MOTOROLA
SEMICONDUCTORS
3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721

MC6803E

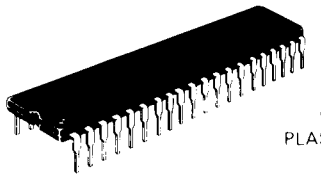
Advance Information

HMOS
(HIGH-DENSITY N-CHANNEL, SILICON-GATE)

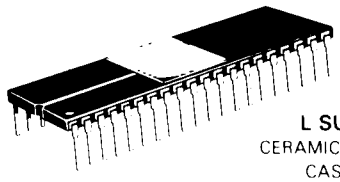
8-BIT MICROPROCESSOR

8-BIT MICROPROCESSOR

The MC6803E is an 8-bit microprocessing unit (MPU) designed for uses in which the internal clock needs to be synchronized with systems, peripherals, or other MPUs. The MC6803E also supports DMA and dynamic RAM refresh with its halt ($\overline{HALT}$) and bus available (BA) pins. The MC6803E has all the features of the MC6801 microcomputer unit except on-chip ROM and an on-chip oscillator. These on-chip features include 128 bytes of RAM, a serial communications interface (SCI), parallel I/O, and a three-function programmable timer. The MC6803E has the same enhanced MC6800 features as the MC6801, which include 64K addresss space, two 8-bit accumulators (which can be concatenated into one 16-bit accumulator), and the enhanced instruction set, as well as extra internal interrupts.



G SUFFIX
PLASTIC PACKAGE
CASE 711



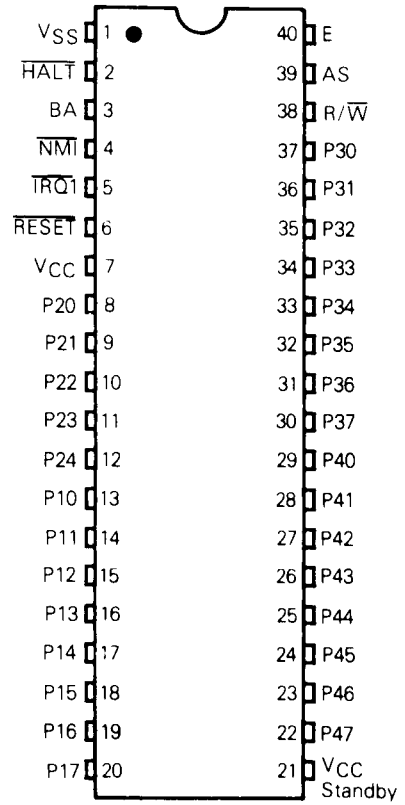
L SUFFIX
CERAMIC PACKAGE
CASE 715

- Enhanced MC6800 Instruction Set
- Upward Source and Object Code Compatible with the MC6800
- Bus Compatible with the M6800 Family
- Direct Source and Object Code Compatible with the MC6801
- 8 × 8 Multiply Instruction
- 64K Memory Map (Unused High Order Address Lines Can Be Used as Input Lines)
- External Clock Inputs (E and AS) Allow Synchronization
- DMA Capability (Clock Stretching) with $\overline{HALT}$ and BA Pins
- Serial Communications Interface (SCI)
- 16-Bit, Three-Function Programmable Timer
- 128 Bytes of RAM
- 64 Bytes of RAM Retainable During Power Down
- Pin-for-Pin Compatible with MC6801 Except for $\overline{HALT}$ and BA Pins

ORDERING INFORMATION ($T_A = 0°C$ to $70°C$)

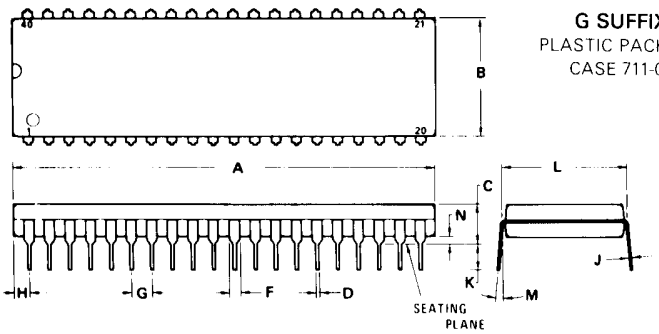
Package Type	Frequency	Order Number
Plastic	1.0 MHz	MC6803EG
G Suffix	1.25 MHz	MC6803EG-1
Ceramic	1.0 MHz	MC6803EL
L Suffix	1.25 MHz	MC6803EL-1

PIN ASSIGNMENT



This document contains information on a new product. Specifications and information herein

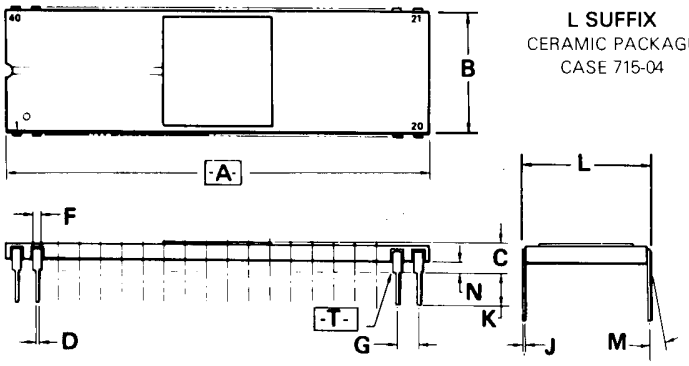
MECHANICAL DATA



G SUFFIX
PLASTIC PACKAGE
CASE 711-03

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0^0	15^0	0^0	15^0
N	0.51	1.02	0.020	0.040

NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.



L SUFFIX
CERAMIC PACKAGE
CASE 715-04

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	50.29	51.31	1.980	2.020
B	14.63	15.49	0.576	0.610
C	2.79	4.32	0.110	0.170
D	0.38	0.53	0.015	0.021
F	0.76	1.52	0.030	0.060
G	2.54 BSC		0.100 BSC	
J	0.20	0.33	0.008	0.013
K	2.54	4.57	0.100	0.180
L	14.99	15.65	0.590	0.616
M	–	10^0	–	10^0
N	1.02	1.52	0.040	0.060

NOTES:
1. DIMENSION -A- IS DATUM.
2. POSITIONAL TOLERANCE FOR LEADS:

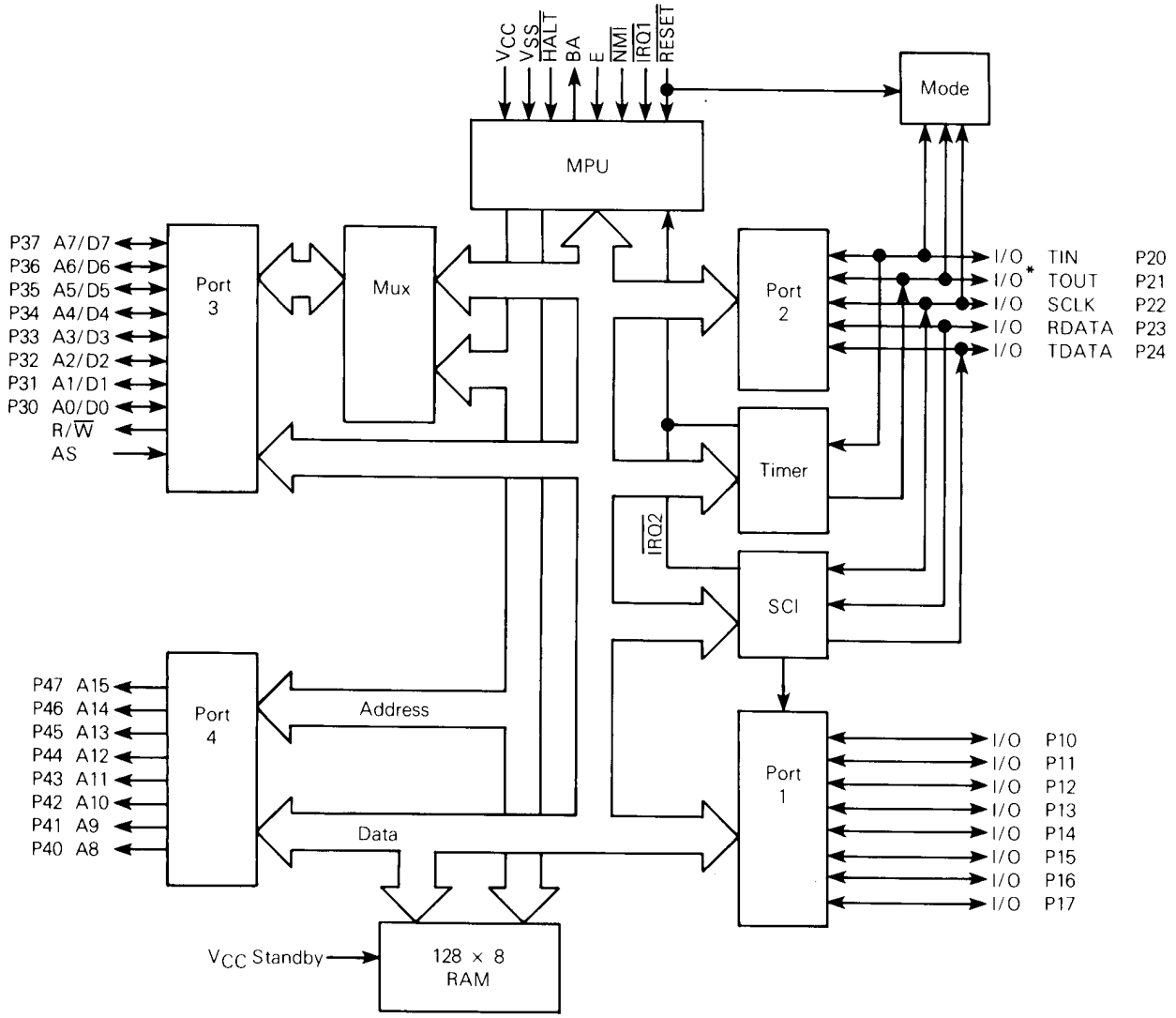
   ⌖ 0.25 (0.010) Ⓜ T A Ⓜ

3. -T- IS SEATING PLANE.
4. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

MOTOROLA *Semiconductor Products Inc.*

3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721 • A SUBSIDIARY OF MOTOROLA INC.

BLOCK DIAGRAM



*The output at this pin (P21) comes from the timer and not a data register.

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	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

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

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 that 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}.

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{PORT} \equiv$ Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} \ll P_{INT}$ 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:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

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

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

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, T_A = 0°C to 70°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	E	V_{EIH}	V_{CC} – 0.75		V_{CC}	V
Input Low Voltage	E	V_{EIL}	V_{SS} – 0.3		V_{SS} + 0.6	V
Input High Voltage	$\overline{RESET}$	V_{IH}	V_{SS} + 4.0		V_{CC}	V
	Other Inputs*		V_{SS} + 2.0		V_{CC}	
Input Low Voltage	All Inputs*	V_{IL}	V_{SS} – 0.3		V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	$\overline{HALT}$, AS, $\overline{NMI}$, $\overline{IRQ1}$, $\overline{RESET}$	I_{in}		1.5	2.5	μA
Hi Z Input Current (V_{in} = 0.5 to 2.4 V)	P10-P17, P20-P24, P30-P37	$\lvert I_{TSI} \rvert$	–	2.0	10	μA
Output High Voltage (I_{load} = –100 μA, V_{CC} = min)	All Outputs	V_{OH}	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
Darlington Drive Current (V_O = 1.5 V)	P10-P17	I_{OH}	1.0	1.5	5.0	mA
Internal Power Dissipation (Measured at T_A = 0°C in Steady-State Operation)		P_{INT}	–		1200	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f_o = 1.0 MHz)	P30-P37, AS	C_{in}	–		12.5	pF
	Other Inputs (Except E)		–		10.0	
V_{CC} Standby	Power Down	V_{SBB}	4.0		5.25	V
	Power Up	V_{SB}	4.75		5.25	
Standby Current	Power Down	I_{SBB}	–		6.0	mA

*Except mode programming levels; see Figure 8.

