

# HD68P01S0, HD68P01V05, HD68P01V07, HD68P01M0 MCU (Microcomputer Unit)

The HD68P01 is an 8-bit single chip microcomputer unit (MCU) which significantly enhances the capabilities of the HMCS6800 family of parts. It can be used in production systems to allow for easy firmware changes with minimum delay or it can be used to emulate the HD6801 for software development. It includes 128 bytes of RAM, Serial Communications Interface (SCI), parallel I/O and a three function Programmable Timer on chip, and 2048 bytes, 4096 bytes or 8192 bytes of EPROM on package. It includes an upgrade HD6800 microprocessing unit (MPU) while retaining upward source and object code compatibility. Execution times of key instructions have been improved and several new instructions have been added including an unsigned 8 by 8 multiply with 16-bit result. The HD68P01 can function as a monolithic microcomputer or can be expanded to a 65k byte address space. It is TTL compatible and requires one +5 volt power supply. A summary of HD68P01 features includes:

## ■ FEATURES

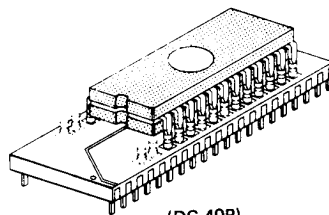
- Expanded HMCS6800 Instruction Set
- 8 x 8 Multiply Instruction
- Serial Communications Interface (SCI)
- Upward Source and Object Code Compatible with HD6800
- 16-bit Three-function Programmable Timer
- Applicable to All Type of EPROM
  - 2048 bytes; HN462716
  - 4096 bytes; HN462732 or HN462532
  - 8192 bytes; HN482764
- 128 Bytes of RAM (64 bytes Retainable on Powerdown)
- 29 Parallel I/O and Two Handshake Control Line
- Internal Clock Generator with Divide-by-Four Output
- Full TTL Compatibility
- Full Interrupt Capability
- Single-Chip or Expandable to 65k Bytes Address Space
- Bus compatible with HMCS6800 Family

## ■ TYPE OF PRODUCTS

Type No.	Bus Timing	EPROM Type No.
HD68P01S0	1 MHz	HN462716
HD68P01V05	1 MHz	HN462532
HD68P01V07	1 MHz	HN462732
HD68P01M0	1 MHz	HN482764

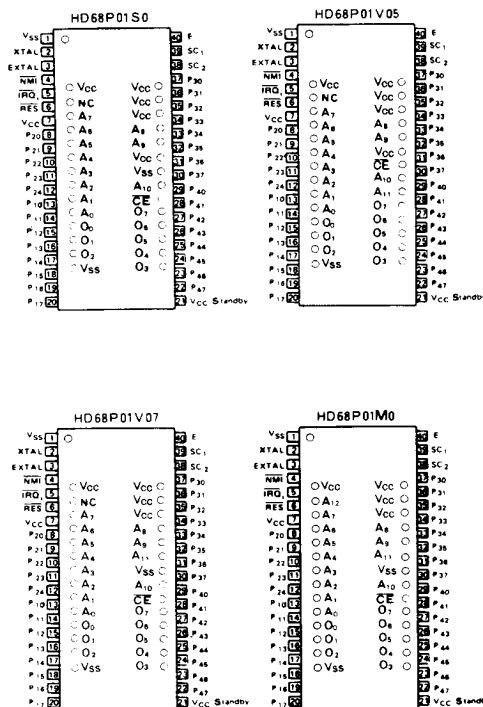
Note) EPROM is not included.

HD68P01S0, HD68P01V05, HD68P01V07, HD68P01M0

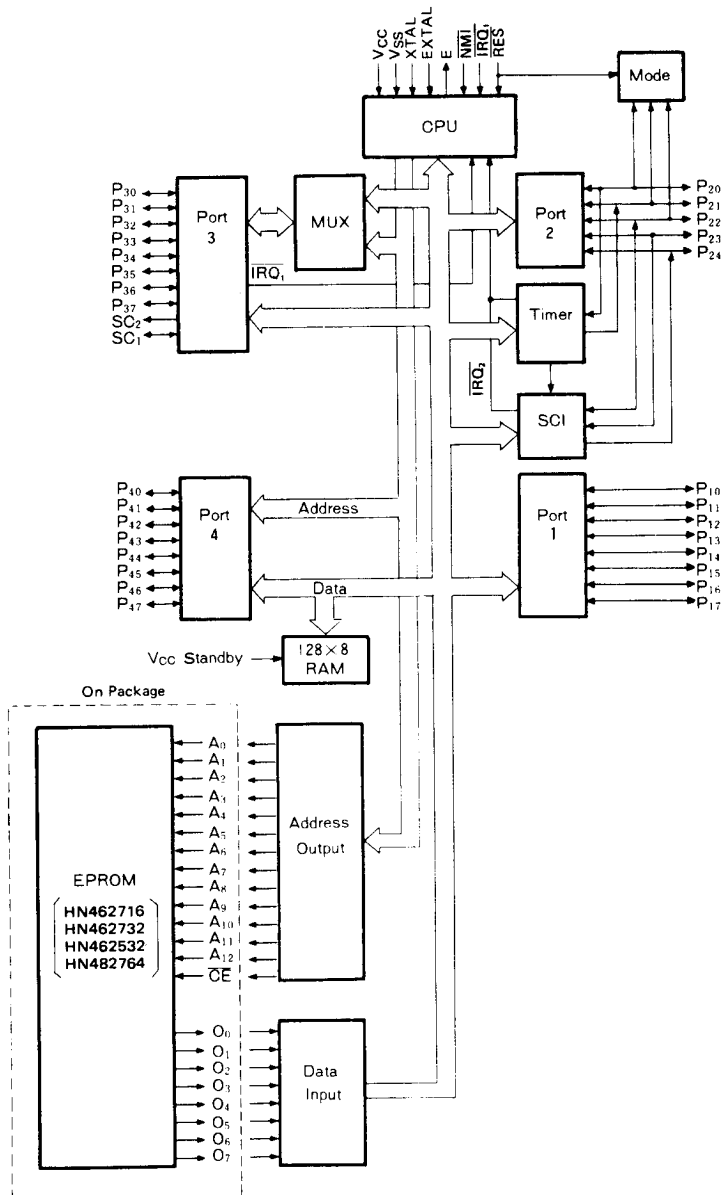


(DC-40P)

## ■ PIN ARRANGEMENT (Top View)



# BLOCK DIAGRAM



## ■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	$V_{CC}^*$	$-0.3 \sim +7.0$	V
Input Voltage	$V_{in}^*$	$-0.3 \sim +7.0$	V
Operating Temperature	$T_{opr}$	$0 \sim +70$	$^{\circ}\text{C}$
Storage Temperature	$T_{stg}$	$-55 \sim +150$	$^{\circ}\text{C}$

\* With respect to  $V_{SS}$  (SYSTEM GND)

[NOTE] Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

## ■ ELECTRICAL CHARACTERISTICS

● DC CHARACTERISTICS ( $V_{CC} = 5.0V \pm 5\%$ ,  $V_{SS} = 0V$ ,  $T_a = 0 \sim +70^{\circ}\text{C}$ , unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	RES		4.0	—	$V_{CC}$	V
	Other Inputs*		2.0	—	$V_{CC}$	
Input "Low" Voltage	All Inputs*		-0.3	—	0.8	V
Input Load Current	$P_{40} \sim P_{47}$	$V_{in} = 0 \sim 2.4V$	—	—	0.5	mA
	$SC_1$		—	—	0.8	
	EXTAL	$V_{in} = 0 \sim V_{CC}$	—	—	1.2	
Input Leakage Current	NMI, $IRQ_1$ , RES	$V_{in} = 0 \sim 5.25V$	—	—	2.5	$\mu\text{A}$
Three State (Offset) Leakage Current	$P_{10} \sim P_{17}$ , $P_{30} \sim P_{37}$	$V_{in} = 0.5 \sim 2.4V$	—	—	10	$\mu\text{A}$
	$P_{20} \sim P_{24}$		—	—	100	
Output "High" Voltage	$P_{30} \sim P_{37}$	$I_{LOAD} = -205 \mu\text{A}$	2.4	—	—	V
	$P_{40} \sim P_{47}$ , E, $SC_1$ , $SC_2$	$I_{LOAD} = -145 \mu\text{A}$	2.4	—	—	
	Other Outputs	$I_{LOAD} = -100 \mu\text{A}$	2.4	—	—	
Output "Low" Voltage	All Outputs	$I_{LOAD} = 1.6 \text{ mA}$	—	—	0.5	V
Darlington Drive Current	$P_{10} \sim P_{17}$	$-I_{OH}$ $V_{out} = 1.5V$	1.0	—	10.0	mA
Power Dissipation		$P_D$	—	—	1200	mW
Input Capacitance	$P_{30} \sim P_{37}$ , $P_{40} \sim P_{47}$ , $SC_1$	$V_{in} = 0V$ , $T_a = 25^{\circ}\text{C}$ , $f = 1.0 \text{ MHz}$	—	—	12.5	pF
	Other Inputs		—	—	10.0	
$V_{CC}$ Standby	Powerdown	$V_{SBB}$	4.0	—	5.25	V
	Operating	$V_{SB}$	4.75	—	5.25	
Standby Current	Powerdown	$I_{SBB}$ $V_{SBB} = 4.0V$	—	—	8.0	mA

\*Except Mode Programming Levels: See Figure 8.

# AC CHARACTERISTICS

**BUS TIMING** ( $V_{CC} = 5.0V \pm 5\%$ ,  $V_{SS} = 0V$ ,  $T_a = 0 \sim +70^\circ C$ , unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Cycle Time	$t_{cyc}$	Fig. 1 Fig. 2	1	—	10	$\mu s$
Address Strobe Pulse Width "High"	$PW_{ASH}$		200	—	—	ns
Address Strobe Rise Time	$t_{ASr}$		5	—	50	ns
Address Strobe Fall Time	$t_{ASf}$		5	—	50	ns
Address Strobe Delay Time	$t_{ASD}$		60	—	—	ns
Enable Rise Time	$t_{Er}$		5	—	50	ns
Enable Fall Time	$t_{Ef}$		5	—	50	ns
Enable Pulse Width "High" Time	$PW_{EH}$		450	—	—	ns
Enable Pulse Width "Low" Time	$PW_{EL}$		450	—	—	ns
Address Strobe to Enable Delay Time	$t_{ASED}$		60	—	—	ns
Address Delay Time	$t_{AD}$		—	—	260	ns
Address Delay Time for Latch ( $f=1.0MHz$ )	$t_{ADL}$		—	—	270	ns
Data Set-up Write Time	$t_{DSW}$		225	—	—	ns
Data Set-up Read Time	$t_{DSR}$		80	—	—	ns
Data Hold Time	Read		10	—	—	ns
	Write		20	—	—	
Address Set-up Time for Latch	$t_{ASL}$		60	—	—	ns
Address Hold Time for Latch	$t_{AHL}$		20	—	—	ns
Address Hold Time	$t_{AH}$		20	—	—	ns
Peripheral Read Access Time	Non-Multiplexed Bus ( $t_{ACCN}$ ) Multiplexed Bus ( $t_{ACCM}$ )		—	—	(610) (600)	ns
Oscillator stabilization Time	$t_{RC}$	Fig. 11	100	—	—	ms
Processor Control Set-up Time	$t_{PCS}$	Fig. 12	200	—	—	ns

**PERIPHERAL PORT TIMING** ( $V_{CC} = 5.0V \pm 5\%$ ,  $V_{SS} = 0V$ ,  $T_a = 0 \sim +70^\circ C$ , unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Peripheral Data Setup Time	Port 1, 2, 3, 4 $t_{PDSU}$	Fig. 3	200	—	—	ns
Peripheral Data Hold Time	Port 1, 2, 3, 4 $t_{PDH}$	Fig. 3	200	—	—	ns
Delay Time, Enable Positive Transition to $\bar{OS3}$ Negative Transition	$t_{OSD1}$	Fig. 5	—	—	350	ns
Delay Time, Enable Positive Transition to $\bar{OS3}$ Positive Transition	$t_{OSD2}$	Fig. 5	—	—	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	Port 1, 2*, 3, 4 $t_{PWD}$	Fig. 4	—	—	400	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	Port 2**, 4 $t_{CMOS}$	Fig. 4	—	—	2.0	$\mu s$
Input Strobe Pulse Width	$t_{PWIS}$	Fig. 6	200	—	—	ns
Input Data Hold Time	port 3 $t_{IH}$	Fig. 6	50	—	—	ns
Input Data Set-up Time	Port 3 $t_{IS}$	Fig. 6	20	—	—	ns

\*Except  $P_{21}$

\*\*10k $\Omega$  pull up register required for Port 2





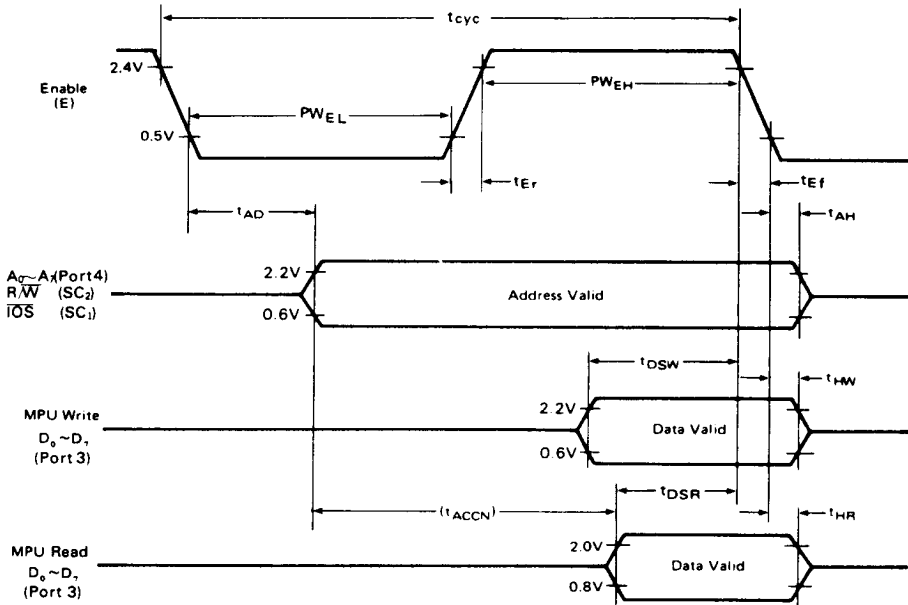
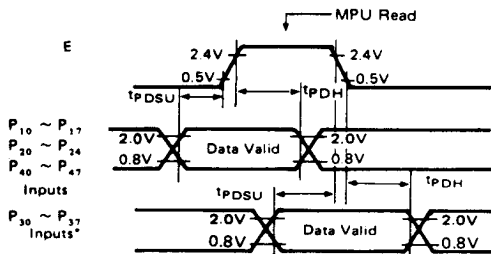
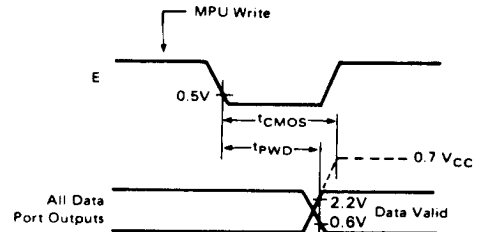


