

P0CBIO0

	Sets input or output of port
0	Sets P0C0 pin in input mode
1	Sets P0C0 pin in output mode

P0CBIO1

	Sets input or output of port
0	Sets P0C1 pin in input mode
1	Sets P0C1 pin in output mode

P0DBIO2

	Sets input or output of port
0	Sets P0D2/AMIFC pin in input mode
1	Sets P0D2/AMIFC pin in output mode

P0DBIO3

	Sets input or output of port
0	Sets P0D3/FMIFC/AMIFC pin in input mode
1	Sets P0D3/FMIFC/AMIFC pin in output mode

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

10.2.4 To use I/O port in input mode

The port pin to be used in the input mode is selected by the I/O select register of each port.

The pin set in the input mode is floated (Hi-Z) and waits for the input of an external signal.

The input data can be read by executing a read instruction (such as SKT instruction) to the port register corresponding to each pin.

"1" is read from the port register when the high level is input to the corresponding pin, and "0" is read from the register when the low level is input to the pin.

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

10.2.5 To use I/O Port in output mode

The port pin to be set in the output mode is selected by the I/O select register corresponding to the port.

The pin set in the output mode outputs the contents of the output latch.

The output data is set by executing a write instruction (such as MOV instruction) to the port register corresponding to each pin.

To output the high level to each pin, "1" is written to the port register, and to output the low level, "0" is written.

The port pin can be floated (Hi-Z) by setting it in the input mode.

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

10.2.6 I/O port status on reset

(1) On power-ON reset

All the ports are set in the input mode.

The contents of the output latch become 0.

(2) On CE reset

All the ports are set in the input mode.

The contents of the output latch are retained.

(3) On execution of clock stop instruction

All the ports are set in the input mode.

The contents of the output latch are retained.

Increasing current consumption can be prevented due to noise of the input buffer, by using the $\overline{\text{RESET}}$ signal, as described in 10.2.1.

(4) In halt status

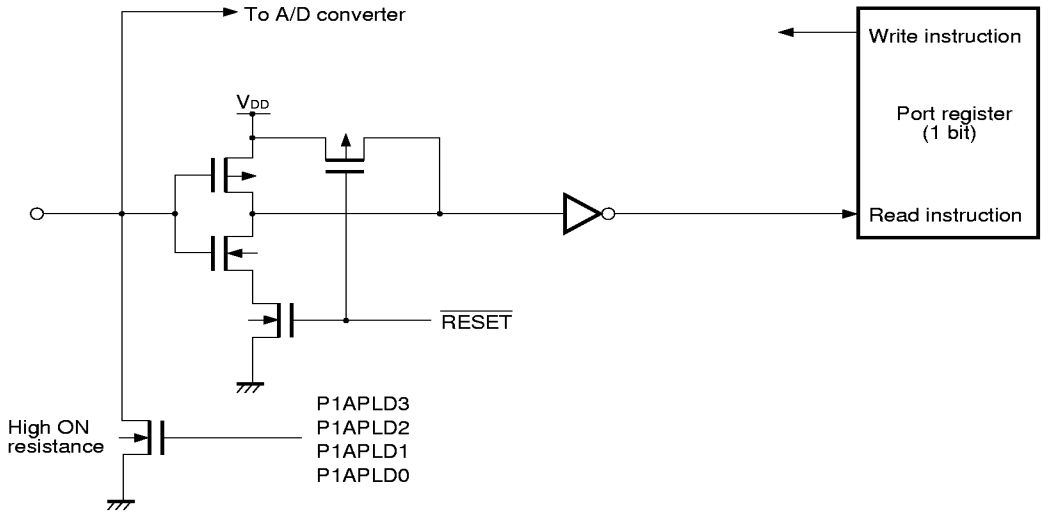
The previous status is retained.

10.3 General-Purpose Input Ports (P1A)

10.3.1 Configuration of input ports

The configuration of the input ports is illustrated below.

P1A (P1A3, P1A2, P1A1, P1A0)



10.3.2 Using input port

The input data can be read by executing an instruction that reads the contents of the port register P1A (such as SKT instruction).

"1" is read from each bit of the port register when the high level is input to the corresponding port pin, and "0" is read when the low level is input.

Nothing is changed even if a write instruction (such as MOV) is executed to the port register.

Port 1A can be connected to or disconnected from a pull-down resistor bitwise by software. Whether the pull-down resistor is connected or disconnected is specified by the port 1A pull-down resistor select register.

Figure 10-1 shows the configuration and function of the port 1A pull-down resistor select register.

Figure 10-1. Configuration of Port 1A Pull-Down Resistor Select Register

Name	Flag symbol b_3	b_2	b_1	b_0	Address	Read/ Write
Port 1A pull-down resistor select register	P1APLD3	P1APLD2	P1APLD1	P1APLD0	(BANK1) 53H	R/W

Selects ON/OFF of pull-down resistor of P1A0 pin	
0	Pull-down resistor ON
1	Pull-down resistor OFF

Selects ON/OFF of pull-down resistor of P1A1 pin	
0	Pull-down resistor ON
1	Pull-down resistor OFF

Selects ON/OFF of pull-down resistor of P1A2/AD0 pin	
0	Pull-down resistor ON
1	Pull-down resistor OFF

Selects ON/OFF of pull-down resistor of P1A3/AD1 pin	
0	Pull-down resistor ON
1	Pull-down resistor OFF

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

10.3.3 Reset status of input port

(1) On power-ON reset

All pins are specified as a input port.
Pulled down internally.

(2) On CE reset

All pins are specified as a input port.
The previous status of the pull-down resistor is retained.

(3) On execution of clock stop instruction

All pins are specified as a input port.
The previous status of the pull-down resistor is retained.

(4) In halt status

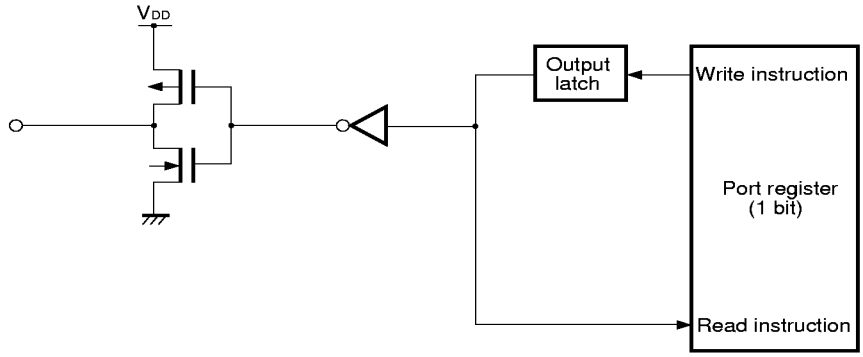
The previous status is retained.

10.4 General-Purpose Output Ports (P0A, P1B, P1C)

10.4.1 Configuration of output ports

The configurations of the output ports are shown below.

P0A (P0A3, P0A2, P0A1, P0A0)
P1B (P1B3, P1B2, P1B1, P1B0)
P1C (P1C0)



10.4.2 Using output port

The output port outputs the contents of the output latch from its pins.

The output data is set by executing an instruction that writes data to the port register corresponding to each pin (such as MOV instruction).

"1" is written to each bit of the port register when the high level is output to the corresponding port pin, and "0" is written when the low level is output.

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

10.4.3 Reset status of output port

(1) On power-ON reset

All the pins output the contents of the output latch.
The contents of the output latch become 0.

(2) On CE reset

Retains the contents of the output latch.
The contents of the output latch are retained; therefore, the output data is not changed on CE reset.

(3) On execution of clock stop instruction

Retains the contents of the output latch.
The contents of the output latch are retained; therefore, the output data is not changed on execution of the clock stop instruction.
Initialize the port through program as necessary.

(4) In halt status

The contents of the output latch are output.
The contents of the output latch are retained; therefore, the output data is not changed in the halt status.

11. INTERRUPT

11.1 General

Figure 11-1 shows the outline of the interrupt block.

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

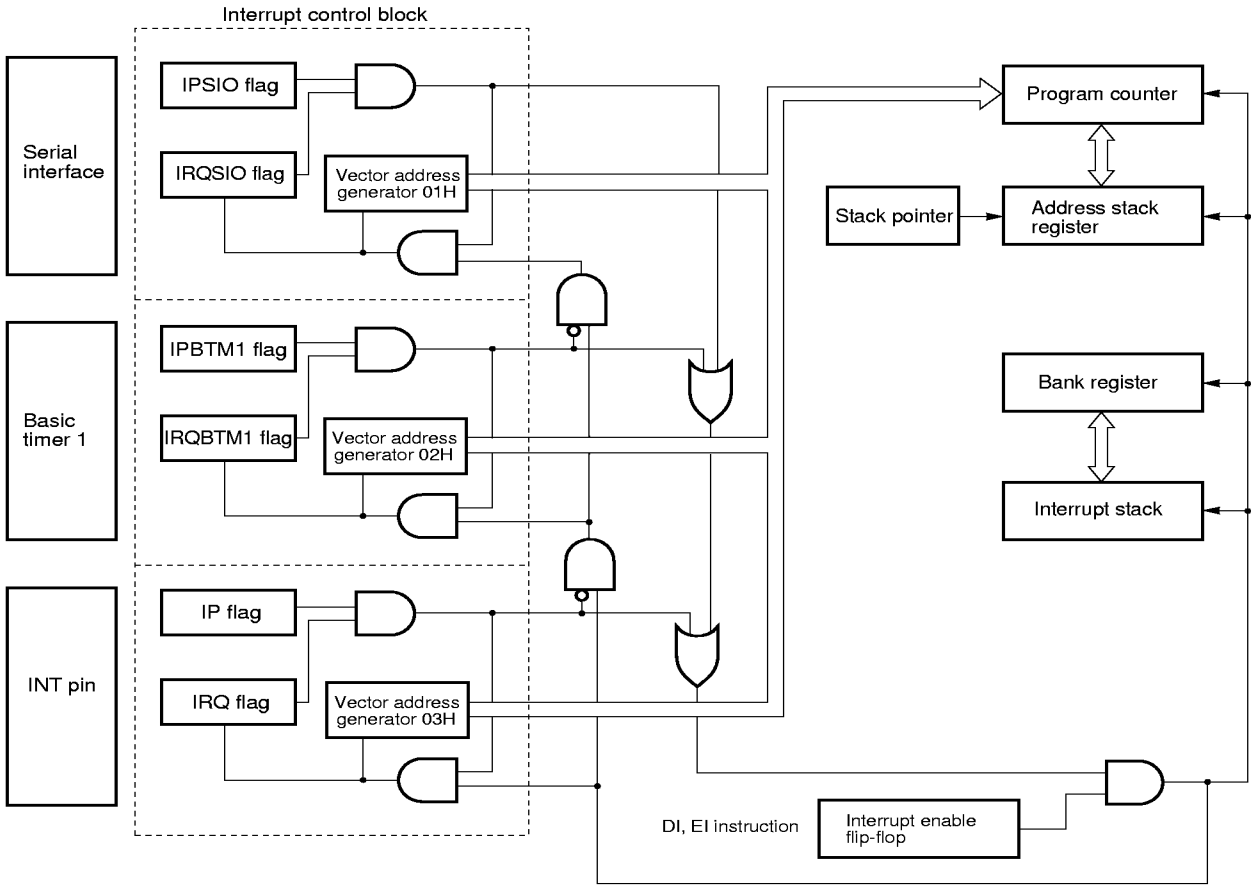
The interrupt block consists of "interrupt control blocks" that control interrupt requests output from the corresponding peripheral hardware, "interrupt enable flip-flop" that enables all the interrupts, "stack pointer" that is controlled when an interrupt is accepted, "address stack register", "program counter", and "interrupt stack".

The "interrupt control block" of each peripheral hardware consists of an "interrupt request flag (IRQxxx)" that detects each interrupt, "interrupt enable flag (IPxxx)" that enables each interrupt, and "vector address generator (VAG)" that specifies each vector address when an interrupt is accepted.

The following peripheral hardware have the interrupt functions:

- INT pin
- Basic timer 1
- Serial interface

Figure 11-1. Outline of Interrupt Block



11.2 Interrupt Control Block

An interrupt control block is available for each peripheral hardware. Each of these blocks detects the presence/absence of an interrupt request, enables/disables the interrupt, and generates a vector address when the interrupt is accepted.

11.2.1 Interrupt request flag (IRQxxx)

The interrupt request flags are set to (1) when an interrupt request has been issued from the corresponding peripheral hardware, and is cleared (0) when the interrupt has been accepted.

Therefore, even when the interrupt is not enabled, whether an interrupt request has been issued can be detected by checking these interrupt request flags.

Writing "1" directly to an interrupt request flag is equivalent to issuance of an interrupt request.

Once this flag has been set, it will not be cleared until the corresponding interrupt has been accepted, or "0" is written to the flag by an instruction.

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

The interrupt request flags are address 58H through 5AH of BANK1 of RAM.

Figures 11-2 through 11-4 show the configuration and functions of each interrupt request register.

Figure 11-2. Configuration of INT Pin Interrupt Request Register

Name	Flag symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
INT pin interrupt request register	0	0	0	IRQ	(BANK1) 58H	R/W

Sets interrupt request issuing status of INT pin.	
0	Interrupt request not issued.
1	Interrupt request issued.

Fixed to "0".

At reset	Power-ON	0	0	0	0
	Clock stop	↓	↓	↓	0
	CE	↓	↓	↓	0

Figure 11-3. Configuration of Basic Timer 1 Interrupt Request Register

Name	Flag symbol b_3	b_2	b_1	b_0	Address	Read/ Write
Basic timer 1 interrupt request register	0	0	0	IRQBTM1	(BANK1) 59H	R/W

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

Fixed to "0".

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

Figure 11-4. Configuration of Serial Interface Interrupt Request Register

Name	Flag symbol b_3	b_2	b_1	b_0	Address	Read/ Write
Serial interface interrupt request register	0	0	0	IRQSIO	(BANK1) 5AH	R/W

IRQSIO	Sets interrupt request issuing status of serial interface.
0	Interrupt request not issued.
1	Interrupt request issued.

Fixed to "0".

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

11.2.2 Interrupt enable flag (IPxxx)

Each interrupt enable flag enables or disables the interrupt request of the corresponding peripheral hardware. So that an interrupt is accepted, all the following three conditions must be satisfied:

- The interrupt must be enabled by the corresponding interrupt enable flag.
- The interrupt request must be issued from the corresponding interrupt request flag.
- The "EI" instruction (which enables all the interrupts) must be executed.

The interrupt enable flags are located on the interrupt enable registers on the register file.

Figure 11-5 shows the configuration and functions of the interrupt enable register.

Figure 11-5. Configuration of Interrupt Enable Register

Name	Flag symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Interrupt enable register	0	IPSIO	IPBTM1	IP	(BANK1) 57H	R/W

IP	
0	Disables interrupt
1	Enables interrupt

Enables interrupt from INT pin

IPBTM1	
0	Disables interrupt
1	Enables interrupt

Enables interrupt from basic timer 1

IPSIO	
0	Disables interrupt
1	Enables interrupt

Enables interrupt from serial interface

Fixed to 0

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

11.2.3 Vector address generator (VAG)

When an interrupt request from peripheral hardware has been accepted, the vector address generator generates a branch address (vector address) to which the program execution is to be branched.

The vector addresses corresponding to each interrupt source are listed in Table 11-1.

Table 11-1. Vector Address of Each Interrupt Source

Interrupt source	Vector address
INT pin	03H
Basic timer 1	02H
Serial interface	01H

11.3 Interrupt Stack Register

11.3.1 Configuration and functions of interrupt stack register

Figure 11-6 shows the configuration of the interrupt stack register.

The interrupt stack saves the contents of the bank registers when an interrupt has been accepted:

When an interrupt has been accepted, and the contents of bank registers have been saved to the interrupt stack, the contents of the registers are reset to "0".

The interrupt stack can save up to one level of the contents of the bank registers; therefore, multiplexed interrupt cannot be performed.

The contents of the interrupt stack register are restored to the respective system registers when an interrupt return ("RETI") instruction has been executed.

Caution With the μPD17073, the contents of the program status word (PSWORD) are not saved to the stack but retained when an interrupt is accepted. Therefore, the contents of the program status word must be backed up by software.

Figure 11-6. Configuration of Interrupt Stack Register

Interrupt stack register (INTSK)				
Name	Bank stack			
Bit	b_3	b_2	b_1	b_0
0H	–	–	–	

Remark –: Bit not saved

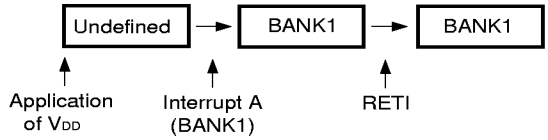
11.3.2 Operations of interrupt stack

Figure 11-7 illustrates the operations of the interrupt stack.

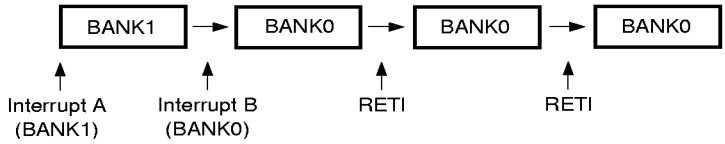
When multiplexed interrupts have been accepted, the first contents saved to the stack are popped. If these contents are necessary, therefore, they must be saved through program.

Figure 11-7. Operations of Interrupt Stack

(a) If interrupt level does not exceed 1



(b) If interrupt level exceeds 1



11.4 Stack Pointer, Address Stack Register, and Program Counter

The address stack register saves the return address to which the program execution is to restore when execution exits from an interrupt service routine.

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

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

Next, when dedicated instruction "RETI" has been executed after the interrupt service routine has been executed, the contents of the address stack register specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

Also refer to **3. ADDRESS STACK (ASK)**.

11.5 Interrupt Enable Flip-Flop (INTE)

The interrupt enable flip-flop enables all the interrupts.

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

This flip-flop is set or reset by dedicated instruction "EI (set)" or "DI (reset)".

The "EI" instruction sets this flip-flop when the instruction next to it has been executed, while the "DI" instruction resets the flip-flop while it is executed.

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

This flip-flop is reset on power-ON reset, execution of the clock stop instruction, or CE reset.

11.6 Accepting Interrupt

11.6.1 Interrupt accepting operation and priority

An interrupt is accepted in the following sequence:

(1) Each peripheral hardware issues an interrupt request signal to an interrupt request block when a certain condition is satisfied (for example, when a falling signal has been input to the INT pin).

(2) Each interrupt request block sets the corresponding interrupt request flag (e.g., IRQ flag for the INT pin) to "1" when it has received an interrupt request signal from peripheral hardware.

(3) When the interrupt request flag is set, the interrupt request block whose interrupt enable flag (e.g., IP flag for IRQ flag) is set to "1" outputs "1".

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

When the interrupt has been accepted, the output of the interrupt enable flip-flop is input to the block that has issued the interrupt request, through an AND circuit, as shown in Figure 11-1.

The signal input to the block that has issued the interrupt request clears the interrupt request flag of that block to "0", and a vector address corresponding to the interrupt is output.

If any of the blocks that have issued an interrupt request outputs "1" at this time, the interrupt accept signal is not transmitted to the next stage. When two or more interrupt request have generated at the same time, therefore, the interrupts are accepted according to the following priority:

INT pin > basic timer 1 > serial interface

The interrupt corresponding to an interrupt source is not accepted unless the interrupt enable flag is set to "1". Therefore, by clearing the interrupt enable flag to "0", an interrupt with a high hardware priority can be disabled.

11.6.2 Timing to accept interrupt

Figure 11-8 is a timing chart illustrating how interrupts are accepted.

(1) in this figure illustrate how one type of interrupt is accepted.

(a) in (1) indicates the case where the interrupt request flag is set to "1" last, while (b) shows the case where the interrupt enable flag is set to "1" last.

In either case, the interrupt is accepted after all the interrupt request flag, interrupt enable flip-flop, and interrupt enable flag have been set.

If it is during the first instruction cycle of the "MOVT DBF, @AR" instruction or an instruction with the skip condition satisfied that sets the last flag or flip-flop to "1", the interrupt is accepted during the second instruction cycle of the "MOVT DBF, @AR" instruction or when the skipped instruction ("NOP") has been executed.

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

(2) in Figure 11-8 shows the timing chart where two or more interrupts are used.

When using two or more interrupts, the interrupt given the highest hardware priority at that time is accepted if all the interrupt enable flags are set. However, the hardware priority can be changed by manipulating the interrupt enable flag through program.

"Interrupt cycle" in Figure 11-8 is a special cycle in which the interrupt request flag is clear, a vector address is specified, and the contents of the program counter are saved after an interrupt has been accepted, and lasts for 53.3 μs, (normal operation) or one instruction execution time.

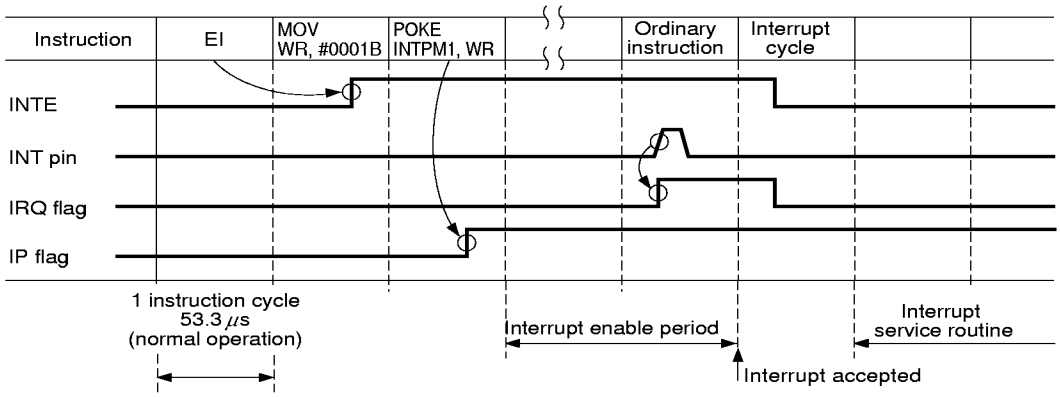
For details, refer to **11.7 Operations after Accepting Interrupt**.

Figure 11-8. Interrupt Accepting Timing Chart (1/2)

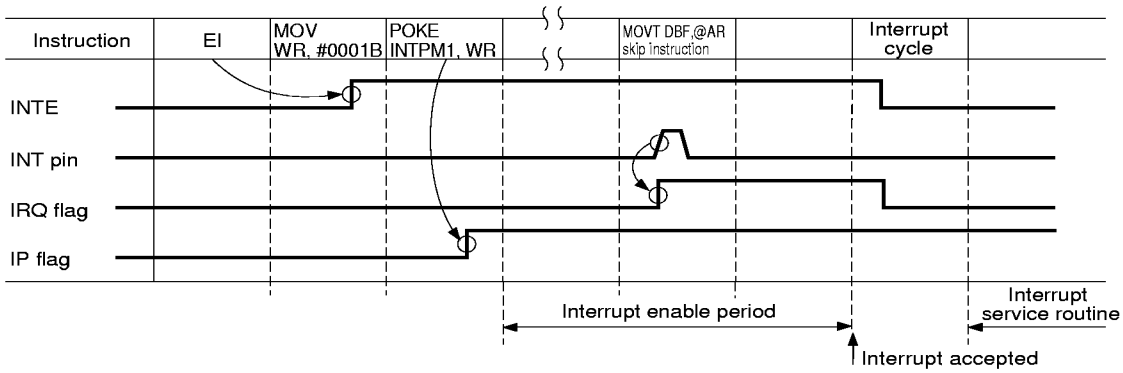
(1) When one type of interrupt (e.g., rising edge of INT pin) is used

(a) When there is no time to mask interrupt by interrupt enable flag (IPxxx)

<1> If an ordinary instruction which is not "MOVT" or does not satisfy the skip condition is executed when interrupt is accepted



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



(b) When there is interrupt pending time by interrupt enable flag

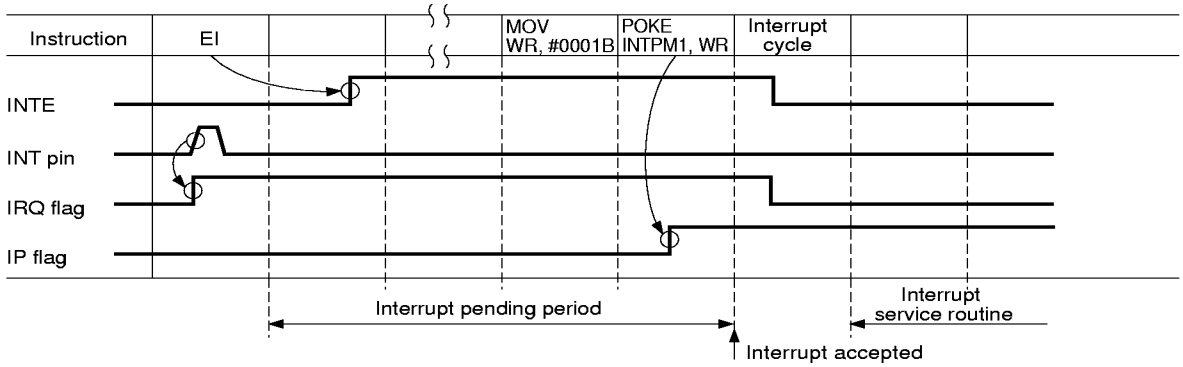
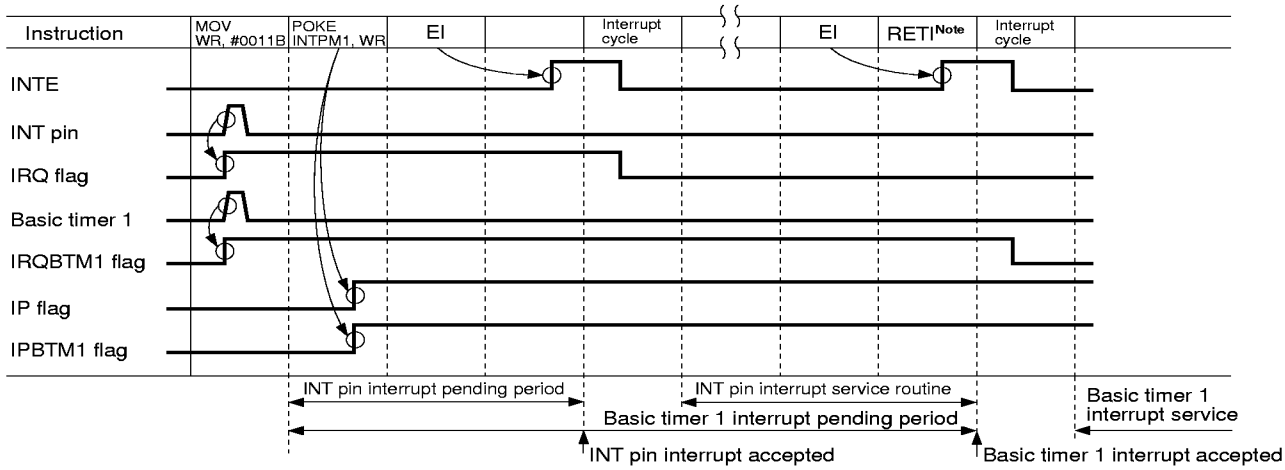


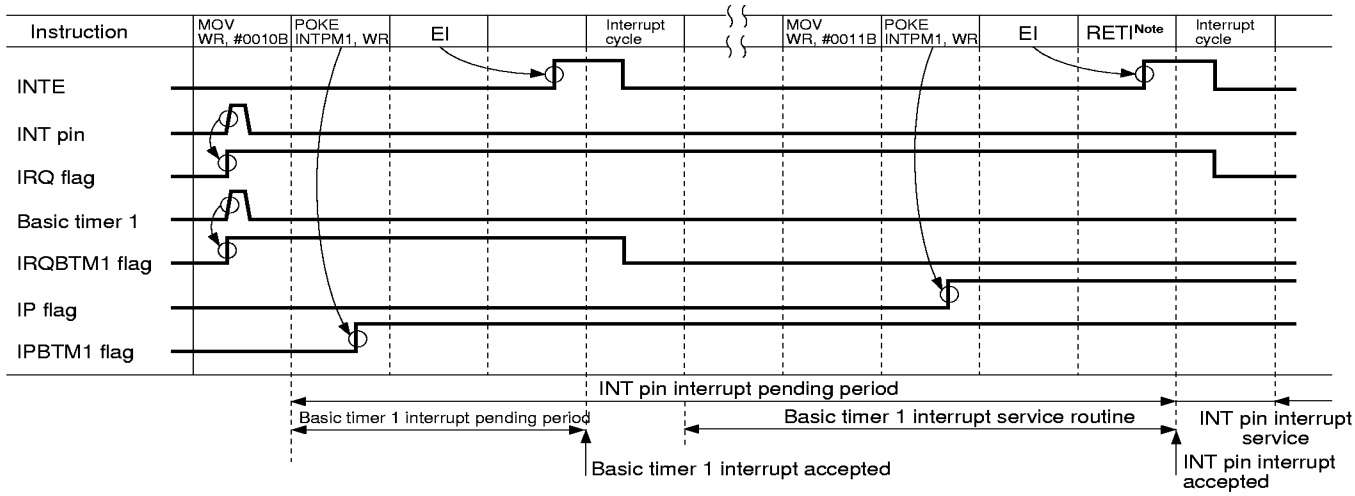
Figure 11-8. Interrupt Accepting Timing Chart (2/2)

(2) When two or more interrupts are used (e.g., INT pin and basic timer 1)

(a) Hardware priority



(b) Software priority



Note Because the level of the interrupt stack is 1, multiplexed interrupt cannot be performed.

11.7 Operations after Accepting Interrupt

When an interrupt has been accepted, the following processing is automatically executed in sequence:

(1) The interrupt enable flip-flop and the interrupt request flag corresponding to the accepted interrupt request are cleared to "0". Therefore, the interrupt is disabled.

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

(3) The contents of the program counter are saved to the address stack register specified by the stack pointer. At this time, the content of the program counter is the program memory address next to the one at which the interrupt has been accepted.
For example, if the interrupt has been accepted while a branch instruction is executed, the branch destination address is loaded to the program counter. If a subroutine call instruction is executed when the interrupt has been accepted, the address that called the subroutine is loaded to the program counter. When the skip condition of a skip instruction is satisfied, the next instruction is treated as a no-operation instruction ("NOP") and then the interrupt is accepted. Consequently, the contents of the program counter are the skipped address.

(4) The lower 1 bit of the bank register (BANK) is saved to the interrupt stack.

Caution At this time, the contents of the program status word (PSWORD) are not saved. Save the contents of the program status word by software as necessary.

(5) The contents of the vector address generator corresponding to the accepted interrupt are transferred to the program counter. Therefore, the execution branches to an interrupt service routine.

The above steps (1) through (5) are executed in one special instruction cycle (53.3 μs: normal operation) that does not involve execution of an ordinary instruction. This instruction cycle is called an interrupt cycle.

Therefore, it takes the CPU one instruction cycle to branch to the corresponding vector address after it has accepted an interrupt.

11.8 Exiting from Interrupt Service Routine

To return to the service that was executed when the interrupt was accepted from the interrupt service routine, a dedicated instruction "RETI" is used.

When this instruction is executed, the following processing is automatically executed in sequence:

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

(2) The contents of the interrupt stack are restored to the lower 1 bit of the bank register (BANK).

Caution If the contents of the program status word are saved in the program, its contents must be restored to the program status word at the same time.

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

The processing (1) through (3) above is executed in one instruction cycle during which the "RETI" instruction is executed.

The only difference between the "RETI" and subroutine return instructions "RET" and "RETSK" is the restore operation of each system register described in step (2) above.

11.9 External (INT Pin) Interrupts

11.9.1 Outline of external interrupts

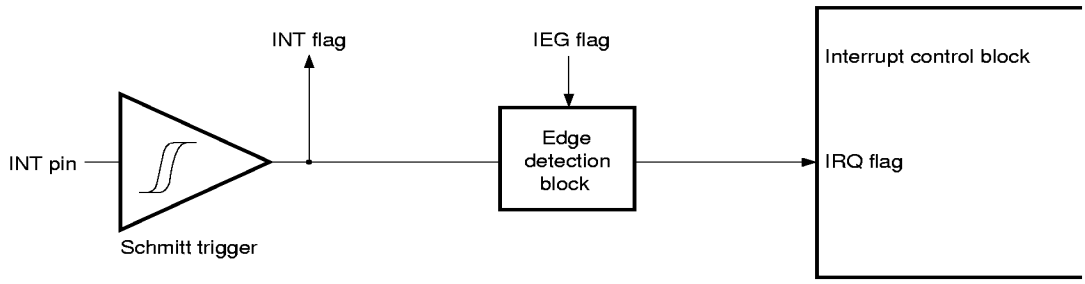
Figure 11-9 outlines the external interrupts.

As shown in this figure, an interrupt request for an external interrupt is issued at the rising or falling edge of the signal input to the INT pin.

Whether the interrupt request is to be issued at the rising or falling edge of INT is specified independently through program.

The INT pin is Schmitt trigger input pin to protect malfunctioning due to noise. This pin do not accept a pulse input that lasts for less than 100 ns.

Figure 11-9. Outline of External Interrupt



Remark INT: detects pin status
IEG: selects interrupt edge

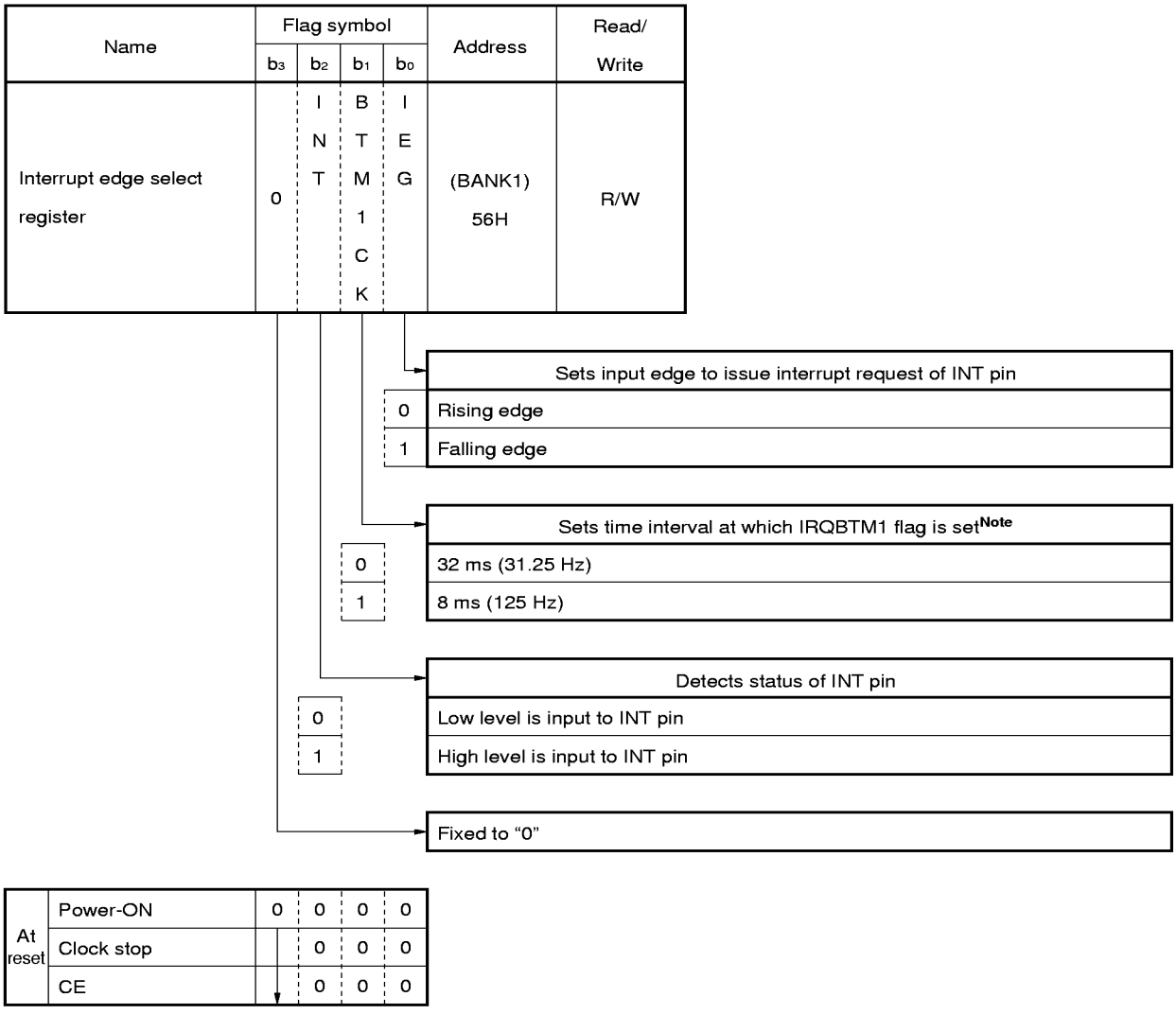
11.9.2 Edge Detection Block

The edge detection block specifies the edge (rising or falling edge) of the input signal that issues the external interrupt request of the INT pin, and detects the specified edge.

The edge is specified by IEG flag.

Figure 11-10 shows the configuration and function of the interrupt edge select register.

Figure 11-10. Configuration of Interrupt Edge Select Register



Note For the function of the BTM1CK flag, refer to **12.3.1 Outline of basic timer 1**.

Note that as soon as the interrupt request issuing edge is changed by the IEG flag, the interrupt request signal may be issued.

Suppose that the IEG flag is set to "1" (specifying the falling edge) and that a high level is input to the INT pin, as shown in Table 11-2. If the IEG flag is cleared at this time, the edge detector circuit judges that a rising edge has been input, and issues an interrupt request.

Table 11-2. Issuing Interrupt Request By Changing IEG Flag

Changes in IEG flag	INT pin status	Interrupt request	IRQ flag status
1 → 0 (falling)	Low level	Not issued	Retains previous status
	High level	Issued	Set to "1"
0 → 1 (rising)	Low level	Issued	Set to "1"
	High level	Not issued	Retains previous status

11.9.3 Interrupt control block

The level of a signal input to the INT pin can be detected by using the INT flag.

This flag is set or cleared independently of interrupts; therefore, it can be used as a 1-bit general-purpose input port when the interrupt function is not used.

The INT flag can also be used as a general-purpose port that can detect the rising or falling edge by reading an interrupt request flag if the interrupt corresponding to the flag is not enabled.

In this case, however, the interrupt request flag is not automatically cleared and must be cleared by program.

Also refer to **Figure 11-10**.

11.10 Internal Interrupt

Two internal interrupt sources, basic timer 1, and serial interface, are available.

11.10.1 Interrupt by basic timer 1

This interrupt request is issued at fixed time intervals.

For details, refer to **12. TIMER**.

11.10.2 Interrupt by serial interface

This interrupt request is issued when a serial output or serial input operation has been completed.

For details, refer to **14. SERIAL INTERFACE**.

12. TIMER

The timers are used to control the program execution time.

12.1 General

As shown in this figure, the μPD17013 is provided with the following two timers:

- Basic timer 0
- Basic timer 1

The basic timer 0 is used to detect the status of a flip-flop that is set at fixed time intervals.

The basic timer 1 is used to issue an interrupt request at fixed time intervals.

The basic timer 0 can also be used to detect a power failure. The clock of each timer is generated by dividing the system clock (75 kHz).

12.2 Basic Timer 0

12.2.1 General

Figure 12-1 outlines the basic timer 0.

The basic timer 0 is used as a timer by detecting the status of a flip-flop which is set at fixed time intervals, by using the BTM0CY flag (BANK1 of RAM: address 51H, bit 0).

The content of the flip-flop corresponds to the BTM0CY flag on a one-to-one basis.

The set time for BTM0CY flag (BTM0CY flag set pulse) is 125 ms (8 Hz).

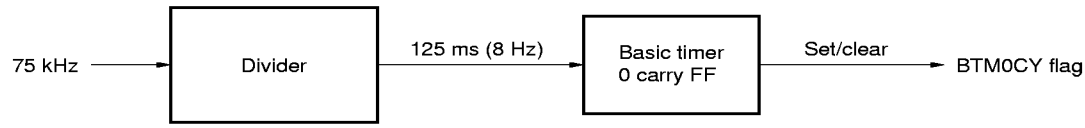
If the BTM0CY flag is read for the first time after power-ON reset, its content is always "0". After that, the flag is set to "1" at fixed time intervals.

If the CE pin goes high, CE reset is effected when the BTM0CY flag is set next time.

By reading the content of the BTM0CY flag at system reset (power-ON reset and CE reset), therefore, a power failure can be detected.

For details on power failure detection, refer to **20. RESET**.

Figure 12-1. Outline of Basic Timer 0



Remark BTM0CY (bit 0 of basic timer 0 carry register: refer to **Figure 12-2**) detects the status of the flip-flop.

12.2.2 Flip-flop and BTM0CY flag

The flip-flop is set at fixed time intervals and its status is detected by the BTM0CY flag of the basic timer 0 carry register.

The BTM0CY flag is a read-only flag, and is reset to "0" if its contents are read (Read & Reset) by using the instructions shown in Table 12-1.

The BTM0CY flag is reset to "0" at power-ON reset, and is set to "1" at CE reset and at CE reset after the clock stop instruction is executed. Therefore, this flag can be used as a power failure detection flag.

The BTM0CY flag is not set until its contents are read by the instruction shown in Table 12-1 after application of the supply voltage. Once a read instruction has been executed, this flag is set at fixed time intervals.

Figure 12-2 shows the configuration and function of the basic timer 0 carry register.

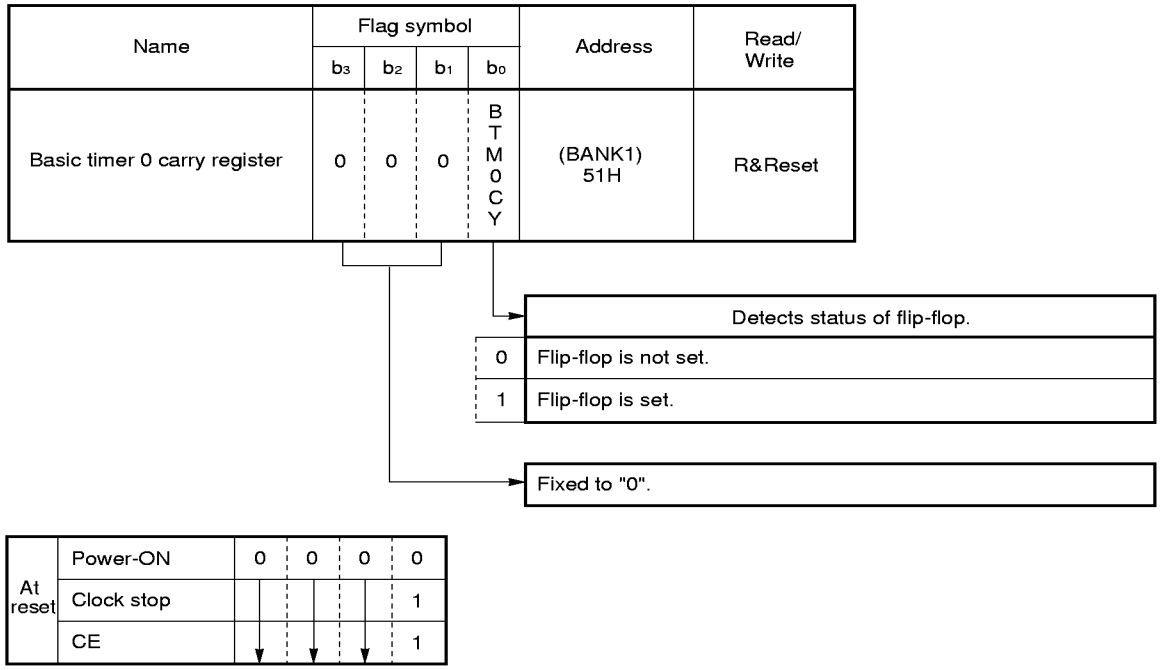
Table 12-1. Instructions to Reset BTM0CY Flag

Mnemonic	Operand	Mnemonic	Operand
ADD	m, #n4	ADD	r, m
ADDC		ADDC	
SUB		SUB	
SUBC		SUBC	
AND		AND	
OR		OR	
XOR		XOR	
SKE		LD	
SKEG		SKT	m, #n
SKLT		SKF	
SKNE		MOV	@r, m
			m, @r[Note]

Note When the row address of m is 5H and 1H is written to r.

