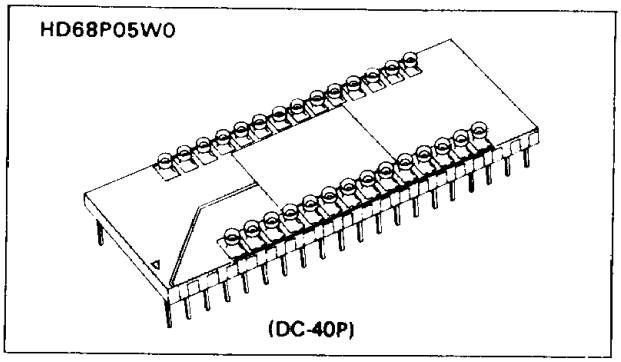
HD68P05W0
MCU (Microcomputer Unit)

The HD68P05W0 is the 8-bit Microcomputer Unit (MCU) which contains a CPU, on-chip clock, RAM, an A/D converter, I/O and two timers. It is designed for the user who needs an economical microcomputer with the proven capabilities of the HD6800-based instruction set. Setting EPROM on the package, this MCU has the same function as the HD6805W1 which has on-chip ROM. It is useful not only for a means of debugging and evaluating the HD6805W1 but also for small-scale-production.

The following EPROMs are available.

4k byte : HN482732A
8k byte : HN482764



■ HARDWARE FEATURES

- 8-Bit Architecture
- 96 Bytes of RAM
- Memory Mapped I/O
- Internal 8-Bit Timer with 7-Bit Prescaler
- Internal 8-bit Programmable Timer (Timer2) with 7-bit Prescaler
- Vectored interrupts : External, Timer and Software
- 23 I/O Ports + 6 Input Ports
  (8 Lines Directly Drive LEDs.)
- On-chip 8 bits A/D Converter
- On-chip Clock Generator
- Master Reset
- Easy for System Development and Debugging
- 5 Vdc Single Supply

■ SOFTWARE FEATURES

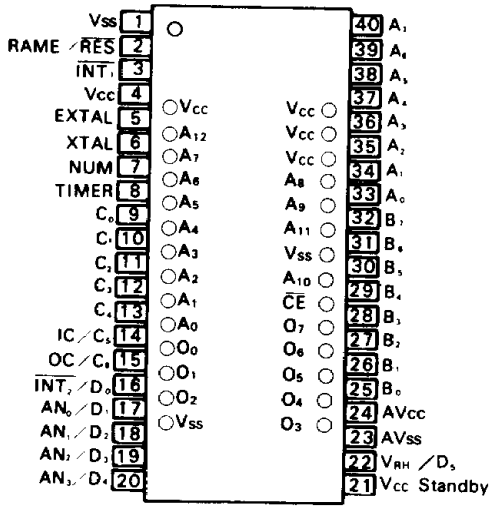
- Similar to HD6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- Ture Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Function
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM and I/O
- Compatible Instruction Set with HD6805

■ TYPE OF PRODUCTS

Type No.	Bus Timing	EPROM Type No.
HD68P05W0	1 MHz	HN482732A-30
		HN482764-3

(NOTE) EPROM is not attached to the MCU.

■ PIN ARRANGEMENT

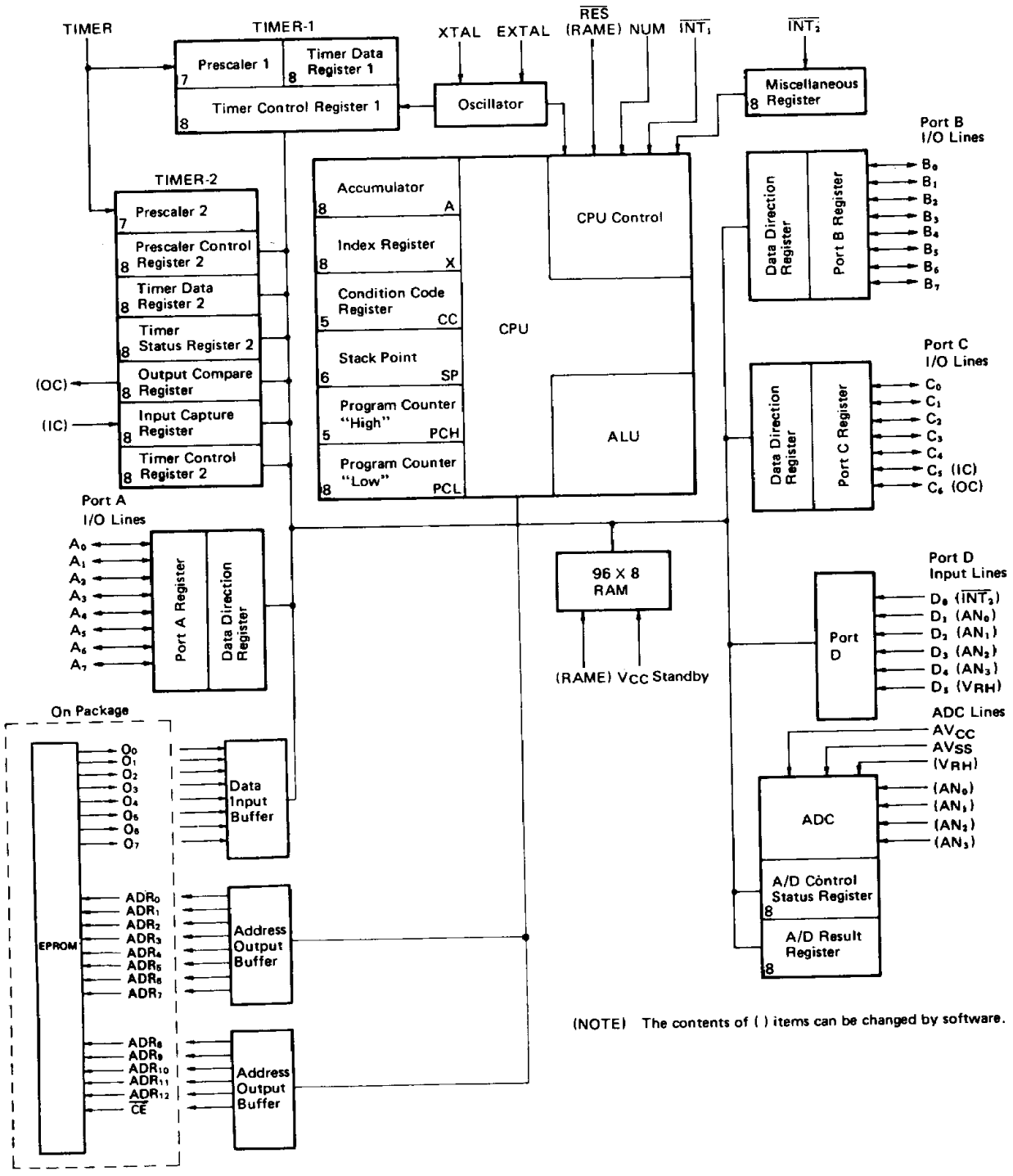


(Top View)

■ PROGRAM DEVELOPMENT SUPPORT TOOLS

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

■ BLOCK DIAGRAM



(NOTE) The contents of ( ) items can be changed by software.

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +7.0	V
Input Voltage (EXCEPT TIMER)	V_{in}	-0.3 ~ +7.0	V
Input Voltage (TIMER)		-0.3 ~ +15.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

(NOTE) This device has an input protection circuit for high quiescent voltage and field, however, be careful not to impress a high input voltage than the insulation maximum value to the high input impedance circuit. To insure normal operation, the following are recommended for V_{in} and V_{out}:
$V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.25V ±0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	$\overline{INT_1}$, $\overline{INT_2}$			3.0	–	V_{CC}	V
	All Others			2.0	–	V_{CC}	V
	Timer			2.0	–	V_{CC}	V
Input "Low" Voltage	$\overline{RES}$	V_{IL}		-0.3	–	0.8	V
	$\overline{INT_1}$, $\overline{INT_2}$			-0.3	–	0.8	V
	EXTAL			-0.3	–	0.6	V
	All Others			-0.3	–	0.8	V
Power Dissipation		P_D		–	–	750	mW
Low Voltage Recover		LVR		–	–	4.75	V
Input Leak Current	TIMER	I_{IL}	V_{in}=0.4V ~ V_{CC}	-20	–	20	μA
	$\overline{INT_1}$, $\overline{INT_2}$			-50	–	50	μA
	EXTAL			-1200	–	0	μA
Standby Voltage	Nonoperation Mode	V_{SBB}		4.0	–	V_{CC}	V
	Operation Mode	V_{SB}		4.75	–	V_{CC}	
Standby Current	Nonoperation Mode	I_{SBB}	V_{SBB}=4.0V	–	–	3	mA

• AC CHARACTERISTICS (V_{CC} = 5.25V ±0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Clock Frequency		f_{cl}		0.4	–	4.0	MHz
Cycle Time		t_{cyc}		1.0	–	10	μs
$\overline{INT}$ Pulse Width		t_{IWL}		t_{cyc}+ 250	–	–	ns
$\overline{RES}$ Pulse Width		t_{RWL}		t_{cyc}+ 250	–	–	ns
TIMER Pulse Width		t_{TWL}		t_{cyc}+ 250	–	–	ns
Oscillation Start-up Time (Crystal Mode)		t_{OSC}	C_L=22pF±20% R_S=60Ω max.	–	–	100	ms
Delay Time Reset		t_{RHL}	External Cap. = 2.2μF	100	–	–	ms
Input Capacitance	XTAL, V_{RH}/Ds	C_{in}	V_{in}=0V	–	–	35	pF
	All Others			–	–	10	pF

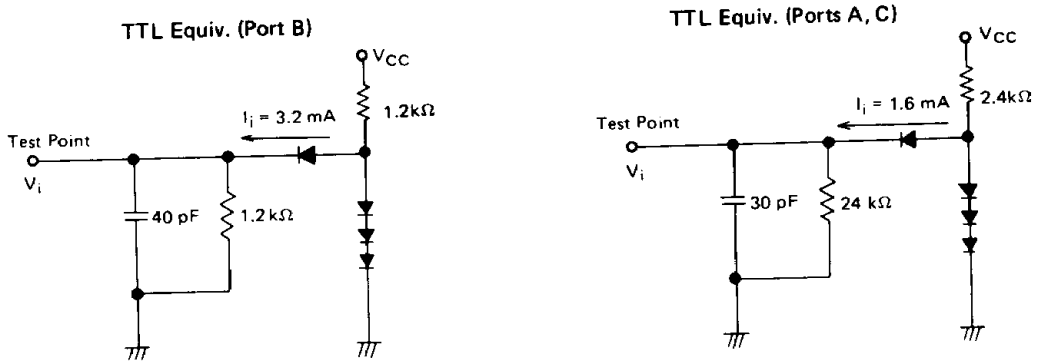
- **PORT ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V ±0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Voltage	Port A	V_{OH}	I_{OH} = -10 μA	3.5	–	–	V
			I_{OH} = -100 μA	2.4	–	–	V
	Port B		I_{OH} = -200 μA	2.4	–	–	V
			I_{OH} = -1 mA	1.5	–	–	V
	Port C		I_{OH} = -100 μA	2.4	–	–	V
Output "Low" Voltage	Ports A and C	V_{OL}	I_{OL} = 1.6 mA	–	–	0.5	V
	Port B		I_{OL} = 3.2 mA	–	–	0.5	V
			I_{OL} = 10 mA	–	–	1.0	V
Input "High" Voltage	Ports A, B, C and D*	V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		-0.3	–	0.8	V
Input Leak Current	Port A	I_{IL}	V_{in} = 0.8V	-500	–	–	μA
			V_{in} = 2V	-300	–	–	μA
	Ports B, C and D*		V_{in} = 0.4V~V_{CC}	-20	–	20	μA

* Port D as digital input

- **A/D CONVERTER ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V±0.5V, V_{SS} = AV_{SS} = GMD, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	min	typ	max	Unit
Analog Power Supply Voltage	AV_{CC}		4.75	5.25	5.75	V
Analog Input Voltage	AV_{in}		0	–	V_{RH}	V
Reference "High" Voltage	V_{RH}	4.75V ≦ V_{CC} ≦ 5.25V	4.0	–	V_{CC}	V
		5.25V < V_{CC} ≦ 5.75V	4.0	–	5.25	V
Analog Multiplexer Input Capacitance			–	–	7.5	pF
Resolution Power			–	8	–	Bit
Conversion Time			76	76	76	t_{cyc}
Input Channels			4	4	4	Channel
Absolute Accuracy		Ta = 25°C	–	–	±1.5	LSB



(NOTE) 1. Load capacitance includes the floating capacitance of the probe and the jig etc.
2. All diodes are 1S2074Ⓗ or equivalent.

Figure 1 Bus Timing Test Loads

■ SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in PIN ARRANGEMENT, are described in the following paragraphs.

- **V_{CC} and V_{SS}**

Voltage is supplied to the MCU using these two pins. V_{CC} is 5.25V ±0.5V. V_{SS} is the ground connection.

- **$\overline{INT_1}$ / $\overline{INT_2}$**

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to INTERRUPTS for additional information.

- **XTAL and EXTAL**

These pins provide connections for the on-chip clock circuit. A crystal (AT cut, 4MHz maximum) or an external signal can be connected to these pins to provide the internal oscillator with varying degrees of stability. Refer to INTERNAL OSCILLATOR for recommendations about these inputs.

