NEC

MOS INTEGRATED CIRCUIT

μPD17015

4-BIT SINGLE-CHIP MICROCONTROLLER FOR PORTABLE RADIOS

The μPD17015 is a super low-voltage 4-bit single-chip microcontroller for digital tuning applications. It features a prescaler that can operate at frequencies up to 220 MHz, a PLL synthesizer, and an LCD controller/driver, all configured in one chip.

Since, for its CPU, the μPD17015 adopts a 17K architecture which allows data memory to be manipulated directly without accumulators, programming efficiency is enhanced. Each instruction is 16 bits (one word) long.

Since this microcontroller can operate at low voltages (V_{DD} = 1.8 to 3.6 V), it is ideal for controlling portable, battery-powered units such as portable radios, headphone stereo sets, and radio-cassette players.

FEATURES

- 17K architecture : General registers
- Program memory (ROM) : 3K bytes (1528 × 16 bits)
- Data memory (RAM) : 97 × 4 bits
- General-purpose I/O ports : 12
- Minimum required peripheral hardware is built into the chip.
  - LCD controller/driver : 9 segments × 4 commons
  - PLL frequency synthesizer and 220-MHz (MAX.) prescaler
  - Clock generator port
- Low-voltage operation : V_{DD} = 1.8 to 3.6 V (T_A = –10 to +50 °C)

ORDERING INFORMATION

Part number	Package
μPD17015GS-×××-GJG	38-pin plastic shrink SOP (300 mil)

Remark ××× is a ROM code number.

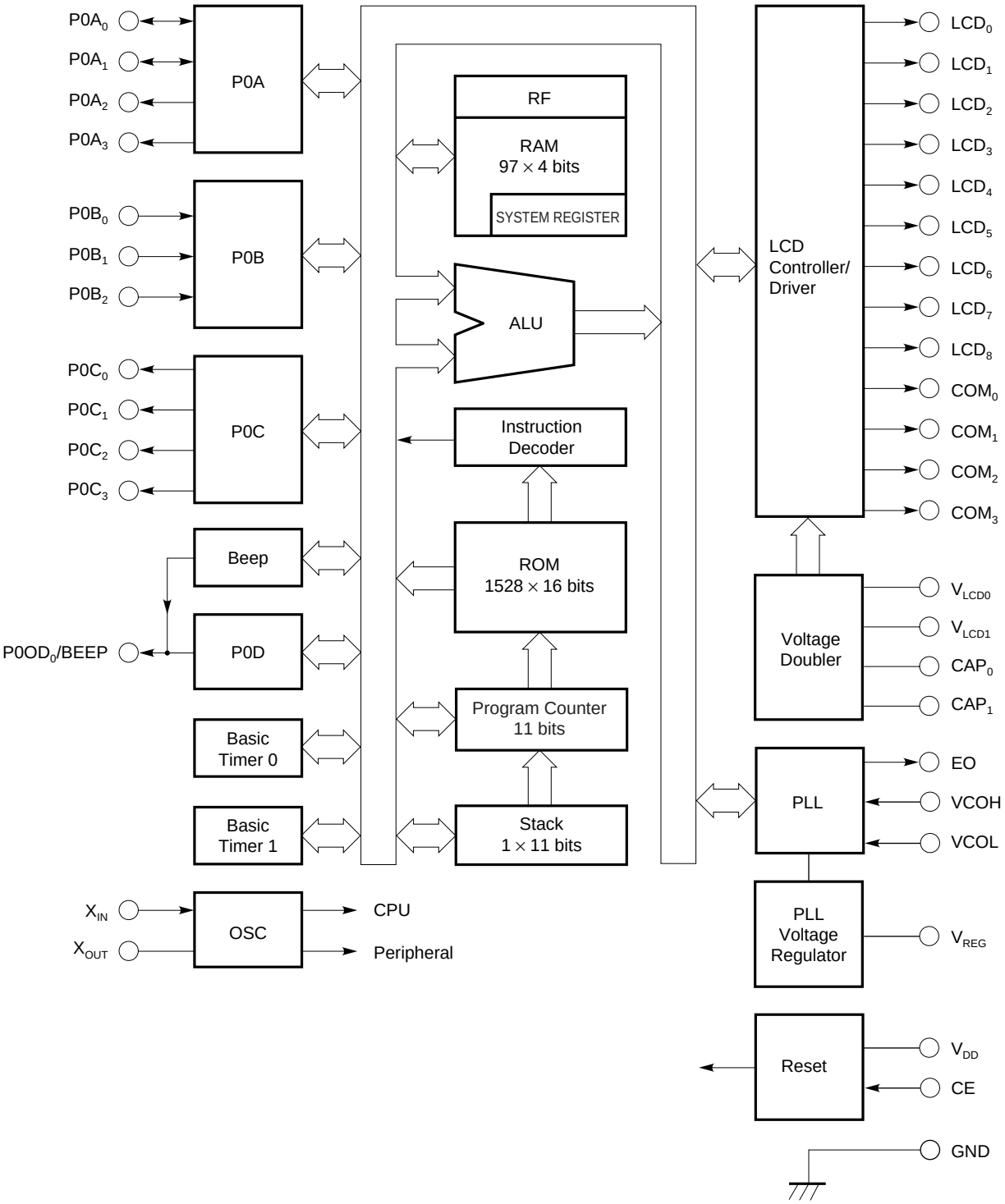
Document No. U10416EJ1V0DS00
(1st edition)
Date Published September 1995 P
Printed in Japan

Major changes in this revision are indicated by stars (★) in the margins.

FUNCTION OVERVIEW

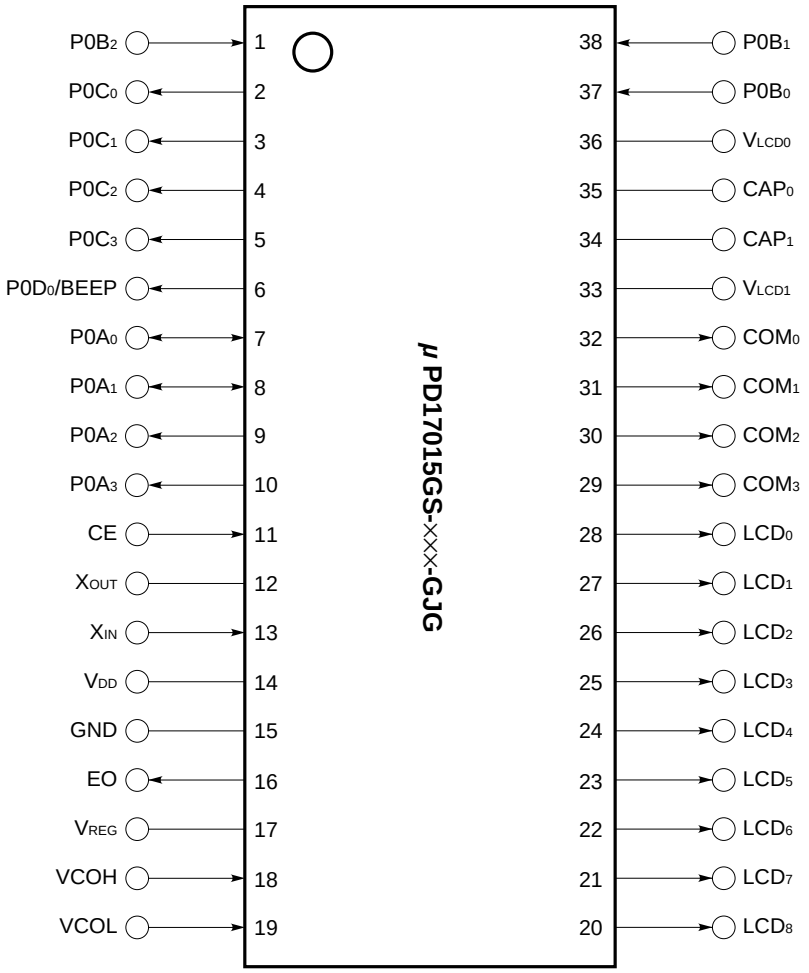
Item		Function
Program memory (ROM)		• 3K bytes (1528 × 16 bits)
General-purpose data memory		• 97 × 4 bits
Instruction execution time		• 53.3 μs (when 75 kHz crystal is used)
★ Stack levels		• 1 level
General-purpose ports		• I/O : 2 ports • Input only : 3 ports } 12 • Output only : 7 ports
BEEP output		1 (3 kHz)
LCD controller/driver		• 9 segments, 4 commons • 1/4 duty, 1/2 bias, frame frequency 62.5 Hz, driving voltage 3.0 V (TYP.)
Timers		2 channels { Basic timer 0 : 125 ms Basic timer 1 : 5 ms
Reset		• Power-on reset • Reset with the CE pin (by switching the CE pin from low to high) • Power-failure detection function
PLL frequency synthesizer	Frequency division method	2 { • Direct division method (VCOL pin (MF mode) : 8 MHz MAX.) • Pulse swallow method (VCOL pin (HF mode) : 55MHz MAX.) (VCOH pin (VHF mode) : 220MHz MAX.)
	Reference frequency	1, 3, 5, or 25 kHz, selected by software.
	Charge pump	Error-out output: 1
	Phase comparator	An unlocked state can be detected by software.
Supply voltage		• 1.8 to 3.6 V (T_A = –10 to +50 °C) • 1.9 to 3.6 V (T_A = –20 to +50 °C)
★ Package		38-pin plastic shrink SOP (0.65-mm pitches, 300 mil)

BLOCK DIAGRAM



PIN CONFIGURATION (TOP VIEW)

38-pin plastic shrink SOP (300 mil)

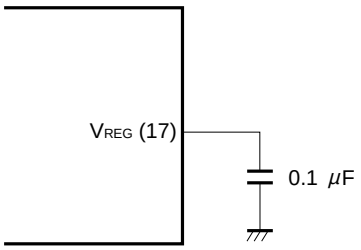


BEEP	:	BEEP output
CAP_0, CAP_1	:	Capacitor connection for LCD drive voltage
CE	:	Chip enable input
COM_0 - COM_3	:	LCD common output
EO	:	Error out
GND	:	Ground
LCD_0 - LCD_8	:	LCD segment output
$P0A_0$, $P0A_1$	:	Port 0A (I/O)
$P0A_2$, $P0A_3$	:	Port 0A (output)
$P0B_0$ - $P0B_2$	:	Port 0B (input)
$P0C_0$ - $P0C_3$	:	Port 0C (output)
$P0D_0$	:	Port 0D (output)
VCOH, VCOL	:	Local oscillation input
V_{DD}	:	Main power supply
V_{LCD0}, V_{LCD1}	:	LCD power supply
V_{REG}	:	Voltage regulator output for PLL
X_{IN}, X_{OUT}	:	Crystal connection

CONTENTS

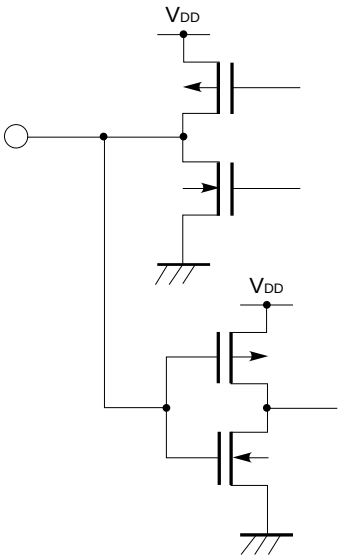
1. PIN FUNCTIONS

1.1 PIN FUNCTIONS

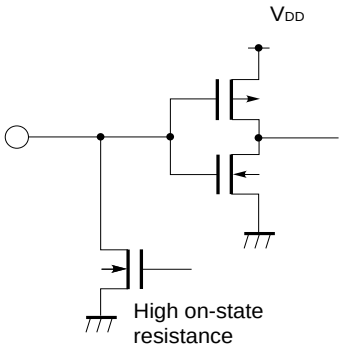
Pin No.	Symbol	Function	Output type	At power-on reset
37 38 1	$P0B_0$ | $P0B_2$	3-bit input port.	Input with a pull-down resistor	Input
2 | 5	$P0C_0$ | $P0C_3$	4-bit CMOS output port	CMOS push-pull	Low-level output
6	$P0D_0$/ BEEP	1-bit general-purpose output port and BEEP signal output • $P0D_0$ • 1-bit output port • BEEP • 3-kHz BEEP output	CMOS push-pull	Low-level output
7 | 10	$P0A_0$ | $P0A_3$	2-bit I/O and output ports • $P0A_0$, $P0A_1$ • 2-bit I/O port • Input or output mode can be specified in units of bits. • $P0A_2$, $P0A_3$ • 2-bit output port	CMOS push-pull	Input ($P0A_0$, $P0A_1$) Low-level output ($P0A_2$, $P0A_3$)
11	CE	Input pin for the μPD17015 operation selection and reset signal	—	Input
12 ★ 13	X_{OUT} X_{IN}	Pins for connecting the crystal (75 kHz) for system clock oscillation. Load capacity: C_I = 12 pF, C_O = 33 pF	CMOS push-pull (X_{OUT})	—
14	V_{DD}	Main power supply pin The pin supplies a voltage of 1.8 to 3.6 V (T_A = –10 to +50 °C) to enable all functions. Note that the voltage applied to each of the other pins must not exceed the voltage applied to the V_{DD} pin.	—	—
15	GND	Ground pin	—	—
16	EO	Output from the charge pump of the PLL frequency synthesizer.	CMOS tristate output	Floating
17	V_{REG}	Output pin for the PLL voltage regulator. Connect this pin to GND through a 0.1-μF capacitor. 	—	—
18 19	VCOH VCOL	Local PLL oscillator frequency input	—	Floating

1.2 EQUIVALENT CIRCUIT OF EACH PIN

(1) P0A (P0A$_0$, P0A$_1$): (I/O)

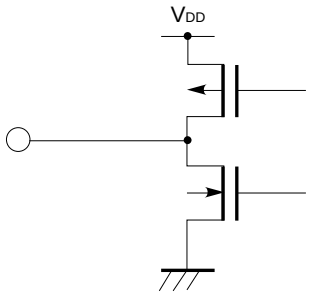


(2) P0B (P0B$_0$, P0B$_1$, P0B$_2$): (Input)

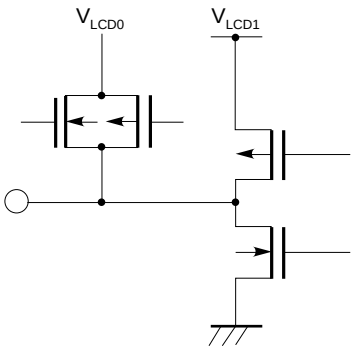


(3) P0A (P0A$_2$, P0A$_3$)
P0C (P0C$_0$, POC$_1$, POC$_2$, POC$_3$)
P0D (P0D$_0$, BEEP)
LCD$_0$ - LCD$_8$
EO

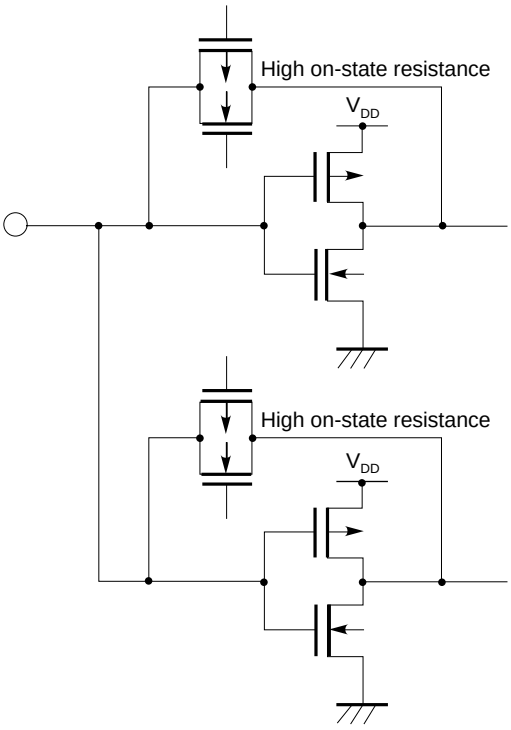
(Output)



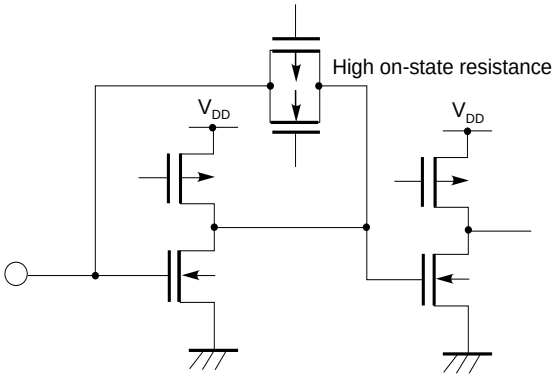
(4) COM_0 - COM_3: (Output)



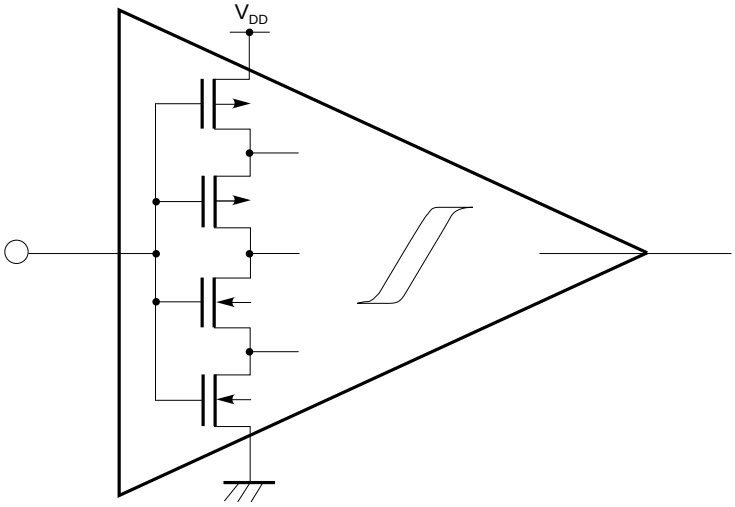
(5) VCOL: (Input)



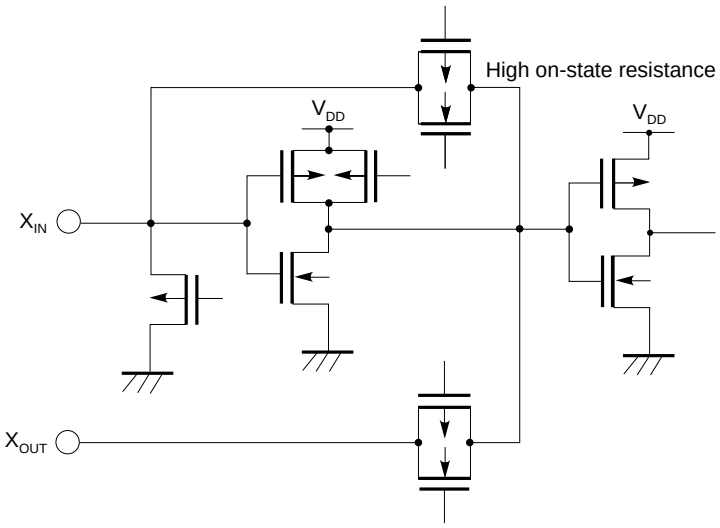
(6) VCOH: (Input)



(7) CE: (Schmitt-triggered input)



(8) X_{OUT}: (Output), X_{IN}: (Input)



★ 1.3 HANDLING UNUSED PINS

When connecting unused pins, the following conditions and handling are recommended:

Table 1-1 Handling Unused Pins

Pin		I/O type	Recommended handling when unused
Port pin	$P0B_0$-$P0B_2$	Input	Connect each pin to ground through a resistor**Note 1**.
	$P0C_0$-$P0C_3$	CMOS push-pull output	Leave open.
	$P0D_0$/BEEP		
	$P0A_2$, $P0A_3$		
	$P0A_0$, $P0A_1$	I/O**Note 2**	Set these pins to input mode by software. Then, connect each pin to V_{DD} or ground through a resistor**Note 1**.
Non-port pin	CAP_0, CAP_1	—	Leave open.
	CE	Input	Connect to V_{DD} through a resistor**Note 1**.
	COM_0-COM_3	Output	Leave open.
	EO	Output	Leave open.
	LCD_0-LCD_8	Output	Leave open.
	VCOH, VCOL	Input	Set these pins to the disabled state by software. Then, leave these pins open.
	V_{LCD0}, V_{LCD1}	—	Leave open.
	V_{REG}	—	Leave open.

Notes 1. When a pin is pulled up (connected to V_{DD} through a resistor) or pulled down (connected to ground through a resistor) outside the chip, caution is required. When using high-resistance pull-up or pull-down resistors, the pin approaches high-impedance and the consumption (through) current increases. Although the optimum pull-up or pull-down resistor varies with the application circuit, in general the use of a resistor of 10 to 100 kilohms is recommended.

2. I/O port is applicable at power-on, at clock stop, at CE reset, or in input mode.

1.4 NOTES ON USE OF THE CE PIN

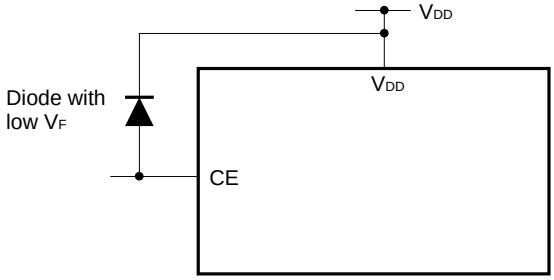
The CE pin has the test mode selecting function for testing the internal operation of the μPD17015 (IC test), besides the functions shown in **Section 1.1**.

Applying a voltage exceeding V_{DD} to the CE pin causes the μPD17015 to enter the test mode. When noise exceeding V_{DD} comes in during normal operation, the device is switched to the test mode.

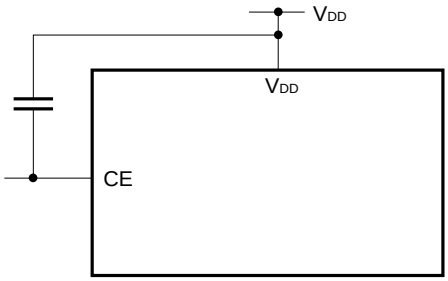
For example, if the wiring from the CE pin is too long, noise may be induced on the wiring, causing this mode switching.

When installing the wiring, lay the wiring in such a way that noise is suppressed as much as possible. If noise yet arises, use an external part to suppress it as shown below.

- Connect a diode with low VF between the pin and V_{DD}.



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



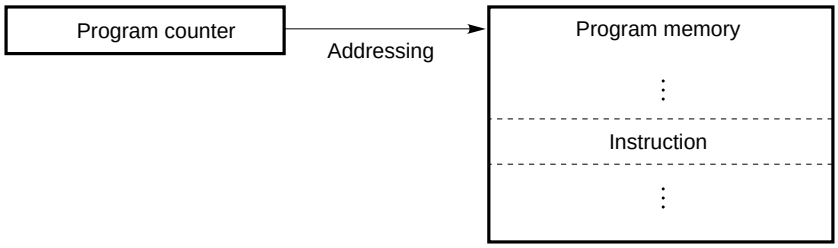
2. PROGRAM MEMORY (ROM)

2.1 OUTLINE OF PROGRAM MEMORY

Fig. 2-1 outlines program memory.

As shown in Fig. 2-1, program memory is addressed with a program counter. Program memory is used to store programs. The program counter is used to specify an address in program memory.

Fig. 2-1 Outline of Program Memory



2.2 PROGRAM MEMORY CONFIGURATION

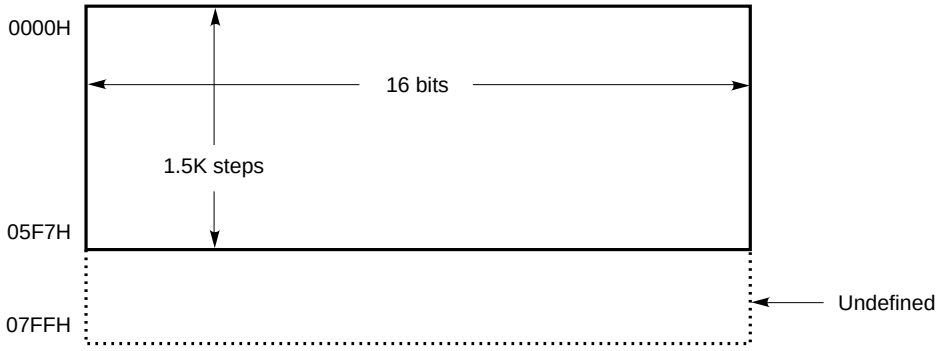
Fig. 2-2 shows the configuration of program memory.

As shown in Fig. 2-2, program memory consists of 1528 steps × 16 bits.

Program memory therefore has addresses 0000H to 05F7H.

All μPD17015 instructions are 16-bit one-word instructions. Each instruction can be stored at a single address of program memory.

Fig. 2-2 Program Memory Configuration



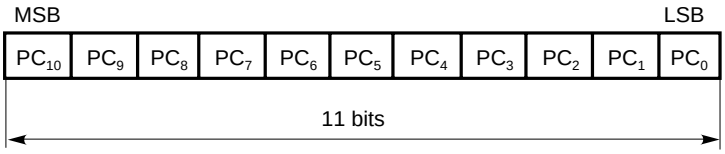
2.3 PROGRAM COUNTER

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

As shown in Fig. 2-3, the program counter consists of a 11-bit binary counter.

The program counter is used to specify an address in program memory.

Fig. 2-3 Program Counter Configuration



2.4 PROGRAM FLOW

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

Fig. 2-4 shows the value set in the program counter when each instruction is executed.

2.4.1 Direct branch (“BR addr”)

A direct branch instruction can branch to all addresses of program memory, 0000H to 05F7H.

2.4.2 Subroutines

To call a subroutine, a direct subroutine call (“CALL addr”) is used.

A direct subroutine call instruction can call a subroutine starting at any address in program memory, 0000H to 05F7H.

RET or RETSK instruction is used as the return instruction from a subroutine.

Fig. 2-4 Program Counter Value When Each Instruction Is Executed

Program counter / Instruction	Value of program counter										
	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
BR addr CALL addr	Instruction operand (addr)										
RET RETSK	Contents of address stack register (return address)										
At power-on reset or CE reset	0	0	0	0	0	0	0	0	0	0	0

2.5 NOTES ON USE OF PROGRAM MEMORY

Although only addresses 0000H to 07FFH can be specified by the program counter, the valid program memory addresses are 0000H to 05F7H. Program memory addresses 05F8H to 07FFH contain undefined values.

Therefore, note the following:

(1) Do not write instructions to addresses 05F8H to 07FFH.
(2) Use a branch instruction to write instructions at address 05F7H.
(3) Do not branch to addresses 05F8H to 07FFH.

3. ADDRESS STACK REGISTER (ASR)

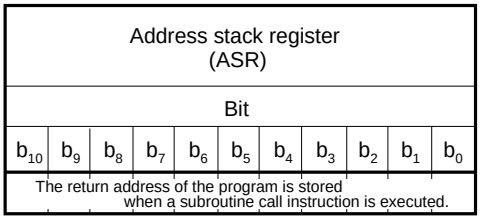
3.1 OUTLINE AND CONFIGURATION OF ADDRESS STACK REGISTER

The address stack register is used to store the return address when a subroutine call instruction is executed.

Fig. 3-1 shows the configuration of the address stack registers.

As shown in Fig. 3-1, there is one address stack register, consisting of 11 bits.

Fig. 3-1 Configuration of Address Stack Registers



3.2 ADDRESS STACK REGISTER OPERATION

When a subroutine call instruction is executed, the return address is stored in the address stack register.

When a return instruction is executed, the contents (return address) of the address stack register are read back into the program counter (see **Table 3-1**).

At power-on reset, the contents of the address stack register are undefined. At CE reset, or when the clock stop instruction is executed, the contents of the address stack register are retained.

Table 3-1 Program Counter Operation Upon Execution of a Subroutine Call Instruction

Instruction	Operation
CALL addr	① Stores the program counter value to the address stack register. ② Transfers the value specified with an operand (addr) to the program counter.
RET RETSK	Reads the value of the address stack register back into the program counter.

3.3 NOTES ON USE OF ADDRESS STACK REGISTER

Note that a subroutine call of more than one level cannot be used because the address stack register has a one-register configuration consisting of 11 bits.

4. DATA MEMORY (RAM)

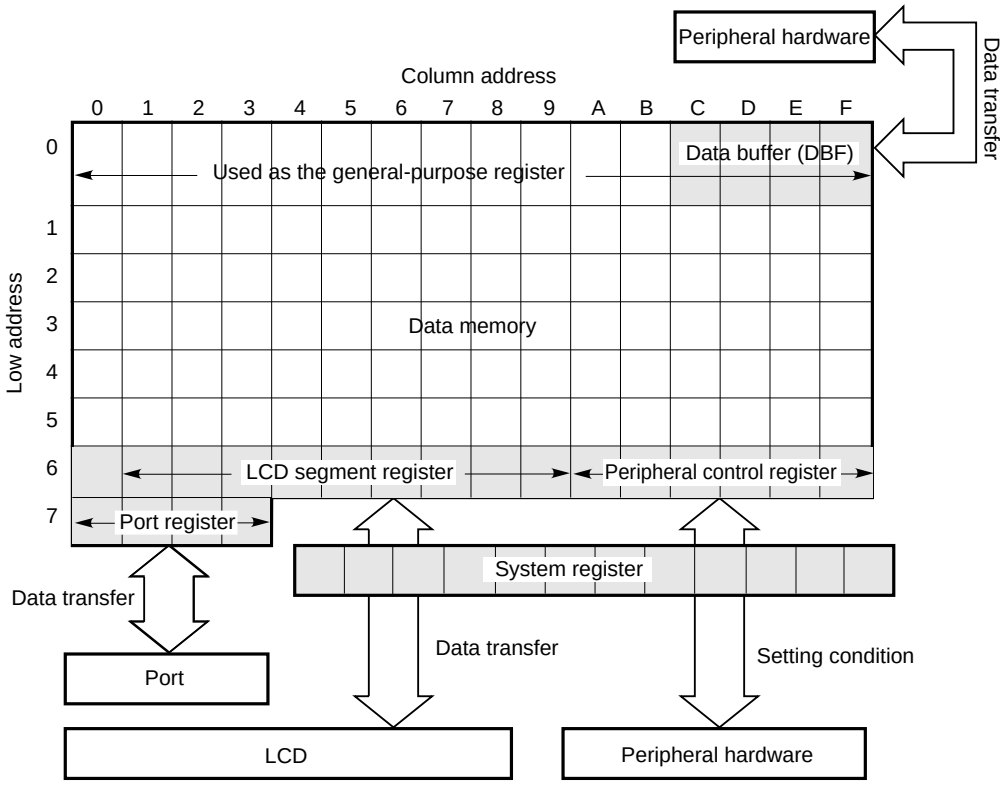
4.1 OUTLINE OF DATA MEMORY

Fig. 4-1 outlines the data memory.

