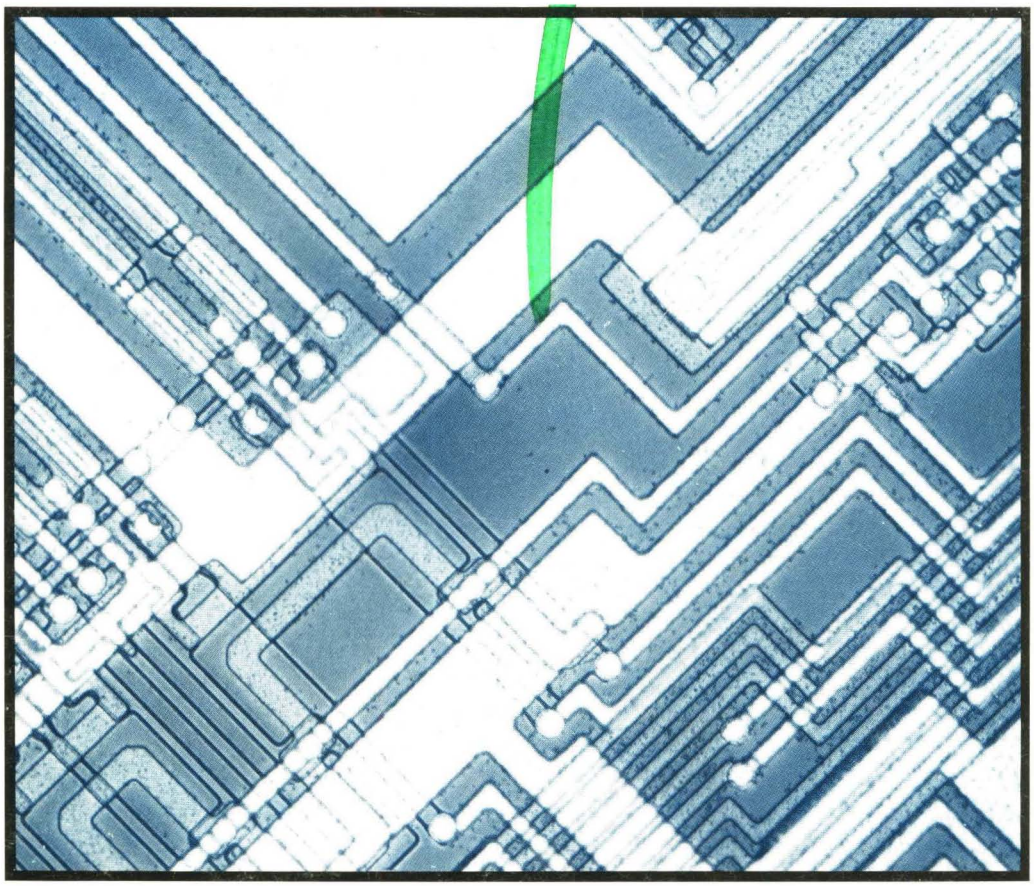
Order this data
sheet by MC6804J1/D

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

Advance Information

8-Bit Microcomputers

MC6804J1
MC6804J2
MC6804P2
MC68704P2



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

MOTOROLA

ADI1248
Replaces ADI-987, ADI-1013

Introduction 1

Functional Pin Description, Memory, CPU, and Registers 2

Timer 3

Interrupt, Self-Test, Reset, and Internal Clock Generator 4

Input/Output Ports 5

Software and Instruction Set 6

Electrical Specifications 7

Ordering Information 8

Mechanical Data 9

MC6804J1 A

MC6804J2 B

MC68704P2 C

MC6804J1/J2/P2 Emulation D

1 Introduction

2 Functional Pin Description, Memory, CPU, and Registers

3 Timer

4 Interrupt, Self-Test, Reset, and Internal Clock Generator

5 Input/Output Ports

6 Software and Instruction Set

7 Electrical Specifications

8 Ordering Information

9 Mechanical Data

A MC6804J1

B MC6804J2

C MC68704P2

D MC6804J1/J2/P2 Emulation

TABLE OF CONTENTS

Paragraph Number	Title	Page Number
	Section 1 Introduction	
1.1	General	1-1
1.2	Features	1-3
	Section 2 Functional Pin Description, Memory, CPU, and Registers	
2.1	Functional Pin Description	2-1
2.1.1	V_{CC} and V_{SS}	2-1
2.1.2	$\overline{IRQ}$	2-1
2.1.3	XTAL and EXTAL	2-1
2.1.4	TIMER	2-1
2.1.5	$\overline{RESET}$	2-1
2.1.6	MDS	2-2
2.1.7	Input/Output Lines (PA0-PA7, PB0-PB7, PC0-PC3)	2-2
2.2	Memory	2-2
2.3	Central Processing Unit	2-2
2.4	Registers	2-4
2.4.1	Accumulator (A)	2-4
2.4.2	Indirect Registers (X, Y)	2-4
2.4.3	Program Counter	2-4
2.4.4	Flags (C, Z)	2-5
2.4.5	Stack	2-5
	Section 3 Timer	
3.1	Introduction	3-1
3.2	Timer Registers	3-3
3.2.1	Timer Count Register (TCR)	3-3
3.2.2	Timer Status/Control Register (TSCR)	3-3
3.2.3	Timer Prescaler Register	3-4
	Section 4 Interrupt, Self-Test, Reset, and Internal Clock Generator	
4.1	Interrupt	4-1
4.1.1	Edge-Sensitive Option	4-1
4.1.2	Level-Sensitive Option	4-1
4.1.3	Power Up and Timing	4-1
4.1.4	Timer Interrupt	4-3
4.2	Self-Test	4-4
4.3	Reset	4-4
4.4	Internal Clock Generator Options	4-4

TABLE OF CONTENTS
(Continued)

Paragraph Number	Title	Page Number
	Section 5 **Input/Output Ports**	
5.1	Input/Output	5-1
5.2	Registers	5-3
5.2.1	Port Data Registers	5-3
5.2.2	Port Data Direction Registers	5-4
	Section 6 **Software and Instruction Set**	
6.1	Software	6-1
6.1.1	Bit Manipulation	6-1
6.1.2	Addressing Modes	6-2
6.1.2.1	Immediate	6-2
6.1.2.2	Direct	6-2
6.1.2.3	Short Direct	6-2
6.1.2.4	Extended	6-2
6.1.2.5	Relative	6-2
6.1.2.6	Bit Set/Clear	6-2
6.1.2.7	Bit Test And Branch	6-3
6.1.2.8	Register-Indirect	6-3
6.1.2.9	Inherent	6-3
6.2	Instruction Set	6-3
6.2.1	Register/Memory Instructions	6-3
6.2.2	Read-Modify-Write Instructions	6-3
6.2.3	Branch Instructions	6-6
6.2.4	Bit Manipulation Instructions	6-6
6.2.5	Control Instructions	6-6
6.2.6	Alphabetical Listing	6-6
6.2.7	Opcode Map Summary	6-6
6.3	Implied Instructions	6-6
	Section 7 **Electrical Specifications**	
7.1	Introduction	7-1
7.2	Maximum Ratings	7-1
7.3	Thermal Characteristics	7-1
7.4	Power Considerations	7-2
7.5	Electrical Characteristics	7-2
7.6	Switching Characteristics	7-2
7.7	Port DC Electrical Characteristics	7-3

TABLE OF CONTENTS
(Continued)

Paragraph Number	Title	Page Number
	Section 8 **Ordering Information**	
8.1	Introduction	8-1
8.2	Flexible Disks	8-1
8.3	EPROMs	8-1
8.4	Verification Media	8-2
8.5	ROM Verification Units (RVUs)	8-2
8.6	Ordering Information	8-2
	Section 9 **Mechanical Data**	
9.1	Pin Assignments	9-1
9.2	Package Dimensions	9-2
	Appendix A **MC6804J1**	
A.1	Introduction	A-1
A.1.1	Features	A-1
A.1.2	Block Diagram	A-1
A.2	Functional Pin Description, Memory, CPU, and Registers	A-1
A.2.1	Input/Output Lines (PA4-PA7, PB0-PB7)	A-1
A.2.2	Memory	A-2
A.3	Interrupt, Self-Test, Reset, and Internal Clock Generator	A-3
A.3.1	Self-Test	A-3
A.4	Input/Output Ports	A-5
A.4.1	Input/Output	A-5
A.4.2	Port A and B Data Registers	A-6
A.4.3	Port A and B Data Direction Registers	A-6
A.5	Ordering Information	A-6
A.6	Mechanical Data	A-6
A.6.1	Pin Assignments	A-7
A.6.2	Package Dimensions	A-7
	Appendix B **MC6804J2**	
B.1	Introduction	B-1
B.1.1	Features	B-1
B.1.2	Block Diagram	B-1
B.2	Functional Pin Description, Memory, CPU, and Registers	B-1
B.2.1	Input/Output Lines (PA4-PA7, PB0-PB7)	B-1
B.2.2	Memory	B-2

TABLE OF CONTENTS
(Concluded)

Paragraph Number	Title	Page Number
B.3	Interrupt, Self-Test, Reset, and Internal Clock Generator	B-3
B.3.1	Self-Test	B-3
B.4	Input/Output Ports	B-5
B.4.1	Input/Output	B-5
B.4.2	Port A and B Data Registers	B-6
B.4.3	Port A and B Data Direction Registers	B-6
B.5	Ordering Information	B-6
B.6	Mechanical Data	B-6
B.6.1	Pin Assignments	B-7
B.6.2	Package Dimensions	B-7
	Appendix C MC68704P2	
C.1	Introduction	C-1
C.1.1	Features	C-1
C.1.2	Block Diagram	C-1
C.2	Functional Pin Description, Memory, CPU, and Registers	C-1
C.2.1	Memory	C-1
C.3	Breakpoint Registers	C-4
C.4	EPROM Mask Option Register	C-4
C.5	Electrical Specifications	C-5
C.5.1	Electrical Characteristics	C-5
C.5.2	Programming Operation Electrical Characteristics	C-5
C.5.3	Port DC Electrical Characteristics	C-5
C.6	Ordering Information	C-5
C.7	Mechanical Data	C-6
	Appendix D MC6804J1, MC6804J2, and MC6804P2 Emulation	
D.1	Introduction	D-1
D.2	Emulation Limitations	D-1
D.3	EPROM Erasing	D-2
D.4	Programming Operation	D-3
D.5	Programming Modes of Operation	D-3
D.5.1	Zero Check	D-3
D.5.2	Program	D-3
D.5.3	Verify	D-3
D.5.4	Test	D-3
D.6	Program Mode Selection	D-4
D.7	Preliminary Procedures	D-4
D.8	Operating Procedures	D-5
D.9	Programming Software	D-6

LIST OF ILLUSTRATIONS

Figure Number	Title	Page Number
1-1	MC6804P2 Block Diagram	1-2
2-1	MC6804P2 Memory Map	2-3
2-2	Programming Model	2-4
3-1	Timer Block Diagram	3-2
4-1	Reset and Interrupt Processing Flowchart	4-2
4-2	Typical Program Restart	4-3
4-3	Timer Interrupt Connection	4-3
4-4	Self-Test Circuit	4-5
4-5	Power-Up Reset Delay Circuit	4-6
4-6	Clock Generator Options	4-6
4-7	Crystal/Ceramic Resonator Motional Arm Parameters and Suggested PC Board Layout	4-7
4-8	Typical Frequency Selection for Resistor-Capacitor Oscillator Options	4-7
4-9	Clock Generator Timing Diagram	4-8
5-1	Typical I/O Port Circuitry	5-1
5-2	Typical Port Connections	5-2
6-1	Bit Manipulation Example	6-1
7-1	LSTTL Equivalent Test Load (Port B)	7-1
7-2	CMOS Equivalent Test Load (Ports A, B, C)	7-1
7-3	LSTTL Equivalent Test Load (Ports A, C, and TIMER)	7-1
7-4	Typical Power Dissipation	7-2
7-5	Typical V_{OL} vs I_{OL} for Ports A, C, and TIMER	7-4
7-6	Typical V_{OH} vs I_{OH} for Ports A, C, and TIMER	7-4
7-7	Typical V_{OH} vs I_{OH} for Ports A and C with CMOS Pullups	7-4
7-8	Typical V_{OL} vs I_{OL} for Port B	7-4
7-9	Typical V_{OH} vs I_{OH} for Port B	7-4
7-10	Typical V_{OH} vs I_{OH} for Port B with CMOS Pullups	7-4
A-1	MC6804J1 Block Diagram	A-2
A-2	MC6804J1 Memory Map	A-3
A-3	Self-Test	A-4
A-4	Typical Port Connections	A-5

LIST OF ILLUSTRATIONS
(Concluded)

Figure Number	Title	Page Number
B-1	MC6804J2 Block Diagram	B-1
B-2	MC6804J2 Memory Map	B-3
B-3	Self-Test	B-4
B-4	Typical Port Connections	B-5
C-1	MC68704P2 Block Diagram	C-2
C-2	MC68704P2 Memory Map	C-3
D-1	MC68704P2 EPROM MCU Programming Board/Circuitry	D-7

LIST OF TABLES

Table Number	Title	Page Number
3-1	Prescaler Coding Table	3-1
6-1	Register/Memory Instructions	6-4
6-2	Read-Modify-Write Instructions	6-5
6-3	Branch Instructions	6-7
6-4	Bit Manipulation Instructions	6-7
6-5	Control Instructions	6-8
6-6	Instruction Set	6-9
6-7	Instruction Set Opcode Map	6-10

SECTION 1
INTRODUCTION

This document describes the MC6804J1, MC6804J2, MC6804P2, and MC68704P2 Microcomputer Unit (MCU) devices which are part of the M6804 Family of 8-bit single-chip MCU devices. The M6804 Family is based on a new design concept which processes 8-bit variables serially, one bit at a time. This serial design objective reduces the die size and provides a cost effective solution to MCU applications without sacrificing performance standards.

The M6804 Family of devices appear as an 8-bit architecture similar to the M6805 Family; however, the internal hardware structure (instruction data buses to RAM and I/O) is 1-bit rather than 8-bit wide. Therefore all operations (data transfers, arithmetic and address operations) are performed serially a bit at a time.

The organization of this document is such that the MC6804P2 MCU device is described throughout **Sections 1 through 9**. Information common to all four devices are also contained in these sections. The appendices contain unique/specific information applicable to the MC6804J1/MC6804J2 MCU and MC68704P2 EPROM MCU devices.

1.1 GENERAL

The MC6804P2 MCU device (shown in Figure 1-1) contains a CPU, on-chip CLOCK, ROM, RAM, I/O, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of the M6800-based instruction set.

The MC6804J1 device is similar to the MC6804P2 device with several exceptions. These exceptions include 504 bytes of user program ROM, 72 bytes of user data space ROM, 12 bidirectional I/O port lines, and 20-pin packaging. Information throughout this document pertaining to the MC6804P2 MCU is also applicable to the MC6804J1 MCU. Difference information applicable to the MC6804J1 MCU is provided in **Appendix A** of this document.

The MC6804J2 device is similar to the MC6804P2 device with several exceptions. These exceptions include 1000 bytes of user program ROM, 72 bytes of user data space ROM, 12 bidirectional I/O port lines, and 20-pin packaging. Information throughout this document pertaining to the MC6804P2 MCU is also applicable to the MC6804J2 MCU. Difference information applicable to the MC6804J2 MCU is provided in **Appendix B** of this document.

The MC68704P2 device is an erasable programmable read-only memory (EPROM) version of the MC6804P2 device and is used to emulate the MC6804J1, MC6804J2, and MC6804P2 devices in low volume applications. The MC68704P2 device is similar to the MC6804P2 device with several exceptions. These exceptions include 1016 bytes of user program EPROM, 72 bytes of user data space EPROM, two breakpoint registers, and an EPROM mask option register. Information throughout this document pertaining to the MC6804P2 MCU is also applicable to the MC68704P2 EPROM MCU. Difference information applicable to the MC68704P2 EPROM MCU is provided in **Appendix C** of this document.

MC6804J1, MC6804J2, and MC6804P2 emulation information is provided in **Appendix D** of this document.

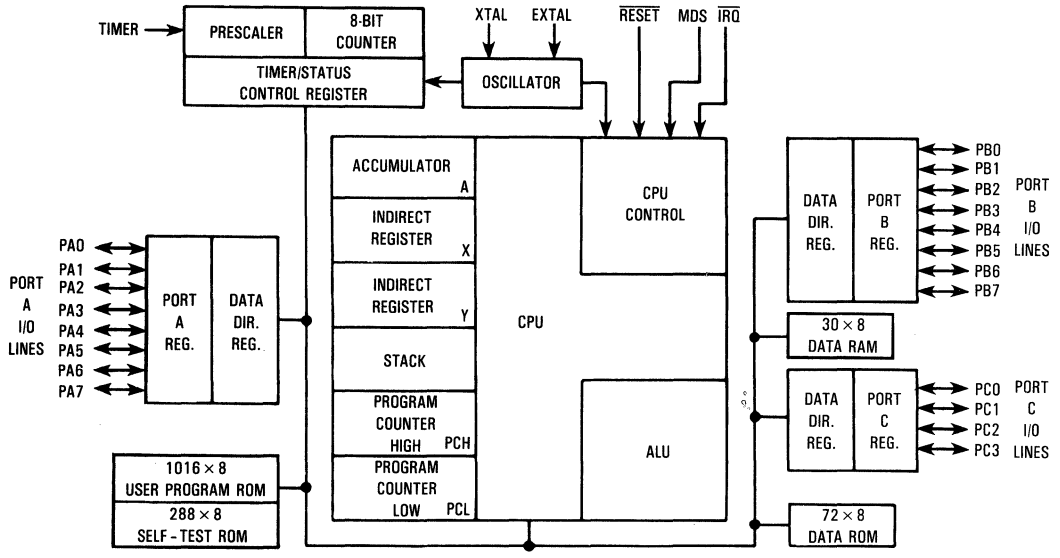


Figure 1-1. MC6804P2 Block Diagram

1.2 FEATURES

The following are some of the hardware and software features of the MC6804P2 MCU.

HARDWARE FEATURES

- 5-Volt Single Supply
- Pin Compatible with MC68HC04P2/P3
- 30 Bytes of RAM
- Memory Mapped I/O
- 1016 Bytes of Program ROM
- 72 Bytes of Data ROM
- X, Y Indirect Registers
- 20 Bidirectional I/O Lines (Eight Lines with High Current Sink Capability)
- On-Chip Clock Generator
- Self-Test Mode
- Master Reset
- Software Programmable 8-Bit Timer Control Register and Timer Prescaler (7 Bits, 2^n)
- Timer Pin Programmable as Input or Output
- On-Chip Circuit for ROM Verify

SOFTWARE FEATURES

- Similar to M6805 HMOS Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instruction
- Separate Flags for Interrupt and Normal Processing
- Versatile Indirect Registers
- Conditional Branches
- Single Instruction Memory Examine/Change
- True LIFO Stack Eliminates Stack Pointer
- Nine Powerful Addressing Modes
- Any Bit in Data Space Memory May be Tested
- Any Bit in Data Space Memory Capable of Being Written to May be Set or Cleared

USER SELECTABLE OPTIONS

- 20 Bidirectional I/O Lines with LSTTL, LSTTL/CMOS, or Open-Drain Interface
- Low Current Input Clamping Diodes Available on LSTTL and Open-Drain Interface Ports
- Crystal or Low-Cost Resistor-Capacitor Oscillator
- Mask Selectable Edge- or Level-Sensitive Interrupt Pin

SECTION 2
FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU, AND REGISTERS

This section provides a description of the functional pins, memory spaces, the central processing unit (CPU), and the various registers and flags.

2.1 FUNCTIONAL PIN DESCRIPTION

2.1.1 V_{CC} and V_{SS}

Power is supplied to the MCU using these two pins. V_{CC} is power and V_{SS} is the ground connection.

2.1.2 $\overline{IRQ}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to **4.1 INTERRUPT** for additional information pertaining to the interrupt operation.

2.1.3 XTAL and EXTAL

These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor and capacitor, or an external signal, depending on the user selectable manufacturing mask option, can be connected to these pins to provide a system clock source with various stability/cost tradeoffs. Lead lengths and stray capacitance on these two pins should be minimized. Refer to **4.4 INTERNAL CLOCK GENERATOR OPTIONS** for recommendations concerning these inputs.

2.1.4 TIMER

The TIMER pin may be configured to operate in the input or output mode of operation. In the input mode, the TIMER pin is connected to the prescaler input and serves as the timer clock. In the output mode, the TIMER pin will reflect the contents of the DOUT bit of the timer status and control register each time the TMZ bit has a low to high transition. Refer to **SECTION 3 TIMER** for additional information.

2.1.5 $\overline{RESET}$

The $\overline{RESET}$ pin is used to restart the processor of the MC6804P2 to the beginning of a program. The program counter will be loaded with the user restart vector ($FFE-$FFF) where the first instruction of the program is located. This should be a jump address to the first instruction of the main program (Refer to Figure 4-2).

This pin, together with the MDS pin, is also used to select the operating mode of the MC6804P2. If the MDS pin is at zero volts, the normal mode is selected and the program counter is loaded with the user restart vector. However, if the MDS pin is at +5 volts, then pins PA6 and PA7 are decoded to allow selection of the operating mode. Refer to **4.3 RESET** for additional information.

2.1.6 MDS

The MDS (mode select) pin is used to place the MCU into special operating modes. If MDS is held at +5 volts at the exit of the reset state, the decoded state of PA6 and PA7 is latched to determine the operating mode (single-chip, self-test, or ROM verify). However, if MDS is held at zero volts at the exit of the reset state, the single-chip operating mode is automatically selected (regardless of PA6 and PA7 state).

For those users familiar with the MC6801 microcomputer, mode selection is similar but much less complex in the MC6804P2. No special external diodes, switches, transistors, etc. are required in the selection of the desired mode.

2.1.7 Input/Output Lines (PA0-PA7, PB0-PB7, PC0-PC3)

These 20 lines are arranged into two 8-bit ports (A and B) and one 4-bit port (C). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to **SECTION 5 INPUT/OUTPUT PORTS** for additional information.