- **TIMER**

This pin allows an external input to be used to count for the internal timer circuitry. Refer to TIMER 1 and TIMER 2 for additional information about the timer circuitry.

- **$\overline{RES}$**

This pin allows resetting of the MCU. Refer to RESETS for additional information.

- **NUM**

This pin is not for user application and should be connected to V_{SS}.

- **I/O Lines ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_6$)**

These 23 lines are arranged into three ports (A, B and C). All lines are programmable as either inputs or outputs under software control of the Data Direction Registers. Refer to INPUT / OUTPUT for additional information.

- **Input Lines ($D_0 \sim D_5$)**

These are TTL compatible input lines, in location $0003. These also allow analog inputs to be used for an A/D converter. Refer to INPUT for additional information.

- **V_{CC} Standby**

V_{CC} Standby provides power to the standby portion of the RAM and the STBY PWR and RAME bits of the RAM Control Register. Voltage requirements depend on whether the MCU is in a powerup or powerdown state. In the powerup state, the power supply should provide V_{CC} and must reach V_{SB} before $\overline{RES}$ reaches 4.0V. During powerdown, V_{CC} Standby must remain above V_{SBB} (min) to sustain the standby RAM and STBY PWR bit. While in powerdown operation, the standby current will not exceed I_{SBB}.

It is typical to power both V_{CC} and V_{CC} Standby from the same source during nomal operation. A diode must be used between them to prevent supplying power to V_{CC} during powerdown operation shown Figure 2.

To sustain the standby RAM during powerdown, the following software or hardware are needed.

(1) Software

When clearing the RAM Enable bit (RAME) which is bit 6 of the RAM Control Register at location $001F, the RAM is disabled.

V_{CC} Standby must remain above V_{SBB} (min).

(2) Hardware

When RAME pin is "Low" before powerdown, the RAM is disabled. V_{CC} Standby must remain above V_{SBB} (min).

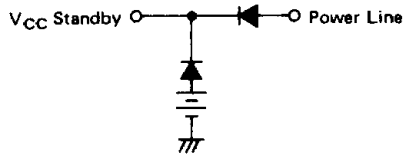


Figure 2 Battery Backup for V_{CC} Standby

- **RAME**

This pin is used for the external control of the RAM. When it is "Low" before powerdown, the RAM is disabled. If V_{CC} Standby remains above V_{SBB} (min), the standby RAM is sustained.

- **AV_{CC}**

This pin is used for the power supply of the A/D converter. When high accuracy is required, a different power source from V_{CC} should be impressed.

Connect to V_{CC} for all other cases. AV_{SS} corresponds to AV_{CC} as a GND terminal.

- **$AN_0 \sim AN_3$**

These pins allow analog inputs to be used for an A/D converter. These inputs are switched by the internal multiplexer and selected by bit 0 and 1 of the A/D Control Status Register (ADCSR: $000E).

- **V_{RH} and AV_{SS}**

The input terminal reference voltage for the A/D converter is "High" (V_{RH}) or "Low" (AV_{SS}). AV_{SS} is fixed at 0V.

- **Input Capture (IC)**

This pin is used for input of Timer2 control. in this case, Port C_5 should be configured as input. Refer to TIMER 2 for more details.

- **Output Compare (OC)**

This pin is used for output of Timer2 when the Output Compare Register is matched with the Timer Data Register 2. In this case, Port C_6 should be configured as an output. Refer to TIMER 2 for more details.

■ REGISTERS

The CPU has five registers available to the programmer, as shown in Figure 3 and explained below.

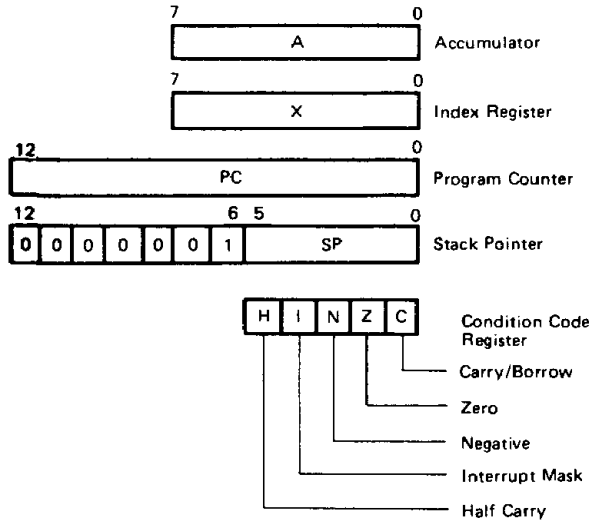


Figure 3 Programming Model

• Accumulator (A)

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

• Index Register (X)

The index register is an 8-bit register used for the indexed addressing mode and contains an 8-bit address that may be added to an offset value to create an effective address. The index register can also be used for limited calculations or data manipulations when using read/modify/write instructions. When not required by a code sequence being executed, the index register can be used as a temporary storage area.

• Program Counter (PC)

The program counter is a 13-bit register that contains the address of the next instruction to be executed.

• Stack Pointer (SP)

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $007F and is decremented as data is being pushed onto the stack and incremented while data is being pulled from the stack. The seven most significant bits of the stack pointer are permanently set to 0000001. During an MCU reset or reset stack pointer (RSP) instruction, the stack pointer is set to location $007F. Subroutines and interrupts may be nested down to location $0041 which allows the programmer to use up to 31 levels of subroutine calls.

• Condition Code Register (CC)

The condition code register is a 5-bit register in which each bit is used to indicate or flag the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained below.

Half Carry (H)

The half carry bit is used during arithmetic operations (ADD or ADC) to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

This bit is set to mask everything. If an interrupt occurs while this bit is set, it is latched and will be processed as soon as the interrupt bit is reset.

Negative (N)

The negative bit is used to indicate that the result of the last arithmetic, logical or data manipulation was negative (bit 7 in a result equal to a logical one).

Zero (Z)

Zero is used to indicate that the result of the last arithmetic, logical or data manipulation was zero.

Carry/Borrow (C)

Carry/borrow is used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts and rotates.

■ TIMER 1

The MCU timer circuitry is shown in Figure 4. The 8-bit counter, Timer Data Register 1 (TDR1), is loaded under program control and counts down toward zero as soon as the clock input is applied. When the TDR1 reaches zero, the timer interrupt request bit (bit 7) in the Timer Control Register 1 (TCR1) is set. The MCU responds to this interrupt by saving the present CPU state in the stack, fetching the timer 1 interrupt vector from locations $0FF8 and $0FF9 and executing the interrupt routine. The timer 1 interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the TCR 1. The interrupt bit (I bit) in the Condition Code Register also prevents a timer 1 interrupt from being processed.

The clock input to the timer 1 can be from an external source applied to the TIMER input pin or it can be the internal ϕ_2 signal. When ϕ_2 is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. The timer 1 continues to count past zero, falling through to $FF from zero and then continuing the count. Thus, the counter (TDR1) can be read at any time by reading the TDR1. This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

At power-up or reset, the prescaler and counter are initialized with all logical ones; the timer 1 interrupt request bit (bit 7) is cleared and the timer 1 interrupt mask bit (bit 6) is set. In order to release the timer 1 interrupt, bit 7 of the TCR 1 must be cleared by software.

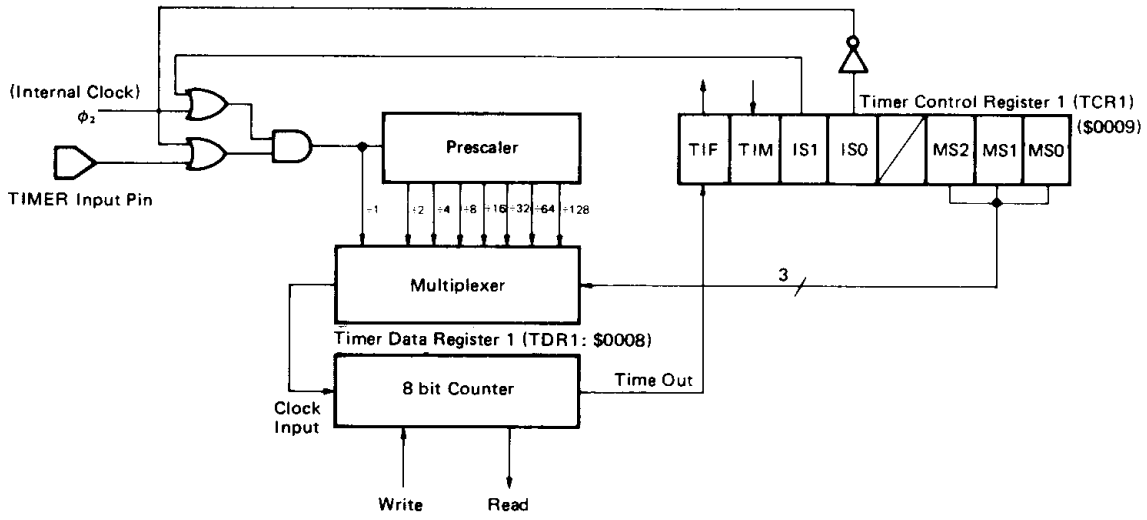
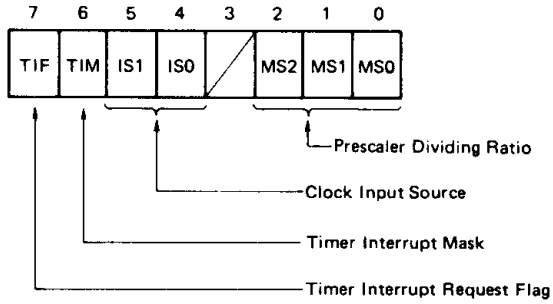


Figure 4 Timer Clock

- **Timer Control Register 1 (TCR1: $0009)**

The Timer Control Register 1 (TCR1: $0009) can control selection of clock input source and prescaler dividing ratio and timer interrupt.

Timer Control Register 1 (TCR1: $0009)



As shown in Table 1, the selection of the clock input source is IS0 and IS1 in the TCR1 (bit 4 and bit 5) and 3 kinds of input are selectable. At reset, internal clock ϕ_2 controlled by the TIMER input (bit 4 = 1, bit 5 = 0) is selected.

The prescaler dividing ratio is selected by MS0, MS1, and MS2 in the TCR1 (bit 0, bit 1, bit 2) as shown in Table 2. The dividing ratio is selectable from eight ways (÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64, ÷128). At reset, ÷1 mode is selected. The prescaler is initialized by writing in the TDR1.

Timer 1 interrupt mask bit (TIM) allows the Timer 1 into interrupt at "0" and masks at "1". Timer 1 interrupt causes Timer 1 interrupt request bit (TIF) to be set. TIF must be cleared by software.

(NOTE) If the MCU Timer1 and Timer2 are not used, the TIMER input pin must be grounded.

Table 1 Selection of Clock Input Source

TCR1		Clock Input Source
Bit 5	Bit 4	
0	0	Internal Clock ϕ_2 *
0	1	ϕ_2 Controlled by TIMER Input
1	0	
1	1	Event Input From TIMER

* The TIMER input pin must be tied to V_{CC}, for uncontrolled ϕ_2 clock input.

Table 2 Selection of Prescaler Dividing Ratio

TCR1			Prescaler Dividing Ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷ 1
0	0	1	÷ 2
0	1	0	÷ 4
0	1	1	÷ 8
1	0	0	÷ 16
1	0	1	÷ 32
1	1	0	÷ 64
1	1	1	÷ 128

■ TIMER 2

The HD68P05W0 includes an 8-bit programmable timer (Timer 2) which can not only measure the input waveform but also generate the output waveform. The pulse width for both input and output waveform can be varied from several microseconds to several seconds.

(NOTE) If the MCU Timer1 and Timer2 are not used, the TIMER input pin must be grounded.

Timer 2 hardware consists of the followings.

- 8-bit Control Register 2
- 8-bit Status Register 2
- 8-bit Timer Data Register 2
- 8-bit Output Compare Register
- 8-bit Input Capture Register
- 5-bit Prescaler Control Register
- 7-bit Prescaler 2

Block Diagram of Timer 2 is shown in Fig. 5.

