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

MC68701U4

Advance Information

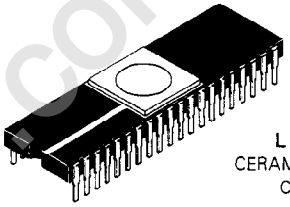
HMOS
(HIGH-DENSITY N-CHANNEL, SILICON-GATE)

8-BIT EPROM
MICROCOMPUTER/
MICROPROCESSOR

8-BIT EPROM MICROCOMPUTER/MICROPROCESSOR (MCU/MPU)

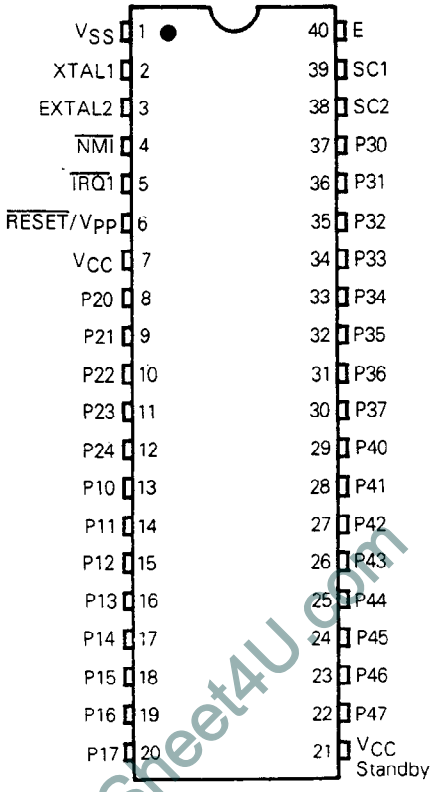
The MC68701U4 is an 8-bit single-chip EPROM microcomputer unit (MCU) which enhances the capabilities of the MC6801 and significantly enhances the capabilities of the M6800 Family of parts. It includes an MC6801 microprocessor unit (MPU) with direct object-code compatibility and upward object-code compatibility with the MC6800. Execution times of key instructions have been improved over the MC6800 and the new instructions found on the MC6801 are included. The MCU can function as a monolithic microcomputer or can be expanded to a 64K byte address space. It is TTL compatible and requires one +5-volt power supply for nonprogramming operation. An additional V_{PP} power supply is needed for EPROM programming. On-chip resources include 4096 bytes of EPROM, 192 bytes of RAM, a serial communications interface (SCI), parallel I/O, and a 16-bit six-function programmable timer.

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



L SUFFIX
CERAMIC PACKAGE
CASE 715

PIN ASSIGNMENT

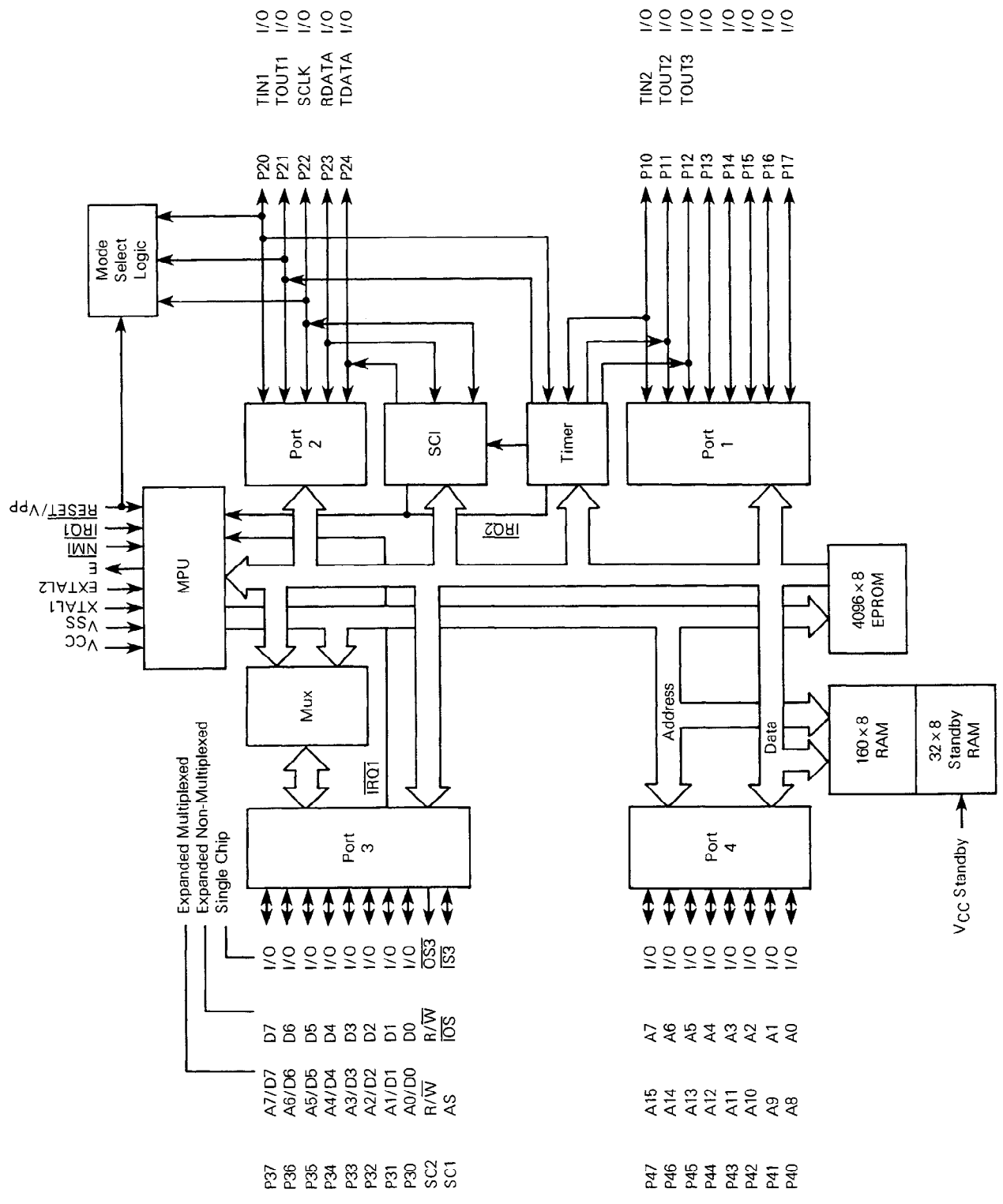


GENERIC INFORMATION
(T_A = 0° to 70°C)

Package Type	Frequency	Generic Number
Ceramic — L Suffix	1.0 MHz	MC68701U4L
	1.25 MHz	MC68701U4L-1

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

BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range Programmed Unprogrammed	T_{stg}	 −55 to +100 −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 applications 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 $GND \leq (V_{in}$ or $V_{out}) \leq V_{CC}$.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either GND or V_{CC}).

THERMAL CHARACTERISTICS

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

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.

CONTROL TIMING (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = 0 to 70°C)

Characteristic	Symbol	MC68701U4		MC68701U4-1		Unit
		Min	Max	Min	Max	
Frequency of Operation	f_o	0.5	1.0	0.5	1.25	MHz
Crystal Frequency	f_{XTAL}	2.0	4.0	2.0	5.0	MHz
External Oscillator Frequency	$4\ f_o$	2.0	4.0	2.0	5.0	MHz
Crystal Oscillator Startup Time	t_{rc}	—	100	—	100	ms
Processor Control Setup Time	t_{PCS}	200	—	170	—	ns

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

Characteristic			Symbol	Min	Typ	Max	Unit
Input High Voltage		$\overline{RESET}$ Other Inputs*	V_{IH}	$V_{SS}+4.0$ $V_{SS}+2.0$	– –	V_{CC} V_{CC}	V
Input Low Voltage		$\overline{RESET}$ Other Inputs*	V_{IL}	$V_{SS}-0.3$ $V_{SS}-0.3$	– –	$V_{SS}+0.4$ $V_{SS}+0.8$	V
Input Current (V_{in} = 0 to 2.4 V)	See Note	Port 4 SC1	I_{in}	– –	– –	0.5 0.8	mA
Input Current (V_{in} = 0 to 5.25 V)		$\overline{NMI}$, $\overline{IRQ1}$	I_{in}	–	–	2.5	μA
Input Current (V_{in} = 0 to 0.4 V) (V_{in} = 4.0 V to V_{CC})	See Note	$\overline{RESET}$/V_{PP}	I_{in}	– –	−2.0 –	– 8.0	mA
Hi-Z (Off State) Input Current (V_{in} = 0.5 to 2.4 V)		P10-P17, P20-P24, P30-P37	I_{TSI}	–	–	10	μA
Output High Voltage (I_{load} = −65 μA, V_{CC} = min) (I_{load} = −100 μA, V_{CC} = min)		P40-P47, SC1, SC2 Other Outputs	V_{OH}	$V_{SS}+2.4$ $V_{SS}+2.4$	– –	– –	V
Output Low Voltage (I_{load} = 2.0 mA, V_{CC} = min)		All Outputs	V_{OL}	–	–	$V_{SS}+0.5$	V
Darlington Drive Current (V_O = 1.5 V)		P10-P17	I_{OH}	1.0	–	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, P40-P47, SC1 Other Inputs	C_{in}	– –	– –	12.5 10.0	pF
V_{CC} Standby		Power Down Power Up	V_{SBB} V_{SB}	4.0 4.75	– –	5.25 5.25	V
Standby Current		Power Down	I_{SBB}	–	–	3.0	mA
Programming Time (Per Byte) (T_A = 25°C)			t_{PP}	25	–	50	ms
Programming Voltage (T_A = 25°C)			V_{PP}	20.0	21.0	22.0	V
Programming Current ($V_{\overline{RESET}} = V_{PP}$) ($T_A$ = 25°C)			I_{PP}	–	30.0	50.0	mA

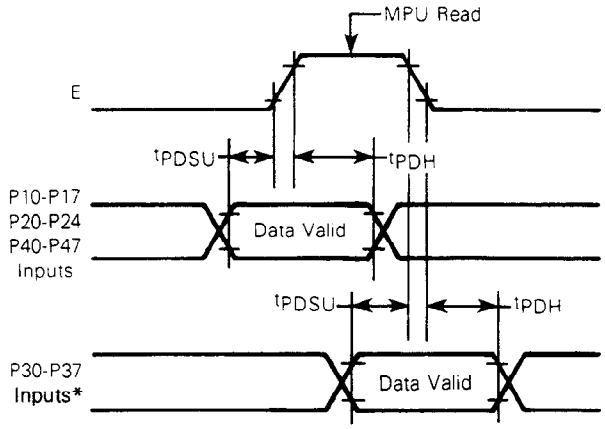
* Except Mode Programming Levels; See Figure 16.

NOTE: $\overline{RESET}$/V_{PP}, V_{IL}, and I_{in} values differ from MC6801U4 values.

PERIPHERAL PORT TIMING (Refer to Figures 1-4)

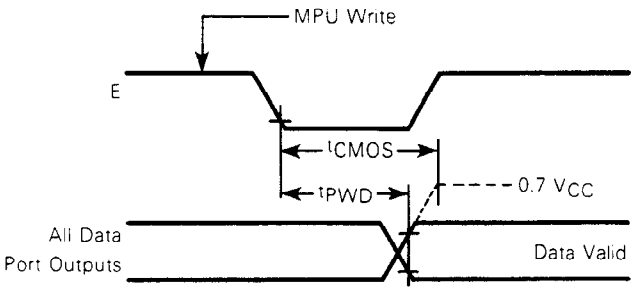
Characteristics	Symbol	Min	Typ	Max	Unit
Peripheral Data Setup Time	t_{PDSU}	200	–	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	–	ns
Delay Time, Enable Positive Transition to OS3 Negative Transition	t_{OSD1}	–	–	350	ns
Delay Time, Enable Positive Transition to OS3 Positive Transition	t_{OSD2}	–	–	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	t_{PWD}	–	–	350	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t_{CMOS}	–	–	2.0	μs
Input Strobe Pulse Width	t_{PWIS}	200	–	–	ns
Input Data Hold Time	t_{IH}	50	–	–	ns
Input Data Setup Time	t_{IS}	20	–	–	ns

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



*Port 3 non-latched operation (Latch enable = 0)

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
3. Port 4 cannot be pulled above V_{CC}

FIGURE 3 — PORT 3 OUTPUT STROBE TIMING
(SINGLE-CHIP MODE)

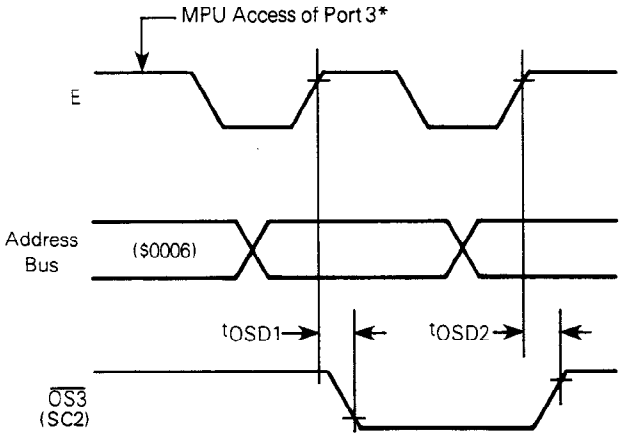
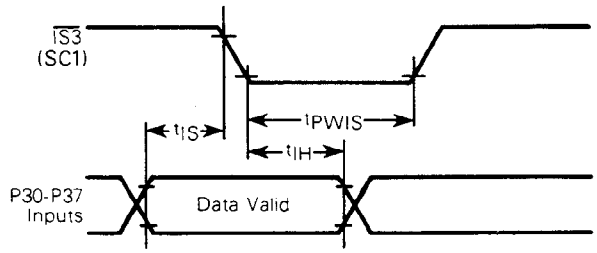