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

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

Fig. 4-1 Outline of Data Memory



4.2 CONFIGURATION AND FUNCTIONS OF DATA MEMORY

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

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

The data memory is divided into the functional blocks described in **Sections 4.2.1** through **4.2.7**.

By using data memory manipulation instructions, 4-bit operations, comparison, decision, and transfer operation can be performed for the data memory. Table 4-1 indicates the data memory manipulation instructions.

4.2.1 System Register (SYSREG)

A system register, allocated to addresses 74H to 7FH, directly controls the CPU. With the μPD17015, only PSWORD (program status word: addresses 7EH and 7FH) can be manipulated.

See **Chapter 5** for details.

4.2.2 Data Buffer (DBF)

A data buffer, allocated at addresses 0CH-0FH, transfers data to and from peripheral hardware.

With the μPD17015, the data buffer transfers data to and from the PLL data register (peripheral address 41H) when the data (16 bits) of the PLL frequency division ratio is set or read.

See **Chapter 8** for details.

4.2.3 General-Purpose Register (GR)

With the μPD17015, the general-purpose register cannot be moved because it is fixed to low address 0 of data memory (i.e., addresses 00H to 0FH).

The general-purpose register can be used to perform an operation on data, or transfer data to and from the data memory, by means of a single instruction.

The general-purpose register, like other data memory, can be controlled by issuing data memory manipulation instructions.

For details, see **Chapter 6**.

4.2.4 LCD Segment Data Register (LCD Segment Register)

The LCD segment register, allocated at addresses 61H to 69H, sets the display data for the LCD controller/driver.

For details, see **Chapter 13**.

4.2.5 Port Data Register (Port Register)

The port register, allocated at addresses 70H to 73H, sets the output data for each general-purpose port and reads the input port data.

For details, see **Chapter 9**.

4.2.6 Peripheral Control Register

The peripheral control register, allocated at addresses 6AH to 6FH, sets the condition of the peripheral hardware (such as PLL or timer).

4.2.7 General-Purpose Data Memory

The general-purpose data memory consists of data memory other than the system register, LCD segment register, port register, and peripheral control register.

For the μPD17015, 97 nibbles (97×4 bits: addresses 00H to 60H) can be used as general-purpose data memory.

Fig. 4-2 Configuration of Data Memory

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0													DBF3	DBF2	DBF1	DBF0
1																
2																
3																
4																
5																
6		LCDD8	LCDD7	LCDD6	LCDD5	LCDD4	LCDD3	LCDD2	LCDD1	LCDD0	LCDE: LCDEN, BEEP0SEL, P0ABIO1, P0ABIO0	PLLMD: PLLMD1, PLLMD0, PLLRFCK1, PLLRFCK0	CEJDG: 0, 0, 0, CE	PLLULJDG: 0, 0, 0, PLLUL	BTM0CYJDG: 0, 0, 0, BTM0CY	BTM1CYJDG: 0, 0, 0, BTM1CY
7	P0A: P0A3, P0A2, P0A1, P0A0	P0B: 0, P0B2, P0B1, P0B0	P0C: P0C3, P0C2, P0C1, P0C0	P0D: 0, 0, 0, P0D0												

System register

AR0	AR1	AR2	AR3	WR	BANK	IXH / MPH	IXM / MPH	IXL	RPH	RPL	PSW
←					0 →					→	BCD, CMP, CY, Z, 0

Table 4-1 List of Data Memory Manipulation Instructions

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

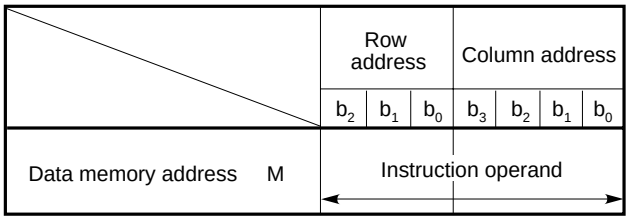
4.3 DATA MEMORY ADDRESSING

Fig. 4-3 shows how a data memory address is specified.

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

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

Fig. 4-3 Data Memory Addressing



4.4 NOTES ON USING DATA MEMORY

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

Initialize the general-purpose data memory as required.

5. SYSTEM REGISTER (SYSREG)

5.1 OUTLINE OF SYSTEM REGISTER

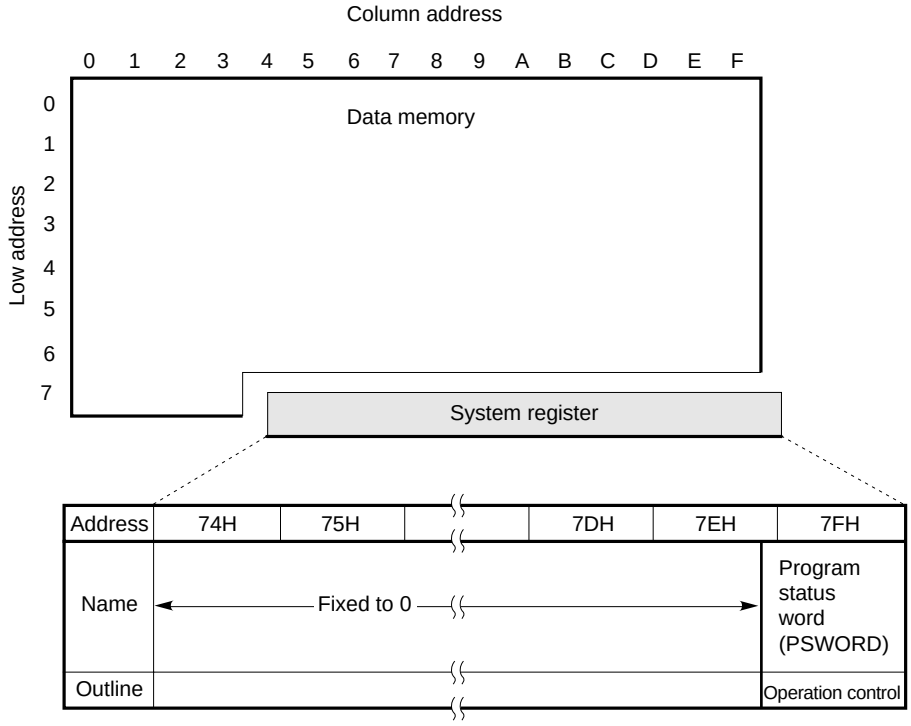
Fig. 5-1 shows where the system registers are located in the data memory, and also outlines the system register.

As shown in Fig. 5-1, a system register is allocated to data memory addresses 74H-7FH.

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

For the μPD17015, only the program status word (PSWORD: 7EH and 7FH) in the system register (addresses 74H to 7FH) can be manipulated.

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



5.2 PROGRAM STATUS WORD (PSWORD)

5.2.1 Format of Program Status Word

Fig. 5-2 shows the format of the program status word.

As shown in Fig. 5-2, the program status word consists of 5 bits: the least significant bit of 7EH (RPL) of the system register, and the 4 bits of address 7FH (PSW) of the system register. Bit 0 of 7FH is always 0.

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

Fig. 5-2 Format of Program Status Word

Address		7EH				7FH			
Register					Program status word (PSWORD)				
Symbol		RPL				PSW			
Bit		b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data		0	0	0	BCD	CMP	CY	Z	0
Upon reset	Power-on	0				0			
	Clock stop	0				0			
	CE	0				0			

5.2.2 Program Status Word Functions

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

Fig. 5-3 outlines the function of each flag of the program status word.

See **Chapter 7** for details.

Fig. 5-3 Outline of Function of Each Flag of Program Status Word

	Program status word (PSWORD)						
RPL				PSW			
b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
0	0	0	BCD	CMP	CY	Z	0

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

5.2.3 Notes on Using Program Status Word

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

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

5.3 NOTES ON USING SYSTEM REGISTER

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

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

6. GENERAL-PURPOSE REGISTER (GR)

6.1 OUTLINE OF GENERAL-PURPOSE REGISTER

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

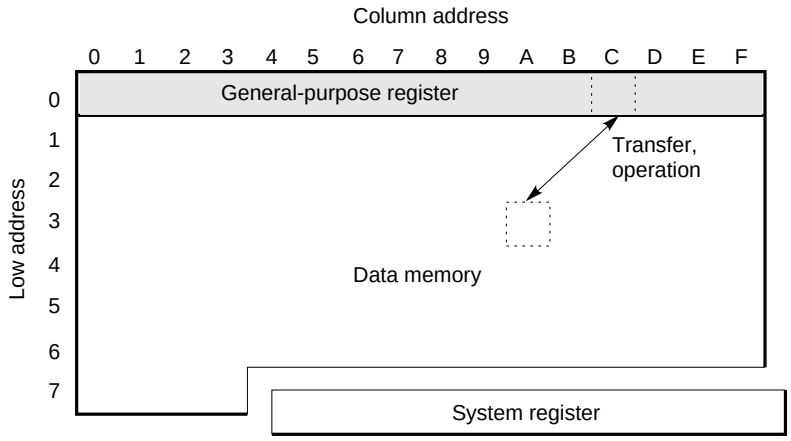
For the μPD17015, the general-purpose register is fixed to low address 0 of the data memory, consisting of 16 nibbles (00H to 0FH, 16 × 4 bits).

16 nibbles of the row address 0 specified as a general-purpose register is used to perform an operation with or transfer data to and from the data memory.

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

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

Fig. 6-1 Outline of General-Purpose Register



6.2 GENERAL-PURPOSE REGISTER ADDRESS GENERATION WITH INSTRUCTIONS

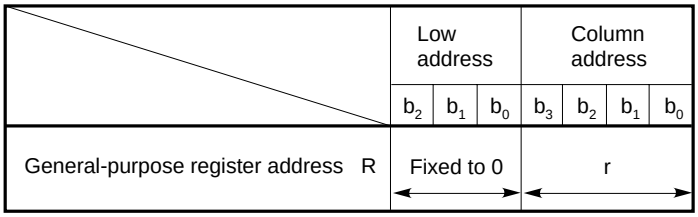
Sections 6.2.1 and **6.2.2** below describe general-purpose register address generation when each instruction is executed.

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

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

Fig. 6-2 indicates a general-purpose register address R specified by operand r of an instruction. Operand r specifies only a column address.

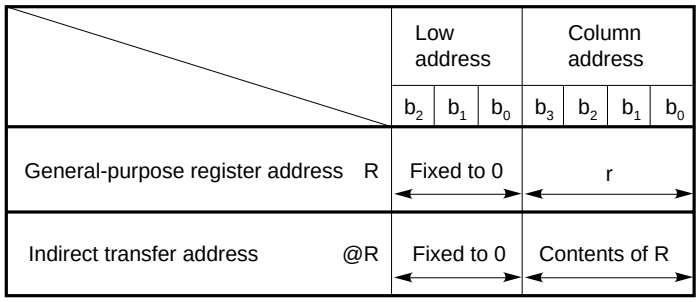
Fig. 6-2 General-Purpose Register Address Generation



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

Fig. 6-3 indicates a general-purpose register address R specified by operand r of an instruction, and an indirect transfer address specified by @R.

Fig. 6-3 General-Purpose Register Address Generation



6.3 NOTES ON USING GENERAL-PURPOSE REGISTER

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

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

7. ARITHMETIC LOGIC UNIT (ALU) BLOCK

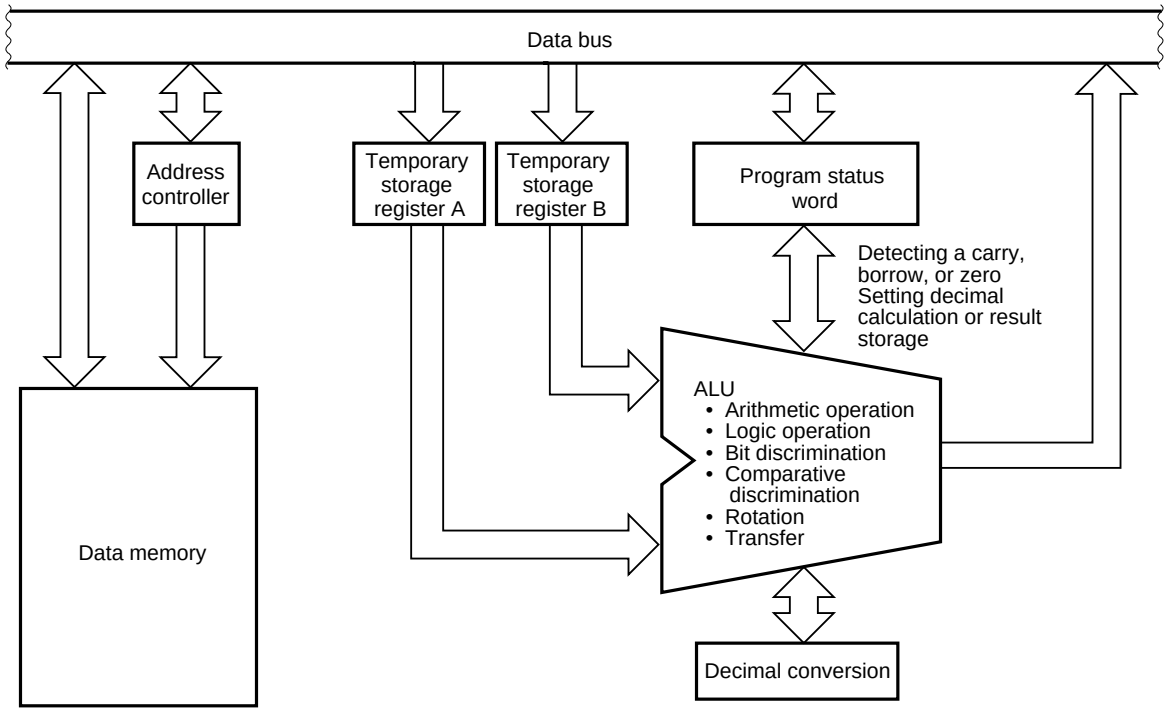
7.1 OVERVIEW

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

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

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

Fig. 7-1 Overview of the ALU Block



7.2 CONFIGURATION AND FUNCTIONS OF THE COMPONENTS OF THE ALU BLOCK

7.2.1 ALU

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

7.2.2 Temporary Storage Registers A and B

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

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

7.2.3 Program Status Word

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

For details of the program status word, see **Section 5.2**.

7.2.4 Decimal Conversion Circuit

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

7.2.5 Address Controller

The address controller specifies an address in data memory.

7.3 ALU OPERATIONS

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

Table 7-2 lists the converted decimal data used in decimal operations.

Table 7-1 ALU Operations

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

Table 7-2 Converted Decimal Data

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

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

Remark Correct decimal conversion is not possible in the shaded area.

7.4 NOTES ON USING THE ALU

7.4.1 Notes on Using the Program Status Word for Operations

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

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

If the CMP flag is set, the result of the arithmetic operation is not stored and the CY and Z flags are set to 1 or cleared to 0 as usual.

7.4.2 Notes on Performing Decimal Operations

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

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

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

8. DATA BUFFER (DBF)

8.1 OVERVIEW

Fig. 8-1 shows an overview of the data buffer.

The data buffer is configured in data memory and used to transfer data to and from peripheral hardware.

Fig. 8-1 Overview of the Data Buffer

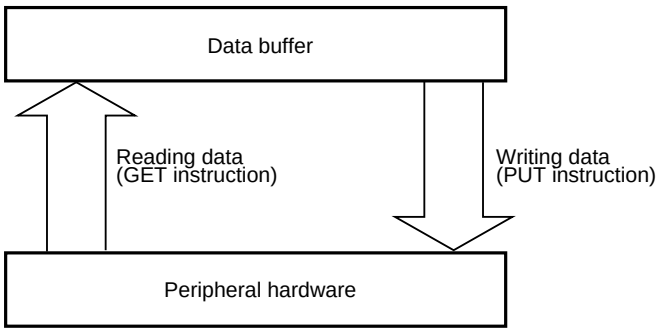


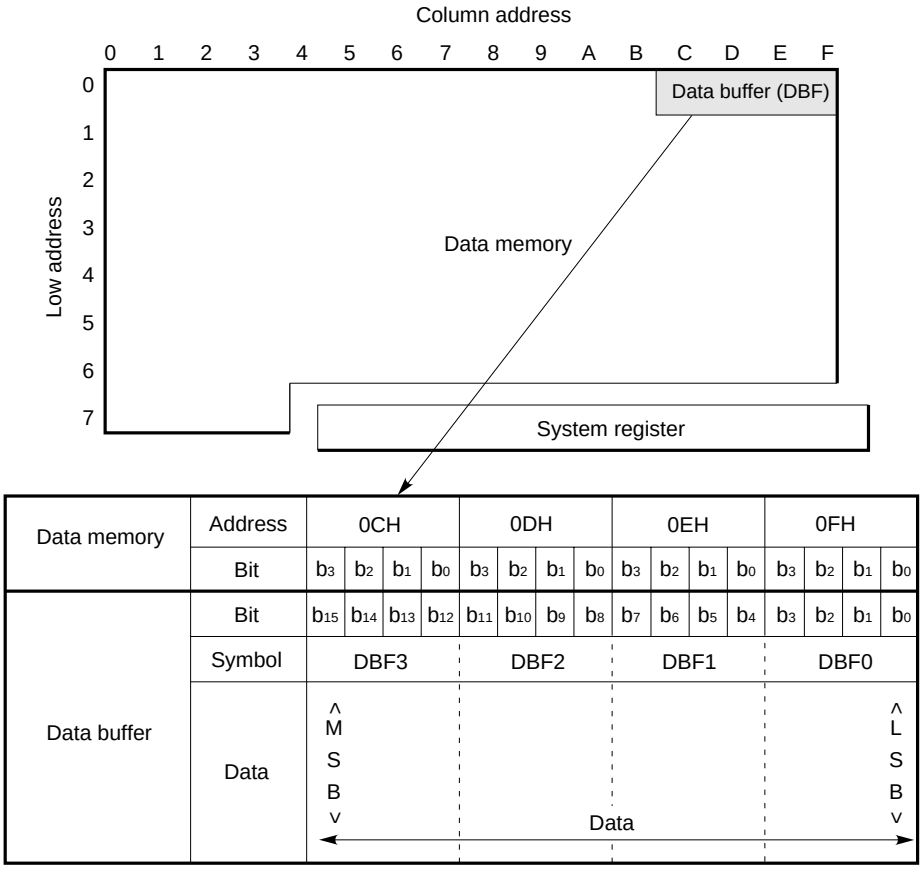
Fig. 8-2 shows the configuration of the data buffer.

As shown in Fig. 8-2, the data buffer consists of 16 bits of addresses 0CH to 0FH in data memory.

The most significant bit (MSB) of the 16-bit data is bit b_3 at address 0CH. The least significant bit (LSB) is bit b_0 at address 0FH.

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

Fig. 8-2 Configuration of the 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	^ MSB v															^ LSB v

Data

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

The PUT and GET instructions operate as described below:

(1) GET DBF, p

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

(2) PUT p, DBF

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

8.2 Relationship between the Peripheral Hardware and Data Buffer

Table 8-1 indicates the relationship between the peripheral hardware and data buffer.

With the μPD17015, the DBF transfers data to and from the PLL data register (peripheral address 41H) when data (16 bits) of the PLL frequency division ratio is set or read.

Table 8-1 Relationship between the Peripheral Hardware and Data Buffer

Peripheral hardware	Peripheral register used to transfer data to or from the data buffer				Function		
	Name	Symbol	Peripheral address	Instruction that can be used	Number of data buffer input bits	Number of bits actually used	Description
PLL frequency synthesizer	PLL data register	PLLR	41H	GET/PUT	16	16	Sets N, by which PLL frequency is divided.

8.3 NOTES ON USING THE DATA BUFFER

If an attempt is made to read the data at an unused peripheral address, an undefined value will be read. If writing to the unused address is attempted, nothing changes.

9. GENERAL-PURPOSE PORTS

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

9.1 OVERVIEW

Table 9-1 indicates the relationship between the ports and port registers.

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

I/O port is a bit I/O port, whose bit (each pin) can be set to input or output mode.

Table 9-1 Relationship between Ports (Pins) and Port Registers

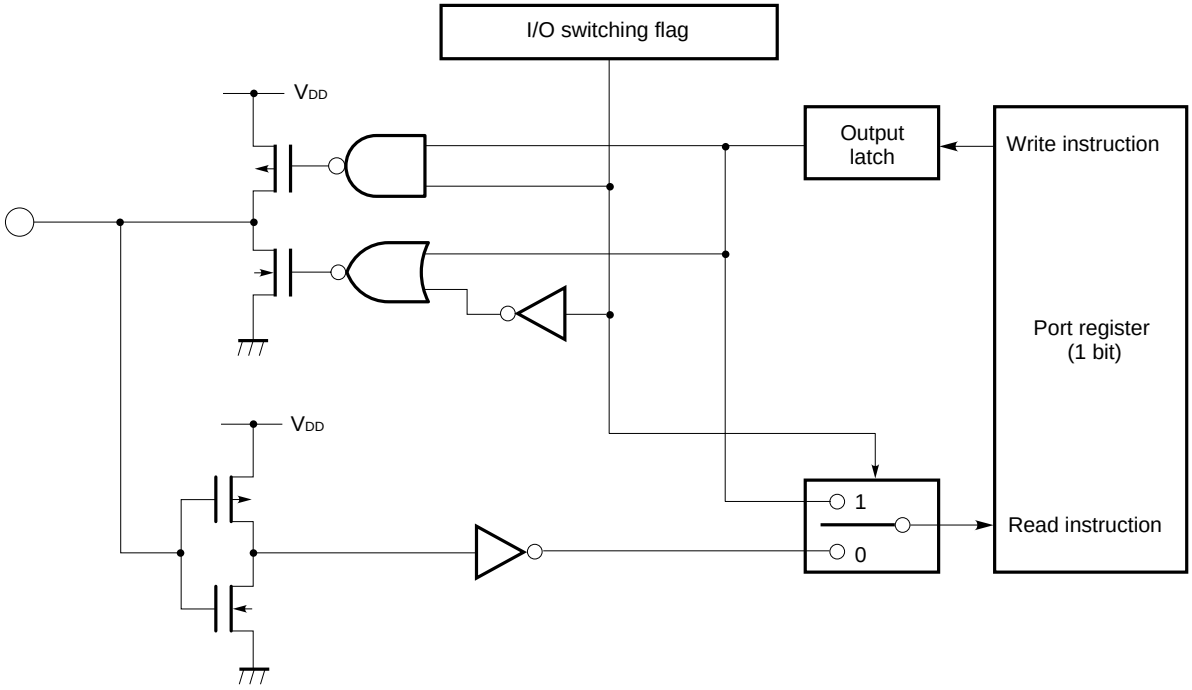
Port	Pin			Data setting method				
				Port register (data memory)				Remarks
	No.	Pin name	I/O	Address	Symbol	Bit symbol (reserved word)		
Port 0A	7	$P0A_0$	I/O (bit I/O)	70H	P0A	b_0	P0A0	I/O switching is performed by the P0ABIO0 or P0ABIO1 flag.
	8	$P0A_1$				b_1	P0A1	
	9	$P0A_2$	Output			b_2	P0A2	
	10	$P0A_3$				b_3	P0A3	
Port 0B	37	$P0B_0$	Input	71H	P0B	b_0	P0B0	
	38	$P0B_1$				b_1	P0B1	
	1	$P0B_2$				b_2	P0B2	
	No target pin					b_3	—	Fixed to 0.
Port 0C	2	$P0C_0$	Output	72H	P0C	b_0	P0C0	
	3	$P0C_1$				b_1	P0C1	
	4	$P0C_2$				b_2	P0C2	
	5	$P0C_3$				b_3	P0C3	
Port 0D	6	$P0D_0$	Output	73H	P0D	b_0	P0D0	Switching to BEEP output is performed by the BEEP0SEL flag.
	No target pin					b_1	—	Fixed to 0.
						b_2	—	
						b_3	—	

9.2 GENERAL-PURPOSE I/O PORTS (P0A0 and P0A1 pins)

9.2.1 Configurations of the I/O ports

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

(1) P0A0 and P0A1 pin



9.2.2 Using the I/O Port

The P0ABIO0 or P0ABIO1 flag of the LCD enable register sets the I/O port to input or output mode.

P0A0 and P0A1 pins are bit I/O ports, each bit of which (each pin) can be set to input or output mode.

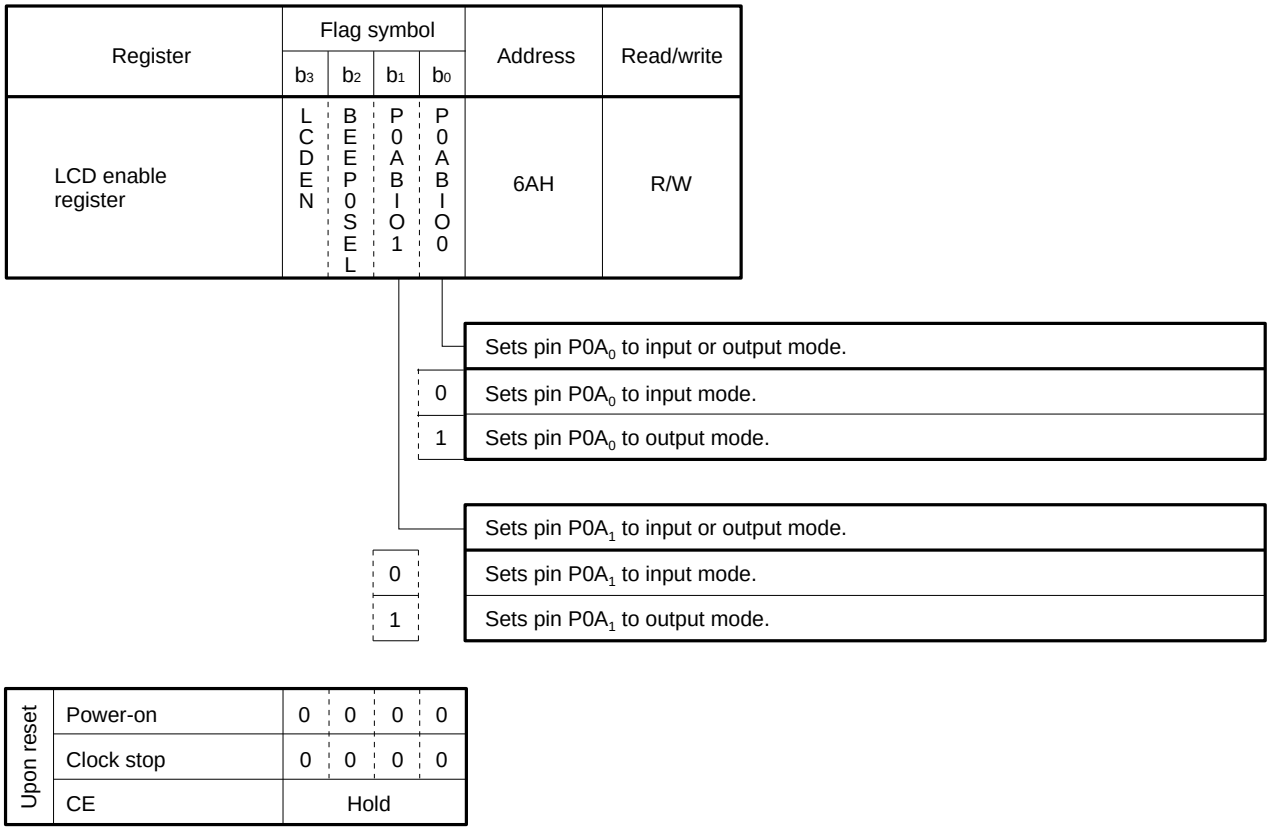
To set the output data, write the data to the corresponding port register. To read the input data, execute an instruction to read the data.

9.2.3 Control Registers of the I/O Ports

The P0ABIO0 and P0ABIO1 flags of the LCD enable register set P0A0 and P0A1 pins in input or output mode, respectively.

Fig. 9-1 shows the configuration and functions of the LCD enable register.

Fig. 9-1 Configuration and Functions of the LCD Enable Registers



9.2.4 Using an I/O Port as an Input Port

Select the pin to be set to input mode, using the P0ABIO0 and P0ABIO1 flags of the LCD enable register.

The $P0A_0$ and $P0A_1$ pins can be set to input mode bit by bit.

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

To read the input data, execute an instruction to read the contents of the port register P0A (such as SKT).

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

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

9.2.5 Using an I/O Port as an Output Port

Select the pin to be set to output mode, using the P0ABIO0 and P0ABIO1 flags of the LCD enable register.

The $P0A_0$ and $P0A_1$ pins can be set in the output mode bit by bit.

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

To set the output data, execute a write instruction (such as MOV) for the port register P0A.

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

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

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

★ **Caution If a read instruction (such as SKT) is executed for a port register specified as an output port, the contents of the output latch and the read data may differ because the status of the output pin is read as is.**

9.2.6 Statuses of the I/O Ports upon Reset

(1) At power-on reset

All pins are set to input mode.

