HD63705Z0, HD637A05Z0, HD637B05Z0
ZTAT™ Micro

Preliminary
AUGUST, 1986

HITACHI

The HD63705Z0 is an 8-bit high performance CMOS single chip microcomputer unit (MCU) which includes 8k bytes of PROM and is object code compatible with the HD6305 Family.

This MCU contains a CPU, 8k bytes of PROM, 384 bytes of RAM, 8-channel A/D converter, five 16-bit timers, PWM timer, serial communication interface (SCI) and 68 parallel I/O pins.

Moreover, the HD63705Z0 provides a slave processor mode which allows to this MCU to operate as an intelligent peripheral interface adapter LSI.

■ FEATURES

<Hardware Features>

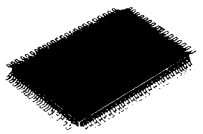
- 7744 bytes of PROM (Compatible with 27256 type)
- 384 bytes of RAM
- 8-bits X 8-channel A/D converter
- Five 16-bit timers
  - 16-bit Free Running Counter X 1 ..... Timer 1
    Input Capture Register X 1
    Output Compare Register X 1
  - 16-bit Reloadable Counter X 2 ..... Timer 2
    ..... Timer 3
- PWM (Pulse Width Modulation) timer
  Cycle: 2^{13} to 2^{16} cyc. (software programmable)
- On-chip serial communication interface (SCI)
  Asynchronous mode (8 kinds of data formats, hardware parity)
  Synchronous mode
- 68 parallel I/O pins (in single chip mode)
  45 I/O pin (including eight input data latch ports)
  15 output pins
  8 input pins
- Internal power-on reset circuit
- Direct Page Register which allows for powerful page manipulation
- A variety of external interrupts
  Three external, three timer, one parallel, one serial, one A/D and one software interrupts provided.
- A variety of operating modes

  MCU mode

Master MCU single chip mode	(Mode 2)
Slave MCU single chip mode	(Mode 6)
Master MCU external extended mode (Internal ROM is enabled.)	(Mode 1)
Master MCU external extended mode (Internal ROM is disabled.)	(Mode 5)

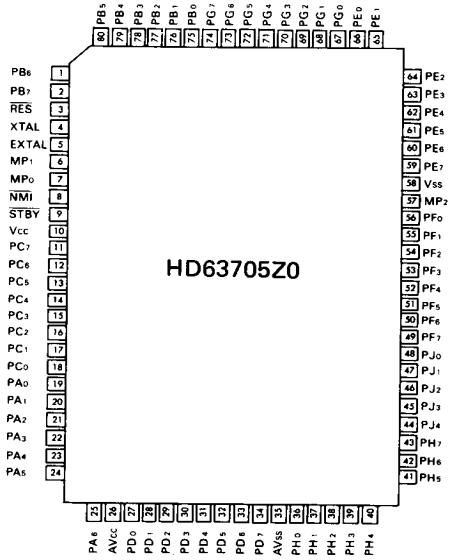
  PROM programming mode (Mode 4)
- Max. 65k bytes of address locations
- Low power modes
  - Wait mode . . . . . Internal oscillator remains active. The CPU processing is halted. The timer, serial, parallel, A/D and external interrupts are enabled.

HD63705Z0F, HD637A05Z0F, HD637B05Z0F



(FP-80)

■ PIN ARRANGEMENT



(Top View)

– Stop mode . . . . . Internal oscillator is halted. The data of registers and RAM remain unaltered. All I/O lines remain unchanged. External interrupts are enabled.

– Standby mode . . Internal oscillator is halted. The data of RAM is unchanged. The MCU internal state is reset.

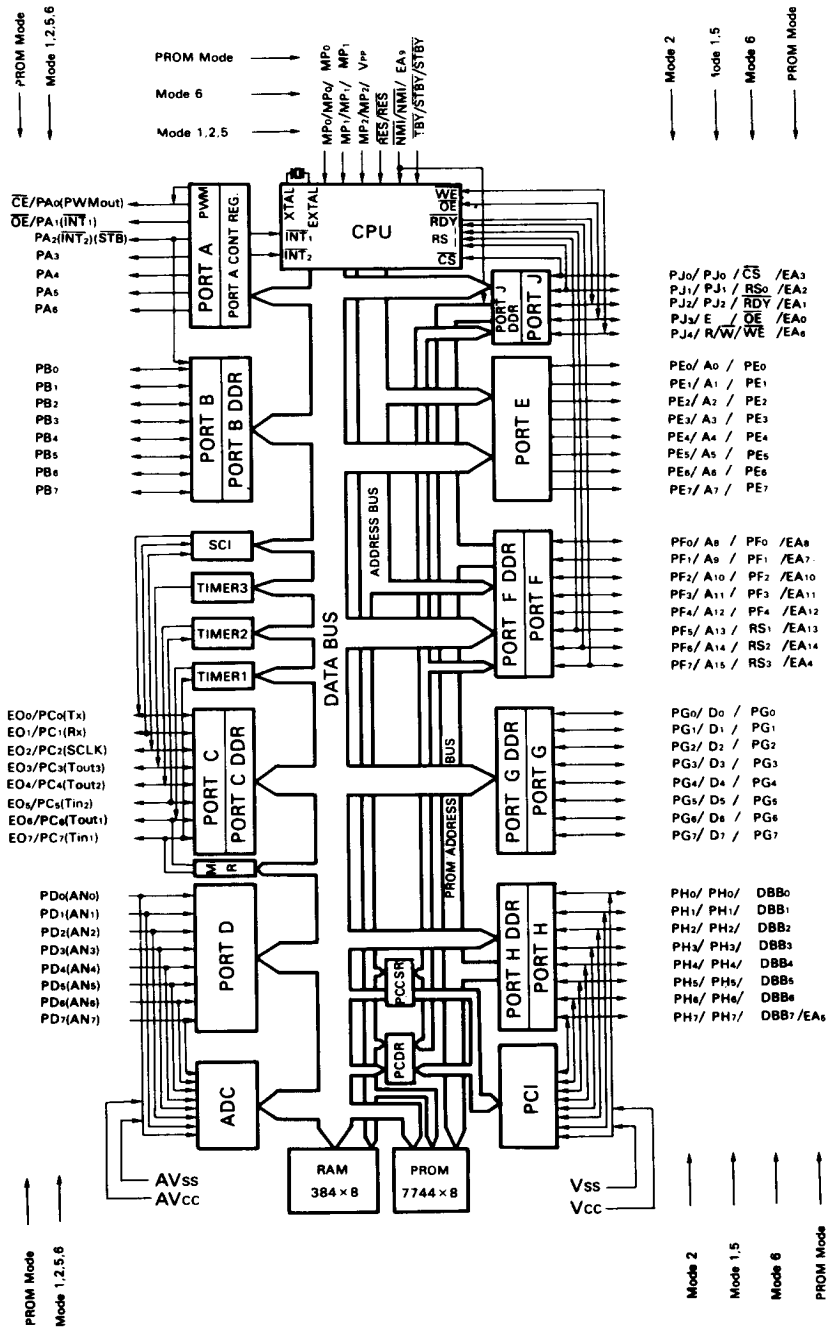
- Minimum instruction cycle time . . . . . 0.5 μs (f = 2 MHz)
- Operating range

V_{CC} = 5V ± 10%	f = 0.1 to 1.0 MHz	:	HD63705Z0
	f = 0.1 to 1.5 MHz	:	HD637A05Z0
	f = 0.1 to 2.0 MHz	:	HD637B05Z0

<Software Features>

- Upward software compatible with the HD6305 family MCUs (Multiply instruction (MUL) is added.)
- Byte efficient instruction set
- Powerful bit manipulation instructions
- Bit test and branch instructions
- Versatile interrupt handling
- Powerful indexed addressing for tables
- Full set of conditional branches
- 10 powerful addressing modes

■ BLOCK DIAGRAM



HITACHI

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit	Note
Supply Voltage	V_{CC}, AV_{CC}	−0.3 ~ +7.0	V	
Programming Voltage	V_{PP}	−0.3 ~ +15.0	V	1
Input Voltage	V_{in}	−0.3 ~ V_{CC} +0.3	V	2
Operating Temperature	T_{opr}	0 ~ +70	°C	
Storage Temperature	T_{stg}	−55 ~ +125	°C	

(NOTE 1) Applies to MP_2 (V_{PP}) pin.

(NOTE 2) Applies to all other pins except for MP_2 (V_{PP}).

(Precaution)
These devices contain circuits to protect the inputs against high static voltages or high electric fields. Be careful not to apply any voltage higher than the absolute maximum ratings to these high input impedance circuits. For normal operation, we recommend the Vin and Vout be constrained to the range $V_{SS} \leqq$ (Vin or Vout) $\leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS IN MCU MODE

• D.C. Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND, T_a = 0 to +70°C, unless otherwise noted.

Item		Symbol	Test Condition	min	typ	max	Unit
Input High Voltage	$\overline{RES}$, $\overline{STBY}$,	V_{IH}		V_{CC} − 0.5	—	V_{CC} + 0.3	V
	EXTAL			V_{CC} × 0.7	—	V_{CC} + 0.3	
	Port D			2.0		AV_{CC} + 0.3	
	SCLK, INT2 (STB)			V_{CC} × 0.8	—	V_{CC} + 0.3	
	PCI Inputs*****			2.4	—	V_{CC} + 0.3	
	All Other Inputs			2.0	—	V_{CC} + 0.3	
Input Low Voltage	PCI Inputs*****	V_{IL}		−0.3	—	0.6	V
	All Other Inputs			−0.3	—	0.8	
Input Leakage Current	$\overline{NMI}$, $\overline{RES}$, $\overline{STBY}$	$\|I_{in}\|$	V_{in} = 0.5 to V_{CC} −0.5V	—	—	1.0	μA
	MP_2 (V_{PP})			—	1.0	100	
Three State Leakage Current	Ports A*, B, C, G, E**, F, H, J, MP_0, MP_1	$\|I_{TSI}\|$	V_{in} = 0.5 to V_{CC} − 0.5V	—	—	1.0	
Output High Voltage	All Outputs but PJ_2	V_{OH}	I_{OH} = −200 μA	2.4	—	—	V
			I_{OH} = −10 μA	V_{CC} − 0.7	—	—	V
Output Low Voltage	All Outputs Including PJ_2****	V_{OL}	I_{OL} = 1.6 mA	—	—	0.4	V
	Port E		I_{OL} = 10 mA	—	—	1.0	V
Input Capacitance	MP_2 (V_{PP})	C_{in}	V_{in} = 0V, f = 1MHz, T_a = 25°C	—	—	100	pF
	All Other Except for MP_2 (V_{PP})			—	—	12.5	
Current Dissipation**	Active	I_{CC}***	f = 1MHz***	—	6	12	mA
	Wait	I_{WAIT}***		—	2	5	mA
	Stop	I_{STOP}		—	3.0	15.0	μA
	Standby	I_{STBY}		—	3.0	15.0	μA
RAM Standby Voltage		V_{RAM}		2.0	—	—	V

* Only in the standby mode.
** V_{IH} min = V_{CC} − 1.0V, V_{IL} max. = 0.8V and no line is connected to any output or $\overline{RES}$ pin.
*** The value at f = XMHz is given by I_{CC} (f = XMHz) = I_{CC} (f = 1MHz, XX.
**** PJ_2 is an NMOS open drain pin.
***** RSO ~ 3, DBBO ~ 7, $\overline{CS}$, $\overline{OE}$ $\overline{WE}$.

- **AC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND, Ta = 0 to +70°C, unless otherwise noted)**

Bus Timing

Item	Symbol	Test Condition	HD63705Z0			HD637A05Z0			HD637B05Z0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Cycle Time	t_{cyc}	Fig. 1	1.0	–	10	0.666	–	10	0.5	–	10	μs
System Clock Rise Time	t_{Er}		–	–	25	–	–	25	–	–	25	ns
System Clock Fall Time	t_{Ef}		–	–	25	–	–	25	–	–	25	ns
System Clock Pulse Width (High)*	PW_{EH}		450	–	–	300	–	–	220	–	–	ns
System Clock Pulse Width (Low)*	PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address, R/$\overline{W}$ Delay Time*	t_{AD}		–	–	250	–	–	190	–	–	160	ns
Data Delay Time (Write)	t_{DDW}		–	–	200	–	–	160	–	–	120	ns
Data Setup Time (Read)	t_{DSR}		80	–	–	70	–	–	70	–	–	ns
Address, R/$\overline{W}$ Hold Time*	t_{AH}		40	–	–	30	–	–	20	–	–	ns
Data Hold Time (Write)*	t_{HW}		40	–	–	30	–	–	20	–	–	ns
Data Hold Time (Read)	t_{HR}		0	–	–	0	–	–	0	–	–	ns

* Bus timing varies with t_{cyc}. The above characteristics are obtained when t_{cyc} is a minimum (i.e. during highest speed operation).

Peripheral Port Timing

Item		Symbol	Test Condition	HD63705Z0			HD637A05Z0			HD637B05Z0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Peripheral Data Setup Time	Port B, C, D, F, G, H, J	t_{PDSU}	Fig. 2	200	–	–	200	–	–	200	–	–	ns
Peripheral Data Hold Time	Port B, C, D F, G, H, J	t_{PDH}		200	–	–	200	–	–	200	–	–	ns
Delay Time *	Port A, B, C E, F, G, H, J	t_{PWD}	Fig. 3	–	–	300	–	–	300	–	–	300	ns
Input Strobe Pulse Width		t_{PWIS}	Fig. 4	300	–	–	300	–	–	300	–	–	ns
Input Data Hold Time	Port B	t_{IH}		200	–	–	200	–	–	200	–	–	ns
Input Data Setup Time	Port B	t_{IS}		100	–	–	100	–	–	100	–	–	ns

* Enable negative transition to peripheral data valid.

- **Control Signal Timing (V_{CC} = 5.0V ± 10%, V_{SS} = GND, Ta = 0 to +70°C unless otherwise noted)**

Item	Symbol	Test Condition	HD63705Z0			HD637A05Z0			HD637B05Z0			Unit
			min	typ	max	min	typ	max	min	typ	max	
$\overline{NMI}$ Pulse Width	t_{IWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{INT_1}$, $\overline{INT_2}$ ($\overline{STB}$) Pulse Width	T_{IWL2}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{RES}$ Pulse Width	t_{RWL}		5	–	–	5	–	–	5	–	–	t_{cyc}
Control Setup Time	t_{CS}	Fig. 10, 11	250	–	–	250	–	–	250	–	–	ns
Oscillation Start Time (Crystal)	t_{OSC}	Fig. 11*	–	–	20	–	–	20	–	–	20	ms
Reset Delay Time	t_{RHL}	**	80	–	–	80	–	–	80	–	–	ms

* C_L = 22 pF ± 20%, R_S = 60Ω max.
** External capacitance is 2.2 μF.

- **PCI Timing**

Read Cycle

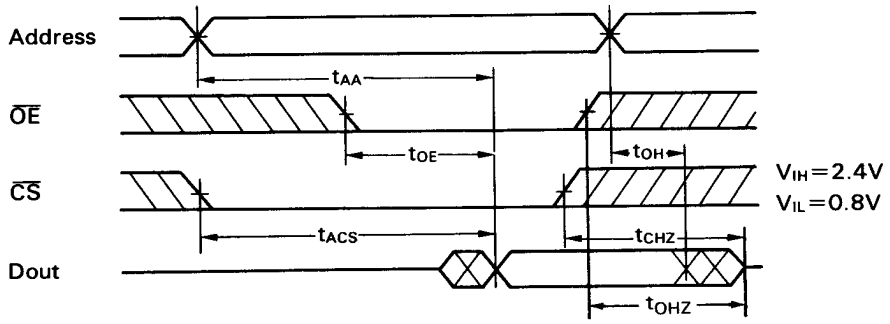
Item	Symbol	HD63705Z0		HD637A05Z0		HD637B05Z0		Unit
		min	max	min	max	min	max	
Address Access Time	t_{AA}	—	300	—	250	—	200	ns
Chip Select Access Time	t_{ACS}	—	300	—	250	—	200	ns
Output Enable Output Delay Time	t_{OE}	—	150	—	120	—	110	ns
$\overline{CS}$ Output Floating	t_{CHZ}	—	150	—	120	—	110	ns
Output Hold Time	t_{OH}	0	—	0	—	0	—	ns
$\overline{OE}$ Output Floating	t_{OHZ}	0	150	0	120	0	110	ns

Write Cycle

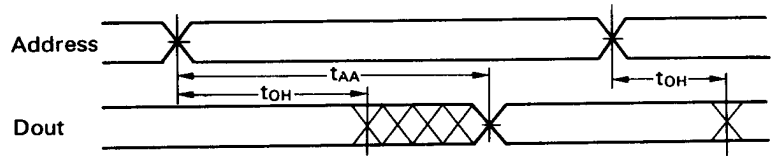
Item	Symbol	HD63705Z0		HD637A05Z0		HD637B05Z0		Unit
		min	max	min	max	min	max	
Chip Select Time	t_{CW}	200	—	150	—	120	—	ns
Address Valid Time	t_{AW}	200	—	150	—	120	—	ns
Address Setup Time	t_{AS}	20	—	20	—	20	—	ns
Write Pulse Width	t_{WP}	150	—	120	—	100	—	ns
Address Hold Time	t_{WR}	20	—	20	—	20	—	ns
Input Data Set Time	t_{DW}	120	—	100	—	80	—	ns
Input Data Hold Time	t_{DH}	20	—	20	—	20	—	ns

Timing Waveform

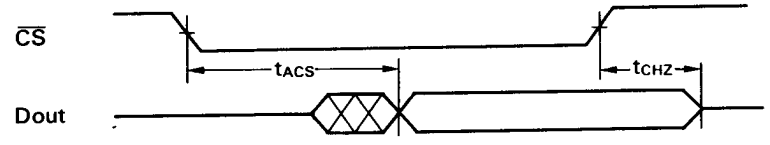
- **Read Cycle (1) *1**



- **Read Cycle (2) *1, 2, 4**

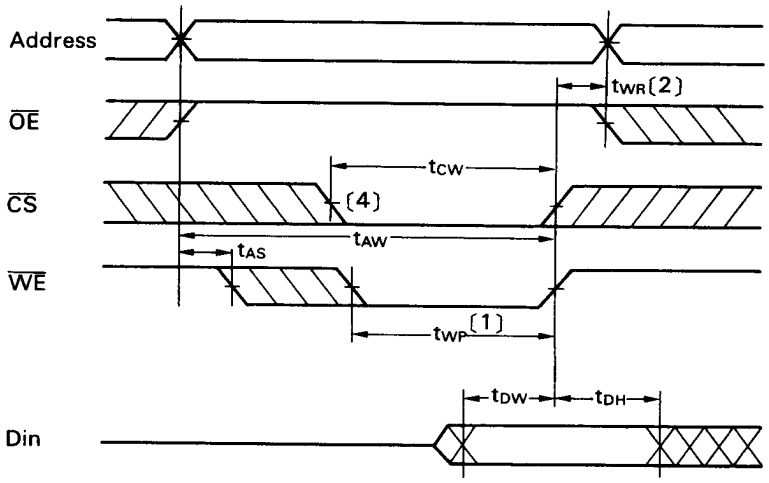


- **Read Cycle (3) *1, 3, 4**



(NOTES) 1. $\overline{WE}$ is high for a Read Cycle.
2. $\overline{CS} = V_{IL}$
3. Addresses valid prior to or coincident with $\overline{CS}$ negative transition.
4. $\overline{OE} = V_{IL}$

- **Write Cycle*1**



(NOTES) 1. A write occurs during the overlap of a low $\overline{CS}$ and a low $\overline{WE}$ (t_{WP}).
2. t_{WR} is measured from the earlier of $\overline{CS}$ or $\overline{WE}$ going high to the end of write cycle.
3. If I/O pins are in the output state, input signal with opposite phase to the outputs must not be applied to them.
4. If the $\overline{CS}$ negative transition occurs simultaneously with the $\overline{WE}$ negative transition, the output remains in a high impedance state.

- **Timer and SCI Timing**

Item		Symbol	Test Condition	HD63705Z0			HD637A05Z0			HD637B05Z0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Timer 1 Input Pulse Width		t_{PWT}	Fig. 8	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Delay Time (Enable Positive Transition to Timer Data Valid)		t_{TOD}	Fig. 6, 7	–	–	400	–	–	400	–	–	400	ns
SCI Input Clock Cycle	Asynchronous Mode	t_{Scyc}	Fig. 8	1.0	–	–	1.0	–	–	1.0	–	–	t_{cyc}
	Synchronous Mode		Fig. 5	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
SCI Transmit Data Delay Time (Synchronous Mode)		t_{TXD}	Fig. 5	–	–	200	–	–	200	–	–	200	ns
SCI Receive Data Setup Time (Synchronous Mode)		t_{SRX}		260	–	–	260	–	–	260	–	–	ns
SCI Receive Data Hold Time (Synchronous Mode)		t_{HRX}		100	–	–	100	–	–	100	–	–	ns
SCI Input Clock Pulse Width		t_{PWSCK}	Fig. 8	0.4	–	0.6	0.4	–	0.6	0.4	–	0.6	t_{Scyc}
Timer 2 Input Clock Cycle		t_{tcyc}	Fig. 8	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Timer 2 Input Clock Pulse Width		t_{PWTCK}	Fig. 8	200	–	–	200	–	–	200	–	–	ns
Timer 1, Timer 2 and SCI Input Clock Rise Time		t_{CKr}		–	–	100	–	–	100	–	–	100	ns
Timer 1, Timer 2 and SCI Input Clock Fall Time		t_{CKf}		–	–	100	–	–	100	–	–	100	ns

- **A/D Converter Characteristics (V_{CC} = 5.0 ± 10%, V_{SS} = AV_{SS} = GND, Ta = 0 to +70°C, unless otherwise noted)**

Item	Symbol	Test Condition	HD63705Z0			HD637A05Z0			HD637B05Z0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Analog Power Supply Voltage	AV_{CC}		4.5	–	5.5	4.5	–	5.5	4.5	–	5.5	V
Analog Input Voltage	AV_{in}		AV_{SS}	–	AV_{CC}	AV_{SS}	–	AV_{CC}	AV_{SS}	–	AV_{CC}	V
Current between AV_{CC} and AV_{SS}	I_{AD}	Operation			1.0			1.0			1.0	mA
		STBY, STOP			250			250			250	μA
Resolution			–	8	–	–	8	–	–	8	–	bit
Conversion Time			34	–	–	34	–	–	34	–	–	μs
Input Count			0	–	8	0	–	8	0	–	8	Channel
Absolute Error		Ta=25°C	–	–	1.5	–	–	1.5	–	–	1.5	LSB

(NOTE) AV_{CC} = V_{CC} ± 0.3V

HITACHI

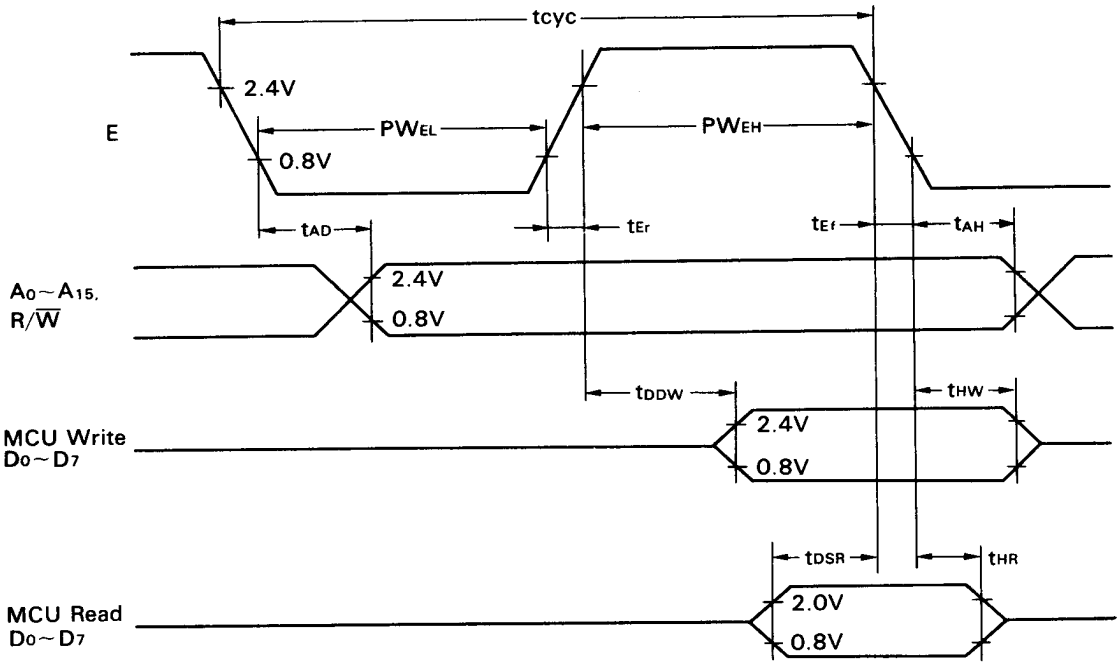


Fig. 1 External Extended Modes Bus Timing

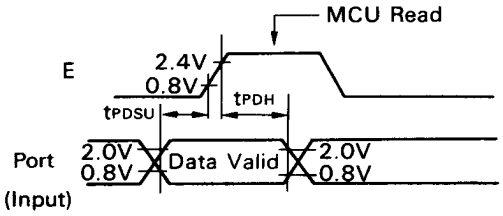


Fig. 2 Peripheral Data Setup and Hold Times (MCU Read)

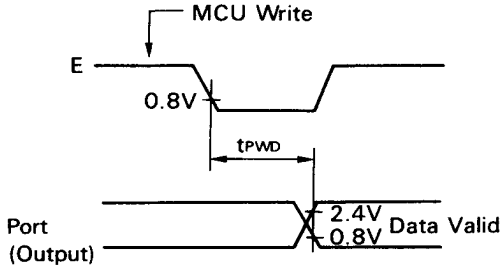


Fig. 3 Peripheral Data Delay Time (MCU Read)

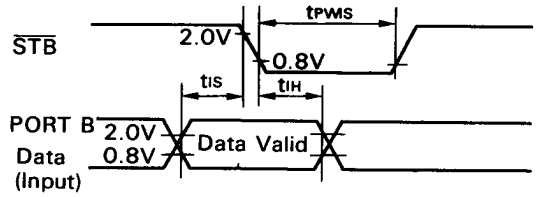
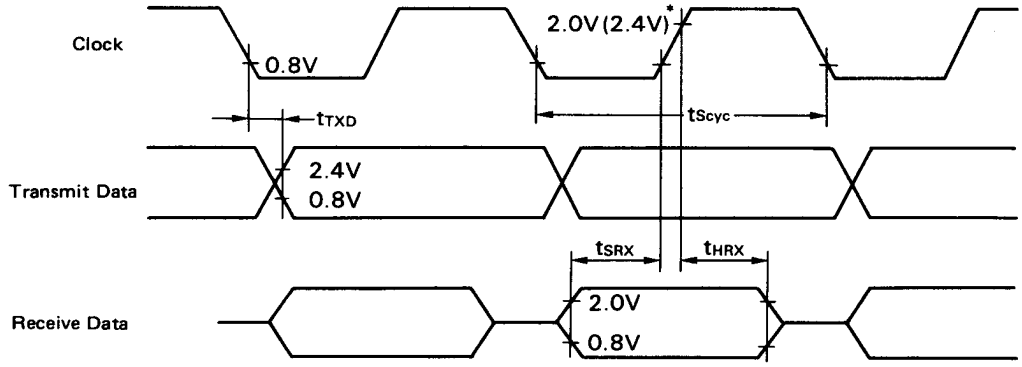


Fig. 4 Port B Input Latch Timing



* 2.0V is high level when clock input.
2.4V is high level when clock output.

Fig. 5 SCI Synchronous Timing

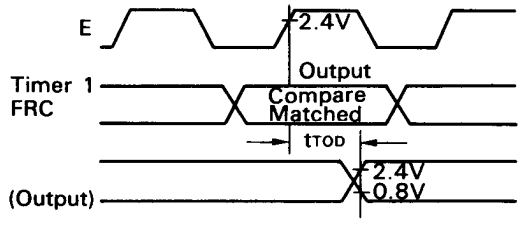
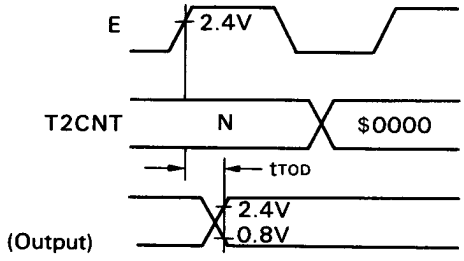


Fig. 6 Timer 1 Output Timing



(TCONR = N)

Fig. 7 Timer 2 and Timer 3 Output Timing

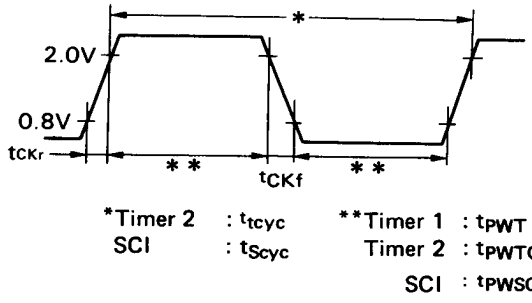


Fig. 8 Timer 1, Timer 2 and SCI Input Clock Timing

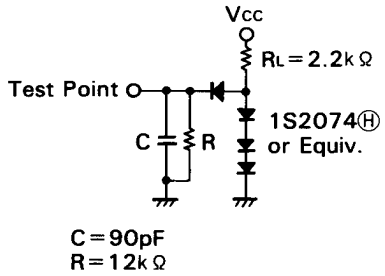


Fig. 9 Bus Timing Test Loads (TTL Load)

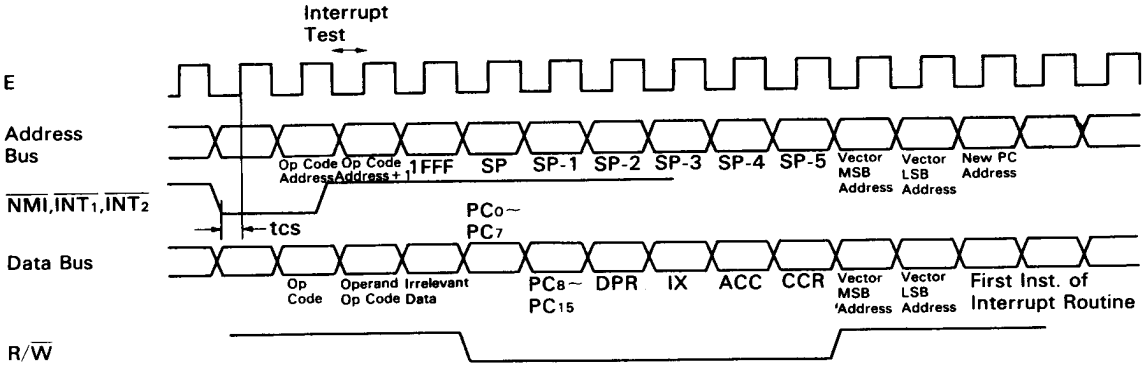


Fig. 10 Interrupt Sequence

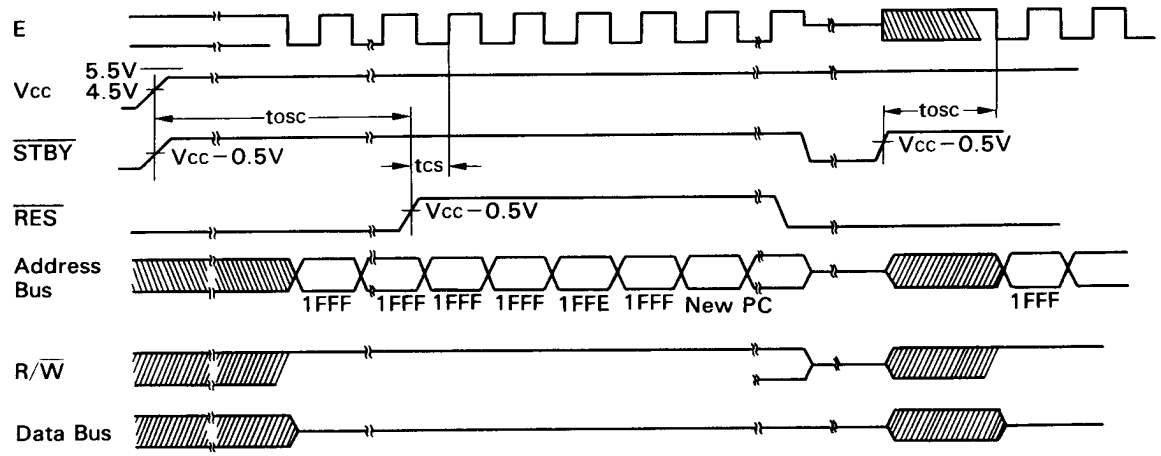


Fig. 11 Reset Timing

■ FUNCTIONAL PIN DESCRIPTION

Of the 80 MCU pins, some function depending on the operating modes. For details, see Table 1.

Mode Independent Pins

The following pins function identically in all modes.

- **V_{CC}, V_{SS}**

Power is supplied to a large portion of the MCU using the V_{CC} and V_{SS} pins. The power supply should provide +5V±10% for V_{CC}.

- **AV_{CC}, AV_{SS}**

These are power supply pins for A/D converter, which can be used separately from V_{SS} and V_{CC} to enhance the precision of A/D conversion.

- **XTAL, EXTAL**

These two input pins are connected to an AT-cut parallel crystal resonator to drive the MCU internal oscillator. Divided-by-four circuitry is included which allows use of the 4 MHz crystal resonator for obtaining a system clock of 1 MHz.

Alternatively, EXTAL can be driven by an external clock with a duty cycle of 45 to 55% with no line connected to XTAL, when a system clock with a fourth frequency of the external clock is generated. The external clock frequency should be less than four times of the maximum operating frequency.

- **$\overline{STBY}$**

This pin is used to put the MCU in the standby mode or PROM programming mode.

Logic low on the $\overline{STBY}$ pin allows the MCU to enter the standby mode. In this mode the internal oscillator is turned off, the internal clock is halted, and the MCU internal state is reset. Writing a 0 to the RAM enable bit (RAME) disables RAM, which allows data in the RAM to be unchanged in the standby mode. RAME is bit 4 of the Miscellaneous Register at $003B. For detailed information about the standby mode, see 'LOW POWER MODES'.

When applying logic lows to the $\overline{STBY}$ pin and mode programming pins MP_0 and MP_1 and applying a logic high to MP_2, the MCU enters the PROM programming mode. For details, see 'PROGRAMMING THE PROM'.

- **RESET ($\overline{RES}$)**

The $\overline{RES}$ pin is used to reset the MCU to provide a startup from a power down condition. On power-up, the $\overline{RES}$ must stay low for minimum of 20 ms.

As the CPU registers (Accumulator, Index Register, Stack Pointer, Condition Code Register except for interrupt mask bit), RAM, port data registers are not initialized at reset, their contents are indeterminate after the startup procedure. To reset the MCU during the operation, the $\overline{RES}$ should be held low for at least five system clock cycles. At the fifth cycle, all address buses go high.

The address buses remain high as long as the $\overline{RES}$ stays low. When applying a logic high to the $\overline{RES}$, the MCU operates as follows:

1) Latches the data on the mode programming pins MP_0, MP_1 and MP_2. (In the PROM programming mode, the $\overline{RES}$ is not used.)
2) Initializes the internal registers. (See Table 7).
3) Sets the interrupt mask bit in the Condition Code Register. For the CPU to recognize the maskable interrupts ($\overline{INT_1}$, $\overline{INT_2}$, and Timer 1, Timer 2, Timer 3, PCI, SCI and A/D interrupts), this bit must be cleared previously.
4) Puts the contents of reset vector addresses ($1FFE, $1FFF) into the program counter, then resumes the program execution.

After power up to the oscillator stabilization (for 20 ms max.), $\overline{RES}$ input is not latched to the MCU, which causes the MCU internal state and I/O pin condition to be undefined.

MCU Mode Pins

The following pins function in the MCU modes (Modes 1, 2, 5, 6).

- **Non Maskable Interrupt ($\overline{NMI}$)**

An $\overline{NMI}$ negative edge signifies an interrupt request, but the current instruction will be completed before the CPU responds to the request. $\overline{NMI}$ is not affected by the interrupt mask bit in the Condition Code Register.

When the interrupt occurs, the contents of Program Counter, Index Register, Accumulator and Condition Code Register are pushed onto the stack. After the registers have been stacked, a vector is fetched from $1FFC and $1FFD, transferred to the Program Counter, then the non-maskable interrupt routine is executed.

- **Interrupts ($\overline{INT_1}$, $\overline{INT_2}$)**

$\overline{INT_1}$ is also used as the Port A bit 1 pin (PA_1). This pin can be programmed as an $\overline{INT_1}$ input by writing a 1 to bit 1 in the Port A Control and Status Register at $000F. $\overline{INT_1}$ is sensed on either a negative edge or a level of the input signal. Writing a 0 to bit 0 of the Port A Control and Status Register selects edge-sensitive triggering, while writing a 1 selects level-sensitive triggering. The Port A Control and Status Register is cleared to $00 during reset.

When $\overline{INT_1}$ or $\overline{INT_2}$ interrupt request is acknowledged, the CPU will complete the current instruction before it responds to the request. If the interrupt mask bit in the Condition Code Register is cleared, the CPU begins an interrupt sequence at the end of the current instruction; if set, the CPU cannot accept the interrupt request. When the interrupt occurs, the contents of Program Counter, Direct page register, Index Register, Accumulator and Condition Code Register are pushed onto the stack, and then the interrupt mask bit is set which inhibits any additional maskable interrupts. Finally a vector is fetched, transferred to the Program Counter, and the CPU starts the interrupt service routine.

$\overline{INT_2}$ ($\overline{STB}$) pin is also used as PA_2. Writing a 1 to bit 2 in the Port A Control and Status Register allows this pin to be used for the $\overline{INT_2}$ input, and writing 1's to both bit 2 and bit 3 allows the pin to be used for $\overline{STB}$ input. After sensing the $\overline{INT_2}$ interrupt request, the MCU processes the interrupt in the same way as the $\overline{INT_1}$ interrupt. For details, see 'INTERRUPT'.

- **Mode Programming Pins (MP_0, MP_1, MP_2)**

These pins are used to define the HD63705Z0 operating mode. For details, see 'MODE SELECTION'.

The following pins function in the external extended modes (Mode 1 and Mode 5).

- **Enable (E: PJ_3)**

This pin provides the system clock for external circuits. This clock is single phase and TTL compatible. Its frequency is one fourth that of the crystal oscillator or external clock.

- **Read/Write (R/$\overline{W}$: PJ_4)**

The Read/Write signal on this pin is used to control the direction of transfers on the external data bus. A low level on the R/$\overline{W}$ pin enables the data to be written to an external device. A high level on the R/$\overline{W}$ pin enables the data to be read from an external device. Normally this pin is in the read state.

The following pins are used in the slave MCU modes (Mode 6).

- **Chip Select ($\overline{CS}$: PJ_0)**

This pin allows the HD63705Z0 to provide the parallel communication with the master MCU. A low level on this pin activates the data bus.

- **Register Select (RS_0 – RS_3 : PJ_1, PF_5 to PF_7)**

RS_0 to RS_3 inputs are used to select either the Parallel Communication Control and Status Register ($002B) or one of the parallel communication data registers (PCDR0: $002C ~ PCDR14: $003A).

- **Ready ($\overline{RDY}$: PJ_2)**

The $\overline{RDY}$ output applies an interrupt to the master MCU when it is pulled low.

- **Output Enable ($\overline{OE}$: PJ_3)**

When a logic low is applied to the $\overline{OE}$ pin, the HD63705Z0 MCU outputs data on the master MCU data bus lines.

- **Write Enable ($\overline{WE}$: PJ_4)**

When a logic low is applied to the $\overline{WE}$ pin, the master MCU writes data to the HD63705Z0 MCU via data bus lines.

PROM Programming Mode Pins

The following pins are used in the PROM programming mode (Mode 4).

- **Chip Enable ($\overline{CE}$: PA_0)**

When a logic low is applied to the $\overline{CE}$ pin, the HD63705Z0 PROM is enabled to be programmed and verified. A high level on this pin disallows PROM programming and verification.

- **Output Enable ($\overline{OE}$: PA_1)**

This input pin controls the PROM data output. A low level on the $\overline{OE}$ pin enables programmed PROM data to be output.

- **Programming Voltage (V_{PP} : MP_2)**

The PROM can be programmed while the programming

voltage is applied to this V_{PP} pin. The PROM address is input via $\overline{NMI}$, Port F, Port H and Port J (EA_0 to EA_{14}). The programmed PROM data is output from Port C (EO_0 to EO_7).

For details, see 'PROGRAMMING THE PROM'.

I/O Ports

The HD63705Z0 provides nine I/O ports: seven 8-bit ports, a 7-bit port and a 5-bit port.

- **Port A (PA_0 to PA_6)**

This is a 7-bit output only port. Of the seven output lines, three lines can be used also for other purposes.

Port A bit 0 (PA_0) can also be utilized as the output of PWM timer. It functions as $\overline{CE}$ input pin in the PROM programming mode (Mode 4).

PA_1 can also be used for $\overline{INT_1}$ input. It is configured as $\overline{OE}$ input in the PROM programming mode (Mode 4).

PA_2 can be also used for $\overline{INT_2}$ ($\overline{STB}$) input.

- **Port B (PB_0 to PB_7)**

This is an 8-bit I/O port. It can also be used as an input data latch port.

- **Port C (PC_0 to PC_7)**

This is an 8-bit I/O port. PC_0 to PC_2 can also be used as SCI I/O lines. PC_3 to PC_7 can also function as I/O pins of Timer 1, Timer 2 and Timer 3. In the PROM programming mode (Mode 4) Port C provides bidirectional PROM data lines (EO_0 to EO_7).

- **Port D (PD_0 to PD_7)**

This is an 8-bit input only port. It can also provide A/D channel input lines (AN_0 to AN_7).

- **Port E (PE_0 to PE_7)**

This is an 8-bit output only port. It is a high-current drive port which produces 10 mA as I_{OL} when V_{OL} = 1.0V. In the external extended modes (Mode 1 and Mode 5) Port E provides the eight low-order address bus output lines (ADR_0 to ADR_7).

- **Port F (PF_0 to PF_7)**

This is an 8-bit I/O port. In the external extended modes (Mode 1 and Mode 5) Port F provides the eight high-order address output lines (ADR_8 to ADR_{15}). In the slave MCU mode (Mode 6), PF_5 to PF_7 function as register select pins (RS_1 to RS_3). In the PROM programming mode (Mode 4). Port F provides PROM address input lines (EA_4, EA_7, EA_8, EA_{10} to EA_{14}).

- **Port G (PG_0 to PG_7)**

This is an 8-bit I/O port. In the external extended modes (Mode 1 and Mode 5) Port G provides eight data lines ($DATA_0$ to $DATA_7$).

- **Port H (PH_0 to PH_7)**

This is an 8-bit I/O port. In the slave MCU mode (Mode 6), Port H functions as 8-bit bidirectional data bus between the HD63705Z0 MCU and the master MCU (DBB_0 to DBB_7).

- **Port J (PJ_0 to PJ_4)**

This is a 5-bit I/O port. In the master MCU external extended modes (Mode 1 and Mode 5), PJ_3 and PJ_4 function as E and R/$\overline{W}$ outputs respectively. In the slave MCU mode (Mode 6), PJ_0, PJ_1, PJ_2, PJ_3 and PJ_4 are used as $\overline{CS}$ input, RS_0 input, $\overline{RDY}$ output, $\overline{OE}$ input and $\overline{WE}$ input respectively. In the PROM programming mode (Mode 4), Port J provides PROM address input lines (EA_0 to EA_3, EA_6).

For detailed information about these ports, see 'I/O PORT'.