FIGURE 4 — PORT 3 LATCH TIMING
(SINGLE-CHIP MODE)



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

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

FIGURE 5 — CMOS LOAD

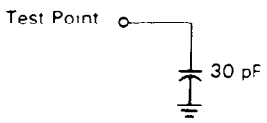
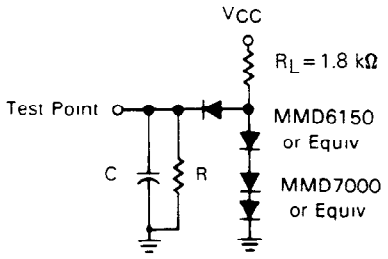


FIGURE 6 — TIMING TEST
LOAD PORTS 1, 2, 3, AND 4



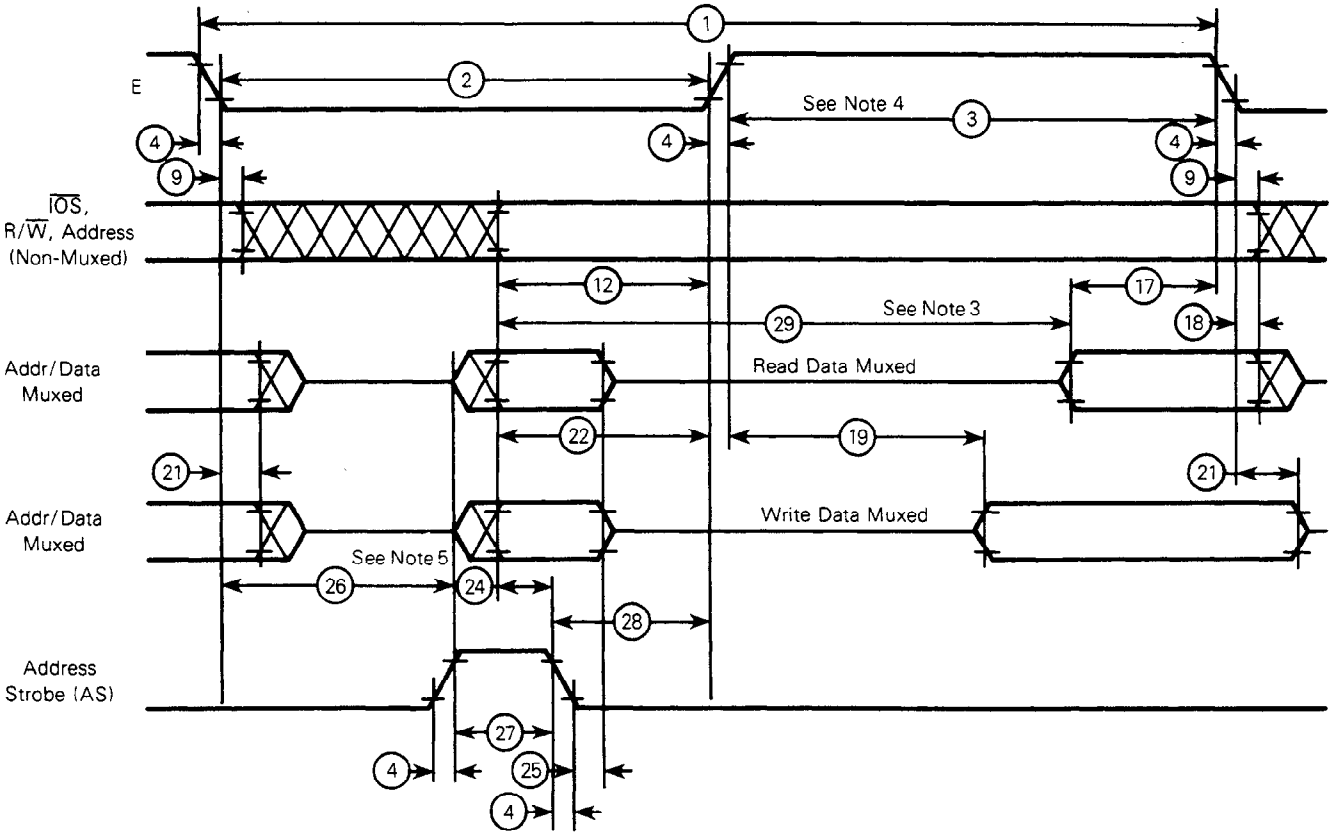
C = 90 pF for P30-P37, P40-P47, E, SC1, SC2
= 30 pF for P10-P17, P20-P24
R = 37 kΩ for P40-P47, SC1, SC2
= 24 kΩ for P10-P17, P20-P24, P30-P37, E

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

Ident. Number	Characteristics	Symbol	MC68701U4		MC68701U4-1		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	Address Hold Time	t_{AH}	20	–	20	–	ns
12	Non-Muxed Address Valid Time to E*	t_{AV}	200	–	150	–	ns
17	Read Data Setup Time	t_{DSR}	80	–	70	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	ns
19	Write Data Delay Time	t_{DDW}	–	225	–	200	ns
21	Write Data Hold Time	t_{DHW}	20	–	20	–	ns
22	Muxed Address Valid Time to E Rise*	t_{AVM}	160	–	120	–	ns
24	Muxed Address Valid Time to AS Fall*	t_{ASL}	40	–	30	–	ns
25	Muxed Address Hold Time	t_{AHL}	20	–	20	–	ns
26	Delay Time, E to AS Rise*	t_{ASD}	200	–	170	–	ns
27	Pulse Width, AS High*	PW_{ASH}	100	–	80	–	ns
28	Delay Time, AS to E Rise*	t_{ASED}	90	–	70	–	ns
29	Usable Access Time* (See Note 3)	t_{ACC}	530	–	410	–	ns

* At specified cycle time.

FIGURE 7 — 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 22 + 3 − 17.
4. Memory devices should be enabled only during E high to avoid port 3 bus contention.
5. Item 26 is different from the MC6801, but it is upward compatible.

INTRODUCTION

The MC68701U4 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 seven 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 and the MC6801. The programming model is depicted in Figure 8 where accumulator D is a concatenation of accumulators A and B. A list of new operations added to the MC6800 instruction set are shown in Table 1.

The basic difference between the MC6801U4 and the MC68701U4 is that the MC6801U4 has an on-chip ROM while the MC68701U4 has an on-chip EPROM. The

FIGURE 8 — PROGRAMMING MODEL

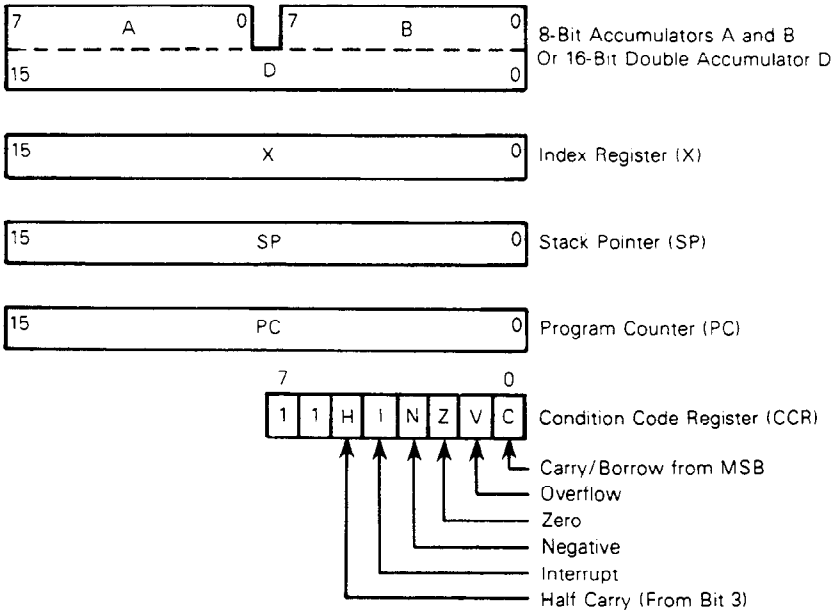


TABLE 1 — NEW INSTRUCTIONS

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

MC68701U4 is pin and code compatible with the MC6801U4 and can be used to emulate the MC6801U4, allowing easy software development using the on-chip EPROM. Software developed using the MC68701U4 can then be masked into the MC6801U4 ROM.

In order to support the on-chip EPROM, the MC68701U4 differs from the MC6801U4 as follows:

(1) Mode 0 in the MC6801U4 is a test mode only, while in the MC68701U4 mode 0 is also used to program the on-chip EPROM.
(2) The MC68701U4 RAM/EPROM control register has two bits used to control the EPROM in mode 0 that are not defined in the MC6801U4 RAM control register.
(3) The $\overline{\text{RESET}}$/V_{PP} pin in the MC68701U4 is dual purpose, used to supply EPROM power as well as to reset the device; while in the MC6801U4 the pin is called $\overline{\text{RESET}}$ and is used only to reset the device.

OPERATING MODES

The MC68701U4 provides seven different operating modes (modes 0 through 3 and 5 through 7). The operating modes are hardware selectable and determine the device memory map, the configuration of port 3, port 4, SC1, SC2, and the physical location of the interrupt vectors.

FUNDAMENTAL MODES

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

SINGLE-CHIP MODE (7) — In the single-chip mode, the four MCU ports are configured as parallel input/output data ports, as shown in Figure 9. The MCU functions as a monolithic microcomputer in this mode without external address or data buses. A maximum of 29 I/O lines and two port 3 control lines are provided. Peripherals or another MCU can be interfaced to port 3 in a loosely coupled dual-processor configuration, as shown in Figure 10.

TABLE 2 — SUMMARY OF OPERATING MODES

Single-Chip (Mode 7)
192 bytes of RAM, 4096 bytes of EPROM
Port 3 is a parallel I/O port with two control lines
Port 4 is a parallel I/O port
Expanded Non-Multiplexed (Mode 5)
192 bytes of RAM, 4096 bytes of EPROM
256 bytes of external memory space
Port 3 is an 8-bit data bus
Port 4 is an input port/address bus
Expanded Multiplexed (Modes 0, 1, 2, 3, 6)
Four memory space options (total 64K address space)
(1) Internal RAM and EPROM with partial address bus (mode 1)
(2) Internal RAM, no EPROM (mode 2)
(3) Extended addressing of internal I/O and RAM
(4) Internal RAM and EPROM with partial address bus (mode 6)
Port 3 is multiplexed address/data bus
Port 4 is address bus (inputs/address in mode 6)
Test/Program mode (mode 0):
May be used to test internal RAM and EPROM
May be used to test ports 3 and 4 as I/O ports by writing into mode 7
Used to program EPROM
Only modes 5, 6, and 7 can be irreversibly entered from mode 0
Resources Common to All Modes
Reserved register area
Port 1 input/output operation
Port 2 input/output operation
Timer operation
Serial communications interface operation

FIGURE 9 — SINGLE-CHIP MODE

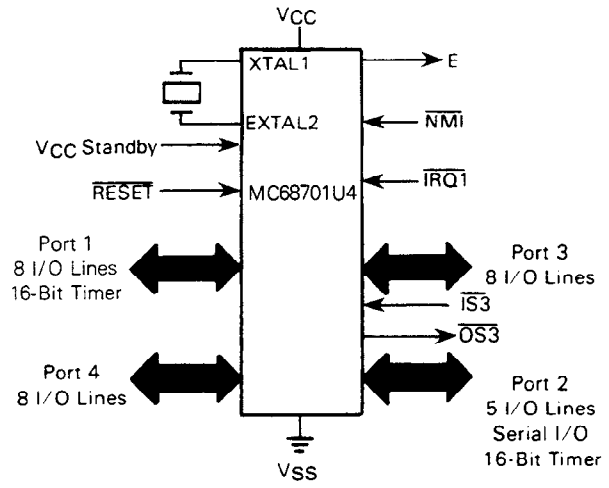
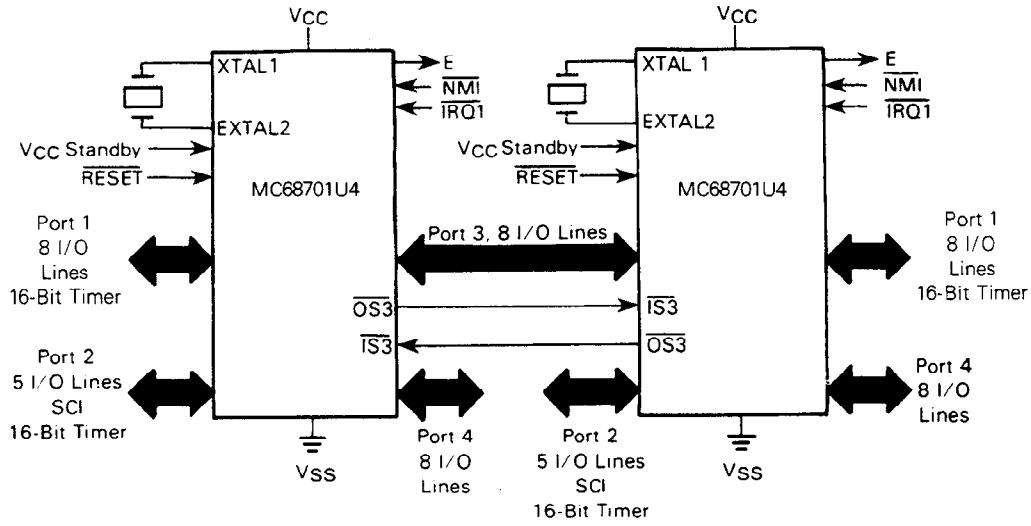


FIGURE 10 — SINGLE-CHIP DUAL PROCESSOR CONFIGURATION



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 11 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.

