DATA SHEET

NEC

MOS INTEGRATED CIRCUIT

μPD17240,17241,17242,17243,17244,17245,17246

4-BIT SINGLE-CHIP MICROCONTROLLERS FOR SMALL GENERAL-PURPOSE INFRARED REMOTE CONTROL TRANSMITTERS

DESCRIPTION

The μPD17240, 17241, 17242, 17243, 17244, 17245, 17246 (hereafter called the μPD17246 Subseries) are 4-bit single-chip microcontrollers for small general-purpose infrared remote control transmitters.

This subseries employs 17K general-purpose register system architecture for the CPU, and can directly execute operations between data memories instead of the conventional method of executing operations through an accumulator. Moreover, all the instructions are 16-bit/1-word instructions, enabling efficient programming.

In addition, a one-time PROM model, the μPD17P246, to which data can be written only once, is also available. This product is convenient either for evaluating the μPD17246 Subseries programs or for small-scale production of application systems.

Detailed function descriptions are provided in the following user's manual. Be sure to read them before designing.

μPD172×× Subseries User's Manual: U12795E

FEATURES

- Infrared remote controller carrier generator (REM output)
- 17K architecture: General-purpose register system
- Program memory (ROM), data memory (RAM)

	μPD17240	μPD17241	μPD17242	μPD17243	μPD17244	μPD17245	μPD17246
Program memory (ROM)	4 KB (2,048 × 16)	8 KB (4,096 × 16)	12 KB (6,144 × 16)	16 KB (8,192 × 16)	20 KB (10,240 × 16)	24 KB (12,288 × 16)	32 KB (16,384 × 16)
Data memory (RAM)	447 × 4 bits						

- 8-bit timer: 1 channel
- Basic interval timer/watchdog timer: 1 channel
- Instruction execution time (can be changed in two steps)
  - @ fx = 4 MHz: 4 μs (high-speed mode)/8 μs (normal mode)
- External interrupt pin (INT/P1B0): 1
- I/O pins: 24
- Supply voltage: V_{DD} = 2.0 to 3.6 V
- On-chip RAM retention detector
- Low-voltage detector (mask option)

Unless otherwise specified, the μPD17246 is treated as the representative model throughout this document.

Document No. U15002EJ1V0DS00 (1st edition)
Date Published April 2003 N CP (K)
Printed in Japan

The mark ★ shows major revised points.

APPLICATIONS

Preset remote controllers, toys, portable systems, etc.

ORDERING INFORMATION

Part Number	Package
μPD17240MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))
μPD17241MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))
μPD17242MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))
μPD17243MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))
μPD17244MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))
μPD17245MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))
μPD17246MC-×××-5A4	30-pin plastic SSOP (7.62 mm (300))

Remark ××× indicates ROM code suffix.

DIFFERENCES BETWEEN μPD17246 SUBSERIES, μPD17236 SUBSERIES, AND μPD17255 SUBSERIES (1/2)

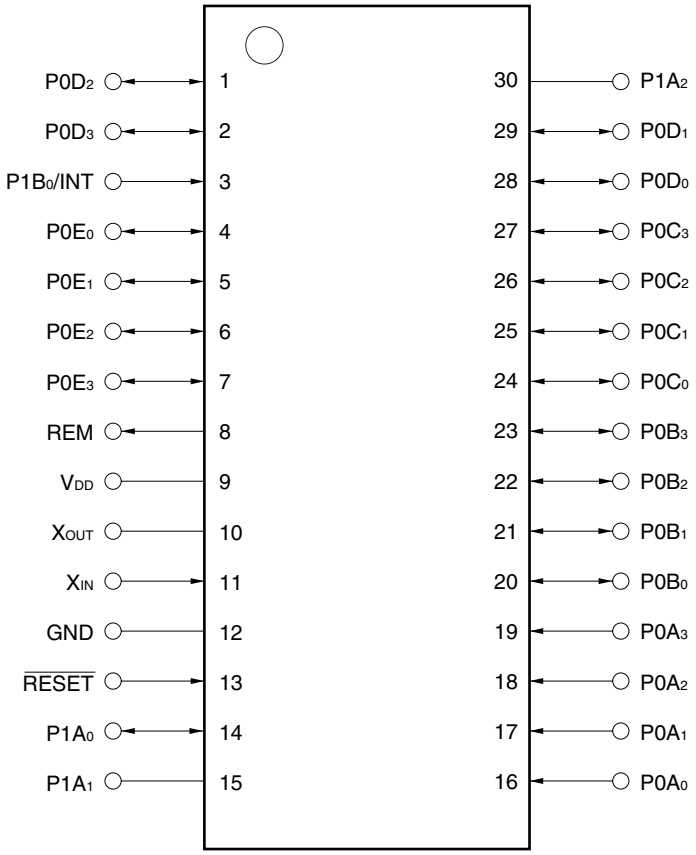
Item	μPD17246 Subseries	μPD17236 Subseries	μPD17225 Subseries
ROM	μPD17240: 2,048 × 16 bits μPD17241: 4,096 × 16 bits μPD17242: 6,144 × 16 bits μPD17243: 8,192 × 16 bits μPD17244: 10,240 × 16 bits μPD17245: 12,288 × 16 bits μPD17246: 16,384 × 16 bits	μPD17230: 2,048 × 16 bits μPD17231: 4,096 × 16 bits μPD17232: 6,144 × 16 bits μPD17233: 8,192 × 16 bits μPD17234: 10,240 × 16 bits μPD17235: 12,288 × 16 bits μPD17236: 16,384 × 16 bits	μPD17225: 2,048 × 16 bits μPD17226: 4,096 × 16 bits μPD17227: 6,144 × 16 bits μPD17228: 8,192 × 16 bits
RAM	447 × 4 bits	223 × 4 bits	111 × 4 bits (μPD17225, 17226) 223 × 4 bits (μPD17227, 17228)
Ports	$P0B_0$ to $P0B_3$: I/O (bit I/O) $P0C_0$ to $P0C_3$: I/O (group I/O) $P0D_0$ to $P0D_3$: I/O (group I/O) $P1A_0$ to $P1A_2$: I/O (bit I/O) $P1B_0$: I/O, functions alternately as INT pin	$P0B_0$ to $P0B_3$: I/O (bit I/O) $P0C_0$ to $P0C_3$: I/O (group I/O) $P0D_0$ to $P0D_3$: I/O (group I/O) $P1A_0$: Input or output selectable by mask option	$P0B_0$ to $P0B_3$: Input $P0C_0$ to $P0C_3$: Output $P0D_0$ to $P0D_3$: Output
Reset • Reset by watchdog timer • Reset by stack pointer • Low-voltage detector (mask option)	The $\overline{\text{RESET}}$ pin is internally pulled down by the occurrence of the internal reset signals on the left, causing a reset (usually, the $\overline{\text{RESET}}$ pin is pulled up).		A low level is output from the $\overline{\text{WDOUT}}$ pin by the occurrence of the internal reset signals on the left, and a reset takes place if the $\overline{\text{WDOUT}}$ pin is externally connected to the $\overline{\text{RESET}}$ pin.
Capacitor for oscillation	Selected by mask option (15 pF)	Not provided	
Vector address	Basic interval timer: 0002H Rising and falling edges of INT pin: 0003H 8-bit timer: 0004H	Basic interval timer: 0001H Rising and falling edges of INT pin: 0002H 8-bit timer: 0003H	
RAM retention flag	Provided	Not provided	

DIFFERENCES BETWEEN μPD17246 SUBSERIES, μPD17236 SUBSERIES, AND μPD17255 SUBSERIES (2/2)

Item	μPD17246 Subseries	μPD17236 Subseries	μPD17225 Subseries
STOP mode release condition	<1> When any of pins $P0A_0$ to $P0A_3$ goes low <2> When pins $P0B_0$ to $P0B_3$, $P0C_0$ to $P0C_3$, and $P0D_0$ to $P0D_3$ are used as input pins and when any of them goes low <3> When an interrupt request (IRQ) of the interrupt for which the IP flag is set is generated at the rising edge or falling edge of the INT pin <4> When $P0E_0$ to $P0E_3$ are used as input pins when a key matrix is used and when any of these pins goes low <5> When $P1A_0$ to $P1A_2$ and $P1B_0$ are used as input pins when a key matrix is used and when the level of any of these pins equals the set clear level	<1> When any of pins $P0A_0$ to $P0A_3$ goes low <2> When pins $P0B_0$ to $P0B_3$, $P0C_0$ to $P0C_3$, and $P0D_0$ to $P0D_3$ are used as input pins and when any of them goes low <3> When an interrupt request (IRQ) of the interrupt for which the IP flag is set is generated at the rising edge or falling edge of the INT pin	When any of pins $P0A_0$ to $P0A_3$ and $P0B_0$ to $P0B_3$ goes low
★ Carrier frequency (f_X = 4 MHz)	Selected by register file (after reset: $f_X/2$) <1> If carrier generation clock is $f_X/2$: 3.9 kHz to 1 MHz <2> If carrier generation clock is f_X: 7.8 kHz to 2 MHz <3> If carrier generation clock is $2f_X$: 15.6 kHz to 4 MHz	Selected by mask option <1> If carrier generation clock is $f_X/2$: 7.8 kHz to 1 MHz <2> If carrier generation clock is f_X: 15.6 kHz to 2 MHz	7.8 kHz to 1 MHz
NRZ low-level period setting modulo register (NRZLTMM) and NRZ high-level period setting modulo register (NRZHTMM)	• NRZLTMM: 8 bits (REM output control bit is bit 1 of register file at address 12H) • NRZHTMM: 8 bits	• NRZLTMM: 7 bits (bit 7 is REM output control bit) • NRZHTMM: 7 bits (bit 7 is fixed to 0)	

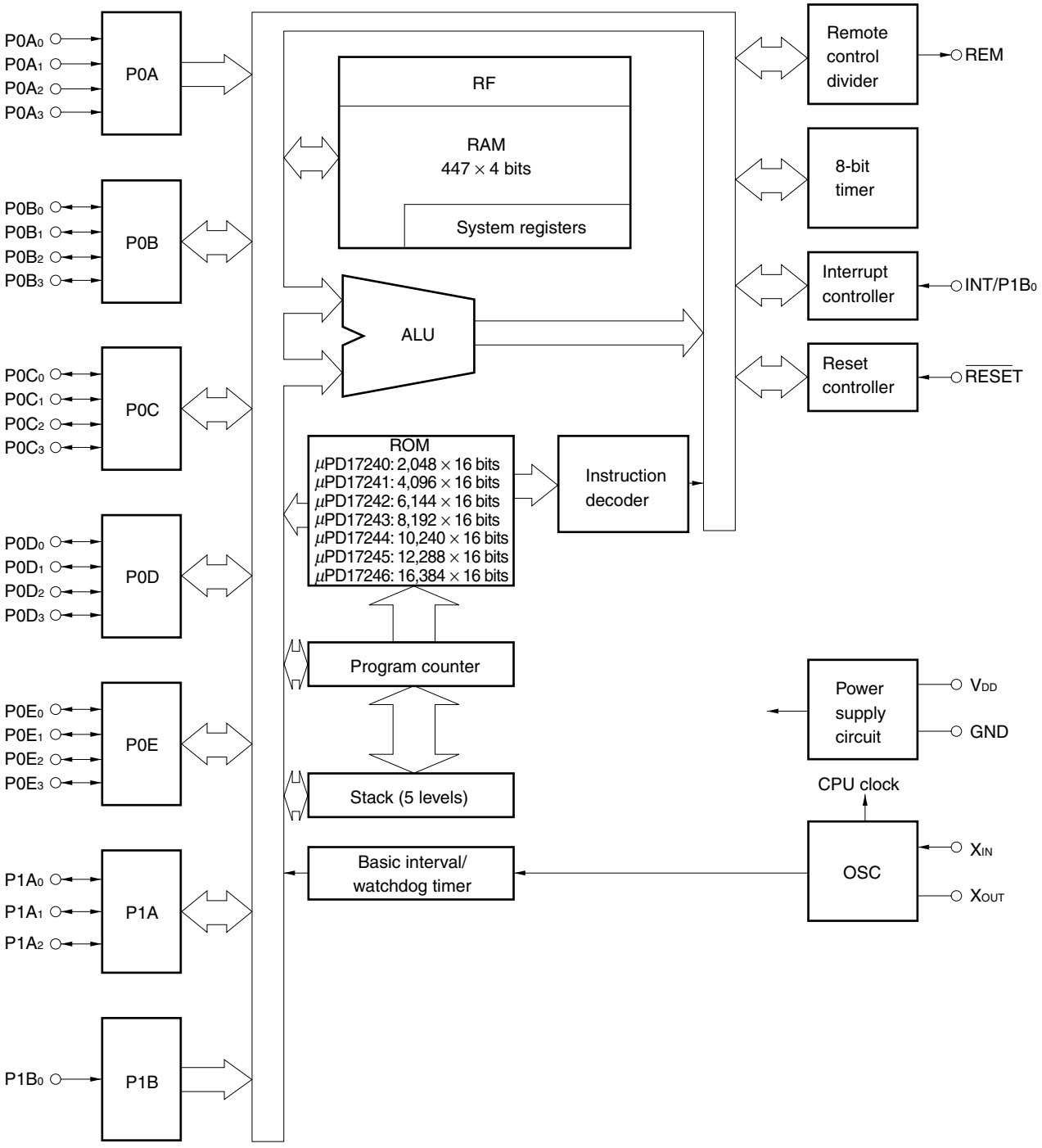
PIN CONFIGURATION (TOP VIEW)

- 30-pin plastic SSOP (7.62 mm (300))
  μPD17240MC-×××-5A4, 17241MC-×××-5A4, 17242MC-×××-5A4, 17243MC-×××-5A4,
  μPD17244MC-×××-5A4, 17245MC-×××-5A4, 17246MC-×××-5A4



GND: Ground
INT: External interrupt request signal input
P0A0 to P0A3: Input port (CMOS input with pull-up resistor)
P0B0 to P0B3: I/O port (CMOS input with pull-up resistor/N-ch open-drain output)
P0C0 to P0C3: I/O port (CMOS input with pull-up resistor/N-ch open-drain output)
P0D0 to P0D3: I/O port (CMOS input with pull-up resistor/N-ch open-drain output)
P0E0 to P0E3: I/O port (when key matrix is used: CMOS input with pull-up resistor/N-ch open-drain output, when key matrix is not used: CMOS input/push-pull output)
P1A0/P1A2: Input port (when key matrix is used: CMOS input/N-ch open-drain output, when key matrix is not used: CMOS input/push-pull output)
P1B0: Input port (CMOS input)
REM: Remote controller output (CMOS push-pull output)
$\overline{\text{RESET}}$: Reset input
VDD: Power supply
XIN, XOUT: Resonator connection

BLOCK DIAGRAM



CONTENTS

1. PIN FUNCTIONS

1.1 Pin Function List (1/3)

Pin No.	Pin Name	Function	Output Format	After Reset
28 29 1 2	$P0D_0$ $P0D_1$ $P0D_2$ $P0D_3$	These pins constitute a 4-bit I/O port which can be set to the input or output mode in 4-bit units (group I/O). In the input mode, these pins serve as CMOS input pins with a pull-up resistor, and can be used to input the key return signals of a key matrix. The standby status must be released when at least one of the input lines goes low. In the output mode, these pins are used as N-ch open-drain output pins and can be used to output the signals of a key matrix.	N-ch open drain	Low-level output
3	$P1B_0$/INT	This is an input port pin. Whether this pin functions as the $P1B_0$ pin or the INT pin can be selected by the register file. • $P1B_0$ This is a 1-bit CMOS input port. This port can be used to input key return signals when a key matrix is used. At this time, whether a pull-up/down resistor is connected to this port and the standby mode release condition (whether it is released when this pin is high or low) can be selected. 1. If connection of a resistor is specified and if it is specified that the standby mode is released when this pin goes low ... A pull-up resistor is connected. If a low level is input to the $P1B_0$ pin, the standby mode is released. 2. If connection of a resistor is specified and if it is specified that the standby mode is released when this pin goes high ... A pull-down resistor is connected. If a high level is input to the $P1B_0$ pin, the standby mode is released. 3. If connection of a resistor is not specified and if it is specified that the standby mode is released when this pin goes low (or high) ... No resistor is connected. If a low (or high) level is input to the $P1B_0$ pin, the standby mode is released. If a key matrix is not used, whether a resistor is connected and whether the resistor is pull-up or pull-down can be selected. • INT This is an external interrupt request signal. It can also be used to release the standby mode if an external interrupt request signal is input to this pin while the INT pin interrupt enable flag (IP) is set.	–	$P1B_0$ input (when key matrix not used and no resistor connected)

1.1 Pin Function List (2/3)

Pin No.	Pin Name	Function	Output Format	After Reset
4 5 6 7	$P0E_0$ $P0E_1$ $P0E_2$ $P0E_3$	These pins constitute a 4-bit I/O port that can be set to the input or output mode in 1-bit units. If this port is set to the input mode when a key matrix is used, it functions as a CMOS input port with a pull-up resistor and can be used to input key return signals. If one of the pins of this port goes low, the standby mode is released. If this port is set to the output mode when a key matrix is used, it functions as an N-ch open-drain output port and can be used to output key matrix signals. If this port is set to the input mode when a key matrix is not used, it functions as a CMOS input port to/from which a resistor can be connected/disconnected in 1-bit units. If this port is set in the output mode when a key matrix is not used, it functions as a high-current CMOS output port.	When key matrix is used: N-ch open-drain, when key matrix is not used: CMOS push-pull	CMOS input (when key matrix is not used and no resistor connected)
8	REM	Outputs transfer signal for infrared remote controller. Active-high output.	CMOS push-pull	Low-level output
9	V_{DD}	Power supply	–	–
10 11	X_{OUT} X_{IN}	Connects ceramic resonator for system clock oscillation. A capacitor (15 pF) for oscillation can be connected by using a mask option.	–	(Oscillation stops)
12	GND	Ground	–	–
13	$\overline{\text{RESET}}$	System reset input. Turns ON pull down resistor if the POC or watchdog timer overflows and if the stack pointer overflows or underflows, and resets the system. Usually, the pull-down resistor is ON.	–	Input

1.1 Pin Function List (3/3)

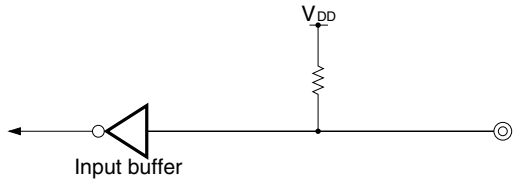
Pin No.	Pin Name	Function	Output Format	After Reset
14 15 30	$P1A_0$ $P1A_1$ $P1A_2$	These pins constitute a 3-bit I/O port that can be set to the input or output mode in 1-bit units. If this port is set to the input mode when a key matrix is used, it functions as a CMOS input port and can be used to input key return signals. At this time, whether a pull-up/down resistor is connected to this port and the standby mode release condition (whether it is released when this pin is high or low) can be selected in 1-bit units 1. If connection of a resistor is specified and if it is specified that the standby mode is released when this port goes low ... A pull-up resistor is connected. If a low level is input to the set key, the standby mode is released. 2. If connection of a resistor is specified and if it is specified that the standby mode is released when this port goes high ... A pull-down resistor is connected. If a high level is input to the set key, the standby mode is released. 3. If connection of a resistor is not specified and if it is specified that the standby mode is released when this port goes low (or high) ... No resistor is connected. If a low (or high) level is input to the set key, the standby mode is released. If this port is set to the output mode when a key matrix is used, it functions as an N-ch open-drain output port and can be used to output key matrix signals. If this port is set to the input mode when a key matrix is used, it functions as a CMOS input port. Connection of a resistor to this port and whether the resistor is pull-up or pull-down can be selected in 1-bit units. If this port is set in the output mode when a key matrix is not used, it functions as a high-current CMOS output port.	When key matrix is used: N-ch open-drain, when key matrix is not used: CMOS push-pull.	CMOS input (when key matrix is not used and no resistor connected)
16 17 18 19	$P0A_0$ $P0A_1$ $P0A_2$ $P0A_3$	These pins are CMOS input pins with a 4-bit pull-up resistor. They can be used to input the key return signals of a key matrix. If any one of these pins goes low, the standby status is released.	–	CMOS input with pull-up resistor
20 21 22 23	$P0B_0$ $P0B_1$ $P0B_2$ $P0B_3$	These pins constitute a 4-bit I/O port that can be set to the input or output mode in 1-bit units. In the input mode, these pins are CMOS input pins with a pull-up resistor, and can be used to input the key return signals of a key matrix. The standby status is released when at least one of these pins goes low. In the output mode, they serve as N-ch open-drain output pins and can be used to output the key return signals of a key matrix.	N-ch open drain	CMOS input with pull-up resistor
24 25 26 27	$P0C_0$ $P0C_1$ $P0C_2$ $P0C_3$	These pins constitute a 4-bit I/O port that can be set to the input or output mode in 4-bit units (group I/O). In the input mode, these pins are CMOS input pins with a pull-up resistor, and can be used to input the key return signals of a key matrix. The standby status is released when at least one of these pins goes low. In the output mode, they serve as N-ch open-drain output pins and can be used to output the key return signals of a key matrix.	N-ch open drain	Low-level output

1.2 I/O Circuits

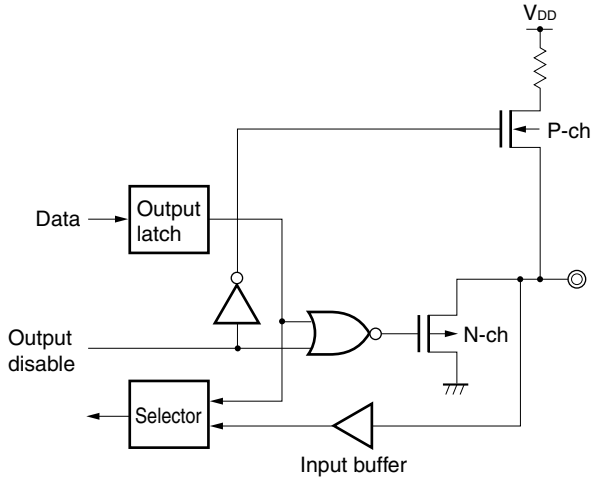
The equivalent I/O circuit for each μPD17246 pin is shown below.

★ **Figure 1-1. I/O Circuits (1/2)**

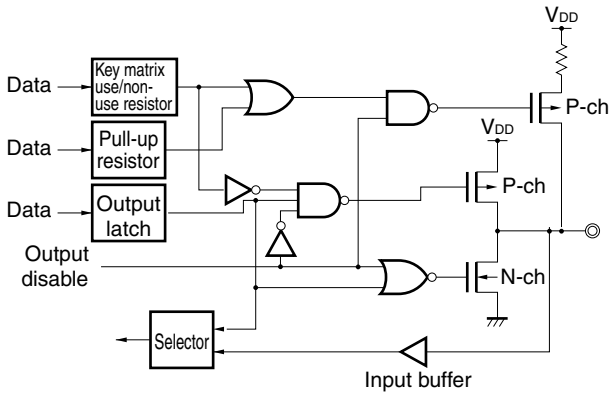
(1) P0A



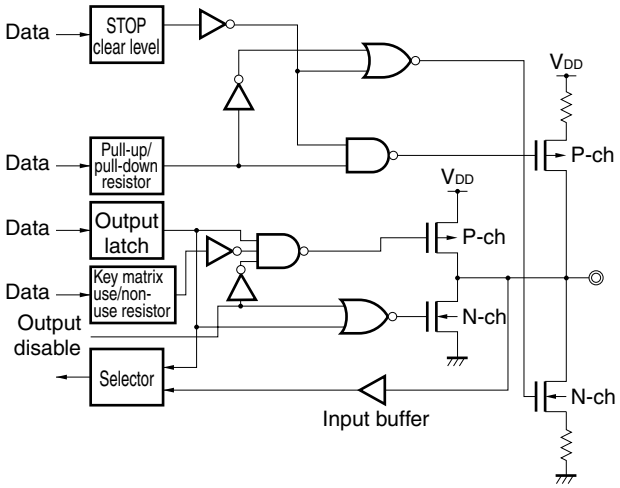
(2) P0B, P0C, P0D



(3) P0E



(4) P1A



(5) P1B

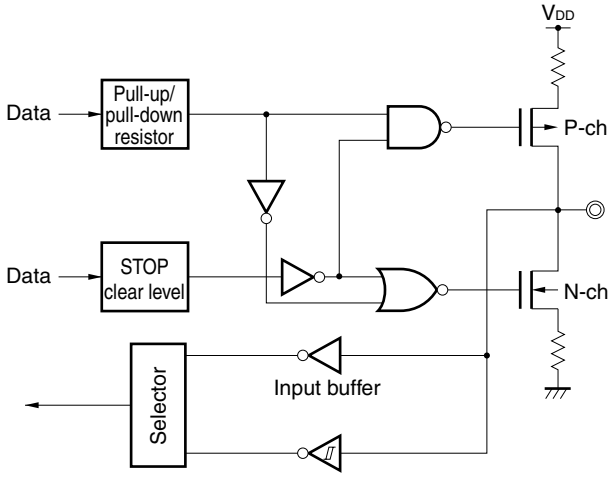
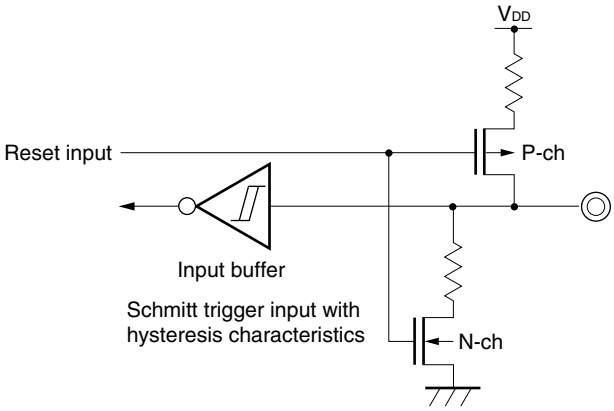
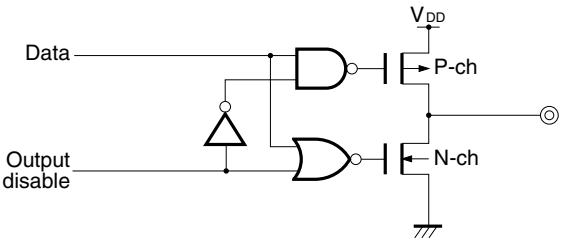


Figure 1-1. I/O Circuits (2/2)

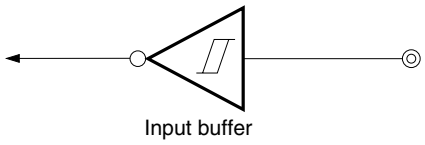
(6) $\overline{\text{RESET}}$



(8) REM



(7) INT



Schmitt trigger input with hysteresis characteristics

1.3 Handling of Unused Pins

Handle the unused pins as follows.