Remark m = 51H

Figure 12-2. Configuration of Basic Timer 0 Carry Register



12.2.3 Application example of basic timer 0

An example of a program in which the basic timer 0 is used is shown below.

In this example, processing A is executed every 1 second.

Example

```
        M1      MEM 1.10H     ; 1-second counter, set to bank 1
LOOP:
        BANK1
        SKT1    BTM0CY        ; Branches to NEXT if BTM0CY flag is “0”
        BR      NEXT
        ADD     M1, #0010B    ; Adds 2 to M1
        SKT1    CY            ; Executes processing A if CY flag is “1”
        BR      NEXT          ; Branches to NEXT if CY flag is “0”
        Processing A

NEXT:
        Processing B
        BR      LOOP          ; Executes processing B and branches to LOOP
```

12.2.4 Error of basic timer 0

The time at which the BTM0CY flag is to be detected must be shorter than the time at which the BTM0CY flag is to be set (refer to **12.2.5 Notes on using basic timer 0**).

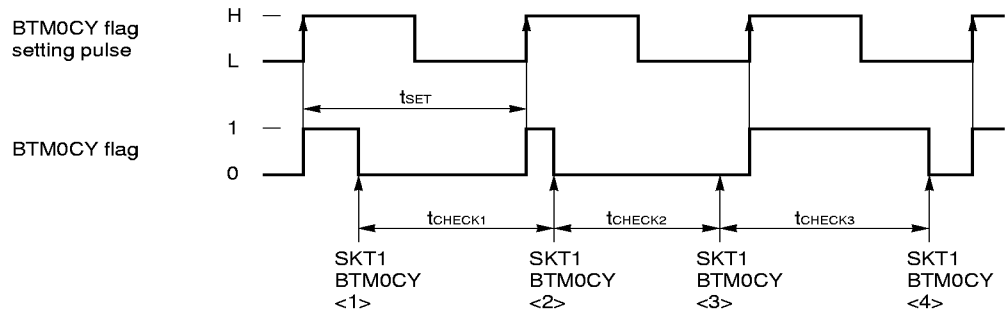
Where the time interval at which the BTM0CY flag is to be detected is t_{CHECK} and the time interval at which the BTM0CY flag is to be set (125 ms) is t_{SET}, the relation between t_{CHECK} and t_{SET} must be as follows:

$$t_{CHECK} < t_{SET}$$

At this time, as shown in Figure 12-3, the timer error when the BTM0CY flag is detected is:

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

Figure 12-3. Error of Basic Timer 0 due to Detection Time of BTM0CY Flag



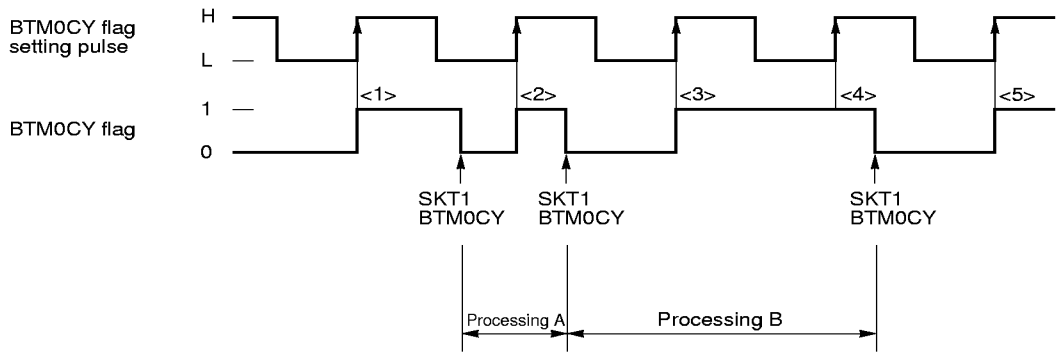
As shown in Figure 12-3, the timer is updated because the BTM0CY flag is detected as "1" in <2>.
In <3>, the flag is "0"; therefore, the timer is not updated until the BTM0CY flag is detected again in <4>.
At this time, the time of the timer is extended by t_{CHECK3}.

12.2.5 Notes on using basic timer 0

(1) BTM0CY flag detection time interval

The time interval at which the BTM0CY flag is to be detected must be shorter than the time interval at which the flag is to be set. This is because, if the time of processing B in Figure 12-4 is longer than the time interval at which the BTM0CY flag is to be set, the BTM0CY flag is not set accurately.

Figure 12-4. Detection of BTM0CY Flag and BTM0CY Flag



Because execution time of processing B is too long after BTM0CY flag, which has been set to "1" in step <2> above, has been detected, BTM0CY flag is not detected in step <3>.

(2) Sum of timer updating processing time and BTM0CY flag detection time interval

As described in (1) above, the time interval t_{CHECK} at which the BTM0CY flag is to be detected must be shorter than the time at which the BTM0CY flag is to be set.

At this time, even if the time interval at which the BTM0CY flag is to be detected is short, the timer processing may not be executed normally when CE reset is effected if the updating processing time of the timer is long. Therefore, the following conditions must be satisfied:

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

where,

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

t_{TIMER}: timer updating processing time

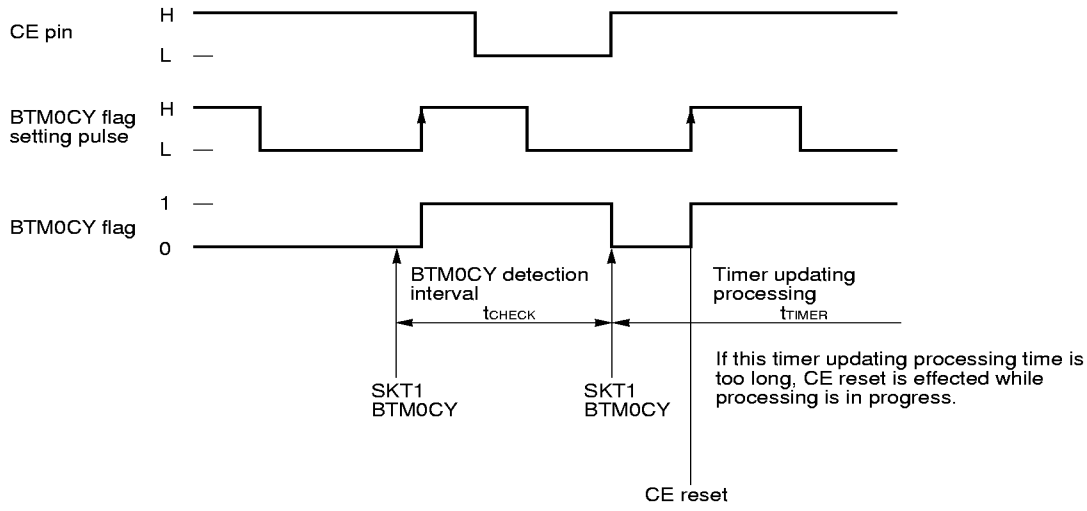
t_{SET}: time interval at which BTM0CY flag is set

An example is shown below.

Example Timer updating processing and BTM0CY flag detection time interval

```
BTIMER:
        BANK1
        SKT1  BTM0CY        ; Executes timer updating processing if BTM0CY flag is "1".
        BR    AAA           ; Branches to AAA if BTM0CY flag is "0".
        [ Timer updating ]
        BR    BTIMER
AAA:
        [ Processing A ]
BR      BTIMER
```

The following is the timing chart of the above program.



(3) Adjusting basic timer 0 at CE reset

An example of adjusting the basic timer 0 at CE reset is shown on the next page.

As shown in this example, the timer may have to be adjusted if the BTM0CY flag is used for power failure detection and, at the same time, the flag is used for a watch timer.

When the power is applied the first time (power-ON reset), the BTM0CY flag is cleared to "0", and not set until the contents of the flag is read again by an instruction shown in Table 12-1.

If the CE pin goes high, CE reset is effected in synchronization with rising edge of the BTM0CY flag setting pulse. At this time, the BTM0CY flag is set to "1" and starts.

Therefore, it can be judged, when system reset (power-ON reset or CE reset) has been effected, whether the system reset is power-ON reset or CE reset, by checking the status of the BTM0CY flag.

That is, if the BTM0CY flag is "0", power-ON reset has been effected; if the flag is "1", CE reset has been effected (for power failure detection).

At this time, the watch timer must continue its operation even when CE reset has been effected.

However, because the BTM0CY flag is cleared to "0" as a result of reading the BTM0CY flag to detect a power failure, the set (1) status of the BTM0CY flag is overlooked once.

Consequently, it is necessary to update the watch timer if CE reset has been detected as a result of power failure detection.

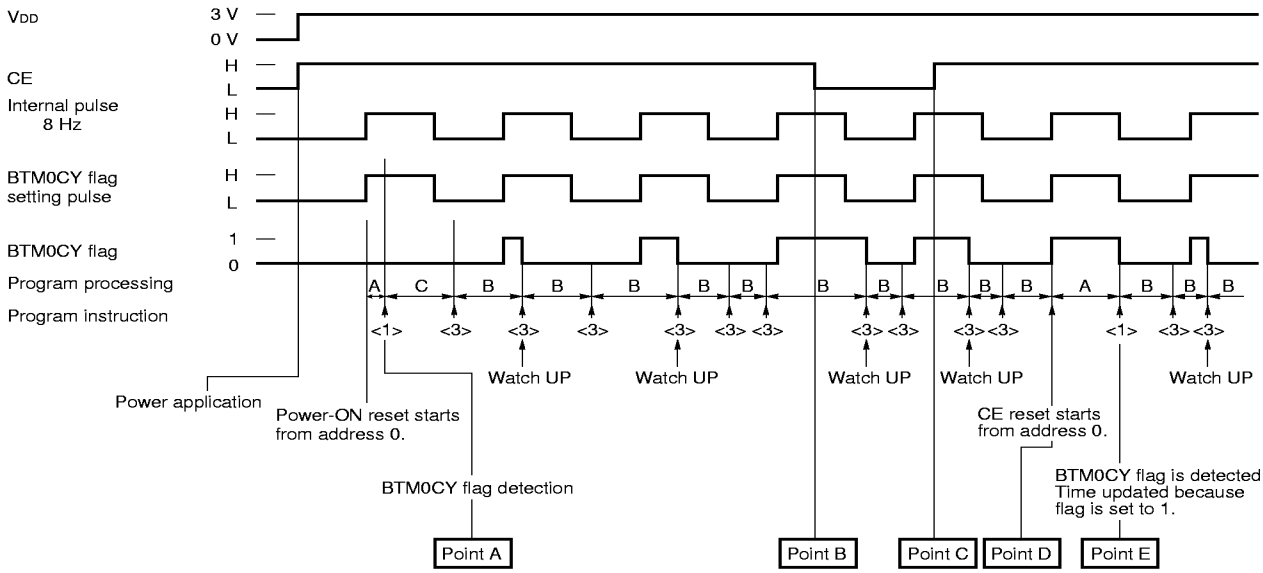
For details on power failure detection, refer to **20.6 Power Failure Detection**.

Example Adjusting timer at CE reset (to detect power failure and update watch by BTM0CY flag)

```
START:                          ; Program address 0000H
        Processing A
      ; <1>
        BANK1
        SKT1    BTM0CY          ; Embedded macro
                                ; Tests BTM0CY flag.
        BR      INITIAL         ; If BTM0CY is "0", branches to INITIAL (power failure detection).
BACKUP:
      ; <2>
        Updates watch 125 ms.   ; Adjusts watch because this is back up (CE reset)
LOOP:
      ; <3>
        Processing B            ; While performing processing B,
        SKF1    BTM0CY          ; tests BTM0CY flag and updates watch.
        BR      BACKUP
        BR      LOOP
INITIAL:
        Processing C            ; Initialization of ports and peripheral hardware.
        BR      LOOP
```

Figure 12-5 shows the timing chart of the above program.

Figure 12-5. Timing Chart



As shown in Figure 12-5, the program is started from address 0000H in synchronization with the rising of the internal 8-Hz pulse when supply voltage V_{DD} is applied first.

When the BTM0CY flag is detected next at point A, the BTM0CY flag is cleared to 0 because power has been just applied. It is therefore judged that a power failure (i.e., power-ON reset) has been detected, and "processing C" is executed.

Because the content of the BTM0CY flag has been read once at point A, the BTM0CY flag is set to 1 every 125 ms afterward.

Next, even if the CE pin goes low at point B and goes high at point C, the program executes "processing B" and increments the watch, unless the clock stop instruction is executed.

Because the CE pin goes high at point C, CE reset is effected at point D where the BTM0CY flag setting pulse rises, and the program is started from address 0000H.

At this time, if the BTM0CY flag is detected at point E, it is judged that back up (CE reset) has been effected, because the BTM0CY flag is set to 1.

As is evident from the above figure, the watch is delayed by 125 ms each time CE reset is effected, unless the watch is updated 125 ms at point E.

If processing A takes 125 ms or longer when a power failure is detected at point E, setting of the BTM0CY flag is overlooked two times; therefore, processing A must be completed within 125 ms.

Therefore, the BTM0CY flag must be detected for a power failure detection within the BTM0CY flag setting time after the program has been started from the address 0000H.

(4) If detection of BTM0CY flag overlaps with CE reset

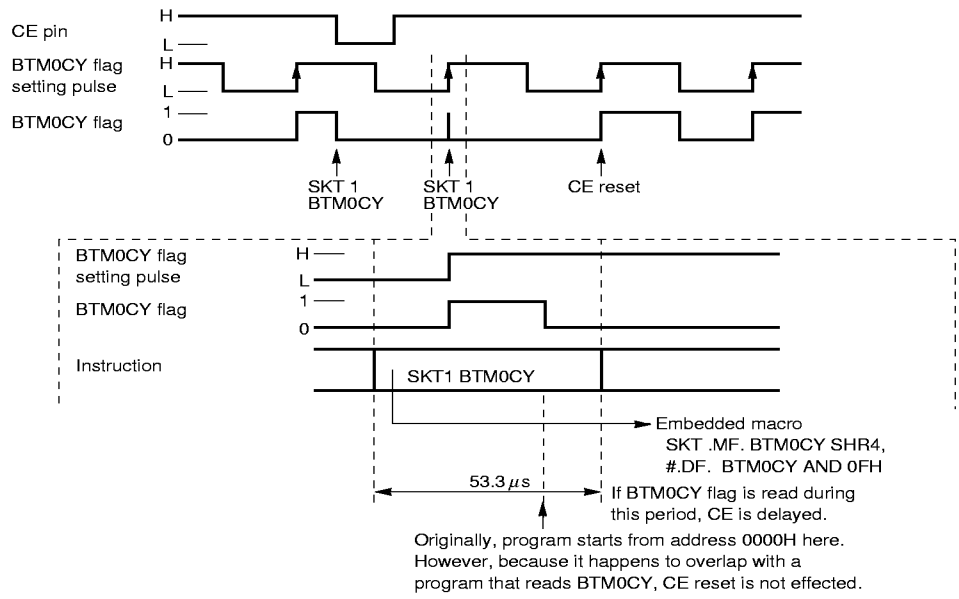
As described in (3), the CE reset is effected as soon as the BTM0CY flag has been set to 1.

If the BTM0CY flag read instruction happens to be executed at the same time as the CE reset, the BTM0CY flag read instruction takes precedence.

Therefore, if setting of the BTM0CY flag after the CE pin has gone high overlaps with the BTM0CY flag read instruction, the CE reset is effected when "the BTM0CY flag is set next time".

This operation is illustrated in Figure 12-6.

Figure 12-6. Operation when CE Reset and BTM0CY Flag Read Instruction Overlap



In a program that cyclically detects the BTM0CY flag, in which the BTM0CY flag detection time interval coincides with the BTM0CY flag setting time, CE reset is never effected.

12.3 Basic Timer 1

12.3.1 General

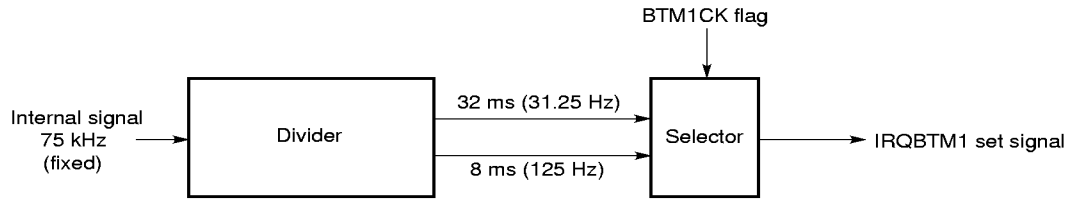
Figure 12-7 outlines the basic timer 1.

The basic timer 1 issues an interrupt request at fixed time interval and sets the IRQBTM1 flag to 1.

The time interval of the IRQBTM1 flag is set by the BTM1CK flag of the interrupt edge select register. Figure 12-8 shows the configuration and function of the interrupt edge select register.

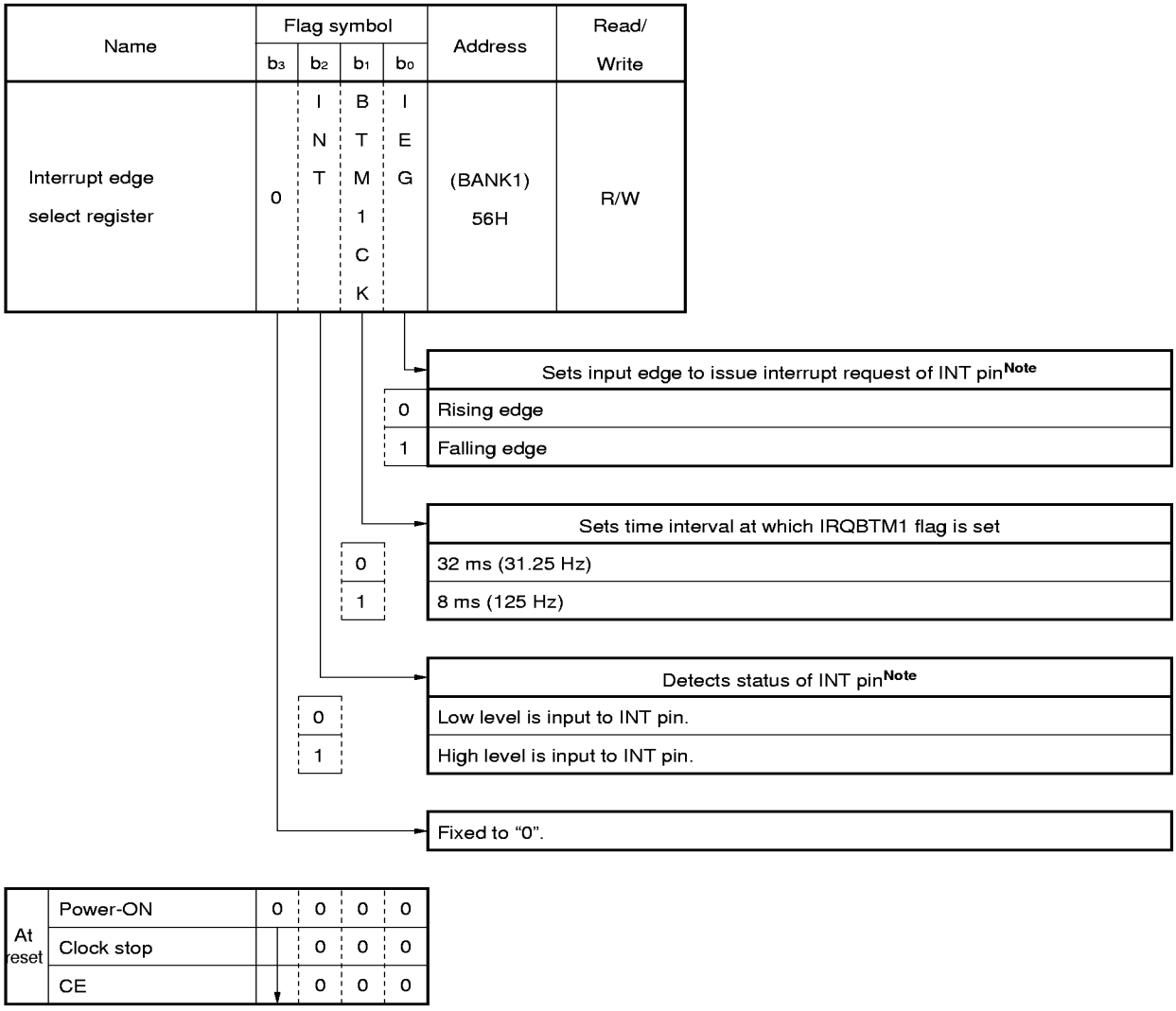
The interrupt generated by the basic timer 1 is accepted when the IRQBTM1 flag is set, if the EI instruction has been issued and the IPBTM1 flag has been set (refer to **11. INTERRUPT**).

Figure 12-7. Outline of Basic Timer 1



Remark BTM1CK (bit 1 of interrupt edge select register. Refer to **Figure 12-8**) set the time interval at which the IRQBTM1 flag is set.

Figure 12-8. Configuration of Interrupt Edge Select Register



Note For the functions of IEG and INT flags, refer to **11.9 External (INT pin) Interrupt**.

12.3.2 Application example of basic timer 1

A program example is shown below.

Example

```
M1          MEM     0.10H       ; 80-ms counter
BTIMER1     DAT     0002H       ; Symbol definition of basic timer 1 interrupt vector address

            BR      START       ; Branches to START
ORG         BTIMER1             ; Program address (0002H)
            ADD     M1, #0001B  ; Adds 1 to M1
            SKT1    CY          ; Tests CY flag
            BR      EI_RETI     ; Returns if no carry
            MOV     M1, #0110B
            Processing A
EI_RETI:
            EI
            RETI
START:
            MOV     M1, #0110B  ; Initializes contents of M1 to 6
            BANK1
            SET1    BTM1CK      ; Embedded macro
                                ; Sets basic timer 1 interrupt pulse to 8 ms
            SET1    IPBTM1      ; Enables basic timer 1 interrupt
            EI                  ; Enables all interrupts
LOOP:
            BANK0
            Processing B
            BR      LOOP
```

This program executes processing A every 80 ms.

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

This means that the interrupt is accepted even if execution exits from an interrupt service routine by execution of the “RETI” instruction, if processing A takes longer than 8 ms.

Consequently, processing B is not executed.

12.3.3 Error of basic timer 1

As described in 12.3.2, the interrupt generated by basic timer 1 is accepted each time the basic timer 1 interrupt pulse falls, if the EI instruction has been executed, and if the interrupt has been enabled.

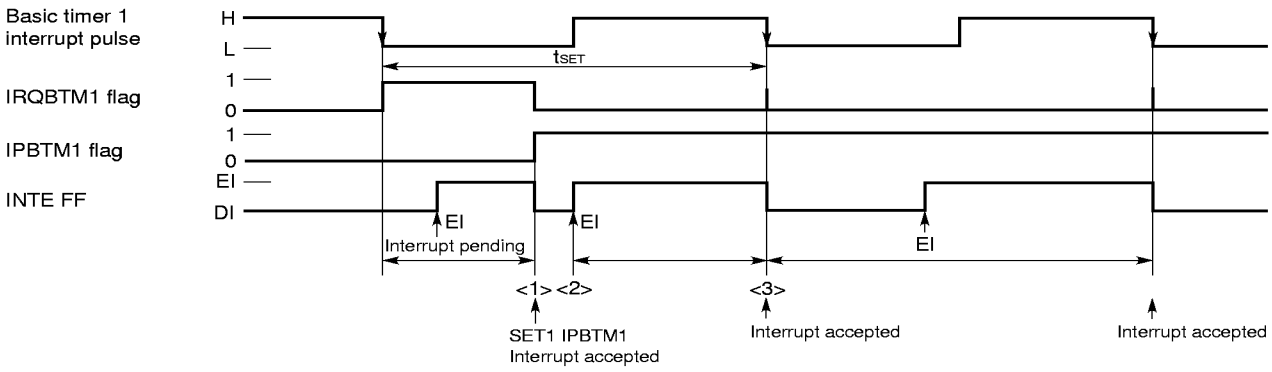
Therefore, an error of basic timer 1 occurs only when any of the following operations are performed:

- When the first interrupt after basic timer 1 interrupt has been enabled has been accepted
- When the time interval at which the IRQBTM1 flag is to be set is changed, i.e., when the first interrupt is accepted after the interrupt pulse has been changed
- When data has been written to the IRQBTM1 flag

Figure 12-9 shows an error in each of the above operations.

Figure 12-9. Error of Basic Timer 1 (1/2)

(a) When interrupt by basic timer 1 is enabled



At point <1> in the above figure, the interrupt by basic timer 1 is accepted as soon as the interrupt is enabled.

At this time, the error is $-t_{SET}$.

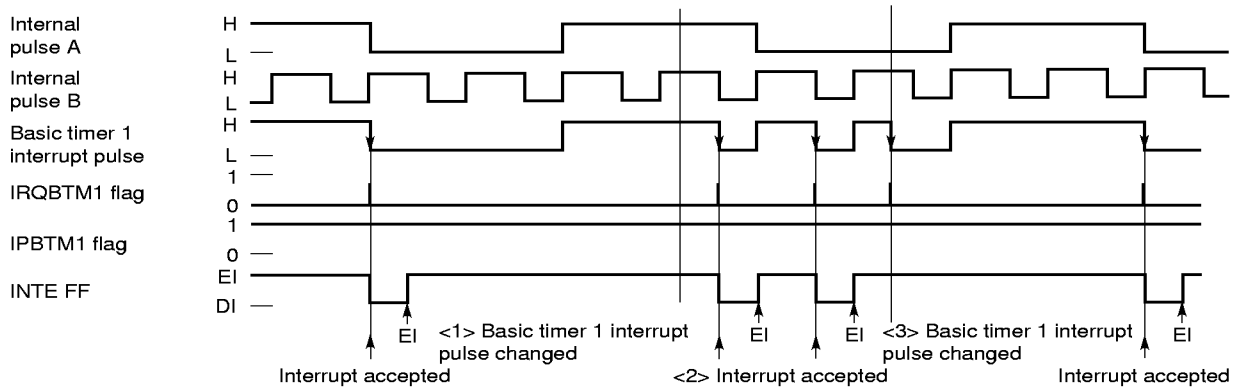
If an interrupt is enabled by the "EI" instruction at the next point <3>, the interrupt occurs at the falling edge of the basic timer 1 interrupt pulse.

At this time, the error is:

$-t_{SET}$ < error < 0

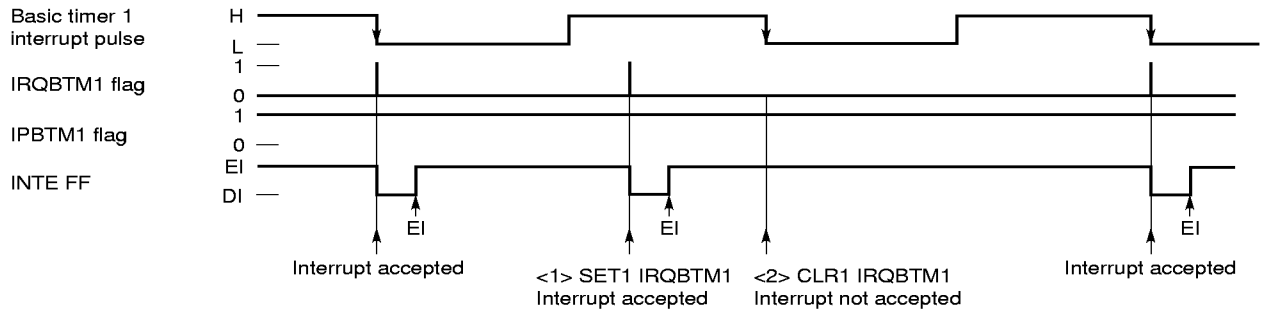
Figure 12-9. Error of Basic Timer 1 (2/2)

(b) When basic timer 1 interrupt pulse is changed



Even if the basic timer 1 interrupt pulse is changed to B at point <1> in the above figure, the interrupt is accepted at the next point <2> because the basic timer 1 interrupt pulse does not fall.
If the basic timer 1 interrupt pulse is changed to A at <3>, the interrupt is immediately accepted because the basic timer 1 interrupt pulse falls.

(c) When IRQBTM1 flag is manipulated



The interrupt is immediately accepted if the IRQBTM1 flag is set to 1 at <1>.
If clearing the IRQBTM1 flag to 0 overlaps with the falling of the basic timer 1 interrupt pulse at <2>, the interrupt is not accepted.

12.3.4 Notes on using basic timer 1

When creating a program, such as a program for watch, in which processing is always performed at fixed time intervals by using the basic timer 1 after the supply voltage has been once applied (power-ON reset), the basic timer 1 interrupt service must be completed in a fixed time.

Let's take the following example:

Example

```
M1          MEM       0.10H          ; 80-ms counter
BTIMER1     DAT       0002H          ; Symbol definition of interrupt vector address of basic timer 1

            BR        START          ; Branches to START
ORG         BTIMER1                  ; Program address (0002H)
            ADD       M1, #0001B     ; Adds 1 to M1
            SKT1      CY             ; Watch processing if carry occurs
            BR        EI_RETI        ; Restores if no carry occurs
            MOV       M1, #0110B
        ; <1>
            Processing B
EI_RETI:
            EI
            RETI
START:
            MOV       M1, #0110B     ; Initializes contetns of M1 to 6
            BANK1
            SET1      BTM1CK
                                     ; Embedded macro
                                     ; Sets time of interrupt by basic timer 1 to 8 ms
            SET1      IPBTM1         ; Embedded macro
                                     ; Enables interrupt by basic timer 1
            EI                       ; Enables all interrupts
LOOP:
            Processing A
            BR        LOOP
```

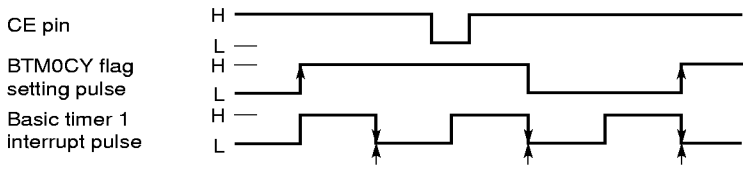
In this example, processing B is executed every 80 ms while processing A is executed.

If the CE pin goes high as shown in Figure 12-10, CE reset is effected in synchronization with the rising of the BTM0CY flag setting pulse.

If issuance of an interrupt request by the basic timer 1 happens to overlap with the setting of the BTM0CY flag at this time, CE reset takes precedence.

When CE reset is effected, the basic timer 1 interrupt request (IRQBTM1) flag is cleared. Consequently, the timer processing is skipped once.

Figure 12-10. Timing Chart



13. A/D CONVERTER

13.1 General

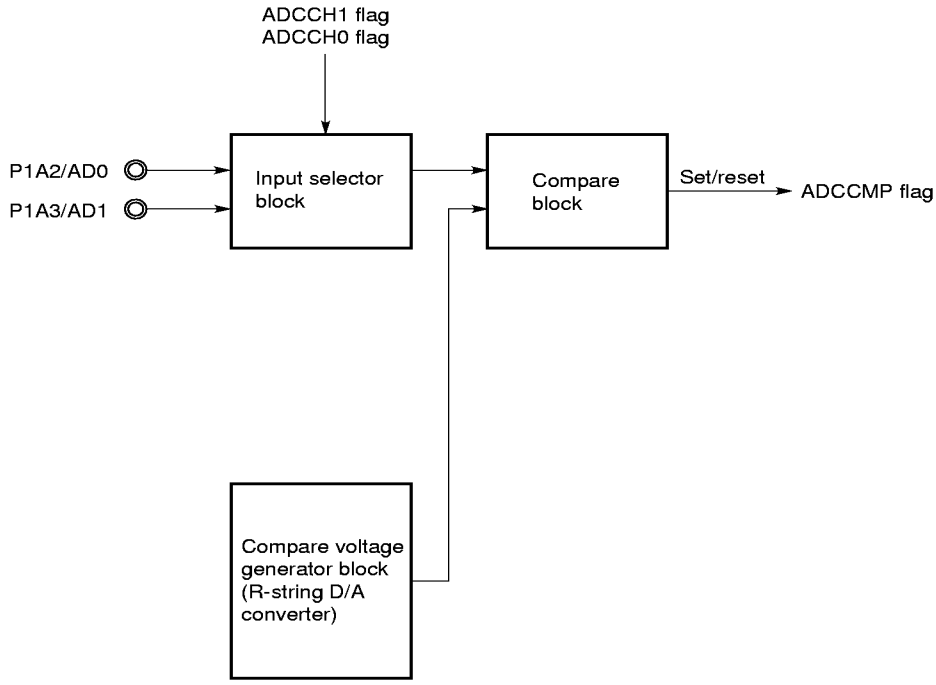
Figure 13-1 outlines the A/D converter.

The A/D converter compares an analog voltage input to the AD0 or AD1 pins with the internal compare voltage, judge the comparison result via software, and converts the analog signal into a 4-bit digital signal.

The comparison result can be detected by the ADCCMP flag.

As the comparison method, successive approximation is employed.

Figure 13-1. Outline of A/D Converter



Remarks 1. ADCCH0 and ADCCH1 (bits 0 and 1 of A/D converter channel select register. Refer to **Figure 13-4**) select the pin used for the A/D converter.

2. ADCCMP (bit 0 of A/D converter compare result detection register. Refer to **Figure 13-7**) detects the result of comparison.

13.2 Setting A/D Converter Power Supply

The μPD17073 has a power supply for the A/D converter. This power supply is also used for LCD display.

When using the A/D converter, therefore, the A/D converter power supply must be set to ON by using the ADCON flag of the LCD driver display start register.

Figure 13-2 shows the configuration and function of the LCD driver display start register.

Figure 13-2. Configuration of LCD Driver Display Start Register

Name	Flag symbol b_3	b_2	b_1	b_0	Address	Read/Write
LCD driver display start register	0	0	ADCON	LCDEN	(BANK1) 50H	R/W

b_1 (ADCON)	b_0 (LCDEN)	Turns ON/OFF A/D converter power supply and all LCD displays
0	0	A/D converter power supply OFF, LCD display OFF
0	1	A/D converter power supply ON, LCD display ON
1	0	A/D converter power supply ON, LCD display OFF
1	1	A/D converter power supply ON, LCD display ON

Fixed to "0".

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

Remark R: Retained

Cautions 1. When the LCD display is ON (LCDEN = 1), the A/D converter power supply is ON regardless of the setting of the ADCON flag.

2. Bit 3 of the LCD driver display start register is a test mode area. Therefore, do not write "1" to this bit.

13.3 Input Selector Block

Figure 13-3 shows the configuration of the input selector block.

The input selector block selects the pin to be used by using the A/D converter channel select register.

Two or more pins cannot be used at the same time with the A/D converter.

Figure 13-4 shows the configuration and function of the A/D converter channel select register.

For the configuration and function of the port 1A pull-down resistor select register, refer to **Figure 10-1 Port 1A Pull-Down Resistor Select Register**.

Figure 13-3. Configuration of Input Selector Block

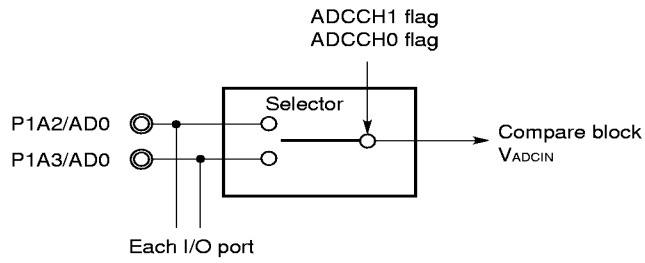
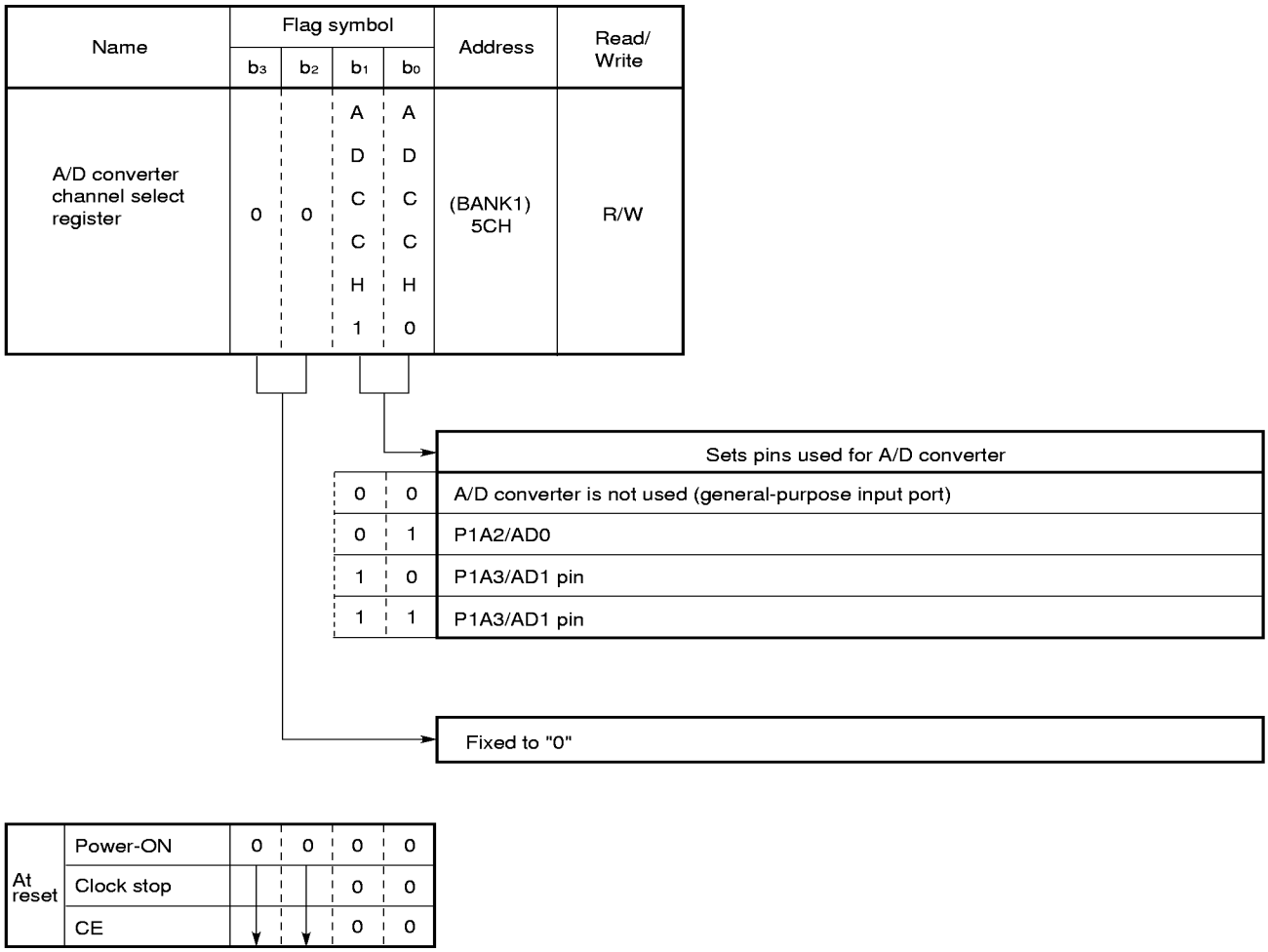


Figure 13-4. Configuration of A/D Converter Channel Select Register



13.4 Compare Voltage Generator Block and Compare Block

Figure 13-5 shows the configuration of the compare voltage generator block and compare block.

The compare voltage generator block switches over the tap decoder by using 4-bit data set to the A/D converter reference voltage setting register to generate 16 steps of compare voltage V_{REF}.

In other words, this block is an R-string D/A converter.

The power source of the R string is the same as V_{DD} that is supplied to the device.

The compare block judges which of the voltage V_{ADCIN} input from a pin and compare voltage V_{REF} is greater.

Data is compared by the comparator as soon as the ADCSTRT flag has been written to. One compare time of the A/D converter is equal to two instruction execution times (106.6 μs in normal operation mode, and 213.2 μs in the low-speed mode).

By reading the content of the ADCSTRT flag, the current operating status of the comparator can be checked.

The compare result is detected by the ADCCMP flag.

Figure 13-6 shows the configuration and function of the A/D converter compare start register.

Figures 13-7 and 13-8 show the configuration and function of the A/D converter compare result detection register and A/D converter reference voltage setting register. Table 13-1 lists the compare voltages.

Figure 13-5. Configuration of Compare Voltage Generator Block and Compare Block

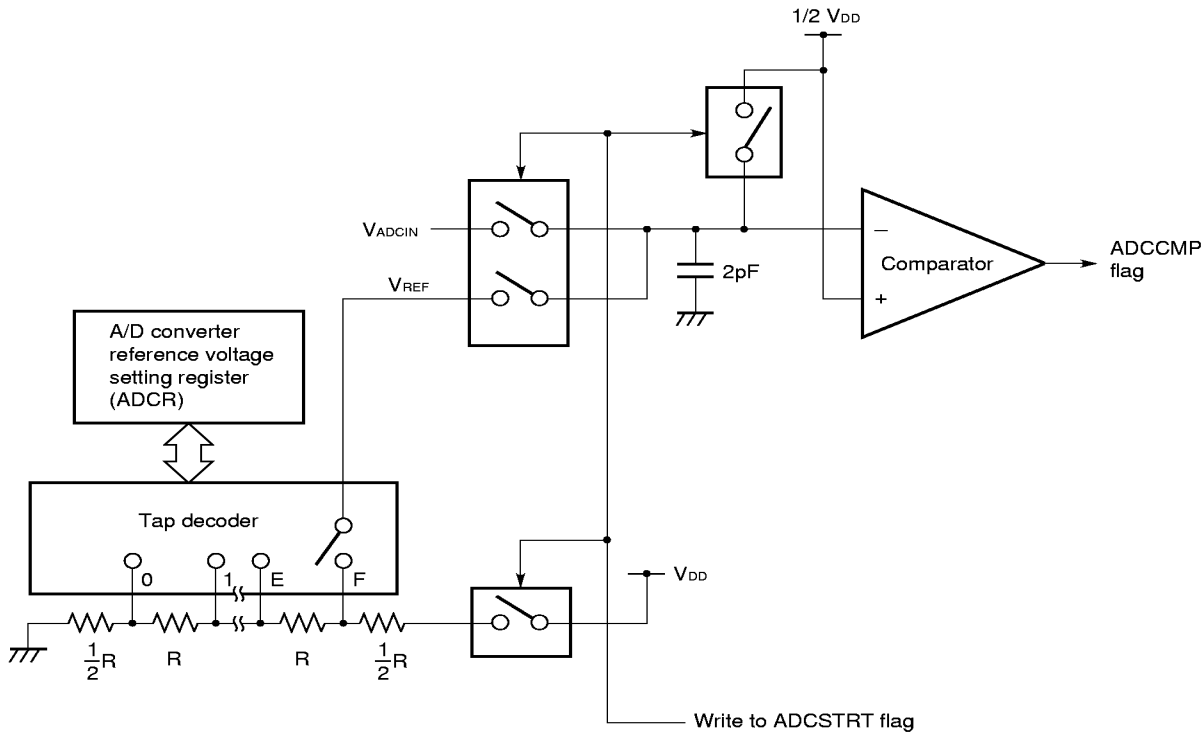
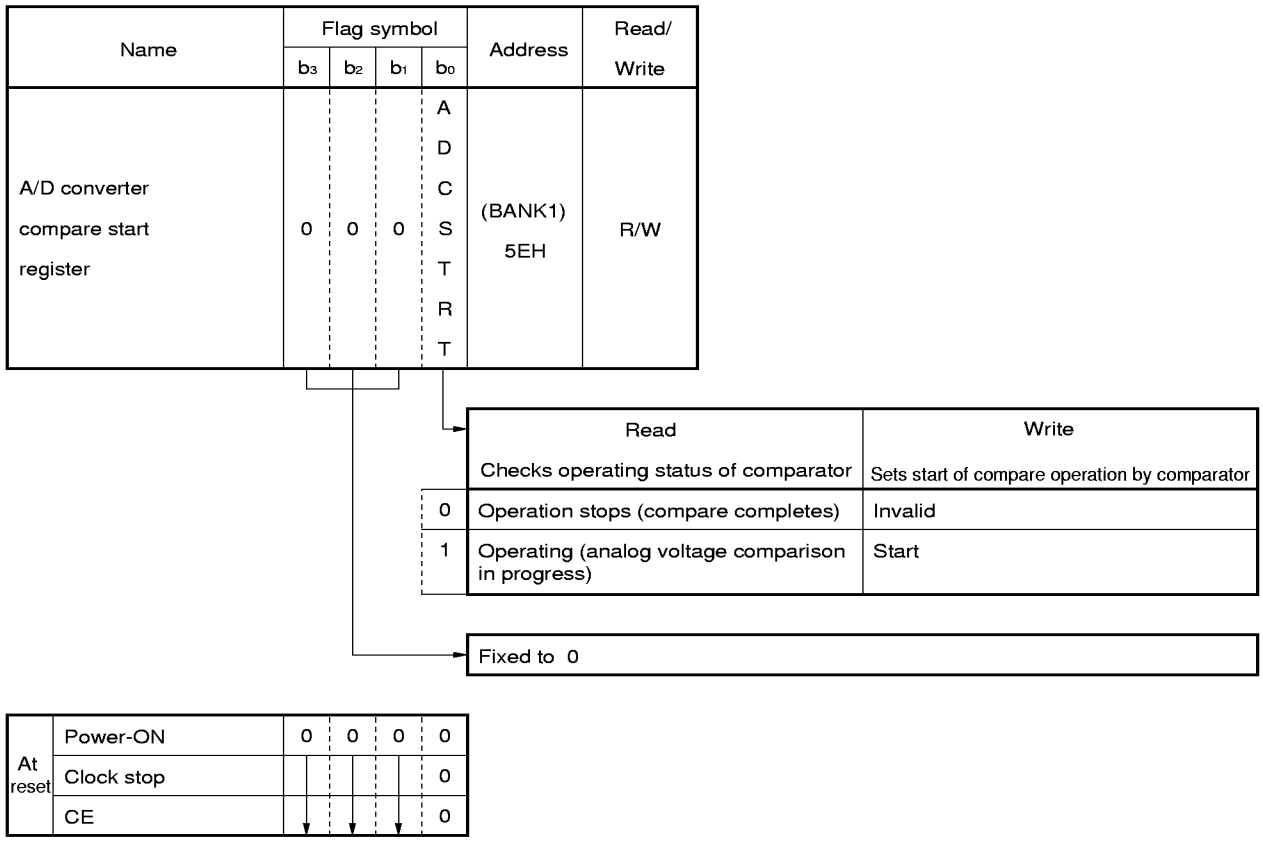


Figure 13-6. Configuration of A/D Converter Compare Start Register



Remarks 1. Even if the A/D converter channel select register or A/D converter reference voltage setting register is manipulated when ADCSTRT = 1 (when comparison by the comparator is in progress), the contents of the register remain unchanged. Therefore, the operating status of the A/D converter cannot be changed while the comparator is operating.