■ MODE SELECTION

The HD63705Z0 is capable of operating in five modes; four MCU modes and a PROM mode. The following is a description of individual modes.

MCU Modes

- **Mode 1 (Master MCU External Extended Mode)**

In this mode Port G provides data lines, Port E provides the eight low-order address bus lines, and Port F provides the eight high-order address bus lines. The HD63705Z0 provides direct interface with the HMCS6800 via these lines. To control the interface, R/$\overline{W}$ and E signals are output from PJ_4 and PJ_3 respectively. In this mode, the internal PROM is enabled which allows the MCU to address up to 56k bytes external memory space.

- **Mode 2 (Master MCU Single Chip Mode)**

In this mode all of the nine ports can be used.

- **Mode 5 (Master MCU External Extended Mode)**

This mode is, like Mode 1, an external extended mode. In this mode, however, the internal PROM is disabled which allows the MCU to address up to 65k byte external memory space.

- **Mode 6 (Slave MCU Single Chip Mode)**

In this mode the HD63705Z0 functions as an intelligent peripheral interface adapter LSI. The device provides a parallel communication interface with the master MCU via Port H. Port F and Port J are used to select a parallel communication register and control the parallel communication. The internal PROM is enabled which disallows the MCU to address external memory space. This mode provides 52 available port lines (all lines of Port A, B, C, D, E, G and PF_0 to PF_4).

PROM Programming Mode (Mode 4)

This mode allows the PROM to be programmed. For details see 'PROGRAMMING THE PROM'.

Table 6 and Fig. 12 shows MCU port condition in each mode and memory map in each mode respectively.

■ PIN FUNCTION SUMMARY

For detail, see 'FUNCTIONAL PIN DESCRIPTION'.

The condition of the pins depends on the operating mode.

Table 1 Pin Function Summary

No.	Mode 1, 5	Mode 2	Mode 6	Mode 4
1	Port B_6 (I/O)	←	←	
2	Port B_7 (I/O)	←	←	
3	$\overline{RES}$	←	←	
4	XTAL	←	←	
5	EXTAL	←	←	
6	MP_1 (I)	←	←	←
7	MP_0 (I)	←	←	←
8	$\overline{NMI}$	←	←	EA_9 (I)
9	$\overline{STBY}$	←	←	←
10	V_{CC}	←	←	←
11	Port C_7 (I/O)/Tin_1	←	←	EO_7 (I/O)
12	Port C_6 (I/O)$Tout_1$	←	←	EO_6 (I/O)
13	Port C_5 (I/O)/Tin_2	←	←	EO_5 (I/O)
14	Port C_4 (I/O)/$Tout_2$	←	←	EO_4 (I/O)
15	Port C_3 (I/O)/$Tout_3$	←	←	EO_3 (I/O)
16	Port C_2 (I/O)/SCLK	←	←	EO_2 (I/O)
17	Port C_1 (I/O)/Rx	←	←	EO_1 (I/O)
18	Port C_0 (I/O)/Tx	←	←	EO_0 (I/O)
19	Port A_0 (O)/PWMout	←	←	$\overline{CE}$ (I)
20	Port A_1 (O)/$\overline{INT_1}$	←	←	$\overline{OE}$ (I)
21	Port A_2 (O)/$\overline{INT_2}$ ($\overline{STB}$)	←	←	
22	Port A_3 (O)	←	←	
23	Port A_4 (O)	←	←	
24	Port A_5 (O)	←	←	
25	Port A_6 (O)	←	←	
26	AV_{CC}	←	←	
27	Port D_0 (I)/AN_0	←	←	
28	Port D_1 (I)/AN_1	←	←	
29	Port D_2 (I)/AN_2	←	←	
30	Port D_3 (I)/AN_3	←	←	
31	Port D_4 (I)/AN_4	←	←	
32	Port D_5 (I)/AN_5	←	←	
33	Port D_6 (I)/AN_6	←	←	
34	Port D_7 (I)/AN_7	←	←	
35	AV_{SS}	←	←	
36	Port H_0 (I/O)	←	DBB_0 (I/O)	
37	Port H_1 (I/O)	←	DBB_1 (I/O)	
38	Port H_2 (I/O)	←	DBB_2 (I/O)	
39	Port H_3 (I/O)	←	DBB_3 (I/O)	
40	Port H_4 (I/O)	←	DBB_4 (I/O)	

(To be continued)

No.	Mode 1, 5	Mode 2	Mode 6	Mode 4
41	Port H_5 (I/O)	←	DBB_5 (I/O)	
42	Port H_6 (I/O)	←	DBB_6 (I/O)	
43	Port H_7 (I/O)	←	DBB_7 (I/O)	EA_5 (I)
44	R/$\overline{W}$ (O)	Port J_4 (I/O)	$\overline{WE}$ (I)	EA_6 (I)
45	E (O)	Port J_3 (I/O)	$\overline{OE}$ (I)	EA_0 (I)
46	Port J_2 (I/O)	←	$\overline{RDY}$ (O)	EA_1 (I)
47	Port J_1 (I/O)	←	RS_0 (I)	EA_2 (I)
48	Port J_0 (I/O)	←	$\overline{CS}$ (I)	EA_3 (I)
49	ADR_{15} (O)	Port F_7 (I/O)	RS_3 (I)	EA_4 (I)
50	ADR_{14} (O)	Port F_6 (I/O)	RS_2 (I)	EA_{14} (I)
51	ADR_{13} (O)	Port F_5 (I/O)	RS_1 (I)	EA_{13} (I)
52	ADR_{12} (O)	Port F_4 (I/O)	Port F_4 (I/O)	EA_{12} (I)
53	ADR_{11} (O)	Port F_3 (I/O)	Port F_3 (I/O)	EA_{11} (I)
54	ADR_{10} (O)	Port F_2 (I/O)	Port F_2 (I/O)	EA_{10} (I)
55	ADR_9 (O)	Port F_1 (I/O)	Port F_1 (I/O)	EA_7 (I)
56	ADR_8 (O)	Port F_0 (I/O)	Port F_0 (I/O)	EA_8 (I)
57	MP_2	←	←	V_{PP}
58	V_{SS}	←	←	←
59	ADR_7 (O)	Port E_7 (O)	Port E_7 (O)	
60	ADR_6 (O)	Port E_6 (O)	Port E_6 (O)	
61	ADR_5 (O)	Port E_5 (O)	Port E_5 (O)	
62	ADR_4 (O)	Port E_4 (O)	Port E_4 (O)	
63	ADR_3 (O)	Port E_3 (O)	Port E_3 (O)	
64	ADR_2 (O)	Port E_2 (O)	Port E_2 (O)	
65	ADR_1 (O)	Port E_1 (O)	Port E_1 (O)	
66	ADR_0 (O)	Port E_0 (O)	Port E_0 (O)	
67	$DATA_0$ (I/O)	Port G_0 (I/O)	Port G_0 (I/O)	
68	$DATA_1$ (I/O)	Port G_1 (I/O)	Port G_1 (I/O)	
69	$DATA_2$ (I/O)	Port G_2 (I/O)	Port G_2 (I/O)	
70	$DATA_3$ (I/O)	Port G_3 (I/O)	Port G_3 (I/O)	
71	$DATA_4$ (I/O)	Port G_4 (I/O)	Port G_4 (I/O)	
72	$DATA_5$ (I/O)	Port G_5 (I/O)	Port G_5 (I/O)	
73	$DATA_6$ (I/O)	Port G_6 (I/O)	Port G_6 (I/O)	
74	$DATA_7$ (I/O)	Port G_7 (I/O)	Port G_7 (I/O)	
75	Port B_0 (I/O)	←	←	
76	Port B_1 (I/O)	←	←	
77	Port B_2 (I/O)	←	←	
78	Port B_3 (I/O)	←	←	
79	Port B_4 (I/O)	←	←	
80	Port B_5 (I/O)	←	←	

(to be continued)

Table 2 Pin Function in Master MCU External Extended Modes (Mode 1 and Mode 5)

Pin No.	Main	Sub	Pin No.	Main	Sub
1	Port B_6 (I/O)		41	Port H_5 (I)	
2	Port B_7 (I/O)		42	Port H_6 (I)	
3	$\overline{RES}$		43	Port H_7 (I)	
4	XTAL		44	R/$\overline{W}$ (O)	
5	EXTAL		45	E (O)	
6	MP_1 ('L')		46	Port J_2 (I/O)	
7	MP_0 ('H')		47	Port J_1 (I/O)	
8	$\overline{NMI}$		48	Port J_0 (I/O)	
9	$\overline{STBY}$		49	ADR_{15} (O)	
10	V_{CC}		50	ADR_{14} (O)	
11	Port C_7 (I/O)	Tin1 (I)	51	ADR_{13} (O)	
12	Port C_6 (I/O)	Tout1 (O)	52	ADR_{12} (O)	
13	Port C_5 (I/O)	Tin2 (I)	53	ADR_{11} (O)	
14	Port C_4 (I/O)	Tout2 (O)	54	ADR_{10} (O)	
15	Port C_3 (I/O)	Tout3 (O)	55	ADR_9 (O)	
16	Port C_2 (I/O)	SCLK (I/O)	56	ADR_8 (O)	
17	Port C_1 (I/O)	Rx (I)	57	MP_2 ('L/H')	
18	Port C_0 (I/O)	Tx (O)	58	V_{SS}	
19	Port A_0 (O)	PWMout (O)	59	ADR_7 (O)	
20	Port A_1 (O)	$\overline{INT_1}$ (I)	60	ADR_6 (O)	
21	Port A_2 (O)	$\overline{INT_2}$ ($\overline{STB}$) (I)	61	ADR_5 (O)	
22	Port A_3 (O)		62	ADR_4 (O)	
23	Port A_4 (O)		63	ADR_3 (O)	
24	Port A_5 (O)		64	ADR_2 (O)	
25	Port A_6 (O)		65	ADR_1 (O)	
26	AV_{CC}		66	ADR_0 (O)	
27	Port D_0 (I)	AN_0 (I)	67	$DATA_0$ (I/O)	
28	Port D_1 (I)	AN_1 (I)	68	$DATA_1$ (I/O)	
29	Port D_2 (I)	AN_2 (I)	69	$DATA_2$ (I/O)	
30	Port D_3 (I)	AN_3 (I)	70	$DATA_3$ (I/O)	
31	Port D_4 (I)	AN_4 (I)	71	$DATA_4$ (I/O)	
32	Port D_5 (I)	AN_5 (I)	72	$DATA_5$ (I/O)	
33	Port D_6 (I)	AN_6 (I)	73	$DATA_6$ (I/O)	
34	Port D_7 (I)	AN_7 (I)	74	$DATA_7$ (I/O)	
35	AV_{SS}		75	Port B_0 (I/O)	
36	Port H_0 (I/O)		76	Port B_1 (I/O)	
37	Port H_1 (I/O)		77	Port B_2 (I/O)	
38	Port H_2 (I/O)		78	Port B_3 (I/O)	
39	Port H_3 (I/O)		79	Port B_4 (I/O)	
40	Port H_4 (I/O)		80	Port B_5 (I/O)	

Internal ROM is enabled in Mode 1.
Internal ROM is disabled in Mode 5.

Table 3 Pin Function in Master MCU Single Chip Mode (Mode 2)

Pin No.	Main	Sub	Pin No.	Main	Sub
1	Port B_6 (I/O)		41	Port H_5 (I/O)	
2	Port B_7 (I/O)		42	Port H_6 (I/O)	
3	$\overline{RES}$		43	Port H_7 (I/O)	
4	XTAL		44	Port J_4 II/O)	
5	EXTAL		45	Port J_3 (I/O)	
6	MP_1 ('H')		46	Port J_2 (I/O)	
7	MP_0 ('L')		47	Port J_1 (I/O)	
8	$\overline{NMI}$		48	Port J_0 (I/O)	
9	$\overline{STBY}$		49	Port F_7 (I/O)	
10	V_{CC}		50	Port F_6 (I/O)	
11	Port C_7 (I/O)	Tin1 (I)	51	Port F_5 (I/O)	
12	Port C_6 (I/O)	Tout1 (O)	52	Port F_4 (I/O)	
13	Port C_5 (I/O)	Tin2 (I)	53	Port F_3 (I/O)	
14	Port C_4 (I/O)	Tout2 (O)	54	Port F_2 (I/O)	
15	Port C_3 (I/O)	Tout3 (O)	55	Port F_1 (I/O)	
16	Port C_2 (I/O)	SCLK (I/O)	56	Port F_0 (I/O)	
17	Port C_1 (I/O)	Rx (I)	57	MP_2 ('L')	
18	Port C_0 (I/O)	Tx (O)	58	V_{SS}	
19	Port A_0 (O)	PWMout (O)	59	Port E_7 (O)	
20	Port A_1 (O)	$\overline{INT_1}$ (I)	60	Port E_6 (O)	
21	Port A_2 (O)	$\overline{INT_2}$ ($\overline{STB}$) (I)	61	Port E_5 (O)	
22	Port A_3 (O)		62	Port E_4 (O)	
23	Port A_4 (O)		63	Port E_3 (O)	
24	Port A_5 (O)		64	Port E_2 (O)	
25	Port A_6 (O)		65	Port E_1 (O)	
26	AV_{CC}		66	Port E_0 (O)	
27	Port D_0 (I)	AN_0 (I)	67	Port G_0 (I/O)	
28	Port D_1 (I)	AN_1 (I)	68	Port G_1 (I/O)	
29	Port D_2 (I)	AN_2 (I)	69	Port G_2 (I/O)	
30	Port D_3 (I)	AN_3 (I)	70	Port G_3 (I/O)	
31	Port D_4 (I)	AN_4 (I)	71	Port G_4 (I/O)	
32	Port D_5 (I)	AN_5 (I)	72	Port G_5 (I/O)	
33	Port D_6 (I)	AN_6 (I)	73	Port G_6 (I/O)	
34	Port D_7 (I)	AN_7 (I)	74	Port G_7 (I/O)	
35	AV_{SS}		75	Port B_0 (I/O)	
36	Port H_0 (I/O)		76	Port B_1 (I/O)	
37	Port H_1 (I/O)		77	Port B_2 (I/O)	
38	Port H_2 (I/O)		78	Port B_3 (I/O)	
39	Port H_3 (I/O)		79	Port B_4 (I/O)	
40	Port H_4 (I/O)		80	Port B_5 (I/O)	

Table 4 Pin Function in Slave MCU Single Chip Mode (Mode 6)

Pin No.	Main	Sub	Pin No.	Main	Sub
1	Port B_6 (I/O)		41	DBB_5 (I/O)	
2	Port B_7 (I/O)		42	DBB_6 (I/O)	
3	$\overline{RES}$		43	DBB_7 (I/O)	
4	XTAL		44	$\overline{WE}$ (I)	
5	EXTAL		45	$\overline{OE}$ (I)	
6	MP_1 ('H')		46	$\overline{RDY}$ (O)	
7	MP_0 ('L')		47	RS_0 (I)	
8	$\overline{NMI}$		48	$\overline{CS}$ (I)	
9	$\overline{STBY}$		49	RS_3 (I)	
10	V_{CC}		50	RS_2 (I)	
11	Port C_7 (I/O)	Tin1 (I)	51	RS_1 (I)	
12	Port C_6 (I/O)	Tout1 (O)	52	Port F_4 (I/O)	
13	Port C_5 (I/O)	Tin2 (I)	53	Port F_3 (I/O)	
14	Port C_4 (I/O)	Tout2 (O)	54	Port F_2 (I/O)	
15	Port C_3 (I/O)	Tout3 (O)	55	Port F_1 (I/O)	
16	Port C_2 (I/O)	SCLK (I/O)	56	Port F_0 (I/O)	
17	Port C_1 (I/O)	Rx (I)	57	MP_2 ('H')	
18	Port C_0 (I/O)	Tx (O)	58	V_{SS}	
19	Port A_0 (O)	PWMout (O)	59	Port E_7 (O)	
20	Port A_1 (O)	$\overline{INT_1}$ (I)	60	Port E_6 (O)	
21	Port A_2 (O)	$\overline{INT_2}$ ($\overline{STB}$) (I)	61	Port E_5 (O)	
22	Port A_3 (O)		62	Port E_4 (O)	
23	Port A_4 (O)		63	Port E_3 (O)	
24	Port A_5 (O)		64	Port E_2 (O)	
25	Port A_6 (O)		65	Port E_1 (O)	
26	AV_{CC}		66	Port E_0 (O)	
27	Port D_0 (I)	AN_0 (I)	67	Port G_0 (I/O)	
28	Port D_1 (I)	AN_1 (I)	68	Port G_1 (I/O)	
29	Port D_2 (I)	AN_2 (I)	69	Port G_2 (I/O)	
30	Port D_3 (I)	AN_3 (I)	70	Port G_3 (I/O)	
31	Port D_4 (I)	AN_4 (I)	71	Port G_4 (I/O)	
32	Port D_5 (I)	AN_5 (I)	72	Port G_5 (I/O)	
33	Port D_6 (I)	AN_6 (I)	73	Port G_6 (I/O)	
34	Port D_7 (I)	AN_7 (I)	74	Port G_7 (I/O)	
35	AV_{SS}		75	Port B_0 (I/O)	
36	DBB_0 (I/O)		76	Port B_1 (I/O)	
37	DBB_1 (I/O)		77	Port B_2 (I/O)	
38	DBB_2 (I/O)		78	Port B_3 (I/O)	
39	DBB_3 (I/O)		79	Port B_4 (I/O)	
40	DBB_4 (I/O)		80	Port B_5 (I/O)	

Table 5 Pin Function in PROM Programming Mode (Mode 4)

Pin No.	Main	Sub*	Pin No.	Main	Sub*
1	(Not Used)		41	(Not Used)	
2	(Not Used)		42	(Not Used)	
3	(Not Used)		43	EA_5	(5)
4	(Not Used)		44	EA_6	(4)
5	(Not Used)		45	EA_0	(10)
6	MP_1 ('L')		46	EA_1	(9)
7	MP_0 ('L')		47	EA_2	(8)
8	EA_9	(24)	48	EA_3	(7)
9	$\overline{STBY}$ ('L')		49	EA_4	(6)
10	V_{CC}	(28)	50	EA_{14}	(27)
11	EO_7	(19)	51	EA_{13}	(26)
12	EO_6	(18)	52	EA_{12}	(2)
13	EO_5	(17)	53	EA_{11}	(23)
14	EO_4	(16)	54	EA_{10}	(21)
15	EO_3	(15)	55	EA_7	(3)
16	EO_2	(13)	56	EA_8	(25)
17	EO_1	(12)	57	V_{PP} (MP_2)	(1)
18	EO_0	(11)	58	V_{SS}	(14)
19	$\overline{CE}$	(20)	59	(Not Used)	
20	$\overline{OE}$	(22)	60	(Not Used)	
21	(Not Used)		61	(Not Used)	
22	(Not Used)		62	(Not Used)	
23	(Not Used)		63	(Not Used)	
24	(Not Used)		64	(Not Used)	
25	(Not Used)		65	(Not Used)	
26	(Not Used)		66	(Not Used)	
27	(Not Used)		67	(Not Used)	
28	(Not Used)		68	(Not Used)	
29	(Not Used)		69	(Not Used)	
30	(Not Used)		70	(Not Used)	
31	(Not Used)		71	(Not Used)	
32	(Not Used)		72	(Not Used)	
33	(Not Used)		73	(Not Used)	
34	(Not Used)		74	(Not Used)	
35	(Not Used)		75	(Not Used)	
36	(Not Used)		76	(Not Used)	
37	(Not Used)		77	(Not Used)	
38	(Not Used)		78	(Not Used)	
39	(Not Used)		79	(Not Used)	
40	(Not Used)		80	(Not Used)	

*Number within parenthesis for Sub Function indicates pin number of the HN27C256.

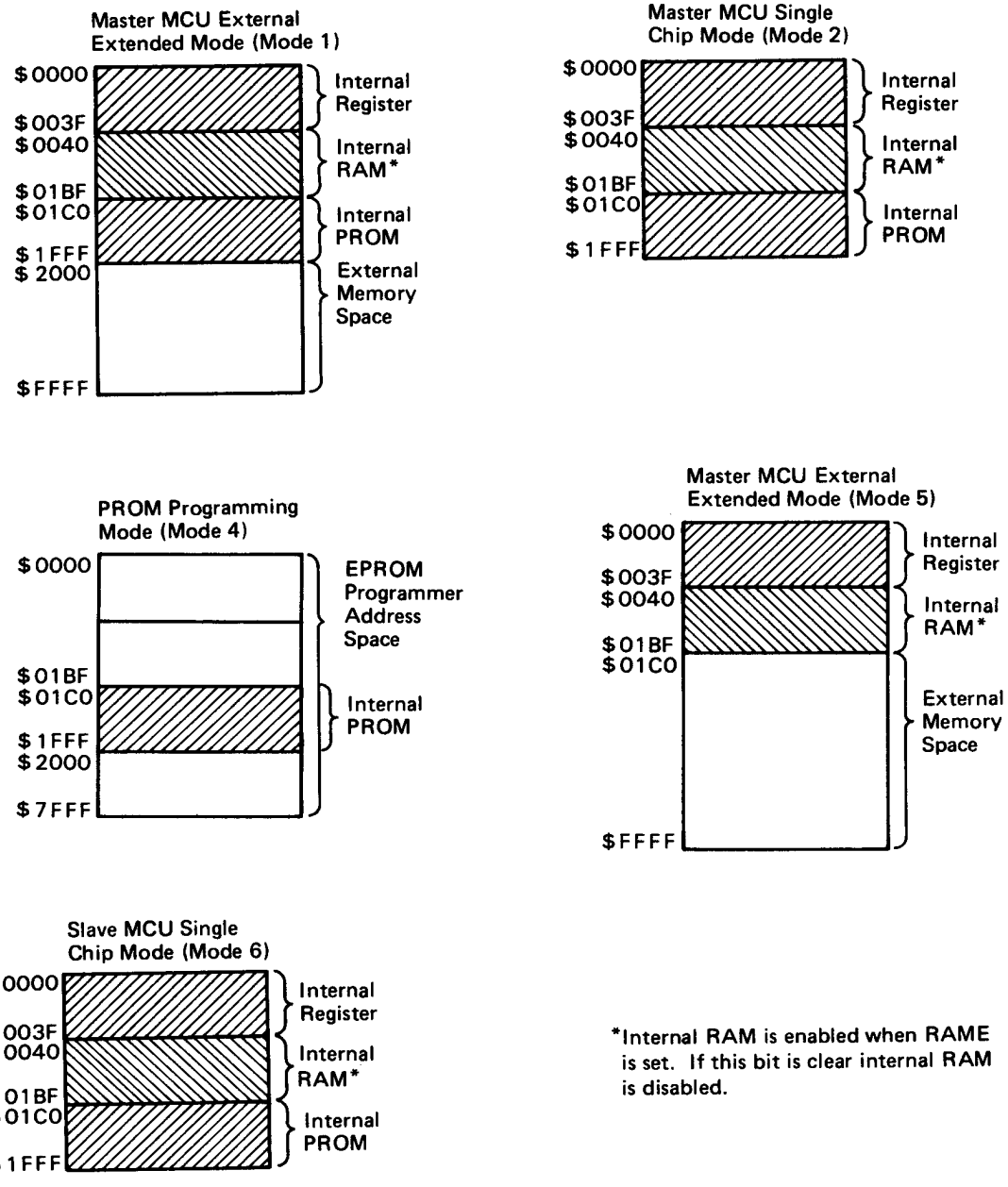
Table 6 MCU Port Condition in Each Operating Mode

Mode \ Port	MP_2	MP_1	MP_0	$\overline{STBY}$	$\overline{NMI}$	$PA_0 \sim PA_6$	$PB_0 \sim PB_7$	$PC_0 \sim PC_7$	$PD_0 \sim PD_7$	$PE_0 \sim PE_7$	$PF_0 \sim PF_7$	$PG_0 \sim PG_7$	$PH_0 \sim PH_7$	$PJ_0 \sim PJ_4$ *1	*2 ROM	*3 RAM	Address
1	I (0)	I (0)	I (1)	I (−)	I (−)	I/O···PA_1, PA_2 0···PA_0, $PA_3 \sim PA_6$	I/O	I/O	I	$A_0 \sim A_7$ (0)	$A_8 \sim A_{15}$ (0)	$D_0 \sim D_7$ (I/O)	I/O	$PJ_0 \sim PJ_2$···(I/O) $PJ_3 \rightarrow E$ (0) $PJ_4 \rightarrow R/\overline{W}$ (0)	I+E	I	$0000 ~ $FFFF
2	I (0)	I (1)	I (0)	I (−)	I (−)	↑	↑	↑	↑	0	I/O	I/O	I/O	I/O	I	I	$0000 ~ $1FFF
4	V_{PP} I (1)	I (0)	I (0)	I (0)	EA_9 (I)	$PA_0 \rightarrow \overline{CE}$ (I) $PA_1 \rightarrow \overline{OE}$ (I) $PA_2 \sim PA_6$ Not Used	Not Used	$EO_0 \sim EO_7$ (I/O)	Not Used	Not Used	EA_8, EA_7, EA_{10}, EA_{11}, EA_{12}, EA_{13}, EA_{14}, EA_4 (I)	Not Used	$PH_0 \sim PH_4$: Not Used $PH_7 \rightarrow EA_5$ (I)	EA_3, EA_2, EA_1, EA_0, EA_4 (I)	P	–	$0000 ~ $1FFF
5	I (1)	I (0)	I (1)	I (−)	I (−)	I/O···PA_1, PA_2 0···PA_0, $PA_3 \sim PA_6$	I/O	I/O	I	$A_0 \sim A_7$ (0)	$A_8 \sim A_{15}$ (0)	$D_0 \sim D_7$ (I/O)	I/O	$PJ_0 \sim PJ_2$···(I/O) $PJ_3 \rightarrow E$ (0) $PJ_4 \rightarrow R/\overline{W}$ (0)	E	I	$0000 ~ $FFFF
6	I (1)	I (1)	I (0)	I (−)	I (−)	↑	↑	↑	↑	(0)	$PF_0 \sim PF_4$ (I/O) $RS_1 \sim RS_3$ (I)	I/O	$DBB_0 \sim DBB_7$ (I/O)	$\overline{CS}$, RS_0, $\overline{OE}$, $\overline{WE}$···(I) $\overline{RDY}$···(O)	I	I	$0000 ~ $1FFF

(NOTE*) 1. PJ_2 ($\overline{RDY}$) provides an N-MOS open drain output.
2. I: Internal E: External P: Programming and verification.
3. I: Internal RAM is programmable as either internal RAM or external RAM under software control.
4. The $\overline{RES}$ input is not used to put the MCU into the Mode 4.
5. MP_0, MP_1 are to be connected to V_{CC} or V_{SS} through registers.

■ MEMORY MAP

The MCU can provide up to 65k byte address space depending on the operating modes. The memory maps in the individual modes are shown in Fig. 12.



*Internal RAM is enabled when RAME is set. If this bit is clear internal RAM is disabled.

Fig. 12 HD63705Z0 Memory Maps

■ INTERNAL REGISTERS

The HD63705Z0 has the following internal registers.

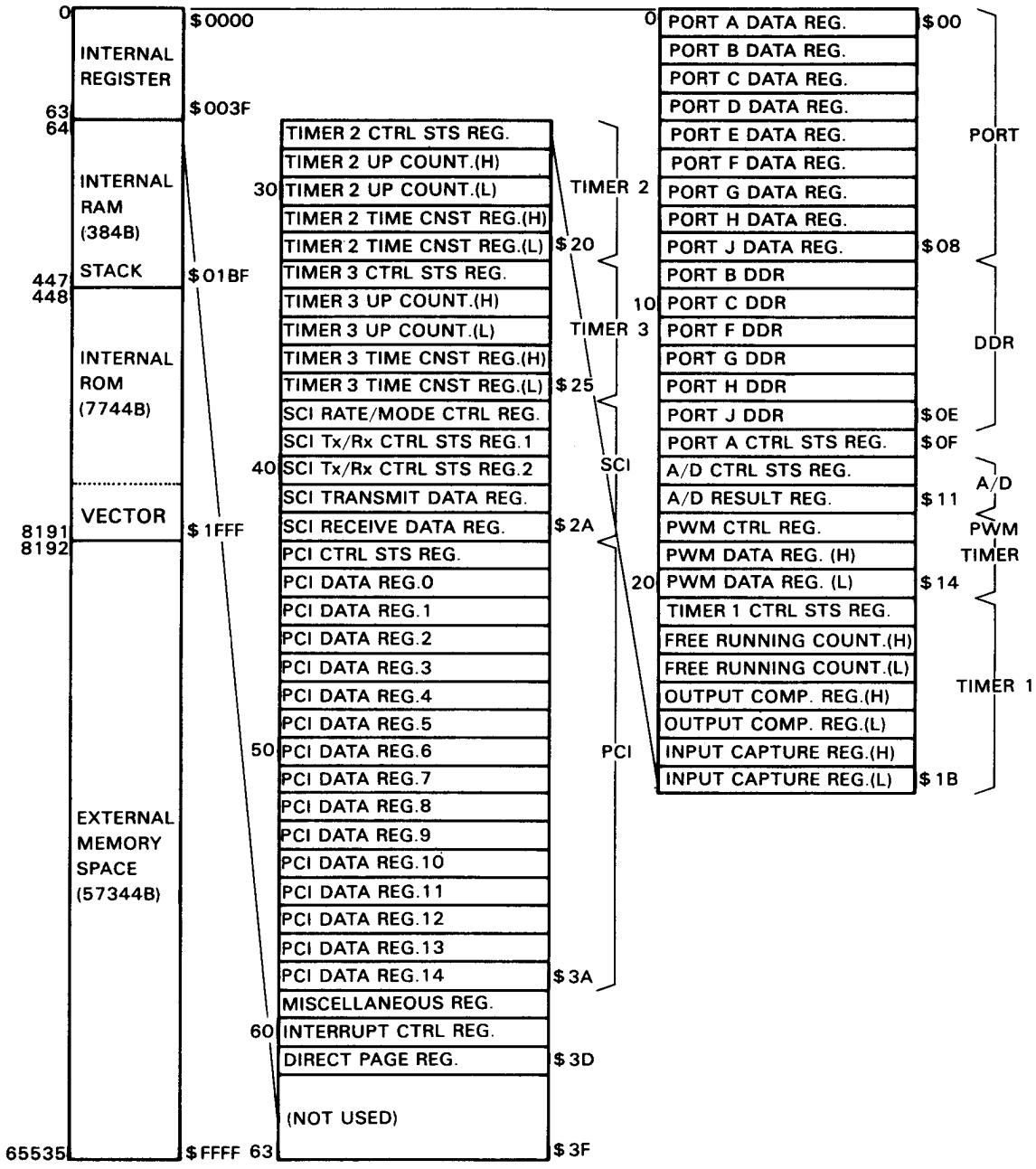


Fig. 13 HD63705Z0 Internal Register Map

Table 7 Internal Register Summary

Register	Address	R/W*1 /Initial Value after Reset							
		7	6	5	4	3	2	1	0
Port A Data Register	$00	NU	R/W						
		1	undetermined						
Port B Data Register	$01	R/W							
		$00							
Port C Data Register	$02	R/W							
		$00							
Port D Data Register	$03	R							
		undetermined							
Port E Data Register	$04	R/W							
		undetermined							
Port F Data Register	$05	R/W							
		$00							
Port G Data Register	$06	R/W							
		$00							
Port H Data Register	$07	R/W							
		$00							
Port J Data Register	$08	Not used			R/W				
		1	1	1	0	0	0	0	0
Port B Data Direction Register	$09	W							
		$00							
Port C Data Direction Register	$0A	W							
		$00							
Port F Data Direction Register	$0B	W							
		$00							
Port G Data Direction Register	$0C	W							
		$00							
Port H Data Direction Register	$0D	W							
		$00							
Port J Data Direction Register	$0E	Not Used			W				
		1	1	1	0	0	0	0	0
Port A Control and Status Register	$0F	R/W							
		$00							
A/D Control and Status Register	$10	R	W*2	R/W	R/W	R/W	R/W	R/W	R/W
		0	0	0	0	0	0	0	0
A/D Result Register	$11	R							
		undetermined							
PWM Control and Status Register	$12	R/W	Not used					R/W	R/W
		0	1	1	1	1	1	0	0

*1 R: read-only W: write-only R/W: read/write NU: Not Used
*2 Only a 1 can be written to this bit.

(Continued)

Register	Address	R/W*1 /Initial Value after Reset							
		7	6	5	4	3	2	1	0
PWM Data Register(High Byte)	$13	W							
		undetermined							
PWM Data Register(Low Byte)	$14	W							
		undetermined							
Timer Control and Status Register 1	$15	R	R	R	R/W	R/W	R/W	R/W	R/W
		0	0	0	0	0	0	0	0
Free Running Counter(High Byte)	$16	R/W							
		$00							
Free Running Counter(Low Byte)	$17	R/W							
		$00							
Output Compare Register(High Byte)	$18	R/W							
		$FF							
Output Compare Register(Low Byte)	$19	R/W							
		$FF							
Input Capture Register(High Byte)	$1A	R							
		$00							
Input Capture Register(Low Byte)	$1B	R							
		$00							
Timer Control and Status Register 2	$1C	R*2	R/W	NU		R/W	R/W	R/W	R/W
		0	0	1	1	0	0	0	0
Timer 2 Up Counter(High Byte)	$1D	R/W							
		$00							
Timer 2 Up Counter(Low Byte)	$1E	R/W							
		$00							
Timer 2 Time Constant Register(High Byte)	$1F	W							
		$FF							
Timer 2 Time Constant Register(Low Byte)	$20	W							
		$FF							
Timer Control and Status Register 3	$21	R*2	R/W	NU		R/W	R/W	NU	
		0	0	1	1	0	0	1	1
Timer 3 Up Counter(High Byte)	$22	R/W							
		$00							
Timer 3 Up Counter(Low Byte)	$23	R/W							
		$00							
Timer 3 Time Constant Register(High Byte)	$24	W							
		$FF							
Timer 3 Time Constant Register(Low Byte)	$25	W							
		$FF							

*1 R: read-only W: write-only R/W: read/write NU: Not Used
*2 A 0 can also be written to this bit.

(Continued)

Register	Address	R/W*1 /Initial Value after Reset							
		7	6	5	4	3	2	1	0
SCI Rate and Mode Control Register	$26	NU		R/W	R/W	R/W	R/W	R/W	R/W
		1	1	0	0	0	0	0	0
Transmit/Receive Control and Status Register 1	$27	R	R	R	R/W	R/W	R/W	R/W	R/W
		0	0	1	0	0	0	0	0
Transmit/Receive Control and Status Register 2	$28	R	R	R	R	NU	R/W	R/W	R/W
		0	0	1	0	1	0	0	0
SCI Transmit Data Register	$29	W							
		undetermined							
SCI Receive Data Register	$2A	R							
		$00							
PCI Control and Status Register	$2B	R	R/W	R	R/W	R	R/W	R	R
		0	0	0	0	0	0	0	0
PCI Data Register 0	$2C	R/W							
		undetermined							
PCI Data Register 1	$2D	R/W							
		undetermined							
PCI Data Register 2	$2E	R/W							
		undetermined							
PCI Data Register 3	$2F	R/W							
		undetermined							
PCI Data Register 4	$30	R/W							
		undetermined							
PCI Data Register 5	$31	R/W							
		undetermined							
PCI Data Register 6	$32	R/W							
		undetermined							
PCI Data Register 7	$33	R/W							
		undetermined							
PCI Data Register 8	$34	R/W							
		undetermined							
PCI Data Register 9	$35	R/W							
		undetermined							
PCI Data Register 10	$36	R/W							
		undetermined							
PCI Data Register 11	$37	R/W							
		undetermined							
PCI Data Register 12	$38	R/W							
		undetermined							

*1 R: read-only W: write-only R/W: read/write NU: Not Used

(Continued)

Register	Address	R/W*1/Initial Value after Reset							
		7	6	5	4	3	2	1	0
PCI Data Register 13	$39	R/W							
		undetermined							
PCI Data Register 14	$3A	R/W							
		undetermined							
Miscellaneous Register	$3B	R/W	R/W	W	R/W	R/W	R	R	R
		0	0	0	1	*2	*3		
Interrupt Control Register	$3C	R/W							
		$FF							
Direct Page Register	$3D	R/W							
		$00							
Not Used	$3E	Not used							
		undetermined							
Not Used	$3F	Not used							
		undetermined							

*1 R: read-only W: write-only R/W: read/write
*2 This bit is cleared only at power-on reset. While V_{CC} is supplied, it is not cleared.
*3 Depends on the operating mode.

■ CPU REGISTERS

The CPU has six registers available to the programmer. They are shown in Fig. 14.

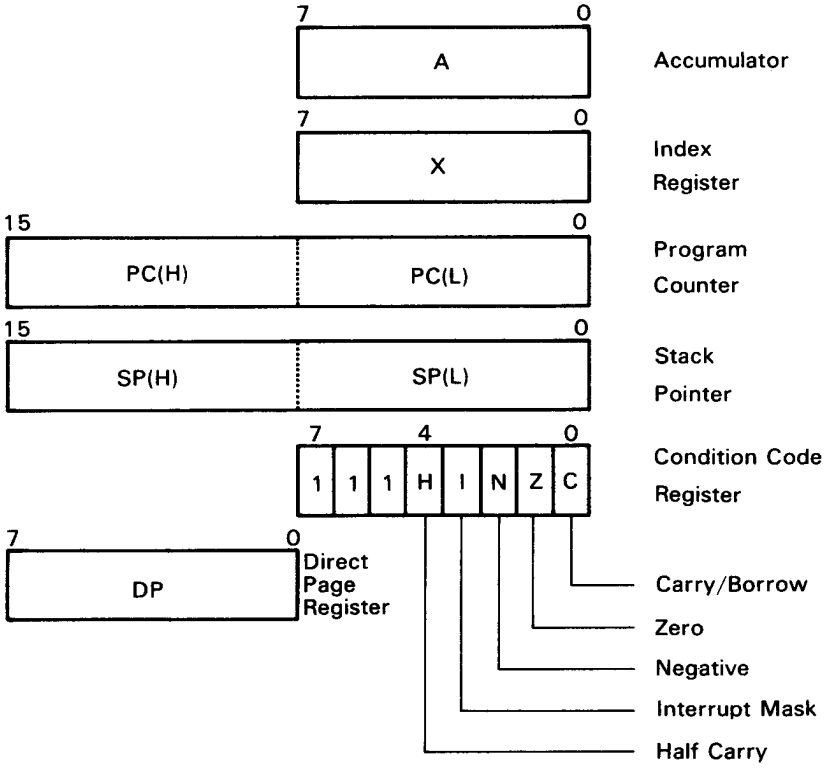


Fig. 14 Programming Model

- **Accumulator (A)**

The accumulator is a general purpose 8-bit register used to hold operands and results of the arithmetic operations or of data manipulation.

- **Index Register (X)**

The index register is an 8-bit register used for the indexed addressing mode.

It contains an 8-bit value which is added to the offset to create an effective address.

The index register can be used for data manipulations with read-modify-write instructions.

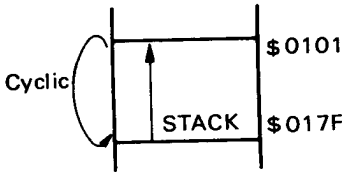
It can be used as a temporary storage area when not performing addressing operations.

- **Program Counter (PC)**

The program counter is a 16-bit register which contains the address of the next instruction to be executed.

- **Stack Pointer (SP)**

The stack pointer is a 16-bit register containing the address of the next free location of the stack. Initially, the stack pointer is set to location $017F. It is decremented as a data is pushed on to the stack and incremented as data is then poped out of the stack. Memory locations $0101 to $017F (127 bytes) can be used for stack.



The 9 high-order bits of the stack pointer are fixed to 000000010.

During an MCU reset or a reset stack pointer (RSP) instruction, the pointer is set to location $017F. Subroutines and interrupts may be nested down to $0101, which allows programmers to use up to 42 levels of subroutine calls and 21 levels of interrupt responses.

- **Direct Page Register (DPR)**

This is an 8-bit register which holds 8 high-order bits of the address. When the CPU executes a direct addressing instruction except for register/memory instructions, the contents of this register is output to the high-order address bus and the operand of the instruction appears in the 8 low-order bits of the address.

All bits in the DPR is cleared on reset, which makes a direct addressing instruction reference to page zero. Write a different data into the DPR to change the page. The DPR is contained in the CPU and also in the memory space, which makes it possible to read and write the contents with load/store instructions.

Refer to "DIRECT PAGE REGISTER" for more information.

- **Condition Code Register (CCR)**

The condition code register is a 5-bit register indicating the results of the instruction just executed. These bits can be individually tested by conditional branch instructions. Each bit is described in the following paragraphs.

Half Carry (H)

When set, this bit indicates that a carry occurred between bit 3 and 4 during an arithmetic operation (ADD, ADC).

Interrupt (I)

Setting this bit masks all interrupts except for $\overline{\text{NMI}}$ and software ones. If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit (I) is cleared. (More precisely, the interrupt enters the servicing routine after the instruction next to the CLI is executed.)

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is negative. (Bit 7 in the result is a logical "1".)

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow occurred during the last arithmetic operation. This bit is also affected by bit test and branch instructions, shifts and rotates.

Interrupt Stacking Order

Fig. 15 shows interrupt stacking order. During the processing of the interrupt, the register contents are pushed onto the stack in the order shown in Fig. 15, decrementing the stack pointer. The low order byte (PCL) of the program counter is stacked first and then the high order byte (PCH) of the program counter, the direct page register (DPR), the index register (X), the accumulator (A) and the condition code register (CCR) are stacked in that order. For subroutine calls, the contents of the program counter (PCH, PCL) and the direct page register (DPR) are pushed onto the stack.

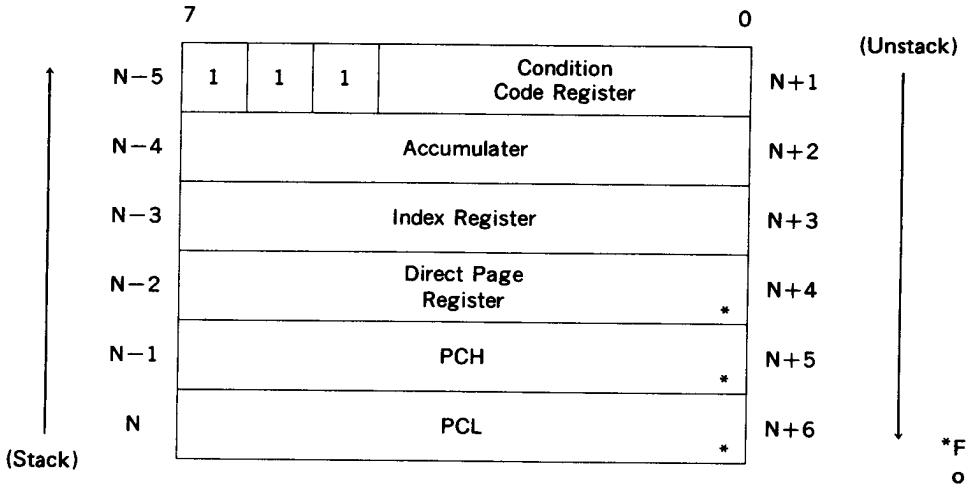


Fig. 15 Interrupt Stacking Order

■ **I/O Port**

The HD63705Z0 provides nine ports: seven 8-bit ports (an input port, an output port and five I/O ports), a 7-bit output port (including two interrupt inputs) and a 5-bit I/O port. Table 8 lists the addresses of ports, data direction registers and control and status registers.

Table 8 Port, Data Direction Register and Control and Status Register Addresses

Port	Port Address	Data Direction Register	Control and Status Register	I/O
A	$0000	——	$000F, $0012 $003C	O (Including two interrupt inputs)
B	$0001	$0009	$000F	I/O
C	$0002	$000A	$0015, $001C, $0021, $0026, $003B, $003C	I/O
D	$0003	——	$0010, $003C	I
E	$0004	——	——	O (High current output)
F	$0005	$000B	——	I/O
G	$0006	$000C	——	I/O
H	$0007	$000D	——	I/O
J	$0008	$000E	——	I/O

● **Port A ($PA_0 - PA_6$)**

Port A provides seven output lines. During reset this port is in a high impedance state. Each output line can drive one TTL load and 90 pF.

