NEC
ELECTRON DEVICE

MOS INTEGRATED CIRCUIT

μPD17204

4-BIT SINGLE-CHIP MICROCONTROLLER WITH 8K-BIT STATIC RAM AND 3-CHANNEL TIMER FOR INFRARED REMOTE CONTROLLER

DESCRIPTION

The μPD17204 is a 4-bit single-chip microcontroller for infrared remote controller. Integrated on the same die are an 8K-bit 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 μPD17204 employs a 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 μPD17204 is also equipped with a model, μPD17P204 PROM, to which a program can be written only once, which can be used for evaluation of the μPD17204 program or for small-scale production applications.

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
- Program memory (ROM): 16K bytes (7936 x 16)
- Data memory (RAM): 336 x 4 bits
- Static RAM (XRAM): 2048 x 4 bits
- Instruction execution time: 4 μs (4 MHz, ceramic or crystal oscillator)
- 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: 28
- Operating voltage range: 2.2 to 5.5 V (main clock: 4 MHz)
  2.0 to 5.5 V (subclock: 32 kHz)

ORDERING INFORMATION

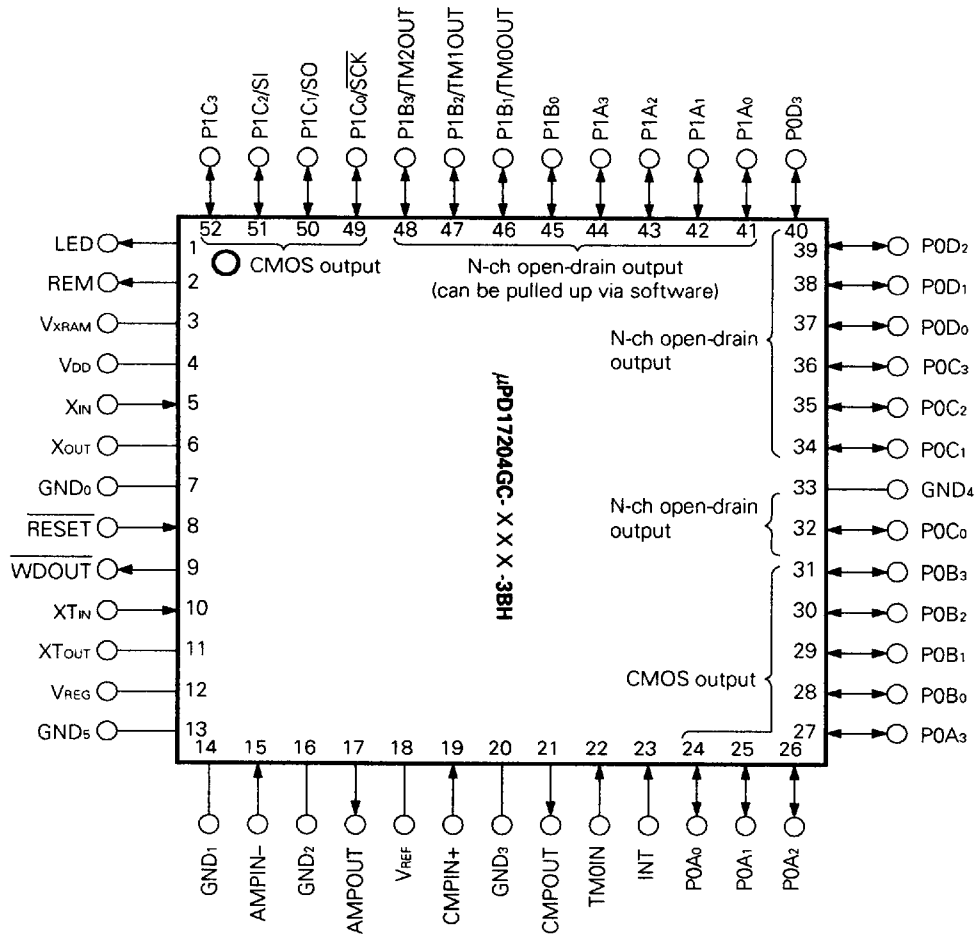
Part Number	Package	Quality grade
μPD17204GC-xxx-3BH	52-pin plastic QFP (14 x 14 mm)	Standard

Please refer to "Quality Grade on NEC Semiconductor Devices" (Document number IEI-1209)published by NEC Corporation to know the specification of quality grade on the devices and its recommended application.

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

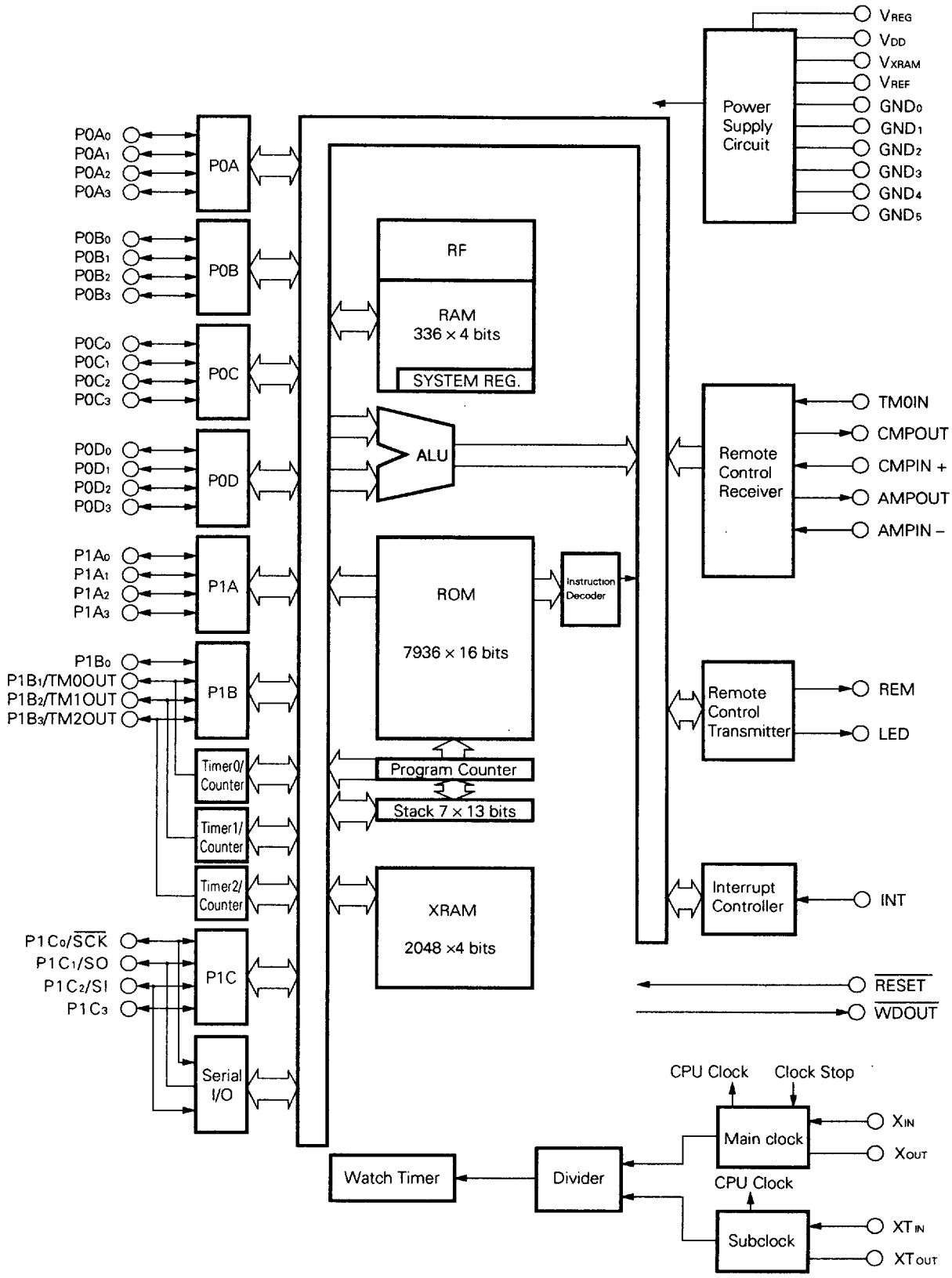
Document No. IC-2852A
(O. D. No. IC-8178B)
Date Published January 1993P
Printed in Japan

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
$P1A_0$-$P1A_3$	:	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
X_{IN}, X_{OUT}	:	Main clock oscillation use
XT_{IN}, XT_{OUT}	:	Subclock oscillation use

BLOCK DIAGRAM



CONTENTS

1. PIN FUNCTIONS 7
   - 1.1 PIN IDENTIFICATION 7
   - 1.2 PIN INPUT/OUTPUT CIRCUITS 9
   - 1.3 PROCESSING OF UNUSED PINS 12

2. PROGRAM COUNTER (PC) 13
   - 2.1 PC FUNCTION 13

3. STACK 14
   - 3.1 STACK 14
   - 3.2 STACK POINTER (SP) 14

4. PROGRAM MEMORY (ROM) 14

5. DATA MEMORY (RAM) 15

6. SYSTEM REGISTER 17
   - 6.1 SYSTEM REGISTER CONFIGURATION 17
   - 6.2 ADDRESS REGISTERS (ARS) 17
   - 6.3 WINDOW REGISTER (WR) 18
   - 6.4 BANK REGISTER (BANK) 18
   - 6.5 MEMORY POINTER ENABLE FLAG (MPE) 18
   - 6.6 INDEX REGISTERS (IXH, IXM, IXL) 19
   - 6.7 REGISTER POINTER (RPH, RPL) 19
   - 6.8 PROGRAM STATUS WORD (PSWORD) 19

7. REGISTER FILE (RF) 20

8. PORTS 22
   - 8.1 PORT 0A ($P0A_0$-$P0A_3$) 22
   - 8.2 PORT 0B ($P0B_0$-$P0B_3$) 22
   - 8.3 PORT 0C ($P0C_0$-$P0C_3$) 22
   - 8.4 PORT 0D ($P0D_0$-$P0D_3$) 22
   - 8.5 PORT 1A ($P1A_0$-$P1A_3$) 22
   - 8.6 PORT 1B ($P1B_0$-$P1B_3$) 23
   - 8.7 PORT 1C ($P1C_0$-$P1C_3$) 23
   - 8.8 INT PIN 24
   - 8.9 PORT CONTROL REGISTER FILE 25
     - 8.9.1 Input/Output Switching of Group I/O 25
     - 8.9.2 Input/Output Switching of Bit I/O 26
     - 8.9.3 Pull-up Resister On/Off Control 28
     - 8.9.4 Timer Output Switching 30
     - 8.9.5 Serial Input/Output Switching 30

9. CLOCK GENERATOR CIRCUITS 31
9.1 SYSTEM CLOCK SELECTION 31
9.2 MAIN CLOCK OSCILLATION CONTROL FUNCTION 32

10. TIMER FUNCTION 33
10.1 8-BIT TIMER 0 AND REMOTE CONTROLLER CARRIER GENERATOR CIRCUIT 33
10.1.1 Configuration of 8-bit timer 33
10.1.2 Carrier Generation Circuit for Remote Controller 37
10.1.3 Countermeasures Against Noise During Transmission (Carrier Output) 39
10.2 10-BIT TIMER 1 40
10.3 16-BIT TIMER 2 44

11. WATCH TIMER/WATCHDOG TIMER 46
11.1 CONFIGURATION OF WATCH TIMER/WATCHDOG TIMER 46
11.2 FUNCTION OF WATCH TIMER/WATCHDOG TIMER 47
11.3 WATCHDOG TIMER OPERATION TIMING 48

12. CARRIER RECEIVER CIRCUIT FOR REMOTE CONTROLLER 49
12.1 ANALOG CIRCUIT, CLOCK NOISE REJECTER CIRCUIT, AND ENVELOPE CIRCUIT 49
12.1.1 Analog circuit 49
12.1.2 Clock noise rejecter circuit 49
12.1.3 Envelope circuit 50

13. SERIAL INTERFACE 54
13.1 SERIAL INTERFACE FUNCTIONS 54
13.1.1 Clock-Synchronous 8-Bit Transmission/Reception (Simultaneous Transmission and Reception) 54
13.1.2 Clock-Synchronous 8-Bit Reception (SO Output High-Impedance) 54
13.2 SERIAL INTERFACE OPERATION 54
13.2.1 Serial Interface Operation Mode 54
13.2.2 Serial Operation Mode 54

14. STATIC RAM (XRAM) 58

15. INTERRUPT FUNCTION 61
15.1 INTERRUPT SOURCES 61
15.2 REGISTER FILE FOR INTERRUPT CONTROL 62
15.2.1 INT 62
15.2.2 INTENV 62
15.2.3 IEG 63
15.2.4 IEGENVM0 and IEGENVM1 63
15.2.5 Interrupt Enable Flags 64
15.2.6 Interrupt Request Flag 65
15.3 INTERRUPT SEQUENCE 67
15.3.1 Operation When Interrupt Is Accepted 67
15.3.2 Exiting from Interrupt Routine 67

16. STANDBY FUNCTION 68
16.1 HALT MODE 68
16.2 HALT INSTRUCTION EXECUTION CONDITION 69
16.3 STOP MODE 69
16.4 STOP INSTRUCTION EXECUTION CONDITION 70
16.5 OPERATIONS AFTER STANDBY MODE HAS BEEN RELEASED 70

17. RESET 72
17.1 RESET BY RESET SIGNAL INPUT 72
17.2 RESET BY WATCHDOG TIMER (CONNECTING $\overline{\text{RESET}}$ AND $\overline{\text{WDOUT}}$ PINS) 72
17.3 RESET BY STACK POINTER (CONNECTING $\overline{\text{RESET}}$ AND $\overline{\text{WDOUT}}$ PINS) 72

18. ASSEMBLER RESERVED WORDS 73
18.1 MASK OPTION DIRECTIVE 73
18.1.1 OPTION and ENDOP directives 73
18.1.2 Mask option definition directives 73
18.2 RESERVED SYMBOLS 74
18.2.1 Control registers on register file 74
18.2.2 Registers and ports on data memory 74
18.2.3 Peripheral circuits 74

19. μPD17204 INSTRUCTION SET 79
19.1 OUTLINE OF INSTRUCTION SET 79
19.2 LEGEND 80
19.3 INSTRUCTION SET 81

20. APPLICATION CIRCUIT EXAMPLE 83

21. ELECTRICAL SPECIFICATIONS 84

22. PACKAGE DRAWINGS 90

23. RECOMMENDED SOLDERING CONDITIONS 92

APPENDIX A. LIST OF MICROCONTROLLERS FOR LEARNING REMOTE CONTROLLER ..93

APPENDIX B. DEVELOPMENT TOOLS 94

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	N-ch open-drain	High inpedance
10 11	XT_{IN} XT_{OUT}	Connect a 32-kHz crystal oscillator for subclock. When an option that does not use the subclock is selected, the divided output of the main clock is used as a timer clock.	–	–
12	V_{REG}	Output pin of voltage regulator for subclock oscillation circuit. To use this pin, an external 0.1-μF capacitor must be connected.	–	–
13	GND_5	Ground	–	–
14	GND_1	Operational amplifier ground.	–	–
15	AMPIN-	Inverted input of operational amplifier.	–	Input
16	GND_2	Operational amplifier ground.	–	–
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
20	GND_3	Operational amplifier ground.	–	–

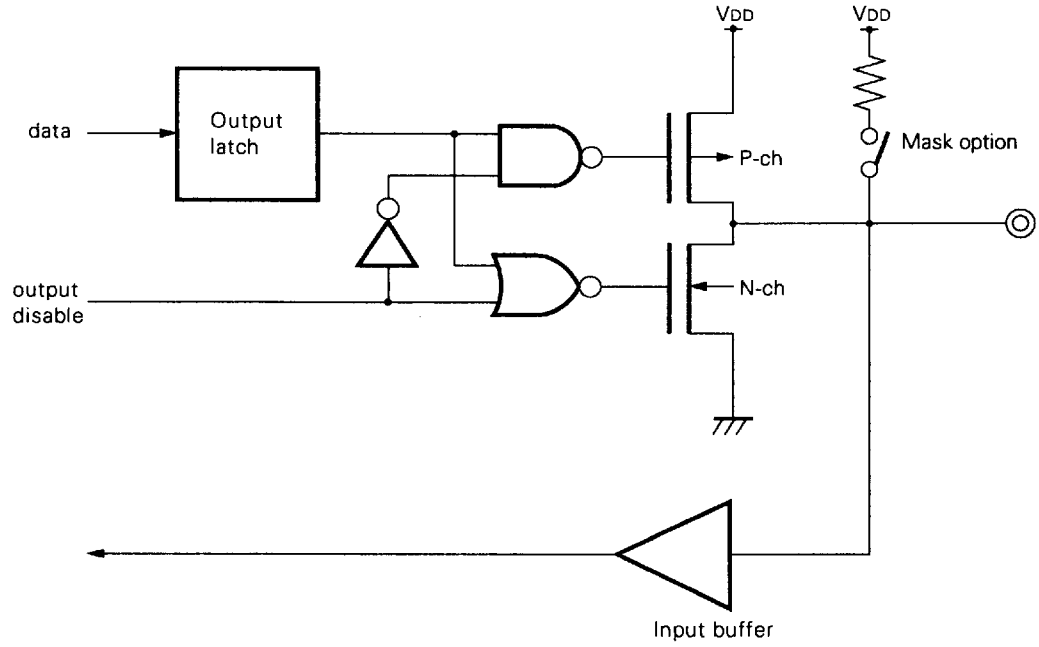
Note : GND_1 to GND_3 are operational amplifier grounds.
To stabilize the operation of these amplifiers, they must be made equipotentional.

