

*Advance Information*

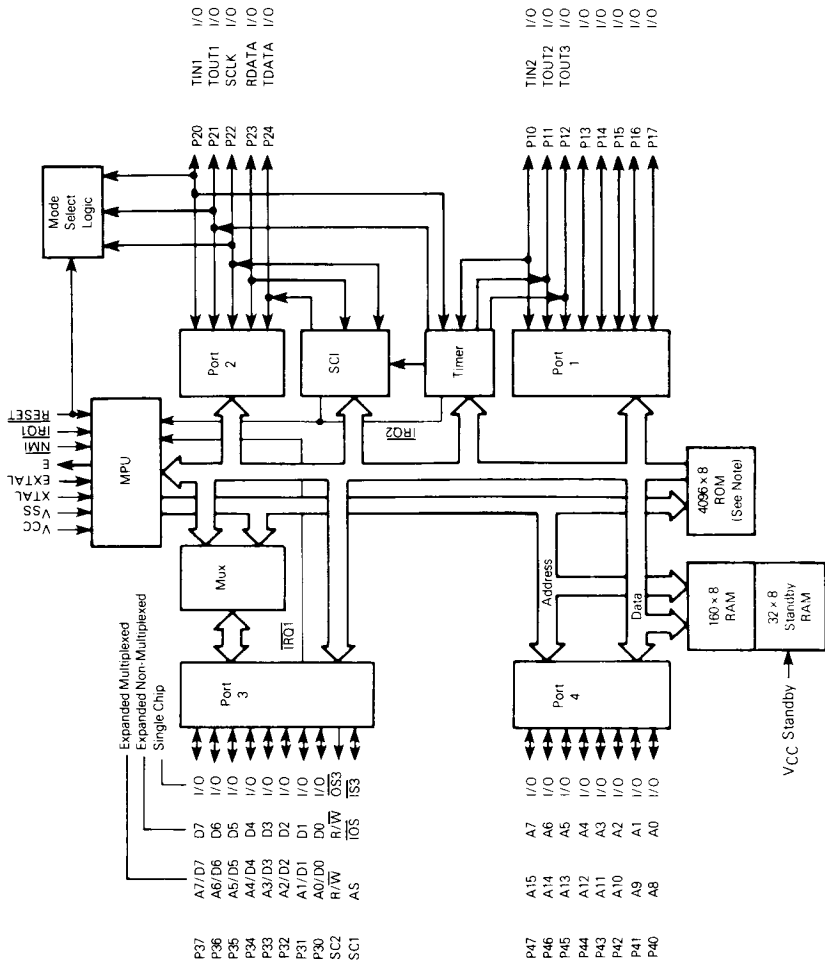
**Microcontroller/Microprocessor (MCU/MPU)**

The MC6801U4 is an 8-bit single-chip microcontroller unit(MCU) that enhances the capabilities of the MC6801 and significantly enhances the capabilities of the M6800 Family of parts. It includes an MC6801 microprocessor unit (MPU) with direct object-code compatibility and upward object-code compatibility with the MC6800. Execution times of key instructions have been improved over the MC6800, and the new instructions found on the MC6801 are included. The MCU can function as a monolithic microcontroller or can be expanded to a 64K-byte address space. It is TTL compatible and requires one +5-volt power supply. On-chip resources include 4096 bytes of ROM, 192 bytes of RAM, a serial communications interface (SCI), parallel I/O, and a 16-bit six-function programmable timer. The MC6803U4 can be considered an MC6801U4 operating in modes 2 or 3; i.e., those that do not use internal ROM.

- Enhanced MC6800 Instruction Set
- Upward Source and Object Code Compatibility with the MC6800 and MC6801
- Bus Compatibility with the M6800 Family
- 8 × 8 Multiply Instruction
- Single-Chip or Expanded Operation to 64K-Byte Address Space
- Internal Clock Generator with Divide-by-Four Output
- Serial Communications Interface (SCI)
- 16-Bit Six-Function Programmable Timer
- Three Output Compare Functions
- Two Input Capture Functions
- Counter Alternate Address
- 4096 Bytes of ROM (MC6801U4)
- 192 Bytes of RAM
- 32 Bytes of RAM Retainable During Powerdown
- 29 Parallel I/O and Two Handshake Control Lines
- NMI Inhibited Until Stack Load
- -40°C to 85°C Temperature Range

This document contains information on a new product. Specifications and information herein are subject to change without notice.

MC6801U4 MICROCOMPUTER FAMILY BLOCK DIAGRAM



NOTE: No functioning ROM in MC6803U4.

MC6801U4 MICROCONTROLLER FAMILY BLOCK DIAGRAM



## MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	$V_{CC}$	-0.3 to +7.0	V
Input Voltage	$V_{in}$	-0.3 to +7.0	V
Operating Temperature Range MC6801U4, MC6803U4 MC6801U4C, MC6803U4C	$T_A$	$T_L$ to $T_H$ -0 to 70 -40 to 85	°C
Storage Temperature Range	$T_{stg}$	-55 to +150	°C

## THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Rating
Thermal Resistance Plastic Ceramic	$\theta_{JA}$	50 50	°C/W

## POWER CONSIDERATIONS

The average chip-junction temperature,  $T_J$ , in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

- $T_A$  = Ambient Temperature, °C
- $\theta_{JA}$  = Package Thermal Resistance,  
Junction-to-Ambient, °C/W
- $P_D$  =  $P_{INT} + P_{PORT}$
- $P_{INT}$  =  $I_{CC} \times V_{CC}$ , Watts — Chip Internal Power
- $P_{PORT}$  = Port Power Dissipation, Watts — User Determined

For most applications  $P_{PORT} < P_{INT}$  and can be neglected.  $P_{PORT}$  may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between  $P_D$  and  $T_J$  (if  $P_{PORT}$  is neglected) is:

$$P_D = K \cdot (T_J - 273^\circ\text{C}) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273^\circ\text{C}) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring  $P_D$  (at equilibrium) for a known  $T_A$ . Using this value of K, the values of  $P_D$  and  $T_J$  can be obtained by solving equations (1) and (2) iteratively for any value of  $T_A$ .

CONTROL TIMING ( $V_{CC} = 5.0\text{ V} \pm 5\%$ ,  $V_{SS} = 0$ )

Characteristic	Symbol	MC6801U4		MC6801U4-1		Unit
		Min	Max	Min	Max	
Frequency of Operation	$f_o$	0.5	1.0	0.5	1.25	MHz
Crystal Frequency	$f_{XTAL}$	2.0	4.0	2.0	5.0	MHz
External Oscillator Frequency	$4 f_o$	2.0	4.0	2.0	5.0	MHz
Crystal Oscillator Startup Time	$t_{rc}$	—	100	—	100	ms
Processor Control Setup Time	$t_{PCS}$	200	—	170	—	ns

# MC6801U4/6803U4

## DC ELECTRICAL CHARACTERISTICS (V<sub>CC</sub> = 5.0 Vdc ± 5%, V<sub>SS</sub> = 0, T<sub>A</sub> = T<sub>L</sub> to T<sub>H</sub>, unless otherwise noted)

Characteristic	Symbol	MC6801U4, MC6803U4		MC6801U4C, MC6803U4C		Unit
		Min	Max	Min	Max	
Input High Voltage RESET Other Inputs*	V <sub>IH</sub>	V <sub>SS</sub> + 4.0 V <sub>SS</sub> + 2.0	V <sub>CC</sub> V <sub>CC</sub>	V <sub>SS</sub> + 4.0 V <sub>SS</sub> + 2.2	V <sub>CC</sub> V <sub>CC</sub>	V
Input Low Voltage All Inputs*	V <sub>IL</sub>	V <sub>SS</sub> - 0.3	V <sub>SS</sub> + 0.8	V <sub>SS</sub> - 0.3	V <sub>SS</sub> + 0.8	V
Input Load Current Port 4 SC1	I <sub>in</sub>	-	0.5 0.8	-	0.8 1.0	mA
Input Leakage Current (V <sub>in</sub> = 0 to 5.5 V) NMI, IRQ1, RESET	I <sub>in</sub>	-	2.5	-	5.0	μA
Hi-Z (Off-State) Input Current (V <sub>in</sub> = 0.5 to 2.4 V) Port 1, Port 2, Port 3	I <sub>TSI</sub>	-	10	-	20	μA
Output High Voltage (I <sub>Load</sub> = -65 μA, V <sub>CC</sub> = Min) (I <sub>Load</sub> = -100 μA, V <sub>CC</sub> = Min) Port 4, SC1, SC2 Other Outputs	V <sub>OH</sub>	V <sub>SS</sub> + 2.4 V <sub>SS</sub> + 2.4	- -	V <sub>SS</sub> - 2.4 V <sub>SS</sub> - 2.4	- -	V
Output Low Voltage (I <sub>Load</sub> = 2.0 mA, V <sub>CC</sub> = Min) All Outputs	V <sub>OL</sub>	-	V <sub>SS</sub> + 0.5	-	V <sub>SS</sub> + 0.6	V
Darlington Drive Current (V <sub>O</sub> = 1.5 V) Port 1	I <sub>OH</sub>	1.0	4.0	1.0	5.0	mA
Internal Power Dissipation (Measured at T <sub>A</sub> = T <sub>L</sub> in Steady-State Operation)***	P <sub>INT</sub>	-	1200	-	1500	mW
Input Capacitance (V <sub>in</sub> = 0, T <sub>A</sub> = 25°C, f <sub>o</sub> = 1.0 MHz) Port 3, Port 4, SC1 Other Inputs	C <sub>in</sub>	-	12.5 10.0	-	12.5 10.0	pF
V <sub>CC</sub> Standby Powerup	I <sub>SBB</sub>	4.0	5.25	4.0	5.25	V
Standby Current Powerdown	I <sub>SBB</sub>	-	3.0	-	3.5	mA

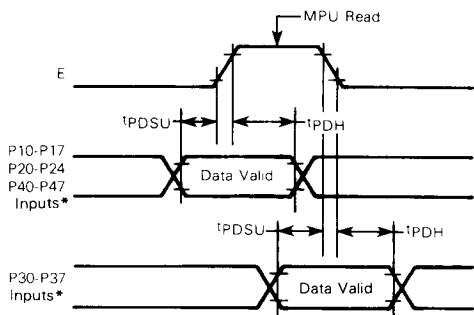
\*Except mode programming levels; see Figure 16.

\*\*Negotiable to -100 μA (for further information contact the factory).

\*\*\*For the MC6801U4/MC6803U4 T<sub>L</sub> = 0°C and for the MC6801U4C/MC6803U4C T<sub>L</sub> = 40°C.

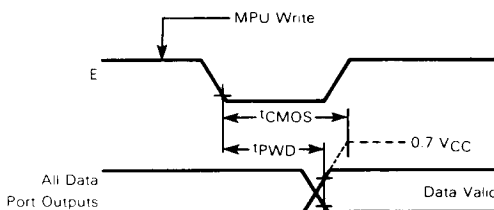
## PERIPHERAL PORT TIMING (Refer to Figure 1-4)

Characteristics	Symbol	Min	Typ	Max	Unit
Peripheral Data Setup Time	t <sub>PDSL</sub>	200	-	-	ns
Peripheral Data Hold Time	t <sub>PDH</sub>	200	-	-	ns
Delay Time, Enable Positive Transition to OS3 Negative Transition	t <sub>OSD1</sub>	-	-	350	ns
Delay Time, Enable Positive Transition to OS3 Positive Transition	t <sub>OSD2</sub>	-	-	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid Port 1 Port 2, 3, 4	t <sub>PWD</sub>	-	-	350 350	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t <sub>CMOS</sub>	-	-	2.0	μs
Input Strobe Pulse Width	t <sub>PWS</sub>	200	-	-	ns
Input Data Hold Time	t <sub>H</sub>	50	-	-	ns
Input Data Setup Time	t <sub>S</sub>	20	-	-	ns



\*Port 3 nonlatched operation (Latch enable = 0)

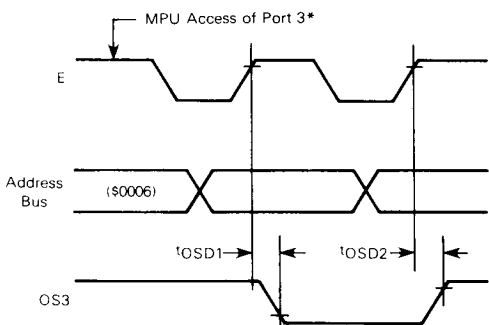
Figure 1. Data Setup and Hold Times (MPU Read)



NOTES:

1. 10 k pullup resistor required for port 2 to reach 0.7 V<sub>CC</sub>
2. Not applicable to P21
3. Port 4 cannot be pulled above V<sub>CC</sub>

Figure 2. Data Setup and Hold Times (MPU Write)



\* Access matches output strobe select (OSS = 0, a read; OSS = 1, a write)

NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

Figure 3. Port 3 Output Strobe Timing (MC6801U4 Single-Chip Mode)

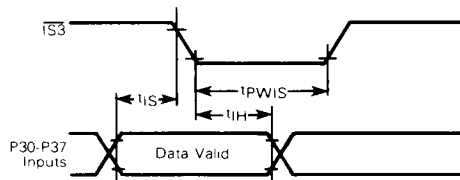


Figure 4. Port 3 Latch Timing (MC6801U4 Single-Chip Mode)

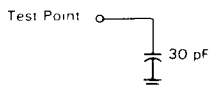
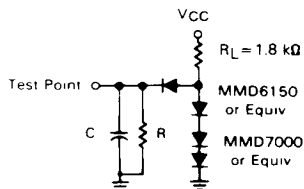


Figure 5. CMOS Load



- C = 90 pF for P30-P37, P40-P47, E, SC1, SC2
- = 30 pF for P10-P17, P20-P24
- R = 37 kΩ for P40-P47, SC1, SC2
- = 24 kΩ for P10-P17, P20-P24
- = 24 kΩ for P30-P37, E