Figure 2 Expanded Non-Multiplexed Bus Timing



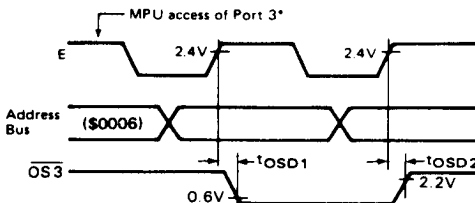
\*Port 3 Non-Latched Operation (LATCH ENABLE = 0)

Figure 3 Data Set-up and Hold Times (MPU Read)



(NOTE) 1. 10 kΩ Pullup resistor required for Port 2 to reach 0.7 V<sub>CC</sub>  
2. Not applicable to P<sub>31</sub>  
3. Port 4 cannot be pulled above V<sub>CC</sub>

Figure 4 Port Data Delay Timing (MPU Write)



\*Access matches Output Strobe Select (OSS = 0, a read; OSS = 1, a write)

Figure 5 Port 3 Output Strobe Timing (Single Chip Mode)

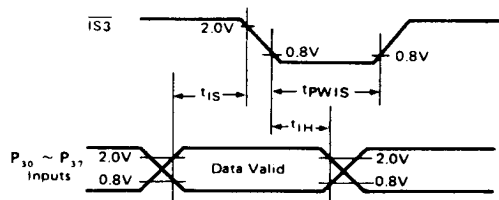


Figure 6 Port 3 Latch Timing (Single Chip Mode)

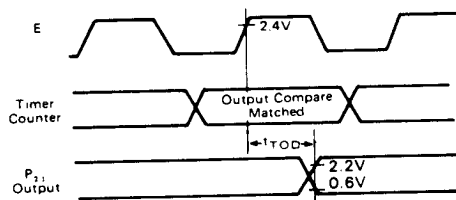


Figure 7 Timer Output Timing

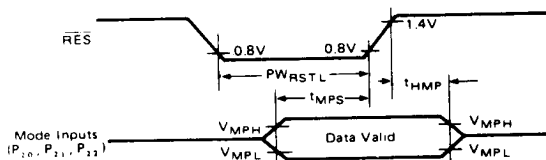
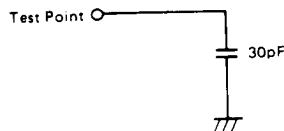
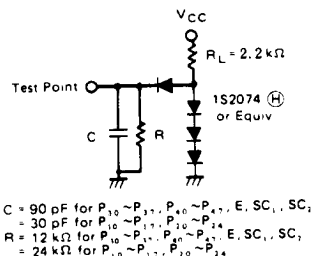


Figure 8 Mode Programming Timing



(a) CMOS Load



(b) TTL Load

Figure 9 Bus Timing Test Loads

## INTRODUCTION

The HD68P01 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into eight different operating modes. The operating mode controls the configuration of 18 of the MCU's 40 pins, available on-chip resources, memory map, location (internal or external) of interrupt vectors, and type of external bus. The configuration of the remaining 22 pins is not dependent on the operating mode.

Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a Data Register and a write-only Data Direction Register. The Data Direction Register is used to define whether corresponding bits in the Data Register are configured as an input (clear) or output (set).

The term "port", by itself, refers to all of its associated hardware. When the port is used as a "data port" or "I/O port", it is controlled by its Data Direction Register and the programmer has direct access to its pins using the port's Data Register. Port pins are labeled as  $P_{ij}$  where  $i$  identifies one of four ports and  $j$  indicates the particular bit.

The Microprocessor Unit (MPU) is an enhanced HD6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the HD6800. The programming model is depicted in Figure 10 where Accumulator D is a concatenation of Accumulators A and B. A list of new operations added to the HMCS6800 instruction set are shown in Table 8.

The basic difference between the HD6801 and the HD68P01 is that the HD6801 has an on-chip ROM while the HD68P01 has

an on-package EPROM. The HD68P01 is pin and code compatible with the HD6801 and can be used to emulate the HD6801, allowing easy software development using the on-package EPROM. Software developed using the HD68P01 can then be masked into the HD6801 ROM.

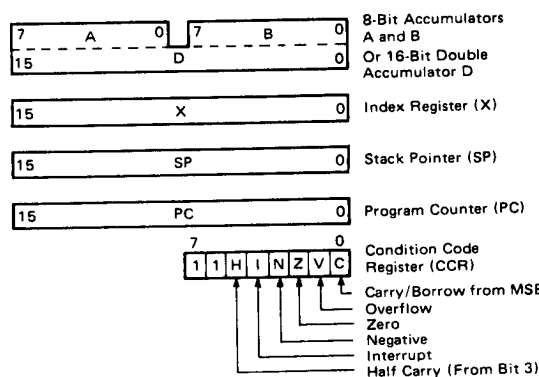


Figure 10 HD68P01 Programming Model

## ■ INTERRUPTS

The MCU supports two types of interrupt requests: maskable and non-maskable. A Non-Maskable Interrupt (NMI) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the Condition Code Register's I-bit and by individual enable bits. The I-bit controls all maskable interrupts. Of the maskable interrupts, there are two types:  $IRQ_1$  and  $IRQ_2$ . The Programmable Timer and Serial Communications Interface use an internal  $IRQ_2$  interrupt line, as shown in BLOCK DIAGRAM. External devices (and IS3) use  $IRQ_1$ . An  $IRQ_1$  interrupt is serviced before  $IRQ_2$  if both are pending.

All  $IRQ_2$  interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order where each is vectored to a separate location. All MCU interrupt vector locations are shown in Table 1.

The Interrupt flowchart is depicted in Figure 13 and is common to every MCU interrupt excluding Reset. The Program Counter, Index Register, A Accumulator, B Accumulator, and Condition Code Register are pushed to the stack. The I-bit is

set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the Program Counter and instruction execution is resumed. Interrupt and RES timing is illustrated in Figure 11 and 12.

Table 1 MCU Interrupt Vector Locations

MSB	LSB	Interrupt
FFFE	FFFF	RES
FFFC	FFFD	NMI
FFFA	FFFB	Software Interrupt (SWI)
FFF8	FFF9	$IRQ_1$ (or IS3)
FFF6	FFF7	ICF (Input Capture)
FFF4	FFF5	OCF (Output Compare)
FFF2	FFF3	TOF (Timer Overflow)
FFF0	FFF1	SCI (RDRF + ORFE + TDRE)

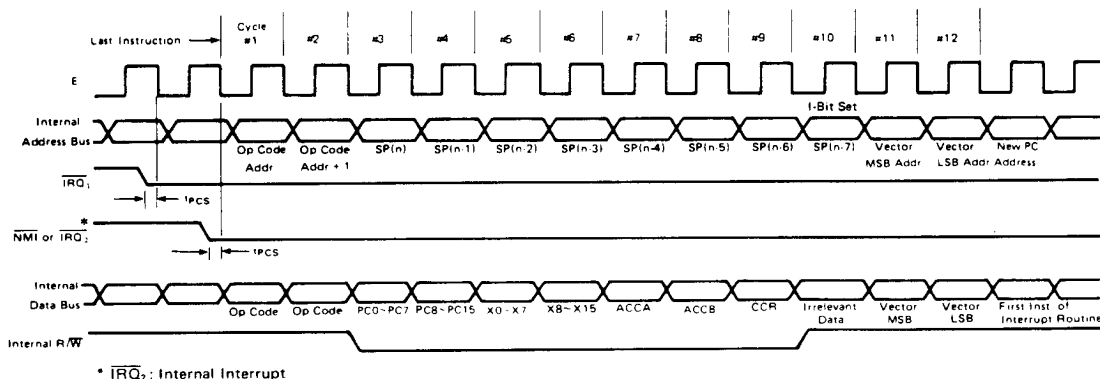


Figure 11 Interrupt Sequence

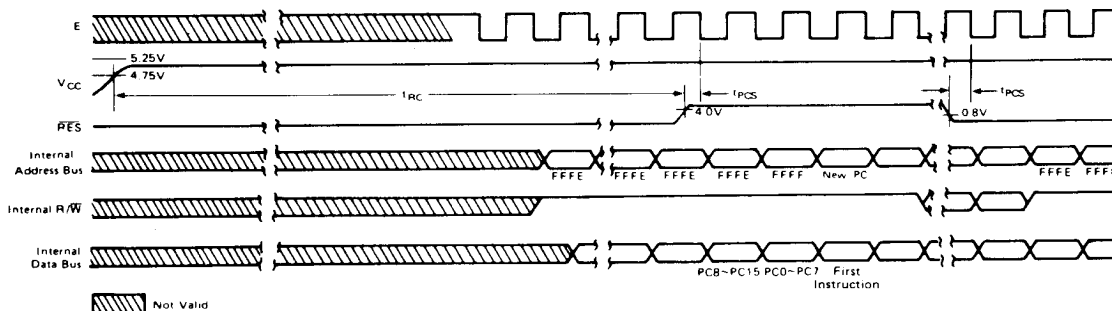


Figure 12 Reset Timing



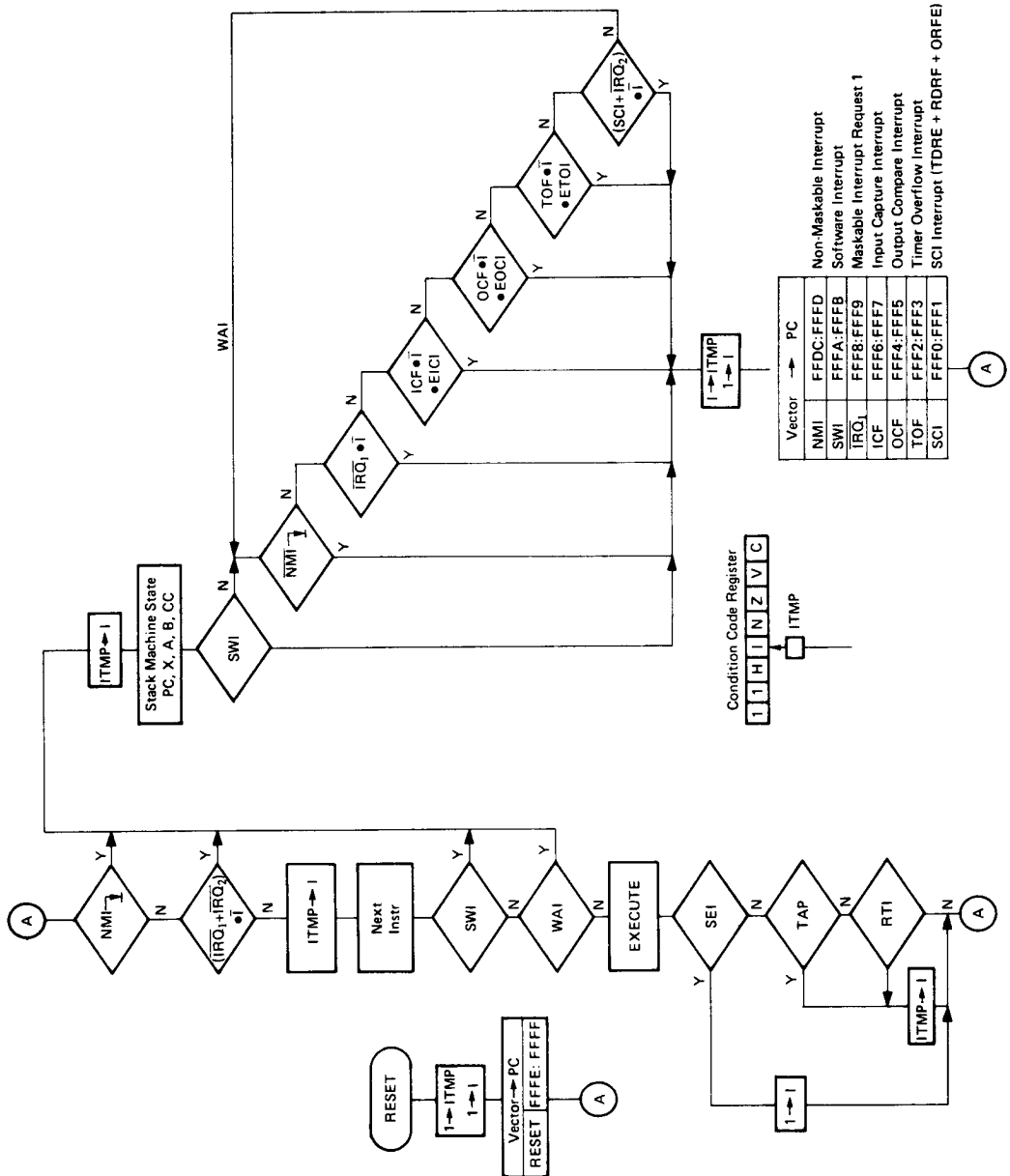


Figure 13 Interrupt Flowchart

## ■ FUNCTIONAL PIN DESCRIPTIONS

### ● $V_{CC}$ and $V_{SS}$

$V_{CC}$  and  $V_{SS}$  provide power to a large portion of the MCU. The power supply should provide +5 volts ( $\pm 5\%$ ) to  $V_{CC}$ , and  $V_{SS}$  should be tied to ground. Total power dissipation (including  $V_{CC}$  Standby), will not exceed  $P_D$  milliwatts.

### ● $V_{CC}$ Standby

$V_{CC}$  Standby provides power to the standby portion (\$80 through \$BF) 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 +5 volts ( $\pm 5\%$ ) and must reach  $V_{SB}$  volts before  $\overline{RES}$  reaches 4.0 volts. 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 normal operation. A diode must be used between them to prevent supplying power to  $V_{CC}$  during powerdown operation.  $V_{CC}$  Standby should be tied to either ground or  $V_{CC}$  in Mode 3.

