

# MOS INTEGRATED CIRCUIT

# $\mu$ PD17201A, 17207

## 4-BIT SINGLE-CHIP MICROCONTROLLER WITH LCD CONTROLLER/DRIVER AND A/D CONVERTER FOR INFRARED REMOTE CONTROL TRANSMITTERS

### DESCRIPTION

The  $\mu$ PD17201A, 17207 is a 4-bit single-chip microcontroller, used for infrared remote control transmitters, which integrates an LCD controller/driver, A/D converter, and remote controller carrier generator circuit on a single chip.

For the CPU, this microcontroller employs the 17K architecture of the general-purpose register method, and it can directly execute operations between data memory addresses which would have been conventionally executed by an accumulator. In addition, all the instructions are 16-bit, 1-word instructions, enabling efficient programming.

A one-time PROM model,  $\mu$ PD17P207, to which data can be written only once is also available. This one-time PROM model is useful for program evaluation of the  $\mu$ PD17201A or 17207.

**Detailed functionins are described in the following manual. Be sure to read this manual when designing your system.**

$\mu$ PD172xx Subseries User's Manual: IEU-1317

### FEATURES

- 17K architecture : General-purpose register method
- Program memory (ROM) : 3072 x 16 bits ( $\mu$ PD17201A)  
4096 x 16 bits ( $\mu$ PD17207)
- Data memory (RAM) : 336 x 4 bits (including LCD register 36 x 4 bits)
- Infrared remote controller carrier generator (REM output)
- LCD controller/driver : Up to 136 segments can be displayed
  - Common pins : 4 (2 can be used as segment pins)
  - Segment pins : 34
  - Voltage booster circuit for driving LCD : LCD drive voltage can be adjusted from 2.4 to 5.4 V with external resistor
- 8-bit A/D converter : 4 channels (successive approximation method in software)
- 8-bit timer : 1 channel
- Watch timer/watchdog timer : 1 channel ( $\overline{\text{WDOUT}}$  output)
- 3-line serial interface : 1 channel
- External interrupt pin (INT) : 1
- I/O pin : 20 (including INT)
- Instruction execution time : 4  $\mu$ s (main clock:  $f_x = 4$  MHz)  
488  $\mu$ s (subclock:  $f_{XT} = 32.768$  kHz)
- Supply voltage :  $V_{DD} = 2.2$  to 5.5 V (main clock:  $f_x = 4$  MHz)  
:  $V_{DD} = 2.0$  to 5.5 V (subclock :  $f_{XT} = 32.768$  kHz)

Unless otherwise specified, the  $\mu$ PD17207 is treated as the representative model throughout this documents.

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

**APPLICATIONS**

- Infrared remote controller for air conditioner
- Infrared remote controller with LCD display

**ORDERING INFORMATION**

Part Number	Package
$\mu$ PD17201AGF-xxx-3B9	80-pin plastic QFP (14 mm x 20 mm)
$\mu$ PD17207GF-xxx-3B9	80-pin plastic QFP (14 mm x 20 mm)

**Remark** xxx indicates the ROM code number.

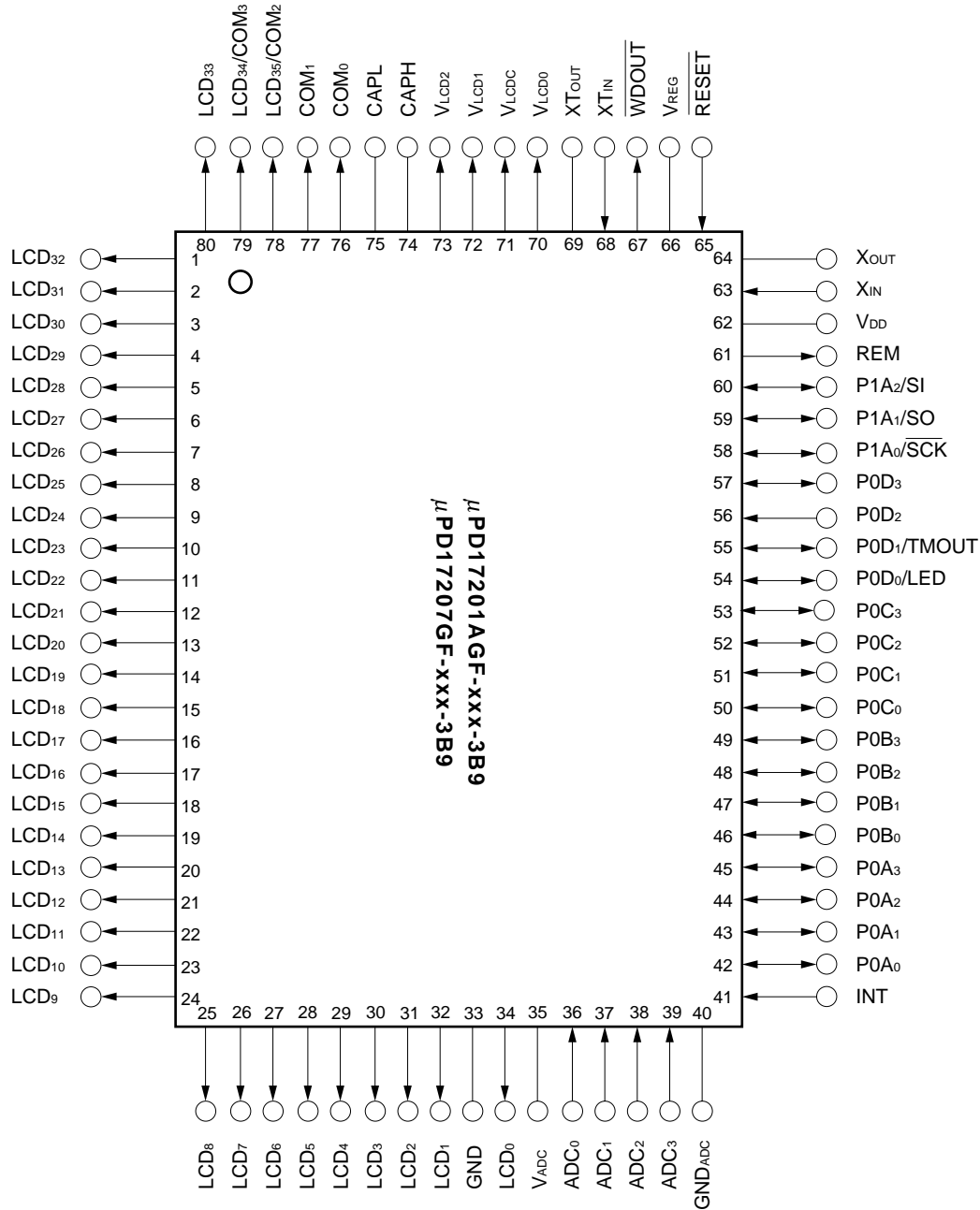
## TABLE OF CONTENTS

<b>1. PIN CONFIGURATION (TOP VIEW)</b> .....	<b>6</b>
<b>2. BLOCK DIAGRAM</b> .....	<b>8</b>
<b>3. PINS FUNCTIONS</b> .....	<b>9</b>
3.1 PIN IDENTIFICATION .....	9
3.2 EQUIVALENT CIRCUITS OF PINS .....	11
3.3 PROCESSING OF UNUSED PINS .....	12
3.4 NOTES ON USING $\overline{\text{RESET}}$ AND INT PINS .....	13
<b>4. MEMORY SPACE</b> .....	<b>14</b>
4.1 PROGRAM COUNTER (PC) .....	14
4.2 PROGRAM MEMORY (ROM) .....	14
4.3 STACK .....	15
4.4 DATA MEMORY (RAM) .....	17
4.5 REGISTER FILE (RF) .....	25
<b>5. PORTS</b> .....	<b>28</b>
5.1 PORT 0A .....	28
5.2 PORT 0B .....	28
5.3 PORT 0C .....	28
5.4 PORT 0D .....	28
5.5 PORT 1A .....	29
5.6 INT PIN .....	29
5.7 PORT CONTROL REGISTER .....	30
<b>6. CLOCK GENERATOR CIRCUIT</b> .....	<b>33</b>
6.1 SWITCHING SYSTEM CLOCK .....	34
6.2 MAIN CLOCK OSCILLATION CONTROL FUNCTION .....	34
<b>7. 8-BIT TIMER AND REMOTE CONTROLLER CARRIER GENERATOR CIRCUIT</b> .....	<b>35</b>
7.1 CONFIGURATION OF THE 8-BIT TIMER (WITH MODULO FUNCTION) .....	35
7.2 FUNCTION OF THE 8-BIT TIMER (WITH MODULO FUNCTION) .....	37
7.3 REMOTE CONTROLLER CARRIER GENERATOR .....	38
<b>8. WATCH TIMER/WATCHDOG TIMER</b> .....	<b>43</b>
8.1 CONFIGURATION OF WATCH TIMER/WATCHDOG TIMER .....	43
8.2 FUNCTION OF WATCH TIMER/WATCHDOG TIMER .....	44
8.3 WATCHDOG TIMER OPERATION TIMING .....	45

<b>9. A/D CONVERTER .....</b>	<b>46</b>
9.1 CONFIGURATION OF A/D CONVERTER .....	46
9.2 FUNCTION OF A/D CONVERTER .....	47
9.3 CONTROL REGISTERS OF A/D CONVERTER .....	48
9.4 OPERATION IN A/D CONVERSION MODE .....	50
9.5 OPERATION IN COMPARE MODE .....	52
<b>10. SERIAL INTERFACE .....</b>	<b>54</b>
10.1 SERIAL INTERFACE FUNCTION .....	54
10.2 SERIAL INTERFACE OPERATION .....	54
<b>11. LCD CONTROLLER/DRIVER .....</b>	<b>58</b>
11.1 CONFIGURATION OF LCD CONTROLLER/DRIVER .....	58
11.2 FUNCTIONS OF LCD CONTROLLER/DRIVER .....	59
11.3 DISPLAY MODE REGISTER .....	59
11.4 LCD REGISTER .....	63
11.5 SEGMENT SIGNALS AND COMMON SIGNALS .....	67
11.6 VOLTAGE BOOSTER CIRCUIT FOR LCD DRIVER .....	70
<b>12. INTERRUPT FUNCTIONS .....</b>	<b>72</b>
12.1 INTERRUPT SOURCES .....	72
12.2 HARDWARE OF INTERRUPT CONTROL CIRCUIT .....	72
12.3 INTERRUPT SEQUENCE .....	77
<b>13. STANDBY FUNCTIONS .....</b>	<b>78</b>
13.1 HALT MODE .....	78
13.2 CONDITIONS OF EXECUTING AN HALT INSTRUCTION .....	80
13.3 STOP MODE .....	81
13.4 CONDITIONS OF EXECUTING AN HALT INSTRUCTION .....	82
<b>14. RESET .....</b>	<b>83</b>
14.1 RESET BY RESET SIGNAL INPUT .....	83
14.2 RESET BY WATCHDOG TIMER (RESET AND WDO $\overline{U}$ T PINS CONNECTED) .....	83
14.3 RESET BY STACK POINTER ( $\overline{R}$ RESET AND $\overline{W}$ DO $\overline{U}$ T PINS CONNECTED) .....	83
<b>15. ASSEMBLER RESERVED WORDS .....</b>	<b>85</b>
15.1 MASK OPTION DIRECTIVES .....	85
15.2 RESERVED SYMBOLS .....	86
<b>16. INSTRUCTION SET .....</b>	<b>94</b>
16.1 OUTLINE OF INSTRUCTION SETS .....	94
16.2 LEGEND .....	95
16.3 LIST OF INSTRUCTION SETS .....	96
16.4 ASSEMBLER (AS17K) EMBEDDED MACROINSTRUCTIONS .....	98
<b>17. ELECTRICAL SPECIFICATIONS .....</b>	<b>99</b>
<b>18. PERFORMANCE CURVE (REFERENCE VALUE) .....</b>	<b>106</b>

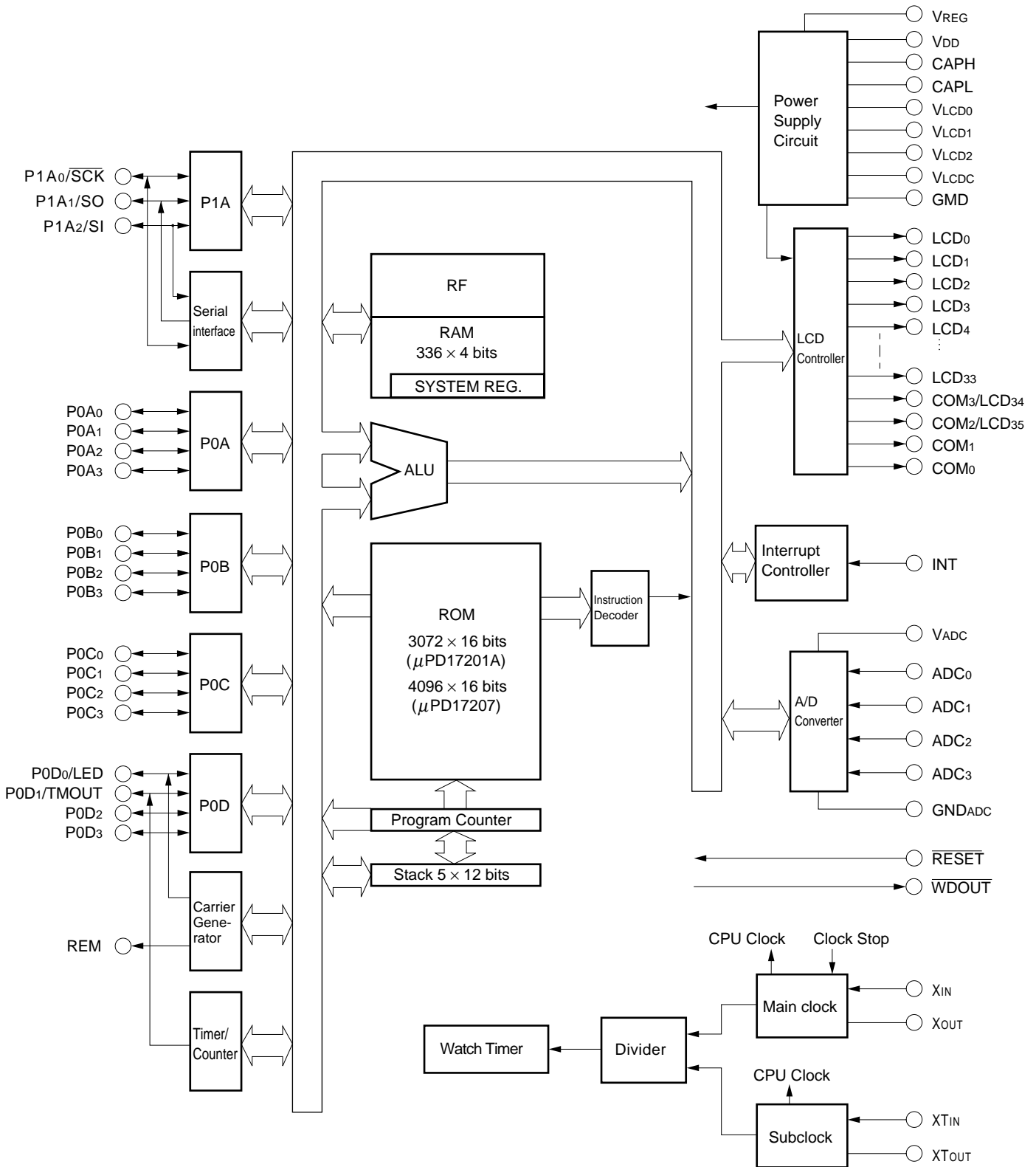
19. EXAMPLE OF APPLICATION CIRCUIT .....	110
20. PACKAGE DRAWINGS.....	111
21. RECOMMENDED SOLDERING CONDITIONS.....	113
APPENDIX A. DIFFERENCES BETWEEN $\mu$ PD17P207 AND $\mu$ PD17201A/17207 .....	114
APPENDIX B. FUNCTIONAL COMPARISON OF $\mu$ PD17201A/17207 RELATED PRODUCTS.....	115
APPENDIX C. DEVELOPMENT TOOLS .....	116

1. PIN CONFIGURATION (TOP VIEW)



Pin Name		REM	: Remote controller transfer output
ADC <sub>0</sub> -ADC <sub>3</sub>	: A/D converter input	RESET	: Reset signal input
CAPH, CAPL	: Booster capacitor connection	SCK	: Serial clock I/O
COM <sub>0</sub> -COM <sub>3</sub>	: LCD common signal output	SI	: Serial data input
GND, GND <sub>ADC</sub>	: Ground	SO	: Serial data output
INT	: External interrupt request signal input	TMOU <sub>T</sub>	: Timer output
LCD <sub>0</sub> -LCD <sub>35</sub>	: LCD segment signal output	V <sub>ADC</sub>	: A/D converter power supply
LED	: Remote controller transfer display output	V <sub>DD</sub>	: Power supply
P0A <sub>0</sub> -P0A <sub>3</sub>	: I/O port	V <sub>LCD0</sub> -V <sub>LCD2</sub>	: LCD driver voltage output
P0B <sub>0</sub> -P0B <sub>3</sub>	: I/O port	V <sub>LCDC</sub>	: LCD driver reference voltage adjustment
P0C <sub>0</sub> -P0C <sub>3</sub>	: I/O port	V <sub>REG</sub>	: Voltage regulator output
P0D <sub>0</sub> -P0D <sub>3</sub>	: I/O port	WDOU <sub>T</sub>	: Overrun detection output
P1A <sub>0</sub> -P1A <sub>2</sub>	: I/O port	X <sub>IN</sub> , X <sub>OUT</sub>	: Main clock oscillator circuit
		X <sub>TIN</sub> , X <sub>TOUT</sub>	: Subclock oscillator circuit

2. BLOCK DIAGRAM





3. PINS FUNCTIONS

3.1 PIN IDENTIFICATION

Pin No.	Symbol	Function	Output Type	On Reset
76 77 78 79 80 1   32 34	COM <sub>0</sub> COM <sub>1</sub> LCD <sub>35</sub> /COM <sub>2</sub> LCD <sub>34</sub> /COM <sub>3</sub> LCD <sub>33</sub> LCD <sub>32</sub>   LCD <sub>1</sub> LCD <sub>0</sub>	Common/segment signal outputs of the LCD driver. These common and segment signal outputs are selected by LCDMD3 to LCDMD0 of the register file. <ul style="list-style-type: none"> <li>• COM<sub>0</sub> to COM<sub>3</sub></li> <li>• Common signal outputs of the LCD driver</li> <li>• LCD<sub>35</sub> to LCD<sub>0</sub></li> <li>• Segment signal outputs of the LCD driver</li> </ul>	CMOS, push-pull	–
33	GND	Device ground	–	–
35	V <sub>ADC</sub>	Positive power supply of the A/D converter (V <sub>ADC</sub> should be equal to V <sub>DD</sub> .)	–	–
36   39	ADC <sub>0</sub>   ADC <sub>3</sub>	Analog inputs of the A/D converter (8-bit resolution)	–	–
40	GND <sub>ADC</sub>	Ground of the A/D converter	–	–
41	INT	External interrupt request signal (Input). The interrupt request is generated at the rising edge of this signal.	–	Input
42   45	P0A <sub>0</sub>   P0A <sub>3</sub>	4-bit I/O port (enabling setting of inputs or outputs in 4-bit units) (Grouped I/O). Each of these pins has a pull-up resistor.	CMOS, push-pull	Input
46   49	P0B <sub>0</sub>   P0B <sub>3</sub>	4-bit I/O port (enabling setting of inputs or outputs in 4-bit units). (Grouped I/O).	N-channel, open-drain	Input
50   53	P0C <sub>0</sub>   P0C <sub>3</sub>	4-bit I/O port (enabling setting of inputs or outputs in 4-bit units). (Grouped I/O).	N-channel, open-drain	Input
54 55 56 57	P0D <sub>0</sub> /LED P0D <sub>1</sub> /TMOUT P0D <sub>2</sub> P0D <sub>3</sub>	Port 0D/LED output or port 0D/8-bit timer output. P0D <sub>0</sub> and LED outputs are switched by NRZEN of the register file. P0D <sub>1</sub> and 8-bit timer outputs are switched by TMOE of the register file. <ul style="list-style-type: none"> <li>• P0D<sub>0</sub> to P0D<sub>3</sub> <ul style="list-style-type: none"> <li>• 4-bit I/O port</li> <li>• Enabling setting of inputs or outputs of each bit (Bitwise I/O)</li> </ul> </li> <li>• LED <ul style="list-style-type: none"> <li>• Outputs NRZ signal in synchronization with infrared remote controller signal (REM)</li> <li>• Outputs high level while remote controller carrier is output from REM pin</li> </ul> </li> <li>• TMOUT <ul style="list-style-type: none"> <li>• Output of the 8-bit timer</li> </ul> </li> </ul>	CMOS, push-pull	Input

(to be cont'd)

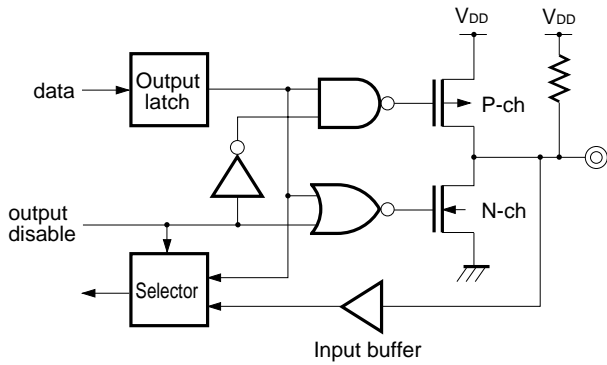
(cont'd)

Pin No.	Symbol	Function	Output Type	On Reset
58	P1A <sub>0</sub> /SCK	Port 1A or serial interface.	CMOS, push-pull	Input
59	P1A <sub>1</sub> /SO	Port 1A and serial interface are switched by SIOEN of the register file.		
60	P1A <sub>2</sub> /SI	<ul style="list-style-type: none"> <li>• P1A<sub>0</sub> to P1A<sub>2</sub> <ul style="list-style-type: none"> <li>• 3-bit I/O port</li> <li>• Enabling setting of inputs or outputs of 3 bits (Grouped I/O).</li> </ul> </li> <li>• SCK, SO, SI                             <ul style="list-style-type: none"> <li>• SCK: Serial clock I/O</li> <li>• SO : Serial data output</li> <li>• SI : Serial data input</li> </ul> </li> </ul>		
61	REM	Signal output to an infrared remote controller. Active-high output.	CMOS, push-pull	Low-level output
62	V <sub>DD</sub>	Positive power supply	–	–
63	X <sub>IN</sub>	These pins are connected to a 4-MHz ceramic or crystal oscillator for main clock oscillation.	–	(Oscillation stops)
64	X <sub>OUT</sub>			
65	RESET	System reset input. System is reset when low level is input to this pin. While this pin is low, oscillation of main clock is stopped. A pull-up resistor can be connected by mask option.	–	Input
66	V <sub>REG</sub>	Output of the voltage regulator for the subclock oscillation circuit. Connect external 0.1-μF capacitor to this pin when using the subclock.	–	–
67	WDOUT	Output for detection of a program overrun. This pin outputs a low level when an overflow in the watchdog timer or an overflow/underflow in the stack is detected. Connect this pin to the RESET pin	N-ch open drain	High- impedance
68	XT <sub>IN</sub>	These pins are connected to a 32.768-kHz crystal oscillator for subclock oscillation.	–	(Oscillates)
69	XT <sub>OUT</sub>			
71	V <sub>LDC</sub>	Input to regulate the reference voltage to LCD driver.	–	–
70	V <sub>LCD0</sub>	Reference voltage outputs to LCD driver.	–	–
72	V <sub>LCD1</sub>	• V <sub>LCD0</sub> : Reference voltage output		
73	V <sub>LCD2</sub>	• V <sub>LCD1</sub> : Doubler output (Two times the reference voltage) • V <sub>LCD2</sub> : Tripler output (Three times the reference voltage)		
74	CAPH	These pins are connected to a capacitor to boost the LCD drive voltage.	–	–
75	CAPL			

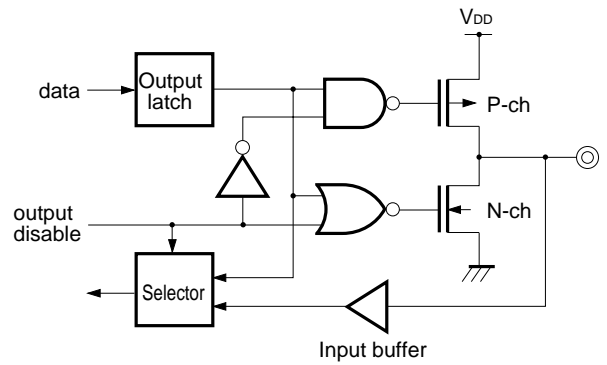
3.2 EQUIVALENT CIRCUITS OF PINS

The followings are equivalent circuits (partially simplified) of the respective pins of the μPD17207.

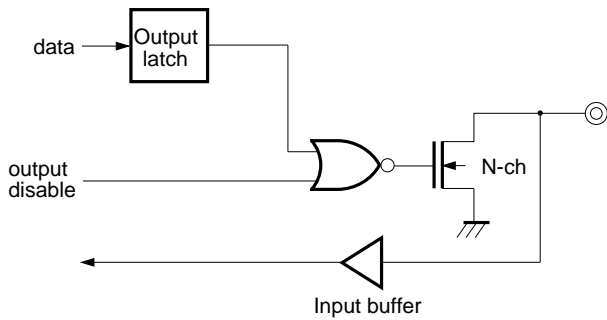
(1) P0A



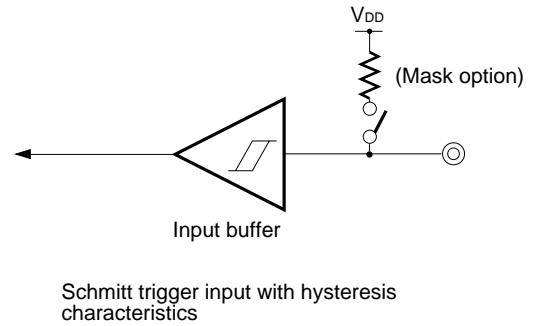
(4) P0D, P1A



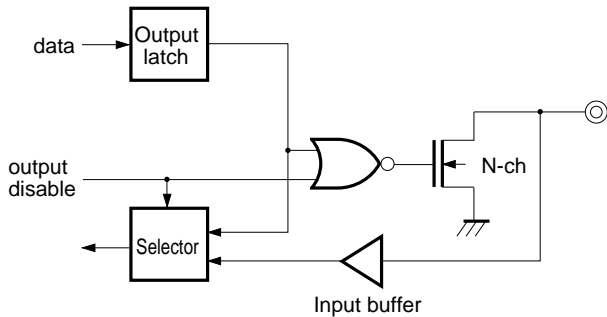
(2) P0B



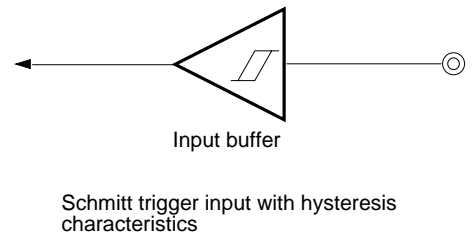
(5) RESET



(3) P0C



(6) INT



3.3 PROCESSING OF UNUSED PINS

Process unused pins as follows:

★

Table 3-1 Processing of Unused Pins

(a) Port pins

Pin Name		Recommended Processing of Unused Pins	
		Internally	Externally
Input mode	P0A	(Connect pull-up resistor.)	Open.
	P0C	–	Directly connect to GND.
	P0D, P1A	–	Connect each pin to V <sub>DD</sub> or GND via resistor <sup>Note</sup> .
Output mode	P0A (CMOS port)	Outputs high level.	Open.
	P0D, P1A (CMOS port)	–	
	P0B, P0C (N-ch open-drain port)	Outputs low level.	

**Note** When pulling this pin up (connecting the pin to V<sub>DD</sub> via resistor) or down (connect the pin to GND via resistor), exercise care not to decrease the drive capability or increase the power consumption of the port. When a high resistance is used for pulling up or down, make sure that noise is not superimposed on the pin.

(b) Pins other than port pins

Pin Name	I/O Format	Recommended Processing of Unused Pin
ADC <sub>0</sub> -ADC <sub>3</sub>	Input	Directly connect to GND.
CAPH, CAPL	Output	Open
COM <sub>0</sub> , COM <sub>1</sub> , COM <sub>2</sub> /LCD <sub>35</sub> , COM <sub>3</sub> /LCD <sub>34</sub>	Output	Open
INT <sup>Note</sup>	Input	Directly connect to GND.
LCD <sub>0</sub> -LCD <sub>33</sub>	Output	Open
REM	Output	Open
V <sub>ADC</sub>	–	Directly connect to V <sub>DD</sub> .
V <sub>LCD0</sub> -V <sub>LCD2</sub>	Output	Open
V <sub>LCDC</sub>	–	Directly connect to V <sub>DD</sub> or V <sub>LCD0</sub> .
WDOUT	Output	Directly connect to GND.
X <sub>IN</sub> , XT <sub>IN</sub>	Input	Directly connect to GND.
X <sub>OUT</sub>	–	Directly connect to V <sub>DD</sub> .
XT <sub>OUT</sub>	–	Directly connect to V <sub>REG</sub> .

**Note** Because the INT pin is also used as a test mode setting pin, directly connect this pin to GND when it is not used.

- Cautions**
1. It is recommended to fix the input or output mode and the output level of the pin by repeatedly setting them in each loop of the program.
  2. Stop the voltage booster circuit by using the display mode register when the LCD driver/controller is not in use.

**3.4 NOTES ON USING  $\overline{\text{RESET}}$  AND INT PINS**

In addition to the functions shown in 3.1 PIN IDENTIFICATION, the  $\overline{\text{RESET}}$  and INT pins also have a function to set a test mode (for IC testing) in which the internal operations of the  $\mu$ PD17207 are tested.

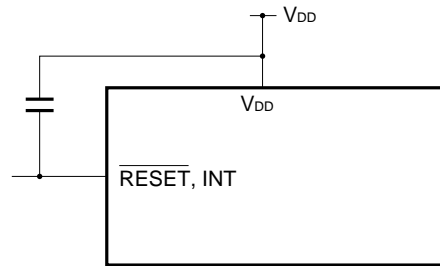
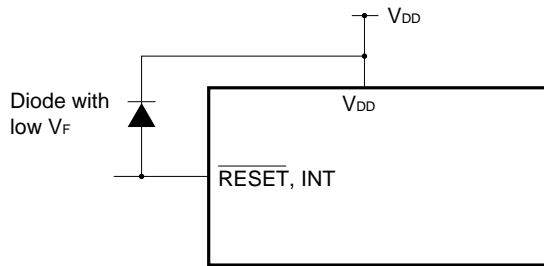
When a voltage higher than  $V_{DD}$  is applied to either of these pins, the test mode is set. This means that, even during ordinary operation, the  $\mu$ PD17207 may be set in the test mode if a noise exceeding  $V_{DD}$  is applied.

For example, if the wiring length of the  $\overline{\text{RESET}}$  or INT pin is too long, noise superimposed on the wiring line of the pin may cause the above problem.

Therefore, keep the wiring length of these pins as short as possible to suppress the noise; otherwise, take noise preventive measures as shown below by using external components.

- Connect diode with low  $V_F$  between  $V_{DD}$  and  $\overline{\text{RESET}}$ /INT pin

- Connect capacitor between  $V_{DD}$  and  $\overline{\text{RESET}}$ /INT pin



#### 4. MEMORY SPACE

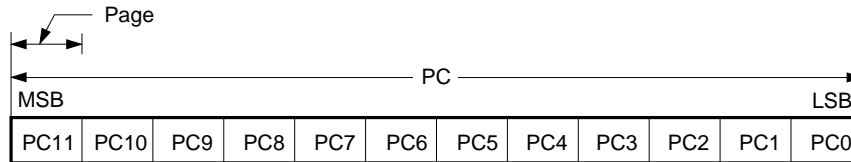
##### 4.1 PROGRAM COUNTER (PC)

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

The program counter is a 12-bit binary counter as shown in Fig. 4-1.

Its contents are initialized to address 0000H at reset.

Fig. 4-1 Configuration of Program Counter



##### 4.2 PROGRAM MEMORY (ROM)

The configuration of the program memory of the μPD17201A/17207 is as follows:

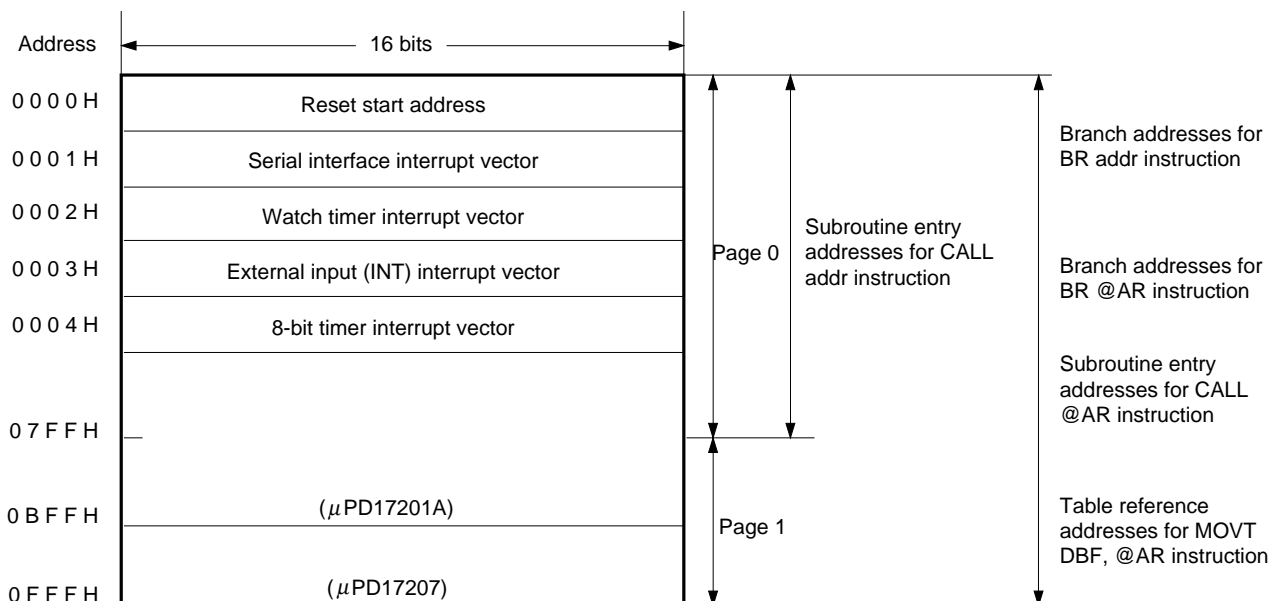
Part Number	Capacity	Address
μPD17201A	3072 x 16 bits	0000H-0BFFH
μPD17207	4096 x 16 bits	0000H-0FFFH

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

The program memory is addressed by the program counter.

Fig. 4-2 shows the program memory map. The entire range of the program memory can be addressed by the BD addr, BR @AR, CALL @AR, MOV T 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.

Fig. 4-2 Program Memory Map



4.3 STACK

A stack is a register 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 accepted.

4.3.1 Stack Configuration

A stack consists of a stack pointer (a 3-bit binary counter), five 12-bit address stack registers (ASR), and three 7-bit interrupt stack registers (INTSK). Refer to Fig. 4-3.

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

The value of this pointer is initialized to 5H at reset.

When the value of the stack pointer is 6H or 7H, the  $\overline{WDOOUT}$  pin goes low.

Fig. 4-3 Stack Configuration

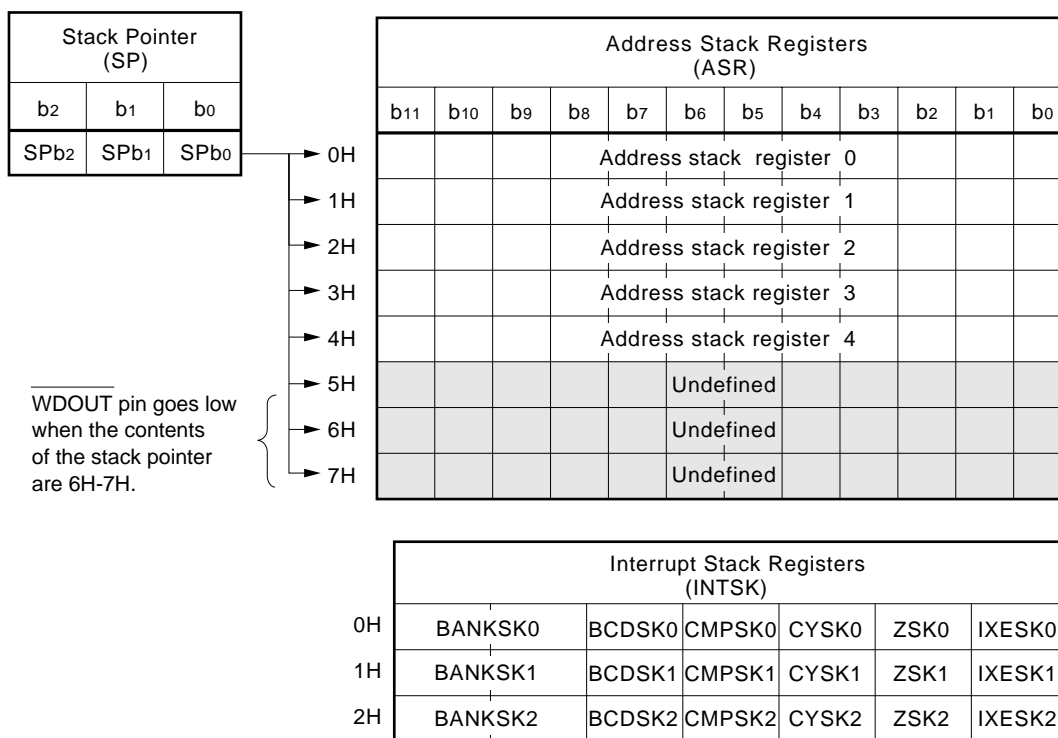
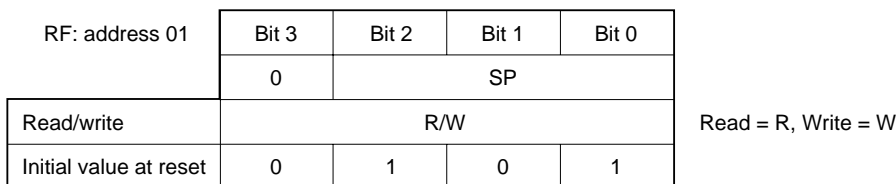


Fig. 4-4 Stack Pointer

★



**4.3.2 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 accepted. It also stores the contents of the address registers (ARs) when a stack manipulation instruction (PUSH AR) is executed.