Port A Data Register ($0000)

7	6	5	4	3	2	1	0
—	PA_6 DATA	PA_5 DATA	PA_4 DATA	PA_3 DATA	PA_2 DATA	PA_1 DATA	PA_0 DATA

Port A bit 0 (PA_0)

In the MCU modes (Modes 1, 2, 5 and 6) PA_0 can also be used as the output of the PWM timer depending on the bit 7 in the PWM Control and Status Register. When this bit is cleared, PA_0 is used as an data output line. If set, it is used for the PWM timer output.

PWM Control and Status Register ($0012)

7	6	5	4	3	2	1	0	
PWM ENABLE	—	—	—	—	—	PW_1	PW_0	
(0						0	0)	← At Reset

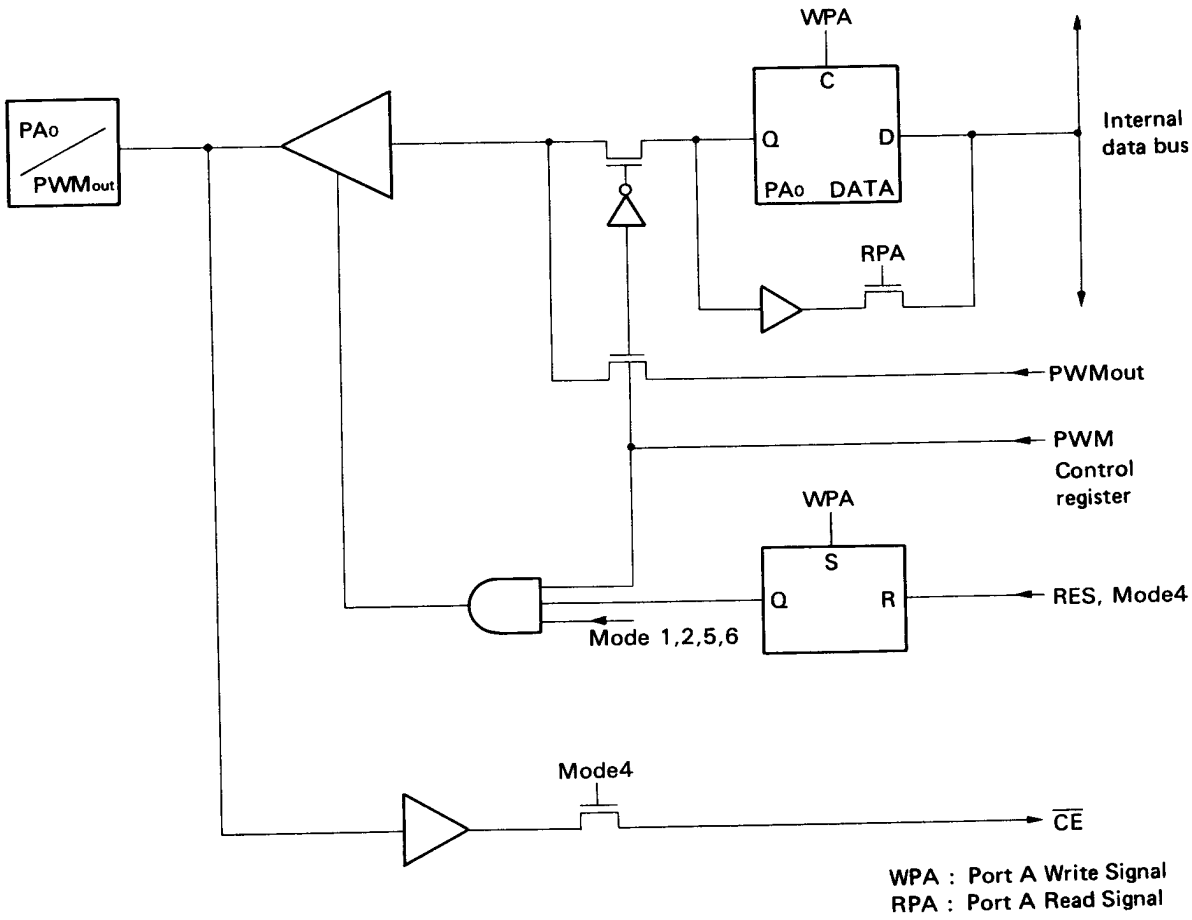


Fig. 16 PA_0 Block Diagram

During reset PA_0 is placed in a high impedance state. If the CPU writes Port A, PA_0 is released from the high impedance state and outputs the written data. Then PA_0 will continue to function as an output line until reset.

In the PROM programming mode (Mode 4), PA_0 is used as the input of chip enable ($\overline{CE}$) signal which controls the PROM. Logic low on this pin enables the PROM.

Port A Bit 1 (PA_1)

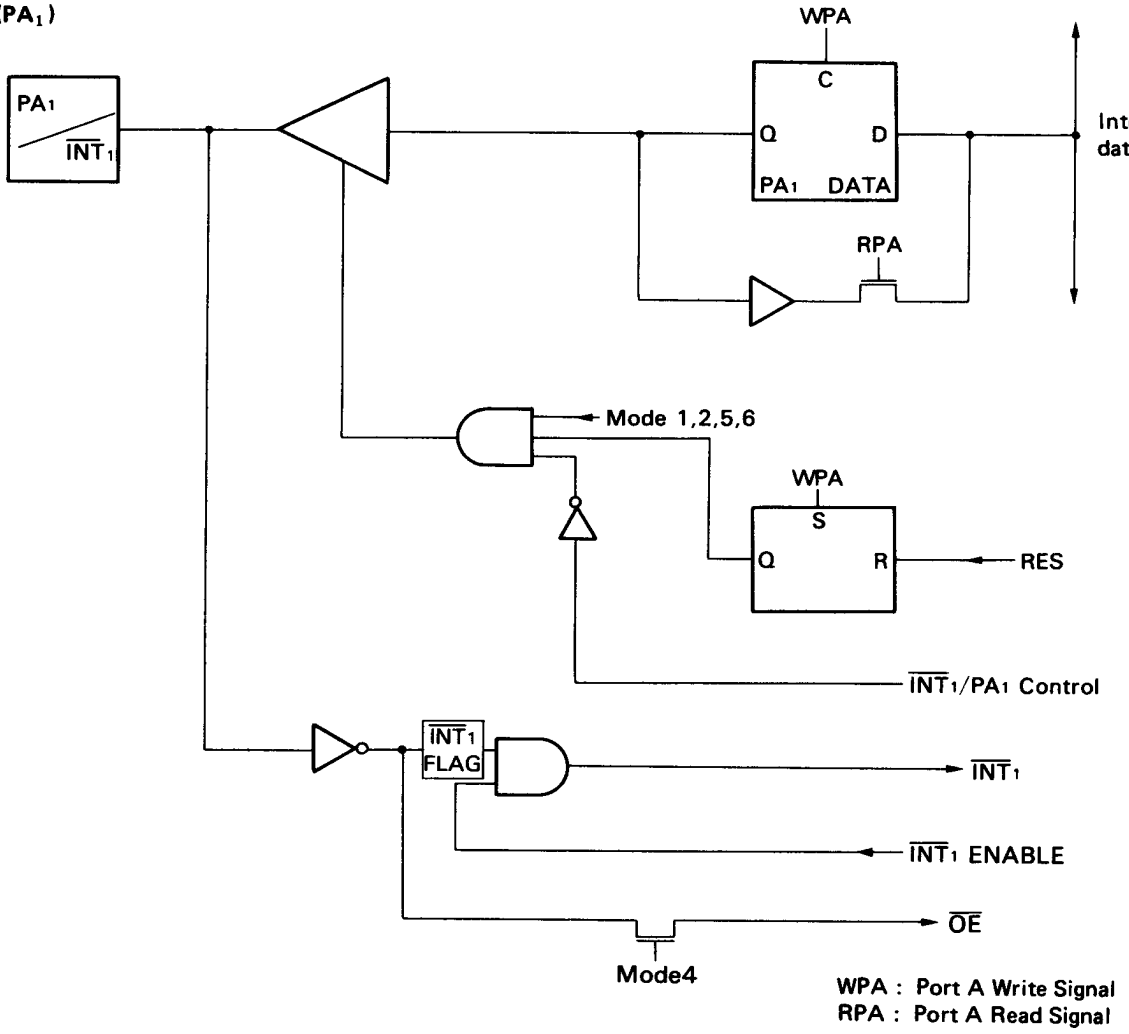


Fig. 17 PA_1 Block Diagram

In the MCU modes (Mode 1, 2, 5 and 6) PA_1 is configured as either as an output line or as the input of $\overline{INT_1}$ depending on bit 1 in the Port A Control and Status Register ($000F).

Port A Control and Status Register ($000F)

7	6	5	4	3	2	1	0
$\overline{INT_2}$ (STB) FLAG	$\overline{INT_1}$ FLAG	$\overline{INT_2}$ (STB) ENABLE	$\overline{INT_1}$ ENABLE	LATCH ENABLE	$\overline{INT_2}$ (STB) /PA_2	$\overline{INT_1}$ /PA_1	$\overline{INT_1}$ L/E
(0	0	0	0	0	0	0	0)

This register is cleared to $00 during reset.

If the bit 1 is cleared, PA_1 is used as an output line; if set, it is used as the input of the $\overline{INT_1}$. When PA_1 functions as the $\overline{INT_1}$ input, the state of the bit 0 determines whether the interrupt is a level-sensitive trigger input or an edge-sensitive trigger input. If the bit is set, the interrupt is detected on the V_{IL} level of the input signal; if cleared, it is detected on the negative edge of the input signal.

The bit 4 of the Port A Control and Status Register is the $\overline{INT_1}$ interrupt enable bit. When this bit is set, $\overline{INT_1}$ is enabled. The $\overline{INT_1}$ is defined in detail in 'INTERRUPT'.

Bit 0: ($\overline{INT_1}$ LEVEL/EDGE SELECT)
- 0: Edge sensitive triggering ← At Reset
- 1: Level sensitive triggering

Bit 1: (PA1/$\overline{INT_1}$ SELECT)
- 0: PA_1 (an output line) ← At Reset
- 1: $\overline{INT_1}$

Bit 4: ($\overline{INT_1}$ ENABLE)
- 0: $\overline{INT_1}$ disabled ← At Reset
- 1: $\overline{INT_1}$ enabled

During reset PA_1 is placed in a high impedance state. If the CPU writes Port A, PA_1 is released from the high impedance state and outputs the written data. Then PA_1 will continue to function as an output line until reset.

In the PROM programming mode (Mode 4), PA_1 is used as the input of output enable ($\overline{OE}$) signal which controls the PROM. Logic low on this pin enables the PROM.

Port A Bit 2 (PA_2)

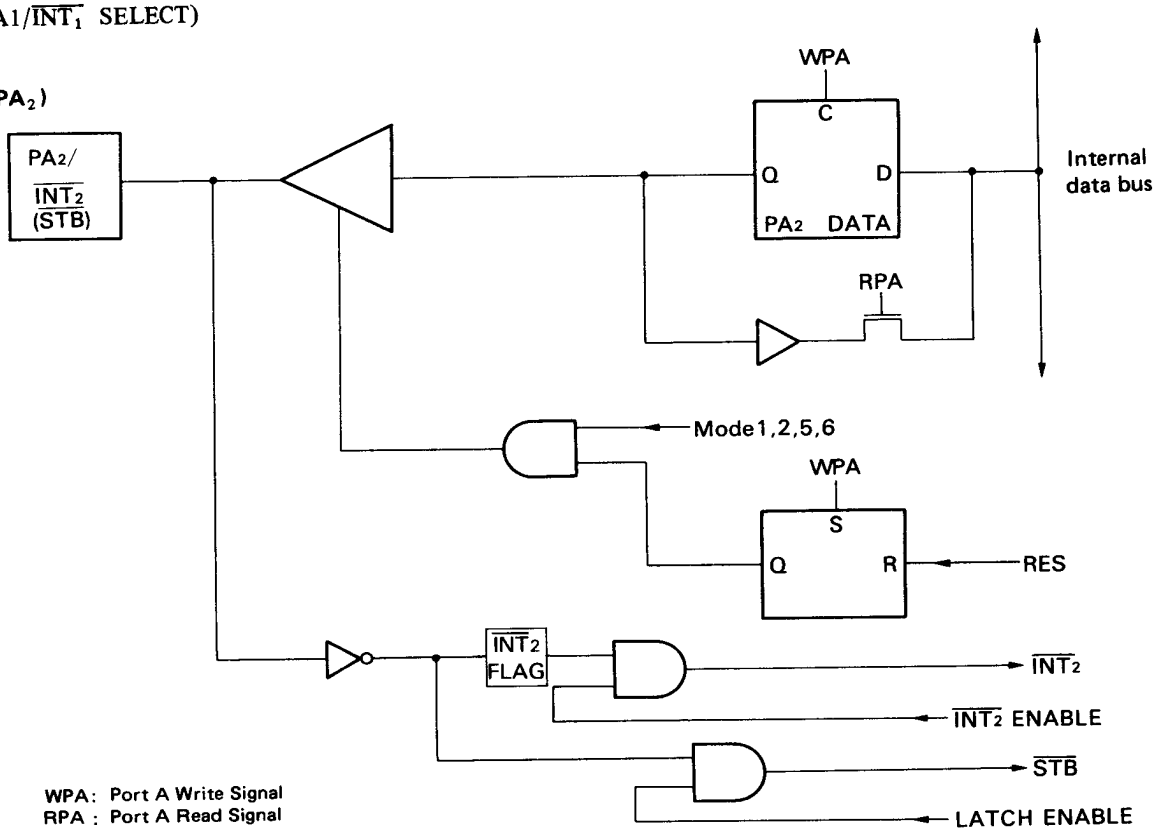


Fig. 18 PA_2 Block Diagram

In the MCU modes (Mode 1, 2, 5 and 6) PA_2 is configured as either as an output line or as the input of $\overline{INT_2}$ ($\overline{STB}$) depending on bit 2 in the Port A Control and Status Register ($000F).

Port A Control and Status Register ($000F)

7	6	5	4	3	2	1	0	
$\overline{INT_2}$ ($\overline{STB}$) FLAG	$\overline{INT_1}$ FLAG	$\overline{INT_2}$ ($\overline{STB}$) ENABLE	$\overline{INT_1}$ ENABLE	LATCH ENABLE	$\overline{INT_2}$ ($\overline{STB}$) /PA_2	$\overline{INT_1}$ /PA_1	$\overline{INT_1}$ L/E	
(0	0	0	0	0	0	0	0)	← Reset

If the bit 2 is cleared, PA_2 is used as an output line; if set, it is used as the input of the $\overline{INT_2}$ ($\overline{STB}$).

The bit 5 in the Port A Control and Status Register is the $\overline{INT_2}$ ($\overline{STB}$) interrupt enable bit. When this bit is set, $\overline{INT_2}$ ($\overline{STB}$) is enabled. The $\overline{INT_2}$ and the $\overline{STB}$ are defined in detail in 'INTERRUPT' and 'Port B, respectively. During reset

PA_2 is placed in a high impedance state. Like PA_1 it is released from the high impedance state and configured as an output line by a CPU write to Port A.

In the PROM programming mode (Mode 4), PA_2 is in a high impedance state.

Port A Bit 3 – Port A Bit 6 (PA_3 to PA_6)

PA_3 to PA_6 are used as data output lines. These port pins are put in a high impedance state during reset. If the CPU writes Port A, they are released from the high impedance state and outputs the written data. Then they will continue to function as output lines until reset.

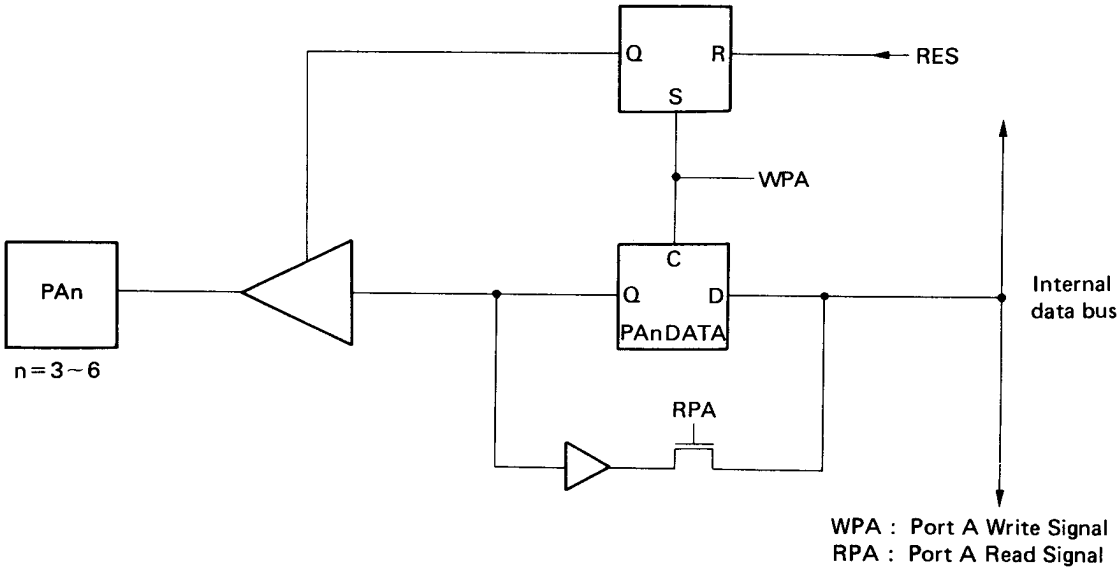


Fig. 19 PA_3 to PA_6 Block Diagram

- **Port B (mode independent port)**

Port B is an 8-bit I/O port. Each pin on the port is individually configured as an input or an output as defined by the Port B Data Direction Register ($0009). Each pin is programmed as an input when the corresponding Port B Data Direction Register bit is cleared to a logic 0 and as an output when it is set to a logic 1. On reset the Port B Data Direction Register is initialized to a logic 0, placing the Port B in the input mode.

Port B can also be used as an input data latch, depending on bit 3 in the Port A Control and Status Register. After reset all bits of the Port A Control and Status Register are initialized to logic 0's. If the bit 3 is set to a logic 1, data is latched into Port B on a $\overline{INT_2}$ ($\overline{STB}$) negative edge. At this point, if the bit 5 of the Port A Control and Status Register has been set, an interrupt occurs on the $\overline{INT_2}$ ($\overline{STB}$) negative edge, setting the bit 7 in the Port A Control and Status Register. If the bit 3 in the Port A Control and Status Register is cleared, this port is configured as a normal input port.

The Port B Data Register is cleared during reset.

Each pin on Port B can drive a TTL load and 90 pF.

Port B Data Direction Register ($0009)
(Write only. Cleared to $00 during reset)

7	6	5	4	3	2	1	0
PB_7 DDR	PB_6 DDR	PB_5 DDR	PB_4 DDR	PB_3 DDR	PB_2 DDR	PB_1 DDR	PB_0 DDR

Port B Data Register ($0001)
(R/W. Cleared to $00 during reset)

PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0

Port A Control and Status Register ($000F)
(R/W. Cleared to $00 during reset)

$\overline{INT_2}$ ($\overline{STB}$) FLAG	$\overline{INT_1}$ FLAG	$\overline{INT_2}$ ($\overline{STB}$) ENABLE	$\overline{INT_1}$ ENABLE	LATCH ENABLE	$\overline{INT_2}$ ($\overline{STB}$) /PA_2	$\overline{INT_1}$ /PA_1	$\overline{INT_1}$ L/E

HITACHI

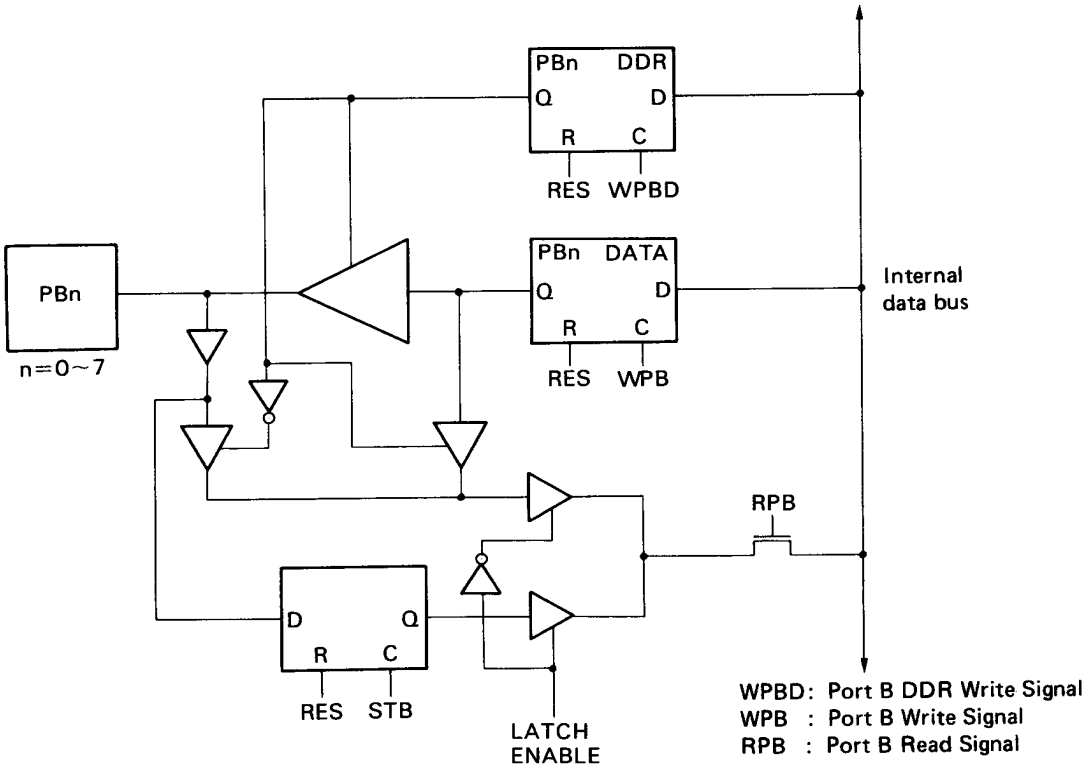


Fig. 20 Port B Block Diagram

- **Port C (PC_0 to PC_7)**

This is an 8-bit I/O pin. In the MCU modes (Modes 1, 2, 5 and 6) all pins on Port C are programmable as either inputs or outputs under software control of the Port C Data Direction Register ($000A). Each pin is programmable as an output when the corresponding Port C Data Direction Register bit is set, and as an input when it is cleared. During reset the Port C Data Direction Register is cleared, configuring Port C as an input port.

Port C is shared with SCI, Timer 1, Timer 2 and Timer 3. When each pin on Port C functions as an I/O pin for the SCI, Timer 1, Timer 2 or Timer 3, it is programmed as an input or an output depending on its function. When the pins revert from I/O pins of the timers or the SCI to data I/O lines, the direction of the pins remains unchanged.

In the PROM programming mode (Mode 4) Port C provides PROM data bus lines (EO_0 to EO_7). In this mode, each line is configured as either an input or output depending on $\overline{CE}$ and $\overline{OE}$ signals, regardless of the Port C Data Direction Register. For details, see 'PROGRAMMING THE PROM'.

Port C Data Register is cleared during reset.

Each output line on Port C can drive one TTL load and 90 pF.

Port C Data Direction Register ($000A)

(Write-only. Cleared to $00 during reset)

7	6	5	4	3	2	1	0
PC_7 DDR	PC_6 DDR	PC_5 DDR	PC_4 DDR	PC_3 DDR	PC_2 DDR	PC_1 DDR	PC_0 DDR

Port C Data Register ($0002)

(R/W. Cleared to $00 during reset)

PC_7	PC_6	PC_5	PC_4	PC_3	PC_2	PC_1	PC_0

PC_0 (Tx)/EO_0, PC_3 ($Tout_3$)/EO_3, PC_4 ($Tout_2$)/EO_4, PC_6 ($Tout_1$)/EO_6

In the MCU modes (Modes 1, 2, 5 and 6) these port pins can also be the SCI transmitter output, Timer 1 output, Timer 2 output and Timer 3 output, respectively. The SCI and timers have individual register bits to enable their outputs. If one of the bits is set, the corresponding port pin will be forced to an output of the SCI or timers.

The configuration of PC_6 depends on the state of the bit 6 in the Miscellaneous Register ($003B). If this bit is cleared, PC_6 is used as an I/O line; if set, it is used as the Timer 1 output ($Tout_1$). The bit 6 is cleared at reset.

If a pin reverts from the output of the SCI or timers to an data I/O line, the direction of the pin will be an output.

In the PROM programming mode (Mode 4), Port C functions as PROM data bus (EOn).

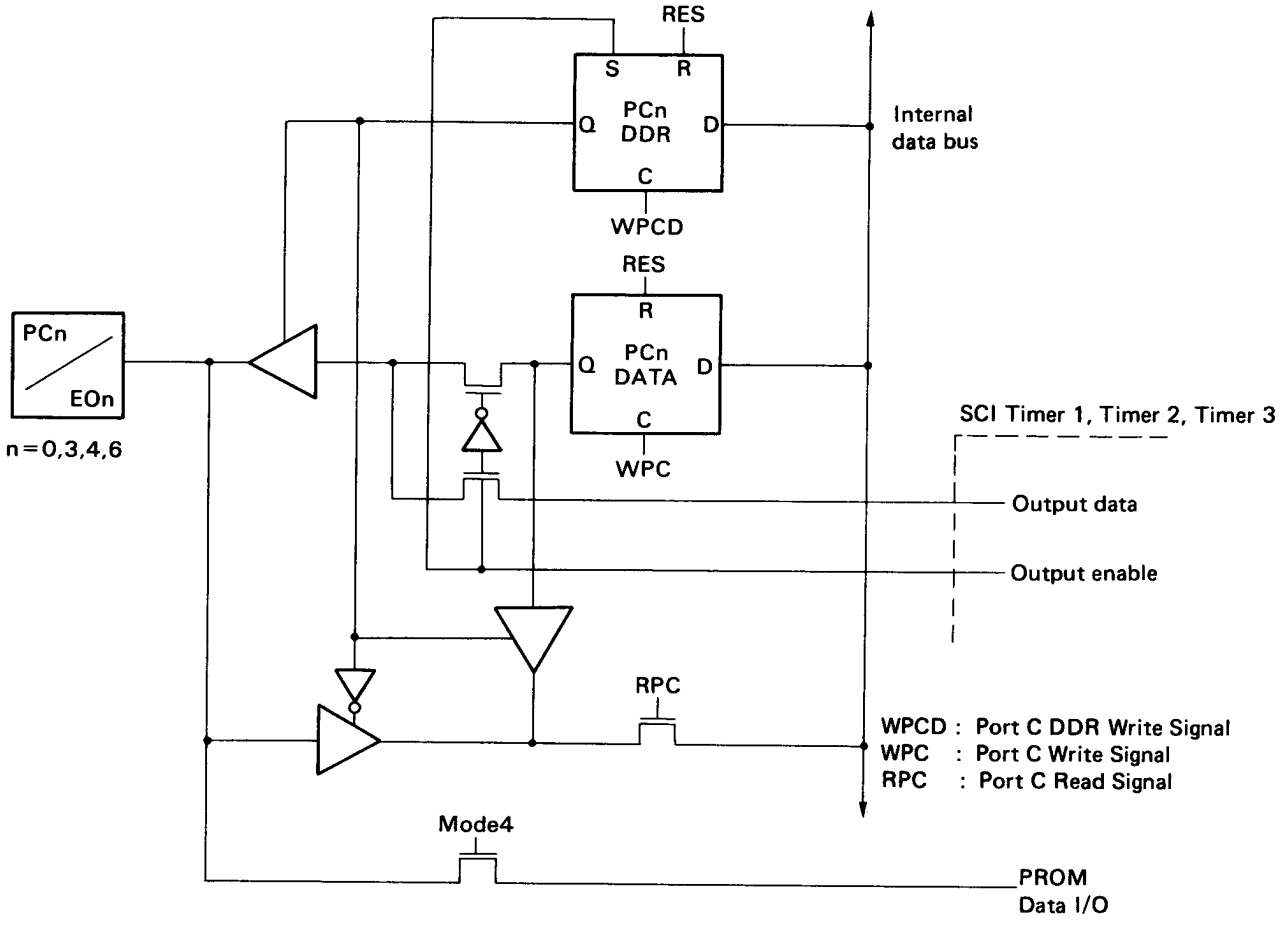


Fig. 21 PC_0, PC_3, PC_4 and PC_6 Block Diagram

PC_1 (Rx)/EO_1, PC_5 (Tin_2)/EO_5, PC_7 (Tin_1)/EO_7

In the MCU modes (Modes 1, 2, 5 and 6) these port pins can also be the SCI receiver input, Timer 1 input and Timer 2 input, respectively. The SCI and timers have individual register bits to enable their inputs. If one of the bits is set, the corresponding port pin will be forced to an input of the SCI or timers.

The configuration of PC_7 depends on the state of the bit 7 in the Miscellaneous Register ($003B). If this bit is cleared, PC_7 is used as an I/O line; if set, it is used as the Timer 1 input ($Tout_1$). The bit 7 is cleared at reset.

If one of PC_1, PC_5 and PC_7 reverts from the input of the SCI or timers to an data I/O line, the direction of the pin will be an input.

In the PROM programming mode (Mode 4), Port C functions as PROM data bus (EOn).

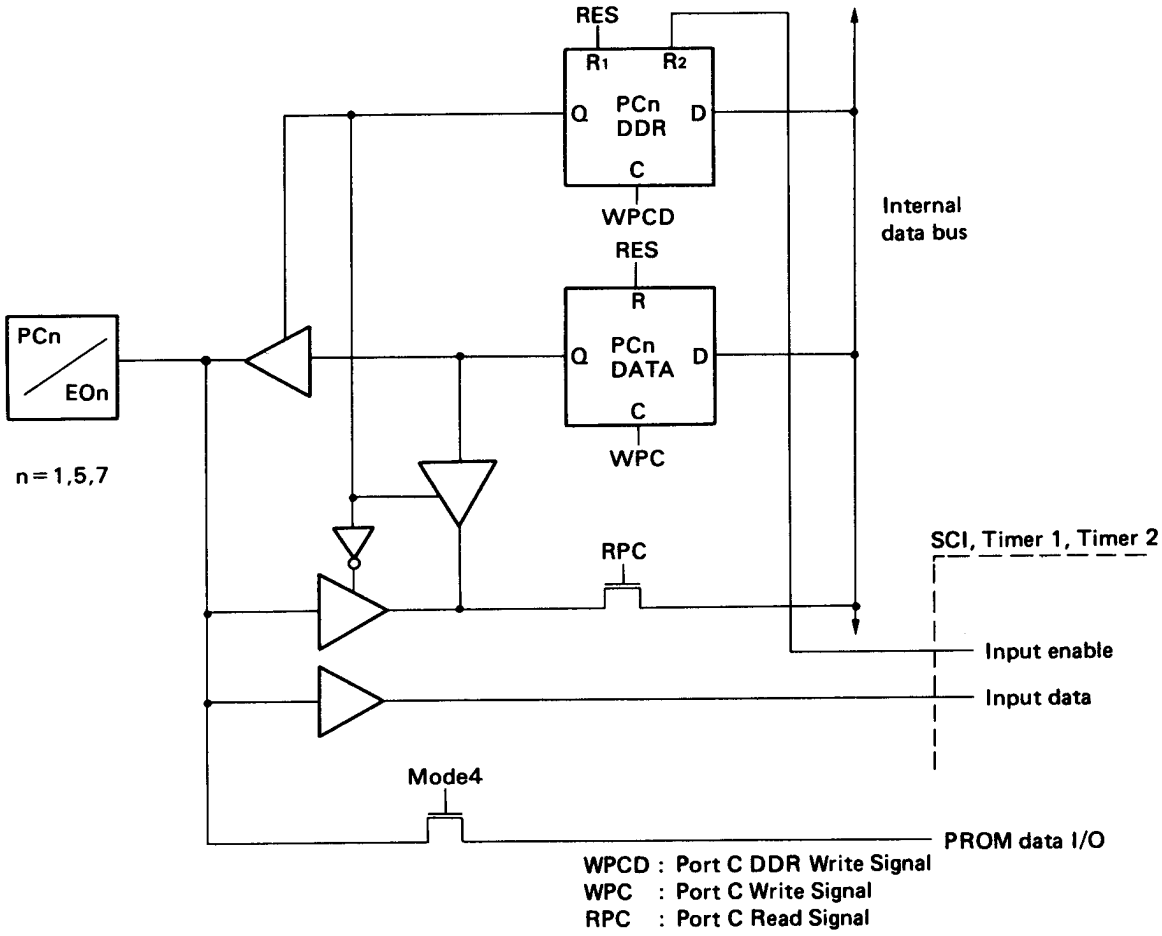


Fig. 22 PC_1, PC_5 and PC_7 Block Diagram

PC_2 (SCLK)/EO_2

In the MCU modes (Modes 1, 2, 5 and 6) PC_2 can also be used as the SCI clock I/O pin. PC_2 is configured as either the external clock input or internal clock output depending on the Rate and Mode Control Register. When the SCI clock is not input or output via PC_2, this pin can be used as a data I/O line. Setting the Data Direction Register bit for PC_2 programs PC_2 as an output and clearing it programs the pin as input.

When PC_2 reverts from the SCI clock I/O pin to an I/O line, the direction of the pin remains unchanged.

7	6	5	4	3	2	1	0
—	—	SS2	CC2	CC1	CC0	SS1	SS0

Rate and Mode Control Register ($0026)

(The bits 0 to 5 are cleared during reset)

In the PROM programming mode (Mode 4) PC_2 is used as an PROM data bus line (EO_2).

* During reset the SCI is placed in the synchronous mode where an external clock is input to the SCI. PC_2 is forced to an input line to accept the external clock. PC_2 can be used as an output line by writing the CC1 : CC0 to 01 then setting the Data Direction Register bit for PC_2.

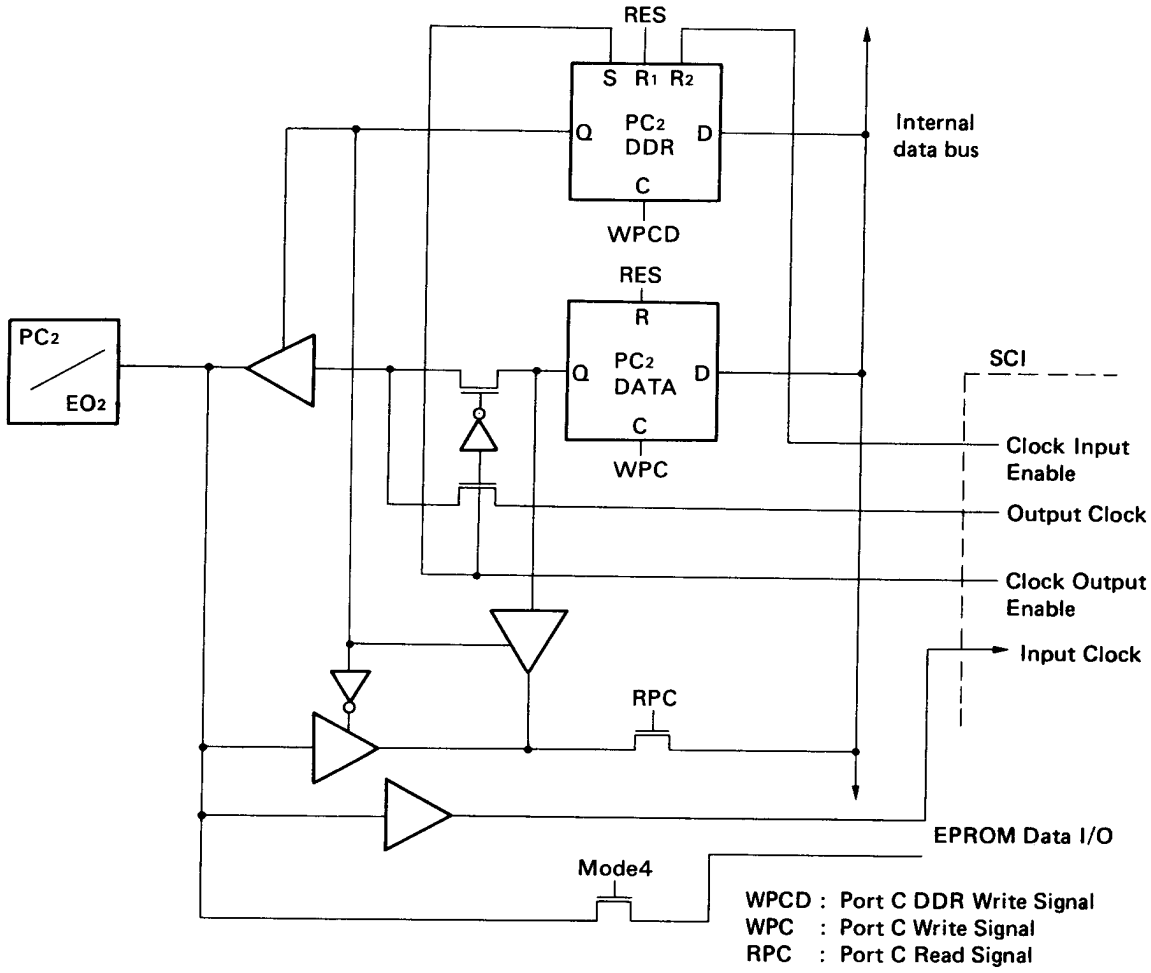


Fig. 23 PC_2 Block Diagram

● Port D (PD_0 to PD_7)

This is an 8-bit input only port. Port D pins can be also used as A/D converter inputs (AN_0 to AN_7): A CPU read of Port D (at $0003) will latch input data on the port pins into the CPU.

When Port D provides A/D converter inputs (AN_0 to AN_7), the states of bits 2 to 4 (CH_0 to CH_2) in the A/D Control and Status Register selects one of the eight inputs.

For detail, see 'A/D CONVERTER'.

Port D Data Register ($0003) (Read only)

7	6	5	4	3	2	1	0
PD_7 DATA	PD_6 DATA	PD_5 DATA	PD_4 DATA	PD_3 DATA	PD_2 DATA	PD_1 DATA	PD_0 DATA

A/D Control and Status Register ($000F) (Cleared to $00 during reset)

7	6	5	4	3	2	1	0
ADEF	ADS	EADEI	CH_2	CH_1	CH_0	ACS1	ACS0

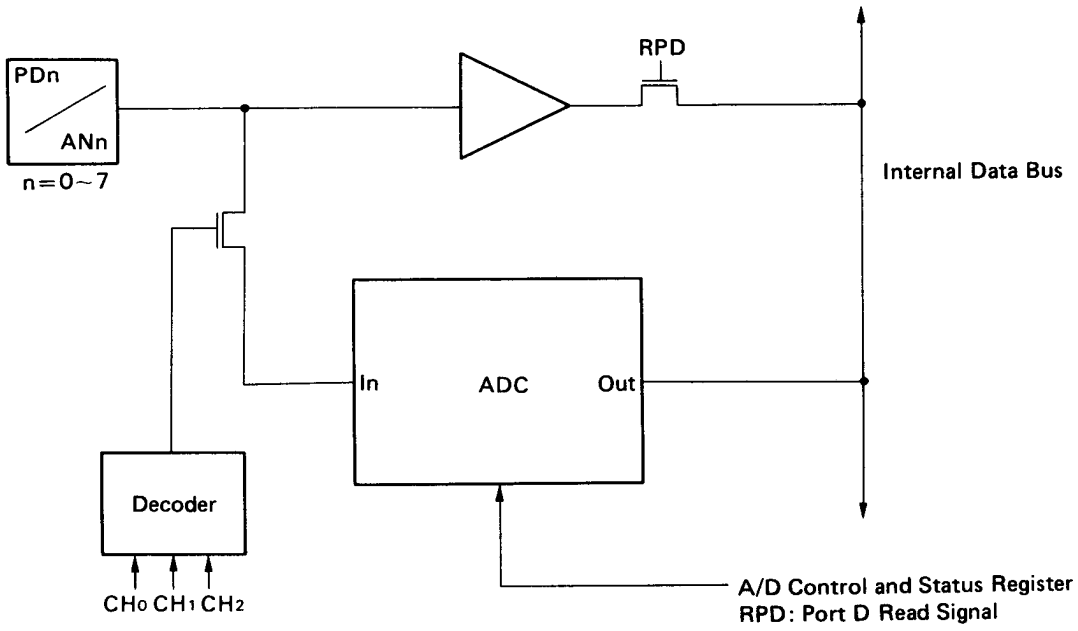


Fig. 24 Port D Block Diagram

- **Port E (PE_0 to PE_7)**

This is an 8-bit output only port. It is a high-current drive output, producing 10 mA as I_{OL} when V_{OL} = 1.0V.

In the single-chip modes (Mode 2, Mode 6) the MCU reset places this port into a high impedance state. A CPU write to Port E causes the port to leave a high impedance state and the port outputs the written data. Then Port E continues to function as an output until reset. Port E can also read the Port E Data Register to execute bit manipulation instructions.

In the extended modes (Mode 1 and Mode 5) Port E provides the eight low-order address bus lines (A_0 to A_7).

Each output line on Port E can drive one TTL load and 90 pF.

Port E Data Register ($0004) (R/W. Undetermined during reset)

7	6	5	4	3	2	1	0
PE_7 DATA	PE_6 DATA	PE_5 DATA	PE_4 DATA	PE_3 DATA	PE_2 DATA	PE_1 DATA	PE_0 DATA

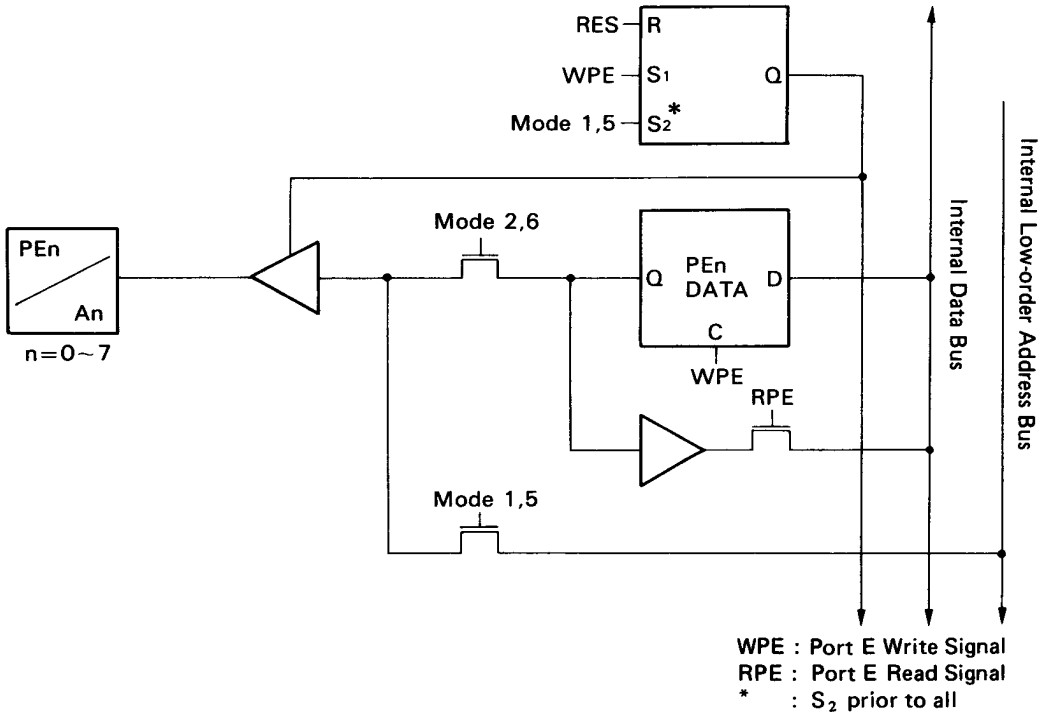


Fig. 25 Port E Block Diagram

- **Port F (PF_0 to PF_7)**

Port F is an 8-bit I/O port. When a pin on Port F provides a data I/O line, it is configured as either an input or an output depending on the Port F Data Direction Register. It is programmed as an input when the corresponding Data Direction Register bit is cleared to a logic 0, and as an output when it is set to a logic 1. The MCU reset clears all the bits of Port F Data Direction Register, causing the Port F to be an input port.