Table 1-1. Handling of Unused Pins

Pin Name	Recommended Connection
$P0A_0$ to $P0A_3$	Leave open.
$P0B_0$ to $P0B_3$	
$P0C_0$ to $P0C_3$	
$P0D_0$ to $P0D_3$	
★ $P0E_0$ to $P0E_3$	Connect to GND (When input).
$P1A_0$ to $P1A_2$	
$P1B_0$/INT	Connect to GND.
REM	Leave open.

2. MEMORY SPACE

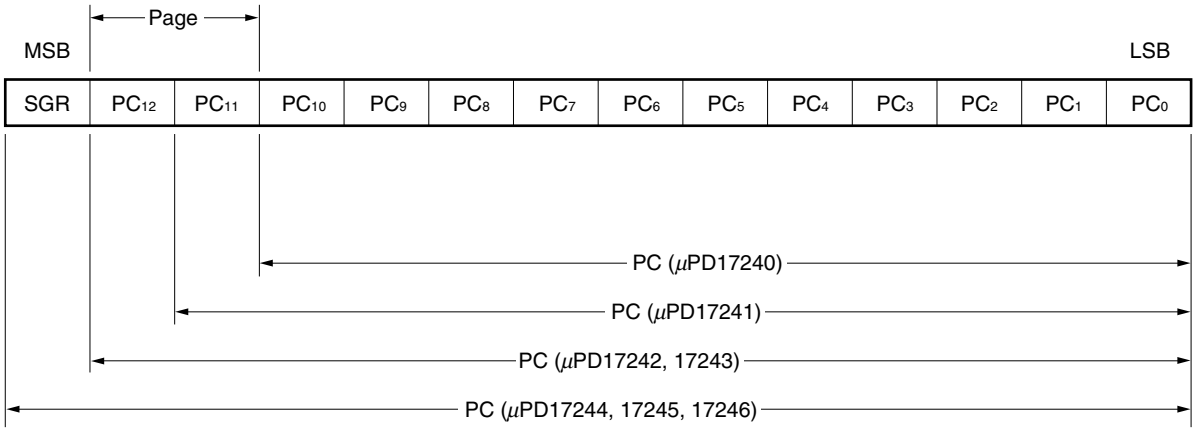
2.1 Program Counter (PC)

The program counter (PC) specifies an address of the program memory (ROM).

The program counter consists of an 11/12/13-bit binary counter and a 1-bit segment register (SGR) as shown in Figure 2-1.

Its contents are initialized to address 0000H at reset.

Figure 2-1. Configuration of Program Counter



2.1.1 Segment register (SGR)

The segment register specifies a segment of the program memory.

Table 2-1 shows the relationship between the segment register and program memory.

Table 2-1. Relationship Between Segment Register and Program Memory

Value of Segment Register	Segment of Program Memory
0	Segment 0
1	Segment 1

The segment register is set when the following instructions are executed.

- BR @AR
- CALL @AR
- SYSCAL entry

The first address of the subroutine that can be called by the system call instruction (“SYSCAL entry”) is the first 16 steps of each block (blocks 0 to 7) in page 0 of segment 1 (system segment).

Figure 2-2. Outline of System Call Instruction

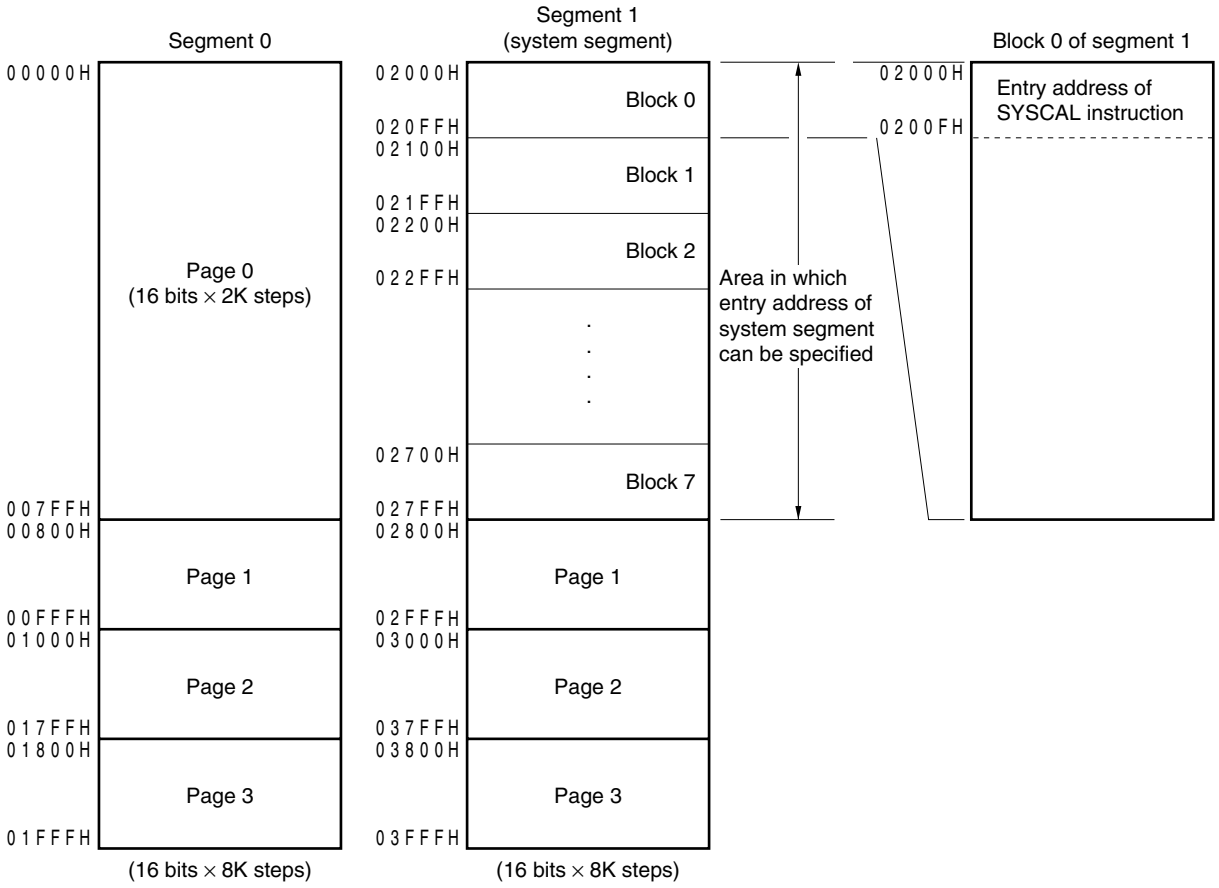


Figure 2-3. Value of Program Counter on Execution of Each Instruction

Instruction \ Program Counter		Contents of Program Counter (PC)[Note]														
		SGR	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
BR addr	Page 0	Retained	0	0	Operand of instruction (addr)										
	Page 1		0	1											
	Page 2		1	0											
	Page 3		1	1											
CALL addr		Retained	0	0	Operand of instruction (addr)										
SYSCAL entry		1	0	0	$entry_H$			0	0	0	0	$entry_L$			
BR @AR CALL @AR MOVT DBF, @AR		Contents of address register													
RET RETSK RETI		Contents (return address) of address stack register (ASR) specified by stack pointer (SP)													
Other instructions (including skip instruction)		Retained	Increment												
On acknowledging interrupt		0	Vector address of each interrupt												
Watchdog timer reset, $\overline{\text{RESET}}$ pin, reset by stack pointer		0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note μPD17240: b_0 to b_{10}
μPD17241: b_0 to b_{11}
μPD17242, 17243: b_0 to b_{12}
μPD17244, 17245, 17246: b_0 to b_{12}, SGR

Remark $entry_H$: Higher 3 bits of entry
$entry_L$: Lower 4 bits of entry

Table 2-2. Interrupt Vector Address

Priority	Internal/External	Interrupt Source	Vector Address
1	Internal	8-bit timer	0004H
2	External	Rising and falling edges of INT pin	0003H
3	Internal	Basic interval timer	0002H

2.2 Program Memory (ROM)

The configuration of the program memory is as follows.

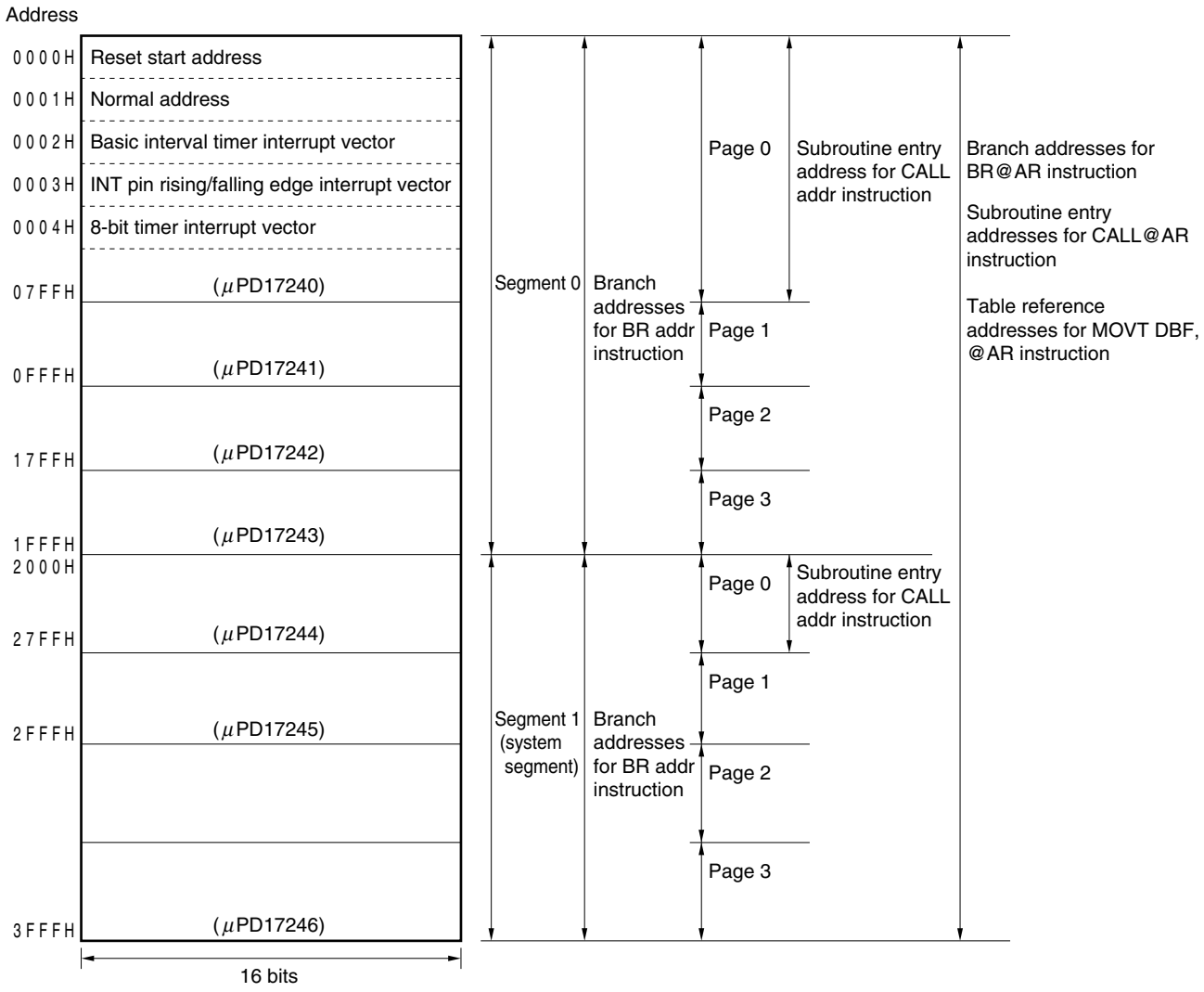
Part Number	Program Memory Capacity	Program Memory Address
μPD17240	2,048 × 16 bits	0000H to 07FFH
μPD17241	4,096 × 16 bits	0000H to 0FFFH
μPD17242	6,144 × 16 bits	0000H to 17FFH
μPD17243	8,192 × 16 bits	0000H to 1FFFH
μPD17244	10,240 × 16 bits	0000H to 27FFH
μPD17245	12,288 × 16 bits	0000H to 2FFFH
μPD17246	16,384 × 16 bits	0000H to 3FFFH

The program memory stores the program, interrupt vector table, and fixed data table.

The program memory is addressed by the program counter.

Figure 2-4 shows the program memory map. The entire range of the program memory can be addressed by the BD addr, BR @AR, CALL @AR, MOVT DBF, and @AR instructions. Note, however, that the subroutine entry addresses that can be specified by the CALL addr instruction are from 0000H to 07FFH.

Figure 2-4. Program Memory Map



2.3 Stack

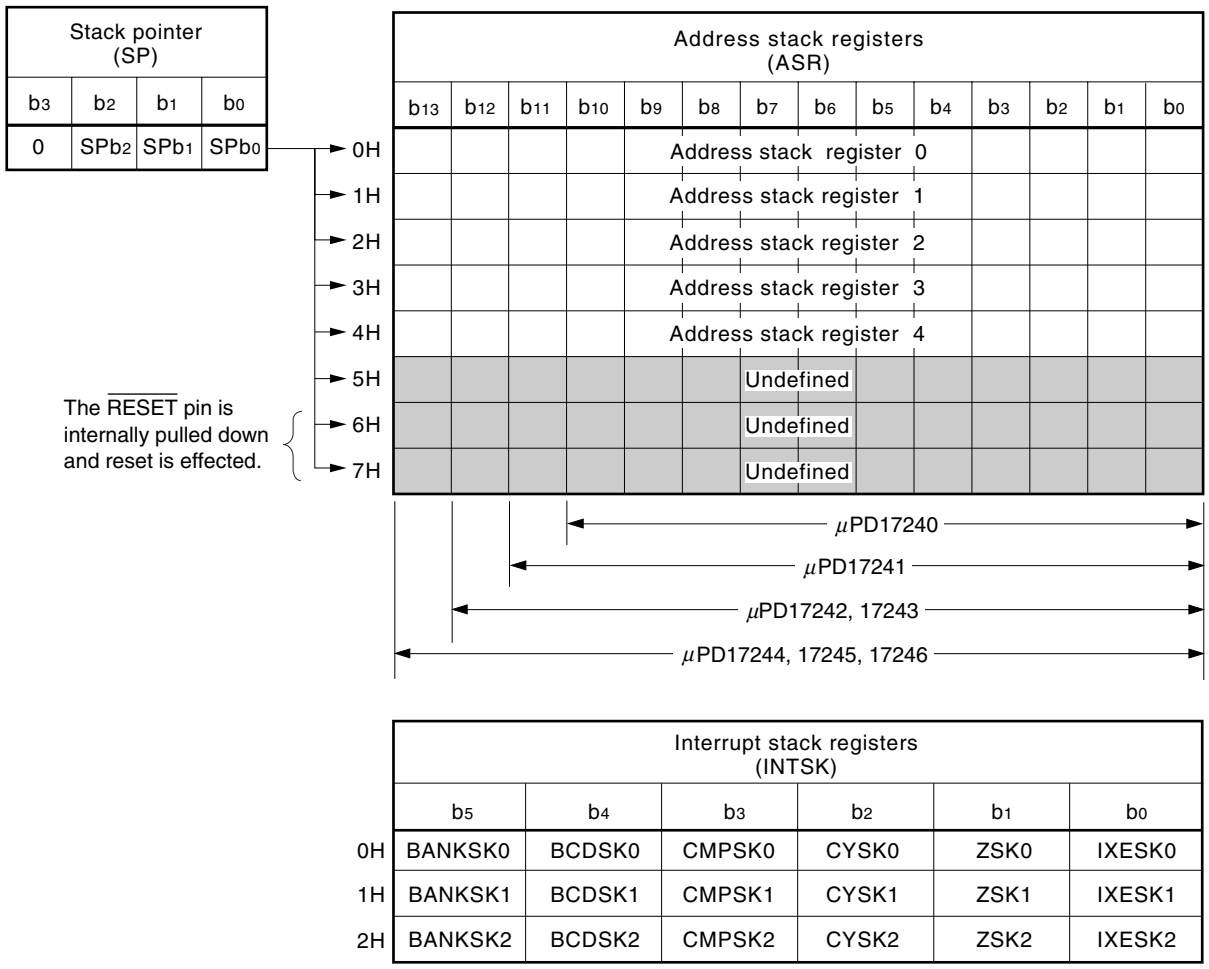
A stack is a register used to save a program return address and the contents of system registers (to be described later) when a subroutine is called or when an interrupt is acknowledged.

2.3.1 Stack configuration

Figure 2-5 shows the stack configuration.

A stack consists of a stack pointer (a 4-bit binary counter, the highest bit fixed to 0), five 11-bit (μPD17240)/12-bit (μPD17241)/13-bit (μPD17242, 17243)/14-bit (μPD17244, 17245, 17246) address stack registers, and three 6-bit interrupt stack registers.

Figure 2-5. Stack Configuration



Interrupt stack registers (INTSK)						
	b5	b4	b3	b2	b1	b0
0H	BANKSK0	BCDSK0	CMPSK0	CYSK0	ZSK0	IXESK0
1H	BANKSK1	BCDSK1	CMPSK1	CYSK1	ZSK1	IXESK1
2H	BANKSK2	BCDSK2	CMPSK2	CYSK2	ZSK2	IXESK2

2.3.2 Function of stack

The address stack register stores a return address when the subroutine call instruction or table reference instruction (first instruction cycle) is executed or when an interrupt is acknowledged. It also stores the contents of the address registers (ARs) when a stack manipulation instruction (PUSH AR) is executed.

If subroutines or interrupts are nested to more than 5 levels, the $\overline{\text{RESET}}$ pin is internally pulled down and a reset is effected.

The interrupt stack register (INTSK) saves the contents of the bank register (BANK) and program status word (PSWORD) when an interrupt is acknowledged. The saved contents are restored when an interrupt return (RETI) instruction is executed.

INTSK saves data each time an interrupt is acknowledged, but **the data stored first is lost if more than 3 levels of interrupts occur**.

2.3.3 Stack pointer (SP) and interrupt stack pointer

Table 2-3 shows the operations of the stack pointer (SP).

The stack pointer can take eight values, 0H to 7H. Because there are only five stack registers available, however, **the $\overline{\text{RESET}}$ pin is internally pulled down and reset is effected if the value of SP is 6 or greater**.

Table 2-3. Operations of Stack Pointer

Instruction	Value of Stack Pointer (SP)	Counter of Interrupt Stack Register
CALL addr CALL @AR MOVT DBF, @AR (1st instruction cycle) PUSH AR SYSCAL entry	–1	0
When interrupt is acknowledged	–1	–1
RET RETSK MOVT DBF, @AR (2nd instruction cycle) POP AR	+1	0
RETI	+1	+1

2.4 Data Memory (RAM)

The data memory (RAM) stores data for operations and control. It can always be read/written by instructions.

2.4.1 Memory configuration

Figure 2-6 shows the configuration of the data memory (RAM).

The data memory consists of four “banks”: BANK0, BANK1, BANK2, and BANK3.

In each bank, every 4 bits of data are assigned an address. The higher 3 bits of the address indicate a “row address” and the lower 4 bits of the address indicate a “column address”. For example, a data memory location indicated by row address 1H and column address 0AH is termed a data memory location at address 1AH. Each address stores data of 4 bits (= 1 nibble).

In addition, the data memory is divided into the following six functional blocks.

(1) System register (SYSREG)

A system register (SYSREG) is resident on addresses 74H to 7FH (12 nibbles) of each bank. In other words, each bank has the same system register at its addresses 74H to 7FH.

(2) Data buffer (DBF)

A data buffer is resident on addresses 0CH to 0FH (4 nibbles) of bank 0 of the data memory.
The reset value is 0320H.

(3) General register (GR)

A general register is resident on any row (16 nibbles) of any bank of the data memory.
The row address of the general register is pointed to by the general register pointer (RP) in the system register (SYSREG).

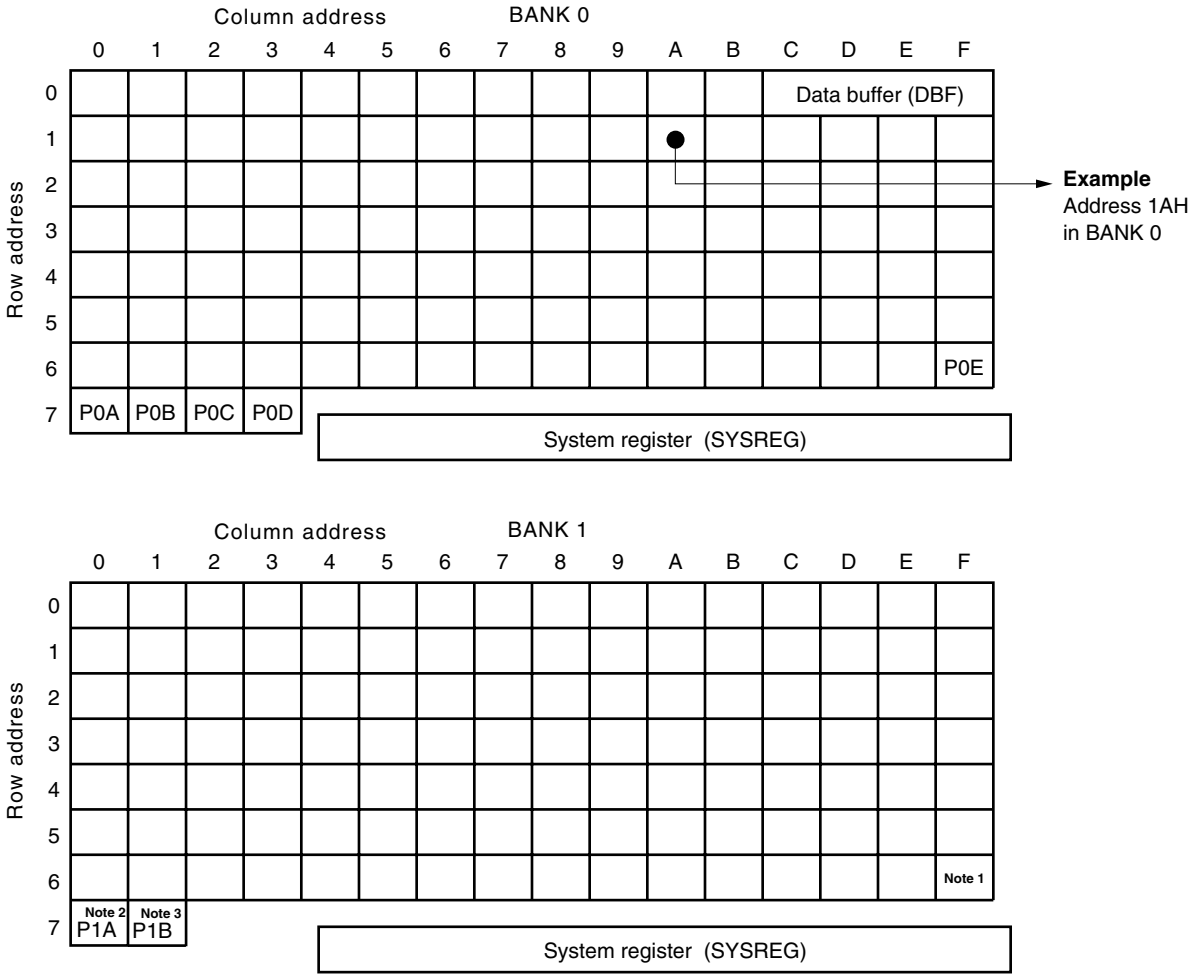
(4) Port register

A port data register is resident on addresses 6FH, and 70H to 73H of BANK0 and addresses 70H and 71H of BANK1 (7 nibbles) of the data memory.
No data can be written to or read from addresses 72H and 73H of BANK1 and addresses 70H to 73H of BANK2 or BANK3.