PERIPHERAL PORT TIMING (Refer to Figures 1 and 2)

Characteristics	Symbol	Min	Typ	Max	Unit
Peripheral Data Setup Time	t_{PDSU}	200	–		ns
Peripheral Data Hold Time	t_{PDH}	200	–	-	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid Ports 1, 2	t_{PWD}	–	–	350	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t_{CMOS}	–	–	2.0	μs

FIGURE 1 – DATA SETUP AND HOLD TIMES (MPU READ)

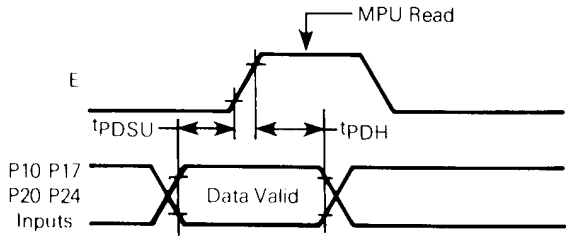
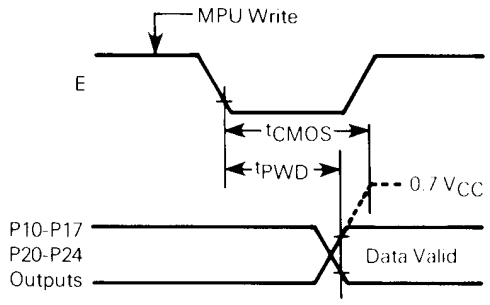


FIGURE 2 – DATA SETUP AND HOLD TIMES (MPU WRITE)



NOTES:
1. 10 k pullup resistor required for port 2 to reach 0.7 V_{CC}.
2. Not applicable to P21.

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

FIGURE 3 – CMOS LOAD

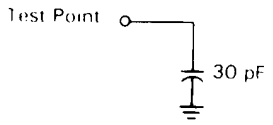
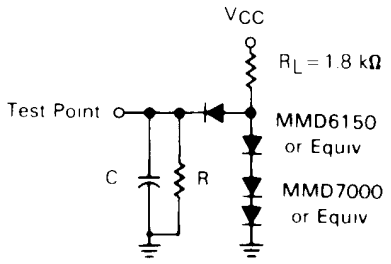


FIGURE 4 – TIMING TEST LOAD PORTS 1, 2, 3, 4

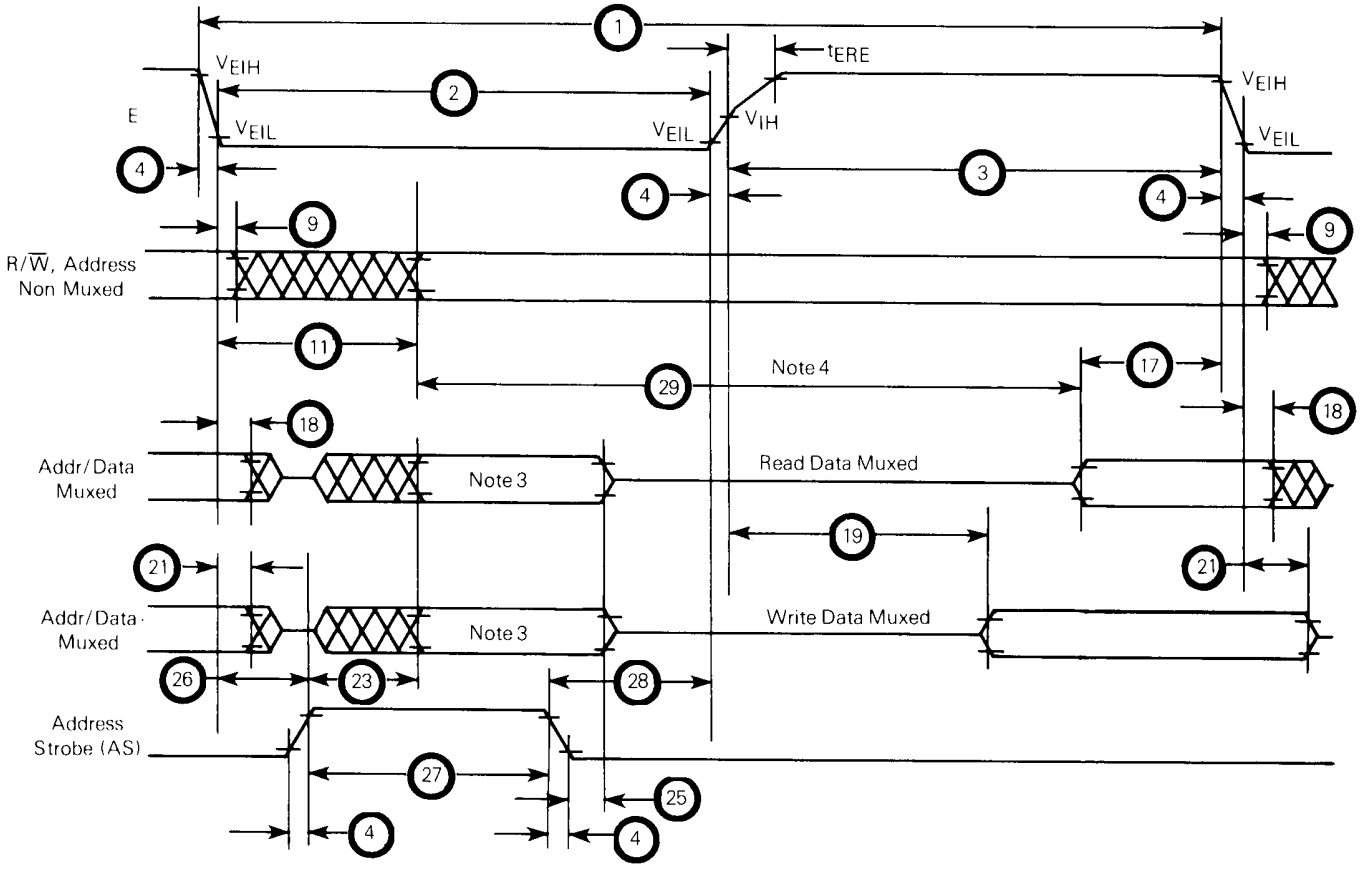


C = 90 pF for P30-P37, P40-P47, R/$\overline{W}$
= 30 pF for P10-P17, P20-P24, BA
R = 24 kΩ for P10-P17, P20-P24, P30-P37, P40-P47, R/$\overline{W}$, BA

BUS TIMING (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	MC6803E		MC6803E-1		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	2.0	0.8	2.0	μs
2	Pulse Width, E Low	PW_{EL}	430	1000	360	1000	ns
3	Pulse Width, E High	PW_{EH}	450	1000	360	1000	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	ns
9	Non-Muxed Address Hold Time	t_{AH}	20	–	20	–	ns
11	Address Delay From E Low	t_{AD}	–	260	–	220	ns
17	Read Data Setup Time	t_{DSR}	80	–	70	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	ns
19	Write Data Delay Time	t_{DDW}	–	225	–	200	ns
21	Write Data Hold Time	t_{DHW}	20	–	20	–	ns
23	Muxed Address Delay from AS	t_{ADM}	–	90	–	70	ns
25	Muxed Address Hold Time	t_{AHL}	20	–	20	–	ns
26	Delay Time E to AS Rise	t_{ASD}	100	–	80	–	ns
27	Pulse Width, AS High	PW_{ASH}	220	–	170	–	ns
28	Delay Time AS to E Rise	t_{ASED}	100	–	80	–	ns
29	Usable Access Time (See Note 4)	t_{ACC}	635	–	485	–	ns
	Enable Rise Time Extended	t_{ERE}	–	80	–	80	ns
	Processor Control Setup Time	t_{PCS}	200	–	200	–	ns
	Processor Control Hold Time	t_{PCH}	20	40	20	40	ns
	Bus Available Delay Time from Enable Low	t_{BA}	0	300	0	300	ns
	$\overline{HALT}$ Rise and Fall Time	t_{PCf}, t_{PCr}	0	100	0	100	ns

FIGURE 5 – BUS TIMING DIAGRAM



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. Address valid on the occurrence of the latter of 11 or 23.
4. Usable access time is computed by smaller of 1 – (4 + 11 + 17) or 1 – (4 + 17 + 23 + 26).

INTRODUCTION

The MC6803E is an MC6801 microcomputer unit without the internal oscillator or the on-chip ROM. The MC6803E is used in the applications in which synchronization to another device or system is needed, or in which clock stretching is a requirement (i.e., direct memory access or dynamic RAM refresh). At reset, the MC6803E is configured into one of two operating modes to control the various functions associated with the memory map. These operating modes are the expanded multiplexed modes of the MC6801 (2 and 3).

The MC6803E has one 10-bit port, two 8-bit ports, and one 5-bit port. Each port except port 3 and port 4 consists of at least a write-only data direction register and a data 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 an "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 3 functions as a time multiplexed address/data bus and does not contain either a data direction register or a data register. Port 4 functions as a non-multiplexed high order address bus and does not contain either a data direction register or a data register. Port pins are labeled as Pij, where i identifies one of four ports and j indicates the particular bit.

The MC6803E is an enhanced MC6800 MPU with additional capabilities and greater throughput. It is directly source and object code compatible with the MC6801 and upward source and object code compatible with the MC6800. The programming model is shown in Figure 6. A list of the new instructions available on the MC6803E, in addition to the M6800 instruction set, are given in Table 1.

FIGURE 6 — PROGRAMMING MODEL

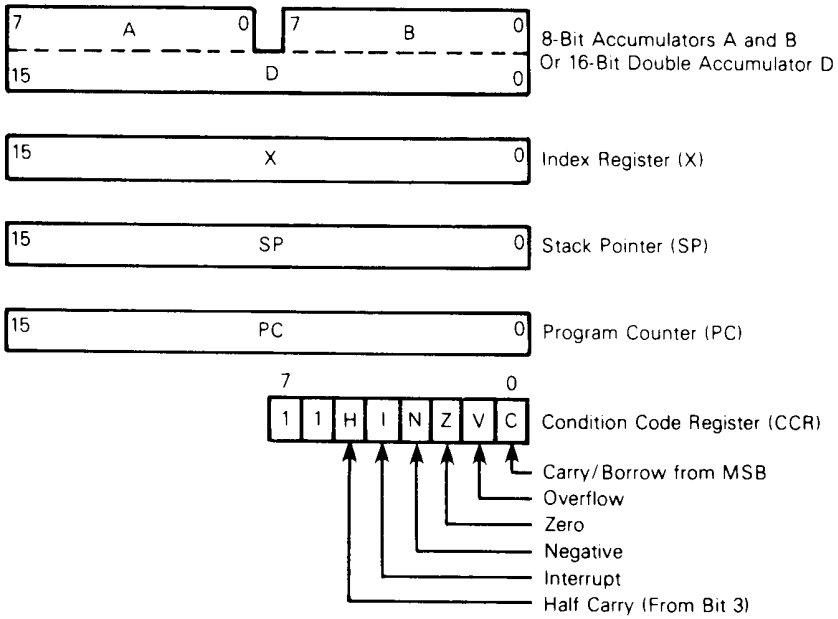


TABLE 1 — NEW INSTRUCTIONS

Instruction	Description
ABX	Unsigned addition of accumulator B to index register
ADDD	Adds (without carry) the double accumulator to memory and leaves the sum in the double accumulator
ASLD or LSLD	Shifts the double accumulator left (towards MSB) one bit; the LSB is cleared and the MSB is shifted into the C bit
BHS	Branch if higher or same; unsigned conditional branch (same as BCC)
BLO	Branch if lower; unsigned conditional branch (same as BCS)
BRN	Branch never
JSR	Additional addressing mode: direct
LDD	Loads double accumulator from memory
LSL	Shifts memory or accumulator left (towards MSB) one bit; the LSB is cleared and the MSB is shifted into the C bit (same as ASL)
LSRD	Shifts the double accumulator right (towards LSB) one bit; the MSB is cleared and the LSB is shifted into the C bit
MUL	Unsigned multiply; multiplies the two accumulators and leaves the product in the double accumulator
PSHX	Pushes the index register to stack
PULX	Pulls the index register from stack
STD	Stores the double accumulator to memory
SUBD	Subtracts memory from the double accumulator and leaves the difference in the double accumulator
CPX	Internal processing modified to permit its use with any conditional branch instruction

OPERATING MODES

The MC6803E has two operating modes (modes 2 and 3). The operating modes are hardware selectable, determining the device memory map. The mode numbers are referred to as 2 and 3 for consistency with the MC6801 and because that is the binary value applied to the mode programming pins during reset. (See **PROGRAMMING THE MODE.**)

A 64K byte memory space is available in both operating modes. In modes 2 and 3, port 4 provides address lines A8 to A15.