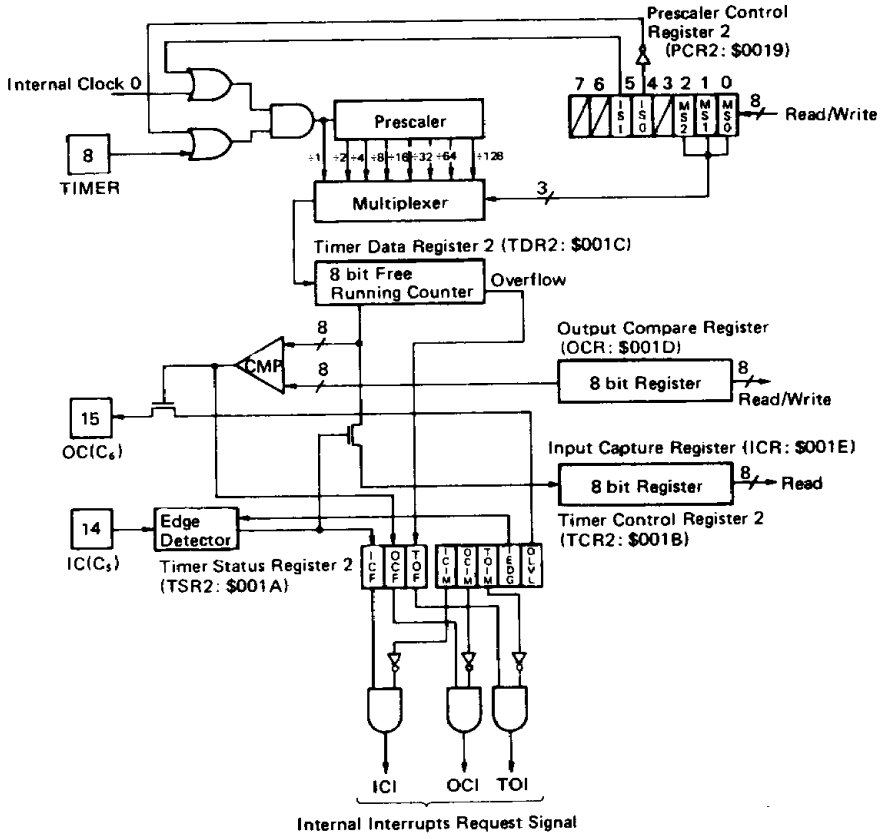


Figure 5 Block Diagram of Timer 2

• Timer Data Register 2 (TDR2; $001C)

The main part of the Timer 2 is the 8-bit Timer Data Register 2 (TDR2) as free-running counter, which is driven by internal clock ϕ_2 or the TIMER input and increments the value. The values in the counter is always readable by software.

The Timer Data Register 2 is Read/Write register and is cleared at reset.

• Output Compare Register (OCR; $001D)

The Output Compare Register (OCR) is an 8-bit read/write register used to control an output waveform. The contents of this register are always compared with those of the TDR2. When these two contents conform to each other, the flag (OCF) in the Timer Status Register 2 (TSR2) is set and the value of the output level bit (OLVL) in the TCR2 is transferred to Port C_6 (OC).

If Port C_6's Data Direction Register (DDR) is "1" (output), this value will appear at Port C_6 (OC). Then the values of OCF and OLVL can be changed for the next compare. The OCR is set to $FF at reset.

• Input Capture Register (ICR; $001E)

The Input Capture Register (ICR) is an 8-bit read-only register used to store the value of the TDR2 when Port C_5 (IC) input transition occurs as defined by the input edge bit (IEDG) of the TCR2.

In order to apply Port C_5 (IC) input to the edge detect circuit, the DDR of Port C_5 should be cleared ("0").*

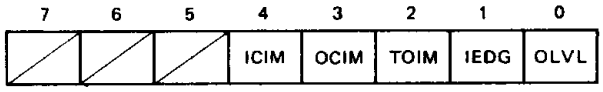
To ensure an input capture under all condition, Port C_5 (IC) input pulse width should be 2 Enable-cycles at least.

*The edge detect circuit always senses Port C_5 (IC) even if the DDR is set with Port C_5 output.

- **Timer Control Register 2 (TCR2; $001B)**

The Timer Control Register 2 (TCR2) consists of an 5-bit register of which all bits can be read and written.

Timer Control Register 2 (TCR2: $001B)



Bit 0 OLVL Output Level

This bit will appear at Port C_6 when the value in the TDR2 equals the value in the OCR, if the DDR of Port C_6 is set. It is cleared by reset.

Bit 1 IEDG Input Edge

This bit determines which level transition of Port C_5 (IC) input will trigger a data store to ICR from the TDR2. When this function is used, it is necessary to clear DDR of Port C_5. When IEDG = 0, the negative edge triggers ("High" to "Low" transition). When IEDG = 1, the positive edge triggers ("Low" to "High" transition). It is cleared by reset.

Bit 2 TOIM Timer Overflow Interrupt Mask

When this bit is cleared, internal interrupt (TOI) is enabled by TOF interrupt but when set, interrupt is inhibited.

Bit 3 OCIM Output Compare Interrupt Mask

When this bit is cleared, internal interrupt (OCI) by OCF interrupt occurs. When set, interrupt is inhibited.

Bit 4 ICIM Input Capture Interrupt Mask

When this bit is cleared, internal interrupt (ICI) by ICF interrupt occurs. When set, interrupt is inhibited.

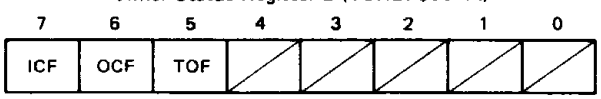
- **Timer Status Register 2 (TSR2: $001A)**

The Timer Status Register 2 (TSR2) is an 8-bit read-only register which indicates that:

(1) A proper leveltransition has been detected on the input pin with a subsequent transfer of the TDR2 value to the ICR (ICF).
(2) A match has been found between the TDR2 and the OCR (OCF).
(3) The TDR2 is zero (TOF).

Each of the event can generate 3 kinds of internal interrupt request and is controlled by an individual inhibit bits in the TCR2. If the I bit in the Condition Code Register is cleared, priority vectors are generated in response to clearing each interrupt mask bit. Each bit is described below.

Timer Status Register 2 (TSR2: $001A)



Bit 5 TOF Timer Overflow Flag

This read-only bit is set when the TDR2 contains $00. It is cleared by reading the TSR2 followed by reading of the TDR2.

Bit 6 OCF Output Compare Flag

This read-only bit is set when a match is found between the OCR and the TDR2. It is cleared by reading the TSR2 and then writing to the OCR.

Bit 7 ICF Input Capture Flag

This read-only bit is set to indicate a proper level transition and cleared by reading the TSR2 and then reading the TCR2.

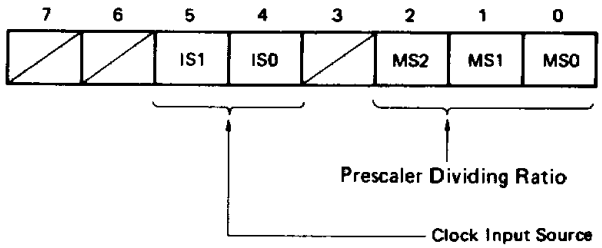
User can write into port C_6 by software.

Accordingly, after port C_6 has output by hardware and is immediately write into by software, simultaneous cyclic pulse control with a short width is easy.

- **Prescaler Control Register 2 (PCR2: $0019)**

The selections of clock input source and prescaler dividing ratio are performed by the Prescaler Control Register 2 (PCR2: $0019).

Prescaler Control Register 2 (PCR2: $0019)



The selection of clock input source is performed in three different ways by bit 4 and bit 5 of the PCR2, as shown in Table 3. At reset, internal clock ϕ_2 controlled by the TIMER input (bit 4 = 1, bit 5 = 0) is selected.

The prescaler dividing ratio is selected by three bits in the PCR2 (bits 0, 1, 2), as shown in Table 4. The dividing ratio can be selected in 8 ways (÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64, ÷128). At reset, ÷1 (bit 0 = bit 1 = bit 2 = 0) is selected.

When writing into the PCR2, or when writing into the TDR2, prescaler is initialized to $FF.

Table 3 Selection of Clock Input Source

PCR2		Clock Input Source
Bit 5	Bit 4	
0	0	Internal Clock ϕ_2 *
0	1	ϕ_2 Controlled by TIMER Input
1	0	
1	1	Event Input from TIMER

* The TIMER input pin must be tied to V_{CC}, for uncontrolled ϕ_2 clock.

Table 4 Selection of Prescaler Dividing Ratio

PCR2			Prescaler Dividing Ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷ 1
0	0	1	÷ 2
0	1	0	÷ 4
0	1	1	÷ 8
1	0	0	÷ 16
1	0	1	÷ 32
1	1	0	÷ 64
1	1	1	÷ 128

CAUTION

(1) When executing Branch instructions* from address $117 to $11C, two flags (TOF and ICF) of the Timer Status Register 2 (TSR2) will be occasionally cleared.

Cause: These instructions have some dummy read cycles so the TSR2 can be read when executing the instructions.

Countermeasure: Don't program branch instructions shown in Table 1 and 3 at address $117 to $11C.

(2) When manipulating or testing the Timer Status Register 2 (TSR2) by Read/Modify/Write instructions**, two flags (TOF and ICF) of the TSR2 will be occasionally cleared.

Cause: These instructions have some dummy read cycles so the TSR2 can be read when executing the instructions.

Countermeasure: Don't use the instructions shown in Table 1, 2 and 4 for read/write/test operation of the TSR2 flags.

* Branch instructions
(Bit Test & Branch ($00 ~ $0F) in Table 5-(1)
Branch ($20 ~ $2F) in Table 5-(4)
SAD)

** Read/Modify/Write instructions
(Bit Test & Branch ($00 ~ $0F) in Table 5-(1)
Bit Set/Clear ($10 ~ $1F) in Table 5-(2)
Memory Manipulation ($30 ~ $3F) in Table 5-(3))

Table 5 Instruction Inhibited to Operate the TSR2

(1) Bit Test and Branch Instruction

Mnemonic	Op Code	# Bytes	# Cycles
BRSET n (n=0~7)	2 · n	3	10
BRCLR n (n=0~7)	01+2 · n	3	10

(2) Bit Set/Clear Instruction

Mnemonic	Op Code	# Bytes	# Cycles
BSET n (n=0~7)	10+2 · n	2	7
BCLR n (n=0~7)	11+2 · n	2	7

(3) Read/Modify/Write Instruction

Mnemonic	Op Code	# Bytes	# Cycles
INC	3C	2	6
DEC	3A	2	6
CLR	3F	2	6
COM	33	2	6
NEG	30	2	6
ROL	39	2	6
ROR	36	2	6
LSL	38	2	6
LSR	34	2	6
ASR	37	2	6
ASL	38	2	6
TST	3D	2	6

(4) Branch Instruction

Mnemonic	Op Code	# Bytes	# Cycles
BRA	20	2	4
BRN	21	2	4
BHI	22	2	4
BLS	23	2	4
BCC	24	2	4
(BHS)	24	2	4
BCS	25	2	4
(BLO)	25	2	4
BNE	26	2	4
BEQ	27	2	4
BHCC	28	2	4
BHCS	29	2	4
BPL	2A	2	4
BMI	2B	2	4
BMC	2C	2	4
BMS	2D	2	4
BIL	2E	2	4
BIH	2F	2	4
BSR	AD	2	8

■ RESETS

The MCU can be reset two ways; by initial power-up and by the external reset input ($\overline{RES}$), see Figure 6. All the I/O ports are initialized to input mode (DDRs are cleared) during reset.

During power-up, a minimum 100 milliseconds is needed before allowing the $\overline{RES}$ input to go "High". This time allows the internal crystal oscillator to stabilize. Connecting a capacitor to the $\overline{RES}$ input, as shown in Figure 7, typically provides

sufficient delay.

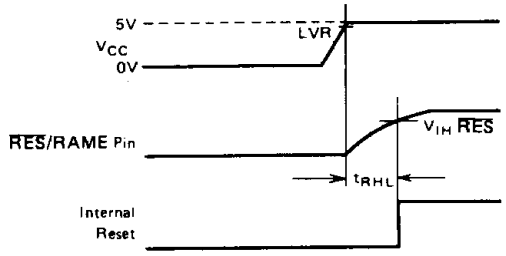


Figure 6 Power Up and Reset Timing

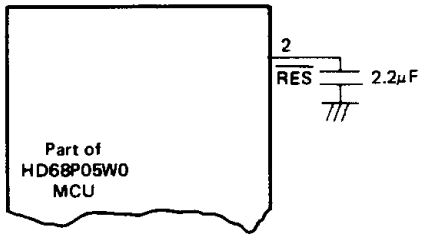


Figure 7 Power Up Reset Delay Circuit

■ INTERNAL OSCILLATOR

The internal oscillator circuit is designed to interface with a crystal (AT cut, 4 MHz max.) which is sufficient to drive it with various stability. As shown in Figure 8, a 22 pF capacitor is required from EXTAL to ground. Crystal specifications are given in Figure 9. Alternatively, EXTAL may be driven with a duty cycle of 50% with XTAL connected to ground.

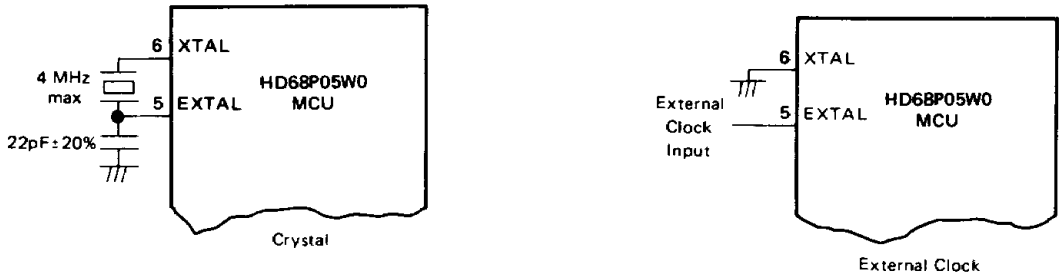
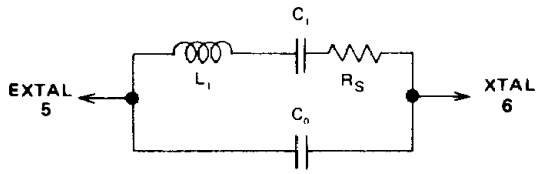


Figure 8 Internal Oscillator Options



AT – Cut Parallel Resonance Crystal
C_0 = 7 pF max.
f = 4 MHz (C_1 = 22pF ± 20%)
R_S = 60 Ω max.