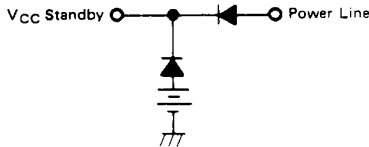


Figure 14 Battery Backup for  $V_{CC}$  Standby

### ● RAM Control Register (\$14)

The RAM Control Register includes two bits which can be used to control RAM accesses and determine the adequacy of the standby power source during powerdown operation. It is intended that RAME be cleared and STBY PWR be set as part of a powerdown procedure.

RAM Control Register

7	6	5	4	3	2	1	0
STBY PWR	RAME	X	X	X	X	X	X

Bit 0~5 Not Used

Bit 6 RAME

**RAM Enable.** This Read/Write bit can be used to remove the entire RAM from the internal memory map. RAME is set (enabled) during Reset provided standby power is available on the positive edge of  $\overline{RES}$ . If RAME is clear, any access to a RAM address is external. If RAME is set and not in Mode 3, the RAM is included in the internal map.

Bit 7 STBY PWR

**Standby Power.** This bit is a Read/Write status bit which is cleared whenever  $V_{CC}$  Standby decreases below  $V_{SBB}$  (min). It can be set only by software and is not affected by  $\overline{RES}$ .

### ● XTAL and EXTAL

These two input pins interface either a crystal or TTL com-

patible clock to the MCU's internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58 MHz Color Burst TV crystals. A 22 pF capacitor is required from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL may be driven with an external TTL compatible clock with a duty cycle of 50% ( $\pm 10\%$ ) with XTAL connected to ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator or a ceramic resonator operated in parallel resonance mode in the frequency range specified for 3.2 ~ 4 MHz. The crystal should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. The MCU is compatible with most commercially available crystals and ceramic resonators and nominal crystal parameters are shown in Figure 15.

### ● $\overline{RES}$

This input is used to reset the MCU's internal state and provide an orderly startup procedure. During powerup,  $\overline{RES}$  must be held below 0.8 volts: (1) at least  $t_{RC}$  after  $V_{CC}$  reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until  $V_{CC}$  Standby reaches 4.75 volts.  $\overline{RES}$  must be held low at least three E-cycles if asserted during powerup operation.

When a "High" level is detected, the MCU does the following:

- 1) All the higher order address lines will be forced "High".
- 2) I/O Port 2 bits, 2, 1, and 0 are latched into programmed control bits PC2, PC1 and PC0.
- 3) The last two (\$FFE, \$FFF) locations in memory will be used to load the program addressed by the program counter.
- 4) The interrupt mask bit is set; must be cleared before the MPU can recognize maskable interrupts.

### ● E (Enable)

This is an output clock used primarily for bus synchronization. It is TTL compatible and is the slightly skewed divide-by-four result of the MCU input frequency. It will drive one Schottky TTL load and 90 pF, and all data given in cycles is referenced to this clock unless otherwise noted.

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

An  $\overline{NMI}$  negative edge request an MCU interrupt sequence, but the current instruction will be completed before it responds to the request. The MCU will then begin an interrupt sequence. Finally, a vector is fetched from \$FFFC and \$FFFD, transferred to the Program Counter and instruction execution resumes.  $\overline{NMI}$  typically requires a 3.3 k $\Omega$  (nominal) resistor to  $V_{CC}$ . There is no internal  $\overline{NMI}$  pullup resistor.  $\overline{NMI}$  must be held low for at least one E-cycle to be recognized under all conditions.

### ● $\overline{IRQ_1}$ (Maskable Interrupt Request 1)

$\overline{IRQ_1}$  is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the interrupt mask bit (I-bit) in the Condition Code Register is clear, the MCU will begin an interrupt sequence. Finally, a vector is fetched from \$FFF8 and \$FFF9, transferred to the Program Counter, and instruction execution is resumed.

$\overline{IRQ_1}$  typically requires an external 3.3 k $\Omega$  (nominal) resistor to  $V_{CC}$  for wire-OR application.  $\overline{IRQ_1}$  has no internal pullup resistor.

### • SC<sub>1</sub> and SC<sub>2</sub> (Strobe Control 1 and 2)

The function of SC<sub>1</sub> and SC<sub>2</sub> depends on the operating mode. SC<sub>1</sub> is configured as an output in all modes except single chip mode, whereas SC<sub>2</sub> is always an output. SC<sub>1</sub> and SC<sub>2</sub> can drive one Schottky load and 90 pF.

### SC<sub>1</sub> and SC<sub>2</sub> in Single Chip Mode

In Single Chip Modes, SC<sub>1</sub> and SC<sub>2</sub> are configured as an input and output, respectively, and both function as Port 3 control lines. SC<sub>1</sub> functions as IS<sub>3</sub> and can be used to indicate that Port 3 input data is ready or output data has been accepted. Three options associated with IS<sub>3</sub> are controlled by Port 3's Control and Status Register and are discussed in Port 3's description. If unused, IS<sub>3</sub> can remain unconnected.

SC<sub>2</sub> is configured as OS<sub>3</sub> and can be used to strobe output data or acknowledge input data. It is controlled by Output Strobe Select (OSS) in Port 3's Control and Status Register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to Port 3's Data Register. OS<sub>3</sub> timing is shown in Figure 5.

### SC<sub>1</sub> and SC<sub>2</sub> in Expanded Non-Multiplexed Mode

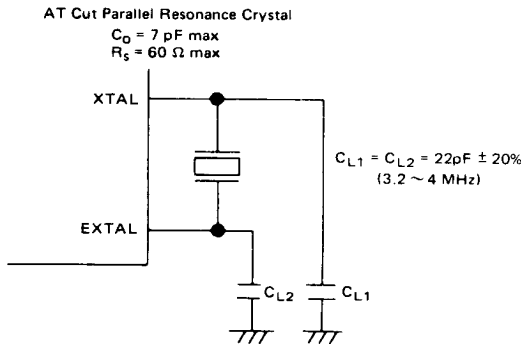
In the Expanded Non-Multiplexed Mode, both SC<sub>1</sub> and SC<sub>2</sub> are configured as outputs. SC<sub>1</sub> functions as Input/Output Select (IOS) and is asserted only when \$0100 through \$01FF is sensed on the internal address bus.

SC<sub>2</sub> is configured as Read/Write and is used to control the direction of data bus transfers. An MPU read is enabled when Read/Write and E are high.

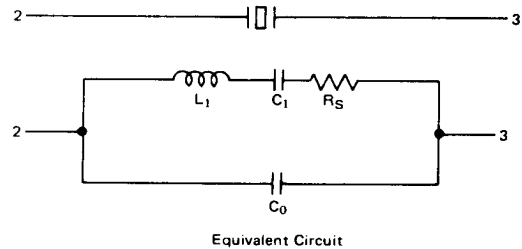
### SC<sub>1</sub> and SC<sub>2</sub> in Expanded Multiplexed Mode

In the Expanded Multiplexed Modes, both SC<sub>1</sub> and SC<sub>2</sub> are configured as outputs. SC<sub>1</sub> functions as Address Strobe and can be used to demultiplex the eight least significant addresses and the data bus. A latch controlled by Address Strobe captures address on the negative edge, as shown in Figure 20.

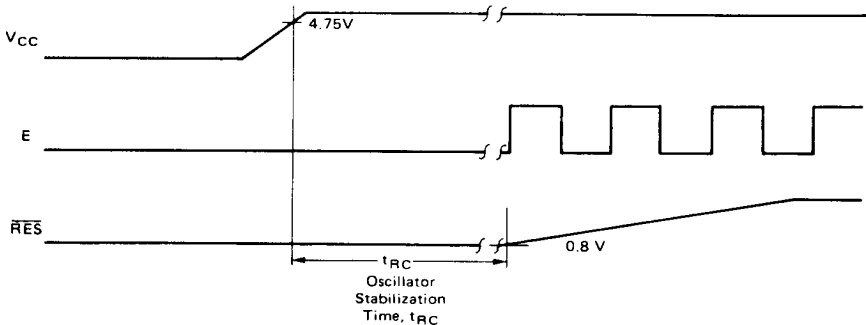
SC<sub>2</sub> is configured as Read/Write and is used to control the direction of data bus transfers. An MPU read is enabled when Read/Write and E are high.



(a) Nominal Recommended Crystal Parameters



Equivalent Circuit



(b) Oscillator Stabilization Time ( $t_{RC}$ )

Figure 15 Oscillator Characteristics

## ■ PORTS

There are four I/O ports on the MCU; three 8-bit ports and one 5-bit port. There are two control lines associated with one of the 8-bit ports. Each port has an associated write only Data Direction Register which allows each I/O line to be programmed to act as an input or an output. A "1" in the corresponding Data Direction Register bit will cause that I/O line to be an output. A "0" in the corresponding Data Direction Register bit will cause the I/O line to be an input. There are four ports: Port 1, Port 2, Port 3, and Port 4. Their addresses and the addresses of their Data Direction registers are given in Table 2.

Table 2 Port and Data Direction Register Addresses

Ports	Port Address	Data Direction Register Address
I/O Port 1	\$0002	\$0000
I/O Port 2	\$0003	\$0001
I/O Port 3	\$0006	\$0004
I/O Port 4	\$0007	\$0005

### ● P10~P17 (Port 1)

Port 1 is a mode independent 8-bit I/O port where each line is an input or output as defined by its Data Direction Register. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30 pF, Darlington transistors, or CMOS devices using external pullup resistors. It is configured as a data input port by RES. Unused lines can remain unconnected.

### ● P20~P24 (Port 2)

Port 2 is a mode independent 5-bit I/O port where each line is configured by its Data Direction Register. During RES, all lines are configured as inputs. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30 pF or CMOS devices using external pullup resistors. P<sub>20</sub>, P<sub>21</sub> and P<sub>22</sub> must always be connected to provide the operating mode. If lines P<sub>23</sub> and P<sub>24</sub> are unused, they can remain unconnected.

P<sub>20</sub>, P<sub>21</sub>, and P<sub>22</sub> provide the operating mode which is latched into the Program Control Register on the positive edge of RES. The mode may be read from Port 2 Data Register as shown where PC2 is latched from pin 10.

Port 2 also provides an interface for the Serial Communications Interface and Timer. Bit 1, if configured as an output, is dedicated to the timer's Output Compare function and cannot be used to provide output from Port 2 Data Register.

Port 2 Data Register

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	\$0003

### ● P30~P37 (Port 3)

Port 3 can be configured as an I/O port, a bidirectional 8-bit data bus, or a multiplexed address/data bus depending on the operating mode. The TTL compatible three-state output buffers can drive one Schottky TTL load and 90 pF. Unused lines can remain unconnected.

#### Port 3 in Single-Chip Mode

Port 3 is an 8-bit I/O port in Single-Chip Mode where each line is configured by its Data Direction Register. There are also

two lines,  $\overline{\text{IS3}}$  and  $\overline{\text{OS3}}$ , which can be used to control Port 3 data transfers.

Three Port 3 options are controlled by the Port 3 Control and Status Register and available only in Single-Chip Mode: (1) Port 3 input data can be latched using  $\overline{\text{IS3}}$  as a control signal, (2)  $\overline{\text{OS3}}$  can be generated by either an MPU read or write to Port 3's Data Register, and (3) an  $\overline{\text{IRQ}}_1$  interrupt can be enabled by an  $\overline{\text{IS3}}$  negative edge. Port 3 latch timing is shown in Figure 6.

Port 3 Control and Status Register

7	6	5	4	3	2	1	0	
$\overline{\text{IS3}}$ Flag	$\overline{\text{IS3}}$ $\overline{\text{IRQ}}_1$ Enable	X	OSS	Latch Enable	X	X	X	\$000F

Bit 0~2

Not used.

Bit 3

LATCH ENABLE. This bit controls the input latch for Port 3. If set, input data is latched by an  $\overline{\text{IS3}}$  negative edge. The latch is transparent after a read of Port 3's Data Register. LATCH ENABLE is cleared by RES.

Bit 4

OSS (Output Strobe Select). This bit determines whether  $\overline{\text{OS3}}$  will be generated by a read or write of Port 3's Data Register. When clear, the strobe is generated by a read; when set, it is generated by a write. OSS is cleared by RES.

Bit 5

Not used.

Bit 6

$\overline{\text{IS3}}$   $\overline{\text{IRQ}}_1$  ENABLE. When set, an  $\overline{\text{IRQ}}_1$  interrupt will be enabled whenever  $\overline{\text{IS3}}$  FLAG is set; when clear, the interrupt is inhibited. This bit is cleared by RES.

Bit 7

$\overline{\text{IS3}}$  FLAG. This read-only status bit is set by an  $\overline{\text{IS3}}$  negative edge. It is cleared by a read of the Port 3 Control and Status Register (with  $\overline{\text{IS3}}$  FLAG set) followed by a read or write to Port 3's Data Register or by RES.

#### Port 3 in Expanded Non-Multiplexed Mode

Port 3 is configured as a bidirectional data bus ( $D_0 \sim D_7$ ) in the Expanded Non-Multiplexed Mode. The direction of data transfers is controlled by Read/Write ( $\text{SC}_2$ ) and clocked by E (Enable).

#### Port 3 in Expanded Multiplexed Mode

Port 3 is configured as a time multiplexed address ( $A_0 \sim A_7$ ) and data bus ( $D_0 \sim D_7$ ) in Expanded Multiplexed Mode where Address Strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high impedance state between valid address and data to prevent potential bus conflicts.

### ● P40~P47 (Port 4)

Port 4 is configured as an 8-bit I/O port, address outputs, or data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90 pF and is the only port with internal pullup resistors. Unused lines can remain unconnected.

#### Port 4 in Single Chip Mode

In Single Chip Mode, Port 4 functions as an 8-bit I/O port where each line is configured by its Data Direction Register.

Internal pullup resistors allow the port to directly interface with CMOS at 5 volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

#### Port 4 in Expanded Non-Multiplexed Mode

Port 4 is configured from RES as an 8-bit input port where its Data Direction Register can be written to provide any or all of address lines, A<sub>0</sub> to A<sub>7</sub>. Internal pullup resistors are intended to pull the lines high until its Data Direction Register is configured.

#### Port 4 in Expanded Multiplexed Mode

In all Expanded Multiplexed modes except Mode 6, Port 4 functions as half of the address bus and provides A<sub>8</sub> to A<sub>15</sub>. In Mode 6, the port is configured from RES as an 8-bit parallel input port where its Data Direction Register can be written to provide any or all of address lines, A<sub>8</sub> to A<sub>15</sub>. Internal pullup resistors are intended to pull the lines high until its Data Direction Register is configured where bit 0 controls A<sub>8</sub>.