(5) General-purpose data memory

The general-purpose data memory area is an area of the data memory excluding the system register area, and the port register area. This memory area has a total of 447 nibbles (111 nibbles in BANK0 and 336 nibbles (112 nibbles × 3) in BANK1 to BANK3).

Figure 2-6. Configuration of Data Memory (1/2)



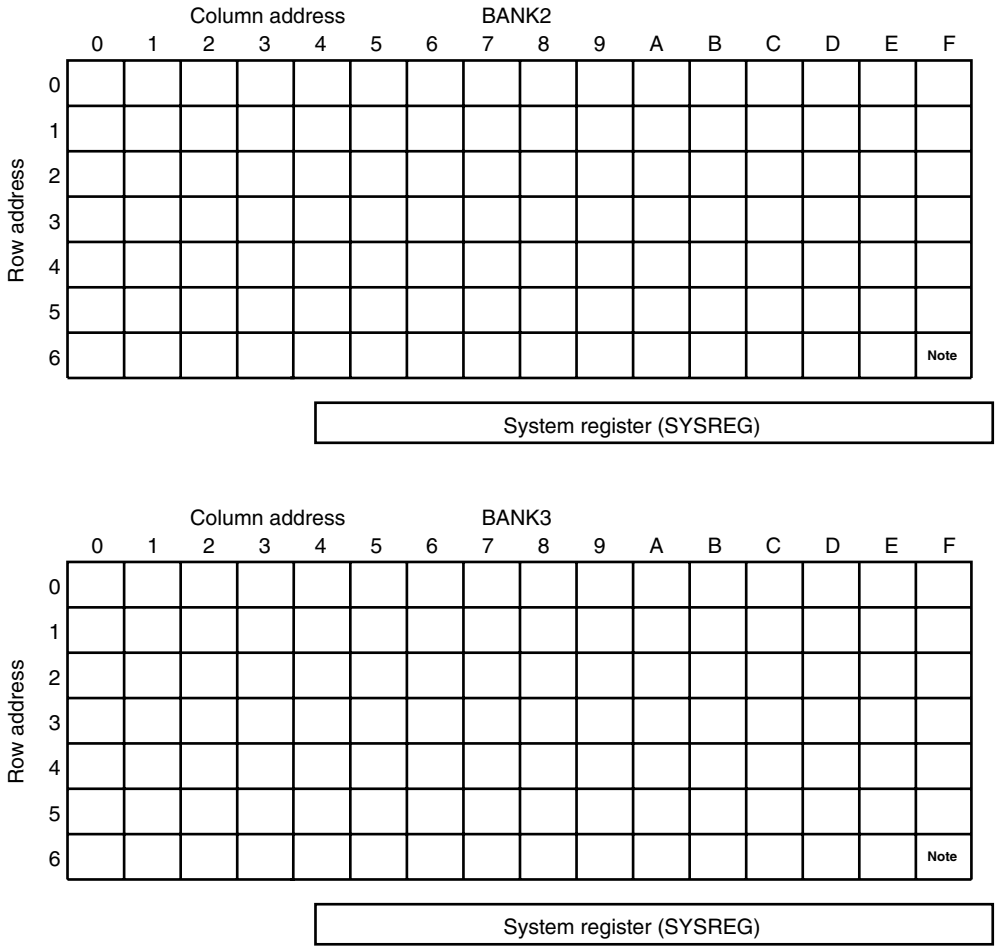
Notes 1. Address 6FH of BANK1 can be used as a general-purpose data memory area.

2. Bits 0 to 2 of address 70H of BANK1 are used. Bit 3 is fixed to 0.

3. Only bit 0 of address 71H of BANK1 is used. Bits 1 to 3 are fixed to 0.

Caution No data can be written to or read from addresses 72H and 73H of BANK1.

Figure 2-6. Configuration of Data Memory (2/2)



Note Address 6FH of BANK2, BANK3 can be used as a general-purpose data memory area.

Caution No data can be written to or read from addresses 70H to 73H of BANK2 and BANK3.

2.4.2 System registers (SYSREG)

The system registers are registers that are directly related to control of the CPU. These registers are mapped to addresses 74H to 7FH on the data memory and can be referenced regardless of bank specification.

The system registers include the following registers.

- Address registers (AR0 to AR3)
- Window register (WR)
- Bank register (BANK)
- Memory pointer enable flag (MPE)
- Memory pointers (MPH, MPL)
- Index registers (IXH, IXM, IXL)
- General register pointers (RPH, RPL)
- Program status word (PSWORD)

Figure 2-7. Configuration of System Registers

Address	74H	75H	76H	77H	78H	79H	7AH	7BH	7CH	7DH	7EH	7FH
Name	Address register (AR)				Window register (WR)	Bank register (BANK)	Index register (IX) / Data memory row address pointer (MP)			General register pointer (RP)		Program status word (PSWORD)
Symbol	AR 3	AR 2	AR 1	AR 0	WR	BANK	IXH / MPH	IXM / MPL	IXL	RPH	RPL	PSW
Bit	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0
Data	0 0 ◄ (AR) (μPD17244,17245,17246) ►				(WR)	(BANK) 0 0	MPE 0 0	(IX) / (MP)		0 0 (RP)		BCD CMP CY Z IXE
	0 0 0 ◄ (AR) (μPD17242,17243) ►											
	0 0 0 0 ◄ (AR) (μPD17241) ►											
	0 0 0 0 0 ◄ (AR) (μPD17240) ►											
Initial value at reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	Undefined	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

2.4.3 General register (GR)

A general register is a register on the data memory and used for arithmetic operations and transfer of data to and from the data memory.

(1) Configuration of general register

Figure 2-8 shows the configuration of the general registers.

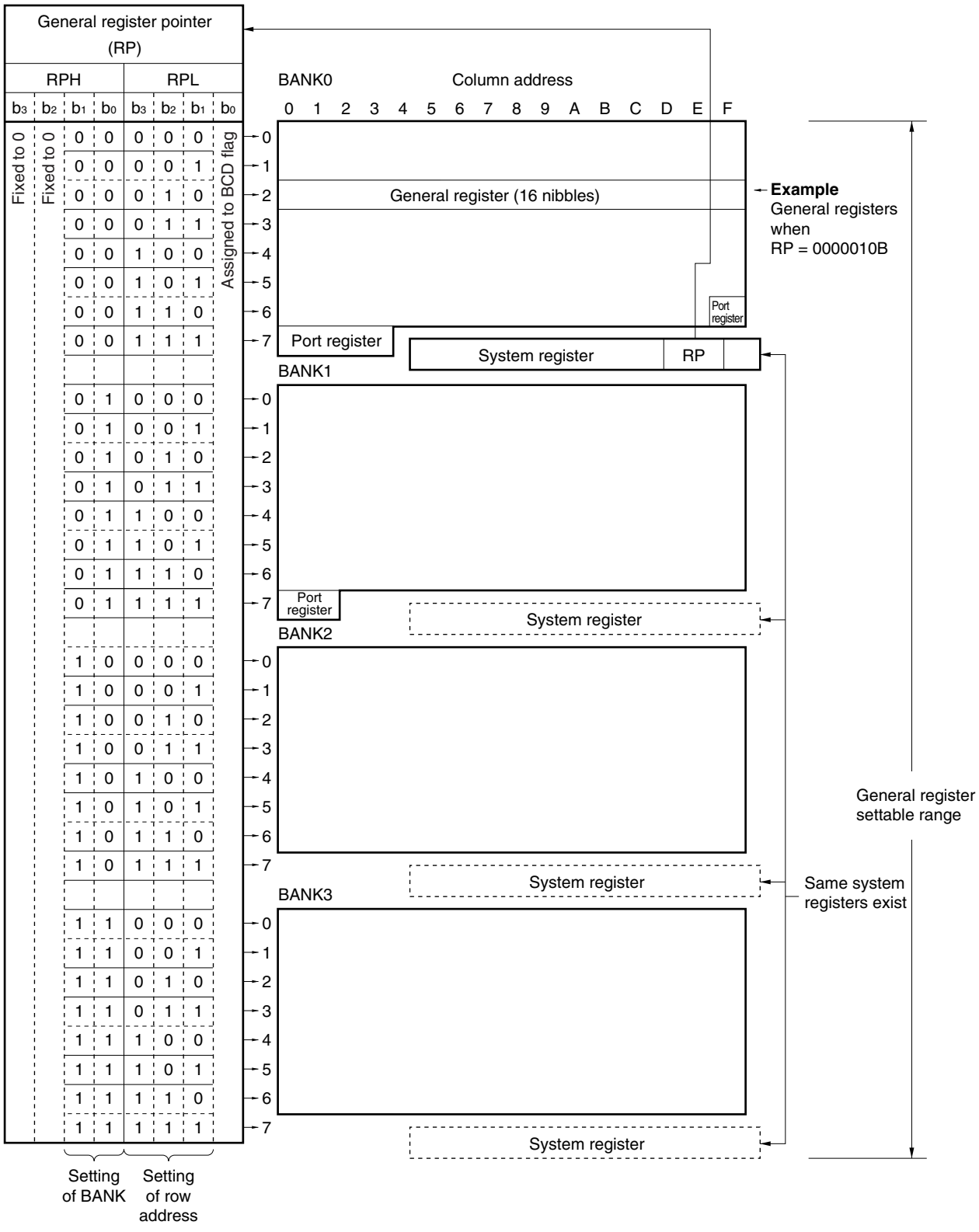
A general register occupies 16 nibbles (16 × 4 bits) on a selected row address of the data memory as shown in Figure 2-8.

The row address is selected by the general register pointer (RP) of the system register. Five bits of RP are valid. Of these, the lower 3 bits (bits 1 to 3 of RPL) are used to set a row address, and the higher 2 bits (bits 0 and 1 of RPH) are used to set a bank. The data memory that can be used as general registers is at row addresses 0H to 7H in BANK0 to 4.

(2) Functions of general registers

A general register enables an arithmetic operation and data transfer between the data memory and a selected general register by a single instruction. As a general register is a part of the data memory, you can say that the general registers enable arithmetic operations and data transfer between two locations of the data memory. Similarly, the general registers can be accessed by a data memory manipulation instruction as they are a part of the data memory.

Figure 2-8. Configuration of General Registers



2.4.4 Data buffer (DBF)

The data buffer on addresses 0CH to 0FH of the data memory is used for data transfer to and from peripheral hardware and for storage of data during table referencing.

(1) Functions of the data buffer

The data buffer has two major functions: a function to transfer data to and from hardware and a function to read constant data from the program memory (for table referencing). Figure 2-9 shows the relationship between the data buffer and peripheral hardware.

Figure 2-9. Data Buffer and Peripheral Hardware

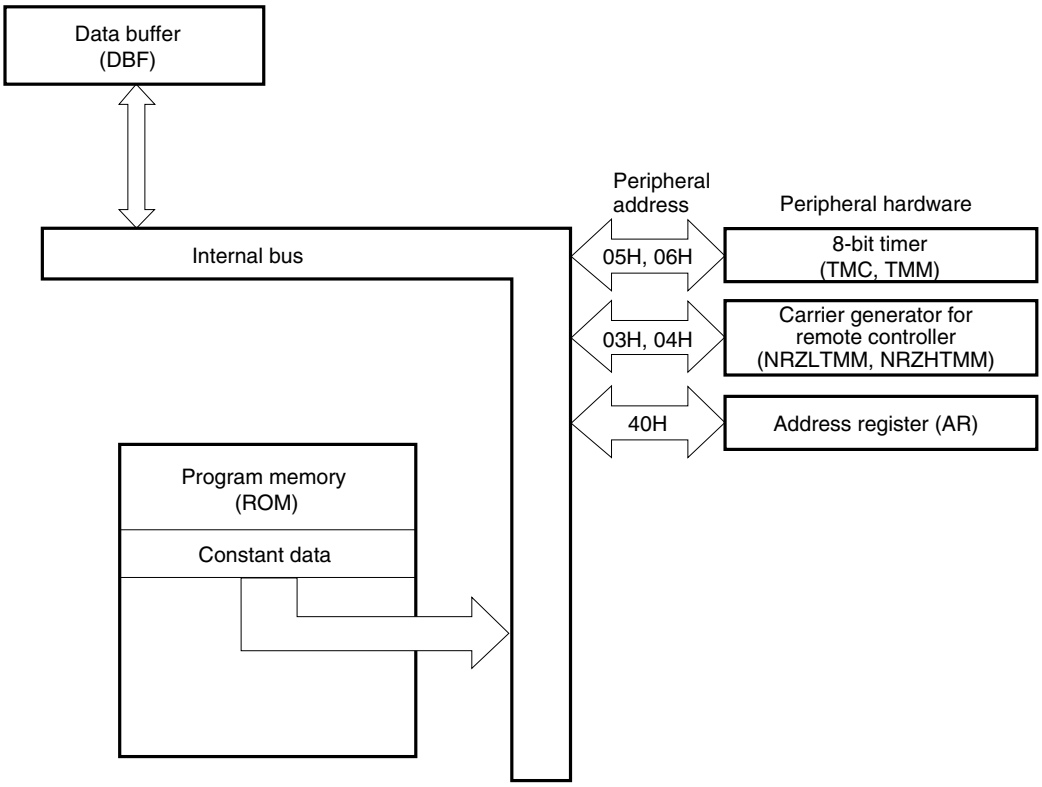


Table 2-4. Relationship Between Peripheral Hardware and Data Buffer

Peripheral Hardware	Peripheral Register Transferring Data with Data Buffer				
	Name	Symbol	Peripheral address	Data buffer used	PUT/GET
8-bit timer	8-bit counter	TMC	05H	DBF0, DBF1	GET only
	8-bit modulo register	TMM	06H	DBF0, DBF1	PUT only
Remote controller carrier generator	NRZ low-level timer modulo register	NRZLTMM	03H	DBF0, DBF1	PUT GET
	NRZ high-level timer modulo register	NRZHTMM	04H	DBF0, DBF1	PUT GET
Address register	Address register	AR	40H	DBF0 to DBF3	PUT[Note 1] GET[Note 2]

Notes 1. In the μPD17240: Bits 0 to 3 of AR3 and bit 3 of AR2 are arbitrary values
In the μPD17241: Bits 0 to 3 of AR3 are arbitrary values
In the μPD17242, 17243: Bits 1 to 3 of AR3 are arbitrary values
In the μPD17244, 17245, 17246: Bits 2 to 3 of AR3 are arbitrary values

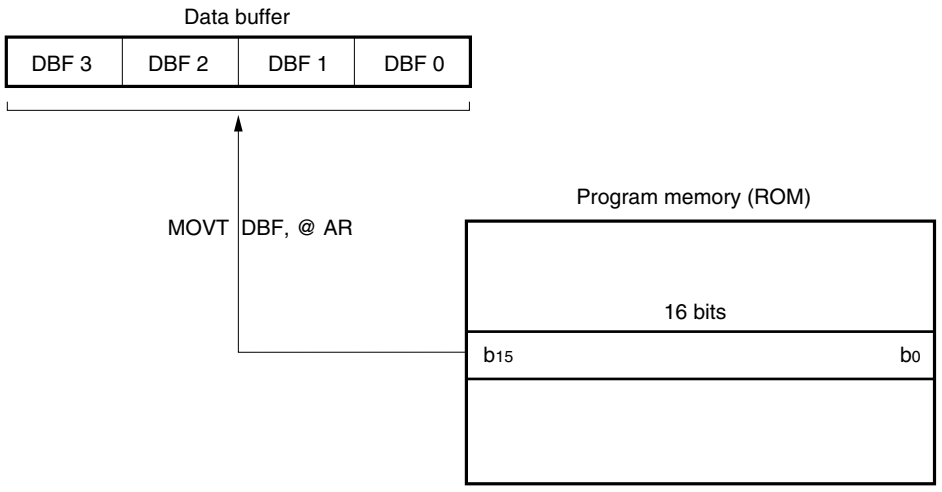
2. In the μPD17240: Bits 0 to 3 of AR3 and bit 3 of AR2 are always 0
In the μPD17241: Bits 0 to 3 of AR3 are always 0
In the μPD17242, 17243: Bits 1 to 3 of AR3 are always 0
In the μPD17244, 17245, 17246: Bits 2 to 3 of AR3 are always 0

(2) Table referencing

An MOVT instruction reads constant data from a specified location of the program memory (ROM) and sets it in the data buffer.

The function of the MOVT instruction is explained below.

MOVT DBF, @AR: Reads data from a program memory location pointed to by the address register (AR) and sets it in the data buffer (DBF).



(3) Notes on using data buffer

When transferring data to/from the peripheral hardware via the data buffer, the unused peripheral addresses, write-only peripheral registers (only when executing PUT), and read-only peripheral registers (only when executing GET) must be handled as follows.

- **When device operates**

  Nothing changes even if data is written to a read-only register.

  If an unused address is read, an undefined value is read. Nothing changes even if data is written to that address.

- **Using assembler**

  An error occurs if an instruction is executed to read a write-only register.

  Again, an error occurs if an instruction is executed to write data to a read-only register.

  An error also occurs if an instruction is executed to read or write an unused address.

- **If an in-circuit emulator (IE-17K or IE-17K-ET) is used (when an instruction is executed for patch processing)**

  An undefined value is read if an attempt is made to read the data of a write-only register, but an error does not occur.

  Nothing changes even if data is written to a read-only register, and an error does not occur.

  An undefined value is read if an unused address is read; nothing changes even if data is written to this address. An error does not occur.

2.5 Register File (RF)

The register file mainly consists of registers that set the conditions of the peripheral hardware.

These registers can be controlled by the dedicated instructions PEEK and POKE, and the embedded macro instructions of RA17K, SETn, CLRn, and INITFLG.

2.5.1 Configuration of register file

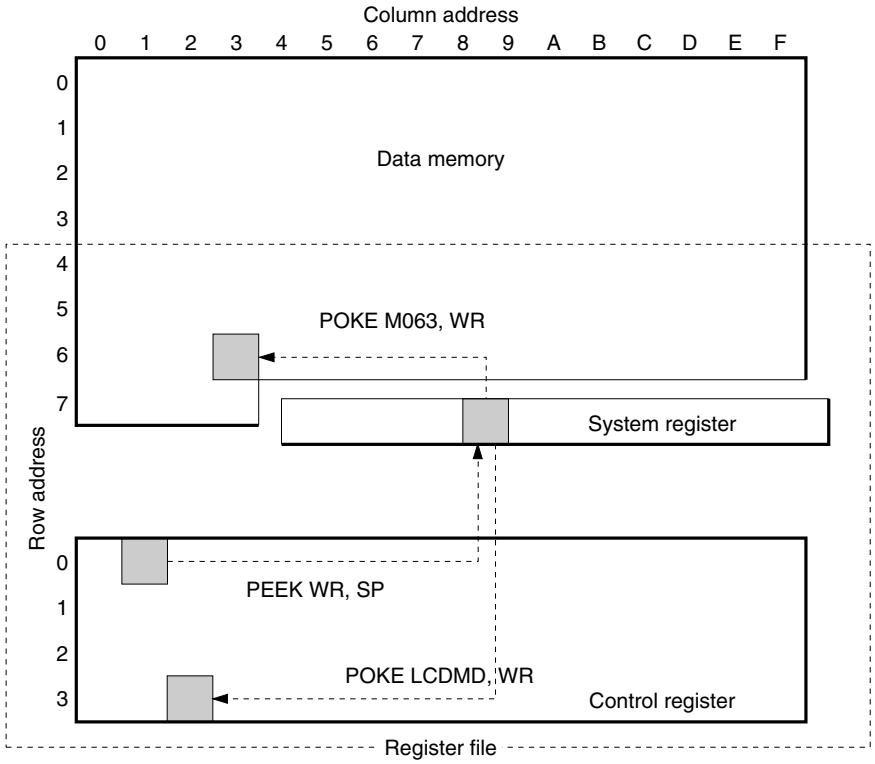
Figure 2-10 shows the configuration of the register file and how the register file is accessed by the PEEK and POKE instructions.

The control registers are controlled by using dedicated instructions PEEK and POKE. Since the control registers are assigned to addresses 00H to 3FH regardless of the bank, the addresses 00H to 3FH of the general-purpose data memory cannot be accessed when the PEEK or POKE instruction is used.

The addresses that can be accessed by the PEEK and POKE instructions are addresses 00H to 3FH of the control registers and 40H to 7FH of the general-purpose data memory. The register file consists of these addresses.

The control registers are assigned to addresses 80H to BFH on the IE-17K to facilitate debugging.

Figure 2-10. Register File Configuration and Register File Access with PEEK or POKE Instructions



2.5.2 Control registers

The control registers consist of a total of 64 nibbles (64 × 4 bits) of addresses 00H to 3FH of the register file.

Of these, however, only 24 nibbles are actually used. The remaining 40 nibbles are unused registers that are prohibited from being read or written.

When the "PEEK WR, rf" instruction is executed, the contents of the register file addressed by "rf" are read to the window register.

When the "POKE rf, WR" instruction is executed, the contents of the window register are written to the register file addressed by "rf".

When using the assembler (RA17K), the macro instructions listed below, which are embedded as flag type symbol manipulation instructions, can be used. The macro instructions allow the contents of the register file to be manipulated in bit units.

For the configuration of the control register, refer to **Figure 12-1 Register File List**.

SETn: Sets flag to "1"
CLRn: Sets flag to "0"
SKTn: Skips if all flags are "1"
SKFn: Skips if all flags are "0"
NOTn: Inverts flag
INITFLG: Initializes flag
INITFLGX: Initializes flag

2.5.3 Notes on using register files

When using the register files, bear in mind the points described below. For details, refer to the **μPD172xx Subseries User's Manual (U12795E)**.

(1) When manipulating control registers (read-only and unused registers)

When manipulating the write-only (W), the read-only (R), and unused control registers by using an assembler or in-circuit emulator, keep in mind the following points.

- **When device operates**

  Nothing changes even if data is written to a read-only register.
  If an unused register is read, an undefined value is read; nothing is changed even if data is written to this register.

- **Using assembler**

  An error occurs if an instruction is executed to read data from a write-only register.
  An error occurs if an instruction is executed to write data to a read-only register.
  An error also occurs if an instruction is executed to read or write an unused address.

- **When an in-circuit emulator (IE-17K or IE-17K-ET) is used (when an instruction is executed for patch processing)**

  An undefined value is read if a write-only register is read, and an error does not occur.
  Nothing changes even if data is written to a read-only register, and an error does not occur.
  An undefined value is read if an unused address is read; nothing changes even if data is written to this address. An error does not occur.

(2) Symbol definition of register file

An error occurs if a register file address is directly specified as a numeral by the operand “rf” of the “PEEK WR, rf” or “POKE rf, WR” instruction if the 17K Series Assembler (RA17K) is being used.

Therefore, the addresses of the register file must be defined in advance as symbols.

To define the addresses of the control registers as symbols, define them as addresses 80H to BFH of BANK0.

The portion of the register file overlapping the data memory (40H to 7FH), however, can be defined as symbols as is.

3. PORTS

3.1 Port 0A ($P0A_0$ to $P0A_3$)

This is a 4-bit input port. Data is read using port register P0A (address 70H of BANK0). This port is a CMOS input port with a pull-up resistor, and can be used as the key return input lines of a key matrix.

In the standby mode, the standby status is released when a low level is input to at least one of these pins.

3.2 Port 0B ($P0B_0$ to $P0B_3$)

This is a 4-bit I/O port which can be set to the input or output mode in 1-bit units by using P0BBIO (address 26H) of the register file.

In the input mode, each bit of this port serves as a CMOS input pin with a pull-up resistor and can be used as a key return input line of a key matrix. In the standby mode, the standby status is released when a low level is input to at least one of these pins.

In the output mode, these pins serve as N-ch open-drain output pins and can be used as the key source lines of a key matrix.

The data input to this port can be read or the data output from this port can be set by using the P0B register (address 71H of BANK0). When this port is read in the output mode, the contents of the output latch are read.

In the input mode, a pull-up resistor of 200 kΩ is connected to each bit of this port. In the output mode, the pull-up resistor is disconnected.

After reset, this port is set to the input mode.

3.3 Port 0C ($P0C_0$ to $P0C_3$)

This is a 4-bit I/O port which can be set to the input or output mode in 4-bit units (group I/O) by using P0CDGIO (bit 2 of address 37H) of the register file.

In the input mode, each bit of this port serves as a CMOS input pin with a pull-up resistor and can be used as a key return input line of a key matrix. In the standby mode, the standby status is released when a low level is input to at least one of these pins.

In the output mode, these pins serve as N-ch open-drain output pins and can be used as the key source lines of a key matrix.

The data input to this port can be read or the data output from this port can be set by using the P0C register (address 72H of BANK0). When this port is read in the output mode, the contents of the output latch are read.

In the input mode, a pull-up resistor of 200 kΩ is connected to each bit of this port. In the output mode, the pull-up resistor is disconnected.

After reset, this port is set to the output mode and outputs a low level.

3.4 Port 0D ($P0D_0$ to $P0D_3$)

This is a 4-bit I/O port which can be set to the input or output mode in 4-bit units (group I/O) by using P0CDGIO (bit 3 of address 37H) of the register file.

In the input mode, each bit of this port serves as a CMOS input pin with a pull-up resistor and can be used as a key return input line of a key matrix. In the standby mode, the standby status is released when a low level is input to at least one of these pins.

In the output mode, these pins serve as N-ch open-drain output pins and can be used as the key source lines of a key matrix.

The data input to this port can be read or the data output from this port can be set by using the P0D register (address 73H of BANK0). When this port is read in the output mode, the contents of the output latch are read.

In the input mode, a pull-up resistor of 200 kΩ is connected to each bit of this port. In the output mode, the pull-up resistor is disconnected.