2.2 MEMORY

The MCU has two separate memory spaces: program space and data space. A representation of these memory spaces is shown in Figure 2-1. The program space contains all of the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, and the self-test and user vectors. The data space contains all of the RAM locations, plus I/O locations and some ROM used for storage of tables and constants. Only program space is addressed by the program counter, thus instructions may only be executed from the program space. Data space may be addressed by an instruction operand or an indirect register. No instructions can be executed out of the data space area.

In addition to the program and data memory spaces, a non-accessible subroutine stack space RAM (not shown in Figure 2-1) is provided. This stack space consists of a last-in-first-out (LIFO) register which is used with inherent addressing to stack the return address for subroutines.

The MCU is capable of addressing 4096 bytes of program space memory with its program counter and 256 bytes of data space memory with its instructions. The data space memory contains three bytes for port data registers, three bytes for port data direction registers, one byte for timer status/control, 72 bytes ROM, 30 bytes RAM, two bytes for X and Y indirect registers, two bytes for timer prescaler and count registers, and one byte for the accumulator. The program space section includes 288 bytes of self-test ROM, 1016 bytes program ROM, and eight bytes of vector address locations for self-test and user programs to store restart/$\overline{\text{IRQ}}$ jump addresses.

Indirect X and Y register locations $80 and $81 are normally used as pointers (e.g., indirect addressing to data space locations). The short direct addressing allows access to the four data space addresses $80-$83 with single byte opcodes. Operations allowed are increment, decrement, load, and store. Data space locations $82 and $83 can be used for 8-bit counter locations.

2.3 CENTRAL PROCESSING UNIT

The CPU of the M6804 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal serial addresses, data, and control buses.

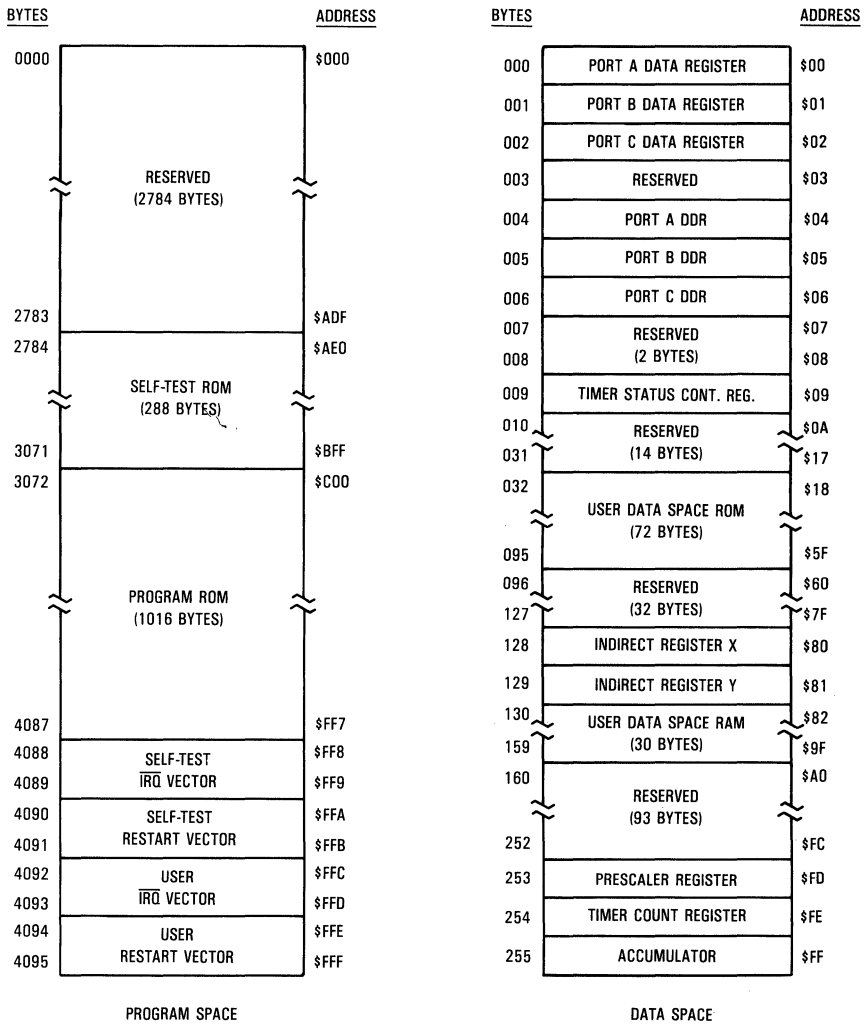


Figure 2-1. MC6804P2 Memory Map

2.4 REGISTERS

The M6804 Family CPU has four registers and two sets of flags available to the programmer. They are shown in Figure 2-2 and are explained in the following paragraphs.

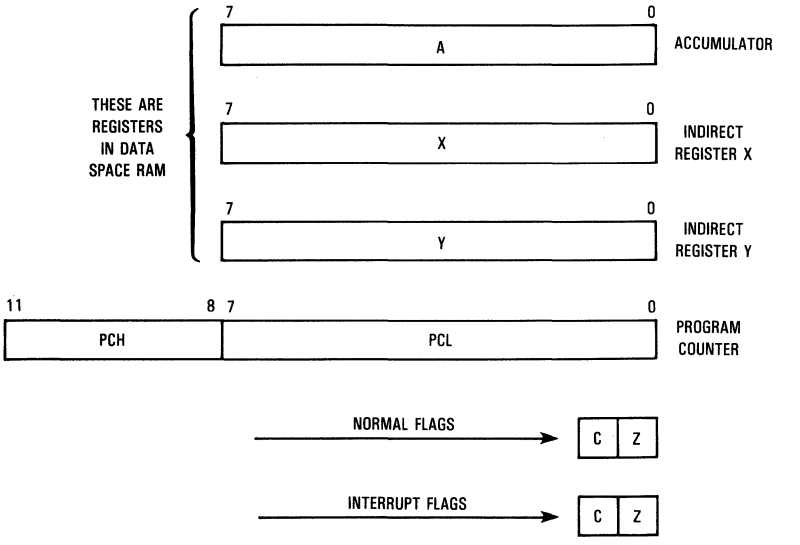


Figure 2-2. Programming Model

2.4.1 Accumulator (A)

The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator is implemented as the highest RAM location ($FF) in data space and thus implies that several instructions exist which are not explicitly implemented. Refer to **6.3 IMPLIED INSTRUCTIONS** for additional information.

2.4.2 Indirect Registers (X, Y)

These two indirect registers are used to maintain pointers to other memory locations in data space. They are used in the register-indirect addressing mode, and can be accessed with the direct, indirect, short direct, or bit set/clear addressing modes. These registers are implemented as data space locations ($80, $81) and may be manipulated in a manner similar to any RAM memory location in data space. Refer to **6.3 IMPLIED INSTRUCTIONS** for additional information.

2.4.3 Program Counter (PC)

The program counter is a 12-bit register that contains the address of the next program space ROM word to be used (may be opcode, operand, or address of operand). The 12-bit program counter is contained in PCL (low byte) and PCH (high nibble).

2.4.4 Flags (C, Z)

The carry (C) bit is set on a carry or a borrow out of the ALU. It is cleared if the result of an arithmetic operation does not result in a carry or a borrow. The (C) bit is also set to the value of the bit tested in a bit test instruction, and participates in the rotate left instruction.

The zero (Z) bit is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared. Bit test instructions do not affect the Z bit.

There are two sets of these flags, one set is for interrupt processing, the other for normal operations. When an interrupt occurs, a context switch is made from the program flags to the interrupt flags (interrupt mode). An RTI forces the context switch back to the program flags (program mode). While in either mode, only the flags for that mode are available. A context switch will not affect the value of the C or Z bits. Both sets of flags are cleared by reset.

2.4.5 Stack

A LIFO stack is incorporated in the MC6804P2 that eliminates the need for a stack pointer. This non-accessible subroutine stack space is implemented in separate RAM (12-bits wide). Whenever a subroutine call (or interrupt) occurs, the contents of the PC are shifted into the top register of the stack. At the same time (same cycle), the top register is shifted one level deeper. This happens to all registers with the bottom register falling out the bottom of the stack.

Whenever a subroutine or interrupt return occurs, the top register is shifted into the PC and all lower registers are shifted one level higher. The stack RAM is four levels deep. If the stack is pulled more than four times without any pushes, the address that was stored in the bottom level will be shifted into the PC.

SECTION 3
TIMER

3.1 INTRODUCTION

A block diagram of the MC6804P2 timer circuitry is shown in Figure 3-1. The timer logic in the MCU is comprised of a simple 8-bit counter (timer count register, TCR) with a 7-bit prescaler, and a timer status/control register (TSCR). The timer count register, which may be read or written to under program control, is decremented towards zero by the prescaler output. The prescaler is used to extend the maximum interval of the overall timer. The prescaler tap is selected by bits 0-2 (PS0-PS2) of the timer status/control register. Bits PS0-PS2 control the actual division of the prescaler within the range of divide-by-1 (2^0) to divide-by-128 (2^7). The timer count register (TCR) is decremented on each rising edge output of the prescaler. The prescaler is decremented on each rising edge on the TIMER pin (input mode only). The coding of the TCSR PS0-PS2 bits produce a division in the prescaler as shown in Table 3-1.

Table 3-1. Prescaler Coding Table

PS2	PS1	PS0	Divide By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8

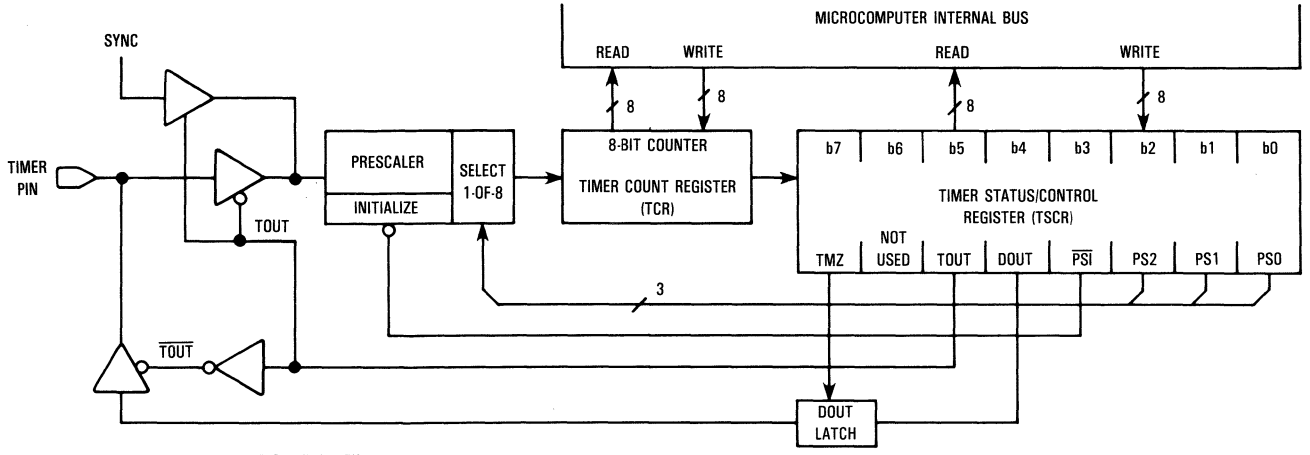
PS2	PS1	PS0	Divide By
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

The TIMER pin may be programmed as either an input or an output depending on the status of TOUT (TSCR bit 5). Refer to Figure 3-1. In the input mode, TOUT is a logic zero and the TIMER pin is connected directly to the prescaler input. Therefore, the timer prescaler is clocked by the signal applied from the TIMER pin. The prescaler then divides its clock input by a value determined by the coding of the TSCR bits PS0-PS2 as shown in Table 3-1. The divided prescaler output then clocks the 8-bit timer count register (TCR). When the TCR is decremented to zero, it sets the TMZ bit in the timer status/control register (TSCR). The TMZ bit can be tested under program control to perform a timer function whenever it goes high. The frequency of the external clock applied to the TIMER pin must be less than t_{byte} ($f_{osc}/48$).

In the output mode, TOUT is a logic one and the TIMER pin is connected to the DOUT latch. Therefore, the timer prescaler is clocked by the internal sync pulse (divide-by-48 of the internal oscillator). Operation is similar to that described above for the input mode. However, in the output mode, the low-to-high TMZ bit transition is used to clock the DOUT bit of the TSCR into a latch which drives the TIMER pin.

NOTE

TMZ is normally set to logic one when the timer times out (TCR decrements to $00); however, it may be set by a write of $00 to the TCR or by a write to bit 7 of the TSCR.



TIMER PIN STATUS

TOUT	PRESCALER CLOCK	TIMER PIN
0	TIMER PIN	INPUT MODE
1	SYNC	OUTPUT MODE

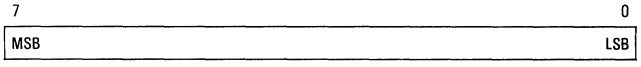
Figure 3-1. Timer Block Diagram

When using a read-modify-write instruction on any bit location in the TSCR, the DOUT bit can be clocked immediately since a set TMZ bit can cause an extra low-to-high transition to occur. Modifying the TSCR register with the MVI instruction is recommended.

During reset, the timer count register and prescaler are set to $FF, while the timer status/control register is cleared to $00 and the DOUT LATCH (TIMER pin is in the high-impedance input mode) is forced to a logic high. The prescaler and timer count register are implemented in data space RAM locations ($FD, $FE); therefore, they are both readable and writeable. A write to either will predominate over the TCR decrement-to-$00 function; i.e., if a write and a TCR decrement-to-$00 occur simultaneously, the write will take precedence, and the TMZ bit is not set until the next timer time out.

3.2 TIMER REGISTERS

3.2.1 Timer Count Register (TCR)



TCR ADDRESS = $FE

The timer count register indicates the state of the internal 8-bit counter.

3.2.2 Timer Status/Control Register (TSCR)

7	6	5	4	3	2	1	0
TMZ	NOT USED	TOUT	DOUT	$\overline{PSI}$	PS2	PS1	PS0

TSCR ADDRESS = $09

b7, TMZ — Low-to-high transition indicates the timer count register has decremented to zero since the timer status/control register was last read. Cleared by a read of TSCR register if TMZ was read as a logic one.

A read-modify-write instruction to the TSCR will inadvertently read the TMZ bit. Therefore, if the TMZ bit is set, the TMZ bit will be cleared and then set again to the state prior to the read-modify-write instruction.

b6 — Not used.

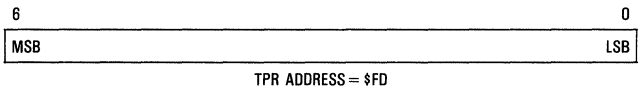
b5, TOUT — When low, this bit selects the input mode for the timer. When high, the output mode is selected.

b4, DOUT — Data sent to the timer output pin when TMZ completes a low-to-high transition.

b3, $\overline{PSI}$ — Used to initialize the prescaler and inhibit its counting while $\overline{PSI} = 0$. The initialized value is set to $FF. The timer count register will also be inhibited (contents unchanged). When $\overline{PSI} = 1$ the prescaler begins to count downward.

b0, b1, b2 PS0-PS1 — These bits are used to select the prescaler divide-by ratio; therefore, effecting the clock input frequency to the timer count register.

3.2.3 Timer Prescaler Register



The timer prescaler register indicates the state of the internal 7-bit prescaler. This 7-bit prescaler divide ratio is normally determined by bits PS0-PS2 of the timer status/control register (see Table 3-1).

SECTION 4
INTERRUPT, SELF-TEST, RESET, AND INTERNAL CLOCK GENERATOR

4.1 INTERRUPT

The MC6804P2 can be interrupted by applying a logic low signal to the $\overline{IRQ}$ pin; however, a mask option selected at the time of manufacture determines whether the negative-going edge or the actual low level is sensed to indicate an interrupt.

4.1.1 Edge-Sensitive Option

When the $\overline{IRQ}$ pin is pulled low, the internal interrupt request latch is set. Prior to each instruction fetch, the interrupt request latch is tested and, if its output is low, an interrupt sequence is initiated at the end of the current instruction (provided the interrupt mask is cleared). Figure 4-1 contains a flowchart which illustrates both the reset and interrupt sequence. The interrupt sequence consists of one cycle during which: the interrupt request latch is cleared, the interrupt mode flags are selected, the PC is saved on the stack, the interrupt mask is set, and the $\overline{IRQ}$ vector jump address (single chip mode = $FFC/$FFD, self-test mode = $FF8/$FF9) is loaded into the PC. Note that the contents of these locations are not inherently decoded as an address to which program control (PC) should jump to, but instead are decoded like any other ROM word. It is therefore essential that the vector contents specify a JMP instruction in addition to the starting address of the interrupt service routine. This routine should save the values of the accumulator, and X and Y registers, if required, since the values will not be stored on the stack. Internal processing of the interrupt continues until an RTI (return from interrupt) instruction is processed. During the RTI instruction, the interrupt mask is cleared and the program mode flags are selected.The next instruction of the program is then fetched and executed. Once the interrupt was initially detected and the interrupt sequence started, the interrupt request latch is cleared so that the next (second) interrupt may be detected even while the previous (first) one is being serviced. However, even though the second interrupt sets the interrupt request latch during processing of the first interrupt, the second interrupt sequence will not be initiated until completion of the interrupt service routine for the first interrupt. Completion of an interrupt service routine is always accomplished using an RTI instruction to return to the main program. The interrupt mask (which is not directly available to the programmer) is cleared during the last cycle of the RTI instruction.

4.1.2 Level-Sensitive Option

The actual operation of the level-sensitive and edge-sensitive options are similar except that the level-sensitive option does not have an interrupt request latch. With no interrupt request latch, the logic level of the $\overline{IRQ}$ pin is checked for detection of the interrupt. Also, in the interrupt sequence, there is no need to clear the interrupt request latch. These differences are illustrated in the flowchart of Figure 4-1.

4.1.3 Power Up and Timing

During the power-up sequence the interrupt mask is set to preclude any false or "ghost" interrupts from occurring, and the PC is loaded with the appropriate restart vector. As with the $\overline{IRQ}$ vector (jump address), this must contain a JMP instruction in addition to a ROM address, which in this case specifies the start

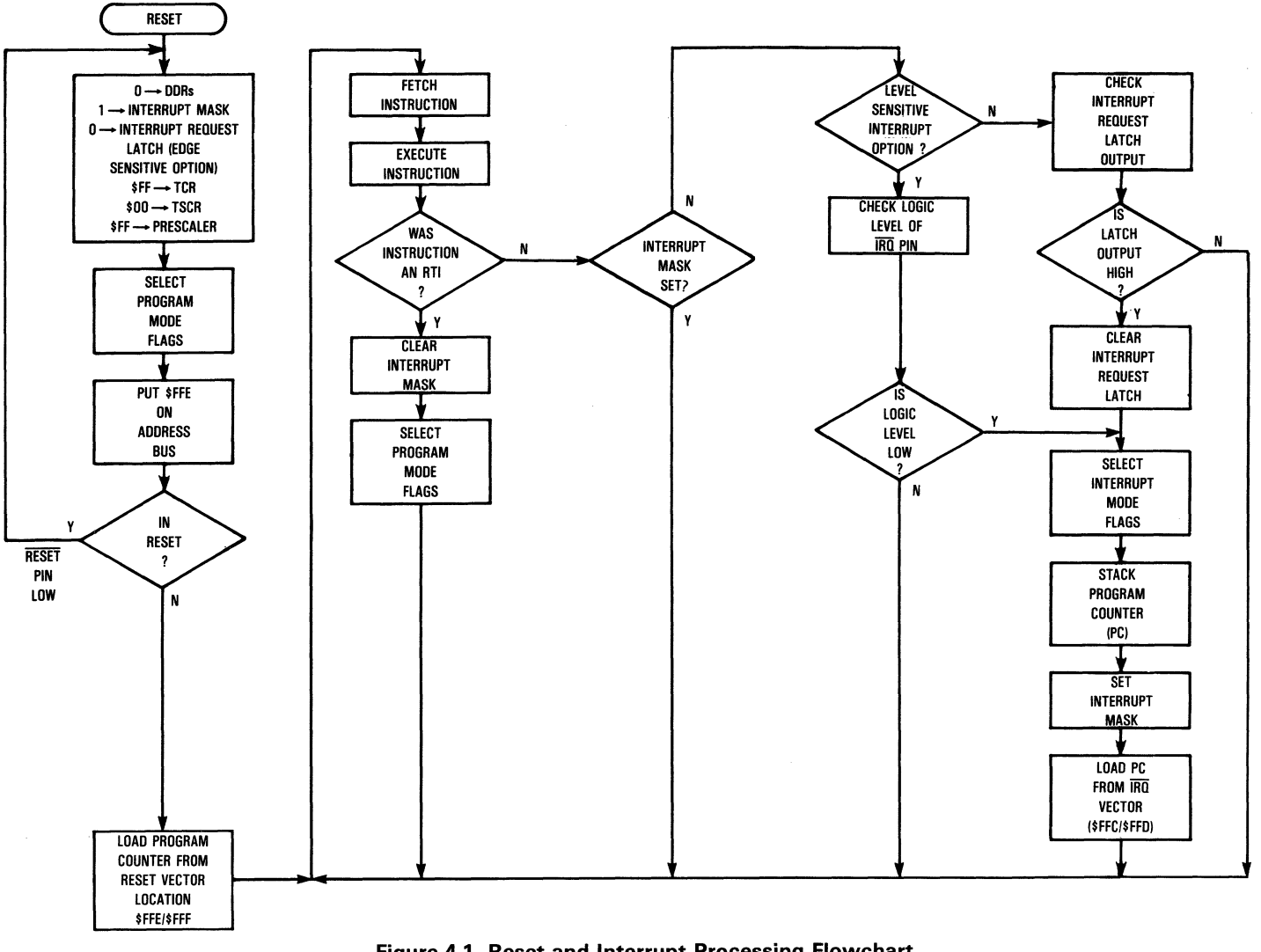


Figure 4-1. Reset and Interrupt Processing Flowchart

of the initialization routine. To clear the interrupt mask, the initialization routine should end with an RTI (instead of RTS), which means the initialization routine should start with a JSR (instead of JMP) in order to push a suitable PC value onto the stack. This is illustrated in Figure 4-2. Maximum interrupt response time is six machine (t_{byte}) cycles (see **4.4 INTERNAL CLOCK GENERATOR OPTIONS**). This includes five machine cycles for the longest instruction, plus one machine cycle for stacking the PC and switching flags. Minimum response time is one machine cycle for stacking PC and switching flags (see **2.4.4 Flags (C, Z)**).

