HD614P080S/HD614P0160S

DESCRIPTION

The HD614P080S and HD614P0160S are 4-bit single chip microcomputers which have mounted a standard EPROM 2764/27128/27256 for program memory.

The HD614P080S and HD614P0160S are pin-compatible with the mask ROM type HMCS402/404/408, but have some differences with them as shown in Table 33. By modifying the program in the EPROM, they can be used for the evaluation of the HMCS402/404/408, or for small-scale production.

■ HARDWARE FEATURES

- 4-bit Architecture
- Application to 4k, 8k or 16k words × 10 bits of EPROM
  - 4096 words × 10 bits HN482764, HN27C64
  - 8192 words × 10 bits HN4827128
  - 16384 words × 10 bits HN27C256
- Data Memory (RAM) Capacity . . . . 576 digits × 4 bits
- 58 I/O Pins — 26 I/O pins are high voltage up to 40V (max).
- 2 Timer/Counters
  - 11-bit Prescaler
  - 8-bit Free Running Counter
  - 8-bit Auto-reload Timer/Event Counter
- Clocked Synchronous 8-bit Serial Interface
- 5 Interrupts
  - External 2
  - Timer/Counter 2
  - Serial Interface 1
- Subroutine Stack
  - Up to 16 levels including interrupts
- Minimum Instruction Execution Time; 1.33 μs
- 2 Low Power Modes
  - Standby — Stops instruction execution while keeping clock generator and interrupt functions included Timer/Counter and Serial Interface in operation
  - Stop — Stops instruction execution and clock generation while retaining RAM data
- Clock Generator
  - External Connection of Crystal Resonator or Ceramic Filter Resonator (externally drivable)
- Power Voltage Range; 5V ± 10%
- I/O Pin Circuit Form
  - All standard pins are "without pull-up MOS".
  - All high voltage pins are "without pull-down MOS".
- Shrink Type 64 Pin EPROM On-package

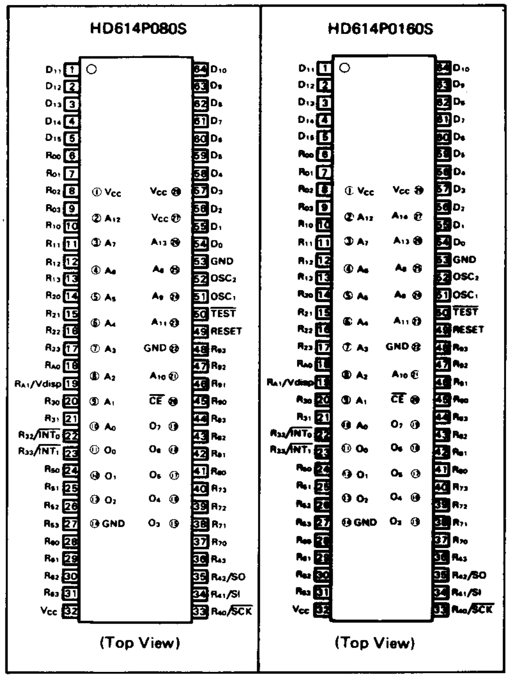
■ SOFTWARE FEATURES

- Software Compatible with HMCS402/404/408
- Instruction Set Similar to and More Powerful than HMCS40 Series; 99 Instructions
- High Programming Efficiency with 10-bit ROM/Word; 79 instructions are single word instructions.
- Direct Branch to All ROM Area
- Direct or Indirect Addressing to All RAM Area
- Subroutine Nesting Up to 16 Levels Including Interrupts
- Binary and BCD Arithmetic Operation
- Powerful Logic Arithmetic Operation
- Pattern Generation — Table Look Up Capability —
- Bit Manipulation for Both RAM and I/O

■ PROGRAM DEVELOPMENT SUPPORT TOOLS

- Cross assembler and simulator software for use with IBM PCs and compatibles
- In circuit emulator for use with IBM PC

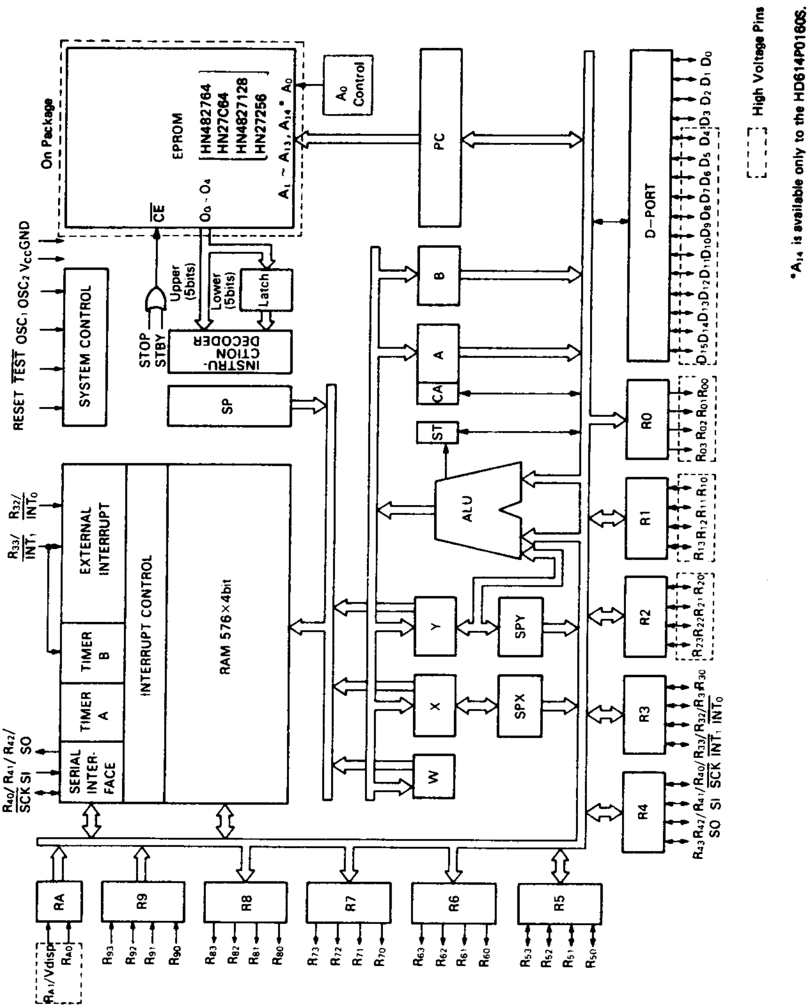
■ PIN ARRANGEMENT



■ RECOMMENDED APPLICABLE EPROM

Type No.	Program Memory Capacity	f_{osc} (MHz)	EPROM Type No.
HD614P080S	4096 words	4	HN27C64-30 HN482764-3
		6	HN27C64-25 HN482764
	8192 words	4	HN4827128-45
		6	HN4827128-25
HD614P0160S	16384 words	4	HN27C256-30
		6	HN27C256-25

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit	Note
Supply Voltage	V_{CC}	–0.3 to +7.0	V	
Pin Voltage	V_T	–0.3 to V_{CC} +0.3	V	3
		V_{CC} –45 to V_{CC} +0.3	V	4
Total Allowance of Input Currents	ΣI_O	50	mA	5
Total Allowance of Output Currents	$-\Sigma I_O$	150	mA	6
Maximum Input Current	I_O	15	mA	7, 8
Maximum Output Current	$-I_O$	4	mA	9, 10
		6	mA	9, 11
		30	mA	9, 12
Operating Temperature	T_{opr}	–20 to +75	°C	
Storage Temperature	T_{stg}	–55 to +125	°C	

(Note 1) Permanent damage may occur if "Absolute Maximum Ratings" of the LSI or the EPROM are exceeded. Normal operation should be under the conditions of "Electrical Characteristics". If these conditions are exceeded, it may cause the malfunction and affect the reliability of LSI.
(Note 2) All voltages are with respect to GND.
(Note 3) Applied to standard pins.
(Note 4) Applied to high voltage I/O pins.
(Note 5) Total allowance of input current is the total sum of input current which flow in from all I/O pins to GND simultaneously.
(Note 6) Total allowance of output current is the total sum of the output current which flow out from V_{CC} to all I/O pins simultaneously.
(Note 7) Maximum input current is the maximum amount of input current from each I/O pin to GND.
(Note 8) Applied to $D_0 \sim D_3$ and R3 ~ R8.
(Note 9) Maximum output current is the maximum amount of output current from V_{CC} to each I/O pin.
(Note 10) Applied to $D_0 \sim D_3$ and R3 ~ R8.
(Note 11) Applied to R0 ~ R2.
(Note 12) Applied to $D_4 \sim D_{15}$.

■ **ELECTRICAL CHARACTERISTICS**

● **DC CHARACTERISTICS (V_{CC} = 4.5V to 5.5V, GND = 0V, Ta = -20 to +75°C, if not specified.)**

Item	Symbol	Pin Name	Test Conditions		Value min	Value typ	Value max	Unit	Note
Input "High" Voltage	V_{IH}	RESET, $\overline{SCK}$, $\overline{INT_0}$, $\overline{INT_1}$			$0.7V_{CC}$	–	$V_{CC}+0.3$	V	
		SI			$0.7V_{CC}$	–	$V_{CC}+0.3$	V	
		OSC_1			$V_{CC}-0.5$	–	$V_{CC}+0.3$	V	
Input "Low" Voltage	V_{IL}	RESET, $\overline{SCK}$, $\overline{INT_0}$, $\overline{INT_1}$			-0.3	–	$0.22V_{CC}$	V	
		SI			-0.3	–	$0.22V_{CC}$	V	
		OSC_1			-0.3	–	0.5	V	
Output "High" Voltage	V_{OH}	$\overline{SCK}$, SO	$-I_{OH}$ = 1.0 mA		$V_{CC}-1.0$	–	–	V	
			$-I_{OH}$ = 0.01 mA		$V_{CC}-0.3$	–	–	V	
Output "Low" Voltage	V_{OL}	$\overline{SCK}$, SO	I_{OL} = 1.6 mA		–	–	0.4	V	
Input/Output Leakage Current	$\|I_{IL}\|$	RESET, $\overline{SCK}$, $\overline{INT_0}$, $\overline{INT_1}$, SI, SO, OSC_1	V_{in} = 0V to V_{CC}		–	–	1	μA	1
Current Dissipation in Operation Mode	I_{CC}	V_{CC}	V_{CC} = 5V	Crystal or Ceramic Filter Resonator f_{osc} = 4MHz	–	–	2.0	mA	2, 5
Current Dissipation in Standby Mode	I_{SBY1}	V_{CC}	Maximum Logic Operation V_{CC} = 5V	Crystal or Ceramic Filter Resonator f_{osc} = 4MHz	–	–	1.2	mA	3, 5
	I_{SBY2}	V_{CC}	Minimum Logic Operation V_{CC} = 5V	Crystal or Ceramic Filter Resonator f_{osc} = 4MHz	–	–	0.9	mA	4, 5
Current Dissipation in Stop Mode	I_{stop}	V_{CC}	V_{in} (TEST) = V_{CC} ~ $V_{CC}-0.3V$, V_{in} (RESET) = 0 ~ 0.3V		–	–	10	μA	
Stop Mode Retain Voltage	V_{stop}	V_{CC}			2.0	–	–	V	

(Note 1) Output buffer current are excluded.

(Note 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} voltage
- D_0 ~ D_3, R3 ~ R9 – V_{CC} voltage
- D_4 ~ D_{15}, R0 ~ R2, R_{A0}, R_{A1} – V_{CC} ~ V_{CC}-40V

(Note 3) The timer/counter with the fastest clock and input/output current does not flow.
Test Conditions: MCU state;
- Standby Mode
- Input/Output; Reset state
- TIMER-A; ÷2 prescaler divide ratio
- TIMER-B; ÷2 prescaler divide ratio
- SERIAL; Stop

Pin state;
- RESET – GND voltage
- $\overline{TEST}$ – V_{CC} voltage
- D_0 ~ D_3, R3 ~ R9 – V_{CC} voltage
- D_4 ~ D_{15}, R0 ~ R2, R_{A0}, R_{A1} – V_{CC} ~ V_{CC}-40V

(Note 4) The timer/counter with the slowest clock and input/output current does not flow.
Test Conditions: MCU state;
- Standby Mode
- Input/Output; Reset state
- TIMER-A; ÷2048 prescaler divide ratio
- TIMER-B; ÷2048 prescaler divide ratio
- SERIAL; Stop

Pin state;
- $\overline{RESET}$ – GND voltage
- $\overline{TEST}$ – V_{CC} voltage
- $D_0 \sim D_3$, R3 ~ R9 – V_{CC} voltage
- $D_4 \sim D_{15}$, R0 ~ R2, R_{A0}, R_{A1} – V_{CC} ~ V_{CC}-40V

(Note 5) The consumption of current in operation and standby mode is proportional to f_{osc}. When $f_{osc} = x$ [MHz], the value of each current is calculated as follows.

$$\text{max. value } (f_{osc} = x) = \frac{x}{4} \times \text{max. value } (f_{osc} = 4 \text{ [MHz]}).$$

• INPUT/OUTPUT CHARACTERISTICS FOR STANDARD PIN

(V_{CC} = 4.5V to 5.5V, GND = 0V, Ta = -20 to +75°C, if not specified.)

Item	Symbol	Pin Name	Test Conditions	Value			Unit	Note
				min	typ	max		
Input "High" Voltage	V_{IH}	$D_0 \sim D_3$, R3 ~ R5, R9		$0.7V_{CC}$	–	V_{CC}+0.3	V	
Input "Low" Voltage	V_{IL}	$D_0 \sim D_3$, R3 ~ R5, R9		–0.3	–	$0.22V_{CC}$	V	
Output "Low" Voltage	V_{OL}	$D_0 \sim D_3$, R3 ~ R8	I_{OL} = 1.6 mA	–	–	0.4	V	
Input/Output Leakage Current	$\lvert I_{IL} \rvert$	$D_0 \sim D_3$, R3 ~ R9	V_{in} = 0V–V_{CC}	–	–	1	μA	1

(Note 1) Output buffer current are excluded.