Each output line on Port F can drive one TTL load and 90 pF.

Port F Data Direction Register ($000B) (Write only. Cleared to $00 during reset)

7	6	5	4	3	2	1	0
PF_7 DDR	PF_6 DDR	PF_5 DDR	PF_4 DDR	PF_3 DDR	PF_2 DDR	PF_1 DDR	PF_0 DDR

Port F Data Register ($0005) (R/W. Cleared to $00 during reset)

PF_7	PF_6	PF_5	PF_4	PF_3	PF_2	PF_1	PF_0

Port F functions differently depending on the operating mode as listed in Table 9.

Table 9 Port F Configuration in Five Operation Modes

No.	Mode	PF_0	PF_1	PF_2	PF_3	PF_4	PF_5	PF_6	PF_7
1	Master MCU External Extended Mode	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
2	Master MCU Single-Chip Mode	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O
4	PROM Programming Mode	EA_8	EA_7	EA_{10}	EA_{11}	EA_{12}	EA_{13}	EA_{14}	EA_4
5	Master MCU External Extended Mode	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
6	Slave MCU Single-Chip Mode	I/O	I/O	I/O	I/O	I/O	RS_1 (I)	RS_2 (I)	RS_3 (I)

Mode 1 (address A_8 to A_{15})

In the Mode 1 (Master MCU External Extended Mode) Port F provides the 8 high-order address output lines. (A_8 to A_{15}). During reset, the Port F Data Direction Register is set, which configures all of the lines as address outputs.

Mode 2 (Port F_0 to F_7)

In the Mode 2 (Master MCU Single-Chip Mode) Port F provides eight parallel I/O lines. Port F Data Direction Register is cleared during reset, which configures all of the lines as inputs.

Mode 4 (PROM address EA_8, EA_7, EA_{10} to EA_{14}, EA_4)

In the Mode 4 (PROM Programming Mode) Port F provides PROM address input lines.

Mode 5 (address A_8 to A_{15})

In the Mode 5 (Master MCU External Extended Mode) Port F provides the eight high-order address output lines like in the Mode 1. However, in the Mode 1, the internal PROM is enabled, while in this mode the internal PROM is disabled.

Mode 6 (Port F_0 to F_4, register select RS_1 to RS_3)

In the Mode 6 (Slave MCU Single-Chip Mode) Port F provides five I/O lines (PF_0 to PF_4) and three register select inputs (RS_1 to RS_3).

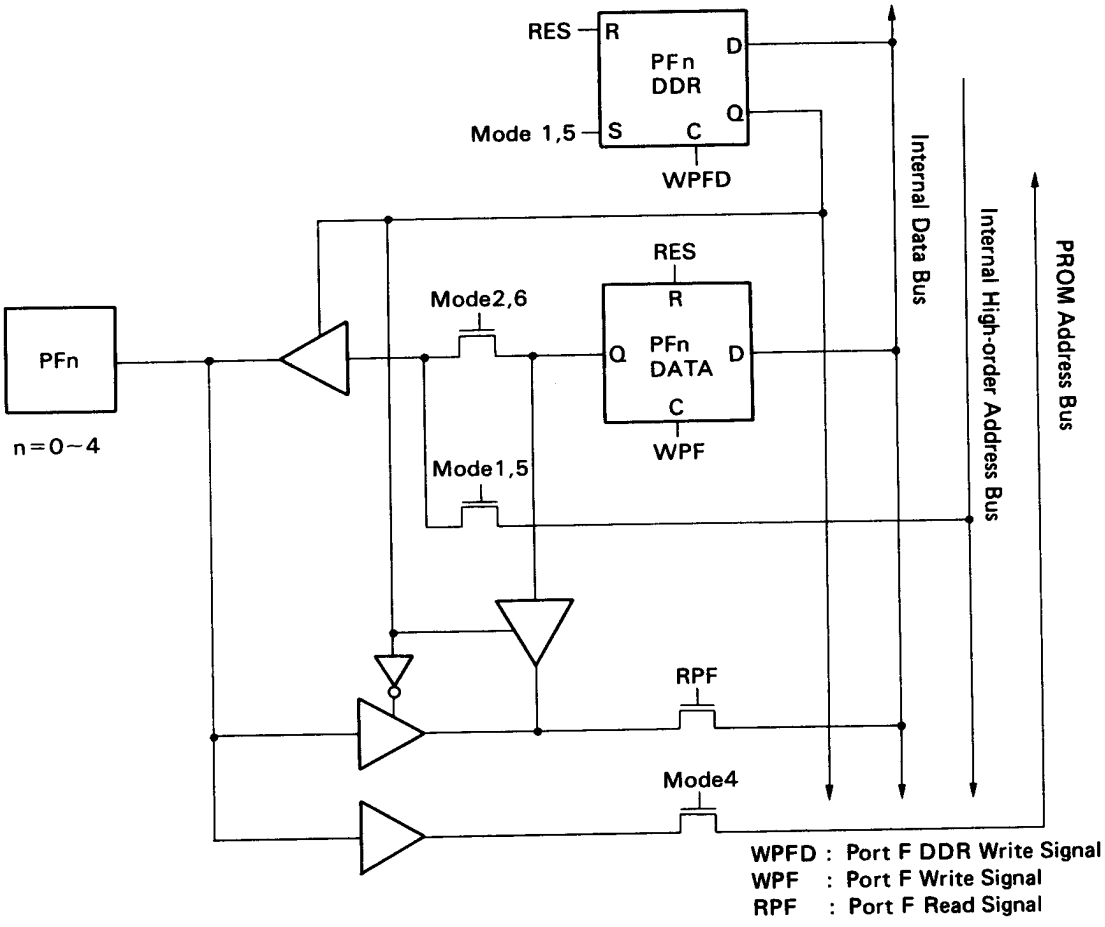


Fig. 26 PF_0 to PF_4 Block Diagram

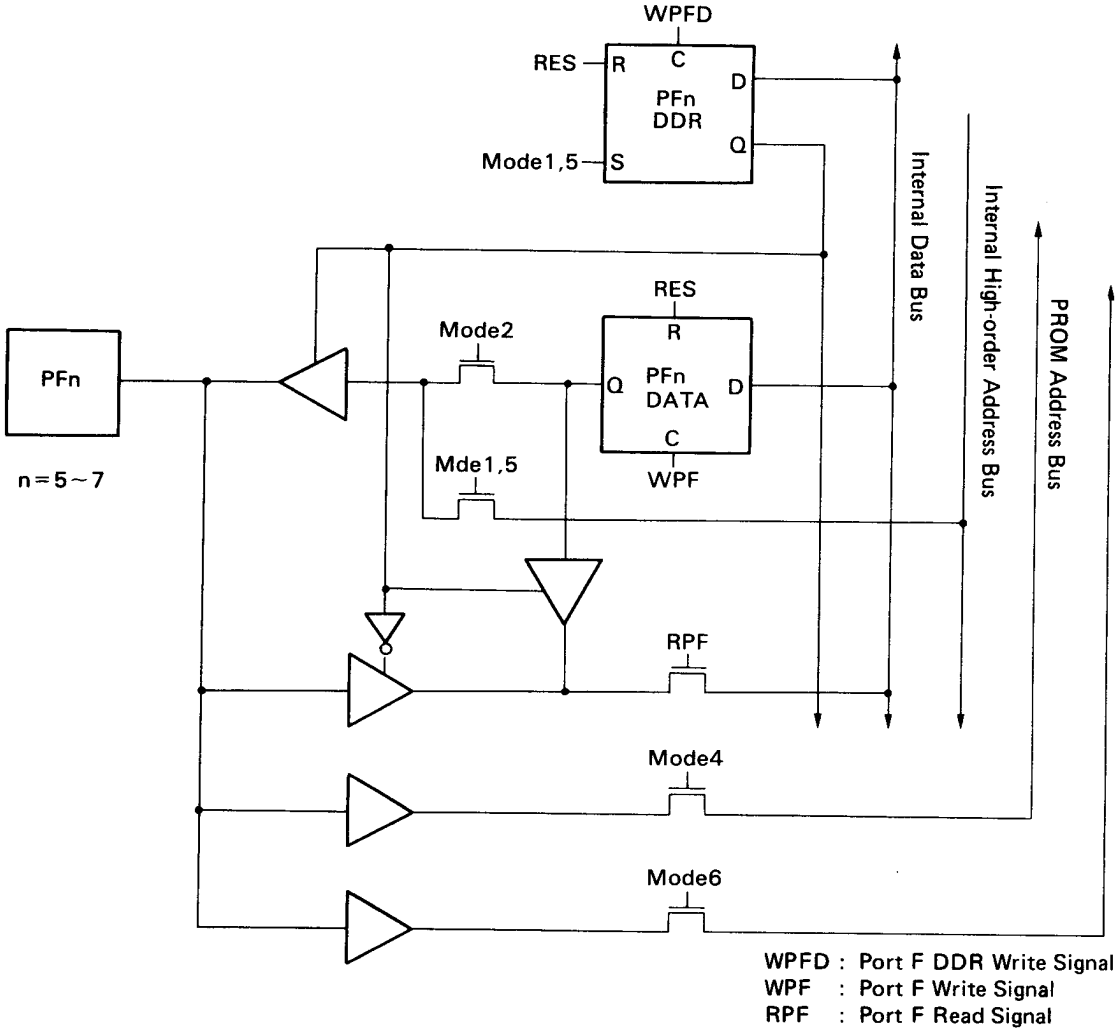


Fig. 27 PF_5 to PF_7 Block Diagram

- **Port G (PG_0 to PG_7)**

Port G is an 8-bit I/O port. In the single chip modes (Mode 2 and Mode 6), each pin on this port is configured as either an input or an output depending on the Port G Data Direction Register. It is programmed as an input when the corresponding DDR bit is cleared to a logic 0, and as an output when it is set to a logic 1. The MCU reset clears the Port G DDR, which configures Port G to be an input port.

In the external extended modes (Mode 1 and Mode 5) Port G provides data bus lines (D_0 to D_7).

Port G Data Register is cleared during reset.

Each pin on Port G can drive one TTL load and 90 pF.

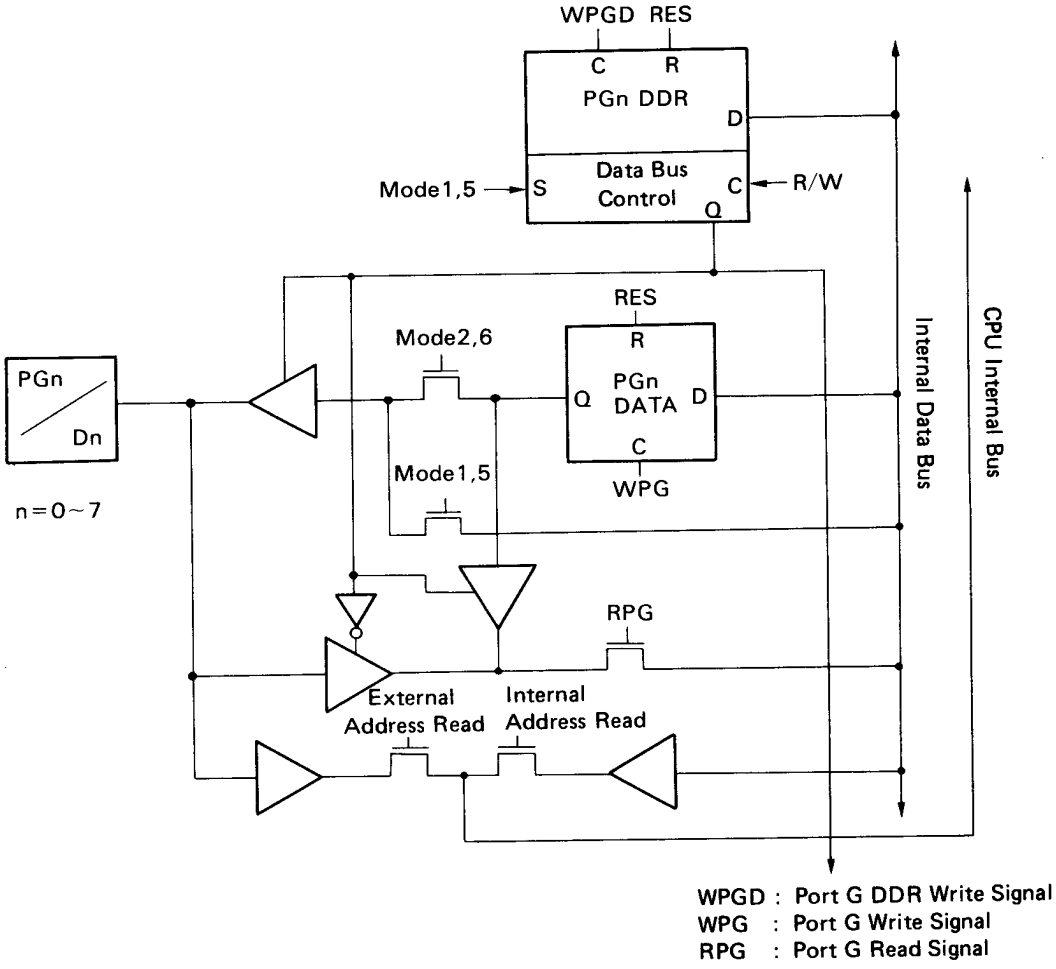


Fig. 28 Port G Block Diagram

Port G Data Direction Register ($000C) (Write only. Cleared to $00 during reset)

7	6	5	4	3	2	1	0
PG_7 DDR	PG_6 DDR	PG_5 DDR	PG_4 DDR	PG_3 DDR	PG_2 DDR	PG_1 DDR	PG_0 DDR

Port G Data Register ($0006) (R/W. Cleared to $00 during reset)

PG_7	PG_6	PG_5	PG_4	PG_3	PG_2	PG_1	PG_0

- **Port H (PH_0 to PH_4)**

Port H is an 8-bit I/O port. In the master MCU modes (Mode 1, Mode 2, Mode 5) each pin on this port is configured as either an input or an output depending on the Port H Data Direction Register. It is programmed as an input when the corresponding DDR bit is cleared to a logic 0, and as an output when it is set to a logic 1.

The MCU reset, clears the Port H Data Register and Port H Data Direction Register, which configures this port as an input port.

In the slave MCU mode (Mode 6) Port H provides data bus lines for the Parallel Communication Interface (DBB_0 to DBB_7). Each data line is configured as either an input or an output by parallel communication control signal.

In the PROM programming mode (Mode 4) only PH_7 is used as a PROM address input (EA_5).

Each pin on Port H is capable of driving one TTL load and 90 pF.

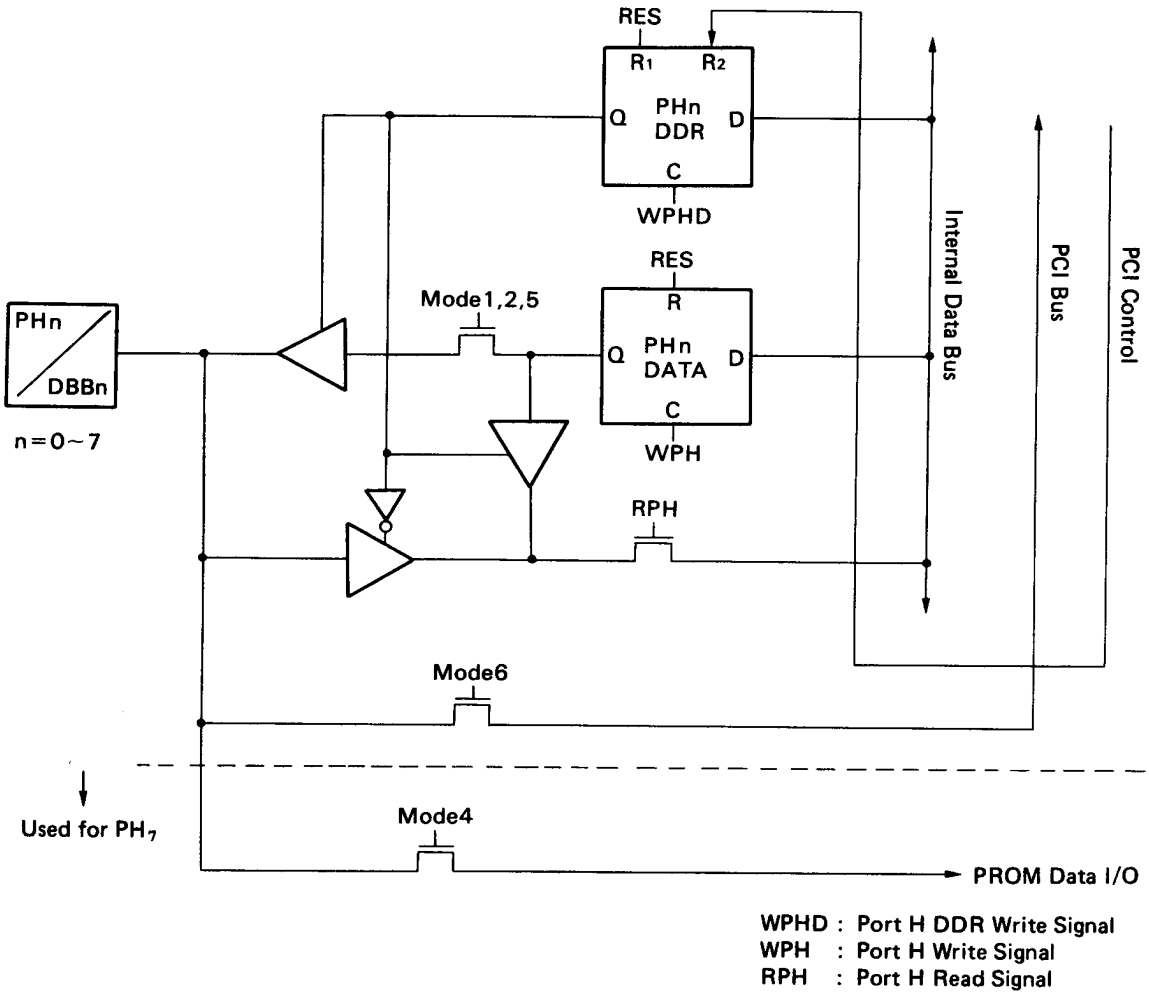


Fig. 29 Port H Block Diagram

Port H Data Direction Register ($000D) (Write only. Cleared to $00 during reset)

7	6	5	4	3	2	1	0
PH7 DDR	PH6 DDR	PH5 DDR	PH4 DDR	PH3 DDR	PH2 DDR	PH1 DDR	PH0 DDR

Port H Data Register ($0007) (R/W. Cleared to $00 during reset)

PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0

- **Port J (PJ_0 to PJ_4)**

Port J is a 5-bit I/O port. When a pin on Port J provides a data I/O line, it is configured as either an input or an output depending on the state of corresponding bit in the Port J Data Direction Register ($000E). It is programmed as an input when the corresponding Data Direction Register bit is cleared to a logic 0, and as an output when it is set to a logic 1. During reset the Port J Data Direction Register is cleared, which configures Port J as an input port.

Port J Data Register are also cleared during reset.

Port J Data Direction Register ($000E) (Write-only. Bit 0 to 4 are Cleared during reset)

7	6	5	4	3	2	1	0
—	—	—	PJ_4 DDR	PJ_3 DDR	PJ_2 DDR	PJ_1 DDR	PJ_0 DDR
(—	—	—	0	0	0	0	0)

Port J Data Register ($0008) (R/W. Bit 0 to 4 are cleared during reset)

—	—	—	PJ_4	PJ_3	PJ_2	PJ_1	PJ_0
(—	—	—	0	0	0	0	0)

Functions of the Port J depend on the operating mode as listed in Table 10.

Table 10 Port J Configuration in Five Operation Modes

No.	Mode	PJ_0	PJ_1	PJ_2	PJ_3	PJ_4
1	Master MCU External Extended Mode	I/O	I/O	I/O	E (O)	R/$\overline{W}$ (O)
2	Master MCU Single-Chip Mode	I/O	I/O	I/O	I/O	I/O
4	PROM Programming Mode	EA_3 (I)	EA_2 (I)	EA_1 (I)	EA_0 (I)	EA_6 (I)
5	Master MCU External Extended Mode	I/O	I/O	I/O	E (O)	R/W (O)
6	Slave MCU Single-chip Mode	$\overline{CS}$ (I)	RS_0 (I)	$\overline{RDY}$ (O)	$\overline{OE}$ (I)	$\overline{WE}$ (I)

(NOTE) PJ_2 is an N-MOS open drain output.

Mode 1 (PJ_0 to PJ_2, E, R/$\overline{W}$)

In the Mode 1 (master MCU external extended mode) Port J provides three I/O lines and two output lines (E and R/$\overline{W}$). Port J DDR bits which correspond to theI/O lines are cleared during reset, which configures the lines as inputs. The other Port J DDR bits are forced to be set, causing PJ_3 and PJ_4 to output control signals E and R/$\overline{W}$ respectively.

Mode 2 (PJ_0 to PJ_4)

In the Mode 2 (Master MCU Single-chip Mode) Port J provides five I/O lines. Reset clears the Port J DDR, which configures this port as an input port.

Mode 4 (EA_3, EA_2, EA_1, EA_0, EA_6)

In the Mode 4 (PROM Programming Mode) Port J is configured as a PROM address input port.

Mode 5 (PJ_0 to PJ_2, E, R/$\overline{W}$)

In the Mode 5 (Master MCU External Extended Mode) Port J provides three I/O lines and two control signal output lines (E and R/$\overline{W}$). The configuration of each port pin is the same as in Mode 1; however in the Mode 1 the PROM is enabled, while in this mode the internal PROM is disabled.

Mode 6 ($\overline{CS}$, RS_0, $\overline{RDY}$, $\overline{OE}$, $\overline{WE}$)

In the Mode 6 (Slave MCU Single-chip Mode) each pin on Port J is configured as a control signal input or output.

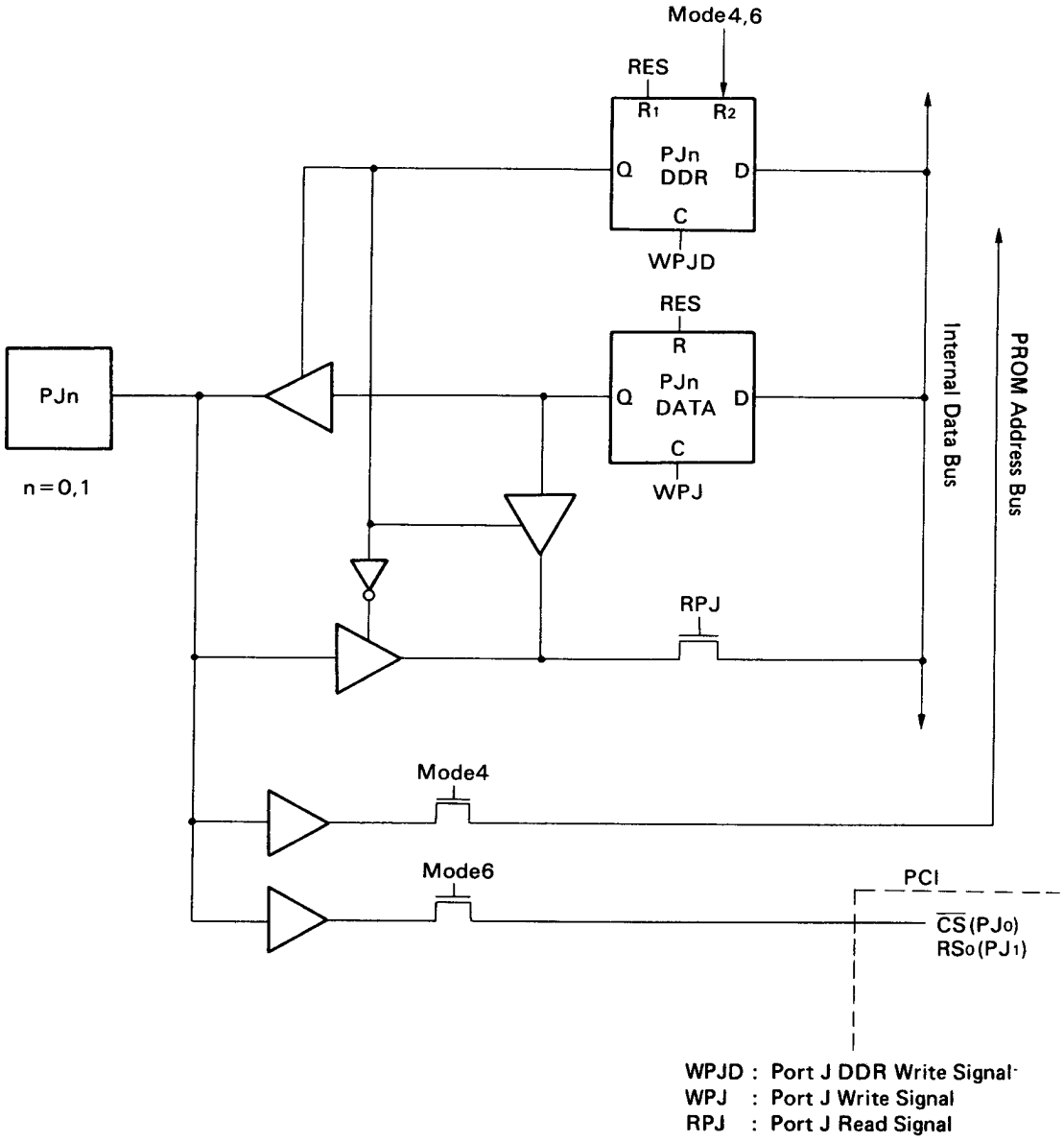


Fig. 30 PJ_0 and PJ_1 Block Diagram

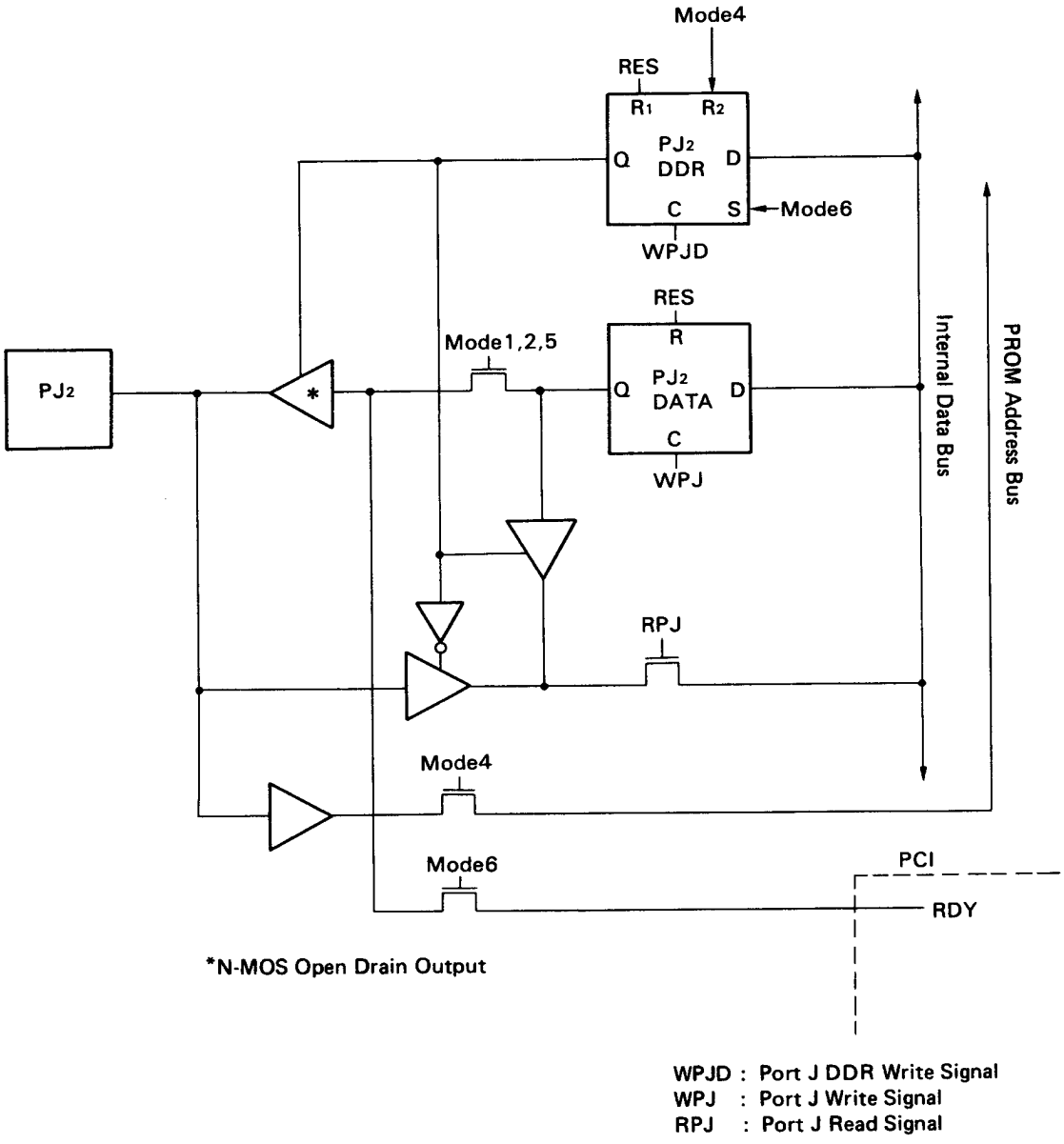


Fig. 31 PJ_2 Block Diagram

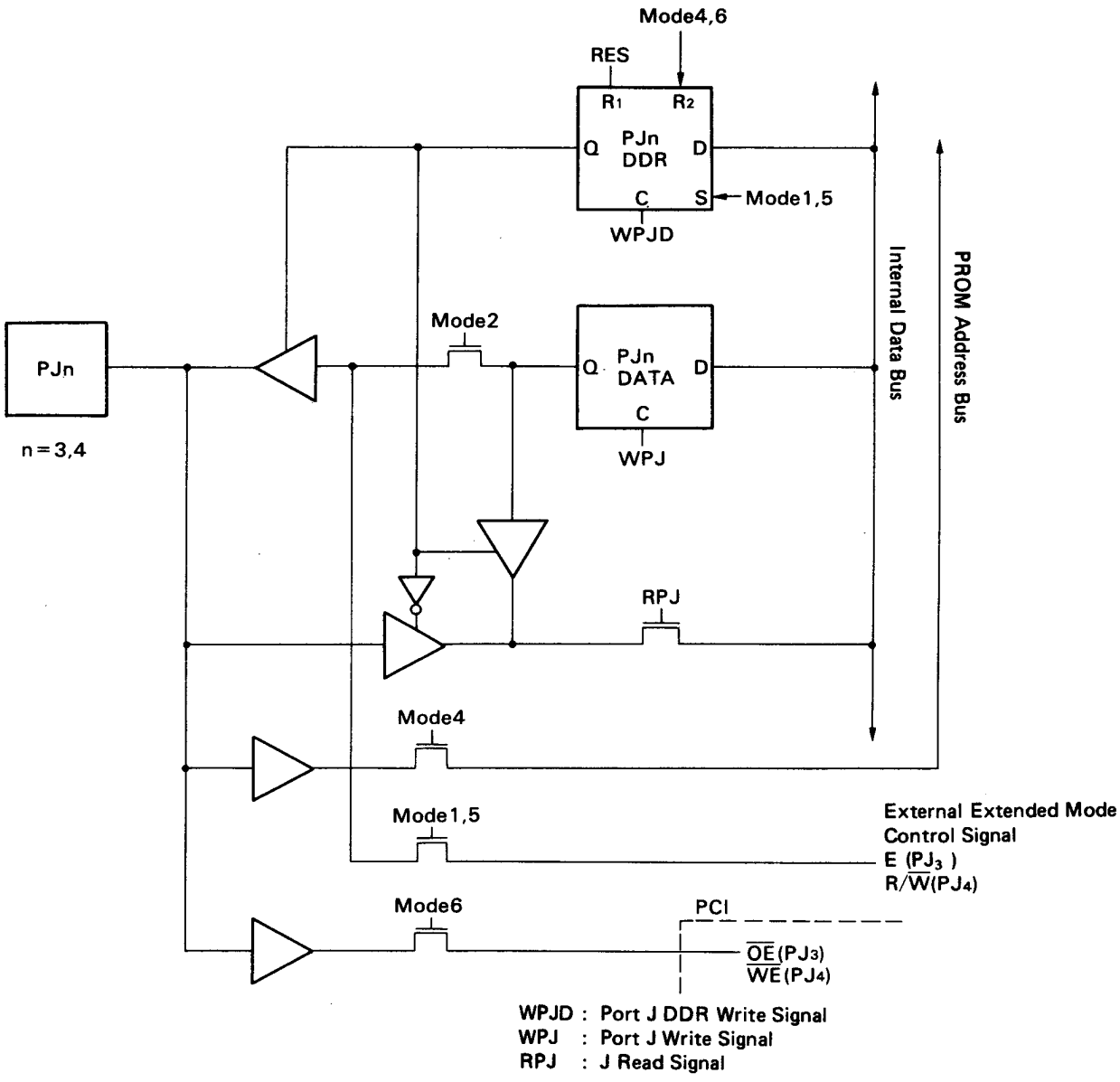


Fig. 32 PJ_3 and PJ_4 Block Diagram

■ TIMER

The HD63705Z0 provides a 16-bit PWM timer and five 16-bit timers, which perform input waveform measurement and generate output waveforms.

• PWM Timer

PWM timer provides pulses with resolutions of 13 to 16-bit which divide clock inputs within the range of divide-by-2^{13} to divide-by-2^{16}. PWM timer includes the followings.

- PWM Control and Status Register (3-bit)
- PWM Data Register (H) (8-bit)
- PWM Data Register (L) (8-bit)
- Free Running Counter (16-bit) (shared with Timer 1)

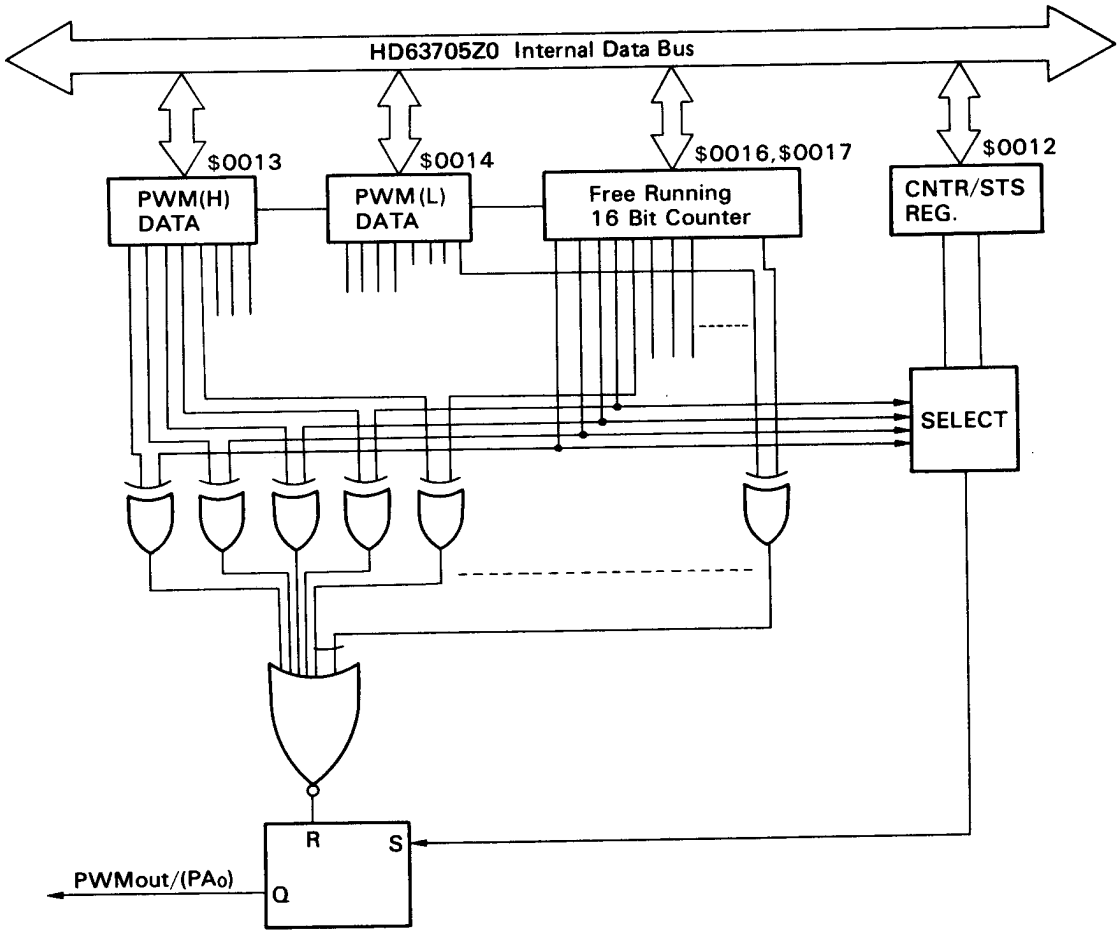


Fig. 33 PWM Timer Block Diagram

• PWM Control and Status Register (PWM CNTR : $0012)

The PWM Control and Status Register determines whether PA_0 is used as an output line or as the PWM timer output and selects the PWM cycle.

PWM Control and Status Register ($0012)

7	6	5	4	3	2	1	0	
PWM ENABLE	—	—	—	—	—	PW1	PW0	
(0	—	—	—	—	—	0	0)	At Reset

Bit 0, 1, and 7 are cleared during reset and are both readable and writable.

Bit 0, Bit 1 (PW0, PW1)
These bits determine the PWM cycle.

PW1	PW0	Resolution	Division	PWM cycle		Data register (bit)	Output duty
				f = 1 MHz	f = 2 MHz		
0	0	16 bit	$\div 2^{16}$	65.5 ms	32.8 ms	15 to 0	0 to 65535/65536
0	1	15 bit	$\div 2^{15}$	32.8 ms	16.4 ms	14 to 0	0 to 32767/32768
1	0	14 bit	$\div 2^{14}$	16.4 ms	8.19 ms	13 to 0	0 to 16383/16384
1	1	13 bit	$\div 2^{13}$	8.19 ms	4.1 ms	12 to 0	0 to 8191/8192

Bit 7 : (PWM ENABLE)
0 : Data Output Line
1 : PWM Output

- **PWM Data Register (PWM DATA : $0013(H), $0014(L))**

The PWM Data Register consists of 8 high-order bits and 8 low-order bits.

When the PWM timer provides a resolution of 13, 14 or 15 bis, unnecessary high-order bits are ignored.

PWM DATA (H: $0013)

15 (7)	14 (6)	13 (5)	12 (4)	11 (3)	10 (2)	9 (1)	8 (0)
PWM D_{15}	PWM D_{14}	PWM D_{13}	PWM D_{12}	PWM D_{11}	PWM D_{10}	PWM D_9	PWM D_8

These write-only bits are not initialized at reset.

PWM DATA (L: $0014)

7	6	5	4	3	2	1	0
PWM D_7	PWM D_6	PWM D_5	PWM D_4	PWM D_3	PWM D_2	PWM D_1	PWM D_0

The PWM Data Register consists of two bytes ($0013, $0014). Write operation must be to the high-order byte ($0013) first and then the low-order byte ($0014).

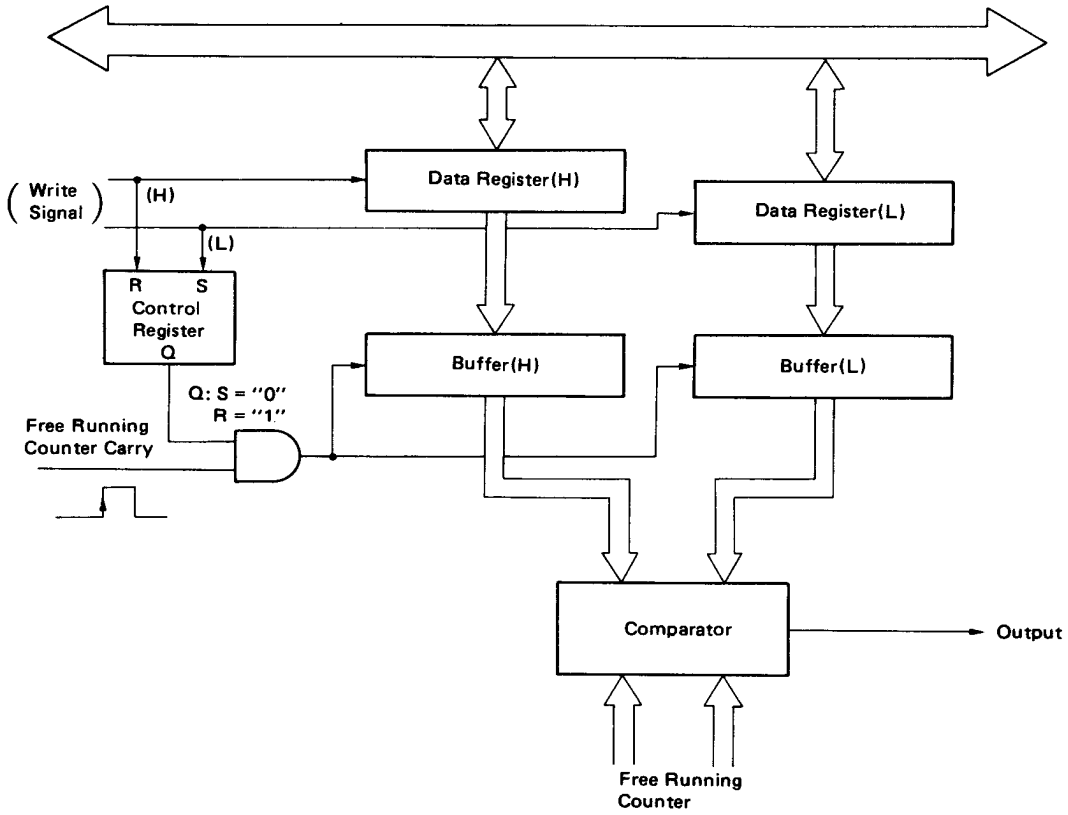


Fig. 34 PWM Data Register Block Diagram

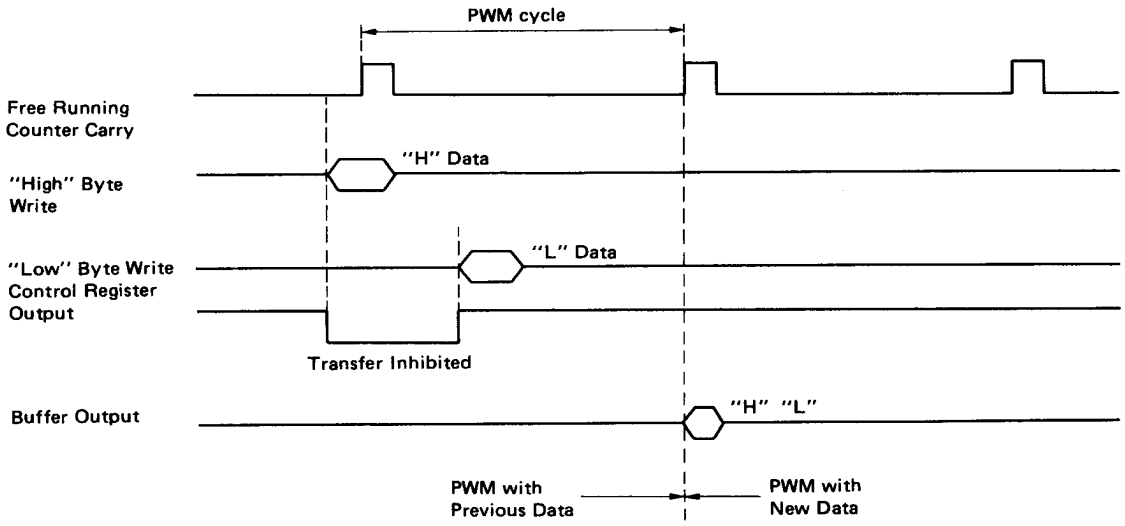


Fig. 35 PWM Data Transfer Timing

A carry in the Free Running Counter ($FFFF→$0000), after a write to the PWM Data Register low-order byte, transfers data to the buffer and begins pulse width modulation (PWM).