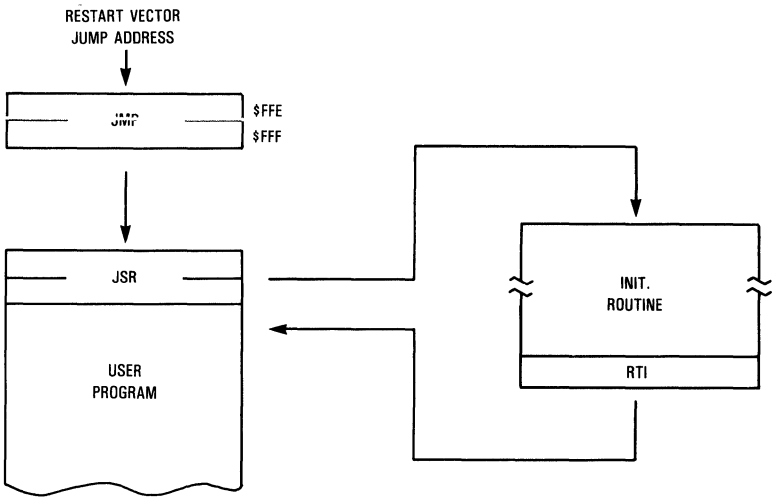


Figure 4-2. Typical Program Restart

4.1.4 TIMER INTERRUPT

The MC6804P2 MCU can only be interrupted by applying a logic low signal to the $\overline{IRQ}$ pin. However, a timer interrupt is possible, provided that the timer is operating in the output mode and the TIMER pin is connected to the $\overline{IRQ}$ pin as shown in Figure 4-3. During the interrupt routine, it is necessary to test the state of the TMZ bit in the TSCR in order to determine whether an interrupt was caused externally or by the timer. If it was due to the timer, the routine should next set the timer output to a logic one in preparation for the next interrupt, and reload the TCR.

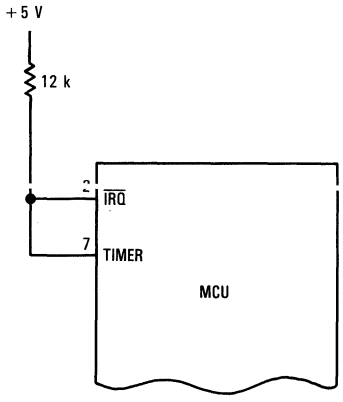


Figure 4-3. Timer Interrupt Connection

4.2 SELF-TEST

The MC6804P2 MCU has a unique internal ROM based off-line self-test capability using signature analysis techniques. A test program stored in the on-chip ROM is initiated by configuring PA6 and PA7 during reset. The test results are sampled on a cycle-by-cycle basis by a 16-bit on-chip signature analysis register configured as a linear feedback shift register (LSFR) using the International Consultative Committee for Telephone and Telegraph (CCITT) standard cycle redundancy check (CRC) 16 polynomial. The self-test connections are illustrated in Figure 4-4A. To perform a test of the MCU, connect the MCU as shown and monitor the LEDs for a 00100 ($04) pattern.

A special ROM self-test utilizing the signature analysis circuitry is also included. To initiate a test of the ROM, connect the circuit as shown in Figure 4-4B. This mode also uses the on-chip signature analysis register to verify the contents of the custom ROM by monitoring an internal bus. The "GOOD" LED indicates that all ROM words have been read and that the result was the correct signature.

The on-chip self-test and the ROM test are the basis of Motorola's production testing for the MC6804P2. These tests have been fault graded using statistical methods (refer to "The M6804 Built-In Self-Test", Proceedings of 1983 International Test Conference, pp. 295-300, Oct. 1983) and have been found to provide high fault coverage using automatic test equipment (ATE) or the circuitry shown in Figure 4-4.

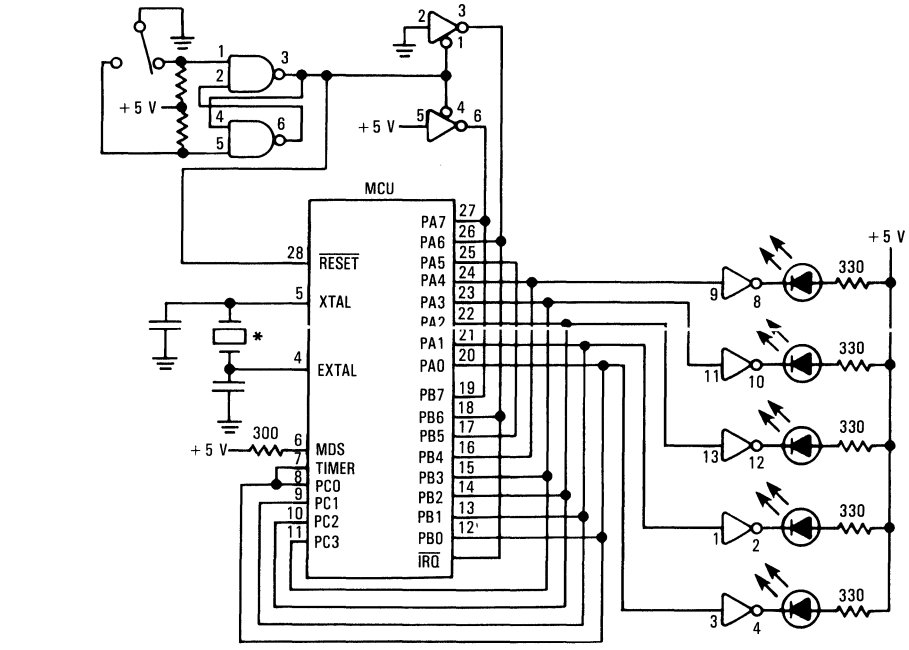
4.3 RESET

The MCU can be reset in two ways: by initial power up (see Figure 4-1) and by the external reset input ($\overline{\text{RESET}}$). During power up, a delay of t_{RHL} is needed before allowing the $\overline{\text{RESET}}$ input to go high. This time delay allows the internal clock generator to stabilize. Connecting a capacitor and resistor to the $\overline{\text{RESET}}$ input, as shown in Figure 4-5, typically provides sufficient delay.

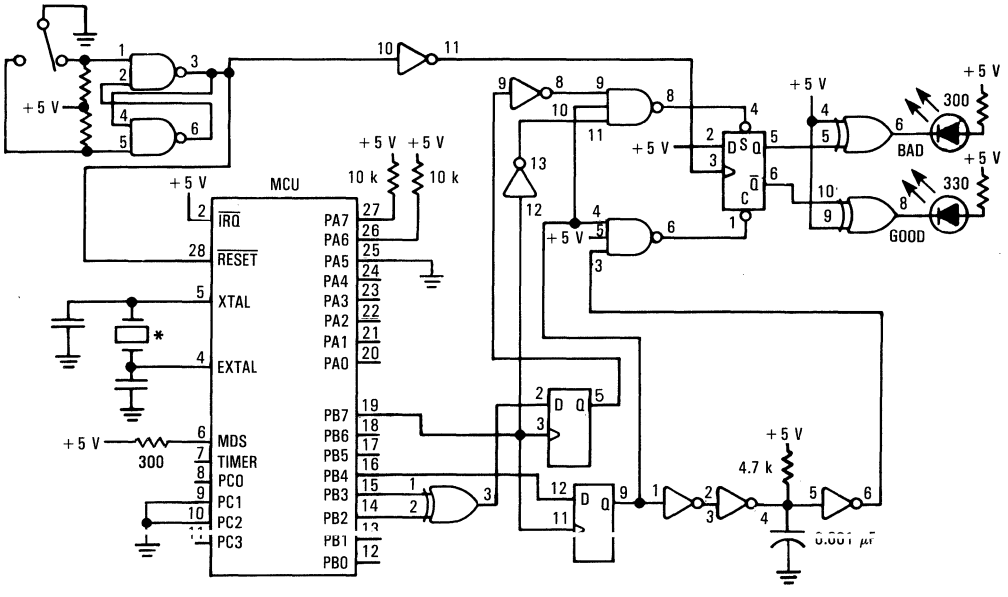
4.4 INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor-capacitor, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. The different clock generator option connection methods are shown in Figure 4-6, crystal specifications and suggested PC board layouts are given in Figure 4-7, resistor-capacitor selection graph is given in Figure 4-8, and a timing diagram is illustrated in Figure 4-9. The crystal oscillator startup time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitance (C_L), IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations.

The oscillator output frequency is internally divided by four to produce the internal $\phi 1$ and $\phi 2$ clocks. The $\phi 1$ clock is divided by twelve to produce a machine byte (cycle) clock. A byte cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five byte cycles to execute.



(a) FUNCTIONAL TEST



(b) SIMPLE ROM VERIFY CHECK

NOTES:
1. *used with crystal option only.
2. 10 k pullup resistors to +5 V are required on all connected I/O port lines for open-drain option.

Figure 4-4. Self-Test Circuit

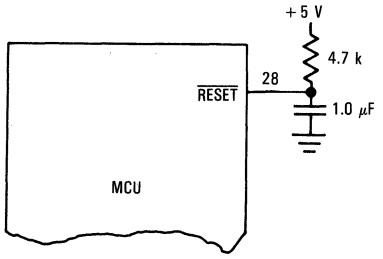


Figure 4-5. Power-Up Reset Delay Circuit

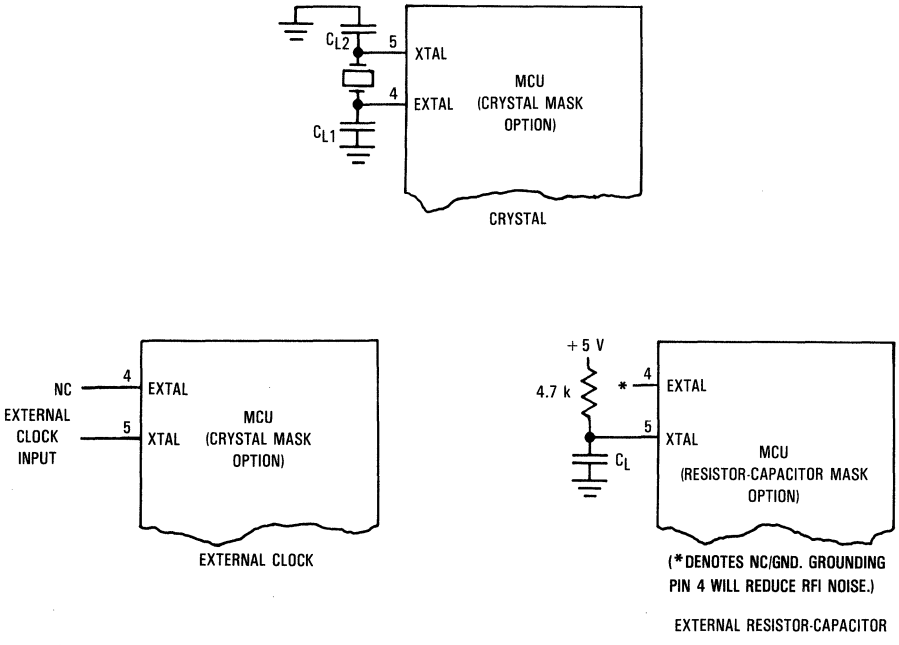
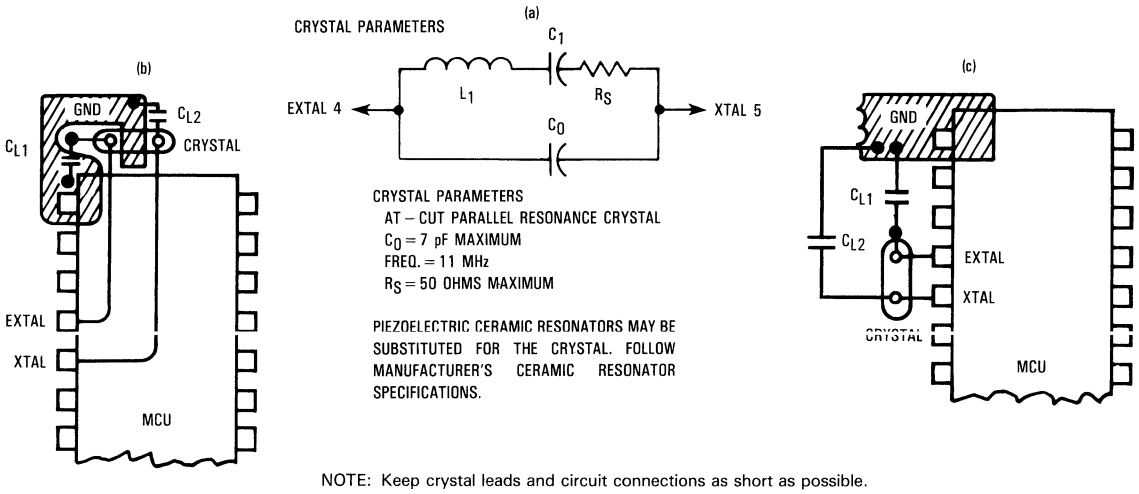
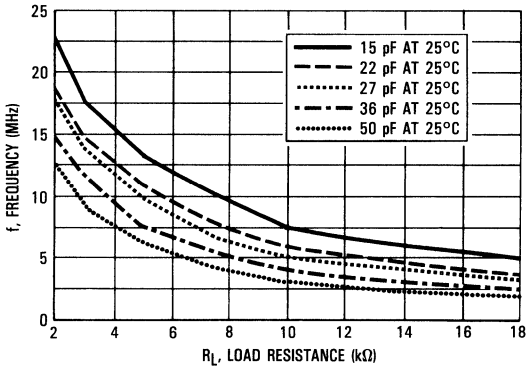


Figure 4-6. Clock Generator Options

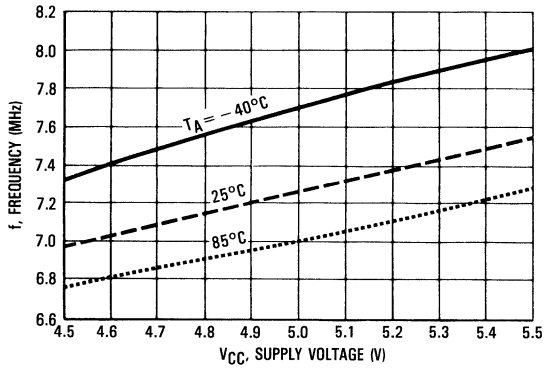


NOTE: Keep crystal leads and circuit connections as short as possible.

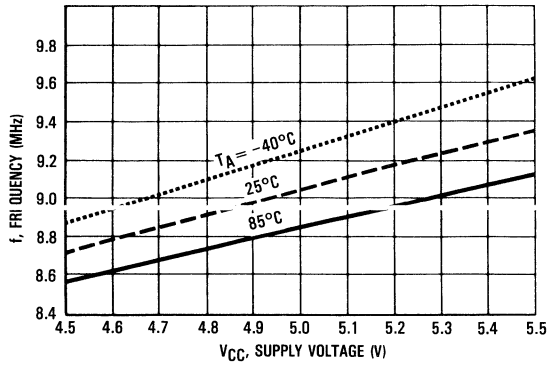
Figure 4-7. Crystal/Ceramic Resonator Motional Arm Parameters and Suggested PC Board Layout



(a) TYPICAL FREQUENCY VS RESISTANCE



(b) TYPICAL FREQUENCY VARIATIONS @ C_L = 15 pF, 10 kΩ



(c) TYPICAL FREQUENCY VARIATIONS @ C_L = 50 pF, 3 kΩ

Figure 4-8. Typical Frequency Selection for Resistor-Capacitor Oscillator Options

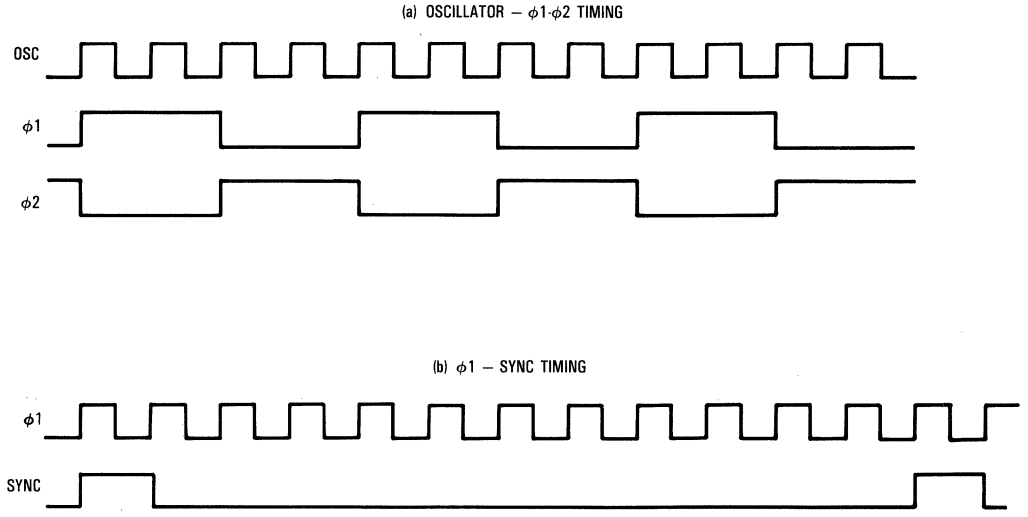
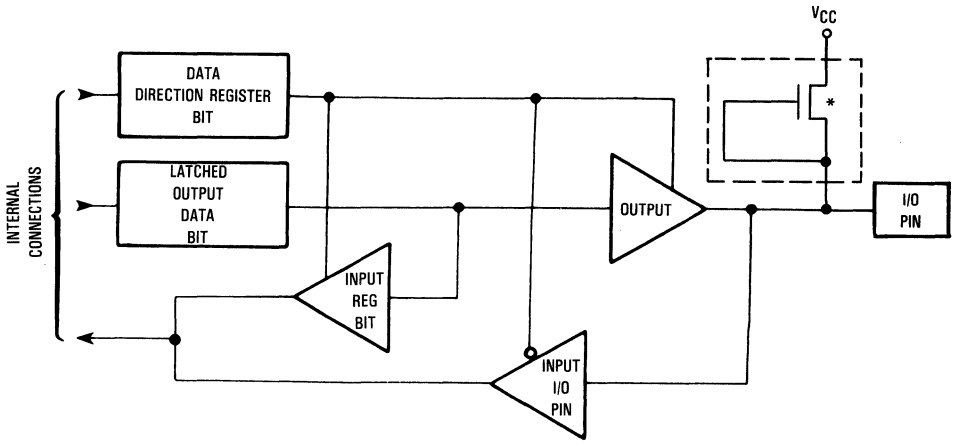


Figure 4-9. Clock Generator Timing Diagram

SECTION 5
INPUT/OUTPUT PORTS

5.1 INPUT/OUTPUT

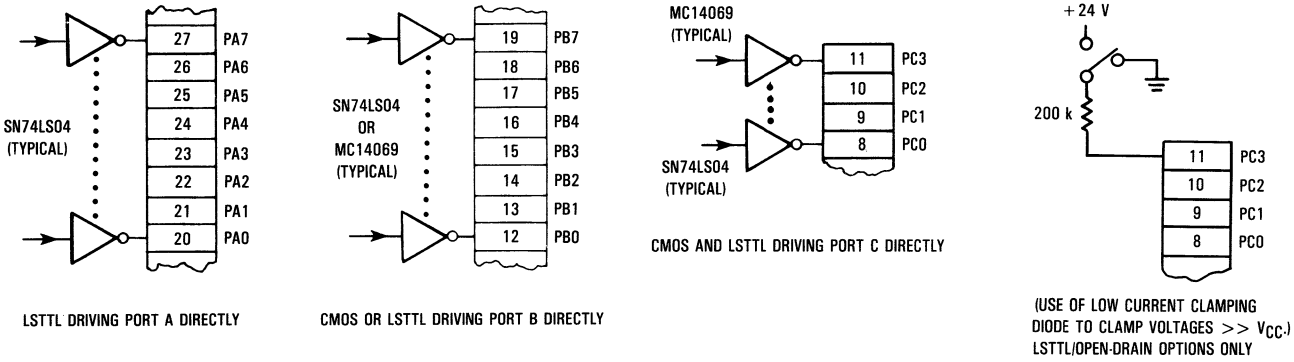
There are 20 input/output pins. All pins (port A, B, and C) are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output or a logic zero for input. On reset, all the DDRs are initialized to a logic zero state to put the ports in the input mode. The port output registers are not initialized on reset but should be initialized before changing the DDR bits to avoid undefined levels. When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading; see Figure 5-1. All input/output pins are LSTTL compatible as both inputs and outputs. In addition, all three ports may have one of two mask options: 1) internal pullup resistor for CMOS output compatibility, or 2) open drain output. The address map in Figure 2-1 gives the address of data registers and DDRs. The register configuration is discussed under the registers paragraph below and Figure 5-2 provides some examples of port connections.