### ■ OPERATING MODES

The MCU provides eight different operating modes which are selectable by hardware programming and referred to as Mode 0 through Mode 7. The operating mode controls the memory map, configuration of Port 3, Port 4, SC<sub>1</sub>, SC<sub>2</sub>, and the physical location of interrupt vectors.

#### ● Fundamental Modes

The MCU's eight modes can be grouped into three fundamental modes which refer to the type of bus it supports: Single Chip, Expanded Non-Multiplexed, and Expanded Multiplexed. Single chip modes include 4 and 7, Expanded Non-Multiplexed is Mode 5 and the remaining five are Expanded Multiplexed modes. Table 3 summarizes the characteristics of the operating modes.

#### Single Chip Modes (4, 7)

In Single-Chip Mode, the MCU's four ports are configured as parallel input/output data ports, as shown in Figure 16. The MCU functions as a monolithic microcomputer in these two modes without external address or data buses. A maximum of 29 I/O lines and two Port 3 control lines are provided. In addition to other peripherals, another MCU can be interfaced to Port 3 in a loosely coupled dual processor configuration, as shown in Figure 17.

In Single-Chip Test Mode (4), the RAM responds to \$X×80 through \$X×FF and the ROM is removed from the internal address map. A test program must first be loaded into the RAM using modes 0, 1, 2, or 6. If the MCU is Reset and then programmed into Mode 4, execution will begin at \$X×FE: \$X×FF. Mode 5 can be irreversibly entered from Mode 4 without going through Reset by setting bit 5 of Port 2's Data Register. This mode is used primarily to test Ports 3 and 4 in the Single-Chip and Non-Multiplexed Modes.

#### Expanded Non-Multiplexed Mode (5)

A modest amount of external memory space is provided in the Expanded Non-Multiplexed Mode while retaining significant on-chip resources. Port 3 functions as an 8-bit bidirectional data bus and Port 4 is configured as an input data port. Any combination of the eight least-significant address lines may be obtained by writing to Port 4's Data Direction Register. Stated alternatively, any combination of A<sub>0</sub> to A<sub>7</sub> may be provided while retaining the remainder as input data lines. Internal pull-

up resistors are intended to pull Port 4's lines high until it is configured.

Figure 18 illustrates a typical system configuration in the Expanded Non-Multiplexed Mode. The MCU interfaces directly with HMCS6800 family parts and can access 256 bytes of external address space at \$100 through \$1FF. IÖS provides an address decode of external memory (\$100–\$1FF) and can be used similarly to an address or chip select line.

Table 3 Summary of HD6800 Operating Modes

<b>Common to all Modes:</b>	
Reserved Register Area	
Port 1	
Port 2	
Programmable Timer	
Serial Communication Interface	
<b>Single Chip Mode 7</b>	
128 bytes of RAM; 2048 bytes of ROM	
Port 3 is a parallel I/O port with two control lines	
Port 4 is a parallel I/O port	
SC <sub>1</sub> is Input Strobe 3 (IS3)	
SC <sub>2</sub> is Output Strobe 3 (OS3)	
<b>Expanded Non-Multiplexed Mode 5</b>	
128 bytes of RAM; 2048 bytes of ROM	
256 bytes of external memory space	
Port 3 is an 8-bit data bus	
Port 4 is an input port/address bus	
SC <sub>1</sub> is Input/Output Select (IÖS)	
SC <sub>2</sub> is read/write (R/W)	
<b>Expanded Multiplexed Modes 1, 2, 3, 6</b>	
Four memory space options (65k address space):	
(1) No internal RAM or ROM (Mode 3)	
(2) Internal RAM, no ROM (Mode 2)	
(3) Internal RAM and ROM (Mode 1)	
(4) Internal RAM, ROM with partial address bus (Mode 6)	
Port 3 is a multiplexed address/data bus	
Port 4 is an address bus (inputs/address in Mode 6)	
SC <sub>1</sub> is Address Strobe (AS)	
SC <sub>2</sub> is Read/Write (R/W)	
<b>Test Modes 0 and 4</b>	
Expanded Multiplexed Test Mode 0	
May be used to test RAM and ROM	
Single Chip and Non-Multiplexed Test Mode 4	
(1) May be changed to Mode 5 without going through Reset	
(2) May be used to test Ports 3 and 4 as I/O ports	

#### Expanded-Multiplexed Modes (0, 1, 2, 3, 6)

In the Expanded-Multiplexed Modes, the MCU has the ability to access a 65k bytes memory space. Port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of Address Strobe (AS) and the data bus valid while E is high. In Modes 0 to 3, Port 4 provides address lines A<sub>8</sub> to A<sub>15</sub>. In Mode 6, however, Port 4 is configured during RES as data port inputs and the Data Direction Register can be changed to provide any combination of address lines, A<sub>8</sub> to A<sub>15</sub>. Stated alternatively, any subset of A<sub>8</sub> to A<sub>15</sub> can be provided while retaining the remainder as input data lines. Internal pullup resistors are intended to pull Port 4's lines high until software configures the port.

Figure 19 depicts a typical configuration for the Expanded-Multiplexed Modes. Address Strobe can be used to control a transparent D-type latch to capture addresses A<sub>0</sub> to A<sub>7</sub>, as shown in Figure 20. This allows Port 3 to function as a Data Bus when E is high.

In Mode 0, the Reset vector is external for the first two E-cycles after the positive edge of RES and internal thereafter. In

addition, the internal and external data buses are connected and there must be no memory map overlap to avoid potential bus conflicts. Mode 0 is used primarily to verify the ROM pattern

and monitor the internal data bus with the automated test equipment.

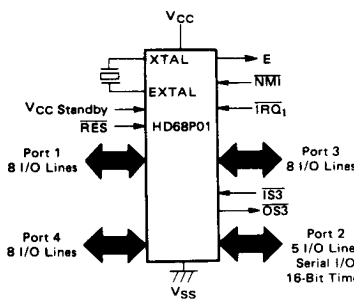


Figure 16 Single Chip Mode

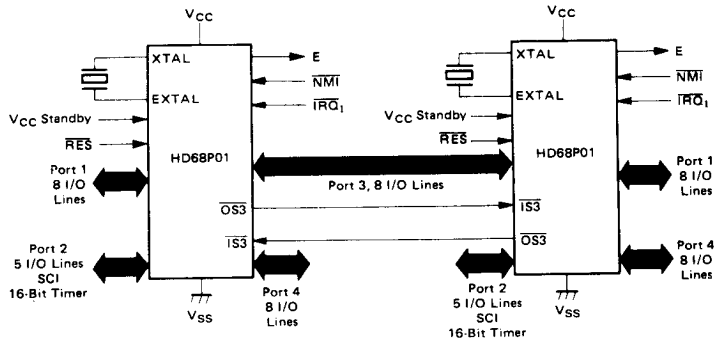


Figure 17 Single Chip Dual Processor Configuration

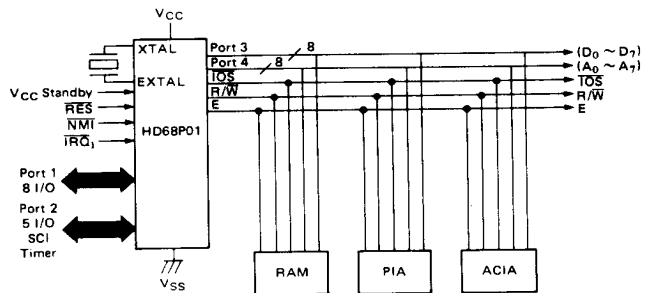
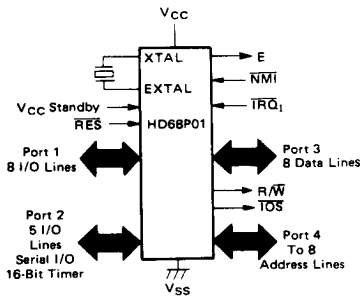


Figure 18 Expanded Non-Multiplexed Configuration

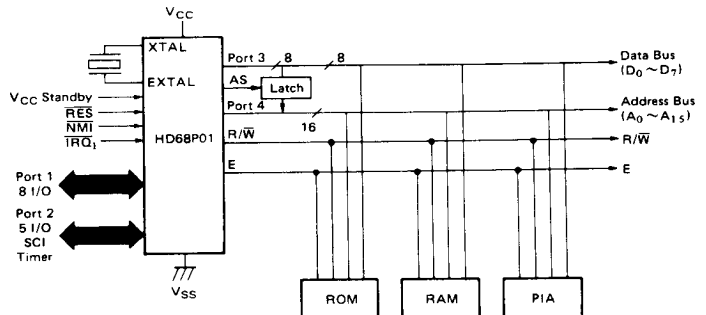
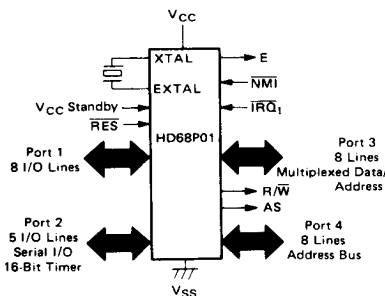


Figure 19 Expanded Multiplexed Configuration

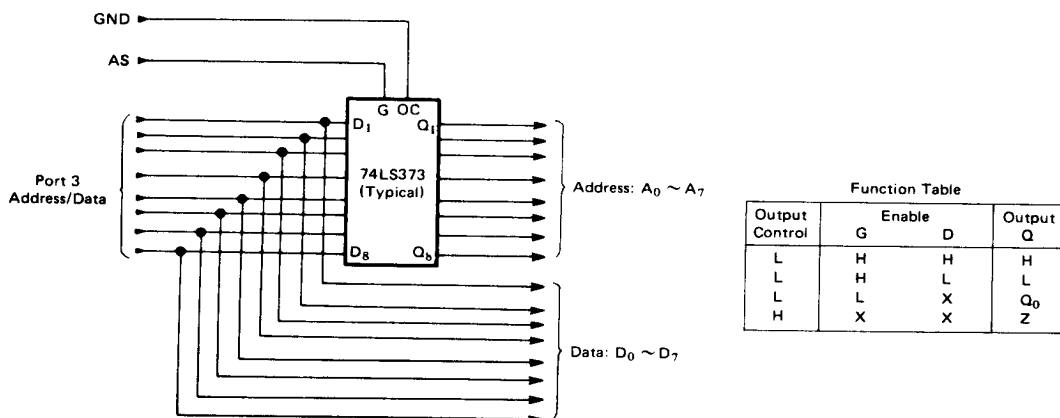


Figure 20 Typical Latch Arrangement

### ● Programming The Mode

The operating mode is programmed by the levels asserted on  $P_{22}$ ,  $P_{21}$ , and  $P_{20}$  which are latched into PC2, PC1, and PC0 of the program control register on the positive edge of RES. The operating mode may be read from Port 2 Data Register as shown below, and programming levels and timing must be met as shown in Figure 8. A brief outline of the operating modes is shown in Table 4.

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the three pins. If configured as outputs, the circuit shown in Figure 21 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

Port 2 Data Register

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	\$0003

Table 4 Mode Selection Summary

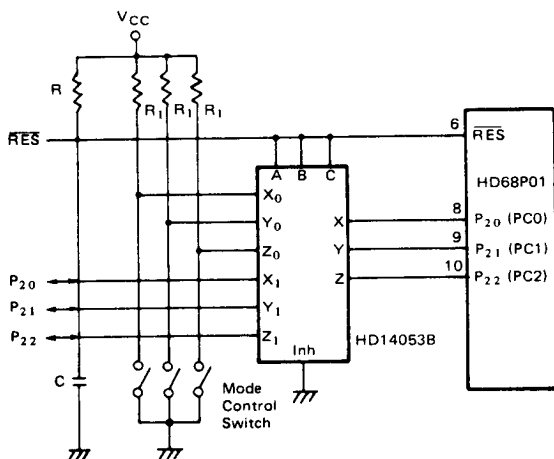
Mode	$P_{22}$ (PC2)	$P_{21}$ (PC1)	$P_{20}$ (PC0)	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX <sup>(5, 6)</sup>	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX <sup>(5, 6)</sup>	Non-Multiplexed/Partial Decode
4	H	L	L	I <sup>(2)</sup>	I <sup>(1)</sup>	I	I	Single Chip Test
3	L	H	H	E	E	E	MUX <sup>(4)</sup>	Multiplexed /No RAM or ROM
2	L	H	L	E	I	E	MUX <sup>(4)</sup>	Multiplexed /RAM
1	L	L	H	I	I	E	MUX <sup>(4)</sup>	Multiplexed/RAM & ROM
0	L	L	L	I	I	I <sup>(3)</sup>	MUX <sup>(4)</sup>	Multiplexed Test

#### Legend:

I — Internal  
E — External  
MUX — Multiplexed  
NMUX — Non-Multiplexed  
L — Logic "0"  
H — Logic "1"

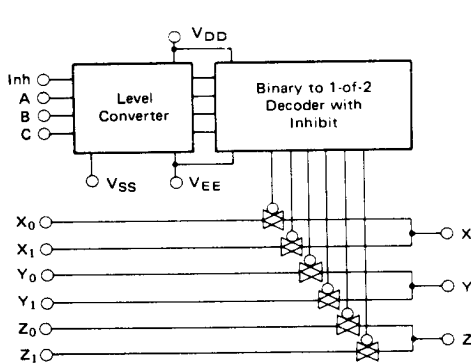
#### Notes:

- (1) Internal RAM is addressed at \$X80
- (2) Internal ROM is disabled
- (3) RES vector is external for 2 cycles after RES goes high
- (4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3
- (5) Addresses associated with Port 3 are considered external in Modes 5 and 6
- (6) Port 4 default is user data input; address output is optional by writing to Port 4 Data Direction Register



[NOTES] 1) Mode 7 as shown  
2)  $RC \approx$  Reset time constant  
3)  $R_1 = 10k\Omega$

Figure 21 Recommended Circuit for Mode Selection



Truth Table						
Control Input				On Switch		
Inhibit	Select			HD14053B		
	C	B	A			
0	0	0	0	Z <sub>0</sub>	Y <sub>0</sub>	X <sub>0</sub>
0	0	0	1	Z <sub>0</sub>	Y <sub>0</sub>	X <sub>1</sub>
0	0	1	0	Z <sub>0</sub>	Y <sub>1</sub>	X <sub>0</sub>
0	0	1	1	Z <sub>0</sub>	Y <sub>1</sub>	X <sub>1</sub>
0	1	0	0	Z <sub>1</sub>	Y <sub>0</sub>	X <sub>0</sub>
0	1	0	1	Z <sub>1</sub>	Y <sub>0</sub>	X <sub>1</sub>
0	1	1	0	Z <sub>1</sub>	Y <sub>1</sub>	X <sub>0</sub>
0	1	1	1	Z <sub>1</sub>	Y <sub>1</sub>	X <sub>1</sub>
1	X	X	X	—		

Figure 22 HD14053B Multiplexers/Demultiplexers



## ■ MEMORY MAPS

The MCU can provide up to 65k byte address space depending on the operating mode. The HD68P01 provides 8k byte address space for EPROM, but the maps differ in EPROM types as follows.