★ The contents of the output latches are reset to 0.

(2) At CE reset

All pins are set to input mode.

The contents of the output latches are retained.

(3) At a clock-stop

All pins are set to input mode.

The contents of the output latches are retained.

(4) In the halt state

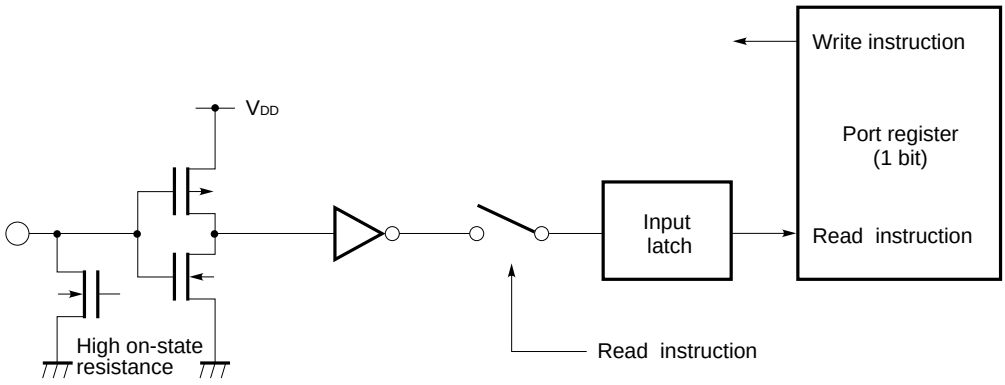
The previous statuses are retained.

9.3 GENERAL-PURPOSE INPUT PORT (P0B$_0$ to P0B$_2$ pins)

9.3.1 Configuration of the Input Port

The configuration of the input port is shown below:

• P0B (P0B$_0$ to P0B$_2$ pins)



9.3.2 Using the Input Port

To read the input data, execute an instruction to read the contents of port register P0B (such as SKT).

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

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

9.3.3 Notes on Using the Input Port

P0B is internally pulled down.

9.3.4 Statuses of the Input Port upon Reset

(1) At power-on reset

All pins are set to input mode.

(2) At CE reset

All pins are set to input mode.

(3) At a clock-stop

All pins are set to input mode.

(4) In the halt state

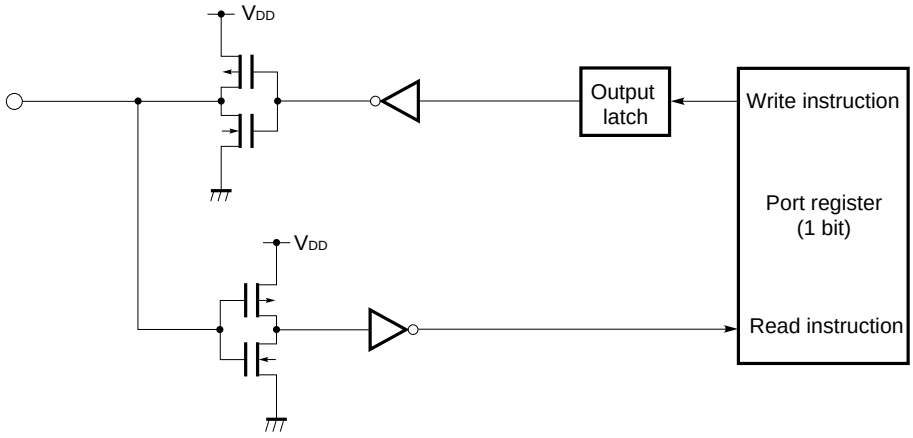
The previous statuses are retained.

9.4 GENERAL-PURPOSE OUTPUT PORTS ($P0A_2$, $P0A_3$, $P0C_0$ to $P0C_3$, and $P0D_0$ pins)

9.4.1 Configurations of the Output Ports

The configurations of the output ports are shown below.

- **P0A ($P0A_2$ and $P0A_3$ pins), P0C ($P0C_0$ to $P0C_3$ pins), P0D ($P0D_0$ pin)**



9.4.2 Using the Output Port

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

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

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

★ **Caution If a read instruction (such as SKT) is executed for a port register, the contents of the output latch and the read data may differ because the status of the output pin is read as is.**

9.4.3 Statuses of the Output Port upon Reset

(1) At power-on reset

The contents of the output latches are output.

★ The contents of the output latches are reset to 0.

(2) Upon CE reset

The contents of the output latch are output.

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

(3) Upon a clock stop

All pins are set to input mode.

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

(4) In the halt state

The previous statuses are retained.

10. TIMERS

The timers in the μPD17015 are used to manage the time required to execute programs.

10.1 OVERVIEW

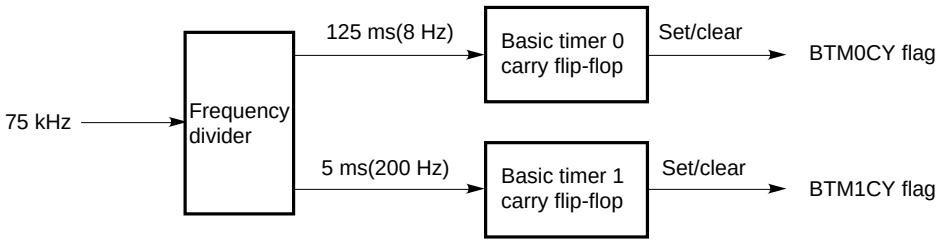
Fig. 10-1 shows the block diagrams of the timers.

The μPD17015 contains the following two different timers.

- Basic timer 0
- Basic timer 1

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

Fig. 10-1 Overview of Timers



10.2 BASIC TIMERS 0 AND 1

10.2.1 Overview of Basic Timers 0 and 1

Basic timer 0 is realized by detecting the state of a basic timer 0 carry flip-flop, using the BTM0CY flag (bit 0 at address 6EH). Basic timer 1 is realized by detecting the state of a basic timer 1 carry flip-flop, using the BTM1CY flag (bit 0 at address 6FH).

The contents of the flip-flops correspond to the states of the BTM0CY and BTM1CY flags on a one-to-one basis, respectively.

Time intervals (pulses) at which the BTM0CY and BTM1CY flags are set are as follows:

BTM0CY flag set pulse: 125 ms (8 Hz)
BTM1CY flag set pulse: 5 ms (200 Hz)

When the BTM0CY and BTM1CY flags are read-accessed for the first time after a power-on reset, these flags are read as 0. Subsequently, the BTM0CY flag is set to 1 after an interval of 125 ms and the BTM1CY flag is set to 1 after an intervals of 5 ms.

Basic timer 0 also controls the timing of the reset (CE reset) by the CE pin.

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

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

See **Chapter 15** for power failure detection.

10.2.2 Basic Timer 0 Carry Flip-Flop, Basic Timer 1 Carry Flip-Flop, BTM0CY Flag, and BTM1CY Flag

The flip-flop is set at constant intervals, and its state is detected using the BTM0CY and BTM1CY flags.

The BTM0CY and BTM1CY flags are read-only. These flags are reset by reading their contents using the instructions listed in Table 10-1 (Read & Reset).

The BTM0CY and BTM1CY flags are “0” at a power-on reset. It becomes “1” at a CE reset. So, it can be used as a power failure detection flag.

Even when power is supplied, the BTM0CY and BTM1CY flags will not be set until their contents have been read using the instructions listed in Table 10-1. Once a read instruction is executed, the flag is set at constant intervals.

Fig. 10-2 shows the configuration and function of the basic timer 0 carry flip-flop judge register. Fig. 10-3 shows the configuration and function of the basic timer 1 carry flip-flop judge register.

Table 10-1 Instructions Used to Reset the BTM0CY and BTM1CY Flags

<table>
<tr><th>Mnemonic</th><th>Operand</th><th>Mnemonic</th><th>Operand</th></tr>
<tr><td rowspan="10">ADD
ADDC
SUB
SUBC
AND
OR
XOR
SKE
SKEG
SKLT
SKNE</td><td rowspan="10">m, #n4</td><td>ADD
ADDC
SUB
SUBC
AND
OR
XOR
LD</td><td>r, m</td></tr>
<tr><td>SKT
SKF</td><td>m, #n</td></tr>
<tr><td rowspan="2">MOV</td><td>@r, m</td></tr>
<tr><td>m, @rNote</td></tr>
</table>

Note When the low address of m is 6H, and 0EH or 0FH is written into r.

Remark m = 6EH or 6FH

Fig. 10-2 Configuration of the Basic Timer 0 Carry Flip-Flop Judge Register

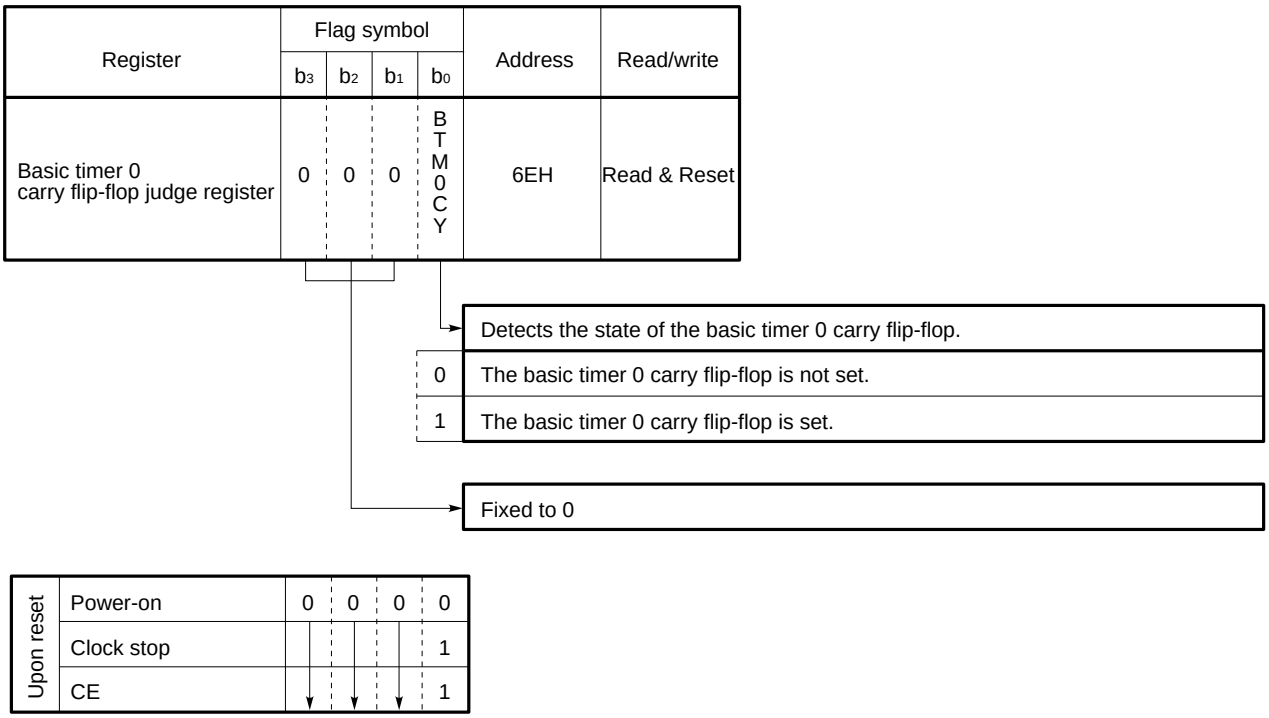
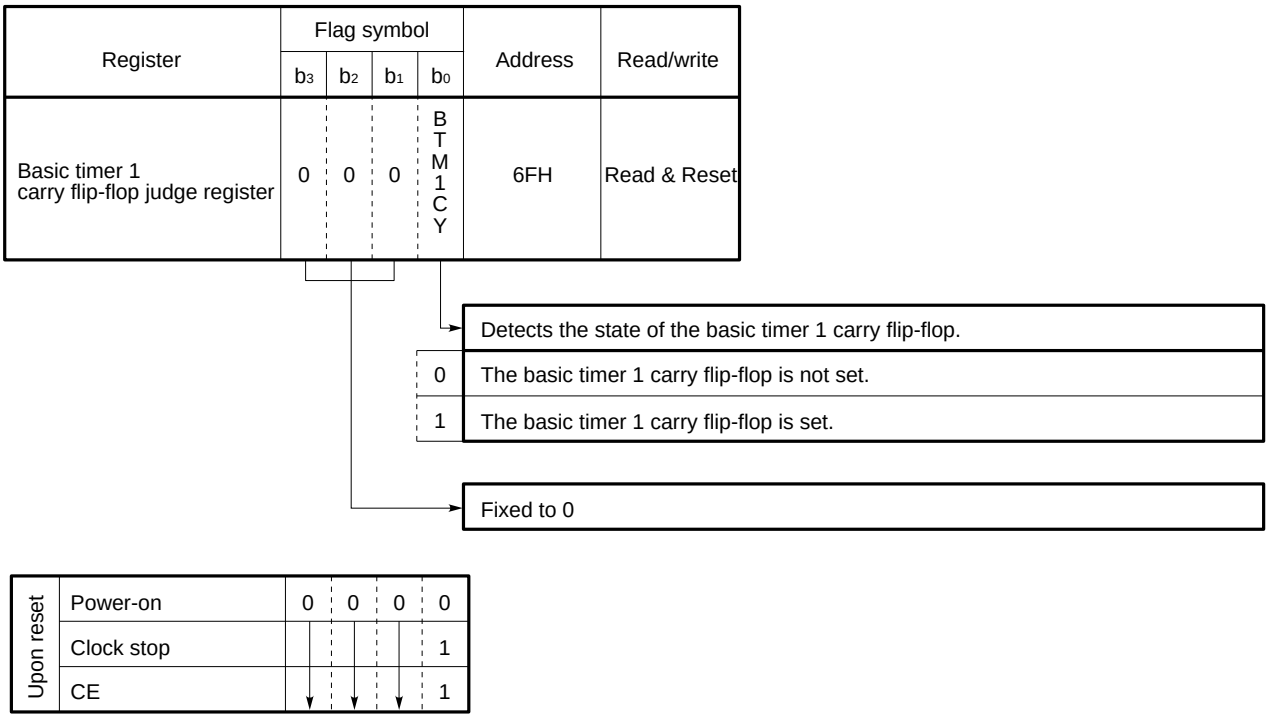


Fig. 10-3 Configuration of the Basic Timer 1 Carry Flip-Flop Judge Register



10.2.3 Example of Using Basic Timer 0

A sample program follows. The following program performs process A at every one second.

Example

```
        M1      MEM 0.10H           ; 1 second counter
LOOP:
        SKT1    BTM0CY              ; Branches to NEXT if BTM0CY is 0.
        BR      NEXT
        ADD     M1, #0100B          ; Adds 4 to M1.
        SKT1    CY                  ; Branches to NEXT if CY flag is 0.
        BR      NEXT                ; Performs process A if CY flag is 1.
        Process A
NEXT:
        Process B                   ; Performs B and branches to LOOP.
        BR      LOOP
```

10.2.4 Time Interval Error in Basic Timers 0 and 1

The time interval at which the BTM0CY and BTM1CY flags are detected must be shorter than the time interval at which the BTM0CY and BTM1CY flags are set. (See **Section 10.2.5.**)

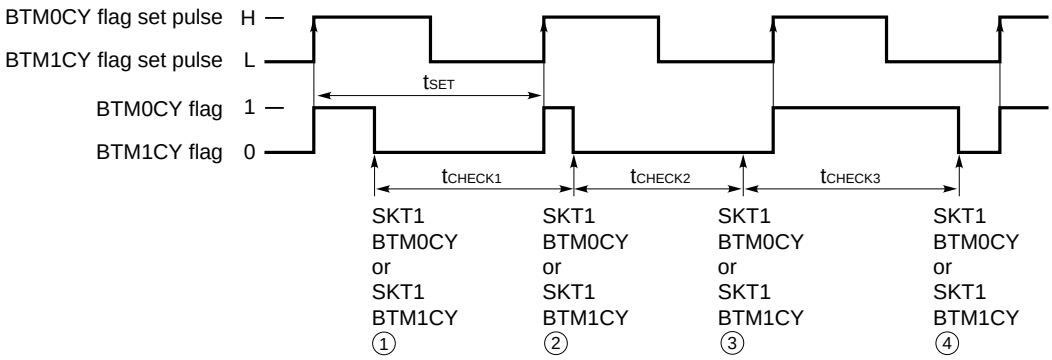
Assuming that the BTM0CY or BTM1CY flag is detected at intervals of t_{CHECK} and set at intervals of t_{SET} (125 or 5 ms, respectively), the relationship between t_{CHECK} and t_{SET} must be as follows:

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

Under this condition, the time interval error encountered when the BTM0CY and BTM1CY flags are detected is as follows:

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

Fig. 10-4 Basic Timers 0 and 1 Error Related to the Detection Time of the BTM0CY and BTM1CY Flags



As shown in Fig. 10-4, when the BTM0CY and BTM1CY flags are detected at ②, the timer is updated because the flag is "1". When the flag is detected at ③, because it is "0", the timer is not updated until it is detected again at ④. Therefore, the timer is incremented by t_{CHECK3}.

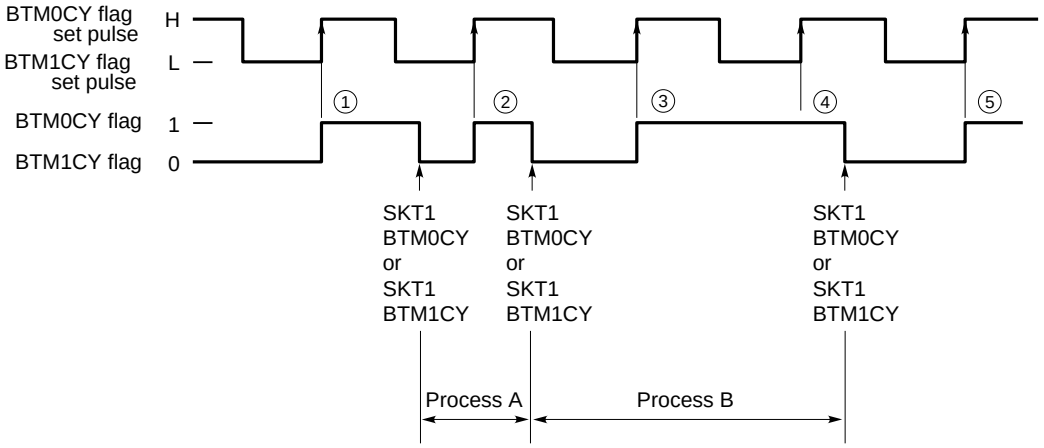
10.2.5 Cautions for Using Basic Timers 0 and 1

(1) BTM0CY and BTM1CY flags detection time interval

As described in **Section 10.2.4**, keep the BTM0CY and BTM1CY flags detection time interval shorter than the BTM0CY and BTM1CY flags set time interval.

Otherwise, the BTM0CY and BTM1CY flags cannot be set if the time required for process B is longer than the BTM0CY and BTM1CY flags set time interval, as shown in Fig. 10-5.

Fig. 10-5 Detection of the BTM0CY and BTM1CY Flags



Because the time required for process B is long after the BTM0CY and BTM1CY flags, which is set to "1" at ②, is detected, the BTM0CY and BTM1CY flags, which is set to "1" at ③, cannot be detected.

(2) Sum of the timer update process time and the BTM0CY and BTM1CY flags detection time interval

As explained in (1), the BTM0CY and BTM1CY flags detection time interval must be kept shorter than the BTM0CY and BTM1CY flags set time interval.

Even when the BTM0CY and BTM1CY flags detection time interval is short, however, if the timer update process time is long, a CE reset may prevent a normal timer update process.

The following conditions must therefore be satisfied.

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

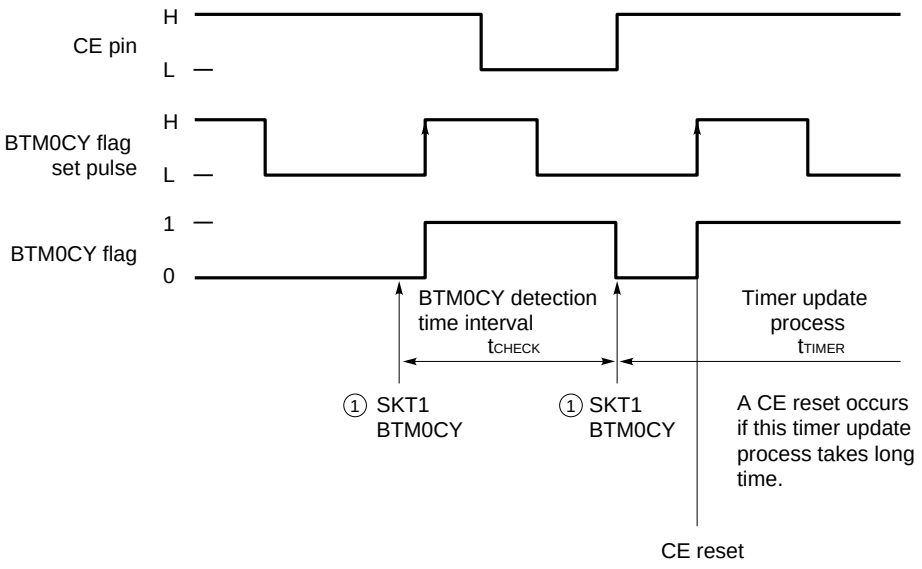
where t_{CHECK} : BTM0CY and BTM1CY flags detection time interval
t_{TIMER} : Timer update process time
t_{SET} : BTM0CY and BTM1CY flags set time interval

The coding that meets these conditions is given below.

Example Timer update process and BTM0CY flag detection time interval

```
START:                              ; Program address 0000H
        Process A
BTIMER:
        ; ①
        SKT1        BTM0CY          ; Updates the timer if the BTM0CY flag is set to 1.
        BR          AAA
        Timer update
        BR          BTIMER
AAA:
        Process B
        BR          BTIMER
```

The timing chart for this coding is as follows:



10.2.6 Cautions for Using Basic Timer 0

(1) Correcting the basic timer 0 at a CE reset

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

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

The BTM0CY flag is cleared (0) when the supply voltage is first applied (at power-on reset). It is not set until its contents have been read by an instruction listed in Table 10-1.

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

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

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

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

To solve this problem, it is necessary to update the clock timer if an attempt to detect a power failure detects a CE reset. See **Section 15.5** for power failure detection.

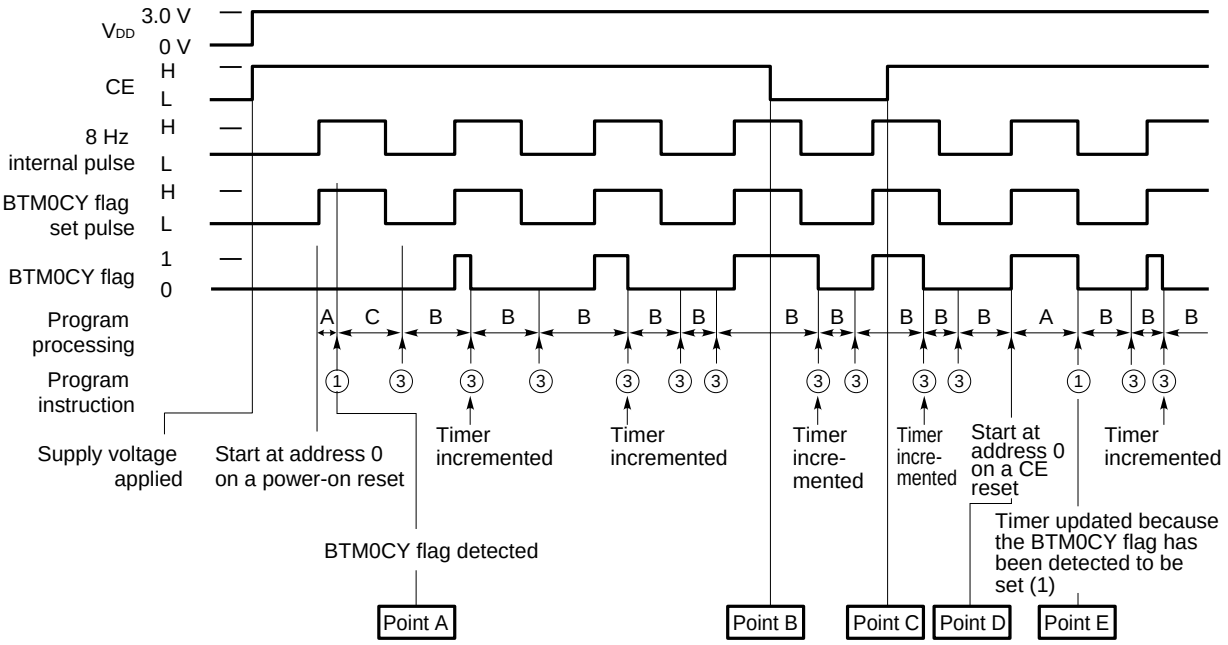
Example Correcting the timer at a CE reset (when the BTM0CY flag is used for both power failure detection and timer update)

```
START :                 ; Program address 0000H
        Process A
      ; ①
        SKT1    BTM0CY      ; Built-in macro
                            ; Checks the BTM0CY flag and branches to INITIAL
        BR      INITIAL     ; if the flag is "0" (power failure detected).

BACKUP :
      ; ②
        Timer update by 125 ms   ; Timer correction because of backup (CE reset)
LOOP:
      ; ③
        Process B           ; While performing process B,
        SKF1    BTM0CY      ; tests the BTM0CY flag and updates the timer.
        BR      BACKUP
        BR      LOOP
INITIAL :
        Process C
        BR      LOOP
```

Fig. 10-6 shows the timing chart for the above program.

Fig. 10-6 Timing Chart



As shown in Fig. 10-6, when supply voltage VDD is applied, the 8 Hz internal pulse rises to make the program start at address 0000H.

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

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

Because the BTM0CY flag is once read-accessed at point A, the BTM0CY flag will be set (1) at intervals of 125 ms.

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

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

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

Also, as easily seen from the figure, if the clock is not updated by 125 ms at point E, the clock loses 125 ms every time a CE reset occurs.

If process A takes more than 125 ms to detect for a power failure at point E, it is impossible to detect the BTM0CY flag for two cycles. Therefore, process A must be performed within 125 ms.

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

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

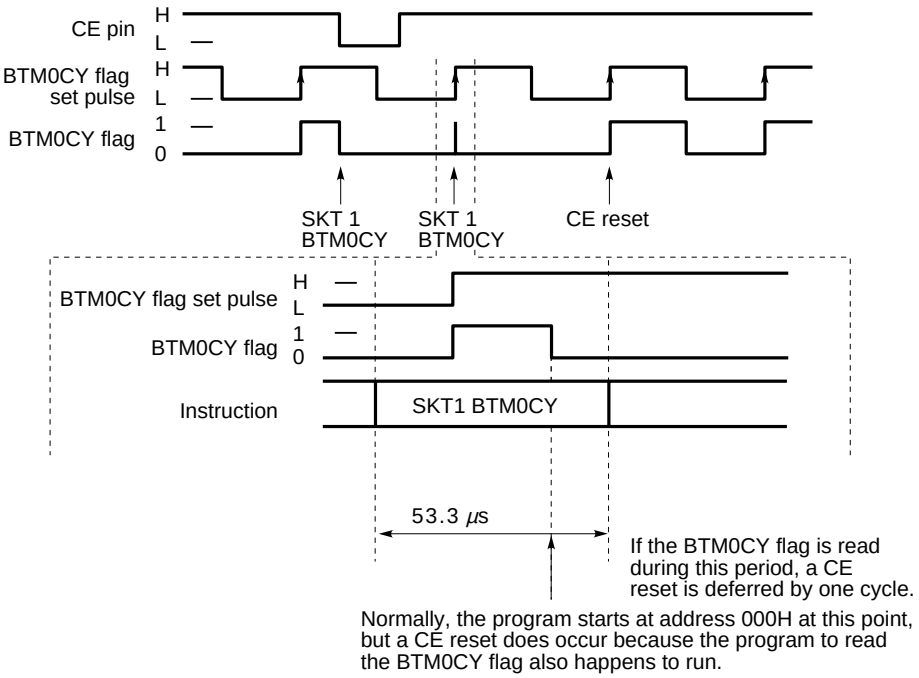
As described in (1), a CE reset occurs at the same time the BTM0CY flag is set (1).

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

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

This operation is illustrated in Fig. 10-7.

Fig. 10-7 Operation That Occurs If a CE Reset Occurs Simultaneously with a BTM0CY Flag Read Instruction



11. PLL FREQUENCY SYNTHESIZER

The PLL (Phase Locked Loop) frequency synthesizer is used to lock MF (Medium Frequency), HF (High Frequency) or VHF (Very High Frequency) band frequencies to a fixed frequency using a phase error comparison system.