Figure 6. Timing Test Load Ports 1, 2, 3, and 4

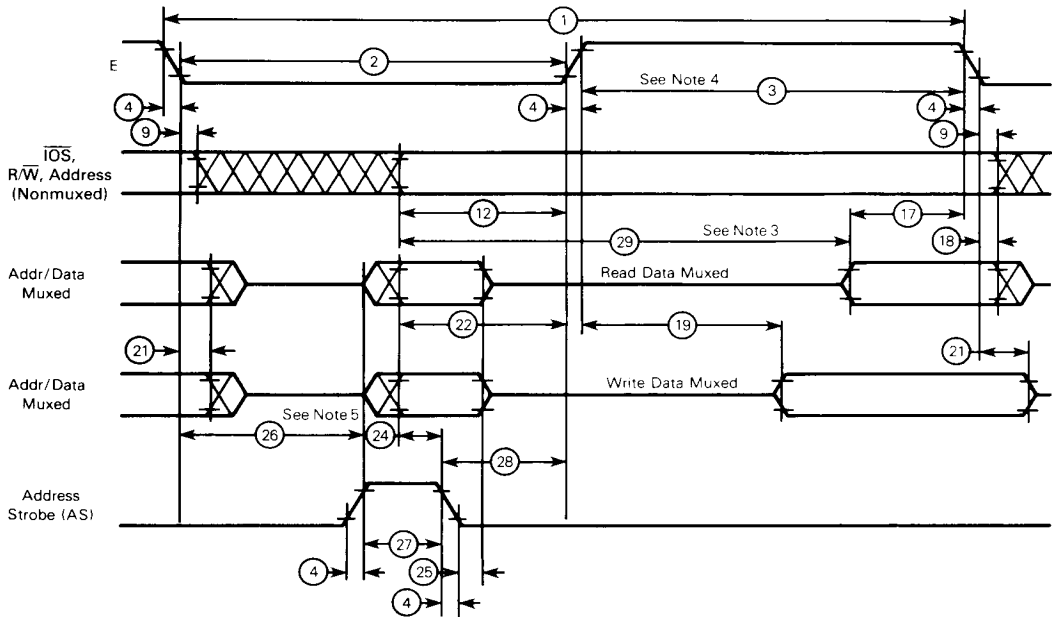


**BUS TIMING** (See Notes 1 and 2, and Figure 7)

Ident. Number	Characteristics	Symbol	MC6801U4 MC6803U4		MC6801U4-1 MC6803U4-1		Unit
			Min	Max	Min	Max	
1	Cycle Time	$t_{cyc}$	1.0	2.0	0.8	2.0	$\mu s$
2	Pulse Width, E Low	PW <sub>EL</sub>	430	1000	360	1000	ns
3	Pulse Width, E High	PW <sub>EH</sub>	450	1000	360	1000	ns
4	Clock Rise and Fall Time	$t_r, t_f$	—	25	—	25	ns
9	Address Hold Time	$t_{AH}$	20	—	20	—	ns
12	Nonmuxed Address Valid Time to E*	$t_{AV}$	200	—	150	—	ns
17	Read Data Setup Time	$t_{DSR}$	80	—	70	—	ns
18	Read Data Hold Time	$t_{DHR}$	10	—	10	—	ns
19	Write Data Delay Time	$t_{DDW}$	—	225	—	200	ns
21	Write Data Hold Time	$t_{DHW}$	20	—	20	—	ns
22	Muxed Address Valid Time to E Rise*	$t_{AVM}$	160	—	120	—	ns
24	Muxed Address Valid Time to AS Fall*	$t_{ASL}$	40	—	30	—	ns
25	Muxed Address Hold Time	$t_{AHL}$	20	—	20	—	ns
26	Delay Time, E to AS Rise*	$t_{ASD}$	200	—	170	—	ns
27	Pulse Width, AS High*	PW <sub>ASH</sub>	100	—	80	—	ns
28	Delay Time, AS to E Rise*	$t_{ASED}$	90	—	70	—	ns
29	Usable Access Time*(See Note 3)	$t_{ACC}$	555	—	435	—	ns

\* At specified cycle time.

3



**NOTES:**

1. Voltage levels shown are  $V_L \leq 0.5 V$ ,  $V_H \geq 2.4 V$ , unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Usable access time is computed by  $22 + 3 - 17 + 4$ .
4. Memory devices should be enabled only during E high to avoid port 3 bus contention.
5. Item 26 is different from the MC6801 but it is upward compatible.

**Figure 7. Bus Timing**

**INTRODUCTION**

The MC6801U4 is an 8-bit monolithic microcontroller that can be configured to function in a wide variety of applications. The facility that 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 40 MCU 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 the hardware associated with the port. When the port is used as a "data port" or "I/O port," it is controlled by the port data direction register and the programmer has direct access to the port pins using the port data register. Port pins are labeled as Pij where i identifies one of four ports and j indicates the particular bit.

The MPU is an enhanced MC6800 MPU with additional capabilities and greater throughput. It is upward source and object-code compatible with the MC6800 and the MC6801. The programming model is depicted in Figure 8 where accumulator D is a concatenation of accumulators A and B. A list of new operations added to the MC6800 instruction set are shown in Table 1.

The MC6803U4 can be considered an MC6801U4 that operates in modes 2 and 3 only.

**OPERATING MODES**

The MC6801U4 provides seven different operating modes (modes 0 through 3 and 5 through 7), and the

MC6803U4 provides two operating modes (modes 2 and 3). The operating modes are hardware selectable and determine the device memory map, the configuration of port 3, port 4, SC1, SC2, and the physical location of the interrupt vectors.

**FUNDAMENTAL MODES**

The seven operating modes (0-3, 5-7) can be grouped into three fundamental modes which refer to the type of bus it supports: single chip, expanded nonmultiplexed, and expanded multiplexed. Single chip is mode 7, expanded nonmultiplexed is mode 5, and the remaining 5 are expanded-multiplexed modes. Table 2 summarizes the characteristics of the operating modes.

**MC6801U4 Single-Chip Mode (7)**

In the single-chip mode, the four MCU ports are configured as parallel I/O data ports, as shown in Figure 9. The MCU functions as a monolithic microcontroller in this mode without external address or data buses. A maximum of 29 I/O lines and two port 3 control lines are provided. Peripherals or another MCU can be interfaced to port 3 in a loosely coupled dual-processor configuration, as shown in Figure 10.

**MC6801U4 Expanded-Nonmultiplexed Mode (5)**

A modest amount of external memory space is provided in the expanded-nonmultiplexed mode while significant on-chip resources are retained. Port 3 functions as an 8-bit bidirectional data bus, and port 4 is configured initially as an input data port. Any combination of the eight least-significant address lines may be obtained by writing to the port 4 data direction register. Stated alternatively, and combination of A0 to A7 may be provided while retaining the remainder as input data lines. Internal pullup resistors pull the port 4 lines high until the port is configured.

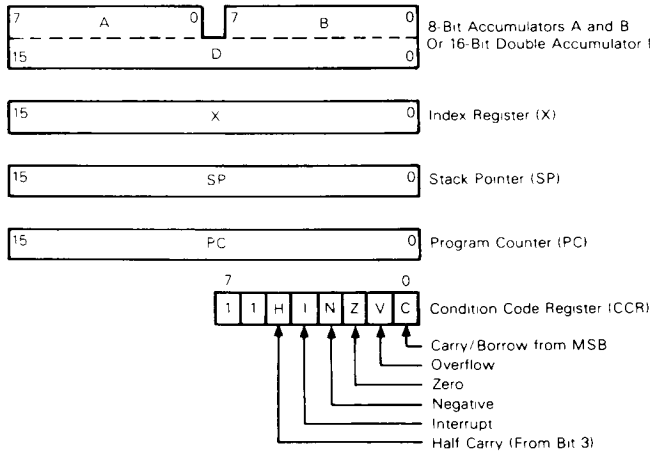


Figure 8. Programming Model



Table 1. 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 or LSLD	Shifts the double accumulator left (towards MSB) one bit, the LSB is cleared, and the MSB is shifted into the C bit
BHS	Branch if higher or same, unsigned conditional branch (same as BCC)
BLO	Branch if lower, unsigned conditional branch (same as BCS)
BRN	Branch never
JSR	Additional addressing mode direct
LDD	Loads double accumulator from memory
LSL	Shifts memory or accumulator left (towards MSB) one bit, the LSB is cleared, and the MSB is shifted into the C bit (same as ASL)
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
CPX	Internal processing modified to permit its use with any conditional branch instruction

Table 2. Summary of MC6801U4/MC6803U4 Operating Modes

<p><b>Single-Chip (Mode 7)</b>                      192 bytes of RAM, 4096 bytes of ROM                      Port 3 is a parallel I/O port with two control lines                      Port 4 is a parallel I/O port</p> <p><b>Expanded Non-Multiplexed (Mode 5)</b>                      192 bytes of RAM, 4096 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</p> <p><b>Expanded Multiplexed (Modes 0, 1, 2, 3, 6*)</b>                      Four memory space options (total 64K address space)                      (1) Internal RAM and ROM with partial address bus (mode 1)                      (2) Internal RAM, no ROM (mode 2)                      (3) Extended addressing of internal I/O and RAM                      (4) Internal RAM and ROM with partial address bus (mode 6)                      Port 3 is multiplexed address/data bus                      Port 4 is address bus (inputs/address in mode 6)                      Test mode (mode 0):                      May be used to test internal RAM and ROM                      May be used to test ports 3 and 4 as I/O ports by writing into mode 7                      Only modes 5, 6, and 7 can be irreversibly entered from mode 0</p> <p><b>Resources Common to All Modes</b>                      Reserved register area                      Port 1 input/output operation                      Port 2 input/output operation                      Timer operation                      Serial communications interface operation</p>
--

\* The MC6803U4 operates only in modes 2 and 3.

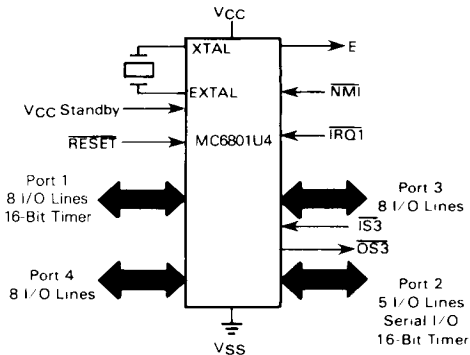


Figure 9. Single-Chip Mode

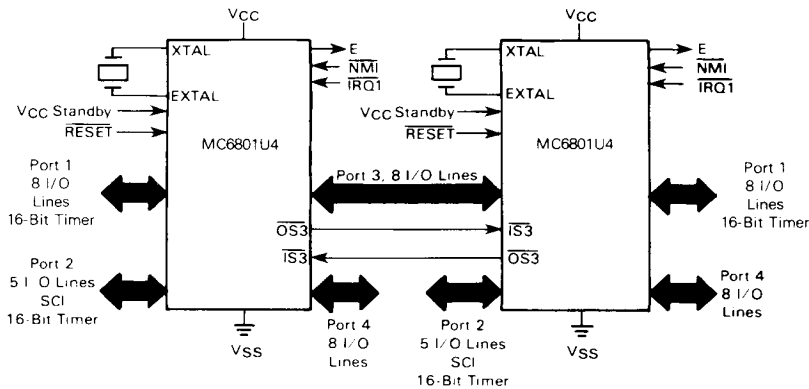


Figure 10. Single-Chip Dual Processor Configuration



Figure 11 illustrates a typical system configuration in the expanded-nonmultiplexed mode. The MCU interfaces directly with M6800 family parts and can access 256 bytes of external address space at \$100 through \$1FF. IOS provides an address decode of external memory (\$100-\$1FF) and can be used as a memory-page select or chip-select line.

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

A 64K-byte memory space is provided in the expanded-multiplexed modes. In each of the expanded-multiplexed modes, port 3 stands as a time-multiplexed address-data bus with address valid on the negative edge of address strobe (AS) and data valid while E is high. In modes 0, 2, and 3, port 4 provides address lines A8 to A15. In modes 1 and 6, however, port 4 initially is configured at reset as an input data port. The port 4 data direction register can then be changed to provide any combination of address lines A8 to A15. Stated alternatively, any subset of A8 to A15 can be provided while retaining the

remaining port 4 lines as input data lines. Internal pullup resistors pull the port 4 lines high until software configures the port. In mode 1, the internal pullup resistors will hold the upper address lines high, producing a value of \$FFXX for a reset vector. A simple method of getting the desired address lines configured as outputs is to have an external EPROM not fully decoded so it appears at two address locations (i.e., \$FXXX and \$BXXX). Then, when the reset vector appears as \$FFFE, the EPROM will be accessed and can point to an address in the top \$100 bytes of the internal or external ROM/EPROM that will configure port 4 as desired.

In mode 0, the reset and interrupt vectors are located at \$BFF0-\$BFFF. In addition, the internal and external data buses are connected; therefore, there must be no memory map overlap to avoid potential bus conflicts. By writing the PC0-PC2 bits in the port 2 data register, modes 5, 6, and 7 can be irreversibly entered from mode 0. Mode 0 is used primarily to verify the ROM pattern and to monitor the internal data bus with the automated test equipment.

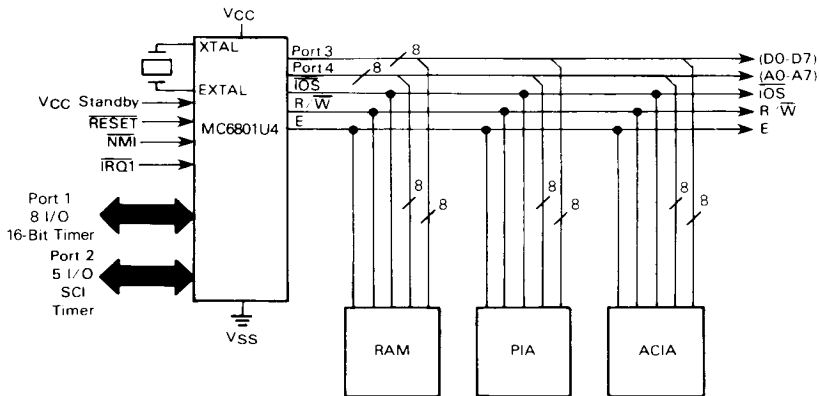
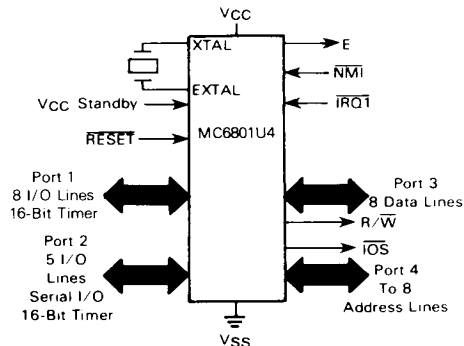


Figure 11. Expanded-Nonmultiplexed Configuration

Only the MC6801U4 can operate in each of the expanded-multiplexed modes. The MC6803U4 operates only in modes 2 and 3.

Figure 12 depicts a typical configuration for the expanded-multiplexed modes. The AS can be used to control a transparent D-type latch to capture addresses A0-A7, which allows port 3 to function as a data bus when E is high, as shown in Figure 13.

**PROGRAMMING THE MODE**

The operating mode is determined at  $\overline{\text{RESET}}$  by the levels asserted on P22, P21, and P20. These levels are latched into PC2, PC1, and PC0 of the program control register on the positive edge of  $\overline{\text{RESET}}$ . The operating mode may be read from the port 2 data register, as shown below, and programming levels and timing must be met as shown in Figure 14. A brief outline of the operating modes is shown in Table 3.