2. The ADCSTRT flag is cleared to "0" only when the voltage comparison operation by the comparator is completed or when the "STOP s" instruction is executed.

Figure 13-7. Configuration of A/D Converter Compare Result Detection Register

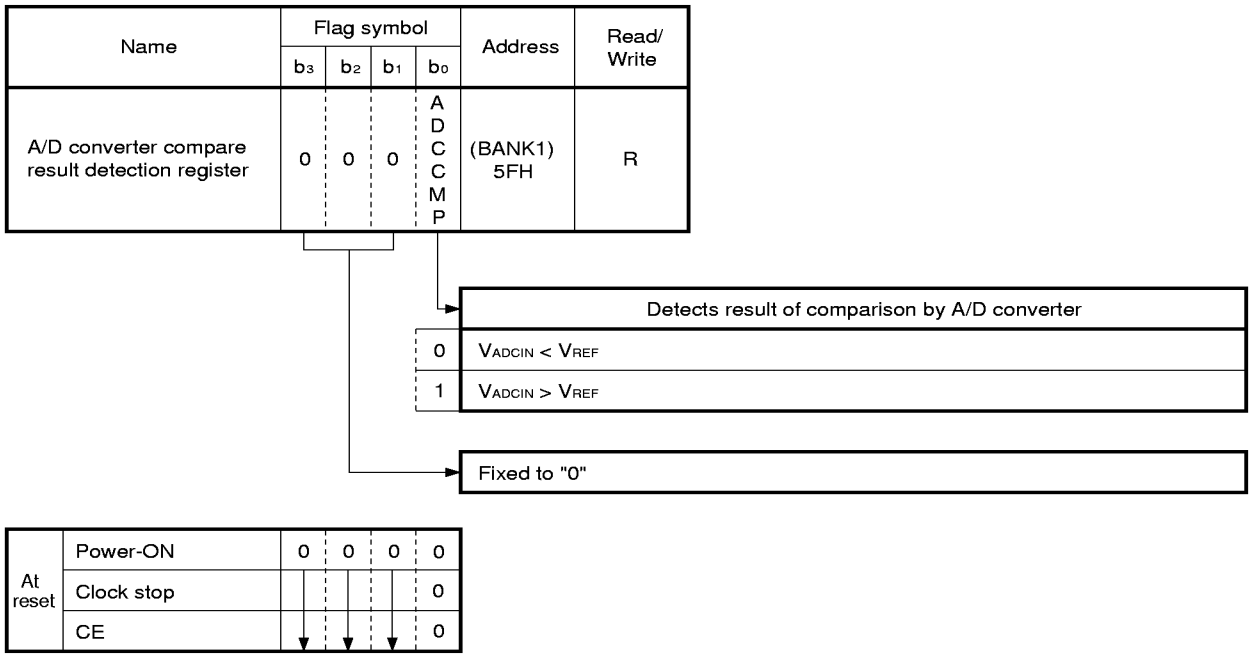


Figure 13-8. Configuration of A/D Converter Reference Voltage Setting Register

Name	Flag symbol b_3	b_2	b_1	b_0	Address	Read/Write
A/D converter reference voltage setting register	ADCRFSEL3	ADCRFSEL2	ADCRFSEL1	ADCRFSEL0	(BANK1) 5DH	R/W

	Sets compare voltage of A/D converter
0	$V_{REF} = \frac{x + 0.5}{16} \times V_{DD}$ (V)
|	
x	
|	
0FH	

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

Table 13-1. Set Value of A/D Converter Reference Voltage Setting Register and Compare Voltage

A/D Converter reference voltage setting register set data		Compare voltage	
Decimal (DEC)	Hexadecimal (HEX)	Logic voltage Unit: × V_{DD} V	At V_{DD} = 3 V Unit: V
0	00H	0.5/16	0.094
1	01H	1.5/16	0.281
2	02H	2.5/16	0.469
3	03H	3.5/16	0.656
4	04H	4.5/16	0.844
5	05H	5.5/16	1.031
6	06H	6.5/16	1.219
7	07H	7.5/16	1.406
8	08H	8.5/16	1.594
9	09H	9.5/16	1.781
10	0AH	10.5/16	1.969
11	0BH	11.5/16	2.156
12	0CH	12.5/16	2.344
13	0DH	13.5/16	2.531
14	0EH	14.5/16	2.719
15	0FH	15.5/16	2.906

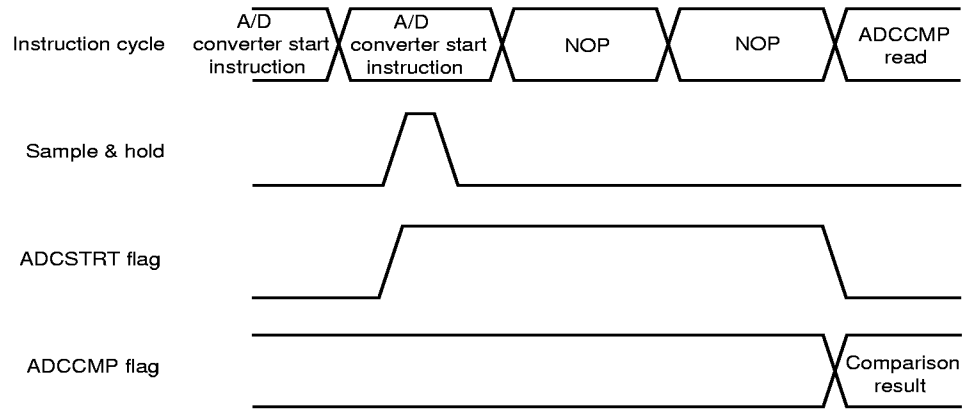
13.5 Comparison Timing Chart

The ADCEN flag is automatically cleared to 0 when the comparison operation has been completed.

The ADCSTRT flag is reset to 0 two instructions after the ADCSTRT flag has been set. At this point, the compare result (ADCCMP flag) can be read.

Figure 13-9 shows the timing chart.

Figure 13-9. Timing Chart of A/D Converter's Compare Operation



13.6 Performance of A/D Converter

Table 13-2 shows the performances of the A/D converter.

Table 13-2. Performances of A/D Converter

Parameter	Performance
Resolution	4 bits
Input voltage range	0-V_{DD}
Quantization error	±1/2 LSB
Over range	15.5/16 × V_{DD}
Error of offset, gain, and non-linearity	±3/2 LSB[Note]

Note Including quantization error

13.7 Using A/D Converter

13.7.1 Comparing one reference voltage

The following shows a program example.

Example To compare voltage input to AD0 pin, V_{ADCIN} against reference voltage V_{REF} (8.5/16 V_{DD}). If $V_{ADCIN} > V_{REF}$, execution branches to AAA; if $V_{ADCIN} < V_{REF}$, execution branches to BBB.

```
BANK1
SET1     ADCON                  ; A/D converter ON
INITFLG  NOT ADCCH1, ADCCH0     ; P1A2/AD0 pin used as A/D converter pin
INITFLG  ADCRFSEL3, NOT ADCRFSEL2, NOT ADCRFSEL1, NOT ADCRFSEL0
                                ; Sets compare voltage VREF to 8.5/16 × VDD
SET1     ADCSTRT                ; A/D operation starts
NOP                             ; Comparison in progress
NOP                             ; Comparison in progress
SKT1     ADCCMP                 ; Detects ADCCMP flag and,
BR       AAA                    ; Branches to AAA if False (0)
BR       BBB                    ; Branches to BBB if True (1)
```

13.7.2 Successive comparison by means of binary search

The A/D converter can compare only one reference voltage at a time.

Consequently, successive comparison must be executed through program in order to convert input voltages into digital signals.

If the processing time of the successive comparison program is different depending on the input voltage, it is not desirable because of the relations with the other programs.

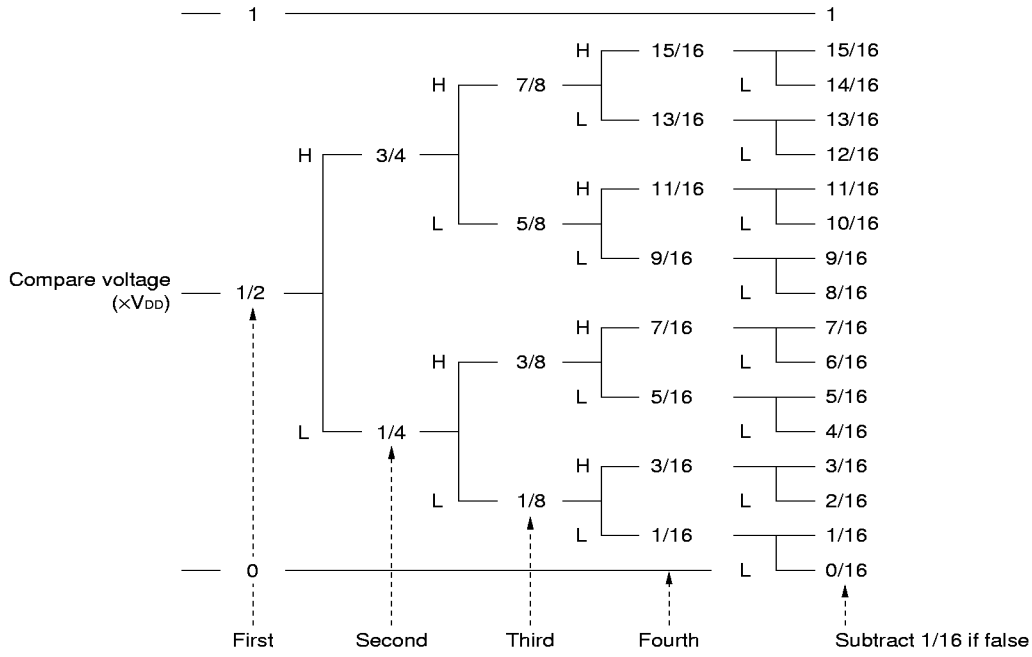
Therefore, the binary search method described in (1) through (3) below is useful.

(1) Concept of binary search

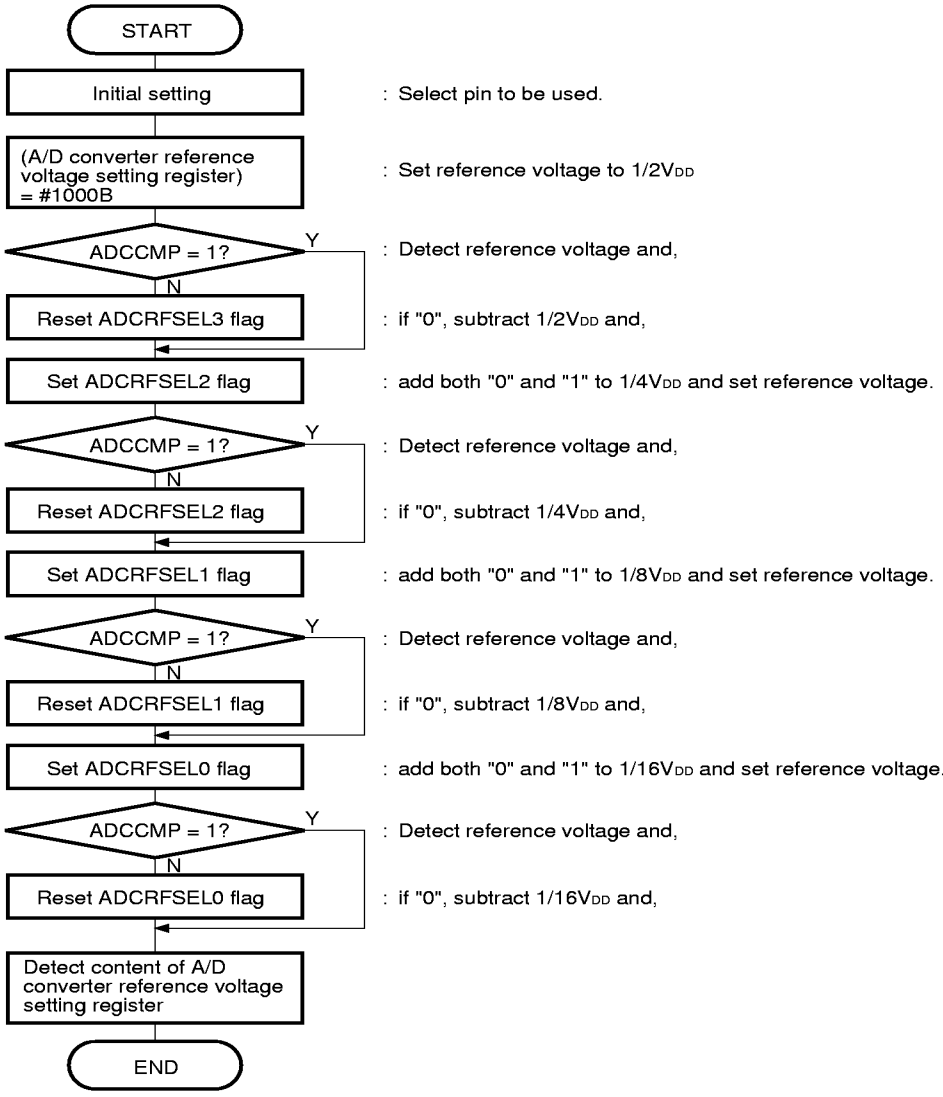
The following figure illustrates the concept of binary search.

First, the reference voltage is set to 1/2V_{DD}. If the result of comparison is True (high level), a voltage of 1/4V_{DD} is applied; if the result is False (low level), a voltage of 1/4V_{DD} is subtracted for comparison.

Similarly, comparison is performed in sequence from 1/8V_{DD} to 1/16V_{DD}. If the result is False after comparison has been executed six times, 1/64V_{DD} is subtracted, and the comparison ends.



(2) Flowchart of binary search



(3) Program example of binary search

```
START:
        BANK1
        INITFLG   NOT ADCCH1, ADCCH0                          ; Selects AD0 pin
        INITFLG   P1APLD2                                     ; Sets pull-down resistor of AD0 pin OFF
        INITFLG   NOT ADCRFSEL3, ADCRFSEL2, ADCRFSEL1, ADRFSEL0 ; Sets compare voltage to 7.5/16 VDD
        SET1      ADCSTRT                                     ; A/D converter starts operating.
        NOP                                                   ; 2 wait
        NOP                                                   ;
        SKF1      ADCCMP                                      ; Detects ADCCMP
        SET1      ADCRFSEL3                                   ; If 0,  adds 7.5/16 VDD and
        CLR1      ADCRFSEL2                                   ; subtracts 3.5/16 VDD
        SET1      ADCSTRT                                     ; A/D converter starts operating.
        NOP                                                   ; 2 wait
        NOP                                                   ;
        SKF1      ADCCMP                                      ; Detects ADCCMP
        SET1      ADCRFSEL2                                   ; If 0, adds 3.5/16 VDD and
        CLR1      ADCRFSEL1                                   ; subtracts 1.5/16 VDD
        SET1      ADCSTRT                                     ; A/D converter starts operating.
        NOP                                                   ; 2 wait
        NOP                                                   ;
        SKF1      ADCCMP                                      ; Detects ADCCMP
        SET1      ADCRFSEL1                                   ; If 0, adds 1.5/16 VDD and
        CLR1      ADCRFSEL0                                   ; subtracts 0.5/16 VDD
        SET1      ADCSTRT                                     ; A/D converter starts operating.
        NOP                                                   ; 2 wait
        NOP                                                   ;
        SKF1      ADCCMP                                      ; Detects ADCCMP
        SET1      ADCRFSEL0                                   ; If 0, adds 0.5/16 VDD

END:
```

13.8 Status at Reset

13.8.1 At power-ON reset

The P1A2/AD0 and P1A3/AD1 pins are set in the general-purpose input port mode, and internally pulled down.

13.8.2 On execution of clock stop instruction

The P1A2/AD0 and P1A3/AD1 pins are set in the general-purpose input port mode.

The previous status of the pull-down resistor is retained.

13.8.3 At CE reset

The P1A2/AD0 and P1A3/AD1 pins are set in the general-purpose input port mode.

The previous status of the pull-down resistor is retained.

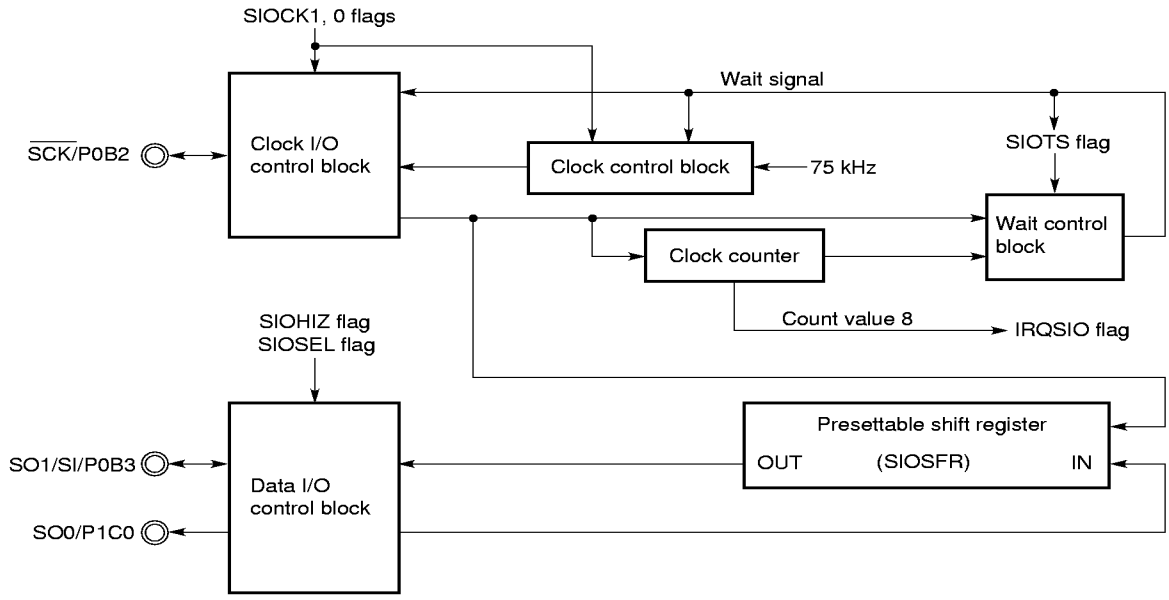
14. SERIAL INTERFACE

14.1 General

Figure 14-1 shows the outline of the serial interface.

This serial interface is of two-wire/three-wire serial I/O type. The former type uses $\overline{SCK}$ and SO1/SI pins. The latter uses $\overline{SCK}$, SI, and SO0 pins.

Figure 14-1. Outline of Serial Interface



Remarks 1. SIOCK1 and 0 (bits 0 and 1 of serial I/O clock select register. Refer to **Figure 14-2**) set a shift clock.

2. SIOTS (bit 0 of serial I/O mode select register. Refer to **Figure 14-3**) starts/stops communication.

3. SIOHIZ (bit 1 of serial I/O mode select register. Refer to **Figure 14-3**) sets the function of the SO0/P1C0 pin.

4. SIOSEL (bit 3 of serial I/O mode select register. Refer to **Figure 14-3**) selects I/O of SO1/SI/P0B3 pin.

14.2 Clock Input/Output Control Block and Data Input/Output Control Block

The clock input/output control block and data input/output control block select the operation mode of the serial interface (2-wire or 3-wire mode), control the transmit/receive operation, and select a shift clock.

The flags that control these blocks are allocated to the serial I/O clock select register and serial I/O mode select register.

Figure 14-2 shows the configuration and function of the serial I/O clock select register.

Figure 14-3 shows the configuration and function of the serial I/O mode select register.

Table 14-1 shows the setting status of each pin by the corresponding control flags. As shown in this table, the input/output setting flag of each pin must be also manipulated in addition to the control flag of the serial interface, to set each pin.

The SIOCK1 and 0 flags select the internal clock (master) or external clock (slave) operation.

The SIOHIZ flag selects whether the SO0/P1C0 pin is used as a serial data output pin.

The SIOSEL flag selects whether the SO1/SI/P0B3 pin is used as a serial data input (SI pin) or serial data output (SO1) pin.

Figure 14-2. Configuration of Serial I/O Clock Select Register

Name	Flag symbol b_3	b_2	b_1	b_0	Address	Read/Write
Serial I/O clock select register	0	0	SIOCK1	SIOCK0	(BANK1) 61H	R/W

b_1	b_0	Sets shift clock of serial interface	
0	0	External clock	
0	1	12.5 kHz	Internal clock
1	0	18.75 kHz	Internal clock
1	1	37.5 kHz	Internal clock

Fixed to "0"

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

Figure 14-3. Configuration of Serial I/O Mode Select Register

Name	Flag symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
Serial I/O mode select register	0	SIOSEL	SIOHIZ	SIOTS	(BANK1) 60H	R/W

SIOTS — Starts/stops serial communication

0	Stops (wait status)
1	Starts

SIOHIZ — Sets function of P1C0/SO0 pin

0	General-purpose output port
1	Serial data output pin

SIOSEL — Selects function of P0B3/SI/SO1 pin

0	As serial data input (SI) pin
1	As serial data output (SO1) pin

b_3 — Fixed to 0

		b_2	b_1	b_0
At reset	Power-ON	0	0	0
	Clock stop	0	0	0
	CE	0	0	0

Table 14-1. Set Status of Each Pin By Control Flags

Control flags of serial interface								I/O setting flag of each pin			
Communication mode	SIOSEL	Serial I/O select	SIOHIZ	Serial interface pin setting	SIOCK1	SIOCK0	Clock setting	Pin name	P0BBIO3	P0BBIO2	Set status of pin
3-wire serial I/O[Note 1] and 2-wire serial I/O[Note 2]					0	0	External clock	P0B2/$\overline{SCK}$		0	During wait: general-purpose input port Wait released: external clock input
										1	General-purpose output port
					0	1	Internal clock			0	General-purpose input port
					1	0				1	During wait: waits for internal clock output
					1	1					Wait released: internal clock output
	0	Internal clock (reception)						P0B3/SI/SO1	0		During wait: general-purpose input port Wait released: serial input
									1		General-purpose output port
	1	Output (transmission)							0		During wait: waits for serial output
									1		Wait released: serial output
			0	General-purpose output				P1C0/SO0			General-purpose output port
			1	Serial output							During wait: waits for serial output Wait released: serial output

Notes 1. To set the 3-wire serial I/O mode, be sure to reset SIOSEL to 0 and set SIOHIZ to 1.

2. To use the 2-wire serial I/O mode, be sure to reset SIOHIZ to 0.

14.2.1 Setting 2-/3-wire mode

The serial interface uses two pins in the two-wire mode: $\overline{SCK}$/P0B2 and SO1/SI/P0B3.

The $\overline{SCK}$/P0B2 pin is used as a shift clock input/output pin, and the SO1/SI/P0B3 pin is used as a serial data input/output pin. The SO0/P1C0 pin is not used for the serial interface and is set in the general-purpose output port mode by the SIOHIZ flag.

In this way, the serial interface operates in the two-wire mode.

In the three-wire mode, three pins, $\overline{SCK}$/P0B2, SO0/P1C0, and SO1/SI/P0B3 are used.

The $\overline{SCK}$/P0B2 is used as a shift clock input/output pin, the SO0/P1C0 pin is used as a serial data output pin, and the SO1/SI/P0B3 pin is used as a serial data input pin.

Unlike in the two-wire mode, the SO0/P1C0 pin is used as a serial data output pin according to the setting of the SIOHIZ flag. The SO1/SI/P0B3 pin is used as a serial data input pin according to the setting of the SIOSEL flag.

In this way, the serial interface operates in the three-wire mode.

14.2.2 Selecting data input/output using 2-wire serial interface

In the two-wire mode, the SO1/SI/P0B3 pin is used as an input/output pin for serial data.

Whether the SO1/SI/P0B3 pin is used as a serial data input pin (SI pin) or serial data output pin (SO1 pin) is specified by the SIOSEL flag (refer to **Figure 14-3 Configuration of Serial I/O Mode Select Register**).

14.3 Clock Control Block

The clock control block generates a clock when the internal clock is used (master operation), and controls clock output timing.

The frequency f_{SC} of the internal clock is set by the SIOCK0 and SIOCK1 flags of the serial I/O clock select register.

Figure 14-2 shows the configuration and function of the serial I/O clock select register.

For the clock generation timing, refer to **14.7 Operation of Serial Interface**.

14.4 Clock Counter

The clock counter counts the shift clock output or input from the shift clock pin ($\overline{SCK}$/P0B2 pin).

Because the clock counter directly reads the status of the clock pin, it cannot identify whether the clock is an internal clock or an external clock.

The contents of the clock counter cannot be directly read by software.

For the operation and timing chart of the clock counter, refer to **14.7 Operation of Serial Interface**.

14.5 Presettable Shift Register

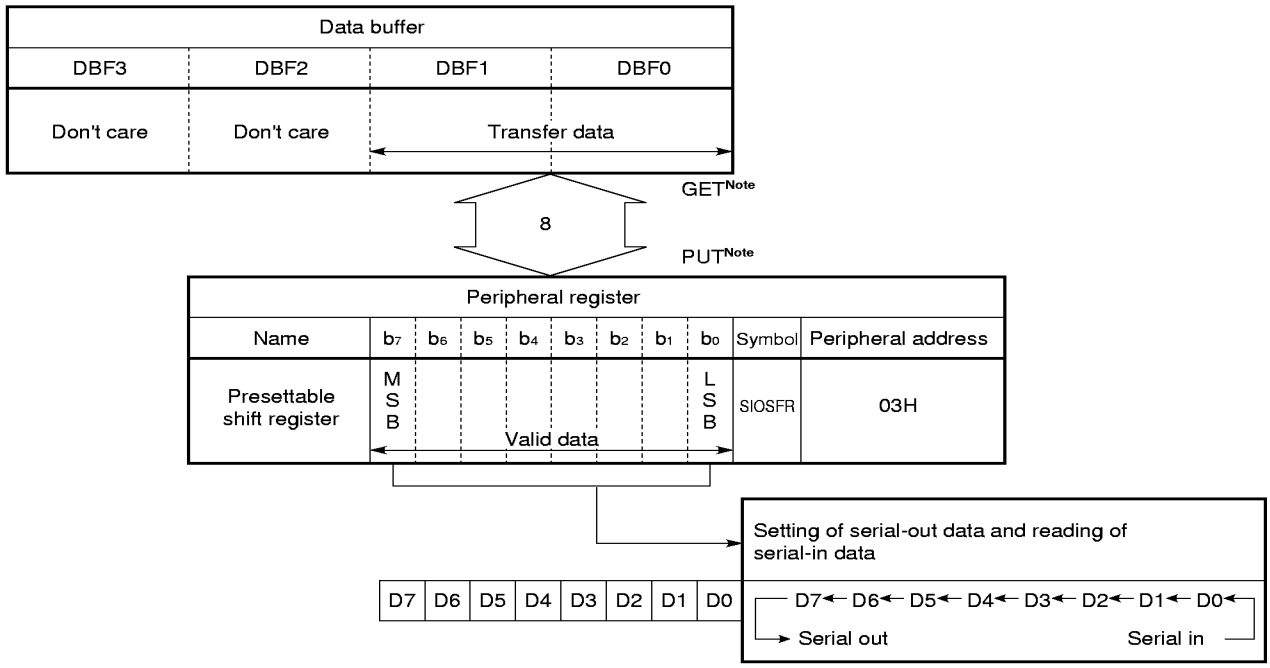
The presettable shift register is an 8-bit shift register that writes serial-out data and reads serial-in data.

Writing/reading data to/from the presettable shift register is performed by PUT and GET instructions via data buffer.

The presettable shift register outputs (transmits) the content of its most significant bit (MSB) from the serial data I/O pin in synchronization with the falling edge of the shift clock, and reads data to its least significant bit (LSB) in synchronization with the rising edge of the shift clock.

Figure 14-4 shows the configuration and function of the presettable shift register.

Figure 14-4. Configuration of Presettable Shift Register



Note If the PUT or GET instruction is executed during serial communication, the data may be lost. For details, refer to **14.8 Notes on Setting and Reading Data**.

14.6 Wait Control Block

The wait control block performs wait (pause) control of communication.

By releasing the wait status by using the SIOTS flag of the serial I/O mode select register, serial communication is started.

After the wait status has been released, and communication has been started, the wait status is resumed if shift clock rises at clock counter "8".

The communication status can be detected by the SIOTS flag.

That is, the communication status can be detected by detecting the status of the SIOTS flag after setting "1" to the SIOTS flag.

If "0" is written to the SIOTS flag while the wait status is released, the wait status is set. This is called a forced wait status.

For the configuration and function of the serial I/O mode select register, refer to **Figure 14-3**.

14.7 Serial Interface Operation

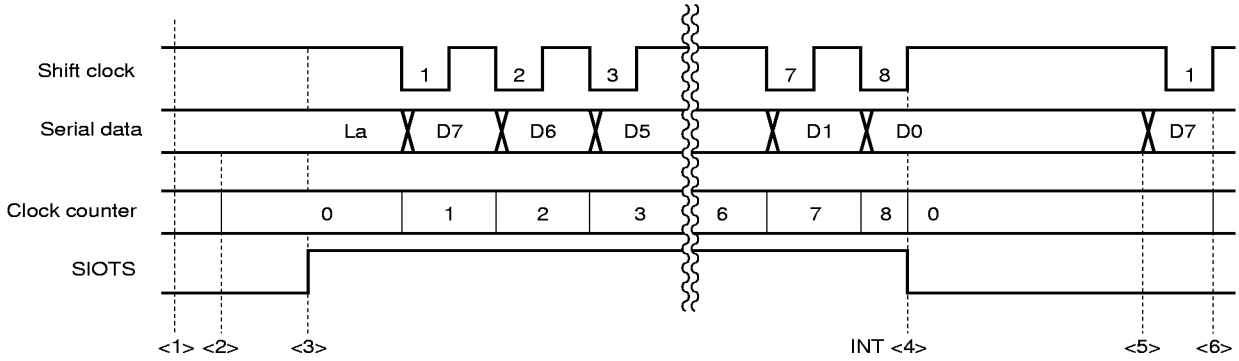
The timing of each operation of the serial interface is described below.

This timing is applicable to both 2-wire and 3-wire modes.

14.7.1 Timing chart

Figure 14-5 shows a timing chart.

Figure 14-5. Timing Chart of Serial Interface



Remarks <1> Initial status (general-purpose input port)
<2> Start condition satisfied by general-purpose I/O port
<3> Wait released
<4> Wait timing
<5> General-purpose input port mode set
<6> Stop condition satisfied by general-purpose I/O port

14.7.2 Operation of clock counter

The initial value of the clock counter is "0". The value of the clock counter is incremented each time the falling edge of the clock pin has been detected. When the value of the clock counter reaches "8", it is reset to "0" at the next rising edge of the clock pin. After the clock counter has been reset to "0", the serial communication is placed in the wait status.

The conditions under which the clock counter is reset are as follows:

(1) At power-ON reset
(2) When clock stop instruction is executed
(3) When "0" is written to SIOTS flag
(4) If shift clock rises while wait status is released and present value of clock counter is "8"

14.7.3 Wait operation and note

When the wait status has been released, serial data is output at the next falling edge of the clock (transmission operation), and the wait released status continues until eight clocks are counted.

After the eight clocks have been output, make the shift clock pin high and stop the operations of the clock counter and presettable shift register.

Note that, if data is written to or read from the presettable shift register while the wait status is released and the shift clock pin is high, the correct data is not set.

If data is written to the presettable shift register while the wait status is released and the shift clock pin is low, the content of the MSB of the data is output to the serial data output pin when the "PUT" instruction is executed.

If the forced wait status is set while the wait status is released, the wait status is immediately set when "0" is written to the SIOTS flag.

14.7.4 Interrupt request issuance timing

An interrupt request is issued when eight clocks have been transmitted (received).

14.7.5 Shift clock generation timing

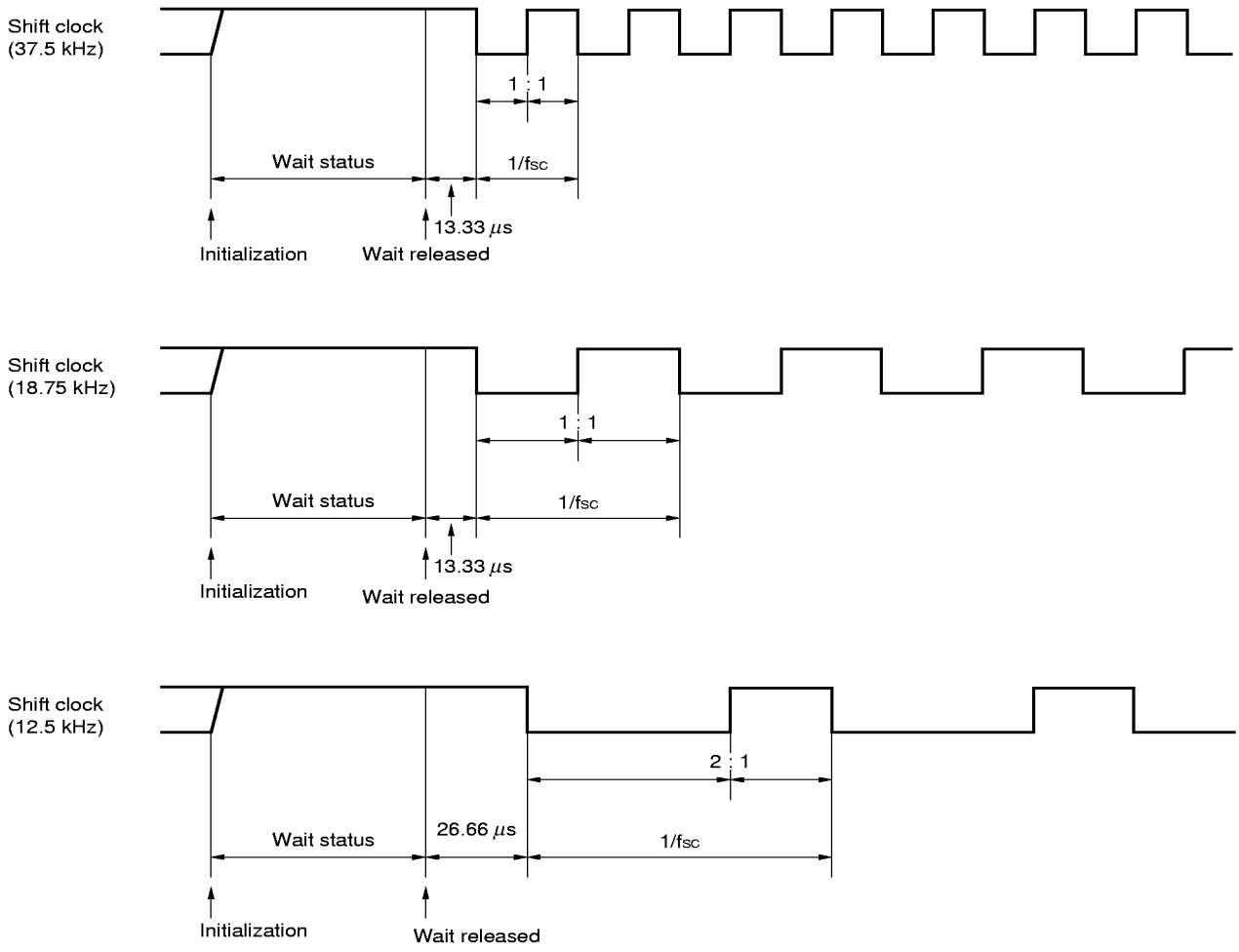
(1) When wait status is released from initial status

The "initial status" means the point at which the P0B2/$\overline{\text{SCK}}$ pin has been made high with the internal clock operation selected.

During the wait status, a high level is output to the shift clock pin.

The wait status can be released and a clock can be selected at the same time.

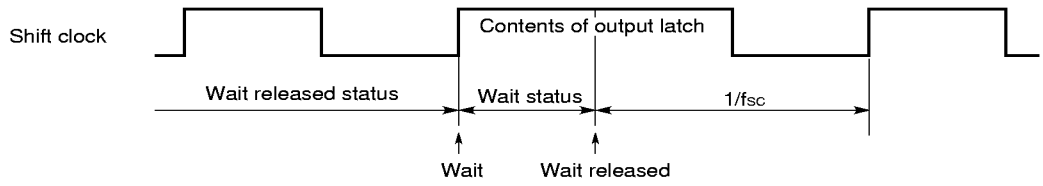
Figure 14-6. Shift Clock Generation Timing of Serial Interface (1/4)



(2) When wait operation is performed

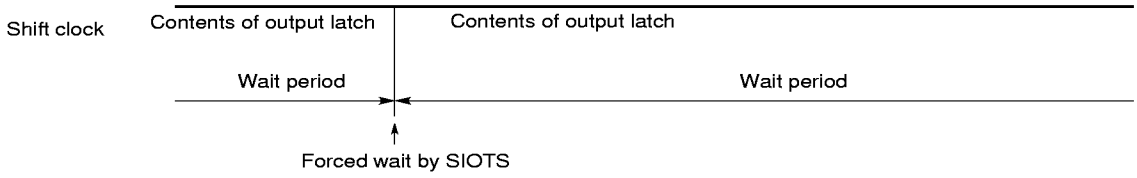
(a) When wait status is set at the 8th clock (normal operation)

Figure 14-6. Shift Clock Generation Timing of Serial Interface (2/4)



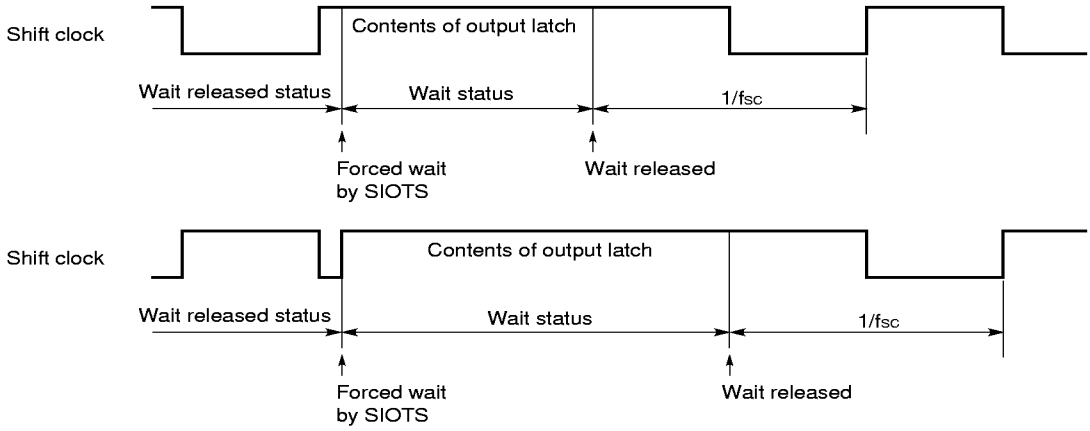
(b) When forced wait status is set during wait status

Figure 14-6. Shift Clock Generation Timing of Serial Interface (3/4)



(c) When forced wait status is set while wait status is released

Figure 14-6. Shift Clock Generation Timing of Serial Interface (4/4)



(d) When wait status is released while wait status is released

The clock output waveform does not change. Neither is the counter reset. However, do not change the clock frequency while the wait status is released.

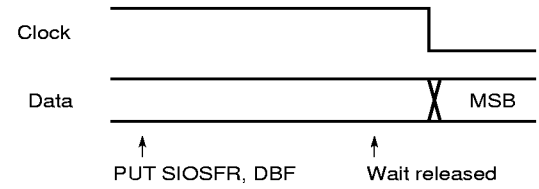
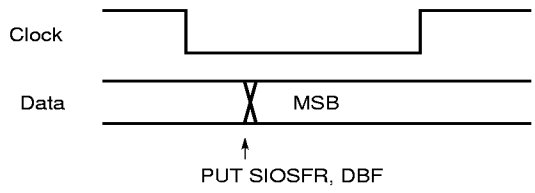
14.8 Notes on Setting and Reading Data

Use the "PUT SIOSFR, DBF" instruction to set data to the presettable shift register. Use the "GET DBF, SIOSFR" instruction to read data.

Set or read the data in the wait status. While the wait status is released, the data may not be correctly set or read depending on the status of the shift clock pin.

The following table describes the points to be noted in setting and reading data.

Table 14-2. Data Read and Write Operations of Presettable Shift Register and Notes

<table>
<tr><th colspan="2">Status on execution of PUT/GET</th><th>Status of shift clock pin</th><th>Operation of presettable shift register</th></tr>
<tr><td rowspan="2">Wait status</td><td>Read (GET)</td><td rowspan="2">• Floating with external clock
• Value of output latch with internal clock. Normally, used with high level</td><td>Normal read</td></tr>
<tr><td>Write (PUT)</td><td>Normal write
Content of MSB is output as data at falling edge of shift clock after wait status is released next (transmission operation).
 