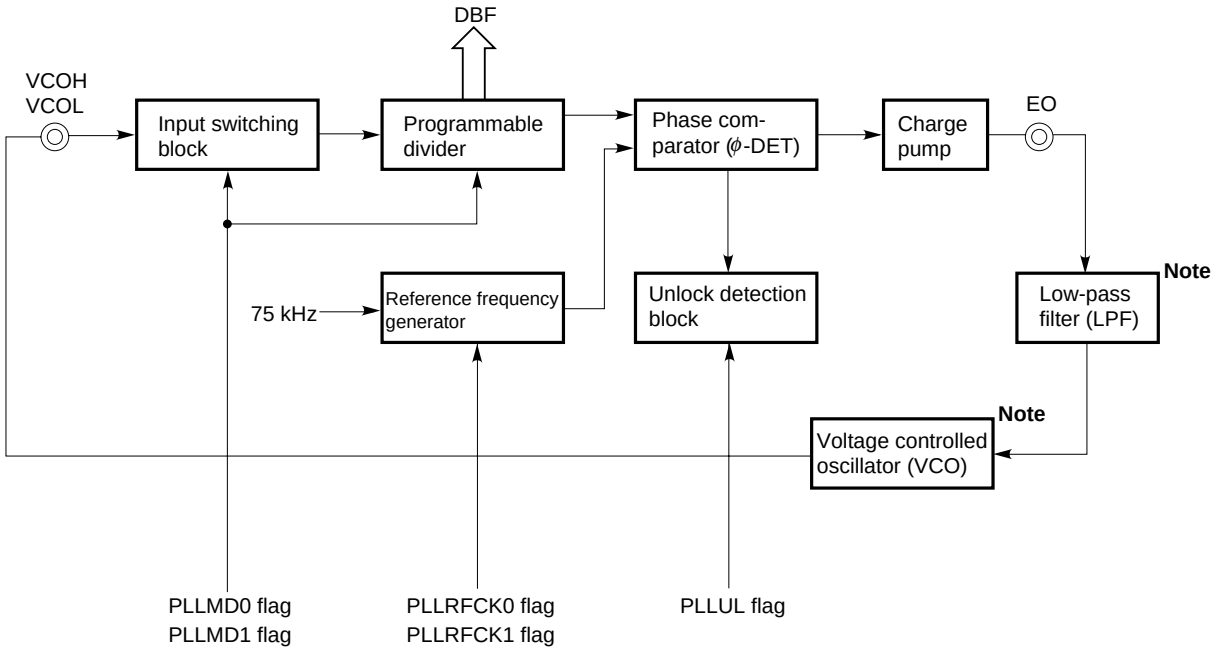
11.1 GENERAL

Fig. 11-1 outlines the PLL frequency synthesizer. A PLL frequency synthesizer can be built by connecting a low pass filter (LPF), voltage controlled oscillator (VCO), and prescaler externally.

The PLL frequency synthesizer divides the signal input from the VCOH or VCOL pin using a programmable divider and outputs the phase error with the reference frequency to the EO pin.

The PLL frequency synthesizer operates only when the CE pin is high level. When the CE pin is low level, the PLL frequency synthesizer is disabled. For a description of the PLL disabled state, see **Section 11.5**.

Fig. 11-1 PLL Frequency Synthesizer



Note External circuit.

Remarks 1. PLLRFCK0 and PLLRFCK1 (bits 0 and 1 of PLL mode selection register: see **Fig. 11-3**): Set the PLL frequency synthesizer reference frequency f_r.

2. PLLMD0 and PLLMD1 (bits 2 and 3 of PLL mode selection register: see **Fig. 11-3**): Set the frequency division method of the PLL frequency synthesizer.

3. PLLUL (bit 0 of PLL unlock flip-flop judge register: see **Fig. 11-8**): Detects the state of the unlock flip-flop.

11.2 INPUT SWITCHING BLOCK AND PROGRAMMABLE DIVIDER

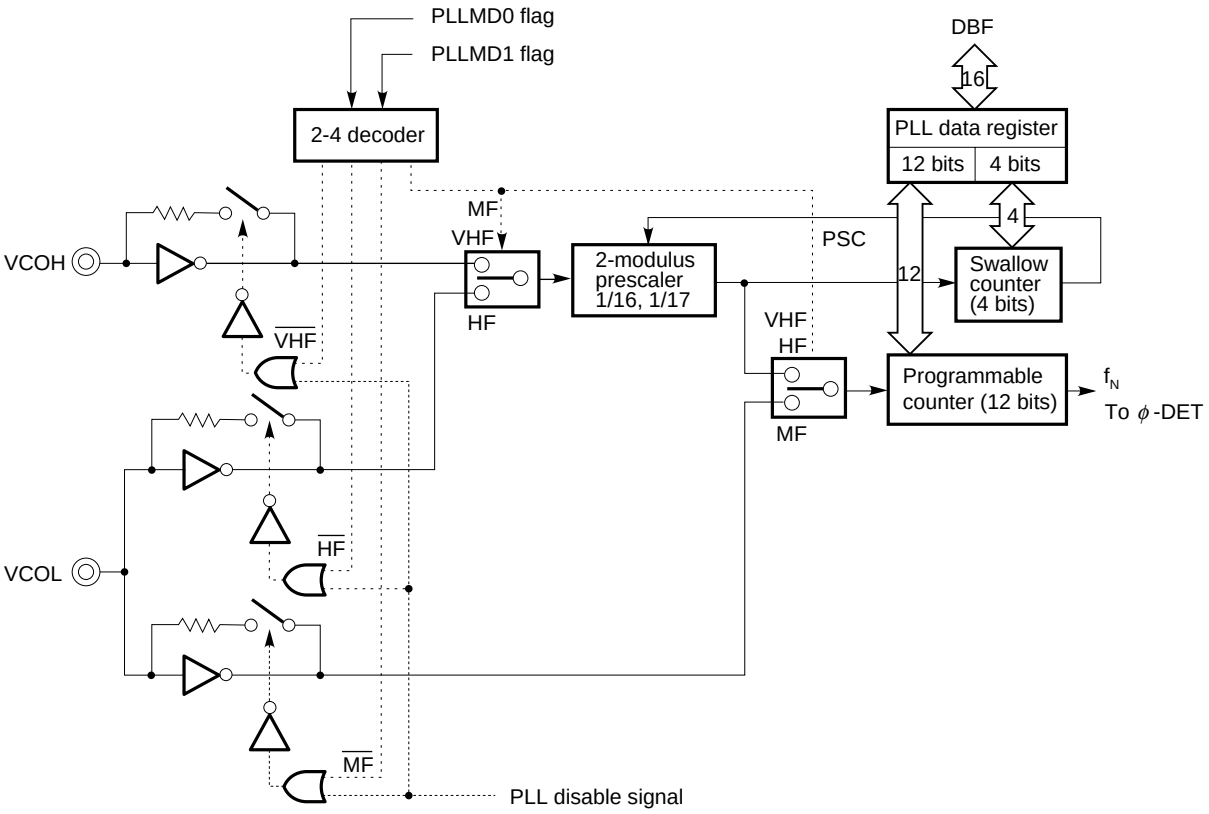
11.2.1 Configuration

Fig. 11-2 shows the configuration of the input switching block and programmable divider.

The input switching block consists of the VCOH and VCOL pins and their input amplifiers.

The programmable divider consists of a two-modulus prescaler, swallow counter, programmable counter, and division method selection switches.

Fig. 11-2 Input Switching Block and Programmable Divider



11.2.2 Functions

The input switching block and programmable divider select the input pin and division method to be used by the PLL frequency synthesizer.

Either the VCOH or VCOL pin can be selected as the input pin.

The selected pin will have an intermediate potential (about half of V_{DD}).

The input to the pin is provided by an AC amplifier. The DC element should be eliminated from the input signal by connecting a capacitor in series with the pin.

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

The programmable divider divides the signal input by means of the selected division method, using the division ratio set by the swallow counter and programmable counter.

Table 11-1 lists the division methods and input pins (VCOH and VCOL pins) to be used.

The input pin and division method to be used are selected according to the PLLMD0 and PLLMD1 flags of the PLL mode select register.

Fig. 11-3 shows the configuration of the PLL mode select register, together with its functions.

The division ratio of the programmable divider is set by the PLL data register via data buffers.

Section 11.2.3 describes the programmable divider and PLL data register.

Table 11-1 Input Pins and Division Methods

Division method	Pin used	Input frequency range (MHz)	Input amplitude ($V_{p\text{-}p}$)	Possible division ratio	Division ratio that can be set in the data buffers
Direct division (MF)	VCOL	0.5 to 8	0.2	16 to $2^{12} - 1$	010×H - FFF×H (×: Don't care about low-order 4 bits.)
Pulse swallow (HF)	VCOL	6 to 55	0.2	256 to $2^{16} - 1$	0100H-FFFFH
Pulse swallow (VHF)	VCOH	40 to 220	0.2	256 to $2^{16} - 1$	0100H-FFFFH

Fig. 11-3 Configuration and Functions of the PLL Mode Select Register

Name	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
PLL mode select register	PLLMD1	PLLMD0	PLLRFCK1	PLLRFCK0	6BH	R/W

Sets reference frequency f_r of the PLL frequency synthesizer.[Note]

0	0	1 kHz
0	1	3 kHz
1	0	5 kHz
1	1	25 kHz

Sets the division method used by the PLL frequency synthesizer.

0	0	Disables pins VCOL and VCOH
0	1	Selects direct division (VCOL pin, MF mode).
1	0	Selects pulse swallow (VCOH pin, VHF mode).
1	1	Selects pulse swallow (VCOL pin, HF mode).

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

Note For details of the reference frequency, see **Section 11.3**.

Items (1) to (4), below, outline the division methods:

(1) Direct division (MF)

The VCOL pin is used.

The VCOH pin enters the floating state.

In direct division, the signal input is divided according to the programmable counter only.

(2) Pulse swallow (HF)

The VCOL pin is used.

The VCOH pin enters the floating state.

When using pulse swallow, the signal input is divided according to the settings of the swallow counter and programmable counter.

(3) Pulse swallow (VHF)

The VCOH pin is used.

The VCOL pin enters the floating state.

When using pulse swallow, the signal input is divided according to the settings of the swallow counter and programmable counter.

(4) VCOL and VCOH pins disabled

The VCOL and VCOH pins enter the floating state.

In this state, the phase comparator and reference frequency generator stop the output, and the charging pump sets the EO output pin to the floating state. This operation is the same as that performed in the PLL disable state, described in **Section 11.5**.

11.2.3 Programmable Divider and PLL Data Register

The programmable divider divides the signal input from the VCOH or VCOL pin, using the division ratio set by the swallow counter and programmable counter.

The swallow counter and program counter are 4-bit and 12-bit binary down-counters, respectively.

The division ratios for the swallow counter and program counter are set in the PLL data register (PLLR: peripheral address 41H) via data buffers.

Data is written to and read from the PLL data register by the PUT PLLR, DBF and GET DBF, PLLR instructions.

The division ratio is referred to as the N value.

For details of setting division ratio (N value) for each division method, see **Section 11.6**.

(1) PLL data register and data buffer

Fig. 11-4 shows the relationship between the PLL data register and data buffers.

In direct division, the high-order 12 bits are valid. When pulse swallow is being used, all 16 bits are valid.

In direct division, the above 12 bits are set in the programmable counter.

When pulse swallow is being used, the high-order 12 bits are set in the programmable counter while the low-order four bits are set in the swallow counter.

(2) Relationship between division value N of the programmable divider and the divided output frequency

The following expressions indicate the relationship between value N, set in the PLL data register, and frequency f_N of the signal output divided by the programmable divider:

(a) Direct division (MF)

$$f_N = \frac{f_{IN}}{N} \quad \text{(N: 12 bits)}$$

(b) Pulse swallow (HF, VHF)

$$f_N = \frac{f_{IN}}{N} \quad \text{(N: 16 bits)}$$

Fig. 11-4 Relationship between the PLL Data Register and Data Buffers

Name	Data buffer															
Symbol	DBF3				DBF2				DBF1				DBF0			
Address	0CH				0DH				0EH				0FH			
Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data	Transfer data															

16

GET instruction allowed

PUT instruction allowed

Peripheral register																		
Name	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Symbol	Peripheral address	Peripheral hardware
PLL data register	Valid data																PLLR	41H	PLL frequency synthesizer

Sets the division ratio of the PLL frequency synthesizer.

Direct division	0	Don't care
	|	Not to be set
	15 (00FH)	Don't care
	16 (010H)	Don't care
	|	Division ratio N:N = x
	x	Don't care
	|	
	2^{12}–1 (FFFH)	Don't care
Pulse swallow	0	Not to be set
	|	
	255 (00FFH)	
	256 (0100H)	Division ratio N:N = x
	|	
	x	
	|	
	2^{16}–1 (FFFH)	

11.3 REFERENCE FREQUENCY GENERATOR

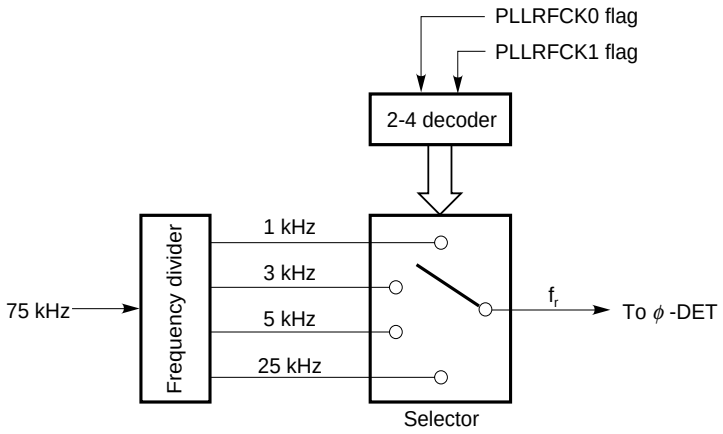
Fig. 11-5 shows the configuration of the reference frequency generator.

The reference frequency generator generates the PLL frequency synthesizer reference frequency "f_r" by dividing the 75-MHz signal of a crystal oscillator.

The reference frequency f_r can be selected from among 1 kHz, 3 kHz, 5 kHz, and 25 kHz.

The reference frequency f_r is selected with the PLLRFCK0 and PLLRFCK1 flags of the PLL mode selection register (see **Fig. 11-3**).

Fig. 11-5 Reference Frequency Generator



11.4 PHASE COMPARATOR (φ-DET), CHARGE PUMP AND UNLOCK DETECTION BLOCK

11.4.1 Configuration of Phase Comparator, Charge Pump and Unlock Detection Block

Fig. 11-6 shows the configuration of the phase comparator, charge pump and unlock detection block.

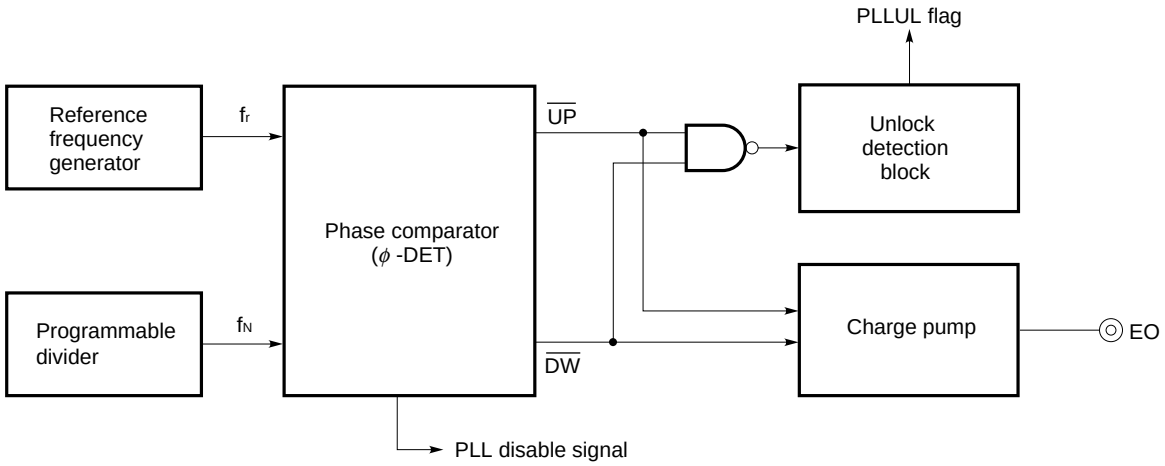
The phase comparator compares the phase of the divided frequency (f_N) signal output from the programmable divider and that of the reference frequency (f_r) signal output from the reference frequency generator and outputs an up request signal ($\overline{UP}$) or down request signal ($\overline{DW}$).

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

The unlock detection block detects the PLL frequency synthesizer unlocked state from $\overline{UP}$ and $\overline{DW}$.

Sections 11.4.2 to **11.4.4** describe the operation of the phase comparator, charge pump, and unlock detection block.

Fig. 11-6 Phase Comparator, Charge Pump and Unlock Detection Block



11.4.2 Phase Comparator Functions

As shown in Fig. 11-6, the phase comparator compares the phase of the programmable divider divided (f_N) output and that of the reference frequency (f_r) signal and outputs an up request signal or down request signal.

That is, if divided frequency "f_N" is lower than reference frequency "f_r", an up request signal is output, and if divided frequency "f_N" is higher than reference signal "f_r", a down request signal is output.

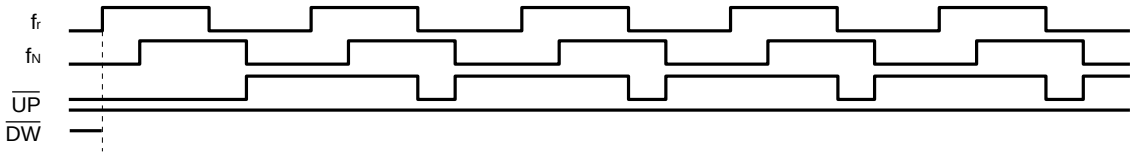
Fig. 11-7 shows the relationship among the reference frequency f_r, division frequency f_N, up request signal, and down request signal.

In the PLL disabled state, neither an up request signal nor a down request signal is output.

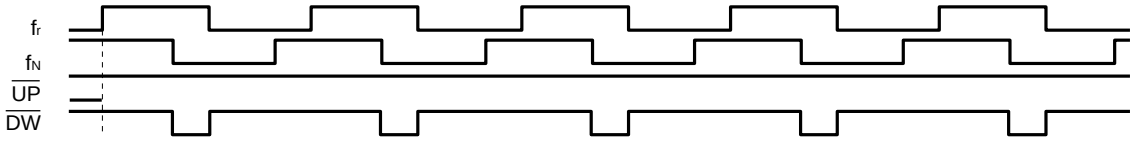
The up request and down request signals are input to the charge pump and unlock detection block.

Fig. 11-7 f_r, f_N, $\overline{UP}$, and $\overline{DW}$ Signal Relationship

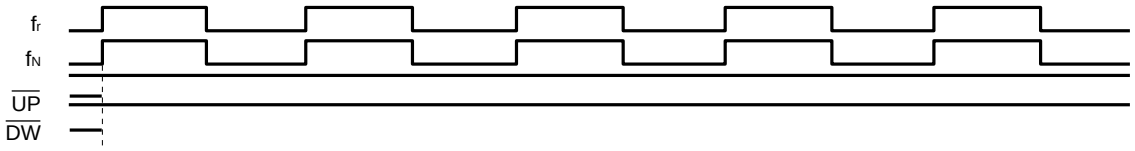
(a) When f_N phase lags f_r phase



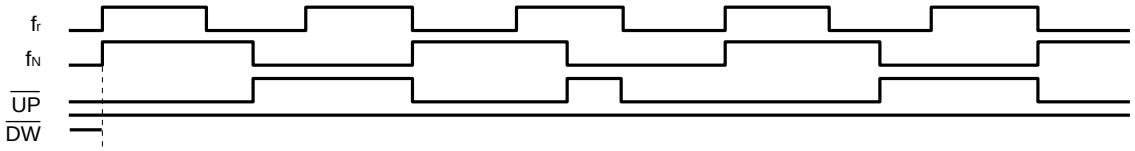
(b) When f_N phase leads f_r phase



(c) When f_N and f_r are same phase



(d) When f_N frequency lower than f_r frequency



11.4.3 Charge Pump

As shown in Fig. 11-6, the charge pump outputs the up request signal or down request signal sent from the phase comparator, from the error out pin (EO pin).

Error output pin output, division frequency f_N, and reference frequency f_r have the following relation:

When reference frequency f_r > division frequency f_N: Low level output
When reference frequency f_r < division frequency f_N: High level output
When reference frequency f_r = division frequency f_N: Floating

11.4.4 Functions of Unlock Detection Block

As shown in Fig. 11-6, the unlock detection block detects the PLL frequency synthesizer unlocked state from the phase comparator up request and down request signals.

That is, since the up request signal or down request signal outputs low level while the PLL frequency synthesizer is in the unlocked state, the unlocked state can be detected by monitoring this low level signal.

When the PLL frequency synthesizer is in the unlocked state, the unlock flip-flop is set (1). The state of the unlock flip-flop is detected by the PLLUL flag of PLL unlock flip-flop judge register (see **Fig. 11-8**). The unlock flip-flop is set at the period of the reference frequency fr selected at the time.

The contents of the PLL unlock flip-flop judge register are read with instructions shown in Table 11-2 and reset (Read & Reset). The unlock flip-flop must be detected at a period longer than reference frequency f_r period $1/f_r$.

Fig. 11-8 Configuration and Functions of PLL Unlock Flip-Flop Judge Register

Register	Flag symbol b_3	b_2	b_1	b_0	Address	Read/write
PLL unlock flip-flop judge register	0	0	0	PLLUL	6DH	Read & Reset

	Detects the state of the unlock flip-flop.
0	Unlock flip-flop = 0 : PLL locked
1	Unlock flip-flop = 1 : PLL unlocked

Fixed to 0.

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

* Undefined

Table 11-2 Instructions Which Reset the PLL Unlock Flip-Flop Judge Register

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

Note When the low address of m is 6H, and 0DH is written into r.

Remark m = 6DH

11.5 PLL DISABLED STATE

The PLL frequency synthesizer is disabled while the CE pin (pin 13) is low level.

The PLL frequency synthesizer is also disabled when VCOH/VCOL pin disabled is selected by the PLL mode selection register, even if the CE pin is set to high level.

Table 11-3 shows the state of each block at PLL disabled.

Since the PLL mode selection register is not initialized (previous state is held) at CE reset, it is reset to its previous state when the CE pin rises to high level after dropping to low level and PLL disabled is set.

Therefore, when PLL disabled must be set at CE reset, the PLL reference clock selection register must be initialized by program.

At power-on reset, PLL disabled is set.

Table 11-3 Operation of Each Block at PLL Disabled

Condition / Block	CE pin = low level (PLL disabled)	CE pin = high level (PLLMD = 0000B: VCOH/VCOL pin disabled)
VCOL/VCOH pin	Floating	
Programmable divider	Frequency division stopped	
Reference frequency generator	Output stopped	
Phase comparator		
Charge pump	EO pin floated	

11.6 PLL FREQUENCY SYNTHESIZER USE

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

(1) Frequency division method : Direct division (MF), pulse swallow (HF, VHF)
(2) Pins to be used : VCOL and VCOH pins
(3) Reference frequency : f_r
(4) Division value : N

Sections 11.6.1 to **11.6.3** describe the PLL data setting method for each division method (MF, HF, or VHF).

11.6.1 Direct Division Method (MF)

(1) Selecting the division method to be used

The direct division method is selected by setting the PLLMD0 or PLLMD1 flag of the PLL mode selection register.

(2) Pins to be used

The VCOL pin becomes available when the direct division method is selected.

(3) Reference frequency f_r setting

The reference frequency is set by the PLLRFCK0 or PLLRFCK1 flag of the PLL mode selection register.

(4) Division value N computation method

Division value N is computed as follows:

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

f_{VCOL} : VCOL pin input frequency
f_r : Reference frequency

(5) PLL data setting example

The method of setting the data to receive a MF band broadcast station is shown below.

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

Division value N is:

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

= 270H (hexadecimal)

Data is written to the PLL data register and PLL mode selection register as follows:

PLLR				PLLMD
0 0 1 0	0 1 1 1	0 0 0 0	Any value	0 1 0 1
2	7	0		MF, 3 kHz

11.6.2 Pulse Swallow Method (HF)

(1) Selection of the division method

The pulse swallow method is selected by the PLLMD0 or PLLMD1 flag of the PLL mode selection register.

(2) Pins to be used

The VCOL pin becomes available when the pulse swallow method is selected.

(3) Reference frequency f_r setting

The reference frequency is set with the PLLRFCK0 or PLLRFCK1 flag of the PLL mode selection register.

(4) Division value N computation

Division value N is computed as follows:

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

f_{VCOL} : VCOL pin input frequency
f_r : Reference frequency

(5) PLL data setting example

The data to be set to receive a SW band broadcast station is shown below.

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

Division value N is:

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

$$= 1446H \text{ (hexadecimal)}$$

Data is written into the PLL data and PLL mode selection registers as follows:

PLLR				PLLMD
0 0 0 1	0 1 0 0	0 1 0 0	0 1 1 0	1 1 1 0
1	4	4	6	HF, 5 kHz

11.6.3 Pulse Swallow Method (VHF)

(1) Selection of division method

The pulse swallow method is selected by setting the PLLMD0 or PLLMD1 flag of the PLL mode selection register.

(2) Pins to be used

The VCOH pin becomes available when the pulse swallow method is selected.

(3) Reference frequency f_r setting

The reference frequency is set by the PLLRFCK0 or PLLRFCK1 flag of the PLL mode selection register.

(4) Division value N computation method

Division value N is computed as follows:

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

f_{VCOH} : VCOH pin input frequency
f_r : Reference frequency

(5) PLL data setting example

The data to be set to receive an FM band broadcast station is shown below.

Receiving frequency : 100.0 MHz (FM band)
Reference frequency : 25 kHz
Intermediate frequency : +10.7 kHz

Division value N is:

$$N = \frac{f_{VCOH}}{f_r} = \frac{100.0 + 10.7}{0.025} = 4428 \text{ (decimal)}$$

$$= 114CH \text{ (hexadecimal)}$$

Data is written into the PLL data and PLL mode selection registers as follows:

PLLR				PLLMD
0 0 0 1	0 0 0 1	0 1 0 0	1 1 0 0	1 0 1 1
1	1	4	C	VHF, 25 kHz

11.7 STATE AT RESET

11.7.1 At Power-On Reset

Since the PLL mode selection register is initialized to 0000B, the PLL disabled state is set.

11.7.2 At Clock-Stop

The PLL disabled state is set at the time the CE pin drops to low level.

11.7.3 At CE Reset

(1) CE reset caused by clock stop

Since clock-stop initializes the PLL mode selection register to 0000B, the PLL disabled state is set.

(2) CE reset when clock not stopped

Since the PLL mode selection register retains its previous state, the previous state is set when the CE pin rises to high level.

11.7.4 During the Halt State

If the CE pin is high level, the set state is held.

12. BEEP

12.1 CONFIGURATION AND FUNCTIONS

Fig. 12-1 shows an overview of BEEP.

A 3-kHz clock is output from the $P0D_0$/BEEP pin.

According to the settings of the BEEP0SEL flag of the LCD enable register, the output switching block controls whether the $P0D_0$/BEEP pin functions as a general-purpose output port or as a BEEP output pin.

The clock generation block generates the 3-kHz clock to be output to the BEEP pin.

Fig. 12-2 shows the configuration and functions of the LCD enable register.

Fig. 12-1 Overview of BEEP

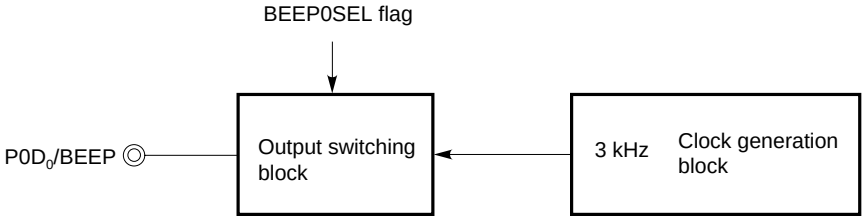


Fig. 12-2 Configuration and Functions of the LCD Enable Register

Name	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
LCD enable register	LCDEN	BEEP0SEL	P0ABIO1	P0ABIO0	6AH	④ R/W

Sets the function of the $P0D_0$/BEEP pin.
0 Uses the $P0D_0$/BEEP pins as a general-purpose output port.
1 Uses the $P0D_0$/BEEP pin as a BEEP output pin.

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

12.2 STATE AT RESET

12.2.1 At Power-On Reset

The BEEP0SEL flag is reset. Consequently, the $P0D_0$/BEEP pin is used as a general-purpose output port. Because the latch value for the output port is not defined, the port outputs undefined data.

12.2.2 At Clock-Stop Reset

The BEEP0SEL flag is reset. Consequently, the $P0D_0$/BEEP pin is used as a general-purpose output port. Because the latch value is not defined for the output port, the port outputs undefined data.

12.2.3 At CE Reset

The $P0D_0$/BEEP pin holds the previous output state.

12.2.4 During the Halt State

The $P0D_0$/BEEP pin holds the previous output state.

13. LCD CONTROLLER/DRIVER

The LCD (liquid crystal display) controller/driver, in combination with the segment signal output, controls an LCD display of up to 36 segments.

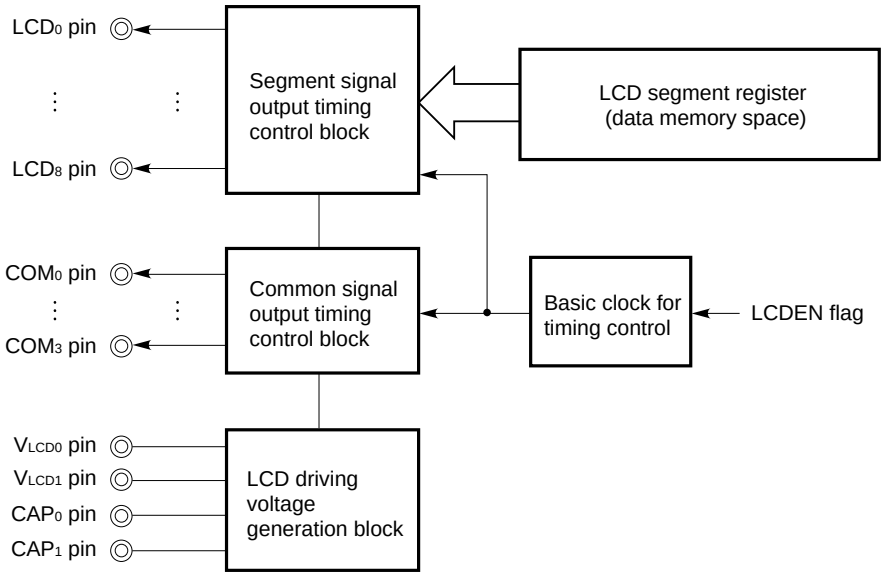
13.1 Overview

Fig. 13-1 shows an overview of the LCD controller/driver.

The LCD controller/driver controls a display consisting of up to 36 segments by using both the common signal output (COM_0 to COM_3 pins) and segment signal output (LCD_0 to LCD_8 pins).