• INPUT/OUTPUT CHARACTERISTICS FOR HIGH VOLTAGE PIN

(V_{CC} = 4.5V to 5.5V, GND = 0V, Ta = -20 to +75°C, if not specified.)

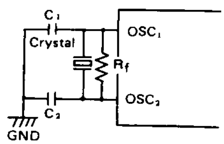
Item	Symbol	Pin Name	Test Conditions	Value			Unit	Note
				min	typ	max		
Input "High" Voltage	V_{IH}	$D_4 \sim D_{15}$, R1 R2, R_{A0}, R_{A1}		$0.7V_{CC}$	–	V_{CC}+0.3	V	
Input "Low" Voltage	V_{IL}	$D_4 \sim D_{15}$, R1 R2, R_{A0}, R_{A1}		V_{CC}-40	–	$0.22V_{CC}$	V	
Output "High" Voltage	V_{OH}	$D_4 \sim D_{15}$	$-I_{OH}$ = 15mA	V_{CC}-3.0	–	–	V	
			$-I_{OH}$ = 9mA	V_{CC}-2.0				
		R0 ~ R2	$-I_{OH}$ = 3mA	V_{CC}-3.0	–	–	V	
			$-I_{OH}$ = 1.8 mA	V_{CC}-2.0				
Output "Low" Voltage	V_{OL}	$D_4 \sim D_{15}$ R0 ~ R2	150kΩ to V_{CC}-40V	–	–	V_{CC}-37	V	
Input/Output Leakage Current	$\lvert I_{IL} \rvert$	$D_4 \sim D_{15}$ R0 ~ R2 R_{A0}, R_{A1}	V_{in} = V_{CC}-40V to V_{CC}	–	–	20	μA	1

(Note 1) Output buffer current are excluded.

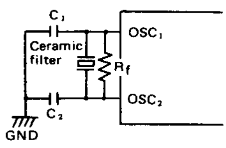
- **AC CHARACTERISTICS** (V_{CC} = 4.5V to 5.5V, GND = 0V, Ta= -20 to +75°C, if not specified.)

Item		Symbol	Pin Name	Test Conditions	Value min	Value typ	Value max	Unit	Note
Crystal Resonator	Oscillation Frequency	f_{osc}	OSC_1, OSC_2		0.4	–	6.2	MHz	
	Instruction Cycle Time	t_{cyc}			1.29	–	20	μs	
	Oscillator Stabilization Time	t_{RC}	OSC_1, OSC_2		–	–	20	ms	1
Ceramic Filter Resonator	Oscillation Frequency	f_{osc}	OSC_1, OSC_2		0.4	–	6.2	MHz	
	Instruction Cycle Time	t_{cyc}			1.29	–	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	–	6.2	MHz	2
	External Clock "High" Level Width	t_{CPH}	OSC_1		70	–	–	ns	2
	External Clock "Low" Level Width	t_{CPL}	OSC_1		70	–	–	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}			1.29	–	20	μs	2
$\overline{INT_0}$ "High" Level Width		t_{I0H}	$\overline{INT_0}$		2	–	–	t_{cyc}	3
$\overline{INT_0}$ "Low" Level Width		t_{I0L}	$\overline{INT_0}$		2	–	–	t_{cyc}	3
$\overline{INT_1}$ "High" Level Width		t_{I1H}	$\overline{INT_1}$		2	–	–	t_{cyc}	3
$\overline{INT_1}$ "Low" Level Width		t_{I1L}	$\overline{INT_1}$		2	–	–	t_{cyc}	3
RESET "High" Level Width		t_{RSTH}	RESET		2	–	–	t_{cyc}	4
Input Capacitance		C_{in}	all pins	f=1MHz V_{in} = 0V	–	–	15	pF	
Reset Fall Time		t_{RSTf}			–	–	20	ms	4

(Note 1) Oscillator stabilization time is the time until the oscillator stabilizes after V_{CC} reaches its minimum allowable voltage V_{CC} = 4.5V after power-on, or after RESET goes high. At power-on or STOP mode release, RESET must be kept high for at least t_{RC}. Since t_{RC} depends on the crystal or ceramic filter's circuit constant and stray capacitance, please get the manufacturer's advice when designing the RESET circuit.

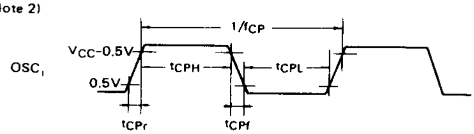


Crystal: 6.0 [MHz]
NC-18C (Nihon Denpa Kogyo)
R_f = 1 [MΩ] ± 2%, C_1 = C_2 = 20 [pF] ± 20%

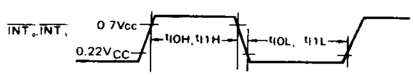


Ceramic filter: CSA6.00 MG (Murata)
R_f = 1 [MΩ] ± 2%, C_1 = C_2 = 30 [pF] ± 20%

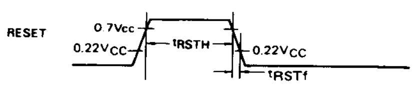
(Note 2)



(Note 3)



(Note 4)



- **SERIAL INTERFACE TIMING CHARACTERISTICS**
  (V_{CC} = 4.5V to 5.5V, GND = 0V, Ta = −20 to +75°C, if not specified.)

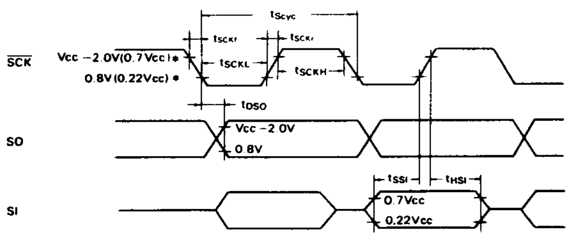
• At Transfer Clock Output

Item	Symbol	Pin Name	Test Conditions	Value			Unit	Note
				min	typ	max		
Transfer Clock Cycle Time	t_{Scyc}	$\overline{SCK}$	(Note 2)	1	–	–	t_{cyc}	1, 2
Transfer Clock "High" Level Width	t_{SCKH}	$\overline{SCK}$	(Note 2)	0.5	–	–	t_{Scyc}	1, 2
Transfer Clock "Low" Level Width	t_{SCKL}	$\overline{SCK}$	(Note 2)	0.5	–	–	t_{Scyc}	1, 2
Transfer Clock Rise Time	t_{SCKr}	$\overline{SCK}$	(Note 2)	–	–	100	ns	1, 2
Transfer Clock Fall Time	t_{SCKf}	$\overline{SCK}$	(Note 2)	–	–	100	ns	1, 2
Serial Output Data Delay Time	t_{DSO}	SO	(Note 2)	–	–	250	ns	1, 2
Serial Input Data Set-up Time	t_{SSI}	SI		300	–	–	ns	1
Serial Input Data Hold Time	t_{HSI}	SI		150	–	–	ns	1

• At Transfer Clock Input

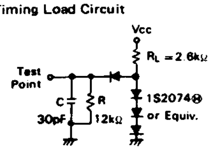
Item	Symbol	Pin Name	Test Conditions	Value			Unit	Note
				min	typ	max		
Transfer Clock Cycle Time	t_{Scyc}	$\overline{SCK}$		1	–	–	t_{cyc}	1
Transfer Clock "High" Level Width	t_{SCKH}	$\overline{SCK}$		0.5	–	–	t_{Scyc}	1
Transfer Clock "Low" Level Width	t_{SCKL}	$\overline{SCK}$		0.5	–	–	t_{Scyc}	1
Transfer Clock Rise Time	t_{SCKr}	$\overline{SCK}$		–	–	100	ns	1
Transfer Clock Fall Time	t_{SCKf}	$\overline{SCK}$		–	–	100	ns	1
Serial Output Data Delay Time	t_{DSO}	SO	(Note 2)	–	–	250	ns	1, 2
Serial Input Data Set-up Time	t_{SSI}	SI		300	–	–	ns	1
Serial Input Data Hold Time	t_{HSI}	SI		150	–	–	ns	1

(Note 1) Timing Diagram of Serial Interface

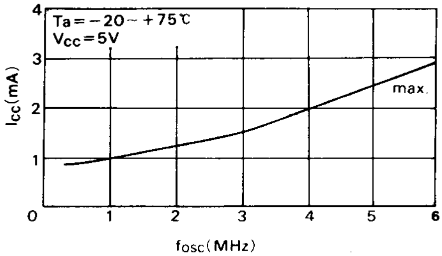


* V_{CC}−2.0V and 0.8V are the threshold voltage for transfer clock output.
0.7V_{CC} and 0.22 V_{CC} are the threshold voltage for transfer clock input.

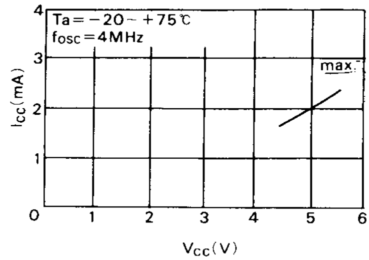
(Note 2) Timing Load Circuit



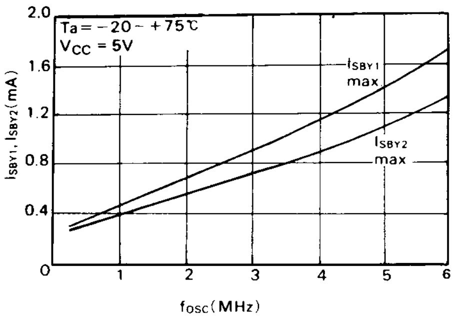
■ **CHARACTERISTICS CURVE (REFERENCE DATA)**



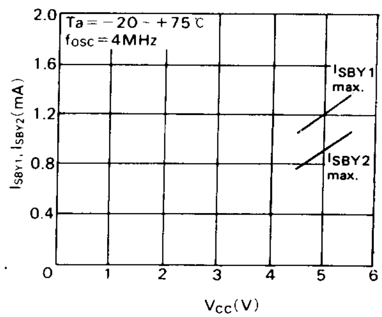
I_{CC} vs. f_{osc} characteristic
(crystal, ceramic resonator)



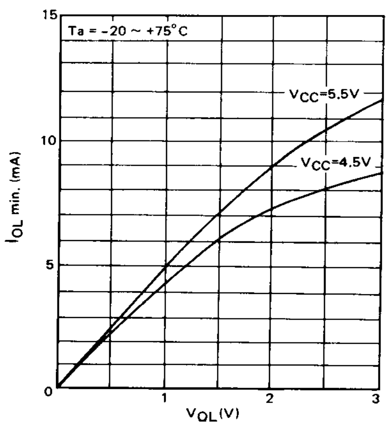
I_{CC} vs. V_{CC} characteristic
(crystal, ceramic resonator)



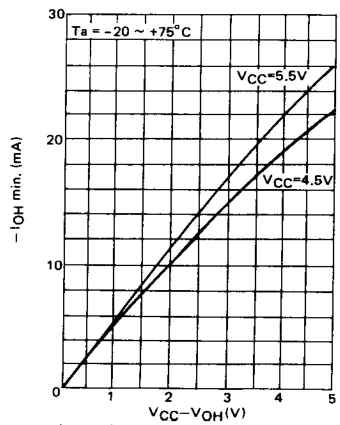
I_{SBY} vs. f_{osc} characteristics
(crystal, ceramic resonator)



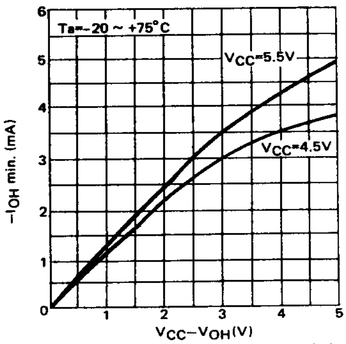
I_{SBY} vs. V_{CC} characteristics
(crystal, ceramic resonator)



I_{OL} min. vs. V_{OL} characteristics
(Standard Pin)



$-I_{OH}$ min. vs. ($V_{CC}-V_{OH}$) characteristics
(D_4 ~ D_{15} pins)



$-I_{OH}$ min. vs. $(V_{CC}-V_{OH})$ characteristics
(R0 ~ R2 pins)

■ DESCRIPTION OF PIN FUNCTIONS

Input and output signals of MCU are described below.

● GND, V_{CC}, V_{disp}

These are power supply pins. Connect GND pin to Earth (0V) and apply V_{CC} power supply voltage to V_{CC} pin. R_{A1}/V_{disp} pin is used for R_{A1} as all high voltage pins are "without pull-down MOS" (PMOS open drain).

● $\overline{TEST}$

$\overline{TEST}$ pin is not for users application. Connect it to V_{CC}.

● RESET

RESET pin is used to reset MCU. For details, see "RESET"

● OSC_1, OSC_2

These are input pins to the internal clock generator circuit. They can be connected to crystal resonator, ceramic filter resonator, or external oscillator circuit. For details, see "INTERNAL OSCILLATOR CIRCUIT."

● D-port (D_0 to D_{15})

D-port is a 1-bit Input/Output common port. D_0 to D_3 are standard type, D_4 to D_{15} are for high voltage. For details, see "INPUT/OUTPUT".