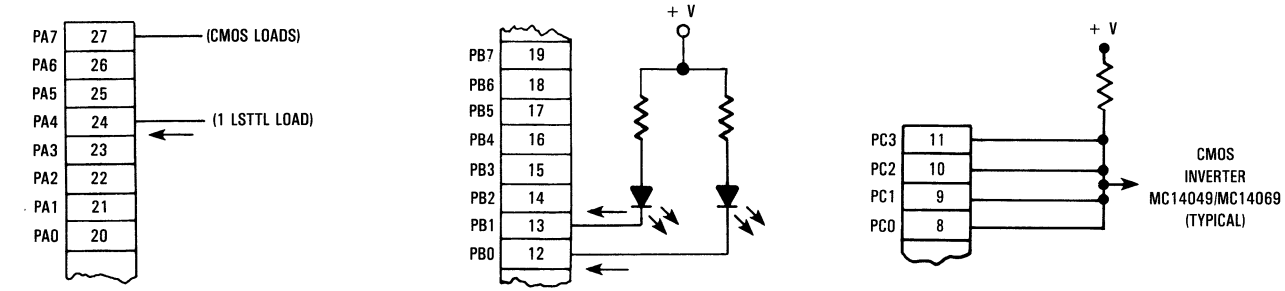
DATA DIRECTION REGISTER BIT	OUTPUT DATA BIT	OUTPUT STATE	INPUT TO MCU
1	0	0	0
1	1	1	1
0	X	HI-Z	PIN

*For CMOS option transistor acts as resistor (approximately 40 kΩ) to V_{CC}.
For LSTTL/open-drain options transistor acts as low current clamping diode to V_{CC}.

Figure 5-1. Typical I/O Port Circuitry



(a) INPUT MODE



(b) OUTPUT MODE

Figure 5-2. Typical Port Connections

The latched output data bit (see Figure 5-1) may always be written. Therefore, any write to a port writes to all of its data bits even though the port DDR is set to input. This may be used to initialize the data registers and avoid undefined outputs; however, care must be exercised when using read- modify-write instructions since the data read corresponds to the pin level if the DDR is an input (0) and corresponds to the latched output data when the DDR is an output (1). The 20 bidirectional lines may be configured by port to be LSTTL (standard configuration), LSTTL/CMOS (mask option), or open drain (mask option). Port B outputs are LED compatible.

NOTE

The mask option only allows changes by port. For example, if the customer wishes PA7 to be open drain, then PA0-PA7 must all be open drain.

5.2 REGISTERS

The registers described below are implemented as RAM locations and thus may be read or written.

5.2.1 Port Data Register

Port A Data Register ($00)

	7	6	5	4	3	2	1	0	
MSB									LSB

Port B Data Register ($01)

	7	6	5	4	3	2	1	0	
MSB									LSB

Port C Data Register ($02)

	7	6	5	4	3	2	1	0	
MSB	1	1	1	1					LSB

NOTE: Four MSB bits are don't care (x) bits when written to. These bits are unused and returned as a logical 1 when read.

The source of data read from the port data register will be the port I/O pin or previously latched output data depending upon the contents of the corresponding data direction register (DDR). The destination of data written to the port data register will be an output data latch. If the corresponding data direction register (DDR) for the port I/O pin is programmed as an output, the data will then appear on the port pin.

5.2.2 Port Data Direction Register

Port A Data Direction Register ($04)

	7	6	5	4	3	2	1	0	
MSB									LSB

Port B Data Direction Register ($05)

	7	6	5	4	3	2	1	0	
MSB									LSB

Port C Data Direction Register ($06)

	7	6	5	4	3	2	1	0	
MSB	X	X	X	X					LSB

NOTE: Four MSB bits are don't care (X) bits.

5

The port DDRs configure the port pins as either inputs or outputs. Each port pin can be programmed individually to act as an input or an output. A zero in the pins corresponding bit position will program that pin as an input while a one in the pins corresponding bit position will program that pin as an output.

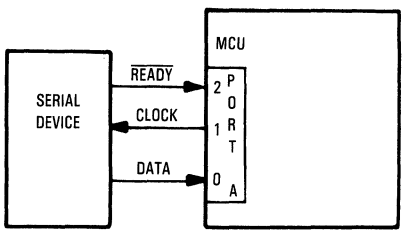
SECTION 6
SOFTWARE AND INSTRUCTION SET

6.1 SOFTWARE

6.1.1 Bit Manipulation

The MC6804P2 MCU has the ability to set or clear any modifiable bit in any register or memory location with a single instruction (BSET, BCLR). Any bit in data space, including ROM, can be tested, using the BRSET and BRCLR instructions, and the program may branch as a result of its state. The carry bit equals the value of the bit referenced by BRSET or BRCLR. A rotate instruction may then be used to accumulate serial input data in a RAM location or register. The capability to work with any bit in data space allowing the user to have individual flags in RAM or to handle I/O bits as control lines.

The coding example in Figure 6-1 illustrates the usefulness of the bit manipulation and test instructions. Assume that the MCU is to communicate with an external serial device. The external device has a data ready signal, a data output line, and a clock line (to clock data one bit at a time, MSB first, out of the device). The MCU waits until the data is ready, clocks the external device, picks up the data in the carry flag (C bit), clears the clock line, and finally accumulates the data bit in the accumulator.




```
                .
                .
                .
SELF    BRSET   2,PORTA,SELF
                .
                .
                .
        BSET    1,PORTA
      BRCLR     0,PORTA,CONT
CONT    BCLR    1,PORTA
        ROLA    .
                .
                .
                .
```

Figure 6-1. Bit Manipulation Example

6.1.2 Addressing Modes

The MC6804P2 MCU has nine addressing modes which are explained briefly in the following paragraphs. The MC6804P2 deals with objects in three different address spaces: program space, data space, and stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains all of the RAM locations, X and Y registers, accumulator, timer, I/O locations, and some ROM (for storage of tables and constants). Stack space contains RAM for use in stacking the return addresses for subroutines and interrupts.

The term "Effective Address" (EA) is used in describing the address modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

6.1.2.1 IMMEDIATE. In the immediate addressing mode, the operand is located in program ROM and is contained in a byte following the opcode. The immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Examples: LDA #$03
ADD #$00

6.1.2.2 DIRECT. In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the 256 bytes in data space memory with a single two-byte instruction.

Examples: LDA $50
SUB $70

6.1.2.3 SHORT DIRECT. The MCU also has two locations in data space RAM ($82, $83), and two indirect registers X and Y which may be used in a short-direct addressing mode. In this mode the opcode determines the data space RAM location, and the instruction is only one byte. Short direct addressing is a subset of the direct addressing mode. (The X and Y registers are at locations $80 and $81 respectively.)

Example: STA $80

6.1.2.4 EXTENDED. In the extended addressing mode, the effective address is obtained by concatenating the four least significant bits of the opcode with the byte following the opcode (12-bit address). Instructions using the extended addressing mode (JMP, JSR) are capable of branching anywhere in program space. An extended addressing mode instruction is two bytes long.

Example: JMP $00

6.1.2.5 RELATIVE. The relative addressing mode is only used in conditional branch instructions. In relative addressing, that address is formed by adding the sign extended lower five bits of the opcode (the offset) to the program counter if and only if the condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from -15 to $+16$ from the opcode address. The programmer need not worry about calculating the correct offset when using the Motorola assembler since it calculates the proper offset and checks to see if it is within the span of the branch.

6.1.2.6 BIT SET/CLEAR. In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified

bit is to be set or cleared. Thus, any bit in the 256 locations of data space memory, which can be written to, can be set or cleared.

```
Examples: BSET 7,$30
          BCLR 6,$25
```

6.1.2.7 BIT TEST AND BRANCH. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit which is to be tested is included in the opcode, and the data space address of the byte to be tested is in the single byte immediately following the opcode byte. The third byte is sign extended to twelve bits and becomes the offset added to the program counter if the condition is true. The single three-byte instruction allows the program to branch based on the condition of any bit in data space memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry flag.

```
Examples: COO BRSET 3,$20,$C05
          BRCLR 4,$30,$C12
```

6.1.2.8 REGISTER-INDIRECT. In the register-indirect addressing mode, the operand is at the address (in data space) pointed to by the contents of one of the indirect registers (X or Y). The particular X or Y register is selected by bit 4 of the opcode. Bit 4 of the opcode is then decoded into an address which selects the desired X or Y register ($80 or $81). A register-indirect instruction is one byte long.

```
Examples: STA [X]
          LDA [Y]
```

6.1.2.9 INHERENT. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

```
Example: COMA
```

6.2 INSTRUCTION SET

The MC6804P2 MCU has a set of 42 basic instructions, which when combined with nine addressing modes produce 242 usable opcodes. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

6.2.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is the accumulator and the other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operands. Refer to Table 6-1.

6.2.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. There are ten instructions which utilize read- modify-write cycles. All INC and DEC forms along with all bit manipulation instructions use this method. Refer to Table 6-2.

Table 6-1. Register/Memory Instructions

		Addressing Modes																			
		Indirect				Immediate			Direct			Inherent			Extended			Short-Direct			
Function	Mnem	Opcode X	Opcode Y	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Special Notes
Load A from Memory	LDA	E0	F0	1	4	E8	2	4	F8	2	4	—	—	—	—	—	—	AC-AF	1	4	1
Load X from Memory	LDXI	—	—	—	—	B0	3	4	—	—	—	—	—	—	—	—	—	—	—	—	4
Load Y from Memory	LDYI	—	—	—	—	B0	3	4	—	—	—	—	—	—	—	—	—	—	—	—	4
Store A in Memory	STA	E1	F1	1	4	—	—	—	F9	2	4	—	—	—	—	—	—	BC-BF	1	4	2
Add to A	ADD	E2	F2	1	4	EA	2	4	FA	2	4	—	—	—	—	—	—	—	—	—	—
Subtract from A	SUB	E3	F3	1	4	EB	2	4	FB	2	4	—	—	—	—	—	—	—	—	—	—
Arithmetic Compare with Memory	CMP	E4	F4	1	4	EC	2	4	FC	2	4	—	—	—	—	—	—	—	—	—	—
AND Memory to A	AND	E5	F5	1	4	ED	2	4	FD	2	4	—	—	—	—	—	—	—	—	—	—
Jump to Subroutine	JSR	—	—	—	—	—	—	—	—	—	—	—	—	—	8 X	2	4	—	—	—	3
Jump Unconditional	JMP	—	—	—	—	—	—	—	—	—	—	—	—	—	9 (TAR)	2	4	—	—	—	3
Clear A	CLRA	—	—	—	—	—	—	—	FB	2	4	—	—	—	—	—	—	—	—	—	—
Clear X	CLRX	—	—	—	—	—	—	—	B0	3	4	—	—	—	—	—	—	—	—	—	—
Clear Y	CLRY	—	—	—	—	—	—	—	B0	3	4	—			—	—	—	—	—	—	—
Complement A	COMA	—	—	—	—	—	—	—	—	—	—	B4	1	4	—	—	—	—	—	—	—
Move Immediate Value to Memory	MVI	—	—	—	—	B0	3	4	B0	3	4	—	—	—	—	—	—	—	—	—	5
Rotate A Left and Carry	ROLA	—	—	—	—	—	—	—	—	—	—	B5	1	4	—	—	—	—	—	—	—
Arithmetic Left Shift of A	ASLA	—	—	—	—	—	—	—	FA	2	4	—	—	—	—	—	—	—	—	—	—

SPECIAL NOTES

1. In Short-Direct addressing, the LDA mnemonic represents opcode AC, AD, AE, and AF. This is equivalent to RAM locations $80 (AC), $81 (AD), $82 (AE), and $83 (AF).
2. In Short-Direct addressing, the STA mnemonic represents opcode BC, BD, BE, and BF. This is equivalent to RAM locations $80 (BC), $81 (BD), $82 (BE), and $83 (BF).
3. In Extended addressing, the four LSBs of the opcode (Mnemonic JSR and JMP) are formed by the four MSBs of the target address. The byte that follows the opcode of the JMP or JSR mnemonic will be the lower 8 bits of the address (e.g., JMP $C00 = 9C 00).
4. In Immediate addressing, the LDXI and LDYI are mnemonics which are recognized as follows:
   LDXI = MVI $80,data
   LDYI = MVI $81,data Where data is a one-byte hexadecimal number.
5. In both Immediate and Direct addressing, the MVI instruction has the same opcode (80).

Table 6-2. Read-Modify-Write Instructions

Function	Mnem	Addressing Modes: Indirect Opcode X	Indirect Opcode Y	Indirect # Bytes	Indirect # Cycles	Direct Opcode	Direct # Bytes	Direct # Cycles	Short-Direct Opcode	Short-Direct # Bytes	Short-Direct # Cycles	Special Notes
Increment Memory Location	INC	E6	F6	1	4	FE	2	4	A8-AB	1	4	1, 3
Increment A	INCA	—	—	—	—	FE	2	4	—	—	—	—
Increment X	INCX	—	—	—	—	—	—	—	A8	1	4	—
Increment Y	INCY	—	—	—	—	—	—	—	A9	1	4	—
Decrement Memory Location	DEC	E7	F7	1	4	FF	2	4	B8-BB	1	4	2, 4
Decrement A	DECA	—	—	—	—	FF	2	4	—	—	—	—
Decrement X	DECX	—	—	—	—	—	—	—	B8	1	4	—
Decrement Y	DECY	—	—	—	—	—	—	—	B9	1	4	—

SPECIAL NOTES

1. In Short-Direct addressing, the INC mnemonic represents opcode A8, A9, AA, and AB. These are equivalent to RAM locations $80 (A8), $81 (A9), $82 (AA), and $83 (AB).
2. In Short-Direct addressing, the DEC mnemonic represents opcode B8, B9, BA, and BB. These are equivalent to RAM locations $80 (B8), $81 (B9), $82 (BA), and $83 (BB).
3. In Indirect addressing, the INC mnemonic represents opcode E6 or F6, and causes the location pointed to by X (E6 opcode) or Y (F6 opcode) to be incremented.
4. In Indirect addressing, the INC mnemonic represents opcode E7 or F7, and causes the location pointed to by X (E7 opcode) or Y (F7 opcode) to be incremented.

6.2.3 Branch Instructions

The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 6-3.

6.2.4 Bit Manipulation Instructions

These instructions are used on any bit in data space memory. One group either sets or clears. The other group performs the bit test branch operations. Refer to Table 6-4.

6.2.5 Control Instructions

The control instructions control the MCU operations during program execution. Refer to Table 6-5.

6.2.6 Alphabetical Listing

The complete instruction set is given in alphabetical order in Table 6-6. There are certain mnemonics recognized by the Motorola assembler and converted to other instructions. The fact that all registers and accumulator are in RAM allows many implied instructions to exist. The implied instructions recognized by the Motorola assembler are identified in Table 6-6.

6.2.7 Opcode Map Summary

Table 6-7 contains an opcode map for the instructions used on the MCU.

6.3 IMPLIED INSTRUCTIONS

Since the accumulator and all other registers are located in RAM many implied instructions exist. The assembler-recognized implied instructions are given in Table 6-6. Some examples not recognized by the assembler are shown below.

BCLR,7 $FF	Ensures accumulator is plus
BSET,7 $FF	Ensures accumulator is minus
BRCLR,7 $FF	Branch if accumulator is plus
BRSET,7 $FF	Branch if accumulator is minus
BRCLR,7 $80	Branch if X is plus (BXPL)
BRSET,7 $80	Branch if X is minus (BXMI)
BRCLR,7 $81	Branch if Y is plus (BYPL)
BRSET,7 $81	Branch if Y is minus (BYMI)

Table 6-3. Branch Instructions

		Relative Addressing Mode			
Function	Mnem	Opcode	# Bytes	# Cycles	Special Notes
Branch if Carry Clear	BCC	40-5F	1	2	1
Branch if Higher or Same	(BHS)	40-5F	1	2	1, 2
Branch if Carry Set	BCS	60-7F	1	2	1
Branch if Lower	(BLO)	60-7F	1	2	1, 3
Branch if Not Equal	BNE	00-1F	1	2	1
Branch if Equal	BEQ	20-3F	1	2	1

SPECIAL NOTES

1. Each mnemonic of the Branch Instructions covers a range of 32 opcodes; e.g., BCC ranges from 40 through 5F. The actual memory location (target address) to which the branch is made is formed by adding the sign extended lower five bits of the opcode to the contents of the program counter.
2. The BHS instruction (shown in parentheses) is identical to the BCC instruction. The C bit is clear if the register was higher or the same as the location in the memory to which it was compared.
3. The BLO instruction (shown in parentheses) is identical to the BCS instruction. The C bit is set if the register was lower than the location in memory to which it was compared.

Table 6-4. Bit Manipulation Instructions

		Addressing Modes						
		Bit Set/Clear			Bit Test and Branch			
Function	Mnem	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Special Note
Branch if Bit n is set	BRSET n (n=0 . . . . 7)	–	–	–	C8+n	3	5	1
Branch if Bit n is clear	BRCLR n (n=0 . . . . 7)	–	–	–	C0+n	3	5	1
Set Bit n	BSET n (n=0 . . . .7)	D8+n	2	4	–	–	–	1
Clear Bit n	BCLR n (n=0 . . . . 7)	D0+n	2	4	–	–	–	1

SPECIAL NOTE

1. The opcode is formed by adding the bit number (0-7) to the basic opcode. For example: to clear bit six using the BSET6 instruction the opcode becomes DE (D8+6); BCLR5 becomes (C0+5); etc.

Table 6-5. Control Instructions

		Addressing Modes									
		Short-Direct			Inherent			Relative			
Function	Mnem	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Special Notes
Transfer A to X	TAX	BC	1	4	—	—	—	—	—	—	—
Transfer A to Y	TAY	BD	1	4	—	—	—	—	—	—	—
Transfer X to A	TXA	AC	1	4	—	—	—	—	—	—	—
Transfer Y to A	TYA	AD	1	4	—	—	—	—	—	—	—
Return from Subroutine	RTS	—	—	—	B3	1	2	—	—	—	—
Return from Interrupt	RTI	—	—	—	B2	1	2	—	—	—	—
No-Operation	NOP	—	—	—	—	—	—	—	—	—	1

SPECIAL NOTE

1 . The NOP instruction is equivalent to a branch if equal (BEQ) to the location designated by PC + 1.

Table 6-6. Instruction Set

Mnemonic	Addressing Modes									Flags	
	Inherent	Immediate	Direct	Short Direct	Bit/Set Clear	Bit-Test-Branch	Register Indirect	Extended	Relative	Z	C
ADD		X	X				X			Λ	Λ
AND		X	X				X			Λ	•
ASLA	Assembler converts this to "ADD $FF"									•	•
BCC									X	•	•
BCLR					X					•	•
BCS									X	•	•
BEQ									X	•	•
BHS	Assembler converts this to "BCC"									•	•
BLO	Assembler converts this to "BCS"									•	•
BNE									X	•	•
BRCLR						X				•	Λ
BRSET						X				•	Λ
BSET					X					•	•
CLRA	Assembler converts this to "SUB $FF"									Λ	Λ
CLRX	Assembler converts this to "MVI $80,#0"									•	•
CLRY	Assembler converts this to "MVI $81,#0"									•	•
CMP		X	X				X			Λ	Λ
COMA	X									Λ	Λ
DEC			X	X			X			Λ	•
DECA	Assembler converts this to "DEC $FF"									Λ	•
DECX	Assembler converts this to "DEC $80"									Λ	•
DECY	Assembler converts this to "DEC $81"									Λ	•
INC			X	X			X			Λ	•
INCA	Assembler converts this to "INC $FF"									Λ	•
INCX	Assembler converts this to "INC $80"									Λ	•
INCY	Assembler converts this to "INC $81"									Λ	•
JMP								X		•	•
JSR								X		•	•
LDA		X	X	X			X			Λ	•
LDXI	Assembler converts this to "MVI $80,DATA"									•	•
LDYI	Assembler converts this to "MVI $81,DATA"									•	•
MVI		X	X							•	•
NOP	Assembler converts this to "BEQ (PC) + 1"									•	•
ROLA	X									Λ	Λ
RTI	X									Λ	Λ
RTS	X									•	•
STA			X	X			X			Λ	•
SUB		X	X				X			Λ	Λ
TAX	Assembler converts this to "STA $80"									Λ	•
TAY	Assembler converts this to "STA $81"									Λ	•
TXA	Assembler converts this to "LDA $80"									Λ	•
TYA	Assembler converts this to "LDA $81"									Λ	•

Flag Symbols: Z = Zero, C = Carry/Borrow, Λ = Test and Set if True, Cleared Otherwise, • = Not Affected

Table 6-7. Instruction Set Opcode Map

	Branch Instructions							
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
1 0001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
2 0010	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
3 0011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
4 0100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
5 0101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
6 0110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
7 0111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
8 1000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
9 1001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
A 1010	2 BNE 1 REL	2 BNE 2 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
B 1011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
C 1100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
D 1101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
E 1110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
F 1111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL

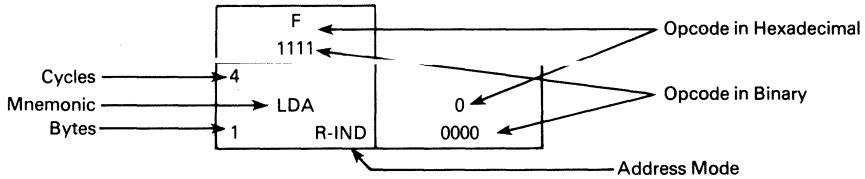
Abbreviations for Address Modes

INH	Inherent
S-D	Short Direct
B-T-B	Bit Test and Branch
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
R-IND	Register Indirect

*	Indicates Instruction Reserved for Future Use
#	Indicates Illegal Instruction

	Register/Memory, Control, and Read/Modify/Write Instructions				Bit Manipulation Instructions		Register/Memory and Read/Modify/Write		
	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 MVI 3 IMM	5 BRCLR0 3 B-T-B	4 BCLR0 2 BSC	4 LDA 1 R-IND	4 LDA 1 R-IND	0 0000
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR1 3 B-T-B	4 BCLR1 2 BSC	4 STA 1 R-IND	4 STA 1 R-IND	1 0001
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTI 1 INH	5 BRCLR2 3 BT-B	4 BCLR2 2 BSC	4 ADD 1 R-IND	4 ADD 1 R-IND	2 0010
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTS 1 INH	5 BRCLR3 3 B-T-B	4 BCLR3 2 BSC	4 SUB 1 R-IND	4 SUB 1 R-IND	3 0011
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 COMA 1 INH	5 BRCLR4 3 B-T-B	4 BCLR4 2 BSC	4 CMP 1 R-IND	4 CMP 1 R-IND	4 0100
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 ROLA 1 INH	5 BRCLR5 3 B-T-B	4 BCLR5 2 BSC	4 AND 1 R-IND	4 AND 1 R-IND	5 0101
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR6 3 B-T-B	4 BCLR6 2 BSC	4 INC 1 R-IND	4 INC 1 R-IND	6 0110
	4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR7 3 B-T-B	4 BCLR7 2 BSC	4 DEC 1 R-IND	4 DEC 1 R-IND	7 0111
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET0 3 B-T-B	4 BSET0 2 BSC	4 LDA 2 IMM	4 LDA 2 DIR	8 1000
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET1 3 B-T-B	4 BSET1 2 BSC	#	4 STA 2 DIR	9 1001
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET2 3 B-T-B	4 BSET2 2 BSC	4 ADD 2 IMM	4 ADD 2 DIR	A 1010
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET3 3 B-T-B	4 BSET3 2 BSC	SUB IMM	4 SUB 2 DIR	B 1011
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET4 3 B-T-B	4 BSET4 2 BSC	4 CMP 2 IMM	CMP DIR	C 1100
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET5 3 B-T-B	4 BSET5 2 BSC	4 AND 2 IMM	4 AND 2 DIR	D 1101
	4 JRSn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET6 3 B-T-B	4 BSET6 2 BSC	#	4 INC 2 DIR	E 1110
	4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET7 3 B-T-B	4 BSET7 2 BSC	#	4 DEC 2 DIR	F 1111

LEGEND



SECTION 7
ELECTRICAL SPECIFICATIONS

7.1 INTRODUCTION

This section contains the electrical specifications and associated timing for the MC6804P2.