Pin No	Symbol	Function	Output Format	At Reset
21	CMPOUT	Output pin for comparator. To use a learning remote controller, CMPOUT and TM0IN must be externally connected.	–	Output
22	TM0IN	Clock input to timer 0. After it has been sampled by the internal clock, the input clock is supplied to timer 0 and, at the same time, to 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 ($P1C_0$-$P1C_3$)

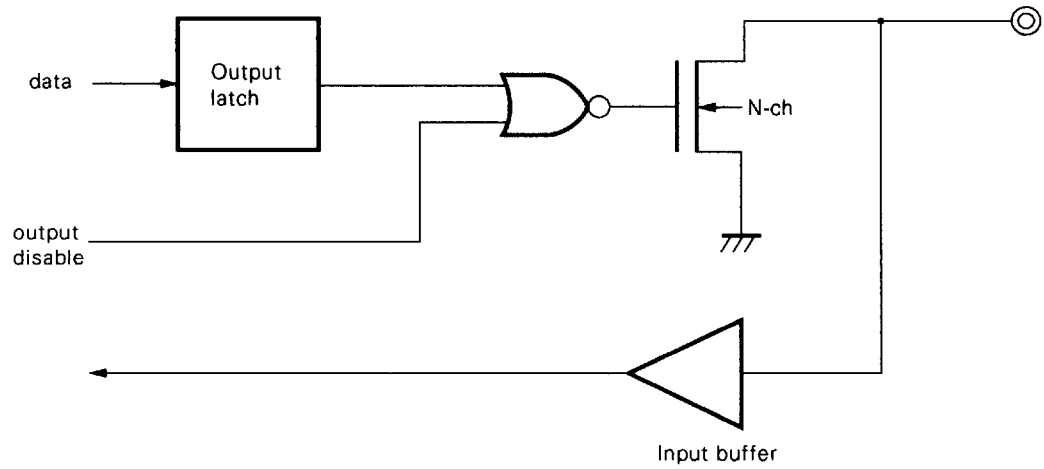
1.2 PIN INPUT/OUTPUT CIRCUITS

This section shows simplified diagrams illustrating the input/output circuits of the μPD17204 pins.

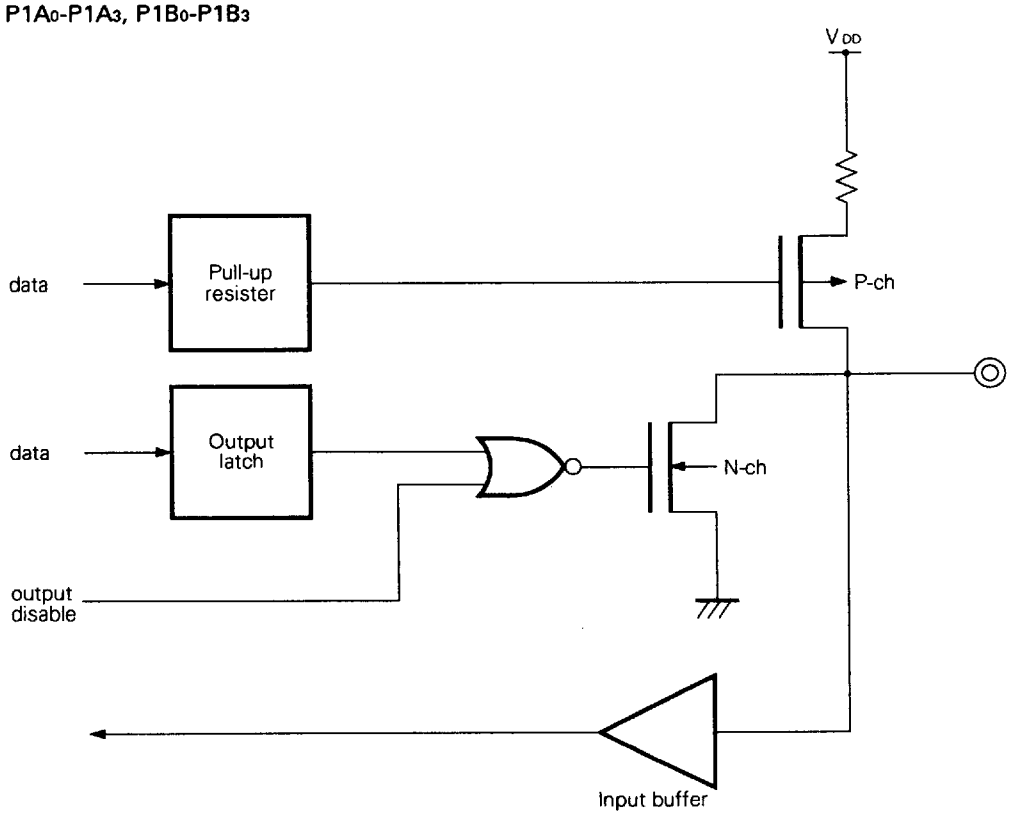
(1) $P0A_0$-$P0A_3$, $P0B_0$-$P0B_3$



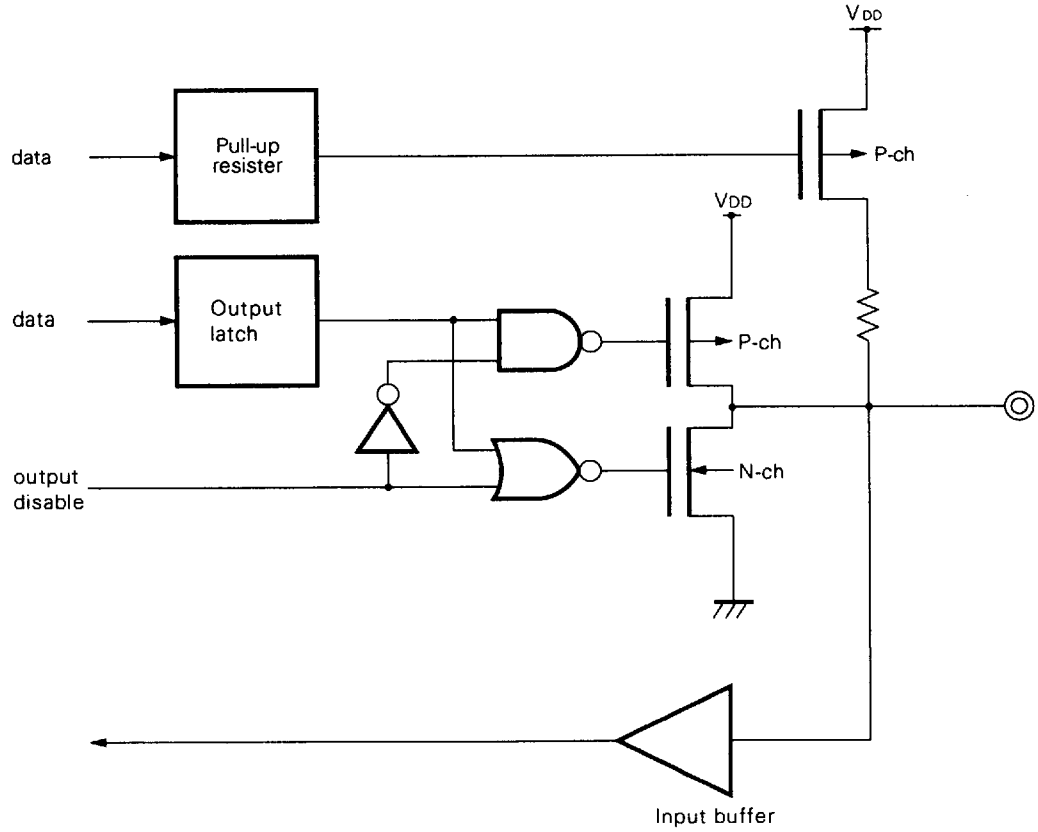
(2) $P0C_0$-$P0C_3$, $P0D_0$-$P0D_3$



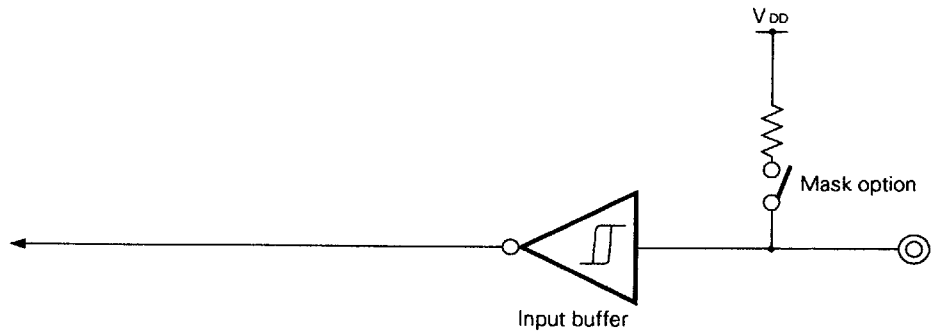
(3) P1A0-P1A3, P1B0-P1B3



(4) P1C0-P1C3

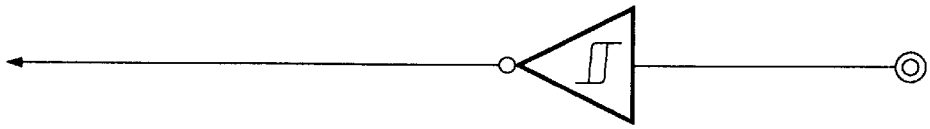


(5) $\overline{\text{RESET}}$



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

2. PROGRAM COUNTER (PC)

2.1 PC FUNCTION

The program counter (PC) is a 13-bit binary counter that addresses program memory (ROM), that is, the program.

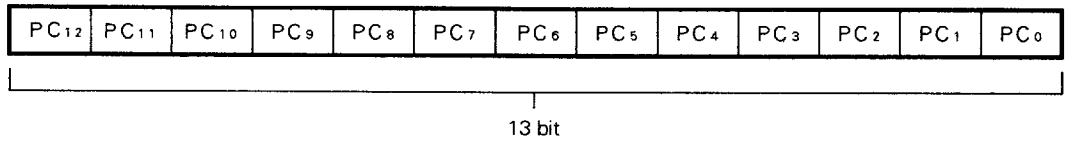


Fig. 2-1 Program Counter (PC)

Usually, the PC is incremented each time an instruction is executed. When a jump instruction or subroutine call instruction is executed, the address specified in the operand field of the instructon is loaded to the PC. When a skip instruction is executed, the PC specifies the address of the next instruction to be executed, regardless of the skip condition. If the skip condition is "satisfied", the instruction next to the skip instrution is assumed to be an NOP (No Operation) instruction. That is, the NOP instruction is executed first and then the next instruction address is specified.

When an interrupt request is accepted, an address (1 to 8) corresponding to the interrupt source is unconditionally loaded into the PC.

At reset, the PC is reset to address 0.

Table 2-1 Vector Address at Interrupt

Priority	Interrupt Source	Internal/External	Vector Address
1	Rising or falling edge or both edges of envelope circuit output	Internal	8H
2	Timer 2 overflow	Internal	7H
3	Timer 0 increment	Internal	6H
4	Timer 1 increment	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

3. STACK

3.1 STACK

A stack consists of 7 x 13 bits. When a subroutine call instruction is executed or when an interrupt request is accepted, the stack stores the value of "PC + 1", that is, the return address. The stack contents are loaded into the PC when the return instruction is executed. Control returns to the main program (the original program flow is restored).

The stack is used for both subroutine call and interrupt. When two levels are used for interrupt, the remaining 5 levels can be used for subroutine call.

When an interrupt occurs, not only the contents of the PC but also those of the PSWORD (5 bits of BCD, CMP, CY, Z, and IXE) and BANK registers are saved. When an RETI instruction is executed, control returns to the calling source. The contents of the PSWORD and BANK registers are saved only in two levels of the stack.

When the MOVT instruction is executed, one level of the stack is temporarily used.

3.2 STACK POINTER (SP)

The stack pointer (SP) indicates the stack level. It is at address 01 on the register file. When data is stored in the stack (when the CALL, MOVT, or PUSH instruction is executed, or when an interrupt is accepted), the SP is decremented by 1. When data is restored from the stack (when the RET, RETSK, RETI, MOVT, or POP instruction is executed), the SP is incremented by 1.

4. PROGRAM MEMORY (ROM)

The read-only memory (ROM) consists of 7936 words x 16 bits. ROM stores the program at ROM addresses 000H through 1EFFH (7936 steps). ROM is divided into 4 pages with each page consisting of 2K steps.

The entire ROM can also be used as a table reference area when the MOVT instruction is used.

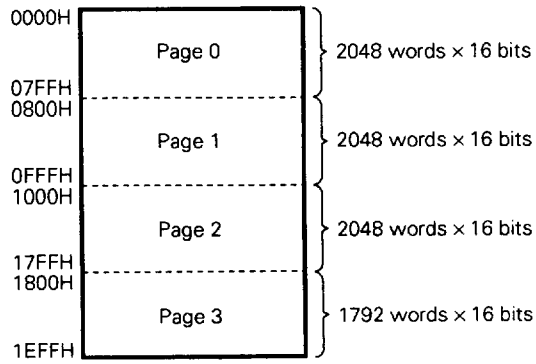


Fig. 4-1 ROM Configuration

5. DATA MEMORY (RAM)

The random-access memory (RAM) consists of 336 words x 4 bits. RAM stores ordinary data.

Four words of 0CH to 0FH of BANK0 are used as a data buffer (DBF). For 8-bit data transfer, two words of 0EH and 0FH are used. For 16-bit data transfer, four words of 0CH to 0FH are used.

This area is used for data exchange with the pheripheral device register specified by the PUT or GET instruction. It is also used as the destination to which 16-bit data is to be read by the ROM data reference instruction (MOVT). The registers that can be specified by the PUT and GET instructions are the count and modulo registers of timers 0 and 1, count register of timer 2, SIO shift register, and address register. The modulo register for setting the high-level period and generating a remote controller carrier (NRZHTMM), the modulo register for setting the low-level period (NRZLTMM), the XRAM start address setting register, and the XRAM stop address detecting register can also be specified.

At reset, address 0CH is initialized to 0H, 0DH to 3H, 0EH to 2H, and 0FH to 0H.

Table 5-1 lists pheripheral device addresses.

Table 5-1 Pheripheral Device Addresses

No.	Name	Address	Number of Effective Bits	Description
1	SIOSFR	01H	8	SIO shift register
2	TM0C TM0M	02H	8	Count and modulo registers of timer 0
3	NRZLTMM	03H	8	Modulo register setting low-level period for generating remote control carrier
4	NRZHTMM	04H	8	Modulo register setting high-level period for generating remote control carrier
5	AR	40H	16	Address register
6	TM1C TM1M	41H	16	Count and modulo registers of timer 1
7	TM2C	42H	16	Count register of timer 2
8	XRAMSTRT	43H	16	XRAM start address setting register
9	XRAMSTP	44H	16	XRAM stop address detecting register

70H to 73H of BANK0 and 70H to 72H of BANK1 are assigned to ports. Input/output of these ports is performed via these banks. 73H of BANK2 is assigned to the data register used when a 16K-bit SRAM is accessed.

74H to 7FH are assigned to system registers which can access any banks.

When reset, RAM becomes undefined. Therefore, be sure to initialize RAM to 0 (RAM clear) at the beginning of the program, to eliminate the cause for an unexpected bug.

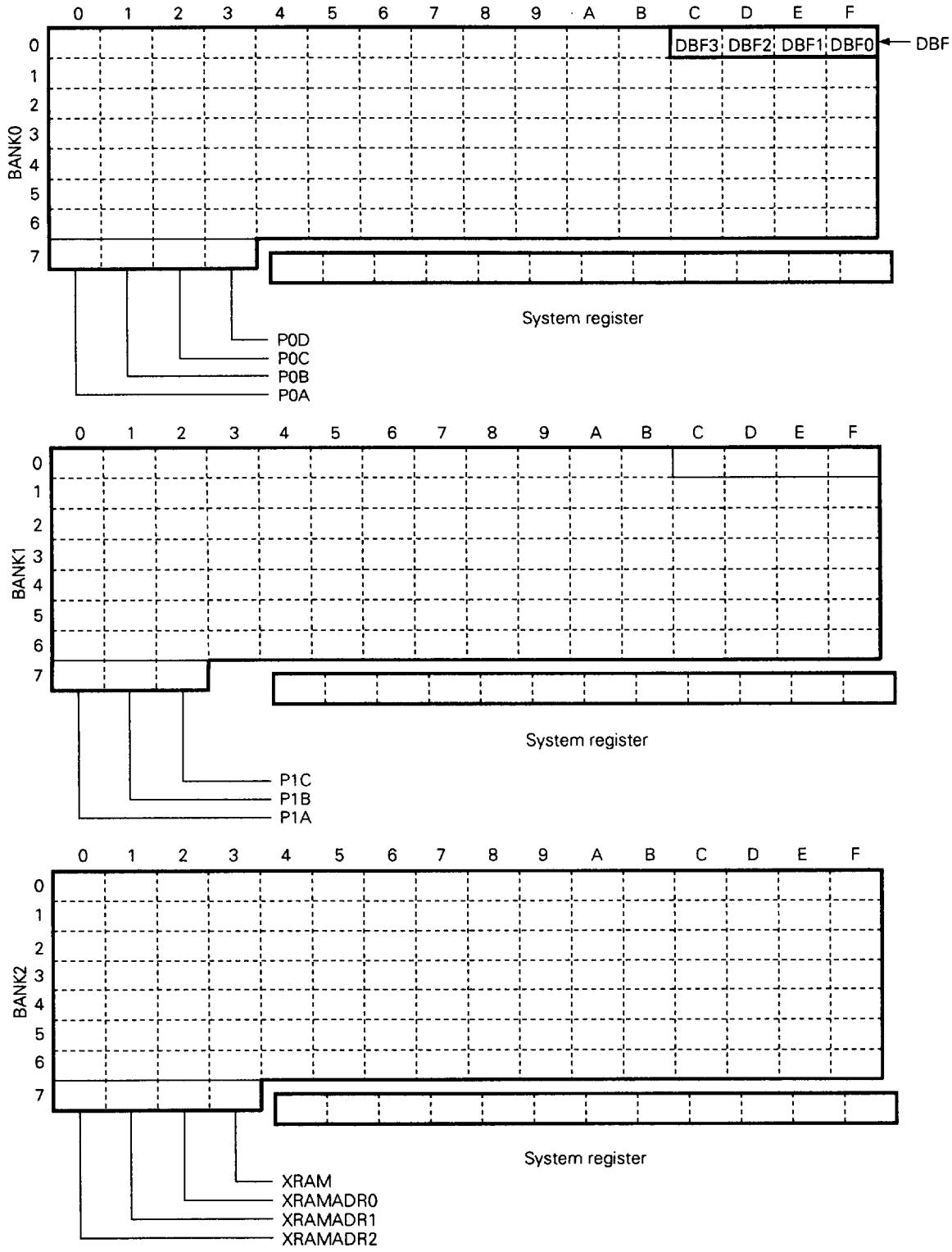


Fig. 5-1 RAM Configuration

6. SYSTEM REGISTER

6.1 SYSTEM REGISTER CONFIGURATION

"System register" is a generic name for the registers directly related to CPU control. The system registers are placed at addresses 74H through 7FH on data memory (RAM) and can be referenced regardless of bank specification.

