NEC

MOS INTEGRATED CIRCUIT

μPD17203A, 17204

4-BIT SINGLE-CHIP MICROCONTROLLER WITH LARGE-CAPACITANCE STATIC RAM AND 3-CHANNEL TIMER FOR INFRARED REMOTE CONTROLLER

DESCRIPTION

The μPD17203A and 17204 are 4-bit single-chip microcontrollers for infrared remote controller. Integrated on the same die are large capacitance static RAM (XRAM), a 3-channel timer, a carrier generating circuit for remote controller, an amplifier for the remote control receive signal, and a waveform shaping circuit.

The μPD17203A and 17204 employ 17K architecture of general-purpose registers for its CPU, and can directly execute operations between data memories, instead of going through an accumulator. In addition, to enhance programming efficiency all the instructions are 16-bit 1-word instructions.

The μPD17203A and 17204 are also equipped with a model, μPD17203A and 17P204 PROM, to which a program can be written only once, which can be used for evaluation of the μPD17203A and 17204 program or for small-scale production applications.

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

μPD172×× Series User's Manual: IEU-1317

FEATURES

- Infrared remote controller carrier generator circuit (REM output)
- Infrared remote controller reception signal amplifier
- Infrared remote controller reception signal waveform shaping circuit
- 17K architecture: General-purpose register format

	μPD17203A	μPD17204
Program memory (ROM)	8K bytes (4096 × 16 bits)	16K bytes (7936 × 16 bits)
Data memory (RAM)	336 × 4 bits	
Static RAM (XRAM)	4096 × 4 bits	2048 × 4 bits

★

- Instruction execution time : 4 μs (main clock: 4 MHz)
  488 μs (subclock: 32.768 kHz) ★
- 8-bit timer/counter : 1 channel
- 10-bit timer : 1 channel
- 16-bit timer : 1 channel
- Watch timer/watchdog timer: 1 channel ($\overline{\text{WDOUT}}$ output)
- Three-line serial interface : 1 channel
- External interrupt pin : 1 (INT)
- I/O pins : 29 (including INT)
- Supply voltage : V_{DD} = 2.2 to 5.5 V (main clock: 4 MHz)
  V_{DD} = 2.0 to 5.5 V (subclock: 32.768 kHz)

Unless otherwise specified, the μPD17203A is treated as the representative model throughout this documents.

Document No. U10334EJ3V1DS00 (3rd edition)
(Previous No. IC-2866)
Date Published August 1997 N
Printed in Japan

The mark ★ shows major revised points.

★ APPLICATION

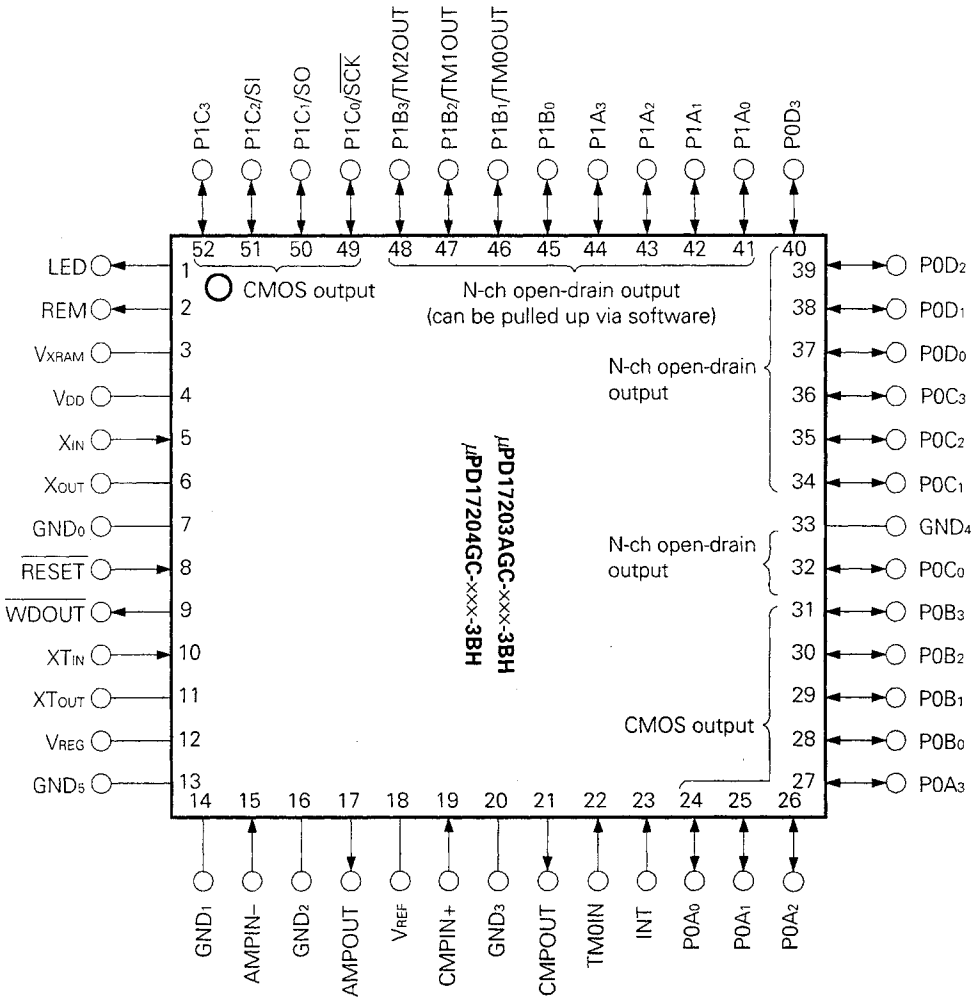
Inflared remote controller for learning, etc.

★ ORDERING INFORMATION

Part Number	Package
μPD17203AGC-xxx-3BH	52-pin plastic QFP (14 × 14 mm)
μPD17204GC-xxx-3BH	52-pin plastic QFP (14 × 14 mm)

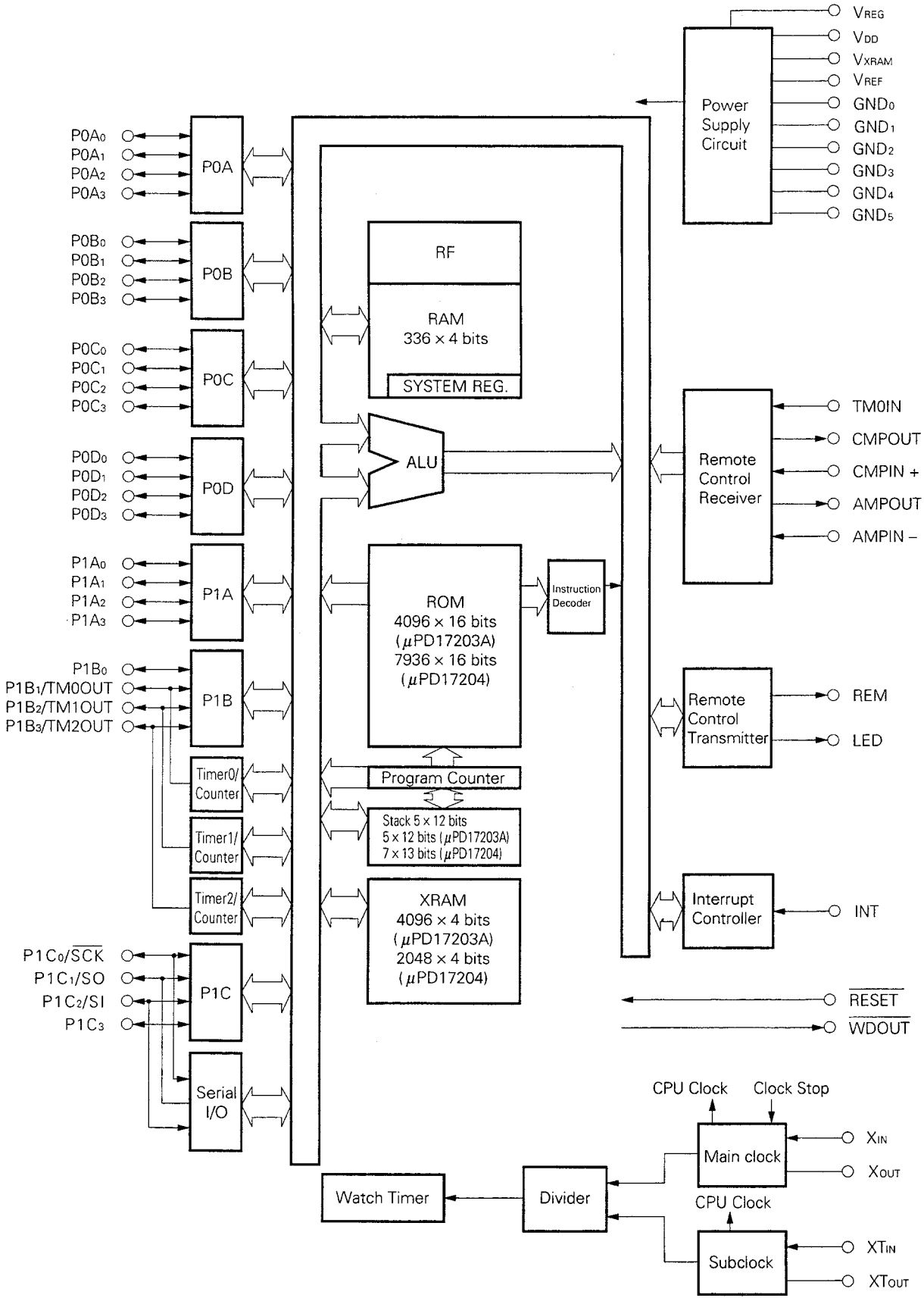
Remark xxx is a ROM code number.

PIN CONFIGURATION (TOP VIEW)



AMPIN−	:	Operational amplifier input
AMPOUT	:	Operational amplifier output
CMPIN+	:	Comparator input
CMPOUT	:	Comparator output
GND_0-GND_5	:	Ground
INT	:	External interrupt input
LED	:	Remote controller transmission output indicator
$P0A_0$-$P0A_3$	:	I/O port 0A
$P0B_0$-$P0B_3$	:	I/O port 0B
$P0C_0$-$P0C_3$	:	I/O port 0C
$P0D_0$-$P0D_3$	:	I/O port 0D
$P1A_0$-$P1A_3$	:	I/O port 1A
$P1B_0$-$P1B_3$	:	I/O port 1B
$P1C_0$-$P1C_3$	:	I/O port 1C
REM	:	Remote controller transmission output
$\overline{RESET}$	:	Reset input
$\overline{SCK}$	:	Serial clock input/output
SI	:	Serial data input
SO	:	Serial data output
TM0IN	:	Timer 0 input
TM0OUT	:	Timer 0 output
TM1OUT	:	Timer 1 output
TM2OUT	:	Timer 2 output
V_{DD}	:	Power supply
V_{REG}	:	Voltage regulator output
V_{REF}	:	Reference voltage output
V_{XRAM}	:	Static RAM (XRAM) power supply
$\overline{WDOUT}$	:	Hang-up detection output
X_{IN}, X_{OUT}	:	Main clock oscillation use
XT_{IN}, XT_{OUT}	:	Subclock oscillation use

BLOCK DIAGRAM



CONTENTS

1. PIN FUNCTIONS

1.1 PIN IDENTIFICATION

Pin No	Symbol	Function	Output Format	At Reset
1	LED	Outputs NRZ signal in synchronization with the infrared remote control signal. Goes low when the remote control carrier is output.	CMOS push-pull	High-level output
2	REM	Outputs an infrared remote control signal (active-high).	CMOS push-pull	Low-level output
3	V_{XRAM}	XRAM power supply. Connect this to V_{DD}.	–	–
4	V_{DD}	Positive power supply.	–	–
5 6	X_{IN} X_{OUT}	Connect a 4-MHz ceramic oscillator for main clock oscillation.	–	(Oscillation stop)
7	GND_0	Ground.	–	–
8	$\overline{RESET}$	Input pin for system reset. The system is reset when a low-level signal is input. Main clock oscillation is halted during low-level input. A pull-up resistor can be provided by the mask option.	–	–
9	$\overline{WDOUT}$	Output for detection N-ch 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 $\overline{RESET}$ pin.	N-ch open-drain	High inpedance ★
10 11	XT_{IN} XT_{OUT}	Connect a 32.768-kHz crystal oscillator for subclock.	–	(Oscillation)
12	V_{REG}	Output pin of voltage regulator for subclock oscillation circuit. When a subclock is used, an external 0.1-μF capacitor must be connected.	–	–
13	GND_5	Ground.	–	–
14, 16, 20	GND_1-GND_3	Operational amplifier ground.	–	–
15	AMPIN-	Inverted input of operational amplifier.	–	Input
17	AMPOUT	Output pin for operational amplifier.	–	Output
18	V_{REF}	Outputs reference voltage. (1/2 V_{DD}). To use this pin, an external 0.1-μF capacitor must be connected.	–	–
19	CMPIN+	Non-inverted input pin for comparator. The comparator output is obtained by CMPOUT.	–	Input
21	CMPOUT	Output pin for comparator. To use a learning remote controller, CMPOUT and TM0IN must be externally connected.	–	Output

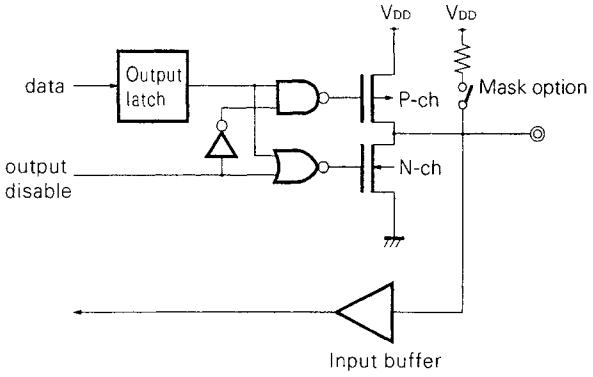
Remark GND_1 to GND_3 are operational amplifier grounds.
To stabilize the operation of these amplifiers, they must be made equipotentional.

Pin No	Symbol	Function	Output Format	At Reset
22	TM0IN	Clock input to timer 0. The input clock is supplied to timer 0 and the envelope signal generation circuit. The frequency of the clock input to TM0IN can be measured by operating this timer in conjunction with timer 1.	–	Input
23	INT	Inputs external interrupt signal.	–	Input
24 | 27	$P0A_0$ | $P0A_3$	4-bit I/O port. Input/output can be set in 4-bit units. Pull-up resistors can be connected by mask option. The standby mode is released when at least one pin of this port goes low.	CMOS push-pull	Input
28 | 31	$P0B_0$ | $P0B_3$	4-bit I/O port. Input/output can be set in 4-bit units. Pull-up resistors can be connected by mask option. The standby mode is released when at least one pin of this port goes low.	CMOS push-pull	Input
32 34 | 36	$P0C_0$ $P0C_1$ | $P0C_3$	4-bit I/O port. Input/output can be set in 4-bit units.	N-ch open-drain	Input
33	GND_4	Ground.	–	–
37 | 40	$P0D_0$ | $P0D_3$	4-bit I/O port. Input/output can be set in 4-bit units.	N-ch open-drain	Input
41 | 44	$P1A_0$ | $P1A_3$	4-bit I/O port. Input/output can be set in bit units. Pull-up resistors can be connected by program.	N-ch open-drain	Input
45 46 47 48	$P1B_0$ $P1B_1$/ TM0OUT $P1B_2$/ TM1OUT $P1B_3$/ TM2OUT	Port 1B and timer output. • $P1B_0$-$P1B_3$ · 4-bit I/O port · Input/output can be set in bit units. · Pull-up resistors can be connected by the program. • TM0OUT-TM2OUT · Timer outputs	N-ch open-drain	Input ($P1B_0$-$P1B_3$)
49 50 51 52	$P1C_0$/$\overline{SCK}$ $P1C_1$/SO $P1C_2$/SI $P1C_3$	Port 1C and input/output for serial interface • $P1C_0$-$P1C_3$ · 4-bit I/O port · Input/output can be set in bit units. • $\overline{SCK}$, SO, SI · $\overline{SCK}$: Serial clock input/output · SO : Serial data output · SI : Serial data input	CMOS push-pull	Input ($P1C_0$-$P1C_3$)

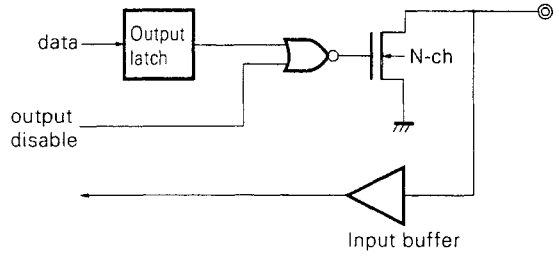
1.2 PIN EQUIVALENT CIRCUITS

This section shows simplified diagrams illustrating the input/output circuits of the μPD17203A pins.

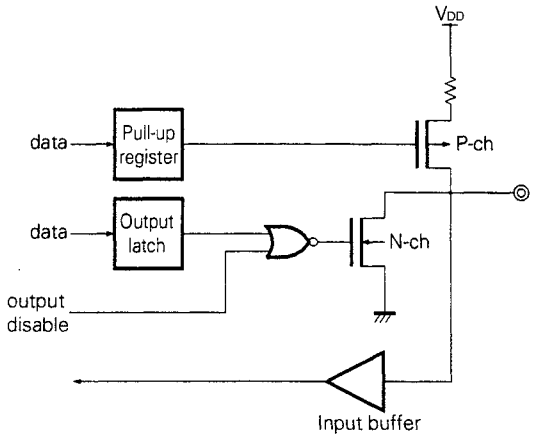
(1) $P0A_0$-$P0A_3$, $P0B_0$-$P0B_3$



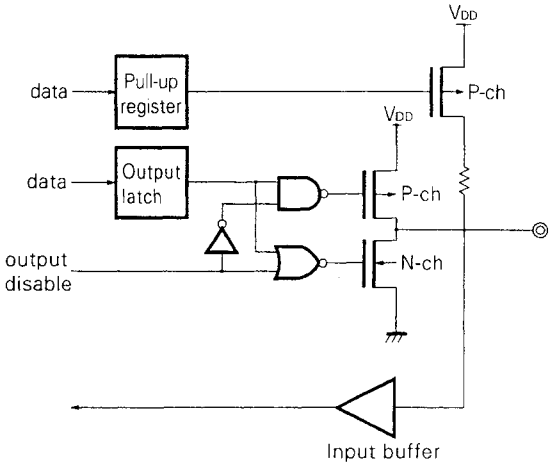
(2) $P0C_0$-$P0C_3$, $P0D_0$-$P0D_3$



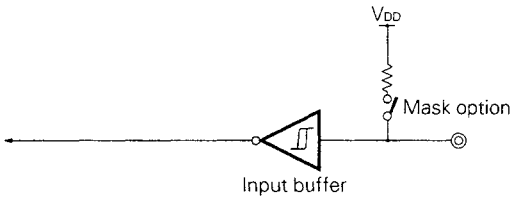
(3) $P1A_0$-$P1A_3$, $P1B_0$-$P1B_3$



(4) $P1C_0$-$P1C_3$

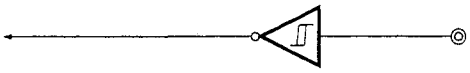


(5) $\overline{\text{RESET}}$



Schmitt triger input with hysteresis characteristics

(6) INT



Schmitt triger input with hysteresis characteristics

1.3 PROCESSING OF UNUSED PINS

Process the unused pins as follows.

Table 1-1. Processing of Unused Pins

PIN	Recommended Connection
INT, TM0IN	Connect to VDD or GND
$P0A_0$-$P0A_3$, $P0B_0$-$P0B_3$	Input : Connect to VDD Output : Open (high-level output)
$P0C_0$-$P0C_3$, $P0D_0$-$P0D_3$ $P1A_0$-$P1A_3$, $P1B_0$-$P1B_3$	Input : Connect to VDD or GND Output : Open (low-level output)
$P1C_0$-$P1C_3$	Input : Connect to VDD or GND Output : Open
LED	Open
REM	Open
$\overline{WDOUT}$	Connect to GND
X_{IN}	
X_{OUT}	Connect to VDD
XT_{IN}	Connect to GND
XT_{OUT}	Connect to VREG
AMPIN–	Connect to GND or AMPOUT
AMPOUT, CMPOUT	Open
CMPIN+	Connect to GND
★ V_{REG}	Open

★ 1.4 NOTES ON USING INT AND $\overline{RESET}$ PINS

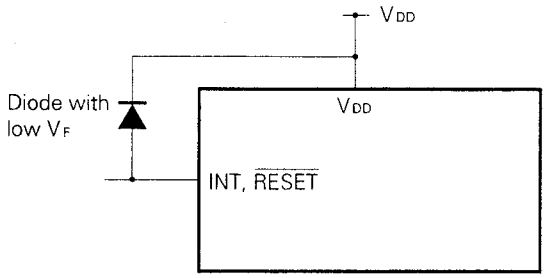
In addition to the functions shown in **1.1 PIN IDENTIFICATION**, the INT and $\overline{RESET}$ pins also have a function to set a test mode (for IC testing) in which the internal operations of the μPD17203A are tested.

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 μPD17203A may be set in the test mode if a noise exceeding V_{DD} is applied.

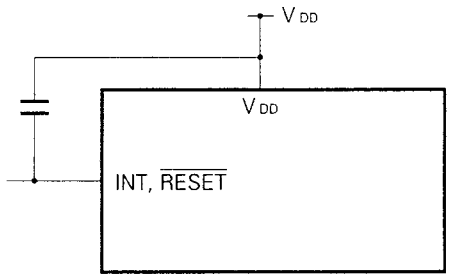
For example, if the wiring length of the INT or $\overline{RESET}$ pin is too long, noise superimposed on the wiring line of the pin may cause the above problem.

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 INT/$\overline{RESET}$ pin



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



2. MEMORY SPACE ★

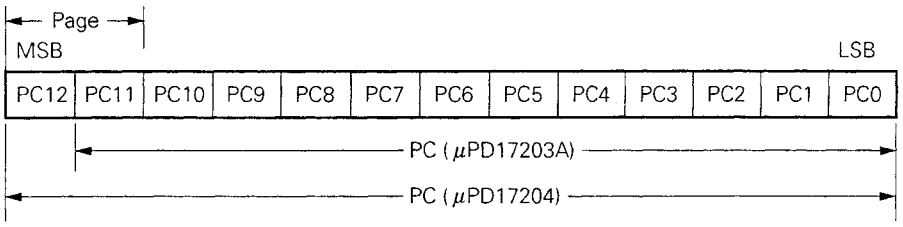
2.1 PROGRAM COUNTER (PC)

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

The program counter is a 12-/13-bit binary counter as shown in Figure 2-1.

Its contents are initialized to address 0000H at reset.

Figure 2-1. Configuration of Program Counter



2.2 PROGRAM MEMORY (ROM)

The configuration of the program memory of the μPD17203A/17204 is as follows:

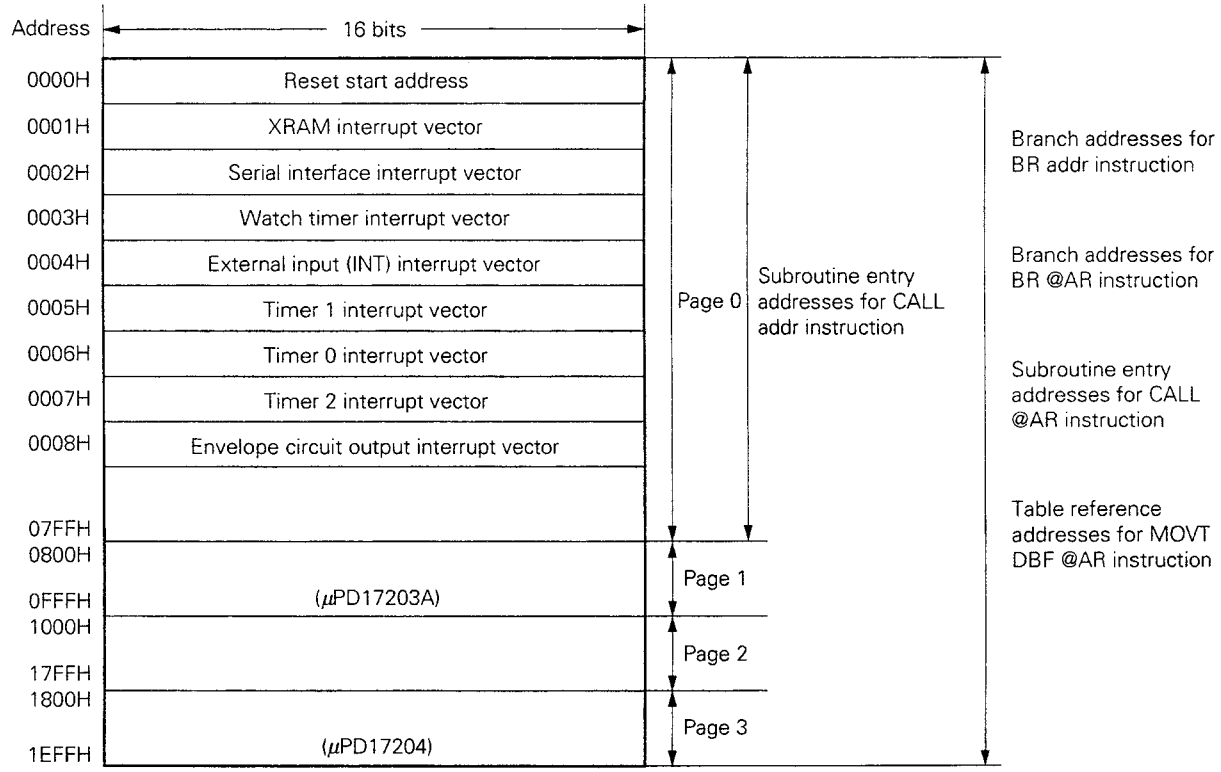
Part Number	Capacity	Address
μPD17203A	4096 × 16 bits	0000H-0FFFH
μPD17204	7936 × 16 bits	0000H-0EFFH

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

The program memory is addressed by the program counter.