### 1) HN462716 (a 2k byte ROM)

In order to support the HD6801S0, EPROM of the HD68P01S0 must be located at \$F800-\$FFFF.

### 2) HN462532, HN462732 (a 4k byte ROM)

In order to support the HD6801V0 EPROM of the HD68P01V05 and the HD68P01V07 must be located at \$F000-\$FFFF.

### 3) HN482764 (a 8k byte ROM)

The HD68P01M0 can provide up to 8k byte address space using HN482764 instead of HN462732. In this case, EPROM of the HD68P01M0 is located at \$E000-\$FFFF.

A memory map for each operating mode is shown in Figure 23. The first 32 locations of each map are reserved for the MCU's internal register area, as shown in Table 5, with exceptions as indicated.

Table 5 Internal Register Area

Register	Address
Port 1 Data Direction Register***	00
Port 2 Data Direction Register***	01
Port 1 Data Register	02
Port 2 Data Register	03
Port 3 Data Direction Register***	04*
Port 4 Data Direction Register***	05**
Port 3 Data Register	06*
Port 4 Data Register	07**
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Port 3 control and Status Register	0F*
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

\* External address in Modes 0, 1, 2, 3, 5, 6; cannot be accessed in Mode 5 (No I/O)

\*\* External addresses in Modes 0, 1, 2, 3

\*\*\* 1 = Output, 0 = Input

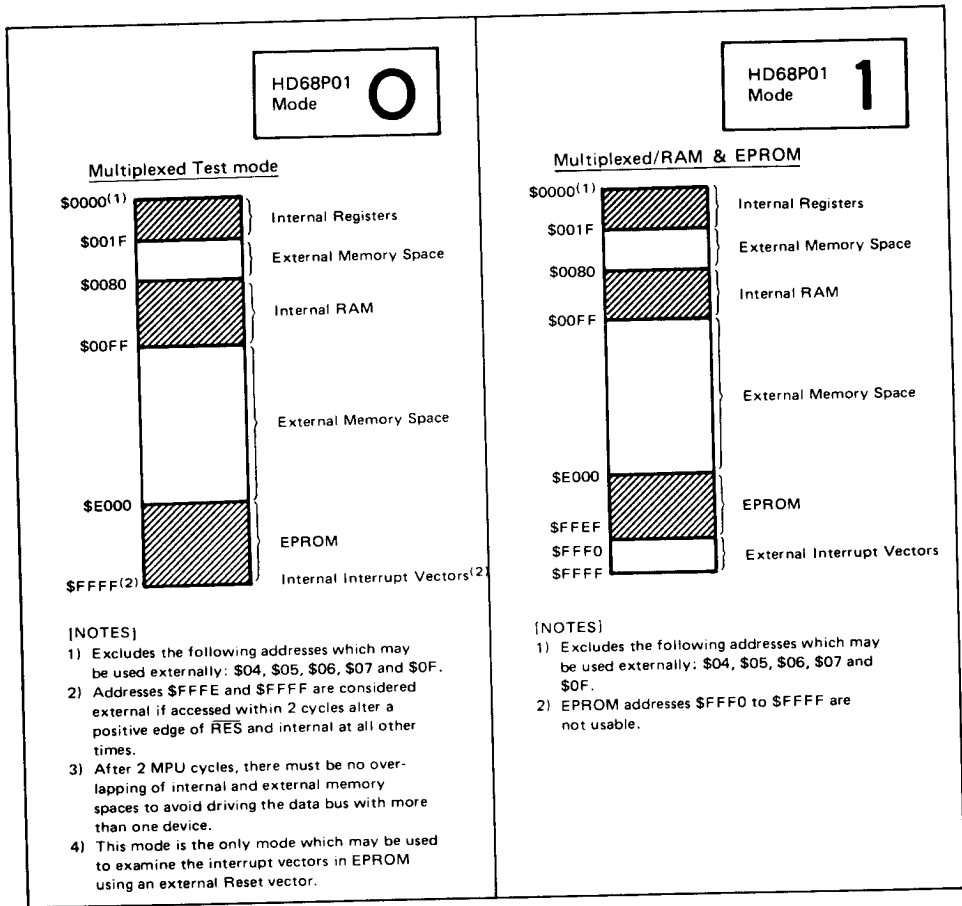


Figure 23 HD68P01 Memory Maps

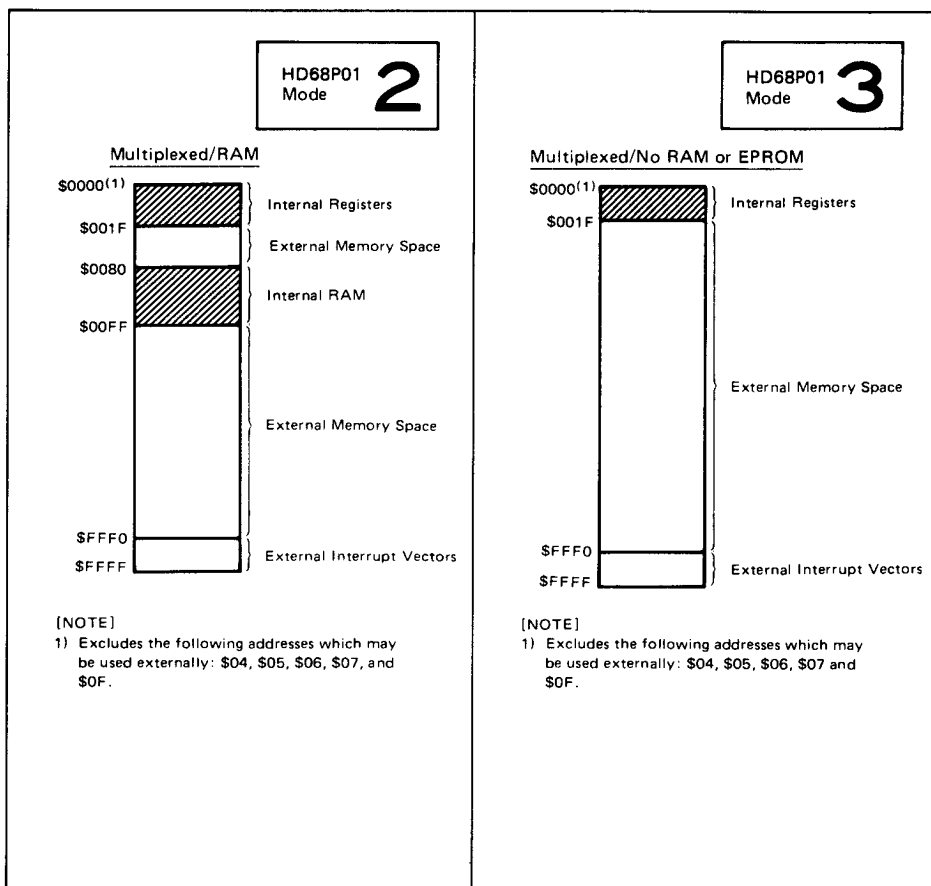


Figure 23 HD68P01 Memory Maps (Continued)

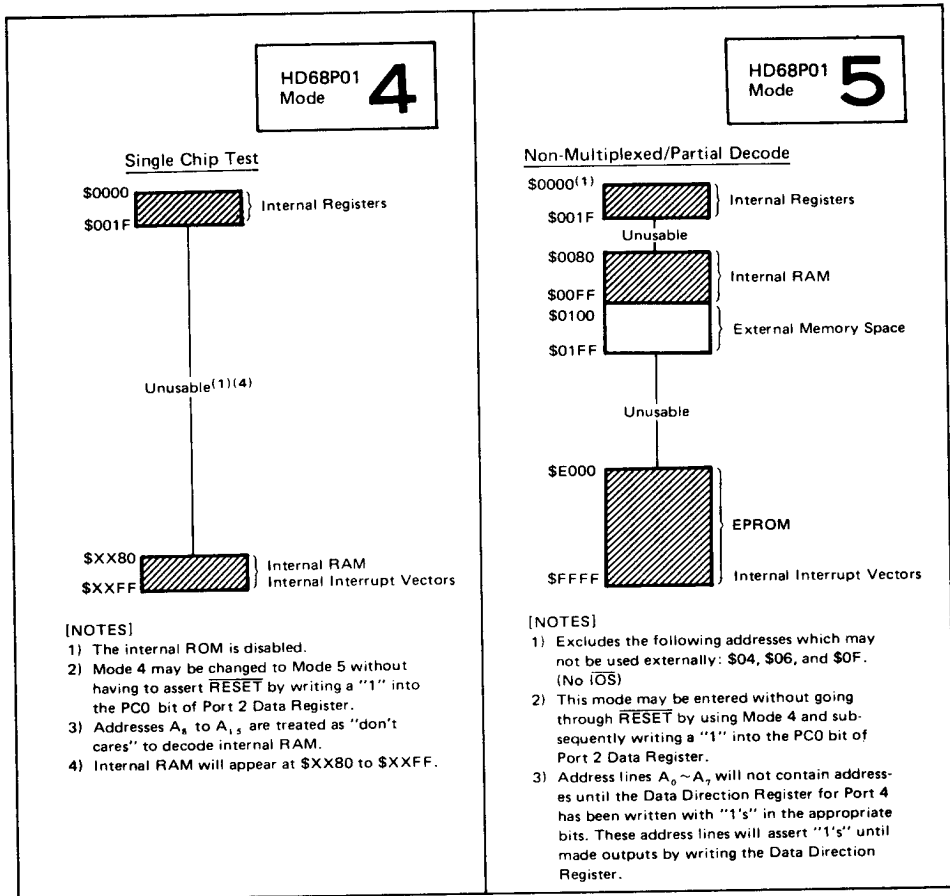


Figure 23 HD68P01 Memory Maps (Continued)

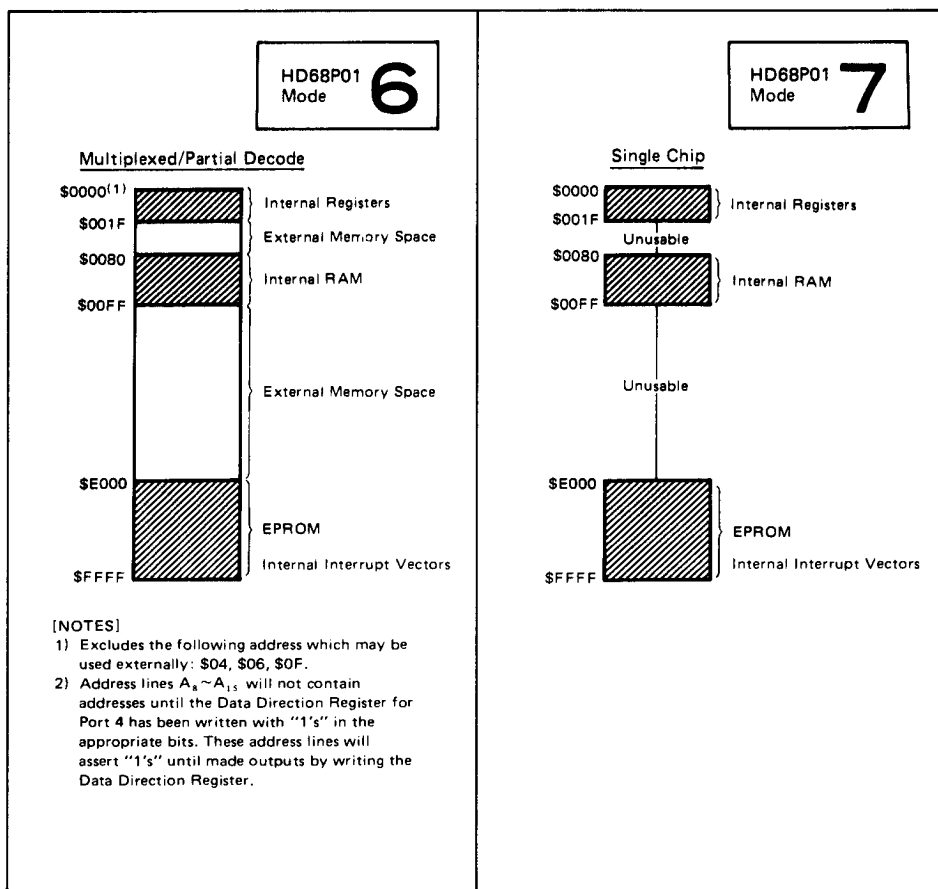


Figure 23 HD68P01 Memory Maps (Continued)



Timer Control and Status Register (TCSR)

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	\$0008

- Bit 0 OLVL** Output level. OLVL is clocked to the output level register by a successful output compare and will appear at P<sub>21</sub> if Bit 1 of Port 2's Data Direction Register is set. It is cleared by RES.
- Bit 1 IEDG** Input Edge. IEDG is cleared by RES and controls which level transition will trigger a counter transfer to the Input Capture Register:  
IEDG = 0 Transfer on a negative-edge  
IEDG = 1 Transfer on a positive-edge.
- Bit 2 ETOI** Enable Timer Overflow Interrupt. When set, an IRQ<sub>2</sub> interrupt is enabled for a timer overflow; when clear, the interrupt is inhibited. It is cleared by RES.
- Bit 3 EOCI** Enable Output Compare Interrupt. When set, an IRQ<sub>2</sub> interrupt is enabled for an output compare; when clear, the interrupt is inhibited. It is cleared by RES.
- Bit 4 EICI** Enable Input Capture Interrupt. When set, an IRQ<sub>2</sub> interrupt is enabled for an input capture; when clear, the interrupt is inhibited. It is cleared by RES.
- Bit 5 TOF** Timer Overflow Flag. TOF is set when the counter contains all 1's. It is cleared by reading the TCSR (with TOF set) followed by the counter's high byte (\$09), or by RES.
- Bit 6 OFC** Output Compare Flag. OFC is set when the Output Compare Register matches the free-running counter. It is cleared by reading the TCSR (with OFC set) and then writing to the Output Compare Register (\$0B or \$0C), or by RES.
- Bit 7 ICF** Input Capture Flag. ICF is set to indicate a proper level transition; it is cleared by reading the TCSR (with ICF set) and then the Input Capture Register High Byte (\$0D), or by RES.