The driving method features a 1/4 duty cycle and 1/2 bias, a frame frequency of 150 Hz, and a driving voltage of V_{DD} - V_{LCD0}.

Fig. 13-1 Overview of LCD Controller/Driver



Remark LCDEN (bit 3 of the LCD enable register; see **Fig. 13-6**) turns the entire LCD display on or off.

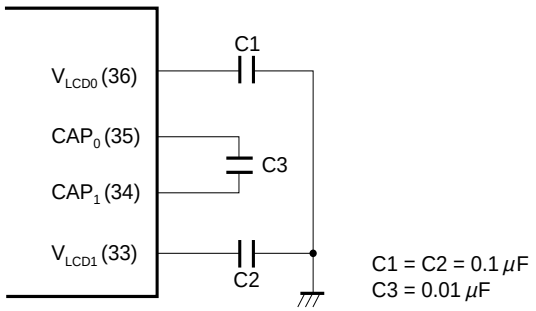
13.2 LCD DRIVING VOLTAGE GENERATION BLOCK

The LCD driving voltage generation block generates the voltage needed to drive the LCD.

An external doubler of the μPD17015 supplies the LCD drive voltage. To configure the doubler, connect capacitors to the V_{LCD0}, CAP_0, CAP_1, and V_{LCD1} pins.

★ Fig. 13-2 shows a sample doubler configuration. To use a voltage of 3.0 V (TYP.), connect capacitors as shown in Fig. 13-2.

★ **Fig. 13-2 Sample Doubler Configuration**



Remark Pin numbers are enclosed in parentheses.

Note that the LCD driving voltage (V_{LCD}) output by the doubler depends on the values of C1, C2, and C3.

13.3 LCD SEGMENT REGISTER

The LCD segment register sets the data indicating the LCD segments to be turned on or off.

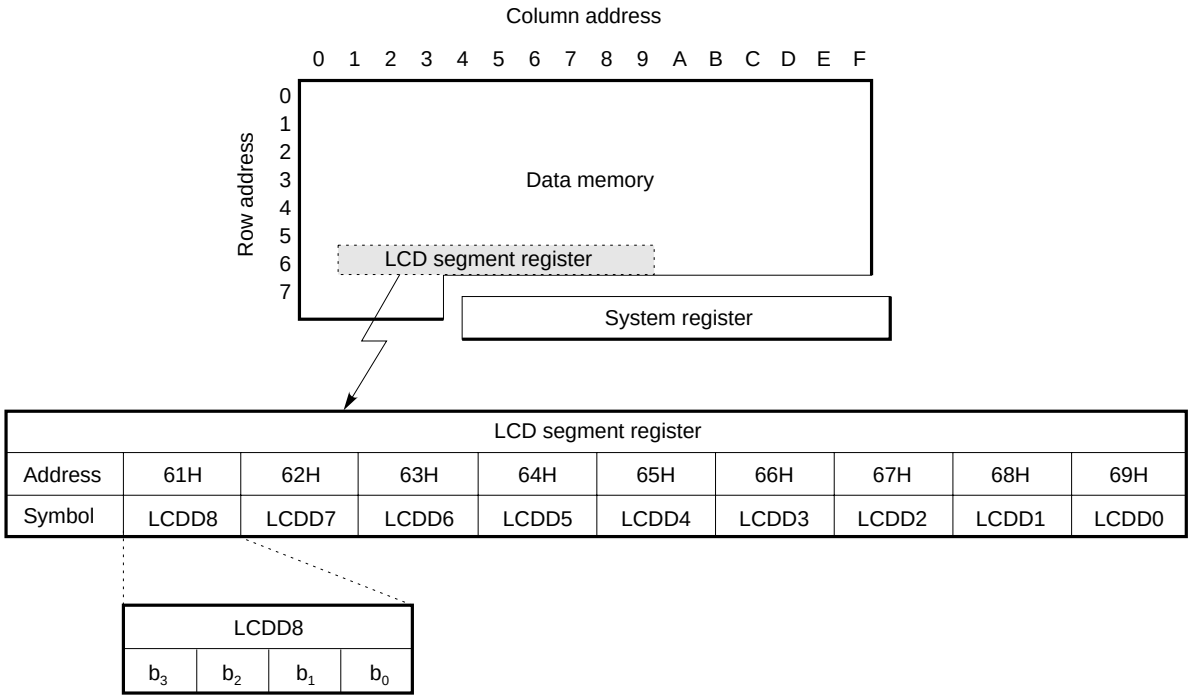
Fig. 13-3 shows the configuration and position of the LCD segment register in data memory.

The LCD segment register exists in data memory and can, therefore, be controlled by all data memory manipulation instructions.

One nibble of the LCD segment register sets the display data (turn-on/off data) for four segments. When a bit of the segment register is set to 1, the corresponding LCD segment is lit. When the bit is set to 0, the segment is not lit.

Fig. 13-4 indicates the relationship between the LCD segment register and LCD segments, together with display examples.

Fig. 13-3 Configuration of LCD Segment Register in Data Memory

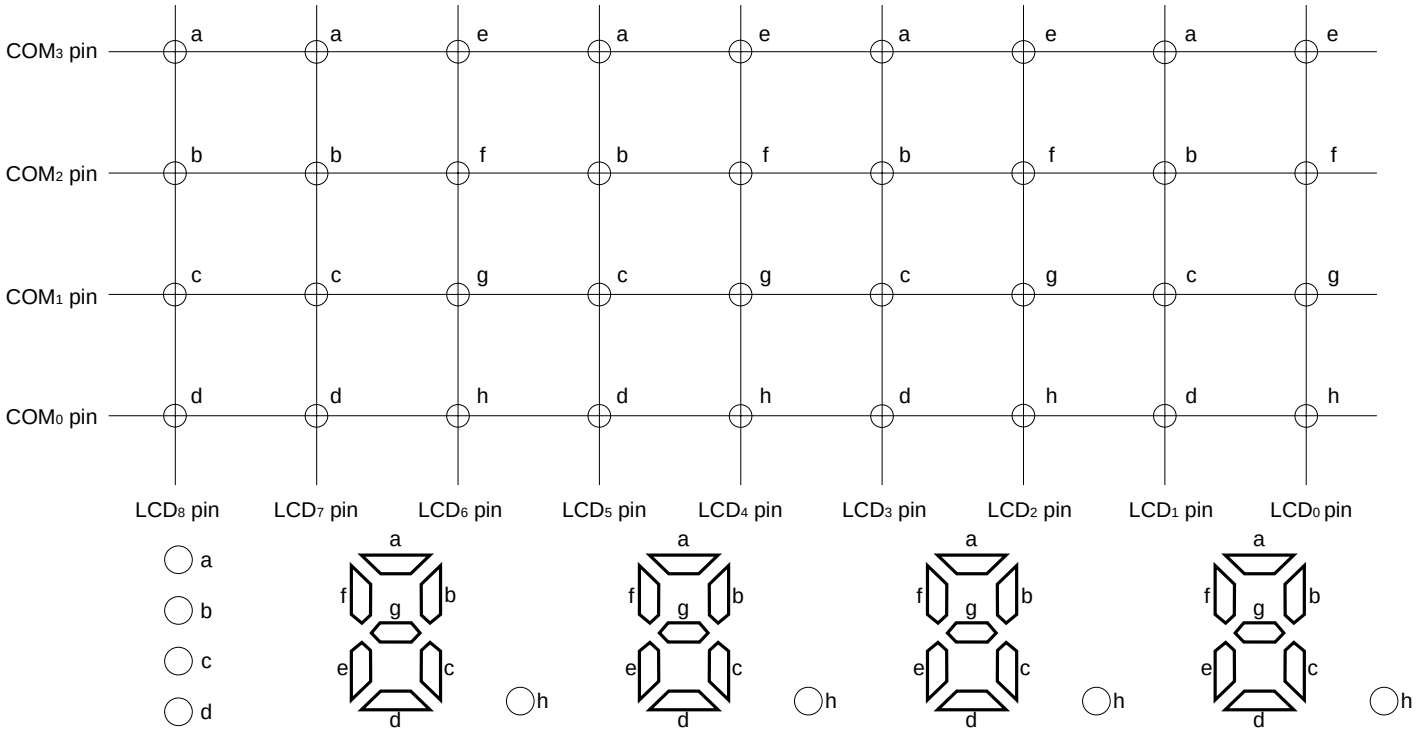


LCD segment register									
Address	61H	62H	63H	64H	65H	66H	67H	68H	69H
Symbol	LCDD8	LCDD7	LCDD6	LCDD5	LCDD4	LCDD3	LCDD2	LCDD1	LCDD0

LCDD8			
b_3	b_2	b_1	b_0

Fig. 13-4 Relationship Between LCD Segment Register and LCD Segments, and Examples of Display

LCD segment register																																	
Address	61H				62H				63H				64H				65H				66H				67H				68H				69H			
Symbol	LCDD8				LCDD7				LCDD6				LCDD5				LCDD4				LCDD3				LCDD2				LCDD1				LCDD0			
Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Display segment	a	b	c	d	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h



13.4 TIMING CONTROL BLOCKS FOR OUTPUTTING COMMON SIGNAL AND SEGMENT SIGNAL

Fig. 13-5 shows the configuration of the timing control blocks for outputting the common signal and segment signal.

The common signal output timing control block controls the timing of the common signal output on pins COM_0 to COM_3.

The segment signal output timing control block controls the timing of the segment signal output on pins LCD_0 to LCD_8.

The common signal and segment signal are output when the LCDEN flag of the LCD enable register is set to 1.

Setting the LCDEN flag to 0 turns off the entire LCD display. (See **Fig. 13-6**.)

When the LCD display is turned off, the signals output from pins COM_0 to COM_3 and pins LCD_0 to LCD_8 are held low.

Fig. 13-5 Configuration of Timing Control Blocks for Outputting Common Signal and Segment Signal

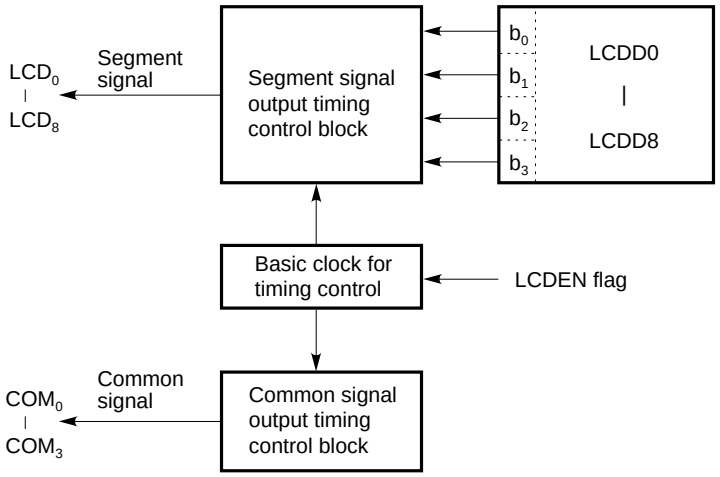


Fig. 13-6 Configuration and Functions of LCD Enable Register

Name	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
LCD enable register	LCDEN	BEEP0SEL	P0ABIO1	P0ABIO0	6AH	R/W

0	Turns the entire LCD display on or off.
1	Turns the display off. (The output from all the segment and common pins is held low.)
	Turns the display on.

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

13.5 COMMON SIGNAL AND SEGMENT SIGNAL OUTPUT WAVEFORMS

Fig. 13-7 shows sample waveforms for common signal and segment signal output.

★ The μPD17015 outputs signals at a frame frequency of 62.5 Hz when driving at a 1/4 duty cycle and 1/2 bias (voltage averaging method).

The common signals output from pins COM_0 to COM_3 have a relative phase difference of 1/8 and three voltage levels (V_{LCD0}, V_{LCD1}, and V_{DD}). The middle voltage level V_{LCD1} of the output common signals increases or decreases by 1/2 V_{DD}. This display method is referred to as the 1/2-bias driving method.

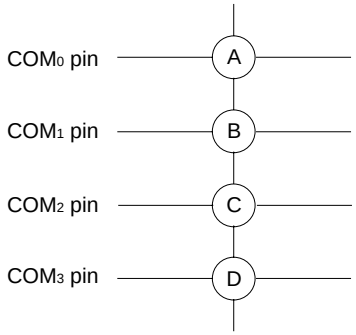
Each segment signal output pin outputs the segment signal with two voltage levels (0 and V_{DD}) and phases corresponding to the display segments. As shown in Fig. 13-7, a single segment pin is used for four display segments (A, B, C, and D). Each segment pin can output 16 different combinations of phases, corresponding to the turning on and off of the display segments.

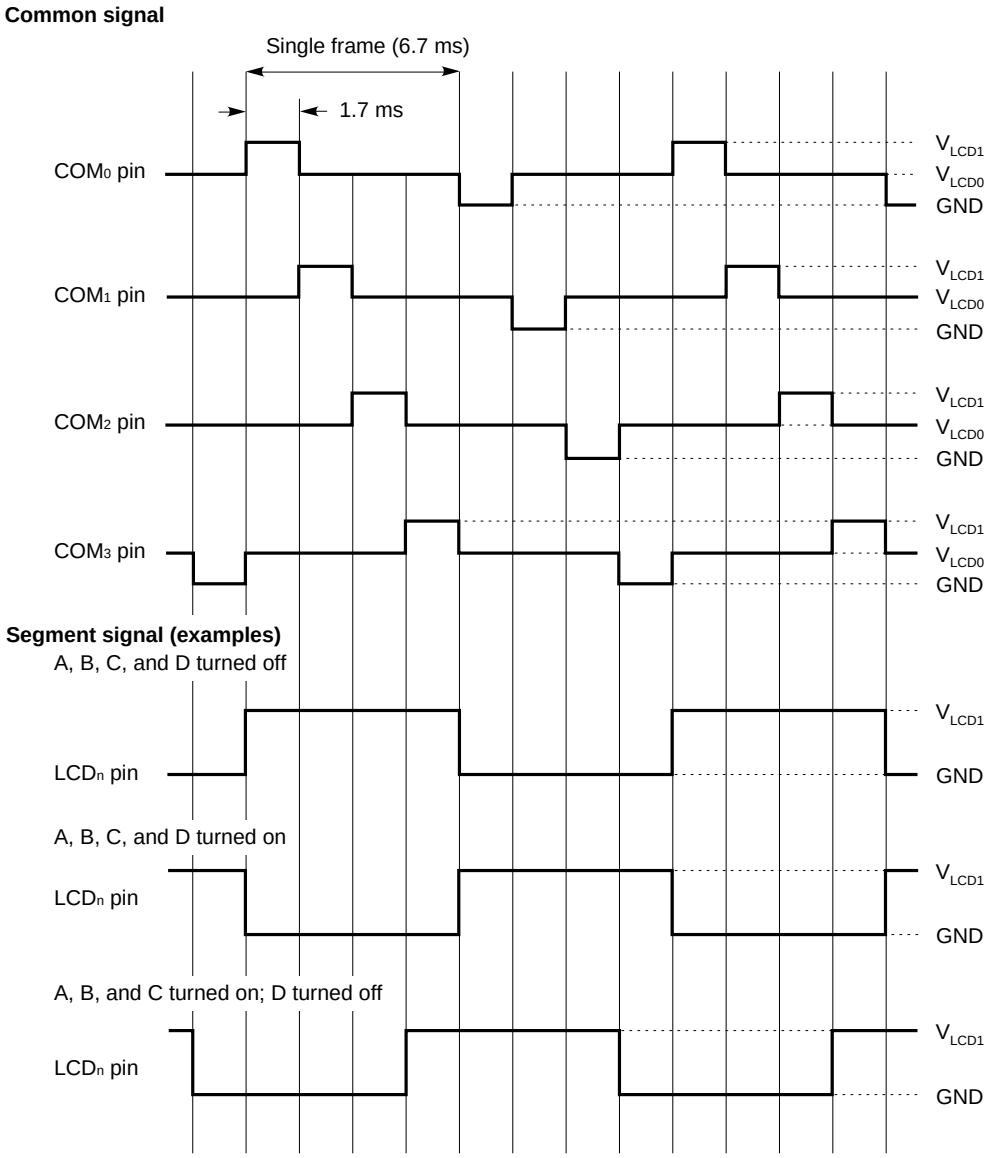
A display segment is lit when the voltage difference between the common signal and segment signal is V_{DD}.

★ The display segments are turned on in 1/4 duty cycles at a frequency of 62.5 Hz.

This is referred to as the 1/4-duty display method. The frequency is referred to as the frame frequency.

Fig. 13-7 Common Signal and Segment Signal Output Waveforms





13.6 USING THE LCD CONTROLLER/DRIVER

Fig. 13-8 shows an example of LCD panel connection.

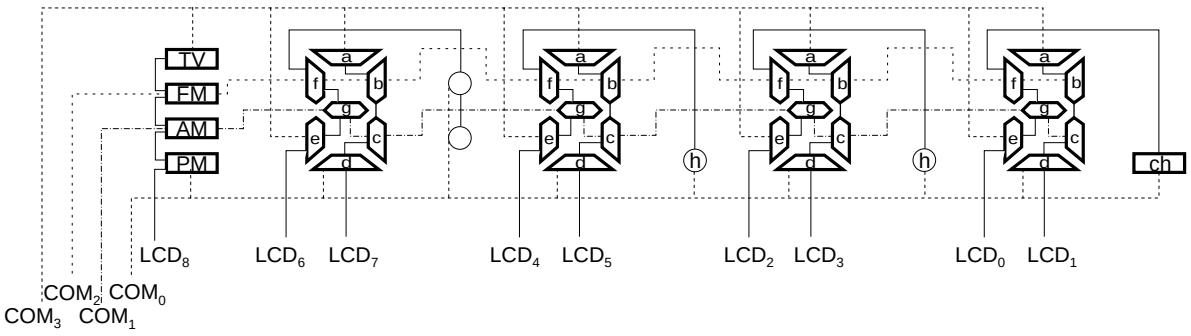
The following sample program turns on the 7-segment display with the LCD0 and LCD1 pins.

Example

```
        WORK    MEM     0.00H           ; Work area
        PMNO    MEM     0.01H           ; Storage area of preset memory number
        CH      FLG     LCDD0.1         ; Defines the low-order one bit of LCDD0 as the
                                          symbol CH display.
;LCDDATA:                               ; Display segment data
 ;                      abcdefg-
 ;      DW      0000000011111100B       ; 0FCH**0
 ;      DW      0000000001100000B       ;  60H**1
 ;      DW      0000000011011010B       ; 0DAH**2
 ;      DW      0000000011110010B       ; 0F2H**3
 ;      DW      0000000001100110B       ;  66H**4
 ;      DW      0000000010110110B       ; 0B6H**5
 ;      DW      0000000010111110B       ; 0BEH**6
 ;      DW      0000000011100100B       ; 0E4H**7
 ;      DW      0000000011111110B       ; 0FEH**8
 ;      DW      0000000011110110B       ; 0F6H**9
CNG_LCD_DAT:
                SKE     WORK0, #0
                BR      LCD1
                MOV     DBF0, #0CH
                MOV     DBF1, #0FH
        LCD1:
                SKE     WORK0, #1
                BR      LCD2
                MOV     DBF0, #0
                MOV     DBF1, #6
        LCD2:
                SKE     WORK0, #2
                BR      LCD3
                MOV     DBF0, #0AH
                MOV     DBF1, #0DH
        LCD3:
                SKE     WORK0, #3
                BR      LCD4
                MOV     DBF0, #2
                MOV     DBF1, #0FH
```

```
        LCD4:
                SKE     WORK0, #4
                BR      LCD5
                MOV     DBF0, #6
                MOV     DBF1, #6
        LCD5:
                SKE     WORK0, #5
                BR      LCD6
                MOV     DBF0, #6
                MOV     DBF1, #0BH
        LCD6:
                SKE     WORK0, #6
                BR      LCD7
                MOV     DBF0, #0EH
                MOV     DBF1, #0BH
        LCD7:
                SKE     WORK0, #7
                BR      LCD8
                MOV     DBF0, #4
                MOV     DBF1, #0EH
        LCD8:
                SKE     WORK0, #8
                BR      LCD9
                MOV     DBF0, #0EH
                MOV     DBF1, #0FH
        LCD9:
                SKE     WORK0, #9
                RET
                MOV     DBF0, #6
                MOV     DBF1, #0FH
                RET
  PRESET_CH:
                LD      WORK, PMNO
                CALL    CNG_LCD_DAT
                ST      LCDD0, DBF0
                ST      LCDD1, DBF1
                SET1    CH
                SET1    LCDEN           ; Display turned on
END
```

Fig. 13-8 Example of LCD Panel Connection



Relationship between the segment and common pins and the display segments on the LCD panel

Segment signal / Common signal		LCD8	LCD7	LCD6	LCD5	LCD4	LCD3	LCD2	LCD1	LCD0
	Pin No.	20	21	22	23	24	25	26	27	28
COM_3	29	TV	a	e	a	e	a	e	a	e
COM_2	30	FM	b	f	b	f	b	f	b	f
COM_1	31	AM	c	g	c	g	c	g	c	g
COM_0	32	PM	d	:	d	h	d	h	d	ch

13.7 STATE AT RESET

13.7.1 At Power-On Reset and Clock-Stop Reset

The signals output from the LCD_0 to LCD_8 pins go low. The signals output from pins COM_0 to COM_3 also go low. Consequently, the LCD display is turned off.

13.7.2 AT CE Reset and During the Halt State

Segment signals are output from pins LCD_0 to LCD_8. Common signals are output from pins COM_0 to COM_3.

14. STANDBY

The standby function is used to reduce the supply current during back-up.

14.1 STANDBY FUNCTIONS

Fig. 14-1 outlines the standby block.

The standby block reduces the device current drain by stopping some, or all, operations of the device. The standby block has the following three functions. These functions can be used to suit the application.

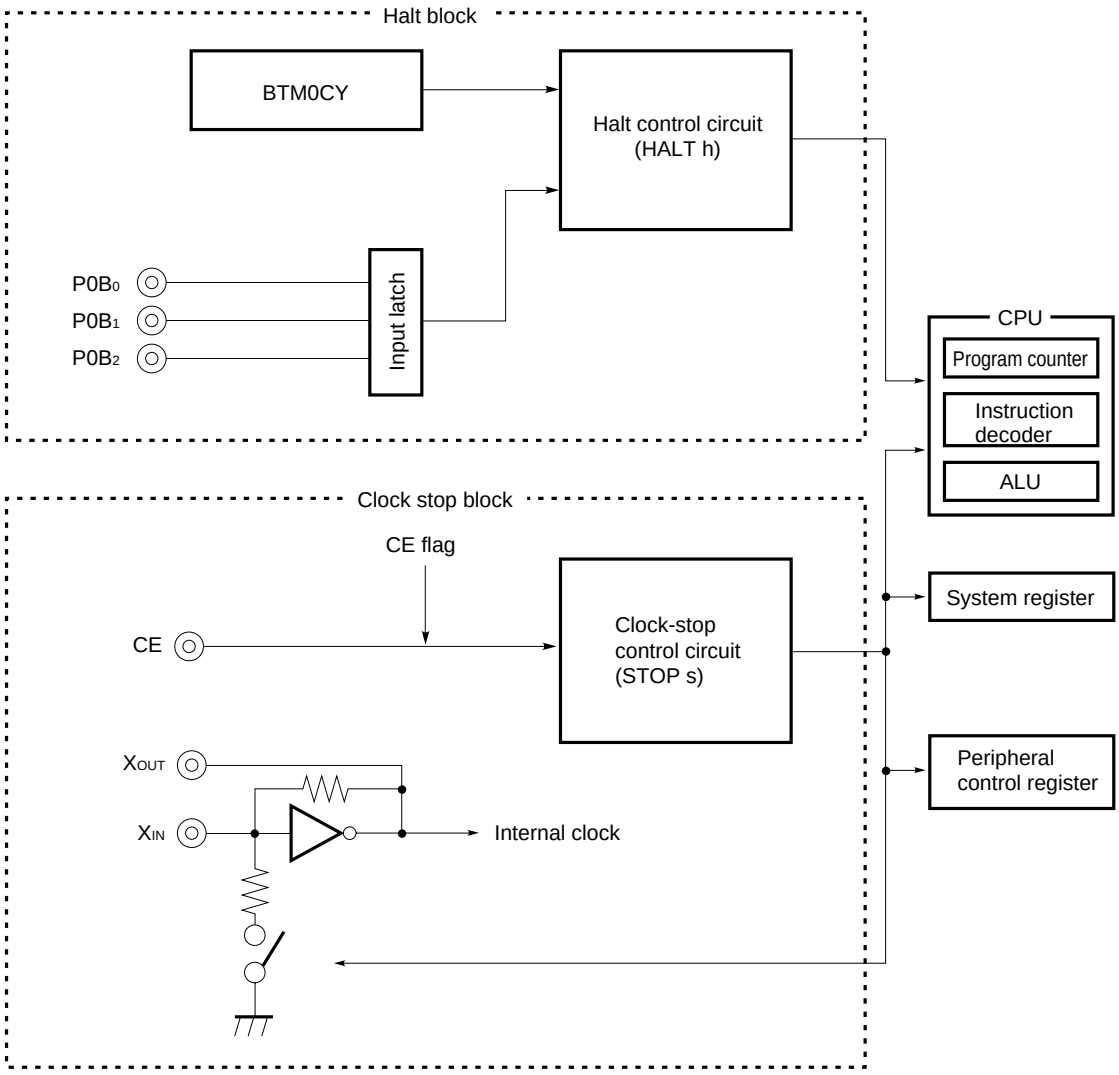
① Halt function
② Clock-stop function
③ Device operation control by CE pin

The halt function reduces the device current drain by stopping CPU operation with a “HALT h” instruction.

The clock-stop function reduces the device current drain by stopping the oscillation circuit with a “STOP s” instruction.

Since the CE pin is used to control operation of the image display controller (IDC) and PLL frequency synthesizer and to reset the device, its operation control function is said to be a standby function.

Fig. 14-1 Standby Block



Remark CE (bit 0 of CE pin level judge register: see **Fig. 14-8**).
Detects the status of the CE pin.

14.2 HALT FUNCTION

14.2.1 General

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

When a “HALT h” instruction is executed, the program halts and remains stopped until the halt state is released. In the halt state, the device current drain is reduced by the amount of the CPU operating current.

The halt state is released by key input or basic timer 0.

The release conditions are specified with the “h” operand of the HALT h instruction.

The “HALT h” instruction is valid regardless of the CE pin input level.

14.2.2 Halt State

In the halt state, all operations of the CPU are stopped. That is, the “HALT h” instruction stops program execution. However, the peripheral hardware remains in the state set before the “HALT h” is executed.

For an operation description of each hardware device, see **Section 14.4**.

14.2.3 Halt Release Conditions

Fig. 14-2 shows the halt release conditions.

The halt release conditions are set with the 4-bit data specified by the “h” operand of the “HALT h” instruction.

The halt state is released when the condition set to 1 at the “h” operand is satisfied.

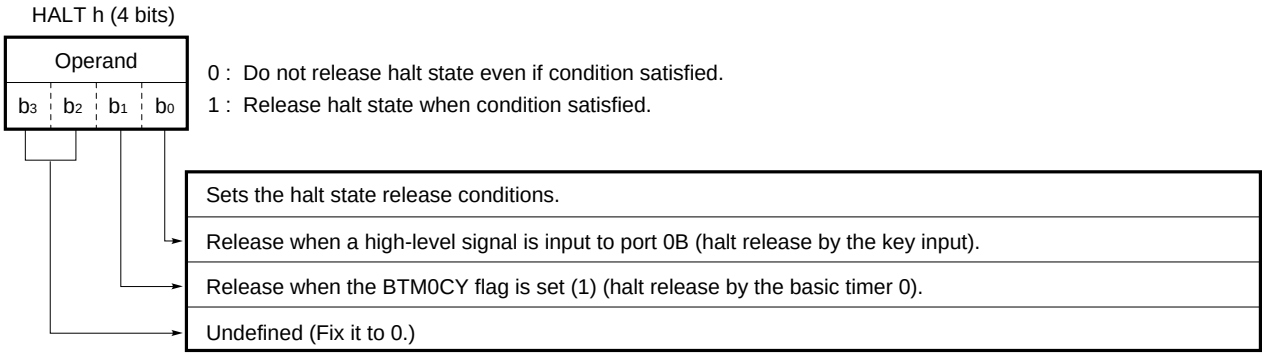
When the halt state is released, the program is executed from the instruction after the “HALT h” instruction.

When multiple release conditions are set, the halt state is released if even one of the set conditions is satisfied.

When reset (power-on reset or CE reset) is applied to the device, the halt state is released and the reset operations are performed.

When 0000B is set at halt release condition “h”, no halt condition is set. If reset (power-on reset or CE reset) is applied to the device at this time, the halt state is released.

Fig. 14-2 Halt Release Conditions



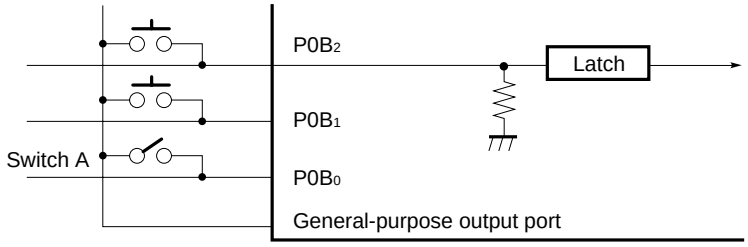
14.2.4 Halt Release by Key Input

Halt release by key input is set by “HALT 0001B” instruction.