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

Figure 2-2. Program Memory Map



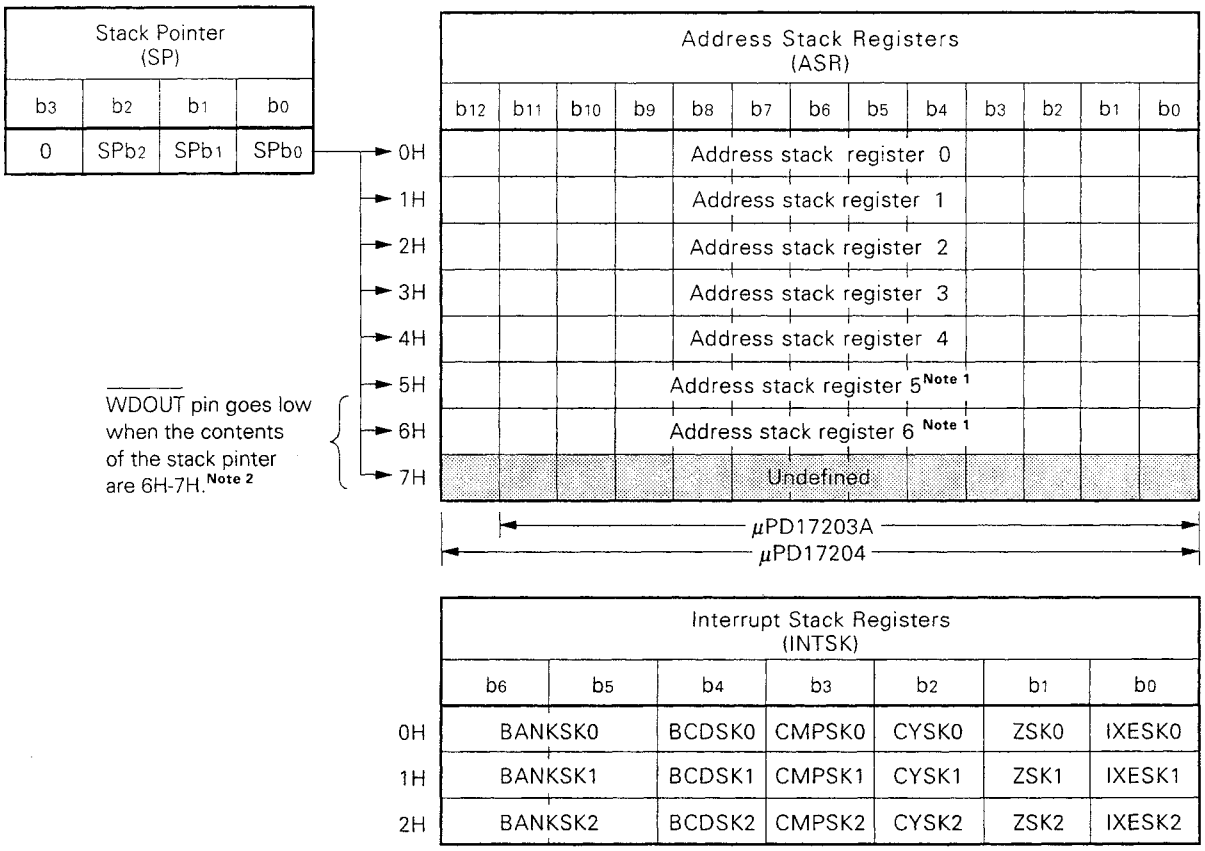
2.3 STACK

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.

2.3.1 Stack Configuration

The stack consists of one 4-bit (higher bit fixed at 0) binary counter stack pointer (SP), five (μPD17203A)/seven (μPD17204) address stack registers, and three 7-bit interrupt stack registers.

Figure 2-3. Stack Configuration



Notes 1. In case of the μPD17203A, it is undefined.

2. The μPD17204: Goes low when exceeded 7H.

2.3.2 Function of Stack

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

The $\overline{\text{WDOUT}}$ pin goes low if a subroutine call or interrupt exceeding 5 levels[Note] 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.**

Note The μPD17204: 7 levels.

2.3.3 Stack Pointer (SP) and Interrupt Stack Pointer

Table 2-1 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 for the μPD17203, however, **the $\overline{\text{WDOUT}}$ pin goes low if the value of SP is 6 or greater.** For the μPD17204, there are only seven address stack registers available, **the $\overline{\text{WDOUT}}$ pin goes low if the value of SP exceeds seven.**

Table 2-1. 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
RET1	+1	+1

2.4 DATA MEMORY (RAM)

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

2.4.1 Memory Configuration

Figure 2-4 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 pointed by the general pointer (RP) in the system register (SYSREG).

(4) Port register

A port data register is resident on addresses 70H to 73H of BANK0 and 70H-72H of BANK1 (7 nibbles long) of data memory.

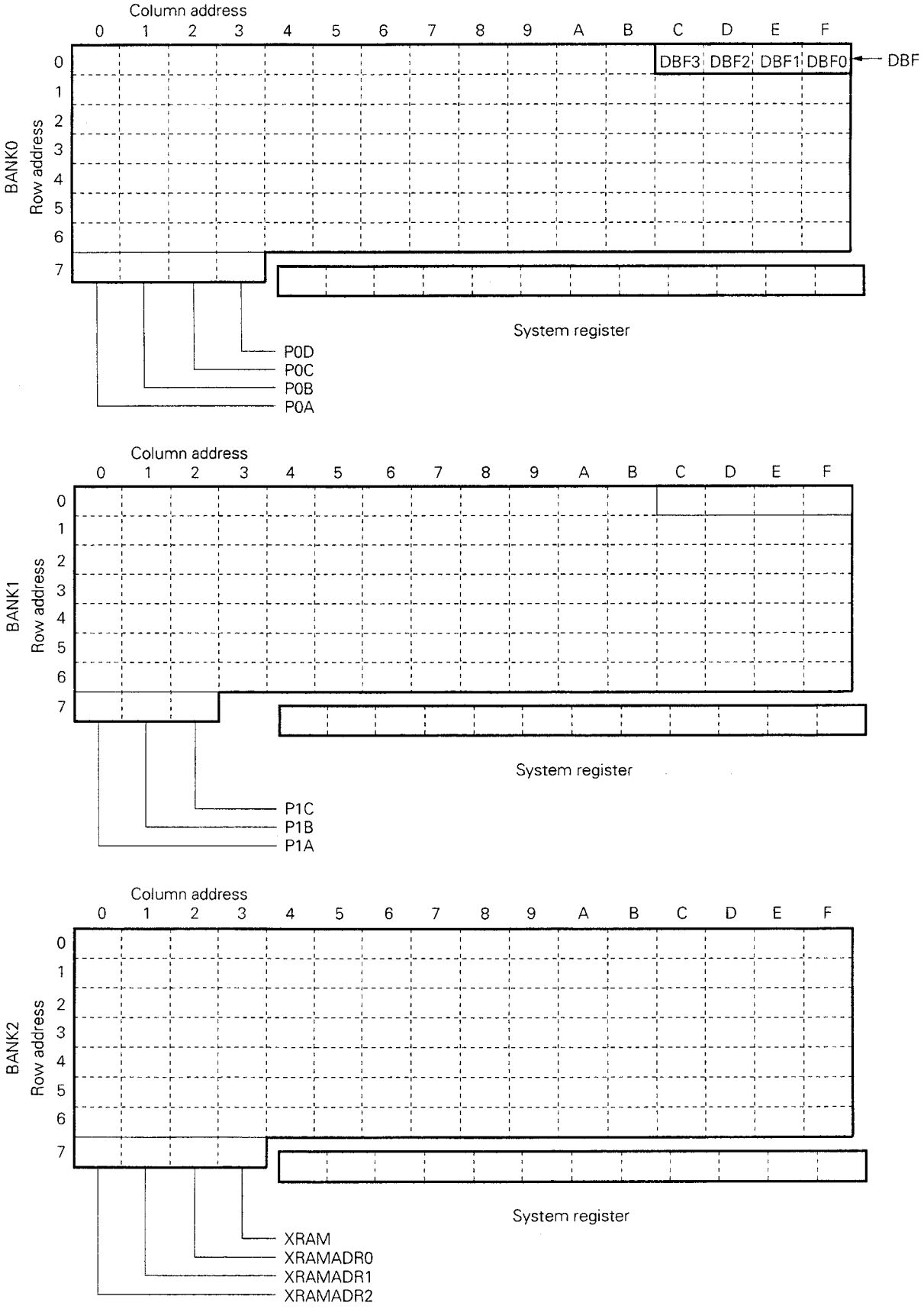
(5) XRAM data register (XRAM register)

An XRAM data register is resident on addresses 70H to 73H (4 nibbles long) of BANK2 of data memory. Refer to **9. STATIC RAM (XRAM)**.

(6) General-purpose data memory

The general-purpose data memory area is an area of the data memory excluding the system register area, the port register area, and the XRAM register area. This memory area has a total of 336 nibbles (in BANK0 to BANK2).

Figure 2-4. RAM Configuration



2.4.2 System Registers (SYSREG)

The system registers are registers that are directly related to control of the CPU. These registers are mapped to addresses 74H-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)

Figure 2-5. 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) / Data memory row address pointer (MP)			General register pointer (RP)		Program status word (PSWORD)
Symbol	AR 3	AR 2	AR 1	AR 0	WR	BANK	IXH / MPH	IXM / MPL	IXL	RPH	RPL	PSW
Bit	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0	b3 b2 b1 b0
Data	0 0 0 0 ← (AR) →				← →	← (BANK) 0 0 →	MPE 0 0	← (IX) → / ← (MP) →		0 0 ← (RP) →		BCD CMP CY Z IXE
Initial Value At Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	Undefined	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

2.4.3 General Register (GR)

A general register is a 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 2-6 shows the configuration of the general register.

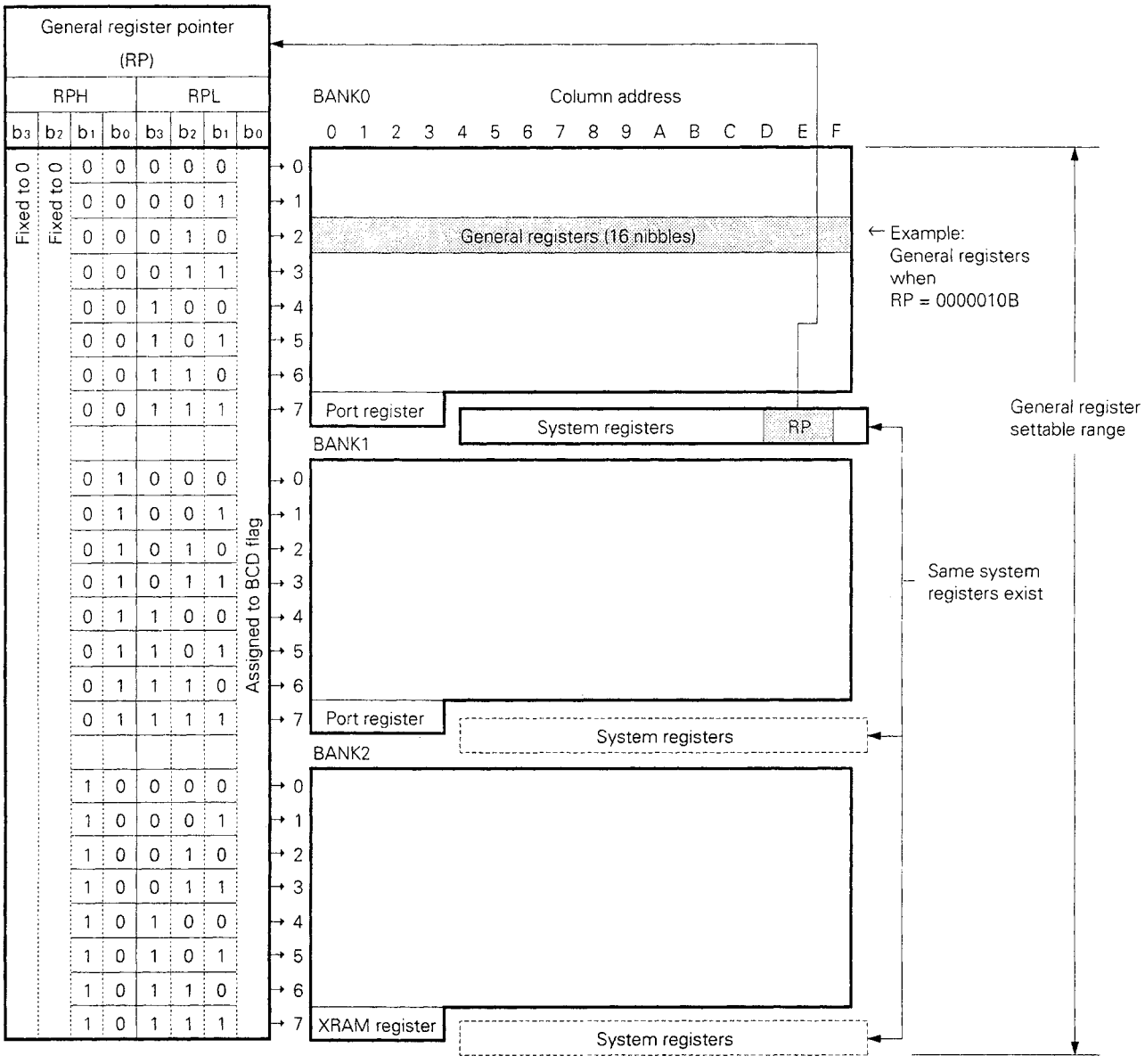
A general register occupies 16 nibbles (16 x 4 bits) on a selected row address of the data memory.

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.

Figure 2-6. Configuration of General Registers



2.4.4 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 2-7 shows the relationship between the data buffer and peripheral hardware.

Figure 2-7. Data Buffer and Peripheral Hardware

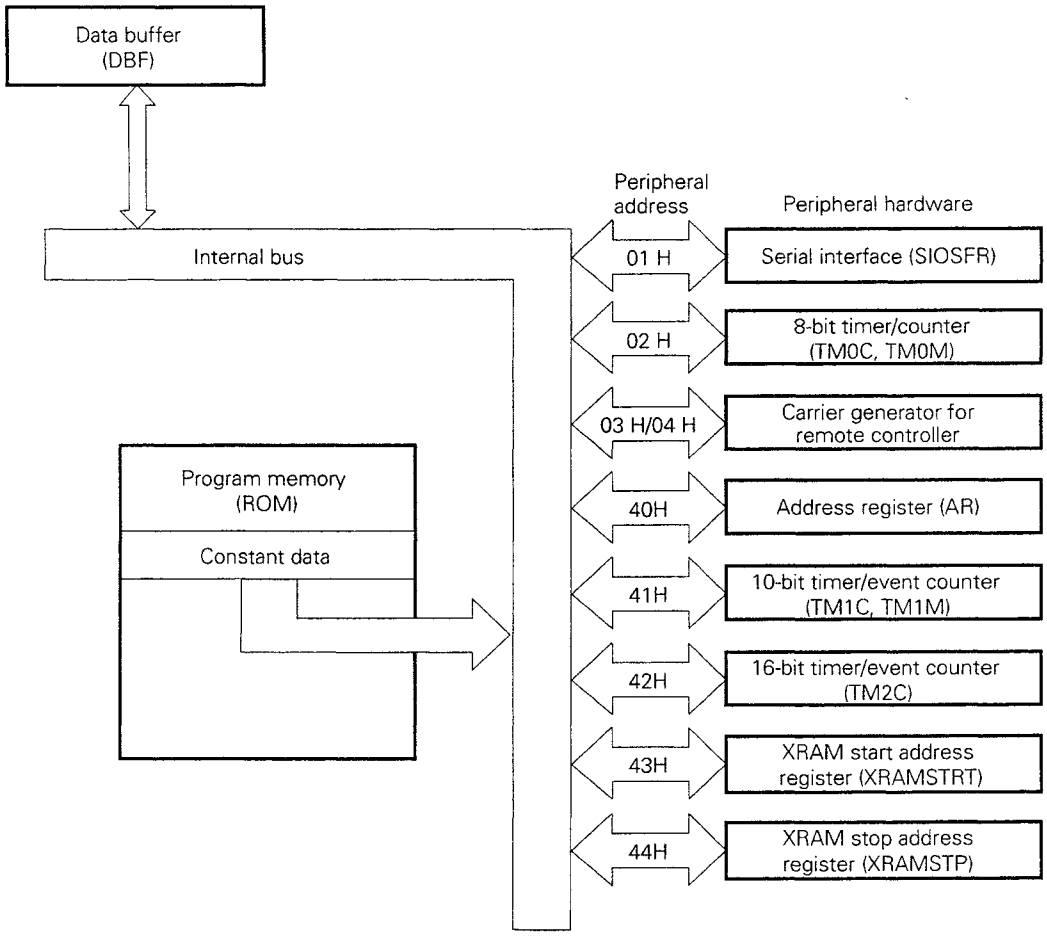


Table 2-2. Relations between Peripheral Hardware and Data Buffer

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/Counter	8-bit counter	TM0C	02H	DBF0, DBF1	Only GET
	8-bit modulo register	TM0M	02H	DBF0, DBF1	Only PUT
Remote Controller Carrier Generator Circuit	NRZ low-level period setting modulo register	NRZLTMM	03H	DBF0, DBF1	Both PUT & GET
	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.)
Address Register	Address register	AR	40H	DBF0-DBF3	PUT (Bits 0-3 of AR3 are don't care.)[Note 1] GET (Always clears bits 0-3 of AR3 to 0.)[Note 2]
10-bit timer/counter	10-bit counter	TM1C	41H	DBF0-DBF3	Only GET
	10-bit modulo register	TM1M	41H	DBF0-DBF3	Only PUT
16-bit timer/counter	16-bit counter	TM2C	42H	DBF0-DBF3	Only GET
XRAM address register	XRAM start address register	XRAMSTRT	43H	DBF0-DBF3	Both PUT & GET
	XRAM stop address register	XRAMSTP	44H	DBF0-DBF3	Both PUT & GET

Notes 1. The μPD17203A: Bits 1-3 of AR3 are don't care.

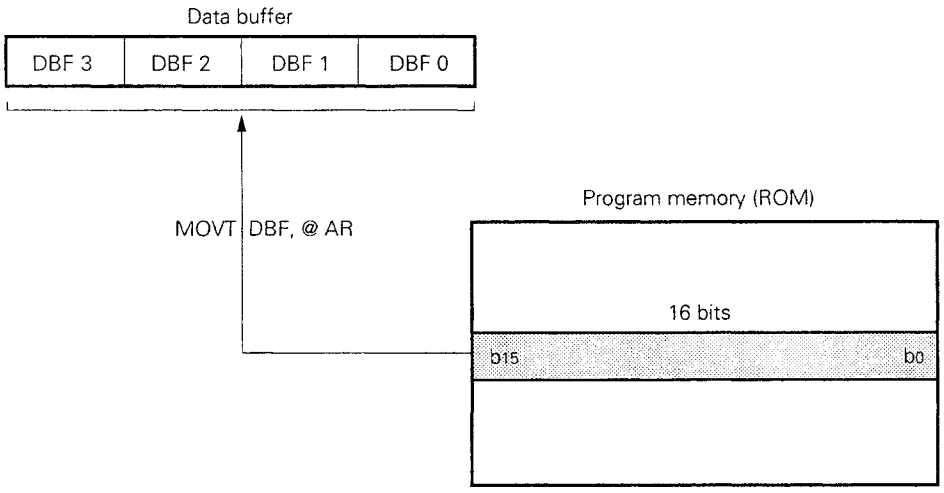
2. The μPD17204 : Always clears bits 1-3 to 0.

(2) Table reference

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

The function of the MOVT instruction is explained below.

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



(3) 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 assembler (AS17K)
  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.

2.5 REGISTER FILE (RF)

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

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

2.5.1 Configuration of Register File

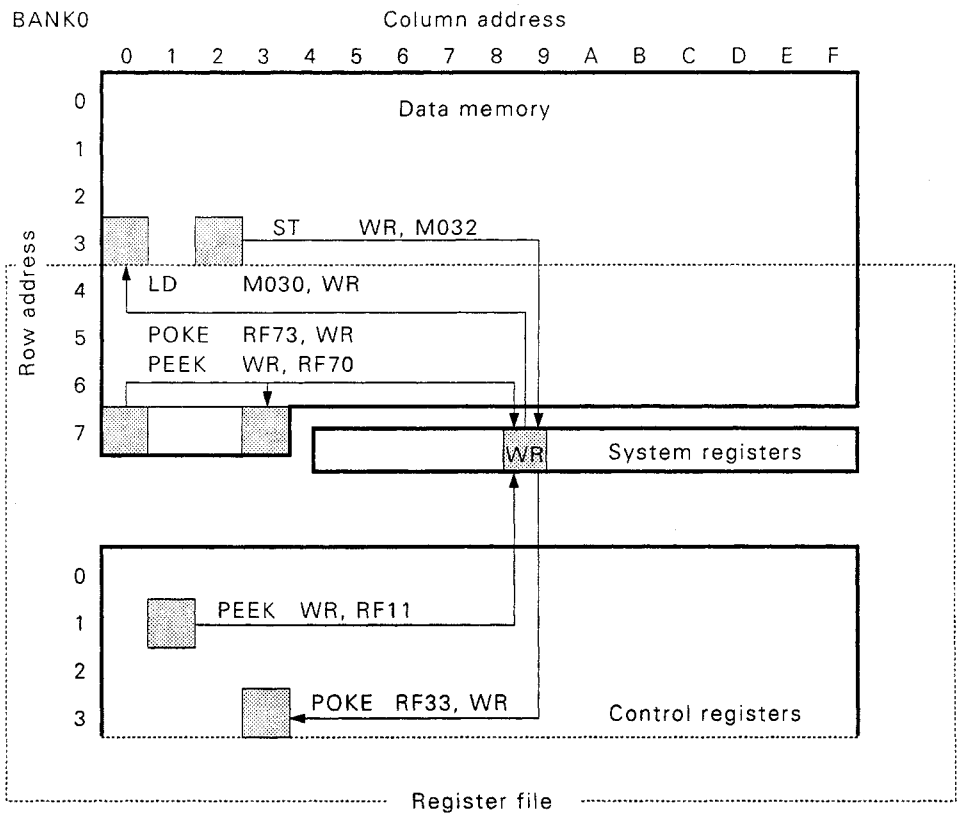
Figure 2-8 shows the configuration of the register file and how the register file is accessed by the PEEK and POKE instructions.

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.

Figure 2-8. Configuration of Register File and Accessing Register File by PEEK and POKE Instructions



2.5.2 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 33 nibbles are actually used. The remaining 31 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 assembler (AS17K), the macro instructions listed below, which are embedded as flag type symbol manipulation instructions, can be used. The macro instructions allow the contents of the register file to be manipulated in bit units.

For the configuration of the control register, refer to **Figure 3-1 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

2.5.3 Notes on Using Register Files

When using the register files, bear in mind the points described below. For details, refer to **μPD172×× Subseries User's Manual.**

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

When manipulating the write-only (W), the read-only (R) and unused control registers by using 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 assembler (AS17K)
  An error occurs if an instruction is executed to read data to the write-only register.
  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)
  An undefined value is read if the write-only register is read, and an error does not occur.
  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 (AS17K) 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.

3. PORTS

3.1 PORT 0A ($P0A_0$-$P0A_3$)

Port 0A is a 4-bit, general-purpose input/output port and can be set in either 4 bit input or output mode by P0AGIO in the register file. Input data can be read and output data can be sent out through port register P0A.

In input mode, this register serves as a CMOS input port. In output mode, it serves as a CMOS output port. If the contents of the P0A register are read with the port in output mode, the status of the port output pins can be determined.

Standby mode is released whenever a low-level signal is input to at least one of the port's pins.

A pull-up resistor can be connected to each bit of this port (mask option).

3.2 PORT 0B ($P0B_0$-$P0B_3$)

Port 0B is also a 4-bit, general-purpose input/output port and can be set in either 4 bit input or output mode by P0BGIO in the register file. Input data can be read and output data can be sent out through port register P0B.

In input mode, this register serves as a CMOS input port. In output mode, it serves as a CMOS output port. If the contents of the P0B register are read with the port in output mode, the status of the port output pins can be determined.

Standby mode is released whenever a low-level signal is input to at least one of the port's pins.

A pull-up resistor can be connected to each bit of this port (mask option).

3.3 PORT 0C ($P0C_0$-$P0C_3$)

Port 0C is another 4-bit, general-purpose input/output port and can be set in either 4 bit input or output mode by P0CGIO in the register file. Input data can be read and output data can be sent out through port register P0C.

In input mode, this register serves as a CMOS input port. In output mode, it serves as an N-ch open-drain output port. If the contents of the P0C register are read with the port in output mode, the status of the port output pins can be determined.

3.4 PORT 0D ($P0D_0$-$P0D_3$)

Port 0D is a 4-bit, general-purpose input/output port and can be set in either 4 bit input or output mode by P0DGIO in the register file. Input data can be read and output data can be sent out through port register P0D.

In input mode, this register serves as a CMOS input port. In output mode, it serves as an N-ch open-drain output port. If the contents of the P0D register are read with the port in output mode, the status of the port output pins can be determined.

3.5 PORT 1A ($P1A_0$-$P1A_3$)

Port 1A is a 4-bit, general-purpose input/output port and can be set in either 4 bit input or output mode by P1ABIO0-P1ABIO3 in the register file. Input data can be read and output data can be sent out through port register P1A.

In input mode, this register serves as a CMOS input port. In output mode, it serves as an N-ch open-drain output port. If the contents of the P1A register are read with the port in output mode, the status of the port output pins can be determined.

A pull-up resistor can be connected to each bit of this port by using P1ABIO0-P1ABIO3.

3.6 PORT 1B ($P1B_0$-$P1B_3$)

Port 1B is a 4-bit, general-purpose input/output port whose pins are multiplexed with the external signal output pins of the internal timer. TM0OE, TM1OE, and TM2OE in the register file, specify whether the pins of this port are used as port pins or as timer output pins.

(1) To use as a 4-bit I/O port

The port can be set in input or output mode in units of 1 bit by P1BBIO0-P1BBIO3 in the register file. Input data can be read and output data can be written via port register P1B.

In input mode, this register serves as a CMOS input port, while in output mode, it serves as an N-ch open-drain output port. If the contents of the P1B register are read with the port in output mode, the status of the port's output pins can be determined.

A pull-up resistor can be connected to each bit of this by using P1BBPU0-P1BBPU3.

(2) To use as internal timer external signal output pins

TM0OE, TM1OE, and TM2OE specify whether the port is used as an I/O port (P1B0, P1B1, P1B2, and P1B3) or as the external signal output pins of the internal timer (TM0OUT, TM1OUT, and TM2OUT).

3.7 PORT 1C ($P1C_0$-$P1C_3$)

Port 1C is a 4-bit, general-purpose input/output port whose pins are multiplexed with serial interface pins. SIOEN on the register file specifies whether the pins of this port are used as port pins or as interface pins .

(1) To use as a 4-bit I/O port

The port can be set in the input or output mode in units of 1 bit by P1CBIO0-P1CBIO3 in the register file. Input data can be read and output data can be set through port register P1C.

In input mode, this register serves as a CMOS input port, while in output mode it serves as an N-ch open-drain output port. The contents of the P1C register are read with the port in output mode, the status of the port's output pins can be determined.

A pull-up resistor can be connected to each bit of this port by using P1CBPU0-P1CBPU3.

(2) To use as serial interface pins

SIOEN specifies whether the port is used as an I/O port ($P1C_0$, $P1C_1$, $P1C_2$, and $P1C_3$) or as serial interface pins.

3.8 INT PIN

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

The level of this 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 flag is reset to "0" (refer to **Figure 10-2 INT and INTENV Flags**).

