HD404418 Series

Description

The HD404418 Series of 4-bit single-chip microcomputers are basically equivalent to the HMCS400 series providing high programming productivity and high-speed operation. The devices incorporate ROM, RAM, I/O, four timer/counters, and two serial interfaces.

The HD404418 Series includes three chips. All pins for the HD404418 and HD4074418 are CMOS standard, and the HD4074408 includes 8 intermediate-voltage pins (+12 V).

The HD4074408 and HD4074418 are PROM versions (ZTAT™ microcomputer). A program can be written to the PROM by a PROM writer, which can dramatically shorten system development periods and smooth the process from debugging to mass production. (The ZTAT™ version is 27256-compatible.)

Features

- 8,192-word × 10-bit ROM (HD404418)
- 8,192-word × 10-bit PROM (HD4074418, HD4074408)
- 512-digit × 4-bit RAM
- 58 I/O pins including 4 input-only pins, 16 high current pins (100 mA max.), and 2 NMOS open-drain pins
  - — HD404418 and HD4074418: All CMOS standard pins
  - — HD4074408: Includes 8 NMOS open-drain intermediate-voltage pins (+12.8 V)
- Four timer/counters
- Four analog inputs
- Two-channel 8-bit clock-synchronous serial interfaces
- 12 interrupt sources
  - — Six by external sources
  - — Six by internal sources
- Subroutine stack up to 16 levels, including interrupts
- Two low-power dissipation modes
  - — Standby mode
  - — Stop mode
- On-chip oscillator: Crystal or ceramic oscillator (an external clock is also possible)
- Instruction cycle time: 0.89 µs (f_{OSC} = 9 MHz)
- Package
  - — 64-pin shrink-type ceramic DIP with window
  - — 64-pin shrink-type plastic DIP
  - — 64-pin flat plastic package

Ordering Information

Type	Product Name	Model Name	Package
Mask ROM	HD404418	HD404418S (01)*	DP-64S
		HD404418F (01)*	FP-64
		HD404418H (01)*	FP-64A
ZTAT™	HD4074408	HD4074408S (01)*	DP-64S
		HD4074408C (01)*	DC-64S
		HD4074408F (01)*	FP-64
		HD4074408H (01)*	FP-64A
	HD4074418	HD4074418S (01)*	DP-64S
		HD4074418C (01)*	DC-64S
		HD4074418F (01)*	FP-64
		HD4074418H (01)*	FP-64A

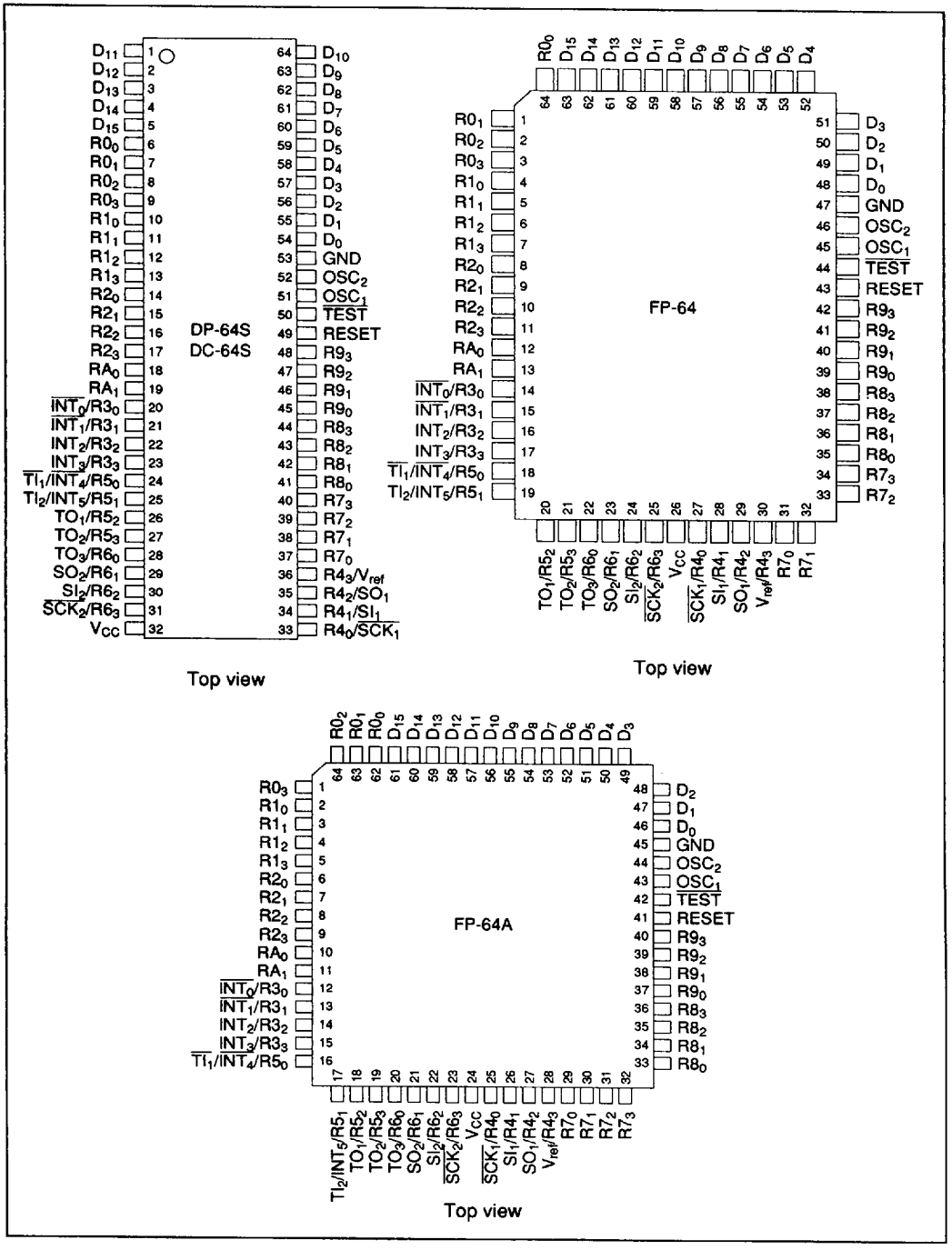
Note: * Watchdog timer version

Differences between HD404418 and HD4074418/HD4074408

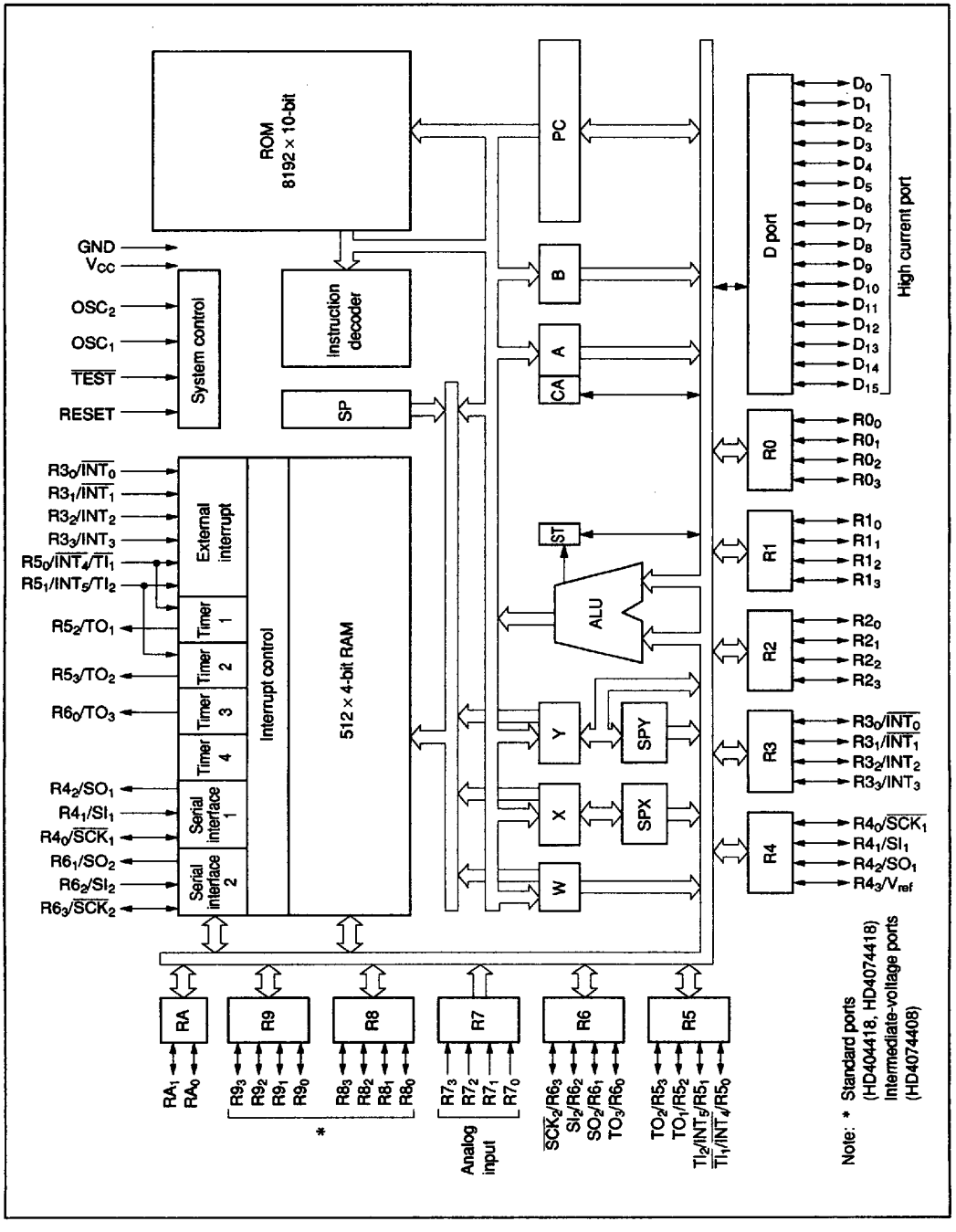
Item	HD4074418 (ZTAT™) HD4074408 (ZTAT™)	HD404418 (Mask ROM)
Input/output leakage current	RA_0/V_{PP}: 10 µA max. Other standard pins: 1 µA max.	All standard pins: 1 µA max.
Input pin capacitance	RA_0/V_{PP}: 70 pF max. Other pins: 15 pF max.	All pins: 15 pF max.
I/O option	None: All pins are without pull-up MOS	With pull-up MOS can be specified (D_0 to D_{15} and R0 to R6)
Timer option	None: One of two options provided for each product can be selected	Free-running timer, watchdog timer
Intermediate-voltage port version (R8 and R9)	HD4074408	None

Precaution: The HD404418 differs from the HD4074418 (HD4074408) in chip design and manufacturing process. Therefore, care must be taken when using the HD4074418 or HD4074408 version in place of the HD404418 since their characteristic values are not exactly the same though their guaranteed values are identical.

Pin Arrangement



Block Diagram



Pin Description

Pin No.				
DC-64S DP-64S	FP-64	FP-64A	Symbol	I/O
1	59	57	D_{11}	I/O
2	60	58	D_{12}	I/O
3	61	59	D_{13}	I/O
4	62	60	D_{14}	I/O
5	63	61	D_{15}	I/O
6	64	62	$R0_0$	I/O
7	1	63	$R0_1$	I/O
8	2	64	$R0_2$	I/O
9	3	1	$R0_3$	I/O
10	4	2	$R1_0$	I/O
11	5	3	$R1_1$	I/O
12	6	4	$R1_2$	I/O
13	7	5	$R1_3$	I/O
14	8	6	$R2_0$	I/O
15	9	7	$R2_1$	I/O
16	10	8	$R2_2$	I/O
17	11	9	$R2_3$	I/O
18	12	10	RA_0	I/O
19	13	11	RA_1	I/O
20	14	12	$R3_0/\overline{INT_0}$	I/O
21	15	13	$R3_1/\overline{INT_1}$	I/O
22	16	14	$R3_2/INT_2$	I/O
23	17	15	$R3_3/INT_3$	I/O
24	18	16	$R5_0/\overline{INT_4}/\overline{TI_1}$	I/O
25	19	17	$R5_1/INT_5/TI_2$	I/O
26	20	18	$R5_2/TO_1$	I/O
27	21	19	$R5_3/TO_2$	I/O
28	22	20	$R6_0/TO_3$	I/O
29	23	21	$R6_1/SO_2$	I/O
30	24	22	$R6_2/SI_2$	I/O
31	25	23	$R6_3/\overline{SCK_2}$	I/O
32	26	24	V_{CC}	

Pin No.				
DC-64S DP-64S	FP-64	FP-64A	Symbol	I/O
33	27	25	$R4_0/\overline{SCK_1}$	I/O
34	28	26	$R4_1/SI_1$	I/O
35	29	27	$R4_2/SO_1$	I/O
36	30	28	$R4_3/V_{ref}$	I/O
37	31	29	$R7_0$	I
38	32	30	$R7_1$	I
39	33	31	$R7_2$	I
40	34	32	$R7_3$	I
41	35	33	$R8_0$	I/O
42	36	34	$R8_1$	I/O
43	37	35	$R8_2$	I/O
44	38	36	$R8_3$	I/O
45	39	37	$R9_0$	I/O
46	40	38	$R9_1$	I/O
47	41	39	$R9_2$	I/O
48	42	40	$R9_3$	I/O
49	43	41	RESET	I
50	44	42	$\overline{TEST}$	I
51	45	43	OSC_1	I
52	46	44	OSC_2	O
53	47	45	GND	
54	48	46	D_0	I/O
55	49	47	D_1	I/O
56	50	48	D_2	I/O
57	51	49	D_3	I/O
58	52	50	D_4	I/O
59	53	51	D_5	I/O
60	54	52	D_6	I/O
61	55	53	D_7	I/O
62	56	54	D_8	I/O
63	57	55	D_9	I/O
64	58	56	D_{10}	I/O

Notes: I/O: Input/output pins
I: Input pins
O: Output pins

Pin Function

Power Supply

V_{CC}: Apply 5 V ±10% to this pin.

GND: Connect to ground.

$\overline{TEST}$: For test purposes only. Connect it to V_{CC}.

RESET: Resets the MCU. For details, see the Reset section.

Oscillators

OSC_1, OSC_2: Internal oscillator circuit pins. These can be connected to a crystal resonator, ceramic resonator, or external oscillator circuit.

Ports

D_0 to D_{15} (D Port): 1-bit I/O port. D_0 to D_{15} are all high current pins. For details, see the Input/Output section.

R0 to RA (R Ports): 4-bit I/O ports. Only RA is a 2-bit port, R0 to R2 and RA are common I/O ports. R3 to R6 are I/O ports multiplexed with other functions. R3 is multiplexed with the input of the external interrupts ($\overline{INT_0}$ to INT_3), $R4_0$ to $R4_2$ are multiplexed with the I/O pins of serial interface 1, $R4_3$ is multiplexed with V_{ref} of the analog input pins, $R5_0$ and $R5_1$ are multiplexed with $\overline{INT_4}$ and INT_5 or with timer 1 and timer 2, $R5_2$ and $R5_4$ are multiplexed with the output of timers 1 and 2, $R6_1$ to $R6_3$ are multiplexed with the I/O pins of serial interface 2, and $R6_0$ is multiplexed with the output of timer 3.

R7 is an input port available as an analog input port. R8 and R9 are I/O ports used as standard ports for the HD404418 and HD4074418, and as intermediate voltage ports for the HD4074408.

Interrupts

$\overline{INT_0}$, $\overline{INT_1}$, INT_2, INT_3, $\overline{INT_4}$, INT_5: External interrupt pins. $\overline{INT_0}$, $\overline{INT_1}$, INT_2, INT_3, $\overline{INT_4}$, and INT_5 are multiplexed with $R3_0$, $R3_1$, $R3_2$, $R3_3$, $\overline{R5_0}/TI_1$, and $R5_1/TI_2$, respectively. Refer to the Interrupt section for details.

Serial Interface

$\overline{SCK_1}$, $\overline{SCK_2}$, SI_1, SI_2, SO_1, SO_2: The transmit clock I/O pin ($\overline{SCK_1}$, $\overline{SCK_2}$), serial data input pins (SI_1, SI_2), and serial data output pin (SO_1, SO_2) are multiplexed with $R4_0$, $R6_3$, $R4_1$, $R6_2$, $R4_2$, and $R6_1$, respectively. Refer to the Serial Interface section for details.

Timers

$\overline{TI_1}$, TI_2, TO_1, TO_2, TO_3: $\overline{TI_1}$ and TI_2 are the external input pins for timers 1 and 2, and TO_1 to TO_3 are the output pins of timers 1 to 3. $\overline{TI_1}$, TI_2, TO_1, TO_2, and TO_3 are multiplexed with $R5_0/\overline{INT_4}$, $R5_1/INT_5$, $R5_2$, $R5_3$, and $R6_0$, respectively.

Analog Reference Inputs

V_{ref}, R7: The reference voltage input pin V_{ref} inputs the threshold voltage of the analog input pins and is multiplexed with $R4_3$. The analog input pins are multiplexed with port R7.

Memory Map

ROM Memory Map

The MCU contains a 8,192-word × 10-bit ROM. It is described in the following paragraphs with the ROM memory map in figure 1.

Vector Address Area ($0000 to $000F): Locations $0000 through $000F are reserved for JMPL instructions to branch to the starting address of the initialization program and of the interrupt programs. After reset or an interrupt, the program is executed from the vector address.

Zero-Page Subroutine Area ($0000 to $003F): Locations $0000 through $003F are reserved for subroutines. The CAL instruction executes the branching to subroutines.

Pattern Area ($0000 to $0FFF): Locations $0000 through $0FFF are reserved for ROM data. The P instruction can reference the ROM data as a pattern.

Program Area ($0000 to $1FFF): Locations from $0000 to $1FFF can be used for program code.

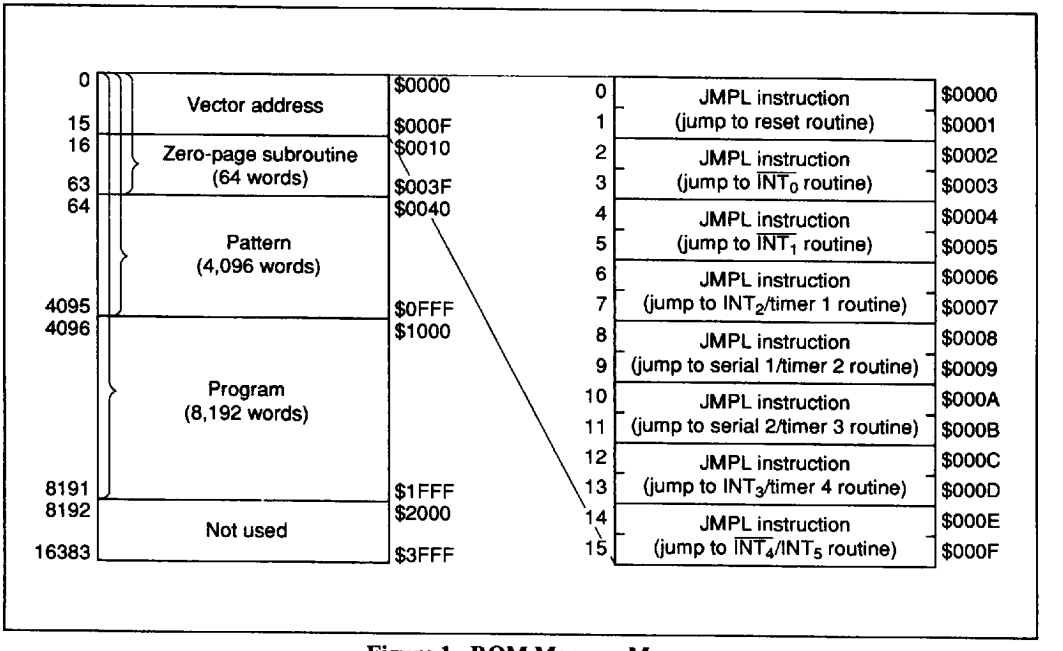


Figure 1 ROM Memory Map

RAM Memory Map

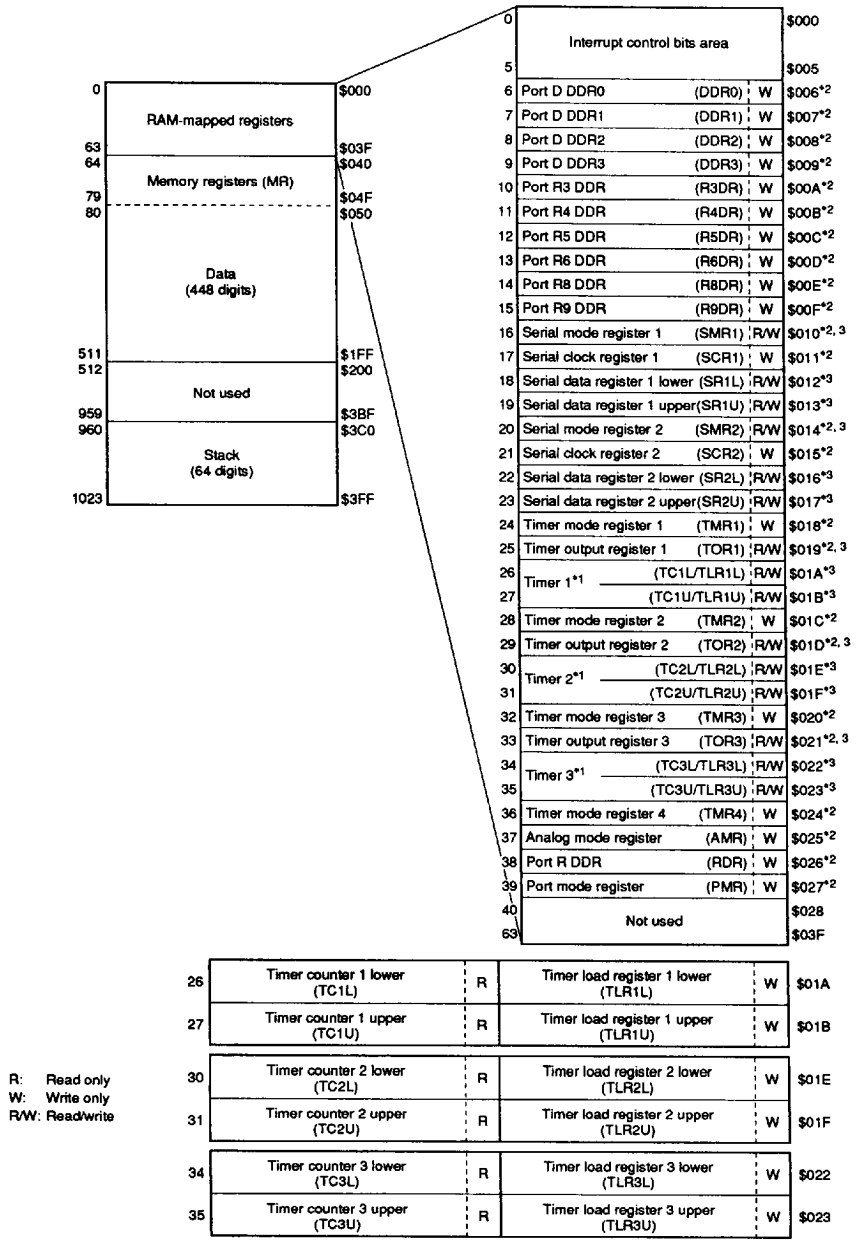
The MCU also contains a 512-digit × 4-bit RAM as the data and stack area. In addition to these areas, interrupt control bits and special function registers are mapped on the RAM memory space. The RAM memory map (figure 2) is described in the following paragraphs.