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. Address strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in Figure 7. This allows port 3 to function as a data bus when E is high.

Figure 8 depicts a typical operating configuration.

PROGRAMMING THE MODE

The operating mode is determined at reset by the levels asserted on P20 and P21. These levels are lached into the PC1 and PC0 bit locations of the program control register on the positive edge of $\overline{\text{RESET}}$. The operating mode may be read from the port 2 data register as shown below, and programming levels and timing must be met as shown in Figure 9. Characteristics and a brief outline of the operating modes are shown in Tables 2 and 3.

PORT 2 DATA REGISTER

7	6	5	4	3	2	1	0	
0	PC1	PC0	P24	P23	P22	P21	P20	$03

Circuitry to provide the programming levels is dependent primarily on the normal system usage of P20 and P21. If configured as outputs, the circuit shown in Figure 10 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

FIGURE 7 — TYPICAL LATCH ARRANGEMENT

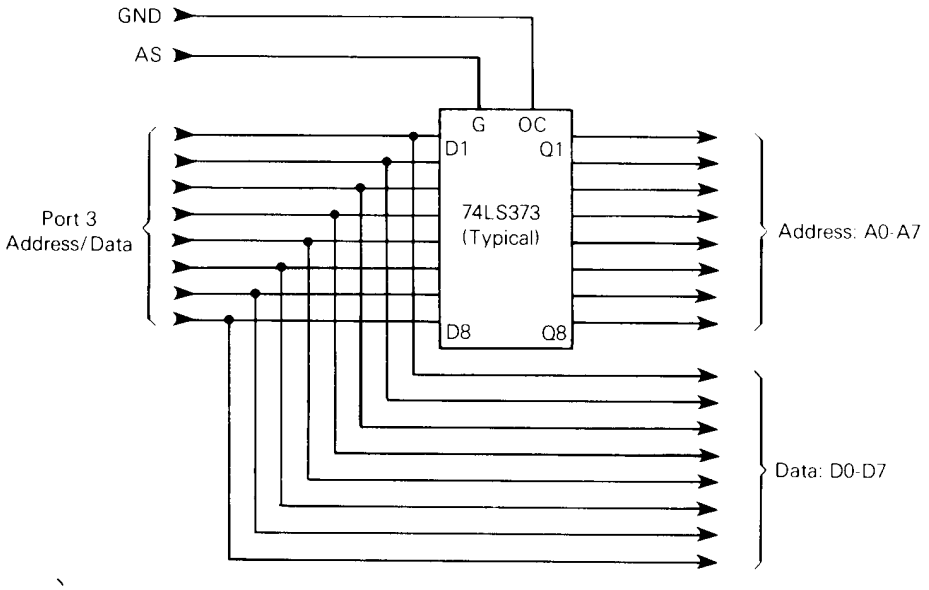
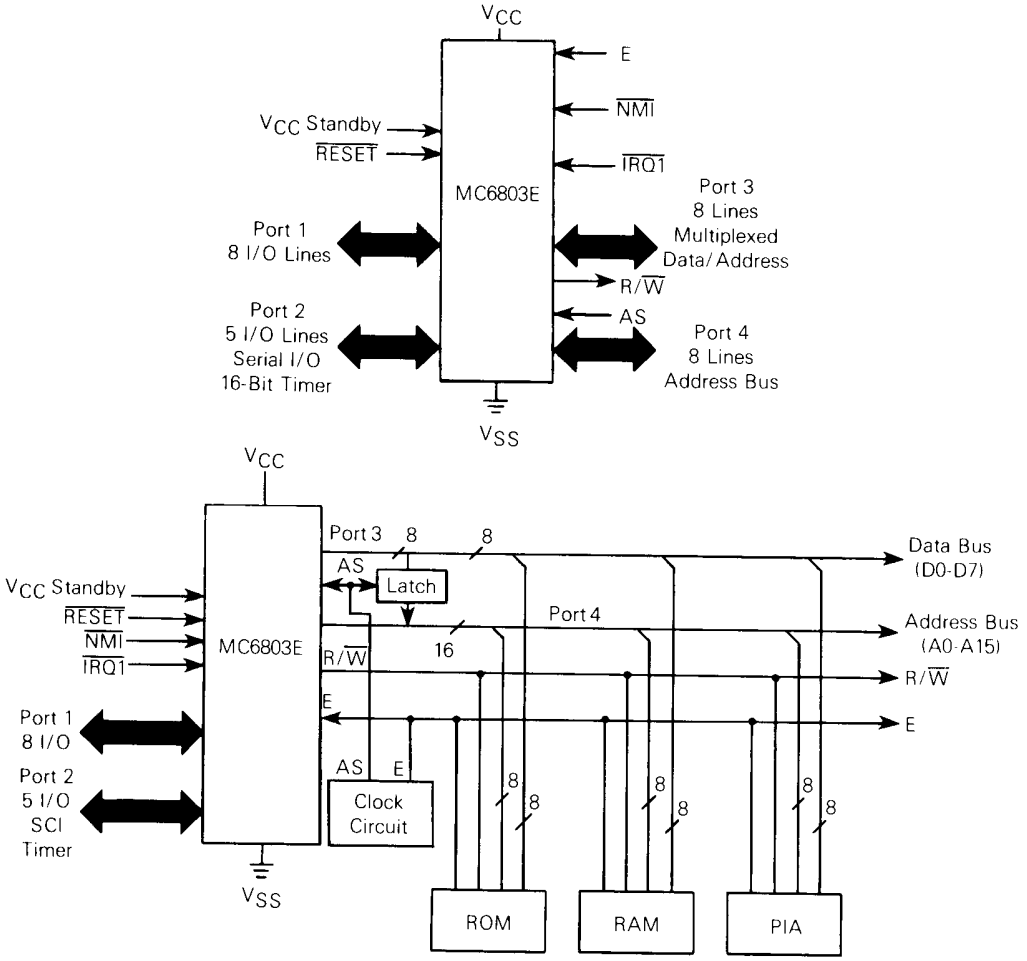
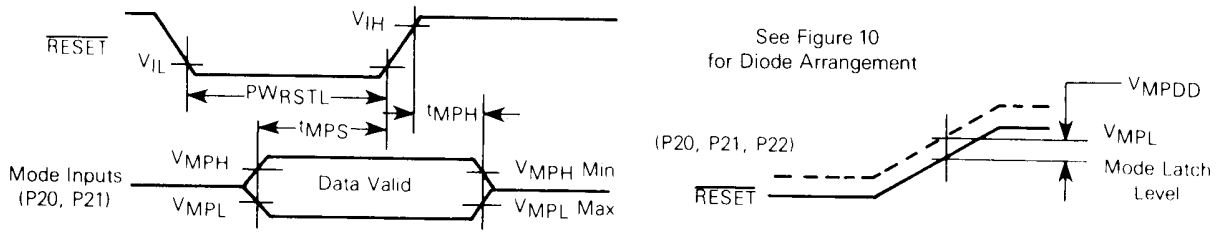


FIGURE 8 — EXPANDED MULTIPLEXED CONFIGURATION



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

FIGURE 9 — MODE PROGRAMMING TIMING



MODE PROGRAMMING (Refer to Figure 9)

Characteristic	Symbol	Min	Max	Unit
Mode Programming Input Voltage Low	V_{MPL}	–	1.8	V
Mode Programming Input Voltage High	V_{MPH}	4.0	–	V
Mode Programming Diode Differential (If Diodes are Used)	V_{MPDD}	0.6	–	V
$\overline{RESET}$ Low Pulse Width	PW_{RSTL}	3.0		E Cycles
Mode Programming Setup Time	t_{MPS}	2.0	–	E Cycles
Mode Programming Hold Time $\overline{RESET}$ Rise Time ≥ 1 μs $\overline{RESET}$ Rise Time < 1 μs	t_{MPH}	0 100		ns

TABLE 2 — SUMMARY OF MC6803E OPERATING MODES

Memory Space Options (64K Address Space)
Mode 2 — Internal RAM
Mode 3 — No Internal RAM

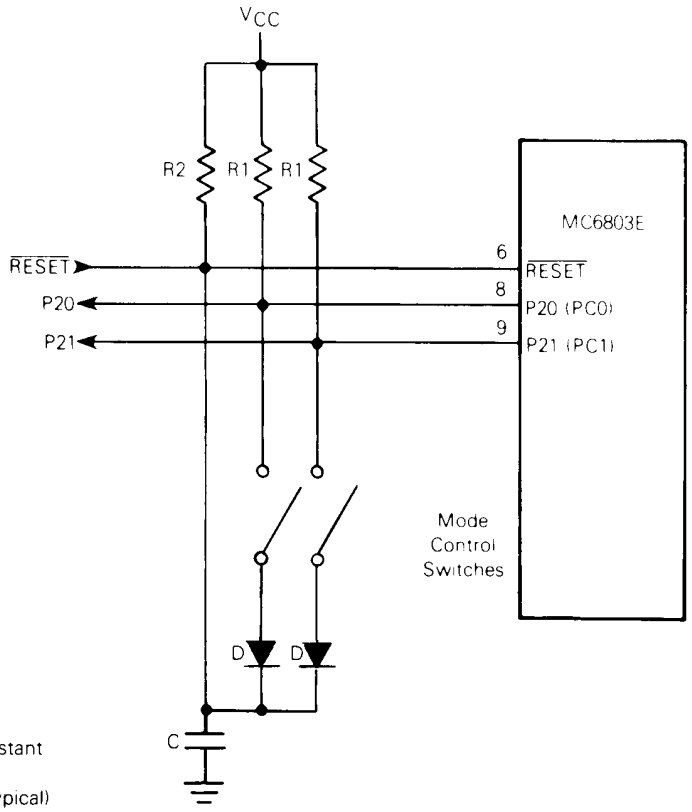
TABLE 3 — MODE SELECTION SUMMARY

Mode	P21 PC1	P20 PC0	RAM	Interrupt Vectors	Bus Mode	Operating Mode
3	H	H	E	E	MUX	Multiplexed/No RAM
2	H	L	I	E	MUX	Multiplexed/RAM
1	L	H				Undefined*
0	L	L				Undefined*

Legend:
I — Internal L — Logic 0
E — External H — Logic 1 MUX — Multiplexed

* These modes are undefined for the MC6803E; device should not be operated in these modes.