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the

**PORT 2 DATA REGISTER**

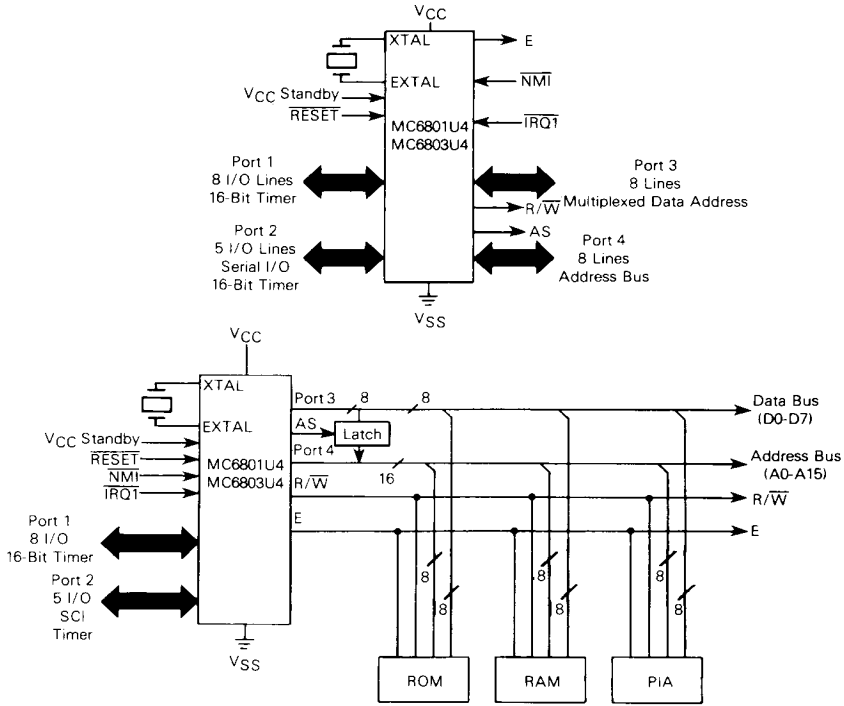
7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	S03

three pins. If configured as outputs, the circuit shown in Figure 15 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode. If diodes are used to program the mode, the diode forward voltage drop must not exceed the  $V_{\text{MPDD}}$  minimum.

**MEMORY MAPS**

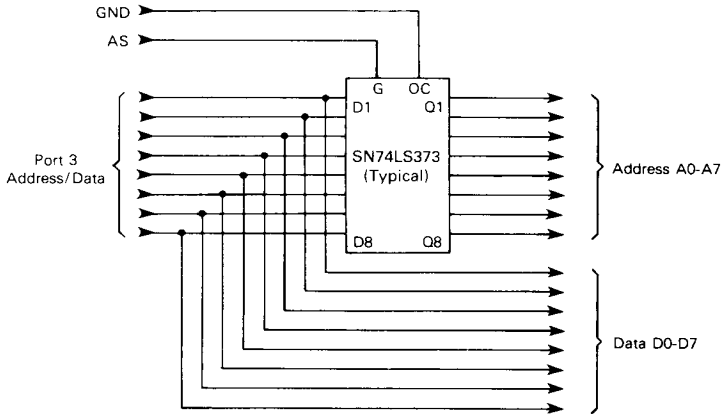
The MC6801U4/MC6803U4 can provide up to 64K-byte address space, depending on the operating mode. A memory map for each operating mode is shown in Figure 16. The first 32 locations of each map are reserved for the internal register area, as shown in Table 4, with exceptions as indicated.

**MC6801U4/6803U4**



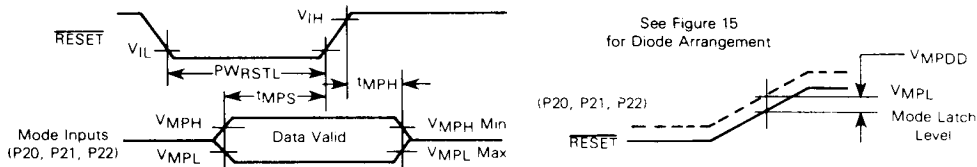
NOTE: To avoid data bus (port 3) contention in the expanded multiplexed modes, memory devices should be enabled only during E high time.

**Figure 12. Expanded-Multiplexed Configuration**



**Figure 13. Typical Latch Arrangement**

**3**



**MODE PROGRAMMING** (Refer to Figure 14)

Characteristic	Symbol	Min	Max	Unit
Mode Programming Input Voltage Low (For $T_A = 0-70^\circ\text{C}$ )	V <sub>MPL</sub>	-	1.8	V
Mode Programming Input Voltage High	V <sub>MPH</sub>	4.0	-	V
Mode Programming Diode Differential (If Diodes are Used) (For $T_A = 0-70^\circ\text{C}$ )	V <sub>MPDD</sub>	0.6	-	V
RESET Low Pulse Width	PWRSTL	3.0	-	E Cycles
Mode Programming Setup Time	t <sub>MPS</sub>	2.0	-	E Cycles
Mode Programming Hold Time	t <sub>MPH</sub>	0	-	ns
RESET Rise Time $\geq 1 \mu\text{s}$		100	-	
RESET Rise Time $< 1 \mu\text{s}$			-	

**NOTE:**

For  $T_A = -40-85^\circ\text{C}$ , V<sub>MPL</sub> Max = 1.7, and V<sub>MPDD</sub> Min = 0.4

**Figure 14. Mode Programming Timing**

**Table 3. Mode Selection Summary**

Mode*	P22 PC2	P21 PC1	P20 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(2, 3)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(2, 3)	Nonmultiplexed Partial Decode
4	H	L	L	-	-	-	-	Undefined <sup>(4)</sup>
3	L	H	H	E	I	E	MUX(1, 5)	Multiplexed/RAM
2	L	H	L	E	I	E	MUX(1)	Multiplexed/RAM
1	L	L	H	I	I	E	MUX (2,3)	Multiplexed/RAM and ROM
0	L	L	L	I	I	E	MUX <sup>(1)</sup>	Multiplexed Test

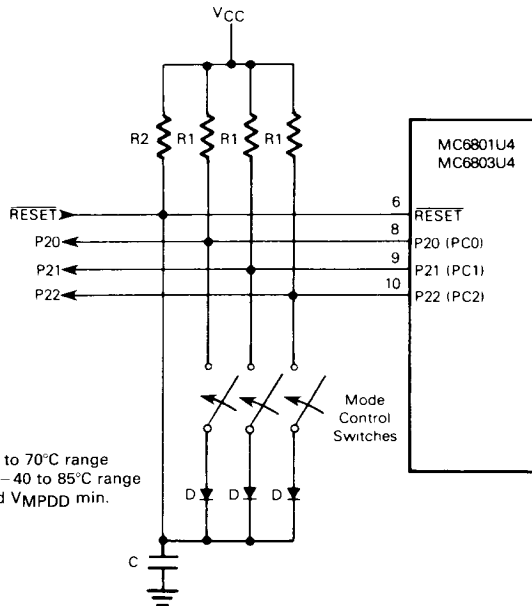
**LEGEND**

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

**NOTES:**

1. Addresses associated with ports 3 and 4 are considered external in modes 0, 2, and 3.
2. Addresses associated with port 3 are considered external in modes 1, 5, and 6.
3. Port 4 default is user data input; address output is optional by writing to port 4 data direction register.
4. Mode 4 is a nonuser mode and should not be used as an operating mode.
5. Mode 3 has the internal RAM and internal registers relocated at \$D000-\$D0FF.

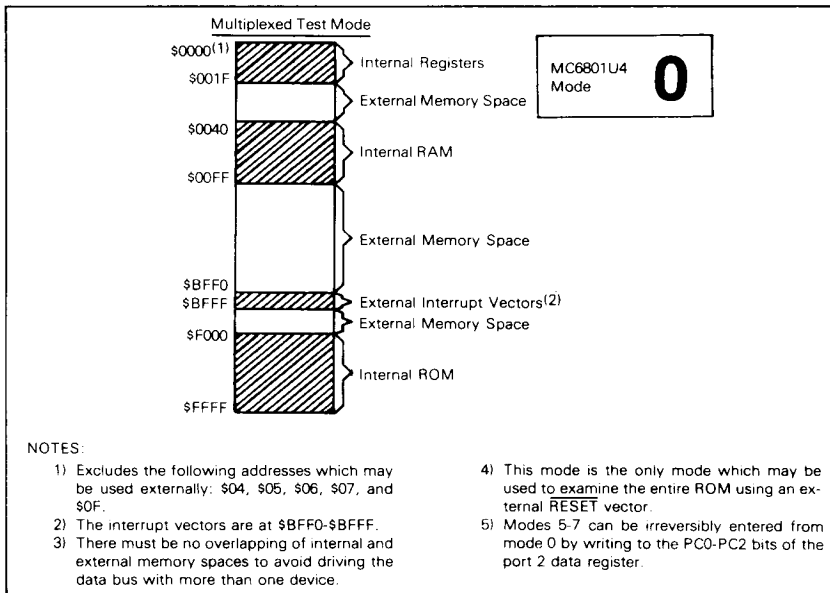
3



NOTES:

1. Mode 7 as shown
2.  $R2 \cdot C$  = Reset time constant
3.  $R1 = 10\text{ k}$  (typical)
4.  $D = 1N914, 1N4001$  in the 0 to 70°C range  
 $D = 1N270, MBD201$  in the -40 to 85°C range
5. Diode  $V_f$  should not exceed  $V_{MPDD}$  min.

Figure 15. Typical Mode Programming Circuit



NOTES:

- 1) Excludes the following addresses which may be used externally: \$04, \$05, \$06, \$07, and \$0F.
- 2) The interrupt vectors are at \$BFF0-\$BFFF.
- 3) There must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
- 4) This mode is the only mode which may be used to examine the entire ROM using an external RESET vector.
- 5) Modes 5-7 can be irreversibly entered from mode 0 by writing to the PC0-PC2 bits of the port 2 data register.

Figure 16. MC6801U4/MC6803U4 Memory Maps (Sheet 1 of 4)

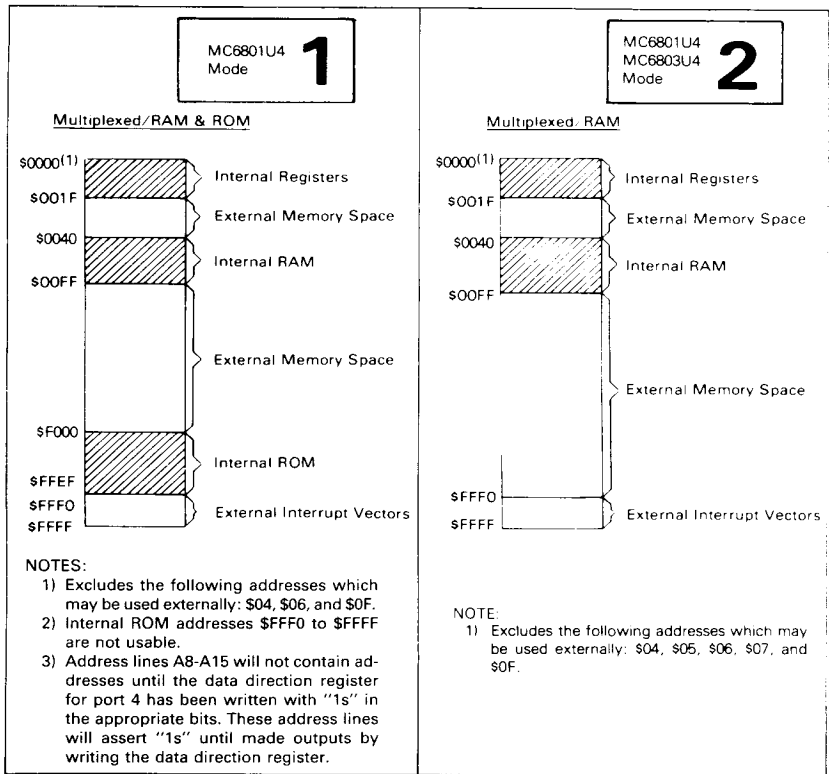
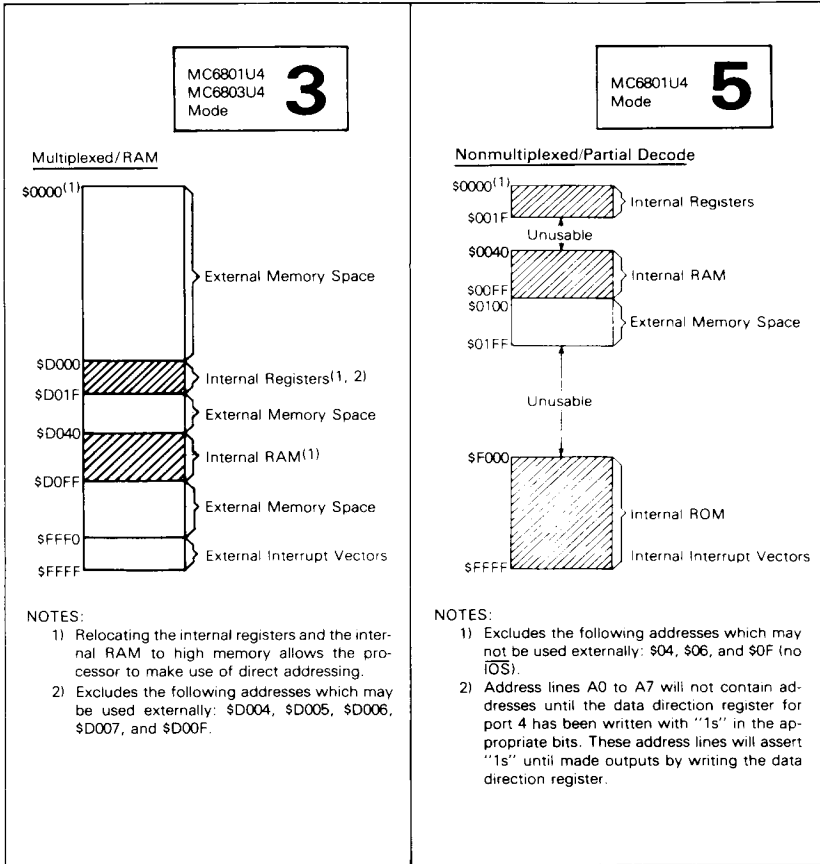


Figure 16. MC6801U4/MC6803U4 Memory Maps (Sheet 2 of 4)



3

Figure 16. MC6801U4/MC6803U4 Memory Maps (Sheet 3 of 4)