After reset, this port is set to the output mode and outputs a low level.

3.5 Port 0E (P0E0 to P0E3)

This is a 4-bit I/O port. The input mode or output mode and whether a key matrix is used or not can be set for this port in 1-bit units.

The input and output modes of this port are selected by using P0EBIO (address 27H) of the register file.

Whether a key matrix is used or not is specified by P0EKEY (address 16H) of the register file.

If this port is set to the input mode when a key matrix is used, it functions as a CMOS input port with a pull-up resistor and can be used to input key return signals. If one of the pins of this port goes low, the standby mode is released.

If this port is set to the output mode when a key matrix is used, it functions as an N-ch open-drain output port and can be used to output key matrix signals.

If this port is set to the input mode when a key matrix is not used, it functions as a CMOS input port to/from which a pull-up resistor can be connected/disconnected in 1-bit units, by using P0EBPU (address 17H) of the register file (if a pull-up resistor is connected, it is not disconnected even if the output mode is set). At this time, the standby mode is not released.

If this port is set to the output mode when a key matrix is not used, it functions as a high-current CMOS output port.

To read the input data from this port or set output data to it, use the P0E register (address 6FH of BANK0). When this port is read in the output mode, the contents of the output latch are read.

After reset, this port is set to the input mode (a key matrix is not used and a resistor is not connected).

3.6 Port 1A (P1A0 to P1A2)

These pins constitute a 3-bit I/O port that can be set in the input or output mode in 1-bit units.

If this port is set to the input mode when a key matrix is used, it functions as a CMOS input port and can be used to input key return signals. At this time, whether a resistor is connected to this port and the standby mode release condition (whether it is released when this port is high or low) can be selected.

1. If connection of a resistor is specified and if it is specified that the standby mode is released when this port goes low
   ... A pull-up resistor is connected. If a low level is input to the set key, the standby mode is released.
2. If connection of a resistor is specified and if it is specified that the standby mode is released when this port goes high
   ... A pull-down resistor is connected. If a high level is input to the set key, the standby mode is released.
3. If connection of a resistor is not specified and if it is specified that the standby mode is released when this port goes low (or high)
   ... No resistor is connected. If a low (or high) level is input to the set key, the standby mode is released.

If this port is set to the output mode when a key matrix is used, it functions as an N-ch open-drain output port and can be used to output key matrix signals.

If this port is set to the input mode when a key matrix is not used, it functions as a CMOS input port. Connection of a resistor to this port and whether a pull-up or pull-down resistor is connected to the port can be selected in 1-bit units. At this time, the standby mode is not released.

If this port is set to the output mode when a key matrix is not used, it functions as a high-current CMOS output port.

To set this port to the input mode or output mode, use P1ABIO (address 25H) of the register file. To specify whether a key matrix is used or not, use P1AKEY (address 06H) of the register file. To specify whether a resistor is connected, use P1ABPU (address 07H) of the register file. To specify the standby mode release condition (to specify whether a pull-down or pull-up resistor is connected when a key matrix is not used), use P1AHL (address 05H) of the register file.

Use the P1A register (address 70H of BANK1) to read the input data from this port or set output data to it. When this port is read in the output mode, the contents of the output latch are read.

After reset, this port is set to the input mode (a key matrix is not used and a resistor is not connected).

3.7 Port 1B (P1B0)

The P1B0 pin functions alternately as the INT pin. To use the P1B0 pin, set INTSEL (bit 1 of address 1FH) of the register file to 0.

The P1B0 pin functions as a 1-bit CMOS input port.

This port can be used to input a key return signal when a key matrix is used. At this time, whether a resistor is connected to this port and the standby mode release condition (whether it is released when this pin is high or low) can be selected.

1. If connection of a resistor is specified and if it is specified that the standby mode is released when this pin goes low
   ... A pull-up resistor is connected. If a low level is input to P1B0, the standby mode is released.
2. If connection of a resistor is specified and if it is specified that the standby mode is released when this pin goes high
   ... A pull-down resistor is connected. If a high level is input to P1B0, the standby mode is released.
3. If connection of a resistor is not specified and if it is specified that the standby mode is released when this pin goes low (or high)
   ... No resistor is connected. If a low (or high) level is input to P1B0, the standby mode is released.

If a key matrix is not used, whether a resistor is connected and whether a pull-up or pull-down resistor is connected can be selected. At this time, the standby mode is not released.

To specify whether a resistor is connected, use P1BPU0 (bit 0 of address 05H) of the register file. To specify whether a key matrix is used or not, use P1BKEY0 (bit 1 of address 05H) of the register file. To specify a standby condition (to specify whether a pull-down or pull-up resistor is connected when a key matrix is not used), use P1BHL0 (bit 2 of address 05H) of the register file.

Use the P1B register (address 71H of BANK1) to read the input data.

After reset, the P1B0 pin is selected and functions as an input port (a key matrix is not used and a resistor is not connected).

3.8 INT Pin

The INT pin functions alternately as the $P1B_0$ pin. To use the INT pin, set INTSEL (bit 1 of address 1FH) of the register file to 1.

This pin inputs an external interrupt request signal. The IRQ flag (RF: bit 0 of address 3EH) is set at either the rising or falling edge of the signal input to this pin.

The status of this pin can be read by using the INT flag (RF: bit 0 of address 0FH). When a high level is input to the pin, the INT flag is set to "1"; when a low level is input, the flag is reset to "0" (refer to **8.2.1 INT**).

Table 3-1. Relationship Between Port Register and Each Pin

Bank	Address	Target Port	Bit		Output Format	Read Contents		Written Contents		After Reset
						Input mode	Output mode	Input mode	Output mode	
0	70H	Port 0A	b_3	P0A3	Input	Pin status	–	–	–	Input mode (with pull-up resistor)
			b_2	P0A2						
			b_1	P0A1						
			b_0	P0A0						
	71H	Port 0B	b_3	P0B3	N-ch open drain		Output latch	Output latch	Output latch	
			b_2	P0B2						
			b_1	P0B1						
			b_0	P0B0						
	72H	Port 0C	b_3	P0C3						Output mode (low-level output)
			b_2	P0C2						
			b_1	P0C1						
			b_0	P0C0						
	73H	Port 0D	b_3	P0D3						
			b_2	P0D2						
			b_1	P0D1						
			b_0	P0D0						
	6FH	Port 0E	b_3	P0E3	CMOS push-pull or N-ch open drain					Input mode (when key matrix not used and no pull-up resistor connected)
			b_2	P0E2						
			b_1	P0E1						
			b_0	P0E0						
1	70H	Port 1A	b_2	P1A2	CMOS push-pull or N-ch open drain					Input mode (when key matrix not used and no resistor connected)
			b_1	P1A1						
			b_0	P1A0						
	71H	Port 1B	b_0	P1B0	Input		–	–	–	Input mode (when key matrix not used and no resistor connected)

3.9 Switching Bit I/O (Port 0B, 0E, 1A)

An I/O that can be set to the input or output mode in bit units is called a bit I/O. P0B, P0E, and P1A are bit I/O ports, which can be set in the input or output mode in bit units by the register file shown below. When the mode is changed from input to output, the P0B, P0E, and P1A output latch contents are output to the port lines as soon as the mode has been changed.

3	2	1	0	Address	After reset	R/W
P0BBIO3	P0BBIO2	P0BBIO1	P0BBIO0	RF: 26H	0H	R/W

P0BBIO0	Sets $P0B_0$ input/output mode
0	Sets $P0B_0$ in input mode
1	Sets $P0B_0$ in output mode

P0BBIO1	Sets $P0B_1$ input/output mode
0	Sets $P0B_1$ in input mode
1	Sets $P0B_1$ in output mode

P0BBIO2	Sets $P0B_2$ input/output mode
0	Sets $P0B_2$ in input mode
1	Sets $P0B_2$ in output mode

P0BBIO3	Sets $P0B_3$ input/output mode
0	Sets $P0B_3$ in input mode
1	Sets $P0B_3$ in output mode

3	2	1	0	Address	After reset	R/W
P0EBIO3	P0EBIO2	P0EBIO1	P0EBIO0	RF: 27H	0H	R/W

P0EBIO0	Sets $P0E_0$ input/output mode
0	Sets $P0E_0$ in input mode
1	Sets $P0E_0$ in output mode

P0EBIO1	Sets $P0E_1$ input/output mode
0	Sets $P0E_1$ in input mode
1	Sets $P0E_1$ in output mode

P0EBIO2	Sets $P0E_2$ input/output mode
0	Sets $P0E_2$ in input mode
1	Sets $P0E_2$ in output mode

P0EBIO3	Sets $P0E_3$ input/output mode
0	Sets $P0E_3$ in input mode
1	Sets $P0E_3$ in output mode

3	2	1	0	Address	After reset	R/W
0	P1ABIO2	P1ABIO1	P1ABIO0	RF: 25H	0H	R/W

P1ABIO0	Sets $P1A_0$ input/output mode
0	Sets $P1A_0$ in input mode
1	Sets $P1A_0$ in output mode

P1ABIO1	Sets $P1A_1$ input/output mode
0	Sets $P1A_1$ in input mode
1	Sets $P1A_1$ in output mode

P1ABIO2	Sets $P1A_2$ input/output mode
0	Sets $P1A_2$ in input mode
1	Sets $P1A_2$ in output mode

3.10 Selecting I/O Mode of Group I/O (Port 0C, 0D)

An I/O that is set to the input or output mode in 4-bit units is called a group I/O. P0C and P0D can be used as group I/O ports. The input and output modes of these ports are selected by using the following register file. If the mode is changed from input to output, the contents of the port register are output to the respective ports as soon as the mode has been changed.

3	2	1	0	Address	After reset	R/W
P0DGIO	P0CGIO	0	0	RF: 37H	CH	R/W

P0CGIO	I/O mode of $P0C_0$ to $P0C_3$
0	Sets $P0C_0$ to $P0C_3$ in input mode
1	Sets $P0C_0$ to $P0C_3$ in output mode

P0DGIO	I/O mode of $P0D_0$ to $P0D_3$
0	Sets $P0D_0$ to $P0D_3$ in input mode
1	Sets $P0D_0$ to $P0D_3$ in output mode

3.11 Selecting Whether Key Matrix Is Used or Not (Port 0E, 1A)

By using the following register file, whether P0E and P1A are used for a key matrix can be selected in bit units.

3	2	1	0	Address	After reset	R/W
0	P1AKEY2	P1AKEY1	P1AKEY0	RF: 06H	0H	R/W

P1AKEY0	Selects whether $P1A_0$ is used for key matrix or not
0	$P1A_0$ not used for key matrix
1	$P1A_0$ used for key matrix

P1AKEY1	Selects whether $P1A_1$ is used for key matrix or not
0	$P1A_1$ not used for key matrix
1	$P1A_1$ used for key matrix

P1AKEY2	Selects whether $P1A_2$ is used for key matrix or not
0	$P1A_2$ not used for key matrix
1	$P1A_2$ used for key matrix

3	2	1	0	Address	After reset	R/W
P0EKEY3	P0EKEY2	P0EKEY1	P0EKEY0	RF: 16H	0H	R/W

P0EKEY0	Selects whether $P0E_0$ is used for key matrix or not
0	$P0E_0$ not used for key matrix
1	$P0E_0$ used for key matrix

P0EKEY1	Selects whether $P0E_1$ is used for key matrix or not
0	$P0E_1$ not used for key matrix
1	$P0E_1$ used for key matrix

P0EKEY2	Selects whether $P0E_2$ is used for key matrix or not
0	$P0E_2$ not used for key matrix
1	$P0E_2$ used for key matrix

P0EKEY3	Selects whether $P0E_3$ is used for key matrix or not
0	$P0E_3$ not used for key matrix
1	$P0E_3$ used for key matrix

3.12 Specifying Resistor Connection (Port 0E, 1A)

(1) Port 0E

If a key matrix is not used, whether or not a pull-up resistor is connected to port P0E can be specified in 1-bit units by using the following registers of the register file[Note].

3	2	1	0	Address	After reset	R/W
P0EBPU3	P0EBPU2	P0EBPU1	P0EBPU0	RF: 17H	0H	R/W

P0EBPU0	Connects pull-up resistor to $P0E_0$
0	Not connected
1	Connected

P0EBPU1	Connects pull-up resistor to $P0E_1$
0	Not connected
1	Connected

P0EBPU2	Connects pull-up resistor to $P0E_2$
0	Not connected
1	Connected

P0EBPU3	Connects pull-up resistor to $P0E_3$
0	Not connected
1	Connected

Note To disconnect the pull-up resistor in the output mode, clear the corresponding bit of the P0EBPU register.

(2) Port 1A

Whether a resistor is connected to each bit of port P1A when a key matrix is not used can be specified in 1-bit units by using the following register file[Note].

To connect a resistor, select whether a pull-down or pull-up resistor is to be connected, by using P1AHL (address 05H) of the register file.

★

3	2	1	0	Address	After reset	R/W
0	P1ABPU2	P1ABPU1	P1ABPU0	RF: 07H	0H	R/W

P1ABPU0	Connects resistor to $P1A_0$
0	Not connected
1	Connected

P1ABPU1	Connects resistor to $P1A_1$
0	Not connected
1	Connected

P1ABPU2	Connects resistor to $P1A_2$
0	Not connected
1	Connected

Note To disconnect the resistor in the output mode, clear the corresponding bit of the P1ABPU register.

3.13 Selecting Standby Mode Release Condition and Whether Pull-Up or Pull-Down Resistor Is Connected (Port 1A)

The standby mode release condition and whether a pull-up or pull-down resistor[Note] is connected to P1A can be specified in 1-bit units by using the following register file.

★ **Note** Specify whether a resistor is connected or not by using P1ABPU (address 07H) of the register file.

(1) When key matrix is used (P1AKEYn = 1)

3	2	1	0	Address	After reset	R/W
0	P1AHL2	P1AHL1	P1AHL0	RF: 05H	0H	R/W

P1AHL0	Connects pull-down/pull-up resistor to $P1A_0$ and selects standby mode release condition	
	Resistor is connected (P1ABPU0 = 1)	Resistor is not connected (P1ABPU0 = 0)
0	Pull-up resistor	No resistor
	Standby mode is released when a low level is input to P1A.	
1	Pull-down resistor	No resistor
	Standby mode is released when a high level is input to P1A.	

P1AHL1	Connects pull-down/pull-up resistor to $P1A_1$ and selects standby mode release condition	
	Resistor is connected (P1ABPU1 = 1)	Resistor is not connected (P1ABPU1 = 0)
0	Pull-up resistor	No resistor
	Standby mode is released when a low level is input to P1A.	
1	Pull-down resistor	No resistor
	Standby mode is released when a high level is input to P1A.	

P1AHL2	Connects pull-down/pull-up resistor to $P1A_2$ and selects standby mode release condition	
	Resistor is connected (P1ABPU2 = 1)	Resistor is not connected (P1ABPU2 = 0)
0	Pull-up resistor	No resistor
	Standby mode is released when a low level is input to P1A.	
1	Pull-down resistor	No resistor
	Standby mode is released when a high level is input to P1A.	

Remark P1AKEY: Address 06H of register file
★ P1ABPU: Address 07H of register file
n = 0 to 2

(2) When key matrix is not used (P1AKEYn = 0)

3	2	1	0	Address	After reset	R/W
0	P1AHL2	P1AHL1	P1AHL0	RF: 05H	0H	R/W

P1AHL0	Connects pull-down/pull-up resistor to $P1A_0$	
	Resistor is connected (P1ABPU0 = 1)	Resistor is not connected (P1ABPU0 = 0)
0	Pull-up resistor	No resistor
1	Pull-down resistor	

P1AHL1	Connects pull-down/pull-up resistor to $P1A_1$	
	Resistor is connected (P1ABPU1 = 1)	Resistor is not connected (P1ABPU1 = 0)
0	Pull-up resistor	No resistor
1	Pull-down resistor	

P1AHL2	Connects pull-down/pull-up resistor to $P1A_2$	
	Resistor is connected (P1ABPU2 = 1)	Resistor is not connected (P1ABPU2 = 0)
0	Pull-up resistor	No resistor
1	Pull-down resistor	

Caution The standby mode is not released by P1A when a key matrix is not used.

Remark P1AKEY: Address 06H of register file
★ P1ABPU: Address 07H of register file
n = 0 to 2

3.14 Selecting Whether Key Matrix Is Used, Standby Mode Release Condition, and Whether Pull-Up or Pull-Down Resistor Is Connected (Port 1B)

Whether a key matrix is used or not, whether a resistor is connected to P1B or not, whether a pull-up or pull-down resistor is connected, and the standby mode release condition can be specified by using the following register file.

3	2	1	0	Address	After reset	R/W
0	P1BHL0	P1BKEY0	P1BPU0	RF: 15H	0H	R/W

P1BPU0	Connects resistor to $P1B_0$
0	Not connected
1	Connected

P1BKEY0	Selects whether $P1B_0$ is used for key matrix or not
0	$P1B_0$ not used for key matrix
1	$P1B_0$ used for key matrix

• When key matrix is used (P1BKEY0 = 1)

P1BHL0	Connects pull-down/pull-up resistor to $P1B_0$ and selects standby mode release condition	
	Resistor is connected (P1BPU0 = 1)	Resistor is not connected (P1BPU0 = 0)
0	Pull-up resistor	No resistor
	Standby mode is released when a low level is input to P1B.	
1	Pull-down resistor	No resistor
	Standby mode is released when a high level is input to P1B.	

• When key matrix is not used (P1BKEY0 = 0)

P1BHL0	Connects pull-down/pull-up resistor to $P1B_0$	
	Resistor is connected (P1BPU0 = 1)	Resistor is not connected (P1BPU0 = 0)
0	Pull-up resistor	No resistor
1	Pull-down resistor	

Caution The standby mode is not released by P1B when a key matrix is not used.

4. CLOCK GENERATOR

4.1 Instruction Execution Time (CPU Clock) Selection

The μPD17246 is equipped with a clock oscillator that supplies clocks to the CPU and peripheral hardware. Instruction execution time can be changed in two steps (normal mode and high-speed mode) without changing the oscillation frequency.

To change the instruction execution time, change the mode of SYSCK (RF: address 02H) of the register file by using the POKE instruction.

Note, that the mode is actually only changed when the instruction next to the POKE instruction has been executed.

When using the high-speed mode, pay attention to the supply voltage. (Refer to **14. ELECTRICAL SPECIFICATIONS**.)

After reset, the normal mode is set.

3	2	1	0	Address	After reset	R/W
0	0	0	SYSCK	RF: 02H	0H	R/W

SYSCK	Selects instruction execution time
0	Normal mode 32/fx (8 μs)
1	High-speed mode 16/fx (4 μs)

Values in parentheses apply to operation when the system clock fx = 4 MHz.

5. 8-BIT TIMER AND REMOTE CONTROLLER CARRIER GENERATOR

The μPD17246 is equipped with an 8-bit timer, which is mainly used to generate the leader pulse of the remote controller signal and to output codes.

5.1 Configuration of 8-Bit Timer (with Modulo Function)

Figure 5-1 shows the configuration of the 8-bit timer.

As shown in this figure, the 8-bit timer consists of an 8-bit counter (TMC), an 8-bit modulo register (TMM), a comparator that compares the value of the timer with the value of the modulo register, and a selector that selects the operation clock of the 8-bit timer.

To start/stop the 8-bit timer, and to reset the 8-bit counter, TMEN (address 33H, bit 3) and TMRES (address 33H, bit 2) of the register file are used. To select the operation clock of the 8-bit timer, use TMCK1 (address 33H, bit 1) and TMCK0 (address 33H, bit 0) of the register file.

The value of the 8-bit counter is read by using the GET instruction through the DBF (data buffer). No value can be set to the 8-bit counter. A value is set to the modulo register by using the PUT instruction through DBF. The value of the modulo register cannot be read.

When the value of the counter matches with that of the modulo register, an interrupt flag (IRQTM: address 3FH, bit 0) of the register file is set.

TMC

7	6	5	4	3	2	1	0
8-bit counter							

Address	After reset	R/W
Peripheral register: 05H	00H	R

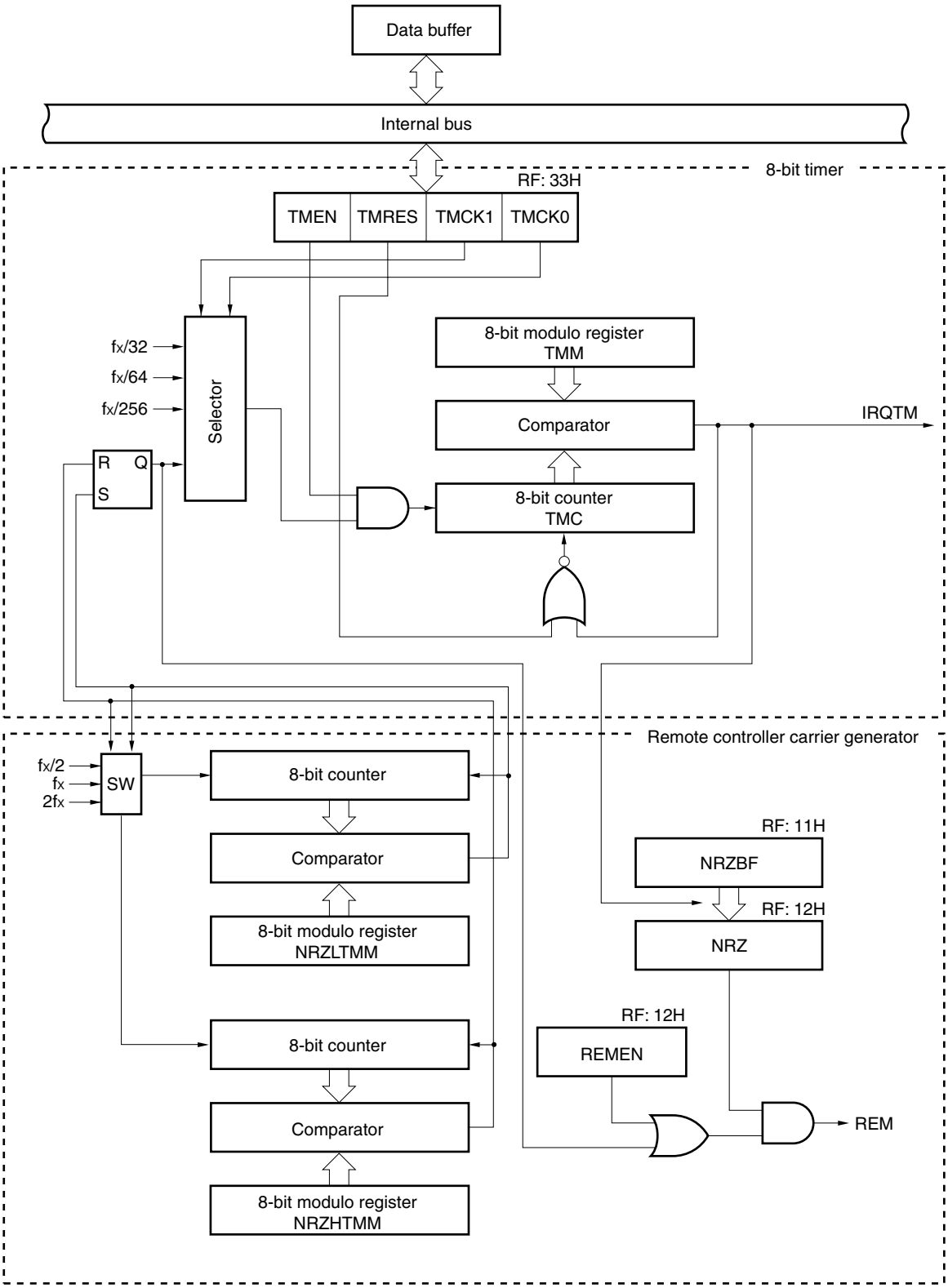
TMM

7	6	5	4	3	2	1	0
8-bit modulo register							

Address	After reset	R/W
Peripheral register: 06H	FFH	W

Caution Do not clear TMM to 0 (IRQTM is not set).

Figure 5-1. Configuration of 8-Bit Timer and Remote Controller Carrier Generator



Remark TMM, TMC, NRZLTMM, and NRZHTMM are peripheral registers.

5.2 Function of 8-Bit Timer (with Modulo Function)

3	2	1	0	Address	After reset	R/W
TMEN	TMRES	TMCK1	TMCK0	RF: 33H	8H[Note 1]	R/W[Note 2]

TMCK1	TMCK0	8-bit timer clock source selection
0	0	Count clock: f_X/32 (measurable time range: 8 μs to 2.048 ms, resolution: 8 μs (error: +8 μs))
0	1	Count clock: f_X/64 (measurable time range: 16 μs to 4.096 ms, resolution: 16 μs (error: +16 μs))
1	0	Count clock: f_X/256 (measurable time range: 64 μs to 16.384 ms, resolution: 64 μs (error: +64 μs))
1	1	Remote controller carrier generator output

Values in parentheses apply to operation when system clock f_X = 4 MHz.

TMRES	8-bit timer reset flag
0	Data read out is always "0"
1	Resets 8-bit counter and IRQTM

TMEN	8-bit timer count enable flag
0	Stops 8-bit timer count operation
1	Enables 8-bit timer count operation (falling edge)

Notes 1. When the STOP mode is released, bit 3 must be set.