(NOTE) When writing the PWM Data Register low-order byte simultaneously with a carry generation, the data is not transferred.
A write to only the low-order byte can cause pulse width modulation. In this case, the previous data is transferred from the PWM Data Register high-order byte to the buffer.
The PWM Data Register is a write-only register and it is not initialized at reset.

The PWM starts when the Free Running Counter carry is generated. To start PWM soon after a carry generation, the Free Running Counter value should be changed by software. The logic high pulse width indicates the contents of the PWM data register.

Each resolution provides a pulse as shown in Fig. 36.

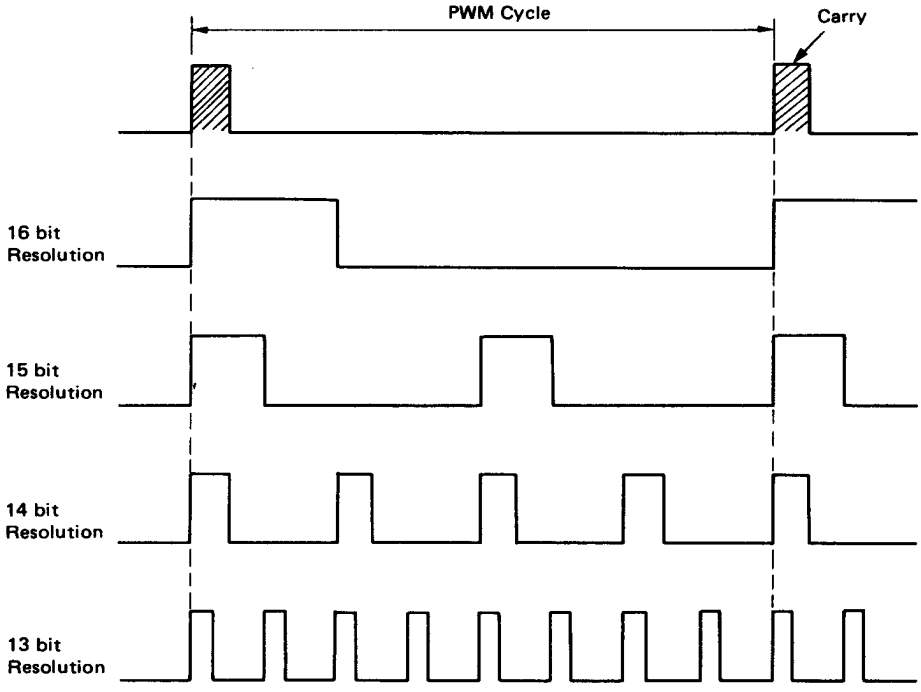


Fig. 36 PWM Output for Each Resolution

PWM timing diagram is as shown in Fig. 37.

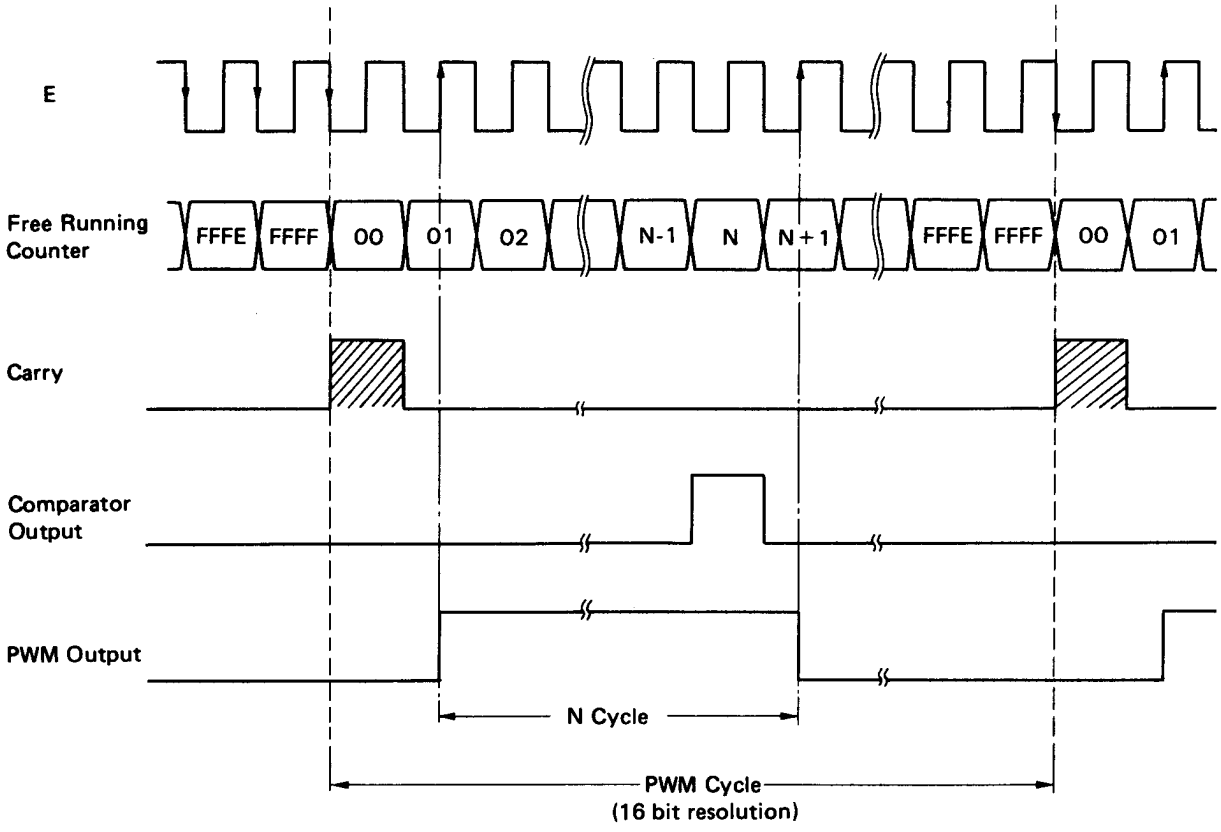


Fig. 37 PWM Timing Diagram

- **Timer 1**

The HD63705Z0 has a 16-bit programmable timer which performs input waveform measurement while independently generating an output waveform. Pulse widths for both input and output waveforms can vary from several micro seconds to many seconds.

Timer 1 consists of the followings.

- 8-bit Control and Status Register
- 16-bit Free Running Counter
- 16-bit Output Compare Register
- 16-bit Input Capture Register

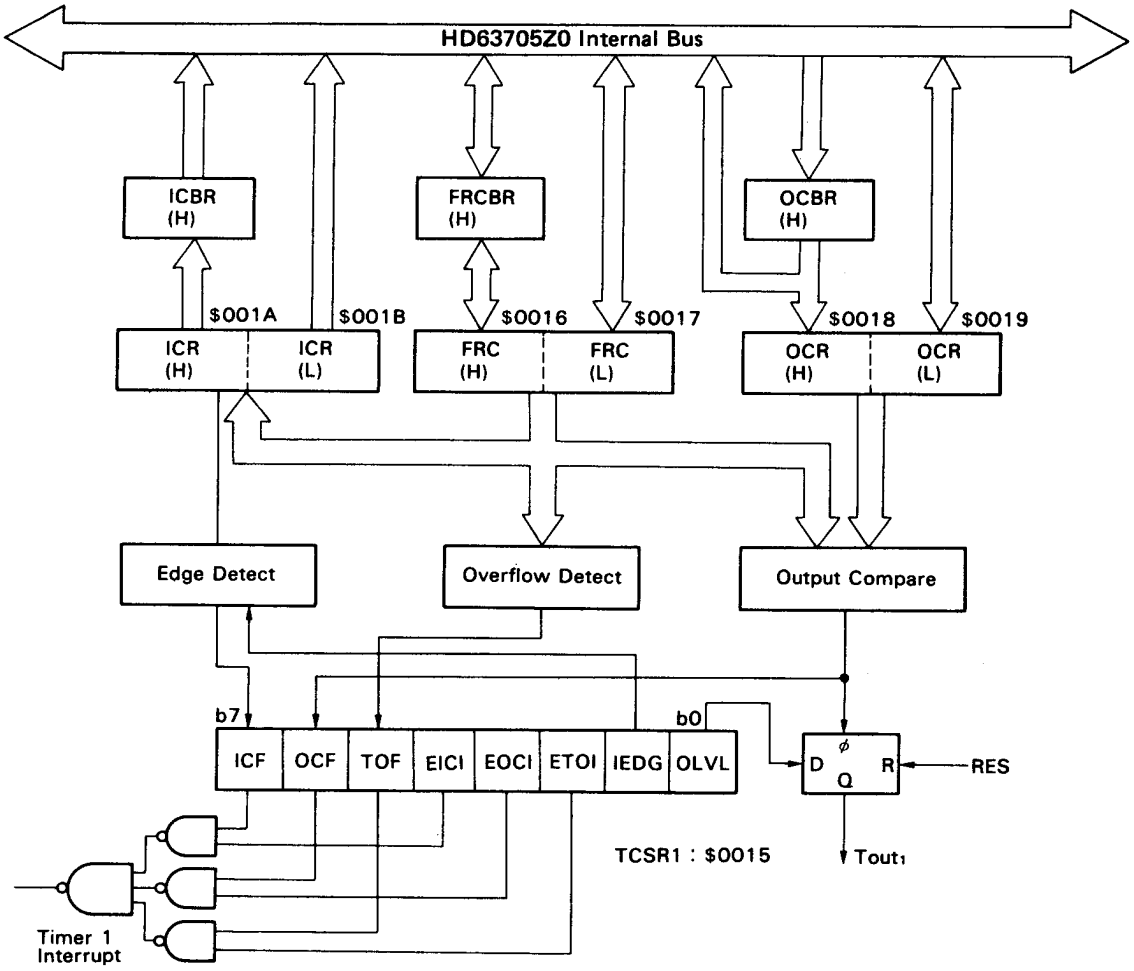


Fig. 38 Timer 1 Block Diagram

- **Free Running Counter (FRC) ($0016, $0017)**

The key element of the programmable timer is a 16-bit Free Running Counter which is clocked and incremented by system clock (E). Software can read the counter without affecting its value. The counter value is cleared during reset.

A CPU write to the high-order byte ($0016) causes its value to be stored in a buffer (FRCBR). Then the buffer is accessed when writing the low-order byte ($0017) and both the bytes are transferred to the Free Running Counter.

A CPU read of the low-order byte ($0017) causes the high-order byte to be transferred to the buffer. The buffer is accessed when reading the high-order byte ($0016) and a read of the FRC value is completed.

(NOTE) Unless the low-order byte has been read, the previous data is obtained by a read of the high-order byte.

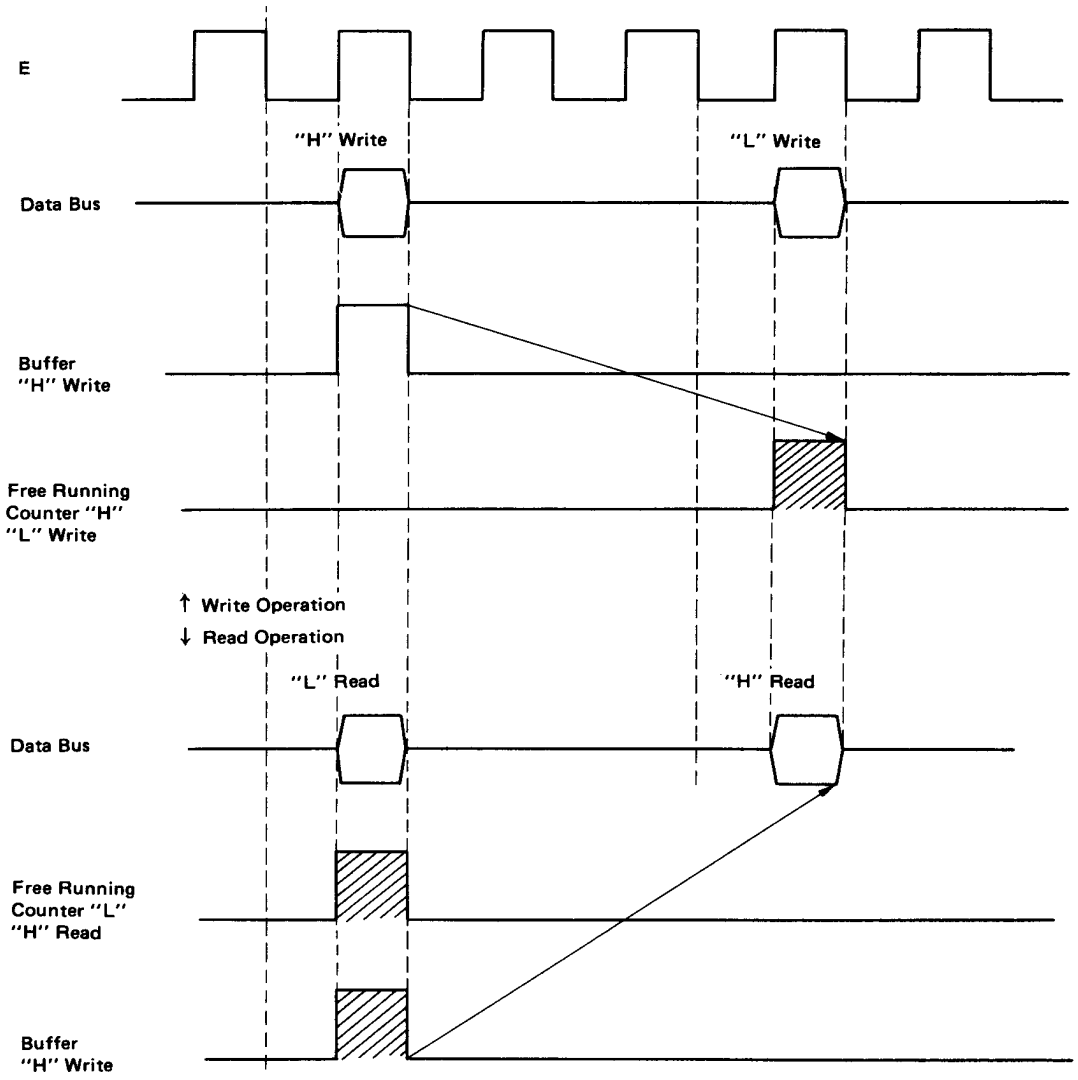


Fig. 39 Free Running Counter Timing

- **Timer Control and Status Register 1 (TCSR1) ($0015)**

The Timer Control and Status Register 1 is an 8-bit register where all bits are readable and the lowest 5 bits can be written. The highest 3 bits provide the following read-only status information.

Bit 5 : The Free Running Counter has overflowed. (TOF: Timer Overflow Flag)

Bit 6 : A match has occurred between the Free Running Counter and the Output Compare Register. (OCF: Output Compare Flag)

Bit 7 : A proper level transition has taken place at PC_7 with an accompanying transfer of the Free Running Counter to the Input Capture Register. (ICF: Input Capture Flag)

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0015
Read only			R/W					

The TCSR1 is illustrated above and each bit is defined as follows:

Bit 0 OLVL (Output Level)

When a match occurs between the Free Running Counter and the Output Compare Register, OLVL value appears at PC_6 ($Tout_1$) if bit 6 in the Miscellaneous Register (Misc: $003B) is set.

OLVL = 0 : low level is output.
OLVL = 1 : high level is output.

Bit 1 IEDG (Input Edge)

This bit determines which level transition at PC_7/Tin_1 will trigger a Free Running Counter transfer to the Input Capture Register. The PC_7 is controlled by bit 7 in the Miscellaneous Register (Misc: $003B). If the bit 7 is set, PC_7 will function as an input of Timer 1.

IEDG = 0 : triggered on a negative edge ("High" to "Low")
IEDG = 1 : triggered on a positive edge ("Low" to "High")

Bit 2 ETOI (Enable Timer Overflow Interrupt)

If this bit is set, a Timer 1 interrupt is enabled whenever TOF is set. If cleared, the interrupt is inhibited.

ETOI = 0 : TOI is inhibited.
ETOI = 1 : TOI is enabled.

Bit 3 EOCI (Enable Output Compare Interrupt)

If this bit is set, a Timer 1 interrupt is enabled whenever OCF bit is set. If cleared, the interrupt is inhibited.

EOCI = "0" : OCI is inhibited.
EOCI = "1" : OCI is enabled.

Bit 4 EICI (Enable Input Capture Interrupt)

If this bit is set, a Timer 1 interrupt is enabled whenever ICF bit is set. If cleared, the interrupt is inhibited.

EICI = "0" : ICI is inhibited.
EICI = "1" : ICI is enabled.

Bit 5 TOF (Time Overflow Flag)

This read-only bit is set when the Free Running Counter is incremented by 1 from $FFFF. It is cleared by reading TCSR1 with TOF set and then the Free Running Counter low-order byte ($0017).

Bit 6 OCF (Output Compare Flag)

This read-only bit is set when a match has occurred between the Output Compare Register and the Free Running Counter. It is cleared by reading TCSR1 with OCF set and then writing the Output Compare Register low-order byte ($0019).

Bit 7 ICF (Input Capture Flag)

This read-only bit is set when the Free Running Counter contents are transferred to the Input Capture Register on a proper edge of the Tin_1 input signal. (See Bit 1). It is cleared by reading TCSR1 with ICF set followed by reading the Input Capture Register low-order byte ($001B).

- **Output Compare Register (OCR) ($0018, $0019)**

The Output Compare Register is a 16-bit read/write register used to control an output waveform. It is always compared with the Free Running Counter.

When a match is found between them, the Output Compare Flag (OCF) bit in the Timer Control and Status Register 1 (TCSR1) is set and the value of Output Level (OLVL) bit appears at PC_6 if the PC_6 is configured as the output of Timer 1 ($Tout_1$) by setting the bit 6 in the Miscellaneous Register (Misc: $003B).

The values of the Output Compare Register and the OLVL bit must be changed after each comparison to control an output waveform. The Output Compare Register is set to $FFFF during reset.

The user must write the Output Compare Register high-order byte ($18) and then the low-order byte ($19).

The output compare function is inhibited during a write cycle to the high-order byte of the Output Compare Register or the Free Running Counter.

(NOTE) While the output compare function is inhibited, a match between the Output Compare Register and the Free Running Counter does not affect outputs.

- **Input Capture Register (ICR) ($001A, $001B)**

The Input Capture Register is a 16-bit read-only register used to latch the value of the Free Running Counter when a defined input transition is sensed by the input capture edge detector. The level transition which triggers a Free Running Counter transfer is determined by the Input Edge Bit (IEDG) in the TCSR1.

Setting bit 7 in the Miscellaneous Register enables the Tin_1, which allows an external signal to input to the edge detector and the Free Running Counter is transferred to the Input Capture Register on the edge determined by IEDG. Setting the bit selects a negative edge and clearing it selects a positive edge.

The input pulse width must be at least 2 system cycles. The Input Capture Register is cleared to $0000 during reset.

A CPU read of the Input Capture Register low-order byte ($001B) causes the high-order byte to be transferred to the buffer. Then the buffer is accessed when reading the high-order byte ($001A) and thus a read of the Input Capture Register is completed.

(NOTE) Unless the low-order byte is read, the previous data is obtained by a read of the high-order byte.

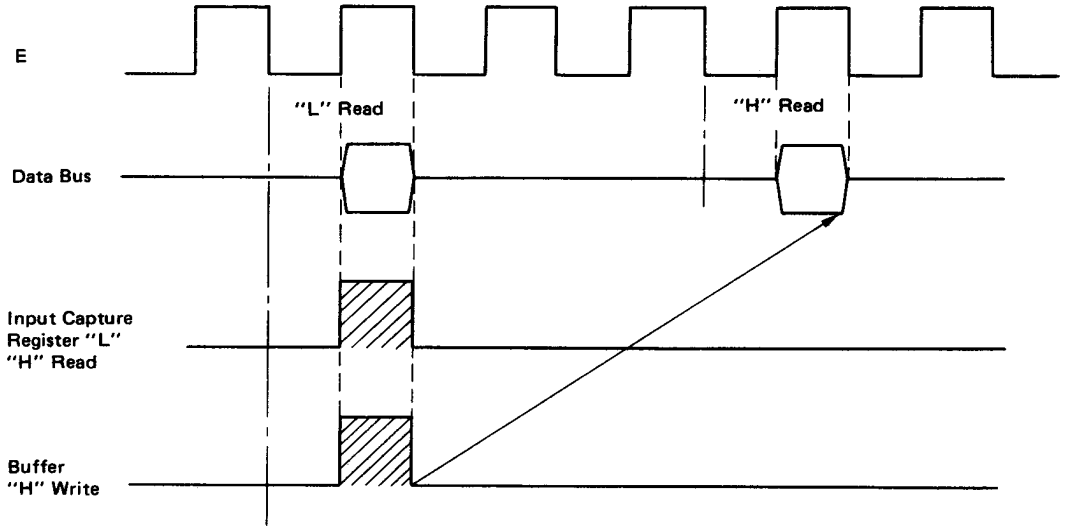


Fig. 40 Input Capture Timing

- **Timer 2/Timer 3**

The HD63705Z0 also provides the Timer 2 and the Timer 3, which generate independent output waveforms.

Timer 2 is a 16-bit reloadable timer which can count external events.

Timer 3 is a 16-bit reloadable timer which counts only the system clock E.

Each timer individually consists of the followings.

Timer 2
- 6-bit Control and Status Register 2
- 16-bit Up Counter
- 16-bit Timer Constant Register

Timer 3
- 4-bit Control and Status Register 3
- 16-bit Up Counter
- 16-bit Time Constant Register

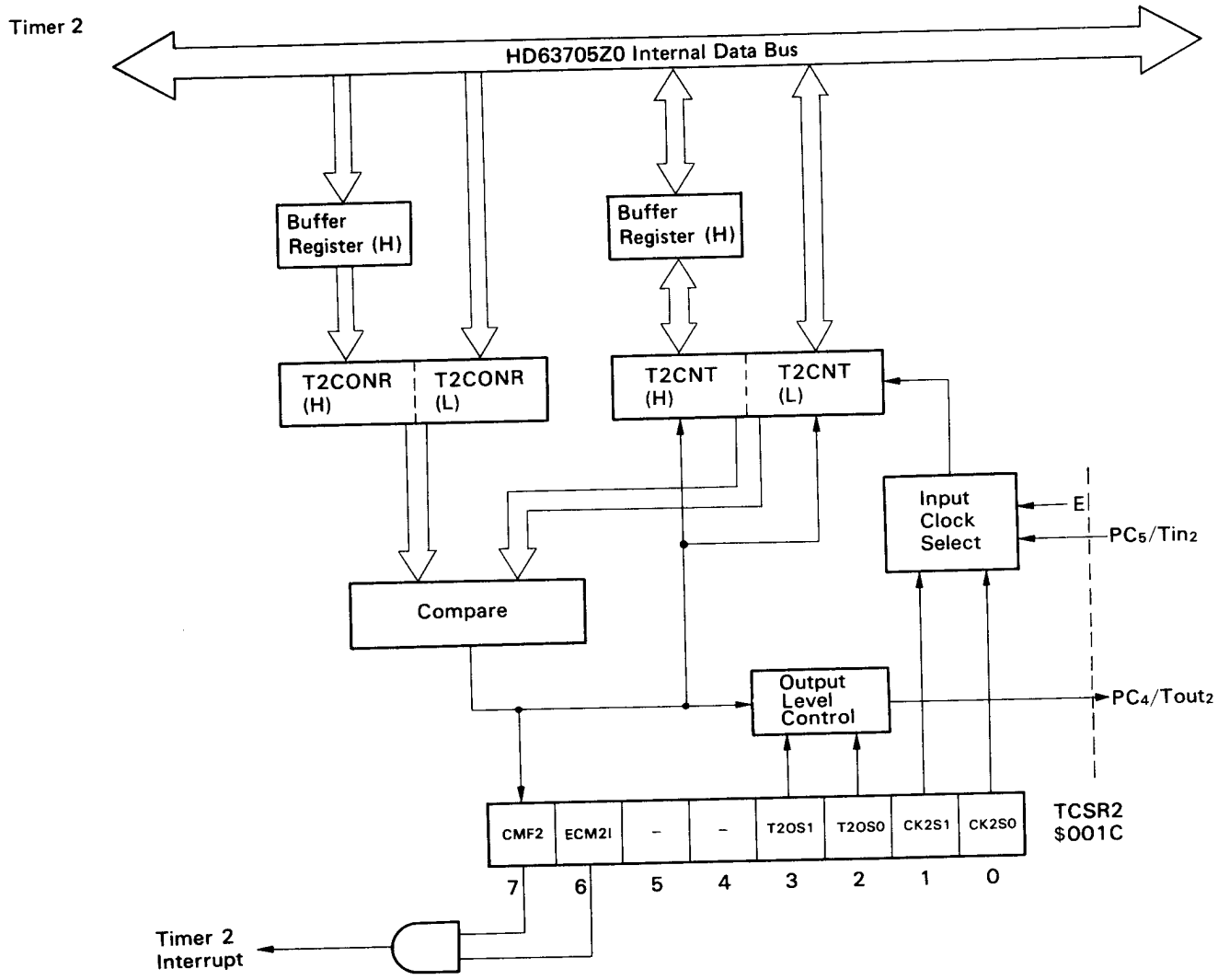


Fig. 41 Timer 2 Block Diagram

- **Timer 2 Up Counter (T2CNT: $001D(H), $001E(L))**

This is a 16-bit Up Counter, which is clocked by the signal defined by CK2S0 and CK2S1 of the TCSR2. Software can read the Timer 2 Up Counter without affecting its value at any time, and can write any value to the counter during count operation.

The Timer 2 Up Counter is cleared during reset when a match occurs between the counter and the Timer 2 Time Constant Register. If a write is made to the Timer 2 Up Counter during the clear cycle, the counter is not cleared but a data is written into the counter.

A CPU write to the Timer 2 Up Counter high-order byte ($001D) causes its value to be transferred to a buffer (T2CNTBR). The buffer is accessed when writing the Timer 2 Up Counter low-order byte ($001E) and thus both the bytes are transferred to the Timer 2 Up Counter.

A CPU read of the low-order byte ($001E) causes the high-order byte to be transferred to the buffer. The buffer is accessed when reading the high-order byte ($001D) and thus a read of the Timer 2 Up Counter is completed.

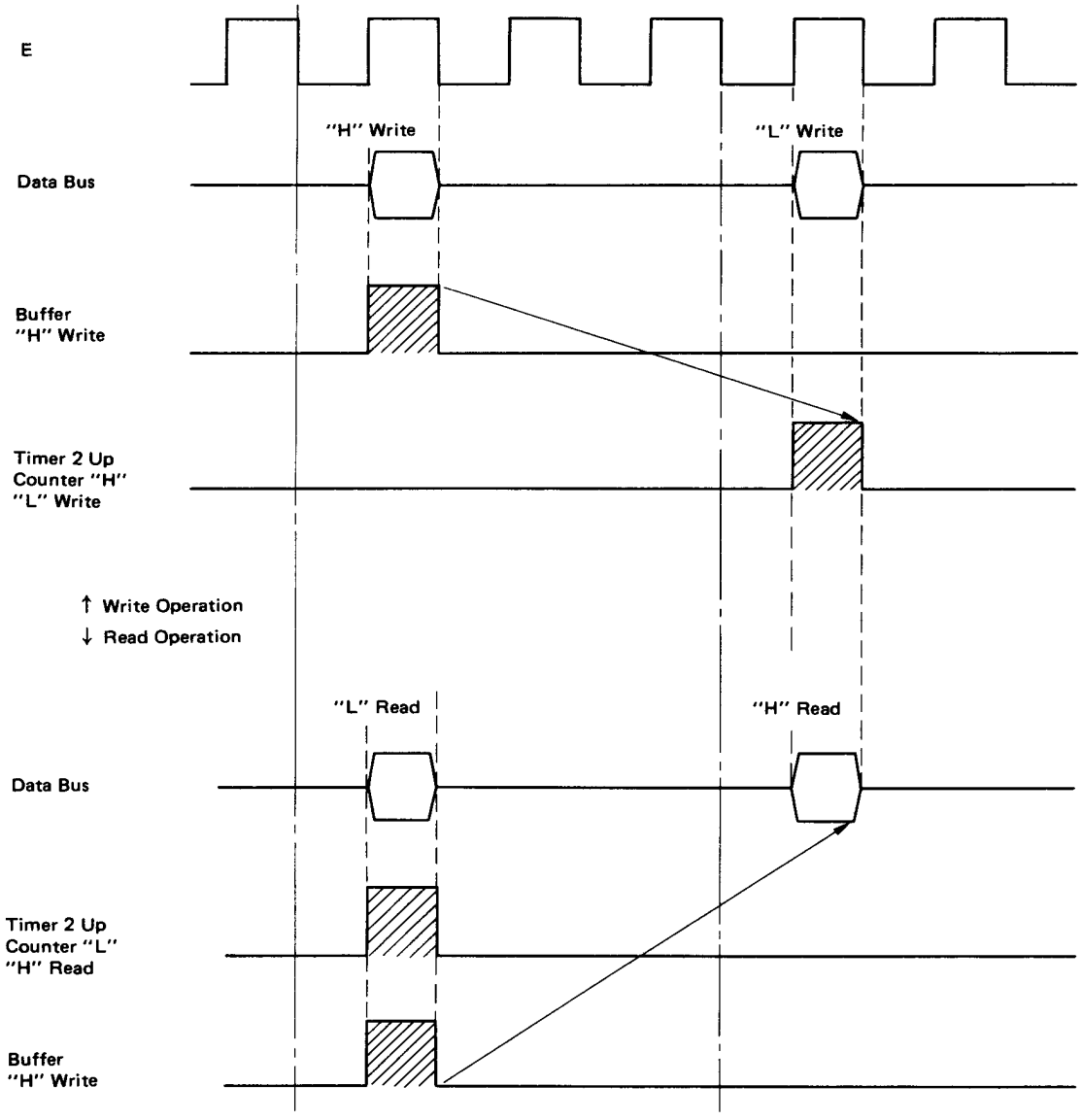


Fig. 42 Timer 2 Up Counter Timing

- **Timer 2 Time Constant Register (T2CONR: $001F(H), $0020(L))**

The Timer 2 Time Constant Register is a 16-bit write-only register. It is always compared with the Timer 2 Up Counter.

When a match is found between them, bit 7 (Count Match Flag 2 (CMF2)) in the Timer Control and Status Register 2 (TCSR2) is set and the value selected by bit 2 and 3 (Timer 2 output select 0, Timer 2 output select 1 (T2OS0, T2OS1)) in the TCSR2 will appear at PC_4 ($Tout_2$).

The Timer 2 Up Counter is cleared at the same time that the CMF2 is set and then counts from $0000. This enables regular interrupts and waveform outputs without any software support. The Timer 2 Time Constant Register is set to $FFFF at reset.

Timer 2 Time Constant Register consists of two bytes. After a write to the high-order byte ($001F), a write to the low-order byte ($0020) must be made. If only a write to the low-byte is made, the previous high-order byte data is automatically written to the high-order byte.

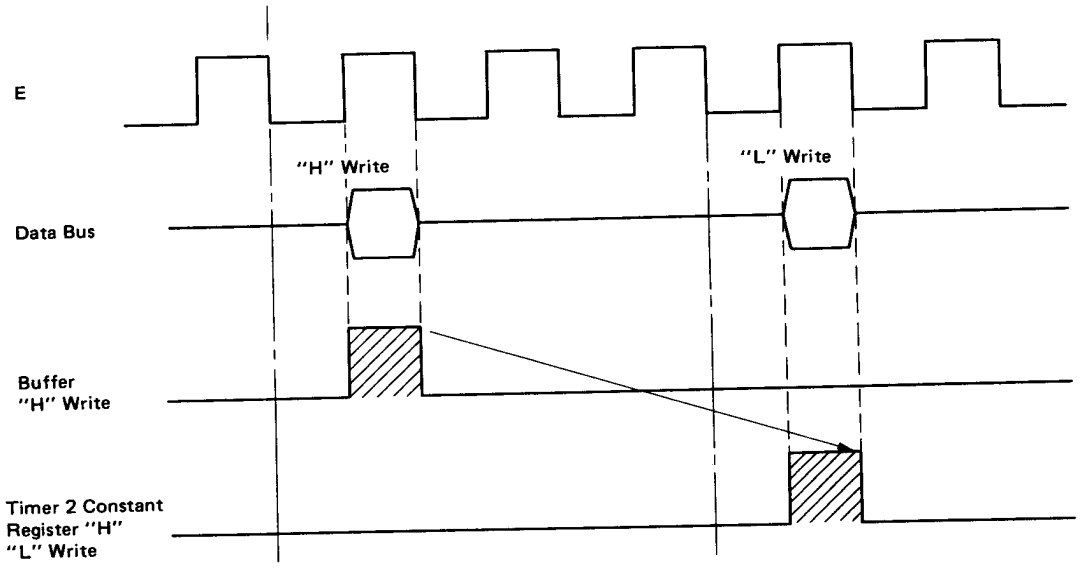


Fig. 43 Timer 2 Time Constant Register Timing

- **Timer Control and Status Register 2 (TCSR2: $001C)**

Timer Control and Status Register 2 is a 6-bit register. All bits are readable and the bits except for CMF2 bit (bit 7) are writable.

Each bit of the TCSR2 is cleared at reset.

The followings are descriptions of individual bits.

7	6	5	4	3	2	1	0	
CMF2	ECM2I	—	—	T2OS1	T2OS0	CK2S1	CK2S0	TCSR2 : $001C
R	R/W	—	—	R/W	R/W	R/W	R/W	
(0	0	—	—	0	0	0	0)	At Reset

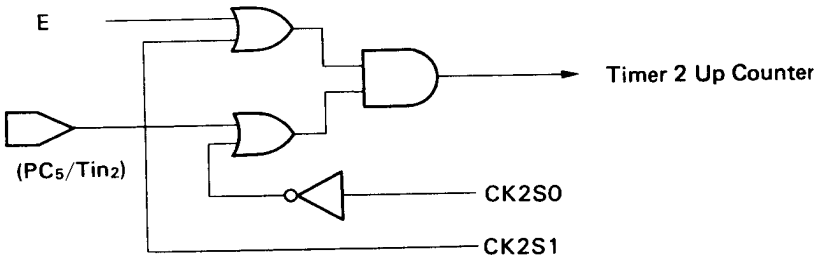
Bit 0 CK2S0 (Input Clock 2 Select 0)
Bit 1 CK2S1 (Input Clock 2 Select 1)

Input clock to the Timer 2 Up Counter is selected as shown below depending on the state of these two bits. When an external clock is selected, PC_5 will be configured as an input of the Timer 2 (Tin_2), where the counter increments on the positive edge of the external clock.

The external clock can be counted up to half of the system clock frequency.

CK2S1	CK2S0	Input Clock to the Counter	PC_5/Tin_2
0	0	System Clock (E)	PC_5 (I/O)
0	1	System Clock Controlled by Tin_2	*Tin_2 (I)
1	0	No Clock Input (stops counting)	PC_5 (I/O)
1	1	External Clock	*Tin_2 (I)

*When PC_5 is used as an I/O line after it functions as a timer input, the port pin is configured as an input.



Bit 2 T2OS0 (Timer 2 Output Select 0)
Bit 3 T2OS1 (Timer 2 Output Select 1)

When a match occurs between the Timer 2 Up Counter and the Timer 2 Time Constant Register, the following outputs will appear at PC_4 depending on these two bits.

T2OS1	T2OS0	Timer Output	PC_4/Tout2
0	0	Timer Output Inhibited	PC_4 (I/O)
0	1	Toggle Output *1	Tout2 (O) *2
1	0	Output "0"	Tout2 (O) *2
1	1	Output "1"	Tout2 (O) *2

*1 Every time a match occurs between the T2CNT and the T2CONR, timer 2 output level is reversed or toggled. This leads to generation of a square waveform with 50% duty without any software support.

*2 When PC_4 is used as an I/O line after it functions as a timer output, the port pin is configured as an output.

Bit 4 Not Used

Bit 5 Not Used

Bit 6 ECM2I (Enable Count Match 2 Interrupt)

If this bit has been set, a Timer 2 interrupt is enabled whenever CMF2 is set. If cleared, the interrupt is inhibited.

ECM2I = "0" : the count match 2 interrupt is inhibited.

ECM2I = "1" : the count match 2 interrupt is enabled.

Bit 7 CMF2 (Count Match Flag 2)

This read-only bit is set when a match occurs between the Timer 2 Up Counter and the Timer 2 Time Constant Register.

CMF2 is cleared by writing a 0 to this bit when CMF2 is set. (Software cannot write a 1 to this bit.)

Timer 3

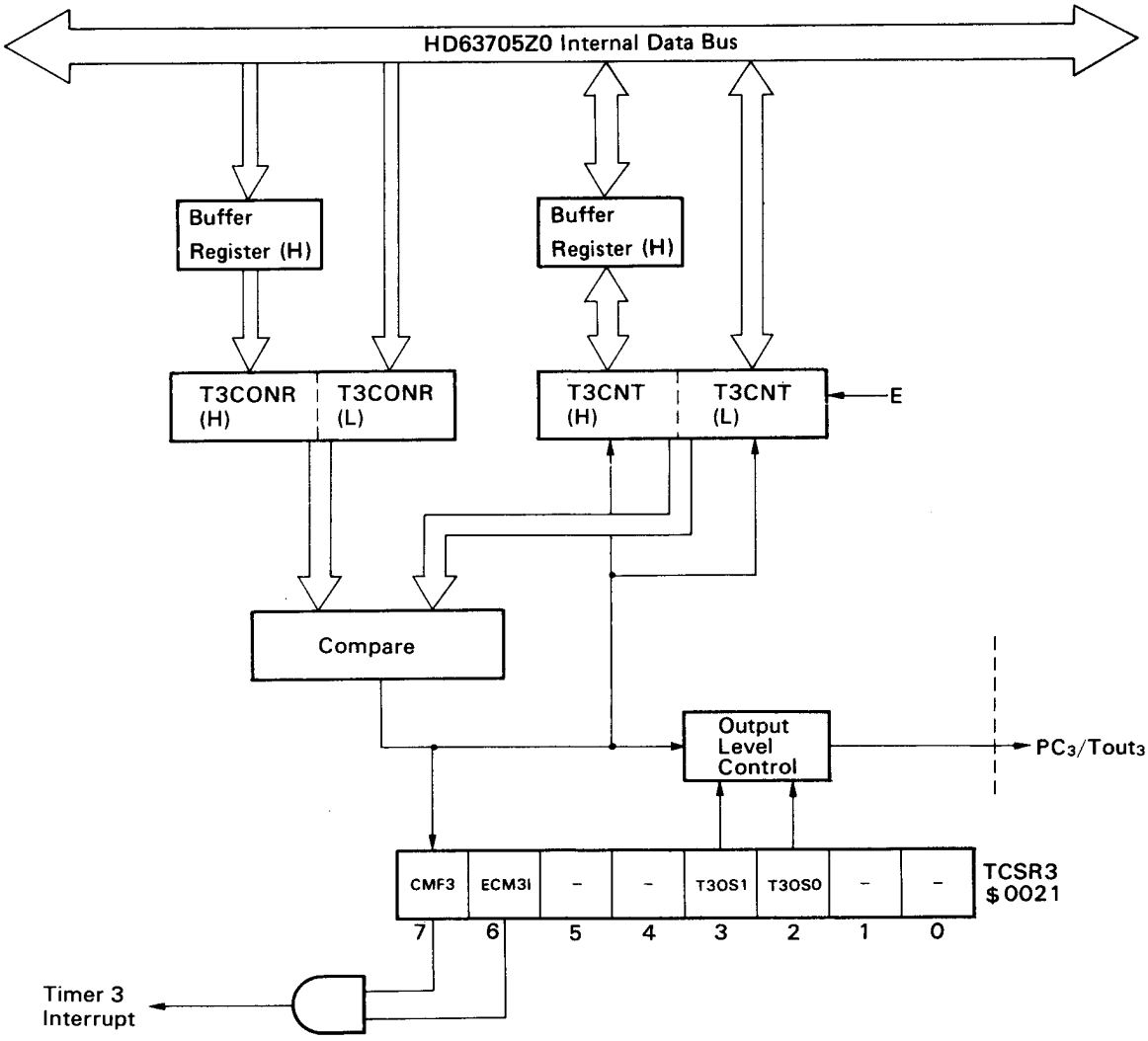


Fig. 44 Timer 3 Block Diagram

- **Timer 3 Up Counter (T3CNT: $0022(H), $0023(L))**

The Timer 3 Up Counter functions in the same way as the Timer 2 Up Counter except that it counts only the system clock (E).

T3CNT(H) : $0022 Timer 3 Up Counter (H)
T3CNT(L) : $0023 Timer 3 Up Counter (L)

- **Timer 3 Time Constant Register (T3CONR: $0024(H), $0025(L))**

Timer 3 Time Constant Register performs the same function as Timer 2 Time Constant Register.

T3CONR(H) : $0024 Timer 3 Time Constant Register (H)
T3CONR(L) : $0025 Timer 3 Time Constant Register (L)

- **Timer Control and Status Register 3 (TCSR3: $0021)**

Timer Control and Status Register 3 is a 4-bit register. All bits are readable and the bits except for CMF3 (bit 7) are writable.

The TCSR3 performs the same functions as the TCSR2 except that the TCSR3 cannot select the input clock to the Timer 3 Up Counter.

The followings are descriptions of individual pins.

7	6	5	4	3	2	1	0	
CMF3	ECM3I	—	—	T3OS1	T3OS0	—	—	TCSR3 : $0021
R	R/W	—	—	R/W	R/W	—	—	
(0	0	—	—	0	0	—	—)	At Reset

Bit 0 Not Used
Bit 1 Not Used

Bit 2 T3OS0 (Timer 3 Output Select 0)
Bit 3 T3OS1 (Timer 3 Output Select 1)

When a match occurs between the Timer 3 Up Counter and the Timer 3 Time Constant Register, the following outputs will appear at PC_3 depending on these two bits.

T3OS1	T3OS0	Timer Output	PC_3/$Tout_3$
0	0	Timer Output Inhibited	PC_3 (I/O)
0	1	Toggle Output *1	$Tout_3$ (O)*2
1	0	Output "0"	$Tout_3$ (O)*2
1	1	Output "1"	$Tout_3$ (O)*2

*1 Every time a match occurs between the Timer 3 Up Counter and the Timer 3 Time Constant Register, Timer 3 output level is reversed or toggled. This leads to generation of a square waveform with 50% duty without any software support.

*2 When PC_3 is used as an I/O line after it functions as a timer output, the port pin is configured as an output.

Bit 4 Not Used
Bit 5 Not Used

Bit 6 ECM3I (Enable Count Match 3 Interrupt)

If this bit is set, a Timer 3 interrupt is enabled whenever CMF3 is set. If cleared, the interrupt is inhibited.

ECM3I = "0" : the count match 3 interrupt is inhibited.
ECM3I = "1" : the count match 3 interrupt is enabled.

Bit 7 CMF3 (Count Match Flag 3)

This read-only bit is set when a match occurs between the Timer 3 Up Counter and the Timer 3 Time Constant Register.

CMF3 is cleared by writing a 0 to this bit when CMF3 is set. (Software cannot write a 1 to this bit.)