★ **Figure 3-1. Relation between Port Registers and Pins**

Bank	Address	Port	Bit		Output Type	Read-out content		Write-in content		On reset
						Input mode	Output mode	Input mode	Output mode	
0	70H	Port 0A	b3	P0A3	CMOS push-pull	Pin status		Output latch		Input mode (Pull-up resistor possible by mask option)
			b2	P0A2						
			b1	P0A1						
			b0	P0A0						
	71H	Port 0B	b3	P0B3	CMOS push-pull					
			b2	P0B2						
			b1	P0B1						
			b0	P0B0						
	72H	Port 0C	b3	P0C3	N-ch open drain					Input mode
			b2	P0C2						
			b1	P0C1						
			b0	P0C0						
	73H	Port 0D	b3	P0D3	N-ch open drain					
			b2	P0D2						
			b1	P0D1						
			b0	P0D0						
1	70H	Port 1A	b3	P1A3	N-ch open drain					Input mode (Pull-up resistor possible in bit unit)
			b2	P1A2						
			b1	P1A1						
			b0	P1A0						
	71H	Port 1B	b3	P1B3 Note 1	N-ch open drain					
			b2	P1B2 Note 1						
			b1	P1B1 Note 1						
			b0	P1B0						
	72H	Port 1C	b3	P1C3	CMOS push-pull					Input mode
			b2	P1C2 Note 2						
			b1	P1C1 Note 2						
			b0	P1C0 Note 2						

Notes 1. When the TM0OE, TM1OE and TM2OE 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.

3.9 PORT CONTROL REGISTER FILE

3.9.1 Input/Output Switching of Group I/O

I/Os that can be set in input/output mode in 4-bit units are called group I/Os. P0A, P0B, P0C, and P0D are available as the group I/O ports. The following register file is used to switch the input mode to the output mode or vice versa. When the mode is changed from input to output, the contents of the port registers are immediately output to the corresponding ports.

Figure 3-2. Input/Output Control Register for Group I/O

3	2	1	0	Address	When reset	R/W
P0DGIO	P0CGIO	P0BGIO	P0AGIO	RF : 37H	0H	R/W

P0AGIO	P0A input/output setting
0	Sets P0A for input mode
1	Sets P0A for output mode

P0BGIO	P0B input/output setting
0	Sets P0B for input mode
1	Sets P0B for output mode

P0CGIO	P0C input/output setting
0	Sets P0C for input mode
1	Sets P0C for output mode

P0DGIO	P0D input/output setting
0	Sets P0D for input mode
1	Sets P0D for output mode

3.9.2 Input/Output Switching of Bit I/O

I/Os that can be set in input/output mode in bit units are called bit I/Os. P1A, P1B, and P1C are available as the bit I/O ports. The following register file is used to switch the input mode to the output mode or vice versa. When the input mode is changed to the output mode, the port register contents of each P1A, P1B, and P1C bit are output to the corresponding port bit.

Figure 3-3. Input/Output Control Register for Bit I/O: P1A

3	2	1	0	Address	When reset	R/W
P1ABIO3	P1ABIO2	P1ABIO1	P1ABIO0	RF : 27H	0H	R/W

P1ABIO0	$P1A_0$ input/output setting
0	Sets $P1A_0$ for input mode
1	Sets $P1A_0$ for output mode

P1ABIO1	$P1A_1$ input/output setting
0	Sets $P1A_1$ for input mode
1	Sets $P1A_1$ for output mode

P1ABIO2	$P1A_2$ input/output setting
0	Sets $P1A_2$ for input mode
1	Sets $P1A_2$ for output mode

P1ABIO3	$P1A_3$ input/output setting
0	Sets $P1A_3$ for input mode
1	Sets $P1A_3$ for output mode

Figure 3-4. Input/Output Control Register for Bit I/O: P1B

3	2	1	0	Address	When reset	R/W
P1BBIO3	P1BBIO2	P1BBIO1	P1BBIO0	RF : 26H	0H	R/W

P1BBIO0	$P1B_0$ input/output setting
0	Sets $P1B_0$ for input mode
1	Sets $P1B_0$ for output mode

P1BBIO1	$P1B_1$ input/output setting
0	Sets $P1B_1$ for input mode
1	Sets $P1B_1$ for output mode

P1BBIO2	$P1B_2$ input/output setting
0	Sets $P1B_2$ for input mode
1	Sets $P1B_2$ for output mode

P1BBIO3	$P1B_3$ input/output setting
0	Sets $P1B_3$ for input mode
1	Sets $P1B_3$ for output mode

Figure 3-5. Input/Output Control Register for Bit I/O: P1C

3	2	1	0	Address	When reset	R/W
P1CBIO3	P1CBIO2	P1CBIO1	P1CBIO0	RF : 25H	0H	R/W

P1CBIO0	$P1C_0$ input/output setting
0	Sets $P1C_0$ for input mode
1	Sets $P1C_0$ for output mode

P1CBIO1	$P1C_1$ input/output setting
0	Sets $P1C_1$ for input mode
1	Sets $P1C_1$ for output mode

P1CBIO2	$P1C_2$ input/output setting
0	Sets $P1C_2$ for input mode
1	Sets $P1C_2$ for output mode

P1CBIO3	$P1C_3$ input/output setting
0	Sets $P1C_3$ for input mode
1	Sets $P1C_3$ for output mode

3.9.3 Pull-up Resister On/Off Control

The P1A and P1B ports are of N-ch open-drain ports. The P1C port is a CMOS port. A pull-up resister can be connected to each of these ports by manipulating the register file as follows.

Figure 3-6. Pull-up Resistor On/Off Control Register: P1A

3	2	1	0	Address	When reset	R/W
P1ABPU3	P1ABPU2	P1ABPU1	P1ABPU0	RF : 17H	0H	R/W

P1ABPU0	$P1A_0$ internal pull-up resistor specification
0	Disconnects $P1A_0$ pull-up resistor
1	Connects $P1A_0$ pull-up resistor

P1ABPU1	$P1A_1$ internal pull-up resistor specification
0	Disconnects $P1A_1$ pull-up resistor
1	Connects $P1A_1$ pull-up resistor

P1ABPU2	$P1A_2$ internal pull-up resistor specification
0	Disconnects $P1A_2$ pull-up resistor
1	Connects $P1A_2$ pull-up resistor

P1ABPU3	$P1A_3$ internal pull-up resistor specification
0	Disconnects $P1A_3$ pull-up resistor
1	Connects $P1A_3$ pull-up resistor

Figure 3-7. Pull-up Resistor On/Off Control Register: P1B

3	2	1	0	Address	When reset	R/W
P1BBPU3	P1BBPU2	P1BBPU1	P1BBPU0	RF : 16H	0H	R/W

P1BBPU0	$P1B_0$ internal pull-up resistor specification
0	Disconnects $P1B_0$ pull-up resistor
1	Connects $P1B_0$ pull-up resistor

P1BBPU1	$P1B_1$ internal pull-up resistor specification
0	Disconnects $P1B_1$ pull-up resistor
1	Connects $P1B_1$ pull-up resistor

P1BBPU2	$P1B_2$ internal pull-up resistor specification
0	Disconnects $P1B_2$ pull-up resistor
1	Connects $P1B_2$ pull-up resistor

P1BBPU3	$P1B_3$ internal pull-up resistor specification
0	Disconnects $P1B_3$ pull-up resistor
1	Connects $P1B_3$ pull-up resistor

Figure 3-8. Pull-up Resistor On/Off Control Register: P1C

3	2	1	0	Address	When reset	R/W
P1CBPU3	P1CBPU2	P1CBPU1	P1CBPU0	RF : 15H	0H	R/W

P1CBPU0	$P1C_0$ internal pull-up resistor specification
0	Disconnects $P1C_0$ pull-up resistor
1	Connects $P1C_0$ pull-up resistor

P1CBPU1	$P1C_1$ internal pull-up resistor specification
0	Disconnects $P1C_1$ pull-up resistor
1	Connects $P1C_1$ pull-up resistor

P1CBPU2	$P1C_2$ internal pull-up resistor specification
0	Disconnects $P1C_2$ pull-up resistor
1	Connects $P1C_2$ pull-up resistor

P1CBPU3	$P1C_3$ internal pull-up resistor specification
0	Disconnects $P1C_3$ pull-up resistor
1	Connects $P1C_3$ pull-up resistor

3.9.4 Timer Output Switching

The TM0OE, TM1OE, and TM2OE bits on the register file can be used to specify the following P1B port pins as the timer output pins:

Figure 3-9. P1B/Timer Output Control Register

3	2	1	0	Address	When reset	R/W
TM2OE	TM1OE	TM0OE	0	RF : 24H	0H	R/W

TM0OE	$P1B_1$ port/timer 0 output selection
0	Uses $P1B_1$ as input/output port
1	Uses $P1B_1$ as timer 0 output

TM1OE	$P1B_2$ port/timer 1 output selection
0	Uses $P1B_2$ as input/output port
1	Uses $P1B_2$ as timer 0 output

TM2OE	$P1B_3$ port/timer 2 output selection
0	Uses $P1B_3$ as input/output port
1	Uses $P1B_3$ as timer 0 output

3.9.5 Serial Input/Output Switching

The SIOEN bit on the register file can be used to specify the $P1C_0$ to $P1C_2$ pins as a serial I/O port. Even if these pins are used as the serial I/O port, the $P1C_3$ pin can be used as an ordinary I/O port.

The SIOCK0, SIOCK1, SIOHIZ, and SIOTS bits on the register files are used to control serial input/output mode. (For details, see **8. SERIAL INTERFACE**.)

Figure 3-10. P1C/Serial Input/Output Control Register

3	2	1	0	Address	When reset	R/W
0	0	0	SIOEN	RF : 23H	0H	R/W

SIOEN	Input/output/serial input/output port selection
0	Uses $P1C_0$ to $P1C_2$ as input/output port
1	Uses $P1C_0$ to $P1C_2$ as serial input/output ports

4. CLOCK GENERATOR CIRCUITS

The μPD17203A has been provided with two clock generator circuits: both a main clock (X) and a subclock (XT) generator circuit. Both of the circuits can be used as the system clock generator circuit.

Figure 4-1 shows the configuration of a register that controls the system clock generator circuits.

The SYSCK flag (RF: address 02H, bit 1) of this control circuit specifies which clock the main or the subclock, will be used as the system clock. By resetting the XEN flag (RF: address 02H, bit 0) to 0, oscillation of the main clock can be stopped and, therefore, the power dissipation from the microcontroller can be reduced.

When using the subclock, make sure to connect a 0.1-μF capacitor to the V_{REG} pin to stabilize oscillation. If the subclock is not used (specified by mask option), connect the XT_{IN} pin to GND and the XT_{OUT} pin to the V_{REG} pin.

Figure 4-1. System Clock Control Register

3	2	1	0	Address	When reset	R/W
0	0	SYSCK	XEN	RF : 02H	Undefined[Note]	R/W

XEN	Enables oscillation of main clock
0	Stops oscillation of main clock
1	Enables oscillation of main clock

SYSCK	System clock selection
0	Selects subclock as the system clock
1	Selects main clock as the system clock

Note Depending on the mask option. ★

4.1 SYSTEM CLOCK SELECTION

The system clock, i.e., the main clock or the subclock, is selected by the SYSCK flag (RF: address 02H, bit 1) as shown in Figure 4-1.

(1) Subclock selection

By resetting the SYSCK flag to 0, the subclock is selected as the system clock.

If NOXT is set as a mask option, the subclock cannot be selected (SYSCK and XEN cannot be reset to 0).

Caution Make sure that sufficient time elapses on power application so that oscillation subclock is stabilized (confirm through a program that the IRQWTM flag (RF: address 3AH, bit 0) is set at a regular interval).

(2) Main clock selection

By setting the SYSCK flag to 1, the main clock will be selected as the system clock.

If NOX is set as a mask option, 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.

4.2 MAIN CLOCK OSCILLATION CONTROL FUNCTION

When the subclock is used as the system clock, ocillation of the main clock can be stopped or started 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 ocillation of the main clock is started (by setting the XEN flag), make sure that an ocillation stabilization time of about 10 ms elapses.

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

5. TIMER FUNCTION

The μPD17203A is provided with four types of timers: timer 0, timer 1, timer 2, and clock timer. These timers are used to read remote controller signals.

The GET and PUT instructions are used to control these timers. Bits on the register files are also used to control the timers.

Timers

(1) Timer 0 .....8-bit timer/counter (with modulo function)
(2) Timer 1 .....10-bit timer (with modulo function)
(3) Timer 2 .....16-bit timer
(4) Clock timer

5.1 8-BIT TIMER 0 AND REMOTE CONTROLLER CARRIER GENERATOR CIRCUIT

5.1.1 Configuration of 8-bit Timer

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

The 8-bit timer consists of an 8-bit counter (TM0C), an 8-bit modulo register (TM0M), a comparator that checks the count value of the timer and the value of the modulo register for coincidence, and a selector that selects the 8-bit timer operating clock.

The starting and stopping of the 8-bit timer, and the resetting of the 8-bit counter are controlled by TM0EN (address 33H, bit 3) and TM0RES (address 33H, bit 2) in the register file. The 8-bit timer operating clock is selected by TM0CK1 (address 33H, bit 1) and TM0CK0 (address 33H, bit 0) in the register file. To set the input clock of timer 0 to external input or clock noise rejecter circuit modes, use TM0CK0, TM0CK1, TM0EXCK (address 14H, bit 0), and TM0INEN (address 13H, bit 3).

The value of the 8-bit counter is read by using the GET instruction through the data buffer (DBF). No value can be input to the 8-bit counter. To put a value in the modulo register, use the PUT instruction through the DBF. The value of the modulo register cannot be read. Since the data register used for the 8-bit counter and the modulo register have the same address, the 8-bit counter is accessed for reading and the 8-bit modulo register is accessed for writing only.

When the count value of the counter coincides with the value of the modulo register, the interrupt request flag (IRQTM0: address 3DH, bit 0) of the register file is set, and the output level of the P1B1/TM0OUT pin is inverted. By setting TM0RES, TM0OUT will be initialized and will output a high level signal.

TM0C ★

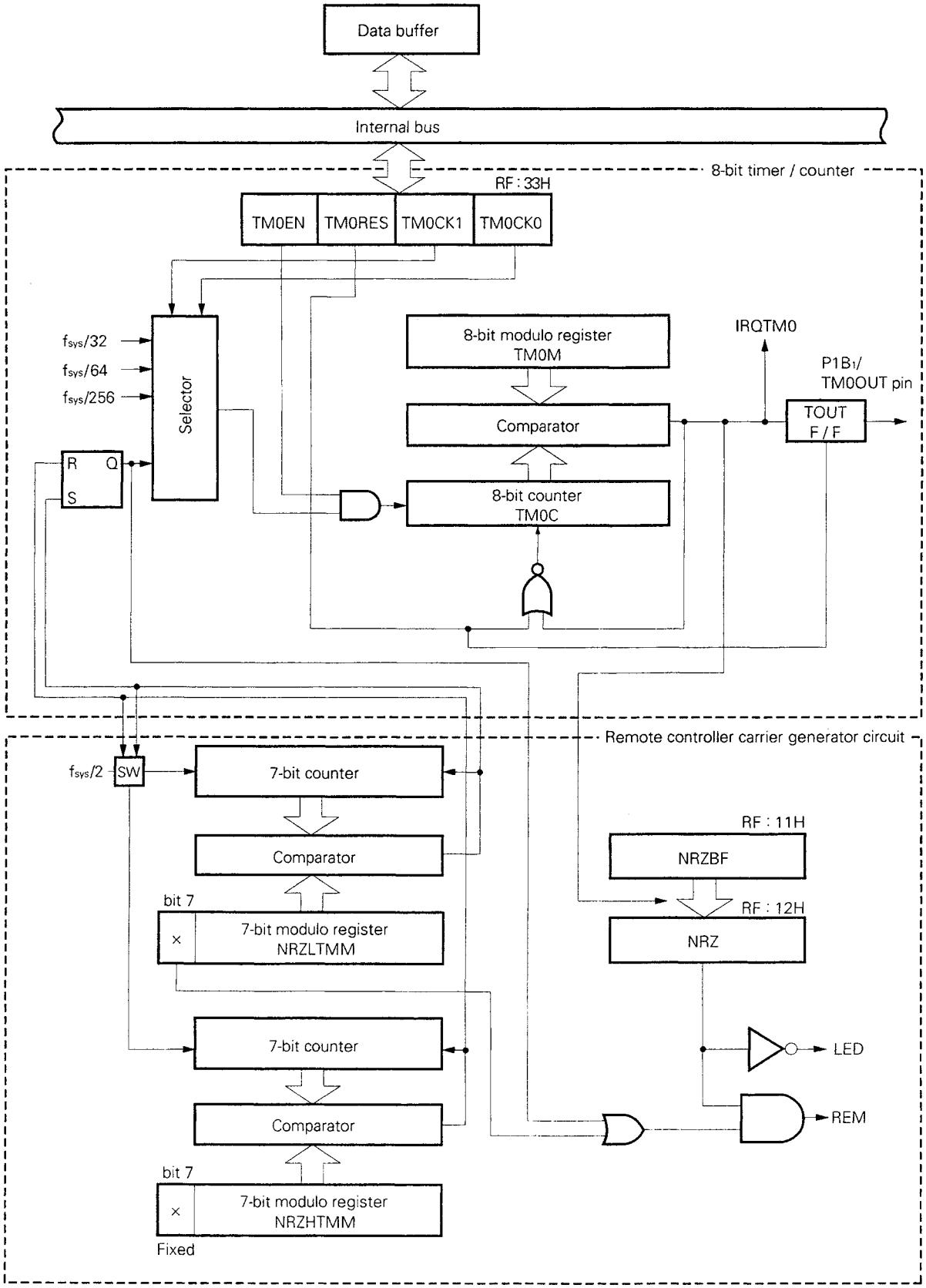
7	6	5	4	3	2	1	0	Address	At reset	R/W
8-bit counter								Peripheral register : 02H	00H	R

TM0M ★

7	6	5	4	3	2	1	0	Address	At reset	R/W
8-bit modulo register								Peripheral register : 02H	FFH	W

Caution Do not clear TM0M to 0. (IRQTM0 is not set)

★ **Figure 5-1. Configuration of 8-bit Timer/Counter and Remote Controller Carrier Generator Circuit**



Remarks 1. f_{sys} (system clock frequency): fx or F_{XT}

2. TM0M, TM0C, NRZLTMM and NRZHTMM are peripheral registers.

The following shows the register files for timer 0.

Figure 5-2. Control Register for Timer 0 (1/2)

3	2	1	0	Address	When reset	R/W
TM0EN	TM0RES	TM0CK1	TM0CK0	RF : 33H	8H	R/W[Note 1]

TM0CK1	TM0CK0	8-bit timer clock source selection
0	0	Count clock: fsys/32 (Measurement time range: 8 μs to 2.048 ms)
0	1	Count clock: fsys/64 (Measurement time range: 16 μs to 4.096 ms)
1	0	Count clock: fsys/256 (Measurement time range: 64 μs to 16.384 ms)
1	1	Remote control carrier generator output or TM0IN input or CMPOUT output[Note 2] ★

Value indicated by parentheses is for when fsys (system clock) = fx = 4MHz

TM0RES	8-bit timer reset flag
0	Data read out is always "0"
1	Resets 8-bit counter and IRQTM0 ★

TM0EN	8-bit timer count enable flag
0	Stops 8-bit timer count operation
1[Note 3]	Enable 8-bit timer count operation

Notes 1. Bit 2 is write-only bit.

2. Switching of remote control carrier generator output/TM0IN input/CMPOUT output is performed by the TM0EXCK bit on the register file. ★

3. When the STOP mode is cleared, it is always set to 1. ★

Figure 5-2. Control Register for Timer 0 (2/2)

3	2	1	0	Address	When reset	R/W
0	0	TM0 TM1EN	TM0 EXCK	RF : 14H	8H	R/W

TM0 EXCK	Timer 0 external clock source selection flag
0	Uses internal remote control carrier generator clock as timer 0 input clock
1	Uses TM0IN input or CMPOUT output as timer 0 input clock

TM0 TM1EN	Tiemer 0, timer 1 start
0	Asynchronous timer 0 and timer 1 operates
1	Synchronous timer 0 and timer 1 operation

Cautions 1. If TM0TM1EN = 1, timers 0 and 1 will start in synchronization by enabling (setting) either of the timer 0 and timer 1 enable flags, TM0EN and TM1EN.
Moreover, timer 1 will be stopped by the coincides with the signal of timer 0. At that time, both TM0EN and TM1EN are stopped (reset).

★ **2. When TM0TM1EN = 1, be sure to set TM0EXCK and (TM0CK1, TM0CK0) = (1, 1) (won't operate with other settings).**
Also, timer 1 counts only while the envelope output (refer to 7.1.3 ENVELOPE CIRCUIT) is at high level.

TM0TM1EN is used to start timer 1 counting in synchronization with the rising edge of the input clock to timer 0, and stop counting of timer 1 when the count value of timer 0 coincides with the value of the modulo register. In this case, timers 0 and 1 automatically stop their counting operations, and retain their values. By using this control bit, the time that elapses until the value put in the modulo register of timer 0 coincides with the count value of timer 0 can be measured. This feature is ideal for such applications as measuring the carrier frequency of the pulse received from a remote controller.

5.1.2 Carrier Generation Circuit for Remote Controller

The μPD17203A is provided with a carrier generation circuit for remote controller.

The carrier generation circuit for remote controller consists of a 7-bit counter, a modulo register for setting NRZ high-level period (NRZHTMM), and a modulo register for setting NRZ low-level period (NRZLTMM). The high- and low-level periods are set in the corresponding modulo registers to determine the carrier duty factor and carrier frequency. Values are set in the modulo registers via the data buffer (DBF). ★

The clock input to the 7-bit counter is obtained by dividing the system clock by 2. That is, when a 4-MHz oscillator (fx) is used, the input clock frequency is 2 MHz. When a 32.768-kHz oscillator (fxT) is used, the input clock frequency is 16.384 kHz. ★

The name of the modulo register for setting NRZ high-level period is NRZHTMM, and the name of the modulo register for setting NRZ low- level period is NRZLTMM. The PUT instruction is used to write data in these registers and the GET instruction used to read data from them.

Also NRZLTMM bit 7 is the bit that controls whether the carrier is output on the REM pin or high level is output. For carrier output, be sure to set bit 7 to 0. ★

NRZLTMM ★

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

Bit 7	REM pin output control
0	When NRZ=1, carrier is output to REM pin
1	When NRZ=1, high level is output to REM pin

NRZHTMM ★

7	6	5	4	3	2	1	0	Address	When reset	R/W
0		7-bit modulo register						Peripheral register : 04H	Undefined	R/W

Bit 7
Fixed at 0

5.1.3 Remote controller signal output control

The NRZ and NRZBF bits on the register file and timer 0 are used to control the output to the REM pin that outputs the carrier. while the NRZ bit is 1, the clock generated in the remote controller carrier generation circuit is output to the REM pin. While the NRZ bit is 0, a low-level signal is output to the REM pin. The content of the NRZBF bit is automatically transmitted to the NRZ bit by the timer 0 interrupt signal. If data is set in the NRZBF bit in advance, the REM pin status changes in synchronization with the count operation of timer 0. The content of the NRZ bit is output to the LED pin. That is, when the NRZ bit is 0, a high-level signal is output to the LED pin. When it is 1, a low-level signal is output to the LED pin.

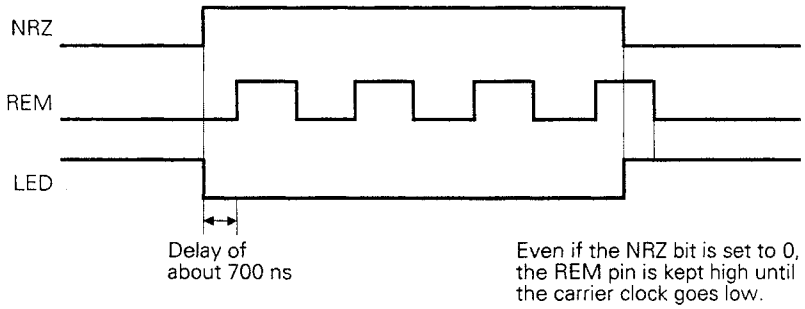
If the timer 0 interrupt signal occurs when the REM pin is high, output to the REM pin does not comply with the content of the NRZ bit until the carrier clock goes low. This processing is effective for holding the high-level pulse width of the output carrier constant (see **Figure 10-3**).

When the NRZ bit is 0, the remote controller carrier generation circuit stops. When the output of the remote control carrier generation circuit is supplied to the timer 0 as the clock, the clock operation is continued even if the NRZ bit is 0.

An example of outputting a remote controller signal to the REM pin is shown below.

Figure 5-3. Example of Remote Controller Carrier Output Waveform

★ **NRZLTMM bit 7 = 0 (with carrier)**



★ **Note** This is the value when (TM0CK1, TM0CK0) ≠ (1, 1).

The value when (TM0CK1, TM0CK0) = (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.

★ **NRZLTMM bit 7 = 1 (without carrier)**

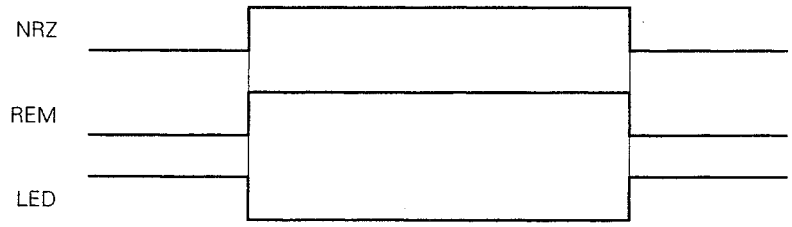


Figure 5-4. NRZ/NRZBF Register File

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

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

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

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

Setting carrier frequency and duty factor

Where the frequency is fx and carrier frequency is fc when the main clock (X) is used as the system clock,

ℓ (division ratio) = fx/(2 × fc)

By dividing ℓ into a duty factor of m:n, values are put in the modulo register as follows:

Set value of high-level period = $\{\ell \times m/(m + n)\} - 1$

Set value of low-level period = $\{\ell \times n/(m + n)\} - 1$

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

ℓ = 4 MHz/(2 × 38 kHz) = 52.6, and m : n = 1 : 2, the value of the modulo register is:

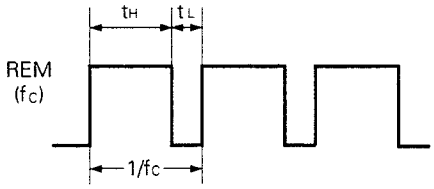
High-level period (NRZHTMM) ≒ 17 (11H)

Low-level period (NRZLTMM) ≒ 34 (22H)

Therefore, the carrier frequency is 37.74 kHz.