The following registers are usable as the system register:

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

Address	74H	75H	76H	77H	78H	79H
Register	AR3	AR2	AR1	AR0	WR	BANK
Value	000x	xxxx	xxxx	xxxx	xxxx	00xx
At reset	0000	0000	0000	0000	****	0000

Address	7AH		7BH	7CH	7DH	7EH		7FH				
Register	IXH/MPH		IXM/MPL	IXL	RPH	RPL		PSW				
	MPE						BCD	CMP	CY	Z	IXE	
Value	x	00x	xxxx	xxxx	00xx	xxx	x	x	x	x	x	
At reset	0	000	0000	0000	0000	000	0	0	0	0	0	

* : Undefined

x : Bits that can be set

Fig. 6-1 System Register Configuration

6.2 ADDRESS REGISTERS (ARS)

The address registers (ARs) specify program memory (ROM) addresses and placed at addresses 74H through 77H. The indirect branch, table reference, and stack operation instructions use the ARs.

BR @AR and CALL @AR are provided for indirect branch. These branch instructions branch program execution to program memory addresses specified by the AR.

MOVT DBF, @AR is provided for table reference. This reference instruction transfers the contents of the addresses specified by the ARs to DBF (0CH to 0FH of BANK0) on RAM (data memory).

PUSH AR and POP AR are provided for stack operation. The PUSH AR instruction stores the AR contents in the stack indicated by the stack pointer and decrements the pointer by 1. The POP AR instruction increments the stack pointer by 1 and loads the contents of the stack indicated by the pointer into the AR.

AR3 in the μPD17204 is fixed to 0. Therefore, program addresses 0000H to 1FFFH (8192 steps) can be specified by the ARs. In other words, the entire internal ROM area can be specified. However, only 7936 steps of ROM (addresses 0000H to 1EFFH) is internally provided. Therefore, when specifying addresses, be sure not to specify the addresses where no ROM exists.

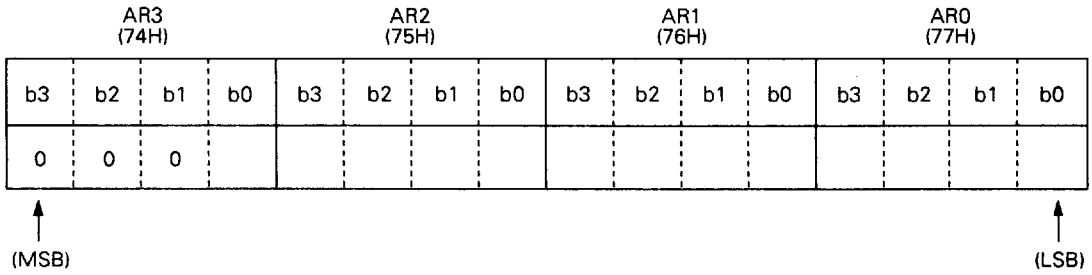


Fig. 6-2 AR Configuration

6.3 WINDOW REGISTER (WR)

The window register (WR) is a 4-bit register mapped at address 78H in the system register. The data of each register on the register file is manipulated via the WR.

6.4 BANK REGISTER (BANK)

The bank register (BANK) specifies a bank on data memory (RAM).

At reset, the BANK is cleared to 0. The high-order two bits of address 79H are always 0.

Table 6-1

Bank		BANK
b1	b0	
0	0	BANK0
0	1	BANK1
1	0	BANK2
1	1	Setting inhibited

When an interrupt is accepted, the BANK contents are saved.

6.5 MEMORY POINTER ENABLE FLAG (MPE)

The memory pointer enable flag (MPE) specifies whether to specify the row address, used when the MOV @r, m or MOV m, @r instruction is executed, by MPL or execute the instruction using the same row address. When the MPE is set, the row address is specified by MPL. When the MPE is reset, the same row address is specified.

The row address specified by MPL, however, must be a row address in the bank currently used.

6.6 INDEX REGISTERS (IXH, IXM, IXL)

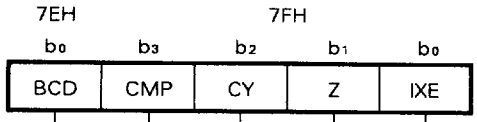
The index registers are used to indirectly address the RAM. When the index enable flag (IXE) is set, the operand field of an instruction is ORed with the contents of the index registers (IXM, IXL) to address the RAM.

6.7 REGISTER POINTER (RPH, RPL)

The register pointers specify the bank and row addresses of a general register. RPH is used to specify the bank address and RPL is used to specify the row address.

6.8 PROGRAM STATUS WORD (PSWORD)

The program status word (PSWORD) consists of five flags indicating the operation result of the arithmetic logic unit (ALU) in the CPU and qualifying the ALU functions. These five flags are: BCD flag, CMP (compared) flag, CY (carry) flag, Z (zero) flag, and IXE (Index enable) flag. Figure 6-3 shows the functions of these flags.



Flag Name	Function
Index Enable	When this flag is set, index qualification is enabled.
Zero	This flag is reset when the arithmetic operation result is not 0. The set condition depends on the content of the CMP flag. (1) CMP = 0 The Z flag is set when the arithmetic operation result is 0. (2) CMP = 1 The Z flag is set only when it is 1 and the arithmetic operation result is 0.
Carry	This flag is set when a carry occurs during execution of an add instruction or when a borrow occurs during execution of a substract instruction. It is reset when neither a carry nor a borrow occurs. This flag is also set if the lowest-order bit of the general register is 1 when the RORC instruction is executed. It is also reset if the lowest-order bit is 0.
Compare	When this flag is set, the arithmetic operation result is not stored in data memory (RAM). It is automatically reset when the SKT or SKF instruction is executed.
BCD	When this flag is set, all arithmetic operations are performed in binary-coded decimal (BCD). When it is reset, arithmetic operations are performed in binary.

Fig. 6-3 PSWORD Configuration

7. REGISTER FILE (RF)

With the μPD17204, the bits on the register file are manipulated to control switching of the input/output mode of peripheral circuits and ports.

Table 7-1 shows the register file configuration.

Table 7-1 Register File Configuration (1/2)

Row Address \ Column Address		0		1		2		3		4		5		6		7	
			*1		*1		*1		*1		*1		*1		*1		*1
0 (8)	Bit 3			SP	0	0	0	WDTRES	0							0	0
	Bit 2				1	0	0	WTMMD	0							0	0
	Bit 1				0	SYSCK	*2	WTMRES	0							0	0
	Bit 0				1	XEN	*2	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

*1 : Status at reset

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

Remarks : Figures in ( ) are addresses for use with the assembler (AS17K).
All of the control register flags are registered in the device file as reserved assembler words, so they are convenient for program development.

Table 7-1 Register File Configuration (2/2)

Row Address \ Column Address		8	*1	9	*1	A	*1	B	*1	C	*1	D	*1	E	*1	F	*1
0	Bit 3															0	0
	Bit 2															0	0
(8)	Bit 1															INTENV	1
	Bit 0															INT	*2
1	Bit 3															0	0
	Bit 2															IEGENVM1	0
(9)	Bit 1															IEGENVM0	0
	Bit 0															IEG	0
2	Bit 3													IPXRAM	0	IPTM1	0
	Bit 2													IPSIO	0	IPTM0	0
(A)	Bit 1													IPWTM	0	IPTM2	0
	Bit 0													IP	0	IPENV	0
3	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
(B)	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

*1 : Status at reset

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

8. PORTS

8.1 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).

8.2 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).

8.3 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.

8.4 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.

8.5 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.

8.6 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).

8.7 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.

8.8 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 Fig. 15-2 INT and INTENV Flags).

Bank	Address	Symbol	Bit		Port	Output Type
0	70H	P0A	b_3	$P0A_3$	Port 0A	CMOS push-pull
			b_2	$P0A_2$		
			b_1	$P0A_1$		
			b_0	$P0A_0$		
	71H	P0B	b_3	$P0B_3$	Port 0B	CMOS push-pull
			b_2	$P0B_2$		
			b_1	$P0B_1$		
			b_0	$P0B_0$		
	72H	P0C	b_3	$P0C_3$	Port 0C	N-ch open drain
			b_2	$P0C_2$		
			b_1	$P0C_1$		
			b_0	$P0C_0$		
	73H	P0D	b_3	$P0D_3$	Port 0D	N-ch open drain
			b_2	$P0D_2$		
			b_1	$P0D_1$		
			b_0	$P0D_0$		
1	70H	P1A	b_3	$P1A_3$	Port 1A	CMOS push-pull
			b_2	$P1A_2$		
			b_1	$P1A_1$		
			b_0	$P1A_0$		
	71H	P1B	b_3	$P1B_3$	Port 1B	N-ch open drain
			b_2	$P1B_2$		
			b_1	$P1B_1$		
			b_0	$P1B_0$		
	72H	P1C	b_3	$P1C_3$	Port 1C	CMOS push-pull
			b_2	$P1C_2$		
			b_1	$P1C_1$		
			b_0	$P1C_0$		

Fig. 8-1 Relation between Port Registers and Pins

8.9 PORT CONTROL REGISTER FILE

8.9.1 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.

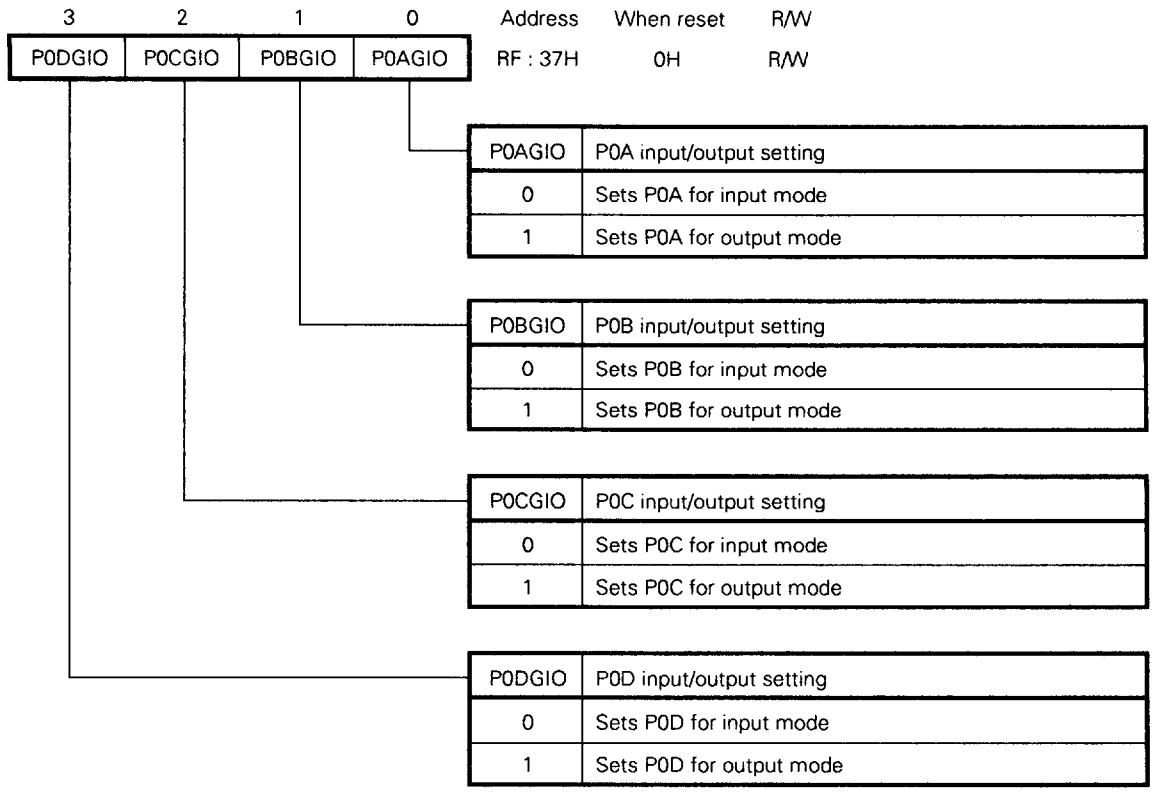


Fig. 8-2 Input/Output Control Register for Group I/O

8.9.2 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 latch contents of each P1A, P1B, and P1C bit are output to the corresponding port bit.

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

Fig. 8-3 Input/Output Control Register for Bit I/O: P1A

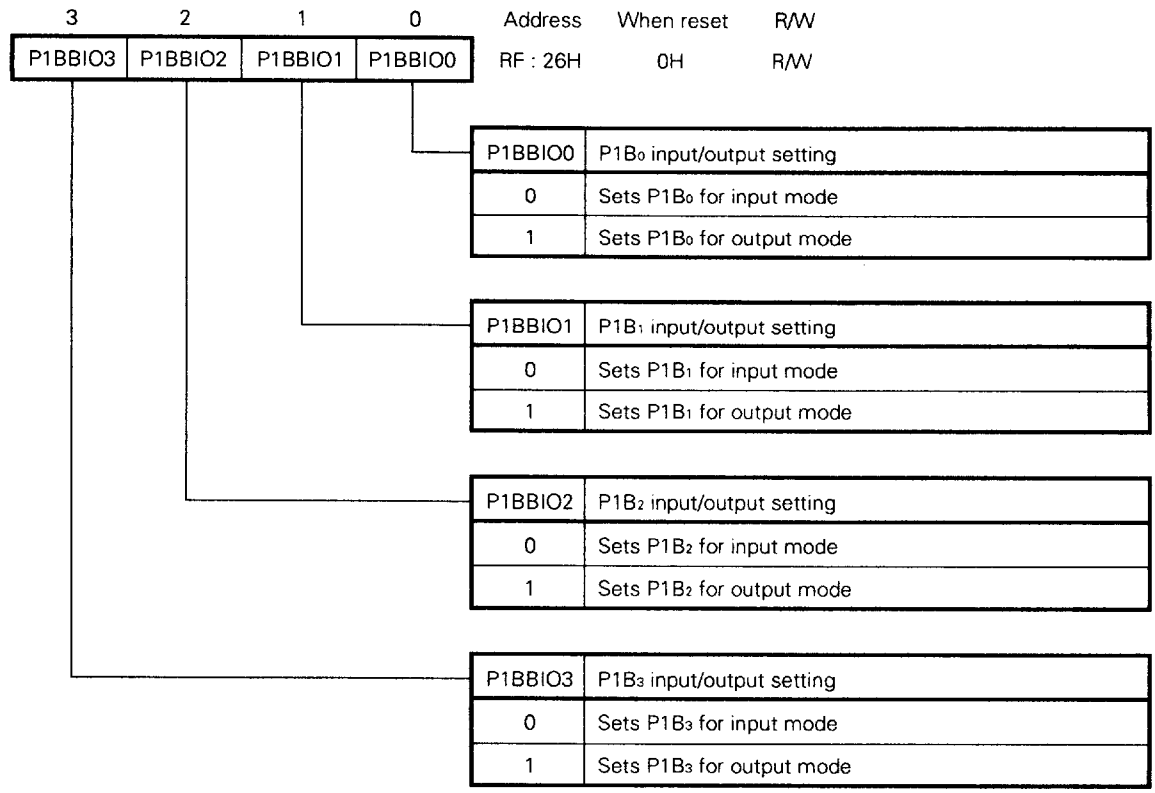


Fig. 8-4 Input/Output Control Register for Bit I/O: P1B

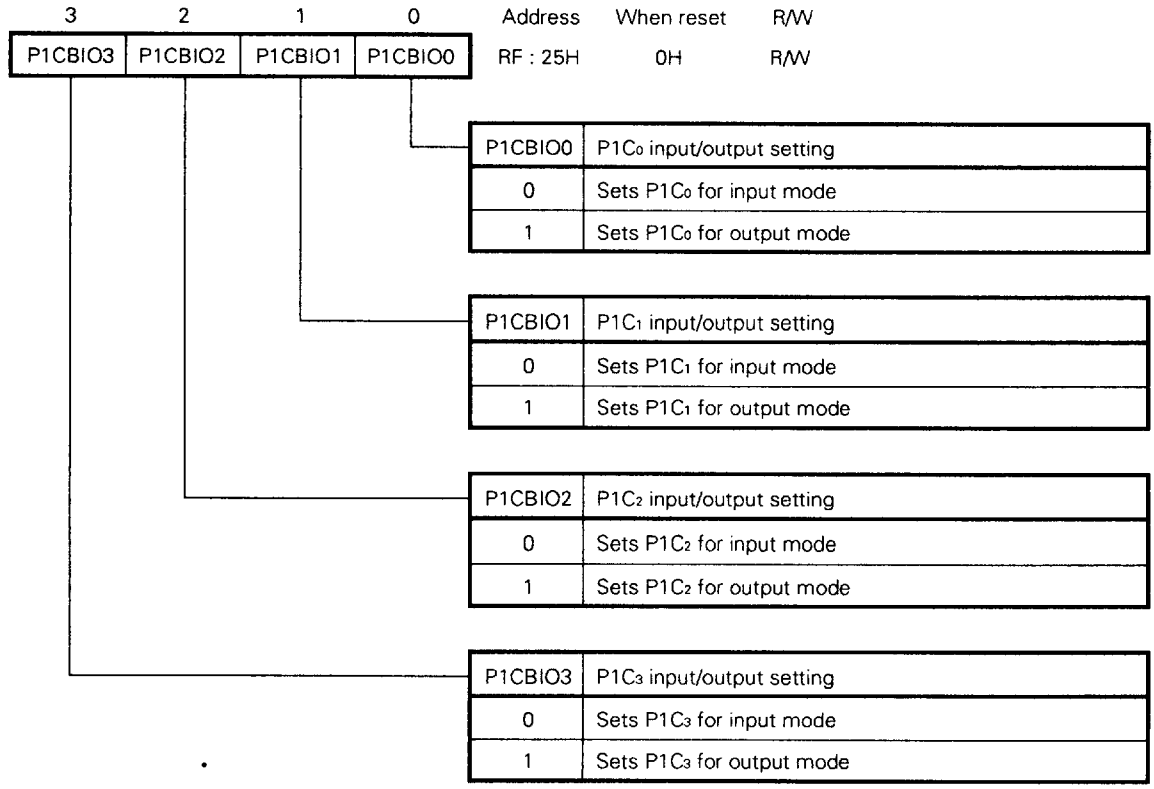


Fig. 8-5 Input/Output Control Register for Bit I/O: P1C

8.9.3 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.