Table 11 Timer 1, Timer 2, Timer 3 and SCI Status Flag Summary with ICF Set

Status register	Flag	Set Condition	Clear condition
Timer 1 (TCSR1)	ICF	FRC → ICR on a positive or negative edge of the input signal to PC_7 (Tin_1)	1. Read the TCSR1 and then read the ICR(L). 2. $\overline{RES}$ = 0
	OCF	OCR = FRC	1. Read the TCSR1 with OCF set and then read the FRC(L). 2. $\overline{RES}$ = 0
	TOF	FRC = $FFFF+1 cycle	1. Read the TCSR1 with TOF set and then read the FRC(L). 2. $\overline{RES}$ = 0
Timer 2 (TCSR2)	CMF2	T2CNT = T2CONR	1. Write a 0 to the CMF2 when this bit is set. 2. $\overline{RES}$ = 0
Timer 3 (TCSR3)	CMF3	T3CNT = T3CONR	1. Write a 0 to the CMF3 when this bit is set. 2. $\overline{RES}$ = 0
SCI TRCSR1 TRCSR2	RDRF	Receive Shift Register → RDR	1. Read the TRCSR1 or TRCSR2 with RDRF set and then read the RDR. 2. $\overline{RES}$ = 0
	ORFE	1. Stop Bit = 0: Framing Error (in asynchronous mode) 2. Receive Shift Register → RDR when the RDRF is set: Overrun Error. (in asynchronous mode)	1. Read the TRCSR1 or TRCSR2 with ORFE set and then read the RDR. 2. $\overline{RES}$ = 0
	TDRE	1. TDR → Transmit Shift Register (in asynchronous mode) 2. Transmit Shift Register is "empty". (in synchronous mode) 3. $\overline{RES}$ = 0	Read the TRCSR1 or TRCSR2 with TDRE set and then write the TDR. (NOTE) Clear TDRE after setting TE.
	PER	Parity error occurs when the PEN is set.	1. Read the TRCSR2 with PER set and then read RDR. 2. $\overline{RES}$ = 0

(NOTE) →: Transfer
= : match
ICR (L); Input Capture Register low-order byte
OCR(L); Output Compare Register low-order byte
FRC(L); Free Running Counter low-order byte
RDR ; Receive Data Register
TDR ; Transmit Data Register

■ PARALLEL COMMUNICATION INTERFACE (PCI)

The HD63705Z0 has a slave MCU mode which allows the device to easily provide parallel communication with the master MCU. In this mode the HD63705Z0 can be used as an intelligent peripheral controller of microcomputer systems.

Table 12 provides a summary of the functional pins in the slave MCU mode.

Table 12 Slave MCU Mode Functional Pin Summary

Pin	I/O	Function
DBB_0 ⟆ DBB_7	I/O	These pins compose an 8-bit data bus which is used for parallel communication between the master MCU and the HD63705Z0
$\overline{CS}$	I	Chip select input. Logic 0 on this pin enables the data bus.
RS_0 ⟆ RS_3	I	Register select inputs. The logic levels on these pins select one of PCI Control and Status Register and PCI Data Registers 0 to 14. Connect RS_0, RS_1, RS_2 and RS_3 respectively to A_0, A_1, A_2 and A_3 of the master MCU. For details, see Table 14.
$\overline{OE}$	I	Output enable input.
$\overline{WE}$	I	Write enable input.
$\overline{RDY}$	*O	This pin is used to interrupt the master MCU.

*NMOS open drain output

Besides above pins the following registers are used in the slave MCU mode.

- Parallel Communication Control and Status Register (8 bit)
- Parallel Communication Data Registers (15 byte)

• Parallel Communication Control and Status Register (RCCSR; $002B)

This is an 8-bit register. A_{11} bits in this register are readable by both the master MCU and the HD63705Z0 MCU. The master MCU can write bit 4, while the HD63705Z0 MCU can write bit 2, bit 4 and bit 6.

Each of the eight bits is explained below.

	7	6	5	4	3	2	1	0	
	MWEF	EMWI	SWEF	EAKAR	MREF	EMRI	MWMF	SWMF	(PCCSR; $002B) Cleared at reset.
HD63705Z0 Master MCU	R	R	R	R	R	R	R	R	
HD63705Z0 Master MCU		W		W		W			
				W					

Bit 0 SWMF (Slave Write Mode Flag)

This bit is set when the HD63705Z0 MCU writes the Parallel Communication Data Register 0 (PCDR0). While this bit is set, PCDR1 to PCDR14 can be written only by the HD63705Z0. After the write of the PCDR0, the HD63705Z0 reads the PCCSR to ensure that SWEF = 1 and then writes PCDR1 through PCDR14. This bit is cleared when the master MCU reads the PCDR0.

Bit 1 MWMF (Master Write Mode Flag)

This bit is set when the master MCU writes the PCDR0. While this bit is set, only the master MCU can write PCDR1 through PCDR14. After the write of PCDR0, the master MCU reads the PCCSR to ensure that MWEF = 1 and then writes PCDR1 through PCDR14. This bit is cleared when HD63705Z0 MCU reads the PCDR0.

Bit 2 EMRI (Enable Master Read Interrupt)

If this bit is set, a PCI interrupt occurs when the master MCU reads the PCDR0.

EMRI = 0 disables master read interrupt.
EMRI = 1 enables master read interrupt.

Bit 3 MREF (Master Read End Flag)

This bit is set when the master MCU reads the PCDR0.

It is cleared when the HD63705Z0 writes or reads PCDR0, and when master MCU writes PCDR0.

If this bit is set, a PCI interrupt occurs provided EMRI is set.

Bit 4 EAKAR Enable Acknowledge and Request

If this bit is set, the HD63705Z0 interrupts the master MCU. ($\overline{RDY}$ goes low.)

- HD63705Z0 ..... Receiver

When the HD63705Z0 reads the PCDR0, it leaves the master MCU write mode and then interrupts the master MCU.

When the master MCU writes PCDR0 or reads PCCSR, $\overline{RDY}$ pin goes high.

- HD63705Z0 . . . . . Transmitter

When the HD63705Z0 writes the PCDR14, it interrupts the master MCU.

When the master MCU reads the PCDR14 or the PCCSR, the $\overline{RDY}$ pin goes high.

EAKAR = 0 disables $\overline{RDY}$ output.

EAKAR = 1 enables $\overline{RDY}$ output.

Bit 5 SWEF Slave Write End Flag

This bit is set when the HD63705Z0 writes the PCDR14. It is cleared when the master MCU reads the PCDR14.

Bit 6 EMWI Enable Master Write Interrupt

If this bit is set, a PCI interrupt occurs when the master MCU writes the PCDR14.

EMWI = 0 masks master write interrupt.

EMWI = 1 enables master write interrupt.

Bit 7 MWEF Master Write End Flag

This bit is set when the master MCU writes the PCDR14. It is cleared when the HD63705Z0 reads the PCDR14.

Each flag in the Parallel Communication Control and Status Register is set or cleared as listed in Table 13.

Table 13 PCI Control and Status Flag Summary

Bit	Name	Set by	Cleared by	Write
0	SWMF	$\overline{NWMF}$·SW0	MR0 (RESET)	
1	MWMF	$\overline{SWMF}$·MW0	SR0 (RESET)	
2	EMRI	———	——— (RESET)	Writable by slave.
3	MREF	MR0	SW0, SR0, MW0 (RESET)	
4	EAKAR	———	——— (RESET)	Writable by both slave and master.
5	SWEF	SW14	MR14 (RESET)	
6	EMWI	———	——— (RESET)	Writable by slave.
7	MWEF	MW14	MWMF·SR14 (RESET)	

SR0 : Slave Read PCDR 0
SR14 : Slave Read PCDR 14
SW0 : Slave Write PCDR 0
SW14 : Slave Write PCDR 14

MR0 : Master Read PCDR 0
MR14 : Master Read PCDR 14
MW0 : Master Write PCDR 0
MW14 : Master Write PCDR 14

- **Parallel Communication Data Register 0 (PCDR0; $002C)**

The Parallel Communication Data Register 0 (PCDR0) is an 8-bit data register which is composed of double buffers.

Even if the master MCU and the HD63705Z0 writes this register simultaneously, the data is protected.

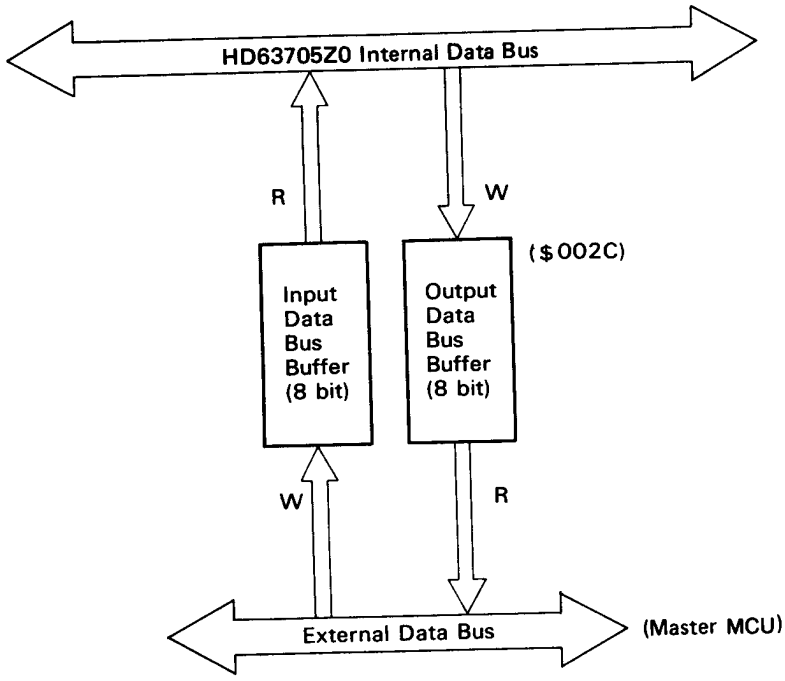


Fig. 45 PCDR0 Block Diagram

This register is used only in the slave mode, so it cannot be used as a general purpose register.

In the mode 6, PCDR0 is used as a slave mode only register.

In the modes 1, 2 and 5, PCDR0 is not used.

- **Parallel Communication Data Registers 1 to 14 (PCDR1 to PCDR 14: $002D to $003A)**

The Parallel Communication Data Registers 1 to 14 (PCDR1 to PCDR14) are 8-bit registers which can be accessed by both the master MCU and the HD63705Z0. The write of this register is controlled by the PCCSR.

In all modes except slave modes (Modes 1, 2 and 5), this register can be used as a general purpose register (RAM).

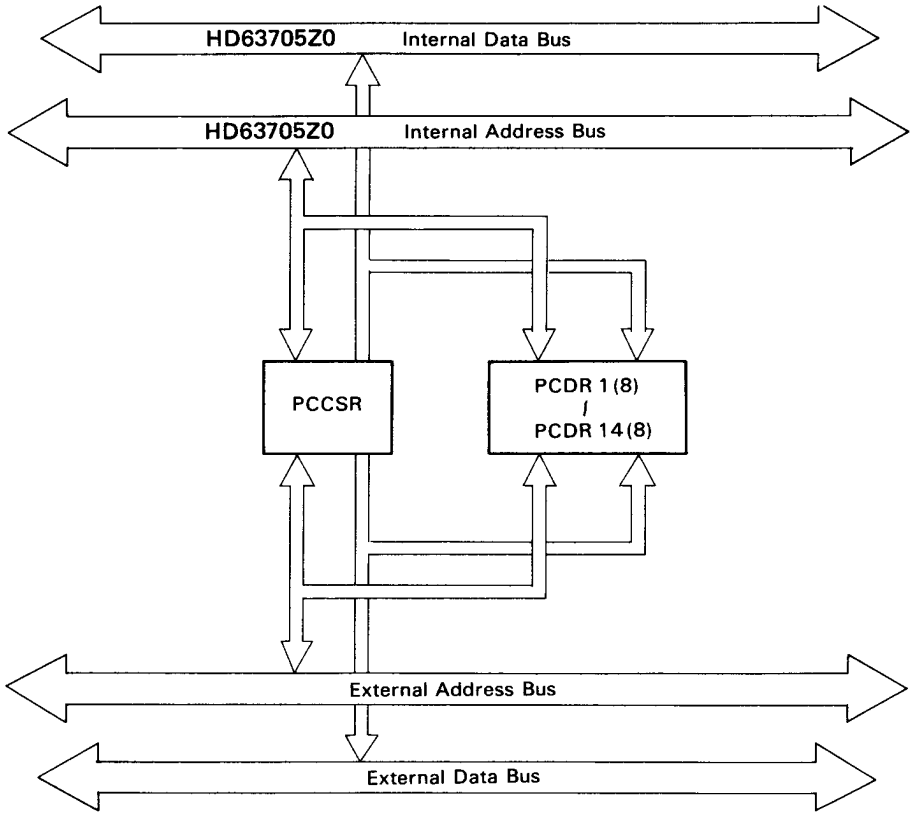


Fig. 46 PCDR1 to PCDR14 Block Diagram

External address bus is connected to the register select pins (RS_0 to RS_3) depending on the memory map of the master MCU.

A parallel communication register is selected depending on the register select inputs as shown below.

Table 14 PCI Register Selection

Register Select Inputs				PCI Register
RS_3	RS_2	RS_1	RS_0	
0	0	0	0	PCCSR
0	0	0	1	PCDR 0
0	0	1	0	PCDR 1
0	0	1	1	PCDR 2
0	1	0	0	PCDR 3
0	1	0	1	PCDR 4
0	1	1	0	PCDR 5
0	1	1	1	PCDR 6
1	0	0	0	PCDR 7
1	0	0	1	PCDR 8
1	0	1	0	PCDR 9
1	0	1	1	PCDR 10
1	1	0	0	PCDR 11
1	1	0	1	PCDR 12
1	1	1	0	PCDR 13
1	1	1	1	PCDR 14

The timing diagrams in the slave MCU mode are shown in Fig. 47.

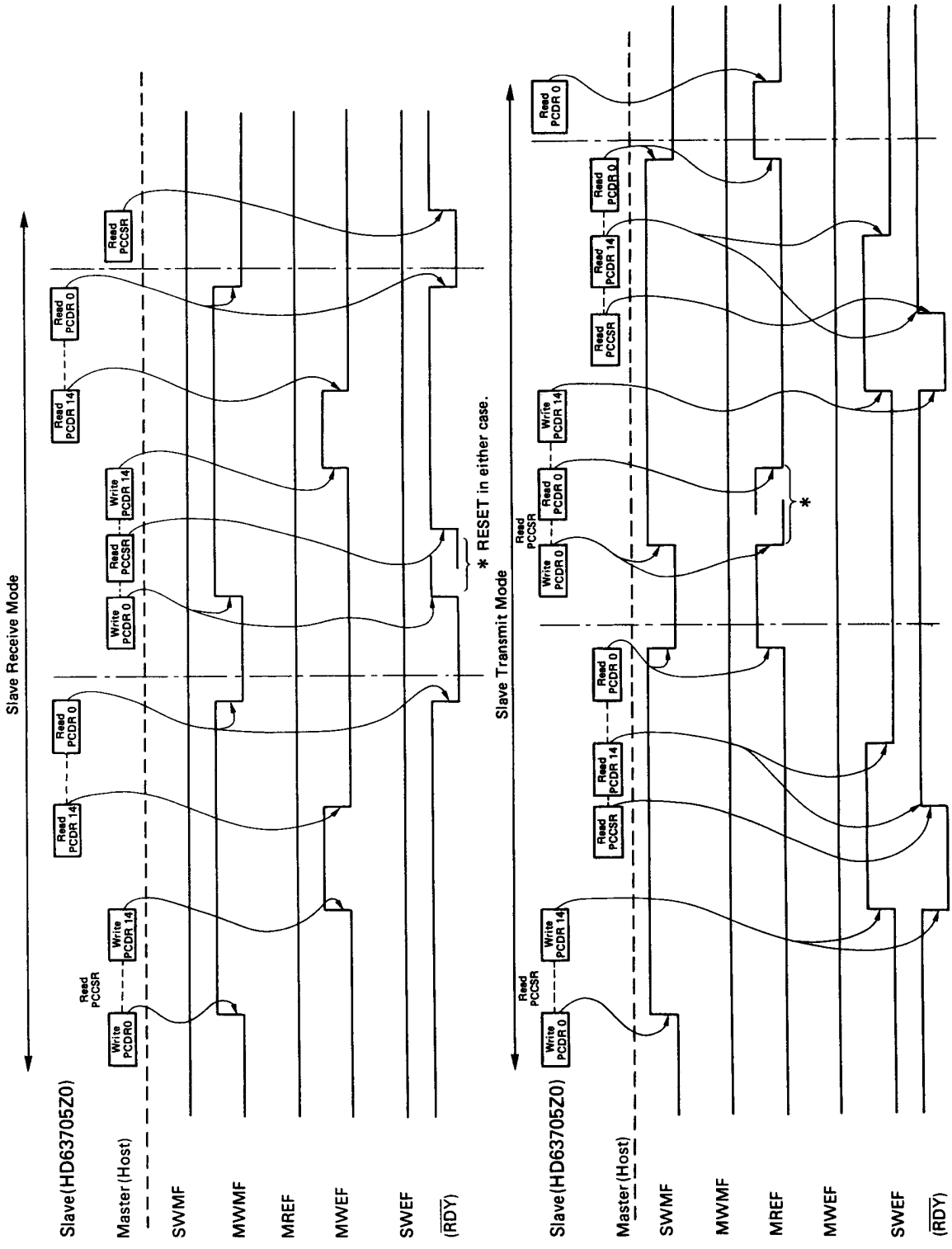


Fig. 47 PCI Transmit/Receive Timing

Examples of the connection between the HD63705Z0 and the master MCU are shown in Fig. 48.

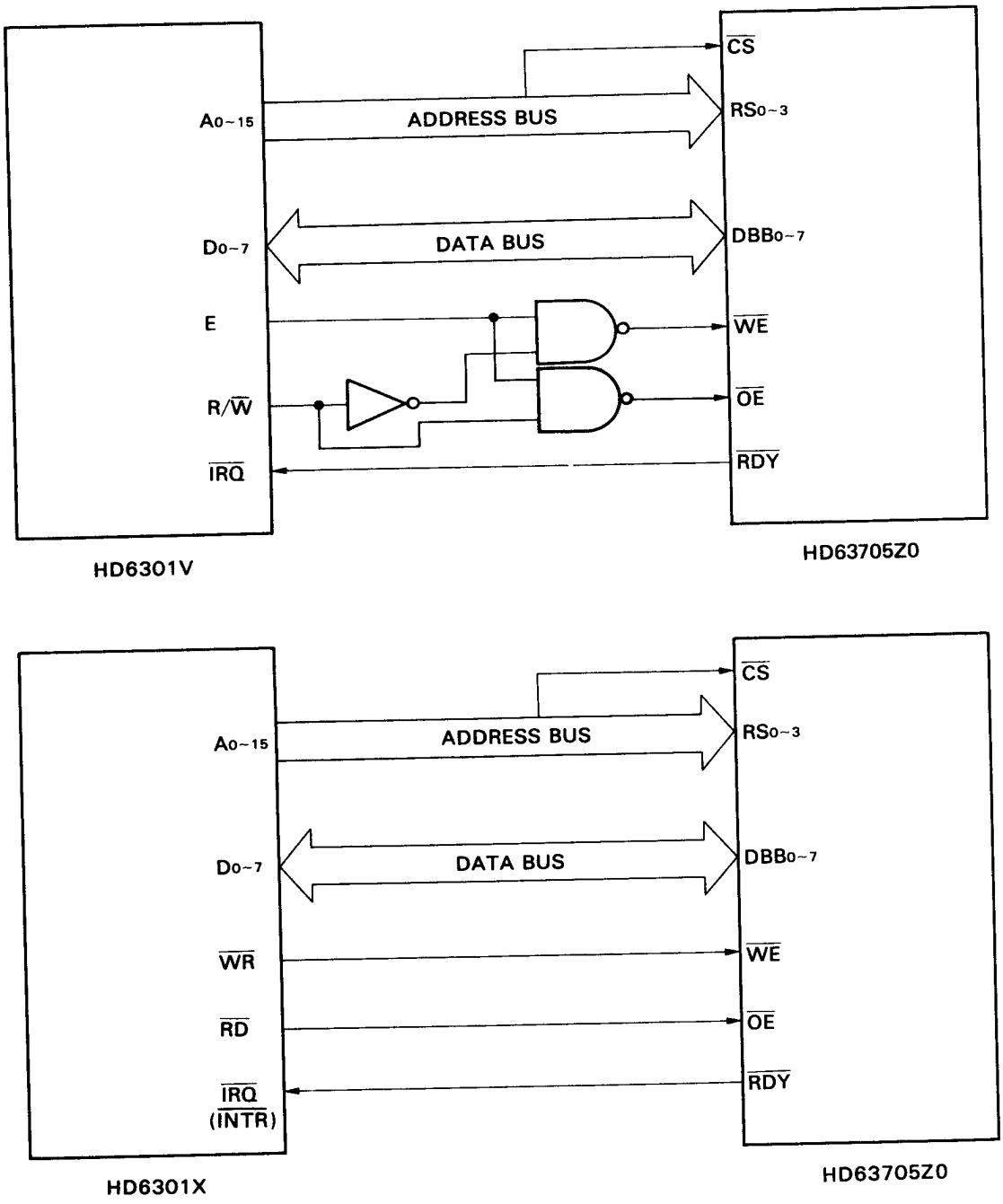


Fig. 48 Examples of Master MCU Connection

■ SERIAL COMMUNICATIONS INTERFACE (SCI)

The HD63705Z0 contains a serial communications interface (SCI) which provides two operating modes: asynchronous mode with a NRZ format and synchronous mode in which data transfer conforms a clock signal.

In asynchronous mode, data length, parity bit and the number of stop bits are selectable and eight types of data formats are available.

The SCI consists of the following registers:

- Transmit/Receive Control and Status Register 1 (TRCSR1)
- Rate and Mode Control Register (RMCR)
- Transmit/Receive Control and Status Register 2 (TRCSR2)
- Receive Data Register (RDR)
- Receive Shift Register
- Transmit Data Register (TDR)
- Transmit Shift Register

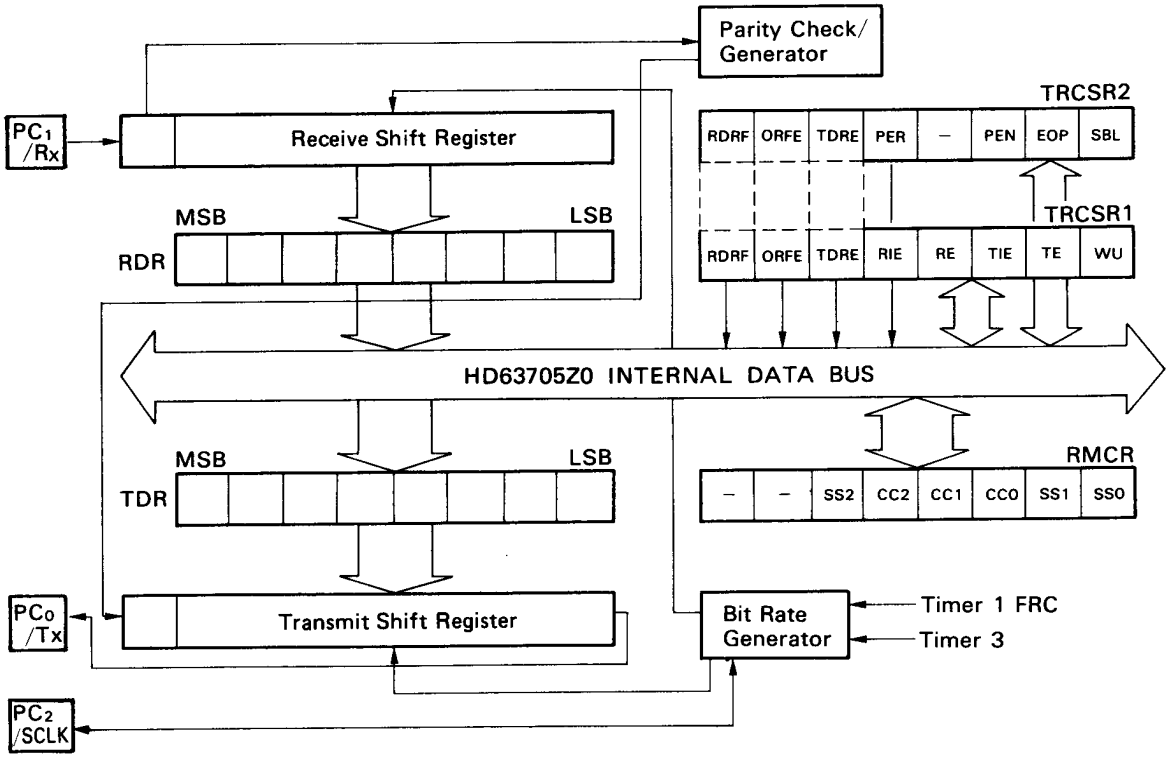


Fig. 49 SCI Block Diagram

● Transmit/Receive Control and Status Register 1 (TRCSR1; $0027)

The TRCSR1 is an 8-bit register. All eight bits can be read but only the five low order bits can be written. During reset this register is initialized to $20. Individual bits of the TRCSR1 are defined below.

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	TRCSR1 : $0027
R	R	R	R/W	R/W	R/W	R/W	R/W	
(0	0	1	0	0	0	0	0)	At Reset

Bit 0 WU (Wake-Up)

In a typical multiprocessor application, the message contains a destination address in its initial bytes. In order to allow non-interested MCUs to ignore the remainder of the message, a wake-up feature is provided. This feature inhibits all further receive processing until the beginning of the next message.

The wake-up is automatically triggered by an idle string of ten (for 8-bit data) or eleven (for 9-bit data) consecutive 1's. The software protocol must provide for the idle period between two consecutive messages. When WU bit is set, the MCU suspends the receive processing until the next message appears. When the MCU wakes up by receipt of ten or eleven consecutive 1's, it clears WU by hardware and resumes normal receive operation. However, it is necessary to set the receive enable bit (RE) before setting WU bit. In synchronous mode WU bit must not be set because the wake-up feature is not available.

Bit 1 TE (Transmit Enable)

When this bit is set in asynchronous mode, the Transmit Shift Register output is applied to Port C bit 0 after a preamble

of ten or eleven consecutive 1's is produced. In synchronous mode, as soon as TE bit is set, the Transmit Shift Register output is applied to Port C bit 0. Both of the above cases are independent of the Data Direction Register value for Port C bit 0. While TE is cleared, the SCI does not affect Port C bit 0.

Bit 2 TIE (Transmit Interrupt Enable)

If this bit is set, a SCI interrupt occurs provided TDRE (bit 5) is set. If cleared, the interrupt is inhibited.

Bit 3 RE (Receive Enable)

If RE bit is set, signals are input to the Receive Shift Register from Port C bit 1 regardless of the Data Direction Register value for the port bit 1 (PC_1). While RE is cleared, the SCI does not affect Port C bit 1.

Bit 4 RIE (Receive Interrupt Enable)

If this bit is set, a SCI interrupt occurs provided RDRF (bit 7) or ORFE (Bit 6) is set. If cleared, the interrupt is inhibited.

Bit 5 TDRE (Transmit Data Register Empty)

This bit is set by hardware to indicate that the contents of the Transmit Data Register have been transferred to the Transmit Shift Register in the asynchronous mode or that the Transmit Data Shift register is empty in the synchronous mode. This bit is cleared by reading the TRCSR1 or the TRCSR2 (with TDRE set), followed by writing a new byte to the Transmit Data Register. Reset sets the TDRE bit.

(NOTE) TDRE should be cleared in the transmission state after the TE set.

Bit 6 ORFE (Overrun/Framing Error)

This bit is set by hardware when an overrun error or a framing error occurs during receive operation. The overrun error occurs when a byte is ready to be transferred to the Receive Data Register with RDRF bit set. The framing error occurs when stop bit is a 0, but the MCU does not detect this error in the synchronous mode. The ORFE bit is cleared by reading TRCSR1 or TRCSR2 (with ORFE set), followed by reading the Receive Data Register. It is also cleared by reset.

Bit 7 RDRF (Receive Data Register Full)

This bit is set by hardware when a byte is normally received and then is transferred from the Receive Shift Register to the Receive Data Register. The RDRF bit is cleared by reading TRCSR1 or TRCSR2 (with RDRF set), followed by reading the Receive Data Register. It is also cleared by reset.

(NOTE) When some of bits 5 to 7 are set, they can be cleared by a previous read of TRCSR1 or TRCSR2. It is unnecessary to read a TRCSR prior to every clear of the each bit.

Rate and Mode Control Register (RMCR; $0026)

This register controls the following SCI variables:

- Baud rate
- Operation mode (asynchronous mode or synchronous mode)
- Port C bit 2 Configuration (SCLK)
- Clock source
- Data format

All six bits of this register can be read and written. During reset these bits are cleared

7	6	5	4	3	2	1	0	
—	—	SS2	CC2	CC1	CC0	SS1	SS0	RMCR : $0026
		R/W	R/W	R/W	R/W	R/W	R/W	
(—	—	0	0	0	0	0	0)	At Reset

Bit 0 SS0
Bit 1 SS1
Bit 5 SS2
} **Speed Select**

These three bits select the baud rate which is used with the SCI. Table 15 lists the baud rates available. The source of the internal clock used with SCI is provided from the Timer 1 Free Running Counter (with SS2 cleared) or the Timer 3 Up Counter (with SS2 set). If the Timer 3 is used as the source, optional baud rates can be obtained depending on the state of bits in the Timer 3 Time Constant Register as shown in Table 16.

(NOTE) If an internal clock is used with the SCI, MCU writes to the timer counter used as the clock source should be avoided.

Bit 2 CC0
Bit 3 CC1
Bit 4 CC2
} **Clock Control/Format Select**

These three bits control the data format and the clock source. (See Table 17.)

Reset clears CC0, CC1 and CC2 and causes the SCI to synchronize with an external clock. Port C bit 2 is forced to an input line to accept the external clock. When using Port C bit 2 as an output port, CC1 and CC0 should be set to 0, 1 and DDR for Port C bit 2 to 1.

- **Transmit/Receive Control and Status Register 2 (TRCSR2; $0028)**

This is a 7-bit register which is used to determine the data format in the asynchronous mode. The three leftmost bits are dual-addressed with the TRCSR1, which allows the states of RDRF, ORFE and TDRE to be read by accessing either TRCSR1 or TRCSR2. Bits 0 to 2 of this register can be read and written. Bits 4 to 7 are read-only bits.

Reset clears bits 0 to 2, bit 4, bit 6 and bit 7, and sets bit 5.

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	PER	—	PEN	EOP	SBL	TRCSR2 : $0028
R	R	R	R		R/W	R/W	R/W	
(0	0	1	0	—	0	0	0)	At Reset

Bit 0 SBL (Stop Bit Length)

This bit determines the number of stop bits in the asynchronous mode. If this bit is clear, a stop bit is used. If set, two stop bits are used. The SBL bit is cleared by reset.

Bit 1 EOP (Even/Odd Parity)

When the PEN is set, this bit selects a parity which will be generated and checked. If EOP is cleared, an even parity is selected and if set an odd parity is selected. Reset clears the EOP bit.

Bit 2 PEN (Parity Enable)

If PEN is cleared in asynchronous mode, no parity bit is generated or checked. If PEN is set, a parity bit is generated or checked. Reset clears the PEN bit.

In synchronous mode, the above three bits don't affect the SCI operation.

Bit 3 Not Used

Bit 4 PER (Parity Error)

This bit is set when a parity error occurs while the PEN bit is set. The PER bit is cleared by reading the TRCSR2 (with PER set), followed by reading the Receive Data Register.

Bit 5 TDRE (Transmit Data Register Empty)

Bit 6 ORFE (Overrun/Framing Error)

Bit 7 RDRF (Receive Data Register Full)

The states of TDRE, ORFE and RDRF can be read by accessing either TRCSR1 or TRCSR2.

Table 15 SCI Bit Times and Rates

(1) Asynchronous Mode

			XTAL	2.4576MHz	4.0MHz	4.9152MHz
SS2	SS1	SS0	E	614.4kHz	1.0MHz	1.2288MHz
0	0	0	E÷16	26μs/38400Baud	16μs/62500Baud	13μs/76800Baud
0	0	1	E÷128	208μs/4800Baud	128μs/7812.5Baud	104.2μs/9600Baud
0	1	0	E÷1024	1.67ms/600Baud	1.024ms/976.6Baud	833.3μs/1200Baud
0	1	1	E÷4096	6.67ms/150Baud	4.096ms/244.1Baud	3.333ms/300Baud
1	—	—	—	*	*	*

*When SS2 is set, Timer 3 provides clock for SCI. The baud rate can be calculated as follows. (Variable N is the value of Timer 3 Time Constant Register.)

$$\text{Baud rate (Baud)} = \frac{f}{32\,(N+1)}$$

f : Frequency of clock input to Timer 3 counter

N : 0 through 65535

(2) Synchronous Mode*

			XTAL	4.0MHz	6.0MHz	8.0MHz
SS2	SS1	SS0	E	1.0MHz	1.5MHz	2.0MHz
0	0	0	E÷2	2μs/bit	1.33μs/bit	1μs/bit
0	0	1	E÷16	16μs/bit	10.7μs/bit	8μs/bit
0	1	0	E÷128	128μs/bit	85.3μs/bit	64μs/bit
0	1	1	E÷512	512μs/bit	341μs/bit	256μs/bit
1	—	—	—	**	**	**

*These are bit rates in case of using an internal clock. An external clock can be used to provide frequencies of DC to 1/2 system clock.

**The bit rate can be calculated as follows. (Variable N is the value of Timer 3 Time Constant Register.)

$$\text{Bit rate (μs/bit)} = \frac{4\,(N+1)}{f}$$

f : Frequency of clock input to Timer 3 counter

N : 0 through 65535

Table 16 Example of Baud Rate and Timer 3 Time Constant Register

Baud Rate \ XTAL	2.4576MHz	3.6864MHz	4.0MHz	4.9152MHz	8.0MHz
110	174	261	283	348	567
150	127	191	207	255	416
300	63	95	103	127	207
600	31	47	51	63	103
1200	15	23	25	31	51
2400	7	11	12	15	25
4800	3	5	—	7	12
9600	1	2	—	3	—
19200	0	—	—	1	—
38400	—	—	—	0	—

Table 17 SCI Format and Clock Source Control

CC2	CC1	CC0	Format	Mode	Clock source	Port C Bit 2	Port C Bit 1 Port C Bit 0
0	0	0	8 bit data	Synchronous	External	Input	When RE is set, bit 1 is used as an SCI input.
0	0	1	8 bit data	Asynchronous	Internal	Not used**	
0	1	0	8 bit data	Asynchronous	Internal	Output*	
0	1	1	8 bit data	Asynchronous	External	Input	When TE is set, bit 0 is used as an SCI output.
1	0	0	8 bit data	Synchronous	Internal	Output	
1	0	1	7 bit data	Asynchronous	Internal	Not Used**	
1	1	0	7 bit data	Asynchronous	Internal	Output*	
1	1	1	7 bit data	Asynchronous	External	Input	

* Clock is output regardless of the states of RE and TE in TRCSR.
** Not used with SCI.

SCI Operation

The SCI must be initialized prior to operation by the following sequence depending on the operation mode (asynchronous mode or synchronous mode) and data format.

(1) Writes the desired operational control bits to the Rate and Mode Control Register.
(2) Writes the desired operational control bits to the TRCSR2.
(3) Set an enable bit (TE or RE) in the TRCSR1.

The enable bits (TE and RE) must be cleared before the operation mode and data format are changed. Then they must be set again at least a bit time (at baud rate or bit rate) after the change. If TE or RE is set within a bit time, the SCI may not be initialized.

Transmit operation in asynchronous mode

A data format is selected depending on the Rate and Mode Control Register and TRCSR2. (See Fig. 50). Then transmit operation is enabled by TE bit in the TRCSR1. When this bit is set, the port C bit 0 is used as an SCI output (Tx) independently of the corresponding bit in the port C Data Direction Register and transmission begins with a preamble consisting of 10 or 11 consecutive 1's.

At this point, one of two situations exists:

1. If the Transmit Data Register is empty (TDRE = 1), a continuous string of 1's is transmitted indicating an idle state or
2. If data has been stored into the Transmit Data Register (TDRE = 0), the data is transferred to the Transmit Shift Register and transmission begins.

A "0" start bit, 7 or 8-bit data, an odd or even parity bit if PEN is set, and finally one or two "1" stop bits are transmitted.

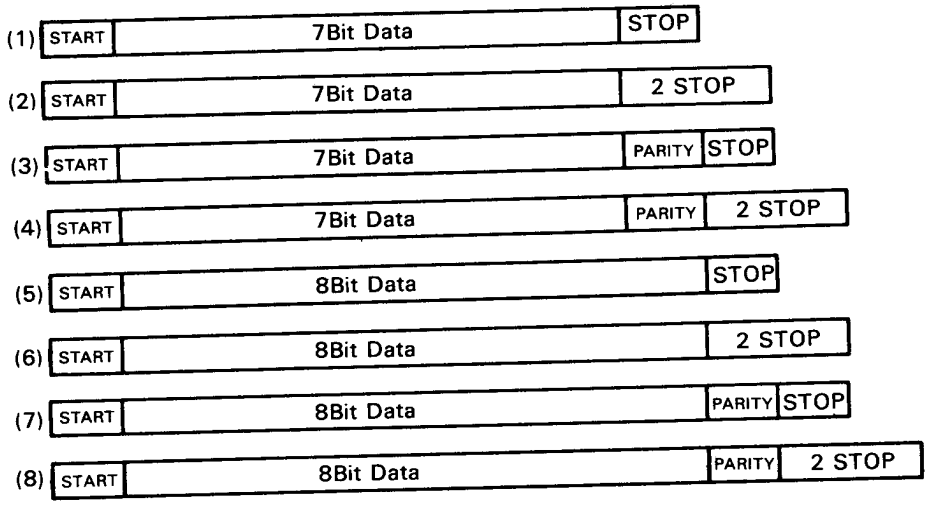


Fig. 50 Asynchronous Data Format

When the Transmit Data Register is emptied, the TDRE flag bit is set by hardware.

If the CPU cannot transmit a data responding to this flag being set (The TDRE remains set until the next byte is transferred from the Transmit Data Register to the Transmit Shift Register), "1" is transmitted instead of "0" start bit, and then 1's are transmitted until more data is provided to the Transmit Data Register. As long as the TDRE is set, "0" is not transmitted.

Receive operation in asynchronous mode

First, a desired data format is selected depending on the Rate and Mode Control Register and TRCSR2 and secondly receive operation is enabled by the RE bit in the TRCSR1. If this bit is set, the Port C bit 1 will be forced to an SCI input (Rx). The received bit stream is synchronized with the leading edge of the first 0 (space) encountered. The SCI input is then sampled at the approximate center of each bit time interval for ten or eleven consecutive bits. If the stop bit is not a 1, a framing error is assumed and ORFE is set. The data with the framing error is transferred to the Receive Data Register so the CPU can read the data which has caused the error. This allows a line break to be detected.

If the PEN bit is set, the parity check is performed. If a match does not occur between a parity bit and the EOP bit, a parity error is assumed and the PER bit is set instead of the RDRF bit. The data received with a parity error can be read in the same way as the data with framing error.

The RDRF flag is set when data is received without a framing error nor a parity error.

If the RDRF is still set when the STOP bit of the next data is received, ORFE will be set indicating that an overrun has occurred. The data received can be read from the Receive Data Register. If the 7 bit data format is selected, the most significant bit of the Receive Data Register is a 0.

(NOTE) Clock source in asynchronous mode

If an internal clock is used with the SCI, the following requirements should be satisfied.

- The CC1: CC0 bit in the Rate and Mode Control Register must be set to 10. (See Table 17.)
- Clock is generated independently of the states of the TE and the RE.
- The maximum clock frequency is E÷16.
- The clock frequency is internal bit rate x 1.

If an external clock is used with the SCI, the following requirements must be satisfied.

- The CC1: CC0 in the Rate and Mode Control Register must be set to 11. (See Table 17.)
- The maximum clock frequency must equal that of the system clock.
- The clock frequency must be sixteen times (16 x) desired baud rate.

Transmit operation in synchronous mode

In synchronous mode, data is transferred synchronizing with a clock.

The operation mode and the clock source must be determined by setting corresponding bits in the RMCR and TRCSR1. Then transmit operation is enabled by the TE bit in the TRCSR1. If the bit is set, Port C bit 0 is forced to an output (Tx) independently of the corresponding bit in the Port C Data Direction Register.

If an external clock is selected, data is transmitted through the Port C Bit 0 (Tx) synchronously with eight external clock pulses which are input to the Port C Bit 2 (SCLK) with TDRE cleared. TDRE flag is set when the Transmit Data Register becomes empty after transmission. If more than eight clock pulses are input, the MCU ignores the surplus clock pulses.

If an internal clock is selected, the MCU outputs a data and the clock when TDRE is cleared.

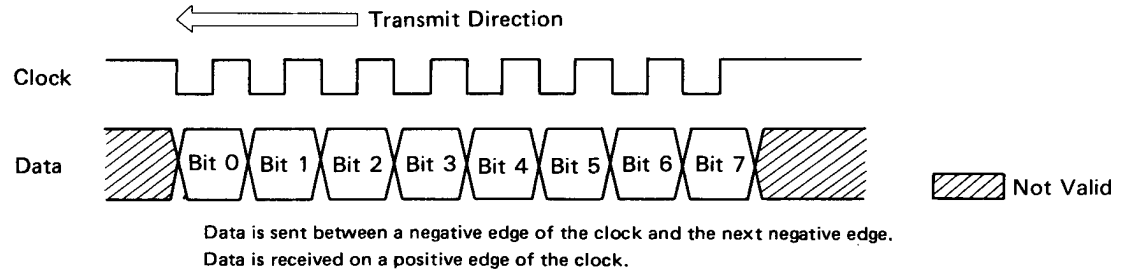


Fig. 51 Synchronous Data Transmission

Receive operation in synchronous mode

The operation mode and the clock source must be determined by the RMCR and TRCSR1. Then data reception is enabled by the RE bit in the TRCSR1. If this bit is set, the Port C Bit 1 (Rx) is forced to an SCI input.

If an external clock is selected, eight clock pulses are input to the Port C Bit 2 (SCLK) and a byte is input to the Port C Bit 1 (Rx) synchronously with the clock. The MCU transfers the byte to the Receive Shift Register and then to the Receive Data Register synchronously with the clock, finally sets RDRF bit. If more than eight external clock pulses are input, the MCU ignores the surplus clock pulses. RDRF must be cleared when the Port C Bit 2 (SCLK) is in a high state because the MCU starts the next byte reception as soon as the RDRF is cleared.

If an internal clock is selected, eight clock pulses are output when RE bit is set. A byte should be input synchronizing with this clock pulses. When the first byte is received, the RDRF bit is set. Then clearing the RDRF causes the clock to be output and initiates the next byte reception. Transmission and reception can be simultaneously performed by sharing the same clock.

■ A/D CONVERTER

The HD63705Z0 has an analog-to-digital converter implemented on the chip using a successive approximation technique with ladder resistance. It provides 8-bit resolution and eight analog inputs.

The A/D converter has the following registers.