2. Bit 2 is a write-only bit.

Caution If the system clock is changed while the timer is counting, an error occurs in the timer as follows (when system clock f_X = 4 MHz):

- **High-speed mode 16/f_X → Normal mode 32/f_X ... (Error due to resolution of set timer) +1.5 μs**
- **Normal mode 32/f_X → High-speed mode 16/f_X ... (Error due to resolution of set timer) –1.5 μs**

5.3 Carrier Generator for Remote Controller

μPD17246 is provided with a carrier generator for the remote controller.

The remote controller carrier generator consists of an 8-bit counter, NRZ high-level timer modulo register (NRZHTMM), and NRZ low-level timer modulo register (NRZLTMM). The high-level and low-level periods are set in the corresponding modulo registers through the DBF to determine the carrier duty factor and carrier frequency.

As a clock input to the 8-bit counter, fx/2, fx, or 2fx can be selected by using REMCK0 and REMCK1 (address 13H, bits 0 and 1) of the register file (this clock for carrier generation is Rfx). When Rfx is oscillated by a 4 MHz resonator, therefore, the input clock is 2 MHz (fx/2), 4 MHz (fx), or 8 MHz (2fx).

The NRZ high-level output timer modulo register is called NRZHTMM, and the NRZ low-level timer modulo register is called NRZLTMM. Data is written to these registers by the PUT instruction. The contents in these register are read by the GET instruction.

Whether the REM pin outputs a carrier or a high level is selected by REMEN (address 12H, bit 1) of the register file. Be sure to clear this bit to 0 to output a carrier.

NRZLTMM

7	6	5	4	3	2	1	0	Address	After reset	R/W
8-bit modulo register								Peripheral register: 03H	Undefined	R/W

NRZHTMM

7	6	5	4	3	2	1	0	Address	After reset	R/W
8-bit modulo register								Peripheral register: 03H	Undefined	R/W

★

3	2	1	0	Address	After reset	R/W
0	0	REMCK1	REMCK0	RF: 13H	0H	R/W

REMCK1	REMCK0	Clock for carrier generation (Rfx)
0	0	Rfx = fx/2 (when fx = 4 MHz, Rfx = 2 MHz)
0	1	Rfx = fx (when fx = 4 MHz, Rfx = 4 MHz)
1	0	Rfx = 2fx[Note] (when fx = 4 MHz, Rfx = 8 MHz)
1	1	

Note Rfx = 2fx can be selected only when fx = 3.5 to 4.5 MHz.

5.3.1 Remote controller signal output control

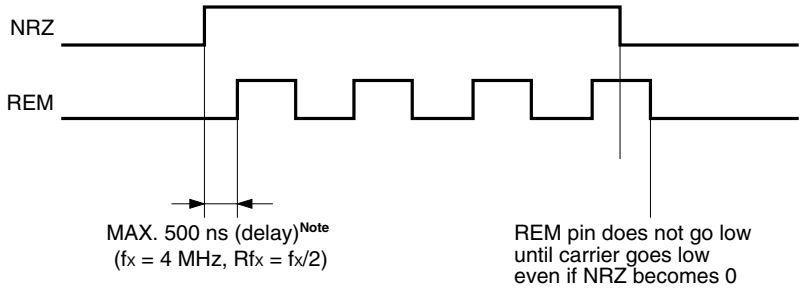
The REM pin, which outputs the carrier, is controlled by bits NRZ and NRZBF of the register file and timer 0. While the NRZ contents are "1", the clock generated by the remote controller carrier generator is output to the REM pin; while the NRZ contents are "0", the REM pin outputs a low level. The NRZBF contents are automatically transferred to NRZ by the interrupt signal generated by timer 0. If data is set in NRZBF in advance, the REM pin status changes in synchronization with the timer 0 counting operation.

If the interrupt signal is generated from timer 0 with the REM pin at the high level (i.e. NRZ is "1") and the carrier clock at the high level, the REM pin output does not accord with the updated contents of NRZ until the carrier clock goes low. This processing is useful for holding the high level pulse width from the output carrier constant (refer to the figure below).

When the contents of NRZ are "0", the remote controller carrier generator stops. However, if the clock for timer 0 is output from the remote controller carrier generator, the clock continues to operate, even when the NRZ contents become "0".

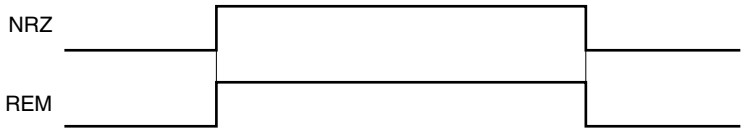
An actual example showing a remote controller signal output to the REM pin is given below.

When REMEN (address 12H, bit 1) of register file is 0 (carrier output)



Note Value when (TMCK1, TMCK0) ≠ (1, 1).
When (TMCK1, TMCK0) = (1, 1), the value differs depending on how NRZ is manipulated. If NRZ is set by an instruction, the width of the first high-level pulse may be shortened. If NRZ is set by data transferred from NRZBF, the high-level pulse is delayed by the low-level pulse of the carrier clock.

When REMEN (address 12H, bit 1) of register file is 1 (carrier not output)



3	2	1	0	Address	After reset	R/W
0	0	REMEN	NRZ	RF: 12H	0H	R/W

REMEN	NRZ	NRZ data
0	0	Outputs low level to REM pin
0	1	Outputs a carrier to REM pin
1	0	Outputs low level to REM pin
1	1	Outputs high level to REM pin

3	2	1	0	Address	After reset	R/W
0	0	0	NRZBF	RF: 11H	0H	R/W

NRZBF	NRZ data output next
0	NRZ buffer bit. Transferred to NRZ by interrupt signal of timer 0.
1	

Setting carrier frequency and duty factor

Where the system clock frequency is fx, carrier frequency is fc, and carrier generation clock is Rfx:

- When Rfx = fx/2: ℓ (division ratio) = fx/(2 × fc)
- When Rfx = fx: ℓ (division ratio) = fx/fc
- When Rfx = 2fx: ℓ (division ratio) = 2fx/fc

ℓ is divided into m:n and is set in the modulo registers as follows:

High-level period set value = { ℓ × m/(m + n)} – 1
Low-level period set value = { ℓ × n/(m + n)} – 1

Example Where fc = 38 kHz, duty factor (high-level period) = 1/3, fx = 4 MHz, and Rfx = 2fx:

ℓ = 2 × 4 MHz/38 kHz = 210.5
m:n = 1:2

From the above, the value of the modulo register is:

High-level period ≒ 69
Low-level period ≒ 139

Therefore, the carrier frequency is 38.10 kHz.

Table 5-1. Carrier Frequency List

(1) Where fx = 4 MHz and Rfx = fx/2

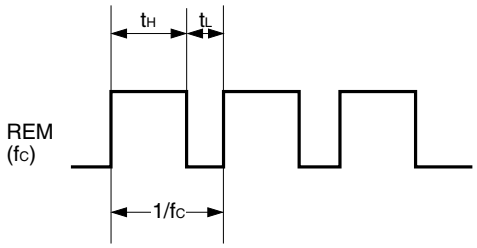
Set Value		t_H (μs)	t_L (μs)	1/fc (μs)	fc (kHz)	Duty
NRZHTMM	NRZLTMM					
00H	00H	0.5	0.5	1.0	1000	1/2
01H	02H	1.0	1.5	2.5	400	2/5
04H	04H	2.5	2.5	5.0	200	1/2
09H	09H	5.0	5.0	10.0	100	1/2
0FH	10H	8.0	8.5	16.5	60.6	1/2
0FH	21H	8.0	17.0	25.0	40.0	1/3
11H	21H	9.0	17.0	26.0	38.5	1/3
11H	22H	9.0	17.5	26.5	37.7	1/3
19H	35H	13.0	27.0	40.0	25.0	1/3
3FH	3FH	32.0	32.0	64.0	15.6	1/2
7FH	7FH	64.0	64.0	128.0	7.8	1/2
★ FFH	FFH	128.0	128.0	256.0	3.9	1/2

(2) Where fx = 4 MHz, Rfx = fx (original oscillation)

Set Value		t_H (μs)	t_L (μs)	$1/f_C$ (μs)	f_C (kHz)	Duty
NRZHTMM	NRZLTMM					
00H	00H	0.25	0.25	0.5	2000	1/2
01H	02H	0.5	0.75	1.25	800	2/5
04H	04H	1.25	1.25	2.5	400	1/2
09H	09H	2.5	2.5	5.0	200	1/2
0FH	10H	4.0	4.25	8.25	121	1/2
0FH	21H	4.0	8.5	12.5	80	1/3
11H	21H	4.5	8.5	13.0	76.9	1/3
11H	22H	4.5	8.75	13.25	75.47	1/3
19H	35H	6.5	13.5	20.0	50	1/3
3FH	3FH	16.0	16.0	32.0	31.25	1/2
7FH	7FH	32.0	32.0	64.0	15.6	1/2
★ FFH	FFH	64.0	64.0	128.0	7.8	1/2

(3) Where fx = 4 MHz, Rfx = 2fx

Set Value		t_H (μs)	t_L (μs)	$1/f_C$ (μs)	f_C (kHz)	Duty
NRZHTMM	NRZLTMM					
00H	00H	0.125	0.125	0.25	4,000	1/2
07H	0BH	1.0	1.5	2.5	400	2/5
13H	13H	2.5	2.5	5.0	200	1/2
27H	27H	5.0	5.0	10	100	1/2
41H	41H	8.25	8.25	16.5	60.6	1/2
41H	85H	8.25	16.75	25	40	1/3
★ 45H	89H	8.75	17.25	26.0	38.5	1/3
★ 45H	8BH	8.75	17.5	26.25	38.10	1/3
★ 69H	D5H	13.25	26.75	40.0	25	1/3
C7H	C7H	25.0	25.0	50.0	20	1/2
FFH	FFH	32.0	32.0	64.0	15.6	1/2



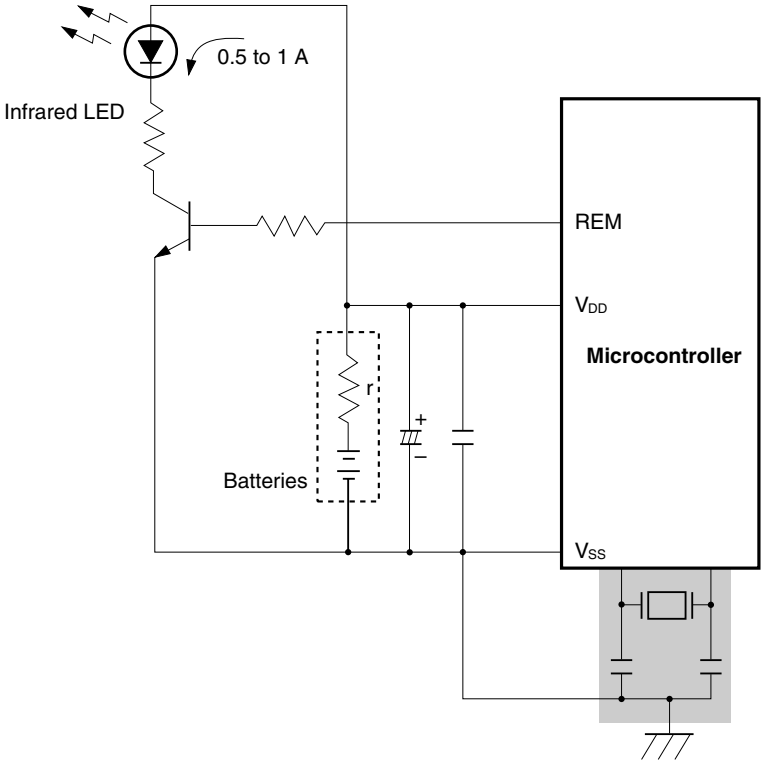
5.3.2 Countermeasures against noise during transmission (carrier output)

When a signal is transmitted from the transmitter of a remote controller, a peak current of 0.5 to 1 A may flow through the infrared LED. Since two batteries are usually used as the power source of the transmitter, several Ω of equivalent resistance (r) exists in the power source as shown in Figure 5-2. This resistance increases to 10 to 20 Ω if the supply voltage drops to 2 V. While the carrier is being output from the REM pin (while the infrared LED lights), therefore, a high-frequency noise may be generated on the power lines due to the voltage fluctuation that may take place especially during switching.

To minimize the influence on the microcontroller of this high-frequency noise, take the following measures.

<1> Separate the power lines of the microcontroller from the power lines of the infrared LED with the terminals of the batteries at the center. Use thick power lines and keep the wiring short.
<2> Locate the resonator as close as possible to the microcontroller and shield it with GND lines (as indicated by the shaded portion in the figure below).
<3> Locate the capacitor for stabilization of the power supply closely to the power lines of the microcontroller. Also, use a capacitor to eliminate high-frequency noise.
<4> To prevent data from changing, do not execute data read/write processing such as key scan, an interrupt that requires a stack, or the CALL/RET instruction, while the carrier is being output.
<5> To improve the reliability in case of program hang-up, use the watchdog timer.

Figure 5-2. Example of Countermeasures Against Noise



6. BASIC INTERVAL TIMER/WATCHDOG TIMER

The basic interval timer has a function to generate the interval timer interrupt signal and watchdog timer reset signal.

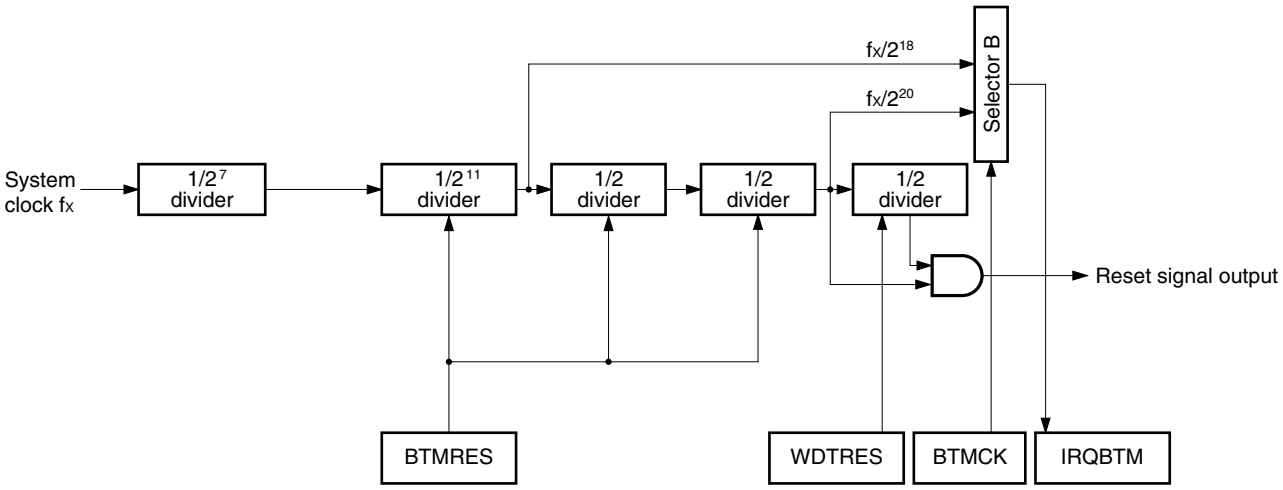
6.1 Source Clock for Basic Interval Timer

The system clock (fx) is divided to generate the source clock for the basic interval timer. The input clock frequency for the basic interval timer is $f_x/2^7$. When the CPU is set in the STOP mode, the basic interval timer also stops.

6.2 Controlling Basic Interval Timer

The basic interval timer is controlled by the bits in the register file. That is, the basic interval timer is reset by BTMRES. The frequency for the interrupt signal, output by the basic interval timer, is selected by BTMMD, and the watchdog timer is reset by WDTRES.

Figure 6-1. Basic Interval Timer Configuration



3	2	1	0	Address	After reset	R/W
WDTRES	BTMCK	BTMRES	0	RF: 03H	0H	R/W[Note]

BTMRES	Basic interval timer reset
0	Data read out is always "0"
1	Writing "1" resets basic interval timer

BTMCK	Basic interval timer mode selection
0	Generates interrupt signal IRQBTM every $f_X/2^{20}$
1	Generates interrupt signal IRQBTM every $f_X/2^{18}$

WDTRES	Watchdog timer reset
0	Data read out is always "0"
1	Writing "1" resets watchdog timer ($f_X/2^{21}$ counter)

Note Bits 1 and 3 are write-only bits.

6.3 Operation Timing for Watchdog Timer

The basic interval timer can be used as a watchdog timer.

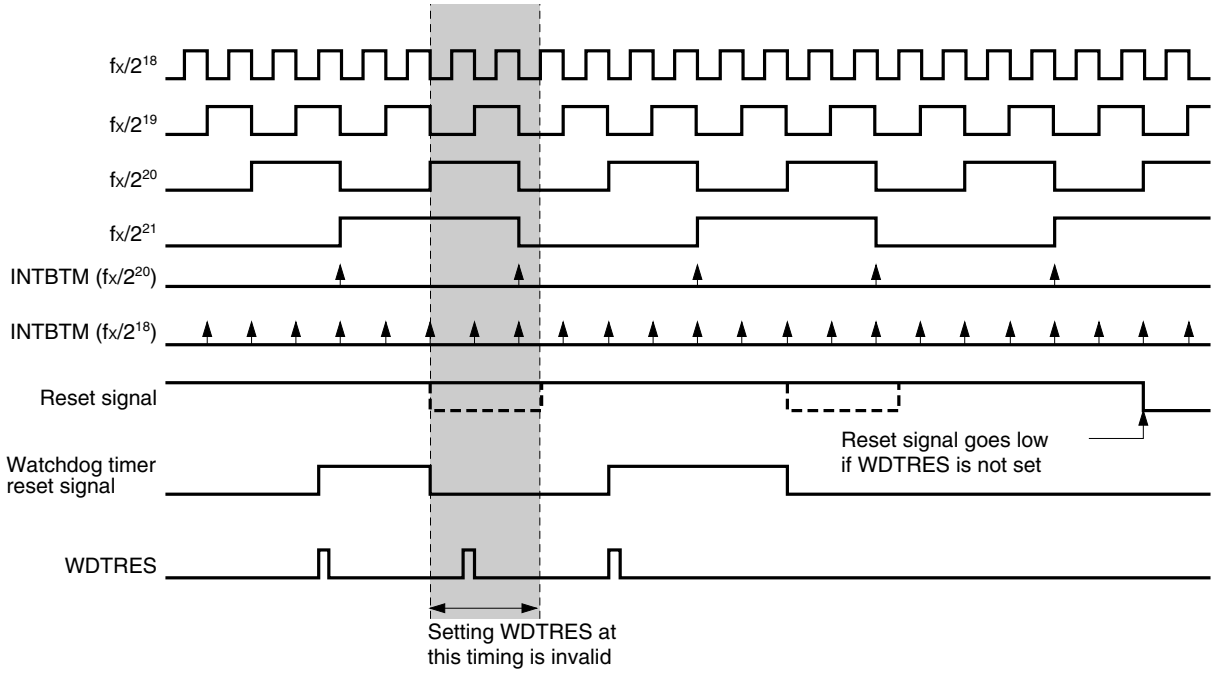
Unless the watchdog timer is reset within a fixed time[Note], it is judged that "the program has hung up", and the μPD17246 is reset. It is therefore necessary to reset the watchdog timer via programming within the fixed time.

The watchdog timer can be reset by setting WDTRES to 1.

Note Fixed time: Approx. 340 ms (at 4 MHz)

Caution The watchdog timer cannot be reset in the shaded range in Figure 6-2. Therefore, set WDTRES before both the $f_X/2^{21}$ and $f_X/2^{20}$ signals go high.

Figure 6-2. Watchdog Timer Operation Timing



7. RAM RETENTION DETECTOR

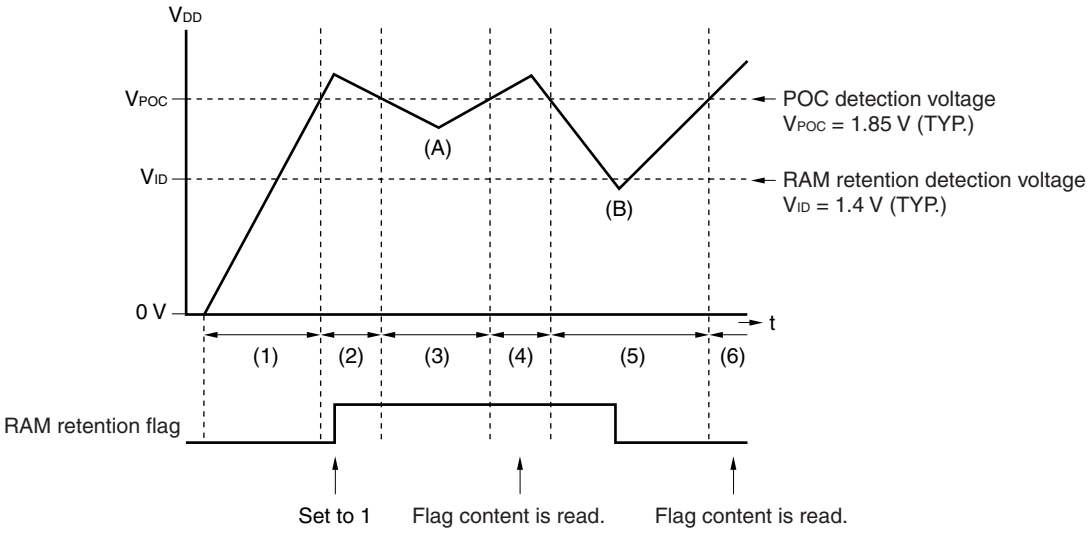
7.1 RAM Retention Flag

The RAM retention flag (bit 0 of the register file at address 21H) indicates whether the supply voltage has dropped below the level at which the contents of the RAM are lost while the battery is being exchanged or when the battery voltage has dropped.

This flag is at bit 3 of control register 0 (P3).

It is cleared to 0 if the supply voltage drops below the RAM retention detection voltage (approx. 1.4 V TYP.). If this flag is 0, it can be judged that the RAM contents have been lost or that power has just been applied. This flag can be used to initialize the RAM via software. After initializing the RAM and writing the necessary data to it, set this RAM retention flag to "1" by software. At this time, 1 means that data has been set to the RAM.

Figure 7-1. Supply Voltage Transition and Detection Voltage



(1) If the supply voltage rises after the battery has been set, and exceeds V_{POC} (POC detection voltage), reset is cleared. Because the supply voltage rises from 0 V, which is lower than V_{ID} (RAM retention detection voltage), the RAM retention flag remains in the initial status 0.

(2) The supply voltage has now risen to the level at which the device can operate. Write the necessary data to the RAM and set the RAM retention flag to 1.

(3) The device is reset if the supply voltage drops below V_{POC}. At point (A) in the above figure, the RAM retention flag remains 1 because the supply voltage is higher than V_{ID} at this point.

(4) If the RAM retention flag is checked by software after reset has been cleared, it is 1. This means that the contents of the RAM have not been lost. It is therefore not necessary to initialize the RAM by software.

(5) The device is reset if the supply voltage drops below V_{POC}. At point (B) in the figure, the voltage is lower than V_{ID}. Consequently, the RAM retention flag is cleared to 0.

(6) If the RAM retention flag is checked by software after reset has been cleared, it is 0. This means that the contents of the RAM may have been lost. If this happens, initialize the RAM by software.

3	2	1	0	Address	After reset	R/W
0	0	0	RAMFLAG	RF: 21H	Undefined[Note]	R/W

RAMFLAG	RAM retention flag
0	RAM data may be undefined.
1	RAM data are retained.

★ **Note** RAMFLAG is “0” when V_{DD} is about 1.4 V or less, and “undefined” when V_{DD} is about 1.4 V or more.

8. INTERRUPT FUNCTIONS

8.1 Interrupt Sources

μPD17246 is provided with three interrupt sources.

When an interrupt has been acknowledged, the program execution automatically branches to a predetermined address, which is called a vector address. A vector address is assigned to each interrupt source, as shown in Table 8-1.

Table 8-1. Vector Address

Priority	Interrupt Source	Ext/Int	Vector Address
1	8-bit timer	Internal	0004H
2	INT pin rising and falling edges	External	0003H
3	Basic interval timer	Internal	0002H

Remark 0001H is normal address

When more than one interrupt request is issued at the same time, the interrupts are acknowledged in sequence, starting from the one with the highest priority.

Whether an interrupt is enabled or disabled is specified by the EI or DI instruction. The basic condition under which an interrupt is acknowledged is that the interrupt is enabled by the EI instruction. While the DI instruction is executed, or while an interrupt is acknowledged, the interrupt is disabled.

To enable acknowledgement of an interrupt after the interrupt has been processed, the EI instruction must be executed before the RETI instruction. Acknowledging the interrupt is enabled by the EI instruction after the instruction next to the EI instruction has been executed. Therefore, no interrupt can be acknowledged between the EI and RETI instructions.

Caution In interrupt processing, only the BCD, CMP, CY, Z, IXE flags are automatically saved to the stack by the hardware, to a maximum of three levels. Also, within the interrupt processing contents, when peripheral hardware (timer, A/D converter, etc.) is accessed, the DBF and WR contents are not saved by the hardware. Accordingly, it is recommended that at the beginning of interrupt processing, DBF and WR be saved by software to RAM, and immediately before finishing interrupt processing, the saved contents be returned to their original location.

8.2 Hardware of Interrupt Controller

This section describes the flags of the interrupt controller.

(1) Interrupt request flag and interrupt enable flag

The interrupt request flag (IRQ×××) is set to 1 when an interrupt request is generated, and is automatically cleared to 0 when the interrupt processing is executed.