**The WDO $\overline{\text{UT}}$  pin goes low if a subroutine call or interrupt exceeding 5 levels is executed.**

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

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

**4.3.3 Stack Pointer (SP) and Interrupt Stack Pointer**

Table 4-1 shows the operations of the stack pointer (SP).

The stack pointer can take eight values, 0H-07. Because there are only five stack registers available, however, **the WDO $\overline{\text{UT}}$  pin goes low if the value of SP is 6 or greater.**

**Table 4-1 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	-1	0
When Interrupt Is Accepted	-1	-1
RET RETSK MOVT DBF, @AR (2nd Instruction Cycle) POP AR	+1	0
RETI	+1	+1



#### 4.4. DATA MEMORY (RAM)

Data memory (random access memory) stores data for operations and control. It can be read-/write-accessed by instructions.

##### 4.4.1 Memory Configuration

Figure 4-4 shows the configuration of the data memory (RAM).

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

In each bank, every 4 bits of data is 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 (= a “nibble”).

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

(1) System register (SYSREG)

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

(2) Data buffer (DBF)

A data buffer is resident on addresses 0CH to 0FH (4 nibbles long) of bank 0 of data memory.

The reset value is 0320H.

(3) General register (GR)

A general register is resident on any row (16 nibbles long) of any bank of data memory.

The row address of the general register is indicated by the general pointer (RP) in the system register (SYSREG).

(4) LCD segment data register (LCD register)

A register sets the segment output data of LCD.

Refer to **11. LCD CONTROLLER/DRIVER**.

An LCD segment data register is resident on addresses 40H to 63H (36 nibbles long) of BANK0 of data memory.

(5) Port register

A port data register is resident on addresses 70H to 73H (12 nibbles) of each bank of data memory.

However, addresses 71H to 73H of BANK1 and addresses 70H to 73H of BANK2 are assigned nothing.

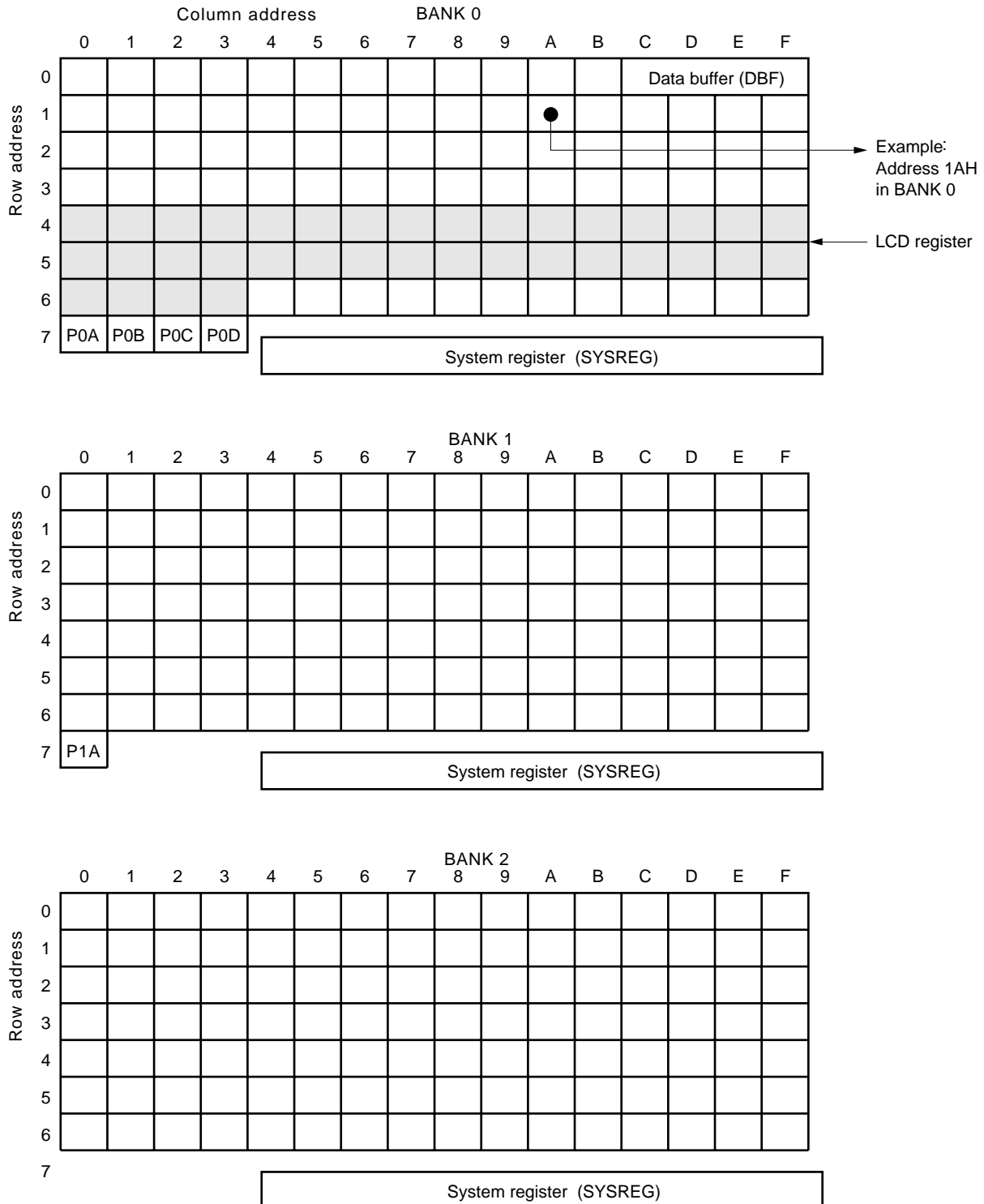
Therefore, a port data register is substantially 4 nibbles long.

The reset value is 0.

(6) General-purpose data memory

The general-purpose data memory area is an area of the data memory excluding the system register area, the LCD register area, and the port register area. This memory area has a total of 300 nibbles (76 nibbles in BANK0 and 224 nibbles in BANK1 and BANK2).

Fig. 4-5 Configuration of Data Memory



**4.4.2 System Registers (SYSREG)**

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

The system registers include the following registers:

- Address registers (AR0-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)

**Fig. 4-6 Configuration of System Register**

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)			General register pointer (RP)		Program status word (PSWORD)
Symbol	AR 3	AR 2	AR 1	AR 0	WR	BANK	IXH	IXM	IXL	RPH	RPL	PSW
							MPH	MPL				
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 0 0	(AR)				(BANK) 0 0	M P E	0 0	(IX)	0 0	(RP)	B C C I C M Y Z D P X E
Initial Value At Reset	0 0 0 0	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

#### 4.4.3 General Register (GR)

A general register is a 16-word 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 4-7 shows the configuration of the general register.

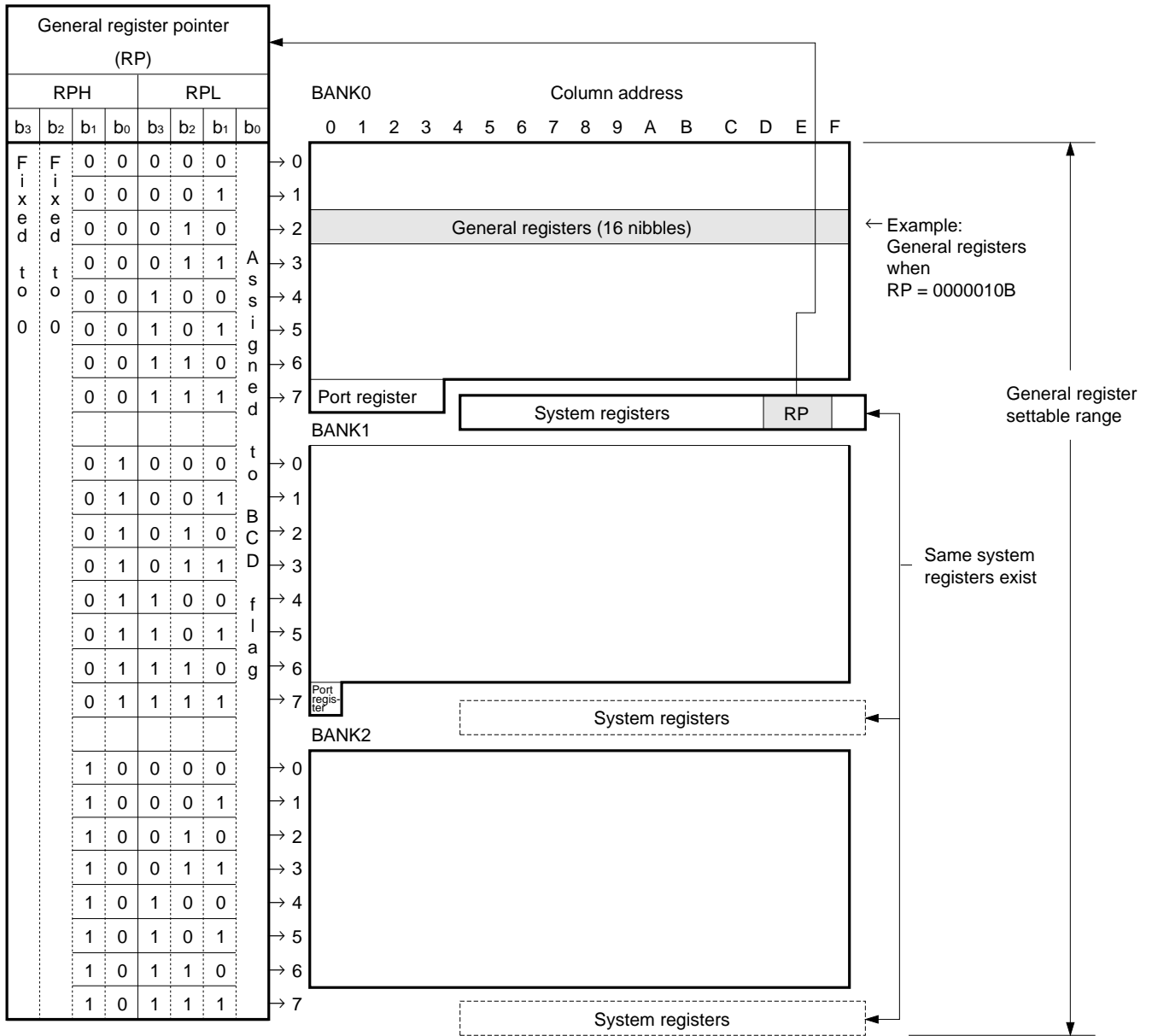
A general register occupies 16 nibbles (16 x 4 bits) on a selected row address of the data memory.

The row address is selected by the general register pointer (RP) of the system register. The RP having five significant bits can point to any row address in the range of 0H to 7H of each bank (BANK0 to BANK2).

(2) Functions of the general register

The 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 register enables arithmetic operation and data transfer between two locations of the data memory. Similarly, the general register can be accessed by a Data Memory Manipulation instruction as it is a part of the data memory.

Fig. 4-7 Configuration of General Registers



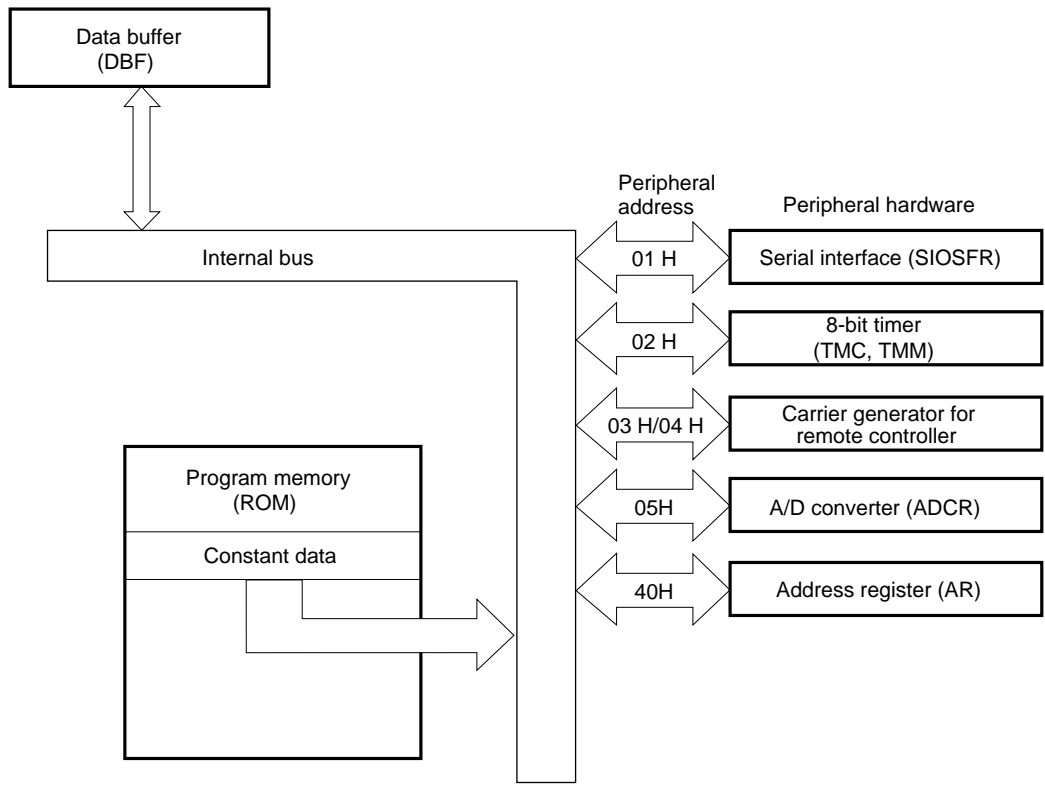
4.4.4 Data Buffer (DBF)

The data buffer on the data memory is used for data transfer to and from peripheral hardware and for storage of data during table reference.

(1) Functions of the Data Buffer

The data buffer has two major functions: a function to transfer to and from hardware and a function to read constant data from the program memory (for table reference). Figure 4-8 shows the relationship between the data buffer and peripheral hardware.

Fig. 4-8 Data Buffer and Peripheral Hardware



**Table 4-2 Relations between Peripheral Hardware and Data Buffer**

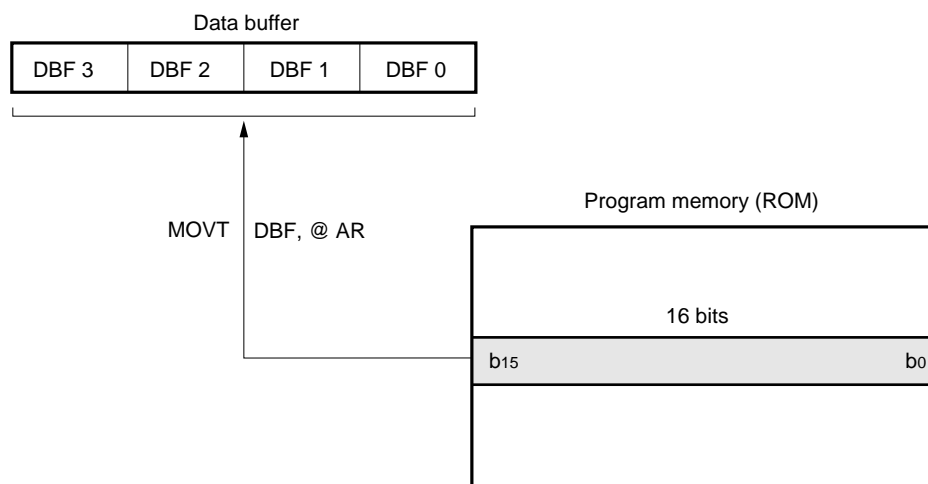
Peripheral Hardware	Peripheral Registers Transferring Data with Data Buffer				
	Name	Symbol	Peripheral Address	Data Buffer Used	Execution of Put/Get
Serial Interface	Shift Register	SIOSFR	01H	DBF0, DBF1	Both PUT & GET
8-bit Timer	8-bit counter	TMC	02H	DBF0, DBF1	Only GET
	8-bit modulo register	TMM	02H	DBF0, DBF1	Only PUT
Remote Controller Carrier Generator	NRZ low-level period setting modulo register	NRZLTMM	03H	DBF0, DBF1	Both PUT & GET
Circuit	NRZ high-level period setting modulo register	NRZHTMM	04H	DBF0, DBF1	PUT (Clears bits 2 and 3 of DBF1 to 0.) GET (Always clears bits 2 and 3 of DBF1 to 0.)
A/D Converter	A/D converter internal reference voltage setting register	ADCR	05H	DBF0, DBF1	Both PUT & GET
Address Register	Address register	AR	40H	DBF0-DBF3	PUT (Bits 0-3 of AR3 are don't care.) GET (Always clears bits 0-3 of AR3 to 0.)

(2) Table reference

A MOV<sub>T</sub> instruction reads constant data from a specified location of the program memory (ROM) and sets it in the data buffer.

The function of the MOV<sub>T</sub> instruction is explained below.

MOV<sub>T</sub> 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) Note 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 the read-only register.  
If the unused address is read, an undefined value is read. Nothing changes even if data is written to that address.
- Using 17K Series 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 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.



**4.5 REGISTER FILE (RF)**

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

These registers can be controlled by dedicated instructions PEEK and POKE, and the embedded macro instructions of AS17K, SETn, CLRn, and INITFLG.

**4.5.1 Configuration of Register File**

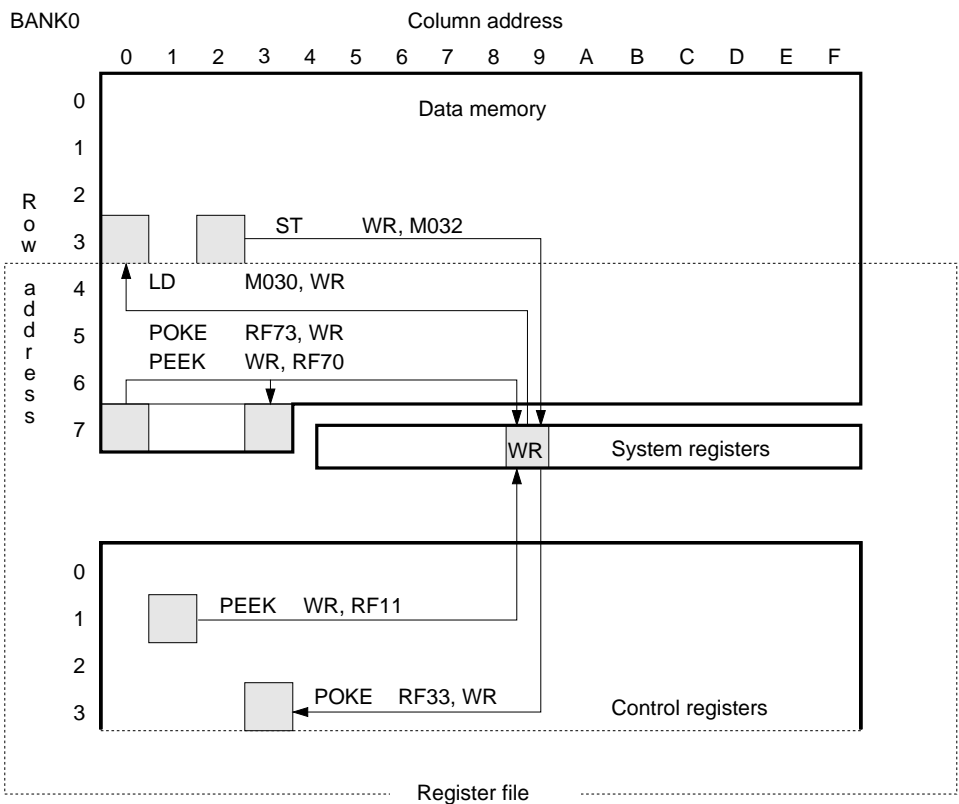
Fig. 4-9 shows the configuration of the register file and how the register file is accessed by the PEEK and POKE instructions.

The control registers are controlled by using dedicated instructions PEEK and POKE. Since the control registers are assigned to addresses 00H-3FH regardless of the bank, the addresses 00H-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 the addresses 00H-3FH of the control registers and 40H-7FH of the general-purpose data memory. The register file consists of these addresses.

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

**Fig. 4-9 Configuration of Register File and Accessing Register File by PEEK and POKE Instructions**



#### 4.5.2 Control Registers

The control registers consists of a total of 64 nibbles (64 x 4 bits) of the addresses 00H-3FH of the register file. Of these, however, only 20 nibbles are actually used. The remaining 44 nibbles are unused registers that are inhibited 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 17K series assembler, 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 Fig. 15-1 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 : Complements flag  
INITFLG : Initializes flag

#### 4.5.3 Notes on Using Register Files

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

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

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

- When device operates  
Nothing changes even if data is written to the read-only register.  
If the unused register is read, an undefined value is read; nothing is changed even if data is written to this register.
- Using 17K series assembler  
An error occurs if an instruction is executed to write data to the read-only register.  
An error also occurs if an instruction is executed to read or write the unused address.
- When an in-circuit emulator (IE-17K or IE-17K-ET) is used (when instruction is executed for patch processing)  
Nothing changes even if data is written to the read-only register, and an error does not occur.  
An undefined value is read if the 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 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 the addresses 80H-BFH of BANK0.

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

## 5. PORTS

### 5.1 PORT 0A

This port is a 4-bit I/O port. The four bits of this port are assigned all inputs or all outputs. This assignment is performed by P0AGIO of the register file. Transferring data from and to this port is performed via the P0A port register (address 70H of BANK0).

This port is set in the input mode at reset.

This port can release the standby mode when the standby mode has been set and if all the bits of the port are not set at high level.

This port is connected to the pull-up resistor regardless of whether the input or output mode is specified.

### 5.2 PORT 0B

This port is a 4-bit I/O port. The four bits of this port are assigned all inputs or all outputs. This assignment is performed by P0BGIO of the register file. Transferring data from and to this port is performed via the P0B port register (address 71H of BANK0).

This port is set in the input mode at reset.

This port can release the standby mode when the standby mode has been set and if all the bits of the port are not at low level.

In the output mode, this port requires an external pull-up resistor because it works as an N-ch open-drain output.

### 5.3 PORT 0C

This port is a 4-bit I/O port. The four bits of this port are assigned all inputs or all outputs. This assignment is performed by P0CGIO of the register file. Transferring data from and to this port is performed via the P0C port register (address 72H of BANK0).

This port is set in the input mode at reset.

In the output mode, this port requires an external pull-up resistor because it works as an N-ch open-drain output.

### 5.4 PORT 0D

This port works as a 4-bit I/O port, an LED output, and an external signal output for the 8-bit timer.

One of these functions is selected by NRZEN and TMOE of the register file.

(1) Using the whole port as a 4-bit I/O port

The pins of this port can be individually assigned input or output. This assignment is performed by P0DBIO3 to P0DBIO0 of the register file. Transferring data from and to this port is performed via the P0D (address 73H of BANK0).

(2) Using the P0D<sub>0</sub> pin as an LED output

The LED output pin and the I/O port (P0D<sub>0</sub>) pins are selected by NRZEN. The LED output pin outputs an NRZ signal in synchronization with the REM output.

(3) Using the P0D<sub>1</sub> pin as an external signal output for the 8-bit timer

The external signal output pin for the 8-bit timer and the I/O port (P0D<sub>1</sub>) pins are selected by TMOE.

**5.5 PORT 1A**

This port works as a 3-bit general I/O port and as serial interface. The I/O port or serial interface is selected by SIOEN of the register file.

- (1) Using the port 1A as a 3-bit I/O port

The three bits of the port 1A can be assigned all inputs or all outputs. This assignment is performed by P1AGIO of the register file. Transferring data from and to this port is performed via the P1A (address 70H of BANK1).

- (2) Using port 1A as serial interface

Serial interface or the I/O port (P1A<sub>0</sub>, P1A<sub>1</sub>, and P1A<sub>2</sub>) is selected by SION.

**5.6 INT PIN**

This pin inputs an external interrupt request signal. At the rising edge of the signal input to this pin, the IRQ flag (RF: address 3DH, bit 1) is set.

The status of the pin can be read by using the INT flag (RF: address 0FH, bit 0). When a high level is input to the INT pin, the INT flag is set to “1”; when a low level is input, the INT flag is reset to “0” (refer to **Fig. 12-1 INT Flag**).

**Table 5-1 Relations between Port Registers and Pins**

Bank	Add-ress	Port	Bit		Output Format	Read Contents		Written Contents		At Reset
						Input mode	Output mode	Input mode	Output mode	
0	70H	Port 0A	b <sub>3</sub>	P0A <sub>3</sub>	CMOS push-pull	Pin status	Output latch	Output latch	Input mode (w/pull-up resistor)	
			b <sub>2</sub>	P0A <sub>2</sub>						
			b <sub>1</sub>	P0A <sub>1</sub>						
			b <sub>0</sub>	P0A <sub>0</sub>						
	71H	Port 0B	b <sub>3</sub>	P0B <sub>3</sub>	N-ch open-drain		Pin status			
			b <sub>2</sub>	P0B <sub>2</sub>						
			b <sub>1</sub>	P0B <sub>1</sub>						
			b <sub>0</sub>	P0B <sub>0</sub>						
	72H	Port 0C	b <sub>3</sub>	P0C <sub>3</sub>	N-ch open-drain		Output latch			
			b <sub>2</sub>	P0C <sub>2</sub>						
			b <sub>1</sub>	P0C <sub>1</sub>						
			b <sub>0</sub>	P0C <sub>0</sub>						
73H	Port 0D	b <sub>3</sub>	P0D <sub>3</sub>	CMOS push-pull	Output latch					
		b <sub>2</sub>	P0D <sub>2</sub>							
		b <sub>1</sub>	P0D <sub>1</sub> <sup>Note1</sup>							
		b <sub>0</sub>	P0D <sub>0</sub> <sup>Note1</sup>							
1	70H	Port 1A	b <sub>3</sub>	—	CMOS push-pull	Input mode				
			b <sub>2</sub>	P1A <sub>2</sub> <sup>Note2</sup>						
			b <sub>1</sub>	P1A <sub>1</sub> <sup>Note2</sup>						
			b <sub>0</sub>	P1A <sub>0</sub> <sup>Note2</sup>						

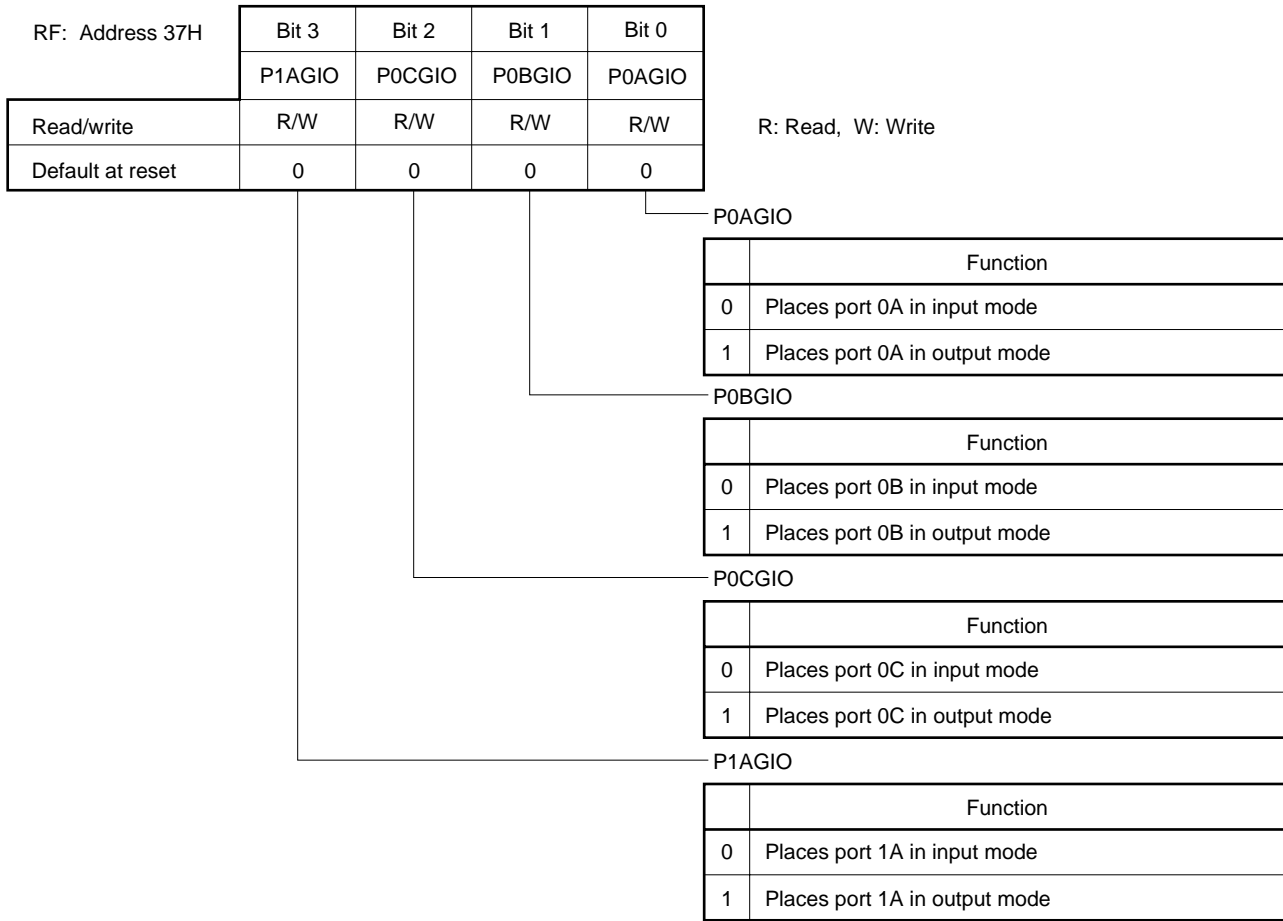
- Notes**
1. When the NRZEN and TMOE flags are set to 1, the output latch is accessed both when these port pins are read and when they are written, regardless of whether the input or output mode is set.
  2. When the SIOEN flag is set to 1, these pins serve as serial interface pins. In this case, the statuses of the pins are read when the pins are read, regardless of the input or output mode. Data written to these pins is invalid.

5.7 PORT CONTROL REGISTER

5.7.1 Switching between Input and Output of Grouped I/O Port

A grouped I/O port is a port whose four bits are assigned all inputs or all outputs at a time. Grouped I/O ports are P0A, P0B, P0C, P0D, and P1A. Their selection of inputs or outputs is performed by the following I/O control register. When the bits of each port are assigned from inputs to outputs, its output latch is output to the port.

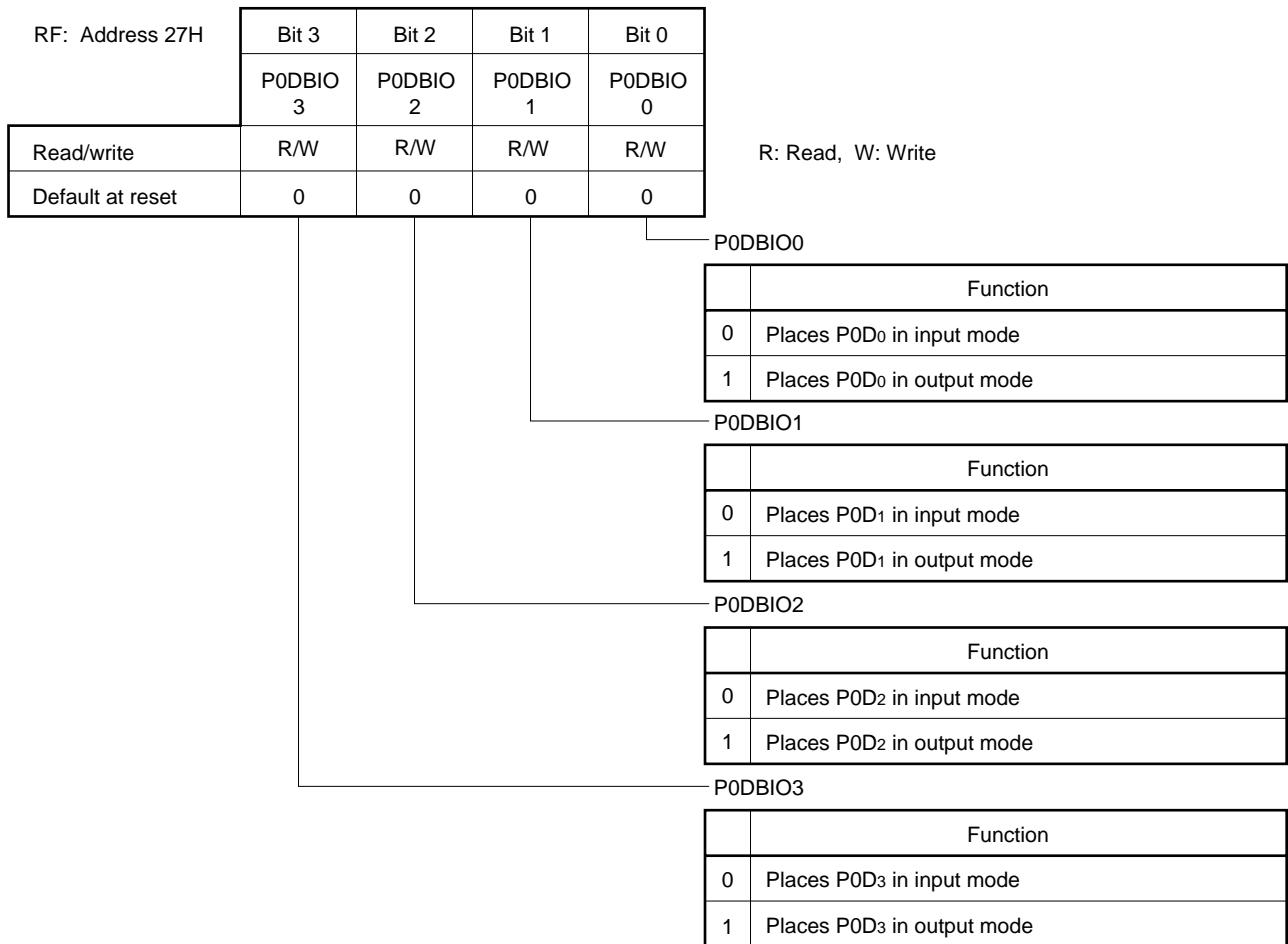
Fig. 5-1 I/O Control Register for Grouped I/O Ports



**5.7.2 Switching between Input and Output of Bitwise I/O Port**

A bitwise I/O port is a port whose four bits are individually assigned inputs or output. The μPD17207 supports only one bitwise I/O port: P0D. The bitwise input/output selection is performed by the following I/O control register. When the bits of this port are assigned from inputs to outputs, output latches of P0A, P0B, P0C, P0D, and P1A are output to the corresponding ports.

**Fig. 5-2 I/O Control Register for Bitwise I/O Ports**



**5.7.3 Switching among Port, Timer Output, and LED Output**

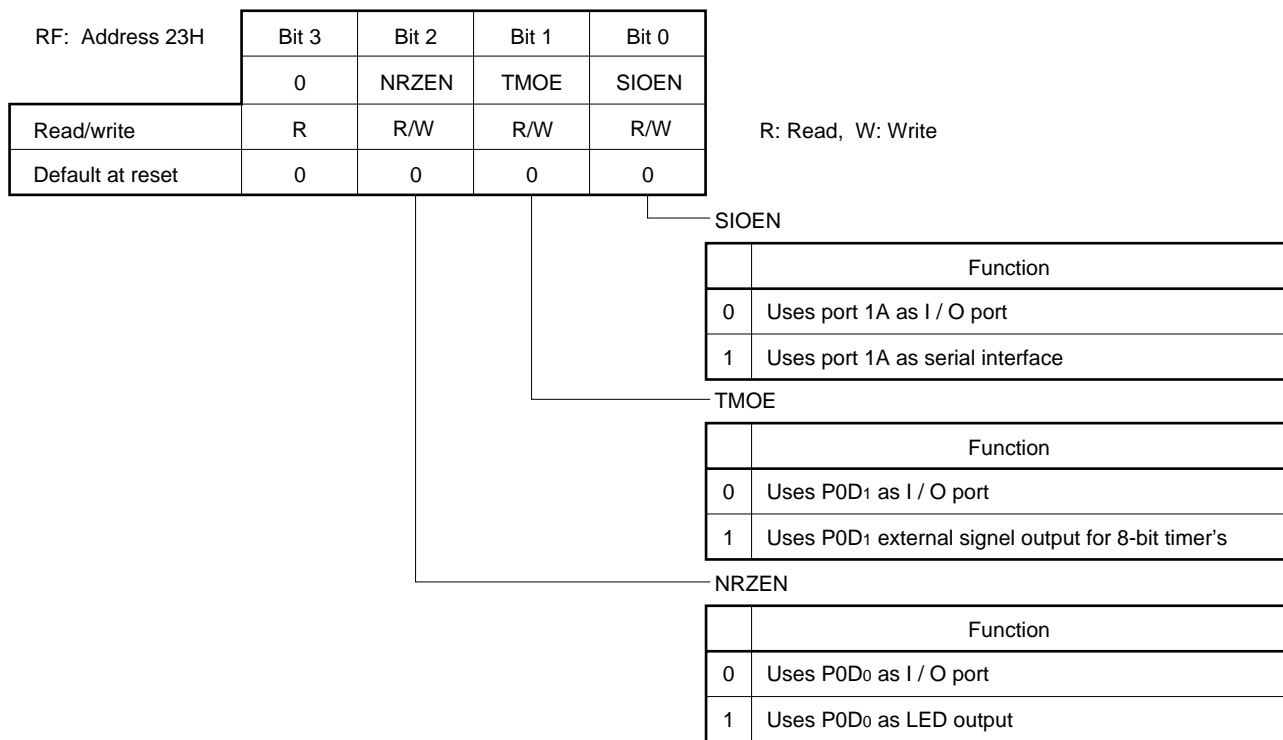
The functions of port 0D (port, timer output, and LED output) are selected by settings of the TMOE and NRZEN bits of register file (TMOE for P0D<sub>1</sub> and NRZEN for P0D<sub>0</sub>). Refer to Fig. 5-3.

**5.7.4 Switching between Port and Serial Interface**

The functions of port 1A (port and serial interface) are selected by settings of the SIOEN bit of register file. Refer to Fig. 5-3.

The mode of serial interface is controlled by SIOTS (bit 3 of address 22H), SIOHIZ (bit 2 of address 22H), SIOCK1 (bit 1 of address 22H), and SIOCK0 (bit 0 of address 22H) of register file.