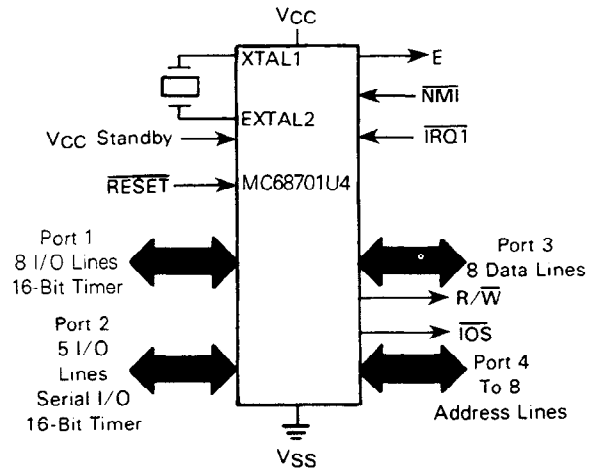
EXPANDED MULTIPLEXED MODES (0, 1, 2, 3, 6) — A 64K byte memory space is provided in the expanded multiplexed modes. In each of the expanded multiplexed modes, port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of address strobe (AS) and data valid while E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

In mode 0, the reset and interrupt vectors are located at $BFF0-$BFFF. 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. By writing the PC0-PC2 bits in the port 2 data register, modes 5, 6, and 7 can be irreversibly entered from mode 0. Mode 0 is used to program the on-chip EPROM.

Figure 12 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 13. This allows port 3 to function as a data bus when E is high.

FIGURE 11 — EXPANDED NON-MULTIPLEXED CONFIGURATION



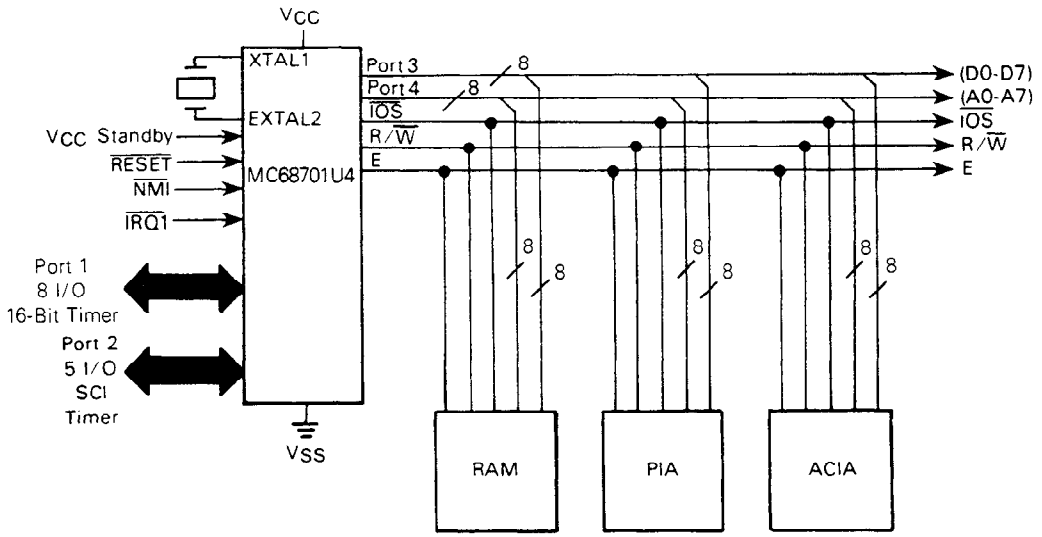
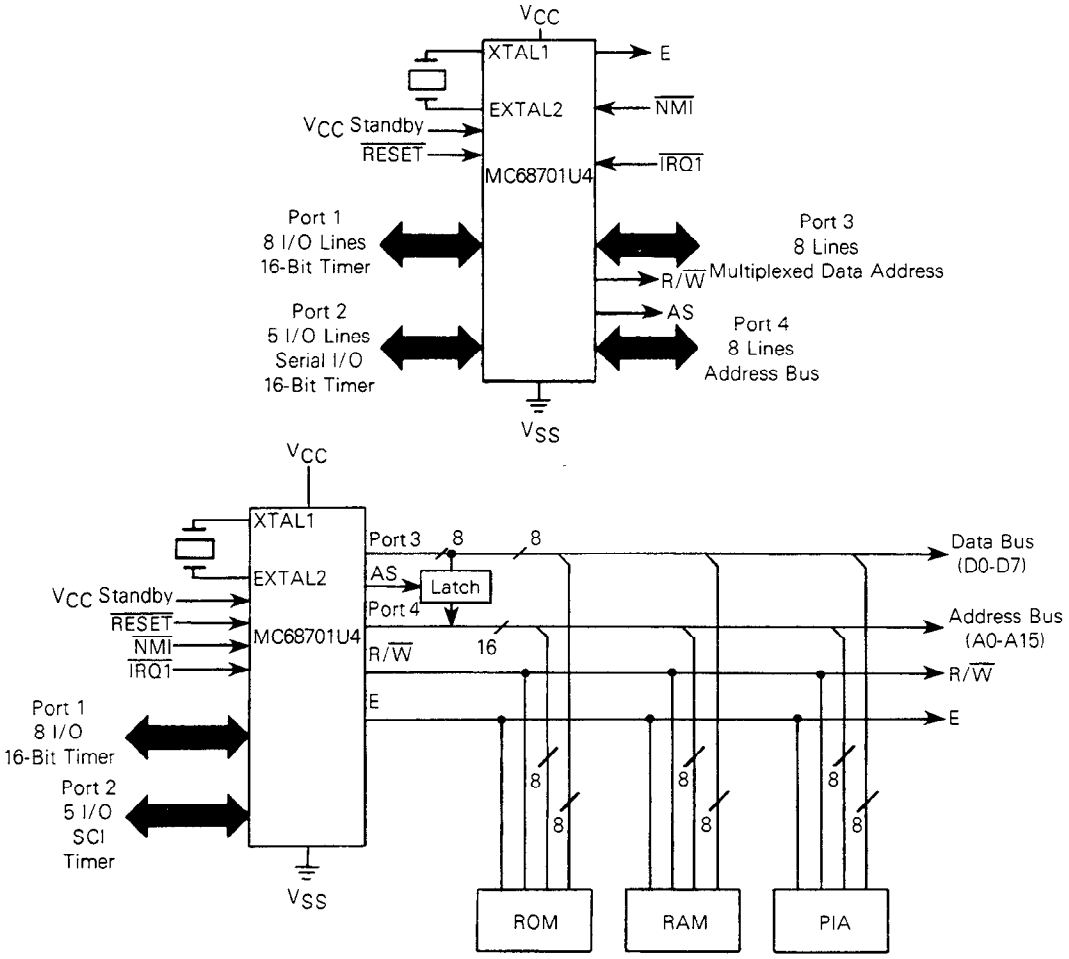
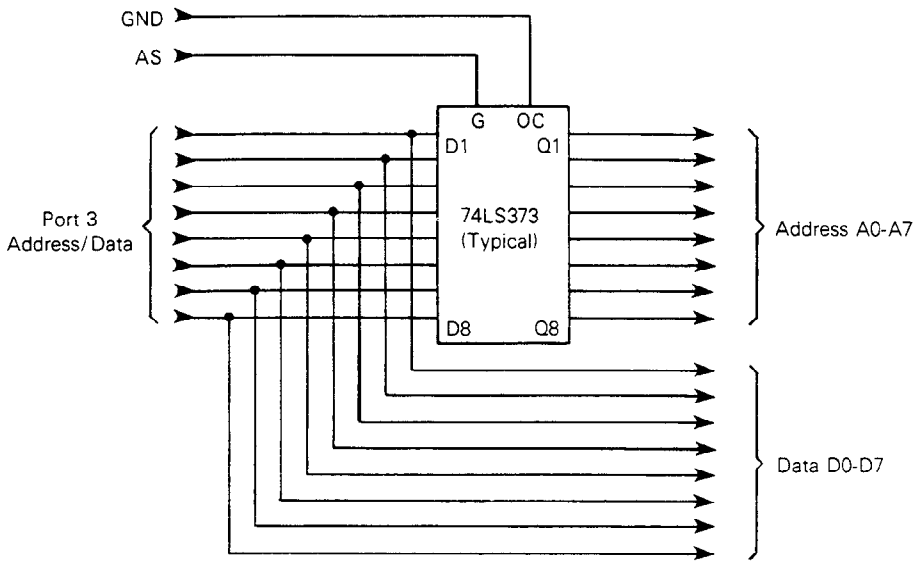


FIGURE 12 — EXPANDED MULTIPLEXED CONFIGURATION



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

FIGURE 13 — TYPICAL LATCH ARRANGEMENT



PROGRAMMING THE MODE

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

PORT 2 DATA REGISTER

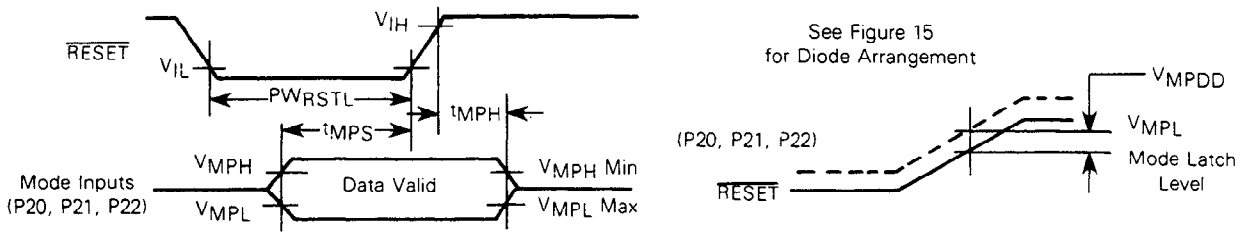
7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$03

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 15 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

MEMORY MAPS

The MC68701U4 can provide up to 64K byte address space depending on the operating mode. A memory map for each operating mode is shown in Figure 16. The first 32 locations of each map are reserved for the internal register area, as shown in Table 4, with exceptions as indicated.

FIGURE 14 — MODE PROGRAMMING TIMING



MODE PROGRAMMING (Refer to Figure 14)

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
$\overline{\text{RESET}}$ Low Pulse Width	PW_{RSTL}	3.0	–	E Cycles
Mode Programming Setup Time	t_{MPS}	2.0	–	E Cycles
Mode Programming Hold Time $\overline{\text{RESET}}$ Rise Time ≥ 1 μs $\overline{\text{RESET}}$ Rise Time < 1 μs	t_{MPH}	0 100	– –	ns

TABLE 3 — MODE SELECTION SUMMARY

Mode	P22 PC2	P21 PC1	P20 PC0	EPROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(2, 3)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(2, 3)	Non-Multiplexed/Partial Decode
4	H	L	L	–	–	–	–	Undefined(4)
3	L	H	H	E	I	E	MUX(1, 5)	Multiplexed/RAM
2	L	H	L	E	I	E	MUX(1)	Multiplexed/RAM
1	L	L	H	I	I	E	MUX(2, 3)	Multiplexed/RAM and EPROM
0	L	L	L	I	I	E	MUX(1)	Multiplex ed Test/Programming

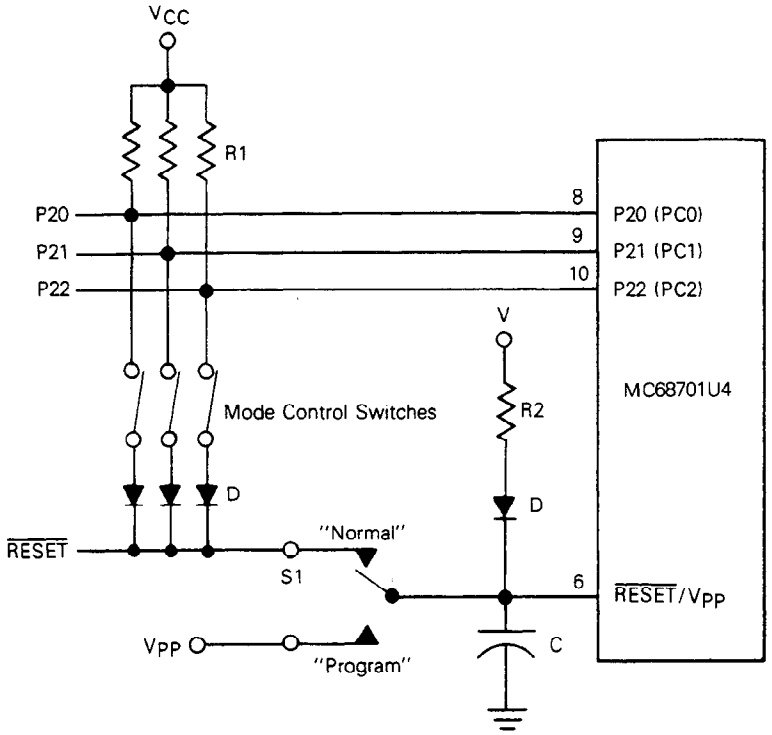
LEGEND

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

NOTES:

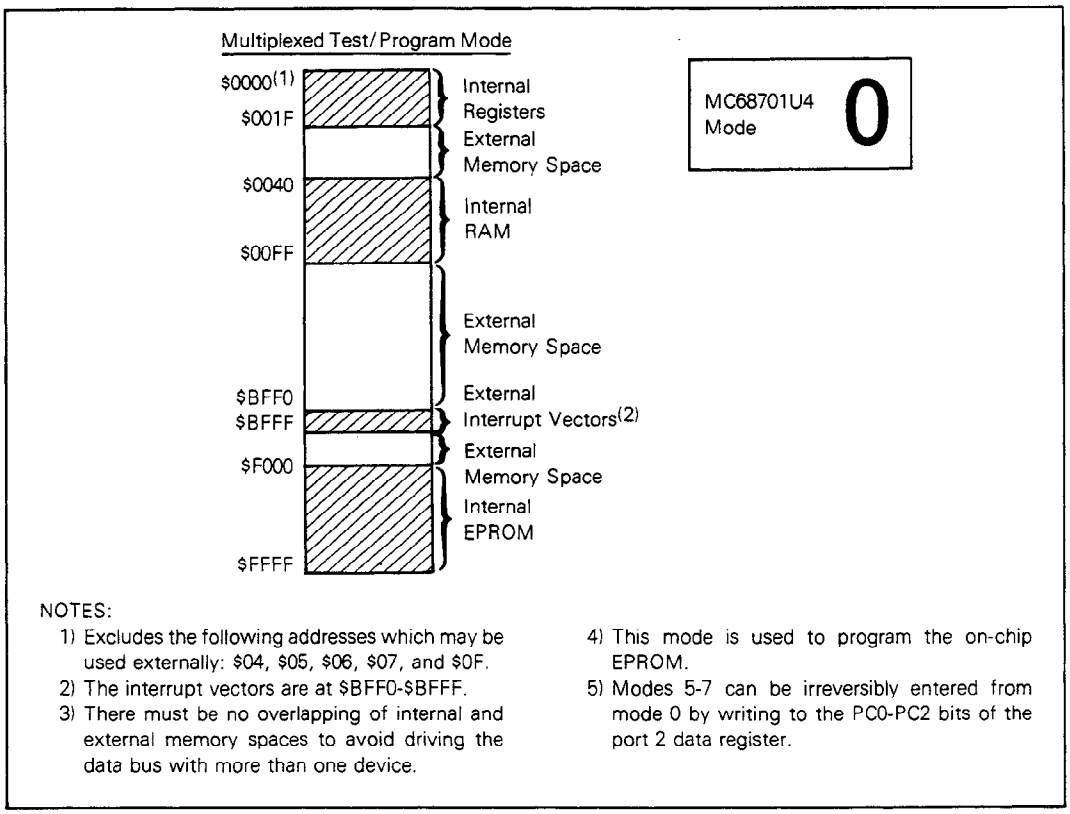
1. Addresses associated with ports 3 and 4 are considered external in modes 0, 2, and 3.
2. Addresses associated with port 3 are considered external in modes 1, 5, and 6.
3. Port 4 default is user data input; address output is optional by writing to port 4 data direction register
4. Mode 4 is a non-user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

FIGURE 15 — TYPICAL MODE PROGRAMMING CIRCUIT



NOTES:
1. Mode 0 as shown (switches closed).
2. R1 = 10 kilohms (typical).
3. The $\overline{\text{RESET}}$ time constant is equal to RC where R is the equivalent parallel resistance of R2 and the number of resistors (R1) placed in the circuit by closed mode control switches.
4. D = 1N914, 1N4001 (typical).
5. If V = V_{CC}, then R2 = 50 ohms (typical) to meet V_{IH} for the $\overline{\text{RESET}}$/V_{PP} pin. V = V_{CC} is also compatible with MC6801U4. The $\overline{\text{RESET}}$ time constant in this case is approximately R2 × C.
6. Switch S1 allows selection of normal ($\overline{\text{RESET}}$) or programming (V_{PP}) as the input to the $\overline{\text{RESET}}$/V_{PP} pin. During switching, the input level is held at a value determined by a diode (D), resistor (R2) and input voltage (V).
7. While S1 is in the "Program" position, $\overline{\text{RESET}}$ should not be asserted.
8. From powerup, $\overline{\text{RESET}}$ must be held low for at least t_{RC}. The capacitor, C, is shown for conceptual purposes only and is on the order of 1000 μF for the circuit shown. Typically, a buffer with an RC input will be used to drive $\overline{\text{RESET}}$, eliminating the need for the larger capacitor.
9. Diode V_f should not exceed V_{MPDD} min.

FIGURE 16 — MEMORY MAPS (Sheet 1 of 3)

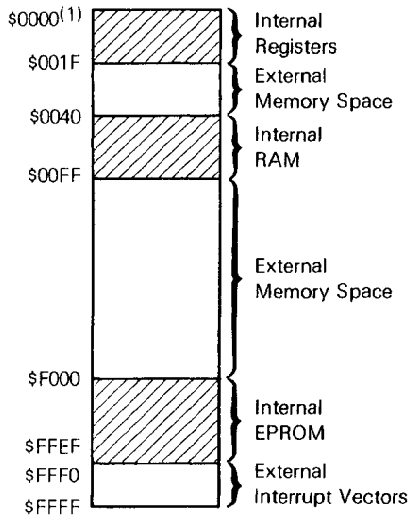


NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.
2) The interrupt vectors are at $BFF0-$BFFF.
3) There must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4) This mode is used to program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

FIGURE 16 — MEMORY MAPS (Sheet 2 of 3)

MC68701U4 Mode 1

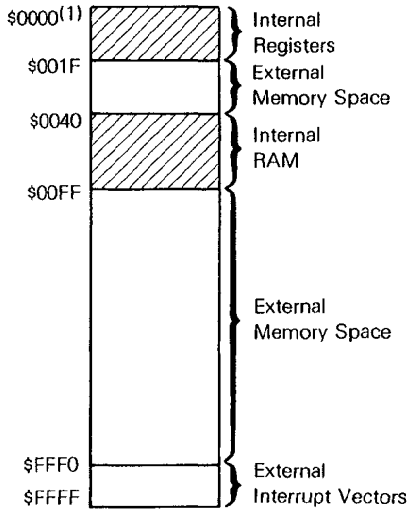
Multiplexed/RAM and EPROM



NOTES:
1) Excludes the following addresses which may be used externally: $05 and $07.
2) Internal EPROM addresses $FFF0 to $FFFF are not usable.
3) Address lines A8-A15 will not contain addresses until the data direction register for port 4 has been written with "1s" in the appropriate bits. These address lines will assert "1s" until made outputs by writing the data direction register.

MC68701U4 Mode 2

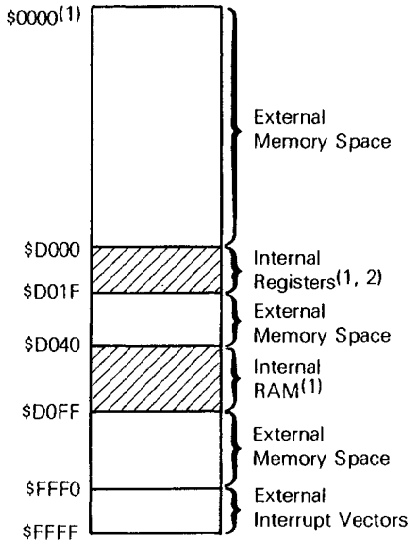
Multiplexed/RAM



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

MC68701U4 Mode 3

Multiplexed/RAM

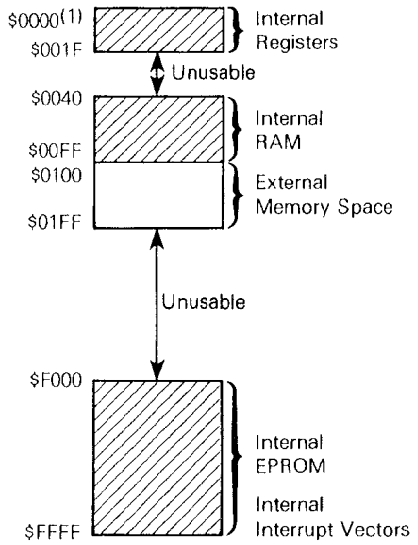


NOTES:
1) Relocating the internal registers and the internal RAM to high memory allows the processor to make use of direct addressing.
2) Excludes the following addresses which may be used externally: $D004, $D005, $D006, $D007, and $D00F.

FIGURE 16 — MEMORY MAPS (Sheet 3 of 3)

MC68701U4 Mode 5

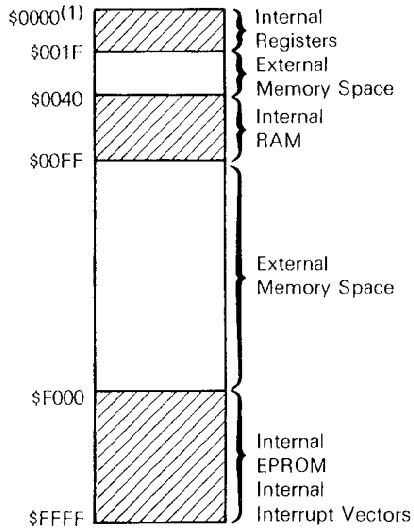
Non-Multiplexed/Partial Decode



NOTES:
1) Excludes the following addresses which may not be used externally: $04, $06, and $0F (no $\overline{IOS}$).
2) Address lines A0 to A7 will not contain addresses until the data direction register for port 4 has been written with "1s" in the appropriate bits. These address lines will assert "1s" until made outputs by writing the data direction register.

MC68701U4 Mode 6

Multiplexed/Partial Decode



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

MC68701U4 Mode 7

Single Chip

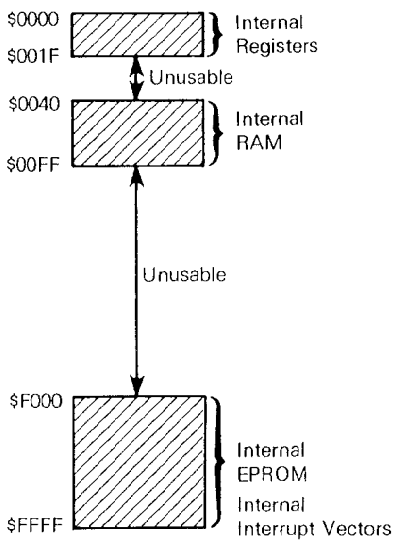


TABLE 4 — INTERNAL REGISTER AREA

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
Counter Alternate Address (High Byte)	15
Counter Alternate Address (Low Byte)	16
Timer Control Register 1	17
Timer Control Register 2	18
Timer Status Register	19
Output Compare Register 2 (High Byte)	1A
Output Compare Register 2 (Low Byte)	1B
Output Compare Register 3 (High Byte)	1C
Output Compare Register 3 (Low Byte)	1D
Input Capture Register 2 (High Byte)	1E
Input Capture Register 2 (Low Byte)	1F

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

MC68701U4 INTERRUPTS

The M6801 Family 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 and IS3 use $\overline{IRQ1}$. An $\overline{IRQ1}$ interrupt is serviced before $\overline{IRQ2}$ if both are pending.

NOTE

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

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. In mode 0, reset and interrupt vectors are defined as $BFF0-$BFFF.

The interrupt flowchart is depicted in Figure 17 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 $\overline{RESET}$ timing are illustrated in Figures 18 and 19.