★ **Table 5-1. Carrier Frequency List (fx = fsys = 4 MHz)**

Set Value		t_H (μs)	t_L (μs)	1/fc (μs)	fc (kHz)	Duty
NRZHTMM	NRZLTMM					
00H	00H	0.5	0.5	1.0	1000	1/2
01H	02H	1.0	1.5	2.5	400	2/5
04H	04H	2.5	2.5	5.0	200	1/2
09H	09H	5.0	5.0	10.0	100	1/2
0FH	10H	8.0	8.5	16.5	60.6	1/2
01H	21H	8.0	17.0	25.0	40.0	1/3
11H	21H	9.0	17.0	26.0	38.5	1/3
11H	22H	9.0	17.5	26.5	37.7	1/3
19H	35H	13.0	27.0	40.0	25.0	1/3
3FH	3FH	32.0	32.0	64.0	15.6	1/2
7FH	7FH	64.0	64.0	120.0	7.8	1/2



5.1.4 Countermeasures Against Noise During Transmission (Carrier Output)

When the μPD17203A is used as the transmitter of a remote controller, a peak current of 0.5 to 1 A may flow through the infrared LED of the remote controller. Since two batteries are used as the power source of a remote controller, an equivalent resistance of several Ω (r) exists in the power source as shown in the figure below. This resistance reaches 10 to 20 Ω when the supply voltage drops to 2 V. Consequently, a high-frequency component noise will be superimposed on the power lines if the supply voltage fluctuates, especially in the case of switching, while the REM pin outputs the carrier frequency (the infrared LED lights).

Therefore, comply with the following points to minimize the adverse influence of the noise on the microcontroller:

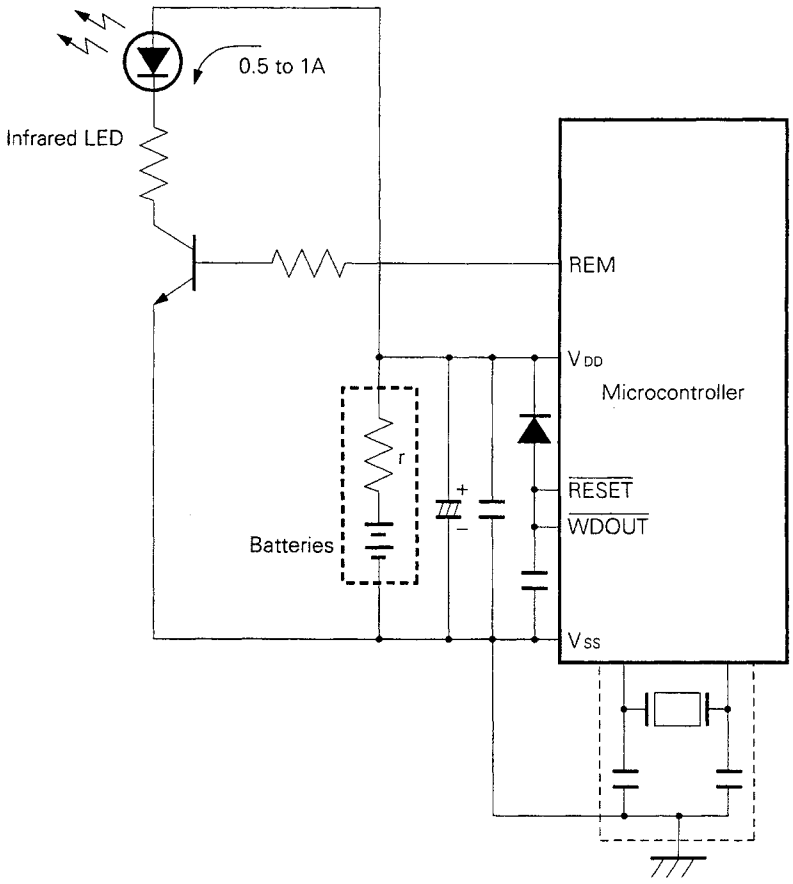
<1> Separate the power lines of the microcontroller from those of the infrared LED, starting from the terminals of the batteries. Keep the wiring length of the power lines as short as possible. Use thick power lines.

<2> Locate the oscillator as close to the microcontroller as possible. Shield the oscillator with the GND line (as indicated by the portion inside the dotted line in the figure below).

<3> Locate the capacitor for stabilizing the power supply as close to the power lines of the microcontroller as possible. Also select a capacitor that rejects high-frequency noise.

<4> While the carrier frequency is output, do not execute data read/write processing such as key scanning, an interrupt calling for stack, or CALL/RET instruction, to prevent the data from being changed.

<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). ★



Remarks 1. The INT and $\overline{\text{RESET}}$ pins are multiplexed with test pins (refer to **1.4 NOTES ON USING INT AND $\overline{\text{RESET}}$ PINS**). ★

2. In this figure, the $\overline{\text{RESET}}$ pin is connected to a pull-up resistor by mask option. ★

5.2 10-BIT TIMER 1

A 10-bit timer is mainly used to measure the carrier frequency of remote controller reception pulses together with the timer 0.

This 10-bit timer 1 consists of a 10-bit timer, a module register, and a comparator that compares the value of the modulo register with that of the timer.

The TM1CK0 and TM1CK1 bits on the register file are used to select the operating clock of the 10-bit timer. The TM1RES bit on the same file is used to reset the 10-bit timer and the TM1EN bit to start and stop the timer. The GET DBF, TM1C instruction is issued to read the 10-bit timer value via the data buffer (DBF). No value can be set in the 10-bit timer.

The PUT TM1M, DBF instruction is sued to set a value in the modulo register via the DBF. No data can be read from the modulo register. The same address is assigned to the modulo register and the 10-bit counter. Therefore, the modulo register is accessed when this address is written. When the address is read, the 10-bit counter is accessed.

When using timer 1, use the TM1K0 and TM1CK1 bits to select the operating clock, set a value in the modulo register via the DBF, and set the TM1RES bit to 1 to reset the 10-bit timer. To start timer 1, set the TM1EN bit to 0.

When the value of timer 1 coincides with that of the modulo register as a result of clock counting, an interrupt request occurs. If the interrupt enable flag (IPTM1) is set at this time, the interrupt function is executed and, at the same time, the 10-bit timer value is reset to 0. Counting is continued later. The interrupt request is output to IRQTM1 on the register file (see **10.2 HARDWARE OF INTERRUPT CONTROL CIRCUIT**).

The timer 1 outputs its signal to the TM1OUT pin, which is shared with the P1B2 pin. The TM1OE bit on the register file is used to specify $P1B_2$ as an I/O port or timer 1 output. To initialize the TM1OUT pin to 0 (low level), operate the TM1RES bit. After that, each time the value of the modulo register coincides with that of the 10-bit timer during timer 1 operation, the TM1OUT pin alternately goes low and high. Since it is of N-ch open-drain configuration, however, the TM1OUT pin goes into a high-impedance state when it outputs a high-level signal.

The register file for timer 1 is explained below.

Figure 5-5. Control Register for Timer 1 (1/2)

3	2	1	0	Address	When reset	R/W
TM1EN	TM1RES	TM1CK1	TM1CK0	RF : 34H	8H	R/W[Note]

TM1CK1	TM1CK0	10-bit timer clock source selection
0	0	Count clock: fsys/2 (Measurement time range: 0.5 μs 512 μs)
0	1	Count clock: fsys/16 (Measurement time range: 4μs to 4.096 ms)
1	0	Count clock: fsys/128 (Measurement time range: 32 μs to 32.768 ms)
1	1	INT pin input

Value indicated by parentheses is for when fsys (system clock) = fx = 4MHz

TM1RES	10-bit timer reset flag
0	Data read out is always "0"
1	Resets 10-bit counter and IRQTM1

★

TM1EN	10-bit timer count enable flag
0	Stops 10-bit timer count operation
1	Enable 10-bit timer count operation (at falling edge)

Note Bit 2 is write-only bit.

Figure 5-5. Control Register for Timer 1 (2/2)

3	2	1	0	Address	When reset	R/W
0	0	TM0 TM1EN	TM0 EXCK	RF : 14H	0H	R/W

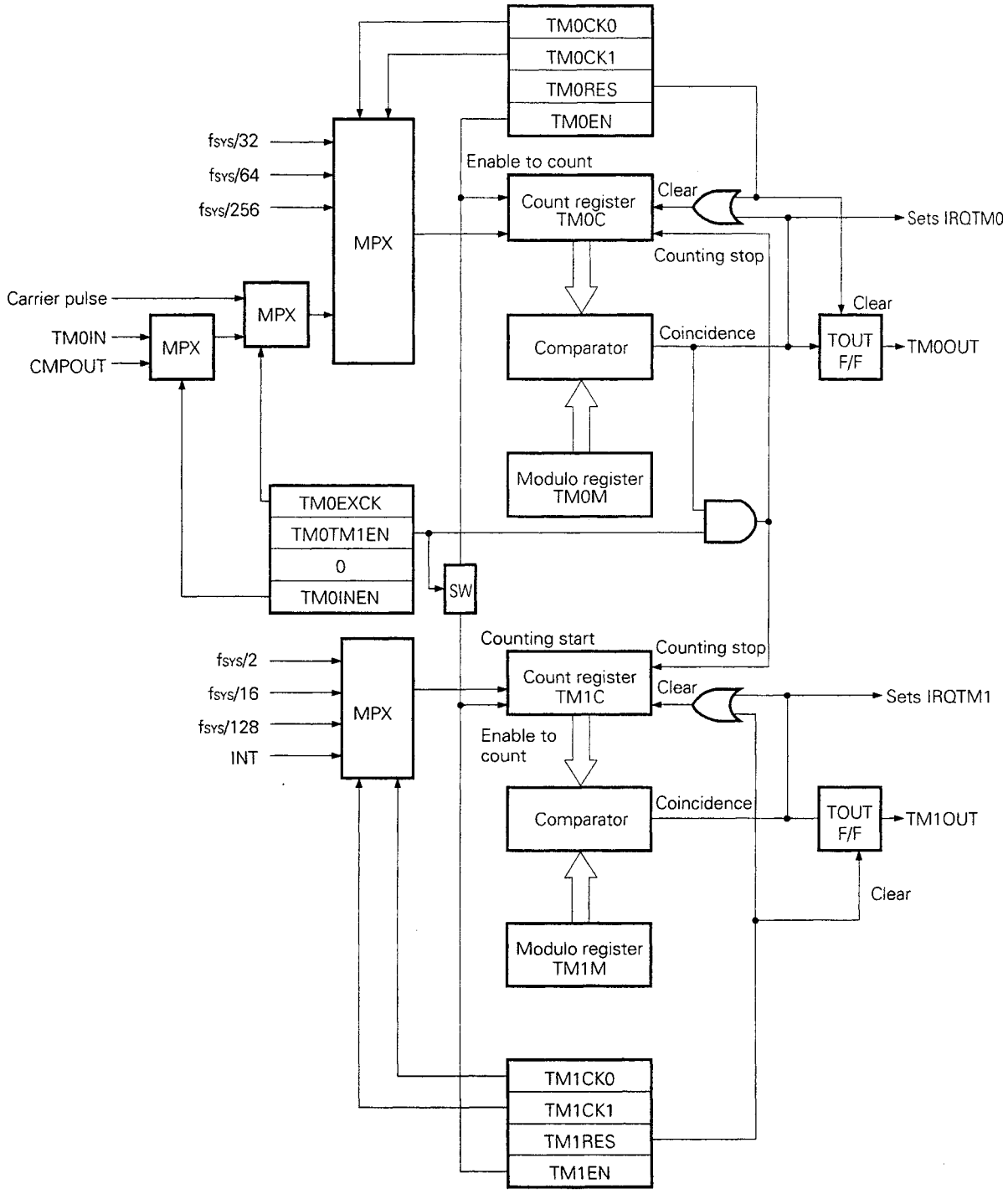
TM0 EXCK	Timer 0 external clock source selection
0	Uses internal remote control carrier generator clock as timer 0 input clock
1	Uses TM0IN input or CMPOUT output as timer 0 input clock

TM0 TM1EN	Tiemer 0, timer 1 start
0	Asynchronous timer 0 and timer 1 operates
1	Synchronous timer 0 and timer 1 operation

Cautions 1. If TM0TM1EN = 1, timers 0 and 1 will start in synchronization by enabling (setting) either of the timer 0 and timer 1 enable flags, TM0EN and TM1EN.
Moreover, timer 1 will be stopped by the coincides with the signal of timer 0. At that time, both TM0EN and TM1EN are stopped (reset).

★ **2. When TM0TM1EN = 1, be sure to set TM0EXCK and (TM0CK1, TM0CK0) = (1, 1) (won't operate with other settings).**
Also, timer 1 counts only while the envelope output (refer to 7.1.3 ENVELOPE CIRCUIT) is at high level.

Figure 5-6. Timer 0, Timer 1 Block Diagram



5.3 16-BIT TIMER 2

This timer is mainly used to measure the pulse width of the pulse received from a remote controller, with or without a carrier, with an envelope circuit output.

It consists only of a 16-bit timer. This timer has no modulo register.

The TM2CK0 and TM2CK1 bits on the register file are used to select the operating clock of the 16-bit timer. The TM2RES bit on the same file is used to reset the 16-bit timer and the TM2EN bit (RF: 35H, bit 3), to start and stop the timer. The GET instruction is issued to read the 16-bit timer value via the data buffer (DBF). No value can be set in the 16-bit timer. The address name of timer 2 is TM2C.

When using timer 2, use the TM2K0 and TM2CK1 bits to select the operating clock and set the TM2RES bit to 1 to reset the 16-bit timer. To start timer 2, set the TM2EN bit to 0.

The timer 2 outputs its signal to the TM2OUT pin, which is shared with the P1B3 pin. The TM2OE bit on the register file is used to specify $P1B_3$ as an I/O port pin or timer 2 output pin. To initialize the TM2OUT pin to 0 (low level), operate the TM2RES bit. After that, each time the 16-bit timer overflows, the TM2OUT pin goes alternately low and high. Since it is of N-ch open-drain configuration, however, the TM2OUT pin goes into a high-impedance state when it outputs a high-level signal.

The register file for timer 2 is explained below.

Figure 5-7. Control Register for Timer 2

3	2	1	0	Address	When reset	R/W
TM2EN	TM2RES	TM2CK1	TM2CK0	RF : 35H	8H	R/W[Note]

TM2CK1	TM2CK0	16-bit timer clock source selection
0	0	Count clock: $f_{SYS}/32$ (Measurement time range: 8 μs to 524.288 ms)
0	1	Count clock: $f_{SYS}/64$ (Measurement time range: 16 μs to 1.048576 s)
1	0	Count clock: $f_{SYS}/128$ (Measurement time range: 32 μs to 2.097152 s)
1	1	Count clock frequency: $f_{SYS}/128$ (Measurement time range: 64 μs to 4.194304 s)

Value indicated by parentheses is for when f_{SYS} (system clock) = f_X = 4MHz

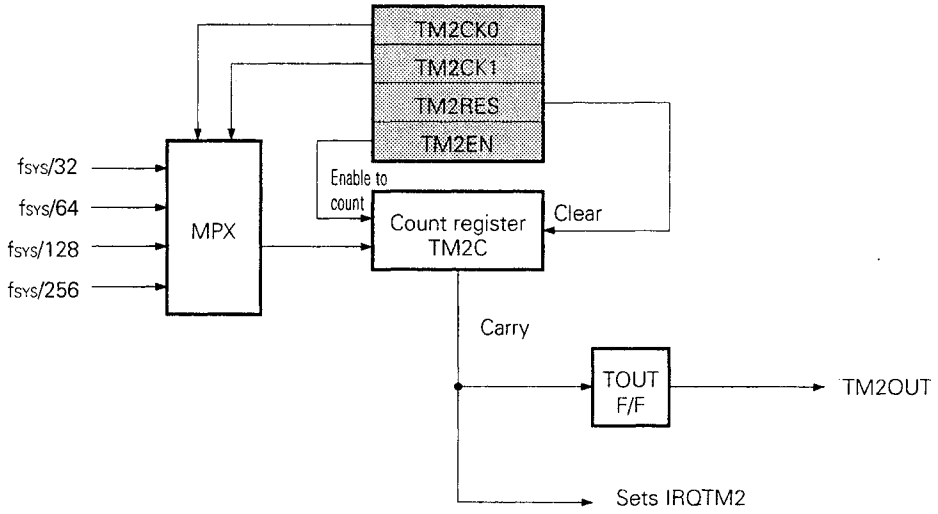
TM2RES	16-bit timer reset flag
0	Data read out is always "0"
1	Resets 16-bit counter and IRQTM2

★

TM2EN	16-bit timer count enable flag
0	Stops 16-bit timer count operation
1	Enable 16-bit timer count operation

Note Bit 2 is write-only bit.

Figure 5-8. Timer 2 Block Diagram



6. WATCH TIMER/WATCHDOG TIMER

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

6.1 CONFIGURATION OF WATCH TIMER/WATCHDOG TIMER

Figure 6-1 shows the configuration of the watch timer/watchdog timer.

As shown in this figure, the watch timer consists of selector A, which selects the 32.768-kHz oscillator circuit output of the subclock (XT) or the divided output ($f_X/2^7$) of the main clock (X) as the source clock, and selector B, which selects the divider of the selected source clock and the frequency to be used as an interrupt signal.

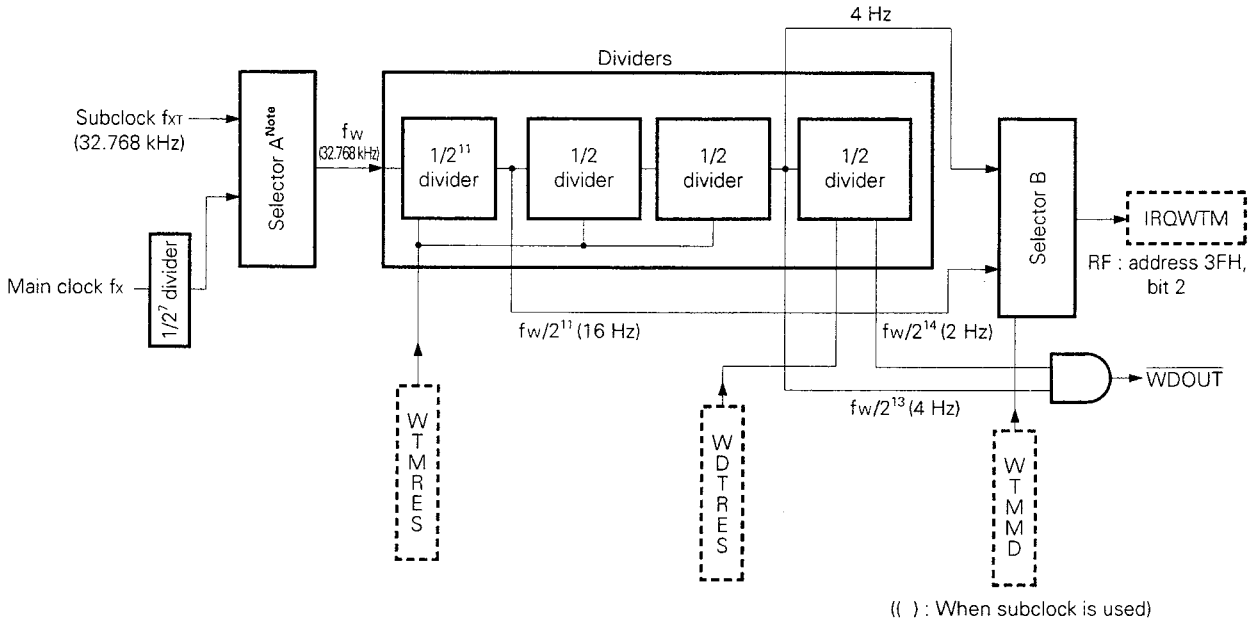
To reset the watch timer and control selector B, WTMRES (address 03H, bit 1) and WTMMD (address 03H, bit 2) are used.

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

When the subclock (f_{XT}) is used as the source clock, the counting operation of the watch timer cannot be stopped. Therefore, the subclock does not stop but continues oscillating even when the CPU is in the STOP mode.

If the divided output of the main clock ($f_X/2^7$) is used as the clock source (when subclock is used), the operation of the watch timer stops when the CPU enters the STOP mode.

Figure 6-1. Configuration of Watch Timer/Watchdog Timer



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

<1> To select use of subclock by mask option
Fixed to subclock.

<2> To not select use of subclock by mask option
Fixed to $f_X/2^7$.

6.2 FUNCTION OF WATCH TIMER/WATCHDOG TIMER

Figure 6-2. Control Register for Clock Timers

3	2	1	0	Address	When reset	R/W
WDTRES	WTMMD	WTMRES	0	RF : 03H	0H	R/W[Note]

WTMRES	Clock timer reset
0	Data read out is always "0"
1	Writing "1" resets 16-bit timer

WTMMD	Clock timer mode selection
0	Generates interrupt signal IRQWTM every $fw/2^{13}$ (4 Hz)
1	Generates interrupt signal IRQWTM every $fw/2^{11}$ (16 Hz)

(When source clock is 32.768 kHz)

WDTRES	Watchdog timer reset
0	Data read out is always "0"
1	Writing "1" resets watchdog timer

Note Bits 1 and 3 are write-only bits.

6.3 WATCHDOG TIMER OPERATION TIMING

Unless the watchdog timer is reset within a fixed time, a low level is output from the $\overline{\text{WDOUT}}$ pin.

By connecting the $\overline{\text{WDOUT}}$ and $\overline{\text{RESET}}$ pins, the watchdog timer can be used to monitor the hang-up of the program.

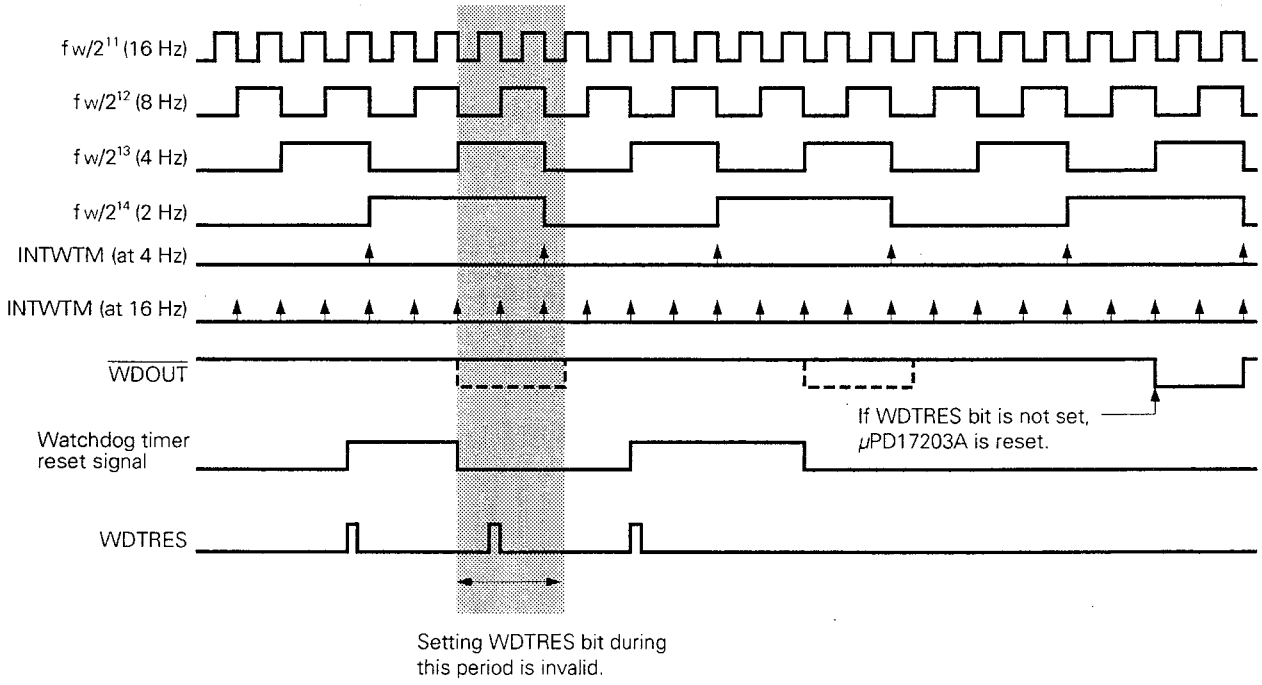
To reset the watchdog timer, set WDTRES (WDTRES = 1).

To disable hang-up detection by the watchdog timer when the subclock is used, create a program that sets WDTRES at a cycle of less than 340 ms.

Cautions 1. The watchdog timer cannot be reset within the shaded range in the above figure. Therefore, set WDTRES before both the $f_W/2^{14}$ (2-Hz) and $f_W/2^{13}$ (4-Hz) signals go high.

★ **2. For further information on the $\overline{\text{WDOUT}}$ pin, also refer to 12. RESET.**

Figure 6-3. Operation Timing of Watchdog Timer



(( ): when subclock is used)

7. CARRIER RECEIVER CIRCUIT FOR REMOTE CONTROLLER

7.1 ANALOG CIRCUIT, CLOCK NOISE REJECTER CIRCUIT, AND ENVELOPE CIRCUIT

The μPD17203A has been provided with circuits that can directly receive a carrier for a remote controller. The functions of these receiver circuits are listed in the following table:

Circuit	Function (f_{SYS} = f_X = 4 MHz)
Analog circuits (operational amplifier, comparator)	Amplifies received carrier for remote controller
Clock noise rejecter circuit	Shapes waveform of carrier signal (replaced with width of more than 4 μs)
Envelope circuit	Maintains a high level while carrier signal is received

7.1.1 Analog Circuit

The analog circuit contains an operational amplifier and a comparator to amplify the received carrier for a remote controller.

The degree of amplification by the operational amplifier can be changed by using the external pins AMPIN– and AMPOUT. To AMPIN+, 1/2V_{DD} is applied as a reference voltage. When the amplified output of the operational amplifier is input to CMPIN+ of the comparator, it is amplified to GND and V_{DD} level.

To CMPIN–, approximately 1/2V_{DD} is applied as a reference voltage.

To improve the reception accuracy, connect a capacitor and an external resistor to the V_{REF} pin to stabilize the reference voltage, depending on the level of the received carrier signal.

7.1.2 Clock Noise Rejecter Circuit

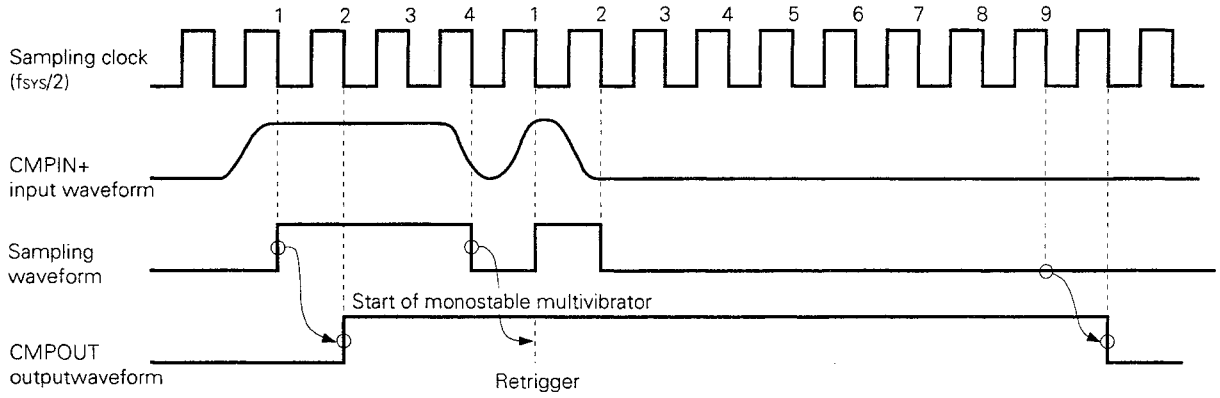
The clock noise rejecter circuit modifies the waveform of the signal amplified by the analog circuit.