## SERIAL COMMUNICATIONS INTERFACE (SCI)

A full-duplex asynchronous Serial Communications Interface (SCI) is provided with a data format and a variety of rates. The SCI transmitter and receiver are functionally independent, but use the same data format and bit rate. Serial data format is standard mark/space (NRZ) and provides one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

### Wake-Up Feature

In a typical serial loop multi-processor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to permit uninterested MPU's to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive 1's or by RES. Software must provide for the required idle string between consecutive messages and prevent it within messages.

### Programmable Options

- The following features of the SCI are programmable:
  - format: Standard mark/space (NRZ)

- clock: external or internal bit rate clock
- Baud (or bit rate): one of 4 per E-clock frequency, or external bit rate (X8) input
- wake-up feature: enabled or disabled
- interrupt requests: enabled individually for transmitter and receiver
- clock output: internal bit rate clock enabled or disabled to P<sub>22</sub>
- Port 2 (bit 3, 4): dedicated or not dedicated to serial I/O individually for transmitter and receiver.

### Serial Communications Registers

The Serial Communications Interface includes four addressable registers as depicted in Figure 25. It is controlled by the Rate and Mode Control Register and the Transmit/Receive Control and Status Register. Data is transmitted and received utilizing a write-only Transmit Register and a read-only Receive Register. The shift registers are not accessible to software.

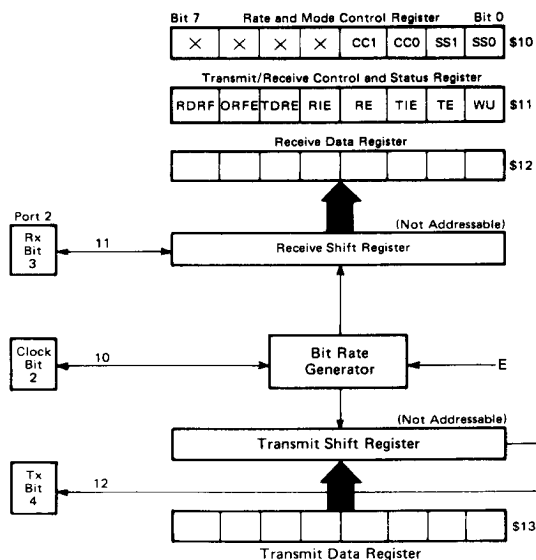


Figure 25 SCI Registers

### Rate and Mode Control Register (RMCR) (\$10)

The Rate and Mode Control Register controls the SCI bit rate, format, clock source, and under certain conditions, the configuration of P<sub>22</sub>. The register consists of four write-only bits which are cleared by RES. The two least significant bits control the bit rate of the internal clock and the remaining two bits control the format and clock source.

Rate and Mode Control Register (RMCR)

7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	SS1	SS0	\$0010

- Bit 1: Bit 0** SS1: SSO Speed Select. These two bits select the Baud when using the internal clock. Four rates may be selected which are a function of the MCU input frequency. Table 6 lists bit time and rates for three selected MCU frequencies.
- Bit 3: Bit 2** CC1:CC0 Clock Control Select. These two bits select the serial clock source. If CC1 is set, the DDR value for P<sub>22</sub> is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 7 defines the clock source, and use of P<sub>22</sub>.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P<sub>22</sub> at eight times (8X) the desired bit rate, but not greater than E, with a duty cycle of 50% (± 10%). If CC1:CC0 = 10, the internal bit rate clock is provided at P<sub>22</sub> regardless of the values for TE or RE.

(Note) The source of SCI internal bit rate clock is the timer's free running counter. An MPU write to the counter can disturb serial operations.

#### Transmit/Receive Control and Status Register (TRCSR) (\$11)

The Transmit/Receive Control and Status Register controls the transmitter, receiver, wake-up feature, and two individual interrupts and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to \$20 by RES.

Transmit/Receive Control and Status Register (TRCSR)

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	\$0011

- Bit 0 WU** "Wake-up" on Idle Line. When set, WU enables the wake-up function; it is cleared by ten consecutive 1's or by RES. WU will not set if the line is idle.
- Bit 1 TE** Transmit Enable. When set, P<sub>24</sub> DDR bit is set, cannot be changed, and will remain set if TE is subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P<sub>24</sub>

- and a preamble of nine consecutive 1's is transmitted. TE is cleared by RES.
- Bit 2 TIE** Transmit Interrupt Enable. When set, an  $\overline{\text{IRQ}}_2$  interrupt is enabled when TDRE is set; when clear, the interrupt is inhibited. TE is cleared by RES.
- Bit 3 RE** Receive Enable. When set, P<sub>23</sub>'s DDR bit is cleared, cannot be changed, and will remain clear if RE is subsequently cleared. While RE is set, the SCI receiver is enabled. RE is cleared by RES.
- Bit 4 RIE** Receiver Interrupt Enable. When set, an  $\overline{\text{IRQ}}_2$  interrupt is enabled when RDRF and/or ORFE is set; when clear, the interrupt is inhibited. RIE is cleared by RES.
- Bit 5 TDRE** Transmit Data Register Empty. TDRE is set when the Transmit Data Register is transferred to the output serial shift register or by RES. It is cleared by reading the TRCSR (with TDRE set) and then writing to the Transmit Data Register. Additional data will be transmitted only if TDRE has been cleared.
- Bit 6 ORFE** Overrun Framing Error. If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the Receiver Data Register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the value of RDRF: if RDRF is set, then an overrun has occurred; otherwise a framing error has been detected. Data is not transferred to the Receive Data Register in an overrun or framing error condition. ORFE is cleared by reading the TRCSR (with ORFE set) then the Receive Data Register, or by RES.
- Bit 7 RDRF** Receive Data Register Full. RDRF is set when the input serial shift register is transferred to the Receive Data Register. It is cleared by reading the TRCSR (with RDRF set), and then the Receive Data Register, or by RES.

Table 6 SCI Bit Times and Rates

SS1 : SS0		XTAL	2.4576 MHz	4.0 MHz
		E	614.4 kHz	1.0 MHz
0	0	E ÷ 16	26 $\mu$ s/38,400 Baud	16 $\mu$ s/62,500 Baud
0	1	E ÷ 128	208 $\mu$ s/4,800 Baud	128 $\mu$ s/7812.5 Baud
1	0	E ÷ 1024	1.67ms/600 Baud	1.024ms/976.6 Baud
1	1	E ÷ 4096	6.67ms/150 Baud	4.096ms/244.1 Baud

Table 7 SCI Format and Clock Source Control

CC1: CC0	Format	Clock Source	Port 2 Bit 2	Port 2 Bit 3	Port 2 Bit 4
0 0	—	—	—	**	**
0 1	NRZ	Internal	Not Used	**	**
1 0	NRZ	Internal	Output*	**	**
1 1	NRZ	External	Input	**	**

\* Clock output is available regardless of values for bits RE and TE.

\*\* Bit 3 is used for serial input if RE = "1" in TRCS; bit 4 is used for serial output if TE = "1" in TRCS.





### ● Internally Generated Clock

If the user wishes for the serial I/O to furnish a clock, the following requirements are applicable:

- the values of RE and TE are immaterial.
- CC1, CC0 must be set to 10
- the maximum clock rate will be  $E \div 16$ .
- the clock will be at 1X the bit rate and will have a rising edge at mid-bit.

### ● Externally Generated Clock

If the user wishes to provide an external clock for the serial I/O, the following requirements are applicable:

- the CC1, CC0, field in the Rate and Mode Control Register must be set to 11,
- the external clock must be set to 8 times (X8) the desired baud rate and
- the maximum external clock frequency is 1.0 MHz.

### ● Serial Operations

The SCI is initialized by writing control bytes first to the Rate and Mode Control Register and then to the Transmit/Receive Control and Status Register.

The Transmitter Enable (TE) and Receiver Enable (RE) bits may be left set for dedicated operations.

### Transmit operations

The transmit operation is enabled by TE in the Transmit/Receive Control and Status Register. When TE is set, the output of the transmit serial shift register is connected to P<sub>24</sub> and the serial output by first transmitting to a ten-bit preamble of 1's. Following the preamble, internal synchronization is established and the transmitter section is ready for operation.

At this point one of two situation exist:

- 1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of ones will be sent indicating an idle line, or,
- 2) if a byte has been written to the Transmit Data Register (TDRE = 0), it is transferred to the output serial shift register and transmission will begin.

During the transfer itself, the start bit (0) is first transmitted.

Then the 8 data bits (beginning with bit 0) followed by the stop bit (1), are transmitted. When the Transmitter Data Register has been emptied, the TDRE flag bit is set.

If the MCU fails to respond to the flag within the proper time, (TDRE is still set when the next normal transfer from the parallel data register to the serial output register should occur) then a 1 will be sent (instead of a 0) at "Start" bit time, followed by more 1's until more data is supplied to the data register. No 0's will be sent while TDRE remains a 1.

### Receive Operations

The receive operation is enabled by RE which configures P<sub>23</sub>. The receive operation is controlled by the contents of the Transmit/Receive Control and Status Register and the Rate and Mode Control Register.

The receiver bit interval is divided into 8 sub-intervals for internal synchronization. In the NRZ Mode, the received bit stream is synchronized by the first 0 (space) encountered.

The approximate center of each bit time is strobed during the next 10 bits. If the tenth bit is not a 1 (stop bit) a framing error is assumed, and ORFE is set. If the tenth bit is a 1, the data is transferred to the Receive Data Register, and interrupt flag RDRF is set. If RDRF is still set at the next tenth bit time, ORFE will be set, indicating an over-run has occurred. When the MCU responds to either flag (RDRF or ORFE) by reading the status register followed by reading the Data Register, RDRF (or ORFE) will be cleared.

### ■ INSTRUCTION SET

The HD68P01 is upward source and object code compatible with the HD6800. Execution times of key instructions have been reduced and several new instructions have been added, including hardware multiply. A list of new operations added to the HD6800 instruction set is shown in Table 8.

In addition, two new special opcodes, 4E and 5E, are provided for test purposes. These opcodes force the Program Counter to increment like a 16-bit counter, causing address lines used in the expanded modes to increment until the device is reset. These opcodes have no mnemonics.

Table 8 New Instructions

Instruction	Description
ABX	Unsigned addition of Accumulator B to Index Register
ADDD	Adds (without carry) the double accumulator to memory and leaves the sum in the double accumulator
ASLD	Shifts the double accumulator left (towards MSB) one bit; the LSB is cleared and the MSB is shifted into the C-bit
BRN	Branch Never
LDD	Loads double accumulator from memory
LSRD	Shifts the double accumulator right (towards LSB) one bit; the MSB is cleared and the LSB is shifted into the C-bit
MUL	Unsigned multiply; multiplies the two accumulators and leaves the product in the double accumulator
PSHX	Pushes the Index Register to stack
PULX	Pulls the Index Register from stack
STD	Stores the double accumulator to memory
SUBD	Subtracts memory from the double accumulator and leaves the difference in the double accumulator

### ● Programming Model

A programming model for the HD68P01 is shown in Figure 10. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulator A and/or B. Other registers are defined as follows:

### Program Counter

The program counter is a 16-bit register which always points to the next instruction.

### Stack Pointer

The stack pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random access memory at a location defined by the programmer.

### Index Register

The Index Register is a 16-bit register which can be used to store data or provide an address for the indexed mode of addressing.

### Accumulators

The MCU contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

### Condition Code Registers

The condition code register indicates the results of an instruction and includes the following five condition bits: Negative (N), Zero (Z), Overflow (V), Carry/Borrow from MSB (C), and Half Carry from bit 3 (H). These bits are testable by the conditional branch instruction. Bit 4 is the interrupt mask (I-bit) and inhibits all maskable interrupts when set. The two unused bits, b6 and b7 are read as ones.

### ● Addressing Modes

The MCU provides six addressing modes which can be used to reference memory. A summary of addressing modes for all instructions is presented in Table 9, 10, 11, and 12 where execution times are provided in E-cycles. Instruction execution times are summarized in Table 13. With an input frequency of 4 MHz, E-cycles are equivalent to microseconds. A cycle-by-cycle description of bus activity for each instruction is provided in Table 14 and a description of selected instructions is shown in Figure 26.

### Immediate Addressing

The operand or "immediate byte(s)" is contained in the following byte(s) of the instruction where the number of bytes matches the size of the register. These are two or three byte instructions.

### Direct Addressing

The least significant byte of the operand address is contained in the second byte of the instruction and the most significant byte is assumed to be \$00. Direct addressing allows the user to access \$00 through \$FF using two byte instructions and execution time is reduced by eliminating the additional memory access. In most applications, the 256-byte area is reserved for frequently referenced data.

### Extended Addressing

The second and third bytes of the instruction contain the absolute address of the operand. These are three byte instructions.

### Indexed Addressing

The unsigned offset contained in the second byte of the instruction is added with carry to the Index Register and used to reference memory without changing the Index Register. These are two byte instructions.

Table 9 Index Register and Stack Manipulation Instructions

Pointer Operations	Mnemonic	Immed			Direct			Index			Extend			Implied			Boolean/ Arithmetic Operation	Cond. Code Reg.					
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		5	4	3	2	1	0
																		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X ← M: M + 1	•	•	↑	↑	↑	↑
Decrement Index Reg	DEX													09	3	1	X ← 1 → X	•	•	•	↑	•	•
Decrement Stack Pntr	DES													34	3	1	SP ← 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	3	1	X ← 1 → X	•	•	•	↑	•	•
Increment Stack Pntr	INS													31	3	1	SP ← 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X <sub>H</sub> , (M + 1) → X <sub>L</sub>	•	•	↑	↑	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP <sub>H</sub> , (M + 1) → SP <sub>L</sub>	•	•	↑	↑	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X <sub>H</sub> → M, X <sub>L</sub> → (M + 1)	•	•	↑	↑	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP <sub>H</sub> → M, SP <sub>L</sub> → (M + 1)	•	•	↑	↑	R	•
Index Reg → Stack Pntr	TXS													35	3	1	X ← 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	3	1	SP ← 1 → X	•	•	•	•	•	•
Add	ABX													3A	3	1	B ← X → X	•	•	•	•	•	•
Push Data	PSHX													3C	4	1	X <sub>L</sub> → M <sub>SP</sub> , SP ← 1 → SP	•	•	•	•	•	•
																	X <sub>H</sub> → M <sub>SP</sub> , SP ← 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	5	1	SP ← 1 → SP, M <sub>SP</sub> → X <sub>H</sub>	•	•	•	•	•	•
																	SP ← 1 → SP, M <sub>SP</sub> → X <sub>L</sub>	•	•	•	•	•	•

The Condition Code Register notes are listed after Table 12.