● R-port (R0 to RA)

R-port is a 4-bit Input/Output port. (only RA is 2-bit construction.) R0 and R6 to R8 are output ports, R9 to RA are input ports, and R1 to R5 are Input/Output common ports. R0 to R2 and RA are the high voltage ports, R3 to R9 are the standard ports. R_{32}, R_{33}, R_{40}, R_{41}, and R_{42} are also available as $\overline{INT_0}$, $\overline{INT_1}$, $\overline{SCK}$, SI and SO respectively. For details, see "INPUT/OUTPUT".

● $\overline{INT_0}$, $\overline{INT_1}$

These are the input pins to interrupt MCU operation externally. $\overline{INT_1}$ can be used as an external event input pin for TIMER-B. $\overline{INT_0}$ and $\overline{INT_1}$ are also available as R_{32}, and R_{33} respectively. For details, see "INTERRUPT".

● $\overline{SCK}$, SI, SO

These are transfer clock I/O pin ($\overline{SCK}$), serial data input pin (SI) and serial data output pin (SO) used for serial interface. $\overline{SCK}$, SI and SO are also available as R_{40}, R_{41}, and R_{42} respectively. For details, see "SERIAL INTERFACE".

■ ROM MEMORY MAP

ROM memory map is illustrated in Fig. 1 and described in the following paragraph.

● Vector Address Area ..... $0000 to $000F

When MCU reset or an interrupt is serviced, the program is executed from the vector address. Program the JMPL instructions branching to the starting addresses of reset routine or of interrupt routines.

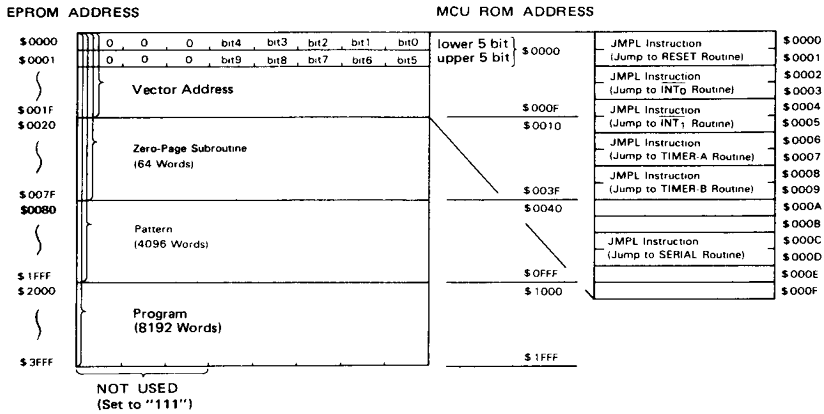
● Zero-Page Subroutine Area ..... $0000 to $003F

CAL instruction allows to branch to the subroutines in $0000 to $003F.

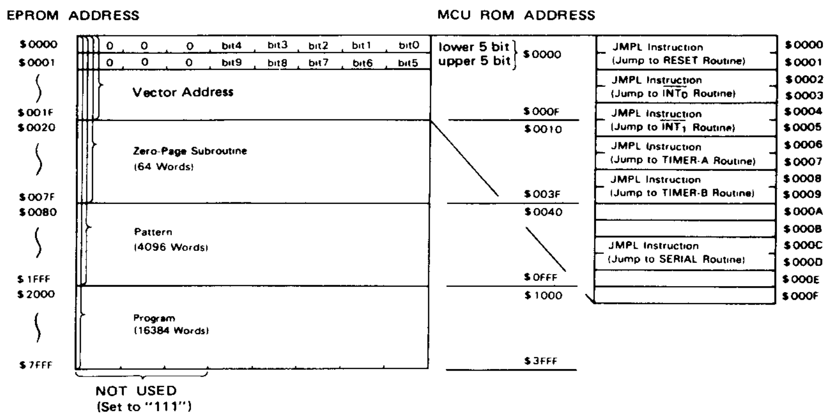
● Pattern Area ..... $0000 to $0FFF

P instruction allows referring to the ROM data in $0000 to $0FFF as a pattern.

● Program Area ..... $0000 to $1FFF; HD614P080S
$0000 to $3FFF; HD614P0160S

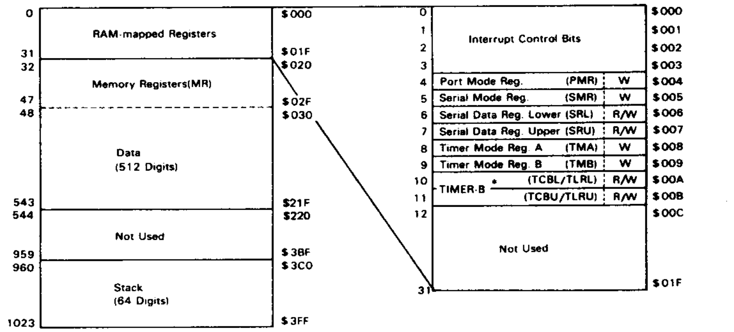


HD614P080S



HD614P0160S

Fig. 1 ROM Memory Map



* Two registers are mapped on same address.

R : Read Only
W : Write Only
R/W : Read/Write

Timer/Event Counter B Lower (TCBL)	R	Timer Load Reg. Lower (TLRL)	W	$00A
Timer/Event Counter B Upper (TCBU)	R	Timer Load Reg. Upper (TLRU)	W	$00B

Fig. 2 RAM Memory Map

■ RAM MEMORY MAP

The MCU includes 576 digits x 4 bits RAM as the data area and stack area. In addition to these areas, interrupt control bits and special registers are also mapped on the RAM memory space. RAM memory map is illustrated in Fig. 2 and described in the following paragraph.

	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)	I/E (Interrupt Enable Flag)	$000
1	IMTA (IM of TIMER-A)	IFTA (IF of TIMER-A)	IM1 (IM of $\overline{INT_1}$)	IF1 (IF of $\overline{INT_1}$)	$001
2	Not Used	Not Used	IMTB (IM of TIMER-B)	IFTB (IF of TIMER-B)	$002
3	Not Used	Not Used	IMS (IM of SERIAL)	IFS (IF of SERIAL)	$003

IF : Interrupt Request Flag
IM : Interrupt Mask
I/E : Interrupt Enable Flag
SP : Stack Pointer
(Note) Each bit in Interrupt Control Bits Area is set by SEM/SEMD instruction, is reset by REM/REMD instruction and is tested by TM/TMD instruction. It is not affected by other instructions. Furthermore, Interrupt Request Flag is not affected by SEM/SEMD instruction. The content of Status becomes invarid when "RSP" bit and "Not Used" bit is tested.

Fig. 3 Configuration of Interrupt Control Bit Area

- **Interrupt Control Bit Area ..... $000 to $003**

This area is used for interrupt controls, and is illustrated in Fig. 3. It is accessable only by RAM bit manipulation instruction. However, the interrupt request flag cannot be set by software. The RSP bit is only used to reset the SP.

- **Special Register Area ..... $004 to $00B**

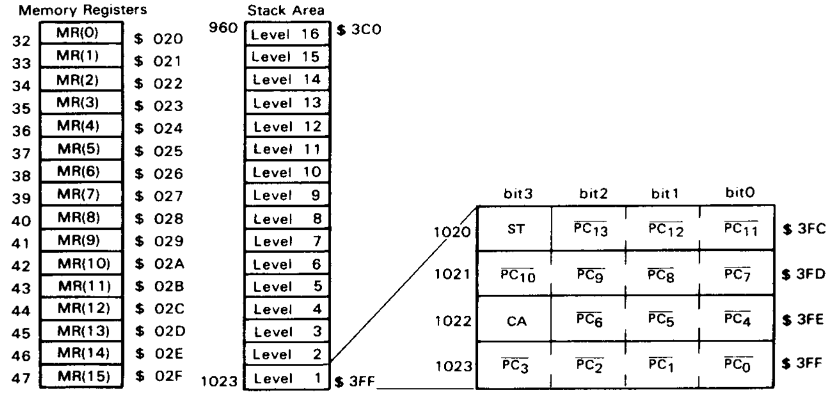
Special Register is a mode or a data register for the external interrupt, the serial interface, and the timer/counter. These registers are classified into 3 types: Write-only, Read-only, and Read/Write as shown in Fig. 2. These registers cannot be accessed by RAM bit manipulation instruction.

- **Data Area ..... $020 to $21F**

16 digits of $020 to $02F are called memory register (MR) and accessable by LAMR and XMRA instructions.

- **Stack Area .... $3C0 to $3FF**

Stack Area is used for LIFO stacks with the contents of the program counter (PC), status (ST) and carry (CA) when processing subroutine call and interrupt. As 1 level requires 4 digits, this stack area is nested to 16 level-stack max. The data pushed in the stack and LIFO stack state are provided in Fig. 4. The program counter is restored by RTN and RTNI instructions. Status and Carry are restored only by RTNI instruction, and not affected by RTN instruction. The area, not used for stacking, is available as a data area.



PC_{13} to PC_0; Program Counter
ST; Status
CA; Carry

Fig. 4 Configuration of Memory Register, Stack Area and Stack Position

■ REGISTER AND FLAG

The MCU has nine registers and two flags for the CPU operations. They are illustrated in Fig. 5 and described in the following paragraphs.

- **Accumulator (A), B Register (B)**

Accumulator and B Register are 4-bit registers used to hold the results of Arithmetic Logic Unit (ALU), and to transfer data to/from memories, I/O and other registers.

- **W Register (W), X Register (X), Y Register (Y)**

W Register is 2-bit, and X and Y Register are 4-bit registers used for indirect addressing of RAM. Y Register is also used for D-port addressing. W Register is write-only and cannot be read.

- **SPX Register (SPX), SPY Register (SPY)**

SPX and SPY Register are 4-bit registers used to assist X and Y Register respectively.

- **Carry (CA)**

Carry (CA) stores the overflow of ALU generated by the arithmetic operation. It is also affected by SEC, REC, ROTL and ROTR instructions.

During interrupt servicing, Carry is pushed onto the stack and restored back from the stack by RTNI instruction. (It's not affected by RTN instruction.)

- **Status (ST)**

Status (ST) holds the ALU overflow, ALU non-zero and the results of bit test instruction for the arithmetic or compare instruction. It is used for a branch condition of BR, BRL, CAL or CALL instructions. The value of the Status remains unchanged until the next arithmetic, compare or bit test instruction is executed. Status becomes "1" after the BR, BRL, CAL or CALL instruction has been executed (irrespective of its execution/skip). During the interrupt servicing, Status is pushed onto the stack and restored back from the stack by RTNI instruction. (It's not affected by RTN instruction.)

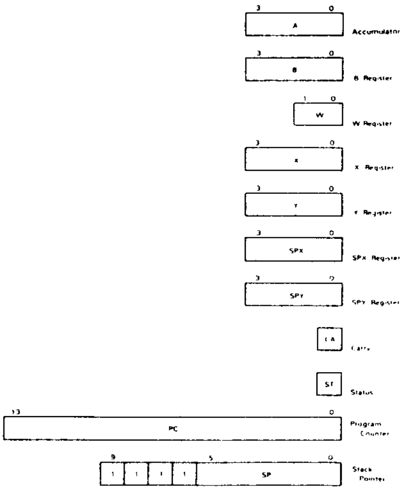


Fig. 5 Register and Flags

- **Program Counter (PC)**

Program Counter is a 14-bit binary counter for ROM addressing.

- **Stack Pointer (SP)**

Stack Pointer is used to point the address of the next stacking area up to 16 levels.

The Stack Pointer is initialized to locate $3FF on the RAM address, and is decremented by 4 as data pushed into the stack, and incremented by 4 as data restored back from the stack.

■ **INTERRUPT**

The MCU can be interrupted by five different sources: the external signals ($\overline{INT_0}$, $\overline{INT_1}$), timer/counter (TIMER-A, TIMER-B), and serial interface (SERIAL). In each sources, the Interrupt Request Flag, Interrupt Mask and interrupt vector address will be used to control and maintain the interrupt request. The Interrupt Enable Flag is also used to control the total interrupt operations.

- **Interrupt Control Bit and Interrupt Service**

The interrupt control bit is mapped on $000 to $003 of the RAM address and accessable by RAM bit manipulation instruction. (The Interrupt Request Flag (IF) cannot be set by software.) The Interrupt Enable Flag (I/E) and Interrupt Request Flag (IF) are set to "0", and the Interrupt Mask (IM) is set to "1" at the initialization by MCU reset.

Fig. 6 shows the interrupt block diagram. Table 1 shows the interrupt priority and vector addresses, and Table 2 shows the conditions that the interrupt service is executed by any one of the five interrupt sources.

The interrupt request is generated when the Interrupt Request Flag is set to "1" and the Interrupt Mask is "0". If the Interrupt Enable Flag is "1", then the interrupt will be activated and vector addresses will be generated from the priority PLA corresponding to the five interrupt sources.

Fig. 7 shows the interrupt services sequence, and Fig. 8 shows the interrupt flowchart. If the interrupt is requested, the instruction finishes its execution in the first cycle. The Interrupt Enable Flag is reset in the second cycle. In the second and third cycles, the Carry, Status and Program Counter are pushed onto the stack. In the third cycle, the instruction is executed again after jumping to the vector address.

In each vector address, program JMPL instruction to branch to a starting address of the interrupt routine. The Interrupt Request Flag which caused the interrupt service has to be reset by software in the interrupt routine.