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

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

* $\overline{IRQ2}$ interrupt
* * $\overline{NMI}$ must be armed (by accessing stack pointer) before an $\overline{NMI}$ is executed

FIGURE 17 — INTERRUPT FLOWCHART

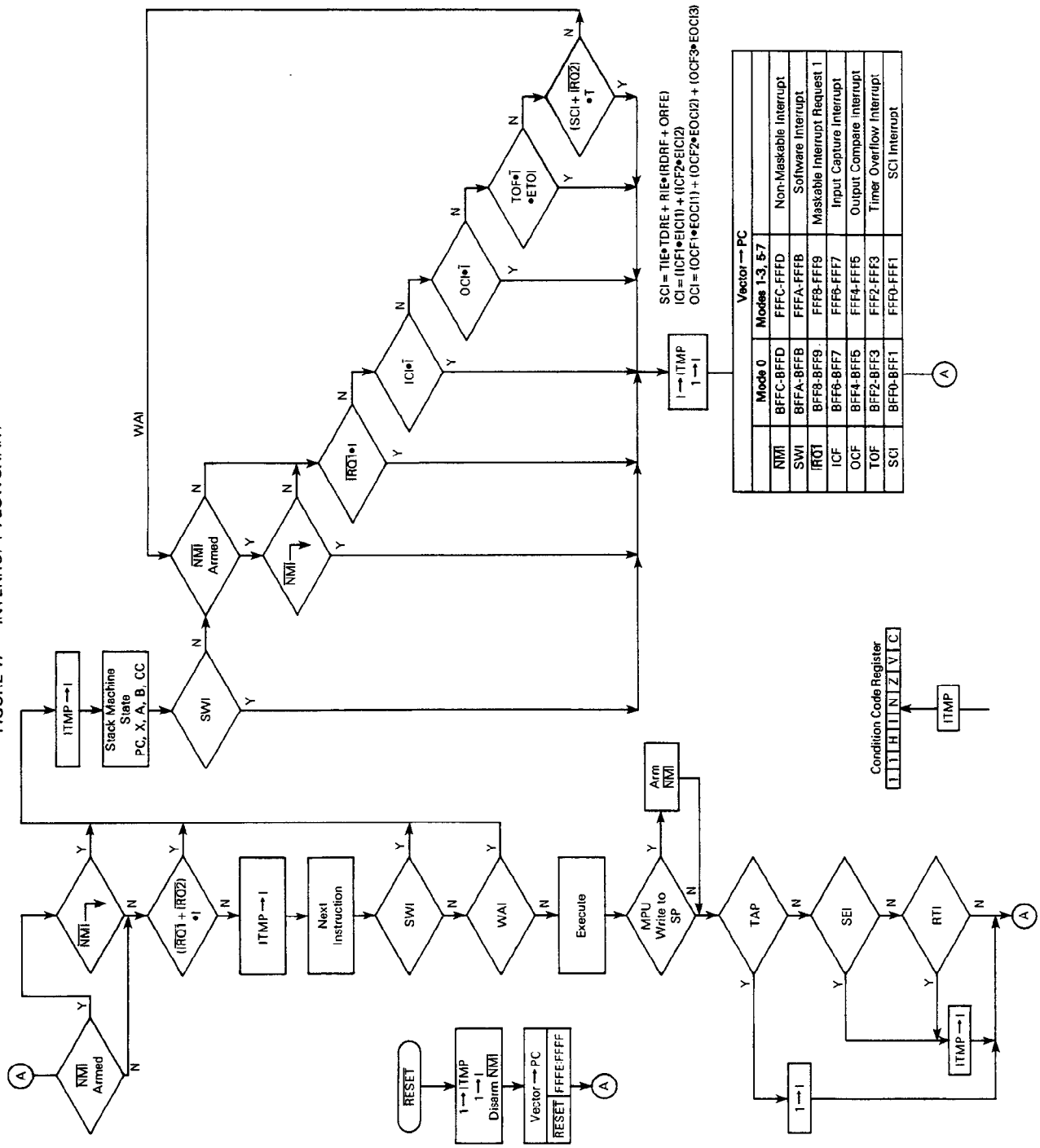


FIGURE 18 — INTERRUPT SEQUENCE

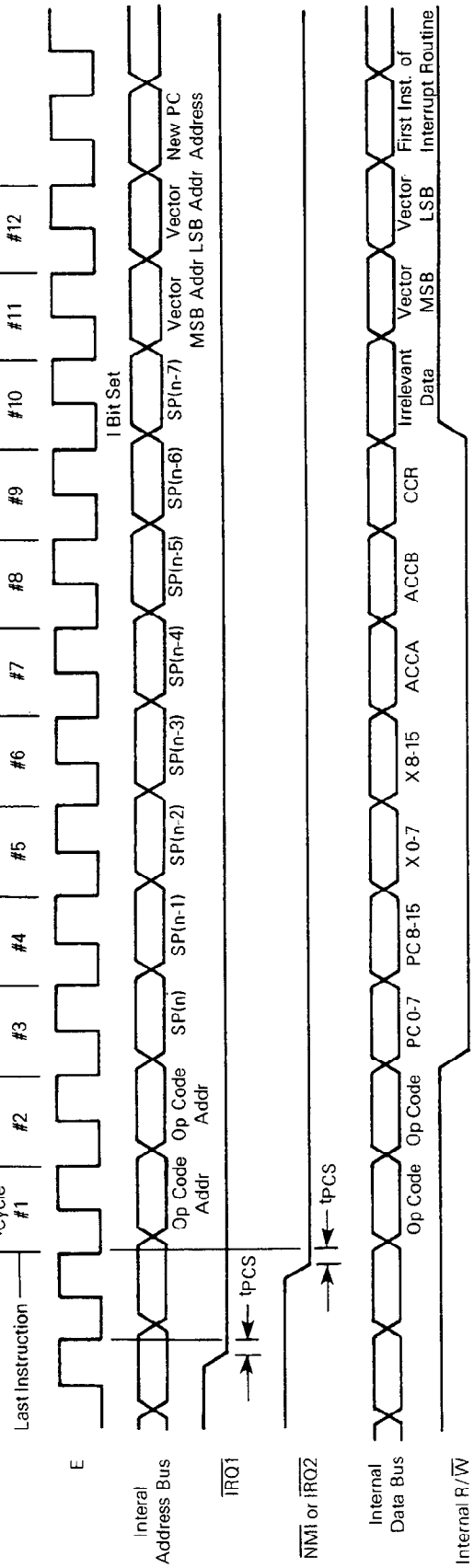
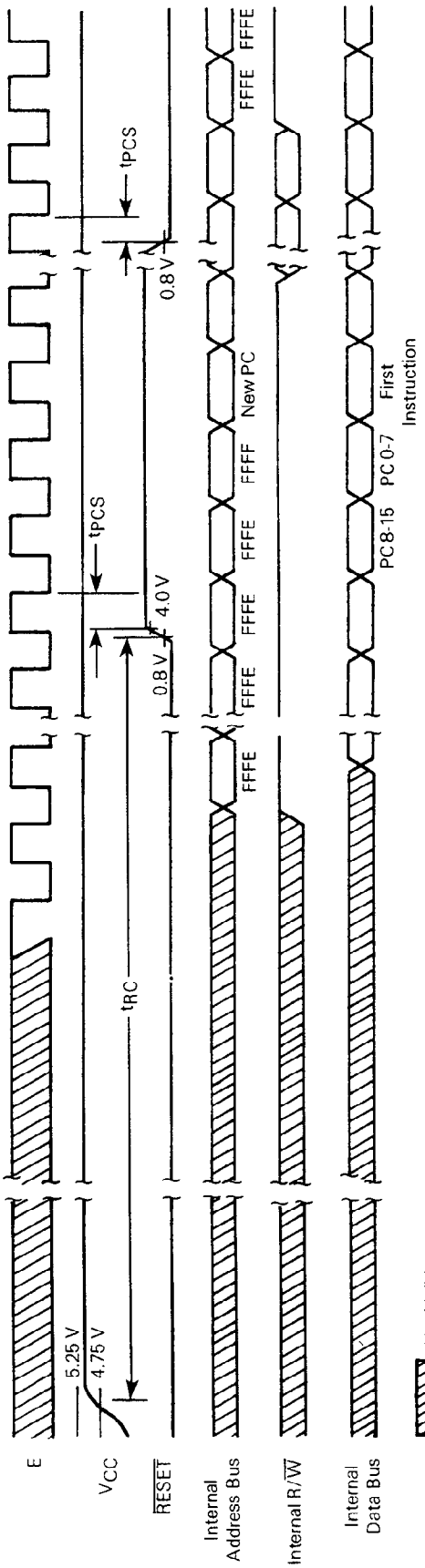


FIGURE 19 — RESET TIMING



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 (±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 ($40 through $5F) 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.

XTAL1 AND EXTAL 2

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, EXTAL2 may be driven by an external TTL-compatible clock at 4 f_o with a duty cycle of 50% (±5%) with XTAL1 connected ground.

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

$\overline{RESET}$/V_{PP}

This input is used to reset the internal state of the device and provide an orderly startup procedure. During power up, $\overline{RESET}$ 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. $\overline{RESET}$ must be held low at least three E cycles if asserted during power-up operation.

This pin is also used to supply V_{PP} in mode 0 for programming the EPROM.

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.

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

An $\overline{NMI}$ negative edge requests an MCU interrupt sequence, but the current instruction will be completed before it responds to the request. The MCU will then begin an interrupt sequence. Finally, a vector is fetched from $FFFC and $FFFD ($BFFC and $BFFD in mode 0), transferred to the program counter, and instruction execution is resumed. $\overline{NMI}$ typically requires a 3.3 kΩ (nominal) resistor to V_{CC}. There is no internal $\overline{NMI}$ pullup resistor. $\overline{NMI}$ must be held low for at least one E cycle to be recognized under all conditions.

NOTE

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

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

$\overline{IRQ1}$ 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 $FFF9 ($BFF8 and $BFF9 in mode 0), transferred to the program counter, and instruction execution is resumed.

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

SC1 and SC2 (STROBE CONTROL 1 AND 2)

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

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

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

SC1 AND SC2 IN EXPANDED 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 modes, both SC1 and SC2 are configured as outputs. SC1 functions as address strobe and can be used to demultiplex the eight least significant addresses and the data bus. A latch controlled by address strobe captures the lower address on the negative edge, as shown in Figure 13.

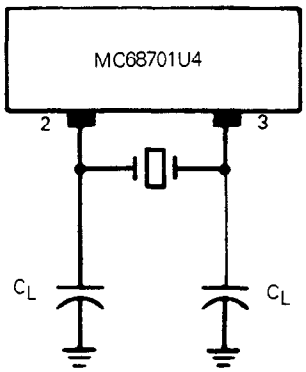
FIGURE 20 — OSCILLATOR CHARACTERISTICS

(a) Nominal Recommended Crystal Parameters

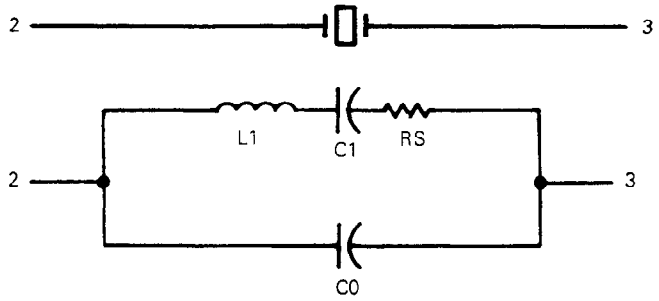
Nominal Crystal Parameters*

	3.58 MHz	4.00 MHz	5.0 MHz
RS	60 Ω	50 Ω	30-50 Ω
C0	3.5 pF	6.5 pF	4-6 pF
C1	0.015 pF	0.025 pF	0.01-0.02 pF
Q	>40 K	>30 K	>20 K

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



C_L = 20 pF (typical)

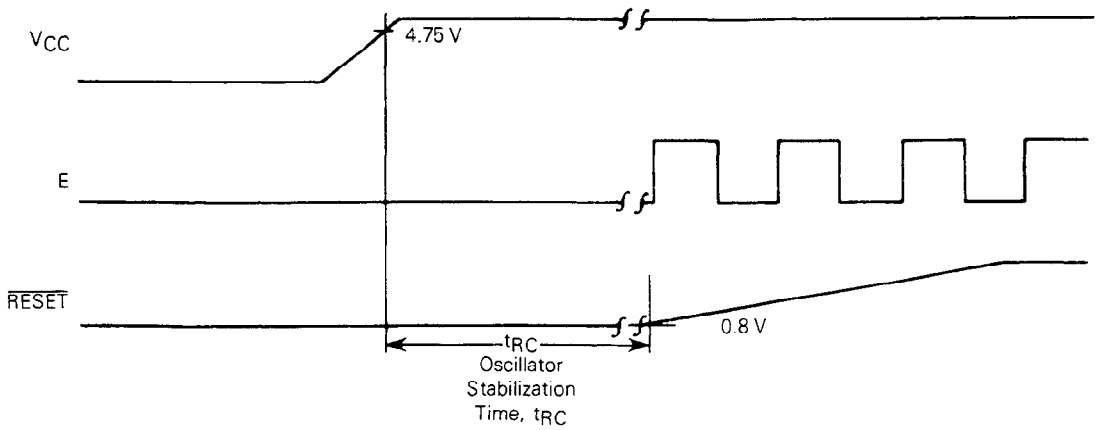


Equivalent Circuit

NOTE
TTL-compatible oscillators may be obtained from:

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

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



SC2 is configured as read/write and is used to control the direction of data bus transfers. An MPU read is enabled when read/write and E are high.

P10-P17 (PORT 1)

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

P20-P24 (PORT 2)

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

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

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

PORT 2 DATA REGISTER

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$03

P30-P37 (PORT 3)

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

PORT 3 IN SINGLE-CHIP MODE — Port 3 is an 8-bit I/O port in the single-chip mode with each line configured by the port 3 data direction register. There are also two lines, IS3 and OS3, which can be used to control port 3 data transfers.

Three port 3 options are controlled by the port 3 control and status register and are available only in single-chip mode: 1) port 3 input data can be latched using IS3 (SC1) as a control signal, 2) OS3 (SC2) can be generated by either an MPU read or write to the port 3 data register, and 3) an $\overline{\text{IRQ1}}$ interrupt can be enabled by an IS3 negative edge. Port 3 latch timing is shown in Figure 4.

PORT 3 CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
IS3 Flag	IS3 $\overline{\text{IRQ1}}$	X	OSS	Latch Enable	X	X	X	$0F

Bits 0-2 Not used.

Bit 3 **Latch Enable** — This bit controls the input latch for port 3. If set, input data is latched by an IS3 negative edge. The latch is transparent after a read of the port 3 data register. Latch enable is cleared during reset.

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

Bit 5 Not used.

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

Bit 7 **IS3 Flag** — This read-only status bit is set by an IS3 negative edge. It is cleared by a read of the port 3 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 (A7-A0) and data bus (D7-D0) in the expanded multiplexed mode where address strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high-impedance state between valid address and data to prevent bus conflicts.