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

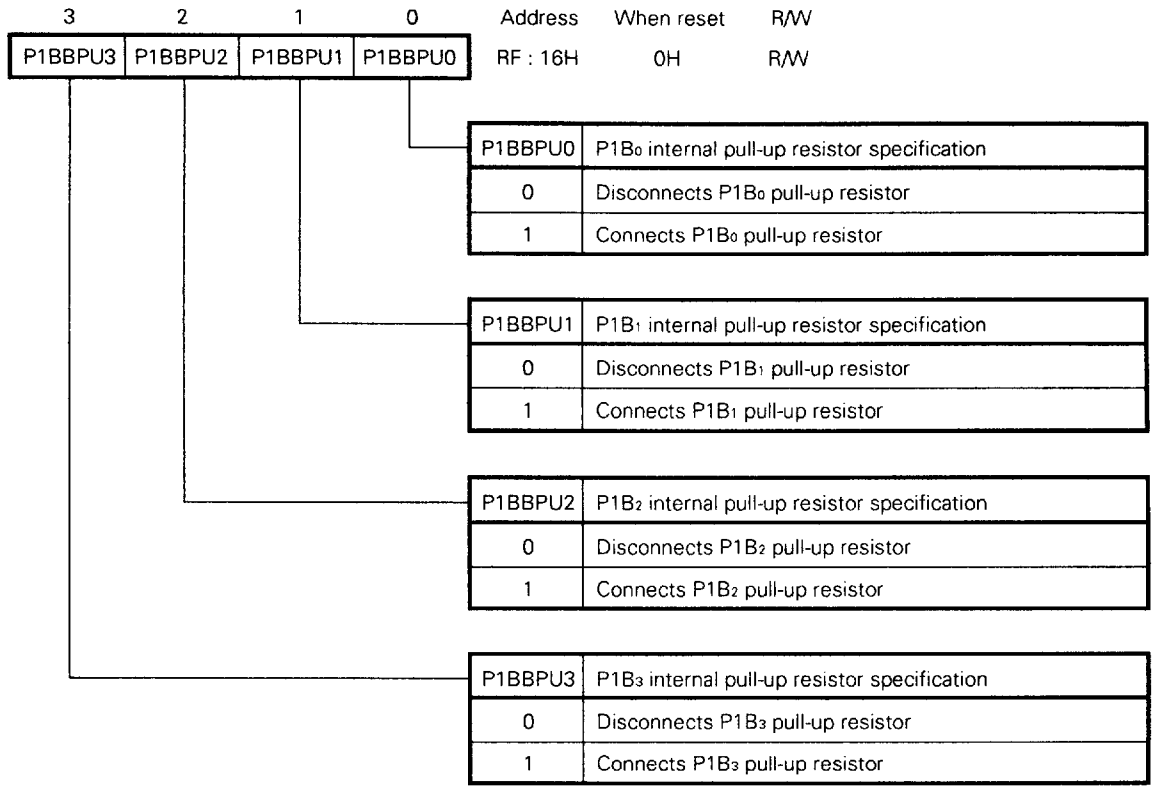
P1ABPU0	$P1A_0$ internal pull-up resistor specification
0	Disconnects $P1A_0$ pull-up resistor
1	Connects $P1A_0$ pull-up resistor

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

Fig. 8-6 Pull-up Resistor On/Off Control Register: P1A



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

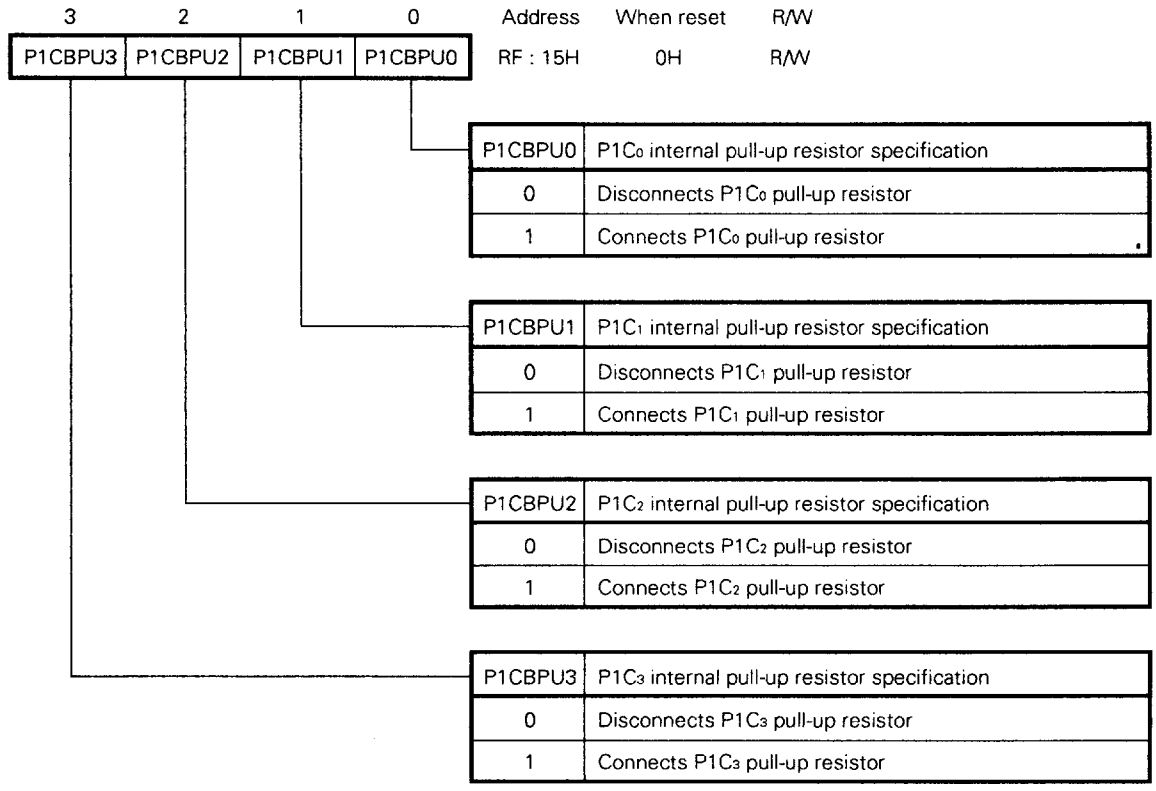
P1BBPU0	P1B0 internal pull-up resistor specification
0	Disconnects P1B0 pull-up resistor
1	Connects P1B0 pull-up resistor

P1BBPU1	P1B1 internal pull-up resistor specification
0	Disconnects P1B1 pull-up resistor
1	Connects P1B1 pull-up resistor

P1BBPU2	P1B2 internal pull-up resistor specification
0	Disconnects P1B2 pull-up resistor
1	Connects P1B2 pull-up resistor

P1BBPU3	P1B3 internal pull-up resistor specification
0	Disconnects P1B3 pull-up resistor
1	Connects P1B3 pull-up resistor

Fig. 8-7 Pull-up Resistor On/Off Control Register: P1B



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

P1CBPU0	P1C0 internal pull-up resistor specification
0	Disconnects P1C0 pull-up resistor
1	Connects P1C0 pull-up resistor

P1CBPU1	P1C1 internal pull-up resistor specification
0	Disconnects P1C1 pull-up resistor
1	Connects P1C1 pull-up resistor

P1CBPU2	P1C2 internal pull-up resistor specification
0	Disconnects P1C2 pull-up resistor
1	Connects P1C2 pull-up resistor

P1CBPU3	P1C3 internal pull-up resistor specification
0	Disconnects P1C3 pull-up resistor
1	Connects P1C3 pull-up resistor

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

8.9.4 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:

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

Fig. 8-9 P1B/Timer Output Control Register

8.9.5 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, SOHIZ, and SIOTS bits on the register files are used to control serial input/output mode. (For details, see Chapter 13 "Serial Interface".)

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

Fig. 8-10 P1C/Serial Input/Output Control Register

9. CLOCK GENERATOR CIRCUITS

The μPD17204 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 9-1 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.

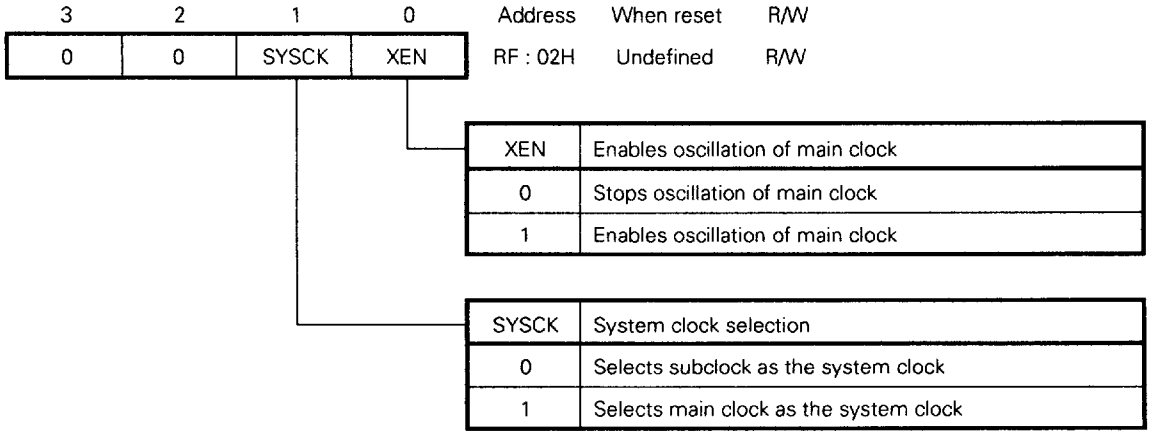


Fig. 9-1 System Clock Control Register

9.1 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 Fig. 9-1.

(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).

Note : 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).

Note : 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.

9.2 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.

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

10. TIMER FUNCTION

The μPD17204 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

10.1 8-BIT TIMER 0 AND REMOTE CONTROLLER CARRIER GENERATOR CIRCUIT

10.1.1 CONFIGURATION OF 8-BIT TIMER

Figure 10-1 shows the configuration of the 8-bit timer.

The 8-bit timer consists of an 8-bit counter, an 8-bit modulo register, 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 same data register, TM0M (peripheral register: address 02H), is used for the 8-bit timer and the modulo register, 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 $P1B_1$/TM0OUT pin is inverted. By setting TMRES, TM0OUT will be initialized and will output a low level signal.

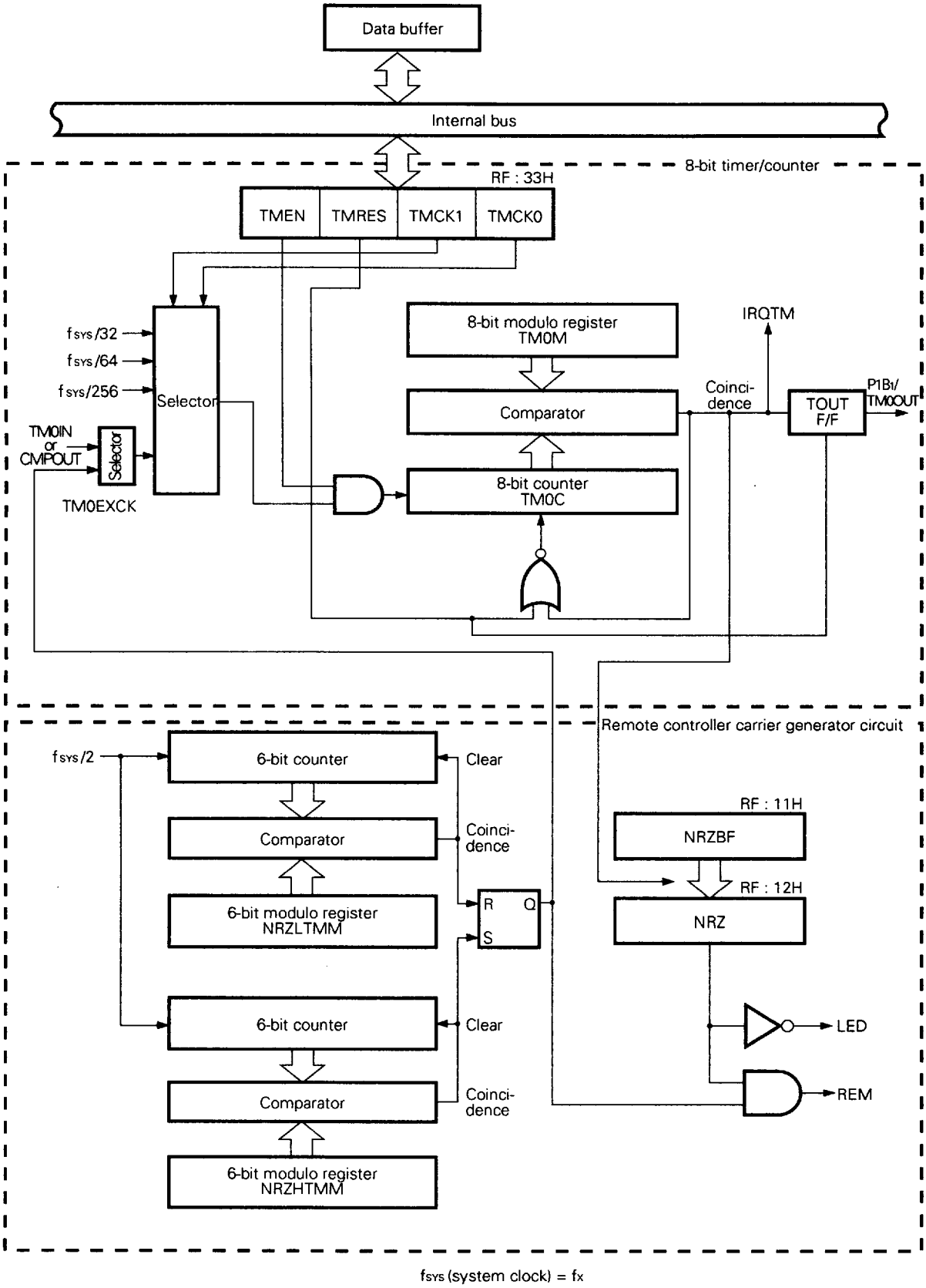


Fig. 10-1 Configuration of 8-bit Timer/Counter and Remote Controller Caririer Generator Circuit

The following shows the register files for timer 0.

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

TM0CK1	TM0CK0	8-bit timer clock source selection
0	0	Count clock frequency: f_{SYS} /32 (Measurement time range: 8 μs to 2.048 ms)
0	1	Count clock frequency: f_{SYS} /64 (Measurement time range: 16 μs to 4.096 ms)
1	0	Count clock frequency: f_{SYS} /256 (Measurement time range: 64 μs to 16.384 ms)
1	1	Remote control carrier generator output or clock noise elimination circuit *2

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

TM0RES	8-bit timer reset flag
0	Data read out is always "0"
1	Writing "1" resets 8-bit timer

TM0EN	8-bit timer count enable flag
0	Stops 8-bit timer count operation
1	Enable 8-bit timer count operation

*1 : Bit 2 is write-only bit.

*2 : Selection of remote control carrier generator output/clock noise elimination circuit output is performed by the TM0EXCK bit on the register file.

Fig. 10-2 Control Register for Timer 0 (1/2)

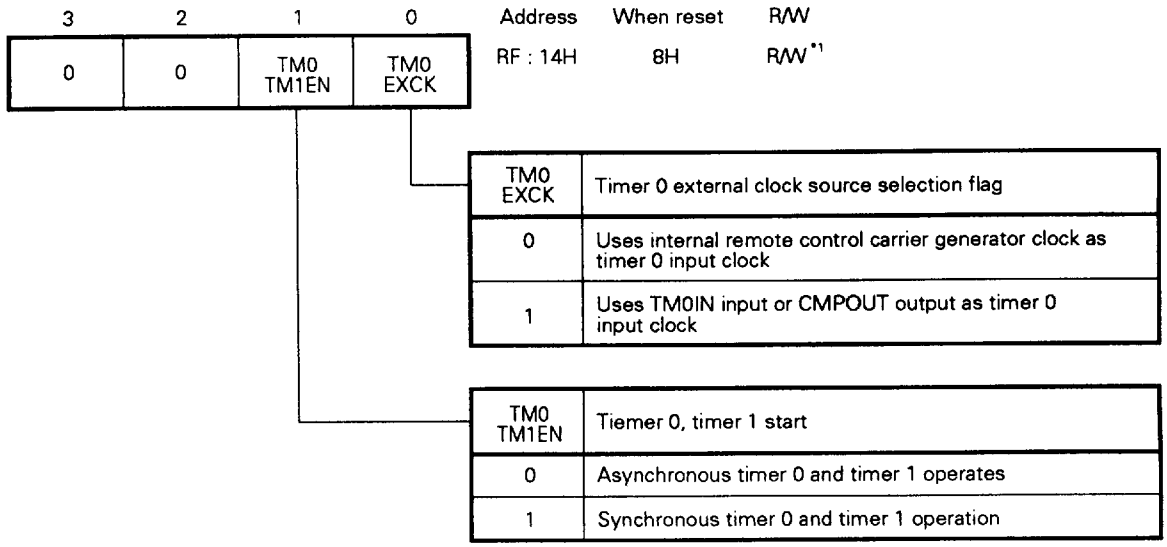


Fig. 10-2 Control Register for Timer 0 (2/2)

Note : 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).

TM0EXCK becomes valid when both TM0CK1 and TM0CK0 are set to "1". By manipulating TM0EXCK, the internal carrier generator output, the input of the TM0IN pin selected by TM0INEN, and the internal comparator output can be selected.

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.

10.1.2 Carrier Generation Circuit for Remote Controller

The μPD17204 is provided with a carrier generation circuit for remote controller.

The carrier generation circuit for remote controller consists of a 6-bit counter, a modulo register for setting high-level period (NRZHTMM), and a modulo register for setting 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 6-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-kHz oscillator (fxT) is used, the input clock frequency is 16 kHz.

The name of the modulo register for setting high-level period is NRZHTMM, and the name of the modulo register for setting 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. When data is read, 0 is forcibly read into the high-order two bits.

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 Fig. 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.

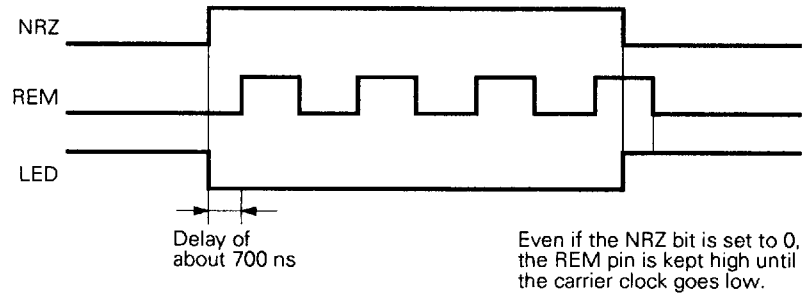


Fig. 10-3 Example of Remote Controller Carrier Output Waveform

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.