3

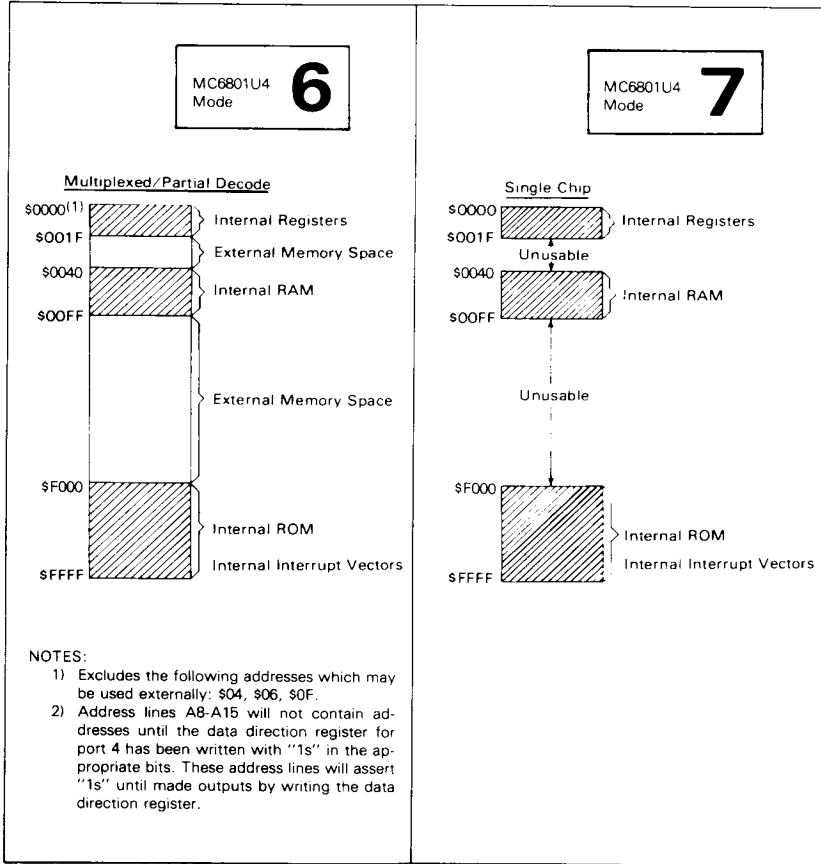


Figure 16. MC6801U4/MC6803U4 Memory Maps (Sheet 4 of 4)

Table 4. Internal Register Area

Register	Address	
	Other Modes	Mode 3
Port 1 Data Direction Register***	0000	D000
Port 2 Data Direction Register***	0001	D001
Port 1 Data Register	0002	D002
Port 2 Data Register	0003	D003
Port 3 Data Direction Register***	0004*	D004*
Port 4 Data Direction Register***	0005**	D005**
Port 3 Data Register	0006*	D006*
Port 4 Data Register	0007**	D007**
Timer Control and Status Register	0008	D008
Counter (High Byte)	0009	D009
Counter (Low Byte)	000A	D00A
Output Compare Register (High Byte)	000B	D00B
Output Compare Register (Low Byte)	000C	D00C
Input Capture Register (High Byte)	000D	D00D
Input Capture Register (Low Byte)	000E	D00E
Port 3 Control and Status Register	000F*	D00F*
Rate and Mode Control Register	0010	D010
Transmit/Receive Control and Status Register	0011	D011
Receive Data Register	0012	D012
Transmit Data Register	0013	D013
RAM Control Register	0014	D014
Counter Alternate Address (High Byte)	0015	D015
Counter Alternate Address (Low Byte)	0016	D016
Timer Control Register 1	0017	D017
Timer Control Register 2	0018	D018
Timer Status Register	0019	D019
Output Compare Register 2 (High Byte)	001A	D01A
Output Compare Register 2 (Low Byte)	001B	D01B
Output Compare Register 3 (High Byte)	001C	D01C
Output Compare Register 3 (Low Byte)	001D	D01D
Input Capture Register 2 (High Byte)	001E	D01E
Input Capture Register 2 (Low Byte)	001F	D01F

\* External addresses in modes 0, 1, 2, 3, 5, and 6 cannot be accessed in mode 5 (no IOS).  
 \*\* External Addresses in Modes 0, 2, and 3.  
 \*\*\* 1 = Output, 0 = Input

MC6801U4/MC6803U4 INTERRUPTS

The M6801 Family supports two types of interrupt requests: maskable and nonmaskable. A nonmaskable interrupt (NMI) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the condition code register I bit and by individual enable bits. The I bit controls all maskable interrupts. Of the maskable interrupts, there are two types: IRQ1 and IRQ2. The programmable timer and serial communications interface use an internal IRQ2 interrupt line, as shown in the block diagram. External devices and IS3 use IRQ1. An IRQ1 interrupt is serviced before IRQ2 if both are pending.

NOTE

After reset, an NMI will not be serviced until the first program load of the stack pointer. Any NMI generated before this load will be remembered by the processor and serviced subsequent to the stack pointer load.

All IRQ2 interrupts use hardware-prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order, and each is vectored to a separate location. All interrupt-vector locations are shown in Table 5. In mode 0, reset and interrupt vectors are defined as \$BFF0-\$BFFF.

The interrupt flowchart, which is depicted in Figure 17, is common to every interrupt excluding reset. During interrupt servicing, 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 RESET timing are illustrated in Figures 18 and 19.

3

Table 5. MCU Interrupt-Vector Locations

Mode 0		Modes 1-3, 5-7		Interrupt***
MSB	LSB	MSB	LSB	
BFFE	BFFF	FFFE	FFFF	RESET
BFFC	BFFD	FFFC	FFFD	Nonmaskable Interrupt**
BFFA	BFFB	FFFA	FFFB	Software Interrupt
BFF8	BFF9	FFF8	FFF9	Maskable Interrupt Request 1
BFF6	BFF7	FFF6	FFF7	Input Capture Flag*
BFF4	BFF5	FFF4	FFF5	Output Compare Flag*
BFF2	BFF3	FFF2	FFF3	Timer Overflow Flag*
BFF0	BFF1	FFF0	FFF1	Serial Communications Interface*

\* IRQ2 interrupt  
 \*\* NMI must be armed (by accessing stack pointer) before an NMI is executed.  
 \*\*\* Mode 4 interrupt vectors are undefined.





## FUNCTIONAL PIN DESCRIPTIONS

**VCC AND VSS**

VCC and VSS provide power to a large portion of the MCU. The power supply should provide +5 volts ( $\pm 5\%$ ) to VCC, and VSS should be tied to ground. Total power dissipation (including VCC standby) will not exceed Pd milliwatts.

**VCC STANDBY**

VCC standby provides power to the standby portion (\$40 through \$5F in all modes except mode 3, which is \$D040 through \$D05F) of the RAM, and the STBY PWR and RAME bits of the RAM control register. Voltage requirements depend on whether the device is in a powerup or powerdown state. In the powerup state, the power supply should provide +5 volts ( $\pm 5\%$ ) and must reach VSB volts before RESET reaches 4.0 volts. During powerdown, VCC standby must remain above VSB (minimum) to sustain the standby RAM and STBY PWR bit. While in powerdown operation, the standby current will not exceed ISBB.

It is typical to power both VCC and VCC standby from the same source during normal operation. A diode must be used between them to prevent supply power to VCC during powerdown operation.

**XTAL AND EXTAL**

These two input pins interface either a crystal or TTL-compatible clock to the MCU internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58-MHz or 4.4336-MHz color-burst TV crystals. A 20-pF capacitor should be tied from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL may be driven by an external TTL-compatible clock at  $4 f_0$  with a duty cycle of 50% ( $\pm 5\%$ ) with XTAL connected ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator operated in parallel resonance mode in the frequency range specified for fXTAL. 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. Nominal crystal parameters are shown in Figure 20.

**RESET**

This input is used to reset the internal state of the device and provide an orderly startup procedure. During powerup, RESET must be held below 0.8 volts: 1) at least tRC after VCC reaches 4.75 volts to provide sufficient time for the clock generator to stabilize, and 2) until VCC standby reaches 4.75 volts. RESET must be held low at least three E cycles if asserted during powerup operation.

**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 device input clock frequency. It will drive one Schottky TTL load and 90 pF, and all data

given in cycles are referenced to this clock unless otherwise noted.

**NMI (NONMASKABLE INTERRUPT)**

An NMI negative edge requests 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 (\$BFFC and \$BFFD in mode 0), transferred to the program counter, and instruction execution is resumed. NMI typically requires a 3.3 k $\Omega$  (nominal) resistor to VCC. There is no internal NMI pullup resistor. NMI must be held low for at least one E cycle to be recognized under all conditions.

**NOTE**

After reset, an NMI will not be serviced until the first program load of the stack pointer. Any NMI generated before this load will remain pending by the processor.

**IRQ1 (MASKABLE INTERRUPT REQUEST 1)**

IRQ1 is a level-sensitive input that 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. A vector is fetched from \$FFF8 and \$FFF9 (\$BFF8 and \$BFF9 in mode 0), transferred to the program counter, and instruction execution is resumed.

IRQ1 typically requires an external 3.3 k $\Omega$  (nominal) resistor to VCC for wire-OR applications. IRQ1 has no internal pullup resistor.

**SC1 AND SC2 (STROBE CONTROL 1 AND 2)**

The function of SC1 and SC2 depends on the operating mode. SC1 is configured as an output in all modes except single-chip mode; whereas, SC2 is always an output. SC1 and SC2 can drive one Schottky load and 90 pF.

**SC1 and SC2 in Single-Chip Mode**

In single-chip mode, SC1 and SC2 are configured as an input and output, respectively, and both function as port 3 control lines. SC1 functions as IS3 and can be used to indicate that port 3 input data is ready or output data has been accepted. Three options associated with IS3 are controlled by the port 3 control and status register and are discussed in the port 3 description; refer to P30-P37 (PORT 3). If unused, IS3 can remain unconnected.

SC2 is configured as OS3 and can be used to strobe output data or acknowledge input data. It is controlled by output strobe select (OSS) in the port 3 control and status register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the port 3 data register. OS3 timing is shown in Figure 3.

**SC1 and SC2 in Expanded-Nonmultiplexed Mode**

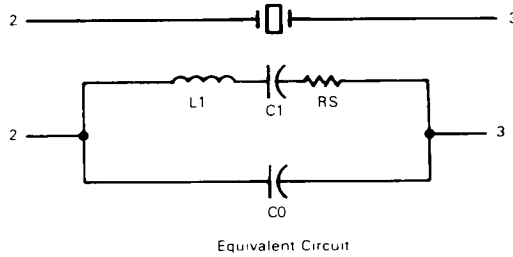
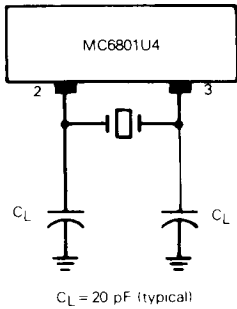
In the expanded-nonmultiplexed mode, both SC1 and SC2 are configured as outputs. SC1 functions as input/output select (IOS) and is asserted only when \$0100 through \$01FF is sensed on the internal address bus.

# MC6801U4/6803U4

## (a) Nominal Recommended Crystal Parameters

Nominal Crystal Parameters*			
	3.58 MHz	4.00 MHz	5.0 MHz
RS	60 $\Omega$	50 $\Omega$	30-50 $\Omega$
C0	3.5 pF	6.5 pF	4-6 pF
C1	0.015 pF	0.025 pF	0.01-0.02 pF
Q	> 40 K	> 30 K	> 20 K

\*NOTE: These are representative AT-cut crystal parameters only. Crystals of other types of cut may also be used.



**NOTE**  
TTL-compatible oscillators may be obtained from:  
Motorola Component Products  
Attn: Crystal Clock Oscillators  
2553 N. Edgington St.  
Franklin Park, IL 60131  
Tel: 312-451-1000  
Telex: 433-0067

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

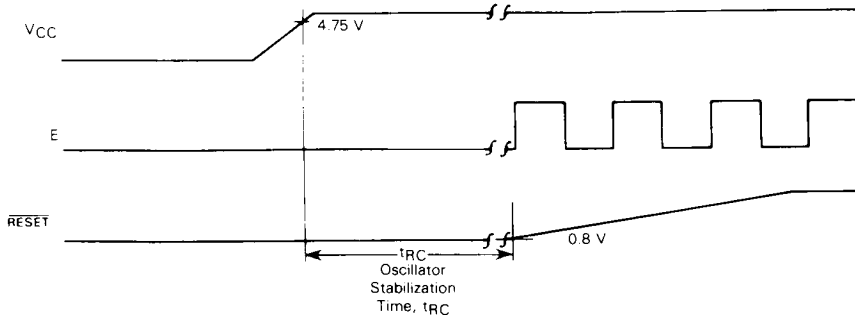


Figure 20. MC6801U4/MC6803U4 Family Oscillator Characteristics

3

SC2 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.

**SC1 and SC2 in Expanded-Multiplexed Mode**

In the expanded-multiplexed modes, both SC1 and SC2 are configured as outputs. SC1 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 the lower address on the negative edge, as shown Figure 13.

SC2 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.

**P10-P17 (PORT 1)**

Port 1 is a mode-independent 8-bit I/O and timer port. Each line can be configured as either an input or output as defined by the port 1 data direction register. Port 1 bits 0, 1, and 2 (P10, P11, and P12) can also be used to exercise one input edge function and two output compare functions of the timer. The TTL-compatible three-state buffers can drive one Schottky TTL load and 30 pF, Darlington transistors or CMOS devices using external pull-up resistors. It is configured as a data input port during RESET. Unused pins can remain unconnected.

**P20-P24 (PORT 2)**

Port 2 is a mode-independent 5-bit multipurpose I/O port. The voltage levels present on P20, P21, and P22 on the rising edge of RESET determine the operating mode of the MCU. The entire port is then configured as a data input port. The port 2 lines can be selectively configured as data output lines by setting the appropriate bits in the port 2 data direction register. The port 2 data register is used to move data through the port. However, if P21 is configured as an output, it is tied to the timer output compare 1 function and cannot be used to provide output from the port 2 data register unless output enable 1 (OE1) is cleared in timer control register 1.

Port 2 can also be used to provide an interface for the serial communications interface and the timer input edge function. These configurations are described in SERIAL COMMUNICATIONS INTERFACE and PROGRAMMABLE TIMER.

The port 2 three-state TTL-compatible output buffers are capable of driving one Schottky TTL load and 30 pF or CMOS devices using external pullup resistors.