Interrupt Control Bits Area ($000 to $005): The interrupt control bits area (figure 3) is used for interrupt control. It is accessible only by RAM bit manipulation instructions. However, the interrupt request flag cannot be set by software. The RSP bit is used only to reset the stack pointer.

Special Function Registers Area ($006 to $027): The special function registers are the mode or data registers for the external interrupt, the serial interface, and the timer/counters. These registers are classified into three types: write-only, read-only, and read/write as shown in figure 2. The SEM/SEMD instruction or REM/REMD instruction is available for the mode register (SMR) and clock register (SCR) of the serial interface, the mode register (TMR) and output register (TOR) of the timer, the analog mode register (AMR) and the port mode register (PMR), and each data direction register (DDR). The TM/TMD instruction is available for the read register. RAM bit manipulation instructions are unavailable to other registers.

Data Area ($040 to $1FF): The 16 digits of $040 through $04F are called memory registers (MR) and are accessible by the LAMR and XMRA instructions (figure 4).

Stack Area ($3C0 to $3FF): Locations $3C0 through $3FF are reserved for LIFO stacks to save the contents of the program counter (PC), status flag (ST), and carry flag (CA) when subroutine calls (CAL or CALL instruction) and interrupts are processed. This area can be used as a 16-level nesting stack in which one level requires 4 digits. Figure 4 shows the save condition. The program counter is restored by the RTN and RTNI instructions. The status and carry flags are restored only by the RTNI instruction. This area, when not used for a stack, is available as a data area.



Notes: 1. Two registers are mapped on the same address.
2. SEM/SEMD instruction or REM/REMD instruction is available to these registers.
3. TM/TMD instruction is available to these registers.

Figure 2 RAM Memory Map

	Bit 3	Bit 2	Bit 1	Bit 0	
0	IM0 (IM of $\overline{INT_0}$)	IF0 (IF of $\overline{INT_0}$)	RSP (Reset SP bit)	IE (Interrupt enable flag)	$000
1	IM2 (IM of INT_2)	IF2 (IF of INT_2)	IM1 (IM of $\overline{INT_1}$)	IF1 (IF of $\overline{INT_1}$)	$001
2	IMS1 (IM of serial 1)	IFS1 (IF of serial 1)	IMT1 (IM of timer 1)	IFT1 (IF of timer 1)	$002
3	IMS2 (IM of serial 2)	IFS2 (IF of serial 2)	IMT2 (IM of timer 2)	IFT2 (IF of timer 2)	$003
4	IM3 (IM of INT_3)	IF3 (IF of INT_3)	IMT3 (IM of timer 3)	IFT3 (IF of timer 3)	$004
5	IM4 (IM of $\overline{INT_4}$/INT_5)	IF4 (IF of $\overline{INT_4}$/INT_5)	IMT4 (IM of timer 4)	IFT4 (IF of timer 4)	$005

IF: Interrupt request flag
IM: Interrupt mask
IE: Interrupt enable flag
SP: Stack pointer

Note: Each bit of the interrupt control bits area is set by the SEM/SEMD instruction, reset by the REM/REMD instruction, and tested by the TM/TMD instruction. It is not affected by other instructions. Furthermore, the interrupt request flag is not affected by the SEM/SEMD instruction. The value of the status flag becomes invalid when the RSP bit is tested.

Figure 3 Configuration of Interrupt Control Bits Area

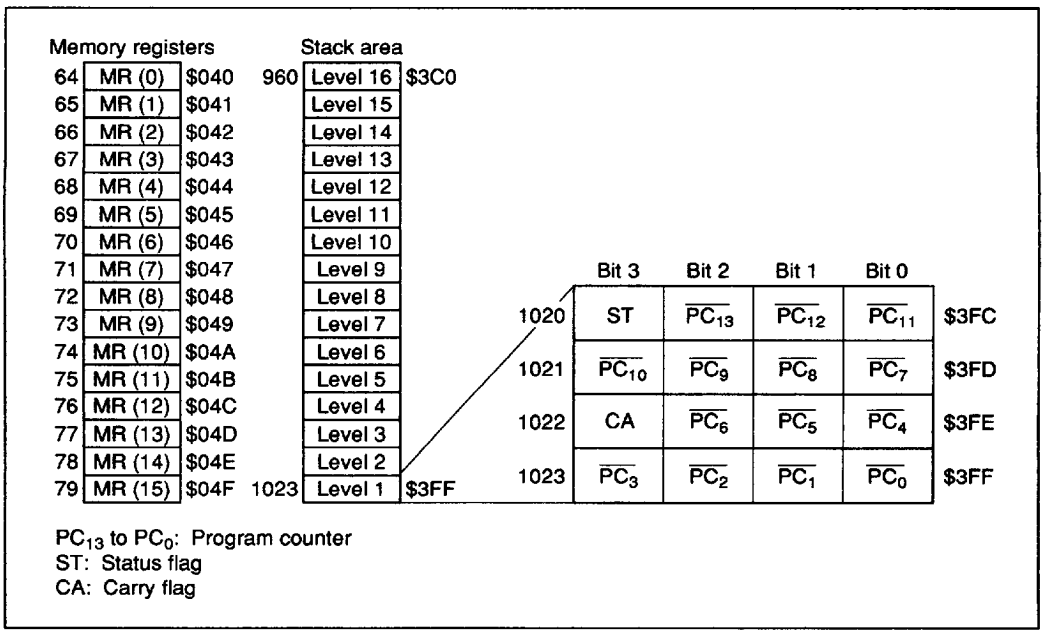


Figure 4 Configuration of Memory Registers, Stack Area, and Stack Position

Functional Description

Registers and Flags

The MCU has nine registers and two flags for CPU operations (figure 5).

Accumulator (A), B Register (B): The 4-bit accumulator and B register hold the results from the arithmetic logic unit (ALU), and transfer data to/from memory, I/O, and other registers.

W Register (W), X Register (X), Y Register (Y): The 2-bit W register and the 4-bit X and Y registers address RAM indirectly. The Y register is also used for D-port addressing.

SPX Register (SPX), SPY Register (SPY): The 4-bit SPX and SPY registers assist registers X and Y, respectively.

Carry Flag (CA): The carry flag (CA) stores the overflow from the ALU generated by an arithmetic operation. It is also affected by the SEC, REC, ROTL, and ROTR instructions.

During an interrupt, the carry flag is pushed onto the stack. It is restored by the RTNI instruction, but not by the RTN instruction.

Status Flag (ST): The status flag (ST) holds the ALU overflow, ALU non-zero, and the results of a bit test instruction for the arithmetic or compare instruction. It is a branch condition of the BR, BRL, CAL, or CALL instruction. The value for the status flag remains unchanged until the next arithmetic, compare, or bit test instruction is executed. The status flag becomes a 1 after the BR, BRL, CAL, or CALL instruction is executed or skipped. During an interrupt, the status flag is pushed onto the stack. It is restored back from the stack by the RTNI instruction, but not by the RTN instruction.

Program Counter (PC): The program counter is a 14-bit binary counter to hold the ROM address.

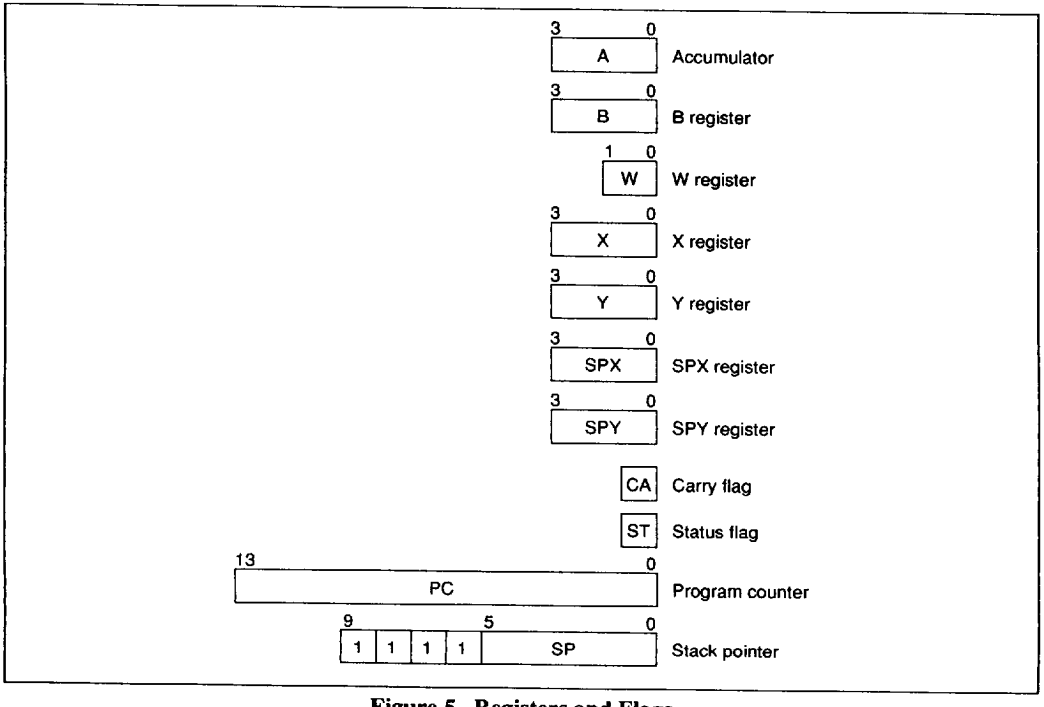


Figure 5 Registers and Flags

Stack Pointer (SP): The stack pointer (SP) points to the address of the next stack area for up to 16 levels.

The stack pointer is initialized to RAM address $3FF. It is decremented by 4 when data is pushed onto the stack, and incremented by 4 when data is restored from it. The stack can only be used up to 16 levels deep because the top four bits of the stack pointer are fixed at 1111.

The stack pointer is initialized to $3FF either by MCU reset or by the RSP bit reset through the REM/REMD instruction.

Reset

Setting the RESET pin high resets the MCU. At power-on, or when cancelling the stop mode, the reset period must satisfy t_{RC} to stabilize the oscillator. In all other cases, at least three instruction cycles are required to reset the MCU.

Tables 1 and 2 show the names and the statuses of each component initialized by MCU reset.

Table 1 Initial Values after MCU Reset

Items			Initial Value	Contents
Program counter (PC)			$0000	Execute the program from the top of ROM address
Status flag (ST)			1	Enable branching with conditional branching instructions
Stack pointer (SP)			$3FF	Stack level is 0
I/O pins, output registers	Standard pins	CMOS	1	Enable input
		NMOS open drain	1	Enable input
	Intermediate-voltage pins	NMOS open drain	1	Enable input
Interrupt flags and mask	Interrupt enable flag (IE)		0	Inhibit all interrupts
	Interrupt request flag (IF)		0	No interrupt request
	Interrupt mask (IM)		1	Mask interrupt request
Mode registers	Serial mode registers (SMR1 to 2)		0000	Refer to Serial Mode Register section
	Serial clock registers (SCR1 to 2)		000	Refer to Serial Clock Register section
	Timer mode registers 1 to 3 (TMR1 to 3)		0000	Refer to Timer Mode Registers 1 to 3 section
	Timer output registers 1 to 3 (TOR1 to 3)		0000	Refer to Timer Output Register section
	Timer mode register 4 (TMR4)		0000	Refer to Timer Mode Register 4 section
	Analog mode register (AMR)		0000	Refer to Analog Mode Register section
	Port mode register (PMR)		0000	Refer to Port Mode Register section
	Data direction registers (DDR0 to 3, R3DR to R9DR, RDR)		0000	Refer to Data Direction Register section
Timer/ counters, serial interface	Prescaler		$000	
	Timer counter 4 (TC4)		$00	
	Timer counters 1 to 3 (TC1 to 3)		$00	
	Timer load registers 1 to 3 (TLR1 to 3)		$00	
	Octal counter		000	

Table 2 Initial Values after Mode Cancellation by MCU Reset

Item	Abbr	After MCU Reset to Recover from Stop Mode	After MCU Reset to Recover from Other Modes
Carry flag	(CA)	The contents of the items just before MCU reset are not assured. It is necessary to initialize them by software again.	The contents of the items just before MCU reset are not assured. It is necessary to initialize them by software again.
Accumulator	(A)		
B register	(B)		
W register	(W)		
X/SPX register	(X/SPX)		
Y/SPX register	(Y/SPY)		
Serial data registers 1 to 2	(SR1 to SR2)		
RAM		The contents of RAM before MCU reset (just STOP instruction) are retained.	The contents of RAM just before MCU reset are not assured. It is necessary to initialize them by software again.

Interrupts

The MCU can be interrupted by 12 different sources: external signals ($\overline{INT_0}$, $\overline{INT_1}$, INT_2, INT_3, $\overline{INT_4}$, and INT_5), timers (timers 1, 2, 3, and 4), and serial interfaces (serial 1 and serial 2).

Each interrupt source provides an interrupt request flag and interrupt mask ($\overline{INT_4}$ and INT_5 provide a common interrupt request flag and interrupt mask) to hold or control interrupt requests. An interrupt enable flag is available for controlling the total interrupt operation. Note that INT_2 and timer 1, serial 1 and timer 2, serial 2 and timer 3, INT_3 and timer 4 use common vector addresses, respectively. Therefore, the interrupt requests must be checked by software initially in the interrupt processing routine.

Interrupt Control Bits Area and Interrupt Servicing: The interrupt control bits area is mapped on \$000 through \$005 of the RAM space. They are accessible by RAM bit manipulation instructions. The interrupt request flag (IF) cannot be set by software. The interrupt enable flag (IE) and IF are cleared to 0, and the interrupt mask (IM) is set to 1 after MCU reset.

Figure 6 is a block diagram of the interrupt control circuit. Table 3 shows the interrupt priority and vector addresses, and table 4 shows the interrupt conditions corresponding to each interrupt source.

An interrupt request is generated when the IF is set to 1 and IM is 0. If the IE is 1 at this time, the interrupt will be activated and vector addresses will be generated from the priority PLA corresponding to the interrupt sources.

Figure 7 shows the interrupt sequence, and figure 8 shows the interrupt processing flowchart. If an interrupt is requested, the instruction being executed finishes in the first cycle. The IE is reset in the second cycle. In the second and third cycles, the carry flag, status flag, and program counter are pushed onto the stack. In the third cycle, the execution of an instruction starts after jumping to the vector address.

At each vector address, program the JMPL instruction to branch to the starting address of the interrupt program. The IF, which caused the interrupt, must be reset by software in the interrupt program.

Table 3 Vector Addresses and Interrupt Priority

Reset/Interrupt	Priority	Vector Addresses
RESET	—	\$0000
$\overline{INT_0}$	1	\$0002
$\overline{INT_1}$	2	\$0004
INT_2/timer 1	3	\$0006
Serial 1/timer 2	4	\$0008
Serial 2/timer 3	5	\$000A
INT_3/timer 4	6	\$000C
$\overline{INT_4}$/INT_5	7	\$000E

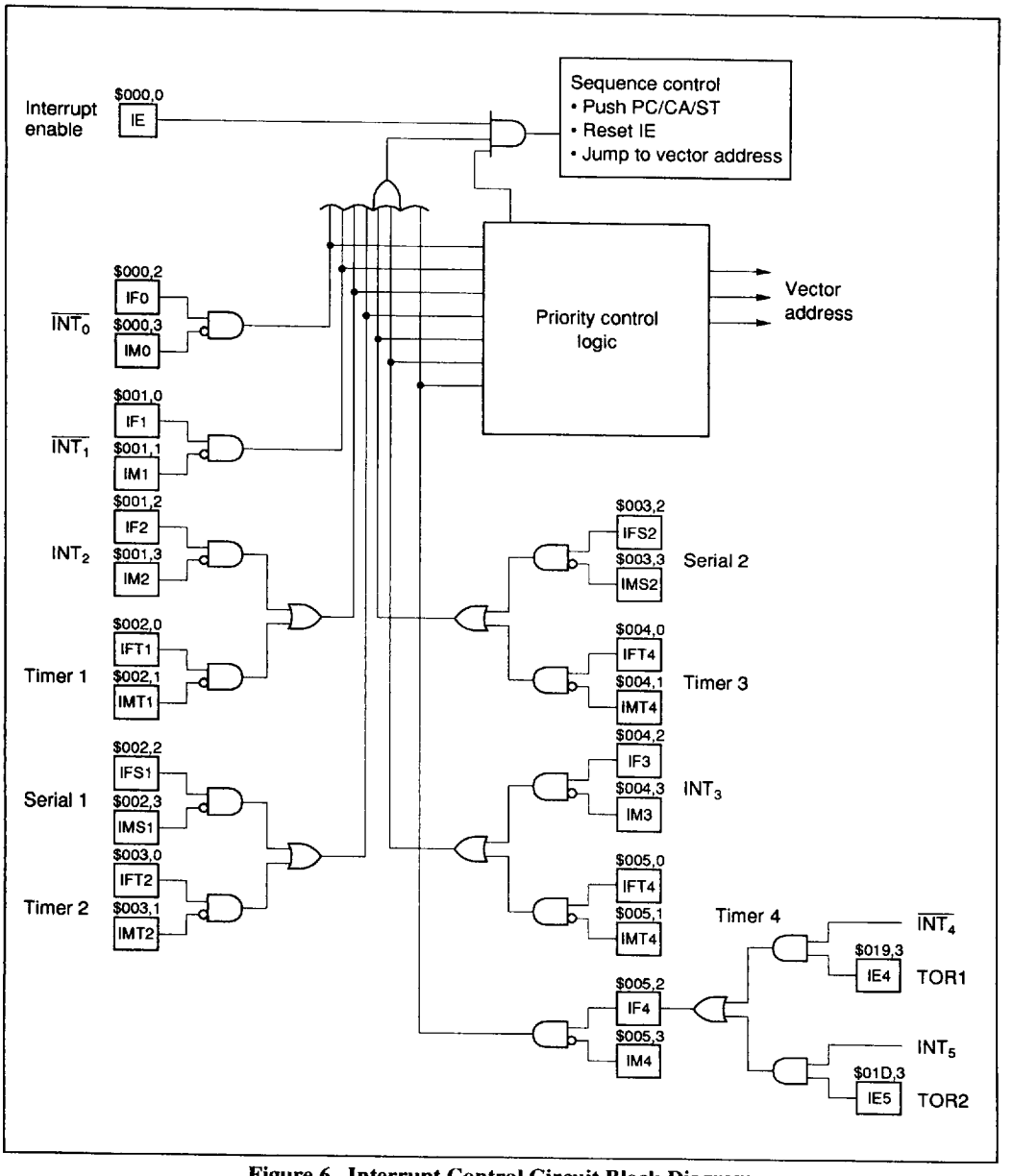


Figure 6 Interrupt Control Circuit Block Diagram

Table 4 Interrupt Conditions

Interrupt Control Bit	$\overline{INT_0}$	$\overline{INT_1}$	INT_2/ Timer 1	Serial 1/ Timer 2	Serial 2/ Timer 3	INT_3/ Timer 4	$\overline{INT_4}$/ INT_5
IE	1	1	1	1	1	1	1
IF0 · $\overline{IM0}$	1	0	0	0	0	0	0
IF1 · $\overline{IM1}$	*	1	0	0	0	0	0
IF2 · $\overline{IM2}$	*	*	1/*	0	0	0	0
IFT1 · $\overline{IMT1}$	*	*	*/1	0	0	0	0
IFS1 · $\overline{IMS1}$	*	*	*	1/*	0	0	0
IFT2 · $\overline{IMT2}$	*	*	*	*/1	0	0	0
IFS2 · $\overline{IMS2}$	*	*	*	*	1/*	0	0
IFT3 · $\overline{IMT3}$	*	*	*	*	*/1	0	0
IF3 · $\overline{IM3}$	*	*	*	*	*	1/*	0
IFT4 · $\overline{IMT4}$	*	*	*	*	*	*/1	0
IF4 · $\overline{IM4}$	*	*	*	*	*	*	1

Note: * Don't care

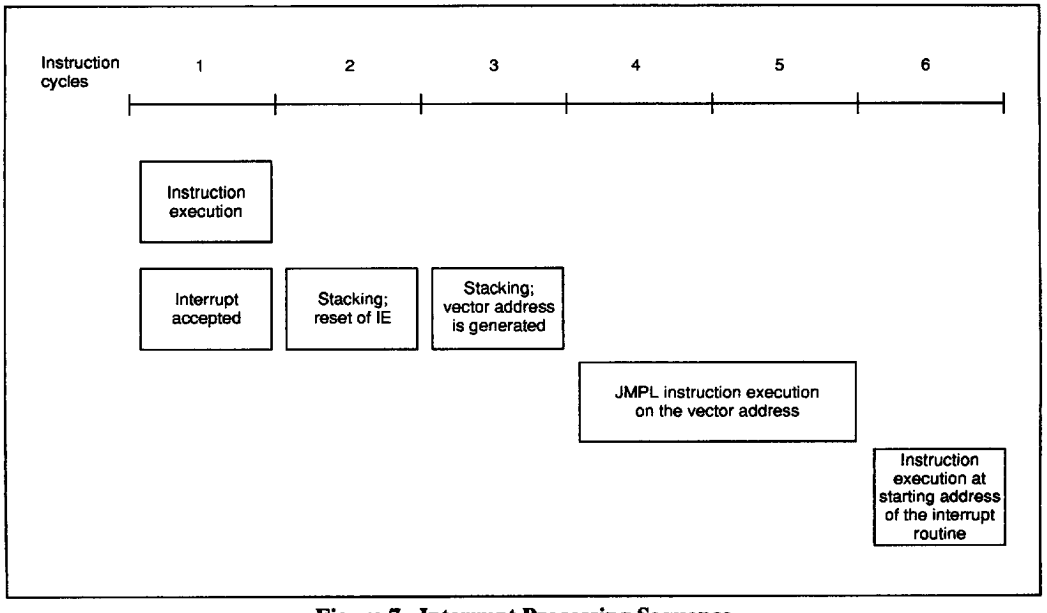


Figure 7 Interrupt Processing Sequence

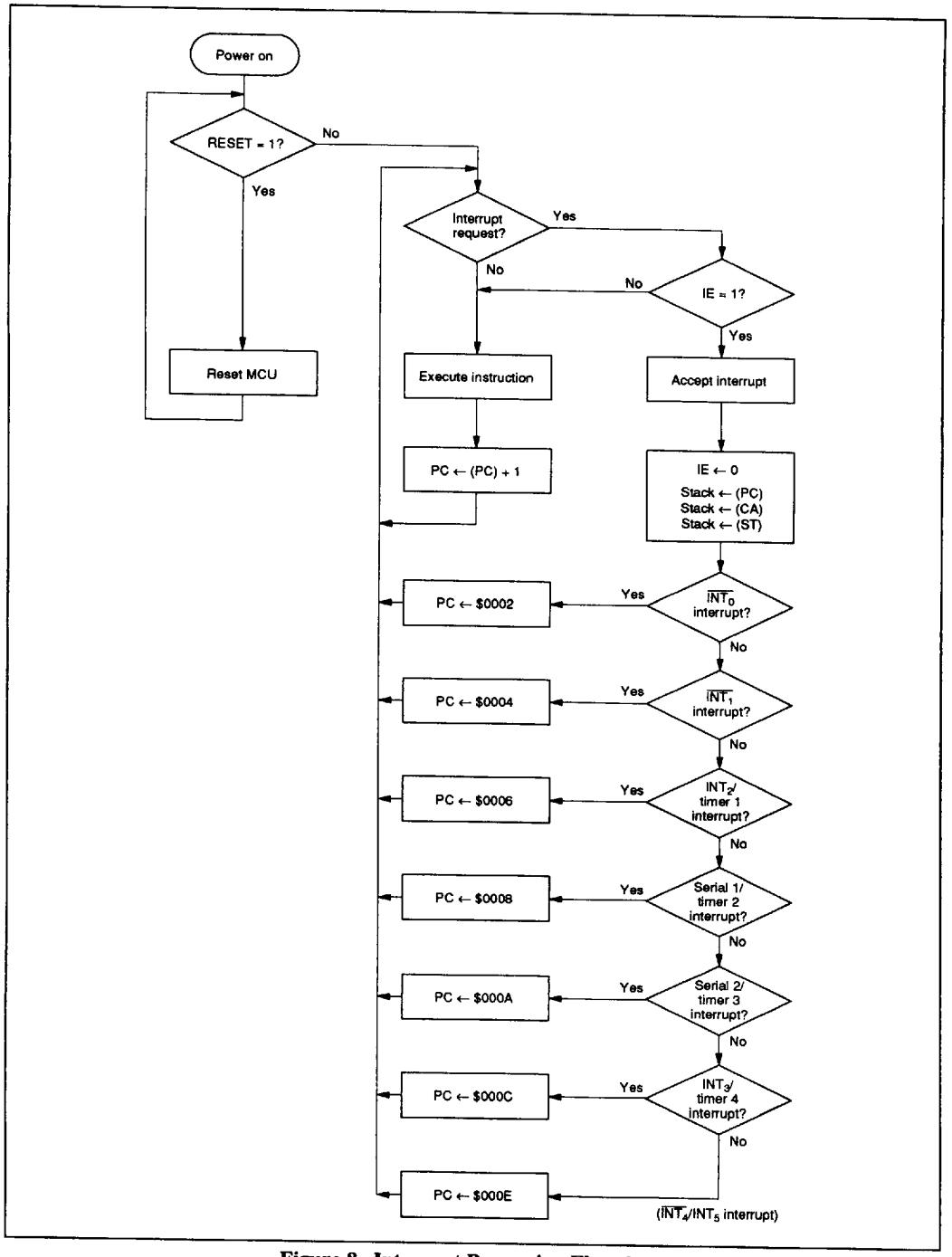


Figure 8 Interrupt Processing Flowchart

Interrupt Enable Flag (IE: $000, Bit 0): The interrupt enable flag enables/disables interrupt requests as shown in table 5. It is reset by an interrupt and set by the RTNI instruction.