**Implied Addressing**

The operand(s) are registers and no memory reference is required. These are single byte instructions.

the branch condition is true, the Program Counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current Program Counter. This provides a branch range of -126 to 129 bytes from the first byte of the instruction. These are two byte instructions.

**Relative Addressing**

Relative addressing is used only for branch instructions. If

Table 10 Accumulator and Memory Instructions

Accumulator and Memory Operations	Mnemonic	Immed		Direct		Index		Extend		Implied		Boolean Expression	Cond. Code Reg.					
		OP	~ #	OP	~ #	OP	~ #	OP	~ #	OP	~ #		H	I	N	Z	V	C
Add Acmltrs	ABA										1B 2 1	$A + B \rightarrow A$	↓	•	↓	↓	↓	↓
Add B to X	ABX										3A 3 1	$B + X \rightarrow X$	•	•	•	•	•	•
Add with Carry	ADCA	89	2 2	99	3 2	A9	4 2	B9	4 3			$A + M + C \rightarrow A$	↓	•	↓	↓	↓	↓
	ADCB	C9	2 2	D9	3 2	E9	4 2	F9	4 3			$B + M + C \rightarrow B$	↓	•	↓	↓	↓	↓
Add	ADDA	8B	2 2	9B	3 2	AB	4 2	BB	4 3			$A + M \rightarrow A$	↓	•	↓	↓	↓	↓
	ADDB	CB	2 2	DB	3 2	EB	4 2	FB	4 3			$B + M \rightarrow A$	↓	•	↓	↓	↓	↓
Add Double	ADDD	C3	4 3	D3	5 2	E3	6 2	F3	6 3			$D + M:M + 1 \rightarrow D$	•	•	↓	↓	↓	↓
And	ANDA	84	2 2	94	3 2	A4	4 2	B4	4 3			$A \cdot M \rightarrow A$	•	•	↓	↓	R	•
	ANDB	C4	2 2	D4	3 2	E4	4 2	F4	4 3			$B \cdot M \rightarrow B$	•	•	↓	↓	R	•
Shift Left, Arithmetic	ASL					68	6 2	78	6 3				•	•	↓	↓	↓	↓
	ASLA									48	2 1		•	•	↓	↓	↓	↓
	ASLB									58	2 1		•	•	↓	↓	↓	↓
Shift Left Dbl	ASLD									05	3 1		•	•	↓	↓	↓	↓
Shift Right, Arithmetic	ASR					67	6 2	77	6 3				•	•	↓	↓	↓	↓
	ASRA									47	2 1		•	•	↓	↓	↓	↓
	ASRB									57	2 1		•	•	↓	↓	↓	↓
Bit Test	BITA	85	2 2	95	3 2	A5	4 2	B5	4 3			$A \cdot M$	•	•	↓	↓	R	•
	BITB	C5	2 2	D5	3 2	E5	4 2	F5	4 3			$B \cdot M$	•	•	↓	↓	R	•
Compare Acmltrs	CBA									11	2 1	$A - B$	•	•	↓	↓	↓	↓
Clear	CLR					6F	6 2	7F	6 3			$00 \rightarrow M$	•	•	R	S	R	R
	CLRA									4F	2 1	$00 \rightarrow A$	•	•	R	S	R	R
	CLRB									5F	2 1	$00 \rightarrow B$	•	•	R	S	R	R
Compare	CMPA	81	2 2	91	3 2	A1	4 2	B1	4 3			$A - M$	•	•	↓	↓	↓	↓
	CMPB	C1	2 2	D1	3 2	E1	4 2	F1	4 3			$B - M$	•	•	↓	↓	↓	↓
	COM					63	6 2	73	6 3			$\bar{M} \rightarrow M$	•	•	↓	↓	R	S
1's Complement	COMA									43	2 1	$\bar{A} \rightarrow A$	•	•	↓	↓	R	S
	COMB									53	2 1	$\bar{B} \rightarrow B$	•	•	↓	↓	R	S
	DAA									19	2 1	Adj binary sum to BCD	•	•	↓	↓	↓	↓
Decrement	DEC					6A	6 2	7A	6 3			$M - 1 \rightarrow M$	•	•	↓	↓	↓	•
	DECA									4A	2 1	$A - 1 \rightarrow A$	•	•	↓	↓	↓	•
	DECB									5A	2 1	$B - 1 \rightarrow B$	•	•	↓	↓	↓	•
Exclusive OR	EORA	88	2 2	98	3 2	A8	4 2	B8	4 3			$A \oplus M \rightarrow A$	•	•	↓	↓	R	•
	EORB	C8	2 2	D8	3 2	E8	4 2	F8	4 3			$B \oplus M \rightarrow B$	•	•	↓	↓	R	•
Increment	INC					6C	6 2	7C	6 3			$M + 1 \rightarrow M$	•	•	↓	↓	↓	•
	INCA									4C	2 1	$A + 1 \rightarrow A$	•	•	↓	↓	↓	•
	INCB									5C	2 1	$B + 1 \rightarrow B$	•	•	↓	↓	↓	•
Load Acmltrs	LDAA	86	2 2	96	3 2	A6	4 2	B6	4 3			$M \rightarrow A$	•	•	↓	↓	R	•
	LDAB	C6	2 2	D6	3 2	E6	4 2	F6	4 3			$M \rightarrow B$	•	•	↓	↓	R	•
Load Double	LDD	CC	3 3	DC	4 2	EC	5 2	FC	5 3			$M:M + 1 \rightarrow D$	•	•	↓	↓	R	•
Logical Shift, Left	LSL					68	6 2	78	6 3				•	•	↓	↓	↓	↓
	LSLA									48	2 1		•	•	↓	↓	↓	↓
	LSLB									58	2 1		•	•	↓	↓	↓	↓
	LSLD									05	3 1		•	•	↓	↓	↓	↓
Shift Right, Logical	LSR					64	6 2	74	6 3				•	•	R	↓	↓	↓
	LSRA									44	2 1		•	•	R	↓	↓	↓
	LSRB									54	2 1		•	•	R	↓	↓	↓
	LSRD									04	3 1		•	•	R	↓	↓	↓

(Continued)

Table 10 Accumulator and Memory Instructions (Continued)

Accumulator and Memory Operations	Mnemonic	Immed			Direct			Index			Extend			Implied			Boolean Expression	Cond. Code Reg.					
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Multiply	MUL													3D	10	1	A X B → D	•	•	•	•	•	↑
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↑	↑	↑	↑
	NEGA													40	2	1	00 - A → A	•	•	↑	↑	↑	↑
	NEGB													50	2	1	00 - B → B	•	•	↑	↑	↑	↑
No Operation	NOP													01	2	1	PC + 1 → PC	•	•	•	•	•	•
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↑	↑	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↑	↑	R	•
Push Data	PSHA													36	3	1	A → Stack	•	•	•	•	•	•
	PSHB													37	3	1	B → Stack	•	•	•	•	•	•
Pull Data	PULA													32	4	1	Stack → A	•	•	•	•	•	•
	PULB													33	4	1	Stack → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↑	↑	↑	↑
	ROLA													49	2	1		•	•	↑	↑	↑	↑
	ROLB													59	2	1		•	•	↑	↑	↑	↑
Rotate Right	ROR							66	6	2	76	6	3					•	•	↑	↑	↑	↑
	RORA													46	2	1		•	•	↑	↑	↑	↑
	RORB													56	2	1		•	•	↑	↑	↑	↑
Subtract Acmltr	SBA													10	2	1	A - B → A	•	•	↑	↑	↑	↑
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↑	↑	↑	↑
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↑	↑	↑	↑
Store Acmltrs	STAA							97	3	2	A7	4	3				A → M	•	•	↑	↑	R	•
	STAB							D7	3	2	E7	4	3				B → M	•	•	↑	↑	R	•
	STD							DD	4	2	ED	5	3				D → M: M + 1	•	•	↑	↑	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	↑	↑	↑	↑
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	↑	↑	↑	↑
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D - M: M + 1 → D	•	•	↑	↑	↑	↑
Transfer Acmltr	TAB													16	2	1	A → B	•	•	↑	↑	R	•
	TBA													17	2	1	B → A	•	•	↑	↑	R	•
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	•	•	↑	↑	R	R
	TSTA													4D	2	1	A - 00	•	•	↑	↑	R	R
	TSTB													5D	2	1	B - 00	•	•	↑	↑	R	R

The Condition Code Register notes are listed after Table 12.

Table 11 Jump and Branch Instructions

Operations	Mnemonic	Direct		Relative			Index			Extend			Implied			Branch Test	Cond. Code Reg.							
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~		#	5	4	3	2	1	0	
																		H	I	N	Z	V	C	
Branch Always	BRA				20	3	2										None	•	•	•	•	•	•	
Branch Never	BRN				21	3	2										None	•	•	•	•	•	•	
Branch If Carry Clear	BCC				24	3	2										C = 0	•	•	•	•	•	•	
Branch If Carry Set	BCS				25	3	2										C = 1	•	•	•	•	•	•	
Branch If = Zero	BEQ				27	3	2										Z = 1	•	•	•	•	•	•	
Branch If ≥ Zero	BGE				2C	3	2										N ⊕ V = 0	•	•	•	•	•	•	
Branch If > Zero	BGT				2E	3	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•	
Branch If Higher	BHI				22	3	2										C + Z = 0	•	•	•	•	•	•	
Branch If Higher or Same	BHS				24	3	2										C = 0	•	•	•	•	•	•	
Branch If ≤ Zero	BLE				2F	3	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•	
Branch If Carry Set	BLO				25	3	2										C = 1	•	•	•	•	•	•	
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	•	•	•	•	•	•	
Branch If < Zero	BLT				2D	3	2										N ⊕ V = 1	•	•	•	•	•	•	
Branch If Minus	BMI				2B	3	2										N = 1	•	•	•	•	•	•	
Branch If Not Equal Zero	BNE				26	3	2										N = 0	•	•	•	•	•	•	
Branch If Overflow Clear	BVC				28	3	2										V = 0	•	•	•	•	•	•	
Branch If Overflow Set	BVS				29	3	2										V = 1	•	•	•	•	•	•	
Branch If Plus	BPL				2A	3	2										N = 0	•	•	•	•	•	•	
Branch To Subroutine	BSR				8D	6	2											•	•	•	•	•	•	
Jump	JMP							6E	3	2		7E	3	3			See Special Operations Figure 26	•	•	•	•	•	•	
Jump To Subroutine	JSR	9D	5	2				AD	6	2		BD	6	3				•	•	•	•	•	•	•
No Operation	NOP														01	2	1	•	•	•	•	•	•	
Return From Interrupt	RTI														3B	10	1	See Special Operations Figure 26	↑	↓	↑	↓	↑	↓
Return From Subroutine	RTS														39	5	1		•	•	•	•	•	•
Software Interrupt	SWI														3F	12	1		•	S	•	•	•	•
Wait For Interrupt	WAI														3E	9	1		•	•	•	•	•	•

The Condition Code Register notes are listed after Table 12.

Table 12 Condition Code Register Manipulation Instructions

Operations	Implied				Boolean Operation	Cond. Code Reg.					
	Mnemonic	OP	~	#		5	4	3	2	1	0
						H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	0 → C	●	●	●	●	●	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	●	R	●	●	●	●
Clear Overflow	CLV	0A	2	1	0 → V	●	●	●	●	R	●
Set Carry	SEC	0D	2	1	1 → C	●	●	●	●	●	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	●	S	●	●	●	●
Set Overflow	SEV	0B	2	1	1 → V	●	●	●	●	S	●
Accumulator A → CCR	TAP	06	2	1	A → CCR	↑	↑	↑	↑	↑	↑
CCR → Accumulator A	TPA	07	2	1	CCR → A	●	●	●	●	●	●

#### LEGEND

OP Operation Code (Hexadecimal)  
~ Number of MPU Cycles  
MSP Contents of memory location pointed to by Stack Pointer  
# Number of Program Bytes  
+ Arithmetic Plus  
- Arithmetic Minus  
• Boolean AND  
X Arithmetic Multiply  
+ Boolean Inclusive OR  
⊕ Boolean Exclusive OR  
M Complement of M  
→ Transfer Into  
0 Bit = Zero  
00 Byte = Zero

#### CONDITION CODE SYMBOLS

H Half-carry from bit 3  
I Interrupt mask  
N Negative (sign bit)  
Z Zero (byte)  
V Overflow, 2's complement  
C Carry/Borrow from MSB  
R Reset Always  
S Set Always  
↑ Affected  
• Not Affected

Table 13 Instruction Execution Times in E-Cycles

	Addressing Mode					
	Immediate	Direct	Extended	Indexed	Implied	Relative
ABA	•	•	•	•	2	•
ABX	•	•	•	•	3	•
ADC	2	3	4	4	•	•
ADD	2	3	4	4	•	•
ADDD	4	5	6	6	•	•
AND	2	3	4	4	•	•
ASL	•	•	6	6	2	•
ASLD	•	•	•	•	3	•
ASR	•	•	6	6	2	•
BCC	•	•	•	•	•	3
BCS	•	•	•	•	•	3
BEQ	•	•	•	•	•	3
BGE	•	•	•	•	•	3
BGT	•	•	•	•	•	3
BHI	•	•	•	•	•	3
BHS	•	•	•	•	•	3
BIT	2	3	4	4	•	•
BLE	•	•	•	•	•	3
BLO	•	•	•	•	•	3
BLS	•	•	•	•	•	3
BLT	•	•	•	•	•	3
BMI	•	•	•	•	•	3
BNE	•	•	•	•	•	3
BPL	•	•	•	•	•	3
BRA	•	•	•	•	•	3
BRN	•	•	•	•	•	3
BSR	•	•	•	•	•	6
BVC	•	•	•	•	•	3
BVS	•	•	•	•	•	3
CBA	•	•	•	•	2	•
CLC	•	•	•	•	2	•
CLI	•	•	•	•	2	•
CLR	•	•	6	6	2	•
CLV	•	•	•	•	2	•
CMP	2	3	4	4	•	•
COM	•	•	6	6	2	•
CPX	4	5	6	6	•	•
DAA	•	•	•	•	2	•
DEC	•	•	6	6	2	•
DES	•	•	•	•	3	•
DEX	•	•	•	•	3	•
EOR	2	3	4	4	•	•
INC	•	•	6	6	•	•
INS	•	•	•	•	3	•

	Addressing Mode					
	Immediate	Direct	Extended	Indexed	Implied	Relative
INX	•	•	•	•	3	•
JMP	•	•	3	3	•	•
JSR	•	5	6	6	•	•
LDA	2	3	4	4	•	•
LDD	3	4	5	5	•	•
LDS	3	4	5	5	•	•
LDX	3	4	5	5	•	•
LSL	•	•	6	6	2	•
LSLD	•	•	•	•	3	•
LSR	•	•	6	6	2	•
LSRD	•	•	•	•	3	•
MUL	•	•	•	•	10	•
NEG	•	•	6	6	2	•
NOP	•	•	•	•	2	•
ORA	2	3	4	4	•	•
PSH	•	•	•	•	3	•
PSHX	•	•	•	•	4	•
PUL	•	•	•	•	4	•
PULX	•	•	•	•	5	•
ROL	•	•	6	6	2	•
ROR	•	•	6	6	2	•
RTI	•	•	•	•	10	•
RTS	•	•	•	•	5	•
SBA	•	•	•	•	2	•
SBC	2	3	4	4	•	•
SEC	•	•	•	•	2	•
SEI	•	•	•	•	2	•
SEV	•	•	•	•	2	•
STA	•	3	4	4	•	•
STD	•	4	5	5	•	•
STS	•	4	5	5	•	•
STX	•	4	5	5	•	•
SUB	2	3	4	4	•	•
SUBD	4	5	6	6	•	•
SWI	•	•	•	•	12	•
TAB	•	•	•	•	2	•
TAP	•	•	•	•	2	•
TBA	•	•	•	•	2	•
TPA	•	•	•	•	2	•
TST	•	•	6	6	2	•
TSX	•	•	•	•	3	•
TXS	•	•	•	•	3	•
WAI	•	•	•	•	9	•

# ■ SUMMARY OF CYCLE BY CYCLE OPERATION

Table 14 provides a detailed description of the information present on the Address Bus, Data Bus, and the Read/Write (R/W) line during each cycle of each instruction.