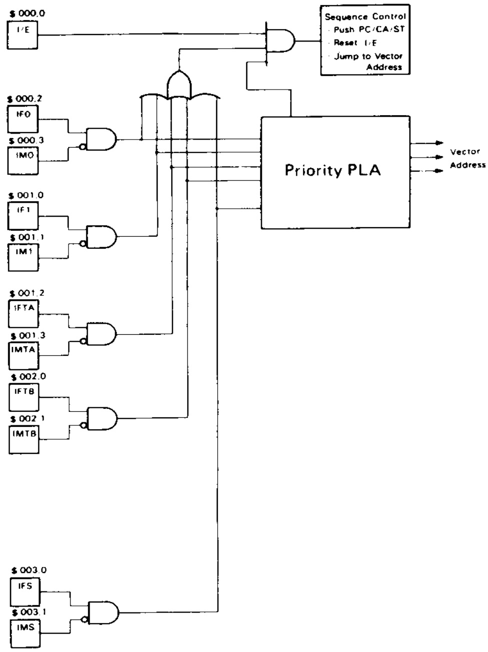


Fig. 6 Interrupt Circuit Block Diagram

Table 1. Vector Addresses and Interrupt Priority

Reset · Interrupt	Priority	Vector addresses
RESET	–	$0000
$\overline{INT_0}$	1	$0002
$\overline{INT_1}$	2	$0004
TIMER-A	3	$0006
TIMER-B	4	$0008
SERIAL	5	$000C

Table 2. Conditions of Interrupt Service

Interrupt control bits \ Interrupt source	$\overline{INT_0}$	$\overline{INT_1}$	TIMER-A	TIMER-B	SERIAL
I/E	1	1	1	1	1
IF0·$\overline{IM0}$	1	0	0	0	0
IF1 · $\overline{IM1}$	*	1	0	0	0
IFTA · $\overline{IMTA}$	*	*	1	0	0
IFTB · $\overline{IMTB}$	*	*	*	1	0
IFS · $\overline{IMS}$	*	*	*	*	1

* Don't care

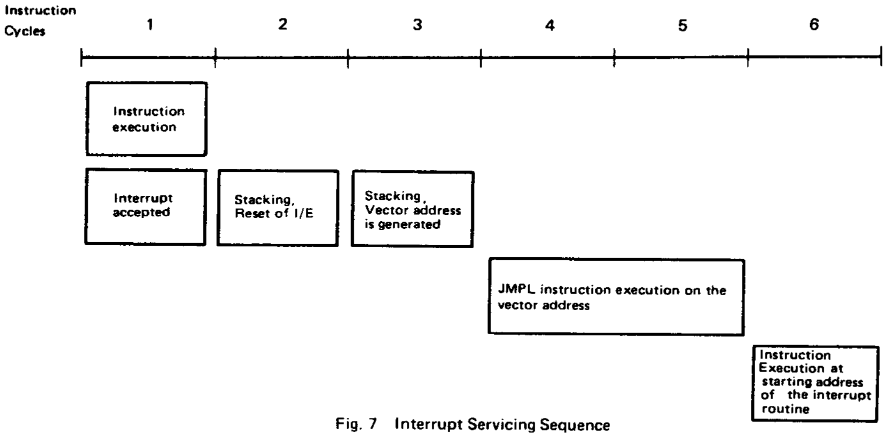


Fig. 7 Interrupt Servicing Sequence

- **Interrupt Enable Flag (I/E: $000,0)**

The Interrupt Enable Flag controls enable/disable of all interrupt requests as shown in Table 3. The Interrupt Enable Flag is reset by the interrupt servicing and set by RTNI instruction.

Table 3. Interrupt Enable Flag

Interrupt Enable Flag	Interrupt Enable/Disable
0	Disable
1	Enable

- **External Interrupt ($\overline{INT_0}$, $\overline{INT_1}$)**

To use external interrupt, select $R_{32}/\overline{INT_0}$, $R_{33}/\overline{INT_1}$ port for $\overline{INT_0}$, $\overline{INT_1}$ mode by setting the Port Mode Register (PMR: $004).

The External Interrupt Request Flags (IF0, IF1) are set at the falling edge of $\overline{INT_0}$, $\overline{INT_1}$ inputs.

$\overline{INT_1}$ input can be used as a clock signal input of TIMER-B. Then, TIMER-B counts up at each falling edge of input. When using $\overline{INT_1}$ as TIMER-B external event, an External Interrupt Mask (IM1) has to be set so that the interrupt request by $\overline{INT_1}$ will not be accepted.

- **External Interrupt Request Flag (IF0: $000,2, IF1: $001,0)**

The External Interrupt Request Flags (IF0, IF1) are set at the falling edges of $\overline{INT_0}$, $\overline{INT_1}$ inputs respectively.

- **External Interrupt Mask (IM0: $000,3, IM1: $001,1)**

The External Interrupt Mask is used to mask the external interrupt requests.

Table 4. External Interrupt Request Flag

External Interrupt Request Flags	Interrupt Requests
0	No
1	Yes

Table 5. External Interrupt Mask

External Interrupt Masks	Interrupt Requests
0	Enable
1	Disable (masks)

- **Port Mode Register (PMR: $004)**

The Port Mode Register is a 4-bit write-only register which controls the $R_{32}/\overline{INT_0}$ pin, $R_{33}/\overline{INT_1}$ pin, R_{41}/SI pin and R_{42}/SO pin as shown in Table 6. The Port Mode Register will be initialized to $0 by MCU reset, so that all these pins are set to a port mode.

Table 6. Port Mode Register

PMR bit 3	$R_{33}/\overline{INT_1}$ pin
0	Used as R_{33} port input/output pin
1	Used as $\overline{INT_1}$ input pin

PMR bit 2	$R_{32}/\overline{INT_0}$ pin
0	Used as R_{32} port input/output pin
1	Used as $\overline{INT_0}$ input pin

PMR bit 1	R_{41}/SI pin
0	Used as R_{41} port input/output pin
1	Used as SI input pin

PMR bit 0	R_{42}/SO pin
0	Used as R_{42} port input/output pin
1	Used as SO output pin

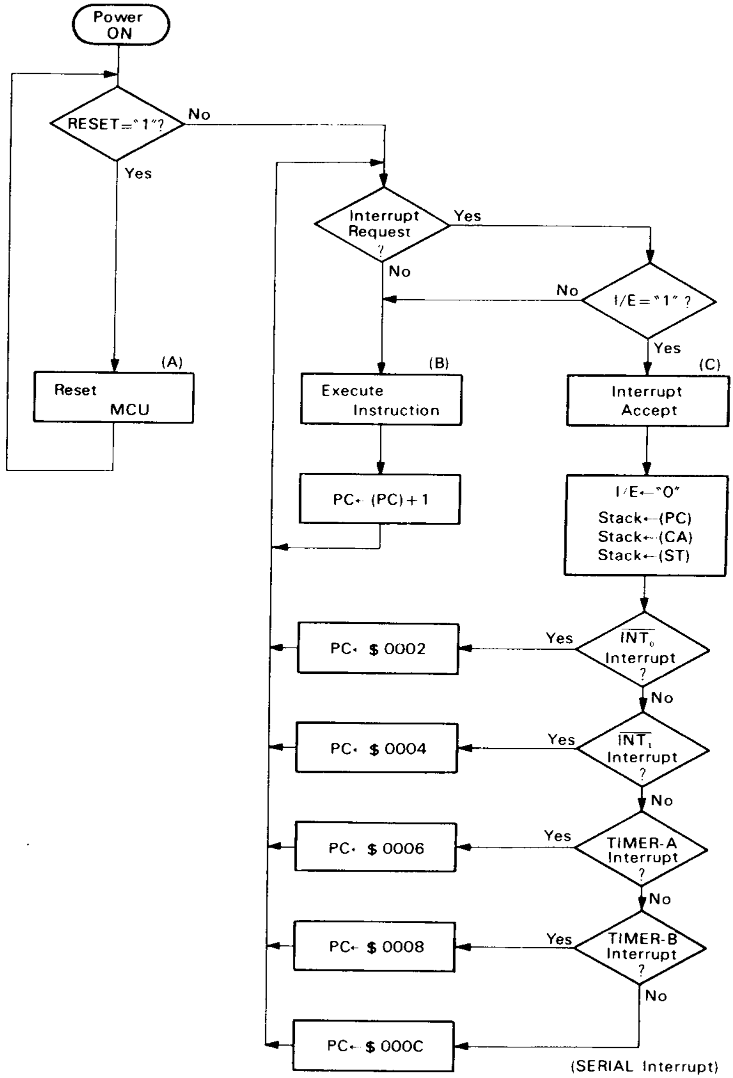


Fig. 8 Interrupt Servicing Flowchart

■ **SERIAL INTERFACE**

The serial interface is used to transmit/receive 8-bit data serially. This consists of the Serial Data Register, the Serial Mode Register, the Octal Counter and the multiplexer, as illustrated in Fig. 9. Pin $R_{40}/\overline{SCK}$ and the transfer clock signal are controlled by the Serial Mode Register. Contents of the Serial Data Register can be written into or read out by the software. The data in the Serial Data Register can be shifted synchronously with the transfer clock signal.

The serial interface operation is initiated with STS instruction. The Octal Counter is reset to $0 by STS instruction. It starts to count at the falling edge of the transfer clock ($\overline{SCK}$) signal and increments by one at the rising edge of the $\overline{SCK}$. When the Octal Counter is reset to $0 after eight transfer clock signals, or discontinued transmit/receive operation by resetting the Octal Counter, the SERIAL Interrupt Request Flag will be set.

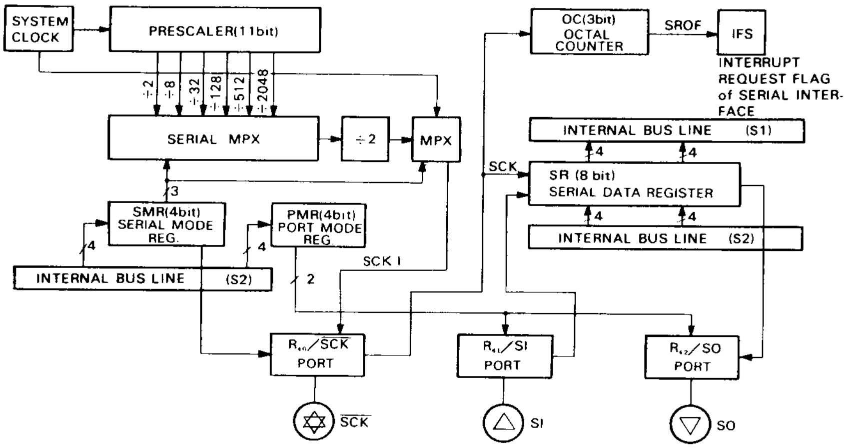


Fig. 9 Serial Interface Block Diagram

● **Serial Mode Register (SMR: $005)**

The Serial Mode Register is a 4-bit write-only register. This register controls the $R_{40}/\overline{SCK}$ and the prescaler divide ratio as the transfer clock source as shown in Table 7.

The Write Signal to the Serial Mode Register controls the operating state of serial interface.

The Write Signal to the Serial Mode Register stops the transfer clock applied to the Serial Data Register and the Octal Counter. And it also reset the Octal Counter to $0 simultaneously.

When the Serial Interface is in the "Transfer State", the Write Signal to the Serial Mode Register causes to quit the data transfer and to set the SERIAL Interrupt Request Flag.

Contents of the Serial Mode Register will be changed on the second instruction cycle after writing into the Serial Mode Register. Therefore, it will be necessary to execute the STS instruction after the data in the Serial Mode Register has been changed completely. The Serial Mode Register will be reset to $0 by MCU reset.

● **Serial Data Register (SRL: $006, SRU: $007)**

The Serial Data Register is an 8-bit read/write register. It consists of a low-order digit (SRL:$006) and a high-order digit (SRU: $007).

The data in the Serial Data Register will be output from the LSB side at SO pin synchronously with the falling edge of the transfer clock signal. At the same time, external data will be input from the LSB side at SI pin to the Serial Data Register synchronously with the rising edge of the transfer clock. Fig. 10 shows the I/O timing chart for the transfer clock signal and the data.

The writing into/reading from the Serial Data Register during its shifting causes the validity of the data.

Therefore complete data transmit/receive before writing into/reading from the serial data register.

Table 7. Serial Mode Register

SMR Bit 3	$R_{40}/\overline{SCK}$
0	Used as R_{40} port input/output pin
1	Used as $\overline{SCK}$ input/output pin

SMR			Transfer Clock			
Bit 2	Bit 1	Bit 0	$R_{40}/\overline{SCK}$ Port	Clock Source	Prescaler Divide Ratio	System Clock Divide Ratio
0	0	0	$\overline{SCK}$ Output	Prescaler	÷ 2048	÷ 4096
0	0	1	$\overline{SCK}$ Output	Prescaler	÷ 512	÷ 1024
0	1	0	$\overline{SCK}$ Output	Prescaler	÷ 128	÷ 256
0	1	1	$\overline{SCK}$ Output	Prescaler	÷ 32	÷ 64
1	0	0	$\overline{SCK}$ Output	Prescaler	÷ 8	÷ 16
1	0	1	$\overline{SCK}$ Output	Prescaler	÷ 2	÷ 4
1	1	0	$\overline{SCK}$ Output	System Clock	–	÷ 1
1	1	1	$\overline{SCK}$ Input	External Clock	–	–

(In the case of SMR Bit 3 = 1)

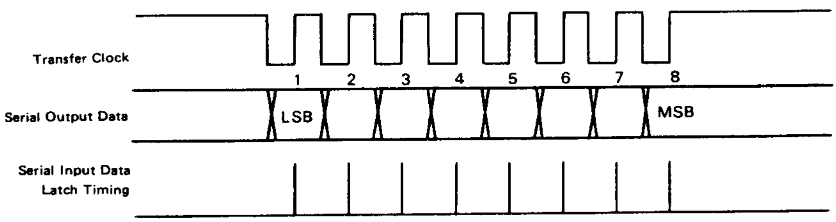


Fig. 10 Serial Interface I/O Timing Chart

- **SERIAL Interrupt Request Flag (IFS: $003, 0)**
  The SERIAL Interrupt Request Flag will be set after the eight transfer clock signals or transmit/receive discontinued operation by resetting the Octal Counter.