7.2 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 (Comm.)	T_A	0 to 70	°C
Operating Temperature Range (Ind.)	T_A	−40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to 150	°C
Junction Temperature Plastic PLCC Cerdip	T_J	 150 150 175	°C

This device contains circuitry to protect the inputs against damage due to high static voltages of electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. Reliability of operation is enhanced if unused inputs except EXTAL are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

7

7.3 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic PLCC Cerdip	θ_{JA}	 70 120 60	°C/W

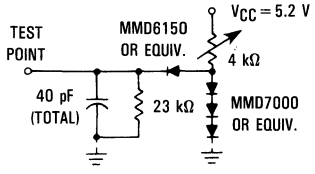


Figure 7-1. LSTTL Equivalent Test Load (Port B)

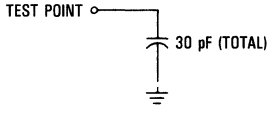


Figure 7-2. CMOS Equivalent Test Load (Ports, A, B, C)

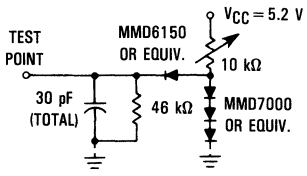


Figure 7-3. LSTTL Equivalent Test Load (Ports A, C, and TIMER)

7.4 POWER CONSIDERATIONS

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

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

where:

T_A ≡ Ambient Temperature, °C
θ_{JA} ≡ Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D ≡ $P_{INT} + P_{PORT}$
P_{INT} ≡ $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
P_{PORT} ≡ 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:

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

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A. 1-1185

7.5 ELECTRICAL CHARACTERISTICS

(V_{CC} = +5.0 Vdc ± 0.5 Vdc, V_{SS} = GND, T_A = 0°C to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Internal Power Dissipation — No Port Loading (See Figure 7-4)	P_{INT}	—	120	165	mW
Input High Voltage	V_{IH}	2.0	—	V_{CC}	V
Input Low Voltage	V_{IL}	−0.3	—	0.8	V
Input Capacitance	C_{in}	—	10	—	pF
Input Current ($\overline{IRQ}$, $\overline{RESET}$)	I_{in}	—	2	20	μA

7.6 SWITCHING CHARACTERISTICS

(V_{CC} = +5.0 Vdc ± 0.5 Vdc, V_{SS} = GND, T_A = 0°C to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency	f_{OSC}	4.0	—	11.0	MHz
Bit Time	t_{bit}	0.364	—	1.0	μs
Byte Cycle Time	t_{byte}	4.36	—	12.0	μs
$\overline{IRQ}$ and TIMER Pulse Width	t_{WL}, t_{WH}	2 × t_{byte}	—	—	—
$\overline{RESET}$ Pulse Width	t_{RWL}	2 × t_{byte}	—	—	—
$\overline{RESET}$ Delay Time (External Capacitance = 1.0 μF)	t_{RHL}	100	—	—	ms

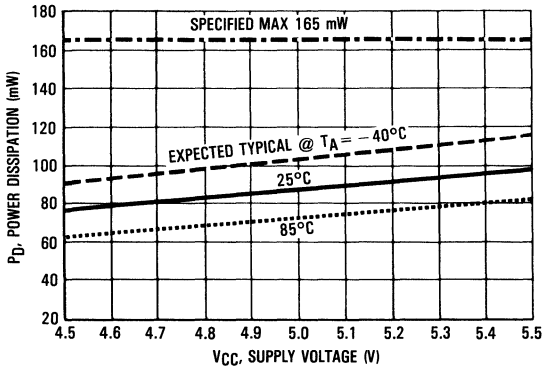


Figure 7-4. Typical Power Dissipation

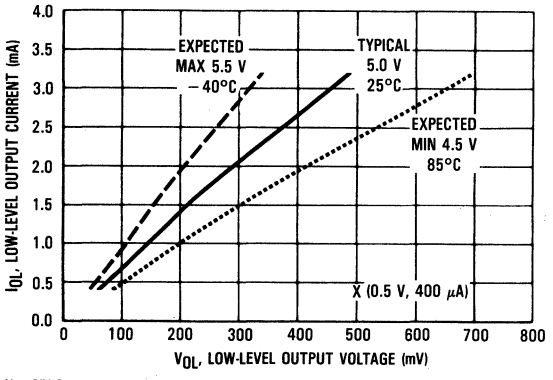
7.7 PORT DC ELECTRICAL CHARACTERISTICS

(V_{CC} = +5.0 Vdc ± 0.5 Vdc, V_{SS} = GND, T_A = 0°C to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Ports A, C, and Timer (Standard)					
Output Low Voltage, I_{Load} = 0.4 mA (See Figure 7-5)	V_{OL}	–	–	0.5	V
Output High Voltage, I_{Load} = −50 μA (See Figure 7-6)	V_{OH}	2.3	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	4	40	μA
Ports A and C (Open Drain)					
Output Low Voltage, I_{Load} = 0.4 mA (See Figure 7-5)	V_{OL}	–	–	0.5	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	4	40	μA
Open Drain Leakage (V_{out} = V_{CC})	I_{LOD}	–	4	40	μA
Ports A and C (CMOS Drive)					
Output Low Voltage, I_{Load} = 0.4 mA (Sink) (See Figure 7-5)	V_{OL}	–	–	0.5	V
Output High Voltage, I_{Load} = −10 μA	V_{OH}	V_{CC} − 1.0	–	–	V
Output High Voltage, I_{Load} = −50 μA (See Figure 7-7)	V_{OH}	2.3	–	–	V
Input High Voltage, I_{Load} = −300 μA Max	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage, I_{Load} = −300 μA Max	V_{IL}	−0.3	–	0.8	V
Hi-Z State Input Current (V_{in} = 0.4 V to V_{CC})	I_{TSI}	–	–	−300	μA
Port B (Standard)					
Output Low Voltage, I_{Load} = 1.0 mA (See Figure 7-8)	V_{OL}	–	–	0.5	V
Output Low Voltage, I_{Load} = 10 mA (Sink)	V_{OL}	–	–	1.5	V
Output High Voltage, I_{Load} = −100 μA (See Figure 7-9)	V_{OH}	2.3	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	8	80	μA
Port B (Open Drain)					
Output Low Voltage, I_{Load} = 1.0 mA (See Figure 7-8)	V_{OL}	–	–	0.5	V
Output Low Voltage, I_{Load} = 10 mA (Sink)	V_{OL}	–	–	1.5	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	8	80	μA
Open Drain Leakage (V_{out} = V_{CC})	I_{LOD}	–	8	80	μA
Port B (CMOS Drive)					
Output Low Voltage, I_{Load} = 1.0 mA (See Figure 7-8)	V_{OL}	–	–	0.5	V
Output High Voltage, I_{Load} = 10 mA (Sink)	V_{OL}	–	–	1.5	V
Output High Voltage, I_{Load} = −10 μA	V_{OH}	V_{CC} − 1.0	–	–	V
Output High Voltage, I_{Load} = −100 μA (See Figure 7-10)	V_{OH}	2.3	–	–	V
Input High Voltage, I_{Load} = −300 μA Max	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage, I_{Load} = −300 μA Max	V_{IL}	−0.3	–	0.8	V
Hi-Z State Input Current (V_{in} = 0.4 V to V_{CC})	I_{TSI}	–	–	−300	μA
Ports A, B, and C (Low Current Clamping Diode*)					
Input High Current V_{IH} = V_{CC} + 1.0 V	I_{IH}	–	–	100	μA
Input Low Current V_{IL} = 0.8 V	I_{IL}	–	–	−4.0	μA

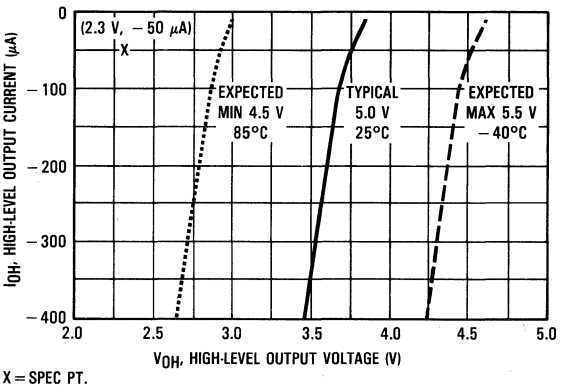
*Denotes not tested unless specified on ordering form.

7



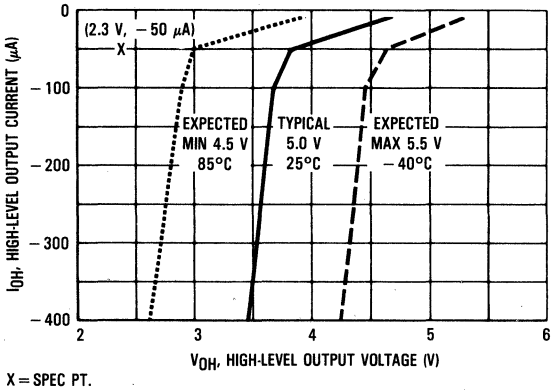
X = SPEC PT.

Figure 7-5. Typical V_{OL} vs I_{OL} for Ports A, C, and TIMER



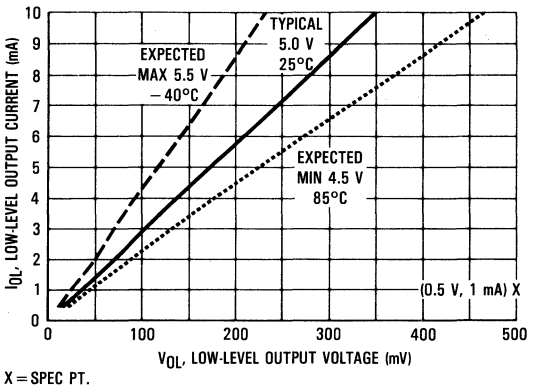
X = SPEC PT.

Figure 7-6. Typical V_{OH} vs I_{OH} for Ports A, C, and TIMER



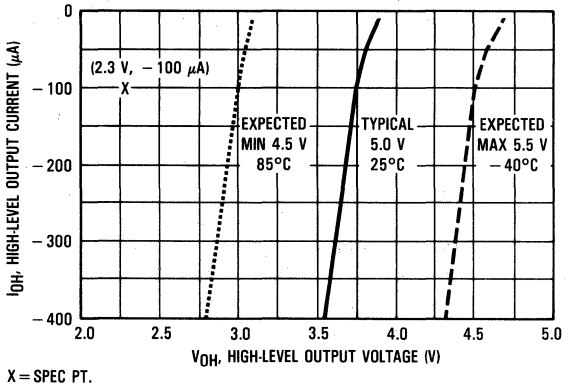
X = SPEC PT.

Figure 7-7. Typical V_{OH} vs I_{OH} for Ports A and C with CMOS Pullups



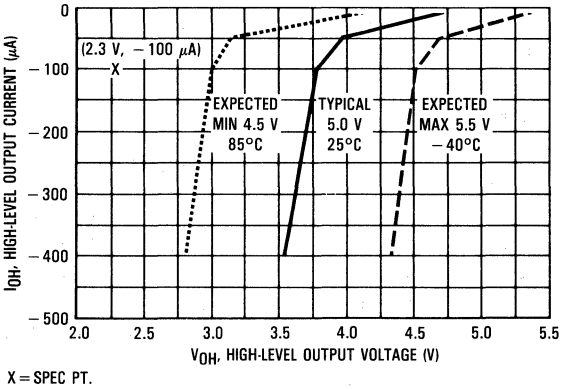
X = SPEC PT.

Figure 7-8. Typical V_{OL} vs I_{OL} for Port B



X = SPEC PT.

Figure 7-9. Typical V_{OH} vs I_{OH} for Port B



X = SPEC PT.

Figure 7-10. Typical V_{OH} vs I_{OH} for Port B with CMOS Pullups

SECTION 8
ORDERING INFORMATION

8.1 INTRODUCTION

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

MDOS, disk file

EPROM(s), 2716 or 2532

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

8.2 FLEXIBLE DISKS

The disk media submitted must be single-sided, single density, 8-inch, MDOS compatible floppies. The customer must clearly label the disk with the ROM pattern file name. The minimum MDOS system files as well as the absolute binary object file (filename.LO type of file) must be on the disk. An object file made from a memory dump; using the ROLLOUT command is also admissable. Consider submitting a source listing as well as: filename, .LX (EXORciser loadable format). This file will of course be kept confidential and is used 1) to speed up the process in house if any problems arise, and 2) to speed up our customer to factory interface if a user finds any software errors and needs assistance quickly from the factory representative.

MDOS is Motorola's Disk Operating System available on development systems such as EXORciser, EXORset, etc.

An MDOS disk, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. **When using the MDOS disk, include the entire memory image of both data and program space. All unused bytes, including the user's space, must be set to zero.**

8.3 EPROMs

A 2716 or 2532 type EPROM, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 2716 or 2532 EPROM, the EPROM must be programmed as follows in order to emulate the MC6804P2 MCU. **Start the data space ROM at EPROM address $018 and start program space ROM at EPROM address $400 (P2), $410 (J2), $600 (J1), and continue to memory space $7FF. All unused bytes, including the user's space, must be set to zero.** For shipment to Motorola, the EPROMs should be placed in a conductive IC carrier and packed securely. Do not use styrofoam.

EXORciser is a registered trademark of Motorola Inc.
EXORset and MDOS are trademarks of Motorola Inc.

8.4 VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, Motorola will program a blank 2716, 2532, or MDOS disk (**supplied by the customer**) from the data file used to create the custom mask to aid in the verification process.

8.5 ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and five volts. These RVUs are free with the minimum order quantity but are not production parts. These RVUs are not backed nor guaranteed by Motorola Quality Assurance.

8.6 ORDERING INFORMATION

The following table provides ordering information pertaining to the package type, temperature, and MC part numbers for the MC6804J1, MC6804J2, MC6804P2, and MC68704P2 devices.

Package Type	Temperature	MC Part Number
Plastic (P Suffix)	0°C to 70°C −40°C to 85°C	MC6804J1P/MC6804J2P/MC6804P2P MC6804J1CP/MC6804J2CP/MC6804P2CP
Cerdip (S Suffix)	0°C to 70°C −40°C to 85°C	MC68704P2S MC68704P2CS
Plastic Leaded Chip Carrier (FN Suffix)	0°C to 70°C −40°C to 85°C	MC6804P2FN MC6804P2CFN

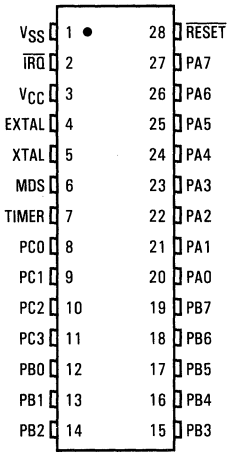
8

SECTION 9
MECHANICAL DATA

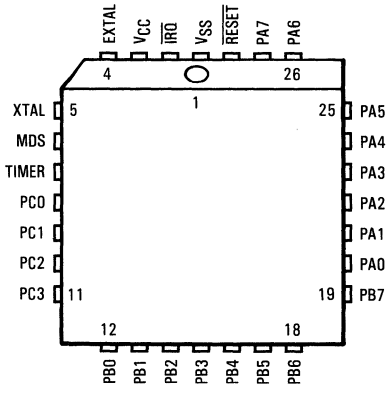
This section contains the pin assignment and package dimension diagrams for the MC6804P2 microcomputer.

9.1 PIN ASSIGNMENTS

28-PIN DUAL-IN-LINE PACKAGE



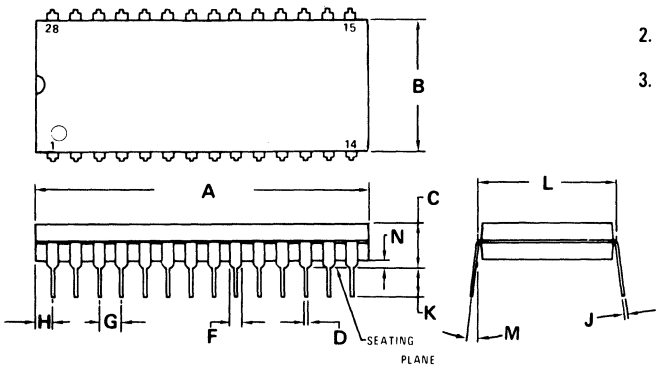
28-LEAD PLCC PACKAGE



9

9.2 PACKAGE DIMENSIONS

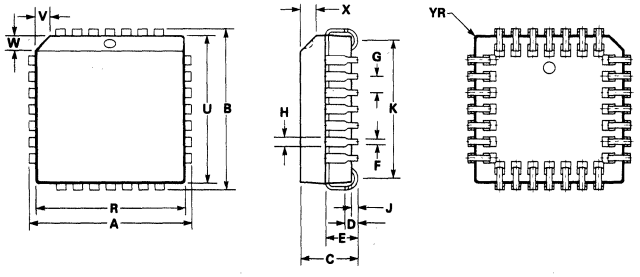
P SUFFIX
PLASTIC PACKAGE
CASE 710-02



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

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0^0	15^0	0^0	15^0
N	0.51	1.02	0.020	0.040

FN SUFFIX
PLASTIC LEADED CHIP CARRIER PACKAGE
CASE 776-01



NOTES:
1. DIMENSIONS R AND U DO NOT INCLUDE MOLD FLASH.
2. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
3. CONTROLLING DIMENSION: INCH

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	12.32	12.57	0.485	0.495
B	12.32	12.57	0.485	0.495
C	4.19	4.57	0.165	0.180
D	0.64	1.01	0.025	0.040
E	2.16	2.79	0.085	0.110
F	0.33	0.53	0.013	0.021
G	1.27 BSC		0.050 BSC	
H	0.66	0.81	0.026	0.032
J	0.38	0.63	0.015	0.025
K	9.91	10.92	0.390	0.430
R	11.43	11.58	0.450	0.456
U	11.43	11.58	0.450	0.456
V	1.07	1.21	0.042	0.048
W	1.07	1.21	0.042	0.048
X	1.07	1.42	0.042	0.056
Y	0.00	0.50	0.000	0.020

9

MC6804J1/MC6804J2/MC6804P2 MCU ORDERING FORM

Device Type: ☐ MC6804J1 ☐ MC6804J2 ☐ MC6804P2

Date _____ Customer PO Number _____

Customer Company _____

Address _____

City _____ State _____ Zip _____

Country _____

Phone _____ Extension _____

Customer Contact Person _____

Customer Part Number _____

(12 Characters Maximum — If Applicable)

Cut Here

MC6804J1/MC6804J2/MC6804P2 MASK OPTION LIST

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Level/Edge Sensitive

Output Drive (Select one Option per Port)

	LSTTL	CMOS/LSTTL	Open Drain
Port A	☐	☐	☐
Port B	☐	☐	☐
Port C	☐	☐	☐

Test Clamping Diode
(Available only with LSTTL or Open Drain Outputs)
- ☐ Yes
- ☐ No

Temperature
- ☐ 0°C to 70°C
- ☐ −40°C to 85°C

Clock Frequency: _____

Special Electrical Provisions: _____

(Customer specifications required.)

Pattern Media
- ☐ MDOS Disk File
- ☐ 2532 EPROMs
- ☐ 2716 EPROMs
- ☐ Other _____

(Requires prior factory approval.)

_____ (SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

_____ (SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required.)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

Cut Here

APPENDIX A
MC6804J1

The MC6804J1 microcomputer unit (MCU) device is similar to the MC6804P2 MCU device with several exceptions. These exceptions include 504 bytes of user program ROM, 72 bytes of user data space ROM, 12 bidirectional I/O port lines, and 20-pin packaging. Information throughout this document pertaining to the MC6804P2 MCU is also applicable to the MC6804J1 MCU. Difference information applicable to the MC6804J1 MCU is provided in this appendix.

A.1 INTRODUCTION

Information contained in **SECTION 1 INTRODUCTION** (general information, features, and block diagram) of this document applies to the MC6804J1 MCU device except for the areas described in the following paragraphs.