FIGURE 10 — TYPICAL MODE PROGRAMMING CIRCUIT



NOTES:
1. Mode 3 as shown
2. R2•C = reset time constant
3. R1 = 10 k (typical)
4. D = 1N914, 1N4001 (typical)
5. Diode V_f should not exceed V_{MPDD} min.

MEMORY MAPS

The MC6803E can provide up to 64K bytes of address space. A memory map for each operating mode is shown in Figure 11. The first 32 locations of each map are reserved for the internal register area, as shown in Table 4, with exceptions as indicated.

FIGURE 11 — MC6803E MEMORY MAPS

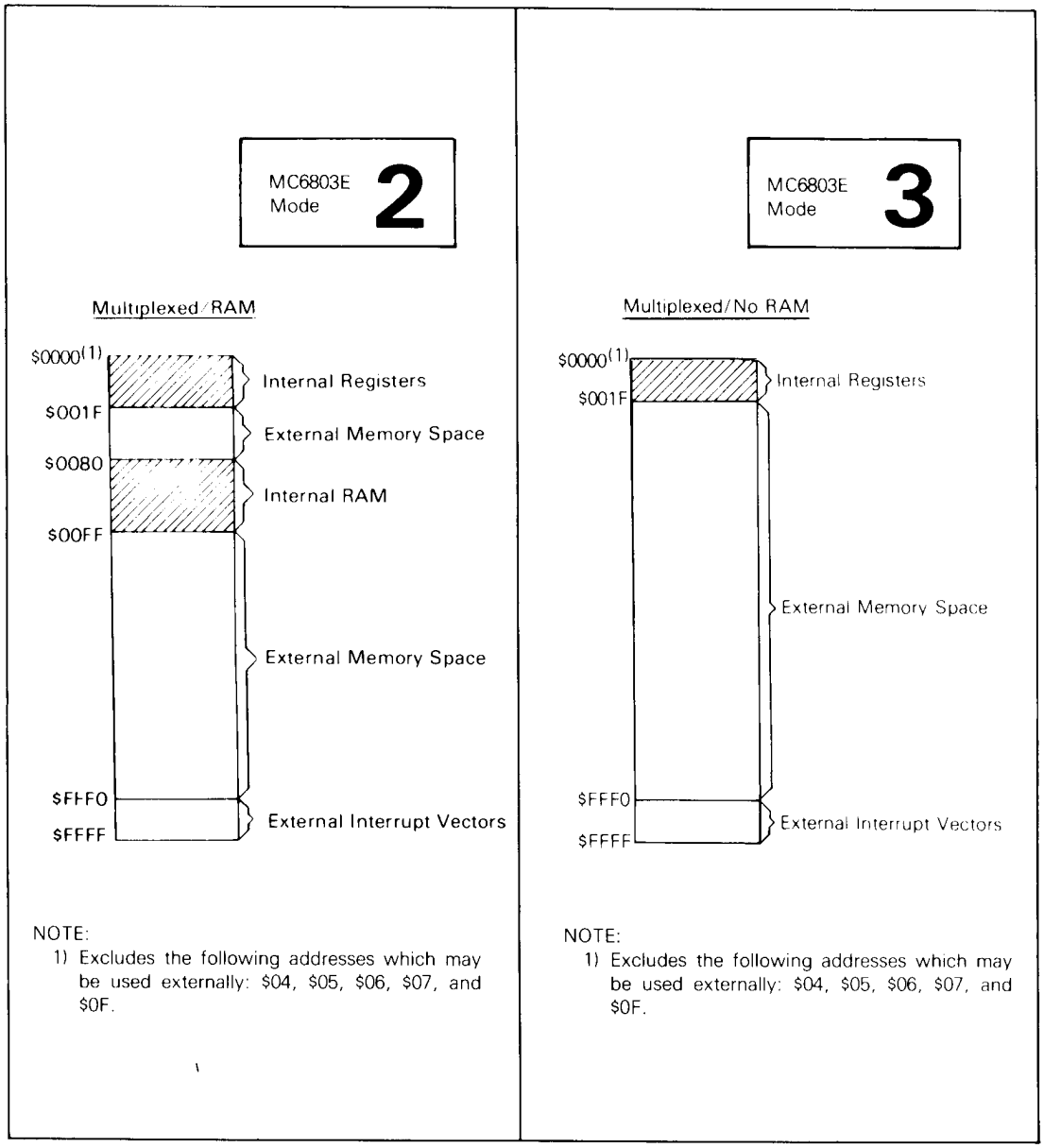


TABLE 4 — INTERNAL REGISTER AREA

Register	Address (Hex)
Port 1 Data Direction Register*	00
Port 2 Data Direction Register*	01
Port 1 Data Register	02
Port 2 Data Register	03
External Memory	04
External Memory	05
External Memory	06
External Memory	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
External Memory	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

*1 = Output, 0 = Input

MC6803E INTERRUPTS

The MC6803E supports two types of interrupt requests: maskable and non-maskable. A non-maskable interrupt ($\overline{NMI}$) 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: $\overline{IRQ1}$ and $\overline{IRQ2}$. The programmable timer and serial communications interface use an internal $\overline{IRQ2}$ interrupt line, as shown in the block diagram. External devices use $\overline{IRQ1}$. An $\overline{IRQ1}$ interrupt is serviced before $\overline{IRQ2}$ if both are pending.

All $\overline{IRQ2}$ 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 12 and is common to every interrupt excluding reset. During interrupt servicing, the program counter, A accumulator, B accumulator, and condition code register are pushed onto 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 $\overline{RESET}$ timing are illustrated in Figures 13 and 14.

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

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

*$\overline{IRQ2}$ Interrupt

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

V_{CC} and V_{SS} provide power to a large portion of the MPU. The power supply should provide +5 volts (± 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 power-up or power-down state. In the power-up state, the power supply should provide +5 volts (± 5%) and must reach V_{SB} volts before $\overline{RESET}$ reaches 4.0 volts. During power down, V_{CC} standby must remain above V_{SBB} (minimum) to sustain the standby RAM and STBY PWR bit. While in power-down 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 power-down operation. V_{CC} standby should be tied to ground in mode 3.

AS (ADDRESS STROBE)

Address strobe is an input strobe used to strobe out the least significant byte of an address on the 8 bit multiplexed bus. The AS line is used to demultiplex the eight least significant bits from the data bus.