- A/D Control and Status Register (8-bit)
- A/D Result Register (8-bit)

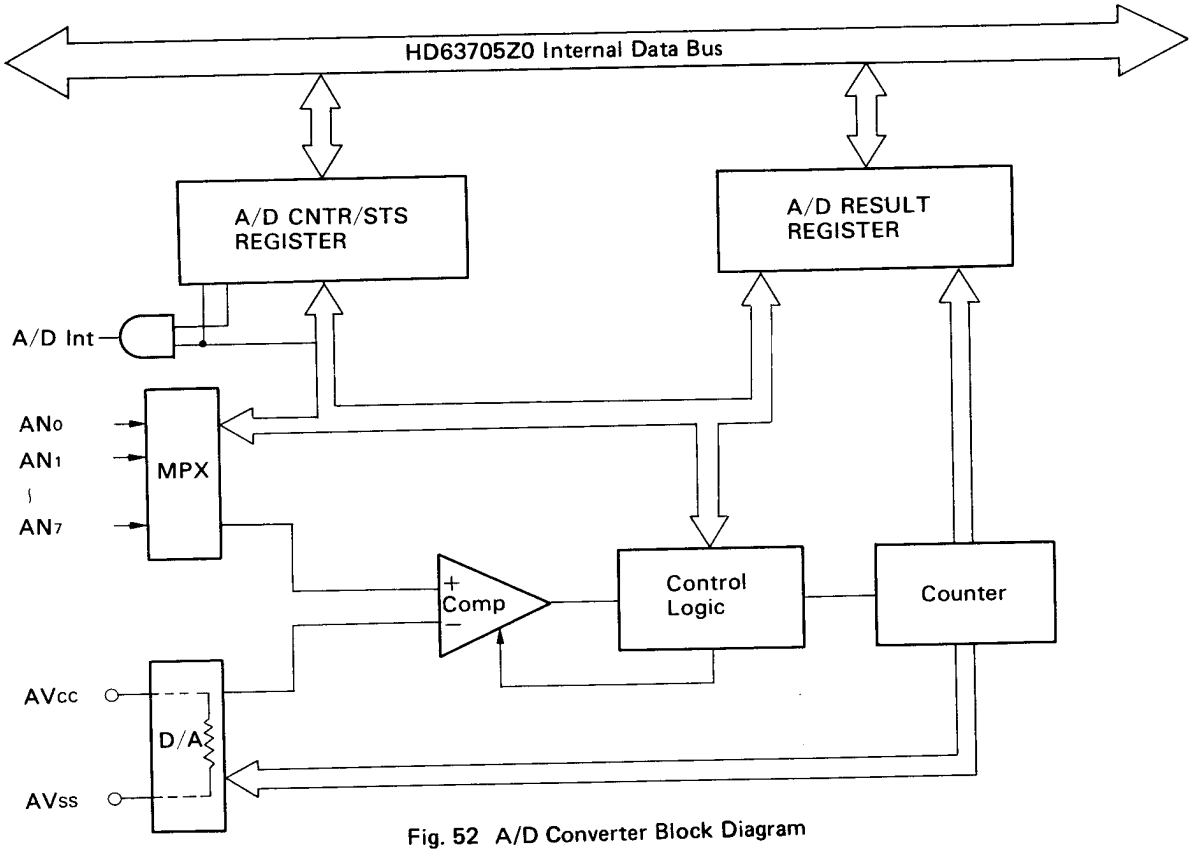


Fig. 52 A/D Converter Block Diagram

• A/D Control and Status Register (ADCSR; $0010)

The A/D Control and Status Register (ADCSR) provides the bits which select the A/D conversion time and the analog input pin and verify the start and end of A/D conversion. The ADCSR is cleared by reset.

The individual bits are discussed below.

7	6	5	4	3	2	1	0
ADEF	ADS	EADEI	CH2	CH1	CH0	ACS1	ACS0
R	W	R/W	R/W	R/W	R/W	R/W	R/W

ADCSR : $0010
Reset ($00)

Bit 0 ACS0 (A/D Clock Select 0)
Bit 1 ACS1 (A/D Clock Select 1)

These two bits are used to select the A/D conversion time at a given system clock frequency. The minimal A/D conversion time is 34 μs.

ACS1	ACS0	f=1.0MHz	f=1.5MHz	f=2.0MHz
0	0	34μs	———	———
0	1	68μs	45.3μs	34μs
1	0	136μs	90.6μs	68μs

Bit 2 CH0 (Analog Channel Select 0)
Bit 3 CH1 (Analog Channel Select 1)
Bit 4 CH2 (Analog Channel Select 2)

These three bits are used to select the analog channel input.

CH2	CH1	CH0	Channel
0	0	0	AN0
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

Bit 5 EADEI (Enable A/D End Interrupt)

If EADEI bit is set, an A/D interrupt occurs at completion of A/D conversion. If cleared, the interrupt is masked.

EADEI = 0 A/D End Interrupt Masked
EADEI = 1 A/D End Interrupt Enabled

Bit 6 ADS (A/D Start)

The ADS bit is a write-only bit. Writing a 1 to this bit starts an A/D conversion. When additional 1 is written during the A/D conversion (ADS = 1), the conversion is restarted.

After completion of an A/D conversion, the data converted is transferred into the A/D Result Register and then the ADS bit is cleared.

Bit 7 ADEF (A/D End Flag)

This read-only bit is set after completion of an A/D conversion, that is, when data is transferred to the A/D Result Register and then the ADS is cleared. This bit is cleared by reading the A/D Control and Status Register, then the A/D Result Register.

- **A/D Result Register (ADRR; $0011)**

7	6	5	4	3	2	1	0	
ADRR7	ADRR6	ADRR5	ADRR4	ADRR3	ADRR2	ADRR1	ADRR0	ADRR : $0011
R	R	R	R	R	R	R	R	

The A/D Result Register is an 8-bit read-only register. Data may not be read during A/D conversion. A converted data is held until the next conversion is started. Reset does not affect this register.

- **Miscellaneous Register (MR) ($003B)**

This is an 8-bit register which consists of three read-only bits, a write-only bit and four read/write bits.

The individual bits in the miscellaneous register are discussed below.

■ MISCELLANEOUS REGISTER

7	6	5	4	3	2	1	0	
Tin_1 /PC_7	$Tout_1$ /PC_6	PULSE OUT	RAME	STBY PWR	MS_2	MS_1	MS_0	(MR : $003B)
R/W	R/W	W	R/W	R/W	R	R	R	
(0	0	0	1	*	—	—	—)	At Reset

*This bit is cleared by power OFF.

Bit 0 MS0 (Mode Status 0)
Bit 1 MS1 (Mode Status 1)
Bit 2 MS2 (Mode Status 2)

These three bits are read-only bits. These bits are set in response to a positive edge of the $\overline{RES}$ signal to indicate the operating mode.

MS_2	MS_1	MS_0	Operating Mode
0	0	1	Mode 1
0	1	0	Mode 2
1	0	1	Mode 5
1	1	0	Mode 6

Bit 3 STBY PWR (Standby Power bit)

This bit is cleared when V_{CC} is not supplied. This is a read/write flag bit which the user can read by software. If the MCU enters the standby mode with this bit set and this bit is still set when the MCU leaves the mode, it indicates that V_{CC} is supplied and data of RAM is held in the standby mode.

This bit is not cleared during reset.

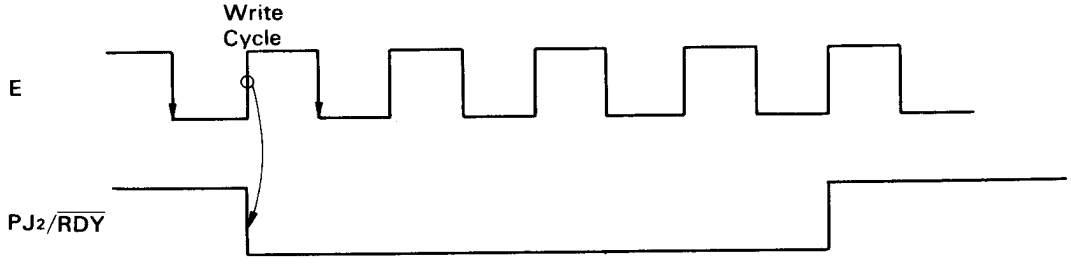
Bit 4 RAME (RAM Enable)

This control bit allows the user to disable the internal RAM. MCU reset sets this bit and enables the internal RAM. If this bit is cleared, the internal RAM becomes invalid and the CPU can read data from the external memory in mode 1 and mode 2. It is necessary to clear this bit before the MCU enters the standby mode for protecting RAM data. When the MCU leaves the mode, this bit is set.

RAME = 0 Internal RAM disabled
RAME = 1 Internal RAM enabled

Bit 5 PULSE OUT (Pulse Out)

This is a write-only bit. When both Data Direction Register bit and Data Register bit for Port J bit 2 are set, 0's are output from $PJ_2/\overline{RDY}$ pin for four cycles by setting PULSE OUT bits.



PULSE Out Timing

Bit 6 $Tout_1/PC_6$

This bit is used to determine the PC_6 ($Tout_1$) pin configuration. If this bit is set, PC_6 ($Tout_1$) is used as timer output 1. If cleared, PC_6 ($Tout_1$) is used as a data I/O line.

This bit is cleared by reset.

$Tout_1/PC_6 = 0$ PC_6 enabled, $Tout_1$ disabled

$Tout_1/PC_6 = 1$ PC_6 disabled, $Tout_1$ enabled

Bit 7 Tin_1/PC_7

This bit is used to determine the PC_7 (Tin_1) pin configuration. If this bit is set, PC_7 (Tin_1) is used as timer input 1. If cleared, PC_7 (Tin_1) is used as a data I/O line.

This bit is cleared by reset.

$Tin_1/PC_7 = 0$ PC_7 enabled, Tin_1 disabled

$Tin_1/PC_7 = 1$ PC_7 disabled, Tin_1 enabled

■ DIRECT PAGE REGISTER

● Direct Page Register (DPR; $003D)

This is an 8-bit read/write register. It is a CPU register but can also be used as an external register which is located in the memory space.

Responding to direct addressing instruction execution, the contents of the Direct Page Register are output to the eight high-order address bus lines. The operand of the instruction appears at the low-order address lines.

The Direct Page Register is cleared by reset. If nothing is written to the Direct Page Register after reset, the instructions in the direct page addressing mode are compatible with 6305U/ V and 6305 X/Y series.

The Direct Page Register can be read/written by loading/ storing the location $003D.

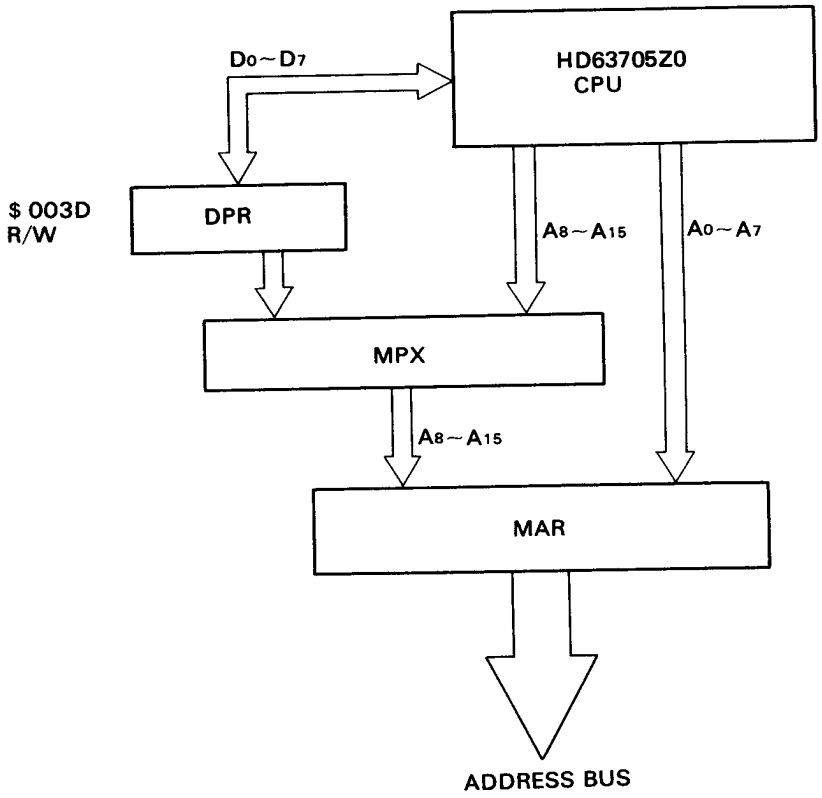


Fig. 53 Direct Page Register Function

■ INTERRUPT CONTROL REGISTER

● Interrupt Control Register (INTCR; $003C)

The Interrupt Control Register is an 8-bit read/write register which is used to control interrupts.

The individual bits in the register are discussed below.

7	6	5	4	3	2	1	0	
$\overline{INT_1}$E	Timer1E	Timer2E	Timer3E	PCIE	SCIE	$\overline{INT_2}$E	ADIE	INTCR: $003C R/W

All these bits are set to $FF during reset.

Bit 0 A/D interrupt enable (ADIE)

This bit controls the A/D interrupt at completion of an A/D conversion. The interrupt is masked when ADIE bit is cleared and allowed when ADIE bit is set.

ADIE = 0 A/D interrupt masked
ADIE = 1 A/D interrupt allowed

Bit 1 $\overline{INT_2}$ interrupt enable ($\overline{INT_2}$E)

This bit controls the $\overline{INT_2}$ interrupt. The interrupt is masked when $\overline{INT_2}$E bit is cleared and allowed when $\overline{INT_2}$E bit is set.

$\overline{INT_2}$E = 0 $\overline{INT_2}$ interrupt masked
$\overline{INT_2}$E = 1 $\overline{INT_2}$ interrupt allowed

Bit 2 SCI interrupt enable (SCIE)

This bit controls the SCI interrupt. The interrupt is masked when SCIE bit is cleared and allowed when SCIE bit is set.

SCIE = 0 SCI interrupt masked
SCIE = 1 SCI interrupt allowed

Bit 3 PCI interrupt enable (PCIE)

This bit controls the PCI interrupt. The interrupt is masked when PCIE bit is cleared and allowed when PCIE bit is set.

PCIE = 0 PCI interrupt masked
PCIE = 1 PCI interrupt allowed

Bit 4 Timer 3 Interrupt Enable (Timer 3E)
Bit 5 Timer 2 Interrupt Enable (Timer 2E)
Bit 6 Timer 1 Interrupt Enable (Timer 1E)

These bits control the timer interrupts.

Timer NE = 0 Timer N interrupt masked
Timer NE = 1 Timer N interrupt allowed
N = 1, 2, 3

Bit 7 $\overline{INT_1}$ Interrupt Enable ($\overline{INT_1}$E)

This bit controls the $\overline{INT_1}$ interrupt. The interrupt is masked if $\overline{INT_1}$E bit is cleared and allowed if $\overline{INT_1}$E bit is set.

$\overline{INT_1}$E = 0 $\overline{INT_1}$ interrupt masked
$\overline{INT_1}$E = 1 $\overline{INT_1}$ interrupt allowed

■ INTERRUPT

The HD63705Z0 MCU may be interrupted by one of eleven different methods: through external interrupt input pins ($\overline{RES}$, $\overline{NMI}$, $\overline{INT_1}$ and $\overline{INT_2}$ ($\overline{STB}$)), hardware interrupt requests (Timer 1, Timer 2, Timer 3, PCI, SCI and A/D) and a software interrupt instruction (SWI). All interrupts except $\overline{RES}$, $\overline{NMI}$ and SWI are maskable. For the $\overline{INT_1}$ interrupt, either a negative sensitive trigger of a V_{IL} level-sensitive trigger is available as an option.

The interrupts are controlled by the Interrupt Control Register (INTCR; $003C) and individual interrupt ($\overline{INT_1}$, $\overline{INT_2}$ ($\overline{STB}$), Timer 1, Timer 2, Timer 3, PCI, SCI or A/D) control registers.

Table 18 Interrupt Summary

Priority	Vector Address		Interrupt	Maskability
	MSB	LSB		
Highest	1FFE	1FFF	$\overline{RES}$	Nonmaskable (L)
	1FFC	1FFD	$\overline{NMI}$	Nonmaskable (N)
	1FFA	1FFB	SWI	Nonmaskable (S)
	1FF8	1FF9	$\overline{INT_1}$	Maskable (N/L)
	1FF6	1FF7	Timer 1	Maskable
	1FF4	1FF5	Timer 2	Maskable
	1FF2	1FF3	Timer 3	Maskable
	1FF0	1FF1	PCI	Maskable
	1FEE	1FEF	SCI	Maskable
	1FEC	1FED	$\overline{INT_2}$ ($\overline{STB}$)	Maskable (N)
Lowest	1FEA	1FEB	A/D	Maskable

N : Negative edge-sensitive interrupt
L : Level-sensitive interrupt
S : Software interrupt

- **Reset ($\overline{RES}$)**

The MCU is reset by applying a logic low to the $\overline{RES}$ input. Reset initializes or holds the internal registers as shown in Table 19.

- **Nonmaskable interrupt ($\overline{NMI}$)**

An $\overline{NMI}$ request is generated by a high-to-low voltage transition on the $\overline{NMI}$ pin, but the interrupt is pending until the current instruction execution is completed. The interrupt mask bit in the Condition Code Register does not affect the $\overline{NMI}$. When the current instruction is completed, processing is suspended, the current contents of the Program Counter, Index Register, Accumulator, Condition Code Register and Direct Page Register are pushed onto the stack, the address of the interrupt routine is obtained from the appropriate interrupt vector address ($1FFC and $1FFD), transferred to the Program Counter and the interrupt routine is executed.

The $\overline{NMI}$ cannot be masked. Upon completion of the interrupt service routine, the RTI instruction causes the register contents to be recovered from the stack, then allows the MCU to resume processing of the program.

- **$\overline{INT_1}$**

This interrupt is produced by an input to the $\overline{INT_1}$ pin. The $\overline{INT_1}$ interrupt request can be cleared by writing a 0 to the bit 6 of the Port A Control and Status Register ($000F). Either a V_{IL} level-sensitive trigger or a negative edge-sensitive trigger is available as an option. The option is selected by the bit 0 in the Port A Control and Status Register. If this bit is clear, $\overline{INT_1}$ is a negative edge-sensitive interrupt. If set, $\overline{INT_1}$ is a level-sensitive interrupt. When the interrupt occurs, the current instruction execution is completed, processing is suspended, the present MCU state is pushed onto the stack, the interrupt mask bit (I bit) in the Condition Code Register is set and the MCU begins the interrupt sequence. The interrupt service routine normally ends with a RTI instruction which causes the MCU state to be recovered from the stack followed by a return to normal processing.

- **$\overline{INT_2}$ ($\overline{STB}$)**

An $\overline{INT_2}$ ($\overline{STB}$) interrupt request is generated on an $\overline{INT_2}$ negative edge and is controlled by the bit 5 in the Port A Control and Status Register. If this bit is clear, an interrupt will be masked. If set, and $\overline{INT_2}$ interrupt will be enabled. The interrupt request can be cleared by writing a 0 to the bit 7 of the port A Control and Status Register.

The $\overline{INT_2}$ ($\overline{STB}$) can be used as an input data latch signal for Port B. If the bit 3 in the Port A Control and Status Register is set, an $\overline{INT_2}$ interrupt occurs on a negative edge of the $\overline{INT_2}$ signal, which allows Port B to latch data.

- **Timer 1 Interrupt**

A Timer 1 interrupt occurs if any one of the following occurs:

- The Timer Overflow Flag (TOF) bit is set when the Enable Timer Overflow Interrupt (ETOI) bit has been set.
- The Output Compare Flag (OCF) bit is set when the Enable Output Compare Interrupt (EOCI) bit has been set.
- The Input Capture Flag (ICF) bit is set when the Enable Input Capture Interrupt (EICI) bit has been set.

When a Timer 1 interrupt occurs, the present MCU state is pushed onto the stack, the interrupt mask bit (I bit) in the Condition Code Register is set, and the Timer 1 interrupt routine is executed.

Software in the Timer 1 interrupt routine must determine the cause of the Timer 1 interrupts by examining the Timer Control and Status Register (TCSR1).

- **Timer 2 Interrupt**

A Timer 2 interrupt occurs when the Count Match Flag 2 (CMF2) bit is set provided the Enable Count Match 2 Interrupt (ECM2I) bit has been set.

- **Timer 3 Interrupt**

The Timer 3 interrupt occurs when the Count Match Flag 3 (CMF3) bit is set provided the Enable Count Match 3 Interrupt (ECM3I) bit has been set.

- **PCI Interrupt**

An interrupt in the PCI occurs if any one of the following occurs:

- The Master Read End Flag (MREF) bit is set if the Enable Master Read Interrupt (EMRI) bit has been set.
- The Master Write End Flag (MWEF) bit is set if the Enable Master Write Interrupt (EMWI) bit has been set.

- **SCI Interrupt**

An interrupt in the SCI occurs if any one of the following occurs:

- The Transmit Register Empty (TDRE) bit is set provided the Transmit Interrupt Enable (TIE) bit has been set.
- The Overrun/Framing Error (ORFE) bit is set provided the Receive Interrupt Enable (RIE) bit has been set.
- The Receive Data Register Full (RDRF) bit is set provided the RIE bit has been set.

Software in the SCI interrupt routine must determine cause of the SCI interrupts by examining the Transmit/Receive Control and Status Register 1 or Transmit/Receive Control and Status Register 2.

- **A/D Interrupt**

The A/D interrupt occurs if the A/D End Flag (ADEF) bit is set if the Enable A/D End Interrupt (EADEI) bit has been set.

- **Software Interrupt (SWI)**

The SWI is an executable instruction. When the SWI instruction is executed, the current MCU state is pushed onto the stack, the interrupt mask (I) bit in the Condition Code Register is set, and then the SWI interrupt routine is executed. The interrupt service routine normally ends with a RTI instruction which causes the MCU state to be recovered from the stack followed by a return to normal processing.

■ LOW POWER MODES

The HD63705Z0 has three low power modes: wait, stop, and standby.

- **Wait Mode**

The MCU enters the wait mode by the execution of WAIT instruction. In this mode, the internal oscillator remains active but the internal clock is not provided to the CPU, which causes the CPU processing to be stopped. The timer, PCI, SCI and A/D converter remain active; however, these functions cannot be triggered after the MCU enters the wait mode. All registers, RAM and I/O pins remain in their previous state.

The MCU can be released from this mode by interrupts ($\overline{NMI}$, $\overline{INT_1}$, Timer 1, Timer 2, Timer 3, PCI, SCI, $\overline{INT_2}$ ($\overline{STB}$), A/D), $\overline{RES}$ and $\overline{STBY}$. The $\overline{RES}$ signal resets the MCU and the $\overline{STBY}$ signal puts the MCU in the standby mode to be described later. Once the CPU accepts an interrupt, the MCU reverts from the wait mode to the active mode and then executes the interrupt routine. However, if interrupts other than $\overline{NMI}$ ($\overline{INT_1}$, Timer 1, Timer 2, Timer 3, PCI, SCI, $\overline{INT_2}$ ($\overline{STB}$), A/D) are masked by the Interrupt Control Register or $\overline{INT_1}$, Timer 1, Timer 2, Timer 3, PCI, SCI, $\overline{INT_2}$ ($\overline{STB}$) or A/D interrupt request, no interrupt is requested and the MCU remains in the wait mode.

- **Stop Mode**

The MCU enters the stop mode by the execution of STOP instruction. In the stop mode, the internal oscillator is turned off, causing the CPU processing and peripherals to be halted. All registers, RAM, and input/output pins remain in their previous state.

The MCU is released from this mode by external interrupts ($\overline{NMI}$, $\overline{INT_1}$, $\overline{INT_2}$ ($\overline{STB}$), $\overline{RES}$ and $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ puts the MCU in the standby mode to be described later. Once an interrupt is accepted, the CPU reverts from the stop mode to the active mode and executes the interrupt routine. However, if $\overline{INT_1}$, $\overline{INT_2}$ ($\overline{STB}$) interrupts are masked by the Interrupt Control Register or $\overline{INT_1}$, $\overline{INT_2}$ ($\overline{STB}$), or interrupt request, there is no interrupt request, so the MCU does not leave the stop mode.

Fig. 56 shows a timing diagram of the release from the stop mode. In case of release caused by an interrupt, the internal oscillator is activated upon input of the interrupt and after a delay time that allows the oscillator to stabilize the CPU becomes active. In case of release by $\overline{RES}$ input, logic low on the $\overline{RES}$ starts the internal oscillator and then a logic high on the $\overline{RES}$ resumes the CPU processing. The duration of $\overline{RES}$ = low must not be less than T_{osc} to allow the oscillator to stabilize.

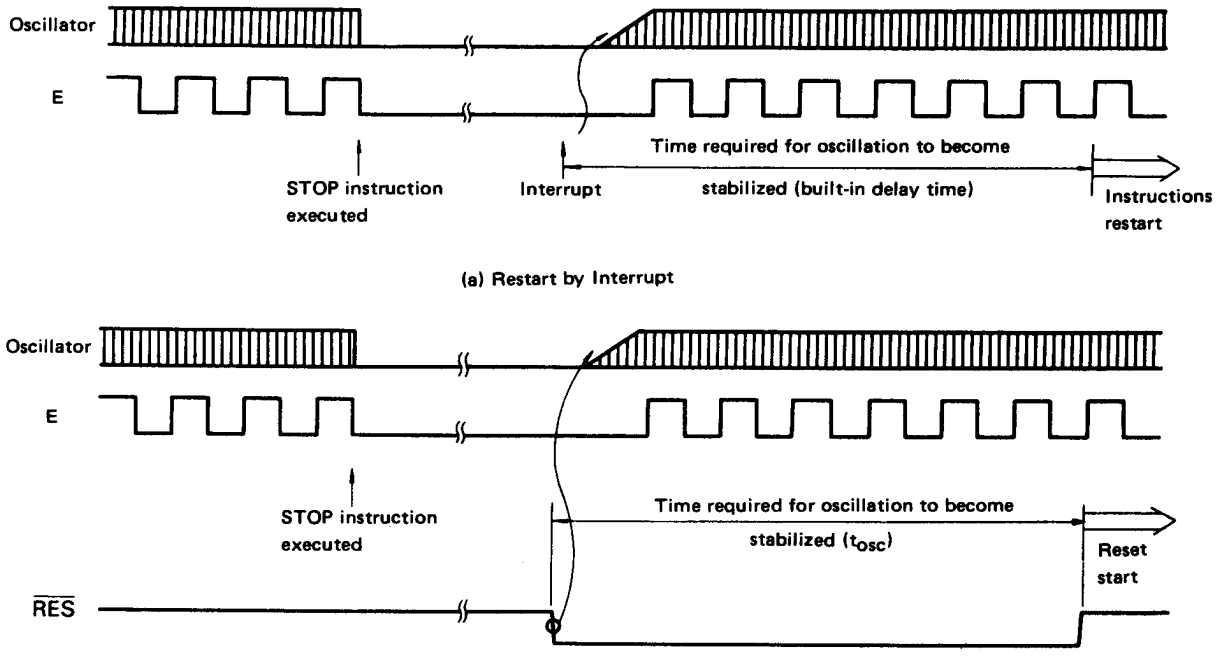


Fig. 54 Timing of Release from Stop Mode

- **Standby Mode**

The MCU enters the standby mode when applying a logic low to the $\overline{STBY}$ input. In this mode, all clocks are halted and the MCU internal state is reset; however, the contents of RAM is held because power is still supplied to the HD63705Z0. All pins except power supply pins (V_{CC}, V_{SS}, AV_{CC} and AV_{SS}), $\overline{STBY}$ pin and XTAL pin (which outputs a logic low) are put in a high impedance state.

Logic high on the $\overline{STBY}$ pin allows the MCU to leave this mode; the internal oscillator is turned on and after a delay time that allows the oscillator to stabilize, the CPU is reset to resume the CPU processing. Refer to the timing diagram in Fig. 55.

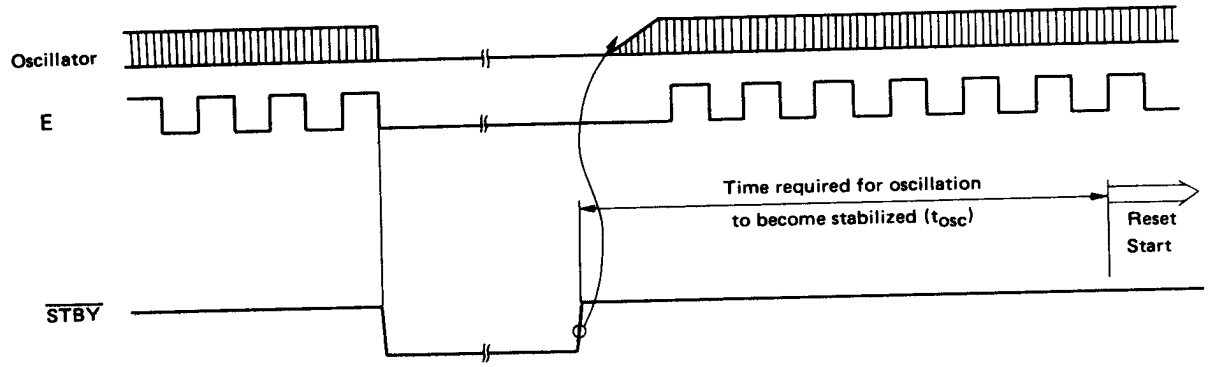


Fig. 55 Timing of Release from Standby Mode

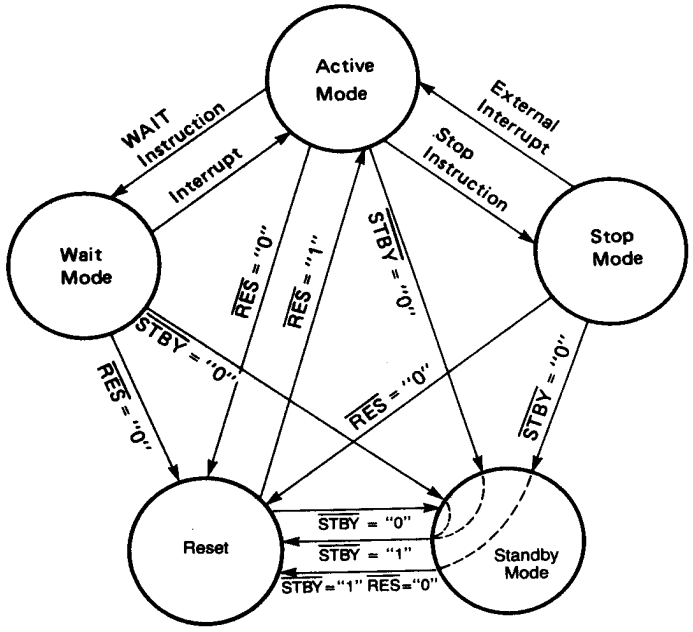


Fig. 56 Transition among Active Mode, Wait Mode, Stop Mode, Standby Mode and Reset Mode

Table 19 State of the MCU in Low Power Modes and Reset Mode

Mode	Started by		State										Released by
			Oscillator	CPU	Timer, SCI PCI, A/D	Register	RAM	I/O	Address Bus	Data Bus	E	R/$\overline{W}$	
Wait Mode	Soft-ware	WAIT instruction	A	S	A	K	K	K	1FFF	T	A	H	$\overline{STBY}$, $\overline{RES}$ or $\overline{NMI}$, $\overline{INT_1}$, $\overline{INT_2}$, Timer 1, Timer 2, Timer 3, SCI, PCI or A/D interrupt
Stop Mode		STOP instruction	S	S	S	K	K	K	1FFF	T	L	H	$\overline{STBY}$, $\overline{RES}$, $\overline{NMI}$, $\overline{INT_1}$, $\overline{INT_2}$
Standby Mode	Hard-ware	$\overline{STBY}$ = low	S	S	S	–	K	T	T	T	T	T	$\overline{STBY}$ = High
Reset	Hard-ware	$\overline{RES}$ = low	A	S	S	I	K	T	1FFF	T	A	H	$\overline{RES}$ = High

H : High, L : Low, T : High Impedance
K : Data is held on output ports and input ports are in high impedance states.
A : Active, S : Stopped, I : Initialized

■ ADDRESSING MODE

The HD63705Z0 MCU has ten addressing modes. They are explained in the following paragraphs.

• Immediate

See Fig. 57. The immediate addressing mode provides access to a constant which does not change during program execution. This access requires an instruction length of 2 bytes. The Effective Address (EA) is the contents of the Program Counter (PC) and the operand is fetched from the byte following the operation code (opcode).

• Direct

See Fig. 58. In the direct addressing mode, the address of the operand is contained in the 2nd byte of the instruction. The user can directly address the lowest 256 bytes in memory. 192 bytes of RAM and I/O registers are located in page zero so that the direct addressing mode may be utilized.

After reset, the Direct Page Register (DPR) is set to page zero; however, pages 0 through 255 can be directly addressed by setting corresponding bits in DPR.

• Extended

See Fig. 59. Extended addressing is used for referencing to all memory locations. The EA is contained in the two bytes following the opcode. An extended addressing instruction requires a length of 3 bytes.

• Relative

See Fig. 60. Relative addressing is only used in branch instructions. When a branch occurs, the contents of the byte following the opcode is added to the PC. EA = (PC)+2+Rel. Where (PC) indicates the contents of the PC and Rel. indicates the 8-bit signed data in the location following the instruction opcode. If no branch occurs, Rel. = 0. When a branch takes place, the program jumps to any byte within the range +129 to −127 of the present instruction. These branch instructions are two bytes long.

• Indexed (No Offset)

See Fig. 61. This addressing mode can access the lowest 256 memory locations. These instructions are one byte long.The EA is the contents of the Index Register.

• Indexed (8-bit Offset)

See Fig. 62. The EA is the sum of the contents of Index Register and byte following the opcode. This addressing mode can access the lowest 511 memory locations. These instructions are 2 bytes long.

• Indexed (16-bit Offset)

See Fig. 63. The EA is the sum of the contents of the Index Register and two bytes following the opcode. This addressing mode can access all memory locations. In this mode all instructions are three bytes long.

• Bit Set/Clear

See Fig. 64. This addressing mode is available to the BSET and BCLR instructions which can set or clear any bit on page 0. The lower 3 bits of the opcode specify the bit to be set or cleared. The byte following the opcode indicates an address within page 0.

On reset Direct Page Register (DPR) is set to page zero; however, setting desired bits in DPR allows the bit set/clear addressing to be applicable to pages 0 through 255 in memory.

• Bit Test and Branch

See Fig. 65. This addressing mode is available to the BRSET and BRCLR instructions which can test the bits within page 0 and can be branched in the relative addressing mode. The address of the byte to be tested is in the second byte following the opcode byte. Individual bits in the byte are specified by the lower 3 bits of the opcode. The relative value is in the third byte and is added to the PC when a branch condition is met. These

instructions are 3 bytes long. The result of the test is written in the carry bit of the Condition Code Register. Setting desired bits in the Direct Page Register allows the bit test and branch addressing to be applicable to pages 0 through 255 in memory.

- **Implied**

See Fig. 66. The implied addressing mode has no EA. All information necessary to execute the instruction is contained in the opcode. Direct manipulation in the Accumulator and Index Register is included in this implied addressing mode. Instructions such as SWI and RTI are also used in this mode. All instructions used in this mode are one byte long.

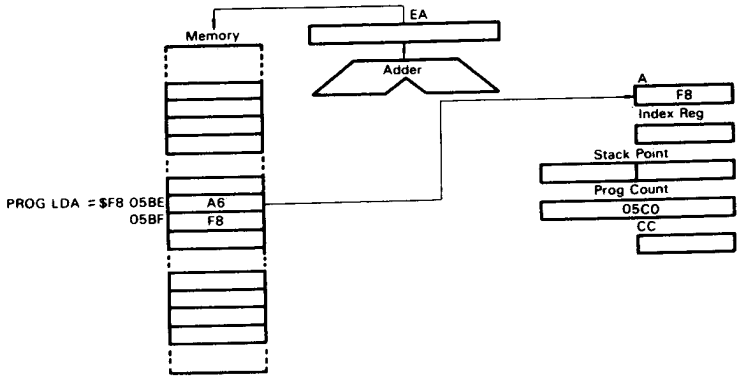


Fig. 57 Example of Immediate Addressing

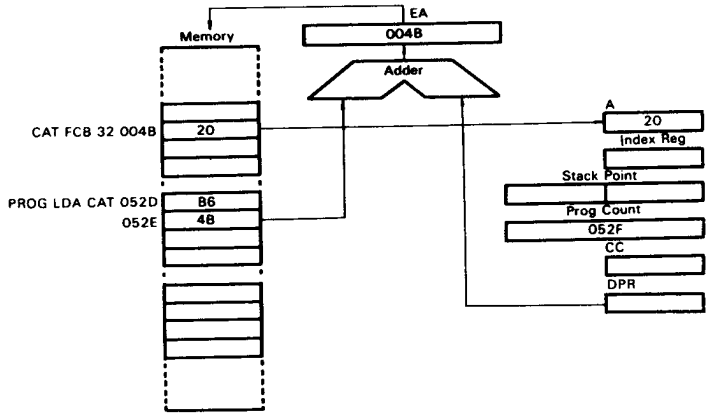


Fig. 58 Example of Direct Addressing

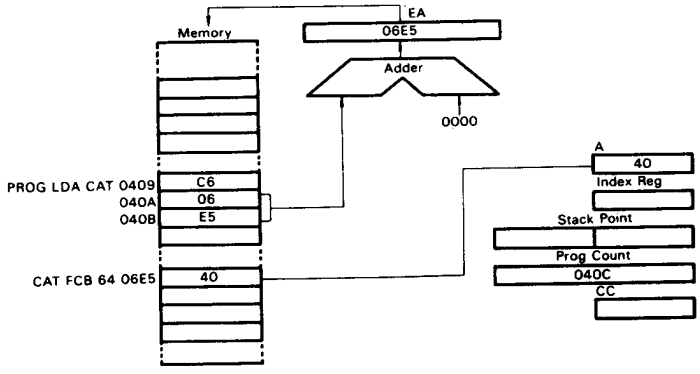


Fig. 59 Example of Extended Addressing

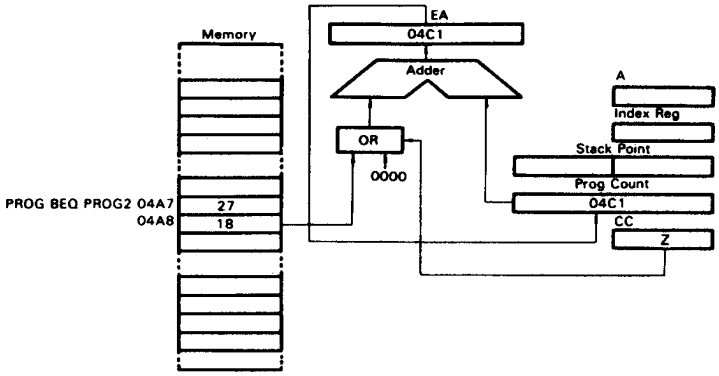


Fig. 60 Example of Relative Addressing

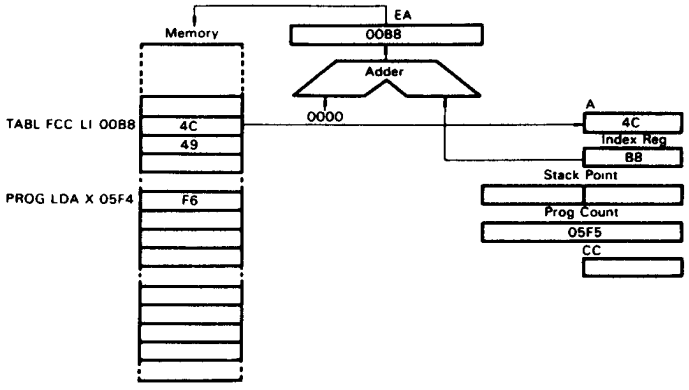


Fig. 61 Example of Indexed Addressing (no offset)

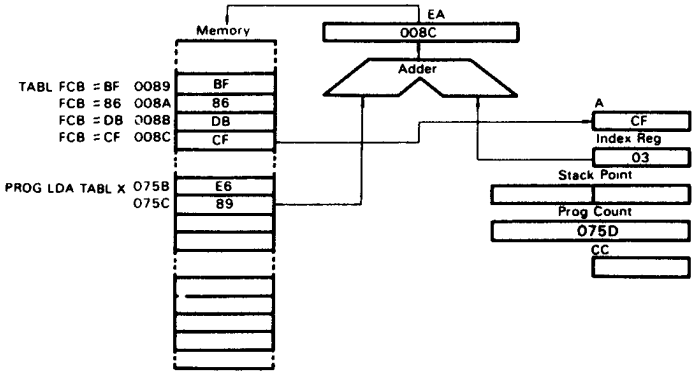


Fig. 62 Example of Indexed Addressing (8 bit offset)

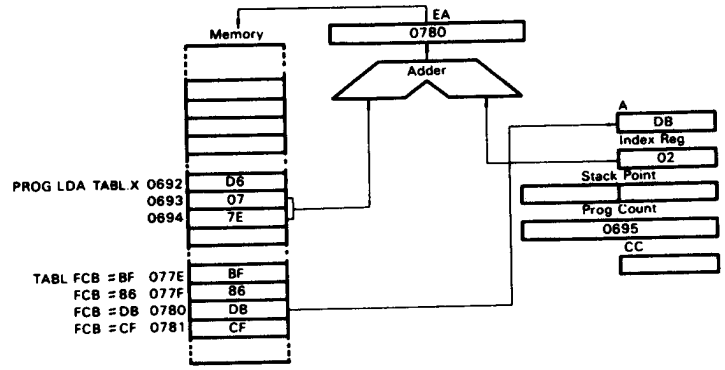


Fig. 63 Example of Indexed Addressing (16 bit offset)

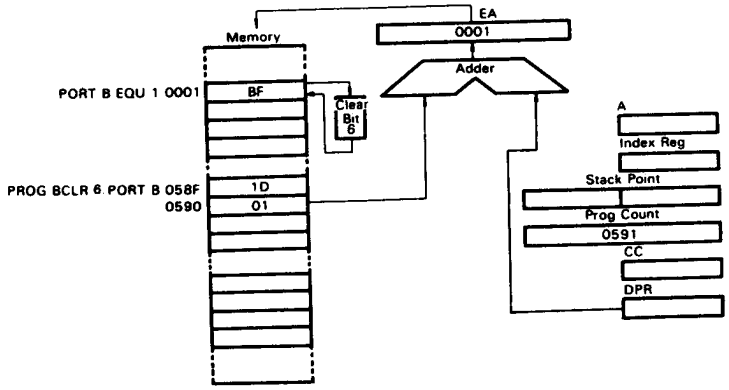


Fig. 64 Example of Bit Set/Clear Addressing

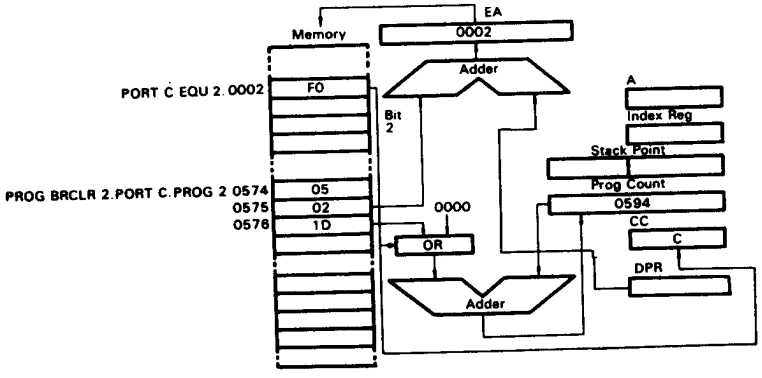


Fig. 65 Example of Bit Test and Branch Addressing

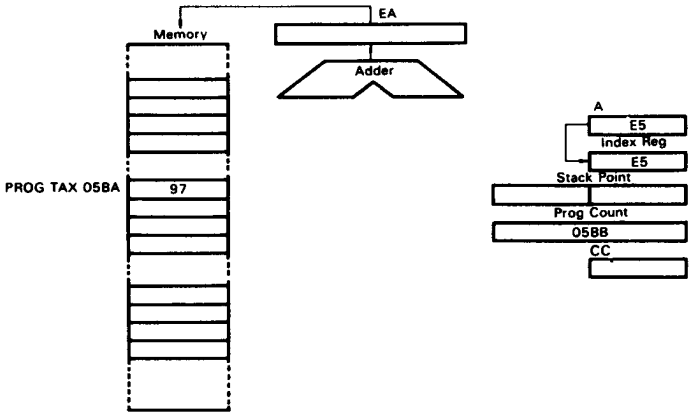


Fig. 66 Example of Implied Addressing

■ INSTRUCTION SET

The HD63705Z0 MCU has 63 basic instructions. They can be classified into five categories: register/memory, read/modify/write, branch, bit manipulation, and control. The details of each instruction are shown in the following tables. All the instructions in a given type are presented in individual tables.