**PORT 2 DATA REGISTER**

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

**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 the single-chip mode with each line configured by the port 3 data direction register. There are also two lines, IS3 and 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 are available only in single-chip mode: 1) port 3 input data can be latched using IS3 as a control signal, 2) OS3 can be generated by either an MPU read or write to the port 3 data register, and 3) an IRQ1 interrupt can be enabled by an IS3 negative edge. Port 3 latch timing is shown in Figure 4.

**PORT 3 CONTROL AND STATUS REGISTER**

	7	6	5	4	3	2	1	0	
IC3 Flag	IS3 IRQ1	X	OSS	Latch Enable	X	X	X		SOF

Bits 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 IS3 negative edge. The latch is transparent after a read of the port 3 data register. Latch enable is cleared during reset.

**Bit 4 Output Strobe Select (OSS)** — This bit determines whether OS3 will be generated by a read or write of the port 3 data register. When clear, the strobe is generated by a read; when set, it is generated by a write. OSS is cleared during reset.

Bit 5 Not used.

**Bit 6 IS3 IRQ1 Enable** — When set, an IRQ1 interrupt will be enabled whenever the IS3 flag is set; when clear, the interrupt is inhibited. This bit is cleared during reset.

**Bit 7 IS3 Flag** — This read-only status bit is set by an IS3 negative edge. It is cleared by a read of the port 3 data register or during reset.

**Port 3 in Expanded-Nonmultiplexed Mode**

Port 3 is configured as a bidirectional data bus (D7-D0) in the expanded-nonmultiplexed mode. The direction of data transfers is controlled by read/write (SC2). Data is clocked by E (enable).

**Port 3 in Expanded-Multiplexed Mode**

Port 3 is configured as a time-multiplexed address (A7-A0), and data bus (D7-D0) in the expanded-multiplexed mode where 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 bus conflicts.

**P40-P47 (PORT 4)**

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating



mode. Port 4, which can drive one Schottky TTL load and 90 pF, 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 with each line configured by the port 4 data direction register. Internal pullup resistors allow the port to directly interface with CMOS at 5-volt levels. However, external pullup resistors to more than 5 volts cannot be used.

**Port 4 in Expanded-Nonmultiplexed Mode**

Port 4 is configured from reset as an 8-bit input port where the port 4 data direction register can be written to provide any or all of eight address lines A0 to A7. Internal pullup resistors pull the lines high until the port 4 data direction register is configured.

**Port 4 in Expanded-Multiplexed Mode**

In all expanded-multiplexed modes except modes 1 and 6, port 4 functions as half of the address bus and provides A8 to A15. In modes 1 and 6, the port is configured from reset as an 8-bit parallel input port where the port 4 data direction register can be written to provide any or all of upper address lines A8 to A15. Internal pullup resistors pull the lines high until the port 4 data direction register is configured where bit 0 controls A8.

**RESIDENT MEMORY**

The MC6801U4 provides 4096 bytes of on-chip ROM and 192 bytes of on-chip RAM.

Thirty-two bytes of the RAM are powered through the VCC standby pin and are maintainable during VCC power-down. This standby portion of the RAM consists of 32 bytes located from \$40 through \$5F in all modes except mode 3, which is \$D040 through \$D05F.

Power must be supplied to VCC standby if the internal RAM is to be used, regardless of whether standby power operation is anticipated.

The RAM is controlled by the RAM control register.

**RAM CONTROL REGISTER (\$14)**

The RAM control register includes two bits, which can be used to control RAM accesses and to determine the adequacy of the standby power source during power-down 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	X	\$14

Bits 0-5 Not used.

Bit 6 **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  $\overline{\text{RESET}}$ . If RAME is clear, any access to a RAM address is external. If RAME is set, the RAM is included in the internal map.

Bit 7 **Standby Power** — This bit is a read/write status bit that, when cleared, indicates VCC standby has decreased sufficiently below V<sub>SBB</sub> (minimum) to make data in the standby RAM suspect. It can be set only by software and is not affected during reset.

**PROGRAMMABLE TIMER**

The programmable timer can be used to perform measurements on two separate input waveforms while independently generating three output waveforms. Pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in Figure 21.

**COUNTER (\$09:0A), (\$15, \$16)**

The key timer element is a 16-bit free-running counter that is incremented by E (enable). It is cleared during reset and is read-only with one exception: in mode 0 a write to the counter (\$09) will configure it to \$FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. The TOF is set whenever the counter contains all ones. If ETOI is set, an interrupt will occur when the TOF is set. The counter may also be read as \$15 and \$16 to avoid inadvertently clearing the TOF.

**OUTPUT COMPARE REGISTERS (\$0B:0C), (\$1A:1B), (\$1C:1D)**

The three output compare registers are 16-bit read/write registers, each used to control an output waveform or provide an arbitrary time-out flag. They are compared with the free-running counter during the negative half of each E cycle. When a match occurs, the corresponding output compare flag (OCF) is set, and the corresponding output level (OLVL) is clocked to an output level register. If both the corresponding output enable bit and data direction register bit are set, the value represented in the output level register will appear on the corresponding port pin. The appropriate OLVL bit can then be changed for the next compare.

The function is inhibited for one cycle after a write to its high byte (\$0B, \$1A, or \$1C) to ensure a valid compare after a double-byte write. Writes can be made to either byte of the output compare register without affecting the other byte. The OLVL value will be clocked out independently of whether the OCF had previously been cleared. The output compare registers are set to \$FFFF during reset.

**INPUT CAPTURE REGISTERS (\$0D:0E), (\$1E:1F)**

The two input capture registers are 16-bit read-only registers used to store the free-running counter when a "proper" input transition occurs as defined by the corresponding input edge bit (IEDG1 or IEDG2). The input pin's data direction register should be configured as an



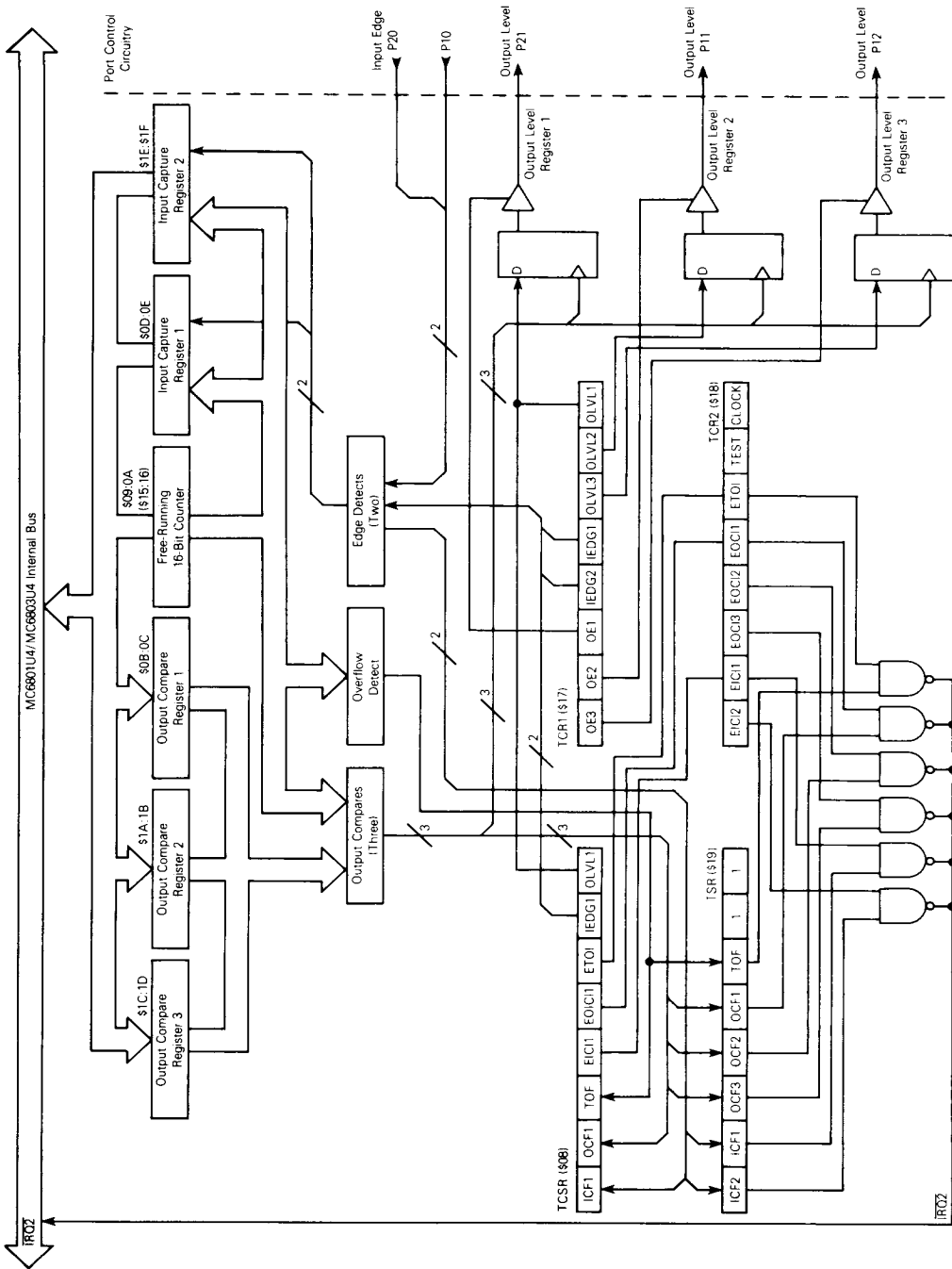


Figure 21. Block Diagram of Programmable Timer

3

input, but the edge detect circuit always senses P10 and P20 even when configured as an output. The counter value will be latched into the into capture registers on the second negative edge of the E clock following the transition.

As input capture can occur independently of ICF; the register always contains the most current value. However, counter transfer is inhibited between accesses of a double-byte MPU read. The input pulse width must be at least two E cycles to ensure an input capture under all conditions.

**TIMER CONTROL AND STATUS REGISTERS**

Four registers are used to provide the MC6801U4/MC6803U4 with control and status information about the three output compare functions, the timer overflow function, and the two input edge functions of the timer. They are as follows:

- Timer Control and Status Register (TCSR)
- Timer Control Register 1 (TCR1)
- Timer Control Register 2 (TCR2)
- Timer Status Register (TSR)

**Timer Control and Status Register (TCSR) (\$08)**

The timer control and status register is an 8-bit register in which all bits are readable, while only bits 0-4 can be written. All the bits in this register are also accessible through the two timer control registers and the timer status register. The three most significant bits provide the timer status and indicate if

1. a proper level transition has been detected at P20;
2. a match has occurred between the free-running counter and output compare register 1; or
3. the free-running counter has overflowed.

Each of the three events can generate an IRQ2 interrupt and is controlled by an individual enable bit in the TCSR.

**TIMER CONTROL AND STATUS REGISTER**

7	6	5	4	3	2	1	0	
ICF1	OCF1	TOF	EIC1	EOC1	ETOI	IEDG1	OLVL1	\$08

**Bit 0 Output Level 1** — OLVL1 is clocked to output level register 1 by a successful output compare and will appear at P21 if bit 1 of the port 2 data direction register is set and the OE1 control bit in timer control register 1 is set. OLVL1 and output level register 1 are cleared during reset. Refer to **TIMER CONTROL REGISTER 1 (TCR1) (\$17)**.

**Bit 1 Input Edge 1** — IEDG1 is cleared during reset and controls which level transition on P20 will trigger a counter transfer to input capture register 1:  
 IEDG1 = 0 transfer on a negative-edge  
 IEDG1 = 1 transfer on a positive-edge  
 Refer to **TIMER CONTROL REGISTER 1 (TCR1) (\$17)**.

**Bit 2 Enable Timer Overflow Interrupt** — When set, an IRQ2 interrupt will be generated when the timer overflow flag is set; when clear, the interrupt is

inhibited. ETOI is cleared during reset. Refer to **TIMER CONTROL REGISTER 2 (TCR2) (\$18)**.

**Bit 3 Enable Output Compare Interrupt 1** — When set, an IRQ2 interrupt will be generated when output compare flag 1 is set; when clear, the interrupt is inhibited. EOC1 is cleared during reset. Refer to **TIMER CONTROL REGISTER 2 (TCR2) (\$18)**.

**Bit 4 Enable Output Capture Interrupt 1** — When set, an IRQ2 interrupt will be generated when input capture flag 1 is set; when clear, the interrupt is inhibited. EIC1 is cleared during reset. Refer to **TIMER CONTROL REGISTER 2 (TCR2) (\$18)**.

**Bit 5 Timer Overflow Flag** — The TOF is set when the counter contains all ones (\$FFFF). It is cleared by reading the TCSR or the TSR (with TOF set) and the counter high byte (\$09), or during reset. Refer to **TIMER STATUS REGISTER (TSR) (\$19)**.

**Bit 6 Output Compare Flag 1** — OCF1 is set when output compare register 1 matches the free-running counter. OCF1 is cleared by reading the TCSR or the TSR (with OCF1 set) and then writing to output compare register 1 (\$0B or \$0C), or during reset. Refer to **TIMER STATUS REGISTER (TSR) (\$19)**.

**Bit 7 Input Capture Flag** — ICF1 is set to indicate that a proper level transition has occurred; it is cleared by reading the TCSR or the TSR (with ICF1 set) and the input capture register 1 high byte (\$0D), or during reset. Refer to **TIMER STATUS REGISTER (TSR) (\$19)**.

**Timer Control Register 1 (TCR1) (\$17)**

Timer control register 1 is an 8-bit read/write register containing the control bits for interfacing the output compare and input capture registers to the corresponding I/O pins.

**TIMER CONTROL REGISTER 1**

7	6	5	4	3	2	1	0	
OE3	OE2	OE1	IEDG2	IEDG1	OLVL3	OLVL2	OLVL1	\$17

**Bit 0 Output Level 1** — OLVL1 is clocked to output level register 1 by a successful output compare and will appear at P21 if bit 1 of the port 2 data direction register is set and the OE1 control bit is set. OLVL1 and output level register 1 are cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) (\$08)**.

**Bit 1 Output Level 2** — OLVL2 is clocked to output level register 2 by a successful output compare and will appear at P11 if bit 1 of port 1 data direction register is set and the OE2 control bit is set. OLVL2 and output level register 2 are cleared during reset.