External Interrupts ($\overline{INT_0}$, $\overline{INT_1}$, INT_2, INT_3, $\overline{INT_4}$, INT_5): External interrupt request flags are set at the falling edge of $\overline{INT_0}$, $\overline{INT_1}$, and $\overline{INT_4}$ inputs or the rising edge of INT_2, INT_3, and INT_5 inputs.

When using $\overline{INT_0}$ to INT_3, select the external interrupt input by setting the appropriate bit of the port mode register (PMR: $027). When the port mode register has been reset, external interrupt input signals are masked to ignore external interrupt requests.

When using $\overline{INT_4}$ and INT_5, set the external interrupt enable bit of the timer output register (TOR1: $019, TOR2: $01D). Then the data direction registers of the corresponding pins are automatically reset to receive the external interrupt input signals. When the interrupt enable bit has been reset, the external interrupt input signal and external interrupt request will be ignored.

Note that $\overline{INT_4}$ and INT_5 use a common external interrupt request flag. Therefore, when using these pins, use one external interrupt input pin, or check the interrupt by software before starting the processing.

External Interrupt Request Flags (IF0: $000, Bit 2; IF1: $001, Bit 0; IF2: $001, Bit 2; IF3: $004, Bit 2; IF4: $005, Bit 2): IF0 and IF1 are set at the falling edge of $\overline{INT_0}$ and $\overline{INT_1}$ inputs, respectively (table 6).

IF2 and IF3 are set at the rising edge of INT_2 and INT_3, respectively.

IF4 is set at the falling edge of $\overline{INT_4}$ or at the rising edge of INT_5.

External Interrupt Masks (IM0: $000, Bit 3; IM1: $001, Bit 1; IM2: $001, Bit 3; IM3: $004, Bit 3; IM4: $005, Bit 3): These bits mask the interrupt request generated by the external interrupt request flags (table 7).

Timer Interrupt Request Flags (IFT1: $002, Bit 0; IFT2: $003, Bit 0; IFT3: $004, Bit 0; IFT4: $005, Bit 0): The timer interrupt request flags are set by the overflow output of timers 1 to 4. When timers 1 to 3 select the PWM operation, the overflow output does not set the interrupt request flag. When timer 4 is selected as the watchdog timer, the overflow output resets the MCU. Therefore, the timer interrupt request flag is not set by the overflow output (table 8).

Timer Interrupt Masks (IMT1: $002, Bit 1; IMT2: $003, Bit 1; IMT3: $004, Bit 1; IMT4: $005, Bit 1): The timer interrupt masks mask the occurrence of an interrupt request generated by timers 1 to 4 interrupt request flags (table 9).

Serial Interrupt Request Flags (IFS1: $002, Bit 2; IFS2: $003, Bit 2): The serial interrupt request flags are set when the octal counter counts eight transmit clocks, or when the data transfer is suspended, then the octal counter is reset (table 10).

Serial Interrupt Masks (IMS1: $002, Bit 3; IMS2: $003, Bit 3): The serial interrupt masks mask the interrupt request which the serial interrupt request flag generates (table 11).

Table 5 Interrupt Enable Flag

IE	Interrupt Enable/Disable
0	Disable
1	Enable

Table 6 External Interrupt Request Flags

IF0, IF1, IF2, IF3, IF4	Interrupt Request
0	No
1	Yes

Table 7 External Interrupt Masks

IM0, IM1, IM2, IM3, IM4	Interrupt Request
0	Enabled
1	Disabled (masked)

Table 8 Timer Interrupt Request Flags

IFT1, IFT2, IFT3, IFT4	Interrupt Request
0	No
1	Yes

Table 9 Timer Interrupt Masks

IMT1, IMT2, IMT3, IMT4	Interrupt Request
0	Enabled
1	Disabled (masked)

Table 10 Serial Interrupt Request Flags

IFS1, IFS2	Interrupt Request
0	No
1	Yes

Table 11 Serial Interrupt Masks

IMS1, IMS2	Interrupt Request
0	Enabled
1	Disabled (masked)

Operation Modes

The MCU has two low-power dissipation modes: standby mode and stop mode (table 12). Figure 9 is a mode transition diagram of these modes.

Standby Mode: Executing the SBY instruction places the MCU into standby mode. In standby mode, the oscillator circuit is active, and interrupts, timer/counters, and serial interfaces remain working. On the other hand, the CPU stops since the clock related to the instruction execution stops. Registers, RAM, and I/O pins retain the states they were in just before the MCU went into standby mode.

Table 12 Low-Power Dissipation Mode Functions

		Condition							
Low-Power Dissipation Mode	Instruction	Oscillator Circuit	Instruction Execution	Registers, Flags	Interrupt Function	RAM	Input/ Output Pins	Timer/ Counters, Serial Interfaces	Cancellation Method
Standby mode	SBY instruction	Active	Stop	Retained	Active	Retained	Retained*2	Active	RESET input, interrupt request
Stop mode	STOP instruction	Stop	Stop	Reset*1	Stop	Retained	High impedance	Stop	RESET input

Notes: 1. The MCU recovers from the stop mode by RESET input. Refer to tables 1 and 2 for the contents of the flags and registers.
2. If an I/O circuit is active, an I/O current may flow, depending on the state of the I/O pin in standby mode. This is the additional current to the current dissipation in standby mode.

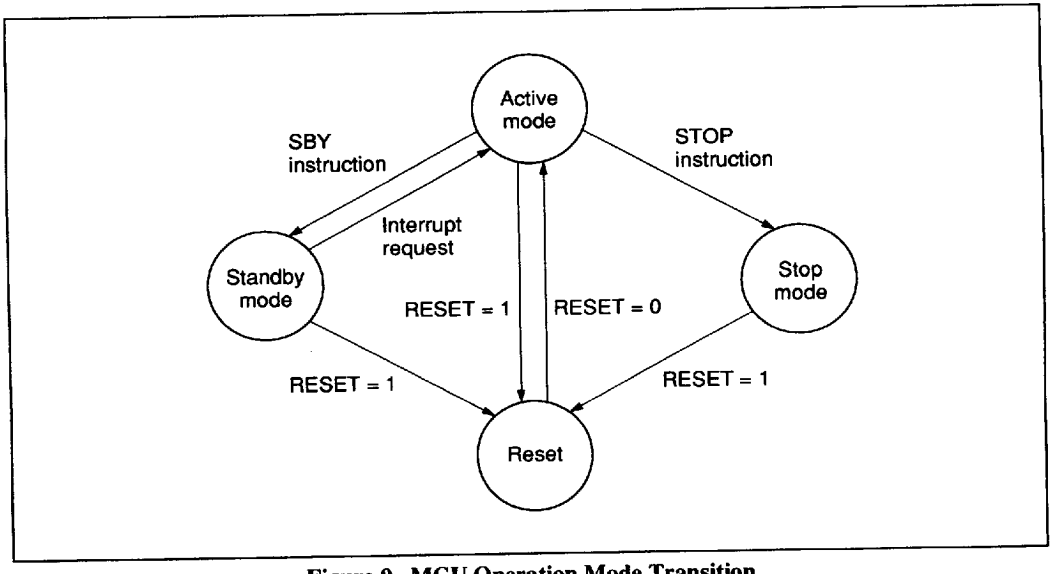


Figure 9 MCU Operation Mode Transition

The standby mode may be cancelled by inputting RESET or by asserting an interrupt request. In the former case the MCU is reset. In the latter case, the MCU becomes active and executes the next instruction following the SBY instruction. If the interrupt enable flag is 1 when an interrupt request is asserted, the interrupt is executed; if it is 0, the interrupt request is put on hold and normal instruction execution continues.

Figure 10 shows the flowchart of the standby mode.

Stop Mode: Executing the STOP instruction brings the MCU into stop mode, in which the oscillator circuit and every function of the MCU stop.

The stop mode may be cancelled by resetting the MCU. At this time, as shown in figure 11, the reset input must be applied for at least t_{RC} to stabilize oscillation. (Refer to the AC Characteristics table.) After stop mode is cancelled, the RAM retains the state it was in just before the MCU went into stop mode, but the accumulator, registers B, W, X/SPX, and Y/SPY, carry flag, and serial data register do not retain their contents.

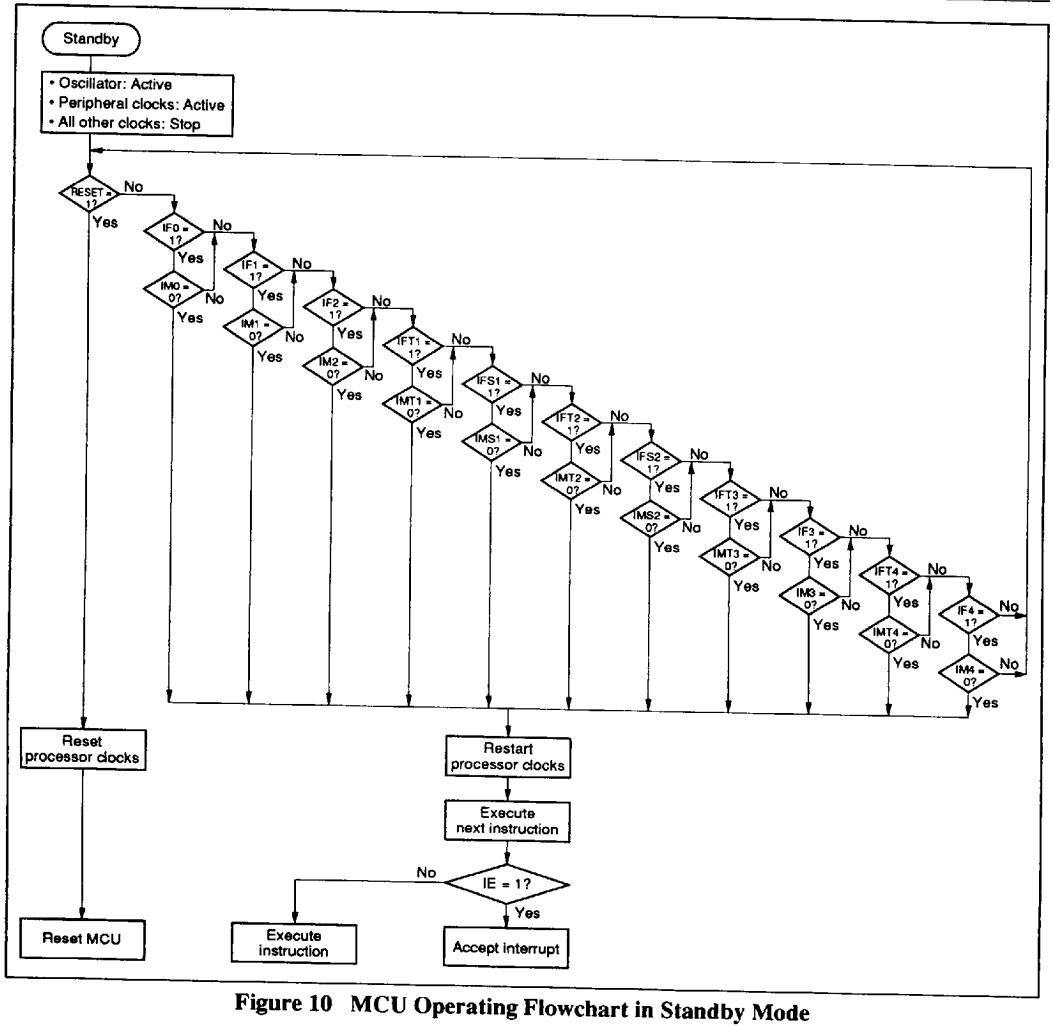


Figure 10 MCU Operating Flowchart in Standby Mode

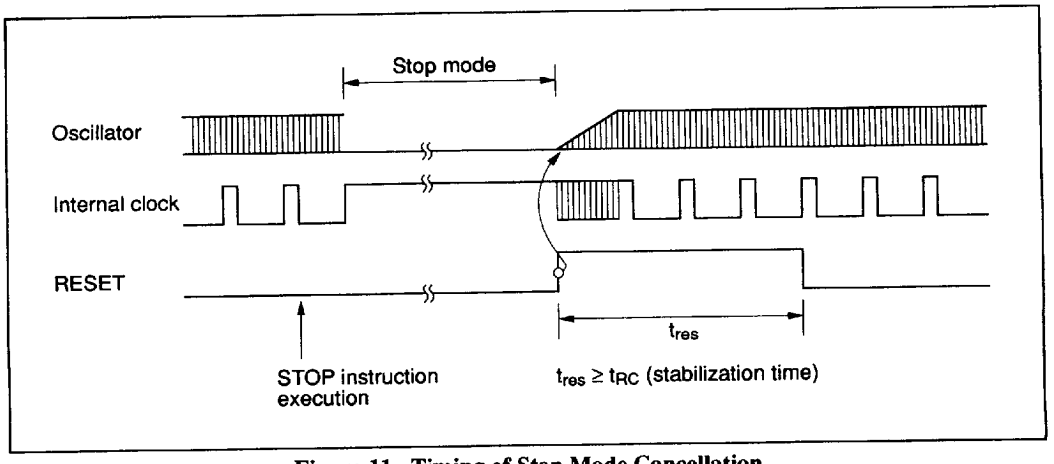


Figure 11 Timing of Stop Mode Cancellation

The low-power mode operation sequence is shown in figure 12. With the IE flag cleared and an interrupt flag set together with its interrupt mask cleared, if a STOP/SBY instruction is executed, the instruction is cancelled (regarded as an NOP) and the following instruction is executed. Before executing a STOP/SBY instruction, make sure all interrupt flags are cleared or all interrupts are masked.

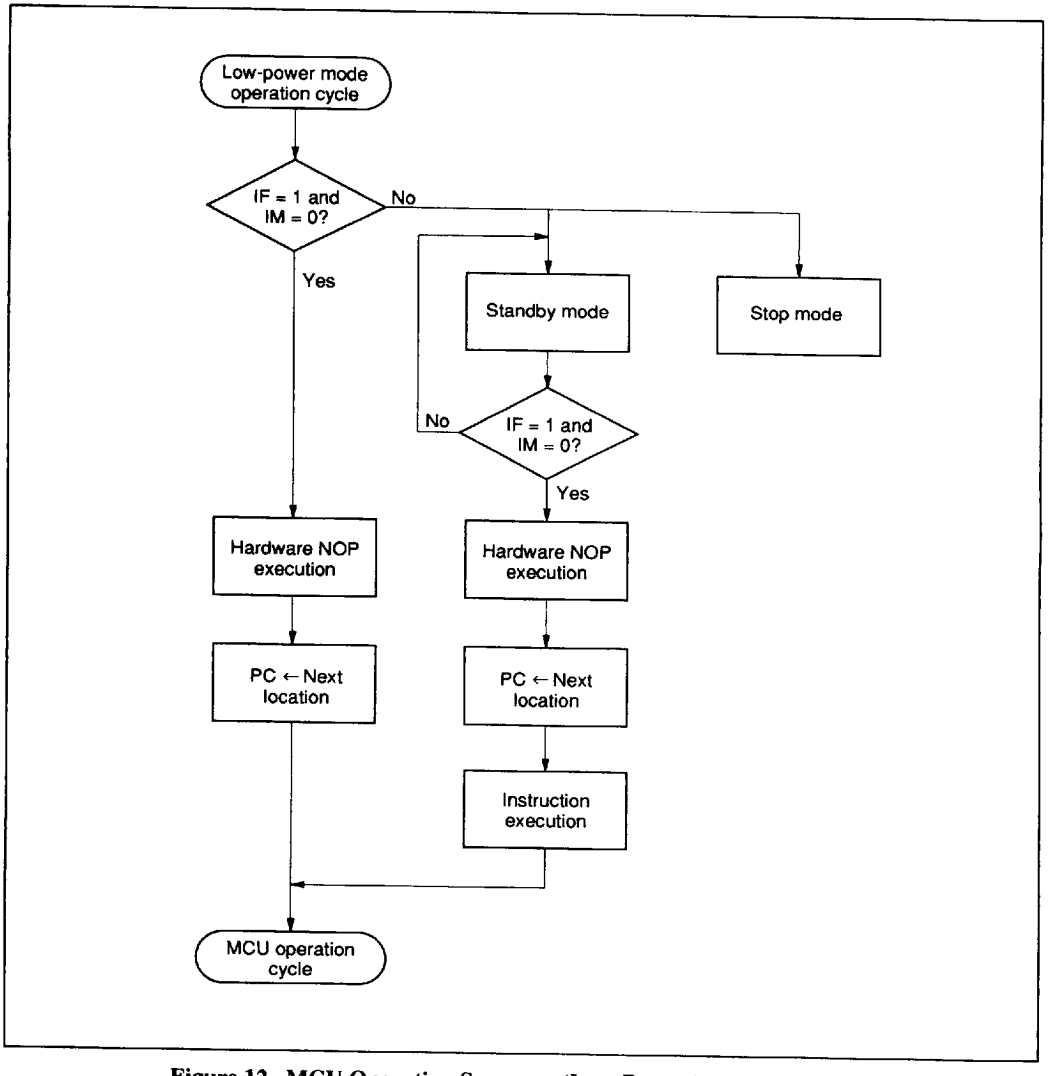
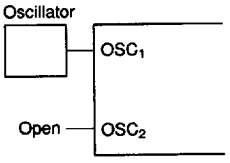
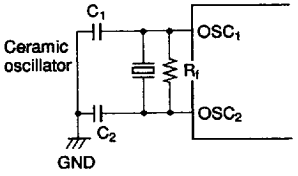
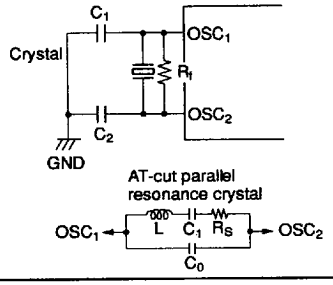


Figure 12 MCU Operating Sequence (Low-Power Mode Operation)

Internal Oscillator Circuit

Figure 13 outlines the internal oscillator circuit. A crystal oscillator or ceramic oscillator can be selected as the oscillator type. Refer to table 13 to select the type. In addition, see figure 14 for the layout of the crystal or ceramic oscillator.

Table 13 Examples of Oscillator Circuits

	Circuit Configuration	Circuit Constants
External clock operation		
Ceramic oscillator		Ceramic oscillator CSA8.00MT (Murata) R_f: 1MΩ ±20% C_1: 30 pF ±20% C_2: 30 pF ±20%
Crystal Oscillator		R_f: 1MΩ ±20% C_1: 10 pF to 22 pF ± 20% C_2: 10 pF to 22 pF ± 20% Crystal: Equivalent circuit shown at bottom left C_0: 7 pF max. R_S: 100 max. f: 1.0 MHz to 9.0 MHz

Notes: 1. Since the circuit constant changes according to the crystal or ceramic resonator and stray capacitance of the board, consult with the crystal or ceramic oscillator manufacturer to determine the circuit parameters.
2. Wiring among OSC_1, OSC_2, and elements should be as short as possible, and never cross other wiring. Refer to figure 14.