When the halt release by key input is set, the halt state is released when high level is input at any one of the 0B port lines ($P0B_2$ to $P0B_0$ pins)

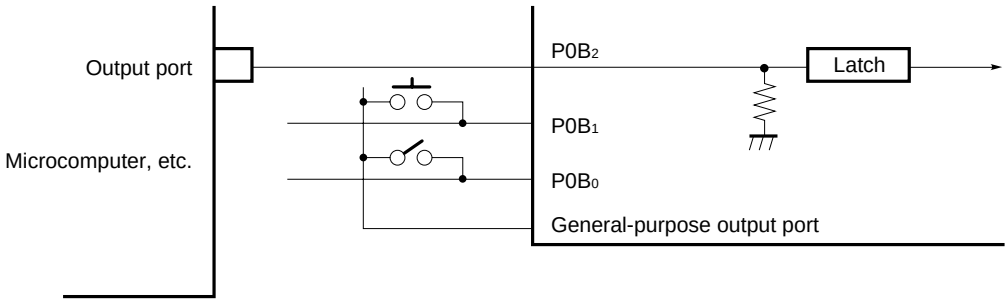
Each 0B port pin has a built-in pull-down resistor.

(1) When general-purpose output port is made key source



Execute a “HALT 0001B” instruction after the key source signal general-purpose output port is made high level. Note that if an alternate switch like switch A in the figure above is used, while switch A is closed, high level is applied to the $P0B_0$ pin and the halt state is immediately released.

(2) When halt released by other microcomputer, etc.



Halt can be released by another microcomputer, etc. as shown in the figure above.

14.2.5 Halt Release by Basic Timer 0

Halt release by basic timer 0 is set by the “HALT 0010B” instruction.

When halt release by basic timer 0 is set, the halt state is released simultaneously with setting (1) of the carry flip-flop of basic timer 0.

The carry flip-flop of basic timer 0 corresponds to the BTM0CY flag, and is set at intervals of 125 ms. The halt state can be released at a fixed cycle.

Example Program that releases the halt state every 125 ms and executes process A every second.

```
M1        MEM   0.10H          ; 1 second counter
HLTTMR    DAT   0010B          ; Symbol definition
LOOP:
          HALT  HLTTMR         ; Sets the condition of release caused by basic timer 0
                               ; and sets to the halt state.
          SKT1  BTM0CY         ; Built-in macro
          BR    LOOP           ; If BTM0CY flag is not set, branches to LOOP.
          ADD   M1, #0010B     ; Adds 0010B to contents of M1.
          SKT1  CY             ; Built-in macro
          BR    LOOP           ; If a carry is generated, executes process A.
          Process A
          BR    LOOP
```

14.2.6 When Two Release Conditions Set Simultaneously

When two halt release conditions are set, the halt state is released if even one of the set release conditions is satisfied.

The following example indicates how to judge two release conditions when they are satisfied simultaneously.

Example Judging between two release conditions when they are satisfied simultaneously

```
            HLTBTMR   DAT   0010B
            HLTKEY    DAT   0001B
START:
            SET3      P0C2, P0C1, P0C0     ; Outputs key source signal.
BTMRUP:
            Process A
            RET
KEYDEC:
            Process B
            RET
LOOP:
            HALT      HLTBTMR OR HLTKEY    ; Sets halt release conditions to basic
                                           ; timer 0 or key input.
            SKF1      BTM0CY               ; Detects BTM0CY flag
            CALL      BTMRUP               ; If set, executes timer carry processing.
            SKF3      P0B2, P0B1, P0B0     ; Key input latch processing
            CALL      KEYDEC               ; If latched, executes key input processing
            BR        LOOP
```

14.3 CLOCK-STOP FUNCTION

The clock-stop function stops the 75-kHz crystal oscillation circuit (clock stopped state) by executing a “STOP s” instruction.

The supply current is reduced by up to 3 μA.

Specify “0000B” at operand “s” of the “STOP s” instruction.

The “STOP s” instruction is valid only when the CE pin is low level. If a “STOP s” instruction is executed while the CE pin is high level, it is executed as a “NOP” instruction. Always execute a “STOP s” instruction when the CE pin is low level.

The clock-stop state is released by raising the CE pin from low level to high level (CE reset).

14.3.1 Clock-Stop State

Since the crystal oscillation circuit is stopped in the clock-stop state, operation of the CPU, peripheral hardware, and other devices is stopped.

For a description of operation of the CPU and each item of peripheral hardware, see **Section 14.4**.

14.3.2 Clock-Stop State Release

The clock-stop state can be released with the two methods described below. For both methods, after the clock-stop state is released, the program starts from address 0000H.

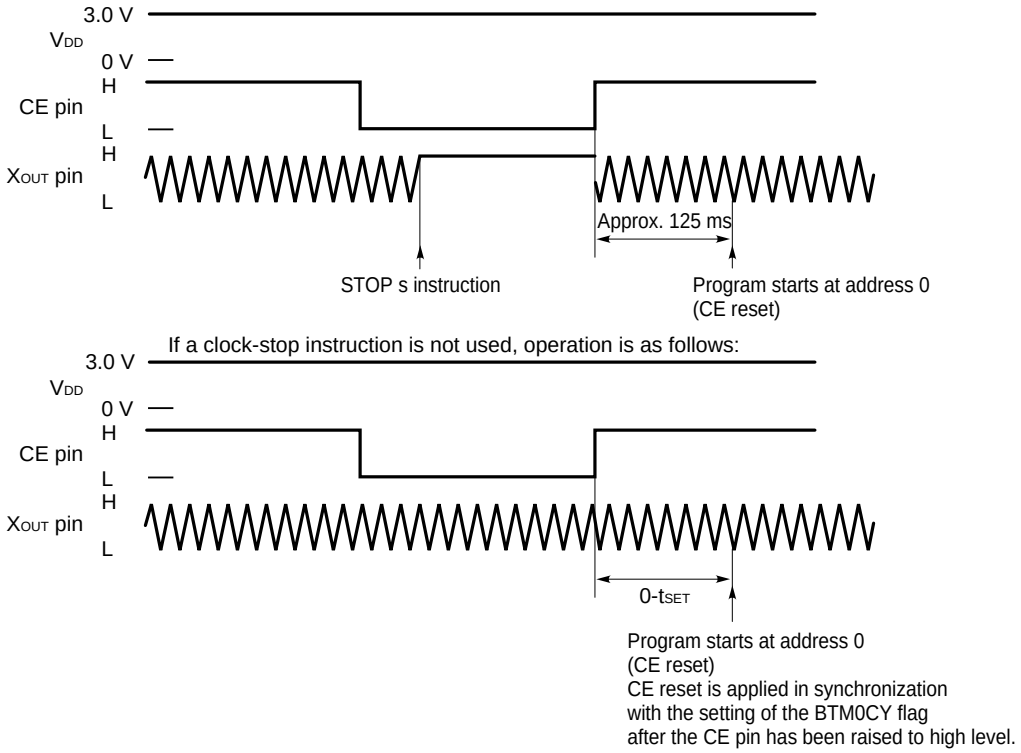
① Raising the CE pin from low level to high level (CE reset)
② Dropping V_{DD} to 1.8 V or less, then raising it to 1.8 V or more (T_A = –10 to +50 °C).

Caution The power-on clear voltage varies with the CPU operating temperature. For details, see Section 15.2.

14.3.3 Clock-Stop Release by CE Reset

Fig. 14-3 shows the clock-stop release by CE reset.

Fig. 14-3 Clock-Stop Release by CE Reset

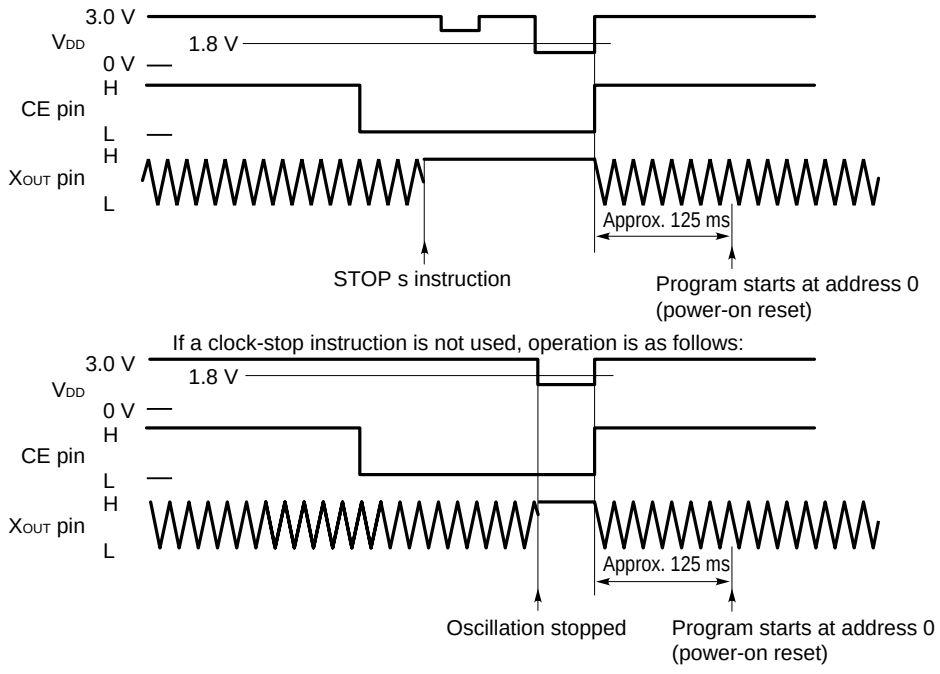


14.3.4 Clock-Stop Release by Power-On Reset

Fig. 14-4 shows the clock-stop release by power-on reset.

If the clock-stop state is released by power-on reset, the power failure detection circuit operates.

Fig. 14-4 Clock-Stop Release by Power-On Reset (T_A = –10 to +50 °C)



14.3.5 Cautions When Using Clock-Stop Instruction

The clock-stop instruction (STOP s instruction) is valid only when the CE pin is low level.

The program must take into account processing when the CE pin is raised unexpectedly to high level.

The description is based on the following example.

Example

```
        XTAL    DAT     0000B       ; Clock-stop condition symbol definition
CEJDG:
  ; ①
        SKF1    CE                  ; Built-in Macro
                                    ; Detects the CE pin input level.
        BR      MAIN                ; If CE = high level, branches to main processing.
        Process A                   ; CE = low level processing
  ; ②
        STOP    XTAL                ; Clock-stop
  ; ③
        BR      $-1
MAIN:
        Main processing
        BR      CEJDG
```

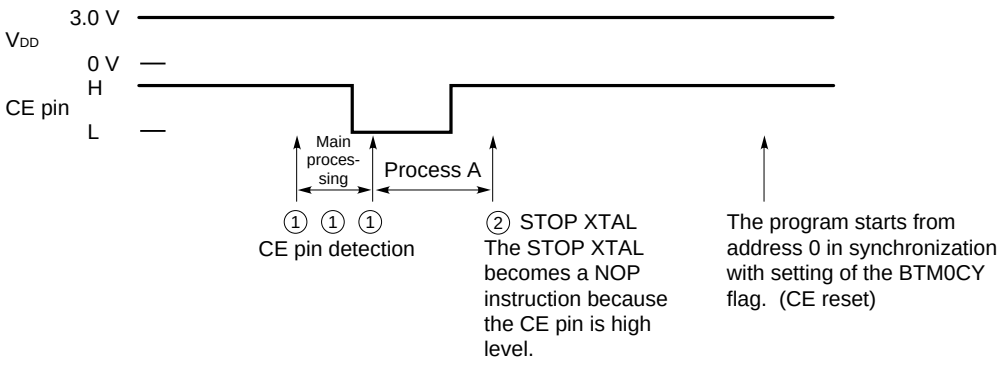
In the example above, the state of the CE pin is detected at ①. If the CE pin is low level, after process A is executed, the clock-stop instruction "STOP XTAL" of ② is executed.

However, if the CE pin becomes high level while the "STOP XTAL" instruction of ② is being executed as shown in the figure below, the "STOP XTAL" instruction operates as a no operation instruction (NOP).

If the branch instruction "BR $-1" of ③ does not exit, the program returns to main processing and erroneous operational occurs.

Therefore, a branch instruction like ③ must be inserted in the program, or the program must be written so that erroneous operational does not occur even if it returns to main processing.

When a branch instruction like ③ is used, CE reset is applied in synchronization with the next setting of the BTM0CY flag, even if the CE pin remains at high level.



14.4 DEVICE OPERATION IN THE HALT AND CLOCK-STOP STATES

Table 14-1 shows the operation of the CPU and peripheral hardware in the halt state and clock-stop state.

In the halt state, all the peripheral hardware units continue to operate normally except that they stop executing instructions.

In the clock-stop state, all the peripheral hardware units stop operating.

In the halt state, the peripheral control register that controls the operating state of the peripheral hardware operates normally (not initialized). However, when a clock-stop instruction is executed, it is initialized to the specified value.

In short, in the halt state, the operation set in the peripheral control register continues and in the clock-stop state, the operating state is determined in accordance with the initialized peripheral control register value.

For the peripheral control register value in the clock-stop state, see the relevant items.

A sample program is shown below.

Example Program that sets the $P0A_1$ and $P0A_0$ pins as output ports and which performs LCD display

```
            HLTBTMR   DAT    0001B          ; Symbol definition
            XTAL      DAT    0000B
            INITFLG   P0ABIO1, P0ABIO0
  ; ①
            SET2      P0A1, P0A0
LCDDATA :
  ; ②
            Process B
  ; ③
            SET1      LCDEN                 ; LCD display on
  ; ④
            HALT      HLTKEY
  ; ⑤
            STOP      XTAL
```

In the example, ① outputs high level from the $P0A_1$ and $P0A_0$ pins, ② sets the display data for the LCD segment register, and ③ turns the LCD display on.

When the HALT instruction is executed at ④ , LCD display continues turning on, and the halt state is released when a key input is accepted.

If the STOP instruction of (is executed instead of the HALT instruction of ④ , all the flags of the peripheral control register set at ①, ② and ③ are initialized. LCD display is terminated and $P0A_1$ and $P0A_0$ pins are made general-purpose input ports.

Table 14-1 Device Operation in Halt State and Clock-Stop State

Peripheral hardware	State			
	CE pin: High level		CE pin: Low level	
	At halt	At clock-stop	At halt	At clock-stop
Program counter	Stopped at HALT instruction address.	STOP instruction invalid (NOP)	Stopped at HALT instruction address.	Initialized to 0000H and stopped.
System register	Held		Held	Initialized[Note 1]
Peripheral hardware register				Held
Peripheral control register				Initialized[Note 2]
Basic timer	Normal operation		Normal operation	Operation stopped
PLL frequency synthesizer			Disabled	
BEEP			Normal operation	
LCD controller/driver				
General-purpose I/O port				Input port
General-purpose input port				Input port
General-purpose output port				Held

Notes 1. For the value that is initialized, see **Chapter 5**.

2. For the value that is initialized, see each section.

14.5 PIN PROCESSING CAUTIONS IN HALT STATE AND CLOCK-STOP STATE

The halt state is used to reduce the supply current when only the clock is operating.

The clock-stop function is used to reduce the supply current for holding only the data memory.

Consequently, the supply current must be reduced as much as possible in the halt and clock-stop states.

The supply current depends on the state of each pin and the cautions shown in Table 14-2 must be observed.

Table 14-2 State of Each Pin and Cautions in Halt and Clock-Stop States

Pin function		Pin symbol	State of each pin and processing precautions	
			Halt state	Clock-stop state
I/O port	Port 0A	$P0A_0$ $P0A_1$	The state before halt is held. **(1) When set as output pins** If externally pulled down while a high-level signal is being output, or if externally pulled up while a low-level signal is being output, the current drain increases. **(2) When set as input pins (other than port 0B)** When floating, noise, etc. increase the current drain. **(3) Port 0B ($P0B_0$ to $P0B_2$)** Since a pull-down resistor is built in, externally pulling the pin up causes the current drain to increase.	Pins are specified as general-purpose input ports. Pins $P0A_0$ and $P0A_1$ must be externally pulled down or pulled up to prevent the current drain from increasing. Port 0B ($P0B_0$ to $P0B_2$) is internally pulled down.
Input port	Port 0B	$P0B_0$ $P0B_1$ $P0B_2$		
Output port	Port 0A	$P0A_2$ $P0A_3$		Pins are specified as general-purpose input ports. The output contents are held. If externally pulled down while a high-level signal is being output, or if externally pulled up while a low-level signal is being output, the current drain increases.
	Port 0C	$P0C_0$ $P0C_1$ $P0C_2$ $P0C_3$		
	Port 0D	$P0D_0$/BEEP		
LCD segment		LCD_0 | LCD_8	The state existing prior to halt is held.	Low level output (display off)
PLL frequency synthesizer		VCOL VCOH EO	At PLL operation, the current drain increases. In the PLL disabled state, all pins enter floating status. When the level of the CE pin goes low, the PLL is automatically disabled.	PLL disabled state. All pins enter the floating state.
Crystal oscillation circuit		X_{IN} X_{OUT}	The supply current changes with the oscillation waveform of the crystal oscillation circuit. The larger the oscillation amplitude, the lower the supply current. Since the oscillation amplitude is governed by the crystal and load capacitor used, evaluation is necessary.	The X_{IN} pin is pulled down internally and the X_{OUT} pin outputs high level.

14.6 DEVICE OPERATION CONTROL BY THE CE PIN

The CE pin controls the following functions by means of the input level and rising edge of a signal received from the outside:

(1) PLL frequency synthesizer
(2) Clock-stop instruction disable/enable
(3) Device reset

14.6.1 PLL Frequency Synthesizer Operation Control

The PLL frequency synthesizer can operate only when the CE pin is high level.
When the level of the CE pin goes low, the PLL is automatically disabled.
In PLL disabled state, the VCOH, VCOL, or EO pin enters the floating state.
The PLL frequency synthesizer can be disabled by program even when the CE pin is high level.

14.6.2 Clock-Stop Instruction Disable/Enable Control

The clock-stop instruction (“STOP s” instruction) is valid only when the CE pin is low level.
If the CE pin is high level, the clock-stop instruction is executed as a no operation instruction (NOP).

14.6.3 Device Reset

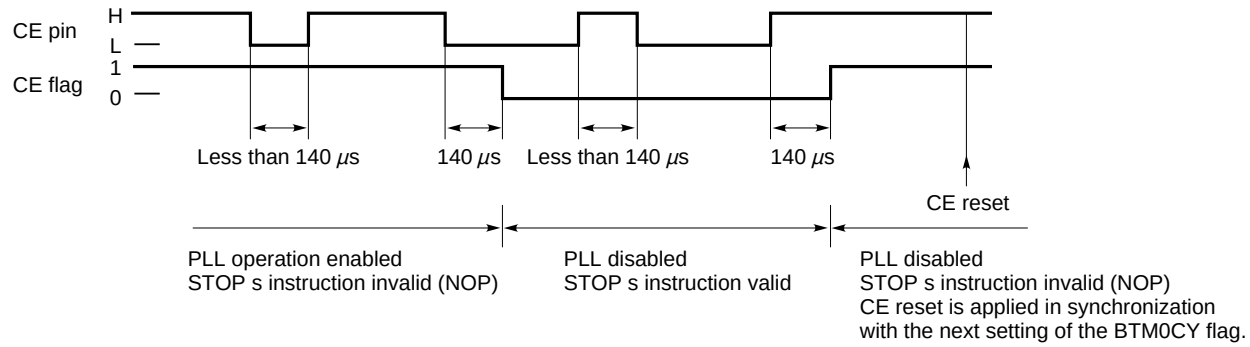
Reset (CE reset) can be applied to the device by raising the CE pin from low level to high level.
Besides CE reset, there is also power-on reset, which is activated when V_{DD} is turned on.
For details, see **Chapter 15**.

14.6.4 Signal Input to CE Pin

To prevent erroneous operation by noise, the CE pin does not accept signals with a low or high level width of less than 140 μs. The level of the signal input to the CE pin can be detected with the CE flag of the CE pin level judge register.

Fig. 14-5 shows the relationship between input signal and CE flag.

Fig. 14-5 Relationship of Signal Input to CE Pin and CE Flag



14.6.5 Organization and Functions of CE Pin Level Judge Register

The CE pin level judge register monitors the CE pin input signal level.

Its organization and functions are shown below.

Fig. 14-6 Configuration of CE Pin Level Judge Register

Register	Flag symbol				Address	Read/write
	b_3	b_2	b_1	b_0		
CE pin level judge register	0	0	0	CE	6AH	R

	Detects the level input at the CE pin.
0	Low level input
1	High level input

Fixed to 0.

Upon reset	Power-on	0	0	0	–
	Clock stop	↓	↓	↓	–
	CE	↓	↓	↓	–

The CE flag also does not change when the CE pin receives signals having a low or high level width of less than 140 μs.

15. RESET

The reset function is used to initialize device operation.

15.1 OUTLINE OF RESET FUNCTION

Fig. 15-1 outlines the reset block.

To reset a device, apply one of the following methods, as necessary.

① Power-on reset : Reset by applying supply voltage V_{DD}
② CE reset : Reset with the CE pin

Power-on reset is applied when V_{DD} rises from a certain voltage.

CE reset is applied when the level of the CE pin goes from low to high.

The main differences between power-on reset and CE reset are the initial values of the peripheral control register and the operation of the power failure detection circuit, as described in **Section 15.5**.

Power-on reset and CE reset are controlled by the reset signals ($\overline{IRES}$, $\overline{RES}$, or $\overline{RESET}$ signal) output from the reset control circuit shown in Fig. 15-1.

Table 15-1 indicates the relationship between the internal reset signal and each reset.

Fig. 15-1 Outline of the Reset Block

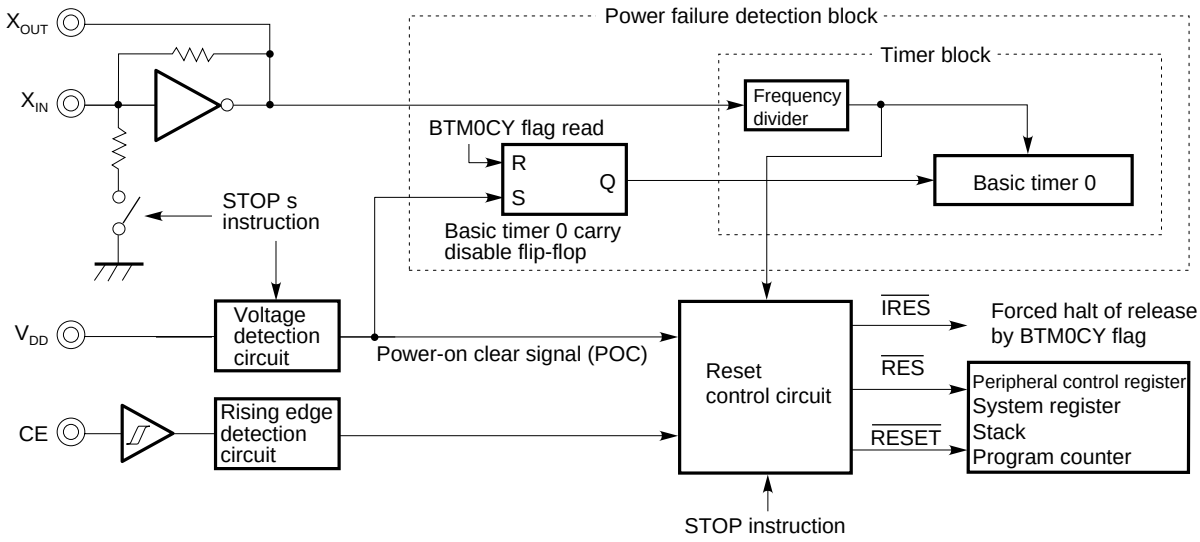


Table 15-1 Relationship Between Internal Reset Signal and Each Reset

Internal reset signal	Output signal			Contents controlled by each reset signal
	At CE reset	At power-on reset	At clock-stop	
$\overline{IRES}$	×	○	○	Forces the device into the halt state. The halt state is released by the setting of the BTM0CY flag.
$\overline{RES}$	×	○	○	Initializes some peripheral control registers.
$\overline{RESET}$	○	○	○	Initializes the program counter, stack, system register, and some peripheral control registers.

15.2 POWER-ON RESET

Power-on reset is executed by raising V_{DD} from a certain voltage (called the power-on clear voltage) or less.

When V_{DD} is less than the power-on clear voltage, the power-on clear signal (POC) is output from the voltage detection circuit shown in Fig. 15-1.

When the power-on clear signal is output, the crystal oscillation circuit stops and the device stops operating.

While the power-on clear signal is being output, the $\overline{IRES}$, $\overline{RES}$ and $\overline{RESET}$ signals are output.

When V_{DD} exceeds the power-on clear voltage, the power-on clear signal is dropped and crystal oscillation starts. At the same time, the $\overline{IRES}$, $\overline{RES}$ and $\overline{RESET}$ signals are also dropped.

Since the $\overline{IRES}$ signal halts release by the BTM0CY flag, power-on reset is applied at the rising edge of the next BTM0CY flag setting signal.

This operation is shown in Fig. 15-2.

At power-on reset, the program counter, stack, system register and control registers are initialized when the power-on clear signal is output. For the initial values, see the relevant items.

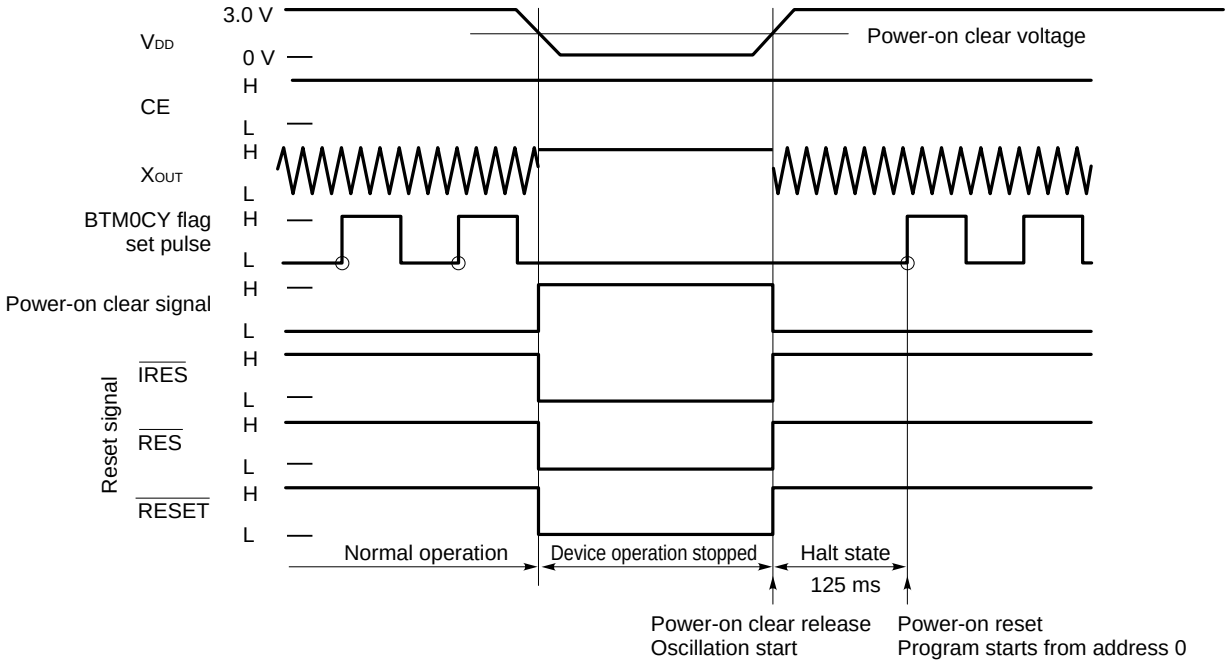
In the CPU operation and clock-stop state, the power-on clear voltage is 1.8 V (rated value).

Caution The power-on clear voltage varies with the CPU operating temperature, as follows:

T_A = –10 to +50 °C: 1.8 V

T_A = –20 to +50 °C: 1.9 V

Fig. 15-2 Power-On Reset Operation



15.2.1 Power-On Reset at CPU Operation

Fig. 15-3 (a) shows power-on reset at normal operation.

As shown in Fig. 15-3 (a), when the V_{DD} drops below 1.8 V (T_A = –10 to +50 °C), the power-on clear signal is output and operation of the device stops regardless of the input level of the CE pin.

When V_{DD} then rises to 1.8 V or greater, after a 125 ms halt, the program starts from address 0000H.

CPU operation refers to the state in which the clock-stop instruction is not used. The power-on clear voltage in the halt state set by the halt instruction is also 1.8 V like CPU operation.

15.2.2 Power-On Reset in Clock-Stop State

Fig. 15-3 (b) shows power-on reset in the clock-stop state.

As shown in Fig. 15-3 (b), when V_{DD} drops below 1.8 V (T_A = –10 to +50 °C), the power-on clear signal is output and device operation stops.

However, since the device is in the clock-stop state, its operation apparently does not change.

When V_{DD} rises to 1.8 V or greater, after a 125 ms halt, the program starts from address 0000H.

15.2.3 Power-On Reset When V_{DD} Rises From 0 V

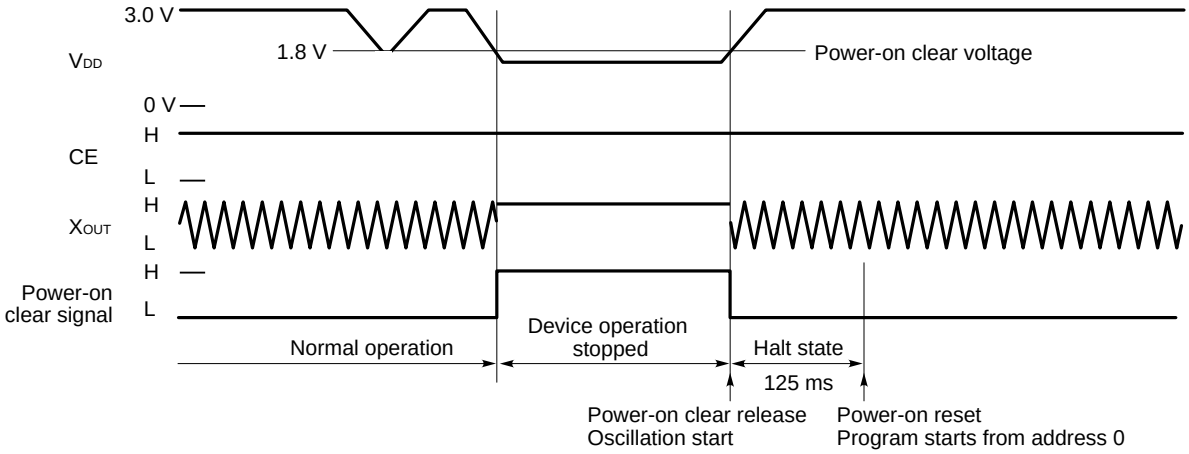
Fig. 15-3 (c) shows power-on reset when V_{DD} rises from 0 V.