- **Register/Memory Instructions**

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other is obtained from memory by using one of the addressing modes. The unconditional jump (JMP) and the jump to subroutine (JSR) instructions have no register operand. See Table 20.

- **Read/Modify/Write Instructions**

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for zero (TST) instruction is an exception to the read/modify/write instructions since it does not write data. See Table 21.

- **Branch Instructions**

The branch instructions cause a branch from the program when a certain condition is met. See Table 22.

- **Bit Manipulation Instructions**

These instructions are used on any bit in the memory. Two groups are available; one either sets or clears and the other performs the bit test and branch operations. See Table 23.

- **Control Instructions**

These instructions control the MCU operation during program execution. See Table 24.

- **Alphabetical Listing**

Table 25 lists the complete instruction set in alphabetical order.

- **Operation Code Map**

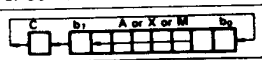
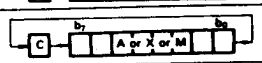
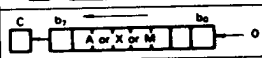
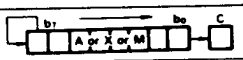
Table 26 is an operation code map for the instructions used on the MCU.

Table 20 Register/Memory Instructions

Operations	Mnemonic	Addressing Modes: Immediate OP	#	~	Direct OP	#	~	Extended OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Indexed (16-Bit Offset) OP	#	~	Boolean/ Arithmetic Operation	Condition Code H	I	N	Z	C
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5	M→A	•	•	^	^	•
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5	M→X	•	•	^	^	•
Store A in Memory	STA	–	–	–	B7	2	3	C7	3	4	F7	1	4	E7	2	4	D7	3	5	A→M	•	•	^	^	•
Store X in Memory	STX	–	–	–	BF	2	3	CF	3	4	FF	1	4	EF	2	4	DF	3	5	X→M	•	•	^	^	•
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5	A+M→A	^	•	^	^	^
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5	A+M+C→A	^	•	^	^	^
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5	A−M→A	•	•	^	^	^
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5	A−M−C→A	•	•	^	^	^
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5	A · M→A	•	•	^	^	•
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5	A+M→A	•	•	^	^	•
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5	A⊕M→A	•	•	^	^	•
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5	A−M	•	•	^	^	^
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5	X−M	•	•	^	^	^
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5	A · M	•	•	^	^	•
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4		•	•	•	•	•
Jump to Subroutine	JSR	–	–	–	BD	2	6	CD	3	7	FD	1	6	ED	2	6	DD	3	7		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 21 Read/Modify/Write Instructions

Operations	Mnemonic	Addressing Modes: Implied(A) OP	#	~	Implied(X) OP	#	~	Direct OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Boolean/Arithmetic Operation	Condition Code H	I	N	Z	C
Increment	INC	4C	1	2	5C	1	2	3C	2	5	7C	1	5	6C	2	6	A+1→A or X+1→X or M+1→M	•	•	^	^	•
Decrement	DEC	4A	1	2	5A	1	2	3A	2	5	7A	1	5	6A	2	6	A−1→A or X−1→X or M−1→M	•	•	^	^	•
Clear	CLR	4F	1	2	5F	1	2	3F	2	5	7F	1	5	6F	2	6	00→A or 00→X or 00→M	•	•	0	1	•
Complement	COM	43	1	2	53	1	2	33	2	5	73	1	5	63	2	6	$\overline{A}$→A or $\overline{X}$→X or $\overline{M}$→M	•	•	^	^	1
Negate (2's Complement)	NEG	40	1	2	50	1	2	30	2	5	70	1	5	60	2	6	00−A→A or 00−X→X or 00−M→M	•	•	^	^	^
Rotate Left Thru Carry	ROL	49	1	2	59	1	2	39	2	5	79	1	5	69	2	6		•	•	^	^	^
Rotate Right Thru Carry	ROR	46	1	2	56	1	2	36	2	5	76	1	5	66	2	6		•	•	^	^	^
Logical Shift Left	LSL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6		•	•	^	^	^
Logical Shift Right	LSR	44	1	2	54	1	2	34	2	5	74	1	5	64	2	6		•	•	0	^	^
Arithmetic Shift Right	ASR	47	1	2	57	1	2	37	2	5	77	1	5	67	2	6		•	•	^	^	^
Arithmetic Shift Left	ASL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6	Equal to LSL	•	•	^	^	^
Test for Negative or Zero	TST	4D	1	2	5D	1	2	3D	2	4	7D	1	4	6D	2	5	A−00 or X−00 or M−00	•	•	^	^	•
Multiply	MUL	42	1	11	–	–	–	–	–	–	–	–	–	–	–	–	X×A→X:A	0	•	•	•	0

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 22 Branch Instructions

Operations	Mnemonic	Addressing Modes: Relative			Branch Test	Condition Code				
		OP	#	~		H	I	N	Z	C
Branch Always	BRA	20	2	3	None	•	•	•	•	•
Branch Never	BRN	21	2	3	None	•	•	•	•	•
Branch IF Higher	BHI	22	2	3	C+Z=0	•	•	•	•	•
Branch IF Lower or Same	BLS	23	2	3	C+Z=1	•	•	•	•	•
Branch IF Carry Clear	BCC	24	2	3	C=0	•	•	•	•	•
(Branch IF Higher or Same)	(BHS)	24	2	3	C=0	•	•	•	•	•
Branch IF Carry Set	BCS	25	2	3	C=1	•	•	•	•	•
(Branch IF Lower)	(BLO)	25	2	3	C=1	•	•	•	•	•
Branch IF Not Equal	BNE	26	2	3	Z=0	•	•	•	•	•
Branch IF Equal	BEQ	27	2	3	Z=1	•	•	•	•	•
Branch IF Half Carry Clear	BHCC	28	2	3	H=0	•	•	•	•	•
Branch IF Half Carry Set	BHCS	29	2	3	H=1	•	•	•	•	•
Branch IF Plus	BPL	2A	2	3	N=0	•	•	•	•	•
Branch IF Minus	BMI	2B	2	3	N=1	•	•	•	•	•
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	3	I=0	•	•	•	•	•
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	3	I=1	•	•	•	•	•
Branch IF Interrupt Line is Low	BIL	2E	2	3	$\overline{NMI}$=0	•	•	•	•	•
Branch IF Interrupt Line is High	BIH	2F	2	3	$\overline{NMI}$=1	•	•	•	•	•
Branch to Subroutine	BSR	AD	2	6	——	•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 23 Bit Manipulation Instructions

Operations	Mnemonic	Addressing Modes: Bit Set/Clear			Bit Test and Branch			Boolean/ Arithmetic Operation	Branch Test	Condition Code				
		OP	#	~	OP	#	~			H	I	N	Z	C
Branch IF Bit n is set	BRSET n(n=0···7)	—	—	—	2·n	3	5	——	Mn=1	•	•	•	•	∧
Branch IF Bit n is clear	BRCLR n(n=0···7)	—	—	—	01+2·n	3	5	——	Mn=0	•	•	•	•	∧
Set Bit n	BSET n(n=0···7)	10+2·n	2	5	—	—	—	1→Mn	——	•	•	•	•	•
Clear Bit n.	BCLR n(n=0···7)	11+2·n	2	5	—	—	—	0→Mn	——	•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 24 Control Instructions

Operations	Mnemonic	Addressing Modes			Boolean Operation	Condition Code				
		Implied				H	I	N	Z	C
		OP	#	~						
Transfer A to X	TAX	97	1	2	A→X	•	•	•	•	•
Transfer X to A	TXA	9F	1	2	X→A	•	•	•	•	•
Set Carry Bit	SEC	99	1	1	1→C	•	•	•	•	1
Clear Carry Bit	CLC	98	1	1	0→C	•	•	•	•	0
Set Interrupt Mask Bit	SEI	9B	1	2	1→I	•	1	•	•	•
Clear Interrupt Mask Bit	CLI	9A	1	2	0→I	•	0	•	•	•
Software Interrupt	SWI	83	1	11		•	1	•	•	•
Return from Subroutine	RTS	81	1	6		•	•	•	•	•
Return from Interrupt	RTI	80	1	9		?	?	?	?	?
Reset Stack Pointer	RSP	9C	1	2	017F→SP	•	•	•	•	•
No-Operation	NOP	9D	1	1	Advance Prog. Cntr. Only	•	•	•	•	•
Decimal Adjust A	DAA	8D	1	2	Converts binary add of BCD characters into BCD format	•	•	^	^	^*
Stop	STOP	8E	1	4	0→I	•	0	•	•	•
Wait	WAIT	8F	1	4	0→I	•	0	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

*Are BCD characters of upper byte 10 or more? (They are not cleared if set in advance.)

Table 25 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		×	×	×		×	×	×			^	•	^	^	^
ADD		×	×	×		×	×	×			^	•	^	^	^
AND		×	×	×		×	×	×			•	•	^	^	•
ASL	×		×			×	×				•	•	^	^	^
ASR	×		×			×	×				•	•	^	^	^
BCC					×						•	•	•	•	•
BCLR									×		•	•	•	•	•
BCS					×						•	•	•	•	•
BEQ					×						•	•	•	•	•
BHCC					×						•	•	•	•	•
BHCS					×						•	•	•	•	•
BHI					×						•	•	•	•	•
(BHS)					×						•	•	•	•	•
BIH					×						•	•	•	•	•
BIL					×						•	•	•	•	•
BIT		×	×	×		×	×	×			•	•	^	^	•
(BLO)					×						•	•	•	•	•
BLS					×						•	•	•	•	•
BMC					×						•	•	•	•	•
BMI					×						•	•	•	•	•
BMS					×						•	•	•	•	•
BNE					×						•	•	•	•	•
BPL					×						•	•	•	•	•
BRA					×						•	•	•	•	•

(to be continued)

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	^	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero	?	Load CC Register From Stack

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
BRN					×						•	•	•	•	•
BRCLR										×	•	•	•	•	^
BRSET										×	•	•	•	•	^
BSET									×		•	•	•	•	•
BSR					×						•	•	•	•	•
CLC	×										•	•	•	•	0
CLI	×										•	0	•	•	•
CLR	×		×			×	×				•	•	0	1	•
CMP		×	×	×		×	×	×			•	•	^	^	^
COM	×		×			×	×				•	•	^	^	1
CPX		×	×	×		×	×	×			•	•	^	^	^
DAA	×										•	•	^	^	^
DEC	×		×			×	×				•	•	^	^	•
EOR		×	×	×		×	×	×			•	•	^	^	•
INC	×		×			×	×				•	•	^	^	•
JMP			×	×		×	×	×			•	•	•	•	•
JSR			×	×		×	×	×			•	•	•	•	•
MUL	×										0	•	•	•	0
LDA		×	×	×		×	×	×			•	•	^	^	•
LDX		×	×	×		×	×	×			•	•	^	^	•
LSL	×		×			×	×				•	•	^	^	^
LSR	×		×			×	×				•	•	0	^	^
NEG	×		×			×	×				•	•	^	^	^
NOP	×										•	•	•	•	•
ORA		×	×	×		×	×	×			•	•	^	^	•
ROL	×		×			×	×				•	•	^	^	^
ROR	×		×			×	×				•	•	^	^	^
RSP	×										•	•	•	•	•
RTI	×										?	?	?	?	?
RTS	×										•	•	•	•	•
SBC		×	×	×		×	×	×			•	•	^	^	^
SEC	×										•	•	•	•	1
SEI	×										•	1	•	•	•
STA			×	×		×	×	×			•	•	^	^	•
STOP	×										•	0	•	•	•
STX			×	×		×	×	×			•	•	^	^	•
SUB		×	×	×		×	×	×			•	•	^	^	^
SWI	×										•	1	•	•	•
TAX	×										•	•	•	•	•
TST	×		×			×	×				•	•	^	^	•
TXA	×										•	•	•	•	•
WAIT	×										•	0	•	•	•

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	^	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 26 Operation Code Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	← HIGH
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	BRSET0	BSET0	BRA	NEG					RTI*	—	SUB						0
1	BRCLR0	BCLR0	BRN	—					RTS*	—	CMP						1
2	BRSET1	BSET1	BHI		MUL*	—			—	—	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	—	CPX						3
4	BRSET2	BSET2	BCC	LSR					—	—	AND						4
5	BRCLR2	BCLR2	BCS	—					—	—	BIT						5
6	BRSET3	BSET3	BNE	ROR					—	—	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					—	TAX*	—	STA				STA(+1)	7
8	BRSET4	BSET4	BHCC	LSL/ASL					—	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					—	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					—	CLI*	ORA						A
B	BRCLR5	BCLR5	BMI	—					—	SEI*	ADD						B
C	BRSET6	BSET6	BMC	INC					—	RSP*	—	JMP(−1)					C
D	BRCLR6	BCLR6	BMS	TST(−1)	TST			TST(−1)	DAA*	NOP	BSR*	JSR(+3)		JSR(+2)		JSR(+3)	D
E	BRSET7	BSET7	BIL	—					STOP*	—	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					WAIT*	TXA*	—	STX				STX(+1)	F
	3/5	2/5	2/3	2/5	1/2	1/2	2/6	1/5	1/*	1/1	2/2	2/3	3/4	3/5	2/4	1/3	

Low

(NOTES) 1. "—" is an undefined operation code.

2. The numbers on the lowest line of each column represent a byte count and number of cycles required (byte count/cycle count).

The cycle count for the mnemonics asterisked (*) is as follows:

RTI	9	TAX	2	MUL	11
RTS	6	RSP	2		
SWI	11	TXA	2		
DAA	2	BSR	6		
STOP	4	CLI	2		
WAIT	4	SEI	2		

3. The parenthesized numbers must be added to the cycle count of the instruction.

- Additional Instructions

In addition to the HD6805 instruction set, the following new instructions are used on the HD63705Z0.

DAA Converts the contents of the accumulator into BCD code.

WAIT Causes the MCU to enter the wait mode. For this mode, see "Wait Mode."

STOP Causes the MCU to enter the stop mode. For this mode, see "Stop Mode".

MUL Multiply the contents of the index register (X) and the accumulator (A). The high-order product is then stored in the index register (X) and the low-order product is stored in the accumulator (A).

■ CYCLE-BY-CYCLE OPERATION

Table 27 Cycle-by-cycle Operation

Address Mode & Instructions	Cycles	Cycle#	Address Bus	R/$\overline{W}$	Data Bus
IMMEDIATE					
ADC, ADD, AND, BIT, CMP, CPX, EOR, LDA, LDX, ORA, SBC, SUB	2	1 2	Op Code Address+1 Op Code Address+2	1 1	Operand Data Next Op Code
DIRECT					
ADC, ADD, AND, BIT, CMP, CPX, EOR, LDA, LDX, ORA, SBC, SUB	3	1 2 3	Op Code Address+1 Address of Operand Op Code Address+2	1 1 1	Address of Operand Operand Data Next Op Code
STA, STX	3	1 2 3	Op Code Address+1 Address of Operand Op Code Address+2	1 0 1	Address of Operand (Data from Acc. (Data from Ix. Next Op Code
JMP	2	1 2	Op Code Address+1 Jump Address	1 1	Jump Address Next Op Code
JSR	6	1 2 3 4 5 6	Op Code Address+1 1FFF Stack Pointer Stack Pointer−1 Stack Pointer−2 Jump Address	1 1 0 0 0 1	Jump Address (LSB) Irrelevant Data Return Address (LSB) Return Address (MSB) DPR First Subroutine Op Code
ASR, CLR, COM, DEC, INC, LSL, LSR, NEG, ROL, ROR	5	1 2 3 4 5	Op Code Address+1 Address of Operand 1FFF Address of Operand Op Code Address+2	1 1 1 0 1	Address of Operand Operand Data Irrelevant Data New Operand Data Next Op Code
TST	4	1 2 3 4	Op Code Address+1 Address of Operand 1FFF Op Code Address+2	1 1 1 1	Address of Operand Operand Data Irrelevant Data Next Op Code
EXTENDED					
ADC, ADD, AND, BIT, CMP, CPX, EOR, LDA, LDX, ORA, SBC, SUB	4	1 2 3 4	Op Code Address+1 Op Code Address+2 Address of Operand Op Code Address+3	1 1 1 1	Address of Operand (MSB) Address of Operand (LSB) Operand Data Next Op Code
STA, STX	4	1 2 3 4	Op Code Address+1 Op Code Address+2 Address of Operand Op Code Address+3	1 1 0 1	Address of Operand (MSB) Address of Operand (LSB) (Data from Acc. (Data from IX. Next Op Code
JMP	3	1 2 3	Op Code Address+1 Op Code Address+2 Jump Address	1 1 1	Jump Address (MSB) Jump Address (LSB) Next Op Code
JSR	7	1 2 3 4 5 6 7	Op Code Address+1 Op Code Address+2 1FFF Stack Pointer Stack Pointer−1 Stack Pointer−2 Jump Address	1 1 1 0 0 0 1	Jump Address (MSB) Jump Address (LSB) Irrelevant Data Return Address (LSB) Return Address (MSB) DPR First Subroutine Op Code
INDEXED (No offset)					
ADC, ADD, AND, BIT, CMP, CPX, EOR, LDA, LDX, ORA, SBC, SUB	3	1 2 3	Op Code Address+1 IX Op Code Address+1	1 1 1	Next Op Code Operand Data Next Op Code
STA, STX	4	1 2 3 4	Op Code Address+1 1FFF IX Op Code Address+1	1 1 0 1	Next Op Code Irrelevant Data (Data from Acc. (Data from IX. Next Op Code

(to be continued)

Address Mode & Instructions	Cycles	Cycle#	Address Bus	R/W	Data Bus
INDEXED (No offset)					
ADC, ADD, AND, BIT, CMP, CPX, EOR, LDA, LDX, ORA, SBC, SUB	3	1	Op Code Address+1	1	Next Op Code
		2	IX	1	Operand Data
		3	Op Code Address+1	1	Next Op Code
STA, STX	4	1	Op Code Address+1	1	Next Op Code
		2	1FFF	1	Irrelevant Data
		3	IX	0	(Data from Acc. / Data from IX.)
		4	Op Code Address+1	1	Next Op Code
JMP	2	1	Op Code Address+1	1	Next Op Code
		2	IX	1	First Op Code of Jump Routine
JSR	6	1	Op Code Address+1	1	Next Op Code
		2	1FFF	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Stack Pointer−2	0	DPR
		6	IX	1	First Subroutine Op Code
ASR, CLR, COM, DEC, INC, LSL, LSR, NEG, ROL, ROR	5	1	Op Code Address+1	1	Next Op Code
		2	IX	1	Operand Data
		3	1FFF	1	Irrelevant Data
		4	IX	0	New Operand Data
		5	Op Code Address+1	1	Next Op Code
TST	4	1	Op Code Address+1	1	Next Op Code
		2	IX	1	Operand Data
		3	1FFF	1	Irrelevant Data
		4	Op Code Address+1	1	Next Op Code
INDEXED (8-bit offset)					
ADC, ADD, AND, BIT, CMP, CPX, EOR, LDA, LDX, ORA, SBC, SUB	4	1	Op Code Address+1	1	Offset
		2	1FFF	1	Irrelevant Data
		3	IX+Offset	1	Operand Data
		4	Op Code Address+2	1	Next Op Code
STA,STX	4	1	Op Code Address+1	1	Offset
		2	1FFF	1	Irrelevant Data
		3	IX+Offset	0	(Data from Acc. / Data from IX.)
		4	Op Code Address+2	1	Next Op Code
JMP	3	1	Op Code Address+1	1	Offset
		2	1FFF	1	Irrelevant Data
		3	IX+Offset	1	First Op Code of Jump Routine
JSR	6	1	Op Code Address+1	1	Offset
		2	1FFF	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Stack Pointer−2	0	DPR
		6	IX+Offset	1	First Subroutine Op Code
ASR, CLR,COM, DEC, INC, LSL, LSR, NEG, ROL, ROR	6	1	Op Code Address+1	1	Offset
		2	1FFF	1	Irrelevant Data
		3	IX+Offset	1	Operand Data
		4	1FFF	1	Irrelevant Data
		5	IX+Offset	0	New Operand Data
		6	Op Code Address+2	1	Next Op Code
TST	5	1	Op Code Address+1	1	Offset
		2	1FFF	1	Irrelevant Data
		3	IX+Offset	1	Operand Data
		4	1FFF	1	Irrelevant Data
		5	Op Code Address+2	1	Next Op Code

(to be continued)

Address Mode & Instructions	Cycles	Cycle#	Address Bus	R/$\overline{W}$	Data Bus
INDEXED (16-bit offset)					
ADC, ADD, AND,	5	1	Op Code Address+1	1	Offset (MSB)
BIT, CMP, CPX,		2	Op Code Address+2	1	Offset (LSB)
EOR, LDA, LDX,		3	1FFF	1	Irrelevant Data
ORA, SBC, SUB		4	IX+Offset	1	Operand Data
		5	Op Code Address+3	1	Next Op Code
STA, STX	5	1	Op Code Address+1	1	Offset (MSB)
		2	Op Code Address+2	1	Offset (LSB)
		3	1FFF	1	Irrelevant Data
		4	IX+Offset	0	(Data from Acc. (Data from IX.
		5	Op Code Address+3	1	Next Op Code
JMP	4	1	Op Code Address+1	1	Offset (MSB)
		2	Op Code Address+2	1	Offset (LSB)
		3	1FFF	1	Irrelevant Data
		4	IX+Offset	1	First Op Code of Jump Routine
JSR	7	1	Op Code Address+1	1	Offset (MSB)
		2	Op Code Address+2	1	Offset (LSB)
		3	1FFF	1	Irrelevant Data
		4	Stack Pointer	0	Return Address (LSB)
		5	Stack Pointer−1	0	Return Address (MSB)
		6	Stack Pointer−2	0	DPR
		7	Ix+Offset	1	First Subroutine Op Code
IMPLIED					
ASR, CLR, COM,	2	1	Op Code Address+1	1	Next Op Code
DEC, INC, LSL,		2	Op Code Address+1	1	Next Op Code
LSR, NEG, ROL,					
ROR, TST					
CLC, NOP, SEC	1	1	Op Code Address+1	1	Next Op Code
RSP, TAX, TXA	2	1	Op Code Address+1	1	Next Op Code
		2	Op Code Address+1	1	Next Op Code
CLI, SEI	2	1	Op Code Address+1	1	Next Op Code
		2	1FFF	1	Irrelevant Data
DAA	2	1	Op Code Address+1	1	Next Op Code
		2	Op Code Address+1	1	Next Op Code
STOP, WAIT	4	1	Op Code Address+1	1	Next Op Code
		2	1FFF	1	Irrelevant Data
		3	1FFF	1	Irrelevant Data
		4	Op Code Address+1	1	Next Op Code
RTI	9	1	Op Code Address+1	1	Next Op Code
		2	1FFF	1	Irrelevant Data
		3	Stack Pointer	1	CC
		4	Stack Pointer+1	1	Acc.
		5	Stack Pointer+2	1	IX.
		6	Stack Pointer+3	1	DPR
		7	Stack Pointer+4	1	Return Address (MSB)
		8	Stack Pointer+5	1	Return Address (LSB)
		9	Return Address	1	First Op Code of Return Routine
RTS	6	1	Op Code Address+1	1	Next Op Code
		2	1FFF	1	Irrelevant Data
		3	Stack Pointer	1	DPR
		4	Stack Pointer+1	1	Return Address (MSB)
		5	Stack Pointer+2	1	Return Address (LSB)
		6	Return Address	1	First Op Code of Return Routine

(to be continued)

Address Mode & Instructions	Cycles	Cycle#	Address Bus	R/$\overline{W}$	Data Bus
SWI	11	1	Op Code Address+1	1	Next Op Code
		2	1FFF	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Stack Pointer−2	0	DPR
		6	Stack Pointer−3	0	Ix.
		7	Stack Pointer−4	0	Acc.
		8	Stack Pointer−5	0	CC
		9	Vector Address 1FFC	1	Address of SWI Routine (MSB)
		10	Vector Address 1FFD	1	Address of SWI Routine (LSB)
		11	Address of SWI Routine	1	First Op Code of SWI Routine
MUL	11	1	Op Code Address+1	1	Next Op Code
		2	1FFF	1	Irrelevant Data
		3	1FFF	1	Irrelevant Data
		4	1FFF	1	Irrelevant Data
		5	1FFF	1	Irrelevant Data
		6	1FFF	1	Irrelevant Data
		7	1FFF	1	Irrelevant Data
		8	1FFF	1	Irrelevant Data
		9	1FFF	1	Irrelevant Data
		10	1FFF	1	Irrelevant Data
		11	Op Code Address+1	1	Next Op Data
RELATIVE					
BCC, BCS, BEQ, BHCC, BHCS, BHI, BIH, BIL, BLS, BMC, BMI, BMS, BNE, BPL, BRA, BRN	3	1	Op Code Address+1	1	Offset
		2	1FFF	1	Irrelevant Data
		3	(Branch Address ·····Test= "1" (Op Code Address+1 ···Test= "0"	1	(First Op Code of Branch Routine (Next Op Code
BSR	6	1	Op Code Address+1	1	Offset
		2	1FFF	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Stack Pointer−2	0	DPR
		6	Branch Address	1	First Op Code of Subroutine
BIT TEST AND BRANCH					
BRCLR, BRSET	5	1	Op Code Address+1	1	Address of Operand
		2	Address of Operand	1	Operand Data
		3	Op Code Address+2	1	Offset
		4	1FFF	1	Irrelevant Data
		5	(Branch Address ·····Test= "1" (Op Code Address+3 ···Test= "0"	1	(First Op Code of Branch Address (Next Op Code
BIT SET/CLEAR					
BCLR, BSET	5	1	Op Code Address+1	1	Address of Operand
		2	Address of Operand	1	Operand Data
		3	1FFF	1	Irrelevant Data
		4	Address of Operand	0	New Operand Data
		5	Op Code Address+2	1	Next Op Code

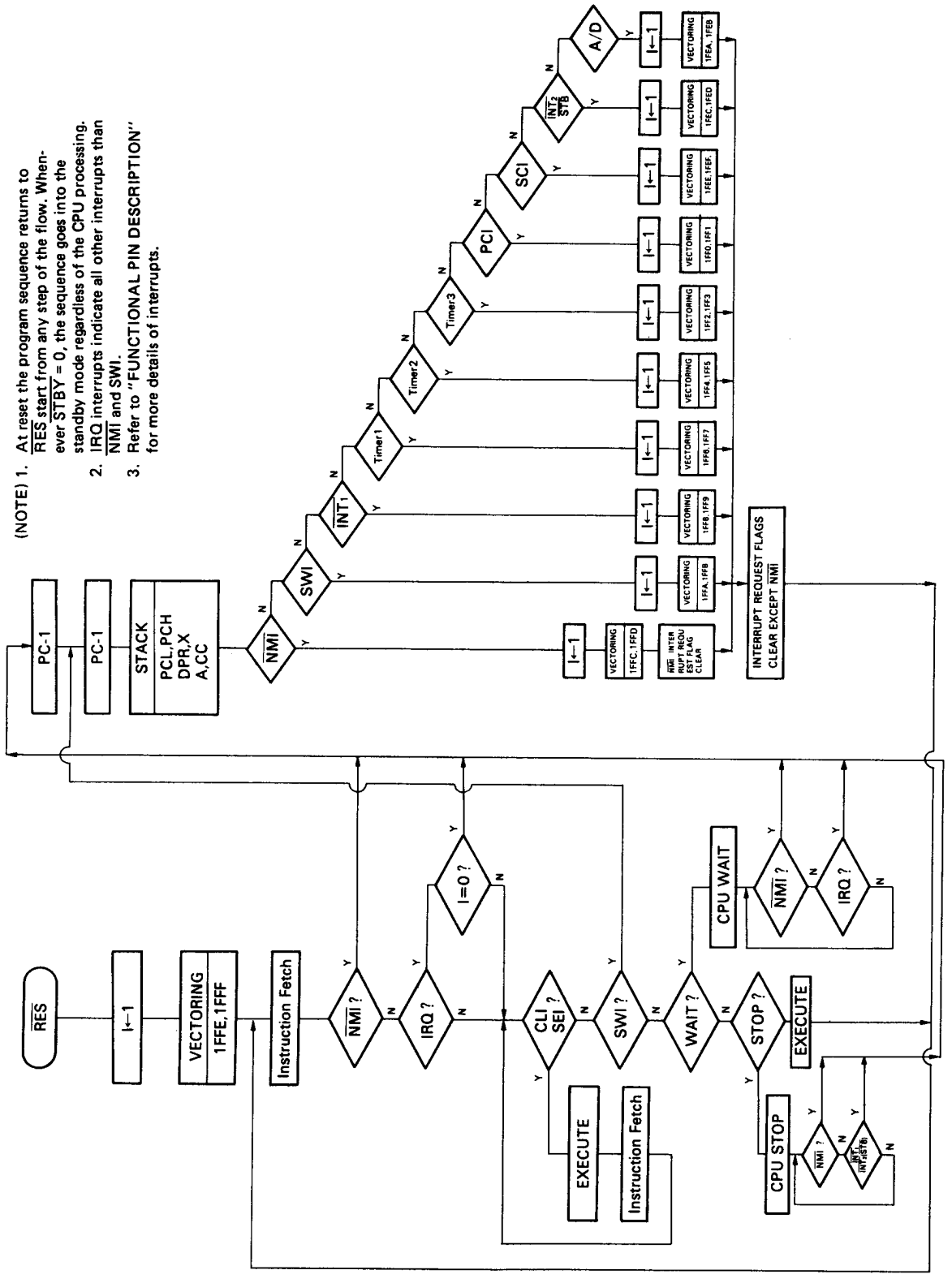


Fig. 67 HD63705Z0 System Flowchart

■ PROGRAMMING THE PROM

• On-Chip PROM Programming

In Mode 4 (PROM programming mode) the HD63705Z0 performs no MCU function but it is programmable as a PROM which is equivalent to the 27256 type. The HD63705Z0 can be put into this mode by applying logic lows to MP_0, MP_1 and $\overline{STBY}$ and applying V_{PP} or V_{CC} to MP_2 pin.

See Table 5 for the pin configuration. Figures in the sub-function column of the list indicate the pin numbers of 27256 type EPROM.

The HD63705Z0 provides 7744 bytes of PROM which is located in $01C0 to $1FFF; however, in this mode the LSI assumes to have 32k bytes of memory located in $0000 to $7FFF to allow the compatibility with the 27256 type EPROM. (Data in addresses $0000 to $01BF and $2000 to $7FFF are not transferred to the HD63705Z0)

When $\overline{CE}$ pin is set to a logic low and $\overline{OE}$ pin is set a logic high after the program voltage (V_{PP}) is applied to V_{PP} pin, the data are written into the EPROM through Port C one byte for each EPROM location. When $\overline{OE}$ pin is set to a logic low after programming, the programmed data is output through Port C. This allows the user to verify the data.

A timing of these signals is shown in Fig. 68.

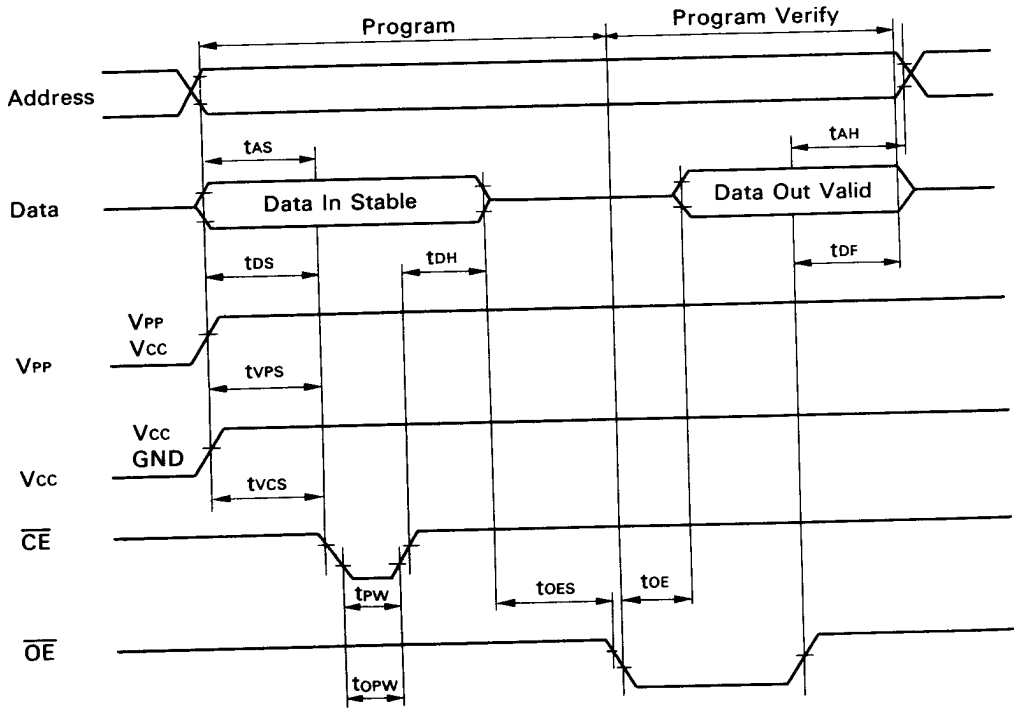


Fig. 68 PROM Programming/Verification Timing

Table 20 shows the state of each pin in PROM programing mode.

Table 28 Pin Conditions in PROM Programming Mode

	MP_0	MP_1	MP_2 (V_{PP})	$\overline{STBY}$	$\overline{CE}$	$\overline{OE}$	Port C (EO_0-EO_7)	Port A, F, H and J
Read	0	0	V_{CC} (1)	0	0	0	Data Out	PROM address input lines are configured as inputs and unused lines are in the standby mode.
Output Disable					0	1	High impedance	
Programming			V_{PP} (1)		0	1	Data In	
Verification					*	0	Data Out	
Program Inhibit					1	1	High impedance	

* 0 or 1

■ ELECTRICAL CHARACTERISTICS

Program Operation

DC Characters (V_{CC} = 6V ± 0.25V, V_{PP} = 12.5V ± 0.3V, Ta = 25°C ± 5°C)

Item	Symbol	Test Condition	min	typ	max	Unit
Input Leakage Current	I_{LI}	V_{in} = 6.25V/0.45V	–	–	2	μA
Output Voltage	V_{OL}	I_{OL} = 2.1 mA	–	–	0.45	V
	V_{OH}	I_{OH} = –400 μA	2.4	–	–	V
Power Supply Current (Active)	I_{CC}		–	–	30	mA
Input Voltage	V_{IL}		–0.1	–	0.8	V
	V_{IH}		2.2	–	V_{CC} + 0.3	V
Programming Current	I_{PP}	$\overline{CE}$ = V_{IL}	–	–	40	mA

(NOTE) 1. V_{PP} (+12.5V) must be applied after V_{CC} (6V) is settled and must be removed before V_{CC}.
2. V_{PP} must not exceed +15V. Be careful to prevent overshoot of the V_{PP} when switching to 12.5V.
3. The device must not be inserted into a board with V_{PP} at 12.5V to prevent damage to the reliability of the device.
4. When $\overline{CE}$ = V_{IL}, V_{PP} must not be changed from V_{CC} to 12.5V or from 12.5V to V_{CC}.

AC Characteristics (V_{CC} = 6V ± 0.25V, V_{PP} = 12.5V ± 0.3V, Ta = 25°C ± 5°C)

Item	Symbol	Test Condition	min	typ	max	Unit
Address Setup Time	t_{AS}		2	–	–	μs
$\overline{OE}$ Setup Time	t_{OES}		2	–	–	μs
Data Setup Time	t_{DS}		2	–	–	μs
Address Hold Time	t_{AH}		0	–	–	μs
Data Hold Time	t_{DH}		2	–	–	μs
Output Disable Delay Time	t_{DF}*		0	–	130	ns*
V_{PP} Setup Time	t_{VPS}		2	–	–	μs
Program Pulse Width	t_{PW}		0.95	1.0	1.05	ms
V_{CC} Setup Time	t_{VCS}		2	–	–	μs
$\overline{OE}$ Output Delay Time	t_{OE}		0	–	150	ns
Overprogramming $\overline{CE}$ Pulse Width	t_{OPW}		2.85	–	78.75	ms

*t_{DF} is defined when the output is in a high impedance state which disallows the output level to be referred.

Switching Characteristics

Test Condition;

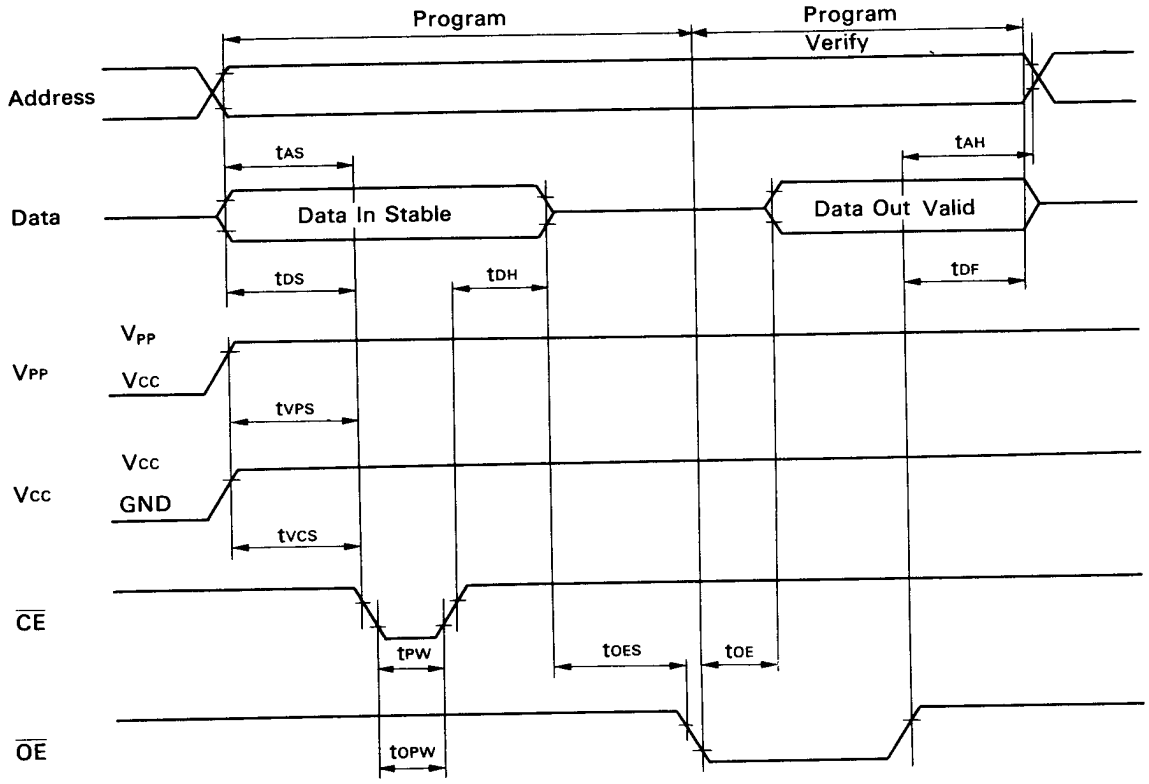
Input pulse level .... 0.8V to 2.2V

Input rise/fall time... ≦ 20 ns

I/O timing reference level

.... input 1V, 2V

output 0.8V, 2V



- **Read Operation**

DC Characteristics (V_{CC} = 5V ± 10%, V_{PP} = V_{CC} ± 0.6V, Ta = 25°C ± 5°C)

Item	Symbol	Test Condition	min	typ	max	Unit
Input Leakage Current	I_{LI}	V_{CC}=5.5V, Vin=GND ~ V_{CC}	–	–	1	μA
Output Leakage Current	I_{LO}	V_{CC}=5.5V, V_{out}=GND ~ V_{CC}	–	–	1	μA
Programming Current	I_{PP}	V_{PP}=V_{CC}+0.6V	–	1	100	μA
Power Supply Current (Active)	I_{CC}	f=1MHz, I_{out}=0mA	–	–	30	mA*
Input Voltage	V_{IL}		–0.3	–	0.8	V
	V_{IH}		2.2	–	V_{CC}+0.3	V
Output Voltage	V_{OL}	I_{OL}=2.1 mA	–	–	0.40	V
	V_{OH}	I_{OH}=–400 μA	2.4	–	–	V

* Excepts straight current through input.

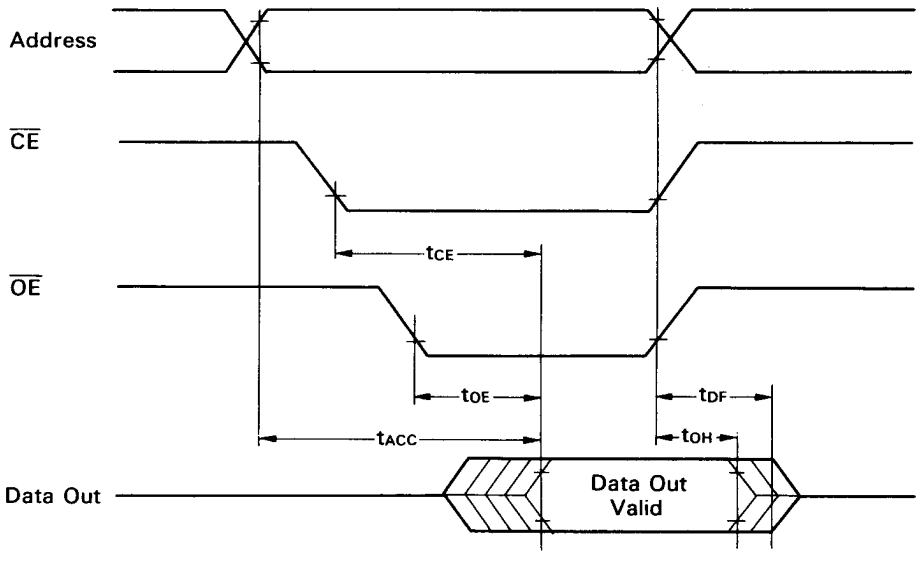
AC Characteristics (V_{CC} = 5V ± 10%, V_{PP} = V_{CC} ± 0.6V, Ta = 25°C ± 5°C)

Item	Symbol	Test Condition	min	max	Unit
Access Time	t_{ACC}	$\overline{CE}$=$\overline{OE}$=V_{IL}	–	500	ns
$\overline{CE}$ Output Delay Time	t_{CE}	$\overline{OE}$=V_{IL}	–	500	ns
$\overline{OE}$ Output Delay Time	t_{OE}	$\overline{CE}$=V_{IL}	10	150	ns
Output Disable Delay Time	t_{DF}	$\overline{CE}$=V_{IL}	0	105	ns*
Data Output Hold Time	t_{OH}	$\overline{CE}$=$\overline{OE}$=V_{IL}	0	–	ns

* t_{DF} is defined when the output is in a high impedance state which disallows the output level to be referred.

Switching Characteristics
Test Condition;
Input pulse level ..... 0.8 to 2.2V
Input rise/fall time.... ≦ 20 ns

Output load ..... 1TTL Gate +100 pF
I/O timing reference level
..... input 1V, 2V
..... output 0.8V, 2V



The information in this data sheet has been carefully checked; however, the contents of this data sheet may be changed and modified without notice. The company shall assume no responsibility for inaccuracies, or any problem involving a patent caused when applying the descriptions in this data sheet.

HITACHI

We make things possible

Hitachi America, Ltd.
Semiconductor and IC Division
2210 O'Toole Avenue, San Jose, CA 95131 1-408-435-8300

AUGUST, 1986 5M Printed in U.S.A.

009097