Fig. 10-4 NRZ/NRZBF Register File

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 x 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,

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

High-level period ≒ 17

Low-level period ≒ 34

Therefore, the carrier frequency is 37.74 kHz.

10.1.3 Countermeasures Against Noise During Transmission (Carrier Output)

When the μPD17204 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 ohms (r) exists in the power source as shown in the figure below. This resistance reaches 10 to 20 ohms 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).

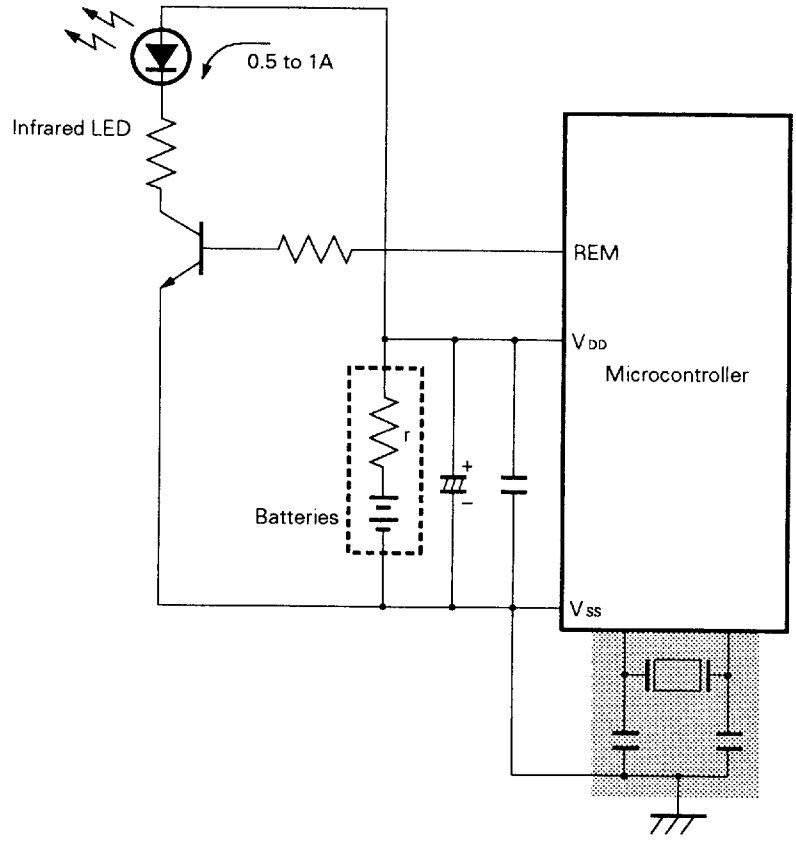
Therefore, comply with the following points to minimize the adverse influence of the noise on the microcontroller:

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.

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.



10.2 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 Section 15.2 "Register File for Interrupt Control").

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.

TM0TM1EN is used to start counting of timer 1 in synchronization with the rising edge to 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, time that elapses until the value set to 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.

The register file for timer 1 is explained below.

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

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

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

TM1RES	10-bit timer reset flag
0	Data read out is always "0"
1	Writing "1" resets 10-bit timer

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

* : Bit 2 is write-only bit.

Fig. 10-5 Control Register for Timer 1 (1/2)

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

Note : 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).

Fig. 10-5 Control Register for Timer 1 (2/2)

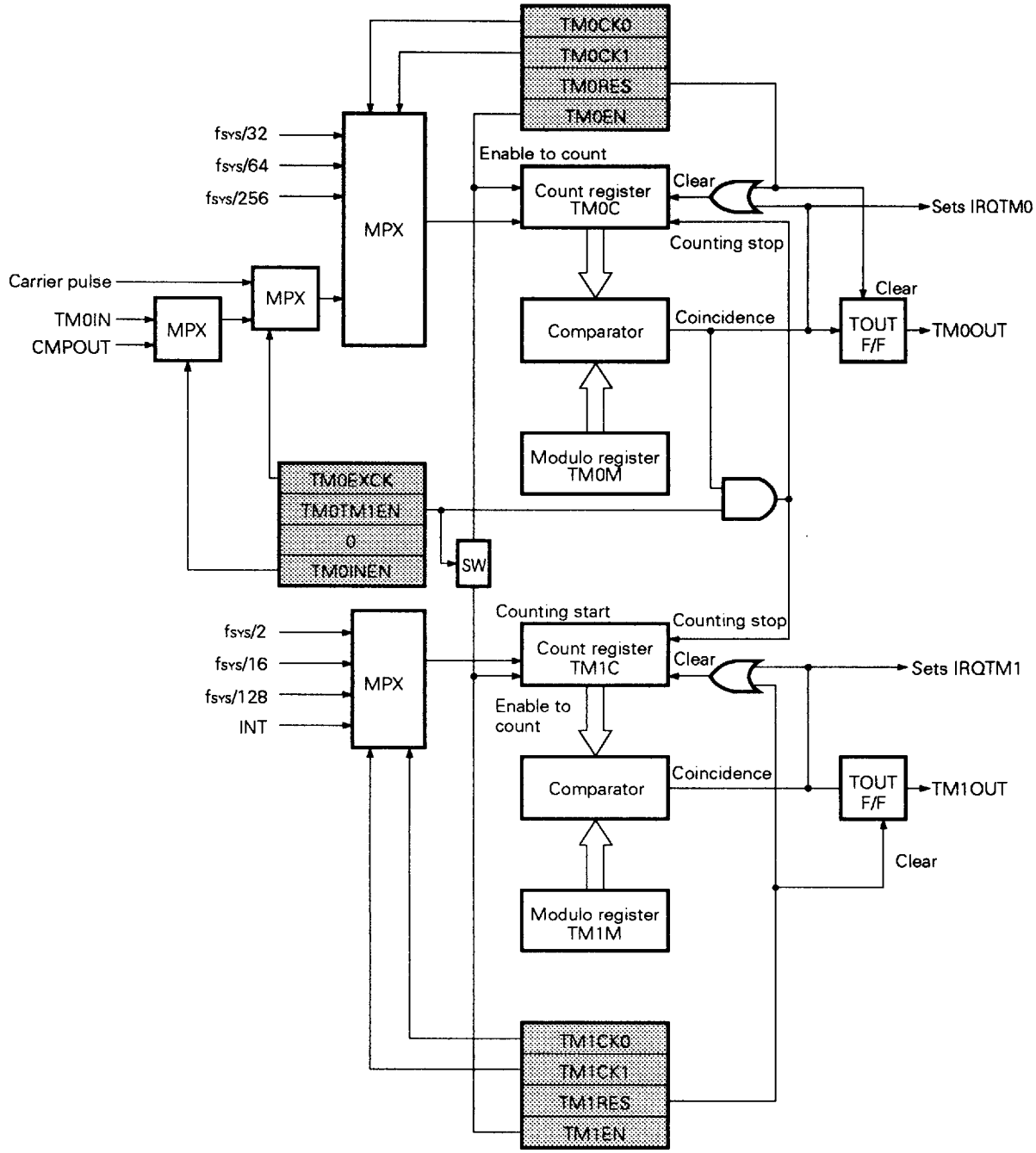


Fig. 10-6 Timer 0, Timer 1 Block Diagram

10.3 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 P1B3 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.

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

TM2CK1	TM2CK0	16-bit timer clock source selection
0	0	Count clock frequency: f_{SYS}/32 (Measurement time range: 8 μs to 524.288 ms)
0	1	Count clock frequency: f_{SYS}/64 (Measurement time range: 16 μs to 1.048576 s)
1	0	Count clock frequency: f_{SYS}/128 (Measurement time range: 32 μs to 2.097152 s)
1	1	Count clock frequency: f_{SYS}/256 (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	Writing "1" resets 16-bit timer

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

* : Bit 2 is write-only bit.

Fig. 10-7 Control Register for Timer 2

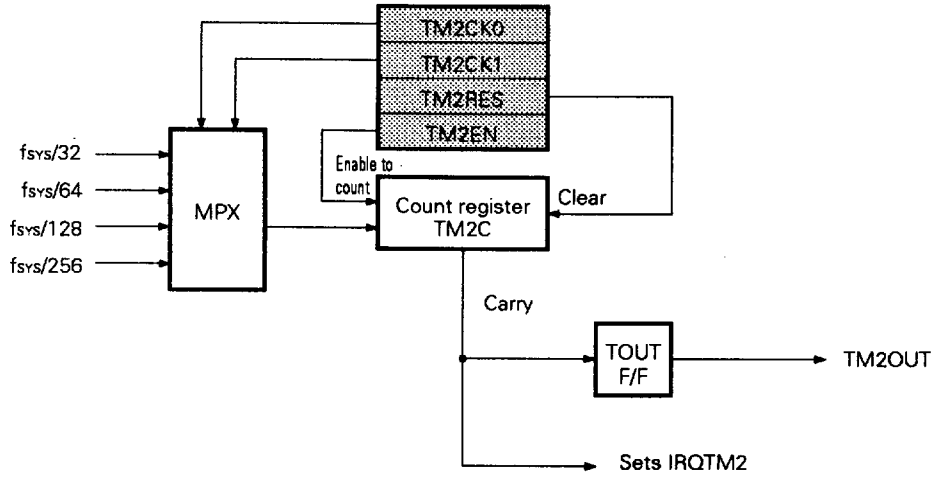


Fig. 10-8 Timer 2 Block Diagram

11. 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.

11.1 CONFIGURATION OF WATCH TIMER/WATCHDOG TIMER

Figure 11-1 shows the configuration of the watch timer/watchdog timer.

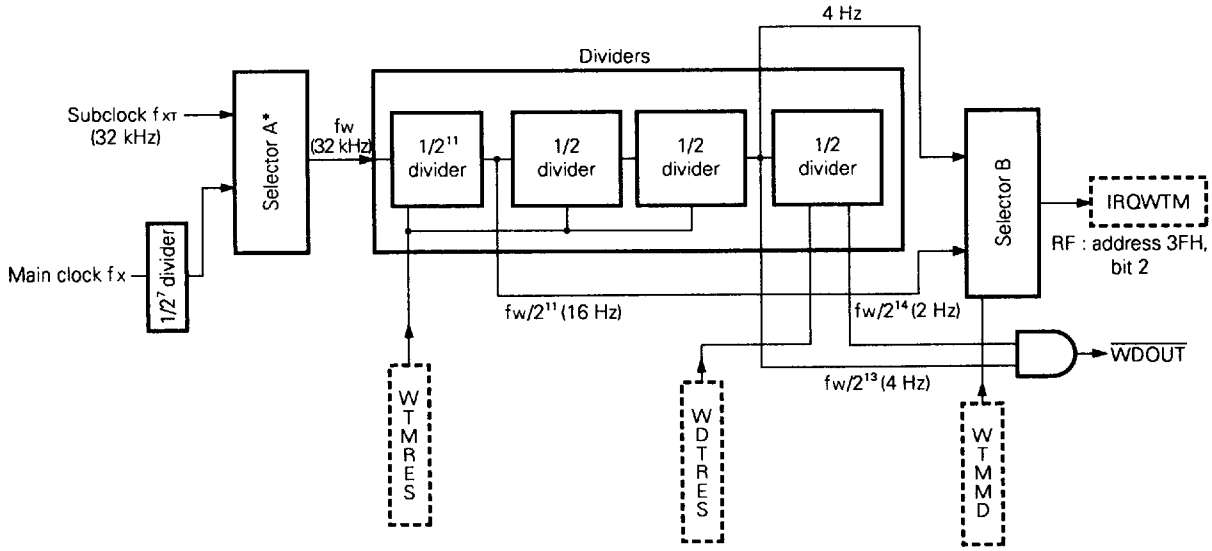
As shown in this figure, the watch timer consists of selector A, which selects the 32-kHz oscillator circuit output of the subclock (XT) or the divided output of the main clock (X, $fx/2^7$) as the source clock, and selector B, which selects the divider of the selected source clock and the frequency 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 ($fx/2^7$) is used as the clock source (when subclock is used), the operation of the watch timer stops when the CPU enters the STOP mode.



* 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 main clock.

Fig. 11-1 Configuration of Watch Timer/Watchdog Timer

11.2 FUNCTION OF WATCH TIMER/WATCHDOG TIMER

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

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

*: Bits 1 and 3 are write-only bits.

Fig. 11-2 Control Register for Clock Timers

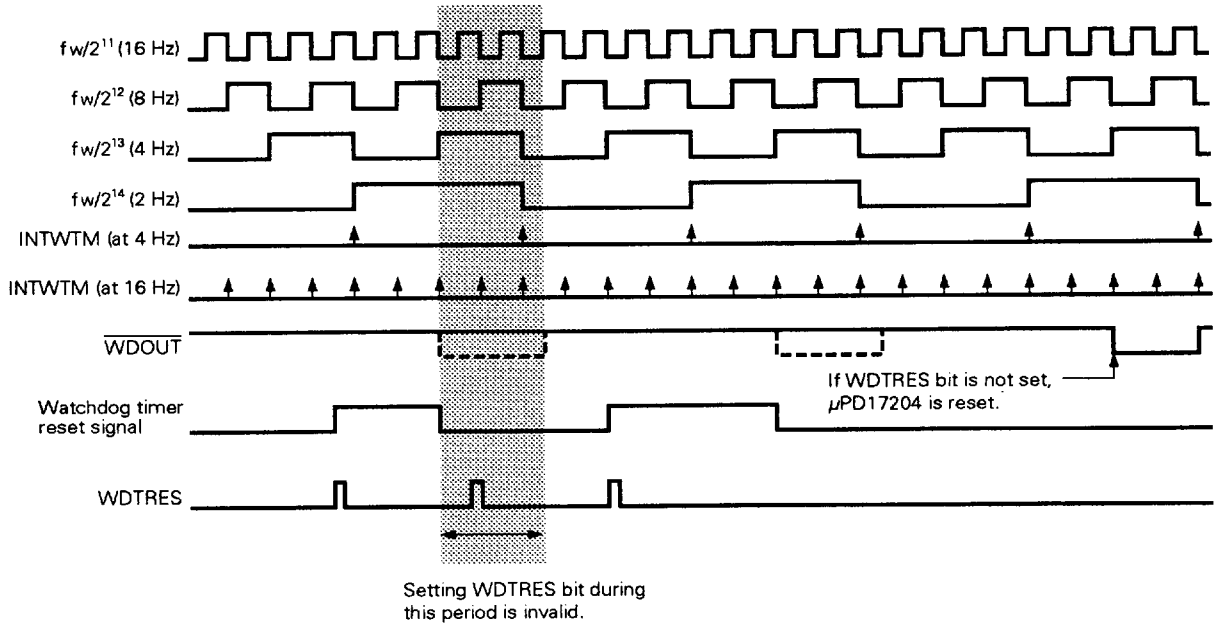
11.3 WATCHDOG TIMER OPERATION TIMING

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.



(( ): when subclock is used)

Note : The watchdog timer cannot be reset within the shaded range in the above figure. Therefore, set WDTRES before both the 2-Hz and 4-Hz signals go high.

Fig. 11-3 Operation Timing of Watchdog Timer

12. CARRIER RECEIVER CIRCUIT FOR REMOTE CONTROLLER

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

The μPD17204 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

12.1.1 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.

12.1.2 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.

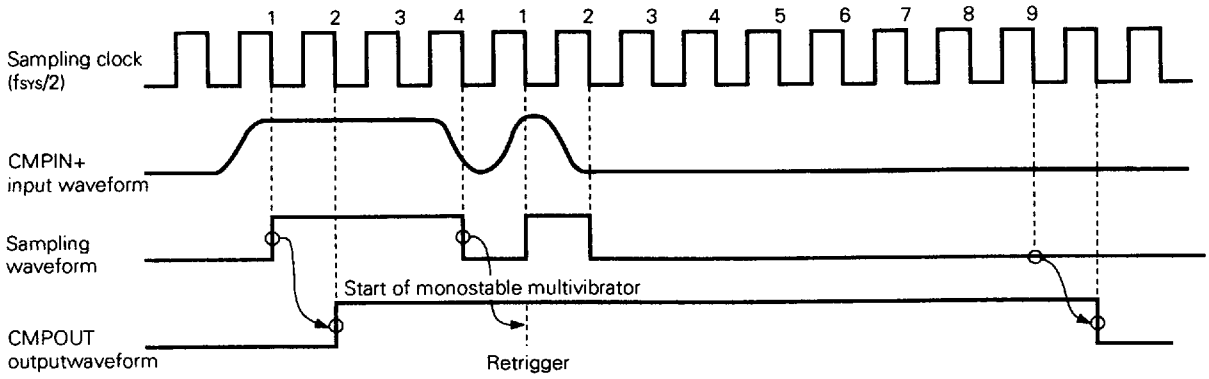


Fig. 12-1 Clock Noise Rejecter Circuit Output

12.1.3 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.

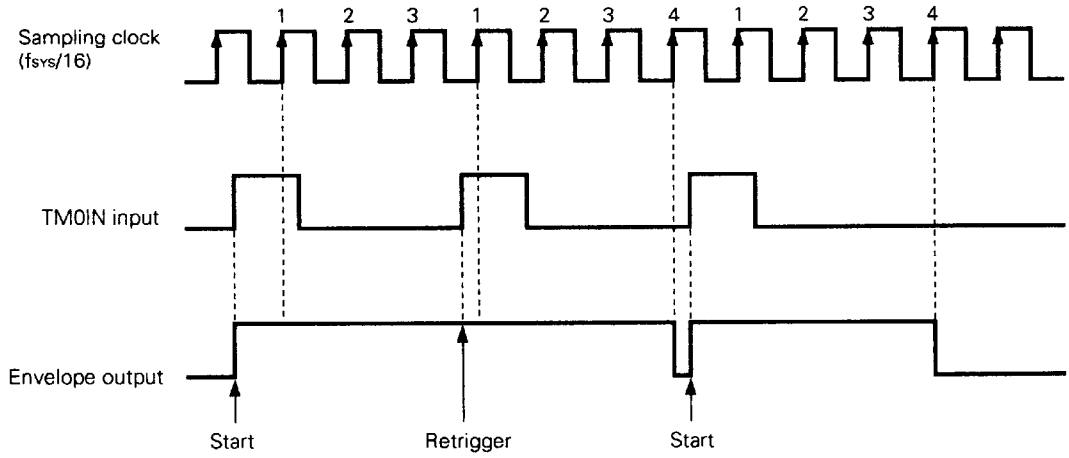
When fsys = fx = 4 MHz, 16, 32, 64, or 128 μs can be set as the maximum envelope output period by bits ENVCK0 and ENVCK1 of the register file.

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.