As shown in Fig. 15-3 (c), the power-on clear signal is being output while V_{DD} is rising from 0 V to 1.8 V (T_A = –10 to +50 °C).

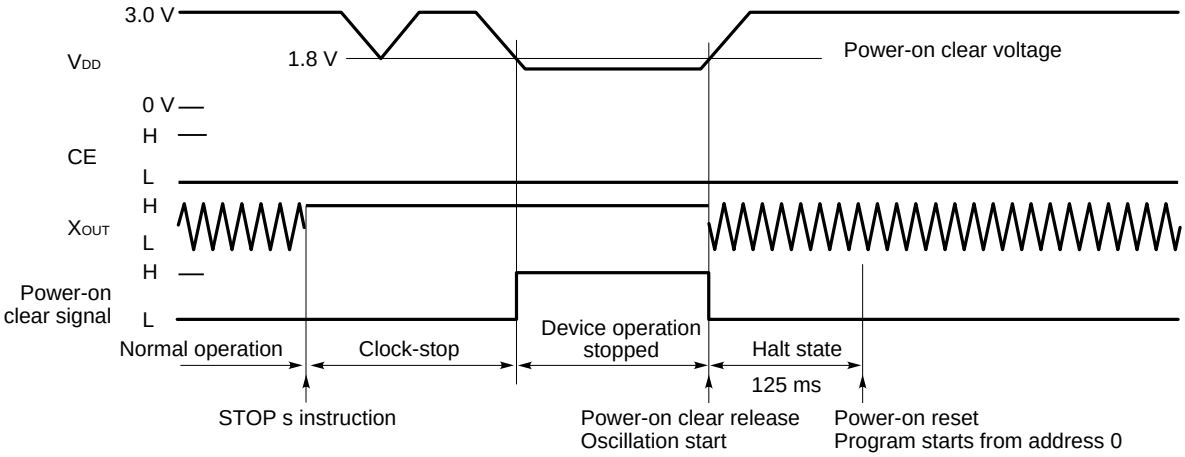
When V_{DD} rises above the power-on clear voltage, the crystal oscillation circuit starts and after a 125 ms halt, the program starts from address 0000H.

Fig. 15-3 Power-On Reset and V_{DD} (T_A = –10 to +50 °C)

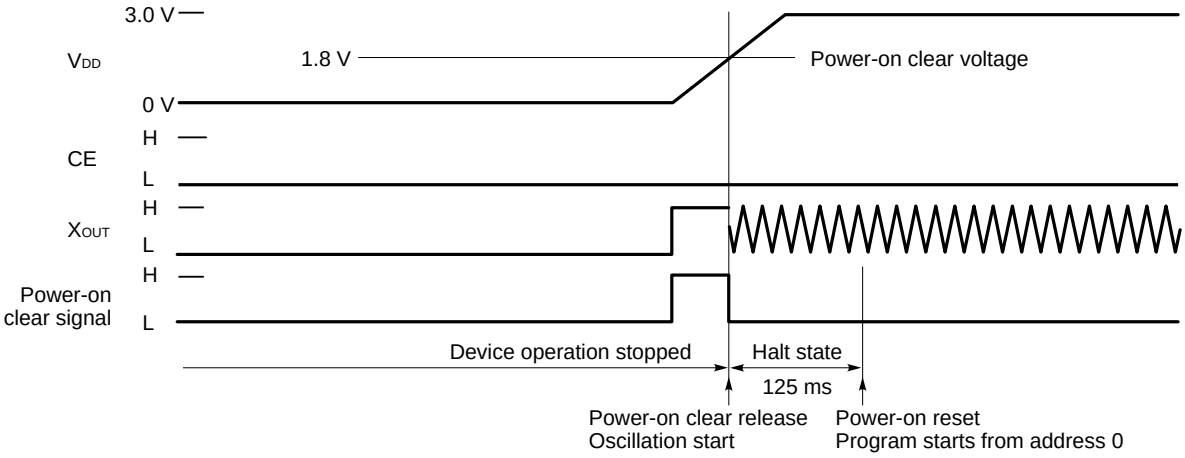
(a) During CPU operation (also in the halt state)



(b) At clock-stop



(c) When V_{DD} rises from 0 V



15.3 CE RESET

CE reset is executed by raising the CE pin from low level to high level.

When the CE pin rises to high level, the $\overline{\text{RESET}}$ signal is output and the device is reset in synchronization with the rising edge of the pulse used for the next setting of the BTM0CY flag.

When CE reset is applied, the $\overline{\text{RESET}}$ signal initializes the program counter, stack, system register, and some peripheral control registers to their initial value and executes the program from address 0000H.

For the initial values, see the relevant item.

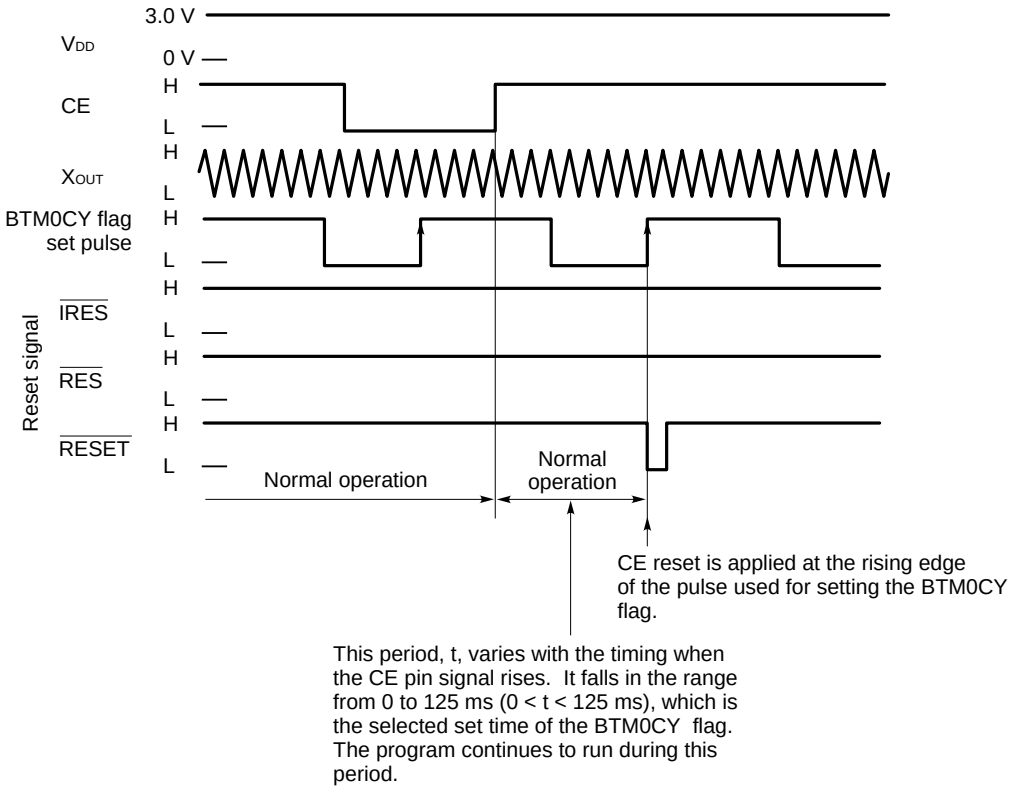
CE reset operation is different when clock-stop is used and when it is not used.

15.3.1 CE Reset When Clock-Stop (STOP s Instruction) Not Used

Fig. 15-4 shows the reset operation when clock-stop is not used.

When clock-stop (STOP s instruction) is not used, the CE pin becomes high level. Then, $\overline{\text{RESET}}$ signal is output, and reset is applied at the rising edge of the selected pulse (125 ms) used for setting the BTM0CY flag.

Fig. 15-4 CE Reset Operation When Clock-Stop Not Used



15.3.2 CE Reset When Clock-Stop (STOP s Instruction) Used

Fig. 15-5 shows the reset operation when clock-stop is used.

When clock-stop is used, the $\overline{\text{IRES}}$, $\overline{\text{RES}}$ and $\overline{\text{RESET}}$ signals are output at the time the “STOP s” instruction is executed.

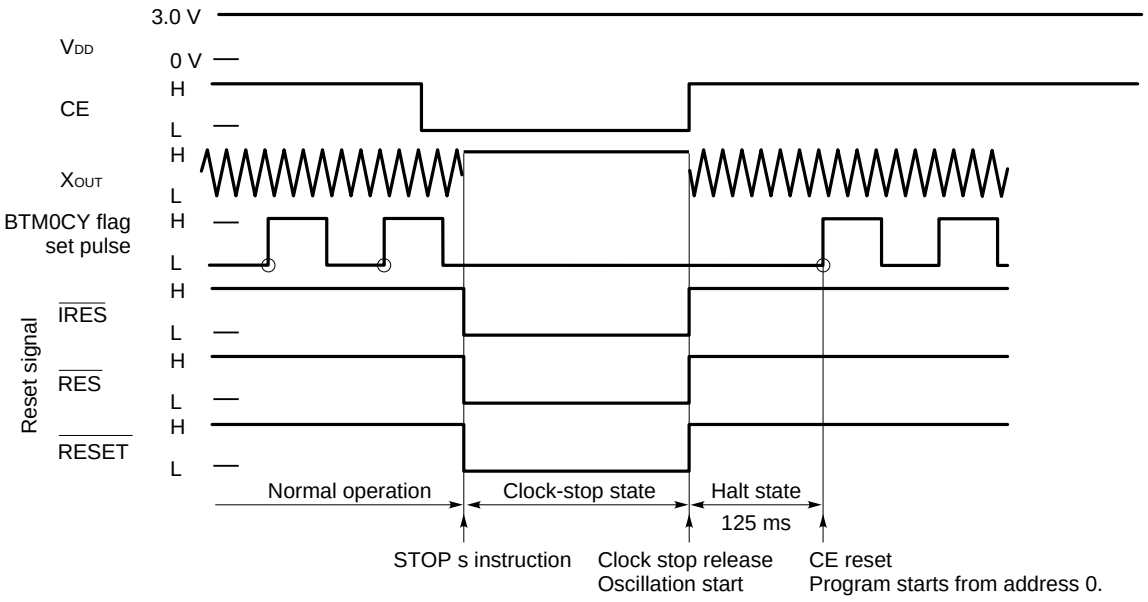
Since the $\overline{\text{IRES}}$ signal is output continuously while the CE pin is low level, release by the BTM0CY flag is forcibly halted.

Since the clock itself stops, the device stops operating.

When the CE pin rises to high level, the clock-stop state is released and oscillation begins.

The $\overline{\text{IRES}}$ signal halts release by the BTM0CY flag. When the pulse used for setting the BTM0CY flag rises after the CE pin rises, the halt state is released and the program starts from address 0.

Fig. 15-5 CE Reset Operation When Clock-Stop Used



15.3.3 Cautions at CE Reset

When CE reset is used, careful attention must be given to points (1) and (2) below regardless of the instruction being executed.

(1) Time required for clock and other timer processing

When writing a clock program by using basic timer 0 and basic timer 1 interrupts, the program must end processing within a certain time.

For details, see **Section 10.2.5**.

(2) Processing of data, flags, etc. used in the program

Care must be exercised when rewriting the contents of data, flags, etc. that cannot be processed by one instruction so that the contents do not change even when CE reset is applied.

Examples are given below:

Example 1

```
R1          MEM     0.01H           ; First digit of security code keyed in
R2          MEM     0.02H           ; Second digit of security code keyed in
R3          MEM     0.03H           ; Data of the first digit when the security code is
                                      changed
R4          MEM     0.04H           ; Data of the second digit when the security code is
                                      changed
M1          MEM     0.11H           ; First digit of the current security code
M2          MEM     0.12H           ; Second digit of the current security code

START:
            Key-in processing       ; Security code input wait mode
             R1 ← Contents of key A ; Assigns the contents of the pressed keys to R1
             R2 ← Contents of key B ; and R2
            SET2    CMP, Z    ; ①   ; Compares the input data with the security code.
            SUB     R1, M1
            SUB     R2, M2
            SKT1    Z
            BR      ERROR           ; Detects that the input data does not match the
                                    ; security code.
MAIN:
            Key-in processing       ; Security code rewrite mode
             R3 ← Contents of key C ; Assigns the contents of the pressed keys to R3
             R4 ← Contents of key D ; and R4.
            ST      M1, R3    ; ②   ; Rewrites the security code.
            ST      M2, R4    ; ③
ERROR:      BR      MAIN
            Stops the operation.
```

Suppose, in the program in example 1, that the security code is 12H. The contents of data memory M1 and M2 are 1H and 2H, respectively.

When a CE reset occurs, the key input is compared with security code 12H in ①. If they match, processing is performed.

If the security code is changed as part of main processing, the new code is written to M1 and M2 in ② and ③.

If the security code is changed to 34H, for example, 3H and 4H are written to M1 and M2 in ② and ③.

If a CE reset occurs after ②, the program starts from 0000H, without executing ③.

The security code is actually changed to 32H, making it impossible to remove the security lock.

If this problem occurs, use a program like that shown in example 2 below:

Example 2

```
R1          MEM     0.01H           ; First digit of security code keyed in
R2          MEM     0.02H           ; Second digit of security code keyed in
R3          MEM     0.03H           ; Data of the first digit when the security code is
                                    ; changed
R4          MEM     0.04H           ; Data of the second digit when the security code is
                                    ; changed
M1          MEM     0.11H           ; First digit of the current security code
M2          MEM     0.12H           ; Second digit of the current security code
CHANGE      FLG     0.13H.0         ; The flag is set to 1 while the security code is being
                                    ; changed.

START:
            Key-in processing       ; Security code input wait mode
             R1 ← Contents of key A ; Assigns the contents of the pressed keys to R1
             R2 ← Contents of key B ; and R2.
            SKT1    CHANGE    ; ④   ; If the CHANGE flag is set to 1
            BR      SECURITY_CHK
            ST      M1, R3          ; Writes the data to M1 and M2 again.
            ST      M2, R4
            CLR1    CHANGE
SECURITY_CHK:
            SET2    CMP, Z    : ①   ; Compares the input data with the security code.
            SUB     R1, M1
            SUB     R2, M2
            SKT1    Z
            BR      ERROR           ; Detects that the input data does not match the
                                    ; security code.
MAIN:
            Key-in processing       ; Security code rewrite mode
             R3 ← Contents of key C ; Assigns the contents of the pressed keys to R3
             R4 ← Contents of key D ; and R4.
            SET1    CHANGE    ; ⑤   ; Holds the CHANGE flag to 1 until the changing of the
                                    ; security code is completed.
            ST      M1, R3    ; ②   ; Rewrites the security code.
            ST      M2, R4    ; ③
            CLR1    CHANGE          ; Sets the CHANGE flag to 0 once changing of the
                                    ; security code is completed.
            BR      MAIN
ERROR:
            Stops the operation.
```

The program in example 2 sets the CHANGE flag in ⑤ to 1 before rewriting the security code in ② and ③. Even if a CE reset occurs before ③, the data is written again in ④ .

15.4 RELATIONSHIP BETWEEN CE RESET AND POWER-ON RESET

When V_{DD} is first turned on, power-on reset and CE reset may be applied simultaneously.

Sections 15.4.1 through **15.4.3** describe this reset operation.

15.4.1 When V_{DD} Pin and CE Pin Rise Simultaneously

Fig. 15-6 (a) shows the reset operation. Power-on reset starts the program from address 0000H.

15.4.2 When CE Pin Raised in Forced Halt State Caused by Power-On Reset.

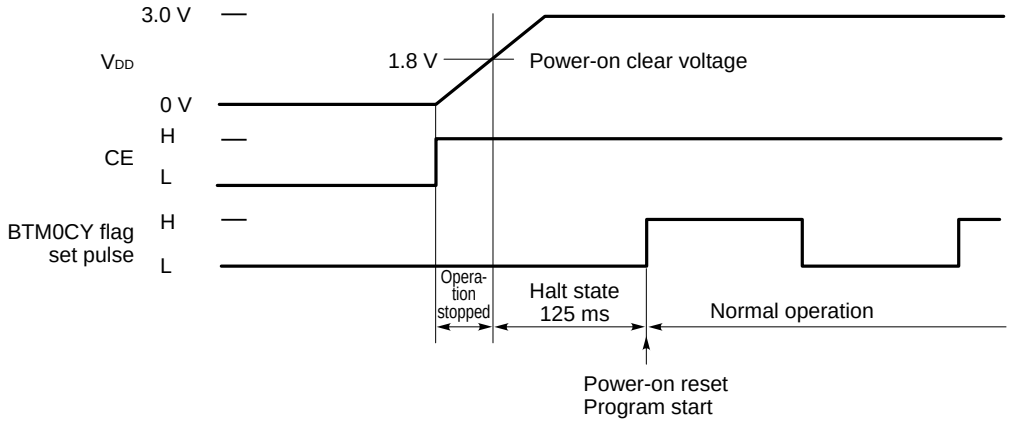
Fig. 15-6 (b) shows the reset operation. Power-on reset starts the program from address 0000H, as in **Section 15.4.1**.

15.4.3 When CE Pin Raised After Power-On Reset

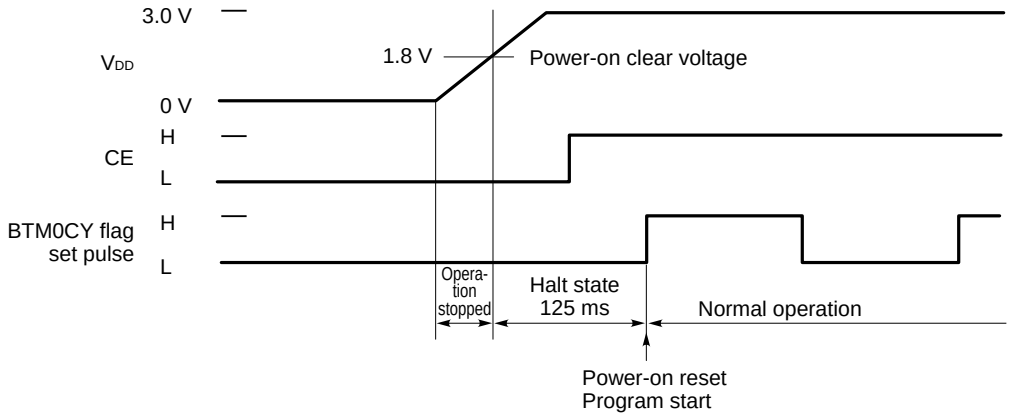
Fig. 15-6 (c) shows the reset operation. Power-on reset starts the program from address 0000H. CE reset restarts the program from address 0000H at the rising edge of the next BTM0CY flag setting signal.

Fig. 15-6 Relationship Between Power-On Reset and CE Reset (T_A = –10 to +50 °C)

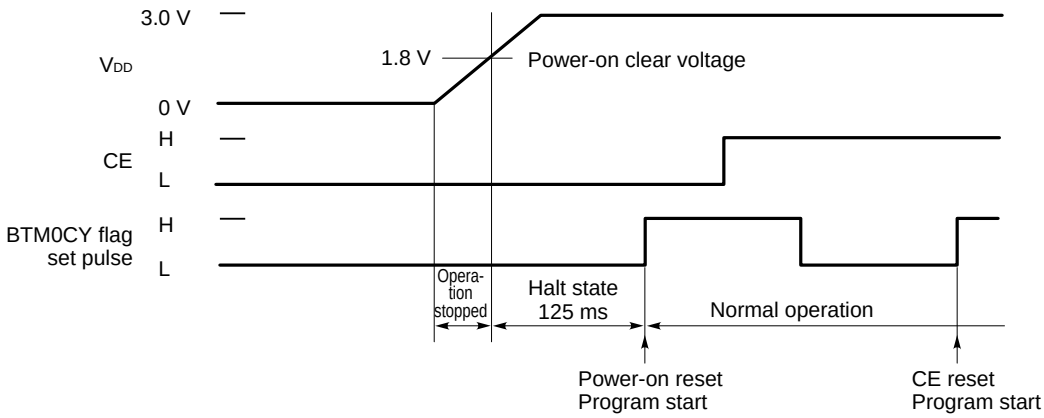
(a) When V_{DD} and CE pin raised simultaneously



(b) When CE Pin Raised in Halt State



(c) When CE Pin Raised After Power-On Reset



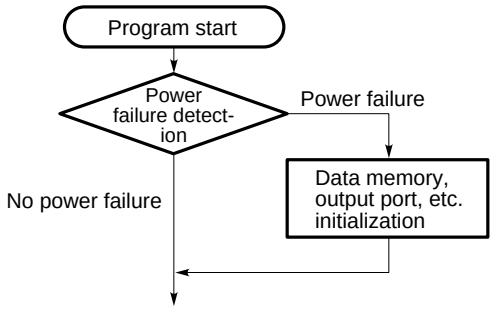
15.5 POWER FAILURE DETECTION

Power failure detection is used to judge whether the device is reset by turning on V_{DD} or by the CE pin, as shown in Fig. 15-7.

Since the contents of the data memory, output ports, etc. become “undefined” when V_{DD} is turned on, they are initialized by power failure detection.

A power failure can be detected by using a power failure detection circuit to detect the BTM0CY flag.

Fig. 15-7 Power Failure Detection Flowchart



15.5.1 Power Failure Detection Circuit

As shown in Fig. 15-1, the power failure detection circuit consists of a voltage detection circuit and timer carry disable flip-flop that is reset by the output (power-on clear signal) of the voltage detection circuit, and basic timer 0 carry.

The basic timer 0 carry disable flip-flop is set (1) by the power-on clear signal and is cleared (0) when a BTM0CY flag read instruction is executed.

When the basic timer 0 carry disable flip-flop is set (1), the BTM0CY flag is not set (1).

That is, when the power-on clear signal is output (at power-on reset), the program starts in the state in which the BTM0CY flag is reset and the setting disabled state is set until a BTM0CY read instruction is executed thereafter.

Once a BTM0CY read instruction is executed, the BTM0CY flag is set at each rising edge of the BTM0CY flag set pulse thereafter. When reset is applied to the device, the contents of the BTM0CY flag are monitored. If the BTM0CY flag has been cleared (0), power-on reset (power failure) is judged and if the BTM0CY flag has been set (1), CE reset (no power failure) is judged.

Since the voltage that can detect a power failure is the same as the voltage applied by power-on reset, V_{DD} becomes 1.8 V (T_A = –10 to +50 °C) at crystal oscillation and clock-stop (see **Section 15.2**).

Fig. 15-8 shows the BTM0CY flag state transition.

Fig. 15-9 shows timing chart and BTM0CY flag operation specified in Fig. 15-8.

Fig. 15-8 BTM0CY Flag State Transition

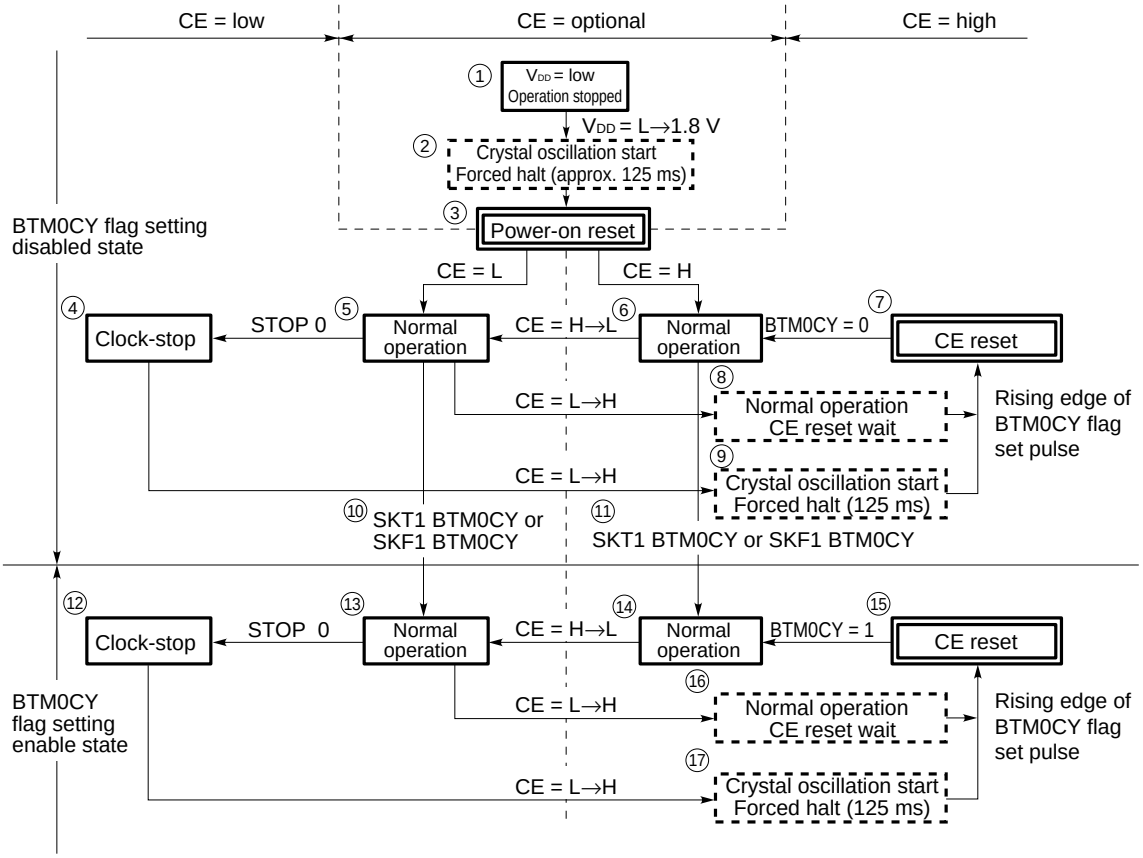
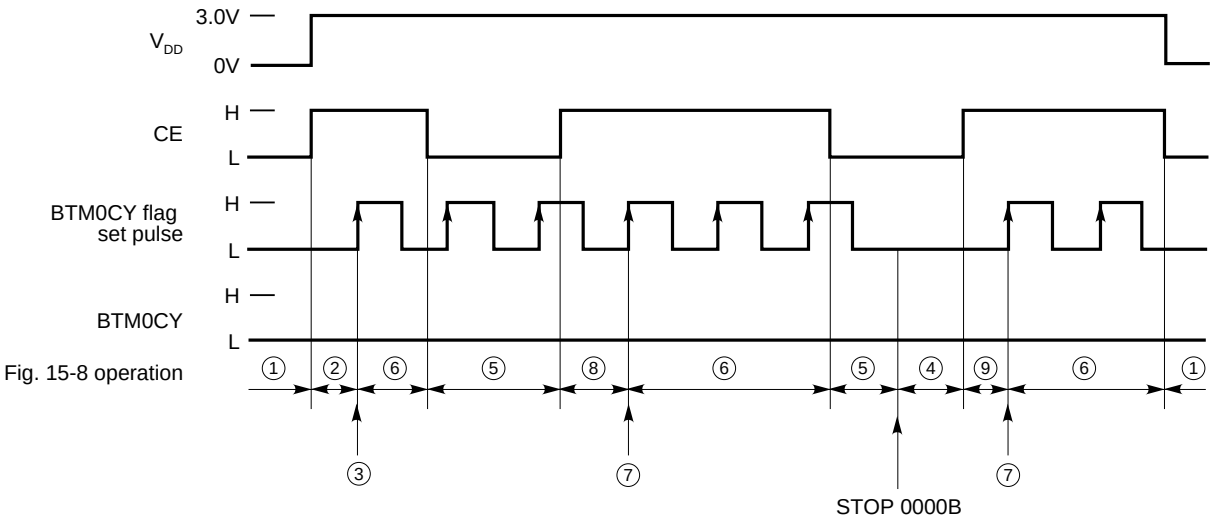
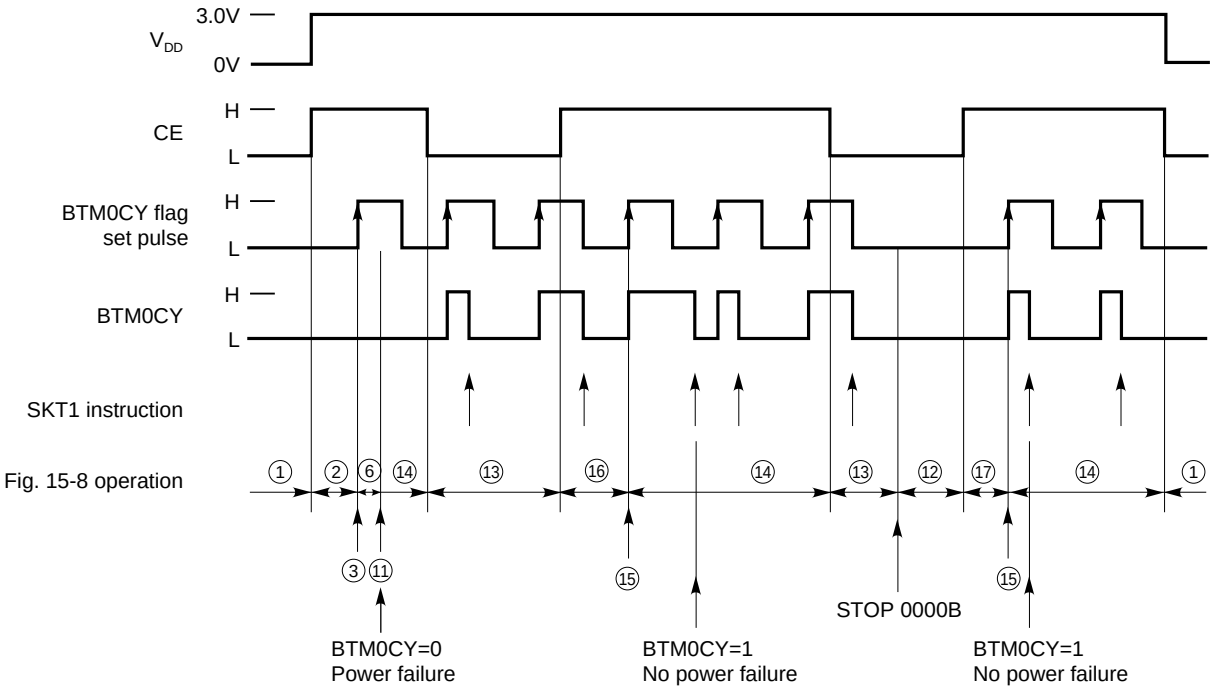


Fig. 15-9 BTM0CY Flag Operation ★

(a) When BTM0CY flag not detected even once (neither SKT1 BTM0CY nor SKF1 BTM0CY executed)



(b) When power failure detected with BTM0CY flag



15.5.2 Cautions at Power Failure Detection with BTM0CY Flag

When clock counting, etc. is performed with the BTM0CY flag, careful attention must be given to the following points.