The comparator output of the analog circuit is sampled by using the 1/2 cycle (f_{SYS}/2) clock of the system clock to modify the waveform.

The output level of the comparator is detected at the falling edge of the sampling clock. When the first high level of the clock is detected, a high level is output from the CMPOUT pin until the ninth sampling clock (f_{SYS} = f_X = 4 MHz, 4 μs) is detected.

While the high level is being output, if a high level is detected again after the low level of the comparator output has been detected once during the high-level output period, retriggering is executed. The high-level period is then extended to the ninth clock count from the falling edge of the retriggered clock.

Figure 7-1. Clock Noise Rejecter Circuit Output



7.1.3 Envelope Circuit

The envelope circuit modifies the waveform so that the high-level output can be maintained while the carrier for the remote controller is continuously received.

The output of the clock noise rejecter circuit or the input of the TM0IN pin (selected by TM0INEN) is sampled at 1/16 cycle of the system clock to shape the waveform.

The level is detected at the rising edge of the sampling clock. When the high level is detected, the high level is output for a fixed period (envelope output). This period can be set in four steps.

When fsys = fx = 4 MHz, 16, 32, 64, or 128 μs can be set as the maximum envelope output period by bits ENVCK0 and ENVCK1 of the register file.

To make the envelope output high continuously, keep the input waveform interval to within (set envelope output time – sampling clock time). When a pulse is directly input from the TM0IN pin, a high-level width less than 1/2 cycle of the system clock may not be detected.

Caution If the input level remains high, the envelope output is performed only once (retrigger is not executed). To make the envelope output high continuously, be sure to input a low level (more than fsys/2 cycle), and reset the envelope circuit so that retriggering is executed.

The envelope output level can be detected by the INTENV flag (RF: 0FH). Moreover, depending on the values of interrupt detect flags IEGENVM0 and IEGENVM1, the interrupt request flag (IRQENV) is reset at the edge set for detection.

Example of Envelope Waveform (when ENVCK1 = 1, ENVCK0 = 1)

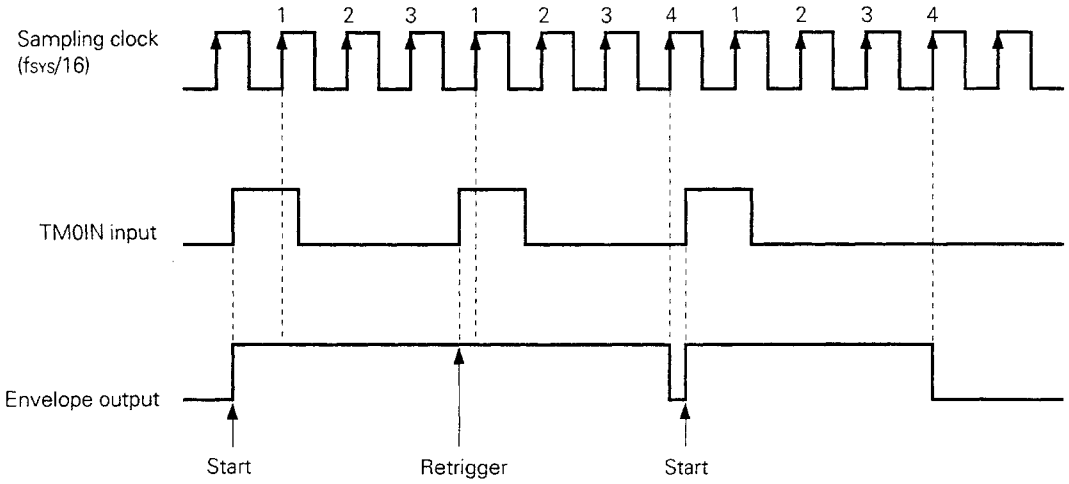


Figure 7-2. Envelope Time Control Register

3	2	1	0	Address	When reset	R/W
TM0 INEN	0	ENVCK1	ENVCK0	RF : 13H	0H	R/W

ENVCK1	ENVCK0	Envelope output time selection
0	0	64/fsys (16 $^{+0}_{-4}$ μs)
0	1	128/fsys (32 $^{+0}_{-4}$ μs)
1	0	256/fsys (64 $^{+0}_{-4}$ μs)
1	1	512/fsys (128 $^{+0}_{-4}$ μs)

Value indicated by parentheses is for when fsys (system clock) = fx = 4MHz

TM0 INEN	TM0IN pin input enable/disable
0	Uses internal comparator output clock as timer 0, envelope input
1	Uses TM0IN pin input as timer 0, envelope input

Figure 7-3. INT and INTENV Flags

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

INT	Detects level of INT pin
0	INT pin is at low level
1	INT pin is at high level

INTENV	Detects level of ENV signal
0	ENV signal is at low level
1	ENV signal is at high level

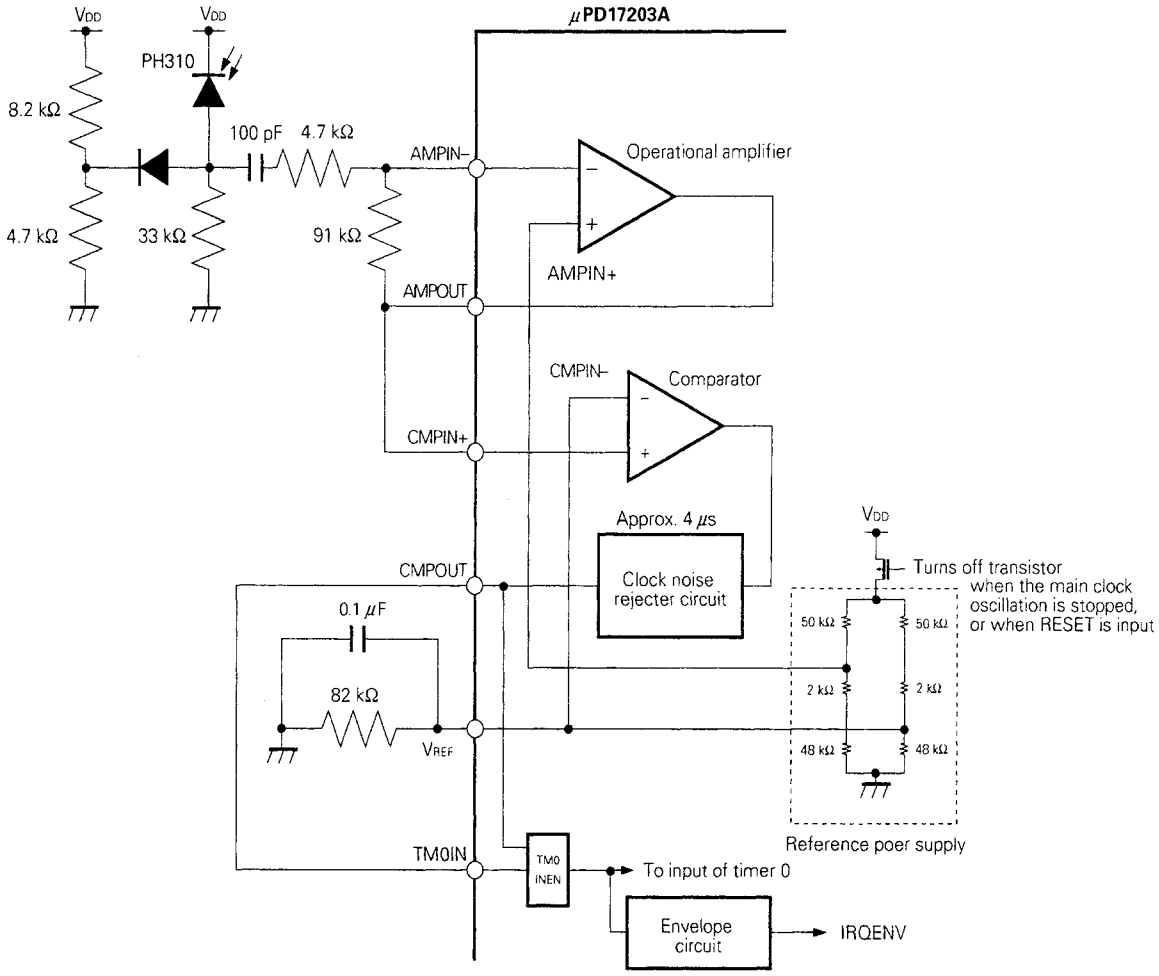
Figure 7-4. Interrupt Detection Edge Selector Flag

3	2	1	0	Address	When reset	R/W
0	IEG ENVM1	IEG ENVM0	IEG	RF : 1FH	0H	R/W

IEG	Selects edge of INT pin to be detected
0	Rising edge of INT pin
1	Falling edge of INT pin

IEG ENVM1	IEG ENVM0	Selects interrupt that occurs at each edge of ENV signal
0	0	Interrupt at rising edge of ENV signal
0	1	Interrupt at falling edge of ENV signal
1	0	Interrupt at both rising and falling edges of ENV signal
1	1	

Figure 7-5. Equivalent Circuit of μPD17203A Internal Preamplifier



Remark Assume that the error of the resistance in the reference power supply is –50 to +100% (relative error: ±5% max.).

8. SERIAL INTERFACE

The serial interface consists of an 8-bit shift register, a 4-bit shift mode register, and a 3-bit counter. This interface is used to input and output serial data.

8.1 SERIAL INTERFACE FUNCTIONS

8.1.1 Clock-Synchronous 8-Bit Transmission/Reception (Simultaneous Transmission and Reception)

Serial data input/output is controlled by the serial clock. The most significant bit (bit 7) of the shift register is output from the SO line at the falling edge of the serial clock ($\overline{SCK}$) signal. As soon as the contents of the shift register have been shifted by one bit at the rising edge of the SCK signal (nth bit → nth + 1 bit), data is loaded from the SI line to the least significant bit (bit 0).

The 3-bit counter (octal counter) counts the serial clock. Each time it counts the serial clock 8 times (end of 8-bit serial data transmission), the 3-bit counter issues a shift end signal and sets a interrupt request flag (IRQSIO).

8.1.2 Clock-Synchronous 8-Bit Reception (SO Output High-Impedance)

The clock-synchronous 8-bit reception operation is performed in the same manner as the clock synchronous 8-bit transmission/reception except that the SO pin goes into an the output high-impedance state and therefore outputs no serial data. For this reason, the SO pin can be used as a port pin to input data.

8.2 SERIAL INTERFACE OPERATION

8.2.1 Serial Interface Operation Mode

When the SIOEN bit is 1, $P1C_0$ to $P1C_3$ are set in serial interface mode. When SIOEN is 0, P1C0 to P1C3 are set in port mode. In port mode, no serial data is transmitted. Usually, the shift register can be used as an 8-bit register.

8.2.2 Serial Operation Mode

There are two types of serial operation modes: serial transmission mode and serial reception mode. When the SIOHIZ bit is set to 0, serial transmission mode is used. When it is set to 1, serial reception mode is used. Figure 8-1 shows the shift timing. The only difference between these two modes is whether serial data is output from the SO pin or the SO pin goes into a high-impedance state without serial data output.

To start serial data transmission, issue the PUT instruction to set send data in the shift register via the data buffer (DBF) and set the SIOTS bit to 1. When 8-bit data transmission terminates, the SIOTS bit is automatically reset to 0 and the IRQSIO flag is set to 1, causing an interrupt to occur. When the interrupt is disabled, it is confirmed that 8-bit data transmission is termi-nated by referencing the SIOTS bit or the IRQSIO flag.

Serial data reception operation is performed in the same manner as serial data transmission operation except for whether data is output from the SO pin.

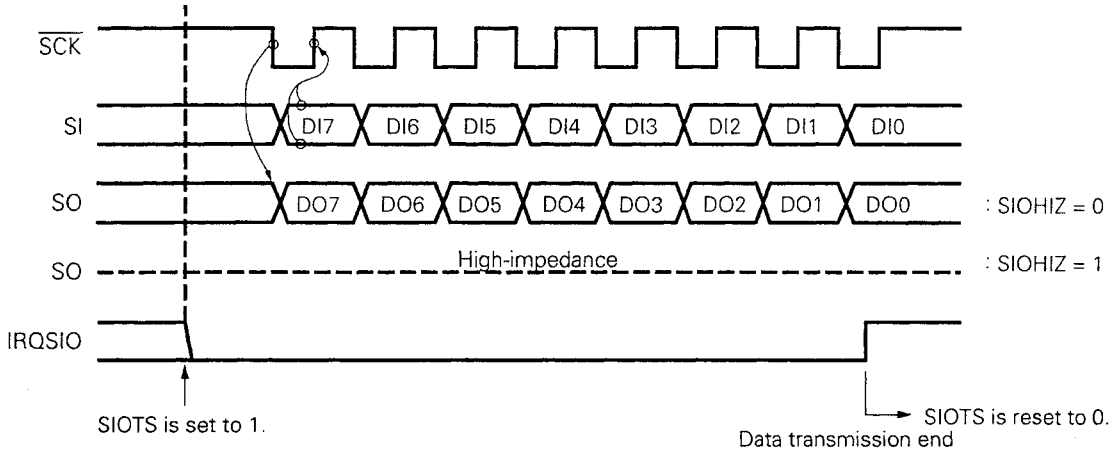
The SIOCK1 and SIOCK0 bits to select one of three internal clocks or an external clock as the serial clock source.

When an internal clock is selected and the SIOTS bit is set to 1, the clock is supplied to the serial interface and output from the $\overline{SCK}$ pin. During this period, serial data input/output is also controlled. When the serial clock is supplied 8 times, the SIOTS bit is automatically reset to 0 and clock supply to the serial interface is stopped. The $\overline{SCK}$ output is kept high and the interrupt request flag (IRQSIO) is set to 1.

When the external clock is selected and the SIOTS bit is set to 1, the clock from the $\overline{SCK}$ pin is supplied to the serial interface. When the external serial clock is supplied 8 times, the SIOTS bit is reset to 0, the IRQSIO flag is set to 1, and clock supply to the serial interface is stopped. When clock supply to the serial interface stops, serial data input/output also stops. When the SIOTS bit is set to 1, the IRQSIO flag is automatically reset to 0. For this reason, the program needs not reset the IRQSIO flag.

Serial data input/output can be forcibly stopped by resetting the SIOTS bit to 0. However, the stopped serial data input/output cannot be resumed because this resetting is forced termination of data input/output.

Figure 8-1. Shift Timing



Remark DI : Serial data input
DO : Serial data output

Figure 8-2. Shift Register Data Transfer

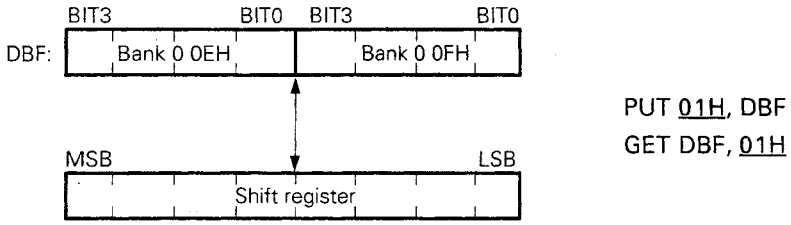


Figure 8-3. Serial I/O Control Register

3	2	1	0	Address	When reset	R/W
SIOTS	SIOHIZ	SIOCK1	SIOCK0	RF : 22H	0H	R/W

SIOCK1	SIOCK0	Serial clock selection
0	0	External clock
0	1	$f_{SYS}/16$
1	0	$f_{SYS}/128$
1	1	$f_{SYS}/1024$

f_{SYS} (system clock) = f_X or f_{XT}

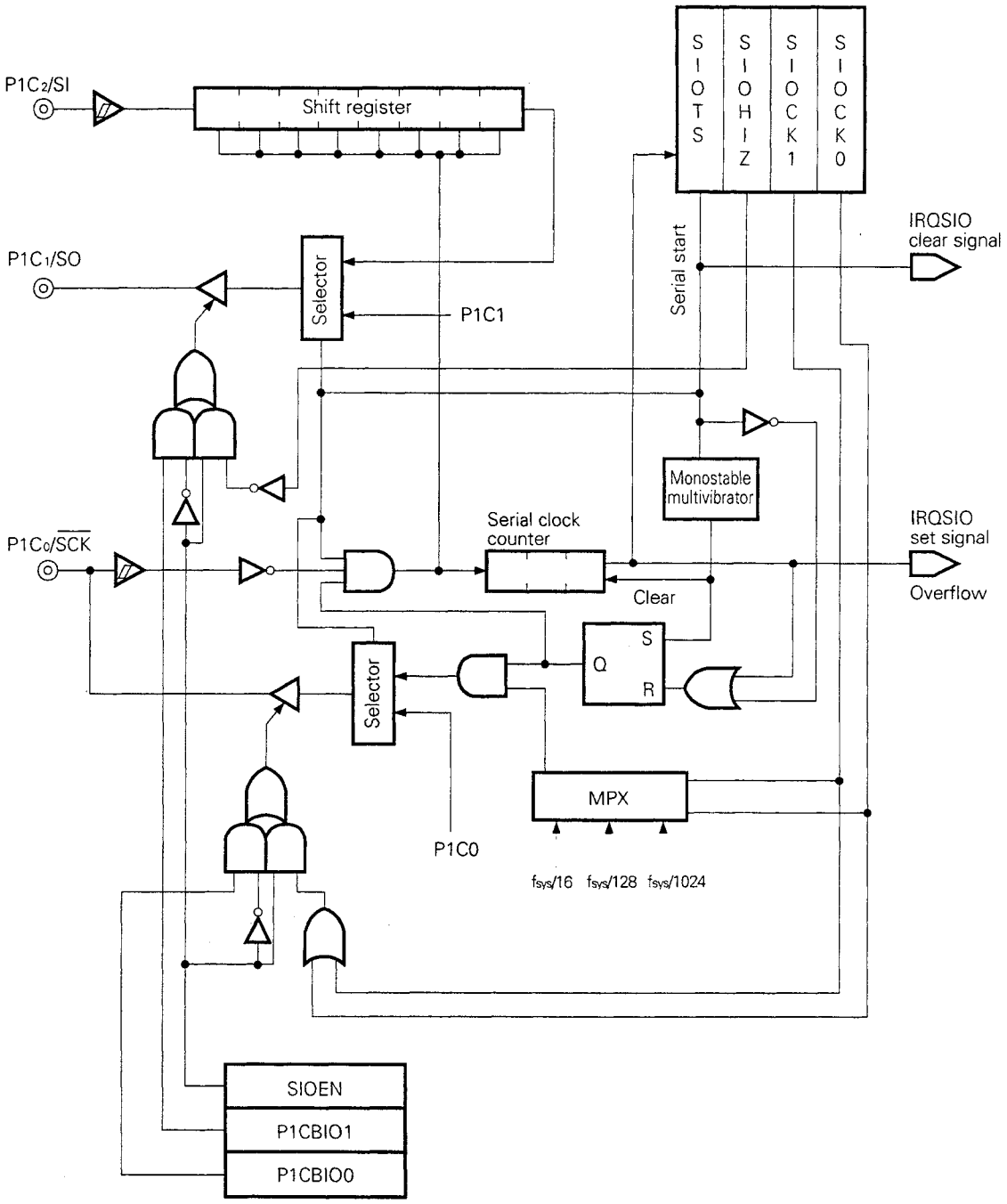
SIOHIZ	SO pin status
0	Serial data output
1	Outut buffer high impedance

SIOTS	Serial interface operation status (when reading)
0	Shift register shigt operation is disabled (stopped). Instruction to output from port 1C to $\overline{SCK}$ and SO is invalid. Logic value of pin can be read by port 1C input instruction.
1	Shift register is in shift operation by $\overline{SCK}$ clock. Instruction to output from $\overline{SCK}$ and SO to port 1C is invalid. Logic value of pin can be read by port 1C input instruction.

SIOTS	Serial interface operation status (when writing)
0	Forcibly stops serial data transfer. Resuming data transfer in the middle is not possible.
1	Can be set by PUT instruction only. Starts serial data transfer. Resets IRQSIO flag to "0". Automatically reset to "0" at the end of transfer.

Caution Serial clock source selection (setting the SIOCK0 and SIOCK bits) and shift start (setting the SIOTS bit to 1) must be performed at the same time.

Figure 8-4. Block Diagram of Serial Interface



9. STATIC RAM (XRAM)

The μPD17203A has been provided with 4096 × 4 bits of static RAM[Note 1].

To access this memory, an address is set on XRAMSTRT (12 bits) of the peripheral address or XRAMADR0-XRAMADR2 (BANK2, 70-72H) of the data memory through DBF. Data is read or written by accessing XRAM (BANK2, 73H). Each time XRAM has been accessed, the address is incremented.

When XRAMSTRT and XRAMSTP have coincided, accessing (reading/writing) XRAM further does not increment the address, but the data of address "XRAMSTP – 1" is accessed. When XRAM is accessed with XRAMSTRT (or XRAMADR0-XRAMADR2) being FFFH[Note 2], the address is incremented to 000H, and the operation continues.

If "an access end address + 1" is set in advance in XRAMSTP of the register file through DBF, interrupt signal IRQXRAM is generated when XRAMSTRT and XRAMSTP coincide.

Notes 1. μPD17203A : 2048 × 4 bits
2. μPD17204 : 7FFH

Caution Be sure to set XRAMSTP before accessing the static RAM.
If the value of XRAMSTRT is the same as that of XRAMSTP, interrupt signal IRQXRAM is set before XRAM is accessed.

Example 1. To clear static RAM
To clear SRAM from address 000H to 0FFH (XRAMSTP = 100H)

```
        MOV     DBF3, #0            ; Sets start address (000H)
        MOV     DBF2, #0            ;
        MOV     DBF1, #0            ;
        MOV     DBF0, #0            ;
        PUT     XRAMSTRT, DBF       ;

        MOV     DBF3, #0            ; Sets end address (100H)
        MOV     DBF2, #0            ; (end address = end address + 1)
        MOV     DBF1, #0            ;
        MOV     DBF0, #0            ;
        PUT     XRAMSTP, DBF        ;

        BANK2                       ; Sets bank 2

RAM_ CLEAR :
        MOV     XRAM, #0            ; Writes 0H to SRAM in specified range
        SKT1    IRQXRAM             ; Specified range (000H-0FFH) cleared?
        BR      RAM_ CLEAR          ; Yes
        ;       No

        BANK0                       ;
```

Example 2. To read static RAM

To read SRAM contents from address 130H to 13FH (XRAMSTP = 140H)

```
        WORK    MEM             0.00H

        MOV     DBF3, #0        ; Sets start address (130H)
        MOV     DBF2, #1        ;
        MOV     DBF1, #2        ;
        MOV     DBF0, #0        ;
        PUT     XRAMSTRT, DBF   ;

        MOV     DBF3, #0        ; Sets end address (140H)
        MOV     DBF2, #0        ; (end address = end address + 1)
        MOV     DBF1, #0        ;
        MOV     DBF0, #0        ;
        PUT     XRAMSTP, DBF    ;

        MOV     RPH, #0010B     ; Uses XRAM as register
        MOV     RPL, #1110B     ;

        MOV     IXH, #0         ; Sets index register
        MOV     IXM, #0         ;
        MOV     IXL, #0         ;

SRAM_ READ :
        SET1    IXE             ; Reads SRAM contents in specified range
        ST      WORK, XRAM      ; to work area (0.00H-0.0FH)
        CLR1    IXE             ;
        INC     IX              ; Increments index register

        SKT1    IRQXRAM         ; Read completed?
        BR      SRAM_ READ      ; No
                                ; Yes
        MOV     RPH, #0         ;
        MOV     RPL, #0         ;
```

Example 3. To write static RAM

To write 0AH to SRAM from address 200H to 20FH (XRAMSTP = 210H)

```
        MOV     DBF3, #0            ; Sets start address (200H)
        MOV     DBF2, #2            ;
        MOV     DBF1, #0            ;
        MOV     DBF0, #0            ;
        PUT     XRAMSTRT, DBF       ;

        MOV     DBF3, #0            ; Sets end address (210H)
        MOV     DBF2, #2            ; (end address = end address + 1)
        MOV     DBF1, #1            ;
        MOV     DBF0, #0            ;
        PUT     XRAMSTP, DBF        ;

        BANK2                       ; Sets bank 2

RAM_ WRITE :
        MOV     XRAM, #0AH          ; Writes 0AH to SRAM in specified range
★       SKT1    IRQXRAM             ; Specified range (200H-20FH) written?
        BR      RAM_ WRITE          ; Yes
                                    ; No
        BANK0                       ;
```

10. INTERRUPT FUNCTION

10.1 INTERRUPT SOURCES

There are eight interrupt sources.

When an interrupt is accepted, the program automatically branches to the address corresponding to the interrupt. This address is called a vector address. Table 10-1 shows the correspondence between the interrupt sources and vector addresses.

Table 10-1. Correspondence between Interrupt Sources and Vector Addresses

Priority	Interrupt Source	Internal/External	Vector Address
1	Rising or falling edge or both edges of envelope circuit output	Internal	8H
2	Timer 2	Internal	7H
3	Timer 0	Internal	6H
4	Timer 1	Internal	5H
5	Rising or falling edge of INT pin input	External	4H
6	Clock timer	Internal	3H
7	Serial input/output	Internal	2H
8	XRAM address	Internal	1H

When two or more interrupt requests are issued simultaneously, they are sequentially accepted starting from the one given the highest priority.

Accepting an interrupt is enabled or disabled by the EI and DI instructions, respectively. To accept an interrupt, the interrupt must be enabled by the EI instruction. While the DI instruction is executed or while an interrupt is accepted, another interrupt is disabled.

To enable an interrupt again after it has been completed, the EI instruction must be executed immediately before the RETI instruction. Accepting an interrupt is enabled by the EI instruction only after the next instruction has been executed; therefore, no interrupt is accepted between the EI and RETI instructions.

Caution In interrupt processing, automatic saving by hardware on the stack includes only the flags BCD, CMP, CY, Z, and IXE, to a maximum of 3 levels. Also, when accessing peripheral hardware (timer, serial interface, etc.) during interrupt processing, the content of DBF and WR are not saved by hardware. Therefore, we recommend saving DBF and WR by software in RAM at the beginning of interrupt processing and restoring the saved values just before the end of interrupt processing. ★

10.2 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 10-2. Interrupt Request Flags and Interrupt Enable Flags

Interrupt Request Flag	Signal Setting Interrupt Request Flag	Interrupt Enable Flag
IRQENV	Set at rising/falling edge of envelope circuit output	IPENV
IRQTM2	Timer 2	IPTM2
IRQTM0	Timer 0	IPTM0
IRQTM1	Timer 1	IPTM1
IRQ	Set when rising/falling edge of INT pin input signal are detected	IP
IRQWTM	Request signal generation interval from watch timer is selected by WTMMD flag (RF: 03H, bit 2)	IPWTM
IRQSIO	Reset by signal indicating end of serial data transfer operation from serial interface	IPSIO
IRQXRAM	Set by XRAM address coincidence	IPXRAM

10.2.1 INT

INT is a flag indicating the status of the INT pin.