**Fig. 5-3 Input/Output Control Register for Selection of Port, Timer Output, LED Output, and Serial Interface**





### 6. CLOCK GENERATOR CIRCUIT

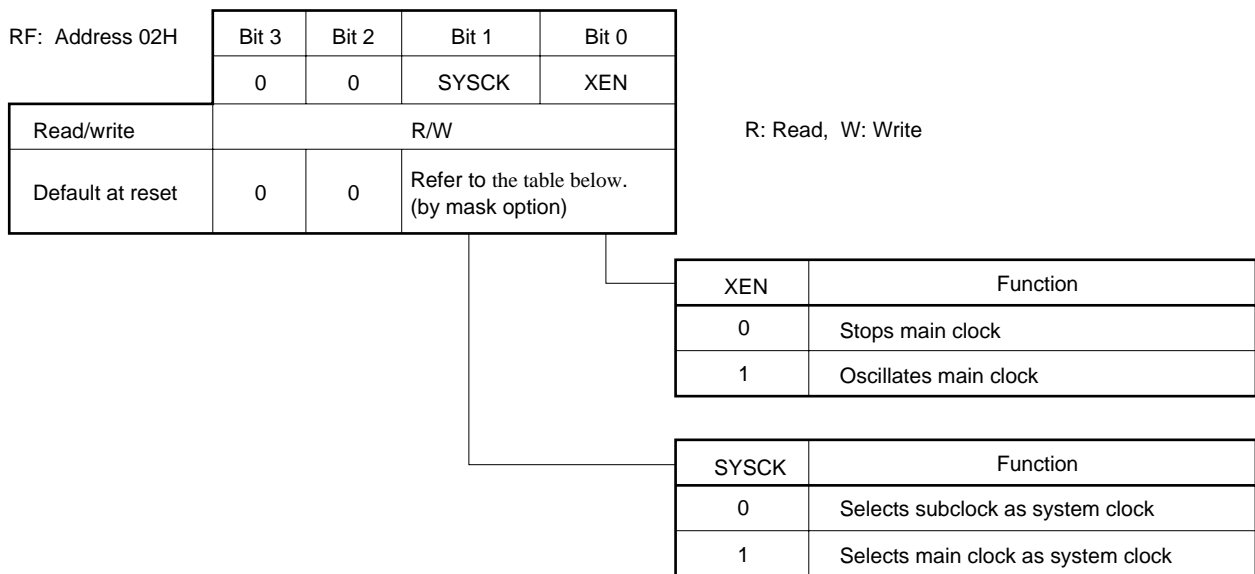
The μPD17207 contains two types of oscillator circuits: the main clock (X) and the subclock (XT) oscillator circuits. The clock oscillated by either of the circuits can be used as the system clock.

Figure 6-1 shows the configuration of the system clock control register.

Whether the main or subclock is used as the system clock is specified by the SYSCK flag (RF: address 02H, bit 1). By resetting the XEN flag (RF: address 02H, bit 0), oscillation of the main clock can be stopped to reduce current dissipation.

To use the subclock, be sure to connect a 0.1-μF capacitor to the VREG pin to stabilize the oscillation of the subclock. When the subclock is not used (which is specified by mask option), connect the XTIN pin to GND, and XTOUT pin to VREG pin.

Fig. 6-1 System Clock Control Register



Specifies mask option		Initial value at reset		Remark
Main clock	Subclock	SYSCK	XEN	
Used (USEX)	Used (USEXT)	1	1	Value can be changed <sup>Note</sup>
	Not used (NOXT)	1	1	Fixed value (cannot be changed in software)
Not used (NOX)	Used (USEXT)	0	0	

**Note** SYSCK cannot be changed to 1 and XEN cannot be changed to 0.

★

## 6.1 SWITCHING SYSTEM CLOCK

The system clock can be switched between the main clock and subclock by using the SYSCK flag (RF: address 02H, bit 1) as shown in Fig. 6-1.

### (1) Switching from main clock to subclock

The system clock can be changed from the main clock to subclock by resetting the SYSCK flag to “0”. When NOXT is set by mask option, however, the subclock cannot be selected (SYSCK and XEN cannot be reset to “0”).

**Caution** When turning on the power, make sure that a sufficient time elapses to stabilize the oscillation of the subclock (confirm that the IRQWTM flag (RF: address 3CH, bit 2) is set by the program at a specific cycle).

### (2) Switching from subclock to main clock

The system clock can be changed from the subclock to the main clock by setting the SYSCK flag to “1”. When NOX is set by mask option, however, the main clock cannot be selected (SYSCK and XEN cannot be set to “1”).

**Caution** Before setting the SYSCK flag, make sure that at least 10 ms elapses after the XEN flag has been set to “1” so that the oscillation stabilizes.

## 6.2 MAIN CLOCK OSCILLATION CONTROL FUNCTION

When the subclock is used as the system clock, oscillation of the main clock can be controlled by manipulating the XEN flag (RF: address 02H, bit 0).

If the system clock is changed from the subclock to main clock (by setting the SYSCK flag) after the main clock is started (by setting the XEN flag), make sure that an oscillation stabilization time of about 10 ms elapses.

**Caution** Do not manipulate the XEN and SYSCK flags simultaneously (execute the POKE instruction twice).

**7. 8-BIT TIMER AND REMOTE CONTROLLER CARRIER GENERATOR CIRCUIT**

The 8-bit timer is mainly used to generate the leader pulse of the remote controller signal, and to output codes. Operations of timers are controlled by the GET instruction, the PUT instruction, and registers on the register file.

**7.1 CONFIGURATION OF THE 8-BIT TIMER (WITH MODULO FUNCTION)**

Figure 7-1 shows the functional block diagram of the 8-bit timer.

The 8-bit timer consists of an 8-bit counter (TMC), an 8-bit modulo register (TMM), a comparator which compares the contents of the timer with the contents of the modulo register, and a selector which selects a count clock of the 8-bit timer.

Starting/stopping of the 8-bit timer and resetting of the 8-bit counter are controlled by TMEN (bit 3 of address 33H) and TMRES (bit 2 of address 33H) of the register file. The count clock of the 8-bit timer is selected by TMCK1 (bit 1 of address 33H) and TMCK0 (bit 0 of address 33H) of the register file.

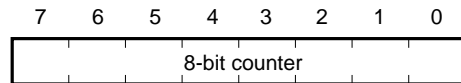
The contents of the 8-bit counter are read via the data buffer (DBF) by the GET instruction. The user cannot write any value in the 8-bit counter. The user can set a value in the modulo register by the PUT instruction via the data buffer (DBF).

The user cannot read the contents of the modulo register. As the 8-bit counter (TMC) and the modulo register (TMM) use an identical address, the CPU accesses the 8-bit counter to read and the 8-bit modulo register to write.

When the current count value of the counter and the value of the modulo register coincide with each other, the interrupt request flag (IRQTM: address 3EH, bit 0) is set, reflecting the output of the POD<sub>1</sub>/TMOU<sub>T</sub> pin.

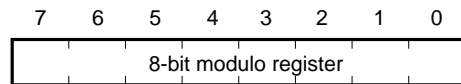
TMOU<sub>T</sub> is initialized and outputs a high level when TMRES is set.

TMC



Address	At reset	R/W
Peripheral register: 02H	00H	R

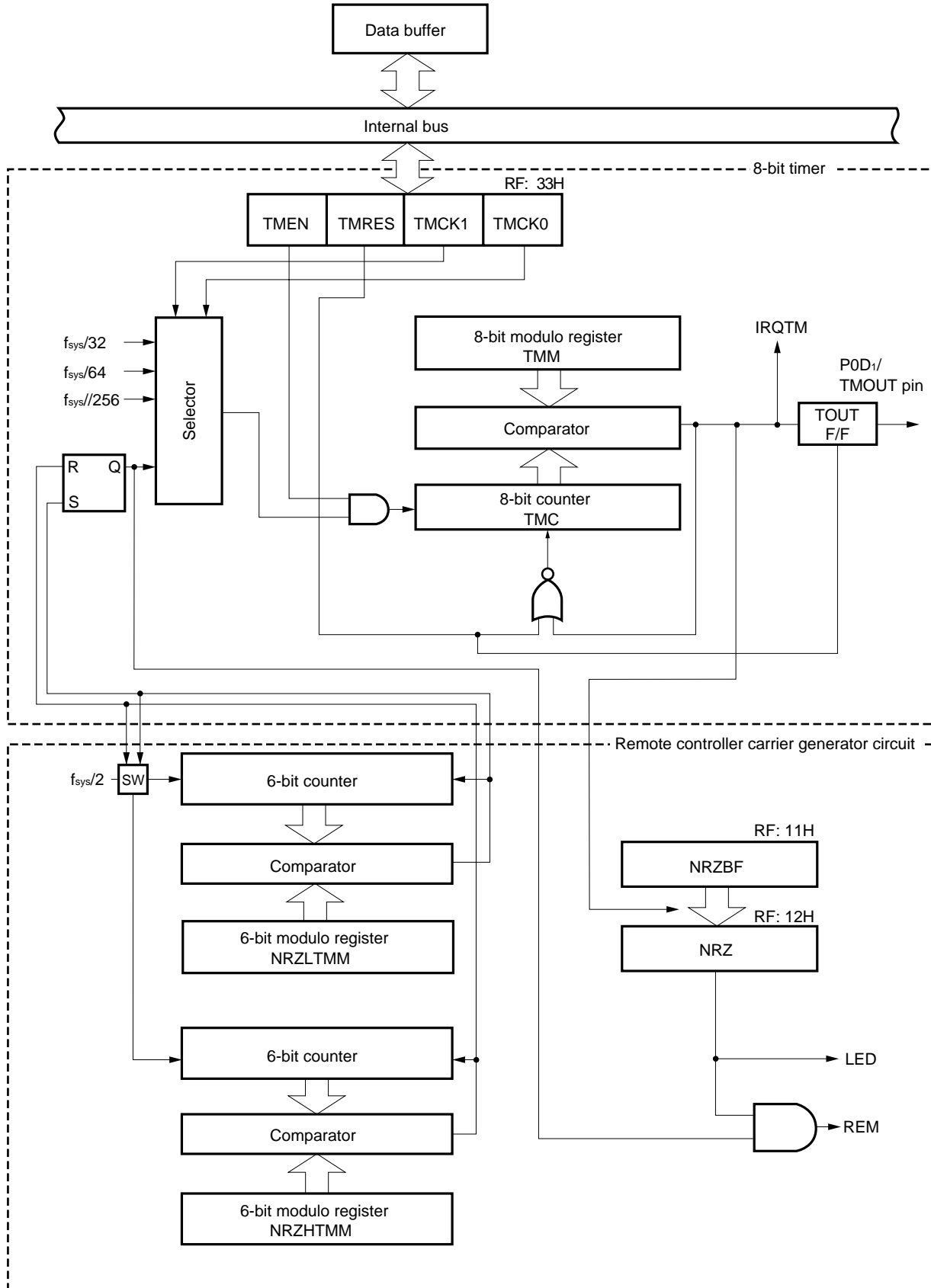
TMM



Address	At reset	R/W
Peripheral register: 02H	FFH	W

**Caution Do not clear TMM to 0 (IRQTM cannot be set.)**

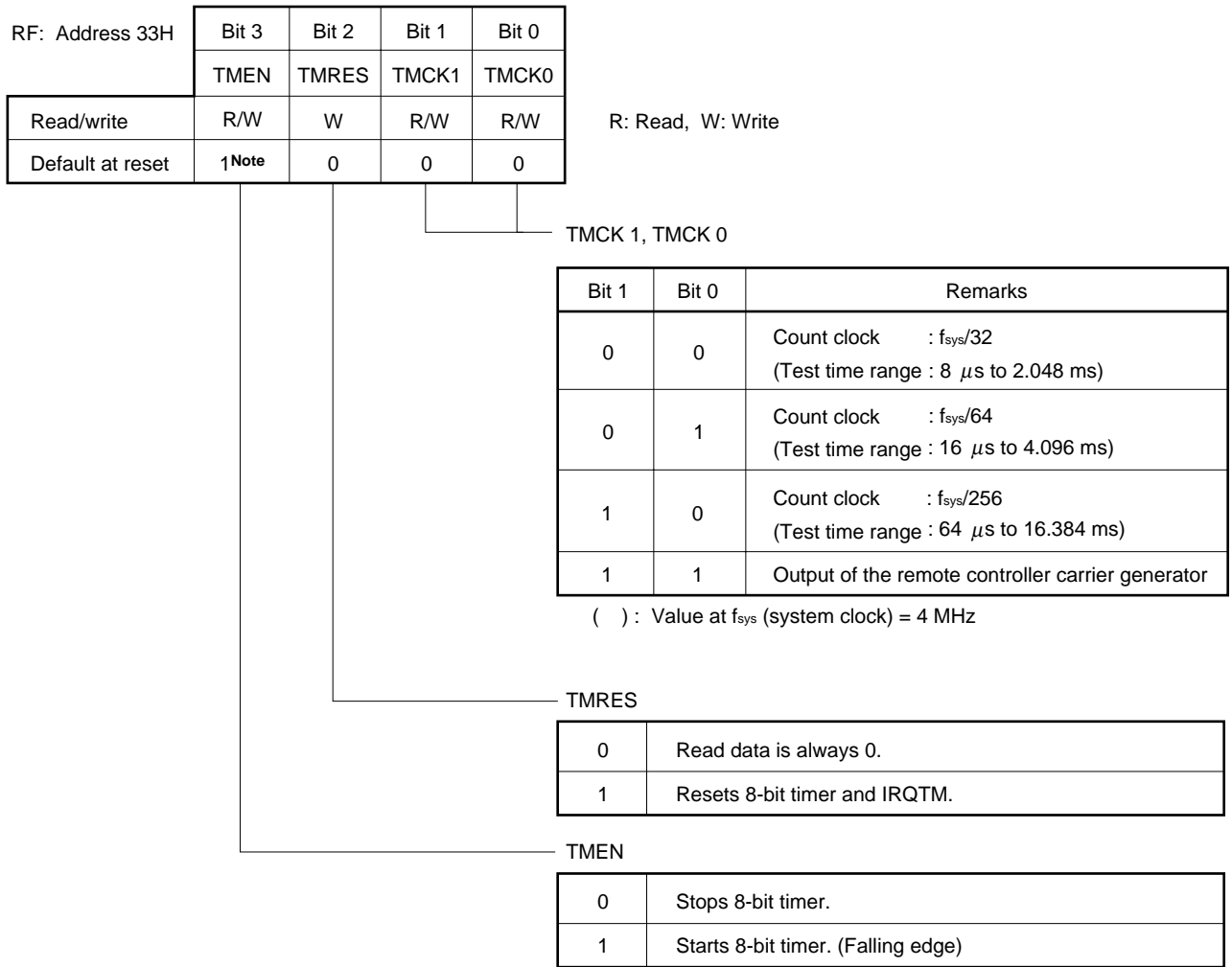
Fig. 7-1 Configuration of 8-Bit Timer and Remote Controller Carrier Generator Circuit



- Remarks 1.  $f_{sys}$  (system clock frequency):  $f_x$  or  $f_{xt}$
- 2. TMM, TMC, NRZLTMM and NRZHTMM are peripheral register.

7.2 FUNCTION OF THE 8-BIT TIMER (WITH MODULO FUNCTION)

Fig. 7-2 8-Bit Timer Control Register



**Note** This bit is always set to 1 when the STOP mode is released.

### 7.3 REMOTE CONTROLLER CARRIER GENERATOR

The μPD17207 is equipped with a circuit to generate carriers for the remote controller.

This circuit consists of a 6-bit counter, a modulo register (NRZHTMM) to determine an NRZ high-level period, a modulo register (NRZLTMM) to determine an NRZ low-level period, and a comparator.

A carrier duty factor and a carrier frequency are determined by the contents of these modulo registers. The values of the high- and low-level periods are set in the corresponding modulo registers via the data buffers (DBF).

A clock signal input to the 6-bit counter is obtained by dividing the frequency of the system clock signal by two (e.g. 2 MHz with a system clock of 4 MHz<sub>fx</sub> or 16.384 kHz with a system clock of  $f_{XT} = 32.768$  kHz).

Modulo registers NRZHTMM and NRZLTMM are respectively resident on peripheral addresses 04H and 03H. These registers can be written by the PUT instruction and read by the GET instruction

#### 7.3.1 Remote Controller Signal Output Control

The output of the REM pin which outputs carriers is controlled by NRZ (bit 0, address 12H of the register file), NRZBF (bit 0, address 11H of the register file), and an 8-bit timer.

While NRZ is "1", the REM pin outputs a carrier signal generated by the remote controller carrier generator. While NRZ is "0", the output of the REM pin is low. The contents of the NRZBF are automatically set in the NRZ flag by an interrupt signal generated by the 8-bit timer. When data is set in the NRZBF flag in advance, the status of the output of the REM pin varies in synchronization with the counting operation of the 8-bit timer.

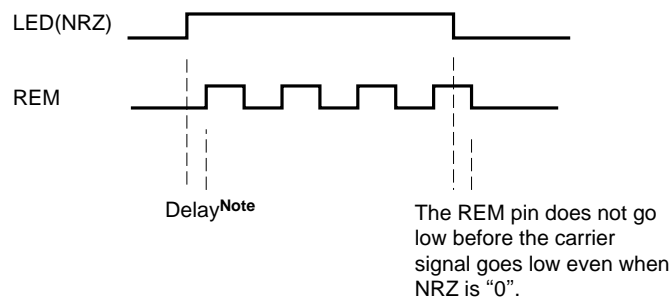
The content of the NRZ flag is output to the LED pin. Namely, the LED pin outputs a high-level signal when NRZ is "1" and a low-level signal when NRZ is "0".

If the 8-bit timer generates an interrupt signal when the output of the REM pin is high, that is, when NRZ is "1" and a carrier signal is high, the output of the REM pin does not match the contents of NRZ until the carrier signal goes low.

This operation is required to hold the pulse width of high carrier pulses constant. (See Fig. 7-3.)

When NRZ is "0", the carrier generation circuit stops. In a system using the output of the remote controller carrier generator as a clock signal for the 8-bit timer, clock pulses are continuously supplied even after NRZ has become "0".

Fig. 7-3 Remote Controller Carrier Output



**Note** This is the value when (TMCK1, TMCK0) ≠ (1,1).

The value when (TMCK1, TMCK0) = (1, 1) differs depending on the manipulation of NRZ.

If NRZ is set to 1 by an instruction, the width of the first high-level pulse may be narrowed. If NRZ is set by means of transfer from NRZBF, the delay in the above chart is equivalent to the low-level pulse width of the carrier clock.

Fig. 7-4 Register to Control Output Signals of the Remote Controller

RF: Address 12H	Bit 3	Bit 2	Bit 1	Bit 0	
	0	0	0	NRZ	
Read / write	R	R	R	R/W	R: Read, W: Write
Default at reset	0	0	0	0	

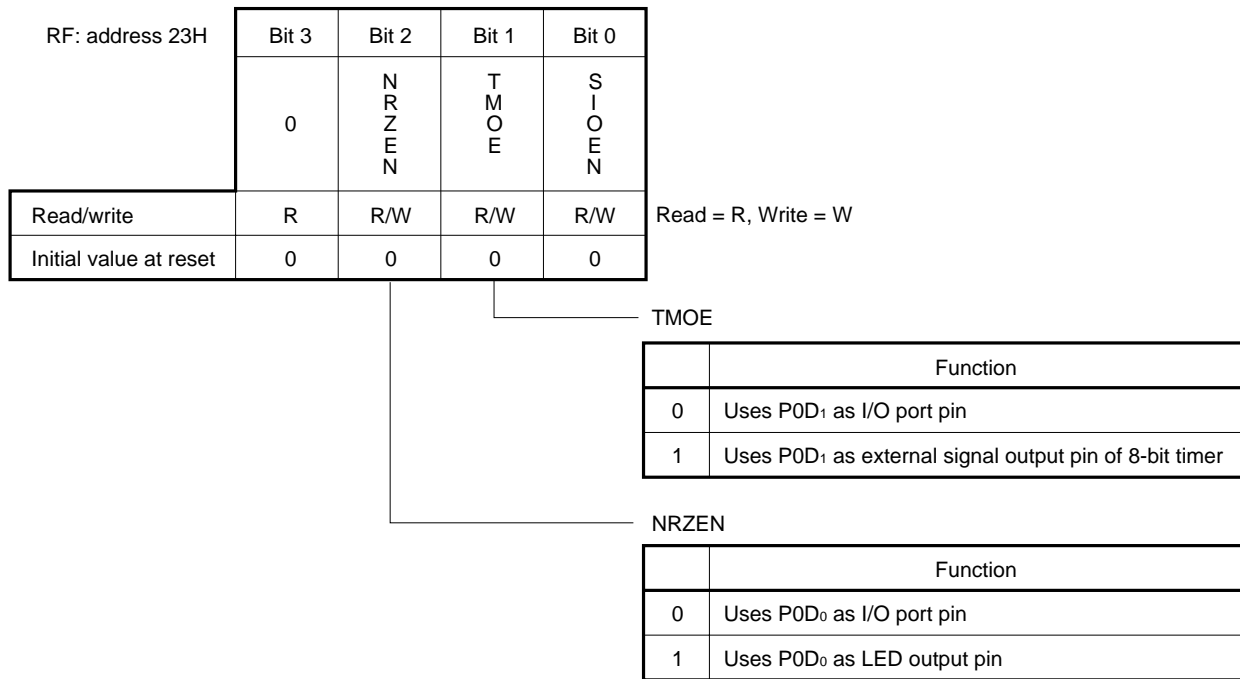
NRZ	
0	Outputs low-level signal to REM pin and low-level signal to LED pin.
1	Outputs carrier signal to REM pin and high-level signal to LED pin.

RF: Address 11H	Bit 3	Bit 2	Bit 1	Bit 0	
	0	0	0	NRZBF	
Read / write	R	R	R	R/W	R: Read, W: Write
Default at reset	0	0	0	0	

NRZBF	
0	NRZ buffer bit. Contents of this bit are transferred to NRZ by interrupt signal generated by 8-bit timer.
1	

Fig. 7-5 Input/Output Control Register for Port/Timer Output, LED Output, and Serial Interface



**7.3.2 Setting a Carrier Frequency and a Duty Factor**

Frequency division ratio  $\ell$  necessary for obtaining carrier frequency  $f_c$  can be calculated by the following expression where the frequency of main clock (X) is  $f_x$ .

$$\ell = f_x / (2 \times f_c).$$

Set the following values to the modulo register to divide  $\ell$  into duty factors m: n.

$$\text{High-level period setting value (NRZHTMM)} = \ell \times m / (m + n) - 1$$

$$\text{Low-level period setting value (NRZLTMM)} = \ell \times n / (m + n) - 1$$

**Example:** Where  $f_x = 4 \text{ MHz}$ ,  $f_c = 38 \text{ kHz}$ , and duty factor = 1/3 (m:n = 1:2)

$$\ell = 4 \text{ MHz} / (2 \times 38 \text{ kHz}) \doteq 52.6$$

Therefore, the values of the modulo registers are as follows:

$$\text{High-level period (NRZHTMM)} \doteq 17 \text{ (11H)}$$

$$\text{Low-level period (NRZLTMM)} \doteq 34 \text{ (22H)}$$

Calculating the carrier frequency with these values,

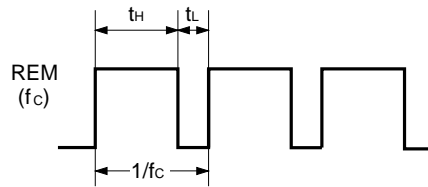
$$f_c = f_x / (2 \times \ell) = 4 \text{ MHz} / (2 \times 53) \doteq 37.74 \text{ kHz}$$

$$\text{(where } \ell = (17 + 1) + (34 + 1) = 53)$$



Table 7-1 Example of Carrier Frequency ( $f_x = f_{sys} = 4 \text{ MHz}$ )

Set Value		$t_H (\mu s)$	$t_L (\mu s)$	$1/f_c (\mu s)$	$f_c (\text{kHz})$	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



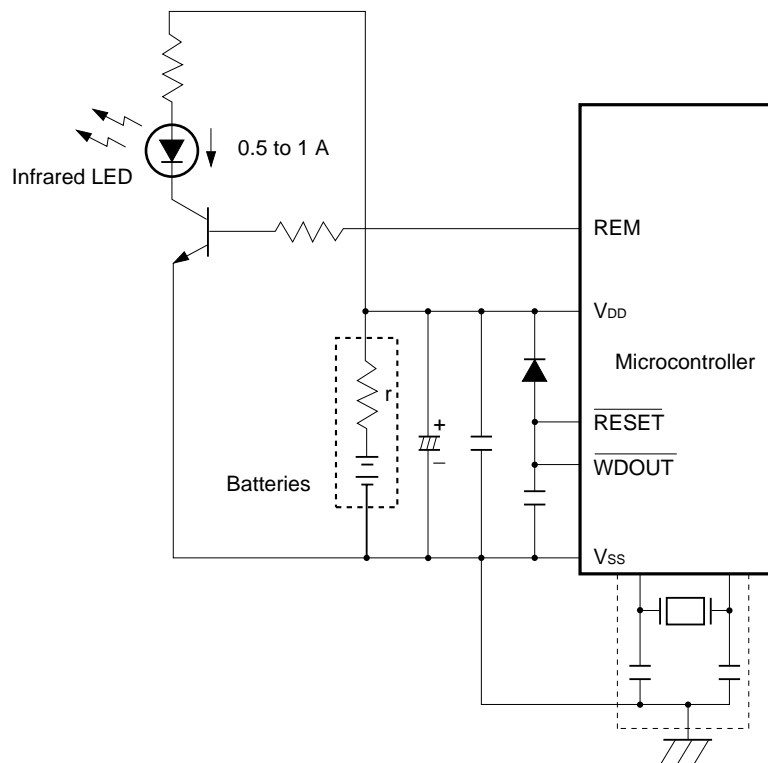
### 7.3.3 Countermeasures against Noise during Transmission (Carrier Output)

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  $\Omega$  of equivalent resistance ( $r$ ) exists in the power source as shown in Fig. 7-6. This resistance increases from 10 to 20  $\Omega$  if the supply voltage drops to 2 V. While the carrier is 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 oscillator as close as possible to the microcontroller and shield it with GND lines (as indicated by the portion inside the dotted line 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 an interrupt that requires read/write processing and stack, such as key scan interrupt, and the CALL/RET instruction, while the carrier is output.
- ⑤ To improve the reliability in case of program hang-up, use the watchdog timer (connect the  $\overline{\text{WDOUT}}$  and  $\overline{\text{RESET}}$  pins).

Fig. 7-6 Example of Countermeasures against Noise



- Remarks 1.** The INT and  $\overline{\text{RESET}}$  pins are multiplexed with test pins (refer to 3.4 NOTES ON USING  $\overline{\text{RESET}}$  AND INT PINS).
- 2.** In this figure, the  $\overline{\text{RESET}}$  pin is connected to a pull-up resistor by mask option.

## 8. WATCH TIMER/WATCHDOG TIMER

The watch timer is used to generate a watch interrupt signal and a signal to reset the watchdog timer.

### 8.1 CONFIGURATION OF WATCH TIMER/WATCHDOG TIMER

Figure 8-1 shows the functional block diagram of the watch timer/watchdog timer.

As shown in Fig. 8-1, the watch timer consists of two selectors, A and B, and a frequency divider. Selector A selects the divided output ( $f_x/2^7$ ) of the 32.768-kHz subclock oscillator (XT) output or of the main clock oscillator (X) output as the source clock by using mask option. Selector B selects a frequency to be used as an interrupt signal. The divider creates a frequency of the source clock.

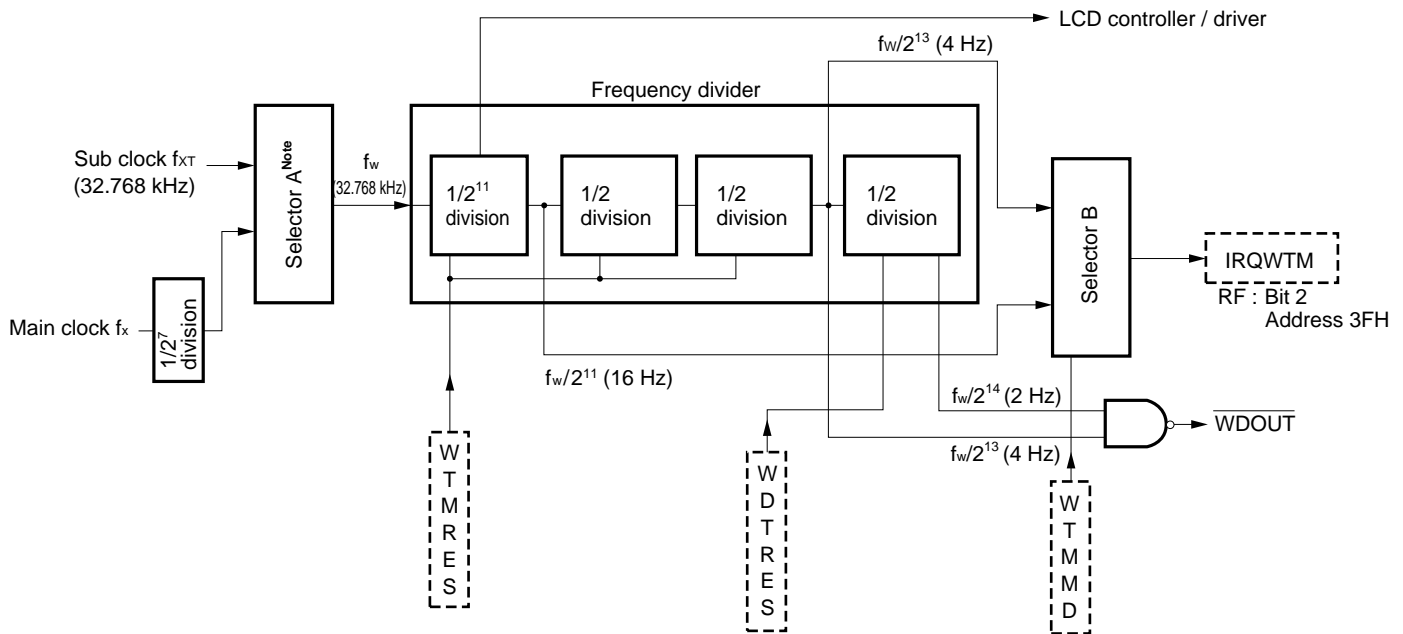
Resetting the watch timer and the operation of selector B is controlled by WTMRES (address 03H, bit 1) and WTMMD (address 03H, bit 2) of the register file.

The watchdog timer is reset by WDTRES (address 03H, bit 3) of the register file.

If the subclock ( $f_{XT}$ ) is the source clock, the watch timer count cannot be stopped. Therefore, the subclock does not stop but continues to oscillate even when the CPU is in the STOP mode.

If the divided output of the main clock ( $f_x/2^7$ ) is the source clock (when the subclock is not used), the watch timer stops when the CPU is set in the STOP mode.

Fig. 8-1 Configuration of Watch Timer/Watchdog Timer



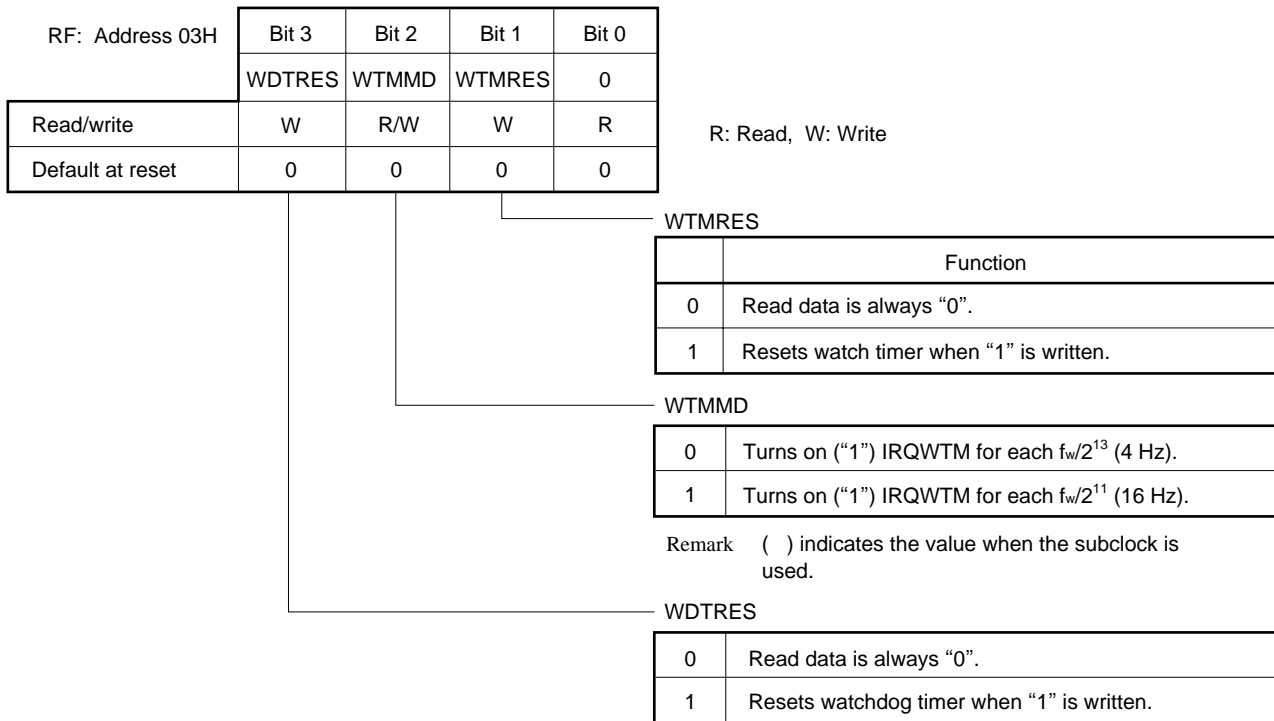
**Remark** ( ) indicates the value when the subclock is used.

**Note** The source clock of the watch timer/watchdog timer is fixed as follows by mask option:

- ① When subclock is selected by mask option  
The source clock is fixed to the subclock.
- ② When subclock is not selected by mask option  
The source clock is fixed to  $f_x/2^7$

8.2 FUNCTION OF WATCH TIMER/WATCHDOG TIMER

Fig. 8-2 Watch Timer/Watchdog Timer Control Register



**8.3 WATCHDOG TIMER OPERATION TIMING**

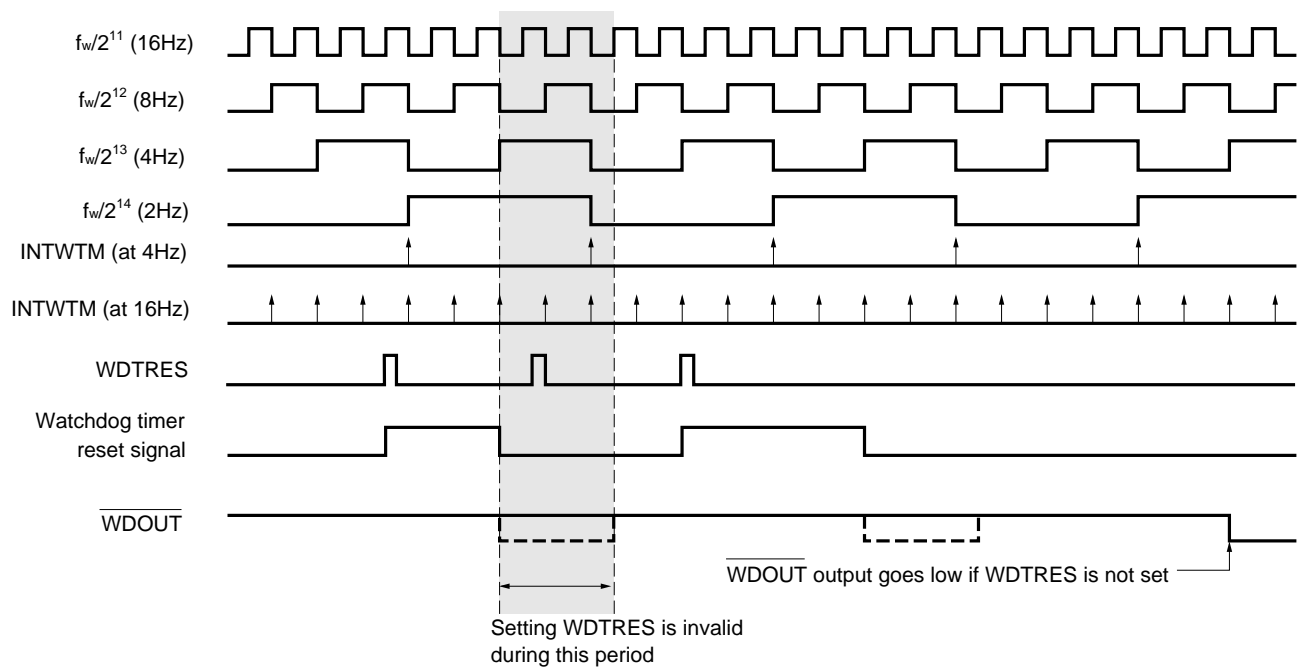
Unless the watchdog timer is reset in a fixed time, the  $\overline{\text{WDOUT}}$  pin outputs a low level. By connecting the  $\overline{\text{WDOUT}}$  pin to the  $\overline{\text{RESET}}$  pin, a program hang-up can be detected by the watchdog timer.

To reset the watchdog timer, set WDTRES (WDTRES = 1).

To disable hang-up detection by the watchdog timer when the subclock is used, program so that WDTRES is set at intervals of approximately 340 ms or less.

- Cautions**
1. The watchdog timer cannot be reset in the shaded range in Fig. 8-3. Therefore, set the watchdog timer before both the  $f_w/2^{13}$  and  $f_w/2^{14}$  signals go high.
  2. For further information on the  $\overline{\text{WDOUT}}$  pin, also refer to 14. RESET.