</td></tr>
<tr><td rowspan="4">Wait release status</td><td rowspan="2">Read (GET)</td><td>Low level</td><td>Normal read</td></tr>
<tr><td>High level</td><td>Cannot be read normally.
Contents of SIOSFR are lost.</td></tr>
<tr><td rowspan="2">Write (PUT)</td><td>Low level</td><td>Cannot be written normally.
Contents of SIOSFR are lost.</td></tr>
<tr><td>High level</td><td>Normal write
Content of MSB is output as data when PUT instruction is executed.
Clock counter is not reset.
 
</td></tr>
</table>

14.9 Operational Outline of Serial Interface

Tables 14-3 and 14-4 outline the operations of the serial interface.

Table 14-3. Operation in 3-wire Serial I/O Mode

Item \ Operation mode		Slave operation (SIOCK1 = SIOCK0 = 0)		Master operation (SIOCK1 = SIOCK0 = other than 0)	
		During wait (SIOTS = 0)	Wait released (SIOTS = 1)	During wait (SIOTS = 0)	Wait released (SIOTS = 1)
Status of each pin	$\overline{SCK}$/P0B2	• When P0BBIO2 = 0 General-purpose input port • When P0BBIO2 = 1 General-purpose output port	• When P0BBIO2 = 0 External clock input port • When P0BBIO2 = 1 General-purpose output port	• When P0BBIO2 = 0 General-purpose input port • When P0BBIO2 = 1 Waits for internal clock output	• When P0BBIO2 = 0 General-purpose input port • When P0BBIO2 = 1 Internal clock output
	SI/SO1/P0B3	SIOSEL = 0			
		• When P0BBIO3 = 0 General-purpose input port • When P0BBIO3 = 1 General-purpose output port	• When P0BBIO3 = 0 Serial input • When P0BBIO3 = 1 General-purpose output port	• When P0BBIO3 = 0 General-purpose input port • When P0BBIO3 = 1 General-purpose output port	• When P0BBIO3 = 0 Serial input • When P0BBIO3 = 1 General-purpose output port
	SO0/P1C0	SIOHIZ = 1			
		Waits for serial output	Serial output	Waits for serial output	Serial output
Program counter		Incremented at falling edge of $\overline{SCK}$ pin			
Operation of presettable shift register	Output	• **When SIOHIZ = 0** Not output • **When SIOHIZ = 1** Shifted from MSB and output from SO0 pin at falling edge of $\overline{SCK}$ pin			
	Input	• **When SIOSEL = 0** Shifted from LSB and status of SI pin is input at rising edge of $\overline{SCK}$ pin. If SI pin is set in output mode, however, contents of output latch are input.			

Table 14-4. Operation in Two-Wire Serial I/O Mode

Item \ Operation mode		Slave operation (SIOCK1 = SIOCK0 = 0)		Master operation (SIOCK1 = SIOCK0 = other than 0)	
		During wait (SIOTS = 0)	Wait released (SIOTS = 1)	During wait (SIOTS = 0)	Wait released (SIOTS = 1)
Status of each pin	$\overline{SCK}$/P0B2	• When P0BBIO2 = 0 General-purpose input port • When P0BBIO2 = 1 General-purpose output port	• When P0BBIO2 = 0 External clock input port • When P0BBIO2 = 1 General-purpose output port	• When P0BBIO2 = 0 General-purpose input port • When P0BBIO2 = 1 Waits for internal clock output	• When P0BBIO2 = 0 General-purpose input port • When P0BBIO2 = 1 Internal clock output
	SI/SO1/P0B3	SIOSEL = 0			
		• When P0BBIO3 = 0 General-purpose input port • When P0BBIO3 = 1 General-purpose output port	• When P0BBIO3 = 0 Serial input • When P0BBIO3 = 1 General-purpose output port	• When P0BBIO3 = 0 General-purpose input port • When P0BBIO3 = 1 General-purpose output port	• When P0BBIO3 = 0 Serial input • When P0BBIO3 = 1 General-purpose output port
		SIOSEL = 0			
		Waits for serial output regardless of P0BBIO3	Serial output regardless of P0BBIO3	Waits for serial output regardless of P0BBIO3	Serial output regardless of P0BBIO3
	SO0/P1C0	SIOHIZ = 1			
		General-purpose output port			
Program counter		Incremented at falling edge of $\overline{SCK}$ pin			
Operation of presettable shift register	Output	• **When SIOSEL = 1** Shifted from MSB and output from SIO1 pin at falling edge of $\overline{SCK}$ pin			
	Input	• **When SIOSEL = 0** Shifted from LSB and status of SI pin is input at rising edge of $\overline{SCK}$ pin. If SI pin is set in output port mode, however, contents of output latch are input.			

14.10 Status on Reset

14.10.1 At power-ON reset

P0B2/$\overline{SCK}$ and P0B3/SI/SO1 pins are set in the general-purpose input port mode.

P1C0/SO0 pin is set in the general-purpose port.

The contents of the presettable shift register are undefined.

14.10.2 At clock stop

P0B2/$\overline{SCK}$ and P0B3/SI/SO1 pins set in the general-purpose input port mode.

P1C0/SO0 pin is set in the general-purpose port.

The previous contents of the presettable shift register are retained.

14.10.3 At CE reset

P0B2/$\overline{SCK}$ and P0B3/SI/SO1 pins are set in the general-purpose input port mode.

P1C0/SO0 pin is set in the general-purpose port.

The previous contents of the presettable shift register are retained.

14.10.4 In halt status

All the pins hold the current status.

The internal clock stops output in the status in which the HALT instruction is executed.

When the external clock is used, the operation continued even if the HALT instruction is executed.

15. PLL FREQUENCY SYNTHESIZER

The PLL (Phase Locked Loop) frequency synthesizer is used to lock a frequency in the MF (Medium Frequency), HF (High Frequency), and VHF (Very High Frequency) bands to a fixed frequency, by means of phase difference comparison.

15.1 General

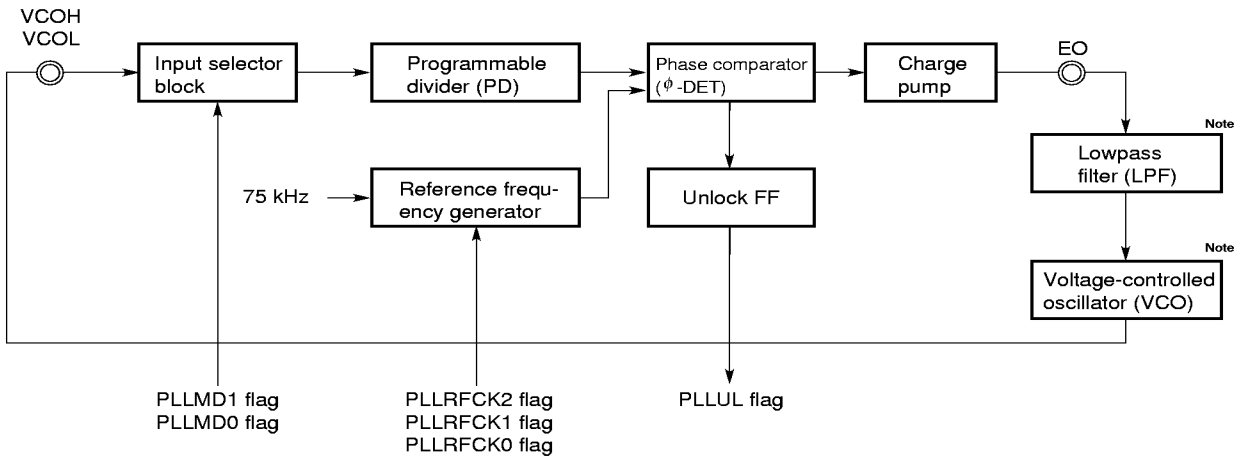
Figure 15-1 outlines the PLL frequency synthesizer. By connecting an external lowpass filter (LPF) and voltage controlled oscillator (VCO), the PLL frequency synthesizer can be configured.

The PLL frequency synthesizer divides a signal input from the VCOH or VCOL pin by using a programmable divider, and outputs the phase difference between the signal and the reference frequency from the EO pin.

However, the signal input from the VCOH pin is halved immediately before it is input to the programmable divider.

The PLL frequency synthesizer operates only while the CE pin is high. When the CE pin is low, the synthesizer is disabled. For details of the PLL disable status, refer to **15.5 PLL Disable Status**.

Figure 15-1. Outline of PLL Frequency Synthesizer



Note External circuit

Remarks 1. PLLMD1 and 0 (bits 1 and 0 of PLL mode select register. Refer to **Figure 15-3**) set the division method of the PLL frequency synthesizer.

2. PLLRFCK2, 1, and 0 (bits 2-0 of PLL reference frequency select register. Refer to **Figure 15-7**) set the reference frequency f_r of the PLL frequency synthesizer.

3. PLLUL (bit 0 of PLL unlock FF register. Refer to **Figure 15-10**) detects the status of the unlock FF.

15.2 Input Selector Block and Programmable Divider

15.2.1 Configuration and function of input selector block and programmable divider

Figure 15-2 shows the configuration of the input selector block and programmable divider.

The input selector block selects the input pin and division method of the PLL frequency synthesizer.

As the input pin, the VCOH or VCOL pin can be selected.

The selected pin is at the intermediate potential (approx. 1/2V_{DD}). The pin not selected is internally pulled down.

These pins have an AC amplifier at the input stage; therefore, cut the DC component of the input signal by connecting a capacitor in series.

As the division method, DC division method or pulse swallow method can be selected.

The programmable divider performs frequency division according to the values set to the swallow counter and programmable counter.

Table 15-1 shows each input pin (VCOH and VCOL) and division method.

The input pin and division method to be used are selected by the PLL mode select register.

Figure 15-3 shows the configuration of the PLL mode select register.

A division value is set by using the PLL data register.

The division value is transferred to the programmable divider using PLL data set register.

Figure 15-2. Configuration of Input Selector Block and Programmable Divider

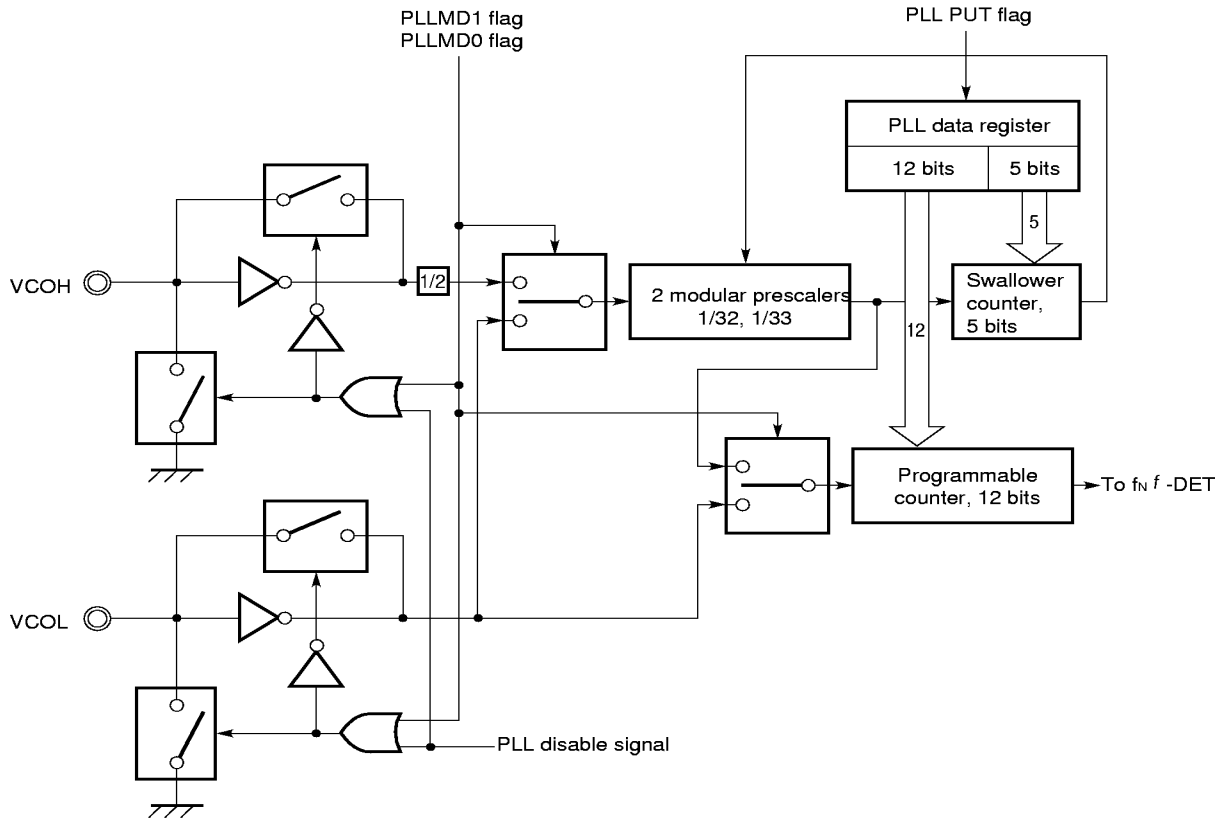
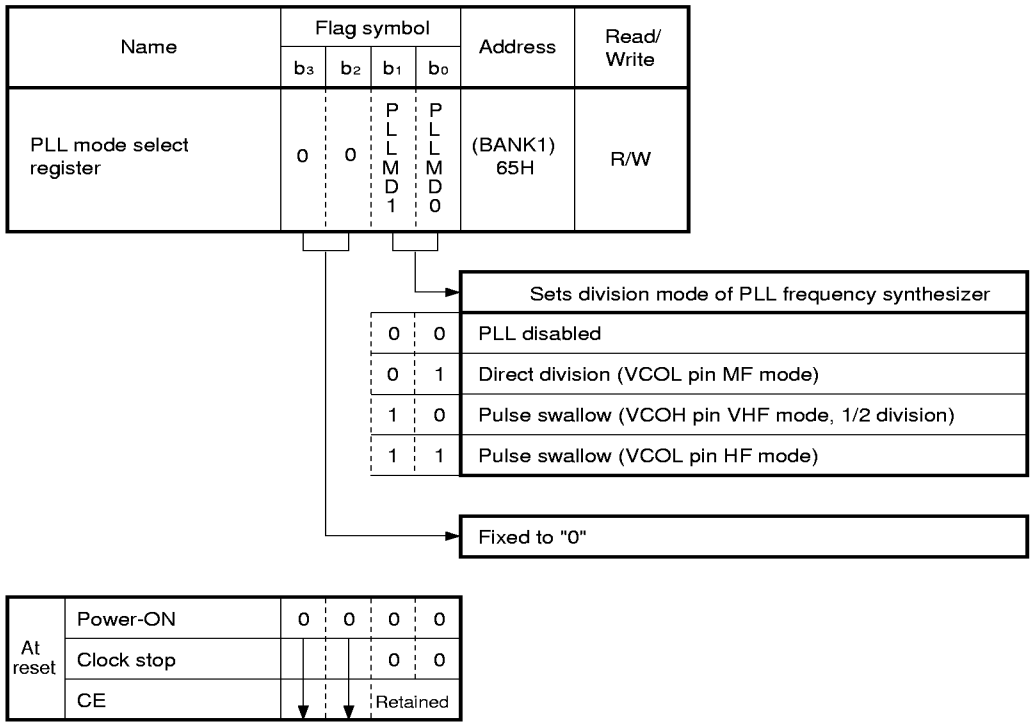


Table 15-1. Input Pins and Division Modes

Division mode	Pin	Input frequency (MHz)	Input amplitude (V_{p-p})	Division value	Division value set to data buffer
Direct division (MF)	VCOL	0.3 - 8	0.2	16 to $2^{12} - 1$	010xH - FFFxH (x: don't care)
Pulse swallow (HF)	VCOL	5 - 130	0.3	1024 to $2^{17} - 1$	0400H – 1FFFFH
Pulse swallow (VHF)	VCOH	40 - 230	0.2	1024 to $2^{17} - 1$	0400H – 1FFFFH

Figure 15-3. Configuration of PLL Mode Select Register



15.2.2 Outline of each division mode

(1) Direct division mode (MF)

In this mode, the VCOL pin is used.
The VCOH pin is floated.
The frequency of the input signal is divided only by the programmable counter in this mode.

(2) Pulse swallow mode (HF)

The VCOL pin is used, and the VCOH pin is floated.
In this mode, the frequency is divided by the swallow counter and programmable counter.

(3) Pulse swallow mode (VHF)

The VCOH pin is used, and the VCOL pin is floated. If this mode is selected 1/2 division is inserted in the stage previous to programmable divider.
In this mode, the swallow counter and programmable counter are used for frequency division.

(4) PLL disable

Refer to **15.5 PLL Disable Status**.

15.2.3 Programmable divider, PLL data register, and PLL data set register

A division value is set to the swallow counter and programmable counter by the PLL data register. The value set by the PLL data register is transferred by the PLL data set register to the swallow counter and programmable counter.

The swallow counter and programmable counter are 5-bit and 12-bit binary counters.

The value to be divided is called an "N value".

For how to set the division value (N value) in each division mode, refer to **15.6 Using PLL Frequency Synthesizer**.

(1) Configuration and functions of PLL data register

The configuration of the PLL data register is shown in Figure 15-4.

The higher 12 bits of the 16-bit PLL data register are valid in the direct division mode, and all the 17 bits of the register are valid in the pulse swallow mode.

In the direct division mode, the 12 valid register bits are set to the programmable counter.

In the pulse swallow mode, the higher 12 bits are set to the programmable counter, and the remaining lower 5 bits are set to the swallow counter.

(2) Configuration and function of PLL data set register

Figure 15-5 shows the configuration of the PLL data set register.

By writing "1" to the PLLPUT flag, the division value set by the PLL data register is transferred to the swallow counter and programmable counter.

After the data has been set, the PLLPUT flag is reset to "0".

(3) Relations between value N of programmable divider and output frequency

Value "N" set to the PLL data register and the frequency "f_N" that is divided and output by the programmable divider are determined as follows.

For details, refer to **15.6 Use of PLL Frequency Synthesizer**.

(a) In direct division mode (MF)

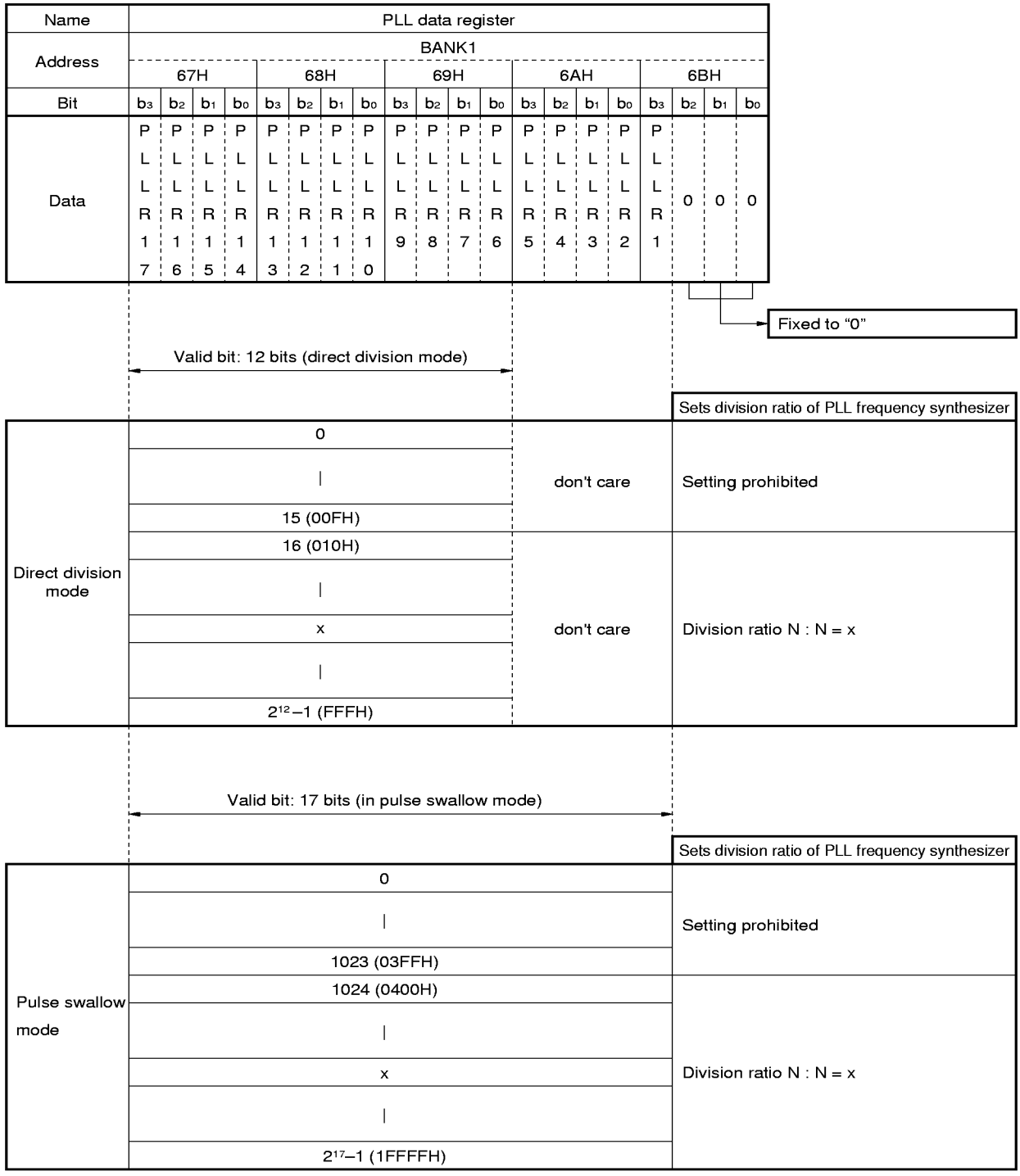
$$f_N = \frac{f_{IN}}{N}$$ N: 12 bits

(b) In pulse swallow mode (HF, VHF)

$$f_N = \frac{f_{IN}}{N}$$ N: 17 bits

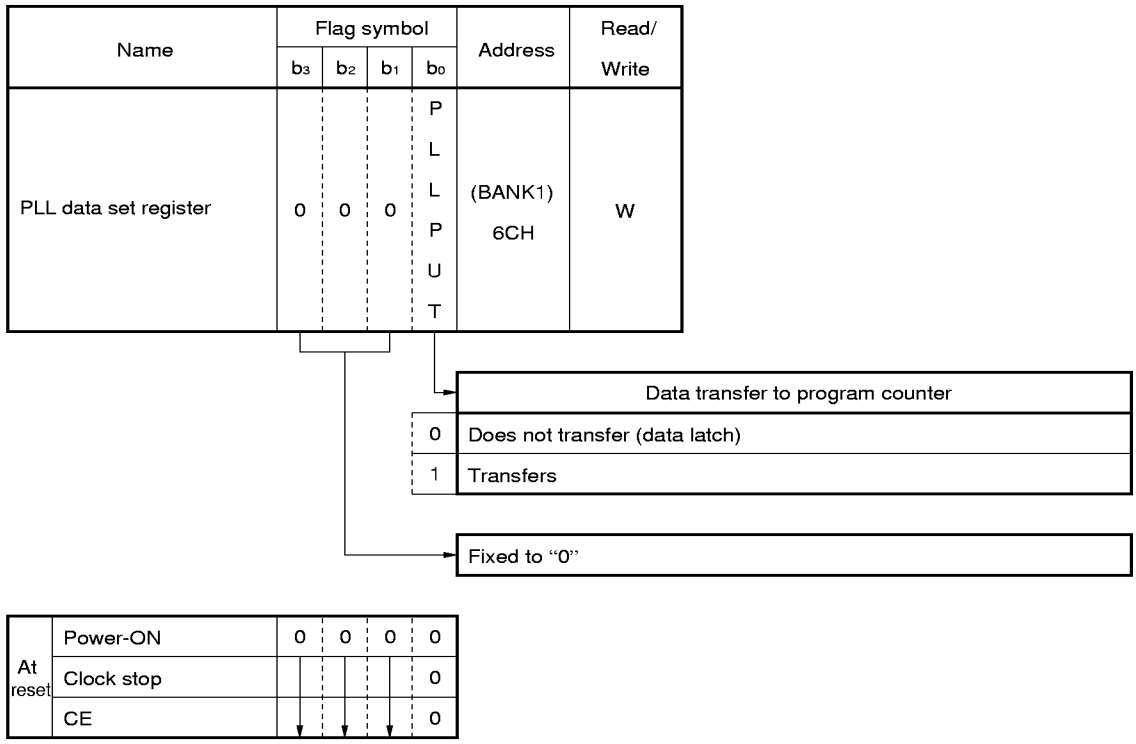
Note In VHF mode, frequency f_{IN} of the signal input from VCOH pin is divided by two immediately before input to the programmable divider. Therefore, $f_N = \frac{1}{2}\frac{f_{IN}}{N}$ in the VHF mode.

Figure 15-4. Configuration of PLL Data Register



Remark On power application and at power-ON reset, the contents of the PLL data register are undefined. On execution of the clock stop instruction and at CE reset, the contents of the PLL data register are retained.

Figure 15-5. Configuration of PLL Data Set Register



15.3 Reference Frequency Generator

Figure 15-6 shows the configuration of the reference frequency generator.

The reference frequency generator divides 75 kHz output by the crystal oscillator to generate the reference frequency "f_r" of the PLL frequency synthesizer.

As reference frequency f_r, six frequencies can be selected: 1, 3, 5, 6.25, 12.5, and 25.

Reference frequency f_r is selected by the PLL reference frequency select register.

Figure 15-7 shows the configuration and functions of the PLL reference frequency select register.

Figure 15-6. Configuration of Reference Frequency Generator

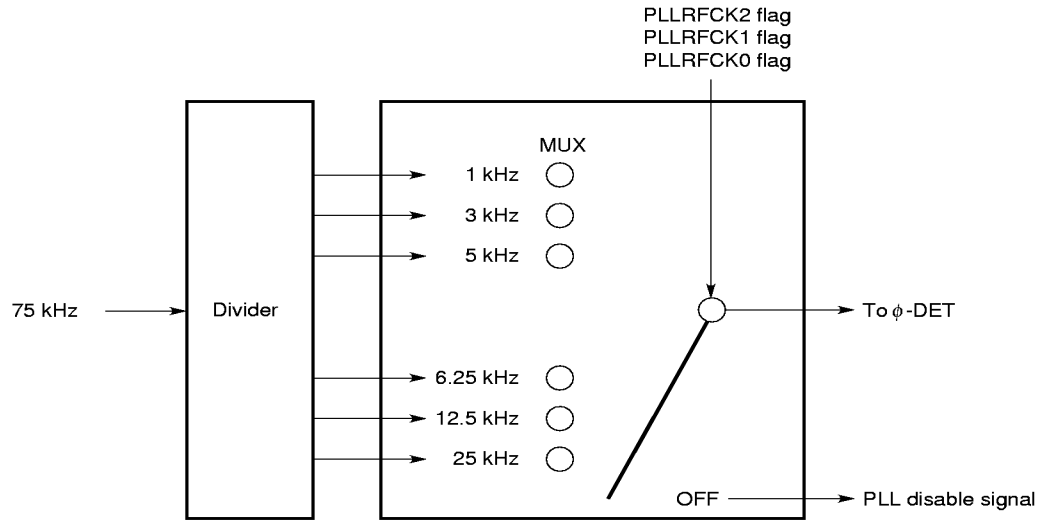


Figure 15-7. Configuration of PLL Reference Frequency Select Register

Name	Flag symbol b_3	b_2	b_1	b_0	Address	Read/ Write
PLL reference frequency select register	0	PLLRFCK2	PLLRFCK1	PLLRFCK0	(BANK1) 66H	R/W

PLLRFCK2	PLLRFCK1	PLLRFCK0	Sets reference frequency f_r of PLL frequency synthesizer
0	0	0	1 kHz
0	0	1	3 kHz
0	1	0	5 kHz
0	1	1	6.25 kHz
1	0	0	12.5 kHz
1	0	1	25 kHz
1	1	0	PLL disable
1	1	1	PLL disable

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

Remark When the PLL reference frequency select register is set to "PLL disable" status, the VCOH and VCOL pins are floated. The EO pin is floated.

15.4 Phase Comparator (φ-DET), Charge Pump, and Unlock FF

15.4.1 Configurations of phase comparator, charge pump, and unlock FF

Figure 15-8 shows the configurations of the phase comparator, charge pump, and unlock FF.

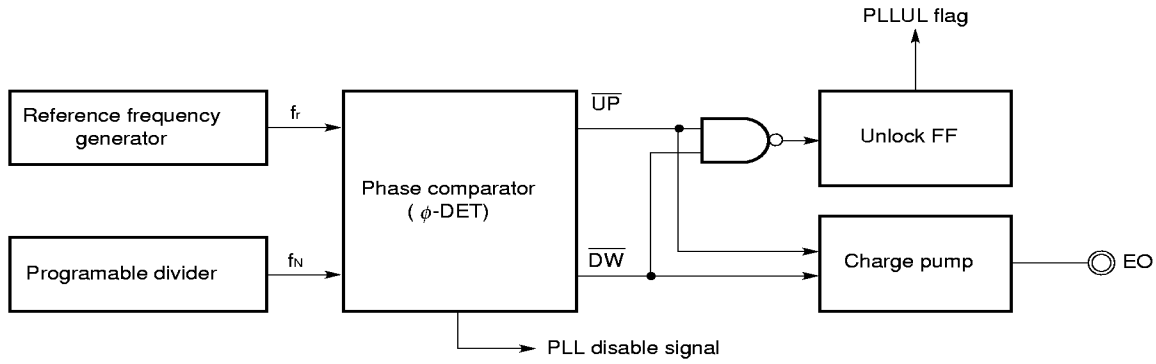
The phase comparator compares the output frequency of the programmable divider, "f_N", against the output frequency of the reference frequency generator, "f_r", and outputs $\overline{UP}$ or $\overline{DW}$ request signal.

The charge pump outputs the output signal of the phase comparator from the error out pin (EO).

The unlock FF detects the unlock status of the PLL frequency synthesizer.

The following paragraphs 15.4.2 through 15.4.4 respectively describe the operations of the phase comparator, charge pump, and unlock FF.

Figure 15-8. Configurations of Phase Comparator, Charge Pump, and Unlock FF



15.4.2 Functions of phase comparator

As shown in Figure 15-8, the phase comparator compares the output frequency of the programmable divider "f_N" against reference frequency "f_r", and outputs UP or DOWN request signal.

That is, if f_N is lower than f_r, it outputs the UP request signal; if f_N is higher than f_r, the DOWN request signal is output.

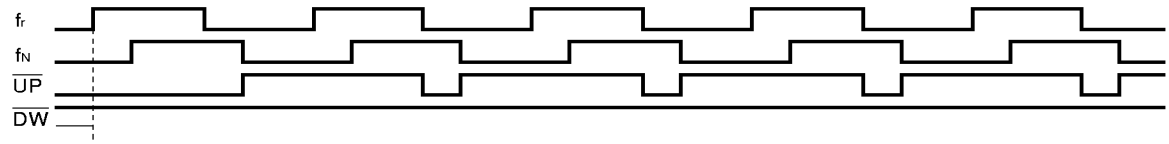
Figure 15-9 shows the relations among f_r, f_N, and UP and DOWN request signals.

In the PLL disable status, neither UP nor DOWN request signal is output.

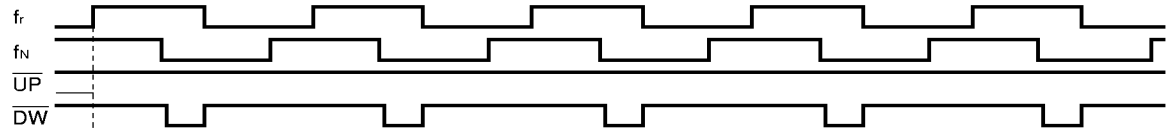
The UP and DOWN request signals are input to the charge pump and unlock FF.

Figure 15-9. Relations among f_r, f_N, $\overline{UP}$, and $\overline{DW}$

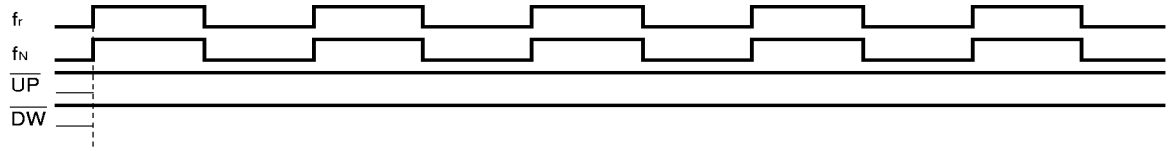
(a) If f_N lags behind f_r



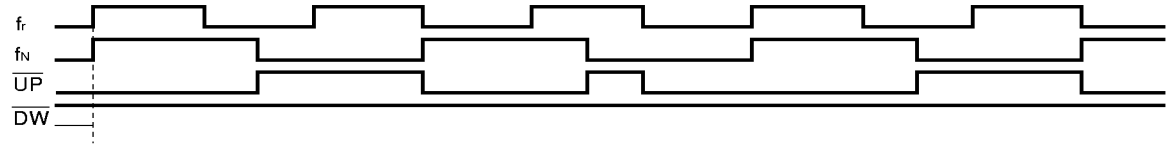
(b) If f_N advances f_r



(c) If f_N and f_r are in phase



(d) If f_N is lower than f_r



15.4.3 Charge pump

As shown in Figure 15-8, the charge pump outputs the UP and DOWN request signals from the phase comparator to the error out pin (EO).

Therefore, the relations among the outputs of the error out pins, divided frequency f_N, and reference frequency f_r are as follows:

When $f_r > f_N$: Low-level output
When $f_r < f_N$: High-level output
When $f_r = f_N$: Floating

15.4.4 Unlock FF

As shown in Figure 15-8, the unlock FF detects the unlock status of the PLL frequency synthesizer in response to the UP or DOWN request signal output from the phase comparator.

In the unlock status, either one of the UP or DOWN request signals goes low. Therefore, the unlock status can be detected when one of the request signals has gone low.

In the unlock status, the unlock FF is set to 1.

The status of the unlock FF is detected by the PLL unlock FF register. Figure 15-10 shows the configuration and function of the PLL unlock FF register.

The unlock FF is set at the cycle of the selected reference frequency f_r.

The unlock FF is reset when the contents of the PLL unlock FF register is read by the instruction shown in Table 15-2 (Read & Reset).

Therefore, the unlock FF must be detected at a cycle longer than the cycle of the reference frequency f_r (which is $1/f_r$).

The delay of the up and down request signals of the phase comparator is fixed to about 1 μs.

Figure 15-10. Configuration of PLL Unlock FF Register

Name	Flag symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
PLL unlock FF register	0	0	0	PLLUL	(BANK1) 6DH	R & Reset

Detects unlock FF status	
0	Unlock FF = 0: PLL lock status
1	Unlock FF = 1: PLL unlock status

Fixed to "0"

At reset	Power-ON	0	0	0	U
	Clock stop				R
	CE				R

Remark U: Undefined
R: Retained

Table 15-2. Instructions to Reset PLL Unlock FF Register

Mnemonic	Operand	Mnemonic	Operand
ADD	m, #n4	ADD	r, m
ADDC		ADDC	
SUB		SUB	
SUBC		SUBC	
AND		AND	
OR		OR	
XOR		XOR	
SKE		LD	
SKEG		SKT	m, #n
SKLT		SKF	
SKNE		MOV	@r, m
			m, @r[Note]

Note When the row address of m is 6H and 0DH is written to r.

Remark m = 6DH

15.5 PLL Disable Status

The PLL frequency synthesizer stops its operation (i.e., is disabled) while the CE pin is low.

Similarly, the synthesizer stops when the "PLL disable status" is selected by the PLL reference frequency select register or PLL mode select register even the CE pin is high.

Table 15-3 shows the operations of the respective blocks when the PLL synthesizer is disabled.

The PLL reference frequency select register and PLL mode select register are not initialized on CE reset, but retains their previous contents; therefore, the original status of the synthesizer is restored when the CE pin goes high after the CE pin has gone low and the PLL disable status has been set.

To set the PLL disable status after the CE reset has been effected, initialization must be performed through program.

The PLL disable status is set on power-ON reset.

Table 15-3. Operations of Respective Blocks in PLL Disable Status

Condition / Block	CE pin = low (PLL disable)	CE pin = high	
		PLL reference frequency select register = 0110B, 0111B	PLL mode select register = 0000B
VCOL, VCOH pins	Floated		
Programmable divider	Division stopped		
Reference frequency generator	Output stopped		
Phase comparator			
Charge pump	EO pin floated		

15.6 Use of PLL Frequency Synthesizer

To control the PLL frequency synthesizer, the following data are necessary:

(1) Division mode : direct division (MF) or pulse swallow (HF, VHF)
(2) Pin to be used : VCOL or VCOH
(3) Reference frequency : f_r
(4) Division value : N

The following paragraphs 15.6.1 through 15.6.3 describe how to set the above data in each division mode (MF, HF, or VHF).

15.6.1 Direct Division Mode (MF)

(1) Selecting division mode

Select the direct division mode by the PLL mode select register.

(2) Pin to be used

The VCOL pin is enabled when the direct division mode is selected.

(3) Setting of reference frequency f_r

Set the reference frequency by using the PLL reference frequency select register.

(4) Calculating division value N

Calculate as follows:

$$N = \frac{f_{VCOL}}{f_r}$$

where,

f_{VCOL} : input frequency of VCOL pin
f_r : reference frequency

(5) Example of PLL data setting

Suppose that broadcasting in the following MW band is to be received:

Receive frequency : 1422 kHz (MW band)
Reference frequency : 3 kHz
Intermediate frequency : +450 kHz

The division value N is:

$$N = \frac{f_{VCOL}}{f_r} = \frac{1422 + 450}{3} = 624 \text{ (decimal)} = 270\text{H (hexadecimal)}$$

Then set the PLL data register, PLL mode select register, and PLL reference frequency select register as follows:

PLL data register															
0	0	1	0	0	1	1	1	0	0	0	0	don't care			
2				7				0							

PLL mode select register				PLL reference frequency select register			
0	0	0	1	0	0	0	1
MF				3 kHz			

15.6.2 Pulse swallow mode (HF)

(1) Selecting division mode

Select the pulse swallow mode by the PLL mode select register.

(2) Pin to be used

The VCOL pin is enabled when the pulse swallow mode is selected.

(3) Setting reference frequency f_r

Set the reference frequency by using the PLL reference frequency select register.

(4) Calculating division value N

Calculate as follows:

$$N = \frac{f_{VCOL}}{f_r}$$

where,

f_{VCOL} : input frequency of VCOL pin
f_r : reference frequency

(5) Example of PLL data setting

Suppose that broadcasting in the following SW band is to be received:

Receive frequency : 25.50 MHz (SW band)
Reference frequency : 5 kHz
Intermediate frequency : +450 kHz

The division ratio N is:

$$N = \frac{f_{VCOL}}{f_r} = \frac{25500 + 450}{5} = 5190 \text{ (decimal)} = 1446H \text{ (hexadecimal)}$$

Then set the PLL data register, PLL mode select register, and PLL reference frequency select register as follows:

PLL data register

0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	1	0
			1				4				4				6	

PLL mode select register				PLL reference frequency select register			
0	0	1	1	0	0	1	0
HF				5 kHz			

15.6.3 Pulse swallow mode (VHF)

(1) Selecting division mode

Select the pulse swallow mode by the PLL mode select register.

(2) Pin to be used

The VCOH pin is enabled when the pulse swallow mode is selected.

(3) Setting of reference frequency f_r

Set the reference frequency by using the PLL reference frequency select register.

(4) Calculating division value N

Calculate as follows:

$$N = \frac{f_{VCOH}}{f_r} \times \frac{1}{2}^{\text{Note}}$$

where,

f_{VCOH} : input frequency of VCOH pin

f_r : reference frequency

(5) Example of PLL data setting

Suppose that broadcasting in the following FM band is to be received:

Receive frequency : 100.0 MHz (FM band)

Reference frequency : 25 kHz

Intermediate frequency : +10.7 MHz

The division ratio N is:

$$N = \frac{f_{VCOH}}{f_r} \times \frac{1}{2}^{\text{Note}} = \frac{100.0 + 10.7}{0.025} \times \frac{1}{2}^{\text{Note}} = 2214 \text{ (decimal)} = 08A6 \text{ (hexadecimal)}$$

Then set the PLL data register, PLL mode select register, and PLL reference frequency select register as follows:

PLL data register

0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	1	0
			0				8				A				6	

PLL mode select register				PLL reference frequency select register			
0	0	1	0	0	1	0	1
VHF				25 kHz			

Note The signal input from VCOH pin is divided by two immediately before input to the programmable divider.

15.7 Status on Reset

15.7.1 On power-ON reset

The PLL mode select register is initialized to 0000B; therefore, the PLL disable status is set.

15.7.2 On clock stop

The PLL disable status is set when the CE pin goes low.

15.7.3 On CE reset

(1) Transition from clock stop to CE reset status

The PLL mode select register has been initialized to 0000B when the clock stop instruction has been executed; therefore, the PLL disable status is set.

(2) On CE reset

The PLL reference frequency select register retains the previous status; therefore, the previous status is restored when the CE pin goes high.

15.7.4 In halt status

The set status is retained if the CE pin is high.

16. INTERMEDIATE FREQUENCY (IF) COUNTER

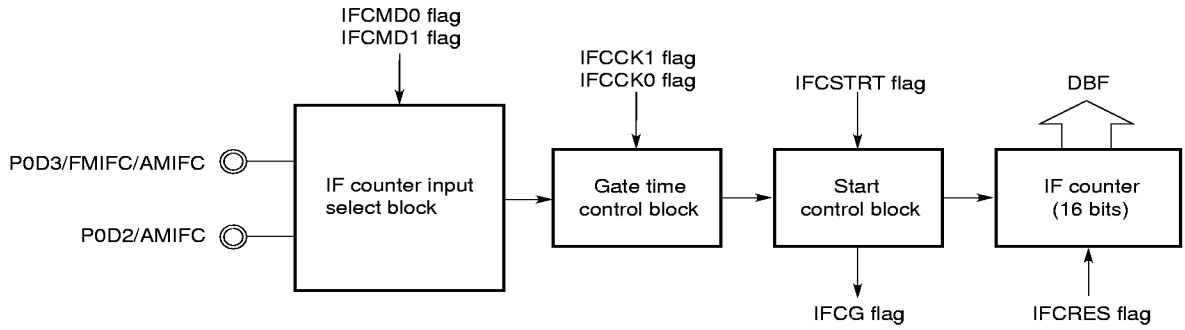
16.1 Outline of Intermediate Frequency (IF) Counter

Figure 16-1 outlines the IF counter.

The IF counter is mainly used to detect broadcasting stations, and is used to count the intermediate frequency (IF) output from a tuner.

The IF counter counts the frequency input to the P0D3/FMIFC/AMIFC or P0D2/AMIFC pin for a fixed time (1 ms, 4 ms, 8 ms, or open), by using a 16-bit counter.

Figure 16-1. Outline of Frequency Counter



Remarks 1. IFCMD1 and IFCMD0 (bits 3 and 2 of IF counter mode select register. Refer to **Figure 16-3**) select the IF counter function.

2. IFCCK1 and IFCCK0 (bits 1 and 0 of IF counter mode select register. Refer to **Figure 16-3**) select the gate time of the IF counter.

3. IFCSTRT (bit 1 of IF counter control register. Refer to **Figure 16-5**) controls starting of the IF counter.

4. IFCG (bit 0 of IF counter gate status detection register. Refer to **Figure 16-6**) detects opening/closing of the gate of the IF counter.

5. IFCRES (bit 0 of IF counter control register. Refer to **Figure 16-5**) resets the count value of the IF counter.

16.2 IF Counter Input Select Block and Gate Time Control Block

Figure 16-2 shows the configuration of the IF counter input select block and gate time control block.

The IF counter input select block selects, by using the IF counter mode select register, whether the P0D3/FMIFC/AMIFC and P0D2/AMFIC pin are used as IF counter function pins or general-purpose I/O port pins.

When using the IF counter function, be sure to set the P0D3/FMIFC/AMIFC and P0D2/AMIFC pins in the input mode. These pins can be set in the input or output mode by using the port 0C bit I/O select register at address 6FH of BANK1 of RAM. For the configuration and function of the port 0C bit I/O select register, refer to **10.2.3 (2) Port 0C bit I/O select register**.