When a high-level signal is input to the INT pin, the INT flag is set to 1. When a low-level signal is input, the INT flag is re-set to 0.

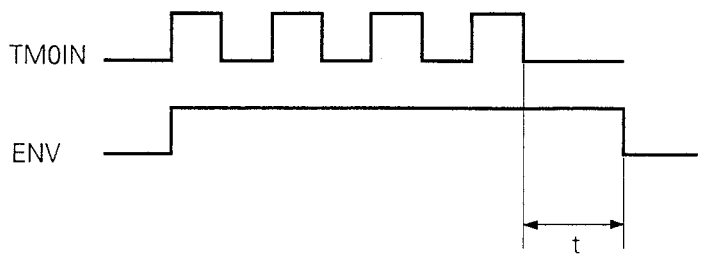
10.2.2 INTENV

INTENV is a flag indicating the output status of the internal envelope circuit (ENV signal).

When the ENV signal is high, the INTENV flag is set to 1. When it is low, the INTENV flag is reset to 0.

The ENV signal is obtained by widening the high-level width of a signal input to the TMOIN pin by the envelope time specified by ENVCK0 and ENVCK1.

Figure 10-1. Waveform of Internal Envelope Circuit Output



t: Envelope time specified by ENVCK0 and ENVCK1

Figure 10-2. INT and INTENV Flags

3	2	1	0	Address	When reset	R/W
0	0	INTENV	INT	RF : 0FH	undefined	R

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

INTENV	ENV signal level detection
0	ENV signal : Low level
1	ENV signal: High level

10.2.3 IEG

IEG is a flag that selects an edge at which an interrupt is to be detected on the INT pin.

When the IEG flag is reset to 0, the interrupt occurs at the rising edge of the INT pin. When it is set to 1, the interrupt occurs at the falling edge.

10.2.4 IEGENVM0 and IEGENVM1

IEGENVM0 and IEGENVM1 are flags that select the interrupt detecting edge of the ENV signal.

When the IEGENVM0 flag is reset to 0, with the IEGENVM1 flag being 0, the interrupt occurs at the rising edge of the ENV signal. When the IEGENVM0 flag is set to 1, with the IEGENVM1 flag being 0, the interrupt occurs at the falling edge.

When the IEGENVM1 flag is set to 1, the interrupt occurs at both the rising and falling edges of the ENV signal.

Figure 10-3. IEGENVM0 and IEGENVM1 Flags

3	2	1	0	Address	When reset	R/W
0	IEG ENVM1	IEG ENVM0	IEG	RF : 1FH	0H	R/W

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

IEG ENVM1	IEG ENVM0	ENV signal interrupt detection edge selection
0	0	Rising edge of ENV signal
0	1	Falling edge of ENV signal
1	0	Both rising and falling edges of ENV signal
1	1	

10.2.5 Interrupt Enable Flags

The interrupt enable flags specify whether to enable or disable the interrupts of the respective interrupt sources. When an interrupt enable flag is set to 1, the corresponding interrupt is enabled. When it is reset to 0, the interrupt is disabled.

Figure 10-4. Interrupt Enable Flags (1/2)

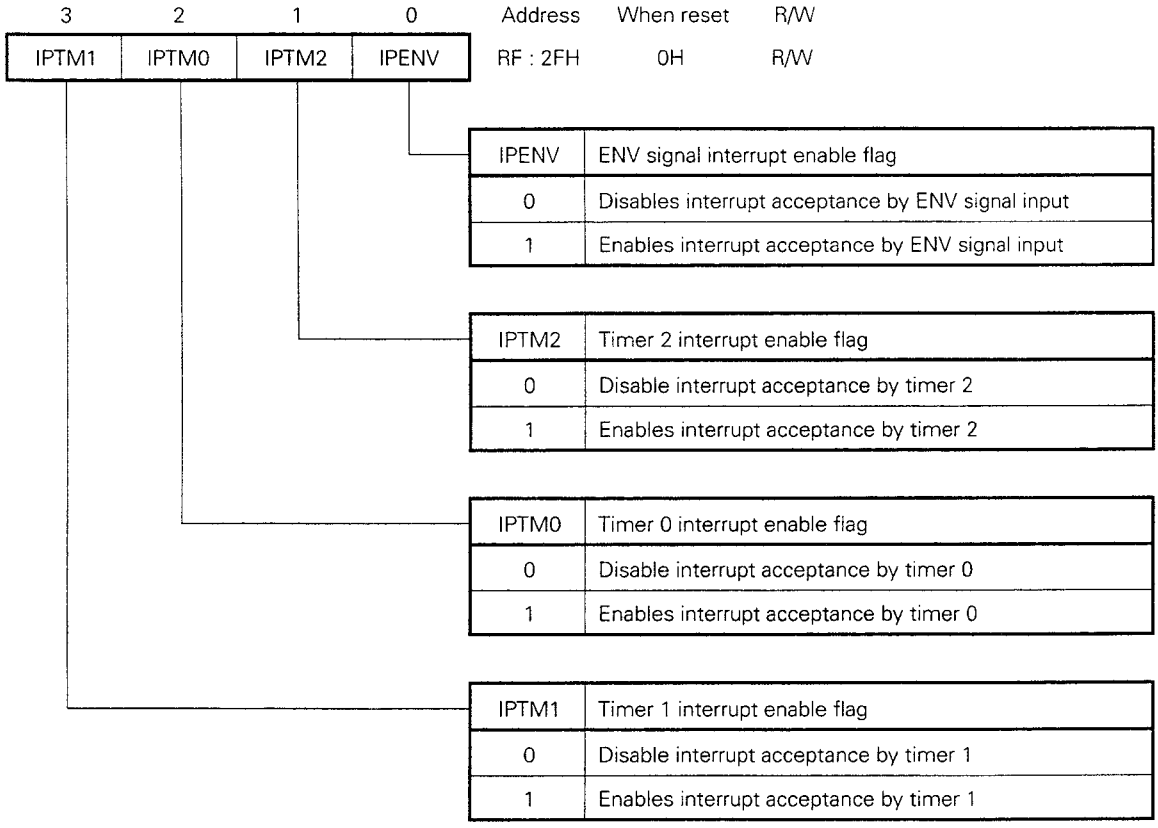


Figure 10-4. Interrupt Enable Flag (2/2)

3	2	1	0	Address	When reset	R/W
IPXRAM	IPSIO	IPWTM	IP	RF : 2EH	0H	R/W

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

IPWTM	Watch timer interrupt enable flag
0	Disable interrupt acceptance by watch timer
1	Enables interrupt acceptance by watch timer

IPSIO	Serial I/O interrupt enable flag
0	Disable interrupt acceptance by serial I/O
1	Enables interrupt acceptance by serial I/O

IPXRAM	XRAM interrupt enable flag
0	Disable interrupt acceptance by coincidence of XRAM address
1	Enables interrupt acceptance by coincidence of XRAM address

10.2.6 Interrupt Request Flag

IRQ is a flag indicating the status of an interrupt request.

When an interrupt request is made, the IRQ flag is set to 1. When the interrupt is accepted (occurs), the IRQ flag is reset to 0.

The IRQ flag can be read and written by the program. Writing 1 to this flag by the program enables software to issue an interrupt. Writing 0 enables software to release the interrupt pending status.

3	2	1	0	Address	When reset	R/W
0	0	0	IRQXRAM	RF : 38H	0H	R/W

IRQXRAM	Interrupt request flag of XRAM
0	No Interrupt request
1	Interrupt request generated by XRAM

3	2	1	0	Address	When reset	R/W
0	0	0	IRQSIO	RF : 39H	0H	R/W

IRQSIO	SIO interrupt request flag
0	No Interrupt request
1	Interrupt request generated by SIO

3	2	1	0	Address	When reset	R/W
0	0	0	IRQWTM	RF : 3AH	0H	R/W

IRQWTM	Watch timer interrupt request flag
0	No Interrupt request
1	Interrupt request generated by watch timer

3	2	1	0	Address	When reset	R/W
0	0	0	IRQ	RF : 3BH	0H	R/W

IRQ	INT pin interrupt request flag
0	No Interrupt request
1	Interrupt request generated by INT pin

3	2	1	0	Address	When reset	R/W
0	0	0	IRQTM1	RF : 3CH	0H	R/W

IRQTM1	Timer 1 interrupt request flag
0	No Interrupt request
1	Interrupt request generated by timer 1

3	2	1	0	Address	When reset	R/W
0	0	0	IRQTM0	RF : 3DH	0H	R/W

IRQTM0	Timer 0 interrupt request flag
0	No Interrupt request
1	Interrupt request generated by timer 0

3	2	1	0	Address	When reset	R/W
0	0	0	IRQTM2	RF : 3EH	0H	R/W

IRQTM2	Timer 2 interrupt request flag
0	No Interrupt request
1	Interrupt request generated by timer 2

3	2	1	0	Address	When reset	R/W
0	0	0	IRQENV	RF : 3FH	0H	R/W

IRQENV	ENV signal interrupt request flag
0	No Interrupt request
1	Interrupt request generated by ENV signal

10.3 INTERRUPT SEQUENCE

When the IRQ×× flag is set to "1" with the IP×× flag set to "1", interrupt processing is started as soon as the instruction cycle of the instruction executed is completed. Since the MOVT instruction requires two instruction cycles, if the IP×× flag is set to "1" while this instruction is executed, the interrupt processing is started as soon as the second instruction cycle is completed.

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

If two or more interrupts are enabled at the same time, the interrupt having the highest priority is processed and the interrupt(s) with the lower priority is kept pending until the processing of the interrupt with the highest priority has been completed.

10.3.1 Operation When Interrupt Is Accepted

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

(1) Resets the IRQ×× flag corresponding to the accepted interrupt to "0"
(2) Decrements the value of the stack pointer by one
(3) Saves the contents of the program counter to the address stack register (ASR) specified by the stack pointer
(4) Loads a vector address to the program counter
(5) Saves the contents of the bank register (BANK) and index enable flag (IXE) to the interrupt stack (only three levels of interrupt stack are available)

To execute the above sequence of processing, one instruction cycle is required.

10.3.2 Exiting from Interrupt Routine

To exit from an interrupt routine, the RETI instruction is used. Then the following processing is executed in an instruction cycle.

(1) Loads the contents of the address stack register (ASR) specified by the stack pointer to the program counter
(2) Loads the contents of the interrupt stack to the bank register and index enable flag
(3) Increments the value of the stack pointer by one

To enable an interrupt again after one interrupt has been processed, the EI instruction must be executed before the RETI instruction.

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

11. STANDBY FUNCTION

The μPD17203A is provided with the HALT mode and STOP mode as the standby function.

The current dissipation can be reduced by using the standby function.

In the HALT mode, the μPD17203A stands by, with the main clock not stopped, until the HALT mode releasing condition is satisfied. In this mode, the program is not executed.

In the STOP mode, the μPD17203A also stands by, but with the main clock stopped, until the STOP mode releasing condition is satis-fied. In this mode, the program is not executed, either.

When the HALT instruction is executed, the μPD17203A enters the HALT mode. When the STOP instruction is executed, the μPD17203A enters the STOP mode.

★ **Caution Do not execute an instruction to clear the interrupt request flag (IRQ×××) whose interrupt enable flag (IP×××) is set immediately before the HALT 8H instruction is executed. If the flag is cleared, the STOP mode may not be set.**

11.1 HALT MODE

HALT mode is used to temporarily stop program execution and reduce the current dissipation, with the main clock oscillated.

Use the HALT instruction to set the HALT mode.

The HALT mode releasing condition can be specified by the operand of the HALT instruction, as shown in Table 11-1.

Table 11-1. HALT Mode Releasing Condition

Operand Value	Releasing Conditions
0010B (02H)	When a TM0 (8-bit timer) interrupt request (IRQTM0) is issued (unrelated to IPTM0)
1000B (08H)	<1> When an interrupt request (IRQ×××, IRQ) is issued to the interrupt whose the interrupt enable flag (IP×××, IP) is set. <2> When any of the $P0A_0$ to $P0A_3$ and $P0B_0$ to $P0B_3$ pins goes low or output mode.
1010B (0AH)	<1> When a TM0 (8-bit timer) interrupt request (IRQTM0) is issued (unrelated to IPTM0) <2> When an interrupt request (IRQ×××, IRQ) is issued to the interrupt whose the interrupt enable flag (IP×××, IP) is set.
Other values	Setting inhibited

11.2 HALT INSTRUCTION EXECUTION CONDITION

The HALT and STOP instructions can be executed only under specific conditions to prevent the program from hang-up. Table 11-2 lists the HALT instruction execution conditions.

If any of these conditions is not satisfied, the HALT instruction is handled as an NOP instruction.

Table 11-2. HALT Instruction Execution Conditions

Operand Value	Execution Conditions
0010B (02H)	<1> The interrupt request flags (IRQTM0) of TM0 (8-bit timer) must be reset.
1000B (08H)	<1> Interrupt request flags (IRQ×××, IRQ) corresponding to interrupt whose interrupt enable flags (IP×××, IP) is set must be reset <2> All $P0A_0$-$P0A_3$ and $P0B_0$-$P0B_3$ pins must be high in input mode ★
1010B (0AH)	<1> Interrupt request flag of 8-bit timer TM0 (IRQTM0) must be reset <2> Interrupt request flags (IRQ×××, IRQ) corresponding to interrupt enable flag (IP×××, IP) must be set
Other values	Setting inhibited

11.3 STOP MODE

STOP mode is used to temporarily stop program execution and minimize the current dissipation, with the main clock oscillation stopped.

Use the STOP instruction to set STOP mode.

The STOP instruction is invalid in the system that operates only on the subclock. When the subclock is selected as the system clock, i.e., when SYSCK = 0, the STOP instruction is processed as NOP.

The STOP mode releasing condition can be specified by the operand of the STOP instruction, as shown in Table 11-4.

After STOP mode has been released, the μPD17203A performs the following processing:

<1> Cleares IRQTM0, IRQTM1, and IRQTM2.
<2> Starts clock timer and watchdog timer. (Not reset)
<3> Resets/starts timer/counters TM0, TM1, and TM2.
<4> When the value of the 8-bit timer/counter (TM0C) coincides with that of the modulo register (TM0M) (sets IRQTM0), executes the instruction next to the "STOP 8H" or a vector address branch instruction is executed.

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

The oscillation stabilization wait time that elapses until the next instruction is executed after "STOP 8H" has been released is calculated from the following expression where TM0M is the value of the modulo register.

(TM0M + 1) × 1024/fx [seconds] (fx: system clock frequency)

Example When the 4-MHz oscillator is used as the main clock, the time that elapses until the next instruction is executed after STOP has been released is:

(TM0M + 1) × 256 [μs]

Caution Do not execute an instruction to clear the interrupt request flag (IRQ×××) whose interrupt enable flag (IP×××) is set immediately before the HALT 8H instruction is executed. If the flag is cleared, the STOP mode may not be set. ★

Table 11-3. STOP Mode Release Condition

Operand Value	Releasing Conditions
1000B (08H)	<1> When an interrupt request (IRQENV, IRQTM1, IRQ, IRQWTM, IRQSIO) is issued to the interrupt whose the interrupt enable flag (IPENV, IPTM1, IP, IPWTM, IPSIO) is set. ★ <2> When any of the $P0A_0$ to $P0A_3$ and $P0B_0$ to $P0B_3$ pins goes low or output mode.
Other values	Setting inhibited

11.4 STOP INSTRUCTION EXECUTION CONDITION

The STOP instruction can be executed only under specific condi-tions to prevent the program from hang-up. Table 11-4 lists the STOP instruction execution conditions.

If any of these conditions is not satisfied, the STOP instruction is handled as an NOP instruction.

Table 11-4. STOP Instruction Execution Conditions

Operand Value	Releasing Conditions
1000B (08H)	<1> The interrupt request flag (IRQENV, IRQTM1, IRQ, IRQWTM, IRQSIO) must be reset for the interrupt whose the interrupt enable flag (IPENV, IPTM1, IP, IPWTM, IPSIO) is set. ★ <2> All the $P0A_0$ to $P0A_3$ and $P0B_0$ to $P0B_3$ pins must be high in input mode.
Other values	Setting inhibited

11.5 OPERATIONS AFTER STANDBY MODE HAS BEEN RELEASED

The operations are performed as follows when the STOP or HALT mode has been released:

Table 11-5. Operations after Standby Mode Has Been Released

(a) When HALT 08H, 0AH, or STOP 08H is executed

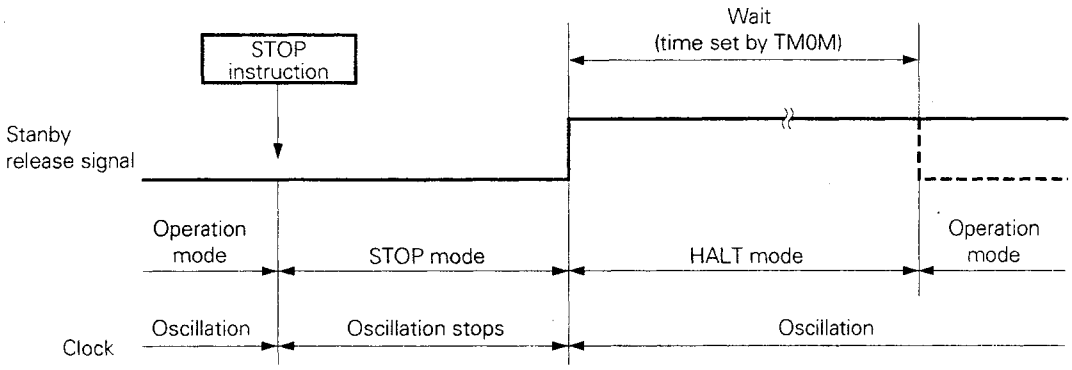
Standby Mode Releasing Conditions	Interrupt Enable Status	Interrupt Enable Flag	Operations After Standby Mode Has Been Released
Low-level input of $P0A_0$-$P0A_3$, $P0B_0$-$P0B_3$	Optional	–	Executes instruction next to STOP or HALT
Establishes releasing condition by interrupt request	DI	Disabled	Standby mode is not released
		Enabled	Executes instruction next to STOP or HALT
	EI	Disabled	Standby mode is not released
		Enabled	Branches to vector address of interrupt

(b) When HALT 02H is executed

Standby Mode Releasing Conditions	Interrupt Enable Status	Interrupt Enable Flag	Operations After Standby Mode Has Been Released
Setting of IRQTM0	DI	Optional	Executes instruction next to HALT
	EI		Branches to vector address of interrupt

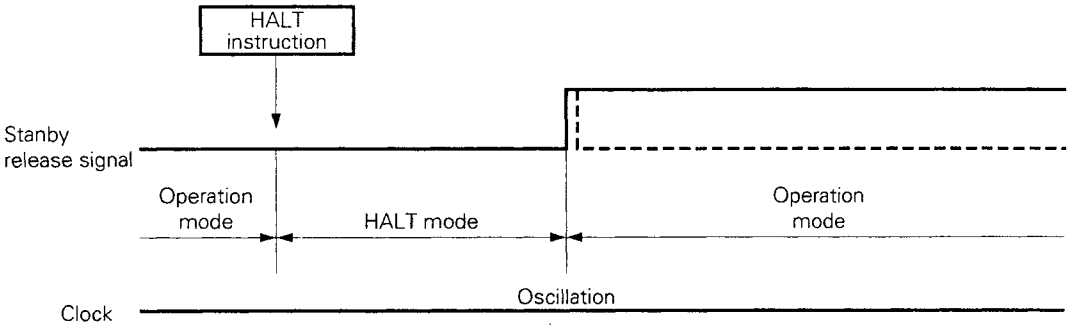
Figure 11-1. Operations after Standby Mode Has Been Released

(a) Releasing STOP mode by interrupt



Remark The dotted line indicates the case where the interrupt that has released the standby mode is accepted.

(b) Releasing HALT mode by interrupt



Remark The dotted line indicates the case where the interrupt that has released the standby mode is accepted.

12. RESET

12.1 RESET BY RESET SIGNAL INPUT

The μPD17203A is reset when a low-level signal is input to the $\overline{\text{RESET}}$ pin for more than 50 μs.

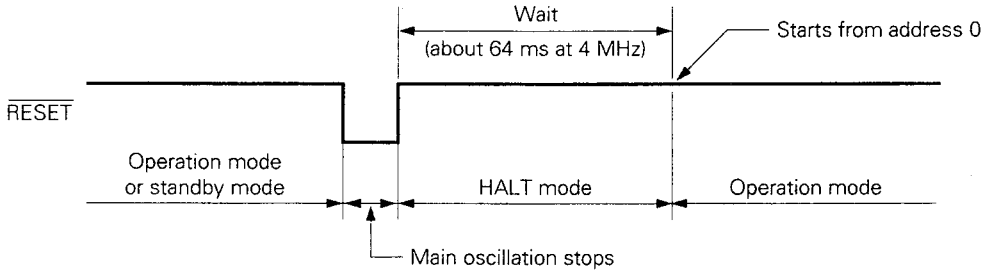
When the power is switched on, reset the μPD17203A at least once because the operations of the internal circuit are undefined.

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

<1> The program counter is reset to 0.
<2> The flags in the register file are initialized (for the initial values, refer to **Figure 13-1 Register File List**).
<3> Initial value 0320H is written to data buffer (DBF).
<4> The hardware peripherals are initialized.
<5> Oscillation of the main clock (X) is stopped.

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

Figure 12-1. Reset Operation by $\overline{\text{RESET}}$ Input



12.2 RESET BY WATCHDOG TIMER (CONNECTING $\overline{\text{RESET}}$ AND $\overline{\text{WDOUT}}$ PINS)

When the watchdog timer activates during program execution, a low level is output to the $\overline{\text{WDOUT}}$ pin, and the program counter is reset to 0.

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

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

12.3 RESET BY STACK POINTER (CONNECTING $\overline{\text{RESET}}$ AND $\overline{\text{WDOUT}}$ PINS)

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

If the level of nesting of an interrupt or subroutine call exceeds 5 (stack overflow), or if the return instruction is executed despite that the correspondence between a CALL instruction and return (RET) instruction is not established and that the stack level is 0 (stack underflow), the program can be restarted from address 0.

Table 12-1. Hardware Status after Reset ★

Hardware		RESET Input in Standby Mode	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 **Figure 13-1 Register File List.**	
8-bit Timer/Counter	Counter (TM0C)	00H	00H
	Modulo Register (TM0M)	FFH	FFH
10-bit Timer/Counter	Counter (TM1C)	00H	00H
	Modulo Register (TM1M)	FFH	FFH
16-bit Timer/Counter	Counter (TM2C)	00H	00H
Static RAM (XRAM)	XRAM Start Address Register (XRAMSTRT)	Undefined	Undefined
	XRAM Stop Address Register (XRAMSTP)	Undefined	Undefined
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)		
Shift Register of Serial Interface (SIOSFR)		Holds previous status	Undefined
Remote Controller Carrier Receive Circuit	Envelope Time Control Register (ENVCTR)	00H	00H
Counter of Watch Timer/Watchdog Timer		00H	00H

13. ASSEMBLER RESERVED WORDS

13.1 MASK OPTION DIRECTIVE

To code a μPD17203A program, the mask option directive must be used in the assembler source program to specify the mask option.

The following items need mask option specification:

- $P0A_0$, $P0A_1$, $P0A_2$, $P0A_3$
- $P0B_0$, $P0B_1$, $P0B_2$, $P0B_3$
- $\overline{RESET}$
- SYSTEM CLOCK

13.1.1 OPTION and ENDOP directives

The instructions between the OPTION and ENDOP directives are called a mask option definition block. The description format of the mask option definition block is given below.

Description format:

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

13.1.2 Mask option definition directives

Table 13-1 lists the mask option definition directives usable in the mask option definition block.

Figure 13-2 is an example of mask option definition.

Description format:

Symbol field	Mnemonic field	Operand field	Comment field
	OPTION		
	OPTRES	OPEN	; RESET pin has no pull-up resistor.
	OPTP0A	P0APLUP, P0APLUP, P0APLUP, P0APLUP	; All the 0A port pins have pull-up registors.
	OPTP0B	P0BPLUP, P0BPLUP, P0BPLUP, P0BPLUP	; All the 0B port pins have pull-up registors.
	OPTCK	USEX, NOXT	; The main clock is used, not the subclock.

Table 13-1. Mask Option Definition Pseudo Instructions

Name	Mask Option Definition Directive	Number of Operands	1st Operand	2nd Operand	3rd Operand	4th Operand
RESET	OPTRES	1	RESPLUP (w/pull-up resistor) OPEN (w/o pull-up resistor)			
P0A0-P0A3	OPTP0A	4	P0APLUP (w/pull-up resistor) OPEN (w/o pull-up resistor)	P0APLUP (w/pull-up resistor) OPEN (w/o pull-up resistor)	P0APLUP (w/pull-up resistor) OPEN (w/o pull-up resistor)	P0APLUP (w/pull-up resistor) OPEN (w/o pull-up resistor)
P0B0-P0B3	OPTP0B	4	P0BPLUP (w/pull-up resistor) OPEN (w/o pull-up resistor)	P0BPLUP (w/pull-up resistor) OPEN (w/o pull-up resistor)	P0BPLUP (w/pull-up resistor) OPEN (w/o pull-up resistor)	P0BPLUP (w/pull-up resistor) OPEN (w/o pull-up resistor)
SYSTEM CLOCK	OPTCK	2	USEX (main clock used) NOX (main clock not used)	USEXT (subclock used) NOXT (subclock not used)		

13.2 RESERVED SYMBOLS

Table 13-2 lists the reserved symbols defined in the μPD17203A device file.

The symbols defined in the device file represent the following register, port, and peripheral hardware names.

13.2.1 Control Registers on Register File

The names of the control registers assigned to data memory addresses 80H through BFH in bank 0 are defined. These registers are accessed by the PEEK or POKE instruction via the window register (WR).

Figure 13-1 shows the list of register file.

13.2.2 Registers and Ports on Data Memory

The names of the registers assigned to data memory addresses 00H through 7FH are defined. The names of the ports and system registers mounted in an area starting from address 70H are also defined.

Figure 13-2 shows the configuration of data memory.

13.2.3 Peripheral Hardware

The names of the peripheral hardware accessed by the GET and PUT instructions are defined.

Table 13-3 lists the peripheral hardware.

