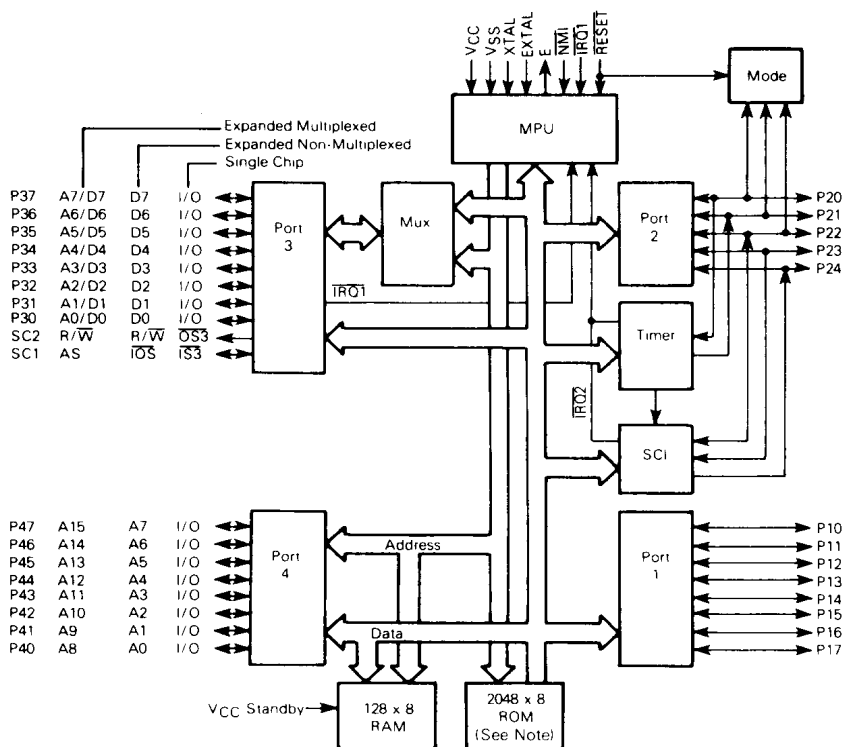**MC6801**
MC6803

Microcontroller/Microprocessor (MCU/MPU)

The MC6801 is an 8-bit single-chip microcontroller unit (MCU) which significantly enhances the capabilities of the M6800 Family of parts. It includes an upgraded M6800 microprocessor unit (MPU) with upward-source and object-code compatibility. Execution times of key instructions have been improved and several new instructions have been added including an unsigned multiply. 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 2048 bytes of ROM, 128 bytes of RAM, a serial communications interface (SCI), parallel I O, and a three-function programmable timer. The MC6803 can be considered as an MC6801 operating in modes 2 or 3. An EPROM version of the MC6801, the MC68701 microcontroller, is available for systems development. The MC68701 is pin and code compatible with the MC6801 MC6803 and can be used to emulate the MC6801 MC6803. The MC68701 is described in a separate Advanced Information publication.

- Enhanced MC6800 Instruction Set
- 8 · 8 Multiply Instruction
- Serial Communications Interface (SCI)
- Upward Source and Object Code Compatibility with the M6800
- 16-Bit Three-Function Programmable Timer
- Single-Chip or Expanded Operation to 64K Byte Address Space
- Bus Compatibility with the M6800 Family
- 2048 Bytes of ROM (MC6801 Only)
- 128 Bytes of RAM
- 64 Bytes of RAM Retainable During Powerdown
- 29 Parallel I O and Two Handshake Control Lines
- Internal Clock Generator with Divide-by-Four Output
- 40 to 85 C Temperature Range

FIGURE 1 — M6801 MICROCOMPUTER FAMILY BLOCK DIAGRAM

NOTE: No functioning ROM in MC6803.

POWER CONSIDERATIONS

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

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

where:

T_A = Ambient Temperature, °C
θ_{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:

$$P_D = K \div (T_J + 273°C) \tag{2}$$

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

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \tag{3}$$

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.

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 MC6801, MC6803 MC6801C, MC6803C	T_A	T_L to T_H 0 to 70 -40 to $+85$	°C
Storage Temperature Range	T_{stg}	-55 to $+150$	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended the V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{CC}$. Input protection is enhanced by connecting unused inputs to either V_{DD} or V_{SS}.

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip	θ_{JA}	 50 50	°C/W

3

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

Characteristic	Symbol	MC6801		MC6801-1		MC68B01		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	0.5	1.0	0.5	1.25	0.5	2.0	MHz
Crystal Frequency	f_{XTAL}	2.0	4.0	2.0	5.0	2.0	8.0	MHz
External Oscillator Frequency	$4f_o$	2.0	4.0	2.0	5.0	2.0	8.0	MHz
Crystal Oscillator Start Up Time	t_{rc}	—	100	—	100	—	100	ms
Processor Control Setup Time	t_{PCS}	200	—	170	—	110	—	ns

DC ELECTRICAL CHARACTERISTICS ($V_{CC} = 5.0$ Vdc $\pm 5\%$, $V_{SS} = 0$, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic		Symbol	MC6801 MC6803		MC6801C MC6803C		Unit
			Min	Max	Min	Max	
Input High Voltage	RESET Other Inputs	V_{IH}	$V_{SS}+4.0$ $V_{SS}+2.0$	V_{CC} V_{CC}	$V_{SS}+4.0$ $V_{SS}+2.2$	V_{CC} V_{CC}	V
Input Low Voltage	All Inputs	V_{IL}	$V_{SS}-0.3$	$V_{SS}+0.8$	$V_{SS}-0.3$	$V_{SS}+0.8$	V
Input Load Current ($V_{in}=0$ to 2.4 V)	Port 4 SC1	I_{in}	— —	0.5 0.8	— —	0.8 1.0	mA
Input Leakage Current ($V_{in}=0$ to 5.25 V)	NMI, IRQ1, RESET	I_{in}	—	2.5	—	5.0	μA
Hi-Z (Off State) Input Current ($V_{in}=0.5$ to 2.4 V)	Ports 1, 2, and 3	I_{TSI}	—	10	—	20	μA
Output High Voltage ($I_{Load}=-65$ μA, $V_{CC}=$Min)* ($I_{Load}=-100$ μA, $V_{CC}=$Min)	Port 4, SC1, SC2 Other Outputs	V_{OH}	$V_{SS}+2.4$ $V_{SS}+2.4$	— —	$V_{SS}+2.4$ $V_{SS}+2.4$	— —	V
Output Low Voltage ($I_{Load}=2.0$ mA, $V_{CC}=$Min)	All outputs	V_{OL}	—	$V_{SS}+0.5$	—	$V_{SS}+0.6$	V
Darlington Drive Current ($V_O=1.5$ V)	Port 1	I_{OH}	1.0	4.0	1.0	5.0	mA
Internal Power Dissipation (Measured at $T_A=T_L$ in Steady-State Operation)		P_{INT}	—	1200	—	1500	mW
Input Capacitance ($V_{in}=0$, $T_A=25$°C, $f_o=1.0$ MHz)	Port 3, Port 4, SC1 Other Inputs	C_{in}	— —	12.5 10	— —	12.5 10	pF
V_{CC} Standby	Powerdown Powerup	V_{SBB} V_{SB}	4.0 4.75	5.25 5.25	4.0 4.75	5.25 5.25	V
Standby Current	Powerdown	I_{SBB}	—	6.0	—	8.0	mA

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

PERIPHERAL PORT TIMING (Refer to Figures 2–5)

Characteristic	Symbol	MC6801 MC6803		MC6801-1 MC6803-1		MC68B01 MC68B03		Unit
		Min	Max	Min	Max	Min	Max	
Peripheral Data Setup Time	t_{PDSU}	200	—	200	—	100	—	ns
Peripheral Data Hold Time	t_{PDH}	200	—	200	—	100	—	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition	t_{OSD1}	—	350	—	350	—	250	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition	t_{OSD2}	—	350	—	350	—	250	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	t_{PWD}	—	350	—	350	—	250	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t_{CMOS}	—	2.0	—	2.0	—	2.0	μs
Input Strobe Pulse Width	t_{PWIS}	200	—	200	—	100	—	ns
Input Data Hold Time	t_{IH}	50	—	50	—	30	—	ns
Input Data Setup Time	t_{IS}	20	—	20	—	20	—	ns

FIGURE 2 — DATA SETUP AND HOLD TIMES (MPU READ)
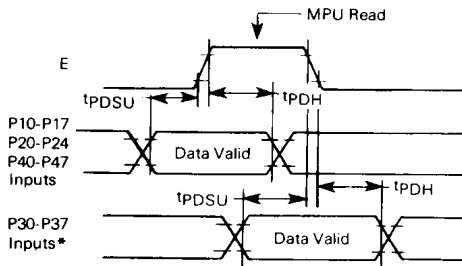
* Port 3 non-latched operation (LATCH ENABLE = 0)

FIGURE 3 — DATA SETUP AND HOLD TIMES (MPU WRITE)
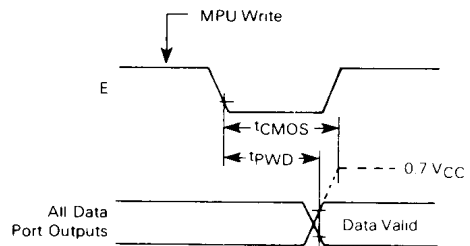
NOTES:
1. 10 k pullup resistor required for port 2 to reach 0.7 V_{CC}.
2. Not applicable to P21.
3. Port 4 cannot be pulled above V_{CC}.

FIGURE 4 — PORT 3 OUTPUT STROBE TIMING (MC6801 SINGLE-CHIP MODE)
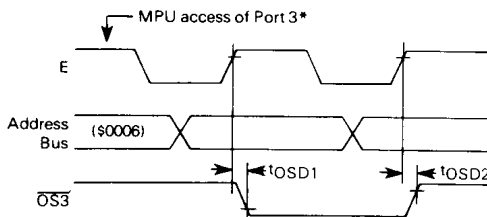
* Access matches output strobe select (OSS = 0, a read; OSS = 1, a write)

FIGURE 5 — PORT 3 LATCH TIMING (MC6801 SINGLE-CHIP MODE)
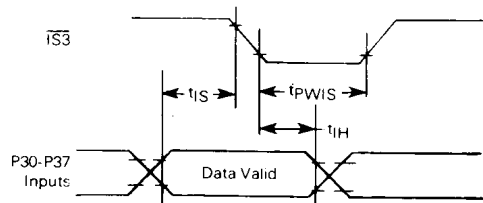
NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

3

BUS TIMING (See Notes 1 and 2)

| Ident. Number	Characteristics	Symbol	MC6801 MC6803		MC6801-1 MC6803-1		MC68B01 MC68B03		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	2.0	0.8	2.0	0.5	2.0	μs
2	Pulse Width, E Low	PW_{EL}	430	1000	360	1000	210	1000	ns
3	Pulse Width, E High	PW_{EH}	450	1000	360	1000	220	1000	ns
4	Clock Rise and Fall Time	t_r, t_f	—	25	—	25	—	20	ns
9	Address Hold Time	t_{AH}	20	—	20	—	10	—	ns
12	Non-Muxed Address Valid Time to E*	t_{AV}	200	—	150	—	70	—	ns
17	Read Data Setup Time	t_{DSR}	80	—	70	—	40	—	ns
18	Read Data Hold Time	t_{DHR}	10	—	10	—	10	—	ns
19	Write Data Delay Time	t_{DDW}	—	225	—	200	—	120	ns
21	Write Data Hold Time	t_{DHW}	20	—	20	—	10	—	ns
22	Muxed Address Valid Time to E Rise*	t_{AVM}	200	—	150	—	80	—	ns
24	Muxed Address Valid Time to AS Fall*	t_{ASL}	60	—	50	—	20	—	ns
25	Muxed Address Hold Time	t_{AHL}	20	—	20	—	10	—	ns
26	Delay time, E to AS Rise*	t_{ASD}	90**	—	70**	—	45**	—	ns
27	Pulse Width, AS High*	PW_{ASH}	220	—	170	—	110	—	ns
28	Delay Time, AS to E Rise*	t_{ASED}	90	—	70	—	45	—	ns
29	Usable Access Time*	t_{ACC}	595	—	465	—	270	—	ns

*At specified cycle time.
**t_{ASD} parameters listed assume external TTL clock drive with 50% ± 5% duty cycle. Devices driven by an external TTL clock with 50% ± 1% duty cycle or which use a crystal have the following t_{ASD} specifications: 100 nanoseconds minimum (1.0 MHz devices), 80 nanoseconds minimum (1.25 MHz device), 50 nanoseconds minimum (2.0 MHz devices).