The gate time control block sets the gate time when the IF counter function is used, by using the IF counter mode select register.

Figure 16-3 shows the configuration and function of the IF counter mode register.

Figure 16-2. Configuration of IF Counter Input Select Block and Gate Time Control Block

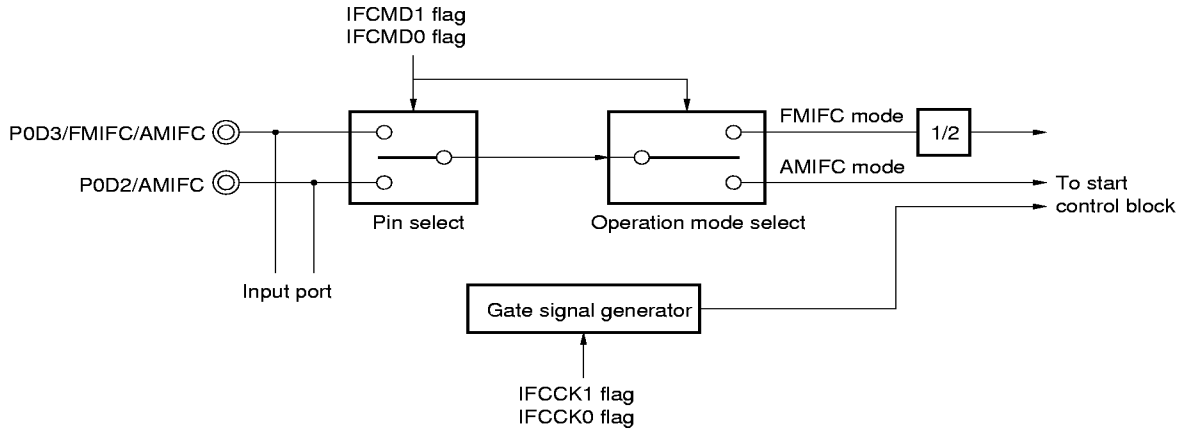
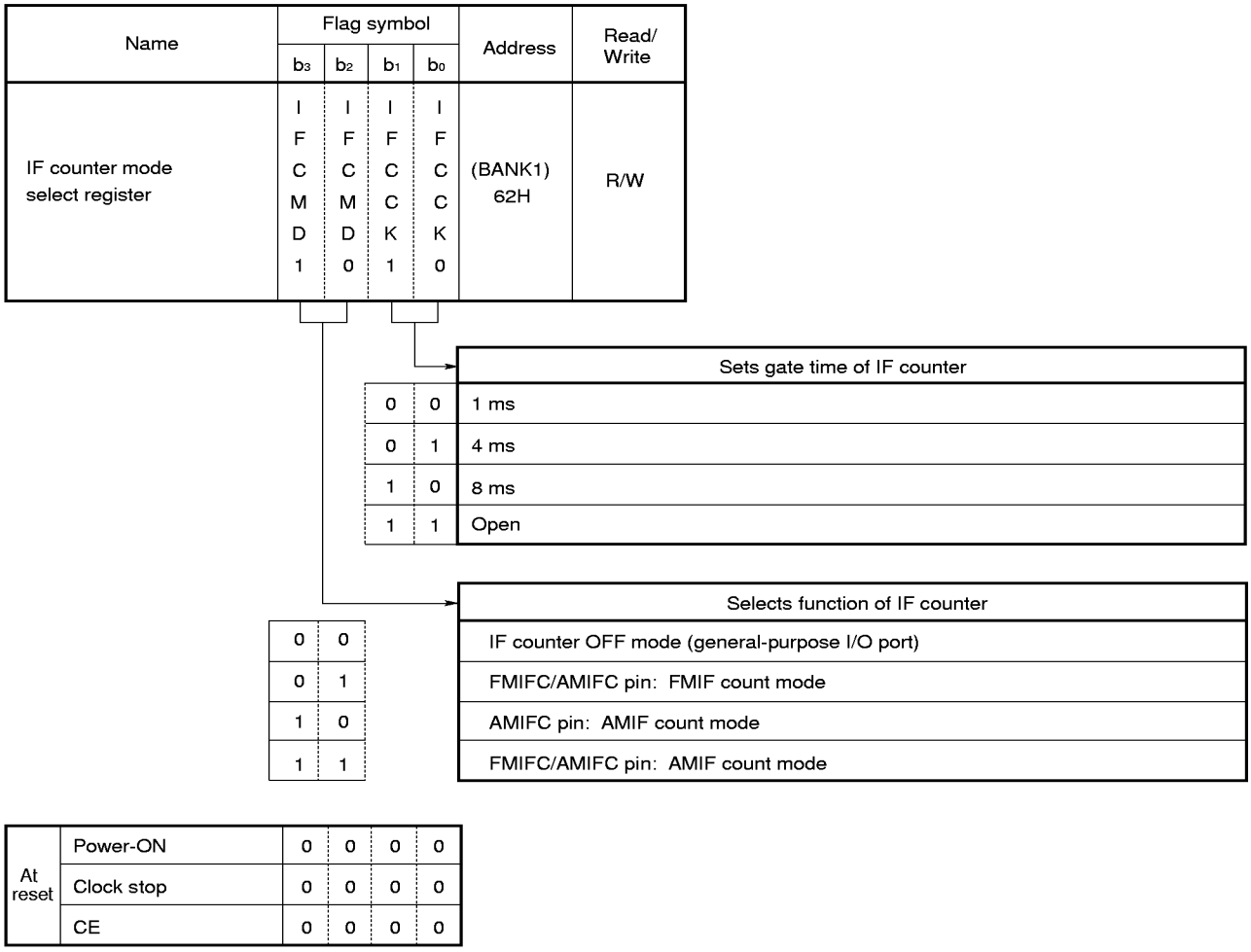


Figure 16-3. Configuration of IF Counter Mode Select Register



Name	Flag symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
IF counter mode select register	IFCMD1	IFCMD0	IFCCK1	IFCCK0	(BANK1) 62H	R/W

Sets gate time of IF counter	
0	0	1 ms
0	1	4 ms
1	0	8 ms
1	1	Open

Selects function of IF counter	
0	0	IF counter OFF mode (general-purpose I/O port)
0	1	FMIFC/AMIFC pin: FMIF count mode
1	0	AMIFC pin: AMIF count mode
1	1	FMIFC/AMIFC pin: AMIF count mode

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

16.3 Start Control Block and IF Counter

16.3.1 Configuration of start control block and IF counter

Figure 16-4 shows the configuration of the start control block and IF counter.

The start control block starts counting of the frequency counter and detects the end of counting.

The counter is started by the IF counter control register.

The end of counting is detected by the IF counter gate status detection register.

Figure 16-5 shows the configuration and function of the IF counter control register.

Figure 16-6 shows the configuration and function of the IF counter gate status detection register.

The 16.3.2, describe the gate operations when the IF counter function is selected.

The IF counter is a 16-bit binary counter that counts up the input frequency when the IF counter function is selected.

When the IF counter function is used, the IF counter counts the frequency input to the pin while the gate is opened by an internal gate signal. Although this frequency is counted as is in the AMIF count mode, it is halved and then counted in the FMIF count mode.

The IF counter is cleared to 0000H when its count value has reached FFFFH, and then continues counting.

The count value is read via data buffer by the IF counter data register (IFC).

The count value is reset by the IF counter control register.

Figure 16-7 shows the configuration and function of the IF counter data register.

Figure 16-4. Configuration of Start Control Block and IF Counter

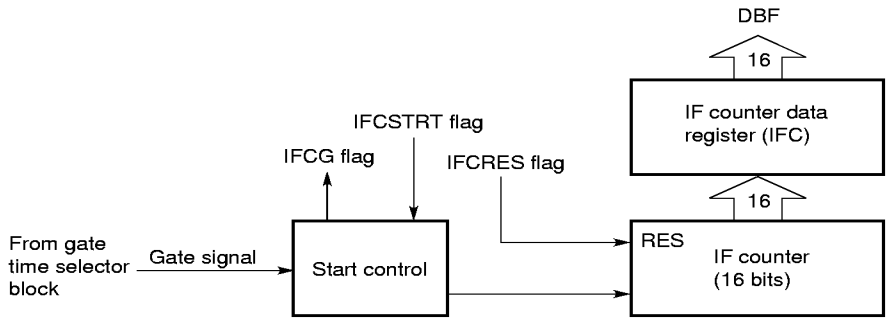
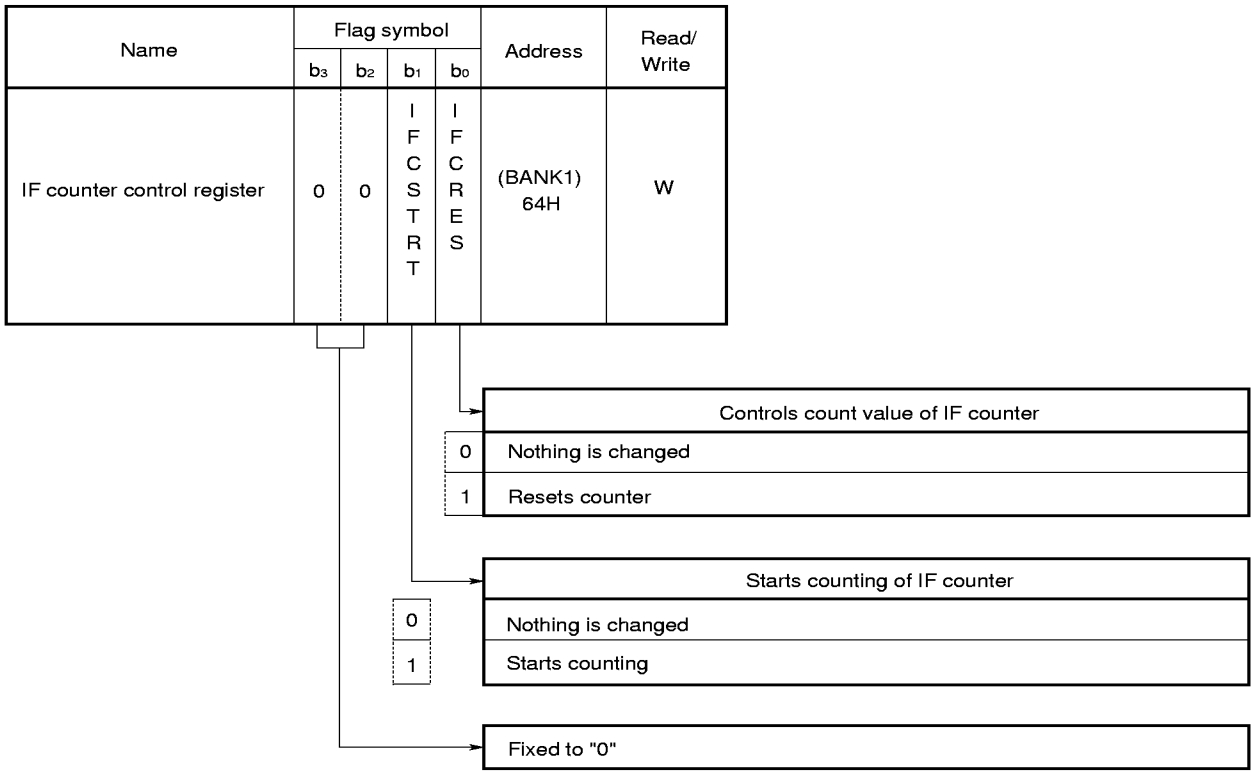
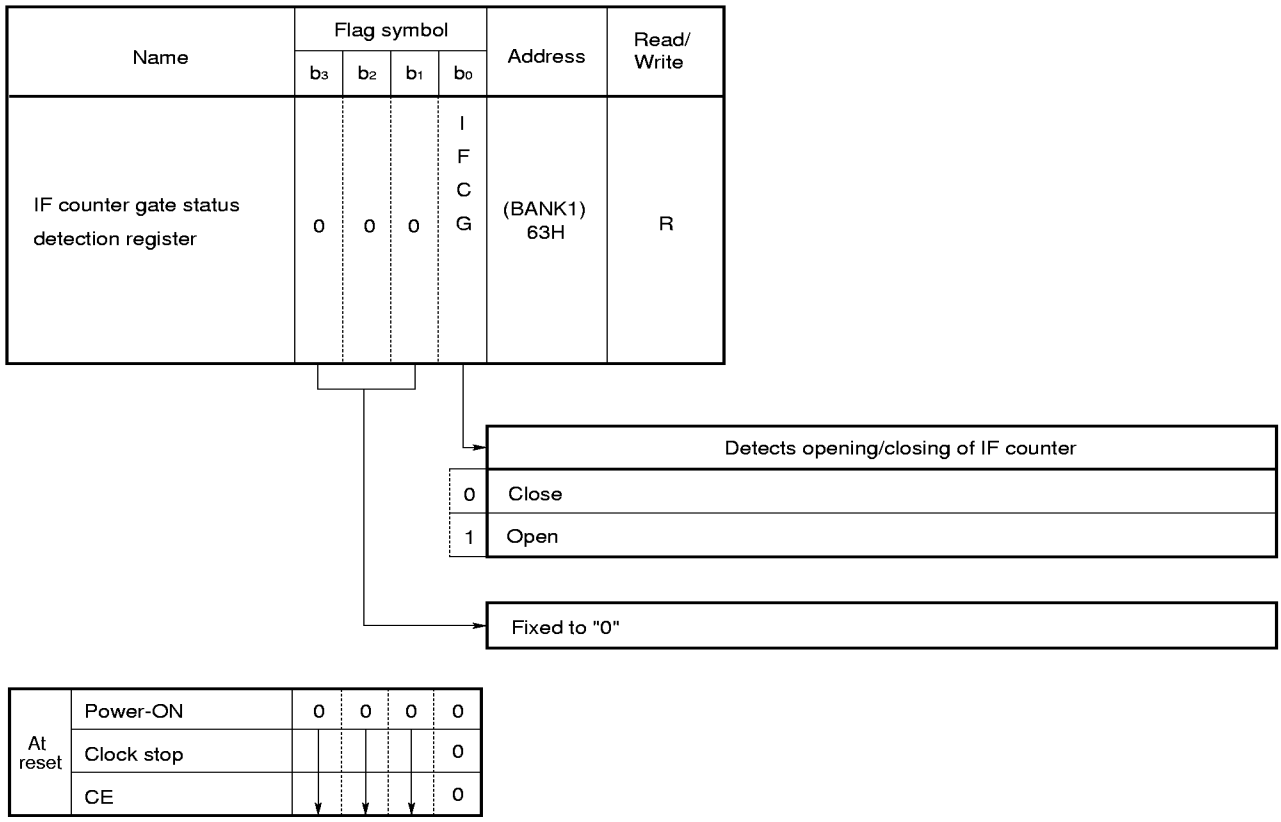


Figure 16-5. Configuration of IF Counter Control Register



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

Figure 16-6. Configuration of IF Counter Gate Status Detection Register



Caution When the IFCG flag is set to 1 (the gate is open), do not read the contents of the IF counter data register (IFC) to the data buffer.

16.3.2 Gate operation of IF counter function

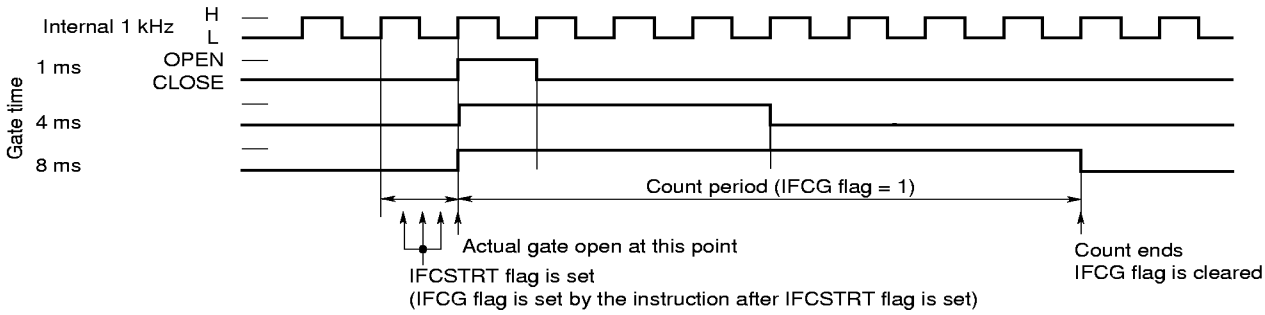
(1) When gate time is set to 1, 4, or 8 ms

As shown below, the gate is opened for 1, 4, or 8 ms starting from the rising edge of an internal 1-kHz signal after the IFCSTRT flag has been set.

While the gate is open, the frequency of a signal input to a specific pin is counted by a 16-bit counter.

When the gate is closed, the IFCG flag is cleared (0).

The IFCG flag is automatically set to "1" as soon as the IFCSTRT flag has been set.



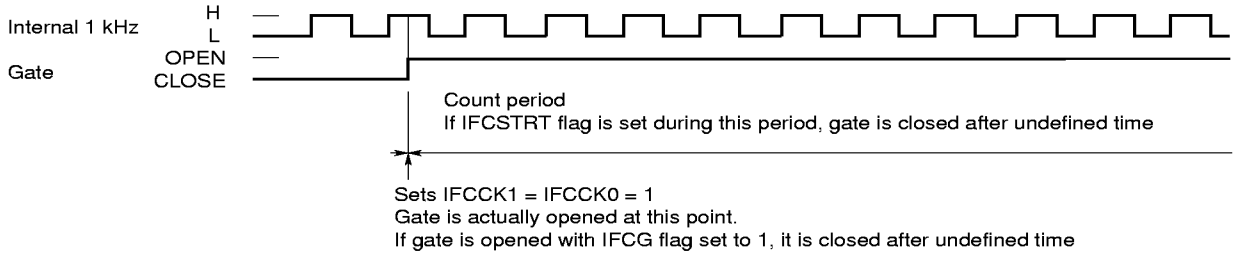
(2) When gate time is open

When the gate time is specified by the IFCCK1 and IFCCK0 flags to be open, the gate is immediately opened.

If counting is started by the IFCSTRT flag while the gate is open, the gate is closed after an undefined time.

Therefore, do not set the IFCSTRT flag to "1" when the gate time is open.

However, the counter can be reset by the IFCRES flag.



When the gate time is open, the gate can be opened and closed by resetting the gate time other than open by using IFCCK1 and IFCCK0 flags.

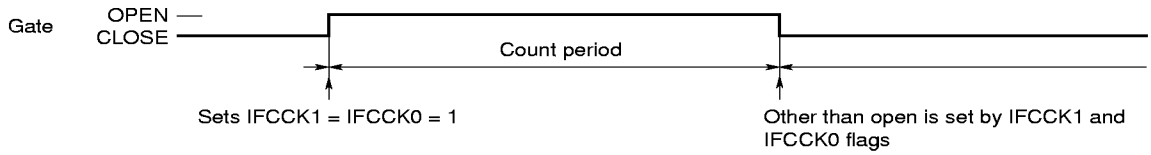
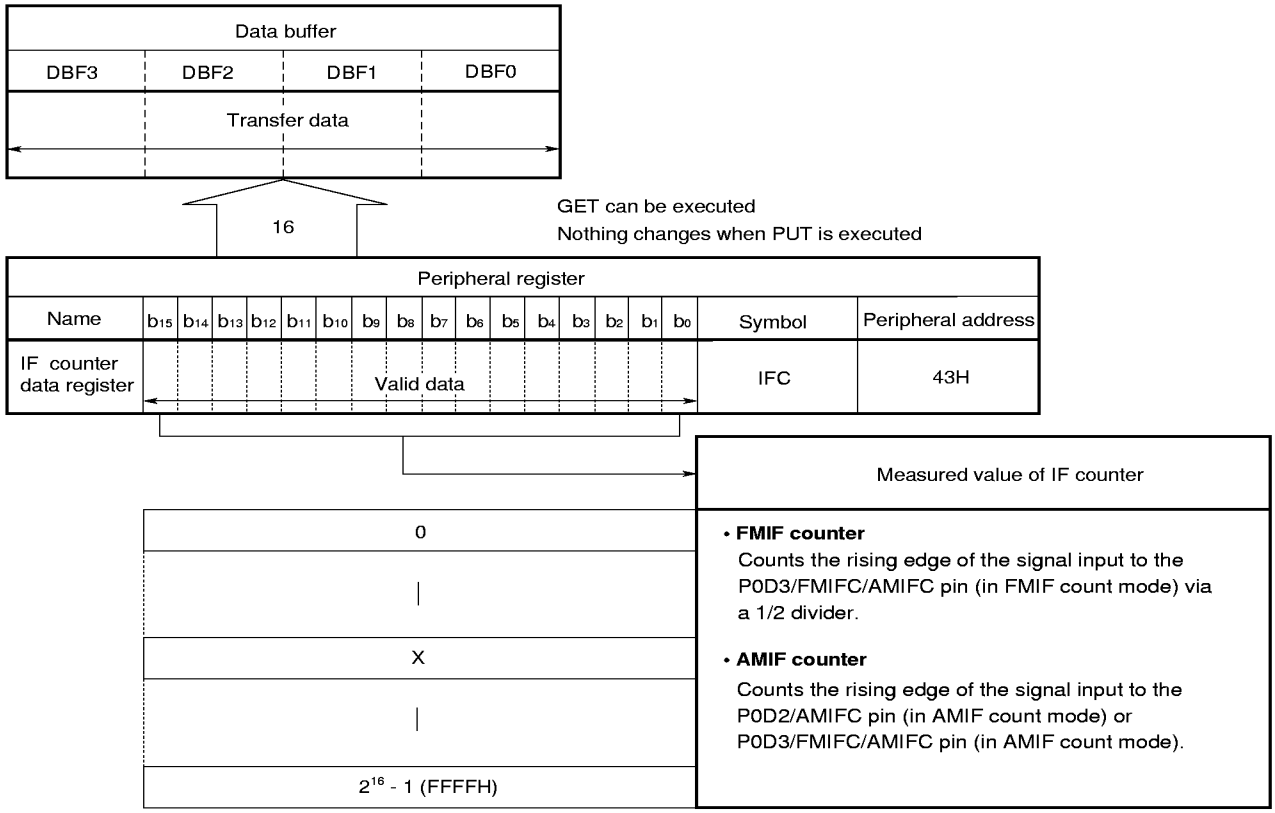


Figure 16-7. Configuration of IF Counter Data Register



16.4 Using IF Counter

The following sections 16.4.1 through 16.4.3 describe how to use the hardware of the IF counter, program example, and count error.

16.4.1 Using hardware of IF counter

Figure 16-8 shows the block diagram when the P0D2/AMIFC pin and P0D3/FMIFC/AMIFC pins are used.

Table 16-1 shows the range of frequency that can be input to the P0D2/AMIFC pin and P0D3/FMIFC/AMIFC pin.

As shown in Figure 16-8, the input pin of the IF counter is provided with an AC amplifier; therefore, cut off the DC component of the input signal with a capacitor C.

When the P0D2/AMIFC or P0D3/FMIFC/AMIFC pin is selected for the IF counter function, switch SW turns ON and the voltage applied to the pin is about $1/2V_{DD}$.

If the intermediate voltage does not rise sufficiently at this time, the AC amplifier is not in the normal operating range; consequently, the IF counter does not function normally.

Therefore, provide a sufficient wait time after each pin has been specified to be used for the IF counter function, until counting is started.

When using the IF counter function for the auto tuning function of a radio to detect broadcasting stations, it is recommended to use the function with the SD (Station Detection) output, and so on, of the tuner.

Figure 16-8. Block Diagram of IF Count Function of Each Pin

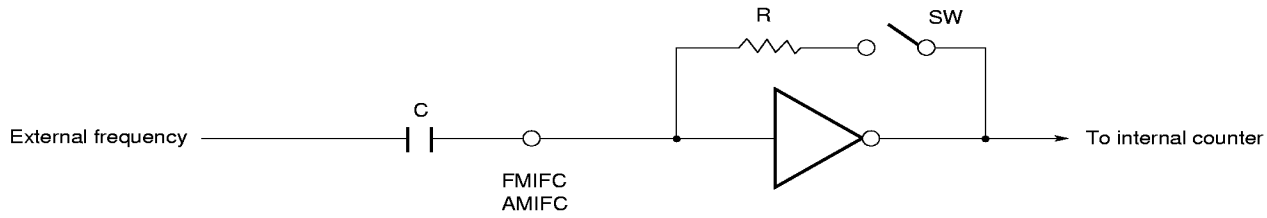


Table 16-1. IF Counter Input Frequency Range

Input pin	Input frequency (MHz)	Input amplitude (V_{P-P})
P0D3/FMIFC/AMIFC FMIF mode	10 - 11	0.1
P0D3/FMIFC/AMIFC	0.4 - 2	0.15
AMIF mode	0.4 - 0.5	0.1
P0D2/AMIFC	0.4 - 2	0.15
AMIF mode	0.4 - 0.5	0.1

16.4.2 Program example of IF counter

A program example of the IF counter is shown below.

As shown in this example, make sure that a wait time of a certain length elapses after an instruction that specifies the P0D2/AMIFC or P0D3/FMIFC/AMIFC pin to be used for the IF counter has been executed, until the counter is actually started.

This is because the internal AC amplifier is not ready for normal operation immediately after the pin has been selected for the IF counter function, as described in 16.4.1.

Example To count frequency of P0D3/FMIFC/AMIFC pin (FMIF count mode) (gate time: 8 ms)

```
        BANK1
        INITFLG   IFCMD1, NOT IFCMD0, IFCCK1, NOT IFCCK0
                                ; Selects FMIFC pin (FMIF count mode) and sets gate time to 8 ms
        [  Wait  ]              ; Internal AC amplifier stabilization time
        SET1      IFCRES        ; Resets IF counter
        SET1      IFCSTRT       ; Starts IF count
LOOP:
        SKT1      IFCG          ; Detects opening/closing of gate
        BR        READ          ; Branches to READ: if gate is closed
        [Processing A]          ; Do not read data of IF counter with this processing A.
        BR        LOOP
READ:
        GET       DBF, IFC      ; Reads value of IF counter data register to data buffer
```

16.4.3 Error of IF counter

The IF counter has a gate time error and count error, as described in (1) and (2) below.

(1) Gate time error

The gate time of the IF counter is created by dividing the 75-kHz system clock frequency.
Therefore, if this frequency has an error of “+x” ppm, the gate time accordingly has an error of “-x” ppm.

(2) Count error

The IF counter counts frequency at the rising edge of the input signal.
Therefore, if a high-level signal is input to the pin when the gate is opened, one extra pulse is counted.
However, this extra pulse may not be counted, depending on the status of the pin, when the gate is closed.
Therefore, the count error is “+1, -0”.

16.5 Status at Reset

16.5.1 Power-ON reset

The P0D3/FMIFC/AMIFC and P0D2/AMIFC pins are set in the general-purpose input port mode.
The contents of the output latch are "0".

16.5.2 On execution of clock stop instruction

The P0D3/FMIFC/AMIFC and P0D2/AMIFC pins are set in the general-purpose input port mode.
The contents of the output latch are retained.

16.5.3 At CE reset

The P0D3/FMIFC/AMIFC and P0D2/AMIFC pins are set in the general-purpose input port mode.
The contents of the output latch are retained.

16.5.4 In halt status

The P0D3/FMIFC/AMIFC and P0D2/AMIFC pins retain the status immediately before the halt mode is set.

17. BEEP

17.1 Configuration and Function of BEEP

Figure 17-1 outlines BEEP.

BEEP outputs a clock of 1.5 kHz or 3 kHz from the BEEP pin.

The output select block selects, by using the BEEP0CK0 and BEEP0CK1 flags of the BEEP clock select register, whether 1.5 kHz or 3 kHz is output from the BEEP pin, or whether the BEEP pin is used as a 1-bit general-purpose output port.

The clock generation block generates the 1.5-kHz or 3-kHz clock to be output to the BEEP pin.

Figure 17-2 shows the configuration and function of the BEEP clock select register.

Figure 17-1. Outline of BEEP

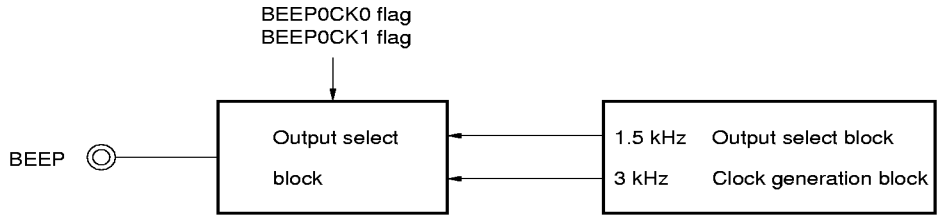


Figure 17-2. Configuration and Function of BEEP Clock Select Register

Name	Flag symbol b_3	b_2	b_1	b_0	Address	Read/ Write
BEEP clock select register	0	0	BEEP0CK1	BEEP0CK0	(BANK1) 5BH	R/W

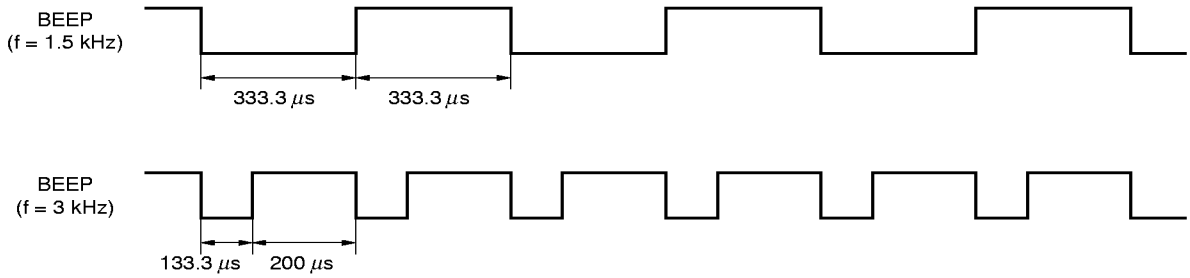
Setting of BEEP pin	
0	0	Used as general-purpose output port and outputs low level
0	1	Used as general-purpose output port and outputs high level
1	0	Outputs 1.5 kHz clock
1	1	Outputs 3 kHz clock

Fixed to "0"

At reset	Power-ON	0	0	0	0
	Clock stop			0	0
	CE			Retained	

17.2 Output Wave Form of BEEP

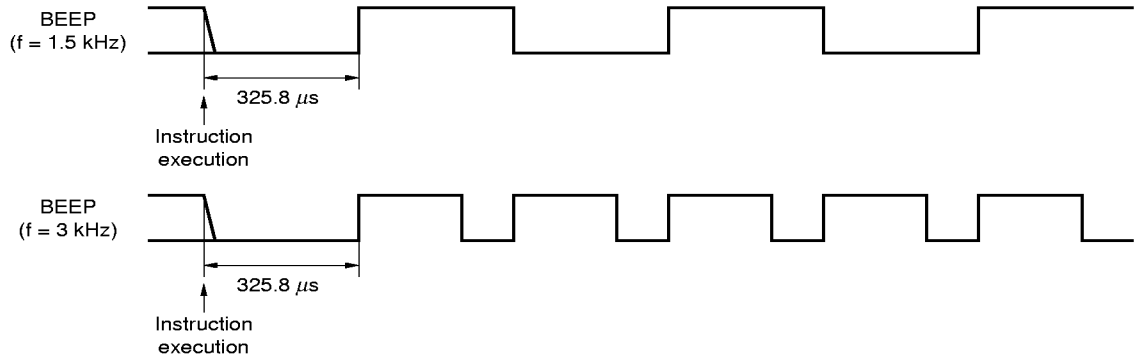
(1) Output wave of f = 1.5 kHz and f = 3 kHz



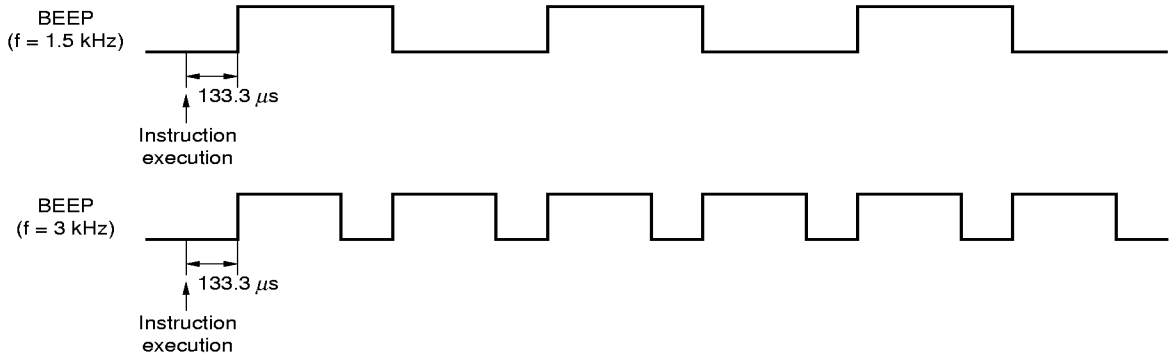
Example Program to output 3-kHz clock from BEEP pin

```
BANK1                ; Same as MOV BANK, #0001B
MOV     5BH, #0011B  ; Writes 0011B to data memory address 5BH
                     ; Outputs 3 kHz from BEEP pin
```

(2) Maximum time until clock is output from BEEP pin after instruction execution



(3) Minimum time until clock is output from BEEP pin after instruction execution



17.3 Status at Reset

17.3.1 At power-ON reset

The BEEP pin is set in the general-purpose output port mode, and outputs a low level.
The value of the latch of the output port is "0".

17.3.2 On execution of clock stop instruction

The BEEP pin is set in the general-purpose output port mode, and outputs a low level.
The value of the latch of the output port is "0".

17.3.3 At CE reset

The BEEP pin retains the previous output status.
The contents of the latch are also retained.

17.3.4 In halt status

The BEEP output pin retains the previous output status.

18. LCD CONTROLLER/DRIVER

The LCD (Liquid Crystal Display) controller/driver can display an LCD of up to 60 dots by a combination of command signal and segment signal outputs.

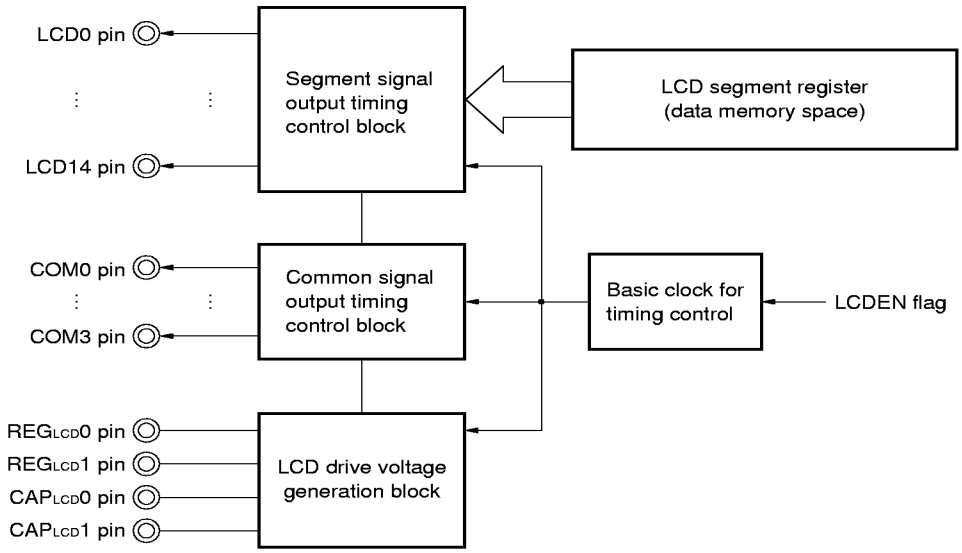
18.1 Outline of LCD Controller/Driver

Figure 18-1 outlines the LCD controller/driver.

The LCD controller/driver can be used to display up to 60 dots by using a combination of common signal output pins (COM0 through COM3) and segment signal output pins (LCD0 through LCD14).

The drive mode is 1/4 duty, 1/2 bias, the frame frequency is 62.5 Hz, and drive voltage is V_{LCD1}.

Figure 18-1. Outline of LCD Controller/Driver



Remark LCDEN (bit 3 of LCD driver display start register: refer to **Figure 18-6**) turns ON/OFF all LCD display.

18.2 LCD Drive Voltage Generation Block

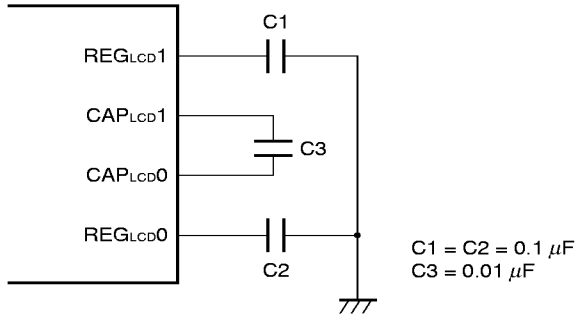
The LCD drive voltage generation block generates a voltage to drive the LCD.

The μPD17073 supplies the LCD drive voltage from an external doubler circuit. To configure a doubler circuit, connect a capacitor to the $REG_{LCD}0$, $CAP_{LCD}0$, $CAP_{LCD}1$, and $REG_{LCD}1$ pins.

Figure 18-2 shows an example of configuration of the doubler circuit. To use a voltage of 3.1 V (TYP.), connect as shown in Figure 18-2.

To operate the doubler circuit, the LCDEN flag of the LCD display start register must be set to "1". Unless this flag is set to "1", the LCD drive voltage generation block does not operate. For the LCDEN flag, refer to **18.4 Common Signal Output and Segment Signal Output Timing Control Blocks**.

Figure 18-2. Configuration of Doubler Circuit



Remark ( ): pin number

Note that, because of the configuration of the doubler circuit, the values of the LCD drive voltages (V_{LCD1} and V_{LCD0}) differ if the values of C1, C2, and C3 are changed.

18.3 LCD Segment Register

The LCD segment register sets dot data to turn on or turn off dots on the LCD.

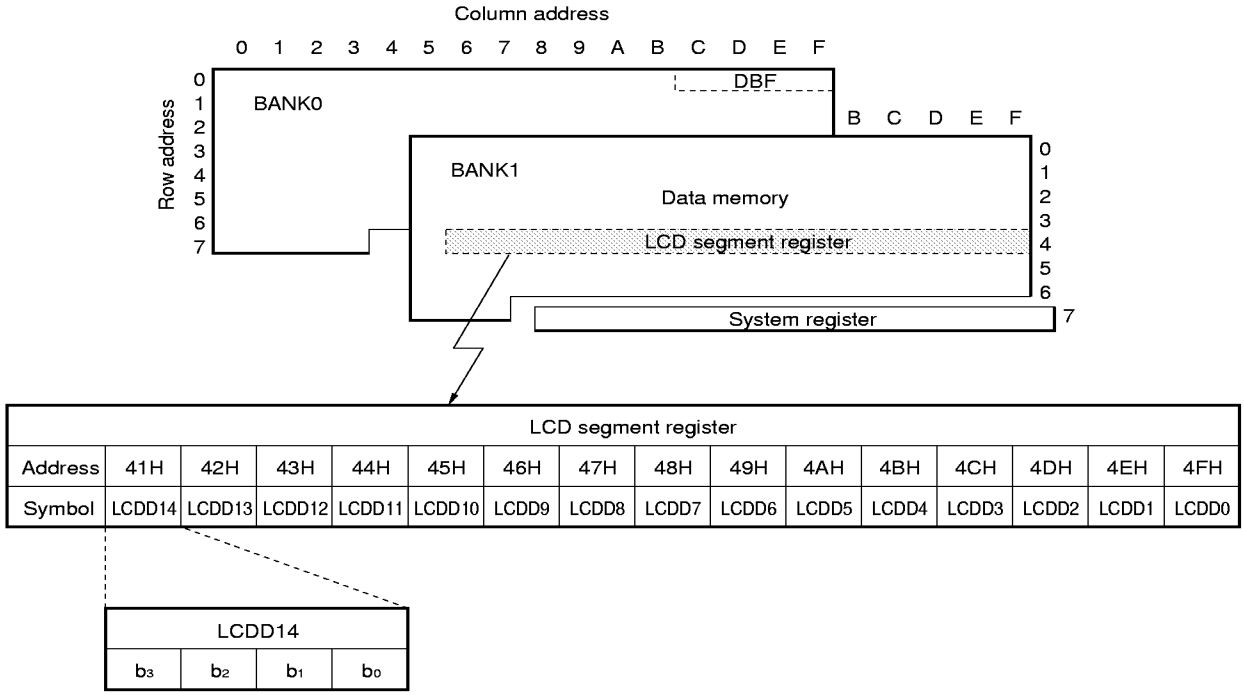
Figure 18-3 shows the location in the data memory and configuration of the LCD segment register.

Because the LCD segment register is located in data memory, it can be controlled by all the data memory manipulation instructions.

One nibble of the LCD segment register can set display data of 4 dots (data to turn dots on or off). If the LCD segment register is set to "1" at this time, the LCD display dot is on; the dot goes off if the register is set to "0".

Figure 18-4 shows the relation between the LCD segment register and LCD display dot.

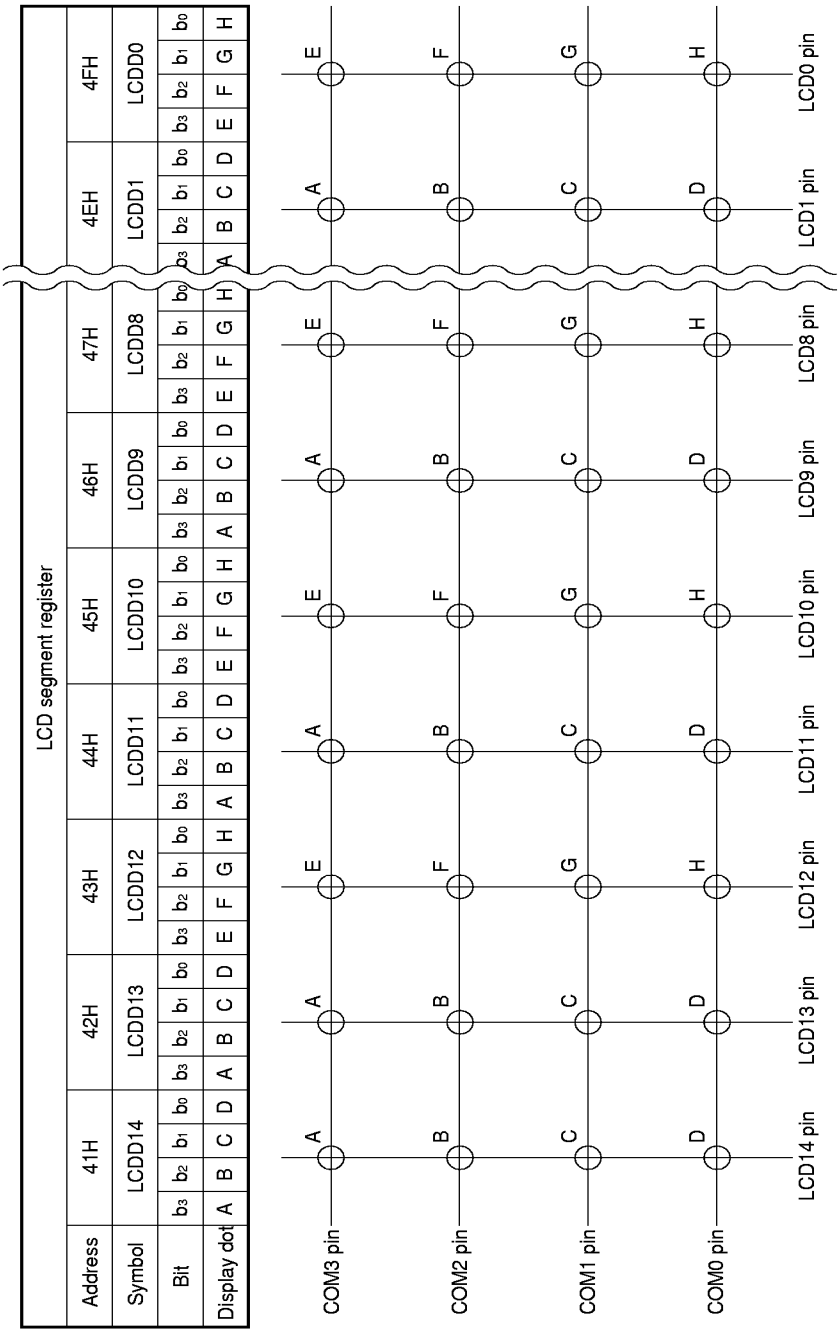
Figure 18-3. Location on Data Memory and Configuration of LCD Segment Register



LCD segment register															
Address	41H	42H	43H	44H	45H	46H	47H	48H	49H	4AH	4BH	4CH	4DH	4EH	4FH
Symbol	LCDD14	LCDD13	LCDD12	LCDD11	LCDD10	LCDD9	LCDD8	LCDD7	LCDD6	LCDD5	LCDD4	LCDD3	LCDD2	LCDD1	LCDD0

LCDD14			
b_3	b_2	b_1	b_0

Figure 18-4. Relation between LCD Segment Register and LCD Display Dot



18.4 Common Signal Output and Segment Signal Output Timing Control Blocks

Figure 18-5 shows the common signal output and segment signal output timing control blocks.