**Fig. 8-3 Watchdog Timer Operation Timing**



**Remark** Figures in the parentheses indicate the value when using the subclock.

★ 9. A/D CONVERTER

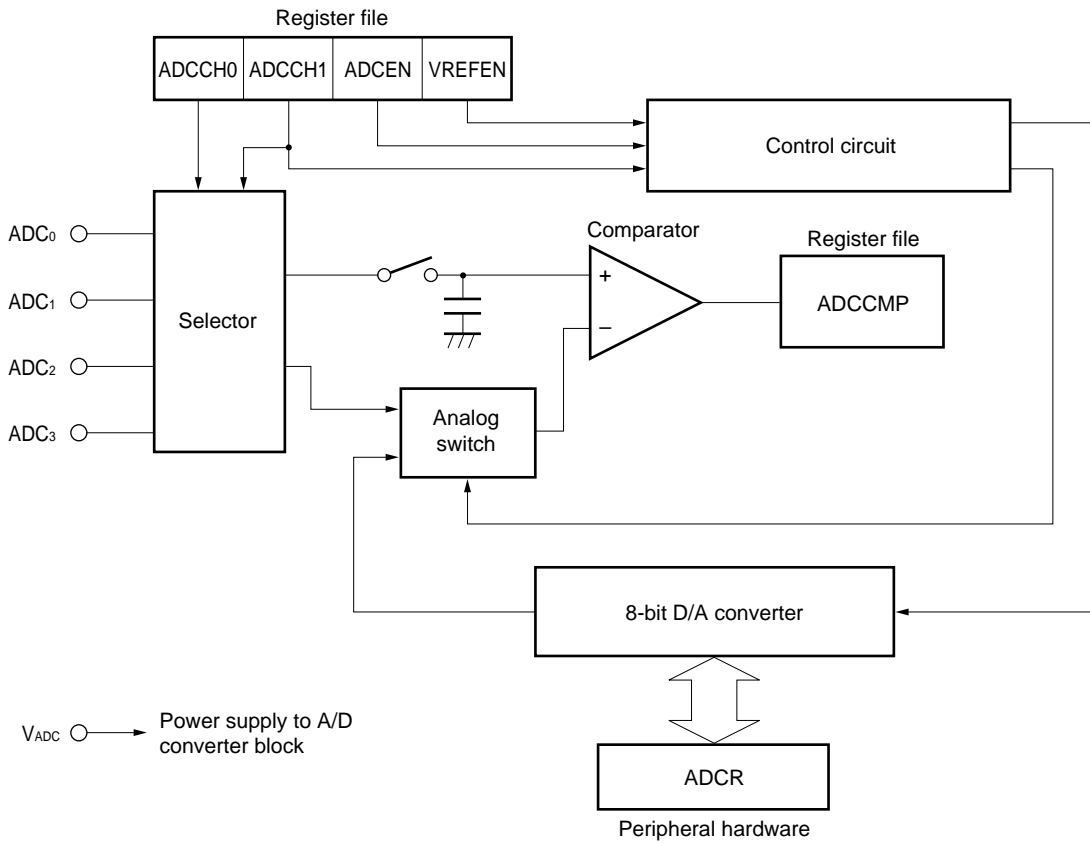
The μPD17207 has an 8-bit successive approximation A/D converter. This A/D converter can be used in the following two modes.

Mode	Description
A/D conversion mode	Converts analog voltage input to one pin into digital signal
Compare mode	Compares analog voltages input to two pins

9.1 CONFIGURATION OF A/D CONVERTER

The 8-bit A/D converter consists of a selector that selects an input pin, analog switch, control circuit, 8-bit resistor string D/A converter, and comparator.

Fig. 9-1 Block Diagram of A/D Converter



**9.2 FUNCTION OF A/D CONVERTER**

**9.2.1 Function in A/D Conversion Mode**

In the A/D conversion mode, the A/D converter compares an analog voltage input to one of the pins ADC3 through AD0 with an internal reference voltage and outputs the result to ADCCMP on the register file.

The pin from which an analog voltage is to be input is selected by the operation mode register (refer to **Figure 9-2**). Two or more pins cannot be selected for A/D conversion at the same time.

The internal reference voltage is created by the 8-bit D/A converter based on the data set by the internal reference voltage setting register (ADCR). The 8-bit D/A converter can create 256 values of the internal reference voltage.

By selecting an internal reference voltage and executing successive approximation in software, the input analog voltage can be converted into a digital value.

Operation Mode Register				Selected Input Pin	Function
VREFEN	ADCEN	ADCCH1	ADCCH0		
1	1	0	0	ADC <sub>0</sub>	Compares analog voltage input to ADC <sub>0</sub> pin with internal reference voltage
		0	1	ADC <sub>1</sub>	Compares analog voltage input to ADC <sub>1</sub> pin with internal reference voltage
		1	0	ADC <sub>2</sub>	Compares analog voltage input to ADC <sub>2</sub> pin with internal reference voltage
		1	1	ADC <sub>3</sub>	Compares analog voltage input to ADC <sub>3</sub> pin with internal reference voltage

**9.2.2 Function in Compare Mode**

In the compare mode, analog voltages input two of the pins ADC3 through ADC0 are compared with each other and the result is output to ADCCMP in the register file. The two pins from which analog voltages are to be input are selected by the operation mode register (refer to **Figure 9-2**).

Two pairs of pins can be selected: pins ADC2 and ADC0, pins or ADC3 and ADC1. Both pairs of pins cannot be selected at the same time.

Operation Mode Register				Selected Input Pin	Function
VREFEN	ADCEN	ADCCH1	ADCCH0		
0	0	1	0	ADC <sub>2</sub> ADC <sub>0</sub>	Compares analog voltages input to ADC <sub>2</sub> and ADC <sub>0</sub> pins
		1	1	ADC <sub>3</sub> ADC <sub>1</sub>	Compares analog voltages input to ADC <sub>3</sub> and ADC <sub>1</sub> pins

9.3 CONTROL REGISTERS OF A/D CONVERTER

9.3.1 Operation Mode Register

The operation mode register selects the operation mode and analog input pin(s) of the A/D converter by using the flags in the register file as illustrated below.

Fig. 9-2 Operation Mode Register

RF: address 21H	Bit 3	Bit 2	Bit 1	Bit 0		
	V R E F E N	A D C E N	A D C C H 1	A D C C H 0		
Read/write	R/W				Read = R, write = W	
Initial value at reset	0	0	0	0		

VREFEN	ADCEN	ADCCH1	ADCCH0	Operation mode	Selected analog input pin	
					+side of internal comparator	—side of internal comparator
0	0	1	0	Compare	ADC <sub>2</sub> pin	ADC <sub>0</sub> pin
0	0	1	1	Compare	ADC <sub>3</sub> pin	ADC <sub>1</sub> pin
1	1	0	0	A/D conversion	ADC <sub>0</sub> pin	(Internal reference voltage)
1	1	0	1	A/D conversion	ADC <sub>1</sub> pin	(Internal reference voltage)
1	1	1	0	A/D conversion	ADC <sub>2</sub> pin	(Internal reference voltage)
1	1	1	1	A/D conversion	ADC <sub>3</sub> pin	(Internal reference voltage)
0	0	0	Don Care	Operation stops	None	
Others				Setting prohibited	Undefined	

**Caution** Set the operation stop mode to reduce the current consumption when the A/D converter is not used.



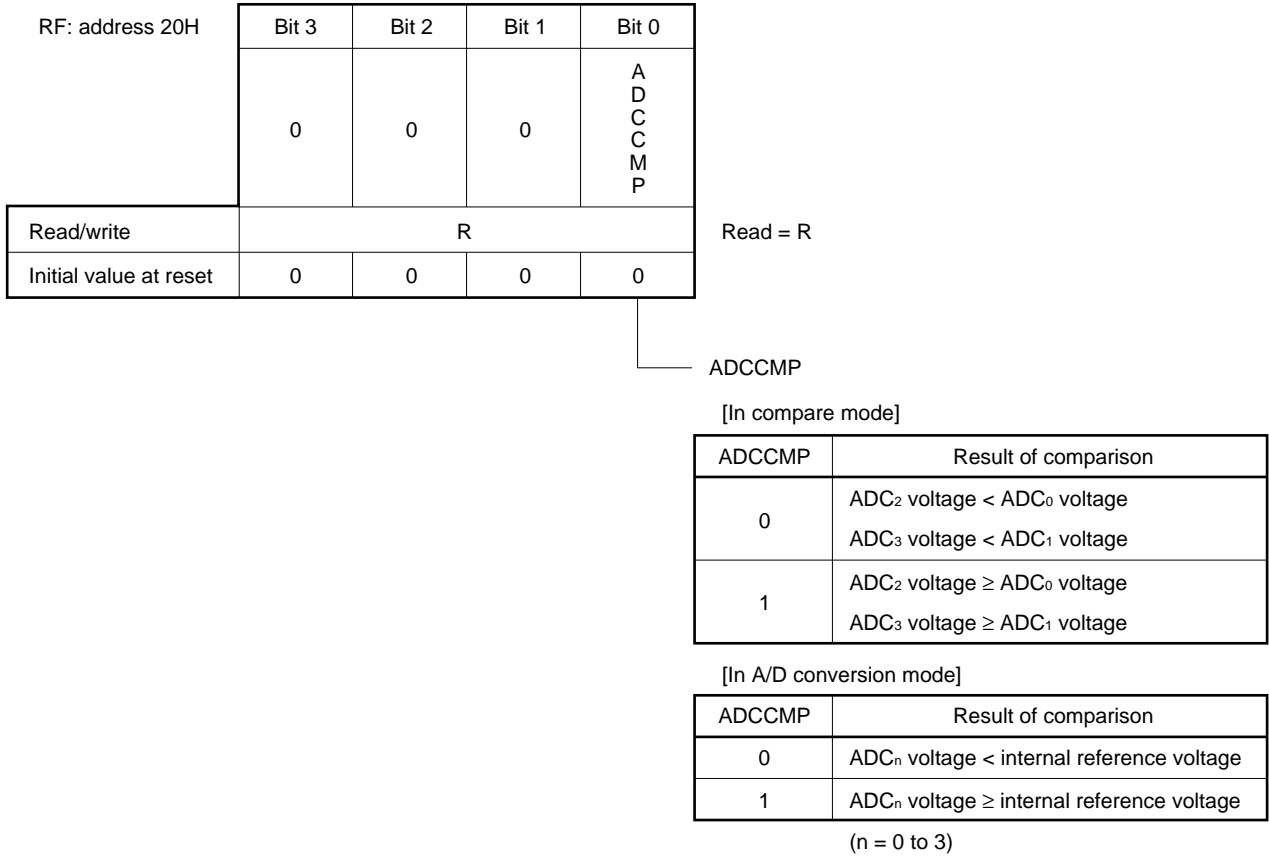
**9.3.2 Internal Reference Voltage Setting Register (ADCR)**

The internal reference voltage setting register (ADCR) is an 8-bit register that sets the reference voltage of the converter. This register is allocated to the peripheral hardware. Data is written to the ADCR via data buffer (DBF). The 8-bit data set to DBF0 and DBF1 is written to the ADCR by using the “PUT ADCR, DBF” instruction.

**9.3.3 Compare Result Register**

The result of comparison by the converter is stored to the ADCCMP flag in the register file.

**Fig. 9-3 Compare Result Register**



9.4 OPERATION IN A/D CONVERSION MODE

The timing necessary for A/D conversion differs depending on whether the main clock or subclock is selected as the system clock.

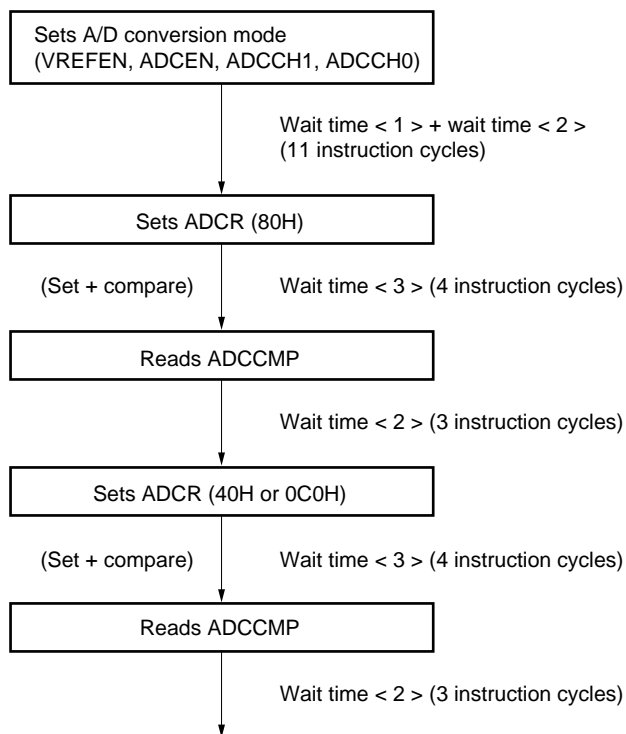
(1) When main clock is selected as system clock

The following wait times must be set in software in the A/D conversion mode.

Wait time <1>: Time of transition from operation stop mode to A/D conversion mode (8 instruction cycles)

Wait time <2>: Wait time until value can be set to ADCR (3 instruction cycles)

Wait time <3>: Wait time until compare result register can be read (4 instruction cycles)



An example of a program for A/D conversion when the main clock is selected is shown below.

```

CMPVAL DAT      80H                ; Reference voltage =VADC × CMPVAL/256
ADCNV:
    BANK0
    INITFLG VREFEN,ADCEN,NOT ADCCH1,NOT ADCCH0 ; Starts sampling of input to ADC0
    MOV     DBF0, #CMPVAL AND 0FH
    MOV     DBF1, #CMPVAL SHR 4 AND 0FH
    REPT    9
    NOP
    ENDR
    PUT     ADCR,DBF                ; Holds input and starts comparison
    NOP
    NOP
    NOP
    NOP
    PEEK    WR,.MF.ADCCMP SHR 4 AND 0FFFH ; Reads result of comparison (starts
                                           sampling)
    MOV     DBF0, #CMPVAL AND 0FH
    MOV     DBF1, #CMPVAL SHR 4 AND 0FH
    NOP
    PUT     ADCR, DBF               ; Holds input and starts comparison
    NOP
    NOP
    NOP
    NOP
    PEEK    WR,.MF.ADCCMP SHR 4 AND 0FFFH ; Reads result of comparison (starts
                                           sampling)

```

**(2) When subclock is being selected as system clock**

Unlike when the main clock is selected, the wait times do not need to be set in software when the subclock is selected. Setting the A/D conversion mode, ADCR, and reading ADCCMP are completed in one instruction cycle, respectively.

An example of a program for A/D conversion when the subclock is selected is shown below.

```

CMPVAL DAT      80H                ; Reference voltage = VADC × CMPVAL/256
ADCNV:
    BANK0
    INITFLG VREFEN,ADCEN,NOT ADCCH1,NOT ADCCH0 ; Starts sampling a voltage input to the ADC0
                                           pin.
    MOV     DBF0,#CMPVAL AND 0FH
    MOV     DBF1,#CMPVAL SHR 4 AND 0FH
    PUT     ADCR,DBF                ; Holds the Input and starts comparison
    PEEK    WR,.MF.ADCCMP SHR 4 AND 0FFFH ; Reads the result of comparison (and starts
                                           sampling).

```

**9.5 OPERATION IN COMPARE MODE**

In the compare mode, the result of comparison stored to ADCCMP is read and then the next comparison is immediately performed. Therefore, comparison is executed successively and the ADCCMP flag is rewritten accordingly.

The timing necessary for compare mode differs depending on whether the main clock or subclock is selected as the system clock.

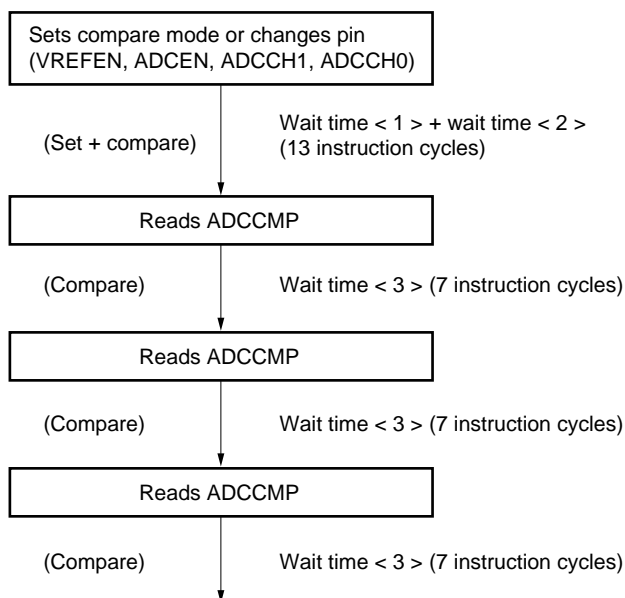
**(1) When main clock is selected as system clock**

The following wait times must be set in software in the compare mode.

Wait time <1>: Time of transition from operation stop mode to compare mode (10 instruction cycles)

Wait time <2>: Wait time until compare result register can be read (first time only) (3 instruction cycles)

Wait time <3>: Wait time until compare result register can be read (second time and onward) (7 instruction cycles)



An example of a program in the compare mode when the main clock is selected is shown below.

COMPARE:

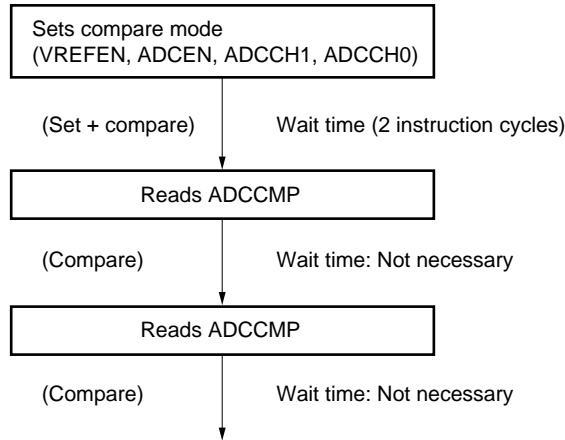
```

INITFLG  NOT VREFEN,NOT ADCEN,ADCCH1,ADCCH0 ; Starts comparing voltages of ADC3
                                                and ADC1
REPT     13
NOP
ENDR    } Waits for duration of 13 instruction cycles
                                                } or more until comparison ends
PEEK    WR,.MF.ADCCMP SHR 4 AND 0FFFH ; Reads result of comparison
REPT     7
NOP
ENDR    } Waits for duration of 7 instruction cycles
                                                } or more until comparison ends
PEEK    WR,.MF.ADCCMP SHR 4 AND 0FFFH ; Reads result of comparison
  
```

**(2) When subclock is selected as system clock**

The following wait times must be set in software during compare operation.

Wait time: Time of transition from operation stop mode to compare mode (2 instruction cycles)



An example of a program in the compare mode when the subclock is selected is shown below.

COMPARE:

```

INITFLG  NOT VREFEN,NOT ADCEN,ADCCH1,ADCCH0 ; Starts comparing voltages on ADC3
                                                and ADC1
NOP                                             } Waits for duration of 2 instruction cycles
NOP                                             } or more until comparison ends
PEEK     WR,.MF.ADCCMP SHR 4 AND 0FFFH        ; Reads result of comparison
PEEK     WR,.MF.ADCCMP SHR 4 AND 0FFFH        ; Reads result of comparison
  
```

## 10. SERIAL INTERFACE

Serial interface consists of an 8-bit shift register, a 4-bit shift mode register, and a 3-bit counter, and transmits data in series to and from the bus.

### 10.1 SERIAL INTERFACE FUNCTION

#### 10.1.1 8-bit Data Transfer in Synchronization with Clocks (Simultaneous transmission and reception)

Input and output of serial data on serial interface is controlled by the serial clock ( $\overline{\text{SCK}}$ ) signal. At the falling edge of the SCK signal, the most significant bit of the shift register is output from the SO pin (pin 59; also used as P1A<sub>1</sub>). At the rising edge of the SCK signal, the contents of the shift register are shifted left by one bit and, at the same time, data input via the SI pin (pin 60; also used as P1A<sub>2</sub>) is set in the least significant bit of the shift register.

The 3-bit counter counts serial clock pulses. Each time the counter counts eight clock pulses (each time serial data of 8 bits is transferred), the IRQSIO flag (bit 3, address 3BH) of the register file is turned on ("1") to make an interrupt request.

#### 10.1.2 8-bit Data Reception in Synchronism with Clocks (High-impedance SO output)

This operation is basically the same as the above operation except that the SI pin (pin 59; also used as P1A<sub>1</sub>) goes into a high-impedance state and does not output serial data. Therefore, the SO pin can be used as a port (P1A<sub>1</sub>).

### 10.2 SERIAL INTERFACE OPERATION

#### 10.2.1 Serial Interface Operation Modes

P1A<sub>2</sub>/SI (pin 60), P1A<sub>1</sub>/SO (pin 59), and P1A<sub>0</sub>/ $\overline{\text{SCK}}$  (pin 58) are placed in Serial Interface mode when the SIOEN flag (bit 0, address 23H) of the register file is turned on ("1"). These pins can be used as port pins when the SIOEN flag is off ("0"). As this operation mode disables transfer of serial data, the shift register can be used as an 8-bit register.

#### 10.2.2 Serial Operation Mode

The serial operation mode is determined by the status of the SIOHIZ flag (bit 2, address 22H) of the register file. When this flag is off ("0"), a clock-synchronous 8-bit transmission/reception mode is set. When this flag is on ("1"), a clock-synchronous 8-bit reception mode is set. Figure 10-1 shows shift timing waveforms. The only difference between these two modes is whether the SO pin (pin 59; also used as P1A<sub>1</sub>) goes into a high-impedance state.

In transmission of serial data, data to be transmitted is set in the shift register SIOSFR (peripheral address 01H) via the data buffer (DBF) by an PUT instruction, and the SIOTS flag (bit3, address 22H) of the register file is turned on ("1"). Thus serial data transfer starts. When 8 bits of data are transferred, the SIOTS flag is automatically turned off ("0") and the IRQSIO flag (bit 3, address 3BH) of the register file is turned on ("1") to generate an interrupt. If generation of an interrupt is disabled, the end of transfer can be indicated by the SIOTS and IRQSIO flags.

Reception of serial data is basically the same as the transmission of serial data except that data is output from the SO pin.

The  $\mu$ PD17207 supports four kinds of clock signals (three internal clocks and one external clock) to be selected as the serial clock source. These clock signals are selected by SIOCK1 (bit 1, address 22H) and SIOCK0 (bit 0, address 22H) of the register file.

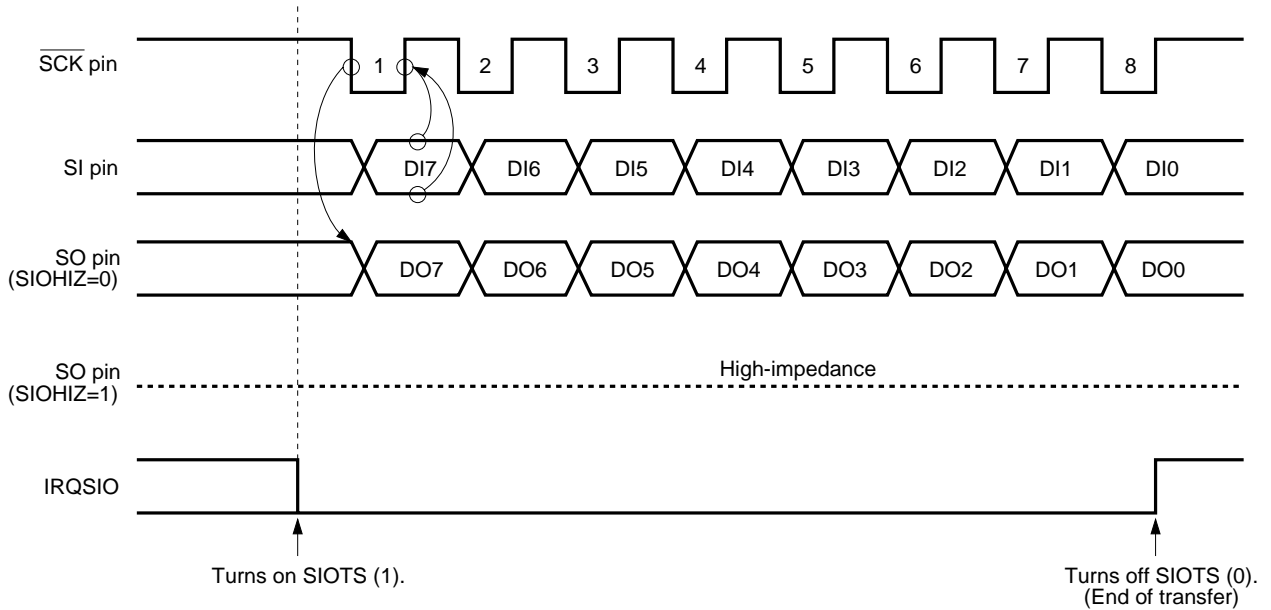
If one of the three internal clock signals is selected as the serial clock source, it is supplied to serial interface when the SIOTS flag turns on ("1"). The clock controls input/output of serial data and is output from the  $\overline{\text{SCK}}$  pin (pin 58; also used as P1A<sub>0</sub>). When eight clock pulses are supplied to the serial interface, the SIOTS flag is automatically turned off ("0") and the supply of clock pulses to the serial interface is stopped. Then, the  $\overline{\text{SCK}}$  pin is held high. At this time, the IRQSIO flag (bit 3, address 3BH) of the register file is turned on ("1").

If the external clock is selected, the clock pulses supplied from the  $\overline{\text{SCK}}$  pin are supplied to serial interface when the SIOTS flag is turned on ("1"). Similarly, when eight clock pulses are supplied to the serial interface, the SIOTS flag is automatically turned off ("0") and the supply of clock pulses to the serial interface is stopped. At this time, the IRQSIO flag (bit 3, address 3BH) of the register file is turned on ("1").

The IRQSIO flag is automatically reset to "0" when the SIOTS flag is turned on ("1").

To forcibly stop transfer of serial data, turn on the SIOTS flag manually. Note, however, that data transfer cannot be resumed from the point at which the transfer has been forcibly stopped.

Fig. 10-1 Shift Timing Waveforms



**Remark** DI : Serial data input  
 DO: Serial data output

Fig. 10-2 Input/Output Control Register for Selection of Port, Timer Output, LED Output, and Serial Interface

RF: Address 23H	Bit 3	Bit 2	Bit 1	Bit 0
	0	NRZEN	TMOE	SIOEN
Read/write	R	R/W	R/W	R/W
Default at reset	0	0	0	0

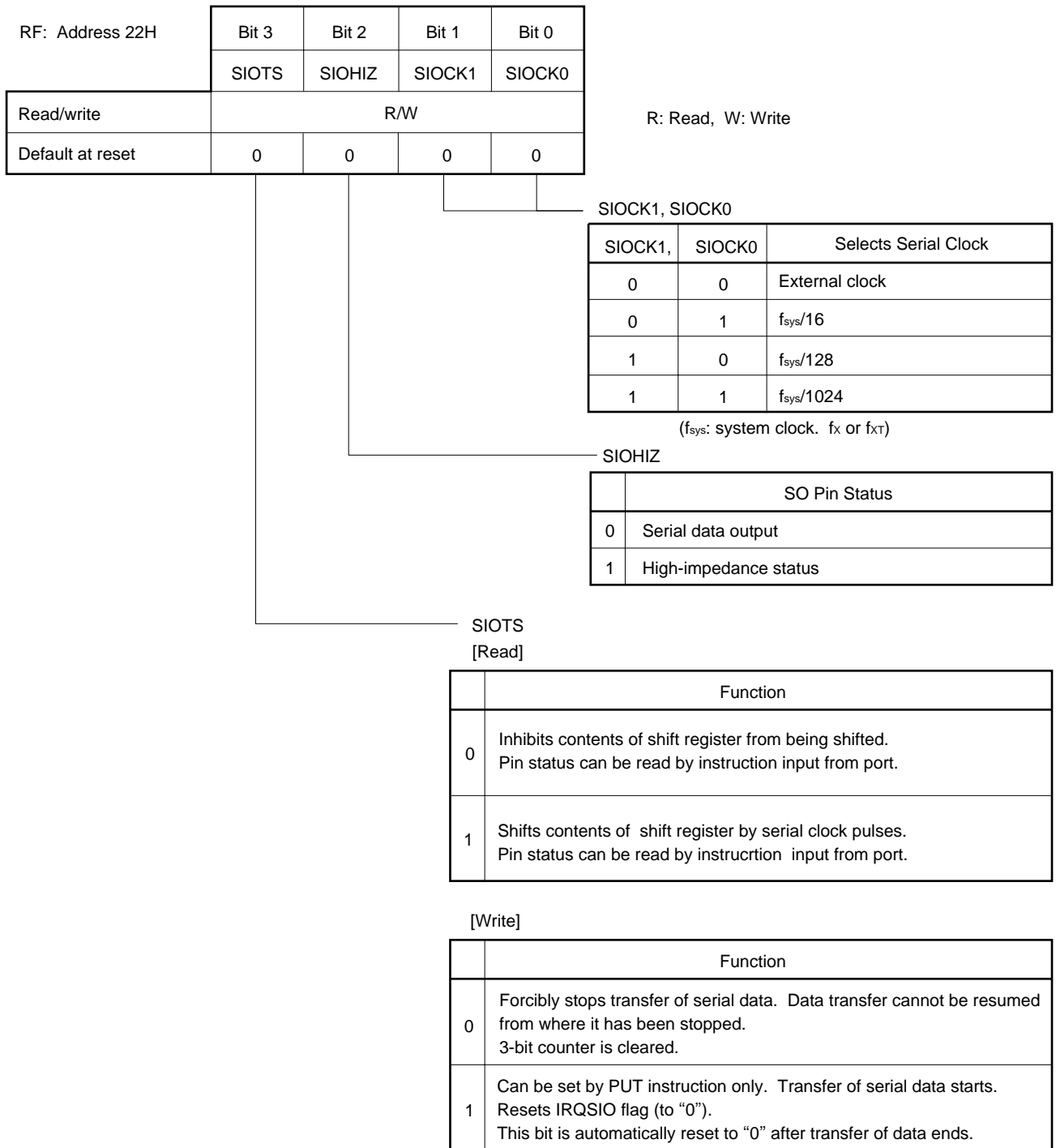
R: Read, W: Write

SIOEN

	Function
0	Uses Port 1A as I / O port.
1	Uses Port 1A as serial interface.



Fig. 10-3 Serial Interface Control Register



**Caution** Be sure to select a serial clock signal before starting transfer of serial data. Never set them at the same time.

**Remark** At the end of transfer of 8-bit serial data, the IRQSIO flag (bit 3, address 3BH of the register file) is turned on ("1") and an interrupt request occurs.

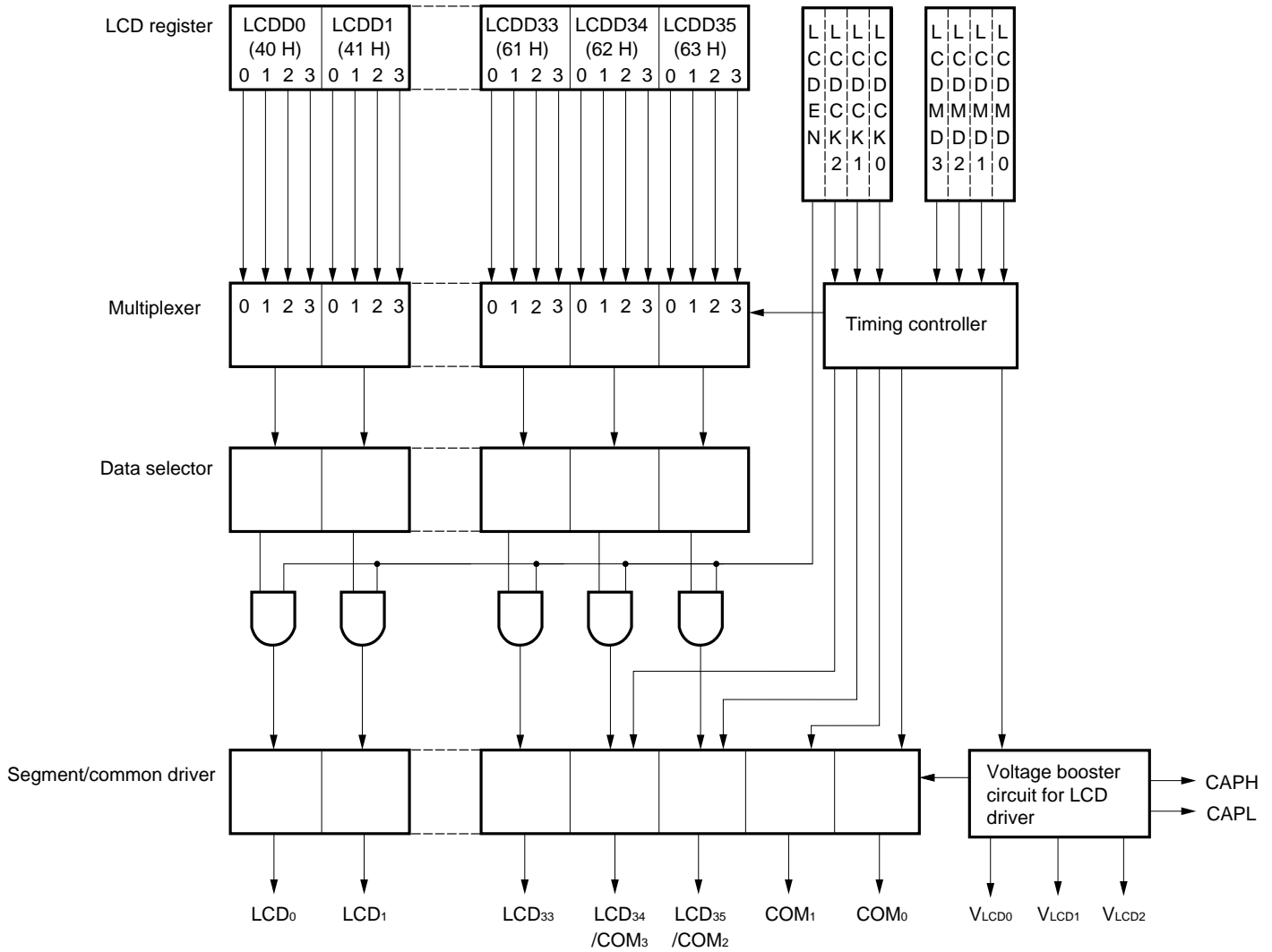
11. LCD CONTROLLER/DRIVER

11.1 CONFIGURATION OF LCD CONTROLLER/DRIVER

The μPD17207 is equipped with an LCD controller which generates segment and common signals according to the data set to the LCD register and a segment/common driver which can directly drive the LCD panel.

Figure 11-1 shows the functional block diagram of the LCD controller/driver.

Fig. 11-1 Block Diagram of LCD Controller/Driver



**11.2 FUNCTIONS OF LCD CONTROLLER/DRIVER**

The LCD controller/driver of the μPD17207 features the following:

- (1) Automatically reads the LCD register and generates segment signals and common signals.
- (2) Three display modes available:
  - Display mode 1: 1/2-duty, 1/3-bias
  - Display mode 2: 1/3-duty, 1/3-bias
  - Display mode 3: 1/4-duty, 1/3-bias
- (3) Four frame frequencies available in each display mode.
- (4) Since a voltage booster circuit for LCD driver is used, constant output voltage not affected by the fluctuation in the supply voltage.
- (5) The LCD register which is not used for display can be used as ordinary data memory.

Table 11-1 shows the maximum number of pixels available in each display mode.

**Table 11-1 Maximum Number of Pixels Displayed**

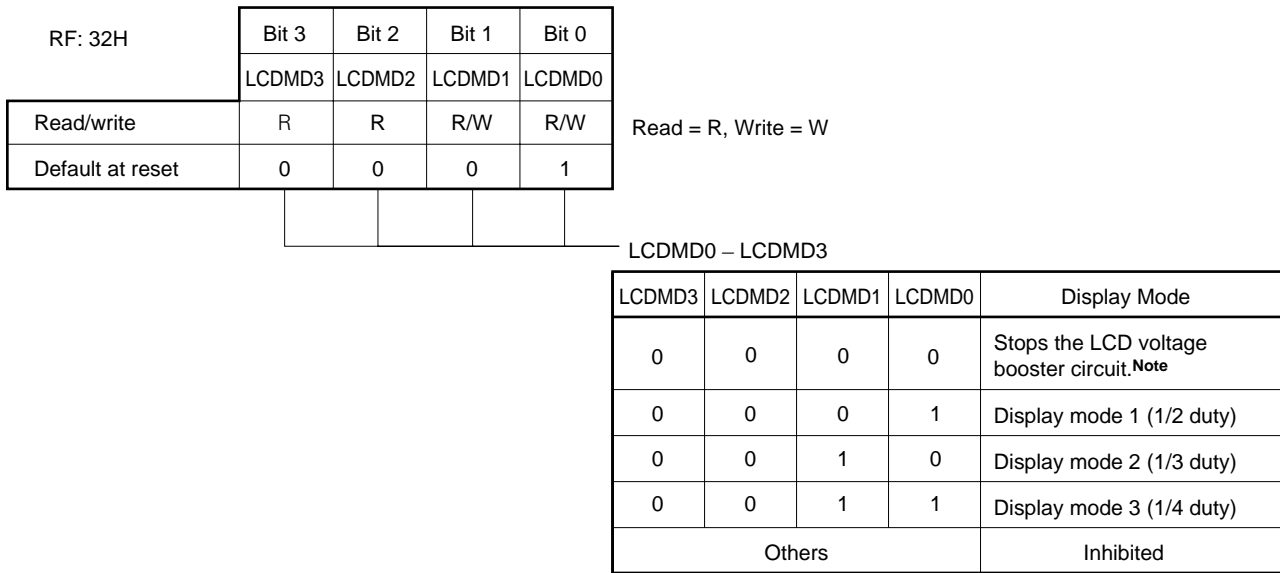
Mode	Duty	Common Signal	Maximum Number of Pixels
1	1/2	COM <sub>0</sub> , COM <sub>1</sub>	72 (36 segment signals by 2 common signals)
2	1/3	COM <sub>0</sub> , COM <sub>1</sub> , COM <sub>2</sub>	105 (35 segment signals by 3 common signals)
3	1/4	COM <sub>0</sub> , COM <sub>1</sub> , COM <sub>2</sub> , COM <sub>3</sub>	136 (34 segment signals by 4 common signals)

**11.3 DISPLAY MODE REGISTER**

The Display Mode register selects a display mode of the LCD controller/driver, a frame frequency, and LCD on/off status.

The display mode register consists of LCDMD0 to LCDMD3 (address 32H of register file) for selection of a display mode, LCDEN (bit 3, address 31H of the register file) for selection of the LCD on/off status, and LCDCK0 to LCDCK2 (bit 2 to bit 0, address 31H of the register file) for selection of a frame frequency.

Fig. 11-2 Display Mode Register



**Note** All segment and common signals are at a preset voltage ( $V_{LCD0}$ )

Fig. 11-3 LCD Controller/Driver Control Register

RF: 31H	Bit 3	Bit 2	Bit 1	Bit 0	
	LCDEN	LCDCK2	LCDCK1	LCDCK0	
Read/write	R/W	R	R/W	R/W	Read = R, Write = W
Default at reset	0	0	0	0	

LCDCK0 – LCDCK2

LCDCK2	LCDCK1	LCDCK0	Frame Frequency		
			Mode 1 1/2 duty	Mode 2 1/3 duty	Mode 3 1/4 duty
0	0	0	$f_w / (2 \times 2^6)$ [256 Hz]	$f_w / (3 \times 2^6)$ [170 Hz]	$f_w / (4 \times 2^6)$ [128 Hz]
0	0	1	$f_w / (2 \times 2^5)$ [512 Hz]	$f_w / (3 \times 2^5)$ [341.3 Hz]	$f_w / (4 \times 2^5)$ [256 Hz]
0	1	0	$f_w / (2 \times 2^8)$ [64 Hz]	$f_w / (3 \times 2^8)$ [42.6 Hz]	$f_w / (4 \times 2^8)$ [32 Hz]
0	1	1	$f_w / (2 \times 2^7)$ [128 Hz]	$f_w / (3 \times 2^7)$ [85.3 Hz]	$f_w / (4 \times 2^7)$ [64 Hz]
Others			Inhibited		

- Remark**
- $f_w = f_{xt}$  or  $f_w / 2^7$
  - [ ]: Frequency for  $f_w = 32.768$  kHz

LCDEN

LCDEN	Function
0	Turns off the LCD display. (all segment signals are on unselected)
1	Turns on LCD display.