The information is useful in comparing actual with expected results during debug to both software and hardware as the program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction.

In general, instructions with the same addressing mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

Note that during MPU reads of internal locations, the resultant value will not appear on the external Data Bus except in Mode 0. "High order" byte refers to the most significant byte of a 16-bit value.

Table 14 Cycle by Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/W Line	Data Bus	
IMMEDIATE						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1	Op Code Address	1	Op Code	
		2	Op Code Address + 1	1	Operand Data	
LDS LDX LDD	3	1	Op Code Address	1	Op Code	
		2	Op Code Address + 1	1	Operand Data (High Order Byte)	
		3	Op Code Address + 2	1	Operand Data (Low Order Byte)	
CPX SUBD ADDD	4	1	Op Code Address	1	Op Code	
		2	Op Code Address + 1	1	Operand Data (High Order Byte)	
		3	Op Code Address + 2	1	Operand Data (Low Order Byte)	
		4	Address Bus FFFF	1	Low Byte of Restart Vector	
DIRECT						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1	Op Code Address	1	Op Code	
		2	Op Code Address + 1	1	Address of Operand	
		3	Address of Operand	1	Operand Data	
STA	3	1	Op Code Address	1	Op Code	
		2	Op Code Address + 1	1	Destination Address	
		3	Destination Address	0	Data from Accumulator	
LDS LDX LDD	4	1	Op Code Address	1	Op Code	
		2	Op Code Address + 1	1	Address of Operand	
		3	Address of Operand	1	Operand Data (High Order Byte)	
		4	Operand Address + 1	1	Operand Data (Low Order Byte)	
STS STX STD	4	1	Op Code Address	1	Op Code	
		2	Op Code Address + 1	1	Address of Operand	
		3	Address of Operand	0	Register Data (High Order Byte)	
		4	Address of Operand + 1	0	Register Data (Low Order Byte)	
CPX SUBD ADDD	5	1	Op Code Address	1	Op Code	
		2	Op Code Address + 1	1	Address of Operand	
		3	Operand Address	1	Operand Data (High Order Byte)	
		4	Operand Address + 1	1	Operand Data (Low Order Byte)	
		5	Address Bus FFFF	1	Low Byte of Restart Vector	
JSR	5	1	Op Code Address	1	Op Code	
		2	Op Code Address + 1	1	Irrelevant Data	
		3	Subroutine Address	1	First Subroutine Op Code	
		4	Stack Pointer	0	Return Address (Low Order Byte)	
		5	Stack Pointer + 1	0	Return Address (High Order Byte)	

(Continued)

Table 14 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/ $\bar{W}$ Line	Data Bus
<b>EXTENDED</b>					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS LDX LDD	5	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS STX STD	5	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR ASR NEG CLR ROL COM ROR DEC TST* INC	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Current Operand Data
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address of Operand	0	New Operand Data
CPX SUBD ADD	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Operand Address (High Order Byte)
		3	Op Code Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)

\* In the TST instruction, the line condition of the sixth cycle does the following: R/ $\bar{W}$  = "High", AB = FFFF, DB = Low Byte of Reset Vector.

(Continued)



Table 14 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/ $\overline{W}$ Line	Data Bus
<b>INDEXED</b>					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS LDX LDD	5	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS STX STD	5	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR ASR NEG CLR ROL COM ROR DEC TST * INC	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Current Operand Data
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Index Register Plus Offset	0	New Operand Data
CPX SUBD ADD	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Op Code
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)

\* In the TST instruction, the line condition of the sixth cycle does the following: R/ $\overline{W}$  = "High", AB = FFFF, DB = Low Byte of Restart Vector.

(Continued)

Table 14 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/W Line	Data Bus
<b>IMPLIED</b>					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Op Code Address Op Code Address + 1	1 1	Op Code Op Code of Next Instruction
ABX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Op Code Address Op Code Address + 1 Previous Register Contents	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 0	Op Code Op Code of Next Instruction Accumulator Data
TSX	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
TXS	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Op Code Op Code of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1	1 1 0 0	Op Code Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)

(Continued)

Table 14 Cycle by Cycle Operation (Continued)

Address Mode & Instruction	Cycles	Cycle #	Address Bus	R/W Line	Data Bus
WAI **	9	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Op Code of Next Instruction
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer - 1	0	Return Address (High Order Byte)
		5	Stack Pointer - 2	0	Index Register (Low Order Byte)
		6	Stack Pointer - 3	0	Index Register (High Order Byte)
		7	Stack Pointer - 4	0	Contents of Accumulator A
		8	Stack Pointer - 5	0	Contents of Accumulator B
		9	Stack Pointer - 6	0	Contents of Cond. Code Register
MUL	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Cond. Code Reg. from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer - 1	0	Return Address (High Order Byte)
		5	Stack Pointer - 2	0	Index Register (Low Order Byte)
		6	Stack Pointer - 3	0	Index Register (High Order Byte)
		7	Stack Pointer - 4	0	Contents of Accumulator A
		8	Stack Pointer - 5	0	Contents of Accumulator B
		9	Stack Pointer - 6	0	Contents of Cond. Code Register
		10	Stack Pointer - 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)

\*\* While the MCU is in the "Wait" state, its bus state will appear as a series of the MCU reads of an address which is seven locations less than the original contents of the Stack Pointer. Contrary to the HD6800, none of the ports are driven to the high impedance state by a WAI instruction.

(Continued)

Table 14 Cycle by Cycle Operation (Continued)

Address Mode & Instruction	Cycles	Cycle #	Address Bus	R/W Line	Data Bus
<b>RELATIVE</b>					
BCC BHT BNE BLO	3	1	Op Code Address	1	Op Code
BCS BLE BPL BHS		2	Op Code Address + 1	1	Branch Offset
BEQ BLS BRA BRN		3	Address Bus FFFF	1	Low Byte of Restart Vector
BGE BLT BVC	6				
BGT BMT BVS					
BSR		1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)


#### ■ SUMMARY OF UNDEFINED INSTRUCTIONS OPERATION

The MCU has 36 undefined instructions. When these are carried out, the contents of Register and Memory in MPU change at random.

When the op codes (4E, 5E) are used to execute, the MPU continues to increase the program counter and it will not stop until the Reset signal enters. These op codes are used to test the LSI.

Table 15 Op Codes Map

HD68P01 MICROCOMPUTER INSTRUCTIONS																		
OP CODE						ACC A	ACC B	IND	EXT	ACCA or SP				ACCB or X				
		HI	0000	0001	0010	0011	0100	0101	0110	0111	IMM	DIR	IND	EXT	IMM	DIR	IND	EXT
LO		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0		SBA	BRA	TSX	NEG								SUB				0
0001	1	NOP	CBA	BRN	INS									CMP				1
0010	2			BHI	PULA (+1)									SBC				2
0011	3			BLS	PULB (+1)	COM				*	SUBD (+2)		*	ADDD (+2)		3		
0100	4	LSRD (+1)		BCC	DES	LSR								AND				4
0101	5	ASLD (+1)		BCS	TXS									BIT				5
0110	6	TAP	TAB	BNE	PSHA	ROR								LDA				6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA			STA		7		
1000	8	INX (+1)		BVC	PULX (+2)	ASL								EOR				8
1001	9	DEX (+1)	DAA	BVS	RTS (+2)	ROL								ADC				9
1010	A	CLV		BPL	ABX	DEC								ORA				A
1011	B	SEV	ABA	BMI	RTI (+7)									ADD				B
1100	C	CLC		BGE	PSHX (+1)	INC						CPX (+2)		*	LDD (+1)		C	
1101	D	SEC		BLT	MUL (+7)	TST				BSR (+4)		JSR (+2)		* (+1)	STD (+1)		D	
1110	E	CLI		BGT	WAI (+6)	**		JMP (-3)		*	LDS (+1)		*	LDX (+1)		E		
1111	F	SEI		BLE	SWI (+9)	CLR				* (+1)		STS (+1)		* (+1)	STX (+1)		F	
BYTE/CYCLE		1/2	1/2	2/3	1/3	1/2	1/2	2/6	3/6	2/2	2/3	2/4	3/4	2/2	2/3	2/4	3/4	

(NOTES) 1. Undefined Op codes are marked with .

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

3. The instructions shown below are all 3 bytes and are marked with "\*\*".

Immediate addressing mode of SUBD, CPX, LDS, ADDD, LDD and LDX instructions, and undefined op codes (BF, CD, CF).

4. The Op codes (4E, 5E) are 1 byte/∞ cycles instructions, and are marked with "\*\*".

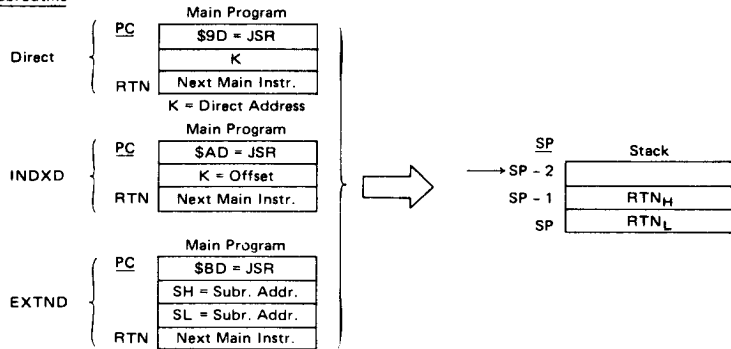
#### ■ COMPATIBLE WITH THE HD6801S AND THE HD6801V

In order to be pin compatible with the HD6801S, ROM of the HD68P01 must be located at \$F800 - \$FFFF. Memory addresses \$E000 to \$F7FF are not usable. The other addresses are available same as the HD6801S's.

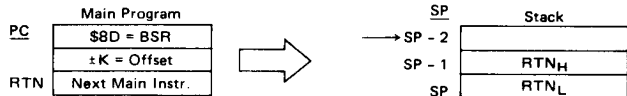
In order to be pin compatible with the HD6801V ROM of the HD68P01 must be located at \$F000 - \$FFFF. Memory addresses \$E000 to \$EFFF are not usable. The other addresses are available same as the HD6801V's.



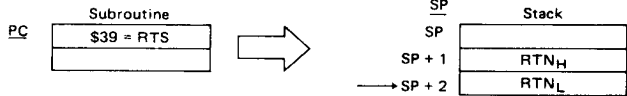
### JSR, Jump to Subroutine



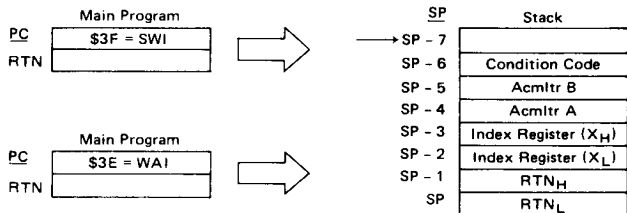
### BSR, Branch to Subroutine



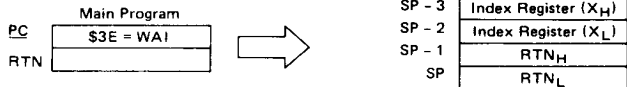
### RTS, Return from Subroutine



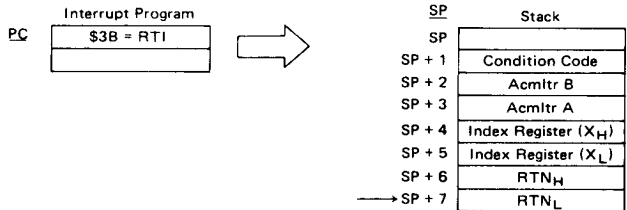
### SWI, Software Interrupt



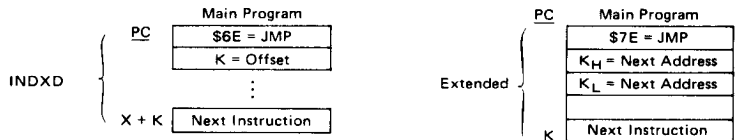
### WAI, Wait for Interrupt



### RTI, Return from Interrupt



### JMP, Jump



Legend:

- RTN = Address of next instruction in Main Program to be executed upon return from subroutine
- RTN<sub>H</sub> = Most significant byte of Return Address
- RTN<sub>L</sub> = Least significant byte of Return Address
- = Stack Pointer After Execution
- K = 8-bit Unsigned Value

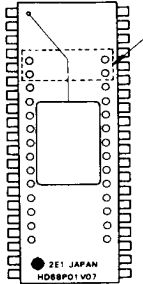
Figure 26 Special Operations



# ■ 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



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

- EPROM (24 pins), let the index-side four pins open.
- (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.