Note : 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)



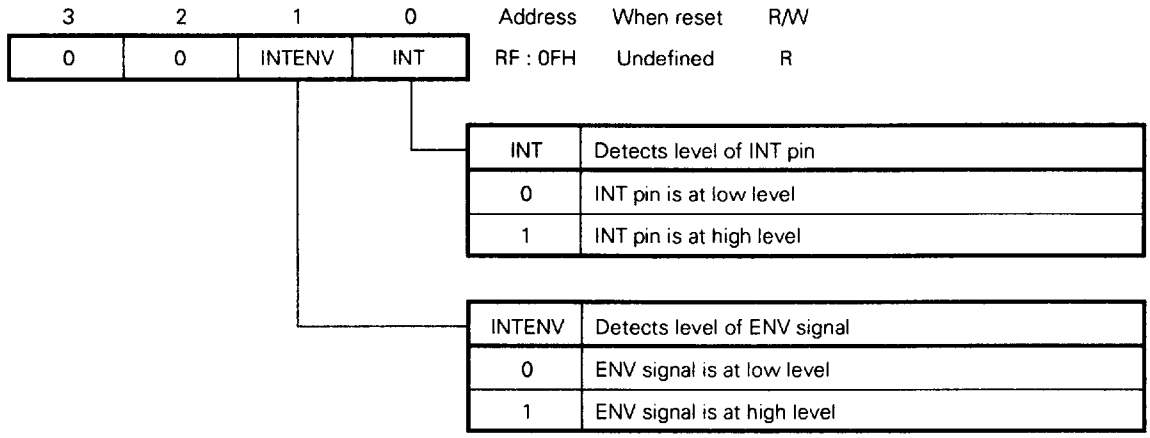
3	2	1	0	Address	When reset	R/W
TM0 INEN	0	ENVCK1	ENVCK0	RF : 13H	0H	R/W

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 noise elimination circuit input
1	Uses TM0IN pin input as noise elimination circuit input

Fig. 12-2 Envelope time Control Register

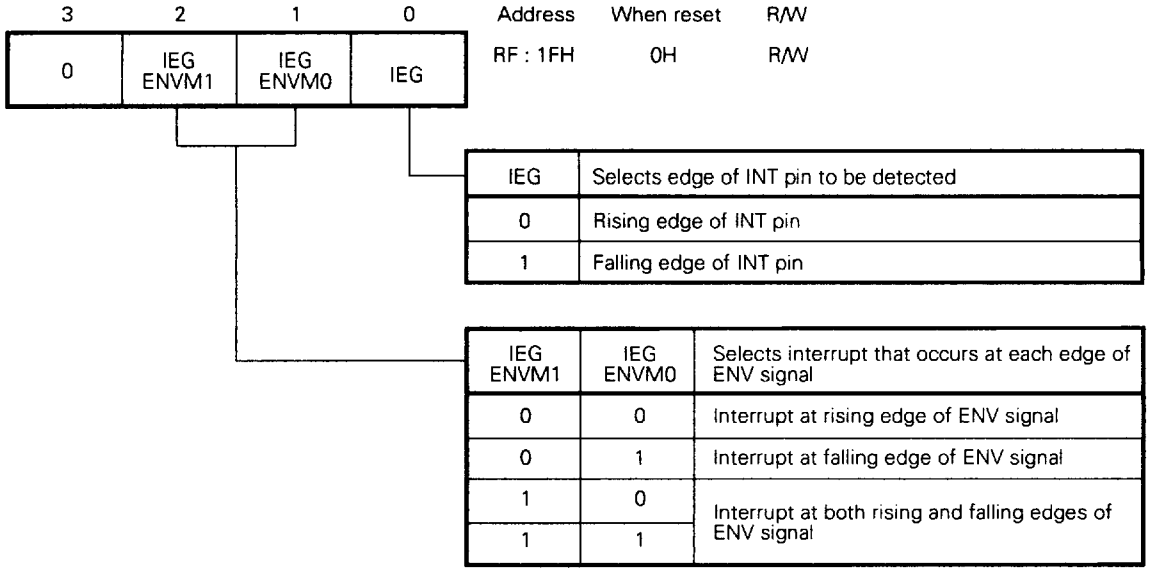


3	2	1	0	Address	When reset	R/W
0	0	INTENV	INT	RF : 0FH	Undefined	R

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

Fig. 12-3 INT and INTENV Flags



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	

Fig. 12-4 Interrupt Detection Edge Selector Flag

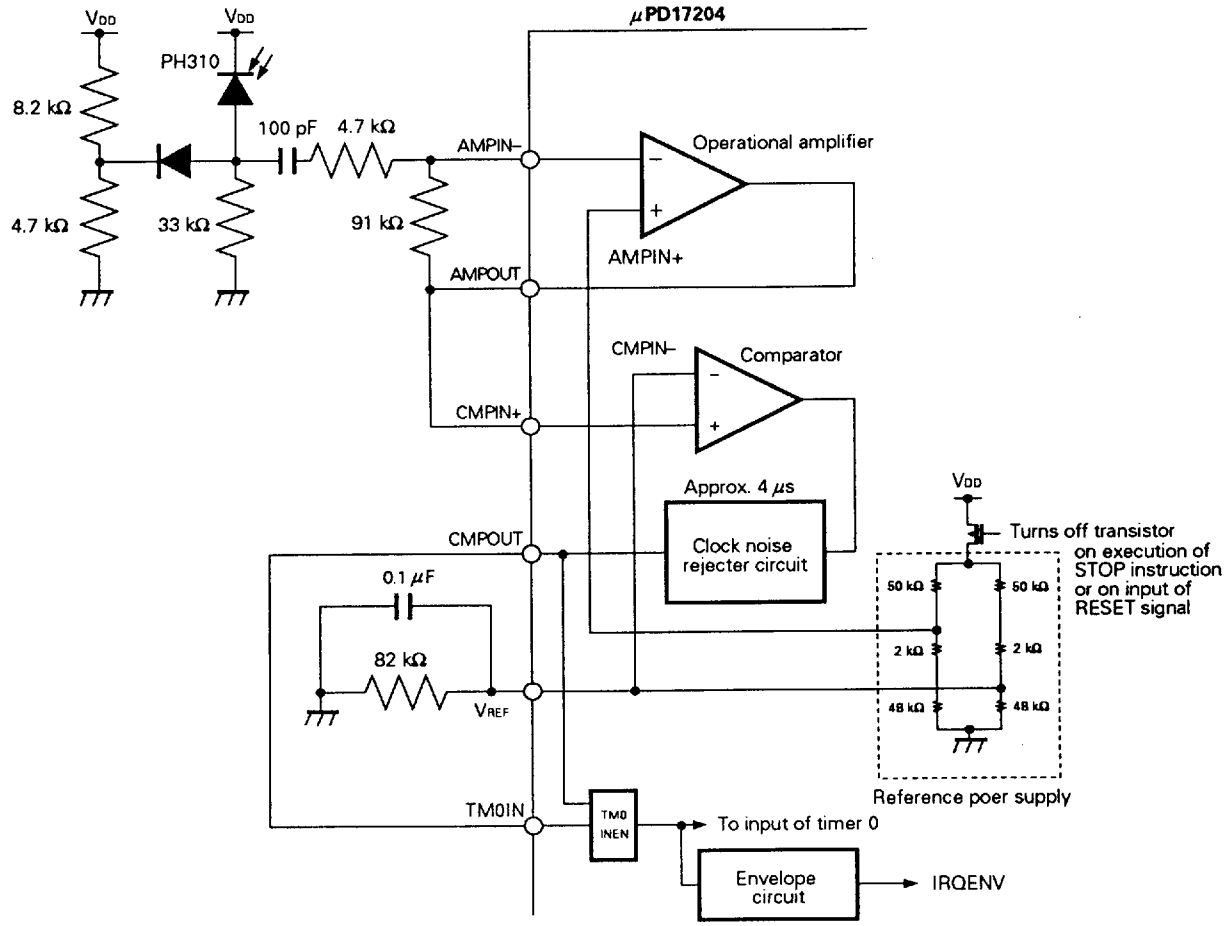


Fig. 12-5 Equivalent Circuit of μPD17204 Internal Preamplifier

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

The application circuits and their parameters are for references only and are not intended for use in actual design-in's.

13. 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.

13.1 SERIAL INTERFACE FUNCTIONS

13.1.1 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).

13.1.2 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.

13.2 SERIAL INTERFACE OPERATION

13.2.1 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.

13.2.2 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 13-1 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.

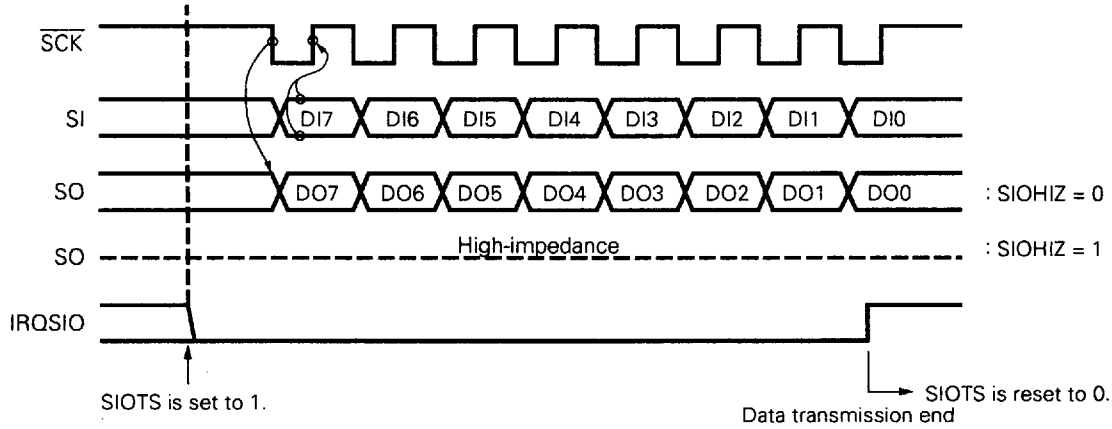
Serial data reception operation is performed in the same manner as serial data transmission operation except for whether data is output from the SO pin.

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.



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

Fig. 13-1 Shift Timing

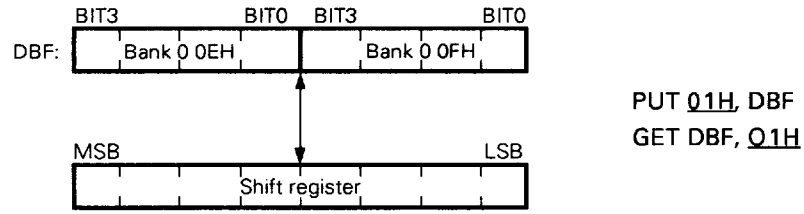


Fig. 13-2 Shift Register Data Transfer

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.

Note : 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.

Fig. 13-3 Serial I/O Control Register

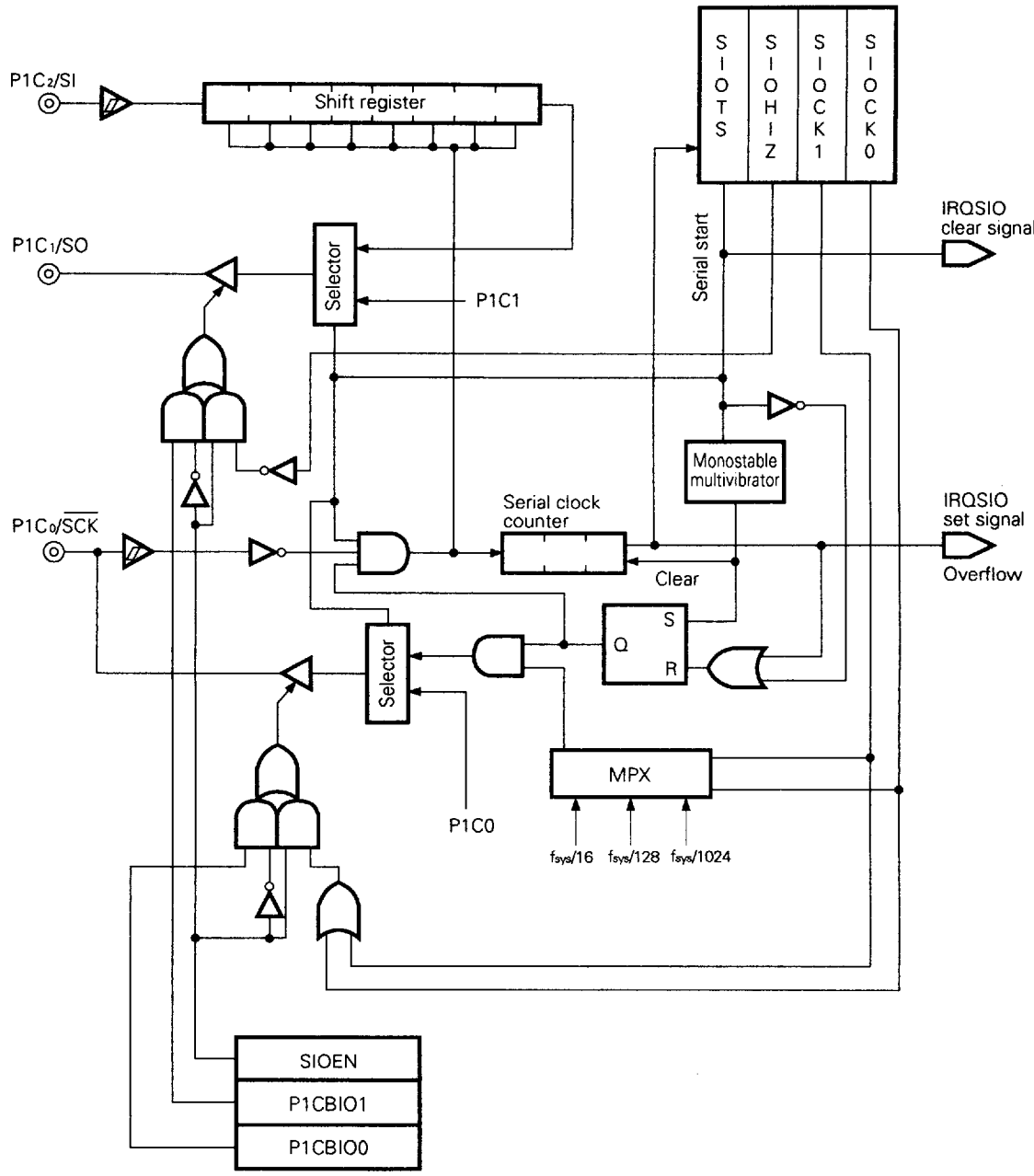


Fig. 13-4 Block Diagram of Serial Interface

14. STATIC RAM (XRAM)

The μPD17204 has been provided with 2048 x 4 bits of static RAM.

To access this memory, an address is set on XRAMSTRT (11 bits) of the register file 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 7FFH, 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.