- Cautions**
1. The LCD clock is supplied from the watch timer; therefore, the LCD flickers if the watch timer is reset during display. Do not reset the watch timer during display.
  2. If the main clock and subclock are used, the source clock of the LCD is the subclock. When the power is switched on, therefore, the LCD may flicker until the oscillation of the subclock stabilizes. Make sure that a sufficiently long time elapses until the oscillation stabilizes before turning on the LCD (it is recommended that all-light mode be used immediately after power application).
  3. The LCD display voltages ( $V_{LCD0}$ ,  $V_{LCD1}$ , and  $V_{LCD2}$ ) become undefined momentarily on turning ON/OFF power, reset, and setting or clearing the STOP mode. As a result, the LCD display may be turned ON (blurring of the LCD). This symptom is conspicuous when only the main clock is used or if the capacitance at the LCD display side is too high. To prevent this, take the following measures.
    - Provide wait time of 1 frame cycle or longer until the voltage booster circuit is stopped by the display mode register after the LCD display has been turned off.
    - Provide wait time of around 2 ms after the voltage booster circuit has been stopped until the STOP instruction is executed.

**11.4 LCD REGISTER**

The LCD register is resident on addresses 40H to 63H (LCDD0 to LCDD35) of BANK 0.  
 The LCD controller/driver reads the LCD register independently of the operation of the CPU.  
 The LCD controller controls segment signals according to the data of the LCD register.  
 The data memory area which is not used for LCD display can be used as ordinary data memory.  
 Figure 11-4 shows the assignment of segment outputs to bits of the LCD register.

**Fig. 11-4 Assignment of Common Signals and Segment Signals to LCD Register**

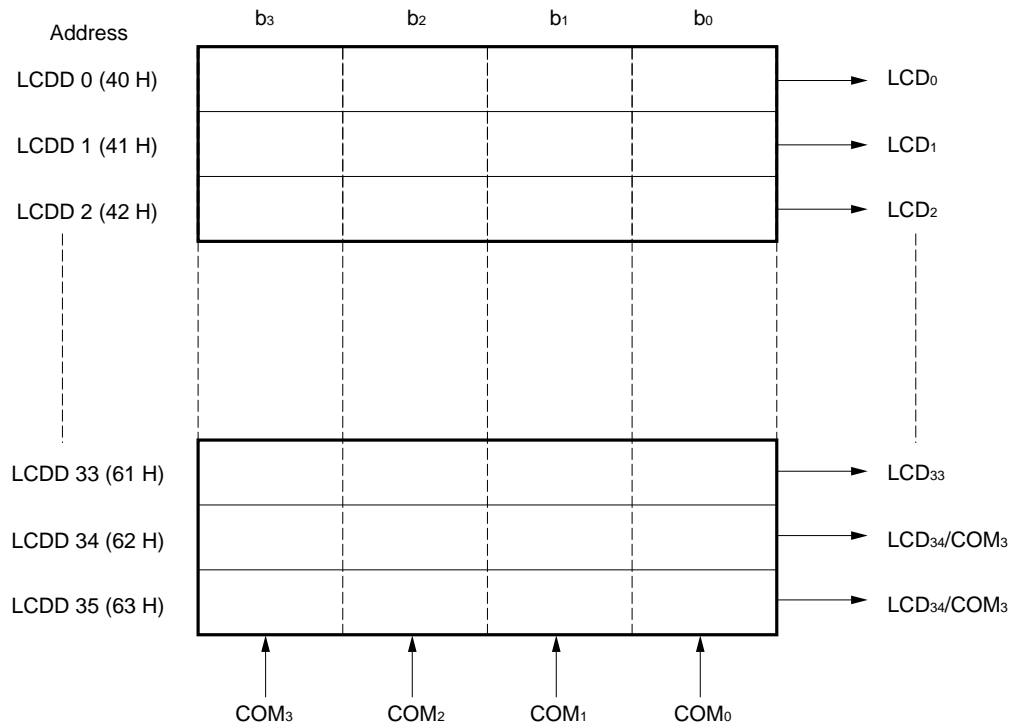
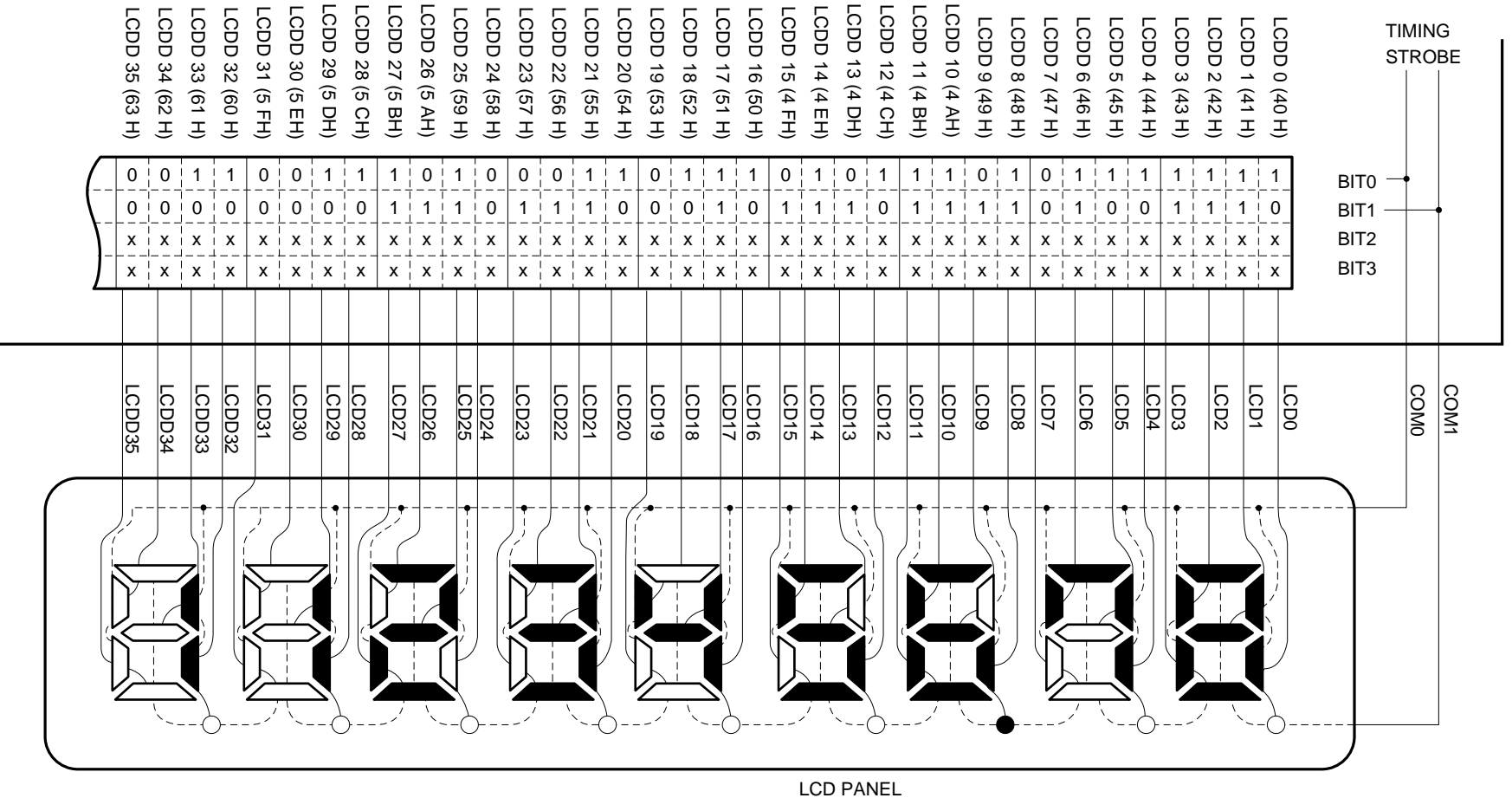


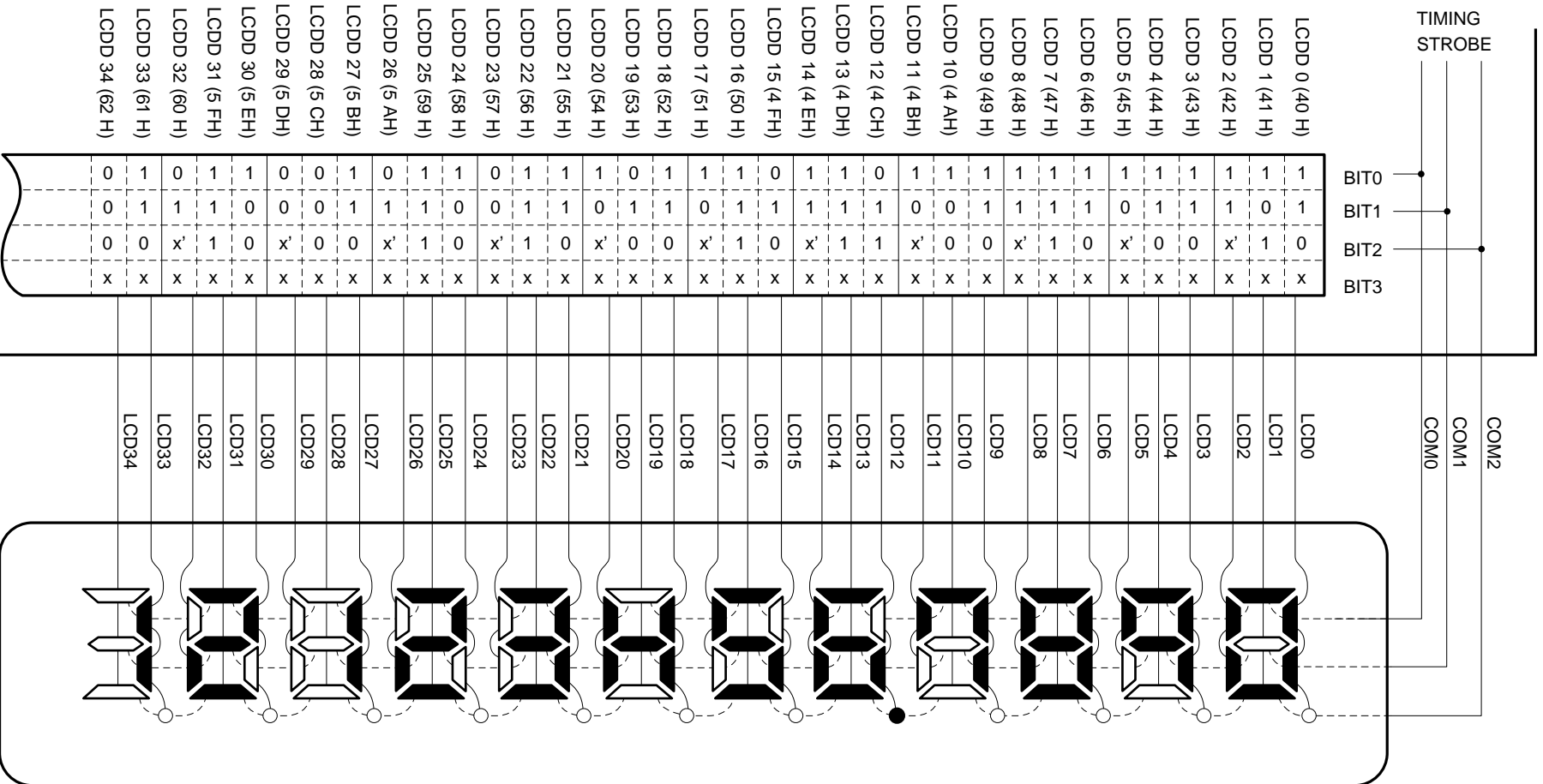
Fig. 11-5 Wiring Example of Secondary Time Sharing LCD Panel



x: Any value because of secondary time sharing display.

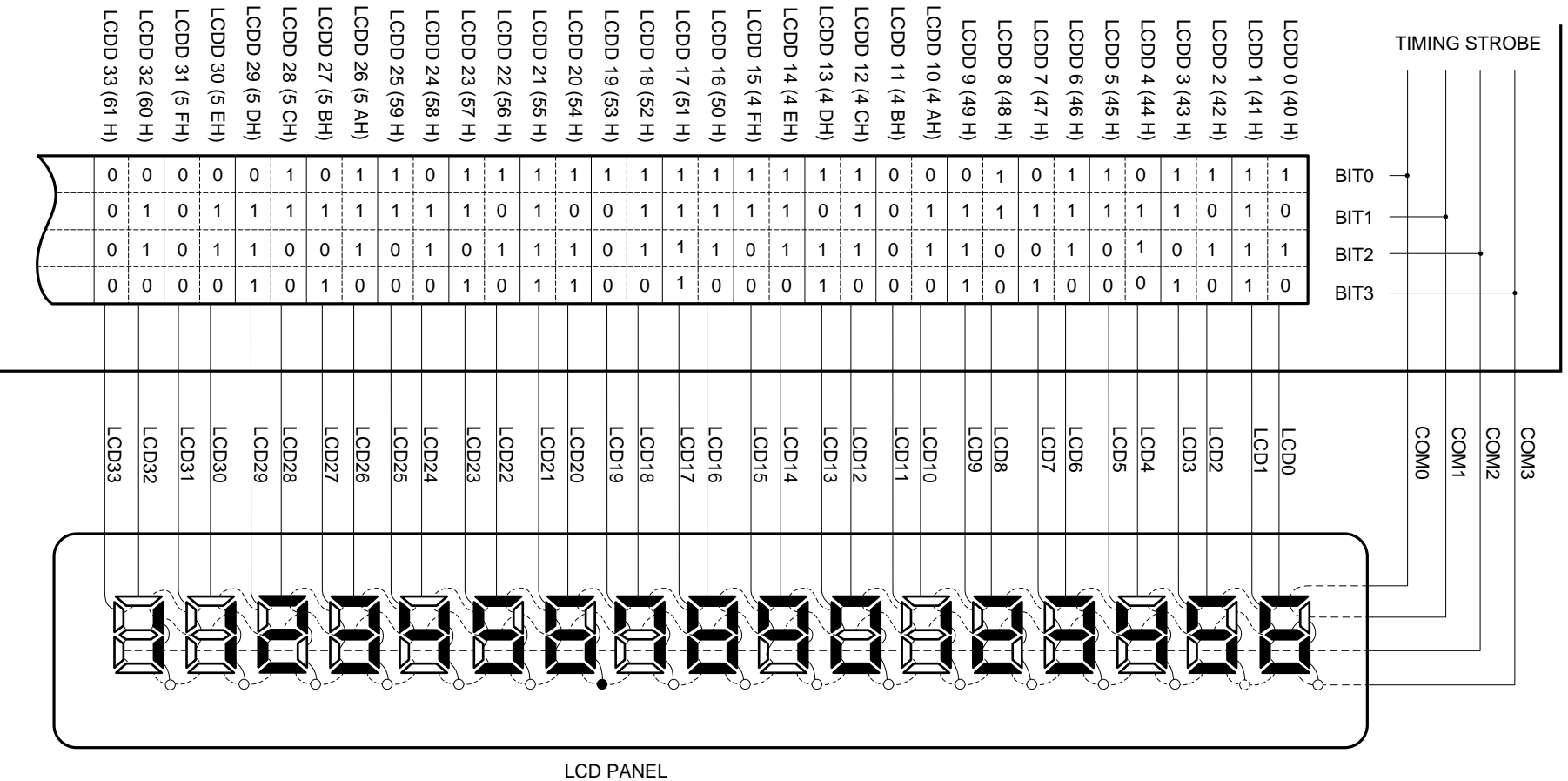


Fig. 11-6 Wiring Example of Tertiary Time Sharing LCD Panel



LCD PANEL    x' : Any data because it is not connected to any segment on the LCD panel.  
 x : Any data because of tertiary time-sharing display.

Fig. 11-7 Wiring Example of Quarternary Time Sharing LCD Panel



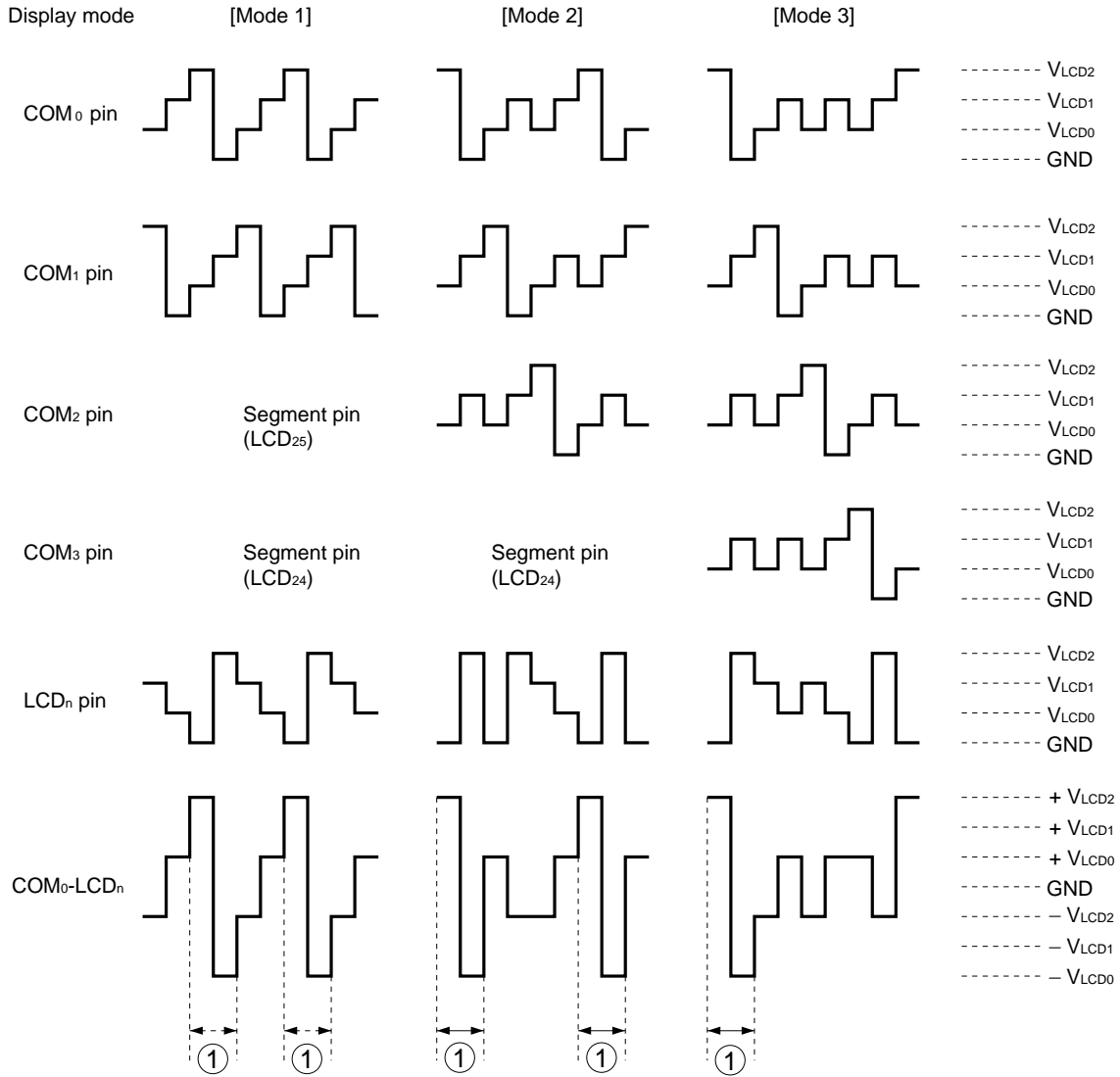
11.5 SEGMENT SIGNALS AND COMMON SIGNALS

Segment pins LCD<sub>0</sub> to LCD<sub>35</sub> are connected to the corresponding front electrodes of the LCD panel and common pins COM<sub>0</sub> to COM<sub>3</sub> are connected to corresponding rear electrodes of the LCD panel. The LCD panel lights when the potential difference between its segment and common signals goes beyond a preset voltage.

The LCD panel is driven on an AC voltage because it degrades quickly if a DC voltage is continuously applied between its segment and common pins.

Figure 11-8 to Fig. 11-10 shows waveforms of segment and common signals in each display mode.

Fig. 11-8 Common and Segment Waveforms in Each Display Mode (for LCDEN = 1)



① : The LCD stripe lights here (by an LCD selection voltage).

Fig. 11-9 Common and Segment Waveforms for LCDEN = 0 (LCD Display Off)

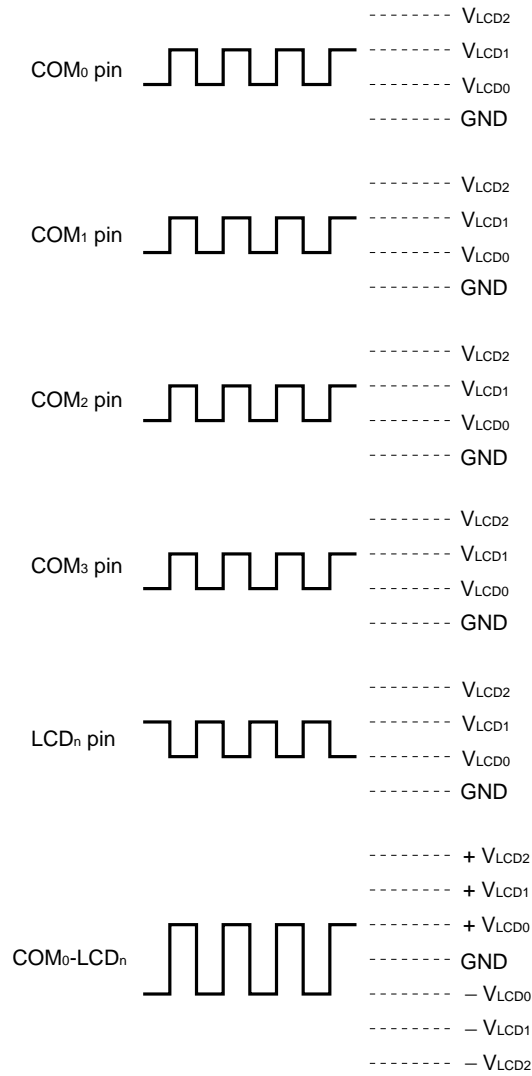
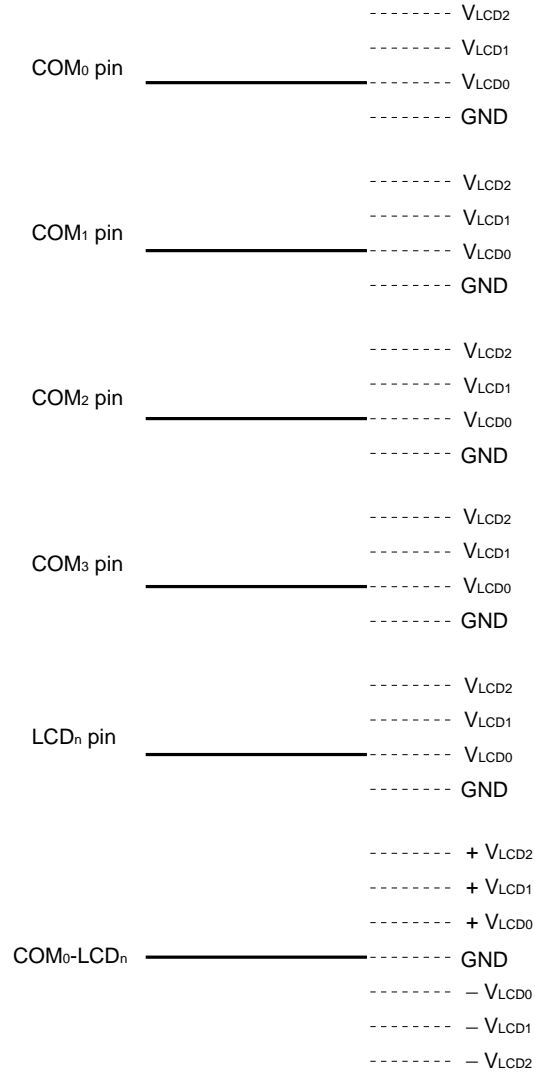


Fig. 11-10 Common and Segment Waveforms for LCDMD0 = 0 and LCDMD1 = 0  
(Voltage Booster Circuit Stop)



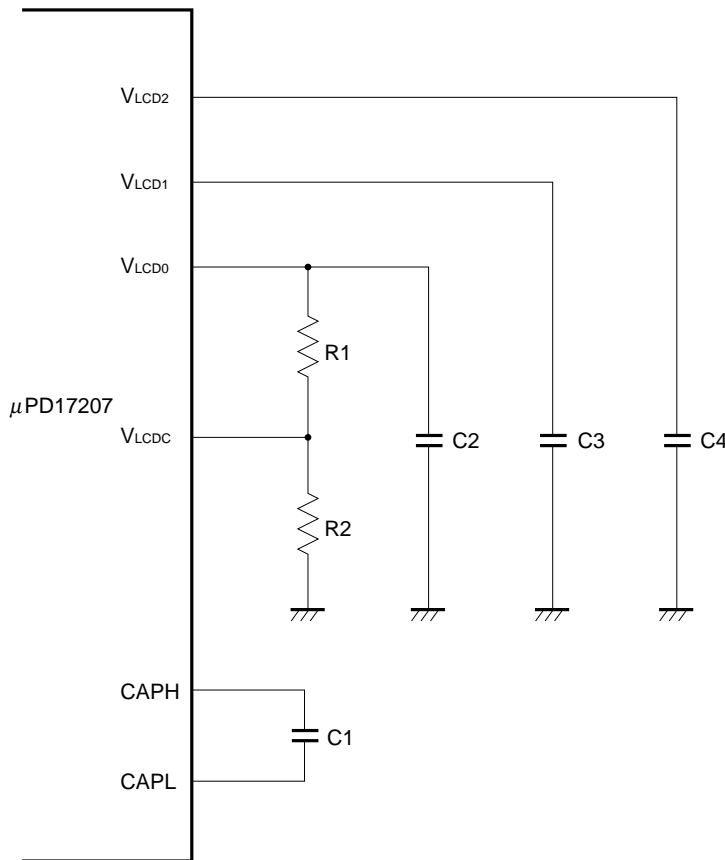
**11.6 VOLTAGE BOOSTER CIRCUIT FOR LCD DRIVER**

The μPD17207 has a voltage booster circuit for LCD driver which prevents the LCD from flickering when the supply voltage fluctuates.

Output signals V<sub>LCD2</sub>, V<sub>LCD1</sub>, and V<sub>LCD0</sub> of the segment and common signals are respectively two times (V<sub>LCD1</sub>), three times (V<sub>LCD2</sub>), and equal to the output (reference voltage, V<sub>LCD0</sub>) of the reference voltage generator. The reference voltage V<sub>LCD0</sub> can be adjusted by a resistor connected to the Reference Voltage Adjuster for LCD driver pin V<sub>LCDC</sub>.

Figure 11-11 shows an example of a circuit of adjusting the reference voltage for the LCD driver. Figure 11-12 shows its operating principle.

**Fig. 11-11 Reference Voltage Adjusting Circuit for LCD Driver (Example)**



R1 + R2 = 2MΩ  
 C1 = C2 = C3 = C4 = 0.47 μF

Reference voltage V<sub>LCD0</sub> can be adjusted by resistors R1 and R2.  
 Where V<sub>LCDC</sub> = 0.6 V,

$$V_{LCD0} = \frac{R1 + R2}{R2} \times 0.6 \text{ (V)}$$

$$V_{LCD1} = 2 \times V_{LCD0} \text{ (V)}$$

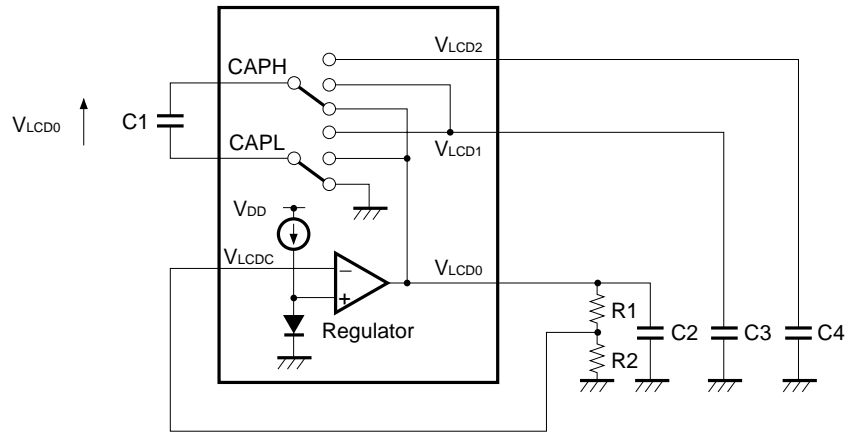
$$V_{LCD2} = 3 \times V_{LCD0} \text{ (V)}$$

**Caution** When the power is switched on, the LCD may light until the supply voltage stabilizes because the voltages of the capacitors for the voltage regulator and driver are undefined. It is therefore recommended that the all-light mode be used immediately after the power is switched on.

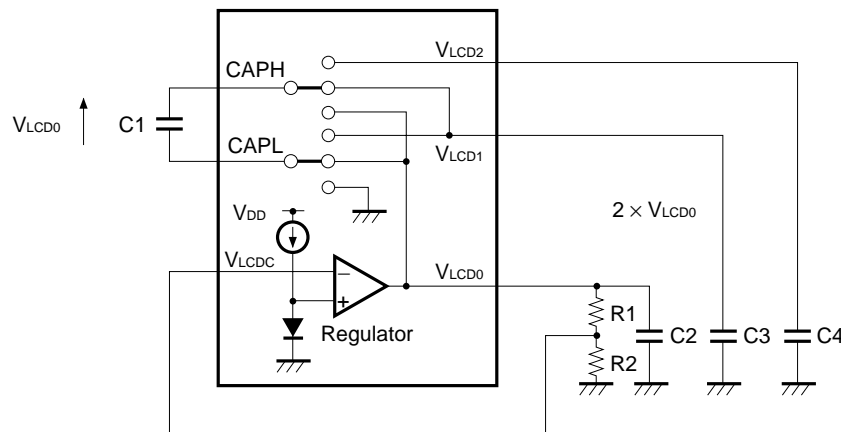
Fig. 11-12 Operating Principle of LCD Driver Voltage Booster Circuit

★

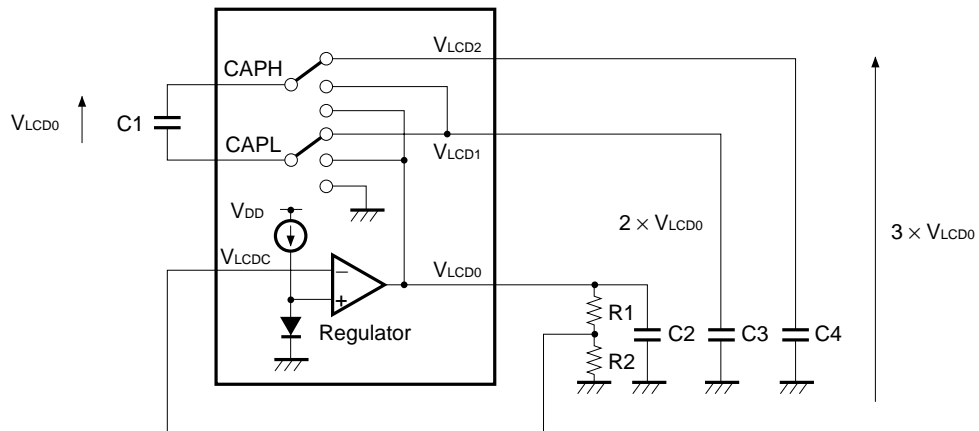
(1) Charge C1 with  $V_{LCD0}$  ( $V_{LCD0}$ ).



(2) Charge C3 with  $V_{LCD0}$  and voltage of C1 ( $V_{LCD0} + V_{LCD0} = 2 \times V_{LCD0}$ ).



(3) Charge C4 with voltage of C3 and voltage of C1 ( $2 \times V_{LCD0} + V_{LCD0} = 3 \times V_{LCD0}$ ).



(1) through (3) are repeated to boost the voltage. ( ) indicates the logical value eventually reached. The voltage is not necessarily boosted to the level in ( ) at a time.

**12. INTERRUPT FUNCTIONS**

When a peripheral hardware unit (INT pin, 8-bit timer, clock timer, or serial interface) makes an interrupt request, the interrupt function temporarily stops the execution of the current program and transfers program control to a predetermined address (termed a vector address).

**12.1 INTERRUPT SOURCES**

The μPD17207 supports the four interrupt sources (see Table 12-1).

When accepting an interrupt, the μPD17207 automatically transfers program control to a predetermined address (called a vector address).

**Table 12-1 Vector Addresses**

Priority	Interrupt Source	Vector Address
1	8-bit timer (Internal)	0004H
2	Rising edge of INT pin input (External)	0003H
3	Clock timer (Internal)	0002H
4	Serial interface (Internal)	0001H

If two or more interrupt requests are issued at the same time, the interrupt requests are accepted according to the priorities assigned to them.

Accepting an interrupt is enabled or disabled by the EI or DI instruction. Basically, the interrupt is accepted when it is enabled by the EI instruction. While the DI instruction is executed or while an interrupt is accepted, the other interrupts are disabled.

To enable accepting another interrupt after one interrupt has been completed, the EI instruction must be executed before the RETI instruction.

The interrupt is accepted by the EI instruction after the instruction next to EI has been executed; therefore, no interrupt is accepted between the EI and RETI instructions.

**12.2 HARDWARE OF INTERRUPT CONTROL CIRCUIT**

This section describes the flags of the interrupt control circuit.

(1) Interrupt request flag and interrupt enable flag

The interrupt request flag (IRQxxx) 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 (IPxxx) is provided to each interrupt request flag. When the IPxxx flag is 1, the interrupt is enabled; when it is 0, the interrupt is disabled.



(2) EI/DI instruction

Whether an accepted 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.

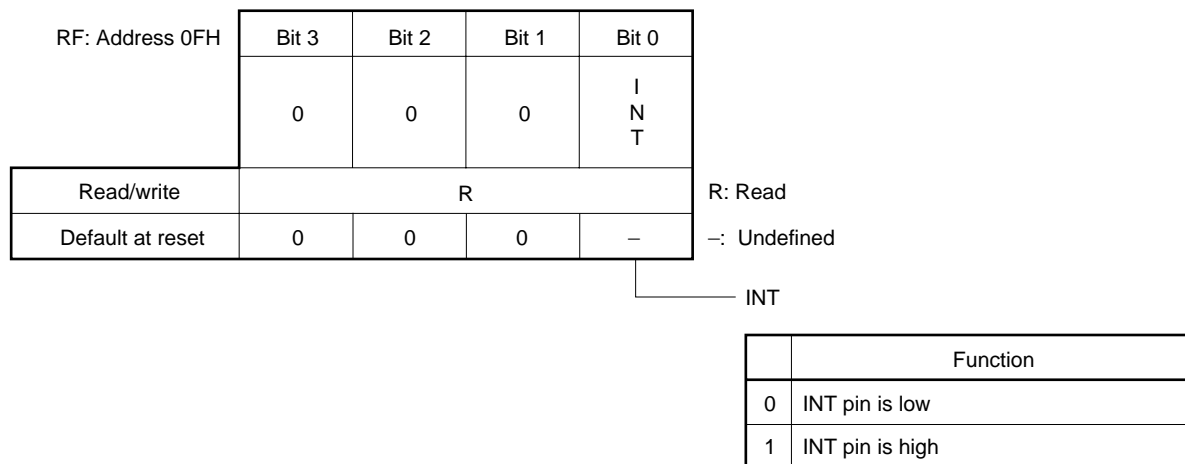
**Table 12-2 Interrupt Request Flags and Interrupt Enable Flags**

Interrupt Request Flag	Signal Setting Interrupt Request Flag	Interrupt Enable Flag
IRQ	Set when rising edge of INT pin input signal is detected	IP
IRQTM	Set by coincidence signal of 8-bit timer	IPTM
IRQWTM	Set by interrupt request signal from watch timer. Interrupt request signal generation interval is selected by WTMMD flag (RF: 03H, bit 2)	IPWTM
IRQSIO	Set by signal indicating end of serial data transfer operation from serial interface	IPSIO

**12.2.1 INT Flag**

This flag reads the status of the INT pin. This flag is “1” when a high-level signal is on the INT pin or “0” when a low-level signal is there.

**Fig. 12-1 INT flag**

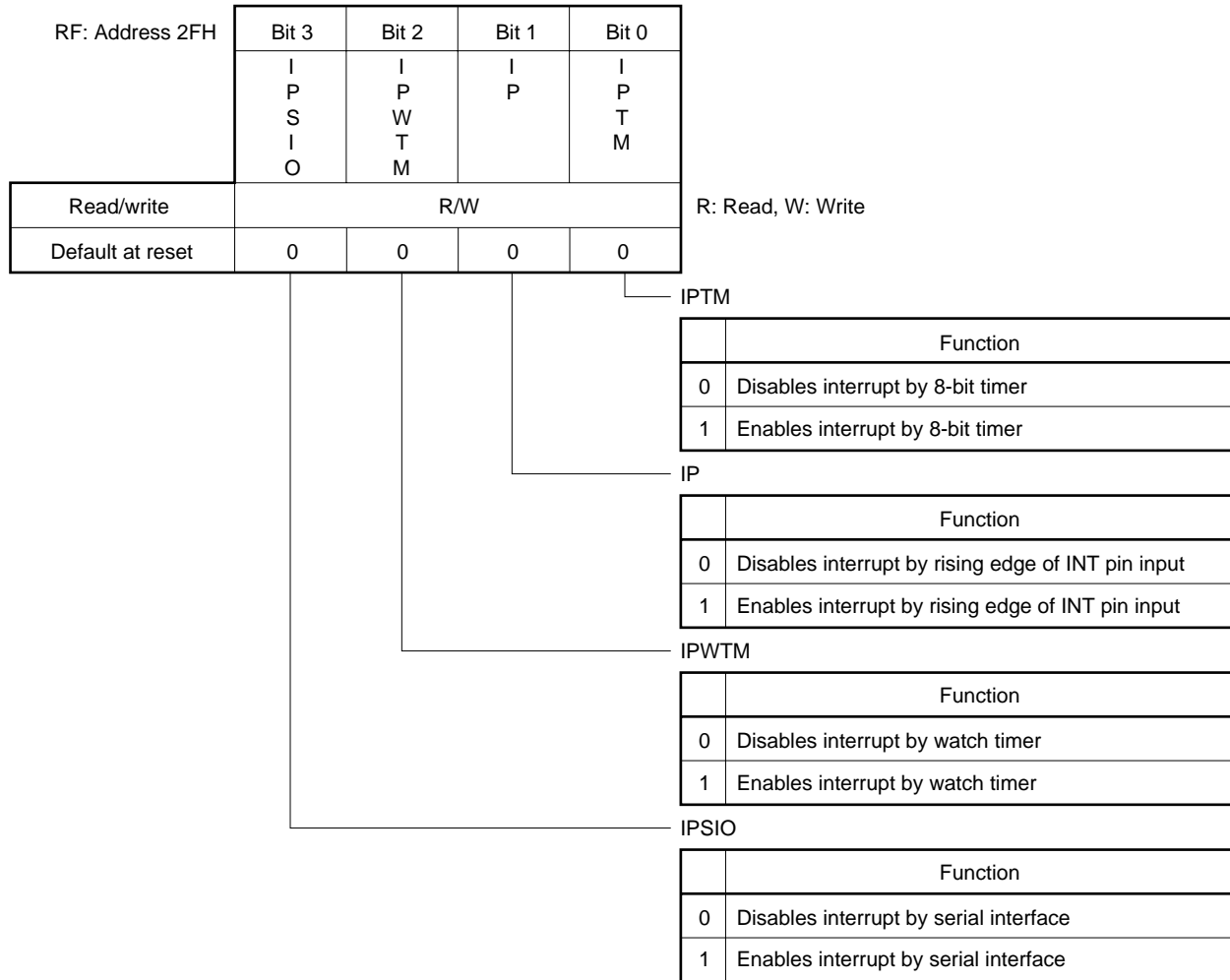


**12.2.2 Interrupt Enable Flags**

These flags enable the corresponding interrupts to be accepted.