- **SERIAL Interrupt Mask (IMS: $003, 1)**
  The SERIAL Interrupt Mask masks the interrupt request.

Table 8. SERIAL Interrupt Request Flag

SERIAL Interrupt Request Flag	Interrupt Request
0	No
1	Yes

Table 9. SERIAL Interrupt Mask

SERIAL Interrupt Mask	Interrupt Request
0	Enable
1	Disable (masks)

- **Selection of the Operation Mode**
  Table 10 shows the operation mode of the serial interface. Select a combination of the value in the Port Mode Register and the Serial Mode Register according to Table 10.
  Initialize the serial interface by the Write Signal to the Serial Mode Register, when the Operation Mode is changed.
- **Operating State of Serial Interface**
  The serial interface has 3 operating states as shown in Fig. 11.
  The serial interface gets into "STS waiting state" by 2 ways: one way is to change the operation mode by changing the data

in the Port Mode Register, the other is to write data into the Serial Mode Register. In this state, the serial interface does not operate although the transfer clock is applied. If STS instruction is executed, the serial interface changes its state to "SCK waiting state".

In the "SCK waiting state", the falling edge of first transfer clock affects the serial interface to get into "transfer state", while the Octal Counter counts-up and the Serial Data Register shifts simultaneously. As an exception, if the clock continuous output mode is selected, the serial interface stays in "SCK waiting state" while the transfer clock outputs continuously.

The Octal Counter becomes "000" again by 8 transfer clocks or execution of STS instruction, so that the serial interface gets back into the "SCK waiting state", and SERIAL Interrupt Request Flag is set simultaneously.

When the internal transfer clock is selected, the transfer clock output are triggered by the execution of STS instruction, and it stops after 8 clocks.

- **Example of Transfer Clock Error Detection**

The serial interface functions abnormally when the transfer clock was disturbed by external noises. In this case, the transfer clock error can be detected in the procedure shown in Fig. 12.

If more than 9 transfer clocks are applied by the external noises in the "SCK waiting state", the state of the serial interface shifts as the following sequence: first "transfer state" (while 1 to 7 transfer clocks), second "SCK waiting state" (at 8th transfer clock) and third "transfer state" again. Then reset the SERIAL Interrupt Request Flag, and make "STS waiting state" by writing to the Serial Mode Register. SERIAL Interrupt Request Flag is set again in this procedure, and it shows that the transfer clock was invalid and that the transmit/receive data were also invalid.

Table 10. Serial Interface Operation Mode

SMR	PMR		Serial Interface Operating Mode
Bit 3	Bit 1	Bit 0	
1	0	0	Clock Continuous Output Mode
1	0	1	Transmit Mode
1	1	0	Receive Mode
1	1	1	Transmit/Receive Mode

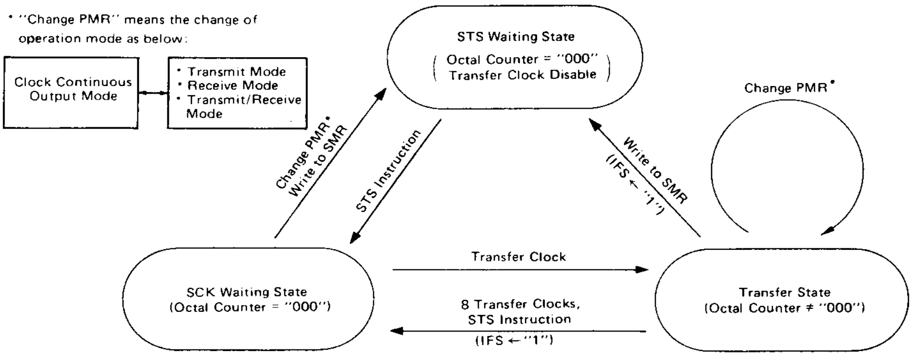


Fig. 11 Serial Interface Operation State

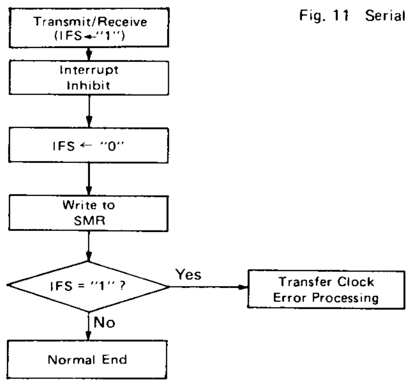


Fig. 12 Example of Transfer Clock Error Detection

■ TIMER

The MCU contains a prescaler and two timer/counters (TIMER-A, TIMER-B). Fig. 13 shows the block diagram. The prescaler is an 11-bit binary counter. TIMER-A is an 8-bit free-run timer. TIMER-B is an 8-bit auto-reload timer/event counter.

- **Prescaler**

The input to the prescaler is a system clock signal. The prescaler is initialized to $000 by MCU reset, and the prescaler starts to count up the system clock signal as soon as RESET input goes to logic "0". The prescaler keeps counting up except MCU reset and stop mode. The prescaler provides clock signals to TIMER-A, TIMER-B and serial interface. The prescaler divide ratio of the clock signals are selected according to the content of the mode registers such as – Timer Mode Register A (TMA), Timer Mode Register B (TMB), Serial Mode Register (SMR).

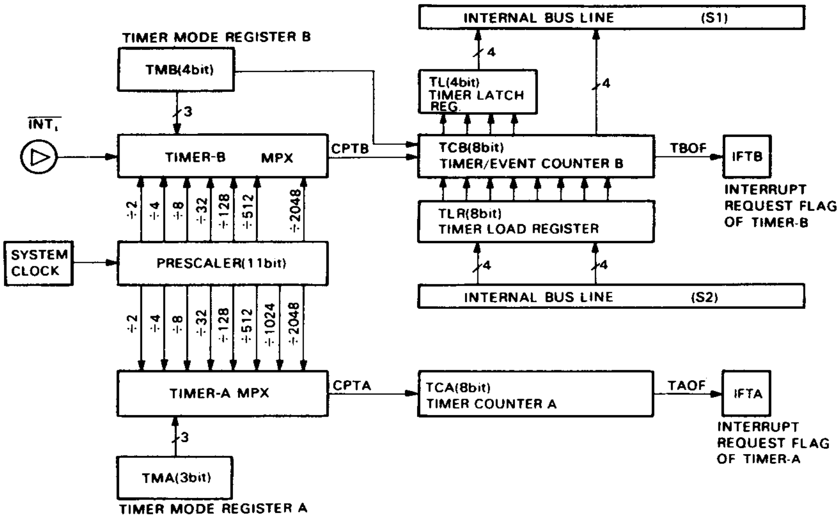


Fig. 13 Timer/Counter Block Diagram

- **TIMER-A Operation**

After TIMER-A is initialized to $00 by MCU reset, it counts up at every clock input signal. When the next clock signal is applied after TIMER-A is counted up to $FF, TIMER-A is set to $00 again, and generating overflow output. This leads to setting TIMER-A Interrupt Request Flag (IFTA: $001, 2) to "1". Therefore, this timer can function as an interval timer periodically generating overflow output at every 256th clock signal input.

The clock input signals to TIMER-A are selected by the Timer Mode Register A (TMA: $008).

- **TIMER-B Operation**

Timer Mode Register B (TMB: $009) is used to select the auto-reload function and the prescaler divide ratio of TIMER-B as the input clock source. When the external event input is used as an input clock signal to TIMER-B, select the $R_{33}/\overline{INT_1}$ as $\overline{INT_1}$ and set the External Interrupt Mask (IM1) to "1" to prevent the external interrupt request from occurring.

TIMER-B is initialized according to the value written into the Timer Load Register by software. TIMER-B counts up at every clock input signal. When the next clock signal is applied to TIMER-B after TIMER-B is set to $FF, TIMER-B will be initialized again and generate overflow output. In this case if the auto-reload function is selected, TIMER-B is initialized according to the value of the Timer Load Register. Else if the auto-reload function is not selected, TIMER-B goes to $00. TIMER-B Interrupt Request Flag (IFTB: $002,0) will be set at this overflow output.

- **Timer Mode Register A (TMA: $008)**

The Timer Mode Register A is a 3-bit write-only register. The TMA controls the prescaler divide ratio of TIMER-A clock input, as shown in Table 11.

The Timer Mode Register A is initialized to $0 by MCU reset.

- **Timer Mode Register B (TMB: $009)**

The Timer Mode Register B is a 4-bit write-only register. The Timer Mode Register B controls the selection for the auto-reload function of TIMER-B and the prescaler divide ratio, and the source of the clock input signal, as shown in Table 12.

The Timer Mode Register B is initialized to $00 by MCU reset.

The operation mode of TIMER—B is changed at the second instruction cycle after writing into the Timer Mode Register B.

Therefore, it is necessary to program the write instruction to TLRU after the content of TMB is changed.

Table 11. Timer Mode Register A

TMA			Prescaler Divide Ratio
Bit 2	Bit 1	Bit 0	
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

Table 12. Timer Mode Register B

TMB Bit 3	Auto-reload Function
0	No
1	Yes

TMB 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	$\overline{INT_1}$ (External Event Input)

- **TIMER-B (TCBL: $00A, TCBU: $00B / TLRL: $00A, TLRU: $00B)**

TIMER-B consists of an 8-bit write-only Timer Load Register, and an 8-bit read-only Timer/Event Counter. Each of them has a low-order digit (TCBL: $00A, TLRL: $00A) and a high-order digit (TCBU: $00B, TLRU: $00B).

The Timer/Event Counter can be initialized by writing data into the Timer Load Register. In this case, write the low-order digit first, and then the high-order digit. The Timer/Event Counter is initialized at the time when the high-order digit is written. The Timer Load Register will be initialized to $00 by the MCU reset.

The counter value of TIMER-B can be obtained by reading the Timer/Event Counter. In this case, read the high-order digit first, and then the low-order digit. The count value of low-order digit is latched at the time when the high-order digit is read.

- **TIMER-A Interrupt Request Flag (IFTA: $001, 2)**

The TIMER-A Interrupt Request Flag is set by the overflow output of TIMER-A.

- **TIMER-A Interrupt Mask (IMTA: $001, 3)**

TIMER-A Interrupt Mask prevents an interrupt request generated by TIMER-A Interrupt Request Flag.

Table 13. TIMER-A Interrupt Request Flag

TIMER-A Interrupt Request Flag	Interrupt Request
0	No
1	Yes

Table 14. TIMER-A Interrupt Mask

TIMER-A Interrupt Mask	Interrupt Request
0	Enable
1	Disable (Mask)

- **TIMER-B Interrupt Request Flag (IFTB: $002, 0)**

The TIMER-B Interrupt Request Flag is set by the overflow output of TIMER-B.

- **TIMER-B Interrupt Mask (IMTB: $002, 1)**

TIMER-B Interrupt Mask prevents an interrupt request generated by TIMER-B Interrupt Request Flag.

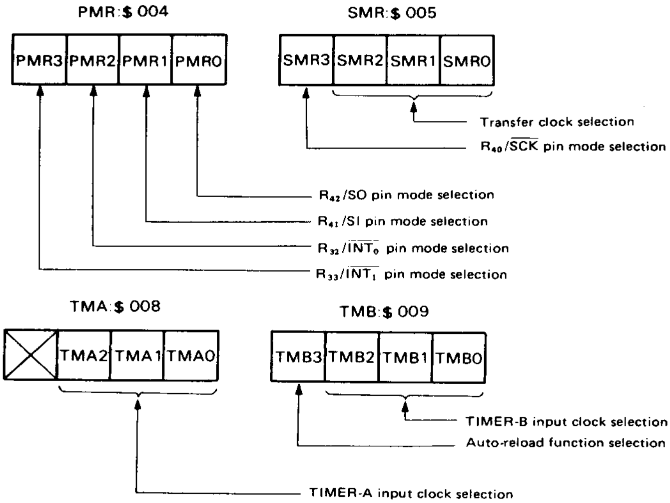


Fig. 14 Mode Register Configuration and Function

Table 15. TIMER-B Interrupt Request Flag

TIMER-B Interrupt Request Flag	Interrupt Request
0	No
1	Yes

Table 16. TIMER-B Interrupt Mask

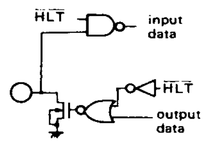
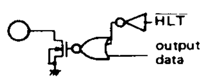
TIMER-B Interrupt Mask	Interrupt Request
0	Enable
1	Disable (Mask)

■ **INPUT/OUTPUT**

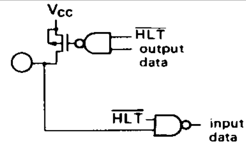
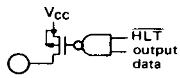
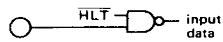
The MCU provides 58 Input/Output pins, and they are consist of 32 standard pins of "Without pull-up MOS (NMOS open drain)" and 26 high voltage pins of "Without pull-down MOS (PMOS open drain)".

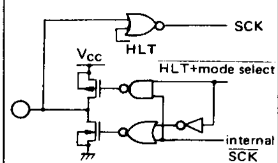
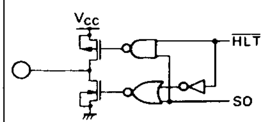
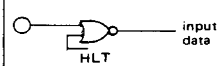
When any input/output common pin is used as input pin, it is necessary to set the output data as shown in Table 18.