(1) Clock updating

When writing a clock program by using basic timer 0, the clock must be updated after a power failure.
This is because the BTM0CY flag is cleared (0) and one clock count is lost by BTM0CY flag reading when a power failure is detected.

(2) Clock update processing time

When the clock is updated, its processing must end before the next rising edge of the BTM0CY flag set pulse.
This is because if the CE pin rises to high level during clock update processing, the clock update processing will not be executed up to the end and a CE reset will be applied.

For (1) and (2) above, see **Section 10.2.6 (1)**.

When processing is performed at a power failure, careful attention must be given to the following point.

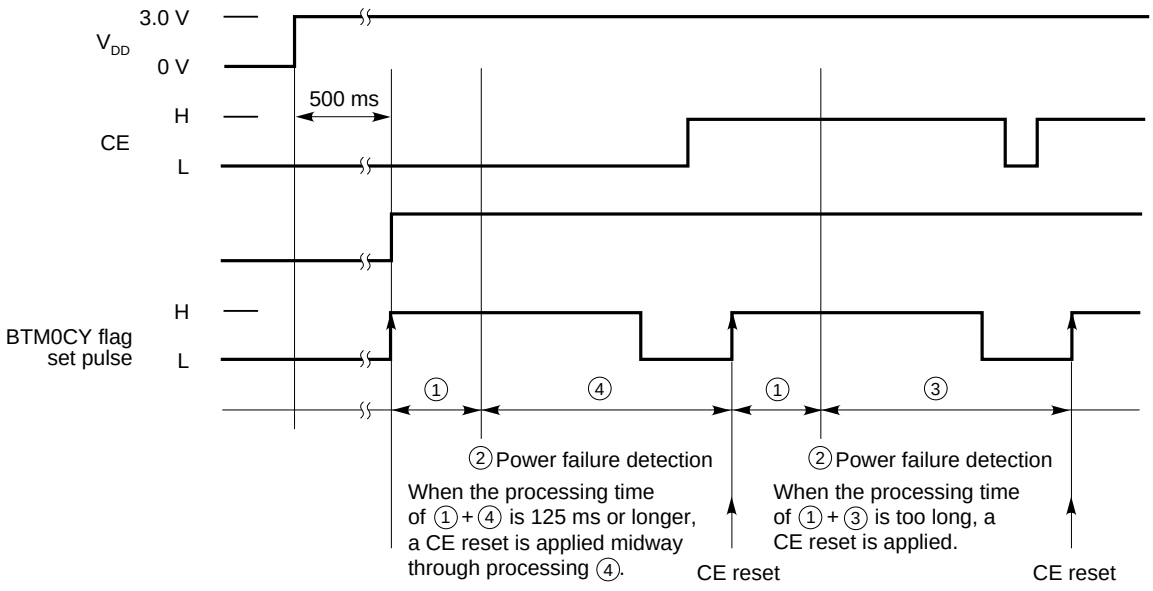
(3) Power failure detection timing

When clock counting, etc. is performed with the BTM0CY flag, the flag must be read for power-failure detection before the next rising edge of the BTM0CY flag set pulse, after a program starts from address 0000H.
This is because when the BTM0CY flag set time is set to 125 ms, for instance, and power failure detection is performed 126 ms after the program starts, one BTM0CY flag is lost.
See **Section 10.2.6 (1)**.
As shown in the example on the next page, power failure detection and initialization must be performed within the BTM0CY flag set time.
This is because when the CE pin is raised and CE reset is applied during power failure processing and initialization, these processings are interrupted and a problem may occur.

Example

```
Sample program
START:                                  ; Program address 0000H
; ①
      Reset processing
; ②
      SKT1    BTM0CY                    ; Power failure detection
      BR      INITIAL
BACKUP:
; ③
      Clock updating
      BR      MAIN
INITIAL:
; ④
      Initialization
MAIN:
      Main process
      SKT1    BTM0CY
      BR      MAIN
      Clock updating
      BR      MAIN
```

Operation example



16. INSTRUCTION SET

16.1 LIST OF INSTRUCTION SET

b_{14}-b_{11} \ b_{15} 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	RET RETSK GET DBF, p PUT p, DBF RORC r STOP s HALT h NOP	
1 0 0 0	8	LD r, m	ST m, r
1 0 0 1	9	SKE m, #n4	SKGE m, #n4
1 0 1 0	A	MOV @r, m	MOV m, @r
1 0 1 1	B	SKNE m, #n4	SKLT m, #n4
1 1 0 0	C	BR addr	CALL addr
1 1 0 1	D		MOV m, #n4
1 1 1 0	E		SKT m, #n
1 1 1 1	F		SKF m, #n

16.2 INSTRUCTIONS

Legend

ASR	:	Address stack register pointed to by the stack pointer
addr	:	Program memory address (11 bits)
CMP	:	Compare flag
CY	:	Carry flag
DBF	:	Data buffer
h	:	Halt release condition
m	:	Data memory address specified by m_R and m_C
m_R	:	Data memory row address (high-order)
m_C	:	Data memory column address (low-order)
n	:	Bit position (four bits)
n4	:	Immediate data (four bits)
PC	:	Program counter
p	:	Peripheral address
p_H	:	Peripheral address (three high-order bits)
p_L	:	Peripheral address (four low-order bits)
r	:	General register column address
s	:	Stop release condition
(×)	:	Contents of ×

Instruction set	Mnemonic	Operand	Operation	Instruction code			
				Op code	Operand		
Add	ADD	r, m	(r) ← (r) + (m)	00000	m_R	m_C	r
		m, #n4	(m) ← (m) + n4	10000	m_R	m_C	n4
	ADDC	r, m	(r) ← (r) + (m) + CY	00010	m_R	m_C	r
		m, #n4	(m) ← (m) + n4 + CY	10010	m_R	m_C	n4
Subtract	SUB	r, m	(r) ← (r) – (m)	00001	m_R	m_C	r
		m, #n4	(m) ← (m) – n4	10001	m_R	m_C	n4
	SUBC	r, m	(r) ← (r) – (m) – CY	00011	m_R	m_C	r
		m, #n4	(m) ← (m) – n4 – CY	10011	m_R	m_C	n4
Logical operation	OR	r, m	(r) ← (r) ∨ (m)	00110	m_R	m_C	r
		m, #n4	(m) ← (m) ∨ n4	10110	m_R	m_C	n4
	AND	r, m	(r) ← (r) ∧ (m)	00100	m_R	m_C	r
		m, #n4	(m) ← (m) ∧ n4	10100	m_R	m_C	n4
	XOR	r, m	(r) ← (r) ⊻ (m)	00101	m_R	m_C	r
		m, #n4	(m) ← (m) ⊻ n4	10101	m_R	m_C	n4
Test	SKT	m, #n	CMP ← 0, if (m) ∧ n = n, then skip	11110	m_R	m_C	n
	SKF	m, #n	CMP ← 0, if (m) ∧ n = 0, then skip	11111	m_R	m_C	n
Compare	SKE	m, #n4	(m) – n4, skip if zero	01001	m_R	m_C	n4
	SKNE	m, #n4	(m) – n4, skip if not zero	01011	m_R	m_C	n4
	SKGE	m, #n4	(m) – n4, skip if not borrow	11001	m_R	m_C	n4
	SKLT	m, #n4	(m) – n4, skip if borrow	11011	m_R	m_C	n4
Rotation	RORC	r	→ CY → $(r)_{b3}$ → $(r)_{b2}$ → $(r)_{b1}$ → $(r)_{b0}$ →	00111	000	0111	r
Transfer	LD	r, m	(r) ← (m)	01000	m_R	m_C	r
	ST	m, r	(m) ← (r)	11000	m_R	m_C	r
	MOV	@r, m	(m_R, (r)) ← (m)	01010	m_R	m_C	r
		m, @r	(m) ← (m_R, (r))	11010	m_R	m_C	r
		m, #n4	(m) ← n4	11101	m_R	m_C	n4
	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	01100	addr		
Sub-routine	CALL	addr	ASR ← PC, PC_{10-0} ← addr	11100	addr		
	RET		PC ← ASR	00111	000	1110	0000
	RETSK		PC ← ASR, and skip	00111	001	1110	0000
Others	STOP	s	STOP	00111	010	1111	s
	HALT	h	HALT	00111	011	1111	h
	NOP		No operation	00111	100	1111	0000

16.3 ASSEMBLER (AS17K) BUILT-IN MACRO INSTRUCTIONS

Legend

flag n : FLG-type symbol
< > : An operand enclosed in < > is optional.

	Mnemonic	Operand	Operation	n
Built-in macro	SKTn	flag 1, ··· flag n	if (flag 1) to (flag n) = all “1”, then skip	1 ≤ 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

17. RESERVED SYMBOLS

17.1 SYSTEM REGISTER (SYSREG)

Symbol	Attribute	Value	Read/write	Description
AR3	MEM	0.74H	R/W	Bits 15 to 12 of the address register
AR2	MEM	0.75H	R/W	Bits 11 to 8 of the address register
AR1	MEM	0.76H	R/W	Bits 7 to 4 of the address register
AR0	MEM	0.77H	R/W	Bits 3 to 0 of the address register
WR	MEM	0.78H	R/W	Window register
BANK	MEM	0.79H	R/W	Bank register
IXH	MEM	0.7AH	R/W	Index register high (Fixed to 0.)
MPH	MEM	0.7AH	R/W	Memory pointer high
MPE	FLG	0.7AH.3	R/W	Memory pointer enable flag
IXM	MEM	0.7BH	R/W	Index register middle
MPL	MEM	0.7BH	R/W	Memory pointer low
IXL	MEM	0.7CH	R/W	Index register low
RPH	MEM	0.7DH	R/W	General register pointer high
RPL	MEM	0.7EH	R/W	General register pointer low (Only low-order one bit is valid.)
PSW	MEM	0.7FH	R/W	Program status word
BCD	FLG	0.7EH.0	R/W	BCD flag
CMP	FLG	0.7FH.3	R/W	Compare flag
CY	FLG	0.7FH.2	R/W	Carry flag
Z	FLG	0.7FH.1	R/W	Zero flag
IXE	FLG	0.7FH.0	R/W	Index enable flag (Fixed to 0.)

17.2 DATA BUFFER (DBF)

Symbol	Attribute	Value	Read/write	Description
DBF3	MEM	0.0CH	R/W	DBF bits 15 to 12
DBF2	MEM	0.0DH	R/W	DBF bits 11 to 8
DBF1	MEM	0.0EH	R/W	DBF bits 7 to 4
DBF0	MEM	0.0FH	R/W	DBF bits 3 to 0

17.3 LCD SEGMENT REGISTER

Symbol	Attribute	Value	Read/write	Description
LCDD8	MEM	0.61H	R/W	LCD segment register
LCDD7	MEM	0.62H	R/W	LCD segment register
LCDD6	MEM	0.63H	R/W	LCD segment register
LCDD5	MEM	0.64H	R/W	LCD segment register
LCDD4	MEM	0.65H	R/W	LCD segment register
LCDD3	MEM	0.66H	R/W	LCD segment register
LCDD2	MEM	0.67H	R/W	LCD segment register
LCDD1	MEM	0.68H	R/W	LCD segment register
LCDD0	MEM	0.69H	R/W	LCD segment register

17.4 PORT REGISTER

Symbol	Attribute	Value	Read/write	Description
P0A3	FLG	0.70H.3	R/W	Bit 3 of port 0A
P0A2	FLG	0.70H.2	R/W	Bit 2 of port 0A
P0A1	FLG	0.70H.1	R/W	Bit 1 of port 0A
P0A0	FLG	0.70H.0	R/W	Bit 0 of port 0A
P0B2	FLG	0.71H.2	R/W	Bit 2 of port 0B
P0B1	FLG	0.71H.1	R/W	Bit 1 of port 0B
P0B0	FLG	0.71H.0	R/W	Bit 0 of port 0B
P0C3	FLG	0.72H.3	R/W	Bit 3 of port 0C
P0C2	FLG	0.72H.2	R/W	Bit 2 of port 0C
P0C1	FLG	0.72H.1	R/W	Bit 1 of port 0C
P0C0	FLG	0.72H.0	R/W	Bit 0 of port 0C
P0D0	FLG	0.73H.0	R/W	Bit 0 of port 0D

17.5 PERIPHERAL CONTROL REGISTER

Symbol	Attribute	Value	Read/write	Description
LCDEN	FLG	0.6AH.3	R/W	LCD enable flag
BEEP0SEL	FLG	0.6AH.2	R/W	$P0D_0$/BEEP pin selection flag
P0ABIO1	FLG	0.6AH.1	R/W	$P0A_1$ I/O selection flag
P0ABIO0	FLG	0.6AH.0	R/W	$P0A_0$ I/O selection flag
PLLMD1	FLG	0.6BH.3	R/W	PLL mode selection flag
PLLMD0	FLG	0.6BH.2	R/W	PLL mode selection flag
PLLRFCK1	FLG	0.6BH.1	R/W	PLL reference clock selection flag
PLLRFCK0	FLG	0.6BH.0	R/W	PLL reference clock selection flag
CE	FLG	0.6CH.0	R	CE pin status flag
PLLUL	FLG	0.6DH.0	R	PLL unlock flip-flop flag
BTM0CY	FLG	0.6EH.0	R	Basic timer 0 carry flag
BTM1CY	FLG	0.6FH.0	R	Basic timer 1 carry flag

17.6 PERIPHERAL HARDWARE REGISTER

Symbol	Attribute	Value	Read/write	Description
PLLR	DAT	41H	R/W	PLL data register

17.7 OTHERS

Symbol	Attribute	Value	Description
DBF	DAT	0FH	Fixed operand value for a PUT/GET instruction

18. ELECTRICAL CHARACTERISTICS ★

ABSOLUTE MAXIMUM RATINGS (T_A = 25 °C)

Parameter	Symbol	Conditions	Rated value	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
		Except for the CE pin	–0.3 to V_{DD} + 0.3	V
Output voltage	V_O		–0.3 to V_{DD} + 0.3	V
Output high current	I_{OH}	Each pin	–3.0	mA
		Total for all pins	–20.0	mA
Output low current	I_{OL}	Each pin	3.0	mA
		Total for all pins	20.0	mA
Operating ambient temperature	T_A		–20 to +50	°C
Storage temperature	T_{stg}		–55 to +125	°C

Caution Absolute maximum ratings are rated values beyond which physical damage may be caused to the product; if any of the parameters in the table above exceeds its rated value, even momentarily, the quality of the product may deteriorate. Therefore, ensure that the product is used within the rated values.

RECOMMENDED OPERATING RANGES (T_A = –10 to +50 °C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Supply voltage	V_{DD1}	CPU operation	1.8	3.0	3.6	V
	V_{DD2}	CPU operation, T_A = –20 to +50 °C	1.9	3.0	3.6	V
Rise time of supply voltage	t_{rise}	V_{DD}: 0 → 1.8 V T_A = –20 to +50 °C			500	ms

DC CHARACTERISTICS (T_A = –10 to +50 °C, V_{DD} = 1.8 to 3.6 V)

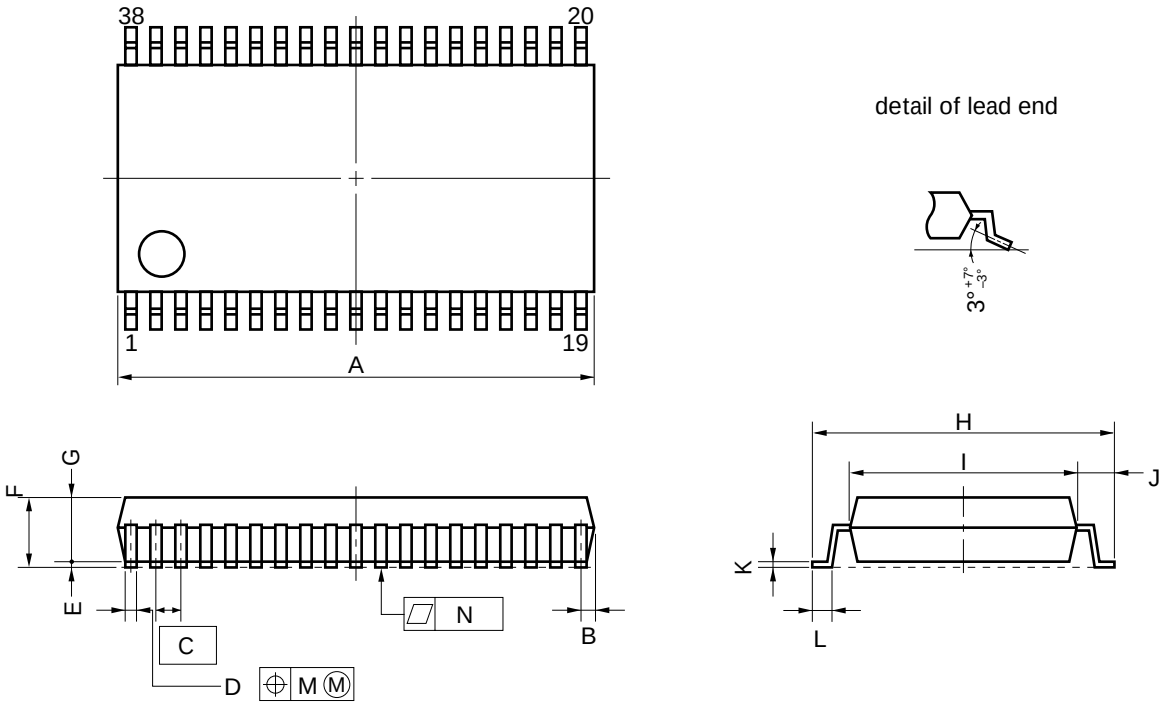
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Supply current	I_{DD1}	When the CPU and PLL are operating, with a sinusoidal wave applied to the X_{IN} pin (f_{IN} = 75 kHz, V_{IN} = V_{DD}) and VCOH pin (f_{IN} = 220 MHz, V_{IN} = 0.2 V_{P-P})		9	16	mA
	I_{DD2}	When the CPU is operating but the PLL is not, with a sinusoidal wave applied to the X_{IN} pin (f_{IN} = 75 kHz, V_{IN} = V_{DD})		30	70	μA
	I_{DD3}	When the CPU is operating but the PLL is not (when the HALT instruction is issued), with a sinusoidal wave applied to the X_{IN} pin (f_{IN} = 75 kHz, V_{IN} = V_{DD})		15	30	μA
Data hold voltage	V_{DDR}	When a power failure is detected with basic timer 0F/F	1.8		3.6	V
Data hold current	I_{DDR1}	When the crystal oscillation is stopped T_A = 25 °C, V_{DD} = 3.0 V			3	μA
	I_{DDR2}	When the crystal oscillation is stopped V_{DD} = 3.0 V			10	μA
Input high voltage	V_{IH1}	CE, $P0A_0$, $P0A_1$	0.8V_{DD}		V_{DD}	V
	V_{IH2}	$P0B_0$-$P0B_2$	0.6V_{DD}		V_{DD}	V
Input low voltage	V_{IL1}	CE, $P0A_0$, $P0A_1$	0		0.2V_{DD}	V
	V_{IL2}	$P0B_0$-$P0B_2$	0		0.1V_{DD}	V
Output high current	I_{OH1}	$P0A_0$, $P0A_1$, $P0C_0$-$P0C_3$, $P0D_0$ V_{OH} = V_{DD} – 1 V	–0.5			mA
	I_{OH2}	EO V_{OH} = V_{DD} – 1 V	–0.2			mA
	I_{OH3}	LCD_0-LCD_8 V_{OH} = V_{LCD1} – 1 V, V_{LCD1} = 2.7 to 3.3 V	–20			μA
Output low current	I_{OL1}	$P0A_0$, $P0A_1$, $P0D_0$ V_{OL} = 1 V	0.5			mA
	I_{OL2}	EO V_{OL} = 1 V	0.2			mA
	I_{OL3}	$P0C_0$-$P0C_3$ V_{OL} = 1 V	5		150	μA
	I_{OL4}	LCD_0-LCD_8 V_{OL} = 1 V, V_{LCD1} = 2.7 to 3.3 V	20			μA
Input high current	I_{IH1}	When the $P0B_0$ to $P0B_2$ pins are pulled down	3		100	μA
	I_{IH2}	When the X_{IN} pin is pulled down V_{IH} = V_{DD}	35			μA
LCD drive voltage	V_{LCD1}	Output between LCD_0 and LCD_8 pins must be left open. Between V_{LCD0} and GND pins = 0.1 μF, between V_{LCD1} and GND pins = 0.1 μF, and between CAP_0 and CAP_1 pins = 0.01 μF T_A = 25 °C	2.7	3.0	3.3	V
Output-off leakage current	I_L	EO			±1	μA

AC CHARACTERISTICS (T_A = –10 to +50 °C, V_{DD} = 1.8 to 3.6 V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Operating frequency	f_{IN1}	VCOL MF mode, with a sinusoidal wave applied at V_{IN} = 0.2 V_{p-p}	0.5		8	MHz
	f_{IN2}	VCOL HF mode, with a sinusoidal wave applied at V_{IN} = 0.2 V_{p-p}	6		55	MHz
	f_{IN3}	VCOH VHF mode, with a sinusoidal wave applied at V_{IN} = 0.2 V_{p-p}	40		220	MHz

19. PACKAGE DRAWING

38 PIN PLASTIC SHRINK SOP (300 mil)



P38GS-65-300B-1

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	12.71 MAX.	0.501 MAX.
B	0.51 MAX.	0.020 MAX.
C	0.65 (T.P.)	0.026 (T.P.)
D	0.30 ± 0.10	$0.012^{+0.004}_{-0.005}$
E	0.125±0.075	0.005±0.003
F	2.0 MAX.	0.079 MAX.
G	1.7±0.1	0.067±0.004
H	8.1±0.3	0.319±0.012
I	6.1±0.2	0.240±0.008
J	1.0±0.2	$0.039^{+0.009}_{-0.008}$
K	$0.15^{+0.10}_{-0.05}$	$0.006^{+0.004}_{-0.002}$
L	0.5±0.2	$0.020^{+0.008}_{-0.009}$
M	0.10	0.004
N	0.10	0.004

★ 20. RECOMMENDED SOLDERING CONDITIONS

The conditions listed below shall be met when soldering the μPD17015.

For details of the recommended soldering conditions, refer to our document *SMD Surface Mount Technology Manual* (IEI-1207).

Please consult with our sales offices in case any other soldering process is used, or in case soldering is done under different conditions.

Table 20-1 Soldering Conditions for Surface-Mount Devices

μPD17015GS-×××-GJG: 38-pin plastic shrink SOP (300 mil)

Soldering process	Soldering conditions	Symbol
Infrared ray reflow	Peak package's surface temperature: 235 °C Reflow time: 30 seconds or less (at 210 °C or more) Maximum allowable number of reflow processes: 2 <Cautions> (1) Do not start reflow-soldering the device if its temperature is higher than the room temperature because of a previous reflow soldering. (2) Do not use water for flux cleaning before a second reflow soldering.	IR35-00-2
VPS	Peak package's surface temperature: 215 °C Reflow time: 40 seconds or less (at 200 °C or more) Maximum allowable number of reflow processes: 2 <Cautions> (1) Do not start reflow-soldering the device if its temperature is higher than the room temperature because of a previous reflow soldering. (2) Do not use water for flux cleaning before a second reflow soldering.	VP15-00-2
Wave soldering	Solder temperature: 260 °C or less Flow time: 10 seconds or less Number of flow process: 1 Preheating temperature: 120 °C max. (measured on the package surface)	WS60-00-1
Partial heating method	Terminal temperature: 300 °C or less Flow time: 3 seconds or less (for each side of device)	—

Caution Do not apply more than a single process at once, except for "Partial heating method."

APPENDIX DEVELOPMENT TOOLS

The following support tools are available for developing programs for the μPD17015.

Hardware

Name	Description
In-circuit emulator [IE-17K IE-17K-ET**Note 1** EMU-17K**Note 2**]	The IE-17K, IE-17K-ET, and EMU-17K are in-circuit emulators applicable to the 17K series. The IE-17K and IE-17K-ET are connected to the PC-9800 series (host machine) or IBM PC/AT™ through the RS-232-C interface. The EMU-17K is inserted into the extension slot of the PC-9800 series (host machine). Use the system evaluation board (SE board) corresponding to each product together with one of these in-circuit emulators. *SIMPLEHOST* ™, a man machine interface, implements an advanced debug environment. The EMU-17K also enables user to check the contents of the data memory in real time.
SE board (SE-17015)	The SE-17015 is an SE board for the μPD17015. It is used solely for evaluating the system. It is also used for debugging in combination with the in-circuit emulator. ★
Emulation probe (EP-17K38GT)	The EP-17K38GT is an emulation probe for the μPD17015GS. It is used to connect the SE board with the target system in combination with the EV-9500GT-38. ★
Flexible printed circuit board for conversion (EV-9500GT-38**Note 3**)	The EV-9500GT-38 is a flexible printed circuit board used for a 38-pin plastic shrink SOP (300 mil). This board is used to connect the EP-17K38GT to the target system. ★

Notes 1. Low-end model, operating on an external power supply

2. The EMU-17K is a product of IC Co., Ltd. Contact IC Co., Ltd. (Tokyo, 03-3447-3793) for details.

3. The EP-17K38GT is supplied together with two EV-9500GT-38s. A set of five EV-9500GT-38s is also available.

Software

Name	Description	Host machine	OS		Distribution media	Part number
17K series assembler (AS17K)	AS17K is an assembler applicable to the 17K series. In developing μPD17015 programs, AS17K is used in combination with a device file (AS17015).	PC-9800 series	MS-DOS™		5.25-inch, 2HD	μS5A10AS17K
					3.5-inch, 2HD	μS5A13AS17K
		IBM PC/AT	PC DOS™		5.25-inch, 2HC	μS7B10AS17K
					3.5-inch, 2HC	μS7B13AS17K
Device file (AS17015)	AS17015 is a device file for the μPD17015. It is used together with the assembler (AS17K), which is applicable to the 17K series.	PC-9800 series	MS-DOS		5.25-inch, 2HD	μS5A10AS17015
					3.5-inch, 2HD	μS5A13AS17015
★		IBM PC/AT	PC DOS		5.25-inch, 2HC	μS7B10AS17015
					3.5-inch, 2HC	μS7B13AS17015
Support software (*SIMPLEHOST*)	*SIMPLEHOST*, running on the Windows™, provides man-machine-interface in developing programs by using a personal computer and the in-circuit emulator.	PC-9800 series	MS-DOS	Windows	5.25-inch, 2HD	μS5A10IE17K
					3.5-inch, 2HD	μS5A13IE17K
		IBM PC/AT	PC DOS		5.25-inch, 2HC	μS7B10IE17K
					3.5-inch, 2HC	μS7B13IE17K

Remark The following table lists the versions of the operating systems described in the above table.

OS	Versions
MS-DOS	Ver. 3.30 to Ver. 5.00A**Note**
PC DOS	Ver. 3.1 to Ver. 5.0**Note**
Windows	Ver. 3.0 to Ver. 3.1

Note MS-DOS versions 5.00 and 5.00A and PC DOS Ver. 5.0 are provided with a task swap function. This function, however, cannot be used in these software packages.

Cautions on CMOS Devices

① **Countermeasures against static electricity for all MOSs**

Caution When handling MOS devices, take care so that they are not electrostatically charged.

Strong static electricity may cause dielectric breakdown in gates. When transporting or storing MOS devices, use conductive trays, magazine cases, shock absorbers, or metal cases that NEC uses for packaging and shipping. Be sure to ground MOS devices during assembling. Do not allow MOS devices to stand on plastic plates or do not touch pins.

Also handle boards on which MOS devices are mounted in the same way.

② **CMOS-specific handling of unused input pins**

Caution Hold CMOS devices at a fixed input level.

Unlike bipolar or NMOS devices, if a CMOS device is operated with no input, an intermediate-level input may be caused by noise. This allows current to flow in the CMOS device, resulting in a malfunction. Use a pull-up or pull-down resistor to hold a fixed input level. Since unused pins may function as output pins at unexpected times, each unused pin should be separately connected to the V_{DD} or GND pin through a resistor.

If handling of unused pins is documented, follow the instructions in the document.

③ **Statuses of all MOS devices at initialization**

Caution The initial status of a MOS device is unpredictable when power is turned on.

Since characteristics of a MOS device are determined by the amount of ions implanted in molecules, the initial status cannot be determined in the manufacture process. NEC has no responsibility for the output statuses of pins, input and output settings, and the contents of registers at power on. However, NEC assures operation after reset and items for mode setting if they are defined.

When you turn on a device having a reset function, be sure to reset the device first.

[MEMO]

M4 94.11