An interrupt enable flag (IP×××) is provided for each interrupt request flag. When the IP××× flag is 1, the interrupt is enabled; when it is 0, the interrupt is disabled.

(2) EI/DI instruction

Whether an acknowledged interrupt is executed or not is specified by the EI or DI instruction.

When the EI instruction is executed, INTE (interrupt enable flag), which enables the interrupt, is set to 1. The INTE flag is not registered on the register file. Consequently, the status of this flag cannot be checked by an instruction.

The DI flag clears the INTE flag to 0 to disable all the interrupts.

The INTE flag is also cleared to 0 at reset, disabling all the interrupts.

Table 8-2. Interrupt Request Flags and Interrupt Enable Flag

Interrupt Request Flag	Signal Setting Interrupt Request Flag	Interrupt Enable Flag
IRQTM	Reset by 8-bit timer.	IPTM
IRQ	Set when edge of INT pin input signal is detected	IP
IRQBTM	Reset by basic interval timer.	IPBTM

8.2.1 INT

This flag reads the INT pin status.

When a high level is input to the INT pin, this flag is set to 1; when a low level is input, the flag is reset to 0.

3	2	1	0	Address	After reset	R/W
0	0	0	INT	RF: 0FH	Undefined	R

INT	INT pin level detection
0	INT pin: Low level
1	INT pin: High level

8.2.2 IEG

This pin selects the interrupt edge to be detected on the INT pin.

When this flag is 0, the interrupt is detected at the rising edge; when it is 1, the interrupt is detected at the falling edge.

8.2.3 INTSEL

This flag selects whether pin 3 is used as the INT pin or $P1B_0$ pin. When INTSEL is cleared to 0, pin 3 functions as the $P1B_0$ pin; when it is set to 1, the pin functions as the INT pin.

After reset, the $P1B_0$ pin is selected.

3	2	1	0	Address	After reset	R/W
0	0	INTSEL	IEG	RF: 1FH	0H	R/W

IEG	INT pin interrupt detection edge selection
0	Rising edge of INT pin
1	Falling edge of INT pin

INTSEL	Selection of pin 3 function
0	As $P1B_0$ pin
1	As INT pin

8.2.4 Interrupt enable flag

This flag enables each interrupt source. When this flag is 1, the corresponding interrupt is enabled; when it is 0, the interrupt is disabled.

3	2	1	0	Address	After reset	R/W
0	IPBTM	IP	IPTM	RF: 2FH	0H	R/W

IPTM	8-bit timer interrupt enable flag
0	Disables interrupt acknowledgement by 8-bit timer
1	Enables interrupt acknowledgement by 8-bit timer

IP	INT pin interrupt enable flag
0	Disables interrupt acknowledgement by INT pin input
1	Enables interrupt acknowledgement by INT pin input

IPBTM	Basic interval timer interrupt enable flag
0	Disables interrupt acknowledgement by basic interval timer
1	Enables interrupt acknowledgement by basic interval timer

8.2.5 IRQ

This is an interrupt request flag that indicates the interrupt request status.

When an interrupt request is generated, this flag is set to 1. When the interrupt has been acknowledged, the interrupt request flag is reset to 0.

The interrupt request flag can be read or written by the program. Therefore, when it is set to 1, an interrupt can be generated by the software. By writing 0 to the flag, the interrupt pending status can be canceled.

3	2	1	0	Address	After reset	R/W
0	0	0	IRQBTM	RF: 3DH	0H	R/W

IRQBTM	Basic interval timer interrupt request flag
0	Interrupt request has not been made.
1	Basic interval timer interrupt request has been made.

3	2	1	0	Address	After reset	R/W
0	0	0	IRQ	RF: 3EH	0H	R/W

IRQ	INT pin interrupt request flag
0	Interrupt request has not been made.
1	Interrupt request has been made at rising edge or falling edge of INT input.

3	2	1	0	Address	After reset	R/W
0	0	0	IRQTM	RF: 3FH	1H[Note]	R/W

IRQTM	8-bit timer interrupt request flag
0	Interrupt request has not been made.
1	8-bit timer interrupt request has been made.

Note It is also set to 1H after the STOP mode is released.

8.3 Interrupt Sequence

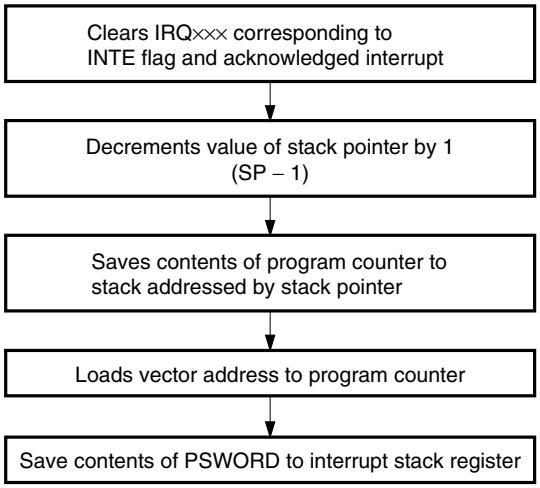
If the IRQ×× flag is set to 1 when the IP×× flag is "1", interrupt processing is started after the instruction cycle of the instruction executed when the IRQ×× flag was set has ended. Since the MOVT instruction, EI instruction, and the instruction that matches the condition to skip use two instruction cycles, the interrupt enabled while this instruction is executed is processed after the second instruction cycle is over.

If the IP×× flag is "0", the interrupt processing is not performed even if the IRQ×× flag is set, until the IP×× flag is set.

If two or more interrupts are enabled simultaneously, the interrupts are processed starting from the one with the highest priority. The interrupt with the lower priority is held pending until the processing of the interrupt with the higher priority is finished.

8.3.1 Operations when interrupt is acknowledged

When an interrupt has been acknowledged, the CPU performs processing in the following sequence:

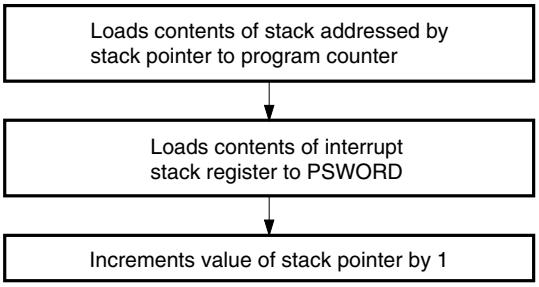


One instruction cycle is required to perform the above processing.

8.3.2 Returning from interrupt processing routine

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

The following processing is then executed within an instruction cycle.



To enable an interrupt after the processing of an interrupt has finished, the EI instruction must be executed immediately before the RETI instruction.

Interrupt acknowledgement is enabled by the EI instruction after the instruction next to the EI instruction has been executed. Therefore, the interrupt is not acknowledged between the EI and RETI instructions.

9. STANDBY FUNCTIONS

The μPD17246 is provided with HALT and STOP modes as standby functions.

By using the standby function, current consumption can be reduced.

In the HALT mode, the program is not executed, but the system clock fx is not stopped. This mode is maintained, until the HALT mode release condition is satisfied.

In the STOP mode, the system clock is stopped and program execution is stopped. This mode is maintained, until the STOP mode release condition is satisfied.

The HALT mode is set, when the HALT instruction has been executed. The STOP mode is set, when the STOP instruction has been executed.

9.1 HALT Mode

In this mode, program execution is temporarily stopped, with the main clock continuing oscillation, to reduce current consumption.

Use the HALT instruction to set the HALT mode.

The HALT mode release condition can be specified by the operand for the HALT instruction, as shown in Table 9-1.

After the HALT mode has been released, the operation is performed as shown in Table 9-2 and Figure 9-1.

Caution Do not execute an instruction that clears the interrupt request flag (IRQ×××) for which the interrupt enable flag (IP×××) is set immediately before the HALT 8H instruction; otherwise, the HALT mode may not be set.

Table 9-1. HALT Mode Releasing Conditions

Operand Value	Release Conditions
0010B (02H)	When interrupt request (IRQTM) occurs for 8-bit timer
1000B (08H)	<1> When interrupt request (IRQTM, IRQBTM, or IRQ), whose interrupt enable flag (IPTM, IPBTM, or IP) is set, occurs <2> When any of $P0A_0$ to $P0A_3$ pins goes low <3> When $P0B_0$ to $P0B_3$, $P0C_0$ to $P0C_3$, and $P0D_0$ to $P0D_3$ are used as input pins and any of these goes low <4> If $P0E_0$ to $P0E_3$ are used as input pins when a key matrix is used and if any of these pins goes low <5> If $P1A_0$ to $P1A_2$ and $P1B_0$ are used as input pins when a key matrix is used and if the level of any of these pins is the set clear level[Note]
Other than above	Setting prohibited

Note Set the clear level by using bits 0 to 2 (P1AHL0 to P1AHL2) of the register file at address 05H, and bit 2 (P1BHL0) at address 15H.

Table 9-2. Operations After HALT Mode Release

(a) HALT 08H

HALT Mode Released by:	Interrupt Status	Interrupt Enable Flag	Operations After HALT Mode Release
When release condition of $P0A_0$ to $P0A_3$, $P0B_0$ to $P0B_3$, $P0C_0$ to $P0C_3$, $P0D_0$ to $P0D_3$, $P0E_0$ to $P0E_3$, $P1A_0$ to $P1A_2$, $P1B_0$ is satisfied	Don't care	Don't care	Instruction next to HALT is executed
When release condition is satisfied by interrupt request	DI	Disabled	Standby mode is not released
		Enabled	Instruction next to HALT is executed
	EI	Disabled	Standby mode is not released
		Enabled	Branches to interrupt vector address

(b) HALT 02H

HALT Mode Released by:	Interrupt Status	Interrupt Enable Flag	Operations After HALT Mode Release
8-bit timer	DI	Disabled	Instructions are executed from the instruction next to the HALT instruction.
		Enabled	
	EI	Disabled	
		Enabled	Branches to interrupt vector address

9.2 HALT Instruction Execution Conditions

The HALT instruction can be executed under special conditions, as shown in Table 9-3, to prevent the program from hanging up.

If the conditions in Table 9-3 are not satisfied, the HALT instruction is treated as a NOP instruction.

Table 9-3. HALT Instruction Execution Conditions

Operand Value	Execution Conditions
0010B (02H)	When all interrupt request flags (IRQTM) of 8-bit timer are reset
1000B (08H)	<1> When interrupt request flag (IRQTH, IRQBTM, or IRQ) is reset, corresponding to interrupt whose interrupt enable flag (IPTM, IPBTM, or IP) is set <2> When high level is input to all $P0A_0$ to $P0A_3$ pins <3> When $P0B_0$ to $P0B_3$, $P0C_0$ to $P0C_3$, and $P0D_0$ to $P0D_3$ are used as input pins, a high level must be input to all the pins. <4> A high level must be input to all the pins if $P0E_0$ to $P0E_3$ are used as input pins when a key matrix is used. <5> A level reverse to the set clear level[Note] must be input to all the pins if $P1A_0$ to $P1A_2$ and $P1B_0$ are used as input pins when a key matrix is used (for example, if the clear level is high, the execution condition is low-level input).
Other than above	Setting prohibited

Note Set the clear level by using bits 0 to 2 (P1AHL0 to P1AHL2) of the register file at address 05H, and bit 2 (P1BHL0) at address 15H.

9.3 STOP Mode

In the STOP mode, the system clock (fx) oscillation is stopped and the program execution is stopped to minimize current consumption.

To set the STOP mode, use the STOP instruction.

The STOP mode release condition can be specified by the STOP instruction operand, as shown in Table 9-4.

After the STOP mode has released, the μPD17246 performs the following.

<1> Resets IRQTM.
<2> Starts the basic interval timer and watchdog timer (does not reset).
<3> Resets and starts the 8-bit timer.
<4> Executes the instruction next to [STOP 8H] when the current value of the 8-bit counter matches the value of the modulo register (IRQTM is set).

The μPD17246 oscillator is stopped when the STOP instruction has been executed (i.e., in the STOP mode). Oscillation is not resumed until the STOP mode is released. After the STOP mode has been released, the HALT mode is set. Set the time required to release the HALT mode by using the timer with modulo function.

The time that elapses from when the STOP mode has been released by occurrence of an interrupt until an operation mode is set is shown in the following table.

Caution Do not execute an instruction that clears the interrupt request flag (IRQ×××) for which the interrupt enable flag (IP×××) is set immediately before the STOP 8H instruction; otherwise, the STOP mode may not be set.

8-Bit Modulo Register Set Value (TMM)	Time Required to Set Operation Mode After STOP Mode Release
	At 4 MHz
40H	4.160 ms (64 μs × 65)
FFH	16.384 ms (64 μs × 256)

Caution To set the time required for an operation mode to be set after the STOP mode has been released, make sure that sufficient time is allowed for oscillation to stabilize.

Remark Set the 8-bit modulo timer before executing STOP instruction.

Table 9-4. STOP Mode Release Conditions

Operand Value	Release Conditions
1000B (08H)	<1> When any of $P0A_0$ to $P0A_3$ pins goes low <2> When $P0B_0$ to $P0B_3$, $P0C_0$ to $P0C_3$, and $P0D_0$ to $P0D_3$ are used as input pins and any of these goes low <3> If the interrupt request (IRQ) of an interrupt for which the INT pin interrupt enable flag (IP) is set is generated at the rising or falling edge of the INT pin <4> If $P0E_0$ to $P0E_3$ are used as input pins when a key matrix is used and if any of these pins goes low <5> If $P1A_0$ to $P1A_2$ and $P1B_0$ are used as input pins when a key matrix is used and if the level of any of these pins is the set clear level[Note]
Other than above	Setting prohibited

Note Set the clear level by using bits 0 to 2 (P1AHL0 to P1AHL2) of the register file at address 05H, and bit 2 (P1BHL0) at address 15H.

9.4 STOP Instruction Execution Conditions

The STOP instruction can be executed under special conditions, as shown in Table 9-5, to prevent the program from hanging up.

If the conditions in Table 9-5 are not satisfied, the STOP instruction is treated as an NOP instruction.

Table 9-5. STOP Instruction Execution Conditions

Operand Value	Execution Conditions
1000B (08H)	<1> High level input for all $P0A_0$ to $P0A_3$ pins <2> When $P0B_0$ to $P0B_3$, $P0C_0$ to $P0C_3$, and $P0D_0$ to $P0D_3$ are used as input pins and all pins are high <3> If the INT pin interrupt request flag (IRQ) for an interrupt for which the INT pin interrupt enable flag (IP) is set is reset <4> A high level must be input to all the pins if $P0E_0$ to $P0E_3$ are used as input pins when a key matrix is used. <5> A level reverse to the set clear level[Note] must be input to all the pins if $P1A_0$ to $P1A_2$ and $P1B_0$ are used as input pins when a key matrix is used (for example, if the clear level is high, the execution condition is low-level input).
Other than above	Setting prohibited

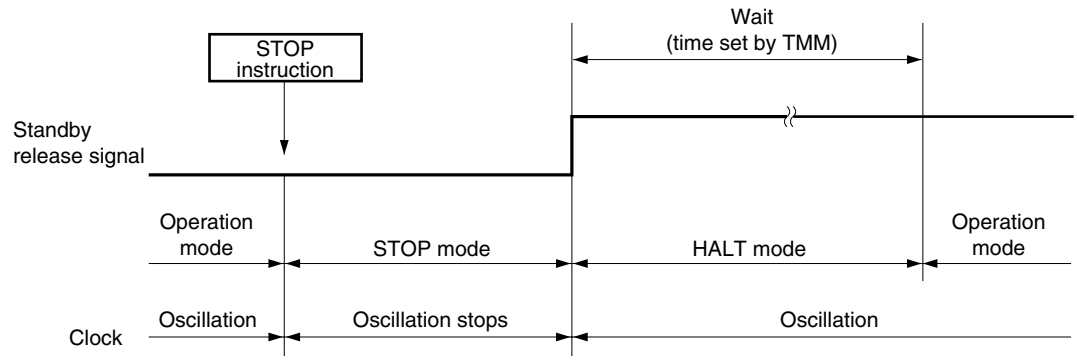
Note Set the clear level by using bits 0 to 2 (P1AHL0 to P1AHL2) of the register file at address 05H, and bit 2 (P1BHL0) at address 15H.

9.5 Releasing Standby Mode

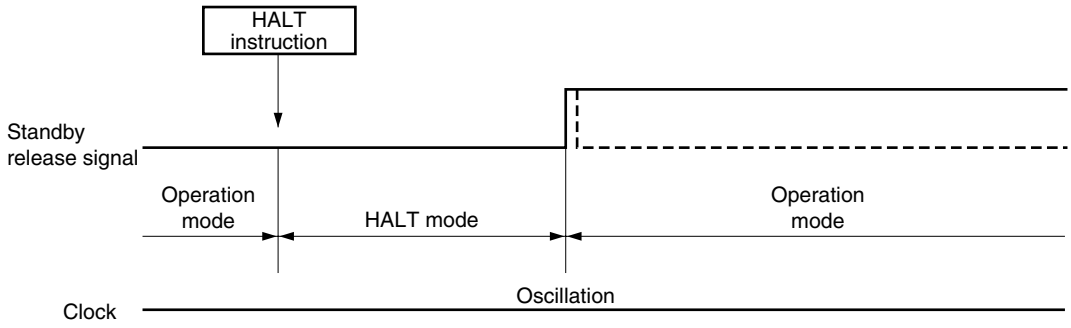
The operations for releasing the STOP and HALT modes are as shown in Figure 9-1.

Figure 9-1. Operations After Standby Mode Release

(a) Releasing STOP mode by interrupt



(b) Releasing HALT mode by interrupt



Remark The dotted line indicates the operation to be performed when the interrupt request releasing the standby mode has been acknowledged.

10. RESET

10.1 Reset by Reset Signal Input

When a low-level signal of more than 10 μs is input to the $\overline{\text{RESET}}$ pin, the μPD17246 is reset.

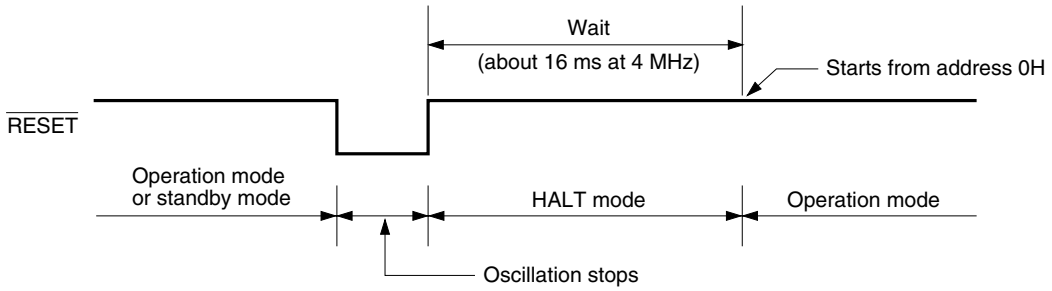
When the system is reset, the oscillator remains in the HALT mode and then enters an operation mode, in the same way as when the STOP mode is released. The wait time after the reset signal has been canceled is 16.384 ms (fx = 4 MHz).

On power application, input the reset signal at least once because the internal circuitry operations are not stable. When μPD17246 is reset, the following initialization takes place.

(1) Program counter is reset to 0.
(2) Flags in the register file are initialized to their default values (for the default values, refer to **Figure 12-1 Register Files**).
(3) The default value (0320H) is written to the data buffer (DBF).
(4) The hardware peripherals are initialized.
(5) The system clock (fx) stops oscillation.

When the $\overline{\text{RESET}}$ pin is made high, the system clock starts oscillating, and the program execution starts from address 0 about 16 ms (at 4 MHz) later.

Figure 10-1. Reset Operation by $\overline{\text{RESET}}$ Input



10.2 Reset by Watchdog Timer (with $\overline{\text{RESET}}$ Pin Internally Pulled Down)

When the watchdog timer operates during program execution, the $\overline{\text{RESET}}$ pin is internally pulled down, and the program counter is reset to 0 (normally, the $\overline{\text{RESET}}$ pin is pulled up).

If the watchdog timer is not reset for a fixed period of time, the program can be restarted from address 0H.

Program so that the watchdog timer is reset at intervals of within 340 ms (at fx = 4 MHz) (set the WDTRES flag).

10.3 Reset by Stack Pointer (with $\overline{\text{RESET}}$ Pin Internally Pulled Down)

When the value of the stack pointer reaches 6H or 7H during program execution, the $\overline{\text{RESET}}$ pin is internally pulled down, and the program counter is reset to 0 (normally, the $\overline{\text{RESET}}$ pin is pulled up).

Therefore, if an interrupt or CALL instruction is executed when the value of the stack pointer is 0 (stack underflow) or if the stack level exceeds 6 as a result of execution of the RET instruction because the correspondence between the CALL and RET instructions is not established (stack overflow), the program can be restarted from address 0H.

Table 10-1. Status of Each Hardware After Reset

Hardware		$\overline{\text{RESET}}$ Input in Standby Mode	$\overline{\text{RESET}}$ Input During Operation
Program counter (PC)		0000H	0000H
Ports	Input/output	Input	Input
	Output latch	0	0
Data memory (RAM)	General-purpose data memory (Except DBF, port register)	Retains previous status	Undefined
	DBF	0320H	0320H
	System register (SYSREG)	0	0
	WR	Retains previous status	Undefined
Control registers		Refer to **Figure 12-1 Register Files**	
8-bit timer	Counter (TMC)	00H	00H
	Modulo register (TMM)	FFH	FFH
Remote controller carrier generator	NRZ high-level timer modulo register (NRZHTMM)	Retains previous status	Undefined
	NRZ low-level timer modulo register (NRZLTMM)		
Basic interval timer/watchdog timer counter		00H	00H

11. LOW-VOLTAGE DETECTOR (WITH $\overline{\text{RESET}}$ PIN INTERNALLY PULLED DOWN)

The $\overline{\text{RESET}}$ pin is internally pulled down for initialization (reset) to prevent program hang-up that may take place when the batteries are replaced, if the low-voltage detector detects a low voltage.

A drop in the supply voltage is detected if the status in which V_{DD} is about 1.7 to 2.0 V lasts for 1 ms or longer. Note, however, that 1 ms is the guaranteed value and that the microcontroller may be reset even if the above low-voltage condition lasts for less than 1 ms.

Although the voltage at which the reset function is effected ranges from about 1.7 to 2.0 V, the program counter is prevented from hanging up even if the supply voltage drops until the reset function is effected. Note that a resonator may stop oscillating before the reset function is effected if normal operation under the low voltage is not guaranteed.

The low-voltage detector can be set arbitrarily by a mask option.

12. ASSEMBLER RESERVED WORDS

12.1 Mask Option Directives

When developing the μPD17246 program, mask options must be specified by using mask option directives in the program.

To select the low-voltage detector and capacitor for oscillation of the μPD17246, a mask option must be specified.

12.1.1 OPTION and ENDOP directives

The portion of the program enclosed by the OPTION and ENDOP directives is called a mask option definition block. This block is described in the following format.

Description format:

Symbol	Mnemonic	Operand	Comment
[Label:]	OPTION		[;Comment]
	⋮		
	ENDOP		

12.1.2 Mask option definition directives

Table 12-1 lists the directives that can be used in the mask option definition block.

Here is an example of mask option definition.

Description example:

Symbol	Mnemonic	Operand	Comment
	OPTION		
	OPTPOC	USEPOC	; Internal low-voltage detector
	OPTCAP	USECAP	; Internal capacitor for oscillation
	ENDOP		

Table 12-1. Mask Option Definition Directives

Name	Directive	Operands	1st Operand	2nd Operand	3rd Operand	4th Operand
CAP	OPTCAP	1	USECAP (capacitor for oscillation provided)			
			NOUSECAP (capacitor for oscillation not provided)			
POC	OPTPOC	1	USEPOC (low-voltage detector provided)			
			NOUSEPOC (low-voltage detector not provided)			

12.2 Reserved Symbols

The symbols defined by the μPD17246 device file are listed in Table 12-2.

The defined symbols are the following register file names, port names, and peripheral hardware names.

12.2.1 Register file

The names of the symbols assigned to the register file are defined. These registers are accessed by the PEEK and POKE instructions via the window register (WR). Figure 12-1 shows the register file.

12.2.2 Registers and ports on data memory

The names of the registers assigned to addresses 00H to 7FH on the data memory and the names of ports assigned to address 70H and those that follow, and system register names are defined. Figure 12-2 shows the data memory configuration.

12.2.3 Peripheral hardware

The names of peripheral hardware accessed by the GET and PUT instructions are defined. Table 12-3 shows the peripheral hardware.