P40-P47 (PORT 4)

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 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 modes 1 and 6, port 4 functions as half of the address bus and provides A8 to A15. In modes 1 and 6, the port is configured from reset as an 8-bit parallel input port where the port 4 data direction register can be written to provide any or all of upper address lines A8 to A15. Internal pullup resistors pull the lines high until the port 4 data direction register is configured where bit 0 controls A8.

RESIDENT MEMORY

The MC68701U4 has 192 bytes of on-chip RAM and 4096 bytes of on-chip UV erasable EPROM. This memory is controlled by four bits in the RAM/EPROM control register.

Thirty-two bytes of the RAM are powered through the V_{CC} standby pin and are maintainable during V_{CC} power-down. This standby portion of the RAM consists of 32 bytes located from $40 through $5F.

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/EPROM control register.

RAM/EPROM CONTROL REGISTER ($14)

The RAM/EPROM control register includes four bits: STBY PWR, RAME, PLC, and PPC. Two of these bits, STBY PWR and RAME, are used to control RAM access 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. RAME and STBY PWR are read/write bits.

The remaining two bits, PLC and PPC, control the operation of the EPROM. PLC and PPC are readable in all modes but can be changed only in mode 0. The PLC bit can be written without restriction in mode 0, but operation of the PPC bit is controlled by the state of PLC.

Associated with the EPROM are an 8-bit data latch and a 16-bit address latch. The data latch is enabled at all times, latching each data byte written to the EPROM. The address latch is controlled by the PLC bit.

A description of the RAM/EPROM control register follows.

RAM/EPROM CONTROL REGISTER

7	6	5	4	3	2	1	0	
STBY PWR	RAME	X	X	X	X	PPC	PLC	$14

Bit 0 **Programming Latch Control (PLC).** This bit controls the latch which captures the EPROM address to be programmed and whether the PCC bit can be cleared. The latch is triggered by an MPU write to a location in the EPROM. This bit is set during reset and can be cleared only in mode 0. The PLC bit is defined as follows:

PLC = 0 — EPROM address latch enabled; EPROM address is latched during MPU writes to the EPROM.

PLC = 1 — EPROM address latch is transparent.

Bit 1 **Programming Power Control (PPC).** This bit gates power from the $\overline{\text{RESET}}$/Vpp pin to the EPROM programming circuit. PPC is set during reset and whenever the PLC bit is set. It can be cleared only if operating in mode 0, and if PLC has been previously cleared. The PPC bit is defined as follows:

PPC = 0 — EPROM programming power (Vpp) applied.

PPC = 1 — EPROM programming power (Vpp) is not applied.

Bit 2-5 Unused.

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 reset. If RAME is clear, any access to a RAM address is external. If RAME is set, the RAM is included in the internal map.

Bit 7 **Standby Power (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.

Note that if PPC and PLC are set, they cannot be simultaneously cleared with a single MPU write. The PLC bit must be cleared prior to attempting to clear PPC. If both PPC and PLC are clear, setting PLC will also set PPC. In addition, it is assumed that Vpp is applied to the $\overline{\text{RESET}}$/Vpp pin whenever PCC is clear. If this is not the case, the result is undefined.

ERASING THE MC68701U4 EPROM

Ultraviolet erasure will clear all bits of the EPROM to the zero state. The MC68701U4 EPROM is programmed by erasing it to zeros and entering ones into the desired bit locations.

The MC68701U4 EPROM can be erased by exposure to high intensity ultraviolet light with a wave length of 2537 angstroms for a minimum of 30 minutes. The recommended integrated dose (ultraviolet intensity times exposure time) is 15 watts/centimeter. The lamps should be used without shortwave filters, the MC68701U4 should be positioned about one inch away from the ultraviolet tubes, and the transparent lid should not be covered.

The MC68701U4 transparent lid should always be covered after erasing. This protects both the EPROM and light-sensitive nodes from accidental exposure to ultraviolet light.

PROGRAMMING THE MC68701U4 EPROM

When the MC68701U4 is released from reset in mode 0, a vector is fetched from location $BFFE:$BFFF. This provides a method for an external program to obtain control of the microcomputer with access to every location in the EPROM.

To program the EPROM, it is necessary to operate the MC68701U4 in mode 0 under the control of a program resident in external memory which can facilitate loading and programming of the EPROM. After the pattern has been loaded

into external memory, the EPROM can be programmed as follows:

a. Apply programming power (V_{PP}) to the $\overline{RESET}/V_{PP}$ pin.
b. Clear the PLC control bit and set the PPC bit by writing $FE to the RAM/EPROM control register.
c. Write data to the next EPROM location to be programmed. Triggered by an MPU write to the EPROM, internal latches capture both the EPROM address and the data byte.
d. Clear the PPC bit for programming time, t_{pp}, by writing $FC to the RAM/EPROM control register and waiting for time, t_{pp}. This step gates the programming power (V_{PP}) from the $\overline{RESET}/V_{PP}$ pin to the EPROM which programs the location.
e. Repeat steps b through d for each byte to be programmed.
f. Set the PLC and PPC bits by writing $FF to the RAM/EPROM control register.
g. Remove the programming power (V_{PP}) from the $\overline{RESET}/V_{PP}$ pin. The EPROM can now be read and verified.

Because the erased state of an EPROM byte is $00, it is not necessary to program a location which is to contain $00. Finally, it should be noted that the result of inadvertently programming a location more than once is the logical OR of the data patterns.

PROGRAMMABLE TIMER

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

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

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

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

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

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

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

The two input capture registers are 16-bit read-only registers used to store the free-running counter when a "proper" input transition occurs as defined by the corresponding input edge bit (IEDG1 or IEDG2). The input pin's data direction register should be configured as an input, but the edge detect circuit always senses P10 and P20 even when configured as an output. The counter value will be latched into the input capture registers on the second negative edge of the E clock following the transition.

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 REGISTERS

Four registers are used to provide the MC68701U4 with control and status information about the three output compare functions, the timer overflow function, and the two input edge functions of the timer. They are:

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

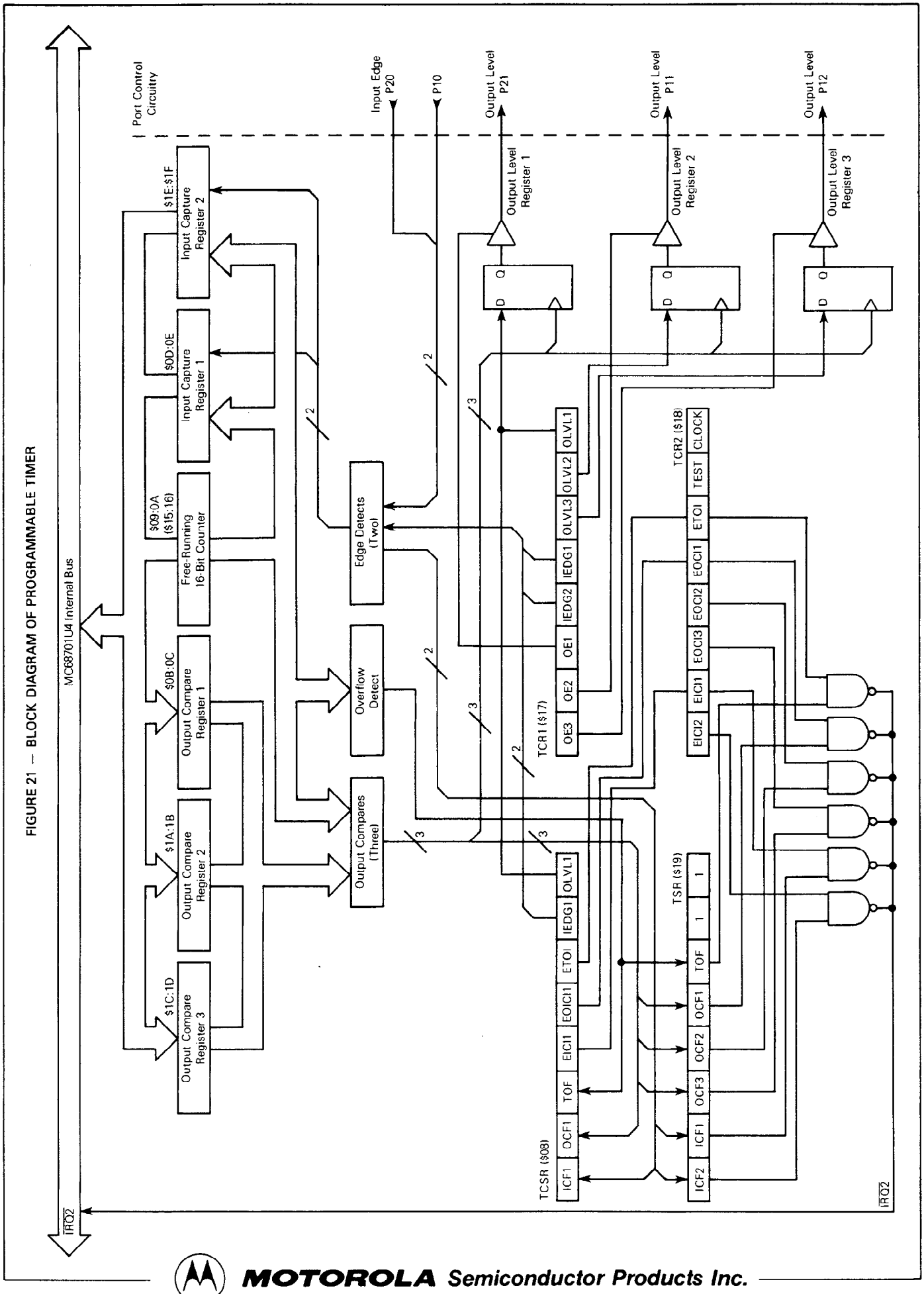


FIGURE 21 — BLOCK DIAGRAM OF PROGRAMMABLE TIMER

TIMER CONTROL AND STATUS REGISTER (TCSR) ($08) — The timer control and status register is an 8-bit register of which all bits are readable, while only bits 0-4 can be written. All the bits in this register are also accessible through the two timer control registers and the timer status register. The three most significant bits provide the timer status and indicate if:

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

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

TIMER CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
ICF1	OCF1	TOF	EICI1	EOCI1	ETOI	IEDG1	OLVL1	$08

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

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

Bit 2 **Enable Timer Overflow Interrupt** — When set, an $\overline{IRQ2}$ interrupt will be generated when the timer overflow flag is set; when clear, the interrupt is inhibited. ETOI is cleared during reset. Refer to **TIMER CONTROL REGISTER 2 (TCR2) ($18)**.

Bit 3 **Enable Output Compare Interrupt 1** — When set, an $\overline{IRQ2}$ interrupt will be generated when output compare flag 1 is set; when clear, the interrupt is inhibited. EOCI1 is cleared during reset. Refer to **TIMER CONTROL REGISTER 2 (TCR2) ($18)**.

Bit 4 **Enable Input Capture Interrupt 1** — When set, an $\overline{IRQ2}$ interrupt will be generated when input capture flag 1 is set; when clear, the interrupt is inhibited. EICI1 is cleared during reset. Refer to **TIMER CONTROL REGISTER 2 (TCR2) ($18).**

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

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

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

TIMER CONTROL REGISTER 1 (TCR1) ($17) — Timer control register 1 is an 8-bit read/write register which contains the control bits for interfacing the output compare and input capture registers to the corresponding I/O pins.

TIMER CONTROL REGISTER 1

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

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

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

Bit 2 **Output Level 3** — OLVL3 is clocked to output level register 3 by a successful output compare and will appear at P12 if bit 2 of port 1 data direction register is set and the OE3 control bit is set. OLVL3 and output level register 3 are cleared during reset.

Bit 3 **Input Edge 1** — IEDG1 is cleared during reset and controls which level transition on P20 will trigger a counter transfer to input capture register 1.
IEDG1 = 0 transfer on a negative-edge
IEDG1 = 1 transfer on a positive-edge
Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 4 **Input Edge 2** — IEDG2 is cleared during reset and controls which level transition on P10 will trigger a counter transfer to input capture register 2.
IEDG2 = 0 transfer on a negative-edge
IEDG2 = 1 transfer on a positive-edge

Bit 5 **Output Enable 1** — OE1 is set during reset and enables the contents of output level register 1 to be connected to P21 when bit 1 of port 2 data direction register is set.
OE1 = 0 port 2 bit 1 data register output
OE1 = 1 output level register 1

Bit 6 **Output Enable 2** — OE2 is cleared during reset and enables the contents of output level register 2 to be connected to P11 when bit 1 of port 1 data direction register is set.
OE2 = 0 port 1 bit 1 data register output
OE2 = 1 output level register 2

Bit 7 **Output Enable 3** — OE3 is cleared during reset and enables the contents of output level register 3 to be connected to P12 when bit 2 of port 1 data direction register is set

OE3 = 0 port 1 bit 2 data register output
OE3 = 1 output level register 3

TIMER CONTROL REGISTER 2 (TCR2) ($18) — Timer control register 2 is an 8-bit read/write register (except bits 0 and 1) which enable the interrupts associated with the free-running counter, the output compare registers, and the input capture registers. In test mode 0, two more bits (clock and test) are available for checking the timer.