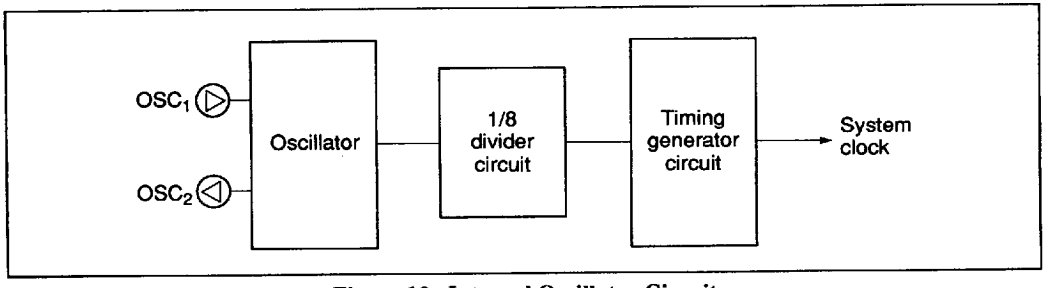


Figure 13 Internal Oscillator Circuit

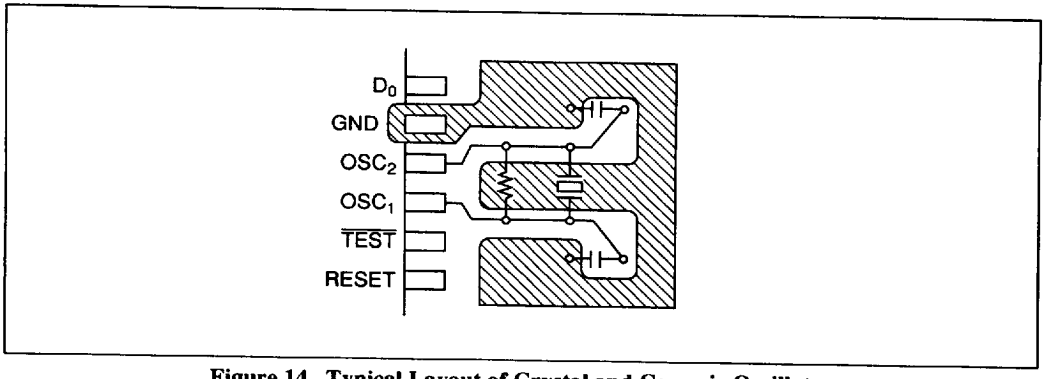


Figure 14 Typical Layout of Crystal and Ceramic Oscillator

Input/Output

The MCU provides 58 I/O pins. Each port provides a data direction register (DDR). Each bit of ports D, R3, R4, R5, R6, R8, and R9 can be individually programmed as input or as output. Ports R0, R1, R2, and RA can be individually programmed as input or as output. Port R7 is a 4-bit input-only port.

D Port: The D port consists of 16 high-current I/O pins. Each bit of port D can be individually programmed as input or output by the D-port data direction registers (DDR0 to DDR3) (0 = input, 1 = output). Port D becomes an input port at MCU reset. Port D can be set/reset by the SED/RED and SEDD/REDD instructions, and tested by the TD/TDD instruction. The data direction register can be set or reset either by the SEM/REM instruction or SEMD/REMD instruction.

R Port: Ports R0 to R9 are 4-bit I/O ports. Port RA is a 2-bit I/O port.

Ports R0, R1, and R2: 4-bit I/O ports. Each port can be individually programmed as input or output by the R-port data direction registers (RDR) (0 = input, 1 = output). Ports R0, R1, and R2 become input ports at MCU reset. These ports receive data by the LAR/LBR instruction and transmit data by the LRA/LRB instruction. A data direction register can be set/reset either by the SEM/REM instruction or SEMD/REMD instruction.

Port R3: A 4-bit I/O port. Each bit of port R3 can be individually programmed as input or output by the R3-port data direction register (R3DR) (0 = input, 1 = output). Port R3 becomes an input port at MCU reset.

Note that port R3 is multiplexed with the external interrupt input pins ($\overline{INT_0}$, $\overline{INT_1}$, INT_2, and INT_3). These pins can be used as interrupt input pins when the data direction register of the appropriate bit is automatically reset with an interrupt enable bit of the port mode register (PMR) being set.

Port R3 receives data by the LAR/LBR instruction, and transmits data by the LRA/LRB instruction. The data direction register can be set/reset either by the SEM/REM instruction or SEMD/REMD instruction.

Port R4: A 4-bit I/O port. Each bit of port R4 can be individually programmed as input or output by the R4-port data direction register (R4DR) (0 = input, 1 = output). Port R4 becomes an input port at MCU reset.

Note that pins $R4_0$, $R4_1$, and $R4_2$ of port R4 are multiplexed with $\overline{SCK_1}$, SI_1, and SO_1 of serial interface 1, respectively. $R4_3$ is multiplexed with the reference voltage input (V_{ref}) for comparator input.

Port R4 receives data by the LAR/LBR instruction, and transmits data by the LRA/LRB instruction. The data direction register can be set/reset either by the SEM/REM instruction or SEMD/REMD instruction.

Port R5: A 4-bit I/O port. Each bit of port R5 can be individually programmed as input or output by the R5-port data direction register (R5DR) (0 = input, 1 = output). Port R5 becomes an input port at MCU reset.

Note that port R5 is multiplexed with the external interrupts ($\overline{INT_4}$, INT_5), timer inputs ($\overline{TI_1}$, TI_2), and timer outputs (TO_1, TO_2). $\overline{TI_1}$ and TI_2 become clock input pins when a timer is used as an event counter. TO_1 and TO_2 become clock output pins of timers 1 and 2, respectively. These pins output clocks (with appropriate cycles) and PWM output signals through the use of the reload function.

Port R5 receives data by the LAR/LBR instruction, and transmits data by the LRA/LRB instruction. The data direction register can be set/reset by the SEM/REM instruction or SEMD/REMD instruction.

Port R6: A 4-bit I/O port. Each bit of port R6 can be individually programmed as input or output by programming the R6-port data direction register (R6DR) (0 = input, 1 = output). Port R6 becomes an input port at MCU reset.

$R6_0$ is multiplexed with TO_3. $R6_1$, $R6_2$, and $R6_3$ are multiplexed with $\overline{SCK_2}$, SI_2, and SO_2 of serial interface 2, respectively.

Port R6 receives data by the LAR/LBR instruction, and transmits data by the LRA/LRB instruction.

The data direction register can be set/reset by the SEM/REM instruction or SEMD/REMD instruction.

Port R7 (analog input port): A 4-bit port which provides the digital input and analog input operation modes. These modes are available to each bit by programming the analog mode register (AMR).

In the digital input mode, port R7 is available as an input-only port with characteristics equivalent to other I/O ports. In the analog input mode, port R7 reads the comparison result between the reference voltage which is input by $R4_3/V_{ref}$ and the input voltage of the port, as input data.

In the analog input mode, direct current constantly flows in the analog comparator to assure its characteristics. Thus the MCU consumes power in the analog input mode. The power consumption cannot be reduced even with the reduction of the operation cycle. Therefore, you should not use port R7 in the analog input mode, except when an analog comparison is required. In this case, 2 instruction cycles are required after R7 goes into the analog input mode until the analog comparator is stabilized to read the precise data. Therefore, read the data at least 2 instruction cycles after placing R7 into the analog input mode. The analog comparator holds its state in the standby mode, but stops operating in the stop mode.

Ports R8 and R9: 4-bit I/O ports. These ports are standard I/O ports for the HD404418 and HD4074418, and intermediate-voltage ports, which can apply 12.8-V maximum voltage for the HD4074408.

I/O direction is specified by the R8-port data direction register (R8DR) and R9-port data direction register (R9DR) on a bit basis (0 = input, 1 = output). Ports 8 and 9 become inputs at reset since the registers are cleared by reset input.

Ports R8 and R9 receive data by the LAR/LBR instruction, and transmit data by the LRA/LRB instruction. The data direction register can be set/reset by the SEM/REM instruction or SEMD/REMD instruction.

Port RA: A 2-bit I/O port. Port RA can be programmed as input or output by the R-port data direction register (RDR)(0 = input, 1 = output). The RDR is cleared at MCU reset, then becomes an input port.

Port RA receives data by the LAR/LBR instruction, and transmits data by the LRA/LRB instruction. The data direction register can be set/reset by the SEM/REM instruction or SEMD/REMD instruction.

Data Direction Register ($006 to $00F, $026): The 4-bit write-only data direction registers (DDR0 to DDR3, RDR, R3DR to R9DR) (table 14) control the input/output selection of the I/O port. Each bit of ports D, R3 to R6, R8, and R9 can be individually programmed as input or output by the DDR for each bit.

Ports R0 to R2 and RA can be individually programmed as input or output by the DDR for each port.

When functioning as an input port, each port reads data from the pins. When functioning as an output port, each port reads data from the data register. Thus, the MCU reads the transmitted data precisely even when the output is changing.

Each DDR is reset to 0 at MCU reset. Then each port becomes an input port immediately after MCU reset. To use as an output port, set the DDR to 1 in the initialization routine of the program.

D Port Data Direction Registers (DDR0: $006, DDR1: $007, DDR2: $008, DDR3: $009): Each bit of the D port can be individually programmed as an input or output port. The D port becomes an input port at MCU reset (figure 15).

R Port Direction Register (RDR: $026): Ports R0, R1, R2, and RA can be individually programmed as input or output. When reset, each port becomes an input port (figure 16).

R3 to R9 Ports Data Direction Registers (R3DR: $00A, R4DR: $00B, R5DR: $00C, R6DR: $00D, R8DR: $00E, R9DR: $00F): Each bit of ports R3 to R9 can be individually programmed as an input port. Each port becomes an input at MCU reset (figure 17).

I/O Circuit Configuration: The basic port I/O circuit type is CMOS. (R7 is an input circuit, and RA is an NMOS open-drain I/O circuit.) See table 15.

Ports 8 and 9 are CMOS I/O circuits for the HD404418 and HD4074418, and NMOS open-drain I/O circuits for the HD4074408.

The direction of any type of I/O pin can be controlled by the data direction register.

To prevent floating input pins on the HD404418, a pull-up MOS can be attached to the I/O circuits (D0 to D15 and R0 to R6) via the mask option. It can be selected for any pin. For the HD4074418 and HD4074408, all circuits are specified as without pull-up MOS. See figures 18 to 20 for the circuit configurations.

Table 14 Data Direction Register (DDR)

DDR	Port Condition
0	Input port (output buffer off)
1	Output port (output buffer on)

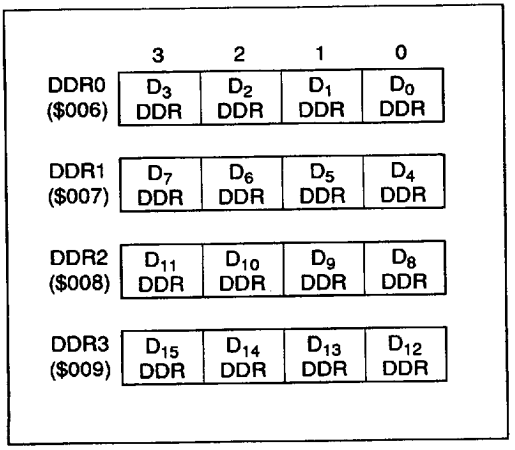


Figure 15 D Port Data Direction Registers

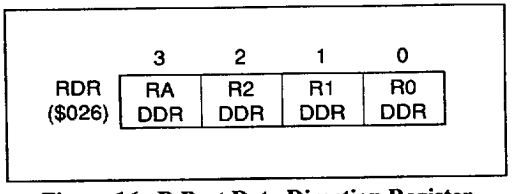


Figure 16 R Port Data Direction Register

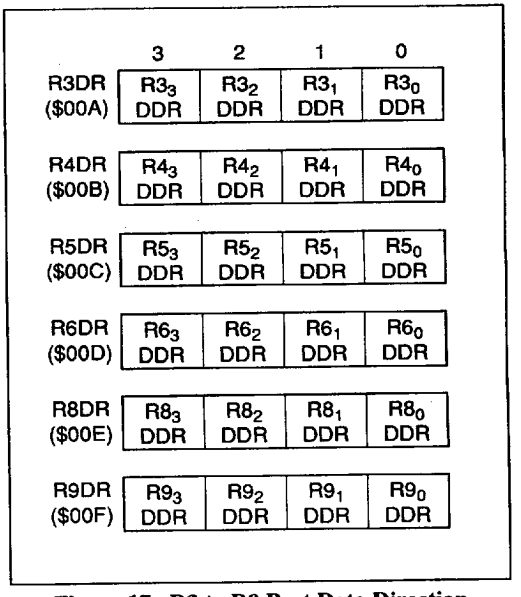


Figure 17 R3 to R9 Port Data Direction Registers

Table 15 I/O Pin Circuit Types

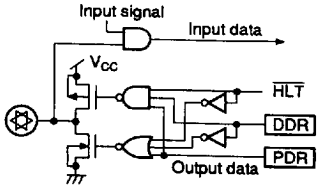
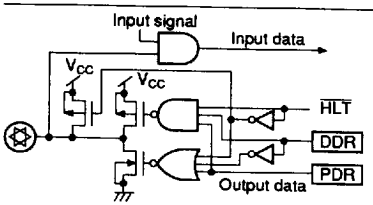
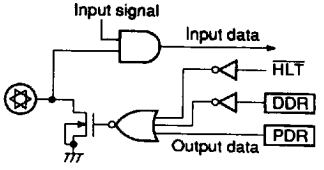
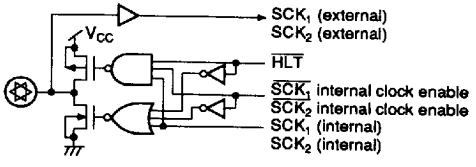
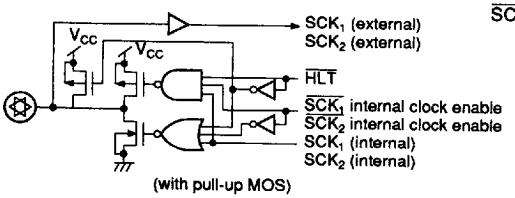
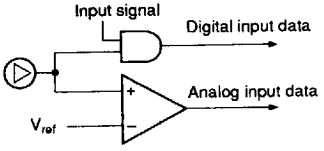
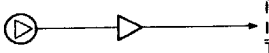
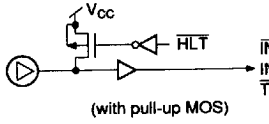
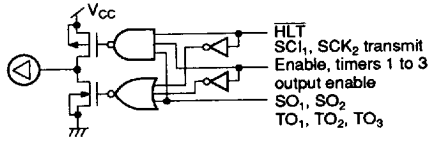
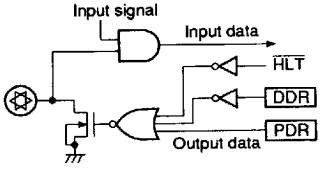
I/O Pins	Circuit	Applied Pins		
		HD404418	HD4074418	HD4074408
Standard I/O pins · high-current pins		D_0 to D_{15}* $R0_0$ to $R0_3$* $R1_0$ to $R1_3$* $R2_0$ to $R2_3$* $R3_0$ to $R3_3$* $R4_0$ to $R4_3$* $R5_0$ to $R5_3$* $R6_0$ to $R6_3$* $R8_0$ to $R8_3$ $R9_0$ to $R9_3$	D_0 to D_{15} $R0_0$ to $R0_3$ $R1_0$ to $R1_3$ $R2_0$ to $R2_3$ $R3_0$ to $R3_3$ $R4_0$ to $R4_3$ $R5_0$ to $R5_3$ $R6_0$ to $R6_3$ $R8_0$ to $R8_3$ $R9_0$ to $R9_3$	D_0 to D_{15} $R0_0$ to $R0_3$ $R1_0$ to $R1_3$ $R2_0$ to $R2_3$ $R3_0$ to $R3_3$ $R4_0$ to $R4_3$ $R5_0$ to $R5_3$ $R6_0$ to $R6_3$
	 (with pull-up MOS)	D_0 to D_{15}* $R0_0$ to $R0_3$* $R1_0$ to $R1_3$* $R2_0$ to $R2_3$* $R3_0$ to $R3_3$* $R4_0$ to $R4_3$* $R5_0$ to $R5_3$* $R6_0$ to $R6_3$*		
		RA_0, RA_1	RA_0, RA_1	RA_0, RA_1
		$\overline{SCK_1}$* $\overline{SCK_2}$*	$\overline{SCK_1}$ $\overline{SCK_2}$	$\overline{SCK_1}$ $\overline{SCK_2}$
	 (with pull-up MOS)	$\overline{SCK_1}$* $\overline{SCK_2}$*		

Table 15 I/O Pin Circuit Types (cont)

I/O Pins		Circuit	Applied Pins HD404418	HD4074418	HD4074408
Standard pins	Input pins		$R7_0$ to $R7_3$	$R7_0$ to $R7_3$	$R7_0$ to $R7_3$
			$\overline{INT_0}$, $\overline{INT_1}$* INT_2, INT_3* $\overline{INT_4}$, INT_5* $\overline{TI_1}$, TI_2* SI_1, SI_2*	$\overline{INT_0}$, $\overline{INT_1}$ INT_2, INT_3 $\overline{INT_4}$, INT_5 $\overline{TI_1}$, TI_2 SI_1, SI_2	$\overline{INT_0}$, $\overline{INT_1}$ INT_2, INT_3 $\overline{INT_4}$, INT_5 $\overline{TI_1}$, TI_2 SI_1, SI_2
		 (with pull-up MOS)	$\overline{INT_0}$, $\overline{INT_1}$* INT_2, INT_3* $\overline{INT_4}$, INT_5* $\overline{TI_1}$, TI_2* SI_1, SI_2*		
	Output pins		SO_1, SO_2 TO_1, TO_2, TO_3	SO_1, SO_2 TO_1, TO_2, TO_3	SO_1, SO_2 TO_1, TO_2, TO_3
Inter-mediate-voltage pins	I/O pins				$R8_0$ to $R8_3$ $R9_0$ to $R9_3$

Notes:
1. Either without pull-up MOS or with pull-up MOS can be specified for each pin via the mask option. (Circuit with pull-up MOS is provided only for the HD404418.)
2. $\overline{HLT}$ becomes 0 in stop mode, the I/O pins and output pins go to in high impedance, and all pull-up MOSs turn off.
3. When bit 2 of both serial clock registers are 1, pins $R4_0/\overline{SCK_1}$ and $R6_3/\overline{SCK_2}$ become $\overline{SCK_1}$ and $\overline{SCK_2}$ outputs, respectively, when $\overline{SCK_1}$ and $\overline{SCK_2}$ internal clock enable signals are set to 0 by placing the serial clock registers in transmit clock output mode. The pins become $\overline{SCK_1}$ and $\overline{SCK_2}$ outputs when $\overline{SCK_1}$ and $\overline{SCK_2}$ internal clock signals are set to 1 by placing the serial clock registers in transmit clock input mode.
4. When serial 1 and 2 are set to transmit or transmit/receive mode, the serial 1 and 2 transmit enable signal becomes 1, then SO_1 and SO_2 pins become outputs. Pins TO_1 to TO_3 become output if the timer output registers 1 to 3 are set to timer output state or PWM active state.

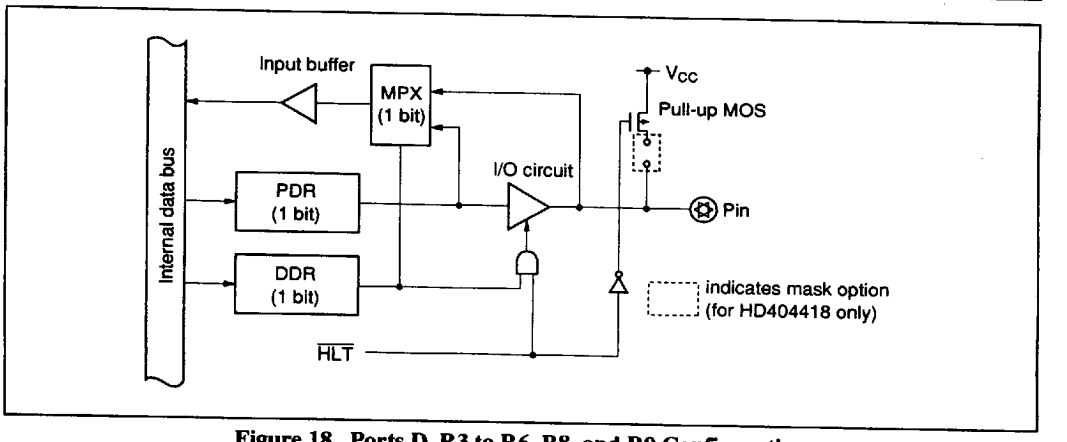


Figure 18 Ports D, R3 to R6, R8, and R9 Configuration

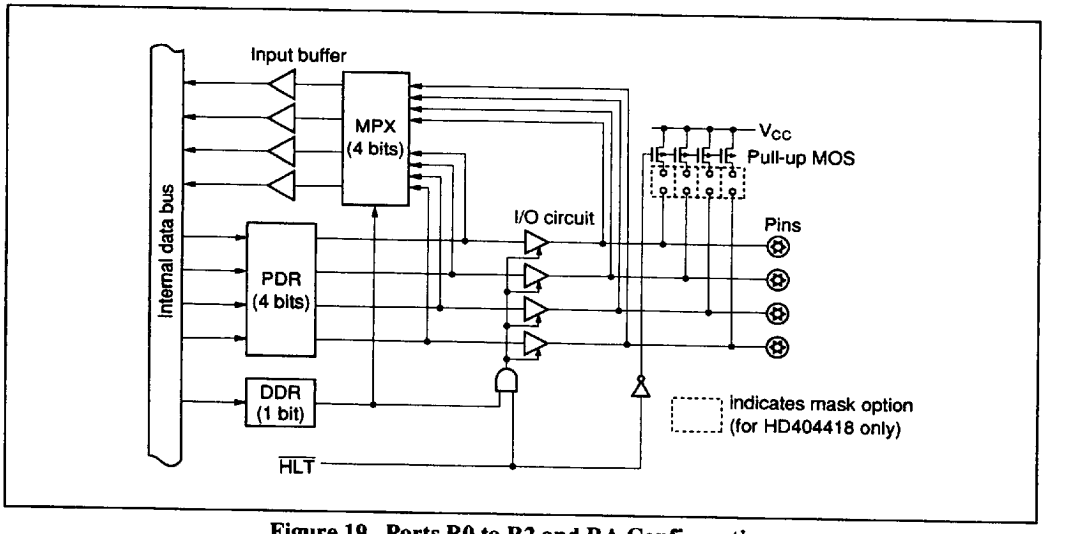


Figure 19 Ports R0 to R2 and RA Configuration

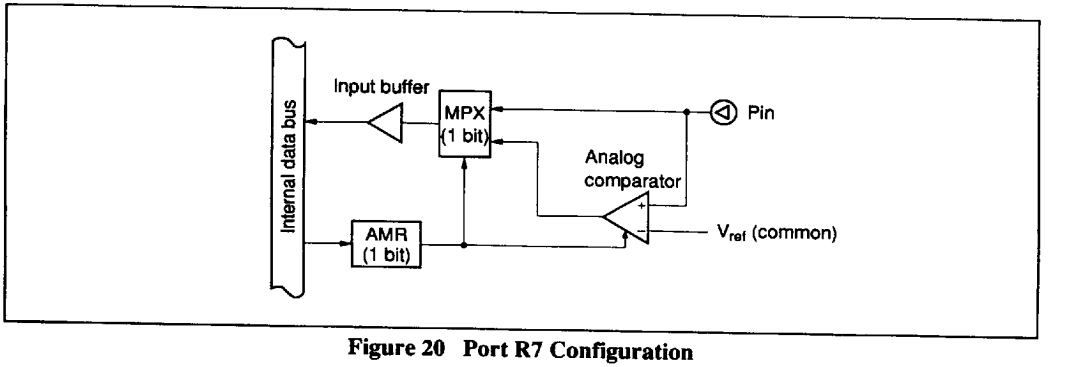


Figure 20 Port R7 Configuration

Pin Function Control: Several pins are multiplexed with the timer or serial interface I/O pins. The functions of these pins are controlled by the corresponding mode registers.