Table 17 I/O Pin Circuit Forms

		Without pull-up MOS (NMOS open drain)	Applied pins
Standard pins	I/O common pins		$D_0 \sim D_3$, $R_{30} \sim R_{33}$, $R_{40} \sim R_{43}$, $R_{50} \sim R_{53}$
Standard pins	Output pins		$R_{60} \sim R_{63}$, $R_{70} \sim R_{73}$, $R_{80} \sim R_{83}$
Standard pins	Input pins		$R_{90} \sim R_{93}$

(Continued)

		Without pull-down MOS (PMOS open drain)	Applied pins
High voltage pins	I/O common pins		$D_4 \sim D_{15}$, $R_{10} \sim R_{13}$, $R_{20} \sim R_{23}$
High voltage pins	Output pins		$R_{00} \sim R_{03}$
High voltage pins	Input pins		R_{A0}, R_{A1}/V_{disp}

		Without pull-up MOS (NMOS open drain)	Applied pins
Standard pins	I/O common pins		$\overline{SCK}$ (Note 2) (Output Mode)
Standard pins	Output pins		SO
Standard pins	Input pins		$\overline{INT_0}$, $\overline{INT_1}$, SI, $\overline{SCK}$ (Note 2) (Input Mode)

(Note 1) In the stop mode, $\overline{HLT}$ signal is "0", HLT signal is "1" and I/O pins are in high impedance state.
(Note 2) If the MCU is interrupted by serial interface in the external clock input mode, the $\overline{SCK}$ terminal becomes input only.

Table 18 Data Input from Input/Output Common Pins

I/O circuit type	Available pin condition for input
For Standard pins "Without pull-up MOS (NMOS open drain)"	"1"
For High voltage pins "Without pull-down MOS (PMOS open drain)"	"0"

- **D-port**

D-port is 1-bit I/O port, and it has 16 Input/Output common pins. It can be set/reset by the SED/RED and SEDD/REDD instructions, and can be tested by the TD and TDD instructions. Table 17 shows the classification of standard pins, high voltage pins and the Input/Output pins circuit types.

- **R-port**

R-port is 4-bit I/O port. It provides 20 input/output common pins, 16 output-only pins, and 6 input-only pins. Data input is processed using the LAR and LBR instructions and data output is processed using the LRA and LRB instructions. The MCU will not be affected by writing into the input-only and/or non-existing ports, invalid data will be read by reading from the output-only and/or non-existing ports.

The R_{32}, R_{33}, R_{40}, R_{41} and R_{42} pins are also used as the $\overline{INT_0}$, $\overline{INT_1}$, $\overline{SCK}$, SI and SO pins respectively. Table 17 shows the classification of standard pins, high voltage pins and Input/Output pins circuit types.

- **RESET**

The MCU is reset by setting RESET pin to "1". At power

Table 19 MCU Initial Value by Reset

Items			Initial value by MCU reset	Contents
Program counter (PC)			$0000	Execute program from the top of ROM address.
Status (ST)			"1"	Enable to branch with conditional branch instructions.
Stack pointer (SP)			$3FF	Stack level is 0.
I/O output register	Standard pin	Without pull-up MOS	"1"	Enable to input.
	High voltage pin	Without pull-down MOS	"0"	Enable to input.
Interrupt flag	Interrupt Enable Flag (I/E)		"0"	Inhibit all interrupts.
	Interrupt Request Flag (IF)		"0"	No interrupt request.
	Interrupt Mask (IM)		"1"	Mask interrupt request.
Mode register	Port Mode Register (PMR)		"0000"	See Item "Port Mode Register".
	Serial Mode Register (SMR)		"0000"	See Item "Serial Mode Register".
	Timer Mode Register A (TMA)		"000"	See Item "Timer Mode Register A".
	Timer Mode Register B (TMB)		"0000"	See Item "Timer Mode Register B".
Timer/Counter, Serial Interface	Prescaler		$000	–
	Timer/Counter A (TCA)		$00	–
	Timer/Event Counter B (TCB)		$00	–
	Timer Load Register (TLR)		$00	–
	Octal Counter		"000"	–

(Note) The values of registers and flags which are not described on above table will become as follows.

Item		After releasing stop mode by MCU Reset	After MCU Reset except the left
Carry	(CA)	The value immediately before MCU reset is not guaranteed. Initialization by the program should be required.	The value immediately before MCU Reset is not guaranteed. Initialization by the program should be required.
Accumulator	(A)		
B register	(B)		
W register	(W)		
X/SPX register	(X/SPX)		
Y/SPY register	(Y/SPY)		
Serial data register (SR)		– ditto –	– ditto –
RAM		The value immediately before MCU reset (the value immediately before executing stop instruction) is retained.	– ditto –

ON or recovering from stop mode, apply RESET input more than t_{RC} to obtain the necessary time for oscillator stabilization. In other cases, the MCU reset requires at least two instructions cycle time of RESET input.

Table 19 shows initialized items by MCU reset and each status after reset.

■ INTERNAL OSCILLATOR CIRCUIT

Fig. 15 gives internal oscillator circuit. The oscillator type can be selected from the followings; crystal resonator, or ceramic filter resonator as shown in Table 20. In any cases, external clock operation is available.

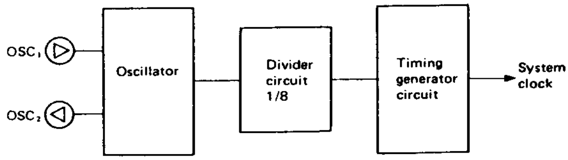
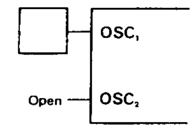
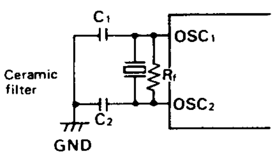
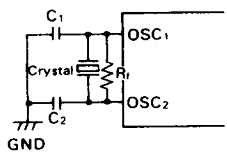
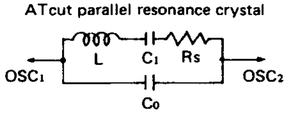


Fig. 15 Internal Oscillator Circuit

Table 20 Oscillator Circuit Example

	Circuit configuration	Remarks
External clock operation		
Ceramic filter resonator		Ceramic filter: CSA 4.00MG (Murata) R_f: 1MΩ ± 2% C_1: 33pF ± 20% C_2: 33pF ± 20% Ceramic filter: CSA 6.00MG (Murata) R_f: 1MΩ ±2% C_1: 30pF ± 20% C_2: 30pF ± 20%
Crystal resonator	 ATcut parallel resonance crystal 	Crystal: 4.194304 (MHz) NC-18C (Nihon Denpa Kogyo) R_f: 1MΩ ± 2% C_1: 22pF ± 20% C_2: 22pF ± 20% Crystal: 6.0 (MHz) NC-18C (Nihon Denpa Kogyo) R_f: 1MΩ ± 2% C_1: 20pF ±20% C_2: 20pF ± 20% Crystal: ATcut parallel resonance crystal C_O : 7 pF max. R_S : 100Ω max. f : 2.0 ~ 6.2MHz

(Note 1) On the crystal and ceramic filter resonator, the upper circuit parameters are the one recommended by crystal or ceramic filter maker. The circuit parameters are changed by crystal, ceramic filter resonator and the floated capacitance in designing the board. In employing the resonator, please consult with the engineers of crystal or ceramic filter maker to determine the circuit parameter.

(Note 2) Wiring among OSC_1, OSC_2 and elements should be as short as possible, and never cross the other wirings. Refer to the layout of crystal and ceramic filter.

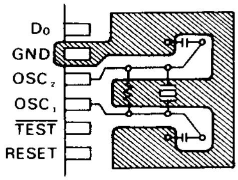


Fig. 16 Layout of Crystal and Ceramic Filter

■ **LOW POWER DISSIPATION MODE**

The MCU provides two low power dissipation modes, that is, a Standby mode and a Stop mode. Table 21 shows the function of the low power dissipation mode, and Fig. 17 shows the diagram of the mode transition.

Table 21 Low Power Dissipation Mode Function

Low Power Dissipation Mode	Instruction	Condition							Recovering method
		Oscillator circuit	Instruction execution	Register, Flag	Interrupt function	RAM	Input/ Output pin	Timer/ Counter, Serial Interface	
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

*1) STOP mode is released only by MCU Reset. Refer to Table 19 as for the values of the registers and flags after releasing stop mode.
*2) Current flows in I/O Circuit by I/O pin state at stand-by mode, because I/O circuit is active.
This current is an addition to stand-by mode power dissipation.

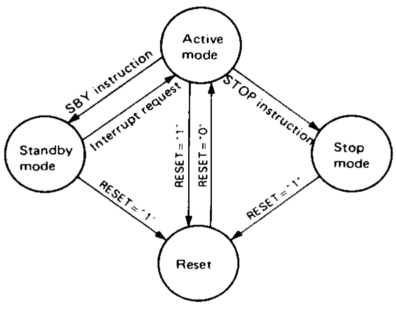


Fig. 17 MCU Operation Mode Transition

● **Standby Mode**

The SBY instruction puts the MCU into the Standby mode. In the Standby mode, the oscillator circuit is active and timer/counter and serial interface continue working. On the other hand, the CPU stops since the clock related to the instruction execution stops. Registers, RAM and Input/Output pins retain the state they had just before going into the Standby mode.

The Standby mode is canceled by the MCU reset or interrupt request. When canceled by the interrupt request, the MCU becomes an active mode and executes the instruction next to the SBY instruction. At this time, if the Interrupt Enable Flag is "1", the interrupt is executed. If the Interrupt Enable Flag is "0", the interrupt request is held on and the normal instruction execution continues.

Fig. 18 shows the flowchart of the Standby Mode.

● **Stop Mode**

The STOP instruction brings the MCU into the Stop mode. In this mode the oscillator circuit and every function of the MCU stop.

The Stop mode is canceled by the MCU reset. At this time, as shown in Fig. 19, apply the RESET input for more than t_{RC} to get enough oscillator stabilization time. (Refer to the "AC CHARACTERISTICS".) After the Stop mode is canceled, RAM retains the state it had just before going into the Stop mode after releasing stop mode by MCU reset, the values of the B register, W register, X/SPX register, Y/SPY register, carry and serial data register are not guaranteed.

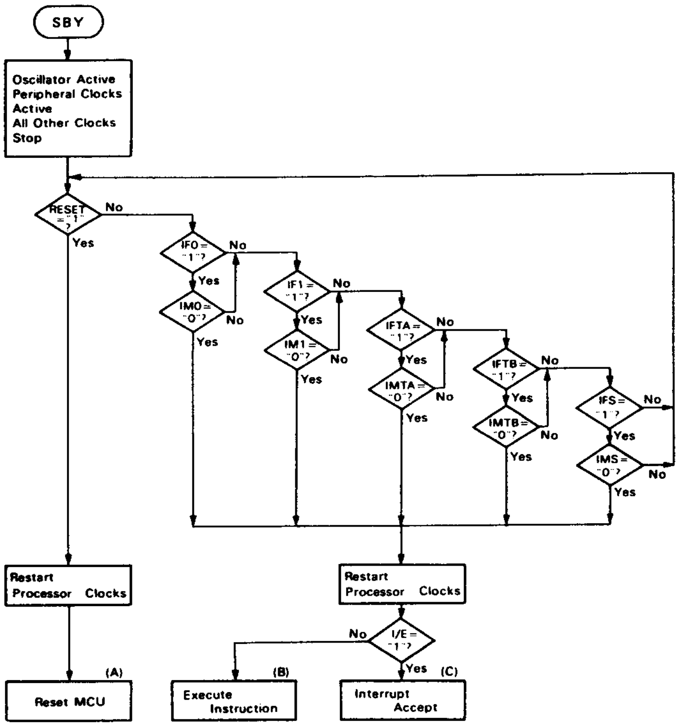


Fig. 18 MCU Operating Flowchart

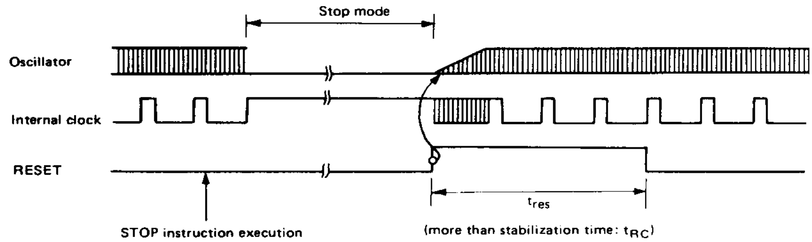


Fig. 19 Stop Mode Cancel Timing Chart

■ **RAM ADDRESSING MODE**

As shown in Fig. 20, the MCU provides three RAM addressing modes; Register Indirect Addressing, Direct Addressing and Memory Register Addressing.

● **Register Indirect Addressing**

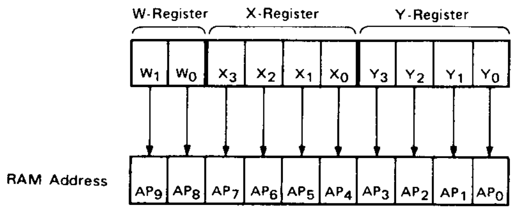
The combined 10-bit contents of W Register, X Register and Y Register is used as the RAM address in this mode.

● **Direct Addressing**

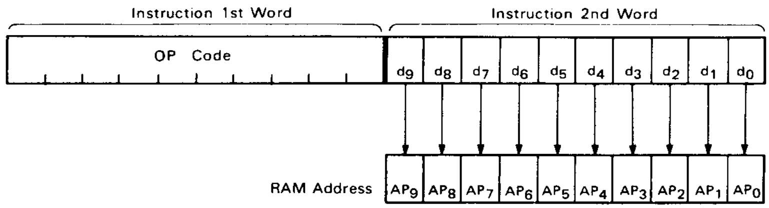
The direct addressing instruction consists of two words and the second word (10 bits) following Op-code (the first word) is used as the RAM address.