FIGURE 6 — BUS TIMING
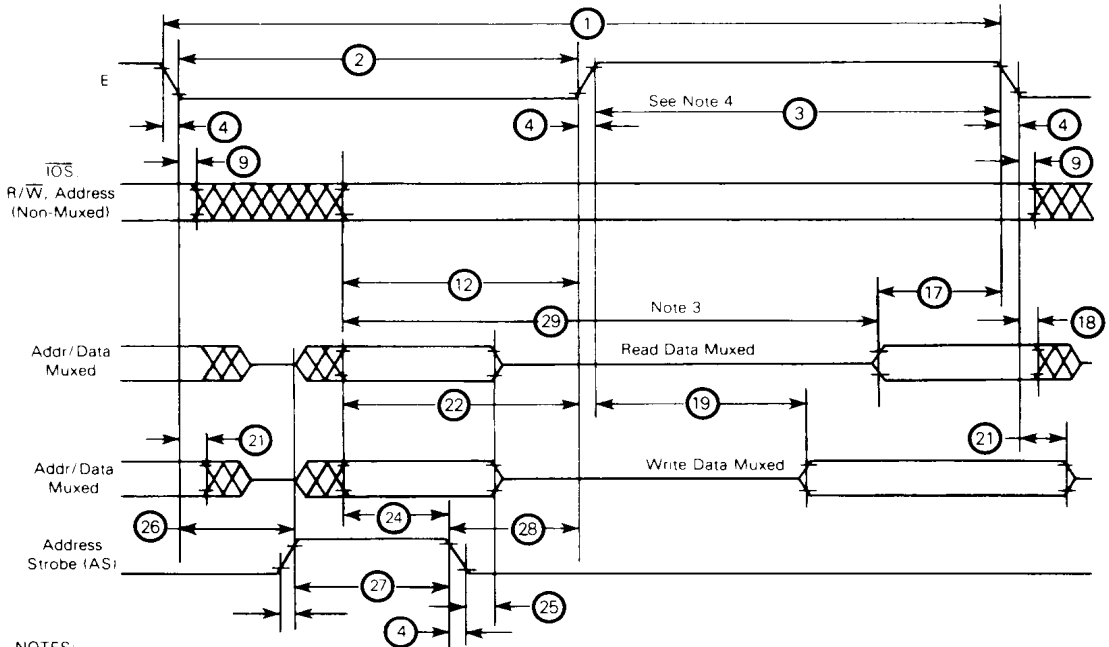
NOTES:
1. Voltage levels shown are $V_L \leq 0.5$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Usable access time is computed by: 12 + 3 − 17 + 4.
4. Memory devices should be enabled only during E high to avoid port 3 bus contention.
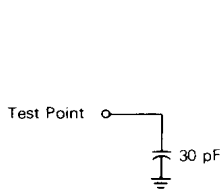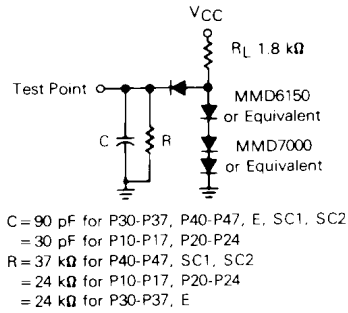
FIGURE 7 — CMOS LOAD

Test Point o——

30 pF

FIGURE 8 — TIMING TEST LOAD PORTS 1, 2, 3, 4

V_{CC}

R_L 1.8 kΩ

Test Point o

MMD6150
or Equivalent

C   R

MMD7000
or Equivalent

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

INTRODUCTION

The MC6801 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into eight different operating modes. The operating mode controls the configuration of 18 of the 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 microprocessor unit (MPU) is an enhanced MC6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the MC6800. The programming model is depicted in Figure 9, where accumulator D is a concatenation of accumulators A and B. A list of new operations added to the M6800 instruction set are shown in Table 1.

The MC6803 can be considered an MC6801 that operates in Modes 2 and 3 only.

3

FIGURE 9 — PROGRAMMING MODEL
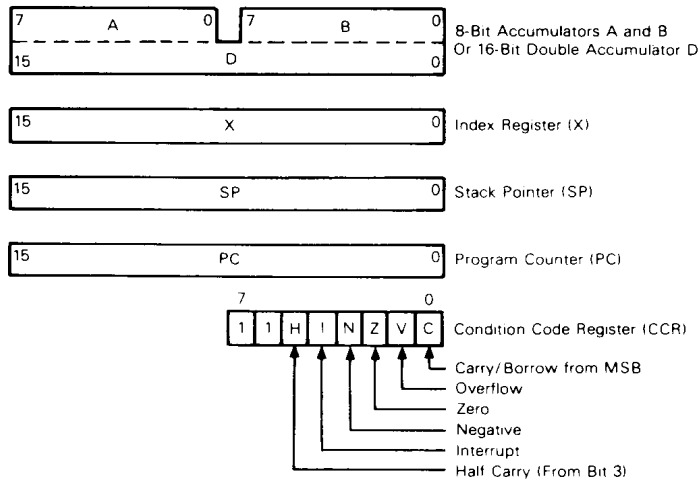
OPERATING MODES

The MC6801 provides eight different operating modes (0 through 7) and the MC6803 provides two operating 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 eight operating modes can be grouped into three fundamental modes which refer to the type of bus it supports: single chip, expanded non-multiplexed, and expanded multiplexed. Single-chip modes include 4 and 7, expanded non-multiplexed mode is 5, and the remaining five modes are expanded multiplexed modes. Table 2 summarizes the characteristics of the operating modes.

MC6801 Single-Chip Modes (4, 7)

In the single-chip mode, the four MCU ports are configured as parallel input/output data ports, as shown in Figure 10. The MCU functions as a monolithic microcomputer in these two modes without external address or data buses. A maximum of 29 I/O lines and two port 3 control lines are provided. Peripherals or another MCU can be interfaced to port 3 in a loosely coupled dual processor configuration, as shown in Figure 11.

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

In single-chip test mode (4), the RAM responds to $XX80 through $XXFF and the ROM is removed from the internal address map. A test program must first be loaded into the RAM using modes 0, 1, 2, or 6. If the MCU is reset and then programmed into mode 4, execution will begin at $XXFE:XXFF. Mode 5 can be irreversibly entered from mode 4 without asserting $\overline{RESET}$ by setting bit 5 of the port 2 data register. This mode is used primarily to test ports 3 and 4 in the single-chip and non-multiplexed modes.

MC6801 Expanded Non-Multiplexed Mode (5)

A modest amount of external memory space is provided in the expanded non-multiplexed 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, any 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 12 illustrates a typical system configuration in the expanded non-multiplexed 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.

TABLE 2 — SUMMARY OF MC6801/03 OPERATING MODES

Common to all Modes: Reserved Register Area Port 1 Port 2 Programmable Timer Serial Communications Interface
Single Chip Mode 7 128 bytes of RAM; 2048 bytes of ROM Port 3 is a parallel I/O port with two control lines Port 4 is a parallel I/O port SC1 is Input Strobe 3 ($\overline{IS3}$) SC2 is Output Strobe 3 ($\overline{OS3}$)
Expanded Non-Multiplexed Mode 5 128 bytes of RAM; 2048 bytes of ROM 256 bytes of external memory space Port 3 is an 8-bit data bus Port 4 is an input port/address bus SC1 is Input/Output Select ($\overline{IOS}$) SC2 is Read/Write (R/$\overline{W}$)

Expanded Multiplexed Modes 1, 2, 3, 6*
Four memory space options (64K address space):
  (1) No internal RAM or ROM (Mode 3)
  (2) Internal RAM, no ROM (Mode 2)
  (3) Internal RAM and ROM (Mode 1)
  (4) Internal RAM, ROM with partial address bus (Mode 6)
Port 3 is a multiplexed address/data bus
Port 4 is an address bus (inputs/address in Mode 6)
SC1 is Address Strobe (AS)
SC2 is Read/Write (R/$\overline{W}$)

Test Modes 0 and 4
Expanded Multiplexed Test Mode 0
  May be used to test RAM and ROM
Single Chip and Non-Multiplexed Test Mode 4
  (1) May be changed to Mode 5 without going through Reset
  (2) May be used to test Ports 3 and 4 as I/O ports

* The MC6803 operates only in modes 2 and 3.
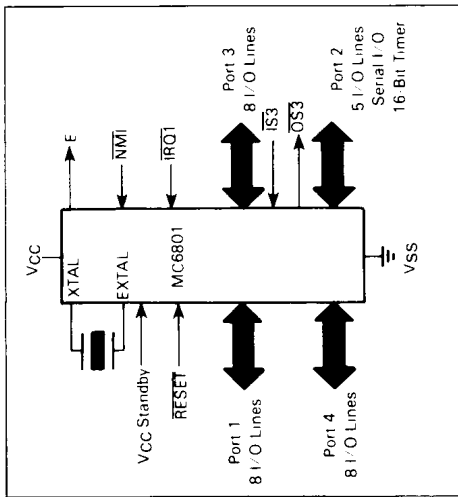
3