Figure 9 Crystal parameters

Table 6 Interrupt Priorities

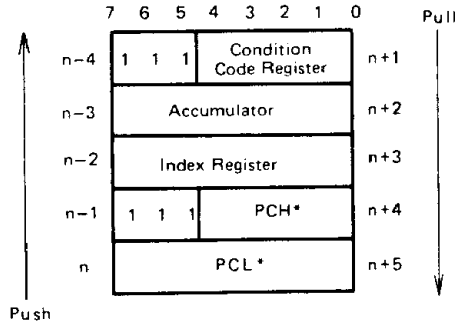
	Interrupt	Priority	Vector Address
4k bytes type	$\overline{RES}$	1	$0FFE, $0FFF
	SWI	2	$0FFC, $0FFD
	$\overline{INT_1}$	3	$0FFA, $0FFB
	Timer/$\overline{INT_2}$	4	$0FF8, $0FF9
	ICI	5	$0FF6, $0FF7
	OCI	6	$0FF4, $0FF5
	OFI	7	$0FF2, $0FF3

	Interrupt	Priority	Vector Address
8k bytes type	$\overline{RES}$	1	$1FFE, $1FFF
	SWI	2	$1FFC, $1FFD
	$\overline{INT_1}$	3	$1FFA, $1FFB
	Timer/$\overline{INT_2}$	4	$1FF8, $1FF9
	ICI	5	$1FF6, $1FF7
	OCI	6	$1FF4, $1FF5
	OFI	7	$1FF2, $1FF3

■ INTERRUPTS

The MCU can be interrupted in seven different ways: through external interrupt input pin ($\overline{INT_1}$ and $\overline{INT_2}$), internal timer interrupt request (Timer 1, ICI, OCI and OFI) and a software interrupt instruction (SWI). $\overline{INT_2}$ and Timer 1 are generated by the same vector address. When interrupt occurs, processing of the program is suspended, the present CPU state is pushed onto the stack. Figure 10 shows interrupt stacking order. Moreover, the interrupt mask bit (I) of the Condition Code Register is set and the external routine priority address is achieved from the special external vector address. After that, the external interrupt routine is executed. The interrupt service routines normally end with a return from interrupt (RTI) instruction which allows the CPU to resume processing of the program prior to the interrupt. The priority interrupts are shown in Table 6 with the vector address that contains the starting address of the appropriate interrupt routine. The interrupt sequence is shown as a flowchart in Figure 11.

Note that the Vector Address when using the 8k byte type EPROM is different from the 4k byte type EPROM.



* For subroutine calls, only PCH and PCL are stacked.

Figure 10 Interrupt Stacking Order

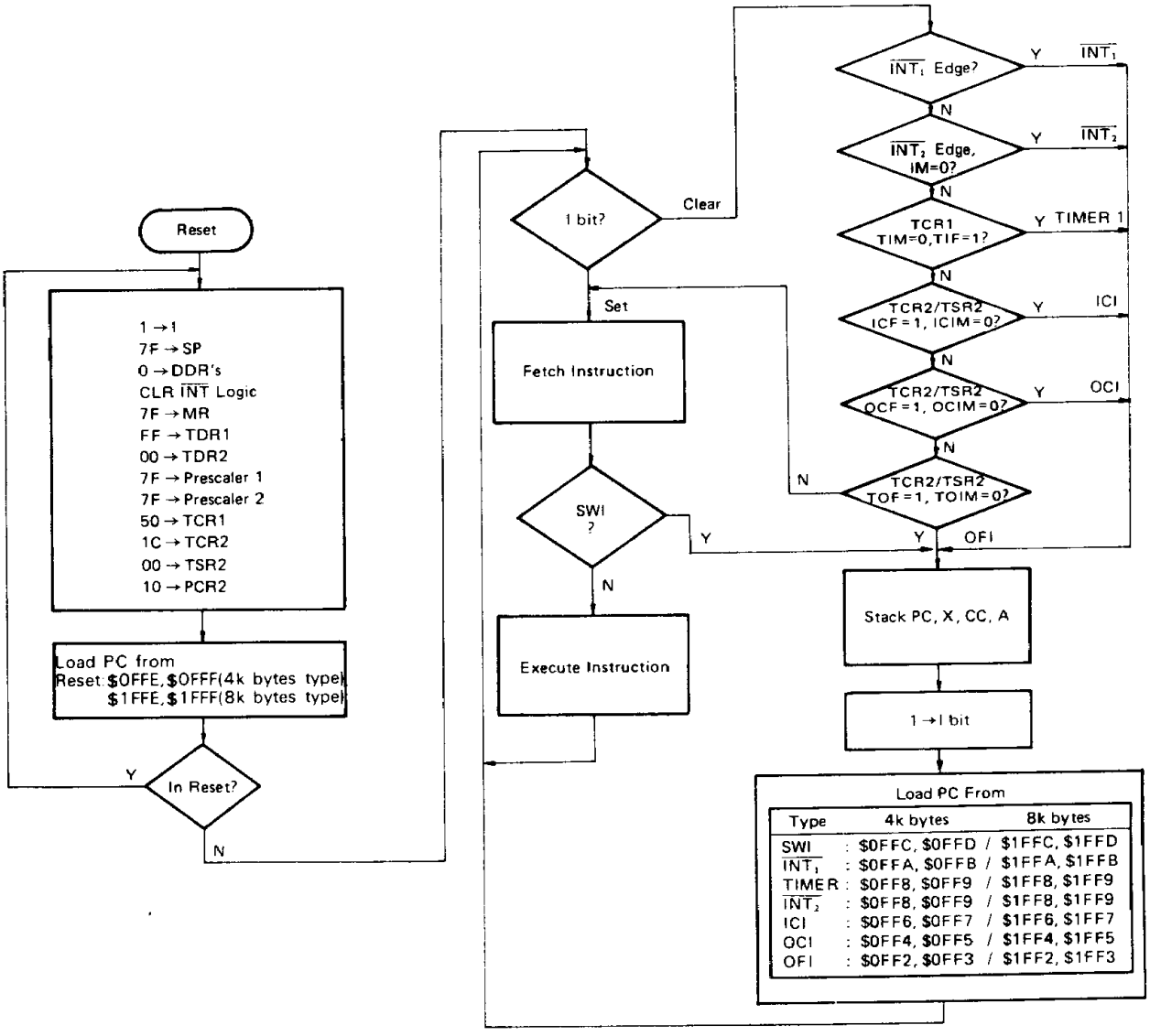
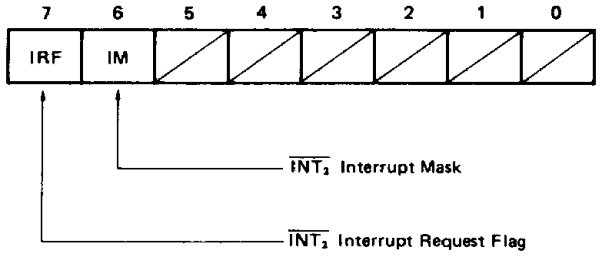


Figure 11 Interrupt Flowchart

- **Miscellaneous Register (MR: $000A)**

The vector address generated by the external interrupt ($\overline{INT_2}$) is the same as that of TIMER1 as shown in Table 6. The miscellaneous register (MR) controls the $\overline{INT_2}$ interrupt.

Miscellaneous Register (MR: $000A)



Bit 7 (IRF) of the MR is used as an $\overline{INT_2}$ interrupt request flag. $\overline{INT_2}$ interrupt occurs at the $\overline{INT_2}$ negative edge, and IRF is set. $\overline{INT_2}$ interrupt or not can be proved by checking IRF by software in the interrupt routine of the vector address ($FF8, $FF9). IRF should be reset by software (BCLR instruction).

Bit 6 (IM) of the MR is an $\overline{INT_2}$ interrupt mask bit. When IM is set, $\overline{INT_2}$ interrupt is disabled. $\overline{INT_2}$ interrupt is also disabled by bit (I) of the Condition Code Register (CC) like other interrupts.

IRF is available for both read and write. However, IRF is not writable by software. Therefore, $\overline{INT_2}$ interrupt cannot be requested by software. At reset, IRF is cleared and IM is set.

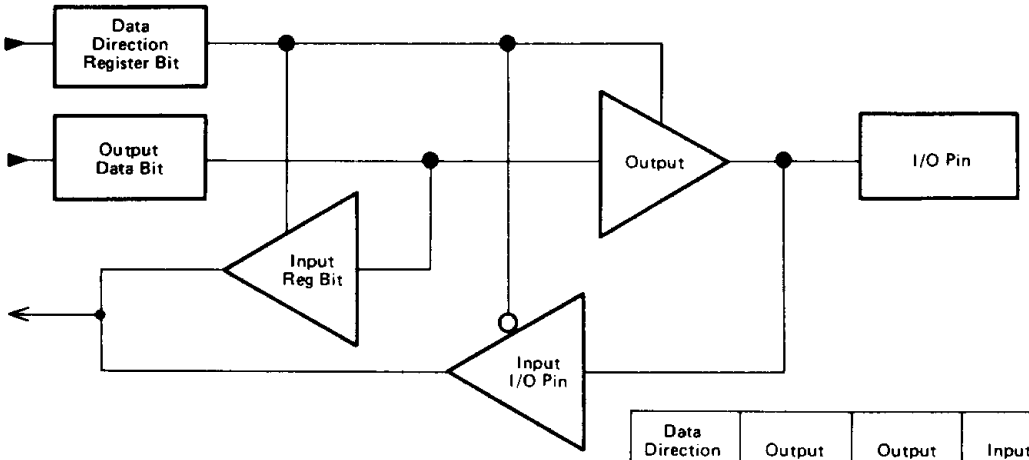
- **INPUT/OUTPUT**

There are 23 input/output pins. All pins are controlled by the Data Direction Register and both input and output are programmable. When programmed as output, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading (See Figure 12.) When Port B is programmed for output, it is capable of sinking 10 mA on each pin (V_{OL} max. = 1V). Furthermore, Port A is CMOS compatible as output. Ports B and C are CMOS compatible as inputs. Some examples of the Port connections are shown in Figure 13.

Port C_5 and C_6 are also used for Timer 2.

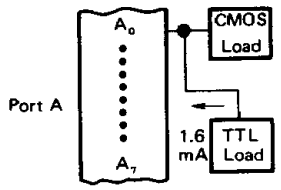
When Port C_5 is used as Timer 2 Input Capture (IC), Port C_5's DDR should be cleared (Port C_5 as input) and bit 4 (ICIM) in the Timer Control Register 2 (TCR2) should be cleared too. The Input Capture Register (ICR) stores the TDR2 when a Port C_5 input transition occures as defined by bit 1 (IDEG) of the TCR2.

When Port C_6 is used as Timer 2 Output Compare (OC), Port C_6's DDR should be set (Port C_6 as output). When the Output Compare Register (OCR) matches the TDR2, bit 0 (OLVL) in the TCR2 is set and OLVL will appear at Port C_6. Port C_6 is writable by software. But the writing by software is unavailable when a match between the TDR2 and the OCR is found at the same time.

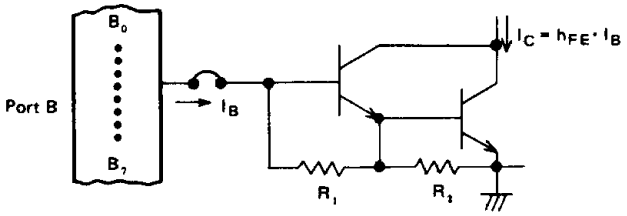


Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	x	3-State	Pin

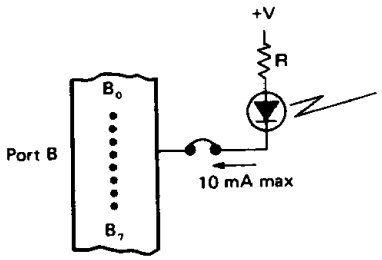
Figure 12 Typical Port I/O Circuitry



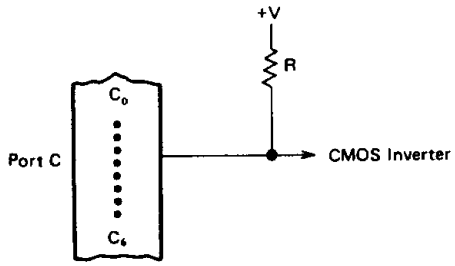
Port A Programmed as output(s), driving CMOS and TTL Load directly.
(a)



Port B Programmed as output(s), driving Darlington base directly.
(b)



Port B Programmed as output(s), driving LED(s) directly.
(c)



Port C Programmed as output(s), driving CMOS load(s), using external pull-up resistors. (d)

Figure 13 Typical Port Connections

■ INPUT

Port D is usable as either TTL compatible inputs or a 4-channel input for an A/D converter. Fig. 14 shows port D logic configuration.

The Port D register at location $003 stores TTL compatible inputs. When using as analog inputs for an A/D converter, refer to "A/D CONVERTER".

■ A/D CONVERTER

The HD68P05W0 has an internal 8 bit A/D converter. The A/D converter, shown in Figure 15, includes 4 analog inputs (AN_0 to AN_3), the Result Register (ADRR) and the Control Status Register (ADCSR).