Port Mode Register (PMR: $027): The port mode register is a 4-bit write-only register which controls pins $R3_0/\overline{INT_0}$, $R3_1/\overline{INT_1}$, $R3_2/INT_2$, and R_3/INT_3, as shown in table 16. The port mode register is initialized to $0 by MCU reset. These pins are therefore initially used as ports.

IF0 or IF1 is set if $R3_0/\overline{INT_0}$ or $R3_1/\overline{INT_1}$ is low when bit 0 or bit 1 of the port mode register is set. IF2 or IF3 is set if $R3_1/INT_2$ or $R3_2/INT_3$ is high when bit 2 or bit 3 of the port mode register is set. Keep these points in mind when using external interrupts.

Analog Mode Register (AMR: $025): The analog mode register is a 4-bit write-only register. Each bit controls the operation modes of pin $R7_0$ to $R7_3$ inputs (table 16). The data direction register of the $R4_3$ pin is automatically reset when the analog input mode is selected by setting a bit of the analog mode register. The analog comparator then receives the reference voltage (V_{ref}) of the analog input.

Note that all bits of the analog mode register are initialized to 0 at MCU reset. Thus, pins $R7_0$ to $R7_3$ go into the digital input mode at MCU reset.

Table 16 Pin Function Control

PMR (Port Mode Register: $027)

	Bit 3	Bit 2	Bit 1	Bit 0
Pin	$R3_3/INT_3$	$R3_2/INT_2$	$R3_1/\overline{INT_1}$	$R3_0/\overline{INT_0}$
Bit = 0	$R3_3$	$R3_2$	$R3_1$	$R3_0$
Bit = 1	INT_3	INT_2	$\overline{INT_1}$	$\overline{INT_0}$

AMR (Analog Mode Register: $025)

	Bit 3	Bit 2	Bit 1	Bit 0
Pin	$R7_3$	$R7_2$	$R7_1$	$R7_0$
Bit = 0	Digital	Digital	Digital	Digital
Bit = 1	Analog	Analog	Analog	Analog

AMR (Analog Mode Register: $025)

Bit 3 to 0	$R4_3/V_{ref}$
0 0 0 0	$R4_3$
not 0 0 0 0	VC_{ref}

SMR1 (Serial Mode Register 1: $010)

	Bit 2	Bit 1	Bit 0
Pin	$R4_0/\overline{SCK_1}$	$R4_1/SI_1$	$R4_2/SO_1$
Bit = 0	$R4_0$	$R4_1$	$R4_2$
Bit = 1	$\overline{SCK_1}$	SI_1	SO_1

SMR2 (Serial Mode Register 2: $014)

	Bit 2	Bit 1	Bit 0
Pin	$R6_3/\overline{SCK_2}$	$R6_2/SI_2$	$R6_1/SO_2$
Bit = 0	$R6_3$	$R6_2$	$R6_1$
Bit = 1	$\overline{SCK_2}$	SI_2	SO_2

Serial Mode Registers (SMR1: $010, SMR2: $014): The serial mode registers are 4-bit read/write registers which control pins $R4_0/\overline{SCK_1}$, $R6_3/\overline{SCK_2}$, $R4_1/SI_1$, $R6_2/SI_2$, $R4_2/SO_1$, and $R6_1/SO_2$, as shown in table 16. For details, refer to the Serial Interface section.

Timer Mode Registers (TMR1: $018, TMR2: $01C): The timer mode registers are 4-bit write-only registers which control pins $R5_0/\overline{INT_4}/\overline{TI_1}$ and $R5_1/INT_5/TI_2$ as shown in table 16. For details, refer to the Timer section.

Timer Output Registers (TOR1: $019, TOR2: $01D, TOR3: $021): The timer output registers are 4-bit read/write registers which control pins $R5_0/\overline{INT_4}/\overline{TI_1}$, $R5_1/INT_5/TI_2$, $R5_2/TO_1$, $R5_3/TO_2$, and $R6_0/TO_3$, as shown in table 16. For details, refer to the Timer section.

Unused I/O Pins: If unused I/O pins are left floating, the LSI may malfunction because of noise. To prevent this, the unused pins should be dealt with as follows:

Standard I/O pins:	Pull up to V_{CC} through a resistor of about 100 kΩ.
Standard input pins:	Connect to V_{CC}.
Intermediate-voltage pins:	Connect to GND.

Unused pins must retain the state of reset in the program. The program should not change DDR and register values.

Table 16 Pin Function Control (cont)

TMR1 (Timer Mode Register 1: $018)

Bits 2 to 0	$R5_0/\overline{INT_4}/\overline{TI_1}$
not 1 1 1	$R5_0/\overline{INT_4}$
1 1 1	$\overline{TI_1}$

TMR2 (Timer Mode Register 2: $01C)

Bits 2 to 0	$R5_1/INT_5/TI_2$
not 1 1 1	$R5_1/INT_5$
1 1 1	TI_2

TOR1 (Timer Output Register 1: $019)

Bit 3	$R5_0/\overline{INT_4}/\overline{TI_1}$
0	$R5_0/\overline{TI_1}$
1	$\overline{INT_4}$

Bit 2	Bit 1	Bit 0	$R5_2/TO_1$
0	0	0	$R5_2$
0	0	1	TO_1
0	1	0	
0	1	1	
1	×	×	

TOR2 (Timer Output Register 2: $01D)

Bit 3	$R5_1/INT_5/TI_2$
0	$R5_1/TI_2$
1	INT_5

Bit 2	Bit 1	Bit 0	$R5_3/TO_2$
0	0	0	$R5_3$
0	0	1	TO_2
0	1	0	
0	1	1	
1	×	×	

TOR3 (Timer Output Register 3: $021)

Bit 2	Bit 1	Bit 0	$R6_0/TO_3$
0	0	0	$R6_0$
0	0	1	TO_3
0	1	0	
0	1	1	
1	×	×	

Note: × = Don't care

Timers

The MCU contains a prescaler and 4 timer/counters (timers 1 to 4). Figures 21 and 22 show their block diagrams.

Timers 1 to 3 are versatile timer/counters providing the following functions: free-running timers, reload timer/counters, and a pulse width modulation (PWM) circuit. These functions are controllable by software. Each timer provides a timer output pin which permits the MCU to transmit the clock signal with an appropriate cycle in combination with the reload function and the PWM output. Timer 4 is a free-running timer which is also available as a watchdog timer (WDT) provided with a hardware reset function selectable by the mask option.

Prescaler: The input clock to the prescaler is the system clock signal. The prescaler is initialized to $000 at MCU reset, then it starts to count up by the system clock. The prescaler keeps counting up except at MCU reset and stop mode. Prescaler outputs provide timer input clocks and serial interface transmit clocks. Their divide ratios can be selected by the timer mode registers (TMR1, TMR4), and the serial clock registers (SCR1, SCR2), respectively.

Timers 1, 2, and 3 Operation: Timers 1 to 3 are versatile timers provided with the following functions: free-running timers, event counters, reload timers, and a PWM circuit. The functions are controllable by software and selectable by the timer mode registers (TMR1, TMR2, TMR3) and the timer output registers (TOR1, TOR2, TOR3) for each timer.

The timer counters (TC1 to TC3) count up with every input clock after they have been initialized at MCU reset. When used as timer counters, the clock divided by the prescaler is available as an input clock. When used as an event counter, an external clock is available. When selecting an external clock input, an interrupt enable bit in the timer output register must be reset in order to disable the external interrupt (timer 3 does not have an external clock input).

The timer interrupt request flags (IFT1, IFT2, IFT3) are set with the input clock after the timer counter has become $FF. When the auto-reload function is not selected, timers 1 to 3 become free-running timer/event counters, and restart counting up after they have been reinitialized to $00. When the auto-reload function is selected, timers 1 to 3 become reload timers, and the data in the timer load registers (TL1R, TL2R, TL3R) are reloaded into the timer counter with the input clock after the timer counter reaches $FF. Then timers 1 to 3 start counting up.

Timers 1 to 3 also provide a timer output circuit, which affects the output level when the input clock is applied after the timer counter has become $FF. This circuit can transmit a clock signal (with an appropriate cycle) in combination with the reload timer. If the PWM function is selected by the timer output register, PWM output is available. The PWM output transmits high during the clock cycle specified in the timer load register (1 cycle = 1 timer input clock × 256), and transmits low otherwise. To obtain a voltage level in proportion to the timer load register value, combine the PWM output with a low-pass filter.

Timer 4 Operation: Timer 4 is an 8-bit free-running/watchdog timer. When timer 4 is used as a free-running timer, 8 clocks divided by the prescaler can be selected as input clocks. Timer 4 is initialized to $00 at MCU reset, then counts up every input clock signal. If a clock signal is applied after the timer becomes $FF, the timer returns to $00, then continues counting. At the same time the timer interrupt request flag is set.

When timer 4 is used as a watchdog timer, the input clock is specified as the 1/2048 output divided by the prescaler. The watchdog timer is initialized to $00 at MCU reset, then counts up every input clock signal. If a clock signal is applied after the timer becomes $FF, an overflow is generated and the hardware reset function is enabled for the MCU.

After reset, the MCU re-executes the program from the beginning. Therefore, to operate the system normally, the program should set the watchdog timer reset bit in a cycle shorter than 2^{19}. The program resets the MCU at overflow generation if the MCU malfunctions because of noise. This function is effective for improving system reliability.

The timer 4 function of the HD404418 is selected via the mask option. For the HD4074418 and HD4074408, watchdog timer versions are provided in addition to free-running timer versions.

Note that the type names of the watchdog timer versions differ from that of the free-running timer versions.

Timer Mode Registers 1 to 3 (TMR1: $018, TMR2: $01C, TMR3: $020): 4-bit write only timer mode registers 1 to 3 select the auto-reload function, the prescaler divide ratio, and the source of the input clock signal (table 17). Timer mode registers 1 to 3 are initialized to $0 at MCU reset.

Writing to this register is valid from the second instruction execution cycle after the execution of the previous timer mode registers 1 to 3. Initialize timers 1 to 3 by programming the timer load register after the data of the TMR are changed.

Note that timer 3 does not provide an event input pin. Therefore, timer 3 is unavailable as an event counter.

Timer Mode Register 4 (TMR4: $024): Timer mode register 4 is a write only register. The function differs depending on the timer 4 function. When timer 4 functions as a free-running timer, bits 0 to 2 affect the operation of timer 4 (table 18). When users select the watchdog timer by the mask option, bit 3 affects the timer 4 (watchdog timer) operation.

Table 17 Timer Mode Registers 1 to 3

TMR1, TMR2, TMR3			
Bit 3			**Auto-Reload Function**
0			No
1			Yes

TMR1, TMR2, TMR3			
Bit 2	**Bit 1**	**Bit 0**	**Prescaler Divide Ratio, Clock Input Source**
0	0	0	÷ 2048
0	0	1	÷ 512
0	1	0	÷ 128
0	1	1	÷ 32
1	0	0	÷ 8
1	0	1	÷ 4
1	1	0	÷ 2
1	1	1	$\overline{TI}_1$, TI_2 (external event inputs)

Note: External event input is unavailable for timer 3, since timer 3 does not provide an event input pin.
If external event input is selected at timers 1 and 2, ports $R5_0$ and $R5_1$ data direction registers are automatically reset before becoming inputs.

Table 18 Timer Mode Register 4

TMR4	
Bit 3	**Function**
0	Does not affect the watchdog timer operation
1	Watchdog timer is reset, then restarts counting up from $00

Note: Bit 3 is available when timer 4 is functioning as a watchdog timer. When timer 4 is functioning as a free-running timer, bit 3 is unavailable.

TMR4			
Bit 2	**Bit 1**	**Bit 0**	**Prescaler Divide Ratio**
0	0	0	÷ 2048
0	0	1	÷ 1024
0	1	0	÷ 512
0	1	1	÷ 128
1	0	0	÷ 32
1	0	1	÷ 8
1	1	0	÷ 4
1	1	1	÷ 2

Note: Bits 0 to 2 are available when timer 4 is functioning as a free-running timer. When timer 4 is functioning as a watchdog timer, these bits become independent of timer 4's operation.

Timer Output Registers (TOR1: $019, TOR2: $01D, TOR3: $021): The timer output registers are 4-bit read/write registers which control timers 1 to 3 output modes, PWM output selection, and external interrupts multiplexed with the timer input pin (table 19).

When bits 0 and 1 of the timer output registers go into any mode other than the timer output inhibit mode, the pins become timer output pins automatically. These modes are available for transmitting clock signals (with appropriate cycles and duties) in combination with each mode of timers 1 to 3. When the PWM output is selected, the timer output pin becomes PWM output independent of the timer output mode.

The data of the timer output register changes after the second instruction cycle of writing into the timer output register. Timer output register 3 becomes a 3-bit register without bit 3, since timer 3 does not provide an event input pin.

Timers 1 to 3 (TC1L: $01A, TC1U: $01B, TL1L: $01A, TL1U: $01B, TC2L: $01E, TC2U: $01F, TL2L: $01E, TL2U: $01F, TC3L: $022, TC3U: $023, TL3L: $022, TL3U: $023): Timers 1 to 3 consist of an 8-bit write-only timer load register and an 8-bit read-only timer counter. Each of them provides low order digits (TC1L: $01A, TC2L: $01E, TC3L: $022) and high-order digits (TC1U: $01B, TC2U: $01F, TC3U: $023). Refer to figure 20.

The timer counter can be initialized by programming the timer load register. In this case, write the low-order digits first, then the high-order digits. The timer counter is initialized according to the timer load register value when the high-order digit is written. The timer load register is initialized to $00 at MCU reset.

The counter value of timers 1 to 3 can be obtained by reading the timer counter. In this case, read the high-order digit first, then the low-order digit. The count value of the low-order digit is latched when the high-order digit is read.

Table 19 Timer Output Registers

TOR1, TOR2

Bit 3	$\overline{INT_4}$, INT_5 External Interrupts
0	Inhibits the $\overline{INT_4}$ and INT_5 external interrupts
1	Enables the $\overline{INT_4}$ and INT_5 external interrupts

Note: Port $R5_0$ and $R5_1$ data direction registers are automatically reset when bit 3 of TOR1 and TOR2 are set, then become inputs.

TOR1, TOR2, TOR3

Bit 2	PWM Function
0	Normal timer operation
1	PWM operation, transmits data from TO_1, TO_2, and TO_3

Note: The data direction register for ports $R5_2$, $R5_3$, and $R6_0$ are automatically set when bit 2 of TOR1, TOR2, and TOR3 are set, then become outputs.

TOR1, TOR2, TOR3

Bit 1	Bit 0	Timer Output Mode
0	0	Inhibits timer output
0	1	Transmits 1 synchronously with timer overflow output (toggle output mode)
1	0	Transmits the inverse level synchronously with timer output (1 output mode)
1	1	Transmits 0 synchronously with timer overflow output (0 output mode)

Note: The data direction registers for ports $R5_2$, $R5_3$, and $R6_0$ are automatically set when bits 0 and 1 for TOR1, TOR2, and TOR3 are not 00.

Operation Mode of Timers 1, 2, and 3: Timers 1 to 3 provide the following three timer functions: free-running, reload, and PWM. Internal/external clock input and the absence/presence of the timer output are also selectable. The combination of these functions are available for selecting a variety of operation modes.

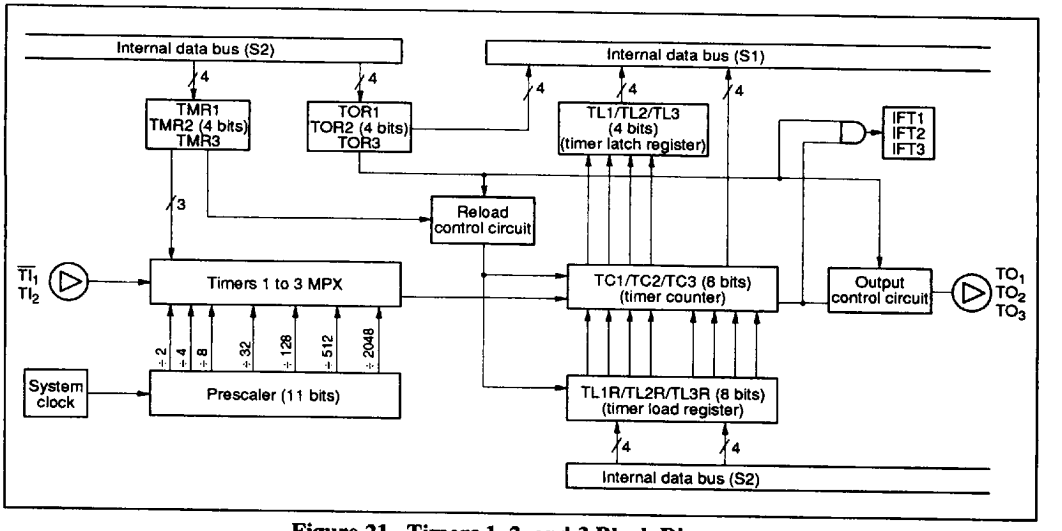
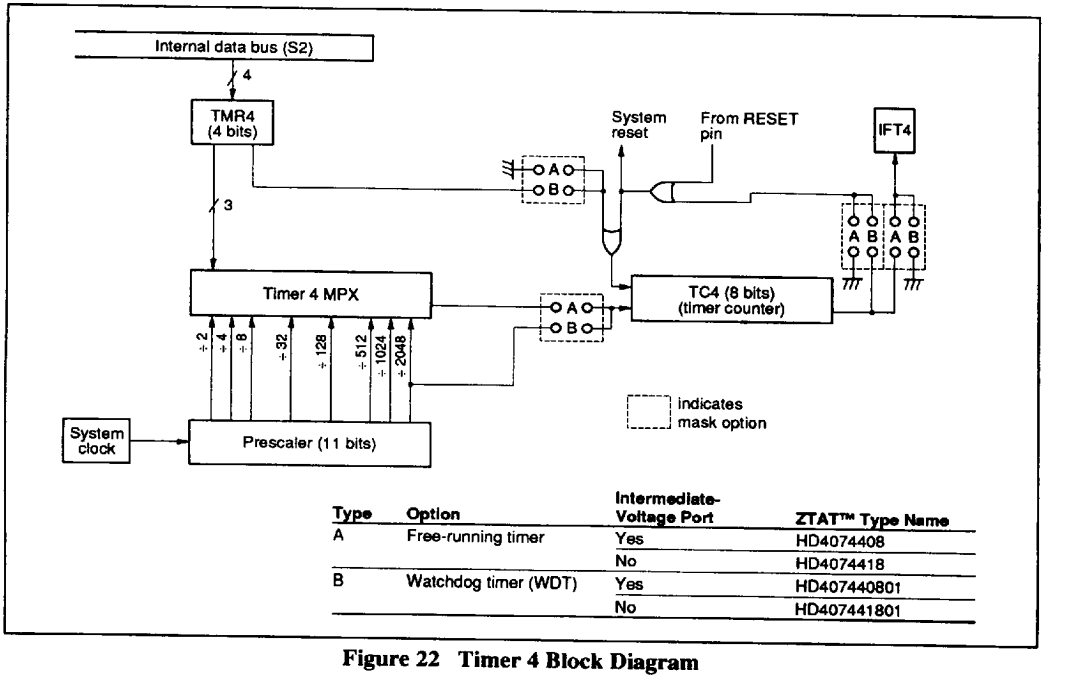


Figure 21 Timers 1, 2, and 3 Block Diagram



Type	Option	Intermediate-Voltage Port	ZTAT™ Type Name
A	Free-running timer	Yes	HD4074408
		No	HD4074418
B	Watchdog timer (WDT)	Yes	HD407440801
		No	HD407441801

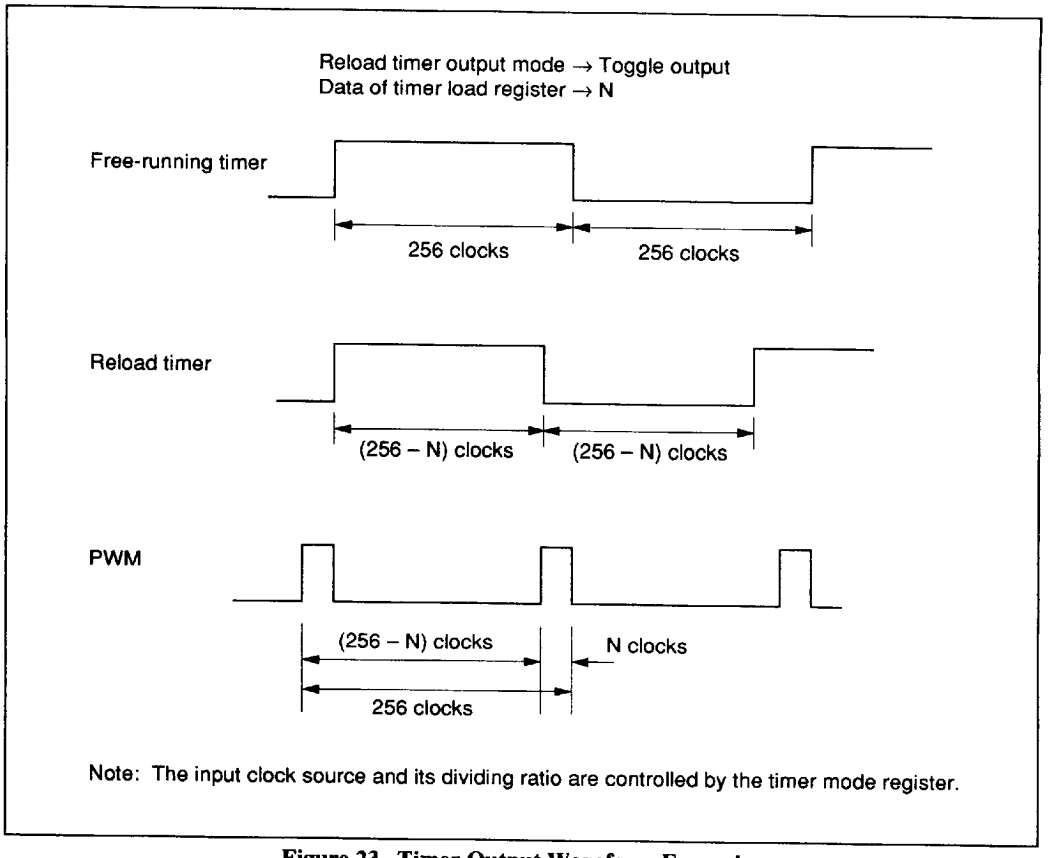
Figure 22 Timer 4 Block Diagram

Table 20 describes the combinations of timers 1 to 3 operation mode selection. Select the operation mode according to table 20 in order to set the appropriate data to the timer mode register and the timer output register. Figure 23 illustrates an example of a timer output waveform. Note that the output waveform differs depending on operation modes.