3

FIGURE 11 — SINGLE-CHIP DUAL PROCESSOR CONFIGURATION
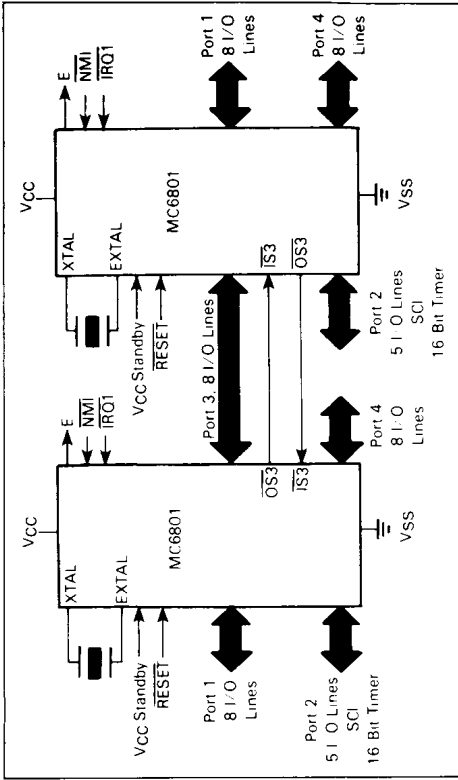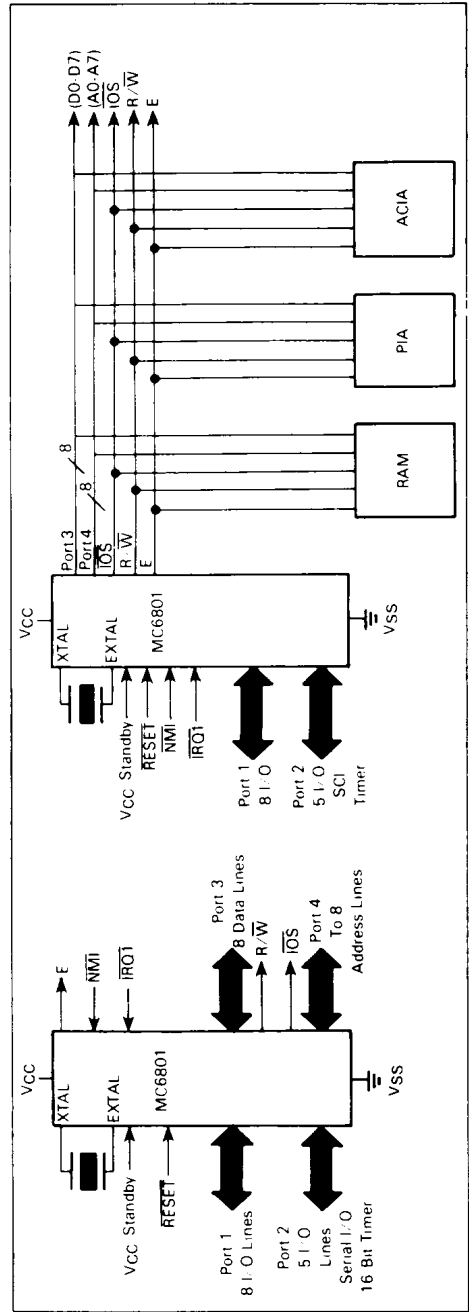
FIGURE 10 — SINGLE-CHIP MODE

FIGURE 12 — EXPANDED NON-MULTIPLEXED CONFIGURATION

Expanded-Multiplex 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 functions 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 to 3, port 4 provides address lines A8 to A15. In mode 6, however, port 4 initially is configured at $\overline{RESET}$ 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 0, the reset vector is external for the first two E cycles after the positive edge of $\overline{RESET}$, and internal thereafter. In addition, the internal and external data buses are connected so there must be no memory map overlap in order to avoid potential bus conflicts. Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with the automated test equipment.

Only the MC6801 can operate in each of the expanded-multiplexed modes. The MC6803 operates only in modes 2 and 3.

Figure 13 depicts a typical configuration for the expanded-multiplexed modes. Address strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in Figure 14. This allows port 3 to function as a data bus when E is high.

PROGRAMMING THE MODE

The operating mode is determined at $\overline{RESET}$ 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 $\overline{RESET}$. 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 15. A brief outline of the operating modes is shown in Table 3. Note that if diodes are used to program the mode, the diode forward voltage drop must not exceed the V_{MPDD} minimum.

PORT 2 DATA REGISTER

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

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the three pins. If configured as outputs, the circuit shown in Figure 16 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

3

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(5, 6)	Multiplexed Partial Decode
5	H	L	H	I	I	I	NMUX(5, 6)	Non-Multiplexed Partial Decode
4	H	L	L	I(2)	I(1)	I	I	Single-Chip Test
3	L	H	H	E	E	E	MUX(4)	Multiplexed No RAM or ROM
2	L	H	L	E	I	E	MUX(4)	Multiplexed RAM
1	L	L	H	I	I	E	MUX(4)	Multiplexed RAM and ROM
0	L	L	L	I	I	I(3)	MUX(4)	Multiplexed Test

Legend:
I — Internal
E — External
MUX — Multiplexed
NMUX — Non-Multiplexed
L — Logic Zero
H — Logic One

NOTES:
(1) Internal RAM is addressed at $XX80.
(2) Internal ROM is disabled.
(3) $\overline{RESET}$ vector is external for two cycles after $\overline{RESET}$ goes high
(4) Addresses associated with ports 3 and 4 are considered external in modes 0, 1, 2, and 3.
(5) Addresses associated with port 3 are considered external in modes 5 and 6.
(6) Port 4 default is user data input; address output is optional by writing to port 4 data direction register.

*The MC6803 operates only in modes 2 and 3.

FIGURE 13 — EXPANDED MULTIPLEXED CONFIGURATION

FIGURE 13 — EXPANDED MULTIPLEXED CONFIGURATION
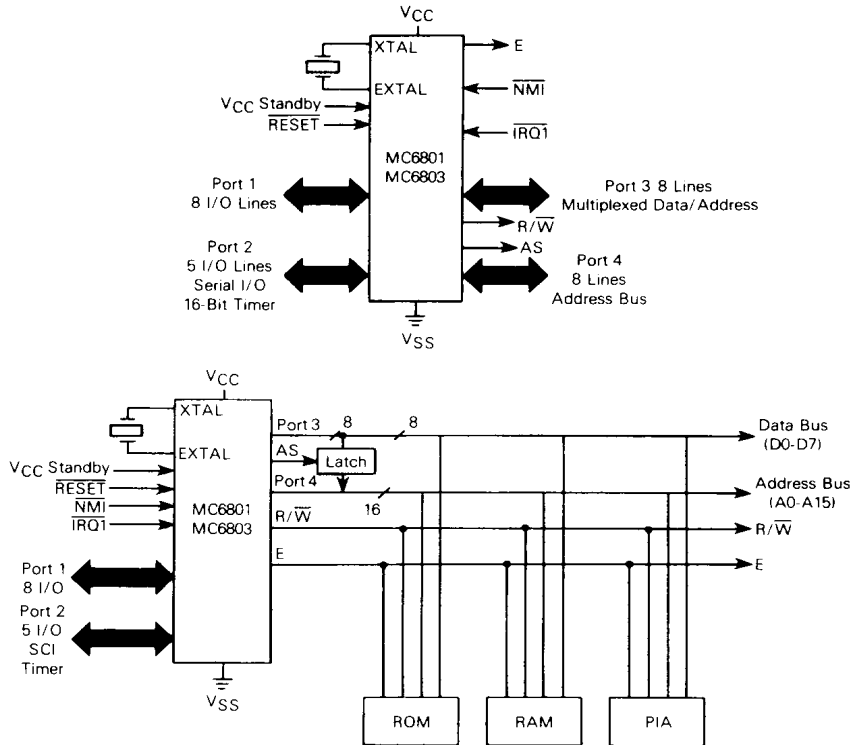
NOTE: To avoid data bus (port 3) contention in the expanded multiplexed modes, memory devices should be enabled only during E high time.

FIGURE 14 — TYPICAL LATCH ARRANGEMENT

FIGURE 14 — TYPICAL LATCH ARRANGEMENT
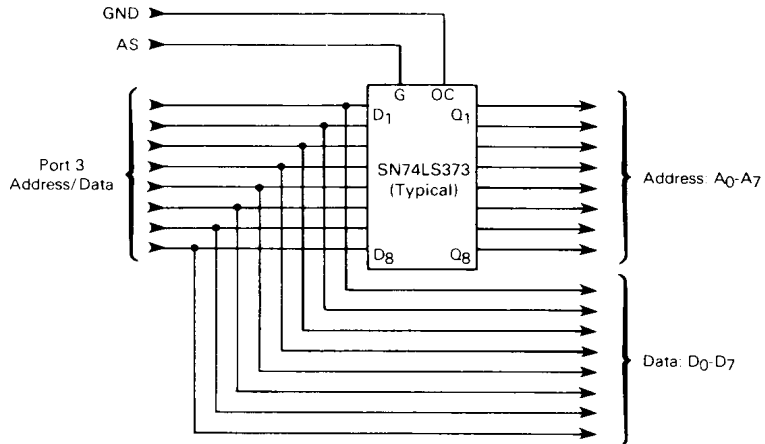
FIGURE 15 — MODE PROGRAMMING TIMING
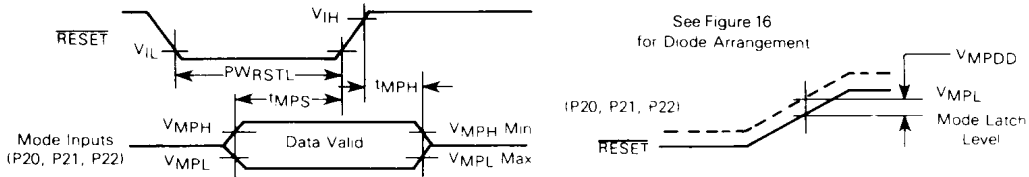
MODE PROGRAMMING (Refer to Figure 15)

Characteristic	Symbol	Min	Max	Unit
Mode Programming Input Voltage Low* (for $T_A = 0$ to 70 C)	V_{MPL}	—	1.7	V
Mode Programming Input Voltage High	V_{MPH}	4.0	—	V
Mode Programming Diode Differential (If Diodes are Used) (for T_A 0 to 70 C)	V_{MPDD}	0.4	—	V
RESET Low Pulse Width	PW_{RSTL}	3.0	—	E Cycles
Mode Programming Setup Time	t_{MPS}	2.0	—	E Cycles
Mode Programming Hold Time RESET Rise Time$\geqslant 1$ μs RESET Rise Time<1 μs	t_{MPH}	0 100	— —	ns

Note: For $T_A = -40$ to 85°C, Maximum $V_{MPL} = 1.7$, and Minimum V_{MPDD} 0.4.

FIGURE 16 — TYPICAL MODE PROGRAMMING CIRCUIT
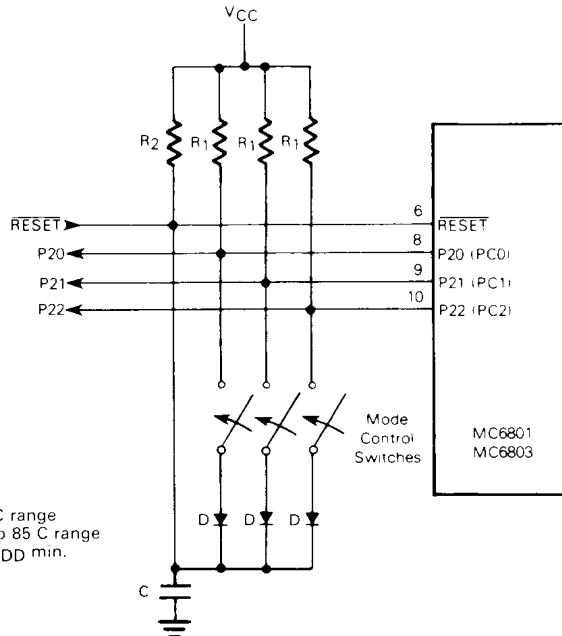
NOTES:
1. Mode 7 as shown
2. $R_2 \cdot C$ — Reset time constant
3. $R_1 = 10$ k (typical)
4. D = IN914, IN4001 in the 0 to 70 C range
   D — 1N270, MBD201 in the −40 to 85 C range
5. Diode V_f should not exceed V_{MPDD} min.