Table 12-2. Reserved Symbols (1/3)

Symbol Name	Attribute	Value	R/W	Description
DBF3	MEM	0.0CH	R/W	Bits 15 to 12 of data buffer
DBF2	MEM	0.0DH	R/W	Bits 11 to 8 of data buffer
DBF1	MEM	0.0EH	R/W	Bits 7 to 4 of data buffer
DBF0	MEM	0.0FH	R/W	Bits 3 to 0 of data buffer
AR3	MEM	0.74H	R/W	Bits 15 to 12 of address register
AR2	MEM	0.75H	R/W	Bits 11 to 8 of address register
AR1	MEM	0.76H	R/W	Bits 7 to 4 of address register
AR0	MEM	0.77H	R/W	Bits 3 to 0 of address register
WR	MEM	0.78H	R/W	Window register
BANK	MEM	0.79H	R/W	Bank register
IXH	MEM	0.7AH	R/W	Index register, high
MPH	MEM	0.7AH	R/W	Data memory row address 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	Data memory row address pointer, low
IXL	MEM	0.7CH	R/W	Index register, low
RPH	MEM	0.7DH	R/W	General register pointer, high
RPL	MEM	0.7EH	R/W	General register pointer, low
PSW	MEM	0.7FH	R/W	Program status word
BCD	FLG	0.7EH.0	R/W	BCD flag
CMP	FLG	0.7FH.3	R/W	Compare flag
CY	FLG	0.7FH.2	R/W	Carry flag
Z	FLG	0.7FH.1	R/W	Zero flag
IXE	FLG	0.7FH.0	R/W	Index enable flag
P0A0	FLG	0.70H.0	R/W	Bit 0 of port 0A
P0A1	FLG	0.70H.1	R/W	Bit 1 of port 0A
P0A2	FLG	0.70H.2	R/W	Bit 2 of port 0A
P0A3	FLG	0.70H.3	R/W	Bit 3 of port 0A
P0B0	FLG	0.71H.0	R/W	Bit 0 of port 0B
P0B1	FLG	0.71H.1	R/W	Bit 1 of port 0B
P0B2	FLG	0.71H.2	R/W	Bit 2 of port 0B
P0B3	FLG	0.71H.3	R/W	Bit 3 of port 0B
P0C0	FLG	0.72H.0	R/W	Bit 0 of port 0C
P0C1	FLG	0.72H.1	R/W	Bit 1 of port 0C
P0C2	FLG	0.72H.2	R/W	Bit 2 of port 0C
P0C3	FLG	0.72H.3	R/W	Bit 3 of port 0C
P0D0	FLG	0.73H.0	R/W	Bit 0 of port 0D
P0D1	FLG	0.73H.1	R/W	Bit 1 of port 0D
P0D2	FLG	0.73H.2	R/W	Bit 2 of port 0D
P0D3	FLG	0.73H.3	R/W	Bit 3 of port 0D

Table 12-2. Reserved Symbols (2/3)

Symbol Name	Attribute	Value	R/W	Description
P0E0	FLG	0.6FH.0	R/W	Bit 0 of port 0E
P0E1	FLG	0.6FH.1	R/W	Bit 1 of port 0E
P0E2	FLG	0.6FH.2	R/W	Bit 2 of port 0E
P0E3	FLG	0.6FH.3	R/W	Bit 3 of port 0E
P1A0	FLG	1.70H.0	R/W	Bit 0 of port 1A
P1A1	FLG	1.70H.1	R/W	Bit 1 of port 1A
P1A2	FLG	1.70H.2	R/W	Bit 2 of port 1A
P1B0	FLG	1.71H.0	R/W	Bit 0 of port 1B
SP	MEM	0.81H	R/W	Stack pointer
SYSCK	FLG	0.82H.0	R/W	System clock select flag
WDTRES	FLG	0.83H.3	R/W	Watchdog timer reset flag
BTMCK	FLG	0.83H.2	R/W	Basic interval timer mode select flag
BTMRES	FLG	0.83H.1	R/W	Basic interval timer mode reset flag
P1AHL0	FLG	0.85H.0	R/W	$P1A_0$ port standby clear level select flag
P1AHL1	FLG	0.85H.1	R/W	$P1A_1$ port standby clear level select flag
P1AHL2	FLG	0.85H.2	R/W	$P1A_2$ port standby clear level select flag
P1AKEY0	FLG	0.86H.0	R/W	$P1A_0$ port key matrix select flag
P1AKEY1	FLG	0.86H.1	R/W	$P1A_1$ port key matrix select flag
P1AKEY2	FLG	0.86H.2	R/W	$P1A_2$ port key matrix select flag
P1ABPU0	FLG	0.87H.0	R/W	$P1A_0$ port pull-up resistor select flag
P1ABPU1	FLG	0.87H.1	R/W	$P1A_1$ port pull-up resistor select flag
P1ABPU2	FLG	0.87H.2	R/W	$P1A_2$ port pull-up resistor select flag
INT	FLG	0.8FH.0	R	INT pin status flag
NRZBF	FLG	0.91H.0	R/W	NRZ buffer data flag
NRZ	FLG	0.92H.0	R/W	NRZ data flag
REMEN	FLG	0.92H.1	R/W	Carrier output select flag
REMCK1	FLG	0.93H.1	R/W	Carrier generation clock select flag
REMCK0	FLG	0.93H.0	R/W	Carrier generation clock select flag
P1BHL0	FLG	0.95H.2	R/W	$P1B_0$ port standby clear level select flag
P1BKEY0	FLG	0.95H.1	R/W	$P1B_0$ port key matrix select flag
P1BBPU0	FLG	0.95H.0	R/W	$P1B_0$ port pull-up resistor select flag
P0EKEY0	FLG	0.96H.0	R/W	$P1E_0$ port key matrix select flag
P0EKEY1	FLG	0.96H.1	R/W	$P1E_1$ port key matrix select flag
P0EKEY2	FLG	0.96H.2	R/W	$P1E_2$ port key matrix select flag
P0EKEY3	FLG	0.96H.3	R/W	$P1E_3$ port key matrix select flag
P0EBPU0	FLG	0.97H.0	R/W	$P0E_0$ pull-up setting flag
P0EBPU1	FLG	0.97H.1	R/W	$P0E_1$ pull-up setting flag
P0EBPU2	FLG	0.97H.2	R/W	$P0E_2$ pull-up setting flag
P0EBPU3	FLG	0.97H.3	R/W	$P0E_3$ pull-up setting flag
INTSEL	FLG	0.9FH.1	R/W	INT select flag

Table 12-2. Reserved Symbols (3/3)

Symbol Name	Attribute	Value	R/W	Description
IEG	FLG	0.9FH.0	R/W	INT pin interrupt edge flag
RAMFLAG	FLG	0.0A1H.0	R/W	RAM retention flag
P1ABIO0	FLG	0.0A5H.0	R/W	$P1A_0$ I/O select flag
P1ABIO1	FLG	0.0A5H.1	R/W	$P1A_1$ I/O select flag
P1ABIO2	FLG	0.0A5H.2	R/W	$P1A_2$ I/O select flag
P0BBIO0	FLG	0.0A6H.0	R/W	$P0B_0$ I/O select flag
P0BBIO1	FLG	0.0A6H.1	R/W	$P0B_1$ I/O select flag
P0BBIO2	FLG	0.0A6H.2	R/W	$P0B_2$ I/O select flag
P0BBIO3	FLG	0.0A6H.3	R/W	$P0B_3$ I/O select flag
P0EBIO0	FLG	0.0A7H.0	R/W	$P0E_0$ I/O setting flag
P0EBIO1	FLG	0.0A7H.1	R/W	$P0E_1$ I/O setting flag
P0EBIO2	FLG	0.0A7H.2	R/W	$P0E_2$ I/O setting flag
P0EBIO3	FLG	0.0A7H.3	R/W	$P0E_3$ I/O setting flag
IPBTM	FLG	0.0AFH.2	R/W	Basic interval timer interrupt enable flag
IP	FLG	0.0AFH.1	R/W	INT pin interrupt enable flag
IPTM	FLG	0.0AFH.0	R/W	Timer interrupt enable flag
TMEN	FLG	0.0B3H.3	R/W	Timer enable flag
TMRES	FLG	0.0B3H.2	R/W	Timer reset flag
TMCK1	FLG	0.0B3H.1	R/W	Timer clock flag
TMCK0	FLG	0.0B3H.0	R/W	Timer clock flag
P0CGIO	FLG	0.0B7H.2	R/W	$P0C_3$ to $P0C_0$ I/O select flag
P0DGIO	FLG	0.0B7H.3	R/W	$P0D_3$ to $P0D_0$ I/O select flag
IRQBTM	FLG	0.0BDH.0	R/W	Basic interval timer interrupt request flag
IRQ	FLG	0.0BEH.0	R/W	INT pin interrupt request flag
IRQTM	FLG	0.0BFH.0	R/W	Timer interrupt request flag
TMC	DAT	05H	R	Timer count register
TMM	DAT	06H	W	Timer modulo register
NRZLTMM	DAT	03H	R/W	NRZ low-level timer modulo register
NRZHTMM	DAT	04H	R/W	NRZ high-level timer modulo register
AR	DAT	40H	R/W	Address register
USECAP	DAT	0FF11H	—	Capacitor with oscillator is used.
NOUSECAP	DAT	0FF22H	—	Capacitor with oscillator is not used.
USEPOC	DAT	0FF33H	—	POC circuit is used.
NOUSEPOC	DAT	0FF44H	—	POC circuit is not used.
DBF	DAT	0FH	—	Fixed operand value for PUT, GET, MOVT instruction
IX	DAT	01H	—	Fixed operand value for INC instruction
AR_EPA1	DAT	8040H	—	Indicates that the EPA bit of AR is ON.

Figure 12-1. Register Files (1/2)

Row Address \ Column Address		0	Note	1	Note	2	Note	3	Note	4	Note	5	Note	6	Note	7	Note
0	Bit 3			SP	0	0	0	WDTRES	0			0	0	0	0	0	0
	Bit 2				1	0	0	BTMCK	0			P1AHL2	0	P1AKEY2	0	P1ABPU2	0
	Bit 1				0	0	0	BTMRES	0			P1AHL1	0	P1AKEY1	0	P1ABPU1	0
	Bit 0				1	SYSCK	0	0	0			P1AHL0	0	P1AKEY0	0	P1ABPU0	0
1	Bit 3			0	0	0	0	0	0			0	0	P0EKEY3	0	P0EBPU3	0
	Bit 2			0	0	0	0	0	0			P1BHL0	0	P0EKEY2	0	P0EBPU2	0
	Bit 1			0	0	REMEN	0	REMCK1	0			P1BKEY0	0	P0EKEY1	0	P0EBPU1	0
	Bit 0			NRZBF	0	NRZ	0	REMCK0	0			P1BPU0	0	P0EKEY0	0	P0EBPU0	0
2	Bit 3			0	0							0	0	P0BBIO3	0	P0EBIO3	0
	Bit 2			0	0							P1ABIO2	0	P0BBIO2	0	P0EBIO2	0
	Bit 1			0	0							P1ABIO1	0	P0BBIO1	0	P0EBIO1	0
	Bit 0			RAMFLAG	0							P1ABIO0	0	P0BBIO0	0	P0EBIO0	0
3	Bit 3							TMEN	1							P0DGIO	1
	Bit 2							TMRES	0							P0CGIO	1
	Bit 1							TMCK1	0							0	0
	Bit 0							TMCK0	0							0	0

Note After reset

Figure 12-2. Data Memory Configuration

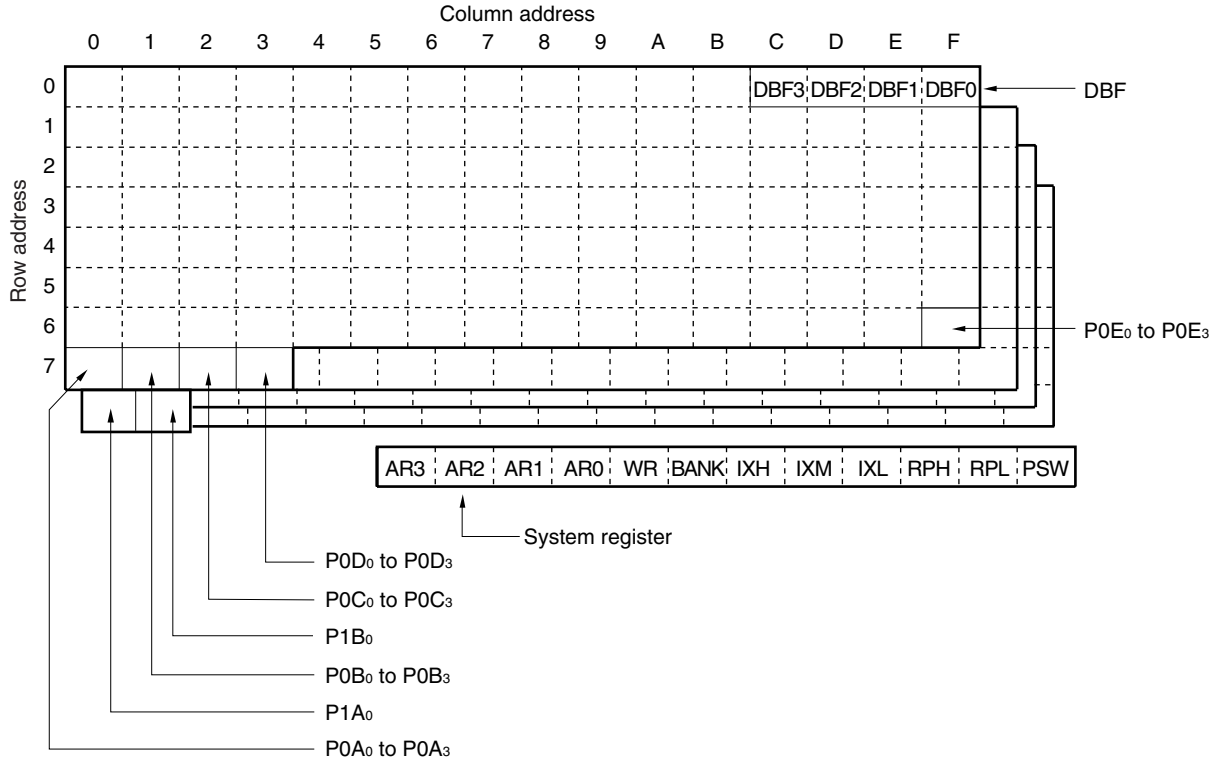


Figure 12-1. Register Files (2/2)

Row Address \ Column Address		8	Note	9	Note	A	Note	B	Note	C	Note	D	Note	E	Note	F	Note
0	Bit 3															0	0
	Bit 2															0	0
	Bit 1															0	0
	Bit 0															INT	P
1	Bit 3															0	0
	Bit 2															0	0
	Bit 1															INTSEL	0
	Bit 0															IEG	0
2	Bit 3															0	0
	Bit 2															IPBTM	0
	Bit 1															IP	0
	Bit 0															IPTM	0
3	Bit 3											0	0	0	0	0	0
	Bit 2											0	0	0	0	0	0
	Bit 1											0	0	0	0	0	0
	Bit 0											IRQBTM	0	IRQ	0	IRQTM	1

Note After reset

P: When INT pin is high level, 1; when INT pin is low level, 0.

Table 12-3. Peripheral Hardware

Name	Address	Valid Bit	Description
TMC	05H	8	Timer count register
TMM	06H	8	Timer modulo register
NRZLTMM	03H	8	Low-level timer modulo register for NRZ
NRZHTMM	04H	8	High-level timer modulo register for NRZ
AR	40H	16	Address register

13. INSTRUCTION SET

13.1 Instruction Set Outline

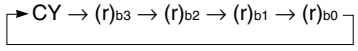
b_{14} to b_{11} \ b_{15}		0		1	
BIN.	HEX.				
0 0 0 0	0	ADD	r, m	ADD	m, #n4
0 0 0 1	1	SUB	r, m	SUB	m, #n4
0 0 1 0	2	ADDC	r, m	ADDC	m, #n4
0 0 1 1	3	SUBC	r, m	SUBC	m, #n4
0 1 0 0	4	AND	r, m	AND	m, #n4
0 1 0 1	5	XOR	r, m	XOR	m, #n4
0 1 1 0	6	OR	r, m	OR	m, #n4
0 1 1 1	7	INC INC MOVT BR CALL RET SYSCAL RETSK EI DI RETI PUSH POP GET PUT PEEK POKE RORC STOP HALT NOP	AR IX DBF, @AR @AR @AR entry[Note] AR AR DBF, p p, DBF WR, rf rf, WR r s h		
1 0 0 0	8	LD	r, m	ST	m, r
1 0 0 1	9	SKE	m, #n4	SKGE	m, #n4
1 0 1 0	A	MOV	@r, m	MOV	m, @r
1 0 1 1	B	SKNE	m, #n4	SKLT	m, #n4
1 1 0 0	C	BR	addr (Page 0)	CALL	addr
1 1 0 1	D	BR	addr (Page 1)	MOV	m, #n4
1 1 1 0	E	BR	addr (Page 2)	SKT	m, #n
1 1 1 1	F	BR	addr (Page 3)	SKF	m, #n

Note μPD17244, 17245, 17246 only

13.2 Legend

AR:	Address register
ASR:	Address stack register specified by stack pointer
addr:	Program memory address (lower 11 bits)
BANK:	Bank register
CMP:	Compare flag
CY:	Carry flag
DBF:	Data buffer
entry:	Entry address of system segment
h:	Halt releasing condition
INTEF:	Interrupt enable flag
INTR:	Register automatically saved to stack in case of interrupt
INTSK:	Interrupt stack register
IX:	Index register
MP:	Data memory row address pointer
MPE:	Memory pointer enable flag
m:	Data memory address specified by m_R, m_C
m_R:	Data memory row address (high)
m_C:	Data memory column address (low)
n:	Bit position (4 bits)
n4:	Immediate data (4 bits)
PAGE:	Page (bits 11 and 12 of program counter)
PC:	Program counter
p:	Peripheral address
p_H:	Peripheral address (higher 3 bits)
p_L:	Peripheral address (lower 4 bits)
r:	General register column address
rf:	Register file address
rf_R:	Register file row address (higher 3 bits)
rf_C:	Register file column address (lower 4 bits)
SP:	Stack pointer
s:	Stop releasing condition
WR:	Window register
(×):	Contents addressed by ×

13.3 List of Instructions

Group	Mnemonic	Operand	Operation	Instruction Code			
				Opcode	Operand		
Add	ADD	r, m	(r) ← (r) + (m)	00000	m_R	m_C	r
		m, #n4	(m) ← (m) + n4	10000	m_R	m_C	n4
	ADDC	r, m	(r) ← (r) + (m) + CY	00010	m_R	m_C	r
		m, #n4	(m) ← (m) + n4 + CY	10010	m_R	m_C	n4
	INC	AR	AR ← AR + 1	00111	000	1001	0000
		IX	IX ← IX + 1	00111	000	1000	0000
Subtract	SUB	r, m	(r) ← (r) – (m)	00001	m_R	m_C	r
		m, #n4	(m) ← (m) – n4	10001	m_R	m_C	n4
	SUBC	r, m	(r) ← (r) – (m) – CY	00011	m_R	m_C	r
		m, #n4	(m) ← (m) – n4 – CY	10011	m_R	m_C	n4
Logical	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
Judge	SKT	m, #n	CMP ← 0, if (m) ∧ n = n, then skip	11110	m_R	m_C	n
	SKF	m, #n	CMP ← 0, if (m) ∧ n = 0, then skip	11111	m_R	m_C	n
Compare	SKE	m, #n4	(m) – n4, skip if zero	01001	m_R	m_C	n4
	SKNE	m, #n4	(m) – n4, skip if not zero	01011	m_R	m_C	n4
	SKGE	m, #n4	(m) – n4, skip if not borrow	11001	m_R	m_C	n4
	SKLT	m, #n4	(m) – n4, skip if borrow	11011	m_R	m_C	n4
Rotate	RORC	r		00111	000	0111	r
Transfer	LD	r, m	(r) ← (m)	01000	m_R	m_C	r
	ST	m, r	(m) ← (r)	11000	m_R	m_C	r
	MOV	@r, m	if MPE = 1 : (MP, (r)) ← (m) if MPE = 0 : (BANK, m_R, (r)) ← (m)	01010	m_R	m_C	r
		m, @r	if MPE = 1 : (m) ← (MP, (r)) if MPE = 0 : (m) ← (BANK, m_R, (r))	11010	m_R	m_C	r
		m, #n4	(m) ← n4	11101	m_R	m_C	n4
	MOVT	DBF, @AR	SP ← SP – 1, ASR ← PC, PC ← AR DBF ← (PC), PC ← ASR, SP ← SP + 1	00111	000	0001	0000

Group	Mnemonic	Operand	Operation	Instruction Code			
				Opcode	Operand		
Transfer	PUSH	AR	SP ← SP – 1, ASR ← AR	00111	000	1101	0000
	POP	AR	AR ← ASR, SP ← SP + 1	00111	000	1100	0000
	PEEK	WR, rf	WR ← (rf)	00111	rf_R	0011	rf_C
	POKE	rf, WR	(rf) ← WR	00111	rf_R	0010	rf_C
	GET	DBF, p	(DBF) ← (p)	00111	p_H	1011	p_L
	PUT	p, DBF	(p) ← (DBF)	00111	p_H	1010	p_L
Branch	BR	addr	**Note 1**	**Note 1**	addr		
		@AR	PC ← AR	00111	000	0100	0000
Subroutine	CALL	addr	SP ← SP – 1, ASR ← PC, PC_{10-0} ← addr, PAGE ← 0	11100	addr		
		@AR	SP ← SP – 1, ASR ← PC, PC ← AR	00111	000	0101	0000
	SYSCAL[Note 2]	entry	SP ← SP – 1, ASR ← PC, SGR ← 1, $PC_{12,11}$ ← 0, PC_{10-8} ← $entry_H$, PC_{7-4} ← 0, PC_{3-0} ← $entry_L$	00111	$entry_H$	0000	$entry_L$
	RET		PC ← ASR, SP ← SP + 1	00111	000	1110	0000
	RETSK		PC ← ASR, SP ← SP + 1 and skip	00111	001	1110	0000
	RETI		PC ← ASR, INTR ← INTSK, SP ← SP + 1	00111	100	1110	0000
Interrupt	EI		INTEF ← 1	00111	000	1111	0000
	DI		INTEF ← 0	00111	001	1111	0000
Other	STOP	s	STOP	00111	010	1111	s
	HALT	h	HALT	00111	011	1111	h
	NOP		No operation	00111	100	1111	0000

Notes 1. The operation and operation codes “BR addr” of the μPD17240, 17241, 17242, 17243, 17244, 17245, and 17246 are as follows.

(a) μPD17240

Operand	Operation	Opcode
addr	PC_{10-0} ← addr	01100

(b) μPD17241

Operand	Operation	Opcode
addr	PC_{10-0} ← addr, Page ← 0	01100
	PC_{10-0} ← addr, Page ← 1	01101

(c) μPD17242

Operand	Operation	Opcode
addr	PC_{10-0} ← addr, Page ← 0	01100
	PC_{10-0} ← addr, Page ← 1	01101
	PC_{10-0} ← addr, Page ← 2	01110

(d) μPD17243, 17244, 17245, 17246

Operand	Operation	Opcode
addr	PC_{10-0} ← addr, Page ← 0	01100
	PC_{10-0} ← addr, Page ← 1	01101
	PC_{10-0} ← addr, Page ← 2	01110
	PC_{10-0} ← addr, Page ← 3	01111

2. μPD17244, 17245, and 17246 only

13.4 Assembler (RA17K) Embedded Macro Instructions

Legend

flag n: FLG type symbol
n: Bit number
< >: Contents in < > can be omitted

	Mnemonic	Operand	Operation	n
Embedded macro	SKTn	flag 1, ...flag n	if (flag 1) to (flag n) = all "1", then skip	1 ≤ n ≤ 4
	SKFn	flag 1, ...flag n	if (flag 1) to (flag n) = all "0", then skip	1 ≤ n ≤ 4
	SETn	flag 1, ...flag n	(flag 1) to (flag n) ← 1	1 ≤ n ≤ 4
	CLRn	flag 1, ...flag n	(flag 1) to (flag n) ← 0	1 ≤ n ≤ 4
	NOTn	flag 1, ...flag n	if (flag n) = "0", then (flag n) ← 1 if (flag n) = "1", then (flag n) ← 0	1 ≤ n ≤ 4
	INITFLG	<NOT> flag 1, ···<<NOT> flag n>	if description = NOT flag n, then (flag n) ← 0 if description = flag n, then (flag n) ← 1	1 ≤ n ≤ 4
	BANKn		(BANK) ← n	n = 0, 1
Expansion instruction	BRX	Label	Jump Label	—
	CALLX	function-name	CALL sub-routine	—
	INITFLGX	<NOT/INV> flag 1, ...<NOT/INV> flag n	if description = NOT (or INV) flag, (flag) ← 0 if description = flag, (flag) ← 1	n ≤ 4

14. ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings (T_A = 25°C)

Item	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.3 to +3.8	V
Input voltage	V_I			–0.3 to V_{DD} + 0.3	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3	V
Output current, high[Note]	I_{OH}	REM pin	Peak value	–36.0	mA
			rms value	–24.0	mA
		1 pin (P0E, P1A pins)	Peak value	–7.5	mA
			rms value	–5.0	mA
		Total of P0E, P1A pins	Peak value	–22.5	mA
			rms value	–15.0	mA
Output current, low[Note]	I_{OL}	1 pin (P0B, P0C, P0D, P0E, P1A, REM pins)	Peak value	7.5	mA
			rms value	5.0	mA
		Total of P0B, P0C, P0D, REM pins	Peak value	22.5	mA
			rms value	15.0	mA
		Total of P0E, P1A pins	Peak value	30.0	mA
			rms value	20.0	mA
Operating temperature	T_A			–40 to +85	°C
Storage temperature	T_{stg}			–65 to +150	°C
Power dissipation	P_d	T_A = 85°C		180	mW

Note Calculate rms value by this expression: [rms value] = [Peak value] × $\sqrt{\text{Duty}}$

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

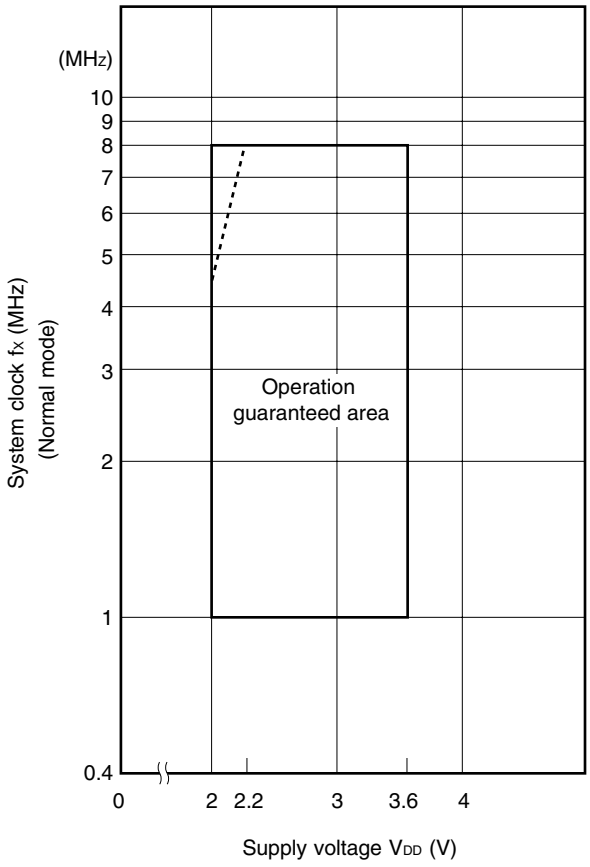
Recommended Operating Ranges (T_A = –40 to +85°C, V_{DD} = 2.0 to 3.6 V)

	Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
★	Supply Voltage	V_{DD1}	fx = 1 MHz	High-speed mode (Instruction execution time: 16 μs)	2.0		3.6	V
		V_{DD2}	fx = 4 MHz	High-speed mode (Instruction execution time: 4 μs)				
		V_{DD3}	fx = 8 MHz	Normal mode (Instruction execution time: 4 μs)				
		V_{DD4}		High-speed mode (Instruction execution time: 2 μs)	2.2		3.6	V
★	Oscillation frequency	fx	Rfx = fx/2 or fx		1.0	4.0	8.0	MHz
			Rfx = 2fx		3.5	4.0	4.5	MHz
★	Operating temperature	T_A			–40	+25	+85	°C
	Low-voltage detector[Note] (Mask option)	t_{CY}			3.5		32	μs

Note Reset if the status of V_{DD} = 1.7 to 2.0 V lasts for 1 ms or longer. Program hang-up does not occur even if the voltage drops, until the reset function is effected. A resonator may stop oscillating before the reset function is effected if normal operation under the low voltage is not guaranteed.