A.1.1 Features

The features of the MC6804J1 are as follows:

- 504 Bytes of User Program Space ROM
- 72 Bytes of User Data Space ROM
- 12 Bidirectional I/O Lines (Eight Lines with High Current Sink Capability)
- User Selectable Output Drive Options (LSTTL, LSTTL/CMOS, or Open-Drain Interface)
- 20-Pin Packaging

A.1.2 Block Diagram

Figure A-1 Illustrates the MC6804J1 MCU device block diagram.

A.2 FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU, AND REGISTERS

Information contained in **SECTION 2 FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU, AND REGISTERS** of this document applies to the MC6804J1 MCU device except for the areas described in the following paragraphs.

A.2.1 Input/Output Lines (PA4-PA7, PB0-PB7)

These 12 lines are arranged into one 4-bit port (A) and one 8-bit port (B). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to paragraph **A.4 INPUT/OUTPUT PORTS** for additional information.

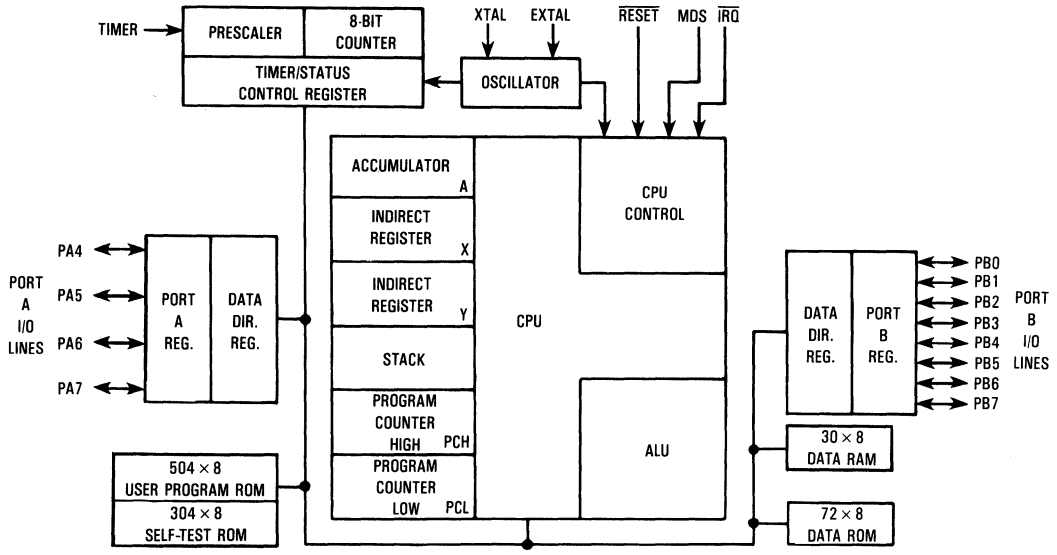


Figure A-1. MC6804J1 Block Diagram

A.2.2 Memory

As shown in Figure A-2, the MC6804J1 MCU memory map consists of 4352 bytes of addressable memory and I/O register locations. The MCU memory map is divided into two groups of memory spaces: 4096 bytes of program space, and 256 bytes of data space.

The program space includes 304 bytes of self-test ROM, 504 bytes of program ROM, eight bytes of self-test and user program vectors, and 3276 bytes of reserved memory locations for a total of 4096 bytes.

The data space includes two port data registers, two port data direction registers, one timer status control register, 72 bytes of user data space ROM, two indirect registers (X, Y), 30 bytes of user data space RAM, one prescaler register, one timer count register, one accumulator register, and 144 bytes of reserved memory locations for a total of 256 bytes.

A

Only program space is addressed by the program counter, thus instructions may only be executed from the program space. Data space may be addressed by an instruction operand or an indirect register. No instructions can be executed out of the data space area.

In addition to the program and data memory spaces, a non-accessible subroutine stack space RAM (not shown in Figure A-2) is provided. This stack space consists of a last-in-first-out (LIFO) register which is used with inherent addressing to stack the return address for subroutines.

Indirect X and Y register locations $80 and $81 are normally used as pointers (e.g., indirect addressing to data space locations). The short direct addressing allows access to the four data space addresses $80–$83 with single byte opcodes. Operations allowed are increment, decrement, load, and store. Data space locations $82 and $83 can be used for 8-bit counter locations.

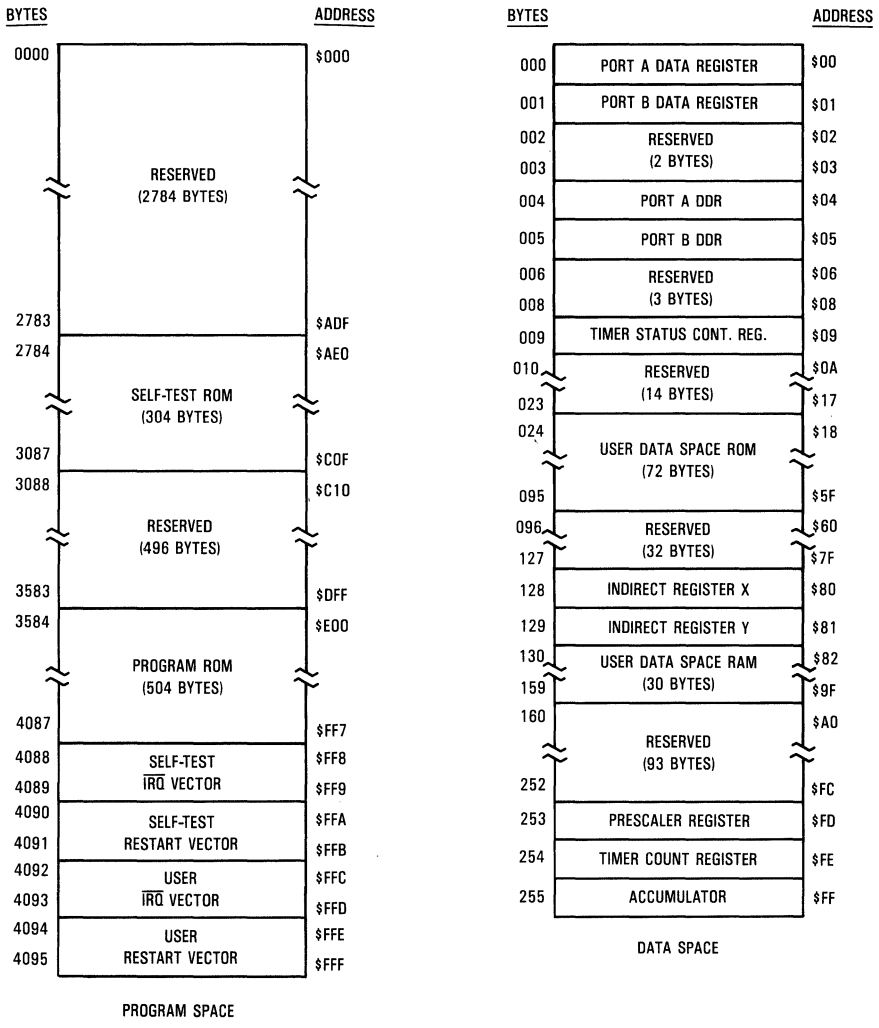


Figure A-2. MC6804J1 Memory Map

A

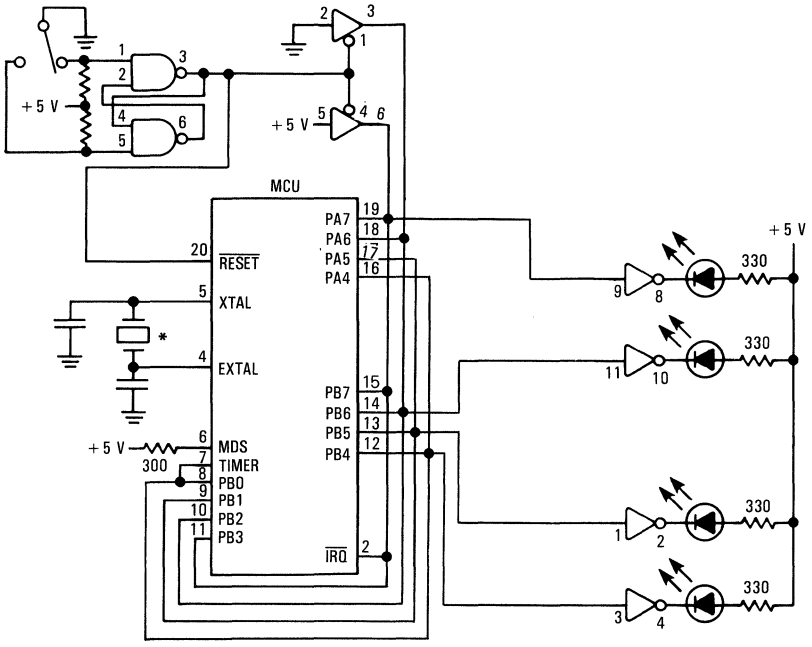
A.3 INTERRUPT, SELF-TEST, RESET, AND INTERNAL CLOCK GENERATOR

Information contained in **SECTION 4 INTERRUPT, SELF-TEST, RESET, AND INTERNAL CLOCK GENERATOR** of this document applies to the MC6804J1 MCU device except for the areas described in the following paragraphs.

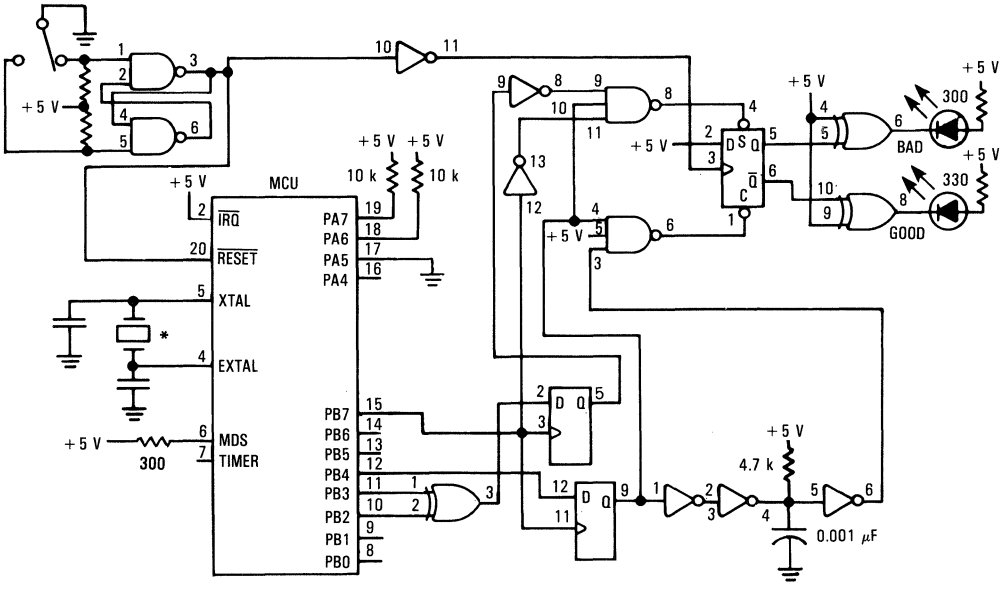
A.3.1 Self-Test

The MC6804J1 MCU has a unique internal ROM based off-line self-test capability using signature analysis techniques. The self-test connections are illustrated in Figure A-3A. To perform a test of the MCU, connect the MCU as shown and monitor the LEDs for a 1101 ($OD) pattern.

A special ROM self-test utilizing the signature analysis circuitry is also included. To initiate a test of the ROM, connect the circuit as shown in Figure A-3B.



(a) FUNCTIONAL TEST



(b) SIMPLE ROM VERIFY CHECK

NOTES:
1. *used with crystal option only.
2. 10 k pullup resistors to +5 V are required on all connected I/O port lines for open-drain option.

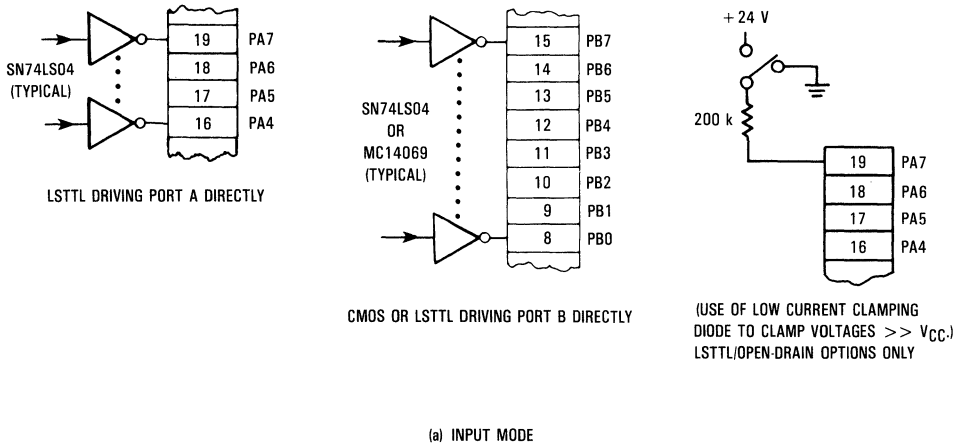
Figure A-3. Self-Test Circuit

A.4 INPUT/OUTPUT PORTS

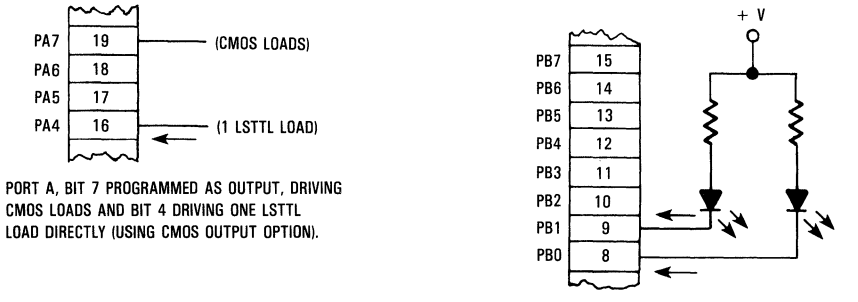
Information contained in **SECTION 5 INPUT/OUTPUT PORTS** of this document applies to the MC6804J1 MCU device except for the areas described in the following paragraphs.

A.4.1 Input/Output

There are 12 input/output pins. All pins (port A and B) are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). Figure A-4 illustrates typical port connections for the MC6804J1 device. The 12 bidirectional lines may be configured by port to be LSTTL (standard configuration), LSTTL/CMOS (mask option), or open drain (mask option). Port B outputs are LED compatible.



(a) INPUT MODE



(b) OUTPUT MODE

Figure A-4. Typical Port Connections

A

A.4.2 Port A and B Data Registers

Port A and B data registers (shown below) are implemented as RAM locations and can be read or written to.

Port A Data Register ($00)

	7	6	5	4	3	2	1	0	
MSB					X	X	X	X	LSB

NOTE: Four LSB bits are don't care (X) bits.

Port B Data Register ($01)

	7	6	5	4	3	2	1	0	
MSB									LSB

A.4.3 Port A and B Data Direction Registers

Port A and B data direction registers (shown below) are implemented as RAM locations and can be read or written to.

Port A Data Direction Register ($04)

	7	6	5	4	3	2	1	0	
MSB					0	0	0	0	LSB

NOTE: Four LSB bits are cleared (logic 0) after reset.
These bits must not be set (logic 1).

Port B Data Direction Register ($05)

	7	6	5	4	3	2	1	0	
MSB									LSB

A.5 ORDERING INFORMATION

Information contained in **SECTION 8 ORDERING INFORMATION** of this document applies to the MC6804J1 MCU device. The MC6804J1 device is only available in the 20-pin plastic dual-in-line (DIP) package.

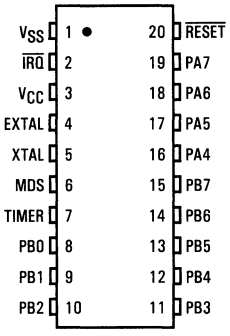
A

A.6 MECHANICAL DATA

The 20-pin dual-in-line pin assignments and package dimensions for the MC6804J1 MCU device is provided on the following page.

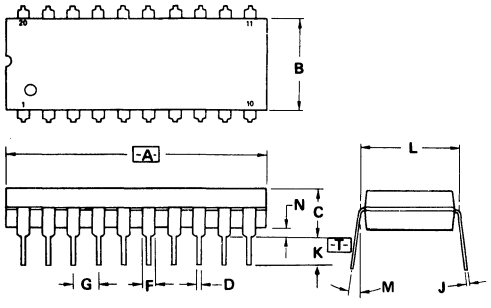
A.6.1 Pin Assignments

PIN ASSIGNMENT



A.6.2 Package Dimensions

P SUFFIX
PLASTIC PACKAGE
CASE 738-02



NOTES:
1. DIM -A- IS DATUM.
2. POSITIONAL TOL FOR LEADS;
   ⌖ Ø 0.25 (0.010) M T A M
3. -T- IS SEATING PLANE.
4. DIM "B" DOES NOT INCLUDE MOLD FLASH.
5. DIM -L- TO CENTER OF LEADS WHEN FORMED PARALLEL.
6. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	25.65	27.18	1.010	1.070
B	6.10	6.60	0.240	0.260
C	3.94	4.57	0.155	0.180
D	0.38	0.56	0.015	0.022
F	1.27	1.78	0.050	0.070
G	2.54 BSC		0.100 BSC	
J	0.20	0.38	0.008	0.015
K	2.79	3.56	0.110	0.140
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

A

A

APPENDIX B
MC6804J2

The MC6804J2 microcomputer unit (MCU) device is similar to the MC6804J1 and MC6804P2 MCU devices with several exceptions. These exceptions include 1000 bytes of user program ROM, 72 bytes of user data space ROM, 12 bidirectional I/O port lines, and 20-pin packaging. Information throughout this document pertaining to the MC6804P2 MCU is also applicable to the MC6804J2 MCU. Difference information applicable to the MC6804J2 MCU is provided in this appendix.

B.1 INTRODUCTION

Information contained in **SECTION 1 INTRODUCTION** (general information, features, and block diagram) of this document applies to the MC6804J2 MCU device except for the areas described in the following paragraphs.

B.1.1 Features

The features of the MC6804J2 are as follows:

- 1000 Bytes of User Program Space ROM
- 72 Bytes of User Data Space ROM
- 12 Bidirectional I/O Lines (Eight Lines with High Current Sink Capability)
- User Selectable Output Drive Options (LSTTL, LSTTL/CMOS, or Open-Drain Interface)
- 20-Pin Packaging

B.1.2 Block Diagram

Figure B-1 Illustrates the MC6804J2 MCU device block diagram.

B.2 FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU, AND REGISTERS

Information contained in **SECTION 2 FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU, AND REGISTERS** of this document applies to the MC6804J2 MCU device except for the areas described in the following paragraphs.

B.2.1 Input/Output Lines (PA4-PA7, PB0-PB7)

These 12 lines are arranged into one 4-bit port (A) and one 8-bit port (B). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to paragraph **B.4 INPUT/OUTPUT PORTS** for additional information.

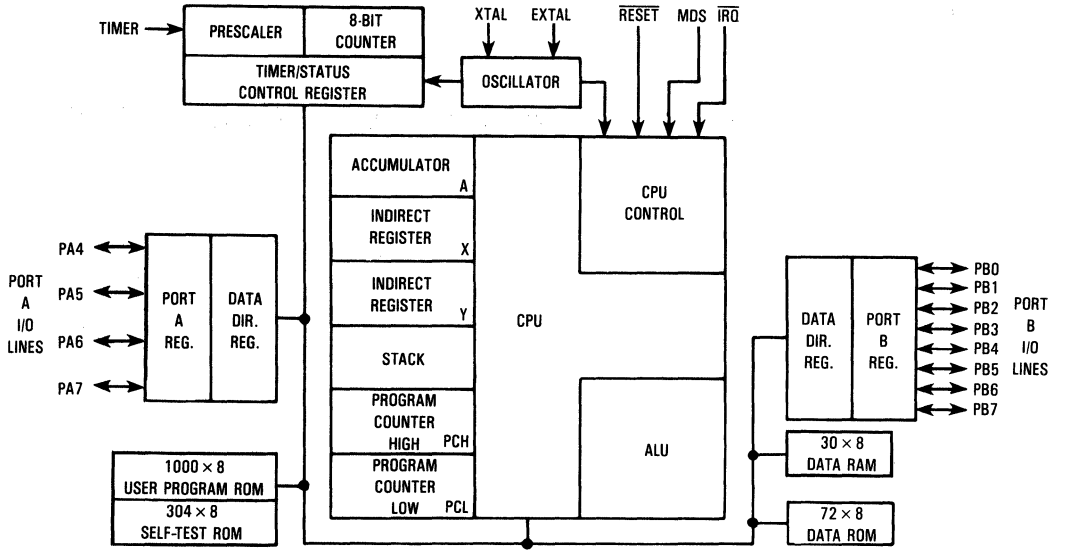


Figure B-1. MC6804J2 Block Diagram

B.2.2 Memory

As shown in Figure B-2, the MC6804J2 MCU memory map consists of 4352 bytes of addressable memory and I/O register locations. The MCU memory map is divided into two groups of memory spaces: 4096 bytes of program space, and 256 bytes of data space.

The program space includes 304 bytes of self-test ROM, 1000 bytes of program ROM, eight bytes of self-test and user program vectors, and 2784 bytes of reserved memory locations for a total of 4096 bytes.

The data space includes two port data registers, two port data direction registers, one timer status control register, 72 bytes of user data space ROM, two indirect registers (X, Y), 30 bytes of user data space RAM, one prescaler register, one timer count register, one accumulator register, and 144 bytes of reserved memory locations for a total of 256 bytes.

Only program space is addressed by the program counter, thus instructions may only be executed from the program space. Data space may be addressed by an instruction operand or an indirect register. No instructions can be executed out of the data space area.

B

In addition to the program and data memory spaces, a non-accessible subroutine stack space RAM (not shown in Figure B-2) is provided. This stack space consists of a last-in-first-out (LIFO) register which is used with inherent addressing to stack the return address for subroutines.