MEMORY MAPS

The M6801 Family can provide up to 64K byte address space depending on the operating mode. A memory map for each operating mode is shown in Figure 17. The first 32 locations of each map are reserved for the internal register area, as shown in Table 4, with exceptions as indicated.

FIGURE 17 — MC6801/03 MEMORY MAPS (Sheet 1 of 3)

3

Multiplexed-Test Mode

MC6801 Mode **0**

$0000(1)
$001F — Internal Registers
— External Memory Space
$0080
— Internal RAM
$00FF

— External Memory Space

$F800
— Internal ROM
$FFFF(2) — Internal Interrupt Vectors(2)

NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.
2) Addresses $FFFE and $FFFF are considered <u>external</u> if accessed within two cycles after a positive edge of RESET and internal at all other times.
3) After two MPU cycles, there must be no over-lapping 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 interrupt vectors in internal ROM using an external RESET vector.

MC6801 Mode **1**

Multiplexed/RAM and ROM

$0000(1)
— Internal Registers
$001F
— External Memory Space
$0080
— Internal RAM
$00FF

— External Memory Space

$F800
— Internal ROM
$FFEF
$FFF0
— External Interrupt Vectors
$FFFF

NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.
2) Internal ROM addresses $FFF0 to $FFFF are not usable.

FIGURE 17 — MC6801/03 MEMORY MAPS (Sheet 2 of 3)

MC6801 MC6803 Mode 2

Multiplexed/RAM

- $0000 (1) — Internal Registers
- $001F
- $0080 — External Memory Space / Internal RAM
- $00FF
- External Memory Space
- $FFF0 — External Interrupt Vectors
- $FFFF

NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.

MC6801 MC6803 Mode 3

Multiplexed/No RAM or ROM

- $0000 (1) — Internal Registers
- $001F
- External Memory Space
- $FFF0 — External Interrupt Vectors
- $FFFF

NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.

MC6801 Mode 4

Single-Chip Test

- $0000 — Internal Registers
- $001F
- Unusable (1)(4)
- $XX80 — Internal RAM / Internal Interrupt Vectors
- $XXFF

NOTES:
1) The internal ROM is disabled.
2) Mode 4 may be changed to Mode 5 without having to assert RESET by writing a one into the PC0 bit of the port 2 data register.
3) Addresses A8 to A15 are treated as "don't cares" to decode internal RAM.
4) Internal RAM will appear at $XX80 to $XXFF.

3

3

FIGURE 17 — MC6801/03 MEMORY MAPS (Sheet 3 of 3)

MC6801 Mode 5

Non-Multiplexed/Partial Decode

$0000(1) — Internal Registers
$001F
— Unusable
$0080 — Internal RAM
$00FF
$0100 — External Memory Space
$01FF
— Unusable
$F800 — Internal ROM
$FFFF — Internal Interrupt Vectors

NOTES:
1) Excludes the following addresses which may **not** be used externally: $04, $06, and $0F (no IOS).
2) This mode may be entered without going through RESET by using mode 4 and subsequently writing a one into the PCO bit of the port 2 data register.
3) Address lines A0 to A7 will not contain addresses until the data direction register for port 4 has been written with ones in the appropriate bits. These address lines will assert ones until made outputs by writing the data direction register.

MC6801 Mode 6

Multiplexed/Partial Decode

$0000(1) — Internal Registers
$001F — External Memory Space
$0080 — Internal RAM
$00FF
— External Memory Space
$F800 — Internal ROM
$FFFF — Internal Interrupt Vectors

NOTES:
1) Excludes the following addresses which may be used externally: $04, $06, and $0F.
2) Address lines A8-A15 will not contain addresses until the data direction register for port 4 has been written with ones in the appropriate bits. These address lines will assert ones until made outputs by writing the data direction register.

MC6801 Mode 7

Single Chip

$0000 — Internal Registers
$001F
— Unusable
$0080 — Internal RAM
$00FF
— Unusable
$F800 — Internal ROM
$FFFF — Internal Interrupt Vectors

MC6801/03 INTERRUPTS

The M6801 Family supports two types of interrupt requests: maskable and non-maskable. A non-maskable interrupt ($\overline{\text{NMI}}$) 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: $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$. The programmable timer and serial communications interface use an internal $\overline{\text{IRQ2}}$ interrupt line, as shown in Figure 1. External devices (and IS3) use $\overline{\text{IRQ1}}$. An $\overline{\text{IRQ1}}$ interrupt is serviced before $\overline{\text{IRQ2}}$ if both are pending.

All $\overline{\text{IRQ2}}$ 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.

The interrupt flowchart is depicted in Figure 18 and 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 $\overline{\text{RESET}}$ timing are illustrated in Figures 19 and 20.

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

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

V_{CC} STANDBY

V_{CC} standby provides power to the standby portion ($80 through $BF) of the RAM and the STBY PWR and RAME bits of the RAM control register. Voltage requirements depend on whether the device is in a powerup or powerdown state. In the powerup state, the power supply should provide + 5 volts ($\pm 5\%$) and must reach V_{SB} volts before $\overline{\text{RESET}}$ reaches 4.0 volts. During powerdown, V_{CC} standby must remain above V_{SBB} (min) to sustain the standby RAM and STBY PWR bit. While in powerdown operation, the standby current will not exceed I_{SBB}.

It is typical to power both V_{CC} and V_{CC} standby from the same source during normal operation. A diode must be used between them to prevent supplying power to V_{CC} during powerdown operation. V_{CC} standby should be tied to ground in mode 3.

TABLE 4 — INTERNAL REGISTER AREA

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

* External addresses in modes 0, 1, 2, 3, 5, and 6, cannot be accessed in mode 5 (no $\overline{\text{IOS}}$).

* * External addresses in modes 0, 1, 2, and 3

* * * 1 = Output, 0 = Input

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

MSB	LSB	Interrupt
FFFE	FFFF	$\overline{\text{RESET}}$
FFFC	FFFD	$\overline{\text{NMI}}$
FFFA	FFFB	Software Interrupt (SWI)
FFF8	FFF9	$\overline{\text{IRQ1}}$ (or IS3)
FFF6	FFF7	ICF (Input Capture) *
FFF4	FFF5	OCF (Output Capture) *
FFF2	FFF3	TOF (Timer Overflow) *
FFF0	FFF1	SCI (RDRF + ORFE + TDRE) *

* $\overline{\text{IRQ2}}$ Interrupt

FIGURE 18 — INTERRUPT FLOWCHART

$SCI = TIE \cdot TDRE + RIE \cdot (RDRF + ORFE)$

Vector		
NMI	FFFC:FFFD	Non Maskable Interrupt
SWI	FFFA:FFFB	Software Interrupt
IRQ1	FFF8:FFF9	Maskable Interrupt Request 1
ICF	FFF6:FFF7	Input Capture Interrupt
OCF	FFF4:FFF5	Output Compare Interrupt
TOF	FFF2:FFF3	Timer Overflow Interrupt
SCI	FFF0:FFF1	SCI Interrupt (TDRE + RDRF + ORFE)