- Bit 2 **Output Level 3** — OLVL3 is clocked to output level register 3 by a successful output compare and will appear at P12 if bit 2 of port 1 data direction register is set and the OE3 control bit is set. OLVL3 and output level register 3 are cleared during reset.
- Bit 3 **Input Edge 1** — IEDG1 is cleared during reset and controls which level transition on P20 will trigger a counter transfer to input capture register 1.  
 IEDG1 = 0 transfer on a negative edge  
 IEDG1 = 1 transfer on a positive edge  
 Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) (\$08)**.
- Bit 4 **Input Edge 2** — IEDG2 is cleared during reset and controls which level transition on P10 will trigger a counter transfer to input capture register 2.  
 IEDG2 = 0 transfer on a negative edge  
 IEDG2 = 1 transfer on a positive edge
- Bit 5 **Output Enable 1** — OE1 is set during reset and enables the contents of output level register 1 to be connected to P21 when bit 1 of port 2 data direction register is set.  
 OE1 = 0 port 2 bit 1 data register output  
 OE1 = 1 output level register 1
- Bit 6 **Output Enable 2** — OE2 is cleared during reset and enables the contents of output level register 2 to be connected to P11 when bit 1 of port 1 data direction register is set.  
 OE2 = 0 port 1 bit 1 data register output  
 OE2 = 1 output level register 2
- Bit 7 **Output Enable 3** — OE3 is cleared during reset and enables the contents of output level register 3 to be connected to P12 when bit 2 of port 1 data direction register is set  
 OE3 = 0 port 1 bit 2 data register output  
 OE3 = 1 output level register 3

**Timer Control Register 2 (TCR2) (\$18)**

Timer control register 2 is an 8-bit read/write register (except bits 0 and 1), which enables the interrupts associated with the free-running counter, the output compare registers, and the input capture registers. In test mode 0, two more bits (clock and test) are available for checking the timer.

**TIMER CONTROL REGISTER 2  
(Nontest Modes)**

7	6	5	4	3	2	1	0	
EIC12	EIC11	EOC13	EOC12	EOC11	ETOI	1	1	\$18

- Bits 0-1 **Read-Only Bits** — When read, these bits return a value of 1. Refer to **TIMER CONTROL REGISTER 2 (Test Mode)**.
- Bit 2 **Enable Timer Overflow Interrupt** — When set, an IRQ2 interrupt will be generated when the timer

overflow flag is set; when clear, the interrupt is inhibited. ETOI is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) (\$08)**.

- Bit 3 **Enable Output Compare Interrupt 1** — When set, an IRQ2 interrupt will be generated when the output compare flag 1 is set; when clear, the interrupt is inhibited. EOC1 is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) (\$08)**.
- Bit 4 **Enable Output Compare Interrupt 2** — When set, an IRQ2 interrupt will be generated when the output compare flag 2 is set; when clear, the interrupt is inhibited. EOC2 is cleared during reset.
- Bit 5 **Enable Output Compare Interrupt 3** — When set, an IRQ2 interrupt will be generated when the output compare flag 3 is set; when clear, the interrupt is inhibited. EOC3 is cleared during reset.
- Bit 6 **Enable Input Capture Interrupt 1** — When set, an IRQ2 interrupt will be generated when the input capture flag 1 is set; when clear, the interrupt is inhibited. EIC1 is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) (\$08)**.
- Bit 7 **Enable Input Capture Interrupt 2** — When set, an IRQ2 interrupt will be generated when the input capture flag 2 is set; when clear, the interrupt is inhibited. EIC2 is cleared during reset.

The timer test bits (test and clock) allow the free-running counter to be tested as two separate 8-bit counters to speed testing.

**TIMER CONTROL REGISTER 2  
(Test Modes)**

7	6	5	4	3	2	1	0	
EIC12	EIC11	EOC13	EOC12	EOC11	ETOI	TEST	CLOCK	\$18

- Bit 0 **CLOCK** — The CLOCK control bit selects which half of the 16-bit free-running counter (MSB or LSB) should be clocked with E. The CLOCK bit is a read/write bit only in mode 0 and is set during reset.  
 CLOCK = 0 — Only the eight most-significant bits of the free-running counter run with TEST = 0.  
 CLOCK = 1 — Only the eight least-significant bits of the free-running counter run when TEST = 0.
- Bit 1 **TEST** — The TEST control bit enables the timer test mode. TEST is a read/write bit in mode 0 and is set during reset.  
 TEST = 0 — Timer test mode enabled:
  - a) The timer LSB latch is transparent, which allows the LSB to be read independently of the MSB.



- b) Either the MSB or the LSB of the timer is clocked by E, as defined by the CLOCK bit.

TEST = 1 — Timer test mode disabled.

Bits 2-7 See **TIMER CONTROL REGISTER 2 (Nontest Modes)**. (These bits function the same as in the nontest modes.)

**Timer Status Register (TSR) (\$19)**

The timer status register is an 8-bit read-only register which contains the flags associated with the free-running counter, the output compare registers, and the input capture registers.

**TIMER STATUS REGISTER**

7	6	5	4	3	2	1	0	
ICF2	ICF1	OCF3	OCF2	OCF1	TOF	1	1	\$19

Bits 0-1 Not used.

- Bit 2 **Timer Overflow Flag** — The TOF is set when the counter contains all ones (\$FFFF). It is cleared by reading the TSR or the TCSR (with TOF set) and then the counter high byte (\$09), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) (\$08)**.
- Bit 3 **Output Compare Flag 1** — OCF1 is set when output compare register 1 matches the free-running counter. OCF1 is cleared by reading the TSR or the TCSR (with OCF1 set) and then writing to output compare register 1 (\$0B or \$0C), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) (\$08)**.
- Bit 4 **Output Compare Flag 2** — OCF2 is set when output compare register 2 matches the free-running counter. OCF2 is cleared by reading the TSR (with OCF2 set) and then writing to output compare register 2 (\$1A or \$1B), or during reset.
- Bit 5 **Output Compare Flag 3** — OCF3 is set when output compare register 3 matches the free-running counter. OCF3 is cleared by reading the TSR (with OCF3 set) and then writing to output compare register 3 (\$1C or \$1D), or during reset.
- Bit 6 **Input Capture Flag 1** — ICF1 is set to indicate that a proper level transition has occurred; it is cleared by reading the TSR or the TCSR (with ICF1 set) and the input capture register 1 high byte (\$0D), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) (\$08)**.
- Bit 7 **Input Capture Flag 2** — ICF2 is set to indicate that a proper level transition has occurred; it is cleared by reading the TSR (with ICF2 set) and the input capture register 2 high byte (\$1E), or during reset.

**SERIAL COMMUNICATIONS INTERFACE**

A full-duplex asynchronous SCI is provided with two data formats and a variety of rates. The SCI transmitter and receiver are functionally independent but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and biphase; both provide 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 multiprocessor configuration, the software protocol will usually identify the address-see(s) at the beginning of the message. To permit uninterested MPUs to ignore the remainder of the message, 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 eleven consecutive ones or during reset. Software must provide for the required idle string between consecutive messages and must prevent it within messages.

**PROGRAMMABLE OPTIONS**

The following features of the SCI are programmable:

- Format: standard mark/space (NRZ) or biphase
- Clock: external or internal bit rate clock
- Baud: one of eight per E clock frequency or external clock (x 8 desired baud)
- Wake-Up Feature: enabled or disabled
- Interrupt Requests: enabled individually for transmitter and receiver
- Clock Output: internal bit rate clock enabled or disabled to P22

**SERIAL COMMUNICATIONS REGISTERS**

The SCI includes four addressable registers as depicted in Figure 22. It is controlled by the rate and mode control register and the transmit/receive control and status register. Data are transmitted and received utilizing a write-only transmit register and a read-only receive register. The shift registers are not accessible to software.

**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 P22. The register consists of five write-only bits which are cleared during reset. The two least-significant bits, in conjunction with bit 7, control the bit rate of the internal clock, and the remaining two bits control the format and clock source.

**RATE AND MODE CONTROL REGISTER**

7	6	5	4	3	2	1	0	
EBE	X	X	X	CC1	CC0	SS1	SS0	\$10

Bit 1:Bit 0 **SS1:SS0 Speed Select** — These two bits select the baud when using the internal clock. Eight



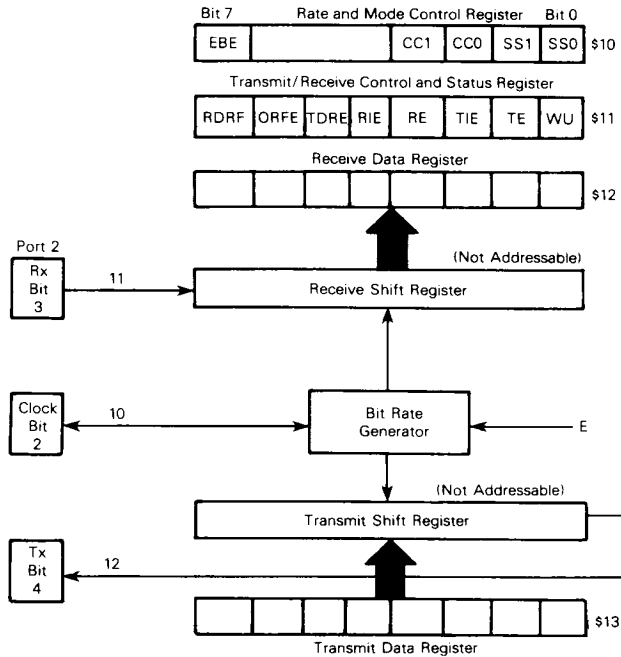


Figure 22. SCI Registers

rates may be selected (in conjunction with bit 7) which are a function of the MCU input frequency. Table 6 lists bit times and rates for three selected MCU frequencies.

**Bit 3:Bit 2 CC1:CC0 Clock Control and Format Select** — These two bits control the format and select the serial clock source. If CC1 is set, the DDR value for P22 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 formats, clock source, and use of P22.

**Bits 4-6** Not used.

**Bit 7 EBE Enhanced Baud Enable** — EBE selects the standard MC6801 baud rates when clear and the additional baud rates when set (Table 6). This bit is cleared by reset and is a write-only control bit.

**Bit 1:Bit 1 EBE = 0** standard MC6801 baud rates  
**Bit 1:Bit 1 EBE = 1** additional baud rates

If both CC1 and CC0 are set, an external TTL-compatible clock must be connected to P22 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 P22 regardless of the values for TE or RE.

**NOTE**

The source of SCI internal bit rate clock is the timer free-running counter. An MPU write to the counter in mode 0 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; bits 0 to 4 are also writable. The register is initialized to \$20 by RESET.

**TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER**

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	\$11

**Bit 0 Wake-Up on Idle Line** — When set, WU enables the wake-up function; it is cleared by eleven consecutive ones or during reset. WU will not be set if the line is idle. Refer to **WAKE-UP FEATURE**.

**Bit 1 Transmit Enable** — When set, P24 DDR bit is set, cannot be changed, and will remain set if TE is



Table 6. SCI Bit Times and Rates

EBE	SS1:SS0		$4 f_0 \rightarrow$ E	2.4576 MHz		4.0 MHz		4.9152 MHz	
				614.4 kHz		1.0 MHz		1.2288 MHz	
				Baud	Time	Baud	Time	Baud	Time
0	0	0	+ 16	38400.0	26 $\mu$ s	62500.0	16.0 $\mu$ s	76800.0	113.0 $\mu$ s
0	0	1	- 128	4800.0	208.3 $\mu$ s	7812.5	128.0 $\mu$ s	9600.0	104.2 $\mu$ s
0	1	0	+ 1024	600.0	1.67 ms	976.6	1.024 ms	1200.0	833.3 $\mu$ s
0	1	1	- 4096	150.0	6.67 ms	244.1	4.096 ms	300.0	3.33 ms
1	0	0	+ 64	9600.0	104.2 $\mu$ s	15625.0	64 $\mu$ s	19200.0	52.0 $\mu$ s
1	0	1	- 256	2400.0	416.6 $\mu$ s	3906.3	256 $\mu$ s	4800.0	208.3 $\mu$ s
1	1	0	+ 512	1200.0	833.3 $\mu$ s	1953.1	512 $\mu$ s	2400.0	416.6 $\mu$ s
1	1	1	+ 2048	300.0	3.33 ms	488.3	2.05 ms	600.0	1.67 ms
External (P22) *				76800.0	13.0 $\mu$ s	125000.0	8.0 $\mu$ s	153600.0	6.5 $\mu$ s

\*Using maximum clock rate

Table 7. SCI Format and Clock Source Control

CC1:CC0	Format	Clock Source	Port 2 Bit 2
00	Biphase	Internal	Not Used
01	NRZ	Internal	Not Used
10	NRZ	Internal	Output
11	NRZ	External	Input

subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P24 and a preamble of nine consecutive ones is transmitted. TE is cleared during reset.

- Bit 2 **Transmit Interrupt Enable** — When set, an  $\overline{\text{IRQ2}}$  is set; when clear, the interrupt is inhibited. TE is cleared during reset.
- Bit 3 **Receive Enable** — When set, the P23 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 during reset.
- Bit 4 **Receiver Interrupt Enable** — When set, an  $\overline{\text{IRQ2}}$  interrupt is enabled when RDRF and/or ORFE is set; when clear, the interrupt is inhibited. RIE is cleared during reset.
- Bit 5 **Transmit Data Register Empty** — TDRE is set when the transmit data register is transferred to the output serial shift register or during reset. 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 **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 state of RDRF: if RDRF is set, then an overrun

has occurred; otherwise, a framing error has been detected. Data are not transferred to the receive data register in an overrun condition. Unframed data causing a framing error are transferred to the receive data register. However, subsequent data transfer is blocked until the framing error flag is cleared. ORFE is cleared by reading the TRCSR (with ORFE set) then the receive data register, or during reset.

Bit 7 **Receive Data Register Full** — RDRF is set when the input serial shift register is transferred to the receive data register, or during reset.

#### 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. When TE is set, the output of the transmit serial shift register is connected to P24, and serial output is initiated by transmitting a 9-bit preamble of ones.

At this point, one of two situations exists: 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 will be transferred to the output serial shift register (synchronized with the bit rate clock), TDRE will be set, and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0), and a stop bit (1) will be transmitted. If TDRE is still set when the next byte transfer occurs, ones will be sent until more data is provided. In biphase format, the output toggles at the start of each bit and at half-bit time when a one is sent. Receive operation is controlled by RE, which configures P23 as an input and enables the receiver. SCI data formats are illustrated in Figure 23.

#### INSTRUCTION SET

The MC6801U4/MC6803U4 is directly source compatible with the MC6801 and upward source and object code compatible with the MC6800. Execution times of key instructions have been reduced, and several instructions have been added, including a hardware multiply. A list

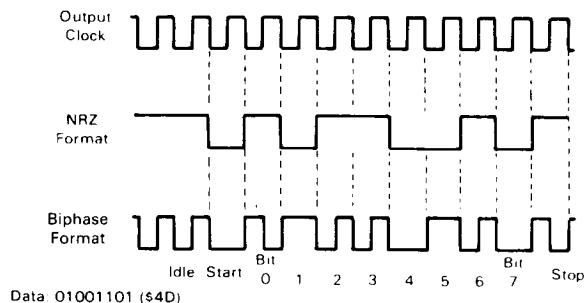


Figure 23. SCI Data Formats

of new operations added to the MC6800 instruction set is shown in Table 1.