FIGURE 12 — INTERRUPT FLOWCHART

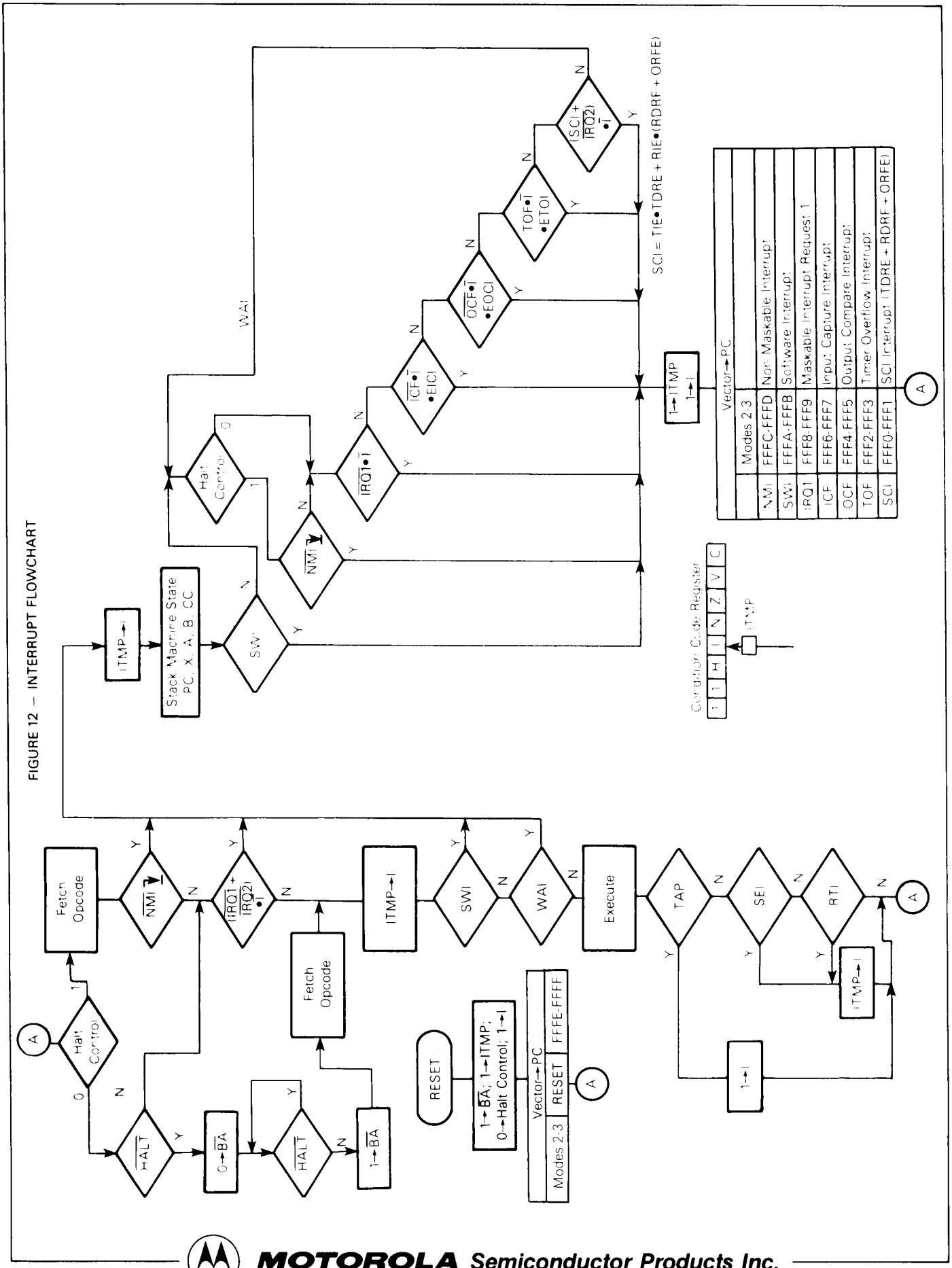


FIGURE 13 — INTERRUPT SEQUENCE

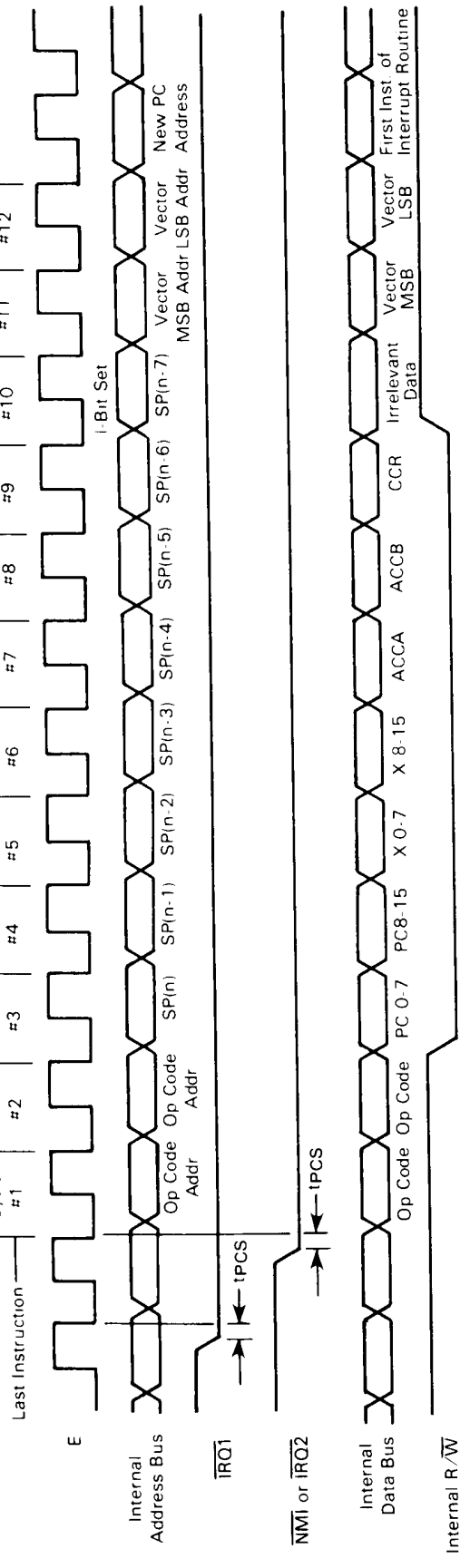
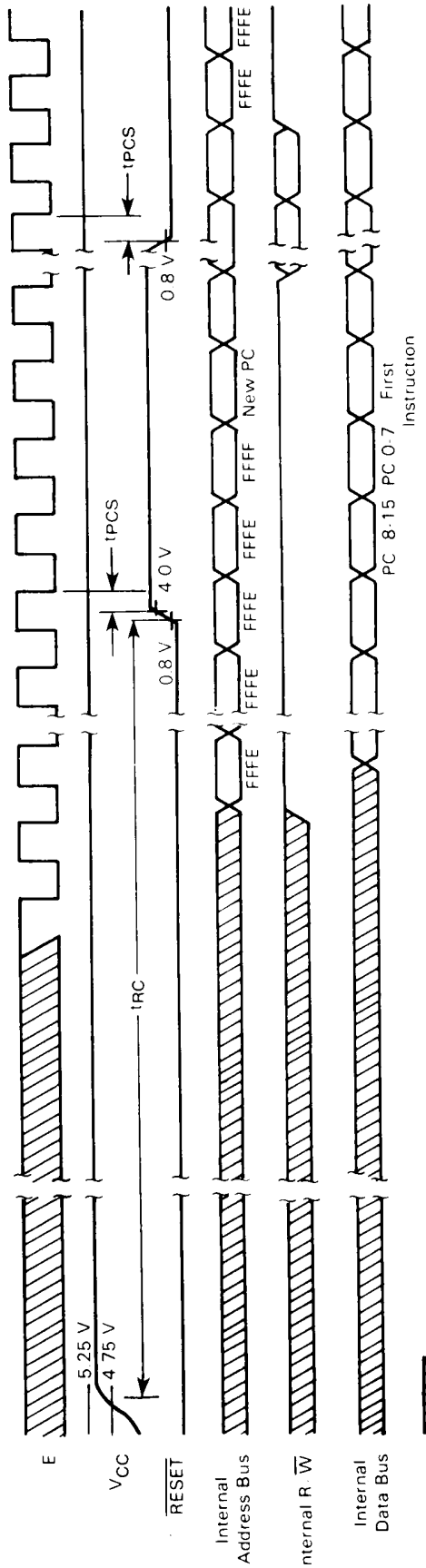


FIGURE 14 — RESET TIMING



$\overline{HALT}$

This level sensitive active low input causes the MPU to halt all activity when a low is applied to it. When the $\overline{HALT}$ input is low, the machine stops at the end of an instruction and bus available (BA) goes to a high state. During this time read/write (R/$\overline{W}$) is high and the address bus displays the address of the next instruction. See Figure 15 for timing requirements.

To debug programs, it is advantageous to step through programs one instruction at a time. To do this, $\overline{HALT}$ must be brought high for one clock cycle and then returned low as shown in Figure 15. The instruction illustrated is a one byte, two cycle instruction, such as CLRA. When the $\overline{HALT}$ line goes low, the MC6803E is halted after completing execution of the current instruction.

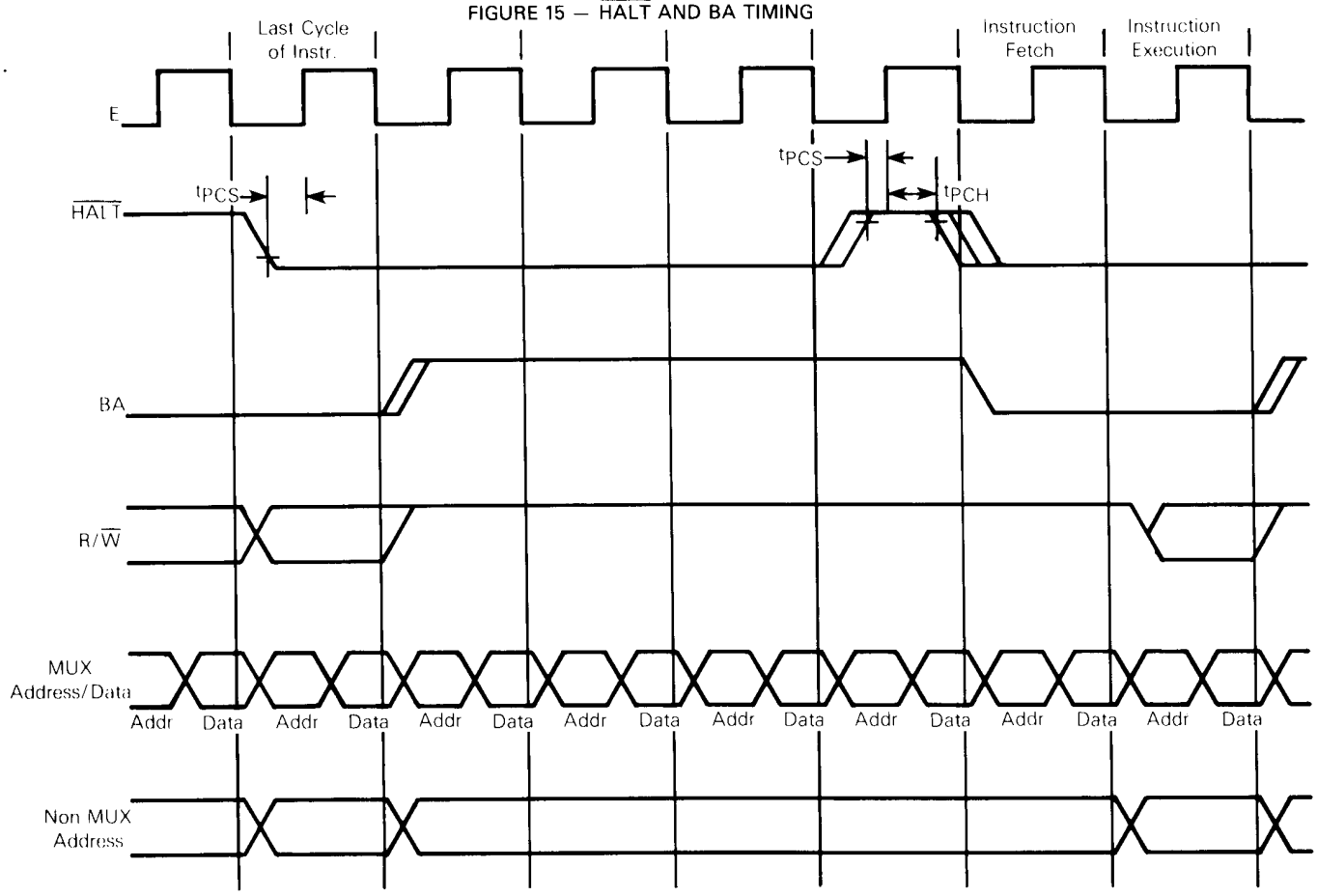
BA (BUS AVAILABLE)

This active high output is used to indicate when the MC6803E is halted. Other devices may then use the address and data buses, providing care is taken to prevent contention on the address and data bus. Alternatives include three-state buffers on the address and data buses, or three-state buffers on the address bus and holding AS low during BA high.

R/$\overline{W}$ (READ/WRITE)

The R/$\overline{W}$ output is used to indicate the direction of data transfer on the data bus. A logic low indicates that the MPU is writing data onto the bus and a logic high indicates that the MPU is reading data from the bus.

FIGURE 15 — $\overline{HALT}$ AND BA TIMING



$\overline{RESET}$

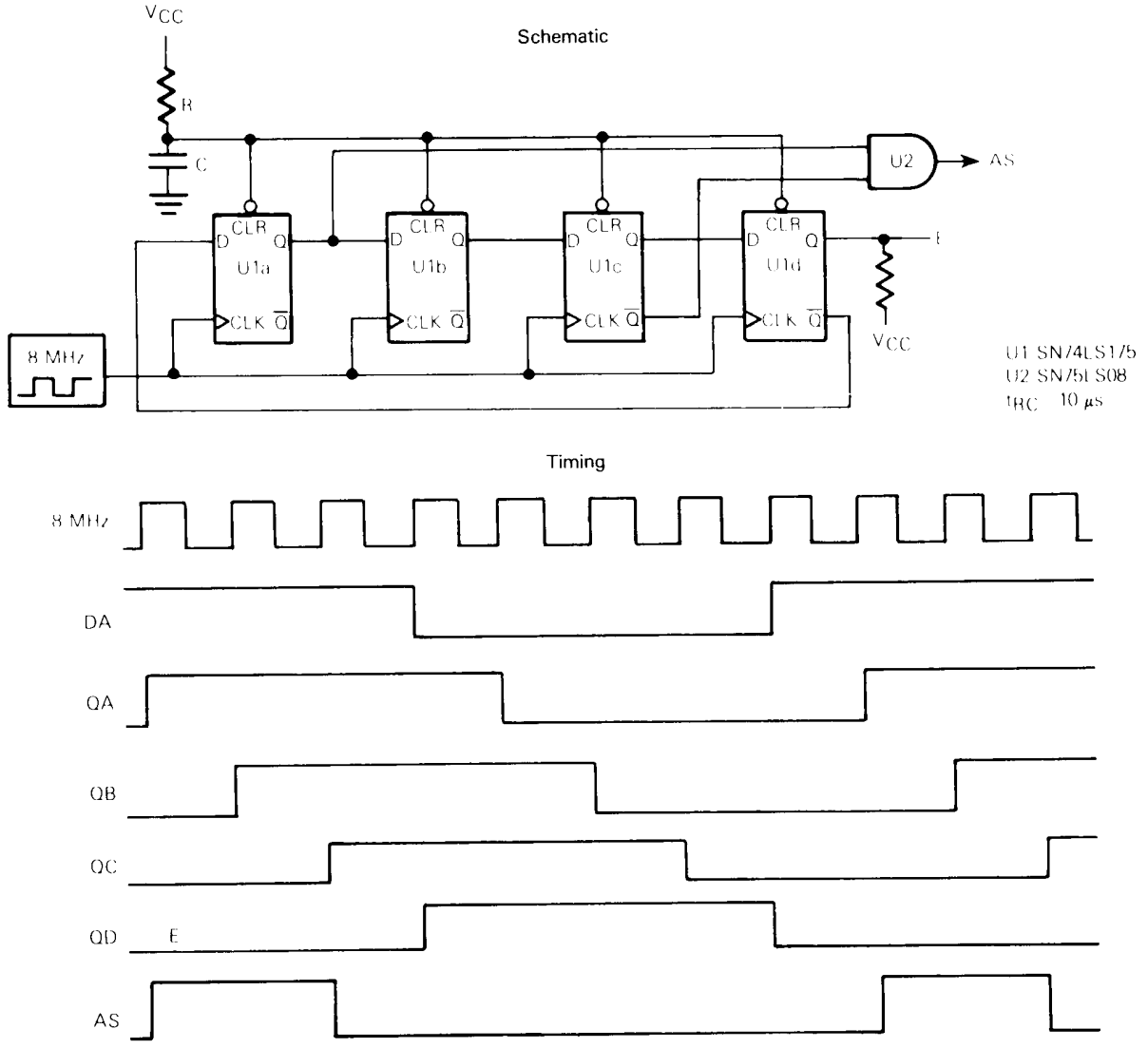
This input is used to reset the internal state of the device and provide an orderly start-up procedure. During power up, $\overline{RESET}$ must be held below 0.8 volts until 1) V_{CC} reaches 4.75 volts and E is stable, and 2) until V_{CC} standby reaches 4.75 volts. $\overline{RESET}$ must be held low at least three E cycles if asserted during power-up operation. During the rising edge of $\overline{RESET}$, the MC6803E also latches in its operating mode. $\overline{RESET}$ timing is shown in Figure 14.