Condition Code Register

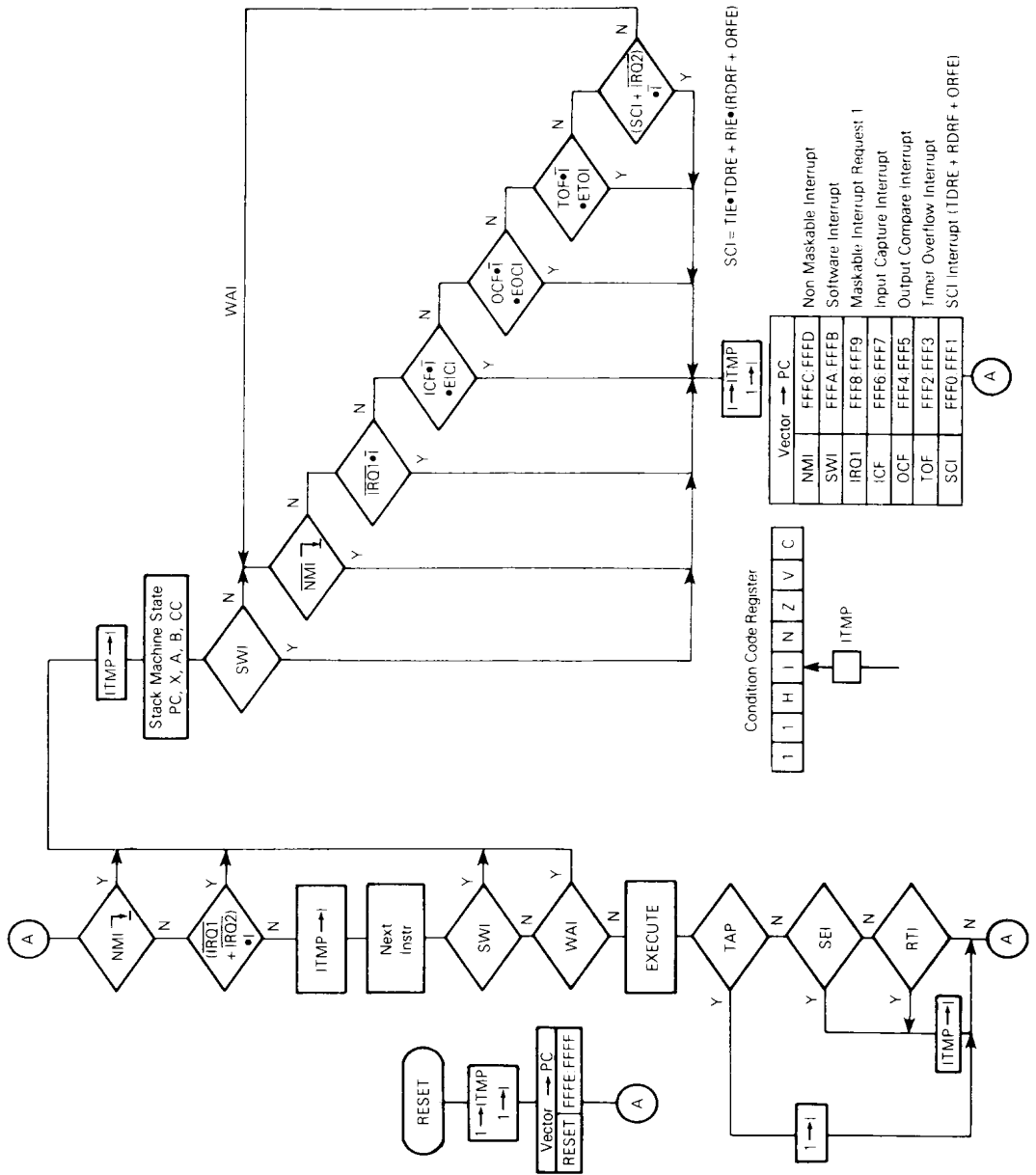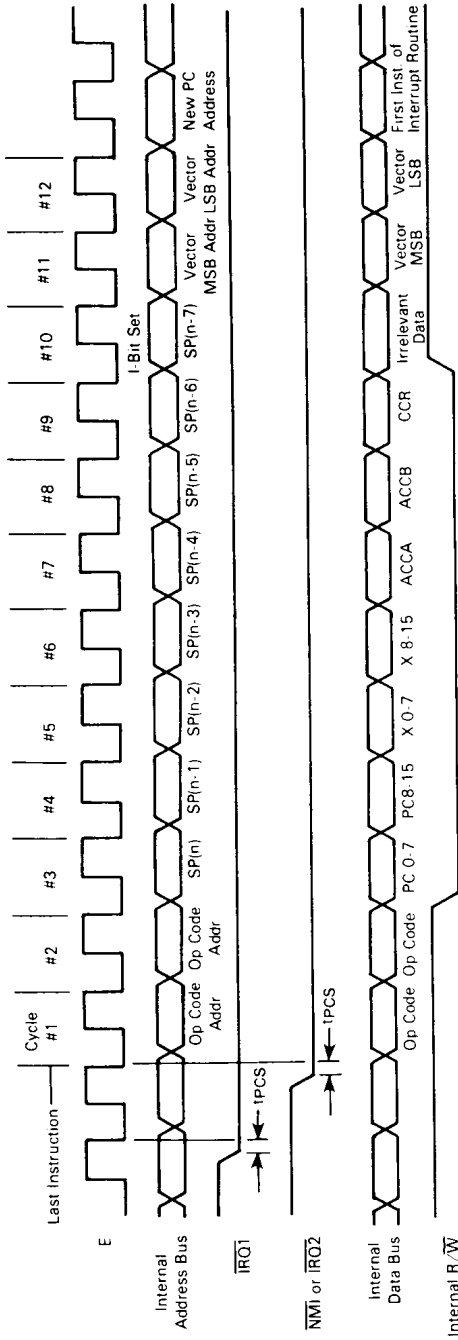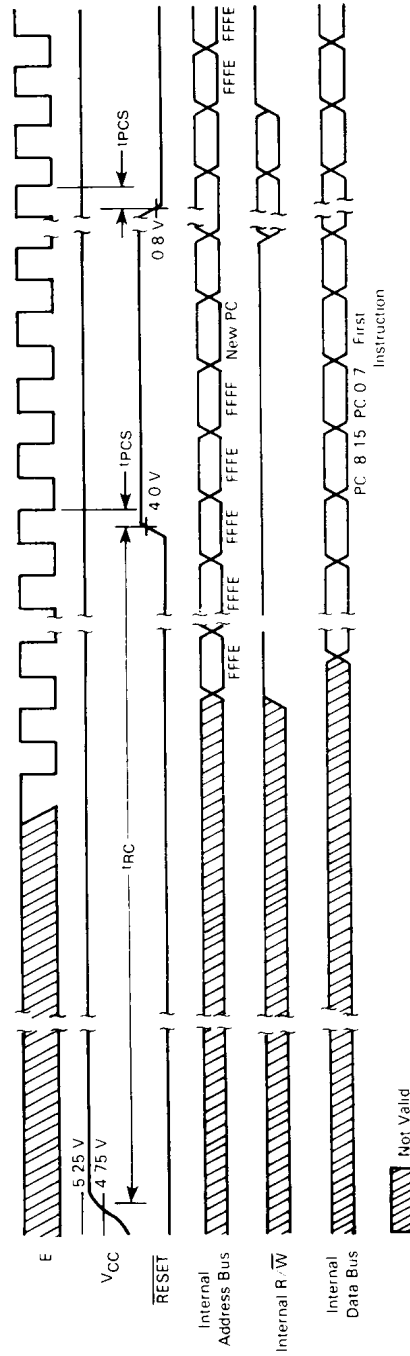| 1	1	H	I	N	Z	V	C

FIGURE 19 — INTERRUPT SEQUENCE

FIGURE 20 — RESET TIMING

3

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 $4f_0$ with a duty cycle of 50% ($\pm$ 5%) with XTAL connected to 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 f_{XTAL}. 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 21.

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 t_{RC} after V_{CC} reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until V_{CC} standby reaches 4.75 volts. 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 is referenced to this clock unless otherwise noted.

NON-MASKABLE INTERRUPT (NMI)

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, transferred to the program counter and instruction execution is resumed. NMI typically requires a 3.3 kΩ (nominal) resistor to V_{CC}. There is no internal NMI pullup resistor. NMI must be held low for at least one E cycle to be recognized under all conditions.

MASKABLE INTERRUPT REQUEST 1 (IRQ1)

IRQ1 is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the interrupt mask bit (I bit) in the condition code register is clear, the MCU will begin an interrupt sequence. A vector is fetched from $FFF8 and $FFF9, transferred to the program counter, and instruction execution is resumed.

IRQ1 typically requires an external 3.3 kΩ (nominal) resistor to V_{CC} for wire-OR applications. IRQ1 has no internal pullup resistor.

STROBE CONTROL 1 AND 2 (SC1 AND SC2)

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 port 3 control and status register and are discussed in the PORT 3 (P30-P37). 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 4.

SC1 and SC2 In Expanded Non-Multiplexed Mode

In the expanded non-multiplexed 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.

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 mode, 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 address on the negative edge, as shown in Figure 14.

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.

PORT 1 (P10-P17)

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

PORT 2 (P20-P24)

PORT 2 DATA REGISTER

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

Port 2 is a mode-independent, 5-bit, multi-purpose 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 will be tied to the timer output compare function and cannot be used to provide output from the port 2 data register.

* Devices made with masks subsequent to M5G, M8D, and T5P incorporate an advanced clock with improved startup characteristics.

FIGURE 21 — M6801 FAMILY OSCILLATOR CHARACTERISTICS

(a) Nominal Recommended Crystal Parameters

Nominal Crystal Parameters*

	3.58 MHz	4.00 MHz	5.0 MHz	6.0 MHz	8.0 MHz
R_S	60 Ω	50 Ω	30-50 Ω	30-50 Ω	20-40 Ω
C_0	3.5 pF	6.5 pF	4-6 pF	4-6 pF	4-6 pF
C_1	0.015 pF	0.025 pF	0.01-0.02 pF	0.01-0.02 pF	0.01-0.02 pF
Q	>40 K	>30 K	>20 K	>20 K	>20 K

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

$C_L = 20$ pF (typical)
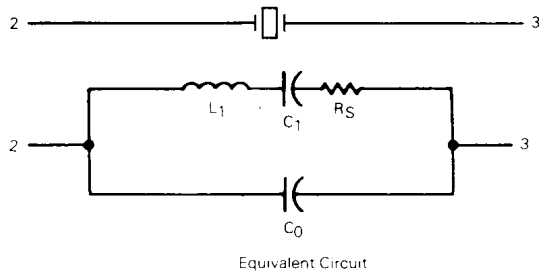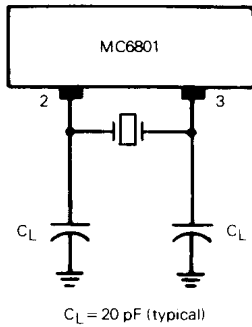
NOTE

TTL-compatible oscillators may be obtained from:

Motorola Component Products
Attn: Data Clock Sales
    2553 N. Edgington St
Franklin Park, IL 60131
Tel: 312-451-1000
Telex: 433-0067

Equivalent Circuit

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

4.75 V

V_{CC}

E

$\overline{RESET}$

0.8 V

t_{RC}
Oscillator
Stabilization
Time, t_{RC}
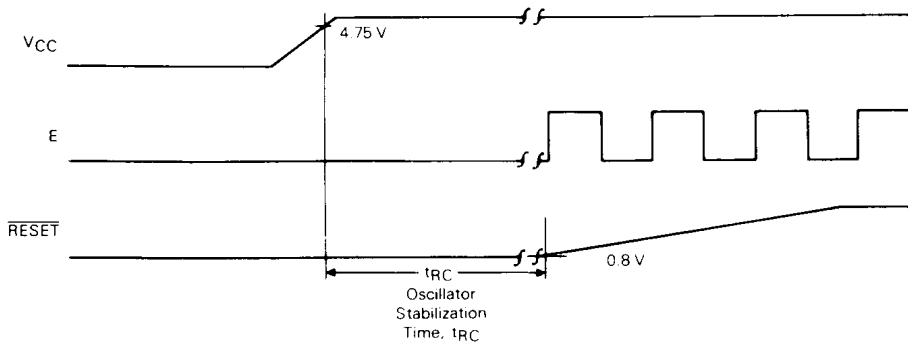
Port 2 can also be used to provide an interface for the serial communications interface and the timer input edge function. These configurations are described in **PROGRAMMABLE TIMER** and **SERIAL COMMUNICATIONS INTERFACE (SCI)**.

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

PORT 3 (P30-P37)

Port 3 can be configured as an I/O port, a bidirectional 8-bit data bus, or a multiplexed address bus depending on the operating mode. The TTL-compatible high-impedance 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, $\overline{IS3}$ and $\overline{OS3}$, 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 $\overline{IS3}$ as a control signal, (2) $\overline{OS3}$ can be generated by either an MPU read or write to the port 3 data register, and (3) an $\overline{IRQ1}$ interrupt can be enabled by an $\overline{IS3}$ negative edge. Port 3 latch timing is shown in Figure 5.

PORT 3 CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
IS3 Flag	IS3 IRQ1 Enable	X	OSS	Latch Enable	X	X	X	$000F

Bit 0-2	Not used.
Bit 3	LATCH ENABLE. This bit controls the input latch for port 3. If set, input data is latched by an $\overline{IS3}$ negative edge. The latch is transparent after a read of the port 3 data register. LATCH ENABLE is cleared during reset.
Bit 4	OSS (Output Strobe Select). This bit determines whether $\overline{OS3}$ 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 $\overline{IRQ1}$ interrupt will be enabled whenever 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 $\overline{IS3}$ negative edge. It is cleared by a read of the port 3 control and status register (with IS3 FLAG set) followed by a read or write to the port 3 data register or during reset.

Port 3 In Expanded Non-Multiplexed Mode

Port 3 is configured as a bidirectional data bus (D7-D0) in the expanded non-multiplexed 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 (A0-A7) and data bus (D7-D0) in the expanded-multiplexed modes, where address strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high-impedance state between valid address and data to prevent bus conflicts.

PORT 4 (P40-P47)

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 can drive one Schottky TTL load and 90 pF and is the only port with internal pullup resistors. Unused lines can remain unconnected.

Port 4 In Single-Chip Mode

In single-chip mode, port 4 functions as an 8-bit I/O port 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. External pullup resistors to more than 5 volts, however, cannot be used.

Port 4 In Expanded Non-Multiplexed 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 mode 6, port 4 functions as half of the address bus and provides A8 to A15. In mode 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 MC6801 provides 2048 bytes of on-chip ROM and 128 bytes of on-chip RAM.

One half of the RAM is powered through the V_{CC} standby pin and is maintainable during V_{CC} powerdown. This standby portion of the RAM consists of 64 bytes located from $80 through $BF.