● **Memory Register Addressing**

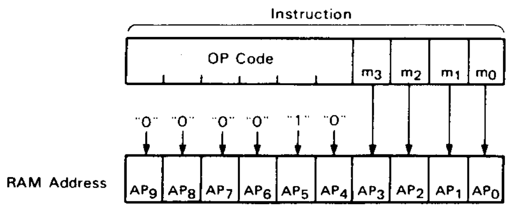
The Memory Register Addressing can access 16 digits (Memory Register: MR) from \$020 to \$02F by using the LAMR and XMRA instruction.



(a) Register Indirect Addressing



(b) Direct Addressing



(c) Memory Register Addressing

Fig. 20 RAM Addressing Mode

■ ROM ADDRESSING MODE AND P INSTRUCTION

The MCU has four kinds of ROM addressing modes as shown in Fig. 21.

● Direct Addressing Mode

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

● Current Page Addressing Mode

ROM memory space is divided into 256 words in each page starting from $0000. The program branches to the address in the same page using BR instruction. This instruction replace the low-order eight bits of program counter (PC_7 to PC_0) with 8-bit immediate data.

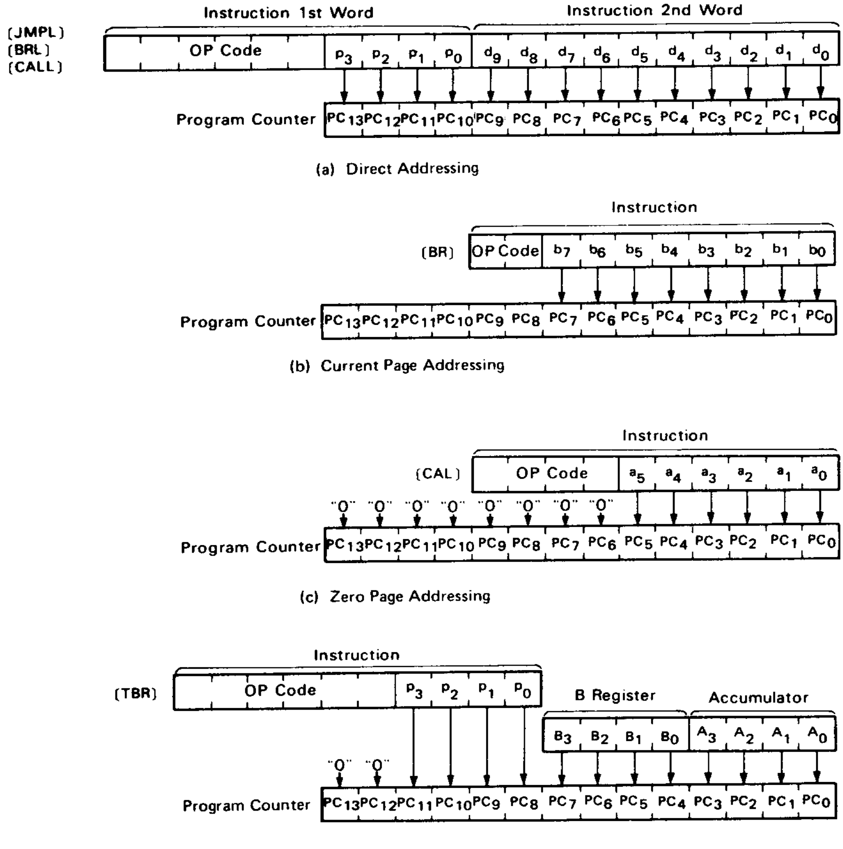
The branch destination by BR instruction on the boundary between pages is in the next page. Refer to Fig. 23.

● Zero Page Addressing Mode

The program branches to the zero page subroutine area, which is located on the address from $0000 to $003F, using CAL instruction. When CAL instruction is executed, 6-bit immediate data is placed in low-order six bits of program counter (PC_5 to PC_0) and "0's" are placed in high-order eight bits (PC_{13} to PC_6).

● Table Data Addressing

The program branches to the address determined by the contents of the 4-bit immediate data, accumulator and B register, using TBR instruction.



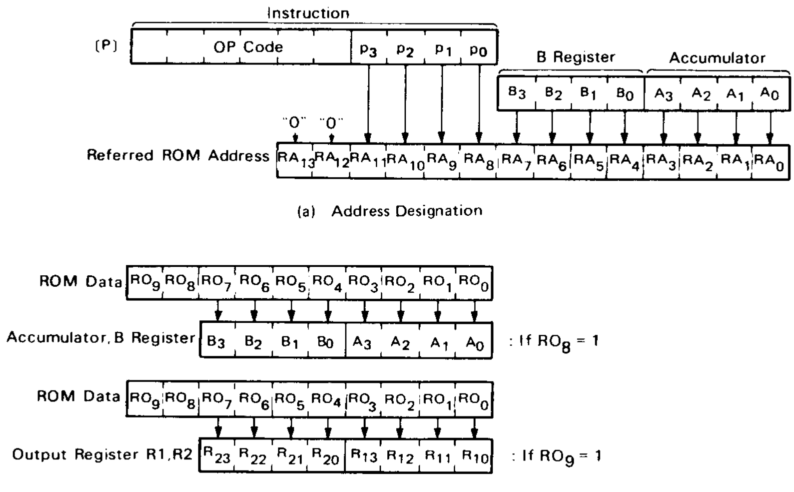
(a) Direct Addressing

(b) Current Page Addressing

(c) Zero Page Addressing

(d) Table Data Addressing

Fig. 21 ROM Addressing Mode



(a) Address Designation

(b) Pattern Output

Fig. 22 P Instruction

- **P Instruction (Pattern Instruction)**

By P instruction, the ROM data determined by Table Data addressing is referred. When bit 8 in referred ROM data is "1", 8 bits of referred ROM data are written into the accumulator and B Register. When bit 9 is "1", 8 bits of referred ROM data are written into the R1 and R2 port output register. When both bit 8 and 9 are "1", ROM data are written into the accumulator and B register and also to the R1 and R2 port output register at a same time.

The P instruction has no effect on the program counter.

- **Description of the branch destination on page boundary**

When BR is on page boundary (256n + 255), BR instruction transfers the contents of PC to the next page with hardware architecture. Therefore, the program branches to the next page when using BR on page boundary.

The HMCS400 series cross macro assembler has automatic paging facility for ROM page.

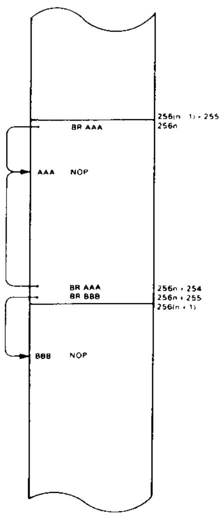


Fig. 23 The Branch Destination by BR Instruction on the Boundary between Pages

■ INSTRUCTION SET

The HD614P080S and HD614P0160S provide 99 instructions. These instructions are classified into 10 groups as follows;

(1) Immediate Instruction
(2) Register-to-Register Instruction
(3) RAM Address Instruction
(4) RAM Register Instruction
(5) Arithmetic Instruction
(6) Compare Instruction
(7) RAM Bit Manipulation Instruction
(8) ROM Address Instruction
(9) Input/Output Instruction
(10) Control Instruction

Table 22. Immediate Instruction

OPERATION	MNEMONIC	OPERATION CODE	FUNCTION	STATUS	WORD/CYCLE
Load A from Immediate	LAI i	1 0 0 0 1 1 i_3 i_2 i_1 i_0	i→A		1/1
Load B from Immediate	LBI i	1 0 0 0 0 0 i_3 i_2 i_1 i_0	i→B		1/1
Load Memory from Immediate	LMID i,d	0 1 1 0 1 0 i_3 i_2 i_1 i_0 d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	i→M		2/2
Load Memory from Immediate, Increment Y	LMIIY i	1 0 1 0 0 1 i_3 i_2 i_1 i_0	i→M, Y+1→Y	NZ	1/1

Table 23. Register-to-Register Instruction

OPERATION	MNEMONIC	OPERATION CODE	FUNCTION	STATUS	WORD/CYCLE
Load A from B	LAB	0 0 0 1 0 0 1 0 0 0	B→A		1/1
Load B from A	LBA	0 0 1 1 0 0 1 0 0 0	A→B		1/1
Load A from Y	LAY	0 0 1 0 1 0 1 1 1 1	Y→A		1/1
Load A from SPX	LASPX	0 0 0 1 1 0 1 0 0 0	SPX→A		1/1
Load A from SPY	LASPY	0 0 0 1 0 1 1 0 0 0	SPY→A		1/1
Load A from MR	LAMR m	1 0 0 1 1 1 $m_3 m_2 m_1 m_0$	MR(m)→A		1/1
Exchange MR and A	XMRA m	1 0 1 1 1 1 $m_3 m_2 m_1 m_0$	MR(m)↔A		1/1

Table 24. RAM Address Instruction

OPERATION	MNEMONIC	OPERATION CODE	FUNCTION	STATUS	WORD/CYCLE
Load W from Immediate	LWI i	0 0 1 1 1 1 0 0 i_1 i_0	i→W		1/1
Load X from Immediate	LXI i	1 0 0 0 1 0 i_3 i_2 i_1 i_0	i→X		1/1
Load Y from Immediate	LYI i	1 0 0 0 0 1 i_3 i_2 i_1 i_0	i→Y		1/1
Load X from A	LXA	0 0 1 1 1 0 1 0 0 0	A→X		1/1
Load Y from A	LYA	0 0 1 1 0 1 1 0 0 0	A→Y		1/1
Increment Y	IY	0 0 0 1 0 1 1 1 0 0	Y+1→Y	NZ	1/1
Decrement Y	DY	0 0 1 1 0 1 1 1 1 1	Y−1→Y	NB	1/1
Add A to Y	AYY	0 0 0 1 0 1 0 1 0 0	Y+A→Y	OVF	1/1
Subtract A from Y	SYY	0 0 1 1 0 1 0 1 0 0	Y−A→Y	NB	1/1
Exchange X and SPX	XSPX	0 0 0 0 0 0 0 0 0 1	X↔SPX		1/1
Exchange Y and SPY	XSPY	0 0 0 0 0 0 0 0 1 0	Y↔SPY		1/1
Exchange X and SPX,Y and SPY	XSPXY	0 0 0 0 0 0 0 0 1 1	X↔SPX, Y↔SPY		1/1

Table 25. RAM Register Instruction

OPERATION	MNEMONIC	OPERATION CODE	FUNCTION	STATUS	WORD/CYCLE
Load A from Memory	LAM(XY)	0 0 1 0 0 1 0 0 y x	M→A, (X↔SPX, Y↔SPY)		1/1
Load A from Memory	LAMD d	0 1 1 0 0 1 0 0 0 0 d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	M→A		2/2
Load B from Memory	LBM(XY)	0 0 0 1 0 0 0 0 y x	M→B, (X↔SPX, Y↔SPY)		1/1
Load Memory from A	LMA(XY)	0 0 1 0 0 1 0 1 y x	A→M, (X↔SPX, Y↔SPY)		1/1
Load Memory from A	LMAD d	0 1 1 0 0 1 0 1 0 0 d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	A→M		2/2
Load Memory from A, Increment Y	LMAIY(X)	0 0 0 1 0 1 0 0 0 x	A→M, Y+1→Y (X↔SPX)	NZ	1/1
Load Memory from A, Decrement Y	LMADY(X)	0 0 1 1 0 1 0 0 0 x	A→M, Y−1→Y (X↔SPX)	NB	1/1
Exchange Memory and A	XMA(XY)	0 0 1 0 0 0 0 0 y x	M↔A, (X↔SPX, Y↔SPY)		1/1
Exchange Memory and A	XMAD d	0 1 1 0 0 0 0 0 0 0 d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	M↔A		2/2
Exchange Memory and B	XMB(XY)	0 0 1 1 0 0 0 0 y x	M↔B, (X↔SPX, Y↔SPY)		1/1

Note) (XY) and (x) have the meaning as follows:

(1) The instructions with (XY) have 4 mnemonics and 4 object codes for each. (example of LAM (XY) is given below.)

MNEMONIC	y	x	FUNCTION
LAM	0	0	
LAMX	0	1	X ↔ SPX
LAMY	1	0	Y ↔ SPY
LAMXY	1	1	X ↔ SPX, Y ↔ SPY

(2) The instructions with (x) have 2 mnemonics and 2 object codes for each. (example of LMAIY(X) is given below.)