Table 20 Combinations of Timers 1, 2, and 3 Operation Modes

TMR1, TMR2, TMR3				TOR1, TOR2, TOR3						
Bit 3	Bit 2	Bit 1	Bit 0	Bit 3	Bit 2	Bit 1	Bit 0	Timer Input Pin	Timer Output Pin	Timers 1 to 3 Function
0	*3	*3	*3	0	0	0	0	$R5_0$, $R5_1$	$R5_2$, $R5_3$, $R6_0$	Free-running timer
0	*3	*3	*3	0	0	*2	*2	$R5_0$, $R5_1$	TO_1, TO_2, TO_3	
0	*3	*3	*3	1	0	0	0	$\overline{INT_4}$, INT_5	$R5_2$, $R5_3$, $R6_0$	
0	*3	*3	*3	1	0	*2	*2	$\overline{INT_4}$, INT_5	TO_1, TO_2, TO_3	
0	1	1	1	0	0	0	0	$\overline{TI_1}$, TI_2	$R5_2$, $R5_3$, $R6_0$	Event counter
0	1	1	1	0	0	*2	*2	$\overline{TI_1}$, TI_2	TO_1, TO_2, TO_3	
1	*3	*3	*3	0	0	0	0	$R5_0$, $R5_1$	$R5_2$, $R5_3$, $R6_0$	Reload timer
1	*3	*3	*3	0	0	*2	*2	$R5_0$, $R5_1$	TO_1, TO_2, TO_3	
1	*3	*3	*3	1	0	0	0	$\overline{INT_4}$, INT_5	$R5_2$, $R5_3$, $R6_0$	
1	*3	*3	*3	1	0	*2	*2	$\overline{INT_4}$, INT_5	TO_1, TO_2, TO_3	
1	1	1	1	0	0	0	0	$\overline{TI_1}$, TI_2	$R5_2$, $R5_3$, $R6_0$	Event counter (with reload function)
1	1	1	1	0	0	*2	*2	$\overline{TI_1}$, TI_2	TO_1, TO_2, TO_3	
*1	*3	*3	*3	0	1	*1	*1	$R5_0$, $R5_1$	TO_1, TO_2, TO_3	PWM *4
*1	*3	*3	*3	1	1	*1	*1	$\overline{INT_4}$, INT_5	TO_1, TO_2, TO_3	
*1	1	1	1	0	1	*1	*1	$\overline{TI_1}$, TI_2	TO_1, TO_2, TO_3	PWM (external clock) *4

Notes: 1. Don't care.
2. One or both of bits 0 and 1 of TOR is 1.
3. Bits 0, 1, and 2 of TMR are not 111.
4. When PWM output is selected, the timer functions the same way as the free-running timer by initializing the timer load register to $00.



Note: The input clock source and its dividing ratio are controlled by the timer mode register.

Figure 23 Timer Output Waveform Examples

Serial Interface

The MCU provides two serial interfaces to transmit/receive 8-bit data serially. The serial interfaces consist of the serial data registers (SR1, SR2), the serial mode registers (SMR1, SMR2), the serial clock registers (SCR1, SCR2), the octal counter, and the multiplexer (figure 24).

The serial mode register controls pins $R4_0/\overline{SCK_1}$, $R6_3/\overline{SCK_2}$, $R4_1/SI_1$, $R6_2/SI_2$, $R4_2/SO_1$, and $R6_1/SO_2$, and the enable/disable of the STS instruction. The serial clock register controls the transmit clock. Writing/reading the contents of the serial data register can be done by software. The data in the serial data register shifts synchronously with the transmit clock signal.

The octal counter is reset to \$0 by the STS instruction. It starts to count at the falling edge of the transmit clock signal ($\overline{SCK_1}$, $\overline{SCK_2}$) and increments by one at the rising edge of the transmit clock. The serial interrupt request flag is set when the octal counter is reset by eight transmit clock signals or by discontinued data transfer.

Each serial mode register has an enable bit (bit 3) which controls a serial interface with a single STS instruction. To activate a serial interface, set the enable bit of that interface, then execute the STS instruction. The serial interface in which an enable bit is set then starts functioning. When both enable bits are reset, neither serial interface functions with the STS instruction. When both enable bits are set, the two serial interfaces both start functioning with a single STS instruction. Enable bits are automatically reset by the STS instruction. Therefore, an enable bit must be set to activate a serial interface before executing the STS instruction.

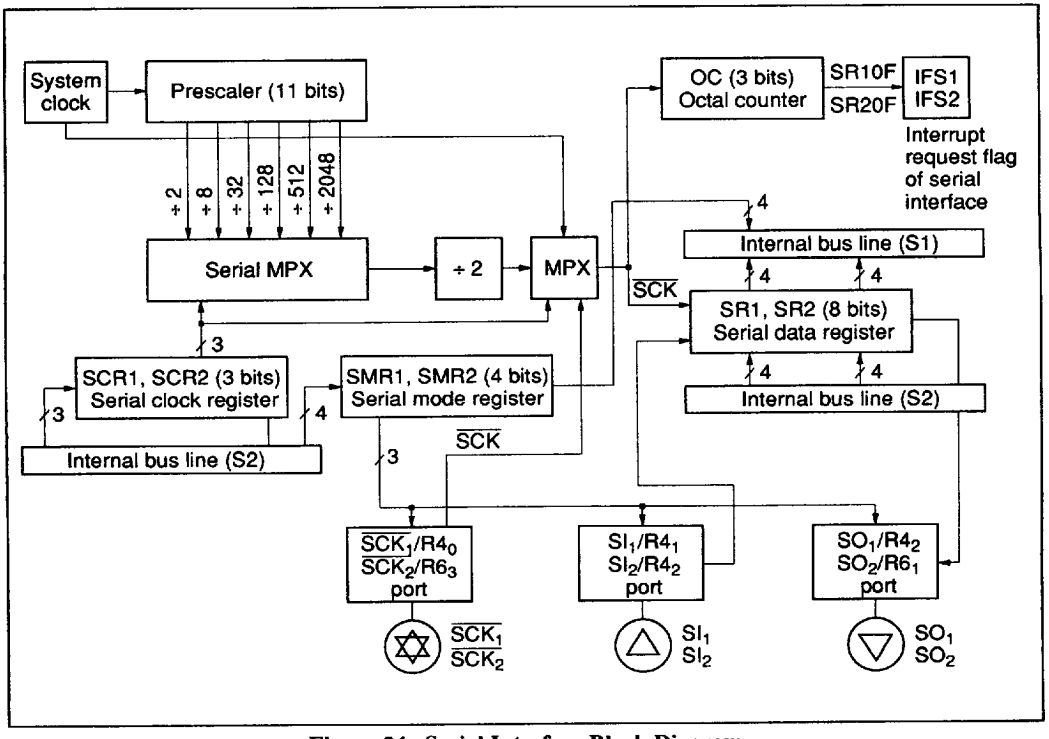


Figure 24 Serial Interface Block Diagram

Serial Mode Register (SMR1: $010, SMR2: $014): The serial mode register is a 4-bit read/write register, which controls the serial interface operation, $\overline{SCK_1}$, $\overline{SCK_2}$, SI_1, SI_2, SO_1, and SO_2 (table 21).

The write signal to the serial mode register initializes the internal state of the serial interface. The write signal stops the transmit clock supplied to the serial data register and the octal counter, and resets the octal counter to $0 simultaneously. Thus, the write signal to the serial mode register causes the data transfer to quit and the serial interrupt request flag to be set while the serial interface is operating.

Data in the serial mode register can be changed in the second instruction cycle after writing into the serial mode register. Therefore, program the serial interface to execute the STS instruction after the data in the serial mode register has been changed completely. The serial mode register is initialized to $0 at MCU reset.

Bit 3 of the serial mode register is an enable bit of the serial interface. Set the enable bit before executing the STS instruction. To activate both serial interfaces, set the enable bits of both serial interfaces, then execute a single STS instruction. An enable bit is automatically reset by the STS instruction execution.

Table 21 Serial Mode Register

SMR1, SMR2

Bit 3	STS Instruction
0	Disabled
1	Enabled

Note: Bit 3 of the SMR1 and SMR2 are automatically reset by the STS instruction.

SMR1, SMR2

Bit 2	$R4_0/\overline{SCK_1}$, $R6_3/\overline{SCK_2}$
0	Used as $R4_0$ and $R6_3$ I/O pins
1	Used as $\overline{SCK_1}$ and $\overline{SCK_2}$ I/O pins

Note: When the internal clocks are selected, ports $R4_0$ and $R6_3$ data direction registers are automatically set by bit 2 of both the SMR1 and SMR2, then become outputs. When the external clocks are selected, ports $R4_0$ and $R6_3$ data direction registers are automatically reset by bit 2 of both the SMR1 and SMR2, then become inputs.

SMR1, SMR2

Bit 1	$R4_1/SI_1$, $R6_2/SI_2$
0	Used as $R4_1$ and $R6_2$ I/O pins
1	Used as SI_1 and SI_2 input pins

Note: Ports $R4_1$ and $R6_2$ data direction registers are automatically reset by bit 1 of both the SMR1 and SMR2, then become inputs.

SMR1, SMR2

Bit 0	$R4_2/SO_1$, $R6_1/SO_2$
0	Used as $R4_2$ and $R6_1$ I/O pins
1	Used as SO_1 and SO_2 output pins

Note: $R4_2$ and $R6_1$ ports data direction registers are automatically set by bit 0 of both the SMR1 and SMR2, then become outputs.

Serial Clock Register (SCR1: $011, SCR2: $015): The serial clock register is a 3-bit write-only register which controls the transmit clock source and the prescaler divide ratio (table 22).

The write signal to the serial clock register initializes the internal state of the serial interface. The transmit clock signal to the serial data register and the octal counter is stopped by programming the serial clock register, which resets the octal counter. Thus, data transfer is stopped and the interrupt request flag is set by programming the serial clock register during serial interface operation.

Serial Data Register (SR1L: $012, SR2L: $016, SR1U: $013, SR2U: $017): The serial data register is an 8-bit read/write register consisting of low-order digits (SR1L, SR2L) and high-order digits (SR1U, SR2U).

The data in the serial data register is transmitted LSB first at SO_1 and SO_2 synchronously with the falling edge of the transmit clock signal. At the same time, the serial data register receives external data LSB first at SI_1 and SI_2 synchronously with the rising edge of the transmit clock. Figure 25 shows the I/O timing chart for the transmit clock signal and the data.

The validity of the data contents cannot be assured when writing/reading the serial data register during data transfer. Therefore, write/read the serial data register after completing the data transfer.

Table 22 Serial Clock Register

SCR1, SCR2						
Bit 2	Bit 1	Bit 0	$R4_0/\overline{SCK_1}$ $R6_3/\overline{SCK_2}$	Clock Source	Prescaler Divide Ratio	System Clock Divide Ratio
0	0	0	$\overline{SCK_1}$, $\overline{SCK_2}$ output	Prescaler	2048	4096
0	0	1	$\overline{SCK_1}$, $\overline{SCK_2}$ output	Prescaler	512	1024
0	1	0	$\overline{SCK_1}$, $\overline{SCK_2}$ output	Prescaler	128	256
0	1	1	$\overline{SCK_1}$, $\overline{SCK_2}$ output	Prescaler	32	64
1	0	0	$\overline{SCK_1}$, $\overline{SCK_2}$ output	Prescaler	8	16
1	0	1	$\overline{SCK_1}$, $\overline{SCK_2}$ output	Prescaler	2	4
1	1	0	$\overline{SCK_1}$, $\overline{SCK_2}$ output	System clock		1
1	1	1	$\overline{SCK_1}$, $\overline{SCK_2}$ input	External clock		

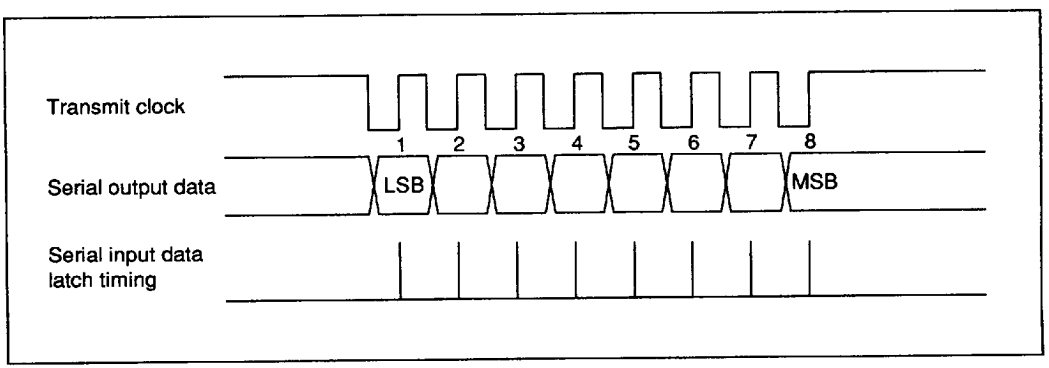


Figure 25 Serial Interface I/O Timing

Selection of the Operation Modes: Table 23 shows the operation modes of the serial interface. Set the serial mode register according to this table.

Operating State of the Serial Interface: The serial interfaces provide four operation states (figure 26).

When the serial interface has been initialized, the MCU is in transfer inhibit state in which the STS instruction and the transmit clocks are both ignored.

Set the enable bit of the serial interface to place the serial interface into the STS waiting state in which serial data transfer is available by executing the STS instruction. Two serial interfaces operate simultaneously with a single STS instruction when the enable bits of serial interfaces 1 and 2 are both set.

In the transmit clock wait state, the falling edge of the first transmit clock places the serial interface into the transfer state, while the octal counter counts up and the serial data register shifts simultaneously. If the clock continuous output mode is selected, the serial interface stays in the transmit clock wait state while continuously outputting transmit clocks.

The octal counter is initialized to 000 by 8 transmit clocks in the transfer state, and the serial interrupt request flag is set. When the internal transmit clock is selected, the MCU goes into the transfer inhibit state. When the external clock is selected, the MCU goes into the transmit clock wait state. The octal counter is initialized to $000, transfer is suspended, and the interrupt request flag is set simultaneously by programming the serial mode register or serial clock register in the transfer state or the transfer inhibit state.

An Example of Transmit Clock Error Detection: The serial interface malfunctions when the transmit clock is disturbed by external noise. In this case, the transmit clock error can be detected by the procedure shown in figure 27.

If more than eight transmit clocks are applied by external noise in the transmit clock wait state, the state of the serial interface shifts in the following sequence: transfer state (for 1 to 7 transmit clocks), transmit clock wait state (at 8th transmit clock), and back to transfer state. To set the serial interrupt flag again after resetting it, program the serial mode register to place the MCU into the STS waiting state.

Table 23 Serial Interface Operation Modes

SMR1, SMR2			Serial Interface
Bit 2	Bit 1	Bit 0	Operation Mode
1	0	0	Clock continuous output mode
1	0	1	Transmit mode
1	1	0	Receive mode
1	1	1	Transmit/receive mode

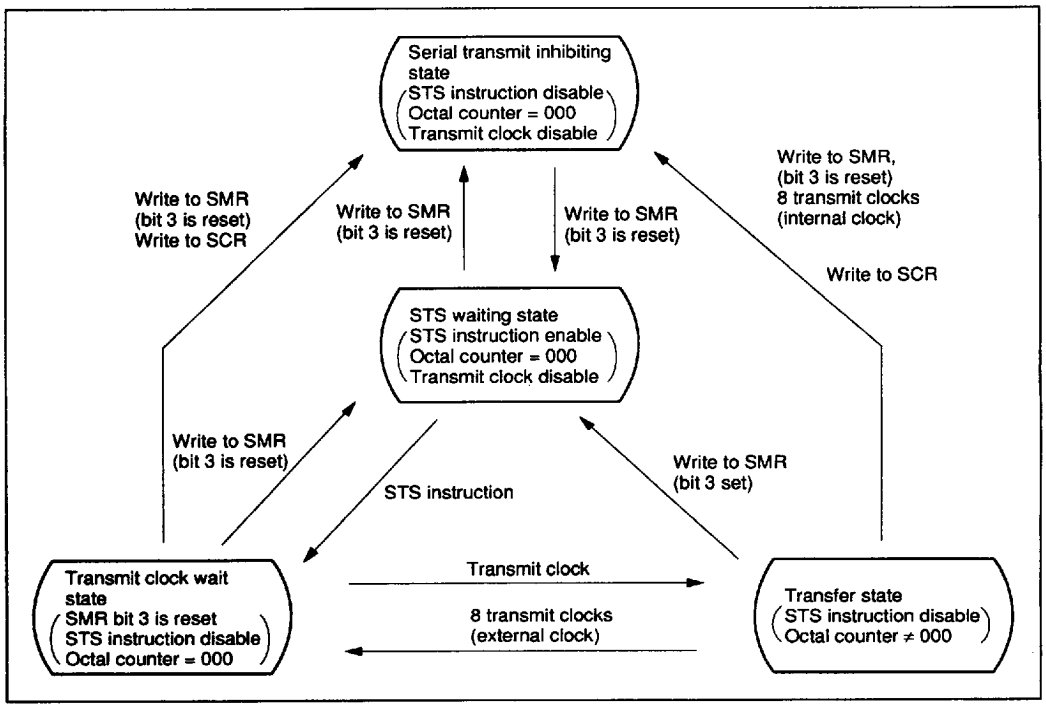


Figure 26 Serial Interface Operation State

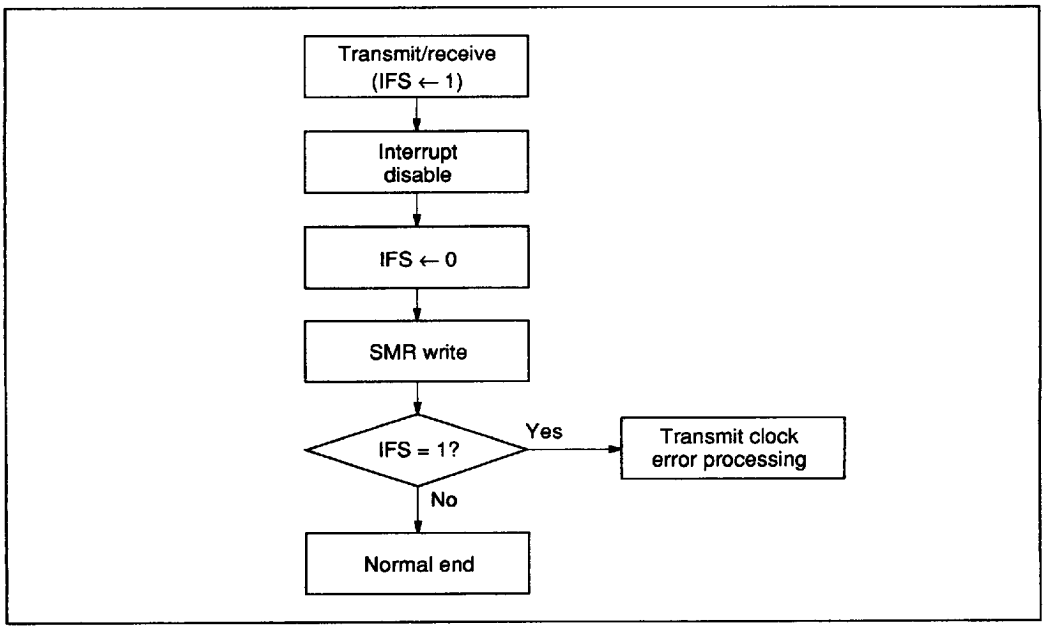


Figure 27 Transmit Clock Error Detection

PROM Mode Pin Description

Table 24 describes the pin functions in PROM mode.

Table 24 PROM Mode Signals

Pin No.			MCU Mode		PROM Mode	
DC-64S DP-64S	FP-64	FP-64A	Symbol	I/O	Symbol	I/O
1	59	57	D_{11}	I/O	V_{CC}	
2	60	58	D_{12}	I/O		
3	61	59	D_{13}	I/O		
4	62	60	D_{14}	I/O	A_{13}	I
5	63	61	D_{15}	I/O	A_{14}	I
6	64	62	$R0_0$	I/O	A_1	I
7	1	63	$R0_1$	I/O	A_2	I
8	2	64	$R0_2$	I/O	A_3	I
9	3	1	$R0_3$	I/O	A_4	I
10	4	2	$R1_0$	I/O	A_5	I
11	5	3	$R1_1$	I/O	A_6	I
12	6	4	$R1_2$	I/O	A_7	I
13	7	5	$R1_3$	I/O	A_8	I
14	8	6	$R2_0$	I/O	A_0	I
15	9	7	$R2_1$	I/O	A_{10}	I
16	10	8	$R2_2$	I/O	A_{11}	I
17	11	9	$R2_3$	I/O	A_{12}	I
18	12	10	RA_0	I/O	V_{PP}	
19	13	11	RA_1	I/O	A_9	I
20	14	12	$R3_0/\overline{INT_0}$	I/O		
21	15	13	$R3_1/\overline{INT_1}$	I/O		
22	16	14	$R3_2/INT_2$	I/O		
23	17	15	$R3_3/INT_3$	I/O		
24	18	16	$R5_0/\overline{INT_4}/\overline{TI_1}$	I/O		
25	19	17	$R5_1/INT_5/TI_2$	I/O		
26	20	18	$R5_2/TO_1$	I/O		
27	21	19	$R5_3/TO_2$	I/O		
28	22	20	$R6_0/TO_3$	I/O		
29	23	21	$R6_1/SO_2$	I/O		
30	24	22	$R6_2/SI_2$	I/O		
31	25	23	$R6_3/\overline{SCK_2}$	I/O		
32	26	24	V_{CC}		V_{CC}	
33	27	25	$R4_0/\overline{SCK_1}$	I/O		
34	28	26	$R4_1/SI_1$	I/O		
35	29	27	$R4_2/SO_1$	I/O		
36	30	28	$R4_3/V_{ref}$	I/O		
37	31	29	$R7_0$	I	$\overline{CE}$	I
38	32	30	$R7_1$	I	$\overline{OE}$	I
39	33	31	$R7_2$	I	$\overline{M_0}$	I
40	34	32	$R7_3$	I	$\overline{M_1}$	I
41	35	33	$R8_0$	I/O		
42	36	34	$R8_1$	I/O		
43	37	35	$R8_2$	I/O		
44	38	36	$R8_3$	I/O		
45	39	37	$R9_0$	I/O		
46	40	38	$R9_1$	I/O		
47	41	39	$R9_2$	I/O		
48	42	40	$R9_3$	I/O		
49	43	41	RESET	I	RESET	I
50	44	42	$\overline{TEST}$	I	$\overline{TEST}$	I
51	45	43	OSC_1	I		
52	46	44	OSC_2	O		
53	47	45	GND	I/O	GND	
54	48	46	D_0	I/O	O_0	I/O
55	49	47	D_1	I/O	O_1	I/O
56	50	48	D_2	I/O	O_2	I/O
57	51	49	D_3	I/O	O_3	I/O
58	52	50	D_4	I/O	O_4	I/O
59	53	51	D_5	I/O	O_5	I/O
60	54	52	D_6	I/O	O_6	I/O
61	55	53	D_7	I/O	O_7	I/O
62	56	54	D_8	I/O		
63	57	55	D_9	I/O		
64	58	56	D_{10}	I/O	V_{CC}	

Notes: I/O: Input/output pins
I: Input pins
O: Output pins

Programming the Built-In PROM

The MCU's on-chip PROM is programmed in PROM mode. This PROM mode is set by pulling $\overline{TEST}$, $\overline{M_0}$, and $\overline{M_1}$ low, and RESET high as shown in figure 28. In PROM mode, the MCU does not operate. It can be programmed like a standard 27256 EPROM using a standard PROM programmer and a 64-to-28-pin socket adapter. Table 26 lists the recommended PROM programmers and socket adapters.