Note : 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) cleared?
        BR      RAM_ WRITE          ; Yes
                                    ; No
        BANK0                       ;
```

15. INTERRUPT FUNCTION

15.1 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 15-1 shows the correspondence between the interrupt sources and vector addresses.

Table 15-1 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 overflow	Internal	7H
3	Timer 0 overflow	Internal	6H
4	Timer 1 overflow	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.

15.2 REGISTER FILE FOR INTERRUPT CONTROL

The register file for interrupt control is used to read the external interrupt signal and check whether all interrupt requests are available. This file is also used to select the interrupt request edge of the external interrupt signal and specify whether to enable or disable the interrupts corresponding to the interrupt sources. This file is detailed below.

15.2.1 INT

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

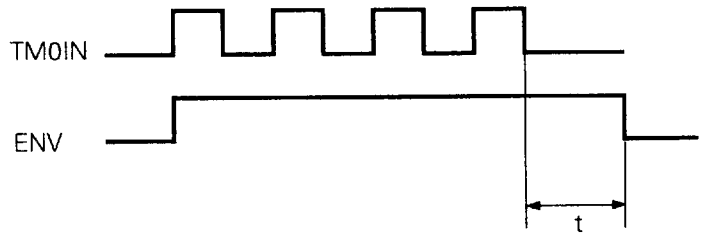
When a high-level signal is input to the INT pin, the INT flag is set to 1. When a low-level signal is input, the INT flag is re-set to 0.

15.2.2 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.



t: Envelope time specified by ENVCK0 and ENVCK1

Fig. 15-1 Waveform of Internal Envelope Circuit Output

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

Fig. 15-2 INT and INTENV Flags

15.2.3 IEG

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

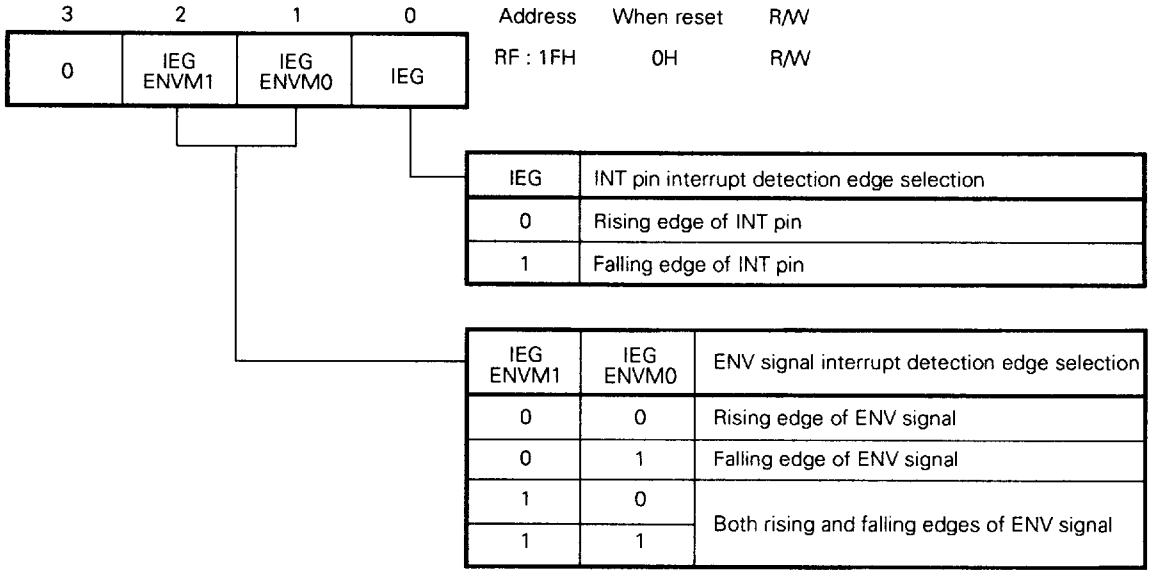
When the IEG flag is reset to 0, the interrupt occurs at the rising edge of the INT pin. When it is set to 1, the interrupt occurs at the falling edge.

15.2.4 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.



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	

Fig. 15-3 IEGENVM0 and IEGENVM1 Flags

15.2.5 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.

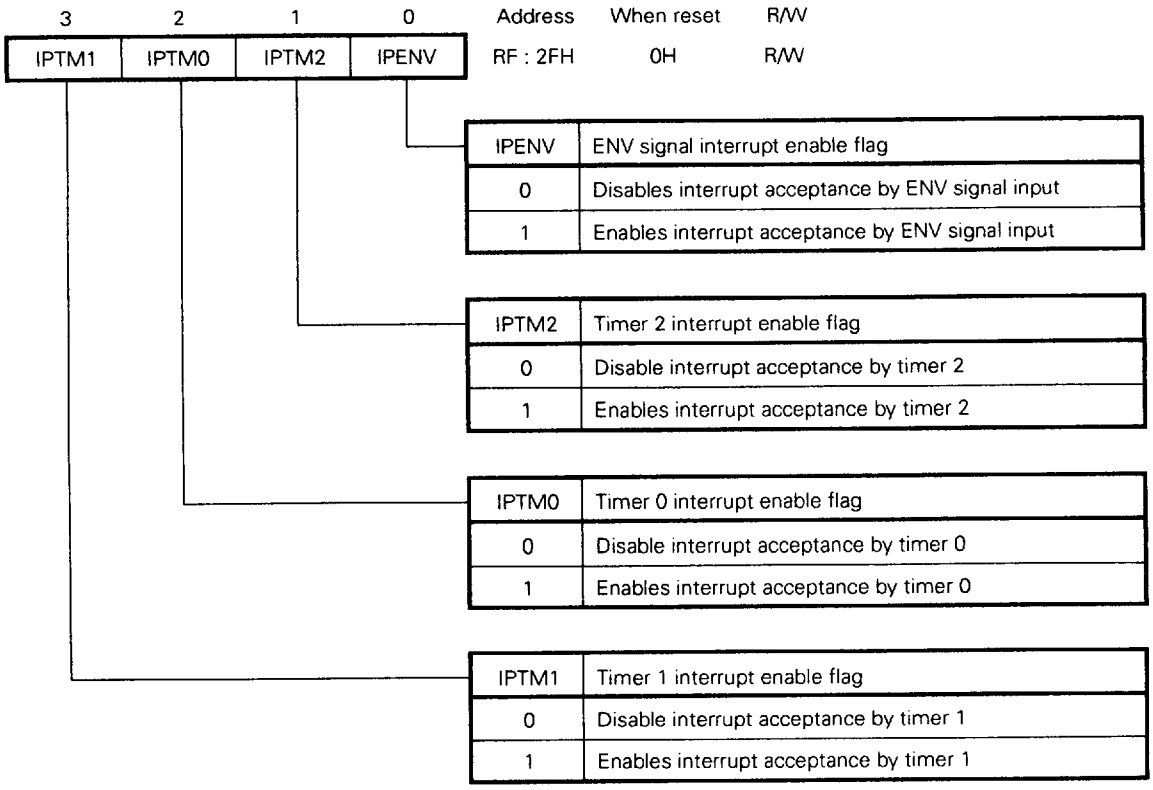


Fig. 15-4 Interrupt Enable Flags (1/2)

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

Fig. 15-4 Interrupt Enable Flag (2/2)

15.2.6 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

15.3 INTERRUPT SEQUENCE

When the IRQ x x flag is set to "1" with the IP x x 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 x x 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 x x flag is "0", the interrupt processing is not executed until the IP x x flag is set even if the IRQ x x 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.

15.3.1 Operation When Interrupt Is Accepted

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

(1) Resets the IRQ x x 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 stack addressed 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.

15.3.2 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 stack addressed 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.

16. STANDBY FUNCTION

The μPD17204 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 μPD17204 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 μPD17204 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 μPD17204 enters the HALT mode. When the STOP instruction is executed, the μPD17204 enters the STOP mode.

16.1 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 16-1.

Table 16-1 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 x x x, IRQ) is issued to the interrupt whose the interrupt enable flag (IP x x x, IP) is set. ② When any of the $P0A_0$ to $P0A_3$ and $P0B_0$ to $P0B_3$ pins goes low.
1010B (0AH)	① When a TM0 (8-bit timer) interrupt request (IRQTM0) is issued (unrelated to IPTM0) ② When an interrupt request (IRQ x x x, IRQ) is issued to the interrupt whose the interrupt enable flag (IP x x x, IP) is set.
Other values	Setting inhibited

16.2 HALT INSTRUCTION EXECUTION CONDITION

The HALT and STOP instructions can be executed only under specific conditions to prevent the program from hang-up. Table 16-2 lists the HALT instruction execution conditions.

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

Table 16-2 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 x x x, IRQ) corresponding to interrupt whose interrupt enable flags (IP x x x, IP) is set must be reset ② All $P0A_0$-$P0A_3$ and $P0B_0$-$P0B_3$ pins must be high
1010B (0AH)	① Interrupt request flag of 8-bit timer TM0 (IRQTM0) must be reset ② Interrupt request flags (IRQ x x x, IRQ) corresponding to interrupt enable flag (IP x x x, IP) must be set
Other values	Setting inhibited

16.3 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 9-4.

After STOP mode has been released, the μPD17204 performs the following processing:

(1) Resets/starts the clock timer counter.
(2) Resets the watchdog timer.
(3) Resets/starts timer counters TM0, TM1, and TM2.
(4) Resets IRQTM0, IRQTM1, IRQTM2, and IRQWTM.
(5) When the value of the 8-bit timer counter (TM0C) coincides with that of the modulo register (TM0M), executes the instruction next to the "STOP 8H" or a vector address branch instruction is executed.

The 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) x 1024/fx [seconds]

Therefore, if the 4-MHz oscillator is used as the main clock, the time that elapses until the next instruction is executed after "STOP 8H" has been released is:

(TM0M + 1) x 256 [microseconds]

Table 16-3 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.
Other values	Setting inhibited

16.4 STOP INSTRUCTION EXECUTION CONDITION

The STOP instruction can be executed only under specific condi-tions to prevent the program from hang-up. Table 16-4 lists the STOP instruction execution conditions.

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

Table 16-4 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.
Other values	Setting inhibited

16.5 OPERATIONS AFTER STANDBY MODE HAS BEEN RELEASED

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

Table 16-5 Operations after Standby Mode Has Been Released

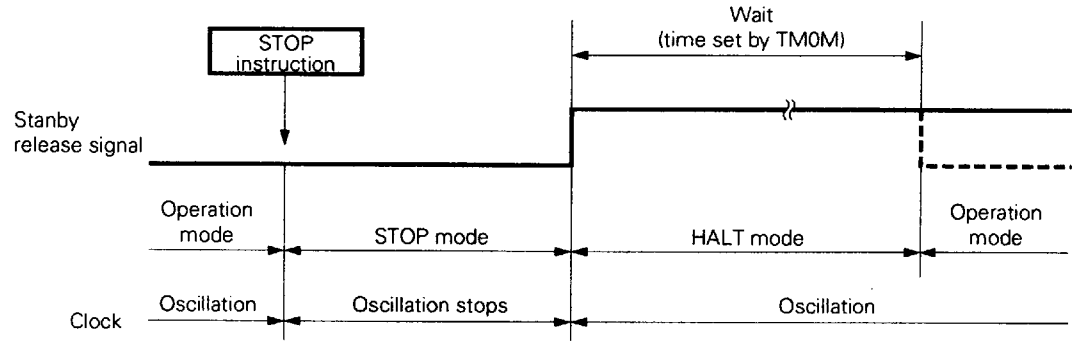
(a) When HALT 08H, 0AH, or STOP 08H is executed

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

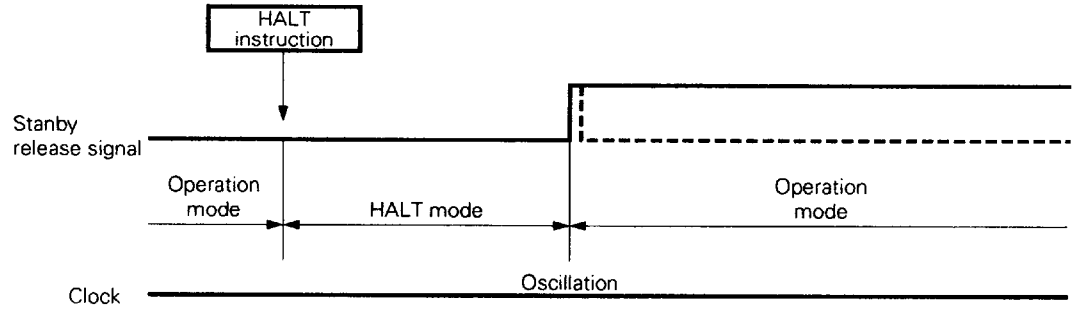
Standby Mode Releasing Conditions	Interrupt Enable Status	Interrupt Enable Flag	Operations After Standby Mode Has Been Released
Setting of IRQTM0	DI	Optional	Executes instruction next to HALT
	EI		Branches to vector address of interrupt

(a) Releasing STOP mode by interrupt



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

(b) Releasing HALT mode by interrupt



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

Fig. 16-1 Operations after Standby Mode Has Been Released

17. RESET

17.1 RESET BY RESET SIGNAL INPUT

The μPD17204 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 μPD17204 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 the description of the register file).
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).

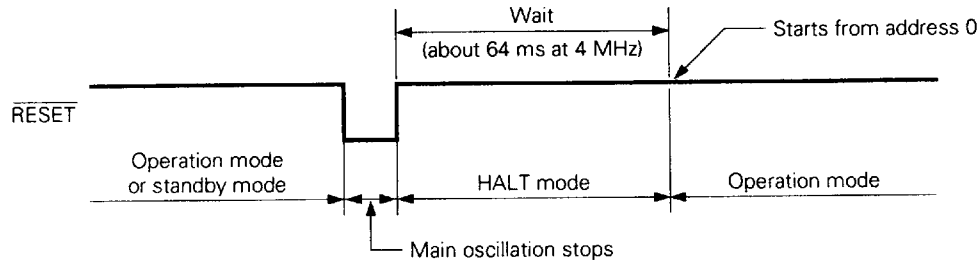


Fig. 17-1 Reset Operation by $\overline{\text{RESET}}$ Input

17.2 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).

17.3 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.

18. ASSEMBLER RESERVED WORDS

18.1 MASK OPTION DIRECTIVE

To code a μPD17204 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

18.1.1 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		

Fig. 18-1 Description Format of Mask Option Definition Block

18.1.2 Mask option definition directives

Table 18-1 lists the mask option definition directives usable in the mask option definition block.

Figure 18-2 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.

Fig. 18-2 Example of Mask Option Definistion

Table 18-1 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)		

18.2 RESERVED SYMBOLS

Table 18-2 lists the reserved symbols defined in the uPD17204 device file.

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

18.2.1 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).

18.2.2 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.

18.2.3 Peripheral circuits

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

Table 18-2 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
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
P0D0	FLG	0.73H.0	R/W	Bit 0 of port 0D
P0D1	FLG	0.73H.1	R/W	Bit 1 of port 0D

Table 18-2 Reserved Symbols (2/4)

Name	Attribute	Value	R/W	Description
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
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

Table 18-2 Reserved Symbols (3/4)

Name	Attribute	Value	R/W	Description
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
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

Table 18-2 Reserved Symbols (4/4)

Name	Attribute	Value	R/W	Description
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

19. μPD17204 INSTRUCTION SET

19.1 OUTLINE OF INSTRUCTION SET

b_{14}-b_{11} \ b_{15}		0		1	
BIN	HEX				
0000	0	ADD	r, m	ADD	m, #i
0001	1	SUB	r, m	SUB	m, #i
0010	2	ADDC	r, m	ADDC	m, #i
0011	3	SUBC	r, m	SUBC	m, #i
0100	4	AND	r, m	AND	m, #i
0101	5	XOR	r, m	XOR	m, #i
0110	6	OR	r, m	OR	m, #i
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, RA RA, WR r s h		
1000	8	LD	r, m	ST	m, r
1001	9	SKE	m, #i	SKGE	m, #i
1010	A	MOV	@r, m	MOV	m, @r
1011	B	SKNE	m, #i	SKLT	m, #i
1100	C	BR	addr (Page 0)	CALL	addr (Page 0)
1101	D	BR	addr (Page 1)	MOV	m, #i
1110	E	BR	addr (Page 2)	SKT	m, #n
1111	F	BR	addr (Page 3)	SKF	m, #n

19.2 LEGEND

M : Data memory address
m : Data memory address (excluding banks)
m_H : Data memory row address
m_L : Data memory column address
R : General register address
r : General register column address
RP : General register pointer
RF : Register file
rf : Register file address
rf_H : Register file address (high-order 3 bits)
rf_L : Register file address (low-order 3 bits)
AR : Address register
IX : Index register
IXE : Index enable flag
DBF : Data buffer
WR : Window register
MP : Data memory row address pointer
MPE : Memory pointer enable flag
PE : Peripheral register
p : Peripheral address
p_H : Peripheral address (high-order 3 bits)
p_L : Peripheral address (low-order 4 bits)
PC : Program memory counter
SP : Stack pointer
STACK : Stack value indicated by stack pointer
BANK : Bank register
(AR) rom : Data in program memory indicated by address register
INTEF : Interrupt enable flag