E (ENABLE)

This is an input clock used primarily for address and data bus synchronization. This input should have some provision to obtain the specified logical high level which is greater than standard TTL levels. Two examples of clock generating circuits are presented in Figures 16 and 17.

Enable is the primary MC6803E system timing signal and all timing data specified as cycles is assumed to be referenced to this clock unless otherwise noted.

FIGURE 16 — CLOCK CIRCUIT EXAMPLE 1



$\overline{\text{NMI}}$ (NON-MASKABLE INTERRUPT)

An $\overline{\text{NMI}}$ negative edge requests an MPU interrupt sequence, but the current instruction will be completed before it responds to the request. The MPU 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. $\overline{\text{NMI}}$ typically requires a 3.3 kilohm (nominal) resistor to V_{CC}. There is no internal $\overline{\text{NMI}}$ pullup resistor. $\overline{\text{NMI}}$ must be held low for at least one E cycle to be recognized under all conditions.

$\overline{\text{IRQ1}}$ (MASKABLE INTERRUPT REQUEST 1)

$\overline{\text{IRQ1}}$ 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 MPU will begin an interrupt sequence. A vector is fetched from $FFF8 and $FFF9, transferred to the program counter, and instruction execution is resumed.

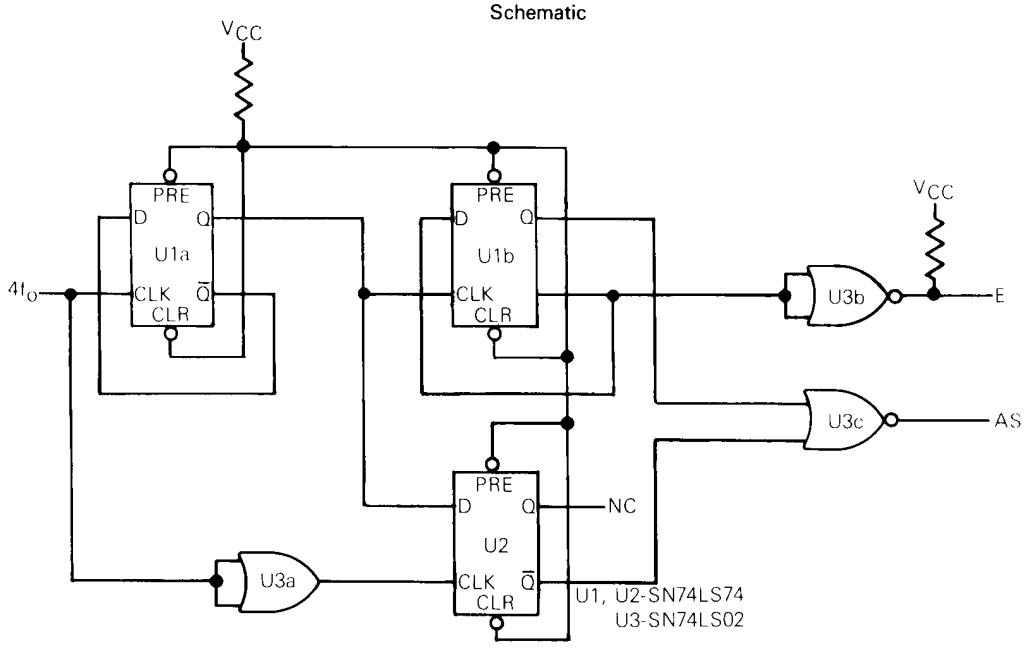
$\overline{\text{IRQ1}}$ typically requires an external 3.3 kilohm (nominal) resistor to V_{CC} for wire-OR applications. $\overline{\text{IRQ1}}$ has no internal pullup resistors.

P10-P17 (PORT 1)

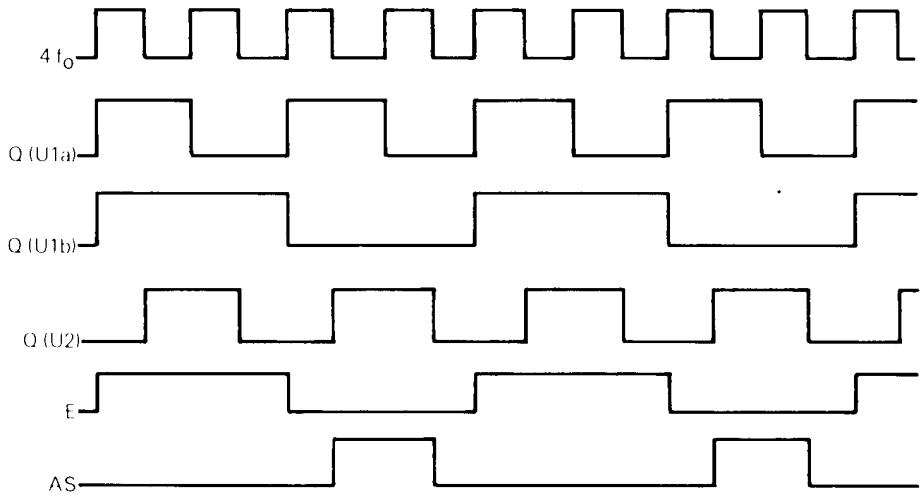
Port 1 is a mode independent 8-bit I/O port with each line an input or output as defined by the port 1 data direction register. The TTL compatible three state output buffers can drive one Schottky TTL load and 30 picofarads, Darlington transistors, or CMOS devices using external pullup resistors. It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

FIGURE 17 — CLOCK CIRCUIT EXAMPLE 2

Schematic



Timing



P20-P24 (PORT 2)

Port 2 is a mode-independent, 5-bit, multipurpose I/O port. The voltage levels present on P20 and P21 on the rising edge of $\overline{\text{RESET}}$ determine the operating mode of the MPU. 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.

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

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

PORT 2 DATA REGISTER

7	6	5	4	3	2	1	0	
0	PC1	PC0	P24	P23	P22	P21	P20	$03

P30-P37 (PORT 3)

Port 3 consists of a time multiplexed address (A7-A0) and data bus (D7 D0) where address strobe (AS) can be used to demultiplex the two buses. The port is held in a high-impedance state between valid address and data to prevent bus conflicts. The TTL-compatible three-state output buffers can drive one Schottky TTL load and 90 picofarads.

P40-P47 (PORT 4)

Port 4 functions as half of the address bus and provides A8 to A15. Port 4 can drive one Schottky TTL load and 90 picofarads and is the only port with internal pullup resistors. Unused lines can remain unconnected.

RESIDENT MEMORY

The MC6803E provides 128 bytes of on-board RAM. One half of the RAM is powered through the V_{CC} standby pin and is maintainable during V_{CC} power down. 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 power-down operation. It is intended that RAME be cleared and STBY PWR be set as part of a power-down 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 **RAM Enable (RAME)** – This read/write bit can be used to remove the entire RAM from the internal memory map. RAME is set (enabled) during reset provided standby power is available on the positive edge of $\overline{\text{RESET}}$. If RAME is clear, any access to a RAM address is external. If RAME is set and not in mode 3, the RAM is included in the internal map.

Bit 7 **Standby Power (STBY PWR)** – This bit is a read/write status bit which, when cleared, indicates that V_{CC} standby has decreased sufficiently below V_{SBB} (minimum) to make data in the standby RAM suspect. It can be set only by software and is not affected during reset.

PROGRAMMABLE TIMER

The programmable timer can be used to perform 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 18.

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.

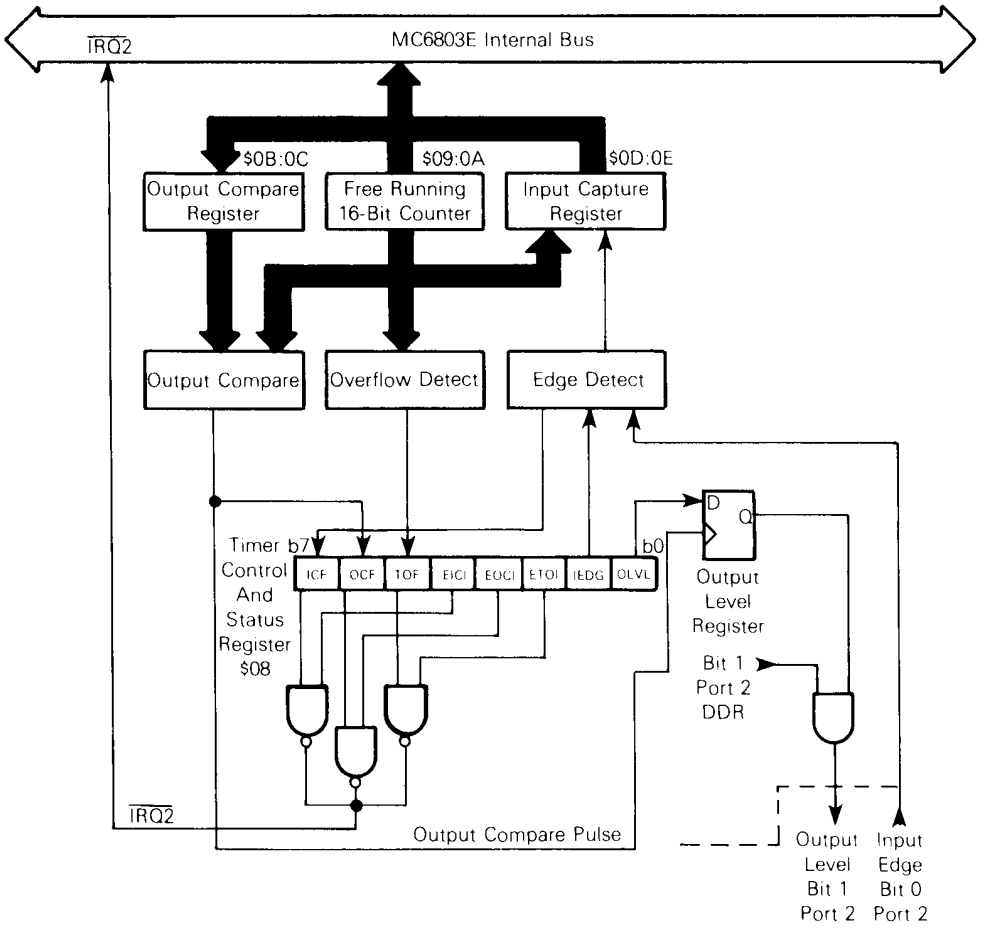
OUTPUT COMPARE REGISTER ($0B:0C)

The output compare register is a 16-bit read/write register used to control an output waveform or to 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 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 $\overline{\text{RESET}}$.