CAUTION

The MCU has circuitry to protect the inputs against damage due to high static voltages or electric field; however, the design of the input circuitry for the A/D converter, $AN_0 \sim AN_3$, V_{RH} and AV_{CC}, does not offer the same level of protection. Precautions should be taken to avoid applications of any voltage higher than maximum-rated voltage or handled in any environment producing high-static voltages.

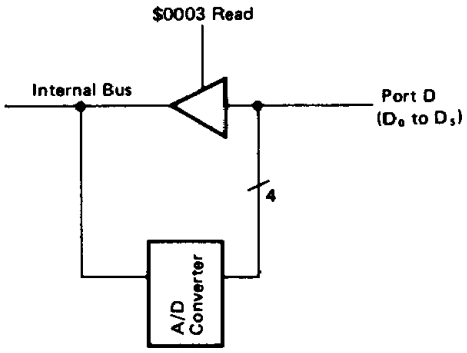


Figure 14 Port D

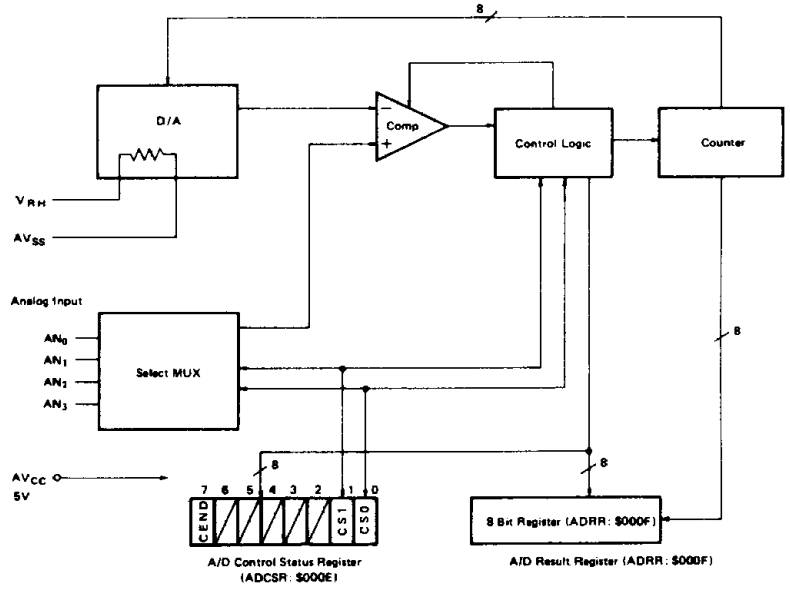


Figure 15 A/D Converter Block Diagram

- **Analog Input (AN_0 to AN_3)**

Analog inputs AN_0 to AN_3 accept analog voltages of 0V to 5V. The resolution is 8 bits (256 divisions) with a conversion time of 76 μs at 1 MHz. Analog conversion starts selecting analog inputs by bit 0 and bit 1 of the ADCSR analog input. Since the CPU is not required during conversion, other user programs can be executed.

Table 7 Analog Input Selection

ADCSR		Analog Input Signal
Bit 1	Bit 0	
0	0	AN_0
0	1	AN_1
1	0	AN_2
1	1	AN_3

- **A/D Control Status Register (ADCSR: $000E)**

The Control Status Register (ADCSR) is used to select analog input pin and confirm A/D conversion termination. An analog input pin is selected by bit 0 and bit 1 as shown in Table 7.

A/D conversion begins when the data is written into bit 0 and bit 1 of the ADCSR. When A/D conversion ends, bit 7 (CEND) is set. Bit 7 is reset after the ADRR is read. Even if bit 7 is set, A/D conversion execution still continues. To end the A/D conversion, the A/D Result Register (ADRR) stores the most current value. During A/D conversion execution, new data is written into the ADCSR selecting the input channel and the A/D conversion execution at that time is suspended. CEND is reset and new A/D conversion begins.

- **A/D Result Register (ADRR: $000F)**

When the A/D conversion ends, the result is set in the A/D Result Register ($000F). When CEND of the ADCSR is set, converted result is obtained by reading the ADRR. Furthermore, CEND is cleared.

■ STANDBY RAM

The portion from $020 to $027 of the RAM can be used for the standby RAM.

When using the standby RAM, V_{CC} Standby should remain above V_{SBB} (min) during powerdown. Consequently, power is provided only to the standby RAM and STBY PWR bit of the RAM Control Register. 8 byte RAM is sustained with small power dissipation. The RAM including the standby RAM is controlled by the RAM Control Register (RCR) or RAME pin.

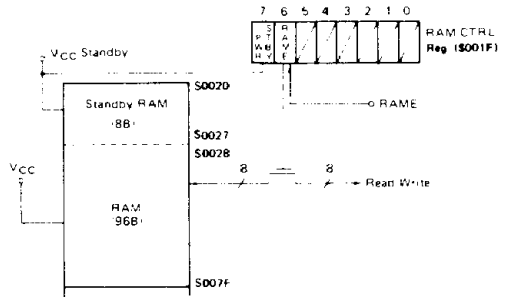
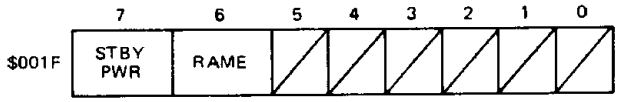


Figure 16 Standby RAM

- **RAM Control Register (RCR: $001F)**

This register at location $01F gives the status information about the RAM. When RAM Enable bit (RAME) is "0", the RAM is disabled. When V_{CC} Standby is greater than V_{SBB}, Standby Power bit (STBY PWR) is set and the standby RAM is sustained during powerdown.

RAM Control Register (RCR: $001F)



Bit 6 RAM Enable

RAME bit is set or cleared by either software or hardware. When the MCU is reset, RAME bit is set and the RAM is enabled. If RAME bit is cleared, the user can neither read nor write the RAM.

When the RAM is disabled (logic "0"), the RAM address is invalid.

Bit 7 Standby Power

STBY PWR bit is cleared whenever V_{CC} standby decreases below V_{SBB} (min). This bit is a read/write status bit that the user can read. When this bit is set, it indicates that the standby power is applied and data in the standby RAM is valid.

- **RAME Signal**

RAME bit in the RCR can be cleared when RAME pin goes "Low" by hardware (RAM is disabled). To make standby mode by hardware, set RAME pin "Low" during V_{CC} Standby remains above V_{SBB} (min) and powerdown sequence should be as shown in Fig. 17.

When RAME pin gets "Low" in the powerup state, RAME bit of the RCR is cleared and the RAM is disabled. During powerdown, RAME bit is sustained by V_{CC} Standby. When RAME pin gets "High" in the powerup state, RAME bit of the RCR is set and the RAM is enabled.

RAME pin can be used to control the RAM externally without software.

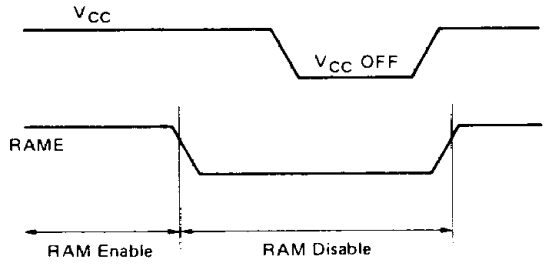


Figure 17 RAM Control Signal (RAME)

■ BIT MANIPULATION

The MCU has the ability to set or clear any single RAM or input/output port (except the data direction registers) with a single instruction (BSET and BCLR). Any bit in the page zero read only memory can be tested by using the BRSET and

BRCLR instructions, and the program branches as a result of its state. This capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines. The example in Figure 18 shows the usefulness of the bit manipulation and test instructions. Assume that bit 0 of port A is connected to a zero crossing detector circuit and that bit 1 of port A is connected to the trigger of a TRIAC which powers the controlled hardware.

This program, which uses only seven bytes of ROM provides turn-on of the TRIAC within 14 microseconds of the zero crossing. The timer is also incorporated to provide turn-on at some later time which permits pulse-width modulation of the controlled power.

```
        .
        .
        .
        .
        .
SELF 1  BRCLR0, PORTA, SELF 1
        BSET 1, PORTA
        BCLR 1, PORTA
        .
        .
        .
        .
        .
```

Figure 18 Bit Manipulation Example

■ ADDRESSING MODES

The MCU has ten addressing modes available for use by the programmer. These modes are explained and illustrated briefly in the following paragraphs.

• Immediate

Refer to Figure 19. The immediate addressing mode accesses constants which do not change during program execution. Such instructions are two bytes long. The effective address (EA) is the PC and the operand is fetched from the byte following the opcode.

• Direct

Refer to Figure 20. In direct addressing, the address of the operand is contained in the secondbyte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in memory. All RAM space, I/O registers and 128 bytes of ROM are located in page zero to take advantage of this efficient memory addressing mode.

• Extended

Refer to Figure 21. Extended addressing is used to reference any location in memory space. The EA is the contents of the two bytes following the opcode. Extended addressing instructions are three bytes long.

• Relative

Refer to Figure 22. The relative addressing mode applies only to the branch instructions. In this mode the contents of the byte following the opcode is added to the program counter when the branch is taken. EA = (PC) + 2 + Rel. Rel is the contents of the location following the instruction opcode with bit 7 being the sign bit. If the branch is not taken, Rel = 0, when a branch takes place, the program goes to somewhere within the range of +129 bytes to -127 bytes of the present instruction. These instructions are two bytes long.

• Indexed (No Offset)

Refer to Figure 23. This mode of addressing accesses the lowest 256 bytes of memory. These instructions are one byte long and their EA is the contents of the index register.

• Indexed (8-bit Offset)

Refer to Figure 24. The EA is calculated by adding the contents of the byte following the opcode to the contents of the index register. In this mode, 511 low memory locations are accessable. These instructions occupy two bytes.

• Indexed (16-bit Offset)

Refer to Figure 25. This addressing mode calculates the EA by adding the contents of the two bytes following the opcode to the index register. Thus, the entire memory space may be accessed. Instructions which use this addressing mode are three bytes long.

• Bit Set/Clear

Refer to Figure 26. This mode of addressing applies to instructions which can set or clear any bit on page zero. The lower three bits in the opcode specify the bit to be set or cleared while the byte following the opcode specifies the address in page zero.

• Bit Test and Branch

Refer to Figure 27. This mode of addressing applies to instructions which can test any bit in the first 256 locations ($0000 through $00FF) and branch to any location relative to the PC. The byte to be tested is addressed by the byte following the opcode. The individual bit within that byte to be tested is addressed by the lower three bits of the opcode. The third byte is the relative address to be added to the program counter if the branch condition is met. These instructions are three bytes long. The value of the bit to be tested is written to the carry bit in the condition code register.

• Implied

Refer to Figure 28. The implied mode of addressing has no EA. All of the information necessary to execute an instruction is contained in the opcode. Direct operations on the accumulator and the index register are included in this mode of addressing. In addition, control instructions such as SWI and RTI belong to this group. All implied addressing instructions are one byte long.