TIMER CONTROL REGISTER 2
(Non-Test Modes)

7	6	5	4	3	2	1	0	
EICI2	EICI1	EOCI3	EOCI2	EOCI1	ETOI	1	1	$18

Bits 0-1 **Read-Only Bits** — When read, these bits return a value of 1. Refer to **TIMER CONTROL REGISTER 2 (Test Mode)**.

Bit 2 **Enable Timer Overflow Interrupt** — When set, an $\overline{IRQ2}$ interrupt will be generated when the timer overflow flag is set; when clear, the interrupt is inhibited. ETOI is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 3 **Enable Output Compare Interrupt 1** — When set, an $\overline{IRQ2}$ interrupt will be generated when the output compare flag 1 is set; when clear, the interrupt is inhibited. EOCI1 is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 4 **Enable Output Compare Interrupt 2** — When set, an $\overline{IRQ2}$ interrupt will be generated when the output compare flag 2 is set; when clear, the interrupt is inhibited. EOCI2 is cleared during reset.

Bit 5 **Enable Output Compare Interrupt 3** — When set, an $\overline{IRQ2}$ interrupt will be generated when the output compare flag 3 is set; when clear, the interrupt is inhibited. EOCI3 is cleared during reset.

Bit 6 **Enable Input Capture Interrupt 1** — When set, an $\overline{IRQ2}$ interrupt will be generated when the input capture flag 1 is set; when clear, the interrupt is inhibited. EICI1 is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 7 **Enable Input Capture Interrupt 2** — When set, an $\overline{IRQ2}$ interrupt will be generated when the input capture flag 2 is set; when clear, the interrupt is inhibited. EICI2 is cleared during reset.

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

TIMER CONTROL REGISTER 2
(Test Mode)

7	6	5	4	3	2	1	0	
EICI2	EICI1	EOCI3	EOCI2	EOCI1	ETOI	TEST	CLOCK	$18

Bit 0 **CLOCK** — The CLOCK control bit selects which half of the 16-bit free-running counter (MSB or LSB) should be clocked with E. The CLOCK bit is a read/write bit only in mode 0 and is set during reset.

CLOCK = 0 — Only the eight most significant bits of the free-running counter run with TEST = 0.
CLOCK = 1 — Only the eight least significant bits of the free-running counter run when TEST = 0.

Bit 1 **TEST** — the TEST control bit enables the timer test mode. TEST is a read/write bit in mode 0 and is set during reset.

TEST = 0 — Timer test mode enabled:
a) The timer LSB latch is transparent which allows the LSB to be read independently of the MSB.
b) Either the MSB or the LSB of the timer is clocked by E, as defined by the CLOCK bit.

TEST = 1 — Timer test mode disabled.

Bits 2-7 See **TIMER CONTROL REGISTER 2 (Non-Test Modes)**. (These bits function the same as in the non-test modes.)

TIMER STATUS REGISTER (TSR) ($19) — The timer status register is an 8-bit read-only register which contains the flags associated with the free-running counter, the output compare registers, and the input capture registers.

TIMER STATUS REGISTER

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

Bits 0-1 Not used.

Bit 2 **Timer Overflow Flag** — The TOF is set when the counter contains all ones ($FFFF). It is cleared by reading the TSR or the TCSR (with TOF set) and then the counter high byte ($09), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 3 **Output Compare Flag 1** — OCF1 is set when output compare register 1 matches the free-running counter. OCF1 is cleared by reading the TSR or the TCSR (with OCF1 set) and then writing to output compare register 1 ($0B or $0C), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 4 **Output Compare Flag 2** — OCF2 is set when output compare register 2 matches the free-running counter. OCF2 is cleared by reading the TSR (with OCF2 set) and then writing to output compare register 2 ($1A or $1B), or during reset.

Bit 5 **Output Compare Flag 3** — OCF3 is set when output compare register 3 matches the free-running counter. OCF3 is cleared by reading the TSR (with OCF3 set) and then writing to output compare register 3 ($1C or $1D), or during reset.

Bit 6 **Input Capture Flag 1** — ICF1 is set to indicate that a proper level transition has occurred; it is cleared by reading the TSR or the TCSR (with ICF1 set) and the input capture register 1 high byte ($0D), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 7 **Input Capture Flag 2** — ICF2 is set to indicate that a proper level transition has occurred; it is cleared by reading the TSR (with ICF2 set) and the input capture register 2 high byte ($1E), or durihg 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 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 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, 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

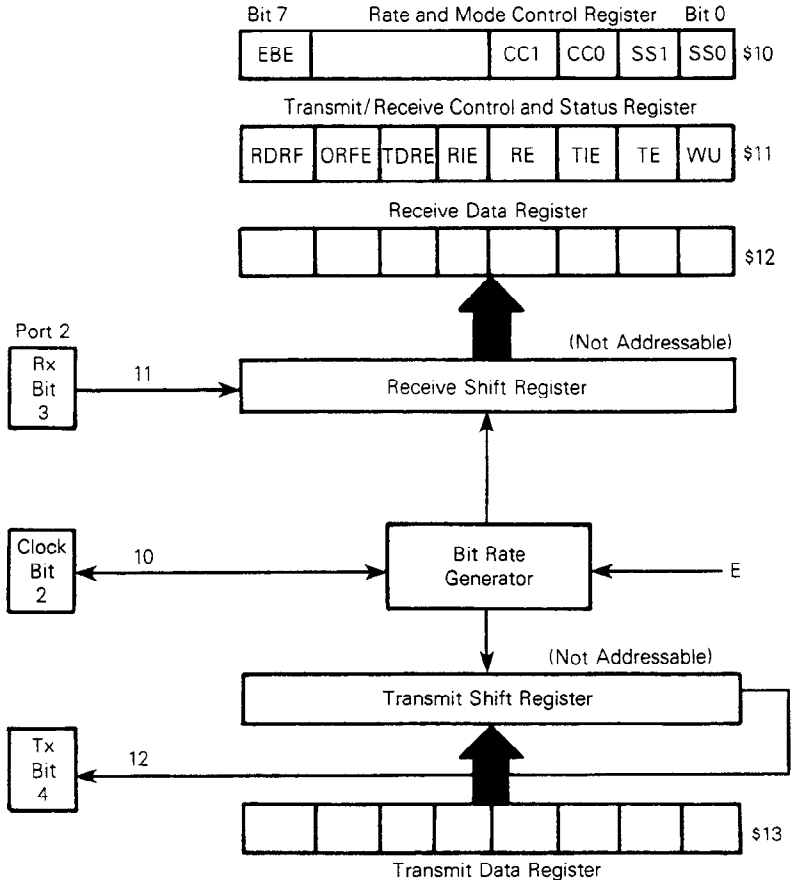
The following features of the SCI are programmable:

- Format: standard mark/space (NRZ) or bi-phase
- Clock: external or internal bit rate clock
- Baud: one of eight 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 22. It is controlled by the rate and mode control register and the transmit/receive control and status register. Data is transmitted and received utilizing a write-only transmit register and a read-only receive register. The shift registers are not accessible to software.

FIGURE 22 — SCI REGISTERS



RATE AND MODE CONTROL REGISTER (RMCR) ($10)

— The rate and mode control register controls the SCI bit rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of five write-only bits which are cleared during reset. The two least significant bits in conjunction with bit 7 control the bit rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER

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

Bit 1:Bit 0 **SS1:SS0 Speed Select** — These two bits select the baud when using the internal clock. Eight rates may be selected (in conjunction with bit 7) 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.

Bits 4-6 Not used.

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

EBE = 0 standard MC6801 baud rates
EBE = 1 additional baud rates

If both CC1 and CC0 are set, an external TTL-compatible clock must be connected to P22 at eight times (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 in mode 0 can disturb serial operations.

TABLE 6 — SCI BIT TIMES AND RATES

EBE	SS1:SS0		$4 f_o \rightarrow$	2.4576 MHz		4.0 MHz		4.9152 MHz	
				614.4 kHz		1.0 MHz		1.2288 MHz	
			E	Baud	Time	Baud	Time	Baud	Time
0	0	0	÷16	38400.0	26 μs	62500.0	16.0 μs	76800.0	13.0 μs
0	0	1	÷128	4800.0	208.3 μs	7812.5	128.0 μs	9600.0	104.2 μs
0	1	0	÷1024	600.0	1.67 ms	976.6	1.024 ms	1200.0	833.3 μs
0	1	1	÷4096	150.0	6.67 ms	244.1	4.096 ms	300.0	3.33 ms
1	0	0	÷64	9600.0	104.2 μs	15625.0	64 μs	19200.0	52.0 μs
1	0	1	÷256	2400.0	416.6 μs	3906.3	256 μs	4800.0	208.3 μs
1	1	0	÷512	1200.0	833.3 μs	1953.1	512 μs	2400.0	416.6 μs
1	1	1	÷2048	300.0	3.33 ms	488.3	2.05 ms	600.0	01.67 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

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** — When set, WU enables the wake-up function; it is cleared by ten consecutive ones or during reset. WU will not be set if the line is idle. Refer to **WAKE-UP FEATURE.**

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

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

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

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

Bit 5 **Transmit Data Register Empty** — TDRE is set when the transmit data register is transferred to the output serial shift register or during reset. It is cleared by reading the TRCSR (with TDRE set) and then writing to the transmit data register. Additional data will be transmitted only if TDRE has been cleared.

Bit 6 **Overrun Framing Error** — If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the receiver data register with RDRF still set. A receiver framing error has occurred when the stop bit (1) is not found in the tenth bit time. 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 is set when the input serial shift register is transferred to the receive data register, or during reset.

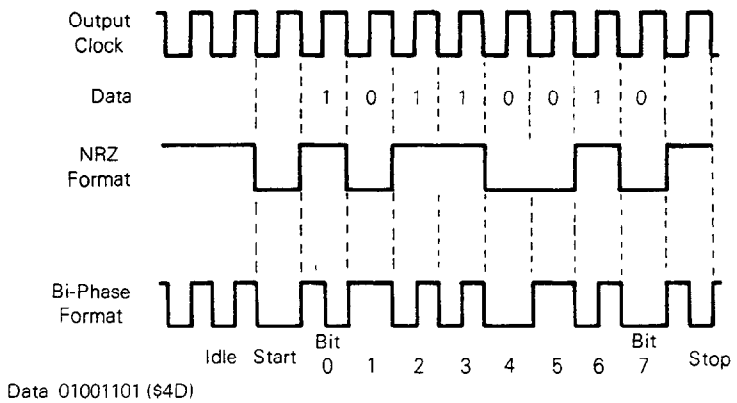
SERIAL OPERATIONS

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

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

The start bit (0), eight data bits (beginning with bit 0), and a stop bit (1) will be transmitted. If TDRE is still set when the next byte transfer occurs, ones will be sent until more data is provided. In 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 23.

FIGURE 23 — SCI DATA FORMATS



Data 01001101 ($4D)

INSTRUCTION SET

The MC68701U4 is directly source compatible with the MC6801 and upward source and object code compatible with the MC6800. Execution times of key instructions have been reduced and several instructions have been added, including a hardware multiply. A list of new operations added to the MC6800 instruction set is shown in Table 1.

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

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

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	INDXD	4	2	D4	ANDB	↑	3	2
05	ASLD	↑	3	1	39	RTS	↑	5	1	6D	TST	↑	6	2	A1	CMPA	↑	4	2	D5	BITB	↑	3	2
06	TAP	↑	2	1	3A	ABX	↑	3	1	6E	JMP	↓	3	2	A2	SBCA	↑	4	2	D6	LDAB	↑	3	2
07	TPA	↑	2	1	3B	RTI	↑	10	1	6F	CLR	INDXD	6	2	A3	SUBD	↑	6	2	D7	STAB	↑	3	2
08	INX	↑	3	1	3C	PSHX	↑	4	1	70	NEG	EXTND	6	3	A4	ANDA	↑	4	2	D8	EORB	↑	3	2
09	DEX	↑	3	1	3D	MUL	↑	10	1	71	•	↑			A5	BITA	↑	4	2	D9	ADCB	↑	3	2
0A	CLV	↑	2	1	3E	WAI	↑	9	1	72	•	↑			A6	LDAA	↑	4	2	DA	ORAB	↑	3	2
0B	SEV	↑	2	1	3F	SWI	↑	12	1	73	COM	↑	6	3	A7	STAA	↑	4	2	DB	ADDB	↑	3	2
0C	CLC	↑	2	1	40	NEGA	↑	2	1	74	LSR	↑	6	3	A8	EORA	↑	4	2	DC	LDD	↑	4	2
0D	SEC	↑	2	1	41	•	↑			75	•	↑			A9	ADCA	↑	4	2	DD	STD	↑	4	2
0E	CLI	↑	2	1	42	•	↑			76	ROR	↑	6	3	AA	ORAA	↑	4	2	DE	LDX	↓	4	2
0F	SEI	↑	2	1	43	COMA	↑	2	1	77	ASR	↑	6	3	AB	ADDA	↑	4	2	DF	STX	DIR	4	2
10	SBA	↑	2	1	44	LSRA	↑	2	1	78	ASL	↑	6	3	AC	CPX	↑	6	2	E0	SUBB	INDXD	4	2
11	CBA	↑	2	1	45	•	↑			79	ROL	↑	6	3	AD	JSR	↑	6	2	E1	CMPB	↑	4	2
12	•	↑			46	RORA	↑	2	1	7A	DEC	↑	6	3	AE	LDS	↓	5	2	E2	SBCB	↑	4	2
13	•	↑			47	ASRA	↑	2	1	7B	•	↑			AF	STS	INDXD	5	2	E3	ADDD	↑	6	2
14	•	↑			48	ASLA	↑	2	1	7C	INC	↑	6	3	B0	SUBA	EXTND	4	3	E4	ANDB	↑	4	2
15	•	↑			49	ROLA	↑	2	1	7D	TST	↑	6	3	B1	CMPA	↑	4	3	E5	BITB	↑	4	2
16	TAB	↑	2	1	4A	DECA	↑	2	1	7E	JMP	↓	3	3	B2	SBCA	↑	4	3	E6	LDAB	↑	4	2
17	TBA	↑	2	1	4B	•	↑			7F	CLR	EXTND	6	3	B3	SUBD	↑	6	3	E7	STAB	↑	4	2
18	•	↓			4C	INCA	↑	2	1	80	SUBA	IMMED	2	2	B4	ANDA	↑	4	3	E8	EORB	↑	4	2
19	DAA	INHER	2	1	4D	TSTA	↑	2	1	81	CMPA	↑	2	2	B5	BITA	↑	4	3	E9	ADCB	↑	4	2
1A	•				4E	T				82	SBCA	↑	2	2	B6	LDAA	↑	4	3	EA	ORAB	↑	4	2
1B	ABA	INHER	2	1	4F	CLRA	↑	2	1	83	SUBD	↑	4	3	B7	STAA	↑	4	3	EB	ADDB	↑	4	2
1C	•	↑			50	NEGB	↑	2	1	84	ANDA	↑	2	2	B8	EORA	↑	4	3	EC	LDD	↑	5	2
1D	•	↑			51	•	↑			85	BITA	↑	2	2	B9	ADCA	↑	4	3	ED	STD	↑	5	2
1E	•	↑			52	•	↑			86	LDAA	↑	2	2	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	•	↓							
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	•								

*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.