Power must be supplied to V_{CC} 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 determine the adequacy of the standby power source during powerdown operation. It is intended that RAME be cleared and STBY PWR be set as part of a powerdown procedure.

RAM CONTROL REGISTER

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

Bit 0-5   Not used.

Bit 6 RAME   RAM Enable. This read/write bit can be used to remove the entire RAM from the internal memory map. RAME is set (enabled) during reset provided standby power is available on the positive edge of $\overline{RESET}$. If RAME is clear, any access to a RAM address is external. If RAME is set and not in mode 3, the RAM is included in the internal map.

Bit 7 STBY PWR   Standby Power. This bit is a read/write status bit which, when once set, remains set as long as V_{CC} standby remains above V_{SBB} (minimum). As long as this bit is set following a period of standby operation, the standby power supply has adequately preserved the data in the standby RAM. If this bit is cleared during a period of standby operation, it indicates that V_{CC} standby had fallen to a level sufficiently below V_{SBB} (minimum) to suspect that data in the

standby RAM is not valid. This bit can be set only by software and is not affected during reset.

PROGRAMMABLE TIMER

The programmable timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in Figure 22.

COUNTER ($09:0A)

The key timer element is a 16-bit free-running counter which is incremented by E (enable). It is cleared during reset and is read-only with one exception: a write to the counter ($09) will preset it to $FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. TOF is set whenever the counter contains all ones.

OUTPUT COMPARE REGISTER ($0B:0C)

The output compare register is a 16-bit read/write register used to control an output waveform or provide an arbitrary timeout flag. It is compared with the free-running counter on each E cycle. When a match occurs, OCF is set and OLVL is clocked to an output level register. If port 2, bit 1, is configured as an output, OLVL will appear at P21 and the output compare register and OLVL can then be changed for the next

3

FIGURE 22 — BLOCK DIAGRAM OF PROGRAMMABLE TIMER
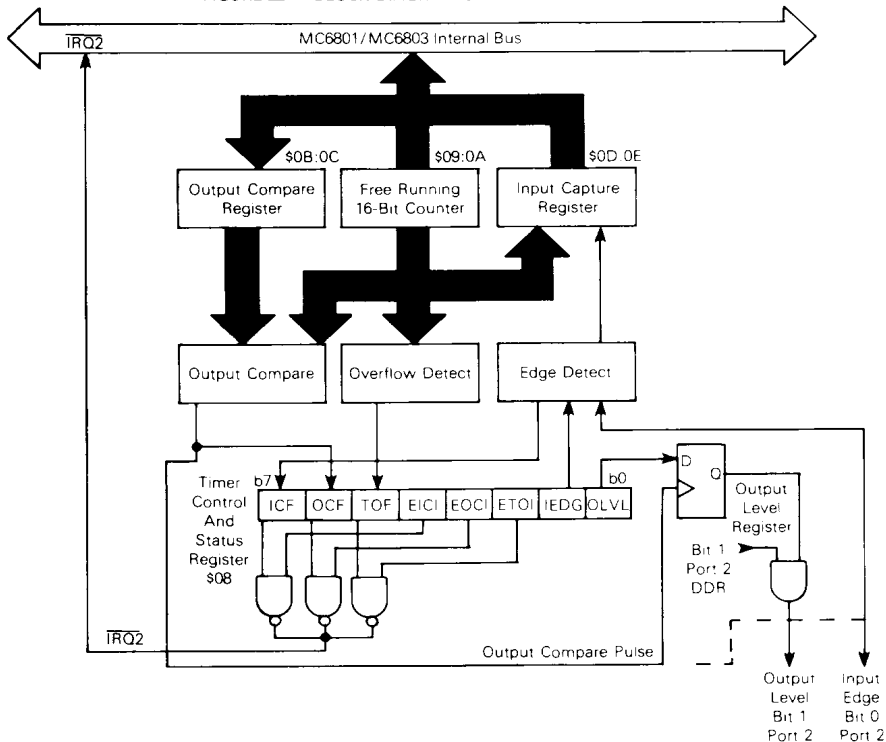
compare. The function is inhibited for one cycle after a write to its high byte ($0B) to ensure a valid compare. The output compare register is set to $FFFF at RESET.

INPUT CAPTURE REGISTER ($0D:0E)

The input capture register is a 16-bit read-only register used to store the free-running counter when a "proper" input transition occurs as defined by IEDG. Port 2, bit 0 should be configured as an input, but the edge detect circuit always senses P20 even when configured as an output. An input capture can occur independently of ICF: the register always contains the most current value. Counter transfer is inhibited, however, 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 REGISTER ($08)

The timer control and status register (TCSR) is an 8-bit register of which all bits are readable, while only bits 0-4 can be written. The three most-significant bits provide the timer status and indicate if:

- a proper level transition has been detected,
- a match has occurred between the free-running counter and the output compare register, and
- 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 (TCSR)

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

Bit 0 OLVL   Output Level. OLVL is clocked to the output level register by a successful output compare and will appear at P21 if bit 1 of the port 2 data direction register is set. It is cleared during reset.

Bit 1 EIDG   Input Edge. IEDG is cleared during reset and controls which level transition will trigger a counter transfer to the input capture register:
IEDG = 0 Transfer on a negative-edge
IEDG = 1 Transfer on a positive-edge.

Bit 2 ETOI   Enable Timer Overflow Interrupt. When set, an IRQ2 interrupt is enabled for a timer overflow; when clear, the interrupt is inhibited. It is cleared during reset.

Bit 3 EOCI   Enable Output Compare Interrupt. When set, an IRQ2 interrupt is enabled for an output compare; when clear, the interrupt is inhibited. It is cleared during reset.

Bit 4 EICI   Enable Input Capture Interrupt. When set, an IRQ2 interrupt is enabled for an input capture; when clear, the interrupt is inhibited. It is cleared during reset.

Bit 5 TOF   Timer Overflow Flag. TOF is set when the counter contains all ones. It is cleared by reading the TCSR (with TOF set) then reading the counter high byte ($09), or during reset.

Bit 6 OCF   Output Compare Flag. OCF is set when the output compare register matches the free-running counter. It is cleared by reading the TCSR (with OCF set) and then writing to the output compare register ($0B or $0C), or during reset.

Bit 7 ICF   Input Capture Flag. ICF is set to indicate a proper level transition; it is cleared by reading the TCSR (with ICF set) and then the input capture register high byte ($0D), or during reset.

SERIAL COMMUNICATIONS INTERFACE (SCI)

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

PROGRAMMABLE OPTIONS

The following features of the SCI are programmable:
- format: standard mark/space (NRZ) or Bi-phase
- clock: external or internal bit rate clock
- Baud: one of four per E clock frequency, or external clock (× 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 serial communications interface includes four addressable registers as depicted in Figure 23. It is controlled by the rate and mode control register and the transmit/ receive control and status register. Data is transmitted and

received utilizing a write-only transmit register and a read-only receive register. The shift registers are not accessible to software.

Rate and Mode Control Registers (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 four write-only bits which are cleared during reset. The two least-significant bits control the bit rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER (RMCR)

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

Bit 1:Bit 0   SS1:SS0 Speed Select. These two bits select the baud rate when using the internal clock. Four rates may be selected which are a function of the MCU input frequency. Table 6 lists bit time and rates for three selected MCU frequencies.

Bit 3:Bit 2   CC1:CC0 Clock Control 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.

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 can disturb serial operations.

3

FIGURE 23 — SCI REGISTERS
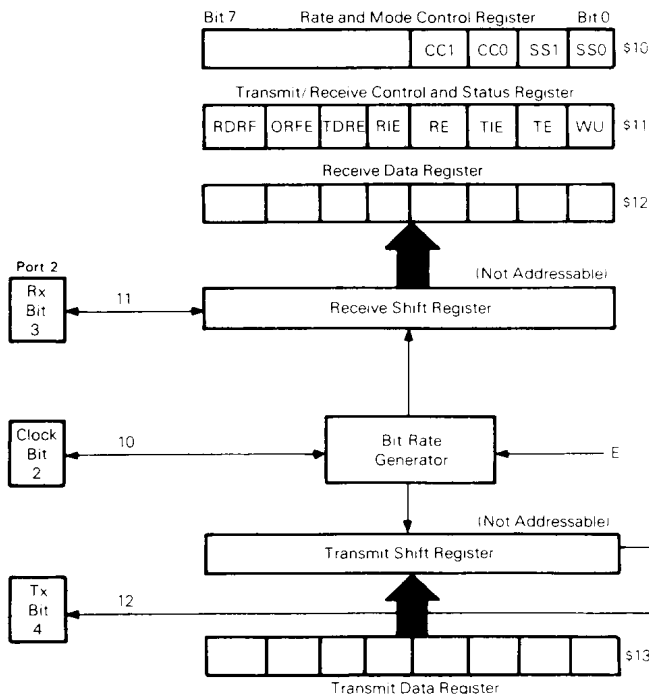
Transmit/Receive Control And Status Register (TRCSR) ($11)

The transmit/receive control and status register controls the transmitter, receiver, wake-up feature, and two individual interrupts and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to $20 by $\overline{RESET}$.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR)

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

Bit 0 WU — "Wake-up" on Idle Line. When set, WU enables the wake-up function; it is cleared by eleven consecutive ones or during reset. WU will not set if the line is idle.

Bit 1 TE — Transmit Enable. When set, P24 DDR bit is set, cannot be changed, and will remain set if TE is subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P24 and a preamble of nine consecutive ones is transmitted. TE is cleared during reset.

Bit 2 TIE — Transmit Interrupt Enable. When set, an $\overline{IRQ2}$ interrupt is enabled when TDRE is set; when clear, the interrupt is inhibited. TE is cleared during reset.

Bit 3 RE — 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 RIE — Receiver Interrupt Enable. When set, an $\overline{IRQ2}$ interrupt is enabled when

RDRF and/or ORFE is set; when clear, the interrupt is inhibited. RIE is cleared during reset.

Bit 5 TDRE — 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 ORFE — Overrun Framing Error. If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the receiver data register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the state of RDRF. if RDRF is set, then an overrun has occurred; otherwise a framing error has been detected. Data is not transferred to the receive data register in an overrun condition. Unframed data causing a framing error is 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 RDRF — Receive Data Register Full. RDRF is set when the input serial shift register is transferred to the receive data register. It is cleared by reading the TRCSR (with RDRF set), and then the receive data register, or during reset.

TABLE 6 — SCI BIT TIMES AND RATES

SS1:SS0		$4f_o \rightarrow$	2.4576 MHz	4.0 MHz	4.9152 MHz
		E	614.4 kHz	1.0 MHz	1.2288 MHz
0	0	÷ 16	26 µs/38,400 Baud	16 µs/62,500 Baud	13.0 µs/76,800 Baud
0	1	÷ 128	208 µs/4,800 Baud	128 µs/7812.5 Baud	104.2 µs/9,600 Baud
1	0	÷ 1024	1.67 ms/600 Baud	1.024 ms/976.6 Baud	833.3 µs/1,200 Baud
1	1	÷ 4096	6.67 ms/150 Baud	4.096 ms/244.1 Baud	3.33 ms/300 Baud
* External (P22)			13.0 µs/76,800 Baud	8.0 µs/125,000 Baud	6.5 µs/153,600 Baud

*Using maximum clock rate

TABLE 7 — SCI FORMAT AND CLOCK SOURCE CONTROL

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

* $\overline{Devices}$ made with mask number M5G, M8D, and T5P do not transfer unframed data to the receive data register.