In addition, two 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.

The coding of the first (or only) byte corresponding to an executable instruction is sufficient to identify the instruction and the addressing mode. The hexadecimal equivalents of the binary codes, which result from the translation of the 82 instructions in all valid modes of addressing, are shown in Table 8. There are 220 valid machine codes, 34 unassigned codes, and 2 codes reserved for test purposes.

### PROGRAMMING MODEL

A programming model for the MC6801U4 MC6803U4 is shown in Figure 8. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most-significant byte. Any operation that modifies the double accumulator will also modify accumulators 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 that can be used to store data or provide an address for the indexed mode of addressing.

#### Accumulators

The MPU 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 Register

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 instructions. 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

Six addressing modes can be used to reference memory. A summary of addressing modes for all instructions is presented in Table 9, 10, 11, and 12; execution times are provided in E cycles. Instruction execution times are summarized in Table 13. With an input frequency of 4 MHz, one E cycle is equivalent to one microsecond. A cycle-by-cycle description of bus activity for each instruction is provided in Table 14; descriptions of selected instructions are shown in Figure 24.

#### 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 Address

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.

Table 8. CPU Instruction Map

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#
00	*				34	DES	INHER	3	1	68	ASL	INDXD	6	2	9C	CPX	DIR	5	2	D0	SUBB	DIR	3	2
01	NOP	INHER	2	1	35	TXS		3	1	69	ROL		6	2	9D	JSR		5	2	O1	CMPB		3	2
02	*				36	PSHA		3	1	6A	DEC		6	2	9E	LDS		4	2	O2	SBCB		3	2
03	*				37	PSHB		3	1	6B	*				9F	STS	DIR	4	2	O3	ADDD		5	2
04	LSRD		3	1	38	PULX		5	1	6C	INC		6	2	A0	SUBA	INDXD	4	2	D4	ANDB		3	2
05	ASLD		3	1	39	RTS		5	1	6D	TST		6	2	A1	CMPA		4	2	D5	BITB		3	2
06	TAP		2	1	3A	ABX		3	1	6E	JMP		3	2	A2	SBCA		4	2	D6	LDAB		3	2
07	TPA		2	1	3B	RTI		10	1	6F	CLR	INDXD	6	2	A3	SUBD		6	2	D7	STAB		3	2
08	INX		3	1	3C	PSHX		4	1	70	NEG	EXTND	6	3	A4	ANDA		4	2	D8	EORB		3	2
09	DEX		3	1	3D	MUL		10	1	71	*				A5	BITA		4	2	D9	ADCB		3	2
0A	CLV		2	1	3E	WAI		9	1	72	*				A6	LDAA		4	2	DA	ORAB		3	2
0B	SEV		2	1	3F	SWI		12	1	73	COM		6	3	A7	STAA		4	2	DB	ADDB		3	2
0C	CLC		2	1	40	NEGA		2	1	74	LSR		6	3	A8	EORA		4	2	DC	LDD		4	2
0D	SEC		2	1	41	*				75	*				A9	ADCA		4	2	DD	STD		4	2
0E	CLI		2	1	42	*				76	ROR		6	3	AA	ORAA		4	2	DE	LDX		4	2
0F	SEI		2	1	43	COMA		2	1	77	ASR		6	3	AB	ADDA		4	2	DF	STX	DIR	4	2
10	SBA		2	1	44	LSRA		2	1	78	ASL		5	3	AC	CPX		6	2	E0	SUBB	INDXD	4	2
11	CBA		2	1	45	*				79	ROL		6	3	AD	JSR		6	2	E1	CMPB		4	2
12	*				46	RORA		2	1	7A	DEC		6	3	AE	LDS		5	2	E2	SBCB		4	2
13	*				47	ASRA		2	1	7B	*				AF	STS	INDXD	5	2	E3	ADDD		6	2
14	*				48	ASLA		2	1	7C	INC		6	3	B0	SUBA	EXTND	4	3	E4	ANDB		4	2
15	*				49	ROLA		2	1	7D	TST		6	3	B1	CMPA		4	3	E5	BITB		4	2
16	TAB		2	1	4A	DECA		2	1	7E	JMP		3	3	B2	SBCA		4	3	E6	LDAB		4	2
17	TBA		2	1	4B	*				7F	CLR	EXTND	6	3	B3	SUBD		6	3	E7	STAB		4	2
18	*				4C	INCA		2	1	80	SUBA	IMMED	2	2	B4	ANDA		4	3	E8	EORB		4	2
19	DAA	INHER	2	1	4D	TSTA		2	1	81	CMPA		2	2	B5	BITA		4	3	E9	ADCB		4	2
1A	*				4E	T				82	SBCA		2	2	B6	LDAA		4	3	EA	ORAB		4	2
1B	ABA	INHER	2	1	4F	CLRA		2	1	83	SUBD		4	3	B7	STAA		4	3	EB	ADDB		4	2
1C	*				50	NEGB		2	1	84	ANDA		2	2	B8	EORA		4	3	EC	LDD		5	2
1D	*				51	*				85	BITA		2	2	B9	ADCA		4	3	ED	STD		5	2
1E	*				52	*				86	LDAA		2	2	8A	ORAA		4	3	EE	LDX		5	2
1F	*				53	COMB		2	1	87	*				8B	ADDA		4	3	EF	STX	INDXD	5	2
20	BRA	REL	3	2	54	LSRB		2	1	88	EORA		2	2	8C	CPX		6	3	F0	SUBB	EXTND	4	3
21	BRN		3	2	55	*				89	ADCA		2	2	8D	JSR		6	3	F1	CMPB		4	3
22	BHI		3	2	56	RORB		2	1	8A	ORAA		2	2	8E	LDS		5	3	F2	SBCB		4	3
23	BLS		3	2	57	ASRB		2	1	8B	ADDA		2	2	8F	STS	EXTND	5	3	F3	ADDD		6	3
24	BCC		3	2	58	ASLB		2	1	8C	CPX	IMMED	4	3	C0	SUBB	IMMED	2	2	F4	ANDB		4	3
25	BCS		3	2	59	ROLB		2	1	8D	BSR	REL	6	2	C1	CMPB		6	2	F5	BITB		4	3
26	BNE		3	2	5A	DECB		2	1	8E	LDS	IMMED	3	3	C2	SBCB		4	2	F6	LDAB		4	3
27	BEQ		3	2	5B	*				8F	*				C3	ADDD		4	3	F7	STAB		4	3
28	BVC		3	2	5C	INCB		2	1	90	SUBA	DIR	3	2	C4	ANDB		2	2	F8	EORB		4	3
29	BVS		3	2	5D	TSTB		2	1	91	CMPA		3	2	C5	BITB		2	2	F9	ADCB		4	3
2A	BPL		3	2	5E	T				92	SBCA		3	2	C6	LDAB		2	2	FA	ORAB		4	3
2B	BMI		3	2	5F	CLRB	INHER	2	1	93	SUBD		5	2	C7	*				FB	ADDB		4	3
2C	BGE		3	2	60	NEG	INDXD	6	2	94	ANDA		3	2	C8	EORB		2	2	FC	LDD		4	3
2D	BGT		3	2	61	*				95	BITA		3	2	C9	ADCB		2	2	FD	STC		5	1
2E	BGT		3	2	62	*				96	LDAA		3	2	CA	ORAB		2	2	FE	---		5	1
2F	BLE	REL	3	2	63	COM		6	2	97	STAA		3	2	CB	ADDB		2	2	FF	---	EXTND	5	3
30	TSX	INHER	3	1	64	LSR		6	2	98	EORA		3	2	CC	LDD		3	3					
31	INS		3	1	65	*				99	ADCA		3	2	CD	*								
32	PULA		4	1	66	ROR		6	2	9A	ORAA		3	2	CE	LDX	IMMED	3	3					
33	PULB		4	1	67	ASR	INDXD	6	2	9B	ADDA		3	2	CF	*								



NOTES:

- Addressing Modes:  
 INHER = Inherent    INDXD = Indexed    IMMED = Immediate  
 REL = Relative    EXTND = Extended    DIR = Direct
- Unassigned opcodes are indicated by "\*" and should not be executed.
- Codes marked by "T" force the PC to function as a 16-bit counter.

Inherited Addressing

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

Inherent Addressing

The operand(s) is a register, and no memory reference is required. These are single-byte instructions.

Relative Addressing

Relative addressing is used only for branch instructions. If the branch condition is true, the program counter

is overwritten with the sum of 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.

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 instruction.

The information is useful in comparing actual results with expected results during debug of both software and

Table 9. Index Register and Stack Manipulation Instructions

Pointer Operations	MNEM	Immed		Direct		Index		Extnd		Inherent		Boolean/ Arithmetic Operation	Condition Codes									
		Op	#	Op	#	Op	#	Op	#	Op	#		5	4	3	2	1	0				
														H	I	N	Z	V	C			
Compare Index Register	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3	$X ← M:M + 1$	•	•	↓	↓	↓	↓		
Decrement Index Register	DEX												09	3	1	$X - 1 → X$	•	•	↓	•		
Decrement Stack Pointer	DES												34	3	1	$SP - 1 → SP$	•	•	•	•		
Increment Index Register	INX												08	3	1	$X + 1 → X$	•	•	•	•		
Increment Stack Pointer	INS												31	3	1	$SP + 1 → SP$	•	•	•	•		
Load Index Register	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3			$M → X_H:(M + 1) → X_L$	•	•	↓	↓	R	•
Load Stack Pointer	LDS	BE	3	3	9E	4	2	AE	5	2	BE	5	3			$M → SP_H:(M + 1) → SP_L$	•	•	↓	↓	R	•
Store Index Register	STX				DF	4	2	EF	5	2	FF	5	3			$X_H → M:X_L → (M + 1)$	•	•	↓	↓	R	•
Store Stack Pointer	STS				9F	4	2	AF	5	2	BF	5	3			$SP_H → M:SP_L → (M + 1)$	•	•	↓	↓	R	•
Index Reg → Stack Pointer	TXS												35	3	1	$X - 1 → SP$	•	•	•	•		
Stack Pntr → Index Register	TSX												30	3	1	$SP + 1 → X$	•	•	•	•		
Add	ABX												3A	3	1	$B + X → X$	•	•	•	•		
Push Data	PSHX												3C	4	1	$X_L → M:SP, SP - 1 → SP$ $X_H → M:SP, SP - 1 → SP$	•	•	•	•		
Pull Data	PULX												38	5	1	$SP + 1 → SP, M:SP → X_H$ $SP + 1 → SP, M:SP → X_L$	•	•	•	•		

Table 10. Accumulator and Memory Instructions (Sheet 1 of 2)

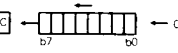
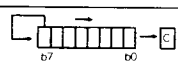
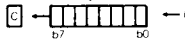
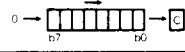
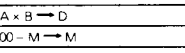
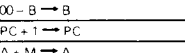
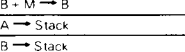
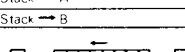
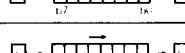
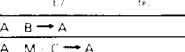
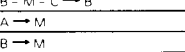
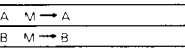
Accumulator and Memory Operations	MNEM	Immed		Direct		Index		Extend		Inher		Boolean Expression	Condition Codes										
		Op	#	Op	#	Op	#	Op	#	Op	#		5	4	3	2	1	0					
														H	I	N	Z	V	C				
Add Accumulators	ABA											1B	2	1	$A + B → A$	•	•	↓	↓	↓	↓		
Add B to X	ABX											3A	3	1	$00 B + X → X$	•	•	•	•	•			
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3			$A + M + C → A$	•	•	↓	↓	↓	↓	
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3			$B + M + C → B$	•	•	↓	↓	↓	↓	
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3			$A + M → A$	•	•	↓	↓	↓	↓	
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3			$B + M → A$	•	•	↓	↓	↓	↓	
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3			$D + M:M + 1 → D$	•	•	↓	↓	↓	↓	
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3			$A \cdot M → A$	•	•	↓	↓	↓	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3			$B \cdot M → B$	•	•	↓	↓	↓	R	•
Shift Left, Arithmetic	ASL							88	6	2	78	6	3				•	•	↓	↓	↓	↓	
	ASLA												48	2	1		•	•	↓	↓	↓	↓	
	ASLB												58	2	1		•	•	↓	↓	↓	↓	
Shift Left Double	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 Accumulators	CBA												11	2	1	$A - B$	•	•	↓	↓	↓	↓	
Clear	CLR					6F	6	2	7F	6	3					$00 → M$	•	•	R	S	R	R	
	CLRA												4F	2	1	$00 → A$	•	•	R	S	R	R	
	CLRB												5F	2	1	$00 → B$	•	•	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			$B - M$	•	•	↓	↓	↓	↓	
1's Complement	COM					63	6	2	73	6	3					$M → M$	•	•	↓	↓	↓	R	S
	COMA												43	2	1	$A → A$	•	•	↓	↓	↓	R	S
	COMB												53	2	1	$B → B$	•	•	↓	↓	↓	R	S

Table 10. Accumulator and Memory Instructions (Sheet 2 of 2)

Accumulator and Memory Operations	M/NEM	Immed		Direct		Index		Extend		Inher		Boolean Expression	Condition Codes								
		Op	#	Op	#	Op	#	Op	#	Op	#		5	4	3	2	1	0			
														H	I	N	Z	V	C		
Decimal Adjust, A	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 Accumulators	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										06	3	2		*	*	*	*	*	*	
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	*	*	*	
Multiply	MUL										3D	10	1	$A \times B \rightarrow D$	*	*	*	*	*	*	
	NEG					60	6	2	70	6	3			$00 - M \rightarrow M$	*	*	*	*	*	*	
	NEGA										40	2	1	$00 - A \rightarrow A$	*	*	*	*	*	*	
NEG B										50	2	1	$00 - B \rightarrow B$	*	*	*	*	*	*		
No Operation	NOP										01	2	1	$PC + 1 \rightarrow PC$	*	*	*	*	*	*	
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3	$A + M \rightarrow A$	*	*	*	*	R	*	
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3	$B + M \rightarrow B$	*	*	*	*	R	*	
Push Data	PSHA										36	3	1	$A \rightarrow \text{Stack}$	*	*	*	*	*	*	
	PSHB										37	3	1	$B \rightarrow \text{Stack}$	*	*	*	*	*	*	
Pull Data	PULA										32	4	1	$\text{Stack} \rightarrow A$	*	*	*	*	*	*	
	PULB										33	4	1	$\text{Stack} \rightarrow B$	*	*	*	*	*	*	
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 Accumulator	SBA										10	2	1	$A - B \rightarrow A$	*	*	*	*	*	*	
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3	$A - M - C \rightarrow A$	*	*	*	*	*	*	
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3	$B - M - C \rightarrow B$	*	*	*	*	*	*	
	STAA					97	3	2	A7	4	2	B7	4	3	$A \rightarrow M$	*	*	*	*	R	*
Store Accumulators	STAB					D7	3	2	E7	4	2	F7	4	3	$B \rightarrow M$	*	*	*	*	R	*
	STD					DD	4	2	ED	5	2	FD	5	3	$D \rightarrow M, M + 1$	*	*	*	*	R	*
	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3	$A - M \rightarrow A$	*	*	*	*	*	*	
Subtract Double	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3	$B - M \rightarrow B$	*	*	*	*	*	*	
	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3	$D - M, M + 1 \rightarrow D$	*	*	*	*	*	*	
Transfer Accumulator	TAB										16	2	1	$A \rightarrow B$	*	*	*	*	R	*	
	TBA										17	2	1	$B \rightarrow A$	*	*	*	*	R	*	
Test, Zero or Minus	TST					6D	6	2	7D	6	3			$M \wedge C$	*	*	*	*	*	H	
	TSTA										4D	2	1	$A \wedge C$	*	*	*	*	*	R	
	TSTB										5D	2	1	$B \wedge C$	*	*	*	*	*	R	

The condition code register notes are listed after Table 12.