Since an instruction of the HMCS400 series consists of 10 bits, the HMCS400 series MCU incorporates a conversion circuit to enable the use of a general-purpose PROM programmer. By this circuit, an instruction is read or programmed using 2 addresses, the lower 5 bits and upper 5 bits. For example, if 8 kwords of on-chip PROM are programmed by a general-purpose PROM programmer, 16 kbytes of addresses ($0000 to $3FFF) should be specified.

Programming and Verification: The MCU can be high-speed programmed without causing voltage stress or affecting data reliability. Table 25 shows how programming and verification modes are selected.

Erasing: The PROMs with ceramic window packages can be erased by ultraviolet light. All erased bits become 1s.

The conditions for erasure are to use an ultraviolet light with a wavelength of 2537 Å with a minimum irradiation of 15 W·s/cm^2 exposing the LSI to a 12,000-µW/cm^2 UV source for 15 to 20 minutes at a distance of 1 inch.

Precautions

1. Addresses $0000 to $3FFF must be specified if the PROM is programmed by a PROM programmer. If addresses of $4000 or higher are accessed, the PROM may not be programmed or verified. The plastic package type LSIs cannot be erased and reprogrammed. Only the ceramic window packages can be erased and reprogrammed. Data in unused addresses must be set to $FF.

2. Be sure that the PROM programmer, socket adapter, and LSI line up (pin 1 positions match). Using the wrong programmer or socket adapter may cause an overvoltage and damage the LSI (table 26). Make sure that the LSI is firmly fixed in the socket adapter, and that the socket adapter is firmly fixed onto the programmer.

3. The PROM should be programmed with V_{PP} = 12.5 V. Other PROMs use 21 V. If 21 V is applied to the MCU, the LSI may be permanently damaged. A voltage of 12.5 V is Intel's 27256 V_{PP}.

Table 25 PROM Modes Selection

	Pin			
Mode	$\overline{CE}$	$\overline{OE}$	V_{PP}	O_0 to O_7
Programming	Low	High	V_{PP}	Data input
Verify	High	Low	V_{PP}	Data output
Programming inhibited	High	High	V_{PP}	High impedance

Table 26 PROM Programmers and Socket Adapters

PROM Programmer		Socket Adapter		
Manufacturer	**Type Name**	**Manufacturer**	**Type Name**	**Package**
DATA I/O	22B 29B	Hitachi	HS448ESS11H	DP-64S DC-64S
			HS440ESF01H	FP-64
			HS440ESF03H	FP-64A
AVAL Corp.	PKW-1000	Hitachi	HS448ESS21H	DP-64S DC-64S
			HS440ESF01H	FP-64
			HS440ESF03H	FP-64A

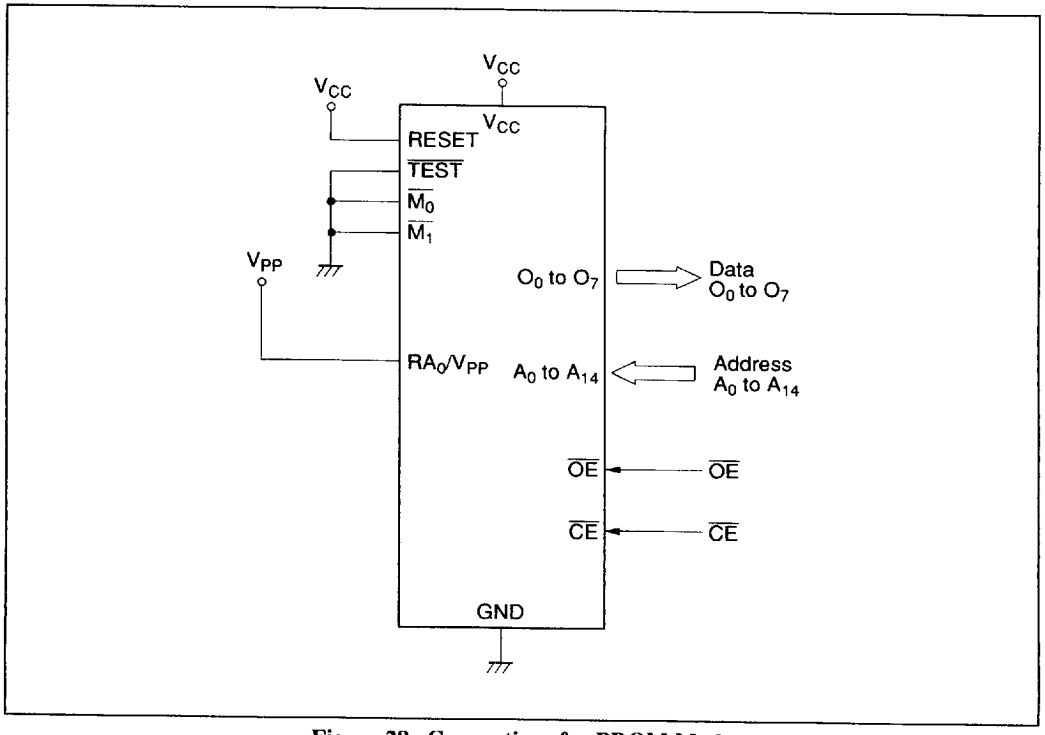


Figure 28 Connections for PROM Mode

Addressing Modes

RAM Addressing Modes

As shown in figure 29, the MCU has three RAM addressing modes: register indirect addressing, direct addressing, and memory register addressing.

Register Indirect Addressing Mode: The contents of the W, X, and Y registers (10 bits in total) are used as the RAM address.

Direct Addressing Mode: A direct addressing instruction consists of two words, with the word (10 bits) following the opcode used as the RAM address.

Memory Register Addressing Mode: The memory registers (16 digits from $040 to $04F) are accessed by executing the LAMR and XMRA instructions.

ROM Addressing Modes and the P Instruction

The MCU has four ROM addressing modes as shown in figure 30.

Direct Addressing Mode: The program can branch to any address in ROM memory space by executing the JMPL, BRL, or CALL instruction. These instructions replace the 14 program counter bits (PC_{13} to PC_0) with 14-bit immediate data.

Current Page Addressing Mode: The MCU has 32 pages of ROM with 256 words per page. By executing the BR instruction, the program can branch to an address on the current page. This instruction replaces the low-order eight bits of the program counter (PC_7 to PC_0) with 8-bit immediate data.

When the BR instruction is on a page boundary (256n + 255) (figure 31), executing the BR instruction transfers the PC contents to the next page according to the hardware architecture. Consequently, the program branches to the next page when the BR instruction is used on a page boundary. The HMCS400-series cross macroassembler has an automatic paging facility for ROM pages.

Zero-Page Addressing Mode: By executing the CAL instruction, the program can branch to the zero-page subroutine area, which is located at $0000 to $003F. When the CAL instruction is executed, 6 bits of immediate data are placed in the low-order 6 bits of the program counter (PC_5 to PC_0) and 0s are placed in the high-order 8 bits (PC_{13} to PC_6).

Table Data Addressing Mode: By executing the TBR instruction, the program can branch to the address determined by the contents of the 4-bit immediate data, accumulator, and B register.

P Instruction: ROM data addressed by table data addressing can be referenced by the P instruction (figure 32). When bit 8 in the referred ROM data is 1, 8 bits of ROM data are written into the accumulator and B register. When bit 9 is 1, 8 bits of ROM data are written into the R1 and R2 port data registers. When both bits 8 and 9 are 1, ROM data are written into the accumulator and B register, and also to the R1 and R2 port data registers at the same time.

The P instruction has no effect on the program counter.

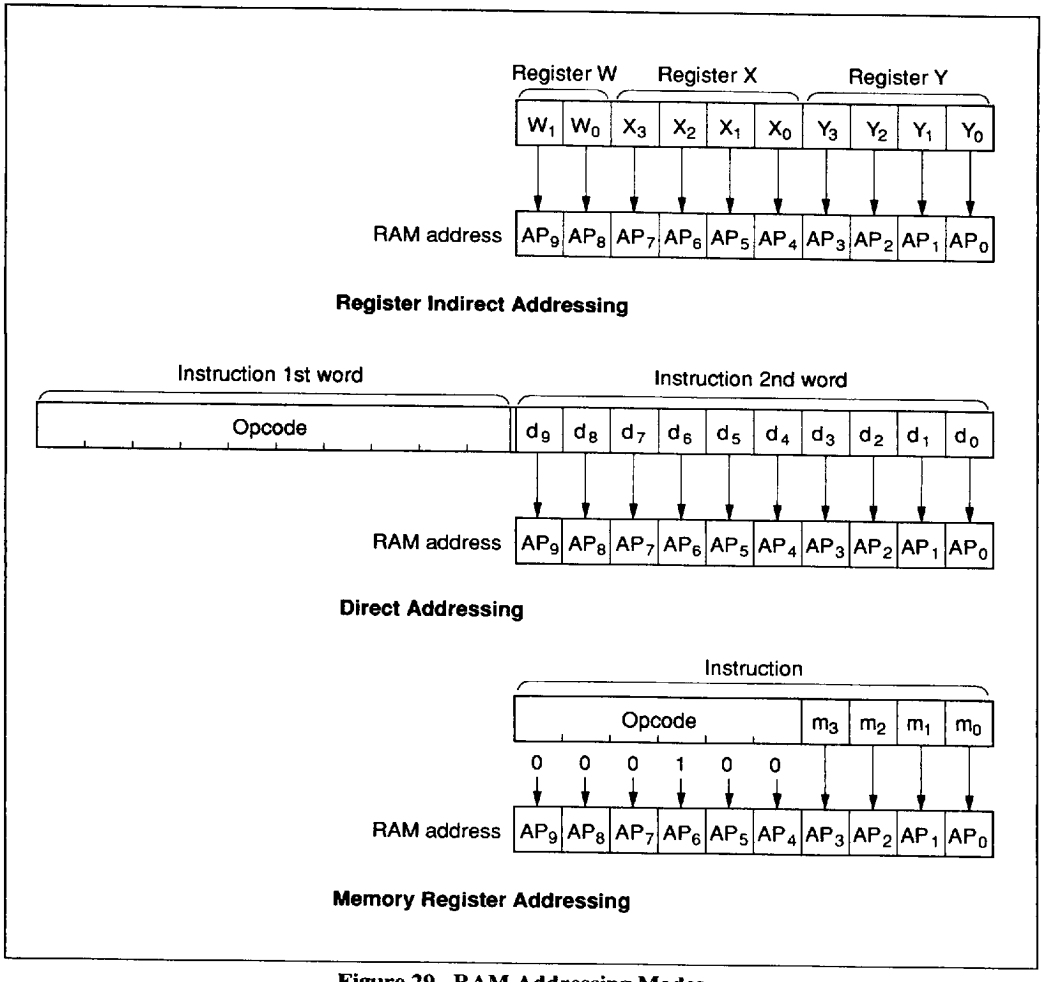


Figure 29 RAM Addressing Modes

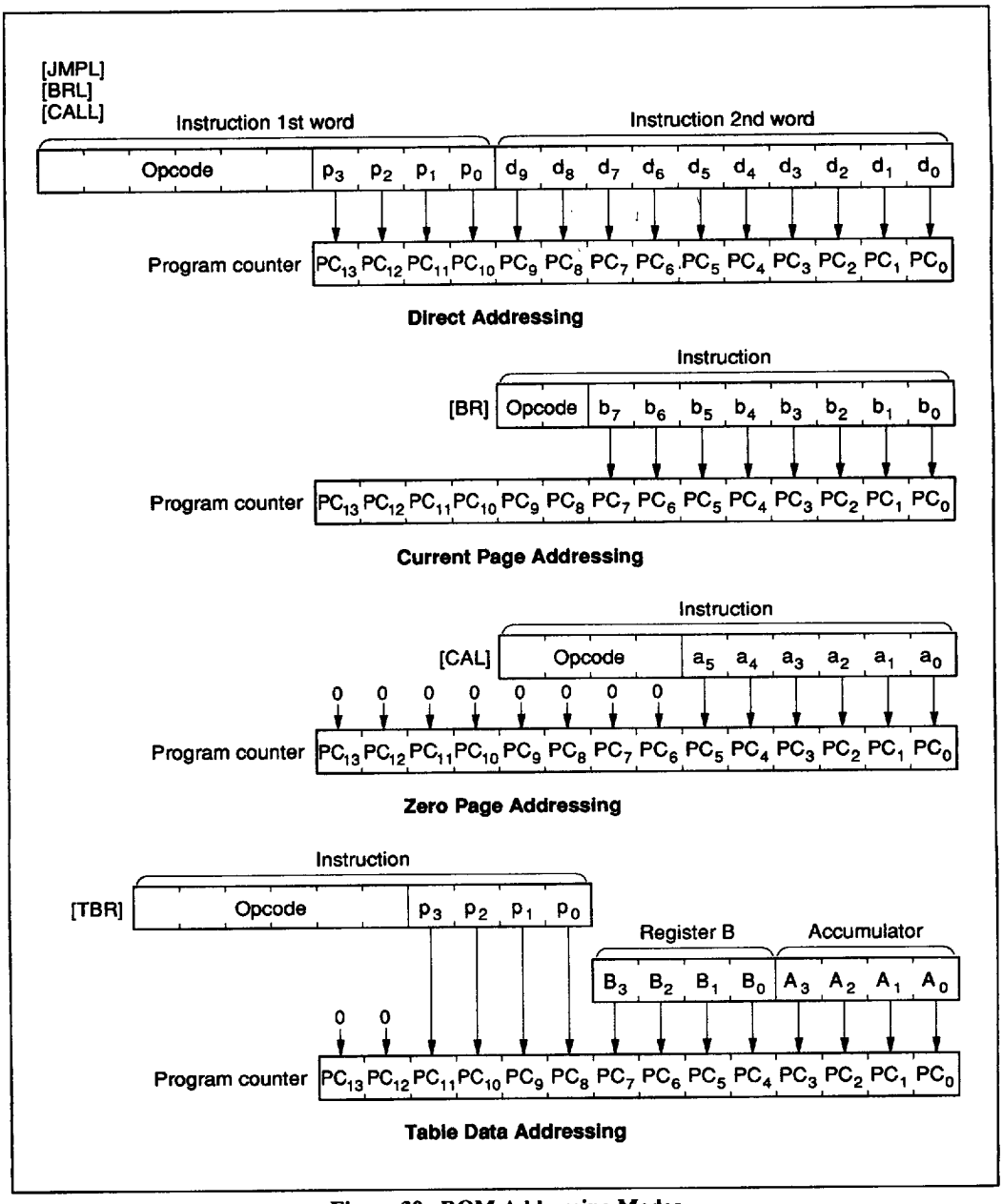


Figure 30 ROM Addressing Modes

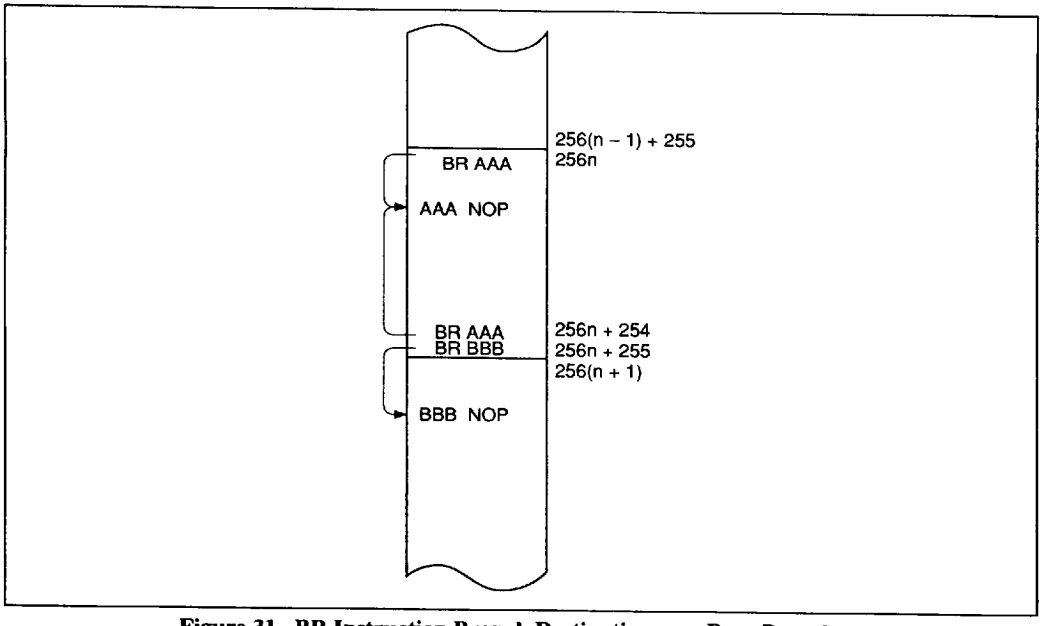


Figure 31 BR Instruction Branch Destination on a Page Boundary

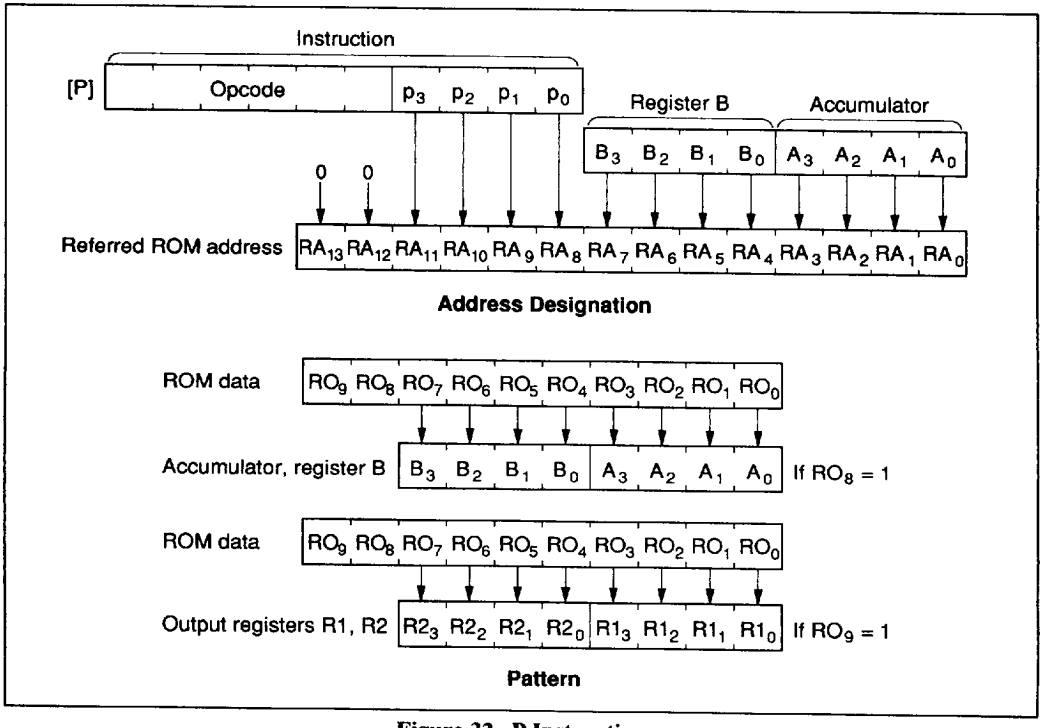


Figure 32 P Instruction

Absolute Maximum Ratings

Item	Symbol	Value	Unit	Notes
Supply voltage	V_{CC}	–0.3 to +7.0	V	
Programming voltage	V_{PP}	–0.3 to +14.0	V	2
Pin voltage	V_T	–0.3 to V_{CC} + 0.3	V	
		–0.3 to +15.0	V	3
Total permissible input current	ΣI_o	100	mA	4
Total permissible output current	$-\Sigma I_o$	50	mA	5
Maximum input current	I_o	4	mA	6, 7
		6	mA	6, 8
		30	mA	6, 9
Maximum output current	$-I_o$	4	mA	10, 11
Operating temperature	T_{opr}	–20 to +75	°C	
Storage temperature	T_{stg}	–55 to +125	°C	

Notes:
1. Permanent damage may occur if these absolute maximum ratings are exceeded. Normal operation should be under the conditions of the electrical characteristics tables. If these conditions are exceeded, it may cause the LSI to malfunction or affect the reliability of the LSI.
2. Applies to RA_0 (V_{PP}) of HD4074418 and HD4074408.
3. Applies to the intermediate-voltage pins (R8 and R9) of HD4074408.
4. Total permissible input current is the total sum of input currents which flow in from all I/O pins to GND simultaneously.
5. Total permissible output current is the total sum of the output currents which flow out from V_{CC} to all I/O pins simultaneously.
6. Maximum input current is the maximum amount of input current from each I/O pin to GND.
7. Applies to R0 to R6.
8. Applies to R8 to RA.
9. Applies to D_0 to D_{15}.
10. Maximum output current is the maximum amount of output current from V_{CC} to each I/O pin.
11. Applies to D_0 to D_{15}, R0 to R6, R8, and R9 of HD404418 and HD4074418, or D_0 to D_{15}, R0 to R6 of the HD4074408.