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 exist: 1) if the transmit data register is empty (TDRE = 1), a continuous string of ones will be sent indicating an idle line, or 2) if a byte has been written to the transmit-data register (TDRE = 0), it 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 should occur, ones will be sent until more data is provided. In Bi-phase 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 24.

INSTRUCTION SET

The MC6801/03 is upward source and object code compatible with the MC6800. Execution times of key instructions have been reduced and several new instructions have been added, including a hardware multiply. A list of new operations added to the MC6800 instruction set is shown in Table 1.

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

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 MC6801/03 is shown in Figure 10. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most-significant byte. Any operation which modifies the double accumulator will also modify accumulator A and/or B. Other registers are defined as follows:

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

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

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

Accumulators — The 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 Registers — The condition code register indicates the results of an instruction and includes the following five condition bits: negative (N), zero (Z), overflow (V), carry/borrow from MSB (C), and half carry from bit 3 (H). These bits are testable by the conditional branch 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.

FIGURE 24 — SCI DATA FORMATS

Data: 01001101 ($4D)
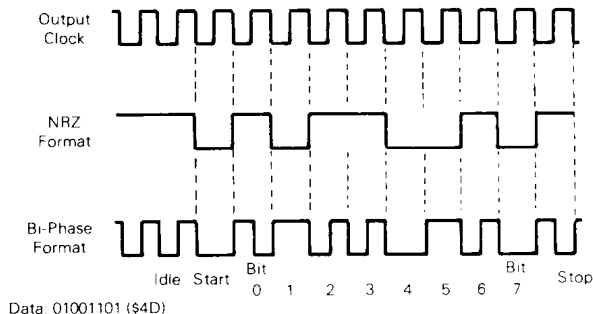
ADDRESSING MODES

Six addressing modes can be used to reference memory. A summary of addressing modes for all instructions is present in Tables 9 through 12, where execution times are provided in E cycles. Instruction execution times are summarized in Table 13. With an input frequency of 4 MHz, E cycles are equivalent to microseconds. A cycle-by-cycle description of bus activity for each instruction is provided in Table 14 and a description of selected instructions is shown in Figure 25.

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

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

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

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

Inherent Addressing — The operand(s) are registers 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 a signed single byte displacement in the second byte of the instruction and the current program counter. This provides a branch range of 126 to -129 bytes from the first byte of the instruction. These are two byte instructions.

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	D1	CMPB		3	2
02	•				36	PSHA		3	1	6A	DEC		6	2	9E	LDS		4	2	D2	SBCB		3	2
03	•				37	PSHB		3	1	6B	•				9F	STS	DIR	4	2	D3	ADDD		5	2
04	LSRD		3	1	38	PULX		5	1	6C	INC		6	2	A0	SUBA	NDXD	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		6	3	AC	CPX		6	2	E0	SUBB	NDXD	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	BA	ORAA		4	3	EE	LDX		5	2
1F	•				53	COMB		2	1	87	•				BB	ADDA		4	3	EF	STX	INDXD	5	2
20	BRA	REL	3	2	54	LSRB		2	1	88	EORA		2	2	BC	CPX		6	3	F0	SUBB	EXTND	4	3
21	BRN		3	2	55	•				89	ADCA		2	2	BD	JSR		6	3	F1	CMPB		4	3
22	BHI		3	2	56	RORB		2	1	8A	ORAA		2	2	BE	LDS		5	3	F2	SBCB		4	3
23	BLS		3	2	57	ASRB		2	1	8B	ADDA		2	2	BF	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		2	2	F5	BITB		4	3
26	BNE		3	2	5A	DECB		2	1	8E	LDS	IMMED	3	3	C2	SBCB		2	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		5	3
2D	BLT		3	2	61	•				95	BITA		3	2	C9	ADCB		2	2	FD	STD		5	3
2E	BGT		3	2	62	•				96	LDAA		3	2	CA	ORAB		2	2	FE	LDX		5	3
2F	BLE	REL	3	2	63	COM		6	2	97	STAA		3	2	CB	ADDB		2	2	FF	STX	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	•					• UNDEFINED OP CODE			
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: 1. Addressing Modes
INHER ≡ Inherent   INDXD ≡ Indexed   IMMED ≡ Immediate
REL ≡ Relative   EXTND ≡ Extended   DIR ≡ Direct
2. Unassigned opcodes are indicated by "•" and should not be executed.
3. Codes marked by "T" force the PC to function as a 16-bit counter.

TABLE 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

Pointer Operations	MNEM	Immed			Direct			Index			Extnd			Inherent			Boolean/Arithmetic Operation	Condition Codes					
																		5	4	3	2	1	0
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		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	$X - 1 \rightarrow X$	•	•	•	↕	•	•
Decrement Stack Pointer	DES													34	3	1	$SP - 1 \rightarrow SP$	•	•	•	•	•	•
Increment Index Register	INX													08	3	1	$X + 1 \rightarrow X$	•	•	•	↕	•	•
Increment Stack Pointer	INS													31	3	1	$1 \; SP + 1 \rightarrow SP$	•	•	•	•	•	•
Load Index Register	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				$M \rightarrow X_H,(M + 1) \rightarrow X_L$	•	•	↕	↕	R	•
Load Stack Pointer	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				$M \rightarrow SP_H,(M + 1) \rightarrow SP_L$	•	•	↕	↕	R	•
Store Index Register	STX				DF	4	2	EF	5	2	FF	5	3				$X_H \rightarrow M,X_L \rightarrow (M + 1)$	•	•	↕	↕	R	•
Store Stack Pointer	STS				9F	4	2	AF	5	2	BF	5	3				$SP_H \rightarrow M,SP_L \rightarrow (M + 1)$	•	•	↕	↕	R	•
Index Reg → Stack Pointer	TXS													35	3	1	$X - 1 \rightarrow SP$	•	•	•	•	•	•
Stack Pntr → Index Register	TSX													30	3	1	$SP + 1 \rightarrow X$	•	•	•	•	•	•
Add	ABX													3A	3	1	$B + X \rightarrow X$	•	•	•	•	•	•
Push Data	PSHX													3C	4	1	$X_L \rightarrow M_{SP},SP - 1 \rightarrow SP$ $X_H \rightarrow M_{SP},SP - 1 \rightarrow SP$	•	•	•	•	•	•
Pull Data	PULX													38	5	1	$SP + 1 \rightarrow SP \; M_{SP} \rightarrow X_H$ $SP + 1 \rightarrow SP \; M_{SP} \rightarrow X_L$	•	•	•	•	•	•

TABLE 10 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 1 of 2)
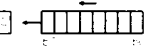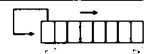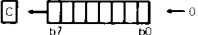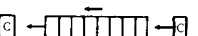
Accumulator and Memory Operations	MNEM	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
																		5	4	3	2	1	0
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Accumulators	ABA													1B	2	1	$A + B \rightarrow A$	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	$00:B + X \rightarrow X$	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				$A + M + C \rightarrow A$	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				$B + M + C \rightarrow B$	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				$A + M \rightarrow A$	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				$B + M \rightarrow B$	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				$D + M:M + 1 \rightarrow D$	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				$A \cdot M \rightarrow A$	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				$B \cdot M \rightarrow B$	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3				(diagram)	•	•	↕	↕	↕	↕
	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				(diagram)	•	•	↕	↕	↕	↕
	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				$00 \rightarrow M$	•	•	R	S	R	R
	CLRA													4F	2	1	$00 \rightarrow A$	•	•	R	S	R	R
	CLRB													5F	2	1	$00 \rightarrow B$	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				$A - M$	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				$B - M$	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M} \rightarrow M$	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A} \rightarrow A$	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B} \rightarrow B$	•	•	↕	↕	R	S

3

TABLE 10 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 2 of 2)

Accumulator and Memory Operations	MNEM	Immed Op	~	#	Direct Op	~	#	Index Op	~	#	Extend Op	~	#	Inher Op	~	#	Boolean Expression	H (5)	I (4)	N (3)	Z (2)	V (1)	C (0)
Decimal Adjust, A	DAA													19	2	1	Adj binary sum to BCD	•	•	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M − 1 → M	•	•	↕	↕	↕	•
	DECA													4A	2	1	A − 1 → A	•	•	↕	↕	↕	•
	DECB													5A	2	1	B − 1 → B	•	•	↕	↕	↕	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	↕	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	↕	•
Load Accumulators	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M M + 1 → D	•	•	↕	↕	R	•
Logical Shift, Left	LSL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	LSLA													48	2	1	C ← b7...b0 ← 0	•	•	↕	↕	↕	↕
	LSLB													58	2	1		•	•	↕	↕	↕	↕
	LSLD													05	3	2		•	•	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					•	•	R	↕	↕	↕
	LSRA													44	2	1	0 → b7...b0 → C	•	•	R	↕	↕	↕
	LSRB													54	2	1		•	•	R	↕	↕	↕
	LSRD													04	3	1		•	•	R	↕	↕	↕
Multiply	MUL													3D	10	1	A × B → D	•	•	•	•	•	↕
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 − M → M	•	•	↕	↕	↕	↕
	NEGA													40	2	1	00 − A → A	•	•	↕	↕	↕	↕
	NEGB													50	2	1	00 − B → B	•	•	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	•	•	•	•	•	•
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	3	1	A → Stack	•	•	•	•	•	•
	PSHB													37	3	1	B → Stack	•	•	•	•	•	•
Pull Data	PULA													32	4	1	Stack → A	•	•	•	•	•	•
	PULB													33	4	1	Stack → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	↕	↕
	ROLA													49	2	1	C ← b7...b0 ← C	•	•	↕	↕	↕	↕
	ROLB													59	2	1		•	•	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	↕	↕
	RORA													46	2	1	C → b7...b0 → C	•	•	↕	↕	↕	↕
	RORB													56	2	1		•	•	↕	↕	↕	↕
Subtract Accumulator	SBA													10	2	1	A − B → A	•	•	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A − M − C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B − M − C → B	•	•	↕	↕	↕	↕
Store Accumulators	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
	STD				DD	4	2	ED	5	2	FD	5	3				D → M:M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A − M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B − M → B	•	•	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D − M:M + 1 → D	•	•	↕	↕	↕	↕
Transfer Accumulator	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M − 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A − 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B − 00	•	•	↕	↕	R	R

The condition code register notes are listed after Table 12

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

Operations	MNEM	Direct Op	~	#	Relative Op	~	#	Index Op	~	#	Extend Op	~	#	Inherent Op	~	#	Branch Test	5 H	4 I	3 N	2 Z	1 V	0 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	BMI				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 25	•	•	•	•	•	•
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	See Special Operations-Figure 25	•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait For Interrupt	WAI													3E	9	1		•	•	•	•	•	•

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

Operations	MNEM	Inherent Op	~	#	Boolean Operation	5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	2	1	$0 \rightarrow C$	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	$0 \rightarrow I$	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	$0 \rightarrow V$	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	$1 \rightarrow C$	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	$1 \rightarrow I$	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	$1 \rightarrow V$	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	2	1	$A \rightarrow CCR$	↕	↕	↕	↕	↕	↕
CCR → Accumulator A	TPA	07	2	1	$CCR \rightarrow A$	•	•	•	•	•	•

LEGEND

Op Operation Code (Hexadecimal)
~   Number of MPU Cycles
M_{SP} 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

	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	●

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

SUMMARY OF CYCLE-BY-CYCLE OPERATION

Table 14 provides a detailed description of the information present on the address bus, data bus, and the read/write (R/W) line during each cycle of each instruction.