The common signal output timing control block controls the common signal output timing of the COM0 through COM3 pins.

The segment signal output timing control block controls the segment signal output timing of the LCD0 through LCD14 pins.

The common and segment signals are output when the LCDEN flag of the LCD driver display start register is set to "1".

When this flag is reset to "0", all the LCD display dots can be extinguished (refer to **Figure 18-6**).

When LCD display is not carried out, the COM0 through COM3 and LCD0 through LCD14 pins output low level.

Figure 18-5. Configuration of Common Signal Output and Segment Signal Output Timing Control Blocks

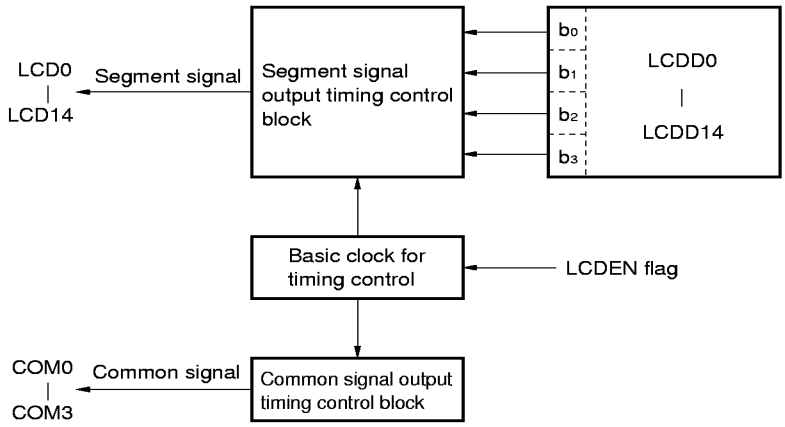
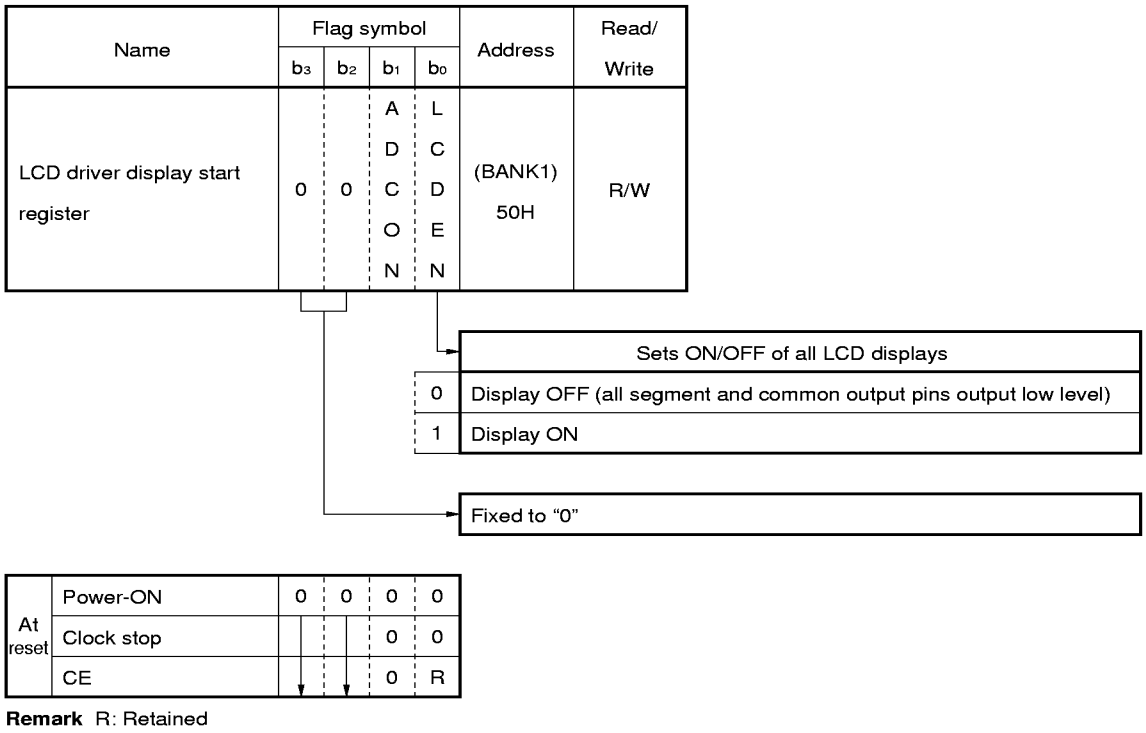


Figure 18-6. Configuration of LCD Driver Display Start Register



Remark R: Retained

Cautions 1. Bit 3 of the LCD display start register is a test mode area. Therefore, do not write "1" to this bit.

2. For the function of the ADCON flag, refer to 13.2 Setting of A/D Converter Power Supply.

18.5 Common Signal and Segment Signal Output Waves

Figure 18-7 shows an example of the common signal and segment signal output waves.

The μPD17073 outputs a signal with a frame frequency of 62.5 Hz using a 1/4 duty, 1/2 bias (voltage average method) drive mode.

As the common signals, the COM0 through COM3 pins output three levels of voltages (GND, V_{LCD0}, and V_{LCD1}) each having a phase difference of 1/8 from the others. In other words, voltages of $\pm 1/2V_{DD}$ are output with the V_{LCD0} as the reference. This display method is called the 1/2 bias drive method.

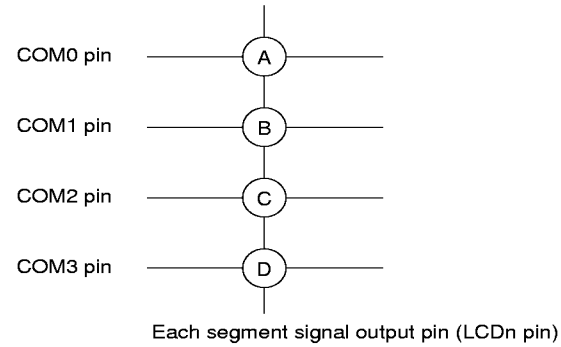
As the segment signals, the segment signal output pins output voltages of two levels (GND and V_{LCD1}) having a phase corresponding to each display dot. Because one segment pin can turn on or off four display dots (A, B, C, and D) as shown in figure 18-7, sixteen phases can be output by combining lighting and extinguishing of each dot.

Each display dot turnd on when the potential difference between a common signal and a segment signal is V_{LCD1}.

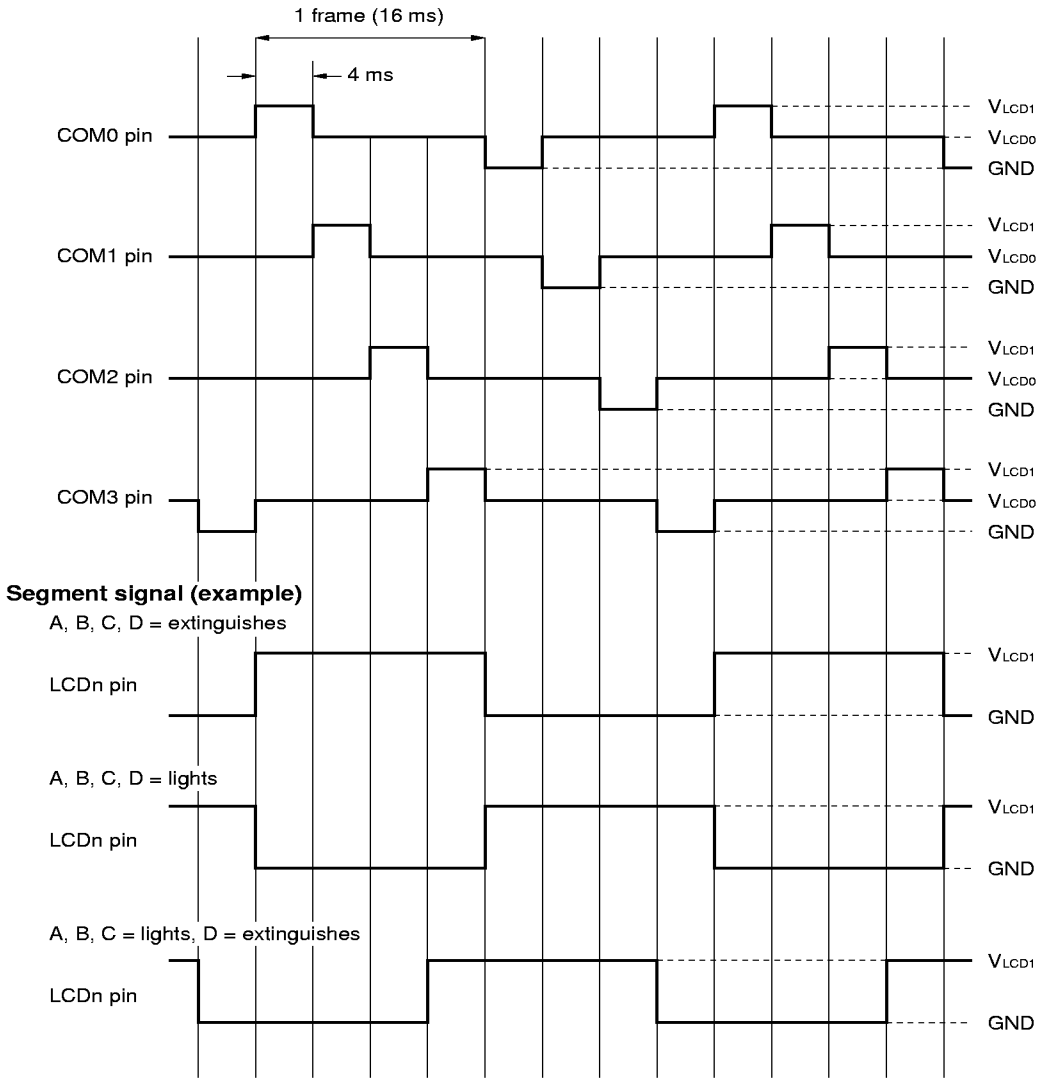
In other words, the duty factor at which each display dot turns on is 1/4.

This display method is called the 1/4 duty display method, and the frame frequency is 62.5 Hz.

Figure 18-7. Common Signal and Segment Signal Output Waves



Common signal



18.6 Using LCD Controller/Driver

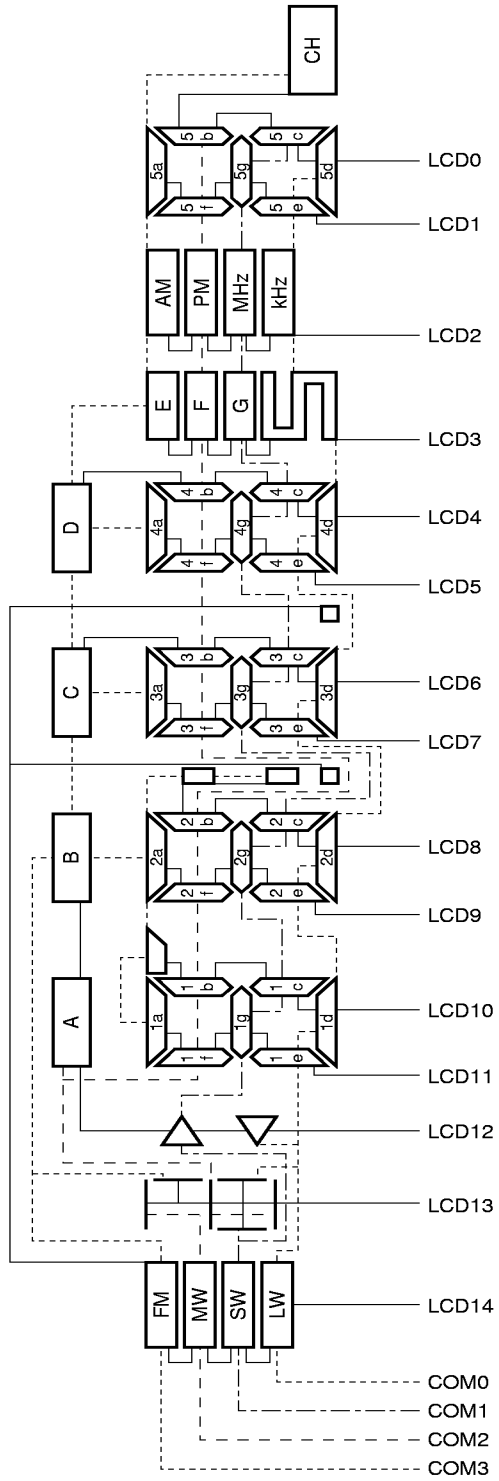
Figure 18-8 shows an example of wiring of an LCD panel

An example of a program that lights the 7 segments connected to LCD0 and LCD1 pins shown in Figure 18-8 is given below.

Example

```
PMN0    MEM     0.01H                   ; Preset number storage area
CH      FLG     LCDD0.3                 ; Defines symbol with high-order 1 bit of LCD0 register for 'CH' display

LCDDATA:                                ; LCD segment table data

        DW      0000000000000000B       ; BLANK
        DW      0000000000000110B       ; 1
        DW      0000000010110101B       ; 2
        DW      0000000010100111B       ; 3
        DW      0000000001100110B       ; 4
        DW      0000000011100011B       ; 5
        DW      0000000011110011B       ; 6
        DW      0000000010000110B       ; 7
        DW      0000000011110111B       ; 8
        DW      0000000011100111B       ; 9

        MOV     AR0, #.DL.LCDDATA SHR 12 AND 0FH
        MOV     AR1, #.DL.LCDDATA SHR  8 AND 0FH
        MOV     AR2, #.DL.LCDDATA SHR  4 AND 0FH
        MOV     AR3, #.DL.LCDDATA        AND 0FH

        LD      DBF0, AR0
        LD      DBF1, AR1
        LD      DBF2, AR2
        LD      DBF3, AR3

        ADD     DBF0, PMN0
        ADDC    DBF1, #0
        ADDC    DBF2, #0
        ADDC    DBF3, #0

        ST      AR0, DBF0
        ST      AR1, DBF1
        ST      AR2, DBF2
        ST      AR3, DBF3

        MOVT    DBF, @AR                ; Table reference instruction

        BANK1
        ST      LCDD0, DBF0
        ST      LCDD1, DBF1
        SET1    CH
        SET1    LCDEN                   ; LCD ON
```

Figure 18-8. Example of Wiring of LCD Panel



Correspondence of Segment and Common Pins, and LCD Panel Display

Segment Pin / Common	LCDD 14	LCDD 13	LCDD 12	LCDD 11	LCDD 10	LCDD 9	LCDD 8	LCDD 7	LCDD 6	LCDD 5	LCDD 4	LCDD 3	LCDD 2	LCDD 1	LCDD 0
COM3	FM		B	1a		2a	·	3a	C	4a	D	E	AM	5a	CH
COM2	MW		A	1f	1b	2f	2b	3f	3b	4f	4b	F	PM	5f	5b
COM1	SW		△	1g	1c	2g	2c	3g	3c	4g	4c	G	MHz	5g	5c
COM0	LW		▽	1e	1d	2e	2d	3e	3d	4e	4d		kHz	5e	5d

18.7 Status at Reset

18.7.1 At power-ON reset

The LCD0 through LCD14 pins output a low level.
The COM0 through COM3 pins also output a low level.
Therefore, the LCD display is OFF.
The contents of the LCD segment register are undefined.

18.7.2 On execution of clock stop instruction

The LCD0 through LCD14 pins output a low level.
The COM0 through COM3 pins also output a low level.
Therefore, the LCD display is OFF.
The LCD segment register retains the previous contents.

18.7.3 At CE reset

The LCD0 through LCD14 pins output segment signals.
The COM0 through COM3 pins output common signals.
The LCD segment register retains the previous contents.

18.7.4 In halt status

The LCD0 through LCD14 pins output segment signals.
The COM0 through COM3 pins output common signals.
The LCD segment register retains the previous contents.

19. STANDBY

The standby function is used for the purpose of reducing the current consumption of the device when the device is in the backup status.

19.1 General

Figure 19-1 shows the outline of the standby block.The standby function is to reduce the current consumption of the device by stopping part of or entire device, or slowing down the CPU clock.

The standby function can be used in the following four modes, which can be selected according to the application:

<1> Halt mode
<2> Clock stop mode
<3> Controlling device operation by CE pin
<4> Low-speed function

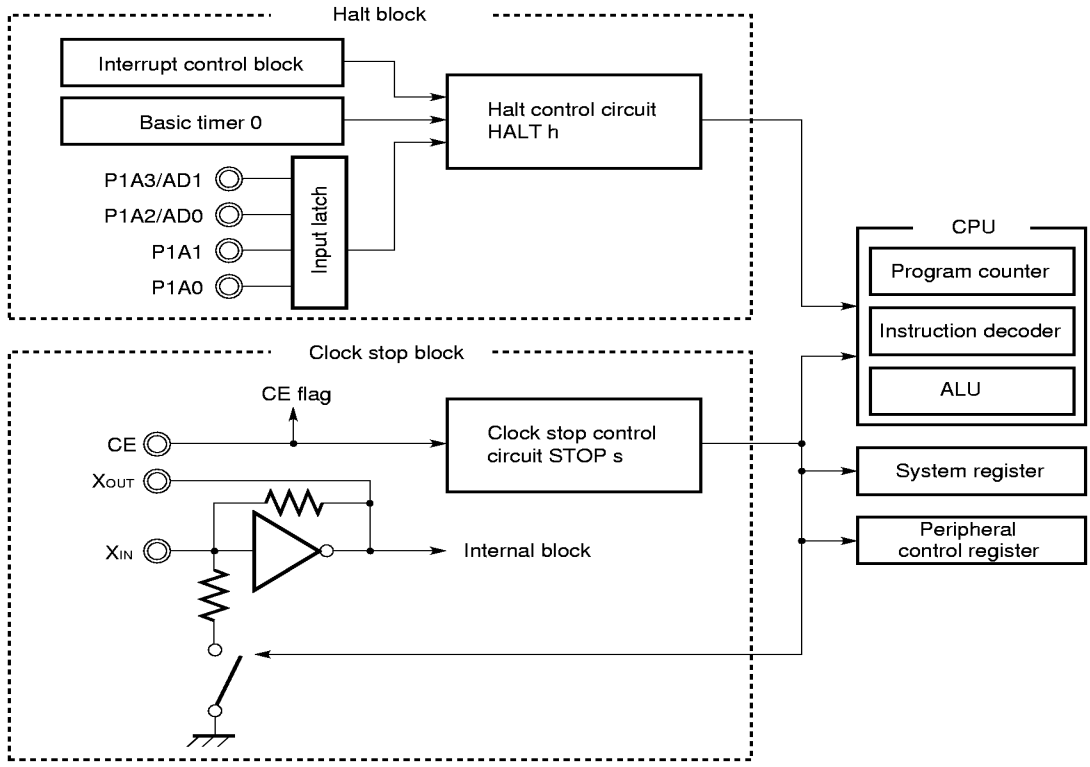
The halt mode is to reduce the current consumption of the device by stopping the operation of the CPU when a dedicated instruction, “HALT h”, has been executed.

The clock stop mode is to reduce the current consumption of the device by stopping the oscillation of the oscillator circuit when a dedicated instruction, “STOP s”, has been executed.

The CE pin is usually used to control the operation of the PLL frequency synthesizer and to reset the device. However, it can be said to be a mode of the standby function in that this pin controls operations.

The low-speed function is to reduce the current consumption of the device by slowing down the CPU clock.

Figure 19-1. Outline of Standby Block



Remark CE flag (bit 0 of CE pin status detection register. Refer to **Figure 19-6**) detects the status of the CE pin.

19.2 Halt Function

19.2.1 General

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

When this instruction has been executed, the program is stopped and is not executed unless the halt status is released. Therefore, the current consumption of the device is reduced by the operating current of the CPU in the halt status.

The halt status is released by key input, basic timer 0, or interrupt.

The releasing condition is specified by the operand “h” of the HALT h instruction.

The HALT h instruction is valid regardless of the input level of the CE pin.

19.2.2 Halt status

In the halt status, all the operations of the CPU are stopped. In other words, the program execution is stopped by the “HALT h” instruction. However, the peripheral hardware retains the status set before the HALT h instruction is executed.

For the operation of each peripheral hardware, refer to **19.4 Device Operations in Halt and Clock Stop Statuses**.

19.2.3 Halt release condition

Figure 19-2 shows the halt release conditions.

The halt release condition is set by 4-bit data that is specified by the operand “h” of the HALT h instruction.

The halt status is released when the condition specified as “1” in operand “h” is satisfied.

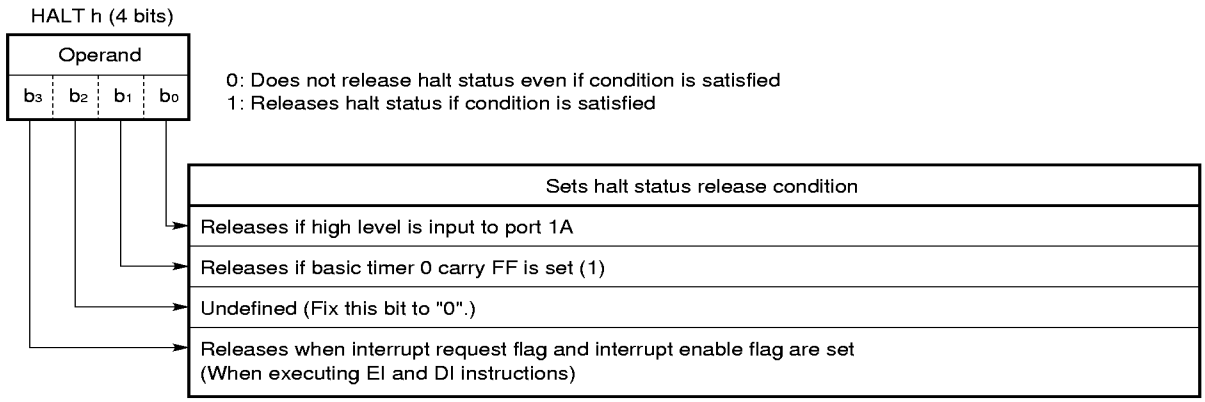
When the halt status has been released, program execution is started from the instruction next to "HALT h" instruction.

If two or more release conditions are specified, the halt status is released if any one of the specified conditions has been satisfied.

When the device has been reset (by means of power-ON reset or CE reset), the halt status is released, and the appropriate reset operation is performed.

If 0000B is set as the halt release condition “h”, no release condition is set. In this case, the halt status is released when the device is reset (power-ON reset or CE reset).

Figure 19-2. Halt Release Condition



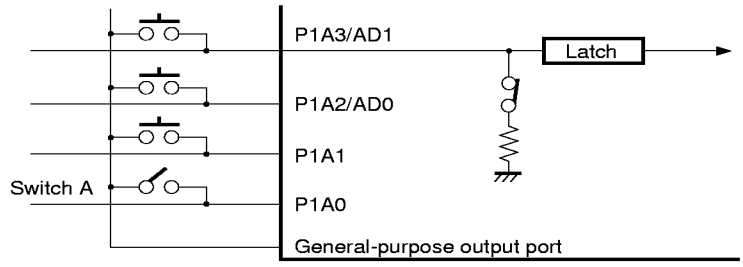
19.2.4 Releasing halt by key input

To release the halt mode by key input, the HALT instruction is specified as "HALT 0001B".

With the key input specified as the halt release condition, the halt mode is released when a high-level signal is input to any one of the P1A0, P1A1, P1A2/AD0, and P1A3/AD1 pin.

However, halt mode cannot be released by a pin disconnected to the pull-down resistor.

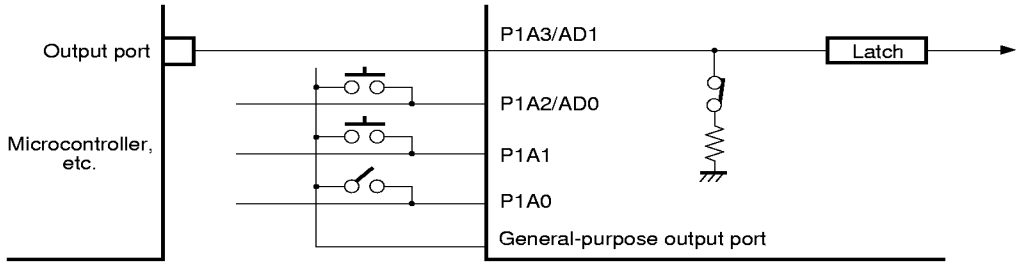
(1) When using general-purpose output port as key source signal



To use a general-purpose output port as the key source signal, make the output port high-level, and execute the "HALT 0001B" instruction.

When an alternate switch is used at this time as switch A in the above figure, a high-level is always input to the P1A0 pin while switch A is closed, and the halt mode is immediately released. Therefore, care must be exercised when key input is specified as the halt mode releasing condition and an alternate switch is used.

(2) To release halt by other microcontroller



The P1A0, P1A1, P1A2/AD0, and P1A3/AD1 pins can also be used as general-purpose input ports with pull-down resistor.

Therefore, other microcontrollers can also be used as shown above after the halt mode has been released.

19.2.5 Releasing halt status with basic timer 0

To release the halt condition by using the basic timer 0, use the "HALT 0010B" instruction.

When it has been set that the halt status is to be released by the basic timer 0, the basic timer 0 carry FF is set to 1, and at the same time, the halt status is released.

The basic timer 0 carry FF corresponds to the BTM0CY flag on a one-to-one basis and is set at fixed time intervals (125 ms). Therefore, the halt status can be released at specific time intervals.

Example To release halt status every 125 ms and perform processing A every 1 second

```
M1      MEM     0.10H           ; 1-second counter
HLTTMR  DAT     0010B           ; Symbol definition
LOOP:
        HALT    HLTTMR          ; Specifies that halt status is released by basic
                                  timer 0 carry FF, and sets halt status
        BANK1
        SKT1    BTM0CY          ; Embedded macro
        BR      LOOP            ; Branches to LOOP if BTM0CY flag is not set
        BANK0
        ADD     M1, #0010B      ; Adds 0010B to contents of M1
        SKT1    CY              ; Embedded macro
        BR      LOOP            ; Executes processing A if carry occurs
        Processing A
        BR      LOOP
```

19.2.6 Releasing halt status by interrupt

To release the halt status by interrupt, use the "HALT 1000B" instruction.

There are three interrupt sources available as explained in 11. INTERRUPT. Therefore, the interrupt by which the halt status is to be released must be specified by software.

The halt status is released if the following conditions (1) through (3) are satisfied:

(1) The "HALT 1000B" instruction is set.
(2) Each interrupt is enabled by the corresponding interrupt enable flag (IP××× flag = 1).
(3) An interrupt request is issued by the corresponding interrupt request flag (IRQ××× flag = 1).

Depending on whether the EI or DI instruction is executed at this time, the operation to be performed after the halt status has been released differs.

If the EI instruction is executed, the program branches to the vector address of the interrupt. If the RETI instruction is executed after the interrupt has been serviced, the program returns to the next instruction after the HALT instruction.

If the DI instruction is executed, the program does not branch to a vector address, but the next instruction next after the HALT instruction is executed as soon as the halt status has been released.

Examples of programs when the EI and DI instruction are executed, and notes on releasing the halt status by interrupt are described below.

Example 1. Example of program when EI instruction is executed

```
HLTINT    DAT     1000B          ; Symbol definition for halt  mode
INTTM     DAT     0002H          ; Defines symbol of interrupt vector address
INTPIN    DAT     0003H          ; Defines symbol of interrupt vector address

START:
          BR      MAIN           ; Program address 0000H
ORG       INTTM                  ; Basic timer 1 interrupt vector address
          BR      INTTIMER
ORG       INTPIN                 ; Interrupt service by INT pin
          Processing A
          BR      EI_RETI
INTTIMER:                        ; Interrupt service by basic timer 1
          Processing B
EI_RETI:
          EI
          RETI
MAIN:
          BANK1
          SET2    IPBTM, IP      ; Embedded macro
          SET1    BTM1CK         ; Sets time interval of basic timer 1 to 8 ms
LOOP:
          Processing C           ; Main routine processing
          EI                     ; Enables all interrupts
          HALT    HLTINT         ; Sets releasing halt mode by interrupt
        ; <1>
          BR      LOOP
```

In this example, the halt mode is released when an interrupt by basic timer 1 has been accepted, processing B is executed, and processing A is executed when an interrupt by INT pin has been accepted.

Each time the halt mode is released, processing C is executed.

If the interrupt request by INT pin and interrupt request by basic timer 1 are issued exactly at the same time in the halt mode, processing A of INT pin, which is assigned the higher hardware priority, is executed.

When “RETI” is executed after execution of processing A, the execution is returned to the “BR LOOP” instruction in <1>. However, the “BR LOOP” instruction is not executed, but the basic timer 1 interrupt is accepted immediately, and processing B is executed.

If the “RETI” instruction is executed after processing B, the “BR LOOP” instruction is executed.

Example 2. Example of program when DI instruction is executed

```
HLTINT   DAT     1000B              ; Symbol definition of halt condition

START:
         DI                         ; Disables all interrupts
         BANK1
         SET2    IPBTM1, IP         ; Embedded macro
         SET1    BTM1CK             ; Sets time of interrupt by basic timer 1 to 8 ms


LOOP:
         HALT    HLTINT             ; Sets releasing halt status by interrupt

         SKT1    IRQ                ; Detects halt release trigger
         BR      INTBTM1

         CLR1    IRQ                ;
         Processing A               ; Interrupt service by INT pin
INTBTM1:
         SKT1    IRQBTM1            ; Detects halt release trigger
         BR      LOOP

         CLR1    IRQBTM1            ;
         Processing B               ; Interrupt service by basic timer 1

         BR      LOOP
```

Because the DI instruction is executed in the above example, the program does not branch to the respective vector addresses but executes the next instruction even if interrupt by basic timer 1 or INT pin is accepted.

Caution When executing the HALT instruction that is released when an interrupt request flag (IRQ×××), for which the corresponding interrupt enable flag (IP×××) is set, is set, describe a NOP instruction immediately before the HALT instruction.
When the NOP instruction is described immediately before the HALT instruction, time of one instruction is generated between the IRQ××× manipulation instruction and HALT instruction. In the case of the CLR1 IRQ××× instruction, for example, clearing IRQ××× correctly is reflected upon the HALT instruction (Example 1 below). If the NOP instruction is not described immediately before the HALT instruction, the CRL1 IRQ××× instruction is not correctly reflected on the HALT instruction, and the HALT mode is not set (Example 2).

Examples 1. To execute HALT instruction correctly

```
                :
                :
                :           ; Sets IRQ×××
                :
                :
        CLR1    IRQ×××
        NOP                 ; NOP instruction is described immediately before HALT instruction
                            ; (Clearing IRQ××× is correctly reflected on HALT instruction)
        HALT    1000B       ; HALT instruction is executed correctly (HALT mode is set)
                :
                :
                :
```

2. Program that does not set HALT mode

```
                :
                :
                :           ; Sets IRQ×××
                :
                :
        CLR1    IRQ×××      ; Clearing IRQ××× is not reflected on HALT instruction
                            ; (It is reflected on instruction next to HALT instruction)
        HALT    1000B       ; HALT instruction is ignored (HALT mode is not set)
                :
                :
                :
```

19.2.7 When two or more release conditions are specified

When two or more halt release conditions are specified, the halt mode is released if any one of the specified conditions is satisfied.

The following example shows how the condition is identified when two or more conditions are specified:

Example

```
            HLTINT   DAT 1000B
            HLTTMR   DAT 0010B
            HLTKEY   DAT 0001B
            INTPIN   DAT 0003H          ;  Vector address symbol definition of INT pin interrupt
START:
            BR       MAIN
ORG         INTPIN
            Processing A                ;  INT pin interrupt service
            EI
            RETI
TMRUP:                                  ;  Basic timer 0 processing
            Processing B
            RET
KEYDEC:                                 ;  Key input processing
            Processing C
            RET
MAIN:
            BANK1
            MOV      P1B, #1111B        ;  Outputs P1B3-P1B0 at high level as key source output
            SET1     IP                 ;  Embedded macro
                                        ;  Enables INT pin interrupt
            EI
LOOP:
            HALT     HLTINT OR HLTTMR OR HLTKEY
                                        ;  Specifies external interrupt (INT pin), basic timer 0,
                                        ;  and key input as halt release condition
            SKT1     BTM0CY             ;  Embedded macro
                                        ;  Detects BTM0CY flag
            BR       KEY_DEC
            CALL     TMRUP              ;  Basic timer 0 processing if set to "1"
            BR       LOOP
KEYDEC:
            Key processing
            BR       LOOP
```

19.3 Clock Stop Function

The clock stop function stops the 75 kHz crystal resonator when the "STOP s" instruction (clock stop status) is executed.

Therefore, the current dissipation of the device is reduced to 3 μA maximum (T_A = 25 °C, V_{DD} = 3.0 V).

As the operand "s" of the "STOP s" instruction, "0000B" is specified.

The "STOP s" instruction is valid only when the CE pin is at the low level, and is executed as a no-operation ("NOP") instruction when the CE is at the high level.

Therefore, the "STOP s" instruction must be executed when the CE pin is at the low level.

The clock stop mode can be released by CE reset rising the CE pin, or by power-ON reset with supply voltage V_{DD} application.

19.3.1 Clock stop status

In the clock stop status, the crystal resonator is stopped. As a result, the CPU and all the peripheral hardware stop their operations.

For the operations of the CPU and peripheral hardware, refer to **19.4 Device Operations in Halt and Clock Stop Statuses**.

19.3.2 Releasing clock stop status

The clock stop status can be released in the following two ways. After the clock stop status has been released, the program is started from address 0000H, regardless of which of (1) and (2) has been used to release the clock stop status.

(1) Raise the CE pin from low to high (CE reset).

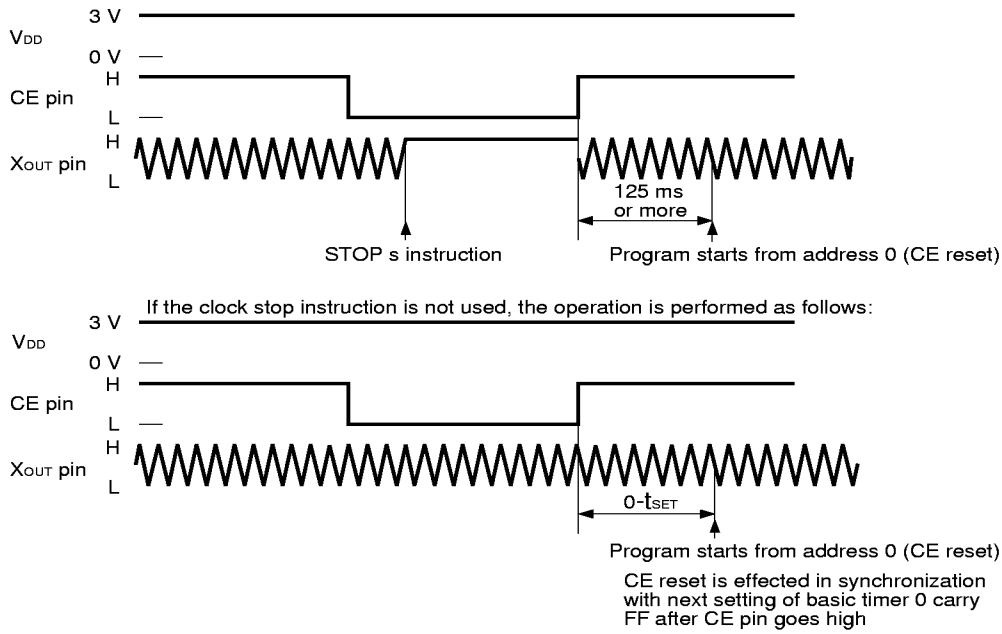
(2) Lower the supply voltage V_{DD} of the device to 1.8 V or less[Note], and then raise it again to 1.8 V or higher (T_A = –20 to +70 °C, normal operation) (power-ON reset).

Note This voltage is called power-ON clear voltage. The maximum value of the power-ON clear voltage is 1.8 V, and the actual value is in a range not exceeding this maximum value. For details, refer to **20.4.1 Power-ON clear voltage**.

19.3.3 Releasing clock stop status by CE reset

Figure 19-3 illustrates how the clock stop status is released by the CE reset.

Figure 19-3. Releasing Clock Stop by CE Reset

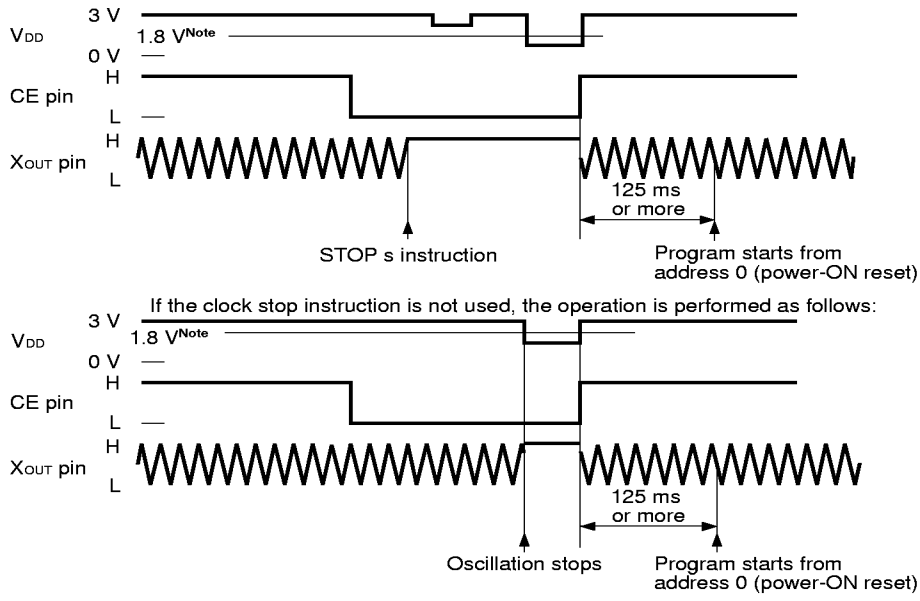


19.3.4 Releasing clock stop status by power-ON reset

Figure 19-4 illustrates how the clock stop status is released by power-ON reset.

When the clock stop status is released by power-ON reset, the power failure detection circuit operates.

Figure 19-4. Releasing Clock Stop by Power-ON Reset



Note This voltage is called power-ON clear voltage. The maximum value of the power-ON clear voltage is 1.8 V, and the actual value is in a range not exceeding this maximum value. For details, refer to **20.4.1 Power-ON clear voltage**.

19.3.5 Notes on using clock stop instruction

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

Therefore, it is necessary to design program taking into consideration the chance that the “STOP s” instruction is to be executed when the CE pin happens to be at the high level.

Here is an example:

Example

```
            XTAL    DAT     0000B       ; Symbol definition of clock stop condition
CEJDG:
        ; <1>
            SKF1    CE                  ; Embedded macro
                                        ; Detects input level of CE pin
            BR      MAIN                ; Branches to main processing
                                        ; if CE = high level
            [ Processing A ]            ; Processing when CE = low

        ; <2>
            STOP    XTAL                ; Clock stops
        ; <3>
            BR      $-1
MAIN:
            [ Main processing ]

            BR      CEJDG
```

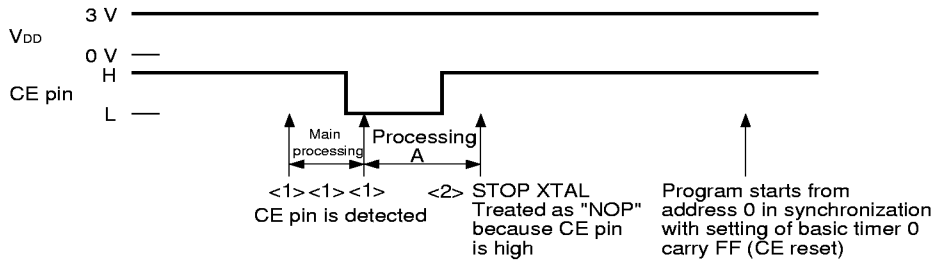
In this program example, the status of the CE pin is detected in <1>. If it is at the low level, the clock stop instruction “STOP XTAL” in <2> is executed after processing A has been performed.

However, if the CE pin goes high while the “STOP XTAL” instruction is executed as shown below, the instruction is treated as a no-operation (“NOP”) instruction.

At this time, assuming that the branch instruction “BR $-1” in <3> is missing, the program execution enters the main processing, and malfunctioning may take place.

Therefore, either insert the branch instruction as shown in <3>, or the program must be designed so that malfunctioning does not take place even after the execution enters the main processing.

If the CE pin is at high level when the “STOP XTAL” instruction is executed, CE reset is effected when the basic timer 0 carry FF is set next time.



19.4 Device Operations in Halt and Clock Stop Statuses

Table 19-1 shows the operations of the CPU and peripheral hardware in the halt and clock stop statuses.

In the halt status, all the peripheral hardware continue the normal operation, except that instruction execution is stopped.

In the clock stop status, all peripheral hardware stop.

The peripheral control register that controls the operating status of the peripheral hardware operates normally (not initialized) in the halt status, but is initialized to a specified value in the clock stop status (when the "STOP s" instruction is executed).

Each peripheral hardware continues the operation set in the peripheral control register in the halt status, and its operation status is determined by the initialized value of the peripheral control register in the clock stop status.

For the value to which the peripheral control register is to be initialized, refer to chapter **Table 8-1. Peripheral Hardware Functions of Peripheral Control Register**.

Here is an example:

Example When P1C0/SO0 pin of port 1C is specified as output port pin, and P0B3/SI/SO1 pin and P0B2/$\overline{SCK}$ pins are used for serial interface

```
HLTINT    DAT       1000B
XTAL      DAT       0000B
INITFLG   P0BBIO3,P0BBIO2
;<1>
SET3      P0B3, P0B2
;<2>
BANK1
CLR1      IRQSIO
INITFLG   SIOCK1, SIOCK0
INITFLG   SIOSEL, NOT SIOHIZ
SET1      IPSIO
EI
;<3>
SET1      SIOTS
;<4>
HALT      HLTINT
;<5>
STOP      XTAL
```

In the above example, the P0B3, P0B2 pins output high level in <1> , the condition of serial interface is set in <2> , and serial communication is started in <3>.

When the "HALT" instruction is executed in <4>, the halt status is set, but the serial communication continues, and the halt status is released when the interrupt by the serial interface is accepted.

If the "STOP" instruction in <5> is executed instead of the "HALT" instruction in <4>, the contents of all the peripheral control registers set in <1>, <2>, and <3> are initialized. Consequently, serial communication is stopped, and all the pins of the port 0B are set in the general-purpose input port mode.