- 1: The interrupt is accepted.
- 2: The interrupt is not accepted.

**Fig. 12-2 Interrupt Enable Flags**



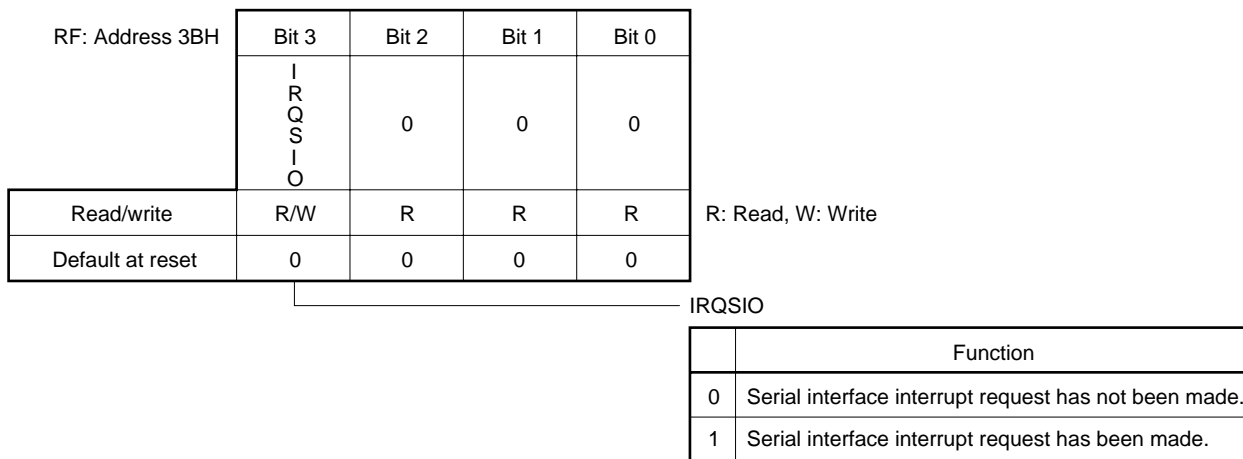
**12.2.3 Interrupt Request Flags**

These flags indicate the occurrence or acceptance of the corresponding interrupt requests.

- 1: The interrupt request has been made.
- 0: The interrupt request has been accepted.

It is possible to set the status of each Interrupt Request flag by programming. When you write "1" in an Interrupt Request flag, the corresponding interrupt can be generated by software. When you write "0" in an Interrupt Request flag, the corresponding interrupt being held is released.

**Fig. 12-3 Interrupt Request Flags (1/4)**



**Fig. 12-3 Interrupt Request Flags (2/4)**

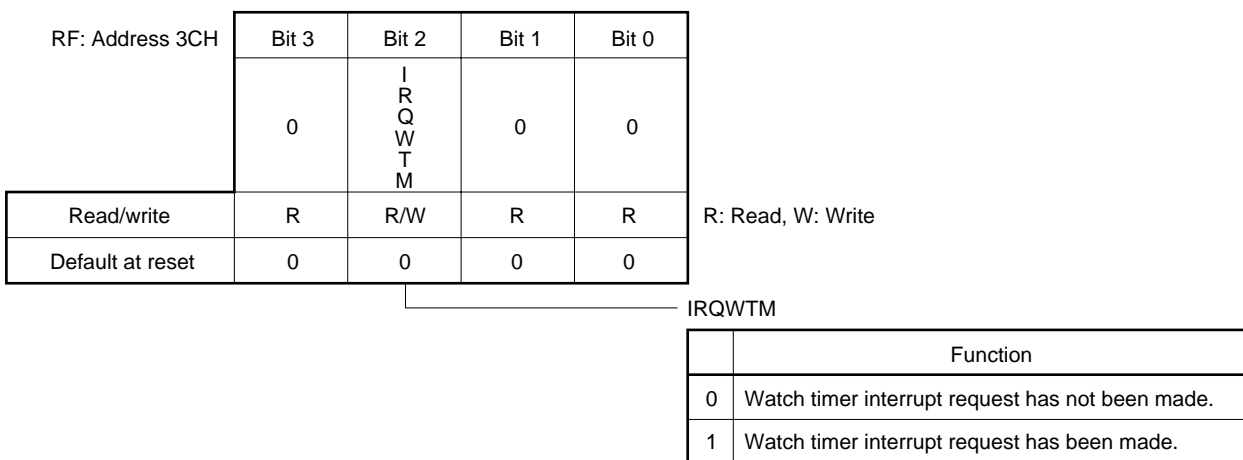


Fig. 12-3 Interrupt Request Flags (3/4)

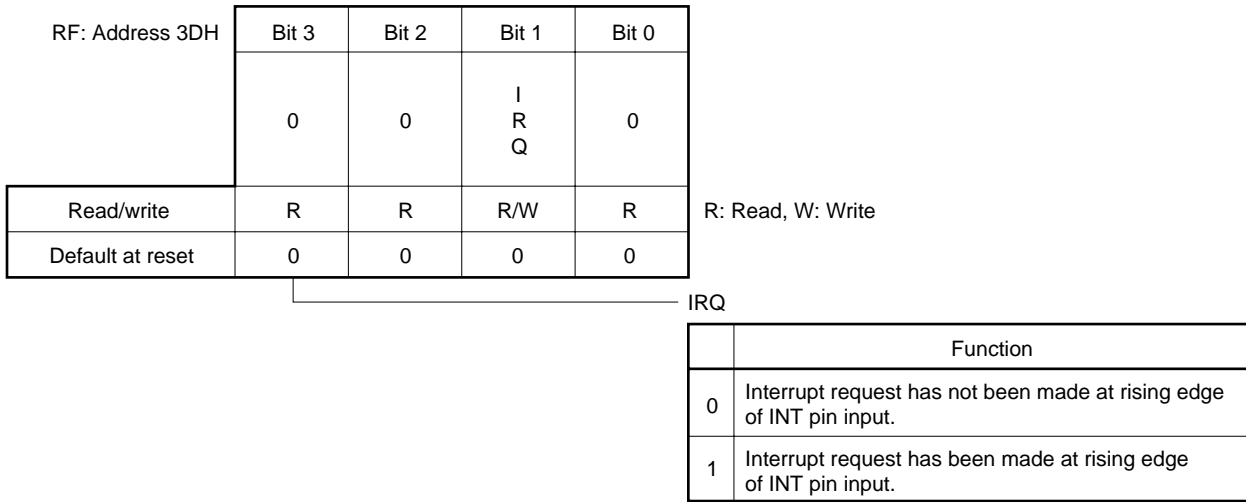
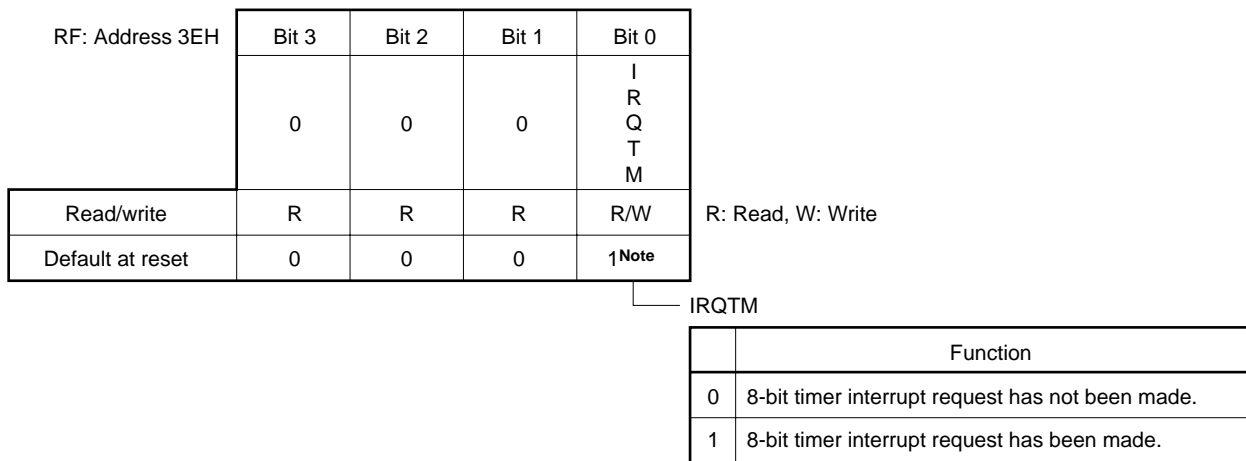


Fig. 12-3 Interrupt Request Flags (4/4)



**Note** 1H even after the STOP mode has been released.

### 12.3 INTERRUPT SEQUENCE

If the IRQxx flag is set to "1" while the IPxx is "1", processing of an interrupt starts at the end of the instruction cycle of the instruction being executed when the IRQxx flag was set. Processing of an instruction made in the execution of an MOVT instruction starts at the end of the second instruction cycle as the MOVT instruction runs in the second instruction cycle.

When the IPxx flag is "0", interrupt processing does not start until the IPxx flag is set even when the IRQxx flag is set.

When two or more interrupts are enabled, they are processed in the ascending order of priorities. (An interrupt must wait until processing of an interrupt of the higher priority ends.)

#### 12.3.1 Operations When Interrupt Is Accepted

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

- (1) Decrements the value of the stack pointer (SP) by 1.
- (2) Saves the current value of the program counter to the address stack register (ASR) specified by the SP.  
If the branch (BR) or subroutine call (CALL) instruction is executed when the interrupt has been accepted, the address of the program memory (ROM) to which execution is to branch, or called is loaded to the PC.
- (3) Saves the value of each flag (BCD, CMP, CY, Z, IXE) of the bank register (BANK) and program status word (PSWORD) to the interrupt stack register (INTSK, three levels).
- (4) Transfers the vector address to the PC.

One instruction cycle is required to perform the above processing.

#### 12.3.2 Returning from Interrupt Processing Routine

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

Then the following processing is executed within an instruction cycle.

- (1) Restores the value of the interrupt stack register (INTSK) to each flag (BCD, CMP, CY, Z, IXE) of the program status word (PSWORD).
- (2) Restores the value of the address stack register (ASR) specified by the stack pointer (SP) to the program counter.
- (3) Increments the value of the stack pointer by 1.

To accept another interrupt after an interrupt has been processed, it is necessary to execute an EI instruction before the RETI instruction.

An interrupt will never be accepted between the EI and RETI instructions as an interrupt is accepted by the EI instruction only after the next instruction has been executed.

### 13. STANDBY FUNCTIONS

The μPD17207 has two modes of standby functions: HALT mode and STOP mode. The standby functions reduce the power dissipation of the μPD17207.

In HALT mode, the μPD17207 stops the execution of the program with the main clock on. (The CPU stops to run.) This mode is kept until a HALT releasing condition is satisfied.

In STOP mode, the μPD17207 stops the execution of the program with the main clock off. The μPD17207 dissipates less circuit current in STOP mode than in HALT mode.

HALT mode is set by the execution of a HALT instruction and STOP mode is set by the execution of a STOP instruction.

#### 13.1 HALT MODE

In HALT mode, the μPD17207 stops the execution of the program with the main clock on for reduction of its power dissipation.

Execute a HALT instruction to set HALT mode.

The condition of releasing HALT mode is determined by the operand of the HALT instruction. See Table 13-1. After HALT mode is released, the μPD17207 performs operations as shown in Table 13-2.

**Caution** Do not execute an instruction to clear the interrupt request flag (IRQxxx) whose interrupt enable flag (IPxxx) is set immediately before the HALT 8H or HALT 0AH instruction is executed. If the flag is cleared, the HALT mode may not be set.

**Table 13-1 HALT Mode Releasing Conditions**

Operand Value	HALT Mode Releasing Condition
0010B (02H)	1) When an 8-bit timer interrupt request (IRQTM) is made
1000B (08H)	1) When an interrupt request (IRQTM, IRQWTM, IRQSIO, or IRQ) is made for an interrupt whose enable flag (IPTM, IPWTM, IPSIO, or IP) is on ("1") 2) When any of pins P0A <sub>0</sub> to P0A <sub>3</sub> goes low
1010B (0AH)	1) When an 8-bit timer interrupt request (IRQTM) is made 2) When an interrupt request (IRQWTM, IRQSIO, or IRQ) is made for an interrupt whose enable flag (IPWTM, IPSIO, or IP) is on ("1")
Others	Inhibited

Table 13-2 Operations after HALT Mode Has Been Released

(a) HALT 02H

Standby Mode Released by:	Interrupt Enable Status	Interrupt Enable Flag	Operation after Release of Standby Mode
Satisfaction of Release Condition by Interrupt Request (IRQTM)	DI	Disable	Execution starts from instruction following HALT
		Enable	
	EI	Disable	Branches to vector address of interrupt
		Enable	

(b) HALT 08H

Standby Mode Released by:	Interrupt Enable Status	Interrupt Enable Flag	Operation after Release of Standby Mode
Low Level Input to Port 0A	Don't care	Don't care	Execution starts from instruction following HALT
Satisfaction of Release Condition by Interrupt Request (IRQTM, IRQWTM, IRQSIO, or IRQ)	DI	Disable	Standby mode is not released
		Enable	Execution starts from instruction following HALT
	EI	Disable	Standby mode is not released
		Enable	Branches to vector address of interrupt

(c) HALT 0AH

Standby Mode Released by:	Interrupt Enable Status	Interrupt Enable Flag	Operation after Release of Standby Mode
Satisfaction of Release Condition by Interrupt Request (IRQTM)	DI	Disable	Execution starts from instruction following HALT
		Enable	
	EI	Disable	Branches to vector address of interrupt
		Enable	
Satisfaction of Release Condition by Interrupt Request (IRQWTM, IRQSIO, or IRQ)	DI	Disable	Standby mode is not released
		Enable	Execution starts from instruction following HALT
	EI	Disable	Standby mode is not released
		Enable	Branches to vector address of interrupt

**13.2 CONDITIONS OF EXECUTING AN HALT INSTRUCTION**

The HALT instruction can be executed only under a specific condition for prevention of malfunction. See Table 13-3.

The HALT instruction which does not satisfy the conditions listed in Table 13-3 is treated as an NOP instruction.

**Table 13-3 Conditions of Executing the HALT Instruction**

Operand Value	Execution Condition
0010B (02H)	1) The 8-bit timer interrupt request (IRQTM) should be reset.
1000B (08H)	1) The interrupt request flag (IRQTM, IRQWTM, IRQSIO, or IRQ) should be reset for an interrupt whose enable flag (IPTM, IPWTM, IPSIO, or IP) is on ("1") 2) All of pins P0A <sub>0</sub> to P0A <sub>3</sub> should be high input or output. 3) All of pins P0B <sub>0</sub> to P0B <sub>3</sub> should be in output mode and output latch should be "0".
1010B (0AH)	1) The 8-bit timer interrupt request (IRQTM) should be reset. 2) The interrupt request flag (IRQWTM, IRQSIO, or IRQ) should be reset for an interrupt whose enable flag (IPWTM, IPSIO, or IP) is on ("1")
Others	Reserved



**13.3 STOP MODE**

In STOP mode, the μPD17207 stops the execution of the program with the main clock temporarily off for great reduction of its power dissipation. Execute a STOP instruction to set STOP mode.

The STOP instruction is not valid for a system using a subclock only. When the system uses a subclock as the system clock (that is, when SYSCK = 0), the STOP instruction is treated as an NOP instruction.

The condition of releasing STOP mode is determined by the operand of the STOP instruction. See Table 13-4. After STOP mode is released, the μPD17207 performs operations as follows:

- ① Clearing IRQTM
- ② Starting watch timer and watchdog timer (not reset)
- ③ Resetting and starting 8-bit timer
- ④ The instruction following STOP 8H or the interrupt vector address is executed when the value of the 8-bit counter coincides with the value of the modulo register (setting of IRQTM).

**Caution** When the subclock is used, the watch timer and watchdog timer do not stop even in the STOP mode.

The time interval between release of STOP mode and start of the execution of the next instruction is set by the contents of the modulo register of the 8-bit timer. This time interval is expressed as follows:

$$(TMM + 1) \times 1024/f_x \text{ [sec]}$$

where,

TMM : Content of the modulo register

f<sub>x</sub> : Frequency of the main clock.

Example: In a system using the main clock of 4 MHz, the time interval between release of STOP mode and start of the execution of the next instruction is:

$$(TMM + 1) \times 256 \text{ [microseconds]}$$

**Caution** Do not set an instruction that would clear the interrupt request flag (IRQ<sub>xxx</sub>) whose interrupt enable flag (IP<sub>xxx</sub>) is set immediately before the STOP 8H instruction, if you set such an instruction, the STOP mode may not be set.

**Table 13-4 STOP Mode Releasing Conditions**

Operand Value	STOP Mode Releasing Condition
1000B (08H)	<ul style="list-style-type: none"> <li>① When an interrupt request (IRQW<sub>TM</sub>, IRQS<sub>IO</sub>, or IRQ) is made for an interrupt whose enable flag (IPW<sub>TM</sub>, IPS<sub>IO</sub>, or IP) is on ("1")</li> <li>② When any of pins P0A<sub>0</sub> to P0A<sub>3</sub> goes low</li> </ul>
Others	Inhibited

13.4 CONDITIONS OF EXECUTING AN HALT INSTRUCTION

The STOP instruction can be executed only under a specific condition for prevention of malfunction. See Table 13-5.

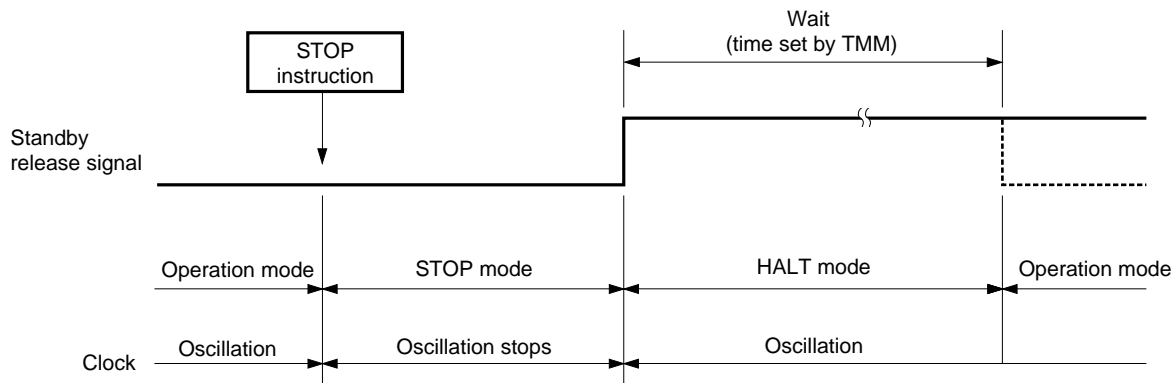
The STOP instruction which does not satisfy the conditions listed in Table 13-5 is treated as an NOP instruction.

Table 13-5 Conditions of Executing the STOP Instruction

Operand Value	Execution Condition
1000B (08H)	1) The interrupt request flag should be reset for an interrupt whose enable flag (IPWTM, IPSIO, or IP) is on ("1") 2) All of pins P0A <sub>0</sub> to P0A <sub>3</sub> should be high input or output. 3) All of pins P0B <sub>0</sub> to P0B <sub>3</sub> should be in output mode and output latch should be "0".
Others	Inhibited

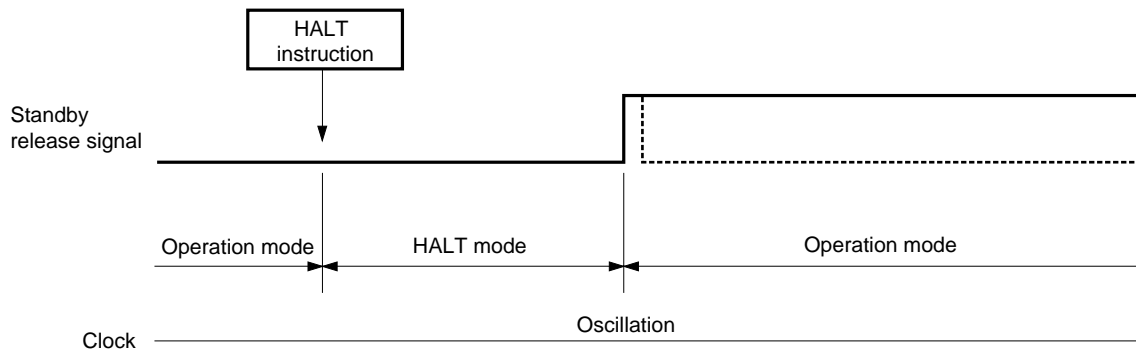
Fig. 13-1 Releasing Standby Mode

(a) Releasing STOP mode by interrupt



**Remark** The dotted line indicates when the interrupt that has released the standby mode is accepted

(b) Releasing HALT mode by interrupt



**Remark** The dotted line indicates when the interrupt that has released the standby mode is accepted

**14. RESET**

**14.1 RESET BY RESET SIGNAL INPUT**

When a low-level signal is input to the  $\overline{\text{RESET}}$  pin for 50 μs or more, the system is reset.

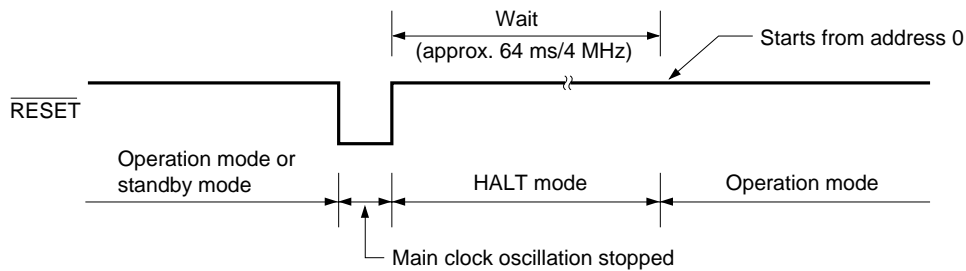
The system must be reset at least once when the power is turned on, as the operation of the internal circuit is undefined.

When reset has been effected, the following circuits are initialized:

- ① The program counter is reset to 0.
- ② The flags of the register file are initialized (for the initial values, refer to Fig. 15-1 Register File List).
- ③ Initial value 0320H is written to the data buffer.
- ④ The peripheral hardware is initialized.
- ⑤ Oscillation of the main clock is stopped.

When the  $\overline{\text{RESET}}$  pin is made high, oscillation of the main clock is started, and about 64 ms after that (where the main clock frequency is 4 MHz), execution of the program is started from address 0.

**Fig. 14-1 Reset Operation by  $\overline{\text{RESET}}$  Input**



**14.2 RESET BY WATCHDOG TIMER ( $\overline{\text{RESET}}$  AND  $\overline{\text{WDOUT}}$  PINS CONNECTED)**

When the watchdog timer is activated while the program is being executed, a low level is output to the  $\overline{\text{WDOUT}}$  pin, and the program counter is reset to 0.

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

When developing a program, reset the watchdog timer (set the WDTRES flag) at intervals of 340 ms or less (where  $f_x = 4$  MHz).

**14.3 RESET BY STACK POINTER ( $\overline{\text{RESET}}$  AND  $\overline{\text{WDOUT}}$  PINS CONNECTED)**

When the value of the stack pointer reaches 6H or 7H while the program is being executed, a low level is output to the  $\overline{\text{WDOUT}}$  pin, and the program counter is reset to 0.

When the level of nesting of interrupt or subroutine call exceeds 5 (stack overflow), or when the return instruction is executed despite the stack level being 0 because the call instruction and return (RET) instruction have not been correctly used in pairs (stack underflow), the program can be restarted from address 0.

Table 14-1 Hardware Status after Reset

Hardware		$\overline{\text{RESET}}$ Input in Standby Mode	$\overline{\text{RESET}}$ Input during Operation
Program Counter (PC)		0000H	0000H
Port	I/O	Input	Input
	Output Latch	0	0
Data Memory (RAM)	General-Purpose Data Memory (except DBF and port register)	Holds previous status	Undefined
	DBF	0320H	0320H
	System Register (SYSREG)	0	0
	WR	Holds previous status	Undefined
Control Register		Refer to Fig. 15-1 Register File List.	
8-bit Timer	Counter (TMC)	00H	00H
	Modulo Register (TMM)	FFH	FFH
Remote Controller Carrier Generator Circuit	NRZ High-Level Period Setting Modulo Register (NRZHTMM)	Holds previous status	Undefined
	NRZ Low-Level Period Setting Modulo Register (NRZLTMM)		
Counter of Watch Timer/Watchdog Timer		00H	00H
Shift Register of Serial Interface (SIOSFR)		Holds previous status	Undefined
Internal Reference Voltage Setting Register of A/D Converter (ADCR)		Holds previous status	Undefined

**15. ASSEMBLER RESERVED WORDS**

**15.1 MASK OPTION DIRECTIVES**

In μPD17207 programming, it is required to specify mask options in assembler source programs by mask option directives.

The following mask options items must be specified:

- Pull-up resistor for the RESET pin
- Connection between the main clock and the subclock (for selection of system clock)

**15.1.1 OPTION and ENDOP Directives**

A mask option is defined in a block between the OPTION and ENDOP directives. This block is formatted as shown below.

Coding format:

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

**15.1.2 Mask Option Definition Pseudo Directives**

Table 15-1 shows directives available in the mask option definition block.

**Table 15-1 Mask Option Definition Directives**

Item	Directives	Number of Operands	1st Operand	2nd Operands
RESET pin pull-up resistor	OPTRES	1	RESET mask option	
			RESPLUP (Built-in pull-up resistor) OPEN (No pull-up resistor)	
System clock	OPTCK	2	Main clock	Subclock
			USEX (Uses the main clock as the system clock.) NOX (Does not use the main clock.)	USEXT (Uses the subclock as the system clock.) NOXT (Does not use the subclock.)

**Remark** When both the main clock and the subclock are specified as the system clock, the main clock is initially selected when the system is reset. After that, the subclock can be selected by an instruction in the program.

Shown below is a coding example of mask options.

Symbol	Mnemonic	Operand	Comment
[Label: ]	OPTION		[;Comment]
	OPTRES	RESPLUP	; Pulls up the RESET pin.
	OPTCK	USEX, USEXT	; Uses the main clock or subclock.
	ENDOP		

**15.2 RESERVED SYMBOLS**

Table 15-2 lists the symbols defined by the device file of the μPD17207.

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

**15.2.1 Register File**

The symbol names assigned to the registers in the register file are defined. These registers are accessed via WR (window register) by the PEEK and POKE instructions. Fig. 15-1 lists the registers in the register file.

**15.2.2 Registers on Data Memory and Ports**

The names of the registers assigned to data memory addresses 00H-7FH, ports assigned to address 70H and those that follow, and system registers are defined. Fig. 15-2 shows the configuration of the data memory.

**15.2.3 Peripheral Hardware**

The names of the peripheral hardware that is accessed by the GET and PUT instructions are defined. Table 15-3 lists the peripheral hardware.

Table 15-2 List of Reserved Symbols (1/4)

Symbol Name	Attribute	Value	R/W	Description
DBF3	MEM	0.0CH	R/W	Bit 15 to bit 12 of data buffer
DBF2	MEM	0.0DH	R/W	Bit 11 to bit 8 of data buffer
DBF1	MEM	0.0EH	R/W	Bit 7 to bit 4 of data buffer
DBF0	MEM	0.0FH	R/W	Bit 3 to bit 0 of data buffer
AR3	MEM	0.74H	R	Bit 15 to bit 12 of address register
AR2	MEM	0.75H	R/W	Bit 11 to bit 8 of address register
AR1	MEM	0.76H	R/W	Bit 7 to bit 4 of address register
AR0	MEM	0.77H	R/W	Bit 3 to bit 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	Bit 11 to bit 8 of index register
MPH	MEM	0.7AH	R/W	Bit 7 to bit 4 of memory pointer
MPE	FLG	0.7AH.3	R/W	Memory pointer enable flag
IXM	MEM	0.7BH	R/W	Bit 7 to bit 4 of index register
MPL	MEM	0.7BH	R/W	Bit 3 to bit 0 of memory pointer
IXL	MEM	0.7CH	R/W	Bit 3 to bit 0 of index register
RPH	MEM	0.7DH	R/W	Bit 7 to bit 4 of register pointer
RPL	MEM	0.7EH	R/W	Bit 3 to bit 0 of register pointer
PSW	MEM	0.7FH	R/W	Program status word
BCD	FLG	0.7EH.0	R/W	BCD operation flag
CMP	FLG	0.7FH.3	R/W	Compare flag
CY	FLG	0.7FH.2	R/W	Carry flag
Z	FLG	0.7FH.1	R/W	Zero flag
IXE	FLG	0.7FH.0	R/W	Index register enable flag
LCDD0	MEM	0.40H	R/W	LCD segment 0
LCDD1	MEM	0.41H	R/W	LCD segment 1
LCDD2	MEM	0.42H	R/W	LCD segment 2
LCDD3	MEM	0.43H	R/W	LCD segment 3
LCDD4	MEM	0.44H	R/W	LCD segment 4
LCDD5	MEM	0.45H	R/W	LCD segment 5
LCDD6	MEM	0.46H	R/W	LCD segment 6
LCDD7	MEM	0.47H	R/W	LCD segment 7
LCDD8	MEM	0.48H	R/W	LCD segment 8
LCDD9	MEM	0.49H	R/W	LCD segment 9
LCDD10	MEM	0.4AH	R/W	LCD segment 10
LCDD11	MEM	0.4BH	R/W	LCD segment 11
LCDD12	MEM	0.4CH	R/W	LCD segment 12
LCDD13	MEM	0.4DH	R/W	LCD segment 13
LCDD14	MEM	0.4EH	R/W	LCD segment 14
LCDD15	MEM	0.4FH	R/W	LCD segment 15

Table 15-2 List of Reserved Symbols (2/4)

Symbol Name	Attribute	Value	R/W	Description
LCDD16	MEM	0.50H	R/W	LCD segment 16
LCDD17	MEM	0.51H	R/W	LCD segment 17
LCDD18	MEM	0.52H	R/W	LCD segment 18
LCDD19	MEM	0.53H	R/W	LCD segment 19
LCDD20	MEM	0.54H	R/W	LCD segment 20
LCDD21	MEM	0.55H	R/W	LCD segment 21
LCDD22	MEM	0.56H	R/W	LCD segment 22
LCDD23	MEM	0.57H	R/W	LCD segment 23
LCDD24	MEM	0.58H	R/W	LCD segment 24
LCDD25	MEM	0.59H	R/W	LCD segment 25
LCDD26	MEM	0.5AH	R/W	LCD segment 26
LCDD27	MEM	0.5BH	R/W	LCD segment 27
LCDD28	MEM	0.5CH	R/W	LCD segment 28
LCDD29	MEM	0.5DH	R/W	LCD segment 29
LCDD30	MEM	0.5EH	R/W	LCD segment 30
LCDD31	MEM	0.5FH	R/W	LCD segment 31
LCDD32	MEM	0.60H	R/W	LCD segment 32
LCDD33	MEM	0.61H	R/W	LCD segment 33
LCDD34	MEM	0.62H	R/W	LCD segment 34
LCDD35	MEM	0.63H	R/W	LCD segment 35
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
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
P1A3	FLG	1.70H.3	R/W	Bit 3 of port 1A



Table 15-2 List of Reserved Symbols (3/4)

Symbol Name	Attribute	Value	R/W	Description
SP	MEM	0.81H	R/W	Stack pointer
SYSCK	FLG	0.82H.1	R/W	Selection of system clock
XEN	FLG	0.82H.0	R/W	Main clock enable
WDTRES	FLG	0.83H.3	R/W	Watchdog timer reset
WTMMD	FLG	0.83H.2	R/W	Selection of watch timer mode
WTMRES	FLG	0.83H.1	R/W	Reset of watch timer mode
INT	FLG	0.8FH.0	R	INT pin status
NRZBF	FLG	0.91H.0	R/W	NRZ buffer data
NRZ	FLG	0.92H.0	R/W	NRZ data
ADCCMP	FLG	0.0A0H.0	R/W	Comparator result
VREFEN	FLG	0.0A1H.3	R/W	A/D converter enable flag
ADCEN	FLG	0.0A1H.2	R/W	A/D converter enable flag
ADCCH1	FLG	0.0A1H.1	R/W	A/D converter channel selection flag
ADCCH0	FLG	0.0A1H.0	R/W	A/D converter channel selection flag
SIOTS	FLG	0.0A2H.3	R/W	Serial interface start flag
SIOHIZ	FLG	0.0A2H.2	R/W	SO pin status
SIOCK1	FLG	0.0A2H.1	R/W	Serial clock selection flag for serial interface
SIOCK0	FLG	0.0A2H.0	R/W	Serial clock selection flag for serial interface
NRZEN	FLG	0.0A3H.2	R/W	NRZ enable flag
TMOE	FLG	0.0A3H.1	R/W	Timer output enable flag
SIOEN	FLG	0.0A3H.0	R/W	SIO enable flag
P0DBIO3	FLG	0.0A7H.3	R/W	I/O setting flag (bit 3 of port P0D)
P0DBIO2	FLG	0.0A7H.2	R/W	I/O setting flag (bit 2 of port P0D)
P0DBIO1	FLG	0.0A7H.1	R/W	I/O setting flag (bit 1 of port P0D)
P0DBIO0	FLG	0.0A7H.0	R/W	I/O setting flag (bit 0 of port P0D)
IPSIO	FLG	0.0AFH.3	R/W	INTSIO interrupt enable flag
IPWTM	FLG	0.0AFH.2	R/W	Watch timer interrupt enable flag
IP	FLG	0.0AFH.1	R/W	Interrupt enable flag
IPTM	FLG	0.0AFH.0	R/W	8-Bit timer interrupt enable flag
LCDEN	FLG	0.0B1H.3	R/W	LCD display enable flag
LCDCK2	FLG	0.0B1H.2	R/W	LCD display clock selection flag
LCDCK1	FLG	0.0B1H.1	R/W	LCD display clock selection flag
LCDCK0	FLG	0.0B1H.0	R/W	LCD display clock selection flag
LCMDM3	FLG	0.0B2H.3	R/W	LCD display mode register bit 3
LCMDM2	FLG	0.0B2H.2	R/W	LCD display mode register bit 2
LCMDM1	FLG	0.0B2H.1	R/W	LCD display mode register bit 1
LCMDM0	FLG	0.0B2H.0	R/W	LCD display mode register bit 0
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	Selection of timer clock source

Table 15-2 List of Reserved Symbols (4/4)

Symbol Name	Attribute	Value	R/W	Description
TMCK0	FLG	0.0B3H.0	R/W	Selection of timer clock source
P1AGIO	FLG	0.0B7H.3	R/W	I/O setting flag for port 1A
P0CGIO	FLG	0.0B7H.2	R/W	I/O setting flag for port 0C
P0BGIO	FLG	0.0B7H.1	R/W	I/O setting flag for port 0B
P0AGIO	FLG	0.0B7H.0	R/W	I/O setting flag for port 0A
IRQSIO	FLG	0.0BBH.3	R/W	SIO interrupt request flag
IRQWTM	FLG	0.0BCH.2	R/W	Watch timer interrupt request flag
IRQ	FLG	0.0BDH.1	R/W	INT interrupt request flag
IRQTM	FLG	0.0BEH.0	R/W	8-bit timer interrupt request flag
SIOSFR	DAT	01H	R/W	Serial interface shift register
TMM	DAT	02H	W	Modulo register for 8-bit timer
TMC	DAT	02H	R	Counter register for 8-bit timer
NRZHTMM	DAT	03H	R/W	Modulo register for NRZ low-level period
NRZLTMM	DAT	04H	R/W	Modulo register for NRZ high-level period
ADCR	DAT	05H	R/W	A/D converter internal reference voltage setting register
AR	DAT	40H	R/W	Peripheral address of address register for GET/PUT/PUSH/CALL/BR/MOVT/INC instruction

[MEMO]

Fig. 15-1 Register File (1/2)

Row Address	Cdcolumn Address	0		1		2		3		4		5		6		7	
		Note		Note		Note		Note		Note		Note		Note		Note	
0 (8)	Bit 3		0	0	0	0	WDTRES	0									
	Bit 2			1	0	0	WTMMD	0									
	Bit 1		SP	0	SYSCK	*	WTMRES	0									
	Bit 0			1	XEN	*	0	0									
1 (9)	Bit 3		0	0	0	0											
	Bit 2		0	0	0	0											
	Bit 1		0	0	0	0											
	Bit 0		NRZBF	0	NRZ	0											
2 (A)	Bit 3	0	0	VREFEN	0	SIOTS	0	0	0							P0DBIO3	0
	Bit 2	0	0	ADCEN	0	SIOHIZ	0	NRZEN	0							P0DBIO2	0
	Bit 1	0	0	ADCCH1	0	SIOCK1	0	TMOE	0							P0DBIO1	0
	Bit 0	ADCCMP	0	ADCCH0	0	SIOCK0	0	SIOEN	0							P0DBIO0	0
3 (B)	Bit 3		LCDEN	0	LCDMD3	0	TMEN	1								P1AGIO	0
	Bit 2		LCDCK2	0	LCDMD2	0	TMRES	0								P0CGIO	0
	Bit 1		LCDCK1	0	LCDMD1	0	TMCK1	0								P0BGIO	0
	Bit 0		LCDCK0	0	LCDMD0	1	TMCK0	0								P0AGIO	0

**Note** Status when the system is reset.

\*: When the mask option selects the main clock (USEX): 1  
 When the mask option does not select the main clock (NOX): 0

Fig. 15-2 Data Memory Configuration

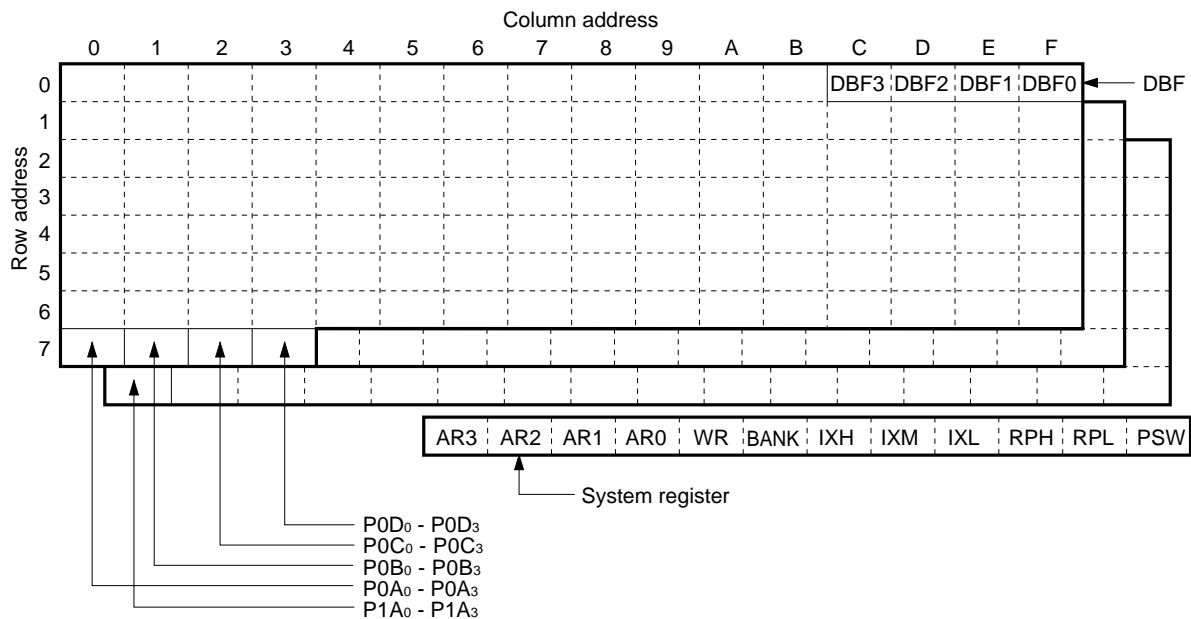


Fig. 15-1 Register File (2/2)

Ccolumn Address		8	9	A	B	C	D	E	F	
		Note	Note	Note	Note	Note	Note	Note	Note	
Row Address	Bit 3								0	0
	Bit 2								0	0
0 (8)	Bit 1								0	0
	Bit 0								INT	P
1 (9)	Bit 3									
	Bit 2									
	Bit 1									
	Bit 0									
2 (A)	Bit 3								IPSIO	0
	Bit 2								IPWTM	0
	Bit 1								IP	0
	Bit 0								IPTM	0
3 (B)	Bit 3				IRQSIO	0	0	0	0	0
	Bit 2				0	0	IRQWTM	0	0	0
	Bit 1				0	0	0	0	IRQ	0
	Bit 0				0	0	0	0	0	IRQTM

**Note** Status when the system is reset.

P: When the INT pin goes high: 1

When the INT pin goes low: 0

**Remark** ( ) indicates an address to be used when the assembler is used.

All the flags of the control registers are registered to the device file as assembler reserved words. It is convenient to use these reserved words when you develop a program.