Table 13-2. Reserved Symbols (1/4)

Name	Attribute	Value	R/W	Description
DBF3	MEM	0.0CH	R/W	Bits 15 to 12 of data buffer
DBF2	MEM	0.0DH	R/W	Bits 11 to 8 of data buffer
DBF1	MEM	0.0EH	R/W	Bits 7 to 4 of data buffer
DBF0	MEM	0.0FH	R/W	Bits 3 to 0 of data buffer
★ XRAMADR2	MEM	2.70H	R/W	Bits 11 to 18 of XRAM address register
★ XRAMADR1	MEM	2.71H	R/W	Bits 7 to 4 of XRAM address register
★ XRAMADR0	MEM	2.72H	R/W	Bits 3 to 0 of XRAM address register
XRAM	MEM	2.73H	R/W	XRAM data
AR3	MEM	0.74H	R	Bits 15 to 12 of address register
AR2	MEM	0.75H	R/W	Bits 11 to 8 of address register
AR1	MEM	0.76H	R/W	Bits 7 to 4 of address register
AR0	MEM	0.77H	R/W	Bits 3 to 0 of address register
WR	MEM	0.78H	R/W	Window register
BANK	MEM	0.79H	R/W	Bank register
IXH	MEM	0.7AH	R/W	Bits 11 to 8 of index register
MPH	MEM	0.7AH	R/W	Bits 7 to 4 of memory pointer
MPE	FLG	0.7AH.3	R/W	Memory pointer enable flag
IXM	MEM	0.7BH	R/W	Bits 7 to 4 of index register
MPL	MEM	0.7BH	R/W	Bits 3 to 0 of memory pointer
IXL	MEM	0.7CH	R/W	Bits 3 to 0 of index register
RPH	MEM	0.7DH	R/W	Bits 7 to 4 of register pointer
RPL	MEM	0.7EH	R/W	Bits 3 to 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 enable flag
P0A0	FLG	0.70H.0	R/W	Bit 0 of port 0A
P0A1	FLG	0.70H.1	R/W	Bit 1 of port 0A
P0A2	FLG	0.70H.2	R/W	Bit 2 of port 0A
P0A3	FLG	0.70H.3	R/W	Bit 3 of port 0A
P0B0	FLG	0.71H.0	R/W	Bit 0 of port 0B
P0B1	FLG	0.71H.1	R/W	Bit 1 of port 0B
P0B2	FLG	0.71H.2	R/W	Bit 2 of port 0B
P0B3	FLG	0.71H.3	R/W	Bit 3 of port 0B
P0C0	FLG	0.72H.0	R/W	Bit 0 of port 0C
P0C1	FLG	0.72H.1	R/W	Bit 1 of port 0C
P0C2	FLG	0.72H.2	R/W	Bit 2 of port 0C
P0C3	FLG	0.72H.3	R/W	Bit 3 of port 0C

Table 13-2. Reserved Symbols (2/4)

Name	Attribute	Value	R/W	Description
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
P1B0	FLG	1.71H.0	R/W	Bit 0 of port 1B
P1B1	FLG	1.71H.1	R/W	Bit 1 of port 1B
P1B2	FLG	1.71H.2	R/W	Bit 2 of port 1B
P1B3	FLG	1.71H.3	R/W	Bit 3 of port 1B
P1C0	FLG	1.72H.0	R/W	Bit 0 of port 1C
P1C1	FLG	1.72H.1	R/W	Bit 1 of port 1C
P1C2	FLG	1.72H.2	R/W	Bit 2 of port 1C
P1C3	FLG	1.72H.3	R/W	Bit 3 of port 1C
SP	MEM	0.81H	R/W	Stack pointer
SYSCK	FLG	0.82H.1	R/W	System clock selection
XEN	FLG	0.82H.0	R/W	Permission of main clock oscillation
WDTRES	FLG	0.83H.3	R	Watchdog timer reset flag
WTMMD	FLG	0.83H.2	R/W	Selection of watch timer interrupt cycle
WTMRES	FLG	0.83H.1	R	Watch timer reset
INTENV	FLG	0.8FH.1	R	TM0IN pin status
INT	FLG	0.8FH.0	R	INT pin status
NRZBF	FLG	0.91H.0	R/W	NRZ data buffer
NRZ	FLG	0.92H.0	R/W	NRZ data
TM0INEN	FLG	0.93H.3	R/W	TM0IN permission
ENVCK1	FLG	0.93H.1	R/W	Envelope clock selection
ENVCK0	FLG	0.93H.0	R/W	Envelope clock selection
TM0TM1EN	FLG	0.94H.1	R/W	Timer 0 or 1 start
TM0EXCK	FLG	0.94H.0	R/W	Timer 0 external clock source selection flag
P1CBPU3	FLG	0.95H.3	R/W	Whether P1C3 pull-up resistor is incorporated or not
P1CBPU2	FLG	0.95H.2	R/W	Whether P1C2 pull-up resistor is incorporated or not
P1CBPU1	FLG	0.95H.1	R/W	Whether P1C1 pull-up resistor is incorporated or not
P1CBPU0	FLG	0.95H.0	R/W	Whether P1C0 pull-up resistor is incorporated or not
P1BBPU3	FLG	0.96H.3	R/W	Whether P1B3 pull-up resistor is incorporated or not
P1BBPU2	FLG	0.96H.2	R/W	Whether P1B2 pull-up resistor is incorporated or not
P1BBPU1	FLG	0.96H.1	R/W	Whether P1B1 pull-up resistor is incorporated or not
P1BBPU0	FLG	0.96H.0	R/W	Whether P1B0 pull-up resistor is incorporated or not

Table 18-3. Reserved Symbols (3/4)

Name	Attribute	Value	R/W	Description
P1ABPU3	FLG	0.97H.3	R/W	Whether P1A3 pull-up resistor is incorporated or not
P1ABPU2	FLG	0.97H.2	R/W	Whether P1A2 pull-up resistor is incorporated or not
P1ABPU1	FLG	0.97H.1	R/W	Whether P1A1 pull-up resistor is incorporated or not
P1ABPU0	FLG	0.97H.0	R/W	Whether P1A0 pull-up resistor is incorporated or not
IEGENVM1	FLG	0.9FH.2	R/W	Selection of TM0IN interrupt edge
IEGENVM0	FLG	0.9FH.1	R/W	Selection of TM0IN interrupt edge
IEG	FLG	0.9FH.0	R/W	Selection of INT interrupt edge
SIOTS	FLG	0.0A2H.3	R/W	SIO operation status
SIOHIZ	FLG	0.0A2H.2	R/W	SO pin status
SIOCK1	FLG	0.0A2H.1	R/W	Serial clock selection
SIOCK0	FLG	0.0A2H.0	R/W	Serial clock selection
SIOEN	FLG	0.0A3H.0	R/W	SIO output enable flag
TM2OE	FLG	0.0A4H.3	R/W	Timer 2 output enable flag
TM1OE	FLG	0.0A4H.2	R/W	Timer 1 output enable flag
TM0OE	FLG	0.0A4H.1	R/W	Timer 0 output enable flag
P1CBIO3	FLG	0.0A5H.3	R/W	P1C3 input/output selection
P1CBIO2	FLG	0.0A5H.2	R/W	P1C2 input/output selection
P1CBIO1	FLG	0.0A5H.1	R/W	P1C1 input/output selection
P1CBIO0	FLG	0.0A5H.0	R/W	P1C0 input/output selection
P1BBIO3	FLG	0.0A6H.3	R/W	P1B3 input/output selection
P1BBIO2	FLG	0.0A6H.2	R/W	P1B2 input/output selection
P1BBIO1	FLG	0.0A6H.1	R/W	P1B1 input/output selection
P1BBIO0	FLG	0.0A6H.0	R/W	P1B0 input/output selection
P1ABIO3	FLG	0.0A7H.3	R/W	P1A3 input/output selection
P1ABIO2	FLG	0.0A7H.2	R/W	P1A2 input/output selection
P1ABIO1	FLG	0.0A7H.1	R/W	P1A1 input/output selection
P1ABIO0	FLG	0.0A7H.0	R/W	P1A0 input/output selection
IPXRAM	FLG	0.0AEH.3	R/W	XRAM interrupt enable flag
IPSIO	FLG	0.0AEH.2	R/W	SIO interrupt enable flag
IPWTM	FLG	0.0AEH.1	R/W	Watch timer interrupt enable flag
IP	FLG	0.0AEH.0	R/W	INT interrupt enable flag
IPTM1	FLG	0.0AFH.3	R/W	Timer 1 interrupt enable flag
IPTM0	FLG	0.0AFH.2	R/W	Timer 0 interrupt enable flag
IPTM2	FLG	0.0AFH.1	R/W	Timer 2 interrupt enable flag
IPENV	FLG	0.0AFH.0	R/W	TM0IN interrupt enable flag

Table 18-2 Reserved Symbols (4/4)

Name	Attribute	Value	R/W	Description
TM0EN	FLG	0.0B3H.3	R/W	Timer 0 count enable flag
TM0RES	FLG	0.0B3H.2	R/W	Timer 0 reset flag
TM0CK1	FLG	0.0B3H.1	R/W	Timer 0 clock selection
TM0CK0	FLG	0.0B3H.0	R/W	Timer 0 clock selection
TM1EN	FLG	0.0B4H.3	R/W	Timer 1 count enable flag
TM1RES	FLG	0.0B4H.2	R/W	Timer 1 reset flag
TM1CK1	FLG	0.0B4H.1	R/W	Timer 1 clock selection
TM1CK0	FLG	0.0B4H.0	R/W	Timer 1 clock selection
TM2EN	FLG	0.0B5H.3	R/W	Timr 2 count enable flag
M2RES	FLG	0.0B5H.2	R/W	Timer 2 reset flag
TM2CK1	FLG	0.0B5H.1	R/W	Timer 2 clock selection
TM2CK0	FLG	0.0B5H.0	R/W	Timer 2 clock selection
P0DGIO	FLG	0.0B7H.3	R/W	Port 0D input/output selection
P0CGIO	FLG	0.0B7H.2	R/W	Port 0C input/output selection
P0BGIO	FLG	0.0B7H.1	R/W	Port 0B input/output selection
P0AGIO	FLG	0.0B7H.0	R/W	Port 0A input/output selection
IRQXRAM	FLG	0.0B8H.0	R/W	XRAM interrupt request flag
IRQSIO	FLG	0.0B9H.0	R/W	SIO interrupt request flag
IRQWTM	FLG	0.0BAH.0	R/W	Watch timer interrupt request flag
IRQ	FLG	0.0BBH.0	R/W	INT interrupt request flag
IRQTM1	FLG	0.0BCH.0	R/W	Timer 1 interrupt request flag
IRQTM0	FLG	0.0BDH.0	R/W	Timer 0 interrupt request flag
IRQTM2	FLG	0.0BEH.0	R/W	Timer 2 interrupt request flag
IRQENV	FLG	0.0BFH.0	R/W	TM0IN interrupt request flag
DBF	DAT	0FH	R/W	GET or PUT instruction operand
IX	DAT	01H	R/W	Index register
SIOSFR	DAT	01H	R/W	SIO shift register
TM0M	DAT	02H	W	Time 0 modulo register
TM0C	DAT	02H	R	Timer 0 count register
NRZLTMM	DAT	03H	R/W	Modulo register for setting NRZ low-level period
NRZHTMM	DAT	04H	R/W	Modulo register for setting NRZ high-level period
AR	DAT	40H	R/W	GET or PUT instruction operand
TM1M	DAT	41H	W	Timer 1 modulo register
TM1C	DAT	41H	R	Timer 1 count register
TM2C	DAT	42H	R	Timer 2 count register
XRAMSTRT	DAT	43H	R/W	XRAM start address setting register
XRAMSTP	DAT	44H	R/W	XRAM stop address detection register

Figure 13-1. Register File List (1/2)

Row Address \ Column Address		0	Note 1	1	Note 1	2	Note 1	3	Note 1	4	Note 1	5	Note 1	6	Note 1	7	Note 1
0 (8)	Bit 3			SP Note 2	0	0	0	WDTRES	0							0	0
	Bit 2				1	0	0	WTMMD	0							0	0
	Bit 1				0	SYSCK	*	WTMRES	0							0	0
	Bit 0				1	XEN	*	0	0							0	0
1 (9)	Bit 3			0	0	0	0	TM0INEN	0	0	0	P1CBPU3	0	P1BBPU3	0	P1ABPU3	0
	Bit 2			0	0	0	0	0	0	0	0	P1CBPU2	0	P1BBPU2	0	P1ABPU2	0
	Bit 1			0	0	0	0	ENVCK1	0	TM0TM1EN	0	P1CBPU1	0	P1BBPU1	0	P1ABPU1	0
	Bit 0			NRZBF	0	NRZ	0	ENVCK0	0	TM0EXCK	0	P1CBPU0	0	P1BBPU0	0	P1ABPU0	0
2 (A)	Bit 3					SIOTS	0	0	0	TM2OE	0	P1CBIO3	0	P1BBIO3	0	P1ABIU3	0
	Bit 2					SIOHIZ	0	0	0	TM1OE	0	P1CBIO2	0	P1BBIO2	0	P1ABIU2	0
	Bit 1					SIOCK1	0	0	0	TM0OE	0	P1CBIO1	0	P1BBIO1	0	P1ABIU1	0
	Bit 0					SIOCK0	0	SIOEN	0	0	0	P1CBIO0	0	P1BBIO0	0	P1ABIU0	0
3 (B)	Bit 3							TM0EN	1	TM1EN	1	TM2EN	1			P0DGIO	0
	Bit 2							TM0RES	0	TM1RES	0	TM2RES	0			P0CGIO	0
	Bit 1							TM0CK1	0	TM1CK1	0	TM2CK1	0			P0BGIO	0
	Bit 0							TM0CK0	0	TM1CK0	0	TM2CK0	0			P0AGIO	0

* When the main clock is selected (USEX) by the mask option, this bit is set to 1. When the main clock is not selected (NOX), this bit is reset to 0.

Notes 1. Status at reset

2. In case of the μPD17204, the reset value of SP is 0111B.

★ **Figure 13-2. Data Memory Configuration**

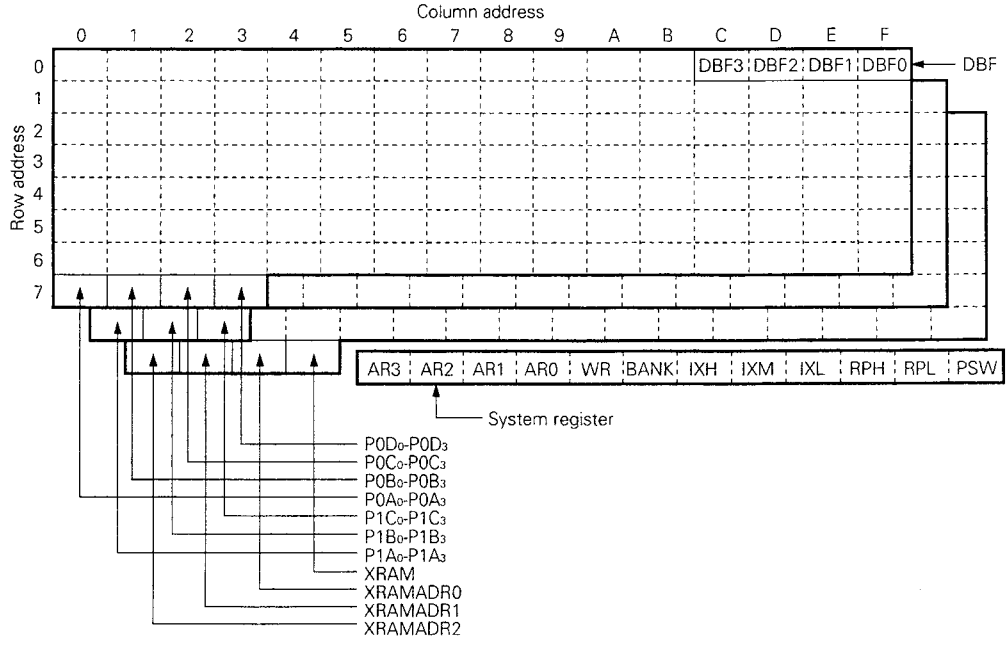


Figure 13-1. Register File List (2/2)

Row Address \ Column Address		8	Note	9	Note	A	Note	B	Note	C	Note	D	Note	E	Note	F	Note
0 (8)	Bit 3															0	0
	Bit 2															0	0
	Bit 1															INTENV	1
	Bit 0															INT	*
1 (9)	Bit 3															0	0
	Bit 2															IEGENVM1	0
	Bit 1															IEGENVM0	0
	Bit 0															IEG	0
2 (A)	Bit 3													IPXRAM	0	IPTM1	0
	Bit 2													IPSIO	0	IPTM0	0
	Bit 1													IPWTM	0	IPTM2	0
	Bit 0													IP	0	IPENV	0
3 (B)	Bit 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bit 0	IRQXRAM	0	IRQSIO	0	IRQWTM	0	IRQ	0	IRQTM1	0	IRQTM0	0	IRQTM2	0	IRQENV	0

* When INT pin is high level, this bit is set to 1; otherwise, it is set to 0.

Note Status at reset.

Table 13-3. Peripheral Hardware ★

Name	Address	Valid Bit	Remarks
SIOSFR	01H	8	Shift register of serial interface
TM0C	02H	8	Count register of 8-bit timer
TM0M	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
AR	40H	16	Address register
TM1C	41H	16	10-bit timer count register
TM1M	41H	16	10-bit timer modulo register
TM2C	42H	16	16-bit timer count register
XRAMSTRT	43H	16	XRAM start address setting register
XRAMSTP	44H	16	XRAM stop address setting register

14. INSTRUCTION SET

14.1 OUTLINE OF INSTRUCTION SET

b_{14}-b_{11} \ b_{15}		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 INC MOVT BR CALL RET RETSK EI DI RETI PUSH POP GET PUT PEEK POKE RORC STOP HALT NOP	AR IX DBF, @AR @AR @AR AR AR DBF, p p. DBF WR, rf rf, WR r s h 		
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	BR	addr (Page 2)	SKT	m, #n
1111	F	BR	addr (Page 3)	SKF	m, #n

14.2 LEGEND ★

AR	:	Address register
ASR	:	Address stack register specified by stack pointer
addr	:	Program memory address (lower 11 bits)
BANK	:	Bank register
COMP	:	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_R, m_C
m_R	:	Data memory row address (high)
m_C	:	Data memory column address (low)
n	:	Bit position (4 bits)
n4	:	Immediate data (4 bits)
PAGE	:	Page (Bit 11 of program counter)
PC	:	Program counter
p	:	Peripheral address
p_H	:	Peripheral address (higher 3 bits)
p_L	:	Peripheral address (lower 4 bits)
r	:	General register column address
rf	:	Register file address
rf_R	:	Register file address (higher 3 bits)
rf_C	:	Register file address (lower 4 bits)
SP	:	Stack pointer
s	:	Stop releasing condition
WR	:	Window register
(×)	:	Contents addressed by ×

14.3 LIST OF INSTRUCTION SETS

Group	Mnemonic	Operand	Operation	Instruction Code			
				OP code	Operand		
Addition	ADD	r, m	(r) ← (r) + (m)	00000	m_R	mc	r
		m, #n4	(m) ← (m) + n4	10000	m_R	mc	n4
	ADDC	r, m	(r) ← (r) + (m) + CY	00010	m_R	mc	r
		m, #n4	(m) ← (m) + n4 + CY	10010	m_R	mc	n4
	INC	AR	AR ← AR +1	00111	000	1001	0000
		IX	IX ← IX +1	00111	000	1000	0000
Subtraction	SUB	r, m	(r) ← (r) − (m)	00001	m_R	mc	r
		m, #n4	(m) ← (m) − n4	10001	m_R	mc	n4
	SUBC	r, m	(r) ← (r) − (m) − CY	00011	m_R	mc	r
		m, #n4	(m) ← (m) − n4 − CY	10011	m_R	mc	n4
Logical	OR	r, m	(r) ← (r) ∨ (m)	00110	m_R	mc	r
		m, #n4	(m) ← (m) ∨ n4	10110	m_R	mc	n4
	AND	r, m	(r) ← (r) ∧ (m)	00100	m_R	mc	r
		m, #n4	(m) ← (m) ∧ n4	10100	m_R	mc	n4
	XOR	r, m	(r) ← (r) ∀ (m)	00101	m_R	mc	r
		m, #n4	(m) ← (m) ∀ n4	10101	m_R	mc	n4
Judge	SKT	m, #n	CMP ← 0, if (m) ∧ n = n, then skip	11110	m_R	mc	n
	SKF	m, #n	CMP ← 0, if (m) ∧ n = 0, then skip	11111	m_R	mc	n
Compare	SKE	m, #n4	(m)-n4, skip if zero	01001	m_R	mc	n4
	SKNE	m, #n4	(m)-n4, skip if not zero	01011	m_R	mc	n4
	SKGE	m, #n4	(m)-n4, skip if not borrow	11001	m_R	mc	n4
	SKLT	m, #n4	(m)-n4, skip if borrow	11011	m_R	mc	n4
Rotate	RORC	r	CY → $(r)_{b3}$ → $(r)_{b2}$ → $(r)_{b1}$ → $(r)_{b0}$	00111	000	0111	r
Transfer	LD	r, m	(r) ← (m)	01000	m_R	mc	r
	ST	m, r	(m) ← (r)	11000	m_R	mc	r
	MOV	@r, m	if MPE = 1 : (MP, (r)) ← (m) if MPE = 0 : (BANK, m_R, (r)) ← (m)	01010	m_R	mc	r
		m, @r	if MPE = 1 : (m) ← (MP, (r)) if MPE = 0 : (m) ← (BANK, m_R, (r))	11010	m_R	mc	r
		m, #n4	(m) ← n4	11101	m_R	mc	n4
	MOVT	DBF, @AR	SP ← SP − 1, ASR ← PC, PC ← AR DBF ← (PC), PC ← ASR, SP ← SP + 1	00111	000	0001	0000

Group	Mnemonic	Operand	Operation		Instruction Code			
					OP code	Operand		
Transfer	PUSH	AR	SP ← SP – 1, ASR ← AR		00111	000	1101	0000
	POP	AR	AR ← ASR, SP ← SP +1		00111	000	1100	0000
	PEEK	WR, rf	WR ← (rf)		00111	rf_R	0011	rf_C
	POKE	rf, WR	(rf) ← WR		00111	rf_R	0010	rf_C
	GET	DBF, p	(DBF) ← (p)		00111	p_H	1011	p_L
	PUT	p, DBF	(p) ← (DBF)		00111	p_H	1010	p_L
Branch	BR	addr	μPD17203A	PC_{10-0} ← addr, PAGE ← 0	01100	addr		
				PC_{10-0} ← addr, PAGE ← 1	01101			
			μPD17204	PC_{10-0} ← addr, PAGE ← 0	01100			
				PC_{10-0} ← addr, PAGE ← 1	01101			
				PC_{10-0} ← addr, PAGE ← 2	01110			
				PC_{10-0} ← addr, PAGE ← 3	01111			
		@AR	PC ← AR		00111	000	0100	0000
Sub-routine	CALL	addr	SP ← SP – 1, ASR ← PC PC_{10-0} ← addr, PAGE ← 0		11100	addr		
		@AR	SP ← SP – 1, ASR ← PC PC ← AR		00111	000	0101	0000
	RET		PC ← ASR, SP ← SP +1		00111	000	1110	0000
	RETSK		PC ← ASR, SP ← SP +1 and skip		00111	001	1110	0000
	RETI		PC ← ASR, INTR ← INTSK, SP ← SP +1		00111	100	1110	0000
Interrupt	EI		INTEF ← 1		00111	000	1111	0000
	DI		INTEF ← 0		00111	001	1111	0000
Other	STOP	s	STOP		00111	010	1111	s
	HALT	h	HALT		00111	011	1111	h
	NOP		No operation		00111	100	1111	0000

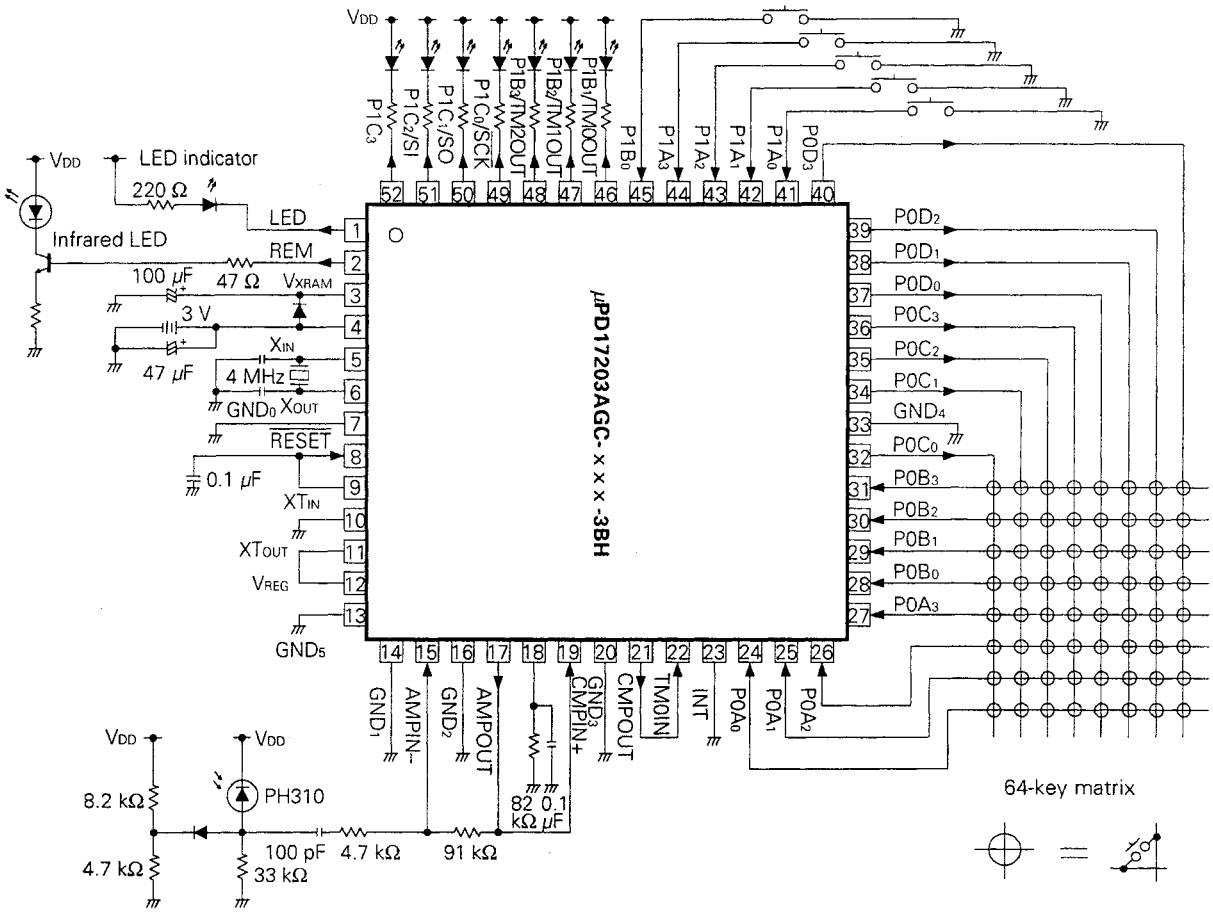
★ 14.4 ASSEMBLER (AS17K) EMBEDDED MACRO INSTRUCTION

Legend

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

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

15. APPLICATION CIRCUIT EXAMPLE



16. ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS (T_A = 25°C)