Table 19-1. Device Operations in Halt and Clock Stop Statuses

Hardware peripheral	Status			
	CE pin = high level		CE pin = low level	
	In halt status	In clock stop status	In halt status	In clock stop status
Program counter	Stops at address before HALT instruction	STOP instruction is invalid ("NOP")	Stops at address before HALT instruction	Initialized to 0000H and stops
System register	Retained		Retained	Initialized[Note]
Peripheral register	Retained		Retained	Retained
Timer	Normal operation		Normal operation	Stops
PLL frequency synthesizer	Normal operation		Disabled	Stops
A/D converter	Normal operation		Normal operation	Stops
BEEP	Normal operation		Normal operation	Stops
Serial interface	Normal operation		Normal operation	Stops
Frequency counter	Normal operation		Normal operation	Stops
LCD controller/driver	Normal operation		Normal operation	Stops
General-purpose I/O port	Normal operation		Normal operation	Input port
General-purpose input port	Normal operation		Normal operation	Input port
General-purpose output port	Normal operation		Normal operation	Retained

Note For the value to which these registers are initialized, refer to **4. DATA MEMORY (RAM), 5. SYSTEM REGISTER (SYSREG)** and **8. PERIPHERAL CONTROL REGISTER**.

19.5 Note on Processing of Each Pin in Halt and Clock Stop Statuses

The halt status is used to reduce the current consumption of the device when, for example, only the watch is to be operated.

The clock stop status is used to reduce the current consumption of the device to retain only the contents of the data memory.

Therefore, the current consumption must be minimized in the halt and clock stop status.

The current consumption may increase depending on the status of each pin and therefore, the points listed in Table 19-2 must be observed.

Table 19-2. Pin Status in Halt and Clock Stop Statuses and Notes (1/2)

Pin function		Pin symbol	Pin status and note on processing	
			Halt status	Clock stop status
General-purpose I/O port	Port 0B	P0B3/SI/SO1 P0B2/SCK P0B1 P0B0	Hold status immediately before halt status. **(1) When specified as output pins** Current consumption increases if any of these pins is externally pulled down while it outputs high level, or externally pulled up when it outputs low level. **(2) When specified as input pins** (Except P1A3/AD1, P1A2/AD0, P1A1, P1A0) Current consumption by noise does not increase if any of these pins is floated. **(3) Port 1A (P1A3/AD1, P1A2/AD0, P1A1, P1A0)** Current consumption increases if these pins externally pulled up when they selected pull-down resistors ON by program. When pull-down resistor OFF is selected, these pins are floated and current consumption by noise increases. **(4) P0D3/FMIFC/AMIFC, P0D2/AMIFC** When P0D3/FMIFC/AMIFC, P0D2/AMIFC pins are used for frequency counter, current consumption increases because internal amplifier operates. Initialize frequency counter by program as necessary because it is not automatically disabled even when CE pin is low.	All pins are specified as general-purpose input port pins. At this time, all input ports except port 1A (P1A3/AD1, P1A2/AD0, P1A1, P1A0) do not increase current consumption by noise, even if they are floated. Port 1A (P1A3/AD1, P1A2/AD0, P1A1, P1A0) is retained the status before clock stop. **(1) When pull-down resistor ON is selected by program:** Current consumption increases if these pins externally pulled up. **(2) When pull-down resistor OFF is selected by program:** These pins are floated and current consumption by noise increases.
	Port 0C	P0C1 P0C0		
	Port 0D	P0D3/FMIFC/AMIFC P0D2/AMIFC		
General-purpose input port	Port 1A	P1A3/AD1 P1A2/AD0 P1A1 P1A0		
General-purpose output port	Port 0A	P0A3 | P0A0		These ports are specified as general-purpose output ports. The output contents are retained as is. Therefore, current consumption increases if these ports are externally pulled down while they output high level, or pulled up while they output low level.
	Port 1B	P1B3 | P1B0		
	Port 1C	P1C0/SO0		
Interrupt		INT	Current consumption increases by external noise if this pin is floated.	
CE reset		CE	Current consumption increases by external noise if this pin is floated.	

Table 19-2. Pin Status in Halt and Clock Stop Statuses and Notes (2/2)

Pin function	Pin symbol	Pin status and note on processing	
		Halt status	Clock stop status
LCD segment	LCD14 | LCD0	When these pins are used as general-purpose output port pins, the same points as those of the general-purpose output port described above must be observed.	All pins are specified as LCD segment signal output pins and output low levels (display off).
PLL frequency synthesizer	VCOL VCOH EO	Current consumption increases when PLL operates. When PLL is disabled, VCOL, VCOH : floated EO : floated When CE pin goes low, PLL is automatically disabled.	PLL is disabled. Each pin is as follows: VCOL, VCOH : floated EO : floated
Crystal oscillator circuit	X_{IN} X_{OUT}	Current consumption changes with waveform oscillated by crystal oscillator circuit. The greater the oscillation amplitude, the lower the current consumption. Oscillation amplitude is varied depending on crystal resonator and load capacitor, and therefore must be evaluated.	X_{IN} pin is internally pulled down and X_{OUT} pin outputs high level

19.6 Device Control Function by CE Pin

The CE pin has the following functions by using the input level and rising edge of a signal input from an external source:

(1) PLL frequency synthesizer
(2) Making clock stop instruction valid or invalid
(3) Resets device

19.6.1 Controlling operation of PLL frequency synthesizer

The PLL frequency synthesizer can operate only when the CE pin is high.
When the CE pin is low, PLL is automatically disabled.
When PLL is disabled, the VCOH and VCOL pins are floated, and the EO pin is also floated.
The PLL frequency synthesizer can also be disabled through program even when the CE pin is high.

19.6.2 Making clock stop instruction valid or invalid

The clock stop instruction ("STOP s") is valid only when the CE pin is low.
The clock stop instruction executed when the CE pin is high is treated as an NOP (no operation) instruction.

19.6.3 Resetting device

The device can be reset by raising the CE pin (CE reset).
The device can also be reset by turning off supply voltage V_{DD} (power-ON reset).
For details, refer to **20. RESET**.

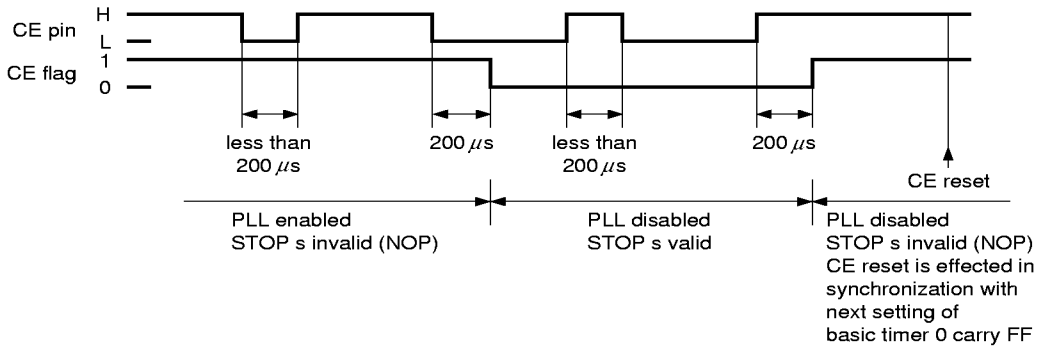
19.6.4 Inputting signal to CE pin

The CE pin does not accept a low- or high-level signal less than 200 μs in order to protect the system from malfunctioning due to noise.

The level of the signal input to the CE pin can be detected by using the CE flag of the CE pin status detection.

Figure 19-5 shows the relations between the input signal and CE flag.

Figure 19-5. Relations between Signal Input to CE Pin and CE Flag



19.6.5 Configuration and functions of CE pin status detection register

The CE pin status detection register detects the level of the signal input to the CE pin.

The configuration and functions of this register are illustrated below.

Figure 19-6. Configuration of CE Pin Status Detection Register

Name	Flag symbol b_3	b_2	b_1	b_0	Address	Read/ Write
CE pin status detection register	0	0	0	CE	(BANK1) 52H	R

	Detects status of CE pin
0	Low level
1	High level

Fixed to "0"

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

Remark –: Determined by status of pin

The CE flag does not change even if a low or high level signal less than 200 μs is input.

19.7 Low-Speed Mode Function

The μPD17073 can slow down the CPU clock when "1" is written to the SYSCK flag of the system clock select register. This function is called a low-speed mode function.

The time required to execute one instruction in the low-speed mode is 106.6 μs. However, the instruction that is executed immediately after the SYSCK flag has been set to "1" takes 103.3 μs.

By slowing down the CPU clock, the current consumption of the device can be lowered as compared with that during normal operation.

Figure 19-7 shows the configuration and function of the system clock select register.

Figure 19-7. Configuration of System Clock Select Register

Name	Flag symbol				Address	Read/ Write
	b_3	b_2	b_1	b_0		
System clock select register	0	0	0	SYSCK	(BANK1) 55H	R/W

SYSCK	Selects system clock (one instruction execution time)
0	53.3 μs
1	106.6 μs

Fixed to "0"

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

Remark R: Retained

19.7.1 Releasing low-speed mode

The low-speed mode is released when the SYSCK flag is reset to "0" by power-ON reset, or when "0" is written to the SYSCK flag.

After the low-speed mode has been released, the CPU clock returns to the normal operation speed (one instruction execution time: 53.3 μs). However, the instruction that is executed immediately after the SYSCK flag has been reset to "0" takes 56.6 μs.

20. RESET

The reset function is to initialize the device operation.

20.1 Configuration of Reset Block

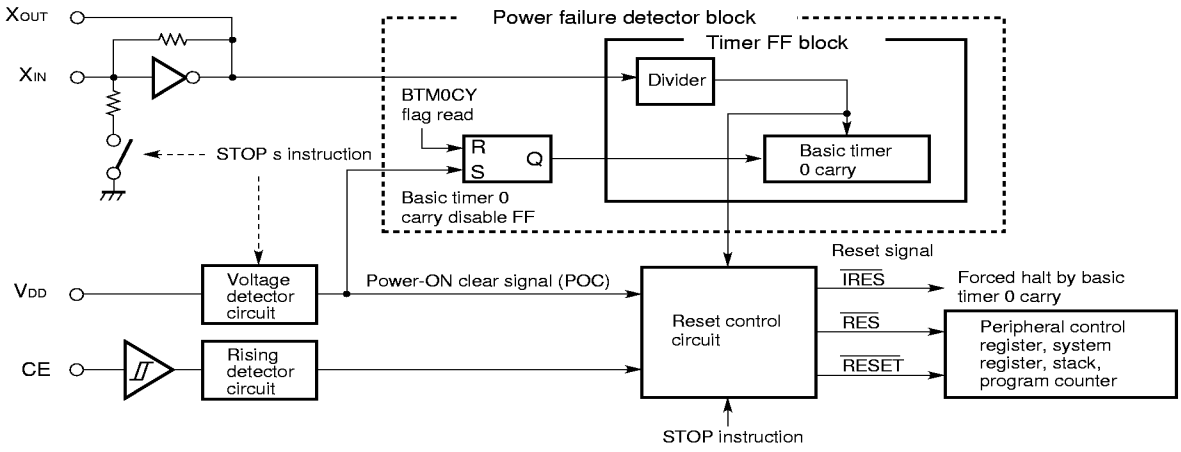
Figure 20-1 shows the configuration of the reset block.

The device can be reset in two ways: by means of power-ON reset (or V_{DD} reset) that is effected by applying supply voltage V_{DD}, and CE reset that is effected by using the CE pin.

The power-ON reset block consists of a voltage detector circuit that detects the voltage input to the V_{DD} pin, a power failure detector circuit, and a reset control circuit.

The CE reset block consists of a circuit that detects the rising of the signal input to the CE pin, and a reset control circuit.

Figure 20-1. Configuration of Reset Block



20.2 Reset Function

The power-ON reset is effected when supply voltage V_{DD} has risen from a specific level, and the CE reset is effected when the CE pin goes high from low.

The power-ON reset is to initialize the program counter, stack, system register, basic timer 0 carry FF and control register, and execute the program from address 0000H.

The CE reset is to initialize part of the program counter, stack, system register, and peripheral control register, and execute the program from address 0000H.

The main differences between the power-ON reset and CE reset are the contents of the peripheral control register to be initialized, and the operation of the power failure detector circuit, which is to be described in 20.6.

The power-ON reset and CE reset are controlled by the reset signals $\overline{IRES}$, $\overline{RES}$, and $\overline{RESET}$ that are output from the reset control circuit shown in Figure 20-1.

Table 20-1 shows the relations among the $\overline{IRES}$, $\overline{RES}$, and $\overline{RESET}$ signals, power-ON reset, and CE reset.

The reset control circuit also operates when the clock stop instruction (STOP s) described in 19. STANDBY has been executed.

The following 20.3 and 20.4 respectively describe the CE reset and power-ON reset.

20.5 describes the relations between the CE reset and power-ON reset.

Table 20-1. Relations 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	
$\overline{IRES}$	×	○	○	Forcibly sets device in halt status, which is released by setting basic timer 0 carry FF.
$\overline{RES}$	×	○	○	Initializes part of peripheral control register
$\overline{RESET}$	○	○	○	Initializes part of program counter, stack, system register, and peripheral control register.

20.3 CE Reset

The CE reset is effected by making the CE pin high.

When the CE pin goes high, the $\overline{\text{RESET}}$ signal is output in synchronization with the rising edge of the next basic timer 0 carry FF setting pulse, and the device is reset.

When the CE reset has been effected, part of the program counter, stack, system register, and peripheral control register is initialized to an initial value by the $\overline{\text{RESET}}$ signal, and the program is executed from address 0000H.

For the initial value, refer to the description of each register.The operation of the CE reset differs depending on whether the clock stop mode is used or not.

This is described in 20.3.1 and 20.3.2.

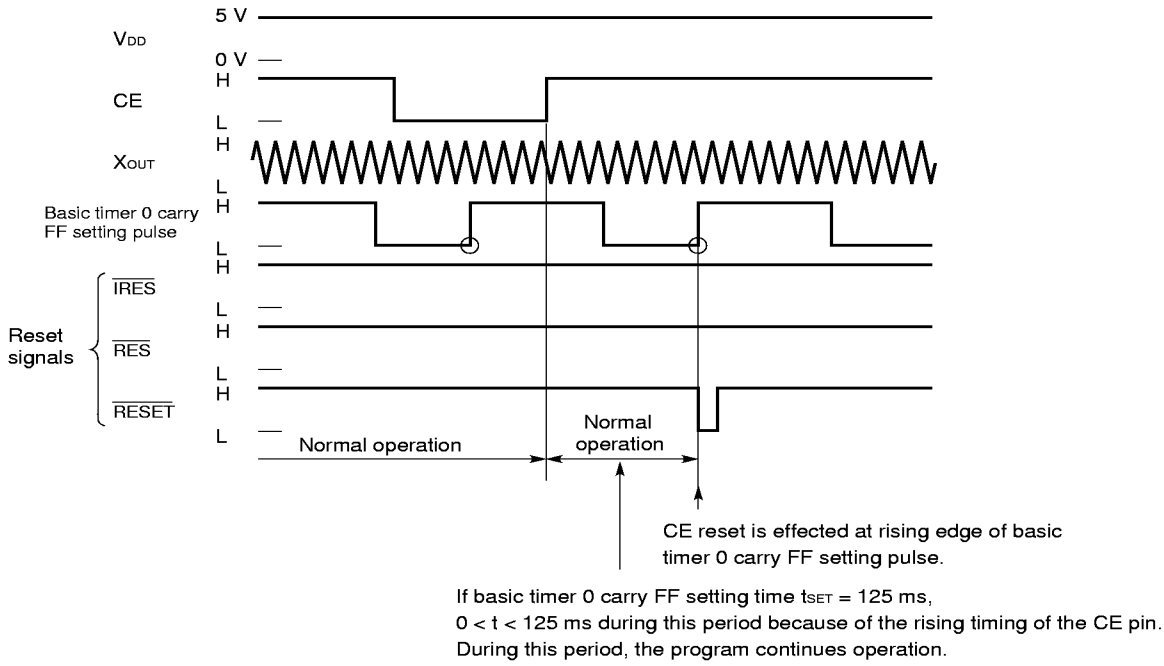
20.3.3 describes the points to be noted when effecting the CE reset.

20.3.1 CE reset when clock stop mode (STOP s instruction) is not used

Figure 20-2 shows the operation.

When the clock stop mode (STOP s instruction) is not used, and after the CE pin has gone high, therefore, the $\overline{\text{RESET}}$ signal is output at the rising edge of the basic timer 0 carry FF setting pulse selected at that time (t_{SET} = 125 ms), and reset is effected.

Figure 20-2. CE Reset Operation When Clock Stop Mode Is Not Used



20.3.2 CE reset when clock stop mode (STOP s instruction) is used

Figure 22-3 shows the operation.

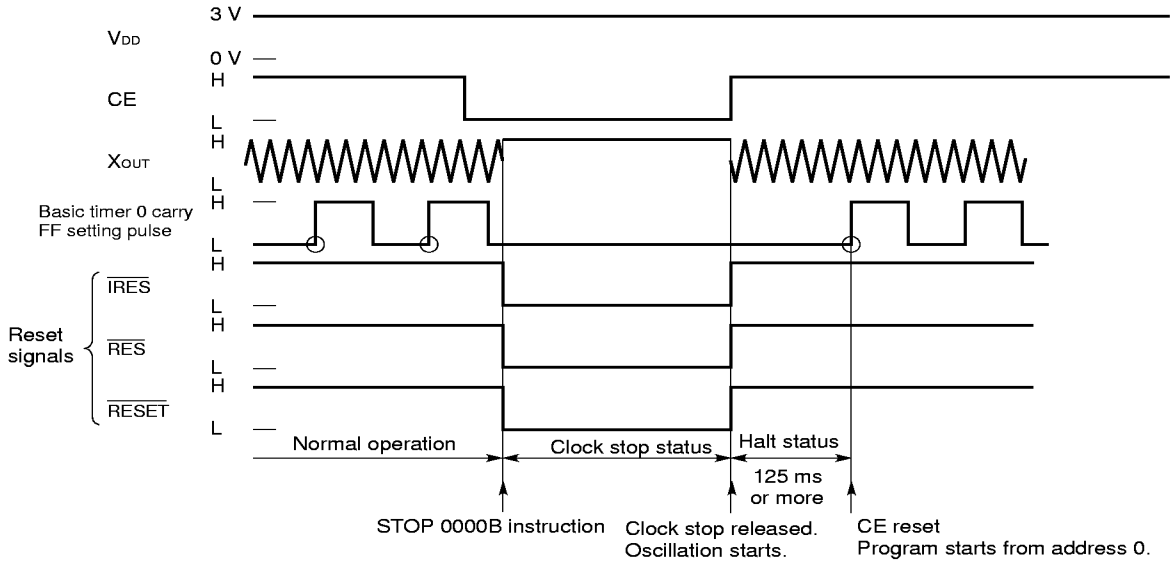
When the clock stop mode is used, the $\overline{\text{IRES}}$, $\overline{\text{RES}}$, and $\overline{\text{RESET}}$ signals are output at the point where the "STOP s" instruction has been executed.

While the CE pin is low, output of the $\overline{\text{IRES}}$ signal continues; therefore, the forced halt status, which is released by the basic timer 0 carry, is set.

However, the device stops operation because the clock is stopped. When the CE pin goes high, the clock stop mode is released and oscillation is started.

At this time, the halt status that is released by the basic timer 0 carry FF is set by the $\overline{\text{IRES}}$ signal. After the CE pin has risen, the oscillation stabilization status lasts (for 125 ms or longer). If the basic timer 0 carry FF setting pulse rises after that, the halt status is released, and program execution is started from address 0.

Figure 20-3. CE Reset Operation When Clock Stop Mode Is Used



20.3.3 Notes on CE reset

Because CE reset is effected regardless of the instruction under execution, the following points (1) and (2) must be noted.

(1) Time for executing timer processing such as watch

To create a watch program by using the basic timer 0 or basic timer 1, the processing of the program must be completed within specific time.

For details, refer to **12.2.5 Notes on using basic timer 0** and **12.3.4 Notes on using basic timer 1**.

(2) Processing of data or flag used in program

Exercise care in rewriting data or flags that cannot be processed with one instruction and whose contents must not be changed even if CE reset is effected, such as security code.

Here is an example:

Example 1.

```
R1      MEM     0.01H                   ; 1st digit of input data of security code
R2      MEM     0.02H                   ; 2nd digit of input data of  security code
R3      MEM     0.03H                   ; Data of 1st digit when security code is changed
R4      MEM     0.04H                   ; Data of 2nd digit when security code is changed
M1      MEM     0.11H                   ; 1st digit of current security code
M2      MEM     0.12H                   ; 2nd digit of current security code

START:
        Key input processing            ; Waits for key input of
          R1 ← Key A contents           ; security code
          R2 ← Key B contents           ; Substitutes contents of pressed code into R1 and R2
        SET2    CMP, Z          ; <1>   ; Compares security code with  input data
        SUB     R1, M1
        SUB     R2, M2
        SKT1    Z
        BR      ERROR                   ; Input data is different from security code
MAIN:
        Key input processing            ; Security code rewriting mode
          R3 ← Key C contents           ; Substitutes contents of
          R4 ← Key D contents           ; pressed key into R3 and R4
        ST      M1, R3          ; <2>   ; Rewrites security code
        ST      M2, R4          ; <3>
        BR      MAIN

ERROR:
          Does not operate
```

Suppose the security code is “12H” in this example. Then the contents of the data memory addresses M1 and M2 are “1H” and “2H”, respectively.

When the CE reset is effected at this time, the contents of the key input in <1> are compared with the security code “12H”, and if they are the same, the ordinary processing is performed.

When the security code is changed by the main processing, the new code is rewritten to M1 and M2 in <2> and <3>.

Suppose the security code is changed to “34H”. Then “3H” and “4H” are written to M1 and M2 in <2> and <3>.

However, if the CE reset happens to occur when <2> has been executed, the program is started from address 0000H without <3> executed.

Consequently, the security code is changed to “32H”, which is not intended, and security cannot be released.

In this case, use the program shown in Example 2.

Example 2.

```
R1        MEM     0.01H                ; 1st digit of input data of security code
R2        MEM     0.02H                ; 2nd digit of input data of  security code
R3        MEM     0.03H                ; Data of 1st digit when security code is changed
R4        MEM     0.04H                ; Data of 2nd digit when security code is changed
M1        MEM     0.11H                ; 1st digit of current security code
M2        MEM     0.12H                ; 2nd digit of current  security code
CHANGE    FLG     0.13H.0              ; "1" while security code is changed

START:
          Key input processing         ; Waits for key input of
            R1 ← Key A contents        ; security code
            R2 ← Key B contents        ; Substitutes contents of pressed code into R1 and R2
          SKT1    CHANGE       ; <4> ; If CHANGE flag is 1,
          BR      SECURITY_CHK
          ST      M1, R3               ; rewrites M1 and M2
          ST      M2, R4
          CLR1    CHANGE
SECURITY_CHK:
          SET2    CMP, Z       ; <1> ; Compares security code with input data
          SUB     R1, M1
          SUB     R2, M2
          SKT1    Z
          BR      ERROR                ; Input data is different from security code
MAIN:
          Key input processing         ; Security code rewriting mode
            R3 ← Key C contents        ; Substitutes contents of
            R4 ← Key D contents        ; pressed key into R3 and R4
          SET1    CHANGE       ; <5> ; Set CHANGE to "1" while security code is rewritten
          ST      M1, R3       ; <2> ; Rewrites security code
          ST      M2, R4       ; <3>
          CLR1    CHANGE               ; Sets CHANGE flag to "0" after security code is rewritten
          BR      MAIN
ERROR:
          Does not operate
```

In this example, the CHANGE flag is set to "1" in <5> before the security code is rewritten in <2> and <3>. Therefore, the security code is written again in <4> even if the CE reset is effected in <3>.

20.4 Power-ON Reset

Power-ON reset is effected by raising the supply voltage V_{DD} of the device from a specific level (called power-ON clear voltage). Power-ON clear voltage is described in 20.4.1.

If the supply voltage V_{DD} is lower than the power-ON clear voltage, a power-ON clear signal (POC) is detected from the voltage detector circuit shown in Figure 20-1.

When the power-ON clear signal is output, the crystal oscillator circuit is stopped, and the device stops operation.

While the power-ON clear signal is output, the $\overline{IRES}$, $\overline{RES}$, and $\overline{RESET}$ signals are output.

When the supply voltage V_{DD} exceeds the power-ON clear voltage, the power-ON clear signal is turned off, the crystal oscillator starts, and the $\overline{IRES}$, $\overline{RES}$, and $\overline{RESET}$ signals are also turned off.

At this time, the halt status that is released by the basic timer 0 carry FF is set by the $\overline{IRES}$ signal. After the power-ON clear signal is deasserted, the oscillation stabilization status lasts (for 125 ms or longer). If the basic timer 0 carry FF setting pulse rises after that, the halt status is released, and power-ON reset is effected.

This operation is illustrated in Figure 20-4.

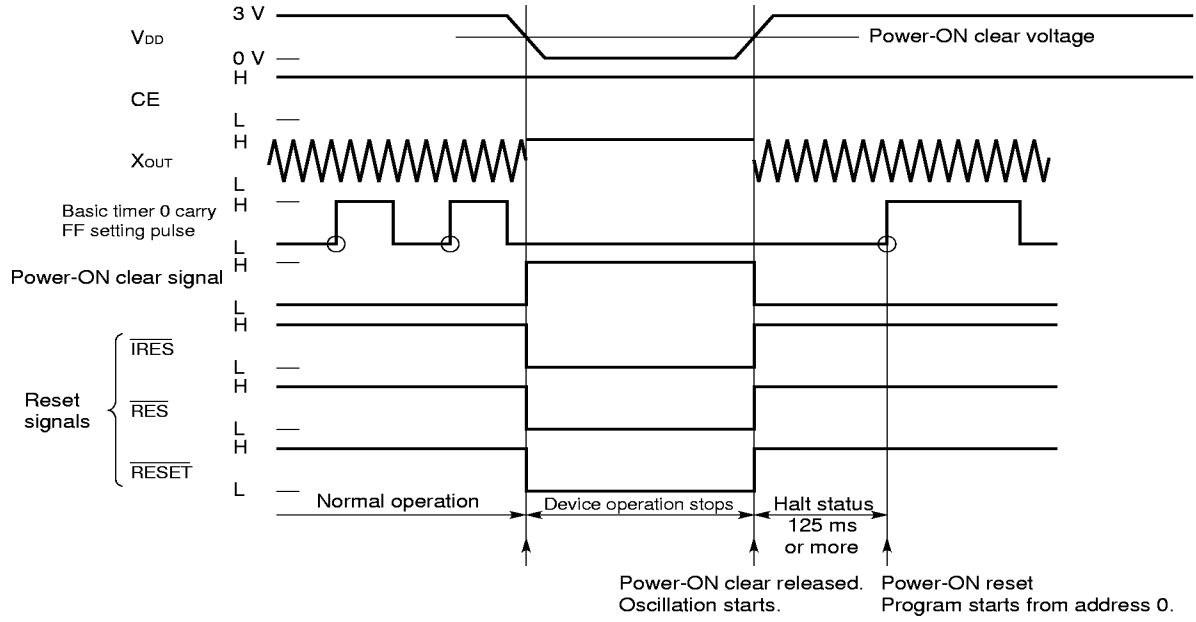
At power-ON reset, the program counter, stack, system register, and peripheral control register are initialized as soon as the power-ON clear signal has been output.

For the power-ON reset while the CPU is operating, refer to **20.4.2**.

For the power-ON reset in the clock stop status, refer to **20.4.3**.

For the power-ON reset when supply voltage V_{DD} rises from 0 V, refer to **20.4.4**.

Figure 20-4. Operation of Power-ON Reset



20.4.1 Power-ON clear voltage

The power-ON clear voltage differs as follows, depending on the CPU operating temperature range and operating conditions:

T_A = 0 to +70 °C : 1.6 V MAX. (when CPU is operating and PLL frequency synthesizer and A/D converter stop)
T_A = –10 to +70 °C : 1.7 V MAX. (when CPU is operating and PLL frequency synthesizer and A/D converter stop)
T_A = –20 to +70 °C : 1.8 V MAX. (when CPU, PLL frequency synthesizer, and A/D converter are operating)

The above values are the maximum values, and the actual power-ON clear voltage must be in a range that does not exceed these maximum values.

The power-ON clear voltage during the CPU operation is the same as that in the clock stop status.

In the description below, the power-ON clear voltage is assumed to be 1.8 V.

20.4.2 Power-ON reset during normal operation

Figure 20-5 (a) shows the operation.

As shown in this figure, the power-ON clear signal is output regardless of the input level of the CE pin when the supply voltage V_{DD} drops below 1.8 V (T_A = –20 to +70 °C, when CPU, PLL, A/D are operating), and the device operation is stopped.

When the supply voltage V_{DD} rises beyond 1.8 V again, the program starts from address 0000H after a halt status of 125 ms or more.

The CPU operation includes when the clock stop instruction is not used, and power-ON clear voltage is 1.8 V during halt status set by the halt instruction.

20.4.3 Power-ON reset in clock stop mode

Figure 20-5 (b) shows the operation.

As shown in this figure, the power-ON clear signal is output and the device operation is stopped when the supply voltage V_{DD} drops below 1.7 V (T_A = –20 to +70 °C, when CPU, PLL, A/D are operating).

However, because the clock stop mode is set, the operation of the device seems not to be changed.

When the supply voltage V_{DD} rises beyond 1.8 V, the program starts from address 0000H after a halt of 125 ms or more.

20.4.4 Power-ON reset when supply voltage V_{DD} rises from 0 V

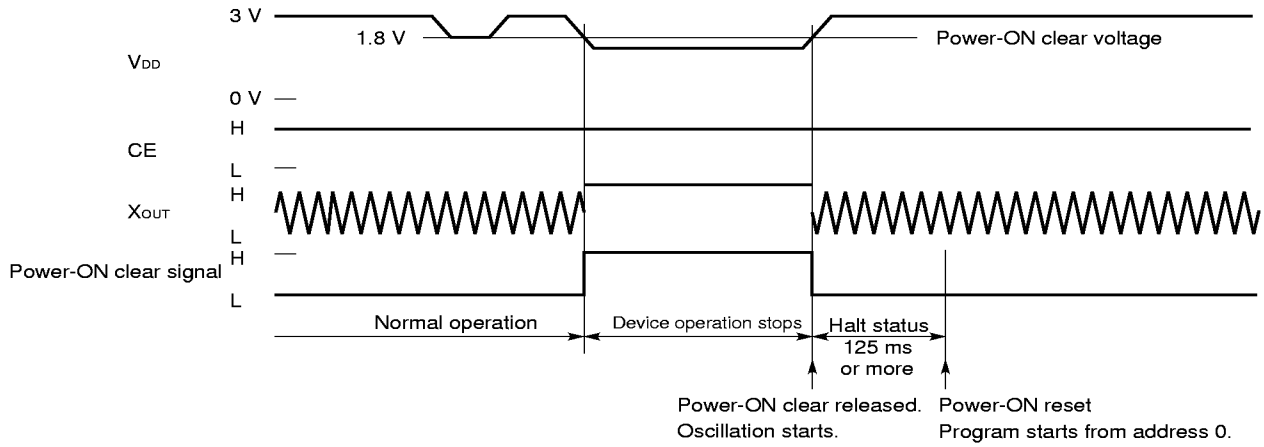
Figure 20-5 (c) shows the operation.

As shown in this figure, the power-ON clear signal is output until the supply voltage V_{DD} rises from 0 V to 1.8 V (T_A = –20 to +70 °C, CPU, PLL, A/D are operating).

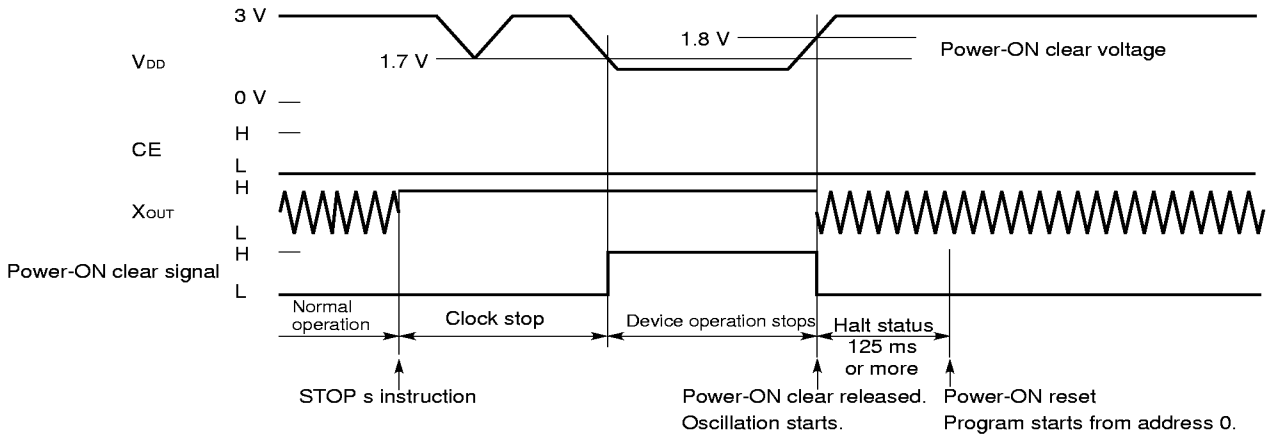
When the supply voltage V_{DD} exceeds the power-ON clear voltage, the crystal oscillator circuit starts operating, and the program starts from address 0000H after a halt of 125 ms or more.

Figure 20-5. Power-ON Reset and Supply Voltage V_{DD}
(T_A = –20 to +70 °C, when CPU, PLL, A/D are operating)

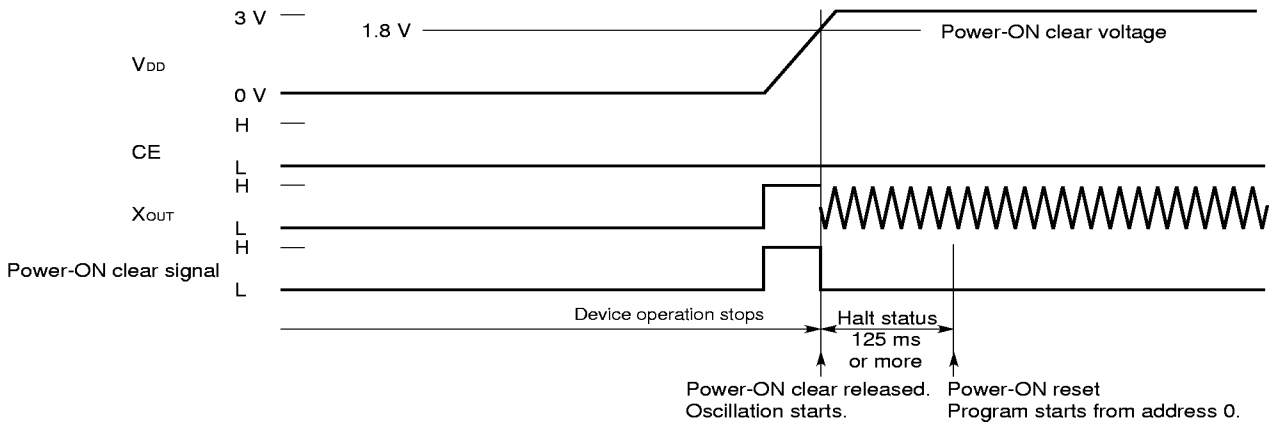
(a) During CPU operation (including halt status)



(b) In clock stop mode



(c) When supply voltage V_{DD} rises from 0 V



20.5 Relations between CE Reset and Power-ON Reset

There is a possibility that power-ON reset and CE reset are effected simultaneously when the supply voltage V_{DD} is applied for the first time.

The reset operations at this time are described in 20.5.1 through 20.5.3.

20.5.1 When V_{DD} pin and CE pin rises simultaneously

Figure 20-6 (a) shows the operation.

At this time, the program starts from address 0000H because of power-ON reset.

20.5.2 When CE pin rises during forced halt status of power-ON reset

Figure 20-6 (b) shows the operation.

At this time, the program starts from address 0000H because of power-ON reset, in the same manner as 20.5.1 above.

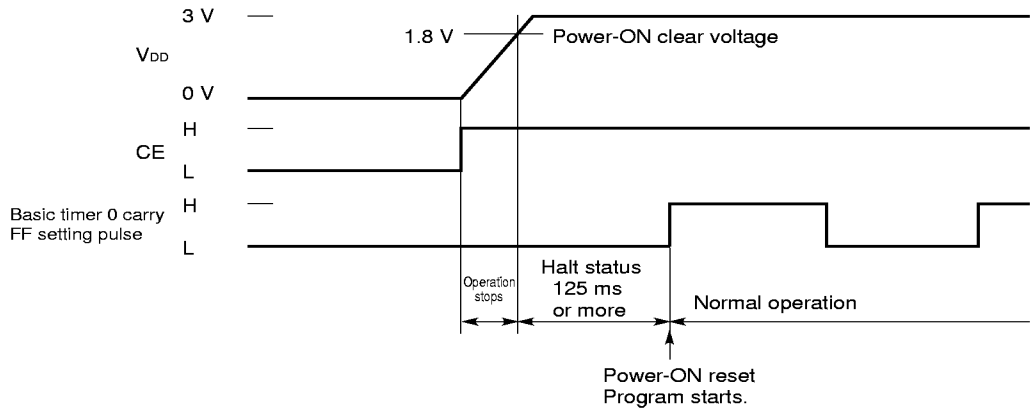
20.5.3 When CE pin rises after power-ON reset

Figure 20-6 (c) shows the operation.

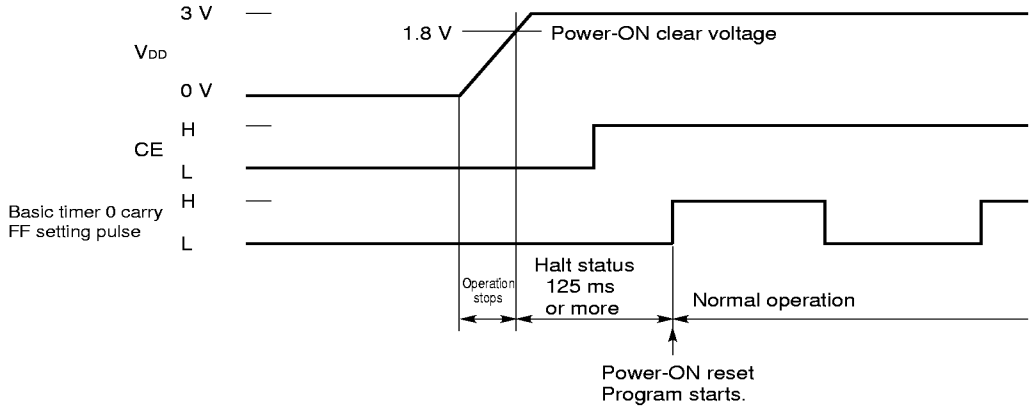
At this time, the program starts from address 0000H because of power-ON reset, and the program starts from address 0000H again at the rising edge of the next basic timer 0 carry FF setting signal because of CE reset.

Figure 20-6. Relations between Power-ON Reset and CE Reset
(T_A = –20 to +70 °C, when CPU, PLL, A/D are operating)

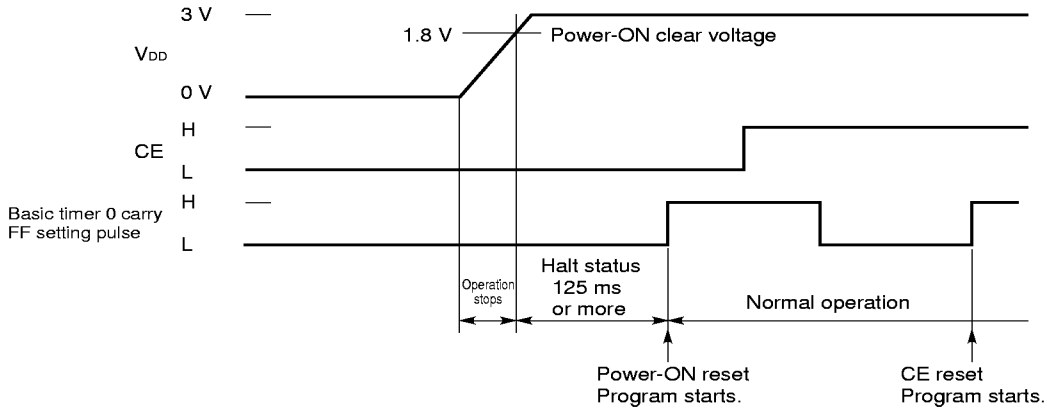
(a) When VDD and CE pins rises simultaneously



(b) When CE pin rises in halt status



(c) When CE pin rises after power-ON reset



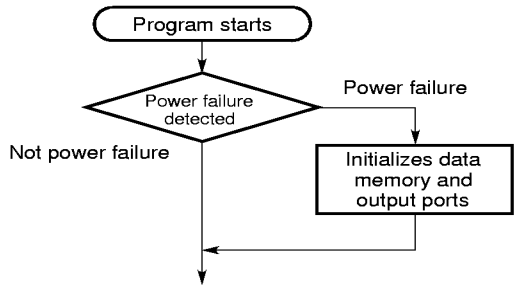
20.6 Power Failure Detection

The power failure detection feature is used to judge, when the device has been reset, whether the reset has been effected by application of supply voltage V_{DD} or by the CE pin.

Because the contents of the data memory and output ports are "undefined" on power application, the contents of these are initialized by detecting a power failure.

The power failure can be detected by detecting the BTM0CY flag by using a power failure detector circuit.

Figure 20-7. Power Failure Detection Flowchart



20.6.1 Power Failure Detector Circuit

The power failure detector circuit consists of a voltage detector circuit as shown in Figure 20-1, basic timer 0 carry disable flip-flop that is reset by the output (power-ON clear signal) of the voltage detector circuit, and basic timer 0 carry.

The basic timer 0 carry disable FF is set to 1 by the power-ON clear signal, and reset to 0 when an instruction that reads the BTM0CY flag has been executed.

While the basic timer 0 carry disable FF is set to 1, the BTM0CY flag is not set to 1.

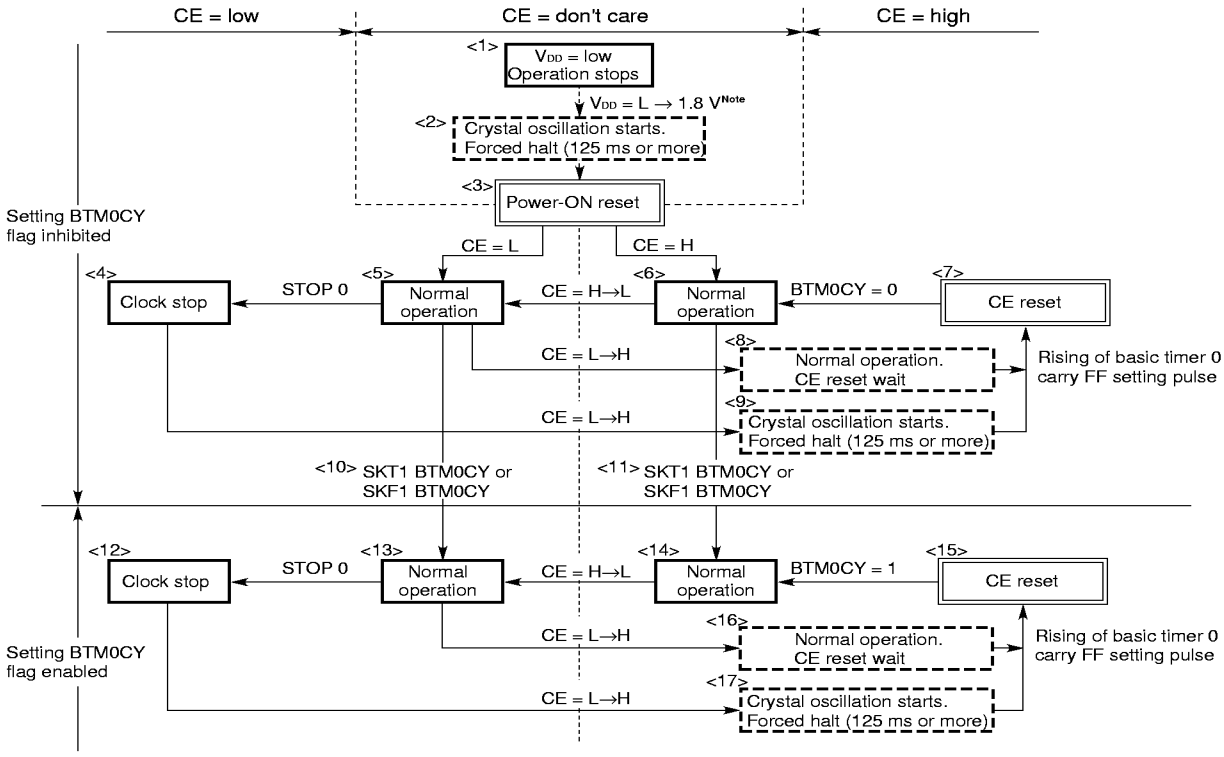
If the power-ON clear signal is output (at power-ON reset), therefore, the program is started with the BTM0CY flag cleared, and setting of the BTM0CY flag is inhibited until an instruction that reads the BTM0CY flag is executed later.