Table 11. Jump and Branch Instructions

Operations	MNEM	Direct			Relative			Index			Extend			Inherent			Branch Test	Condition 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													$N \oplus V = 0$	*	*	*	*	*	*
Branch If > Zero	BGT				2E	3	2													$Z + (N \oplus V) = 0$	*	*	*	*	*	*
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													$Z + (N \oplus V) = 1$	*	*	*	*	*	*
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													$N \oplus V = 1$	*	*	*	*	*	*
Branch If Minus	BM!				2B	3	2													N = 1	*	*	*	*	*	*
Branch If Not Equal Zero	BNE				26	3	2													Z = 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 24	*	*	*	*	*	*
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					↓	↓	↓	↓	↓	↓
Return From Subroutine	RTS													39	5	1					*	*	*	*	*	*
Software Interrupt	SWI													3F	12	1					*	S	*	*	*	*
Wait For Interrupt	WAI													3E	9	1					*	*	*	*	*	*

Table 12. Condition Code Register Manipulation Instructions

Operations	MNEM	Inherent			Boolean Operation	Condition Code Register																	
		Op	~	#		5	4	3	2	1	0												
																				H	I	N	Z
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	SFV	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	Inherent	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	Inherent	Relative
INX	●	●	3	3	3	●
JMP	●	●	●	●	●	●
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	●



Table 14. Cycle-By-Cycle Operation (Sheet 1 of 5)

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

3

Table 14. Cycle-By-Cycle Operation (Sheet 2 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/W Line	Data Bus
<b>EXTENDED</b>					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Jump Address (High Order Byte)
		3	Opcode Address + 2	1	Jump Address (Low Order Byte)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand
		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Destination Address (High Order Byte)
		3	Opcode Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS LDX LDD	5	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
		3	Opcode 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	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
		3	Opcode 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	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
		3	Opcode 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 ADDD	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Operand Address (High Order Byte)
		3	Opcode 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	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Subroutine (High Order Byte)
		3	Opcode Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)

\* TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus - \$FFFF.

Table 14. Cycle-By-Cycle Operation (Sheet 3 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/W Line	Data Bus
<b>INDEXED</b>					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode 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	Opcode Address	1	Opcode
		2	Opcode 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	Opcode Address	1	Opcode
		2	Opcode 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	Opcode Address	1	Opcode
		2	Opcode 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	Opcode Address	1	Opcode
		2	Opcode 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	Opcode Address	1	Opcode
		2	Opcode 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 ADDD	6	1	Opcode Address	1	Opcode
		2	Opcode 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	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Opcode
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)

\* TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = \$FFFF

Table 14. Cycle-By-Cycle Operation (Sheet 4 of 5)

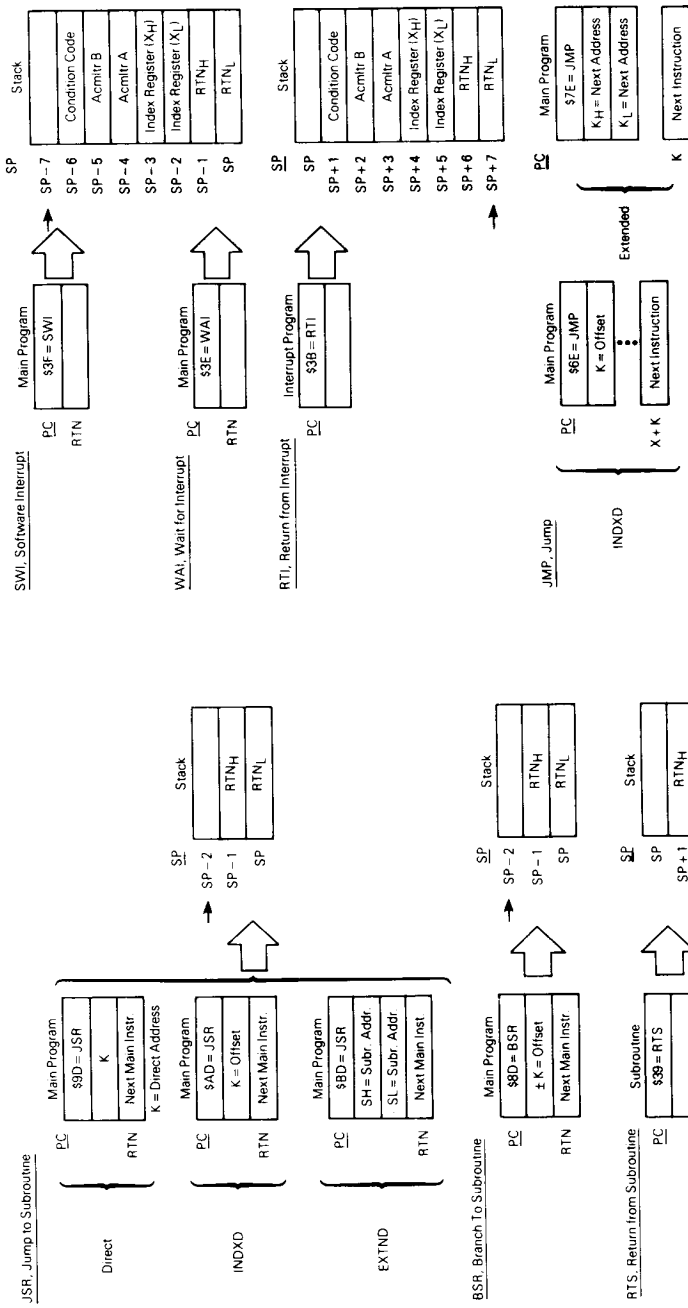
Address Mode and Instructions			Cycles	Cycle #	Address Bus	R/W Line	Data Bus
<b>INHERENT</b>							
ABA	DAA	SEC	2	1	Opcode Address	1	Opcode
ASL	DEC	SEI		2	Opcode Address + 1	1	Opcode of Next Instruction
ASR	INC	SEV					
CBA	LSR	TAB					
CLC	NEG	TAP					
CLI	NOP	TBA					
CLR	ROL	TPA					
CLV	ROR	TST					
COM	SBA						
ABX			3	1	Opcode Address	1	Opcode
				2	Opcode Address + 1	1	Irrelevant Data
				3	Address Bus FFFF	1	Low Byte of Restart Vector
ASLD			3	1	Opcode Address	1	Opcode
LSRD				2	Opcode Address + 1	1	Irrelevant Data
				3	Address Bus FFFF	1	Low Byte of Restart Vector
DES			3	1	Opcode Address	1	Opcode
INS				2	Opcode Address + 1	1	Opcode of Next Instruction
				3	Previous Stack Pointer Contents	1	Irrelevant Data
INX			3	1	Opcode Address	1	Opcode
DEX				2	Opcode Address + 1	1	Opcode of Next Instruction
				3	Address Bus FFFF	1	Low Byte of Restart Vector
PSHA			3	1	Opcode Address	1	Opcode
PSHB				2	Opcode Address + 1	1	Opcode of Next Instruction
				3	Stack Pointer	0	Accumulator Data
TSX			3	1	Opcode Address	1	Opcode
				2	Opcode Address + 1	1	Opcode of Next Instruction
				3	Stack Pointer	1	Irrelevant Data
TXS			3	1	Opcode Address	1	Opcode
				2	Opcode Address + 1	1	Opcode of Next Instruction
				3	Address Bus FFFF	1	Low Byte of Restart Vector
PULA			4	1	Opcode Address	1	Opcode
PULB				2	Opcode Address + 1	1	Opcode of Next Instruction
				3	Stack Pointer	1	Irrelevant Data
				4	Stack Pointer + 1	1	Operand Data from Stack
PSHX			4	1	Opcode Address	1	Opcode
				2	Opcode Address + 1	1	Irrelevant Data
				3	Stack Pointer	0	Index Register (Low Order Byte)
				4	Stack Pointer - 1	0	Index Register (High Order Byte)
PULX			5	1	Opcode Address	1	Opcode
				2	Opcode Address + 1	1	Irrelevant Data
				3	Stack Pointer	1	Irrelevant Data
				4	Stack Pointer + 1	1	Index Register (High Order Byte)
				5	Stack Pointer + 2	1	Index Register (Low Order Byte)
RTS			5	1	Opcode Address	1	Opcode
				2	Opcode Address + 1	1	Irrelevant Data
				3	Stack Pointer	1	Irrelevant Data
				4	Stack Pointer + 1	1	Address of Next Instruction (High Order Byte)
				5	Stack Pointer + 2	1	Address of Next Instruction (Low Order Byte)
WAI			9	1	Opcode Address	1	Opcode
				2	Opcode Address + 1	1	Opcode 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 Condition Code Register

3

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 5 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/W Line	Data Bus
<b>INHERENT</b>					
MUL	10	1	Opcode Address	1	Opcode
		2	Opcode 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	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Condition Code Register 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	Opcode Address	1	Opcode
		2	Opcode 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 Condition 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)
<b>RELATIVE</b>					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMI BVS	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
BSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)

3



**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 24. Special Operations

## ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola using the following media:

MDOS, disk file  
PC-DOS disk file (360K)  
EPROM(s) Two 2516 or 2716, or a single 2532, 2732, or MC68701U4

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field-service office, sales person, or Motorola representative.

### FLEXIBLE DISKS

Several types of flexible disks (MDOS™ or PC-DOS disk file) may be submitted for pattern generation. They should be programmed with the customer's program, using positive logic sense for address and data. The diskette should be clearly labeled with the customer's name, date, project or product name, and the filename containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

### MDOS Disk File

MDOS is Motorola's disk operating system available on the EXORciser development system. The disk media submitted must be a single-sided, single-density, 8-inch, MDOS-compatible floppy diskette. The diskette must contain the minimum set of MDOS system files in addition to the pattern file.

The .L0 output of the M6801 cross assembler should be furnished. In addition, the file must be produced using the ROLLOUT command, so that it contains the absolute image of the M6801 memory. The entire memory image of both program and data space must be included. All unused bytes, including those in the user space, must be set to logic zero.

### PC-DOS Disk File

PC-DOS is IBM® personal computer disk operating system. Submitted disk media must be standard-density (360K), double-sided, 5-1/4-inch-compatible floppy diskette. The diskette must contain the object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M6801 cross assemblers and linkers on IBM PC-style machines.

### EPROMS

Two K of EPROM are necessary to contain the entire MC6801U4 program. Two 2516 or 2716 type EPROMS, a

single 2532 or 2732 type EPROM, or an MC68701U4 can be submitted for pattern generation. The EPROM is programmed with the customer program, using positive logic sense for address and data. Submissions on two EPROMS must be clearly marked. All unused bytes, including the user's space, must be set to zero.

Whether the MC6801U4 MCU ROM pattern is submitted on a single 2532 or 2732 type EPROM, an MC68701U4, or on two 2516 or 2716 type EPROMS, memory map addressing is one-for-one. When using a single 2532 or 2732 EPROM, the ROM pattern to be copied runs from EPROM address \$000 to \$FFF. If an MC68701U4 is used, the ROM map runs from \$F000 to \$FFFF. If a pair of 2516 or 2716 type EPROMS is used, then they must be clearly marked; the data-space ROM runs from EPROM address \$000 to \$7FF, and the program-space ROM from \$7FF to \$FFF.

For shipment to Motorola, EPROMS should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.

### VERIFICATION MEDIA

All original pattern media, EPROMS or floppy disks, are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola will program customer-supplied blank EPROM(s) or DOS disks from the data file used to create the custom mask.

### ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum-order quantity, but are not production parts. These RVUs are not guaranteed by Motorola Quality Assurance.

### ORDERING INFORMATION

The following table provides generic information pertaining to the package type and temperature for the MC6801 and MC6803. These MCU devices are available in 40-pin CERDIP and plastic dual-in-line (DIP) packages.

MDOS is a trademark of Motorola Inc.

IBM is a registered trademark of International Business Machines Corporation.

**MECHANICAL DATA AND ORDERING INFORMATION**

The following table provides generic information pertaining to the package type and temperature for the MC6801

and MC6803. These MCU devices are available in 40-pin Cerdip and plastic dual-in-line (DIP) packages.

**GENERIC INFORMATION**

Package Type	Frequency (MHz)	Temperature	Part Number
Cerdip (S Suffix)	1.0	0° to 70°C	MC6801U4S1
	1.0	-40° to 85°C	MC6801U4CS1
	1.25	0° to 70°C	MC6801U4S1-1
	1.25	-40° to 85°C	MC6801U4CS1-1
	1.0	0° to 70°C	MC6803U4S
	1.0	-40° to 85°C	MC6803U4CS
	1.25	0° to 70°C	MC6803U4S-1
	1.25	-40° to 85°C	MC6803U4CS-1
Plastic (P Suffix)	1.0	0° to 70°C	MC6801U4P1
	1.0	-40° to 85°C	MC6801U4CP1
	1.25	0° to 70°C	MC6801U4P1-1
	1.25	-40° to 85°C	MC6801U4CP1-1
	1.0	0° to 70°C	MC6803U4P
	1.0	-40° to 85°C	MC6803U4CP
	1.25	0° to 70°C	MC6803U4P-1
	1.25	-40° to 85°C	MC6803U4CP-1



**PIN ASSIGNMENT**