Item	Symbol	Condition		Rating	Unit
Supply Voltage	V_{DD}			−0.3 to +7.0	V
Input Voltage	V_I			−0.3 to V_{DD} + 0.3	V
Operating Ambient Temperature	T_A			−20 to +75	°C
Storage Temperature	T_{stg}			−40 to +125	°C
High-level Output Current	I_{OH1}	REM pin	Peak value	−30.0	mA
	I_{OH2}		Effective value	−20.0	mA
	I_{OH3}	1 pin (except for REM pin)	Peak value	−7.5	mA
	I_{OH4}		Effective value	−5.0	mA
	I_{OH5}	Total of all pins (except for REM pin)	Peak value	−22.5	mA
	I_{OH6}		Effective value	−15.0	mA
Low-Level Output Current	I_{OL1}	1 pin	Peak value	7.5	mA
	I_{OL2}		Effective value	5.0	mA
	I_{OL3}	Total of all pins	Peak value	30.0	mA
	I_{OL4}		Effective value	20.0	mA

Remark Effective value = Peak value × $\sqrt{\text{Duty}}$

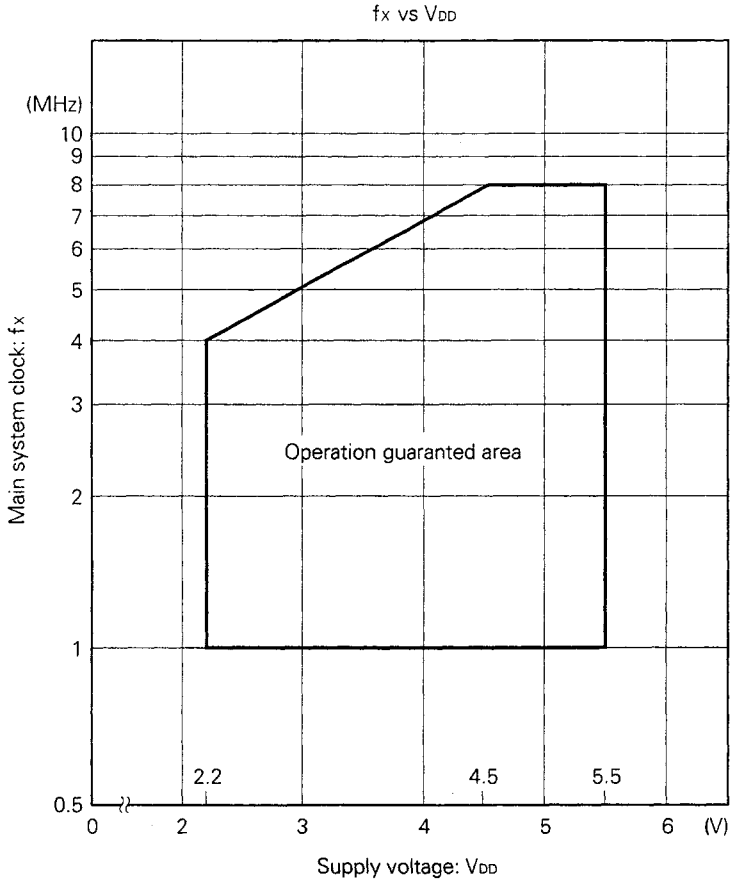
★ **Caution Even if one of the parameters exceeds its absolute maximum rating even momentarily, the quality of the product may be degraded. The absolute maximum rating therefore specifies the upper or lower limit of the value at which the product can be used without physical damages. Be sure not to exceed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

CAPACITANCE (T_A = 25°C, V_{DD} = 0V)

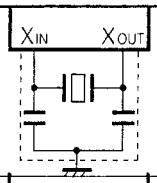
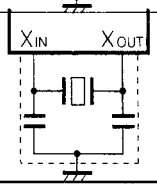
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input Capacitance	C_{IN}	INT and $\overline{\text{RESET}}$ pins			10	pF
	C_{PIN}	Otehr than INT and $\overline{\text{RESET}}$ pins			10	pF

RECOMMENDED OPERATING RANGE (T_A = −20 to +75°C)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Supply Voltage	V_{DD1}	System clock fx = 4 MHz	2.2	3.0	5.5	V
	V_{DD2}	System clock fx = 8 MHz	4.5	5.0	5.5	V
	V_{DD3}	System clock f_{XT} = 32.768kHz	2.0	3.0	5.5	V
Main Clock Oscillation Frequency	fx		1.0	4.0	8.0	MHz
Subclock Oscillation Frequency	f_{XT}			32.768		kHz



MAIN SYSTEM CLOCK OSCILLATOR CHARACTERISTICS (T_A = −20 to +75°C, V_{DD} = 2.2 to 5.5V)

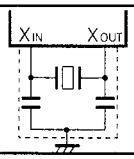
Resonator	Recomended Constants	Item	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic[Note 3] Oscillator		Oscillation frequency (f_X)[Note 1]		1.0	4.0	8.0	MHz
		Oscillation stabilization time[Note 2]	From when V_{DD} reaches the minimum oscillation voltage			4	ms
★ Crystal[Note 3] Oscillator		Oscillation frequency (f_X)[Note 1]		1.0	4.0	8.0	MHz
		Oscillation stabilization time[Note 2]	V_{DD} = 4.5 to 5.5 V			10	ms
						30	ms

Notes 1. The oscillation frequency is indicated onlyto express the oscillator characteristics. Refer to the AC characteristics for instruction execution time.

2. The oscillation stabilization time is the time required for stabilizing the oscillation after V_{DD} is applied or the STOP mode is released.

3. The recommended oscillators are shown in the table on the next page.

SUBSYSTEM CLOCK OSCILLATOR CHARACTERISTICS (T_A = −20 to +75°C, V_{DD} = 2.0 to 5.5V)

Resonator	Recomended Constants	Item	Conditions	MIN.	TYP.	MAX.	Unit
Crystal Oscillator		Oscillation frequency (f_{XT})			32.768		kHz
		Oscillation stabilization time				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 within the area enclosed by dotted lines 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

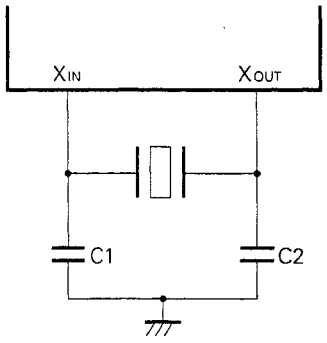
Main system clock: ceramic oscillator

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					
	CSA4.19MG					
	CST3.58MGW	Not required	Not required			C contained type
	CST4.00MGW					
	CST4.19MGW					
KYOCERA	KBR3.58MS	33	33	2.0	6.0	
	KBR4.0MS					
	KBR4.19MS					
TOKO	CRHF4.00	18	18	2.0	6.0	
DAISHINKU	PRS0400BCSAN	39	33	2.0	6.0	

Main system clock: crystal oscillator

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	

Oscillator Circuit



DC CHARACTERISTICS (T_A = −20 to +75°C, V_{DD} = 3 V, fx = 4 MHz, f_{XT} = 32.768 kHz)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
High-Level Input Voltage	V_{IH1}	$\overline{RESET}$ and INT pins		2.4		3.0	V
	V_{IH2}	Other than $\overline{RESET}$ and INT pins		2.1		3.0	V
Low-Level Input Voltage	V_{IL1}	$\overline{RESET}$ and INT pins		0		0.6	V
	V_{IL2}	Other than $\overline{RESET}$ and INT pins		0		0.9	V
High-Level Input Current	I_{IH1}	INT	V_{IH} = 3.0 V			0.2	μA
	I_{IH2}	TM0IN	V_{IH} = 3.0 V			0.2	μA
	I_{IH3}	$\overline{RESET}$	V_{IH} = 3.0 V			0.2	μA
	I_{IH4}	P0A-P0D	V_{IH} = 3.0 V			0.2	μA
	I_{IH5}	P1A-P1C	V_{IH} = 3.0 V			0.2	μA
Low-Level Input Current	I_{IL1}	INT	V_{IL} = 0 V			−0.2	μA
	I_{IL2}	TM0IN	V_{IL} = 0 V			−0.2	μA
	I_{IL3}	$\overline{RESET}$	V_{IL} = 0 V (pull-up resistor is not incorporated)			−0.2	μA
	I_{IL4}		V_{IL} = 0 V (pull-up resistor is incorporated)	−30	−60	−120	μA
	I_{IL5}	P0A,P0B	V_{IL} = 0 V (pull-up resistor is not incorporated)			−0.2	μA
	I_{IL6}		V_{IL} = 0 V (pull-up resistor is incorporated)	−8	−15	−30	μA
	I_{IL7}	P0C,P0D	V_{IL} = 0 V			−0.2	μA
	I_{IL8}	P1A-P1C	V_{IL} = 0 V (pull-up resistor is not incorporated)			−0.2	μA
	I_{IL9}		V_{IL} = 0 V (pull-up resistor is incorporated)	−30	−60	−120	μA
High-Level Output Current	I_{OH1}	P0A,P0B	V_{OH} = 2.7 V	−0.6	−2.0	−4.0	mA
	I_{OH2}	P1C	V_{OH} = 2.7 V	−0.6	−2.0	−4.0	mA
	I_{OH3}	REM	V_{OH} = 1.0 V	−7.0	−15.0	−25.0	mA
	I_{OH4}	LED	V_{OH} = 2.7 V	−0.3	−1.0	−2.0	mA
	I_{OH5}	CMPOUT	V_{OH} = 2.7 V	−0.3	−1.0	−2.0	mA
Low-Level Output Current	I_{OL1}	P0A,P0B, P1C	V_{OH} = 0.3 V	0.5	1.5	2.5	mA
	I_{OL2}	P0C,P0D,P1B	V_{OH} = 0.3 V	0.5	1.5	2.5	mA
	I_{OL3}	REM	V_{OL} = 0.3 V	0.5	1.5	2.5	mA
	I_{OL4}	LED	V_{OL} = 0.3 V	0.5	1.5	2.5	mA
	I_{OL5}	CMPOUT	V_{OL} = 0.3 V	0.5	1.5	2.5	mA
	I_{OL6}	P1A	V_{OL} = 0.3 V	1.5	4.5	7.5	mA
V_{REF} Output Voltage	V_{REF}	C = 0.1 μF, R = 82 kΩ		0.8	1.1	1.6	V
Supply Current	I_{DD1}	Operation mode	Both XT and X oscillate.	0.5	1.0	2.0	mA
	I_{DD1}		Only XT oscillates.		15	30	μA
	I_{DD3}	HALT mode	Both XT and X oscillate.			2.0	mA
	I_{DD4}[Note]		Only XT oscillates		7	15	μA
XRAM Holding voltage	V_{XRAM}			1.3	3.0	5.5	V
XRAM Supply Current	I_{XRAM1}	Operation mode, V_{XRAM} = 3 V		3.0	5.0	7.0	μA
	I_{XRAM2}	HALT mode, V_{XRAM} = 3 V, T_A = 25°C			0.2	1.0	μA

Note The specifications of the main STOP mode (sub mounting) are the same as those of the sub HALT mode (main oscillation stops).

OPERATIONAL AMPLIFIER/COMPARATOR CHARACTERISTICS

(T_A = −20 to +75°C, V_{DD} = 3 V, f_X = 4 MHz, f_{XT} = 32 kHz)

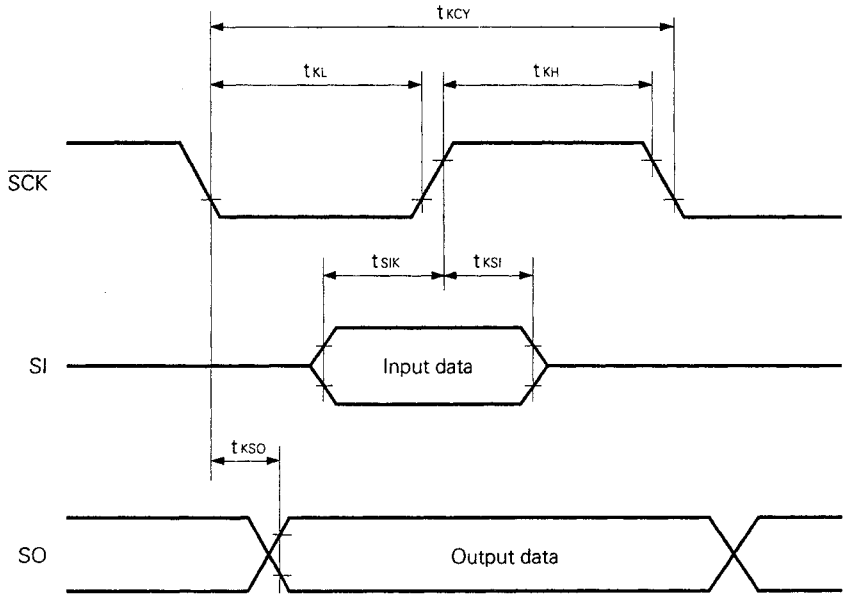
Item	Symbol	Test Condition	MIN.	TYP.	MAX.	Unit
Unity Gain Frequency of Operational Amplifier			0.5	1.0	5	MHz
Input Offset Voltage of Operational Smplifier				20		mV
Same Phase Input Voltage Range of Operational Amplifier			0.3		2.7	V
Output Voltage Range of Operational Amplifier			0.1		2.9	V
Operational Amplifier Through Rate			1			V/μs
Input Offset Voltage of Comparator			40	60	80	mV
Same Phase Input Voltage Range of Comparator			0		3.0	V
Minimum Output Pulse Width of Comparator			3	4	5	μs

AC CHARACTERISTICS (T_A = −20 to +75°C, V_{DD} = 3 V) ★

Parameter	Symbol	Test Condition		MIN.	TYP.	MAX.	Unit
$\overline{SCK}$ Input Cycle Time	t_{KCY}		Data Input	5.0			μs
			Data Output	13.0			μs
$\overline{SCK}$ Input High- and Low-Widths	t_{KH}, t_{KL}		Data Input	2.5			μs
			Data Output	6.5			μs
SI Setup Time (vs. $\overline{SCK}$↑)	t_{SIK}			100			ns
SI Hold Time (vs. $\overline{SCK}$↑)	t_{KSI}			100			ns
SCK↓→to SO Output Delay Time	t_{KSO}	C_L=100 pF				4.5	μs
INT High/Low-Level Width	t_{IOH} t_{IOL}			50			μs
$\overline{RESET}$ Low-Level Width	t_{RSL}			50			μs

SERIAL TRANSFER TIMING

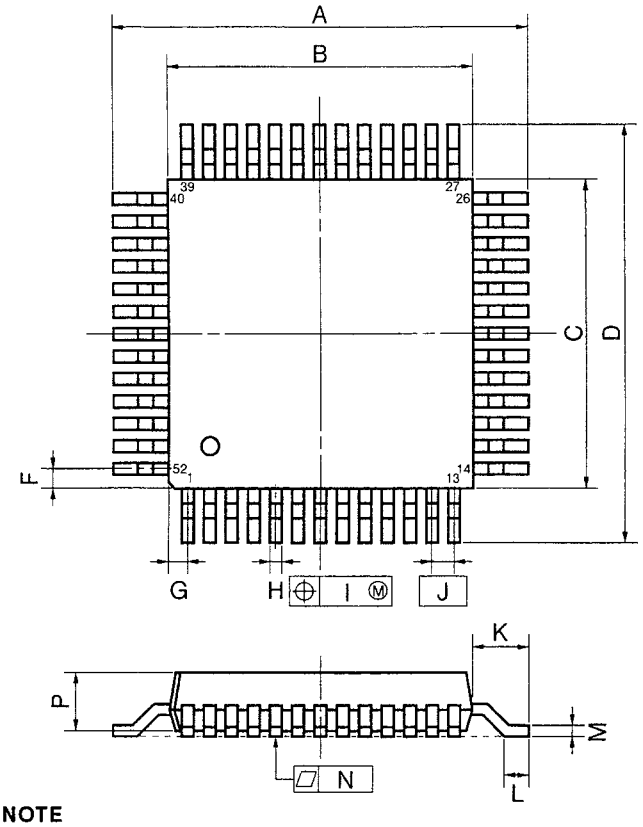
3-line Serial I/O Mode



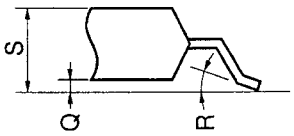
17. PACKAGE DRAWINGS

PACKAGE DRAWINGS OF MASS-PRODUCTION PRODUCT

52 PIN PLASTIC QFP (□14)



detail of lead end



NOTE

Each lead centerline is located within 0.20 mm (0.008 inch) of its true position (T.P.) at maximum material condition.

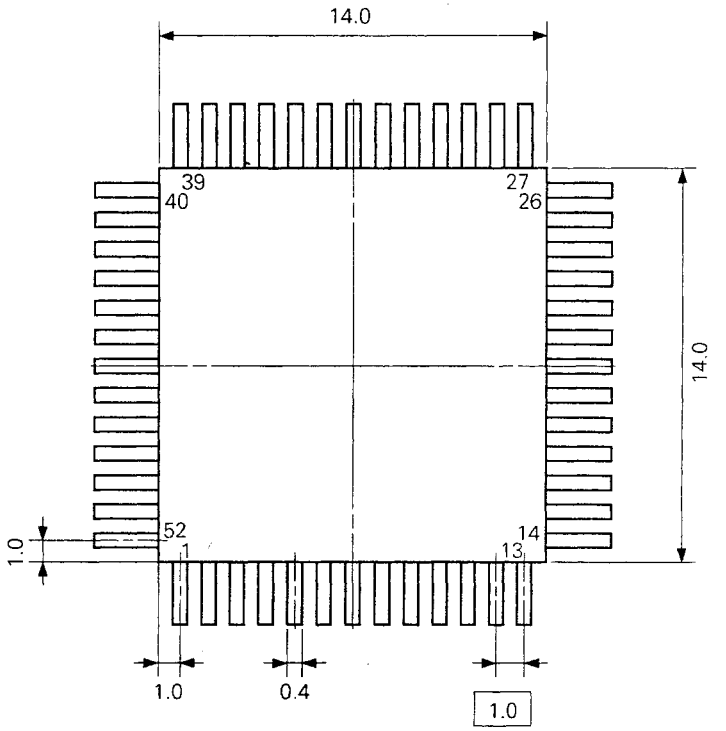
ITEM	MILLIMETERS	INCHES
A	17.2±0.2	0.677±0.008
B	14.0±0.2	$0.551^{+0.009}_{-0.008}$
C	14.0±0.2	$0.551^{+0.009}_{-0.008}$
D	17.2±0.2	0.677±0.008
F	1.0	0.039
G	1.0	0.039
H	0.40±0.10	$0.016^{+0.004}_{-0.005}$
I	0.20	0.008
J	1.0 (T.P.)	0.039 (T.P.)
K	1.6±0.2	0.063±0.008
L	0.8±0.2	$0.031^{+0.009}_{-0.008}$
M	$0.15^{+0.10}_{-0.05}$	$0.006^{+0.004}_{-0.003}$
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.

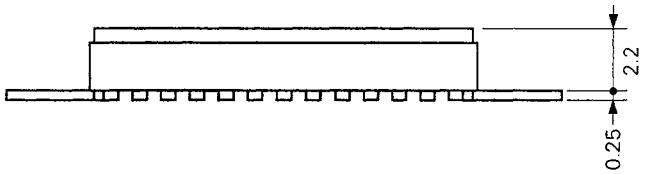
S52GC-100-3BH-2

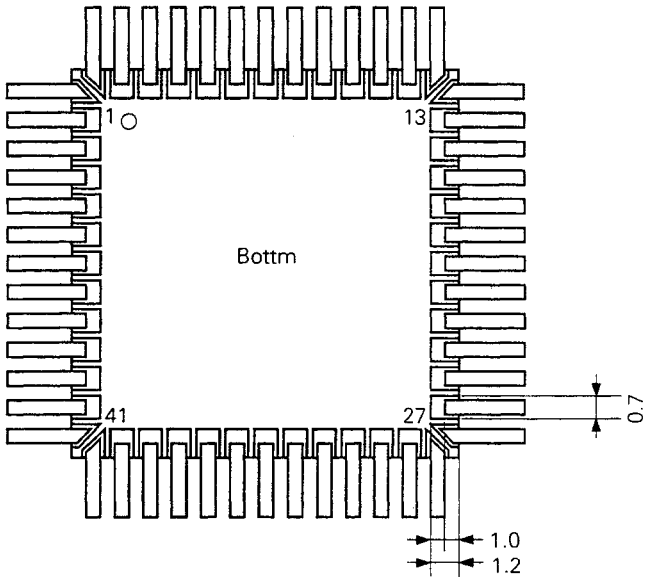
★ **Caution The ES and mass-production products differ in external shape and materials. Please refer to the package drawing for the ES product.**

PACKAGE DRAWINGS OF ES PRODUCT

52-PIN CERAMIC QFP (14 × 14) (Unit: mm)







Caution The lead length is not rated because of the lead tip cutting process.

X52B-100B

18. RECOMMENDED SOLDERING CONDITIONS

When mounting the μPD17203A and 17204 by soldering, soldering should be performed under the following recommended conditions.

For details on recommended soldering conditions refer to the information document **Semiconductor Device Mounting Technology Manual (IEI-1207)**.

For other soldering methods, please consult with NEC sales personnel.

Table 18-1. Soldering Conditions of Surface Mount Type

***m*PD17203AGC-×××-3BH : 52-pin plastic QFP (14 × 14 mm)**
***m*PD17204GC-×××-3BH : 52-pin plastic QFP (14 × 14 mm)**

Soldering Method	Soldering Conditions	Recommended Conditions Reference Code
Infrared reflow	Package peak temperature: 230°C, Time: 30 seconds max. (210°C min.), Number of soldering operations: 1, Maximum number of days[Note]: 2 days (beyond this period, 16 hours of pre-baking is required at 125°C)	IR30-162-1
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (200°C min.), Number of soldering operations: 1, Maximum number of days[Note]: 2 days (beyond this period, 16 hours of pre-baking is required at 125°C)	VP15-162-1
Wave soldering	Solder bath temperature: 260°C max., Time: 10 seconds max. Number of soldering operations: 1, Pre-heating temperature: 120°C max. (package surface temperature) Maximum number of days[Note]: 2 days (beyond this period, 16 hours of pre-baking is required at 125°C)	WS60-162-1
Pin partial heating	Pin temperature: 300°C max., Timer: 3 seconds max.(per side)	–

Note Number of days after unpacking the dry pack. Storage conditions are 25°C and 65%RH max.

Caution Do not use different soldering methods together (however, pin partial heating can be performed with other soldering methods).

APPENDIX A. LIST OF MICROCONTROLLERS FOR LEARNING REMOTE CONTROLLER

Item / Product Name		μPD17203A	μPD17P203A	μPD17204	μPD17P203A
ROM Capacity		8K bytes (4096 × 16)		16K bytes (7936 × 16)	
		(Mask ROM)	(One-time PROM)	(Mask ROM)	(One-time PROM)
RAM Capacity		336 × 4 bits			
Static RAM Capacity		4096 × 4 bits		2048 × 4 bits	
Infrared Remote Controller Carrier Generator Circuit		Provided			
Infrared Remote Controller Receive Preamplifier		Provided			
I/O Port		28			
External Interrupt (INT)		1			
Timer		Four channels { 8-bit timer: 3 channels; Watch timer: 1 channel			
Watchdog Timer		Provided ($\overline{\text{WDOUT}}$ output)			
Serial Interface		1 channel			
Stack		5 levels (up to 3 levels for multiplexed interrupt)		7 levels (up to 3 levels for multiplexed interrupt)	
Standby Function		STOP mode, HALT mode			
Instruction Execution Time (Supply Voltage) T_A = −20 to +75°C	Main System Clock	4 μs : @4MHz			
		(V_{DD} = 2.2 to 5.5 V)	(V_{DD} = 2.9 to 5.5 V[Note])	(V_{DD} = 2.2 to 5.5 V)	(V_{DD} = 2.9 to 5.5 V[Note])
	Sub System Clock	488 μs : @32.768 kHz (V_{DD} = 2.0 to 5.5 V)			
Package		52-pin plastic QFP			

Note The supply voltage varies with the operating ambient temperature. For details, refer to **16. ELECTRICAL SPECIFICATIONS.**

APPENDIX B. DEVELOPMENT TOOLS

The following tools are available for development of μPD17203A progams:

Hardware

Name	Outline
In-circuit Emulators (IE-17K, IE-17K-ET[Note 1], EMU-17K[Note 2])	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, SIMPLEHOST™. The EMU-17K has a function that allows you to check the contents of the data memory real-time.
SE Board (SE-17204)	The SE-17204 is an SE board for the μPD17203A, 17204, 17P203A and 17P204. It may be used alone to evaluate a system, or in combination with an in-circuit emulator for debugging.
Emulation Probe (EP-17203GC)	The EP-17203GC is an emulation probe for the μPD17203A and 17P204. It connects an SE board and the user system by using in combination with EV-9200G-52[Note 3].
Conversion Socket (EV-9200G-52[Note 3])	The EV-9200G-52 is a socket for a 52-pin plastic QFP (14 × 14 mm) and connects the EP-17203GC and the target system.
★ PROM Programmer (AF-9703[Note 4], AF-9704[Note 4] AF-9705[Note 4], AF-9706[Note 4])	The AF9703, AF9704, AF9705, and AF9706 are PROM programmers that can program the μPD17P203 and 17P204. When connected with programmer adapter AF-9808B, this PROM programmer can program the μPD17P203 and 17P204.
★ Program Adapter (AF-9808B[Note 4])	The AF-9808B is an adapter for programming the μPD17P203 and 17P204 and is used in combination with the AF-9703, AF-9704, AF-9705 or 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.

★ **3.** One EV-9200G-80 is supplied with the EP-17203GC. Five EV-9200G-52s are optionally available as a set.

★ **4.** These are products from Ando Electric. For details, consult Ando Electric.

Software ★

Name	Outline Machine	Host	OS Media		Supply	Order Code
17K Series Assembler (AS17K)	AS17K is an assembler common with the 17K series products. When developing the program of the μPD17203A and the μPD17204, AS17K is used in combination with a device file (AS17203A, AS17204).	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 (AS17203)	AS17203 is a device file for μPD17203A and 17P203A, and it is used in combination with an assembler for the 17K series (AS17K)	PC-9800 series	MS-DOS		5" 2HD	μS5A10AS17203
					3.5" 2HD	μS5A13AS17203
		IBM PC/AT	PC DOS		5" 2HC	μS7B10AS17203
					3.5" 2HC	μS7B13AS17203
Device File (AS17204)	AS17204 is a device file for μPD17204 and μPD17P204, and it is used in combination with an assembler for the 17K series (AS17K).	PC-9800 series	MS-DOS		5" 2HD	μS5A10AS17204
					3.5" 2HD	μS5A13AS17204
		IBM PC/AT	PC DOS		5" 2HC	μS7B10AS17204
					3.5" 2HC	μS7B13AS17204
Support Software (SIMPLE-HOST)	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[Note]
PC DOS	Ver. 3.1 to Ver. 5.0[Note]
Windows	Ver. 3.0 to Ver. 3.1

Note Ver. 5.00/5.00A of MS-DOS and Ver. 5.0 of PC DOS have a task swap function, but 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.)
Santa Clara, 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.1.
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.
Velizy-Villacoublay, 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: 08-63 80 820
Fax: 08-63 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

J96. 8

SIMPLEHOST is a trademark of NEC Corporation.
MS-DOS and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
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-in's.

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.

M4 96.5