Electrical Characteristics

DC Characteristics (V_{CC} = 4.5 V to 5.5 V, GND = 0 V, T_a = –20°C to +75°C, unless otherwise specified)

Item	Symbol	Pin	Min	Typ	Max	Unit	Test Condition	Notes
Input high voltage	V_{IH}	RESET	$0.85V_{CC}$		V_{CC} + 0.3	V		
		$\overline{INT_0}$, $\overline{INT_1}$, INT_2, INT_3, $\overline{INT_4}$, INT_5, $\overline{SCK_1}$, $\overline{SCK_2}$	$0.8V_{CC}$		V_{CC} + 0.3	V		
		$\overline{TI_1}$, TI_2	$0.8V_{CC}$		V_{CC} + 0.3	V		
		SI_1, SI_2	$0.8V_{CC}$		V_{CC} + 0.3	V		
		OSC_1	V_{CC} – 0.5		V_{CC} + 0.3	V	External clock operation	
Input low voltage	V_{IL}	RESET	–0.3		$0.15V_{CC}$	V		
		$\overline{INT_0}$, $\overline{INT_1}$, INT_2, INT_3, $\overline{INT_4}$, INT_5, $\overline{SCK_1}$, $\overline{SCK_2}$	–0.3		$0.2V_{CC}$	V		
		$\overline{TI_1}$, TI_2	–0.3		$0.2V_{CC}$	V		
		SI_1, SI_2	–0.3		$0.2V_{CC}$	V		
		OSC_1	–0.3		0.5	V	External clock operation	
Output high voltage	V_{OH}	$\overline{SCK_1}$, SO_1, $\overline{SCK_2}$, SO_2, TO_1, TO_2, TO_3	V_{CC} – 1.0			V	$-I_{OH}$ = 1.0 mA	
			V_{CC} – 0.3			V	$-I_{OH}$ = 0.01 mA	
Output low voltage	V_{OL}	$\overline{SCK_1}$, SO_1, $\overline{SCK_2}$, SO_2, TO_1, TO_2, TO_3			0.4	V	I_{OL} = 1.6 mA	
Input/output leakage current	$\|I_{IL}\|$	RESET, OSC_1, $\overline{INT_0}$, $\overline{INT_1}$, INT_2, INT_3, $\overline{INT_4}$, INT_5, $\overline{TI_1}$, TI_2, $\overline{SCK_1}$, SI_1, $\overline{SCK_2}$, SI_2			1	µA	V_{in} = 0 V to V_{CC}	1
Current dissipation in active mode	I_{CC}	V_{CC}		4	8	mA	V_{CC} = 5 V R7 digital input mode f_{OSC} = 8 MHz	2, 5
	I_{CMP}			7	12	mA	V_{CC} = 5 V R7 analog input mode f_{OSC} = 8 MHz	3
Current dissipation in standby mode	I_{SBY}	V_{CC}		2	4	mA	V_{CC} = 5 V, f_{OSC} = 8 MHz	4, 5
Current dissipation in stop mode	I_{STOP}	V_{CC}			10	µA	$V_{in(\overline{TEST})}$ = V_{CC} to V_{CC} – 0.3 V $V_{in(RESET)}$ = 0 to 0.3 V	

DC Characteristics (V_{CC} = 4.5 V to 5.5 V, GND = 0 V, T_a = –20°C to +75°C, unless otherwise specified) (cont)

Item	Symbol	Pin	Min	Typ	Max	Unit	Test Condition	Notes
Stop mode retaining voltage	V_{STOP}	V_{CC}	2			V		
Comparator input reference voltage scope	VC_{ref}	V_{ref}	0		V_{CC} – 1.2	V		

Notes: 1. Pull-up MOS current and output buffer current are excluded.

2. The MCU is in the reset state. The input/output current does not flow.

   Test conditions: MCU state:
   - Reset state in operation mode

   Pin state:
   - RESET, $\overline{TEST}$: V_{CC}

3. The current source when I/O current does not flow with all pins of R7 in the analog input mode.

   Test condition: V_{ref}, $R7_0$ to $R7_3$: GND

4. The timer operates with the fastest clock and input/output current does not flow.

   Test conditions: MCU state:
   - Standby mode
   - Input/output: Reset state
   - Timer: Divide-by-2 prescaler divide ratio
   - Serial interface: Stop

   Pin state:
   - RESET: GND
   - $\overline{TEST}$: V_{CC}
   - D_0 to D_{15}, R0 to RA: V_{CC}

5. When $f_{OSC} = \chi$ MHz, the current dissipation in operation mode and standby mode is estimated as follows:

   Maximum value ($f_{OSC} = \chi$ MHz) = $\frac{\chi}{8}$ × maximum value (f_{OSC} = 8 MHz)

   When f_{OSC} is less than 1 MHz, the current dissipation occasionally becomes greater than the estimate.

Input/Output Characteristics for Standard Pins (V_{CC} = 4.5 to 5.5 V, GND = 0 V, T_a = –20°C to +75°C, unless otherwise specified)

Item	Symbol	Pin	Min	Typ	Max	Unit	Test Condition	Notes
Input high voltage	V_{IH}	R0 to RA	$0.7V_{CC}$		V_{CC} + 0.3	V		1
Input low voltage	V_{IL}	R0 to RA	–0.3		$0.3V_{CC}$	V		1
Output high voltage	V_{OH}	R0 to R6, R8, R9	V_{CC} – 1.0			V	$-I_{OH}$ = 1.0 mA	1
		R0 to R6 R8, R9	V_{CC} – 0.3			V	$-I_{OH}$ = 0.01 mA	1
Output low voltage	V_{OL}	R0 to R6 R8 to RA			0.4	V	I_{OL} = 1.6 mA	1
Input/output leakage current	$\|I_{IL}\|$				1	µA	V_{in} = 0 V to V_{CC}	2, 3
					10	µA	V_{in} = 0 V to V_{CC}	2, 4
Pull-up MOS current	$-I_{PU}$	R0 to R6	15		120	µA	V_{CC} = 5 V, V_{in} = 0 V	5
Input high voltage	V_{IHA}	R7 (analog compare mode)	VC_{ref} + 0.1		V_{CC} – 1.2	V		
Input low voltage	V_{ILA}	R7 (analog compare mode)	0		VC_{ref} – 0.1	V		

Notes: 1. Does not apply to R8 and R9 of the HD4074408.
2. Pull-up MOS current and output buffer current are excluded.
3. Applies to R0 to RA of the HD404418, R0 to R9 and RA_1 of the HD4074418, and R0 to R7 and RA_1 of the HD4074408.
4. Applies to RA_0 of the HD4074418 and HD4074408.
5. Applies to HD404418 pins which are specified as with pull-up MOS via the mask option.

Input/Output Characteristics for High-Current Pins (V_{CC} = 4.5 to 5.5 V, GND = 0 V, T_a = –20°C to +75°C, unless otherwise specified)

Item	Symbol	Pin	Min	Typ	Max	Unit	Test Condition	Note
Input high voltage	V_{IH}	D_0 to D_{15}	$0.7V_{CC}$		V_{CC} + 0.3	V		
Input low voltage	V_{IL}	D_0 to D_{15}	–0.3		$0.3V_{CC}$	V		
Output high voltage	V_{OH}	D_0 to D_{15}	V_{CC} – 1.0			V	$-I_{OH}$ = 1.0 mA	
		D_0 to D_{15}	V_{CC} – 0.3			V	$-I_{OH}$ = 0.01 mA	
Output low voltage	V_{OL}	D_0 to D_{15}			1.5	V	I_{OL} = 10 mA	
Input/output leakage current	$\|I_{IL}\|$	D_0 to D_{15}			1	µA	V_{in} = 0 V – V_{CC}	1
Pull up MOS current	$-I_{PU}$	D_0 to D_{15}	15		120	µA	V_{CC} = 5 V, V_{in} = 0 V	2

Notes: 1. Pull-up MOS and output buffer current are excluded.
2. Applies to HD404418 pins which are specified as with pull-up MOS via mask option.

Input/Output Characteristics for Intermediate-Voltage Pins (V_{CC} = 4.5 to 5.5 V, GND = 0 V, T_a = –20°C to +75°C, unless otherwise specified)

Item	Symbol	Pin	Min	Typ	Max	Unit	Test Condition	Note
Input high voltage	V_{IH}	R8, R9	$0.7V_{CC}$		12.8	V		1
Input low voltage	V_{IL}	R8, R9	–0.3		$0.3V_{CC}$	V		1
Output high voltage	V_{OH}	R8, R9	11			V	7.5 kΩ at 12 V	1
Output low voltage	V_{OL}	R8, R9			0.4	V	I_{OL} = 1.6 mA	1
Input/output leakage current	$\|I_{IL}\|$	R8, R9			10	µA	V_{in} = 0 V – 12.8 V	1

Note: 1. Applies to the HD4074408.

AC Characteristics (V_{CC} = 4.5 to 5.5 V, GND = 0 V, T_a = –20°C to +75°C, unless otherwise specified)

Item		Symbol	Pin	Min	Typ	Max	Unit	Test Condition	Note
Crystal or ceramic oscillator	Oscillation frequency	f_{OSC}	OSC_1, OSC_2	0.4	8	9.0	MHz		
	Instruction cycle time	t_{cyc}		0.89	1	20	µs		
	Oscillator stabilization time	t_{RC}	OSC_1, OSC_2			20	ms		1
External clock	External clock frequency	f_{CP}	OSC_1	0.4		9.0	MHz		2
	External clock high width	t_{CPH}	OSC_1	41			ns		2
	External clock low width	t_{CPL}	OSC_1	41			ns		2
	External clock rise time	t_{CPr}	OSC_1			20	ns		2
	External clock fall time	t_{CPf}	OSC_1			20	ns		2
	Instruction cycle time	t_{cyc}		0.89		20	µs		2
External interrupt signal high width		t_{IH}	$\overline{INT_0}$, $\overline{INT_1}$, $\overline{INT_2}$, INT_3, $\overline{INT_4}$, INT_5	2			t_{cyc}		3
External interrupt signal low width		t_{IL}	$\overline{INT_0}$, $\overline{INT_1}$, $\overline{INT_2}$, INT_3, $\overline{INT_4}$, INT_5	2			t_{cyc}		3
Timer input high width		t_{TH}	$\overline{TI_1}$, TI_2	2			t_{cyc}		3
Timer input low width		t_{TL}	$\overline{TI_1}$, TI_2	2			t_{cyc}		3
RESET high width		t_{RSTH}	RESET	3			t_{cyc}		4
Input capacitance		C_{in}				70	pF	f = 1 MHz, V_{in} = 0 V	5
						15	pF	f = 1 MHz, V_{in} = 0 V	6
RESET fall time		t_{RSTf}				20	ms		4
Analog comparator stabilization time		t_{CSTB}	R7 (in analog input mode)			2	t_{cyc}		7

Notes:
1. The oscillator stabilization time is the period after V_{CC} reaches 4.5 V at power-on until the oscillator stabilizes, or after the RESET input goes high to cancel the stop mode. When using a crystal or ceramic oscillator, consult with the manufacturer to determine what stabilization time is required, since it will depend on the circuit constants and stray capacitance.
2. See figure 33.
3. See figure 34.
4. See figure 35.
5. Applies to RA_0 of HD4074418 and HD4074408.
6. Applies to all input pins of HD404418 and input pins except for RA_0 of HD4074418 and HD4074408.
7. Analog comparator stabilization time is the period after R7 goes into the analog input mode until the analog comparator stabilizes to read valid data.

Serial Interface Timing Characteristics (V_{CC} = 5 V ±10%, GND = 0 V, T_a = –20°C to +75°C, unless otherwise specified)

During Transmit Clock Output

Item	Symbol	Pin	Min	Typ	Max	Unit	Test Condition	Note
Transmit clock cycle time	t_{Scyc}	$\overline{SCK_1}$, $\overline{SCK_2}$	1			t_{cyc}	Refer to figure 37	1
Transmit clock high width	t_{SCKH}	$\overline{SCK_1}$, $\overline{SCK_2}$	0.5			t_{Scyc}	Refer to figure 37	1
Transmit clock low width	t_{SCKL}	$\overline{SCK_1}$, $\overline{SCK_2}$	0.5			t_{Scyc}	Refer to figure 37	1
Transmit clock rise time	t_{SCKr}	$\overline{SCK_1}$, $\overline{SCK_2}$			100	ns	Refer to figure 37	1
Transmit clock fall time	t_{SCKf}	$\overline{SCK_1}$, $\overline{SCK_2}$			100	ns	Refer to figure 37	1
Serial output data delay time	t_{DSO}	SO_1, SO_2			250	ns	Refer to figure 37	1
Serial input data setup time	t_{SSI}	SI_1, SI_2	300			ns		1
Serial input data hold time	t_{HSI}	SI_1, SI_2	150			ns		1

During Transmit Clock Input

Item	Symbol	Pin	Min	Typ	Max	Unit	Test Condition	Note
Transmit clock cycle time	t_{Scyc}	$\overline{SCK_1}$, $\overline{SCK_2}$	1			t_{cyc}		1
Transmit clock high width	t_{SCKH}	$\overline{SCK_1}$, $\overline{SCK_2}$	0.5			t_{Scyc}		1
Transmit clock low width	t_{SCKL}	$\overline{SCK_1}$, $\overline{SCK_2}$	0.5			t_{Scyc}		1
Transmit clock rise time	t_{SCKr}	$\overline{SCK_1}$, $\overline{SCK_2}$			100	ns		1
Transmit clock fall time	t_{SCKf}	$\overline{SCK_1}$, $\overline{SCK_2}$			100	ns		1
Serial output data delay time	t_{DSO}	SO_1, SO_2			250	ns	Refer to figure 37	1
Serial input data setup time	t_{SSI}	SI_1, SI_2	300			ns		1
Serial input data hold time	t_{HSI}	SI_1, SI_2	150			ns		1

Note: 1. See figure 36.

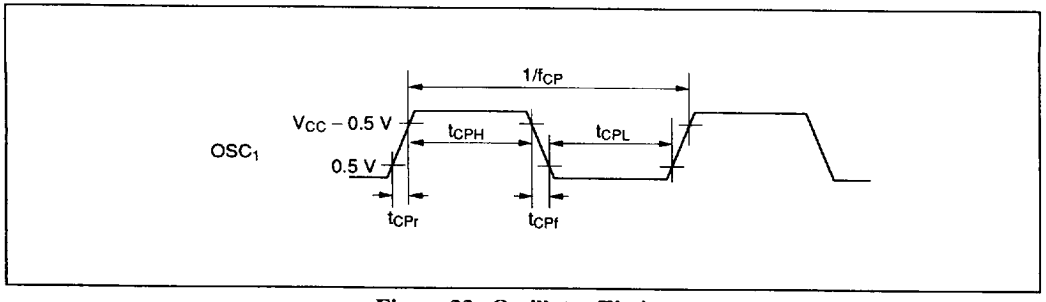


Figure 33 Oscillator Timing

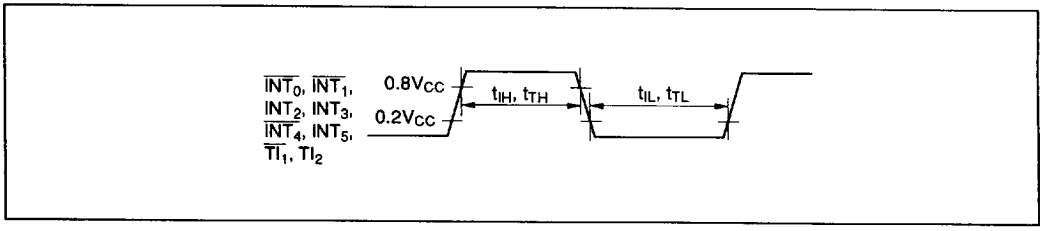


Figure 34 Interrupt and Timer Input Timing

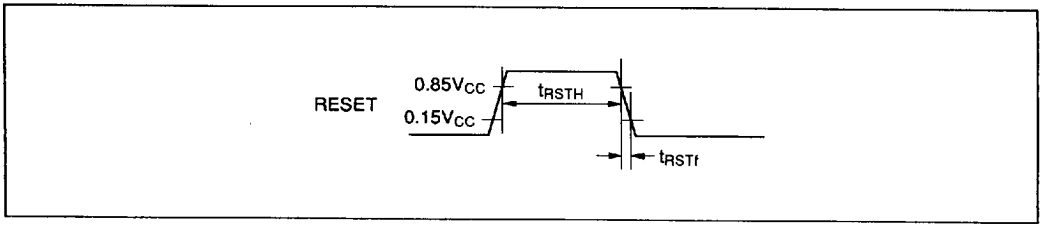
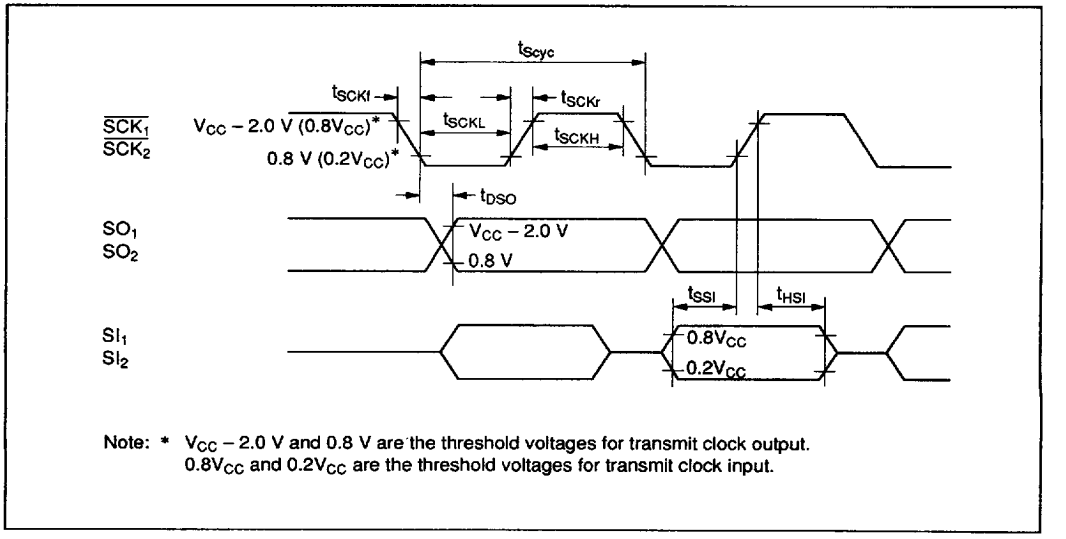


Figure 35 Reset Timing



Note: * V_{CC} – 2.0 V and 0.8 V are the threshold voltages for transmit clock output.
$0.8V_{CC}$ and $0.2V_{CC}$ are the threshold voltages for transmit clock input.

Figure 36 Timing of Serial Interface

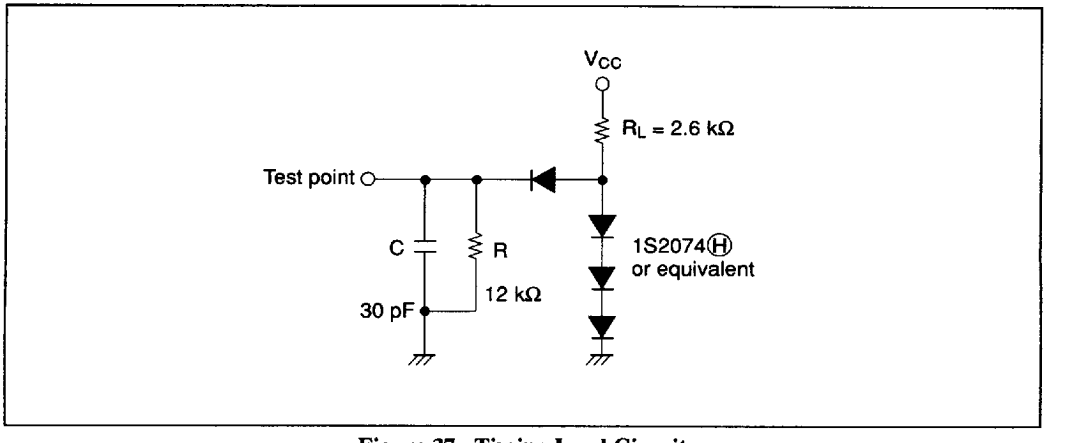


Figure 37 Timing Load Circuit

HD404418 Option List

Please check off the appropriate applications and enter the necessary information.

Date of order	
Customer	
Department	
Name	
ROM code name	
LSI type number (Hitachi's entry)	

1. I/O Options

Note: Shaded options are not available

Pin name		I/O		I/O option A	I/O option B	I/O option C
D0		High current pins	I/O			
D1			I/O			
D2			I/O			
D3			I/O			
D4			I/O			
D5			I/O			
D6			I/O			
D7			I/O			
D8			I/O			
D9			I/O			
D10			I/O			
D11			I/O			
D12			I/O			
D13			I/O			
D14			I/O			
D15			I/O			
R0	R00	Standard pins	I/O			
	R01		I/O			
	R02		I/O			
	R03		I/O			
R1	R10		I/O			
	R11		I/O			
	R12		I/O			
	R13		I/O			
R2	R20		I/O			
	R21		I/O			
	R22		I/O			
	R23		I/O			

Pin name		I/O		I/O option A	I/O option B	I/O option C
R3	R30	Standard pins	I/O			
	R31		I/O			
	R32		I/O			
	R33		I/O			
R4	R40		I/O			
	R41		I/O			
	R42		I/O			
	R43		I/O			
R5	R50		I/O			
	R51		I/O			
	R52		I/O			
	R53		I/O			
R6	R60		I/O			
	R61		I/O			
	R62		I/O			
	R63		I/O			
R7	R70		I	○		
	R71		I	○		
	R72		I	○		
	R73		I	○		
R8	R80		I/O	○		
	R81		I/O	○		
	R82		I/O	○		
	R83		I/O	○		
R9	R90		I/O	○		
	R91		I/O	○		
	R92		I/O	○		
	R93		I/O	○		
RA	RA0		I/O			○
	RA1		I/O			○

A: Without pull-up MOS (CMOS), B: With pull-up MOS (CMOS),
C: NMOS open drain

2. Timer 4

☐ Free-running timer
☐ Watchdog timer

3. ROM Code Media
Please specify the first type listed below (the upper bits and lower bits are mixed together) when using the EPROM on-package microcomputer type (including ZTAT™ version).

☐ EPROM: The upper bits and lower bits are mixed together. The upper five bits and lower five bits are programmed to the same EPROM in alternating order (i.e., LULULU...).
☐ EPROM: The upper bits and lower bits are separated. The upper five bits and lower five bits are programmed to different EPROMS.

4. System Oscillator (OSC1 and OSC2)

☐ Ceramic oscillator	f =	MHz
☐ Crystal oscillator	f =	MHz
☐ External clock	f =	MHz

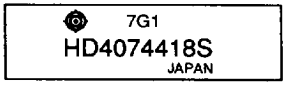
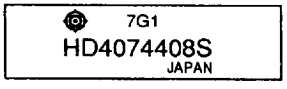
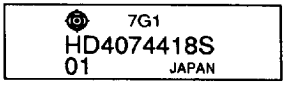
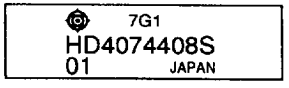
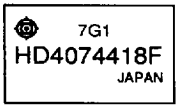
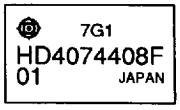
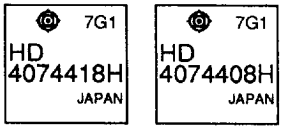
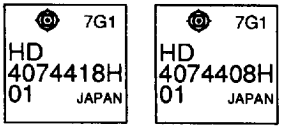
5. Stop Mode

☐ Used
☐ Not used

6. Package

☐ DP-64S
☐ FP-64
☐ FP-64A

HD4074418/HD4074408 Type Name and Marking

Package		Timer Option: Free-Running Timer	Timer Option: Watchdog Timer
DC-64S	Type name	HD4074418C HD4074408C	HD4074418C01 HD4074408C01
	Marking	 	 
DP-64S	Type name	HD4074418S HD4074408S	HD4074418S01 HD4074408S01
	Marking	 	 
FP-64	Type name	HD4074418F HD4074408F	HD4074418F01 HD4074408F01
	Marking	 	 
GP-64A	Type name	HD4074418H HD4074408H	HD4074418H01 HD4074408H01
	Marking		

Note: The markings are of standard specification.