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

FIGURE 18 — BLOCK DIAGRAM OF PROGRAMMABLE TIMER



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 REGISTERS ($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:

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

Each of the three events can generate an $\overline{IRQ2}$ 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	$08

Bit 0 **Output Level (OLVL)** — 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. OLVL is cleared during reset.

Bit 1 **Input Edge (IEDG)** — IEDG is cleared during reset and controls which level transition on P20 will trigger a counter transfer to input capture register:
IEDG = 0 transfer on a negative edge
IEDG = 1 transfer on a positive edge

Bit 2 **Enable Timer Overflow Interrupt (ETOI)** When set, an $\overline{IRQ2}$ interrupt will be generated when the timer overflow flag is set; when clear, the interrupt is inhibited. ETOI is cleared during reset.

Bit 3 **Enable Output Compare Interrupt (EOCI)** — When set, an $\overline{IRQ2}$ interrupt will be generated when output compare flag is set; when clear, the interrupt is inhibited. EOCI is cleared during reset.

Bit 4 **Enable Input Capture Interrupt (EICI)** When set, an $\overline{IRQ2}$ interrupt will be generated when input capture flag is set; when clear, the interrupt is inhibited. EICI is cleared during reset.

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

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

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

SERIAL COMMUNICATIONS INTERFACE

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 biphase 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 multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to permit uninterested MPUs to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive 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:

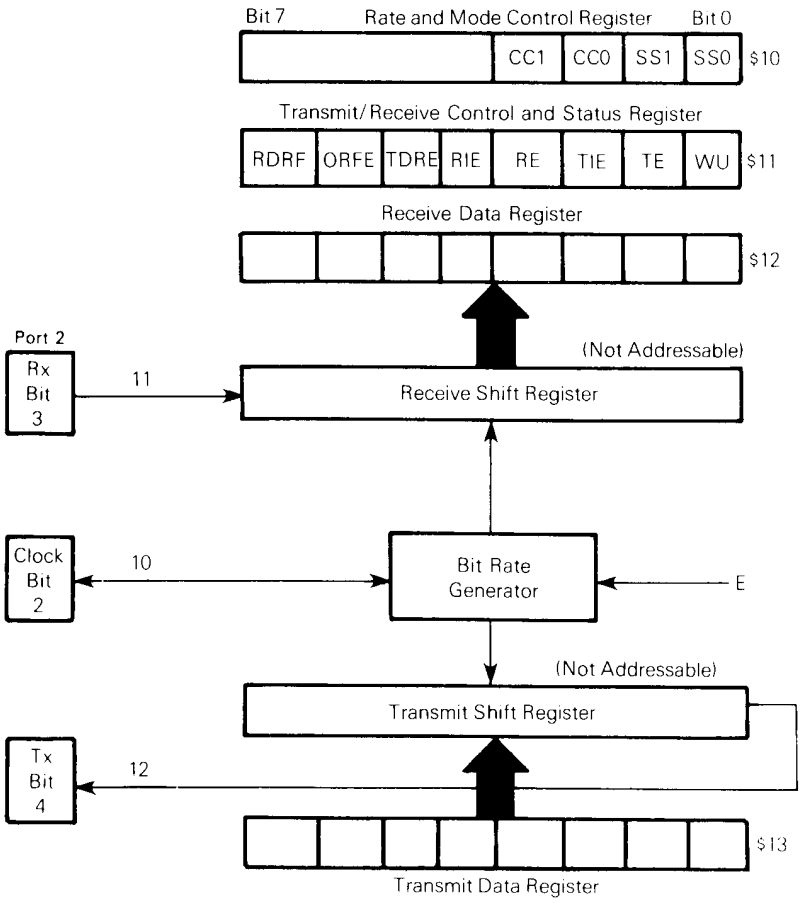
- format : standard mark/space (NRZ) or bi-phase
- clock: external or internal bit rate clock
- baud: one of 4 per E clock frequency, or external clock (8x desired baud)
- wake-up feature: enabled or disabled
- interrupt requests: enabled individually for transmitter and receiver
- clock output: internal bit rate clock enabled or disabled to P22

SERIAL COMMUNICATIONS REGISTERS

The serial communications interface includes four addressable registers as depicted in Figure 19. 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 REGISTER (RMCR) ($10) — The rate and mode control register controls the SCI bit rate, format, clock source, and under certain conditions the configuration of P22. The register consists of 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.

FIGURE 19 — SCI REGISTERS



RATE AND MODE CONTROL REGISTER

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

Bit 1:Bit 0 **SS1:SS0 Speed Select** – These two bits select the baud when using the internal clock. Four rates may be selected which are a function of the MPU input frequency. Table 6 lists bit time and rates for three selected MPU 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 (8×) 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.

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 $\overline{\text{RESET}}$.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER

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

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

Bit 1 **Transmit Enable (TE)** – When set, the P24 DDR bit is set and cannot be changed. P24 DDR 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 **Transmit Interrupt Enable (TIE)** – When set, an $\overline{\text{IRQ2}}$ is enabled when TDRE is set; when clear, the interrupt is inhibited. TIE is cleared during reset.

Bit 3 **Receive Enable (RE)** – When set, the P23 DDR bit is cleared and cannot be changed. P23 DDR will remain clear if RE is subsequently cleared. While RE is set, the SCI receiver is enabled. RE is cleared during reset.

Bit 4 **Receiver Interrupt Enable (RIE)** – When set, an $\overline{\text{IRQ2}}$ interrupt is enabled when RDRF and/or ORFE is set; when clear, the interrupt is inhibited. RIE is cleared during reset.

Bit 5 **Transmit Data Register Empty (TDRE)** – 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.

TABLE 6 – SCI BIT TIMES AND RATES

SS1:SS0		E	614.4 kHz		1.0 MHz		1.2288 MHz	
			Baud	Time	Baud	Time	Baud	Time
0	0	÷16	38400.0	26 μs	62500.0	16.0 μs	76800.0	13.0 μs
0	1	÷128	4800.0	208.3 μs	7812.5	128.0 μs	9600.0	104.2 μs
1	0	÷1024	600.0	1.67 ms	976.6	1.024 ms	1200.0	833.3 μs
1	1	÷4096	150.0	6.67 ms	244.1	4.096 ms	300.0	3.33 ms
External (P22)*			76800.0	13.0 μs	125000.0	8.0 μs	153600.0	6.5 μs

*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

Bit 6 **Overrun Framing Error (ORFE)** – If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the receive 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 **Receive Data Register Full (RDRF)** – RDRF is set when the input serial shift register is transferred to the receive data register, or during reset.

SERIAL OPERATIONS

The SCI is initialized by writing control bytes first to the rate and mode control register and then to the transmit/receive control and status register. When TE is set, the output of the transmit serial shift register is connected to P24 and serial output is initiated by transmitting a 9-bit preamble of ones.

At this point, one of two situations exists: 1) if the transmit data register is empty (TDRE = 1), a continuous string of ones will be sent indicating an idle line, or 2) if a byte has been written to the transmit-data register (TDRE = 0), it will be transferred to the output serial shift register (synchronized with the bit rate clock), TDRE will be set, and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0), and a stop bit (1) will be transmitted. If TDRE is still set when the next byte transfer 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 20.

INSTRUCTION SET

As stated earlier, the MC6803E 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.

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 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 two codes reserved for test purposes.

PROGRAMMING MODEL

A programming model for the MC6803E is shown in Figure 6. The registers are defined in the following paragraphs.

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 be concatenated and referred to as the D (double) accumulator. Any operation which modifies the D accumulator automatically modifies the A and B accumulators.

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.

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.

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

FIGURE 20 – SCI DATA FORMATS

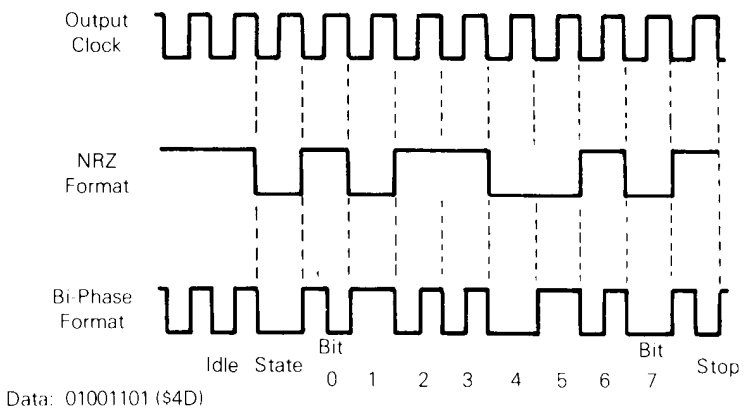


TABLE 8 — CPU INSTRUCTION MAP

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
01	NOP	INHER	2	1	35	TXS		3	1	69	ROL		6	2	9D	JSR		5	2
02	*				36	PSHA		3	1	6A	DEC		6	2	9E	LDS		4	2
03	*				37	PSHB		3	1	6B	*				9F	STS	DIR	4	2
04	LSRD		3	1	38	PULX		5	1	6C	INC		6	2	A0	SUBA	INDXD	4	2
05	ASLD		3	1	39	RTS		5	1	6D	TST		6	2	A1	CMPA		4	2
06	TAP		2	1	3A	ABX		3	1	6E	JMP		3	2	A2	SBCA		4	2
07	TPA		2	1	3B	RTI		10	1	6F	CLR	INDXD	6	2	A3	SUBD		6	2
08	INX		3	1	3C	PSHX		4	1	70	NEG	EXTND	6	3	A4	ANDA		4	2
09	DEX		3	1	3D	MUL		10	1	71	*				A5	BITA		4	2
0A	CLV		2	1	3E	WAI		9	1	72	*				A6	LDAA		4	2
0B	SEV		2	1	3F	SWI		12	1	73	COM		6	3	A7	STAA		4	2
0C	CLC		2	1	40	NEGA		2	1	74	LSR		6	3	A8	EORA		4	2
0D	SEC		2	1	41	*				75	*				A9	ADCA		4	2
0E	CLI		2	1	42	*				76	ROR		6	3	AA	ORAA		4	2
0F	SEI		2	1	43	COMA		2	1	77	ASR		6	3	AB	ADDA		4	2
10	SBA		2	1	44	LSRA		2	1	78	ASL		6	3	AC	CPX		6	2
11	CBA		2	1	45	*				79	ROL		6	3	AD	JSR		6	2
12	*				46	RORA		2	1	7A	DEC		6	3	AE	LDS		5	2
13	*				47	ASRA		2	1	7B	*				AF	STS	INDXD	5	2
14	*				48	ASLA		2	1	7C	INC		6	3	B0	SUBA	EXTND	4	3
15	*				49	ROLA		2	1	7D	TST		6	3	B1	CMPA		4	3
16	TAB		2	1	4A	DECA		2	1	7E	JMP		3	3	B2	SBCA		4	3
17	TBA		2	1	4B	*				7F	CLR	EXTND	6	3	B3	SUBD		6	3
18	*				4C	INCA		2	1	80	SUBA	IMMED	2	2	B4	ANDA		4	3
19	DAA	INHER	2	1	4D	TSTA		2	1	81	CMPA		2	2	B5	BITA		4	3
1A	*				4E	*				82	SBCA		2	2	B6	LDAA		4	3
1B	ABA	INHER	2	1	4F	CLRA		2	1	83	SUBD		4	3	B7	STAA		4	3
1C	*				50	NEGB		2	1	84	ANDA		2	2	B8	EORA		4	3
1D	*				51	*				85	BITA		2	2	B9	ADCA		4	3
1E	*				52	*				86	LDAA		2	2	BA	ORAA		4	3
1F	*				53	COMB		2	1	87	*				BB	ADDA		4	3
20	BRA	REL	3	2	54	LSRB		2	1	88	EORA		2	2	BC	CPX		6	3
21	BRN		3	2	55	*				89	ADCA		2	2	BD	JSR		6	3
22	BHI		3	2	56	RORB		2	1	8A	ORAA		2	2	BE	LDS		5	3
23	BLS		3	2	57	ASRB		2	1	8B	ADDA		2	2	BF	STS	EXTND	5	3
24	BCC		3	2	58	ASLB		2	1	8C	CPX	IMMED	4	3	C0	SUBB	IMMED	2	2
25	BCS		3	2	59	ROLB		2	1	8D	BSR	REL	6	2	C1	CMPB		2	2
26	BNE		3	2	5A	DECB		2	1	8E	LDS	IMMED	3	3	C2	SBCB		2	2
27	BEQ		3	2	5B	*				8F	*				C3	ADDD		4	3
28	BVC		3	2	5C	INCB		2	1	90	SUBA	DIR	3	2	C4	ANDB		2	2
29	BVS		3	2	5D	TSTB		2	1	91	CMPA		3	2	C5	BITB		2	2
2A	BPL		3	2	5E	*				92	SBCA		3	2	C6	LDAB		2	2
2B	BMI		3	2	5F	CLRB	INHER	2	1	93	SUBD		5	2	C7	*			
2C	BGE		3	2	60	NEG	INDXD	6	2	94	ANDA		3	2	C8	EORB		2	2
2D	BLT		3	2	61	*				95	BITA		3	2	C9	ADCB		2	2
2E	BGT		3	2	62	*				96	LDAA		3	2	CA	ORAB		2	2
2F	BLE	REL	3	2	63	COM		6	2	97	STAA		3	2	CB	ADDB		2	2
30	TSX	INHER	3	1	64	LSR		6	2	98	EORA		3	2	CC	LDD		3	3
31	INS		3	1	65	*				99	ADCA		3	2	CD	*			
32	PULA		4	1	66	ROR		6	2	9A	ORAA		3	2	CE	LDX	IMMED	3	3
33	PULB		4	1	67	ASR	INDXD	6	2	9B	ADDA		3	2	CF	*			