PROGRAMMING MODEL

A programming model for the MC68701U4 is shown in Figure 8. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulators A and/or B. Other registers are defined as follows:

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

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

INDEX REGISTER – The index register is a 16-bit register 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 REGISTER – The condition code register indicates the results of an instruction and includes the following five condition bits: negative (N), zero (Z), overflow (V), carry/borrow from MSB (C), and half carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I bit) and inhibits all maskable interrupts when set. The two unused bits, B6 and B7, are read as ones.

ADDRESSING MODES

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

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

DIRECT 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 is used to reference memory without changing the index register. These are two byte instructions.

INHERENT ADDRESSING – The operand(s) is a register and no memory reference is required. These are single byte instructions.

RELATIVE ADDRESSING – Relative addressing is used only for branch instructions. If the branch condition is true, the program counter is overwritten with the sum of 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.

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 except in mode 0. "High order" byte refers to the most significant byte of a 16-bit value. During unused bus cycles, the address bus is forced to $FFFF and R/$\overline{W}$ is high.

TABLE 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

Pointer Operations	MNEM	Immed			Direct			Index			Extend			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 Register	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X − M:M + 1	•	•	↕	↕	↕	↕
Decrement Index Register	DEX													09	3	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pointer	DES													34	3	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Register	INX													08	3	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pointer	INS													31	3	1	1 SP + 1 → SP	•	•	•	•	•	•
Load Index Register	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	↕	↕	R	•
Load Stack Pointer	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M + 1) → SP_L	•	•	↕	↕	R	•
Store Index Register	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	↕	↕	R	•
Store Stack Pointer	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	↕	↕	R	•
Index Reg → Stack Pointer	TXS													35	3	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Register	TSX													30	3	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	3	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	4	1	X_L → M_{SP}, SP − 1 → SP X_H → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	•	•	•	•	•	•

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

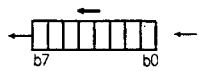
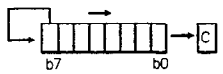
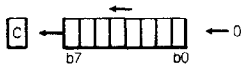
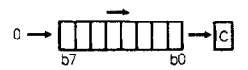
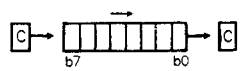
Accumulator and Memory Operations	MNEM	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		5 H	4 I	3 N	2 Z	1 V	0 C
Add Accumulators	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A • 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		•	•	↕	↕	↕	↕
Shift Left Double	ASLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•	↕	↕	↕	↕
	ASRA													47	2	1		•	•	↕	↕	↕	↕
	ASRB													57	2	1		•	•	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A • M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B • M	•	•	↕	↕	R	•
Compare Accumulators	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				M → M	•	•	↕	↕	R	S
	COMA													43	2	1	A → A	•	•	↕	↕	R	S
	COMB													53	2	1	B → B	•	•	↕	↕	R	S

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

Accumulator and Memory Operations		Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
																		5	4	3	2	1	0
	MNEM	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Decimal Adjust, A	DAA													19	2	1	Adj binary sum to BCD	•	•	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M − 1 → 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		•	•	↕	↕	↕	↕
	LSLB													58	2	1		•	•	↕	↕	↕	↕
	LSLD													05	3	2		•	•	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					•	•	R	↕	↕	↕
	LSRA													44	2	1		•	•	R	↕	↕	↕
	LSRB													54	2	1		•	•	R	↕	↕	↕
	LSRD													04	3	1		•	•	R	↕	↕	↕
Multiply	MUL													3D	10	1	A × 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 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			Relative			Index			Extend			Inherent			Branch Test	Condition Code Reg.					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		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 ⊕ 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 24	•	•	•	•	•	•
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1		•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1	See Special Operations-Figure 24	•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait For Interrupt	WAI													3E	9	1		•	•	•	•	•	•

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

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

LEGEND

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

CONDITION CODE SYMBOLS

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

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E CYCLES

	ADDRESSING MODE					
	Immediate	Direct	Extended	Indexed	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	●

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

Address Mode and 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	Opcode Address Opcode Address + 1	1 1	Opcode Operand Data
LDS LDX LDD	3	1 2 3	Opcode Address Opcode Address + 1 Opcode Address + 2	1 1 1	Opcode Operand Data (High Order Byte) Operand Data (Low Order Byte)
CPX SUBD ADDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Opcode Address + 2 Address Bus FFFF	1 1 1 1	Opcode 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	Opcode Address Opcode Address + 1 Address of Operand	1 1 1	Opcode Address of Operand Operand Data
STA	3	1 2 3	Opcode Address Opcode Address + 1 Destination Address	1 1 0	Opcode Destination Address Data from Accumulator
LDS LDX LDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Opcode Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte)
STS STX STD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Address of Operand + 1	1 1 0 0	Opcode Address of Operand Register Data (High Order Byte) Register Data (Low Order Byte)
CPX SUBD ADDD	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Operand Address Operand Address + 1 Address Bus FFFF	1 1 1 1 1	Opcode 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	Opcode Address Opcode Address + 1 Subroutine Address Stack Pointer Stack Pointer − 1	1 1 1 0 0	Opcode Irrelevant Data First Subroutine Opcode Return Address (Low Order Byte) Return Address (High Order Byte)

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 2 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ 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	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/$\overline{W}$ 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.

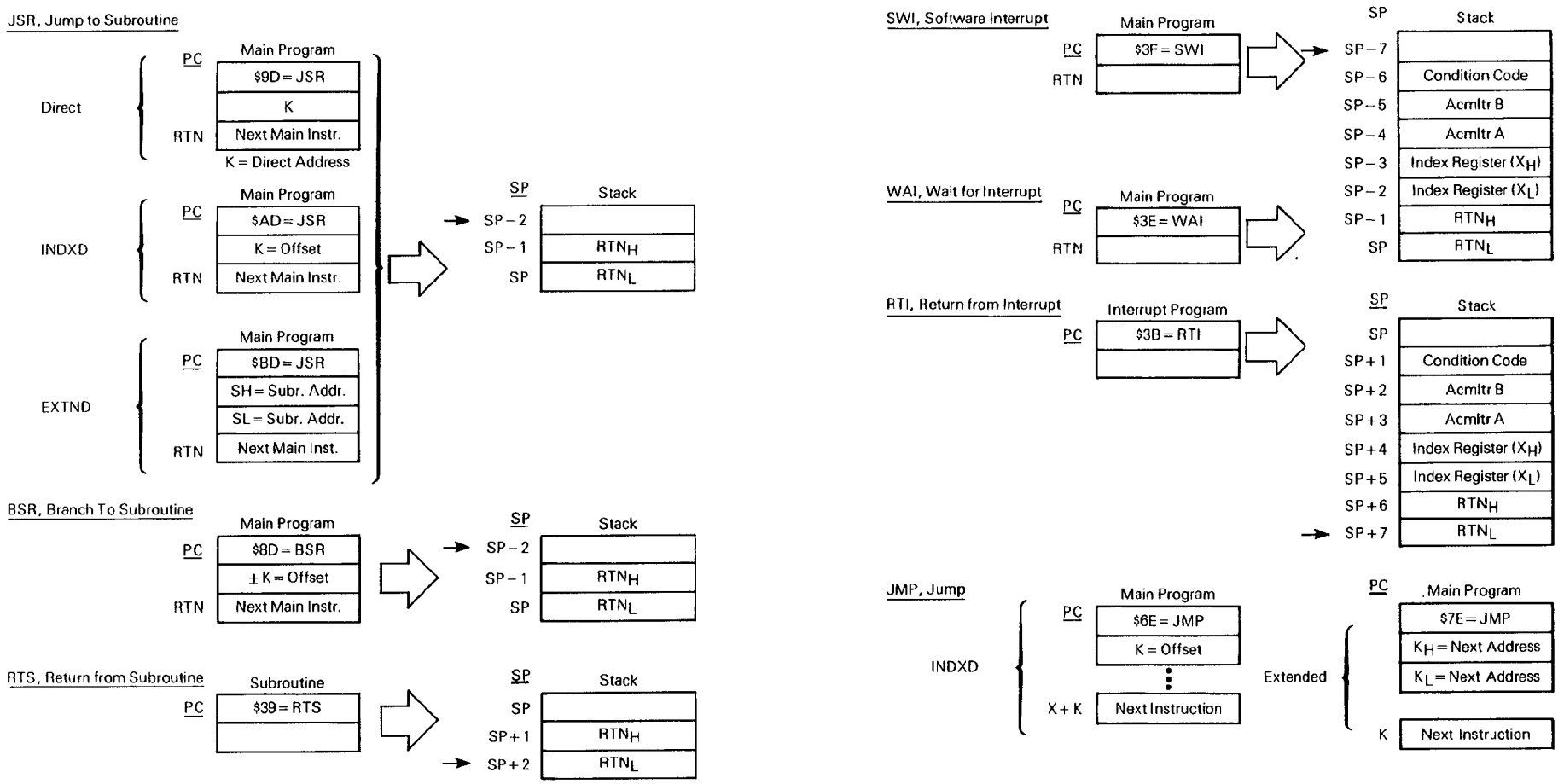
TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 4 of 5)

Address Mode and 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	Opcode Address Opcode Address + 1	1 1	Opcode Opcode of Next Instruction
ABX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Opcode Address Opcode Address + 1 Previous Stack Pointer Contents	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 0	Opcode Opcode of Next Instruction Accumulator Data
TSX	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
TXS	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Opcode Opcode of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer − 1	1 1 0 0	Opcode Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI	9	1 2 3 4 5 6 7 8 9	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer − 1 Stack Pointer − 2 Stack Pointer − 3 Stack Pointer − 4 Stack Pointer − 5 Stack Pointer − 6	1 1 0 0 0 0 0 0 0	Opcode Opcode of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte) Index Register (Low Order Byte) Index Register (High Order Byte) Contents of Accumulator A Contents of Accumulator B Contents of Condition Code Register

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

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT (Continued)					
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)

FIGURE 24 — SPECIAL OPERATIONS

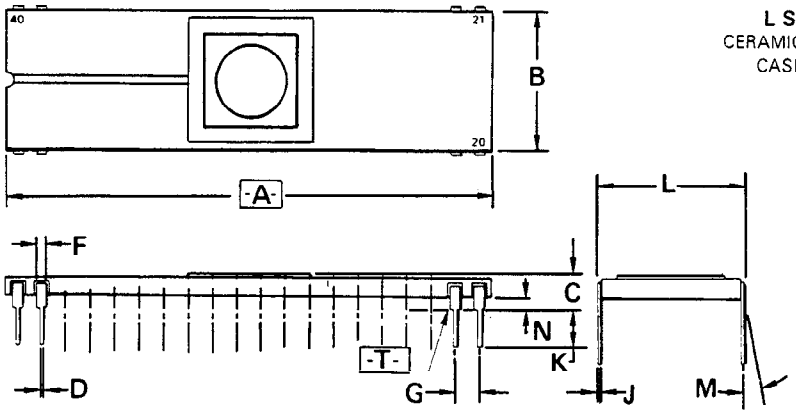


Legend:
RTN = Address of next instruction in Main Program to be executed upon return from subroutine
RTN$_H$ = Most significant byte of Return Address
RTN$_L$ = Least significant byte of Return Address
→ = Stack Pointer After Execution
K = 8-bit Unsigned Value

This page intentionally left blank.

This page intentionally left blank.

PACKAGE DIMENSIONS



L SUFFIX
CERAMIC PACKAGE
CASE 715-06

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	50.29	51.31	1.980	2.020
B	14.63	15.49	0.576	0.610
C	3.18	5.08	0.125	0.200
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°	–	10°
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.