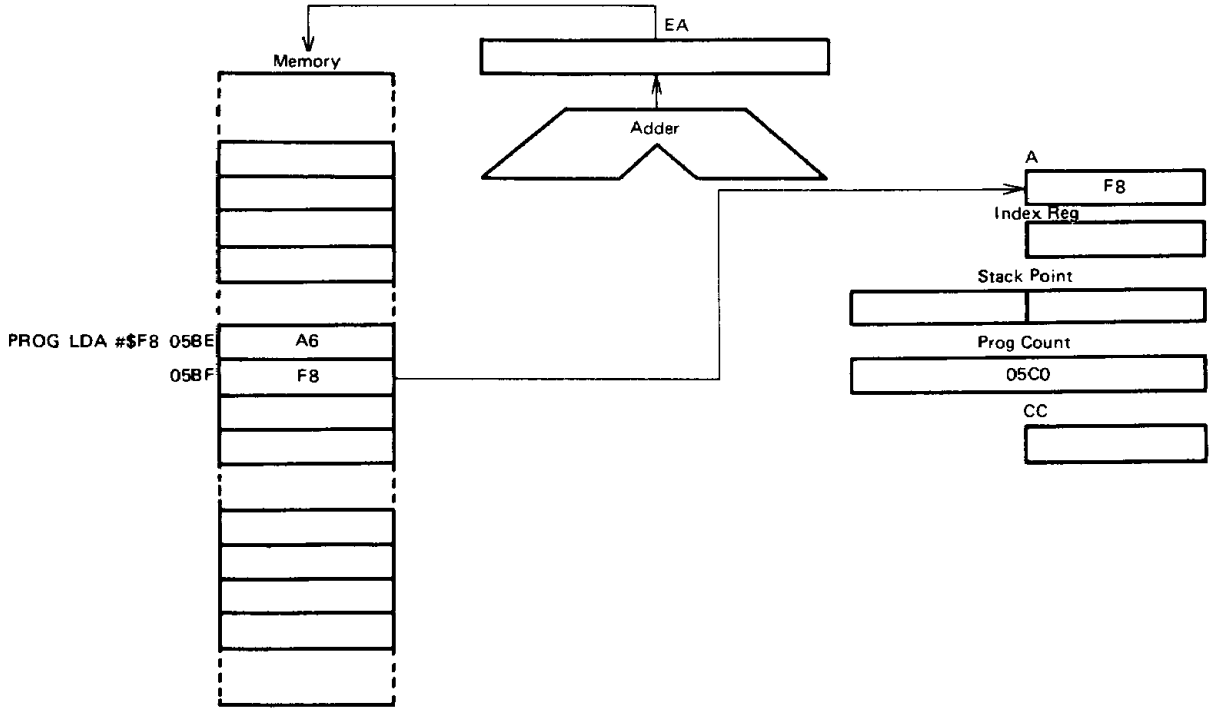


Figure 19 Immediate Addressing Example

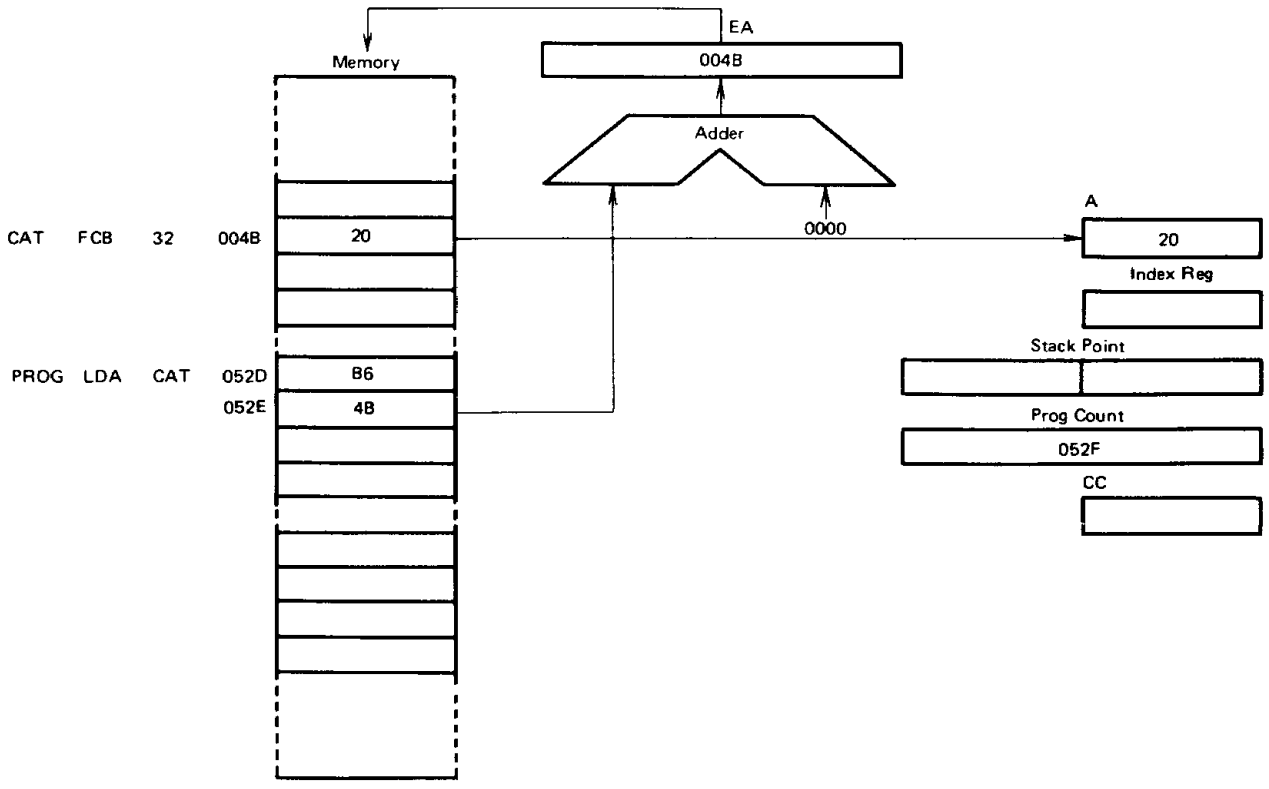


Figure 20 Direct Addressing Example

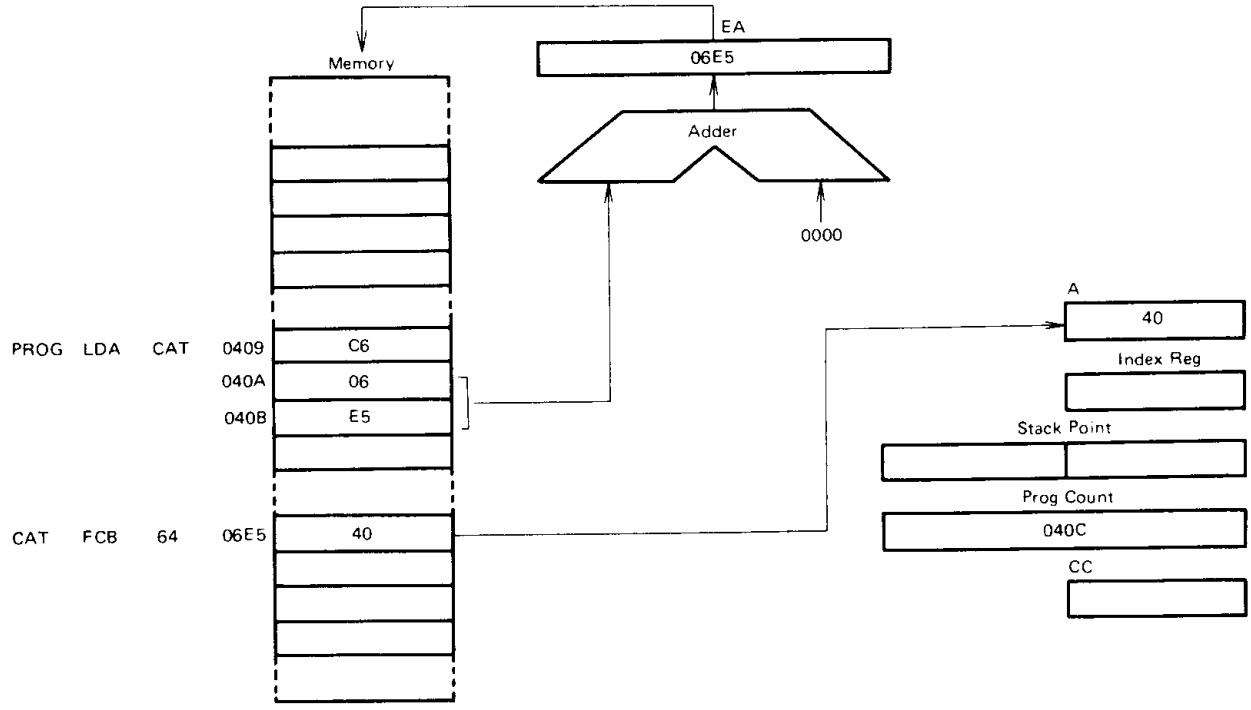


Figure 21 Extended Addressing Example

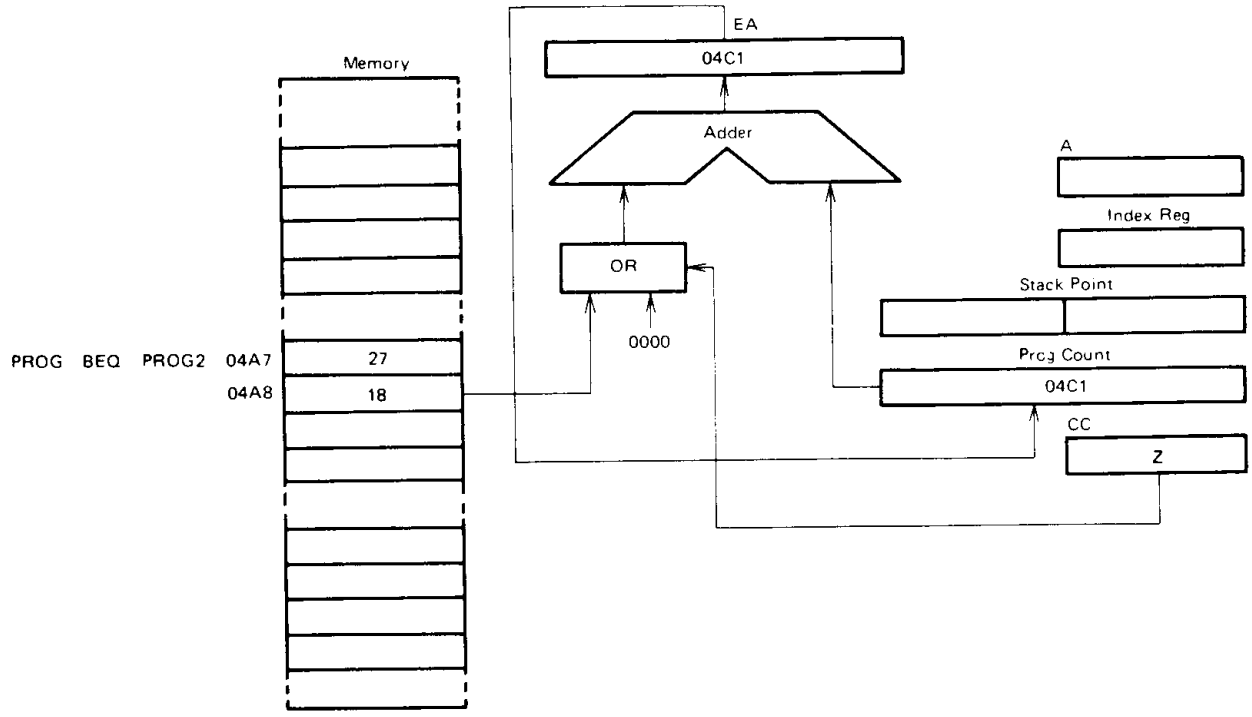


Figure 22 Relative Addressing Example

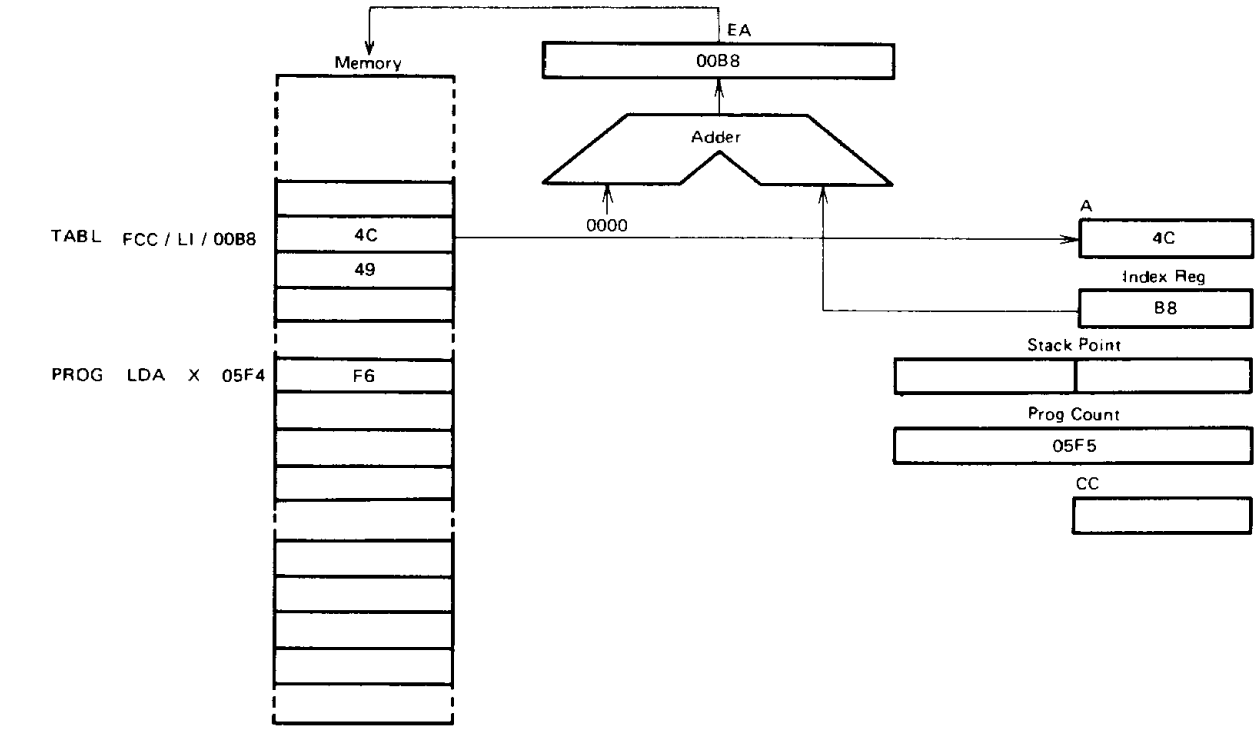


Figure 23 Indexed (No Offset) Addressing Example

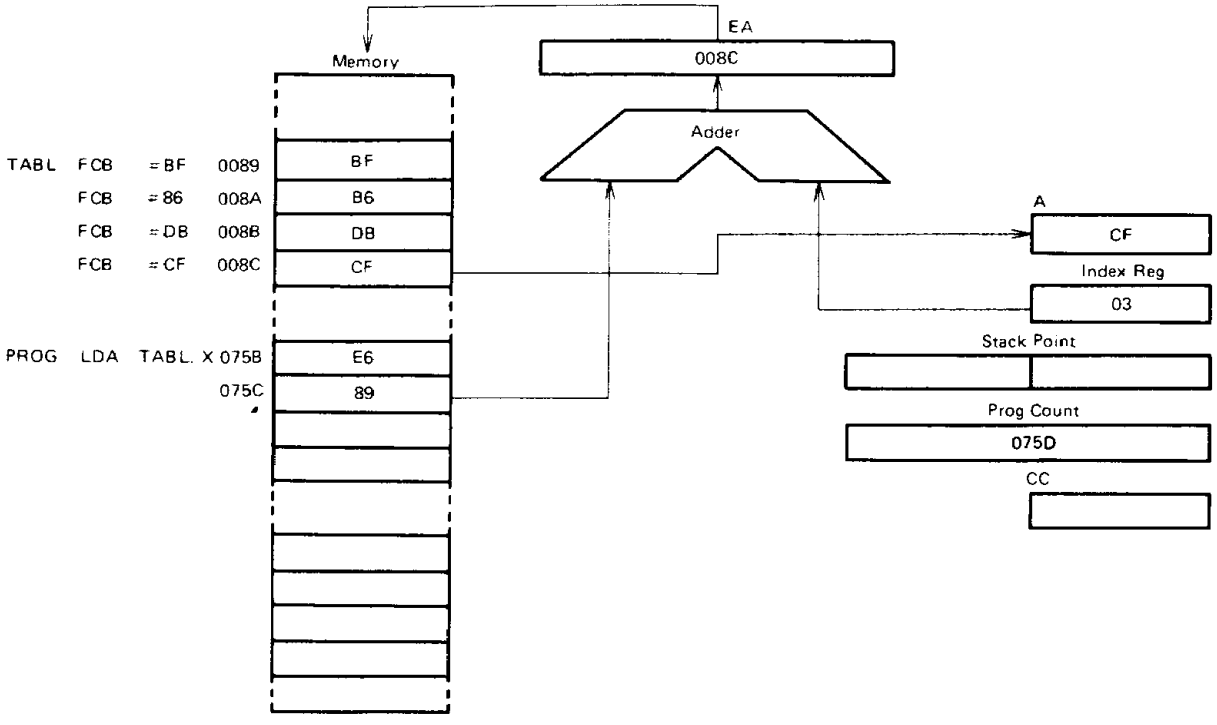


Figure 24 Indexed (8-Bit Offset) Addressing Example

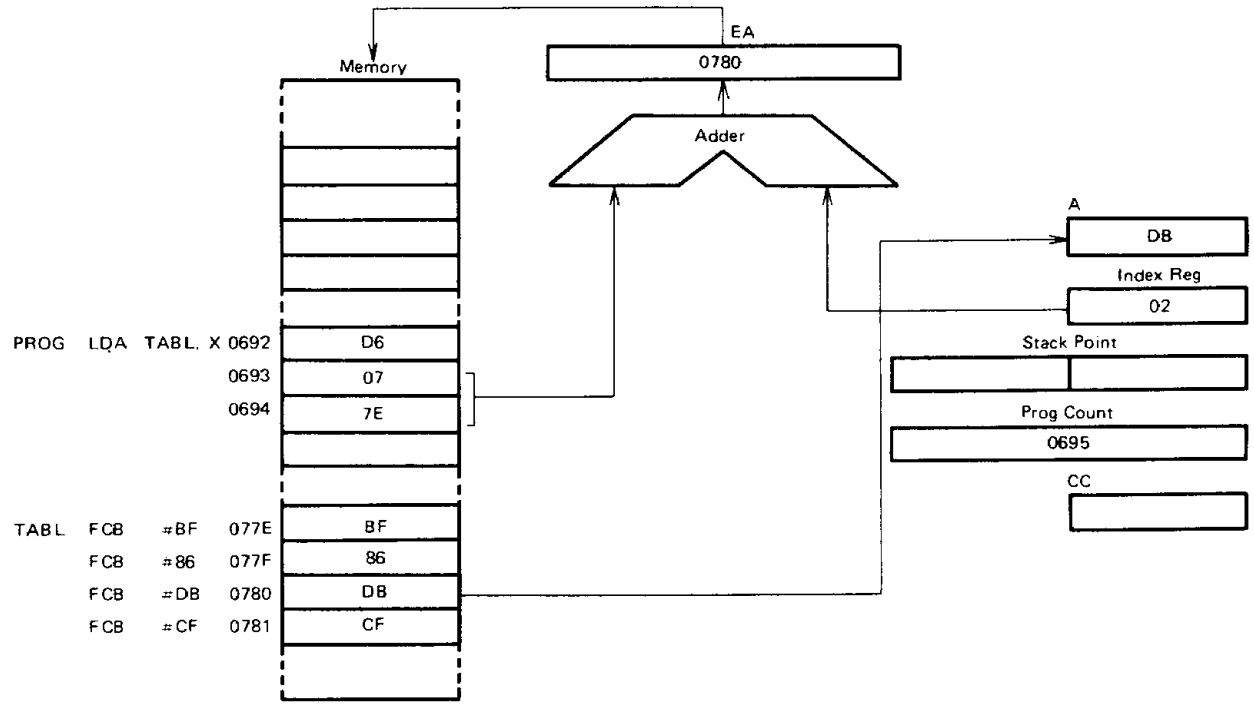


Figure 25 Indexed (16-Bit Offset) Addressing Example

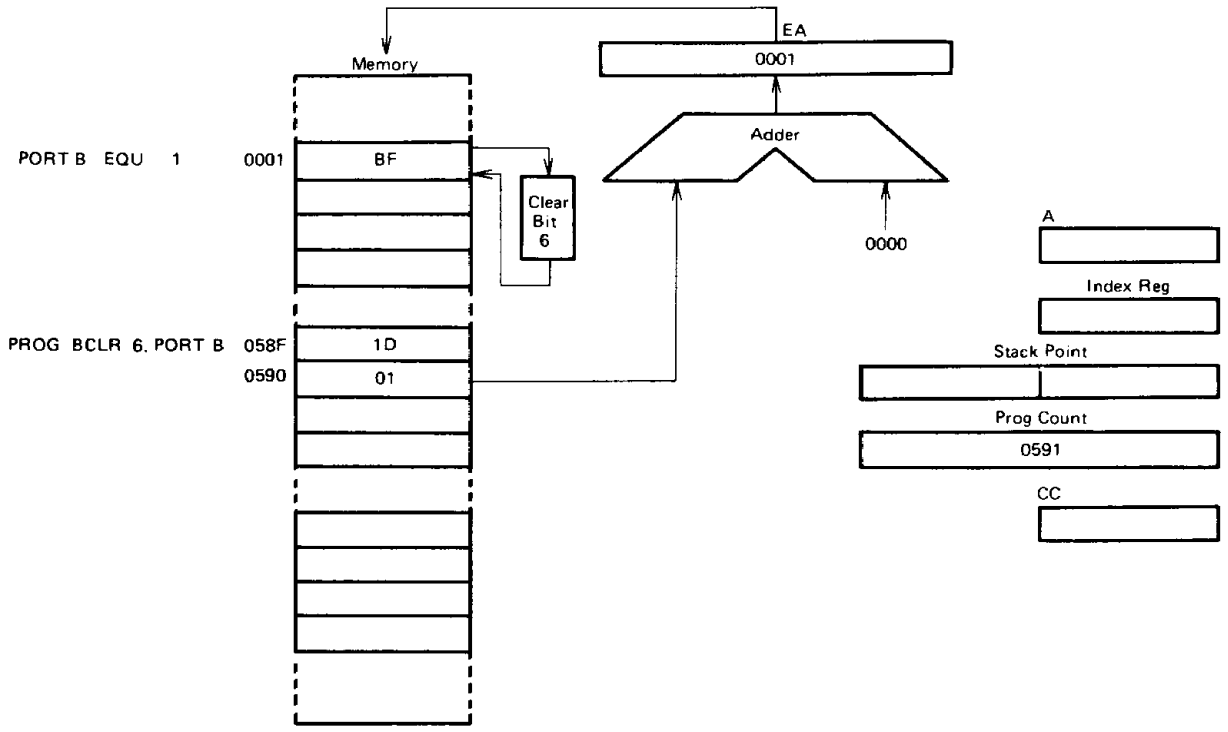