OP	MNEM	MODE	~	#
D0	SUBB	DIR	3	2
D1	CMPB		3	2
D2	SBCB		3	2
D3	ADDD		5	2
D4	ANDB		3	2
D5	BITB		3	2
D6	LDAB		3	2
D7	STAB		3	2
D8	EORB		3	2
D9	ADCB		3	2
DA	ORAB		3	2
DB	ADDB		3	2
DC	LDD		4	2
DD	STD		4	2
DE	LDX		4	2
DF	STX	DIR	4	2
E0	SUBB	INDXD	4	2
E1	CMPB		4	2
E2	SBCB		4	2
E3	ADDD		6	2
E4	ANDB		4	2
E5	BITB		4	2
E6	LDAB		4	2
E7	STAB		4	2
E8	EORB		4	2
E9	ADCB		4	2
EA	ORAB		4	2
EB	ADDB		4	2
EC	LDD		5	2
ED	STD		5	2
EE	LDX		5	2
EF	STX	INDXD	5	2
F0	SUBB	EXTND	4	3
F1	CMPB		4	3
F2	SBCB		4	3
F3	ADDD		6	3
F4	ANDB		4	3
F5	BITB		4	3
F6	LDAB		4	3
F7	STAB		4	3
F8	EORB		4	3
F9	ADCB		4	3
FA	ORAB		4	3
FB	ADDB		4	3
FC	LDD		5	3
FD	STD		5	3
FE	LDX		5	3
FF	STX	EXTND	5	3

*UNDEFINED OP CODE

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.

CONDITION CODE REGISTER — The condition code register indicates the results of an instruction and includes the following five condition bits: negative (N), zero (Z), overflow (V), carry/borrow from MSB (C), and half carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I bit) and inhibits all maskable interrupts when set. The two unused bits, B6 and B7, are read as ones.

ADDRESSING MODES

Six addressing modes can be used to reference memory. A summary of the addressing modes for all instructions is presented 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 megahertz, one E cycle is equivalent to one microsecond. A description of selected instructions is shown in Figure 21.

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) is a register and no memory reference is required. These are single byte instructions.

RELATIVE ADDRESSING — Relative addressing is used only for branch instructions. If the branch condition is true, the program counter is overwritten with the sum of 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 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

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

TABLE 10 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 1 of 2)

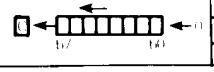
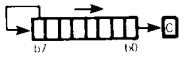
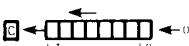
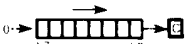
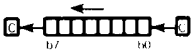
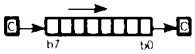
Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A · M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B · M → B	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	ASLA													48	2	1		•	•	↕	↕	↕	↕
	ASLB													58	2	1		•	•	↕	↕	↕	↕

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

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Shift Left Dbl	ASLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•	↕	↕	↕	↕
	ASRA													47	2	1		•	•	↕	↕	↕	↕
	ASRB													57	2	1		•	•	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	•	•	↕	↕	R	•
Compare Acmltrs	CBA													11	2	1	A - B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Decimal Adj, 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 Acmltrs	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		•	•	↕	↕	↕	↕
	LSLB													58	2	1		•	•	↕	↕	↕	↕
	LSLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					•	•	R	↕	↕	↕
	LSRA													44	2	1		•	•	R	↕	↕	↕
	LSRB													54	2	1		•	•	R	↕	↕	↕
	LSRD													04	3	1		•	•	R	↕	↕	↕
Multiply	MUL													3D	10	1	A X B → D	•	•	•	•	•	↕
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	↕	↕
	NEGA													40	2	1	00 - A → A	•	•	↕	↕	↕	↕
	NEGB													50	2	1	00 - B → B	•	•	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	•	•	•	•	•	•
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	3	1	A → Stack	•	•	•	•	•	•
	PSHB													37	3	1	B → Stack	•	•	•	•	•	•
Pull Data	PULA													32	4	1	Stack → A	•	•	•	•	•	•
	PULB													33	4	1	Stack → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	↕	↕
	ROLA													49	2	1		•	•	↕	↕	↕	↕
	ROLB													59	2	1		•	•	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	↕	↕
	RORA													46	2	1		•	•	↕	↕	↕	↕
	RORB													56	2	1		•	•	↕	↕	↕	↕
Subtract Acmltr	SBA													10	2	1	A - B → A	•	•	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↕	↕	↕	↕
Store Acmltrs	STAA				97	3	2	A7	4	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 Acmltr	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A - 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B - 00	•	•	↕	↕	R	R

The condition code register notes are listed after Table 12

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

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

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

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

LEGEND

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

CONDITION CODE SYMBOLS

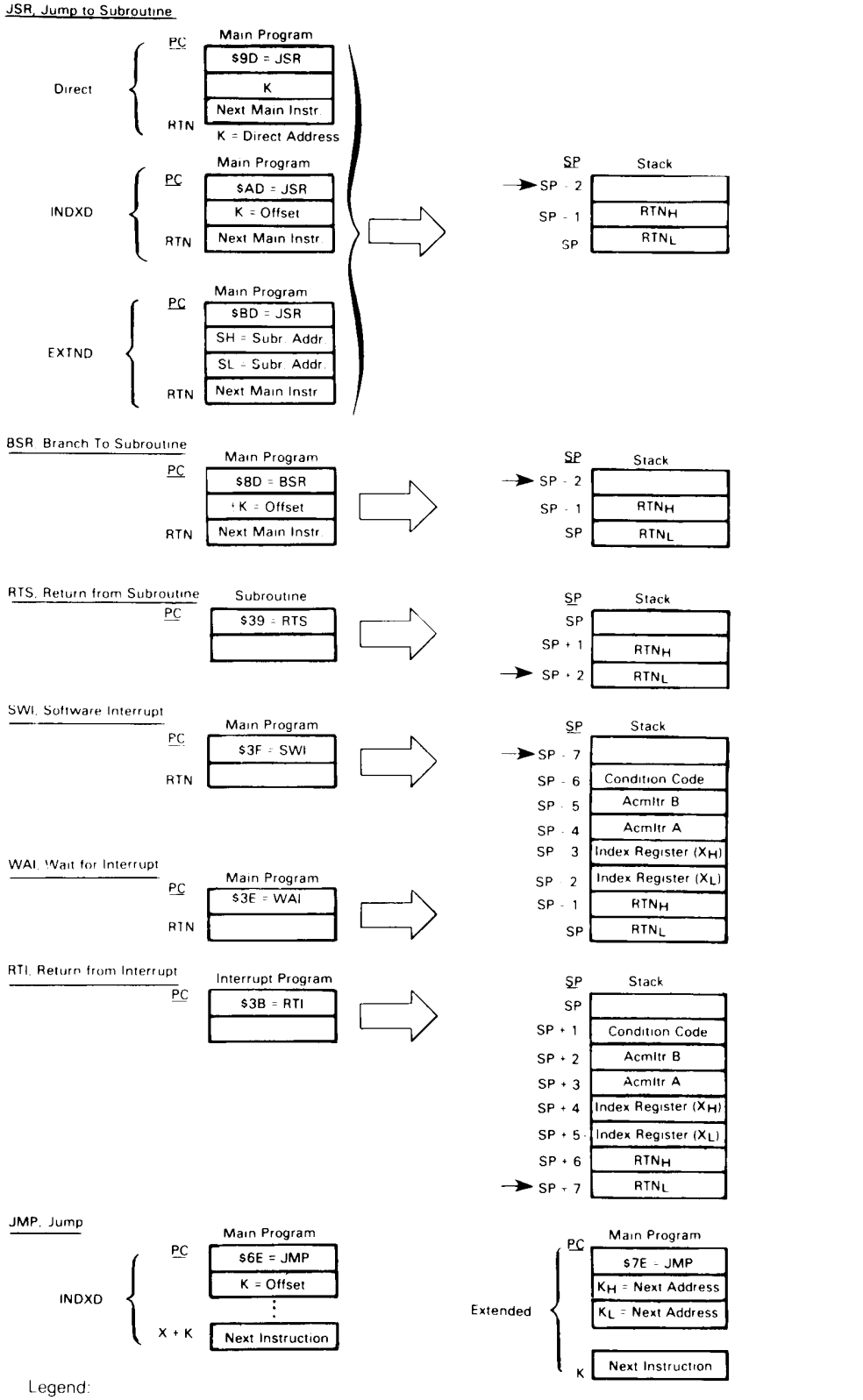
H Half-carry from bit 3
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from MSB
R Reset Always
S Set Always
↕ Affected
● Not Affected

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E CYCLES

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

	ADDRESSING MODE					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
INX	●	●	●	●	3	●
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	●

FIGURE 21 — SPECIAL OPERATIONS



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

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/$\overline{W}$) 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. High order byte refers to the most significant byte of a 16-bit value.

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 1 of 6)

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

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 2 of 6)

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

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 3 of 6)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code 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	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code 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	Op Code Address	1	Op Code
STX		2	Op Code 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	Op Code Address	1	Op Code
ASR NEG		2	Op Code 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 (1)		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code 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	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Op Code
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 4 of 6)

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

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 5 of 6)

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

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 6 of 6)

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