The information is useful in comparing actual with expected results during debug of both software and hardware as the program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. In general, instructions with the same addressing mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

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

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 1 of 5)

Address Mode and Instructions		Cycles	Cycle #	Address Bus	R/W Line	Data Bus
IMMEDIATE						
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
DIRECT						
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/W̄ Line	Data Bus
EXTENDED					
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	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Address of Operand
AND ORA		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		4	Address of Operand	1	Operand Data
CMP SUB					
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	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
LDD		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	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
STD		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	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
CLR ROL		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	.	Operand Address (High Order Byte)
ADDD		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/W̄ Line	Data Bus
INDEXED						
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	4	1	Opcode Address	1	Opcode
ADD	LDA		2	Opcode Address + 1	1	Offset
AND	ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT	SBC		4	Index Register Plus Offset	1	Operand Data
CMP	SUB					
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		5	1	Opcode Address	1	Opcode
LDX			2	Opcode Address + 1	1	Offset
LDD			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		5	1	Opcode Address	1	Opcode
STX			2	Opcode Address + 1	1	Offset
STD			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	6	1	Opcode Address	1	Opcode
ASR	NEG		2	Opcode Address + 1	1	Offset
CLR	ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM	ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC	TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC			6	Index Register Plus Offset	0	New Operand Data
CPX		6	1	Opcode Address	1	Opcode
SUBD			2	Opcode Address + 1	1	Offset
ADDD			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		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)

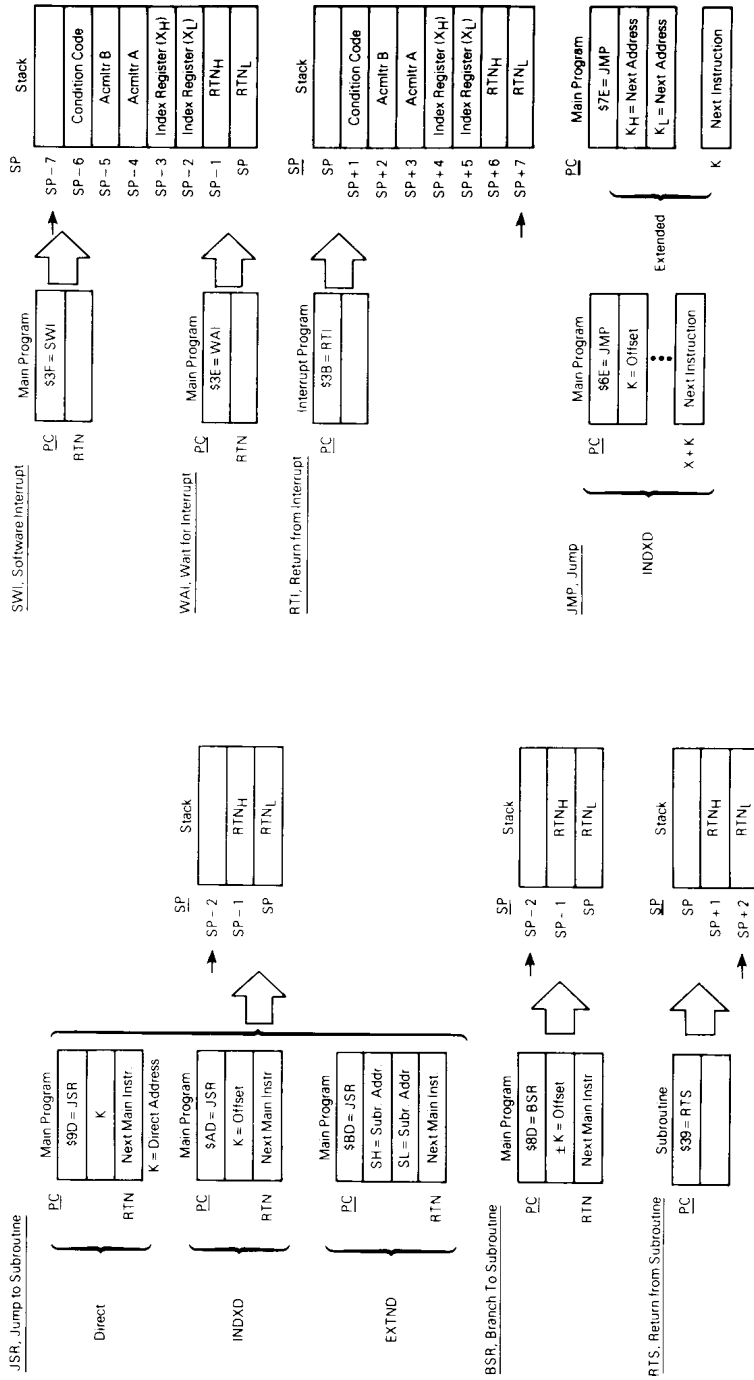*TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF

3

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 4 of 5)

Address Mode and Instructions			Cycles	Cycle #	Address Bus	R/W̄ Line	Data Bus
INHERENT							
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/W̄ Line	Data Bus
INHERENT					
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)
RELATIVE					
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 Buss 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

FIGURE 25 — SPECIAL OPERATIONS

JSR, Jump to Subroutine

Direct

PC	Main Program
	$9D = JSR
	K
RTN	Next Main Instr
	K = Direct Address

INDXD

PC	Main Program
	$AD = JSR
	K = Offset
RTN	Next Main Instr

EXTND

PC	Main Program
	$BD = JSR
	SH = Subr. Addr.
	SL = Subr. Addr
RTN	Next Main Inst

	Stack
SP − 2	RTN_H
SP − 1	RTN_L
SP	

BSR, Branch To Subroutine

PC	Main Program
	$8D = BSR
	± K = Offset
RTN	Next Main Instr

	Stack
SP − 2	RTN_H
SP − 1	RTN_L
SP	

RTS, Return from Subroutine

PC	Subroutine
	$39 = RTS

	Stack
SP	RTN_H
SP + 1	RTN_L
SP + 2	

SWI, Software Interrupt

PC	Main Program
	$3F = SWI
RTN	

	Stack
SP − 7	
SP − 6	Condition Code
SP − 5	Acmltr B
SP − 4	Acmltr A
SP − 3	Index Register (X_H)
SP − 2	Index Register (X_L)
SP − 1	RTN_H
SP	RTN_L

WAI, Wait for Interrupt

PC	Main Program
	$3E = WAI
RTN	

RTI, Return from Interrupt

PC	Interrupt Program
	$3B = RTI

	Stack
SP	
SP + 1	Condition Code
SP + 2	Acmltr B
SP + 3	Acmltr A
SP + 4	Index Register (X_H)
SP + 5	Index Register (X_L)
SP + 6	RTN_H
SP + 7	RTN_L

JMP, Jump

Extended

PC	Main Program
	$7E = JMP
	K_H = Next Address
	K_L = Next Address

K	Next Instruction

INDXD

PC	Main Program
	$6E = JMP
	K = Offset
X + K	Next Instruction

Legend
RTN = Address of next instruction in Main Program to be executed upon return from subroutine
RTN_H = Most significant byte of Return Address
RTN_L = Least significant byte of Return Address
→ = Stack Pointer After Execution
K = 8 bit Unsigned Value

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) 2516, 2716, MC68701
To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, sales person, or a 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 .LO 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. It is necessary to include the entire memory image of both program and data space. All unused bytes, including those in the user space, must be set in logic zero.

PC-DOS Disk File

PC-DOS is the IBM® Personal Computer Disk Operating System. Disk media submitted 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

A single 2K EPROM is necessary to contain the entire MC6801 program. The EPROM is programmed with the customer program using positive logic sense for address and data. All unused bytes, including the user's space, must be set to zero.

If the MC6801 MCU ROM pattern is submitted on a single 2516 or 2716 type EPROM, memory map addressing is one-for-one. The data space ROM runs from EPROM address $000 to $7FF. If an MC68701 is used, the ROM map runs from $F800 to $FFFF.

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 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.
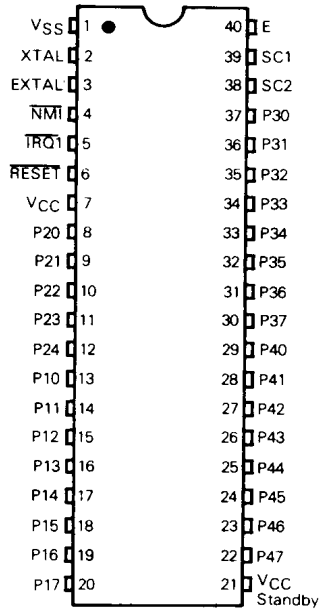
Ordering Information

The following table provides generic information pertaining to the package type and temperature for the MC6801/MC6803. This MCU device is available only in the 40-pin dual-in-line (DIP) package in the Cerdip and Plastic packages.

GENERIC INFORMATION

Frequency (MHz)	Temperature (Degrees C)	Cerdip Package (S Suffix)	Plastic Package (P Suffix)
1.0	0 to 70	MC6801S1	MC6801P1
1.0	−40 to +85	MC6801CS1	MC6801CP1
1.25	0 to 70	MC6801S1-1	MC6801P1-1
1.25	−40 to +85	MC6801CS-1	MC6801CP-1
2.0	0 to 70	MC68B01S1	MC68B01P1
1.0	0 to 70	MC6803S	MC6803P
1.0	−40 to +85	MC6803CS	MC6803CP
1.25	0 to 70	MC6803S-1	MC6803P-1
1.25	−40 to +85	MC6803CS-1	MC6803CP-1
2.0	0 to 70	MC68B03S	MC68B03P

PIN ASSIGNMENT

```
              ┌───────⌣───────┐
    Vss  ┤ 1  ●          40 ├ E
   XTAL  ┤ 2             39 ├ SC1
  EXTAL  ┤ 3             38 ├ SC2
    NMI  ┤ 4             37 ├ P30
   IRQ1  ┤ 5             36 ├ P31
  RESET  ┤ 6             35 ├ P32
    Vcc  ┤ 7             34 ├ P33
    P20  ┤ 8             33 ├ P34
    P21  ┤ 9             32 ├ P35
    P22  ┤ 10            31 ├ P36
    P23  ┤ 11            30 ├ P37
    P24  ┤ 12            29 ├ P40
    P10  ┤ 13            28 ├ P41
    P11  ┤ 14            27 ├ P42
    P12  ┤ 15            26 ├ P43
    P13  ┤ 16            25 ├ P44
    P14  ┤ 17            24 ├ P45
    P15  ┤ 18            23 ├ P46
    P16  ┤ 19            22 ├ P47
    P17  ┤ 20            21 ├ Vcc
              └───────────────┘  Standby
```