i : Immediate data (4 bits)
n : Bit position (4 bits)
addr : Program memory address (11 bits)
CY : Carry flag
CMP : Compare flag
s : Stop release condition
h : Halt release condition
[ ] : Data memory or register address
( ) : Data memory or register value

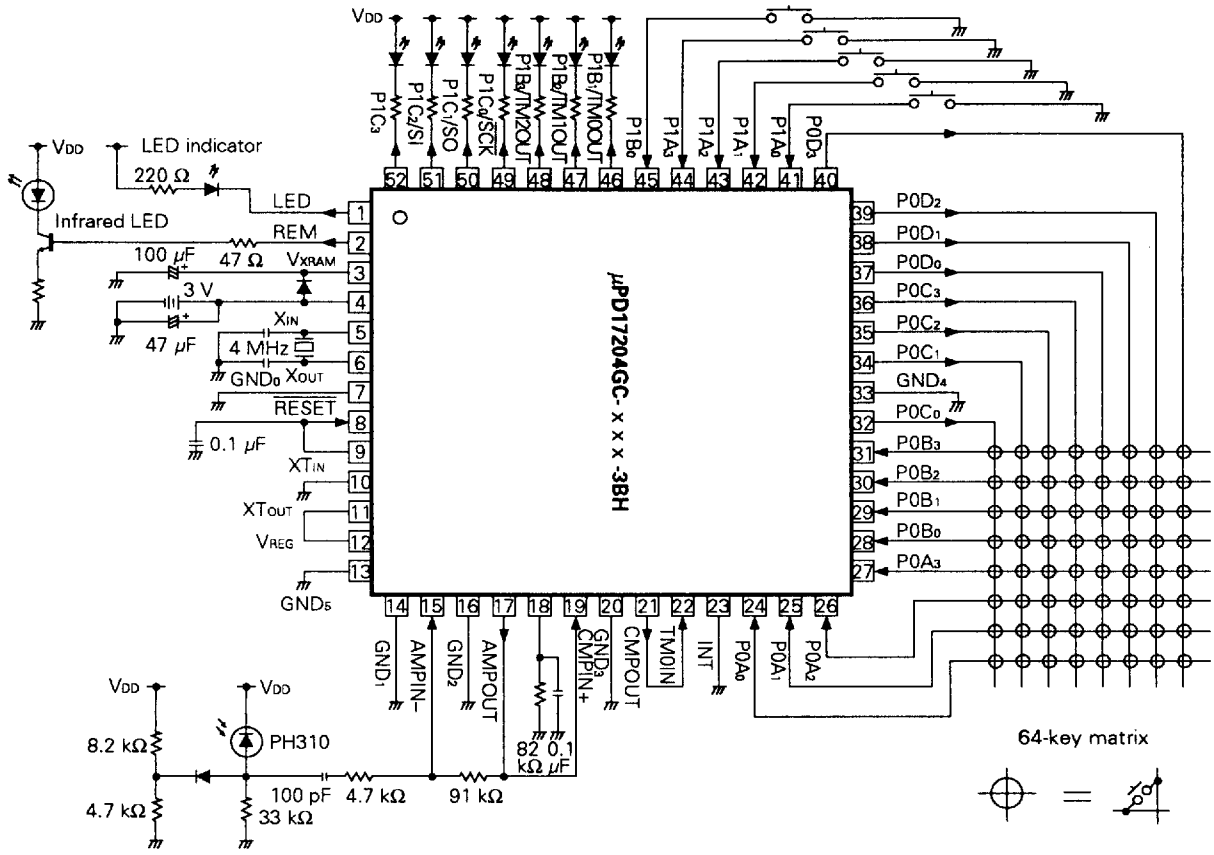
19.3 INSTRUCTION SET

Instructions	Mnemonic	Operand	Operation	Machine Code			
				OP Code	3-bit	4-bit	4-bit
Addition	ADD	r, m	(R), CY ← (R)+(M)	00000	m_H	m_L	r
		m, #i	(M), CY ← (M)+i	10000	m_H	m_L	i
	ADDC	r, m	(R), CY←(R)+(M)+(CY)	00010	m_H	m_L	r
		m, #i	(M), CY ← (M)+i+(CY)	10010	m_H	m_L	i
	INC	AR	(AR) ← (AR)+1	00111	000	1001	0000
		IX	(IX) ← (IX)+1	00111	000	1000	0000
Subtraction	SUB	r, m	(R), CY ← (R)–(M)	00001	m_H	m_L	r
		m, #i	(M), CY ← (M)–i	10001	m_H	m_L	i
	SUBC	r, m	(R), CY ← (R)–(M)–(CY)	00011	m_H	m_L	r
		m, #i	(M), CY ← (M)–i–(CY)	10011	m_H	m_L	i
Comparison	SKE	m, #i	(M)–i, skip if zero	01001	m_H	m_L	i
	SKGE	m, #i	(M)–i, skip if not borrow	11001	m_H	m_L	i
	SKLT	m, #i	(M)–i, skip if borrow	11011	m_H	m_L	i
	SKNE	m, #i	(M)–i, skip if not zero	01011	m_H	m_L	i
Logical operation	AND	m, #i	(M) ← (M) AND i	10100	m_H	m_L	i
		r, m	(R) ← (R) AND (M)	00100	m_H	m_L	r
	OR	m, #i	(M) ← (M) OR i	10110	m_H	m_L	i
		r, m	(R) ← (R) OR (M)	00110	m_H	m_L	r
	XOR	m, #i	(M) ← (M) XOR i	10101	m_H	m_L	i
		r, m	(R) ← (R) XOR (M)	00101	m_H	m_L	r
Transfer	LD	r, m	(R) ← (M)	01000	m_H	m_L	r
	ST	m, r	(M) ← (R)	11000	m_H	m_L	r
	MOV	@r, m	if MPE=1, [(MP), (R)] ← (M) if MPE=0, [(m_H), (R)] ← (M)	01010	m_H	m_L	r
		m, @r	if MPE=1, (M) ← [(MP), (R)] if MPE=0, (M) ← [(m_H), (R)]	11010	m_H	m_L	r
		m, #i	M ← i	11101	m_H	m_L	i
	MOVT	DBF, @AR	SP ← (SP)–1, STACK ← PC DBF ← (AR)rom, PC ← STACK, SP ← (SP)+1	00111	000	0001	0000
	PUSH	AR	(SP) ← (SP)–1, (STACK) ← (AR)	00111	000	1101	0000
	POP	AR	AR ← (STACK), (SP) ← (SP)+1	00111	000	1100	0000
	PEEK	WR, RA	(WR) ← (RF)	00111	rf_H	0011	rf_L
	POKE	RA, WR	(RF) ← (WR)	00111	rf_H	0010	rf_L
	GET	DBF, p	(DBF) ← (PE)	00111	p_H	1011	p_L
	PUT	p, DBF	(PE) ← (DBF)	00111	p_H	1010	p_L

(cont'd)

Instructions	Mnemonic	Operand	Operation	Machine Code			
				OP Code	3-bit	4-bit	4-bit
Judgment	SKT	m, #n	CMP ← 0 skip if M_n=all"1"	11110	m_H	m_L	n
	SKF	m, #n	CMP ← 0 skip if M_n=all"0"	11110	m_H	m_L	n
Branch	BR	addr	PC ← addr, PAGE ← 00	01100	a_H	a_M	a_L
			PC ← addr, PAGE ← 01	01101	a_H	a_M	a_L
			PC ← addr, PAGE ← 10	01110	a_H	a_M	a_L
			PC ← addr, PAGE ← 11	01111	a_H	a_M	a_L
		@AR	PC ← AR	00111	000	0100	0000
Shift	RORC	r	→ (CY) → R (3) → R (2) → R (1) → R (0) →	00111	000	0111	r
Subroutine	CALL	addr	SP ← (SP)−1, STACK ← ((PC)+1), PC ← ADDR, PAGE ← 00	11100	a_H	a_M	a_L
		@AR	SP ← (SP)−1 STACK ← ((PC)+1), PC ← (AR)	00111	000	0101	0000
	RET		PC ← (STACK), SP ← (SP)+1	00111	000	1110	0000
	RETSK		PC ← (STACK), SP ← (SP)+1 and skip	00111	001	1110	0000
	RETI		PC ← (STACK), SP ← (SP)+1	00111	100	1110	0000
Interrupt	EI		INTEF ← 1	00111	000	1111	0000
	DI		INTEF ← 0	00111	001	1111	0000
Others	STOP	8H	STOP	00111	010	1111	1000
	HALT	h	HALT	00111	011	1111	h
	NOP		No operation	00111	100	1111	0000

20. APPLICATION CIRCUIT EXAMPLE



The application circuits and their parameters are for references only and are not intended for use in actual design-in's.

21. ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS (Ta = 25°C)

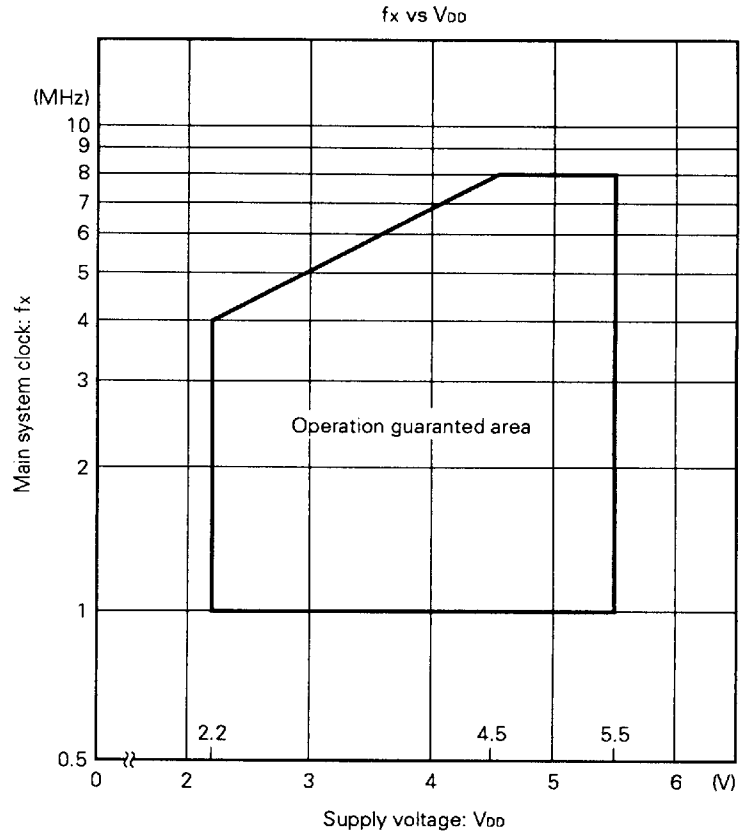
Item	Symbol	Condition		Rating	Unit
Supply Voltage	V_{DD}			−0.3 to +7.0	V
Input Voltage	V_I			−0.3 to V_{DD} + 0.3	V
Operating Temperature	T_{opt}			−20 to +75	°C
Storage Temperature	T_{stg}			−40 to +125	°C
High-level Output Current	I_{OH1}	REM pin	Peak value	−30.0	mA
	I_{OH2}		Effective value	−20.0	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

CAPACITANCE (Ta = 25°C, V_{DD} = 0V)

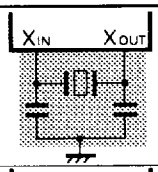
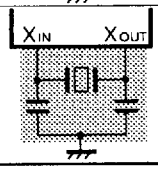
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input Capacitance	C_{IN}	INT and $\overline{\text{RESET}}$ pins			10	pF
	C_{PIN}	Otehr than INT and $\overline{\text{RESET}}$ pins			10	pF

RECOMMENDED OPERATING RANGE (Ta = –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} = 32kHz	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 (Ta = –20 to +75°C, V_{DD} = 2.2 to 5.5V)

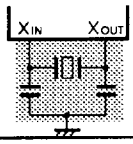
Resonator	Recomended Constants	Item	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic Oscillator *3		Oscillation frequency (fx) *1		1.0	4.0	8.0	MHz
		Oscillation stabilization time *2	From when V_{DD} reaches the minimum oscillation voltage			4	ms
Crystal Oscillator *3		Oscillation frequency (fx) *1		1.0	4.0	8.0	MHz
		Oscillation stabilization time *2	V_{DD} = 4.5 to 6.0 V			10	ms
						30	ms

*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 described later.

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

Resonator	Recomended Constants	Item	Conditions	MIN.	TYP.	MAX.	Unit
Crystal Oscillator		Oscillation frequency (fXT)			32.768		kHz
		Oscillation stabilization time				10	s

Note : 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 shaded area 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

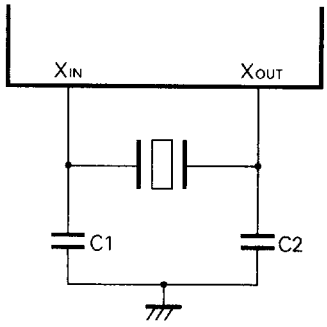
Main system clock: ceramic oscillator

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	30	30	2.0	6.0	
	CSA4.19MG	30	30	2.0	6.0	
	CST3.58MGW	Not required	Not required	2.0	6.0	C contained type
	CST4.00MGW	Not required	Not required	2.0	6.0	
	CST4.19MGW	Not required	Not required	2.0	6.0	
KYOCERA	KBR3.58MS	33	33	2.0	6.0	
	KBR4.0MS	33	33	2.0	6.0	
	KBR4.19MS	33	33	2.0	6.0	
TOKO	CRHF4.00	18	18	2.0	6.0	
DAISHINKU	PRS0400BCSAN	39	33	2.0	6.0	

Main system clock: 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

(V_{DD} = 3V, Ta = −20 to +75°C, fx = 4 MHz, f_{XT} = 32 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

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

Cont'd

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Supply Current	I_{DD1}	Operation mode	Bothe 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 oscillates.			2.0	mA
	I_{DD4}*		Only XT oscillate		7	15	μA
XRAM Golding Voltage	V_{XRAM}			1.3	3.0	5.5	V
XRAM Supply Current	I_{XRAM1}	Operation mode, VXRAM = 3 V		3.0	5.0	7.0	μA
	I_{XRAM2}	HALT mode, VXRAM = 3 V, Ta = 25°C			0.2	1.0	μA

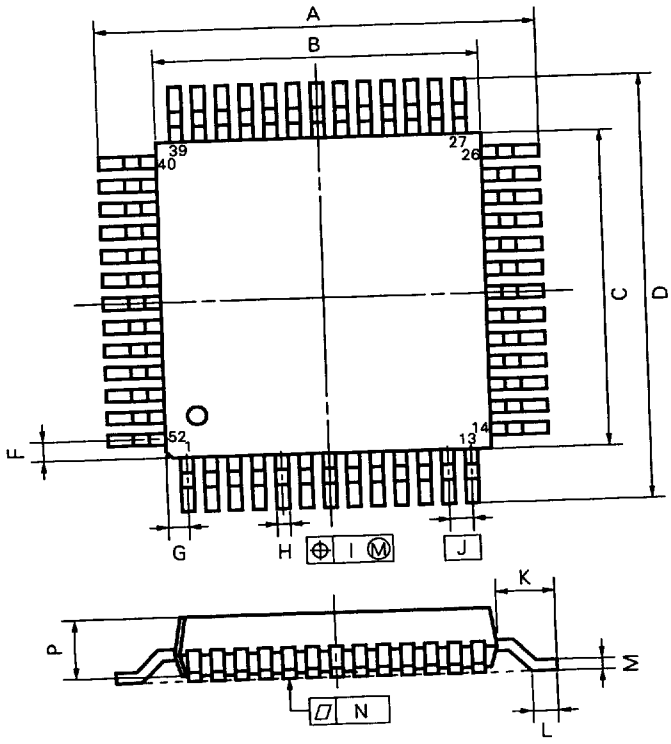
OPERATIONAL AMPLIFIER/COMPARATOR CHARACTERISTICS

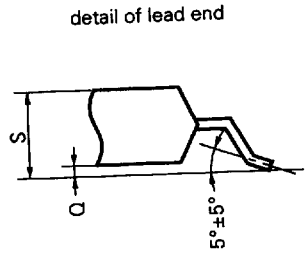
(V_{DD} = 3V, Ta = –20 to +75°C, fx = 4 MHz, f_{XT} = 32 kHz)

Item	Symbol	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

22. PACKAGE DRAWINGS

52 PIN PLASTIC QFP (□14)



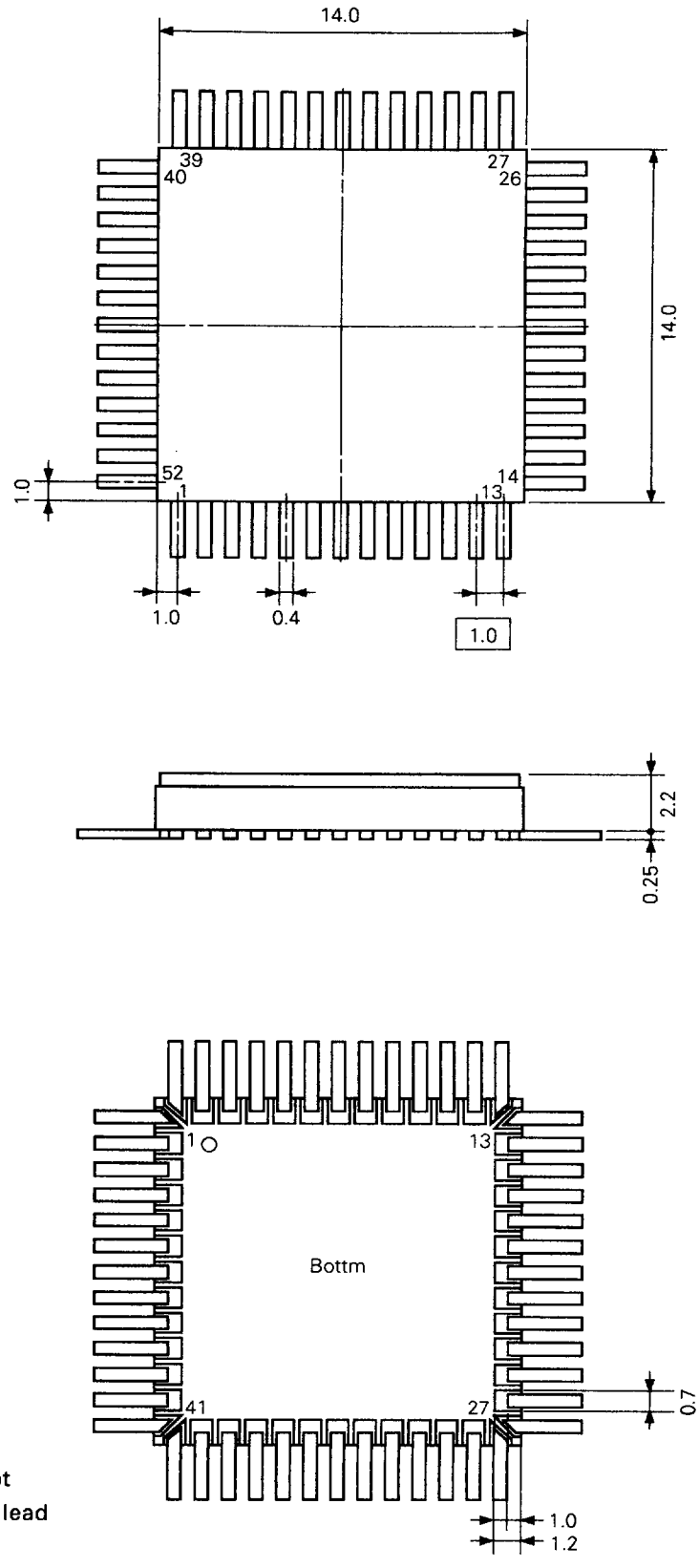


S52GC-100-3BH-1

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±0.4	0.677±0.016
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±0.4	0.677±0.016
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	0.12	0.005
P	2.7	0.106
Q	0.1±0.1	0.004±0.004
S	3.0 MAX.	0.119 MAX.

52-PIN CERAMIC QFP (14 sq.) (for ES) (UNIT: mm)



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

X52B-100B

23. RECOMMENDED SOLDERING CONDITIONS

When mounting the μPD17204 by soldering, soldering should be performed under the following recommended conditions.

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

Table 23-1 Soldering Conditions

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

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

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

Remarks : For details on recommended soldering conditions refer to the information document "Surface mount device mounting manual" (IEI-616).

APPENDIX A. LIST OF MICROCONTROLLERS FOR LEARNING REMOTE CONTROLLER

Item / Product Name		μPD17203A	μPD17P203A	μPD17204	μPD17P204
ROM Capacity		8K bytes (4096 x 16)		16K bytes (7936 x 16)	
		(Mask ROM)	(One-time PROM)	(Mask ROM)	(One-time PROM)
RAM Capacity		336 x 4 bits			
Static RAM Capacity		4096 x 4 bits		2048 x 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 (Operation Voltage) Ta = −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*)	(V_{DD} = 2.2 to 5.5 V)	(V_{DD} = 2.9 to 5.5 V*)
	Sub System Clock	488 μs : @32.768 kHz (V_{DD} = 2.0 to 5.5 V)			
Package		52-pin plastic QFP			

* : The operating voltage range varies with the operating temperature range. For details, refer to 21. Electrical Characteristics.

APPENDIX B DEVELOPMENT TOOLS

The following tools are available for μPD17204 program development.

Hardware

Name	Outline	Order Code
In-Circuit Emulator	1E-17K is an in-circuit emulator commonly usable for the 17K Series. To develop a μPD17204 program, use 1E-17K with system evaluation board (SE board). IE-17K is a tool operating on the RAM base. Simply connecting a console to IE-17K enables prompt addition and modification of the program on the console. In addition, support software SIMPLEHOST™ offers program development environments of higher level.	IE-17K IE-17K-ET*
SE Board	SE-17204 is a system evaluation board. This board can be used by itself or together with IE-17K.	SE-17204
Emulation Probe	EP-17203GC is a probe used to connect the SE-17204 and target system.	EP-17203GC
Conversion Socket	Used in combination with the emulation probe to connect the target system.	EV-9200G-52
One-Time PROM	μPD17P204 is a one-time PROM (OTP) for μPD17204 program evaluation and small-scale production of μPD17204 system.	μPD17P204GC-001-3BH μPD17P204GC-002-3BH μPD17P204GC-003-3BH

* : Low-price model : external power supply type

Software

Name	Outline	Host Machine	OS		Supply Media	Order Name
17K Series Assembler (AS17K)	AS17K is an assembler commonly usable with all 17K Series products. To develop a μPD17204 program, use this AS17K with a device file (AS17204).	PC-9800 Series	MS-DOS™ (Ver.3.10 to Ver.3.30C)		5"2HD	μS5A10AS17K
					3.5"2HD	μS5A13AS17K
		IBM PC/AT™	PC DOS™ (Ver.3.1)		5"2HC	μS7B10AS17K
Device File (AS17204)	AS17204 is a μPD17204 dedicated device file. Use this file with the assembler (AS17K) common to all 17K Series products.	PC-9800 Series	MS-DOS (Ver.3.10 to Ver.3.30C)		5"2HD	μS5A10AS17204
					3.5"2HD	μS5A13AS17204
		IBM PC/AT	PC DOS (Ver.3.1)		5"2HC	μS7B10AS17204
Support Software (SIMPLEHOST)	SIMPLEHOST is software that performs machine interface on MS-WINDOWS™ when IE-17K and a personal computer are used to develop a program.	PC-9800 Series	MS-DOS	MS-WINDOWS (Ver.2.1 to Ver.3.0)	5"2HD	μS5A10IE17K
					3.5"2HD	μS5A13IE17K
		IBM PC/AT	PCDOS		5"2HC	μS7B10IE17K