★

Table 15-3 Peripheral Hardware

Name	Address	Valid Bit	Remarks
SIOSFR	01H	8	Shift register of serial interface
TMC	02H	8	Count register of 8-bit timer
TMM	02H	8	Modulo register of 8-bit timer
NRZLTMM	03H	8	Low-level period setting modulo register for remote controller carrier generator
NRZHTMM	04H	8	High-level period setting modulo register for remote controller carrier generator
ADCR	05H	8	Compare voltage setting register of A/D converter
AR	40H	16	Address register

16. INSTRUCTION SET

16.1 OUTLINE OF INSTRUCTION SETS

b <sub>14</sub> -b <sub>11</sub>		b <sub>15</sub>		0		1	
		BIN	HEX				
0000	0	ADD	r, m	ADD	m, #n4		
0001	1	SUB	r, m	SUB	m, #n4		
0010	2	ADDC	r, m	ADDC	m, #n4		
0011	3	SUBC	r, m	SUBC	m, #n4		
0100	4	AND	r, m	AND	m, #n4		
0101	5	XOR	r, m	XOR	m, #n4		
0110	6	OR	r, m	OR	m, #n4		
0111	7	INC	AR				
		INC	IX				
		MOVT	DBF, @AR				
		BR	@AR				
		CALL	@AR				
		RET					
		RETSK					
		EI					
		DI					
		RETI					
		PUSH	AR				
		POP	AR				
		GET	DBF, p				
		PUT	p. DBF				
PEEK	WR, rf						
POKE	rf, WR						
RORC	r						
STOP	s						
HALT	h						
NOP							
1000	8	LD	r, m	ST	m, r		
1001	9	SKE	m, #n4	SKGE	m, #n4		
1010	A	MOV	@r, m	MOV	m, @r		
1011	B	SKNE	m, #n4	SKLT	m, #n4		
1100	C	BR	addr (Page 0)	CALL	addr (Page 0)		
1101	D	BR	addr (Page 1)	MOV	m, #n4		
1110	E			SKT	m, #n		
1111	F			SKF	m, #n		

## 16.2 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	
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 <sub>R</sub> , m <sub>C</sub>	
m <sub>R</sub>	: Data memory row address (high)	
m <sub>C</sub>	: Data memory column address (low)	
n	: Bit position (4 bits)	
n4	: Immediate data (4 bits)	
PAGE	: Page (Bit 11 of program counter)	
PC	: Program counter	
p	: Peripheral address	
p <sub>H</sub>	: Peripheral address (higher 3 bits)	
p <sub>L</sub>	: Peripheral address (lower 4 bits)	
r	: General register column address	
rf	: Register file address	
rf <sub>R</sub>	: Register file address (higher 3 bits)	
rf <sub>C</sub>	: Register file address (lower 4 bits)	
SP	: Stack pointer	
s	: Stop releasing condition	
WR	: Window register	
(x)	: Contents addressed by x	

16.3 LIST OF INSTRUCTION SETS

Group	Mnemonic	Operand	Operation	Instruction Code			
				OP code	Operand		
Addition	ADD	r, m	$(r) \leftarrow (r) + (m)$	00000	m <sub>R</sub>	mc	r
		m, #n4	$(m) \leftarrow (m) + n4$	10000	m <sub>R</sub>	mc	n4
	ADDC	r, m	$(r) \leftarrow (r) + (m) + CY$	00010	m <sub>R</sub>	mc	r
		m, #n4	$(m) \leftarrow (m) + n4 + CY$	10010	m <sub>R</sub>	mc	n4
	INC	AR	$AR \leftarrow AR + 1$	00111	000	1001	0000
		IX	$IX \leftarrow IX + 1$	00111	000	1000	0000
Subtraction	SUB	r, m	$(r) \leftarrow (r) - (m)$	00001	m <sub>R</sub>	mc	r
		m, #n4	$(m) \leftarrow (m) - n4$	10001	m <sub>R</sub>	mc	n4
	SUBC	r, m	$(r) \leftarrow (r) - (m) - CY$	00011	m <sub>R</sub>	mc	r
		m, #n4	$(m) \leftarrow (m) - n4 - CY$	10011	m <sub>R</sub>	mc	n4
Logical	OR	r, m	$(r) \leftarrow (r) \vee (m)$	00110	m <sub>R</sub>	mc	r
		m, #n4	$(m) \leftarrow (m) \vee n4$	10110	m <sub>R</sub>	mc	n4
	AND	r, m	$(r) \leftarrow (r) \wedge (m)$	00100	m <sub>R</sub>	mc	r
		m, #n4	$(m) \leftarrow (m) \wedge n4$	10100	m <sub>R</sub>	mc	n4
	XOR	r, m	$(r) \leftarrow (r) \nabla (m)$	00101	m <sub>R</sub>	mc	r
		m, #n4	$(m) \leftarrow (m) \nabla n4$	10101	m <sub>R</sub>	mc	n4
Judge	SKT	m, #n	CMP $\leftarrow 0$ , if $(m) \wedge n = n$ , then skip	11110	m <sub>R</sub>	mc	n
	SKF	m, #n	CMP $\leftarrow 0$ , if $(m) \wedge n = 0$ , then skip	11111	m <sub>R</sub>	mc	n
Compare	SKE	m, #n4	(m)-n4, skip if zero	01001	m <sub>R</sub>	mc	n4
	SKNE	m, #n4	(m)-n4, skip if not zero	01011	m <sub>R</sub>	mc	n4
	SKGE	m, #n4	(m)-n4, skip if not borrow	11001	m <sub>R</sub>	mc	n4
	SKLT	m, #n4	(m)-n4, skip if borrow	11011	m <sub>R</sub>	mc	n4
Rotate	RORC	r		00111	000	0111	r
Transfer	LD	r, m	$(r) \leftarrow (m)$	01000	m <sub>R</sub>	mc	r
	ST	m, r	$(m) \leftarrow (r)$	11000	m <sub>R</sub>	mc	r
	MOV	@r, m	if MPE = 1 : (MP, (r)) $\leftarrow$ (m) if MPE = 0 : (BANK, m <sub>R</sub> , (r)) $\leftarrow$ (m)	01010	m <sub>R</sub>	mc	r
		m, @r	if MPE = 1 : (m) $\leftarrow$ (MP, (r)) if MPE = 0 : (m) $\leftarrow$ (BANK, m <sub>R</sub> , (r))	11010	m <sub>R</sub>	mc	r
		m, #n4	$(m) \leftarrow n4$	11101	m <sub>R</sub>	mc	n4
	MOV <sup>Note</sup>	DBF, @AR	SP $\leftarrow$ SP - 1, ASR $\leftarrow$ PC, PC $\leftarrow$ AR DBF $\leftarrow$ (PC), PC $\leftarrow$ ASR, SP $\leftarrow$ SP + 1	00111	000	0001	0000

**Note** Two instruction cycles are necessary only for executing the MOV<sup>Note</sup> instruction.



Group	Mnemonic	Operand	Operation	Instruction Code			
				OP code	Operand		
Transfer	PUSH	AR	$SP \leftarrow SP - 1, ASR \leftarrow AR$	00111	000	1101	0000
	POP	AR	$AR \leftarrow ASR, SP \leftarrow SP + 1$	00111	000	1100	0000
	PEEK	WR, rf	$WR \leftarrow (rf)$	00111	r <sub>R</sub>	0011	r <sub>C</sub>
	POKE	rf, WR	$(rf) \leftarrow WR$	00111	r <sub>R</sub>	0010	r <sub>C</sub>
	GET	DBF, p	$(DBF) \leftarrow (p)$	00111	p <sub>H</sub>	1011	p <sub>L</sub>
	PUT	p, DBF	$(p) \leftarrow (DBF)$	00111	p <sub>H</sub>	1010	p <sub>L</sub>
Branch	BR	addr	$PC_{10-0} \leftarrow addr, PAGE \leftarrow 0$	01100	addr		
			$PC_{10-0} \leftarrow addr, PAGE \leftarrow 1$	01101			
	@AR	$PC \leftarrow AR$	00111	000	0100	0000	
Subroutine	CALL	addr	$SP \leftarrow SP - 1, ASR \leftarrow PC$ $PC_{10-0} \leftarrow addr, PAGE \leftarrow 0$	11100	addr		
			$SP \leftarrow SP - 1, ASR \leftarrow PC$ $PC \leftarrow AR$	00111			
	RET		$PC \leftarrow ASR, SP \leftarrow SP + 1$	00111	000	1110	0000
	RETSK		$PC \leftarrow ASR, SP \leftarrow SP + 1$ and skip	00111	001	1110	0000
	RETI		$PC \leftarrow ASR, INTR \leftarrow INTSK, SP \leftarrow SP + 1$	00111	100	1110	0000
Interrupt	EI		$INTEF \leftarrow 1$	00111	000	1111	0000
	DI		$INTEF \leftarrow 0$	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

★ 16.4 ASSEMBLER (AS17K) EMBEDDED MACROINSTRUCTIONS

Legend

- flag n : FLG symbol
- n : Bit number
- < > : Can be omitted

	Mnemonic	Operand	Operation	n
Embedded macro	SKTn	flag 1, ... flag n	if (flag 1) ~ (flag n) = all "1", then skip	1 ≤ n ≤ 4
	SKFn	flag 1, ... flag n	if (flag 1) ~ (flag n) = all "0", then skip	1 ≤ n ≤ 4
	SETn	flag 1, ... flag n	(flag 1) ~ (flag n) ← 1	1 ≤ n ≤ 4
	CLRn	flag 1, ... flag n	(flag 1) ~ (flag n) ← 0	1 ≤ n ≤ 4
	NOTn	flag 1, ... flag n	if (flag n) = "0", then (flag n) ← 1 if (flag n) = "1", then (flag n) ← 0	1 ≤ n ≤ 4
	INITFLG	<NOT> flag 1, ... <<NOT> flag n>	if description = NOT flag n, then (flag n) ← 0 if description = flag n, then (flag n) ← 1	1 ≤ n ≤ 4
	BANKn		(BANK) ← n	0 ≤ n ≤ 2

17. ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS (T<sub>A</sub> = 25°C)

Parameter	Symbol	Test Condition	Rating	Unit	
Supply voltage	V <sub>DD</sub>		-0.3 to +7.0	V	
Analog supply voltage	V <sub>ADC</sub>		-0.3 to V <sub>DD</sub> +0.3	V	
Input voltage	V <sub>I</sub>		-0.3 to V <sub>DD</sub> +0.3	V	
Output voltage	V <sub>O</sub>		-0.3 to V <sub>DD</sub> +0.3	V	
High-level output current	I <sub>OH</sub>	REM pin	Peak value	-30	mA
			Effective value	-20	mA
		One pin (except REM)	Peak value	-7.5	mA
			Effective value	-5	mA
		All pins (except REM)	Peak value	-22.5	mA
			Effective value	-15	mA
Low-level output current	I <sub>OL</sub>	One pin	Peak value	7.5	mA
			Effective value	5	mA
		All pins (except REM)	Peak value	22.5	mA
			Effective value	15	mA
Operating ambient temperature	T <sub>A</sub>		-20 to +75	°C	
Storage temperature	T <sub>stg</sub>		-40 to +125	°C	

**Note** Effective value = Peak value × √Duty

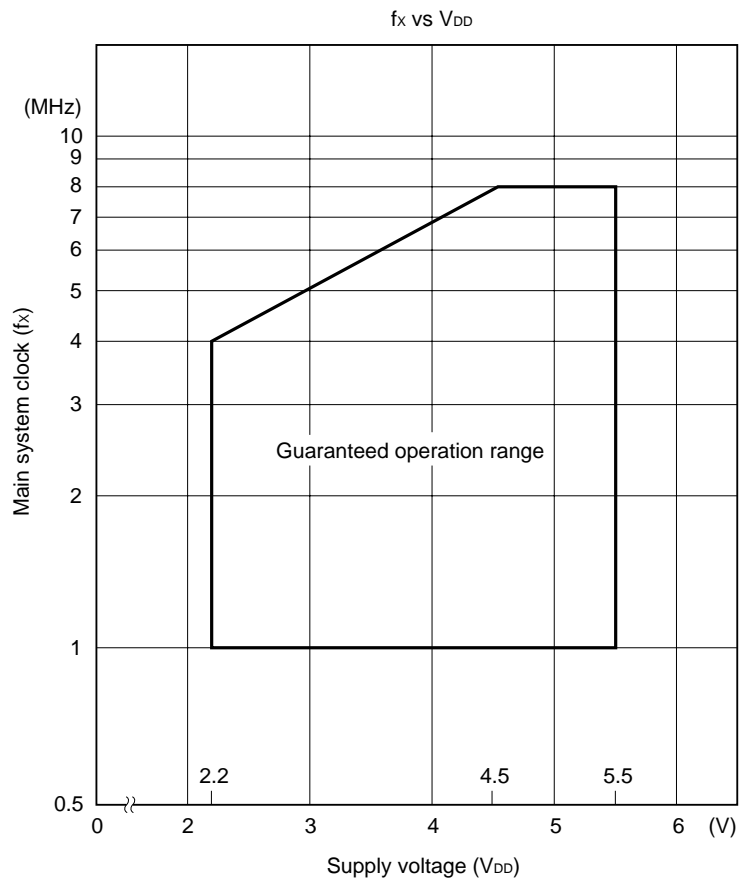
**Caution** Even if one of the parameters exceeds its absolute maximum rating even momentarily, the quality of the product may be degraded. The absolute maximum rating therefore specifies the upper or lower limit of the value at which the product can be used without physical damages. Be sure not to exceed or fall below this value when using the product.

CAPACITANCE (T<sub>A</sub> = 25°C, V<sub>DD</sub> = 0 V)

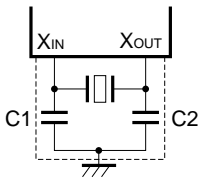
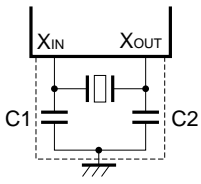
Parameter	Symbol	Test Condition	MIN.	TYP.	MAX.	Unit
Input capacitance	C <sub>IN</sub>	INT, SI, and $\overline{\text{RESET}}$ pins			10	pF

RECOMMENDED OPERATING RANGES (T<sub>A</sub> = -20 to +75°C)

Parameter	Symbol	Test Condition	MIN.	TYP.	MAX.	Unit
Supply voltage	V <sub>DD1</sub>	System clock f <sub>X</sub> = 4 MHz	2.2	3.0	5.5	V
	V <sub>DD2</sub>	System clock f <sub>X</sub> = 8 MHz	4.5	5.0	5.5	V
	V <sub>DD3</sub>	System clock f <sub>XT</sub> = 32.768 kHz	2.0	3.0	5.5	V
Main clock oscillation frequency	f <sub>X</sub>		1.0	4	8.0	MHz
Subclock oscillation frequency	f <sub>XT</sub>			32.768		kHz

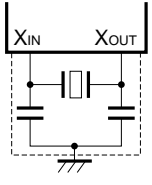


**MAIN SYSTEM CLOCK OSCILLATOR CHARACTERISTICS** (T<sub>A</sub> = -20 to +75°C, V<sub>DD</sub> = 2.2 to 5.5 V)

Resonator Constants	Recommended	Item	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic <sup>Note 3</sup> oscillator		Oscillation frequency (f <sub>X</sub> ) <sup>Note 1</sup>		1.0	4	8.0	MHz
		Oscillation stabilization time <sup>Note 2</sup>	From when V <sub>DD</sub> reaches the minimum oscillation voltage			4	ms
Crystal <sup>Note 3</sup> oscillator		Oscillation frequency (f <sub>X</sub> ) <sup>Note 1</sup>		1.0	4	8.0	MHz
		Oscillation stabilization time <sup>Note 2</sup>	V <sub>DD</sub> = 4.5 to 5.5V			10	ms
						30	ms

- Notes**
1. The oscillation frequency is indicated only to express the oscillator characteristics.
  2. The oscillation stabilization time is the time required for stabilizing the oscillation after V<sub>DD</sub> is applied or the STOP mode is released.
  3. The recommended oscillators are shown in the table described later.

**SUBSYSTEM CLOCK OSCILLATOR CHARACTERISTICS**

Resonator Constants	Recommended	Item	Conditions	MIN.	TYP.	MAX.	Unit
Crystal oscillator		Oscillation frequency (f <sub>Xτ</sub> )			32.768		kHz
		Oscillation stabilization time			5	10	s

**Caution** When using the main system clock and the subsystem clock generators, in order to avoid wiring capacitance effects, the following notations must be read and observed for wiring the portion inside the dotted line in the table:

- Wiring length must be minimized.
- Do not cross with other signal lines. Do not wire close to a large current line.
- Capacitors used in the oscillators must always be grounded to GND potential level. Never ground the grounding pattern having a large current flow.
- Do not take the signal directly out of the oscillator.

In order to reduce the power consumption, the subsystem clock oscillator employs a low amplification factor circuit. Because of this, the subsystem clock oscillator is more sensitive to noise than the main system clock oscillator. Therefore, when using the subsystem clock, wiring must be carefully planned.

**RECOMMENDED OSCILLATORS**

Main System Clock Oscillator (made of ceramic)

Manufacturer	Part Name	External Capacitance (pF)		Oscillation Voltage Range (V)		Remarks
		C1	C2	MIN.	MAX.	
MURATA Mfg.	CSA3.58MG	30	30	2.0	6.0	
	CSA4.00MG	30	30	2.0	6.0	
	CSA4.19MG	30	30	2.0	6.0	
	CST3.58MGW	Not required	Not required	2.0	6.0	Built-in capacitor
	CST4.00MGW	Not required	Not required	2.0	6.0	
	CST4.19MGW	Not required	Not required	2.0	6.0	
KYOCERA	KBR3.58MS	33	33	2.0	6.0	
	KBR4.0MS	33	33	2.0	6.0	
	KBR4.19MS	33	33	2.0	6.0	
TOKO	CRHF4.00	18	18	2.0	6.0	
DAISHINKU	PRS0400BCSAN	39	33	2.0	6.0	

Main System Clock Oscillator (made of crystal)

Manufacturer	Frequency (MHz)	Holder	External Capacitance (pF)		Oscillation Voltage Range (V)		Remarks
			C1	C2	MIN.	MAX.	
KINSEKI	4.0	HC-49U-S	22	22	2.0	6.0	

**DC CHARACTERISTICS** (T<sub>A</sub> = -20 to +75°C, V<sub>DD</sub> = V<sub>ADC</sub> = 2.2 to 3.6 V)

Parameter	Symbol	Test Condition	MIN.	TYP.	MAX.	Unit
High-Level Input Voltage	V <sub>IH1</sub>	$\overline{\text{RESET}}$ and INT pins	0.8V <sub>DD</sub>		V <sub>DD</sub>	V
	V <sub>IH2</sub>	Other than $\overline{\text{RESET}}$ and INT pins	0.7V <sub>DD</sub>		V <sub>DD</sub>	V
Low-Level Input Voltage	V <sub>IL1</sub>	$\overline{\text{RESET}}$ and INT pins	0		0.2V <sub>DD</sub>	V
	V <sub>IL2</sub>	Other than $\overline{\text{RESET}}$ and INT pins	0		0.3V <sub>DD</sub>	V
High-Level Input Leakage Current	I <sub>LIH1</sub>	X <sub>TIN</sub> , X <sub>TOUT</sub> , X <sub>IN</sub> , and X <sub>OUT</sub> pins			20	μA
	I <sub>LIH2</sub>	Other than X <sub>TIN</sub> , X <sub>TOUT</sub> , X <sub>IN</sub> , and X <sub>OUT</sub> pins			3	μA
Low-Level Input Leakage Current	I <sub>LIL1</sub>	X <sub>TIN</sub> , X <sub>TOUT</sub> , X <sub>IN</sub> , and X <sub>OUT</sub> pins			-20	μA
	I <sub>LIL2</sub>	Other than X <sub>TIN</sub> , X <sub>TOUT</sub> , X <sub>IN</sub> , and X <sub>OUT</sub> pins			-3	μA
High-Level Output Current	I <sub>OH1</sub>	REM pin V <sub>OH</sub> =V <sub>DD</sub> -1.2 V	-7	-15		mA
	I <sub>OH2</sub>	<b>Note 1</b> V <sub>OH</sub> =V <sub>DD</sub> -0.3 V	-0.3	-0.7		mA
Low-Level Output Current	I <sub>OL</sub>	<b>Note 2</b> V <sub>OL</sub> =0.3 V	0.5	0.9		mA
Built-In Pull-Up Resistor	R <sub>P0A</sub>	P0A <sub>0</sub> to P0A <sub>3</sub> pins	100	200	350	k Ω
	R <sub>RES</sub>	$\overline{\text{RESET}}$ pins (mask option)	24	47	94	k Ω
A/D Absolute Precision					±2	LSB
A/D Resolution				8		bits
A/D Converter Circuit Current	I <sub>ADC</sub>			60	120	μA
Comparator Error		In comparator mode		10	20	mV
Supply Current	I <sub>DD1</sub>	X installed (f <sub>X</sub> =4.19 MHz)				
	I <sub>DD2</sub>	XT not installed				
	I <sub>DD3</sub>	V <sub>DD</sub> =3 V				
	I <sub>DD4</sub>	X not installed or STOP mode				
	I <sub>DD5</sub> <sup>Note 3</sup>	XT installed (f <sub>XT</sub> =32.768 kHz) V <sub>DD</sub> =3 V				
		RUN mode		0.8	2.0	mA
		HALT mode		0.3	1.5	mA
		STOP mode		2.0	10	μA
		RUN mode		7.0	25	μA
		HALT mode		3.0	15	μA

- Notes**
1. P0A<sub>0</sub> to P0A<sub>3</sub>, P0D<sub>0</sub> to P0D<sub>3</sub>, and P1A<sub>0</sub> to P1A<sub>2</sub> pins
  2. P0A<sub>0</sub> to P0A<sub>3</sub>, P0B<sub>0</sub> to P0B<sub>3</sub>, P0C<sub>0</sub> to P0C<sub>3</sub>, P0D<sub>0</sub> to P0D<sub>3</sub>, P1A<sub>0</sub> to P1A<sub>2</sub>,  $\overline{\text{WDOUT}}$ , and REM pins
  3. The specifications of the main STOP mode (sub-mounting) are the same as the sub-HALT mode (with the main clock oscillation stopped).

**LCD CHARACTERISTICS** (T<sub>A</sub> = -20 to +75°C, V<sub>DD</sub> = 2.2 to 3.6 V)

Parameter	Symbol	Test Condition	MIN.	TYP.	MAX.	Unit
V <sub>LCDC</sub> Output Voltage	V <sub>LCDC</sub>	V <sub>DD</sub> = 3 V, T <sub>a</sub> = 25°C, R <sub>1</sub> = R <sub>2</sub> = 1 MΩ	0.5	0.6	0.7	V
LCD Reference Output Voltage	V <sub>LCD0</sub>	External variable resistance (0 to 2.2 MΩ)	0.8		1.8	V
Doubler Output Voltage	V <sub>LCD1</sub>	C1 to C4 = 0.47 μF	1.9	2.0		V <sub>LCD0</sub>
Tripler Output Voltage	V <sub>LCD2</sub>	C1 to C4 = 0.47 μF	2.85	3.0		V <sub>LCD0</sub>
LCD Common Output Current	I <sub>COM</sub>	Output voltage deviation = 0.2 V	30			μA
LCD Segment Output Current	I <sub>LCD</sub>	Output voltage deviation = 0.2 V	5			μA

**DC CHARACTERISTICS** (T<sub>A</sub> = -20 to +75°C, V<sub>DD</sub> = V<sub>ADC</sub> = 5 V±10%)

Parameter	Symbol	Test Condition	MIN.	TYP.	MAX.	Unit	
High-Level Input Voltage	V <sub>IH1</sub>	$\overline{\text{RESET}}$ and INT pins	0.8V <sub>DD</sub>		V <sub>DD</sub>	V	
	V <sub>IH2</sub>	Other than $\overline{\text{RESET}}$ and INT pins	0.7V <sub>DD</sub>		V <sub>DD</sub>	V	
Low-Level Input Voltage	V <sub>IL1</sub>	$\overline{\text{RESET}}$ and INT pins	0		0.2V <sub>DD</sub>	V	
	V <sub>IL2</sub>	Other than $\overline{\text{RESET}}$ and INT pins	0		0.3V <sub>DD</sub>	V	
High-Level Input Leakage current	I <sub>LIH1</sub>	X <sub>TIN</sub> , X <sub>TOUT</sub> , X <sub>IN</sub> , and X <sub>OUT</sub> pins			20	μA	
	I <sub>LIH2</sub>	Other than X <sub>TIN</sub> , X <sub>TOUT</sub> , X <sub>IN</sub> , and X <sub>OUT</sub> pins			3	μA	
Low-Level Input Leakage current	I <sub>LIL1</sub>	X <sub>TIN</sub> , X <sub>TOUT</sub> , X <sub>IN</sub> , and X <sub>OUT</sub> pins			-20	μA	
	I <sub>LIL2</sub>	Other than X <sub>TIN</sub> , X <sub>TOUT</sub> , X <sub>IN</sub> , and X <sub>OUT</sub> pins			-3	μA	
High-Level Output Current	I <sub>OH1</sub>	REM pin V <sub>OH</sub> = V <sub>DD</sub> -0.6 V	-7	-15		mA	
	I <sub>OH2</sub>	<b>Note 1</b> V <sub>OH</sub> = V <sub>DD</sub> -0.3 V	-0.8	-1.2		mA	
Low-Level Output Current	I <sub>OL</sub>	<b>Note 2</b> V <sub>OL</sub> = 0.3 V	1.0	1.5		mA	
Built-In Pull-Up Resistor	R <sub>P0A</sub>	P0A <sub>0</sub> to P0A <sub>3</sub> pins	140	200	350	k Ω	
	R <sub>RES</sub>	$\overline{\text{RESET}}$ pins (mask option)	27	47	94	k Ω	
A/D Absolute Precision					±2	LSB	
A/D Resolution				8		bits	
A/D Converter Circuit Current	I <sub>ADC</sub>			60	120	μA	
Comparator Error		In comparator mode		10	20	mV	
Supply Current	I <sub>DD1</sub>	X installed (f <sub>X</sub> =4.19 MHz)					
			RUN mode		1.8	5.0	mA
	I <sub>DD2</sub>	XT not installed					
			HALT mode		0.6	2.0	mA
	I <sub>DD3</sub>	V <sub>DD</sub> = 5 V					
		STOP mode		2.6	20	μA	
I <sub>DD4</sub>	X not installed or STOP mode						
	XT installed						
		RUN mode		10.5	40	μA	
I <sub>DD5</sub>	(f <sub>X</sub> = 32.768 kHz) V <sub>DD</sub> = 5 V						
		HALT mode		6.0	20	μA	

- Notes**
1. P0A<sub>0</sub> to P0A<sub>3</sub>, P0D<sub>0</sub> to P0D<sub>3</sub>, and P1A<sub>0</sub> to P1A<sub>2</sub> pins
  2. P0A<sub>0</sub> to P0A<sub>3</sub>, P0B<sub>0</sub> to P0B<sub>3</sub>, P0C<sub>0</sub> to P0C<sub>3</sub>, P0D<sub>0</sub> to P0D<sub>3</sub>, P1A<sub>0</sub> to P1A<sub>2</sub>, REM, and  $\overline{\text{WDOOUT}}$  pins
  3. The specifications of the main STOP mode (sub-mounting) are the same as the sub-HALT mode (with the main clock oscillation stopped).

**LCD CHARACTERISTICS** (T<sub>A</sub> = -20 to +75°C, V<sub>DD</sub> = 5 V±10%)

Parameter	Symbol	Test Condition	MIN.	TYP.	MAX.	Unit
LCD Reference Output Voltage	V <sub>LCD0</sub>	External variable resistance (0 to 2.2 MΩ)	0.8		1.8	V
Doubler Output Voltage	V <sub>LCD1</sub>	C1 to C4 = 0.47 μF	1.9	2.0		V <sub>LCD0</sub>
Tripler Output Voltage	V <sub>LCD2</sub>	C1 to C4 = 0.47 μF	2.85	3.0		V <sub>LCD0</sub>
LCD Common Output Current	I <sub>COM</sub>	Output voltage deviation = 0.2 V	30			μA
LCD Segment Output Current	I <sub>LCD</sub>	Output voltage deviation = 0.2 V	5			μA



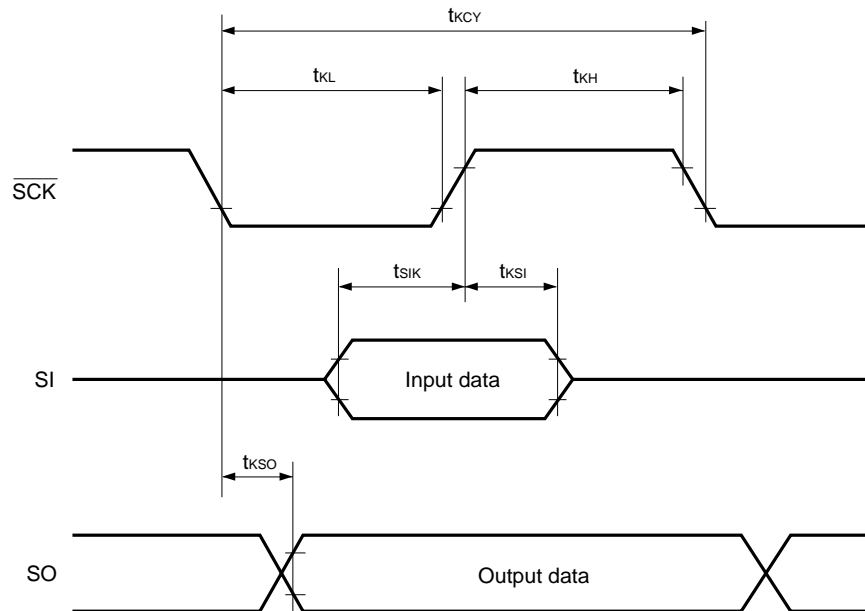
AC CHARACTERISTICS (T<sub>A</sub> = -20 to +75°C, V<sub>DD</sub> = 2.2 to 5.5 V)

Parameter	Symbol	Test Condition	MIN.	TYP.	MAX.	Unit	
$\overline{\text{SCK}}$ Input Cycle Time	t <sub>KCY</sub>	V <sub>DD</sub> = 5 V ± 10 %	Data Input	2.0			μs
			Data Output	10			μs
			Data Input	5			μs
			Data Output	13			μs
$\overline{\text{SCK}}$ Input High- and Low-Level Widths	t <sub>KH</sub> , t <sub>KL</sub>	V <sub>DD</sub> = 5 V ± 10 %	Data Input	1.0			μs
			Data Output	5.0			μs
			Data Input	2.5			μs
			Data Output	6.5			μs
SI Setup Time (vs. $\overline{\text{SCK}}\uparrow$ )	t <sub>SIK</sub>		100			ns	
SI Hold Time (vs. $\overline{\text{SCK}}\uparrow$ )	t <sub>KSI</sub>		100			ns	
$\overline{\text{SCK}}\downarrow \rightarrow$ to SO Output Delay Time	t <sub>KSO</sub>	C <sub>L</sub> = 100 pF			4.5	μs	
INT High- and Low-Level Width	t <sub>IOH</sub> , t <sub>IOL</sub>		50			μs	
$\overline{\text{RESET}}$ Low-Level Width	t <sub>RSL</sub>		50			μs	
P0A Low-Level Width	t <sub>RSLSL</sub>	Standby Release	10			μs	

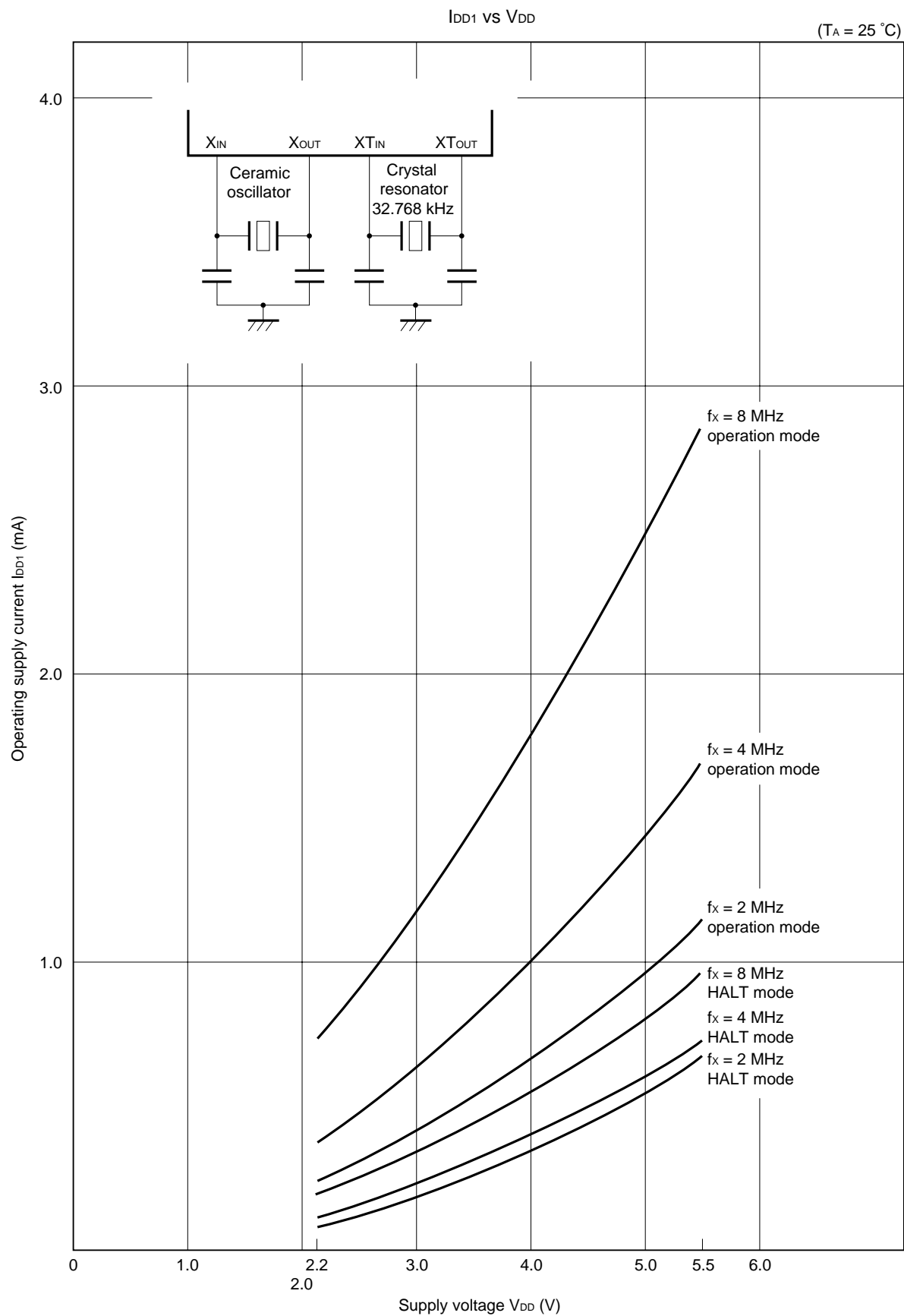
★

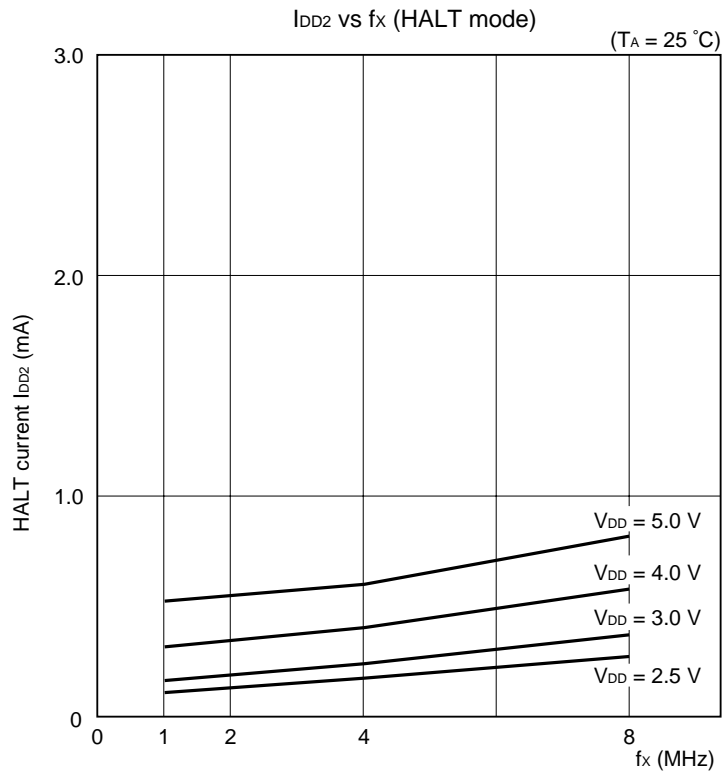
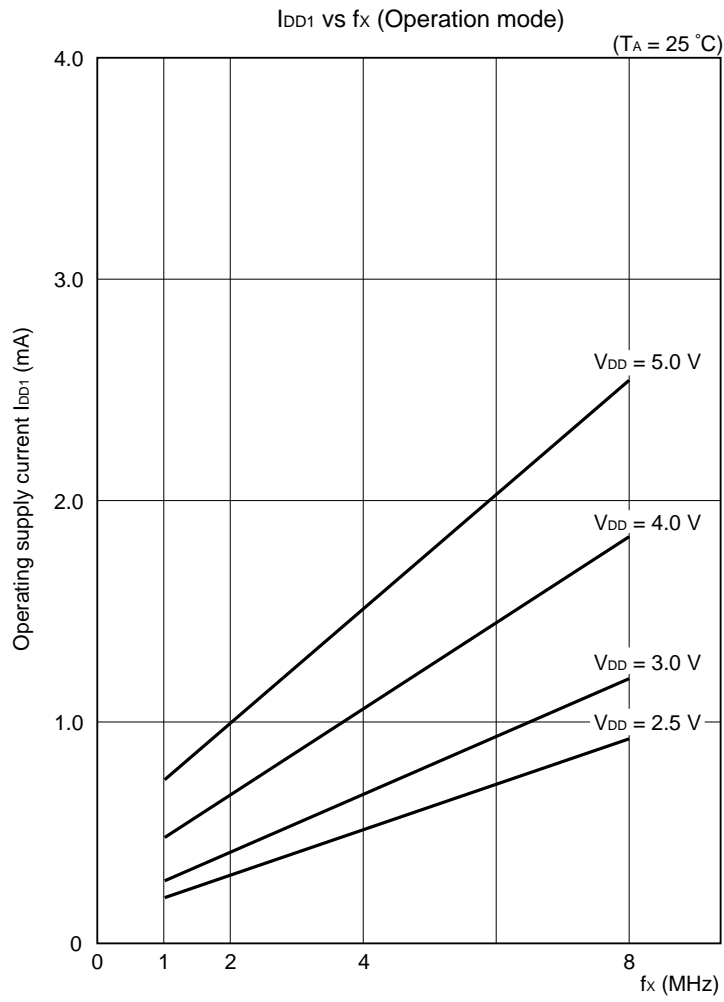
SERIAL TRANSFER TIMING