Figure 26 Bit Set/Clear Addressing Example

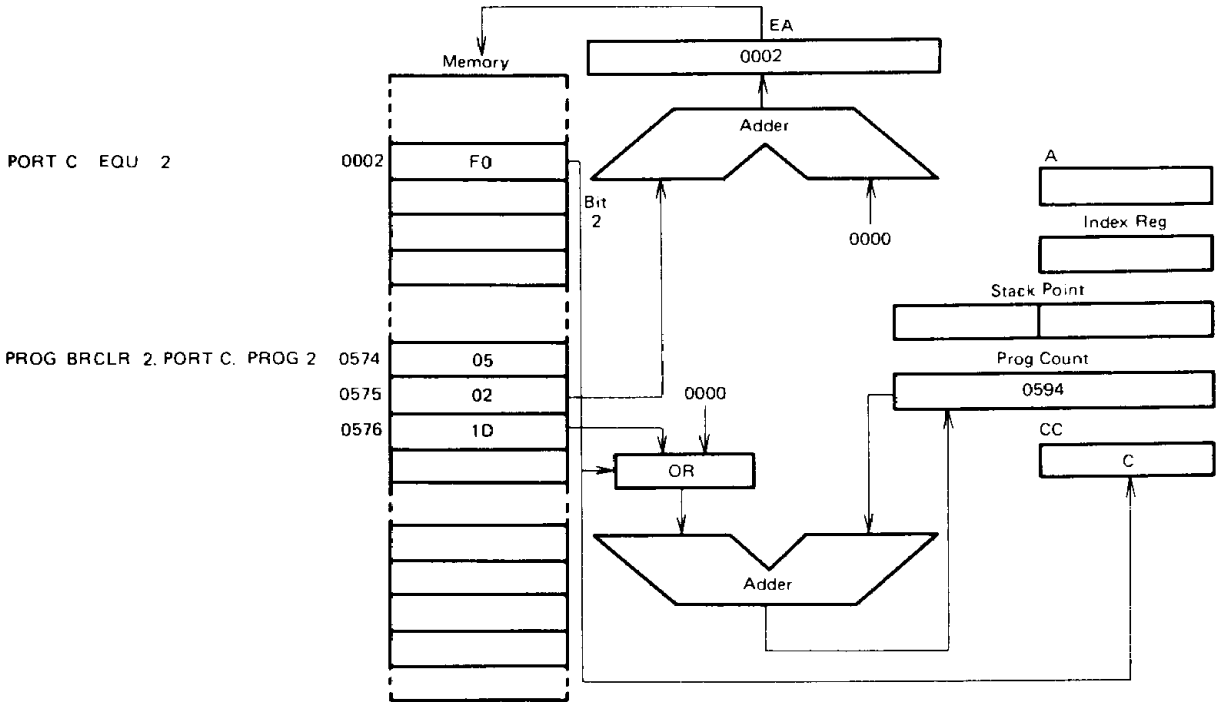


Figure 27 Bit Test and Branch Addressing Example

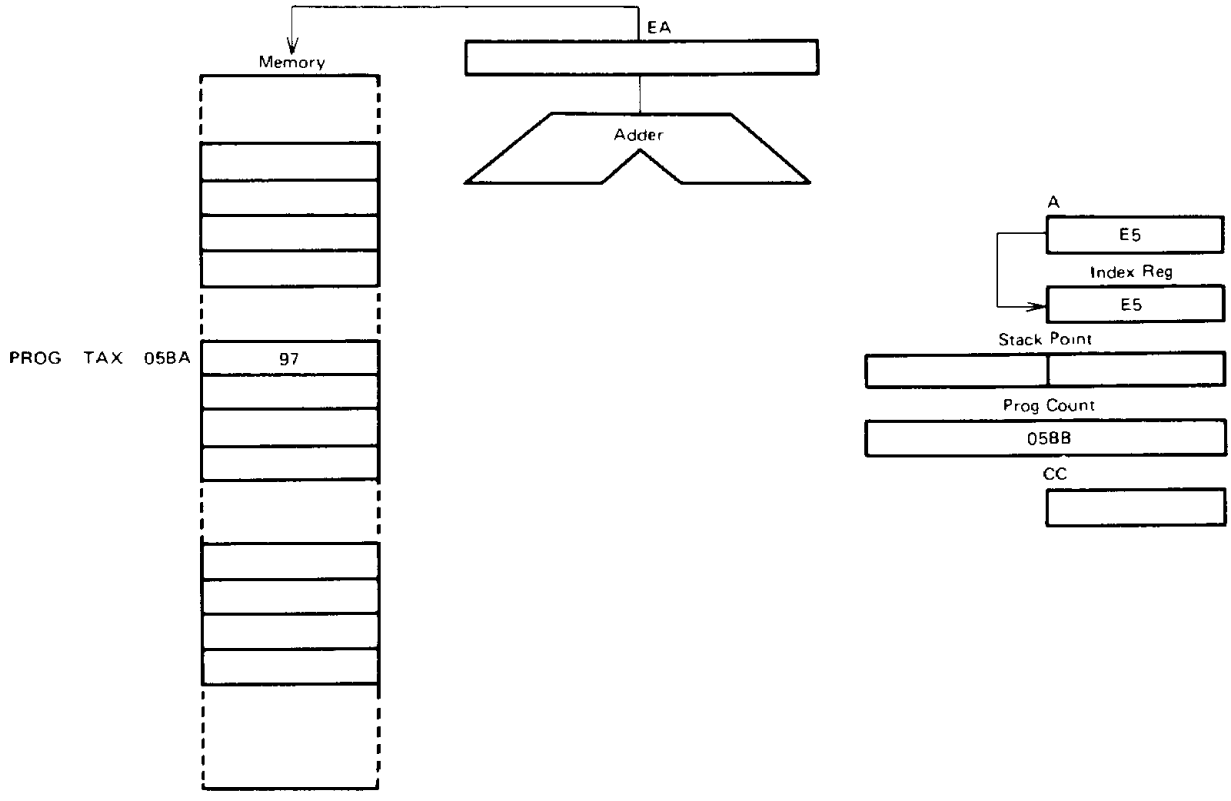


Figure 28 Implied Addressing Example

■ INSTRUCTION SET

The MCU has a set of 59 basic instructions. These instructions can be divided into five different types; register/memory, read/modify/write, branch, bit manipulation and control. Each instruction is briefly explained below. All of the instructions within 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 operand is obtained from memory by using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 8.