Once the instruction that reads the BTM0CY flag has been executed, the BTM0CY flag is set each time the basic timer 0 carry FF setting pulse rises. Therefore, whether power-ON reset (power failure) or CE reset (not power failure) has been effected can be judged by checking the content of the BTM0CY flag, when the device has been reset. That is, if the BTM0CY flag is cleared to 0, power-ON reset has been effected; if the flag is set to 1, CE reset has been effected.

The voltage at which a power failure can be detected is the same voltage at which power-ON reset is effected.

Figure 20-8 illustrates the status transition of the BTM0CY flag. Figure 20-9 shows the timing chart of Figure 20-8 and the operation of the BTM0CY flag.

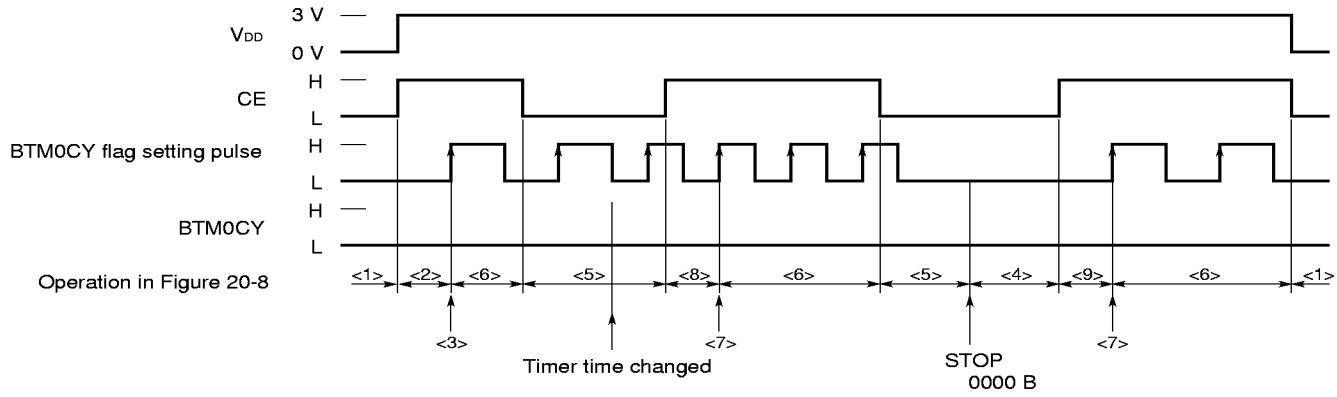
Figure 20-8. Status Transition of BTM0CY Flag



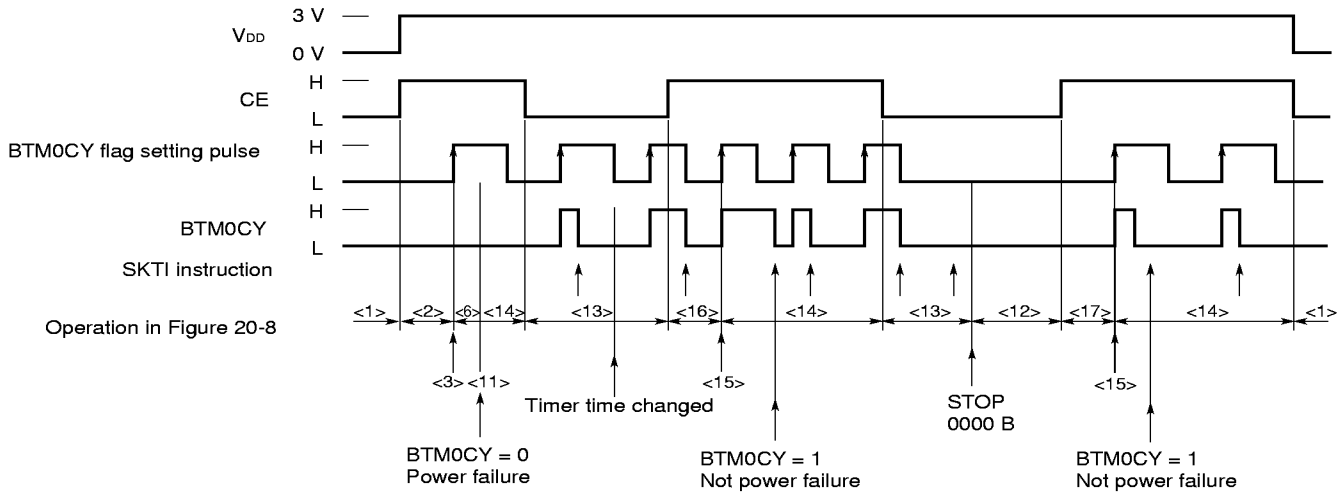
Note 1.8 V is the maximum value and the actual power-ON clear voltage is in a range that does not exceed this maximum value. For details, refer to **20.4.1 Power-ON clear voltage**.

Figure 20-9. Operation of BTM0CY Flag

(a) When BTM0CY flag is never detected (SKT1 BTM0CY or SKF1 BTM0CY is not executed)



(b) To detect power failure with BTM0CY flag



20.6.2 Notes on power failure detection with BTM0CY flag

Keep in mind the following points when using the BTM0CY flag for watch counting:

(1) Updating watch

When creating a watch program by using the basic timer 0, it is necessary to update the watch after a power failure has been detected.

This is because watch counting is skipped once because the BTM0CY flag is read when a power failure has been detected, and thus the BTM0CY flag is cleared to 0.

(2) Watch updating processing time

To update the watch, its processing must be completed before the next basic timer 0 carry FF setting pule rises.

This is because CE reset is effected without the watch updating processing completed if the CE pin goes high while the watch updating processing is in progress.

For further information on (1) and (2) above, refer to **12.2.5 (3) Adjusting basic timer 0 at CE reset**.

To perform power failure processing, the following points must be noted.

(3) Power failure detection timing

Watch counting with the BTM0CY flag must be completed before the next basic timer 0 carry FF setting pulse rises after the BTM0CY flag for power failure detection has been read and the program has been started from address 0000H.

This is because, the basic timer 0 carry FF setting time is 125 ms, and if power failure detection is performed 126 ms after the program has been started, the BTM0CY flag is not detected once.

For details, refer to **12.2.5 (3) Adjusting basic timer 0 at CE reset**.

Moreover, power failure detection and initial processing must be completed within the basic timer 0 carry FF setting time, as shown in the example on the next page.

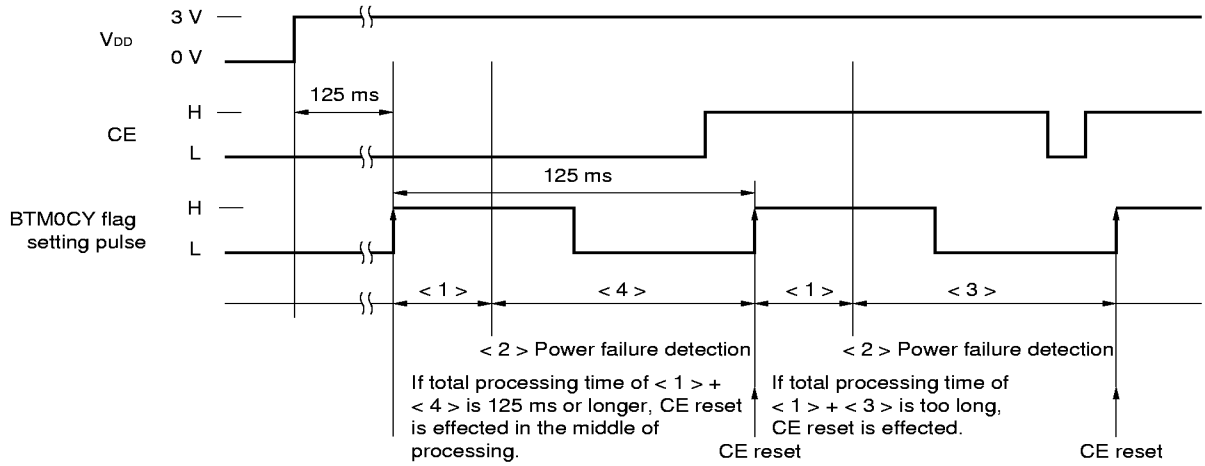
This is because, if he CE pin rises and CE reset is effected during power failure detection or initial processing, the processing is interrupted, resulting in troubles.

To change the basic timer 0 carry FF setting time in the initial processing, one instruction must be used to change the setting time at the end of the initial processing.

Example

```
Program example
START:                                  ; Program address 0000H
;<1>
        Processing at reset
;<2>
        BANK1
        SKT1        BTM0CY              ; Power failure detection
        BR          INITIAL
BACKUP:
;<3>
        Watch updating
        BR          MAIN
INITIAL:
;<4>
        Initial processing
MAIN:
            Main processing
        SKT1        BTM0CY
        BR          MAIN
        Watch updating
        BR          MAIN
```

Operation example



21. μPD17012 INSTRUCTIONS

21.1 Instruction Set Outline

b15 / b14–b11 BIN.				HEX.	0		1	
0	0	0	0	0	ADD	r, m	ADD	m, #n4
0	0	0	1	1	SUB	r, m	SUB	m, #n4
0	0	1	0	2	ADDC	r, m	ADDC	m, #n4
0	0	1	1	3	SUBC	r, m	SUBC	m, #n4
0	1	0	0	4	AND	r, m	AND	m, #n4
0	1	0	1	5	XOR	r, m	XOR	m, #n4
0	1	1	0	6	OR	r, m	OR	m, #n4
0	1	1	1	7	INC MOVT BR CALL RET RETSK EI DI RETI PUSH POP GET PUT RORC STOP HALT NOP	AR DBF, @AR @AR @AR AR AR DBF, p p, DBF 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			SKT	m, #n
1	1	1	1	F			SKF	m, #n

21.2 Legend

AR	: Address register
ASR	: Address stack register indicated by stack pointer
addr	: Program memory address (lower 11 bits)
BANK	: Bank register
CMP	: Compare flag
CY	: Carry flag
DBF	: Data buffer
h	: Halt release condition
INTEF	: Interrupt enable flag
INTR	: Register automatically saved to stack when interrupt occurs
INTSK	: Interrupt stack register
MP	: Data memory row address pointer
MPE	: Memory pointer enable flag
m	: Data memory address indicated by m_R, m_C
m_R	: Data memory row address (higher)
m_C	: Data memory column address (lower)
n	: Bit position (4 bits)
n4	: Immediate data (4 bits)
PAGE	: Page (bits 11 of program counter)
PC	: Program counter
p	: Peripheral address
p_H	: Peripheral address (higher 3 bits)
p_L	: Peripheral address (lower 4 bits)
r	: General register column address
SP	: Stack pointer
s	: Stop release condition
(×)	: Contents address by ×

21.3 Instruction List

Instruction group	Mnemonic	Operand	Operation	Instruction code			
				OP code	Operand		
Addition	ADD	r, m	(r) ← (r) + (m)	00000	m_R	m_C	r
		m, #n4	(m) ← (m) + n4	10000	m_R	m_C	n4
	ADDC	r, m	(r) ← (r) + (m) + CY	00010	m_R	m_C	r
		m, #n4	(m) ← (m) + n4 + CY	10010	m_R	m_C	n4
	INC	AR	AR ← AR + 1	00111	000	1001	0000
Subtraction	SUB	r, m	(r) ← (r) – (m)	00001	m_R	m_C	r
		m, #n4	(m) ← (m) – n4	10001	m_R	m_C	n4
	SUBC	r, m	(r) ← (r) – (m) – CY	00011	m_R	m_C	r
		m, #n4	(m) ← (m) – n4 – CY	10011	m_R	m_C	n4
Logical operation	OR	r, m	(r) ← (r) ∨ (m)	00110	m_R	m_C	r
		m, #n4	(m) ← (m) ∨ n4	10110	m_R	m_C	n4
	AND	r, m	(r) ← (r) ∧ (m)	00100	m_R	m_C	r
		m, #n4	(m) ← (m) ∧ n4	10100	m_R	m_C	n4
	XOR	r, m	(r) ← (r) ∀ (m)	00101	m_R	m_C	r
		m, #n4	(m) ← (m) ∀ n4	10101	m_R	m_C	n4
Judgment	SKT	m, #n	CMP ← 0, if (m) ∧ n = n, then skip	11110	m_R	m_C	n
	SKF	m, #n	CMP ← 0, if (m) ∧ n = 0, then skip	11111	m_R	m_C	n
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
Rotate	RORC	r	CY → $(r)_{b3}$ → $(r)_{b2}$ → $(r)_{b1}$ → $(r)_{b0}$ → CY	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 group	Mnemonic	Operand	Operation	Instruction code			
				OP code	Operand		
Transfer	PUSH	AR	SP ← SP – 1, ASR ← AR	00111	000	1101	0000
	POP	AR	AR ← ASR, SP ← SP + 1	00111	000	1100	0000
	GET	DBF, p	DBF ← (p)	00111	p_H	1011	p_L
	PUT	p, DBF	(p) ← DBF	00111	p_H	1010	p_L
Branch	BR	addr	PC_{10-0} ← addr, PAGE ← 0	01100	addr		
			PC_{10-0} ← addr, PAGE ← 1	01101			
		@AR	PC ← AR	00111	000	0100	0000
Subroutine	CALL	addr	SP ← SP – 1, ASR ← PC, PC_{10-0} ← addr, PAGE ← 0	11100	addr		
		@AR	SP ← SP – 1, ASR ← PC, PC ← AR	00111	000	0101	0000
	RET		PC ← ASR, SP ← SP + 1	00111	000	1110	0000
	RETSK		PC ← ASR, SP ← SP + 1 and skip	00111	001	1110	0000
	RETI		PC ← ASR, INTR ← INTSK, SP ← SP + 1	00111	010	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

21.4 Assembler (AS17K) Embedded Macroinstructions

Legend

flag n : FLG type symbol

< > : Item in < > can be omitted.

	Mnemonic	Operand	Operation	n
Embedded macro	SKTn	flag 1, ...flag n	if (flag 1) to (flag n) = all "1", then skip	1 ≤ n ≤ 4
	SKFn	flag 1, ...flag n	if (flag 1) to (flag n) = all "0", then skip	1 ≤ n ≤ 4
	SETn	flag 1, ...flag n	(flag 1) to (flag n) ← 1	1 ≤ n ≤ 4
	CLRn	flag 1, ...flag n	(flag 1) to (flag n) ← 0	1 ≤ n ≤ 4
	NOTn	flag 1, ...flag n	if (flag n) = "0", then (flag n) ← 1 if (flag n) = "1", then (flag n) ← 0	1 ≤ n ≤ 4
	INITFLG	<NOT> flag 1, ... <<NOT> flag n>	if description = NOT flag n, then (flag n) ← 0 if description = flag n, then (flag n) ← 1	1 ≤ n ≤ 4

22. μPD17073 RESERVED WORDS

22.1 Data Buffer (DBF)

Symbol name	Attribute	Value	R/W	Description
DBF3	MEM	0.0CH	R/W	Bits 15-12 of DBF
DBF2	MEM	0.0DH	R/W	Bits 11-8 of DBF
DBF1	MEM	0.0EH	R/W	Bits 7-4 of DBF
DBF0	MEM	0.0FH	R/W	Bits 3-0 of DBF

22.2 System Register (SYSREG)

Symbol name	Attribute	Value	R/W	Description
AR3	MEM	0.74H	R	Bits 15-12 of address register (fixed to "0")
AR2	MEM	0.75H	R/W	Bits 11-8 of address register
AR1	MEM	0.76H	R/W	Bits 7-4 of address register
AR0	MEM	0.77H	R/W	Bits 3-0 of address register
WR	MEM	0.78H	R	Window register (fixed to "0")
BANK	MEM	0.79H	R/W	Bank register (Only lower 1 bit is valid)
IXH	MEM	0.7AH	R	Index register, high
MPH	MEM	0.7AH	R	Memory pointer, high
MPE	FLG	0.7AH.3	R	Memory pointer enable flag
IXM	MEM	0.7BH	R	Index register, middle (fixed to "0")
MPL	MEM	0.7BH	R	Memory pointer, low
IXL	MEM	0.7CH	R	Index register, low
RPH	MEM	0.7DH	R	General register pointer, high
RPL	MEM	0.7EH	R/W	General register pointer, low (only lower 1 bit is valid)
PSW	MEM	0.7FH	R/W	Program status word
BCD	FLG	0.7EH.0	R/W	BCD operation 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	Index enable flag (fixed to "0")

22.3 LCD Segment Register

Symbol name	Attribute	Value	R/W	Description
LCDD14	MEM	1.41H	R/W	LCD segment register
LCDD13	MEM	1.42H	R/W	
LCDD12	MEM	1.43H	R/W	
LCDD11	MEM	1.44H	R/W	
LCDD10	MEM	1.45H	R/W	
LCDD9	MEM	1.46H	R/W	
LCDD8	MEM	1.47H	R/W	
LCDD7	MEM	1.48H	R/W	
LCDD6	MEM	1.49H	R/W	
LCDD5	MEM	1.4AH	R/W	
LCDD4	MEM	1.4BH	R/W	
LCDD3	MEM	1.4CH	R/W	
LCDD2	MEM	1.4DH	R/W	
LCDD1	MEM	1.4EH	R/W	
LCDD0	MEM	1.4FH	R/W	

22.4 Port Register

Symbol name	Attribute	Value	R/W	Description
P0A3	FLG	0.70.3	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
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/W	Bit 3 of port 0D
P0D2	FLG	0.73H.2	R/W	Bit 2 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
P1C0	FLG	1.72H.0	R/W	Bit 0 of port 1C

22.5 Peripheral Control Register

Symbol name	Attribute	Value	R/W	Description
ADCON	FLG	1.50H.1	R/W	A/D converter control signal power setting flag
LCDEN	FLG	1.50H.0	R/W	LCD driver display start flag
BTM0CY	FLG	1.51H.0	R&Res	Basic timer 0 carry FF status detection flag
CE	FLG	1.52H.0	R	CE pin status detection flag
P1APLD3	FLG	1.53H.3	R/W	P1A3 pin pull-down resistor select flag
P1APLD2	FLG	1.53H.2	R/W	P1A2 pin pull-down resistor select flag
P1APLD1	FLG	1.53H.1	R/W	P1A1 pin pull-down resistor select flag
P1APLD0	FLG	1.53H.0	R/W	P1A0 pin pull-down resistor select flag
SP	MEM	1.54H	R/W	Stack pointer
SYSCK	FLG	1.55H.0	R/W	System clock select flag
INT	FLG	1.56H.2	R/W	INT pin status detection flag
BTM1CK	FLG	1.56H.1	R/W	Basic timer 1 clock select flag
IEG	FLG	1.56H.0	R/W	INT pin interrupt request detection edge direction select flag
IPSIO	FLG	1.57H.2	R/W	Serial interface interrupt enable flag
IPBTM1	FLG	1.57H.1	R/W	Basic timer 1 interrupt enable flag
IP	FLG	1.57H.0	R/W	INT pin interrupt enable flag
IRQ	FLG	1.58H.0	R/W	INT pin interrupt request detection flag
IRQBTM1	FLG	1.59H.0	R/W	Basic timer 1 interrupt request detection flag
IRQSIO	FLG	1.5AH.0	R/W	Serial interface interrupt request detection flag
BEEP0CK1	FLG	1.5BH.1	R/W	BEEP clock select flag
BEEP0CK0	FLG	1.5BH.0	R/W	
ADCCH3	FLGQ	1.5CH.3	R	A/D converter channel select flag (fixed to “0”)
ADCCH2	FLG	1.5CH.2	R	
ADCCH1	FLG	1.5CH.1	R/W	A/D converter channel select flag
ADCCH0	FLG	1.5CH.0	R/W	
ADCRFSEL3	FLG	1.5DH.3	R/W	A/D converter reference voltage setting flag
ADCRFSEL2	FLG	1.5DH.2	R/W	
ADCRFSEL1	FLG	1.5DH.1	R/W	
ADCRFSEL0	FLG	1.5DH.0	R/W	
ADCSTRT	FLG	1.5EH.0	R/W	A/D converter compare start flag
ADCCMP	FLG	1.5FH.0	R	A/D converter compare result detection flag
SIOSEL	FLG	1.60H.2	R/W	Serial in/serial out pin select flag
SIOHIZ	FLG	1.60H.1	R/W	Serial interface/general-purpose port select flag
SIOTS	FLG	1.60H.0	R/W	Serial interface transmit/receive start flag
SIOCK3	FLG	1.61H.3	R	Serial interface I/O clock select flag (fixed to “0”)
SIOCK2	FLG	1.61H.2	R	
SIOCK0	FLG	1.61H.1	R/W	Serial interface I/O clock select flag
SIOCK0	FLG	1.61H.0	R/W	
IFCMD1	FLG	1.62H.3	R/W	IF counter mode select flag (10, 11: AMIF)

Symbol name	Attribute	Value	R/W	Description
IFCMD0	FLG	1.62H.2	R/W	IF counter mode select flag (00: general-purpose I/O port, 01: FMIF)
INCCK1	FLG	1.62H.1	R/W	IF counter clock select flag
IFCCK0	FLG	1.62H.0	R/W	
IFCG	FLG	1.63H.0	R	IF counter gate status detection flag (1: open, 0: close)
IFCSTRT	FLG	1.64H.1	W	IF counter count start flag
IFCRES	FLG	1.64H.0	W	IF counter reset flag
PLLMD3	FLG	1.65H.3	R	PLL mode select flag (fixed to "0")
PLLMD2	FLG	1.65H.2	R	
PLLMD1	FLG	1.65H.1	R/W	PLL mode select flag
PLLMD0	FLG	1.65H.0	R/W	
PLLRFCK3	FLG	1.66H.3	R	PLL reference frequency select flag (fixed to "0")
PLLRFCK2	FLG	1.66H.2	R/W	PLL reference frequency select flag
PLLRFCK1	FLG	1.66H.1	R/W	
PLLRFCK0	FLG	1.66H.0	R/W	
PLLR17	FLG	1.67H.3	R/W	PLL data flag
PLLR16	FLG	1.67H.2	R/W	
PLLR15	FLG	1.67H.1	R/W	
PLLR14	FLG	1.67H.0	R/W	
PLLR13	FLG	1.68H.3	R/W	
PLLR12	FLG	1.68H.2	R/W	
PLLR11	FLG	1.68H.1	R/W	
PLLR10	FLG	1.68H.0	R/W	
PLLR9	FLG	1.69H.3	R/W	
PLLR8	FLG	1.69H.2	R/W	
PLLR7	FLG	1.69H.1	R/W	
PLLR6	FLG	1.69H.0	R/W	
PLLR5	FLG	1.6AH.3	R/W	
PLLR4	FLG	1.6AH.2	R/W	
PLLR3	FLG	1.6AH.1	R/W	
PLLR2	FLG	1.6AH.0	R/W	
PLLR1	FLG	1.6BH.3	R/W	
PLLPUT	FLG	1.6CH.0	W	PLL data set flag
PLLUL	FLG	1.6DH.0	R&Res	PLL unlock FF flag
P0BBIO3	FLG	1.6EH.3	R/W	P0B3 input/output select flag
P0BBIO2	FLG	1.6EH.2	R/W	P0B2 input/output select flag
P0BBIO1	FLG	1.6EH.1	R/W	P0B1 input/output select flag
P0BBIO0	FLG	1.6EH.0	R/W	P0B0 input/output select flag
P0DBIO3	FLG	1.6FH.3	R/W	P0D3 input/output select flag
P0DBIO2	FLG	1.6FH.2	R/W	P0D2 input/output select flag
P0CBIO1	FLG	1.6FH.1	R/W	P0C1 input/output select flag
P0CBIO0	FLG	1.6FH.0	R/W	P0C0 input/output select flag

22.6 Peripheral Hardware Register

Symbol name	Attribute	Value	R/W	Description
SIOSFR	DAT	03H	R/W	Serial interface presettable shift register
AR	DAT	40H	R/W	Address register of GET/PUT/PUSH/CALL/BR/MOVT instruction
IFC	DAT	43H	R	Intermediate frequency (IF) counter data register

22.7 Others

Symbol name	Attribute	Value	Description
DBF	DAT	0FH	Fixed operand value of PUT, GET, and MOVT instructions

23. ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings (T_A = 25 °C)

Parameter	Symbol	Condition	Rating	Unit
Supply voltage	V_{DD}		–0.3 to +4.0	V
Input voltage	V_I	CE pin	–0.3 to V_{DD} +0.6	V
		Other than CE pin	–0.3 to V_{DD} +0.3	V
Output voltage	V_O		–0.3 to V_{DD} +0.3	V
Output current, high	I_{OH}	1 pin	–3.0	mA
		Total of all pins	–20.0	mA
Output current, low	I_{OL}	1 pin	3.0	mA
		Total of all pins	20.0	mA
Operating ambient temperature	T_A		–20 to +70	°C
Storage temperature	T_{stg}		–55 to +125	°C

Caution If the absolute maximum rating of even one of the above parameters is exceeded even momentarily, the quality of the product may be degraded. In other words, the absolute maximum ratings specify the values exceeding which the product may be physically damaged. Be sure to use the product with these ratings never exceeded.

Recommended Operating Range

Parameter	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Supply voltage	V_{DD1}	With CPU, PLL, and AD operating T_A = –20 to +70 °C		1.8	3.0	3.6	V
	V_{DD2}	With CPU operating and	T_A = –10 to +70 °C	1.7	3.0	3.6	V
		PLL and AD stopped	T_A = 0 to +70 °C	1.6	3.0	3.6	V
Supply voltage rise time	t_{rise}	V_{DD}: 0 → 1.8 V				500	mS

DC Characteristics (T_A = –20 to +70 °C, V_{DD} = 1.8 to 3.6 V)

Parameter	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Supply voltage	V_{DD1}	With CPU, PLL, and AD operating T_A = –20 to +70 °C		1.8	3.0	3.6	V
	V_{DD2}	With CPU operating,	T_A = –10 to +70 °C	1.7	3.0	3.6	V
		and PLL and AD stopped	T_A = 0 to +70 °C	1.6	3.0	3.6	V
Supply current	I_{DD1}	With CPU and PLL operating Sine wave input to VCOH pin (f_{IN} = 230 MHz, V_{IN} = 0.2 V_{P-P}) V_{DD} = 3 V, T_A = 25 °C			6.5	10	mA
	I_{DD2}	With CPU operating and PLL stopped (IF counter stopped) Sine wave input to X_{IN} pin (f_{IN} = 75 kHz, V_{IN} = V_{DD}) V_{DD} = 3 V, T_A = 25 °C			35	45	μA
	I_{DD3}	With CPU and PLL stopped (with HALT instruction used) Sine wave input to X_{IN} pin (f_{IN} = 75 kHz, V_{IN} = V_{DD}) LCD display OFF, V_{DD} = 3 V, T_A = 25 °C			10	18	μA
Data retention voltage	V_{DDR}	On power failure detection		1.7			V
Data retention current	I_{DDR}	When crystal oscillation stopped T_A = 25 °C, V_{DD} = 3.0 V				3	μA
Input voltage, high	V_{IH1}	CE, INT, P0B0-P0B3, P0C0, P0C1, P0D2, P0D3		0.8 V_{DD}			V
	V_{IH2}	P1A0-P1A3		0.5 V_{DD}			V
Input voltage, low	V_{IL1}	CE, INT, P0B0-P0B3, P0C0, P0C1, P0D2, P0D3				0.2 V_{DD}	V
	V_{IL2}	P1A0-P1A3				0.05 V_{DD}	V
Output current, high	I_{OH1}	P0A0-P0A3, P0B0-P0B3, P1B0-P1B3, P0C0, P0C1, P0D2, P0D3, P1C0, BEEP V_{OH} = V_{DD} – 0.5 V		–0.5			mA
	I_{OH2}	EO	V_{OH} = V_{DD} – 0.5 V	–0.2			mA
	I_{OH3}	LCD0-LCD14	V_{OH} = V_{DD} – 0.5 V	–20			μA
Output current, low	I_{OL1}	P0A0-P0A3, P0B0-P0B3, P0C0, P0C1, P0D2, P0D3, P1C0, BEEP V_{OL} = 0.5 V		0.5			mA
	I_{OL2}	EO	V_{OL} = 0.5 V	0.2			mA
	I_{OL3}	P1B0-P1B3	V_{OL} = 0.5 V	5			μA
	I_{OL4}	LCD0-LCD14	V_{OL} = 0.5 V	20			μA
Input current, high	I_{IH1}	With P1A0-P1A3 pulled down V_{IH} = V_{DD} = 1.8 V		3		30	μA
	I_{IH2}	With X_{IN} pulled down V_{IH} = V_{DD} = 1.8 V		40			μA
LCD drive voltage	V_{LCD1}	With LCD0-LCD14 output open C_1 = 0.1 μF, C_2 = 0.01 μF T_A = 25 °C		2.8	3.1	3.3	V
Output off leakage current	I_L	EO				±1	μA

AC Characteristics (T_A = –20 to +70 °C, V_{DD} = 1.8 to 3.6 V)

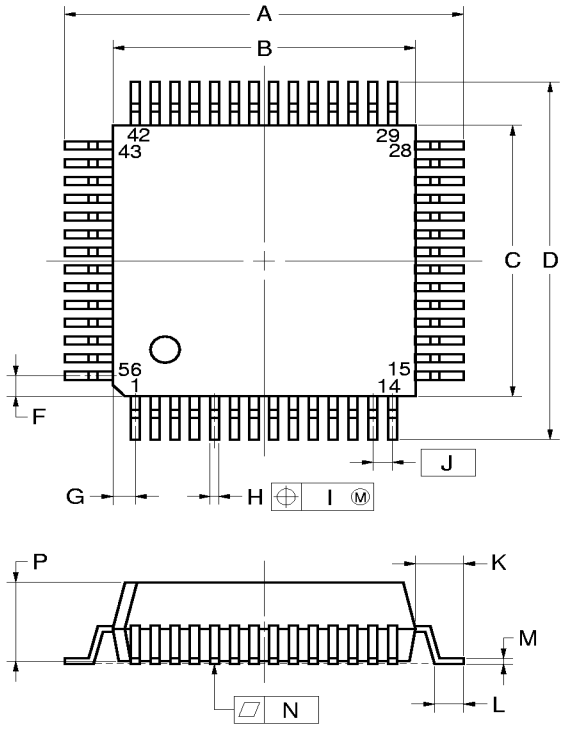
Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{IN1}	VCOL pin, MF mode Sine wave input, V_{IN} = 0.2 V_{P-P}	0.3		8	MHz
	f_{IN2}	VCOL pin, HF mode Sine wave input, V_{IN} = 0.3 V_{P-P}	5		130	MHz
	f_{IN3}	VCOH pin, VHF mode Sine wave input, V_{IN} = 0.2 V_{P-P}	40		230	MHz
	f_{IN4}	AMIFC pin, FMIFC pin, AMIF count mode Sine wave input, V_{IN} = 0.1 V_{P-P}	400		500	kHz
	f_{IN5}	AMIFC pin, FMIFC pin, AMIF count mode Sine wave input, V_{IN} = 0.15 V_{P-P}	0.4		2	MHz
	f_{IN6}	FMIFC pin, FMIF count mode Sine wave input, V_{IN} = 0.1 V_{P-P}	10		11	MHz

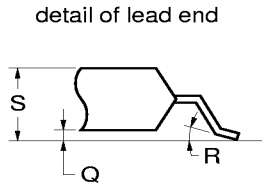
A/D Converter Characteristics (T_A = 25 °C, V_{DD} = 1.8 V)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
A/D converter		4-bit resolution			±1.5	LSB

24. PACKAGE DRAWINGS

56 PIN PLASTIC QFP (10×10)





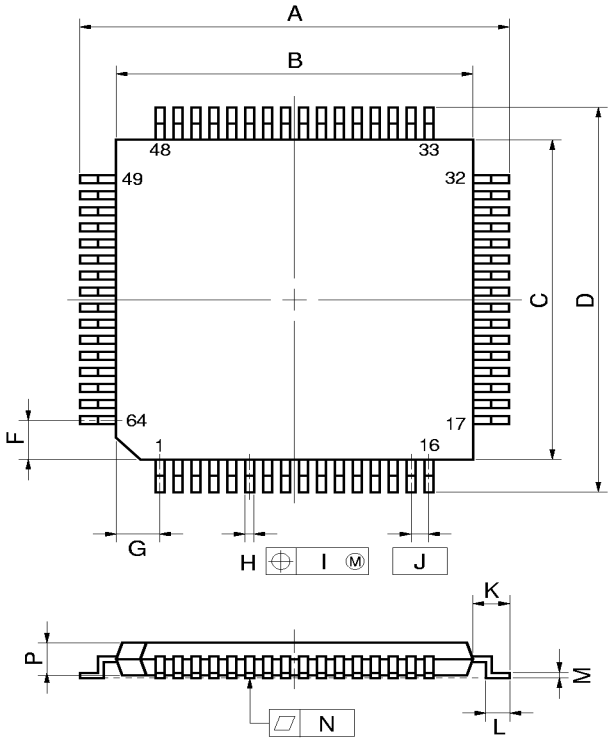
NOTE

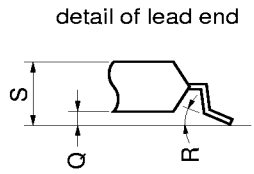
Each lead centerline is located within 0.13 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
A	12.8±0.4	0.504±0.016
B	10.0±0.2	0.394±0.008
C	10.0±0.2	0.394±0.008
D	12.8±0.4	0.504±0.016
F	0.8	0.031
G	0.8	0.031
H	0.30±0.10	0.012±0.004
I	0.13	0.005
J	0.65 (T.P.)	0.026 (T.P.)
K	1.4±0.2	0.055±0.008
L	0.6±0.2	$0.024^{+0.008}_{-0.009}$
M	$0.15^{+0.10}_{-0.05}$	$0.006^{+0.004}_{-0.003}$
N	0.10	0.004
P	1.7	0.067
Q	0.125±0.075	0.005±0.003
R	5°±5°	5°±5°
S	2.0 MAX.	0.079 MAX.

S56GB-65-1A7-3

64 PIN PLASTIC TQFP (FINE PITCH) (□10)





NOTE

Each lead centerline is located within 0.10 mm (0.004 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
A	12.0±0.2	$0.472^{+0.009}_{-0.008}$
B	10.0±0.2	$0.394^{+0.008}_{-0.009}$
C	10.0±0.2	$0.394^{+0.008}_{-0.009}$
D	12.0±0.2	$0.472^{+0.009}_{-0.008}$
F	1.25	0.049
G	1.25	0.049
H	$0.22^{+0.055}_{-0.045}$	0.009±0.002
I	0.10	0.004
J	0.5 (T.P.)	0.020 (T.P.)
K	1.0±0.2	$0.039^{+0.009}_{-0.008}$
L	0.5±0.2	$0.020^{+0.008}_{-0.009}$
M	$0.145^{+0.055}_{-0.045}$	0.006±0.002
N	0.10	0.004
P	1.0±0.1	$0.039^{+0.005}_{-0.004}$
Q	0.1±0.05	0.004±0.002
R	$3°^{+7°}_{-3°}$	$3°^{+7°}_{-3°}$
S	1.27 MAX.	0.050 MAX.

S64GB-50-9EU-1

25. RECOMMENDED SOLDERING CONDITIONS

Solder the μPD17073 under the following recommended conditions.

For the details of the recommended soldering conditions, refer to Information Document **Semiconductor Device Mounting Technology Manual (C10535E)**.

For the soldering methods and conditions other than those recommended, consult NEC.

Table 25-1. Soldering Conditions for Surface-Mount Type

(1) μPD17072GB-×××-1A7: 56-pin plastic QFP (10 × 10 mm, 0.65-mm pitch)
μPD17073GB-×××-1A7: 56-pin plastic QFP (10 × 10 mm, 0.65-mm pitch)

Soldering Method	Soldering Condition	Symbol of Recommended Soldering
Infrared reflow	Package peak temperature: 235 °C, Time: 30 seconds MAX. (210 °C MIN.) Number of times: 2 MAX.	IR35-00-2
VPS	Package peak temperature: 215 °C, Time: 40 seconds MAX. (200 °C MIN.) Number of times: 2 MAX.	VP15-00-2
Wave soldering	Soldering bath temperature: 260 °C MAX., Time: 10 seconds MAX., Number of times: 1, Preheating temperature: 120 °C MAX. (package surface temperature)	WS60-00-1
Pin partial heating	Pin temperature: 300 °C MAX., Time: 3 seconds MAX. (per side of device)	—

(2) μPD17072GB-×××-9EU: 64-pin plastic TQFP (10 × 10 mm, 0.5-mm pitch)
μPD17073GB-×××-9EU: 64-pin plastic TQFP (10 × 10 mm, 0.5-mm pitch)

Soldering Method	Soldering Condition	Symbol of Recommended Soldering
Infrared reflow	Package peak temperature: 235 °C, Time: 30 seconds MAX. (210 °C MIN.) Number of times: 1, Number of days: 2[Note] (after that, prebaking at 125 °C for 10 hours is necessary)	IR35-102-1
VPS	Package peak temperature: 215 °C, Time: 40 seconds MAX. (200 °C MIN.) Number of times: 1, Number of days: 2[Note] (after that, prebaking at 125 °C for 10 hours is necessary)	VP15-102-1
Pin partial heating	Pin temperature: 300 °C MAX., Time: 3 seconds (per side of device)	—

Note The number of days for which the device can be stored after the dry pack is opened, at 25 °C, 65%RH MAX.

Caution Do not use two or more soldering methods in combination (except pin partial heating).

APPENDIX A. NOTES ON CONNECTING CRYSTAL RESONATOR

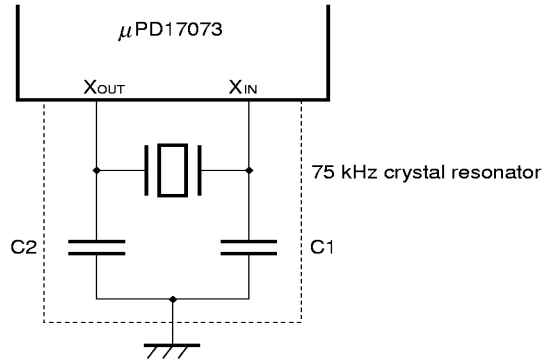
When connecting a crystal resonator to the μPD17073, connect the part enclosed by dotted line in Figure A-1 below as follows to avoid adverse influence of wiring capacitance:

- Keep the wiring length as short as possible.
- Do not cross the wiring with any other signal lines. Do not route the wiring in the vicinity of lines through which a high current flows.
- The ground of the capacitors of the oscillation circuit must be always at the same potential as GND. Do not ground to a ground pattern through which a high current flows.
- Do not extract signals from the oscillation circuit.

To connect the capacitors or to adjust the oscillation frequency, keep in mind the following points (1) through (3):

(1) If the values of C1 and C2 are too high, the oscillation characteristics may be degraded or the current consumption increases.
(2) A trimmer capacitor for oscillation frequency adjustment is generally connected to the X_{IN} pin. Depending on the crystal resonator to be used, however, the oscillation stability may be degraded as a result of connecting a trimmer capacitor to the X_{IN} pin (in this case, connect the trimmer capacitor to the X_{OUT} pin). Therefore, evaluate oscillation by using the crystal resonator to be actually used.
(3) Adjust the oscillation frequency while measuring the LCD drive waveform (62.5 Hz) or VCO oscillation frequency. If a probe is connected to the X_{OUT} or X_{IN} pin, accurate adjustment cannot be made due to the capacitance of the probe.

Figure A-1. Connecting Crystal Resonator



APPENDIX B. DEVELOPMENT TOOLS

The following development tools are available for development of the program of the μPD17073:

Hardware

Name	Outline
In-circuit emulator [IE-17K, IE-17K-ET[Note1], EMU-17K[Note2]]	IE-17K, IE-17K-ET, and EMU-17K are in-circuit emulators common to 17K series. IE-17K and IE-17K-ET are connected to host machine such as PC-9800 series or IBM PC/AT™ with RS-232-C. EMU-17K is mounted in expansion slot of host machine, PC-9800 series. By using these in-circuit emulators in combination with system evaluation board (SE board) dedicated to each model of microcontroller, they operate as emulators dedicated to that microcontroller. When *SIMPLEHOST*®, which is man-machine interface, is used, more sophisticated debugging environment can be created. EMU-17K also has function that allows you to monitor contents of data memory real-time.
SE board (SE-17072)	SE-17072 is SE board for μPD17072 and 17073. It may be used alone to evaluate system, or in combination with in-circuit emulator for debugging.
Emulation probe (EP-17K56GB) (EP-17K56GB-1: Bend lead package EP-17K56GB-2: Inverted lead package)	EP-17K56GB is an emulation probe for the 17K series 56-pin QFP (10 × 10 mm). By using this emulation probe with the EV-9500GB-56[Note 3], the SE board and target system are connected.
Emulation probe (EP-17K64GB: bend lead package)	EP-17K64GB is an emulation probe for the 17K series 64-pin TQFP (10 × 10 mm). By using this emulation probe with the EV-9500GB-64[Note 3], the SE board and target system are connected.
Conversion adapter (EV-9500GB-56)	EV-9500GB-56 is a conversion adapter for a 56-pin QFP (10 × 10 mm). It is used to connect the EP-17K56GB and target system.
Conversion adapter (EV-9500GB-64)	EV-9500GB-64 is a conversion adapter for a 64-pin TQFP (10 × 10 mm). It is used to connect the EP-17K64GB and target system.

Notes 1. Low-cost model: External power supply type

2. This is a product from I.C. For details, contact I.C Corp. ((03) 3447-3793).

3. One EV-9500GB-56 is supplied with the EP-17K56GB. Five EV-9500GB-56 are also available as a set. One EV-9500GB-64 is supplied with the EP-17K64GB. Five EV-9500GB-64 are also available as a set.

Software

Name	Outline	Host Machine	OS		Supply Media	Order Code
17K-Series Assembler (AS17K)	AS17K is an assembler that can be commonly used with the 17K series products. To develop the program of the μPD17072 and 17073, AS17K and a device file (AS17071) are used.	PC-9800 series	MS-DOS™		5"2HD	μS5A10AS17K
					3.5"2HD	μS5A13AS17K
		IBM PC/AT	PC DOS™		5"2HC	μS7B10AS17K
					3.5"2HC	μS7B13AS17K
Device File (AS17071)	AS17071 is a device file for μPD17072 and 17073, and is used with 17K series assembler (AS17K).	PC-9800 series	MS-DOS		5"2HD	μS5A10AS17071
					3.5"2HD	μS5A13AS17071
		IBM PC/AT	PC DOS		5"2HC	μS7B10AS17071
					3.5"2HC	μS7B13AS17071
Support Software (*SIMPLEHOST*)	*SIMPLEHOST* is software that serves as a man-machine interface on Windows™ when a program is developed with an in-circuit emulator and a personal computer.	PC-9800 series	MS-DOW	Windows	5"2HD	μS5A10IE17K
					3.5"2HD	μS5A13IE17K
		IBM PC/AT	PC DOS		5"2HC	μS7B10IE17K
					3.5"2HC	μS7B13IE17K

Remark Supported OS versions are listed below.

OS	Version
MS-DOS	Ver. 3.30 to Ver. 5.00A[Note]
PC DOS	Ver. 3.1 to Ver. 5.0[Note]
Windows	Ver. 3.0 to Ver. 3.1

Note Ver. 5.00/5.00A of MS-DOS and Ver. 5.0 of PC DOS have a task swap function, but it is not supported by this software product.