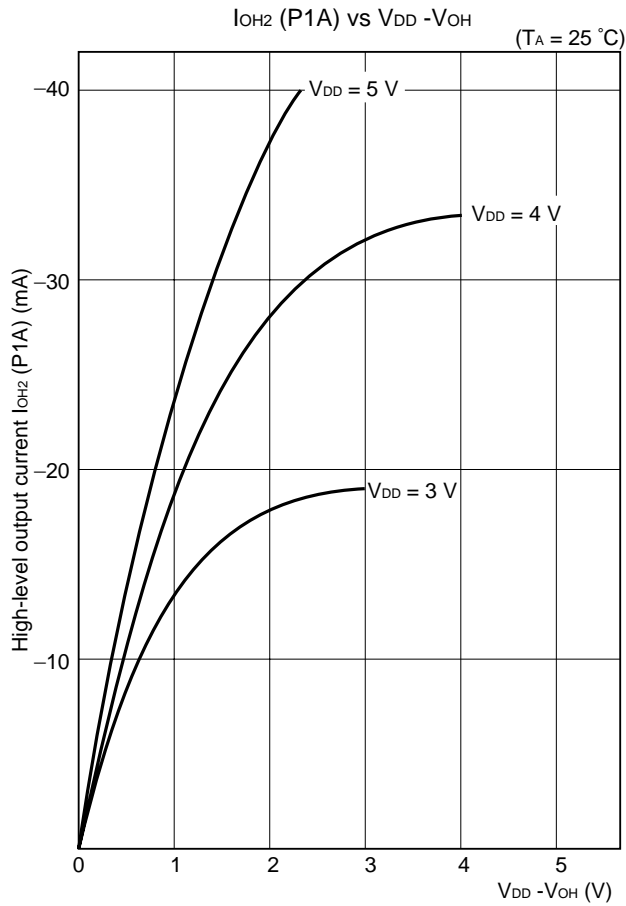
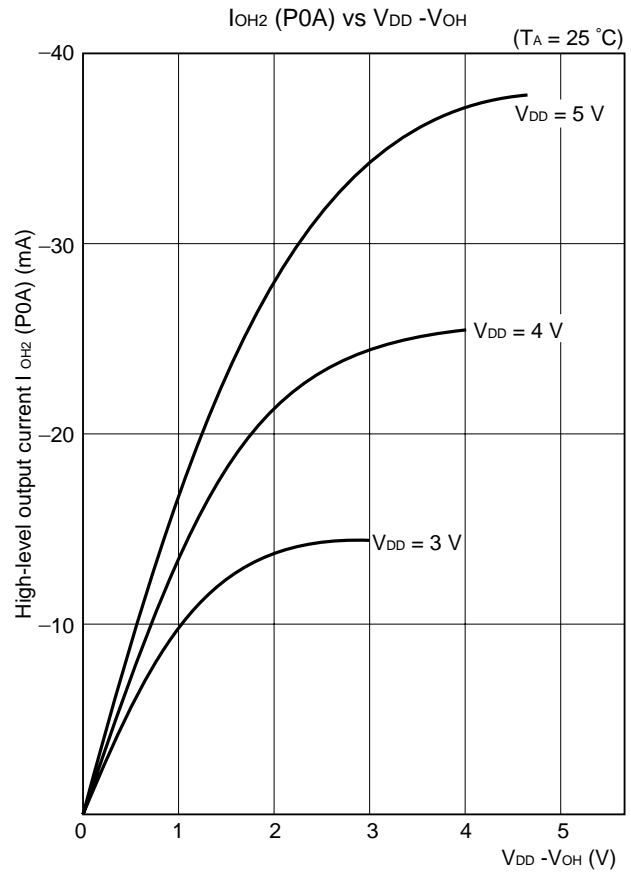
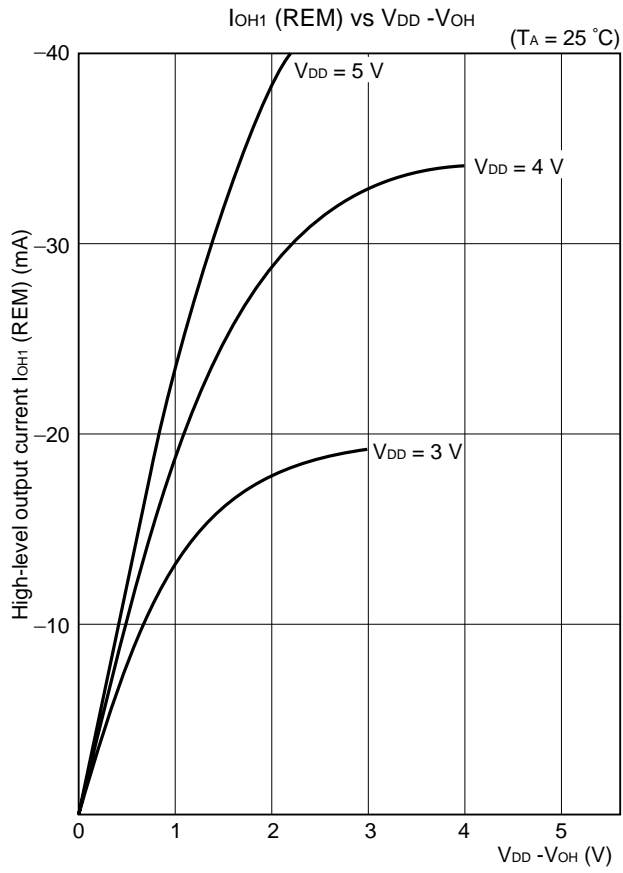
3-line Serial I/O Mode

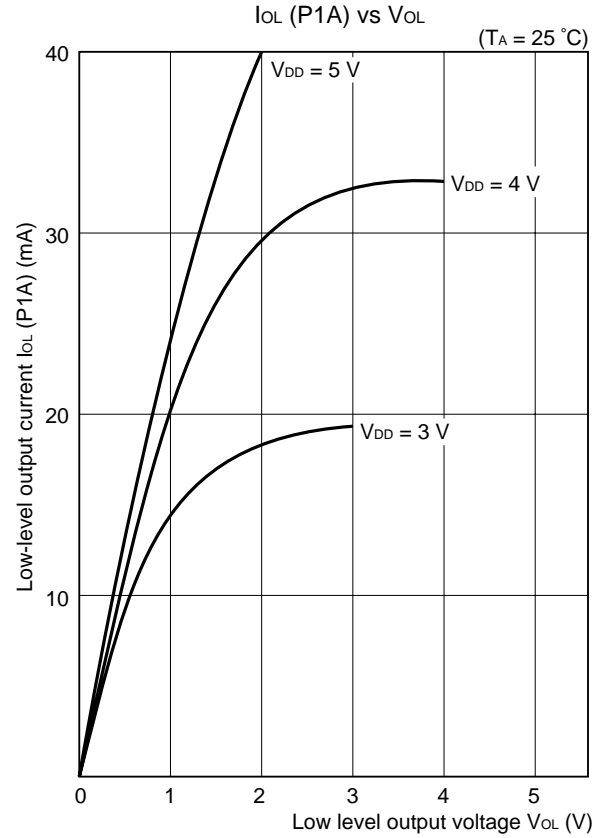
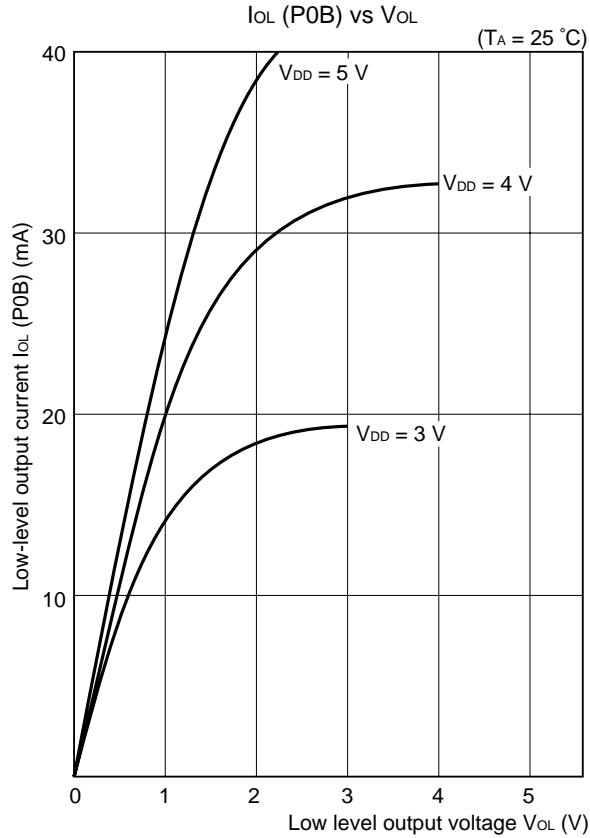
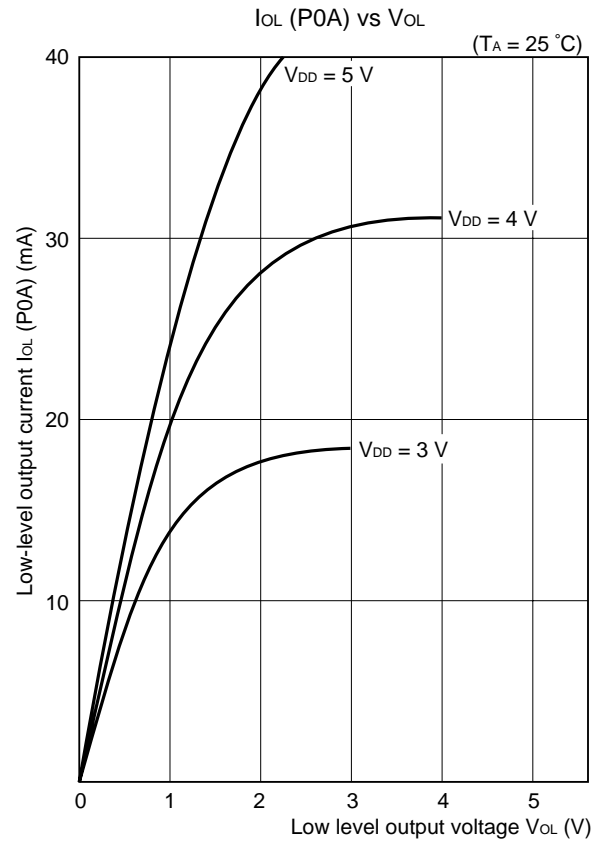
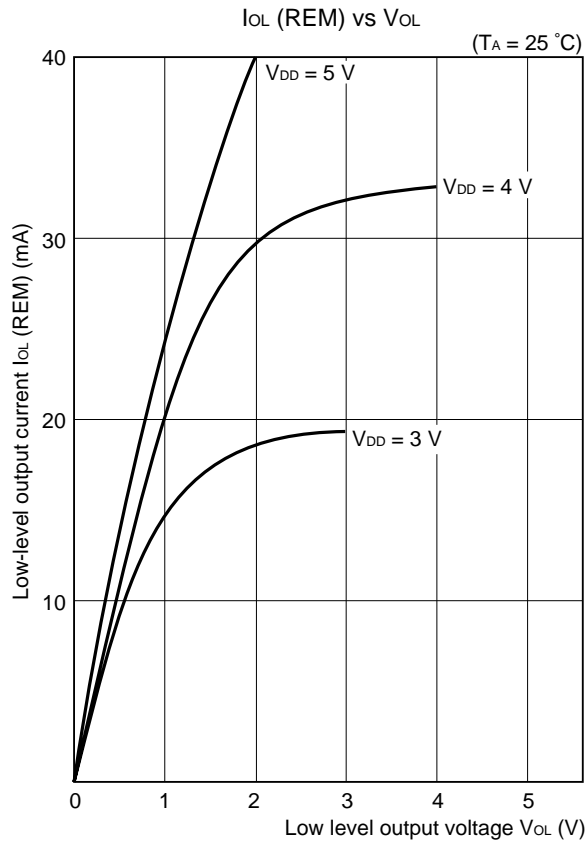


18. PERFORMANCE CURVE (REFERENCE VALUE)



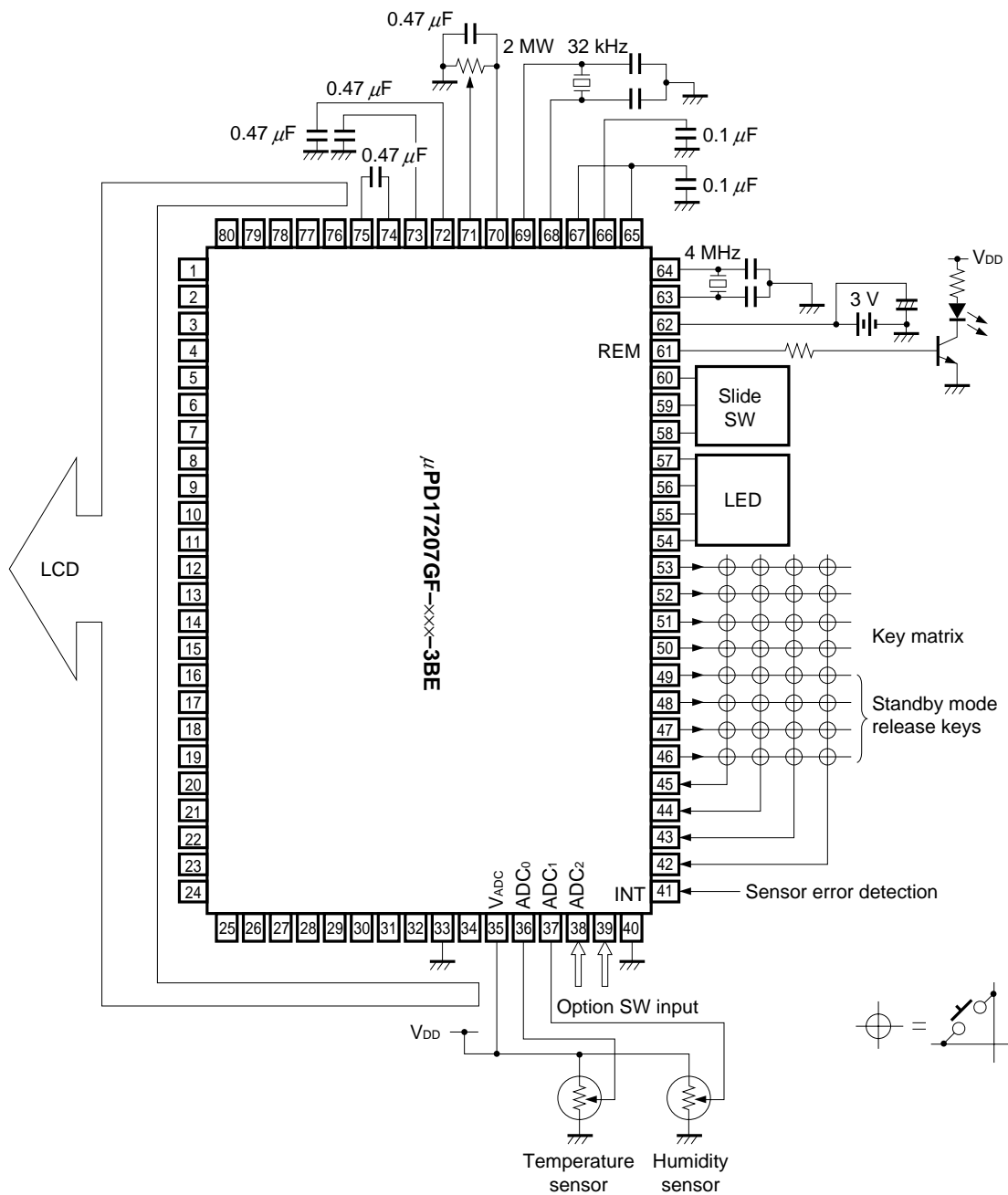






19. EXAMPLE OF APPLICATION CIRCUIT

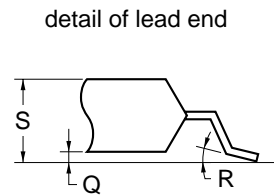
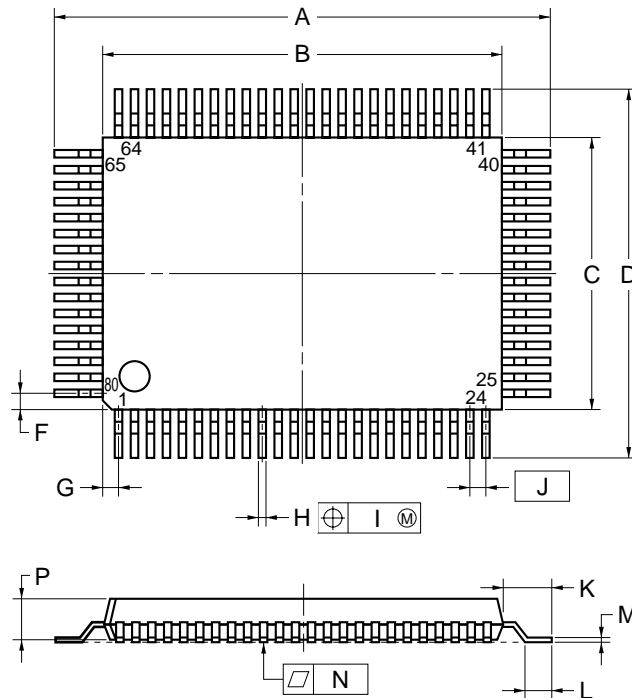
- Remote Controller for Air Conditioner



20. PACKAGE DRAWINGS

PACKAGE DRAWINGS OF MASS-PRODUCTION PRODUCT

80 PIN PLASTIC QFP (14×20)



NOTE

Each lead centerline is located within 0.15 mm (0.006 inch) of its true position (T.P.) at maximum material condition.

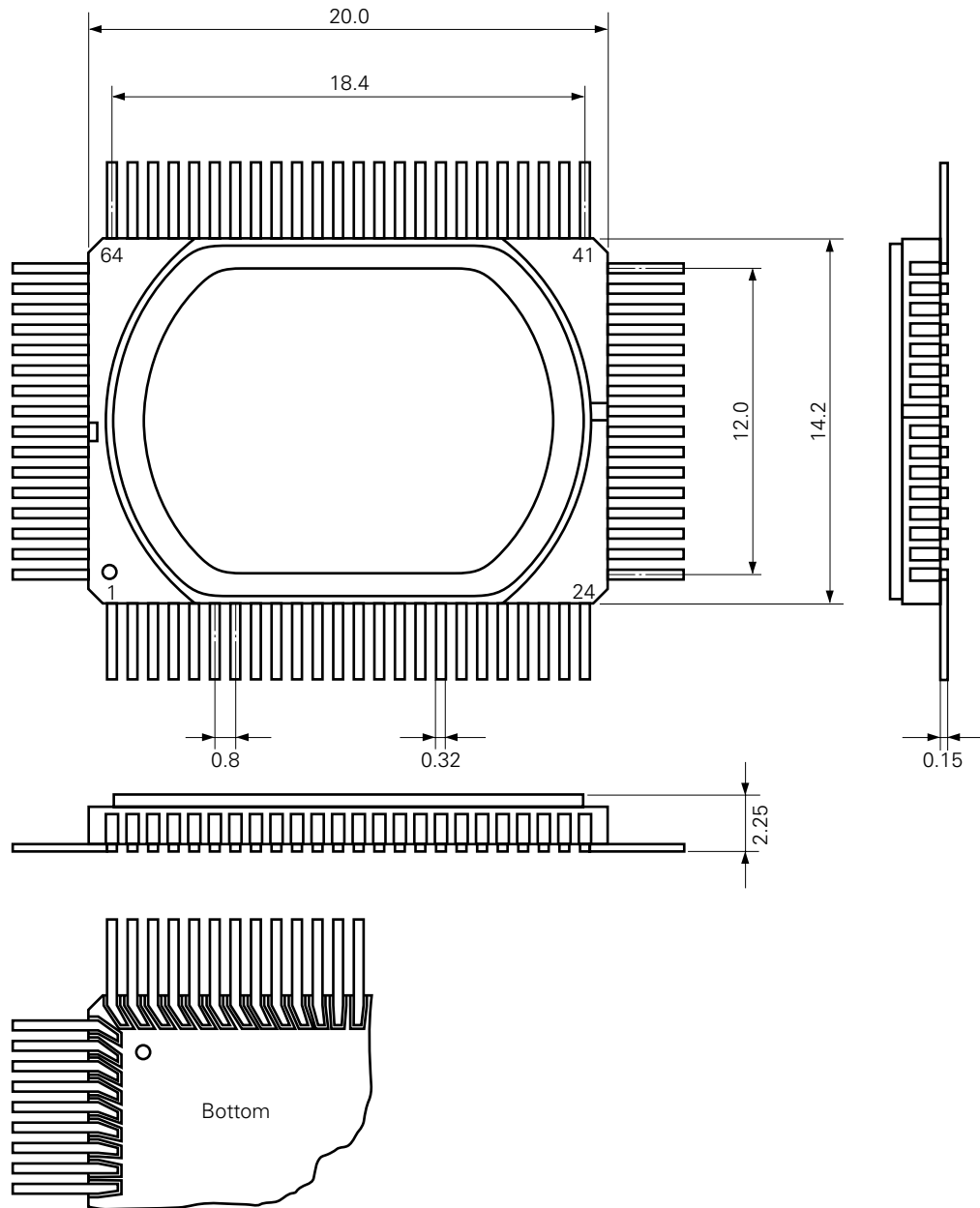
ITEM	MILLIMETERS	INCHES
A	23.2±0.2	0.913 <sup>+0.009</sup> <sub>-0.008</sub>
B	20.0±0.2	0.787 <sup>+0.009</sup> <sub>-0.008</sub>
C	14.0±0.2	0.551 <sup>+0.009</sup> <sub>-0.008</sub>
D	17.2±0.2	0.677±0.008
F	1.0	0.039
G	1.8	0.031
H	0.35±0.10	0.014 <sup>+0.004</sup> <sub>-0.005</sub>
I	0.15	0.006
J	0.8 (T.P.)	0.031 (T.P.)
K	1.6±0.2	0.063±0.008
L	0.8±0.2	0.031 <sup>+0.009</sup> <sub>-0.008</sub>
M	0.15 <sup>+0.10</sup> <sub>-0.05</sub>	0.006 <sup>+0.004</sup> <sub>-0.003</sub>
N	0.10	0.004
P	2.7	0.106
Q	0.125±0.075	0.005±0.003
R	5°±5°	5°±5°
S	3.0 MAX.	0.119 MAX.

S80GF-80-3B9-3

Caution The ES and mass-production products differ in external shape and materials. Please refer to the package drawing for the ES product.

ES PRODUCT PACKAGE DRAWINGS

ES 80-PIN CERAMIC QFP (For Reference) (UNIT: mm)



- Caution**
1. The metal cap is connected to pin 33 and is at the GND level.
  2. Leads on the bottom of the package are guided slantingly.
  3. Package length does not include end flash burr.



**21. RECOMMENDED SOLDERING CONDITIONS**

For the μPD17207, soldering must be performed under the following conditions.

For details of recommended conditions for surface mounting, refer to information document “**Semiconductor Device Mounting Technology Manual**” (C10535E).

For other soldering methods, please consult with NEC personnel.

**Table 21-1 Soldering Conditions of Surface Mount Type**

★

μPD17201AGF-xxx-3B9: 80-pin plastic QFP (14 × 20 mm)

μPD17207GF-xxx-3B9: 80-pin plastic QFP (14 × 20 mm)

Soldering Method	Soldering Conditions	Symbol
Infrared Reflow	Package peak temperature: 235°C, Time: 30 seconds max. (210°C min.), Number of times: 2 max., Days: 7 days <b>Note</b> (after that, prebaking is necessary for 20 hours at 125°C) <Caution> Cannot be baked while packed in anything other than a heat-resistant tray (i.e. magazine, taping or non-heat resistant tray).	IR35-207-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (200°C min.), Number of times: 2 max., Days: 7 days <b>Note</b> (after that, prebaking is necessary for 20 hours at 125°C) <Caution> Cannot be baked while packed in anything other than a heat-resistant tray (i.e. magazine, taping or non-heat resistant tray).	VP15-207-2
Wave Soldering	Solder bath temperature: 260°C max., Time: 10 seconds max. Number of times: 1, Pre-heating temperature: 120°C max. (package surface temperature), Days: 7 days <b>Note</b> (after that, prebaking is necessary for 20 hours at 125°C)	WS60-207-1
Partial Heating	Pin temperature: 300°C max., Time: 3 seconds max. (per side of device)	—

**Note** The number of days the device can be stored after the dry pack was opened, under storage conditions of 25°C and 65% RH max.

**Caution** Do not use two or more soldering methods in combination (except the partial heating method).

**APPENDIX A. DIFFERENCES BETWEEN μPD17P207 AND μPD17201A/17207**

The μPD17P207 has a PROM to which the user can write a program in place of the internal mask ROM of the μPD17201A and 17207. Therefore, the μPD17P207 is identical to μPD17201A and 17207 except for the program memory and mask option. However, **some of the electrical characteristics, such as supply current and V<sub>LCD0</sub> voltage of the μPD17P207, are different from that of the μPD17201A and 17207.**

The following table lists the differences between the μPD17P207 and μPD17201A/17207.

Item	μPD17P207 -001	μPD17P207 -002	μPD17P207 -003	μPD17201A	μPD17207
Product Name					
Program Memory	One-Time PROM			Mask ROM	
	0000H-0FFFH			0000H-0BFFH	0000H-0FFFH
	4096 × 16 bits			3072 × 16 bits	4096 × 16 bits
Pull-Up Resistor of RESET Pin	Provided	Not provided	Not provided	Any (mask option)	
Main Clock Oscillator Circuit		Provided			
Subclock Oscillator Circuit		Not provided	Provided		
V <sub>PP</sub> Pin, PROM Program Pin	Provided			Not provided	
Supply Voltage (T <sub>A</sub> = -20 to +75°C)	V <sub>DD</sub> = 2.5 to 5.5 V (at f <sub>x</sub> = 4 MHz) V <sub>DD</sub> = 2.4 to 5.5 V (at f <sub>x</sub> = 4 MHz, T <sub>A</sub> = -20 to +60°C)			V <sub>DD</sub> = 2.2 to 5.5 V (at f <sub>x</sub> = 4 MHz)	
Package	80-pin plastic QFP (14 × 20 mm)				

**Caution** When using the μPD17P207-001, be sure to connect an oscillator to both the main clock oscillation circuit and subclock oscillation circuit.

APPENDIX B. FUNCTIONAL COMPARISON OF μPD17201A/17207 RELATED PRODUCTS

Item		Product Name					
		μPD17201A	μPD17207	μPD17215	μPD17216	μPD17217	μPD17218
ROM Capacity (bits)		3072 × 16	4096 × 16	2048 × 16	4096 × 16	6144 × 16	8192 × 16
RAM Capacity (bits)		336 × 4		111 × 4		233 × 4	
LCD Controller/Drive		136 segments max.		Not provided			
Infrared Remote Controller Carrier Generator (REM)		LED output is high-active		Internal (no LED output)			
I/O Ports		19 lines		20 lines			
External Interrupt (INT)		1 line (rising-edge detection)		1 line (rising-edge, falling-edge detection)			
Analog Input		4 channels (8-bit A/D)		Not provided			
Timer		2 channels { 8-bit timer Watch timer		2 channels { 8-bit timer Basic interval timer			
Watchdog Timer		Internal ( $\overline{\text{WDOUT}}$ output)					
Low-Voltage Detection Circuit		Not provided		Internal ( $\overline{\text{WDOUT}}$ output)			
Serial Interface		1 channel		Not provided			
Stack		5 levels (3 levels for multiplexed interrupt)					
Instruction Execution Time	Main System Clock	4 μs (4 MHz: with ceramic or crystal oscillator)		<ul style="list-style-type: none"> <li>• 2 μs (8 MHz ceramic oscillator: in high-speed mode)</li> <li>• 4 μs (4 MHz ceramic oscillator: in high-speed mode)</li> <li>• 16 μs (1 MHz ceramic oscillator: in high-speed mode)</li> </ul>			
	Subsystem Clock	488 μs (32.768 kHz: with crystal oscillator)		Not provided			
Supply Voltage	Main System Clock	2.2 to 5.5 V (at f <sub>x</sub> = 4 MHz)		2.2 to 5.5 V (at f <sub>x</sub> = 4 MHz, in high speed)			
	Subsystem Clock	2.0 to 5.5 V (at f <sub>XT</sub> = 32.768 kHz)		Not provided			
Standby Function		STOP, HALT					
Package		80-pin plastic QFP		28-pin plastic SOP 28-pin plastic shrink DIP			
One-Time PROM Product		μPD17P207		μPD17P218			

**Caution** The electrical characteristics differ between the mask ROM model and one-time PROM model.

APPENDIX C. DEVELOPMENT TOOLS

The following tools are available for development of μPD17207 programs:

Hardware

Name	Outline
In-circuit Emulators ( IE-17K IE-17K-ET <sup>Note 1</sup> EMU-17K <sup>Note 2</sup> )	The IE-17K, IE-17K-ET, and EMU-17K are in-circuit emulators that can be commonly used with the 17K series products. The IE-17K and IE-17K-ET are connected to the host machine, which is a PC-9800 series product or IBM PC/AT™, via RS-232-C. The EMU-17K is inserted into an expansion slot of a PC-9800 series product. When these in-circuit emulators are used in combination with a system evaluation board (SE board) dedicated to each model of the device, they operate as the emulator dedicated to that model. A more sophisticated debugging environment can be created by using the man-machine interface software, <i>SIMPLEHOST</i> ™. The EMU-17K has a function that allows you to check the contents of the data memory real-time.
SE Board (SE-17207)	The SE-17207 is an SE board for the μPD17201A, 17207, and 17P207. It may be used alone to evaluate a system, or in combination with an in-circuit emulator for debugging.
Emulation Probe (EP-17201GF)	The EP-17201GF is an emulation probe for the μPD17201A, 17207, and 17P207. It connects an SE board and the target system.
Conversion Socket (EV-9200G-80 <sup>Note 3</sup> )	The EV-9200G-80 is a socket for an 80-pin QFP (14 x 20 mm) and connects the EP-17201GF and the target system.
PROM Programmer (AF-9703 <sup>Note 4</sup> , AF-9704 <sup>Note 4</sup> , AF-9705 <sup>Note 4</sup> , AF-9706 <sup>Note 4</sup> )	The AF9703, AF9704, AF9705, and AF9706 are PROM programmers that can program the μPD17P207. When connected with programmer adapter AF-9808A, this PROM programmer can program the μPD17P207.
Program Adapter (AF-9808A <sup>Note 4</sup> )	The AF-9808A is an adapter for programming the μPD17P207 and is used in combination with the AF-9703, AF-9704, AF-9705 and AF-9706.

- Notes**
1. Low-cost model: external power supply type
  2. This is a product from I.C Corp. For details, consult I.C Corp.
  3. Two EV-9200G-80s are supplied with the EP-17201GF. Five EV-9200G-80s are optionally available as a set.
  4. These are products from Ando Electric Co., Ltd. For details, consult Ando Electric.

Software

Name	Outline Machine	Host	OS Media		Supply	Order Code
17K Series Assembler (AS17K)	AS17K is an assembler in common with the 17K series products. When developing the program of the μPD17201A and the μPD17207, AS17K is used in combination with a device file (AS17201A, AS17207).	PC-9800 series	MS-DOS™		5" 2HD	μS5A10AS17K
					3.5" 2HD	μS5A13AS17K
		IBM PC/AT	PC DOS™		5" 2HC	μS7B10AS17K
					3.5" 2HC	μS7B13AS17K
Device File (AS17201)	AS17201 is a device file for μPD17201A, and it is used in combination with an assembler for the 17K series (AS17K)	PC-9800 series	MS-DOS		5" 2HD	μS5A10AS17201
					3.5" 2HD	μS5A13AS17201
		IBM PC/AT	PC DOS		5" 2HC	μS7B10AS17201
					3.5" 2HC	μS7B13AS17201
Device File (AS17207)	AS17207 is a device file for μPD17207, and it is used in combination with an assembler for the 17K series (AS17K).	PC-9800 series	MS-DOS		5" 2HD	μS5A10AS17207
					3.5" 2HD	μS5A13AS17207
		IBM PC/AT	PC DOS		5" 2HC	μS7B10AS17207
					3.5" 2HC	μS7B13AS17207
Support Software (SIMPLEHOST)	SIMPLEHOST is a software package that enables man-machine interface on the WINDOWS™ when a program is developed by using an incircuit emulator and a personal computer.	PC-9800 series	MS-DOS	Windows	5" 2HD	μS5A10IE17K
					3.5" 2HD	μS5A13IE17K
		IBM PC/AT	PC DOS		5" 2HC	μS7B10IE17K
					3.5" 2HC	μS7B13IE17K

**Remark** The corresponding OS versions are as follows:

OS	Version
MS-DOS	Ver. 3.30 to Ver. 5.00A <sup>Note</sup>
PC DOS	Ver. 3.1 to Ver. 5.0 <sup>Note</sup>
Windows	Ver. 3.0 to Ver. 3.1

**Note** Ver. 5.00/5.00A of MS-DOS and Ver. 5.0 of PC DOS have a task swap function, but this function cannot be used with this software.

## 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 device 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 product in your application, please contact the NEC 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 Inc. (U.S.)**

Mountain View, California  
Tel: 800-366-9782  
Fax: 800-729-9288

### **NEC Electronics (Germany) GmbH**

Duesseldorf, Germany  
Tel: 0211-65 03 02  
Fax: 0211-65 03 490

### **NEC Electronics (UK) Ltd.**

Milton Keynes, UK  
Tel: 01908-691-133  
Fax: 01908-670-290

### **NEC Electronics Italiana s.r.l.**

Milano, Italy  
Tel: 02-66 75 41  
Fax: 02-66 75 42 99

### **NEC Electronics (Germany) GmbH**

Benelux Office  
Eindhoven, The Netherlands  
Tel: 040-2445845  
Fax: 040-2444580

### **NEC Electronics (France) S.A.**

France  
Tel: 01-30-67 58 00  
Fax: 01-30-67 58 99

### **NEC Electronics (France) S.A.**

Spain Office  
Madrid, Spain  
Tel: 01-504-2787  
Fax: 01-504-2860

### **NEC Electronics (Germany) GmbH**

Scandinavia Office  
Taeby Sweden  
Tel: 8-63 80 820  
Fax: 8-63 80 388

### **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 Singapore Pte. Ltd.**

United Square, Singapore 1130  
Tel: 253-8311  
Fax: 250-3583

### **NEC Electronics Taiwan Ltd.**

Taipei, Taiwan  
Tel: 02-719-2377  
Fax: 02-719-5951

### **NEC do Brasil S.A.**

Sao Paulo-SP, Brasil  
Tel: 011-889-1680  
Fax: 011-889-1689

**SIMPLEHOST is a trademark of NEC Corporation**

**MS-DOS and Windows are trademarks of Microsoft Corporation.**

**PC/AT and PC DOS are trademarks of IBM Corporation.**

The application circuits and their parameters are for reference only and are not intended for use in actual design-ins.

The export of this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.

NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.

While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.

NEC devices are classified into the following three quality grades:

"Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots

Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)

Specific: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

Anti-radioactive design is not implemented in this product.