Indirect X and Y register locations $80 and $81 are normally used as pointers (e.g., indirect addressing to data space locations). The short direct addressing allows access to the four data space addresses $80–$83 with single byte opcodes. Operations allowed are increment, decrement, load, and store. Data space locations $82 and $83 can be used for 8-bit counter locations.

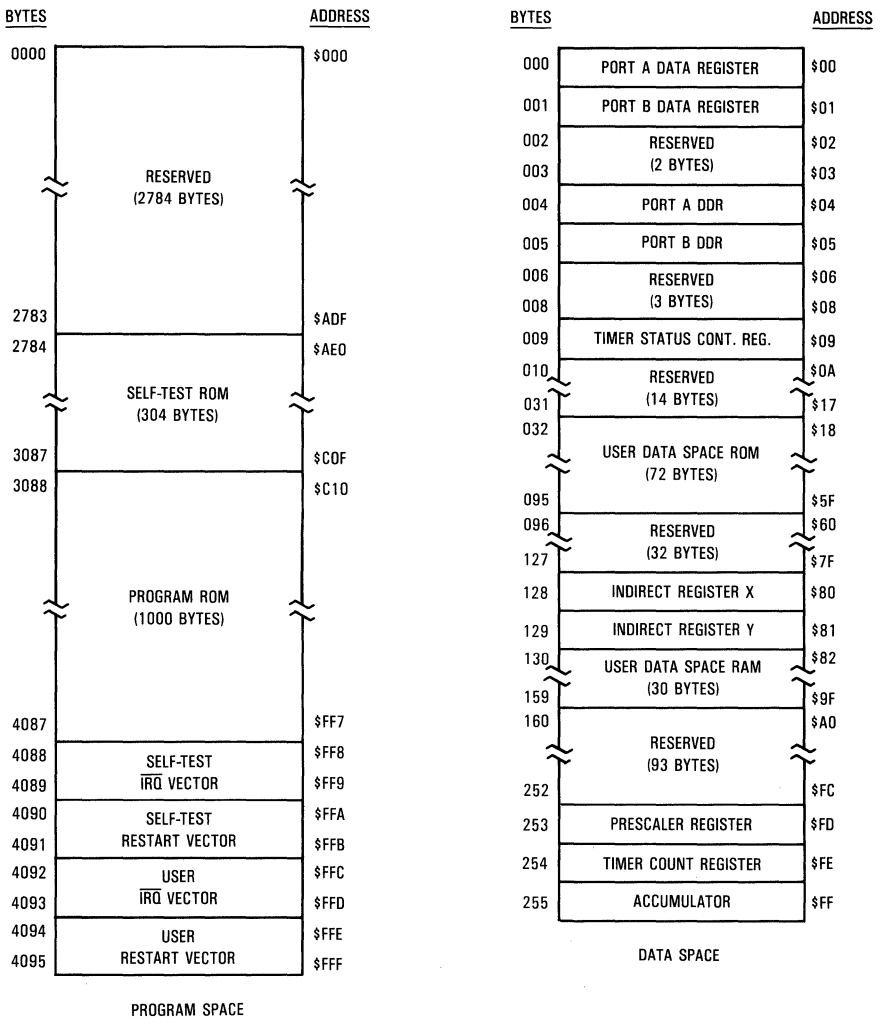


Figure B-2. MC6804J2 Memory Map

B.3 INTERRUPT, SELF-TEST, RESET, AND INTERNAL CLOCK GENERATOR

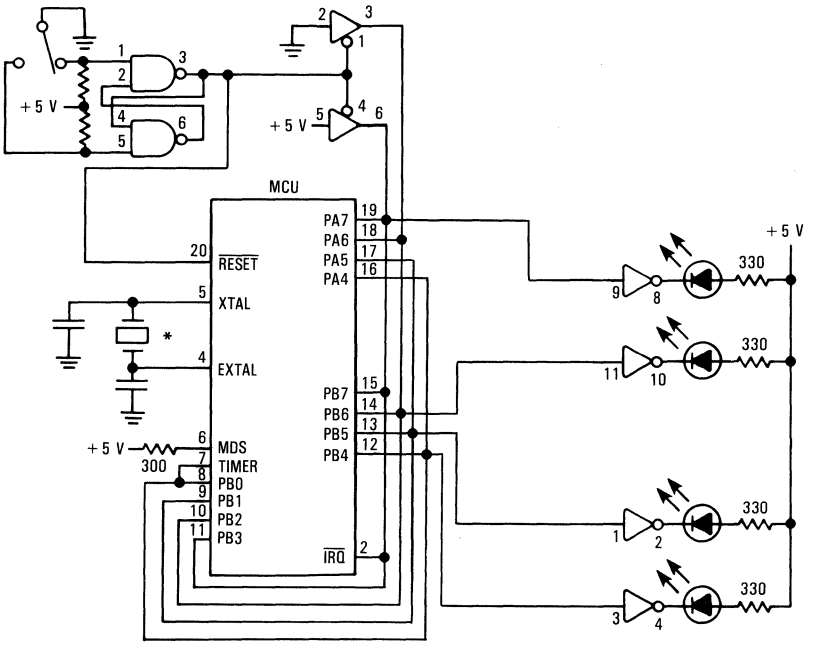
Information contained in **SECTION 4 INTERRUPT, SELF-TEST, RESET, AND INTERNAL CLOCK GENERATOR** of this document applies to the MC6804J2 MCU device except for the areas described in the following paragraphs.

B

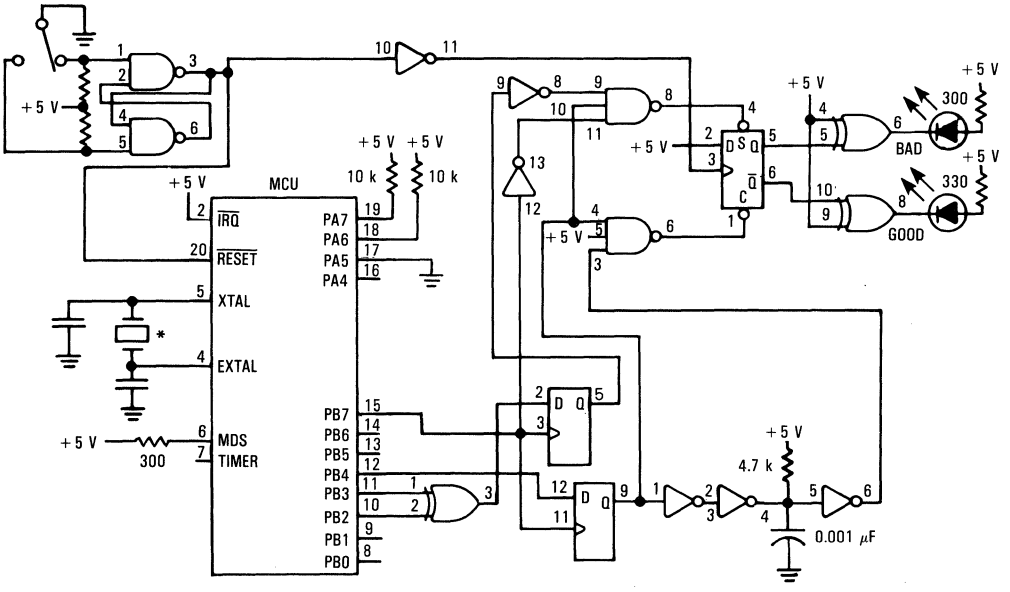
B.3.1 Self-Test

The MC6804J2 MCU has a unique internal ROM based off-line self-test capability using signature analysis techniques. The self-test connections are illustrated in Figure B-3A. To perform a test of the MCU, connect the MCU as shown and monitor the LEDs for a 1101 ($OD) pattern.

A special ROM self-test utilizing the signature analysis circuitry is also included. To initiate a test of the ROM, connect the circuit as shown in Figure B-3B.



(a) FUNCTIONAL TEST



(b) SIMPLE ROM VERIFY CHECK

NOTES:
1. *used with crystal option only.
2. 10 k pullup resistors to +5 V are required on all connected I/O port lines for open-drain option.

Figure B-3. Self-Test Circuit

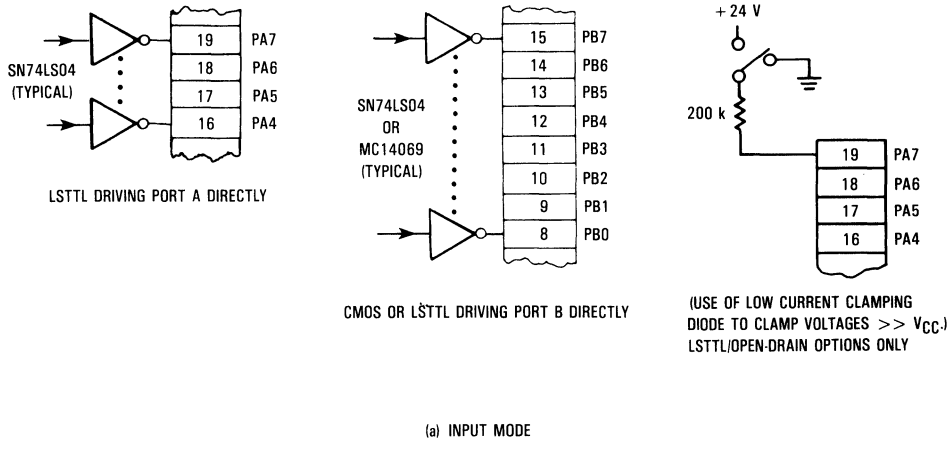
B

B.4 INPUT/OUTPUT PORTS

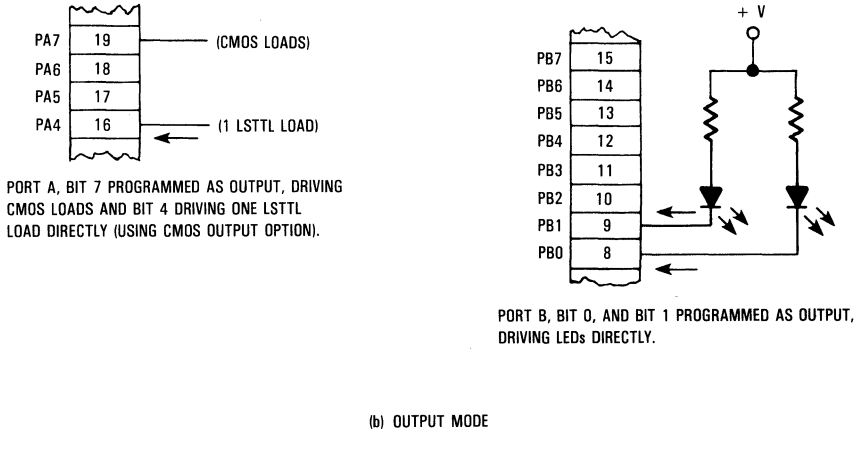
Information contained in **SECTION 5 INPUT/OUTPUT PORTS** of this document applies to the MC6804J2 MCU device except for the areas described in the following paragraphs.

B.4.1 Input/Output

There are 12 input/output pins. All pins (port A and B) are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). Figure B-4 illustrates typical port connections for the MC6804J2 device. The 12 bidirectional lines may be configured by port to be LSTTL (standard configuration), LSTTL/CMOS (mask option), or open drain (mask option). Port B outputs are LED compatible.



(a) INPUT MODE



(b) OUTPUT MODE

Figure B-4. Typical Port Connections

B

B.4.2 Port A and B Data Registers

Port A and B data registers (shown below) are implemented as RAM locations and can be read or written to.

Port A Data Register ($00)

	7	6	5	4	3	2	1	0	
MSB					X	X	X	X	LSB

NOTE: Four LSB bits are don't care (X) bits.

Port B Data Register ($01)

	7	6	5	4	3	2	1	0	
MSB									LSB

B.4.3 Port A and B Data Direction Registers

Port A and B data direction registers (shown below) are implemented as RAM locations and can be read or written to.

Port A Data Direction Register ($04)

	7	6	5	4	3	2	1	0	
MSB					0	0	0	0	LSB

NOTE: Four LSB bits are cleared (logic 0) after reset.
These bits must not be set (logic 1).

Port B Data Direction Register ($05)

	7	6	5	4	3	2	1	0	
MSB									LSB

B.5 ORDERING INFORMATION

Information contained in **SECTION 8 ORDERING INFORMATION** of this document applies to the MC6804J2 MCU device. The MC6804J2 device is only available in the 20-pin plastic dual-in-line (DIP) package.

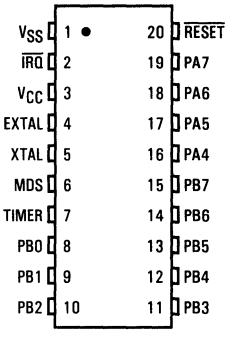
B.6 MECHANICAL DATA

The 20-pin dual-in-line pin assignments and package dimensions for the MC6804J2 MCU device is provided on the following page.

B

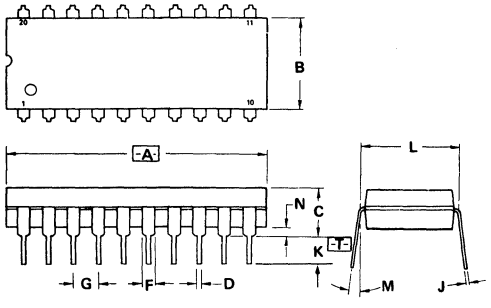
B.6.1 Pin Assignments

PIN ASSIGNMENT



B.6.2 Package Dimensions

P SUFFIX
PLASTIC PACKAGE
CASE 738-02



NOTES:
1. DIM -A- IS DATUM.
2. POSITIONAL TOL FOR LEADS;
   ⌖ Ø 0.25 (0.010) M T A M
3. -T- IS SEATING PLANE.
4. DIM "B" DOES NOT INCLUDE MOLD FLASH.
5. DIM -L- TO CENTER OF LEADS WHEN FORMED PARALLEL.
6. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	25.65	27.18	1.010	1.070
B	6.10	6.60	0.240	0.260
C	3.94	4.57	0.155	0.180
D	0.38	0.56	0.015	0.022
F	1.27	1.78	0.050	0.070
G	2.54 BSC		0.100 BSC	
J	0.20	0.38	0.008	0.015
K	2.79	3.56	0.110	0.140
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

B

B

APPENDIX C
MC68704P2

The MC68704P2 erasable programmable read-only memory (EPROM) microcomputer unit (MCU) device is similar to the MC6804P2 MCU device with several exceptions. These exceptions include 1016 bytes of user program EPROM, 72 bytes of user data space EPROM, two breakpoint registers, and an EPROM mask option register (MOR). The EPROM feature of the MC6704P2 MCU enables the user to emulate either of the MC6804J1, MC6804J2, or MC6804P2 MCU devices. Information pertaining to the EPROM emulation feature is contained in **APPENDIX D MC6804J1, MC6804J2, AND MC6804P2 EMULATION**. Information throughout this document pertaining to the MC6804P2 MCU is also applicable to the MC68704P2 EPROM MCU. Difference information applicable to the MC68704P2 EPROM MCU is provided in this appendix.

C.1 INTRODUCTION

Information contained in **SECTION 1 INTRODUCTION** (general information, features, and block diagram) of this document applies to the MC68704P2 EPROM MCU device except for the areas described in the following paragraphs.

C.1.1 Features

The features of the MC68704P2 are as follows:

- Emulation of MC6804J1, MC6804J2, and MC6804P2
- 1016 Bytes of User Program Space EPROM
- 72 Bytes of User Data Space EPROM
- Two Breakpoint Registers
- EPROM Mask Option Register

C.1.2 Block Diagram

Figure C-1 illustrates the MC68704P2 EPROM MCU device block diagram.

C.2 FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU, AND REGISTERS

Information contained in **SECTION 2 FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU, AND REGISTERS** of this document applies to the MC68704P2 EPROM MCU device except for the areas described in the following paragraphs.

C

C.2.1 Memory

As shown in Figure C-2, the MC68704P2 EPROM MCU memory map consists of 4352 bytes of addressable memory and I/O register locations. The MCU memory map is divided into two groups of memory spaces: 4096 bytes of program space, and 256 bytes of data space.

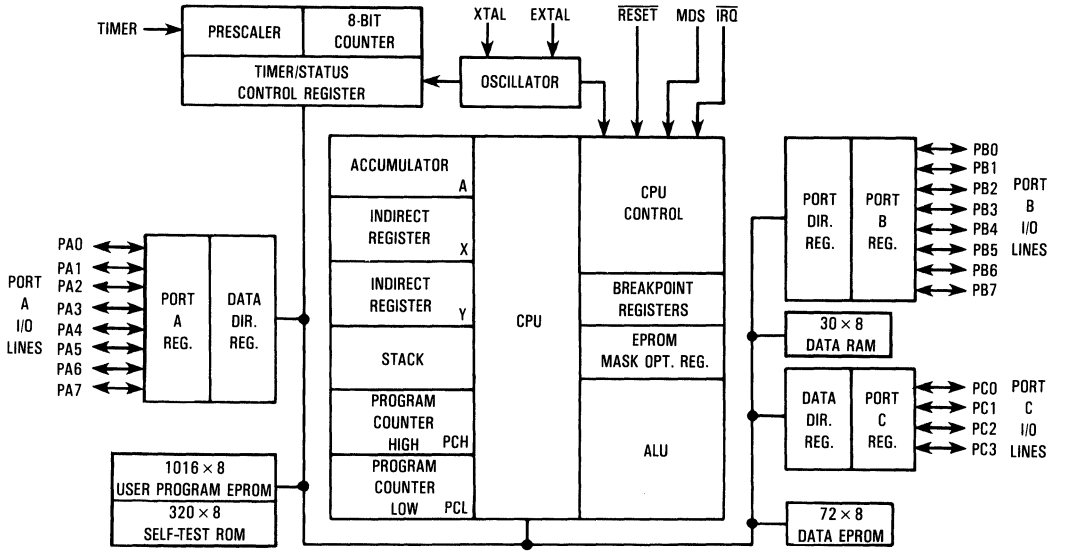


Figure C-1. MC68704P2 Block Diagram

The program space includes 320 bytes of self-test ROM, 1016 bytes of program EPROM, four bytes of self-test vectors (mask ROM), four bytes of user program vectors (EPROM), and 2752 bytes of reserved memory locations for a total of 4096 bytes.

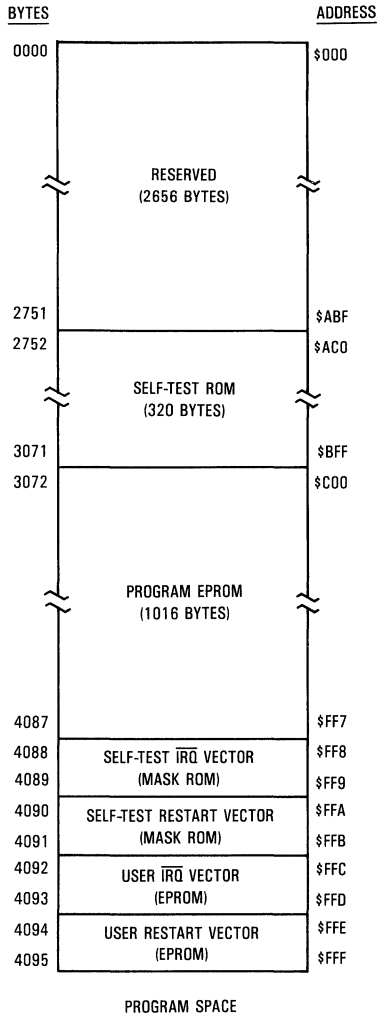
The data space includes three port data registers, three port data direction registers, one timer status control register, two breakpoint registers, one EPROM mask option register (MOR), 72 bytes of user data space EPROM, two indirect registers (X,Y), 30 bytes of user data space RAM, one prescaler register, one timer count register, one accumulator register, and 139 bytes of reserved memory locations for a total of 256 bytes.

Only program space is addressed by the program counter, thus instructions may only be executed from the program space. Data space may be addressed by an instruction operand or an indirect register. No instructions can be executed out of the data space area.

In addition to the program and data memory spaces, a non-accessible subroutine stack space RAM (not shown in Figure C-2) is provided. This stack space consists of a last-in-first-out (LIFO) register which is used with inherent addressing to stack the return address for subroutines.

Indirect X and Y register locations $80 and $81 are normally used as pointers (e.g., indirect addressing to data space locations). The short direct addressing allows access to the four data space addresses $80-$83 with single bytes opcodes. Operations allowed are increment, decrement, load, and store. Data space locations $82 and $83 can be used for 8-bit counter locations.

C



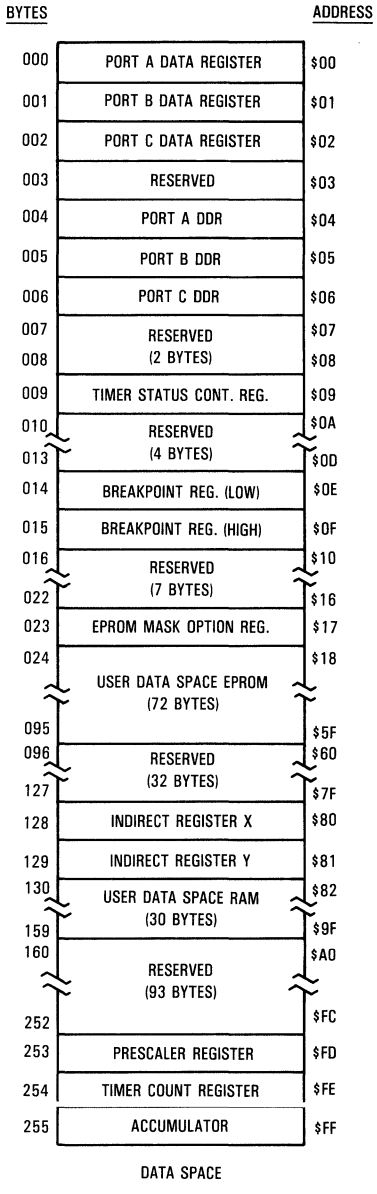


Figure C-2. MC68704P2 Memory Map

C

C.3 BREAKPOINT REGISTERS

The breakpoint registers (shown below) are used as a program debugging aid. To enable the breakpoint registers, the MDS pin must be pulled high via a 300 ohm resistor to +5 volts, and both PA6 and PA7 pins must be pulled low via 10 kilohm resistors to ground.