MNEMONIC	x	FUNCTION
LMAIY	0	
LMAIYX	1	X ↔ SPX

Table 26. Arithmetic Instruction

OPERATION	MNEMONIC	OPERATION CODE	FUNCTION	STATUS	WORD/CYCLE
Add Immediate to A	AI i	1 0 1 0 0 0 i_3 i_2 i_1 i_0	A + i → A	OVF	1/1
Increment B	IB	0 0 0 1 0 0 1 1 0 0	B + 1 → B	NZ	1/1
Decrement B	DB	0 0 1 1 0 0 1 1 1 1	B − 1 → B	NB	1/1
Decimal Adjust for Addition	DAA	0 0 1 0 1 0 0 1 1 0			1/1
Decimal Adjust for Subtraction	DAS	0 0 1 0 1 0 1 0 1 0			1/1
Negate A	NEGA	0 0 0 1 1 0 0 0 0 0	$\bar{A}$ + 1 → A		1/1
Complement B	COMB	0 1 0 1 0 0 0 0 0 0	$\bar{B}$ → B		1/1
Rotate Right A with Carry	ROTR	0 0 1 0 1 0 0 0 0 0			1/1
Rotate Left A with Carry	ROTL	0 0 1 0 1 0 0 0 0 1			1/1
Set Carry	SEC	0 0 1 1 1 0 1 1 1 1	1 → CA		1/1
Reset Carry	REC	0 0 1 1 1 0 1 1 0 0	0 → CA		1/1
Test Carry	TC	0 0 0 1 1 0 1 1 1 1		CA	1/1
Add A to Memory	AM	0 0 0 0 0 0 1 0 0 0	M + A → A	OVF	1/1
Add A to Memory	AMD d	0 1 0 0 0 0 1 0 0 0 d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	M + A → A	OVF	2/2
Add A to Memory with Carry	AMC	0 0 0 0 0 1 1 0 0 0	M+A+CA→A OVF→CA	OVF	1/1
Add A to Memory with Carry	AMCD d	0 1 0 0 0 1 1 0 0 0 d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	M+A+CA→A OVF→CA	OVF	2/2
Subtract A from Memory with Carry	SMC	0 0 1 0 0 1 1 0 0 0	M−A−$\overline{CA}$→A NB→CA	NB	1/1
Subtract A from Memory with Carry	SMCD d	0 1 1 0 0 1 1 0 0 0 d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	M−A−$\overline{CA}$→A NB→CA	NB	2/2
OR A and B	OR	0 1 0 1 0 0 0 1 0 0	A ∪ B → A		1/1
AND Memory with A	ANM	0 0 1 0 0 1 1 1 0 0	A ∩ M → A	NZ	1/1
AND Memory with A	ANMD d	0 1 1 0 0 1 1 1 0 0 d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	A ∩ M → A	NZ	2/2
OR Memory with A	ORM	0 0 0 0 0 0 1 1 0 0	A ∪ M → A	NZ	1/1
OR Memory with A	ORMD d	0 1 0 0 0 0 1 1 0 0 d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	A ∪ M → A	NZ	2/2
EOR Memory with A	EORM	0 0 0 0 0 1 1 1 0 0	A ⊕ M → A	NZ	1/1
EOR Memory with A	EORMD d	0 1 0 0 0 1 1 1 0 0 d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	A ⊕ M → A	NZ	2/2

Table 27. Compare Instruction

OPERATION	MNEMONIC	OPERATION CODE	FUNCTION	STATUS	WORD/CYCLE
Immediate Not Equal to Memory	INEM i	0 0 0 0 1 0 $i_3 i_2 i_1 i_0$	$i \neq M$	NZ	1/1
Immediate Not Equal to Memory	INEMD i,d	0 1 0 0 1 0 $i_3 i_2 i_1 i_0$ $d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0$	$i \neq M$	NZ	2/2
A Not Equal to Memory	ANEM	0 0 0 0 0 0 0 1 0 0	$A \neq M$	NZ	1/1
A Not Equal to Memory	ANEMD d	0 1 0 0 0 0 0 1 0 0 $d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0$	$A \neq M$	NZ	2/2
B Not Equal to Memory	BNEM	0 0 0 1 0 0 0 1 0 0	$B \neq M$	NZ	1/1
Y Not Equal to Immediate	YNEI i	0 0 0 1 1 1 $i_3 i_2 i_1 i_0$	$Y \neq i$	NZ	1/1
Immediate Less or Equal to Memory	ILEM i	0 0 0 0 1 1 $i_3 i_2 i_1 i_0$	$i \leq M$	NB	1/1
Immediate Less or Equal to Memory	ILEMD i,d	0 1 0 0 1 1 $i_3 i_2 i_1 i_0$ $d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0$	$i \leq M$	NB	2/2
A Less or Equal to Memory	ALEM	0 0 0 0 0 1 0 1 0 0	$A \leq M$	NB	1/1
A Less or Equal to Memory	ALEMD d	0 1 0 0 0 1 0 1 0 0 $d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0$	$A \leq M$	NB	2/2
B Less or Equal to Memory	BLEM	0 0 1 1 0 0 0 1 0 0	$B \leq M$	NB	1/1
A Less or Equal to Immediate	ALEI i	1 0 1 0 1 1 $i_3 i_2 i_1 i_0$	$A \leq i$	NB	1/1

Table 28. RAM Bit Manipulation Instruction

OPERATION	MNEMONIC	OPERATION CODE	FUNCTION	STATUS	WORD/CYCLE
Set Memory Bit	SEM n	0 0 1 0 0 0 0 1 $n_1 n_0$	$1 \rightarrow M(n)$		1/1
Set Memory Bit	SEMD n,d	0 1 1 0 0 0 0 1 $n_1 n_0$ $d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0$	$1 \rightarrow M(n)$		2/2
Reset Memory Bit	REM n	0 0 1 0 0 0 1 0 $n_1 n_0$	$0 \rightarrow M(n)$		1/1
Reset Memory Bit	REMD n,d	0 1 1 0 0 0 1 0 $n_1 n_0$ $d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0$	$0 \rightarrow M(n)$		2/2
Test Memory Bit	TM n	0 0 1 0 0 0 1 1 $n_1 n_0$		M(n)	1/1
Test Memory Bit	TMD n,d	0 1 1 0 0 0 1 1 $n_1 n_0$ $d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0$		M(n)	2/2

Table 29. ROM Address Instruction

OPERATION	MNEMONIC	OPERATION CODE	FUNCTION	STATUS	WORD/CYCLE
Branch on Status 1	BR b	1 1 $b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$		1	1/1
Long Branch on Status 1	BRL u	0 1 0 1 1 1 $p_3 p_2 p_1 p_0$ $d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0$		1	2/2
Long Jump Unconditionally	JMPL u	0 1 0 1 0 1 $p_3 p_2 p_1 p_0$ $d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0$			2/2
Subroutine Jump on Status 1	CAL a	0 1 1 1 $a_5 a_4 a_3 a_2 a_1 a_0$		1	1/2
Long Subroutine Jump on Status 1	CALL u	0 1 0 1 1 0 $p_3 p_2 p_1 p_0$ $d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0$		1	2/2
Table Branch	TBR p	0 0 1 0 1 1 $p_3 p_2 p_1 p_0$			1/1
Return from Subroutine	RTN	0 0 0 0 0 1 0 0 0 0			1/3
Return from Interrupt	RTNI	0 0 0 0 0 1 0 0 0 1	$1 \rightarrow I/E$ CA RESTORE		1/3

Table 30. Input/Output Instruction

OPERATION	MNEMONIC	OPERATION CODE	FUNCTION	STATUS	WORD/CYCLE
Set Discrete I/O Latch	SED	0 0 1 1 1 0 0 1 0 0	$1 \rightarrow D(Y)$		1/1
Set Discrete I/O Latch Direct	SEDD m	1 0 1 1 1 0 $m_3 m_2 m_1 m_0$	$1 \rightarrow D(m)$		1/1
Reset Discrete I/O Latch	RED	0 0 0 1 1 0 0 1 0 0	$0 \rightarrow D(Y)$		1/1
Reset Discrete I/O Latch Direct	REDD m	1 0 0 1 1 0 $m_3 m_2 m_1 m_0$	$0 \rightarrow D(m)$		1/1
Test Discrete I/O Latch	TD	0 0 1 1 1 0 0 0 0 0		D(Y)	1/1
Test Discrete I/O Latch Direct	TDD m	1 0 1 0 1 0 $m_3 m_2 m_1 m_0$		D(m)	1/1
Load A from R-Port Register	LAR m	1 0 0 1 0 1 $m_3 m_2 m_1 m_0$	$R(m) \rightarrow A$		1/1
Load B from R-Port Register	LBR m	1 0 0 1 0 0 $m_3 m_2 m_1 m_0$	$R(m) \rightarrow B$		1/1
Load R-Port Register from A	LRA m	1 0 1 1 0 1 $m_3 m_2 m_1 m_0$	$A \rightarrow R(m)$		1/1
Load R-Port Register from B	LRB m	1 0 1 1 0 0 $m_3 m_2 m_1 m_0$	$B \rightarrow R(m)$		1/1
Pattern Generation	P p	0 1 1 0 1 1 $p_3 p_2 p_1 p_0$			1/2

Table 31. Control Instruction

OPERATION	MNEMONIC	OPERATION CODE	FUNCTION	STATUS	WORD/CYCLE
No Operation	NOP	0 0 0 0 0 0 0 0 0 0			1/1
Start Serial	STS	0 1 0 1 0 0 1 0 0 0			1/1
Stand-by Mode	SBY	0 1 0 1 0 0 1 1 0 0			1/1
Stop Mode	STOP	0 1 0 1 0 0 1 1 0 1			1/1

Table 32. Op-Code Map

R8 = 0

R9	H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	NOP	XSPX	XSPY	XSPXY	ANEM				AM				ORM			
0	1	RTN	RTNI			ALEM				AMC				EORM			
0	2	INEM i(4)															
0	3	ILEM i(4)															
0	4	LBM(XY)				BNEM				LAB				IB			
0	5	LMAIY(X)				AYY				LASPY				IY			
0	6	NEGA				RED				LASPX							TC
0	7	YNEI i(4)															
0	8	XMA(XY)				SEM n(2)				REM n(2)				TM n(2)			
0	9	LAM(XY)				LMA(XY)				SMC				ANM			
0	A	ROTR	ROTL					DAA				DAS					LAY
0	B	TBR p(4)															
0	C	XMB(XY)				BLEM				LBA							DB
0	D	LMADY(X)				SYY				LYA							DY
0	E	TD				SED				LXA				REC			SEC
0	F	LWI i(2)															
1	0	LBI i(4)															
1	1	LYI i(4)															
1	2	LXI i(4)															
1	3	LAI i(4)															
1	4	LBR m(4)															
1	5	LAR m(4)															
1	6	REDD m(4)															
1	7	LAMR m(4)															
1	8	AI i(4)															
1	9	LMIIY i(4)															
1	A	TDD m(4)															
1	B	ALEI i(4)															
1	C	LRB m(4)															
1	D	LRA m(4)															
1	E	SEDD m(4)															
1	F	XMRA m(4)															

R8 = 1

R9	H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0					ANEMD				AMD				ORMD			
0	1					ALEMD				AMCD				EORMC			
0	2	INEMD i(4)															
0	3	ILEMD i(4)															
0	4	COMB				OR				STS				SBY	STOP		
0	5	JMPL p(4)															
0	6	CALL p(4)															
0	7	BRL p(4)															
0	8	XMAD				SEMD n(2)				REMD n(2)				TMD n(2)			
0	9	LAMD				LMAD				SMCD				ANMD			
0	A	LMID i(4)															
0	B	P p(4)															
0	C–F	CAL a(6)															
1	0–F	BR b(8)															

... 1 word/2 cycle Instruction

... 1 word/3 cycle Instruction

... RAM Direct Address Instruction (2 word/2 cycle)

... 2 word/2 cycle Instruction

■ PRECAUTION TO USE THE EPROM ON-PACKAGE 4 BIT SINGLE CHIP MICROCOMPUTER

Please pay attention to the followings, since this MCU has special structure with pin socket on the package.

(1) Don't apply high static voltage or surge voltage over MAXIMUM RATINGS to the socket pins as well as the LSI pins.
If not, that may cause permanent damage to the device.

(2) When using this in production like mask ROM type single chip microcomputer, pay attention to the followings to keep the good contact between the EPROM pins and socket pins.

(a) When soldering the LSI on a print circuit board, the recommended condition is

Temperature: lower than 250°C
Time : within 10 sec.

Over time/temperature may cause the bonding solder of socket pin to melt and the socket pin may drop.

(b) Note that the detergent or coating will not get in the socket during flux washing or board coating after soldering, because that may cause bad effect on socket contact.

(c) Avoid permanent application of this under the condition of vibratory place and system.

(d) The socket, inserted and pulled repeatedly loses its contactability. It is recommended to use new one when applied in production.

Table 33 Differences among HD614P080S/HD614P0160S and HMCS402/404/408

Item \ Type name		HD614P080S	HD614P0160S	HMCS404AC	HMCS404C	HMCS404CL	HMCS402AC	HMCS402C	HMCS402CL	HMCS408C
Minimum instruction execution time		1.33 μs	1.33 μs	1.33 μs	2 μs	4 μs	1.33 μs	2 μs	4 μs	2 μs
Power supply voltage		4.5 ~ 5.5V	4.5 ~ 5.5V	4.5 ~ 6V	4 ~ 6V	2.7 ~ 6V	4.5 ~ 6V	4 ~ 6V	2.7 ~ 6V	4 ~ 6V
ROM		o 4096 words x 10 bits (using standard EPROM 2764) o 8192 words x 10 bits (using standard EPROM 27128)	16384 words x 10 bits (using standard EPROM 27256)	4,096 words x 10 bits Mask ROM			2,048 words x 10 bits Mask ROM			8192 words x 10 bits Mask ROM
RAM		576 digits x 4 bits		256 digits x 4 bits			160 digits x 4 bits			512 digits x 4 bits
I/O pin circuit	Standard pins	All pins are "without pull-up MOS (NMOS open drain)".		Each pin selects "without pull-up MOS (NMOS open drain)", "with pull-up MOS", or "CMOS".						
	High Voltage pins	All pins are "without pull-down MOS (PMOS open drain)."		Each pin selects "without pull-down MOS (PMOS open drain)" or "with pull-down MOS."						
Clock generator	Crystal resonator	Available		Available	Available	Available	Available	Available	Available	Available
	Ceramic filter resonator	Available		Available	Available	Available	Available	Available	Available	Available
	Resistance oscillator	–		–	Available	–	–	Available	–	–
Package		Shrink type 64-pin EPROM on package (DC-64SP). The base chip pins are compatible with those of the HMCS404 AC/C/CL, HMCS402AC/C/CL and HMCS408C		Shrink type 64-pin dual-in-line plastic package (DP-64S) or 64-pin flat plastic package (FP-64).						
	Type	DC-64SP		DP-64S				FP-64		
	Occupied area (mm²)	23 x 57.3		17 x 57.6				19.6 x 25.6		
	High from stand-off (mm)	7.5 (max.) EPROM on package		5.1 (max.)				2.9 (max.)		