• 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 the memory or register. The TST instruction for test of negative or zero is an exception to the read/modify/write instructions since it does not perform the write. Refer to Table 9.

• Branch Instructions

The branch instructions cause a branch from a program when a certain condition is met. Refer to Table 10.

• Bit Manipulation Instructions

These instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test and branch operations. Refer to Table 11.

• Control Instructions

The control instructions control the MCU operations during program execution. Refer to Table 12.

• Alphabetical Listing

The complete instruction set is given in alphabetical order in Table 13.

• Opcode Map

Table 14 is an opcode map for the instructions used on the MCU.

Table 8 Register/Memory Instructions

Function	Mnemonic	Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	–	–	–	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–	–	–	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–	–	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–	–	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Symbols:
Op : Operation Abbreviation
: Instruction Statement

Hitachi America, Ltd. • Hitachi Plaza • 2000 Sierra Point Pkwy. • Brisbane, CA 94005-1819 • (415) 589-8300

Table 9 Read/Modify/Write Instructions

Function	Mnemonic	Addressing Modes														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Arithmetic Shift Left	ASL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Symbols:
Op: Operation Abbreviation
: Instruction Statement

Table 10 Branch Instructions

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IF Higher	BHI	22	2	4
Branch IF Lower or Same	BLS	23	2	4
Branch IF Carry Clear	BCC	24	2	4
(Branch IF Higher or Same)	(BHS)	24	2	4
Branch IF Carry Set	BCS	25	2	4
(Branch IF Lower)	(BLO)	25	2	4
Branch IF Not Equal	BNE	26	2	4
Branch IF Equal	BEQ	27	2	4
Branch IF Half Carry Clear	BHCC	28	2	4
Branch IF Half Carry Set	BHCS	29	2	4
Branch IF Plus	BPL	2A	2	4
Branch IF Minus	BMI	2B	2	4
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IF Interrupt Line is Low	BIL	2E	2	4
Branch IF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Symbols: Op: Operation Abbreviation #: Instruction Statement

Table 11 Bit Manipulation Instructions

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is set	BRSET n (n=0 ..... 7)	–	–	–	2•n	3	10
Branch IF Bit n is clear	BRCLR n (n=0 .....7)	–	–	–	01+2•n	3	10
Set Bit n	BSET n (n=0 ..... 7)	10+2•n	2	7	–	–	–
Clear bit n	BCLR n (n=0 ..... 7)	11+2•n	2	7	–	–	–

Symbols: Op: Operation Abbreviation #: Instruction Statement

Table 12 Control Instructions

Function	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

Symbols: Op: Operation Abbreviation #: Instruction Statement

Table 13 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		x	x	x		x	x	x			↕	•	↕	↕	↕
ADD		x	x	x		x	x	x			↕	•	↕	↕	↕
AND		x	x	x		x	x	x			•	•	↕	↕	•
ASL	x		x			x	x				•	•	↕	↕	↕
ASR	x		x			x	x				•	•	↕	↕	↕
BCC					x						•	•	•	•	•
BCLR									x		•	•	•	•	•
BCS					x						•	•	•	•	•
BEQ					x						•	•	•	•	•
BHCC					x						•	•	•	•	•
BHCS					x						•	•	•	•	•
BHI					x						•	•	•	•	•
BHS					x						•	•	•	•	•
BIH					x						•	•	•	•	•
BIL					x						•	•	•	•	•
BIT		x	x	x		x	x	x			•	•	↕	↕	•
BLO					x						•	•	•	•	•
BLS					x						•	•	•	•	•
BMC					x						•	•	•	•	•
BMI					x						•	•	•	•	•
BMS					x						•	•	•	•	•
BNE					x						•	•	•	•	•
BPL					x						•	•	•	•	•
BRA					x						•	•	•	•	•
BRN					x						•	•	•	•	•
BRCLR										x	•	•	•	•	↕
BRSET										x	•	•	•	•	↕
BSET									x		•	•	•	•	•
BSR					x						•	•	•	•	•
CLC	x										•	•	•	•	0
CLI	x										•	0	•	•	•
CLR	x		x			x	x				•	•	0	1	•
CMP		x	x	x		x	x	x			•	•	↕	↕	↕
COM	x		x			x	x				•	•	↕	↕	1
CPX		x	x	x		x	x	x			•	•	↕	↕	↕
DEC	x		x			x	x				•	•	↕	↕	•
EOR		x	x	x		x	x	x			•	•	↕	↕	•
INC	x		x			x	x				•	•	↕	↕	•
JMP			x	x		x	x	x			•	•	•	•	•
JSR			x	x		x	x	x			•	•	•	•	•
LDA		x	x	x		x	x	x			•	•	↕	↕	•
LDX		x	x	x		x	x	x			•	•	↕	↕	•

Condition Code Symbols
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry Borrow
- ↕ Test and Set if True, Cleared Otherwise
- • Not Affected

(to be continued)

Table 13 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
LSL	x		x			x	x				•	•	∧	∧	∧
LSR	x		x			x	x				•	•	0	∧	∧
NEG	x		x			x	x				•	•	∧	∧	∧
NOP	x										•	•	•	•	•
ORA		x	x	x		x	x	x			•	•	∧	∧	•
ROL	x		x			x	x				•	•	∧	∧	∧
ROR	x		x			x	x				•	•	∧	∧	∧
RSP	x										•	•	•	•	•
RTI	x										?	?	?	?	?
RTS	x										•	•	•	•	•
SBC		x	x	x		x	x	x			•	•	∧	∧	∧
SEC	x										•	•	•	•	1
SEI	x										•	1	•	•	•
STA			x	x		x	x	x			•	•	∧	∧	•
STX			x	x		x	x	x			•	•	∧	∧	•
SUB		x	x	x		x	x	x			•	•	∧	∧	∧
SWI	x										•	1	•	•	•
TAX	x										•	•	•	•	•
TST	x		x			x	x				•	•	∧	∧	•
TXA	x										•	•	•	•	•

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 14 Opcode 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	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	←HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	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					–	NOP	BSR*	JSR(+3)					D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/10	2/7	2/4	2/6	1/4	1/4	2/7	1/6	1/*	1/2	2/2	2/4	3/5	3/6	2/5	1/4	

[NOTE] 1. Undefined opcodes are marked with "–".
2. The number at the bottom of each column denotes the number of bytes and the number of cycles required (Bytes/Cycles). Mnemonics followed by a "*" require a different number of cycles as follows:

RTI	9
RTS	6
SWI	11
BSR	8

3. ( ) indicates that the number in parenthesis must be added to the cycle count for that instruction.

■ HD68P05W0 USED FOR HD6805W1

The HD6805W1 provides mask option of the internal oscillator and low voltage inhibit, while the HD68P05W0 provides only crystal option and without low voltage inhibit function.

The address from $0F7A to $0FF1 cannot be used for user program because the self test program of the HD6805W1 (on-chip ROM version) is located at these addresses.

In order to be pin compatible with the HD6805W1, the address of the HD68P05W0's ROM must be located at $0080 – $0FFF. Memory addresses $1000 to $1FFF should not be usable.

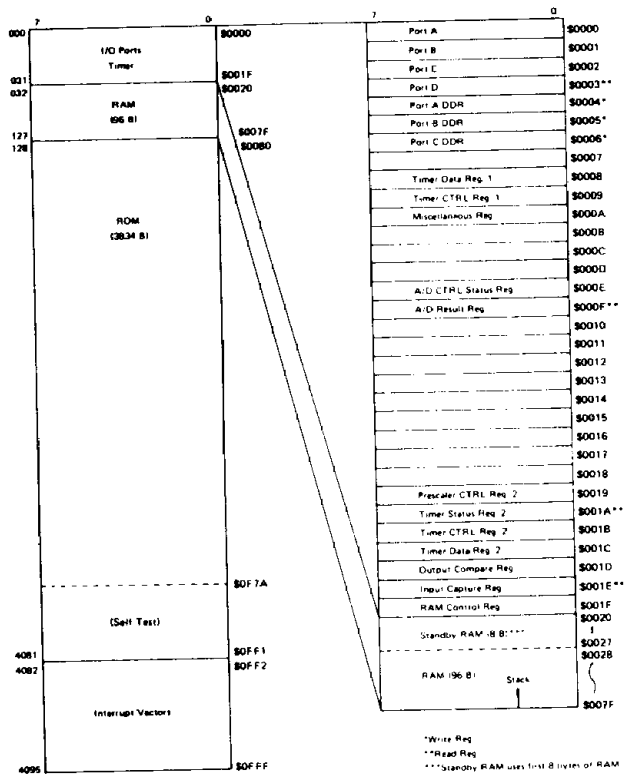


Figure 29 MCU Memory Structure (For 4k bytes)

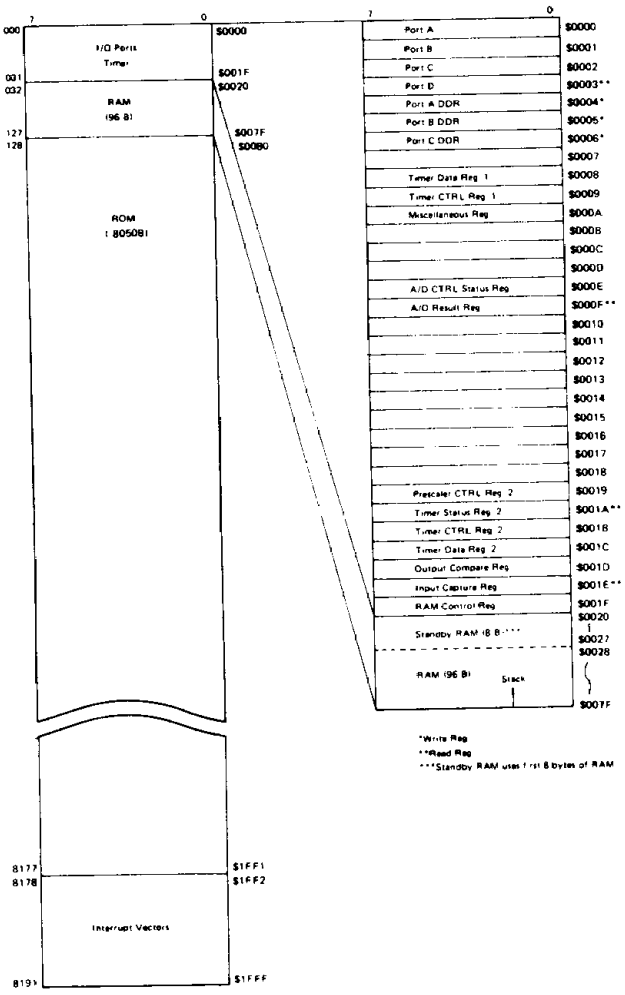


Figure 30 MCU Memory Structure (For 8k bytes)

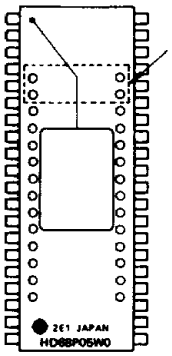
– CAUTION –
This 8k bytes type should not be used debugging on-chip ROM of the HD6805W1.

■ **PRECAUTION TO USE EPROM ON THE PACKAGE 8-BIT SINGLE-CHIP MICROCOMPUTER**

As this microcomputer takes a special packaging type with pin sockets on its surface, pay attention to the followings;

(1) Do not apply higher electro-static voltage or serge voltage etc. than maximum rating, or it may cause permanent damage to the device.

(2) There are 28 pin sockets on its surface. When using 32k EPROM (24 pins), let the index-side four pins open.



Let the index-side four pins open. When using 24 pin EPROM, match its index and insert it into lower 24 pin sockets.

(3) When assembling this LSI into user's system products as well as the mask ROM type 8-bit single-chip microcomputer, pay attention to the followings to keep the good ohmic contact between EPROM pins and pin sockets.

(a) When soldering on a printed circuit board, etc., keep its condition under 250°C within 10 seconds. Over-time/temperature may cause the bonding solder of socket pins to meet and the sockets may drop.

(b) Keep out detergent or coater from the pin sockets at aft-solder flux removal or board coating. The flux or coater may make pin socket contactivity worse.

(c) Avoid the permanent use of this LSI under the ever-vibratory place and system.

(d) Repeating insertion/removal of EPROMs may damage the contactivity of the pin sockets, so it is recommended to assemble new ones to your system products.

Ask our sales agent about anything unclear.