Breakpoint Address Register Low (ARL)

7	6	5	4	3	2	1	0	
A7	A6	A5	A4	A3	A2	A1	A0	$0E

b7-b0 A7-A0 — Breakpoint address bits A7 through A0. Reset clears address bits A7 through A0.

Breakpoint Address Register High (ARH)

7	6	5	4	3	2	1	0	
X	X	X	X	A11	A10	A9	A8	$0F

b7-b4 — Don't care bits (not used).

b3-b0 A11-A8 — Breakpoint address bits A11 through A8. Reset clears address bits A11 through A8.

A breakpoint address is written into address registers ARL and ARH by the user. The lower eight bits (A0-A7) of the breakpoint address are written into the ARL. The upper four bits (A8-A11) of the breakpoint address are written into the ARH.

NOTE

ARL must be written to after writing to ARH.

ARL and ARH are then concatenated to form the breakpoint address. When the processor fetches an instruction with the same address as the breakpoint address, MDS pin goes low for one machine cycle. This operation will not alter program flow.

C.4 EPROM MASK OPTION REGISTER

The EPROM mask option register (MOR) is used by the MC68704P2 device (during emulation) to select clock/oscillator, port, and interrupt request (IRQ) edge- and level-sensitive triggering options that are available on the MC6804J1/MC6804J2/MC6804P2 devices. The EPROM register is located at address location $17.

EPROM Mask Option Register

7	6	5	4	3	2	1	0	
OSC	X	PORT A	X	PORT C	PORT B	IRQ	X	$17

b7, OSC — Oscillator option bit. When zero (erased), selects the crystal mode operation. When high, resistor/capacitor (R/C) mode is selected. The crystal mode is selected in the EPROM programming mode, regardless of the state of this bit.

C

b6, X — Don't care bit (not used).

b5, PORT A — Port A output select bit. When zero (erased), selects three-state output mode for port A. When high, open drain output mode is selected.

b4, X — Don't care bit (not used).

b3, PORT C — Port C output select bit. When zero (erased), selects three-state output mode for port C. When high, open drain output mode is selected.

b2, PORT B — Port B output select bit. When zero (erased), selects three-state output mode for port B. When high, open drain output mode is selected.

b1, IRQ — Interrupt request bit. When zero (erased), selects edge-sensitive triggering input mode for the $\overline{IRQ}$ pin. When high, level-sensitive triggering input mode is selected.

b0, X — Don't care bit (not used).

C.5 ELECTRICAL SPECIFICATIONS

Information contained in **SECTION 7 ELECTRICAL SPECIFICATIONS** of this document applies to the MC68704P2 EPROM MCU device except for the areas described in the following paragraphs.

C.5.1 Electrical Characteristics

(V_{CC} = +5.0 Vdc ±0.5 Vdc, V_{SS} = GND, T_A = 0°C to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Internal Power Dissipation — No Port Loading	P_{INT}	—	165	275	mW

C.5.2 Programming Operation Electrical Characteristics

(V_{CC} = +5.0 Vdc ±0.5 Vdc, V_{SS} = GND, T_A = 20°C to 30°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage	V_{PP}	20	21	22	V
V_{PP} Supply Current V_{PP} = 22.0 V	I_{PP}	—	10	20	mA
Programming Oscillator Frequency	f_{oscp}	—	10	11	MHz
Programming Time (Per Byte)	T_{PRG}	—	5	50	mS

C.5.3 Port DC Electrical Characteristics

Refer to the port dc electrical characteristics in **SECTION 7** of this document. The CMOS drive specifications for ports A, B, and C are not applicable to the MC68704P2 EPROM MCU device.

C.6 ORDERING INFORMATION

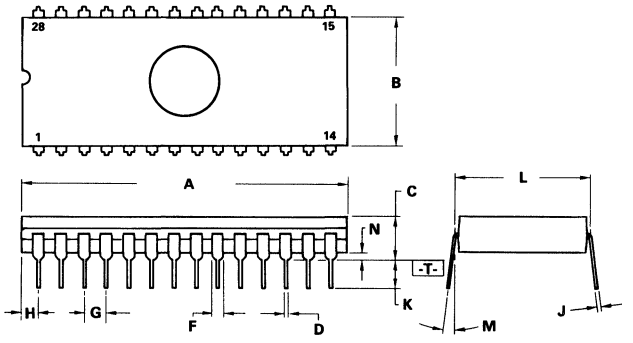
Information contained in **SECTION 8 ORDERING INFORMATION** of this document applies to the MC68704P2 EPROM MCU device. The MC68704P2 device is only available in the 28-pin ceramic dual-in-line (CERDIP) package.

C

C.7 MECHANICAL DATA

The 28-pin dual-in-line pin assignment information contained in **SECTION 9 MECHANICAL DATA** of this document applies to the MC68704P2 EPROM MCU device except for the packaging information which is provided below. The MC68704P2 device is only available in the 28-pin ceramic dual-in-line (CERDIP) package.

S SUFFIX
CERDIP PACKAGE
CASE 733A-01



NOTES:
1. DIMENSION "A" IS A DATUM. T IS BOTH A DATUM AND A SEATING PLANE.
2. POSITIONAL TOLERANCE FOR LEADS: (28 PLACES)
   ⌖ | ⌀ 0.25 (0.010) Ⓜ | T | A Ⓜ
3. DIMENSIONS "A" & B INCLUDE MENISCUS.
4. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
6. CONTROLLING DIMENSION: INCH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.84	1.435	1.490
B	12.70	15.36	0.500	0.605
C	4.06	6.09	0.160	0.240
D	0.38	0.55	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.17	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.27	0.020	0.050

C

APPENDIX D
MC6804J1, MC6804J2, AND MC6804P2 EMULATION

D.1 INTRODUCTION

This appendix describes EPROM erasing and the programming technique used to program the MC68704P2 MCU internal EPROM to emulate either the MC6804J1, MC6804J2, or MC6804P2 MCU device. Unlike the M6805 family of EPROM MCUs which have an on-chip program capability (bootstrap-loader program stored in mask ROM), the MC68704P2 does not have this capability. Therefore, additional programming hardware and software are required to program the MC68704P2 EPROM MCU device. This appendix also provides a schematic diagram (Figure D-1) which illustrates a typical MC68704P2 EPROM MCU programming board/circuitry used in conjunction with the programming software.

D.2 EMULATION LIMITATIONS

The MC68704P2 EPROM MCU device is designed to emulate the functions of either the MC6804J1, MC6804J2, or MC6804P2 MCU device as closely as possible. The only limitations pertain to the CMOS pullup option, execution out of data space, and packaging pin assignments of the MCU device to be emulated. These limitations do not apply to the timing, execution speed, and functionality of the MCU device being emulated. The limitations are as follows:

1. The MC68704P2 EPROM MCU cannot emulate the CMOS pullup option. To implement the CMOS option, external 40K ohm pullup resistors are connected to the specific I/O port signal lines. All other options are available through the correct use of the mask option register (MOR) bytes.

2. Due to the implementation of the MCU programming hardware, it was necessary that the program counter (PC) have access to both the data space and program space EPROM. Therefore, the MC68704P2 EPROM MCU will execute code out of the data space EPROM ($18-$5F). This anomaly is not permitted on the MC6804J1/MC6804J2/MC6804P2 ROM devices. Therefore, it is necessary when planning on operating ROM patterns from the MC68704P2 EPROM MCU device not to use data space as extra program space.

3. The MC6804J1 and MC6804J2 devices are packaged in 20-pin dual-in-line (DIL) packages, and the MC6804P2 and MC68704P2 devices are packaged in 28-pin DIL packages. Device pin assignments must be carefully adhered to. When emulating a 20-pin MCU device with the MC68704P2, all unused pins (PA0-PA3 and PC0-PC3) should be grounded externally through a 10K ohm resistor. This will allow the MC68704P2 device to emulate the software execution exactly as it would occur on the 20-pin device.

   User program must start at location $E00 for the MC6804J1 device (504 user program space EPROM bytes), and at location $C10 for the MC6804J2 device (1000 user program space EPROM bytes).

Emulation cabling for the MC6804J1/J2 devices consists of a 28-pin socket, cable harness/wiring, and a 20-pin DIL plug. This cable assembly facilitates the interconnection of the MC68704P2 device into a MC6804J1/J2 target system environment. The MC68704P2 device is inserted into the 28-pin socket (after being programmed), and the 20-pin DIL plug is connected directly into the target system MC6804J1/J2 socket.

MC68704P2 28-PIN SOCKET	MC6804J1/J2 20-PIN DIL PLUG	FUNCTION
Pin 1	Pin 1	V_{SS}
Pin 2	Pin 2	$\overline{IRQ}$
Pin 3	Pin 3	V_{CC}
Pin 4	Pin 4	EXTAL
Pin 5	Pin 5	XTAL
Pin 6	Pin 6	MDS
Pin 7	Pin 7	TIMER
Pin 8		10 kΩ pulldown
Pin 9		10 kΩ pulldown
Pin 10		10 kΩ pulldown
Pin 11		10 kΩ pulldown
Pin 12	Pin 8	PB0
Pin 13	Pin 9	PB1
Pin 14	Pin 10	PB2
Pin 15	Pin 11	PB3
Pin 16	Pin 12	PB4
Pin 17	Pin 13	PB5
Pin 18	Pin 14	PB6
Pin 19	Pin 15	PB7
Pin 20		10 kΩ pulldown
Pin 21		10 kΩ pulldown
Pin 22		10 kΩ pulldown
Pin 23		10 kΩ pulldown
Pin 24	Pin 16	PA4
Pin 25	Pin 17	PA5
Pin 26	Pin 18	PA6
Pin 27	Pin 19	PA7
Pin 28	Pin 20	$\overline{RESET}$

D.3 EPROM ERASING

The MC68704P2 EPROM MCU device is erased by the exposure of a high-intensity ultraviolet (UV) light with a wavelength of 2537 Angstrom (Å). The recommended dose (UV intensity × exposure time) is 15 Ws/cm^2. UV lamps should be used without shortwave filters, and the EPROM MCU device positioned about one inch from the UV lamps.

D.4 PROGRAMMABLE OPERATION

The programming board shown in Figure D-1 along with a software program (stored in 2K EPROM) is designed to perform in four modes of operation. All that is required to program the MC68704P2 EPROM MCU is the programming module, 2K EPROM, and a +5 Vdc power supply.

The MC68704P2 EPROM MCU programming module was designed to utilize either MC68705P3 or MC6805P2 MCUs. Four jumper pads (J1A, J1B, J2, and J3) are provided on the module to facilitate MCU interconnection as follows:

a. MC68705P3 MCU — Install insulated jumper wire between J1A and J3.
b. MC6805P2 MCU — Install insulated jumper wire between J1B and J2.

Programming module operation is identical when using either the MC68705P3 or MC6805P2 MCU.

D.5 PROGRAMMING MODES OF OPERATION

The programming modes of operation are as follows:

a. Zero check
b. Program
c. Verify
d. Test

D.5.1 Zero Check

The zero check mode of operation allows the user to determine if the EPROM MCU is erased (blank). The erased value is $00 (hexadecimal). Upon completion of the zero check operation, the user is notified of the results via the module zero check LED (labeled "Z"). This mode of operation should be performed prior to any programming operation.

D.5.2 Program

The program mode of operation will store the data code located in the 2K EPROM into the EPROM MCU. Each byte programmed is also verified against the 2K EPROM contents. Upon completion of the program operation, the user is notified of the results via the module program LED (labeled "P").

D.5.3 Verify

The verify mode of operation compares the data code stored in the EPROM MCU against the 2K EPROM contents. Upon completion of the verify operation, the user is notified of the results via the module verify LED (labeled "V").

D.5.4 Test

The test mode of operation tests the hardware operation of the serial to parallel conversion circuits (74LS164s), octal transparent latches (74LS374s), and the octal buffer (74LS241) which represents the majority of the interconnecting circuitry between the module MC68705P3 MCU and the MC68704P2 MCU programming socket.

D.6 PROGRAM MODE SELECTION

The programming module operating modes are selected by the placement of the mode select switches S1 and S2. The functions of these switches are as follows:

S1	S2	MODE
OFF	OFF	Test
OFF	ON	Zero Check
ON	OFF	Program
ON	ON	Verify

After selecting the initial mode, reconfiguration of switches S1 and S2 (to any mode) followed by the placement of the RESET switch S3 to the ON and OFF positions will initiate a new mode of operation.

D.7 PRELIMINARY PROCEDURES

Prior to performing any programming operations, the following steps are performed:

1. Place mode select switches S1, S2, and POWER switch S4 to the OFF positions, and the RESET switch S3 to the ON position.

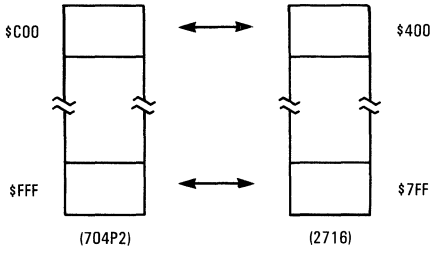
2. Connect +5 Vdc power supply to the programming module terminals labeled +5 and GND.

3. Install pre-programmed 2K EPROM device into the Zero Insertion Force (ZIF) 24-pin socket U10. Code stored in pre-programmed device is as follows:

Address	Contents
$012 — $017	Option bytes*
$018 — $05F	User data space
$400 — $7F7	User program**
$7FC — $7FD	IRQ vector**
$7FE — $7FF	Restart vector**

NOTES

1. *Mask option register (MOR) is located at $017. Programming board requires that locations $012-$016 be programmed with the same values as the MOR.
2. **User program is assembled at $C00-$FFF and the resulting object code is offset by $800 and programmed into the 2716 EPROM at locations $400-$7FF.



4. The MC68704P2 EPROM MCU device should be erased as described in paragraph D.3.

D

D.8 OPERATING PROCEDURES

After completion of the preliminary procedures, perform the following procedural steps:

1. Insert erased MC68704P2 EPROM MCU into the Zero Insertion Force (ZIF) 28-pin programming socket U14.

2. Place POWER switch S4 to the ON position. Place RESET switch S3 to the ON position, and then to the OFF position. When S3 is placed to the OFF position, the hardware test of the programming module is initiated.
   a. If all LEDs remain illuminated, the module is operating correctly and the user proceeds to step 3.

   b. If all three LEDs flash for approximately 4 seconds, a problem exists with the progamming module. Module POWER switch is placed to the OFF position, and the EPROM MCU device is removed from the programming socket. Upon fixing the module malfunction, the user proceeds to step 1.

3. Place mode select switch S2 to the ON position. Place RESET switch S3 to the ON position, and then to the OFF position. This step initiates the zero check of the EPROM MCU.
   a. If EPROM MCU is completely erased, the zero check LED (labeled "Z") will illuminate continuously, and the user proceeds to step 4.

   b. If EPROM MCU is not completely erased, the zero check LED will flash. Module POWER switch is placed to the OFF position, and the EPROM MCU device is removed from the programming socket and re-erased. Upon completion of the EPROM MCU erasing, the user places switch S2 to the OFF position and proceeds to step 1.

4. Place mode select switches S1 and S2 to the ON and OFF positions, respectively. Place RESET switch S3 to the ON position, and then to the OFF position. This step initiates the programming of the EPROM MCU. The EPROM MCU is programmed from the pre-programmed 2K EPROM residing in socket U10. EPROM MCU programming takes approximately two minutes to be completed.
   a. If no errors are encountered during the programming sequence, the program LED (labeled "P") will illuminate continuously, and the user proceeds to step 5.

   b. If errors are encountered, the program LED will flash. Module POWER switch is placed to the OFF position, and the EPROM MCU device is removed from the programming socket and re-erased. Upon completion of the EPROM MCU erasing, the user places switch S1 to the OFF position and proceeds to step 1.

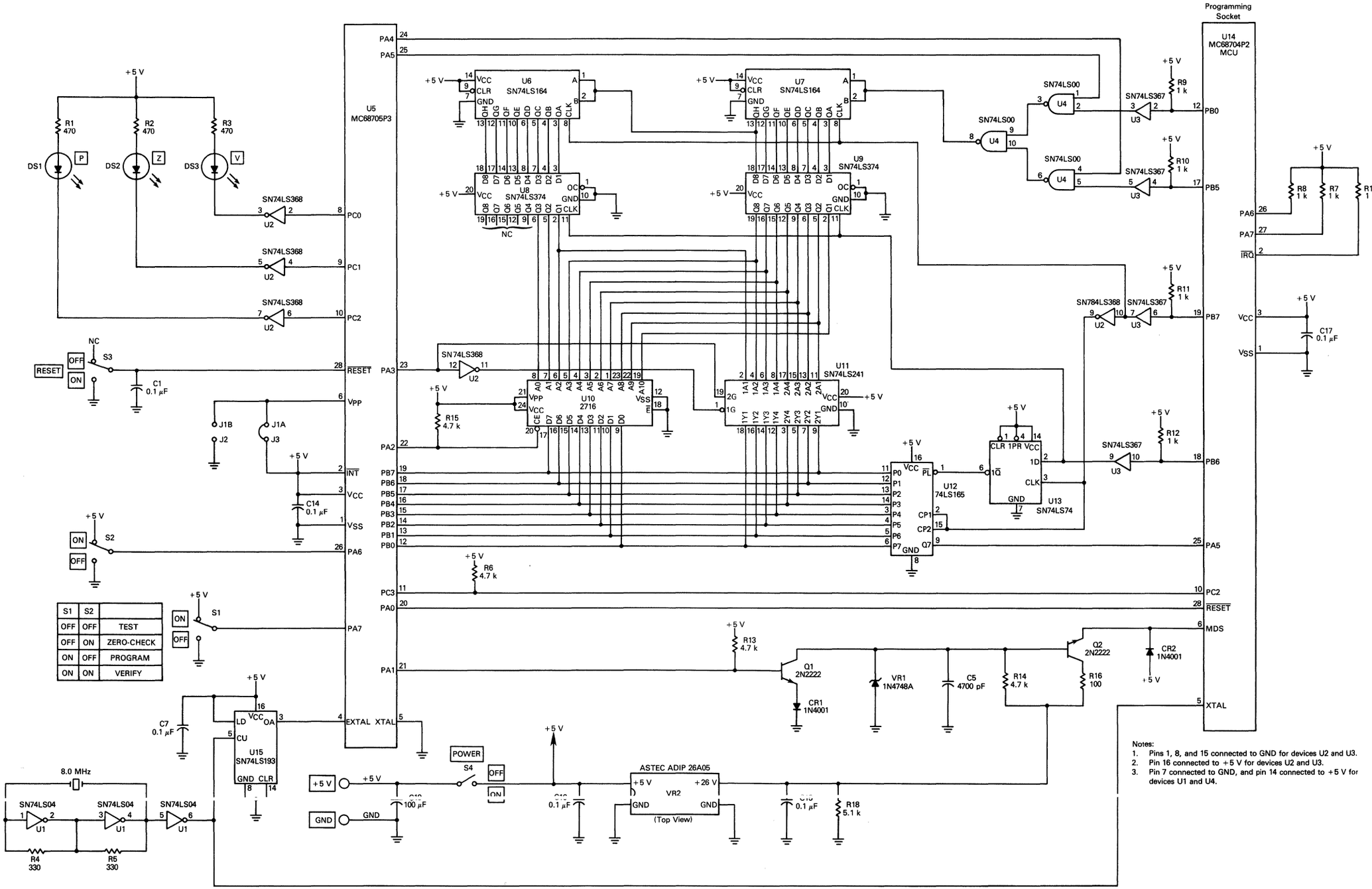
5. Place mode select switch S2 to the ON position. Place RESET switch S3 to the ON position, and then to the OFF position. This step verifies the EPROM MCU programming operation just performed. The EPROM MCU contents is compared against the code stored in the 2K EPROM device. Verification process takes approximately 4 seconds.
   a. If a valid comparison is made, the verify LED (labeled "V") will illuminate continuously. The programming operation is now complete. Module POWER switch is placed to the OFF position, and the EPROM MCU device is removed from the programming socket.

   b. If a mismatch is detected, the verify LED will flash. Module POWER switch is placed to the OFF position, and the EPROM MCU device is removed from the programming socket and re-erased. Upon completion of the EPROM MCU erasing, the user places both switches S1 and S2 to the OFF position and proceeds to step 1.

D.9 PROGRAMMING SOFTWARE

The MC68704P2 programming board requires programming software to perform the zero check, program, verify, and test operations. The MC68704P2 EPROM MCU device does not have an on-chip program capability (bootstrap-loader program stored in mask ROM). Therefore, additional programming software is required to program the MC68704P2 EPROM MCU device in conjunction with the programming board.

An application note has been prepared that describes the programming board in conjunction with a developed software program. Refer to Motorola Application Note AN942, MC68704P2 8-Bit EPROM Microcomputer Programming Module, for additional information.



S1	S2	
OFF	OFF	TEST
OFF	ON	ZERO-CHECK
ON	OFF	PROGRAM
ON	ON	VERIFY

Notes:
1. Pins 1, 8, and 15 connected to GND for devices U2 and U3.
2. Pin 16 connected to +5 V for devices U2 and U3.
3. Pin 7 connected to GND, and pin 14 connected to +5 V for devices U1 and U4.

Figure D-1. MC68704P2 EPROM MCU Programming Board/Circuitry

Literature Distribution Centers:

USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive, Blakelands Milton Keynes, MK145BP, England.
ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.; P.O. Box 80300; Cheung Sha Wan Post Office; Kowloon Hong Kong.

MOTOROLA

A19564 PRINTED IN USA 5-86 IMPERIAL LITHO C39830 18,000

ADI1248