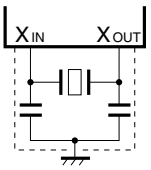
Caution Design the application circuit so that the $\overline{\text{RESET}}$ pin goes low when the supply voltage is less than 2.2 V.

f_X vs V_{DD}



Remark The region indicated by the broken lines in the above figure is the guaranteed operating range in the high-speed mode.

System Clock Oscillator Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 3.6 V)

Resonator	Recommended Constants	Item	Conditions	MIN.	TYP.	MAX.	Unit
★ Ceramic resonator		Oscillation frequency (f_X)Note 1		1.0	4.0	8.0	MHz
		Oscillation stabilization timeNote 2	After V_{DD} reached MIN. in oscillation voltage range			4	ms

Notes 1. The oscillation frequency only indicates the oscillator characteristics.

2. The oscillation stabilization time is necessary for oscillation to be stabilized after V_{DD} application or STOP mode release.

Caution When using the system clock oscillator, wire as follows in the area enclosed by the dotted lines in the above figure, to avoid an adverse effect from wiring capacitance.

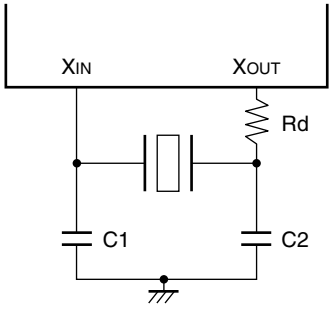
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with other signal lines. Do not route the wiring near a signal line through which a large current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as GND. Do not ground the capacitor to a ground pattern through which a large current flows.**
- **Do not fetch signals from the oscillator.**

★ **Recommended Oscillator Constant**

Ceramic resonator (T_A = –40 to +85°C)

Manufacturer	Part Number	Frequency (MHz)	Recommended Circuit Constant (pF)		Oscillation Voltage Range (V_{DD})		Remarks
			C1	C2	MIN.	MAX.	
Murata Mfg. Co., Ltd.	CSBLA1M00J58-B0[Note]	1.0	100	100	1.8	3.6	Rd = 3.3 kΩ
	CSBFB1M00J58-R1[Note]						
	CSTLS2M00G56-B0[Note]	2.0	–	–			Rd = 1.0 kΩ
	CSTCC2M00G56-R0[Note]						On-chip capacitor
	CSTLS3M00G56-B0[Note]	3.0					Rd = 470 Ω
	CSTCC3M00G56-R0[Note]						On-chip capacitor
	CSTLS4M00G56-B0	4.0					On-chip capacitor
	CSTCR4M00G55-R0						
	CSTLS6M00G56-B0	6.0					
	CSTCR6M00G55-R0						
	CSTLS8M00G56-B0	8.0					
	CSTCC8M00G56-R0						
TDK	FCR3.52MC5	3.52	–	–	1.8	3.6	On-chip capacitor
	FCR4.0MC5	4.0					
	FCR4.0MSC5	4.0					
	FCR6.0MC5	6.0					
	FCR8.0MC5	8.0					
Kyocera Corp.	KBR-2.0MS	2.0	68	68	1.8	3.6	–
	KBR-3.0MS	3.0	47	47			
	KBR-4.0MKE	4.0	–	–			On-chip capacitor
	KBR-4.0MSE		33	33			–
	KBR-6.0MKC	6.0	–	–			On-chip capacitor
	KBR-6.0MSB		33	33			–
	KBR-8.0MKC	8.0	–	–			On-chip capacitor
	KBR-8.0MSB		33	33			–

Note A limiting resistor is required when these ceramic resonators are used (refer to the following figure). When other recommended resonators are used, the limiting resistor is not necessary.



Caution The oscillator constant is a reference value based on evaluation in specific environments by the resonator manufacturer. If the oscillator characteristics need to be optimized in the actual application, request the resonator manufacturer for evaluation on the implementation circuit. Note that the oscillation voltage and oscillation frequency merely indicate the characteristics of the oscillator. The internal operation conditions of the μPD17240, 17241, 17242, 17243, 17244, 17245, and 17246 must be within the specifications of the DC and AC characteristics.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 3.6 V)

	Item	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
	Input voltage, high	V_{IHI1}	$\overline{RESET}$, INT			0.80V_{DD}		V_{DD}	V
		V_{IH2}	P0A, P0B, P0C, P0D			0.70V_{DD}		V_{DD}	V
		V_{IH3}	P0E, P1A, P1B			0.70V_{DD}		V_{DD}	V
	Input voltage, low	V_{IL1}	$\overline{RESET}$, INT			0		0.2V_{DD}	V
		V_{IL2}	P0A, P0B, P0C, P0D			0		0.3V_{DD}	V
		V_{IL3}	P0E, P1A, P1B			0		0.3V_{DD}	V
	Input leakage current, high	I_{LIH}	P0A, P0B, P0C, P0D, P0E, P1A, P1B_0/INT, $\overline{RESET}$	V_{IH} = V_{DD} w/o pull-down resistor				3.0	μA
	Input leakage current, low	I_{LIL}	P0E, P1A, P1B_0/INT	V_{IL} = 0 V w/o pull-up resistor				–3.0	μA
	Internal pull-up resistor	R_1	P0E, P1A, P1B, $\overline{RESET}$ (pulled up)			25	50	100	kΩ
		R_2	P0A, P0B, P0C, P0D			100	200	400	kΩ
★	Internal pull-down resistor	R_3	P1A, P1B			25	50	100	kΩ
	Output current, high	I_{OH}	REM	V_{OH} = 1.0 V, V_{DD} = 3 V		–6	–13	–24	mA
	Output voltage, high	V_{OH}	P0E, P1A, REM	I_{OH} = –0.5 mA		V_{DD}–0.3		V_{DD}	V
	Output voltage, low	V_{OL1}	P0B, P0C, P0D, REM	I_{OL} = 0.5 mA		0		0.3	V
		V_{OL2}	P0E, P1A	I_{OL} = 1.5 mA		0		0.3	V
	Data retention characteristics	V_{DDDR}	$\overline{RESET}$ = Low level or STOP mode			1.3		3.6	V
	Low-voltage detection voltage (mask option)	V_{DT}	$\overline{RESET}$ pin pulled down, V_{DT} = V_{DD}				1.85	2.0	V
	RAM retention detection voltage	V_{ID}	V_{ID} = V_{DD}, RAMFLAG = 0 (RF21H.0)				1.40	1.50	V
★	Supply current	I_{DD1}	Operating mode (high-speed)	V_{DD} = 3 V ±10%	f_X = 1 MHz		0.6	1.1	mA
					f_X = 4 MHz		0.75	1.3	mA
					f_X = 8 MHz		0.9	1.6	mA
		I_{DD2}	Operating mode (low-speed)	V_{DD} = 3 V ±10%	f_X = 1 MHz		0.48	0.9	mA
					f_X = 4 MHz		0.6	1.1	mA
					f_X = 8 MHz		0.8	1.4	mA
		I_{DD3}	HALT mode	V_{DD} = 3 V ±10%	f_X = 1 MHz		0.4	0.75	mA
					f_X = 4 MHz		0.45	0.85	mA
					f_X = 8 MHz		0.5	0.95	mA
		I_{DD4}	STOP mode	V_{DD} = 3 V ±10%			2.0	20.0	μA
				built-in POC	T_A = 25°C		2.0	5.0	μA

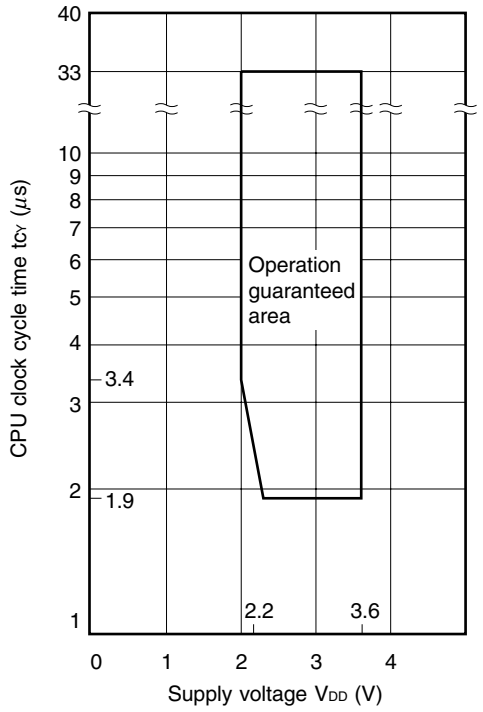
Note This does not include the current that flows through the internal pull-up resistors.

AC Characteristics (T_A = –40 to +85°C, V_{DD} = 2.0 to 3.6 V)

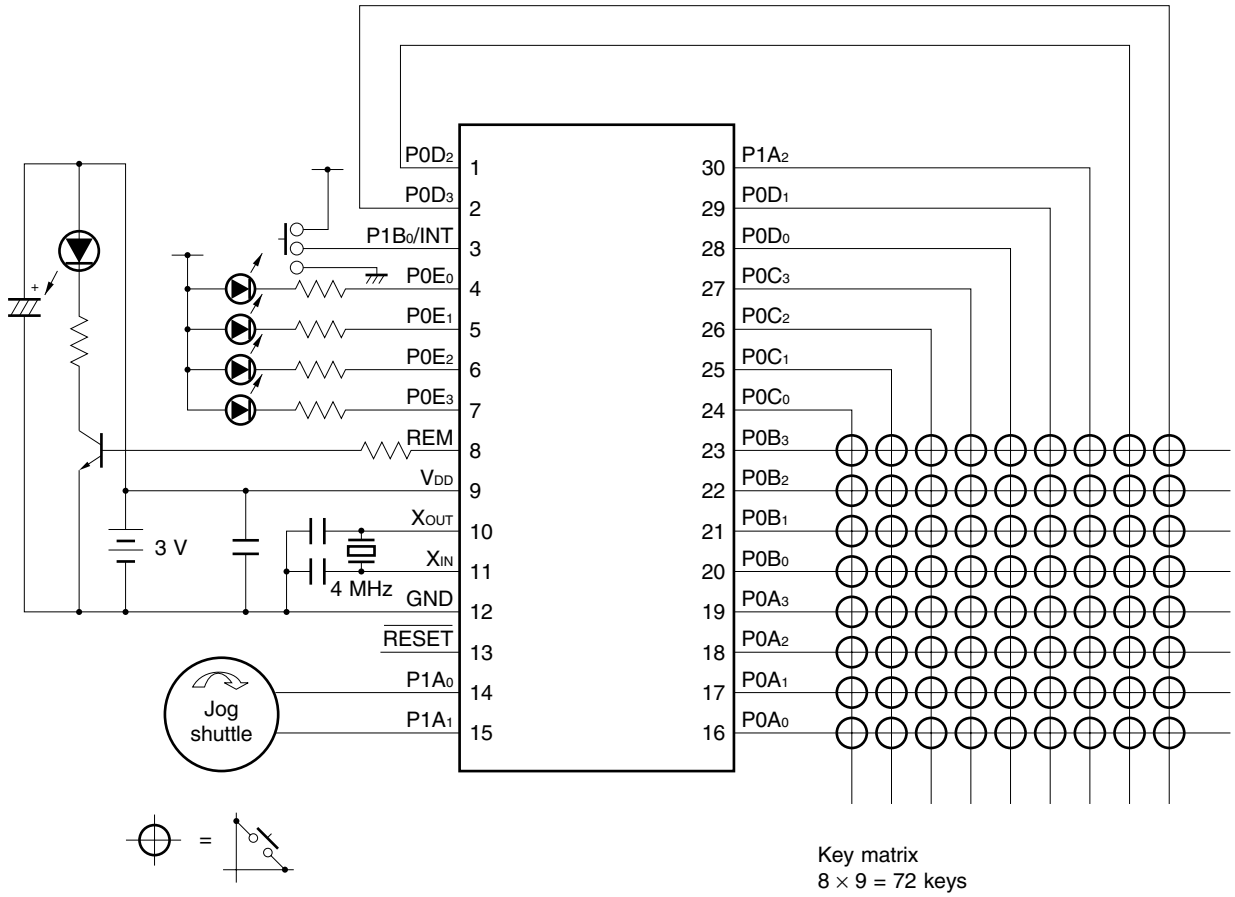
Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
★ CPU clock cycle time[Note]	t_{CY1}	V_{DD} = 2.0 to 3.6 V	3.4		33	μs
(Instruction execution time)	t_{CY2}	V_{DD} = 2.2 to 3.6 V	1.9		33	μs
INT high-/low-level width	t_{INTH}, t_{INTL}		20			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}		10			μs

★ **Note** The CPU clock cycle time (instruction execution time) is determined by the oscillation frequency of the resonator connected and SYSCK (RF: address 02H) of the register file. The figure below shows the CPU clock cycle time t_{CY} vs. supply voltage V_{DD} characteristics (refer to **4. CLOCK GENERATOR**).

t_{CY} vs V_{DD}

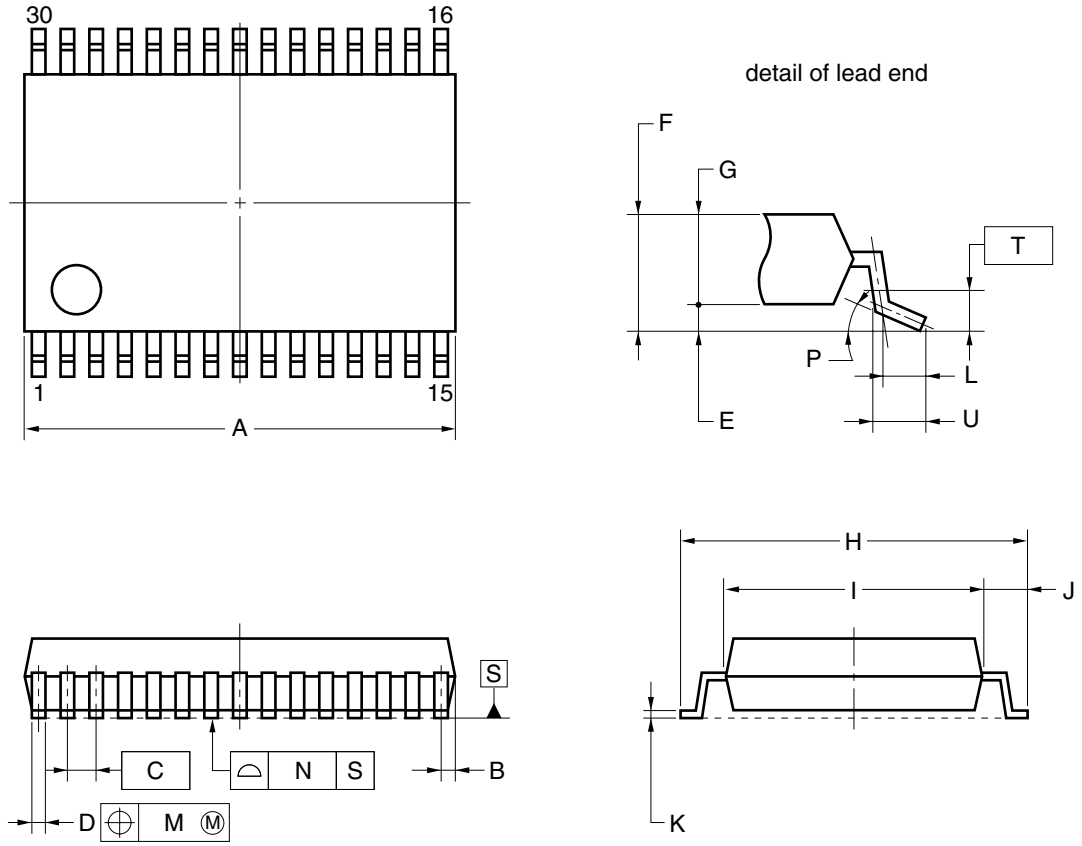


15. APPLICATION CIRCUIT EXAMPLE



16. PACKAGE DRAWING

30-PIN PLASTIC SSOP (7.62 mm (300))



NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	9.85±0.15
B	0.45 MAX.
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25
U	0.6±0.15

S30MC-65-5A4-2

★ 17. RECOMMENDED SOLDERING CONDITIONS

The μPD17240, 17241, 17242, 17243, 17244, 17245, and 17246 should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 17-1. Surface Mounting Type Soldering Conditions

μPD17240MC-×××-5A4: 30-pin plastic SSOP (7.62 mm (300))
μPD17241MC-×××-5A4: 30-pin plastic SSOP (7.62 mm (300))
μPD17242MC-×××-5A4: 30-pin plastic SSOP (7.62 mm (300))
μPD17243MC-×××-5A4: 30-pin plastic SSOP (7.62 mm (300))
μPD17244MC-×××-5A4: 30-pin plastic SSOP (7.62 mm (300))
μPD17245MC-×××-5A4: 30-pin plastic SSOP (7.62 mm (300))
μPD17246MC-×××-5A4: 30-pin plastic SSOP (7.62 mm (300))

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Three times or less	IR35-00-3
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Three times or less	VP15-00-3
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max., Count: Once, Preheating temperature: 120°C max. (package surface temperature)	WS60-00-1
Partial heating	Pin temperature: 300°C max., Time: 3 seconds max. (per pin row)	–

Caution Do not use different soldering methods together (except for partial heating).

APPENDIX A DIFFERENCES BETWEEN μPD17246 AND μPD17P246

The μPD17P246 is equipped with PROM to which data can be written by the user instead of the internal mask ROM (program memory) of the μPD17246.

Table A-1 shows the differences between the μPD17246 and μPD17P246.

The CPU functions and internal hardware of the μPD17P246, 17240, 17241, 17242, 17243, 17244, 17245, and 17246 are identical. Therefore, the μPD17P246 can be used to evaluate the program developed for the μPD17240, 17241, 17242, 17243, 17244, 17245, and 17246 system. Note, however, that some of the electrical specifications such as supply current and low-voltage detection voltage of the μPD17P246 differ from those of the μPD17240, 17241, 17242, 17243, 17244, 17245, and 17246.

Table A-1. Differences Between μPD17246 and μPD17P246

Item \ Product Name	μPD17P246 (μPD17P246M1, 17P246M2)	μPD17246
Program memory	One-time PROM	Mask ROM
	32 KB (16,384 × 16) (0000H to 3FFFH)	
Data memory	447 × 4 bits	
Capacitor for oscillator	• Not provided (μPD17P246M1) • Provided (μPD17P246M2)	Any (mask option)
Low-voltage detector[Note 1]	Provided	Any (mask option)
V_{PP} pin, operation mode select pin	Provided	Not provided
Instruction execution time[Note 2]	4 μs (V_{DD} = 2.2 to 3.6 V)	4 μs (V_{DD} = 2.0 to 3.6 V)
Supply voltage[Note 2]	V_{DD} = 2.2 to 3.6 V	V_{DD} = 2.0 to 3.6 V
Package	30-pin plastic SSOP (7.62 mm (300))	

Notes 1. Although the circuit configuration is identical, the electrical characteristics differ depending on the product.

★ **2.** When fx = 4 MHz and high-speed mode operation is set.

APPENDIX B DEVELOPMENT TOOLS

The following development tools are available to develop the programs for the μPD17246 Subseries.

Hardware

Name	Remarks
In-circuit emulator (IE-17K, IE-17K-ET[Note 1])	The IE-17K and IE-17K-ET are in-circuit emulators used in common with the 17K Series microcontrollers. The IE-17K and IE-17K-ET are connected to the PC-9800 series or IBM PC/AT™ compatible machines as the host machine via RS-232C. By using these in-circuit emulators with a system evaluation board corresponding to the microcomputer, the emulators can emulate the microcomputer. A higher level debugging environment can be provided by using the human interface *SIMPLEHOST*™.
★ EM board (EM-17246[Note 2])	This is an EM board for the μPD17246 Subseries. It can be used alone to evaluate a system or in combination with an in-circuit emulator for debugging.
Emulation probe (EP-17K30GS)	The EP-17K30GS is an emulation probe for a 17K Series 30-pin shrink SOP (MC-5A4). When used with the EV-9500GT-30[Note 3], it connects an EM board to the target system.
Conversion adapter (EV-9500GT-30[Note 3])	The EV-9500GT-30 is a conversion adapter for a 30-pin shrink SOP (MC-5A4). It is used to connect the EP-17K30GS and target system.
PROM programmer (AF-9706[Note 4], AF-9708[Note 4], AF-9709[Note 4])	The AF-9706, AF-9708, and AF-9709 are PROM programmers corresponding to the μPD17P246. By connecting the program adapter PA-17P236 to this PROM programmer, the μPD17P246 can be programmed.
Program adapter (PA-17P236)	The PA-17P236 is an adapter used to program the μPD17P236, and is used in combination with the AF-9706, AF-9708, or AF-9709.

Notes 1. Low-cost model: External power supply type

2. This is a product of Naito Densei Machida Mfg., Co., Ltd. (TEL +81-45-475-4191)

3. Two EV-9500GT-30 units are supplied with the EP-17K30GS. Five EV-9500GT-30 units are optionally available as a set.

4. These are products of Ando Electric Co., Ltd. (TEL: +81-53-576-1560).

Software

Name	Outline	Host Machine	OS	Supply	Order Code
17K assembler (RA17K)	The RA17K is an assembler common to 17K Series products. When developing the programs of devices, RA17K is used in combination with a device file (AS17225).	PC-9800 series	Japanese Windows™	3.5" 2HD	μSAA13RA17K
		IBM PC/AT compatible machine	Japanese Windows	3.5" 2HC	μSAB13RA17K
			English Windows		μSBB13RA17K
★ Device file (AS17246)	The AS17246 is a device file for the μPD17240, 17241, 17242, 17243, 17244, 17245, and 17246, and is used in combination with an assembler for the 17K Series (RA17K).	PC-9800 series	Japanese Windows	3.5" 2HD	μSAA13AS17246
		IBM PC/AT compatible machine	Japanese Windows	3.5" 2HC	μSAB13AS17246
			English Windows		μSBB13AS17246
Support software (*SIMPLEHOST*)	*SIMPLEHOST* is a software package that enables a human interface on Windows when a program is developed by using an in-circuit emulator and a personal computer.	PC-9800 series	Japanese Windows	3.5" 2HD	μSAA13ID17K
		IBM PC/AT compatible machine	Japanese Windows	3.5" 2HC	μSAB13ID17K
			English Windows		μSBB13ID17K

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC Electronics product in your application, please contact the NEC Electronics office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics America, Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01
Fax: 0211-65 03 327

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
  Fax: 091-504 28 60
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
  Fax: 01-30-67 58 99
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
  Fax: 02-66 75 42 99
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45
  Fax: 040-244 45 80
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 80 820
  Fax: 08-63 80 388
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133
  Fax: 01908-670-290

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Shanghai, Ltd.
Shanghai, P.R. China
Tel: 021-6841-1138
Fax: 021-6841-1137

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 6253-8311
Fax: 6250-3583

J02.11